Installation of dev-Environment for Sensormodule

Only tested on Windows 10!

Download and install git from https://git-scm.com/downloads with default options

Download and install visual studio code from https://code.visualstudio.com/download (User installer, 64 bit)

Start visual studio code

Go to extensions (Ctrl-Shift-X)

Enter "platformio" in search field

Install "PlatformIO IDE" extension

Wait until installation is finished, do the necessary reload window afterwards (may take some time)

Click on the new PlatformIO-Icon on the left 

In "Quick Access", choose open

In the new "PIO Home" tab, click on "New Project..."

In the upcoming dialog, provide the name "Test", Board "Sparkfun SAMD21 Dev Breakout", Framework "Arduino" and Location "Use default location"

Click "Finish" and wait until finished. Visuals Studio Code will open the newly created project afterwards. The new project is just used to create default envoronment and can be deleted afterwards.

Click again the PlatformIO Icon 

Again "Quick Access" appears, click "Miscellaneous->PlatformIO Core CLI"

A new terminal (within Visual Studio Code) appears, the path is home of the new test project. We don't need the test project, it was just used to create all necessary path for development. From now on we work in this terminal window:

```
cd ..
```

You should be now in a directory ending with ...\Documents\PlatformIO\Projects

```
pio lib -g install 805
pio lib -g install 166
pio lib -g install 31
pio lib -g install 5449
```

These commands should install following libraries:
"ClosedCube_HDC1080" Library
"Adafruit_BME280" Library
"Adafruit_Sensor.h" Library
"SparkFun_SCD30_Arduino_Library" Library

```
git clone https://github.com/mumpf/knx.git
git clone https://github.com/mumpf/knx-common.git
git clone https://github.com/mumpf/knx-logic.git
git clone https://github.com/mumpf/knx-sensor.git
cd knx
git checkout release
cd ..\knx-sensor
code Sensormodul.code-workspace
```

Now a new instance of Visual Studio Code is started. You can close the other (previous) instance.

If you use the board from MASIFI version v1 or v2, you need to change one or two settings:

```
In knx-sensor, edit the file platformio.ini:
- change the line
        -DBOARD_MASIFI_V3
  to
        -DBOARD_MASIFI_V2
  or
        -DBOARD_MASIFI_V1

- there exist different versions with CRYSTALLESS setting.
  Ensure that
        -DCRYSTALLESS
  is always
        ;-DCRYSTALLESS
  or the line is removed.
```

Press Ctrl-Shift-B, select the "**Build PlatformIO** knx-sensor" build task and press enter.

Now the compiler starts, this may take a while, there will be many yellow warnings, they can be ignored.

At the end, there should be a message like

```
Linking .pio\build\build\firmware.elf
Building .pio\build\build\firmware.bin
Checking size .pio\build\build\firmware.elf
Advanced Memory Usage is available via "PlatformIO Home > Project Inspect"
RAM:   [==        ]  22.0% (used 7216 bytes from 32768 bytes)
```

```
Flash: [======    ]  55.7% (used 145892 bytes from 262144 bytes)
============================ [SUCCESS] Took 34.60 seconds ======
```

Now you successfully build the Firmware for the Sensormodule, containing up to 80 logic channels.

There is also a precompiled version of the firmware available on github for the different module versions.

How to upload the Firmware to your Hardware

Connect your device via USB to your PC

Open (again) the file Sensormodul/src/Sensormodul.cpp

Press Ctrl-Shift-B, select "**Upload USB** knx-sensor" build task and press enter.

Wait until file is uploaded.

If you use a precompiled version of the firmware, you can upload it with the following command (installed PlatformIO is still needed):

```
C:\Users\<username>\.platformio\packages\tool-bossac\bossac --info --debug
--port "<COM9>" --write --verify --reset --erase -U true
firmware_masifi_v<x>.bin
```

Of course you have to replace <username>, <COM9> and <x> accordingly.

How to build a knxprod for this firmware

Open https://github.com/mumpf/multiply-channels/releases

Download the newest release of multiply-channels, currently it is the first final.

The executable is MultiplyChannels.exe

Save it to C:\Users\<username>\bin (usually you have to create bin directory)

If this is not your ETS-PC, install ETS5 on this PC (ETS5.7.x demo is sufficient, even any 5.6.x should do)

Go to the Visual Studio Code instance, which is containing the knx-sensor project

Press Ctrl-Shift-P, enter "run test task" and click the appearing "Tasks: Run Test Task"

In the following dropdown select "**MultiplyChannels-Release** knx-sensor"

Wait for the success message in the terminal window

The freshly build

- Sensormodul-v1.4-10.knxprod
- Sensormodul-v1.5-20.knxprod

- Sensormodul-v1.6-40.knxprod
- Sensormodul-v1.7-80.knxprod

you will find in the release directory of the knx-sensor project

You can import this knxprod in your ETS (minimum 5.6) like any other knxprod.

Programming with ETS

This works the same way as with all other KNX devices. For the initial programming you should program the physical address (PA) first, then transfer the application program (do not use ETS function "PA + Application program").

Afterwards you can use partial programming as usual.