OpenSprinkler Firmware 2.1.4 API Document

(May *17, 2015*)

1. Overview

This document describes the OpenSprinkler Firmware 2.1.4 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 automatically MD5 hashed by the application and therefore when using the API commands the password should be MD5 hashed as well**
- 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. Exceptions are the binary-value parameters explained in **Section 5**.
- If a received command does not match any of the listed keywords, it is assumed to be **requesting a file on the microSD card**. For example: http://os-ip/home.js is looking for home.js in the root directory of card. If the file exists, it will be served; otherwise a **Page Not Found** error is returned. Sub-directories are supported, for example: http://os-ip/scripts/home.js
- **Return Values** are all formatted in JSON. For example **{"fwv": 214}, {"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":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. 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.
- **rd**: Rain delay bit (1: rain delay is currently in effect; 0: no rain delay).
- **rs**: Rain sensor status bit (1: rain is detected from rain sensor; 0: no rain detected).
- **rdst:** Rain delay stop time (0: rain delay no in effect; otherwise: the time when rain delay is over).
- **loc**: Location string.
- **wtkey**: Wunderground API key.
- **sunrise**: Today's sunrise time (minutes from midnight).
- **sunset**: Today's sunset time (minutes from midnight).
- **eip**: external IP, calculated as (ip[3]<<24)+(ip[2]<<16)+(ip[1]<<8)+ip[0]
- **lwc**: last weather call/query (epoch time)
- **lswc**: last successful weather call/query (epoch time)
- **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 3-field array that stores the **[pid,rem,start]** of a station, where **pid** is the program index (0 means non), **rem** is the remaining water time (in seconds), **start** is the start time. 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.
- **lrun**: Last run record, which stores the [**station index, program index, duration, end time**] of the last run station.

Example Return:

{ "devt":1413630371, "nbrd":1, "en":1, "rd":0, "rs":1, "rdst":0, "loc":"Boston, MA", "wtkey":"12345678", "sunrise":412, "sunset":1065, "sbits":[64], "ps":[[0,0,0],[6,10800,1413637861],[0,0,0],[7,10800,1413648721],[4,7200,1413659581], [0,0,0],[3,7430,1413573001],[0,0,0]], "lrun":[0,0,0,0]}

3. Change Controller Variables [Keyword /cv]

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

Parameters:

- **rsn**: Reset all stations (i.e. stop all stations immediately, including those waiting to run). Binary value.
- **rbt**: Reboot the controller. Binary value.
- **en**: Operation enable. Binary value.
- **rd**: Set rain delay time (in hours). A value of 0 turns off rain delay.

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)

4. Get Options [Keyword /jo]

/jo?pw=xxx

Return Variables:

- **fwv**: Firmware version (214 means Firmware 2.1.4).
- **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 96.
- **ntp**: Use NTP sync. Binary value.
- **dhcp**: Use DHCP. Binary value.
- **ip{1,2,3,4}**: Static IP (ignored if dhcp=1).
- **gw{1,2,3,4}**: Gateway (router) IP (ignored if dhcp=1).
- **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.
- **ext**: Number of expansion boards (not including the main controller).
- **sdt**: Station delay time (in seconds). Acceptable range is -60 to +60 seconds, in steps of seconds, or -59 to 59 minutes.
- **mas/mas2**: Master stations 1 and 2 (a value of 0 means none). Note that this firmware supports up to 2 master stations.
- **mton/mton2**: Master 1 and 2 on delay time. Acceptable range is 0 to 60.
- **mtof/moof2**: Master off delay time. Acceptable range is -60 to 60.
- **urs**: Use rain sensor. Binary value.
- **rso**: Rain sensor type. Binary value. 0: normally closed; 1: normally open.
- **wl**: Water level (i.e. % Watering). Acceptable range is 0 to 250.
- **den**: Operation enable bit. Binary value.
- **ipas**: Ignore password. Binary value.
- **devid**: Device ID.
- **con/lit/dim**: LCD contrast / backlight / dimming values.
- **rlp**: Relay pulse time (in milli-seconds). Acceptable range is 0 to 2000.
- **uwt**: Weather adjustment method. 0: manual adjustment; 1: Zimmerman method. Water restriction information for California is encoded in the last bit.
- **lg**: Enable logging.
- **dexp/mexp**: Detected/maximum number of zone expansion boards (-1 means cannot auto-detect).

Example Return:

{"fwv":213,"tz":32,"ntp":1,"dhcp":1,"ip1":192,"ip2":168,"ip3":1,"ip4":22,"gw1":192,"gw2":168,"gw3":1,"gw4":66,"hp0":80,"hp1":0,"hwv":22,"ext":0,"sdt":60,"mas":0,"mton":60,"mtof":0,"urs":0,"rso":0,"wl":100,"den":1,"ipas":0,"devid":0,"con":110,"lit":100,"dim":5,"rlp":0,"uwt":0,"ntp1":165,"ntp2":193,"ntp3":126,"ntp4":229, "lg":1,"reset":0, "dexp":0,"mexp":5}

5. Change Options [Keyword /co]

/co?pw=xxx&o?=x&loc=x&wtkey=x&ttt=x

IMPORTANT: parameters displayed in **shaded cells** below are binary, and they will be set to 1 as long as the parameter name appears in the command, regardless of the given value; conversely, they are set to 0 if the name does not appear in the command. For example, **o21=1**, **o21=0**, **o21=on** all set parameter **o21** to 1. Conversely, if the name **o21** does not appear in the command, it will be cleared (set to 0). Keep this in mind!

Also note that **o14, o16, o24** cannot be set using the /co command.

Parameters:

- **loc**: Location string (url encoded).
- **wtkey**: Wunderground API key.
- **ttt**: Set time manually (epoch time, ignored if ntp=1).
- **o?**: ? is the option index. This corresponds to the same list of options in Section 4 (there the options are listed using JSON names). Please refer to that section for detailed explanations and acceptable range of values.

Index	JSON	Meaning	Index	JSON	Meaning	Index	JSON	Meaning
o1	**tz**	Time zone	**o12**	**hp0**	HTTP Port lower byte	**o23**	**wl**	Water level
o2	**ntp**	Use NTP	**o13**	**hp1**	HTTP Port upper byte	**o24**	*(not applicable, use /cv)*	
o3	**dhcp**	Use DHCP	**o14**	*(hardware version, not writtable)*		**o25**	**ipas**	Ignore password
o4	**ip1**	Static IP[0]	**o15**	**ext**	# of expansion boards	**o26**	**devid**	Device ID
o5	**ip2**	Static IP[1]	**o16**	*(removed since firmware 2.1.0)*		**o27**	**con**	LCD contrast
o6	**ip3**	Static IP[2]	**o17**	**sdt**	Station delay time	**o28**	**lit**	LCD backlight
o7	**ip4**	Static IP[3]	**o18**	**mas**	Master station	**o29**	**dim**	LCD dimming
o8	**gw1**	Gateway IP[0]	**o19**	**mton**	Master on delay time	**o30**	**rlp**	Relay pulse time
o9	**gw2**	Gateway IP[1]	**o20**	**mtoff**	Master off delay time	**o31**	**uwt**	Weather method
o10	**gw3**	Gateway IP[2]	**o21**	**urs**	Use rain sensor	**o32**	**ntp1**	NTP IP[0]
o11	**gw4**	Gateway IP[3]	**o22**	**rso**	Rain sensor type	**o33**	**ntp2**	NTP IP[1]
						o34	**ntp3**	NTP IP[2]
						o35	**ntp4**	NTP IP[3]
						o37	**mas2**	Master2
						o38	**mton2**	Master2 on delay
						o39	**mtof2**	Master 2 off delay

Examples:

- http://os-ip/co?pw=xxx&loc=95050 (change location to zip code 95050)
- http://os-ip/co?pw=xxx&wtkey=abc (change Wunderground API key to abc)
- http://os-ip/co?pw=xxx&o12=144&o13=31 (change HTTP port to 8080, i.e. 31*256+144=8080)
- http://os-ip/co?pw=xxx&o17=30 (set station delay time to 30 seconds)
- http://os-ip/co?pw=xxx&o21=on&o22=on (use rain sensor, and rain sensor type is normally open)

(*Keep in mind that binary parameters whose names do not appear in the command will all be set to 0*).

6. Set Password [Keyword/sp]

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

Parameters:

- **npw**: New password.
- **cpw**: Confirmation.

Examples:

- http://os-ip/sp?pw=xxx&npw=sprinkler&cpw=sprinkler (change password to '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.

7. 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 bit field (LSB). For example, 254 means the 1st station on this board does not use master, and all other stations on this board use master.
- **ignore_rain**: Ignore rain bit field. Defined similarly to masop. For example, 192 means the 7th and 8th stations on this board ignore rain, and the other stations on this board do not.
- **stn_dis**: Station disable bit field. Defined similarly to masop.
- **stn_seq**: Station sequential bit field. Defined similarly to masop.
- **rfstn**: RF station bit field. Defined similarly to masop. Each bit indicates whether a station is an RF station.

Example Return:

{"snames":["Master","1. Front Yard","2. Back Yard","3. Left Lawn","4. Right Lawn","5. Flower Bed","Landscape Light","Garage Door"],"masop":[254],"ignore_rain":[0],"masop2":[0],"stn_dis":[0], "rfstn":[0], "stn_seq":[0],"maxlen":16}

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

/cs?pw=xxx&s?=x&m?=x&i?=x&n?=&d?=x&q?=x

Parameters:

- **s?**: Station name. ? is the station index (starting from 0). For example, **s0=abc** assigns name 'abc' to the first station.
- **m?:** Master operation bit field. ? is the board index (starting from 0). Specifically, m0 is the main controller, m1 is the first expansion board, and so on. Refer to the **masop** variable in Section 7 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?**: Ignore rain bit field.
- **n?:** Master 2 operation bit field. Similar to m?, except this is for Master Station 2.
- **d?**: Station disable bit field.
- **q?**: Station sequential bit field.

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)

9. 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}

(the above return shows that the 1st and 2nd stations are currently open).

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

/cm?pw=xxx&sid=x&en=x&t=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**.

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)

An error code will return if you try to open the master station (since the master cannot be operated independently), or open a station that's either already running or in the queue waiting to run.

11. Get Program Data [Keyword /jp]

/jp?pw=xxx

Return Variables:

- **nprogs**: Number of existing programs.
- **nboards**: Number of boards (including main controller).
- **mnp**: Maximum number of programs allowed.
- **mnst**: Maximum number of program start times (fixed time type) allowed.
- **pnsize**: Maximum number of characters allowed in program name.
- **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]

- **flag:** a bit field storing program flags
  - bit 0: program enable (1: enabled; 0: disabled)
  - bit 1: use weather adjustment (1: yes; 0: no)
  - bit 2-3: odd/even restriction (0: none; 1: odd-day restriction; 2: even-day restriction; 3: undefined)
  - bit 4-5: program schedule type (0: weekday; 1: undefined; 2: undefined; 3: interval day)
  - bit 6: start time type (0: repeating type; 1: fixed time type)
  - bit 7: undefined
- **days0/days1:**
  - *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.
  - *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:**
  - **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. The duration value is compressed using the following algorithm:
  0 – 59 seconds, in steps of seconds
  1 to 179 minutes, in steps of minutes
  3 to 16 hours, in steps of hours

  65534 represents sunrise to sunset duration
  65535 represents sunset to sunrise duration
- **name:** Program name

Example Return:
{"nprogs":7, "nboards":1, "mnp":14, "mnst":4, "pnsize":12,
"pd":[[3,127,0,[480,2,240,0],[0,2700,0,2700,0,0,0,0],"Summer"],
[2,9,0,[120,0,300,0],[0,3720,0,0,0,0,0,0],"Fall Prog"],
[67,16,0,[1150,-1,-1,-1],[0,0,0,0,0,0,64800,0],"Pipe"]]}

12. Change Program Data [Keyword /cp]

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

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.
- **v:** Program data structure. The format is the same as explained in **Section 11** above, except the name field, given below.
- **name:** Program name (url encoded, without quotes).

Examples:
- 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, 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").

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

/dp?pw=xxx&pid=x

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)

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

/up?pw=xxx&pid=x

Parameters:

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

15. 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.

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)

16. Get Log Data [Keyword /jl]

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

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), **sid** is the station index (starting from 0), **dur** is the duration (in seconds), **end** is the end time (epoch time). If program index is 0, this is a special event (rain delay or rain sensor), in which case the station index stores the event name ("rd" or "rs").

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], [5,4,1200,1413560461], [5,5,1200,1413561721]]

17. Delete Log Data [Keyword /dl]

/dl?pw=xxx&day=x

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)

18. Change Javascript URL [Keyword /cu]

/cu?pw=xxx&jsp=x

Parameters:

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

Examples:

- http://os-ip/cu?pw=xxx&jsp=http%3A%2F%2Frayshobby.net%2Fscripts%2Fsprinklers%2Fjs
  (change Javascript path to http://rayshobby.net/scripts/sprinkler/js)
- http://os-ip/cu?pw=xxx&jsp=.
  (change Javascript path the local root directory, serving script files from the microSD card)
- http://os-ip/cu?pw=xxx&jsp=./js
  (change Javascript path to /js, serving script files from the /js directory of the microSD card)