Setting up the ThetaMon-Probe

We use Visual-Studio Code together with the PlattformIo Plugin.

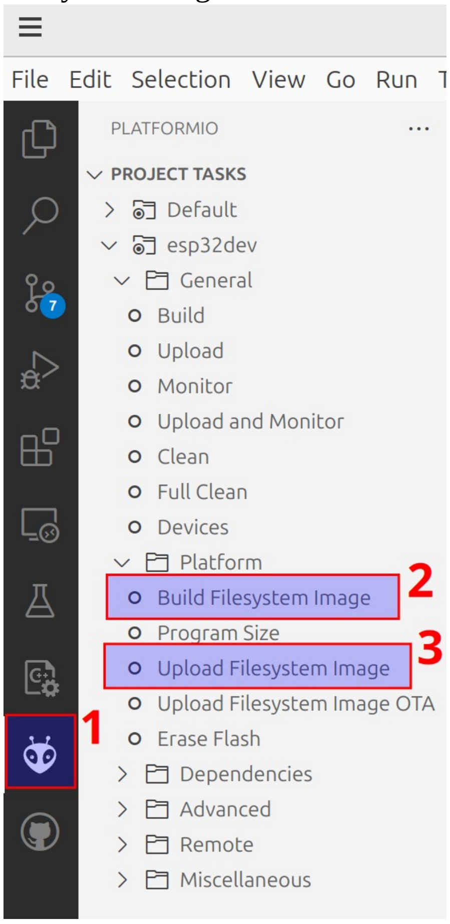
1. Clone the project and open the directory „ThetaMon3/Probes/ThetaProbe/“ in VSC. Build the firmware and flash it to the ESP32 board.

2. Prepare a file, named „MqttConfig.txt“ and place it in the „data“ directory. The content should be following:

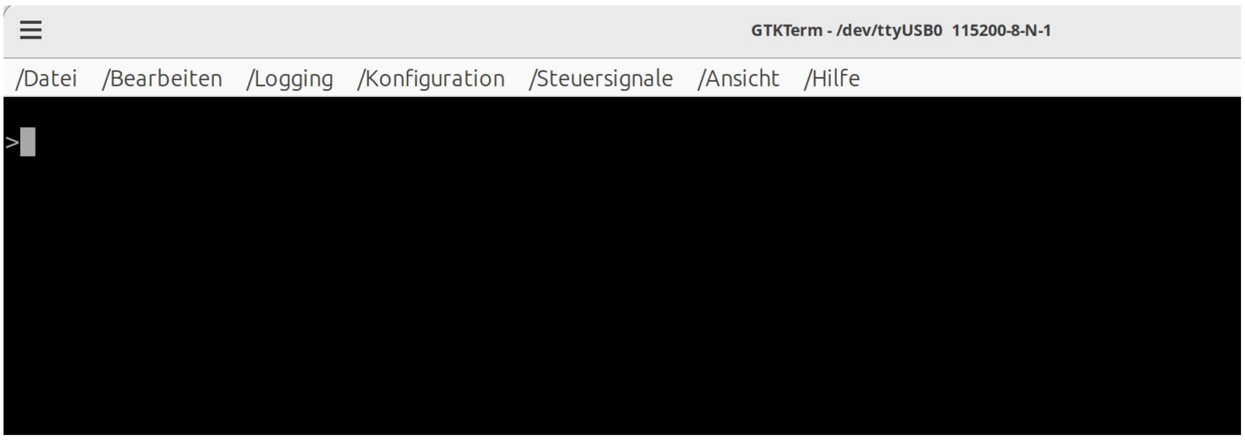
   ```
   setMqttHost 192 201 273 83  <-- Your Mqtt broker IP (use spaces instead of dots!)
   setMqttPort 1883            <-- Your Mqtt broker Port
   setMqttSpot "LivingRoom"    <-- Name of the place, where the probe is placed
   startWifi "<Your-SSID>" "<Your-SSID-Passphrase>"
   ```

   The IP-Address must be given with blanks, instead of dots.
   The Mqtt-Spot-Name, is used to connect to HomeAssistant, or alike.
   The SSID and SSID-Passphrase have to be double-quoted.

3. In Visual Studio Code, go to PlattformIO, then „Build Filesystem Image“, then „Upload Filesystem Image“

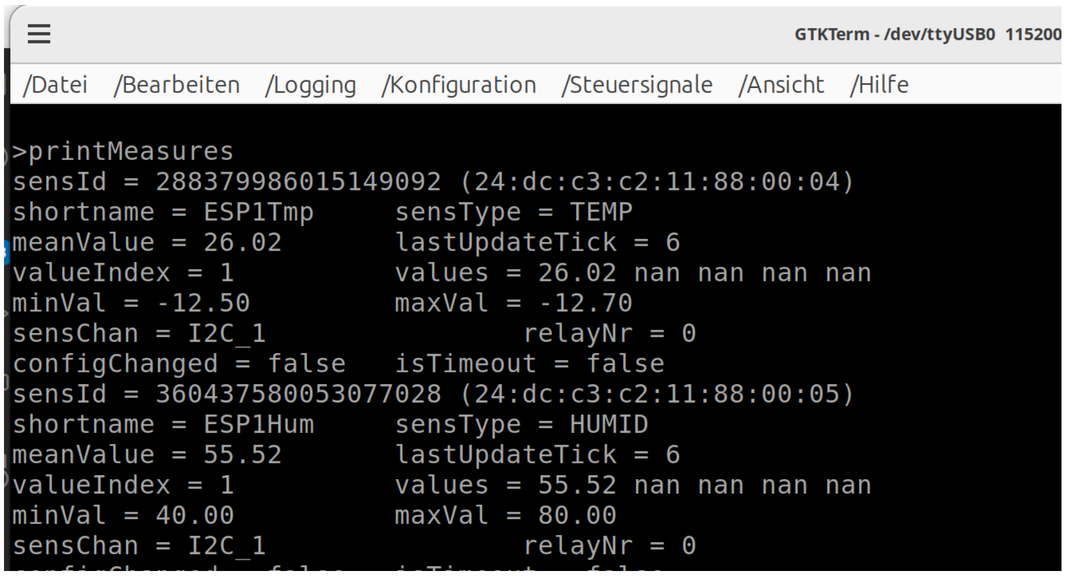


4. Open a Terminal program, like GTK-Term, connect to the ESP32-console, press Enter and you should see a command prompt.


GTKTerm - /dev/ttyUSB0  115200-8-N-1
/Datei /Bearbeiten /Logging /Konfiguration /Steuersignale /Ansicht /Hilfe

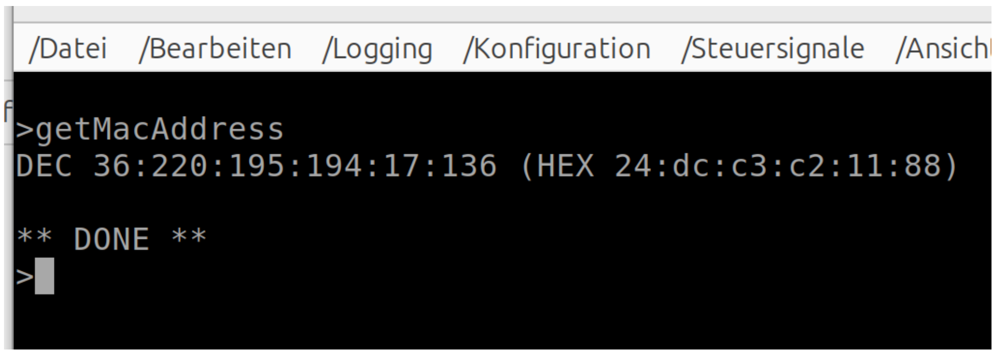
```
>
```

5. Type „printMeasures“, you should see data of the connected sensors.


GTKTerm - /dev/ttyUSB0  115200
/Datei /Bearbeiten /Logging /Konfiguration /Steuersignale /Ansicht /Hilfe

```
>printMeasures
sensId = 288379986015149092 (24:dc:c3:c2:11:88:00:04)
shortname = ESP1Tmp      sensType = TEMP
meanValue = 26.02        lastUpdateTick = 6
valueIndex = 1           values = 26.02 nan nan nan nan
minVal = -12.50          maxVal = -12.70
sensChan = I2C_1                 relayNr = 0
configChanged = false    isTimeout = false
sensId = 360437580053077028 (24:dc:c3:c2:11:88:00:05)
shortname = ESP1Hum      sensType = HUMID
meanValue = 55.52        lastUpdateTick = 6
valueIndex = 1           values = 55.52 nan nan nan nan
minVal = 40.00           maxVal = 80.00
sensChan = I2C_1                 relayNr = 0
```

The BME280 Sensor has three entries for Hummidity, Air-Pressure and Temperature. Your printing may look different, because the sensor-Ids are not yet registered in “ID_Table_U64.txt”. The first six segments of the Id are the Mac-Address of your ESP device.
You could get the Mac-Address of your device, simply by typing „getMacAddress“ into the console:


/Datei /Bearbeiten /Logging /Konfiguration /Steuersignale /Ansich

```
>getMacAddress
DEC 36:220:195:194:17:136 (HEX 24:dc:c3:c2:11:88)

** DONE **
>
```

6. Open „ID_Table_U64.txt“ in the data-directory. You see a collection of commands, that set up the sensor configuration. Every line can also be typed into the console.
   Example line:

   setSensId 288253003812035532 -12.5 -12.7 0 0 "WohnzTmp"

   „setSensId“ is the command to define sensor attributes, followed by a hash-value of the sensor-address (more to this below).

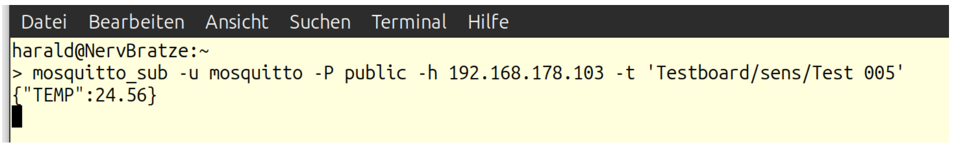
   -12.5: the lower threshold, where the relay-channel is switched on.

   -12.7: the upper theshold, where the relay-channel is switched off,

   0: (the first one) is the sensor-type.
   0 – Temperature
   1 – Humidity
   2 – Pressure
   3 – Relay (We treat Relays as sensor, to read their states)

   0: (the second one) defines the relay-channel, that this sensor is bound to.
   0 – No channel. This sensor doesn’t switch anything.
   1 – Channel 1.
   2 – Obviously, Channel 2.

   Refer to the schematic, connector J2 to see, how to connect the corresponding relay-channels are to be connected. I use two-channel Arduino rely-boards
   .
   "WohnzTmp": is the short Name of the sensor. It must not be longer than eight characters. This name is used together with the location-name in „MqttConfig.txt“ to build the Mqtt publishing name, in example “Testboard/sens/WohnzTmp” (note the ‚sens‘ in the middle!). If you have mosquitto installed at your machine, the following command shows the message updating the sensor data



```
harald@NervBratze:~
> mosquitto_sub -u mosquitto -P public -h 192.168.178.103 -t 'Testboard/sens/Test 005'
{"TEMP":24.56}
```

   > mosquitto_sub -u mosquitto -P public -h 192.168.178.103 -t 'Testboard/sens/Test 005'

   Please note, that this message is sent only every five minutes.

7. If you now type „printMeasures“ again, you will be shown the actual setting. Bring the hashes into your Id-Table-file, give every sensor a shortname and if you need it define the switching thresholds and the relay-binding.

   To get out, which sensor belongs to which address, simply warm them up with your hands and print the measures again. Never ever use a lighter, this will destroy the sensor!

8. Save this file into the data directory, build the file-system again and upload it. Reboot your ESP. Remember to close your terminal-connection before connecting with Visual Studio.

9. Go back to your console-connection. Check that every sensor is recognized. Type „reboot“ and watch the output. If a sensor was set up correct, you get something like:

   ```
   >setSensId 4179915027513875496 -12.5 -12.7 0 0 "INNEN_O2"
   UpdateConfig done.

   ** DONE **
   ```

   otherwise:

   ```
   >setSensId 14484146576875228968 -12.5 -12.7 0 0 "Test 012"
   SensId ignored: 14484146576875228968 (28:8f:56:45:92:06:02:c9)

   ** DONE **
   ```

   Of course you could set all of your sensors into a single file and download it to all of your probes. Not connected sensors will simply be ignored.

10. If everything went well, you are done. The probe will send the actual values every five minutes to your mqtt-broker.

    There are some python tools, you could have a look at.