Getting Started SCμM (Single Chip Micro-Mote)

Table of Contents

Preface

0.1 Software versions used

Keil µVision MDK-ARM: 5.36
https://www.keil.com/demo/eval/arm.htm

Python: 3.10.5
https://www.python.org/downloads/release/python-3105/

Pip: 22.2.1
https://pypi.org/project/pip/

Pyserial: 3.5
https://pypi.org/project/pyserial/

PuTTY: 0.77
https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html

0.2 Hardware used

Sulu V2 (Board)

SCµM (Processor on Sulu Board)

Nordic Semiconductor nRF52840-DK
https://www.nordicsemi.com/Products/Development-hardware/nrf52840-dk

DSD TECH SH-U09C5 USB to TTL UART Converter
https://www.amazon.com/DSD-TECH-SH-U09C5-Converter-Support/dp/B07WX2DSVB

0.3 Code

SCµM Test Code
https://github.com/PisterLab/scum-test-code

SCµM Programmer
https://github.com/filmak/SCuM-programmer/blob/develop_12/test_bootload.py

Sulu Programming With nRF Setup (Needed only for custom nRF boot configuration)
https://crystalfree.atlassian.net/wiki/spaces/SCUM/pages/1901559821/Sulu+Programming+With+nRF+Setup

J-Link (BOOTLOADER mode)
https://www.nordicsemi.com/Products/Development-hardware/nrf52-dk/download#infotabs

Documentation

1.0 Hardware

1.1 Hardware Setup nRF52840-DK Board

The “nRF POWER SOURCE” switch needs to be set to “USB SOURCE”, the “Power” switch needs to be set to “ON” and “nRF only|DEFAULT” should be switched to “DEFAULT”.

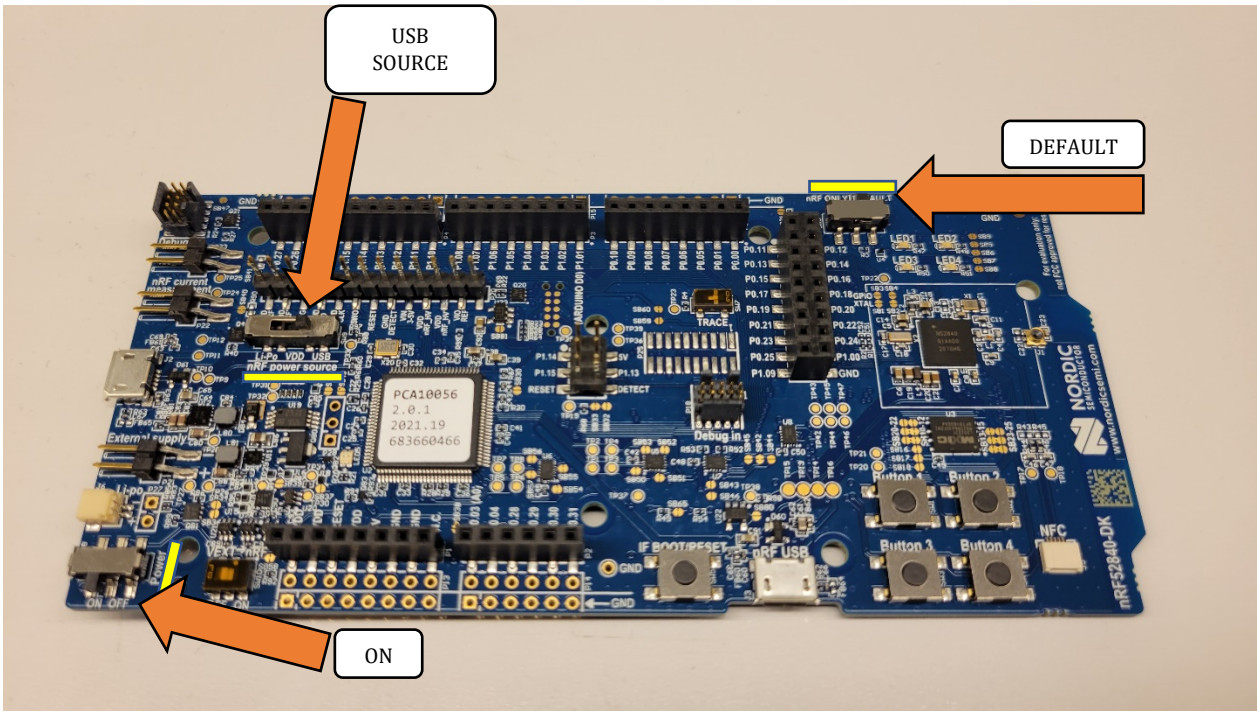

Figure 1: nRF52840-DK

The bottom micro-USB titled “nRF USB” is for powering the board and the micro-USB on the left is for the J-Link communication. Use both ports to connect to your PC. When connected, a J-Link drive should show up in your File Explorer (Do not connect to SCµM board until after all of the software is setup).

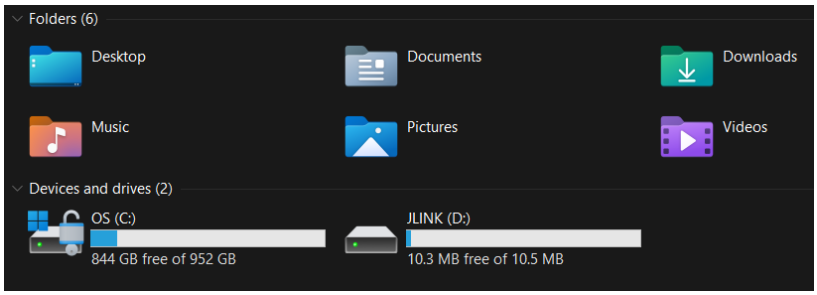

Figure 2: J-Link in File Explorer

If the J-Link drive doesn’t show up you might have to do a custom boot. A helpful resource for this is under section 0.3 Code Sulu Programming With nRF Setup. However, I would first try to enter “BOOTLOADER” mode by holding down “IF BOOT/RESET” button of nRF board while powering on. Then add J-Link file(.bin) to “BOOTLOADER” drive. If “BOOTLOADER” drive doesn’t appear, you most likely need to do the custom boot.

1.11 nRF52840-DK Custom Boot

You first need to install the SEGGER Embedded studio IDE for ARM (Version 5.70a is recommended). Then you need to download this git repository. Next, open the .emProject

found at “SCuM-programmer/scum-programmer/scum-programmer.emProject” in the repository you just downloaded. You need to make sure the “CMIS 5 CMIS-CORE Support Package” is installed. The Package Manager window is found under the “Tools” tab. If you’re installing it for the first time you need to close SEGGER Embedded Studio after installing and then re-open it for the package to work.

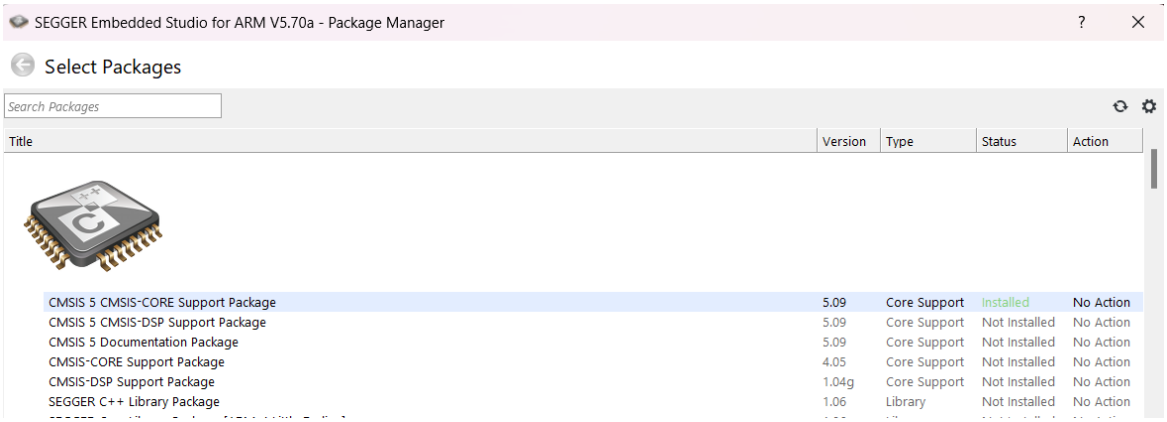

Figure 3: Package Manager with CMIS 5 CMIS-CORE Support Package installed

Now you can build the project by hitting the “Build scum-programmer” button under the Build tab. While the nRF is on and connected to the computer you first press the “” Connect J-Link” button under the “Target” tab. Then, in the same Target tab, you press the “Download scum-programmer” button. If done correctly the nRF LED’s(1-4) in the top right corner should light up like in the GIF on the GitHub.

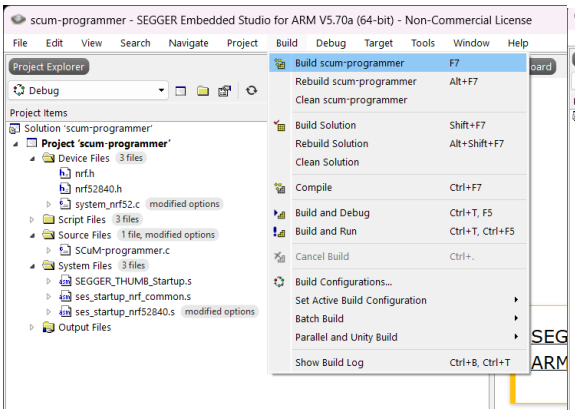

Figure 4: Building the scum-programmer Project

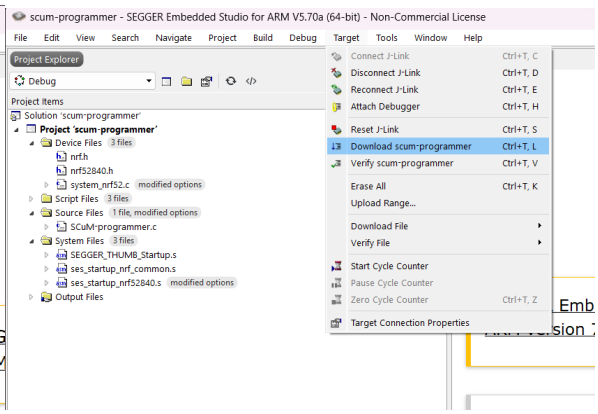

Figure 5: Already Connected to J-Link, Programming the nRF

Now that the J-Link has shown up, we can find the COM port by opening the Device Manager in the Windows Control Panel. You can find the Device Manager by either searching for “Device Manager” in the Windows search bar or going to the Control Panel then “Hardware and Sound”.

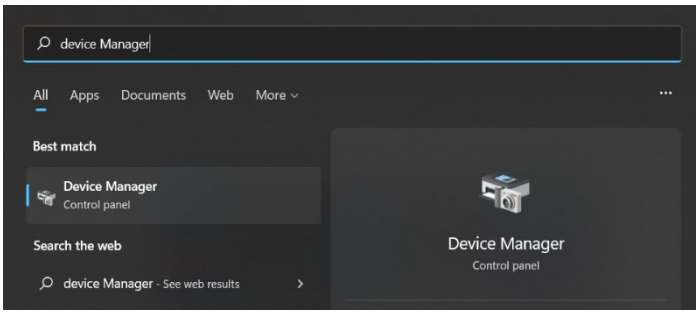

Figure 6: Windows Search Device Manager

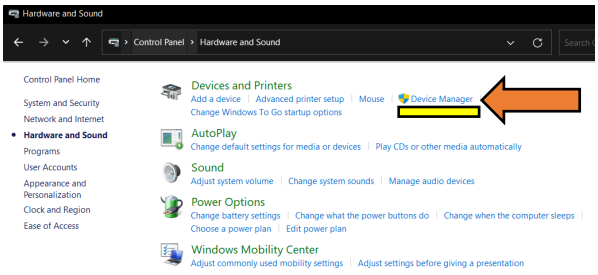

Figure 7: Control Panel to Device Manager

The port number should be under “Ports (COM & LPT)”. The number next to COM needs to be added to the test_bootload.py file. This will be done later in section 2.4, Software Setup nRF52840-DK Board.

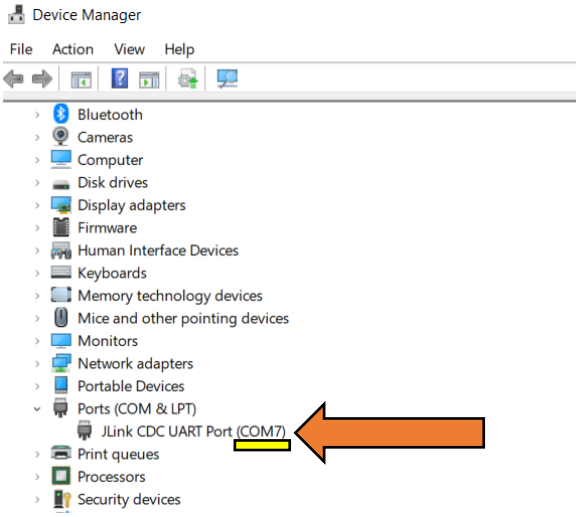

Figure 8: COM Port in Device Manager

1.2 FTDI TTL UART Converter

Place the jumper clip over the "1V8" pins like in figure 9. Do not connect to SCµM board until after all of the software is setup.

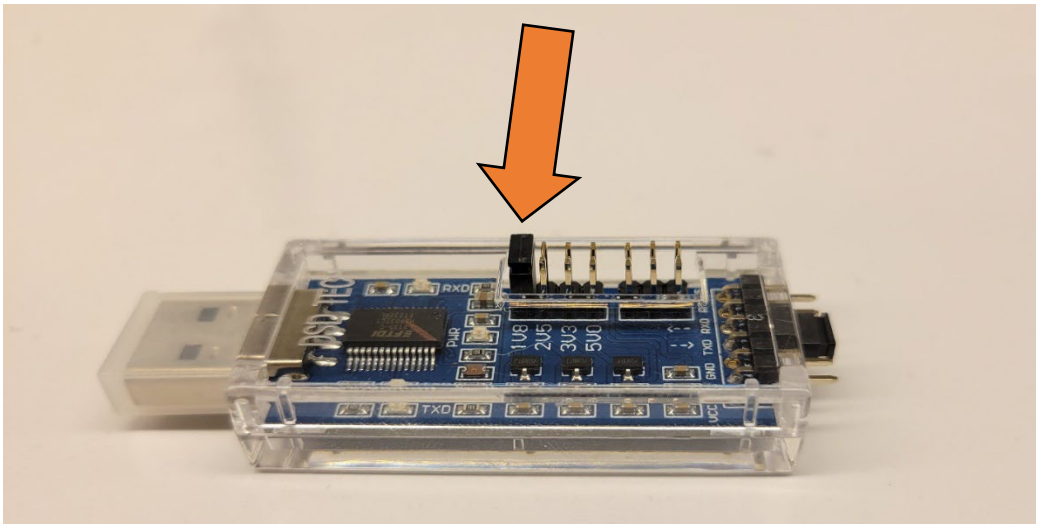

Figure 9: SH-U09C5 USB to TTL UART Converter Set to 1.8V

2.0 Software

2.1 Installation of Keil µVision

Keil µVision can be downloaded from the link provided in "Software versions used". However, check the SCµM Test Code GitHub for which versions are compatible with SCµM. You can follow the link to find out how to download older versions of Keil (Tutorial). Any version using compiler version 5 should work.

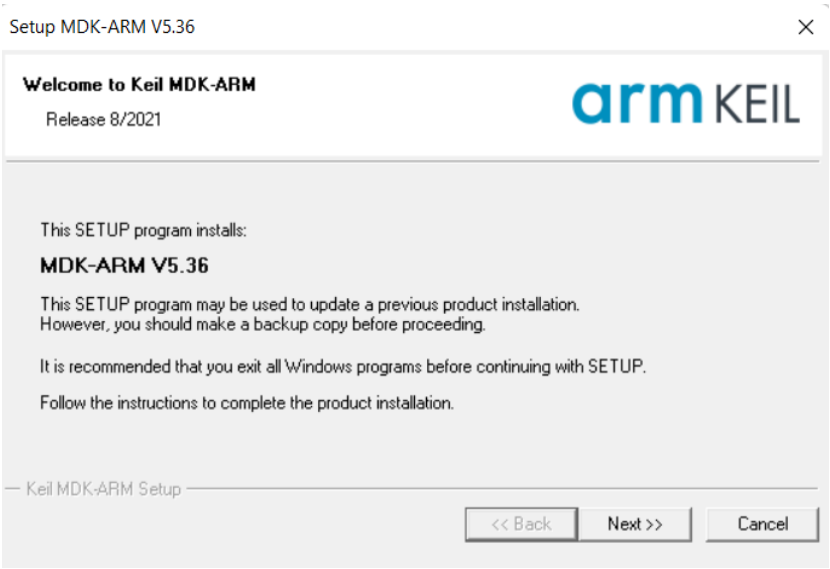

Figure 10: MDK-ARM V5.36 Installation

2.2 Installation of python

Make sure "Add Python (version) to PATH" box **is checked**. Pip is installed with "Install Now" but, ensure pip box is checked when doing "Customize installation". Older versions of Python might not have pip included in the installation. If that's the case then you have to install pip manually.

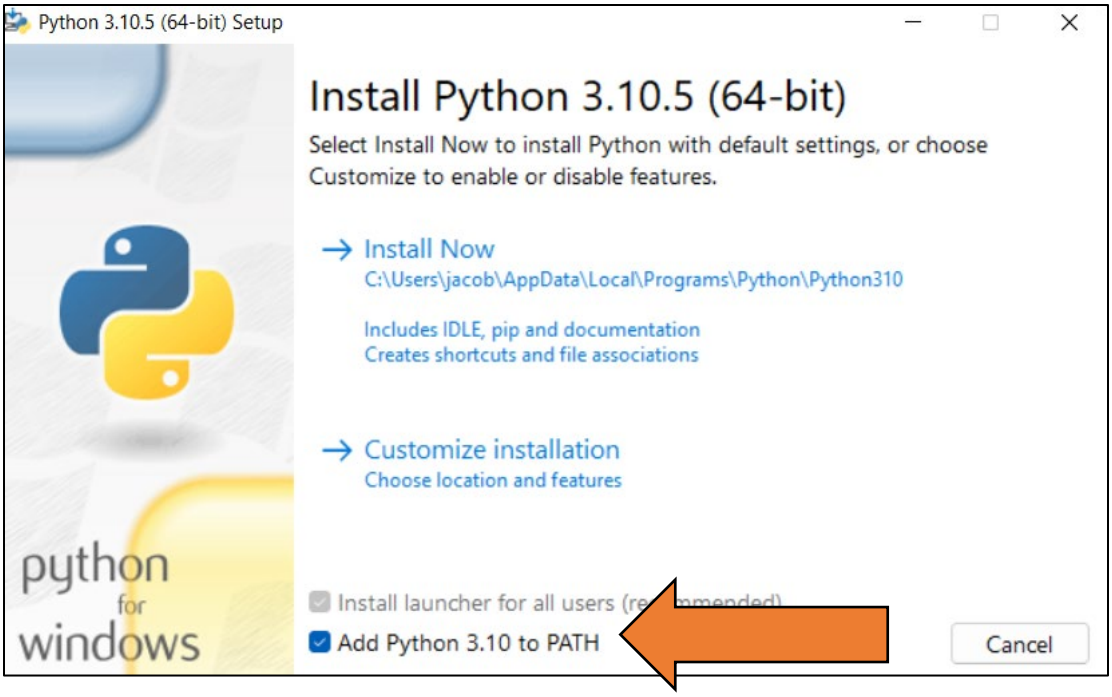

Figure 11: Python3.10.5 Installation Showing "PATH" Box Checked

In the Windows Command Prompt type "python" into the command line to test if it was installed correctly (make sure it's the right version). You might need to go to the Microsoft Store and download Python 3.10 if the Windows Command Prompt sends you there when you type python (the test_bootload.py file needs to be run in Windows Command Prompt).

Figure 12: Windows Store Python 3.10

Next, pyserial needs to be installed. This can be done by typing "pip install pyserial" into the Windows Command Line. More info on pyserial can be found at the link.

2.3 SCµM Test Code

Download SCµM Test Code from GitHub link.

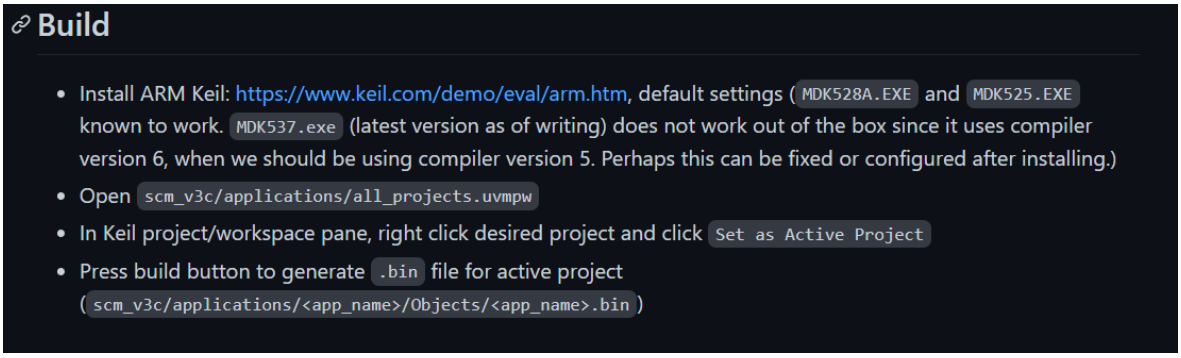

Figure 13: SCµM Test Code: Build

All the projects are in "all_projects.uvmpw" file. This can be found in the downloaded folder "scum-test-code-develop/scm_v3c/applications/all_projects.uvmpw". Opening this file will open the Keil µVision application. Within Keil, right click on "Project: hello_world" and set as active project. Open the hello_world.c file under "Project: hello_world/hello_world/app/hello_world.c". Then, hit the build button in the top right corner.

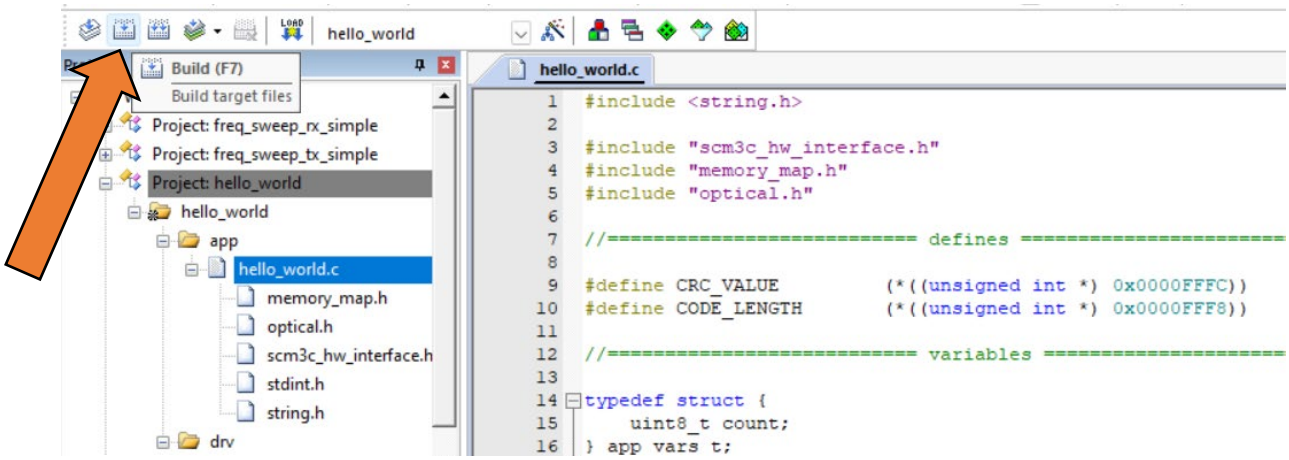

Figure 14: Building hellow_world.c in Keil µVision

This should create a .bin file that will be used for the test_bootload.py. The bin file can be found in the downloaded folder "scum-test-code-develop/scm_v3c/applications/hello_world/Objects /hello_world.bin". All the warnings can be ignored. By hitting the build button a second time it will finish building (warnings can also be disabled in the c/c++ tab under project options).

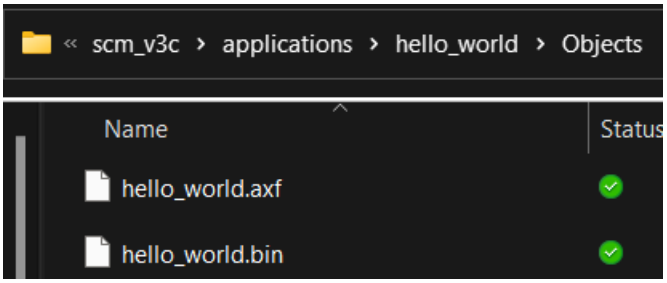

Figure 15: hello_world.bin file in File Explorer

2.4 Software Setup nRF52840-DK Board

The nRF board needs to be flashed to output 1.8V. Copy and paste SCµM Programmer code into a new file named "test_bootload.py". The only things that need to be changed in this file are lines 12 and 13. The line 12 code should say "nRF_port=COM27". The COM number needs to be replaced with the COM number we got in section 1.1. Line 13 should start with "binary_image=". This needs

to be the path to the .bin file created in section 2.3. Make sure the path uses forward slashes instead of backlashed. Whenever you change projects, you have to change the path to where the new .bin file will be before running the test_bootload.py program. The COM port number also needs to be changed when using a different hardware setup.

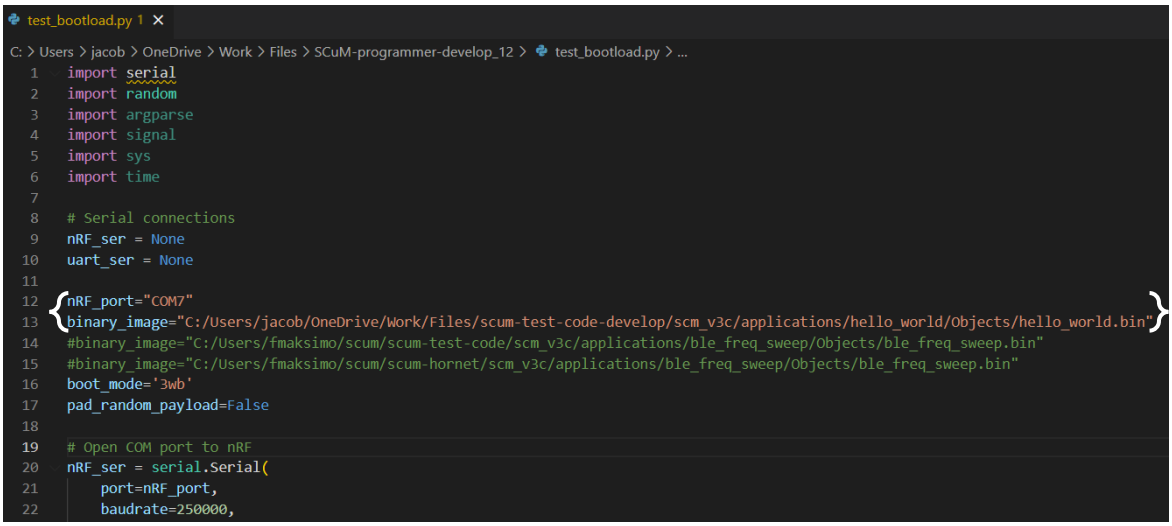

Figure 16: Modified test_bootload.py File

Next, we need to run the modified test_bootload.py file. Find where you have the test_bootload.py file saved and right click on it. Go to “Properties” and copy the location.

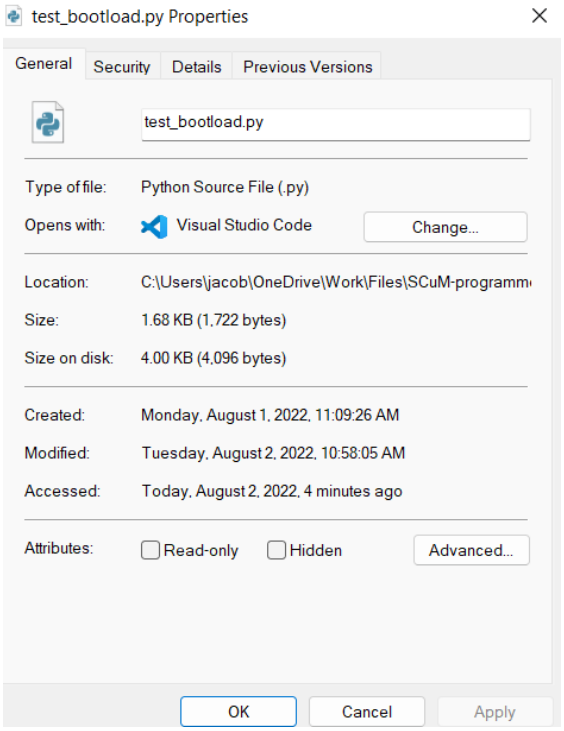

Figure 17: test_bootload.py Properties

Make sure you have both micro-USB cables connected to the nRF board and the J-Link drive is available. Open the Windows Command Prompt and type “python” then paste the location you got from the properties window. Make sure the end of the location has “\test_bootload.py” at the end. If not, type it in. After hitting enter you should see “SRAM load complete” and “3WB bootload complete” if done correctly. After having everything wired up later on, you will run this same

command to actually program the SCµM chip. However, double check if the nRF boad is actually outputting 1.8V before hooking it up to the SCµM board. If you aren't getting 1.8V you will probably have to do a custom boot.

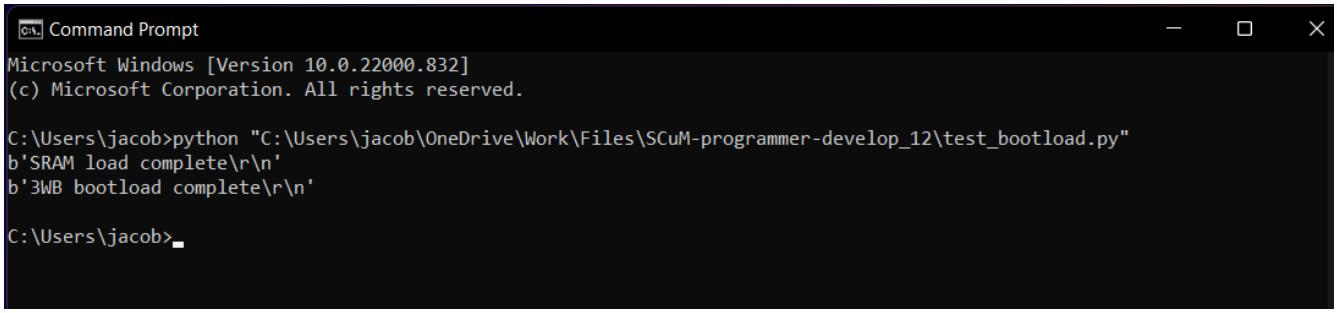

Figure 18: Windows Command Prompt Runing test_bootload.py

2.5 PuTTY Setup

Putty will be used as the serial monitor. We need to know the COM port for the FTDI(for UART). The COM port can be found the same way as in section 1.1.

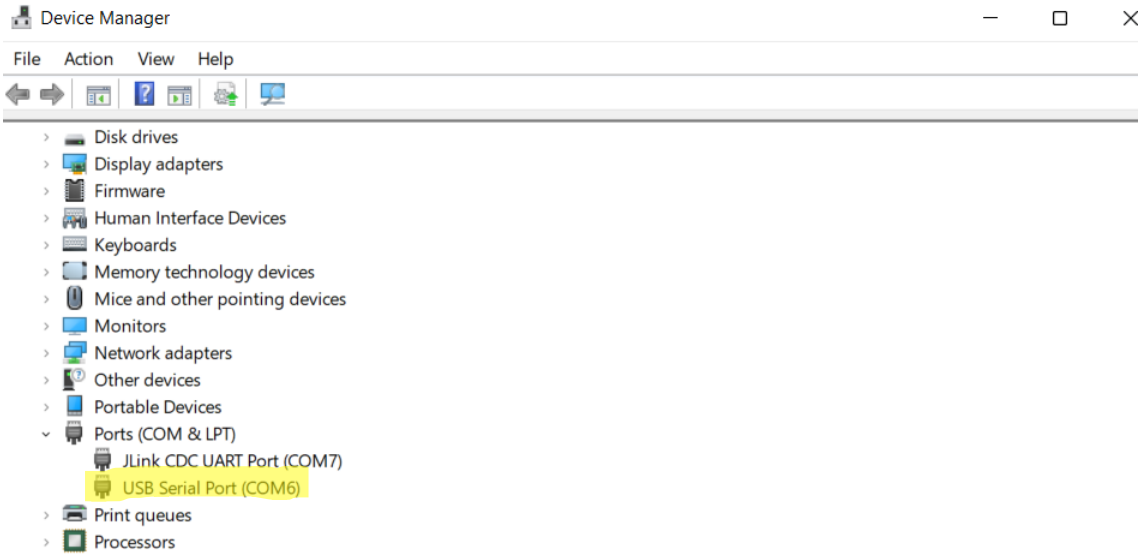

Figure 19: FTDI UART COM Port

After Downloading and installing PuTTY we have to set it up for the SCµM board. The first thing that needs to be changed is the "Connection type". This needs to be set to "Serial". The speed should be around 19200. However, this can fluctuate at bit which is why PuTTY is recommended because you can fine tune the speed (baud rate). You have to enter the COM port that corresponds to the UART communication device under "serial line". When done you can save these settings and then hit open and a blank terminal should open up. You should have the serial monitor open before running the test_bootloader.py script.

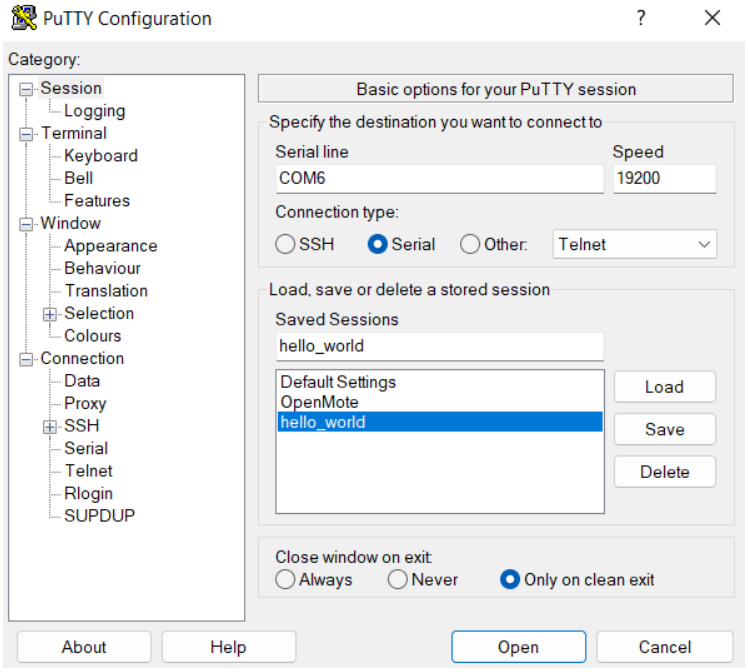

Figure 20: PuTTY Configuration

3.0 Connections

3.1 Sulu to nRF52840-DK

Now that everything is set up you can make the connections to the Sulu board. After making all the connections, to program SCUM you have to run the test_bootload.py script (section 2.4).

nRF52840-DK		Sulu V2.0
5V	->	EXT_BAT
GND	->	GND
P0.31	->	HRESET
P0.30	->	3WB_EN
P0.29	->	3WB_DATA
P0.28	->	3WB_CLK

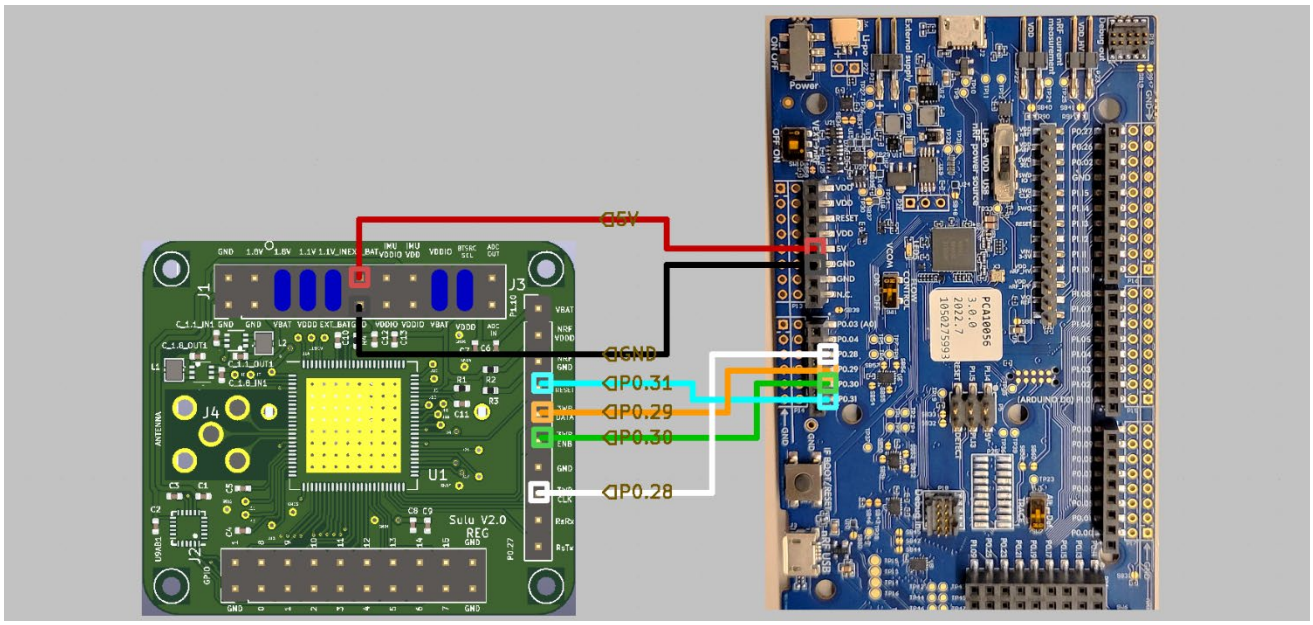

Figure 21: Sulu board connections to nRF board

3.2 Sulu to FTDI TTL UART Converter

FTDI TTL UART Converter		**Sulu V2.0**
RXD	->	RsTx
GND	->	GND