

# Domoticz MicroPython Projects - Block Diagrams - <Have Fun/>

Addendum to the [Domoticz Homeautomation Workbook](#).

By rwbl

# Introduction

## **Purpose**

To explore how to use the MicroPython programming language running on embedded hardware, the Microcontroller Unit MCU, interfacing with the Domoticz Home Automation System.

The core of the projects uses the Raspberry Pi Pico W, with actuators & sensors, acting as a Web Server to communicate with the Domoticz Home Automation System.

The projects can function as a base for projects or to trigger ideas for use.

The intention is to provide some practical guidance and not to explain Domoticz nor programming languages.

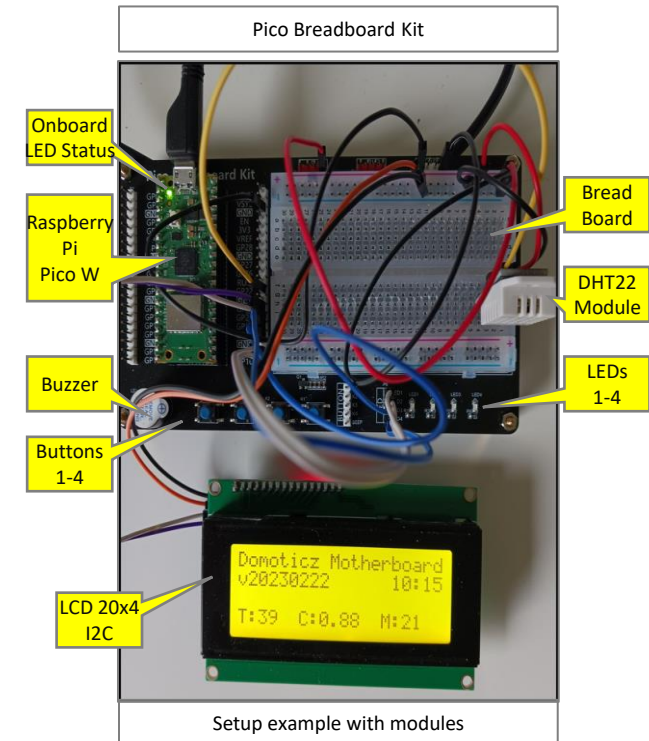
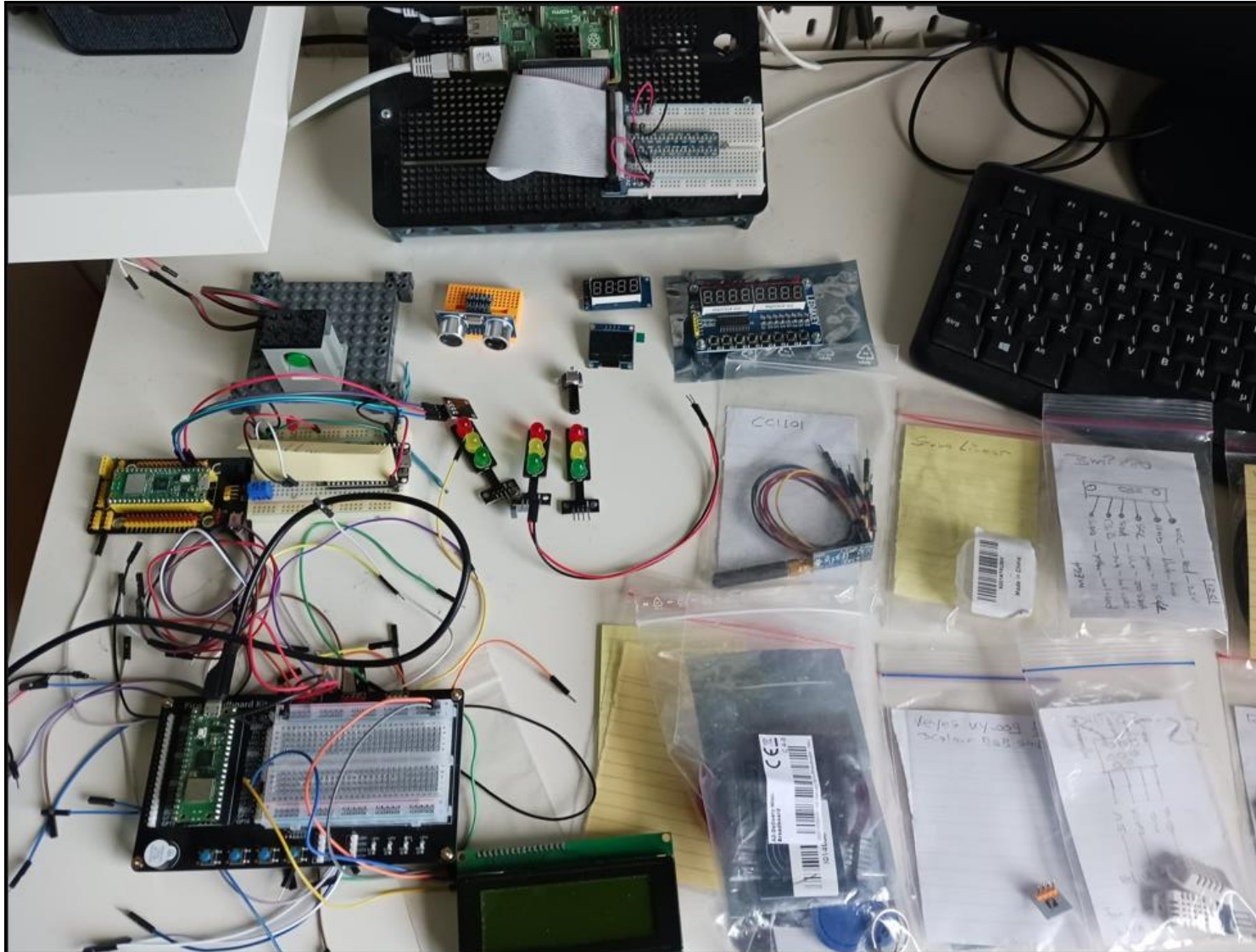
## **Prerequisites**

Basic knowledge of

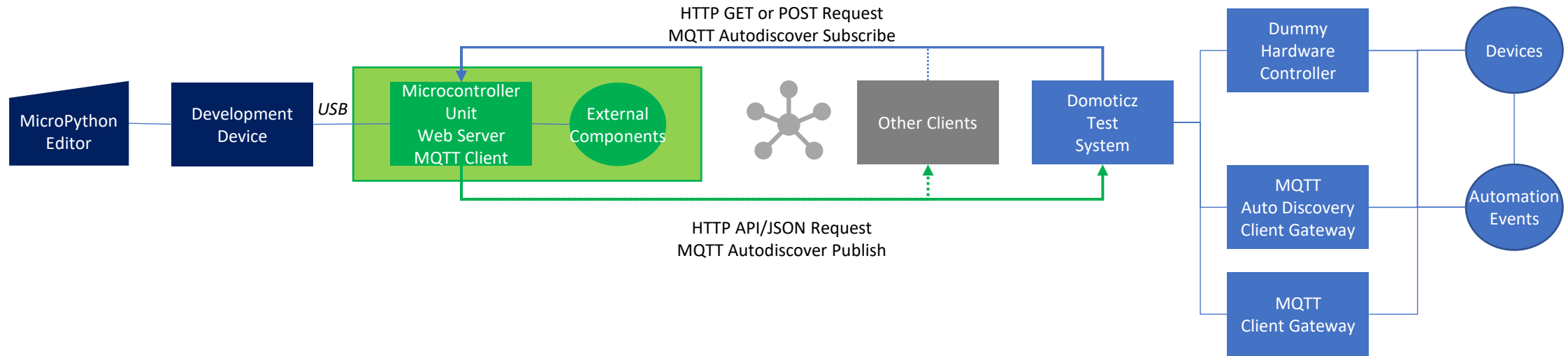
- Domoticz Home Automation System.
- Domoticz Automation Event system dzVents & Lua.
- Programming languages Python and MicroPython.
- Raspberry Pi Pico / Pico W and ESP microcontrollers.
- Thonny Integrated Development Environment.
- JavaScript Object Notation (JSON).
- Message Queuing Telemetry Transport MQTT and MQTT Autodiscover.

# Workbench

## *Overview of the Author's workbench*

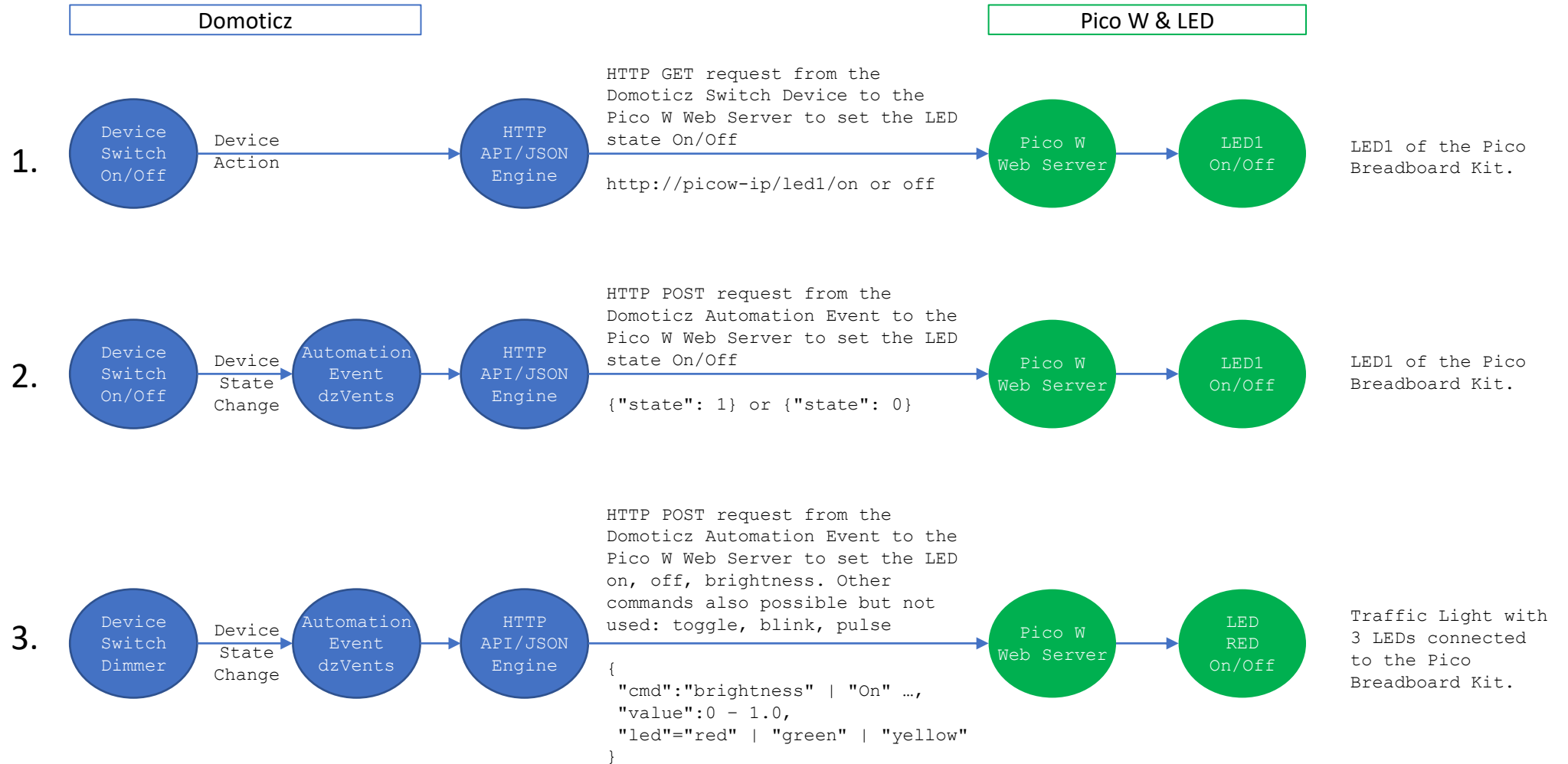


# Blockdiagram Setup



# Project LED Control

*Domoticz Switch Device triggers Device On/Off/Brightness Action to set the Pico W LED1 of the Pico Breadboard Kit.*



# Project LED Control – Device On/Off Action

*Domoticz Switch Device triggers Device On/Off Action to set the Pico W LED1 of the Pico Breadboard Kit On/Off.*

## Domoticz Hardware Dummy

Idx	Name	Enabled	Type	Address	Port	Data Timeout
3	VirtualSensors	Yes	Dummy (Does nothing, use for virtual switches only) <a href="#">Create Virtual Sensors</a>			Disabled

Create Virtual Sensor

Name: LED1 Control

Sensor Type: Switch

OK Cancel

## Device created

Idx	Hardware	ID	Unit	Name	Type	SubType	Data
16	VirtualSensors	00014060	1	LED1 Control	Light/Switch	Switch	On

## Device Widget

LED1 Control On


Last Seen: 2023-02-23 19:12:58  
Type: Light/Switch, Switch, On/Off

Log Edit Timers Notifications

Idx: 16

Name: LED1 Control

Switch Type: On/Off

Switch Icon:  Default icon

On Delay: 0 (Seconds) 0 = Disabled

Off Delay: 0 (Seconds) 0 = Disabled

On Action: <http://picow-ip/led1/on>

Off Action: <http://picow-ip/led1/off>

Protected: ☐

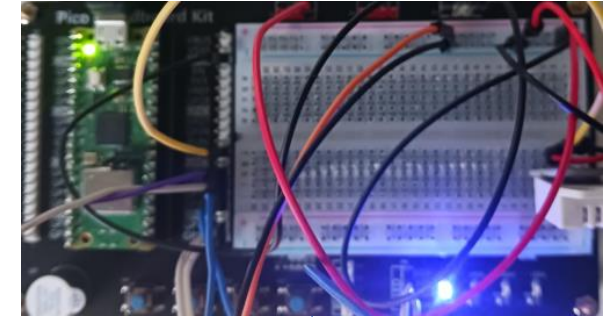
Description: Control the Raspberry Pi Pico W LED1 via HTTP GET request to the Pico W running as a Webserver (MicroPython).

Edit Switch Device Properties with HTTP GET request actions set to switch the Pico W LED1 On / Off

On/Off Action Command:  
<http://picow-ip/led1/on> or <http://picow-ip/led1/off>

Device HTTP On / Off Actions to Pico W Web Server

## Pico W with Breadboard Kit & LED1 On



## Thonny - Pico W Web Server MicroPython Script & Log

```
Thonny - C:\Users\user\Documents\Domoticz\domoticz-microcontroller-workbook\Chapter1\Raspberry Pi Pico W\Projects\LEDControl\Source...
File Edit View Run Tools Help
New Ctrl-N
This computer
C:\Users\user\Documents\Domoticz\domoticz-microcontroller-workbook\Chapter1\Raspberry Pi Pico W\Projects\LEDControl\Source...
config.py
ledcontrol.dvts
ledcontrol.py
Raspberry Pi Pico
lib
config.py
Shell
>>> %Run -c $EDITOR_CONTENT
LEDControl v20230220
Network - Connected
Network - IP = picow-ip
Network Client connected from picow-ip
HTTP Command: /led1/~#
HTTP Response: {"status": "OK", "title": "/led1/off", "message": "Off"}
Network Client connected from picow-ip
HTTP Command: /led1/on
HTTP Response: {"status": "OK", "title": "/led1/on", "message": "On"}
MicroPython (Raspberry Pi Pico) - COM11
```



# Project LED Control - Custom Event

*Domoticz Switch Device State Change triggers Automation Event to set the Pico W LED1 of the Pico Breadboard Kit On/Off.*

## Domoticz Hardware

Idx	Name	Enabled	Type	Address	Port	Data Timeout
3	VirtualSensors	Yes	Dummy (Does nothing, use for virtual switches only) <a href="#">Create Virtual Sensors</a>			Disabled

Create Virtual Sensor

Name: LED1 Control

Sensor Type: Switch

OK

Cancel

## Domoticz Devices

Idx	Hardware	ID	Unit	Name	Type	SubType	Data
16	VirtualSensors	00014060	1	LED1 Control	Light/Switch	Switch	On

## Device Widget

LED1 Control

On

Last Seen: 2023-02-23 19:12:58

Type: Light/Switch, Switch, On/Off

Log

Edit

Timers

Notifications

```
#!/usr/bin/env python3
# LED1 Control
# This script is used to control the LED1 of the Pico W Breadboard Kit via Domoticz.
# It uses the Domoticz API to send commands to the LED1.

import requests
import json
import sys
import logging

# Domoticz API URL
DOMOTICZ_URL = "http://192.168.1.100:8080"

# Domoticz API Token
DOMOTICZ_TOKEN = "YOUR_TOKEN_HERE"

# LED1 ID
LED1_ID = 16

# LED1 Name
LED1_NAME = "LED1 Control"

# LED1 Type
LED1_TYPE = "Light/Switch"

# LED1 SubType
LED1_SUBTYPE = "Switch"

# LED1 Data
LED1_DATA = "On"

# LED1 State
LED1_STATE = "On"

# LED1 Status
LED1_STATUS = "OK"

# LED1 Title
LED1_TITLE = "/led1/on"

# LED1 Message
LED1_MESSAGE = "On"

# LED1 Command
LED1_COMMAND = "/led1/on"

# LED1 Response
LED1_RESPONSE = {"status": "OK", "title": "/led1/on", "message": "On"}

# LED1 Log
LED1_LOG = "LED1 Control"

# LED1 Logger
LED1_LOGGER = logging.getLogger(LED1_LOG)

# LED1 Main Function
def led1_main():
    # LED1 Log
    LED1_LOGGER.info("LED1 Control")

    # LED1 State
    LED1_STATE = "On"

    # LED1 Status
    LED1_STATUS = "OK"

    # LED1 Title
    LED1_TITLE = "/led1/on"

    # LED1 Message
    LED1_MESSAGE = "On"

    # LED1 Command
    LED1_COMMAND = "/led1/on"

    # LED1 Response
    LED1_RESPONSE = {"status": "OK", "title": "/led1/on", "message": "On"}

    # LED1 Log
    LED1_LOGGER.info("LED1 Control")

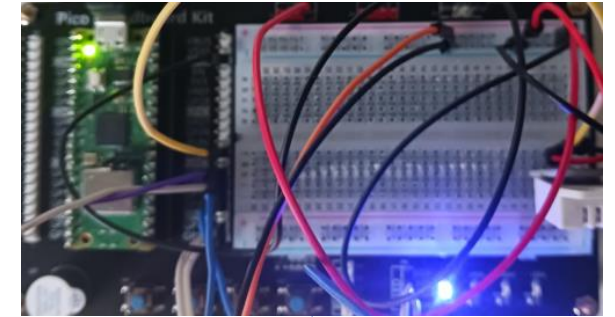
# LED1 Main
led1_main()
```

Domoticz Automation Event dzVents triggered by state change of the Domoticz Switch device (IDX=16).

The event submits a HTTP GET request to the Pico W Web Server to switch the LED1 On/Off.

Device HTTP On / Off Actions to the Pico W Web Server

## Pico W with Breadboard Kit & LED1 On



## Thonny - Pico W Web Server MicroPython Script & Log

```
File Edit View Run Tools Help
C:\Data\projects\homeautomation\domoticz\microcontroller-workbook\Chapter1\Raspberry Pi Pico W\Projects\LEDControlSource...
New Ctrl+N
This computer
C:\Data\projects\homeautomation\domoticz\microcontroller-workbook\Chapter1\Raspberry Pi Pico W\Projects\LEDControlSource...
config.py
ledcontrol.py
Raspberry Pi Pico
config.py

91 # Return Command, i.e. /led1/on
92 def getCommand(request):
93     cmd = "UNKNOWN"
94     # Split the decoded request string into a list
95     data = str(request.decode()).split(CRLF)
96     # Check if there is data to get the first item
97     if len(data) > 0:
98         # print(data[0])
99         # Split the command string into 3 list with 3 items
100         cmd = data[0].split(SPACE)
101         # Check and get the 2nd item
102         if len(cmd) == 3:
103             cmd = cmd[1]
104         # Return the command, i.e. /led1/on etc.
105         print(f'HTTP Command: {cmd}')
106     return cmd
107
108 # Main
109 print(f'({NAME}) ({VERSION})')
110
111 # Connect to the network and listen for incoming connections
112 server = socket.socket()
113 server.listen(1)

Shell
>>> !Run -c $EDITOR_CONTENT
LEDControl v20230220
Network - Connected
Network - IP = 192.168.1.100
Network Client connected from ipicow-ip
HTTP Command: /led1/off
HTTP Response: {"status": "OK", "title": "/led1/off", "message": "Off"}
Network Client connected from ipicow-ip
HTTP Command: /led1/on
HTTP Response: {"status": "OK", "title": "/led1/on", "message": "On"}
MicroPython (Raspberry Pi Pico) - COM11
```

# Project DHT22 - HTTP API/JSON

*Pico W samples DHT22 sensor data and triggers updating the Domoticz Temp + Humidity device.*

## Domoticz Hardware

Idx	Name	Enabled	Type	Address	Port	Data Timeout
3	VirtualSensors	Yes	Dummy (Does nothing, use for virtual switches only) <a href="#">Create Virtual Sensors</a>			Disabled

**Create Virtual Sensor**


Name:

Sensor Type:

## Domoticz Devices

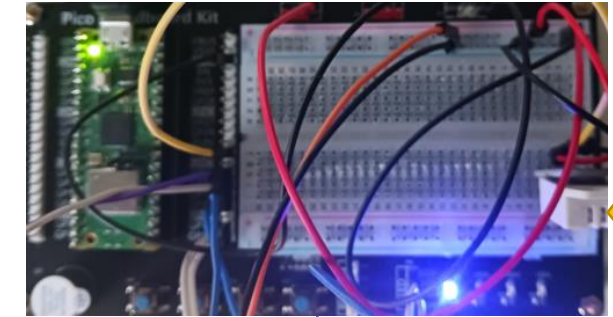
Idx	Hardware	ID	Unit	Name	Type	SubType	Data
15	VirtualSensors	1405F	1	DHT22	Temp + Humidity	THGN122/123/132, THGR122/228 /238/268	17.0 C, 58 %

**DHT22** 17° C / 58%

 Normal, Dew Point: 8.68° C  
Last Seen: 2023-02-24 15:57:26

Pico W Web Server HTTP API/JSON GET request to Domoticz to update the device.  
Example HTTP URL:  
<http://domoticz-ip:8080/json.htm?type=command&param=udevice&idx=15&nvalue=0&svalue=17;58;0>

## Pico W with Breadboard Kit & DHT22



Get DHT22 Data

## Thonny - Pico W Web Server MicroPython Script & Log

```
Thonny - C:\Users\pico\Documents\domoticz-microcontroller-workbook\Chapters\Raspberry Pi Pico W\Projects\DHT22\Server\dh22domoticz.py @ 50.1
File Edit View Run Tools Help
dh22domoticz.py
This computer
C:\Users\pico\Documents\domoticz-microcontroller-workbook\Chapters\Raspberry Pi Pico W\Projects\DHT22\Server
domoticz-microcontroller-w
domoticz-microcontroller-w
Raspberry Pi Pico W
Projects\DHT22\Server
dh22domoticz.py
config.py
dh22domoticz.py
# Constants
VERSION = "PicoW-DHT22toDomoticz v20230224"
CRLF = chr(13) + chr(10)
SPACE = chr(32)
# Init the LED objects led using data from config.py
# Create the onboard LED object to indicate controller is up and network connected
ledstatus = Pin("LED", Pin.OUT)
ledstatus.off()
# Create the led object indicating dht22 measurement in progress
led = Pin(config.PIN_LED1, Pin.OUT)
# DHT22
## DHT22 Signal Pin GP22 #Pin 20
PIN_DHT22 = 22
## DHT22 measurement sampling rate in seconds
SAMPLING_RATE_DHT22 = 60
51 ## DHT22_IDX of the Domoticz TempHum device
52 DHT22_IDX = 15
53 ## URL Domoticz
54 ## Note the idx of the domoticz device ( see GUI > Setup > Devices)
55 ## The svalue is added in the main loop after getting the data from the DHT22 module.
56
Shell
$ curl -X GET http://192.168.1.179:8080/json.htm?type=command&param=udevice&idx=15&nvalue=0&svalue=17;58;0
Sending data OK
DHT22 measuring...
DHT22 measuring OK
DHT22 read data...
DHT22 read data OK
Sending data to http://192.168.1.179:8080/json.htm?type=command&param=udevice&idx=15&nvalue=0&svalue=17;58;0
Sending data OK
Domoticz-ip
```



# Project DHT22 - Custom Event

*Pico W samples DHT22 sensor data and triggers Automation Event to update the Domoticz Temp + Humidity device.*

## Domoticz Hardware

Idx	Name	Enabled	Type	Address	Port	Data Timeout
3	VirtualSensors	Yes	Dummy (Does nothing, use for virtual switches only) <a href="#">Create Virtual Sensors</a>			Disabled

Create Virtual Sensor

Name: DHT22

Sensor Type: Temp+Hum

OK

Cancel

## Domoticz Devices

Idx	Hardware	ID	Unit	Name	Type	SubType	Data
15	VirtualSensors	1405F	1	DHT22	Temp + Humidity	THGN122/123/132, THGR122/228 /238/268	17.0 C, 58 %

DHT22

Normal, Dew Point: 8.68° C  
Last Seen: 2023-02-24 15:57:26

Log

Edit

Notifications

```
1 -- picow_dht22_customew
2 -- Listen to picow webserver request custom event command and update the temp+hum dev
3 -- 20230225 rubi
4
5 -- Test:
6 -- http://192.168.1.179:8080/json.htm?type=command&param=customevent&event=DHT22&data={
7 -- [{"
8 2023-02-24 16:33:03.190 Status: dzVents: Info: Handling Domoticz custom event for: "DHT22"
9 2023-02-24 16:33:03.190 Status: dzVents: Info: Start Internal script: picow_d
10 2023-02-24 16:33:03.191 Status: dzVents: Info: [{"isXML"}]=false, [{"isSystem"}]=false,
11 2023-02-24 16:33:03.191 Status: dzVents: Info: [{"t"}]=19, [{"h"}]=58, [{"s"}]=0
12 2023-02-24 16:33:03.192 Status: dzVents: Info: t=19,h=58,s=0
13 2023-02-24 16:33:03.192 Status: dzVents: Info: ----- Finished picow_dht22_customew
14 ]}
15
16 local IDX_DHT22 = 15
17 local CUSTOM_EVENT_NAME = "DHT22"
18
19 return {
20   on = {
21     customEvents = { CUSTOM_EVENT_NAME }
22   },
23   data = {},
24   logging = {},
25   execute = function(domoticz, triggeredItem)
26     if (triggeredItem.isCustomEvent) then
27       domoticz.log(triggeredItem)
28       domoticz.log(triggeredItem.data)
29       local data = triggeredItem.json
30       domoticz.log(string.format("t=%d,h=%d,s=%d", data.t, data.h, data.s))
31       domoticz.devices(IDX_DHT22).updateTempHum(data.t, data.h, data.s)
32       -- Take any other action based on temp or hum or hum_stat
33     end
34   end
35 }
```

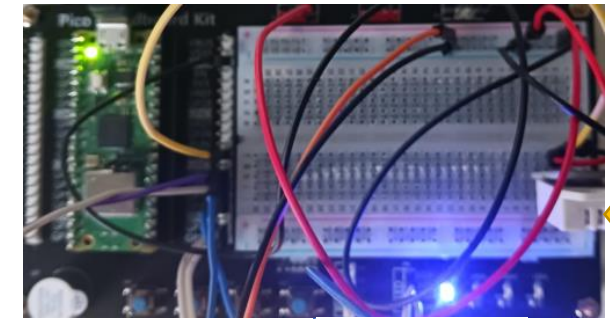
Domoticz Automation Event dzVents  
Trigger: Custom Event

Pico W Web Server HTTP API/JSON POST request to Domoticz Automation Event to update the device.

Example HTTP URL:

[http://domoticz-ip:8080/json.htm?type=command&param=customevent&event=DHT22&data={\"h\": 58, \"t\": 17, \"s\": 0}](http://domoticz-ip:8080/json.htm?type=command&param=customevent&event=DHT22&data={\)

## Pico W with Breadboard Kit & DHT22



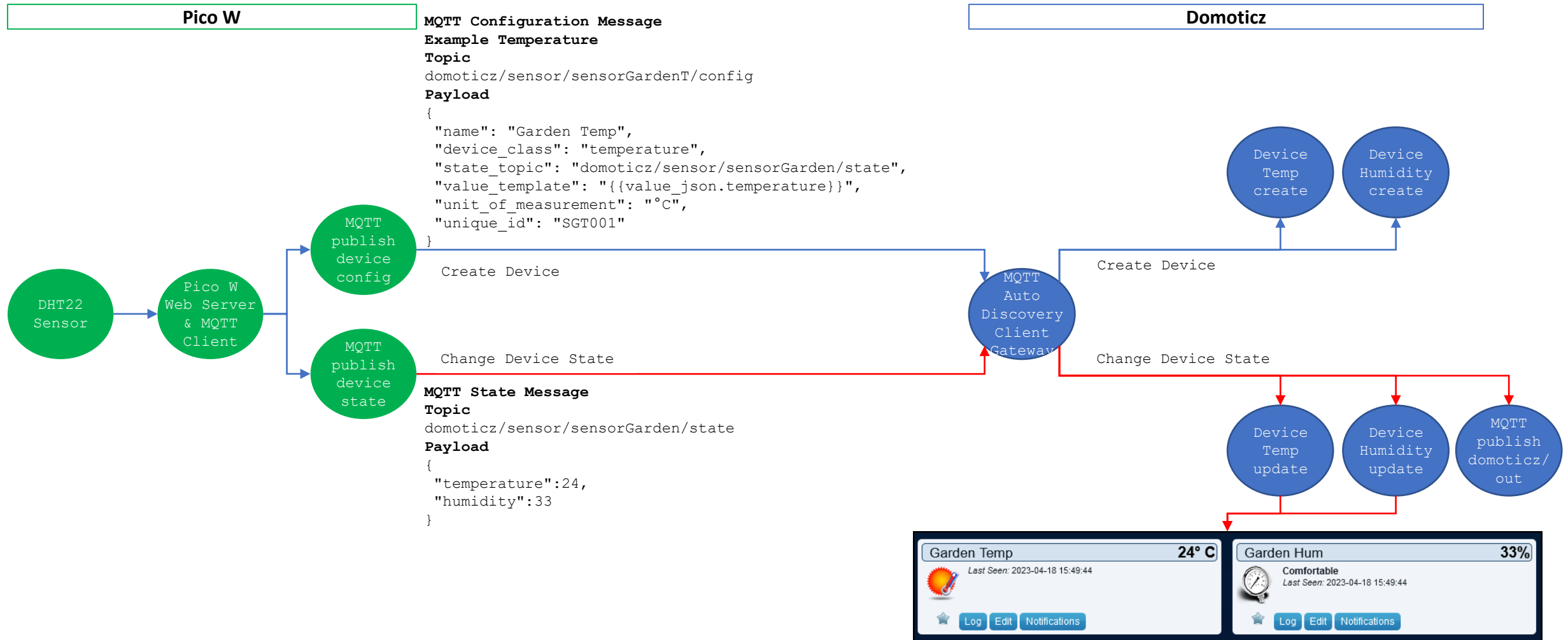
Get DHT22 Data

## Thonny - Pico W Web Server MicroPython Script & Log

```
1 import json
2 from machine import Pin
3 from utime import sleep
4 # DHT22 module internal lib
5 from dht import DHT22
6 # Configuration (must be uploaded to the Pico W)
7 import config
8
9 # Constants
10 VERSION = "PicoW-DHT22toDomoticz v20230224"
11
12 CRLF = chr(13) + chr(10)
13 SPACE = chr(32)
14
15 # Init the LED objects led using data from config.py
16 # Create the onboard LED object to indicate controller is up and network connected
17 ledstatus = Pin("LED", Pin.OUT)
18 ledstatus.off()
19 # Create the led object indicating dht22 measurement in progress
20 led = Pin(config.PIN_LED1, Pin.OUT)
21
22 # DHT22
23 # DHT22 Signal Pin GP22 WPin 29
24 PIN_DHT22 = 22
25
26 # DHT22 measurement sampling rate in seconds
27 SAMPLING_RATE_DHT22 = 60
28
29 # DHT22 IDK of the Domoticz Temp+Hum device
30 IDX_DHT22 = 15
31
32 # URL Domoticz
33 # Note the idea of the domoticz device ( see GUI > Setup > Devices)
34 # The swalue is added in the main loop after getting the data from the DHT22 module.
35
36 # Main loop
37 while True:
38     # Read data from DHT22
39     dht22 = DHT22(PIN_DHT22)
40     temp, hum, temp_humid = dht22.read_data()
41     # Send data to Domoticz
42     url = "http://domoticz-ip:8080/json.htm?type=command&param=customevent&event=DHT22&data={\"h\": %d, \"t\": %d, \"s\": 0}" % (hum, temp)
43     req = urllib.request.Request(url)
44     resp = urllib.request.urlopen(req)
45     # Log the response
46     log.info(resp.read())
```

# Project DHT22 - MQTT Autodiscover

*Pico W publishes MQTT Config & State Messages to Domoticz MQTT Auto Discovery Client Gateway.*



# Project LCD LED Control

*Domoticz Switch Device State Change triggers Automation Event to set the Pico W LED1 & LCD2004 display.*

## Domoticz Hardware

Idx	Name	Enabled	Type	Address	Port	Data Timeout
3	VirtualSensors	Yes	Dummy (Does nothing, use for virtual switches only) <a href="#">Create Virtual Sensors</a>			Disabled

**Create Virtual Sensor**

Name:


Sensor Type:


## Domoticz Devices

Idx	Hardware	ID	Unit	Name	Type	SubType	Data
16	VirtualSensors	00014060	1	LED1 Control	Light/Switch	Switch	On

## Device Widget

**LED1 Control** On

 Last Seen: 2023-02-23 19:12:58  
Type: Light/Switch, Switch, On/Off



```
16 2023-02-23 19:12:58: LOG: LOG_PICOW_LED1CONTROL: 1
17 2023-02-23 19:12:58: LOG: LOG_PICOW_LED1CONTROL: 1
18 2023-02-23 19:12:58: LOG: LOG_PICOW_LED1CONTROL: 1
19 2023-02-23 19:12:58: LOG: LOG_PICOW_LED1CONTROL: 1
20 2023-02-23 19:12:58: LOG: LOG_PICOW_LED1CONTROL: 1
21 2023-02-23 19:12:58: LOG: LOG_PICOW_LED1CONTROL: 1
22 2023-02-23 19:12:58: LOG: LOG_PICOW_LED1CONTROL: 1
23 2023-02-23 19:12:58: LOG: LOG_PICOW_LED1CONTROL: 1
24 2023-02-23 19:12:58: LOG: LOG_PICOW_LED1CONTROL: 1
25 2023-02-23 19:12:58: LOG: LOG_PICOW_LED1CONTROL: 1
26 2023-02-23 19:12:58: LOG: LOG_PICOW_LED1CONTROL: 1
27 2023-02-23 19:12:58: LOG: LOG_PICOW_LED1CONTROL: 1
28 2023-02-23 19:12:58: LOG: LOG_PICOW_LED1CONTROL: 1
29 2023-02-23 19:12:58: LOG: LOG_PICOW_LED1CONTROL: 1
30 2023-02-23 19:12:58: LOG: LOG_PICOW_LED1CONTROL: 1
31 2023-02-23 19:12:58: LOG: LOG_PICOW_LED1CONTROL: 1
32 2023-02-23 19:12:58: LOG: LOG_PICOW_LED1CONTROL: 1
33 2023-02-23 19:12:58: LOG: LOG_PICOW_LED1CONTROL: 1
34 2023-02-23 19:12:58: LOG: LOG_PICOW_LED1CONTROL: 1
35 2023-02-23 19:12:58: LOG: LOG_PICOW_LED1CONTROL: 1
36 2023-02-23 19:12:58: LOG: LOG_PICOW_LED1CONTROL: 1
37 2023-02-23 19:12:58: LOG: LOG_PICOW_LED1CONTROL: 1
38 2023-02-23 19:12:58: LOG: LOG_PICOW_LED1CONTROL: 1
39 2023-02-23 19:12:58: LOG: LOG_PICOW_LED1CONTROL: 1
40 2023-02-23 19:12:58: LOG: LOG_PICOW_LED1CONTROL: 1
41 2023-02-23 19:12:58: LOG: LOG_PICOW_LED1CONTROL: 1
42 2023-02-23 19:12:58: LOG: LOG_PICOW_LED1CONTROL: 1
43 2023-02-23 19:12:58: LOG: LOG_PICOW_LED1CONTROL: 1
44 2023-02-23 19:12:58: LOG: LOG_PICOW_LED1CONTROL: 1
45 2023-02-23 19:12:58: LOG: LOG_PICOW_LED1CONTROL: 1
46 2023-02-23 19:12:58: LOG: LOG_PICOW_LED1CONTROL: 1
47 2023-02-23 19:12:58: LOG: LOG_PICOW_LED1CONTROL: 1
48 2023-02-23 19:12:58: LOG: LOG_PICOW_LED1CONTROL: 1
49 2023-02-23 19:12:58: LOG: LOG_PICOW_LED1CONTROL: 1
50 2023-02-23 19:12:58: LOG: LOG_PICOW_LED1CONTROL: 1
51 2023-02-23 19:12:58: LOG: LOG_PICOW_LED1CONTROL: 1
52 2023-02-23 19:12:58: LOG: LOG_PICOW_LED1CONTROL: 1
53 2023-02-23 19:12:58: LOG: LOG_PICOW_LED1CONTROL: 1
54 2023-02-23 19:12:58: LOG: LOG_PICOW_LED1CONTROL: 1
55 2023-02-23 19:12:58: LOG: LOG_PICOW_LED1CONTROL: 1
56 2023-02-23 19:12:58: LOG: LOG_PICOW_LED1CONTROL: 1
57 2023-02-23 19:12:58: LOG: LOG_PICOW_LED1CONTROL: 1
58 2023-02-23 19:12:58: LOG: LOG_PICOW_LED1CONTROL: 1
59 2023-02-23 19:12:58: LOG: LOG_PICOW_LED1CONTROL: 1
60 2023-02-23 19:12:58: LOG: LOG_PICOW_LED1CONTROL: 1
61 2023-02-23 19:12:58: LOG: LOG_PICOW_LED1CONTROL: 1
62 2023-02-23 19:12:58: LOG: LOG_PICOW_LED1CONTROL: 1
63 2023-02-23 19:12:58: LOG: LOG_PICOW_LED1CONTROL: 1
64 2023-02-23 19:12:58: LOG: LOG_PICOW_LED1CONTROL: 1
65 2023-02-23 19:12:58: LOG: LOG_PICOW_LED1CONTROL: 1
66 2023-02-23 19:12:58: LOG: LOG_PICOW_LED1CONTROL: 1
67 2023-02-23 19:12:58: LOG: LOG_PICOW_LED1CONTROL: 1
```

Domoticz Automation Event dzVents triggered by state change of the Domoticz Switch device (IDX=16).

The event submits a HTTP POST request to the Pico W Web Server to switch the LED1 On/Off and set the state on the LCD2004.

## Pico W with Breadboard Kit & LCD 20x4 & LED1 On



## Thonny - Pico W Web Server Script & Log

```
109 return server
110 except OSError as e:
111     ledstatus.off()
112     cl.close()
113     raise RuntimeError('[ERROR] M
114
115 # Get the command from the HTTP respo
116 # The last line of the HTTP response
117 # The HTTP response is decoded and sp
118 # Example {"state":"on"}
119 # Return Command
120 def getCommand(request):
121     cmd = {"state":config.MESSAGE_CMD
122     # Split the decoded request strin
123     data = str(request.decode()).spli
124     # print(data)
125     # Check if there is data to get t
126     if (len(data) == 9):
127         # JSON parse the last list it
128         # Convert the string to a JSD
129         cmd = json.loads(data[len(dat
130     else:
131         print('[ERROR] HTTP POST requ
132         # Return the command as JSON obje
133         print(f'HTTP Command: {cmd}')
134         return cmd
```

# Project LCD Motherboard

*Domoticz sends RPi Motherboard sensor data to set the LCD 20x4 I2C display connected to the Pico W.*

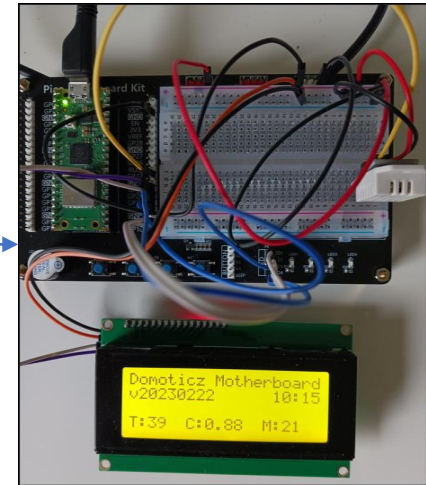
## Domoticz Hardware

Idx	Name	Enabled	Type	Address	Port	Data Timeout
5	RPi Motherboard	Yes	Motherboard sensors			Disabled

## Domoticz Devices (selective)

Idx	Hardware	ID	Unit	Name	Type	SubType	
18	RPi Motherboard	0001	1	Internal Temperature	Temp	LaCrosse TX3	38.9 C
26	RPi Motherboard	0000044D	1	CPU_Usage	General	Percentage	0.41%
22	RPi Motherboard	0000044C	1	Memory Usage	General	Percentage	18.35%

Pico W with  
Breadboard Kit & LCD 20x4



```
On Off picow_lcdmotherboard
44 -- setSensor(3,5,'T',round(domoticz.devices(IDX_INTERNALTEMPERATURE
45 - local function setSensor(row,col,pre,data)
46   return setText(row,col,string.format('%s%s', pre, tostring(data))
47 end
48 -- [{"internaltemperature"}]= [{"col"}=5, [{"data"}="T:39.4", [{"row"}]=3
49
50
51 - local function getMotherboardData(domoticz)
52   local data = {}
53
54   -- row=1, col=15, data=HH:MM len=5, range = 15+5-1=19
55   data['timestamp'] = setText(1, 15, string.sub(domoticz.time.rawT,
56
57   -- row=3, col=0, data=T:NN len=4, range = 0+4-1=3
58   data['internaltemperature'] = setSensor(3, 0, 'T:', round(domoti
59   -- row=3, col=6, data=C:NN.NN len=5, range = 6+5-1=10
60   data['cpuusage'] = setSensor(3, 6, 'C:', round(domoti
61   -- row=3, col=14, data=M:NN len=4, range = 14+4-1=17
62   data['memoryusage'] = setSensor(3, 14, 'M:', round(domoti
63
64
65
66 --[[
67 data['internaltemperature'] = round(domoticz.devices(IDX_INTERNA
68 data['armclockspeed'] = round(domoticz.devices(IDX_ARMCLOC
69 data['v3dclockspeed'] = round(domoticz.devices(IDX_V3DCLOC
70 data['coreclockspeed'] = round(domoticz.devices(IDX_CORECLOC
71 data['memoryusage'] = round(domoticz.devices(IDX_MEMORYUS
72 data['processusage'] = round(domoticz.devices(IDX_PROCESSU
73 data['hddboot'] = round(domoticz.devices(IDX_HDDBOOT)
74 data['hdd'] = round(domoticz.devices(IDX_HDD)
75 data['cpuusage'] = round(domoticz.devices(IDX_CPUUSAG
76 ]])
77
78 -- Test embedding json: r=row 0-3, c=col 0-19, d=data
79 --[[
80 local x = {}
81 x['r']=3
82 x['c']=5
83 x['d']=string.format('T:%s', tostring(data['internaltemperature']
84 data['temp'] = x
85 ]])
86 domoticz.log(data)
87 return data
88 end
```

## Domoticz Automation Event dzVents

Trigger: Timer

An event is reading in regular intervals the RPi motherboard data.

The data is converted to a JSON object and sent as HTTP POST request to the Pico W Web Server.

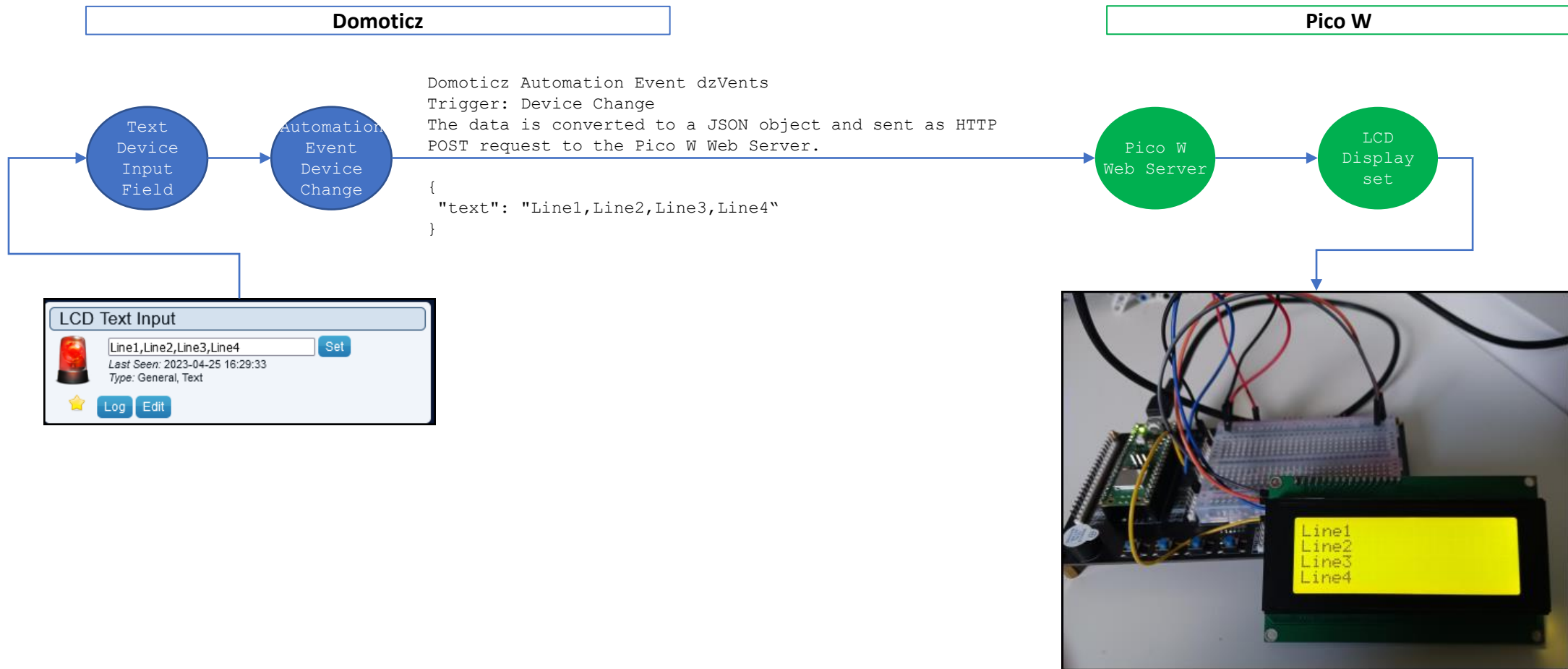
```
{
  ["internaltemperature"]=
  { ["col"]=0, ["text"]="T:38", ["row"]=3},
  ["timestamp"]=
  { ["col"]=15, ["text"]="16:19", ["row"]=1},
  ["memoryusage"]=
  { ["col"]=14, ["text"]="M:21", ["row"]=3},
  ["cpuusage"]=
  { ["col"]=6, ["text"]="C:0.35", ["row"]=3}
}
```

## Thonny - Pico W Web Server Script & Log

```
Files
This computer
C:\Daten\projects\homeautomation\
domoticz\
domoticz-microcontroller-workbook\
Chapters\ Raspberry Pi Pico W \
Experiments\LCDMotherboard\ Source
  lcdmotherboard.dzvents
  lcdmotherboard.py
Raspberry Pi Pico
  lcd_api.py
  machine_i2c_lcd.py
  config.py
lcdmotherboard.py
139 raise RuntimeError('[ERROR] Network Connection closed')
140
141 # Get the data from the HTTP response.
142 # The last line of the HTTP response contains the data.
143 # The HTTP response is decoded and split as a string list.
144 # Return data
145 def getHttpData(request):
146   # Split the decoded request string into a list
147   data = str(request.decode()).split(CRLF)
148   # print(data)
149   # Check if there is data to get the first item
150   if (len(data) == 0):
151     # JSON parse the last list item holding the data as JSON
152     # Convert the string to a JSON object
153     data = json.loads(data[len(data) - 1])
154   else:
155     print('[ERROR] HTTP POST request not valid.')
156   # Return the data as JSON object
157   print(f'HTTP POST Data from Domoticz: {CRLF}{data}')
158   return data
159
160 # Main
161 # Listen for incoming connections from the Domoticz Automation Event
162 print(f' (VERSION)')
163
164 # Set the LCD display object
165 lcd = setLCD(LCD_I2C_ADDRESS, LCD_PIN_SDA, LCD_PIN_SCL, LCD_ROWS,
166
167 # Show initial info on the LCD. Waiting is replaced by RPi mother
168 setLCDWelcome(NAME, VERSION, '', WAITING)
169
Shell
>>> %Run -c $EDITOR_CONTENT
Domoticz Motherboard v20230222
==: [39]
Domoticz-ip - red. IP: 192.168.1.108
Network - Client connected from 192.168.1.23
HTTP POST Data from Domoticz:
{'timestamp': {'text': '11:51', 'col': 15, 'row': 1}, 'memoryusage':
ture': {'text': 'T:38', 'col': 0, 'row': 3}}
HTTP Response to Domoticz:
{'status': 'OK', 'title': 'Set LCD', 'message': ''}
```

# Project LCD Text Input

*Domoticz sends text from a Text Device enabling Input to set the LCD 20x4 I2C display connected to the Pico W.*



Inspired by [this](#) Domoticz Forum Post (thanks for sharing).

# Project LED Display TM1637

*Domoticz sends RPi Motherboard Internal Temperature to set the 4-Digit 7-Segment display connected to the Pico W.*

## Domoticz Hardware

Idx	Name	Enabled	Type	Address	Port	Data Timeout
5	RPi Motherboard	Yes	Motherboard sensors			Disabled

## Domoticz Devices

Idx	Hardware	ID	Unit	Name	Type	SubType	Value
18	RPi Motherboard	0001	1	Internal Temperature	Temp	LaCrosse TX3	37.0 C

```
picow_tm1637_motherboard_internal_temperature x +
On Off picow_tm1637_motherboard
20 HTTP_RES = 'RES_PICOW_LCDLEDCONTROL'
21
22 LOG_MARKER = 'LOG_PICOW_LCDLEDCONTROL'
23
24 TIMER_RULE = "every minute"
25
26 -- Helpers
27 local function round(number, decimals)
28     local power = 10^decimals
29     return math.floor(number * power) / power
30 end
31
32 -- Post Data to the Pico W Webserver.
33 -- The data is a JSON object with the temperature rounded
34 local function HTTPPost(domoticz)
35     local data = {}
36     data['data'] = domoticz.devices(IDX_INTERNAL_TEMPERATURE).value
37     domoticz.log(string.format('device=%s, temperature=%s', data['data'], data['data']))
38     domoticz.log(data)
39     -- Submit remote HTTP POST request to set the tm1637 display
40     domoticz.openURL({
41         url = HTTP_URL,
42         method = 'POST',
43         headers = { ['content-type'] = 'application/json' },
44         postData = data,
45         callback = HTTP_RES,
46     })
47 end
```

Domoticz Automation Event dzVents  
Trigger: Timer or Switch  
An event is reading in regular intervals the RPi motherboard internal temperature sensor.

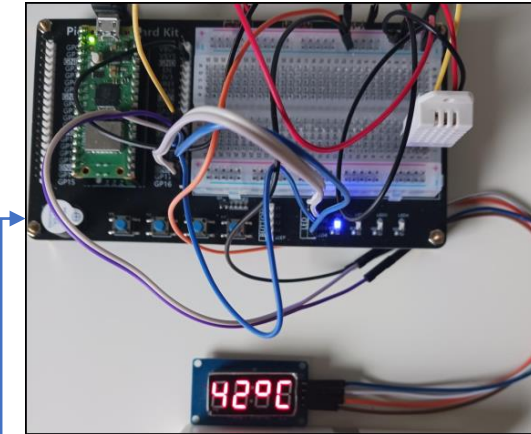
The data is converted to a JSON object and sent as HTTP POST request to the Pico W Web Server.

```
{
  ["data"]=40.900001525879
}
```

## Thonny - Pico W Web Server Script & Log

```
tm1637.py
146 # Create the HTTP response JSON object
147 response = {}
148
149 # Init title (holding the command as JSON object)
150 response[config.KEY_TITLE] = getCommand(request)
151 response[config.KEY_STATE] = config.STATE_ERROR
152 response[config.KEY_MESSAGE] = config.MESSAGE_OKAY
153
154 # TM1637 display set
155 tm.show(' ')
156 sleep(.3)
157
158 # Select the command from the response[config.KEY_TITLE]
159 cmd = response[config.KEY_TITLE]
160 # Get the temperature rounded (no digits)
161 temperature = round(cmd['data'])
162 # Set the display - number only, i.e. 41
163 # tm.number(temperature)
164 # Set the display NN°C
165 tm.temperature(temperature)
```

Pico W with Breadboard Kit & TM1637





# Project Servo Control

*Domoticz Dimmer Device State Change triggers Automation Event to set the angle of a servo motor connected to the Pico W.*

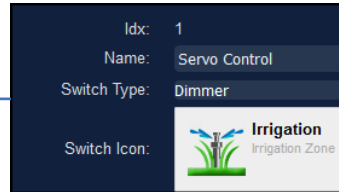
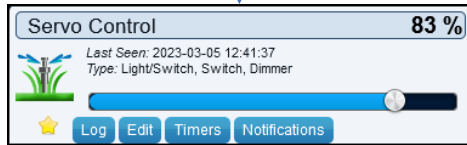
## Domoticz Hardware

Idx	Name	Enabled	Type	Address	Port	Data Timeout
3	VirtualSensors	Yes	Dummy (Does nothing, use for virtual switches only) <a href="#">Create Virtual Sensors</a>			Disabled

## Domoticz Devices

Idx	Hardware	ID	Unit	Name	Type	SubType	
1	VirtualSensors	00014051	1	Servo Control	Light/Switch	Switch	Set Level: 83 %

## Device Widget



Edit Set Switch  
Type to Dimmer  
and Switch Icon  
to Irrigation

```
On Off picow_servocontrol
35 LOG_MARKER = "LOG_PICOW_SERVOCONTROL"
36
37 -- Helpers
38 -- Round to the nearest number
39 local function roundNearest(number)
40     return math.floor(number + 0.5)
41 end
42
43 -- Post Data to the Pico W Webserver.
44 -- The data is a JSON object with the servo angle and
45 local function HTTPPost(domoticz, angle)
46     local data = {}
47     data['angle'] = angle
48     --data['angle'] = domoticz.devices(IDX_INTERNAL_1)
49     domoticz.log(string.format("device=%s, angle=%d",
50     domoticz.openURL({
51         url = HTTP_URL,
52         method = "POST",
53         headers = { ['content-type'] = 'application/json' },
54         postData = data,
55         callback = HTTP_RES,
56     })
57 end
58
59 return {
60     -- Listen to dimmer device changes and HTTP response
61     on = {
62         devices = { IDX_DIMMER },
63         httpResponses = { HTTP_RES },
64     },
65     logging = { level = domoticz.LOG_INFO, marker = LOG_MARKER },
66     execute = function(domoticz, item)
67         -- domoticz.log(item)
68         if item.isDevice then
69             -- Check if state is on, Get the slider
```

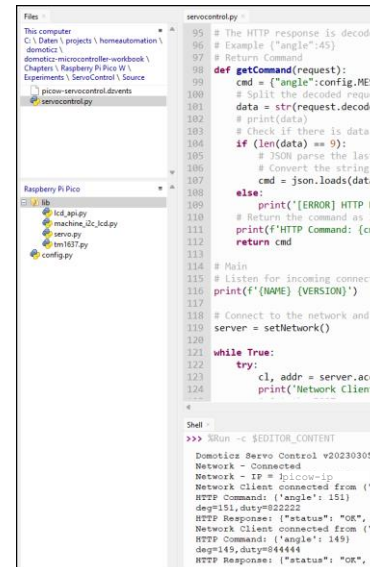
Domoticz Automation Event dzVents triggered by state change of the Domoticz Dimmer device (IDX=1). The servo angle is converted from the dimmer level 0-100% to 0-180 deg. 83% = 149 deg.

The event submits HTTP POST request to the Pico W Web Server to set the servo angle. The post-data is a JSON Object:

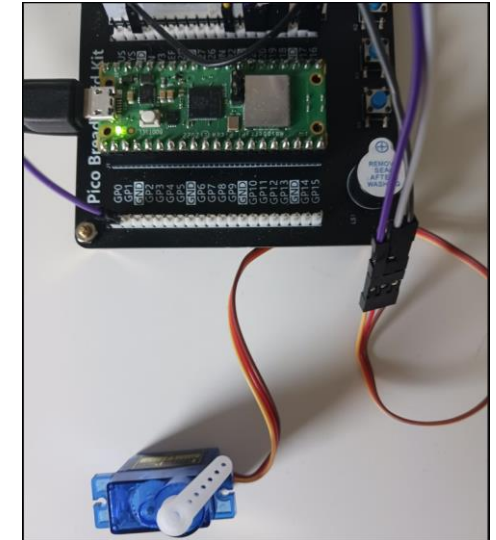
```
{"angle": 149}
```

Device HTTP POST Request to Pico W Web Server

## Thonny - Pico W Web Server MicroPython Script & Log



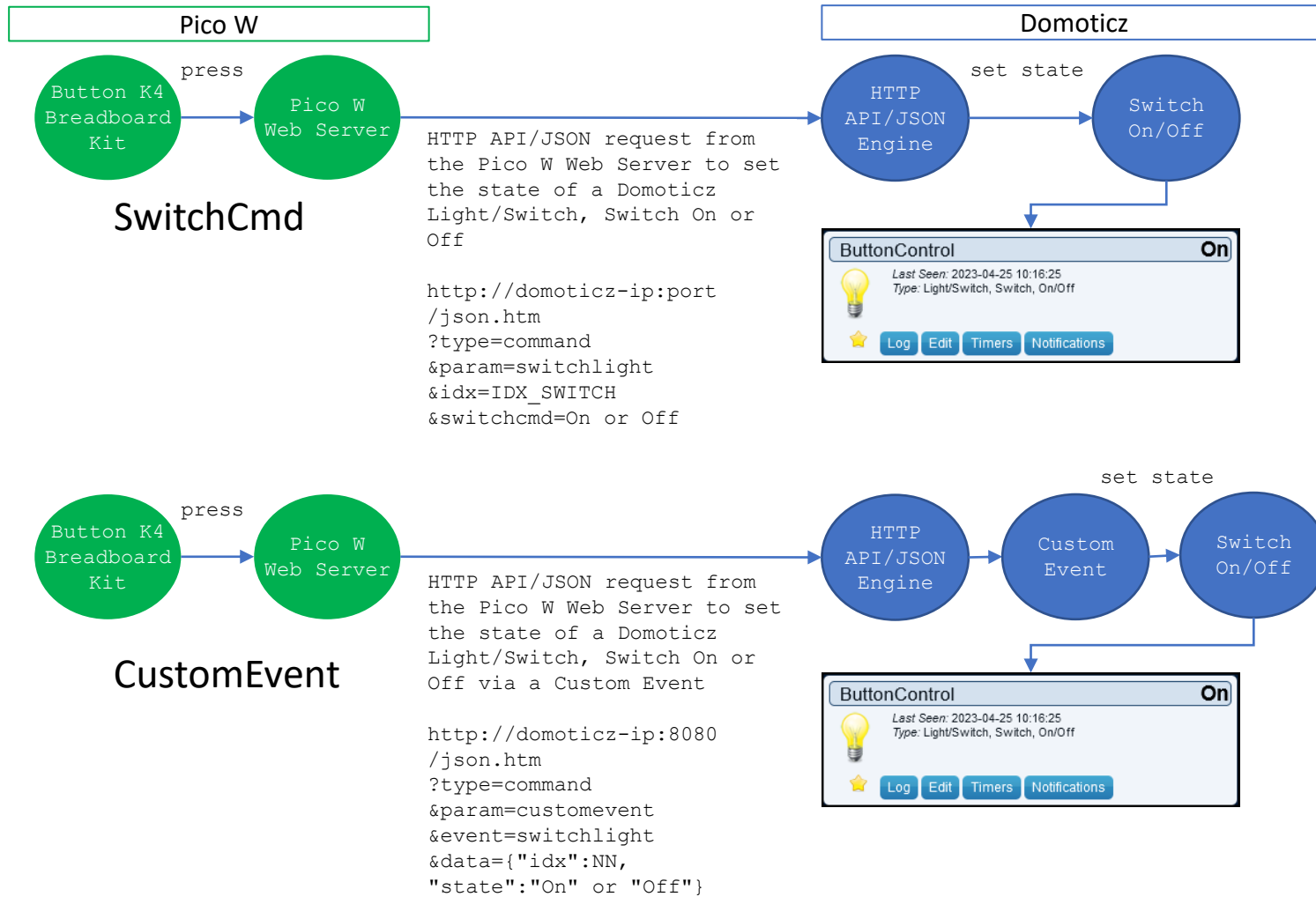
## Pico W with Breadboard Kit & Servo Motor SG90



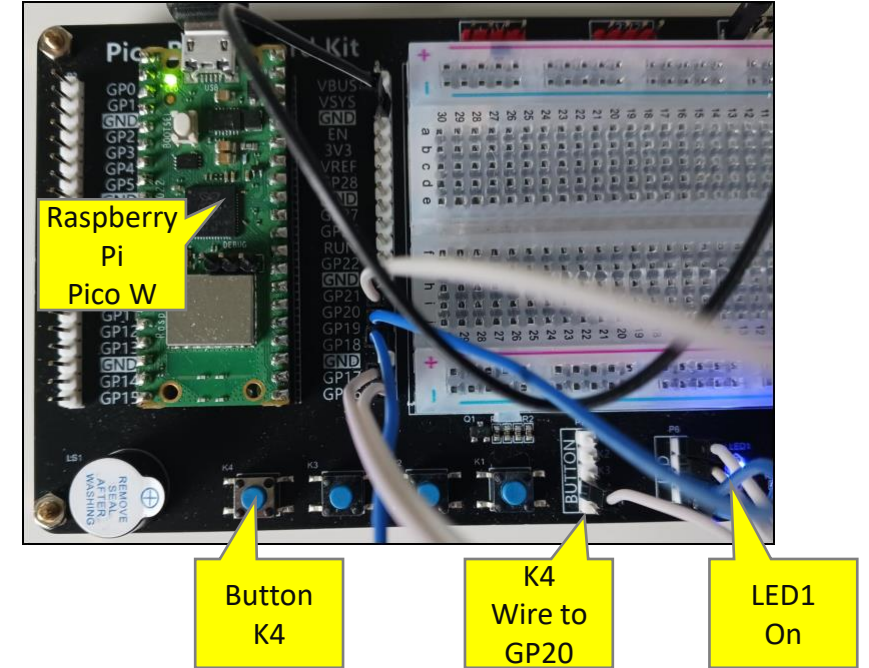
Servo Angle  
149 deg

# Project Button Control

*Pico W with Pushbutton to set the state of a Domoticz Switch On/Off Device.*

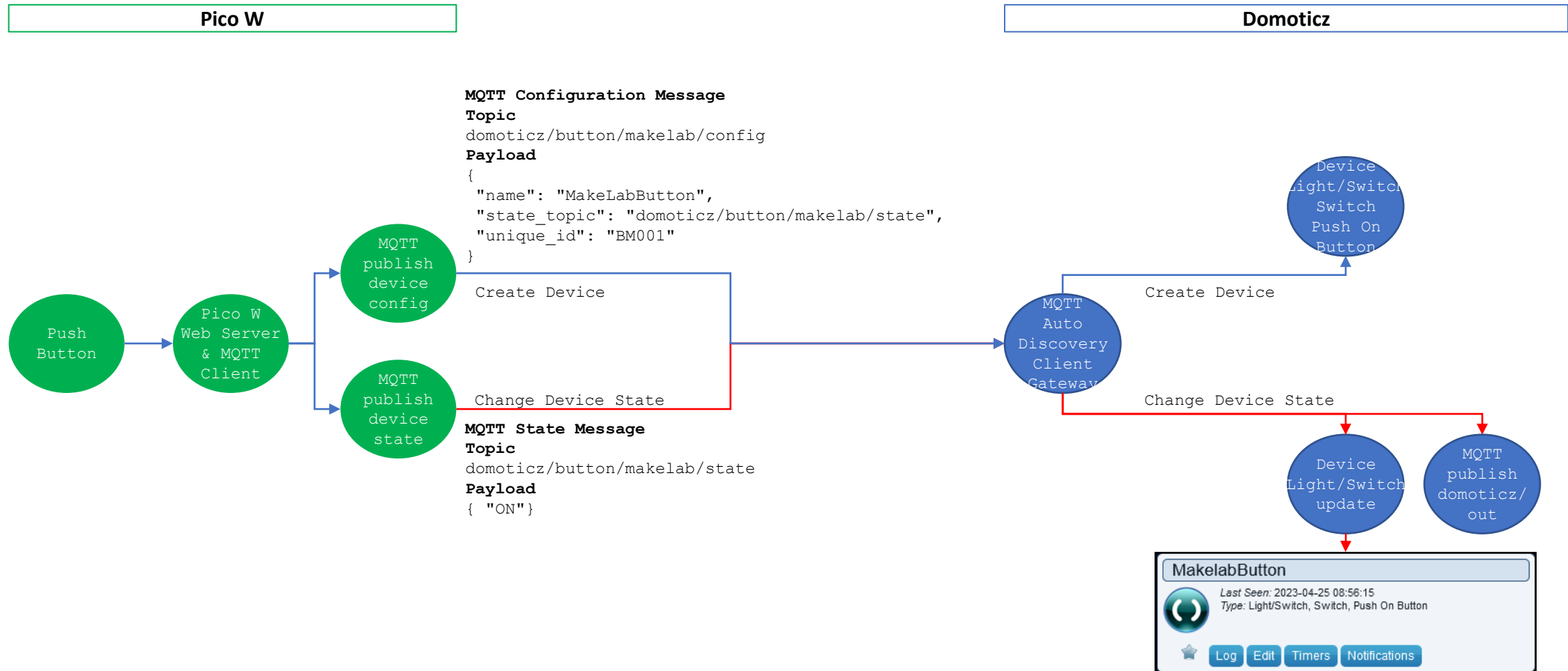


Pico W with Breadboard Kit Button K4 connected



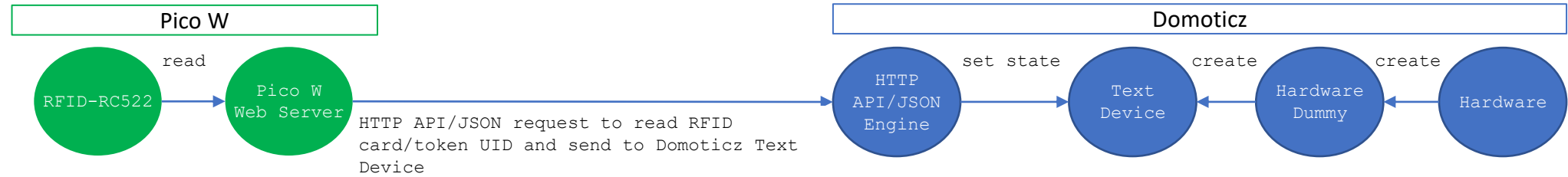
# Project Button Control - MQTT Autodiscover

*Pico W Push button set state of Domoticz Push On Button Device using Domoticz MQTT Auto Discovery Client Gateway.*



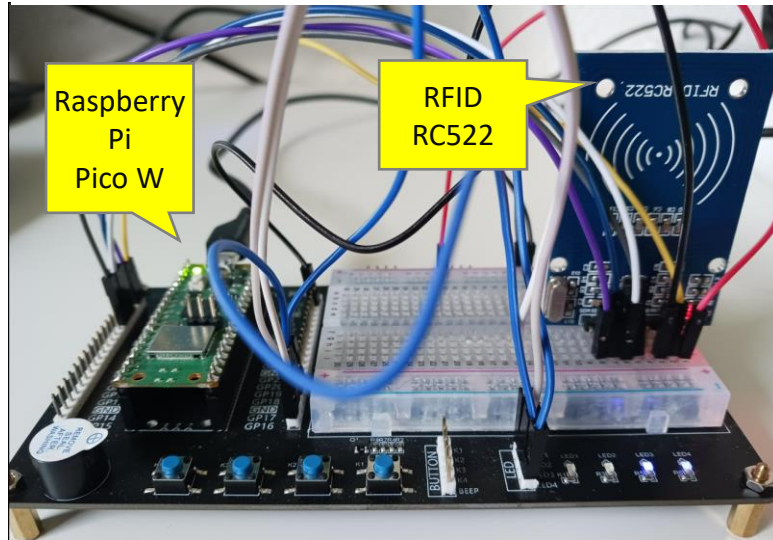
# Project RFID Reader

*Pico W to read RFID cards/tokens via RFID-RC522 module and send the UID to a Domoticz Text Device.*

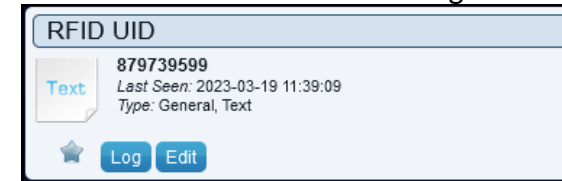


```
http://domoticz-ip:port
/json.htm
?type=command
&param=udevice
&idx=IDX_RFID
&nvalue=0
&svalue=879739599
```

Pico W with Breadboard Kit & RFID-RC522 connected



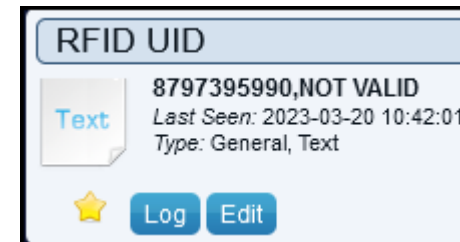
Domoticz Text Device Widget



The RFID card/token UID is:  
879739599 (HEX 346FC2CF).  
The card type is 10.

Domoticz Text Device Log

Date	
2023-03-19 12:04:31	1446518016
2023-03-19 12:04:26	879739599
2023-03-19 12:04:19	249942437
2023-03-19 11:39:08	879739599
2023-03-19 11:39:04	1446518016
2023-03-19 11:37:42	879739599
2023-03-19 11:37:22	1446518016
2023-03-19 11:32:21	879739599

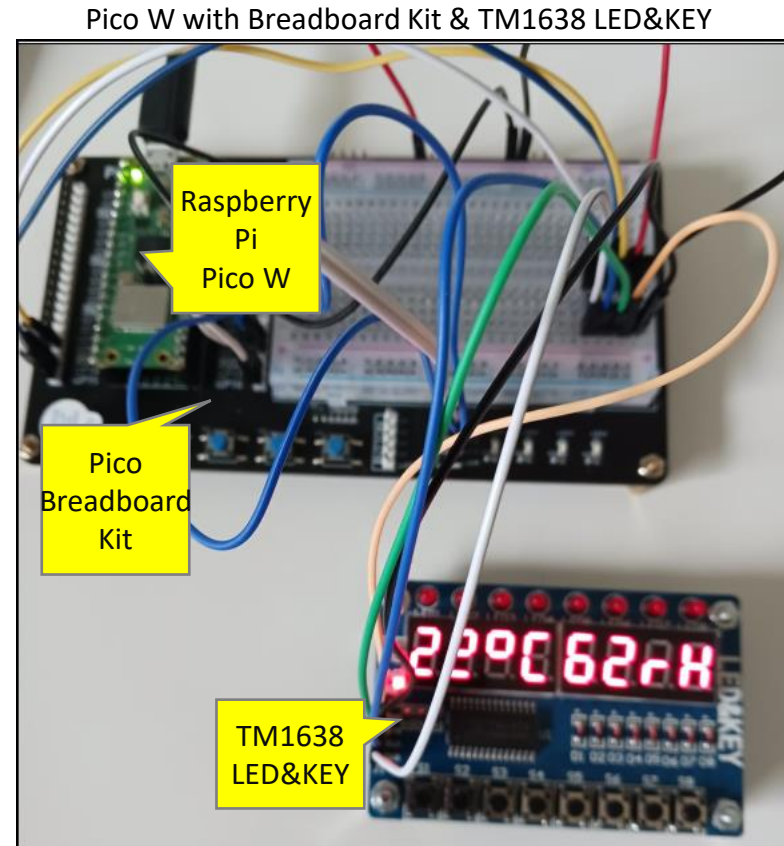
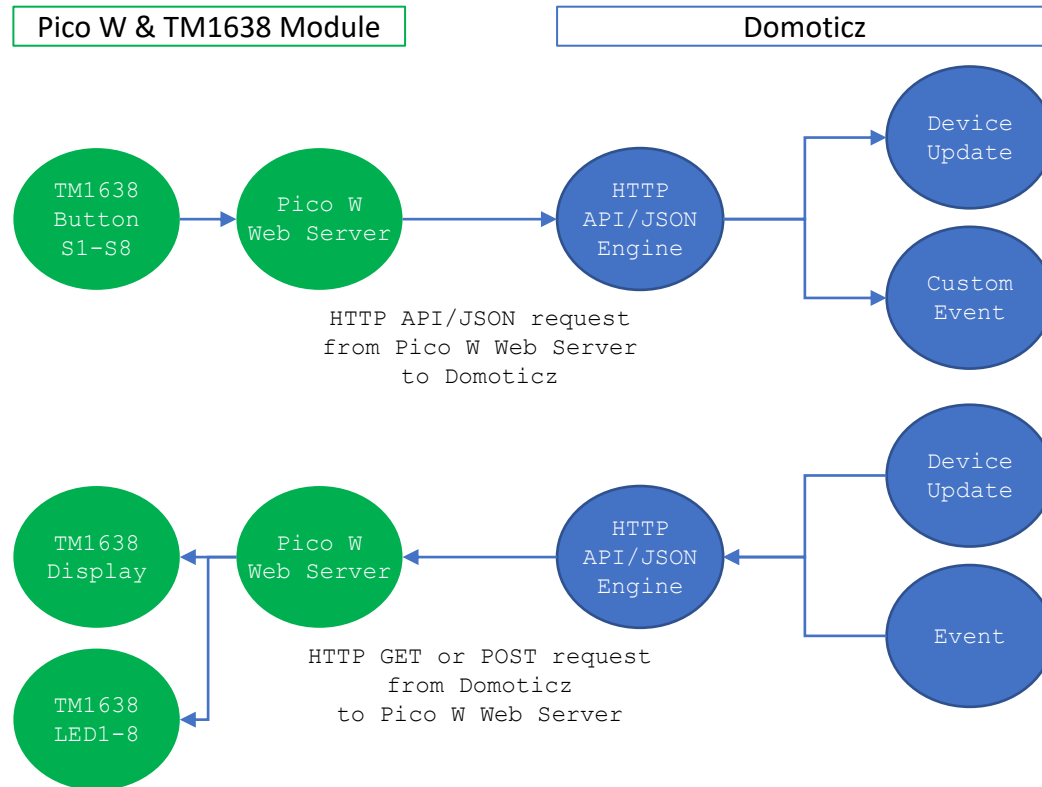


Domoticz Text Device Log with Flag

Date	
2023-03-20 10:42:01	8797395990,NOT VALID
2023-03-20 10:42:01	8797395990
2023-03-20 10:41:49	879739599,VALID
2023-03-20 10:41:49	879739599

# Project TM1638 LED&KEY

*Pico W with TM1638 to trigger Domoticz device action or Domoticz to set TM1638 LED or Display.*





# Project OLED Motherboard

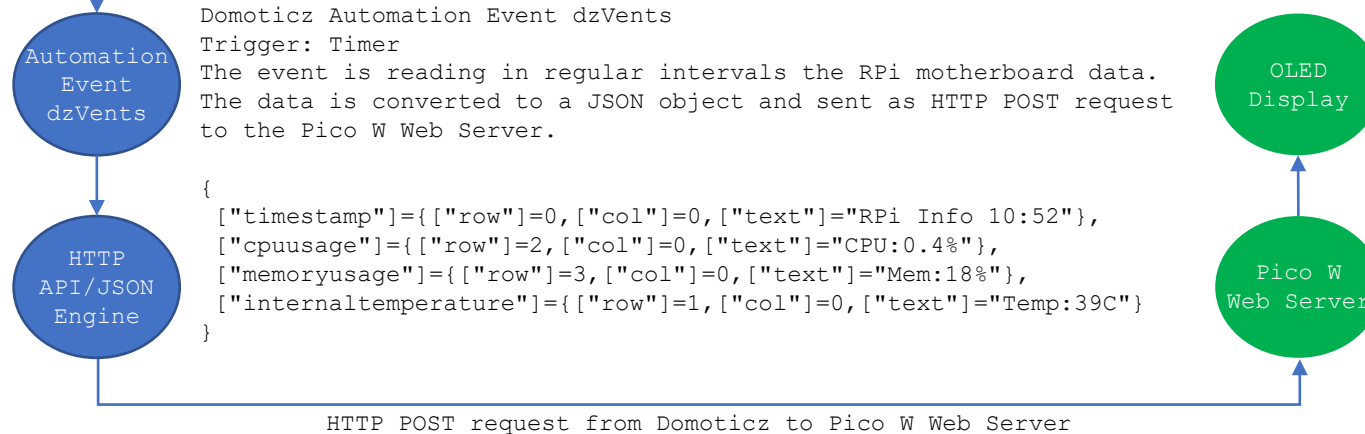
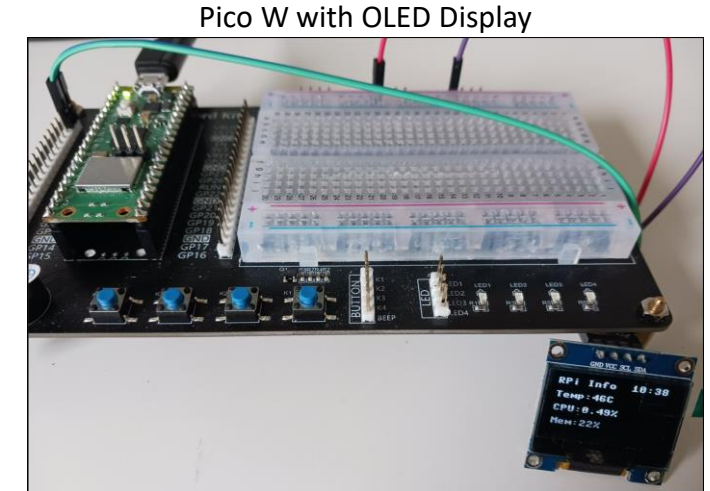
*Domoticz sends RPi Motherboard sensor data to set the 0,96" I2C OLED display connected to the Pico W.*

## Domoticz Hardware

Idx	Name	Enabled	Type	Address	Port	Data Timeout
5	RPi Motherboard	Yes	Motherboard sensors			Disabled

## Domoticz Devices (selective)

Idx	Hardware	ID	Unit	Name	Type	SubType	
18	RPi Motherboard	0001	1	Internal Temperature	Temp	LaCrosse TX3	38.9 C
26	RPi Motherboard	0000044D	1	CPU_Usage	General	Percentage	0.41%
22	RPi Motherboard	0000044C	1	Memory Usage	General	Percentage	18.35%





# Project OLED Motherboard /2

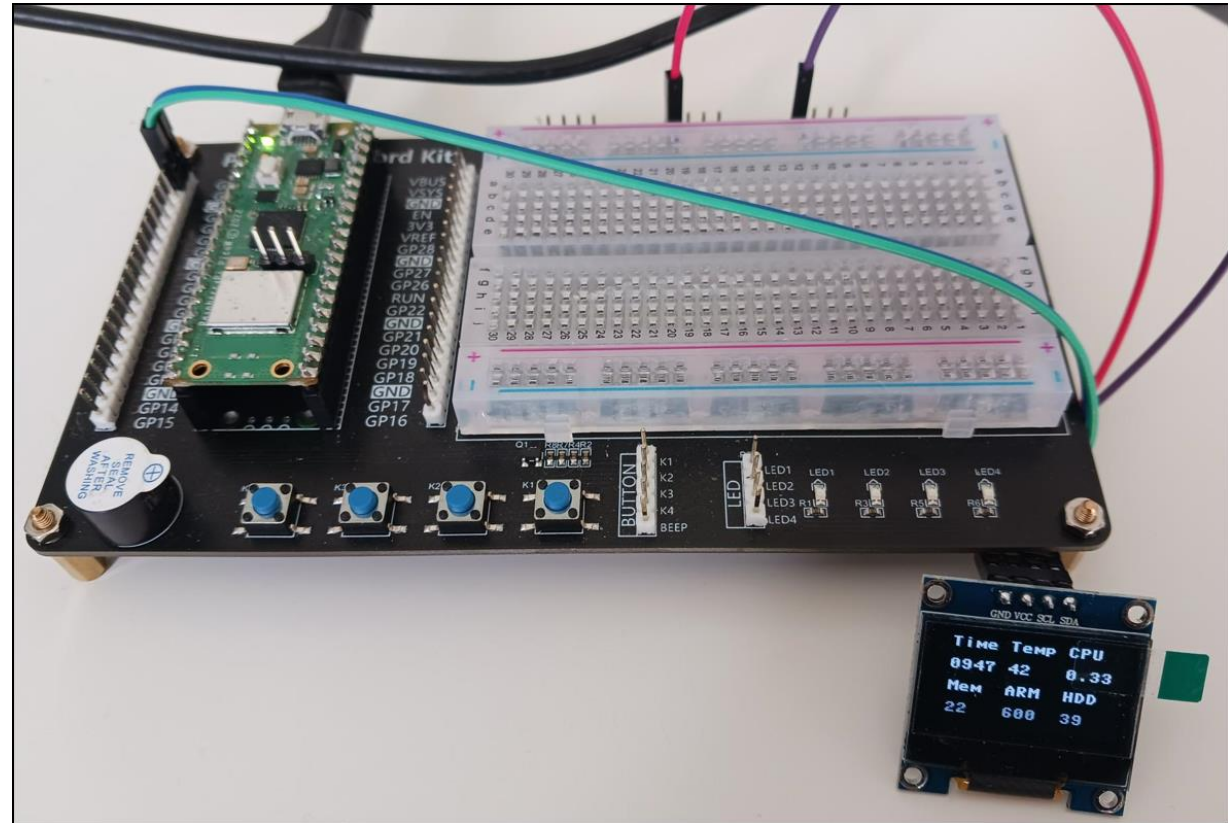
*Domoticz sends RPi Motherboard sensor data to set the 0,96" I2C OLED display connected to the Pico W.  
The sensor data is displayed in up-to 6 blocks with title & value*

OLED Display 6 Blocks Setup  
Each block 2 rows with 4 characters

Col	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Row																
0																
1																
2																
3																

Block Data generated by Domoticz Automation Event.  
Example JSON array with 6 blocks:

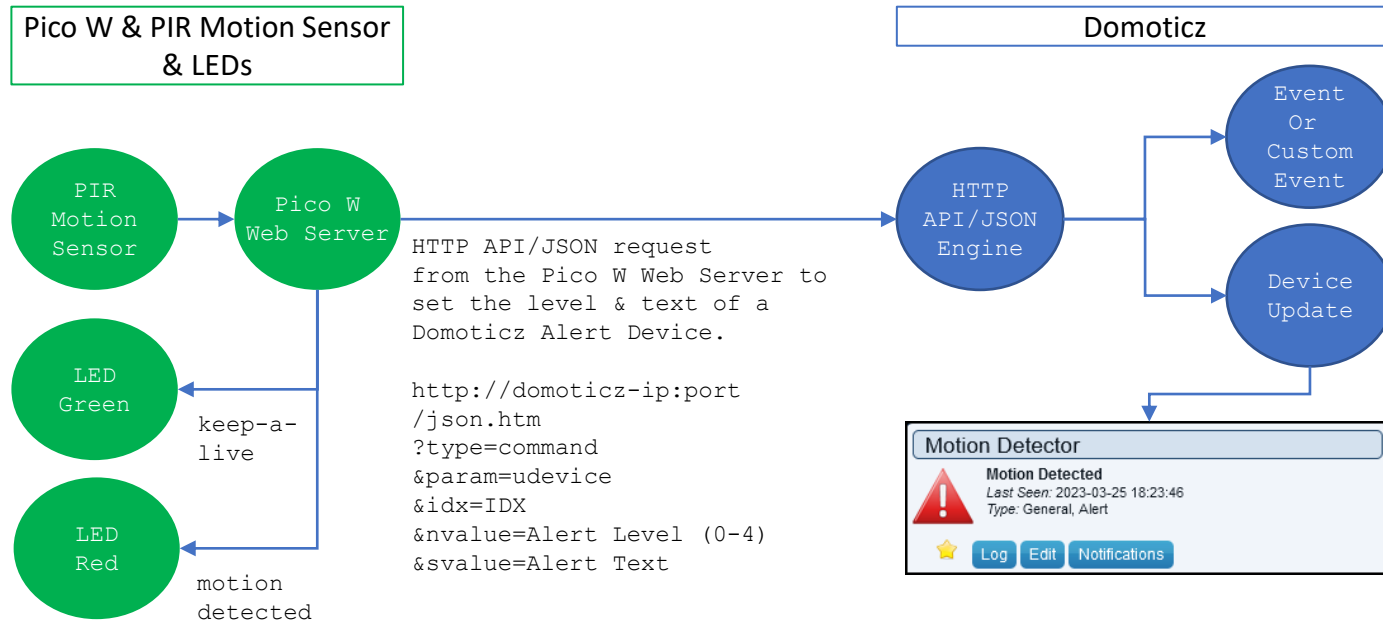
```
[
  {'block': 1, 'title': 'Time', 'value': '0947'},
  {'block': 2, 'title': 'Temp', 'value': 42},
  {'block': 3, 'title': 'CPU', 'value': 0.33},
  {'block': 4, 'title': 'Mem', 'value': 22},
  {'block': 5, 'title': 'ARM', 'value': 600},
  {'block': 6, 'title': 'HDD', 'value': 39}
]
```



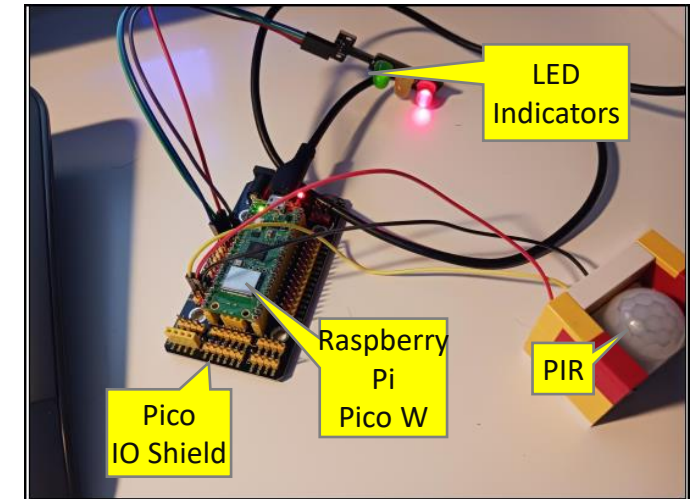
Pico W with OLED Display showing 6 blocks with RPi Motherboard sensor data

# Project PIR Motion Sensor

*PicoW to detect motion and sent message to a Domoticz Alert sensor.*



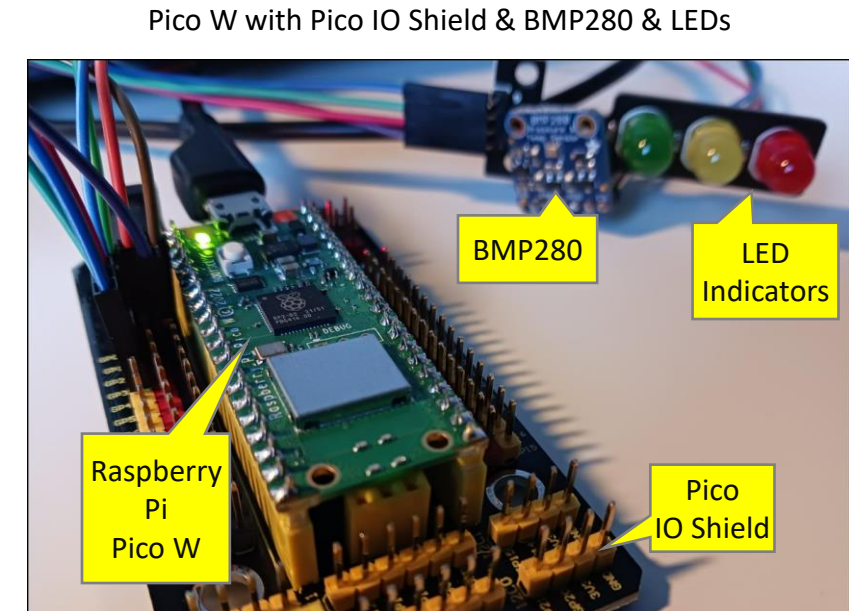
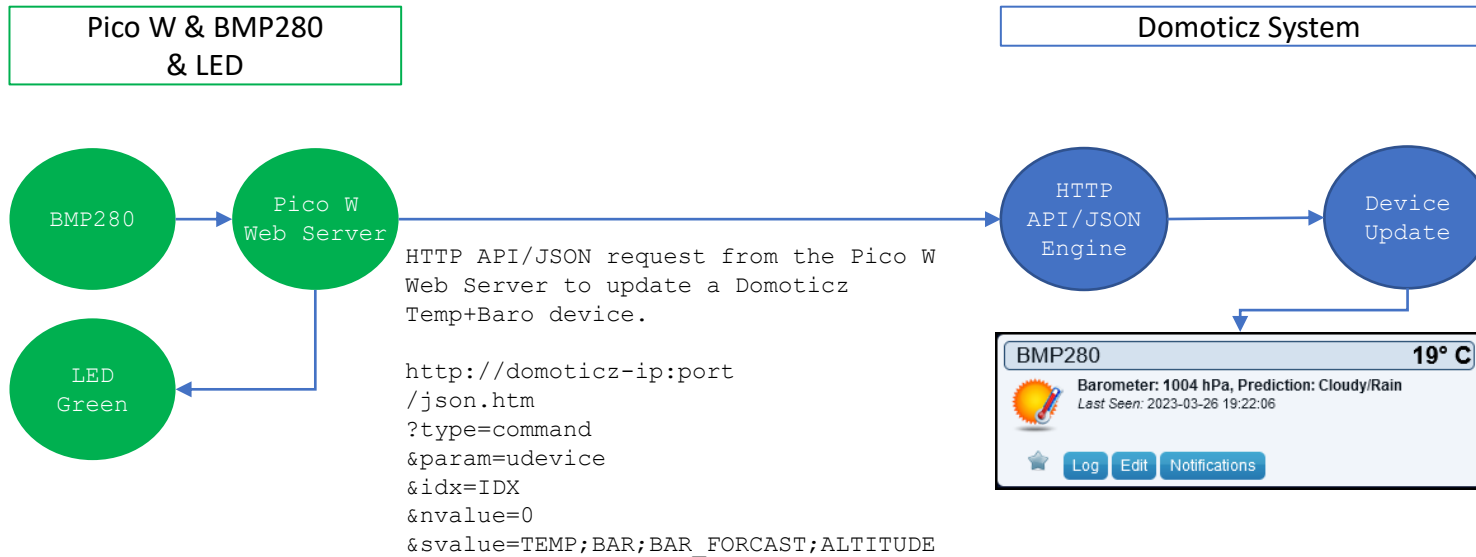
### Pico W with PIR connected, and motion detected



Note: The yellow LED from the LED traffic light is not used.

# Project BMP280

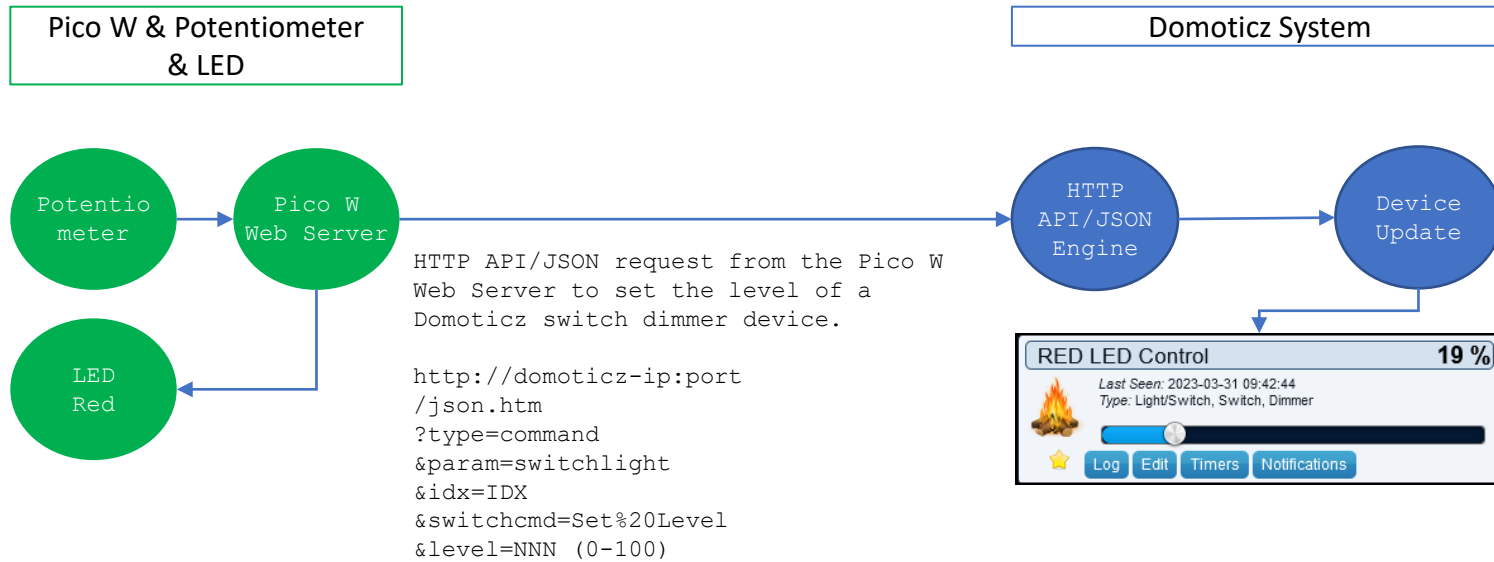
*Pico W samples BMP280 sensor data and triggers updating the Domoticz Temp+Baro device.*



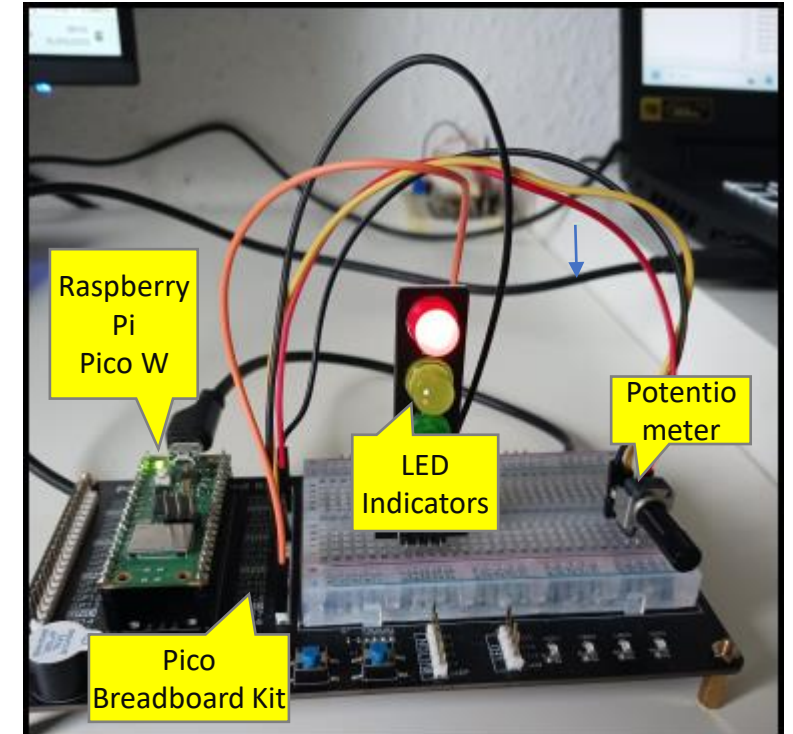
Note: The green LED is used only from the LED traffic light.

# Project Potentiometer Dimmer

*Pico W to set the level of a Domoticz Switch Dimmer device via Potentiometer.*



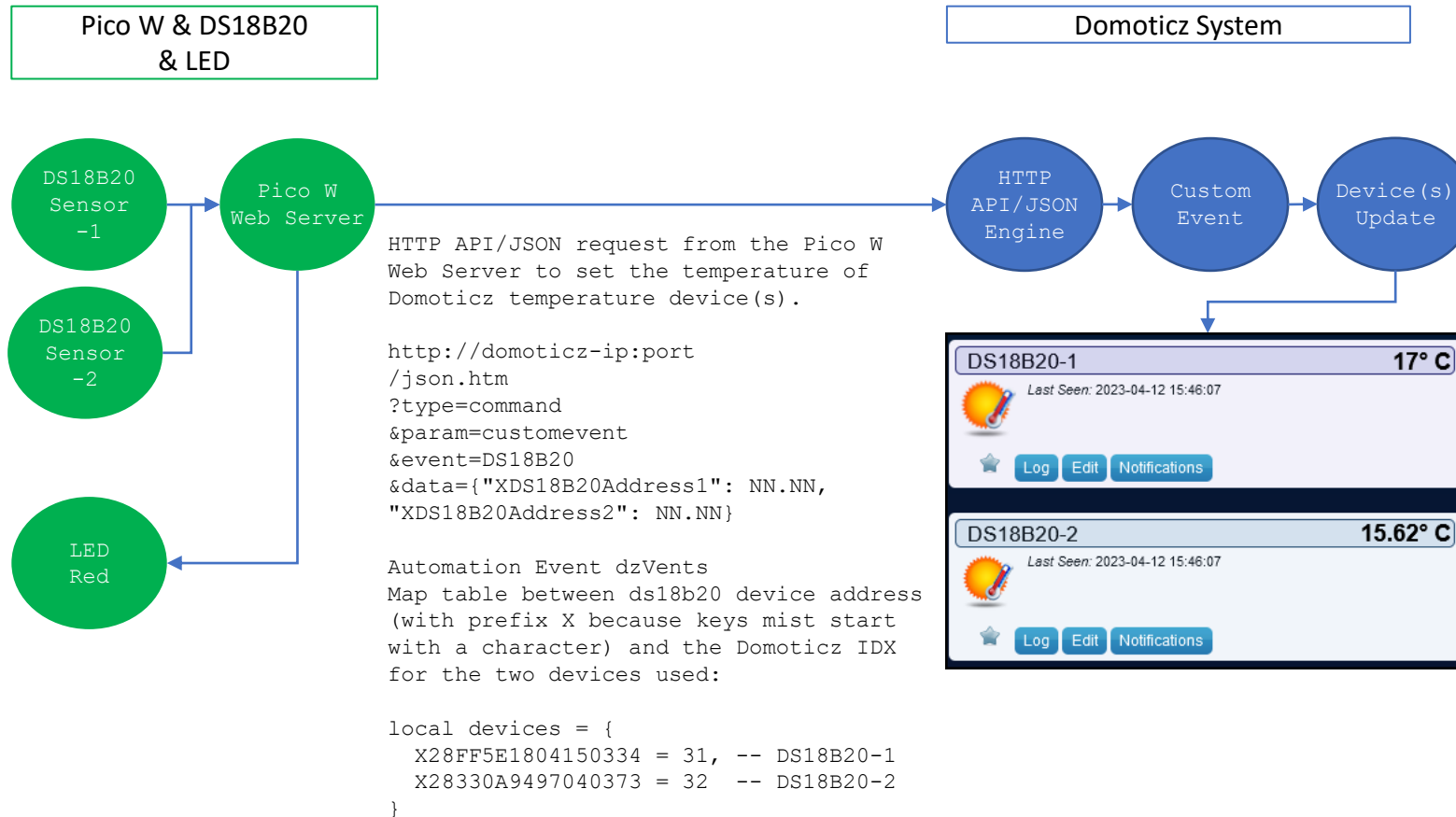
Pico W with Breadboard Kit & Potentiometer & LED



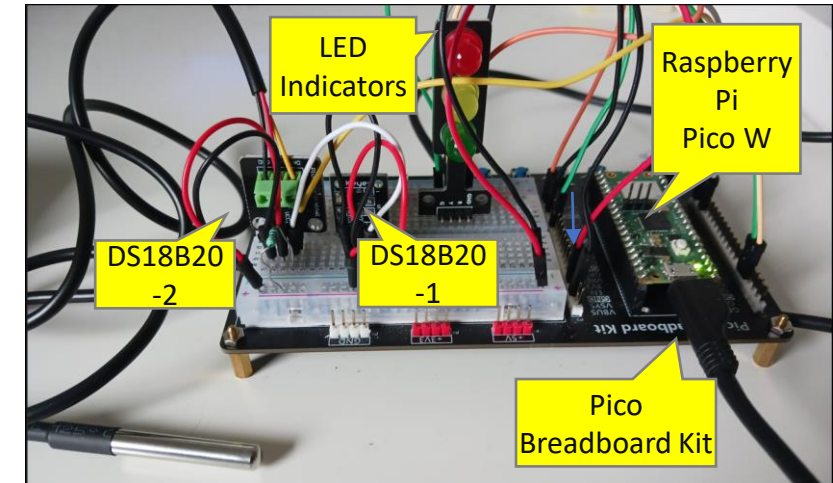
Note: The red LED is used only from the LED traffic light.

# Project DS18B20 - Custom Event

*Pico W samples DS18B20 sensor data and triggers Automation Event to update the Domoticz Temperature devices.*



Pico W with Breadboard Kit & DS18B20 & LED

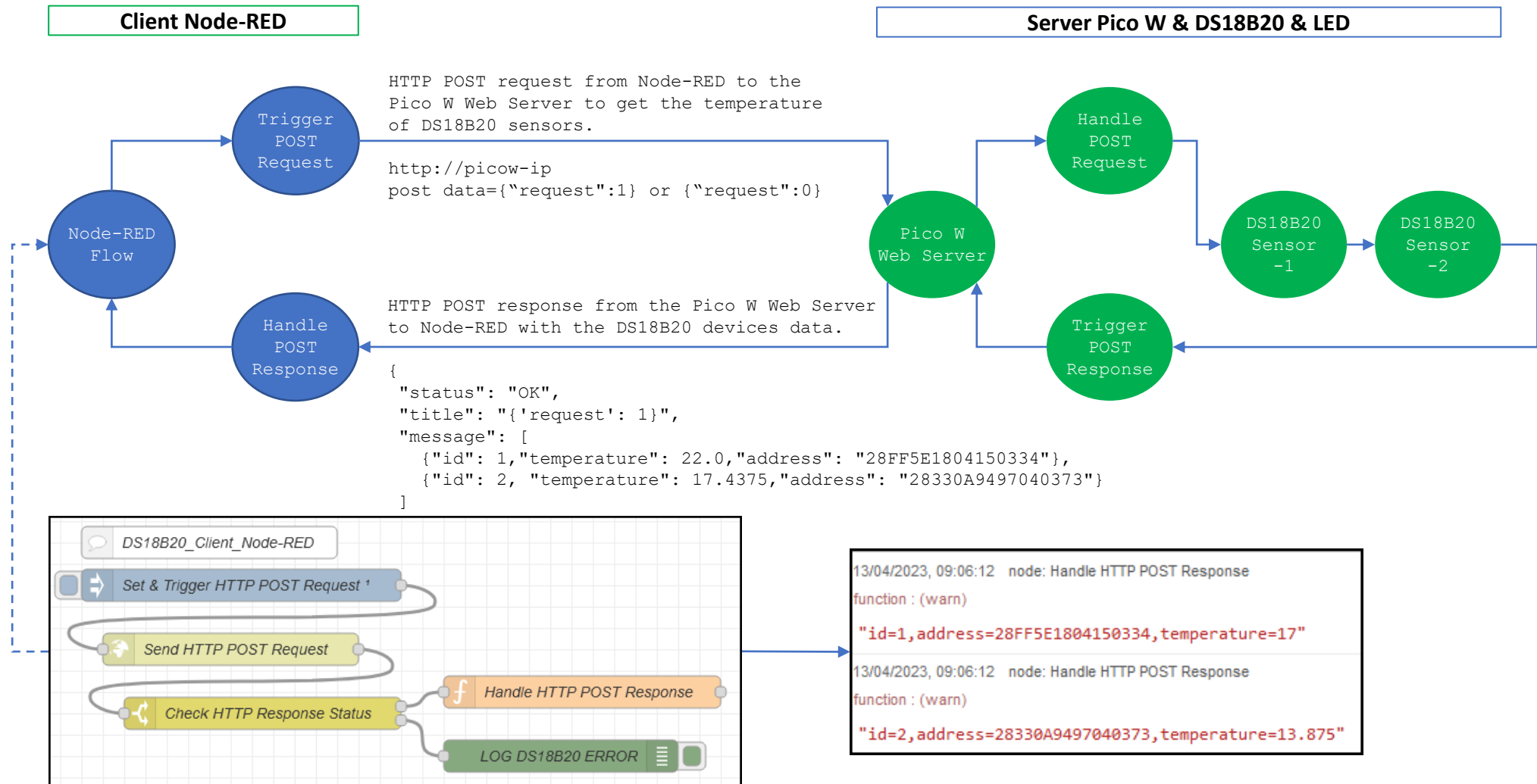


Note: There are two DS18B20 connected (DS18B20-1, DS18B20-2).  
From the LED traffic light, the red LED is used only.



# Project DS18B20 - Node-RED (PULL)

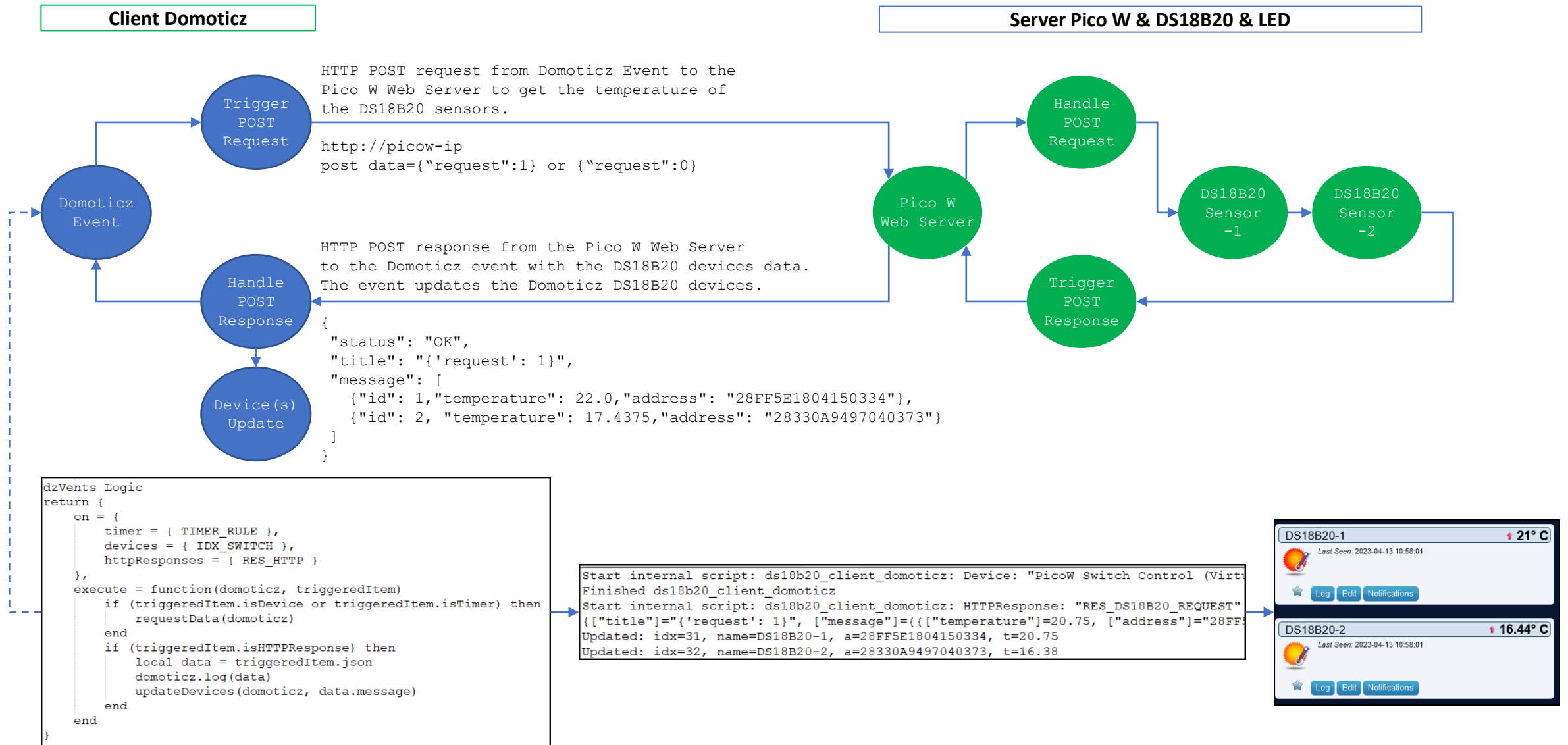
*Pico W Web Server listens to HTTP POST request from Node-RED and sends response with DS18B20 sensor data.*





# Project DS18B20 - Domoticz (PULL)

*Pico W Web Server listens to HTTP POST request from Domoticz and sends response with DS18B20 sensor data.*



# Project Stepper Motor – Selector Switch Angle Move

*Domoticz Selector Switch Device State Change triggers Automation Event to move the Stepper Motor by Angle.*

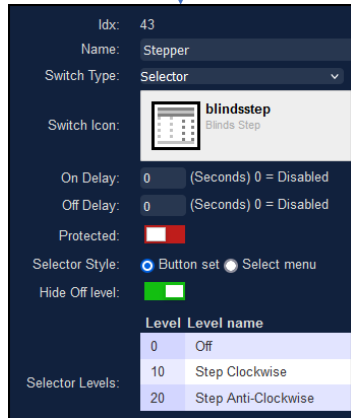
## Domoticz Hardware

Idx	Name	Enabled	Type	Address	Port	Data Timeout
3	VirtualSensors	Yes	Dummy (Does nothing, use for virtual switches only) <a href="#">Create Virtual Sensors</a>			Disabled

## Domoticz Devices

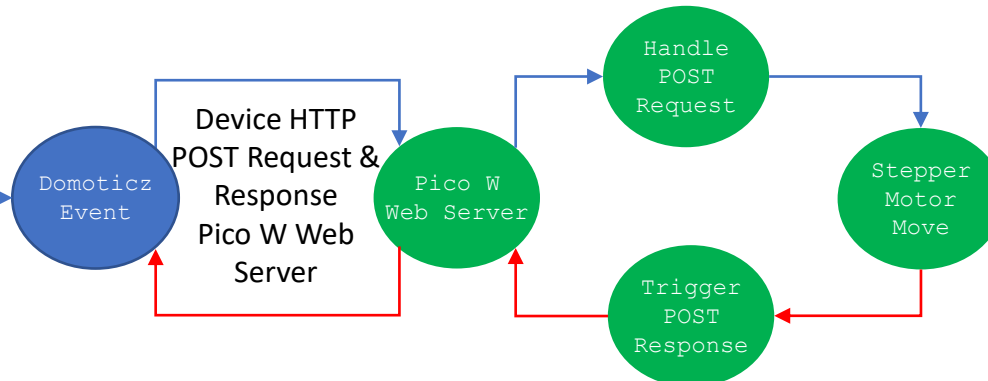
Idx	Hardware	ID	Unit	Name	Type	SubType	
43	VirtualSensors	0001407B	1	Stepper	Light/Switch	Selector Switch	Set Level: 20 %

## Device Widget



Device properties with

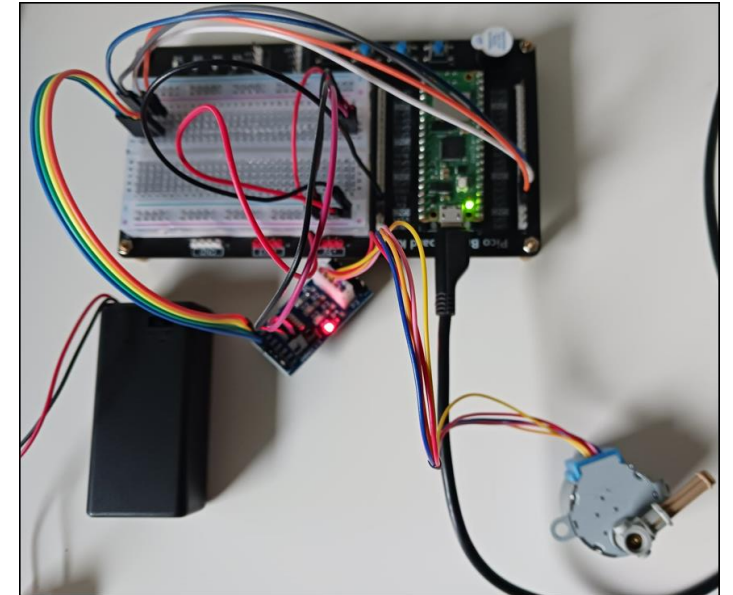
- custom icon
- 3 levels; Off level hidden.



Domoticz Automation Event dzVents triggered by state change of the Domoticz Selector Switch device (IDX=43).  
The stepper angle moves 5° (hardcoded in the Automation event script).  
The event submits HTTP POST request to the Pico W Web Server to move the stepper motor by the angle specified.

The post-data is a JSON Object:  
{ "angle": 5 } or { "angle": -5 }

Pico W with Breadboard Kit & UNL2003 driver & 28BYJ-48 stepper motor



# Project Stepper Motor - Blind Simulation

*Simulate Domoticz Blind Device State Change triggers Automation Event to move the Angle of a Stepper Motor.*

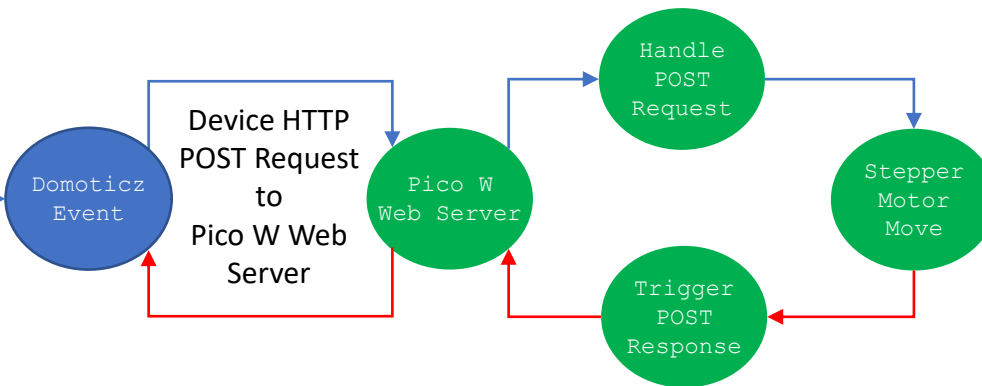
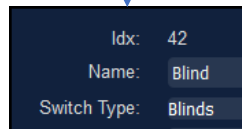
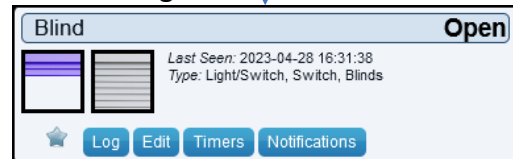
## Domoticz Hardware

Idx	Name	Enabled	Type	Address	Port	Data Timeout
3	VirtualSensors	Yes	Dummy (Does nothing, use for virtual switches only) <a href="#">Create Virtual Sensors</a>			Disabled

## Domoticz Devices

Idx	Hardware	ID	Unit	Name	Type	SubType	Open
42	VirtualSensors	0001407A	1	Blind	Light/Switch	Switch	Open

## Device Widget

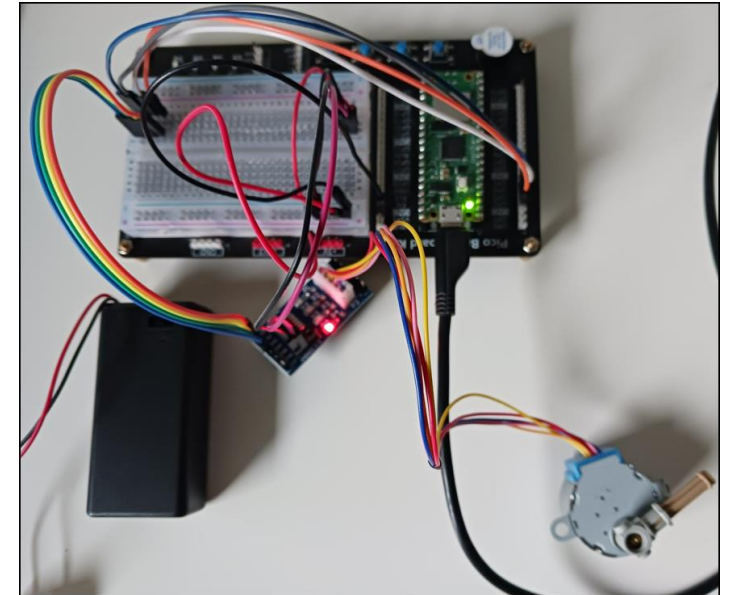


Domoticz Automation Event dzVents triggered by state change of the Domoticz Blinds device (IDX=42).  
The stepper angle to move is for state Open 180° and Closed -180°.  
The event submits HTTP POST request to the Pico W Web Server to move the stepper motor by the angle specified.

The post-data is a JSON Object:  
{ "angle": 180 } or { "angle": -180 }

Note: Instead of using the command "angle", the command "rotate" N times is another possibility.

Pico W with Breadboard Kit & UNL2003 driver & 28BYJ-48 stepper motor



# Project Stepper Motor – Timer Run Stop

*Domoticz Selector Switch State Change triggers Automation Event to run or stop a Stepper Motor.*

## Domoticz Hardware

Idx	Name	Enabled	Type	Address	Port	Data Timeout
3	VirtualSensors	Yes	Dummy (Does nothing, use for virtual switches only) <a href="#">Create Virtual Sensors</a>			Disabled

## Domoticz Devices

Idx	Hardware	ID	Unit	Name	Type	SubType	Off
43	VirtualSensors	0001407B	1	Stepper	Light/Switch	Selector Switch	Off

## Device Widget

**Stepper** **Run Clockwise**

Last Seen: 2023-05-06 17:55:18  
Type: Light/Switch, Selector Switch, Selector

Stop

Run Clockwise


Run Anti-Clockwise

Log

Edit

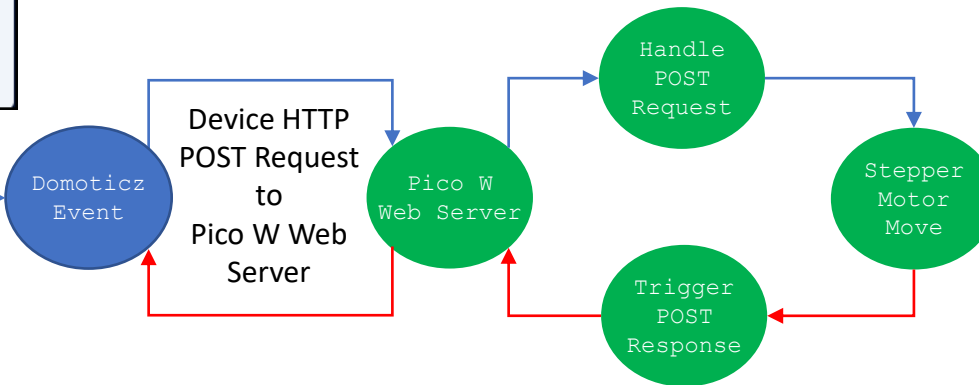
Timers

Notifications

Idx: 43  
Name: Stepper  
Switch Type: Selector  
Switch Icon: 

On Delay: 0 (Seconds) 0 = Disable  
Off Delay: 0 (Seconds) 0 = Disable  
Protected: ☒  
Selector Style: ☒ Button set ☐ Select menu  
Hide Off level: ☒

Level	Level name
0	Stop
10	Run Clockwise
20	Run Anti-Clockwise



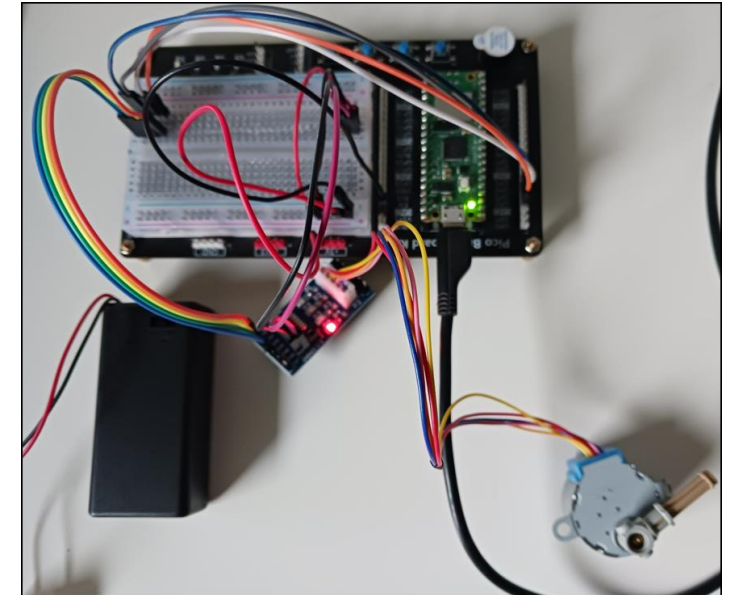
Domoticz Automation Event dzVents triggered by state change of the Domoticz Selector Switch device (IDX=43).

The event submits HTTP POST request to the Pico W Web Server to run or stop the stepper motor.

The post-data is a JSON Object depending action:

- Stop = {"command"="stop"}
- Run Clockwise = {"command"="run", "direction"=1}
- Run Anti-Clockwise = {"command"="run", "direction"=-1}

Pico W with Breadboard Kit & UNL2003 driver & 28BYJ-48 stepper motor



# Project Distance Sensor - HTTP API/JSON

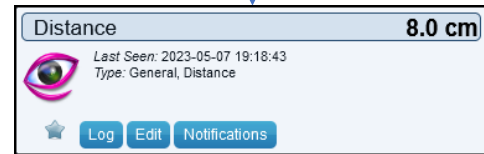
*Pico W samples Distance sensor HC-SR04 data and triggers updating the Domoticz Distance device.*

## Domoticz Hardware

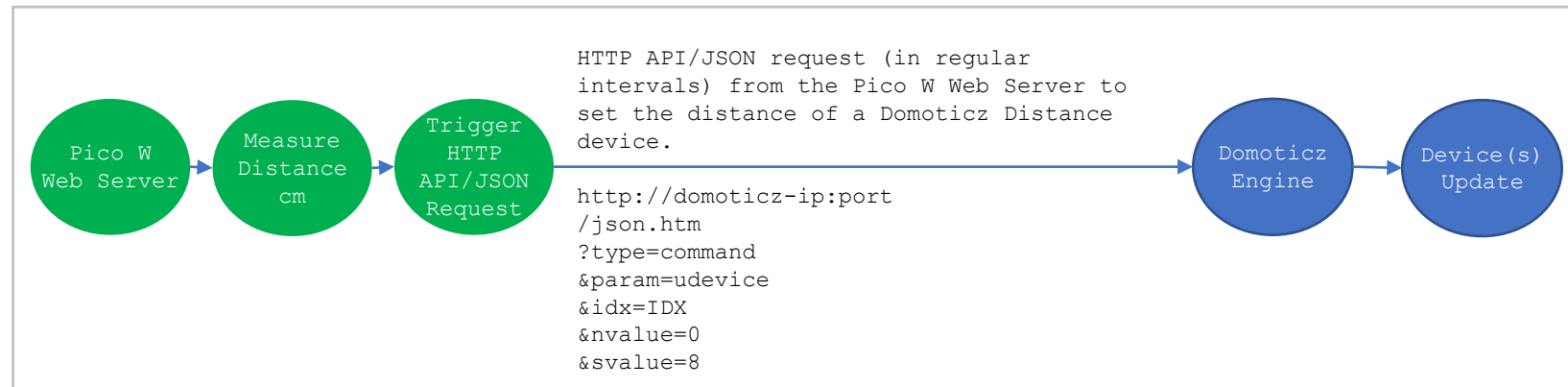
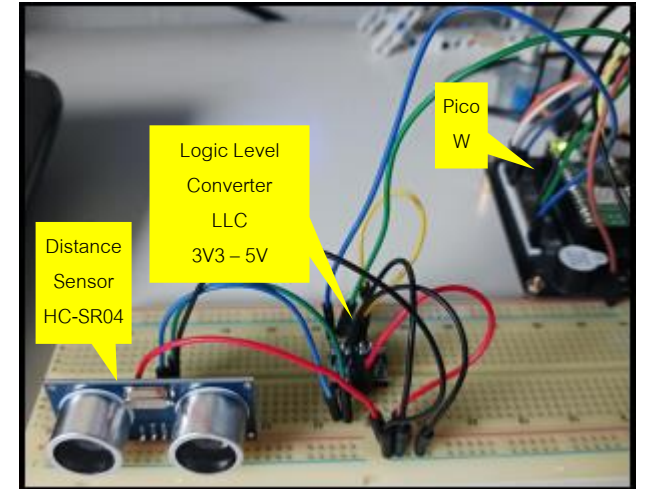
Idx	Name	Enabled	Type	Address	Port	Data Timeout
3	VirtualSensors	Yes	Dummy (Does nothing, use for virtual switches only) <a href="#">Create Virtual Sensors</a>			Disabled

## Domoticz Devices

Idx	Hardware	ID	Unit	Name	Type	SubType	
44	VirtualSensors	00082044	1	Distance	General	Distance	8.0 cm

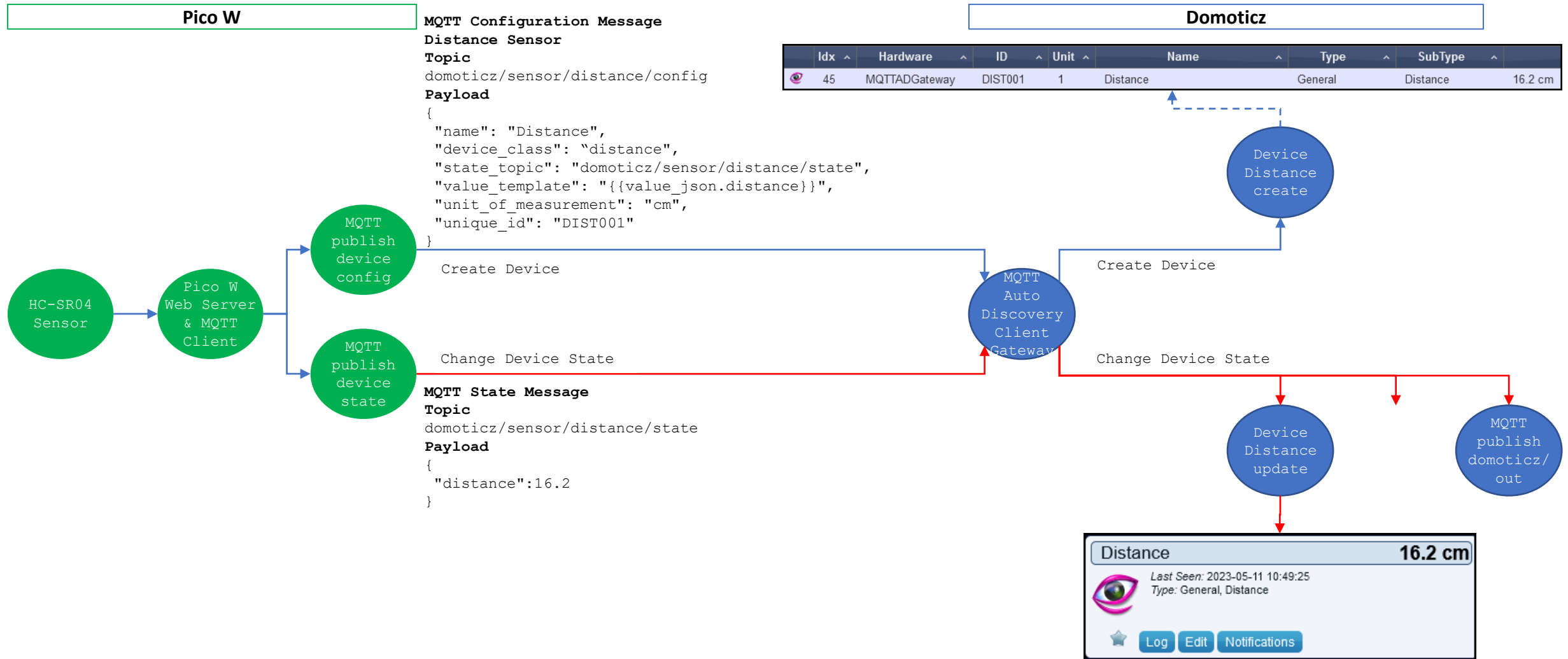


## Pico W with Distance Sensor HC-SR04 and LLC



# Project Distance Sensor - MQTT Autodiscover

*Pico W publishes MQTT Config & State Messages to Domoticz MQTT Auto Discovery Client Gateway.*





# Project IKEA VINDRIKTNING Air Quality Sensor – HTTP API/JSON

*IKEA VINDRIKTNING Air Quality sensor sends PM 2.5 ug/m3 values to the Pico W to update Domoticz Custom Sensor.*

## Domoticz Hardware

Idx	Name	Enabled	Type	Address	Port	Data Timeout
3	VirtualSensors	Yes	Dummy (Does nothing, use for virtual switches only) <a href="#">Create Virtual Sensors</a>			Disabled

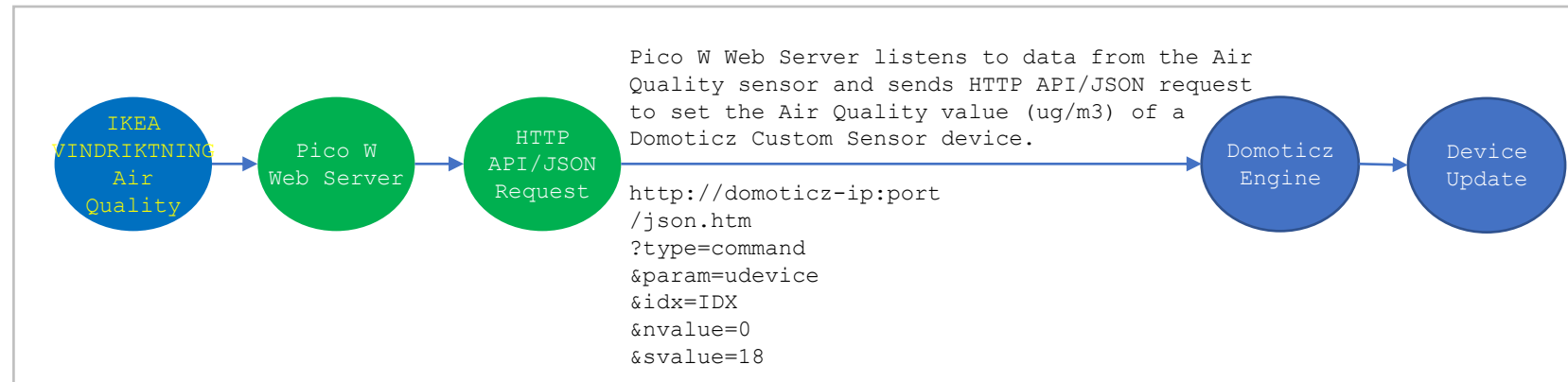
## Domoticz Devices

Idx	Hardware	ID	Unit	Name	Type	SubType	
46	VirtualSensors	00082046	1	Air Quality	General	Custom Sensor	18 ug/m3

**Air Quality** **18 ug/m3**

Last Seen: 2023-05-15 10:48:34  
Type: General, Custom Sensor

[Log](#) [Edit](#) [Notifications](#)



## Pico W with IKEA VINDRIKTNING Air Quality sensor

