Technical Document

How to log sensor data into a file on PC via APP3.0 BLE connection

Bosch Sensortec

BOSCH

Invented for life

Table of contents

1 Introduction

Application board 2.0 (APP2.0 board) has been discontinued. Users can find more information about APP2.0 board at https://www.digikey.com/en/products/detail/bosch-sensortec/0330-AB0-111/2416313. There is a Bluetooth module BT33 installed on APP2.0 board. When PC is searching for Bluetooth device, it will show as “Amp’ed UP”. After pairing PC will assign a virtual COM port to BT33. Therefore, when Development Desktop 2.0 (DD2.0) GUI software is launched on PC, users can choose the virtual COM port to connect PC to APP2.0 board through Bluetooth. This also means that users can use a battery pack to power on APP2.0 board with a sensor shuttle board plugged in and then use DD2.0 software to evaluate that sensor or log that sensor data into a *.csv file wirelessly.

Currently the new application board 3.0 (APP3.0 board) is available at https://www.digikey.com/en/products/detail/bosch-sensortec/APPLICATION-BOARD-3-0/14617532. It has ublox module NINA-B302 installed. The ublox module has Nordic nRF52840 BLE embedded. Different sensor shuttle board 3.0 can be plugged onto APP3.0 board and users can connect APP3.0 board to PC through a USB cable and then use DD2.0 GUI SW to evaluate sensor(s) or log sensor data into a file. DD2.0 GUI SW doesn’t support Bluetooth low energy (BLE) connection yet at this moment. However, Bosch Sensortec provided a way to do that. Users can download COINES SW at https://www.bosch-sensortec.com/software-tools/tools/coines/ and then install it on PC. Users can find datalogger folder at C:\COINES\v2.8.8\datalogger where there are user manual document, APP3.0 firmware and Android APK file inside. This means that users can evaluate sensor and log sensor data into a file between APP3.0 board and an Android device through BLE connection. The logged data file is located inside ublox flash memory on the APP3.0 board, not inside the Android device memory. So there is file size limit for data logging.

This technical document presents another way about how to run Python code on PC to evaluate sensor that is plugged on APP3.0 bard and log sensor data into a file on PC via BLE connection. Section 2 shows the hardware setup and section 3 shows the software needed to download and install on PC. Detailed step-by-step instructions are shown in Section 4 of this document. Two pieces of Python code “scan.py” and “ble_NUS.py” are included in Appendix A and B respectively.

2 Hardware setup

The hardware includes one APP3.0 base board, one BMA400 shuttle board 3.0 as an example and a battery pack shown in Figure 1.

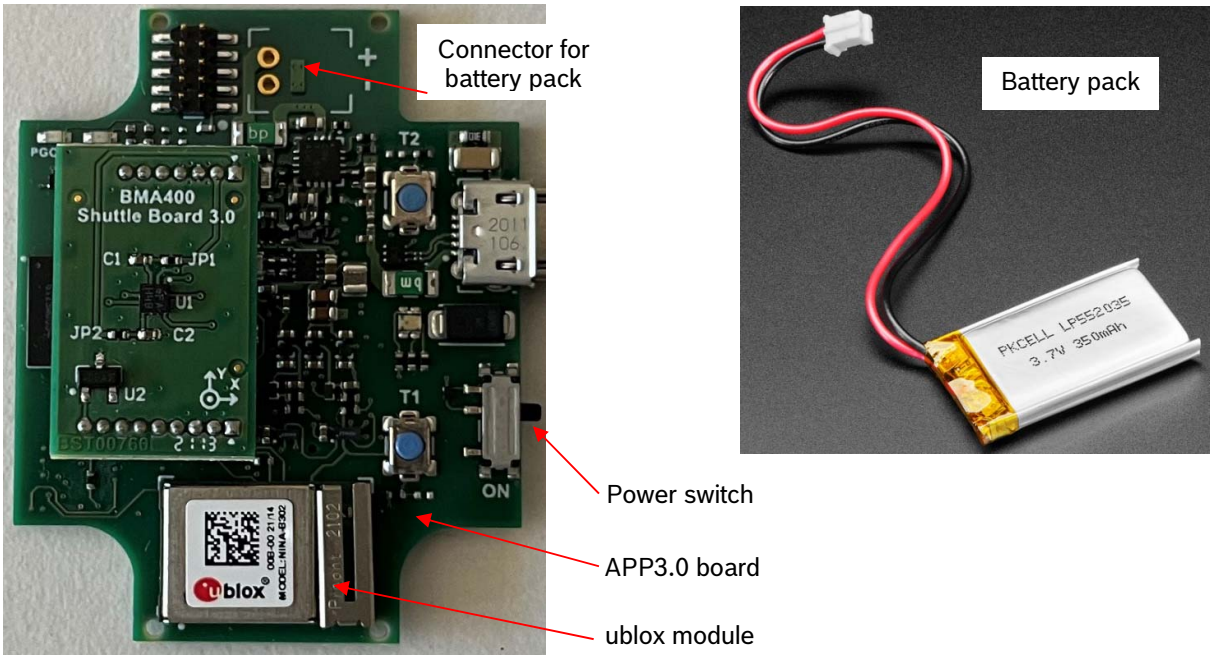


Figure 1 Hardware setup

The male connector for battery pack is available at https://www.digikey.com/en/products/detail/molex/0532540270/1952198. Users can buy it and then solder it on APP3.0 board.

The battery pack for APP3.0 board can be purchased at https://www.adafruit.com/product/2750. Users can then attach the battery pack to the bottom side of APP3.0 board for example with a piece of double stick tape. Then users can plug the battery pack female connector to the above male connector on the APP3.0 board.

Once the final firmware is downloaded to APP3.0 board from PC through a USB cable, users don't need the USB cable anymore. Users can simply switch on the power switch and run Python code on PC to log sensor data via BLE connection. After data logging is done, users can switch off the power switch. If another data logging session is needed, users can switch on the power switch again.

Since the logged data file is located on PC, the size of logged data file is up to the size of PC hard disk drive. Because BLE connection is slow there is a limit on the sampling rate when logging sensor data.

3 Software needed

In order to log sensor data into a file on PC via BLE connection, the following software is needed to download and install on PC. All the software here is free of charge to download.

(1) COINES software – it enables users to configure sensor settings at low level and log sensor data into a file on PC. There is an example folder for different sensors in C code after installation.

(2) TDM-GCC – this is the compiler to compile the C code after users modify COINES example code. The compiler can generate *.exe file that can be executed on local PC. It can also generate *.bin and *.elf binary files that are ready to download to the ublox module on APP3.0 board.

(3) ARM GNU Toolchain – this is the tool to download compiled *.bin and *.elf binary files to APP3.0 as embedded firmware. Every time when the power switch of APP3.0 board is switched on the firmware will start running automatically.

(4) Anaconda Python – this is Python environment that includes a few pre-installed packages for data scientific computing.

a. PyCharm community edition – this is a powerful and popular IDE for programming in Python. It offers many features such as code completion, debugging, and refactoring. It is a package for Anaconda Python.

b. Bleak – this is a GATT client software, capable of connecting to BLE devices acting as GATT servers. It is designed to provide a asynchronous, cross-platform Python API to connect and communicate with e.g. sensors.

c. Scan.py – this is a Python source code script. Running scan.py will find local BLE devices. After APP3.0 board is listed users can copy and paste the MAC address to ble_NUS.py code for data logging.

d. Ble_NUS.py – this is a Python source code script. After the APP3.0 MAC address is included, it can start BLE communication between PC and APP3.0 board and the sensor data will be stored in the file “BLE_receive.txt”.

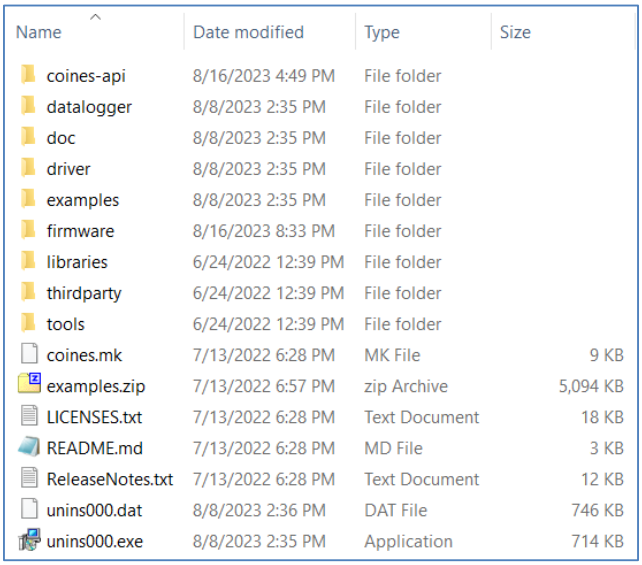
4 Getting started

The following is step-by-step instructions about how to log sensor data on PC via APP3.0 BLE connection. BMA400 shuttle board 3.0 is used here as an example.

4.1 Download and install COINES SW

COINES v2.8.8 SW can be downloaded online at https://www.bosch-sensortec.com/media/boschsensortec/downloads/software/communication_with_inertial_and_environmental_sensors_coines/v2_8/coines_external_v2-8_rc_installer.zip. Users can unzip this file and then double click "COINES_External_V2.8_RC.exe" to install the COINES SW. After installation users are able to find the folder C:/Windows/COINES/v2.8.8 as shown in Figure 2.

COINES ("COmmunication with INertial and Environmental Sensors") provides a low-level interface to APP3.0 board and each sensor shuttle board 3.0. Users can access MEMS sensors through a C interface. COINES can be used with the SensorAPI of the sensor. The SensorAPI is available at https://github.com/BoschSensortec. Source code of sample applications and SensorAPI are provided with the COINES library as a package. Users can modify, compile and run the sample applications.



Name	Date modified	Type	Size
coines-api	8/16/2023 4:49 PM	File folder	
datalogger	8/8/2023 2:35 PM	File folder	
doc	8/8/2023 2:35 PM	File folder	
driver	8/8/2023 2:35 PM	File folder	
examples	8/8/2023 2:35 PM	File folder	
firmware	8/16/2023 8:33 PM	File folder	
libraries	6/24/2022 12:39 PM	File folder	
thirdparty	6/24/2022 12:39 PM	File folder	
tools	6/24/2022 12:39 PM	File folder	
coines.mk	7/13/2022 6:28 PM	MK File	9 KB
examples.zip	7/13/2022 6:57 PM	zip Archive	5,094 KB
LICENSES.txt	7/13/2022 6:28 PM	Text Document	18 KB
README.md	7/13/2022 6:28 PM	MD File	3 KB
ReleaseNotes.txt	7/13/2022 6:28 PM	Text Document	12 KB
unins000.dat	8/8/2023 2:36 PM	DAT File	746 KB
unins000.exe	8/8/2023 2:35 PM	Application	714 KB

Figure 2 COINES folder

The subfolder of "datalogger" contains the APP3.0 firmware, Android APK file and user manual PDF file about how to log sensor data via BLE connection between an Android device and APP3.0 board.

The subfolder of "examples" contains subfolders of each sensor that have C source code for evaluation.

4.2 Download and install TDM-GCC compiler

Go to https://github.com/jmeubank/tdm-gcc/releases/download/v10.3.0-tdm64-2/tdm64-gcc-10.3.0-2.exe to download TDM-GCC compiler version 10.3.0 and then install it on PC.

4.3 Download and install ARM GNU Toolchain

Go to https://developer.arm.com/-/media/Files/downloads/gnu/12.3.rel1/binrel/arm-gnu-toolchain-12.3.rel1-mingw-w64-i686-arm-none-eabi.exe?rev=aa6116d1af064a16bdf76e4e58ad7d9f&hash=366EA764314E1A4615E216DDBE7C437E to download ARM GNU Toolchain.

4.4 Test one BMA400 example

- Connect APP3.0 board to PC through a USB cable. BMA400 shuttle board 3.0 is plugged onto APP3.0 bard.

- Switch on the power switch of APP3.0 board. The LED close to the USB connector on APP3.0 board will light up with red color.

- Go to folder C:\COINES\v2.8.8\examples\bma400\examples\accelerometer in Windows File Explorer. Then press and hold “Shift” key on the keyboard and then right click the mouse. Select “Open PowerShell window here” as shown in Figure 3.

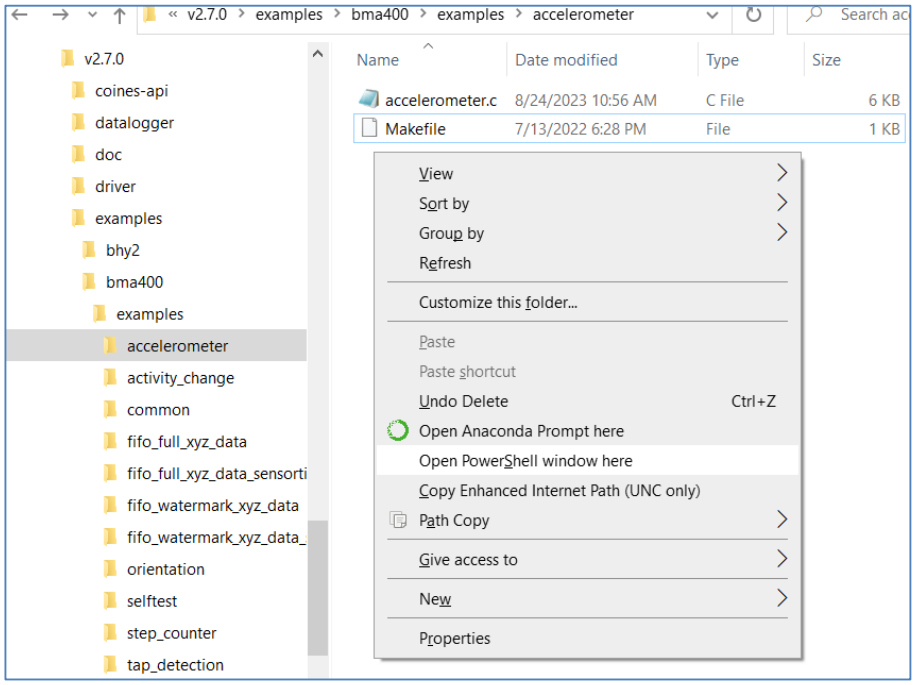


Figure 3 COINES example folder

- Type mingw32-make and then press Enter key. The “accelerometer.exe” file will be generated as shown in Figure 4.

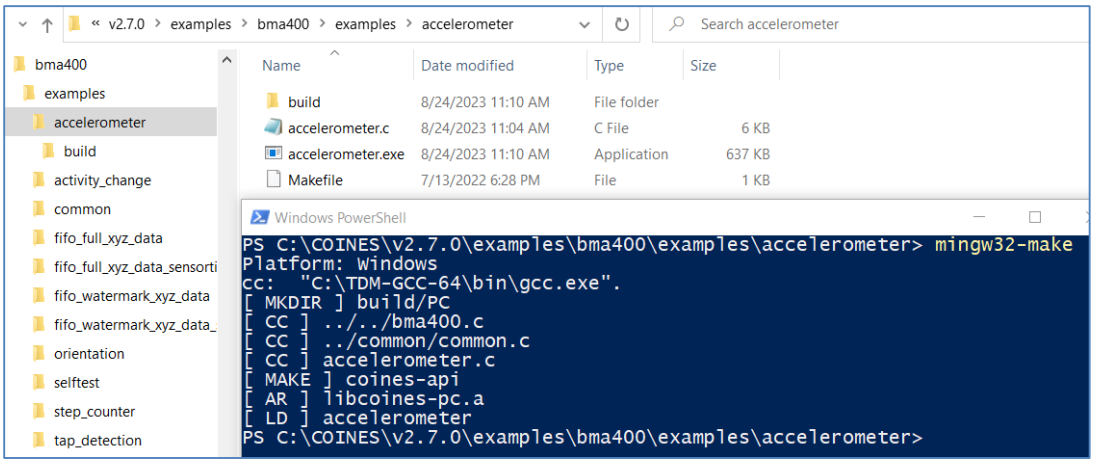


Figure 4 The EXE file is generated

- Type .\accelerometer.exe and then press Enter key. The BMA400 x/y/z raw data in the unit of LSB signed integers is printed in the command window as shown in Figure 5. The converted values in the unit of m/s^2 and sensor time are also printed.

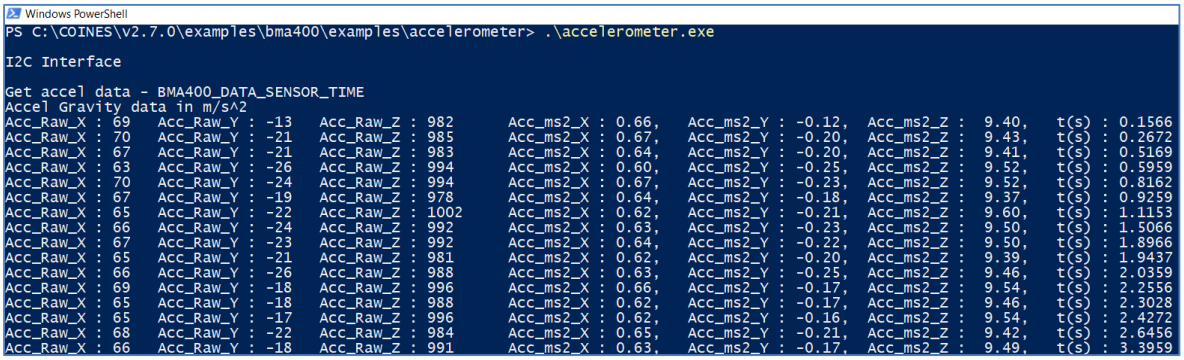


Figure 5 COINES example results of BMA400 measurements

Till now it can be seen that the hardware setup of APP3.0 board and BMA400 shuttle board 3.0 is working and the COINES SW and TDM-GCC compiler are working as well.

4.5 Modify COINES code to enable APP3.0 BLE

- Locate the common.c file in the folder C:\COINES\v2.8.8\examples\bma400\examples\common and open it with Notepad++ SW.

- Change one line code from,

int16_t result = coines_open_comm_intf(COINES_COMM_INTF_USB, NULL);
to,
int16_t result = coines_open_comm_intf(COINES_COMM_INTF_BLE, NULL);
as shown in Figure 6.

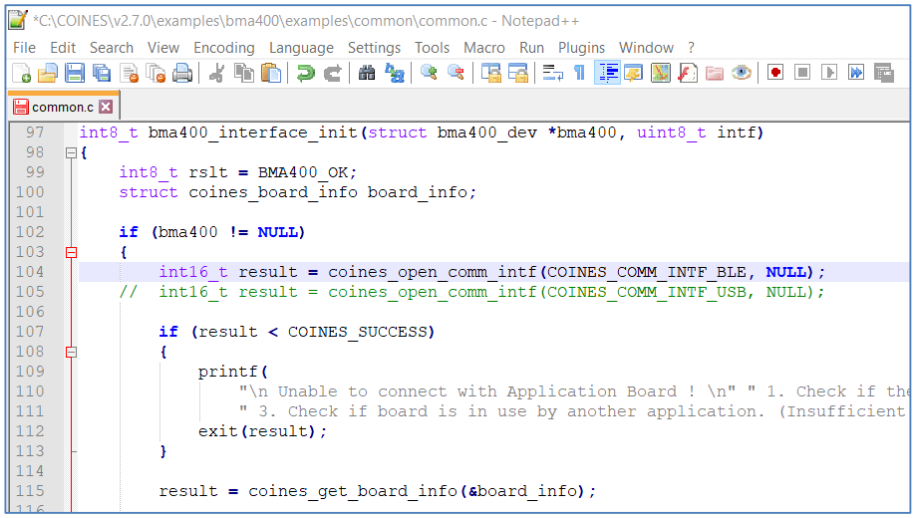


Figure 6 Change COINES common.c file

- Save the “common.c” file and then close it.
- Locate “accelerometer.c” file in the folder C:\COINES\v2.8.8\examples\bma400\examples\accelerometer and open it with Notepad++ SW.
- Add **#include "coines.h"** at the beginning of the file.
- In main() function change printf(“...”) to fprintf(bt_w, ...) function as shown in Figure 7.

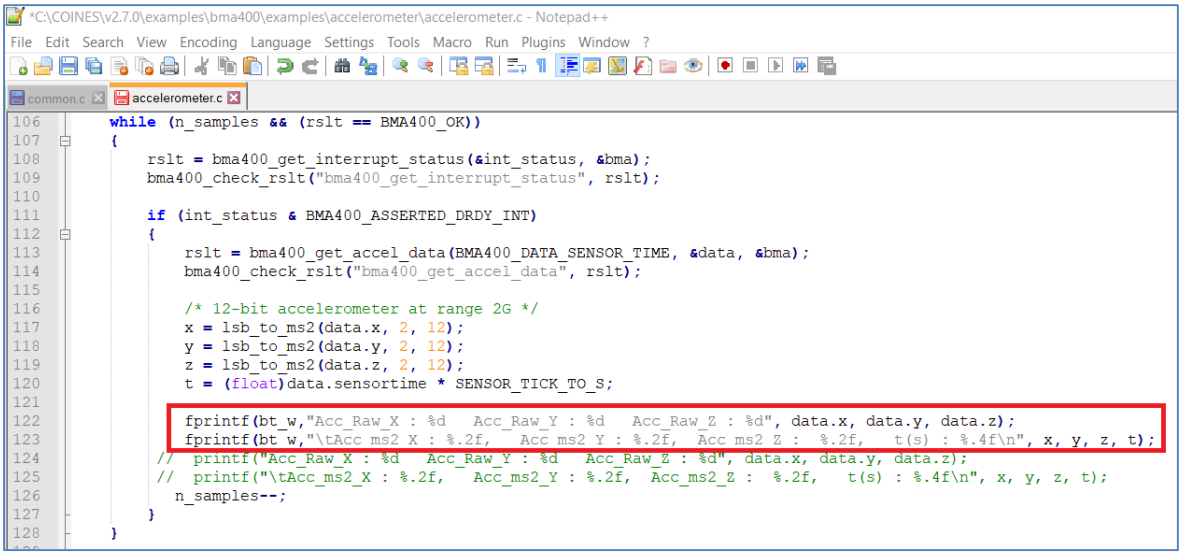


Figure 7 Change COINES accelerometer.c file

- Save the “accelerometer.c” file and then close it. Please note that in the example “accelerometer.c” code, BMA400 is configured to +/-2g full scale range and 100Hz sampling rate. Users can change the configurations as needed.

- In PowerShell window as shown in Figure 4 type command,
  mingw32-make LOCATION=FLASH TARGET=MCU_APP30 download
  and then press Enter key as shown in Figure 8.
  If users type command,
  mingw32-make LOCATION=RAM TARGET=MCU_APP30 download
  and then press Enter key, the firmware will be downloaded to the RAM of APP3.0 MCU. When APP3.0 is powered off the firmware will be gone. Users need to run the above command again.

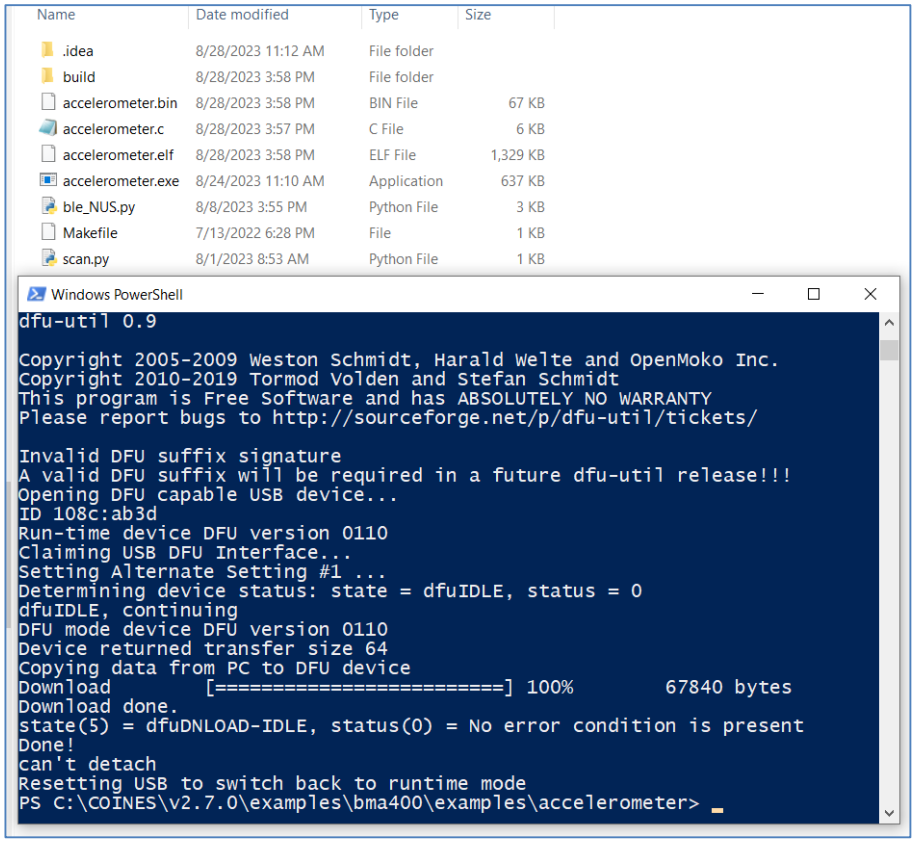


Figure 8 Compile and download final firmware to APP3.0 board

- The compiled final firmware “accelerometer.bin” file will be generated and then downloaded to the ublox module flash memory. When APP3.0 board is powered on again the firmware will start running automatically.

- Switch off APP3.0 board and unplug the USB cable from PC.

- Power APP3.0 board by using battery pack or connecting USB cable to another PC or a USB wall charger for example to charge smartphone.

- Switch on APP3.0 board. Now APP3.0 board is ready to be found and then transmit data to PC via BLE connection.

4.6 Download and install Anaconda Python

Now it is time to setup Python environment and associated packages to enable BLE connection on PC side.

Go to https://repo.anaconda.com/archive/Anaconda3-2023.07-2-Windows-x86_64.exe. Download it and then install it on PC. During the installation, please check the checkbox of “Add Anaconda3 to my PATH environment variable” as shown in Figure 6. Then click the “Install” button and follow the instructions to finish the installation.

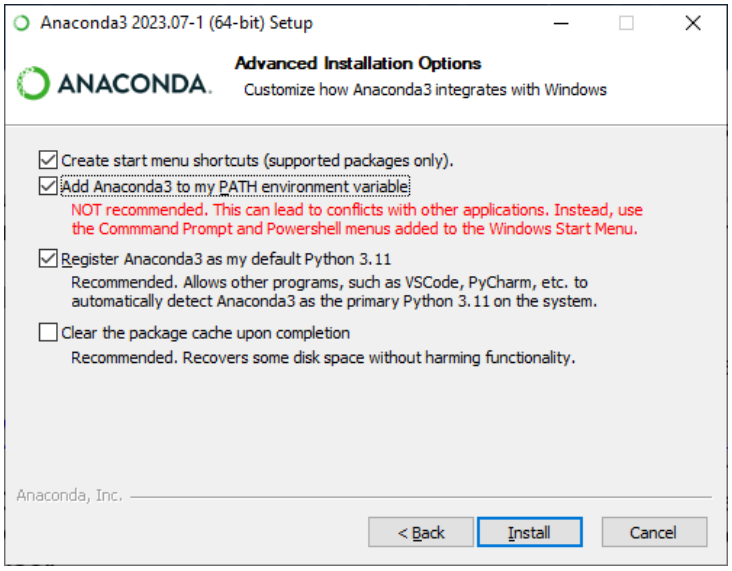


Figure 6 Anaconda installation

4.7 Download and install PyCharm Python IDE community edition

Go to https://www.jetbrains.com/pycharm/download/download-thanks.html?platform=windows&code=PCC to download free PyCharm community edition to PC. During installation, please check the checkbox of “Add “bin” folder to the PATH” as shown in Figure 7. Then click the “Next” button and follow the instructions to finish the installation.

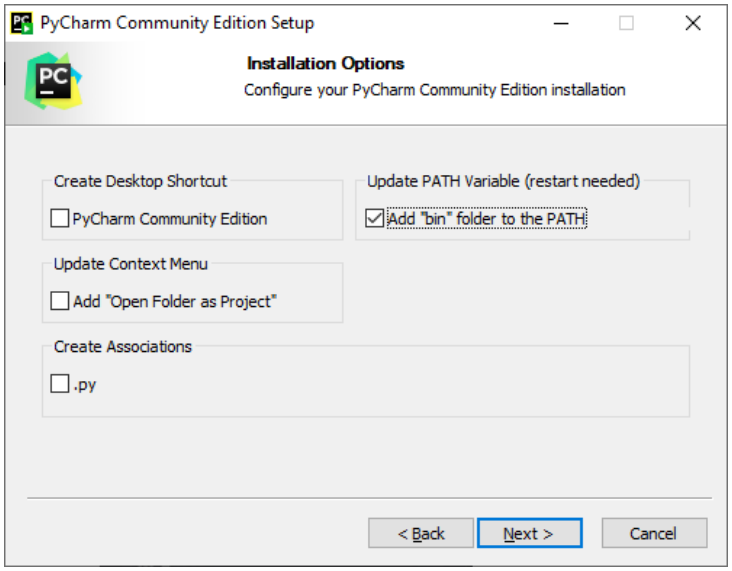


Figure 7 PyCharm installation

4.8 Download Bleak software

Go to https://files.pythonhosted.org/packages/87/95/a6f614fae12a6fe1cf517f8600004dd6abd4af0e0e1177c03164d0637e81/bleak-0.20.2.tar.gz to download the Bleak software.

4.9 Install Bleak software

- From Windows PC Start menu locate “Anaconda Prompt (Anaconda3)” as shown in Figure 8 and then click it.

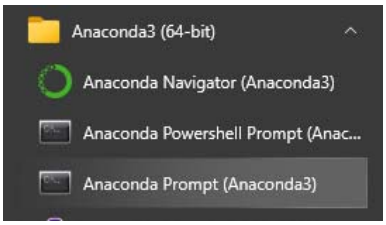


Figure 8 Launch Anaconda Prompt command window

- In the Anaconda command window type “python -m pip install bleak” as shown in Figure 9 and then press Enter key to install Bleak software.

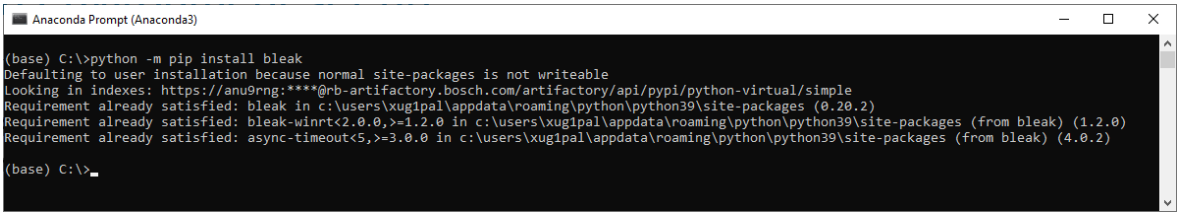


Figure 9 Install Bleak software

4.10 Setup Python environment

- From Windows PC Start menu locate “Anaconda Navigator (Anaconda3)” as shown in Figure 8 and then click it.

- In the home window PyCharm will be shown as a package. Click PyCharm “Launch” button as shown in Figure 10 to launch PyCharm.

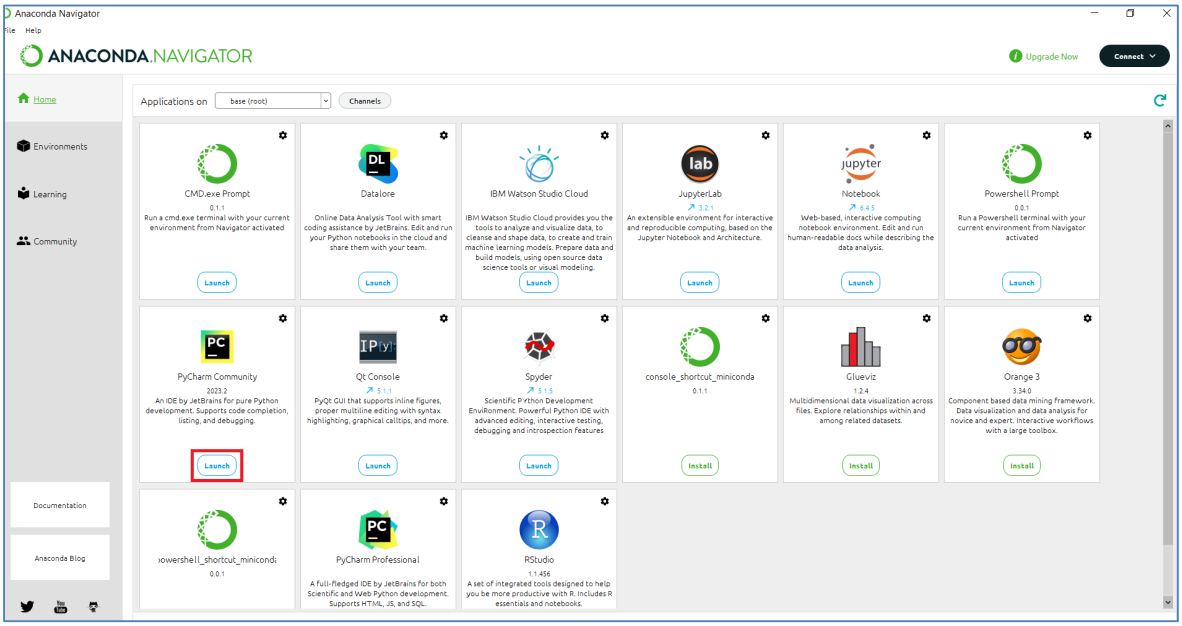


Figure 10 Launch Anaconda Navigator

- In PyCharm window click menu “File -> Open…” and then locate the COINES folder where two Python files have been copied and pasted as shown in Figure 11.

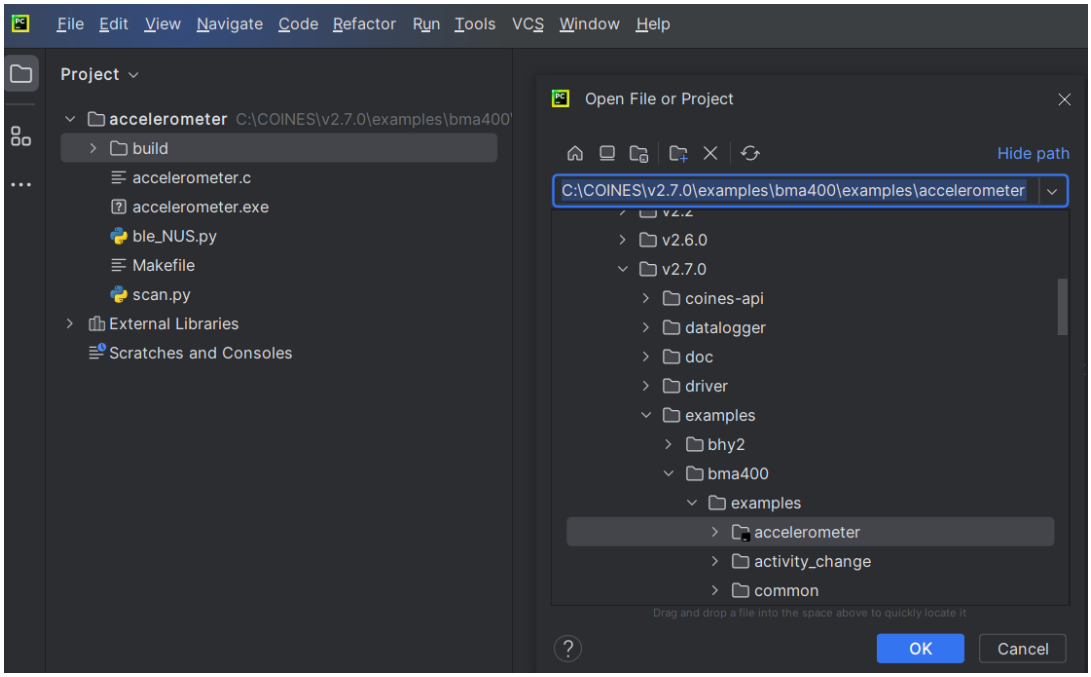


Figure 11 Locate BMA400 example folder in PyCharm

- If this is the first time setting up Python environment for PyCharm, Python Interpreter needs to be installed. Click menu “File -> Settings” and then click “Project: accelerometer -> Python Interpreter”. If there is no Python Interpreter listed, then click “Add Interpreter” in the red rectangle as shown in Figure 12. Note that Figure 12 already has Python Interpreter installed.

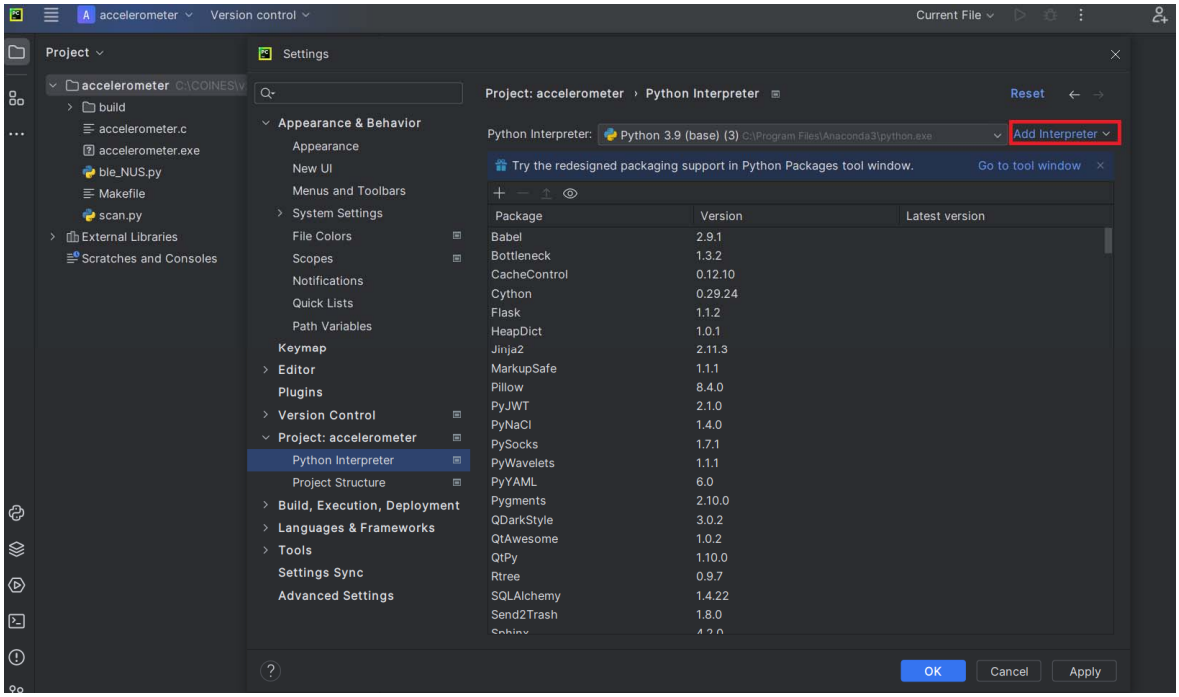


Figure 12 Add Python Interpreter 1 of 2

- In Figure 13 select “Existing” radio button and then click “...” button. Select “python.exe” and then click OK button. Users will see Figure 12 with Python Interpreter installed.

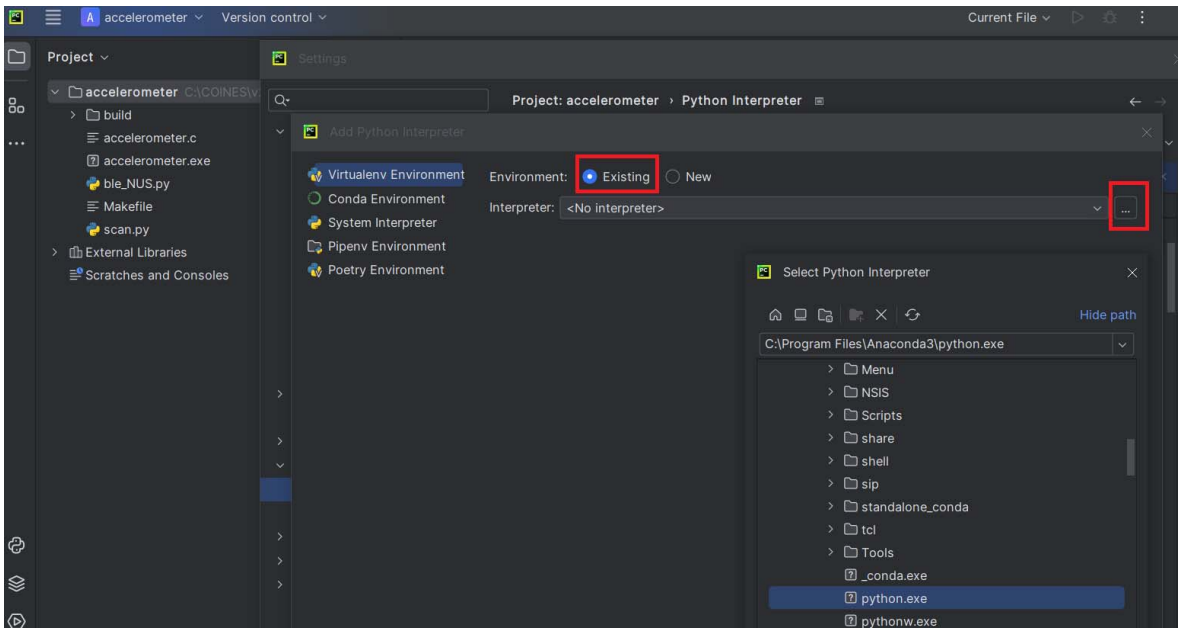


Figure 13 Add Python Interpreter 2 of 2

4.11 Test BLE connection between APP3.0 board and PC

- On the left panel under “accelerometer” project, right click “scan.py” and select “Run ‘scan’” as shown in Figure 14.

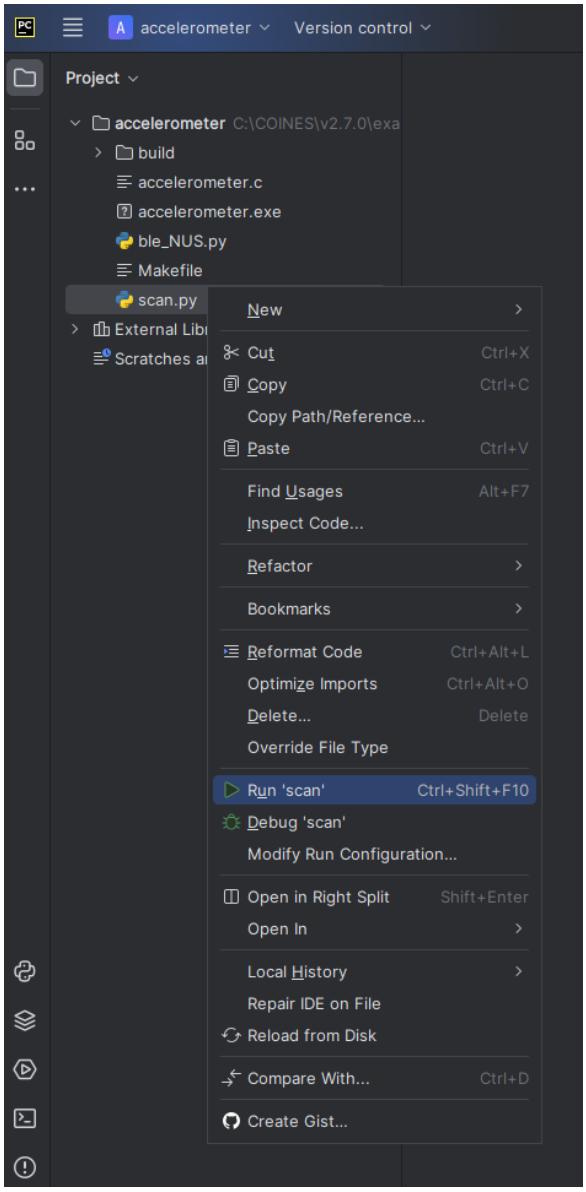


Figure 14 Run “scan.py”

- APP3.0 board will be found with MAC address listed as shown in Figure 15. The MAC address for this APP3.0 board is unique C8:90:B7:F9:1B:73.

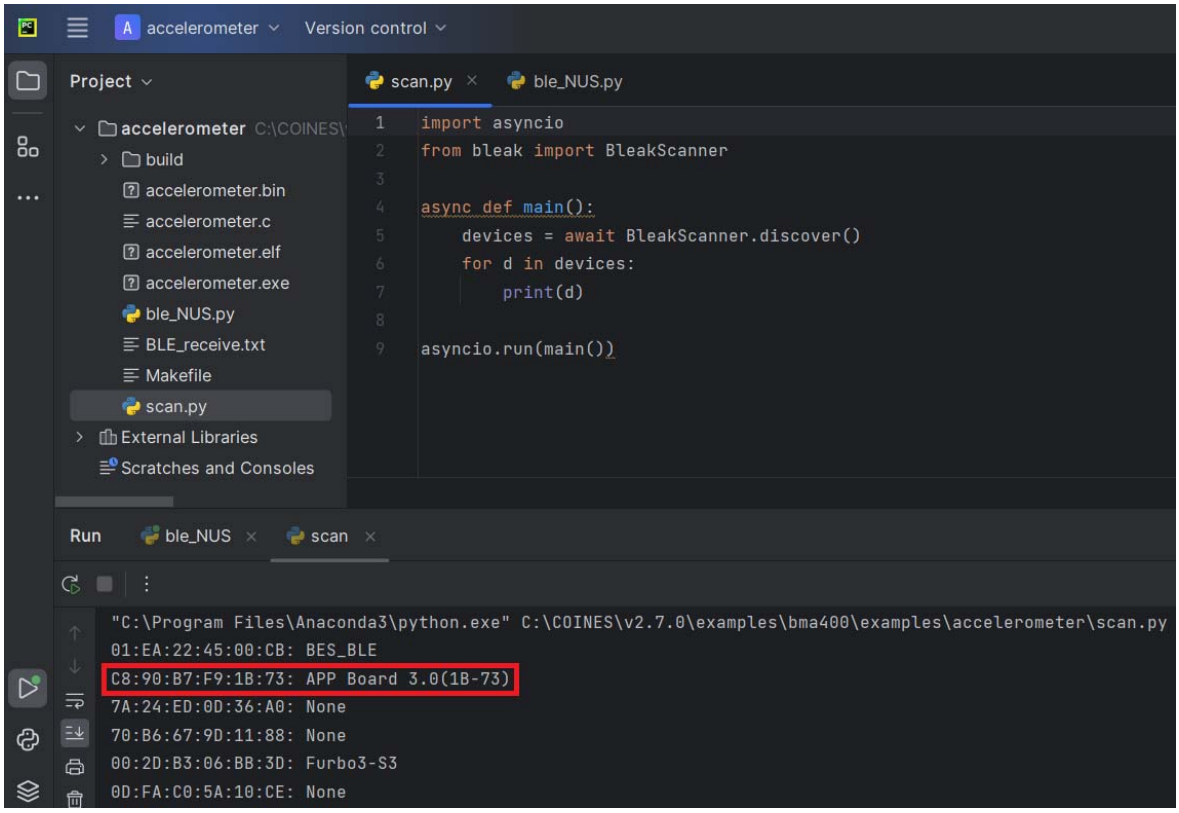


Figure 15 Find APP3.0 MAC address for BLE communication

- Click “ble_NUS.py” file and paste the MAC address to the place as shown in Figure 16.

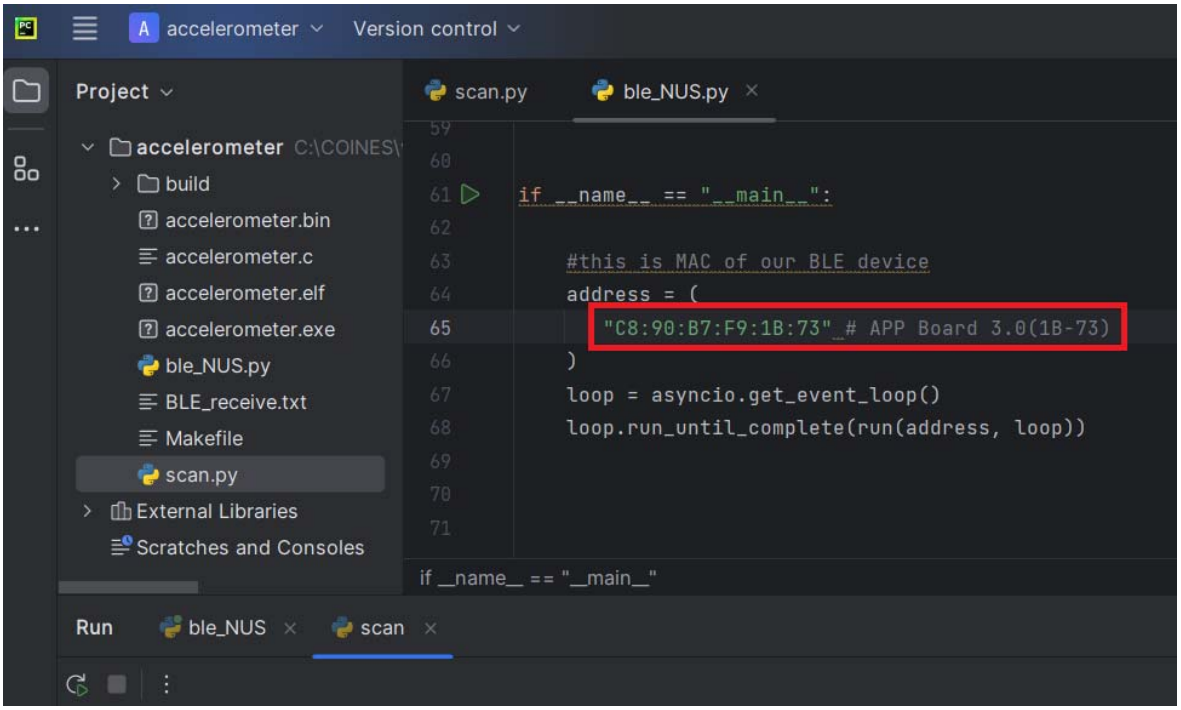


Figure 16 Copy and paste the MAC address to “ble_NUS.py” file

- On the left panel under “accelerometer” project, right click “ble_NUS.py” and select “Run ‘ble_NUS’”. The results are printed as shown in Figure 17.

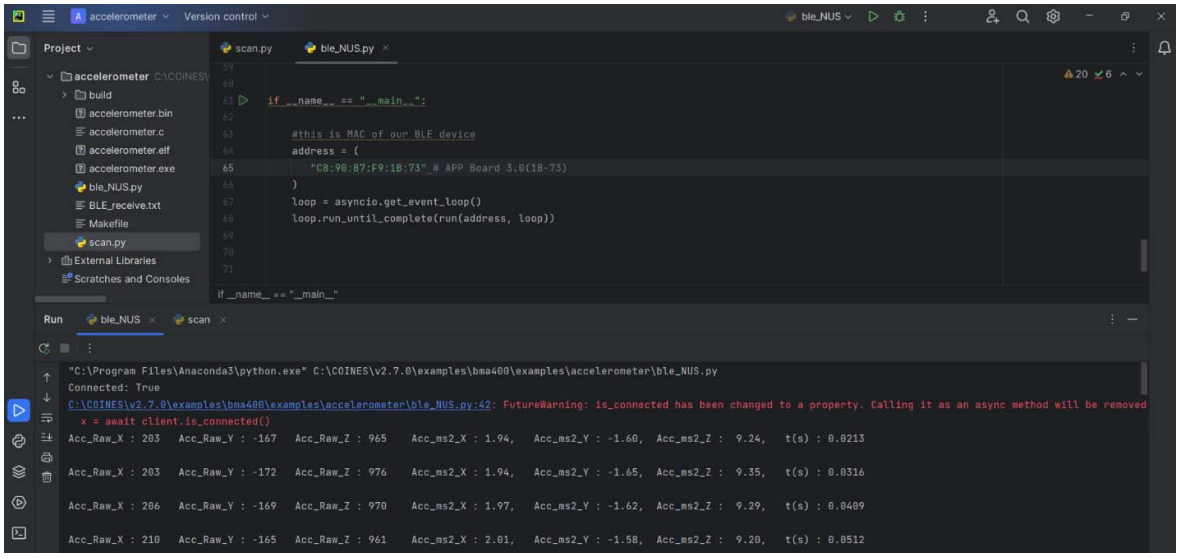


Figure 17 BMA400 data printed on PC from APP3.0 board via BLE connection

- In the meantime, BMA400 data is also logged in “BLE_receive.txt” file on PC as shown in Figure 18. It is the same as the BMA400 data printed.

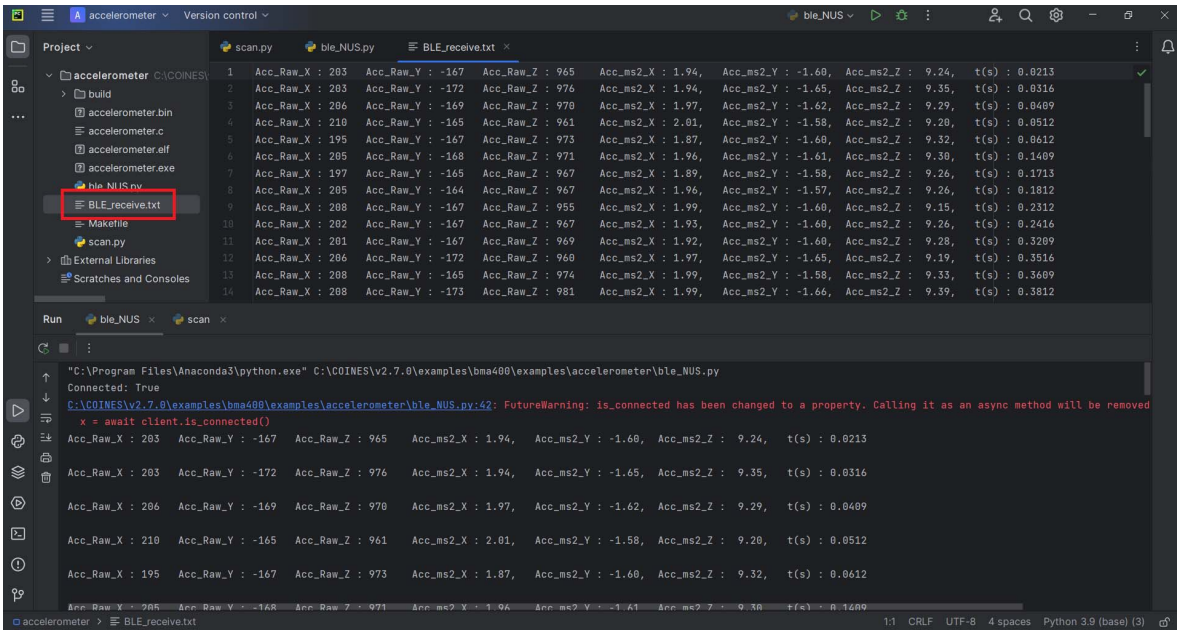


Figure 18 BMA400 data logged in “BLE_receive.txt” file on PC

- Done.

5 Appendix A Source code of “scan.py” file

```
import asyncio
from bleak import BleakScanner

async def main():
    devices = await BleakScanner.discover()
    for d in devices:
        print(d)

asyncio.run(main())
```

6 Appendix B Source code of “ble_NUS.py” file

```
"""
Coyt Barringer - 2020
Test program demonstrating data transmission between Adafruit Bluefruit BLE
libraries running
on nrf52840 and Python
This uses the Nordic Uart Service (NUS) and should work concurrently with
other BLE services such as HID
On the python side, the Bluetooth Low Energy platform Agnostic Klient for
Python (Bleak) project
is used for Cross Platform Support and has been tested with windows 10
"""


import asyncio
from bleak import BleakClient


UART_TX_UUID = "6e400002-b5a3-f393-e0a9-e50e24dcca9e" #Nordic NUS
characteristic for TX
UART_RX_UUID = "6e400003-b5a3-f393-e0a9-e50e24dcca9e" #Nordic NUS
characteristic for RX

dataFlag = False #global flag to check for new data
index = 0
dataList = []
file = open("BLE_receive.txt", "w")


def notification_handler(sender, data):
    """Simple notification handler which prints the data received."""
    global file

    # print("{0}: {1}".format(sender, data))
    readstr = data.decode('utf-8')
    print(readstr)
    file.write(readstr)
    file.flush()
```

```
    global dataFlag
    dataFlag = True


async def run(address, loop):

    async with BleakClient(address, loop=loop) as client:

        #wait for BLE client to be connected
        x = await client.is_connected()
        print("Connected: {0}".format(x))

        #wait for data to be sent from client
        await client.start_notify(UART_RX_UUID, notification_handler)


        while True :

            #give some time to do other tasks
            await asyncio.sleep(0.0)

            #check if we received data
            global dataFlag
            if dataFlag :
                dataFlag = False



if __name__ == "__main__":

    #this is MAC of our BLE device
    address = (
       "C8:90:B7:F9:1B:73" # APP Board 3.0(1B-73)
    )
    loop = asyncio.get_event_loop()
    loop.run_until_complete(run(address, loop))
```

7 Legal disclaimer

7.1 Engineering samples

Engineering Samples are marked with an asterisk (*) or (e) or (E). Samples may vary from the valid technical specifications of the product series contained in this data sheet. They are therefore not intended or fit for resale to third parties or for use in end products. Their sole purpose is internal client testing. The testing of an engineering sample may in no way replace the testing of a product series. Bosch Sensortec assumes no liability for the use of engineering samples. The Purchaser shall indemnify Bosch Sensortec from all claims arising from the use of engineering samples.

7.2 Product use

Bosch Sensortec products are developed for the consumer goods industry. They may only be used within the parameters of this product data sheet. They are not fit for use in life-sustaining or security sensitive systems. Security sensitive systems are those for which a malfunction is expected to lead to bodily harm or significant property damage. In addition, they are not fit for use in products which interact with motor vehicle systems.

The resale and/or use of products are at the purchaser's own risk and his own responsibility. The examination of fitness for the intended use is the sole responsibility of the Purchaser.

The purchaser shall indemnify Bosch Sensortec from all third party claims arising from any product use not covered by the parameters of this product data sheet or not approved by Bosch Sensortec and reimburse Bosch Sensortec for all costs in connection with such claims.

The purchaser must monitor the market for the purchased products, particularly with regard to product safety, and inform Bosch Sensortec without delay of all security relevant incidents.

7.3 Application examples and hints

With respect to any examples or hints given herein, any typical values stated herein and/or any information regarding the application of the device, Bosch Sensortec hereby disclaims any and all warranties and liabilities of any kind, including without limitation warranties of non-infringement of intellectual property rights or copyrights of any third party. The information given in this document shall in no event be regarded as a guarantee of conditions or characteristics. They are provided for illustrative purposes only and no evaluation regarding infringement of intellectual property rights or copyrights or regarding functionality, performance or error has been made.

8 Document history and modification

Rev. No	Chapter	Description of modification/changes	Date
1.0		Document creation	August 28th, 2023
2.0	4.1	Changed to COINES v2.8.8 from v2.7.0	November 1st, 2023

Bosch Sensortec GmbH
Gerhard-Kindler-Strasse 8
72770 Reutlingen / Germany

Contact@bosch-sensortec.com
www.bosch-sensortec.com