

SAMOVAR

User manual

Firmware 7.00

Current version: <https://www.samovar-tool.ru/doc/en/manual/>
Built 01.10.2026

Contents

Where to start

- Assembly guide for the SAMOVAR complex
- Preparing the ESP32 for upload
- Initial sketch upload
- Connecting to Wi-Fi
 - Checking that the installation is correct
- Sketch setup*
- Updating the sketch
- Remote control
 - Mobile apps and website
- Voltage regulators used
- Using a boost heating element
- Cooling
- Emergency button. Additional emergency sensors
- LCD display and the universal button

Web interface. Main page

- Settings. Main
 - Temperature
 - Pump
 - Beer
 - Cheese
 - NBK
 - Other
 - Saving settings
- Additional tab
- System messages
- Logs and charts in the cloud
- Local logs and charts

Samovar operating modes

- First distillation mode
 - Distillation program
- Wash column
- Continuous wash column
- Rectification mode
 - Take-off program in Rectification mode
 - Editing the take-off program
 - Program calculator
 - Heads take-off
 - Hearts take-off
 - Pre-flooding
 - Tails take-off
 - Pause
- Beer mode
 - Mashing program
 - Stirrer, pump, temperature
 - Sparge water
 - Fermentation program
 - Additional tab
 - Loading BeerXML recipes
 - Beer brewing journal on the website and in the apps
 - Sous-vide
- Cheesemaking mode
 - Cheesemaking program
 - Cheese recipe catalog from the website
 - pH calibration
- Working in manual mode
 - Manual take-off rate setting
 - Manual voltage setting of the regulator
- Working in monitoring mode

Extending capabilities

- I2CStepper
- Additive dosers (Beer and Cheese)
- Autoclave control
- About the Lua scripting language
 - Script examples

Questions and answers. FAQ

Links. Practical tips

Version history

Where to start

The goal of Samovar is to combine various ready-made devices and make controlling them as simple as possible within a single concept.

The main concept of Samovar: through automation, to relieve the user of the routine operations involved in controlling brewing, distillation, rectification and so on.

Safety comes first in Samovar. Any event outside the permitted limits is stopped by Samovar immediately. At the same time, thanks to the open source code and the ability to use lua scripts, Samovar is flexible enough to adapt its parameters to almost any task.

What is the minimum configuration in which you can build Samovar?

It is enough to take an ESP32 microcontroller and load a program (sketch) into it. Then let the ESP32 connect to your WiFi network, and the Samovar interface will appear on your computer. The ESP will be powered over the USB connector. But it will not be able to show any data. **Nor will it be able to control equipment**, of course.

Now you need to connect Samovar to the equipment.

See [Assembly guide for the SAMOVAR hardware and software complex](#) (opens in a new window)

Connect the temperature sensors first. And since the computer is usually in one place and the column in another, you can power the ESP from any phone charger.

This is where Samovar already turns into your adviser. It will show on the monitor and send to your smartphone messages: boost complete, turn on the water, stabilization complete, temperatures exceeded, pause, and so on. And at the same time you see all the temperatures you need. If vapor breaks through, a message appears, there are charts, you can analyze the behavior of the column, and so on.

But stability and safety are up to you. Samovar only advises, while you turn the knobs and "taps". It starts to get tiresome. Something needs to be automated.

As an example, let us take a beginner's standard equipment.

A 3 kW heating element and an ordinary thyristor regulator with a knob...

Let us automate the boost. In the heating element assembly, leave 2 elements on the regulator and connect one to a 16 A solid-state relay (SSR). Connect the relay control to the corresponding pin (the fourth relay channel, RELE_CHANNEL4 — D13)*.

Switch on. Set on the regulator only the voltage needed for normal column operation, say 200 volts, then give Samovar the "Turn on heating" («Включить нагрев») command, and in addition to the regulator it will connect the boost heating element at full power through the SSR. When vapor reaches the column and the sensor at the top of the column detects it, Samovar switches off the boost heating element, leaving only the regulator.

So you no longer need to watch and wait for the boiler to boil in order to lower the voltage yourself. Samovar will do that.

But again, constantly turning the regulator knob to stabilize the column is no great pleasure... Especially if rectification lasts about 20 hours.

So, to save your nerves and health, it is highly desirable first of all to get a [stabilizer](#) (RMV-k, KVIC, RM) or to build a SEM_AVR regulator yourself.

And now previously unavailable possibilities open up:

1. Automatic power control depending on the running program, including constantly holding the column in the pre-flooding state**, switching the heating off by boiler temperature and, most importantly, automatically switching the heating off if vapor breaks through or the water temperature is exceeded!
2. And of course, stable heating.

In this way, step by step, with the help of Samovar, you can automate almost the entire process by gradually connecting other devices. Each new device opens up new possibilities.

Keep in mind that connecting an SSR or a stabilizer is probably the very limit of what you can do with a phone charger. Any further progress will require a more serious power supply.

* The ESP32 outputs 3.3 volts with limited current, so not every solid-state relay will work "out of the box".

** When a level sensor and certain types of regulators are used.

Assembly guide for the SAMOVAR complex

The documentation for the Samovar hardware and software complex consists of two separate parts:

- The guide to programming, setup and operation (this is the one you are in right now) and
- The guide to building the Samovar complex yourself.

Below is the list of sections of the assembly guide, but since it is still being filled in, links to some sections may be out of date.

It is recommended to consult the [table of contents](#) directly in the [guide to building it yourself](#) for the Samovar complex.

- [Where to start](#)
- [Mounting the ESP32](#)
- [Minimal configuration](#)
- [Buzzer](#)
- [Atmospheric pressure sensors](#)
- [Temperature sensors](#)
- [Voltage regulators in use](#)
- [Overview of regulators](#)
- [How to connect a regulator to Samovar wirelessly](#)
- [Heating element](#)
- [Using a boost heating element](#)
- [Contactor. RCD. Emergency shutdown](#)
- [Cooling](#)
- [Using a diaphragm pump](#)
- [Using a centrifugal pump](#)
- [Without a pump](#)
- [Reflux level sensor](#)
- [Driver. Stepper motor.](#)
- [Distribution by fractions](#)
- [Emergency button. Additional emergency sensors](#)
- [Recommended configuration](#)
- [LCD display, encoder, button](#)
- [How to connect a relay module correctly](#)
- [Extended configuration](#)
- [Meter](#)
- [I2CStepper](#)
- [Additional features](#)
- [NBK \(continuous wash column\)](#)
- [Building and flashing with PlatformIO](#)

Preparing the ESP32 for upload

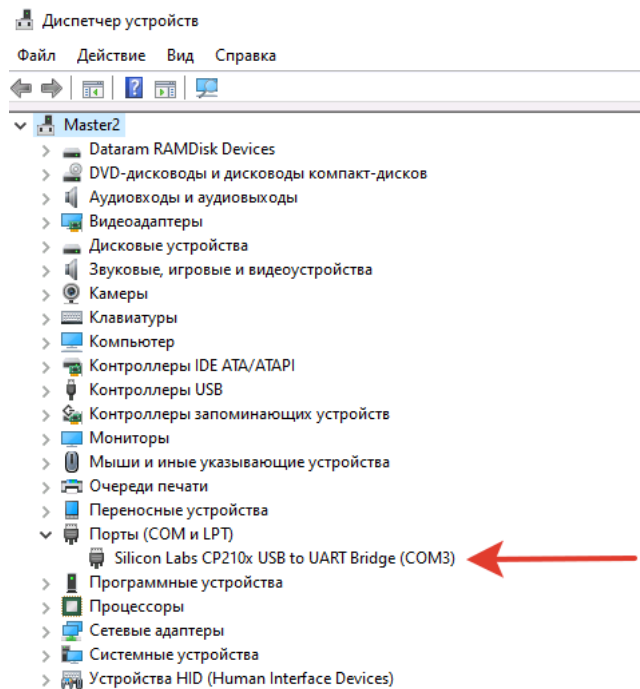
Samovar 7.00: the instructions below using the Arduino IDE remain a valid installation method. As an equal option you can use the configurator: flash_windows.bat on Windows (only 64-bit Windows 10/11) or flash_macos_linux.sh on macOS/Linux. The configurator installs PlatformIO and the required ESP32 core version itself; the Arduino IDE is not needed for it. See [Building and flashing with PlatformIO](#). The USB controller driver (below) is needed in any case.

If you have an ESP32, you can already upload the sketch and test how Samovar works.

First, simply connect the ESP32 to a USB port of the computer.

The ESP32 board has an onboard USB and a **CH340/CH341** or **CP210x** controller (this is usually stated on the product page)

Look in "This PC > Properties > Device Manager - Ports (COM & LPT)" («Этот компьютер > Свойства > Диспетчер устройств - Порты (COM и LPT)»); a new port should appear.



If the computer has detected new hardware but has not installed the driver, the new port will not appear. You need to download the drivers for the controller on your ESP32 board and install them manually.

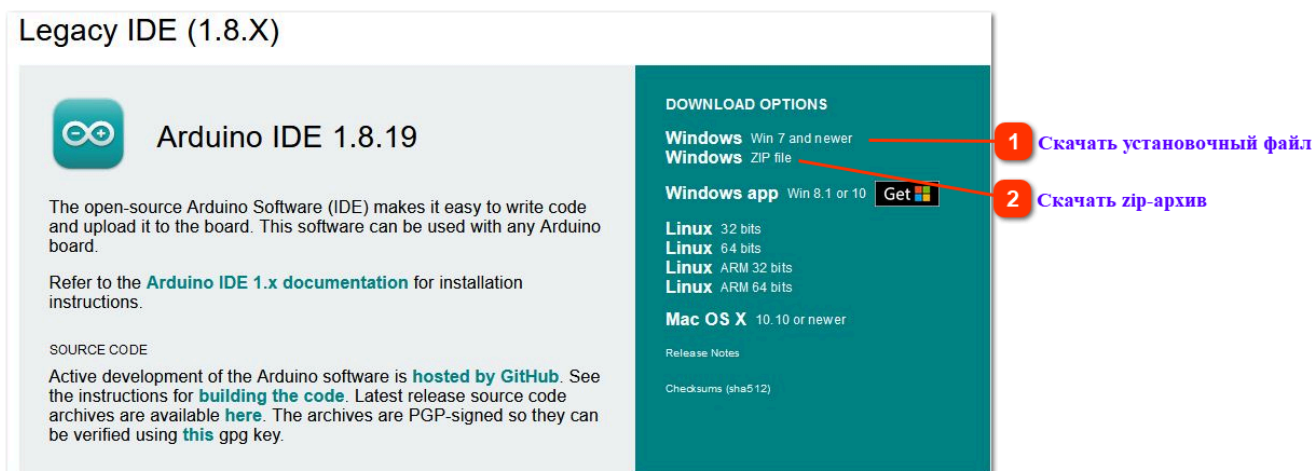
Download the driver [CH340/CH341](#) (Windows, Mac, Linux)

Download the drivers [210/341](#) from our website, Windows only.

To upload the sketch to the ESP32, you need to install the Arduino IDE. It is used as a sketch editor, as a sketch compiler, and also for uploading the compiled program to the ESP32..

Below and further on, a method of the initial installation of the Samovar sketch to the ESP32 is offered that will give a guaranteed positive result.

We download and install the Arduino IDE [version 1.8.x](#).



- 1 **Download the installer**
The installer for installing in the operating system
- 2 **Download the zip archive**
For running without installing in the OS

Whichever option you choose, you will be asked to thank the developers; that is up to you.

If you just want to download the program, press the "JUST DOWNLOAD" button. Another window opens offering to subscribe to the news; press the "JUST DOWNLOAD" button again.

A file will be downloaded. If you chose the zip archive, unpack it anywhere on the computer (it is advisable to avoid Cyrillic characters in the directory path).

If you chose to install the program, it is recommended not just to install it into the default directory but to add the version number to the directory name. This will make later use easier.

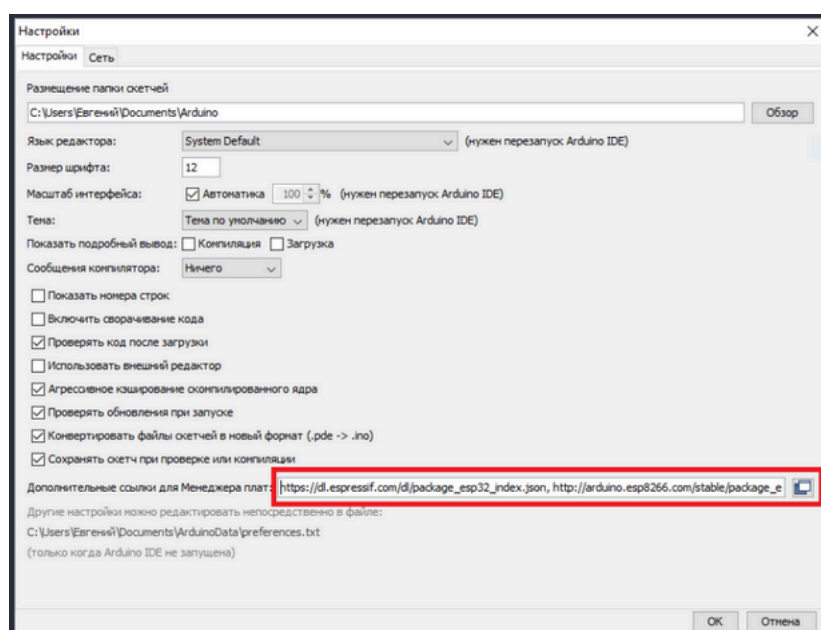
For example.

C:\Program Files(x86)\Arduino1_1_18;

C:\Program Files(x86)\Arduino1_1_16

After installing the Arduino IDE, it must be taught to work with the ESP32; by default it cannot.

- Go to the unpacked or installed folder and run the file *arduino.exe*
- In the Arduino IDE, open the "**Preferences**" («Настройки») window. To do this, click "**File**" > "**Preferences**" («Файл» > «Настройки»).
- Paste the link https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json into the field "**Additional Boards Manager URLs**" («Дополнительные ссылки для менеджера плат»), as shown in the screenshot below.



Note:

If this field already contains a link to the descriptions of other boards, the links can be separated with commas. Like this:
https://raw.githubusercontent.com/esp8266/arduino-esp8266-gh-pages/package_esp8266_index.json,
http://arduino.esp8266.com/stable/package_esp8266com_index.json

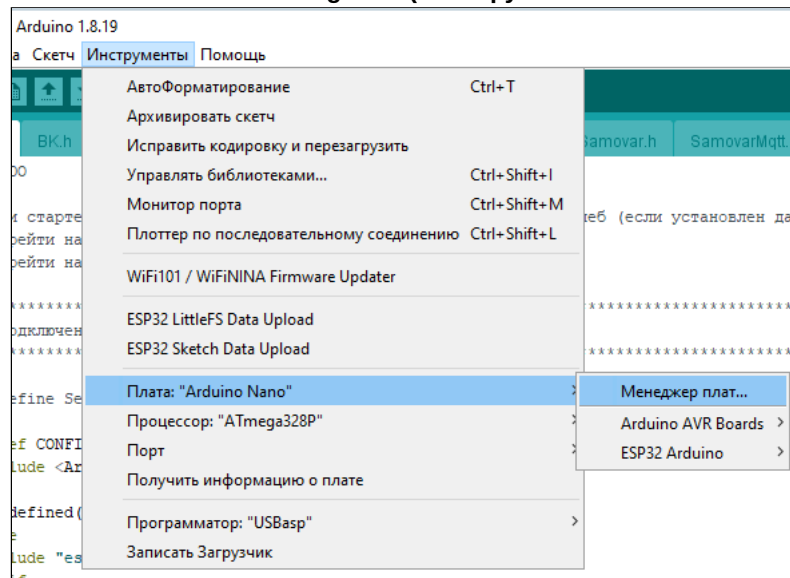
Attention!

If this field contains the link https://dl.espressif.com/dl/package_esp32_index.json, it must be removed.

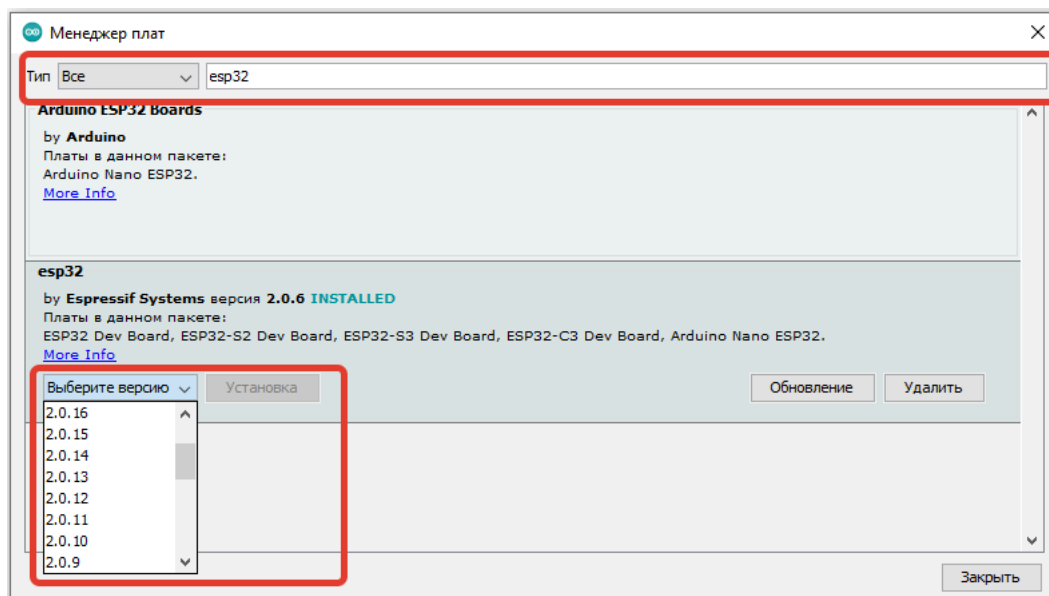
- Then click the “OK”.

Next, in the Arduino IDE open the “Boards Manager” («Менеджер плат»).

To do this, click “Tools” > “Board” > “Boards Manager...” («Инструменты» > «Плата» > «Менеджер плат...»).



A new window opens. In the search field type “esp32”. The add-on (or another commonly used name, SDK) for ESP32 will be found: “ESP32 by Espressif Systems”.



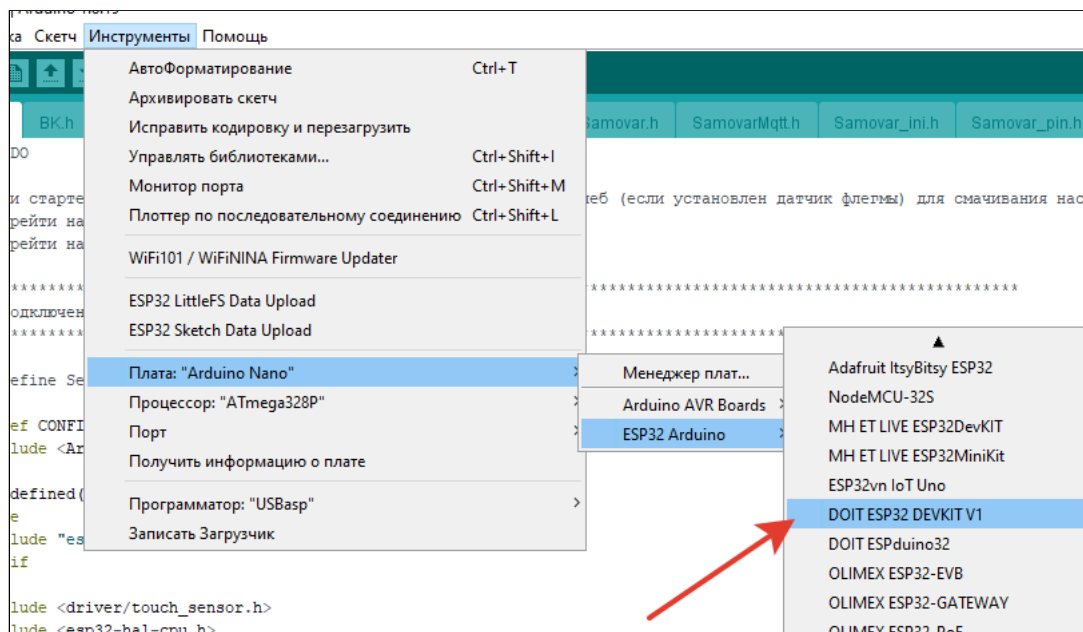
In the drop-down menu select version 2.0.17, press - Install («Установка») and wait for the installation to finish

Note:

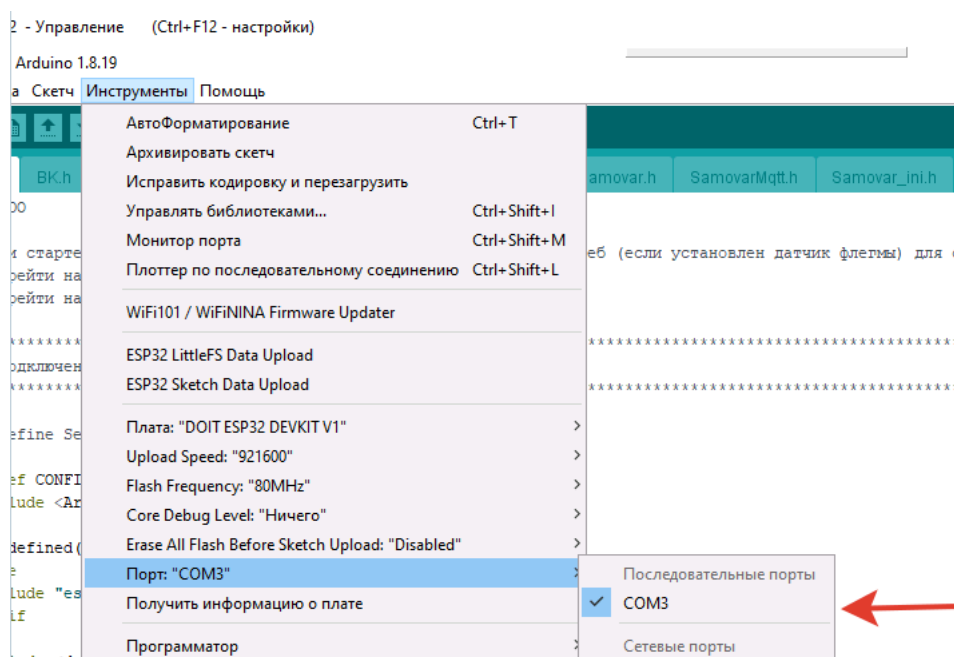
If the add-on is already installed but its version is different, remove it and install 2.0.17

The PlatformIO configurator uses exactly version 2.0.17 too. The sketch also builds with core 3.x, but the WiFi update (USE_UPDATE_OTA) is automatically disabled in it, and the Samovar partition table must be added to the core manually (see [Sketch update](#)).

Restart the Arduino IDE, then select the required board in the menu **"Tools" > "Board > ESP32 Arduino"** (**«Инструменты» > «Плата > ESP32 Arduino»**).
In our case it is **"DOIT ESP32 DEVKIT V1"**.



Check that in **"Tools > Port > Serial ports"** (**«Инструменты > Порт > Последовательные порты»**) the port the board is connected to is selected.

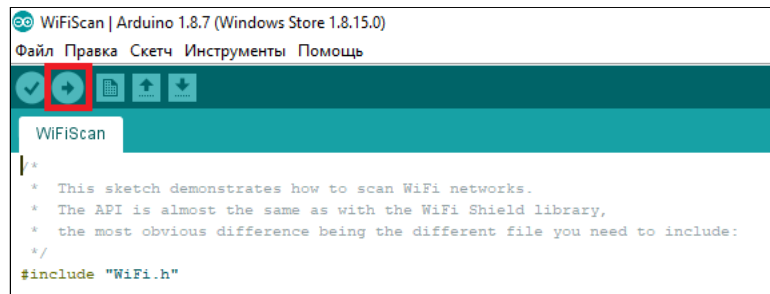


Now we need to check that everything is done correctly.

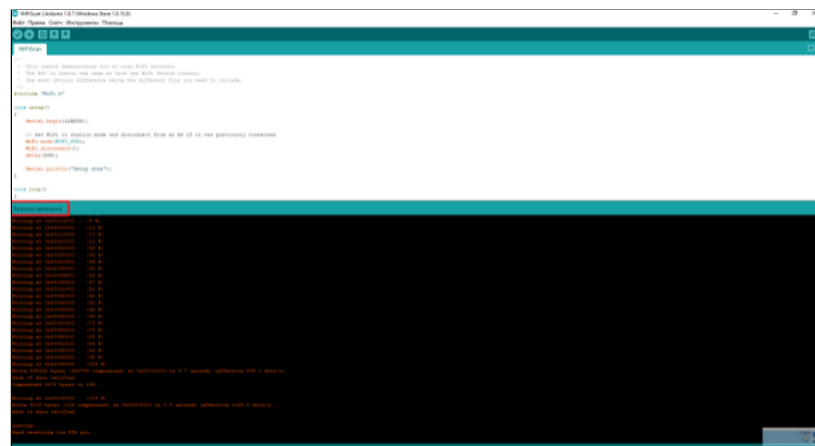
In the Arduino IDE, open the example sketch **"WiFi Scan"**.

To do this, click **"File" > "Examples" > (subsection: examples for DOIT ESP32 DEVKIT V1) > "WiFi" > "WiFi Scan"** (**«Файл» > «Примеры» > ... > «WiFi» > «WiFi Scan»**); the sketch opens in the editor window.

Click the **"Upload" («Загрузка»)** button in the Arduino IDE. Wait a while for the code to compile and upload to the ESP32 board. The upload button icon looks like this:



When the sketch is uploaded to the ESP32 successfully, you should see the message **"Done uploading"** («Загрузка завершена»)



In the Arduino IDE open **"Tools" > "Serial Monitor"** («Инструменты» > «Монитор порта») and set the data rate to 115200 baud.
 Reconnect the ESP32 to the USB port. As a result you should see in the serial monitor all the Wi-Fi networks around the ESP32 board.
 This completes the setup and check of the ESP32.
 Now you can proceed to the [first upload of the Samovar sketch to the ESP32](#)

If the ESP32 was used before in other projects or the Samovar sketch uploads incorrectly, you need to erase the flash memory of the ESP32 module

In the Arduino IDE

The Arduino IDE has an "Erase Flash" option. To erase the flash memory:

1. Select the suitable ESP32 board from the "Tools" → "Board" menu («Инструменты» → «Доска»).
2. Select the correct COM port from the "Tools" → "Port" menu.
3. Go to the "Tools" → "Erase Flash" menu. In the drop-down menu choose "All Flash Contents" («Все содержимое флэш-памяти»).
4. Press the "Erase Flash" option («Стереть флэш»). The progress is shown in the output console at the bottom of the IDE. When the process finishes, a confirmation message appears.

With ESPTool

ESPTool is an open-source utility based on Python. To erase the flash memory of the ESP32 module:

1. Open a terminal window on the computer and hold the BOOT button on the ESP32 module.
2. Enter the command in the terminal, replacing COMx with the real COM port of the ESP32:
`python -m esptool --port COMx erase_flash.`
3. Press Enter.
4. When the erasing process starts in the terminal, you can release the BOOT button. After a few seconds the ESP32 flash memory will be completely erased.

With the configurator

Run `flash_windows.bat` (`flash_macos_linux.sh`), select the COM port of the board and press the "Erase flash completely" («Полностью очистить флэш») button. Confirm: the firmware, LittleFS and all saved settings will be deleted. The button works only over a USB connection.

Important: if after the message "Connecting..." new series of dots appear on the screen, it means the ESP32 module is not in bootloader/flashing mode. In this case repeat the process while holding the BOOT button while performing

Initial Sketch Upload

Samovar 7.00: on first start, Samovar 7.00 creates a profile with default values by itself if there are no saved settings yet. Restoring Samovar_default_settings.txt manually is not required.

Samovar 7.00: the initial installation through Arduino IDE is supported. The alternative way is to choose the board and parameters in the configurator, press "Flash" («Прошить»), then, for a new board, separately "Upload LittleFS" («Загрузить LittleFS»).

Before the first upload of the Samovar sketch to the ESP32, the computer and the board must be prepared; read about it in ["Preparing the ESP32 for Upload"](#)

When you install **Arduino IDE**, the program creates a new directory named **Arduino** in your documents. The default path is C:\User\Documents\Arduino. This directory needs to be prepared.

When installed, **Arduino IDE** may put nested directories into ...Documents\Arduino, including ones named libraries and tools. These directories need to be removed.

Warning, this is important!

To install the Samovar sketch correctly, you must use the libraries and tools that come in the archive with each Samovar release. Using libraries from earlier releases or from other sources will cause an error when uploading the Samovar sketch to the ESP32.

The recommendations below will help you avoid mistakes at first through inattention and the like. Later you can develop your own method.

Create a nested directory OLD in ...Documents\Arduino and drag (or cut and paste) the libraries and tools directories created by **Arduino IDE** during installation into the OLD directory.

If everything is done correctly, there should be no directories named libraries and tools in C:\User\Documents\Arduino\ . Any other directory names are allowed.

Now download the archive with the latest firmware from this [link](#) .

Warning!

It is recommended to download the archive from the link above. That is a finished and tested version . The resource also has [the Master version](#). This is a future version that is under development. It may not work correctly or may not work at all.

6.17 Latest

В настройках температуры добавилась кнопка - обновить датчики температуры. Можно по одному датчику подключать и сохранять в настройках. Включили один датчик, выбрали, к чему он относится, нажали сохранить, вернулись в настройки, подключили еще один датчик, выбрали, к чему он относится, нажали сохранить и т. д. Обновилась библиотека для работы с меню экрана. Оптимизация.

▼ **Assets** 2

ZIP 1 [Source code \(zip\)](#) last week

TAR 2 [Source code \(tar.gz\)](#) last week

1 **ZIP**
Download the archive in ZIP format

2 **TAR**
Download the archive in TAR format (RAR, GZ)

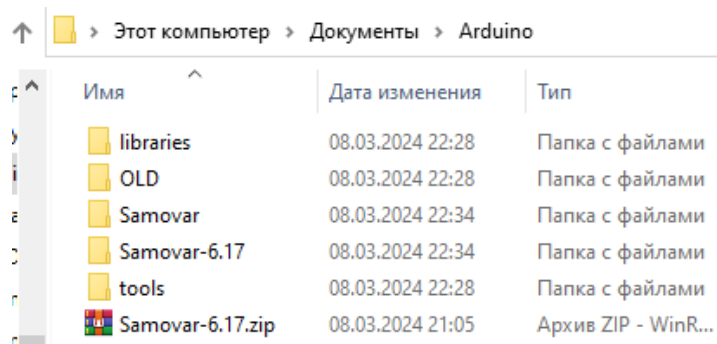
As a rule, the archive is saved to the "Downloads" directory of your browser.

The name of the downloaded archive: Samovar-X.XX, where X.XX corresponds to the version number (*all the examples below use version Samovar-7.00*).

Copy (or cut and paste) the downloaded archive into the directory C:\User\Documents\Arduino\ (your path may be different)

- Unpack the archive. A directory Samovar-7.00 appears.
- Go inside - there will be another directory Samovar-7.00. Rename it to Samovar.
- Move exactly this directory Samovar, by dragging (or cut - paste) one level up, into C:\User\Documents\Arduino\
- Go into the Samovar directory. Inside there are directories and files. Move the two directories libraries and tools by dragging one level up, into C:\User\Documents\Arduino\

As a result, the following directory structure should form in ...Documents\Arduino\:

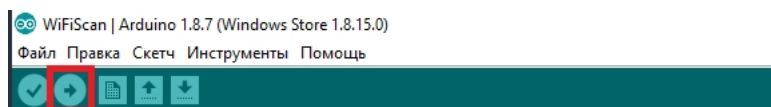


- libraries - the libraries
- OLD - old files
- Samovar - the Samovar sketch
- Samovar-7.00 - an empty directory (do not delete it, we will still need it)
- tools - the tools
- Samovar-7.00.zip - the archive with the original sketch.

To compile the sketch, **Arduino IDE** will use the directories: Samovar, libraries and tools.

Now the sketch is ready to be uploaded to the ESP32.

- Connect the ESP32 to a USB port.
- Start **Arduino IDE**.
- In **Arduino IDE** open the file **Samovar.ino** from the directory ...\\Documents\\Arduino\\Samovar\\.
- Upload the sketch to the ESP32 exactly as in the [upload of the standard example](#) when preparing the ESP32.



Clicking the arrow starts uploading the release to Samovar

Note.

- *Compiling the code can take a long time - do not worry, be patient.*
- *Ignore the lines highlighted in red in the upload monitor - this is normal. If a critical error occurs, the upload stops and an orange bar with the error message appears.*

Warning!

Some clones of the ESP32 board do not allow the sketch to be uploaded to the ESP without pressing the **BOOT** button. There are two ways to upload using the **BOOT** button.

- If you hold the **BOOT** button before powering the ESP, apply power, keep holding the button for 3 seconds and then release it, you can flash the ESP and all the settings made earlier are kept. In particular, the WiFi connection settings.

- If you apply power right away and give the "Upload" command, after the sketch is compiled and the upload to the ESP32 starts, the Arduino IDE console, after several red lines, shows the message:

"Connecting" after which dots start to appear.

At that moment press the **BOOT** button on the ESP32 board, and release it when the dots stop appearing.

Pressing **BOOT** only puts the board into bootloader mode and does not by itself erase the Wi-Fi settings.

Installing the Web Interface.

Warning!

Starting with version 6.0 the file system in use has changed. The move to a different file system was made to optimize Samovar's speed.

After the sketch is installed, the web interface must be installed locally.

Without disconnecting the ESP32 from the computer, in the "Tools" menu select:

- ESP32 LittleFS Data Upload - for versions 6.0 and higher
- ESP32 Sketch Data Upload - for versions below 6.0.

Press the "Upload" button in Arduino IDE again and wait for the installation to finish. On some ESP32 versions you also need to use the **BOOT** button to upload the interface (see above).

Warning! Important!

Recently, perfectly working clones of the ESP32 board have appeared, but if you select in the "Tools" menu > Board > ESP32 Arduino>**DOIT ESP32 DEVKIT V1**, the web interface is not installed. If you select the ESP32 **Dev Module** board, the interface uploads normally.

Warning! Important!

Starting with version 6.0, provided Samovar is connected to the internet, the interface is automatically downloaded and installed without your involvement.

If you need to roll back to an older version of the sketch (for versions 6.0 and higher), the interface also rolls back to the matching version by itself.

If there is no internet connection, the interface must be installed for each version yourself - that is, locally.

The firmware compares the interface version on the server (web.samovar-tool.ru, the directory of its own version) with the local file version.txt and downloads files only if they differ. The serial monitor then prints the lines WEB interface version and Local interface version. If there is less than 64 KB of free space in LittleFS, the update is cancelled with the message "Web interface update cancelled: low disk space" («Обновление веб-интерфейса отменено: мало места на диске»). If the file system did not mount (a new board, an erased flash), the firmware formats it by itself - the message "Failed to connect to the file system, formatting..." («Не удалось подключиться к файловой системе, форматируем...»).

Addendum.

But, as it turned out, something has changed and this does not work for everyone.

Starting with version 6.20 a parameter has been added to the file Samovar_ini.h:

```
//#define NOT_USE_INTERFACE_UPDATE //do not use the interface update at boot
```

When you upload the sketch for the first time, you need to uncomment it:

```
#define NOT_USE_INTERFACE_UPDATE //do not use the interface update at boot
```

The interface will then not be updated over the internet. It can be updated only manually. Later, while setting up the sketch, you can return the line to its original form (comment it out) and see whether the update over the internet works correctly.

Tip

It is recommended to do the initial interface installation locally, over USB. The point is that if the option to update the interface over the internet is enabled, downloading and updating the interface can take a long time. During that time the interface is not displayed.

Warning.

Local LittleFS upload is supported in Arduino IDE 1.8.x with the classic uploader and in Arduino IDE 2.x with the ESP32 LittleFS Data Upload plugin.

Addendum.

In the ESP32 by Espressif Systems board package version 3.0.0 and higher, it became possible to install the interface locally with a plugin for Arduino IDE 2. The instructions are [here](#). The plugin itself (the file arduino-littlefs-upload-1.1.7.vsix) is in the Samovar archive in the directory tools\ESP32LittleFS_Arduino IDE 2.X: copy it to the directory ~/.arduinoIDE/plugins/ (on Windows - C:\Users\<name>\.arduinoIDE\plugins), restart Arduino IDE, press Ctrl+Shift+P and select "Upload LittleFS to Pico/ESP8266/ESP32".

Note.

On some ESP32 versions, when flashing versions 6.0 and higher using ESP32 LittleFS Data Upload the interface is still not installed. Why this happens remains a mystery. If this is the case, try several times, alternately, to flash the ESP32 both with ESP32 LittleFS Data Upload and with ESP32 Sketch Data Upload. After each flashing, check whether the interface has appeared.

After the sketch and the web interface are uploaded, reboot Samovar with the power disconnected, and then [connect Samovar to your Wi-Fi network](#).

When Samovar is connected to your WI-FI network, open the Samovar interface on a computer by entering the IP address assigned to it, or the name **http://samovar.local**, in the browser address bar.

The recommended default values do not need to be loaded manually.

Samovar 7.00: if there is no saved profile yet, the firmware creates and saves the default settings itself. The file **Samovar_default_settings.txt** from the tools directory can be used for manual restoring, but it is not required for the first start.

To manually restore a previously saved profile, open "Settings → Other" («Настройки → Прочие»), press "Upload settings file" («Загрузить файл настроек»), select the file, press "Save" («Сохранить») and reboot Samovar. You can save your own profile with the "Download settings file" («Скачать файл настроек») button on the same tab.

Note.

If the upload of the sketch or of the Samovar interface goes wrong, you need to [erase the flash memory](#) of the ESP32 module and do the upload again.

The initial Samovar setup is finished - you can proceed [to setting up the file Samovar_ini.h](#) for your own equipment.

Connecting to Wi-Fi

If the SSID and password were entered in the configurator (the "Network" («Сеть») section, the "Wi-Fi SSID" («SSID Wi-Fi») and "Wi-Fi password" («Пароль Wi-Fi») fields) and the board memory does not yet hold another network, Samovar first tries to connect to them automatically. The serial monitor then shows the line **WiFi: initial credentials loaded from local configuration**. If a network was saved earlier, the saved data is used.

If the connection fails, the ESP32 creates the **Samovar** access point. Connect to it from a phone or computer. If the portal does not open by itself, open **http://192.168.4.1** in a browser, select your home network and enter its password. The same page has a **blynk token** field - you can enter the device token from the Samovar mobile app here right away (32 characters); it is saved in the Samovar settings. You can leave the field empty and enter the token later on the "Settings" («Настройки») page.

You have **6 minutes** to enter the data. If the setup is not completed, the device keeps running as a standalone Samovar access point with the password **SamApp123**; the interface will be available at **http://192.168.4.1**.

After a successful connection, you can see the IP address on the display, in the serial monitor (the **IP address:** line), with the **"Get IP"** («Получить IP») button in the configurator's "Serial Monitor" («Монитор порта») window, in the mobile app, or in the router's client list. Open this IP address in the browser of a device on the same local network. Instead of the address you can use the name **http://samovar.local** (mDNS; it does not work in all networks and on all devices).

To reset the saved network, hold the encoder button for at least two seconds during startup. A short press must not erase Wi-Fi. Other ways to reset:

hold the **BOOT** button (GPIO0) on the ESP32 board for at least two seconds right after power-up - the serial monitor shows **GPIO0 held low: erasing WiFi settings**;

on the LCD, select the **Reset WiFi** menu item;

in the web interface, open "Settings → Other" («Настройки → Прочие») and press the **"Reset WiFi settings"** («Сбросить настройки WiFi») button.

After the reset, Samovar reboots and brings up the Samovar access point again.

Checking that the installation is correct

This check confirms three things: the firmware has started, the ESP32 has connected to the network, and the web interface opens.

1. Open the serial monitor

- . Connect the ESP32 to your computer with a USB cable that carries data.
- . In the configurator, press "**Serial monitor**" («Монитор порта»): the log opens in a separate window. In the Arduino IDE, open **Tools** → **Serial Monitor**. In PlatformIO, open **PROJECT TASKS** → **selected environment** → **General** → **Monitor**.
- . Set the speed to **115200** baud - this is the speed of communication with the board.
- . Press RESET/EN on the ESP32, or briefly unplug and replug the USB.

2. Find the connection result

When connecting to your home network, the line **Connected to ...** appears, followed by **IP address:** Write down the address after the colon. If it fails to connect to the saved network, Samovar raises the Samovar access point with a setup portal for 6 minutes (see [Connecting to Wi-Fi](#)).

If Samovar has started as its own access point, **Started as WiFi AP** appears along with the address **192.168.4.1**. Connect to the Samovar network and open this address.

If the same startup lines keep appearing again and again, the ESP32 is rebooting. This is not normal operation: save the whole monitor text from the first startup line to the reboot.

3. Open the web interface

- . Your computer or phone must be connected to the same home network as Samovar, or directly to the Samovar access point.
- . Type **http://** and the IP address into the browser address bar, for example **http://192.168.2.100**. Use exactly HTTP, without the letter s.
- . The Samovar page should open, showing the version, the temperatures and the **Settings** («Настройки») button. Press Settings and make sure the page opens. If a router page or another device's page opens instead of the interface, the IP address is wrong. Look up the address again in the serial monitor, on the display, with the "**Get IP**" («Получить IP») button in the configurator's "Serial monitor" window, or in the router's client list. On the same network you can open **http://samovar.local**.

The web interface is updated by Samovar itself from the server for the installed firmware version; in the serial monitor these are the lines **WEB interface version** and **Local interface version**. If there is no network connection, **WEB interface update skipped: no WiFi connection** appears. For the first check, do not disable the update mechanism and do not replace individual files by hand.

Sketch setup*

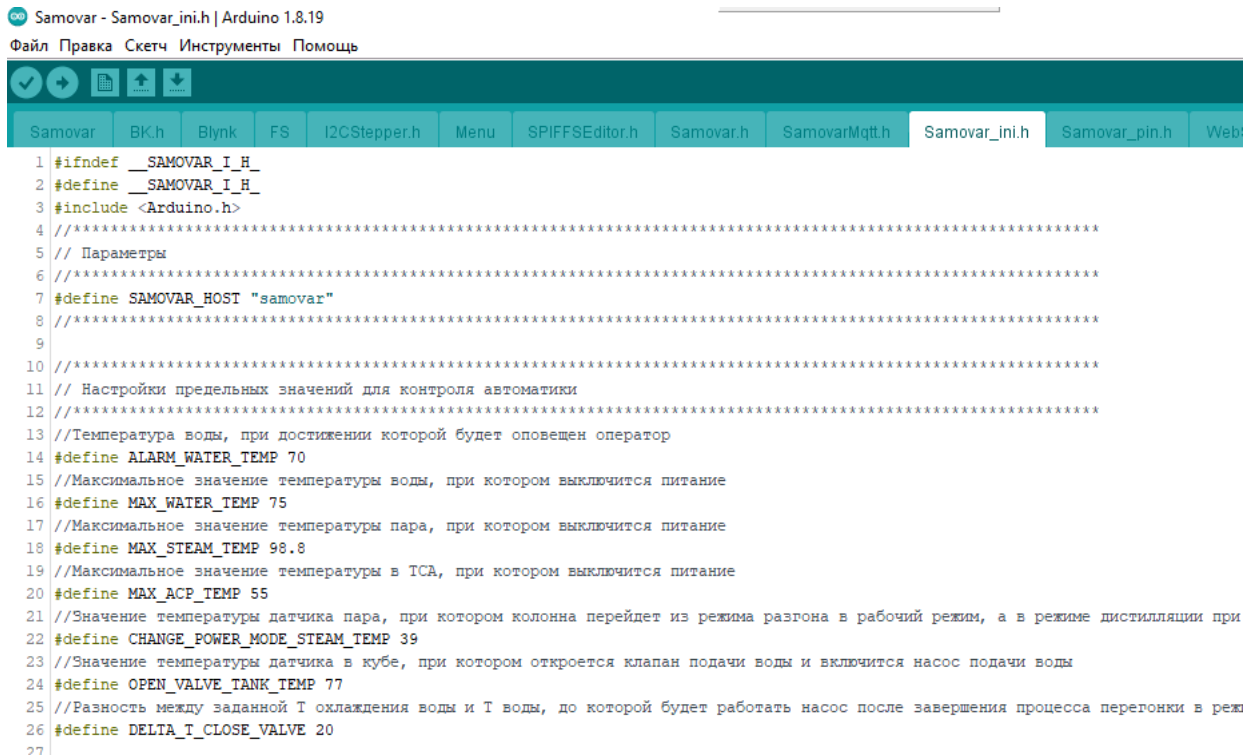
Samovar 7.00: the parameters can be set up either in the way described below for the Arduino IDE or through the configurator. The configurator saves the selected values in user_config_override.h, without changing the main Samovar_ini.h file. The firmware includes this file automatically, and its values override Samovar_ini.h; it can also be created by hand from the user_config_override.example.h template. In the configurator the parameters are arranged in sections (the "Section" ("Раздел») list): Main («Основные»), Temperatures («Температуры»), Pumps («Насосы»), NBK («НБК»), Regulator («Регулятор»), Equipment («Оборудование»), Rectification («Ректификация»), BK («БК»), Network («Сеть»), Sensors («Датчики»), Stepper motor («Шаровый двигатель»).

To set up the sketch for your equipment configuration, open it in the Arduino IDE and make changes in the Samovar_ini.h file

In the Arduino IDE, open the file **Samovar.ino** from the folder ...\\Documents\\Arduino\\Samovar\\.

This loads into the Arduino IDE all the files located in the ..\\Documents\\Arduino\\Samovar\\ folder

In the top row of the Arduino IDE, find the Samovar_ini.h tab and start editing.



```
1 #ifndef __SAMOVAR_I_H_
2 #define __SAMOVAR_I_H_
3 #include <Arduino.h>
4 //*****
5 // Параметры
6 //*****
7 #define SAMOVAR_HOST "samovar"
8 //*****
9
10 //*****
11 // Настройки предельных значений для контроля автоматики
12 //*****
13 //Температура воды, при достижении которой будет оповещен оператор
14 #define ALARM_WATER_TEMP 70
15 //Максимальное значение температуры воды, при котором выключится питание
16 #define MAX_WATER_TEMP 75
17 //Максимальное значение температуры пара, при котором выключится питание
18 #define MAX_STEAM_TEMP 98.8
19 //Максимальное значение температуры в TCA, при котором выключится питание
20 #define MAX_ACP_TEMP 55
21 //Значение температуры датчика пара, при котором колонна перейдет из режима разгона в рабочий режим, а в режиме дистилляции при
22 #define CHANGE_POWER_MODE_STEAM_TEMP 39
23 //Значение температуры датчика в кубе, при котором откроется клапан подачи воды и включится насос подачи воды
24 #define OPEN_VALVE_TANK_TEMP 77
25 //Разность между заданной Т охлаждения воды и Т воды, до которой будет работать насос после завершения процесса перегонки в реж
26 #define DELTA_T_CLOSE_VALVE 20
27
```

To keep a line of code from being compiled and therefore executed by Samovar, put the characters // in front of the line (comment it out).

And vice versa, to make a line run, remove the characters // (uncomment it)

It is recommended to exclude exactly those lines that relate to equipment you do not have, rather than switching off in a row all the parameters that you think you will not need.

For convenience, the code in the file is divided into blocks.

#define SAMOVAR_HOST "samovar"

Samovar's name on the network. The device is available under it as http://samovar.local (mDNS), and under the same name it is visible in the network ports of the Arduino IDE and the configurator when updating over WiFi. In the configurator: section "Main" («Основные»), "Device name on the network" («Имя устройства в сети»).

Block

//Selecting the ESP32 board in use

//*****

//#define BOARD DEVKIT

//#define BOARD LILYGO

//#define BOARD ESP32S3

//*****

Skip this. The firmware determines the board type by itself from the board selected in the Arduino IDE (DOIT ESP32 DEVKIT V1 / ESP32 Dev Module - DEVKIT, ESP32S3 Dev Module - ESP32S3), or from the board selected in the configurator. If nothing is detected, DEVKIT is used.

You can start editing values and enabling/disabling lines of code from this line:

//Water temperature at which the operator will be alerted

#define ALARM_WATER_TEMP 70

Here 70 is the temperature in degrees Celsius. When it is reached, a message about the critical coolant temperature (hereinafter CT) is issued. If a buzzer is used and event notification is enabled in the interface, Samovar also alerts with a sound. The alerts keep coming until the temperature drops. If you want the alert to come at a different temperature, change the number 70.

//Maximum water temperature at which the power will be turned off

#define MAX_WATER_TEMP 75

When the CT reaches 75°, Samovar gives the command to turn the heating off.

//Maximum steam temperature at which the power will be turned off

#define MAX_STEAM_TEMP 98.8

When the steam sensor temperature reaches 98.8°, Samovar gives the command to turn the heating off.

//Maximum temperature in the TCA at which the power will be turned off

#define MAX_ACP_TEMP 75

When the TCA sensor temperature reaches 75°, Samovar gives the command to turn the heating off. (see also [TCA sensor setpoint](#))

//Steam sensor temperature at which the column switches from boost mode to working mode, and in distillation mode, with a steam sensor installed, detects the start of boiling

#define CHANGE_POWER_MODE_STEAM_TEMP 39

When the steam sensor temperature (or the temperature of the upper sensor in the column section) reaches 39°, Samovar gives the regulator the "stop boost" command and sets on the regulator the voltage specified in the first line of the take-off program. In distillation mode, Samovar uses the same steam sensor temperature to detect the start of boiling. In the configurator: "Temperatures" («Температуры»), "Boost transition temperature, °C" («Температура перехода из разгона, °C»).

//Boiler sensor temperature at which the water supply valve opens and the water supply pump turns on

#define OPEN_VALVE_TANK_TEMP 77

At a boiler temperature of 77°, the command to open the water supply valve is given, the PWM control signal to turn on the pump is sent at the same time, and a message is sent to the operator that the water must be opened.

The message is sent in any case, even if neither a pump nor a valve is used, or the water is already open.

#define DELTA_T_CLOSE_VALVE 20

//The difference between the set water cooling T and the water T, down to which the pump will keep running after the distillation process ends in distillation and rectification modes. The value is in degrees C°, it can be changed. Disabling it is strongly discouraged. (for details see [here](#))

//Lower value (in percent) for the PWM regulator for the pump. Choose a value that ensures water flow

#define PWM_LOW_VALUE 10

As is well known, a motor needs a higher voltage to start than to keep running. This line sets the minimum possible value (in percent) at which the motor can keep the water supply pump running without stopping.

//Lower value (in percent) at which the PWM regulator for the pump starts working. Choose a value that is guaranteed to start the motor

#define PWM_START_VALUE 40

The value in % of the rated voltage at which the pump motor is guaranteed to start.

That is, if your pump (provided that it is already installed in the system and has liquid in it) reliably starts when a voltage of 6 volts or more is applied, and does not start at less, then with its rated voltage of 12 volts this is 50% of the rated value. So the number 40 must be replaced with 50.

//Correction for the servo rotation angle.

int8_t servoDelta[11] = {0, -2, -3, -4, -3, -2, 0, 0, 0, 0, -2};

Correction of the rotation angle of the take-off table servo for precise positioning. It is given in degrees. Each position is corrected individually. It does not depend on the correction of neighboring positions.

For example, if the rocker arm has "overshot" position No. 2 and we set minus 10°, this value (-10) does not need to be taken into account when correcting position No. 3.

//Difference between the target temperature and the current temperature (for Beer and sous-vide modes) until which heating will be at full power (in boost mode).

#define HEAT_DELTA 1

For Beer and sous-vide modes. The value is given in degrees Celsius. This parameter relates to the main heater. It means that heating is at full power as long as the difference between the actual temperature and the one set in the mashing program line is greater than 1°. As soon as the difference is less than or equal to the one specified in #define HEAT_DELTA, the regulator is told to reduce heating so as to reach the set temperature smoothly and/or not exceed it by much.

If you have a fairly powerful heater, increase the value so as not to "overshoot" the target temperature.

//If the difference between the target temperature and the current temperature (for Beer and sous-vide modes) is greater than the set one, then when a UART-controlled regulator is used, relay No. 4 additionally turns on the boost heating element; when the difference is smaller, the boost heating element turns off

#define ACCELERATION_HEATER_DELTA 4

For Beer and sous-vide modes. The value is given in degrees Celsius. The parameter relates exclusively to the additional heater.

If the boiler temperature is lower than the target (in this case by 4°), the additional heater turns on.

//Boiling temperature (for Beer mode)

#define BOILING_TEMP 98.9

For Beer and sous-vide modes. The value is given in degrees Celsius. The value is set manually and tells Samovar at what temperature on your equipment real boiling started in the boiler. The value [affects](#) the operating algorithm of the main and additional heaters

//Default distillation end temperature

#define DEFAULT_DIST_TEMP 99.9

For distillation mode. The boiler temperature at which the run ends, if the distillation program does not set another value. In the configurator: "Temperatures" («Температуры»), "Distillation end temperature, °C" («Температура завершения дистилляции, °C»).

Settings for using Samovar subsystems

#define SAMOVAR_USE_BLYNK *//use Blynk in the project*

If you do not use Blynk, this line must be commented out with *//*.

#define BLYNK_SAMOVAR_TOOL "samovar-tool.ru" *//use the free Blynk server samovar-tool.ru instead of the Blynk cloud*

If you use a paid Blynk server rather than the server provided by the developer, the line must be commented out.

//define USE_MQTT *//send readings to your own MQTT broker*

To save logs to the cloud this parameter is not needed: if the device token is set in the settings and SAMOVAR_USE_BLYNK is enabled, Samovar's log and messages go to the samovar-tool.ru server on their own. Register on the www.samovar-tool.ru website and enter the device token when editing your profile; the logs will then be available on the site through the menu: User - Logs. More details in the section - [Logs and charts in the cloud](#).

USE_MQTT is a separate, optional sending of a shortened readings line to an arbitrary MQTT broker (without TLS, every 4 seconds during a process and every 5 seconds when idle). This line is not in Samovar_ini.h: the parameter is enabled in the configurator (section "Network" («Сеть»): "MQTT server" («Сервер MQTT»), "MQTT port" («Порт MQTT»), "MQTT user" («Пользователь MQTT»), "MQTT password" («Пароль MQTT»), "MQTT topic" («Топик MQTT»)) or in user_config_override.h, where MQTT_SERVER, MQTT_PORT, MQTT_USER, MQTT_PASSWORD and MQTT_TOPIC must be set; without them a build with USE_MQTT stops with an error.

Selecting the power regulator

In the file **Samovar_ini.h** the selection is made with ordinary lines:

#define SAMOVAR_USE_POWER

//define KVIC_USE_9600

//define SAMOVAR_USE_RMVK

//define SAMOVAR_USE_SEM_AVR

When building with the **Arduino IDE** leave the lines as follows:

KVIC: change nothing. The **SAMOVAR_USE_POWER** line is already enabled in the original file, and the RMVK and SEM_AVR lines are commented out.

RMVK: uncomment only the line **SAMOVAR_USE_RMVK**. The firmware enables the necessary general regulator support automatically.

SEM_AVR: uncomment only the line **SAMOVAR_USE_SEM_AVR**. The firmware enables RMVK protocol support and general regulator support automatically.

Without a UART-controlled regulator: comment out **SAMOVAR_USE_POWER**, **SAMOVAR_USE_RMVK** and **SAMOVAR_USE_SEM_AVR**.

The **KVIC_USE_9600** parameter applies only to KVIC: uncomment it for a regulator with a UART speed of 9600. If the regulator uses 38400, leave the line commented out.

When building with **PlatformIO** you do not need to change these lines by hand: choose a ready-made environment, **Samovar**, **Samovar_no_power**, **Samovar_rmvk** or **Samovar_sem** (there are also **Samovar_s3** for the ESP32-S3 board, **Samovar_lua** and **Samovar_alarm_button**). The configurator chooses the environment by itself from the "Power regulator" («Регулятор мощности») group (Do not use («Не использовать») / KVIC / RMVK / SEM_AVR) and the board. Details are given in the instructions [Building and flashing with PlatformIO](#).

A detailed description of the equipment is given in the section [Voltage regulators used](#).

Parameters for the NBK

#define NBK_MULT_PAUSE_OVERFLOW 2 *// number of inertias used as the pause after flooding*

#define NBK_PUMP_LIMIT 30 *// maximum wash pump capacity for Optimization, l/h*

#define NBK_WORK_PRESSURE_RATIO 0.5f *// working pressure ceiling = a fraction of the flooding pressure*

#define NBK_PRESSURE_MARGIN 5 *// mm Hg below the pressure at the moment of flooding when learning the ceiling*

//define USE_NBK_DELTA_PRESSURE *// stillage temperature correction by the pressure in the stillage drain*

//define USE_NBK_END_BY_STEAM_RISE *// finish NBK operation when the steam T has risen by NBK_END_STEAM_RISE relative to the minimum*

#define NBK_END_STEAM_RISE 5.0f *// steam T rise threshold (°C) for USE_NBK_END_BY_STEAM_RISE*

Needed only for continuous wash column mode; in the configurator: section "NBK" («НБК»). If the NBK is not used, leave them as they are. More details in the section [NBK](#).

#define SAMOVAR_USE_POWER_START_TIME 2000 *//delay before sending the boost command to the regulator. Ensures "cold" closing of the power circuit by the contactor.*

To avoid an electric arc on the contactor's power contacts. After the "Turn on heating" («Включить нагрев») command, relay No. 1 operates and turns on the contactor, which in turn closes the power circuit to the heater. After the time set in milliseconds, the regulator supplies voltage.

Switching off happens in the reverse order. First the regulator cuts off the voltage, and then, after a delay, the contactor opens.

(See [Load connection/disconnection algorithm](#))

#define LCD_RESET_PERIOD_MS 240000UL *//period (ms) of the periodic re-initialization of the LCD display, 4 minutes*

Protection against the display "freezing" on I2C. No need to change it. In the configurator: "Equipment" («Оборудование»), "Display reset period, ms" («Период сброса дисплея, мс»).

#define USE_WATERSENSOR *//use the cooling water flow sensor. If used, then if there is no water for the set time Samovar turns the column heating off*

The flow sensor works on the Hall sensor principle. If there is coolant flow, Samovar receives periodic pulses. If there are no pulses or they arrive too rarely, Samovar turns the heating off and sends an alert.

`#define WF_CALIBRATION 98` //Value for the flow sensor $F=98 \cdot Q(L/min)$; for a different sensor, set a different value.
Flow sensor correction. If 1 liter of water is passed through the sensor, 5880 pulses are received from it. In reality the number of pulses per 1 liter differs at different water flow rates. The correction can be made by changing the value 98.

`#define WATER_FLOW_MIN_PULSES 7` //Flow is considered real if the number of pulses per cycle is not less than this threshold

Filters out single pulses of the flow sensor when the water is closed. If Samovar sees flow where there is none, increase the value; if it turns the heating off at a weak flow, decrease it. In the configurator: "Pumps" («Насосы»), "Minimum flow pulses" («Минимум импульсов потока»).

`#define USE_HEAD_LEVEL_SENSOR` //use the reflux level sensor in the head (P-N-P sensor). If used, when the sensor triggers, the power is reduced by $1 \cdot PWR_FACTOR$ (for KVIC: 1 V, for RMVK: 2 V; for SEM_AVR: by 3% of the current value)

Use a reflux level sensor whose output is a logic one when it triggers.

`#define IGNORE_HEAD_LEVEL_SENSOR_SETTING` //If the reflux sensor is used, ignore the command to disable the sensor in the WEB interface. (If you comment it out, a checkbox appears in the interface, on the Settings («Настройки») tab, with which you can enable or disable the use of the reflux sensor.)

For example.

If the line

`//define USE_HEAD_LEVEL_SENSOR` // use the reflux level sensor... is commented out, the sensor is not used, and it does not matter in what state (commented out or uncommented) the next line is.

If the line `#define USE_HEAD_LEVEL_SENSOR` is uncommented, the state of the next line is taken into account:

`#define IGNORE_HEAD_LEVEL_SENSOR_SETTING` - If it is uncommented, there is no checkbox for controlling the reflux sensor in the WEB interface. Accordingly, the sensor cannot be disabled from the WEB interface.

If `//define IGNORE_HEAD_LEVEL_SENSOR_SETTING` is commented out, the use of the reflux level sensor can be quickly [switched on and off \(item 5\)](#) from the WEB interface.

`//define WHLS_HIGH_PULL` //use an N-P-N liquid level sensor

Use a reflux level sensor whose output is a logic zero when it triggers.

`#define USE_WATER_PUMP` //use PWM control for the water pump and/or for switching on the stirrer for Beer mode

If the pump for automatically maintaining the CT is not used and the stirrer in Beer mode is not connected, the line must be commented out. In the configurator: "Pumps" («Насосы»), "Control the water pump or stirrer" («Управлять насосом воды или мешалкой»).

`//define USE_WATER_VALVE LOW` //use valve control to adjust the water pressure

When a bypass valve is used, the line must be uncommented. **The valve is connected in place of the pump.** It works on the thermostat principle. If the temperature exceeds the set one, the valve opens; when the temperature falls below the set one, it closes. More details [here](#).

If a normally closed valve is used on the bypass line, the value LOW must be replaced with HIGH.

If the CT is regulated manually or by some third-party device, both lines must be commented out.

`//define USE_ALARM_BTN` //use the emergency button.

Pressing it stops the water supply and cuts off the power. You can use it by installing leak sensors (an unlimited number of devices with "dry contacts" connected in parallel is allowed), or an ordinary button (it can also be connected in parallel with the sensors)

More details in the section [Emergency button. Additional emergency sensors](#)

`#define USE_BODY_TEMP_AUTOSET` //use automatic hearts T correction for the first hearts or pre-flooding take-off program after the heads, and also for the pre-flooding program if it comes before the second-to-last hearts or pre-flooding take-off program

In the standard Samovar 7.00 build this function is enabled. It cannot be commented out: the firmware enables `USE_BODY_TEMP_AUTOSET` forcibly (Samovar.h). In the configurator it is in the "Rectification" («Ректификация») section ("Hearts temperature auto-correction" («Автокоррекция температуры тела»)), and the rise limit `BODY_TEMP_AUTOSET_MAX_RISE` is there too (see below).

The take-off will not be paused provided that:

For the line **Hearts («Тело»)**. If a Hearts line comes right after a Heads line and is followed by two Hearts or Pre-flooding lines in any combination.

For the line **Pre-flooding («Предзахлеб»)**. If a Pre-flooding line is anywhere in the take-off program and is followed by two Hearts or Pre-flooding lines in any combination

`//Choose which pressure sensor is used and leave only one line. The other lines must be commented out.`

`//define USE_BMP180` //use the BMP180/BMP085 pressure sensor

`//define USE_BMP280` //use the BMP280 pressure sensor

`##define USE_BMP280_ALT` *//use the BMP280 pressure sensor with an alternative address*

`##define USE_BME280` *//use the BME280 pressure sensor*

`#define USE_BME680` *//use the BME680 pressure sensor*

Choose which atmospheric pressure sensor is used and leave only one line. The other lines must be commented out. If an atmospheric pressure sensor is not used, you can comment out all the lines. But then Samovar will not correct the boiling temperature depending on the atmospheric pressure.

`#define USE_BTN` *//use the button*

Use the universal button on Samovar. It is connected to the D39 output of the ESP32. A handy button, but if it is not used, it is better to disable it.

`#define USE_UPDATE_OTA` *//use ArduinoOTA for updating over WiFi. Handy for debugging; during operation it puts extra load on the ESP32*

If a serial wired connection is used to update the ESP32 sketch, this line can be commented out. Updating over WiFi will then be unavailable. When building with ESP32 core version 3.x, the firmware disables this parameter by itself. In the configurator: "Network" («Сеть»), "Allow update over Wi-Fi" («Разрешить обновление по Wi-Fi»).

`##define USE_EXPANDER 0x20` *//use the PCF8575 port expander*

`##define USE_ANALOG_EXPANDER 0x48` *//use the PCF8591 analog port expander*

`##define USE_LUA` *//use Lua for writing scripts to control Samovar and the port expander*

Enabling the use of port expanders and the Lua scripting language. More details in the section [Extending the capabilities](#)

//Selecting the type of pressure sensor used in the column. Only one type can be used.

`##define USE_PRESSURE_XGZ 32` *//use the XGZP6897D pressure sensor (or similar). The value is the K parameter from the sensor's table*

`##define USE_PRESSURE_1WIRE {0x22, 0x22, 0x22, 0x22, 0x22, 0x22, 0x22, 0x55}` *//use a pressure sensor over 1Wire*

`##define USE_PRESSURE_MPX` *//use the MPX5010D pressure sensor*

Pressure sensor in the column (not to be confused with the atmospheric pressure sensor above). It is needed for the NBK and for monitoring the pressure in the column. Uncomment only one line; if two or three are uncommented, the MPX5010D is used. If there is no sensor, leave all three lines commented out. In the configurator: the "Column pressure sensor" («Датчик давления в колонне») group.

`##define USE_STEPPER_ACCELERATION` *//use smooth start and braking of the stepper motor*

Turn on smooth acceleration of the take-off pump; it will also stop smoothly.

`##define STEPPER_REVERSE` *//reverse the stepper motor rotation direction*

If you need the take-off pump motor (Nema17) to rotate in the other direction, the line must be uncommented.

`##define I2CStepperStepMl 16000` *//number of steps per ml for the I2CStepper*

Only when an external I2CStepper module is used. Sets the initial calibration of the take-off pump (steps per milliliter). In the configurator: "Stepper motor" («Шаговый двигатель»), "Steps per ml for I2CStepper" («Шагов на мл для I2CStepper»).

`##define COLUMN_WETTING` *//if a reflux level sensor is installed, at the start of rectification bring the column to pre-flooding to wet the packing*

`##define WETTING_POWER 220` *//regulator voltage/power (for SEM_AVR) when wetting the packing*

Wetting the packing before rectification. Works only with the reflux level sensor (USE_HEAD_LEVEL_SENSOR): after boost, Samovar holds on the regulator the voltage of the first program line (if it is not set, 220 V; for SEM_AVR, WETTING_POWER), reduces it in steps down to 80% if necessary and waits for the reflux level sensor to trigger, for no more than 20 minutes. In the messages: "Column packing wetting started" («Начало смачивания насадки колонны») and "Column packing wetted successfully!" («Насадка колонны успешно смочена!»). In the configurator: "Rectification" («Ректификация»), "Wetting the packing before rectification" («Смачивание насадки перед ректификацией») and "Packing wetting power" («Мощность смачивания насадки»).

`##define NOT_USE_INTERFACE_UPDATE` *//do not update the interface on startup*

Samovar compares the locally installed version of the interface with the version available remotely and, if they do not match, updates it.

This approach lets you keep files with your own developments in Samovar. A local update overwrites everything, while an update over the internet updates only the interface, and scripts and settings are kept.

For a new version you do not need to upload the interface; it updates automatically.

But, as it turned out, something has changed and this does not work for everyone.

If this parameter is uncommented, the interface will not be updated over the internet.

`##define KVIC_DEBUG` *//debug messages of the exchange with the KVIC regulator to the serial monitor*

Needed only when troubleshooting communication with the regulator. In the configurator: "Regulator" («Регулятор»), "KVIC debug messages" («Отладочные сообщения KVIC»).

Rectification algorithm parameters

`#define PAUSE_RESUME_HYSTERESIS_DELTA 0.07f` *//hysteresis (°C) between the pause entry threshold by steam/column section T and the exit threshold*

`#define PROGRAM_ROW_STOP_PAUSE_LIMIT 3` *//number of stop pauses during the current program line after which the line's base speed is reduced automatically*

```
#define PROGRAM_ROW_STOP_PAUSE_SPEED_CUT_PCT 10 //by how many % to reduce the line's base speed when
PROGRAM_ROW_STOP_PAUSE_LIMIT is triggered
#define PROGRAM_DONE_AUTO_POWEROFF_MIN 30 //time (min) to hold the status "Program execution finished"
(«Выполнение программы завершено») (the pump is stopped, heating continues), after which it switches off automatically. 0 =
switch off immediately
#define BODY_TEMP_AUTOSET_MAX_RISE 0.3f //limit (°C) of the automatic rise of the hearts T within one program line
The default values are chosen for a typical column; do not change them unless necessary. In the configurator: section "Rectification" («Ректификация»).
```

BK mode parameters

```
#define BK_STEAM_SETPOINT_MIN 30 //lower limit of the BK dephlegmator steam T setpoint (5th field of the program
line), °C; 0 - water by hand
#define BK_STEAM_SETPOINT_MAX 100 //upper limit of the BK dephlegmator steam T setpoint, °C
#define BK_WATER_ADJUST_PERIOD_MS 60000 //BK water regulator step period, ms
#define BK_WATER_DEADBAND 0.2f //dead band (°C) around the BK steam setpoint
#define BK_WATER_PWM_STEP 30 //BK water PWM regulator step (out of 1023)
Setpoint limits and the step of automatic water regulation in wash column mode. In the configurator: section "BK" («БК»).
```

After editing the file, you must upload the sketch to the ESP32 again. To do this, click the arrow on the top panel of the Arduino IDE and wait for the upload to finish.



Your Samovar is ready to work.

Note.

From now on, when you install a new release, you do not have to edit `Samovar_ini.h` every time.

When updating through the Arduino IDE, compare the new and old `Samovar_ini.h` and transfer only your own values. Do not replace the new file entirely with the old one: a new release may have parameters without which the configuration would be incomplete. The configurator stores user differences separately in `user_config_override.h`.

It is better to edit the new file itself rather than add lines from the new code to the old one, to avoid accidental errors ([see Updating the sketch](#)).

For comparing and editing you can use special programs that can compare two different files, visually showing the differences between them.

Warning.

It is recommended to download the archive with the new Samovar release from [this link](#). It will be a stable and tested version.

* For developers. A description of the sketch sections and a block diagram are on github.com

1. [User interaction \(web and LCD\)](#)
2. [Process program execution](#)
3. [System state and mode management](#)
4. [Sensor data acquisition](#)
5. [Hardware control \(actuators\)](#)
6. [Safety monitoring and alarms](#)
7. [Configuration persistence](#)
8. [Network and external communication](#)

Other files and lines with explanations:

file `Samovar_pin.h`

`Samovar 7.00`: the `TARGET_WATER_TEMP` constant is no longer in this file. The water temperature maintained by the pump PWM regulator is taken in all modes (including the NBK) from the "Water sensor setpoint" («Уставка датчика воды») field on the [Settings - Temperature \(«Настройки - Температура»\)](#) tab; if the field is empty, the firmware uses 48°. When the process has finished and the outlet water temperature becomes lower than the setpoint by

DELTA_T_CLOSE_VALVE (20° by default, see above), Samovar stops the pump, gives the command to close the water valve and sends an alert that the water must be closed.

file pumppwm.h

define PUMP_PWM_FREQ 15

Minimum PWM frequency of the cooling pump, in hertz

Updating the sketch

Samovar 7.00: updating through the Arduino IDE is supported. As an equal option, you can run the configurator of the new release and press "Flash" («Прошить»); LittleFS is uploaded with a separate "Upload LittleFS" («Загрузить LittleFS») button and replaces the web interface files (user files in LittleFS may be lost in the process). The configurator stores your values in the user_config_override.h file next to the sketch - copy it to the directory of the new release, and the settings will be kept. You can also flash over Wi-Fi: the "Update" («Обновить») button finds Samovar on the network if "Allow Wi-Fi update" («Разрешить обновление по Wi-Fi») is enabled in the "Network" («Сеть») section (USE_UPDATE_OTA).

As new versions come out, updating will become necessary. A new release may come with new or changed libraries and tools. Therefore, to avoid errors, it is recommended each time to use exactly the libraries and tools that are in the archive of the specific release.

Attention! Important!

Starting with version 6.0, provided that Samovar is connected to the internet, the interface will be updated automatically for the specific sketch version, i.e. if you need to roll back to an older sketch version, the interface will also roll back to the corresponding version. If there is no internet connection, the interface must be installed by yourself, locally.

Updating the sketch

Using the section [Initial sketch upload](#) we prepared a certain structure of files and directories in ...Documents\Arduino. We will continue based on the previous organization.

Prepare the ... Documents\Arduino\ directory in the following order:

- Move the libraries and tools directories by dragging (or cut and paste) into the Samovar directory.
- Move the Samovar directory into an empty Samovar-7.00 directory.
- Move the Samovar-7.00 directory into the OLD directory.
- Move the Samovar-7.00.zip archive with the original sketch into the OLD directory.

This way only the OLD directory remains in the ... Documents\Arduino\ directory.

And inside OLD there will be: the Samovar-7.00 archive, and the Samovar-7.00 directory with the nested Samovar directory.

This storage system rules out errors during updates, lets you use the previous settings for the new release and, if necessary, quickly roll back to the previous or an even earlier version.

Next we proceed step by step, as in the first installation: "unpack" the new release in ... Documents\Arduino\ until the previous [directory structure](#) is obtained. Accordingly, the archive and the directory obtained from the archive will have the new version number in their names.

But now we do not upload the sketch to the ESP right away; first we make the necessary settings in the Samovar_ini.h file. This can be done in two ways:

1. The right way.

For convenience you will need special programs that can compare two different files, visually showing the differences between them.

The recommended program is [WinMerge](#) — (A compare and merge tool for Windows that can compare both files and folders, showing the differences in a visual text form . The program is easy to use and has a Russian interface.)

Or a more advanced, but also harder to learn, program **Notepad++** with the installed plugin **Compare**.

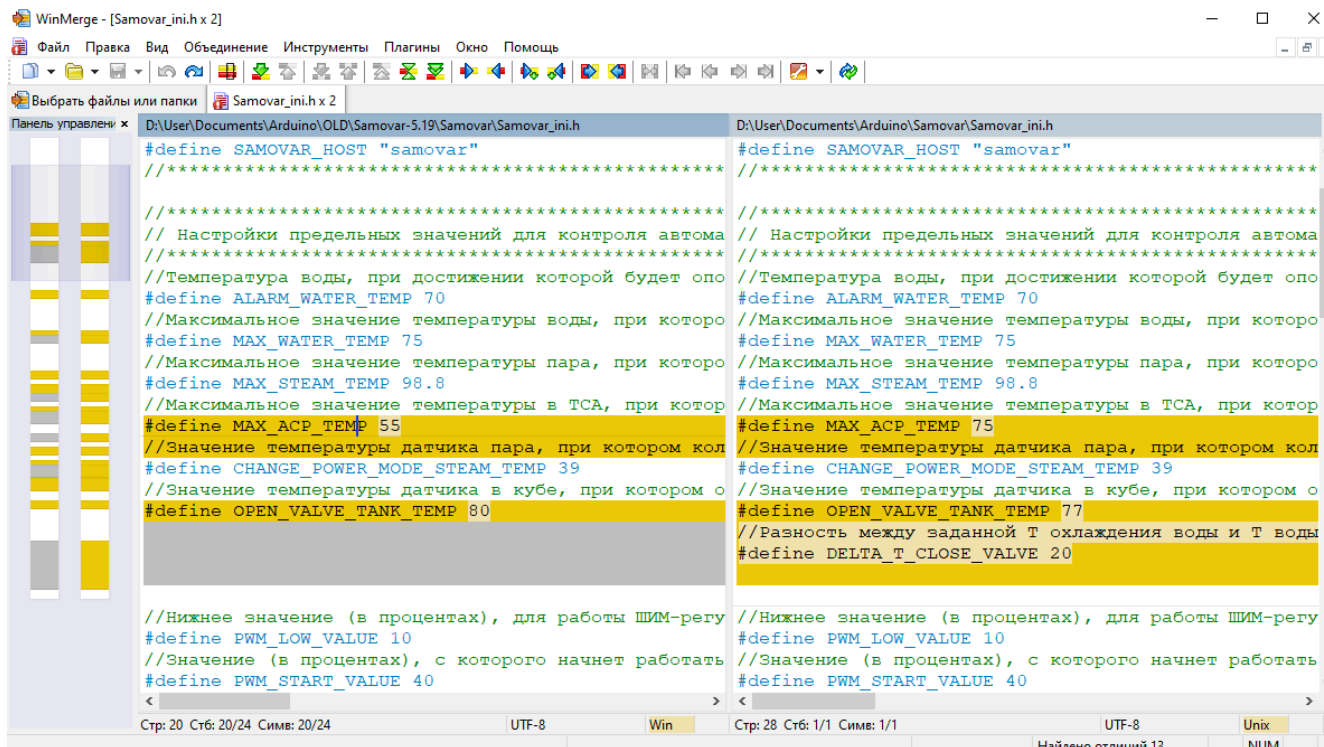
An example using the WinMerge program.

To compare, you need to select two different Samovar_ini.h files

First select the file from the OLD directory, where we have just moved the previous Samovar version. It is located at: ... \Documents\Arduino\OLD\Samovar-7.00\Samovar\

Take the second file from the directory with the new Samovar release. It is located in ... \Documents\Arduino\Samovar\

The program compares both files and highlights the lines with differences.



The differences will be highlighted in a lighter tone. You can edit the values on the right manually, or you can place the cursor on the left and give the command **"Copy to Right"** («Копировать направо»).

This way, line by line, the settings file is adjusted for your equipment. When all edits are done, click the "Save" icon

The main thing is not to mix up that the old settings file is on the left, and the file Samovar will work with is on the right.

2. The quick way

When updating through the Arduino IDE, compare Samovar_ini.h of the new and previous releases and carry over only your own values. Do not paste the old Samovar_ini.h over the new one: even a small release may contain new required parameters.

If the new release introduces new Samovar functions, you must edit the Samovar_ini.h file, otherwise the new functions will be unavailable and Samovar will not work correctly.

If the settings were made in the configurator, you do not need to edit Samovar_ini.h: your values are stored in user_config_override.h, which overrides Samovar_ini.h. It is enough to carry this file over to the new release.

The setup is finished. You can upload the sketch to the ESP. Since this is an update of a previously installed sketch, you will not have to set up the Wi-Fi connection again and you do not need to load the default settings.

When updating, you can already skip connecting the ESP to the USB port and update the firmware ["over the air"](#)

"Rollback" to an old version.

Something in the new release did not please you, or it does not work correctly. You want to go back to the previous versions. With this system of storing old versions, it is very simple.

Prepare the ... \Documents\Arduino directory as for an update:

- Move the libraries and tools directories by dragging (or cut and paste) into the Samovar directory.
- Move the Samovar directory into an empty Samovar-7.00 directory.
- Move the Samovar-7.00 directory into the OLD directory
- Move the Samovar-7.00.zip archive with the original sketch into the OLD directory

Next:

- Go to the OLD directory, choose the previous version and go inside to the Samovar directory: ... \Documents\Arduino\OLD\Samovar-6.16\
- Copy (copying rather than dragging is recommended) the whole Samovar directory into ... \Documents\Arduino\

From the Samovar directory, move by dragging (or cut and paste) one level up, into ... \Documents\Arduino\, the two directories libraries and tools.

That is all. You can upload the sketch to the ESP.

Later, when updating to a new version, you will just need to clear the ... \Documents\Arduino directory by deleting Samovar, libraries and tools. There is no need to move them to OLD, since they are already there and we used a copy.

When updating the sketch you can already use **Arduino IDE versions 2.X.x**. But these versions do not have the item **"ESP32 LittleFS Data Upload"**, and without the plugin you will not be able to update the interface locally. Accordingly, if Samovar has no access to the internet, the web interface will not be updated.

Addition. In Arduino IDE 2.x.x versions it became possible to install the interface locally using a plugin. The instructions are [here](#). The plugin (arduino-littlefs-upload-1.1.7.vsix) is in the Samovar archive in the tools\ESP32LittleFS_Arduino IDE 2.X directory, see [Initial sketch upload](#).

You can install several Arduino IDE 2.x.x and Arduino IDE 1.x.x versions on your computer at the same time, and they will not cause any problems for each other. So you can keep using Arduino IDE 2.x as your main IDE and use Arduino IDE 1.x only when you need to upload the LittleFS file system (including the interface) to your ESP32 locally.

When using different versions of the Arduino IDE you need to install them in different directories. For this it is enough to add the version number to the directory name.

For example.

C:\Program Files(x86)\Arduino1_1_18;

C:\Program Files\Arduino IDE_2_3_2;

C:\Program Files\Arduino IDE_2_3_1

This makes it possible to use in each version of the Arduino IDE different versions of the ESP32 SDK from Espressif Systems. But keep in mind that each of the Arduino IDE versions you use, will have to be configured separately for ESP32.

Attention!

If you use the ESP32 add-on “**ESP32 by Espressif Systems**” version 3.0.0 or higher, the firmware disables Wi-Fi (OTA) updating by itself: the USE_UPDATE_OTA parameter has no effect in this core, upload only over USB. In addition, you need to change the board type used and select the right partitions. The Samovar partition table is absent in core 3.x; it is added manually (the description is at the beginning of the Samovar.ino file):

. copy the partitions\samovar.csv file from the sketch directory to ...\\Arduino15\\packages\\esp32\\hardware\\esp32\\3.x.x\\tools\\partitions\\ (on Windows the Arduino15 directory is in C:\\Users\\<name>\\AppData\\Local\\);

. add three lines to the ...\\esp32\\3.x.x\\boards.txt file:

esp32.menu.PartitionScheme.samovar=Samovar

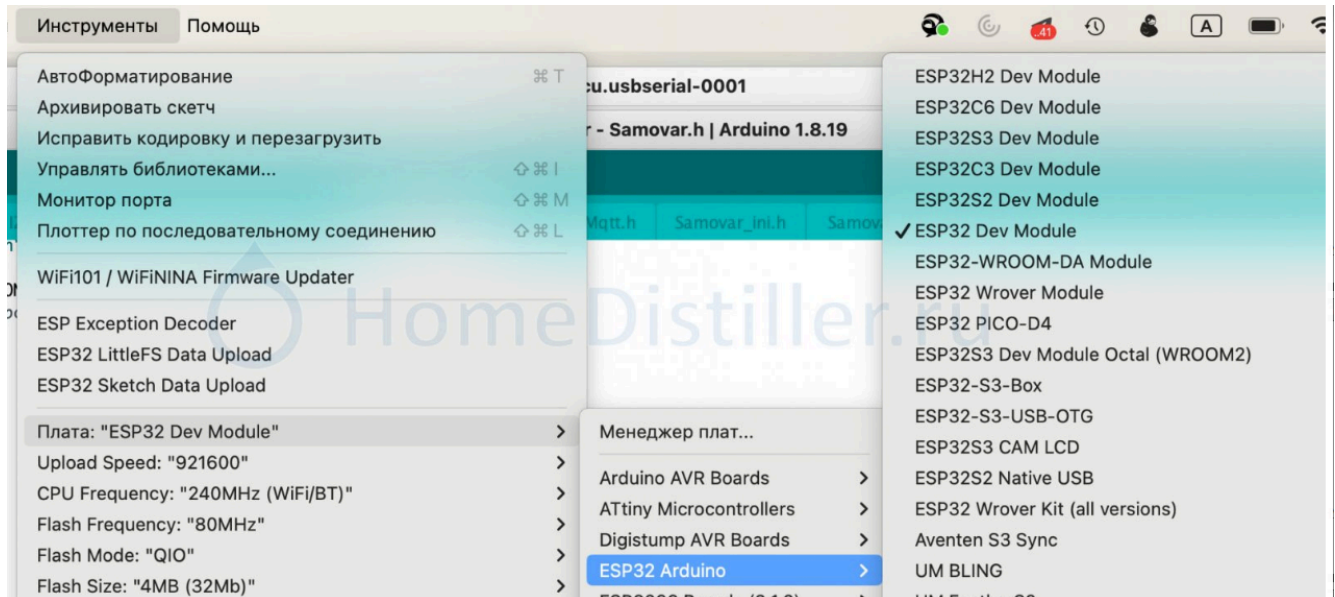
esp32.menu.PartitionScheme.samovar.build.partitions=samovar

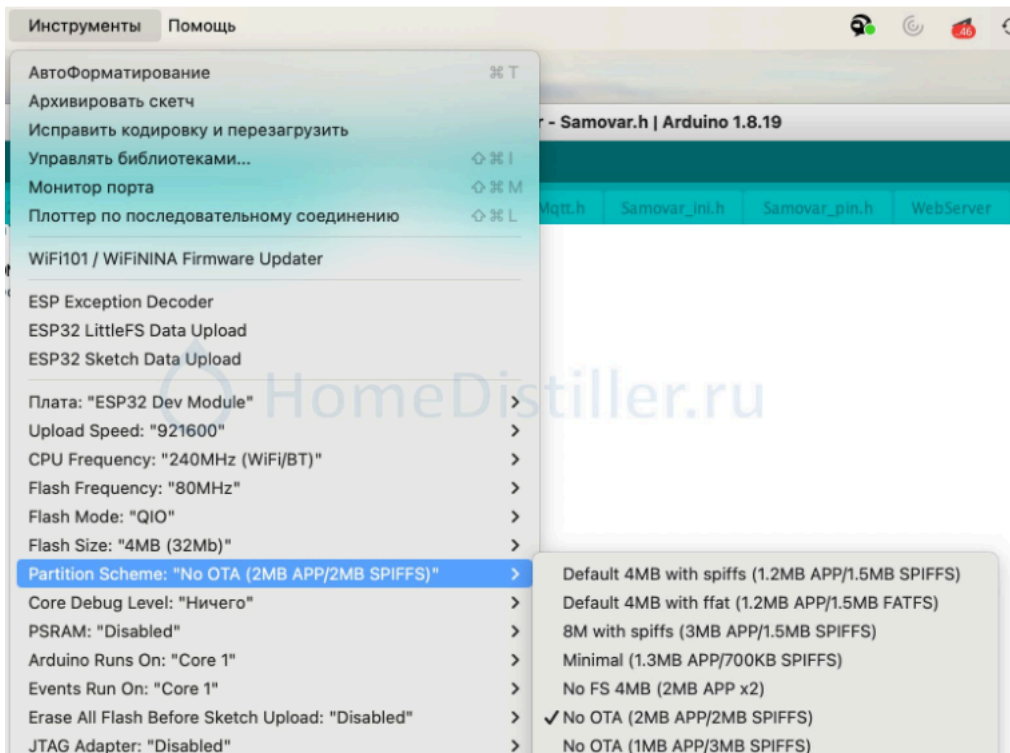
esp32.menu.PartitionScheme.samovar.upload.maximum_size=1507328

. restart the Arduino IDE and choose “Tools” → “Partition Scheme” → “Samovar”.

The PlatformIO configurator uses core 2.0.17 and the same partition table (partitions/samovar.csv); nothing needs to be added.

[Post on the forum.](#)





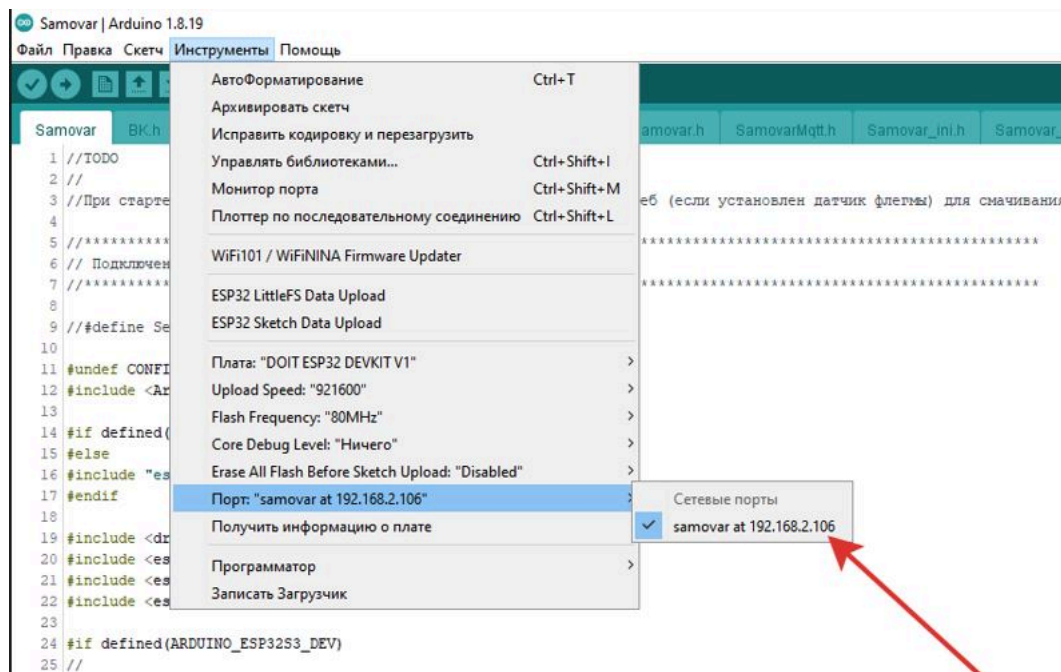
Questions and answers. FAQ

I could not find anywhere how to update the firmware "over the air". Please advise...

The first time, you in any case need to flash by wire and connect Samovar to your Wi-Fi network. The `USE_UPDATE_OTA` parameter must be enabled in `Samovar_ini.h` (enabled by default; in the configurator it is "Allow Wi-Fi update" («Разрешить обновление по Wi-Fi»)), the ESP32 core is 2.0.17.

Reboot Samovar and restart the Arduino IDE. After that, the IP address of Samovar will appear in the network ports of the Arduino IDE (a network port named `samovar`).

Instead of the USB (COM) port, select the IP address, and the update will go over the air.



Later, if you connect Samovar to the computer with a USB cable, two options will be available in the ports: updating "over the air" or over the COM port. Whichever channel you choose, the sketch will be sent through it.

The configurator works the same way: press "Update" («Обновить»), and both the COM port and Samovar found on the network appear in the port list; select the network one and press "Flash" («Прошить»). The "Upload LittleFS" («Загрузить LittleFS») button also works over the network.

Remote Control

The Samovar web interface is designed to work in a trusted local network. The firmware has no built-in username and password check for the control pages.

Do not publish the local Samovar web interface on the internet. Anyone who gets such an address can open the interface and send commands to the device.

To access it from another network, use a secure connection to your home network, for example a VPN on the router or on a separate server. A VPN is an encrypted connection after which a phone or computer works as if it were at home. The setup depends on the router model and is not a function of the Samovar firmware.

To monitor, receive notifications and control over the internet without opening the web interface, use the samovar-tool.ru server: Samovar connects to it by itself, and through it work the [Samovar mobile apps for Android and iOS and the control page on the website](#), push notifications from Samovar, and [logs and charts in the cloud](#). A single app can work by token without registering on the website. For several Samovars, the shared list and names are set in the website menu "User → Devices". The token of each device is entered on the "[Other](#)" tab of its settings and must not be published.

Before a remote start, make sure the equipment is assembled safely and that someone nearby can immediately cut the power and the product supply. The remote interface does not replace hardware protections and supervision of the installation.

Mobile apps and website

Samovar sends readings, program state, the process log and messages to samovar-tool.ru. You can watch the process and control it over the internet in the mobile app for Android, the same app for iOS, and on the "Samovar Control" page of the website. The apps and the website page have identical screens, so they are shown below in screenshots from the website.

Nothing needs to be opened to the internet for this: Samovar establishes the connection itself, see [Remote control](#).

Device token

The token is a key of 32 Latin letters, digits, "-" or "." characters, by which the server tells your Samovar apart. Each device has its own token. It is used by the firmware, the apps, the control page and the cloud logs.

First device. In the mobile app on first launch choose "Create account" and enter an email and password. A Blynk account will be created, the Samovar project will be copied into it, and the token will appear on the "Settings" screen. If you already have a Blynk account, press "Sign in and get token" or enter a known token manually. Registration on the website is not required for a single device.

Where to enter it. In the Samovar web interface open Settings («Настройки»), the tab [Other \(«Прочие»\)](#), paste the token into the "Blynk token" («Токен Blynk») field and press "Save" («Сохранить»). Then reboot Samovar by switching the power off: the connection to the server is established at start. The token can also be entered during the first Wi-Fi connection in the "blynk token" field on the [Samovar access point page](#).

Do not tell anyone the token. Anyone who knows it can see the readings and control your Samovar.

Website

Register on the website www.samovar-tool.ru and sign in. Open the menu "User → Devices". On registration the website creates a device named "Device" for you: the cheese, beer and distillation journals are attached to it. It is a full-fledged device with its own token. To connect a Samovar to it, press "Show token" on its card, enter your current website password and write the shown token into the ESP32 settings.

If you already have a Samovar project in Blynk, press "Import device" and enter the name, the Blynk token and your current website password. The "Create device" button creates one more Samovar project and shows its new token once. Write the token into the settings of the corresponding ESP32. The token of any device in the list can be viewed later with the "Show token" button.

The number of devices is limited. By default one device per user is allowed, and the "Device" created at registration counts too. How many are used and how many are allowed is shown at the top of the page: "Devices used: 1 of 1". When the limit is reached, the "Import device" and "Create device" buttons are not shown, and migrating a Blynk account is rejected if it has more Samovars than free places. To connect more Samovars, write to the website administrator: they will raise the limit. Devices already added are not deleted if the limit is lowered.

The name is saved on the website and in Blynk. On the same page you can rename a device and choose the Samovar for the page ["Control"](#) and the "Logs" page, see [Logs and charts in the cloud](#).

Several devices in the app

In the app settings enter your website email and password and press "Connect site devices". The website and Blynk email and password must match. If they differ, first migrate the account following the instructions below. The app will load the list of devices. If there is one device, it is selected automatically. If there are several, a list of names appears in the settings.

Android, iOS and the website store the selection independently. For example, Android can control the Samovar at home, iOS the Samovar in the garage, and the website a third device. Changing the selection in one place does not switch the others. After changes on the website, press "Refresh device list" in the app. Push notifications arrive from all devices in the common list. When you tap a notification, the app automatically switches to the Samovar that sent it. Receiving a notification does not change the current selection by itself. In single-device mode push works after Blynk sign-in or registration; a token alone, without an email and password, is not enough to register the phone for notifications.

If the Blynk email is different

On the "User → Devices" page press "Migrate Blynk account". Enter the old Blynk email, the old Blynk password and your current website password. The website will check the entered data and move suitable projects to the account with the website email and password.

Attention. Only projects named Samovar (case-insensitive) with one device and the tabs "Main" («Основное»), "Control" («Управление»), "Charts" («Графики») are migrated. A project with a different name, a different set of tabs or several devices is not migrated. After a successful migration the old Blynk account is deleted together with all its other projects. If no Samovar project is found, the old account is deleted anyway. There is no preliminary list and no undo after a successful migration.

The tokens of migrated Samovars are kept, so the ESP32 settings do not need to be changed.

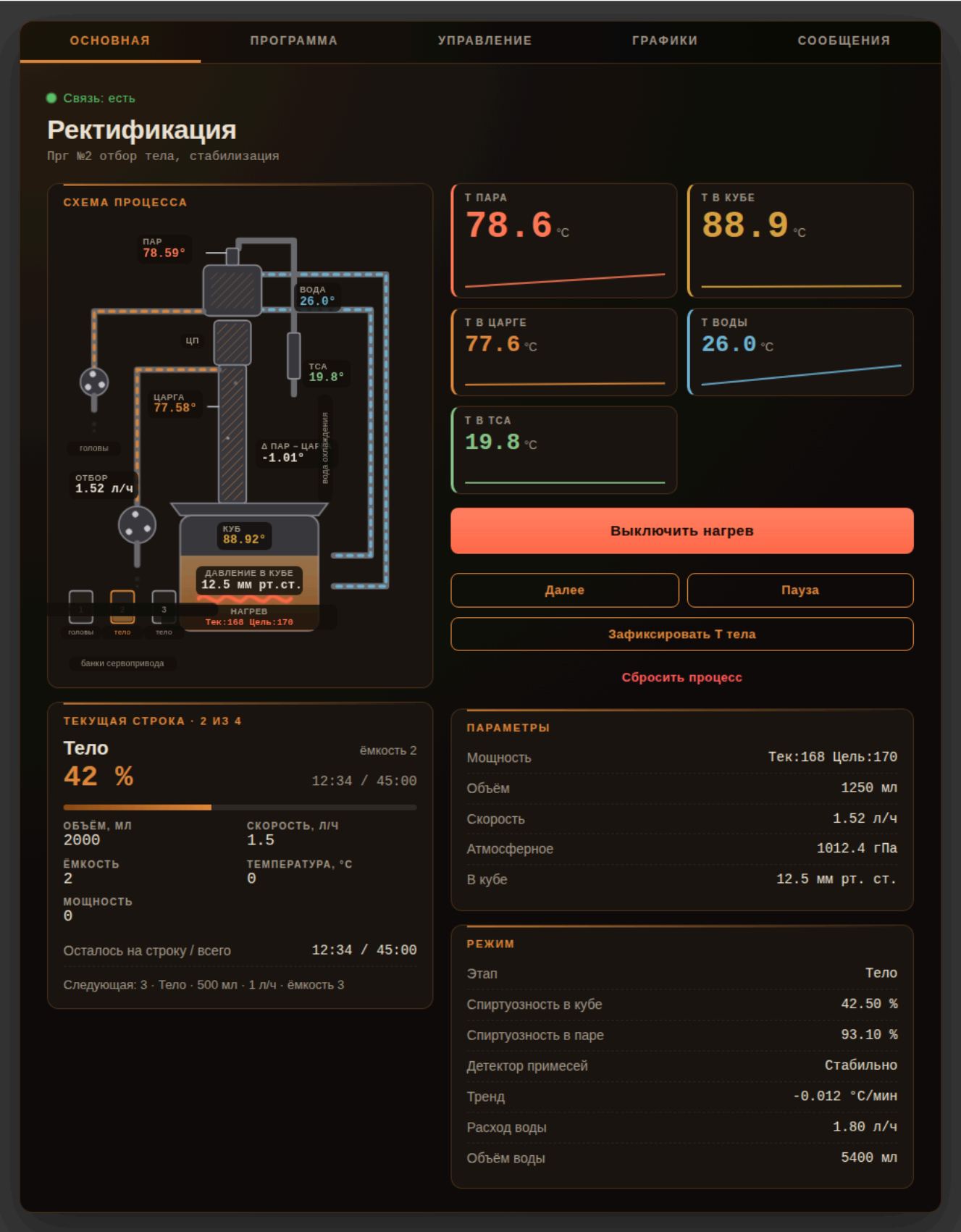
If the website reports an error during migration, do not rerun it right away. First sign in to Blynk with the website email and password and open the "Devices" page again: the Blynk migration may already have finished while the list update on the website has not.

After migration use the common website and Blynk email and password. To change the password in both places at once, press "Change shared password" on the devices page. After changing the password, sign in to the website and the apps again.

The control page refreshes data every 3 seconds while the browser tab is open. Alarms are shown in a pop-up window until it is closed. Push notifications to the phone arrive only through the mobile app.

Screens

"Main". Connection state, mode and status line, a process diagram with current temperatures (the same as in the web interface: by mode, beer brewing order and the second pump above the cooling water), sensor cards with mini-charts, the buttons "Turn heat on/Turn heat off", "Next", "Pause", "Lock body T" and "Reset process", the current program line, parameters and mode.



The "Main" screen on the website; in the apps it is the same

"Program". The program lines of the current mode with a decoding of the fields, the running line highlighted. The program is view-only here: it can be changed in the Samovar web interface.



The "Program" screen

"Control". Regulator voltage, take-off rate, locking the hearts temperature, the I2CStepper stirrer and pump (parameters, start and stop, relays R1-R4, pump calibration), running a Lua line and system information: the Samovar IP, firmware version, mode, time.

● Связь: есть

Ректификация

Прг №2 отбор тела, стабилизация

МОЩНОСТЬ (НАПРЯЖЕНИЕ)

Сейчас: Тек:168 Цель:170

Напряжение, В

Установить

СКОРОСТЬ ОТБОРА

Сейчас: 1.52 л/ч

Скорость, л/ч

Установить

Зафиксировать T тела

МЕШАЛКА I2CSTEPPER

Статус: работает · Скорость: 120 шаг/с

RPM

60

Работа, сек

30

Пауза, сек

10

☐ Реверс после паузы

☒ Плавный старт

☐ Датчик активен уровнем 1

☐ Стоп по внешнему датчику

Применить

Работает — остановить

Сохранить

R1: выкл

R2: выкл

R3: выкл

R4: выкл

НАСОС / НАПОЛНЕНИЕ I2CSTEPPER

Статус: работает, калибровка · Осталось: 320 мл · Скорость: 400 шаг/с

☐ Прокачка ☒ Наполнение

Прокачка, мл/час

1000

Пауза насоса, сек

5

Наполнение, мл

500

Скорость наполнения, мл/час

2000

Шагов на 1 мл

150

☒ Плавный старт

Применить

Работает — остановить

Сохранить

Калибровка старт

100 мл стоп

R1: вкл

R2: выкл

R3: вкл

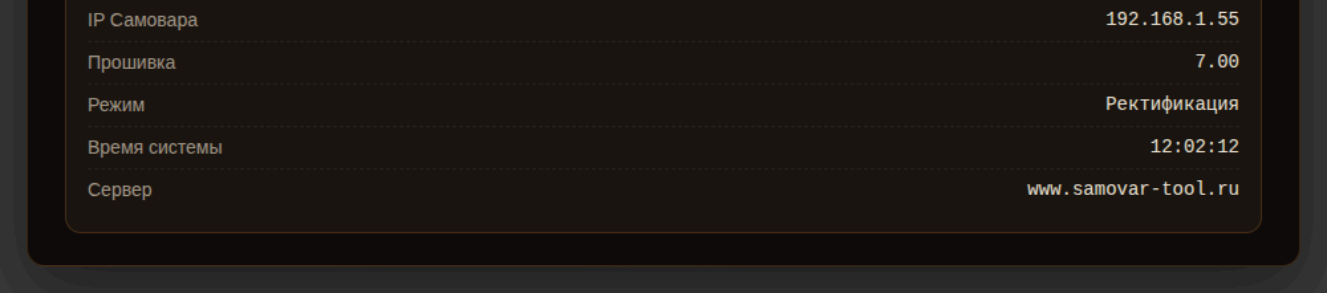
R4: выкл

ВЫПОЛНИТЬ LUA-СТРОКУ

Команда Lua

Выполнить

СИСТЕМНАЯ ИНФОРМАЦИЯ



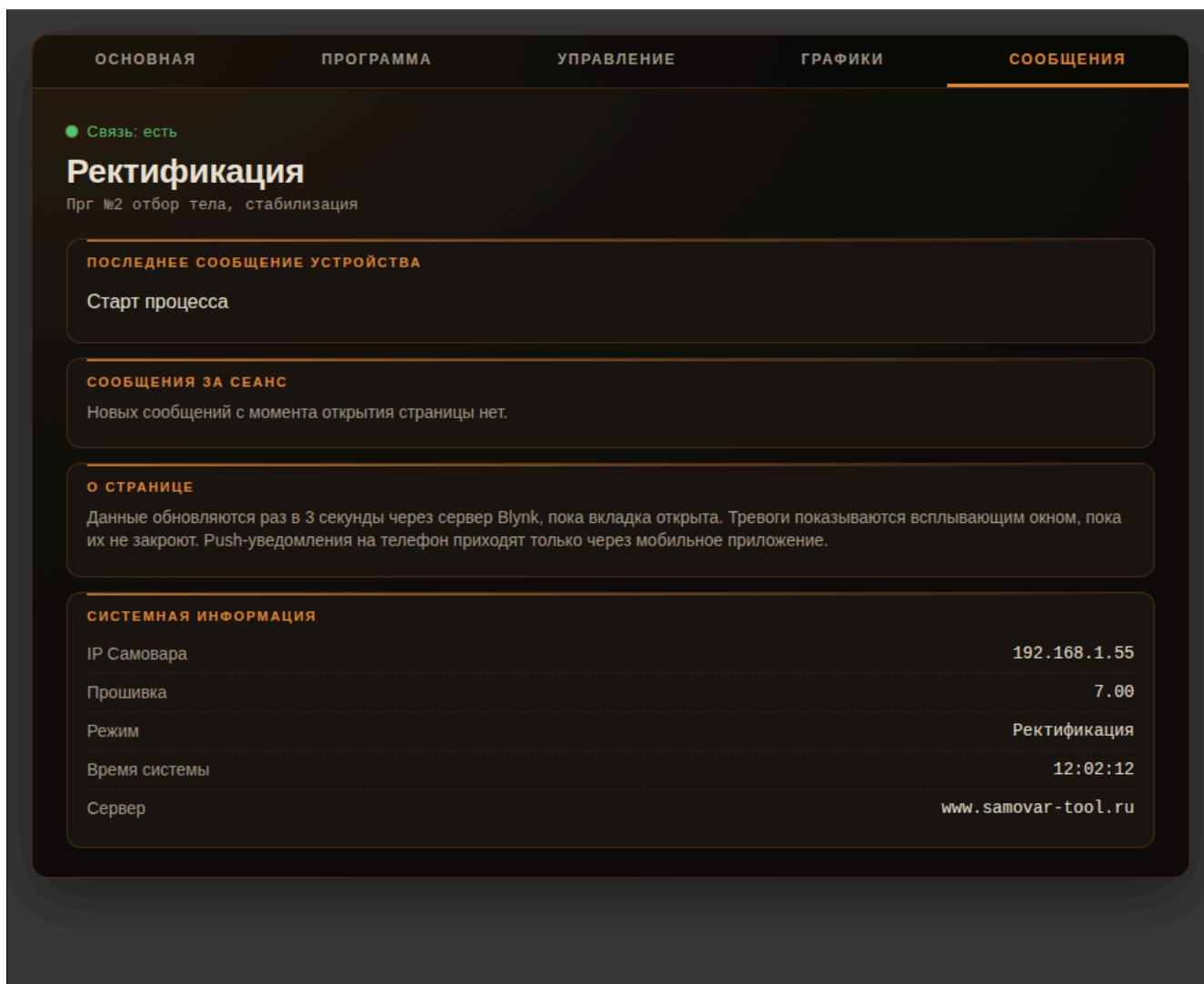
The "Control" screen

"Charts". The history of the current session from the server: the groups "Temperatures", "Pressure", "Takeoff and water", "Alcohol", "Power", an interval from 1 minute to the whole session, and individual lines are switched on and off with the buttons under the chart.



The "Charts" screen

"Messages" on the website, "Settings" in the app. On the website the fifth tab shows the last message of the selected device and the messages received during the session. In the app, the "Settings" screen takes its place: the account, a single token or device selection, and notifications; the app shows messages as push notifications.



The "Messages" tab on the website

How it works

Samovar sends the readings to the server by itself: temperatures, pressure, voltage and take-off rate every 5 seconds, the rest (status, program line number, the program, heating and pause state) on change. Commands from the app and the website are applied immediately, without a confirmation prompt. Leave a pause of 1-2 seconds between commands.

Samovar messages are shown on the "Messages" tab and arrive as push notifications in the app; alarms come with the prefix "Alarm!" («Тревога!»), warnings with "Warning!" («Предупреждение!»). Through the same server the firmware transmits the process log and the session start; they are stored on the server and available on the website, see [Logs and charts in the cloud](#).

Attention. Before a remote start, make sure the equipment is assembled safely and a person is nearby who can immediately cut the power and the product supply. The remote interface does not replace hardware protections and supervision of the installation.

Voltage regulators in use

WARNING!

All regulators connected to Samovar must be physically isolated in terms of voltage. These can be relays, optocouplers, data bus isolators such as ADUM, etc.

Never connect the regulators directly, otherwise the mains voltage will reach you and Samovar. The best that can happen in that case is that Samovar burns out. Other consequences will be much worse.

Samovar 7.00: the regulator type is selected on the "Regulator" («Регулятор») tab of the configurator (the "Power regulator" («Регулятор мощности») field): KVIC, RMV-K, SEM_AVR or "Do not use" («Не использовать»). The configurator is started with the script flash_windows.bat or flash_macos_linux.sh and selects the required build environment by itself; the PlatformIO environment names below are given for manual builds.

Samovar works with the following types and models of voltage regulators:

Voltage regulators controlled over the UART protocol

Attention! The algorithm for connecting/disconnecting the load when UART control is used:

Connecting.

After the "Turn heating on" («Включить нагрев») command, Samovar switches on relay No. 1, which in turn switches on the contactor. Then it waits for the time specified in the line

#define [SAMOVAR_USE_POWER_START_TIME](#) of the file **Samovar_ini.h** (2 seconds by default; for SEM_AVR the firmware waits another 3 seconds for the regulator to start) and commands the regulator to apply voltage. This ensures that the contactor closes the load power circuit while there is no voltage in it at that moment, which prevents sparking and extends the contactor life. Simultaneously with the voltage being applied by the regulator, relay No. 4 switches on; it is used to connect the [boost heating element](#) (in the modes where the boost heating element is used).

Disconnecting.

After the "Turn heating off" («Отключить нагрев») command, Samovar commands the regulator to cut the voltage, waits for it to be done and after 0.2 seconds switches off relay No. 4 and relay No. 1, which in turn switches off the contactor. If the regulator did not respond to the command, the relays are switched off anyway on timeout, and the alarm "Regulator command timeout" («Таймаут команды регулятора») appears in the messages.

1. Regulator KVIC

In the configurator select "Power regulator" — **KVIC** (this is the default value). In **PlatformIO** it corresponds to the **Samovar** environment. The whole block is explained in the section [Sketch setup](#).

Fully compatible with Samovar. The adjustment step on the encoder is 0.5 volt. Stabilization is 0.1 volt. It can be shut down in an emergency by an independent sensor.

2. Regulator RMV-K (only versions No. 5 and 6 work with Samovar)

All RMV-K models have power isolation on the control contacts by default, and no modification is needed to work with Samovar.

In the configurator select "Power regulator" — **RMV-K**. In **PlatformIO** it corresponds to the **Samovar_rmvk** environment (the **SAMOVAR_USE_RMVK** directive). The firmware enables the general regulator support automatically.

The adjustment step, both on the regulator and from Samovar, is 1 volt. Stabilization is 1 volt. Version No. 6 can be shut down in an emergency by an independent sensor. Through the RMV-K, Samovar fully controls the boost, the stabilization and the emergency heating shutdown.

Attention! Since the RMV-K works only with whole values when the stabilization voltage is set, this has a peculiarity when Samovar works in the [pre-flooding](#) mode.

3. Regulator SEM_AVR

In the configurator select "Power regulator" — **SEM_AVR**. In **PlatformIO** it corresponds to the **Samovar_sem** environment (the **SAMOVAR_USE_SEM_AVR** directive). The firmware enables the RMV-K protocol support and the general regulator support automatically.

Compatible with Samovar. It has a peculiarity: in the Samovar interface the power delivered to the load is set and displayed not in volts (as with the two previous regulators) but in watts.

To work correctly with Samovar, use the Stab-avr.ino sketch that comes with Samovar and is located in the directory ...Samovar\Stab-avr

Voltage regulators with no external control over protocols.

For example:

RM2, RM2M, RM2-Pro (hereinafter simply RM) .

All RM models have power isolation on the control contacts by default, and no modification is needed to work with Samovar.

If Samovar does not control the regulator over UART, select "Power regulator" — **"Do not use"** («Не использовать») in the configurator. In PlatformIO it corresponds to the **Samovar_no_power** environment: the **SAMOVAR_USE_POWER**, **SAMOVAR_USE_RMVK** and **SAMOVAR_USE_SEM_AVR** directives are off. The [NBK](#) mode is not available in such a build.

Controlling these regulators comes down to two parameters:

- Switching the regulator itself on/off.

- Starting/stopping the "Boost" («Разгон») mode

This is implemented with two relays whose contacts control the regulator. On the Samovar command "Turn heating on" («Включить нагрев»), the contacts of relay No. 1 switch the RM on and at the same time the contacts of relay No. 4 give the command to turn on the "Boost" mode.

When the vapor temperature reaches the value specified in the file `Samovar_ini.h` : (39° in this case)

//Vapor sensor temperature at which the column switches from boost mode to working mode

```
#define CHANGE_POWER_MODE_STEAM_TEMP 39
```

Relay No. 4 opens its contacts and the RM turns off the boost mode. When the run is finished, Samovar switches the RM off with the contacts of relay No. 1.

Attention.

If you turn the heating off and then on again, but the vapor sensor temperature is above the value specified in the sketch, the boost will not turn on.

When the safety system sensors trigger, Samovar switches the RM off completely.

Note:

- All the listed regulators work as independent units and can be used with Samovar without a UART connection, following the principle and algorithm of the RM regulator.

- Help with choosing: [regulator overview](#)

- Besides controlling the regulators, Samovar can [control an additional boost heating element](#)

Questions and answers. FAQ

Samovar and the voltage regulator show different power readings. How to fix it?

By and large the absolute power is not that important, but it is possible to make the displayed power readings on the regulator and on Samovar match, and essentially there is only one method.

None of the regulators, Samovar included, measures power. They calculate it like calculators, from two values: the voltage currently applied to the heater and the entered resistance of that heater.

The entered resistance is constant, while the voltage changes. The whole adjustment comes down to choosing the heater resistance value.

You can use the calculated resistance: voltage squared divided by power. The voltage is taken from the nameplate on the heating element. Example for a 3 kW 220 V heating element: $220 \times 220 / 3000 = 16.133333333333$ Ohm.

Or measure it with a multimeter.

*For the **RMV**k regulators and the Simple one built from Chinese modules (**Stab-avr**).*

Attention! *If we are not actually going to heat the boiler, we disconnect the heater.*

- In the **RMV**k you can enter the power and voltage values, and the resistance is calculated automatically. Or enter the resistance and the power is calculated. [Video on how to set it up](#). Watch it, do it, and do not forget to switch the regulator to the watts display mode.

- In the **Stab-avr**, set the actual mains voltage at the moment with the potentiometer and enter the power of your heater.

Attention! *For the Stab-avr, the heating element power must be entered for a mains voltage of 230 volts. If the heater manufacturer specified the power for 220 volts in the datasheet, recalculate it for 230 volts.*

*Start Samovar heating. There is no point in adjusting anything in the Boost mode — the voltage keeps changing all the time, and so does the power. Give the command from Samovar to set some specific voltage on the regulator (say 170 volts for the RMV*k*, or 2000 watts for the Stab-avr). The regulators switch to stabilization mode.*

Compare the power values displayed in Samovar and on the regulator.

Now there are two ways — either change the resistance value in Samovar (web interface: "Settings" («Настройка»), the "Other" («Прочие») tab, the "Heating element resistance for power calculation" («Сопротивление тэна для расчета мощности») field), which is much more convenient, or change the entered power value in the regulator. The goal is to get the values on the regulator and in Samovar as close as possible.

*Regulator **Kvic** .*

It has no power readout. So only Samovar is adjusted, to the power you want to see...

That is the first option

The second is to use a chain: a separate volt-ammeter — regulator — Samovar.

100A

Closed current transformer



Вольтметр Амперметр переменного тока 50-300 в 20 А, измеритель мощности KWS, цифровой электронный измеритель мощности, ваттметр переменного тока, Электрический измеритель с функцией сброса 0-а

★★★★★ 4,8 222 отзыва 345 продано

[Описание](#) [Характеристики](#)

Цвет: 100A-Close



Название бренда	Interest Maker
Номер модели	AC 50-300V 20A 100A Voltmeter Ammeter KWS Power Energy Meter LED Digi
Происхождение	Китай
Сертификация	Евротест (CE), RoHS
Размеры	84.6*44*22.3mm
Товары для ремонта	ELECTRICAL

This is convenient if different heating elements are used from time to time and/or the heating elements of the heating element assembly itself are switched.

The volt-ammeter really measures the consumed current and shows the actual power on its display. So if you want Samovar to show this power, just change the heater resistance value in Samovar on the fly.

Using a boost heating element

Samovar can control an additional boost heating element in the "Rectification", "Distillation" and "Beer" modes. The boost heating element is connected to relay No. 4; the wiring is described in the [assembly guide](#).

There is no separate button for the boost heating element in the web interface: Samovar switches it on and off by itself based on temperatures. The thresholds are set in the configurator on the "Temperatures" («Температуры») tab ("Boost exit temperature, °C" («Температура перехода из разгона, °C»), "Cooling opening temperature, °C" («Температура открытия охлаждения, °C»), "Boost heating element threshold, °C" («Порог разгонного ТЭНа, °C»); when building in the Arduino IDE, the same lines are in [Samovar_ini.h](#).

Boost heating element operation in "Rectification" («Ректификация») mode

To use the additional boost heating element function, nothing needs to be changed in the sketch. This function is enabled by default.

The logic is similar to that of the [RM](#) in "Boost" («Разгон») mode:

On the "Turn on heating" («Включить нагрев») command (in the web interface, in the mobile app, or with the [universal button](#)), besides Samovar switching on the main heating element, relay No. 4 switches on a contactor or solid-state relay, which in turn supplies voltage to the boost heating element. The status line then shows "Boost" («Разгон»). When the steam temperature reaches the value set in the settings (39° by default):

```
//Steam sensor temperature at which the column switches from boost mode to working mode
#define CHANGE_POWER_MODE_STEAM_TEMP 39
```

Relay No. 4 opens its contacts and switches off the boost heater. Samovar prints the message "Boost finished. Stabilization/self-reflux." («Разгон завершён. Стабилизация/работа на себя.»)

With a UART-controlled regulator, the boost heating element switches off earlier if the boiler temperature rises above the cooling opening temperature (77° by default), even when the steam has not yet reached 39°. If the boiler sensor is not connected, Samovar does not control the boost heating element.

If the command "Turn on heating" («Включить нагрев») has not been given, or the sensor temperature is above the value set in the sketch, the boost will not start.

Boost heating element operation in "Distillation" («Дистилляция») mode

The boost heating element switches on together with the main one when distillation starts. It switches off at the first transition to the next line of the [distillation program](#). If the checkbox "Use boost heating element during boiling" («Использовать разгонный тэн при кипении») is cleared in the settings ("Settings" («Настройки») → "Beer" («Пиво») tab), the boost heating element is switched off earlier, as soon as Samovar detects the start of boiling.

Boost heating element operation in "Beer" («Пиво») mode

The heating element is connected as described above.

But the algorithm is different.

During mashing, the boost heating element is governed by the "Boost heating element threshold, °C" («Порог разгонного ТЭНа, °C») parameter (the "Temperatures" («Температуры») tab of the configurator); in the file [Samovar_ini.h](#) the line is:

```
#define ACCELERATION_HEATER_DELTA 4
//If the difference between the target temperature and the current temperature (for beer and sous-vide modes) is greater than the set value,
then with a UART-controlled regulator relay No. 4 additionally switches on the boost heating element; with a smaller difference, the boost
heating element switches off
```

Enabled by default; 4 is the value in degrees C°. Works only with a UART-controlled regulator.

When switched on, Samovar commands the regulator to supply full power to the main heating element. If the boiler temperature is lower than the one set in the line of the [mashing program](#) by 4°, Samovar switches on relay No. 4, which in turn commands the boost heating element to switch on. If the difference is less than 4°, the boost heating element does not switch on and heating comes only from the main heater.

When the next temperature rest is complete, or the operator gives the "Next program" («Следующая программа») command, Samovar compares the boiler temperature with the one set in the next program line. If the difference is greater than 4°, the boost heating element is used.

If the "Turn on heating" («Включить нагрев») command has not been given, or the delta is less than the set one, the boost heating element does not switch on.

When the wort is boiling, the boost heating element switches on if the "Use boost heating element during boiling" («Использовать разгонный тэн при кипении») checkbox is set, otherwise it is switched off. During PID controller autotuning the boost heating element is not used.

Samovar 7.00: In "[Cheesemaking](#)" («Сыроварение») mode relay No. 4 is occupied by the brine drain valve, so the boost heating element and this valve cannot be connected at the same time. In the "BK" («БК»), "NBK" («НБК») and "Sous-vide" («Су-вид») modes the boost heating element is not used.

Cooling

How it works (using Rectification mode and the original sketch as an example).

When the cooling start condition is reached (the boiler sensor temperature has reached 77°), the command to open the water supply valve (relay No. 3) is given, and at the same time a PWM control signal is sent to turn on the pump. For the first ten seconds the pump runs at starting power (40% by default), and then the PWM regulator selects the speed so as to keep the coolant temperature (hereinafter the coolant) at the dephlegmator outlet according to the settings, without dropping below the minimum power (10% by default). As the column warms up and the coolant temperature rises, the pump will increase the flow rate. When boost is finished, the pump again gets starting power for a few seconds to increase the water flow, after which the regulator continues its work. If a TCA sensor is connected and its temperature is above 39° and above the water temperature, the pump works harder: the regulator treats the water as 3° warmer than the setpoint.

Cooling is configured in two places:

1. In the configurator on the "Temperatures" («Температуры») and "Pumps" («Насосы») tabs or, when building in the Arduino IDE, in the lines of the [sketch](#):
//Значение температуры датчика в кубе, при котором откроется клапан подачи воды и включится насос подачи воды
#define OPEN_VALVE_TANK_TEMP 77

In the configurator it is "Cooling opening temperature, °C" («Температура открытия охлаждения, °C»). At a boiler temperature of 77°, the command to open the water supply valve is given, a PWM control signal is sent to turn on the pump, and the message "Open the water supply!" («Откройте подачу воды!») is sent to the operator with a sound signal. The message is sent in any case, even if neither a pump nor a valve is used. If a TCA sensor is connected, cooling opens even earlier - when the TCA temperature rises to 70°.

#define DELTA_T_CLOSE_VALVE 20

//The difference between the set water cooling temperature and the water temperature up to which the pump keeps running after the distillation process is finished in the distillation and rectification modes. The value is in degrees C°, it can be changed. Turning it off is strongly discouraged. (for details see [here](#)). In the configurator it is "Cooling shutdown temperature margin, °C" («Запас температуры выключения охлаждения, °C»).

//Нижнее значение (в процентах), для работы ШИМ-регулятора для насоса. Необходимо подобрать такое значение, которое обеспечивает поток воды

#define PWM_LOW_VALUE 10

In the configurator it is "Minimum pump power, %" («Минимальная мощность насоса, %»). As is well known, a motor needs a higher voltage to start than to keep running. This line sets the lowest possible value, in %, at which the motor can keep the water supply pump running without stopping.

//Значение (в процентах), с которого начнет работать ШИМ-регулятор для насоса. Необходимо подобрать такое значение, которое гарантированно обеспечивает запуск мотора

#define PWM_START_VALUE 40

In the configurator it is "Pump starting power, %" («Стартовая мощность насоса, %»). The value, in % of the rated voltage, at which the pump motor is guaranteed to start.

That is, if your pump (assuming it is already installed in the system and has liquid in it) starts reliably when supplied with 4 volts and does not start with less, then with its rated voltage of 12 volts this is 30% of the rated value. So the value 40 can be replaced with 30.

2. In the settings, on the [Temperature](#) («Температура») tab - the water sensor setpoint. The pump will hold this temperature during the process. After heating is turned off, Samovar turns off the pump and closes the water supply valve when one of the conditions is met: the water temperature has dropped 20° below the water sensor setpoint and at least three minutes have passed since heating was turned off, or the boiler, which had been heated above 77°, has cooled below 70°. In Rectification mode with a TCA sensor connected, the TCA must additionally cool to 65°. The message "Close the water supply!" («Закройте подачу воды!») will be sent to the operator.

Samovar 7.00: previously the water was closed only when the water cooled down. In summer, when the inlet water is warm, the pump could run until the operator intervened; now the water is also closed when the boiler cools down.

Water protection.

If the water sensor does not respond while heating is on, Samovar performs an emergency heating shutdown.

At a water temperature of 65° and above, Samovar sounds a signal and shows the message "Critical water temperature!" («Критическая температура воды!»); at 70° and above it also reduces the regulator power; at 75° and above it performs an emergency heating shutdown. On an emergency shutdown the valve closes and the pump stops.

If on the [Other](#) («Прочие») tab the "Use water flow control" («Использовать контроль потока воды») checkbox is set and a flow sensor is installed, then with cooling on and no flow for 20 seconds, Samovar performs an emergency heating shutdown with the message "Emergency shutdown! Water supply stopped." («Аварийное отключение! Прекращена подача воды.»)

In **Distillation** mode, cooling works the same as in rectification, except that the TCA cooling is not checked when closing.

In **Wash Column** mode the valve opens the same way, and the pump power is set with the "Pump speed" («Скорость насоса») slider on the BK page or automatically from the steam temperature setpoint in the program line, see [Wash Column](#).

In **NBK** mode cooling opens only while heating is on; it closes, as in rectification, when the water has cooled below the water sensor setpoint by DELTA_T_CLOSE_VALVE (20°) or the boiler has cooled.

In **Beer** mode the valve and pump turn on at the "Cooling" («Охлаждение») program line and, when the wort needs to be cooled, at the "Fermentation" («Брожение») line; the pump then runs at full power. The same pump output turns on the wort transfer pump at the lines where this is set in the [mashing program](#).

In **Cheesemaking** mode the valve and pump are used at the cooling stages, see [Cheesemaking Mode](#).

In **Sous-vide** mode cooling is not used.

For the implementation, see the [assembly guide](#).

Emergency Button. Additional Emergency Sensors

You can add to Samovar a button or a sensor (for example, a water leak sensor) that works by closing a contact. The input depends on the board: GPIO35 on ESP32 DEVKIT, GPIO48 on ESP32-S3. The emergency button is not supported on LILYGO boards.

When a low level is applied to the input (a short to ground), Samovar performs an emergency stop: it turns off heating and puts the voltage regulator into sleep mode, closes the valve and stops the water supply, stops the take-off pump, the stirrer and the I2CStepper dosing pump, sounds the buzzer and sends the message "Emergency shutdown: the emergency button is pressed" («Аварийное отключение: нажата аварийная кнопка»). You can connect the button and several sensors in parallel. The only requirement is that the sensors are rated for 3.3V. The emergency mode can be exited only by a reboot. If the button is pressed (the sensor is closed) when the power is switched on, Samovar goes into emergency mode right after startup.

Warning. On ESP32 DEVKIT the GPIO35 input has no internal pull-up, so an external resistor (10 kOhm) between the input and 3.3V is required. ESP32-S3 uses an internal pull-up, so no external resistor is needed.

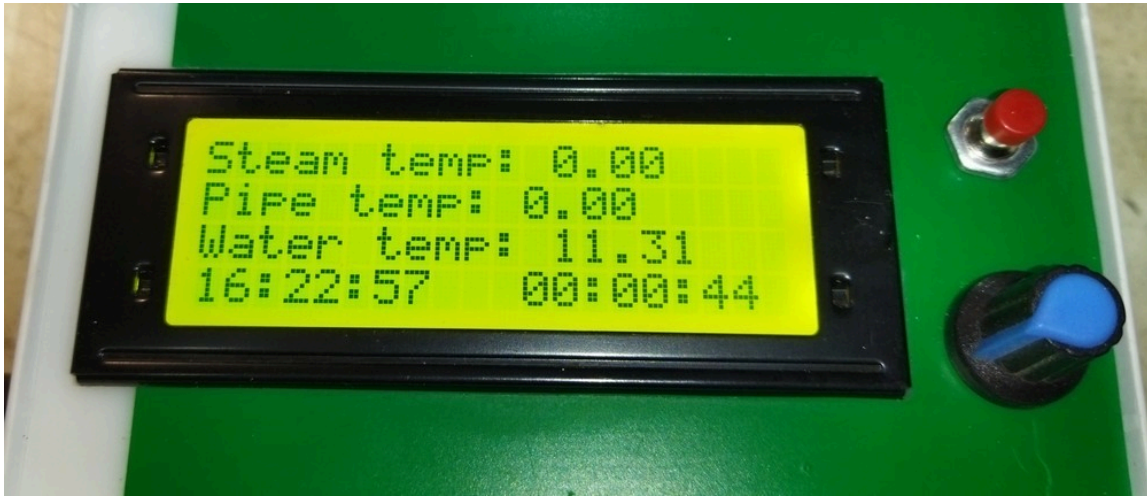
The button is disabled by default. **Samovar 7.00:** the button is enabled on the "Equipment" («Оборудование») tab of the configurator (the "Use the emergency button" («Использовать аварийную кнопку») checkbox); in PlatformIO there is a ready-made environment **Samovar_alarm_button** for this. If you configure it manually, in the file **Samovar_ini.h** you need to replace

```
///define USE_ALARM_BTN
with
#define USE_ALARM_BTN
```

More details are in the [Manual](#) for DIY assembly

LCD display and the universal button

Recommended setup: a four-line display, an encoder with a button, and a separate button with normally open contacts. The connection is described in the [assembly guide](#). Samovar also works without the display, encoder and button - all control is available in the web interface.



The universal button.

What a button press does depends on the Samovar operating mode. A short press and a hold of about two seconds are distinguished.

In rectification mode:

- If heating is off - pressing the button switches heating on; while the vapor temperature is below the [set value](#), the Boost stage runs
- If heating is on - the first press is equivalent to the "Start take-off" («Начать отбор») command
- After take-off starts, pressing the button pauses and resumes the take-off pump. If a "Pause" line of the program is running, the press moves to the next line.
- Holding the button while a program is running - move to the next program line, like the "Next" («Далее») button in the web interface.
- During take-off pump calibration, a short press finishes the calibration.

In distillation mode:

- If heating is off – heating switches on
- If heating is on – a short press moves to the next line of the distillation program, a hold ends the distillation and switches heating off

In wash column and NBK mode:

- If heating is off – heating switches on
- If heating is on – the process ends, heating switches off

In Beer mode:

- If Samovar is not running a program, a program starts
- If a program is running, pressing the button moves to the next program line.

In Cheesemaking mode - the same as in Beer mode; a hold ends the program.

In Sous-vide and Lua modes the button does nothing.

When power is switched on:

- If held for 2 sec. - you can connect to Samovar as to a standalone [access point](#).
- If you hold the encoder button for 2 sec. when switching on the power - the saved WiFi settings will be deleted.

The display.

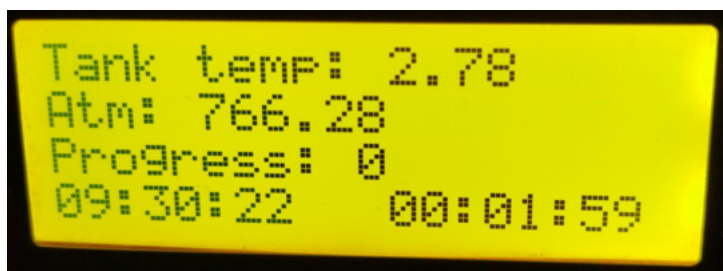
When power is switched on, the display shows the progress of connecting to WiFi and to the samovar-tool.ru server. Then, with heating off, the display shows three screens that replace one another. The screens are changed by turning the encoder.

The first one shows:



Steam T - vapor temperature
 Pipe T - column section temperature
 Water T - water temperature
 The current time and the Samovar run time

The second



Tank T - boiler temperature
 Atm - atmospheric pressure
 Progress - completion of the current take-off program line, in percent

The third



Get power - heating (ON - switch on, OFF - switch off). Switching off through this item stops the current process.
 Setup - settings
 IP - the Samovar address in your local network

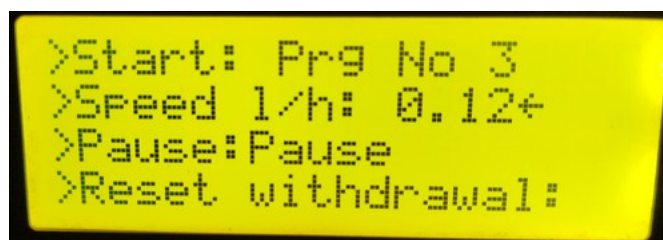
If there is a (>) symbol before a parameter, the item performs an action or changes its value when the encoder is turned
 If there is a (^) symbol before a parameter, it is a numeric value.

The /Setup and Prog> items open submenus, <BACK returns.

To change a parameter, press the encoder; an arrow (<) appears to the right of the item. Each press moves the arrow down. Turning the encoder changes the item's value or opens the submenu. Turning with the encoder button pressed changes the value in steps 10 times larger.

When the arrow is on the last line, pressing makes it disappear and you can turn the encoder to go to the next screen. If you press once more - the arrow appears on the first line.

After heating is switched on, a fourth screen is added - the current state.



Start - number of the program being run -> move to the next program (in Rectification mode)
 Speed l/h - take-off rate -> increase/decrease by 0.01 l/h
 Pause - pause/resume (the label changes to Continue)
 Reset withdrawal - reset the process: the program stops, heating switches off.

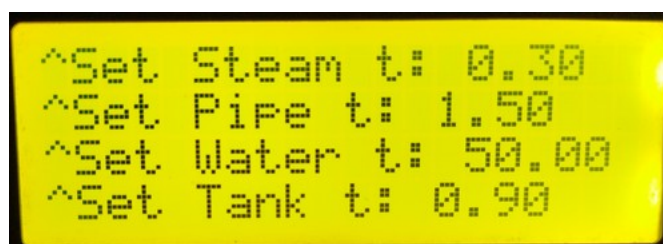
The Setup item submenu, first screen



Steam T - correction of the vapor temperature sensor
 Pipe T - correction of the column section temperature sensor
 Water T - correction of the water temperature sensor
 Tank T - correction of the boiler temperature sensor

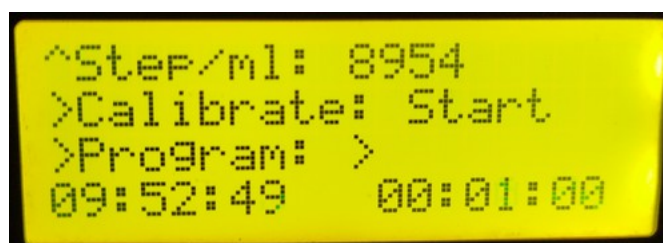
The values can be changed. Press the encoder, an arrow appears. The line next to the arrow can be changed: turning increases or decreases the values by 0.01. The changes are saved to the settings when you leave the submenu through the <BACK item and then appear in the Samovar web interface, on the ["Temperature" \(«Температура»\)](#) tab. If saving failed, Samovar shows the message "Menu settings not saved" («Настройки меню не сохранены»).

The Setup item submenu, second screen



Set Steam T - setpoint of the vapor temperature sensor
 Set Pipe T - setpoint of the column section temperature sensor
 Set Water T - setpoint of the water temperature sensor
 Set Tank T - setpoint of the boiler temperature sensor

The Setup item submenu, third screen



Step/ml - the number of stepper motor steps to take off 1 milliliter
 Calibrate - [peristaltic pump calibration](#) - the pump starts, the stepper motor speed is adjusted with the encoder. When 100 ml is reached, pressing the encoder finishes the calibration, and Samovar calculates and saves the number of steps per 1 ml.
 Prog - go to the take-off program submenu

The Setup item submenu, fourth screen

<BACK - return to the initial screen, saving the changed settings

The Prog item submenu



The first screen shows the first line of the take-off program. Editing is not available

Type - the line type, shown as a character code (72 - H, 66 - B, 84 - T)

Volume - volume

Speed - take-off rate in liters

Capacity - container number.

The second screen



Reset WiFi - reset the WiFi settings, Samovar will reboot

<BACK - return to the Setup submenu

Attention!

Holding the button or the encoder at startup does not reset the saved parameters: the button starts the access point, the encoder deletes only the WiFi settings.

WEB Interface: Main Page

Samovar 7.00: the main page has been rebuilt. The readings are grouped into the cards "Actions" («Действия»), "Sensors" («Датчики»), "Impurity detector" («Детектор примесей»), "Current line" («Текущая строка»), "Regulator" («Регулятор»), "Pressure" («Давление») and "System" («Система»), with a diagram of the installation showing current temperatures in the center. The take-off rate and the "Fix hearts take-off T" («Зафиксировать T отбора тела») button have been moved from the "Additional" («Дополнительно») tab to the "Regulator" card. A message history (the ≡ button) and a dark theme (the ☾ button) have been added. If the banner "Configuration loaded with an error" («Конфигурация загружена с ошибкой») is shown, do not turn on heating: first check the sensor assignments and the settings named in the message.

After power-up, Samovar opens with heating off. It shows the sensor readings, but the process has not been started yet. The main page of the "Rectification" («Ректификация») mode is described below; in other modes the set of cards and buttons differs, see [Samovar Operating Modes](#).

Header and readings

Header and readings

1. **Operating mode**
Ректификация, Программа, Дополнительно

2. **Самовар v. 7.00**
Расчёт I2CStepper Внешний насос График Настройки

3. 12:34:56 Работа 01:12:30

4. 5.

6. **Тело; Прг №2; Осталось:00:45|03:10;Т тела пар:78.150;Т тела царга:77.900** строка 2 из 4

Этап: Строка программы · Строка: 2 · Завершение: объём: не ниже 500 мл по объёму · Далее: строка 3

7. Отобрано 350 мл Скорость 0.50 л/ч Прогресс 35%

8. **ДЕЙСТВИЯ**
Выключить нагрев
Поставить на паузу
Далее
Сбросить процесс
График Настройки
отбор: задано 0.5 л/ч, применяется 0.5 л/ч (по программе)

9. **ДАТЧИКИ**
Пар уставка 78.15 **78.2 °C**
Царга уставка 77.90 **77.9 °C**
Δ пар – царга **-0.3 °C**
Куб **84.3 °C**
Вода **22.4 °C**
ТСА **40.1 °C**

10. **ДЕТЕКТОР ПРИМЕСЕЙ**
Состояние Тренд
Стабильно, слежу 0.000 °C/мин

11. **ТЕКУЩАЯ СТРОКА · 2 ИЗ 4**
Тело ёмкость 2
35 % · объём строки 500 мл
Скорость Дельта
0.5 л/ч -0.25 °C
Напряжение строки Время строки
100 В 1 ч 00 мин
СЛЕДУЮЩАЯ 3 · Предзахлёб · 3000 мл · 1.2 л/ч · ёмк. 3

12. **РЕГУЛЯТОР**
Текущее напр.: **180.5 V** Целевое напр.: **180.0 V**
Режим **1** Мощность **1450 Вт**
Напряжение регулятора:
Установить напряжение
Скорость, л/ч Установить
Зафиксировать T отбора тела

13. **ДАВЛЕНИЕ**
Атмосферное В колонне:
752.3 мм рт.ст. 1.2 мм рт.ст.

СИСТЕМА
Нарп=152000; BME temp=24.5; RSSI = -58; Свободно: 300000; ШИМ насоса воды: 0

Main page of the "Rectification" («Ректификация») mode

1. **Operating mode**
The first tab is named after the current mode. The mode is changed in Settings → Basic.

Version

- 2** The version of the installed firmware.
- 3** **Connection and time**
The connection indicator for Samovar (green — data is arriving, red — connection lost), the current time and the uptime since power-on.
- 4** **Message history**
Opens the messages saved in this browser, with the "Clear history" («Очистить историю») and "Close" («Закрыть») buttons.
- 5** **Theme**
Switches the page between the light and dark theme.
- 6** **Status line**
The current state of Samovar: the type and number of the line being executed, the remaining time, the fixed hearts temperatures, pause, warm-up, stabilization. Next to it are the line number and the progress bar.
- 7** **Take-off**
The collected volume, the actual take-off rate and the progress of the current line.
- 8** **Command buttons**
"Turn heating on/off" («Включить/Выключить нагрев»), "Pause/Resume" («Поставить на паузу/Продолжить»), "Start take-off/Next" («Начать отбор/Далее»), "Reset process" («Сбросить процесс»), and links to "Chart" («График») and "Settings" («Настройки»).
- 9** **Temperatures**
Readings of the vapor, column, boiler, water and TCA sensors. Colors are set in Settings → Temperature.
- 10** **Impurity detector**
The state of the impurity breakthrough detector and the vapor temperature trend. The card is visible if the detector is enabled in the settings.
- 11** **Installation diagram**
The boiler, column, dephlegmator, TCA, take-off unit and jars with the current temperatures, pressure and heating. The jar of the current line is highlighted.
- 12** **Right column**
The cards "Current line" («Текущая строка»), "Regulator" («Регулятор»), "Pressure" («Давление») and "System" («Система») — see the next screenshot.
- 13** **Detailed explanation**
The stage, the completion condition, the wait and how to continue, the next line. The block is present only on firmware that sends the new stage data.

What is displayed

Version and mode — the firmware version is shown in the header next to the "Samovar" («Самовар») label. The first header tab is named after the current mode (for rectification, "Rectification" («Ректификация»)). The mode is selected in [Settings → Basic](#): Rectification («Ректификация»), Distillation («Дистилляция»), Beer («Пиво»), Wash column («Бражная колонна»), Continuous wash column («Непрерывная бражная колонна»), Cheesemaking («Сыроварение»).

Connection and time — a green indicator means the page is receiving data from Samovar. If several requests in a row go unanswered, the indicator turns red and the message "Connection lost!" («Обрыв связи!») appears. Next to it are the current time and "Uptime" («Работа») — the time since Samovar was switched on.

Status line — the current state: "Off" («Выключено»), "Column warm-up" («Разгон колонны»), "Warm-up complete. Stabilization/Self-reflux" («Разгон завершен. Стабилизация/Работа на себя»), "Stabilization complete/Self-reflux" («Стабилизация завершенна/Работа на себя»), the type and number of the program line being executed (for example, "Hearts; Prg #2" («Тело; Прг №2»)), "Pause" («Пауза»), "Program completed" («Выполнение программы завершен»). During take-off, "Remaining:" («Осталось:») is added to the status — the estimated time to the end of the line and of the whole program — and, after the hearts temperatures are fixed, "Hearts T vapor" («Т тела пар») and "Hearts T column" («Т тела цапра»). Next to it are shown the line number ("line 2 of 4" («строка 2 из 4»)), the line progress bar and the take-off totals: "Collected" («Отобрано»), "Rate" («Скорость»), "Progress" («Прогресс»).

Detailed explanation — on firmware with the new data, under the status are shown the stage, the current line, its completion condition and target, the reason for waiting and how to continue, and the next line. If it says "Detailed explanation unavailable" («Подробное объяснение недоступно»), the firmware did not send the new data; the usual readings and status remain available.

Key values — a card for narrow screens only (smartphone): "Δ vapor – column" («Δ пар – цапра»), "Vapor to setpoint" («Пар к уставке») (the deviation of the vapor temperature from the fixed hearts take-off temperature), "Boiler" («Куб») and a short impurity detector state. On a wide screen these values are visible in the "Sensors" and "Impurity detector" cards.

Sensors — the temperatures "Vapor" («Пар»), "Column" («Цапра»), "Δ vapor – column" («Δ пар – цапра»), "Boiler" («Куб»), "Water" («Вода»), "TCA" («ТСА»). The color of each sensor is set in [Settings → Temperature](#). For vapor and column the setpoint is shown after the hearts temperatures are fixed. A zero or an obviously wrong value must be fixed in Settings → Temperature before starting.

Impurity detector — the card is visible if the detector is enabled in the settings. It shows the state ("Stable, monitoring" («Стабильно, слежу»), "Temperature rising: reducing the rate" («Рост температуры: снижаю скорость»), "BREAKTHROUGH (PAUSE)" («ПРОСКОК (ПАУЗА)»), on heads and tails — "observation" («наблюдение»), after a line start or resume — "Pause after line start" («Пауза после старта строки»), "Waiting for vapor to stabilize" («Жду стабилизации пара»), "Collecting history" («Набираю историю»)) and the vapor temperature trend in °C/min. On heads and tails lines the detector only collects observations: it does not reduce the rate and does not pause the take-off. More in [Hearts take-off](#).

Installation diagram — the boiler, column, dephlegmator, TCA, take-off unit and jars. The diagram is labeled with the current temperatures, the pressure in the boiler or at start, and the heating voltage and power. During operation the cooling water flow, heating, take-off pump and product drops are animated; the jar receiving the take-off for the current program line is highlighted. On pause, the label "PAUSE" («ПАУЗА») appears on the boiler. If "Use an I2C pump above the pasteurization section" («Использовать I2C-насос над ЦП») is enabled on the [Pump](#) tab and the pump board was found at startup, a pasteurization section («ЦП») is drawn above the column, and to the left of it a second pump with a separate heads take-off line; it is animated while it pumps.

Right column: current line, regulator, pressure, system

ТЕКУЩАЯ СТРОКА · 2 ИЗ 4

Тело ёмкость 2

35 % · объём строки 500 мл

Скорость

Дельта

0.5 л/ч

-0.25 °C

Напряжение строки

Время строки

100 В

1 ч 00 мин

СЛЕДУЮЩАЯ 3 · Предзахлёб · 3000 мл · 1.2 л/ч · ёмк. 3

РЕГУЛЯТОР

Текущее напр.:

Целевое напр.:

180.5 V

180.0 V

Режим

Мощность

1

1450 Вт

Напряжение регулятора:

Установить напряжение

Скорость, л/ч

Установить

Зафиксировать T отбора тела

ДАВЛЕНИЕ

Атмосферное

В колонне:

752.3 мм рт.ст.

1.2 мм рт.ст.

СИСТЕМА

Неар=152000; BME temp=24.5; RSSI = -58; Свободно: 300000; ШИМ насоса воды: 0

The cards "Current line" («Текущая строка»), "Regulator" («Регулятор»), "Pressure" («Давление»), "System" («Система»)

- 1 Current line**
 The line type, container, progress, volume, rate, transition temperature, voltage and line time, the next line.
- 2 Regulator**
 The current and target voltage (power), the regulator mode and the power in watts.
- 3 Regulator voltage**
 Manual setting of voltage (power): enter a value and press "Set voltage" («Установить напряжение»).
- 4 Rate, l/h**
 Manual change of the take-off rate of the current line. Enter a value and press "Set" («Установить»).
- 5 Fix hearts take-off T**
 The current vapor and column temperatures become the hearts take-off temperatures.
- 6 Pressure**
 Atmospheric pressure and the pressure at start ("At start" («В начале»)) or in the boiler ("In column" («В колонне»)) if a pressure sensor is connected.
- 7 System**
 Free memory, the temperature of the pressure sensor, the Wi-Fi level, free space in the file system, the PWM of the water pump and the state of the I2C pump.

Current line — the line type (Heads, Hearts, Pre-flooding, Pause, Tails, Lua), the container number, the progress bar, the line volume, rate, transition temperature, line voltage (power), the estimated line time and a short description of the next line. The program itself is edited on the "Program" («Программа») tab, see [Take-off Program for Rectification Mode](#).

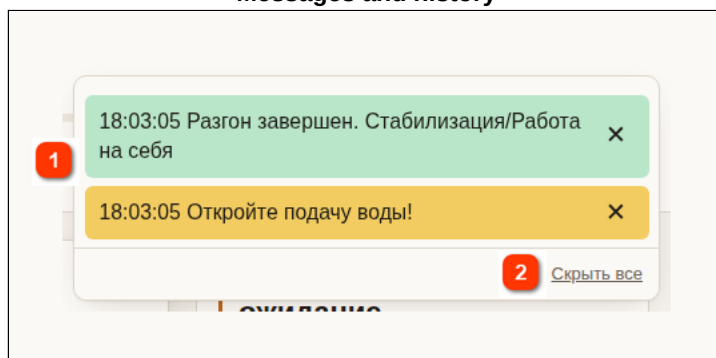
Actions — under the buttons, explanations of active commands may appear: the equipment, what is set, what is actually applied, and where the command came from — the program, manual input, automation or a protection.

Regulator — the current and target voltage (or power, if the regulator works by power), the regulator mode and the power in watts. In the "Regulator voltage" («Напряжение регулятора») field you can set the voltage manually and press "Set voltage" («Установить напряжение»), see [Manual Setting of the Regulator Voltage](#). In the "Rate, l/h" («Скорость, л/ч») field you can change the take-off rate of the current line with the "Set" («Установить») button, see [Manual Setting of the Take-off Rate](#). The "Fix hearts take-off T" («Зафиксировать T отбора тела») button records the current vapor and column temperatures as the hearts take-off temperatures. If the regulator is not used in the firmware, the card is hidden.

Pressure — atmospheric pressure from the BME/BMP sensor and a second value: "At start" («В начале») — the pressure at the moment of start, or "In column" («В колонне») if a pressure sensor in the boiler is connected.

System — Heap (free memory of the controller), BME temp (temperature of the pressure sensor), RSSI (Wi-Fi signal level), "Free" («Свободно») (free space in the file system) and, when the devices are connected, the PWM of the water pump and the state of the external I2C pump.

Messages and history



System messages

1 System messages

New notifications with the time they appeared. The cross closes a single message; emergency ones only after confirmation.

2 Hide all

Removes all messages from the screen. They stay in the history (the ≡ button).

New notifications appear in the top right corner of the page. Each message is closed with a cross; emergency messages are closed only after confirmation. The "Hide all" («Скрыть все») button removes all messages from the screen. The ≡ button in the header opens the history: messages saved in this browser, including those from before the page was refreshed (the last 500 are kept). The "Clear history" («Очистить историю») button deletes them from the browser, "Close" («Закрыть») collapses the history. The list of Samovar messages is on the page [System Messages](#).

Main buttons

Turn heating on / Turn heating off («Включить нагрев / Выключить нагрев») — starts or stops heating for the selected mode. Confirmation is required when turning off. If the vapor, water and boiler sensors are not assigned in the settings, heating does not turn on and the message "Sensors not assigned" («Датчики не назначены») appears on the screen.

Start take-off / Next («Начать отбор / Далее») — starts the rectification program. While heating is off, the command is not executed. If the column is still warming up or stabilizing, the page asks you to confirm the start. During take-off the button is called "Next" and, after confirmation, moves Samovar to the next program line.

Pause / Resume («Поставить на паузу / Продолжить») — stops and resumes take-off, see [Pause](#).

Reset process («Сбросить процесс») — after confirmation, ends the current process with the mode's standard command and turns heating off.

Chart («График») — opens the current local chart, see [Local Logs and Charts](#). **Settings** («Настройки») — opens the Samovar parameters.

Tabs

Settings: Main

Samovar 7.00: the impurity detector is switched on with a single "Use detector" («Использовать детектор») switch and works only in rectification. During heads it only observes; on a critical temperature rise it pauses the take-off. Smooth speed reduction is a separate "Reduce take-off rate when temperature rises" («Снижать скорость отбора при росте температуры») switch.

On this tab you choose the operating mode and the general limits. After changing something, press **Save** («Сохранить») at the bottom of the page. The mode changes without rebooting the ESP32, but only when the heating is off: while a process is running Samovar does not accept the form and shows an error at the "Operating mode" («Режим работы») field.

Error: I2CStepper недоступен: HTTP 404 — File not found ✕

Настройки

Основные Температура Насос Пиво Сыр НБК Прочие I2CStepper

Режим работы **Ректификация**

Ректификация, дистилляция и БК до температуры в кубе: 0 °C

Время для контроля завершения процесса дистилляции: 0 мин

- ☐ Использовать корректировку давления
- ☐ Использовать датчик уровня флегмы при ректификации
- ☐ Не показывать схему процесса

⚙️ Параметры оборудования (колонны)

Диаметр: 1.5 дюймов

Высота насадки: 0.50 метров

Плотность насадки: 80 % (0-100)

Детектор проскока примесей (ректификация)

- ☒ Использовать детектор
 - ☐ Снижать скорость отбора при посто температуры
- Шаг снижения скорости: 0 % (0 = по умолчанию 5, максимум 20)

Давление, при котором включится авария: 0 мм. Рт. ст.

Нулевая точка МРХ: 0

Отсчётов МРХ на 1 мм рт. ст.: 0

Настройки зуммера

- ☐ Использовать зуммер для нотификации
- ☐ Использовать зуммер для нотификации в браузере
- ☐ Включать зуммер при смене программы

Самотестирование

Внимание: включится подача воды (насос или клапан), запустится перистальтический насос отбора, сервопривод пройдёт 5 положений ёмкостей.

Сохранить

На главную

Редактор

Settings → Main («Настройки → Основные»)

1 **Operating mode («Режим работы»)**
Mode selection: Rectification («Ректификация»), Distillation («Дистилляция»), Beer («Пиво»), Wash column («Бражная колонна»), Continuous wash column («Непрерывная бражная колонна»), Cheesemaking («Сыроварение»). It can be changed only while the heating is off.

2 **Process end temperature («Температура окончания процесса»)**
The boiler temperature at which distillation, BK and the take-off after the rectification program finish.

- 3 Time for distillation completion check («Время для контроля завершения дистилляции»)**
Minutes without a rise in the boiler temperature after which distillation or BK finishes. 0 - do not monitor.
- 4 Pressure correction («Корректировка давления»)**
Correction of the sensor readings for atmospheric pressure.
- 5 Reflux level sensor («Датчик уровня флегмы»)**
Works only in builds with a reflux level sensor.
- 6 Column parameters («Параметры колонны»)**
Diameter, packing height and packing density - for the program calculator.
- 7 Use detector («Использовать детектор»)**
The main switch of the impurity breakthrough detector.
- 8 Reduce take-off rate («Снижать скорость отбора»)**
Smooth rate reduction when the vapor temperature rises; works only with the detector.
- 9 Rate reduction step («Шаг снижения скорости»)**
The percentage of a single rate change (0 = 5%, maximum 20%).
- 10 Alarm pressure («Давление аварии»)**
The excess pressure in the boiler (mmHg) at which Samovar goes into an emergency. 0 - off.
- 11 Buzzer («Зуммер»)**
Sound on the device, sound in the browser, signal on a program line change.
- 12 Self-test («Самотестирование»)**
A check of the water, the take-off pump and the servo. Only while the heating is off.
- 13 Save («Сохранить»)**
Writes the settings of all tabs. Next to it are "Home" («На главную») (leave without saving) and "Editor" («Редактор») (file manager).

Mode and process completion

Operating mode («Режим работы») — Rectification («Ректификация»), Distillation («Дистилляция»), Beer («Пиво»), Wash column («Бражная колонна»), Continuous wash column («Непрерывная бражная колонна») or Cheesemaking («Сыроварение»). The Sous-vide («Су-вид») and Lua items are hidden in the list and cannot be selected through it. If a mode is not built into your firmware, Samovar reports it when you save.

Rectification, distillation and BK up to boiler temperature («Ректификация, дистилляция и БК до температуры в кубе») — the normal completion temperature. In distillation and BK the process ends at it; in rectification, after all program lines are done, Samovar keeps taking off until the boiler reaches this temperature. If pressure correction is off, at altitude the boiling temperature is lower and a limit that is too high may never be reached.

Time for distillation completion check («Время для контроля завершения процесса дистилляции») — for how many minutes the boiler temperature (above 90°C) must stay almost unchanged (rise of less than 0.1°C) before completion. Applies in distillation and BK. Zero turns this criterion off, the maximum is 255.

Use pressure correction («Использовать корректировку давления») — a correction of the vapor, column section, boiler and TCA sensor readings for atmospheric pressure (0.037°C per mmHg of deviation from 760). The water sensor is not corrected.

Use reflux level sensor during rectification («Использовать датчик уровня флегмы при ректификации») — includes it in the rectification algorithm if the sensor is provided by the build. In the standard build the reflux level sensor is disabled at compile time, and this switch has no effect.

Equipment (column) parameters

Diameter («Диаметр»), Packing height («Высота насадки») and Packing density («Плотность насадки») describe your column. They are used by the [program calculation calculator](#) (the "Calculation" («Расчёт») tab), and they are written to the process log.

Impurity breakthrough detector (rectification)

Use detector («Использовать детектор») — the main switch of the impurity detector. The detector watches the vapor temperature trend in the Pre-flooding («Предзахлёб»), Hearts («Тело») and Tails («Хвосты») lines of the rectification program; on a dangerous rise it pauses the take-off and resumes it by itself when the temperature returns. The detector state is shown in the "Impurity detector" («Детектор примесей») block on the [main page](#).

Reduce take-off rate when temperature rises («Снижать скорость отбора при росте температуры») — works only together with the detector. On a temperature rise warning the take-off rate is reduced in steps until the trend levels off. The pause on a critical rise happens regardless of this switch.

Rate reduction step («Шаг снижения скорости») — the percentage of a single rate change. 0 - the default 5%, the maximum is 20%.

Pressure at which the alarm triggers («Давление, при котором включится авария») — the limit of the excess pressure in the boiler, mmHg, for a connected pressure sensor. When this pressure is reached, Samovar goes into emergency mode and switches the heating off. 0 - the control is off. A wrong value can cause a false alarm or no expected stop. If the MPX5010DP sensor is enabled in the firmware, two calibration fields for it appear below: "MPX zero point" («Нулевая точка МРХ») and "MPX counts per 1 mmHg" («Отсчётов МРХ на 1 мм рт. ст.»).

Buzzer and self-test

The buzzer parameters separately enable the sound on the device, the sound in the browser and the signal on a program line change. When the first parameter is off, the buzzer is disabled completely.

The **Self-test** («Самотестирование») button checks the actuators: the water supply is switched on (the pump starts or the valve opens), the peristaltic take-off pump starts, and the servo steps through the 5 container positions in turn. You must confirm the action before the start. The test lasts about 15 seconds and ends with the message "Self-test finished." («Самотестирование закончено.»). The self-test does not start while the heating is on or a process is running. Before pressing the button, disconnect any dangerous load and direct the hoses into a safe container.

At the bottom of the page there are three buttons: **Save** writes all settings tabs and returns to the main page; **Home** («На главную») leaves without saving; **Editor** («Редактор») opens the firmware file manager. If the form has unsaved changes, Samovar asks for confirmation before you leave. If a field has an error, saving is not done: the tab with that field opens and the message "Settings not saved. Check the fields: ..." («Настройки не сохранены. Проверьте поля: ...») is shown.

Temperature

Selecting and configuring temperature sensors

Samovar works with DS18B20 digital sensors. Each sensor has its own unique address, an identifier. For Samovar to read values from the right sensor, you must select it in the "Sensor address" («Адрес датчика») drop-down menu. There are five mounting places in total: steam before the dephlegmator, column section, water, boiler and TCA (atmospheric vent tube). Samovar finds up to six sensors on the bus.

Warning.

If a sensor is assigned in the settings but its data stops arriving while heating is on, Samovar will regularly show a message such as "Steam temperature sensor error!" («Ошибка датчика температуры пара!») on the main screen, whether the sensor is used in the process or only for information. These messages are not shown while heating is off.

[Rescanning](#) the sensors is possible only when no process is running (heating is off).



Settings → Temperature: steam sensor before the dephlegmator («Настройки → Температура: датчик пара перед дефлегматором»)

1

Sensor address («Адрес датчика»)

Select the address of the required sensor from the drop-down menu. The list contains the sensors found during scanning and the NONE item (no sensor is installed at this place).

2

Correction («Корректировка»)

The sensor readings can be corrected. You can set both positive and negative values. The correction is added to the raw sensor reading. If pressure correction is enabled on the Main («Основные») tab, a correction for atmospheric pressure is additionally added to all sensors except the water sensor.

3

Setpoint excursion («Вылет за уставку»)

Applied in the "Rectification" («Ректификация») mode while running the Pre-flooding and Hearts program lines.

When Samovar finishes heads take-off and moves to the hearts take-off program line, it remembers the Steam and Column section temperatures. If the actual sensor temperature deviates from the recorded one by the set value, Samovar pauses take-off, shows the message "Pause by steam T" («Пауза по Т пара») and waits until the temperature returns within the set limit. The value 0 disables control by this sensor.

Warning.

It is possible to cancel pausing take-off when temperatures go beyond the setpoint. See the section "[Hearts take-off](#)".

4

Take-off delay («Задержка отбора»)

The time, in seconds, after which the take-off pump is paused when the temperature goes beyond the setpoint. When this time has passed, Samovar checks the temperature: if it has returned within the limit, take-off continues with the message "Resuming take-off after automatic pause" («Продолжаем отбор после автоматической паузы»); if not, the delay countdown starts again.

Setting up the column section sensor is the same as setting up the steam sensor.

Note.

You can set new Steam and Column section temperature values at any moment with the "Fix hearts take-off T" («Зафиксировать Т отбора тела») button in the "Regulator" («Регулятор») block on the [main page](#).

Setting up the water sensor is similar to the previous ones, but with a significant difference.

Settings → Temperature: water sensor («Настройки → Температура: датчик воды»)

1

Water sensor setpoint («Уставка датчика воды»)

The Water sensor setpoint field takes the actual value that the cooling supply pump must maintain. You can change it at any moment **while Samovar is running**. If you leave 0, Samovar uses 48°C.

If a bypass valve is used instead of a pump, the valve opens when the setpoint is exceeded by 1°C and closes when the temperature drops 1°C below the setpoint.

After heating is turned off, the water supply is closed when the water temperature falls 20°C below the setpoint (not earlier than after 3 minutes) or when the boiler cools down.

2

Trigger delay («Задержка срабатывания»)

Not used in current versions. Delay = 0

Setting up the **boiler sensor** is similar to the previous ones, but it also has its own specifics.

Settings → Temperature: boiler sensor («Настройки → Температура: датчик в кубе»)

The boiler sensor is used to end the process by temperature ("[Rectification, distillation and BK up to boiler temperature](#)" («Ректификация, дистилляция и БК до температуры в кубе»)), to monitor the plateau in distillation and BK (wash column), and to hold the temperature in Beer, Sous-vide and Cheesemaking modes. In Beer and Cheesemaking modes, the **Boiler sensor setpoint excursion** («Вылет за уставку датчика в кубе») field sets the temperature tolerance of the program line: from 0.3° to 3°; a value below 0.3° (including 0) is treated as 0.3°; see [Mashing program](#) and [Cheesemaking program](#). The excursion of any other sensor works the same way if that sensor is selected in the program line; for the steam and column section sensors this is the same value that applies during hearts take-off in Rectification. The **Trigger delay** («Задержка срабатывания») field is not used in the current version: the value is saved, but no mode reads it.

Note.

The settings on the Temperature («Температура») tab are the same for all modes. The reading corrections and sensor addresses do not switch together with the operating mode.

Settings → Temperature: TCA sensor («Настройки → Температура: датчик в ТСА»)

1 TCA sensor setpoint («Уставка датчика в ТСА»)

The default setpoint is 43°C; if you leave 0, Samovar uses 43°C.

In Rectification mode the hot TCA threshold equals the setpoint but cannot be lower than 45°C. In Distillation and NBK modes the setpoint itself is used, without the 45°C lower limit.

With heating on and the TCA temperature above this threshold, Samovar shows a high TCA temperature warning once per episode. The next warning is allowed after the TCA cools 2°C below the threshold.

When the same threshold is exceeded, the cooling pump increases the water supply if the TCA is hotter than the water and the cooling valve is open.

The emergency threshold does not depend on the setpoint: when the TCA temperature is at or above MAX_ACP_TEMP (75°C by default), Samovar turns heating off.

For example, with a setpoint of 47°C the warning and, if the conditions above are met, the pump boost are triggered above 47°C; after cooling below 45°C the warning is allowed again; at 75°C and above heating is shut off as an emergency.

2 Trigger delay («Задержка срабатывания»)

Not used in current versions.

1 Scan sensors («Сканировать датчики»)

After you press the button, Samovar polls the bus again and determines which sensors are present or absent, after which the settings page reloads with a fresh list of addresses.

You can connect and disconnect temperature sensors "hot", without restarting Samovar. But scanning is performed only while heating is off: if a process is running, Samovar answers with the message "Sensor scan rejected: process is active." («Сканирование датчиков отклонено: процесс активен.»)

You can connect sensors one at a time and save them in the settings. Connect one sensor, press "Scan sensors" («Сканировать датчики»), choose what it belongs to, press "Save" («Сохранить»), go back to the settings, connect another sensor, scan, choose its location, press "Save", and so on. The main thing is to save the settings after every sensor you add and assign a location to.

To save the selected settings, press "Save"; to leave without saving, press "To main page" («На главную»). If the form has unsaved changes, Samovar asks for confirmation before you leave the page. When a sensor address is changed, its previous readings are reset and collected again.

Warning.

The sensor address at each mounting place must be unique. Make sure the sensor addresses do not repeat when you choose them. This is necessary for Samovar to work normally. However, using one identifier at several addresses at once is not prohibited.

The settings for setpoints, excursions, temperature correction and so on do not move together with the sensor identifier; they belong to a specific mounting place.

Note.

The absence of a temperature sensor is not a problem if it was absent when the process started (even if its address is specified in the settings). If a sensor drops out during the process, Samovar will report it.

Pump

Peristaltic pump motor setup.

One parameter is set for the pump: the number of stepper motor steps needed to take off 1 ml.

Settings → Pump («Настройки → Насос»)

1

Number of stepper motor steps

Motor steps per 1 ml of take-off. Filled in manually or from the calibration result.

2

Pump calibration

Opens the "Stepper motor calibration" («Калибровка шагового двигателя») window.

You can enter this parameter manually, or calibrate the pump: take off 100 ml, Samovar counts the motor steps made to take off 100 ml, divides that number by 100, and the value appears in the "Number of stepper motor steps: ____/ml." («Количество шагов шагового двигателя: ____/мл.») line.

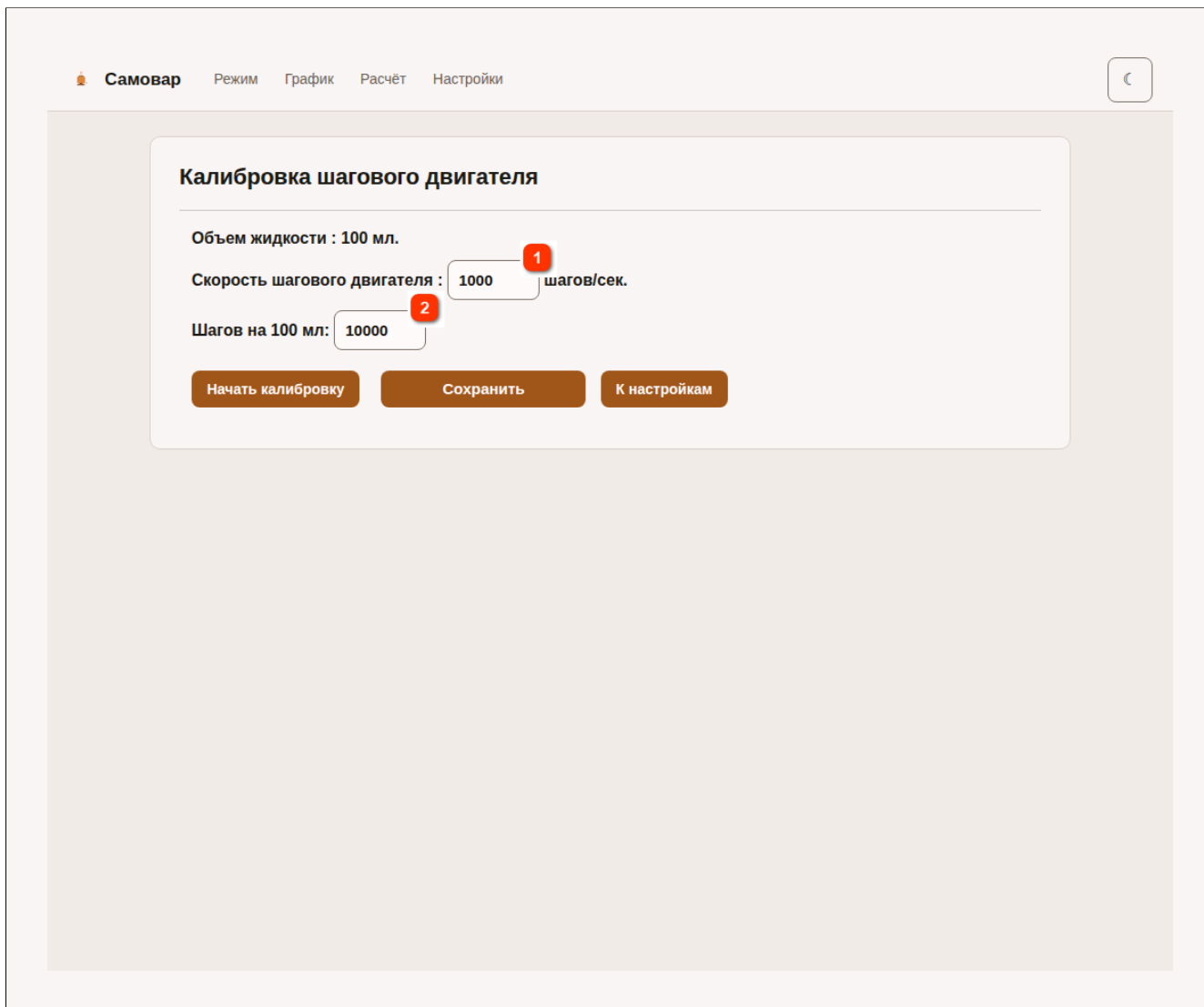
Samovar 7.00: if an external pump is connected to Samovar over the I2C bus (an [I2CStepper](#) module), a second block appears on the tab: the "Use I2C pump above the pasteurization section" («Использовать I2C-насос над ЦП») switch and its capacity in l/h. If there is no I2C pump, the block is hidden. The steps per milliliter for the I2C pump are stored in the board's own memory (Settings («Настройки») → I2CStepper, the "Steps/ml" («Шагов/мл») field) and are calibrated with the "External pump calibration" («Калибровка внешнего насоса») button on the I2CStepper page.

Pump calibration

You can calibrate during a real distillation run, but before the "Start take-off" («Начать отбор») command of the program.

After the message "Stabilization complete, the column is running stably." («Стабилизация завершена, колонна работает стабильно.») wait about 30-40 minutes, direct the pump output into a measuring container and press the "Pump calibration" («Калибровка насоса») button.

The following window opens: "Stepper motor calibration" («Калибровка шагового двигателя»)



Stepper motor calibration («Калибровка шагового двигателя»)

1

Motor speed

Optimal speed of the NEMA17 motor, steps per second (from 1 to 8000).

2

Steps per 100 ml.

The number of steps needed to take off 100 ml, obtained after calibration. You can correct the value manually and press "Save" («Сохранить») (the number must be a multiple of 100). The "To settings" («К настройкам») button returns to the settings page.

Press "Start calibration" («Начать калибровку»). The pump starts. The button label changes to "Capture 100 ml" («Зафиксировать 100 мл»). When 100 ml has been taken off, press the "Capture 100 ml" button. Samovar calculates and remembers the number of steps needed to take off 1 ml. The result is saved to the settings immediately, and the obtained value appears in the "Steps per 100 ml" («Шагов на 100 мл») field.

During calibration the status line of the main page shows "Calibration" («Калибровка»). When calibration ends, Samovar returns to the "Self-circulation" («Работа на себя») state.

Then you can start your take-off program with the "Start take-off" command and continue the distillation in the standard way.

Warning.

You cannot calibrate while the take-off program is running.

Or calibrate without a distillation run.

Arrange a liquid supply to the pump inlet in a volume of more than 100 ml. It is recommended to use spirit, low wines or heads from previous runs.

Warning.

The number of steps needed to take off 1 ml is written to the samovarbackup.txt file when you [save settings](#), but it is not loaded when settings are restored. It is not overwritten either. This is intentional.

If you need to restore the previous value manually, find the "StepperStepMl" parameter in the samovarbackup.txt file; after the colon, in quotes, is the number of steps. Enter it in the "Number of stepper motor steps: ____ /ml." («Количество шагов шагового двигателя: ____ /мл.») field on the Settings («Настройка») -> Pump («Насос») tab (the first screenshot).

Tip.

If you have installed an encoder, you can adjust the stepper motor speed with the encoder during calibration, and when 100 ml is reached, pressing the encoder finishes the calibration.

Important.

It is highly advisable to lubricate the pump tube and rollers with neutral silicone grease from time to time. The grease affects not only the durability of the pump parts but also the accuracy of take-off.

Beer

The values on this tab are used in Beer mode. Exceptions: the "BK power after boiling starts" («Мощность БК после закипания») field belongs to [Wash column](#) mode, and the "Use boost heating element when boiling" («Использовать разгонный тэн при кипении») checkbox also works in Distillation mode.

Settings → Beer

- 1 Brewing order**
The brewhouse layout, not a recipe. "All-in-one / BIAB" - heating by the boiler sensor (mash), the stirrer works as set in the program. "HERMS" - PID by the water sensor (boiler), the pump continuously circulates the mash through the coil, the boiler sensor monitors the mash. "RIMS" - PID by the column section sensor (return line), the pump runs continuously; heating without flow on a real RIMS is dangerous.
- 2 PID regulator coefficients for heating**
Kp, Ki, Kd. Found by automatic calibration (the "Autotune" («Автотюнинг») program line) or manually.
- 3 Regulator voltage/power in temperature holding mode**
The maximum voltage (power) the regulator applies to the heater while holding a temperature rest.
- 4 Regulator voltage/power in heating and boiling mode**
The voltage (power) when heating up to the set temperature and on the "Boil" («Кипячение») line.
- 5 BK power after boiling starts**
The working power of Wash column mode after the boiler starts boiling.
- 6 Use boost heating element when boiling**
If cleared, the boost heating element is not switched on on the "Boil" line. Also works in Distillation mode.

PID regulator coefficients

You can run an automatic calibration of the PID regulator coefficients.
To do this, fill the boiler with water and run the following program:

Программа затирки:

Варочный порядок из настроек: All-in-one / VIAВ. ПИД по кубу (затор), мешалка как в универсальной программе.

№	Тип программы	Темп	Время	Мешалка	Датчик темп.	
01	Пауза	50	1	0^0^0^0^0	куб (затор)	+ -
02	Автотюнинг	50	0	0^0^0^0^0	куб (затор)	+ -

► Описание программы затирки:

► Описание программы затирки в текстовом формате для сохранения в облаке:

Сохранить программу

Загрузить программу

Установить программу

Очистить программу

Загрузить рецепт из XML

Program for PID automatic calibration: a "Pause" («Пауза») line and an "Autotune" («Автотюнинг») line with the same temperature

Samovar will heat up to the specified temperature and then start the automatic calibration. When the calibration is complete, Samovar will write the new coefficients to the settings, start using them immediately and move on to the next program line. You do not need to reboot Samovar. If the coefficients could not be saved, the message "PID autotune not saved" («PID autotune не сохранён») appears. The calibration process can take several hours. But the accuracy of temperature holding should improve. You can also adjust the PID regulator coefficients manually in the settings.

Warning. Without special knowledge, you should not change the PID regulator coefficients arbitrarily.

Regulator voltage in temperature holding mode

If you think the temperature rests are not being held correctly, you can change the regulator voltage in temperature holding mode.

Samovar holds the temperature during rests using PWM.

The value in the "Regulator voltage/power in temperature holding mode" («Напр./мощность регулятора в режиме поддержания темп.») field (for example, 100 volts) is the maximum voltage that can be applied to the heater in temperature holding mode.

That is, no more than 100 volts will be applied to the heater in temperature holding mode (a temperature rest). How often this voltage is applied is determined by Samovar's logic.

Practical examples:

If you have a large, uninsulated mash vessel in a cold room and a weak main heater, the mash temperature may drop while passing through the temperature rests.

In that case the following can happen: Samovar has to apply 100 volts to the heater constantly, but even so the mash temperature keeps falling.

When the temperature falls by more than the value set (in degrees C°) in the line:

```
#define HEAT_DELTA 1
```

of the Samovar_ini.h file, then to return to the set temperature Samovar starts applying, through the regulator, the voltage from the "Regulator voltage/power in heating and boiling mode" («Напр./мощность регулятора в режиме нагрева и кипения») field (heating) to the load.

To avoid this, increase the regulator voltage in temperature holding mode. For example, to 120 volts.

Or another example.

The temperature in the brewing vessel first exceeds (overshoots) the temperature rest set in the mash program line. This can happen if a heater that is too powerful for the vessel is used. In that case, leave the regulator voltage in temperature holding mode at 100, and increase the value of the line:

```
#define HEAT_DELTA 1
```

by a few units.

A description of the algorithms of the main and additional heaters can be found in the section [Stirrer, pump, temperature](#)

Regulator voltage in heating and boiling mode

You can set a reduced regulator voltage for when the "Boil" («Кипячение») program line is used. This is needed when brewing certain beer styles, or when a powerful heating element is used on a small brewing vessel.

Note.

The value set in this line also applies to heating up to the rest temperature (while the mash is colder than the setpoint by more than HEAT_DELTA), which lets you set, and quickly change, the main heater's power for the whole Beer mode.

BK power after boiling starts

The working power of the heater in [Wash column](#) mode after the boiler starts boiling. If the first line of the BK program sets its own power, that one is used. A value below the minimum that the firmware substitutes is not accepted.

Use boost heating element when boiling

Lets you disable the use of the boost heating element when the "Boil" program line is used. The boost heating element will still work during brewing with other program lines, according to the settings.

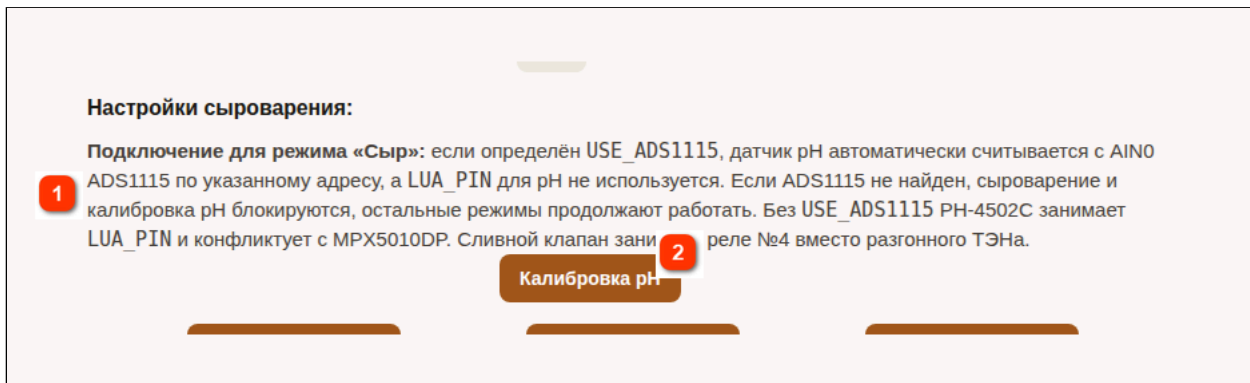
The checkbox also works in Distillation mode: if it is cleared, the boost heating element is switched off immediately when the boiler starts boiling.

Back to ["Beer" mode](#)

Cheese

Settings of the [Cheesemaking](#) mode.

The Settings («Настройки») → Cheese («Сыр») tab contains only a wiring hint and a link to the pH sensor calibration. The cheesemaking program itself is set on the main page of the mode (see [Cheesemaking program](#)).



Settings → Cheese («Настройки → Сыр»)

1

Wiring for the "Cheese" mode

A hint on which inputs and relays cheesemaking uses and what they conflict with.

2

pH calibration

Opens the page for calibrating the pH sensor with two buffer solutions. If the form has unsaved changes, Samovar asks for confirmation.

Wiring. The pH sensor is connected through an ADS1115 or to LUA_PIN. Schematics and pin assignments are in the technical section [pH sensor and how to connect it](#). The drain valve is connected to relay No. 4; do not connect a valve and a boost heating element to it at the same time.

Warning. When LUA_PIN is used, voltage matching is required; the PH-4502C and the MPX5010DP cannot be connected to this input at the same time. When an ADS1115 is used, the LUA_PIN input stays free.

The relay No. 4 trigger level (closing on a low or high level) is set on the [Other](#) («Прочие») tab, in the "Relay 4" («Реле 4») field. The drain valve follows this setting.

Cooling in cheesemaking uses the same outputs as Beer mode: a cooling pump (in the standard build) or a valve. There are no separate cooling settings on this tab.

pH calibration. The button opens the [pH calibration](#) page: using two buffer solutions (7.00 and 4.00 by default), Samovar calculates the coefficient k and the offset d and stores them with the "Save calibration" («Сохранить калибровку») button. A pH reading is considered valid in the range 0–14; if the sensor data has not been updated for 5 seconds, the value is marked as stale.

Note. The pH calibration coefficients are not part of the settings form and do not go into the samovarbackup.txt file (see [Saving settings](#)). After restoring settings from the file you do not need to repeat the pH calibration, but you also cannot move it to another controller with the file.

NBK

Settings of the [Continuous wash column \(NBK\)](#) mode.

The Settings → NBK («Настройки → НБК») tab holds the parameters of the continuous wash column automation: how fast the column responds to changes, in what steps the automation changes the power and the wash feed, and by what thresholds it detects flooding and completion. The default values suit most columns; change them after the first runs, when you can see how your column behaves.

Настройки НБК:

Инерция: 0

Дельта: 0

Шаг мощности: 0

Шаг подачи: 0

Т пара: 0

Давление захлёба: 0

Т завершения (барда): 0

8 ☐ **Переключать поток сервоприводом при захлёбе**

Settings → NBK («Настройки → НБК»)

- 1 Inertia («Инерция»)**
The time in seconds it takes the column to reach a new stable state after a change of power or feed. 180 by default. All automation pauses are counted from it (waiting after a correction, recovery after flooding). A value of 1 or less - the NBK start is rejected. Increase it for thick grain washes and for columns with preheating.
- 2 Delta («Дельта»)**
The permissible deviation of the spent wash (stillage) temperature from the reference T_n (°C) within which the power and the feed are not changed. 0.5 by default. A deviation of more than three deltas is treated as wash spillover - the feed is reduced by one step.
- 3 Power step («Шаг мощности»)**
By how many watts the automation changes the power in one correction. 100 by default. In the Optimization («Оптимизация») line the power grows by this step per iteration; in the Work («Работа») line after flooding the power is reduced by a tenth of the step.
- 4 Feed step («Шаг подачи»)**
By how many liters per hour the automation changes the wash feed in one correction. 0.5 by default. In the Optimization line - a full step, in the Work line - a tenth of the step, on spillover - a full step down. The feed manual adjustment buttons on the main page use the same step.
- 5 Vapor T («Т пара»)**
The reference vapor temperature, °C. 81 by default; allowed values are from 80 to 97. The boost (line H) is considered finished at a vapor temperature of 75°C. The hint in the interface calls this value the upper limit, but in the automation it is used as the lower bound of the working vapor temperature.
- 6 Flooding pressure («Давление захлёба»)**
The threshold on the boiler pressure sensor (mmHg) at which flooding is registered. 40 by default. The working pressure ceiling is half of this value (20 mm for 40 mm); during the run the automation itself lowers the ceiling after each flooding.
- 7 Completion T (stillage) («Т завершения (барда)»)**
The reference temperature of the stripped stillage, °C. 98.5 by default. On the first Optimization iteration it is refined upward automatically from the sensor reading (up to 102°C).
- 8 Switch the stream with the servo on flooding («Переключать поток сервоприводом при захлёбе»)**
On flooding the servo selects position 1 - the dirty stream. Position 0 - the clean stream - is restored only after the full recovery pause, on moving to the Work line and when the NBK finishes. Off by default.

Besides this tab, the NBK needs the mains voltage and the heating element resistance on the [Other \(«Прочие»\)](#) tab: Samovar uses them to calculate the maximum heating power. The flooding pressure threshold works only when a pressure sensor is connected to the boiler.

The NBK program (the lines Boost («Разгон»), Manual setup («Ручная настройка»), Optimization («Оптимизация»), Work («Работа»)) is set on the main page of the mode, see [Continuous wash column \(NBK\)](#).

Other

Settings («Настройки») → Other («Прочие»)

- 1 Blynk token («Токен Blynk»)**
The token from the email after registering on samovar-tool.ru. The log and messages go to the cloud by the same token.
- 2 Camera video URL («URL видео камеры»)**
The video stream address for the video widget in the Blynk app.
- 3 Time zone («Часовой пояс»)**
The offset from Greenwich, a whole number of hours.
- 4 Use regulator connection check («Использовать проверку подключения регулятора»)**
If the regulator does not answer requests, Samovar switches the heating off.
- 5 Heating element resistance for power calculation («Сопротивление тэна для расчета мощности»)**
Ohms. Needed to calculate power when working with a feedback regulator.
- 6 Mains voltage («Напряжение сети»)**
Volts. Together with the heating element resistance it sets the heater's maximum power (NBK mode).

- 7 Use water flow control («Использовать контроль потока воды»)**
Take the coolant flow sensor readings into account.
- 8 Colors for temperatures («Цвета для температур»)**
The color of each sensor on the main page, the "Temperature" («Температура») tab and the local chart.
- 9 Relay level settings («Настройки уровней для реле»)**
The active level for each of the four relays: LOW for Low Level Trigger modules, HIGH for High Level Trigger.
- 10 Download settings file («Скачать файл настроек»)**
Save all settings to your computer in the file samovarbackup.txt.
- 11 Reset WiFi settings («Сбросить настройки WiFi»)**
Forget the Wi-Fi network and reboot.
- 12 Upload settings file («Загрузить файл настроек»)**
Put the settings from a previously saved file into the form. To apply them, press "Save" («Сохранить»).
- 13 Reboot Samovar («Перезагрузить Самовар»)**
Reboot without turning the power off.

Samovar 7.00: the "Log data recording" («Запись данных в лог») field is no longer on the tab; the log is written at a fixed interval of 4 seconds. Messages and the log go to the samovar-tool.ru site through the samovar-tool.ru server, see [Logs and charts in the cloud](#).

Blynk token («Токен Blynk»)

The device token for the samovar-tool.ru server. It is shown by the Samovar mobile app after registration, see [Mobile apps and website](#). The token can also be entered in the [Wi-Fi connection](#) portal. The process log and messages reach the cloud by this same token.

Camera video URL («URL видео камеры»)

If you have a camera that can serve a video stream over http or rtp stream, you can specify the camera access URL. The video will be available in the Blynk app (with a limitation: the video is available only within the local network).

Time zone («Часовой пояс»)

Your time zone: the time offset from Greenwich, a whole number of hours.

Use regulator connection check («Использовать проверку подключения регулятора»)

Used with feedback regulators. Samovar periodically sends a request to the regulator. If there is no answer to several requests, Samovar shows the message "No connection with the regulator!" («Нет связи с регулятором!») and then stops the process with the message "Emergency shutdown! No connection with the heating regulator." («Аварийное отключение! Нет связи с регулятором нагрева.»)

Heating element resistance for power calculation («Сопротивление тэна для расчета мощности»)

The resistance of the heating element. Used to calculate power when working with a feedback regulator. The current power value is shown on the main page.

Mains voltage («Напряжение сети»)

The supply mains voltage. Together with the heating element resistance it sets the heater's maximum power; used in [NBK](#).

Use water flow control («Использовать контроль потока воды»)

Use of the coolant flow sensor. If the sensor is connected but the checkbox is cleared, its readings are not taken into account when monitoring the water supply.

Colors for temperatures («Цвета для температур»)

Setting the color for temperature sensors.

Setting the color for a particular sensor. The sensor readings will be shown in the selected color on:

the "Main" («Главная») tab, the "Temperature" («Температура») tab and the local charts; in the cloud, for now, the chart color settings do not change.

Relay level settings («Настройки уровней для реле»)

Relay modules are mostly of the Low Level Trigger type (LLT; enabled by default). This means a logical "0" must be applied to the relay control input (IN1) to switch the relay on. When a logical "1" is applied to the input (IN1), the relay switches off.

Select the level at which a particular relay module switches on. For an ordinary Low Level Trigger module select LOW. After boot Samovar sets the opposite, off level; the selected active level is applied only when the command to switch the relay on is given.

Buttons («Кнопки»)

Download settings file («Скачать файл настроек»)

The settings are saved to your computer as the text file samovarbackup.txt. The file can be renamed (for example, to the name of the mode).

Reset WiFi settings («Сбросить настройки WiFi»)

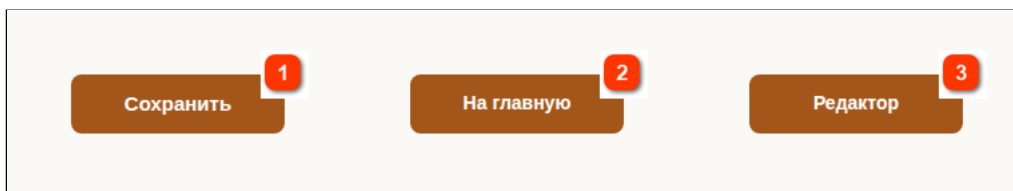
Forget the Wi-Fi network Samovar is currently connected to. After confirmation Samovar reboots and raises an access point for [Wi-Fi connection](#).

Upload settings file («Загрузить файл настроек»)

Restore settings from a previously saved file. The values are put into the fields of this page, after which the hint "Settings loaded into the form, press "Save" to apply." («Настройки загружены в форму, нажмите «Сохранить», чтобы применить.») appears. Fields with invalid values are listed in the error message.

Reboot Samovar («Перезагрузить Самовар»)

Rebooting Samovar without turning the power off.



Buttons at the bottom of the settings page

- 1 Save («Сохранить»)**
Save the settings of all tabs of the page.
- 2 To main page («На главную»)**
Return to the main page.
- 3 Editor («Редактор»)**
Open the Samovar file editor.

Save («Сохранить»)

Save the settings on this page. After saving, the main page opens. If a field value is outside the valid range, the message "Settings not saved. Check the fields: ..." («Настройки не сохранены. Проверьте поля: ...») appears. The mode cannot be changed while heating is on: Samovar refuses to save.

To main page («На главную»)

Return to the main page. If there are unsaved changes, Samovar asks for confirmation.

Editor («Редактор»)

Switch to the Samovar file editor (the /edit page). Here you can delete the old log data_old.csv or other files and view the contents of the file system.

Saving settings

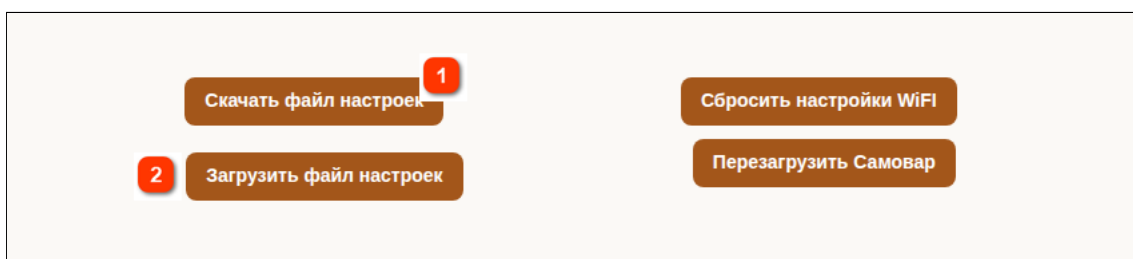
Samovar 7.00: the "Download settings file" («Скачать файл настроек») and "Upload settings file" («Загрузить файл настроек») buttons are in Settings («Настройки») → Other («Прочие»). The file contains all fields of all settings tabs, so the table below is an example of the old set of fields and does not include the new parameters. After the file is loaded, the values are only filled into the form: to apply them, press "Save" («Сохранить»).

Samovar can save all settings to a text file and later restore them. This is convenient if you use several sets of temperature sensors, so you do not have to enter their addresses every time. After you press "Download settings file", the browser creates the file *samovarbackup.txt*, which is saved to your browser's downloads folder.

The saved file can be renamed (raw material, mode and so on) and loaded as needed.

To restore, press "Upload settings file" and select the file. Samovar checks that it is a settings file in JSON format, fills the values into the form fields and shows the message "Settings loaded into the form, press Save to apply" («Настройки загружены в форму, нажмите «Сохранить», чтобы применить»). If the file contains values outside the allowed range, the message lists the faulty fields; fix them before saving. The loaded values take effect only after you press "Save".

This file also contains information that you can copy and paste into Samovar without loading all the previous settings. For example, the Blynk token or the video camera URL.



Settings → Other: saving and loading the settings file («Настройки → Прочие: сохранение и загрузка файла настроек»)

1 Download settings file («Скачать файл настроек»)
Creates the *samovarbackup.txt* file with all fields of the settings form and saves it to the browser's downloads folder.

2 Upload settings file («Загрузить файл настроек»)
Opens a file picker. The values from the file are filled into the form; to apply them, press "Save" («Сохранить»). The StepperStepMI field is not loaded from the file.

Warning.

In some cases, when a new sensor is added, the sensors may change the mounting place assigned to them earlier. This is related to the sensor identifier and to the logic Samovar uses to arrange the sensors.

The *samovarbackup.txt* file looks like this:

```
{ "mode": "0", "DistTemp": "98.000", "usepressure": true, "useautospeed": true, "autospeed": "2", "useautopowerdown": false, "UseBuzzer": true, "UseBBuzzer": true, "ChangeProgramBuzzer": false, "SteamAddr": "0", "DeltaSteamTemp": "-2.000", "SetSteamTemp": "0.300", "SteamDelay": "20", "PipeAddr": "2", "DeltaPipeTemp": "0.000", "SetPipeTemp": "2.000", "PipeDelay": "20", "WaterAddr": "3", "DeltaWaterTemp": "0.000", "SetWaterTemp": "0.000", "WaterDelay": "20", "TankAddr": "4", "DeltaTankTemp": "0.000", "SetTankTemp": "0.000", "TankDelay": "20", "ACPAddr": "1", "DeltaACPTemp": "0.000", "SetACPTemp": "0.000", "ACPDelay": "20", "StepperStepMI": "8954", "Kp": "150.000", "Ki": "1.400", "Kd": "1.400", "StbVoltage": "100.00", "blynkauth": "9u0XzN*****R1e7xHYp", "videourl": "", "TimeZone": "3", "CheckPower": false, "HeaterR": "35.000", "LogPeriod": "3", "SteamColor": "#2c03fc", "PipeColor": "#f90126", "WaterColor": "#08c7f7", "TankC
```

The parameters are separated by commas, and each parameter and its value are enclosed in quotes and joined by a colon.

For example: "DistTemp": "98.000",

Keep in mind when loading a saved settings file into Samovar: all values from the file, except StepperStepMI, replace the current ones. If you changed settings between saving and loading (for example, calibrated an external I2C pump, the StepperStepMI2C field), the old values will be filled into the form after loading. To avoid this, delete such parameters together with their values from the file.

Warning.

*Starting with version 5.20, the "Number of motor steps needed to take off 1 ml." («Количество шагов двигателя необходимых для отбора 1 мл.») value (StepperStepMI) is present in the *samovarbackup.txt* file, but it is not loaded into Samovar when settings are restored. The value that exists at the time of restoring settings is not overwritten. You can restore it manually.*

The table lists the parameter names with explanations. The values are taken, as an example, from the *samovarbackup.txt* file shown above.

"mode"	0	Operating mode
DistTemp	98.000	Rectification and distillation up to boiler temperature («Ректификация и дистилляция до температуры в кубе»)
usepressure	true	Use pressure correction («Использовать корректировку давления»)

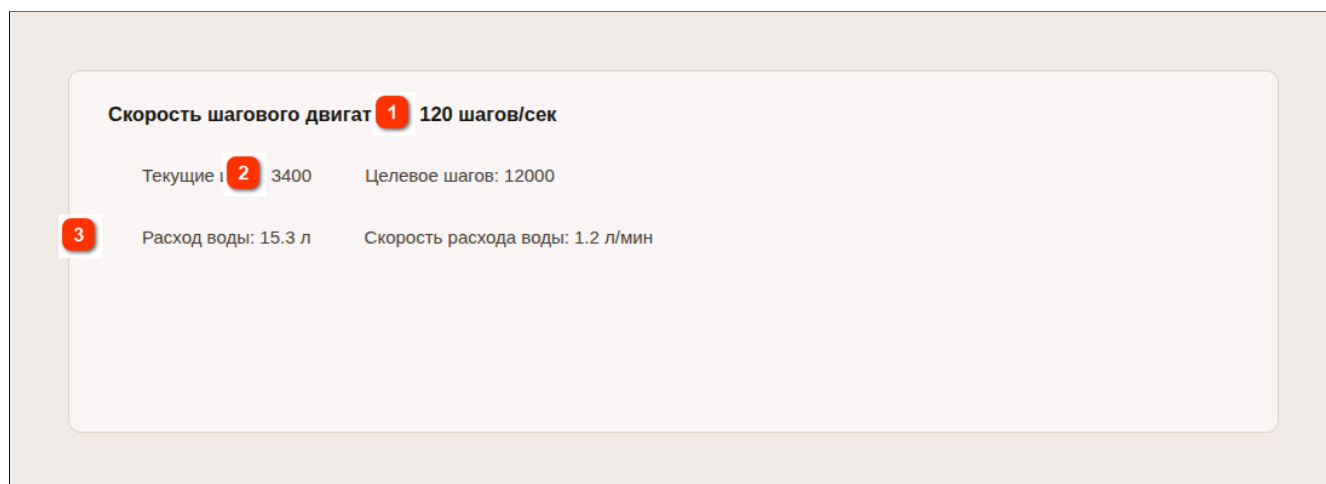
useautospeed	true	Use pump speed correction («Использовать корректировку скорости насоса»)
autospeed	2	Correction percentage («Процент корректировки»)
useautopowerdown	false	Use applied power correction («Использовать корректировку подаваемой мощности»)
UseBuzzer	true	Use the buzzer for notification («Использовать зуммер для нотификации»)
UseBBuzzer	true	Use the buzzer for notification in the browser («Использовать зуммер для нотификации в браузере»)
ChangeProgramBuzzer	false	Sound the buzzer when the program changes («Включать зуммер при смене программы»)
SteamAddr	0	Steam sensor address before the dephlegmator («Адрес датчика пара перед дефлегматором»)
DeltaSteamTemp	-2.000	Steam sensor correction before the dephlegmator («Корректировка датчика пара перед дефлегматором»)
SetSteamTemp	0.300	Steam sensor setpoint before the dephlegmator («Уставка датчика пара перед дефлегматором»)
SteamDelay	20	Take-off delay when steam T goes beyond the setpoint (sec) («Задержка отбора при выходе Т пара за уставку (сек)»)
PipeAddr	2	Column section sensor address («Адрес датчика в царге»)
DeltaPipeTemp	0.000	Sensor correction («Корректировка датчика»)
SetPipeTemp	2.000	Sensor setpoint («Уставка датчика»)
PipeDelay	20	Take-off delay when T goes beyond the setpoint (sec) («Задержка отбора при выходе Т за уставку (сек)»)
WaterAddr	3	Water sensor address («Адрес датчика воды»)
DeltaWaterTemp	0.000	Sensor correction («Корректировка датчика»)
SetWaterTemp	0.000	Sensor setpoint («Уставка датчика»)
WaterDelay	20	Take-off delay when T goes beyond the setpoint (sec) («Задержка отбора при выходе Т за уставку (сек)»)
TankAddr	4	Boiler sensor address («Адрес датчика в кубе»)
DeltaTankTemp	0.000	Sensor correction («Корректировка датчика»)
SetTankTemp	0.000	Sensor setpoint («Уставка датчика»)
TankDelay	20	Take-off delay when T goes beyond the setpoint (sec) («Задержка отбора при выходе Т за уставку (сек)»)
ACPAddr	1	TCA sensor address («Адрес датчика ТСА»)
DeltaACPTemp	0.000	Sensor correction («Корректировка датчика»)
SetACPTemp	0.000	Sensor setpoint («Уставка датчика»)
ACPDelay	20	Take-off delay when T goes beyond the setpoint (sec) («Задержка отбора при выходе Т за уставку (сек)»)
StepperStepMl	8954	Number of motor steps needed to take off 1 ml. («Кол-во шагов двигателя необходимых для отбора 1 мл.»)
Kp	150.000	Coefficient for the heating PID regulator («Коэффициент для PID регулятора нагрева»)
Ki	1.400	Coefficient for the heating PID regulator («Коэффициент для PID регулятора нагрева»)
Kd	1.400	Coefficient for the heating PID regulator («Коэффициент для PID регулятора нагрева»)
StbVoltage	100.00	Regulator voltage in temperature holding mode (V) («Напр. регулятора в режиме поддержания темп (V)»)
blynkauth	9u0Xz....7xHYp	Blynk token («Токен Blynk»)
videourl		Video camera URL («URL видеокамеры»)
TimeZone	3	Time zone («Временная зона»)
CheckPower	false	Use the regulator connection check («Использовать проверку подключения регулятора»)
HeaterR	35.000	Heating element resistance for power calculation («Сопротивление тэна для расчета мощности»)
LogPeriod	3	Interval of writing data to the log (sec) («Периодичность записи данных в лог (сек)»)
SteamColor	#2c03fc	Steam sensor temperature color («Цвет температуры датчика пара»)
PipeColor	#f90126	Column section sensor temperature color («Цвет температуры датчика царги»)
WaterColor	#08c7f7	Water sensor temperature color («Цвет температуры датчика воды»)

TankColor	#338a05	Boiler sensor temperature color («Цвет температуры датчика куба»)
ACPColor	#000000	TCA sensor temperature color («Цвет температуры датчика TCA»)
rele1	1	Relay 1 level («Уровень реле 1»)
rele2	0	Relay 2 level («Уровень реле 2»)
rele3	0	Relay 3 level («Уровень реле 3»)
rele4	0	Relay 4 level («Уровень реле 4»)

Note.

If there is no saved profile, Samovar 7.00 creates the standard settings itself. The Samovar_default_settings.txt file from the tools folder of the firmware sources can be loaded as an ordinary settings file, but after the first flashing this is not required.

"Additional" («Дополнительно») tab



"Additional" («Дополнительно») tab in Rectification mode

- 1 Stepper motor speed**
Steps per second while running the current program line.
- 2 Steps**
Current steps is how many steps have already been made in the current line; Target steps is how many must be made before it ends.
- 3 Water flow**
Total liters and rate in l/min from the flow sensor. The block is visible only when the sensor is connected.

Samovar 7.00: the "Speed, l/h" («Скорость, л/ч») field with the "Set" («Установить») button, the "Lock hearts take-off T" («Зафиксировать T отбора тела») button and the system parameters line have been moved to the "Rectification" («Ректификация») tab (the main one). Their description is kept on this page.

Speed, l/h ("Rectification" tab)

Manual entry of the take-off rate for the peristaltic pump. It has priority over the program. It applies provided the program is already running and after the "Set" button is pressed. Take-off at the set rate continues until the next line starts.

Stepper motor speed

How many steps per second the stepper motor makes while running the program currently being executed.

Steps

Current steps is the number of steps already made in the program currently being executed.

Target steps is how many steps are left to make before the currently running program ends.

Water flow

Coolant flow data. Total (liters) and rate in l/min. Shown only if a flow sensor is connected.

Note.

Keep in mind that the flow sensor reads with a very large error, which depends strongly on the flow rate. So, by and large, it can only be used to check that water is present and for a very rough estimate of the flow. It can be calibrated in the Samovar_ini.h file, line #define WF_CALIBRATION 98.

System parameters ("Rectification" tab)

Shown on the main page in the "System" («Система») line. Information about the state of the cooling pump has been added.

Heap - the amount of free dynamic memory in bytes

BME temp - the temperature on the pressure sensor chip. In effect, the temperature inside the enclosure with the ESP32 installed

RSSI - an indicator of the power level of the received WiFi signal.

Free («Свободно») - free space in the Samovar file system in bytes. The process log is written here.

Water pump PWM («ШИМ насоса воды») - the current PWM duty of the cooling pump, if Samovar controls the pump.

Note.

The RSSI value is measured in dBm. The minimum value acceptable for network operation: RSSI = -85 dBm. Important! The higher the RSSI number, or the less negative it is, the stronger the signal (for example, -48 is greater than -78).

Lock hearts take-off T ("Rectification" tab)

Set new Steam and Column section temperatures.

Explanation.

Samovar records the Steam and Column section temperatures at the moment heads take-off ends and the "Hearts" («Тело») or "Pre-flooding" («Предзахлеб») program starts. At this moment power is usually increased to improve separation and late heads are taken off. After the power is raised, the temperatures may go outside the setpoints specified on the [Temperature](#) tab. Samovar will then pause the take-off and wait for the temperatures to return to the recorded values. Pressing the button makes the current Column section and Steam temperatures the main ones, and take-off resumes accordingly. Samovar confirms this with the message "New hearts T: steam = ..., column section = ..." («Новые Т тела: пар = ..., ципра = ...»).

System Messages

During operation, Samovar's warnings are shown in the WEB interface.

Messages appear in a panel in the top right corner on any page of the interface. Each message can be hidden with the cross; the "Hide all" («Скрыть все») button hides the whole panel. Before hiding an emergency message, Samovar asks for confirmation. The ≡ button in the header opens the message history; the history is stored in the browser, and the "Clear history" («Очистить историю») button deletes it.

The same messages go to the [mobile apps and the site](#) (push notifications and the "Messages" tab) and to the [cloud](#), where they can be viewed on the log's "Messages" page.

Samovar 7.00: a pause or wait may be recorded as two service messages: a start and an end. In the charts of the local interface and the site, and on the "Control" («Управление») page, the pair is decoded into a reason, an outcome and a duration. In the ordinary message list the service header @P1 may be visible. If only one part arrived, it is an incomplete event without a duration. If the connection drops or Samovar reboots, part of the history may be missing.

The screenshot displays the main control interface of the Samovar 7.00 system. At the top, there are tabs for 'Ректификация', 'Программа', and 'Дополнительно'. Below them, a status bar shows 'Самовар v. 7.00' and various operational parameters like 'Расчёт', 'I2CStepper', 'Внешний насос', and 'График'. A green dot indicates '12:34:56 Работа 01:1 3'. A message panel is overlaid on the right side, containing three messages: '18:02:51 Переход к строке программы №2' (green), '18:02:51 Пауза по Т царги' (yellow), and '18:02:51 Сработал датчик захлёба!' (red). A 'Скрыть все' button is at the bottom of the panel. The main interface shows a central diagram of the distillation process with various temperature and pressure readings. On the left, there are buttons for 'Выключить нагрев', 'Поставить на паузу', 'Далее', 'Сбросить процесс', 'График', and 'Настройки'. Below these are sensor readings for 'Пар' (78.2°C), 'Царга' (77.9°C), 'Δ пар – царга' (-0.3°C), 'Куб' (84.3°C), 'Вода' (22.4°C), and 'ТСА' (40.1°C). At the bottom left, a 'ДЕТЕКТОР ПРИМЕСЕЙ' section shows 'Состояние: Стабильно, слежу' and 'Тренд: 0.000 °C/мин'. On the right, there are sections for 'РЕГУЛЯТОР' (current voltage 180.5 V, target 180.0 V, power 1450 W) and 'ДАВЛЕНИЕ' (atmospheric 752.3 mm Hg, column 1.2 mm Hg). A 'СИСТЕМА' section at the bottom right shows system parameters like 'Неар=152000', 'BME temp=24.5', 'RSSI = -58', and 'Свободно: 300000'.

The message panel on the main page

- 1 Messages («Сообщения»)**
The message list; the background color depends on severity. The cross on the right hides one message.
- 2 Hide all («Скрыть все»)**
Hides the panel. It reappears when new messages arrive.



Message history («История сообщений»)

Opens the list of all messages received by the browser, with the "Clear history" («Очистить историю») and "Close" («Закрыть») buttons.

Messages

Depending on severity, messages have different background colors: emergency, warning, notification. In the mobile apps, emergency messages arrive with the prefix "Alarm!", and warnings with "Warning!".

Some warnings do not require immediate action:

For example: "Open the water supply!" («Откройте подачу воды!»)

It appears as a warning when the boiler temperature set in the sketch is reached, regardless of whether the cooling tap is open or closed.

If the warning repeats, check whether the tap is really open and whether the flow is sufficient. If a pump is used to hold the temperature, check whether the pump is connected. Check the connections of the water flow sensor.

Some warnings, on the contrary, require the operator's immediate intervention.

For example: "Critical water temperature!" («Критическая температура воды!»)

It means that the coolant at the outlet is approaching the limit value set in the [sketch](#) (70° by default; the message appears 5° before it). If the operator does not react, Samovar reduces the heating power, and if the temperature keeps rising to 75°, it turns heating off.

Alarm sound

On an emergency heating lockout and when the connection is lost, an alarm sounds in the browser. If the browser has blocked playback, the hint "The browser has blocked the alarm sound. Click the page to allow sound." («Браузер заблокировал звук тревоги. Нажмите на страницу, чтобы разрешить звук») appears.

Below is the list of Samovar messages:

Flooding sensor triggered! («Сработал датчик захлёба!»)

- Each time the reflux level sensor triggers and the event is actually counted. Samovar counts only those trigger periods that last 3 or more seconds (WHLS_ALARM_TIME).

Reducing voltage (power) from XX.X («Напряжение (Мощность) снижаем с XX,X»)

- After the flooding sensor triggers, the voltage is reduced from the current one.

- When the column section or steam setpoint is exceeded with the ["Use supplied power correction"](#) («Использовать коррективку подаваемой мощности») option enabled.

Critical water temperature! («Критическая температура воды!»)

- The water temperature has come within 5° of the value in the line [#define ALARM_WATER_TEMP 70](#). The buzzer turns on and take-off is paused for 30 seconds.

Critical water temperature! Water supply error. Reducing voltage from XXX («Критическая температура воды! Ошибка подачи воды. Напряжение снижаем с XXX»)

- If there was no reaction to the previous message, the voltage is reduced by 5 volts to prevent the column from overheating. In the Distillation and Wash Column modes the text is "Critical water temperature! Reducing ... from XXX" («Критическая температура воды! Понижаем ... с XXX»).

Emergency shutdown! Maximum Water (Steam, TCA) temperature exceeded («Аварийное отключение! Превышена максимальная температура Воды (Пара, TCA)»)

- The water temperature has exceeded the value in the line [#define MAX_WATER_TEMP 75](#), steam - MAX_STEAM_TEMP 98.8, TCA - MAX_ACP_TEMP 75. Heating is off, the water valve is closed.

Emergency shutdown! Water supply stopped. («Аварийное отключение! Прекращена подача воды.»)

- If a flow sensor is used but no data comes from it.

Distiller heating turned on («Включен нагрев дистиллятора»)

- Distillation. The "Turn on heating" («Включить нагрев») command was given.

Pump control is not supported by your hardware («Управление насосом не поддерживается вашим оборудованием»)

- Wash column. PWM control of the cooling pump is not enabled in the firmware.

Wash column heating turned on («Включен нагрев бражной колонны»)

- Wash column. The "Turn on heating" command was given.

Wash column operation finished («Работа бражной колонны завершена»)

- Wash column. Heating is turned off when the set [maximum boiler temperature](#) is reached.

- The "Turn off heating" («Выключить нагрев») command was given.

Mashing program finished («Программа затираания завершена»)

- Beer mode. All lines of the [beer brewing program](#) have been executed. Heating is off.

Cooling water valve opened! («Открыт клапан воды охлаждения!»)

- Beer mode. The start of the "Cooling" («Охлаждение») line.

- Beer mode. The "Fermentation" («Брожение») line. When the mash temperature exceeds the boiler setpoint by the setpoint amount.

Cooling water valve closed! («Закрыт клапан воды охлаждения!»)

- Beer mode. The end of the "Cooling" line.

- Beer mode. The "Fermentation" line. The mash temperature has returned to the set limits.

Malt dough-in temperature reached! («Достигнута температура засыпи солода!»)

- Beer mode. The malt dough-in temperature has been reached. Waiting.

Temperature rest of XX° reached. Waiting N minutes. («Достигнута температурная пауза XX°. Ждем N минут.»)

- Beer mode. The time count for the "Pause" («Пауза») line has started.

Hold paused: temperature below target («Пауза выдержки: температура ниже цели»)

- Beer mode. The "Pause" line. The temperature has dropped 3° or more below the set one, and the time count is stopped. Heating works. With a smaller undershoot there is no message, and the count runs slower.

Hold suspended: temperature out of band («Выдержка приостановлена: температура вне полосы»)

- Cheesemaking mode. The "Hold" («Выдержка») stage. An undershoot of 3° or more or an overshoot above the tolerance; the hold time is not counted.

Hold resumed («Выдержка продолжена»)

- Beer and Cheesemaking modes. The temperature has recovered and the hold time count continues.

Add the hops! («Засыпьте хмель!»)

- Beer mode. 30 seconds before the end of the "Boil" («Кипячение») program line.

Transition to program line No. («Переход к строке программы №»)

- A transition to the next program line. In Beer mode it is supplemented with "Heating to the malt dough-in temperature: XX°" («Нагрев до температуры засыпи солода: XX°»).

No alcohol left in the boiler («В кубе не осталось спирта»)

- Distillation, Wash column. Condition: the boiler temperature is above 90 degrees and heating is on. If, over the time set in the settings by the "Time for monitoring the end of the distillation process" («Время для контроля завершения процесса дистилляции») field (60 minutes by default, 0 - do not check), the boiler temperature changes by no more than 0.1 degree.

Distillation finished. Total time: N min. («Дистилляция завершена. Общее время: N мин.»)

- Heating is turned off when the set [maximum boiler temperature](#) is reached.

- Heating is turned off.

- No alcohol is left in the boiler.

Pause on column section (or steam) T («Пауза по T царги (или пара)»)

- The column section (or steam) sensor temperature has gone beyond the [setpoint](#).

Program execution finished. («Выполнение программы завершено.»)

- The whole program held in Samovar's memory has been executed.

- Heating is turned off.

Maximum boiler temperature limit. Program finished. («Лимит максимальной температуры куба. Программа завершена.»)

- Rectification. Heating is turned off when the set [maximum boiler temperature](#) is reached.

Boost finished. Stabilization/total reflux. («Разгон завершён. Стабилизация/работа на себя.»)

- The steam sensor temperature value in the line #define CHANGE_POWER_MODE_STEAM_TEMP 39 has been reached.

Stabilization finished, the column is working stably. («Стабилизация завершена, колонна работает стабильно.»)

- Condition: heating is on. The steam sensor has passed the boost cut-off mark. For six minutes the steam temperature changes by no more than 0.1 degree.

Open the water supply! («Откройте подачу воды!»)

- Rectification. The boiler temperature given in the line #define OPEN_VALVE_TANK_TEMP 77 has been reached.

- Beer. The start of the "Cooling" line.

- Beer. The "Fermentation" line. When the mash temperature exceeds the boiler setpoint by the setpoint amount.

Close the water supply! («Закройте подачу воды!»)

- Rectification, Distillation, Wash column. Heating is off and the water temperature has dropped below the [water T setpoint](#) by the amount set in the sketch.

Boiling has started in the boiler! («Началось кипение в кубе!»)

- Condition: heating is on; the steam temperature has risen above 39°, or the boiler temperature has stopped rising while the cooling water has warmed up. In Distillation mode the calculated alcohol content in the boiler is added to the message.

No connection with the regulator! («Нет связи с регулятором!»)

- The [regulator connection check](#) is enabled and the regulator has not answered several requests in a row. If the connection is not restored - "Emergency shutdown! No connection with the heating regulator." («Аварийное отключение! Нет связи с регулятором нагрева.»)

Emergency shutdown! Heater control error. («Аварийное отключение! Ошибка управления нагревателем.»)

- The actual power reported by the regulator differs from the set one by more than 20% several times in a row.

Steam (column section, water, boiler, TCA) temperature sensor error! («Ошибка датчика температуры пара (царги, воды, куба, в ТСА)!»)

- There is no data from the temperature sensor.

Running out of memory! Total: xxxxxx, used: xxxxxx («Заканчивается память! Всего: xxxxxx, использовано: xxxxxx»)

- There is no space left in the file system for the log. Samovar deletes the old log data_old.csv itself when there is less than 16 KB of free space. Delete unneeded files through the [editor](#).

Emergency shutdown: the emergency button was pressed («Аварийное отключение: нажата аварийная кнопка»)

- The [emergency button was pressed or an emergency sensor triggered](#).

Heating is blocked by emergency protection, a reboot is required («Нагрев заблокирован аварийной защитой, требуется перезагрузка»)

- After an emergency shutdown, heating will not turn on until Samovar is rebooted. While the lockout is active, an alarm sounds in the browser.

An unfinished session was found, mode N, program No. M; heating was NOT resumed automatically. («Обнаружена незавершённая сессия, режим N, программа №M; нагрев НЕ возобновлён автоматически.»)

- Samovar rebooted during a process. Heating must be turned on manually; if less than 30 minutes have passed since the reboot, the log in the cloud continues the previous session.

Connection lost! («Обрыв связи!»)

- A browser message: Samovar did not answer three polls in a row. The page is dimmed and blocked until the connection is restored, and the clock line shows the time of the last data received.

Configuration loaded with an error: ... («Конфигурация загружена с ошибкой: ...»)

- The settings were read with an error at startup, and Samovar works on default values. The reason is shown on the main page in the strip under the readings.

Logs and charts in the cloud

Viewing information in the cloud.

For Samovar to send the log to the cloud, get a device token and enter it in the "Blynk token" («Токен Blynk») field on the tab [Settings → Other \(«Настройки → Прочие»\)](#). To view this log in your profile, add the device on the website www.samovar-tool.ru in the menu "User → Devices".

While the token is set and there is a connection to the server, when writing the local log Samovar sends data and messages to the cloud, where they can be viewed on the website www.samovar-tool.ru on the "Logs" page (samovar-tool.ru/logi/).

One website user can have several Samovars if the website administrator has raised the device limit (one by default, see [Mobile apps and website](#)). The "Logs" page shows the processes of the device selected in the menu "User → Devices". To open the archive of another Samovar, select it there first.

Samovar 7.00: there are no separate settings for sending the cloud log any more. While the token is set and there is an internet connection, log lines are sent automatically once every 4 seconds. If Samovar rebooted during a process and heating is switched on again within 30 minutes, recording continues in the same session.

On the chart, new pauses are shown as an interval with the reason, result and duration. If part of the data was not received or the device time was changed, a single mark may remain instead of an interval. In old logs such intervals do not appear after the website update.

The list of logs in the cloud looks as follows:

Список логов					
Дата логa	Детализация	График	Скачать лог	Сообщения	Редактировать
16.09.2022 10:20:59	Детализация	График	Сохранить лог	Сообщения	Удалить Объединить лог Расшарить лог
22.09.2022 09:09:22	Детализация	График	Сохранить лог	Сообщения	Удалить Объединить лог Расшарить лог
23.09.2022 08:04:33	Детализация	График	Сохранить лог	Сообщения	Удалить Объединить лог Расшарить лог
23.09.2022 12:55:09	Детализация	График	Сохранить лог	Сообщения	Удалить Объединить лог Расшарить лог
24.09.2022 09:11:44	Детализация	График	Сохранить лог	Сообщения	Удалить Объединить лог Расшарить лог
26.09.2022 09:40:13	Детализация	График	Сохранить лог	Сообщения	Удалить Объединить лог Расшарить лог
05.10.2022 07:50:03	Детализация	График	Сохранить лог	Сообщения	Удалить Объединить лог Расшарить лог
05.10.2022 12:39:39	Детализация	График	Сохранить лог	Сообщения	Удалить Объединить лог Расшарить лог

[Details](#) - the take-off program and additional information ("process memo")

Chart - a chart of the process will be built.

Save log - the log file will be downloaded to your computer. The file name is data.csv

Messages - all messages that appeared during this process.

Delete - delete this log. Cannot be undone.

Merge log - merge the log with the previous one. Cannot be undone.

Share log - allow everyone who has the link to this log to view it. Can be undone.

Logs and charts are available only to you, but you can share them with other participants.

To share a log:

Press "Share log". Answer yes to the question "Make the log available to everyone?".

The "Share log" label will change to "Make log private".

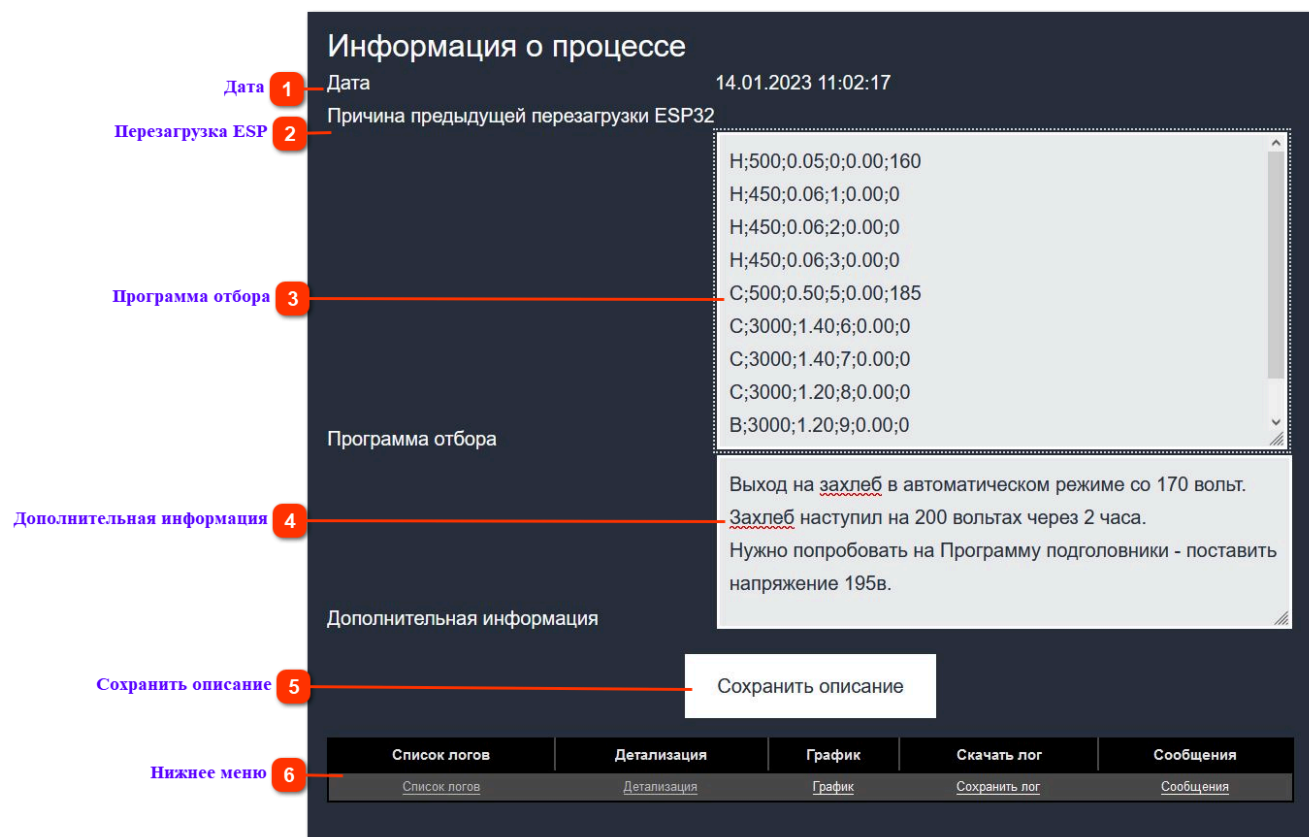
Open the log, for example on "Details", copy the link from the browser address bar and send it in any way to the person you want to share with.

Everyone who has the link will be able to view the chart and download the log file - data.csv

To forbid viewing: press "Make log private"; the label will change to "Share log". The log will no longer be available to others.

Details

Clicking "Details" opens a new window.



1. Date

The date and time the process started. The start time is recorded by the [condition](#) for starting the local log recording.

2. Reason for the previous ESP32 reboot

The reason Samovar stopped working may be shown.

3. Take-off program

The take-off program in text format. It can be copied and used in another process by pasting it into the text field on the "Program" («Программа») tab. Not editable.

4. Additional information

A description of the take-off program in text format. Free text. Can be edited.

5. Save description

Press after editing the take-off program description to save the changes.

6. Bottom menu

Log list - return to the previous window with the list of all logs

Details - this very window

Chart - view the chart for this particular process

Messages - all messages that appeared during this process.

Chart

In addition to the local one, the following data are added to the chart in the cloud:

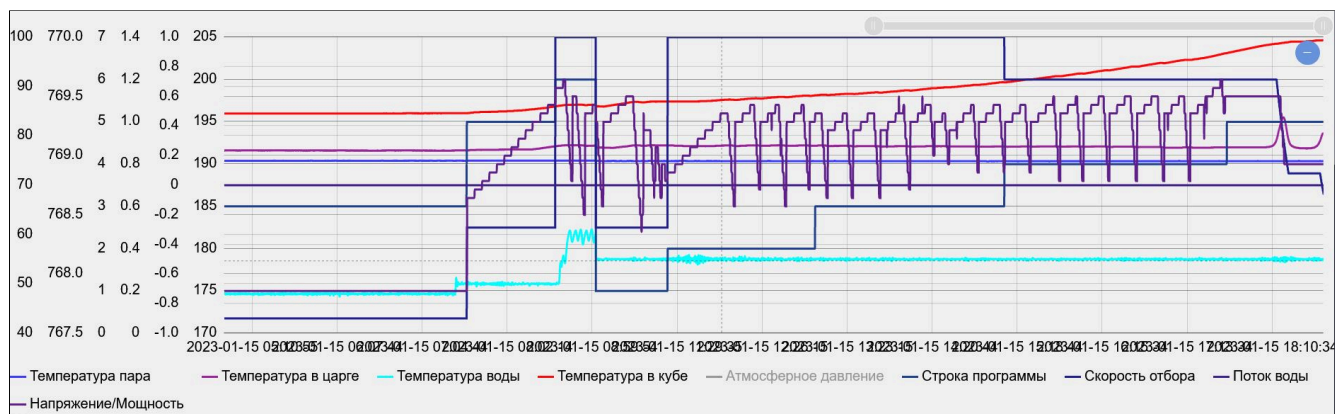
- Take-off rate
- Water flow
- Voltage/Power.

- ABV in the boiler
- ABV in the take-off (the data are relevant only for "Distillation" («Дистилляция») mode)

All conditions for starting the plot and controlling the view are the same as for the [local one](#). The data are stored in the cloud, and the chart image is built by the browser on your computer.

The less data, the faster your internet connection and the more powerful your computer, the faster the chart is built.

Example chart:



Thanks to the additional data on the chart, the example shows how the voltage regulator worked in pre-flooding mode.

Messages

If you missed a message, you can see the history here. For example: why the heating "suddenly" switched off.

Example:

A fragment of the messages window: heads take-off and the column reaching the pre-flooding mode.

Список сообщений		
Дата сообщения	Сообщение	Критичность сообщения
14.01.2023 11:02:18	Программа: старт строки №1; отбор в ёмкость 0	2
14.01.2023 21:02:32	Программа: старт строки №2; отбор в ёмкость 1	2
15.01.2023 03:28:12	Программа: старт строки №3; отбор в ёмкость 2	2
15.01.2023 07:34:41	Программа: старт строки №4; отбор в ёмкость 3	2
15.01.2023 07:34:54	Новые Т тела: пар = 75.00; чарга = 77.06	1
15.01.2023 07:34:54	Программа: старт строки №5; отбор в ёмкость 5	2
15.01.2023 08:35:14	Программа: старт строки №6; отбор в ёмкость 6	2
15.01.2023 08:41:50	Напряжение снижаем с 200.00	0
15.01.2023 08:42:35	Напряжение снижаем с 198.00	0
15.01.2023 08:43:21	Напряжение снижаем с 196.00	0
15.01.2023 08:44:06	Напряжение снижаем с 194.00	0
15.01.2023 08:44:51	Напряжение снижаем с 192.00	0
15.01.2023 08:45:37	Напряжение снижаем с 190.00	0
15.01.2023 08:46:00	Напряжение снижаем с 188.00	0

Questions and answers. FAQ

How long are logs kept in the cloud?

Logs shorter than 30 minutes are deleted one day after creation. The rest are kept indefinitely.

Does a brew in "Beer" («Пиво») mode appear in the website cabinet journal by itself?

Yes. A session started in "Beer" mode creates an entry in the journal of the "Brewer's cabinet" on the website automatically, from the same data that reach this page (mashing steps, boiling with the "Add the hops!" («Засыпьте хмель!») messages, cooling); the device sends nothing additional. More details are on the page [Beer brewing journal on the website and in the apps.](#)

Local logs and charts

Local viewing

Samovar keeps a temperature log and stores it in its own memory. For each mode, logging starts on a certain condition:

Rectification - when the "Start take-off" («Начать отбор») command is given.

Distillation - on the "Turn heating on" («Включить нагрев») command.

Beer - when the first line of the brewing program starts.

Wash column (BK) - on the "Turn heating on" command.

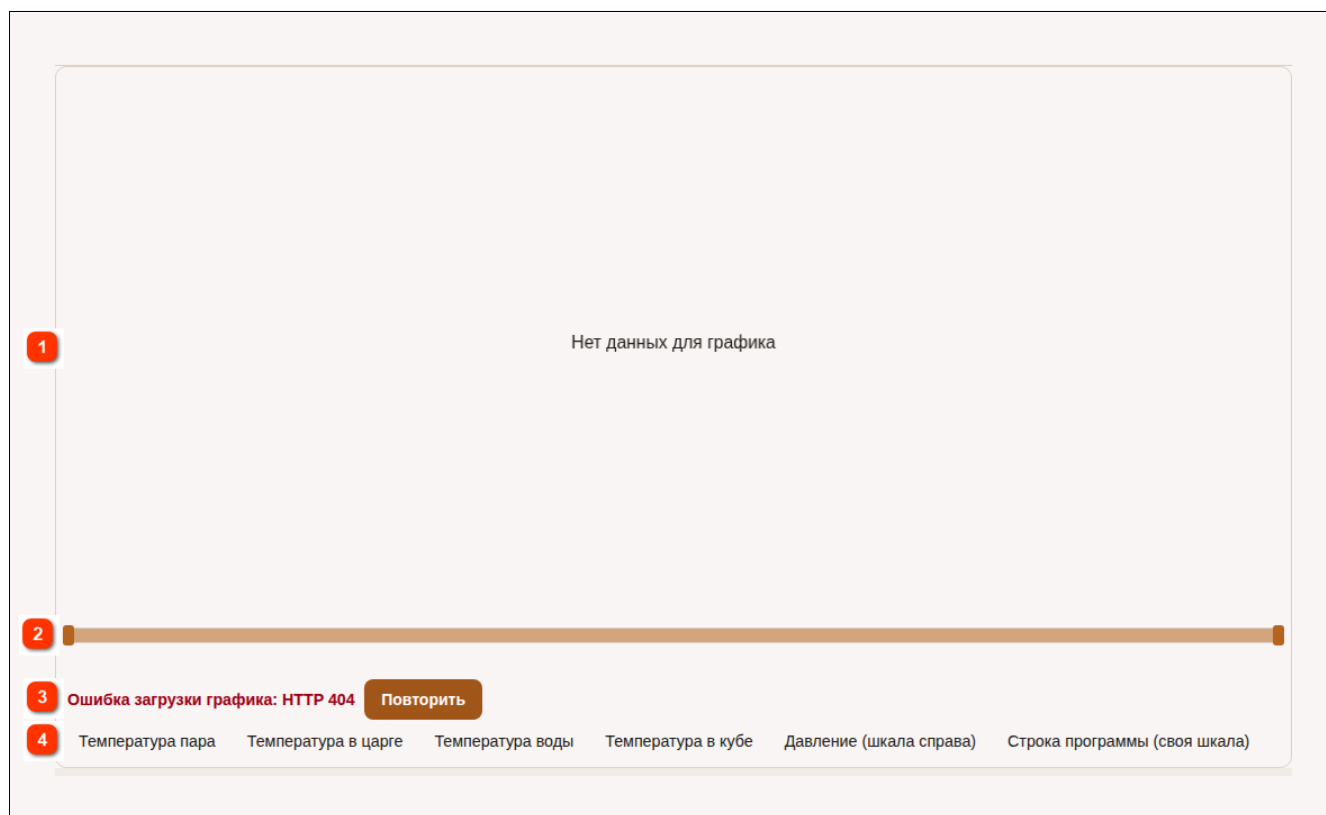
NBK and Cheesemaking - when the program starts.

In all modes logging ends on the "Turn heating off" («Отключить нагрев») or "Reset process" («Сбросить процесс») command

If the heating was switched off during one run and the process was then resumed, the log starts to be recorded anew each time, by the conditions listed above.

Samovar 7.00: the log interval is not configurable - a line is written once every 4 seconds. The "Log data recording" («Запись данных в лог») field has been removed from the [Settings - Other](#) tab. If less than 16 KB of free memory is left, Samovar deletes the old log data_old.csv by itself.

The local chart viewing window looks like this:



Local chart

- 1 Chart**
The temperature scale is on the left, the pressure scale on the right. The lower value of the temperature scale is taken from the minimum value of any sensor.
- 2 Section handles**
The left and right handles set the displayed section. Double-click on the bar - the whole chart.
- 3 Status line**
How many points are loaded and a control hint.
- 4 Legend**
Clicking a series hides or shows it.

The chart is drawn by the firmware without external libraries. On the left is the temperature scale, on the right the pressure scale; the program line number is shown as a separate series. Below the chart is a status line: "Points loaded: N. The sliders under the chart set the section, double-click on the bar - the whole chart, legend - on/off, hover - values." («Загружено точек: N. Ползунки под графиком задают участок, двойной щелчок по полосе - весь график, легенда - вкл/выкл, наведение - значения.»)

Samovar 7.00: while the chart is open, marks of pauses and waits may appear next to the numeric series. These are separate messages of the current run that the page keeps in memory; they are not lines of data.csv or data_old.csv. If both the start and the end of one reason were received, the interval, the duration and the outcome are shown; if only one part was received, it is an incomplete event with no invented duration. Missed messages or a reboot of Samovar can leave the history incomplete or clear it.

A mark is placed on the time axis only when the time is reliable. If there is no time or the Samovar clock has jumped noticeably, the information about the reason and the duration stays in the status line, but no interval is drawn on the chart. Older program line marks and pause events remain separate observations, not intervals.

Legend

You can turn off any series by clicking its legend entry. For example, if you click "Pressure" («Давление»), the pressure series is not displayed. Clicking again shows it again. Sensors hidden on the main page are hidden here as well.

Chart section

Under the chart there is a bar with two handles. By dragging the left and right handles you can choose any level of detail you need for analysis. A click on the bar shifts the selected section, a double-click returns the chart to its original view. The handles can also be moved with the keyboard arrows.

When you hover the cursor over the chart, a vertical line appears with the exact values of all series at the cursor position.

While you are viewing it, the chart is automatically extended with new points no more often than once every 15 seconds, but this update can be turned off by clearing the "Update chart" («Обновлять график») checkbox in the window below.

Статус

1 ☒ Обновлять график

Текущее время: 12:34:56 ● Время работы: 01:12:30

Т в царе: **77.900 °C** Т воды: **22.400 °C**

Т пара: **78.150 °C** Т в кубе: **84.300 °C**

Δ температур: **-0.250 °C** Т в ТСА: **40.100 °C**

Скорость отбора: 600.00 л/ч Прогресс отбора: 35 %

Отобранный объем: 350 мл

Статус: Отбор тела; Тело

Выключить нагрев Назад Сохранить лог 2 Сохранить старый лог 3

Системные параметры: Непр=152000; BME temp=24.5; RSSI = -58; Свободно: 300000

The "Status" («Статус») window under the chart

- 1 **Update chart («Обновлять график»)**
Clear the checkbox so the chart is not extended with new points.
- 2 **Save log («Сохранить лог»)**
Download the current log data.csv.
- 3 **Save old log («Сохранить старый лог»)**
Download the previous log data_old.csv.

Turning off the chart update

Turning off the update makes it more comfortable to view and analyze the chart. Writing the log to the file does not stop. If you check the box again, all events that happened in the meantime are shown on the chart.

Attention!

The chart is built on your computer; if you work with a large log, this can take a considerable time. If Samovar is busy, the log download is retried automatically.

On the same page (in the second window) there is a "Save log" («Сохранить лог») button. When you press it, at the end of the process the current log is saved to your computer, into the "Downloads" folder of your browser. The file name is data.csv .

If a new log has already started recording, you can still download the previous one by pressing the "Save old log" («Сохранить старый лог») button. The file name is data_old.csv

Even if you have only switched Samovar on but have not started heating or take-off, two logs are available for saving: the latest one (data.csv) and the previous one (data_old.csv).

When the process starts under the condition, the new log replaces the old one.

Example.

If during a run the heating was switched off five times and the process was resumed, and the log start conditions were met each time, then locally you can view only the chart built from the data of the last log. The previous log can only be downloaded as the data file data_old.csv. The first three logs are no longer available locally.

But in the [cloud](#) all five logs are saved and five charts can be viewed. Moreover, in the cloud they can be merged into one log, and one chart is then built. You can also download the separate or the merged log to your computer.

Later you can view the chart on your computer using the chartlocal.htm page, which is located in the Samovar folder - tools

Samovar Operating Modes

1

Режим работы

Ректификация

Ректификация, дистилляция и БК до температуры в кубе:

0

°C

Время для контроля завершения процесса дистилляции:

0

мин

☐ Использовать корректировку давления

☐ Использовать датчик уровня флегмы при ректификации

☐ Не показывать схему процесса

Параметры оборудования (колонны)

Диаметр:

1.5

дюймов

Высота насадки:

0.50

метров

Плотность насадки:

80 % (0-100)

Детектор проскока примесей (ректификация)

☐ Использовать детектор

Давление, при котором включится авария:

0

мм. Рт. ст.

Нулевая точка MPX:

0

Отсчётов MPX на 1 мм рт. ст.:

0

Настройки зуммера

☐ Использовать зуммер для нотификации

☐ Использовать зуммер для нотификации в браузере

☐ Включать зуммер при смене программы

Самотестирование

Внимание: включится подача воды (насос или клапан), запустится перистальтический насос отбора, сервопривод пройдёт 5 положений ёмкостей.

Choosing the operating mode on the "Settings → Main" («Настройки → Основные») tab

1

Operating mode («Режим работы»)

The list of modes. Sous-vide and Lua mode are hidden in the list.

In the web interface you can choose one of the modes:

[Rectification](#) («Ректификация»)

[Distillation](#) («Дистилляция»)

[Beer](#) («Пиво»)

[Wash column \(BK\)](#) («Бражная колонна (БК)»)

[Continuous wash column \(NBK\)](#) («Непрерывная бражная колонна (НБК)»)

[Cheesemaking](#) («Сыроварение»)

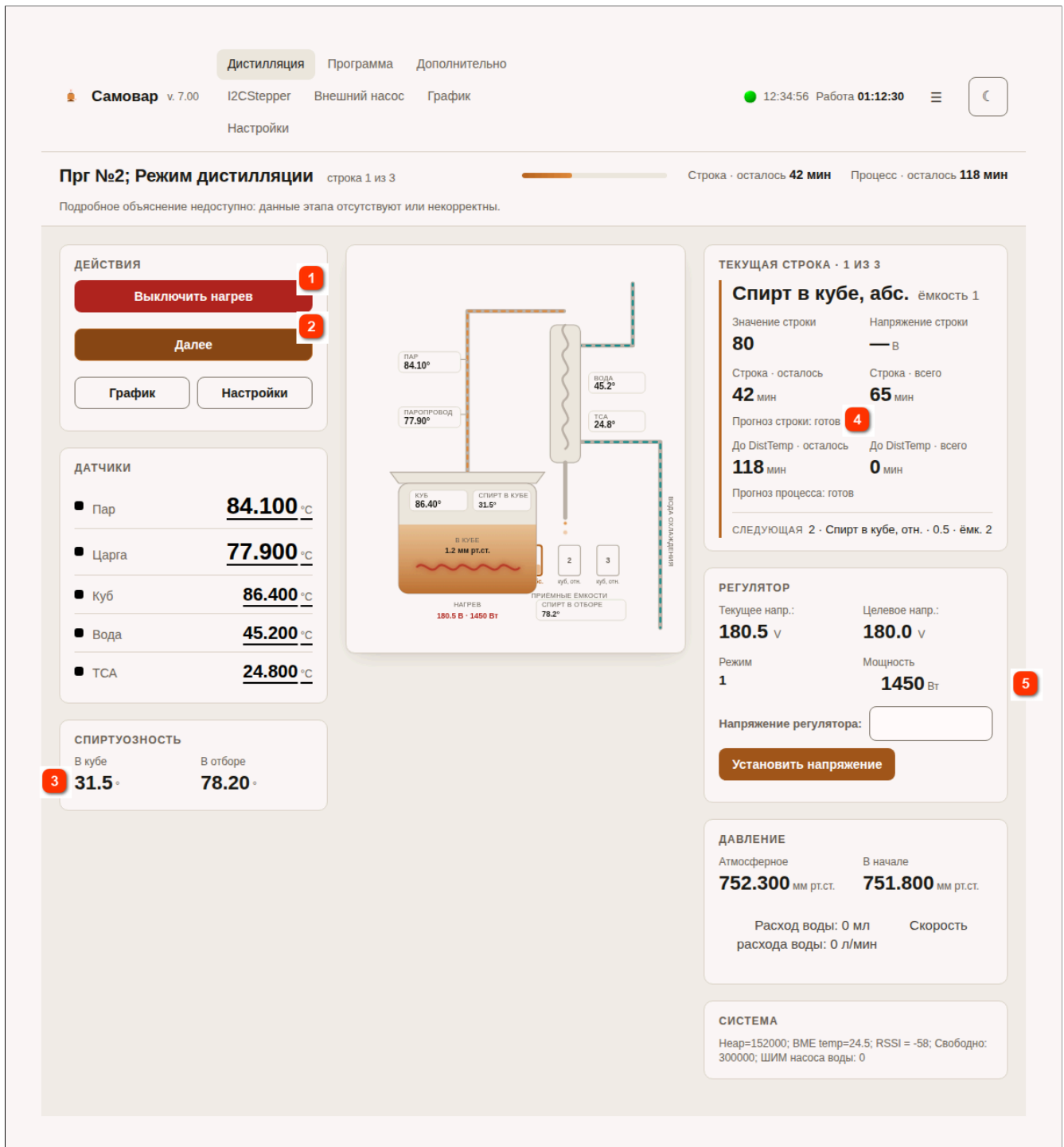
The items [Sous-vide](#) («Су-вид») and [Lua mode](#) («Lua-режим») are hidden in the list and are not turned on by the usual mode switching. The NBK and Lua modes are not present in every firmware build: if the mode saved in the settings is not available in the current build, it is shown in the list, but it cannot be selected again.

Samovar 7.00: the Cheesemaking mode has been added. Changing the mode no longer requires restarting the ESP32.

How to change the mode

- . If a process is running, stop it normally and wait for the heating and pumps to turn off. While heating is on, settings with a new mode are not saved.
 - . Open **Settings** → **Main** («Настройки → Основные»). In the **Operating mode** («Режим работы») field, select the mode you need.
 - . Before saving, connect the sensors and equipment that the new mode needs.
 - . Press **Save** («Сохранить»). Samovar closes the log, cancels pending commands and switches the mode without a restart, and the browser goes to the main page of the selected mode.
 - . Open **Settings** → **Temperature** («Настройки → Температура»), press the sensor list refresh and assign each sensor to its place. Press Save again.
 - . Check the program of the selected mode and run the [self-test](#) before turning on heating.
- Saved programs and settings are separated by mode. If you use different sets of sensors, save a backup settings file for each configuration on the [Other](#) («Прочие») tab.

First distillation mode



The main window of Distillation («Дистилляция») mode

- 1 **Turn on heating («Включить нагрев»)**
Starts and stops distillation. When heating is on, the button is called "Turn off heating" («Выключить нагрев»).
- 2 **Next («Далее»)**
Forces a move to the next line of the distillation program. Available only when heating is on.
- 3 **ABV («Спиртуозность»)**
The calculated ABV in the boiler and in the take-off. Until the boiler starts boiling, 100% is shown.
- 4 **Time forecast («Прогноз времени»)**
How many minutes are left until the condition of the current line is met and until the end of the process. Recalculated every 30 seconds.
- 5 **Regulator («Регулятор»)**
The current and target voltage (power) of the regulator and the manual setting field. See [Manual regulator voltage setting](#).

Warning!

Only **three** temperature sensors are monitored: **water, boiler and TCA**

The interface shows the readings of all temperature sensors, if they are connected and assigned on the Settings («Настройка») → Temperature («Температура») tab.

This way you can use spare sensors as informational ones for your own needs. For example: to watch the temperature of the incoming water, etc.

Warning.

ABV before boiling starts is displayed as 100% - this is needed to determine the correct actual ABV later on.

On the "Turn on heating" («Включить нагрев») command, Samovar tells the regulator to switch on the "Boost" («Разгон») mode and displays the message "Distiller heating is on" («Включен нагрев дистиллятора»). You can change the power by typing values into the "Regulator voltage" («Напряжение регулятора») field. The set power is kept until the move to the next take-off program line.

If a [boost heating element](#) is connected, it turns on together with the heating and turns off at the first move to the next program line. If the "Use boost heating element during boiling" («Использовать разгонный тэн при кипении») checkbox is cleared on the Settings («Настройки») → Main («Основные») tab, the boost heating element turns off earlier - as soon as Samovar detects that the boiler has started boiling.

The "Next" («Далее») button forces Samovar to move to the next line of the [distillation program](#), without waiting for the condition of the current line to be met. It works only when heating is on.

In first distillation mode, Samovar displays and monitors the temperatures from three sensors:

Water. The water temperature can be regulated (provided that a water supply pump is used) by changing the setpoint of the water temperature sensor on the Settings («Настройки») - [Temperature](#) («Температура») tab. If the sensor in the TCA has heated above its setpoint and is hotter than the water, the pump temporarily increases the water flow.

Boiler. When the boiler temperature reaches the value specified:

1. In the file [Samovar.ini.h](#), line:

```
#define OPEN_VALVE_TANK_TEMP 77
```

The command will be given to open the water valve and turn on the cooling pump.

When the boiler starts boiling, Samovar determines the ABV and displays the message "Boiling has started in the boiler! ABV XX.X" («Началось кипение в кубе! Спиртуозность XX.X»).

Distillation ends (and the heating is accordingly switched off) when any of the following conditions occurs:

- The boiler temperature exceeds the [distillation end temperature](#) set in the settings.
- There is no alcohol left in the boiler (this is determined by the boiler temperature not changing by 0.1° during the time specified in the "Time to monitor distillation completion" («Время для контроля завершения процесса дистилляции») field on the [Settings](#) («Настройки») tab, provided that the boiler temperature is above 90°). Samovar reports "No alcohol left in the boiler" («В кубе не осталось спирта»). A value of 0 disables this check.

When it ends, the message "Distillation finished. Total time: N min." («Дистилляция завершена. Общее время: N мин.») is displayed.

After the heating is switched off, the water supply valve closes and the pump turns off when any of the following conditions occurs:

- the water temperature has dropped 20° below the water sensor setpoint, but not sooner than 3 minutes after the heating was switched off;
- the boiler, which was hot during this process, has cooled below 70°.

Samovar 7.00: previously the water was shut off only by the first condition. If the water setpoint was lower than the incoming water temperature plus 20°, the cooling pump ran until the operator turned it off. Now the water is also shut off when the boiler cools down.

The line responsible for the temperature difference is in the file [Samovar.ini.h](#)

```
#define DELTA_T_CLOSE_VALVE 20
```

```
//The difference between the set water cooling T and the water T, down to which the pump will keep running after the distillation process ends in distillation and rectification modes
```

TCA. If steam breaks through, Samovar turns the heating off.

It is recommended to make a simple monitoring unit and use it during distillation together with [additional protection](#). In this case the TCA temperature sensor shows the temperature of the product coming out of the take-off spout, which lets you set the cooling water flow to the optimal rate.

When the water temperature approaches the critical value, Samovar sounds a signal, displays the message "Critical water temperature!" («Критическая температура воды!») and lowers the regulator voltage by 5 units; if the water or TCA temperature keeps rising, the heating is shut off in an emergency.

If, on the Settings («Настройки») tab, the checkbox [Use pressure correction](#) («Использовать корректировку давления») is set, Samovar will correct the boiler temperature according to how the boiling point of the alcohol-water mixture depends on pressure.

Time forecast. In the "Current line" («Текущая строка») block, Samovar shows how many minutes are left until the condition of the current program line is met and until the end of the process (until the set distillation end temperature). The forecast is recalculated every 30 seconds; until the boiler starts boiling, "waiting for actual boiling" («ожидание фактического кипения») is shown, and for the first minutes after that, "collecting data" («накопление данных»). If all program lines have been executed, no line forecast is made.

A log is kept, information is sent to Blynk and the cloud, and charts are drawn. The safety group is active.

Questions and answers. FAQ

Since version 6.0 a take-off program has been added to distillation mode; does this mean the servo drive must be used for the take-off?

If you do not need to distribute the product among containers, you do not have to connect the servo drive. It will then be plain distillation, regardless of what is written in the distillation program.

Distillation program

The program for the Distillation mode. Its purpose is to change the take-off container and/or change the voltage based on temperature or alcohol content. Nothing more.

Attention!

Note that the logic of switching containers differs from the logic in the rectification mode. There, the switch to a container happens when moving to the program line that specifies this container; in the distillation mode, the switch to another container happens when the event monitored by the current program line occurs.

There is one more difference from the rectification mode: there, finishing the program lines means the end of rectification, while in the distillation mode the process continues until the operator stops it or a stop by boundary conditions triggers: the boiler temperature exceeds the distillation [end temperature](#) set in the settings, or there is [no alcohol left](#) in the boiler.

The algorithm for starting and finishing distillation has not changed, so whether there is a program or not does not matter to you if you do not plan to use distribution among containers.

The alcohol content is determined both in the boiler and in the vapor (which for distillation is equivalent to in the take-off). The calculation is purely mathematical; in real conditions it may differ from hydrometer readings. Both values are shown in the interface.

If you do not need to distribute among containers, you do not have to connect the servo. Then it will be plain distillation, regardless of what is written in the distillation program. Or change containers manually.

№	Тип программы	Значение	№ емкости	Напряжение
01	Спирт в кубе абс	80	4	0
02	Спирт в кубе отн.	0.50	2	0
03	Спирт в кубе отн.	0.30	3	0

Описание программы дистилляции в текстовом формате для сохранения в облаке:

Программа дистилляции:

Установить программу

Очистить программу

Program tab of the Distillation mode («Вкладка «Программа» режима Дистилляция»)

1 Number («Номер»)
The program line number.

2 Program type («Тип программы»)
Т (Temperature, «Температура») - distillation by the boiler temperature.

Example program line:

T;85;1;150 - When the boiler temperature reaches 85°, the servo moves to position No. 1, the regulator sets the voltage to 150 volts, and the move to the next take-off program line is made.

S (Alcohol in boiler, relative, «Спирт в кубе отн.») - distillation by the relative change of the alcohol content in the boiler. This is the ratio of the current alcohol content to the one at the start of the program line. For example, if you need to cut off take-off when the boiler alcohol content falls by 50% since the start of take-off. The value is a number greater than 0 and less than 1.

Example program line:

S;0.5;1;0 - When the boiler alcohol content falls by 50%, the servo moves to position No. 1, the regulator keeps the voltage specified in the previous line, and the move to the next take-off program line is made. Samovar remembers the boiler strength at the moment of the move.

Note:

To switch to the next container when 1/4 of the boiler alcohol content is reached and, say, change the heating power, add one more line to the program:

S;0.5;2;200 - then, when the alcohol content in the remaining part of the boiler contents falls by 50%, the servo moves to position No. 2, the regulator sets the voltage to 200 volts, and the move to the next take-off program line is made. Samovar again remembers the boiler strength at the moment of the move.

A (Alcohol in boiler, absolute, «Спирт в кубе абс») - distillation by the absolute change of the alcohol content in the boiler. The calculation is based on the absolute alcohol content in the boiler.

Example program line:

A;15;1;0 - When the boiler alcohol content falls to 15%, the move to the next program line is made.

P (Alcohol in take-off, absolute, «Спирт в отборе абс») - distillation by the absolute change of the alcohol content in the vapor. The same as mode A, but by the calculated alcohol content in the vapor (in fact, in the take-off).

R (Alcohol in take-off, relative, «Спирт в отборе отн.») - distillation by the relative change of the alcohol content in the vapor. The same as mode S, but by the calculated alcohol content in the vapor.

3**Value («Значение»)**

Depends on the program type: a temperature, or an absolute or relative alcohol content value.

4**Container number («Номер емкости»)**

The take-off container number (0..10). The servo moves to this position when the line condition triggers.

5**Voltage («Напряжение»)**

The voltage for the regulator (for SEM_AVR it is power in watts; the column header changes with the regulator type). Zero means ignore. Applied when the line condition triggers.

6**Set program («Установить программу»)**

Checks and saves the program. The "Clear program" («Очистить программу») button deletes all lines.

A program line in text form: *Type;Value;Container;Voltage*. You can edit the program either in the table or in the text field under it; after editing, press "Set program" («Установить программу»). The program cannot be changed while it is running.

Besides the temperature and alcohol content lines, there is a **Lua** type: the line is controlled by a Lua script from a file on Samovar. When you select this type, the "Lua stage" («Lua-этап») window opens (a .lua file, a timeout in seconds, script parameters); in text form such a line looks like *L;timeout;0;file.lua^parameters*. For details see [About the Lua scripting language](#).

In the program you can use either the alcohol content in the boiler or the alcohol content in the take-off, determined by the type of the take-off line.

Note:

On every move to the next program line, Samovar remembers the alcohol content value and takes it as 100% in the mathematical expression.

Program validation. When you press "Set program", Samovar checks the lines and shows a message on an error. Valid values: for T - temperature from 0 to 150; for A and P - alcohol content from 0 to 100; for S and R - a fraction strictly greater than 0 and less than 1 (for example 0.50); container number from 0 to 10; voltage (power) - an integer, may be negative. The first line with a nonzero voltage must set an absolute value (more than 40 V or 400 W), otherwise Samovar has nothing to count the increment from.

The program loaded "by default" (see the screenshot above for a graphical view)

A;80.00;1;0

S;0.50;2;0

S;0.30;3;0

will run as follows:

When the "Turn on heating" («Включить нагрев») command is given, Samovar reports:

"Distiller heating is on" («Включен нагрев дистиллятора»)

"Moving to program line No. 1" («Переход к строке программы №1»)

When the boiler temperature set in the file Samovar_ini.h

/The boiler sensor temperature at which the water supply valve opens and the water supply pump turns on

#define OPEN_VALVE_TANK_TEMP 77

is reached, the command to open the water supply valve is given, and at the same time a PWM control signal is sent to turn the pump on.

Note that when a cooling pump is used, Samovar will hold the water temperature set on the "[Temperature](#)" («Температура») tab as the water sensor setpoint. If you need the pump to run "at full power", lower the setpoint value.

When the boiler starts boiling, Samovar determines the alcohol content and shows the message:
"Boiling started in the boiler! Alcohol content XX.X" («Началось кипение в кубе! Спиртуозность XX.X»)

Then it moves to the first program line, which says:

- if the absolute alcohol content in the boiler is less than 80%, move the servo to position No. 1.

And since the alcohol content in the boiler is obviously below 80%, it moves on to executing program line No. 2 and shows the message:

"Moving to program line No. 2"

Thus, from the moment boiling starts, take-off goes into container No. 1, and program line No. 2 is executed.

Next:

- when 50% of the boiler alcohol content relative to the start of executing line No. 1 is reached, move the servo to position No. 2 and go to the next line.

Accordingly, Samovar moves on to executing program line No. 3 and shows the message

"Moving to program line No. 3"

Take-off now goes into container No. 2.

That is: the alcohol content dropped by 50%, the container changed, and the move to the next program line occurred.

Next:

On moving to program line No. 3, Samovar determines the boiler alcohol content and takes this value as 100%.

When 30% of the boiler alcohol content from the start of line No. 3 is reached, it shows the message

"Programs finished, continuing take-off" («Выполнение программ закончилось, продолжение отбора»)

Explanation.

Suppose that on moving to line No. 3 the boiler alcohol content was 6%.

$6/100 \cdot 30$ (the value set in the program line) = 1.8%.

So when 1.8% alcohol is left in the boiler, Samovar reports that the execution of all program lines is finished, but the process will continue.

Low wines take-off into container No. 3 will continue until any of the conditions for turning heating off occurs, namely:

- the boiler temperature exceeds the distillation end temperature [set in the settings](#);
- the boiler [runs out of alcohol](#)

Then Samovar turns heating off and, [when the water temperature drops by 20° or the boiler cools down](#), stops the pump and closes the water valve.

Note: the "Next" («Далее») button in the main window moves Samovar to the next line without waiting for the condition. The container and voltage of the current line are applied in the same way as when the condition triggers.

Examples of take-off programs.

Example No. 1

A relative take-off program with tails separation:

To determine the boiler alcohol content **accurately**, put two lines at the start of the program:

T;85;0;150

A;80;0;220

The first line: when the boiler temperature reaches 85°, Samovar drops the voltage to 150 volts and slowly brings the boiler to boiling. This is necessary for an accurate determination of the boiler alcohol content, if you need to distribute the take-off among containers exactly, relative to the total alcohol content in the boiler.

When the boiler starts boiling, Samovar determines the boiler alcohol content, remembers it and moves on to the next line:

A;80.00;0;220

Which says: if the absolute alcohol content in the boiler is less than 80%, set the voltage to 220 volts and go to the next line. Since the boiler alcohol content is in any case obviously below 80%, it will move on to the next line.

Further in the program you need to create lines based on the goals and tasks of the actual take-off,

S;0.50;1;0

This line means: take off until the **strength in the boiler** drops to 50% of the one remembered at the moment of moving to this line, move the servo to position No. 1 and go to the next program line.

S;0.30;2;0

Take off until the **strength in the boiler** drops to 30% of the one remembered at the moment of moving to this line, move the servo to position No. 2 and go to the next program line.

P;20.00;3;0

Take off until the **strength in the take-off** drops to 20%, then move the servo to position No. 3.

So this program

T;85;0;150

A;80;0;220

S;0.50;1;0

S;0.30;2;0

P;20.00;3;0

will run like this:

When the "Turn on heating" command is given, Samovar tells the regulator "Boost" («Пазрон»). After the boiler temperature reaches 85°, it reduces the voltage to 150 volts and waits for the boiler to boil. When the boiler boils, it determines the absolute alcohol content in the boiler (taking this value as 100% in the mathematical expression), remembers it, and since the absolute alcohol content of the boiler will obviously be below 80%, sets 220 volts on the heating element.

Low wines take-off into container No. 0 begins.

When the alcohol content in the boiler is 50% of the remembered one, it moves the servo to position No. 1, again remembers the new boiler alcohol content (taking this value as 100%) and takes off low wines into container No. 1 until the boiler alcohol content drops to 30% of the **new** remembered value.

When the condition is met, it moves the servo to position No. 2 and takes off low wines into container No. 2 until the **strength in the take-off** drops to 20%. It moves the servo to position No. 3 and takes off low wines into container No. 3 until any of these conditions occurs:

- the boiler temperature exceeds the distillation end temperature [set in the settings](#);
- the boiler [runs out of alcohol](#)

Then it turns heating off and, [when the water temperature drops by 20° or the boiler cools down](#), stops the pump and closes the water valve.

Example No. 2

If very precise separation of the run is not needed and you want to save time, but still want to split the run into two parts relative to the boiler alcohol content, you can use the following program:

P;60.00;0;0

S;0.50;1;0

It will run as follows:

After boosting, the low wines start flowing into container No. 0. There will be no power drop. When the strength in the take-off falls below 60% (which happens roughly 3-5 minutes after the start), Samovar remembers the new boiler alcohol content and continues take-off into container No. 0.

When the alcohol content in the boiler falls to 50% of the remembered one, it moves the servo to position No. 1 and takes off into container No. 1 until one of the conditions for ending distillation occurs.

Of course, this program can be supplemented with other conditions.

The voltage/power setting logic is the same as in the rectification mode. If 0, the current value does not change; if the absolute value is less than 40 volts (400 watts), it is added to the current one (or subtracted if it is below 0); otherwise the set value is applied.

The same line types T, S, A, P, R are used in the program of the [Wash column](#) mode, where a line has a fifth field, the steam temperature.

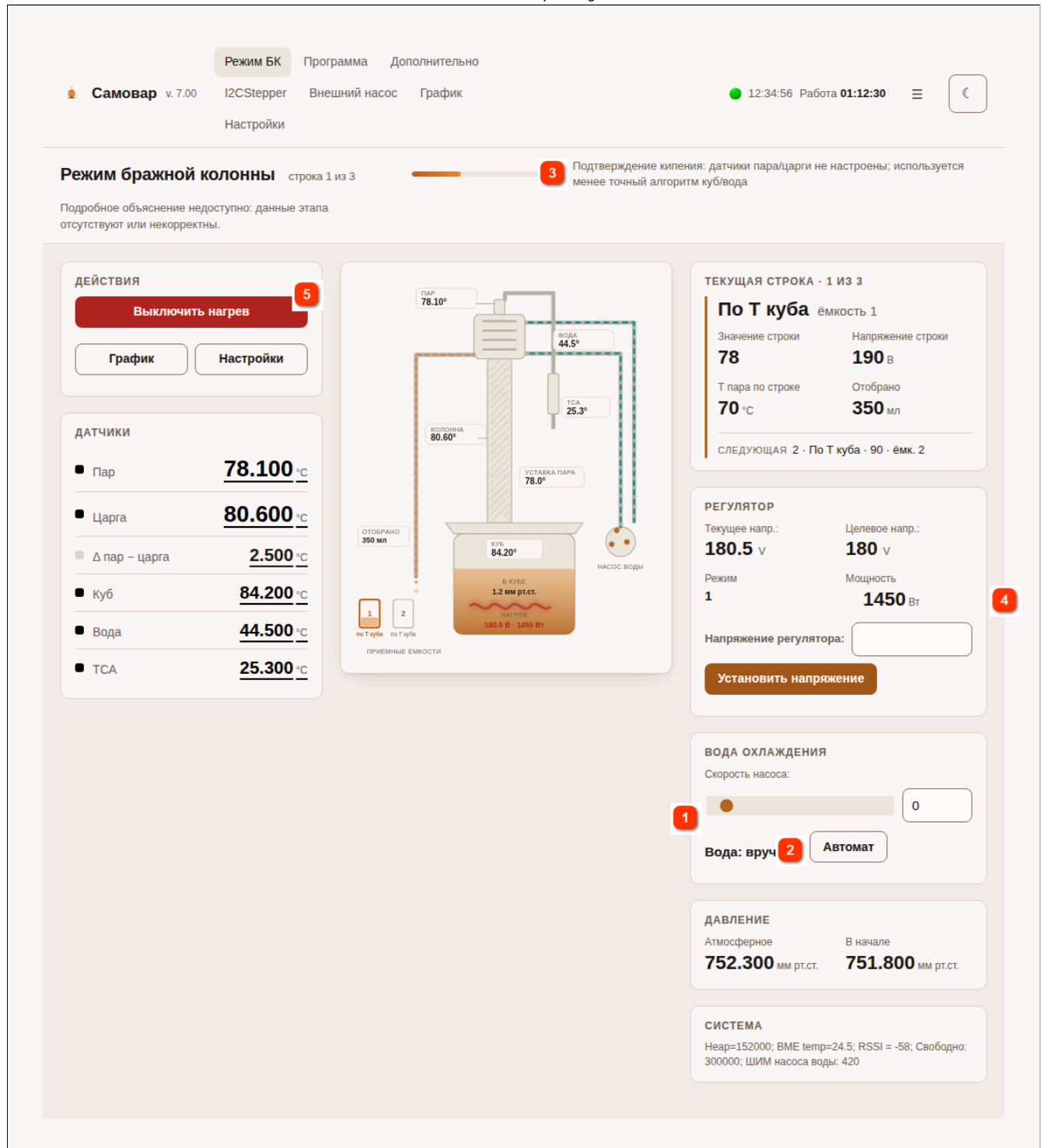
Wash column (BK)

A wash column (BK) is a reflux column fitted with a flow-through dephlegmator and a short column section with coarse or very sparse packing. It has proven itself for producing a purified distillate that keeps the taste and aroma of the original raw material.

The product is taken off "as vapor after the dephlegmator". Controlling the column comes down to two parameters: heating power and dephlegmator cooling. The take-off rate is regulated by the speed of the cooling pump.

Samovar 7.00: BK mode now has a take-off program. It defines up to what boiler temperature (or ABV) to continue the take-off, into which container, at what voltage, and what vapor temperature to hold. Samovar holds the vapor temperature by itself, changing the water supply to the dephlegmator. Lines with a vapor temperature of 0 leave the cooling under manual control, as in older versions.

Main window of the operating mode



Main window of the Wash column (BK) mode

- Cooling pump**
The slider adjusts the coolant flow rate.

The power is changed by PWM modulation.

You can control it either with the slider on the speed scale or by entering values directly in the field.

The leftmost slider position is the minimum pump speed (PWM_LOW_VALUE, 100 units by default).

If the slider is in the rightmost position, 12 volts are applied to the pump without modulation. The input field value =1023.

So, as a rough estimate for simplicity, 10 units = 1% of power. Any manual change turns off the automatic vapor temperature control until the end of the line.

2

Auto («Автомат»)

Returns the cooling to Samovar control based on the vapor temperature from the current program line. The button is visible only when the heating is on and the line has a vapor temperature set.

3

Boiling confirmation

Which sensor Samovar used to confirm that the boiler is boiling: vapor and column section is an accurate sign; boiler, water or their combination is less accurate. Before boiling - "waiting for confirmation" («ожидание подтверждения»).

4

Regulator

Voltage regulator control. See [Manual voltage setting of the regulator](#).

5

Turn heating on

Starts and stops the column. When the heating is on, the button is called "Turn heating off" («Выключить нагрев»).

The remaining data is the same as in the main window of the "[Rectification](#)" («Ректификация») mode

Start and boost

Before starting, assign the boiler sensor: without it Samovar refuses to start with the message "BK not started: boiler sensor not assigned or not responding" («БК не запущена: датчик куба не назначен или не отвечает»). If the program has lines with a vapor temperature above 0, a vapor sensor is also required ("BK not started: program requires a vapor sensor" («БК не запущена: программа требует датчик пара»)).

After the heating is switched on, Samovar shows the message "Wash column heating on" («Включен нагрев бражной колонны») and sends the "Boost" («Разгон») command to the regulator.

When the boiler temperature reaches 77°C

in the file [Samovar.ini.h](#) the line:

```
#define OPEN_VALVE_TANK_TEMP 77 //Boiler sensor temperature at which the water supply valve opens
```

Samovar opens the water supply valve and switches on the cooling pump at the speed set by the "Pump speed" («Скорость насоса») slider. The minimum pump speed is set by the line

```
//Lower value (in percent) for the PWM regulator of the pump. Choose a value that ensures water flow
```

```
#define PWM_LOW_VALUE 10
```

Attention.

It is recommended to set maximum cooling right away.

The boost continues until Samovar confirms that the boiler is boiling. Signs of boiling: the vapor or column section temperature is above 39°C, or the boiler temperature stops rising while the water temperature keeps rising. The sensor that confirmed boiling is shown in the "Boiling confirmation" («Подтверждение кипения») line in the main window.

After the boiler starts boiling, Samovar sets the configured BK working power. The default value is 45 V for a voltage regulator or 200 W for SEM_AVR. You can change it in the "BK power after boiling" («Мощность БК после закипания») field on the [settings](#) page. If the first program line specifies a voltage, it is applied on top of the working power (as an increment if it is less than 40 V / 400 W, otherwise as an absolute value). Then Samovar proceeds to the first program line.

Program

Программа БК:

№	Тип программы	Значение	№ емкости	Напряжение	Т пара, °C
01	Температура	78	1	19	70
02	Температура	90	2	0	78
03	Температура	93	2	0	0

► Описание программы БК в текстовом формате для сохранения в облаке:

► Программа БК:

Установить программу

Очистить программу

The "Program" («Программа») tab of the Wash column (BK) mode

- Program type**
The condition for moving to the next line: boiler temperature (T), ABV in the boiler or in the take-off, absolute or relative (A, S, P, R). The types are described on the page [Distillation program](#).
- Voltage**
Regulator voltage (power), applied when the line condition is met. 0 - do not change. For the first line it is applied right after boiling starts, on top of the BK working power.
- Vapor T, °C**
The vapor temperature that Samovar holds on this line by supplying water to the dephlegmator: 0 - water is controlled manually, 30..100 - automatically. Other values are not accepted ("Program error: Vapor T: 0 or 30..100" («Ошибка программы: Т пара: 0 или 30..100»)).

Program line: *Type;Value;Container;Voltage;Vapor T*. The first four fields and the line types T, S, A, P, R, L (Lua stage: the .lua file, the timeout in seconds and the parameters are set in the "Lua stage" («Lua-этап») window) are the same as in the [distillation program](#): when the line condition is met, the servo moves to the specified container, the regulator receives the specified voltage, and Samovar moves on to the next line. The fifth field is the vapor temperature to hold on this line: 0 - water is controlled manually, 30..100 - Samovar selects the water supply to the dephlegmator by itself.

Example program:

T;78;1;190;70

T;90;2;0;78

T;93;2;0;0

After boiling starts, Samovar holds a vapor temperature of 70°C (heads take-off). When the boiler heats up to 78°C, the servo moves to container No. 1, the regulator gets 190 V, and the vapor setpoint becomes 78°C. At 90°C in the boiler - container No. 2, the voltage does not change, and the cooling goes to manual control. At 93°C in the boiler the lines end, and the take-off continues until the end condition.

If the program has no lines, after boiling starts Samovar works as before: the operator controls the power and cooling manually.

Dephlegmator cooling

With automatic cooling, once a minute Samovar compares the vapor temperature with the line setpoint. If the vapor is warmer than the setpoint by more than 0.2°C, the water supply is increased by one step (30 PWM units out of 1023); if it is colder, the supply is decreased, but not below the pump minimum and not when the water is already hotter than 65°C. The state is shown by the label "Water: auto 78.0 °C" («Вода: авто 78,0 °C») or "Water: manual" («Вода: ручную»).

Any movement of the "Pump speed" («Скорость насоса») slider (or entering a value in the field next to it) turns the automation off until the end of the line, and Samovar reports "Dephlegmator water: manual control" («Вода дефлегматора: ручное управление»). You can turn the automation back on with the "Auto" («Автомат») button; it is visible only when the current line has a vapor temperature set.

Following the process sheet of the upcoming run, the operator sets the required heating power and, gradually reducing the cooling supply, achieves drops from the take-off spout of the condenser.

- The heads take-off has started.

Then, during the hearts take-off, the required take-off rate is likewise achieved by adjusting the heating and cooling.

Samovar does not take atmospheric pressure into account in this mode.

Completion and safety

The safety system is active.

If vapor breaks through and the TCA sensor is connected through the [control unit](#) to the take-off spout, the heating is switched off.

When the water temperature approaches the critical value, Samovar sounds a beep, shows "Critical water temperature!" («Критическая температура воды!») and reduces the regulator voltage by 5 units.

Samovar switches off the heating with the message "Wash column run finished" («Работа бражной колонны завершена») on any of these conditions:

- the [set boiler temperature](#) is reached;

- there is no alcohol left in the boiler: the boiler temperature is above 90°C and does not rise by 0.1° during the time from the "Time for distillation completion check" («Время для контроля завершения процесса дистилляции») field (0 - the check is off).
After the heating is switched off, the water is shut off in the same way as in [distillation mode](#): when the water temperature drops 20° below the setpoint, or when the boiler cools down.

A log is kept and charts are plotted.

Continuous wash column

NBK is a continuous wash column: the wash is fed by a pump constantly, while Samovar controls the power, the feed and the transitions between stages. Below are a description of the mode and the sequence of actions in the Samovar web interface.

Do not start by switching on the heating. First check the pump, the sensors, the cooling and the program without a power load.

How the mode works

The NBK program always consists of four lines - stages that go in order:

Warm-up (H) («Прогрев»). Samovar switches on the power and the feed from line H and waits until the vapor temperature reaches 75 °C. Then it moves on to Setup by itself.

Setup (S) («Настройка»). A manual stage with no time limit: Samovar sets the power and the feed from line S, and from then on the operator changes them. To go further, use the "Next" («Далее») button.

Optimization (O) («Оптимизация»). An automatic search for the working point: Samovar changes the feed and the power in steps, waiting the "Inertia" («Инерция») time after each change, and remembers the optimal values it finds, **Mo** («Мо») (power) and **Po** («По») (feed). It requires a level sensor in the head, enabled when the firmware is built.

Operation (W) («Работа»). Samovar holds the working point it found and gradually adjusts the feed according to the stillage and vapor temperatures and the pressure. Manual control of the power and the feed is locked at this stage.

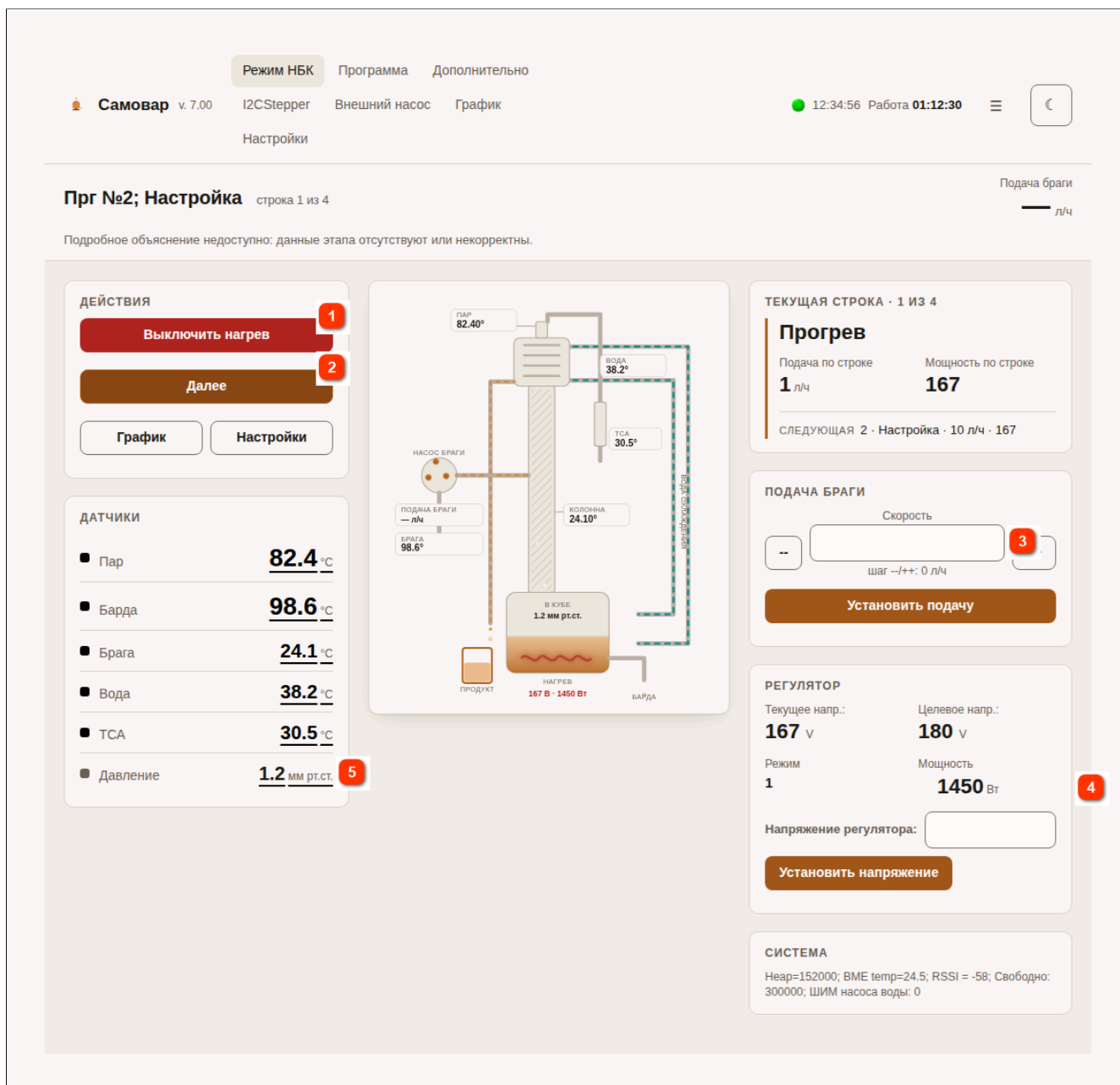
The current stage is shown in the status of the main page, for example "Prg no. 2; Setup" («Ппр №2; Настройка»). At any stage other than Setup, when the vapor temperature is above 98 °C, Samovar considers that the wash has run out and ends the program with the message "The wash has run out. The NBK program is finished." («Кончилась брара. Проррамма НБК завершена.») At the end, statistics are shown: how many liters of wash were processed, the average rate and the session time.

Flooding. Flooding is indicated by the level sensor in the head (DZ) or by the pressure in the stillage drain (DD) rising above the "Flooding pressure" («Давление захлёба») threshold. At Setup, Samovar reduces the feed to 1/3 and the power to 1/2 and reports it; at Optimization, until the optimum is found, flooding stops the process, and after that it switches to Operation; in Operation, Samovar holds a pause (two "Inertia" times) at reduced power and feed, then continues with reduced Mo and Po. If "Switch the flow with the servo on flooding" («Переключать поток сервоприводом при захлёбе») is enabled in the settings, then during flooding the servo moves the flow to position 1 (dirty product) and returns it to position 0 after the pause.

Cooling. The water valve and the cooling pump switch on when the boiler (stillage) temperature reaches 77 °C and run until the heating is switched off; they close when the water has cooled to 20° below the water sensor setpoint ([Settings - Temperature](#)) or the boiler has cooled. If the TCA sensor heats up to 60 °C or the water to 75 °C and stays that way for 60 seconds, Samovar stops the process with the message "Insufficient cooling! Stop." («Недостаточное охлаждение! Останов.»)

Diagram. In the center of the main window is a picture of the column with live readings: the wash pump with the current feed in L/h, the wash, vapor, water and TCA temperatures, the pressure in the boiler, the product and the stillage drain. The pump rotor spins while the feed is running with the heating on, the heating element is highlighted during heating, the product jar and drops are shown while running, and a pause icon appears on the diagram during a pause. The same diagram is in the apps and on the website ([Mobile apps and website](#)).

Main window of the operating mode



Main window of NBK mode at the Setup («Настройка») stage

- 1 Turn on heating**
Starts the NBK program from the Warm-up stage. While running, the button is called "Turn off heating" («Выключить нагрев») and stops the process normally.
- 2 Next**
Goes to the next stage: from Setup to Optimization, skipping Optimization (in the first 30 seconds), resuming Operation from safe standby.
- 3 Rate**
The wash feed, L/h. Enter a value and press "Set feed" («Установить подачу»). The "--" and "++" buttons change the feed by one "Feed step" from the settings. Available only with the heating on at Warm-up and Setup.
- 4 Regulator**
The current and target voltage (power). Manual setting works at Warm-up and Setup; it is locked at Optimization and in Operation.
- 5 Pressure**
The pressure in the stillage drain from the DD sensor. Exceeding the "Flooding pressure" threshold from the settings is considered flooding.

1. Check that the mode is available

- . The firmware must be built with power regulator control. For PlatformIO, suitable environments are **Samovar** (KVIC), **Samovar_rmvk**, **Samovar_sem** or a properly configured **Samovar_s3**. In the **Samovar_no_power** build NBK does not start: Samovar will answer "NBK start rejected: the power regulator is not available in this build." («Запуск НБК отклонён: регулятор мощности недоступен в этой сборке.»)
- . On the main page, press **Settings** («Настройки»). On the **Main** («Основные») tab, in the **Operating mode** («Режим работы») field choose **Continuous wash column** («Непрерывная бражная колонна»).
- . Press **Save** («Сохранить») at the bottom of the page. After saving, open the main page: the **NBK mode** («Режим НБК») page should appear.

2. What must be connected

A **controllable power regulator**, chosen when the firmware was built. Test it with a manual command on a safe load.

An **external I2CStepper pump** for feeding the wash. It must be shown on the NBK page and accept commands. For connection, see the section [I2CStepper](#).

The **vapor sensor** is required for Warm-up, Optimization and Operation. The **stillage sensor** is assigned in Samovar as the boiler sensor and is required for Optimization and Operation. A sensor assigned as the column section sensor is shown on the page as **Wash** («Бпара»).

Cooling: water must flow freely through the condenser, and the valve and the cooling pump must switch on in the right direction. Check them before feeding the wash.

Flooding protection: the level sensor in the head (DZ) and/or the pressure sensor in the stillage drain (DD). Physically check that the sensor triggers before a hot start. If DZ is enabled in the firmware, Samovar considers it available and cannot tell that the sensor itself is not connected; so the absence of a warning does not prove that the protection works. If there is no sensor at all, at startup Samovar warns: "Attention: there is no flooding sensor at all (DZ is off, DD does not respond). Flooding protection does not work." («Внимание: нет ни одного датчика захлёба (ДЗ выключен, ДД не отвечает). Защита от захлёба не работает.») Automatic Optimization requires the level sensor in the head to be enabled at build time.

Before the first hot start, open the NBK page and make sure that the temperatures are not zero and do not show an obvious error, that the DD pressure is displayed if it is present, and that the I2C pump has the "connected" («подключён») state.

3. Calibrate the wash feed pump

The calibration tells Samovar how many motor steps are needed to deliver one milliliter. Without it you cannot set the rate in liters per hour correctly.

Open the **I2CStepper** page (a button on the main page); if there are several boards, choose the pump in the "Device" («Устройство») list and press **External pump calibration** («Калибровка внешнего насоса»).

Put the inlet hose in water and point the outlet into a measuring container.

Set a small, stable motor speed and press **Start calibration** («Начать калибровку»).

When the measuring container holds exactly 100 mL, press **Capture 100 mL** («Зафиксировать 100 мл»).

Samovar will calculate the steps per milliliter and write them into the board's memory. On the **Settings** → **I2CStepper** tab, check that the **Steps/mL** («Шагов/мл») field is greater than zero. Repeat the calibration after changing the driver microstepping, the motor, the pump head or the tubing.

4. Fill in the NBK settings

Open [Settings](#) → [NBK](#). For the first start you can leave the factory values, and then refine them for your column:

Inertia - 180 s. This is the waiting time after changing the power or the feed. If your column takes longer to reach a new stable state, enter the time you actually measured.

Delta - 0.5 °C. The permitted deviation of the stillage temperature from the reference temperature.

Power step - 100. By this many watts the automation changes the internal calculated power while searching for the mode.

Feed step - 0.5 L/h. The feed changes by this much during Optimization and with the "--"/"++" buttons; in Operation the normal correction is one tenth of this step.

Vapor T - 81 °C. The lower limit of the vapor temperature for the algorithm. The permitted value in the firmware: above 80 and no higher than 97 °C.

Flooding pressure - 40 mm Hg. The DD trigger threshold. Refine it by the actual flooding of your own column, not by someone else's unit.

Finish T (stillage) - 98.5 °C. The reference temperature of the stripped stillage. At the first iteration of Optimization, Samovar may raise it according to the sensor, but not above 102 °C.

Then open the **Other** («Прочие») tab and enter the **heating element resistance** and the **mains voltage** correctly. These two values are needed to calculate the maximum power; if one is empty or wrong, the start will be rejected with the message "NBK start rejected: incorrect setting - ..." («Запуск НБК отклонён: некорректная настройка - ...»). Press **Save**.

5. Prepare the four program lines

Программа НБК

№	Тип программы	С ¹ ть	Напряжение
01	Прогрев	1	167
02	Настройка	10	167
03	Оптимизация	0	0
04	Работа	0	0

▶ Программа НБК:

▶ Как работает автоматика и что влияет на НБК:

▶ Описание программы НБК в текстовом формате для сохранения в облаке:

Установить программу

Очистить программу

The "Program" («Программа») tab of NBK mode: four fixed lines

1

Rate

The wash feed at the stage, L/h. For Warm-up and Setup it must be greater than zero; for Optimization and Operation zero means "take the current or found value".

2

Voltage / Power

The heating power at the stage: volts for KVIC and RMV-K, watts for SEM_AVR. For Warm-up zero means the maximum calculated power; for Setup a non-zero value is required.

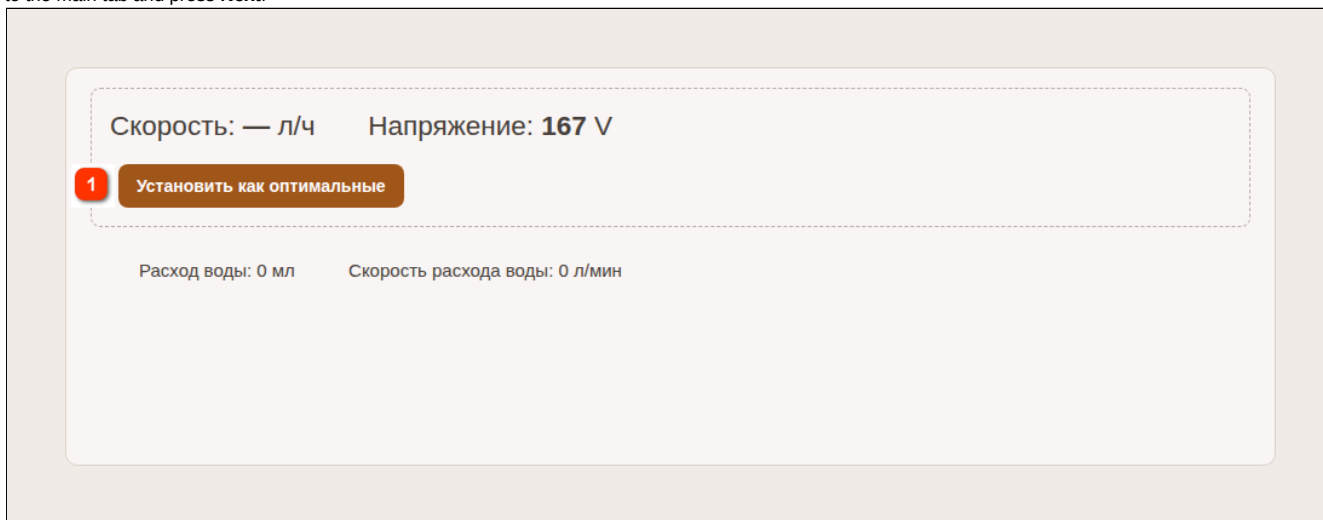
- Open the **NBK mode** page, then the **Program** («Программа») tab. The program must have exactly four stages: **Warm-up** → **Setup** → **Optimization** → **Operation**. In text form these are the lines *H;rate;power, S;...; O;...; W;...;*; any other set or order of lines is not accepted.
 - In the **Warm-up (H)** line, set a non-zero feed. If the power is zero, Samovar uses the calculated maximum power. Warm-up ends automatically at a vapor temperature of 75 °C.
 - In the **Setup (S)** line, be sure to set a non-zero feed rate and power. Start with safe values recommended by the manufacturer of your NBK and regulator.
 - In the **Optimization (O)** and **Operation (W)** lines, zeros mean using the values found at Setup or Optimization. If you skip automatic Optimization, set non-zero rate and power in line W in advance.
 - Press **Set program** («Установить программу»). Make sure the values stayed in the four lines and no format error message appeared.
- The power units depend on the regulator:** for SEM_AVR the field is set in watts, for KVIC and RMV-K - in volts; the column header on the tab changes accordingly. The rate in all lines is set in liters per hour. Do not copy someone else's power values without checking your own heating element and regulator.

6. Start Warm-up

- Fill the steam generator, make sure the stillage can drain, put the pump inlet into the prepared wash and open the path for the cooling water. Check that the hoses are not pinched.
- On the **NBK mode** tab, check the temperatures, the pressure and the pump state once more.
- Press **Turn on heating** («Включить нагрев»). Samovar will report "Starting NBK program. Warm-up" («Запуск программы НБК. Прогрев»), and the status will change to Warm-up. Samovar will switch on the power and the feed from line H.
- Do not walk away from the unit. Check that the liquid is actually moving, that there are no leaks, that the stillage drains normally and that the cooling works. At 75 °C on the vapor sensor, Samovar will move on to the Setup stage by itself.

7. Do the manual Setup

- After the transition to **Setup**, wait at least one **Inertia** time so that the temperatures and the output have time to settle.
- Change only one parameter at a time. Enter the power in the "Regulator voltage" («Напряжение регулятора») field and press **Set voltage** («Установить напряжение») (for SEM_AVR - **Set power** («Установить мощность»)); enter the rate in the **Rate** («Скорость») field of the "Wash feed" («Подача браги») block and press **Set feed** («Установить подачу»). The -- and ++ buttons change the feed by one "Feed step".
- After each change, again wait at least one Inertia time. Watch the vapor temperature, the stillage temperature, the pressure, a steady stillage drain and the absence of flooding.
- If flooding occurs, Samovar reports its source and reduces the actual feed to one third and the power to one half. Do not raise the parameters back right away; wait until the signs of flooding disappear and find the cause.
- If the vapor temperature at Setup stays at 100 °C or above for 60 seconds, Samovar stops the process: the steam generator may be running dry.
- When the mode has become stable, open the **Additional** («Дополнительно») tab and press **Set as optimal** («Установить как оптимальные»). Then go back to the main tab and press **Next**.



The "Additional" («Дополнительно») tab of NBK mode

1

Set as optimal

Remembers the current power and the actual feed as Mo and Po. Works only at Setup and in Operation.

8. Go through Optimization

After entering Optimization, there are 30 seconds during which you can skip it with the **Next** button. If you press nothing and the level sensor in the head is enabled, Samovar starts an automatic search for the working point from the power and the feed of line O (if they are zeros - from the current ones). If the stillage and vapor temperatures are not below the set limits, Samovar remembers the current values as optimal and increases the feed by one "Feed step". If the temperatures are below the limits, Samovar decreases the feed by one step and increases the power by one "Power step". After each change it waits the Inertia time.

After the optimum is found, the automation moves to Operation on flooding, on reaching the working pressure ceiling or another search limit (limit feed 30 L/h, limit power). The message "Optimization complete..." («Оптимизация завершена...») will give **Mo** - the power found and **Po** - the feed found; before Operation, Samovar holds a pause of two Inertia times at reduced power and feed.

If the level sensor in the head is not enabled in the firmware, automatic Optimization will not start. Samovar will report "Optimization is impossible - there is no flooding sensor..." («Оптимизация невозможна - отсутствует датчик захлёба...»). So the non-zero power and rate of line W must be set and saved **before pressing "Turn on heating"**. When Samovar moves to Optimization and shows the message, press **Next**. The program cannot be changed while the process is active. If you did not prepare line W in advance, turn the heating off, wait for it to stop, fix the program and start again. Do not leave the unit waiting: after 10 minutes Samovar goes into safe standby with the heating and the feed switched off.

9. Supervise Operation

At the **Operation** stage, manual setting of the power and the feed is locked: the algorithm controls them (the message "Manual control is not available at Optimization and in NBK Operation." («Ручное управление недоступно на Оптимизации и в Работе НБК.)). To change the working point, first stop the

process normally, fix the program or the settings and start again.

If the stillage or vapor temperature drops below the set limit, Samovar gradually reduces the feed. If the stillage temperature drops sharply (a wash spill), the feed is reduced at once by the full "Feed step".

If the pressure stays near the working ceiling, the feed is reduced. If the stillage temperature is steadily above the reference one, the feed may gradually grow, but not above the optimal value found.

On flooding, Samovar temporarily reduces the feed and the heating, holds a pause and then continues with reduced optimal values. Do not interfere until the pause is over, unless there are signs of a dangerous situation.

The **Set as optimal** button on the "Additional" tab also works in Operation: it replaces Mo and Po with the current power and the actual feed.

If the vapor temperature rises above 98 °C, Samovar considers that the wash has run out and ends the program normally.

10. How to stop the process and what to do on a message

For a normal stop, press the button that is called **Turn off heating** («Выключить нагрев») while running, and confirm the action. Wait for the feed to reach zero and the heating to switch off on the page. Samovar will report "NBK operation finished" («Работа НБК завершена»).

The message **Safe standby** («Безопасное ожидание») means that Samovar has already stopped the pump and the heating. First eliminate the stated cause.

At the Operation stage, resuming is done with the **Next** button; at Warm-up, Setup or Optimization this button ends the session, after which a new start is needed.

On messages about insufficient cooling, a pressure sensor failure ("No readings for more than 60 seconds" («Нет показаний более 60 секунд»)), or an unconfirmed pump or regulator command, do not try to continue the process until the cause is eliminated.

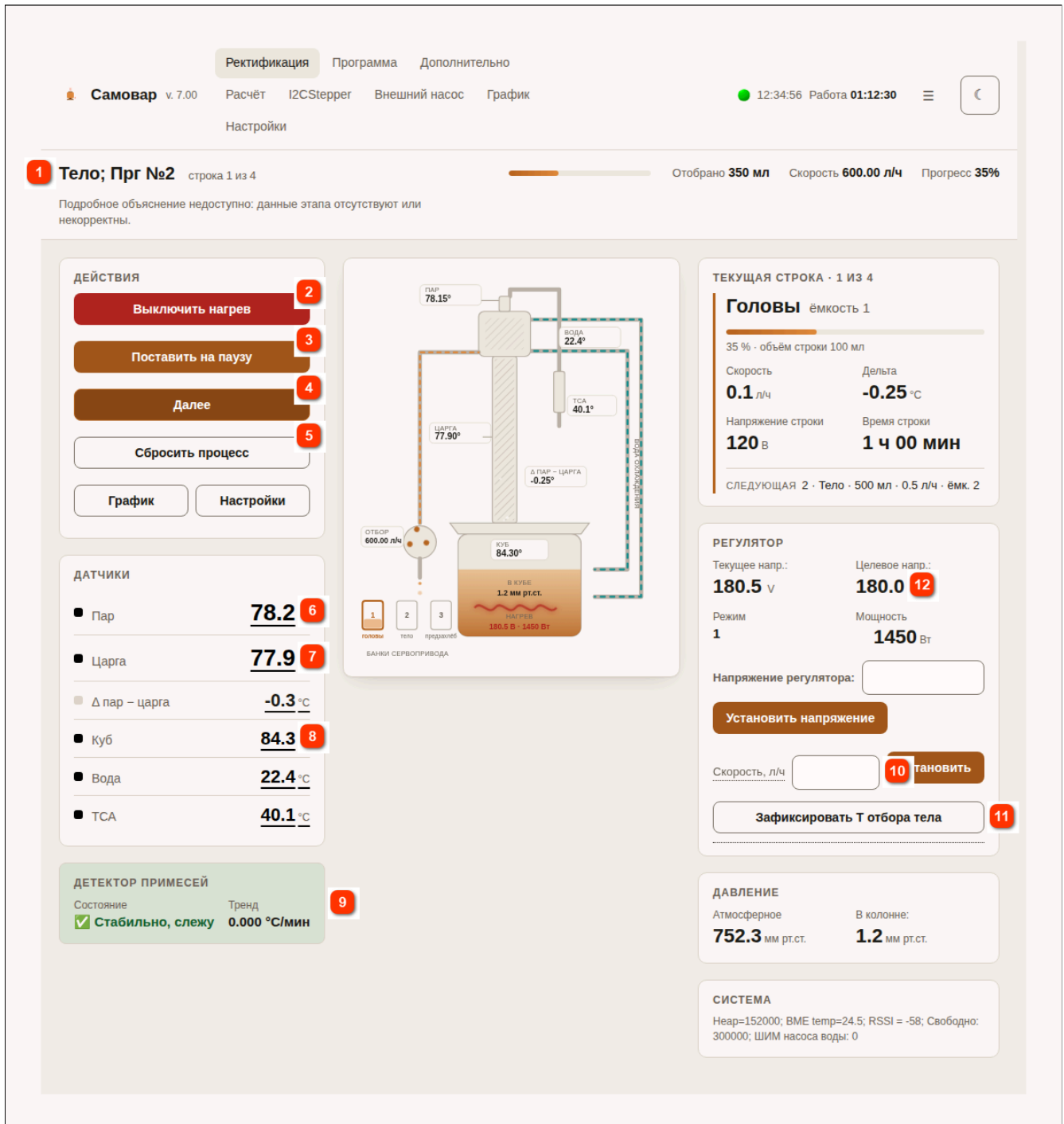
After the end, wait for the temperature to drop and the cooling to switch off normally. Only then shut off the water, take apart hot connections or de-energize the equipment.

Quick check before every start

- . NBK mode is selected, the settings are saved.
- . The I2CStepper pump is connected, calibrated and pumps in the right direction.
- . The vapor and stillage sensors show plausible temperatures; DZ/DD are checked.
- . The cooling is open and working; the stillage drain is clear.
- . The program has four lines H, S, O, W; H has a feed, S has a feed and a power.
- . The heating element resistance and the mains voltage are filled in correctly.
- . The operator stays next to the unit until stable Operation is reached.

Rectification mode

Main screen of the rectification mode



The main screen of the rectification mode, the "Rectification" («Ректификация») tab

- 1 **Status («Статус»)**
The current state of Samovar: the type and number of the running program line, pauses, the fixed hearts temperatures.
- 2 **Turn heating on («Включить нагрев»)**
Switches the power regulator on in boost mode. While the heating is on, the button changes to "Turn heating off" («Выключить нагрев»).
- 3 **Pause («Поставить на паузу»)**
Stops the take-off pump. To continue, press "Continue" («Продолжить»).
- 4 **Start take-off («Начать отбор»)**
Starts the take-off program. During the take-off the button is called "Next" («Далее») and moves Samovar to the next program line.
- 5 **Reset process («Сбросить процесс»)**
Stops the take-off and switches the heating off after confirmation.

Vapor temperature («Температура пара»)

- 6** The vapor sensor before the dephlegmator. It is used to determine the end of the boost and the column stabilization.
- 7** **Column section temperature («Температура в царге»)**
The sensor at 2/3 of the column height.
- 8** **Boiler temperature («Температура в кубе»)**
It opens the water valve (77°C) and ends the program when the set limit is reached.
- 9** **Impurity detector («Детектор примесей»)**
The state of the impurity breakthrough detector: observation, rate reduction or pause.
- 10** **Rate, L/h («Скорость, л/ч»)**
Manual change of the take-off rate of the current line. Enter a value and press "Set" («Установить»).
- 11** **Fix hearts take-off T («Зафиксировать T отбора тела»)**
The current vapor and column section temperatures become the hearts take-off temperatures.
- 12** **Regulator («Регулятор»)**
The set and the actual voltage (power) of the regulator.

Samovar 7.00: the take-off rate and the "Fix hearts take-off T" («Зафиксировать T отбора тела») button have been moved from the "Additional" («Дополнительно») tab to the "Regulator" («Регулятор») card on the main screen. An [impurity breakthrough detector](#) has been added. After the last program line Samovar does not switch the heating off immediately, but keeps the column in the "self-reflux" mode for 30 minutes.

When switched on or rebooted, Samovar starts in the "Off" («Выключено») state.

In this state it shows the sensor readings but controls nothing. If you press the "Turn heating on" («Включить нагрев») button in the web interface, Samovar switches the power regulator on (if connected) in boost mode and waits for the column to warm up. If the vapor, water and boiler sensors are not assigned in the settings, the heating does not turn on and the message "Sensors not assigned" («Датчики не назначены») appears on the screen. You can also start the boost with the [Universal button](#).

Until the boiler temperature reaches 77°C, the [boost heating element](#) (if connected) works together with the main heating element.

When the boiler temperature reaches 77°C

file Samovar_ini.h line:

```
#define OPEN_VALVE_TANK_TEMP 77 //Boiler sensor temperature at which the water supply valve opens
```

Samovar switches on relay No. 3 (we assume a valve that opens the water supply is actually connected to it), sounds a beep and shows the message: "Open the water supply!" («Откройте подачу воды!»).

The notification appears in any case - even if there is no valve or it has already been opened manually. At start the cooling pump runs on its starting power for a few seconds, then the PWM regulator adjusts the speed to hold the set water temperature,

file Samovar_ini.h line:

```
//Lower value (in percent) for the PWM regulator of the pump. Choose a value that ensures water flow
```

```
#define PWM_LOW_VALUE 10
```

When the vapor temperature reaches 39 degrees

file Samovar_ini.h line:

```
#define CHANGE_POWER_MODE_STEAM_TEMP 39 //Vapor sensor temperature at which the column switches from boost mode to working mode
```

Samovar turns off the boost mode, commands the regulator to set the voltage (power) specified in the first line of the take-off program, and switches to the column stabilization mode. The notification "Boost finished. Stabilization/self-reflux." («Разгон завершён. Стабилизация/работа на себя») is sent to the smartphone (if configured) and shown on the main screen.

Meanwhile, using the cooling pump, it maintains the cooling water temperature. When the boiler starts boiling, Samovar estimates and shows the ABV of the boiler liquid.

Samovar waits until the vapor temperature deviations are no more than 0.1 degree for six minutes. After that, stabilization is considered complete. The message "Stabilization complete, the column is running stably." («Стабилизация завершена, колонна работает стабильно.») is shown in the web interface. The six-minute counter takes only the vapor sensor into account; on any change of the reading greater than 0.1° the count starts over.

This logic is implemented in the file alarm.h

The decision to start the take-off is made by the operator. If the decision is positive, press the "Start take-off" («Начать отбор») button in the web interface, in the mobile app, or use the [Universal button](#). If the column has not stabilized yet, the web interface asks you to confirm the start. Before starting, Samovar checks the program: the pump must be calibrated, the first line must set an absolute voltage (power), and every take-off line must have a volume or a transition temperature set; otherwise the take-off does not start, and the reason appears in a message.

After the command, the [take-off program](#) starts and the peristaltic pump begins to work at the speed set in the program. The rocker arm of the distribution unit moves to the position specified in the take-off program.

Until all the lines of the **heads take-off** program are completed, Samovar:

- takes off the volume specified in the program,
- switches the flow between the take-off containers as the program specifies
- may change the take-off rate according to the program

And at the same time it ignores changes of the column section and vapor temperatures. That is, it simply takes off according to the program. On heads the impurity detector only observes.

When the program lines related to the heads take-off are finished, Samovar moves on to the **hearts take-off**

When the first line of the hearts take-off program starts, Samovar remembers the vapor temperature and the column section temperature, and from then on relies on these fixed temperatures. If the hearts line sets its own voltage (power), the temperatures are fixed not immediately, but after the vapor temperature has settled at the new power.

Note:

You can correct these temperatures at any moment while a hearts take-off program line is running by pressing the "Fix hearts take-off T" («Зафиксировать T отбора тела») button in the "Regulator" («Регулятор») card on the main screen. Samovar then overwrites the previously stored hearts take-off temperatures with the current sensor readings.

Note:

Also, while any program line is running, you can change the take-off rate at any moment by entering the required rate in liters per hour in the "Rate, L/h" («Скорость, л/ч») field of the "Regulator" card on the main screen and pressing the "Set" («Установить») button. The new rate applies only to the current program line.

If the "Use pressure correction" («Использовать корректировку давления») option is enabled in the settings, Samovar reduces the readings of all temperature sensors to a pressure of 760 mmHg using the pressure sensor. So the fixed hearts take-off temperatures do not depend on changes in atmospheric pressure during the run: when the pressure drops, the actual boiling temperature falls, but the reduced value stays in place, and there is no false pause.

The hearts take-off program lines work until the volume specified in the take-off program has been collected.

If during the hearts take-off the temperature of the vapor or column section sensors goes beyond the limits specified in the settings, Samovar pauses the take-off. This is the so-called "Start-Stop" mode. For more details, see the page [Hearts take-off](#).

- If the impurity detector and the "Reduce take-off rate when temperature rises" («Снижать скорость отбора при росте температуры») parameter are enabled in the settings, then when the vapor temperature rises Samovar reduces the take-off rate by the step set in the settings, and on a critical rise it pauses the take-off.

- If within one line the take-off is paused by temperature three times in a row, Samovar reduces the base rate of this line by 10%.

After the hearts take-off program lines are finished, Samovar moves on to the **tails take-off** lines, if the program has them. When the last program line finishes, Samovar stops the take-off pump, moves the rocker arm to position 0 and leaves the heating on — the column runs on self-reflux for 30 minutes (the message "Program finished. Self-reflux (30 min until auto-off)." («Выполнение программы завершено. Работа на себя (30 мин до автоотключения.)). During this time you can press "Next" («Далее») to finish the process at once, or add lines to the program. After 30 minutes Samovar switches the heating off, waits for the column to cool down and shuts off the water.

If you press the "Next" button while any program line is running, Samovar, after confirmation, moves on to the next line. You can also press the "Pause" («Поставить на паузу») button while the program is running, or use the [Universal button](#) again, and Samovar pauses the pump. To continue the take-off you need to press the "Continue" («Продолжить») button, and the program carries on. The "Reset process" («Сбросить процесс») button, after confirmation, stops the take-off and switches the heating off.

If the boiler temperature reaches the value set in the "Rectification, distillation and BK up to boiler temperature" («Ректификация, дистилляция и БК до температуры в кубе») field in the settings, Samovar finishes the program and switches the heating off with the message "Boiler maximum temperature limit. Program finished." («Лимит максимальной температуры куба. Программа завершена.»).

Questions and answers. FAQ

How is it determined that the column is stabilized?

Samovar determines by itself that the column is stabilized. After the boost ends, if the vapor sensor reading does not change by more than 0.1° for 6 minutes, the column is considered stabilized.

Can I start the take-off before the column is stabilized?

Yes. Samovar does not forbid it, but the web interface asks you to confirm the start.

Will Samovar send a message that it is time to start taking the product? Does it have an alcohol quality sensor?

No. You make the decision to start the hearts take-off yourself. [Here](#) it is explained in plain terms how Samovar works in rectification mode.

Take-off program in Rectification mode

Samovar 7.00: the [program calculation calculator](#) now calculates equipment parameters and recommended speeds, with an "Apply recommendations" («Применить рекомендации») button; before installing the program, check the power units: volts or watts. A "Clear program" («Очистить программу») button has been added to the "Program" («Программа») tab. When the program is installed, it is checked: the first line must set an absolute voltage (power), and every take-off line must have a volume or a transition temperature set.

The "Program" («Программа») tab

After Samovar is switched on or rebooted, the "Program" («Программа») tab shows the last installed take-off program; Samovar keeps it in its memory. On a new device an example is loaded (three H-B-H lines of 450 ml each), which must be [adjusted](#) to your needs.

The main take-off program window contains the following items:

Программа отбора:

Рассчитать программу 11

№	Тип	Выход	Скорость	Ёмкость	Напр	Время
01	Головы	100	0.1	1	120	01:00
02	Тело	500	0.5	2	100	01:00
03	Предзахлеб	3000	1.2	3	0	02:30
04	Хвосты	300	1	4	100	00:18

Итого: 04:47 8

9 Описание программы отбора в текстовом формате для сохранения в облаке:

10 Программа отбора:

Установить программу 12

Очистить программу 13

The "Program" («Программа») tab in Rectification mode

- 1 Тип («Тип»)**
The take-off mode for the line: Heads («Головы»), Hearts («Тело»), Pre-flooding («Предзахлеб»), Tails («Хвосты»), Pause («Пауза»).
- 2 Output («Выход»)**
The take-off volume in milliliters. For a "Pause" line, the pause time in seconds.
- 3 Speed («Скорость»)**
Take-off rate, liters per hour.
- 4 Container («Ёмкость»)**
The number of the container to take off into (0-10). Position 0 is the home position.
- 5 Volt. («Напр.»)**
Regulator voltage (for SEM_AVR, power in watts). 0 means do not change; a small or negative number is an offset from the current value.
- 6 Time («Время»)**
The calculated execution time of the line.
- 7 Add / delete a line**
The plus adds an empty line below; the minus deletes the current line.
- 8 Total («Итого»)**
The total execution time of the program (without pauses).
- 9 Take-off program description («Описание программы отбора»)**
A collapsible field for free text that is sent to the cloud when the take-off starts.
- 10 Take-off program («Программа отбора»)**
A collapsible field with the program in text format. It is linked to the table.
- 11 Calculate program («Рассчитать программу»)**
Go to the program calculation calculator.

12 Install program («Установить программу»)
Checks the program and writes it to Samovar's memory.

13 Clear program («Очистить программу»)
Deletes all program lines after confirmation.

Column descriptions

No. - Program line number. Lines are numbered automatically as you add or delete them. The number of the line being executed is shown in the status line on the main "Rectification" («Ректификация») tab.

Type («Тип») - A drop-down menu with the name of the take-off mode: [Heads \(«Головы»\)](#), [Hearts \(«Тело»\)](#), [Pre-flooding \(«Предзахлеб»\)](#), [Tails \(«Хвосты»\)](#), [Pause \(«Пауза»\)](#), Lua. Depending on the choice, [different logic](#) is applied. The lines in the table are colored by type.

Output («Выход») - The value is set in milliliters: how much to take off before moving on to the next program line. For a "Pause" line this field sets the pause time in seconds; for a "Lua" line it sets the stage timeout in seconds.

Lua - a line controlled by a Lua script: choosing this type opens the "Lua stage" («Lua-этап») window, where you select a .lua file from Samovar's memory, the timeout in seconds and the script parameters. Speed, container and voltage are not set for such a line, and the first-line voltage requirement does not apply to it. More details: [About the Lua scripting language](#).

Speed («Скорость») - Take-off rate of the peristaltic pump. The value is set in liters per hour.

Container («Ёмкость») - The number of the container to take the product off into. Numbering starts from zero (0-10). That is, the "home" position of the servo rocker arm is container No. 0.

Volt. («Напр.») - The voltage in volts that must be set on the [voltage regulator](#). Zero means do not change the current value; a small or negative number is an offset from the current value (see [Editing the take-off program](#)).

Note. For the SEM_AVR regulator the value is set in watts, and the column is called "Power" («Мощн.»).

Time («Время») - The time it will take to execute a particular program line (volume divided by speed). (since version 5.25)

The table does not show one more line field, the **transition temperature** (the fifth value in the text format). It can be set only in the "Take-off program" («Программа отбора») text field; the value is preserved when you edit the table. If it is 0, it is not used. If a number from 0 to 20 is set, the line ends when the steam temperature exceeds the steam temperature at the moment the line started plus this number. If 20 or more is set, the line ends when the steam temperature exceeds this value. The move to the next line happens on whichever condition (volume or temperature) is met first.

Program

The program, in table form, that the take-off will follow.

The take-off program is executed line by line and ends after the last line has been executed. Lines can be added and deleted. Clicking the plus (+) at the end of a line adds the next empty line; clicking the minus (-) deletes the current line. Once the program has been installed, the action cannot be undone.

The required parameter is set at the intersection of a row and a column.

Each line describes what Samovar should do at a given moment. The run time of a line is determined either by the volume of liquid the pump must move at the set speed, or by the transition temperature, or by an explicitly set time (for a "Pause" line). The table is directly linked to the "Take-off program" text field.

The line being executed at the moment is highlighted in dark red italics, and lines already passed are in italics. While you are editing a table cell, updating the table with data from Samovar is postponed; the "Data updated" («Данные обновилась») icon next to the heading tells you this.

Note.

The maximum number of program lines that Samovar accepts for execution is 20. A program with more lines will not be installed (the message "too many lines" («слишком много строк»)).

Total time

The "Total" («Итого») field is the execution time of the whole take-off program (not counting "Pause" lines).

Take-off program description

Free text (up to 250 characters). Any information about this take-off program.

▼ Описание программы отбора в текстовом формате для сохранения в облаке:

Описание программы отбора:

Произвольный текст.

The "Take-off program description" («Описание программы отбора») field

A collapsible text field. You can enter information related to a particular take-off program. The description is stored in Samovar together with the program by the "Install program" («Установить программу») command, and is sent to the cloud once when the session starts with the "Start take-off" («Начать отбор») command. Afterwards the text can be edited only in the cloud. More details: [logs and charts in the cloud](#).

Take-off program

The program, in text format, that the take-off will follow.

▼ Программа отбора:

Текст программы отбора:

```
H;100;0.1;1;0;120  
B;500;0.5;2;0;100  
C;3000;1.2;3;0;0  
T;300;1;4;0;100
```

- **Тип отбора**

Н - головы (Heads)

В - тело (Body)

Р - пауза (Pause)

Т - хвост (Tail)

- **Объем отбора (мл)**

- **Скорость отбора (л/ч)**

- **Номер емкости для отбора**

- **Температура отбора**

Если значение Температуры равно 0 - не используется.

Если значение Т указано и больше 20°C, то при Т пара больше, чем указанная температура, произойдет переход на следующую строку программы.

Если значение Т указано больше 0 и меньше 20°C, то при Т пара больше, чем сумма Т пара, которая была запомнена при переходе на эту строку программы и Т в программе, произойдет переход на следующую строку программы.

- **Напряжение отбора**

Пример программы отбора:

```
H;100;0.1;1;0.00;120  
P;120;0.00;0;0.00;140  
B;5000;0.75;2;0.00;90
```

The "Take-off program" («Программа отбора») field in text format

A collapsible text field. The take-off program is shown in text format: six values separated by semicolons: type; volume (ml); speed (l/h); container number; transition temperature; voltage (power). Below the field you will find an explanation of the values and an example program. The text can be copied and saved locally as an ordinary text file. It is directly linked to the table.

The data will be copied to the cloud once, on the "Start take-off" command. The text cannot be edited in the cloud.

Calculate program («Рассчитать программу»)

Opens a separate page with the [program calculation calculator](#) for the take-off, based on the volume of low wines and the percentage of each fraction. If the table has unsaved edits, Samovar asks you to confirm the move.

Install program

Copies the program to Samovar. When pressed, it checks the program and installs it in Samovar's memory for later execution. If the program fails the check (wrong line format, more than 20 lines, the first line does not set an absolute voltage, a take-off line has neither volume nor temperature set), it is not installed, and the reason is shown in a message with the line number.

Clear program

After confirmation, deletes all program lines from Samovar's memory.

Editing the Take-off Program

How to create and change the take-off program.

There are several ways to create a take-off program:

Directly in the table, on the "Program" («Программа») tab;

In the "[Calculation calculator](#)" («Калькулятор расчета»);

In the "Calculation AS-3.xlsx" («Расчет AC-3.xlsx») file, which is in the root of the folder with the Samovar sketch.

The first way. Edit the program directly, by adding/deleting rows in the table and setting the parameters you need, according to your preferences. Some parameters can be chosen from drop-down menus, and some you have to set yourself, observing certain conditions.

The program line being executed at the moment is highlighted in dark red italics, and lines already executed are in italics.

Программа отбора:

Рассчитать программу

№	Тип	Выход	Скорость	Ёмкость	Напр	Время	
01	Головы	450	0.07	0	160	06:25	+ -
02	Головы	150	0.1	1	165	01:29	+ -
03	Предзахлеб	300	0.5	2	170	00:36	+ -
04	Предзахлеб	3000	1.2	3	0	02:30	+ -
05	Предзахлеб	3000	1.2	4	0	02:30	+ -
06	Тело	3000	1	5	5	03:00	+ -
07	Хвосты	1000	0.9	6	7	01:06	+ -

Итого: 17:38

► Описание программы отбора в текстовом формате для сохранения в облаке:

► Программа отбора:

Установить программу

Очистить программу

The program table during take-off: line No. 3 is being executed

It is best to edit the first line directly in the line itself:

Type («Тип») - from the drop-down menu choose "[Heads](#)" («Головы»)

Output («Выход») - set the take-off amount in milliliters.

Rate («Скорость») - liters per hour

Container («Ёмкость») - the number of the container into which the take-off goes. Remember that counting starts from position No. 0 (zero)

Volt. («Напр.») - set the real value your [regulator](#) works with. This is the voltage Samovar will command the regulator to set after boost is finished. For the SEM_AVR regulator the column is called "Power" («Мощн.») and is set in watts.

Time («Время») - calculated automatically. At the bottom of the column it is added up into a grand total.

(+) plus - adds a line below.

(-) minus - deletes the current row.

The second and following lines.

You can edit the remaining lines of the preset program and/or delete them by pressing (-), add a line below (+) and enter values in it.

Editing is done the same way as for the first line, but in the voltage settings you can now apply other conditions as well:

You can set the required value explicitly, for example 160.

Or:

- if the voltage is set to 0 (zero), the regulator is left alone: the actual voltage being applied at the moment stays.

For example: in the "[Pre-flooding](#)" («Предзахлеб») mode, 175 volts was set in the previous line, but by the time of the transition to the line with 0.00 set, Samovar had raised the voltage to 180 volts. Exactly 180 volts will be applied to the new take-off line.

- if a value of no more than 40 volts is set, this voltage is added to the current voltage.

For example: if 160 volts was set in the previous line (or Samovar itself raised the voltage to 160 volts), then when executing a line with 5 volts set, Samovar adds 5 volts to the actual voltage (160+5).

- if a negative value is set, this voltage is subtracted from the current one.

For example: if 175 volts was set in the previous line (or Samovar itself raised the voltage to 175 volts), then when applying a line with -5 (minus 5) volts set, Samovar subtracts 5 volts from the actual voltage (175-5).

For the SEM_AVR regulator (watts) the boundary between a correction and an absolute value is 400 W.

These voltage setting conditions apply to all take-off modes, including the "Pause" («Пауза») line.

Mandatory condition: in the first line of the program the voltage must be set as a real value (more than 40 volts or 400 watts). Otherwise the program will not be set and will not start, and the message will state the reason: without this, the column would stay at full power after boost.

When the program is edited in the table, it is duplicated in text format in the "Take-off program" («Программа отбора») field

▼ Программа отбора:

Текст программы отбора:

H;450;0.07;0;0;160
H;150;0.1;1;0;165
C;300;0.5;2;0;170
C;3000;1.2;3;0;0
C;3000;1.2;4;0;0
P;3000;1.5;0.5

• Тип отбора

H - головы (Heads)
B - тело (Body)
P - пауза (Pause)
T - хвост (Tail)

• Объем отбора (мл)

• Скорость отбора (л/ч)

• Номер емкости для отбора

• Температура отбора
Если значение Температуры равно 0 - не используется.
Если значение T указано и больше 20°C, то при T пара больше, чем указанная температура, произойдет переход на следующую строку программы.
Если значение T указано больше 0 и меньше 20°C, то при T пара больше, чем сумма T пара, которая была запомнена при переходе на эту строку программы и T в программе, произойдет переход на следующую строку программы.

• Напряжение отбора

Пример программы отбора:

H;100;0.1;1;0.00;120
P;120;0.00;0;0.00;140
B;5000;0.75;2;0.00;90

The same program in the "Take-off program" («Программа отбора») text field

This field and the table are linked. A change in the table is immediately shown in this field, and accordingly, if you change a value in this field, it will change in the table too.

The fifth value of the line, the transition temperature, is not shown in the table and can be set only in this field (see [Take-off Program of the Rectification Mode](#)). It is not lost when the table is edited.

This lets you copy the program to a text file and save it, and in turn paste any of your programs into this field by copy and paste, including one from the cloud. You can paste a program used for an earlier distillation run and, if needed, quickly adjust it in the table. This function saves you from having to type the program again every time.

After the take-off program is created, it must be set in Samovar by pressing "Set program" («Установить программу») on the same tab. Samovar checks the program and, if it finds an error, shows a message with the line number; the program is not changed in that case.

Give the "Set program" command after any adjustment of the program if you want Samovar to accept the changes.

Attention!

On the "Start take-off" («Начать отбор») command, Samovar starts executing the first line of the program. If, during execution, you change the program line that is currently being executed and give the "Set program" command, nothing will happen. Samovar will continue executing the currently running program line with the volume, rate, container and voltage that were set at the moment it started.

That is, if the operator decides to change the program when Samovar has already started take-off, the changed take-off program will be accepted by Samovar for execution only in the part whose execution has not yet begun.

Tip.

If you want to change the program by altering the process being executed at the moment, there are several ways to do it:

- Add an empty line right after the one being executed, set the values you need in the new line (taking into account the volume already collected), give the "Set program" command and on the main tab press the "Next" («Далее») button (this is what the "Start take-off" button is called during take-off). Samovar will ask you to confirm the transition to the next line.

- Switch to [manual control](#). Samovar will then continue executing the program as far as the take-off amount and container are concerned. The take-off rate can be changed in the "Rate, L/h" («Скорость, л/ч») field of the "Regulator" («Регулятор») card on the main tab, and the voltage in the voltage field of the same card. The changes apply until the transition to the next line.

Program Calculator

Samovar 7.00: the program calculator now includes the equipment parameters, the calculation of flooding, working power and rates, the "Apply recommendations" («Применить рекомендации») button and a check of the V/W units. Check the selected units before transferring the result to the program.

On the "Program" («Программа») tab there is a "Calculate program" («Рассчитать программу») key; press it to go to the page with the program calculator, which contains an example of calculating a program based on the volume of low wines and their ABV.

This calculator is intended for quickly composing a take-off program based on the percentage ratio of the fractions and immediately setting the program for the upcoming run.

Samovar v. 7.00

Расчет программы отбора **Фрукты** ¹ диаметр колонны **2 дюйма** ²

ПАРАМЕТРЫ ОБОРУДОВАНИЯ

Диаметр выбран выше (1.5 / 2 / 3 дюйма). Высота: **1 м** | Насадка: **80%**

Мощность ТЭНа для расчета: **3000** ³ Вт

Захлёб: **171 В**

Рабочая: **148 В**

Макс.отбор: **500 мл/ч**

Головы: **0.330 л/ч** | Тело: **0.600...1.000 л/ч** | Тарелок: **39.2**

Стратегия "Фрукты": Плёночный режим (низкое ФЧ) для сохранения аромата.

ПРИМЕНИТЬ РЕКОМЕНДАЦИИ

Программа в вольтах, как текущий регулятор.

Объем спирта-сырца **12.25** ⁴ л | Содержание АС в с-с **34** ⁵ % | Выход спирта **94** ⁶ %
 Объем голов **8** ⁷ % | Объем хвостов **5** ⁸ % | Кубовый остаток **6.84** ⁹ %

Информация о параметрах программы отбора

Итоги программы отбора

Головы

От АС Объем
8% **354 мл**
 Время
2 ч 54 мин

По строкам: **100%** — распределено полностью

Тело

От АС Объем
87% **3855 мл**
 Время
4 ч 02 мин

По строкам В+С: **100%** — распределено полностью

Хвосты От АС: **5%** Объем: **88 мл** Время: **0 ч 39 мин** По строкам: **20%** — информационно

Итого по программе Отбор: **4297 мл** Время: **7 ч 47 мин** Паузы: **0 ч 11 мин**

№	Ёмкость	Скорость	Процент	Тип	Напр	Выход	Время	
01	0	0.07	30	Головы	135	0.106	01:30	+ -
02	0	0.1	20	Головы	0	0.071	00:42	+ -
03	1	0.2	20	Головы	0	0.071	00:21	+ -
04	2	0.3	20	Головы	0	0.071	00:14	+ -
05	3	0.4	10	Головы	140	0.035	00:05	+ -
06	0	300	0	Пауза	145	300	00:05	+ -
07	4	1	10	Тело	140	0.385	00:23	+ -
08	4	0.9	40	Тело	0	1.542	01:42	+ -
09	0	200	0	Пауза	173	200	00:03	+ -
10	0	200	0	Пауза	170	200	00:03	+ -
11	5	1.1	40	Предзахлеб	3	1.542	01:24	+ -
12	6	0.9	5	Предзахлеб	0	0.193	00:12	+ -
13	6	0.7	3	Предзахлеб	0	0.116	00:09	+ -
14	6	0.5	2	Тело	140	0.077	00:09	+ -
15	7	0.2	10	Хвосты	140	0.044	00:13	+ -

Take-off program calculation calculator

- 1 Take-off program calculation («Расчет программы отбора»)**
Raw material type: Fruit, Grain or Sugar. Loads the corresponding template.
- 2 Column diameter («Диаметр колонны»)**
1.5, 2 or 3 inches - used to calculate the equipment parameters.
- 3 Heating element power for calculation («Мощность ТЭНа для расчета»)**
Taken from the settings (heating element resistance and mains voltage). Needed for the calculation of flooding and working power.
- 4 Low wines volume («Объем спирта-сырца»)**
The volume of low wines in liters.
- 5 Pure alcohol content in low wines («Содержание АС в с-с»)**
The content of pure alcohol in the low wines, in %.
- 6 Alcohol yield («Выход спирта»)**
The predicted strength at the column outlet, in %.
- 7 Heads volume («Объем голов»)**
The predicted yield of heads, in %.
- 8 Tails volume («Объем хвостов»)**
The predicted yield of tails, in %.
- 9 Boiler residue («Кубовый остаток»)**
Calculated automatically: the residue in the boiler after distillation, in l.
- 10 Calculation table («Таблица расчёта»)**
The program lines: container, rate, percent of the fraction, type, voltage (power), and the calculated yield and time.
- 11 Total take-off time («Итоговое время отбора»)**
The time required for rectification with the set parameters.
- 12 Save calculation («Сохранить расчет»)**
Downloads the programbackup.txt file with the program lines.
- 13 Load calculation («Загрузить расчет»)**
Loads a previously saved calculation file into the calculator.

Three template files are stored in the Samovar file system (together with the web interface):

program_shugar.txt,
program_grain.txt,
program_fruit.txt.

They contain templates for calculating take-off programs for sugar, grain and fruit. These are what is loaded as the base calculations. The same files are in the tools folder of the sketch. In the templates, the power of each line is written both in volts and in watts (for the reference heating element of 3480 W / 230 V); the calculator chooses which column to use according to the regulator type.

In the "Take-off program calculation" («Расчет программы отбора») drop-down menu, select the type of raw material (Fruit, Grain, Sugar), and next to it the "Column diameter" («Диаметр колонны») (1.5, 2 or 3 inches).

Equipment parameters

This block shows the packing height and packing density set on the "Settings" («Настройки») tab in the "Equipment (column) parameters" («Параметры оборудования (колонны)») section ("Diameter" («Диаметр»), "Packing height" («Высота насадки»), "Packing density" («Плотность насадки»), and the "Heating element power for calculation" («Мощность ТЭНа для расчета») field, which is calculated from the heating element resistance and the mains voltage set in the settings. From this data the calculator computes the flooding power, the working power, the maximum take-off (ml/h), the recommended rates for heads and hearts, and the number of theoretical plates.

The "Apply recommendations" («Применить рекомендации») button transfers the calculated rates and powers to the program lines. If the recommended power is greater than the power of your heating element, the recommendations are not applied and the reason is shown in a message. Below the button there is a hint saying in which units (volts or watts) the power is written in the loaded template.

Calculation data

For the calculation you need to enter your data: "Low wines volume" («Объем спирта-сырца») (l), "Pure alcohol content in low wines" («Содержание АС в с-с») (%), and set the expected ABV at the column outlet - "Alcohol yield" («Выход спирта») (%). You can also set the percentage of heads and tails for your low wines ("Heads volume" («Объем голов»), "Tails volume" («Объем хвостов»), depending on the type of raw material used). The values "Boiler residue" («Кубовый остаток»), "Yield" («Выход») in each line, "Time" («Время») and "Total take-off time" («Итоговое время отбора») are calculated automatically. Below the table the totals by fraction are shown: how much pure alcohol, volume and time fall on the heads, hearts and tails. The columns of the calculation table are: No. («№»), Container («Ёмкость»), Rate («Скорость») (l/h; for a "Pause" («Пауза») line - the pause time in seconds), Percent («Процент») (the share of the volume of a fraction of the given type), Type («Тип»), Volt. («Нап.») (or Pow. («Мощн.»)), Yield («Выход»), Time («Время»). Lines are added and removed with the plus and minus, as on the "Program" («Программа») tab.

When you hover over a field label, hints appear on how to fill it in correctly or under which conditions to use it. For example (the "Information about take-off program parameters" («Информация о параметрах программы отбора») hint):

When working with low wines from sugar wash

Column of maximum height, tightly packed packing

Take-off - heads in film mode, hearts in maximum separation mode (pre-flooding)

Heads - 10%-15%, tails - 5%

When working with low wines from fruit

Column one meter high, loose packing

Start take-off in film mode

After taking off the heads, increase the separation, but do not bring it to pre-flooding

Heads - 8%, tails - 5%

Alcohol yield - 94%

When working with low wines from grain

Column one meter high, loose packing

Start take-off in film mode

After taking off the heads, increase the separation, but do not bring it to pre-flooding

Heads - 6-10%, tails - 5-7%

Alcohol yield - 94%

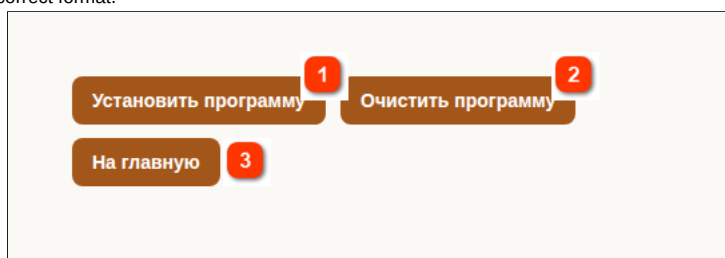
These calculations can be used as is by loading them into Samovar for execution. To do this, give the "Set program" («Установить программу») command. Before sending, the calculator checks the program and the power units: if they do not match the regulator, the message "The program power units do not match the regulator. Press "Apply recommendations"." («Единицы мощности программы не совпадают с регулятором. Нажмите «Применить рекомендации»») will appear.

After the message "Program saved." («Программа сохранена»), the program will appear on the "Program" («Программа») tab of the main page, and you can start running it right away.

You can adjust it as you see fit in the calculator and again give the "Set program" command. And then start the run.

The calculation can be saved to a local folder. With the "Save calculation" («Сохранить расчет») command, a file named programbackup.txt is saved to your browser's Downloads folder. The program lines are written to the file with the power in both volts and watts, so the heating element power must be set in order to save.

Load the saved calculation into the calculator (the "Load calculation" («Загрузить расчет») button) and then transfer it to the "Program" tab. The file is checked on loading: no more than 20 lines, correct format.



Calculator buttons

1 Set program («Установить программу»)
Checks the calculation and writes the program to Samovar.

2 Clear program («Очистить программу»)
Deletes the program from Samovar memory after confirmation.

3 To main («На главную»)
Leave the calculator without setting the program.

The "Clear program" («Очистить программу») button deletes the program from Samovar memory after confirmation. To leave the calculator without saving and setting the program, press "To main" («На главную»).

Note.

A calculation made in the calculator can either be transferred right away to the "Program" tab in Samovar, or saved as a calculation.

A previously made calculation can be loaded into Samovar for execution only through the calculator.

Warning.

Keep in mind that the calculator is intended for calculating a take-off program right before the run. If you reload the page, the calculation is not kept. The saved file stores only the program lines: the volume of low wines, the strength and the percentages of heads and tails are not written to it, so when a calculation is loaded the data fields keep their "default" values, and the calculator immediately recalculates the program from them. To make your saved calculation relevant again, enter your own values in the data fields.

For a preliminary calculation with subsequent saving, it is recommended to use the file "Пакет AC-3.xlsx" ("AS-3 calculation") - an Excel file for generating a program. You enter the take-off parameters - the volume of low wines, the strength, the amount of heads, the amount of tails - and the output is a take-off program for Samovar. At any moment, copy the ready program from the file, paste it into the text field on the "Program" tab, give the "Set program" command and you can start the take-off. The file is downloaded with every release and, after unpacking the archive, is located in the root of the Samovar folder.

Heads take-off

To run a heads take-off program, either select the "Heads" («Головы») mode in the drop-down menu of the "Program" («Программа») tab, or create a line manually in the text field.

Put the Latin letter H (Heads) at the beginning of the line.

A take-off program line will look something like this:

H;250;0.10;1;0.00;120

All values must be separated by a semicolon.

The first parameter (H) is the take-off type. Here it is H (heads).

The second parameter (250), separated from the take-off type by a semicolon, defines how much liquid must be taken off before moving on to the next line. In our case it is 250 milliliters.

The third parameter (0.10), after the semicolon, is the rate at which Samovar must take off the given volume; for the first line in the example it is 100 milliliters per hour, or 0.1 liters per hour.

The fourth parameter (1) defines the container the liquid will be collected into, container No. 1 in the example.

The fifth parameter (0.00) – the transition temperature. 0 — not used, the line ends by volume. A value from 0 to 20 — the line ends when the vapor temperature exceeds the vapor temperature at the moment the line started plus this value. A value of 20 or more — the line ends when the vapor temperature exceeds it. Whichever condition occurs first applies: volume or temperature. At least one of them must be set, otherwise the program will not be accepted. This field is not shown in the table of the "Program" («Программа») tab — it can only be set in the text field.

The sixth parameter (120) defines the voltage Samovar must set on the regulator during this particular line, 120 volts (for the SEM_AVR regulator — power in watts). In the first line of the program it must be a real value: more than 40 volts (400 watts). In the other lines, 0 — do not change the voltage, a number no greater than 40 — add to the current value, a negative number — subtract.

The heads take-off algorithm is such that Samovar does not take into account the setpoint or the difference between the column section temperature sensor and the vapor temperature sensor. Even if at some stage the delta becomes negative, Samovar will continue the take-off. The previously fixed hearts take-off temperatures are reset when a heads line starts. On heads, the [impurity detector](#) only observes: it shows the temperature trend but does not change the rate and does not pause — a rise in vapor temperature on heads is a normal process. All other parameters (reflux level sensor, TCA temperature, water flow rate, etc.) will be monitored and handled by Samovar. When the reflux level sensor triggers, Samovar sounds a signal, shows the message "Flooding sensor triggered!" («Сработал датчик захлёба!») and lowers the voltage by one step (see [Pre-flooding](#)).

If the "Use I2C pump above the pasteurization section" («Использовать I2C-насос над ЦП») option is enabled in the settings and such a pump was detected when Samovar started, then on heads lines it becomes the main take-off pump and runs at the rate set in the line.

Note: *The take-off rate and regulator voltage of the program line currently being executed can be changed at any moment in [manual mode](#) — in the "Regulator" («Регулятор») card on the main tab. The changes apply until the next line starts.*

Hearts take-off

Samovar 7.00: when the impurity detector is enabled, a warning reduces the take-off rate, and a critical state pauses the process. After a manual resume, the automatic reaction is temporarily not applied for about a minute; repeated stop-pauses can lower the base rate of the current line.

When Samovar moves on to running the hearts take-off program, it remembers the temperatures of the vapor sensor and of the column section sensor. From then on, Samovar relies on these temperatures during operation. The fixed temperatures are shown in the status line on the main tab, and the message "New hearts T: vapor = ..., section = ..." («Новые Т тела: пар = ..., цэпра = ...») appears in the messages.

Note: In some cases you may need to overwrite the fixed temperatures. For example, if you increased the voltage supplied to the heating element while the program was running, this changes the temperature in the column section and may push it past the setpoint, so the take-off is paused. To avoid this, at any moment you can press the "Fix hearts take-off T" («Зафиксировать Т отбора тела») button in the "Regulator" («Регулятор») card on the main tab "[Rectification](#)" («Ректификация»), after which the current temperatures in the column section and the vapor will be used. The button works only on the Hearts («Тело»), Pre-flooding («Предзахлеб») and Pause («Пауза») lines.

Warning!

When moving to a new Hearts or Pre-flooding line, Samovar acts as follows:

- if the line sets its own voltage (not 0), the previous temperatures are reset, and new ones are fixed after the vapor temperature settles at the new power, but no later than 10 minutes. Until then, setpoint pauses are not applied;
 - if the voltage in the line is 0, the temperatures stay the same, and if there are none yet (the first hearts line after heads or after a "Pause" line), they are fixed immediately, as with a manual press of the "Fix hearts take-off T" button.
- If during hearts take-off (the Hearts or Pre-flooding program lines) the temperature of the vapor and/or column section sensors goes beyond the setpoints specified in the settings, Samovar will pause the take-off.

The setpoints are set on the "Settings" («Настройки») → "Temperature" («Температура») tab: "Vapor sensor overshoot before the dephlegmator" («Вылет за уставку датчика пара перед дефлегматором») and "Column section sensor overshoot at 2/3 of the column" («Вылет за уставку датчика в цэпре на 2/3 колонны») (°C), as well as "Take-off delay when vapor T exceeds the setpoint" («Задержка отбора при выходе Т пара за уставку») and "Take-off delay when section T exceeds the setpoint" («Задержка отбора при выходе Т в цэпре за уставку») (seconds). If the overshoot is 0, control by that sensor is disabled. When the sensor temperature rises to the fixed one plus the setpoint, Samovar stops the pump, sounds a beep and shows the message "Pause by vapor T" («Пауза по Т пара») or "Pause by section T" («Пауза по Т цэгри»). The take-off resumes (message "Resuming take-off after automatic pause" («Продолжаем отбор после автоматической паузы»)) once the set delay has passed and the temperature has dropped below the threshold (with a 0.07° margin so the pause does not chatter at the boundary). If three such pauses (from the sensors or from the detector) pile up during one line, Samovar lowers the base rate of this line by 10% and reports it.

You can cancel pausing the take-off when temperatures exceed the setpoint: instead of pausing, Samovar fixes new hearts temperatures.

The take-off will not be paused provided that:

For a Hearts or Pre-flooding line.

It is the first Hearts or Pre-flooding line after a Heads line ("Pause" lines between them do not count).

For example:

- . Heads («Головы»)
- . Hearts («Тело»)
- . Pre-flooding («Предзахлеб»)
- . Hearts
- . Tails («Хвосты»)
- . etc.

While running line no. 2, Samovar will not pause the take-off but will record new vapor and column section temperatures. While running lines no. 3 and no. 4, the take-off will be paused.

For a Pre-flooding line

If a Pre-flooding line is anywhere in the take-off program and is followed by two Hearts or Pre-flooding lines in any combination (the first of them may also be a "Pause" line).

For example:

- . Heads
- . Hearts
- . Pause («Пауза»)
- . Pre-flooding
- . Pre-flooding
- . Hearts

If temperatures exceed the setpoint while line no. 2 (the first hearts line after heads) and line no. 4 (pre-flooding followed by two more hearts lines) are running, Samovar will not pause the take-off but will remember the new vapor and column section temperatures. While running lines no. 5 and no. 6, the take-off will be paused if temperatures exceed the setpoint.

The automatic temperature rise is limited: within one line, Samovar raises them by no more than 0.3° from the first vapor temperature fixed in that line. Above this limit, exceeding the setpoint causes a normal pause.

The function is enabled in the Samovar_ini.h file. The line

```
#define USE_BODY_TEMP_AUTOSET
```

//use automatic correction of the hearts T for the first hearts or pre-flooding take-off program after heads, and also for a pre-flooding program if it comes before the second-to-last hearts or pre-flooding take-off program

In the standard build the function is enabled. You can disable it in the user_config_override.h file, and the rise limit is set by the line `#define BODY_TEMP_AUTOSET_MAX_RISE 0.3f` in Samovar_ini.h.

- If the "Use pressure correction" («Использовать корректировку давления») option was enabled in the settings, Samovar converts the temperature sensor readings to a pressure of 760 mm Hg using the pressure sensor. The fixed hearts take-off temperatures then do not depend on changes in atmospheric

pressure during distillation: when the pressure drops, the actual boiling temperature falls, but the converted value stays in place, and there will be no false pause.

Impurity breakthrough detector

It is enabled in the settings with the "Use detector" («Использовать детектор») checkbox (without it the detector does not work at all). On Hearts and Pre-flooding lines the detector monitors the rate of temperature rise (°C/min) - using the vapor sensor, and when the boiler temperature is above 92.5°, using the column section sensor. The detector picks the trigger threshold itself, after measuring its own temperature noise during a calm period. The detector state is shown in the card on the main tab: "stable" («стабильно»), "rising" («рост») (or "reducing rate" («снижаю скорость»)), "breakthrough" («проскок»).

- When the rise is above the threshold (warning), if the "Reduce take-off rate when temperature rises" («Снижать скорость отбора при росте температуры») checkbox is enabled, Samovar reduces the take-off rate step by step by the "Rate reduction step" («Шаг снижения скорости») (5% by default, 20% maximum), but not below 70% of the line's base rate. When the rise stops, the rate returns to the base rate in the same step-by-step way. Without this checkbox the detector only reports the rise.

- On the first hearts line after heads, the detector warning fixes new hearts temperatures (within the same 0.3°) instead of reducing the rate.

- On a critical rise (2.5 times above the threshold, two measurements in a row), Samovar pauses the take-off, sounds a beep and shows the message "Detector: Critical trend! Take-off paused" («Детектор: Критический тренд! Пауза отбора»). The take-off resumes by itself when the sensor delay has passed and the rise has stopped (message "Detector: Resuming take-off after pause" («Детектор: Продолжаем отбор после паузы»)).

- For the first 30 seconds after a line starts, and for a minute after a manual "Continue" («Продолжить»), the detector does not intervene. On the first hearts line after heads it waits for the vapor temperature to stabilize (up to 10 minutes, status "Waiting for vapor to stabilize" («Жду стабилизации пара»)).

On heads the detector only observes; on tails it works with a threshold 1.2 times higher.

Note:

In hearts take-off mode the reflux level sensor also works, and if flooding threatens, on the sensor's command Samovar will lower the voltage supplied to the heating element by one regulator step every 40 seconds until the sensor stops giving commands (see [Pre-flooding](#)). The lowered voltage remains the working one until the operator intervenes or until a program line that sets its own voltage is reached.

The lines of the hearts take-off program run until the volume specified in the take-off program is collected or the transition temperature (the fifth value of the line) is reached, after which Samovar moves on to the next line.

When the last line of the program finishes, Samovar stops the take-off, moves the sampling unit's rocker arm to position 0 and keeps the column in "self-reflux" («работа на себя») mode for 30 minutes, then switches off the heating and, when the column has cooled down, switches off the water. If the maximum boiler temperature set in the settings is reached earlier, Samovar finishes the program and switches off the heating immediately.

Note: The take-off rate and the regulator voltage of the program line currently running can be changed at any moment in [manual mode](#) - in the "Regulator" card on the main tab. A rate set manually becomes the base rate for the detector.

Pre-flooding

The "Pre-flooding" («Предзахлеб») program

Automatic holding of the column in the pre-flooding state. This mode works the same way as the [hearts take-off](#) mode, but with the additional function of holding the column in the pre-flooding state. Samovar raises and lowers the voltage on the regulator by itself. Manual voltage adjustment is possible as a supplement to the automatic one and takes priority.

The mode requires a reflux level sensor and the "Use the reflux level sensor during rectification" («Использовать датчик уровня флегмы при ректификации») option enabled in the settings.

How it works.

Warning. *The time intervals are the same for all regulators, while the power adjustment amounts differ. For each regulator model a constant called the "adjustment step" is used.*

The automatic voltage drop is 2 steps, and the increase is 1 step.

At the start of the "Pre-flooding" program line, Samovar sets the voltage specified in the line and waits 4 minutes. If the reflux level sensor has not triggered during this time, it raises the voltage by 1 step. Then it waits 4 minutes again and, if the sensor still has not triggered, raises the voltage again, and so on.

When the sensor triggers, Samovar remembers the current voltage and makes the first drop of 2 steps (the message "Lowering the voltage from ..." («Напряжение снижаем с ...»)).

It waits 40 seconds, ignoring sensor triggers, and then, if the signal is still present, drops the voltage again and waits another 40 seconds. The drops continue until the reflux sensor stops triggering.

48 seconds after the last sensor trigger, Samovar raises the voltage to the previously stored voltage of the first sensor trigger minus 2 adjustment steps of the regulator in use (the message "Raising the voltage to ..." («Напряжение повышаем до ...»)).

Then it waits 4 minutes. If the sensor again does not trigger during this time, it raises the voltage by 1 step again, and so on until the sensor triggers.

The cycle starts to repeat.

On every new cycle, the voltage at which the level sensor triggers for the first time is overwritten in Samovar's memory. The voltage returns to exactly this value, minus 2 regulator steps, after the sensor stops triggering. After a few iterations the column settles into a steady and almost constant pre-flooding mode.

When moving to the next line, if it is not "Pre-flooding", the counters and the stored voltage are reset. If a "Pre-flooding" line is followed by another such line, the cycle continues without a restart.

Note.

A reflux level sensor trigger is a period of 3 seconds during which the sensor LED stays continuously lit. If the LED blinks at intervals shorter than 3 seconds, it does not count as a trigger.

Note.

Samovar works with the actual voltage value at the moment of the event. This lets you change the voltage manually and quickly reach a stable result while tuning the mode.

For example: visually, the reflux level has already left the sight glass, but the sensor LED is still lit and Samovar keeps dropping the voltage. While we adjust the sensor, the power has fallen quite a lot. There is no point in waiting for Samovar to raise it automatically. Raise it manually on the regulator or from Samovar. Samovar will make its next step from the actual voltage.

To use this mode, select "Pre-flooding" in the drop-down menu, and the program line in the text field will look something like this:

C;2500;1.00;3;0.00;170

(C - hold the pre-flooding mode; 2500 - take-off volume in ml; 1.00 - take-off rate in l/h; 3 - take-off into container No. 3; 0.00 - transition temperature, 0 - not used; 170 - set the voltage on the heating elements)

Example with a Kvic regulator. The regulator step is 0.5 volt.

Suppose the take-off line in "Pre-flooding" mode is set to 170 volts

At the start of the take-off program on this line, Samovar applies 170 volts to the load through the regulator and waits 4 minutes. If the reflux level sensor has not triggered, it adds 0.5 volt and waits 4 minutes. If there is no trigger, it adds 0.5 volt again and waits 4 minutes again. And so on until the sensor triggers.

At the moment the sensor triggers (say, 174 volts), Samovar remembers the voltage and drops 1 volt. Then it waits 40 seconds, ignoring signals from the sensor. If the triggers continue, it drops another 1 volt, and so on at 40-second intervals, until the sensor stops giving a signal.

Say it drops to 168 volts. Then, 48 seconds after the last trigger, it raises the voltage to the recorded flooding voltage minus 1 volt (174-1=173) and again waits 4 minutes. If the sensor has not triggered, it adds 0.5 volt.

Samovar with an RMVK regulator. The regulator step is 1 volt

Suppose the take-off line in "Pre-flooding" mode is set to 170 volts

At the start of the take-off program on this line, Samovar applies 170 volts to the load through the regulator and waits 4 minutes. If the reflux level sensor has not triggered, it adds 1 volt and waits 4 minutes. If there is no trigger, it adds 1 volt again and waits 4 minutes again. And so on until the sensor triggers.

At the moment the sensor triggers (say, 174 volts), Samovar remembers the voltage and drops 2 volts. Then it waits 40 seconds, ignoring sensor triggers. If the triggers continue, it drops another 2 volts, and so on, until the sensor stops giving a signal.

Say it drops to 168 volts. Then, 48 seconds after the last trigger, it raises the voltage to the recorded flooding voltage minus 2 volts (174-2=172) and again waits 4 minutes. If the sensor has not triggered, it adds 1 volt.

Samovar with a SEM_AVR regulator.

Similar to working with the RMVK regulator, but this regulator sets power in watts rather than volts, and the steps are calculated as a percentage of the current power: the drop is 3%, the rise after the triggers end is up to the stored power minus 3%, and the scheduled increase every 4 minutes is 1%.

For example: the current power is 1800 W. When the sensor triggers, Samovar lowers it by 54 W (3%), and at a scheduled increase it adds 18 W (1%).

Note

Due to the way power is set, the actual value may differ slightly from the one set by Samovar (for example 1803 W instead of

1800). *This is not critical at all.*

If you need to change the reaction time of the reflux level sensor, it is set here:

In the file alarm.h:

//If we have already reacted - wait 40 seconds, because the process is inertial

*alarm_h_min = millis() + 1000 * 40;*

40 is the number of seconds after which the voltage is dropped again when the reflux level sensor triggers. When a reflux level sensor is installed, this works in any mode, because it belongs to the safety group that prevents column flooding: outside the "Pre-flooding" line, when the sensor triggers Samovar also sounds a beep and shows the message "Flooding sensor triggered!" («Сработал датчик захлёба!»).

in Samovar_pin.h:

#define WHLS_ALARM_TIME 3 //Seconds before the reflux level alarm triggers

Because of reflux fluctuations the sensor may blink; this parameter sets how many seconds of continuous level sensor triggering count as an actual trigger.

Applies only to pre-flooding operation:

in Samovar.h:

#define TIME_C 4 // Wait time in minutes for the pre-flooding mode program

– This is the time after which Samovar raises the voltage if the reflux level sensor has not triggered. The pause before restoring the voltage after the triggers end is one fifth of this time (48 seconds).

Tails take-off

The "Tails fractions take-off" («Отбор хвостовых фракций») program

Tails take-off. (Not to be confused with squeezing alcohol out of the tails.) This is a program for taking off the tails fractions specifically: aromatic waters or product with a raised content of fatty acids and so on.

To use this mode, put the Latin letter "T" (Tail) at the beginning of the program line. A tails take-off program line will look something like this:

T;2500;1.00;5;0.00;176

All functions work as in hearts take-off, except for accounting for the difference between the column section and steam temperatures and their going outside the set values. The hearts take-off temperatures recorded at the start of the tails line are reset.

Take-off will not pause on the steam and column section sensor setpoints. [The impurity detector](#), if it is enabled in the settings, keeps collecting history on tails and shows the temperature trend, but does not reduce the take-off rate and does not pause it.

That is, Samovar takes off reflux at the set rate into a given container at a fixed voltage until the set volume is collected or the transition temperature triggers (the fifth value of the line), and finishes the take-off at the boiler temperature set in the [settings](#) in the "Rectification, distillation and BK up to boiler temperature" («Ректификация, дистилляция и БК до температуры в кубе») field.

Note:

When the last program line finishes, Samovar stops the pump, moves the rocker arm to position 0 and holds heating for another 30 minutes: the column runs on itself. During this time you can press "Next" («Далее») to finish the process immediately or add lines to the program. After 30 minutes the heating switches off. If the boiler temperature reaches the limit set in the settings earlier, the program ends and the heating switches off immediately.

Pause

The "Pause" program

The Latin letter P stands for pause (Pause). This is what a take-off program line may look like:

P;120;0;0;0;172

It means: wait 120 seconds, set the voltage on the regulator to 172 volts:

When a line of type P runs, Samovar stops the take-off pump and waits for the time set in this line. The time is set in seconds in the second value of the line (in the table of the "Program" («Программа») tab it is the "Output" («Выход») column); it must be greater than zero. All other parameters are ignored (in particular, the vapor and column section temperatures), except the sixth parameter – the voltage. The voltage is applied by the same rules as in take-off lines: 0 — do not change, a small or negative number — an offset from the current value. The status on the main tab will show the pause and the remaining time.

Note:

The parameters that Samovar monitors for safety are not ignored in any operating mode.

You can use a pause, for example, if you need to raise (or lower) the regulator voltage by a large amount. Since the boiler and the column have quite a lot of inertia, this takes time. How long to wait you have to decide yourself, based on your experience.

Attention!

A special feature of this program line is that when it starts, Samovar resets in its memory the column section and vapor temperature values, if they were fixed earlier. If the next program line is Hearts or Pre-flooding and no voltage is set in it (0), then on moving to it the temperatures current at the moment of the transition are fixed. If the next line sets its own voltage, the temperatures are fixed later — when the vapor temperature settles at the new power (no longer than 10 minutes). That is, it is the same as if the operator wanted to set new values of the "Column section" («Царза») and "Vapor" («Пар») temperatures and physically pressed "Fix the hearts take-off T" («Зафиксировать T отбора тела») in the "Regulator" («Регулятор») card on the main tab (see [Hearts take-off](#)).

Questions and answers. FAQ

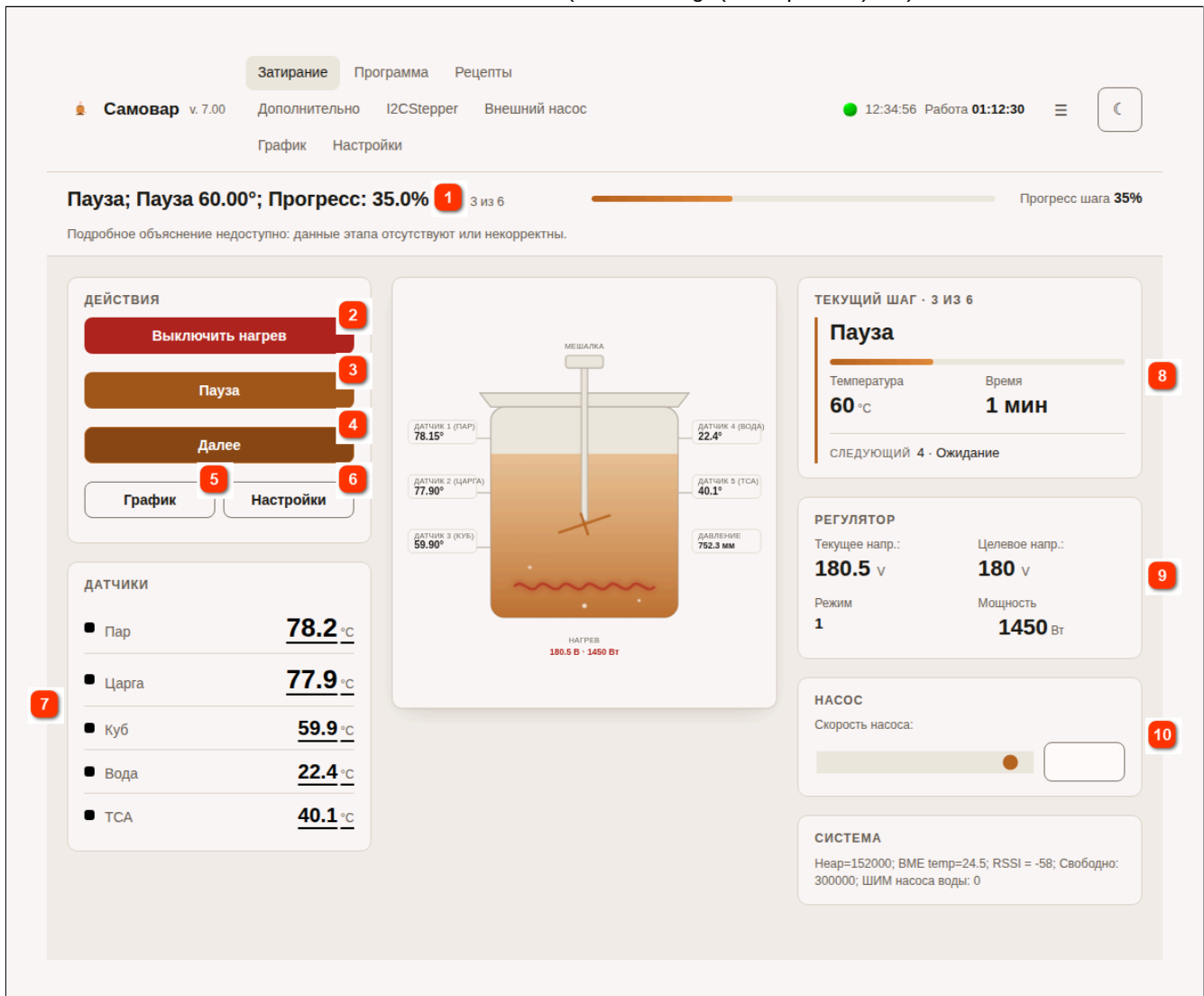
The help says that a pause resets the column section and vapor temperature values. Do I have to set new values after every pause?

No. When the work on the "Pause" line of the take-off program is finished, Samovar assigns the new hearts take-off temperatures itself. (The vapor temperature and the column section temperature)

The "Pause" («Поставить на паузу») command only stops the take-off pump and the timer of the program line being executed; the fixed temperature values are not reset. After the "Continue" («Продолжить») command, the impurity detector does not interfere with the take-off for about a minute.

Beer Mode

Main window of Beer mode (the "Mashing" («Затирание») tab)



Main window of Beer mode, the "Mashing" («Затирание») tab

- 1 Status line («Строка состояния»)**
The type and state of the current program line, the step number "N of M" («N из M»). Below it is the step progress in percent.
- 2 Turn heating on («Включить нагрев»)**
Start or stop executing the mashing program.
- 3 Pause / Resume («Пауза / Продолжить»)**
Manual brew pause: turns off heating, cooling, the stirrer and the pump until "Resume" («Продолжить») is pressed.
- 4 Next («Далее»)**
Move to the next line of the mashing program.
- 5 Chart («График»)**
Open the local chart.
- 6 Settings («Настройки»)**
Go to the settings page.
- 7 Sensors («Датчики»)**
The readings of all assigned sensors. Only the sensor selected in the program line controls heating.
- 8 Current step («Текущий шаг»)**
The line type, the temperature, the time and the next program line.
- 9 Regulator («Регулятор»)**
The current and target voltage (power), the regulator mode. Manual voltage entry is not used in Beer mode.



Pump («Насос»)

The speed of the wort transfer pump (PWM 0-1023).

Warning! Important!

Samovar's operation when brewing beer differs significantly from Rectification mode.

Samovar is set up for Beer mode **in 3 places**:

1. In the Samovar_ini.h file, the lines:

[#define BOILING_TEMP 98.9](#) - the boiling temperature on your equipment. On the Boiling («Кипячение») line, Samovar heats the boiler up to BOILING_TEMP + 5° and uses the same value to turn off heating if the wort overheats. The start of boiling itself is detected automatically, by the boiler temperature stabilizing.

[#define ACCELERATION_HEATER_DELTA 4](#) - how many degrees before the set temperature the boost heating element turns off.

[#define HEAT_DELTA 1](#) - how many degrees before the set temperature the main heater switches from boost to PID control.

2. On the [Beer tab](#) («Пиво») in Settings: the brewing layout (All-in-one / HERMS / RIMS), the PID controller coefficients, the regulator voltage (power) in temperature holding mode and in heating and boiling mode, and use of the boost heating element during boiling.

3. On the "Program" («Программа») and "Advanced" («Дополнительно») tabs while Beer mode is on.

Note.

The interface shows the readings of all temperature sensors, if they are connected and assigned on the Settings («Настройка») → Temperature («Температура») tab.

This is necessary to implement the various brewing layouts

But, only one temperature sensor is monitored. The one selected in the mashing program line.

This way you can use spare sensors as information sensors for your own needs. For example: to watch the temperature of the rinse water in an additional container, etc.

The line temperature is taken from the mashing program. The tolerance for reaching it is the "setpoint overshoot" («вылет за уставку») from the Temperature («Температура») tab for the sensor selected in the program line: from 0.3° to 3°; a value below 0.3° (including 0) is treated as 0.3°.

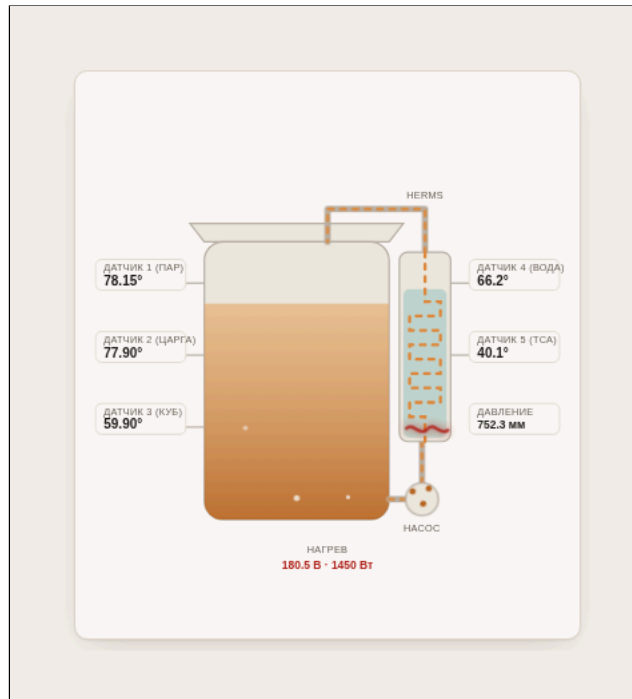
What is shown in the window

Status line (at the top): the type and state of the current program line and the step number "N of M" («N из M»). Below it is a bar and the **Step progress** («Прогресс шара») in percent: the share of the line time that has elapsed. It is calculated only for the Pause («Пауза») and Boiling («Кипячение») lines, and only after the temperature has been reached (for boiling - after boiling has been detected). While heating up, the progress is 0%.

Key values: the boiler temperature, the setpoint and time of the current step, the stirrer state.

Sensors: the readings of all assigned sensors (steam, column section, boiler, water, TCA).

Diagram: the picture depends on the brewing layout selected on the Beer («Пиво») tab. All-in-one - a kettle with a stirrer and a heating element; HERMS - a mash tun, a recirculation pump and a boiler with a coil; RIMS - a mash tun, a pump and a flow-through pipe with a heating element. The stirrer or pump is animated while the device of the current program line is on, the heating element is highlighted while heating, and during a pause the label "PAUSE" («ПАУЗА») appears.



Process diagram for the HERMS brewing layout: a recirculation pump and a boiler with a coil

Current step: the line type, the temperature, the time and the next program line.

Regulator: the current and target voltage (power), the regulator mode and power. There is no manual voltage entry in Beer mode - Samovar works by its own algorithm for reaching and holding the set temperature (see [Heater operation algorithm](#)).

Pump: the speed slider of the wort transfer pump (circulation pump), see [Advanced tab](#).

Status line

Example:

The malt has been added. "Next" («Далее») was pressed.

Samovar will send the message "Transition to program line No. 2; Temperature pause: 68.00°, time: 60.00 min"

(«Переход к строке программы №2; Температурная пауза: 68.00°, время: 60.00 мин») and will show in the status line: "Pause; Pause 68.00°; **Heating**; Current T: 61.50°" («Пауза; Пауза 68.00°; Разогрев; Текущая Т: 61.50°»).

That is, it is heating up, and the program time is not counting.

When the temperature reaches 68° (more precisely, 68° minus the tolerance, 0.3° by default), the message

"Temperature pause 68.00° reached. Waiting 60.00 minutes." («Достигнута температурная пауза 68.00°. Ждем 60.00 минут.») will arrive, and the line will change to "Pause; Pause 68.00°; **Progress**: 1.7%" («Пауза; Пауза 68.00°; Прогресс: 1.7%»).

The step time in the "Current step" («Текущий шаг») card shows the duration of the line according to the program; the heating time between pauses is not included in it.

Other states: "Heating to malt mash-in temperature" («Разогрев до температуры засыпи солода»), "Waiting for malt mash-in" («Ожидание засыпи солода»), "Boiling - heating" («Кипячение - нагрев»), "Boiling 60.00 min" («Кипячение 60.00 мин»), "Cooling to 30.00" («Охлаждение до 30.00»), "Waiting. Press 'Next program'" («Ожидание. Нажмите 'Следующая программа'»), "Fermentation; Hold T=26.00°" («Ферментация; Поддерж. T=26.00°»). If the brew is paused, the line starts with "On pause" («На паузе»).

The countdown for a program line starts slightly before the set temperature is reached. More details in the section [Mashing program](#).

Buttons

Turn heating on («Включить нагрев») / **Turn heating off** («Выключить нагрев»): start or stop executing the set program. Before the start, Samovar checks all program lines; if there is an error in some line, the start is cancelled with a message giving the line number. When stopped, the message "Mashing program finished" («Программа затирания завершена») arrives.

Pause («Пауза») / **Resume** («Продолжить»): a manual brew pause. It turns off heating, cooling, the stirrer and the pump and does not let the line advance further. The pause time does not count toward the line hold time; after "Resume" the stirrer schedule continues from where it was interrupted. Messages: "Mashing paused." («Затирание поставлено на паузу.») and "Mashing resumed." («Затирание продолжено.»). On the Autotune («Автотюнинг») and Lua Script («Lua Скрипт») lines the pause is postponed until the nearest brewing line.

Next («Далее»): move to the next line of the mashing program (with confirmation). If you press "Next" on a Cooling («Охлаждение») line while the wort is hotter than the target, Samovar will ask you to repeat the press within 10 seconds.

Chart («График»): view the local chart

Settings («Настройки»): go to the [settings](#) tab.

Tabs at the top of the window: "Mashing" («Затириание») (this window), ["Program"](#) («Программа»), ["Advanced"](#) («Дополнительно»). If the [I2CStepper](#) expander is connected, an "I2CStepper" tab appears, and if there is an external I2C pump, an "External pump" («Внешний насос») tab.

Mashing Program

A program for mashing malt for beer, grain washes and [sous-vide](#)

The program is stored in Samovar's memory and is not lost on reboot. If there is no user program, the factory example is loaded. *

The program loaded by default is shown below:

Программа затирки:

Варочный порядок из настроек: All-in-one / BIAB. ПИД по кубу (затор), мешалка как в универсальной программе.

1	Тип программы	2	Темп	3	Время	Мешалка	4	Датчик темп.	5
01	Темп. закладки солода	45	0	0^0^0^0^0	Куб (затор)				+ -
02	Пауза	45	1	0^0^0^0^0	Куб (затор)				+ -
03	Пауза	60	1	0^0^0^0^0	Куб (затор)				+ -
04	Ожидание	0	0	0^0^0^0^0	Куб (затор)				+ -
05	Кипячение	0	1	0^0^0^0^0	Куб (затор)				+ -
06	Охлаждение	30	0	0^0^0^0^0	Куб (затор)				+ -

► Описание программы затирки:

► Описание программы затирки в текстовом формате для сохранения в облаке:

Сохранить программу 6

Загрузить программу 7

Установить программу 8

Очистить программу 9

Загрузить рецепт из XML 10

The "Program" («Программа») tab: the factory mashing program

- 1 Program type («Тип программы»)**
Choice of the program to execute: Pause («Пауза»), Malt dough-in temp. («Темп. закладки солода»), Boil («Кипячение»), Cooling («Охлаждение»), Wait («Ожидание»), Fermentation («Брожение»), Autotune («Автотюнинг»), Lua script («Lua Скрипт»).
- 2 Темп. («Темп.»)**
The line temperature in °C. It is set for Pause, Cooling, Malt dough-in temp., Fermentation and Autotune; for Boil and Wait it is 0.
- 3 Time («Время»)**
Time in minutes. It is set only for Pause and Boil; for the other types it is 0.
- 4 Stirrer («Мешалка»)**
The operating mode of the stirrer and circulation pump: device^stirrer speed^pump speed^run time, s^pause, s. Clicking the field opens the editing window.
- 5 Temp. sensor («Датчик темп.»)**
The sensor that controls heating or cooling in this line: Boiler («Куб»), Water («Вода»), Column section («Цапра»), Steam («Пар») or TCA («ТСА»).
- 6 Save program («Сохранить программу»)**
Save the program and its description to the programbackup.txt file.
- 7 Load program («Загрузить программу»)**
Load a program from a file into the editing window (it reaches Samovar's memory after "Set program" («Установить программу»)).
- 8 Set program («Установить программу»)**
Write the program to Samovar's memory.
- 9 Clear program («Очистить программу»)**
Delete all program lines from Samovar's memory.



Load recipe from XML («Загрузить рецепт из XML»)

Go to the BeerXML recipe import page.

Above the table a hint is shown, "Brewing order from settings: ..." («Варочный порядок из настроек: ...») - the brewery layout chosen on the [Beer tab](#) («Пиво») in the Settings. It does not affect how the program runs; it is used by the BeerXML recipe import.

Columns:

№ - The program line number. No more than 20 lines can be set in total.

Program type («Тип программы») - choice of the program to execute

Temp. («Темп.») - temperature setting in degrees C°

Time («Время») - time in minutes.

Stirrer («Мешалка») - a field for setting the operating modes of the [stirrer and the circulation pump](#). It is filled in through a window that opens when you click the field.

Temp. sensor («Датчик темп.») - the choice of which temperature sensor to use to control heating or cooling (needed to implement different brewing orders). You can choose one of 5 sensors: "Boiler (mash)" («Куб (затоп)»), "Water (HERMS boiler)" («Вода (бойлер HERMS)»), "Column section (RIMS return)" («Цапра (обратка RIMS)»), "Steam" («Пар»), "TCA" («ТСА»).

The "+" and "-" buttons at the end of a line add a line after the current one and delete the current one.

Below the program there is a collapsible "Mashing program description" («Описание программы затирки») field, in which the program is shown in text format.

Программа затирки:

Варочный порядок из настроек: All-in-one / BIAB. ПИД по кубу (затор), мешалка как в универсальной программе.

№	Тип программы	Темп	Время	Мешалка	Датчик темп.	
01	Темп. закладки солода	45	0	0^0^0^0^0	Куб (затор)	+ -
02	Пауза	45	1	0^0^0^0^0	Куб (затор)	+ -
03	Пауза	60	1	0^0^0^0^0	Куб (затор)	+ -
04	Ожидание	0	0	0^0^0^0^0	Куб (затор)	+ -
05	Кипячение	0	1	0^0^0^0^0	Куб (затор)	+ -
06	Охлаждение	30	0	0^0^0^0^0	Куб (затор)	+ -

▼ Описание программы затирки:

Текст программы затирки:

```
M;45;0;0^0^0^0^0;0
P;45;1;0^0^0^0^0;0
P;60;1;0^0^0^0^0;0
W;0;0;0^0^0^0^0;0
B;0;1;0^0^0^0^0;0
C;30;0;0^0^0^0^0;0
```

1

Параметры программы затирания: Тип программы;Температура;Время;Устройства;Датчик

M - malt application temp (температура закладки солода), P - pause (пауза при заданной температуре),

B - boil (кипячение и засыпка хмеля в конце), C - cool (охлаждение до заданной температуры),

W - wait (режим ожидания, ничего не делает до смены строки программы)

F - ferment (ферментирование; время 0 - держать температуру бесконечно, время в минутах - перейти к следующей строке по его истечении; ручная пауза выключает и нагрев, и охлаждение до продолжения).

L - Lua скрипт: все параметры строки должны быть нулевыми. Вход ждёт подтверждённый запуск Lua-job, выход ждёт его остановку; ошибка Lua останавливает варку.

Варочный порядок (All-in-one / HERMS / RIMS) задаётся в Настройках, вкладка «Пиво». Импорт BeerXML подставляет датчик и насос из этого порядка; TYPE шара (Infusion/Temperature/Decoction) берётся из рецепта.

Пример программы отбора:

```
M;45.00;0;1^20^0^1^1;0
P;60.00;1;1^20^0^1^1;0
W;0.00;0;1^20^0^1^1;0
B;0.00;1;1^20^0^1^1;0
C;30.00;0;1^20^0^1^1;0
```

Описание устройств: предпоследнее поле имеет вид 3^100^1200^30^10, где

3 — мешалка и насос; -100 — 100 об/мин со сменой направления после паузы; 1200 — скорость шагового I2C-насоса в мл/ч; 30 — работа, сек; 10 — пауза, сек.

Первое число — устройство: 1 — мешалка, 2 — насос, 3 — оба устройства.

Второе число: 0 — мешалка на реле 1 I2CStepper, другое значение — обороты шаговой мешалки. Отрицательное значение включает смену направления после каждой паузы.

Третье число: 0 — насос на реле 1 I2CStepper, больше 0 — скорость шагового I2C-насоса в мл/ч. Мешалку и насос нельзя одновременно назначить на реле 1. Четвёртое и пятое — время работы и паузы в секундах.

▼ Описание программы затирки в текстовом формате для сохранения в облаке:

Описание программы затирки:

Венское. 25 л. Затирание 64/67/72/78, кипячение 60 мин.

2

Сохранить программу

Загрузить программу

Установить программу

Очистить программу

Загрузить рецепт из XML

The program in text form and the description field

1

Mashing program text («Текст программы затирки»)

The same program in the text format Type;Temperature;Time;Stirrer;Sensor. You can edit the lines right here and the table will update.

2

Mashing program description («Описание программы затирки»)

A free-form description (up to 250 bytes). It is saved to the file together with the program and is sent to the cloud at the start of the brewing session.

Line format: **Type;Temperature;Time;Stirrer;Sensor**, for example "P;60.00;1;1^-30^0^2^3;0". The stirrer is set with five numbers separated by "^" (see [Stirrer, Pump, Temperature](#)), the sensor by a number from 0 (boiler) to 4 (TCA).

Line designations explained:

M (malt application temp) – "Malt dough-in temp." («Темп. закладки солода»): heat to the set temperature and hold it until the operator switches to the next program line. When the set temperature is reached, a message will arrive saying the malt must be added. The time in the line must be 0.

P (pause) – "Pause" («Пауза»): heat to the specified temperature and hold it for the set time. When the time runs out, Samovar switches to the next program line. The temperature and time must be greater than 0.

W (wait) – "Wait" («Ожидание»): do nothing and wait until the operator switches to the next program line. Heating and cooling are off; the stirrer and pump work on the line's schedule. The temperature and time are 0.

B (boil) – "Boil" («Кипячение»): boil for the set time. 30 seconds before the time runs out, Samovar sends a message that hops must be added. That is, if hops must be added several times, there must be several boil lines. You can use just one, but you will have to track the hop addition time yourself. The temperature must be 0, the time greater than 0.

C (cool) – "Cooling" («Охлаждение»): cool to the set temperature. The cooling water supply valve opens and the cooling pump turns on (if one is used). The time must be 0.

F (ferment) - "Fermentation" («Брожение»): hold the set temperature by heating and cooling. Time 0 - hold indefinitely, until the line is changed manually; time greater than 0 (in minutes, up to 43200) - go to the next line when it runs out, see [Fermentation Program](#).

A – "Autotune" («Автотюнинг»): automatic tuning of the heating PID controller coefficients at the set temperature. When it finishes, the Kp, Ki, Kd coefficients are saved in the settings (Beer tab), and Samovar moves to the next line. The time is 0, the stirrer is off.

L – "Lua" («Lua»): the stage is controlled by a Lua script from a file on Samovar (only in firmware with Lua support, see [About the Lua Scripting Language](#)). When you choose type L, the "Lua stage" («Lua-этап») window opens: in it you choose the script file (any .lua file from Samovar's memory), set the stage timeout in seconds (1–65535) and, if needed, the parameters the script will receive in the arg table. The temperature, device and sensor are not set for such a line. The script must call `setNextProgram()`; if it does not manage to do so before the timeout or ends with an error, brewing stops.

If the program lines run out, Samovar turns off heating and finishes the program with the message "Mashing program finished" («Программа затириания завершена»).

At each transition to a new line the message "Transition to program line No. N" («Переход к строке программы №N») arrives with the line parameters. If "Turn on the buzzer when the program changes" («Включать зуммер при смене программы») is enabled in the Settings (Main tab), the buzzer also sounds.

The next collapsible field is the mashing program description in free text format (up to 250 bytes). It is saved together with the program in the backup file and is sent to the cloud at the start of the brewing session, for example:

"Vienna. 25 L. Mashing 64/67/72/78, boil 60 min, hops 33 g at 60, 30 and 5 min."

Explanations based on the starting program loaded by default.

Attention. There are line types that ignore time or ignore temperature. A program with an incorrectly filled line will not be accepted: when you try to set it, the message "Program error." («Ошибка программы.») appears with the number of the wrong line and the reason (for example, "Line 1: "Pause" - temperature and time greater than 0 are required" («Строка 1: «Пауза» - нужны температура и время больше 0»)), and at the start of brewing a message with the number of the faulty line appears.

Line 01. Malt dough-in temperature. Temperature 45° . Time 0

Water is poured into the mash vessel and heating is turned on. If a boost heating element is used, it also turns on.

When the temperature reaches 41°C the boost heating element turns off (see [Heater operating algorithm](#)).

In "Boost" («Разгон») mode, Samovar will keep heating the boiler up to 44°C.

Then the "Boost" mode turns off and heating is controlled by the PID controller, to approach the set temperature smoothly.

After the temperature of 45° is reached (more precisely, 44.7° with a tolerance of 0.3°, more on the tolerance below), Samovar shows the message "Malt dough-in temperature reached!" («Достигнута температура засыпи солода!»), and "Waiting for malt dough-in" («Ожидание засыпи солода») appears in the status line. Samovar will hold this temperature and wait indefinitely.

When the malt is added, the temperature in the mash tun starts to drop, and Samovar increases heating to hold the temperature.

After the malt is added, press "Next" («Далее») in the main window of the ["Beer" \(«Пиво»\)](#) mode. Execution of line No. 2 will begin

Line 02. Pause. Temperature: 45° . Time 1

Holding the temperature of 45° for 1 minute and an automatic transition to line No. 3. The message "Transition to program line No. 3; Temperature rest: 60.00°, time: 1.00 min" («Переход к строке программы №3; Температурная пауза: 60.00°, время: 1.00 мин») will be sent

Line 03. Pause. Temperature: 60° . Time 1

To reach the temperature, Samovar gives the regulator the "Boost" («Разгон») command. A few degrees before the target temperature the boost heating element turns off, the main heating element keeps working in "Boost" mode and then under the PID controller, i.e. the heater works on the same principle as for line No. 1

Attention!

Depending on the boiler volume, heating the last fractions of a degree can take a long time. So the time count for a line starts when the temperature is short of the set one by the tolerance. The tolerance is the "Setpoint overshoot" («Вылет за уставку») field of the line's sensor on the [Settings - Temperature](#) tab: from 0.3° to 3°; a value below 0.3° (including 0) is counted as 0.3°. The new value takes effect immediately after the settings are saved, including during brewing. For line No. 3 with a tolerance of 0.3°, the count starts at 59.7°, and the message "Temperature rest of 60.00° reached. Waiting 1.00 minutes." («Достигнута температурная пауза 60.00°. Ждем 1.00 минут.») arrives.

If during the rest the temperature drops below the tolerance (for example, after topping up), the time is not counted in full: the greater the undershoot, the slower the count. With a tolerance of 0.3° and a temperature of 58.35° (an undershoot of 1.65°), a minute is counted as half a minute. With an undershoot of 3° or more, the count stops and the message "Hold paused: temperature below target" («Пауза выдержки: температура ниже цели») arrives; after recovery, "Hold resumed" («Выдержка продолжена»). Heating works all this time. Overheating above the set temperature does not slow the count. The time of a manual pause (the "Pause" («Пауза») button) is also not counted toward the hold.

When the execution time is over, Samovar automatically moves on to the next program line

Line 04. Wait. Temperature 0.00. Time 0.

Wait for an operator command, doing nothing. No time is counted. Heating does not work either. The stirrer and pump, if set in the line, work.

Line 05. Boil. Temperature 0.00. Time 1.

Boil for the set time. No temperature value is entered in the line, only the time counts; zero time is not allowed. Samovar heats the boiler with a setpoint of `BOILING_TEMP + 5°` (the algorithm for reaching the temperature is the same as for line No. 1; the status line shows "Boil - heating" («Кипячение - нагрев»)).

Samovar detects the start of boiling by itself: when the temperature on the line's sensor is not below 98° and stops rising for about a minute. Then the message "Boiling mode started" («Начался режим кипячения») arrives, the time count begins, and heating is switched to the "Regulator voltage/power in heating and boiling mode" («Напр./мощность регулятора в режиме нагрева и кипения») value from the [Beer tab](#); the boost heating element works during boiling only if "Use boost heating element during boiling" («Использовать разгонный тэн при кипении») is enabled there. If boiling is not detected within 120 minutes, brewing stops with a message to check the boiler sensor, the liquid volume and the lid.

30 seconds before the line time runs out, Samovar sends the message "Add the hops!" («Засыпьте хмель!»), turns on the buzzer and makes a stepper motor step (for a hop dispenser, see [Additive Dispensers](#)). If hops must be added several times, there must be several boil lines: on moving to the next "Boil" («Кипячение») line, boiling is not detected again and the time count starts at once. You can use just one line, but you will have to track the hop addition time yourself.

The boiling temperature for your equipment is set manually, in the Samovar_ini.h file, the line [#define BOILING_TEMP 98.9](#). If the boiler temperature exceeds `BOILING_TEMP + 5°`, Samovar performs an emergency heating shutdown.

Line 06. Cooling. Temperature 30. Time 0

Cool to the set temperature. The time is not counted. Heating turns off, the cooling water supply valve opens and the cooling pump turns on at full power (if pump control is enabled in the firmware). When the temperature on the line's sensor drops to the set one, the valve closes, the pump stops and Samovar moves on to the next line. If the boiler has not cooled within 120 minutes, brewing stops with a message to check the sensor, the water supply and the valve. To skip cooling with the "Next" («Далее») button while the wort is hotter than the target, the press must be repeated within 10 seconds; Samovar warns about this with a message.

The command buttons are at the very bottom of the window:

Save program («Сохранить программу») - the mashing program together with its description will be saved to the programbackup.txt file in your browser's Downloads folder.

Load program («Загрузить программу») - load a mashing program from a file. Both a file saved with the "Save program" button and a plain text file with program lines are accepted.

Set program («Установить программу») - set the program in Samovar's memory for execution. The message "Program saved." («Программа сохранена.») will appear.

Clear program («Очистить программу») - delete all program lines from Samovar's memory (with confirmation).

Load recipe from XML («Загрузить рецепт из XML») - go to the page for [loading a recipe from a BeerXML file](#)

There is also a second way to get a recipe, without saving a file to your computer: press "Load recipe from XML"; in "Beer" («Пиво») mode the "Beer recipes from the site" («Рецепты пива с сайта») card is shown at the top of the page that opens - the samovar-tool.ru recipe catalog, from which a recipe is chosen right in the browser. For details, see the page on [loading a recipe from a BeerXML file](#).

During operation, as in the other modes, the operation log is saved to internal memory and to the cloud. Charts are plotted.

*** How do I make the Beer program load your recipe by default?**

The factory example program is set in the prepare_default_program_for_mode() function of the sensorinit.h file. Do not rely on the line number: it changes between releases.

Set and save the program for an upcoming brew through the web interface. Change the source only if you need a different factory example, which is created when there is no user program.

After changing the factory example, the firmware must be rebuilt and uploaded again. After every Samovar version update, the change will have to be repeated.

Description of the operation of the stirrer and wort transfer pump (circulation pump)

Attention.

Samovar 7.00: In Beer («Пиво») mode the stirrer and the wort transfer pump can run either from the outputs of Samovar itself or through the expander [I2CStepper](#). If a stepper stirrer or a stepper pump is set in the program line but the required board is absent, heating will not start: when the program starts, a message with the reason is shown ("I2C stirrer unavailable in line N" («I2C-мешалка недоступна в строке N»), "I2C pump unavailable in line N" («I2C-насос недоступен в строке N») or "I2C pump relay unavailable in line N" («I2C-реле насоса недоступно в строке N»)).

The following connection options are possible:

Stirrer on a relay - Samovar relay No. 2 switches on (rpm is 0), no expander is needed;

Stirrer on a stepper motor - an I2CStepper board with an odd address. The rpm and rotation direction are set; relay No. 2 switches on as well;

Pump on a stepper motor - an I2CStepper board with an even address. The flow rate in ml/h is set;

Pump without a stepper motor (flow rate 0) - the Samovar cooling pump output switches on; if an I2CStepper is connected, relay 1 of the board switches on with it. The cooling pump output exists only if the firmware is built with the PWM pump (USE_WATER_PUMP, enabled by default); without it the pump works only from relay 1 of the I2CStepper. Relay 1 for the pump is taken from the stirrer board if it has a relay, otherwise from the pump board.

Clicking the stirrer values input field opens an additional selection menu.

"Edit device mode" («Редактирование режима устройства») window (click the "Stirrer" («Мешалка») field of a program line)

- 1 **Device («Устройство»)**
Off, Stirrer, Pump, Stirrer and pump.
- 2 **Stirrer speed («Скорость мешалки»)**
0 - the stirrer on Samovar relay No. 2, otherwise the speed of the I2CStepper stepper stirrer in revolutions per minute; minus - change the direction after each pause.
- 3 **Run, sec («Работа, сек»)**
Run time, maximum 65535 sec.
- 4 **Pause, sec («Пауза, сек»)**
Pause time; 0 - work without pauses.
- 5 **OK («Ок»)**
Write the values into the program line field.

Device («Устройство») - you can choose what exactly works while this program line is running: the stirrer, the pump, or both devices. Or switch both off ("Off" («Выключено»)).

Stirrer: 0 — ESP32 relay 2, otherwise I2C, rpm («Мешалка: 0 — реле 2 ESP32, иначе I2C, об/мин») - 0 - the stirrer on relay No. 2, otherwise the speed of the stepper stirrer. A positive value is forward rotation; with a negative one

the stirrer changes its rotation direction after each pause. The field is shown only if a stepper stirrer board is connected; without it the field is replaced by "Stirrer: ESP32 relay 2" («Мешалка: реле 2 ESP32»).

Pump («Насос») - 0 - a pump without a stepper motor (cooling pump output and/or relay 1), greater than 0 - the stepper pump flow rate in ml/h. The field is shown only if a stepper pump board is connected; without it the text is "Pump: I2CStepper relay 1" («Насос: реле 1 I2CStepper») (if a stepper stirrer board is connected) or "Pump: ESP32 output" («Насос: выход ESP32»).

Run, sec («Работа, сек») - run time. (**Attention! Starting with version 6.22 the time is set in seconds**)

Pause, sec («Пауза, сек») - pause time. The maximum value of each field is 65535 sec (18 h 12 min).

When "OK" («Ок») is pressed, the values are checked: if a non-zero value is set for a device that is not selected, the message "Specify 0 for a disabled device." («Для выключенного устройства укажите 0.») is shown.

The order of stirrer and pump operation can also be set manually in the text field of the mashing program (see [Mashing program](#)). The field is set with five numbers; the (^) symbol is the value separator:

First - the device type: 0 - off, 1 - stirrer, 2 - pump, 3 - stirrer and pump at the same time;

Second - the stirrer rpm (from -32768 to 32767): 0 - the stirrer on Samovar relay No. 2, positive - forward rotation, negative - change the rotation direction after each pause. If the stirrer is not selected - 0;

Third - the stepper pump flow rate in ml/h (from 0 to 65535): 0 - a pump without a stepper motor (the Samovar cooling pump output and/or I2CStepper relay 1). If the pump is not selected - 0;

Fourth - run time in seconds (**Attention! Starting with version 6.22 the time is set in seconds**);

Fifth - idle (pause) time in seconds.

Examples:

1^~30^0^2^3 - only the stepper stirrer, 30 rpm: 2 seconds of rotation, 3 seconds of pause, the rotation direction changes after the pause.

1^0^0^20^40 - only the stirrer on relay No. 2: 20 seconds of work, 40 seconds of pause.

3^30^500^60^0 - the stepper stirrer at 30 rpm and the stepper pump at 500 ml/h without pauses.

If the stirrer/pump must work continuously, set any run time value (maximum 65535) and pause = 0, or leave the pause field empty.

A manual pause (the "Pause" («Пауза») button on the main window) stops the stirrer and the pump; after "Continue" («Продолжить») the schedule continues from where it was interrupted.

There is a validity check on how the "Stirrer" («Мешалка») field is formed. **Samovar 7.00**: allowed values are either 0^0^0^0^0 (devices off) or type 1, 2 or 3 with a non-zero run time; the rpm is set only for a selected stirrer, the flow rate only for a selected pump. A record of four numbers (the format of earlier versions) is not accepted. An incorrectly formed line is not saved: when "Set program" («Установить программу») is pressed, "Program error." («Ошибка программы») is shown with the line number and the reason.

Samovar outputs while the stirrer and pump work

Stirrer.

When the stirrer works, relay No. 2 switches on, connected to PIN D15 of the ESP32 (pin 19 for the LILYGO board, pin 42 for ESP32-S3), - both for a stirrer on a relay and together with the I2CStepper stepper stirrer.

Wort transfer pump

If the firmware is built with the PWM pump (USE_WATER_PUMP), then when the pump runs per the program line, the cooling pump output switches on - both for a pump without a stepper motor and together with the I2CStepper stepper pump. A pump with a 12 volt DC motor can be connected to it instead of the cooling pump.

There is an important peculiarity: the run time and the pause time apply to both the stirrer and the wort transfer pump. That is, if the stirrer is paused, the pump does not work either.

To connect a pump and a stirrer powered by something other than 12 volts (for example 220 volts), use intermediate devices such as solid-state relays.

Explanation of the stirrer and wort transfer pump algorithms:

If the device type "Pump" («Насос») is set in the program, the stirrer will not switch on.

If only the pump is used, it will work taking into account the pauses specified in the program line.

If only the stirrer is selected, the pump will not switch on.

Heater algorithm in "Boost" («Разгон») mode.

The settings can be changed in the file `Samovar_ini.h`.

To reach the temperature set in the mashing program, Samovar applies to the main heater the voltage from the field "Regulator voltage/power in heating and boiling mode" («Напр./мощность регулятора в режиме нагрева и кипения») ([Settings - Beer \(«Настройки - Пиво»\)](#)). If a [boost heating element](#) is used, it switches on too.

A few degrees before the target temperature the boost heating element switches off. This value is set in the line

`#define ACCELERATION_HEATER_DELTA 4` (4 is the value in degrees C°)

But the main heating element keeps working at this voltage until the set temperature minus the value in the line `#define HEAT_DELTA 1` (1 is the value in degrees C°)

Closer to the setpoint, heating is controlled by the PID controller, and the voltage is limited by the field "Regulator voltage/power in temperature holding mode" («Напр./мощность регулятора в режиме поддержания темп.»).

This algorithm keeps the set temperature from being "overshot".

Additional settings can be made in the settings - the tab ["Beer" \(«Пиво»\)](#).

Note. Control of the boost heating element by `ACCELERATION_HEATER_DELTA` works in a build with a UART regulator. In a relay build the boost heating element switches on together with the main one while the difference from the setpoint is greater than `HEAT_DELTA`, and switches off on transition to PID.

Set the boiling temperature for your equipment manually, in the file `Samovar_ini.h` the line `#define BOILING_TEMP 98.9`

Samovar 7.00: the start of boiling is detected not by reaching `BOILING_TEMP` but automatically - when the boiler temperature stops rising (not lower than `MIN_BOILING_TEMP 98°`). At that moment the message "Boiling mode started" («Начался режим кипячения») is shown and the time of the "Boiling" («Кипячение») line starts counting. If boiling is not detected within 120 minutes, the brew is aborted with the message "Failed to detect the start of boiling within 120 minutes..." («Не удалось зафиксировать начало кипения за 120 минут...»).

The value of `BOILING_TEMP` is used as follows:

- before boiling is detected, the "Boiling" line heats the boiler with the setpoint `BOILING_TEMP + 5°` by the algorithm described above (the "heating and boiling mode" voltage, the boost heating element while the difference is greater than `ACCELERATION_HEATER_DELTA`);
- after boiling is detected, the heater gets the voltage from the field "Regulator voltage/power in heating and boiling mode" («Напр./мощность регулятора в режиме нагрева и кипения»), and the boost heating element is on only if the "Use boost heating element during boiling" («Использовать разгонный тэн при кипении») checkbox is set ([Settings - Beer \(«Настройки - Пиво»\)](#));
- `BOILING_TEMP + 5°` is the emergency threshold for the boiler sensor on any program line: "Emergency shutdown! Maximum wort temperature exceeded" («Аварийное отключение! Превышена максимальная температура сусла»).

Cooling («Охлаждение»)

Program line: "Cooling" («Охлаждение»). Time is not taken into account. **Samovar 7.00:** the cooling water supply valve opens and the cooling pump switches on at full power. When the temperature set in the "Cooling" line is reached, the water supply valve closes, the pump switches off too, and Samovar moves on to the next line.

Since the wort transfer pump is connected in place of the cooling pump, a cooling pump must be connected in its place for the duration of the "Cooling" line - otherwise power will be applied to the pump output for the whole cooling time.

If the boiler has not cooled to the set temperature within 120 minutes, the brew is aborted with the message "Failed to cool the boiler to the target temperature within 120 minutes..." («Не удалось охладить куб до целевой температуры за 120 минут...»). Cooling can be skipped with the "Next" («Далее») button - press it again within 10 seconds.

Sparge water

Preparing the sparge water during the temperature rests.

Most beer brewing recipes, after all the rests have been completed and the wort has been drained, use sparge water at a temperature of about 80°C, to rinse the spent grain.

What can go wrong: the maximum power the electrical network can supply. If you heat the boiler and the sparge water vessel at the same time, the network may not handle the load.

There is a solution. Heat the water in the sparge vessel while the temperature rests of the mashing program are being carried out.

This is easy to arrange.

Connect the control (permission to run) of the sparge vessel heater to the normally closed (NC) contacts of relay No. 4, which controls the boost heating element.

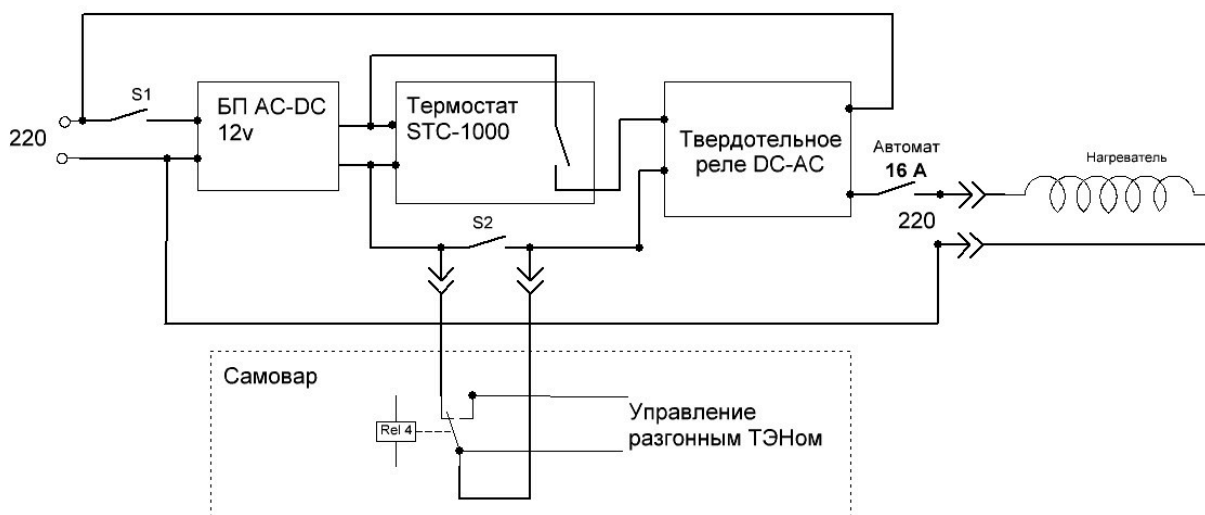
How it works.

If you fill the sparge vessel with water, you can switch the heating on, even if Samovar is not switched on, because the contacts of relay No. 4 are closed.

When Samovar starts working and the command to run the mashing program is given, relay No. 4, which controls the boiler's boost heating element, switches on, opens the NC contacts and the heating in the sparge vessel stops.

When the set temperature is reached, relay No. 4 switches off, the NC contacts close and the heating of the sparge vessel continues.

For those who actually use a boost heating element, which means there is enough power, this is simple to do. Relay No. 4, switching its contacts over, will turn on either the boiler's boost heating element or the sparge vessel heater.



Notes on the diagram:

Switch S1 is used to turn on the thermostat; S2 closes the connector so the device can work without Samovar.

The 16 A circuit breaker provides protection and is used to turn the heater on and off.

The AC-DC 220-12 volt module and the thermostat powered from 12 volts are used to increase safety. You can use a thermostat powered from 220 volts and an AC-AC solid-state relay.

The STC-1000 thermostat has two operating modes: heating and cooling, and accordingly two relays. One is responsible for heating, the other for cooling, which allows you to use this device on its own, able to maintain a set temperature in the fermentation vessel without using Samovar.

Note.

The control of relay No. 4 by the `ACCELERATION_HEATER_DELTA` parameter described below works in the build with a UART voltage regulator. In the relay build (without a regulator), relay No. 4 turns on together with the main heating element while the mash is colder than the setpoint by more than `HEAT_DELTA`, and turns off when it switches to temperature holding.

For those who do not use a boost heating element.

The wiring diagram is exactly the same, but...

According to Samovar's logic, the boost heating element turns off a few degrees before the target temperature is reached, and with our wiring the ranges overlap:

The main heating element is still running at full power, and the second heating element has already turned on.

To avoid this, in the Samovar_ini.h file, in the line

#define ACCELERATION_HEATER_DELTA 4

replace the value 4 with 0 (zero) or, even better, with -(minus)0.3

This gives the following operating algorithm:

1°C before the set rest temperature (set in the line #define HEAT_DELTA 1)

Samovar switches to PID control: the main heating element will receive, through the regulator, no more than the voltage from the "Regulator volt./power in temperature holding mode" («Напр./мощность регулятора в режиме поддержания темп.») field ([Settings - Beer \(«Настройки - Пиво»](#))).

When the temperature reaches the set value, the main heating element stops heating, relay No. 4 also switches off, the NC contacts close and permission is given to heat the sparge vessel.

If the line **#define ACCELERATION_HEATER_DELTA** is set to -0.3, the relay is de-energized only when the boiler temperature is 0.3°C above the target; this guarantees that the two heating elements cannot be on at the same time, even for a few seconds.

Added.

If you achieve an almost perfect approach to the temperature rests, the setting described above in the line

#define ACCELERATION_HEATER_DELTA

with values of both 0 and -0.3 may not work fully. The water in the intermediate vessel will not have time to heat up, because the temperature will first overshoot the target slightly, by about 0.4, but then drop and will be either equal to the target or 0.1-0.2°C below it. In that case the heating element in the additional vessel simply does not turn on.

A different setting is suggested.

For example:

With these values:

Regulator volt./power in temperature holding mode: 150 volts

#define HEAT_DELTA 2

In the line

#define ACCELERATION_HEATER_DELTA enter the value 1.5

And limit the power of the heating element in the sparge vessel to 1 kW

When 2° remain to the target temperature, 150 volts starts being supplied to the main heater. First continuously, then in pulses.

1.5° before the target, the heater in the intermediate vessel turns on at a power of 1 kW.

The network is not overloaded. The second heating element has enough time to heat the sparge water during the temperature rests and to maintain the temperature

The description of how the heaters work is in the section [Stirrer, pump, temperature](#).

Fermentation Program

Samovar can hold the temperature in a fermentation vessel.

Any selected sensor can be used - the boiler temperature sensor is recommended.

To use it, select the "Fermentation" («Брожение») line in the program list and set the temperature.

You can leave the time at 0 - then Samovar holds this temperature indefinitely, until you stop the program yourself or move to the next line.

If you set a time (in minutes, no more than 43200 - that is 30 days), Samovar moves to the next line by itself when it expires. This lets you build a program of several "Fermentation" lines with decreasing temperature, for example: 4320 minutes (3 days) at 20°, then 10080 minutes (7 days) at 18°, then a line with a temperature of 4° and time 0 - keep it cold until you come back. The time of a "Fermentation" line counts from the moment of switching to it; a manual pause is not counted. If a line with a time is the last one, the program finishes when it ends and the power is switched off - to prevent this, leave the time of the last line at 0.

In the text field, a program line starts with the character:

F (ferment) - fermenting, hold the set temperature until the program line changes.

The program runs without regard to time, so the line can be changed or the run stopped only manually.

The time in the line must be 0, and the devices (stirrer, pump) must be off: `0^0^0^0`. Otherwise the program will not be accepted (see [Mashing Program](#)).

Samovar 7.00: the width of the dead zone around the setpoint is not set on the "Temperature" («Температура») tab but is fixed in the firmware:

BEER_TEMP_HYSTERESIS 0.3 (file beer.h). During fermentation the cooling pump is used together with the valve.

How it works, by example:

Given: the temperature must be held at 26°C, +/- 0.3° (the tolerance is the sensor's "Setpoint deviation" («Вылет за уставку») of the line, see [Mashing Program](#)).

In text form the line looks like this : `F;26;0;0^0^0^0`

Suppose the temperature in the vessel is 28°C. Samovar turns off heating, opens the water valve and turns the cooling pump on at full power. Cooling works until the temperature drops to 26.3°.

If the temperature rises above 26.3° again, the valve and pump turn on again.

In the range from 25.7 to 26.3 the system stays idle: heating is off, the valve is closed.

When the temperature drops below 25.7°, the heating element is switched on through the PID controller, with the maximum voltage limited to the value from the "Regulator voltage/power in temperature holding mode" («Напр./мощность регулятора в режиме поддержания темп.») field ([Settings - Beer](#)). The lower the temperature falls, the more power is applied to the heating element. The boost heating element is not switched on during fermentation.

The "Pause" («Пауза») button on the main window turns off heating, cooling, the stirrer and the pump until you press "Resume" («Продолжить»).

During operation a log is kept, temperatures are sent to Blynk, and charts are plotted.

Samovar 7.00: protections work on the "Fermentation" line: if the maximum water or TCA temperature is exceeded, if the boiler temperature is above `BOILING_TEMP + 5°`, or if there is no water flow (when a flow sensor is installed), heating is shut off in an emergency with the message "Emergency shutdown!..." («Аварийное отключение!...»).

Warning!

Once every 12-18 hours heating may be shut off because of a communication error with the regulator. In that case, clear the "Use regulator connection check" («Использовать проверку подключения регулятора») checkbox on the Settings - [Other](#) tab.

Note.

Samovar 7.00: the log line recording period is no longer configurable (it is fixed at 4 seconds). The advice to increase the log recording interval for long fermentation does not apply to 7.00.

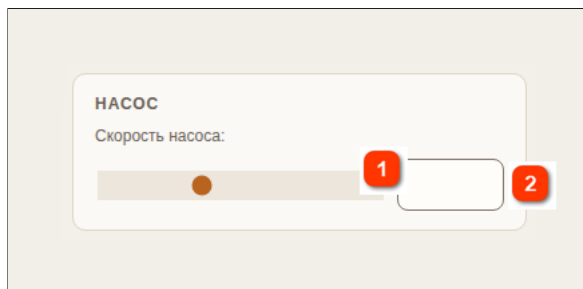
Tip.

As a rule, the mash needs close monitoring during the first 1-2 days after pitching, to avoid killing the yeast through self-heating of the wash or a significant slowdown of fermentation from strong cooling. Later, when the temperature has settled (you can see this on the chart), you can connect Samovar only at night in cold weather, or during the day in warm weather, to even out the mash temperature.

The Additional tab

Samovar 7.00: the pump speed slider has been moved to the "Mashing" («Затириание») tab (the "Pump" («Насос») panel of the main window of [Beer](#) mode). Only the Lua panel is left on the "Additional" («Дополнительно») tab.

Pump speed



The "Pump" («Насос») panel on the "Mashing" («Затириание») tab

1

Pump speed

PWM slider: the left edge is PWM_LOW_VALUE (100), the right edge is 1023 (12 V without modulation).

2

Input field

The same value as a number; you can type it in directly. 400 is the PWM_START_VALUE start value.

Here you can control the speed of the wort transfer pump (circulation pump) connected in place of the cooling pump. Or of the cooling pump itself, provided that in both cases the pump motor runs on 12 volts DC.

The power is changed by PWM modulation.

You can control it either with the slider on the speed scale or by typing a value directly into the field.

If the slider is at the far right position, 12 volts are applied to the pump without modulation. The input field value is 1023.

So, as a rough approximation, 10 units = 1% of power.

If the firmware was built without PWM pump support (USE_WATER_PUMP), trying to change the speed shows the message "Pump control is not supported by your hardware" («Управление насосом не поддерживается вашим оборудованием»).

Start and operating conditions:

- If the pump is allowed to run, then when a program line starts executing the pump turns on at the power set in the line of the Samovar_ini.h file

//The value (in percent) at which the PWM regulator for the pump starts working. Choose a value that is guaranteed to start the motor

```
#define PWM_START_VALUE 40
```

(40% = 400 units) and keeps running at this power. If you move the slider, Samovar remembers the value and from then on works at the newly set power, but only during the current session. After Samovar is rebooted, if you need a different power you must set it again.

Warning!

In cyclic operation on a schedule (work/pause in a program line), after every pause the pump starts again at the PWM_START_VALUE start value; to get the required speed back, you have to move the slider again. The slider value is applied immediately if the pump is running at that moment.

- If the slider is at the far left position, the pump does not stop but runs at the minimum power set in the line of the Samovar_ini.h file

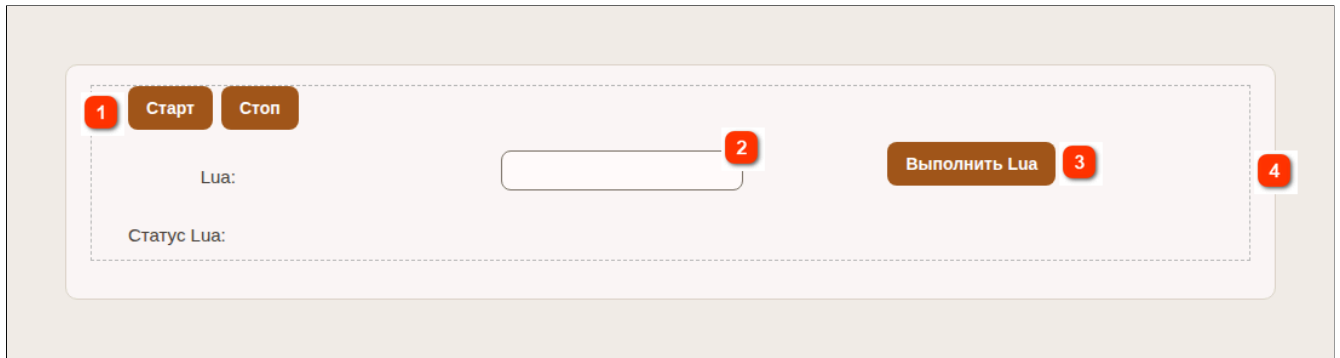
//The lower value (in percent) for the PWM regulator for the pump. Choose a value that ensures water flow

```
#define PWM_LOW_VALUE 10
```

(10% = 100 units - this is the left edge of the slider).

- On the "Cooling" («Охлаждение») and "Fermentation" («Брожение») lines, the firmware turns the cooling pump on at full power (1023) regardless of the slider position.

Lua panel



The "Additional" («Дополнительно») tab: the Lua panel

- 1 Script buttons**
Buttons declared by the Lua script. If the script has not defined any buttons, the panel is hidden.
- 2 Lua**
A field for a short piece of Lua code that is run once.
- 3 Run Lua («Выполнить Lua»)**
Send the code from the field for execution.
- 4 Lua status («Статус Lua»)**
The script state and the values it outputs.

The "Additional" («Дополнительно») tab contains the Lua script control panel. The panel is visible only if the script has defined a list of its buttons; the buttons from that list run script functions.

The "Lua status" («Статус Lua») line shows whether the script is running or stopped, and also displays information and values during execution. To run a script in Beer mode, it must be placed in the beer.lua file through the [editor](#).

In the field provided, you can enter a short piece of Lua code once and give the "Run Lua" («Выполнить Lua») command. The code will be executed.

Loading BeerXML Recipes

The BeerXML format is used in both amateur and professional brewing and makes it easier to exchange brewing data over the Internet. BeerXML is a free, fully defined XML data description intended for exchanging brewing recipes and other brewing data. The ability to import recipes into Samovar lets you take a ready-made beer recipe and, with suitable equipment, repeat an experienced brewer's success in "two clicks".

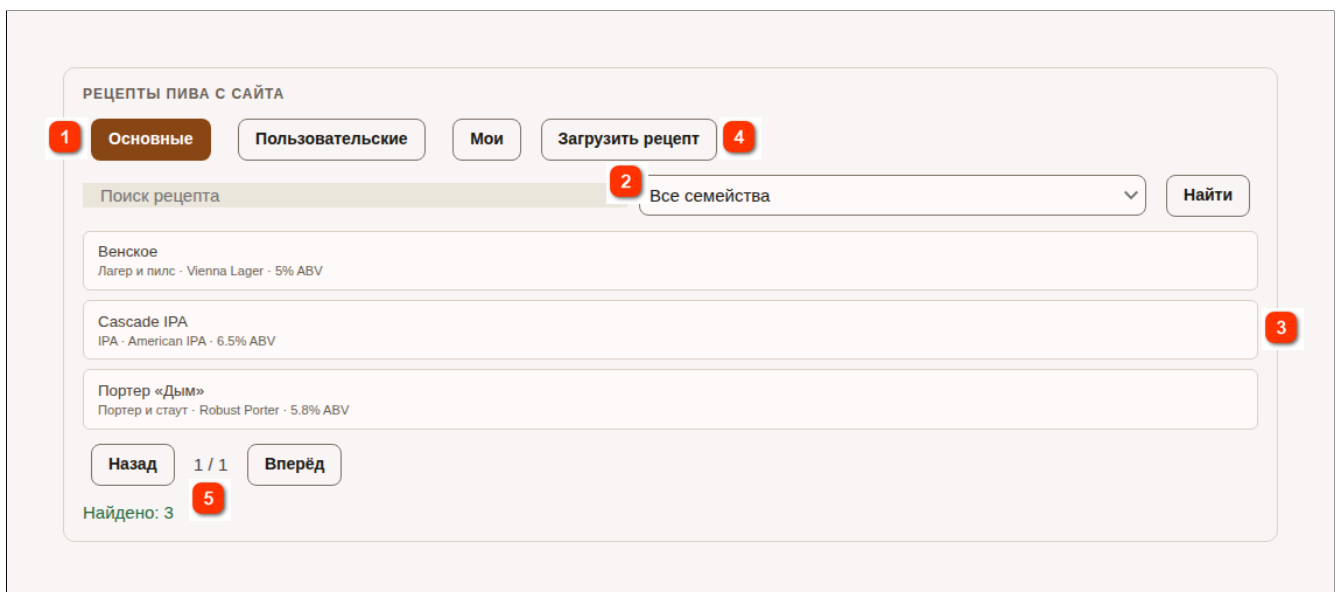
Samovar 7.00: besides BeerXML, recipe files of the BrewMate program (in windows-1251 encoding) are accepted. If the file contains several recipes, a "Recipe in file:" («Рецепт в файле:») list appears for choosing the one you need.

Beer recipes from the site

In "Beer" («Пиво») mode, the "Beer recipes from the site" («Рецепты пива с сайта») card is shown at the top of this page: the samovar-tool.ru recipe catalog right from the web interface, without saving a file to your computer. The same card has the "Load recipe" («Загрузить рецепт») button for a file from your computer.

There are three recipe sources: "Main" («Основные») (the site's general catalog), "User" («Пользовательские») (recipes of other users) and "Mine" («Мои») (your own recipes; for this list a Blynk token must be set in the device settings, otherwise the message "No Blynk token is set in the device for the "Mine" list." («Для списка «Мои» в устройстве не задан токен Blynk.») is shown). There is a search by name and a filter by beer family, and the recipe list is paged.

Choosing a recipe in the list loads it right away on the same page, under the card, just like when loading a file. Under the recipe name, "Source: ..." («Источник: ...») is shown; when loading a file from your computer this line is hidden.



The "Beer recipes from the site" («Рецепты пива с сайта») card at the top of the recipe loading page in "Beer" («Пиво») mode

- 1 Recipe source («Источник рецептов»)**
"Main" («Основные») is the site's general catalog, "User" («Пользовательские») is recipes of other users, "Mine" («Мои») is your own recipes (a Blynk token set in the device settings is required).
- 2 Search and filter («Поиск и фильтр»)**
Search by name and filter by beer family (IPA, "Porter and stout" («Портер и стаут»), "Lager and pilsner" («Лареп и пилс») and others).
- 3 Recipe list («Список рецептов»)**
Clicking a recipe loads it on the same page, under the card, just like when loading a file.
- 4 Load recipe («Загрузить рецепт»)**
Loading a BeerXML or BrewMate file from your computer.
- 5 Paged list («Постраничный список»)**
Switching the list pages with the "Back" («Назад») / "Forward" («Вперёд») buttons.

Loading a recipe from a file on your computer

Let's look at importing a recipe from the site www.brewmate.ru

Having chosen the recipe you are interested in on the site, save it to a specific place on your computer, on the "Program" («Программа») tab press "Load recipe from XML" («Загрузить рецепт из XML») (or the button of the "Recipes" («Рецепты») tab), and on the page that opens, in the "Beer recipes from the site" card, press "Load recipe".

File Explorer will open.

Go to the folder and choose the saved file to load. Samovar will load the recipe and show the main recipe data under the card. The example is based on the "Vienna" («Венское») beer recipe.

Самовар

РЕЦЕПТЫ ПИВА С САЙТА

Основные

Пользовательские

Мои

Загрузить рецепт

1

Поиск рецепта

Все семейства

▼

Найти

Венское
Лагер и пилс - Vienna Lager - 5% ABV

Назад

1 / 1

Вперёд

Найдено: 1

Программа затирания

Описание рецепта:

Стиль: Vienna Lager

Сорт: Венское

Тип: All Grain

Пивовар: Самовар

Выход: 25 л.

Объем кипячения: 30 л.

Температура брожения: 18 °C

Время кипячения: 60 мин.

Время брожения: 14 дней

OG: 1.05

FG: 1.012

IBU: 24

ABV: 5 %

КПД: 75 %

Ингредиенты:

Тип	Название	Количество	Параметры
Солод	Венский солод	5 кг	4 °L, выход 80 %
Хмель	Saaz	30 г	Кипячение, 60 мин, α 3.5 %, пеллеты
Хмель	Saaz	20 г	Кипячение, 15 мин, α 3.5 %, пеллеты
Дрожжи	Saflager W-34/70	11.5 мл	Fermentis, сухие, атт. 82 %

Температурные паузы:

Название	Тип	Температура	Время	Вода
Мальтозная пауза	Infusion	64	40	15 л
Осахаривание	Temperature	72	20	
Мэш-аут	Temperature	78	10	

Варочный порядок из настроек: All-in-one / BIAB. ПИД по кубу (затор), мешалка как в универсальной программе.

Установить программу

На главную

Recipe loading page after loading a file (the "Vienna" («Венское») recipe)

1

Load recipe («Загрузить рецепт»)

A button in the "Beer recipes from the site" («Рецепты пива с сайта») card: choose a BeerXML or BrewMate file on your computer.

2

Recipe description («Описание рецепта»)

Variety, style, volumes, boil time, fermentation temperature and time, OG/FG/IBU/ABV and other data from the file.

3

Ingredients («Ингредиенты»)

Malt, hops (with the addition time), yeast and other additives.

4

Temperature rests («Температурные паузы»)

Mashing steps from the recipe: name, type, temperature, time, water volume.

Hints («Подсказки»)

- 5 The brewing order from the settings and explanations of the "Wait" («Ожидание») lines for Infusion/Decoction steps.
- 6 **Set program («Установить программу»)**
Write the generated program to Samovar's memory.
- 7 **Home («На главную»)**
Return to the main page without saving the program.

Note. You can ignore the "Not in recipe" («Нем в рецепте») messages, since the absence of this data is of little importance for the mashing program. Description lines that are absent from the file are hidden; only "Variety" («Сорт»), "Fermentation temperature" («Температура брожения») and "Fermentation time" («Время брожения») are always shown.

The mashing program is built from the recipe:

- "Malt dough-in temp." («Темп. закладки солода») - the water temperature of the first mashing step (INFUSE_TEMP); if absent, the temperature of the first step, otherwise 45°;
- "Pause" («Пауза») - one for each mashing step from the "Temperature rests" («Температурные паузы») section. Before an Infusion step (except the first) and around a Decoction step, a "Wait" («Ожидание») line is added, for topping up water or for the decoction. The stirrer, pump and sensor in these lines are filled in according to the "Brewing order" («Варочный порядок») from the settings (All-in-one / BIAB, HERMS, RIMS, see [Settings - Beer](#)); the current order and hints are shown under the list of rests;
- "Boil" («Кипячение») - the total boil time of the recipe, split by hop addition times (USE = Boil or First Wort), so that Samovar reminds you "Add the hops!" («Засыпьте хмель!») before each addition;
- "Cooling" («Охлаждение») and "Fermentation" («Брожение») with the fermentation temperature from the recipe (PRIMARY_TEMP).

The program can have no more than 20 lines in total. If the file is damaged or has no mashing steps, an error message is shown ("The recipe file is damaged or not in BeerXML/BrewMate format: ..." («Файл рецепта повреждён или не в формате BeerXML/BrewMate: ...»), "the recipe has no mashing steps" («в рецепте нет шагов затирания»)).

Next, press "Set program" («Установить программу») (the message "Program saved." («Программа сохранена.») will appear) and the brewing program will appear in the "Program" («Программа») window. The "Home" («На главную») button returns to the main page without saving the program.

Программа затирки:

Варочный порядок из настроек: All-in-one / BIAB. ПИД по кубу (затор), мешалка как в универсальной программе.

№	Тип программы	Темп	Время	Мешалка	Датчик темп.	
01	Темп. закладки солода	70	0	1^60^0^2^	Куб (затор)	+ -
02	Пауза	64	40	1^60^0^2^	Куб (затор)	+ -
03	Пауза	72	20	1^60^0^2^	Куб (затор)	+ -
04	Пауза	78	10	1^60^0^2^	Куб (затор)	+ -
05	Кипячение	0	45	1^60^0^2^	Куб (затор)	+ -
06	Кипячение	0	15	1^60^0^2^	Куб (затор)	+ -
07	Охлаждение	18	0	0^0^0^0^0	Куб (затор)	+ -
08	Брожение	18	0	0^0^0^0^0	Куб (затор)	+ -

Описание программы затирки:

Описание программы затирки в текстовом формате для сохранения в облаке:

Сохранить программу

Загрузить программу

Установить программу

Очистить программу

Загрузить рецепт из XML

The "Program" («Программа») tab after importing the "Vienna" («Венское») recipe

- 1 **Program from the recipe**
Malt dough-in, rests at 64/72/78°, boil 45 + 15 min (hop additions), cooling and fermentation at 18°. Any line can be edited.

You can adjust the program as you see fit and to suit the equipment you use. The mashing program will also appear in the text field.

Программа затирки:

Варочный порядок из настроек: All-in-one / VIAВ. ПИД по кубу (затор), мешалка как в универсальной программе.

№	Тип программы	Темп	Время	Мешалка	Датчик темп.	
01	Темп. закладки солода	70	0	1^60^0^2^	Куб (затор)	+ -
02	Пауза	64	40	1^60^0^2^	Куб (затор)	+ -
03	Пауза	72	20	1^60^0^2^	Куб (затор)	+ -
04	Пауза	78	10	1^60^0^2^	Куб (затор)	+ -
05	Кипячение	0	45	1^60^0^2^	Куб (затор)	+ -
06	Кипячение	0	15	1^60^0^2^	Куб (затор)	+ -
07	Охлаждение	18	0	0^0^0^0^0	Куб (затор)	+ -
08	Брожение	18	0	0^0^0^0^0	Куб (затор)	+ -

▼ Описание программы затирки:

Текст программы затирки:

```
M;70;0;1^60^0^2^3;0
P;64;40;1^60^0^2^3;0
P;72;20;1^60^0^2^3;0
P;78;10;1^60^0^2^3;0
B;0;45;1^60^0^2^3;0
B;0;15;1^60^0^2^3;0
C;18;0;0^0^0^0^0;0
E;18;0;0^0^0^0^0;0
```

Параметры программы затирания: Тип программы;Температура;Время;Устройства;Датчик

M - malt application temp (температура закладки солода), P - pause (пауза при заданной температуре),

B - boil (кипячение и засыпка хмеля в конце), C - cool (охлаждение до заданной температуры),

W - wait (режим ожидания, ничего не делает до смены строки программы)

F - ferment (ферментирование; время 0 - держать температуру бесконечно, время в минутах - перейти к следующей строке по его истечении; ручная пауза выключает и нагрев, и охлаждение до продолжения).

L - Lua скрипт: все параметры строки должны быть нулевыми. Вход ждёт подтверждённый запуск Lua-job, выход ждёт его остановку; ошибка Lua останавливает варку.

Варочный порядок (All-in-one / HERMS / RIMS) задаётся в Настройках, вкладка «Пиво». Импорт BeerXML подставляет датчик и насос из этого порядка; TYPE шара (Infusion/Temperature/Decoction) берётся из рецепта.

Пример программы отбора:

```
M;45.00;0;1^20^0^1^1;0
P;60.00;1;1^20^0^1^1;0
W;0.00;0;1^20^0^1^1;0
B;0.00;1;1^20^0^1^1;0
C;30.00;0;1^20^0^1^1;0
```

Описание устройств: предпоследнее поле имеет вид 3^-100^1200^30^10, где

3 — мешалка и насос; -100 — 100 об/мин со сменой направления после паузы; 1200 — скорость шагового I2C-насоса в мл/ч; 30 — работа, сек; 10 — пауза, сек.

Первое число — устройство: 1 — мешалка, 2 — насос, 3 — оба устройства.

Второе число: 0 — мешалка на реле 1 I2CStepper, другое значение — обороты шаговой мешалки. Отрицательное значение включает смену направления после каждой паузы.

Третье число: 0 — насос на реле 1 I2CStepper, больше 0 — скорость шагового I2C-насоса в мл/ч. Мешалку и насос нельзя одновременно назначить на реле 1. Четвёртое и пятое — время работы и паузы в секундах.

▼ Описание программы затирки в текстовом формате для сохранения в облаке:

Описание программы затирки:

Венское. 25 л. Хмель Saaz на 60 и 15 мин.

Сохранить программу

Загрузить программу

Установить программу

Очистить программу

Загрузить рецепт из XML

The same program in text form and the program description

- 1 Mashing program text**
Lines Type;Temperature;Time;Stirrer;Sensor generated from the recipe.
- 2 Mashing program description («Описание программы затирки»)**
A reminder for the program, up to 250 bytes.
- 3 Set program («Установить программу»)**
Write the program to Samovar's memory after editing.
- 4 Save program («Сохранить программу»)**
Save the program and description to the programbackup.txt file.

In the "Mashing program description" («Описание программы затирки») text field you can enter a description of the program's features as a reminder (up to 250 bytes); it is sent to the cloud at the start of the brewing session.

After all the changes are made, the program must be set in Samovar.

To do this, press "**Set program**" («Установить программу») at the bottom of the page, and you can start brewing the mash.

With the "**Save program**" («Сохранить программу») command, a text file of your mashing program and its description named programbackup.txt will be saved to your browser's Downloads folder.

Beer Brew Journal on the Website and in the Apps

As in other modes, the brew log in Beer mode goes to the samovar-tool.ru cloud and is available on the page [Logs and Charts in the Cloud](#). In addition to this log, the website and the mobile apps have a separate "Brew journal": a catalog of recipes, a list of brews with scheduled tasks, and a card for each brew with the plan and the actual results.

Beer recipe catalog

The website page ["Beer recipes"](#) is the same catalog as the "Beer recipes from the website" card on the page [Loading BeerXML Recipes](#), but with the full set of filters: family, style type, recipe type, source, ranges of ABV (strength, % alcohol), IBU (bitterness) and OG (original gravity of the wort), search by name, and sorting by name, strength, bitterness, gravity, number of likes or update date.

The recipe card: a colored bar on the SRM scale (beer color - from straw to black, like a stout), targets for OG/FG (gravity before and after fermentation)/ABV/IBU/SRM, the grain bill (malt and other fermentables), hops with the time of addition, yeast, mashing steps, the author's notes, the source and the license. A slider scales the recipe from ¼ to 4× of the original volume - this changes only what is shown on the screen: the "Download BeerXML" button always gives the file in its original, unscaled form.

The "Start journal" button creates an entry in the account journal based on this recipe. The like icon lets you mark a recipe you liked. The file format itself is described on the website page ["BeerXML specification"](#).

Beer recipes

Search:

Catalogue:

Family:

Style type:

Recipe type:

Source:

ABV min:

ABV max:

IBU min:

IBU max:

Sort:

3 recipes

Dry Stout

Портер и стаут · Dry Stout (Ale)

ABV 4.3% · IBU 32 · OG 1.044

Batch: 20 L

Likes: 12

By brewer_anna

Blonde Session Ale

Портер и стаут · Session Ale (Ale)

ABV 3.1% · IBU 18 · OG 1.032

Batch: 19 L

Likes: 4

By brewer_anna

[Next](#)

Beer recipe catalog: filter by family and colored SRM bars on the cards (the screenshot was taken on a test mock-up of the interface with English labels; in the Russian version of the website and apps the same elements are labeled in Russian)

Brewer's account

The page ["Brewer's account"](#) uses the same website account as the other samovar-tool.ru pages (the same login, the same device list).

The **task board** at the top of the page shows what needs to be done for the brews in progress, and when. Tasks appear automatically when a brew moves to the "Fermentation" state, according to the recipe or, if there is no recipe, according to the beer family:

"Gravity check" - 3 and 7 days after fermentation starts;

"Dry hop" - on day 5 for each hop added "dry" (Dry Hop), or at the time given in the recipe;

"Diacetyl rest" - after 10 days, for lagers only;

"Cold crash" - 2 days before "Packaging";

"Packaging" - on day 14 (day 21 for a lager), if the recipe does not specify its own maturation time;

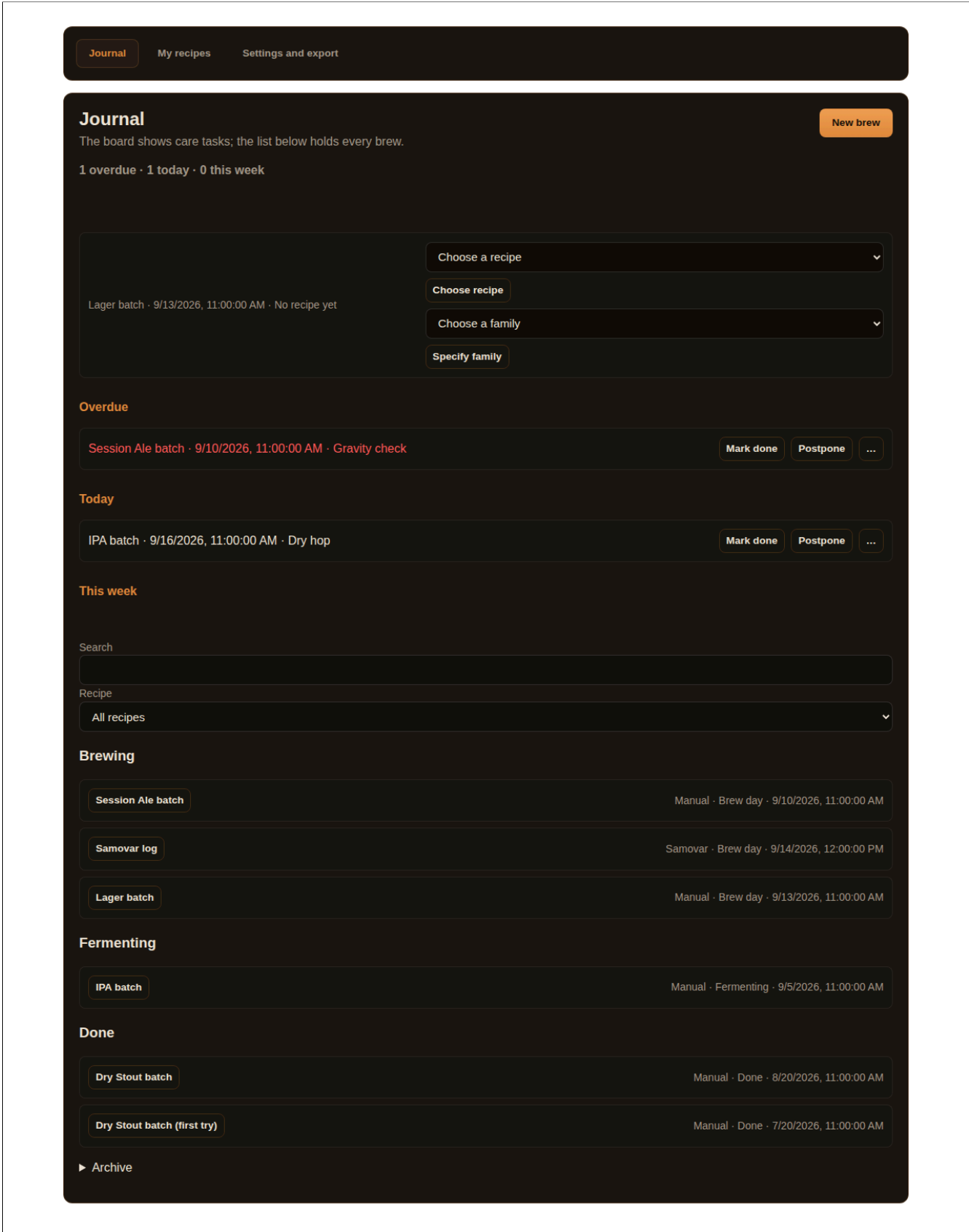
"Carbonation check" - 14 days after packaging;

"Ready" - the day after the carbonation check.

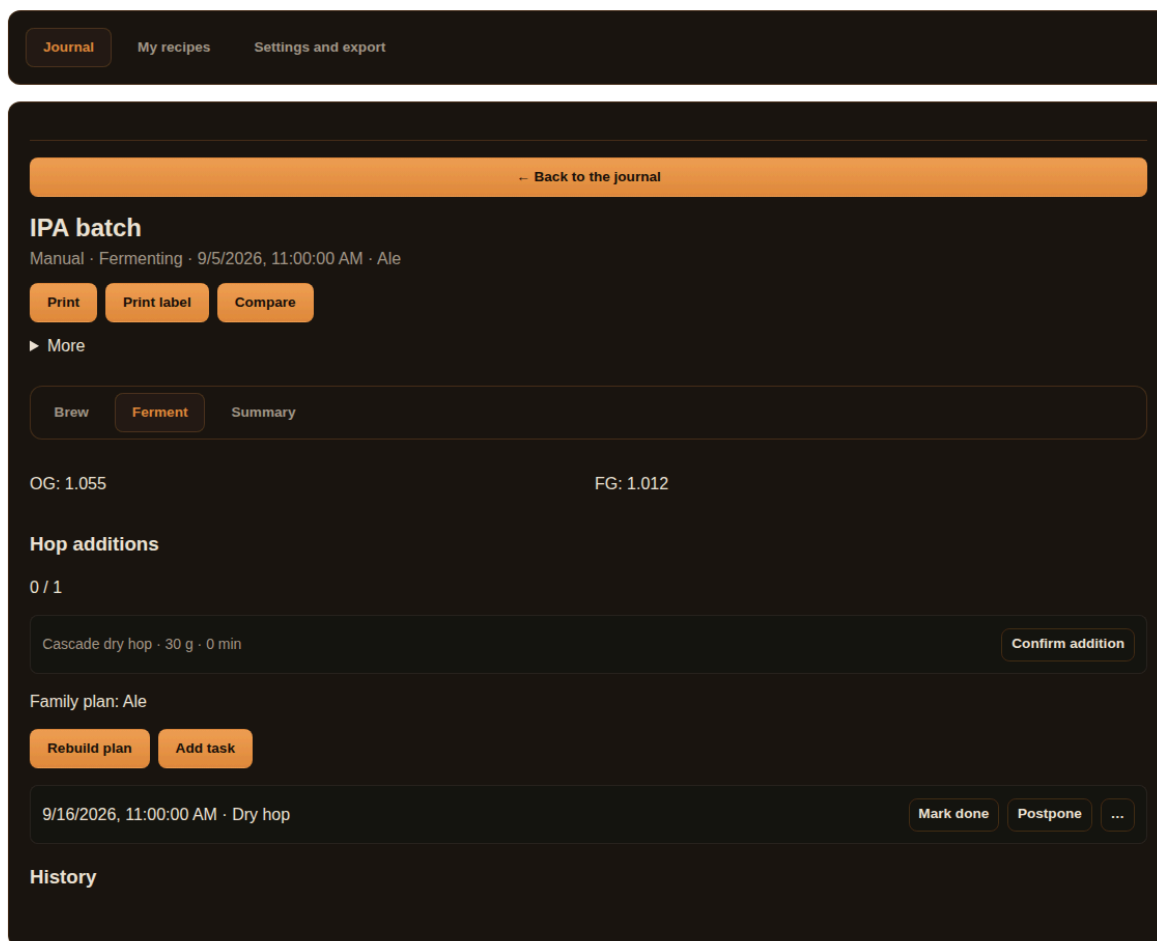
Without a recipe (for example, for a brew started directly on the device), the time until packaging is taken from the family: ale - 14 days, lager - 21 days (and with a diacetyl rest), wheat beer - 10 days, Belgian - 18 days.

The **list of brews** is divided into the groups "Brewing", "Fermenting", "Done" and an archive (cancelled and merged entries). The "New brew" button opens a dialog: a recipe from the catalog or just a beer family, the start date and the batch volume.

The **brew card** opens when you click an entry in the list and consists of three tabs:
"Brew" - plan and actual results for the mashing steps, boiling and hop additions, quick gravity and temperature measurements;
"Ferment" - scheduled tasks (see above) and measurements taken during fermentation;
"Summary" - OG/FG/ABV, attenuation (what share of the sugar has fermented, in percent), brewhouse efficiency, IBU, the number of fermentation days, comparison with previous brews, as well as tasting and off-flavors.
Printing the card and printing a label (name, date, strength) are available only on the website, not in the apps.



The task board and the list of brews by group (a screenshot of the same test mock-up, English labels)



The brew card: the fermentation tab with hops and confirmation of the addition (English labels in the screenshot)

Automatic entries. You do not have to create a brew on the website: if it is started on the device in Beer mode, an entry in the journal appears by itself, from the process log data - the same data that is shown on the page [Logs and Charts in the Cloud](#). It includes the mashing steps, boiling, the messages "Add the hops!" («Засыпьте хмель!») (as confirmation of the hop addition) and cooling. The device sends nothing extra for this - all the markup is done on the website from the same log data. If a brew was created on the website in advance by hand and then started on the device, the manual and automatic entries can be merged into one.

Beer in the apps

In the mobile apps (Android and iOS) the beer brew journal has a separate "Brewer" tab with two sections: "Journals" (the task board, the list and the brew card - the same as in the account on the website) and "Recipes" (a catalog with filters and a recipe card, without editing). The "Start batch" dialog is the same as on the website: a recipe with search, the date and the batch volume (the volume from the recipe is offered as a hint).

A push notification about a task or a change to a brew opens the app directly on the card of the relevant brew in the "Brewer" tab.

How it differs from Cheesemaking

The brewer's account is organized like the cheesemaker's account, but shorter: the card tabs have no "Wheels" (beer does not produce several cheese wheels from one batch), no "Care" (cheese aging with its own schedule of turning and rind washing) and no separate "Inventory". The task board, the list of entries and the merging of a manual entry with an automatic one work the same way for both modes.

See also: [Loading BeerXML Recipes](#), [Logs and Charts in the Cloud](#).

Sous-Vide

How Sous-Vide Works

The sous-vide technique was born in France, a country that highly values the art of cooking. From the French, sous-vide means "under vacuum", which directly points to the principle of the method. For some time the technique remained exclusive. The equipment was expensive and not every restaurant could afford it. It took root in molecular cuisine, which absorbs everything innovative. Now sous-vide dishes can be cooked even at home.

The sous-vide technique consists of two stages:

- packaging;
- heat treatment.

The food is packed in a vacuum bag. This is done with a special device, a vacuum sealer. Then the food is placed in a water bath, where it is cooked for a long time at a constant, relatively low temperature. The main difference from traditional methods is the strictly controlled temperature. On average it is 55-85 °C for different foods.

Sous-vide can be used to cook a wide variety of foods: meat, fish, seafood, vegetables and fruit. The food keeps its texture and natural look. Its natural flavor is revealed and enhanced. Even tough, rather dry meat comes out juicy, literally melting in the mouth. Vegetables do not turn into a tasteless puree and keep a light crunch. The unique sous-vide technique turns ordinary fruit into a refined dessert. Cooked apricots, plums or melon taste like marmalade.

Sous-vide in Samovar

Samovar 7.00: there is no separate "Sous-vide" («Су-вид») mode in the mode list (nor a Lua mode; both items are hidden in the [Settings](#)). For sous-vide use the [Beer](#) («Пиво») mode: the "Pause" («Пауза») (P) program line heats the water to the set temperature and holds it for the set time, and the pump, on the schedule of the same line, stirs the water so that the temperature in the container is even.

How to set it up:

- . In the [Settings](#), select the "Beer" («Пиво») operating mode and save.
- . On the [Settings - Beer](#) («Настройки - Пиво») tab, set the regulator voltage or power for the temperature holding mode; Samovar will use it to top up the heating during the hold.
- . Connect a circulation pump to the wort transfer pump output (see [Stirrer, Pump, Temperature](#)) and lower it into the container of water. The boiler sensor must be submerged in the water.
- . On the Beer mode page, set a program of one "Pause" («Пауза») line: the temperature is the one the food needs, the time is the hold duration in minutes, and in the "Stirrer" («Мешалка») field turn on the pump. The line format is described in the [Mashing Program](#) section.
- . Put the bagged food in and turn on heating. The timer starts when the water has warmed up to the set temperature; when the program ends, Samovar turns off the heating and the pump.

Example:

P;65;300;2^0^600^0;0

This program heats the water to 65° by the boiler sensor (the last field is 0) and holds it for 300 minutes (5 hours). The "Stirrer" field 2^0^600^0: device 2 is the pump, run time 600 seconds, pause 0 - the pump runs without stopping, stirring the water. If you need to run the pump with breaks, specify the pause in seconds, for example 2^0^60^120.

If during the hold the temperature drops below the tolerance (for example, after topping up the water), the timer slows down, and with an undershoot of 3° or more it stops until the temperature recovers, see [Mashing Program](#).

Questions and Answers. FAQ

For sous-vide, does the sensor in the tank have to be submerged in the water? And what if it does not reach the liquid? Maybe there are some correction deltas?

You can just top up the water.

Or you can try to find a delta. It will be very approximate and specific to a particular container and water volume.

It would also be good to arrange water circulation for stirring.

For home cooking a difference of one or two degrees is not critical (but it will be much more than 1-2° if the sensor is above the water),

Cheesemaking mode

Samovar 7.00: Cheesemaking mode first appeared in this version.

In Cheesemaking mode, Samovar runs a cheesemaking process by a program: it heats the milk at a set rate, holds the temperature, cools, switches the stirrer on by schedule, waits for the required acidity by the pH sensor, opens the drain valve and stops where the cheesemaker has to act (add the starter culture, cut the curd, move it into molds). The program consists of stages, and each stage is one line. The lines are described on the page [Cheesemaking program](#).

Equipment

Heating. The main heating element is switched on through relay no. 1 and the regulator, as in Beer mode: while the difference between the setpoint and the temperature is greater than [HEAT_DELTA](#), full power is applied, and then the PID regulator holds the temperature. The boost heating element is not used in this mode - relay no. 4 is taken by the drain valve.

Temperature sensors. In every line with a temperature you choose the sensor that the stage works by: "Boiler" («Куб»), "Water" («Вода»), "Column section" («Цапра»), "Vapor" («Пар») or "TCA" («ТСА») (the roles are assigned on the [Temperature](#) tab). One sensor in the vessel is usually enough - the "Boiler" role. The readings of the other sensors are shown on the main page for reference.

Stirrer. Either through relay no. 2 (on/off), or through the [I2CStepper](#) board with a stepper motor - then the speed and direction are set. The stirrer is set separately in each program line.

PH-4502C pH sensor. It is connected through the ADS1115 or to the LUA_PIN analog input with voltage matching. Wiring diagrams, the galvanically isolated variant and the operation of the filter are described in the technical section [pH acidity sensor and its connection](#). Before use, perform the [pH calibration](#).

Drain valve. It is connected to relay no. 4. In other modes the boost heating element sits on this relay - connect either the valve or the heating element, but not both at once. The valve opens at the "Drain" («Слив») stage and closes on the transition to the next stage and on any end of the program.

Cooling. Used at the "Cooling" («Охлаждение») stage. The scheme depends on the firmware build and is shown in the status line in the "Cooling" field:

pump ([#define USE_WATER_PUMP](#)) - the water supply valve opens and the cooling pump switches on; while the temperature is above the target by more than 1°C, the pump runs at full power, and closer to the target - at the minimum ([PWM_LOW_VALUE](#));

two valves ([#define USE_WATER_VALVE](#)) - the water supply valve opens, and a second valve on the [WATER_PUMP_PIN](#) pin opens for a stronger flow;

unavailable - the build has neither a pump nor a valve; a program with a "Cooling" stage will not start.

Dispenser (optional). The "Dosing" («Дозирование») stage can feed a component with Samovar's standard peristaltic pump - the volume and rate are converted to steps using the pump calibration ([Pump](#)).

Warning.

Without the ADS1115, the LUA_PIN input is taken by the pH sensor during cheesemaking: do not connect the PH-4502C and the MPX5010DP to it at the same time. When pH is connected through the ADS1115, this input stays free. In Cheesemaking mode, relay no. 4 controls the drain valve.

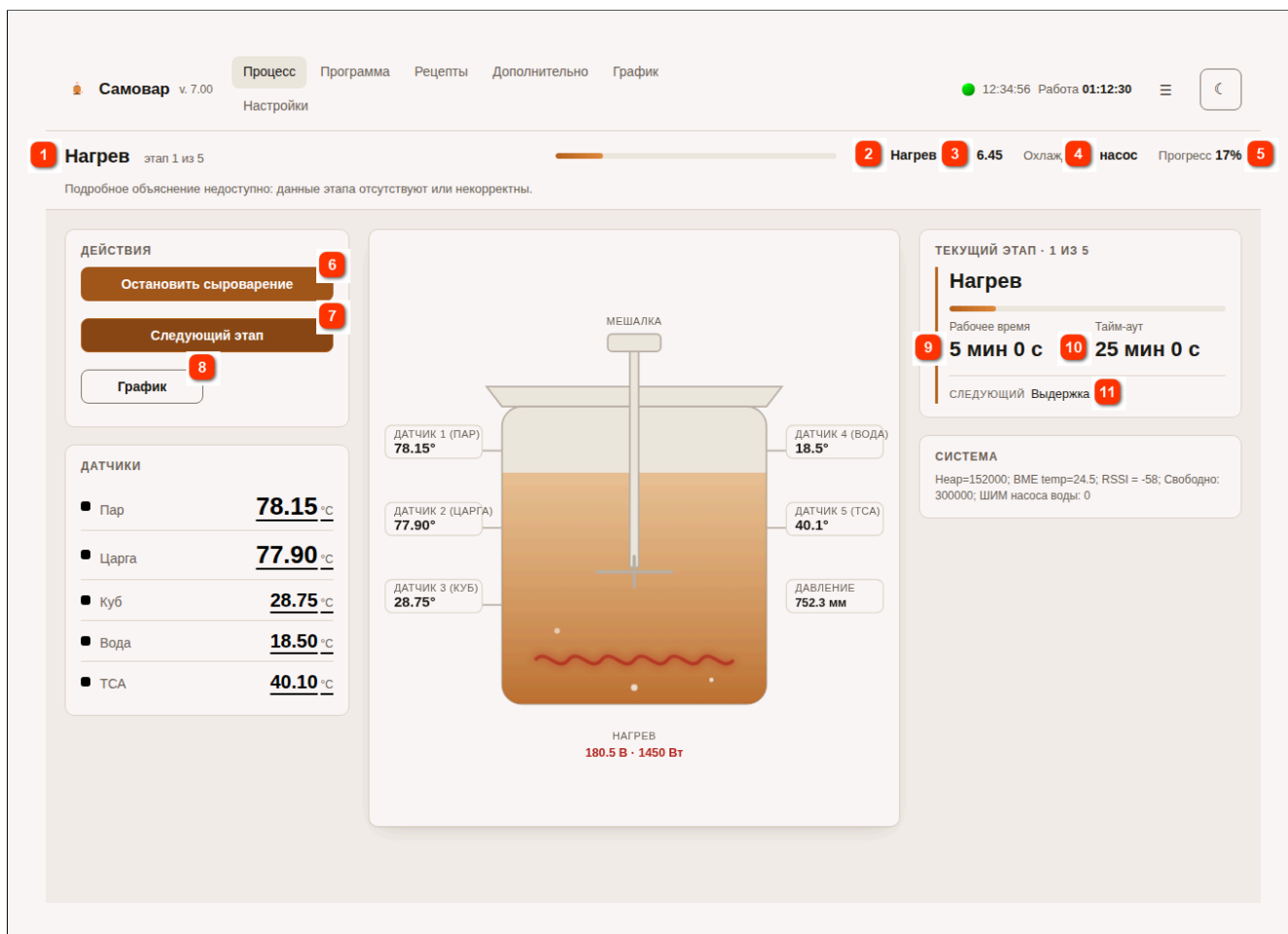
How to enable

Open [Settings → Main](#), in the "Operating mode" («Режим работы») field choose "Cheesemaking" («Сыроварение») and save the settings. The main page of Samovar becomes the cheesemaking page. The settings related to the mode are collected on the [Settings → Cheese](#) tab.

The first time the mode is enabled, a default program is loaded (heating to 63°C, hold, cooling to 34°C, adding the starter culture and the enzyme, cutting, draining and pressing). Replace it with your own on the "Program" («Программа») tab.

Main page of the mode

The page consists of four tabs: "Process" («Процесс»), "Program" («Программа»), "Recipes" («Рецепты») and "Additional" («Дополнительно»). The "Chart" («График») button opens the [local chart](#).



Main page of Cheesemaking mode, the "Process" («Процесс») tab

- 1 Status**
The name of the current stage and its number out of the total number of stages. While the program is not running - "Idle" («Ожидание»).
- 2 Stage**
The type of the current stage: Heating («Нагрев»), Hold («Выдержка»), Cooling («Охлаждение»), Stirring («Перемешивание»), Dosing («Дозирование»), pH wait («Ожидание pH»), Manual action («Ручное действие»), Drain («Слив») or Lua.
- 3 pH**
The current acidity from the PH-4502C sensor. If the sensor is not connected, not calibrated, or the value is outside 0-14 - "no data" («нет данных»).
- 4 Cooling**
The cooling scheme in this firmware build: "pump" («насос»), "two valves" («два клапана») or "unavailable" («недоступно»).
- 5 Progress**
The share of the stage time that has passed since it began, relative to the "Timeout, min" («Тайм-аут, мин») field (for a hold - relative to the hold time).
- 6 Start / Stop cheesemaking**
Starts the program from the first stage. While running, the button is called "Stop cheesemaking" («Остановить сыроварение») and ends the program with a confirmation.
- 7 Next stage / Confirm**
Goes to the next stage. At the "Manual action", "Drain" and manual "Dosing" stages the button is called "Confirm" («Подтвердить») - the stage will not end without pressing it.
- 8 Chart**
Goes to the local temperature chart.
- 9 Working time**
How long the stage has been running. For a hold, time with underheating beyond the tolerance is counted partially, and time with underheating of 3°C or more and with overheating above the tolerance is not counted.
- 10 Timeout**
How much time is left until the stage is forcibly stopped by the timeout.
- 11 Current and next stage**
The card of the current stage: name, progress, working time, timeout and the name of the next program stage.

On a phone screen, instead of separate cards, a "Key values" («Ключевое») block is shown: the vessel temperature, the stage setpoint, the current temperature, the working time and the timeout.

The "Sensors" («Датчики») block shows all connected temperature sensors, and the diagram in the center shows the vessel with its current temperature. At the bottom is the "System" («Система») line: free memory (Heap), board temperature (BME temp), Wi-Fi signal level (RSSI), free space in the file system and, if the water pump is used, its PWM.

The "Program" tab is the stage editor, see [Cheesemaking program](#).

The "Recipes" tab goes to the page [Cheese recipe catalog from the website](#): the samovar-tool.ru recipe catalog, a program preview by recipe and applying it.

The "Additional" tab has the "pH calibration" («Калибровка pH») button (goes to the page [pH calibration](#)), a reminder about connecting the PH-4502C and relay no. 4, as well as a field for running [Lua](#) commands and Lua buttons, if they are defined in the files btn_cheese_button1.lua and btn_cheese_button2.lua.

Workflow

- . Assign the temperature sensors and, if necessary, calibrate the pH sensor and the dosing pump.
- . On the "Program" tab compose the program and press "Save program" («Сохранить программу»).
- . On the "Process" tab press "Start cheesemaking" («Начать сыроварение»). Samovar checks the program: stages with a temperature must have a sensor assigned, for a stirrer over I2C the I2CStepper board must be connected, for a local dispenser the pump calibration must be done, for the "Cooling" stage a cooling scheme must be available, for the "pH wait" («Ожидание pH») stage the pH calibration must be saved, and for the "Lua" stage the file /cheese.lua must exist. If the check fails, the reason with the line number appears in the messages, and the heating does not switch on.
- . After the check, a log file is created, the heating switches on and the first stage begins. Each transition to a stage is accompanied by the message "Line N; stage name" («Строка N; название этапа»).
- . The "Manual action" («Ручное действие»), "Drain" and "Dosing" stages with the manual method wait for the cheesemaker: do the action and press "Confirm" («Подтвердить») (this is what the "Next stage" («Следующий этап») button is called at that time). The "Next stage" button can finish any stage early. A short press of the [universal button](#) does the same, and a long press ends the cheesemaking.
- . After the last stage, Samovar switches off the heating, cooling, stirrer and dispenser, closes the drain valve and shows the message "Cheesemaking program finished" («Программа сыроварения завершена»). The status line shows "Idle" («Ожидание»).

Before a stage changes, all actuators are put into a safe state (heating and cooling off, stirrer and dispenser stopped, drain closed), and then whatever the new stage needs is switched on.

Stopping, timeouts and errors

Stop cheesemaking («Остановить сыроварение») - ends the program at any moment (with a confirmation). The result is the same as at a normal end: all outputs are off, the drain valve is closed.

Timeout. Every stage has a time limit (the "Timeout, min" («Тайм-аут, мин») field; for the "Hold" («Выдержка») stage it is "Total timeout, min" («Общий тайм-аут, мин»)). If the stage does not finish within this time, cheesemaking stops with the message "Line N: Cheese program operation timeout" («Строка N: Тайм-аут операции сырной программы») (for a hold - "Hold timeout" («Тайм-аут выдержки»)). The exception is "Stirring" («Перемешивание»): its duration is the completion condition.

Stage errors also stop the program with the message "Line N: reason": no data from the selected temperature sensor, a stirrer or cooling error, an unreliable pH for longer than 10 s at the "pH wait" stage, a Lua script error.

The safety system works as in Beer mode: emergency shutdown when the vessel temperature is above [BOILING_TEMP](#) + 5°C, when the limit water or TCA temperature is exceeded at the cooling stage, and when there is no water flow (if flow control is enabled). Overheating and emergencies are described on the page [Emergency button, additional emergency sensors](#).

Reboot during cheesemaking. Samovar remembers the mode and the number of the current stage. After a reboot the heating does not resume, and the message "Session interrupted by reboot: line N/M" («Сессия прервана перезагрузкой: строка N/M») appears in the messages. If the program from the snapshot fails the check, it is cleared, and the default program is not substituted - compose the program again.

Cheesemaking program

The program is set on the "Program" («Программа») tab of the main page of the [Cheesemaking](#) mode. Each line is one stage; stages run in order. A program can have up to 20 stages.

1	2	3	4	5	10	6	7	8	9	11	12
01	Нагрев	32	60	1	Куб	RELAY	0	Прямое	0	0	+
02	Выдержка	32	30	60	Куб	RELAY	0	Прямое	0	0	+
03	Ручное действие		30	Внести ферме		NONE	0	Прямое	0	0	+
04	Ожидание pH	32	120	6.4	Куб	NONE	0	Прямое	0	0	+
05	Слив		30								+

13 Применить программу
Экспорт
Импорт

The "Program" tab of the Cheesemaking mode («Вкладка «Программа» режима Сыроварение»)

- 1 Stage number («Номер этапа»)**
The sequence number of the line. Stages run from top to bottom.
- 2 Stage type («Тип этапа»)**
Heating, Hold, Cooling, Stirring, Dosing, Wait for pH, Manual action, Drain or Lua ("Нагрев, Выдержка, Охлаждение, Перемешивание, Дозирование, Ожидание pH, Ручное действие, Слив, Lua"). When the type changes, the line fields are reset to the defaults.
- 3 Value 1 («Значение 1»)**
The target temperature (for dosing - the volume in ml). The label above the field changes with the stage type.
- 4 Value 2 («Значение 2»)**
The stage timeout in minutes; for a hold - the hold time, for stirring - the duration.
- 5 Value 3 («Значение 3»)**
For heating - the rate in °C/min, for a hold - the total timeout, for waiting for pH - the target pH, for dosing - the rate or the component. For a manual action this place holds the list of actions.
- 6 Stirrer («Мешалка»)**
The stirrer device: NONE, RELAY (relay No. 2) or I2C (an I2CStepper board).
- 7 RPM**
The I2C stirrer revolutions. For RELAY and NONE - 0. The rotation direction is chosen in the adjacent list.
- 8 Run, s («Работа, с»)**
The stirrer run time in seconds. 0 - the stirrer is on for the whole stage.
- 9 Pause, s («Пауза, с»)**
The pause between starts. 0 with a set run time - the stirrer is switched on once.
- 10 Sensor / Method («Датчик / Способ»)**
The temperature sensor the stage works by (Boiler, Water, Column section, Steam, TCA; "Куб, Вода, Цапра, Пар, TCA"). For dosing - the method: "Manual" or "Local stepper motor" ("Вручную" or "Локальный шаговый двигатель").
- 11 Add line («Добавить строку»)**
Inserts a new stage after the current line.
- 12 Delete line («Удалить строку»)**
Deletes the line. The last remaining line cannot be deleted.
- 13 Save program («Сохранить программу»)**
Checks all lines and sends the program to Samovar. Next to it are "Export" to a file and "Import" from a file ("Экспорт", "Импорт").

When you change the stage type, the line fields are reset to the defaults for that type, and unneeded fields are hidden. The values are checked on saving: temperature from 0 to 150°C, time from 0 to 1440 min. On an error the message "Check the fields of stage N" («Проверьте поля этапа N») is shown, and such a program is not sent to Samovar.

Save program («Сохранить программу») - send the program to Samovar. While cooking is in progress, the "Process" («Процесс») tab uses the last saved program, not unsaved edits.

Export («Экспорт») - download the program to the file cheese-program.txt.

Import («Импорт») - load a program from a text file. The file is checked in the same way as editor lines; on an error the current program does not change.

Line format

In the file and in Samovar's memory a line is stored as six fields separated by semicolons:

[Type;Value1;Value2;Value3;Stirrer;Value4](#)

The meaning of the "Value1-4" fields depends on the stage type (see the table below). The "Stirrer" field is four numbers separated by $\wedge$: device^{rpm}run, s^{pause}, s. For example:

[H;63;90;1;1^0^0^0;0](#) - heat to 63°C at 1°C/min, timeout 90 min, the stirrer runs through the relay for the whole stage, the "Boiler" sensor;

[M;0;1;0;1^0^30^0;0](#) - stirring: the stirrer is switched on through the relay once for 30 s;

[W;0;15;1;0^0^0^0;0](#) - the manual action "Add starter culture", wait for confirmation for no more than 15 min;

[N;32;120;6.4;0^0^0^0;0](#) - hold 32°C until the pH drops to 6.4, for no more than 120 min.

Stage types

In all stages except "Stirring" («Перемешивание»), the "Timeout, min" («Тайм-аут, мин») field is the maximum time of the stage: if the stage does not finish within this time, cheesemaking stops. "Sensor" («Датчик») is the temperature sensor the stage works by: "Boiler" («Куб»), "Water" («Вода»), "Column section" («Цапра»), "Steam" («Пар») or "TCA" («ТСА»). The temperature is considered reached when the sensor reading stays within $\pm 0.3^\circ\text{C}$ of the target for 10 seconds in a row.

H - Heating («Нагрев»). Fields: "Target, °C" («Цель, °C»), "Timeout, min", "Rate, °C/min" («Скорость, °C/мин»), "Sensor". The heating setpoint rises from the starting temperature at the given rate until it reaches the target. The stage ends when the sensor has reached the target.

P - Hold («Выдержка»). Fields: "Target, °C", "Hold, min" («Выдержка, мин»), "Total timeout, min" («Общий тайм-аут, мин»), "Sensor". Samovar holds the target. The hold time is counted in full while the temperature differs from the target by no more than the tolerance. The tolerance is the "Setpoint excursion" («Вылет за уставку») field of the stage sensor on the [Settings - Temperature](#) tab («Настройки - Температура»): from 0.3°C to 3°C; a value below 0.3°C (including 0) is treated as 0.3°C. When the temperature is under the target by more than the tolerance, the time is counted partially - the less, the bigger the shortfall; with a shortfall of 3°C or more, and also with an overshoot above the tolerance, the time is not counted at all: the message "Hold paused: temperature is outside the band" («Выдержка приостановлена: температура вне полосы») arrives, and after the temperature returns, "Hold resumed" («Выдержка продолжена»). The stage ends when the set time has accumulated. The total timeout counts from the start of the stage and cannot be less than the hold; when it expires, the process stops with "Hold timeout" («Тайм-аут выдержки»).

C - Cooling («Охлаждение»). Fields: "Target, °C", "Timeout, min", "Sensor". Heating is off; while the temperature is above the target, cooling water is supplied (more than 1°C above the target - full flow, closer - minimum). The stage ends when the sensor has reached the target; cooling is then turned off. The "pump" or "two valves" cooling scheme is required.

M - Stirring. Field: "Duration, min" («Длительность, мин»). A stirrer is required. The stage ends when the duration expires, or, if the stirrer has a run time without a pause set, immediately after it has run for that time once. The stage has no timeout.

D - Dosing («Дозирование»). Fields: "Volume, ml" («Объём, мл»), "Timeout, min", "Method" («Способ») and a third field that depends on the method.

"Manual" («Вручную») - the third field is "Component" («Компонент») (the same list as for the manual action). Samovar waits until the cheesemaker adds the component and presses "Confirm" («Подтвердить»).

"Local stepper motor" («Локальный шаговый двигатель») - the third field is "Rate, ml/min" («Скорость, мл/мин»). The volume is delivered by Samovar's standard peristaltic pump; the volume and rate are converted to steps using the pump calibration, so the pump must be calibrated ([Pump](#)). The stage ends when the pump has run the set volume. The dosing device options are described on the page [Additive dosing units](#).

N - Wait for pH («Ожидание pH»). Fields: "Temperature, °C" («Температура, °C»), "Timeout, min", "Target pH" («Целевой pH»), "Sensor". Samovar holds the temperature and watches the pH sensor. The stage ends when the pH is not above the target for 30 seconds in a row. If the pH reading is unreliable (no data or outside 0-14) for more than 10 seconds, cheesemaking stops with the message "pH unreliable for 10 seconds" («pH недостоверен 10 секунд»). The stage needs a saved [pH calibration](#).

W - Manual action («Ручное действие»). Fields: "Timeout, min", "Action" («Действие»). Samovar does nothing (except for the stirrer, if one is set) and waits for "Confirm" to be pressed. The list of actions: "Add starter culture" («Внести закваску»), "Add enzyme" («Внести фермент»), "Add additive" («Внести добавку»), "Cut the curd" («Разрезать сгусток»), "Transfer to molds" («Переложить в формы»), "Start or change pressing" («Начать или изменить прессование»), "Flip the cheese" («Перевернуть сыр»), "Other action" («Другое действие»). The action name serves as a hint in the program; in the status line Samovar shows only "Manual action".

S - Drain («Слив»). Field: "Timeout, min". The drain valve (relay No. 4) opens and Samovar waits for "Confirm" to be pressed. The valve closes on moving to the next stage. A stirrer cannot be set on this stage.

L - Lua. Field: "Timeout, sec" («Тайм-аут, сек»). A script from the file selected in the "Lua stage" («Lua-этап») window is started (any .lua file from Samovar's memory; the timeout and the parameters that the script receives in the arg table are also set there, see [Lua](#)). To move to the next stage, the script must call [setNextProgram\(\)](#). If the script ends without this call or with an error, cheesemaking stops. A stirrer cannot be set on this stage.

Stirrer

The stirrer is set separately in each line and runs only during that stage. Five fields:

Stirrer («Мешалка») - the device: "NONE" - no stirrer; "RELAY" - relay No. 2 (on/off); "I2C" - the stepper motor on an [I2CStepper](#) board. For "I2C" the board must be connected, otherwise the program will not start.

RPM - revolutions for "I2C", an integer greater than zero. For "RELAY" and "NONE" it must be 0.

Direction («Направление») - "Forward" («Прямое») or "Reverse" («Обратное»), only for "I2C". In the file the reverse direction is written as negative revolutions (for example [2^-120^0^0](#)).

Run, s («Работа, с») and **Pause, s** («Пауза, с») - the schedule. Run 0 - the stirrer is on for the whole stage. Run greater than 0, pause 0 - the stirrer is switched on once for the given time and then switched off. Run and pause both greater than 0 - the stirrer alternates run and pause until the end of the stage. A pause without a run is not allowed.

When moving to the next stage, the stirrer stops and is started again according to the settings of the new line.

Cheese recipe catalog from the website

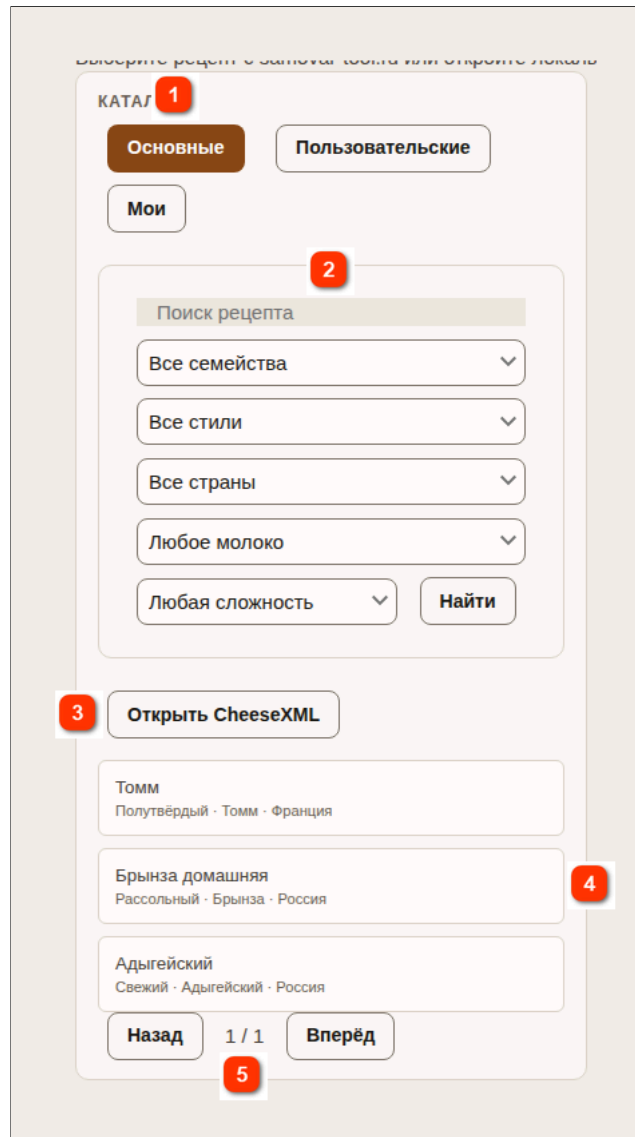
The "Recipes" («Рецепты») tab on the main page of [Cheesemaking mode](#) opens this page - the samovar-tool.ru cheese recipe catalog, with a preview and the ability to apply a program right in the browser, without saving a file to your computer.

Catalog

There are three recipe sources: "Main" («Основные»), "User" («Пользовательские») and "Mine" («Мои») (buttons above the list). For the "Mine" source, a Blynk token must be set in the device settings - otherwise the message "A Blynk token is not set in the device for the "Mine" list." («Для списка «Мои» в устройстве не задан токен Blynk.») is shown, and the filters become unavailable: only searching by name among the already loaded recipes works.

For "Main" and "User", search by name and filters are available: cheese family, style, country, milk type, difficulty. The recipe list is paged; the "Back" («Назад») and "Forward" («Вперёд») buttons switch pages.

The "Open CheeseXML" («Открыть CheeseXML») button next to the search loads a recipe from a local .xml file on your computer - it works the same way as choosing a recipe from the catalog, except that the source shown in the preview panel is "Local CheeseXML" («Локальный CheeseXML»).



Cheese recipe catalog on the "Recipes" («Рецепты») page

- 1 Recipe source**
"Main" («Основные»), "User" («Пользовательские») or "Mine" («Мои») ("Mine" requires the Blynk token set in the device settings).
- 2 Search and filters**
Search by name and filters: cheese family, style, country, milk type, difficulty. Filters are unavailable for the "Mine" source.
- 3 Open CheeseXML**
Loads a recipe from a local .xml file on your computer instead of from the website catalog.
- 4 Recipe list**
Clicking a recipe loads it into the program preview panel.

Paged list

5

Switch list pages with the "Back" («Назад») / "Forward" («Вперёд») buttons.

Program preview

Clicking a recipe in the list (or loading a local file) opens it in the right panel: name, source, number of stages. If the recipe has a stage with a temperature range instead of a fixed value, a field for choosing one value from that range appears above the "Convert" («Преобразовать») button - without it, conversion is not possible ("Stage "...": choose one temperature value from the range." («Этап «...»: выберите одно значение температуры из диапазона.»)).

The "Convert" («Преобразовать») button builds a Samovar program from the recipe stages and shows it in a read-only text field, together with a list of warnings. For example: stages that the automation cannot perform (adding the starter culture by hand, cutting the curd, and so on) become manual actions with the message "Stage "... is not supported by the automation and was converted to a manual action." («Этап «...» не поддерживается автоматикой и преобразован в ручное действие.»); ripening stages are not included in the program ("Ripening stages are excluded from the Samovar program." («Стадии созревания исключены из программы Самовара.»)); if the stage timeout cannot be taken from the recipe, Samovar substitutes its own - a warning is shown about that too. A program cannot be longer than 30 stages - if it is, the message "The program has more than 30 stages; applying is unavailable." («В программе больше 30 этапов; применение недоступно.») is shown, and the program cannot be applied.

Program preview: choosing a temperature from a range, conversion and warnings

1

Recipe

Name, source (catalog, "My recipe" («Мой рецепт») or "Local CheeseXML" («Локальный CheeseXML»)) and number of stages.

2

Temperature choice

For stages with a temperature range instead of a fixed value, you must choose one value from the range.

3

Convert

Builds a Samovar program from the recipe stages.

4

Warnings

For example, which stages became manual actions and where the default timeout was substituted.

Program preview

- 5** The finished program lines, view only. No more than 30 stages.
- 6** **Apply program**
Available only in "Cheese" («Сыр») mode; sends the program to the device without starting the cheesemaking run.
- 7** **Open editor**
Goes to the "Program" («Программа») tab of Cheesemaking mode, where you can refine and save the program.

The "Apply program" («Применить программу») button sends the finished program to Samovar's memory - the same as "Set program" («Установить программу») on the "Program" («Программа») tab of Cheesemaking mode, without starting the cheesemaking run ("The program was accepted by the device. It was not started." («Программа принята устройством. Запуск не выполнялся.»)). The button is available only while Samovar is in "Cheese" («Сыр») mode and the conversion finished without blocking errors; otherwise one of these messages is shown under the buttons: "To apply, switch Samovar to "Cheese" mode." («Для применения переключите Самовар в режим «Сыр».») or "The conversion cannot be applied." («Преобразование нельзя применить.»).

The "Open editor" («Открыть редактор») link leads to the main page of Cheesemaking mode (the "Program" («Программа») tab), where you can view the applied program, refine it and save it in the usual way - see [Cheesemaking program](#).

pH calibration

The PH-4502C pH sensor outputs a voltage that Samovar reads on the [LUA_PIN](#) input as an ADC value (0-4095) and converts to pH with the formula $pH = k \cdot ADC + d$. Every electrode has its own coefficients k and d, so before the first brew and after replacing the electrode the sensor must be calibrated with two buffer solutions. Without calibration the pH field on the main page shows "no data" («нет данных»), and the "Waiting for pH" («Ожидание pH») stage in the program will not start.

The calibration page is opened with the "pH calibration" («Калибровка pH») button on the "Additional" («Дополнительно») tab of the [Cheesemaking](#) («Сыроварение») mode or on the [Settings → Cheese](#) («Настройки → Сыр») tab. The sensor is polled only in Cheesemaking mode - in other modes the page shows "no data".

Самовар Режим График Рецепты Настройки

Калибровка pH

К сыроварению Настройки

Текущее значение **1** 6.45 pH · ADC: 1830 **2**

Промойте электрод, поместите его в буфер и дождитесь стабильного ADC. Повторите для второго буфера.

pH первой точки: 7.00 **3** ADC: — **4** Зафиксировать **5**

pH второй точки: 4.00 **6** ADC: — Зафиксировать

Рассчитать **7**

Коэффициент k: -0.0035 **8**

Смещение d: 12.87 **9**

Сохранить калибровку **10**

Коэффициенты рассчитаны. Сохраните их. **11**

Без USE_ADS1115 PH-4502C подключается к LUA_PIN через согласование уровня до 3,3 В и конфликтует с MPX5010DP.

The pH sensor calibration page

- 1 Current value («Текущее значение»)**
The pH converted with the saved coefficients. "no data" («нет данных») - the sensor is not being polled or the result is outside 0-14.
- 2 ADC**
The raw reading of the LUA_PIN input (0-4095), updated once a second. The calibration points are captured from it.
- 3 pH of the first point («pH первой точки»)**
The pH of the first buffer solution (7.00 by default).
- 4 Point ADC («ADC точки»)**
The ADC value stored when the point was captured.
- 5 Capture («Зафиксировать»)**
Stores the current ADC together with the entered pH. Fresh data from the sensor and a pH from 0 to 14 are required.
- 6 pH of the second point («pH второй точки»)**
The pH of the second buffer solution (4.00 by default). The ADC of the second point must differ from the first.
- 7 Calculate («Рассчитать»)**
Calculates k and d from the two captured points and places them into the fields below.

- 8 Coefficient k («Коэффициент k»)**
Slope: the pH change per one ADC unit. Can be entered manually.
- 9 Offset d («Смещение d»)**
Offset: the pH at ADC = 0. Can be entered manually.
- 10 Save calibration («Сохранить калибровку»)**
Writes k and d to the Samovar settings. The allowed range is from -100 to 100.
- 11 Message («Сообщение»)**
The result of the last action: "Point 1 captured" («Точка 1 зафиксирована»), "Coefficients calculated. Save them" («Коэффициенты рассчитаны. Сохраните их»), "Calibration saved" («Калибровка сохранена») or an error text.

Calibration procedure

- . Rinse the electrode, put it into the first buffer solution and wait until the ADC value stops changing. The reading is updated once a second.
- . In the "pH of the first point" («pH первой точки») field enter the pH of the buffer (7.00 by default) and press "Capture" («Зафиксировать»). The stored ADC value appears next to it. If there is no data from the sensor or the pH is outside 0-14, the message "A fresh ADC and a pH from 0 to 14 are required" («Нужны свежий ADC и pH от 0 до 14») appears.
- . Rinse the electrode, put it into the second buffer (4.00 by default) and capture the second point in the same way. The ADC values of the two points must differ.
- . Press "Calculate" («Рассчитать»). The coefficients are calculated from the two points: $k = (pH2 - pH1) / (ADC2 - ADC1)$, $d = pH1 - k \cdot ADC1$ - and are placed into the "Coefficient k" («Коэффициент k») and "Offset d" («Смещение d») fields.
- . Press "Save calibration" («Сохранить калибровку»). The coefficients are written to the Samovar settings and are kept after a reboot. In firmware with the median filter, after the coefficients change you have to wait for five new measurements: then the "Current value" («Текущее значение») line shows the pH of the buffer the electrode is in.

You can also enter the coefficients manually, if they are known, and save them without capturing points. The allowed range of k and d is from -100 to 100. Before the first calibration the settings hold k = 1 and d = 1: the converted value falls outside 0-14, and Samovar treats the pH as unreliable.

Check. After saving, put the electrode into a buffer and compare the "Current value" line with the pH of the buffer. The same value is shown on the main page of the mode and at the "Waiting for pH" stage. In firmware with the median filter, the pH is determined from five consecutive measurements one second apart. The ADC field remains the raw reading for calibration: wait for it to settle before capturing each point.

Attention.

When the PH-4502C is connected to LUA_PIN, voltage level matching is required; the MPX5010DP must not be connected to the same input. When an ADS1115 is used, the LUA_PIN input stays free. Diagrams and maintenance are in the section [The pH sensor and how to connect it](#).

Working in manual mode

Manual control

For this, you can set a single program line. For example, this one:

H;65535;0.10;0;0;120

The volume in the line cannot exceed 65535 ml, and the first program line must specify the voltage regulator voltage (power), otherwise take-off will not start.

After the "Turn on heating" («Включить нагрев») command, Samovar tells the regulator to start boosting. When the set temperatures are reached, Samovar turns on the water, finishes boosting and moves on to column stabilization. After you press "Start take-off" («Начать отбор»), Samovar goes to the program line above and works by it until it has taken off a virtual 65.5 liters at 100 ml/h ($65.535/0.1 = 655$ hours); while it does this, you can control the take-off parameters. You can change the [take-off rate](#) and the [regulator voltage](#).

The safety group, holding the set water temperature and the "Rectification, distillation and BK up to boiler temperature" («Ректификация, дистилляция и БК до температуры в кубе») setting will also work.

The temperature delta and thermometer excursions beyond the setpoint are not taken into account. On the heads line the impurity detector only observes. The sorting table rocker arm will stay in position No. 0 all the time.

Note:

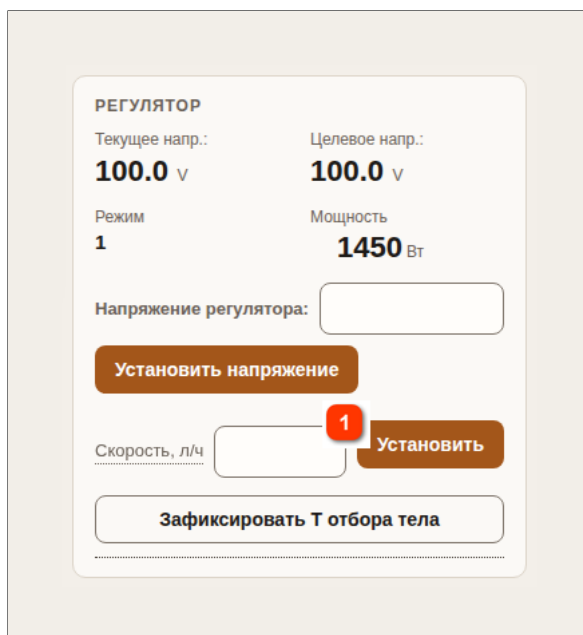
You can switch to automatic program execution at any moment. Just add a program line for the required mode, run the "Set program" («Установить программу») command and then press "Next" («Далее») in the main window.

Samovar will start executing the new program according to the built-in algorithm. Manual adjustment of the regulator voltage and take-off rate will still be available.

Manual Take-off Rate Setting

The take-off rate of the program that is currently running can be changed in the "Regulator" («Регулятор») card on the "Rectification" («Ректификация») tab ([main](#) page).

Samovar 7.00: the take-off rate field has been moved from the "Advanced" («Дополнительно») tab to the "Regulator" card on the main screen and is called "Rate, l/h" («Скорость, л/ч»).



The "Regulator" («Регулятор») card on the "Rectification" («Ректификация») tab

1

Value

Enter the take-off rate in liters per hour. Press "Set" («Установить») to apply it

The take-off rate can be changed at any moment while the program is running, any number of times. A manually set take-off rate has priority over the program one and is applied immediately after you press "Set" («Установить»). It stays in effect until Samovar moves on to the next program line. While the take-off program is not running, the command is not executed.

After the rate is changed, Samovar recalculates the remaining time of the current line, and for the impurity detector the new rate becomes the base one.

Tip.

So if you have "found" the right rate (say, for distilling aromatic raw material), you can adjust the next line so that the rate does not change at the transition. Don't forget to press "Set program" («Установить программу»).

Manual voltage setting of the regulator

You can change the voltage supplied to the regulator at any moment while the take-off program is running and the heating is on. Do it on the "[Main](#)" («Главная») tab, in the "Regulator" («Регулятор») card

РЕГУЛЯТОР

Текущее напр.: 100.0 V

Целевое напр.: 100.0

Режим 1

Мощность 1450 Вт

Напряжение регулятора:

Установить напряжение

Скорость, л/ч Установить

Зафиксировать Т отбора тела

The "Regulator" («Регулятор») card on the "Rectification" («Ректификация») tab

- 1 Manual input**
Enter a value in volts (in watts for the Stab-avr regulator) and press "Set voltage" («Установить напряжение»).
- 2 Target V («Целевое напр.»)**
The voltage (power) currently set on the regulator: from the program line or entered manually.

Enter a value in volts (for example: 150 or 171) and press "Set voltage" («Установить напряжение»). For the **Stab-avr** regulator the field is called "Regulator power" («Мощность регулятора»), the value is entered in watts, and the button is "Set power" («Установить мощность»). The manual voltage setting takes priority over the program one and stays in effect until Samovar moves on to the next line of the take-off program

Note: if you want the manually set voltage to apply to the next line of the take-off program as well, put 0 (zero) in the "Voltage" («Напряжение») field of that line

Note: when the next line of the take-off program starts, the regulator is set to the voltage specified in that program line, but the value entered by the operator manually is still shown in the input field. To find out which setting the regulator is following, compare two values: the input field and "Target V" («Целевое напр.») in the "Regulator" card.

Attention: in the "Pre-flooding" («Предзахлеб») mode the manual voltage change is applied, but Samovar keeps running the [operating algorithm](#) of the column in the pre-flooding mode.

Attention: in [NBK](#) mode, at the "Optimization" («Оптимизация») and "Work" («Работа») stages, the manual voltage change is not performed — Samovar shows the message "Manual control is not available during NBK Optimization and Work" («Ручное управление недоступно на Оптимизации и в Работе НБК»).

Working in monitoring mode

If you do not give the "Turn on heating" («Включить нагрев») command, Samovar shows the current temperatures from all sensors and the atmospheric pressure in the web interface.

The temperature readings are also sent to Blynk (every 5 seconds) and are visible in the mobile app.

This lets you monitor the temperature of any equipment, whether it is a column, a mash (fermentation) vessel, a pot-still reflux column or a still with indirect heating that is not controlled automatically.

WARNING!

In this mode **the safety group does not work**: the limit temperatures of the steam, water, boiler and the sensor on the additional input are not monitored. No logs are kept (neither the local log nor the one on the samovar-tool.ru server) and no charts are drawn.

This mode is not recommended, and the operator is fully responsible for the safety of the processes!

Extending Capabilities

Samovar's capabilities can be extended significantly with additional modules connected over the I2C bus and with additional scripts:

PCF8575 — an expander with 16 digital ports. It is enabled in the firmware with the line `#define USE_EXPANDER 0x20` (the module's address on the bus).

PCF8591 — an expander with four analog inputs and one analog output. It is enabled with the line `#define USE_ANALOG_EXPANDER 0x48`.

[I2CStepper](#) — a board with a stepper motor for a stirrer or a dosing pump and four relays. It is detected automatically when Samovar starts.

Programming of additional functions is implemented on the basis of the [scripting language Lua](#). Lua support is enabled when the firmware is built: in the configurator, with the "Use Lua" («Использовать Lua») checkbox (the "Hardware" («Оборудование») section); in PlatformIO, with the **Samovar_lua** environment; in a manual build, with the `#define USE_LUA` line in the `user_config_override.h` file. The script editor is opened with the "Editor" («Редактор») button on the Settings page.

Any actuators or buttons (sensors that work as buttons) can be attached to the port expander, and you can define yourself what should happen at a given moment while Samovar is running.

The initialization script runs when Samovar starts, and after that another script runs once a second. In the first you can describe the devices you need, in the second their behavior. For example, it can check the temperature or the current Samovar status and/or turn a pump on or off, open or close a valve, or control an additional heating element.

You can save variables in Samovar's memory, run up to 10 timers, get the Samovar operating status, control the stirrer, pump and relays of the I2CStepper board, and send HTTP requests. That is, you can connect additional equipment and control it automatically while Samovar is running.

You can check that a safety sensor has not tripped, and if it has, you can turn Samovar off right from the script, or increase or stop the water supply. There can be many scenarios. They are determined by the specific equipment.

The basic Samovar functionality does not change.

I2CStepper

Extending the capabilities of Samovar with the additional I2CStepper board with a stepper motor and four relays.

The module can be used as a standalone device. It can be controlled on its own with the menu on a 16x2 LCD screen and an encoder, or while connected to Samovar.

The expander is connected to Samovar over the I2C bus with four wires (like the LCD screen). Samovar detects that the expander is connected and can control it.

Samovar 7.00: the firmware works with the boards using protocol version 3. Up to ten boards with I2C addresses from 1 to 10 can be on the bus: boards with an **odd** address (1, 3, 5...) are **stirrers**, boards with an **even** address (2, 4, 6...) are **pumps** (continuous pumping or filling a set volume). Each board tells Samovar what it can do: stirrer, pumping, filling, relays, external sensor. The boards are searched for when Samovar starts and are then polled in the background; if at least one is found, an "I2CStepper" button appears on the main page of every mode and opens the board control page. A board that has once responded on the bus stays in the list until Samovar is rebooted; if communication with it is lost, it is marked as unavailable.

During operation (rectification program, Lua, Beer and Cheese modes) Samovar uses one **working stirrer** and one **working pump** — the boards with the lowest address among those found. They are fixed when the process starts and do not change until it ends, even if a board with a lower address appears during the run.

In Beer mode the first relay is used to switch the wort transfer pump on and off (for example, to circulate the wort in the brew kettle). The firmware does not use the other relays — you decide what they do, and you can switch them from Lua. The relays are served by the working stirrer if it has declared relay support, otherwise by the working pump.

The board settings themselves (mode, speed, stirrer run time and pause, pump flow, fill volume and rate, steps per milliliter) are stored in the Nano memory and edited on the **Settings** → **I2CStepper** tab («Настройки → I2CStepper»), see below. The "I2CStepper" page is used only for manual start and stop.

The "I2CStepper" page

The "I2CStepper" page: the pump board with address 2 is selected

- 1 Device («Устройство»)**
The list of boards found on the bus; shown if there are two or more boards. Stirrers have an odd address, pumps an even one.
- 2 Board state**
"Running" («Работает») or "Stopped" («Остановлен»); below it are the current pump speed and the remaining volume.
- 3 Speed and volume**
Pump speed in liters per hour and volume in milliliters for the "Run" («Запустить») button. "Stop" («Стоп») stops the pump.
- 4 Relay 1**
Shown if the board supports relays; switches the first relay of the board.



External pump calibration

Only for pump boards: opens the "Steps/mL" («Шагов/мл») calibration page for the selected board.

If two or more boards are found, a "Device" («Устройство») list appears at the top — use it to choose the board you work with: "Stirrer, address N" («Мешалка, адрес N») or "Pump, address N" («Насос, адрес N»). With a single board the list is hidden. If the selected board is unavailable, the message "The selected I2CStepper is unavailable" («Выбранный I2CStepper недоступен») is shown instead of the panel — another board is not substituted automatically, you have to choose it in the list.

For the selected board the page shows its type and address, its state ("Running" («Работает») or "Stopped" («Остановлен»)), the current speed in steps per second and how many steps are left before it stops. The data is refreshed every two seconds.

Control. "Speed" («Скорость») is the motor speed in steps per second (from 1 to 18000), "Target, steps" («Цель, шагов») is how many steps to make. The "Run" («Запустить») button sends these two values to the board and starts the motor in the mode stored in the Nano memory: for a stirrer it is rotation, for a pump it is filling with the given number of steps. For a pump it is easier to calculate the number of steps from the volume: multiply the required milliliters by "Steps/mL" («Шагов/мл») from the Nano settings. "Stop" («Стоп») stops the motor. The start is impossible if the pump is not calibrated ("Steps/mL" is zero) or Samovar is switching its mode at that moment — an error message is shown then.

Relays. If the board has declared relay support, a "Turn relay 1 on" («Включить реле 1») / "Turn relay 1 off" («Выключить реле 1») button is shown — it switches the first relay of the board (in Beer mode this is the wort pump). Relays 2–4 cannot be switched from this page, only from Lua.

External pump calibration. The button is available only for pump boards that support filling. It opens the calibration page for the selected board: prepare a measuring container, set the motor speed and press "Start calibration" («Начать калибровку») — the pump starts pumping. When exactly 100 mL has been collected, press "Set 100 mL" («Зафиксировать 100 мл»): Samovar calculates "Steps/mL" and writes the value to the Nano memory. Repeat the calibration after changing the driver microsteps, the motor, the pump head or the tubing.

Samovar settings

Settings → I2CStepper. The "Nano I2CStepper settings" («Настройки Nano I2CStepper») tab shows the settings stored in the memory of the board itself. The "Rescan I2C devices" («Пересканировать I2C-устройства») button searches the bus for boards again (for example, after connecting a new one). If there are several boards, choose the one you need in the "Device" list. Fields:

"New address" («Новый адрес») — the board address on the bus, from 1 to 10. An odd address makes the board a stirrer, an even one makes it a pump; an address that is already taken cannot be chosen. The new address takes effect after the Nano is rebooted.

"Mode" («Режим») — "Stirrer" («Мешалка», odd address only), "Pump" («Насос») or "Fill" («Наполнение», even address only). The board works in this mode when you press "Run" and on commands from Lua and the apps.

"Steps/mL" — the pump calibration coefficient: it is used to convert milliliters and liters per hour into motor steps. It is filled in by calibration (see above) or manually. While it is zero, the pump cannot be started.

"Stirrer speed" («Обороты мешалки»), "Stirrer run time, s" («Работа мешалки, с»), "Stirrer pause, s" («Пауза мешалки, с») — the stirrer parameters for standalone operation of the board; with a pause of 0 the stirrer runs continuously.

"Pump flow, mL/h" («Подача насоса, мл/ч») and "Pump pause, s" («Пауза насоса, с») — for the "Pump" mode (continuous pumping); "Fill volume, mL" («Объём наполнения, мл») and "Fill rate, mL/h" («Скорость наполнения, мл/ч») — for the "Fill" mode.

The "Save to Nano" («Сохранить в Nano») button writes the fields to the non-volatile memory of the board; they survive a restart of both Samovar and the board.

Settings → Pump («Настройки → Насос»). If an I2C pump is found, two fields for rectification appear on the tab:

"Use I2C pump above the pasteurization section" («Использовать I2C-насос над ЦП») — the pump takes part in rectification if it was detected when Samovar started: during heads take-off it becomes the main pump, during hearts and pre-flooding it works together with the local pump.

"Take-off rate above the pasteurization section" («Скорость отбора над ЦП») — the constant speed of the I2C pump on hearts lines B and pre-flooding lines C. On heads line H the speed of the program line itself is used.

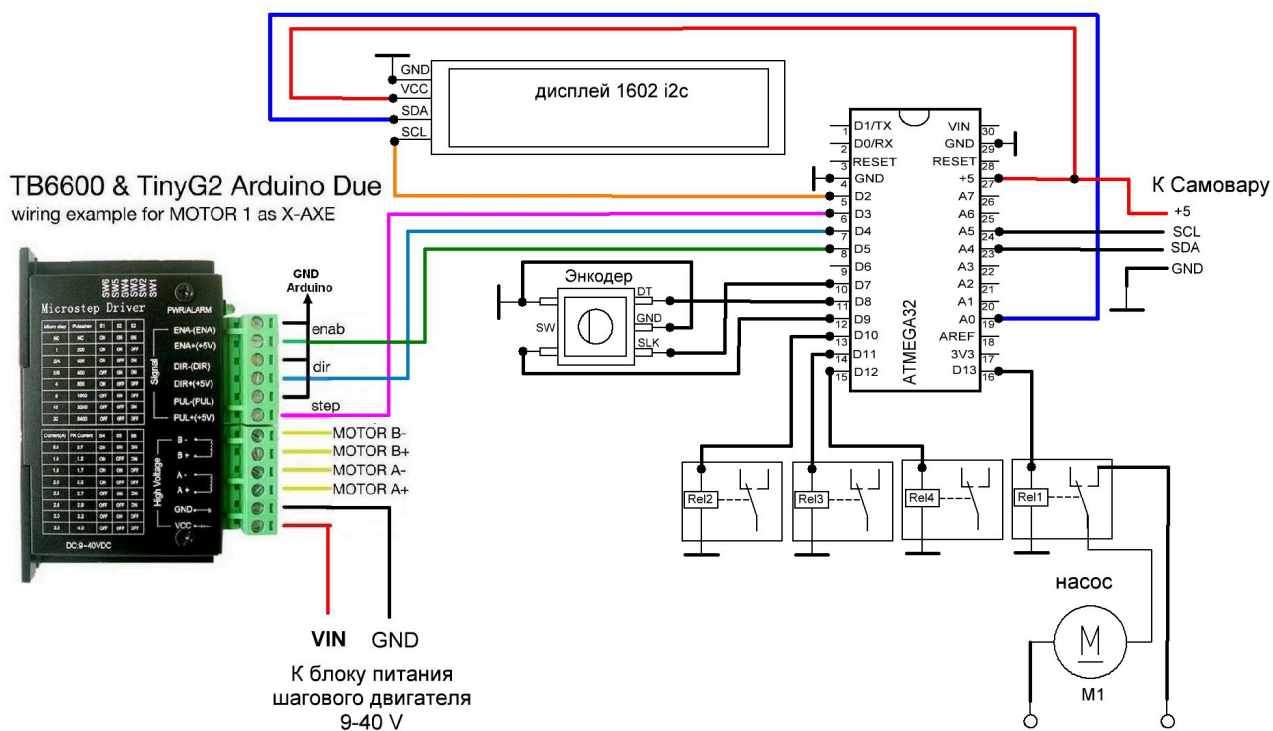
The speed in liters per hour for this pump is converted into steps using "Steps/mL" from the Nano memory; the separate "I2C pump steps count" («Количество шагов I2C насоса») field no longer exists in the Samovar settings.

Mobile apps and website

The Samovar apps and the Samovar control page on the samovar-tool.ru website receive the list of boards and their state (availability, mode, speed, remaining amount, relay mask) every five seconds and let you start and stop a board and switch the relays. The Nano settings and calibration are not available through the apps — only through the Samovar web interface. If a command is rejected, the reason arrives in the Samovar messages in the form "Blynk V37 a=N: code" («Blynk V37 a=N: код»).

Diagram and wiring

Wiring diagram



Pinout for connecting peripherals:

```
// Pins for I2C Master - connecting the screen and potentially other devices
// SDA_PIN 0 //A0
// SCL_PIN 2 //D2
```

```
// Pins for I2C Slave - communication with Samovar
// SDA_PIN A4
// SCL_PIN A5
```

```
// Pins for the stepper motor
#define STEPPER_STEP 3
#define STEPPER_DIR 4
#define STEPPER_EN 5
```

```
//Pins for the relays
#define MIXER_PUMP_PIN 13 // RELE_PIN1
#define RELE_PIN2 10 // RELE_PIN2
#define RELE_PIN3 11 // RELE_PIN3
#define RELE_PIN4 12 // RELE_PIN4
```

```
// Pins for the encoder
#define ENC_CLK 7 //S2
#define ENC_DT 8 //S1
#define ENC_SW 9 //KEY
```

Tip.

If you plan to use the expander as a separate device in its own enclosure, it makes sense to connect it to Samovar with a USB A-B cable (computer - printer). The module power (+/- 5 volts) and the two I2C bus signals are taken directly from the Samovar display. A USB-B connector is mounted on the Samovar enclosure.

A USB-A connector is mounted on the expander enclosure.

This way, when connected to Samovar, the Arduino Nano gets both power and the I2C bus at the same time. When used as a standalone device, a separate power supply is used instead of Samovar, for example a phone charger.

When a built-in power supply is used, a USB-B connector is fitted to the power supply output, and the same USB A-B cable is used to deliver power, which rules out accidentally feeding +5 volts into Samovar.

Peripherals.

The relay block is used at 5 volts, preferably with optocoupler control. You can also use 12-volt relays, wired the same way as in Samovar.

Any pump can be used: a DC motor or an AC single-phase or three-phase one. The pump motor is started and stopped through a relay and then through any power switch: a contactor or a solid-state relay. For small volumes, consider the 15RM magnetic-drive pump. Stainless steel, quiet, with good flow.

For the stirrer or the NBK, any bipolar stepper motor can be used, provided it has enough power and torque. A Nema 23 with a matching driver is recommended.

The current sketch for the Arduino Nano that controls the stepper motor and the four relays can be downloaded from this [link](#). You need the version with protocol v3; when moving from protocol v2 the board memory is initialized anew — check the address, the mode and "Steps/mL" again. Its main purpose is to be used together with Samovar as an expansion module. You must use the libraries supplied with Samovar.

Control from Lua

The functions are available when the firmware is built with Lua (see [About the Lua scripting language](#)). The stirrer functions work with the working stirrer, the pump functions with the working pump (the boards with the lowest address, see above). All start functions return 1 if the command was accepted by the board and 0 if not — including when the required board is not on the bus.

`set_stepper_by_time(spd, direction, time)` – start the stirrer: `spd` – speed in steps per second, `direction` – direction 0 or 1, `time` – run time in seconds. With `time = 0` the stirrer rotates continuously until a stop command, `spd = 0` stops the stirrer.

`set_stepper_target(spd, direction, target)` – start the pump in fill mode: `spd` – speed in steps per second (up to 18000), `direction` – direction 0 or 1, `target` – number of steps. With `spd = 0` or `target = 0` the pump stops. The function does not control the standard Samovar stepper motor.

`get_stepper_status()` – how many steps the pump still has to make (0 – it is stopped or there is no board).

`i2cpump_start(rate, volume)` – pump volume mL at rate L/h, converted to steps using "Steps/mL" from the Nano memory. Returns 0 if the pump board is not connected, not calibrated or the arguments are invalid.

`i2cpump_stop()` – stop the pump.

`i2cpump_get_speed()` – current pump speed in steps per second.

`i2cpump_get_target_ml()` – the volume set by the last `i2cpump_start` command, mL.

`i2cpump_get_remaining_ml()` – how much is left to pump, mL.

`i2cpump_get_running()` – 1 if the pump is running, otherwise 0.

`set_mixer_pump_target(Val)` – switch the first relay – the wort pump – on (1) or off (0).

`get_mixer_pump_status()` – state of the first relay: 0 or 1; 255 if there is no board with relays.

`set_i2c_rele_state(Num, Val)` – switch relay `Num` (1–4) on (1) or off (0). Returns 1 if the command was executed.

`get_i2c_rele_state(Num)` – state of relay `Num` (1–4): 0 or 1; 255 if there is no board with relays.

`check_I2C_device(Addr)` – check whether the device with address `Addr` responds on the I2C bus; returns the address if the device is found.

Attention. *The relays are served by the working stirrer if it has declared relay support; otherwise by the working pump. Functions that change the state are not executed in Lua simulation mode and while Samovar is switching its mode.*

Example: `set_stepper_by_time(800, 0, 0)` – rotate the stirrer continuously at 800 steps per second; `set_stepper_by_time(0, 0, 0)` – stop it. `i2cpump_start(6, 250)` – pump 250 mL at 6 L/h; while `i2cpump_get_running() == 1`, the pump is running.

Additive dosers (Beer and Cheese)

In Beer («Пиво») and Cheesemaking («Сыроварение») modes, additives must be added as the program runs: hops in beer (usually 2-4 times per boil), and starter culture, enzyme, calcium chloride, salt and spices in cheese. Samovar can do this itself or prompt you at the right moment. Below are four built-in ways to implement a doser, from the simplest to the most flexible, with what each needs and how it is set in the program, followed by an overview of homemade designs used by brewers around the world and how to connect them to Samovar.

General information about the programs is in the sections [Mashing program](#) and [Cheesemaking program](#); the expansion board is described on the page [I2CStepper](#); the scripts are in the section [Lua](#).

Option 1. Manually, on a Samovar signal



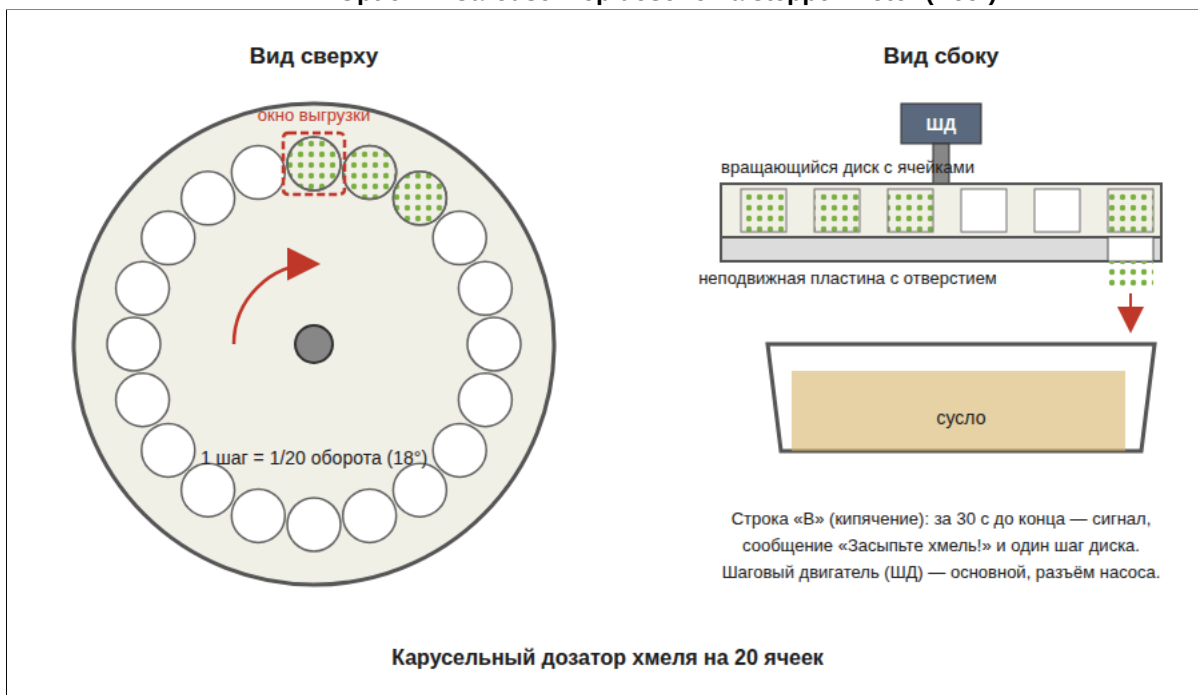
No hardware is needed. At the right moment Samovar sounds the buzzer, shows a message on the display and in the web interface, and sends a push notification to the mobile app and the website. You add the additive yourself.

Beer. 30 seconds before the end of every "B" line (boil), Samovar sends the message "Add the hops!" («Засыпьте хмель!»). There are as many "B" lines as hop additions: for example, a 60-minute boil with hops at 60, 30 and 5 minutes before the end is written as "B" lines of 30, 25 and 5 minutes. The program does not wait for confirmation and moves on.

Cheese. A "W" line (manual action) with the action "Add starter culture" («Внести закваску»), "Add enzyme" («Внести фермент») or "Add additive" («Внести добавку»): Samovar stops the program (the stirrer, if set, keeps running) and waits for the "Confirm" («Подтвердить») button to be pressed on the mode page, in the app or on the website. A "D" (dosing) line with the "Manual" («Вручную») method does the same; it additionally records the component and volume.

Pros: nothing to build. Cons: you must be nearby; in beer the exact timing depends on you.

Option 2. Carousel hop doser on a stepper motor (Beer)



This is the built-in automatic dry-additive doser for Beer mode. Together with the message "Add the hops!" Samovar turns the main stepper motor (the one that drives the take-off pump in other modes) by exactly 1/20 of a turn (18°). A disc with cells is mounted on the motor shaft, and below it is a fixed plate with a single hole above the boiler. After each turn the next cell is over the hole, and its contents fall into the wort.

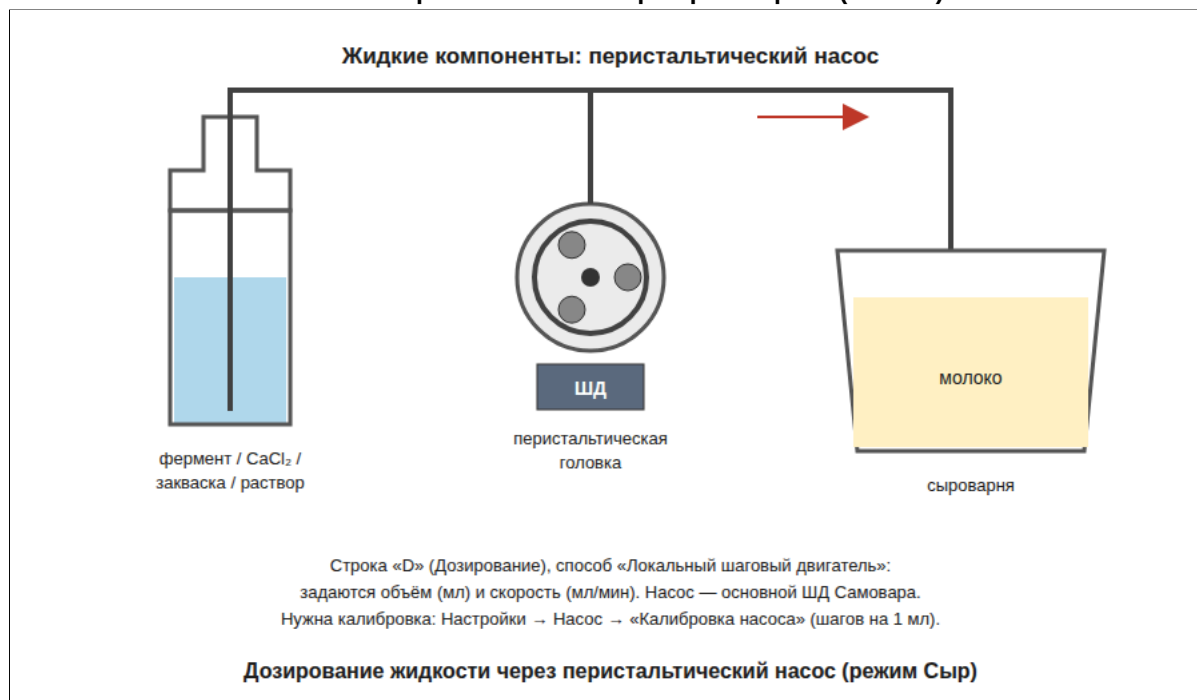
What you need. A NEMA17 stepper motor (connected to the Samovar pump connector, no separate driver or settings required), a disc made of food-grade plastic or stainless steel 150-200 mm in diameter with 20 cells around the circle (or fewer, see below), a bottom plate with one hole, and a lid so the steam from the boiling wort does not dampen the hops. The disc is convenient to 3D-print. Hops are put into the cells in the order of rotation, starting with the cell next to the discharge window.

How it is set in the program. Nothing extra: each "B" line is one addition. The step is made at the end of the line, so the first portion (for example, "hops at 60 minutes") drops at the end of the first "B" line, which can be made short, for example 1 minute. The driver microstepping mode must be 1/16, as for the pump.

If there are not 20 cells. The angle of one step is set in the firmware in the HopStepperStep() function (file beer.h): TargetSteps = 360 / 1.8 * 16 / 20, where 20 is the number of cells in a full circle. For a 10-cell disc, replace 20 with 10 and reflash Samovar. Instead of a disc you can make a drum with longitudinal compartments on a horizontal shaft; the calculation is the same.

Pros: fully automatic brewing, no electronics beyond the standard. Cons: dry additives only; this step is not performed in Cheese mode (the motor is busy with the pump there).

Option 3. Peristaltic pump for liquids (Cheese)



In Cheesemaking mode, the main Samovar stepper motor works as a dosing pump. A "D" (dosing) program line with the "Local stepper motor" («Локальный шаговый двигатель») method delivers the set volume in milliliters at the set rate in ml/min: for example, 25 ml of calcium chloride solution in 1 minute or 50 ml of diluted enzyme in 2 minutes. When delivery ends, the program moves to the next line by itself; if the volume is not delivered within the "Timeout, min" («Тайм-аут, мин») time, cheesemaking stops with a message.

What you need. A peristaltic head for a NEMA17 stepper motor (the same as for take-off in rectification) and food-grade silicone tubing. The inlet end of the tube is lowered into the container with the solution, the outlet end is led into the cheese vat above the milk level. Prepare the solution in advance; starter culture and enzyme are diluted in boiled water, so that the volume is large enough for the pump to deliver it accurately.

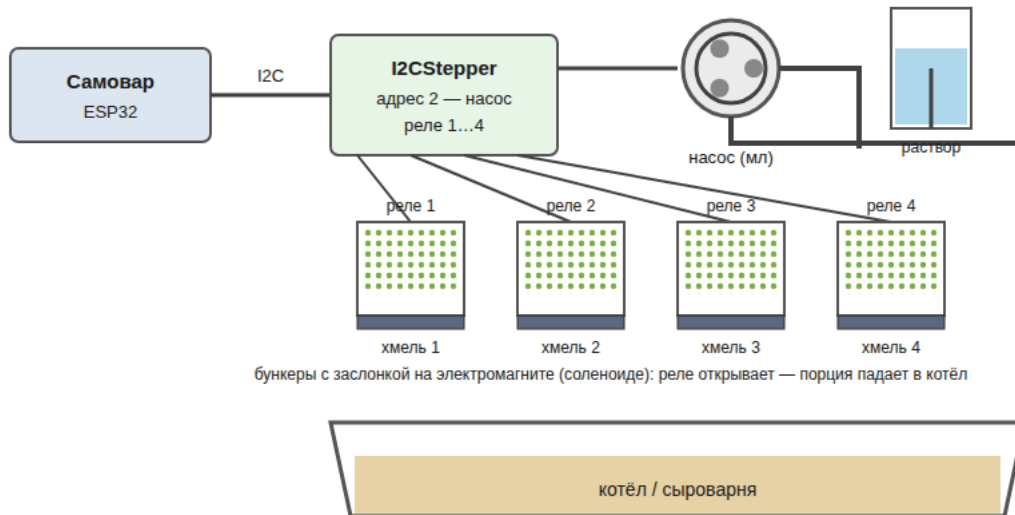
Calibration. The volume is converted to steps using the "Pump step count" («Количество шагов насоса») value (steps per 1 ml). It is set on the Settings → Pump tab with the "Pump calibration" («Калибровка насоса») button (see [Pump](#)): the pump pushes liquid into a measuring container, you enter the actual volume, and Samovar saves the coefficient. Calibrate with the same tube and the same liquid you will dose, and fill the tube beforehand, otherwise the first portion goes to filling it.

Several components. One pump, one solution. If two or three different solutions must be added during a brew, put a "W" line "Add additive" («Внести добавку») between the "D" lines and move the inlet tube to another container (rinsing it), or use a second pump on the I2CStepper board (option 4).

Pros: exact volume and time, no manual work. Cons: liquids only; one component per pump.

Option 4. I2CStepper board: pump and relays from a Lua script (Beer and Cheese)

Плата I2CStepper: насос и реле для бункеров с заслонками



Управление из Lua-строки программы («L»): `set_i2c_rele_state(n, 1)` и `i2cpump_start(rate, volume)`.

Дозирование через плату I2CStepper (Пиво и Сыр)

The [I2CStepper](#) board gives Samovar a second stepper motor (address 2 - the pump) and four relays. The firmware does not dose with them by itself, but all of it is available from a Lua script started by the "L" program line: in the line you choose a .lua file and give it parameters (for example, a relay number and a volume), so one script serves all additions. The script switches on the pump or relay, waits for it to finish and calls `setNextProgram()` to move to the next line.

Liquids - the I2C pump. The `i2cpump_start(rate, volume)` function pumps volume ml at rate l/h; the steps per 1 ml are taken from the "I2C pump step count" («Количество шаров I2C насоса») field (Settings → Pump), calibration is on the "I2CStepper" page. An example script `dose_pump.lua` that adds the volume from the first parameter of the line (in the program: file `dose_pump.lua`, parameter 20) at 6 l/h:

```
-- dose_pump.lua: add arg[1] ml of solution with the I2CStepper pump
if i2cpump_start(6, tonumber(arg[1])) == 1 then
    while i2cpump_get_running() == 1 do delay(500) end
    sendMsg("Solution added", 2)
else
    sendMsg("I2CStepper pump is not responding", 1)
end
setNextProgram()
```

Dry additives - relays and hoppers with a gate. For each portion of hops (or salt, spices) a small hopper is made above the boiler, whose bottom is closed by a gate on a 12 V electromagnet (solenoid). The board relay powers the electromagnet, the gate opens, and the portion falls into the boiler. Four relays, four portions; each is measured out in advance, when loading. An example script `dose_rele.lua` that opens the hopper gate on the relay whose number is in the line parameter (in the program: file `dose_rele.lua`, parameter 3) for one and a half seconds:

```
-- dose_rele.lua: drop a portion from the hopper on relay arg[1]
set_i2c_rele_state(tonumber(arg[1]), 1)
delay(1500)
set_i2c_rele_state(tonumber(arg[1]), 0)
sendMsg("Hops added", 2)
setNextProgram()
```

For each addition the program gets its own "L" line with the same file and its own parameter (relay number or volume); the line timeout is set in seconds.

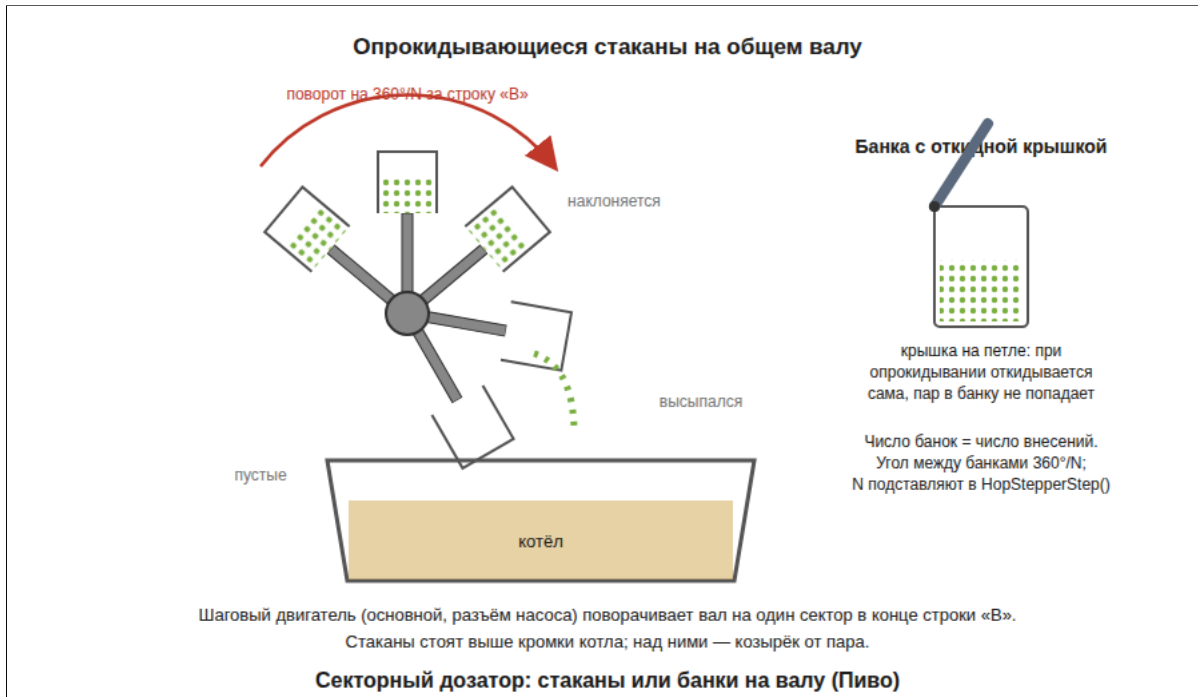
Which relays are free. In Beer mode the first board relay is taken by the wort transfer pump (see the "I2CStepper" page); the firmware does not use the other three. In Cheesemaking mode the firmware does not occupy the board relays (the stirrer and the drain valve sit on Samovar's own relays 2 and 4). You can check that the board responds with the `check_I2C_device(2)` function.

A peculiarity in Beer. The "L" line is a separate program line, and boiling is not automatically maintained on it, so make it short (a few seconds); the next "B" line detects boiling again - the wort is still boiling by then, so this is quick. If you only need hops on a schedule, option 2 is simpler: it needs neither a board nor a script.

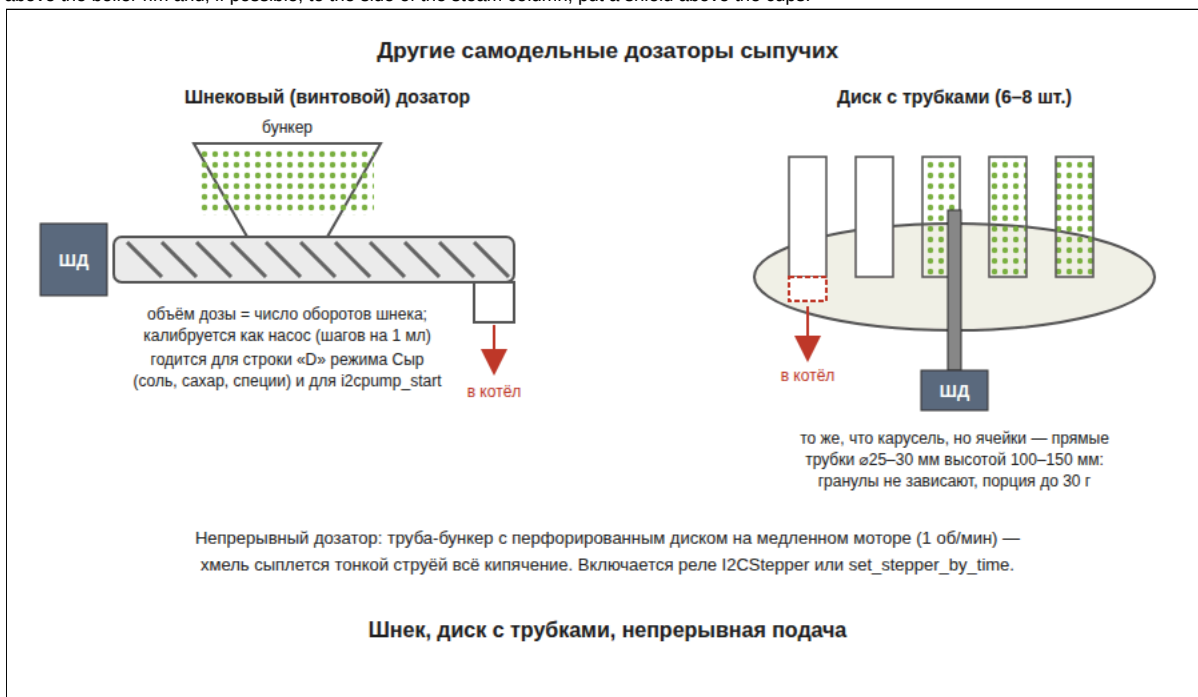
Pros: any number of components, liquid and dry, complete freedom in logic. Cons: you need an I2CStepper board, firmware with Lua support and a script.

Option 5. Homemade designs: what else brewers do

Commercial automatic breweries solve the same task in different ways: Brewie and PicoBrew Zymatic keep hops in four baskets and at the right moment run wort through the next basket; MiniBrew puts a carousel with six cylinders on the lid; Grainfather and Braumeister do not add hops, they only send a notification (like option 1). In home workshops (Homebrew Talk forums, projects for CraftBeerPi and BrewTroller, Brew Your Own articles) the designs below are the most common. All of them connect to Samovar in one of the ways already described: a step of the main stepper motor on a "B" line, a "D" line in cheese, or an I2CStepper board relay or motor from Lua.



Sector doser: cups or jars on a shaft. N cups (plastic cups, cut-off bottles, small jars with a hinged lid) are rigidly fixed on the horizontal shaft of a stepper motor in a fan, every $360^\circ/N$. While a cup is in the upper half of the circle, it stands open side up; a turn to the next sector moves the next cup past the horizontal, and the portion falls into the boiler. A jar with a hinged lid is more convenient than an open cup: until it tips, the lid protects the hops from steam, and when it tips it opens under its own weight. The advantage is simplicity: no precise cells and no bottom plate are needed. For Samovar: the motor goes on the pump connector, in `HopStepperStep()` substitute the number of cups N for 20 (for 4 cups one step is 90°), and each "B" line tips one cup. The shaft must be above the boiler rim and, if possible, to the side of the steam column; put a shield above the cups.



Disc with tubes ("Lazy Susan"). The most common homemade design on Homebrew Talk: a disc 150 mm in diameter with 6-8 vertical tubes ($\varnothing 25\text{--}30$ mm, height 100-150 mm, holding about 30 g of pellets), with a fixed plate with one hole under it; the drive is a high-torque servo or an inexpensive 28BYJ-48 stepper with a ULN2003 driver, and for an exact stop a microswitch or an optical sensor on a cam is installed. It is the same carousel as in option 2, only with tall straight tubes instead of small cells. The authors' experience: pellets hang up in tubes with a conical bottom, so the walls must be straight and smooth; powdered and wet hops do not pour, dry pellets are needed; a shield or steam outlet above the boiler is a must, otherwise after 20-30 minutes of boiling the pellets get damp and clump. With Samovar it is used as option 2, the number of tubes is substituted in `HopStepperStep()`; if desired, the disc can be put on an I2CStepper board motor and turned from Lua with the `set_stepper_by_time` function.

Auger (screw) doser. A design from automatic fish and cat feeders (see projects on Instructables and GitHub): a hopper above a horizontal tube, with an auger on a stepper motor shaft inside the tube; the dose volume is set by the number of turns. The parts are 3D-printed, the auger is 20-30 mm in diameter. It doses any free-flowing product from one hopper in portions of any size: salt, sugar, spices, hop pellets. For Samovar such a doser is the same "pump": in Cheese mode it goes on the pump connector, is calibrated with the "Pump calibration" («Калибровка насоса») button (the product is poured into a measuring cup and its volume in ml is entered) and is dosed with a "D" line with the "Local stepper motor" («Локальный шаговый двигатель») method. A second identical doser can be hung on the I2CStepper board (address 2) and started from Lua with the `i2cpump_start(rate, volume)` function. Con: one hopper, one product; for 3-4 different hop varieties you need 3-4 augers or a carousel.

Continuous hop feeding. For varieties with "continuous hopping" (hops added in a thin stream throughout the boil), Brew Your Own describes a doser made of a piece of $\varnothing 100$ mm pipe 60 cm long, with a perforated disc rotating at the bottom end driven by a 1 rpm motor; the pellets fall at an average rate of about 8 g per 10 minutes, and pins on the disc keep them from forming an arch. With Samovar the motor is switched on by an I2CStepper board relay (set_i2c_rele_state) for the whole "B" line, or turned by the board stepper motor (set_stepper_by_time).

Gate on a servo or gear motor. In CraftBeerPi projects (the HopDropper plugin) a tray with sections is moved by a slow 12 V gear motor through a pair of gears, and the controller only supplies power for a set time; another popular scheme is a cup on a servo arm hung on the boiler wall away from the steam. The servo connects to the Samovar SERVO_PIN output (the same one that switches containers in rectification) and turns from Lua with the setServoAngle(angle) function: for example, setServoAngle(90) opens the gate, delay(1500), setServoAngle(0) closes it. Power a high-torque servo from a separate 5-6 V source, not from the board. A 12 V gear motor mechanism is switched on by an I2CStepper board relay for 1-3 seconds (the time a turn to the stop or to a cam with a microswitch takes). Control is as in option 4. An example script tip_cup.lua with a cup on a servo arm (in the program, an "L" line with this file):

```
-- tip_cup.lua: tip the cup on the servo arm
setServoAngle(150)
delay(1500)
setServoAngle(0)
sendMsg("Hops added", 2)
setNextProgram()
```

Baskets rinsed with wort. The Brewie/Zymatic principle: hops lie in mesh baskets outside the boiler, and the wort transfer pump directs the flow through the next basket at the right moment. At home this is the most complicated way (you need a pump, several solenoid valves on I2CStepper relays and plumbing), but the hops are guaranteed to be protected from steam and nothing has to be poured above the boiler. For Samovar the valves are switched from Lua with the set_i2c_rele_state function, and the pump by the first board relay (in Beer mode it already controls the wort transfer pump).

General lessons from others' designs.

The main enemy of any dry doser is steam: put a shield above the boiler, move the doser to the side, and better still close the portions with lids until the moment of addition (jars with hinged lids, gates).

Pellets do not pass well through narrowings and cones: make cells, tubes and trays with straight walls, a flat bottom, and a discharge window no smaller than the cell cross-section.

Measure out portions in advance when loading, and let the automation only choose the moment; then accuracy does not depend on the mechanics.

Before brewing, run the doser empty in test mode, and do the first brew with it while standing nearby: the "B" lines will send a notification anyway, and a portion can be added by hand.

Sources of ideas:

[Lazy Susan hop dropper \(Homebrew Talk\)](#), [Automatic pellet hop dropper \(Homebrew Talk\)](#), [3D printed hop carousel \(Homebrew Talk\)](#), [Automated hop addition \(Homebrew Talk\)](#), [HopDropper for CraftBeerPi \(GitHub\)](#), [Continuous wort hopper \(Brew Your Own\)](#), [Auger micro-doser \(Instructables\)](#), [MiniBrew Hop Carousel, overview of automatic breweries \(Brew Your Own\)](#).

What to choose

Beer, hops 2-4 times - option 2 (a carousel on the built-in motor) or its simplified version from option 5, cups or jars on a shaft. Until the doser is built, option 1: the "B" lines already send a signal.

Cheese, starter culture / enzyme / CaCl_2 - option 3 (the "D" line, the built-in pump). One pump, one component per brew; the rest are added by hand with a "W" line. Salt and dry spices: the auger doser from option 5 on the same connector.

Several liquid and dry components automatically - option 4 (I2CStepper and Lua).

General tips: keep additives under a lid (steam and condensation are the main enemy of dry dosers), test the doser beforehand on water and in test mode, and make liquid volumes no less than 10-20 ml, so the pump error is not noticeable.

Autoclave control

A guide to using Samovar to control an autoclave.

Working temperature.

For autoclaves it is in the range of **110-120 degrees**. The difference seems small, but it is critical for some kinds of products. Vegetables can be preserved even at 110 degrees. Fish needs 115, and meat needs a full 120 degrees.

In fact, Samovar's work with an autoclave comes down to one thing: using the voltage regulator to control heating and, as a result, the temperature inside.

A possible problem is choosing a good-quality DS18B20 sensor.

According to the specification this sensor works up to 125°C, but in real life far from every one works normally at 120°C. It may happen that sensors work wonderfully in distillation and rectification, and fail when used in an autoclave.

Practical tip.

It is recommended to make the sensor yourself, buying several DS18B20 chips from a well-known store from a supplier with good reviews. The chance of making a good-quality sensor yourself that withstands operation at 120 degrees is much higher than of buying a ready-made one. Ready-made sensors often fail at elevated temperatures. When making it, it is important to provide a thermal interface between the sensor and the surface of the sleeve. For this it is convenient to use a neutral heat-resistant sealant. Do not use an acidic sealant: it will corrode the wire. You can use hot-melt thermal glue, but if something happens to the sensor, thermal glue is much harder to remove.

Important: the inside of the autoclave sleeve where we place the sensor must also be filled with a thermal interface. Glycerin is the most convenient for this.

To work with an autoclave, switch to the "Beer" («Пиво») mode.

1. Check:

Settings («Настройки») / Temperature («Температура») / "Boiler sensor address" («Адрес датчика в кубе»). Make sure the address of the correct sensor is set here.

2. Settings / Beer («Пиво»).

Leave the "Brewing order" («Варочный порядок») at All-in-one / BIAB: in this option heating is regulated by the boiler sensor. HERMS and RIMS control the temperature by the water and column section sensors and are not suitable for an autoclave.

"Regulator voltage/power in temperature holding mode: ____ v («Напр./мощность регулятора в режиме поддержания темп.: ____ v»)." This parameter needs to be tuned and the value chosen for your equipment. It is the minimum voltage/power that is enough for the autoclave temperature to keep rising after boosting is switched off. By default boosting is switched off when the temperature is 1 degree below the one set in the program (this can be changed in the file `Samovar_ini.h`, the line `#define HEAT_DELTA 1`). For that one degree, the autoclave should keep heating slowly with the voltage/power value set in this field. To start, you can set 100 volts.

3. Program.

We use the malt mashing program. Here we set the temperature and the time to hold this temperature.

These values may change slightly depending on the products being prepared. The classic is 120 degrees for one hour.

Set the PAUSE («ПАУЗА») mode, temperature 120 degrees, time 60 minutes.

Practice has shown that the chance of damaging jars and of products boiling over is lower if you heat to 120 degrees not as fast as possible in boost mode, but gradually, roughly over half an hour, for example in "steps" at 105, 110 and 115 degrees with a hold of 3 - 5 minutes each.

The result is this program:

P;105;4;0^0^0;0

P;110;4;0^0^0;0

P;115;4;0^0^0;0

P;120;60;0^0^0;0

The last field of the line is the number of the sensor by which the temperature is held: 0 is the boiler sensor. The device field 0^0^0 means that the stirrer and the pump are not used in the line.

Safety.

To ensure safety you can use the following methods:

- Use a bimetallic thermostat KSD9700 (NO - normally open) rated 55-70°, installed near the autoclave's emergency valve.

Connect the thermostat outputs in parallel with Samovar's emergency button (applying a logical one to ALARM_BTN_PIN: D35 for the DEVKIT board, D48 for the ESP32-S3; the firmware must be built with the emergency button, that is the `Samovar_alarm_button` environment or `#define USE_ALARM_BTN`). When steam is released, Samovar will tell the regulator to turn heating off.

- Replace (or modify yourself) the pressure gauge with an electric contact one set for a maximum pressure of 0.25 MPa and use its contacts to connect in parallel with Samovar's emergency button.
- Connect to the emergency valve a unit that provides [additional protection](#) by means of a contactor or an RCD. If steam breaks through, Samovar will switch off, and so will the heating.

The setup is finished; press the "Turn on heating" («Включить нагрев») button and enjoy the automatic process until it ends.

When the process ends, Samovar will tell the regulator to turn heating off. Wait for the autoclave to cool down and take out the finished product.

Note.

The temperature and pressure in an autoclave are directly related. As heating increases, so does the pressure. Accordingly, the temperature regime can be calculated and checked from the working pressure and vice versa.

For example, for an autoclave to reach a temperature of 120 degrees, excess pressure is required.

There are tables of temperature versus pressure on the internet. You can use them quite well, taking into account the correction for your location.

About the Lua scripting language.

Samovar supports automation based on the scripting programming language [Lua](#). Lua support is enabled when building the firmware: in the configurator — the "Use Lua" («Использовать Lua») checkbox ("Hardware" («Оборудование») section), in PlatformIO — the **Samovar_lua** environment, in a manual build — the line **#define USE_LUA** in the `user_config_override.h` file. The script editor is opened with the "Editor" («Редактор») button on the Settings page (address <http://samovar.local/edit>). No login or password is requested. The current Lua version is 5.4.4

When Samovar starts, the `init.lua` file runs. Its purpose is to enable/disable the one-second script execution loop and to initialize variables that will be used in other scripts. As shipped, `init.lua` disables the loop: `setNumVariable("loop_lua_fl",0)`. If running scripts in a loop is enabled, two scripts are run every second: first `script.lua`, then the script whose name depends on the operating mode: `rectificat.lua` (Rectification), `dist.lua` (Distillation), `beer.lua` (Beer), `bk.lua` (Wash column), `nbk.lua` (NBK), `suvid.lua` (Sous-vide), `cheese.lua` (Cheesemaking). In "Lua mode" («Lua-режим») (an item in the mode list in Settings, visible only in a build with Lua) there is no mode script — all the mode logic is written in `script.lua`. A pair of scripts can also be run once by going to <http://samovar.local/lua>, and a separate file at http://samovar.local/lua?script=file_name. The scripts run one after another in a single task: while one is running, the other waits. A single script run cannot last longer than 20 seconds — after that it is aborted with the message "Lua: chunk execution aborted by timeout" («Lua: выполнение чанка прервано по таймауту»). If a mode script ends with an error 5 times in a row, the one-second loop stops (`loop_lua_fl` is reset to 0) and one summary message is printed to the console; `script.lua` stops being run after 5 errors in a row until the scripts are reloaded, without stopping the loop. Script errors are written to the browser console and to the port monitor.

You can also add buttons to the interface to run files. The file name: `btn_<mode>_button<N>.lua`, where mode is `rect`, `dist`, `beer`, `bk`, `nbk`, `suvid` or `cheese`, and N is the button number. For example, `btn_rect_button1.lua`, `btn_beer_button2.lua`. The buttons are shown on the "Additional" («Дополнительно») tab of the main page of the mode they were created for, in the Lua block; there is also the "Lua:" («Lua:») field with the "Run Lua" («Выполнить Lua») button for running one or two lines of code, and "Lua status" («Статус Lua») — a line that the script sets with the `setLuaStatus` function. The button name is set in the first line of the file by a comment of the form `--|Name^` (for example, `--|Start^`). If there is no such line, the buttons are named `LUA1`, `LUA2`, and so on, in the order of the files found.

As shipped, each mode has two buttons: "Start" («Начать») (`btn_*_button1.lua` — enables the one-second loop) and "Stop" («Остановить») (`btn_*_button2.lua` — disables it).

The main purpose of the buttons is to start running scripts in a loop (for example, by setting a variable value that can then be read in the script running in the loop), or to switch a control device (for example, a pump) on or off.

So the architecture of using the buttons looks as follows (as an example - for distillation mode):

- The main controlled loop: once a second `script.lua => dist.lua => script.lua => dist.lua => ...` (controlled in the sense that it can be started/stopped programmatically from any script through a global variable)
- The `script.lua` script is common to all modes; you can put here code that must run in any mode (rectification, beer, sous-vide, etc.)
- The `dist.lua` script will be used only in distillation mode; you can put in it all the logic that must run once a second depending on parameters this script gets from the Samovar global variables, for example `getNumVariable("TankTemp")`, or from variables saved by any script in the common namespace, for example `getObject("StartTankFilling")`. That is, in the code of this script you can put, for example, the condition: `if (StartTankFilling == "true" and TankFillingPercent < 80) then StartPump() else StopPump() end` and every second the script will check whether there is a flag that the boiler needs to be filled, and if there is, whether it is already 80% full, and whether the previously started pump should now be stopped.
- The button script `btn_dist_button1.lua`, which, when the button is pressed, sets the flag that the boiler needs to be filled via `setObject("StartTankFilling", "true")`. On the next cycle a second later, the `dist.lua` script will read this flag and switch the pump on (if the boiler is not yet filled). Thus all scripts share a common namespace, available through `setObject/getObject`, which lets them exchange data.
- The button script `btn_dist_button2.lua`, which, when the button is pressed, can stop the main controlled loop via `setNumVariable("loop_lua_fl", 0)`, for example if something goes wrong with the scripts.

A program line of type L (a Lua stage). In the programs of all modes except NBK (Rectification, Distillation, Wash column, Beer, Cheesemaking) there is a line of type L — a stage controlled by Lua. When this type is selected in the program editor, the "Lua stage" («Lua-этап») window opens, in which you set:

Lua file — any file with the `.lua` extension from the Samovar memory (the list is taken from the file editor). You can upload your own script through the web interface file editor.

Stage timeout, sec — from 1 to 65535 seconds. If the script has not called `setNextProgram()` within this time, Samovar shows the message "Lua did not finish the operation before the timeout" («Lua не завершила операцию до тайм-аута») and ends the mode.

Parameters — any number of strings passed to the script. In the script they are available in the `arg` table: `arg[0]` is the file name, `arg[1]`, `arg[2]`... are the parameters in order. An empty parameter is written as "".

When the program reaches an L line, Samovar runs the selected file and waits for a `setNextProgram()` call — this is the move to the next line. A script error ("Lua finished with an error" («Lua завершилась с ошибкой»)) or a timeout stops the mode. The other fields of the line (temperature, rate, container, device) are not set for type L. In text form such a line is stored as the file and parameters separated by the ^ character, for example `L;120;hops.lua^3^45;0;0;0` for rectification.

An example of the `dose.lua` script, which switches on the relay whose number is in the first parameter for the time in the second (in the program: file `dose.lua`, parameters 3 and 1500):

```
set_i2c_rele_state(tonumber(arg[1]), 1)
delay(tonumber(arg[2]))
set_i2c_rele_state(tonumber(arg[1]), 0)
setNextProgram()
```

If during operation you upload through the file editor a new version of the file that is currently being run by an L line, Samovar will re-read the script without restarting the program.

Samovar functions

From scripts you can access Samovar's internal variables (some of them cannot be changed, they are read-only) and call Samovar functions. While a mode switch is in progress, functions that change the Samovar state end with the error "mode switch blocks state changes".

Pin numbers depend on the board. Here and below, pass their numeric value:

ESP32 DEVKIT: RELE_CHANNEL1 – 2, RELE_CHANNEL2 – 15, RELE_CHANNEL3 – 14, RELE_CHANNEL4 – 13, WATER_PUMP_PIN – 4, LUA_PIN – 34, ALARM_BTN_PIN – 35.

ESP32-S3: RELE_CHANNEL1 – 2, RELE_CHANNEL2 – 42, RELE_CHANNEL3 – 41, RELE_CHANNEL4 – 40, WATER_PUMP_PIN – 11, LUA_PIN – 4, ALARM_BTN_PIN – 48.

LILYGO T-Relay: RELE_CHANNEL1 – 21, RELE_CHANNEL2 – 19, RELE_CHANNEL3 – 18, RELE_CHANNEL4 – 5, WATER_PUMP_PIN – 2, LUA_PIN – 12. The constants INPUT, OUTPUT, LOW, HIGH and the command result codes ACTUATOR_COMMAND_ACCEPTED, ACTUATOR_COMMAND_PENDING, ACTUATOR_COMMAND_APPLIED, ACTUATOR_COMMAND_FAILED are defined in scripts.

pinMode(pin, mode) – like the Arduino function of the same name, sets the operating mode of the given input/output (pin) as an input or an output. Changing the mode is available only for the ports RELE_CHANNEL1–RELE_CHANNEL4 and LUA_PIN. That is, if you need to set the RELE_CHANNEL4 port of the DEVKIT board as an input, call pinMode(13, INPUT); as an output - pinMode(13, OUTPUT).

digitalWrite(pin, Value) – like the Arduino function of the same name, applies a HIGH or LOW value to a digital output (pin). The ports RELE_CHANNEL1–RELE_CHANNEL4, WATER_PUMP_PIN, LUA_PIN, ALARM_BTN_PIN and BTN_PIN are available, the others are ignored. That is, if you need to set a high value on the RELE_CHANNEL4 output, call digitalWrite(13, 1). For WATER_PUMP_PIN with a PWM pump, the value 1 switches the pump on at full speed, 0 stops it. If the emergency heating protection has triggered, writing to the heater channels (RELE_CHANNEL1, RELE_CHANNEL4) is not performed.

digitalRead(pin) – like the Arduino function of the same name, reads the value from the given input - HIGH or LOW. That is, if you need to read the value set on the RELE_CHANNEL4 input or output, call digitalRead(13)

analogRead() – like the Arduino function of the same name, reads the value from the analog input LUA_PIN (no parameters). The voltage applied to the analog input will be converted to a value from 0 to 4095. ATTENTION! Applying a voltage higher than 3.3 volts to the input can damage the port or the whole ESP32. In Cheesemaking mode LUA_PIN is occupied by the pH sensor: pinMode and digitalWrite for it end with an error.

exp_pinMode(pin, mode) – like pinMode(pin, mode), but for controlling the PCF8575 port expander; pin is the expander port number (0–15), mode is INPUT, OUTPUT or INPUT_PULLUP. All expander ports can be selected as either read or output

exp_digitalWrite(pin, Value) – like digitalWrite(pin, Value), but for controlling the PCF8575 port expander; pin is the expander port number (0–15), Value is 1 or 0. All expander ports are available for writing

exp_digitalRead(pin) – like digitalRead(pin), but for controlling the PCF8575 port expander; pin is the expander port number (0–15). All expander ports are available for reading

exp_analogWrite(Value) – like analogWrite(pin, Value), but for controlling the PCF8591 port expander, Value is a value from 0 to 255. The expander has one port available for output.

exp_analogRead(pin) – like analogRead(), but for controlling the PCF8591 port expander; pin is the expander input number (0–3), returns a value from 0 to 255. The exp_* functions exist only in firmware built with USE_EXPANDER (PCF8575) and USE_ANALOG_EXPANDER (PCF8591).

delay(ms) – like the Arduino function of the same name, pauses program execution for the number of milliseconds given in the parameter (1000 milliseconds in 1 second). No more than 1000.

millis() – like the Arduino function of the same name, returns the number of milliseconds since Samovar started

setTimer(Num, Sec) – set timer number Num for Sec seconds (up to 65535). There are 10 timers, 1 to 10. That is, scripts can work with no more than 10 timers at the same time

getTimer(Num) – get the time remaining on timer number Num in seconds. If the timer is not set or the time has run out, the function returns 0

sendMsg(Msg, Level) – If Level = -1, the message Msg is printed to the COM port and to the browser console, convenient for debugging. If Level is 0, 1 or 2, the message is sent as a system message (0 – alarm, 1 – warning, 2 – notification): to the display, the web interface, the apps and the website.

setPower(Power) – switches Samovar on/off. setPower(0) switches it off, setPower(1) switches it on (the same as the heating button of the current mode)

setCurrentPower(Value) – set the value Value on the regulator. If a regulator with power control (SEM/AVR) is used, then Value is the power in watts to set, otherwise it is the voltage with 0.1 V accuracy. The function exists only in a build with a power regulator.

setBodyTemp() – set the hearts temperature during rectification. It works only in rectification mode. In other modes it prints a message to the console that the hearts temperature cannot be set

setMixer(Val) – switch the stirrer on/off. setMixer(0) – off, setMixer(1) – on

openValve(Val) – open/close the water valve. openValve(0) – close, openValve(1) – open

setPumpPwm(Val) – set the water pump speed, Val from 0 to 1023. Exists only in a build with a PWM pump (USE_WATER_PUMP).

setMixer, openValve, setCurrentPower and setPumpPwm return the command result code (the ACTUATOR_COMMAND_* constants).

setAlarm() – set the alarm mode. Samovar will switch off the power, close the water valve and switch off the water pump

setNextProgram() – go to the next program. Like pressing the "Next program" («Следующая программа») button in the interface. Works only when heating is on.

setPauseWithdrawal(Val) – pause/resume the take-off. setPauseWithdrawal(0) – resume, setPauseWithdrawal(1) – pause

setCapacity(Num) – turn the servo to container number Num (0 – drain, 1 and up – containers).

setServoAngle(Angle) – turn the servo to Angle degrees (0–180). Returns 1 if the firmware has a servo output (SERVO_PIN), otherwise 0. Used for homemade additive dispensers (see [Additive dispensers](#)); the servo is shared with setCapacity, so after turning it in rectification the container must be set again.

getState() – get the Samovar status (a number), convenient for determining the state Samovar is in: 0 – idle; rectification: 10 – take-off in progress, 15 – automatic pause between program lines, 20 – program completed, 30 – take-off pump calibration, 50 – boost, 51 – stabilization, 52 – stabilization completed; 40 – manual pause (in any mode); 1000 – distillation in progress, 2000 – beer, 3000 – wash column, 4000 – NBK, 5000 – cheesemaking.

setNumVariable("Variable", Val) – set an internal numeric Samovar variable. Not all variables can have values set (list below). setNumVariable("loop_lua_fl", 1) will enable the one-second script loop. Attention! If you do not understand how Samovar works, it is better not to use this function, since you could potentially disrupt Samovar's operation

setStrVariable("Variable", Val) – set an internal string Samovar variable. Not all variables can have values set. setStrVariable("SamovarStatus", "Test Samovar status")

getNumVariable("Variable") – get an internal numeric Samovar variable. TankTemp = getNumVariable("TankTemp") will assign the boiler temperature value to the script variable TankTemp

getStrVariable("Variable") – get an internal string Samovar variable.

program_type = getStrVariable("program_type") – will assign the type of the program currently running to the script variable program_type.

Script variables exist only while the script is running. On the next run their values are not preserved. Sometimes you need to remember a variable's value in one script run and read it in another. There are two functions for this – setObject("Object", Val) and getObject("Object")/getObject("Object", "NUMERIC"). The set variable values are kept until reboot. There can be no more than 32 different objects; if exceeded, setObject ends with an error.

setLuaStatus(Status) - to show the script status in the interface ("Lua status" («Статус Lua») on the "Additional" («Дополнительно») tab). For example, tracking script execution:

```
i = getObject("cnt", "NUMERIC")
```

```
i = i + 1
```

```
setLuaStatus("Counter cnt = "..i)
```

```
setObject("cnt", i)
```

```
setObject("Object", Val) – will remember the object Object with the value Val in the Samovar memory.
```

```
getObject("Object") – gets the previously saved value of the object Object. If no such object was saved, an empty string is returned. If you try to use an empty string as a number, it will take the value nil – undefined. To make working with numeric values easier and avoid checking for nil, you can call this function with the additional parameter "NUMERIC"; in this case, if the object has not been initialized yet, 0 is returned.
```

An example of working with setObject/getObject:

```

n = 156
setObject("MyObject", n)
print (n)
n = n + 5
print (n)
n = getObject("MyObject")
print (n)

```

If you run this script, it will print in the Arduino port monitor

```

156
161
156

```

http_request(Url) – perform a GET request to the address Url and return the response body (or the string "error"). For example, you can send readings to your own web service. You can also send a POST request by passing four parameters: the address, the method, the Content-Type header and the body:

```
http_request("http://test.com:80/post?foo1=bar1", "POST", "Content-Type: text/text; charset=utf-8", "body")
```

An example of a script that sends the boiler temperature to an external web service (replace the address and parameters with your own):

```

local char_to_hex = function(c)
    return string.format("%02X", string.byte(c))
end

```

```

local function urlencode(url)
    if url == nil then
        return
    end
    url = url:gsub("\n", "\r\n")
    url = url:gsub("[^%w ]", char_to_hex)
    url = url:gsub(" ", "+")
    return url
end

```

```

function SendToServer(text)
    local key = "my_secret_key" -- your service access key
    http_request("http://example.com/notify?key=" .. key .. "&text=" .. urlencode(text))
end

```

```

local text = "Current boiler temperature = " .. getNumVariable("TankTemp")
SendToServer(text)

```

The functions of the I2CStepper board (stirrer, dosing pump, relay) – set_stepper_by_time, set_stepper_target, get_stepper_status, i2cpump_start, i2cpump_stop, i2cpump_get_speed, i2cpump_get_target_ml, i2cpump_get_remaining_ml, i2cpump_get_running, set_mixer_pump_target, get_mixer_pump_status, set_i2c_rele_state, get_i2c_rele_state, check_I2C_device – are described on the page [I2CStepper](#).

Variables

The list of internal variables available in a script by default (their values are filled in before each script run, changing them by assignment is useless):

bme_pressure – the current atmospheric pressure value
 capacity_num – the number of the container being taken off into
 SamovarStatusInt – the Samovar operation status (the same as getState())
 ProgramNum – the current program number
 ProgramLen – the number of lines in the program
 ActualVolumePerHour – the current take-off rate in liters
 WthdrwlProgress - the progress of the current take-off
 PowerOn – the heating on/off flag. 0 – off, 1 - on
 PauseOn – the pause flag. 0 – no, 1 – yes
 StepperMoving – the peristaltic pump running flag. 0 – not running, 1 – running
 program_Pause – the flag that a pause program is running. 0 – no, 1 – yes
 program_Wait – the flag that the program is paused. 0 – no, 1 – yes
 program_Wait_Type – the reason for the pause – exceeded column section or vapor temperature. The values are the Russian strings "(nap)" (vapor) or "(qapra)" (column section)
 WFlowMilliLitres, WFlowTotalMilliLitres, WFlowRate – the volume for the last interval, the total volume (ml) and the cooling water flow rate from the flow sensor (only in a build with a flow sensor)
 WthdrwTimeAll – the remaining take-off time
 WthdrwTime – the take-off time of the current program line
 WthdrwTimeAllS – the remaining take-off time as a string
 WthdrwTimeS – the take-off time of the current program line as a string
 pump_started – the water pump running status. 0 – no, 1 – yes
 heater_state – the heating status during mashing. 0 – not heating, 1 – heating
 mixer_status – the stirrer running status. 0 – not running, 1 – running
 alarm_event – the flag that the alarm button has triggered. 0 – no, 1 – yes
 acceleration_heater – the flag that the boost heating element is on. 0 – off, 1 – on
 valve_status – the water supply valve status. 0 – closed, 1 – open
 program_type – the type of the current program (from the description of rectification and beer programs)
 program_volume – the take-off volume in ml
 program_speed – the take-off rate in l/h
 program_temp – the temperature at which this part of the distillate is taken off. 0 - determined automatically
 program_power – the voltage/power at which this part of the distillate is taken off
 program_time – the time needed for this program line to run
 program_capacity_num – the container number for take-off

SamSetup_Mode – the Samovar operating mode: 0 – rectification, 1 – distillation, 2 – beer, 3 – wash column, 4 – NBK, 5 – sous-vide, 6 – Lua mode, 7 – cheesemaking
test_num_val – a numeric variable for debugging
test_str_val – a string variable for debugging
SteamTemp – the vapor sensor temperature
PipeTemp – the column section sensor temperature
WaterTemp – the water sensor temperature
TankTemp – the boiler sensor temperature
ACPTemp – the TCA sensor temperature
current_power_mode – the current regulator operating mode
target_power_volt – the set regulator operating voltage
wp_count

Variables available through getNumVariable (read-only): WfpulseCount, pump_started, valve_status, SamSetup_Mode, Samovar_Mode, Samovar_CR_Mode (the current and switchable mode, numbers as in SamSetup_Mode), SteamTemp, PipeTemp, WaterTemp, TankTemp, ACPTemp, WftotalMilliLitres, WfflowRate, program_volume, program_speed, program_temp, program_power, program_time, program_capacity_num, capacity_num, target_power_volt, PowerOn, PauseOn, program_Wait, alcohol (the alcohol content in the boiler by temperature), alcohol_s (the alcohol content by vapor), water_pump_speed (the water pump speed 0–1023), pressure_value (the boiler pressure), YY, MM, DD, HH, MI, SS (year, month, day, hours, minutes, seconds).
Variables available through getNumVariable and setNumVariable: acceleration_temp, boil_temp (the remembered boiling temperature), wp_count, test_num_val, loop_lua_fl, SetScriptOff, show_lua_script. Only through setNumVariable: pmpKp, pmpKi, pmpKd – the PID coefficients of the water pump.
String variables (getStrVariable/setStrVariable): SamovarStatus – the status text on the main page (read and write), test_str_val (read and write), program_type (read only), Msg – when read, returns the next unread system message or an empty string, when written, adds a message.

Variables for controlling scripts:

loop_lua_fl – values 0 or 1. If 0 – do not run the every-second script loop, 1 – run it.

SetScriptOff – values 0 or 1. If 1 – stop the every-second loop (resets loop_lua_fl). Samovar sets it to 1 itself at the end of a program.

show_lua_script – values 0 or 1. If 0 – do not show the script being run in the console and the port monitor, 1 – show it. Can be used for debugging. It will also show the values of all variables set for this script.

For example, if you call these two functions in the init.lua script, two scripts (script.lua and the one determined by the current Samovar mode) will run every second, and each script will be printed to the port monitor and the browser console

```
setNumVariable("loop_lua_fl",1)
```

```
setNumVariable("show_lua_script",1)
```

Script Examples

An example of a script that you can write directly in the editor in the web interface. It runs once a second and, when the boiler temperature reaches 70 degrees, starts switching on and off the device connected to the first port of the PCF8575 expander, and also writes about it to the UART and to the browser console

```
if TankTemp > 70 then
  if exp_digitalRead(1) == 0 then
    exp_digitalWrite(1,1)
    print("Включено")
  else
    exp_digitalWrite(1,0)
    print("Выключено")
  end
end
end
```

An example of a script that gets a value from an analog level sensor on the LUA_PIN input (34 for the DEVKIT board) and controls the water pump (for example, to pump wash into the boiler). If the signal is in the range from 1000 to 2000, it turns the pump on; at a zero signal it turns it off. The script is written right in the browser and is available for execution immediately after saving. Pin 4 is WATER_PUMP_PIN of the DEVKIT board; for other boards, substitute your own number.

```
start_pump = getObject("start_pump", "NUMERIC") + 0 --get the numeric value from a previously saved object and convert it to a number
sensor = analogRead() --read the analog value of LUA_PIN (reserved for Lua)
--sensor = 1500
if sensor >= 1000 and sensor <= 2000 and start_pump == 0 then
  setObject("start_pump", 1) --save the value in an object in Samovar memory so that it can be used in the next script run cycle
  digitalWrite(4,1) --set a high level on pin 4
  print("Start pump") --[write to the COM port. You can also use the sendMsg("Msg", Level) command. If Level = -1, the message is output to the COM port and to the browser console, which is handy for debugging. If Level is 0, 1 or 2, the message is sent as a system message]=]
  --sendMsg("Start pump",-1)
elseif sensor == 0 then
  setObject("start_pump", 0) --save the value in an object in Samovar memory so that it can be used in the next script run cycle
  if (start_pump == 1) then
    digitalWrite(4,0) --set a low level on pin 4
    print("Finish pump")
  end
end
end
```

Ready-made scripts that are uploaded to Samovar together with the web interface:

rectificat.lua – filling the boiler with a pump by a level sensor or by a water flow sensor (the volume and sensor thresholds are set at the beginning of the file).
dist.lua – distillation split into three containers: it remembers the boiler temperature at the start of boiling and, when it rises by a given setpoint (or when the ABV drops), switches the servo to the next container, lowering the power if desired.
cheese.lua – a script template for the L line of a cheesemaking program.
btn_<mode>_button1.lua "Start" («Начать») and btn_<mode>_button2.lua "Stop" («Остановить») – turning the one-second cycle on and off for each mode.
More examples are in the Lua_script folder of the [Samovar repository on GitHub](#): fermentation temperature control (beer_mash.lua), water pump speed control with buttons, distillation with concentration (dist_degirate.lua), sending readings to an external web service over HTTP (http_request_telegram.lua), holding pre-flooding by a pressure sensor, and scripts for the NBK.

Questions and answers. FAQ

I do not have a thermowell for the steam sensor. What should I do? Can I go without this sensor?

Samovar needs the readings of the column section and steam sensors to work. You can do this:

1st method. Install one sensor 15-20 cm from the top edge of the packed column section and use it as the steam sensor. Install the second, lower sensor at least 30 cm below the upper one, or, for a short column section, at 2/3 of the column height.

2nd method. Install only the lower sensor. On the Temperature («Температура») tab, select it as the column section sensor. Select the same sensor as the steam sensor. For visual comfort, correct its readings to 1-2 degrees above the column section sensor, and set the overshoot beyond the setpoint to more than that of the column section sensor.

What is 2/3 of the column height? Where is that?

This refers only to the height of the part of the column that contains packing. Empty space at the top or bottom is not counted. Take a tape measure and measure the height the packing occupies in the column. Divide it by 3. Two parts from the top is the 2/3. Or one part from the bottom. This is a recommendation. But you can install the column section temperature sensor according to your preferences or the configuration of your equipment.

Where should the reflux level sensor be installed?

As a standard, the level sensor is installed directly on the sight glass, 5-7 mm from the lower edge of the glass. The sensor must be adjusted. After the column reaches stabilization mode and the sight glass has warmed up, set the trigger threshold.

What is a setpoint?

A trigger setpoint is the value of the controlled process variable set in the regulator, upon reaching which the regulator takes action. In other words: the permissible "overshoot of the actual value" beyond the set value at which the system stays at rest.

The column is assembled and the sensors are under the insulation. How do I find out which sensor has which address?

It is not difficult. First, from the drop-down menu, choose different sensor addresses for each installation location. Any ones, but all different.

Then we see which of the sensors is accessible. Say, Water: we warm it with a hand or lightly with a hair dryer. We look at the main page. The temperature of one of the sensors starts to rise. Say it is Steam. We go to the Temperature («Температура») tab and swap the addresses of the Water and Steam sensors.

The next sensor is the TCA. We do the same. We swap the TCA with the one whose temperature started to rise. Two sensors are set up.

The rest cannot be reached under the insulation. No problem. We pour the low wines into the boiler and get ready for the run. Ready? We turn on the boost. The Boiler sensor will start heating first, but your Steam sensor is rising again. We swap the addresses. And so on. The next sensor will be the Column section, and finally Steam. All the sensors are in their places, and we continue the run.

Warning. Make sure that addresses are not duplicated when you choose them. Samovar does not check this. The sensor address at each installation location must be unique.

Warning. The settings of setpoints, overshoots, temperatures, etc. do not move together with the sensor address; they belong to the specific installation location.

How do I calibrate the sensors and remove errors?

This is done by entering values in the "Correction" («Корректировка») field. We should say right away that absolutely exact actual temperature values are not needed. If someone's steam temperature is 77 and yours is 75, it does not matter much. But let us return to calibration.

The most accurate way is to calibrate all the sensors at the same time, at a temperature close to the one at which they will work.

You can place all the sensors in boiling water and set the temperature to 99 for all of them.

But not all sensors can be put in water. You can put them in an empty vessel and set it in a water bath. The temperature at the bottom will be close to 99.

It is easier to calibrate the sensors relative to each other.

For example, in winter, put them on a radiator, cover them, and when the temperatures settle, set one value for all of them. You can come up with quite a lot of options.

But what if the sensors are already under thermal insulation? You can lay all the parts of the column on a table, place the free sensors inside the column section, cover them with something dense and, after some time, calibrate them relative to each other.

The description of the beer mashing program mentions "different brewhouse systems". What is that?

A brewhouse system for beer production is special equipment for brewing wort. Brewhouse systems differ depending on the technological tasks. [This article](#) describes in detail the types of home brewhouse systems and how they work.

Which devices and sensors can Samovar detect as present or absent?

Sensors and actuators whose presence Samovar does not check, because there is no feedback:

- Contactor / voltage regulator switch-on (relay No. 1)
- Additional heater relay / voltage regulator boost (relay No. 4)
- Buzzer
- Water opening valve (relay No. 3)
- Reflux level sensor
- Peristaltic pump stepper motor
- Take-off table servo drive
- Cooling pump
- Atmospheric pressure sensor
- Boiler pressure sensor.
- Stirrer (relay No. 2)

Sensors and actuators whose presence Samovar does check, if:

- All temperature sensors, if the sensor was present at the start of the process. Messages about its absence will arrive regularly.
- Water flow sensor, if its use is enabled.
- Heating regulator, if the regulator usage check is enabled.
- I2CStepper board (stirrer, pump): polled when Samovar starts and then in the background; if communication is lost, a message is shown, and the "I2CStepper" tab and the fields related to the board are hidden.

How to use the bypass control valve

You will need two taps, two valves and two tees.

One valve is NC (normally closed) and the other can be either NC or NO (normally open); NC is preferable. The layout is as follows:

Supply line - first valve - this is the main valve that will open the water supply for cooling.

Next - a tee - both taps are attached to the tee. One of them (usually a needle valve) will be the main one and the other the additional one. After the additional tap comes another valve, then both lines join again through a tee, into one, and the water is supplied to Samovar.

That is, a bypass line is set up around the main tap.

On the main tap we set the water flow so that the temperature at the dephlegmator outlet is 50°. Say it is 40 liters per hour. And with the additional tap we set 10 l/h. In the Samovar_ini.h file we comment out the line `#define USE_WATER_PUMP` and uncomment `#define USE_WATER_VALVE` (the pump and the valve cannot work at the same time).

How it works. When the boiler temperature reaches the configured threshold (77° by default), Samovar opens the main valve, and 40 l/h goes to cooling. If the water pressure drops, the temperature starts to rise, the additional valve opens and an extra 10 l/h goes to cooling. When the temperature returns to the set limit, the additional valve closes.

Links. Practical tips.

Links to various resources with useful information about components and how to set them up.

Help for Samovar versions up to 6.26, in PDF and DOC formats.

[Download the help in PDF format](#)

[Download the help in .doc format](#)

Attention.

The PDF documentation is updated only when versions with critical changes in the interface or in Samovar operation are released, so it may differ from the documentation on the website.

[Download a Samovar](#) release.

[How to install the Arduino IDE](#)

[More about working with the Arduino IDE](#)

[Installing and configuring the ESP32 board in the Arduino IDE](#)

[Meter](#) and circuits by dranek.

History of Versions

7.00 — September 4, 2026

A major update. Main changes:

New "Cheesemaking" («Сыроварение») mode: heating, holds, cooling, acidity control with a pH sensor (PH-4502C on the LUA_PIN input, calibrated on the "pH calibration" («Калибровка pH») page), a stirrer, a brine drain valve on relay No. 4, and its own program line types. The settings are on the "Cheese" («Сыр») tab.

Impurity breakthrough detector in rectification mode: it analyzes the temperature trend with a pressure correction, picks the threshold itself from the column noise, can lower the take-off rate when the temperature rises, pauses the take-off on a critical rise and resumes by itself; on heads it only observes, and the reason for the wait is shown in the interface.

A major update of the Beer, BK and NBK modes: the All-in-one, HERMS and RIMS brewing layouts, improved BeerXML import, reworked NBK algorithms (warm-up, tuning, optimization, operation); an L line - a stage controlled by Lua - appeared in the Beer and Cheese programs.

The L line (Lua stage) in all programs except NBK: Rectification, Distillation, BK, Beer, Cheesemaking. The script file, the timeout in seconds and the parameters (the arg table in the script) are set in the "Lua stage" («Lua-этап») window of the program editor; when a new version of the file is uploaded through the editor, the running scenario is re-read. In BK the "Next program" («Следующая программа») button now works.

New web interface: a dark "copper" theme, a process diagram with live readings (in Beer mode - by brewing layout, in rectification - with the dephlegmator and the second pump, if it is enabled), the current program line, charts, mode pages, the "External pump" («Внешний насос») page and the "I2CStepper" page; the pages are served compressed and get their initial data in a single request.

Connection to the samovar-tool.ru server over TLS instead of MQTT: readings, commands, the process log and messages go through this channel; the logs page on the site, the control page on the site and the Android and iOS mobile apps with push notifications about alarms are built on it. Sending to your own MQTT broker remains as an optional build setting.

The I2CStepper board, protocol version 2: two boards (a stirrer and a pump), priming, filling a set volume, "steps per milliliter" calibration, four relays, an external sensor. Control from the "I2CStepper" page, from the program, Lua, the apps and the site. The I2C pump can be used as the heads take-off pump above the pasteurization section.

Lua: a limit on the time of a single run (20 s), the cycle stops after repeated errors, new functions setPumpPwm and setCapacity, I2CStepper board functions, and Lua mode as a separate operating mode.

State snapshots: after an emergency reboot the firmware restores the program and continues the same log session if the process is restarted within 30 minutes.

A configurator for Windows, macOS and Linux: building, flashing over USB or Wi-Fi, a device file editor, a configuration report. ESP32-S3 support.

The calculations of rectification and distillation programs, saving of programs and process state, and equipment error handling have been reworked.

6.27 — May 11, 2026

Fixed the time counter, network timeouts, starting the access point with the universal button, the completion of Distillation, UART regulator control in Beer mode, power display, PID autotune and cooling pump readings.

6.26 — January 1, 2026

Libraries updated, stability and memory usage improved. Mode switching without a reboot added. Before updating to this version it was recommended to save the settings via Settings («Настройки») → Other («Прочие») → Save settings («Сохранить настройки»).

6.25

- User suggestions added, libraries updated, additional minor improvements.
- The partition file fixed. The function for sending messages to Telegram also corrected.

6.24.

- Previously found bugs fixed.
- The NBK mode is also ready for use.
- Network handling improved.

6.23.

- The web interface corrected.
- Bugs fixed.
- The pump operation in the Beer program corrected.

6.22.

- The key change - the libraries have been updated.
- In Beer mode, the stirrer run time is now set in seconds, not in minutes.
- Also, in distillation mode a forecast of the completion time is shown.

6.20

- Bugs fixed.
- The NBK mode has been implemented, based on the research by [dranek](#)
- The operating modes of the heater and of the wort transfer pump when brewing beer have been changed.
- Notifications to Telegram added, since they stopped working in Blynk.

6.19.

- The NBK mode added. The main functionality has been implemented for the safety group and manual control of the cooling pump. Automation is for now supposed to be done through Lua.
- Numeric variables for working with the current time added to Lua.

YY, MM, DD, HH, MI, SS - year, month, day, hours, minutes, seconds respectively.

Call example:

```
setLuaStatus(getNumVariable("HH"))
```

6.18.

- Variables for Lua added.
- The interface for versions 6.16 and 6.17 in Beer mode has been updated - the PID autotune mode is back.
- It is now possible to turn off the stirrer in the mash program by selecting it in the interface.
- Minor code optimization

The interface will update automatically the next time Samovar boots.

6.17.

- The file editor version has been updated ([in the settings](#)).
- The library for working with the screen menu has been updated.
- The Samovar and [I2CStepper](#) sketches have been slightly optimized.
- The archive now includes the latest version of the Meter and the diagrams from [dranek](#), as well as a 3D model of the enclosure for the SEM(Stab)_AVR regulator
- Beer mode, using Lua scripts (at this stage), makes it possible to implement any [brewing layout](#) with full automation and the use of recipes in the beerxml or BrewMate format. Recipes are supported to the minimum extent needed to build a mash program. That is, they may not show all the ingredients needed to prepare the mash
- Among the functional updates - a button to refresh the temperature sensors has been added in the temperature settings. You can connect sensors one at a time and save them in the settings. Turn on one sensor, select what it belongs to, press save, go back to the settings, connect another sensor, select what it belongs to, press save, and so on.
- Nothing has changed in the operating logic.

6.16.

- It is now possible to enable a smooth ramp-up of the take-off pump. However, it will also stop smoothly, which may not be very good for process stability when moving from one program line to another.

To enable acceleration/deceleration of the stepper motor, replace the following in the Samovar_ini.h file

```
##define USE_STEPPER_ACCELERATION //использовать плавный старт и торможение шагового двигателя
```

with

```
#define USE_STEPPER_ACCELERATION //использовать плавный старт и торможение шагового двигателя
```

- Instructions added on how to connect the regulator to Samovar without wires. The instructions are also available on the [site](#).
 - The library for working with the stepper motor has been updated, and interaction with I2CStepper has been improved. Version 0.3 of I2CStepper has been released, bugs fixed. If you need accuracy in speed, lower the microstep multiplier on the driver (no more than 1/2); for greater accuracy, set it to 1. By default the microsteps in the sketch are set to 1/2. This can be adjusted in the I2CStepper.h file, line 22. The divider is specified there.
- The interface has been updated. In "Beer" mode, the stirrer parameters in a program line are edited in a separate window.

Warning! The libraries for Samovar and I2CStepper must be taken from the ones supplied with Samovar.

6.15

- New default values have been added to the initial settings file tools/Samovar_default_settings.txt. On the first start of Samovar after building, you must load the default values. There are already many options, it is difficult to set them all correctly at once, and the default values will give you a working Samovar. Then you can tune it to your own needs.
- A bug with the stepper motor for adding hops in Beer mode fixed.
- An option to control the use of the reflux sensor from the WEB interface has been added to the Samovar_ini.h file (enabled by default)

```
#define IGNORE_HEAD_LEVEL_SENSOR_SETTING //If the reflux sensor is used, ignore the command to disable the sensor in the WEB interface (details here).
```

6.14

- If you press the button before powering on Samovar (release it 2 seconds after power-on), Samovar will start in access point mode. Network: Samovar, password: SamApp123
- Support for the Blynk terminal added - you can run a small piece of Lua code (a few lines; running large scripts this way is not recommended). How to add a terminal in Blynk - [here](#)

- You can also run a few lines of Lua in the web interface - available in all modes on the Advanced («Дополнительно») tab when Lua is set up in Samovar
- Lua can now control the PID coefficients of the water pump - pmpKp, pmpKi, pmpKd respectively. To set a coefficient, you can run this Lua code
`setNumVariable("pmpKp", 6)`
 You can get the current value like this:
`setLuaStatus("pmpKp"..getNumVariable("pmpKp"))`
- Stepper motor control for automatic hop feeding added in Beer mode. Needs testing.
- The problem with updating the web interface over the internet should be gone.
- Minor bugs also fixed.

6.12

- Support for three pressure sensors added:
 - XGZP6897D (connected via I2C),
 - MPX5010D (analog, connected to pin 34 of the ESP32)
 - Via 1Wire. For a sensor connected via 1Wire, the address is fixed - {0x22, 0x22, 0x22, 0x22, 0x22, 0x22, 0x22, 0x22}.
 The choice of which sensor to use (or none) is made in Samovar_ini.h:

```
//Definition of the pressure sensor type used. Only one type can be used. If three or two
sensors are uncommented, only MPX5010D will be used
//#define USE_PRESSURE_XGZ 32 //использовать датчик давления XGZP6897D (или аналогичный). Какое
использовать значение - смотреть параметр K в таблице:
https://github.com/fanfanlatulipe26/XGZP6897D
//#define USE_PRESSURE_1WIRE {0x22, 0x22, 0x22, 0x22, 0x22, 0x22, 0x22, 0x22} //использовать
датчик давления по 1Wire
//#define USE_PRESSURE_MPX //использовать датчик давления MPX5010D
```

 There is no logic or display in the interface yet, but it can be used in a Lua script, outputting the value to the Lua status and implementing the required Samovar behavior there when certain values are reached. Also, in the Samovar settings it will be possible to set a pressure limit at which Samovar will switch to emergency mode. If the pressure value in the settings is set to 0, Samovar will ignore the pressure sensor readings.
 An example of outputting pressure to the Lua status:

```
pressure_value = getNumVariable("pressure_value") + 0
setLuaStatus("давление = "..pressure_value)
```
- For viewing the pressure sensor information, output to Blynk has been added. You can add it to the Blynk interface yourself, using virtual pin 22.
- Support for a 1Wire pressure sensor via a 1Wire expander and the ability to view it in Blynk added.
- The readings of all 5 temperature sensors are now shown on the Samovar display.

6.11

- Beer mode.
 - Besides the water supply valve, the cooling pump will be turned on.
 - Cooling pump speed control added.
 - A check for the correct formation of the "Stirrer" («Мешалка») field added. If it is formed incorrectly, it is changed to 0^0.00^0^0
- Rectification mode. An option added to Settings («Настройки») > Basic («Основные»). Whether or not to use the reflux level sensor.
- Distillation mode. An option added to Settings («Настройки») > Basic («Основные»). The time in minutes after which distillation is stopped, provided that the boiler sensor value does not change. If 0, the time is not taken into account.
- The font of the current heating element power has been made larger.

6.10

- In Beer mode, a choice of which temperature sensor to use for controlling heating has been added to the program line (needed to implement various brewing layouts). A choice of 5 sensors is available.
- The pump run time in self-test mode has been increased
- In Beer and Distillation modes, the program and the current program execution status are now saved to text files, in the same way as it is done in Rectification mode.
- Lua can now regulate the water pump output. The function setPumpPwm(p) added, p=0 - the pump is off, p=1023 - maximum output.
- It is now possible to turn off the water flow sensor monitoring in the settings.
- An option added to the settings - do not reduce power during boiling in Beer mode.

Warning! You can use SDK 2.0.13. If you have stability problems, roll back to 2.0.9

6.9

The possibilities of controlling the stepper motor through LUA scripts have been extended.

If the I2CStepper expander on Arduino Nano is not connected, a call to the Lua function `set_stepper_target` controls the built-in stepper motor; otherwise control goes through the expander.

When controlling the built-in stepper motor, only one parameter is taken into account - `spd`, the speed in steps per second; all three parameters must still be passed in the function call.

--set_stepper_target(*spd, direction, target*) //*spd* - speed in steps per second, *direction* - forward or reverse direction, *target* - number of steps

`set_stepper_target(950, 0, 0)`

When `spd = 0` the stepper motor is stopped.

Warning! When updating to this release you must use SDK version 2.09

6.8

- The way a Lua script is stopped by a button in the interface has changed.
- The `SetScriptOff` variable added. In the `btn_button2.lua` button script it is set to one, after which one more Lua script run cycle passes.
- This is done so that the script stop event can be intercepted and the necessary actions performed to finish the script correctly.

6.7

- Bug fixes and minor improvements.
- Of note - version 6.6 did not detect the start of boiling in distillation mode. Fixed.

6.6

- In Distillation mode a button to move to the next program has been added. Bugs related to detecting the start of boiling and the operation of the program in Distillation mode fixed.

6.5

- The sketch now makes it possible to change the rotation direction of the take-off peristaltic pump stepper motor. To do this, uncomment the following line in `Samovar_ini.h`:
`//#define STEPPER_REVERSE` //изменить направление вращения шагового двигателя
- Premature detection of the start of boiling also fixed.

6.4

- The calculation of the ABV in the boiler has been fixed
- The calculation of the ABV in the take-off has also been fixed.
- Uploading of ABV to the cloud added; on the site, the ABV will be shown in the logs when viewing distillation and rectification charts.
- When installing the update, update the SDK to the latest version.

6.3

- The changes affected distillation mode - a distillation program has been added. Its purpose is to change the take-off container by temperature or ABV, and it does nothing else. The algorithm for starting and finishing distillation has not changed, so whether there is a program or not makes no difference to you if you do not plan to use distribution among containers. I also added determination of the alcohol content in the vapor (which for distillation is equivalent to that in the take-off). I checked the calculation purely mathematically; in real conditions there may be a discrepancy with hydrometer readings. Both values are shown in the interface.
- In the program you can use either the ABV in the boiler or the ABV in the take-off; this is determined by the type of the take-off line.
- If distribution among containers is not needed, the servo does not need to be connected. Then it will be plain distillation, regardless of what is written in the distillation program.

6.2

Bugs fixed:

- On SDK 2.0.5, Samovar could reboot in rectification mode.
- In rectification mode the take-off program was written to the `prg.csv` file incorrectly. Code optimization.

6.1

- Improvement - a time delay when determining the boiling temperature and the initial alcohol content in the boiler.
- Data on the strength of the boiler contents will appear in messages, in the interface and will arrive in Blynk.

6.0

Main improvements:

- It used to be inconvenient that Samovar's memory was constantly overwritten and you had to restore your work every time (for those who had any) - rectification programs, Lua scripts. Now, when there is an internet connection, the interface updates automatically when the Samovar sketch version changes, without overwriting other files in Samovar's memory, which makes it possible to keep the settings for each mode in Samovar's memory and load them when the mode is changed. If you are setting up a new ESP32, all files from the data folder will be downloaded. The option of installing manually (not over the internet) remains, but the file system used has changed, so for manual upload you must use not ~~ESP32 Sketch Data Upload~~, but ESP32 LittleFS Data Upload. The switch to a different file system was made to optimize Samovar's speed.
- The interface will be downloaded automatically for the specific sketch version (starting with 6.0), i.e. if you need to roll back to an old sketch version, the interface will also roll back to the corresponding version.
(That is, when updating or rolling back, the web interface updates itself. You need to update the sketch and the libraries manually)
- Interaction with Lua scripts has been improved. Example scripts and Lua documentation are placed in a separate folder (Lua_script).
- Using a scripting language makes it possible to use all of Samovar's functionality, extending it for your specific tasks. As an example - distillation by Gabriel or NBK control.
- A change in the take-off program. A "Steam temperature" («Температура пара») value is introduced (disabled in later versions).
- The TCA temperature value at which the pump increases cooling has been moved to the "Temperature" («Температура») tab.
- Bugs found have been fixed.

5.27

- The changes concern the work with the stirrer and the pump in Beer mode. One more parameter has been added to the Beer mode program - the stirrer operation template.
- In Beer mode, when the stirrer is turned on, Samovar turns on relay No. 2 and the SSD relay, and, if an expander on Arduino Nano (which controls the stepper motor) is connected, sends it a command to turn on the stepper motor and one more relay.
- In the stirrer operation template you can set the stirrer run time, the rotation direction (for the stepper motor only) and the pause. For example: stir for 2 minutes, wait for 1 minute, then stir for 2 minutes in the other direction, wait for 1 minute.

5.27b (beta version)

- A version for testing the Lua script. Details [in the forum thread](#).

5.26

- 1. During rectification, saving of the rectification status and the take-off program has been added. The take-off program is saved to the prg.csv file, the status is saved to the state.csv file. The files can be viewed through settings - the editor (the default login and password are admin) or downloaded at: YOUR_IP/prg.csv and YOUR_IP/state.csv.
- In Samovar_ini.h a parameter has been added for the delay in applying voltage by the regulator after the contactor is triggered.
- Minor improvements, optimizations and fixes of bugs found were also made.

5.25

- Functionality for supporting I2CStepper added.
- Interface changes of the program in rectification mode - when composing a program, the time for each line and the total time for the whole program are now shown.

5.24

Interface change.

5.23

In the Rectification and Beer modes, the type of the program line being executed has been added to the status. Bugs found have also been fixed.

The ability to load beer recipes in the following formats has been implemented:

1. [BeerXML](#) (very popular in the West), the main recipe collection is [here](#), but downloading is paid, [here](#) and in many other places.

2. [Brewmate](#) (popular in the countries of the former USSR).

Examples of recipes can be viewed at the links. You can download them from the site, load them into Samovar, and it will build a mash program based on these recipes. They contain not only recipes but also recipe calculators. Any recipe can be adapted for subsequent loading into Samovar.

Both formats are very flexible, most likely not all of their features are accounted for. If you run into problems working with recipes, send them by email info@samovar-tool.ru

5.22.

Among the key changes - the functionality for automatic correction of the hearts T for the Pre-flooding («Предзахлеб») and Hearts («Тело») programs has been turned off by default. Minor changes related to the work on documenting Samovar's functionality were also made.

5.21

A queue of Samovar messages has been added in the browser. Now they will be hard to miss: they will be shown on the page until they are explicitly cleared by clicking on the message or on the "clear" («очистить») button.

A sound alert for critical events has been added in the browser - when a critical message arrives, the browser will signal it with a sound until the message is cleared. It can be turned on/off in the settings. Modern browsers, out of concern for users, block automatic sound playback by default. There are two options: click on any element on the page, or add the Samovar address to the exceptions in the browser settings.

The following functionality has also been implemented:

If the current program line is "Pre-flooding" («Предзахлеб») and it is followed by another hearts take-off line, the hearts T is corrected automatically. Manual correction is still available.

The run time of the cooling water pump during self-test has been increased.

Bugs fixed:

Logs from Samovar were not always downloaded. Fixed.

In self-test mode the take-off pump stepper motor might not turn on. Fixed.

5.20

Fixes of bugs found.

5.19.

Fixes of bugs found.

5.18.

A bug in Samovar's operation in pre-flooding mode has been fixed. The bug had been present since version 5.11. If you use versions 5.11-5.17 and use pre-flooding mode, you must update.

5.17.

The work with voltage regulators over the UART protocol has been optimized, and the logic of voltage reset when the flooding sensor triggers and the water temperature is exceeded at the same time has been edited.

5.16.

Supports working with the BeerXML format. (The format is very flexible, possibly not all variants of the format were accounted for)

5.15.

Self-test improved. The ability to load the BeerXML format added (for now in view-only mode). When the Samovar operating mode is changed and the settings are saved, it reboots to switch to the set mode. In distillation mode, the take-off goes either up to the temperature set in the settings or until the boiler temperature does not change for 10 minutes. Additionally, small system improvements.

5.14.

Improved control of the regulator from **kvic**, optimized control of the regulator **SEM_AVR**.

A button has been added to the Settings («Настройка») page - Self-test («Самотестирование»): for 5 seconds the water supply pump will turn on and/or the water valve will open, the peristaltic take-off pump will start, and the servo will move through the first 5 positions and return.

5.13.

A new program type - Fermentation («Брожение») - has been added to Beer mode. It is used to hold the set temperature in the boiler within the setpoint band. If the boiler temperature is lower than the set one minus the setpoint, the heater and the boost relay turn on; when the boiler T reaches the set value minus the setpoint, the heating turns off (the PID controller works, as for the other Beer modes). If the boiler T is higher than the set one plus the setpoint, a relay is triggered. A water supply valve, or a pump, or another actuator can be connected to it, depending on your equipment configuration. When the boiler T drops to the set value plus the setpoint, the relay turns off. Additional logic can be implemented in Lua scripts, using a port expander if necessary to control any actuators or sensors.

Bugs were also fixed and libraries updated. The SDK must be updated to version 2.0.5.

In some cases Samovar did not save the WiFi settings; this problem is solved by updating the library and updating the SDK.

There were bugs related to controlling the regulator from kvic (old firmware). This problem is also solved.

New firmware of the kvic regulator does not work with Samovar yet.

If someone uses the "Simple heating element power stabilizer from Chinese modules" («Простой стабилизатор мощности ТЭНа из китайских модулей»), to work correctly with Samovar with the new version it must be updated, using the sketch supplied with Samovar in the Stab-avr folder.

5.12. Minor changes. Code optimization.

5.10.

A change of the take-off programs for fruit, grain and sugar (the files with them are in the tools folder and are also loaded into Samovar's memory when the interface is updated). Also, in the take-off program calculation interface it is now possible to choose the raw material type (fruit, grain, sugar), and the required take-off program is loaded immediately.