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

```
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-wire.git
git clone https://github.com/mumpf/knx-sensor.git
```

```
cd knx
git checkout release
cd ..\knx-common
git checkout release
cd ..\knx-logic
git checkout release
cd ..\knx-wire
git checkout release
cd ..\knx-sensor
git checkout release
code Sensormodul.code-workspace
```

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

The current board version from MASIFI is v3.1. The current version for use outdoor ist v1.3. Due to the fact, that I have to test the release with diferent versions, it might happen, that the firmware is released for any of the tested versions v2, v3 or v3.1.

Please ensure always that the released version fits to your hardware! To do this, do the following: Find the version of your hardware board (v1, v2, v3 or v3.1). Or - if it is the outdoor module - it is v1.3.

```
In knx-sensor, edit the file platformio.ini:
- there is a line
        -D BOARD_MASIFI_V...
- change the line to the according version of your hardware
        -D BOARD_MASIFI_V31
  or
        -D BOARD_MASIFI_V3
  or
        -D BOARD_MASIFI_V2
  or
        -D BOARD_MASIFI_V1
  or (for the outdoor module)
        -D BOARD_MASIFI_AUSSEN_V13

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

With this firmware you get watchdog-support. With the setting -D WATCHDOG the watchdog functionality is enabled. The default is ;-D WATCHDOG which disables watchdog functionality. 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 a ETS configurable knx stack, a logic module with 80 logic channels, a one wire module for up to 30 one wire sensors and a sensor module for up to 7 measurements (temperature, humidity, air pressure, voc value, co2 value, brightness and distance).

Precompiled firmware versions are not released anymore, you have always to compile your own.

How to upload the Firmware to your Hardware

Connect your device via USB to your PC. If you are using sensormodule v3.1, you need to connect it to the KNX bus.

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.

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 version 2.1.2.

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-v3.x.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. There is no need anymore to program the physical address first or to transfer the complete application program after initial programming.

Simply use the "Program"-Button or any programming way you are used to and ETS will do the correct thing in the fastest way possible.