Firmware 2.2.0(1) API Document

(Dec 23, 2022)

1. Overview

This document describes the OpenSprinkler Firmware 2.2.0(1) API, including JSON and HTTP GET commands.

- **Password is required** for all commands (the **pw** parameter). Your device password is referred to as **xxx.**
  - **The password is MD5 hashed by the app / web UI and therefore when using the API commands the password should be MD5 hashed too (all lower-case). You can find online MD5 hash tools to convert plaintext password to MD5.**
- In the following, OpenSprinkler's IP address is referred to as **os-ip**.
- For most commands, **parameters are optional** and the order of parameters does not matter. Parameters that do not appear in the command remain unchanged.
- **This firmware is only available for OS 2.3, 3.x, and all Linux-based platforms such as OSPi.**
- **Return Values** are all formatted in JSON. For example **{"fwv": 220}, {"result": 1},** etc. **Return Error Codes:**
  - **{"result":1}** **Success**
  - **{"result":2}** **Unauthorized** (e.g. missing password or password is incorrect)
  - **{"result":3}** **Mismatch** (e.g. new password and confirmation password do not match)
  - **{"result":16}** **Data Missing** (e.g. missing required parameters)
  - **{"result":17}** **Out of Range** (e.g. value exceeds the acceptable range)
  - **{"result":18}** **Data Format Error** (e.g. provided data does not match required format)
  - **{"result":19}** **RF code error** (e.g. RF code does not match required format)
  - **{"result":32}** **Page Not Found** (e.g. page not found or requested file missing)
  - **{"result":48}** **Not Permitted** (e.g. cannot operate on the requested station)

2. Major Changes

Compared to previous firmwares, some of the major changes in this firmware that affect the APIs include:

- For OS v3 only: support for remote access via OpenThings Cloud (OTC); also support for Over-the-Air (OTA) firmware update via both WiFi and wired Ethernet.
- Support for sequential groups (which generalizes and replaces the previous per-zone sequential flag); program date range (i.e. support setting a start date and an end date of a program); pausing and resuming station runs; shift stations forward when manually stopping a zone; negative master on adjustment time and positive master off adjustment time.

In the following, updated or newly introduced JSON variables (relative to the previous firmware) are highlighted by **green color**.

3. Get Controller Variables [Keyword /jc]

/jc?pw=xxx

Return Variables:

- **devt**: Device time (epoch time). This is always the local time.
- **nbrd**: Number of 8-station boards (including main controller).
- **en**: Operation enable bit.
- **sn1**: Sensor1 status bit (1: sensor is active; 0: sensor is inactive).
- **sn2**: Sensor2 status bit (OS v3 and OSPi only).
- **rd**: Rain delay bit (1: manually triggered rain delay is active; 0: no rain delay).
- **rdst:** Rain delay stop time (0: rain delay not in effect; otherwise: the time when rain delay is over).
- **sunrise**: Today's sunrise time (number of minutes from midnight).
- **sunset**: Today's sunset time (number of minutes from midnight).
- **eip**: external IP, calculated as (ip[3]<<24)+(ip[2]<<16)+(ip[1]<<8)+ip[0]
- **lwc**: timestamp of the last weather call/query (epoch time)
- **lswc**: timestamp of the last weather server response (epoch time, a value of 0 means it did not receive a successful response)
- **lupt**: last device reboot time (epoch time)
- **lrbtc:** cause of the last reboot cause (see List of Reboot Causes)
- **lrun**: Last run record, which stores the [**station index, program index, duration, end time**] of the last run station.
- **RSSI**: WiFi signal strength (in dB, available for OS 3.0 only)
- **mac**: MAC address

- **loc**: Location in GPS coord.
- **jsp:** javascript url
- **wsp:** weather script url
- **wto**: weather adjustment options (such as the weights and baseline for the **t**emperature, **h**umidity, **r**ainfall parameters).
- **wtdata:** raw data used to calculate the watering percentage (this is relayed from the weather server).
- **wterr**: error code of the last response from the weather server (see List of Weather Error Codes).
- **ifkey**: IFTTT Make channel key (*only available for OS 2.3 or above, and Linux-based OpenSprinkler*).
- **mqtt:** MQTT configuration parameters
- **curr**: total current draw of all zones (in milliamps). This is only available for controllers with current sensing capability.
- **sbits**: Station status bits. Each byte in this array corresponds to an 8-station board and represents the bit field (LSB). For example, 1 means the 1st station on the board is open, 192 means the 7th and 8th stations are open.
- **ps**: Program status data: each element is a 4-field array that stores the **[pid,rem,start,gid]** of a station, where **pid** is the program index (0 means none), **rem** is the remaining water time (in seconds), **start** is the start time, and **gid** is the (sequential) group id of the station. If a station is not running (sbit is 0) but has a non-zero pid, that means the station is in the queue waiting to run.
- **flwrt**: flow count window in unit of seconds (the firmware defines this as 30 seconds by default).
- **flcrt**: real-time flow count (i.e. number of flow sensor clicks during the last flwrt seconds).
- **pq:** pause (queue) state (0: not in pausing state; 1: in pausing state)
- **pt:** count-down timer of pausing state (in units of seconds)
- **nq:** number of elements in the program queue (i.e. number of zones running or waiting to run)
- **otc:** OpenThings Cloud (OTC) configuration parameters
- **otcs:** OTC connection status (0: not enabled, 1: connecting, 2: disconnected, 3: connected)
- **dname:** device name
- **gpio:** array of free gpio pins (can be used for defining GPIO stations).

4. Change Controller Variables [Keyword /cv]

/cv?pw=xxx&rsn=x&rbt=x&en=x&rd=x&re=x&update=x&ap=x

Parameters:

- **rsn**: Reset all stations (including those waiting to run). The value doesn't matter: action is triggered if parameter appears.
- **rbt**: Reboot the controller. Binary value (a value of 1 triggers this action).
- **en**: Operation enable. Binary value.
- **rd**: Set rain delay time (in hours). Range is 0 to 32767. A value of 0 turns off rain delay.
- **re**: Set the controller to be in remote extension mode (a value of 1 turns on remote extension mode).
- **ap**: reset to AP mode (without erasing controller and program settings, available for OS 3.0 only, for re-configure WiFi)
- **update**: trigger a firmware update, for OSPi or Linux based OpenSprinkler only.

Examples:

- http://os-ip/cv?pw=xxx&rbt=1 // reboot controller
- http://os-ip/cv?pw=xxx&en=0 // disable operation
- http://os-ip/cv?pw=xxx&rd=24 // set a 24-hour rain delay

5. Get Options [Keyword /jo]

/jo?pw=xxx

Return Variables: (options marked RO are read-only and not modifiable via the /co command)

- **fwv**: Firmware version (219 means Firmware 2.1.9). RO.
- **fwm**: Firmware minor version (increments with minor revisions to the firmware). RO.
- **tz**: Time zone (floating point time zone value * 4 + 48). For example, GMT+0:00 is 48; GMT-4:00 is 32, GMT+9:30 is 86. Acceptable range is 0 to 108.
- **dhcp**: Use DHCP (1: Yes; 0: No).
- **ip{1,2,3,4}**: Static IP (ignored if dhcp=1).
- **gw{1,2,3,4}**: Gateway (router) IP (ignored if dhcp=1).

- **dns{1,2,3,4}:** DNS IP (ignored if dhcp=1).
- **subn{1,2,3,4}**: Subnet Mask (ignored if dhcp=1).
- **ntp**: Use NTP sync. (1: Yes; 0: No).
- **ntp{1,2,3,4}**: NTP server IP (ignored if ntp=0).
- **hp{0,1}**: The lower and upper bytes of the HTTP port number. So http_port=(hp1<<8)+hp0.
- **hwv**: Hardware version. RO.
- **hwt**: Hardware type. Values are as follows: 0xAC (172) = AC power type, 0xDC (220) = DC power type. RO.
- **ext**: Number of 8-station expansion boards (not including the main controller).
- **sdt**: Station delay time (in seconds). Acceptable range is -600 to +600 seconds (-10 to +10 minutes), in steps of 5 seconds.
- **mas/mas2**: Master stations 1 and 2 index (a value of 0 means none). This firmware supports up to 2 master stations.
- **mton/mton2**: Master 1 and 2 on adjusted time (in steps of 5 seconds). Acceptable range is 0 to 600 (note: positive).
- **mtof/mtof2**: Master 1 and 2 off adjusted time (in steps of 5 seconds). Acceptable range is -600 to 0 (note: negative).
- **sn1t**: Sensor 1 type. (0: not using sensor; 1: rain sensor; 2: flow sensor; 3: soil sensor; 240 (i.e. 0xF0): program switch).
- **sn1o**: Sensor 1 option. (0: normally closed; 1: normally open). Default is normally open.
  (*note the previous urs and rso options are replaced by sn1t and sn1o*)
- **sn1on/sn1of**: Sensor 1 delayed on time and delayed off time (unit is minutes).
- **sn2t/sn2o**: Sensor 2 type and sensor 2 option (similar to sn1t and sn1o, ***for OS 3.0 only***).
- **sn2on/sn2of**: Sensor 2 delayed on time and delayed off time (unit is minutes).
- **wl**: Water level (i.e. % Watering). Acceptable range is 0 to 250.
- **den**: Operation enable bit. To modify this option, you must use the /cv command in Section 4.
- **ipas**: Ignore password.
- **devid**: Device ID.
- **con/lit/dim**: LCD contrast / backlight / dimming values.
- **bst**: Booster time for DC-powered controllers (in milliseconds). Acceptable range is 0 to 1000 (in steps of 4 milliseconds).
- **uwt**: Weather adjustment method. 0: manual adjustment; 1: Zimmerman method; 2: auto rain delay; 3: monthly adjustment
  Water restriction setting (for California) is indicated by bit 7.
- **lg**: Enable logging.
- **fpr{0,1}**: flow pulse rate (scaled by 100) lower/upper byte. The actual flow pulse rate is ((fpr1<<8)+fpr0)/100.0
- **re**: Remote extension mode. To modify this option, you must use the /cv command in Section 4.
- **dexp/mexp**: Detected/maximum number of zone expansion boards (-1 means cannot auto-detect). RO.
- **sar**: Special station auto-refresh. When enabled, special stations (e.g. RF or HTTP stations) will be automatically refreshed to make sure they are in sync with the controller status.
- **ife**: IFTTT events enable. This is a bit field. When a bit is set to 1, an IFTTT notification is sent upon that corresponding event.

○ bit 0: Program scheduled	○ bit 4: Reboot
○ bit 1: Sensor1 status update	○ bit 5: Status run
○ bit 2: Flow sensor status update	○ bit 6: Sensor2 status update
○ bit 3: Weather update (e.g. water level change, eip change)	○ bit 7: rain delay update

6. Change Options [Keyword /co]

/co?pw=xxx&[opname]=[opvalue]&loc=x&wto=x&ifkey=x&ttt=x&mqtt=x&otc=x&dname=x

Parameters:

- **loc**: Location string (url encoded). This can be city,state name, zip code, GPS coords, PWS station ID. Every time you use the app to set the location, the app will automatically change it to GPS cords (except for PWS station ID).
- **wto**: Weather options in JSON form and stripped of the terminal braces (url encoded). For example, Zimmerman method has three parameters for adjusting weights and baseline parameters of temp (t), humidity (h) and rain (r).
- **ifkey**: IFTTT Maker channel key (*only available for OS 2.3 or above, and Linux-based OpenSprinkler*).
- **ttt**: Set time manually (epoch time, ignored if ntp=1).
- **[opname]**: option's JSON name (see list of options in Section 5). Note that read-only (RO) options are not modifiable.
- **[opvalue]:** value assigned to the option

- **otc**: OpenThings Cloud (OTC) configuration parameters in JSON format but without the starting and ending curly brackets.
  Example: `otc="en":1,"token":"your_otc_token","server":"ws.cloud.openthings.io","port":80`
- **mqtt:** MQTT configuration parameters in JSON format but without the starting and ending curly brackets.
  Example: `mqtt="en":1,"host":"server","port":1883,"user":"","pass":""`
- **dname**: device name

Examples:

- http://os-ip/co?pw=xxx&loc=42.01,-75.22 // change location to GPS coord 42.01,-75.22
- http://os-ip/co?pw=xxx&hp0=144&hp1=31 // change HTTP port to 8080, i.e. 31*256+144=8080
- http://os-ip/co?pw=xxx&sdt=30 // set station delay time to 30 seconds
- http://os-ip/co?pw=xxx&sn1t=1&sn1o=1 // configure sensor1 to rain sensor, and set it to normally open type

7. Set Password [Keyword/sp]

/sp?pw=xxx&npw=xxx&cpw=xxx

Parameters:

- **npw**: New password (should be in MD5 hash).
- **cpw**: Confirmation (should be in MD5 hash).

Examples:

- http://os-ip/sp?pw=xxx&npw=e0ff85143dfa717536cbb668cc8f8e8b&cpw=e0ff85143dfa717536cbb668cc8f8e8b
  (change password to the MD5 hash of 'sprinkler')

Return Error Code:

- Refer to Section 1 for error codes. In particular, if npw or cpw is missing, the controller will return Data Missing error; if npw and cpw do not match, the controller will return Mismatch error.

8. Get Station Names and Attributes [Keyword /jn]

/jn?pw=xxx

Return Variables:

- **snames**: Array of station names.
- **maxlen**: Maximum number of characters allowed in each name.
- **masop/masop2**: Master/Master2 operation flag in the form of LSB bit field. One byte per 8-zone group, so each is an array where the number of elements is the total number of zones divided by 8. For example, if you have a total of 24 zones, this will be a 3-element array where each element corresponds to a group of 8 zones: the first byte for zones 1-8, second for 9-16 and so on. Say an element value is 254, that means the first zone in this group does not use master, while all other zones in this group use master.
- **ignore_rain/ignore_sn1/ignore_sn2**: Ignore rain delay / sensor1 / sensor2 flags (bit fields), in the same format as masop. For example, if ignore_rain contains an element whose value is 192, that means the 7th and 8th zones in this group ignore rain delay, while the other zones in this group obeys rain delay.
- **stn_dis/stn_spe**: zone disable / special flags (bit fields).
- **stn_seq:** zone sequential flag (bit fields) → although this variable is still present, it's only for backward compatibility and the firmware no longer uses this parameter.
- **stn_grp:** zone's sequential group id (0, 1, 2, 3…). Any number below 255 means a sequential group; id value of 255 means the parallel group.

Example Return:


```
{"masop":[255],"masop2":[0],"ignore_rain":[0],"ignore_sn1":[0],"ignore_sn2":[0],"stn_dis":[0],"stn_seq":[247]
,"stn_spe":[0],"stn_grp":[0,0,0,1,0,0,0,0],"snames":["S01","S02","S03","S04","S05","S06","S07","S08"],"maxlen
":32}
```

9. Get Special Station Data [Keyword /je]

/je?pw=xxx

Return Data Format:

{"0":{"st":xxx, "sd":"xxx"}, "1":{"st":xxx, "sd":"xxx"}, …}

The integer (0, 1, …) corresponds to the station id (only stations whose stn_spe bits are 1 would appear in the list). "st" indicates the special station type (see list below), and "sd" stores the corresponding special station data:

- st=0: standard station (this should never occur because only stations with st!=0 should appear in the list)
- st=1: RF (radio frequency) station. sd (16 characters) stores the 16-digit hex RF code.
- st=2: remote station. sd stores the 14-digit hex code of the remote station in the following order: ip address (8 bytes), port number (4 bytes), station index (2 bytes).
- st=3: GPIO station. sd stores 3 bytes: the first two define the GPIO index, the third byte indicates active state (1 or 0).
- st=4: HTTP station. sd stores up to 240 bytes, comma separated HTTP GET command with the following data: 1) server name (can be either a domain name or IP address), port number, on command, off command.

10. Change Station Names and Attributes [Keyword /cs]

/cs?pw=xxx&s?=x&m?=x&i?=x&j?=x&k?=x&n?=x&d?=x&p?=x&g?=x&sid=xx&st=xx&sd=xx

Parameters:

- **s?**: Station name. ? is the station index (starting from 0). For example, **s0=abc** assigns name 'abc' to the first station.
- **m?/n?:** Master1/Master2 operation flag bit field. ? is the board (i.e. group of 8) index (starting from 0). Specifically, m0 corresponds to the first 8 zones (1 to 8), m1 corresponds to zones 9 to 16, and so on. Refer to the **masop** variable in Section 8 above. For example, **m0=255** sets all stations on the main controller to use master; **m1=4** sets the 3rd station on the first expansion board to use master.
- **i?/j?/k?/d?/p?**: Station ignore_rain/ignore_sn1/ignore_sn2/disable/special flag bit field.
- **g?**: station's sequential group id → note that this is NOT a bit field, instead, the ? is the station index. For example, g0 is the first station, g1 is the second etc. The group id can be any value between [0, 255] inclusive.
- **sid, st, sd:** Station special data: sid is the station index, st is the special type, sd is the corresponding data. See Section 8.

Examples:

- http://os-ip/cs?pw=xxx&s0=Front%20Lawn&s1=Back%20Lawn // set the name of the first two stations
- http://os-ip/cs?pw=xxx&m0=127&s7=Garage //set name for station s7 and master operation bits for board m0)
- http://os-ip/cs?pw=xxx&sid=2&st=1&sd=00553300553c001c // set special station data for station s2
- http://os-ip/cs?pw=xxx&g4=1 // assign group id 1 to station s4.

11. Get Station Status [Keyword /js]

/js?pw=xxx

Return Variables:

- **sn**: Array of binary values showing the on/off status of each station. This tells you which stations are open.
- **nstations**: The number of stations (equal to the number of boards times 8).

Example Return: {"sn":[1,1,0,0,0,0,0,0],"nstations":8} // this value shows that the 1st and 2nd stations are currently open.

12. Manual Station Run (previously manual override) [Keyword /cm]

/cm?pw=xxx&sid=xx&en=x&t=xxx&ssta=x

Parameters:

- **sid**: Station index (starting from 0)
- **en:** Enable bit (1: open the selected station; 0: close the selected station).
- **t**: Timer (in seconds). Acceptable range is 0 to 64800 (18 hours). **The timer value must be provided if opening a station**.
- **ssta:** shift remaining stations in the same sequential group (0: do not shift remaining stations; 1: shift remaining stations forward). Only applicable when turning off a zone (i.e. en=0)

Examples:

- http://os-ip/cm?pw=xxx&sid=0&en=1&t=360 // open the 1st station s0 for 6 minutes
- http://os-ip/cm?pw=xxx&sid=3&en=0 // close the 4th station s3
- http://os-ip/cm?pw=xxx&sid=3&en=0&ssta=1 // close the 4th station s3 and shift remaining stations in the same sequential group forward

An error code will return if you try to open the master station (as the master cannot be operated independently), or open a station that's either already running or in the queue waiting to run. Manually started stations are assigned program ID of 99.

13. Manually Start a Program [Keyword /mp]

/mp?pw=xxx&pid=xx&uwt=x

Parameters:

- **pid**: program index (starting from 0 as the first program)
- **uwt:** use weather (i.e. applying current water level / percentage). Binary value.

Examples:

- http://os-ip/mp?pw=xxx&pid=0&uwt=0 // start the first program at 100% water percentage, i.e. not using weather
- http://os-ip/mp?pw=xxx&pid=1&uwt=1 // start the second program using the current water percentage

Note: all existing queued stations will be reset before launching the manually started program. An error code is returned if the program index is out of bound (either negative or larger than the last program index).

14. Get Program Data [Keyword /jp]

/jp?pw=xxx

Return Variables:

- **nprogs**: Number of existing programs.
- **nboards**: Number of 8-zone groups (including main controller).
- **mnp**: Maximum number of programs allowed (40 in this firmware).
- **mnst**: Maximum number of program start times (fixed time type) allowed (4 in this firmware).
- **pnsize**: Maximum number of characters allowed in program name (32 in this firmware).
- **pd**: Array of program data. Each element corresponds to a program. See below for data structure.

Program Data Structure:

[[flag, days0, days1, [start0, start1, start2, start3], [dur0, dur1, dur2…], name, [endr, from, to]]]

- **flag:** a bit field storing program flags
  - **o** bit 0: program enable 'en' bit (1: enabled; 0: disabled)
  - **o** bit 1: use weather adjustment 'uwt' bit (1: yes; 0: no)
  - **o** bit 2-3: odd/even restriction (0: none; 1: odd-day restriction; 2: even-day restriction; 3: undefined)
  - o bit 4-5: program schedule type (0: weekday; 1: undefined; 2: undefined; 3: interval day)
  - o bit 6: start time type (0: repeating type; 1: fixed time type)
  - o bit 7: enable date range (0: do not use date range; 1: use date range)
- **days0/days1:**
  - o *If (**flag.bits[4..5]==0**), this is a weekday schedule:*
    - **days0.bits[0..6]** store the binary selection bit from Monday to Sunday; days1 is unused.
      For example, **days0=127** means the program runs every day of the week; **days0=21** (0b0010101) means the program runs on Monday, Wednesday, Friday every week.
  - o *If (**flag.bits[4..5]==3**), this is an interval day schedule:*
    - **days1** stores the interval day, **days0** stores the remainder (i.e. starting in day).
      For example, days1=3 and days0=0 means the program runs every 3 days, starting from today.
- **start0/start1/start2/start3 (***a value of -1 means the start time is disabled***):**
  - o **Start times support using sunrise or sunset with a maximum offset value of +/- 4 hours in minute granularity:**
    - If bits 13 and 14 are both cleared (i.e. 0), this defines the start time in terms of minutes since midnight.
    - If bit 13 is 1, this defines sunset time as start time. Similarly, if bit 14 is 1, this defines sunrise time.

If either bit 13 or 14 is 1, the remaining 12 bits then define the offset. Specifically, bit 12 is the sign (if true, it is negative); the absolute value of the offset is the remaining 11 bits (i.e. start_time&0x7FF).

- *If (**flag.bit6==1**), this is a fixed start time type:*
  - **start0, start1, start2, start3** store up to 4 fixed start times (minutes from midnight). Acceptable range is -1 to 1440. If set to -1, the specific start time is disabled.
- *If (**flag.bit6==0**), this is a repeating start time type:*
  - **start0** stores the first start time (minutes from midnight), **start1** stores the repeat count, **start2** stores the interval time (in minutes); start3 is unused. For example, [480,5,120,0] means: start at 8:00 AM, repeat every 2 hours (120 minutes) for 5 times.

- **dur0, dur1…:** The water time (in seconds) of each station. 0 means the station will not run. The number of elements here must match the number of stations. ***Unlike the previous firmwares, this firmware allows full second-level precision water time from 0 to 64800 seconds (18 hours)***. The two special values are: 1) 65534 represents sunrise to sunset duration; 2) 65535 represents sunset to sunrise duration.
- **name:** Program name
- **[endr, from, to]:** date range parameters, inclusive on both 'from' and 'to'.
  - **endr**: date range enable (the same value as bit 7 of flag).
  - **from:** integer value storing the start date. It's encoded as (month<<5)+day. For example, Feb 3 is encoded as (2<<5)+3=67. The default value is 33 (Jan 1).
  - **to:** the end date (encoded the same way as 'from'). The default value is 415 (Dec 31). Note that 'from' can be either smaller than, larger than, or equal to 'to'. If 'from' is larger than 'to', the range goes from 'from' to the 'to' date of the following year.

Example Return:

{"nprogs":3, "nboards":1, "mnp":40, "mnst":4, "pnsize":32,"pd":[[3,127,0,[480,2,240,0],[0,2700,0,2700,0,0,0,0],"Summer",[0,33,415]],
[2,9,0,[120,0,300,0],[0,3720,0,0,0,0,0,0],"Fall Prog",[0,33,415]],[195,16,0,[1150,-1,-1,-1],[0,0,0,0,0,0,0,64800,0],"Pipe",[1,67,415]]]}

15. Change Program Data [Keyword /cp]

/cp?pw=xxx&pid=xx&en=x&uwt=x&name=xxx&v=[flag,days0,days1,[start0,start1,start2,start3],[dur0,dur1,dur2…]]&from=xxx&to=xxx

Parameters:

- **pid**: Program index (starting from 0). Acceptable range is -1 to N-1, where N is number of existing programs. If pid=-1, this is adding a new program; otherwise this is modifying an existing program.
- **en:** enable/disable this program. NOTE: this is used to directly modify the enable bit. If this parameter is present, all other parameters are ignored!
- **uwt:** toggle using/not using weather adjustment on this program. NOTE: this is used to directly modify the uwt bit. If this parameter is present, all other parameters are ignored!
- **name:** Program name (url encoded, without quotes).
- **v:** Program data structure. The format is the same as explained in **Section 14** above, except **name**, **from**, and **to**.
- **from/to:** date range parameters, encoded the same way as explained in **Section 14** above.

Examples:

- http://os-ip/cp?pw=xxx&pid=0&en=0 // set the first program (index 0) to disabled.
- http://os-ip/cp?pw=xxx&pid=-1&v=[3,127,0,[480,2,240,0],[1800,1200,0,0,0,0,0,0]]&name=Summer%20Prog
  (add a new program, enabled, not using date range, use weather adjustment, no restriction, weekday schedule that runs on every day, repeating start time type, start at 8:00 AM, repeat every 4 hours for 2 times, and the running stations are 1st station – 30 minute, 2rd station – 20 minutes, program name is "Summer Prog").
- http://os-ip/cp?pw=xxx&pid=-1&v=[131,127,0,[480,2,240,0],[1800,1200,0,0,0,0,0,0]]&name=Winter%20Program&from=353&to=67
  (add a new program, enabled, using date range from Nov 1 to Feb 3, named "Winter Program", the other parameters being the same as the Summer Prog example above).

16. Delete Program(s) [Keyword /dp]

/dp?pw=xxx&pid=xxx

Parameters:

- **pid**: Program index (starting from 0). A value of -1 deletes all existing programs. Acceptable range is -1 to N-1, where N is number of existing programs.

Examples:

- http://os-ip/dp?pw=xxx&pid=1 // delete the second program
- http://os-ip/dp?pw=xxx&pid=-1 // delete all programs

17. Move Up (Re-order) a Program [Keyword /up]

/up?pw=xxx&pid=xxx

Parameters:

- **pid**: Program index (starting from 0). Acceptable range is 0 to N-1, where N is the number of existing programs.

Examples:

- http://os-ip/up?pw=xxxr&pid=2 // move the third program up before the second, i.e. switch the order of them
- http://os-ip/up?pw=xxx&pid=0 // will do nothing because the first program cannot be moved up

18. Start Run-Once Program [Keyword /cr]

/cr?pw=xxx&t=[x,x,x,….x,x]

Parameters:

- **t**: Timer value for each station. A value of 0 means the station will not run. Run-Once started stations are assigned program ID of 254.

Examples:

- http://os-ip/cr?pw=xxx&t=[60,0,60,0,60,0,600,0] // start a run-once program that turns on the 1st, 3rd, 5th, and 7th stations for 1 minute each)

19. Get Log Data [Keyword /jl]

/jl?pw=xxx&start=xxx&end=xxx&type=xxx or **/jl?pw=xxx&hist=n&type=xxx**

Return Value:

An array of log records for the time between **start** and **end** (both are epoch times), or the past n days (using **hist**). The maximum time span is 365 days. Each record is a 4-element array in the format of **[pid, sid, dur, end]**, where:

- **pid** is the program index (starting from 1 as the first program. 0 is a special value, see below)
- **sid** is the station index (starting from 0)
- **dur** is the duration (in seconds)
- **end** is the end time (epoch time).
- if flow sensor is enabled, there will be an additional field recording the flow rate during the station run.

When **pid=0**, it indicates a special event record, and the second field (sid) is a string indicating the event **type** (see list below).
Parameter **hist** specifies the number of days to go back in history, starting from today. So hist=0 returns the log data of the current day; hist=1 returns the log data of the current day and previous day, and so on.

Parameter **type** specifies which type of special log event you want to query. The supported types are:

- s1 or s2: sensor1/sensor2 events
- rd: rain delay events
- fl: flow sensor readings
- wl: watering level/percentage logs

Example Return: (e.g. by requesting http://os-ip/jl?pw=xxx&start=1413567367&end=1413657367)
[[3,17,616,1413511817], [0,"rd",86400,1413511845], [254,1,5,1413512107], [1,3,2700,1413552661], [5,3,1200,1413559201]]

20. Delete Log Data [Keyword /dl]

/dl?pw=xxx&day=n

Parameters:

- **day**: The day for which log data should be deleted. The parameter value is the epoch time of the day divided by 86400. For example, 16361 is Oct 18, 2014. If day=all, all log files will be deleted.

Examples:

- http://os-ip/dl?pw=xxx&day=16361 (delete the log file for Oct 18, 2014)
- http://os-ip/dl?pw=xxx&day=all (delete all log files)

21. Change Javascript URL [Keyword /cu]

/cu?pw=xxx&jsp=xxx

Parameters:

- **jsp**: Javascript path to be changed to (url encoded).

Examples:

- http://os-ip/cu?pw=xxx&jsp=https%3A%2F%2Fui.opensprinkler.com%2Fjs
  (change Javascript path to https://ui.opensprinkler.com/js)

22. Get all [Keyword /ja]

/ja?pw=xxx

This command returns the combined result of /jc, /jo, /jn, /js, /jp in one command, and put them under the "settings", "options", "stations", "status", "programs" subsections respectively.

Example Return:

{"settings":{.....},"options":{.....},"stations":{.....},"status":{.....},"programs":{.....}}
where …… is identical to the data obtained from querying each command individually.

23. Pause Queue [Keyword /pq]

/pq?pw=x&dur=xxx

This command triggers a pause with the specified duration (in units of seconds). Calling it first with a non-zero duration will start the pause; calling it again with duration 0 will cancel the pause and resume station runs.

Examples:

- http://os-ip/pq?pw=xxx&dur=600 (pause station runs for 10 minutes, i.e. 600 seconds)

24. Debug printout [Keyword /db]

/db

Print out debugging information, such as firmware build date, time, available heap/RAM size etc.

25. List of Reboot Causes:

(Reference: https://github.com/OpenSprinkler/OpenSprinkler-Firmware/blob/master/defines.h)

0: none/unknown	1: reboot due to factory reset	2: reboot triggered by buttons	3: reset to AP mode
4: API triggered reboot	5: API triggered reboot	6: switch from AP to client mode	7: firmware update
8: weather call failed for more than 24 hours	9: network failed for too many times	10: reboot due to first-time NTP sync	99: power-on

26. List of Weather Error Codes:

Weather error code of 0 means success. If the error code is a negative number, it means a network problem:

-1: request not received	-2: cannot connect to weather server	-3: request time out	-4 received empty return

If the weather error code is a positive number, refer to https://github.com/OpenSprinkler/OpenSprinkler-Weather/blob/master/errors.ts