**GC97 coulometer data exchange via ESP8266 microcontroller**

**Remote monitoring of battery charge/discharge/capacity with support for settings via a web interface and sending data via the MQTT protocol**

[](https://github.com/alidonet/GC97_ESP/blob/main/manual/GC97_ESP.jpg)

Briefly about GC97

[The GC97 coulometer](https://aliexpress.ru/wholesale?catId=0&SearchText=GC97%20coulometer) is a simple Chinese product designed to be connected to a rechargeable battery (UPS, devices powered by Li-Ion, LeFePo4, etc. batteries). Allows you to visually monitor the main characteristics, such as voltage, current consumption (power), charge level, residual capacity, etc. It can be configured for different batteries, indicating individual parameters and the possibility of precise adjustments. An “emergency” signal output is available with notification, for example, of a critical discharge or exceeding the permissible load power. But the main charm of this coulometer is its support for the TTL protocol. Those. You can take readings from the device in digital form, which is great for various systems that need to be monitored remotely or be part of the now fashionable “smart home”. One of the simple, cheap and fast solutions is to use microcontrollers based on [ESP8266 and its derivatives](https://aliexpress.ru/wholesale?CatId=0&SearchText=ESP8266+wifi&ltype=wholesale&SortType=total_tranpro_desc&groupsort=1) . This allows you to remotely take readings from the coulometer via requests over a local network (web interface) or integrate it with a smart home system using the MQTT protocol.

To create such a bundle you need to have:

* coulometer GC97
* ESP8266 controller (in any form available to you, from debug versions of NodeMCU to miniature D1 mini, etc.)
* provide power to the ESP controller (power can be supplied via the USB port, if available, from a separate power supply, external or miniature step-down DC-DC converter; It is preferable to have a common ground with the battery/coulometer, the main thing is to ensure that the permitted 3.3 or 5V is supplied) .

Connect the GC97 to the battery according to the diagram shown on it and provide power to the ESP. The GC97 itself has Rx/Tx pins that need to be connected to the corresponding ESP pins (if it doesn’t work right away, try swapping the Rx/Tx positions). It’s quite easy for the ESP board (and a miniature DC-DC to hide inside the GC97 case - there’s plenty of space there).

You can find reviews, examples of use and connections on YouTube, for example on the [dima espirans](https://www.youtube.com/watch?v=-X8CvknmFzM) channel .

So, here you will find the ESP8266 firmware to work with the GC97 coulometer, which I created for my personal needs. This is my first ESP firmware and experience of this kind, so there may be some errors and shortcomings. But there it is :)

*The following features are available in the firmware* :

* AP (wifi access point) - for independent operation and initial setup. By default, an open (password-free) network is created with the name GC97-XXXX (where XXXX is the last characters of the ESP mac address), IP address 192.168.4.1;
* Connection to a WiFi network (the ability to scan and select a 2.4GHz network, auto-connection by timeout);
* mDNS local network virtual host (<AP\_name>.local);
* SSDP support (the device is visible on the local network and you can open the control panel simply by clicking);
* Built-in web server with basic coulometer information, the ability to configure via the web interface, reboot, reset and flash; adaptation for display on mobile devices;
* Support for OTA (over-the-air) updates;
* Support for sending data through an MQTT broker with the ability to specify the server address and port, name, password, topic and support for the Retain flag; Auto-connection; Optimization of sending data only for changes;
* The Discovery option is available for Home Assistant: automatic addition of devices and objects;
* Request data via the web in JSON format (by type, general request);
* Selecting the sampling period for the GC97 coulometer, monitoring and averaging a series of data to smooth out “outliers” in the values;
* Light indication of ESP/connection status and operating mode;
* Displays debugging information about the main actions and status of the ESP controller when connected via a serial port (connection speed 19200).

**Web interface**

[](https://github.com/alidonet/GC97_ESP/blob/main/manual/manual_web_main.png)

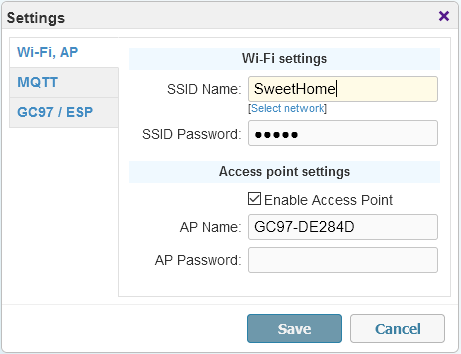
Page available:

* by IP address of the built-in access point (192.168.4.1, after connecting to the GC97-XXXX network);
* by IP address issued by DHCP to the wifi network (if the correct SSID and password are specified), for example 192.168.0.100;

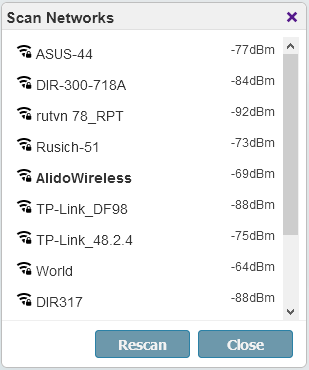
This page displays the current status of the coulometer and all connections (data is updated every 10-15 seconds).

**Settings dialog:**

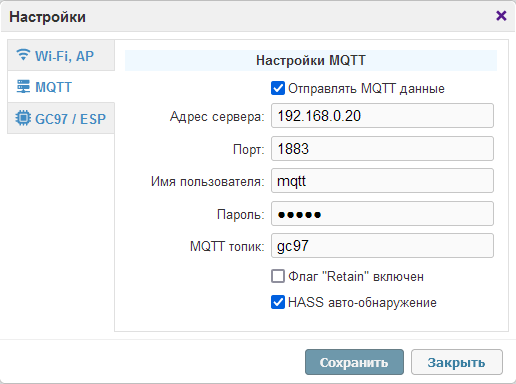
Settings for the built-in access point (AP) are available (if the name is empty, the name GC97-XXXX will be used, where XXXX is the last character of the ESP mac address). You can also set authorization data for the WiFi network (priority, recommended mode):

[](https://github.com/alidonet/GC97_ESP/blob/main/manual/manual_options_wifi.png)

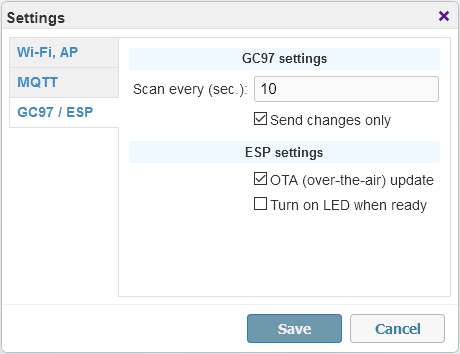
Network scanning is available with the ability to select the required one:

[](https://github.com/alidonet/GC97_ESP/blob/main/manual/manual_scan_networks.png)

You can specify the necessary MQTT broker settings:

[](https://github.com/alidonet/GC97_ESP/blob/main/manual/manual_options_mqtt.png)

Specify the period for polling the coulometer (no more than 1 time per second), this is also the period for sending data via the MQTT protocol:

[](https://github.com/alidonet/GC97_ESP/blob/main/manual/manual_options_misc.png)

If the "Send changes only" option is enabled, then requests to the MQTT broker will be executed only in case of changes and forcibly once every 10 polling cycles (for cases when retain is not used). You can also disable support for over-the-air updates (OTA, a reboot is required to apply the option). The "Always on LED" option inverts the operation of the built-in LED on the ESP board: constant light in operating mode.

**LED indication**

When loading the ESP, there is a series of short and rapid flashes for a second. Further, at intervals of every 2 seconds, single flashes inform about attempts to connect to the WiFi network, double flashes - to the MQTT server (if the parameters of both are specified).

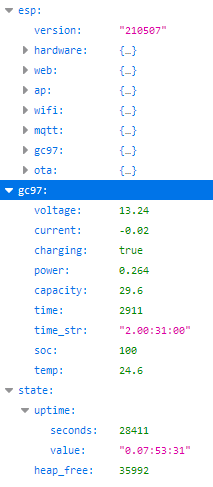
With the coulometer polling frequency (once every 10 seconds by default), you may see LED flashes. Quantity means condition:

* Single, moderately long flash - normal operation;
* A double flash indicates a Wifi connection problem;
* A triple flash indicates a problem connecting to the MQTT broker;
* Four flashes - unable to obtain data from GC97 coulometer (not connected or data is incorrect).

**Coulometer data**

All coulometer information, as well as most ESP status information and settings, are sent to the MQTT broker at a specified frequency in the form of json. The same data in json format can be directly requested at any time via a permanent web address:

* http://<ip\_address>/get/all.json -- the entire data packet
* http://<ip\_address>/get/gc97.json -- coulometer data
* http://<ip\_address>/get/esp.json -- ESP status and settings
* http://<ip\_address>/get/state.json -- uptime ESP

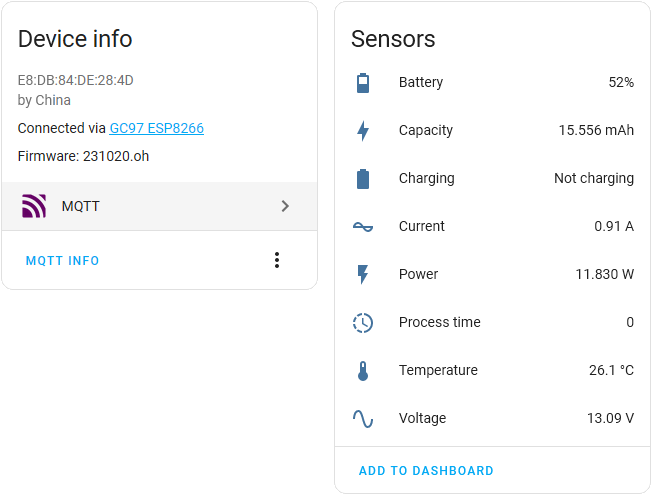
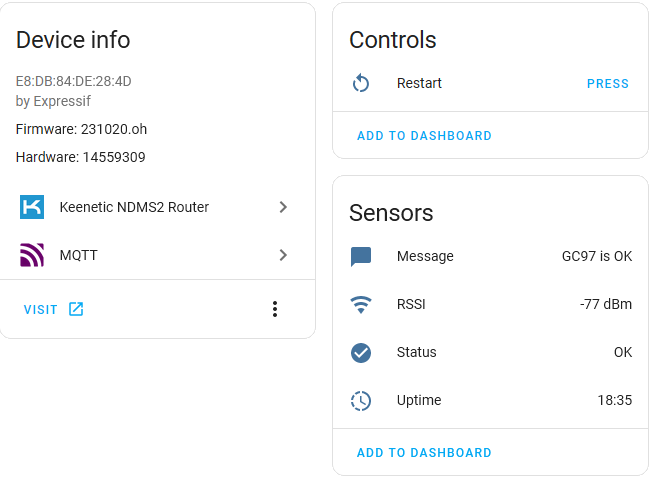
[](https://github.com/alidonet/GC97_ESP/blob/main/manual/manual_json.png)

**HASS Discovery**

If you use the GC97 as part of a Home Assistant-based smart home, the HASS Discovery option is available. When enabled, MQTT will send complete descriptions of all sensors and devices. As a result, within a few seconds you will have 2 new entries in the list of devices - for the coulometer itself and for the ESP board:

[](https://github.com/alidonet/GC97_ESP/blob/main/manual/manual_hass_devices.png)

At the same time, it supports working with several devices simultaneously. You can also remotely restart the device directly from the smart home interface and monitor its current state.

[](https://github.com/alidonet/GC97_ESP/blob/main/manual/manual_hass_gc97.png) [](https://github.com/alidonet/GC97_ESP/blob/main/manual/manual_hass_esp.png)

**Additional commands**

The following commands are available for web control:

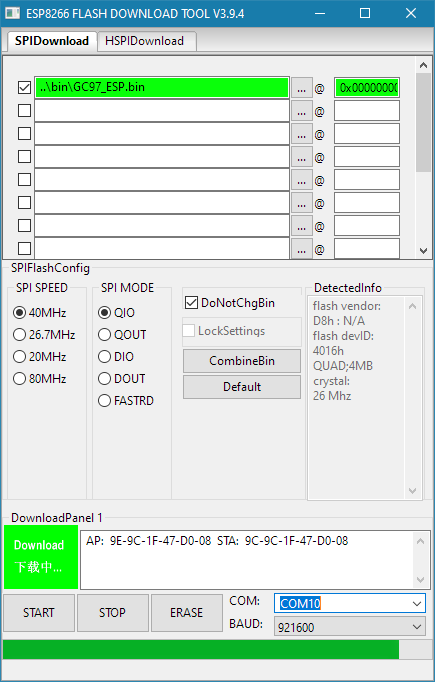
* http://<ip\_address>/cmd/reset -- reboot (available via the web interface);
* http://<ip\_address>/cmd/factory - reset all settings (can sometimes be useful, but if the controller is still accessible via the web :), otherwise only flashing).

**Firmware**

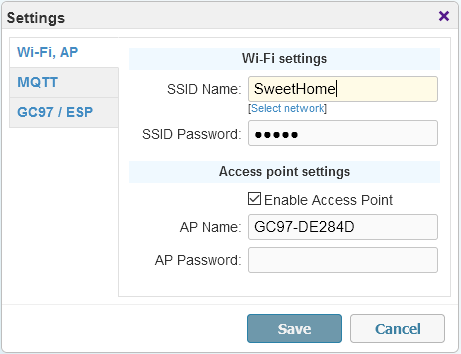
In the current version, only the main code needs to be flashed for correct operation. This can be done using any utility for flashing ESP, via Arduino IDE, etc. I recommend using [Flash download tools](https://www.espressif.com/en/support/download/other-tools) .

* **GC97\_ESP.bin** -- Main firmware with code (address 0x0000000) -- can also be used for OTA updates;

1. Launch the firmware utility, specify the ESP8266 controller and select the file with the firmware (first term):

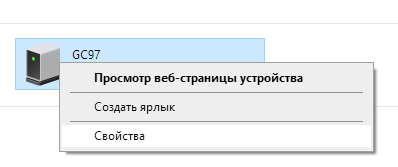
[](https://github.com/alidonet/GC97_ESP/blob/main/manual/manual_flashing_single.png)

1. At the bottom, select the COM port to which the controller is connected and press the "START" button;
2. After finishing the firmware, close the utility and reboot the controller; If desired, the operation of the controller can be monitored using any COM port monitoring utility;
3. We find an open access point named GC97-XXXXXX, where XXXXXX is the last characters of the mac address of your ESP board; We connect to this access point (by default - without a password);
4. Open the page [http://192.168.4.1](http://192.168.4.1/) in the browser ; The interface should be displayed - in the top line, click “Settings”;

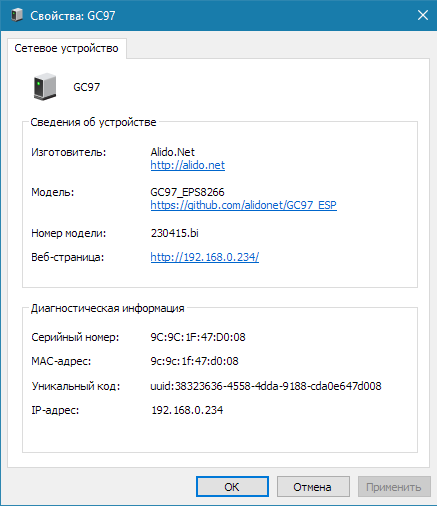
[](https://github.com/alidonet/GC97_ESP/blob/main/manual/manual_options_wifi.png)

1. Specify the network name and password (case sensitive) of your WiFi network; Save the settings, wait 15-20 seconds;
2. Reboot the ESP and wait for it to come online (may take up to 30 seconds);
3. Open [http://192.168.4.1](http://192.168.4.1/) and make sure that WiFi is connected and the controller has received an IP address on your network (for example, 192.168.1.100); Otherwise, check your network settings (point 6);
4. Now you can disconnect from the GC97-XXXXXX network and open a full interface at the new address issued to the controller in your WiFi network (for example, 192.168.1.100).

Alternatively, if you are using Windows, you can skip steps 8-9 and view the devices on your local network. This is usually a lengthy process, but this way you can also find out the address of the controller and open the control panel simply by clicking it. The GC97 will appear in the list of network devices:

[](https://github.com/alidonet/GC97_ESP/blob/main/manual/manual_gc97_network.png)

Context menu:

[](https://github.com/alidonet/GC97_ESP/blob/main/manual/manual_ssdp.png)

Alternatively, if the device is working normally and you want to update the firmware over the air (OTA), then this can be done from the console by running the command from the folder where the firmware file is located: , curl -F "image=@GC97\_ESP.bin" <ip-адрес>/updatewhere *<ip address>* should be replaced with yours.