OpenGarage Firmware 1.2.3 API Document [Oct 9, 2024]

1. Overview

This document describes OpenGarage (OG) Firmware 1.2.3 API.

- This firmware has added support for Email notifications.
- Note: new changes since the last API (1.2.0) are highlighted in green.
- When device key is required, use **`dkey=xxx`**. At factory reset, the default device key is opendoor.
- The device IP address is referred to as **`devip`**
- For most commands, parameters are optional and the order of parameters does not matter.
- **Return values** are all formatted in JSON, for example: **`{"result":1}`**
- **Return error code**:
  - 1 Success
  - 2 Unauthorized (e.g. missing device key or device key is incorrect)
  - 3 Mismatch (e.g. new device key and confirmation key do not match)
  - 16 Data Missing (e.g. missing required parameters)
  - 17 Out of Range (e.g. value exceeds the acceptable range)
  - 18 Data Format Error (e.g. provided data does not match required format)
  - 32 Page Not Found (e.g. page not found or requested file missing)
  - 48 Not Permitted (e.g. cannot operate on the requested station)
  - 64 Upload failed (e.g. OTA firmware update failed)

2. Get Controller Variables: `http://devip/jc`

Returned JSON variables:

- dist: distance sensor value (unit: cm)
- sn2: switch sensor value (only if switch sensor is enabled)
- door: door status (binary, 1 means open, 0 means closed)
- vehicle: vehicle status (1 means vehicle detected, 0 no, 2 means unknown)
- rcnt: read count (increments every time distance sensor is read)
- fwv: firmware version
- name: device name
- mac: MAC address
- cid: WiFi chip ID
- rssi: WiFi signal strength (dBm)
- cld: cloud option (0:none; 1:Blynk; 2:OTC)
- clds: cloud connection status. The values are different for Blynk and OTC:
  for Blynk: 0:disconnected; 1:connected
  for OTC: 0:not enabled; 1:connecting; 2:disconnected; 3:connected
- temp: temperature reading (Celcius), only if T/H sensor is enabled
- humid: humidity reading (relative percentage), only if T/H sensor is enabled

3. Change Controller Variables:

http://devip/cc?dkey=xxx&click=1&close=1&open=1&reboot=1&apmode=1

Parameters:

- dkey: (**required**) device key (factory default device key is opendoor)
- click/close/open: (optional) trigger relay click / close door / open door
- reboot: (optional) reboot device
- apmode: (optional) reset device in AP mode (to reconfigure WiFi settings)

Examples:

- http://devip/cc?dkey=xxx&click=1 trigger relay click (i.e. toggle door)
- http://devip/cc?dkey=xxx&close=1 close door (ignored if the door is already closed)
- http://devip/cc?dkey=xxx&reboot=1 reboot device

4. Get Options: http://devip/jo

Returned JSON variables: (factory default values indicated in bold font)

- fwv: firmware version (read-only)
- sn1: sn1 (distance sensor) mounting type (**0: ceiling mount**; 1: side mount)
- sn2: sn2 (switch sensor) type (**0: none**; 1: normally closed; 1: normally open)
- sno: sensor logic -- door 'open' status is determined by: (**0: use sn1 only**; 1: sn2 only; 2: sn1 AND sn2; 3: sn1 OR sn2)
- dth: door distance threshold (unit: cm, used to detect if door is open)
- vth: vehicle distance threshold (unit: cm, used to detect if vehicle is present)
- riv: status check and report interval (unit: second, default **5**)
- alm: alarm (0: no alarm; **1: 5-second alarm**; 2: 10-second alarm)
- aoo: alarm on opening (**0: no**; 1: yes)
- lsz: log size (i.e. 50 means the controller keeps the most recent 50 records)
- tsn: temperature/humidity sensor (**0:none**; 1:AM2320; 2:DT11; 3:DHT22; 4:DS18B20)
- htp: http port (default **80**)
- cdt: click delay time (unit: ms, default **1000**)
- dri: distance reading interval (unit: ms, default **500**)
- sfi: sensor filtering method (0: median; **1: consensus**)
- cmr: consensus margin for the consensus method (unit: cm, default **10**)
- sto: sensor timeout option (**0: ignore**; 1: cap to maximum value)
- ati: automation time a (unit: minutes, detect if door is open for longer than ati)
- ato: automation option a (bit 0: notify; bit 1: auto-close)
- atib: automation time b (unit: UTC hour, detect if door is open after atib)
- atob: automation option b (bit 0: notify; bit 1: auto-close)
- noto: notification options (bit 0: door open events; bit 1: door close events)
- usi: use static IP (**0: use DHCP**; 1: use static IP)
  when usi=1, three additional string options are available: dvip, gwip, subn, which represent the custom device IP, gateway IP, subnet mask
- cld: cloud connection option (0:none; 1:Blynk; 2:OTC)
- auth: cloud authorization token
- bdmn: cloud server domain name (see Section 10 for details)
- bprt: cloud server port (see Section 10 for details)
- name: device name (default "**My OpenGarage**")
- iftt: IFTTT maker channel token
- ssid: the WiFi network OG is connected to currently

- mqen: mqtt enable (0:disable; 1:enable) Note: this option is added so that the other mqtt options values can be preserved when temporarily disabling mqtt.
- mqtt: mqtt server url (IP or domain name both allowed)
- mqpt: mqtt port (default is 1883)
- mqur: mqtt server user name (optional, if authentication is required)
- mqpw: mqtt password (this is NOT sent over /jo; instead, it can be changed by /co command)
- mqtp: mqtt topic (optional, if empty it will use the device name as topic)
- emen: email enable
- smtp: smtp server name (default: **smtp.gmail.com**)
- sprt: smtp server port (default: **465**)
- send: email sender (for authentication with the smtp server)
- apwd: email aapp password (for authentication with the smtp server)
- recp: recipient email (for receiving notification, can be the same as the sender)
- ntp1: NTP server url (optional, if customizing NTP server)
- host: Custom Host name (using mDNS feature: the full host name is host**.local/**, for example if host is 'testog' then the full host name should be **testog.local/**)
- dvip/gwip/subn/dns1: device ip / gateway ip / subnet mask / dns ip (when in static IP mode)

5. Change Options:

http://devip/co?dkey=xxx&nkey=xxx&ckey=xxx&opname=opvalue...

Options:

- For the list of option names (opnames), refer to Section 4 above.
- Some options cannot be modified through the /co command: including fwv, dkey. These are either read-only options, or should be set in a different way.
- To change device key, you must provide both nkey and ckey (new key, and confirm key respectively).

Examples:

- devip/cc?dkey=xxx&nkey=abc&ckey=abc — set device key to 'abc'
- devip/cc?dkey=xxx&dth=75 — set distance threshold to 75cm
- devip/cc?dkey=xxx&cdt=500 — set click delay time to 500ms
- devip/cc?dkey=xxx&auth=0123456789abcdef — set cloud authorization token
- devip/cc?dkey=xxx&htp=8080&riv=5 — set http port to 8080 and read interval to 5 seconds
- devip/cc?dkey=xxx&ifft=xxxx — set IFTTT maker channel token
- devip/cc?dkey=xxx&ati=5&ato=3 — set automation time a to 5 minutes and option to 0b11 (i.e. auto-notify and close if door is open for > 5 minutes).

6. Get Log Data: http://devip/jl

Returned JSON variables:

- name: device name
- time: device time (UTC epoch time)
- ncols: number of columns (3 or 4, depending on if sn2_value is attached)
- logs: log data - an array of log entries, each entry in the form of [time_stamp, door_status, distance_value, sn2_value]

Note: sn2_value is attached if sensor 2 (SN2) is enabled in options. This is indicated by the ncols parameter: it is 3 if sn2_value is not attached; 4 if it is.

7. Clear Log Data:

`http://devip/clearlog?dkey=xxx`
This command clears the log data.

8. Reset All

`http://devip/resetall?dkey=xxx`
This command resets the device to factory default settings.

9. MQTT
This firmware supports MQTT. To use it:

- Define MQTT broker (i.e. the '`mqtt`' parameter). If authentication is required, you can provide username and password ('`mqur`' and '`mqpw`' parameters).
- You can define a custom MQTT topic ('`mqtp`' parameter). If left empty, the firmware will use the device name as the topic. In the examples below, it's referred to as `OGTOPIC`.
- A new option `mqen` (i.e. MQTT enable) has been added so that the other option values are preserved when temporarily disabling mqtt. The firmware only connects to the MQTT server when both `mqen=1` and also `mqtt` is a valid server name. When `mqen=0`, MQTT is disabled.

- **Published messages**:
  - `/OGTOPIC/OUT/NOTIFY: sent when door has just opened or just closed`
  - `/OGTOPIC/OUT/STATE:  sent every 15 seconds to refresh current state`
  - `/OGTOPIC/OUT/STATUS: report device online offline messages`
  - `/OGTOPIC/OUT/JSON:   sent every 15 seconds with basic controller parameters such as distance value, switch sensor value, temperature/humidity value (if available)`

- **Subscribed messages**:
  - `/OGTOPIC/IN/STATE:   accepts state change request. If payload message is:`
    - `"open" or "close": will trigger action if door is not already in that state`
    - `"click": will trigger action regardless of the state of the door`

10. Cloud Connection

This section describes the support for Blynk cloud connection as well as OTC (OpenThingsCloud connection).

- Option `cld` (refer to Section 4) defines the cloud option (`0:none; 1:Blynk; 2:OTC`).

- **Blynk**: as the Blynk team officially ended support for the Blynk legacy app and legacy server, we have replicated Blynk server (which is publicly available) on `openthings.io` so that existing users who have the Blynk legacy app can continue using the Blynk cloud connection. When using Blynk connection:
  - The default server domain name (`bdmn`) is `blynk.openthings.io`
  - The default server port (`bprt`) is: `8080`
    If you prefer to use your own Blynk server and port, you can change these options accordingly

    After the firmware connects to the Blynk server successfully, the application side can use the **Blynk legacy HTTP API** to retrieve data and trigger garage door action. This can be done either through secure https at port `9443` (https://blynk.openthings.io:9443) or non-secure http at port `8080` (https://blynk.openthings.io:8080). For example:
  - https://blynk.openthings.io:9443/token/project where `token` is the authorization token (32-character long). This returns the entire project in JSON format.
  - https://blynk.openthings.io:9443/token/get/Vx returns the value of pin `Vx`. Examples:
    - `V0: door status`
    - `V3: distance sensor value`
    - `V4: car status`
    - `V6: temperature sensor value`
    - `V7: humidity sensor value`
  - https://blynk.openthings.io:9443/token/update/V1?value=1 updates pin `V1` to value `1`, which will trigger a garage door action.

- **OTC** (OpenThingsToken): this is based on our own OpenThingsFramework (OTF) proxy. When using OTC:
  - The default server domain name (`bdmn`) is `ws.cloud.openthings.io`
  - The default server port (`bprt`) is: `80`
    If you prefer to use your own OTC proxy, you can change these options accordingly.

    After the firmware connects to the OTC server successfully, the application side can use the same **OpenGarage API described in the previous sections of this document** to interact with the device, via secure connection at https://cloud.openthings.io. Examples:
  - https://cloud.openthings.io/forward/v1/token/ returns the homepage of the device, where `token` is the OTC authorization token.
  - https://cloud.openthings.io/forward/v1/token/jc returns the controller status in JSON.
  - https://cloud.openthings.io/forward/v1/token/jl returns the log data in JSON.
  - The other APIs are similar to previously described. The only difference is that when the request is coming from OTC cloud connection, device key (`dkey`) is not required as the authorization token itself serves as a globally unique secret key.