

DEIMIC Scripting manual

Content:

1. Javascript functions

2. Deimic functions

2.1. GET FUNCTIONS

2.2. OUTPUTS

2.3. DIMMERS

2.4. ROLLERS

2.5. RECUPERATOR

2.6. CLIMACONVECTOR

2.7. HEATING

2.8. GLOBAL AND DEBUG

2.9. NETWORK TCP AND UDP CONNECTIONS

2.10. TIME

2.11. WEATHER

3. Examples

3.1. Close the roller and turn on the lights if it is too hot outside in a specific days and time

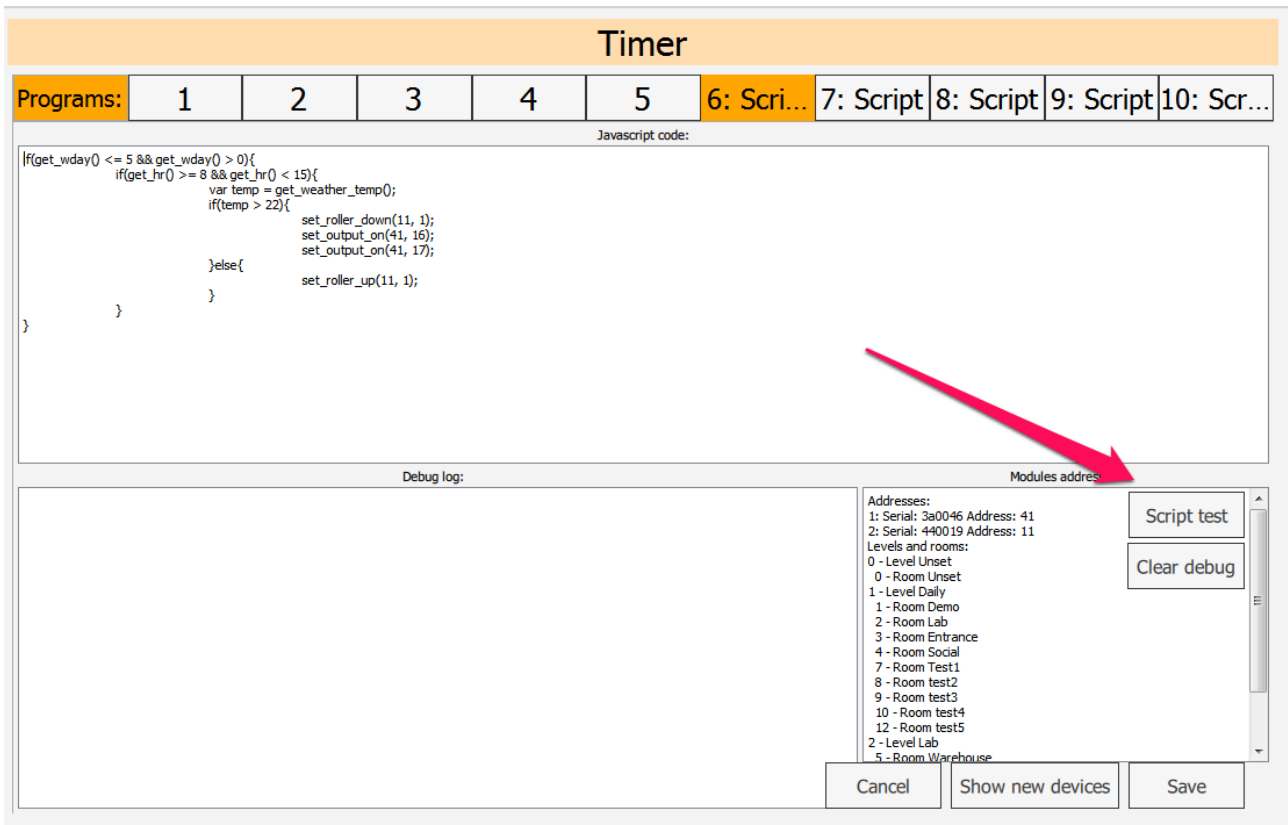
3.2. Send data using UDP

3.3. Send and read data using TCP – DENON Mute On/MuteOff

4. Version and changelog

Deimic API:

Always test your script before setting it(Picture 1.).



Picture 1. Test your script.

Language used in a scripts is Javascript. To declare variable use 'var' word(example 'var a').

Maximum length of a script is 4095.

1. Javascript functions:

Math.abs(value) - absolute of given value

Math.round(value) - returns nearest round of given value

Math.min(a,b) - returns minimum of two given values

Math.max(a,b) - returns maximum of two given values

Math.range(value,min,max) - returns value limited between two given values

Math.sign(value) - returns sign of given value (-1==negative,0=zero,1=positive)

Math.PI() - returns PI value

Math.toDegrees(a) - returns degree value of a given angle in radians

Math.toRadians(a) - returns radians value of a given angle in degrees

Math.sin(a) - returns trig. sine of given angle in radians

Math.asin(a) - returns trig. arcsine of given angle in radians

Math.cos(a) - returns trig. cosine of given angle in radians

Math.acos(a) - returns trig. arccosine of given angle in radians

Math.tan(a) - returns trig. tangent of given angle in radians

Math.atan(a) - returns trig. arctangent of given angle in radians

Math.sinh(a) - returns trig. hyperbolic sine of given angle in radians

Math.asinh(a) - returns trig. hyperbolic arcsine of given angle in radians

Math.cosh(a) - returns trig. hyperbolic cosine of given angle in radians

Math.acosh(a) - returns trig. hyperbolic arccosine of given angle in radians

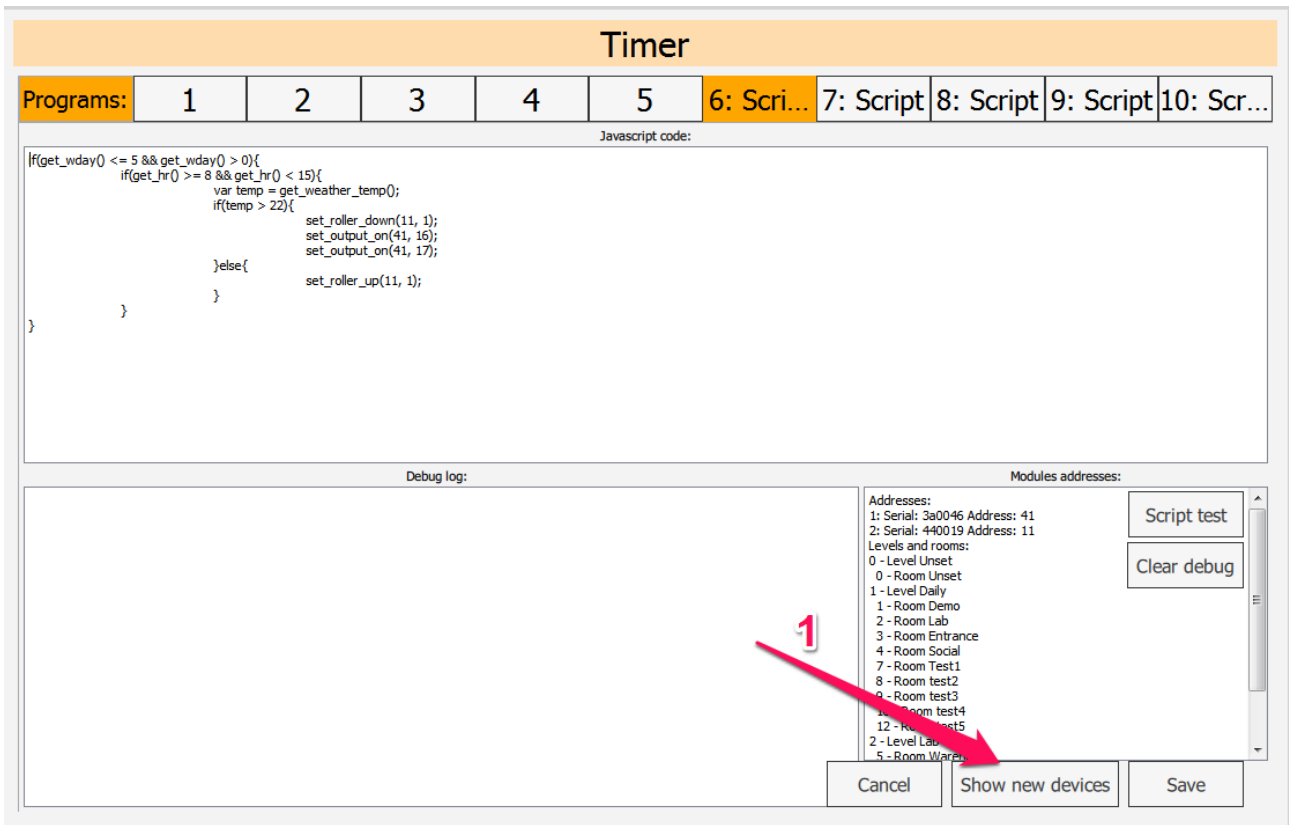
Math.tanh(a) - returns trig. hyperbolic tangent of given angle in radians

Math.atan(a) - returns trig. hyperbolic arctangent of given angle in radians
Math.E() - returns E Neplero value
Math.log(a) - returns natural logaritm (base E) of given value
Math.log10(a) - returns logaritm(base 10) of given value
Math.exp(a) - returns e raised to the power of a given number
Math.pow(a,b) - returns the result of a number raised to a power $(a)^{(b)}$
ath.sqr(a) - returns square of given value
Math.sqrt(a) - returns square root of given value
Math.rand() - returns random value(maximum is INT_MAX)
Math.randInt(min, max) - returns random value between min and max
charToInt(ch) – changes char to int value
String.indexOf(search) - return index of search in a string, returns -1 if not found
String.substring(lo,hi) – creates substring lo is start, hi is stop
String.charAt(pos) – gets char at pos
String.charCodeAt(pos) – gets char code at pos
String.fromCharCode(char) – get string from char code
String.split(separator) – splits string in array of strings
Integer.parseInt(str) – parses int
Integer.valueOf(str) – tries to convert string into int
Array.contains(object) – checks if array contains object
Array.remove(object) – removes object from array
Array.join(separator) – creates string from array with seperator

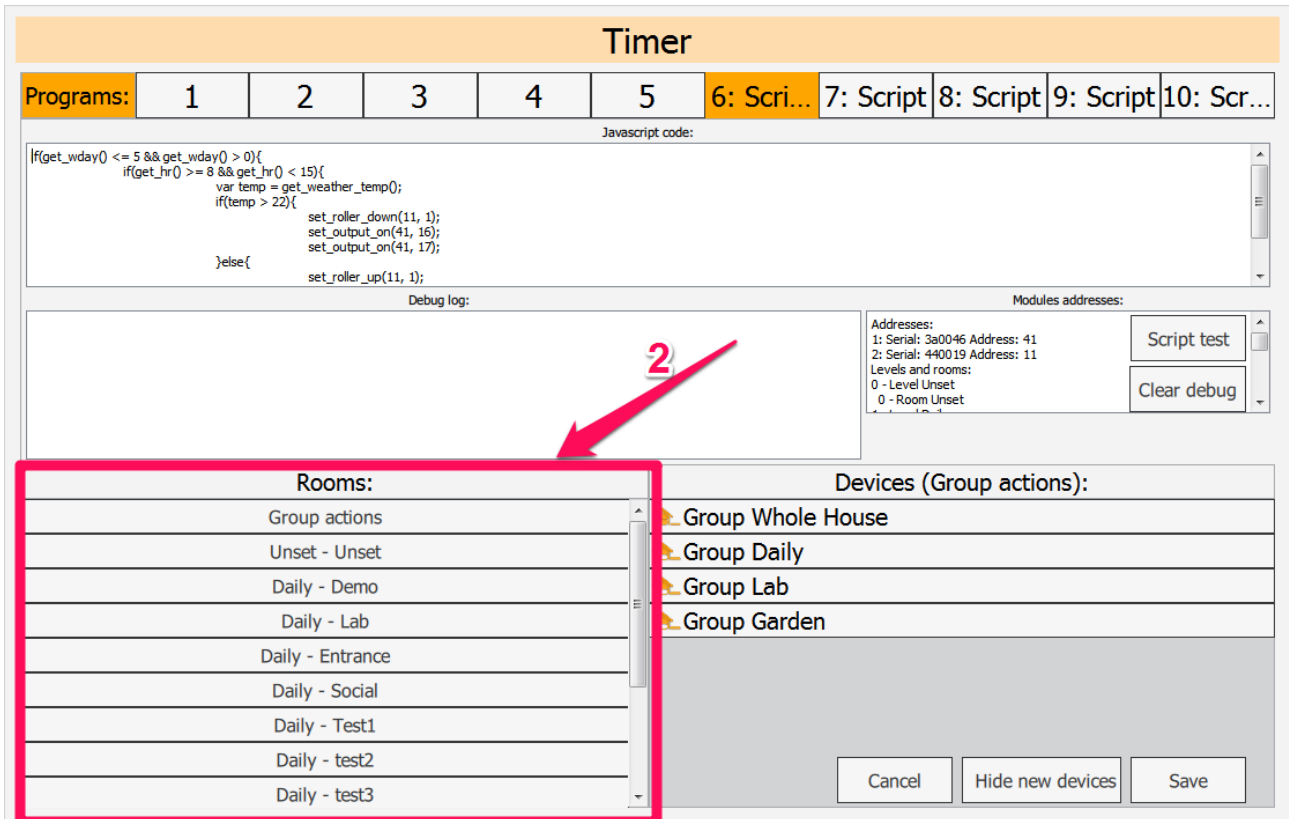
2. Deimic functions:

IMPORTANT:

To get output/roller/dimmer/recuperator/clicmaconvector address and number just click show new devices(1) at the bottom of the screen, select the room(2) and click on desired device(3) - it automatically writes address and number in javascript code



Picture 2. Show new devices.



Picture 3. Select room.

Timer

Programs:	1	2	3	4	5	6: Scri...	7: Script	8: Script	9: Script	10: Scr...
-----------	---	---	---	---	---	------------	-----------	-----------	-----------	------------

Javascript code:

```

if(get_wday() <= 5 && get_wday() > 0){
  if(get_hr() >= 8 && get_hr() < 15){
    var temp = get_weather_temp();
    if(temp > 22){
      set_roller_down(11, 1);
      set_output_on(41, 16);
      set_output_on(41, 17);
    }else{
      set_roller_up(11, 1);
    }
  }
}

```

Debug log:

Modules addresses:

Addresses:

1: Serial: 3a0046 Address: 41

2: Serial: 440019 Address: 11

Levels and rooms:

0 - Level Unset

0 - Room Unset

Rooms:

Group actions
Unset - Unset
Daily - Demo
Daily - Lab
Daily - Entrance
Daily - Social
Daily - Test1
Daily - test2
Daily - test3

Devices (Daily - Demo):

ON OFF	Window II
ON OFF	Table
ON OFF	Couch
ON OFF	Output
ON OFF	Aquarium
ON OFF	UV
ON OFF	LED
ON OFF	TV
ON OFF	Pump

Picture 4. Select device.

To get input address and number just check the address connected to serials on the right of the screen (picture 5), you can check input number in module configuration.

Timer

Programs:	1	2	3	4	5	6: Scri...	7: Script	8: Script	9: Script	10: Scr...
-----------	---	---	---	---	---	------------	-----------	-----------	-----------	------------

Javascript code:

```

if(get_wday() <= 5 && get_wday() > 0){
  if(get_hr() >= 8 && get_hr() < 15){
    var temp = get_weather_temp();
    if(temp > 22){
      set_roller_down(11, 1);
      set_output_on(41, 16);
      set_output_on(41, 17);
    }else{
      set_roller_up(11, 1);
    }
  }
}

```

Debug log:

Modules addresses:

Addresses:

1: Serial: 3a0046 Address: 41

2: Serial: 440019 Address: 11

Levels and rooms:

0 - Level Unset

0 - Room Unset

Rooms:

Group actions
Unset - Unset
Daily - Demo
Daily - Lab
Daily - Entrance
Daily - Social
Daily - Test1
Daily - test2
Daily - test3

Devices (Daily - Demo):

ON OFF	Window II
ON OFF	Table
ON OFF	Couch
ON OFF	Output
ON OFF	Aquarium
ON OFF	UV
ON OFF	LED
ON OFF	TV
ON OFF	Pump

Picture 5. Address of a modules.

If you got DEIMIC ONE Green module, then inputs are group in two modules(addresses 524 and 525), first 32 inputs are in 524 module and rest 23 inputs are in 525 module. So input number 20 is 524,20 and input number 32 is 525,0, input number 40 is 525,8

sleep(seconds) - sleeps the thread for a number of seconds

msleep(msecons) - sleeps the thread for a number of milliseconds

get_serial() - gets system serial

2.1. GET FUNCTIONS:

get_input(address, number) - gets state of an input(0/1 short click, 2 long click) from a module address and input number - all known addresses are shown on the right side of the screen
returns -1 if device is not available

get_output(address, number) - gets state of an output from a module address and output number
returns -1 if device is not available

get_dimmer(address, number) - gets state of a dimmer(value between 0% and 100%) from a module address and dimmer number
returns -1 if device is not available

get_roller(address, numer) - get state of a roller from a module address and roller number
return -1 if device is not available
return 0 - state unknown/stop
return 1 - state stop
return 2 - state run up
return 3 - state run down
return 4 - state up
return 5 - state down
return 6 - state wait(it is the state when roller is running up/down and we click to run the other direction and it stops for a second)

2.2. OUTPUTS:

set_output_on(address, number) - sets on selected output, returns void

set_output_off(address, number) - sets off selected output, returns void

set_output_toggle(address, number) - toggles the output(changes the state between on and off), returns void

set_output_set(address, number, time1, time2, count) - blinks the output for a count number, time1 is on time, time2 is off time, if you want to turn on the light for a specific time set time2 and count to 0

set_group(group, level, room, subgroup, value) - sets value to a group of devices(outputs and dimmers) - group of devices we want to change coded in bits, levels coded in bits, rooms coded in bits, subgroup coded in bits, value(from 0 - turn off to 100 - dimmer maximum value)

Available groups:

- 1 - Lights
- 2 - Sockets/Radio/etc.
- 3 - Gates
- 4 - Outside lights
- 5 - LEDs
- 6 - Technical

So to send to a specific group we need to set correct bits. We want to send to lights group so we set 1st bit ($1 \ll 1$) which is 0b00000010

We can also combine groups together example lights, outside lights and leds ($1 \ll 1$) | ($1 \ll 4$) | ($1 \ll 5$) which is 0b00110010

Subgroup we need to set to 0xFF

We can see level codes on the right of the screen, same for rooms.

So to set for a specific level just move the bit with ($1 \ll \text{LEVEL_ID}$) and room id to 0xFFFFFFFF(0 bit is for unused level and room, we do not want to set unused devices)

For setting only one room we need to find level and room ids and set them correctly level = ($1 \ll \text{LEVEL_ID}$) and room = ($1 \ll \text{ROOM_ID}$)

2.3. DIMMERS:

set_dimmer_set(address, number, value, time) - sets dimmer to a specific value(0-100%) in a specified time(1 time means 0.01s, so to set it to 1s time should be set to 100), it is time to reach the value, it changes the value smoothly

set_dimmer_toggle(address, number, value, time) - toggle dimmer between value(1 - 100%) and 0% in a specified time(1 time means 0.01s, so to set it to 1s time should be set to 100), it is time to reach the value, it changes the value smoothly,

set_dimmer_wavergb(address, number, time) - starts/stop rgb wave, time 1 is equal to 760ms

set_dimmer_wave(address, number, time) - starts wave for 1 dimmer, time 1 is equal to 230ms

set_dimmer_set3x(address, value_red, time_red, value_green, time_green, value_blue, time_blue) - sets the value of a RGB, value and times are set in the same way as in set_dimmer_set command

2.4. ROLLERS:

set_roller_upstop(address, number) - run roller up or stops it if running

set_roller_downstop(address, number) - run roller down or stops it if running

set_roller_up(address, number) - run roller up

set_roller_down(address, number) - run roller down

set_roller_stop(address, number) - stops the roller

set_roller_step(address, number) - changes the state of a roller in order(run up, stop, run down, stop)

set_roller_groupup(group,level,room,subgroup) - sets group of rollers to run up, levels and rooms are the same as in set_group command, available groups and subgroups are shown below
set_roller_groupstop command

set_roller_groupdown(group,level,room,subgroup) - sets group of rollers to run down, levels and rooms are the same as in set_group command, group and subgroups are shown below
set_roller_groupstop command

set_roller_groupstop(group, level, room, subgroup) - set group of rollers to stop, levels and rooms are the same as in set_group command

Available groups:

- 1 - Roller
- 2 - Shutter facade
- 3 - Unused
- 4 - Marquise
- 5 - Projector

Available subgroup(cardinal direction):

- 0 - N
- 1 - NE
- 2 - E
- 3 - SE
- 4 - S
- 5 - SW
- 6 - W
- 7 - NW

Set group and subgroup in the same way as in set_group command.

Example:

We want to change state of a rollers, shutter facades and marquises and subgroup set to N, NE or NW. So group value would be 0b00010110
and subgroup 0b10000011

2.5. RECUPERATOR:

set_recuperatorset(address, number, value) - set recuperator gear to value

set_recuperatorstop(address, number) - stop recuperator

2.6. CLIMACONVECOTR:

set_climaon(address, number) - turns on climaconvector to previous mode

setclimaoff(address, number) - turns off climaconvector

2.7. HEATING:

setheatprofile(address, number, value) - set heat profile, value can be in range:

0 - OFF

1 - ECO

2 - COMFORT

3 - MAX

get_temperature(address, number) - gets temperature of heating section

2.8. GLOBAL AND DEBUG:

set_global(number, value) - sets global variable(number is the name of variable) to value, it can be used for example to block the script from running twice, at first line of script check global and after that set it to 1, so if next script starts it stops.

Remember this variable are set to 0 after every reboot of the system, so they are no good for saving some data

get_global(number) - gets global variable(number is the name of variable)

set_global_file(number, value) - sets global variable(number is the name of variable), the difference is that this variables are saved in a file, so after reboot values are restored

get_global_file(number) - gets global variable(number is the name of variable)

printstr(str) - prints string value to debug console

printnum(num) - prints int value to debug console

printfloat(float) - prints float value to debug console

2.9. NETWORK TCP AND UDP CONNECTIONS

send_tcp(address, port, data) - sends string data to a tcp server, address of a host, port of a host, data(string) to send, returns number of bytes written

receive_tcp(address, port, length) - connects to tcp, waits for data(length), address of a host, port of a host

send_receive_tcp_time(address, port, data, time) – connects to server using tcp, sends data, wait time and returns data received(time in 100ms)

send_receive_tcp_http_get(address, data, length) - connects to http server and reads length of data, address of a host, data to send to host, GET method arguments

send_tcp_http_get(address) - connects to http server, used when read is not needed

send_udp(address, port, data) - sends string data to udp server, address of a server, port of a server, data(string) to send

2.10. TIME

get_light() - gets value if it is day or night(1 - day, 0 - night)

get_time_t() - gets current time_t - seconds since 1 Jan 1970

get_min() - gets current number of minutes

get_hr() - gets current hour

get_wday() - gets weekday (0 - Sunday, 1 - Monday, 2 - Tuesday, 3 - Wednesday, 4 - Thursday, 5 - Friday, 6 - Saturday)

get_mday() - gets day of a month

get_yday() - gets day of a year

get_year() - gets current year number

get_time() - gets time a double form(example 1:30 PM is 13,5 - hour 13 and 0,5 as 30 minutes)

get_sunrise_time() - gets sunrise time in double form same as in get_time() command

get_sunset_time() - gets sunset time in double form same as in get_time() command

2.11. WEATHER

get_weather_temp() - gets current weather temperature in double in celcius

get_weather_temp_min() - gets todays weather minimum temperature

get_weather_temp_max() - gets today's weather maximum temperature

get_weather_wind_speed() - gets current wind speed in m/s

get_weather_wind_dir() - A numerical value representing the direction that the wind is coming from in degrees, with true north at 0° and progressing clockwise

get_weather_raining() - binary value, 1 - currently raining, 0 - currently no rain

get_weather_cloud() - A numerical value between 0 and 1 (inclusive) representing the percentage of sky occluded by clouds.

get_weather_pressure() - A numerical value representing the sea-level air pressure in hPa

get_weather_humidity() - A numerical value between 0 and 1 (inclusive) representing the relative humidity

3. Examples

It is good to write some print functions just to be sure if the script works. After the script is finished you should remove these prints so they will not be sent next time.

3.1. Close the roller and turn on the lights if it is too hot outside in a specific days and time

```
//This script should be set in a timer
```

```
//(11, 1) - is the roller address
```

```
//(41, 16) - is light address
```

```
if(get_wday() <= 5 && get_wday() > 0){ //If it is weekday
```

```
    if(get_hr() >= 8 && get_hr() < 16){ //Between 8 AM and 4 PM
```

```
        var temp = get_weather_temp(); //Read current temperature outside
```

```

        if(temp > 22){ //If temperature is more than 22 celcius degrees
            set_roller_down(11, 1); //Run down the roller
            set_output_on(41, 16); //turn on the light
        }else{ //If temperature goes down then open the roller
            set_roller_up(11, 1);
        }
    }
}

```

3.2. Send data using UDP

Command below sends data to IR transmitter.

```

    send_udp("192.168.50.103", 7777, "@IR_SEND@100;100;100;" +
String.fromCharCode(13, 10));
//String.fromCharCode(13, 10) is carriage return(\r - code 13) and new line(\n - code 10)

```

3.3. Send and read data using TCP – DENON Mute On/MuteOff

```
send_receive_tcp_time
```

```
//Sends data and waits for 100ms(last parameter) and reads data
```

```
var string = send_receive_tcp_time("192.168.50.201", 23, "MU?" + String.fromCharCode(13), 1);
```

```
msleep(100); //Need to wait to create new tcp connection
```

```
if(string.indexOf("MUOFF") >= 0) //Check if string contains "MUOFF"
```

```
send_tcp("192.168.50.201", 23, "MUON" + String.fromCharCode(13)); //If contains than turn mute
on
```

```
else
```

```
send_tcp("192.168.50.201", 23, "MUOFF" + String.fromCharCode(13)); //Else turn mute off
```

4. Version and changelog

1.0 - Initial version