Opensprinkler Bee (OSBee) WiFi Firmware 1.0.2 API Document

(*Oct 2022*)

1. Overview

This document describes OpenSprinkler Bee WiFi (OSBeeWiFi) Firmware 1.0.2 API.

- In the following, when a device key is required, it is referred to as **dkey=xxx**. The default device key at factory reset condition is opendoor.
- The device IP address is referred to as **devip** (e.g. 192.168.1.2)
- For most commands, the order of parameters does not matter.
- Note: this firmware supports either Blynk cloud connection or OTC (OpenThingsCloud) connection. A new section (Section 12) about cloud connection has been added.
- New changes since firmware 1.0.0 are highlighted with green background color.
- **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)
  - `32` Page Not Found (e.g. page not found or requested file missing)
  - `48` Not Permitted (e.g. cannot operate on the requested zone)
  - `64` Upload failed (e.g. OTA firmware update failed)

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

Returned JSON variables:

- `fwv:` firmware version (e.g. 100 stands for 1.0.0)
- `sot:` solenoid type. 0: latching (default); 1: non-latching
- `utct:` current UTC time
- `pid:` current running program id (-1 indicates no program is running)
- `tid:` current task id (-1 indicates no task is running)
- `np:` number of programs
- `nt:` number of tasks
- `mnp:` maximum number of programs (6 by default)
- `prem:` program remaining time
- `trem:` task remaining time
- `zbits:` zone status bits (bit 0, 1, 2 for zone 1, 2, 3 respectively; active high)
- `name:` device name
- `mac:` device MAC address
- `cid:` device chip ID
- `rssi:` WiFi signal strength (in dB)
- `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
- `zons:` zone names

3. Change Controller Variables (cc): `http://devip/cc?dkey=xxx&reset=x&reboot=x&resetwifi=x`

Parameters:

- `dkey:` device key (factory default device key is opendoor)
- `reset:` reset all zones (the value does not matter)
- `reboot:` reboot device (the value does not matter)
- `resetwifi:` reset device to WiFi AP (Access Point) mode (the value does not matter)

Examples:

- http://devip/cc?dkey=xxx&reset=1 reset all zones
- http://devip/cc?dkey=xxx&reboot=1 reboot the device

4. Get Options (jo): `http://devip/jo`

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

- `fwv:` firmware version (e.g. 100 stands for 1.0.0). Read-only.
- `tmz:` time zone (each increment is a quarter time zone, with 48 being GMT+0:00) for example, 52 is GMT+1:00, 44 is GMT-1:00 and so on
- `sot:` solenoid type. 0: latching (default); 1: non-latching.
- `htp:` http port (default value is 80)
- `cld:` cloud connection option (0:none; 1:Blynk; 2:OTC)
- `auth:` cloud (e.g. Blynk) authorization token
- `cdmn:` cloud server domain name (see Section 12 for details)
- `cprt:` cloud server port (see Section 12 for details)
- `name:` device name
- `zon0:` zone 1 name
- `zon1:` zone 2 name
- `zon2:` zone 3 name
- `zons:` zone names in JSON array format
- `bsvo:` boosted voltage for opening (default: 21)
- `bsvc:` boosted voltage for closing (default: 21)
- `dim:` LCD dimming when idling (range: [0,10], default value is 2)

5. Change Options (co): `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 option are 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, use the `nkey` and `ckey` pairs (new key, and confirm key).

Examples:

- devip/cc?dkey=xxx&nkey=abc&ckey=abc set device key to 'abc'
- devip/cc?dkey=xxx&tmz=32 set time zone to GMT-4:00 (i.e. (32-48)/4)
- devip/cc?dkey=xxx&auth=abcdef set cloud authorization token
- devip/cc?dkey=xxx&zon0=abc&zon1=def set zone 1 and zone 2 names
- devip/cc?dkey=xxx&htp=8080&name=hello set http port to 8080 and device name to hello

6. Get Log Data (jl): `http://devip/jl`

Returned JSON variables:

- name: device name
- logs: log data - an array of log entries. Each entry is integer array and records a zone event in the following format:

  [time_stamp, duration, event, zone_id, prog_id, task_id]

  ★ time_stamp: time when the event happens (UTC epoch time)
  ★ duration: duration in seconds
  ★ event: 'o' stands for zone 'open' event; 'c' stands for 'close' event
  ★ zone_id: zone id (zone 1's id is 0 and so on)
  ★ prog_id: program id (program 1's id is 0 and so on)
  ★ task_id: task id (task 1's id is 0 and so on)

Examples:

- [1234567, 0, 'o', 1, 0, 3]: at time 1234567, zone 2 opened running program 1 and task 4
- [4567890, 100, 'c', 0, 3, 5]: at time 4567890, zone 1 closed. It ran for 100 seconds on program 4, task 6.

7. Delete Log Data (dl): `http://devip/dl?dkey=xxx`

Delete the entire log data.

8. Get Program Data (jp): `http://devip/jp`

Returned JSON variables:

- tmz: time zone (refer to Section 4)
- progs: program data - an array of programs. Each program is in the form of:

  {"config":xx, "sts":[...], "nt":xx, "pt":[...], "name": "xxx"}

  ★ config: a **3-byte long integer** encoding configuration data
    - **Byte[0]:** a bit field as follows:
      bit[0]: enable bit (0 means program is disabled, 1 means enabled)
      bit[1]: day type (0: weekly type; 1: interval day type)
      bit[2-3]: restriction (0: none; 1: odd day; 2: even day)
      bit[4-5]: additional start time type (0: none; 1: fixed; 2: repeating)
      bit[6-7]: (not used currently)
    - **Bytes[1] and [2]:**
      When day type is 'weekly', Byte 1 is a bit field where bits 0 to 6 define the enable bits for the 7 days from Monday to Sunday (e.g. a value of 5, or binary 0b101 means the program runs on Monday and Wednesday). Byte 2 is unused.

      When day type is 'interval', Byte 1 is the remainder, Byte 2 is the interval (e.g. [2, 5] means the program runs every 5 days, starting in 2 days).

  ★ sts: an array of start times. The first element is always the first start time.
    - When additional start time type is 'fixed', this is up to 5 fixed start times (number of minutes since midnight). A value of -1 means the start time is

disabled. For example, [360, 630, 1000, 1200, -1] means the program will run at 6:00, 10:30, 16:40, and 20:00.
    - When additional start time type is 'repeating', the first element is the first start time, the second element is the number of repeat times, and the third element is the interval. For example, [360, 3, 90, -1, -1] means the first start time is 6:00, then it repeats every hour and half (90 minutes) for 3 times.

★ nt: number of tasks (the maximum number of tasks is 32)

★ pt: program task array. Each element is a long integer encoding:
    - **Byte[0]:** a bit field defining the zone enable bits (bits 0, 1, 2 correspond to zones 1, 2, 3 respectively)
    - **Byte[1-2]**: unsigned integer defining the duration (in seconds) of the task.
    - For example, a task with value 81925 means it turns on zones 1 and 3 (81925 & 0xFF = 5 or 0b101) and the task lasts for 320 seconds (81925 >> 8 = 320).

★ name: program name (the maximum number of characters is 32)

9. Run Program (rp): http://devip/rp?dkey=xxx&pid=x&...

This command allows you to immediately start a program: it can be one of the existing programs, a manual program, a test program, or a quick run-once program.

Parameters:

- When pid = an existing program index (starting from 0), this command starts that program.
- When pid=77 (ASCII value of 'M'), this command starts a manual program. You will need to supply two additional parameters:
    - **zbits=x**: zone enable bits (e.g. zbits=4 means zone 3 will turn on)
    - **dur=x** : duration (in seconds)
    - Example: **pid=77&zbits=3&dur=300** turns on zones 1 and 2 manually for 5 minutes
- When pid=81 (ASCII value of 'Q'), this command starts a quick program that runs zones sequentially. You will need to supply one additional parameter:
    - **durs=[x,x,x]**: an array of running times (in seconds) for each zone.
    - Example: **pid=81&durs=[0,300,600]** turns on zone 2 for 5 minutes, and then zone 3 for 10 minutes.
- When pid=84 (ASCII value of 'T'), this command starts a test program. You will need to supply two additional parameters:
    - **zid=x**: zone index (0 is the first zone and so on)
    - **dur=x**: test duration (in seconds)
    - Example: **pid=84&zid=1&dur=330** tests zone 2 for five and half minutes.

10. Change Program Data (cp):

http://devip/cp?dkey=xxx&pid=x&config=x&sts=[x,x,x,x,x]&nt=x&pt=[...]&name=xxx

Parameters:

- The same as the program data elements explained in Section 8 above.
- All elements are required, except 'name', which is assigned a default name (e.g. Program 1, Program 2 etc.) if not supplied.
- When pid = an existing program index (starting from 0), this command modifies that program. When pid=-1, this commands appends a new program.

11. Delete Program Data (dp): `http://devip/dp?dkey=xxx&pid=x`

Delete the selected program, or all program data.

Parameters:

- `When pid = an existing program index, this command deletes that program.`
- `When pid=-1, this command deletes / resets all program data.`

12. 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

- **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 **OSBee 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.