Getting started with BLE Sensor GUI

# Introduction

The PowerON BLE Sensor GUI, which runs on Python on a Windows computer, was designed to operate with the BLE Touch Detect. All the steps to run the BLE Sensor GUI are detailed in this document.

# Install Python

Please download **Python version 3.8.9** and install it. The version is very important. The GUI does not work with Python 3.9 and probably the later version of Python. This is mainly due to the ‘Bleak’ Python package.

Python can be downloaded from <https://www.python.org/downloads/>

# Install Python IDE

There are various Python IDE that can be used, and it is up to one’s preference.

PyCharm (<https://www.jetbrains.com/pycharm/>) is used in PowerON NZ.

A famous alternative is Anaconda. Unfortunately, Zak has not used Anaconda, so he won’t be able to support you. However, there are support available online given that it is a widely used software.

# Downloading BLE Python GUI files

The files for the BLE Python GUI can be found here:

* Look up GitLab for the most recent file.
* Alternative: Nextcloud\Documents\PowerOn\_general\D R&D\new WA5 Supporting Technologies\Interfacing electronics for TouchDetect\PWNP00009\_Bluetooth with STM32WB\PythonGUI\_BLE\ble-python-gui

The files required is in the folder called ‘Standard’. It consists of 4 files:

* main.py
* Logo\_PWN\_TM3.png
* Element\_64x.png
* Logo.ico

Please download it to your computer.

# Installing Python packages

Below are the packages required to run the BLE Python GUI:

* **Matplotlib**
* **Bleak**
* **Asyncio**
* **Numpy**
* **Tkinter**
* **Pillow**
* Threading (should be part of Python 3.8)
* CSV (should be part of Python 3.8)
* Time (should be part of Python 3.8)

Those which are part of Python do not need to be specially downloaded.

To build the executable the following packages are also required:

* **PyInstaller**
* os (should be a part of Python 3.8)

There are various ways to download python packages, where most use pip.

If you are using PyCharm:

1. Open PyCharm
2. Select the directory of the Python GUI files
   1. Download the Python files from GitLab
3. Select ‘File 🡪 Settings 🡪 Project: (Project Name) 🡪 Python Interpreter’
4. Select the Python Interpreter to be ‘Python 3.8.9’ (see Figure 1)
5. Select the ‘+’ button (highlighted with red circle in Figure 1)
6. In the search bar, find the packages stated in the list above.
7. Click ‘Install Package’

Ensure that you have downloaded those in bold as those are crucial for the software to operate.

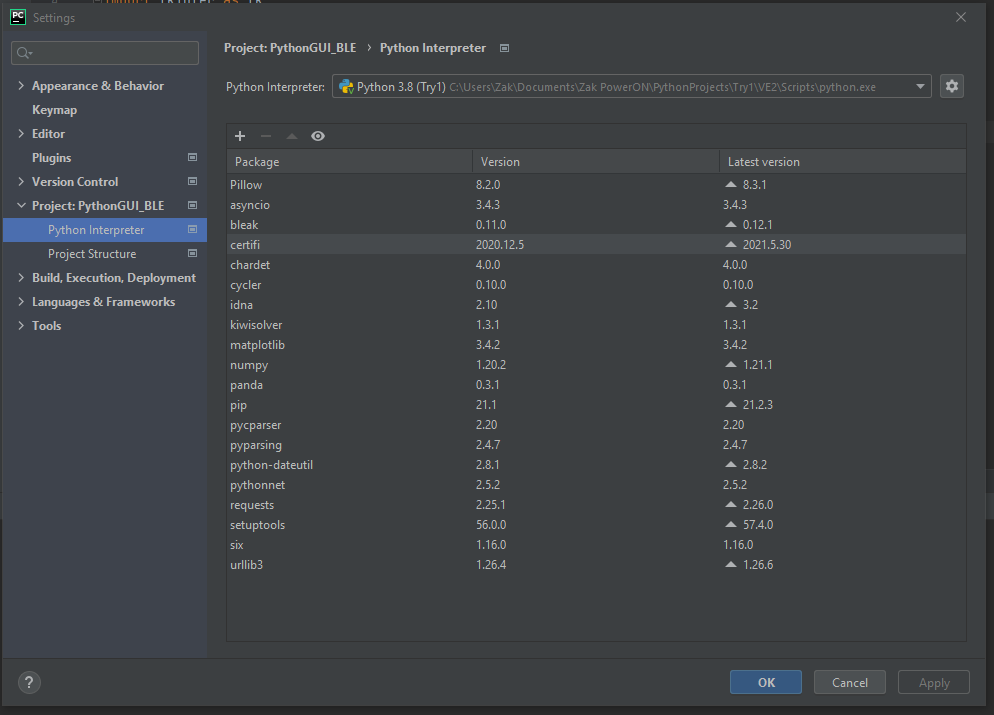


Figure . PyCharm setting and adding packages.

# Operating BLE Python GUI with BLE Touch Detect

In PyCharm, simply click on the ‘Run 🡪 Run’ to run the software. A Run pop-up may appear. Just select ‘main’ to run the software.

Once the Run button is clicked, the Python program will appear as shown in Figure 2.

# Building and Running an Executable

In PyCharm, simply click on the ‘Run 🡪 Run’ to run the software when viewing the build\_exe.py program. A Run pop-up may appear. Just select ‘main’ to run the software. The executable will then appear in the dist/ folder. To run an executable simply double click it.

Calendar

Description automatically generated

Figure . PowerON BLE Sensor GUI.

## Viewing data from BLE Touch Detect

To read values from the BLE Touch Detect PCB:

1. Ensure that the PowerON BLE Sensor GUI is running
2. Ensure that the BLE Touch Detect PCB is turned on
3. Ensure that the Device Name is correct
4. Click on ‘Connect’ button.
   1. Please wait a few seconds for the BLE connection to be established.
5. Once connected, click on ‘Receive Data’.

## Stop viewing data and exiting

To stop viewing the data, simply click on the ‘Stop Data’ button when receive data mode is active.

To exit the program, click on ‘Exit’ button. Please wait a while for the program to exit and close the program window.

## Data logging

To enable data logging, once the BLE connection is established, click on the ‘Yes’ radio button in the Data Logging section. A .csv file with a time stamp will be in the same directory as the python files.

Whenever the ‘No’ radio button of the data logging section is pressed, the data will be saved and closed. Any subsequent click of ‘Yes’ in the data logging will create a new file with a new timestamp.

Note that a data logging file cannot be created when receiving data has been started. One will need to stop receiving data, and then select ‘Yes’ for data logging and restart receiving data.

The data saved in the .csv files are raw data. They are not processed as that displayed in the GUI.

## Changing data display

To change the data displayed, simply click on ‘Voltage’, ‘Resistance’ or ‘Force’ buttons in the Data Display section.

This is currently under development. The options require calibration to fit the sensor.

# Known bugs in the PowerON BLE Sensor GUI

Some of the known bugs with the Python GUI are described in the following.

## Program stalls when no longer receiving data

If the BLE Touch Detect PCB were to stop transmitting data (like due to insufficient power), there is a tendency for the Python GUI to stall/freeze. Similar click on the ‘Stop’ button in PyCharm to forcefully exit the program.

## Exit bug

There is a known bug in the GUI, where it generates a ‘known issue disconnecting’ in the console when the ‘disconnect’ button or exit button (when receive data mode is running) is clicked. This is mainly due to the ‘bleak’ python package used. It does take a while for the program to stop running and close the program window but does not affect the performance of receiving data etc.