```
% Watering-sonoff ver 1.3, GNU LGPL license, (C)2018 m.sillano
% telnet: script http://192.168.43.242:85/www/sonoff/ws0103.eub

% you can change the topic and the ID in next line
config @6 /msillanohome/hidro/1

on init
do
% @8 is the run counter
% test for first run: $tmp = 0 also if @8 = ''
  setvar $tmp = (@8 * 1)
  if $tmp = 0 then
% cool start, only for first run:
% @1: cycle timer count down (min: 0 off)
    setvar @1 = 0
% @2: cycle duration (min)
    setvar @2 = 2
% @3: time T1
    setvar @3 = "01:00"
% @4: time T2
    setvar @4 = "02:00"
% @5: mode (0..4)
    setvar @5 = 2
% @7: day counter (for mode 1)
    setvar @7 = 0
    endif
% hot start
% increments run counter
  setvar @8 = ($tmp + 1)
% sets topics
  setvar $cmt = @6 | "/command"
  setvar $set = @6 | "/set"
  subscribe local $cmt
  subscribe local $set
% set moisure status in var $ms
  gpio_pinmode 14 input
  setvar $ms = gpio_in(14)
% publish status/moisure.
  setvar $tmp = @6 | "/status/moisure"
  publish local $tmp "{\"Time\":\"" | $timestamp | "\",\"moisure\":" | $ms | "}"
retained
% relay restore status, in var $rs
% restore from retained message...
  setvar $tmp = @6 | "/status/watering"
  setvar $rs = json_parse("watering",retained_topic($tmp))
  if not($rs = 1) then
% default is 0
    setvar $rs = 0
    endif
% or from cycle timer count down
  if @1 > 0 then
    setvar $rs = 1
% restart timer 1 (1 min)
    settimer 1 60000
    endif
% like a call: this publish status/watering and  status/moisure
publish local $cmt "go"
% sets alarms
  setalarm 3 @3
  setalarm 4 @4

on mqttconnect
do
% subscrive from new client all commands
```

```
  subscribe remote $cmt
  subscribe remote $set
% re-publish status to update new client
  publish local $set "{\"data\":\"send\"}"
  publish local $cmt "send"

on topic remote $set
do
% debug echo on console
  println "on Remote Set " | $this_data
% re-publish local for execution
  publish local $set $this_data

on topic remote $cmt
do
% debug echo on console
  println "on Remote Command " | $this_data
% re-publish local for execution
  publish local $cmt $this_data

on topic local $set
do
% debug echo on console
  println "on Local Set " | $this_data
% decodes the 'set' data, puts it in $tmp
  setvar $tmp = json_parse("data",$this_data)
  if $tmp = "long" then
% sets cycle duration (min)
    setvar @2 = json_parse("value",$this_data) | "   "
    endif
  if $tmp = "when1" then
% sets time T1...
    setvar @3 = json_parse("value",$this_data)
% and update alarm 3
    setalarm 3 @3
    endif
  if $tmp = "when2" then
% sets time T1...
    setvar @4 = json_parse("value",$this_data)
% and update alarm 4
    setalarm 4 @4
    endif
  if $tmp = "mode" then
% sets mode (0..4)
    setvar @5 = json_parse("value",$this_data)
    endif
  if $tmp = "reset" then
% does reset
    system reset
    endif
  if $tmp = "timezone" then
% sets timezone (see https://github.com/martin-ger/esp_mqtt/issues/36 )
    system "set ntp_timezone " | json_parse("value",$this_data)
    endif
% implicit 'send'
% in any case, after a 'set', publish status/config
  setvar $tmp = @6 | "/status/config"
  publish local $tmp "{\"run\":" | @8 | ",\"long\":" | @2 | ",\"tx\":[\"" | @3 |
"\",\"" | @4 | "\"],\"mode\":" | @5 | "}" retained

on topic local $cmt
do
% debug echo on console
println "on Local Command " | $this_data
```

```
% decodes the value
  if $this_data = "on" then
% sets ON
    setvar $rs = 1
    endif
  if $this_data = "off" then
% sets 0 cycle timer count down
    setvar @1 = 0
% sets OFF
    setvar $rs = 0
    endif
  if $this_data = "toggle" then
% sets 0 cycle timer count down
    setvar @1 = 0
% inverts relay status
    setvar $rs = not($rs)
    endif
% cycle starts only if moisure dry ($ms=1)
  if (($this_data = "cycle")*($ms = 1)) > 0 then
% sets ON
    setvar $rs = 1
% initialize cycle timer count down
    setvar @1 = (@2 + 1)
% start timer (60 sec)
    settimer 1 60000
    endif
  setvar $tmp = @6 | "/status/watering"
  if $this_data = "send" then
% only re-send status/*
    publish local $tmp  retained_topic($tmp)
    setvar $tmp = @6 | "/status/moisure"
    publish local $tmp  retained_topic($tmp)
  else
% implicit 'go': set output and publish
    if (($rs = 1)*(@5 = 4)) > 0 then
% mode 4, ON. Like $ms
      gpio_out 12 $ms
      gpio_out 13 not($ms)
    else
% default
      gpio_out 12 $rs
      gpio_out 13 not ($rs)
    endif
% publish the watering status
    publish local $tmp "{\"Time\":\"" | $timestamp | "\",\"watering\":" | $rs |
"}" retained
    endif

% Sonoff-basic button pressed:
on gpio_interrupt 0 pullup
do
% debug echo on console
  println "New state GPIO 0: " | $this_gpio
  if $this_gpio = 0 then
% toggle action
    publish local $cmt "toggle"
    endif

% status change of moisure probe
on gpio_interrupt 14 nopullup
do
% debug echo on console
  println "New state GPIO 14: " | $this_gpio
% real change?
```

```
  if $ms = not($this_gpio) then
    setvar $ms = $this_gpio
% publish the moisure status
    setvar $tmp = @6 | "/status/moisure"
    publish local $tmp "{\"Time\":\"" | $timestamp | "\",\"moisure\":" | $ms |
"}" retained
    if (($rs = 1)*(@5 = 4)) > 0 then
% relay ON/OFF if mode 4 and $rs=1
      gpio_out 12 $ms
      gpio_out 13 not($ms)
      endif
    endif

% timer 1 min
on timer 1
do
% debug echo on console
  println "on int timer1 " | @1
  if @1 > 0 then
% down cycle timer count
    setvar @1 = (@1 - 1)
% restart timer
    settimer 1 60000
    endif
  if @1 = 1 then
% end count, sets OFF
    publish local $cmt "off"
    endif

% time T1 alarm
on alarm 3
do
% debug echo on console
  println "on intT1"
  setvar @7 = not(@7*1)
  if ((@5 = 1)*(@7 = 1)) > 0 then
% mode 1, if @7, starts cycle
    publish local $cmt "cycle"
    endif
  if @5 > 1 then
    if @5 = 4 then
% mode 4, sets ON
      publish local $cmt "on"
    else
% mode 2, 3: starts cycle
      publish local $cmt "cycle"
      endif
    endif

%time T2 alarm
on alarm 4
do
% debug echo on console
  println "on intT2"
  if @5 = 4 then
% mode 4: OFF
    publish local $cmt "off"
    endif
  if @5 = 3 then
% mode 3: starts cycle
    publish local $cmt "cycle"
    endif
```