

Update procedure for a new firmware or application version

Only tested on Windows 10!

This description is just valid, if you successfully built and installed a firmware and application the first time according to the [dev setup](#) instructions.

Open Visual Studio Code. It opens with the last project you used.

In case the opened project is not the sensormodule project, open "Sensormodul (Workplace)", you will find it in File->Open Recent menu.

Press Ctrl-Shift-G (Opens Source Control).

Below "SOURCE CONTROL PROVIDERS" you will find all projects necessary for the firmware:

- knx-sensor
- knx-logic
- knx-common
- knx

Do for each of them the following:

Click on the project name (i.e. knx-sensor).

There is an additional area called "KNX-SENSOR" (in uppercase letters), having a text box below with a text "Message". If you hover the mouse over this additional area, some command icons appear, followed by 3 dots (...) indicating a menu. Click on these 3 dots.

In the upcoming menu click on the topmost entry "Pull".

As said, do this for each project.

As soon as all 4 pulls are finished, continue with the build steps from the initial documentation:

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. In case you have already the newest version downloaded, you can skip this and the following copy step.

The executable is MultiplyChannels.exe

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

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.

Application update procedure is described in the Applicationdescription Sensor.