OpenGarage (Firmware v1.1.2) User Manual

This is the user manual for OpenGarage (Firmware v1.1.2). OpenGarage is a fully open-source product. Hardware and software details are all published at the **OpenGarage Github repository**. For additional details, video tutorials, technical support, and user forum, visit **http://opengarage.io**

What's New in Firmware 1.1.2?

Supports using both the built-in ultrasonic distance sensor as well as an external switch sensor; improved MQTT features including supporting authentication, custom topic; support for custom NTP server; new sensor noise filtering method; some bug fixing; temperature and humidity values are now reported to both MQTT and Blynk (legacy) app. **NOTE**: this firmware has changed the ESP8266 core library. Upgrading from a previous firmware to this firmware may trigger a factory reset. So keep a copy of your settings before updating, as the controller may reset to factory default after updating -- if that happens, the controller will revert back to WiFi Access Point mode, please follow WiFi setup step again.

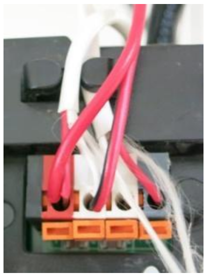
Software Setup

- **Update**: as of September 2021, Blynk (legacy) app has closed new user registration. To allow new OpenGarage users to continue using the Blynk app, we've replicated the Blynk server at blynk.openthings.io Please refer to the instructions in this solution article.
- OpenGarage supports a built-in web interface for local access, and remote access through the **Blynk (legacy)** app. We also provide the cross-platform **OpenGarage mobile app**. Before proceeding, please install the **Blynk (legacy) app**, (if you are a new Blynk user, please follow the solution article above to create an account), and scan **the QR code image here** to create the OpenGarage project.
- You can log into **the same Blynk account** on multiple mobile phones, so that you and your family members can all share access to the same device.
- If you need to set up multiple OpenGarage devices, just scan the same QR code as many times as you need. Each time you scan, the project will be replicated. You can give each project a unique name, and each comes with its unique authorization token.
- After the project is scanned, go to project settings and click **Email All (or Copy All)** to obtain the **32-digit authorization token**, which you will need in the following steps.
- **Power on** the device by plugging in a USB cable. The first time it powers on (or after WiFi/factory reset), the device will create an open WiFi AP named **OG_** followed by the last 6 digits of its MAC address (example: OG_67FG8A). This is the so-called **AP mode**. Use your phone or computer to **connect to this WiFi.**
- Your phone or computer should prompt you to Sign In to the WiFi network. If you don't see the prompt, just open a browser and type: http://192.168.4.1. Follow the instructions there to select or input your WiFi router's SSID, and your WiFi password. If you already have a Blynk token, paste it to the **Cloud Token** box; if not, just leave that box empty and you can always set it in later.
- Once the device successfully connects to your WiFi, you will **hear a short tune** from the buzzer on OpenGarage, and the device will reboot in 10 seconds. At this point it has remembered your router's SSID and password, and the next time it's powered on it will automatically enter client mode, and obtain an IP from your router. To access the device using a browser, type in **the client mode IP** (note: this is **NOT** the 192.168.4.1 IP as you did in AP mode step above). To perform certain actions, such

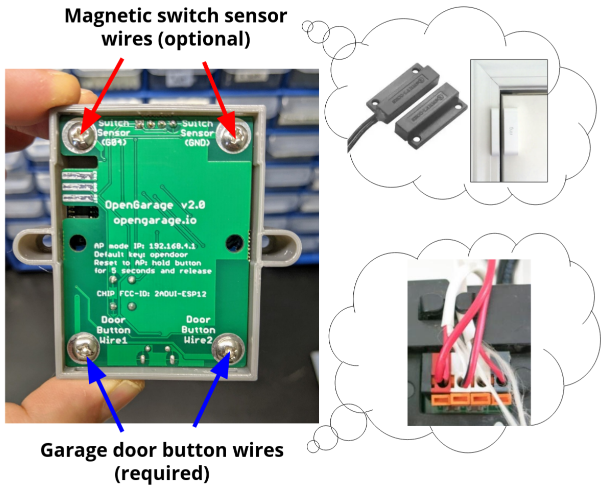
as trigger a button click, change options, you will need a device key. The default key is: **opendoor**

- If you don't know your OpenGarage's IP address, there are several ways to obtain it: you can find it on your WiFi router's configuration page; alternatively, there is an audible-way to obtain the IP address by hearing sound from the buzzer, as explained below; in addition, the device creates a broadcast DNS name, which is the AP mode name followed by .local/ which you can type into browser URL directly to access it (using the AP name example above, it would be OG_67FG8A.local/). This host name can be customized in the options (in the options->Advanced tab).
- To re-configure WiFi, you can **Reset it to AP mode**, either through the web UI, or using the pushbutton on OpenGarage. Specifically, press and hold the button for **5 seconds, until the LED changes status** (i.e. from on to off, or vice versa), but no more than 10 seconds, then release the button. Resetting the device to AP mode will NOT erase settings or log data.
- To perform **Factory Reset**: hold the push-button on the device for **10 seconds or more**, during which the LED will turn on and then turn off. Then release the button.
- If you briefly **click the pushbutton**, it triggers the relay. If you press and hold the button for more than 2 seconds but less than 5 seconds (i.e. before LED turns on), then release the button, that triggers the **Report IP** feature, which reports the IP using buzzer tones. Each digit of the IP is indicated by counting notes, starting from C4 as 1. For example, if the IP is 192.168.1.10, you will hear C4 (the leading 1), followed by a pause; then C4-C#4-D4… continuously until G#5, indicating 9, followed by a pause; then C4-C#4-pause, indicating 2; then a high-pitch tone, indicating a **dot**. Then C4-pause again, indicating the leading 1 in 168, and so on. By counting the number of continuously increasing notes, you can obtain each digit of the IP.

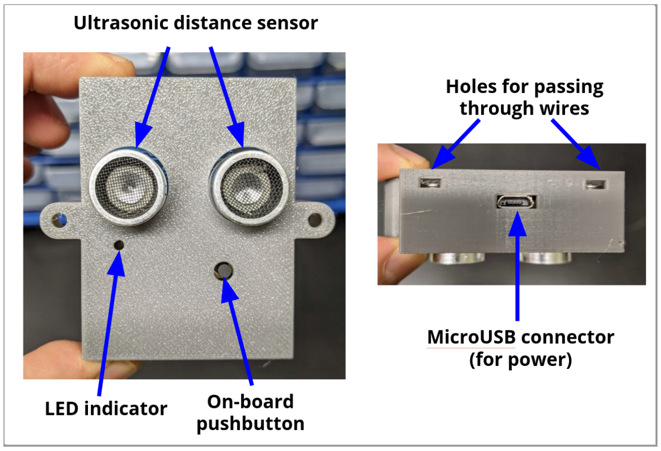
Hardware Setup

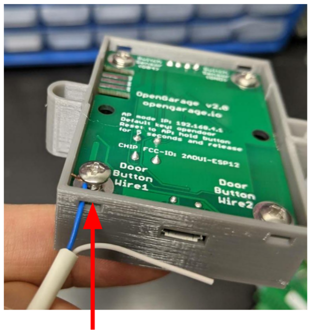


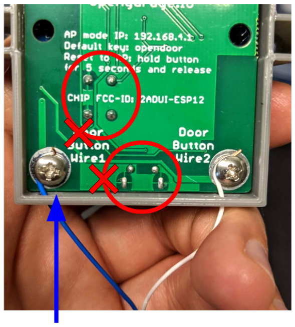
- **NOTE**: OpenGarage is **NOT waterproof**. If you plan to use it outdoors, you must place it inside a waterproof box.
- **Locate Door-button Terminals**: OpenGarage uses a **built-in relay** to simulate button click. First, on your garage door system, **locate the terminals that connect to your door-button** (a.k.a. wall-button or console). Most garage door systems have four terminal ports: two connect to the door-button, and two connect to sensors. Refer to your garage door system user manual if you have trouble locate the terminals.
- **Wiring**. Take out the two-wire cable from the package, strip the wires to appropriate length. Next, connect the wires to the controller, depending on the specific hardware version of your OpenGarage:
  - For **OpenGarage v2.0**: the device comes with a 3D printed enclosure and integrated screw terminals. Flip the controller to the back and locate the screw terminals at the four corners. The two at the bottom are for connection to door button wires (required), and the two at the top are for connection to a magnetic switch sensor (optional).



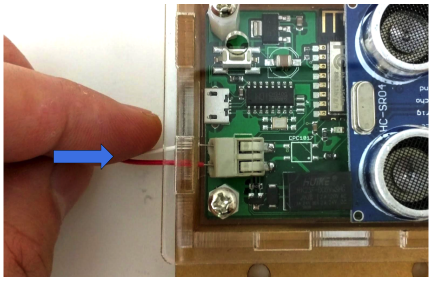
The images below show the front and side views of OpenGarage v2.0 and the steps to connect a wire. First, **untighten the screw**; then, **insert the stripped copper wire** through the enclosure hole; next, use a tweezer, screw-driver, or needle-nose plier to **wrap the copper wire around the screw** for at least one full circle; finally **tighten the screw** securely. Double check the wiring and make sure to trim the remaining copper wire so that it stays away from the exposed solder joints as shown on the right below.

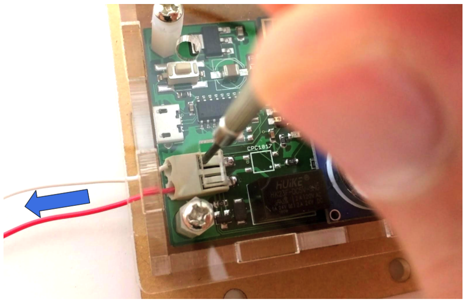

Ultrasonic distance sensor
Holes for passing through wires
MicroUSB connector (for power)
LED indicator
On-board pushbutton


Wrap the stripped copper wire around the screw for at least one full circle


Trim the remaining copper wires and make sure they stay away from nearby solder joints as indicated by the red circles

- o For **OpenGarage 1.x**: if your device comes with an **orange** or **blue** terminal block, insert the stripped wires into the terminal block and tighten the screws on the terminal block.
- o If your device comes with a **white** surface mount terminal block (as shown below), slowly and firmly push the stripped wires into the terminal holes. The terminal block has internal hooks to prevent the wires from coming out on their own. To release the wires, use a small screwdriver to press on the tabs of the terminal block, then pull the wires out.

Insert: 

Release: 

- o Insert the other end of the stripped wires to the door button terminal you located above. Because OpenGarage uses a relay to simulate button click, **the two wires have no polarity**.

- **Mounting**. The typical way is to mount OpenGarage to the ceiling, with the distance sensor facing down. Generally you position it such that **when the garage door is fully open, the distance sensor can 'see' the door.** After you've decided on the location, use screws or double sided tape to securely mount the device to the ceiling.
- Next, **plug in the provided microUSB cable to OpenGarage**, and power OpenGarage through a USB adapter (any USB adapter with at least 1amp output current is sufficient).
- **Security+ 2.0 System**. If your garage door system has a **Yellow antenna** and **Yellow learn button**, it's one of the newest Security+ 2.0 systems. For these systems, you need a Security+ 2.0 adapter, which

you can purchase as an add-on to OpenGarage. If your garage door system has a different colored antenna (other than yellow), most likely you don't need this adapter.

User Interface - Settings

The device key is required to change any option below. The default device key is **opendoor**. Many options require a reboot to take effect. On the Basic tab, you have:

- **Device Name:** custom device name (this name will show up on the OpenGarage homepage).
- **Distance Sensor:** select the method the controller is mounted:
  - **Ceiling Mount**: this is the default option, which uses the built-in ultrasonic distance sensor and the controller is mounted on the ceiling facing down.
  - **Side Mount**: for **rollup** type garage door systems where ceiling mount is not feasible; in this case, it's easier to set up OpenGarage on the side of the door, facing the outside.
- **Door Threshold:** the distance at which the door is regarded as open. This threshold should be slightly larger than the distance from the ceiling to the door when the door is fully open, and smaller than the vehicle threshold below. The unit is centimeter. For example, if the distance from the sensor to the door when the door is fully open is 30cm, you can set this to be 50cm to allow some margin.
- **Car Threshold:** the distance at which car is detected. This threshold should be slightly larger than the distance from the ceiling to the top of your vehicle parked in the garage. The unit is centimeter. For example, if the distance from the sensor to the top of your car is 80cm, you can set this to 100cm to allow some margin.
- **Status Check Interval:** how often the controller checks door status. The unit is second.
- **Click Time:** time that the relay holds when it clicks. Default is 1000ms (1 second).
- **Switch Sensor:** you can install an additional switch sensor (e.g. magnetic sensor) between pins G04 and GND to detect door status. Google 'magnetic sensor garage' you will see plenty of examples. Once enabled, the switch sensor status will be reported at the homepage. The options here are:
  - **(none)**: switch sensor is not installed
  - **Normally Closed**: this is the most common type (i.e. when door is closed, the magnetic sensor is closed)
  - **Normally Open**: when the door is closed the magnetic sensor is open.
- **Sensor Logic:** if switch sensor is enabled, this option becomes available. You can choose which sensors are used to determine the door 'open' status:
  - **Distance Sensor Only**: use only distance sensor to determine door status
  - **Switch Sensor Only**: use only switch sensor to determine door status (distance value will still be reported at the homepage, but will not be used to determine door status).
  - **Distance AND Switch**: use AND logic -- in other words, the door is regarded as open only if both distance sensor and switch sensor report 'open' status.
  - Distance OR Switch: use OR logic -- in other words, the door is regarded as open if either distance sensor or switch sensor report 'open' status.
- **T/H sensor**: choose the type of temperature/humidity sensor attached to the controller (requires additional sensor and soldering). Once configured and sensors are connected correctly, the temperature and humidity values will be reported on the homepage, as well as MQTT and Blynk. *[effective after reboot]*. Supported sensors include: AM2320 (SDA on G04, SCL on G05), DHT11/DHT22 (on G05), DS18B20 (on G05, requires 10K pullup resistor). All can be powered by VCC (3.3V) and GND.

- **Sound Alarm:** time that the alarm will sound to alert the user before each door action.
- **Log size:** defines how many log records you'd like to keep. If you change the log size, please go to the Homepage and click Clear Log for the change to take effect.

On the Integration tab, you have:

- **Blynk token:** Blynk authorization token [*effective after reboot*].
- **Blynk Domain and Port**: if you use custom Blynk server or port, configure them here.
- **IFTTT Key:** IFTTT webhook service key. To use this feature, go to ifttt.com, create an account, search 'webhook' service and create a webhook key. Copy and paste the key here. You can then **create IFTTT Applets** that use 'webhook' as trigger (this), **opengarage** as event name, and SMS, email, or push notification as action (that). When the device sends notifications, you will receive the message through SMS, email, or push notifications. The content of the message is passed via parameter **value1** in IFTTT Applet. Without IFTTT, you can still receive push notifications through the Blynk app.
- **MQTT Server and port:** MQTT server url (either IP or domain name) and port. Once MQTT server is filled in, the following three options also become enabled:
- **MQTT Username and password:** used for authentication if your MQTT broker requires them.
- **MQTT Topic**: custom MQTT topic. If left empty, the firmware will use the device name as topic.
- **Choose Notifications:** choose what notification messages you'd like to receive (this applies to both Blynk notifications within the Blynk app, and also IFTTT and MQTT notifications).
- **Automation:** if the door is left open for more than the specified amount of time, you can choose for the device to **notify** you and / or **auto-close** the door. When auto-close is selected, a minimum 5-second sound alarm will be enforced to alert the user before the door is closed.
  Similarly, if the door is left open till a specified UTC time, you can choose for the device to notify you an / or auto-close the door. Note that this is UTC time (as the controller does not know your local time zone). For example, if you want it to trigger at 6pm (18:00) and your local time zone is UTC-4, then set the time in the box to 22 (as the UTC time would be 22 when it's 6pm your local time).

On the Advanced tab, you have:

- **Read Interval**: time between every distance sensor reading. Default is 500ms. *[effective after reboot]*. Increase the reading interval may help reduce sensing noise.
- **Sensor Filter**: choose sensor noise filtering method.
  - **Median Filter:** use the median value of the most recent 7 readings.
  - **Consensus**: checks whether the most recent 7 readings are in consensus (i.e. the difference between max and min is within a margin) -- if so the average value is returned otherwise it's ignored until next time the readings are in consensus again.
  - **Margin**: margin for the consensus filter (default value is 10 cm).

  By default the firmware uses the consensus filter. If you find it leads to frequent failure of readings, suitably increase the margin to 20 or more. If this still doesn't solve the problem, change to Median filter and try again.
- **Distance Timeout**: if the distance sensor encounters a timeout (i.e. the ultrasonic signal was not received after a maximum amount of time has passed), should the reading be ignored or capped at maximum value (450cm). Default is 'Ignore', which can help reduce sensing noise.
- **HTTP port:** custom HTTP port (default is 80) [*effective after reboot*].

- **Host name:** custom host name. If left empty, it will use the WiFi AP name as host name. This allows you to use domain name **host_name.local/** to access the controller instead of using the numeric IP.
- **NTP server**: custom NTP server url. If left empty, it will use a default NTP server (e.g. pool.ntp.org).
- **Use Static IP**: configure the device to use custom IP [*effective after reboot*].
- You can also change the device key on this tab.

Firmware Upgrade

As new firmwares become available, you can download the new firmware (call **og_x.x.x.bin**) and upgrade firmware by clicking on the **Update** button next to the Firmware Version at the bottom of the webpage. Then follow the instructions on the Update page.

Links and Resources

- **OpenGarage Homepage** (including product page, technical support, and user forum)
- **OpenGarage Github Repository**
- **OpenGarage Firmware Files**
- **OpenGarage Documentation** (including API)
- **OpenGarage Blog Post**