Firmware 2.2.1(3) API Document

(Sep 1, 2025)

1. Overview

This document describes the OpenSprinkler Firmware 2.2.1(3) API.

- **Password is required** for all commands (via the **pw** parameter). Your device password is referred to as **xxx.**
  - **The password must be MD5 hashed (in lowercase), as the app and web UI handle hashing automatically. You can use online MD5 hash tools to convert a plaintext password to MD5 hash.**
- Throughout the document, OpenSprinkler's IP address is referred to as **os-ip**.
- Commands do NOT require all parameters to be included, and their order does not matter. Any parameter omitted from a request will remain unchanged.
- **This firmware is only available for OS 2.3, 3.x, and all Linux-based platforms such as OSPi.**
- Each command has a return value, formatted in JSON, such as **{"fwv":221}, {"result":1}**.
  Below is a list of **return error codes:**

{"result":1}	**Success**
{"result":2}	**Unauthorized** (missing password or password is incorrect)
{"result":3}	**Mismatch** (new password and confirmation password do not match)
{"result":16}	**Data Missing** (missing required parameters)
{"result":17}	**Out of Range** (value exceeds the acceptable range)
{"result":18}	**Data Format Error** (provided data does not match required format)
{"result":19}	**RF code error** (RF code does not match required format)
{"result":32}	**Page Not Found** (page not found or requested file missing)
{"result":48}	**Not Permitted** (cannot operate on the requested station)

2. Major Changes

Compared to previous firmwares, this update introduced the following API changes:

- AC- and DC-powered OpenSprinkler v2.3 and v3.x: **added options imin** and **imax** are added to define undercurrent and overcurrent thresholds.
- **Controller variables** (/jc) added three additional parameters: **ocs** (overcurrent status), **wtrestr** (weather restriction status), **wls** (aray of current multi-day average watering levels).
- **Run-once program** (/cr) added parameter **anno**, which carries the program name annotation.
- **Change controller variables** (/cv) added two parameters: **rrsn** (reset all running zones), and **rocs** (reset overcurrent status).

In the following, variables that are updated or newly introduced in this firmware (compared to the previous version) **are highlighted in green**.

3. Get Controller Variables [/jc]

/jc?pw=xxx

Returned Variables:

- **devt**: Device time (epoch time). This is always the local time (i.e. adjusted to the local time zone).
- **nbrd**: Number of 8-station groups/boards (including the 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 call response (a value of 0 means it did not successfully receive a response)
- **lupt**: Last device reboot time (epoch time)
- **lrbtc**:Cause of the last reboot (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.x only)
- **mac**: Hardware MAC address
- **loc**: GPS coord of the device location
- **jsp**: Javascript url
- **wsp**: Weather script url
- **wto**: Weather adjustment options (the specific fields depend on the adjustment method chosen).
- **wtdata**: Raw data relayed from the weather server, used to calculate the watering percentage.
- **wterr**: Error code of the last weather server response (see List of Weather Error Codes).
- **ifkey**: IFTTT Webhooks key (to enable IFTTT notifications).
- **mqtt**: MQTT configuration parameters
- **curr**: Total current draw of all zones (in milliamps). Only available for controllers with current sensing capability.
- **sbits**: Station status bits. Each byte in this array represents a LSB bit field for a 8-station group
  For example, 2 (0b00000010) means the 2nd station of the group is open, 192 means the 7th and 8th stations are open.
- **ps**: Program status data: each element represents a station and is an array of four numbers **[pid,rem,start,gid]**, where **pid** is the program index (0 means not running any program), **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 not currently running but is in the queue waiting to start.
- **flwrt**: Flow count window in units of seconds (by default this value is 30 as defined by the firmware).
- **flcrt**: Real-time flow count (number of flow sensor clicks during the last flwrt seconds).
- **pq**: Pause state (0: pause it inactive; 1: pause is active)
- **pt**: Count-down timer of an active pause state (in units of seconds)
- **nq**: Number of elements in the program queue (equivalent to the 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 (included in notification messages to help identify the controller that sent the messages)
- **gpio**: Array of free/spare gpio pins, to be used for defining GPIO stations.
- **email**: Email notification settings
- **ocs**: Overcurrent status (0=no overcurrent; 255=system overcurrent; other value = index of the station that triggered overcurrent. Station index starts at 1).
- **wtrestr**: Weather restriction status (0: inactive; 1: active).
- **wls**: Array of current multi-day average watering levels. The array length depends on the weather provider's available data (e.g. Apple supports up to 10 days; others may support fewer).

4. Change Controller Variables [/cv]

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

Parameters:

- **rsn**: Reset all stations (both running and queued). A value of 1 triggers the reset. 0 has no effect.
- **rbt**: Reboot the controller. A value of 1 triggers the reset. 0 has no effect.
- **en**: Operation enable. 1: enable operation; 0: disable.
- **rd**: Set rain delay time (in hours). Range is 0 to 32767. A value of 0 turns off an active rain delay.
- **re**: Set the controller in remote extension mode (a value of 1 turns on remote extension mode).
- **ap**: Reset to AP mode (for re-configure WiFi without erasing controller settings and programs).
- **update**: Trigger a firmware update (for OSPi or Linux based OpenSprinkler only).
- **rrsn**:Reset all running stations (but not queued ones). A value of 1 triggers the reset. 0 has no effect.
- **rocs**:Reset the overcurrent status (ocs). A value of 1 triggers the reset, 0 has no effect.

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 [/jo]

/jo?pw=xxx

Returned 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 revision number. 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.

Accepted value must be in the range 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). Accepted range is -600 to +600 seconds (-10 to +10 minutes), in increments of 5 seconds.
- **mas/mas2**: Master stations 1 and 2 indices (a value of 0 means none).
- **mton/mton2**: Master 1 and 2 On Adjusted time (in increments of 5 seconds). Accepted range is -600 to 600.
- **mtof/mtof2**: Master 1 and 2 Off Adjusted time (in increments of 5 seconds). Accepted range is -600 to 600.
- **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.
- **sn1on/sn1of**: Sensor 1 delayed-on time and delayed-off time (in units of minutes).
- **sn2t/sn2o**: Sensor 2 type and sensor 2 option (similar to sn1t and sn1o, for but Sensor 2).
- **sn2on/sn2of**: Sensor 2 delayed-on time and delayed-off time (in units of minutes).
- **wl**: Water level (i.e. % Watering). Accepted 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**: Boost time for DC-powered and Latch controllers (in milliseconds). Accepted range is 0 to 1000 (in increments of 4 milliseconds).
- **uwt**: Weather adjustment method. 0: Manual; 1: Zimmerman; 2: Auto Rain Delay; 3: ETo; 4: Monthly
  Water restriction setting (for California) is indicated by bit 7 of this byte.
- **lg**: Enable logging.
- **fpr{0,1}**: Flow pulse rate (scaled by 100) lower/upper byte. The total 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 expanders (-1 means the controller cannot auto-detect). RO.
- **sar**: Special station auto-refresh. When enabled, special stations (e.g. RF or HTTP stations) will be automatically refreshed to ensure they are synchronized with the master controller
- **fwire**: Force wired connection (OS 3.x only). Force the controller to stay in wired mode. (1: Yes; 0: No)
- **ife**: Notification events enable bits. It's a bit field. When a bit is set to 1, that notification type is enabled.

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

- **ife2**: Extended notification enable bits. Similar to **ife** but defines additional notification events.

○ bit 0: Station start ○ bit 1: Flow alert ○ bit 2: Current alert

- **imin**: Undercurrent threshold, in mA (scaled by 10). Example: imin=10 → 100 mA.
- **imax**: Overcurrent threshold, in mA (scaled by 10). Example: imax=120 → 1200 mA. A value of 0 indicates using the system default; 255 disables overcurrent detection.

6. Change Options [/co]

/co?pw=xxx&[opn]=[opv]&loc=x&wto=x&ifkey=x&ttt=x&otc=x&dname=x&mqtt=x&email=x

Parameters:

- **[opn]**: Option's name (see the option list in Section 5). Note that read-only (RO) options are not modifiable by /co.
- **[opv]**: New value to be assigned to the option
- **loc**: Location string (url encoded). This can be either "city,state name", "zip code" or "GPS coords".
  Every time you use the app to set the location, the app will automatically convert it to GPS coords.
- **wto**: Weather options in JSON, url encoded, and stripped of the braces on both sides. Example: Zimmerman method has six parameters for adjusting the weights and baseline values of temp (t), humidity (h) and rain (r).
- **ifkey**:IFTTT Webhooks key.
- **ttt**: Set time manually (epoch time, ignored if ntp=1).
- **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 stripped of the starting and ending curly brackets.
  Example: mqtt="en":1,"host":"server","port":1883,"user":"","pass":"","pubt":"opensprinkler","subt":""
- **email**:Email configuration parameters in JSON format but stripped of the starting and ending curly brackets.
  Example: email="en":1,"host":"smtp_server","port":465,"user":"user@gmail.com","pass":"","recipient":""
- **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 [/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 the plaintext password 'sprinkler')

Returned Error Code:

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

8. Get Station Names and Attributes [/jn]

/jn?pw=xxx

Returned Variables:

- **snames**: Array of station names.
- **maxlen**: Maximum number of characters allowed in each station name.
- **masop/masop2**: Master/Master2 operation flags. Each represents a LSB bit field, with one byte per 8-zone group. The data is stored as an array, where the number of elements equals the total number of zones divided by 8. For example, if there are 24 zones, the array will contain 3 elements, each corresponding to a group of 8 zones:
  First byte → zones 1-8; second byte → zones 9-16; third byte → 17-24. Each byte value indicates which zones in that group activate the master zone. For instance, if a byte value is 254 (0b11111110), it means the first zone in that group does NOT use master, while all other zones do.
- **ignore_rain/ignore_sn1/ignore_sn2**: These are bit field flags that determine whether a zone ignores rain delay / sensor1 / sensor2. They follow the same format as **masop** above, with one byte per 8-zone group, stored as an array. For example, if **ignore_rain** contains an element with value 192 (0b11000000), it means the 7th and 8th zones in this group ignore rain delay, while the others obey.

- **stn_dis/stn_spe**: Zone disable / special station bit field flags.
- **stn_grp:** Zone's sequential group id (0, 1, 2, 3…). Any number below 255 indicates a sequential group; while a value of 255 designates the parallel group.

Example Return:

{"masop":[255],"masop2":[0],"ignore_rain":[0],"ignore_sn1":[0],"ignore_sn2":[0],"stn_dis":[0],"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 [/je]

/je?pw=xxx

Return Data Format:

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

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

- st=0: Standard station (this should never occur, as only stations with st!=0 should appear in this list)
- st=1: RF (Radio Frequency) station. sd is either the legacy version RF code (16-digit hex code representing RF protocol 1 and 24-bit length), or the new version (25-digit hex code, prefixed with H, with full support for all RF protocols and bitlength).
- st=2: Remote station (IP). sd stores a 14-digit hex code of the remote station in the following structure: ip address → 8 bytes, port number → 4 bytes, station index -> 2 bytes.
- st=3: GPIO station. sd has 3 bytes: the first two are the GPIO index, the third indicates active state (1 or 0).
- st=4: HTTP station. sd stores up to 240 bytes, comma-separated HTTP GET command with the following structure: server name (it can be either a domain name or IP address), port number, on command, off command.
- st=5: HTTPS stations. sd stores up to 240 bytes, comma-separated HTTPs GET command with the following structure: server name (it can be either a domain name or IP address), port number, on command, off command.
- st=6: Remote station (OTC). sd stores 35 bytes, with the first 32 bytes storing the remote OTC token, followed by a '\0', and then 2 bytes storing the station index.

10. Change Station Names and Attributes [/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 the first station name as "abc".
- **m?/n?:** Master1/Master2 operation flag bit field. ? is the board index (starting from 0). Specifically, m0 corresponds to the first group of 8 zones, 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 8 zones on the main controller to activate master; **m1=4** sets the 3rd station on the first expansion board to activate master (while the others don't).
- **i?/j?/k?/d?/p?:** Station ignore_rain/ignore_sn1/ignore_sn2/disable/special bit fields.
- **g?:** Station's sequential group id. Note that this is NOT a bit field, instead, ? 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].
- **sid, st, sd:** Special station data: sid is the station index, st is the special type, sd is the 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 station 7's name 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 2
- http://os-ip/cs?pw=xxx&g4=1 // assign group id 1 to station s4.

11. Get Station Status [/js]

/js?pw=xxx

Returned Variables:

- **sn:** Array of binary values with the on/off status of each station. This indicates which stations are open.
- **nstations:** The number of stations (equals the number of boards times 8).

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

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

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

Parameters:

- **sid:** Station index (starting from 0)
- **en:** Enable bit (1: open; 0: close).
- **t:** Timer (in units of seconds), with a value between 0 to 64800 (18 hours). **This is required if opening a station.**
- **ssta:** Shift remaining stations in the same sequential group (0: do not shift remaining stations; 1: do).
  This parameter is only applicable when closing 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 group

An error code will occur if you attempt to open any master station (as they cannot be operated independently), or open a station that's either already running or queued to start. Manually started stations carry a program ID of 99.

13. Manually Start a Program [/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.

If the program name contains annotations, they are observed (previous firmwares ignored them when manually starting the program).

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 a 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 [/jp]

/jp?pw=xxx

Returned Variables:

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

Program Data Structure:

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

- **flag:** A byte representing the bit field storing program flags
  - bit 0: Program enable flag "en" (1: enabled; 0: disabled)
  - bit 1: Use weather adjustment flag "uwt" (1: yes; 0: no)
  - bits 2-3: Odd/even restriction (0: none; 1: odd-day restriction; 2: even-day restriction; 3: undefined)
  - bits 4-5: Start day type (0: weekly; 1: single-run; 2: monthly; 3: interval day)
  - bit 6: Start time type (0: repeating type; 1: fixed type)
  - bit 7: Enable date range (0: do not use date range; 1: use date range)
- **days0/days1:** These two bytes together store the start day data, and are formatted as follows:
  - If (**flag.bits[4-5]==0**), this is a **weekly** schedule. In this case:
    - **days0.bits[0-6]** store the binary selections (LSB) from Monday to Sunday, while **days1** is unused. For example:
      days0=127 (0b11111111) means the program runs every day of the week
      days0=21 (0b00010101) means the program runs on Monday, Wednesday, Friday every week.

- If (**flag.bits[4-5]==1**), this is a **single-run** schedule. In this case:
  - **days0** and **days1** together store a 16-bit unsigned integer encoding the day in epoch time (epoch time / 86400)

- If (**flag.bits[4-5]==2**), this is a **monthly** schedule. In this case:
  - **days0** stores the day of the month (starting from 1). A value of 0 indicates the last day of each month.

- If (**flag.bits[4-5]==3**), this is an **interval** day schedule. In this case:
  - **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:** Start times data, each a 16-bit signed integer.
  If (**flag.bit6==1**), the program is a fixed start time type. Each of start0, 1, 2, 3 is a 16-bit signed integer that represents an independent start time, in the following format:
  - If bit 15 is 1 (i.e. the integer value is negative), this start time is disabled.
  - If bits[13-14] are both 0: this is a standard time with a value between 0 (00:00 AM) and 1439 (11:59 PM).
  - If bit 13 is 1: this represents a **sunset** time. If bit 14 is 1, this represents a **sunrise** time.
    In either case, the remaining 13 bits define an offset from the sunset / sunrise time. Specifically, bit 12 is the sign (1 means negative); the remaining 11 bits (e.g. start0 & 0x7FF) give the absolute value of the offset.

  If (**flag.bit6==0**), the program is a repeating start time type. In this case:
  - **start0** stores the first start time in the same format as described above; **start1** stores the repeat count; **start2** stores the interval time (in units of 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...:** Each station's water duration (in units of seconds), in the range of 0 to 64800 (18 hours).
  0 means the station will not run. The number of elements in this array must match the number of stations.
  Two special values are: 65534 represents sunrise-to-sunset duration; and 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 (this value must be the same as **flag.bit7**).
  - **from:** An 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 in the same way as 'from'). The default value is 415 (Dec 31).

Note that 'from' may be either smaller than, larger than, or equal to 'to'. If 'from' is larger than 'to', it is assumed that the 'to' date is in 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],[0,0,0,0,0,0,64800,0],"Pipe",[1,67,415]]]}

15. Change Program Data [/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, in the range of -1 to N-1, where N is the 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 parameter is used to directly modify the 'en' bit.
  If this parameter is present, all other parameters are ignored!
- **uwt:** Set the Use Weather flag 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 explained in Section 14 above, except for **name**, **from**, and **to**.
- **from/to:** Date range parameters, in the format 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) [/dp]

/dp?pw=xxx&pid=xxx

Parameters:

- **pid**: Program index, in the range of -1 to N-1, with N being the number of existing programs. A value of -1 deletes ALL 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 [/up]

/up?pw=xxx&pid=xxx

Parameters:

- **pid**: Program index, in the range of 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 further moved up

18. Start Run-Once Program [/cr]

/cr?pw=xxx&t=[x,x,x,....x,x]&cnt=xxx&int=xxx&uwt=xxx&anno=xxx

Parameters:

- **t**: An array that defines the duration value for each station. A value of 0 means the station will not run. Stations started via this command are assigned a program ID of 254.
- **cnt**: Repeat count for the program. A value of 0 means the program will only run once and won't repeat.
- **int**: Repeat interval for the program (in units of minutes). Only allowed to be non-0 if **cnt** is also non-0.
- **uwt**: Use weather (i.e. applying current watering level). Binary value.
- **anno**: Program name annotation (in the format of **>X** where **X** is any supported letter defined in the OpenSprinkler User Manual under the Program name annotation section).

If a run-once program is set to repeat (both **cnt** and **int** are non-zero), the firmware automatically creates a single-run program named “Run-once with repeat”, followed by any optional annotation.

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)
- http://os-ip/cr?pw=xxx&t=[60,0,60,0,60,0,600,0]&cnt=3&int=15&anno=>N // start a run-once program similar to the above but also set it to repeat every 15 minutes for 3 times. This will automatically create a single-run program, and append the annotation >N to the program name (>N means run zones in reverse order of zone names).

19. Get Log Data [/jl]

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

Return Value:
If using **start** and **end** parameters, the return value is an array of log data recorded in the time range **start** and **end** (both are epoch times). If using the **hist** parameter, it returns the log records of the past n days. 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 units of seconds)
- **end** is the end time (epoch time).
- If the flow sensor is enabled, there will be a 5th element storing the flow rate during the station run.

Parameter **hist** specifies the number of days to go back in history, starting from today. So hist=0 returns the log data of today; hist=1 returns the log data of the today and yesterday, and so on.

When **pid=0**, it indicates a special event, and the second field (sid) will be a string storing the event type. The supported event types are listed below. You can use parameter **type** to specify the specific type to query.

- s1 or s2: Sensor1/sensor2 events
- rd: Rain delay events
- fl: Flow sensor readings
- wl: Watering level/percentage logs

Example Return value: (the request is: 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 [/dl]

/dl?pw=xxx&day=n

Parameters:

- **day**: The day of which the log data is to be deleted. Specifically, it is the epoch time 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 [/cu]

/cu?pw=xxx&jsp=xxx

Parameters:

- **jsp**: New Javascript path

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 [/ja]

/ja?pw=xxx

Returns the aggregated result of /jc, /jo, /jn, /js, /jp in one command, and put each under the "settings", "options", "stations", "status", and "programs" variables respectively.

Example Return:
{"settings":{......},"options":{......},"stations":{......},"status":{......},"programs":{......}}
where …… is identical to the data returned by querying each command individually.

23. Pause Queue [/pq]

/pq?pw=x&dur=xxx&repl=xxx

Parameters:

- **dur:** Triggers (and toggles) a pause with the specified duration (in units of seconds). If there is no active pause, calling it with a non-zero duration will start a pause. If there is an active pause, calling it (regardless of duration value) will cancel the existing pause and resume station runs (i.e. it's a toggle).
- **repl:** This command replaces the current pause (or starts a new pause) with the specified duration (in units of seconds). Using it with a value of 0 will cancel an active pause. If there is no active pause, this command will start a new pause. **Note:** If both **repl** and **dur** appear, **repl** will take over and **dur** will be ignored.

Examples:

- http://os-ip/pq?pw=xxx&dur=600 (pause station runs for 10 minutes, i.e. 600 seconds)
- http://os-ip/pq?pw=xxx&repl=250 (replace the current pause with a new pause of 250 seconds)

24. Debug printout [/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: triggered by buttons	3: reset to AP mode
4: API/timer triggered	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 too many times	10: reboot due to 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 the list of error code in the source code:
https://github.com/OpenSprinkler/OpenSprinkler-Weather/blob/master/errors.ts