RoboKit User Guide

Windows Application

InvenSense, a TDK Group Company
1745 Technology Drive, San Jose, CA 95110 U.S.A
+1(408) 988–7339
invensense.tdk.com

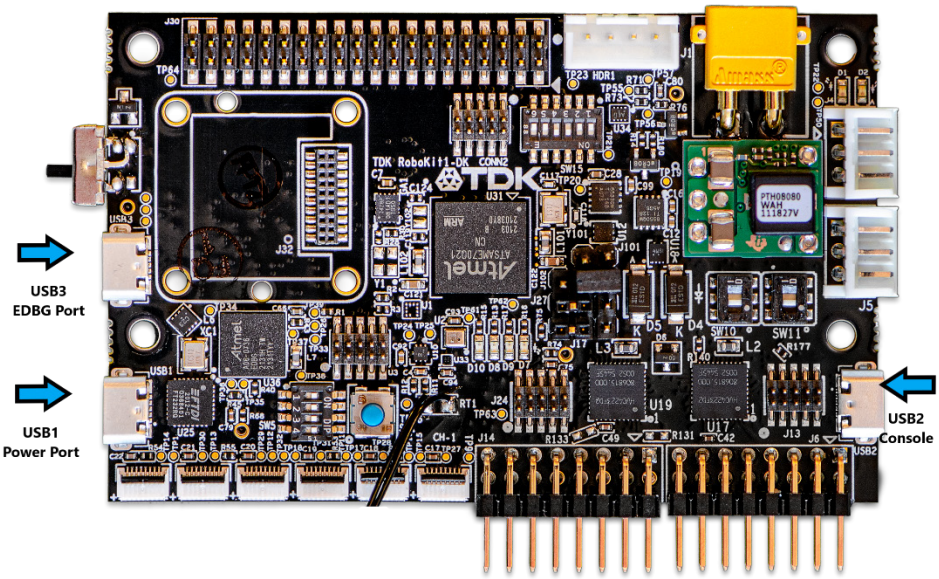
Document Number: AN-000346
Revision: 1.1
Release Date: 02/28/2022

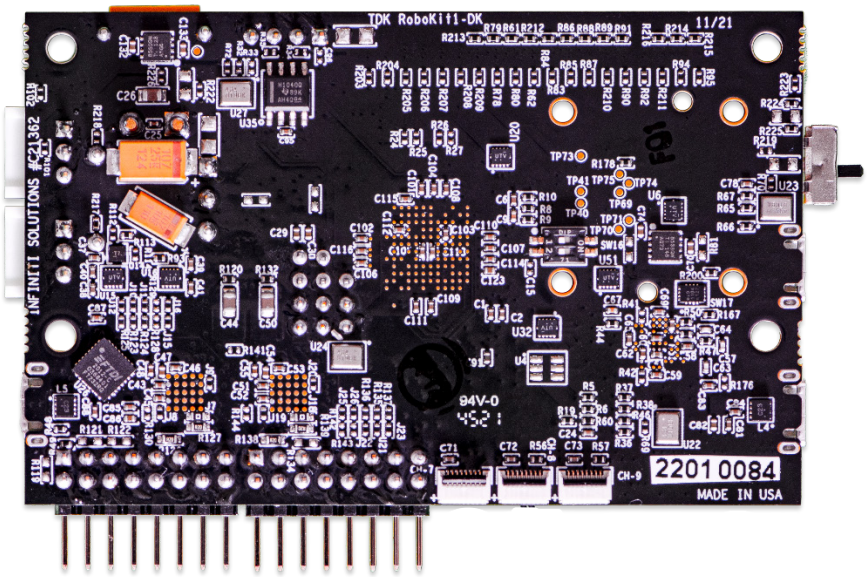
TABLE OF CONTENTS

TABLE OF FIGURES

1 ROBOKIT1 MODULE

This document provides an overview of how to utilize the RoboKit1 app for data capture and logging. Figure 1 and Figure 2 show the top and bottom views of a RoboKit1 module. This module includes a Pressure sensor, IMU, Temperature sensor, Magnetometer, Audio, Microphone, Motor controller, and Battery connector on board. It also provides the facility to connect 9 Chirp sensors. USB1 is used for power supply and the interface with the board. USB2 is used as port for debug text messages and USB3 is to load the Firmware image.



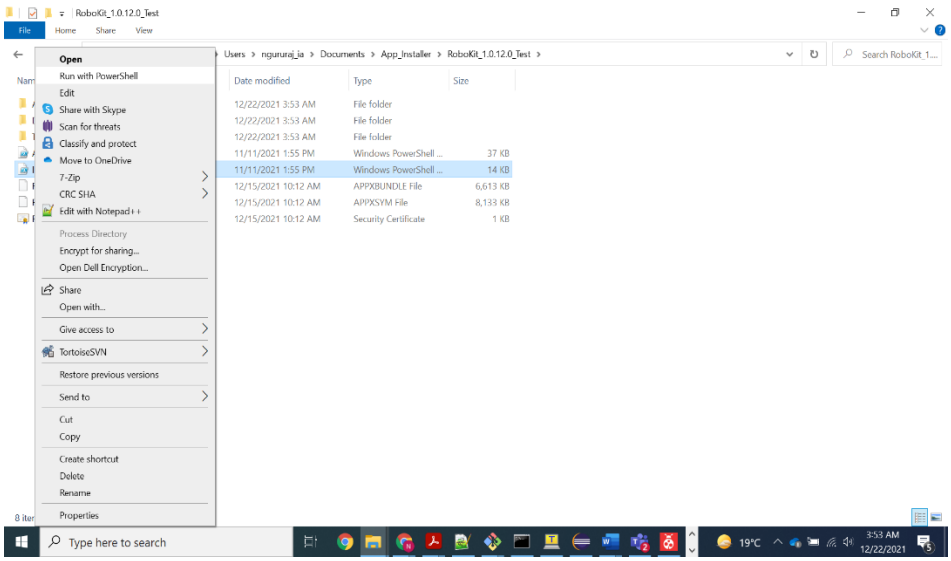


2 INSTALLING & RUNNING ROBOKIT

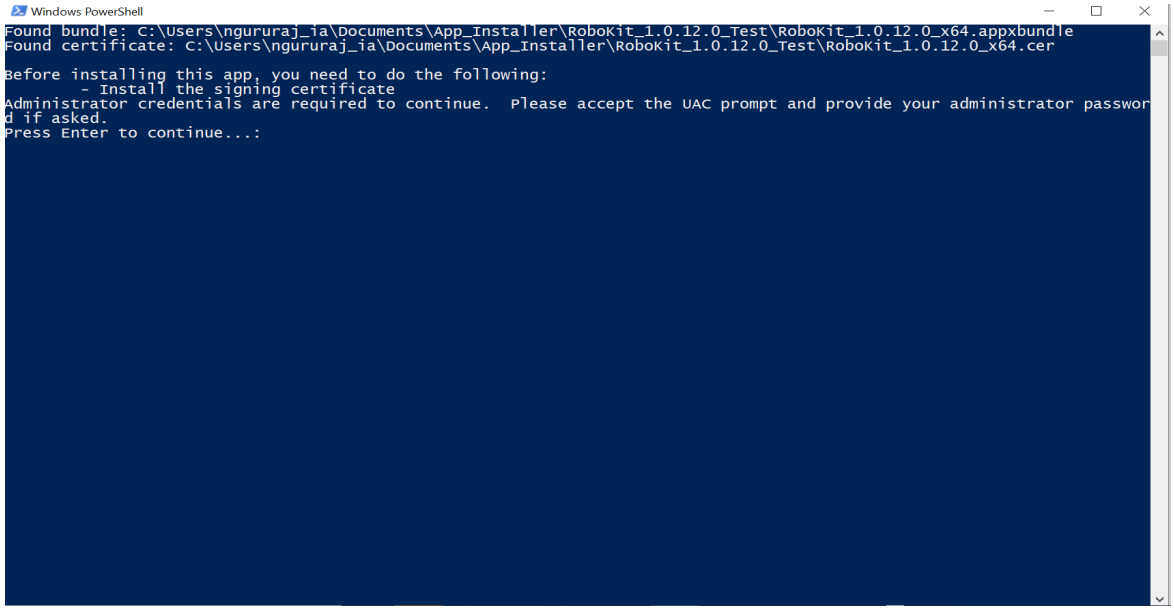
2.1 INSTALLING FROM LOCAL DIRECTORY

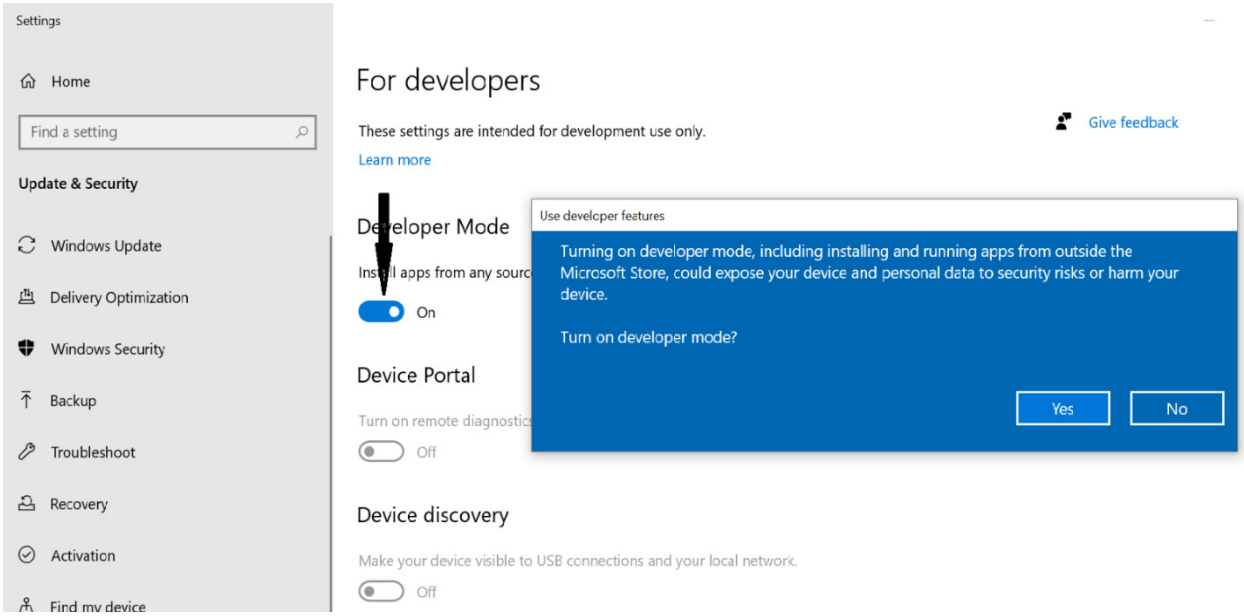
Follow the process below to install the application from the local directory:

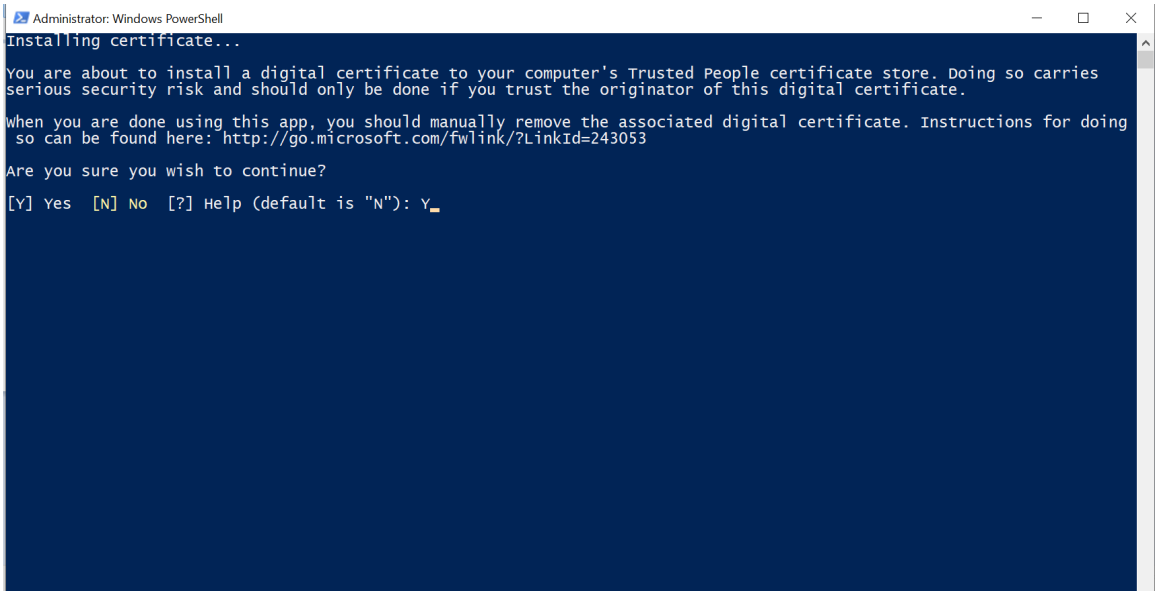
1. Right click on the "**install**" file and click on "**Run with PowerShell**"



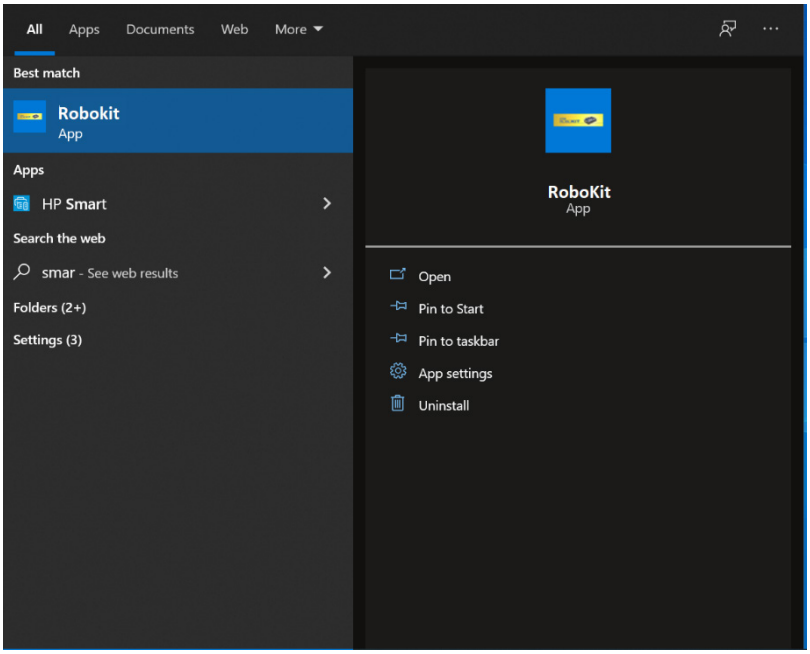
2. Press “**Enter**” to provide administrator access to the allow installation of the certificates related to the application.







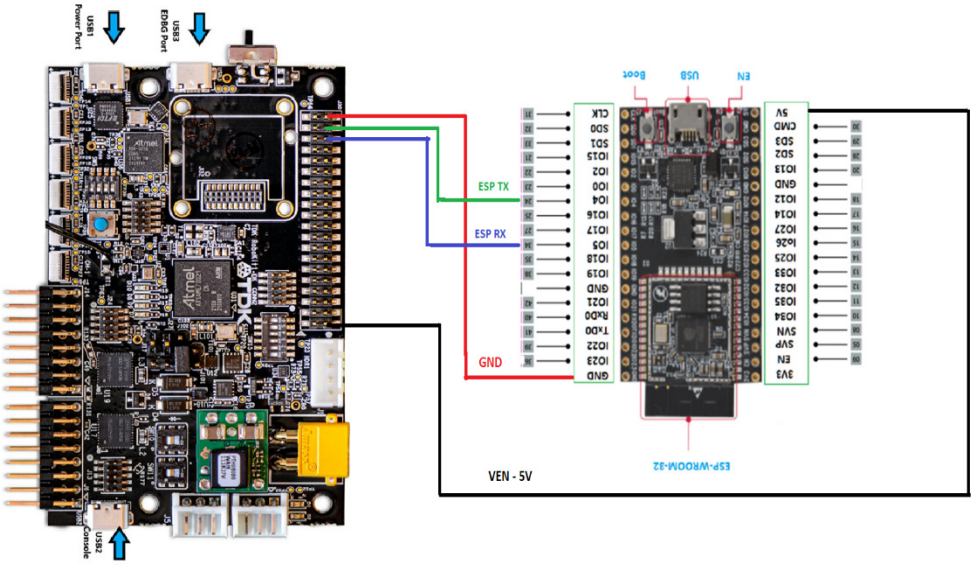
3. Press the Windows key, type "**RoboKit**" and then right-click on it to launch the application.



3 CONNECTING ESP32 WITH ROBOKIT

RoboKit in a standalone mode offers the capability to communicate with the Host device via USB mode. To achieve wireless data transfer over the BLE interface, an extension board based on ESP 32 is used. Data transfer between RoboKit and ESP happens over the UART at the rate of the 1 Mbps.

Details of the physical connection between the boards is as detailed Figure 8:



Note: The ESP32 might need CP210x USB to UART Bridge Virtual COM Port (VCP) drivers which need to installed depending on the user's OS platform. The drivers are available as part of firmware download package in TDK website.

4 UPDATING ESP32 FIRMWARE

The Bluetooth connection to RoboKit1 is enabled by ESP32 and the firmware running on it must be kept updated. The ESP32 BLE firmware is provided with the release package. Please follow the steps below to update the firmware:

1. Unzip the esp-ble-firmware.zip file
2. Connect to ESP32 from your Windows Machine with USB. Identify the Serial Port (Ex. COM1)
3. Open Windows PowerShell and navigate to the folder **esp-ble-firmware**
4. Execute the below command

   `./flash-binary.sh <serial-com-port>` esp-ble-firmware-vx.y.z.bin

   where,

serial-com-port	Serial Port connected to ESP32
esp-ble-firmware-vx.y.z.bin	Latest firmware, x, y and z are version numbers

5. On successful firmware upgrade, the RTS pin will be reset as shown in Figure 9.

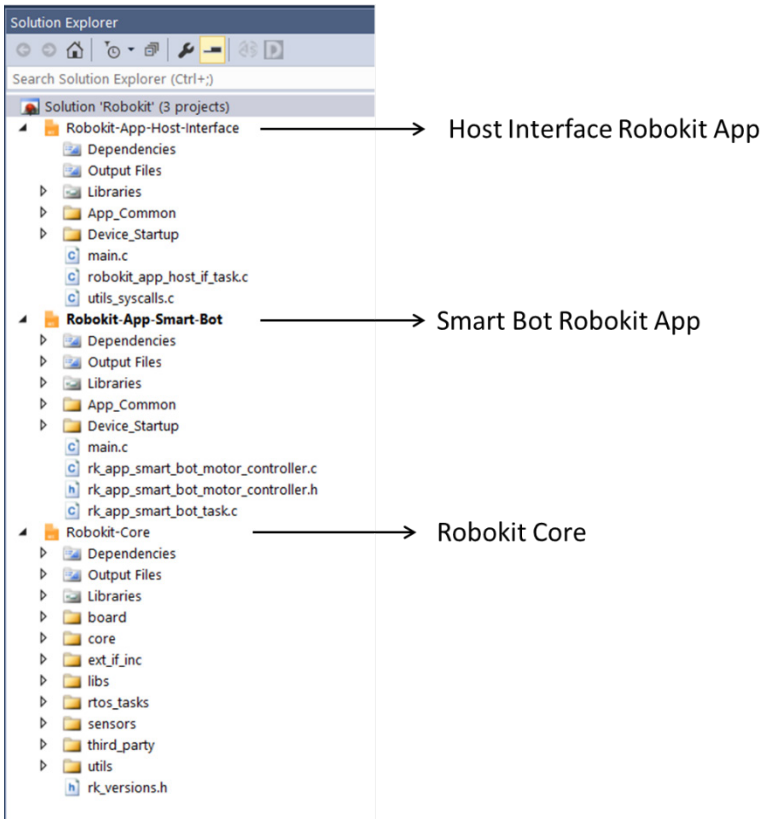
```
esptool.py v3.1-dev
Serial port COM16
Connecting....
Chip is ESP32-D0WD (revision 1)
Features: WiFi, BT, Dual Core, 240MHz, VRef calibration in efuse, Coding Scheme None
Crystal is 40MHz
MAC: 98:cd:ac:62:d3:28
Uploading stub...
Running stub...
Stub running...
Changing baud rate to 460800
Changed.
Configuring flash size...
Auto-detected Flash size: 4MB
Flash will be erased from 0x00001000 to 0x00007fff...
Flash will be erased from 0x00008000 to 0x00008fff...
Flash will be erased from 0x00010000 to 0x000befff...
Flash params set to 0x0220
Compressed 25088 bytes to 15383...
Wrote 25088 bytes (15383 compressed) at 0x00001000 in 0.8 seconds (effective 267.3 kbit/s)...
Hash of data verified.
Compressed 3072 bytes to 103...
Wrote 3072 bytes (103 compressed) at 0x00008000 in 0.1 seconds (effective 406.8 kbit/s)...
Hash of data verified.
Compressed 714976 bytes to 427266...
Wrote 714976 bytes (427266 compressed) at 0x00010000 in 10.5 seconds (effective 543.5 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
```

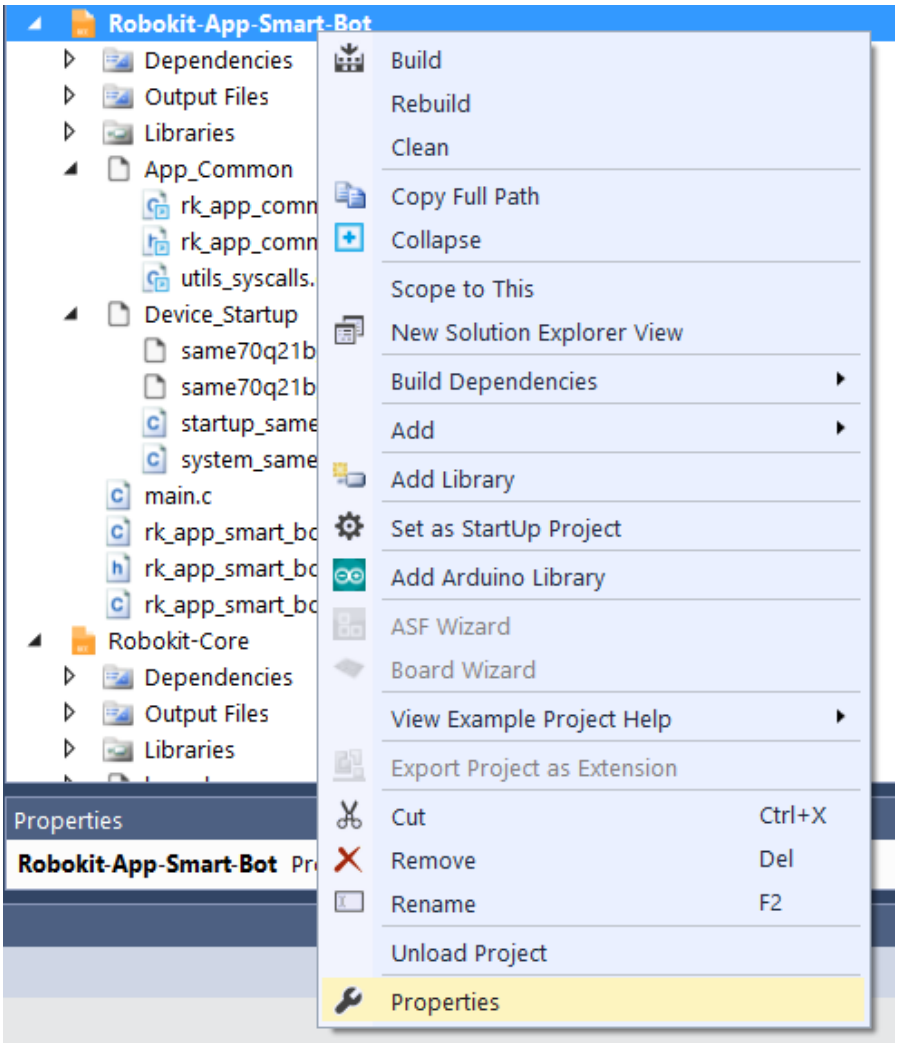
6. Restart the ESP32 with a power cycle.

5 LOADING ROBOKIT ATMEL BINARY

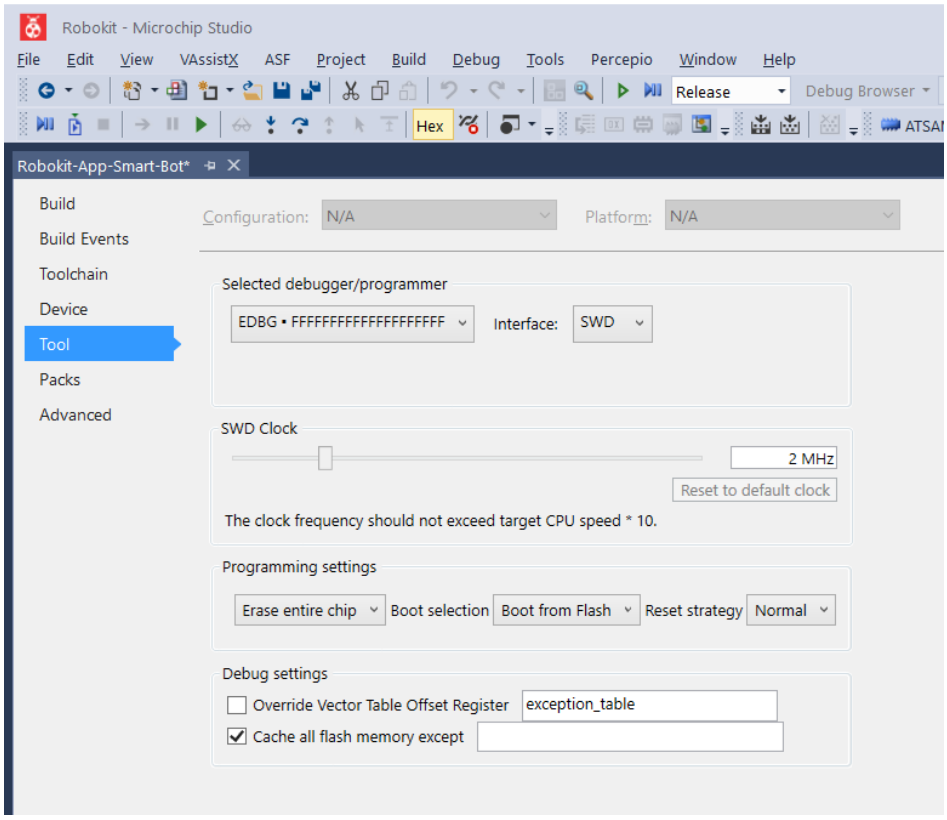
The main solution file, “Robokit.atsln”, for firmware is located at the root folder of the software package and can be opened with Microchip Studio. It includes Robokit-Core and two application project files as shown in the Figure 10. To compile the needed application project, set the application project file as the startup project by right clicking on it and selecting “Set as Startup Project” and them compile it.



To load the image, select properties of Robokit-App as shown in Figure 11.



Set EDBG as debugger/Programmer as shown in Figure 12 and click “Start Debugging” or F5.


Robokit - Microchip Studio
File Edit View VAssistX ASF Project Build Debug Tools Percepio Window Help
Release
Debug Browser
Hex
Robokit-App-Smart-Bot*
Build
Build Events
Toolchain
Device
Tool
Packs
Advanced
Configuration: N/A
Platform: N/A
Selected debugger/programmer
EDBG • FFFFFFFFFFFFFFFFFFFF
Interface: SWD
SWD Clock
2 MHz
Reset to default clock
The clock frequency should not exceed target CPU speed * 10.
Programming settings
Erase entire chip
Boot selection
Boot from Flash
Reset strategy
Normal
Debug settings
Override Vector Table Offset Register
exception_table
Cache all flash memory except

6 CONNECTING TO THE ROBOKIT

TDK offers two products: a stand-alone development board package (TDK_RoboKit1-DK) and a full robot package (TDK_RoboKit1).

TDK_RoboKit1-DK includes:

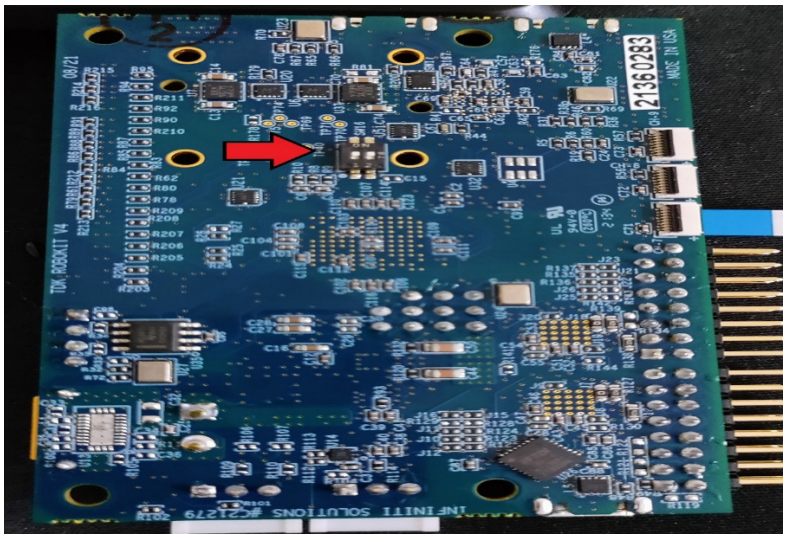
- Industrial IMU
- Chirp CH101(3x)
- Chirp Connector Cables(3x)
- Windows & Android App for Data collection
- Open-sourced based Board Design files, Firmware and ROS Drivers

Full Robot Package includes:

- Complete “Development Board Package” +
- 3D printed Robot shell with sockets for Time of Flight sensors
- Chassis with Metal plates, Standoffs, Wheels, and Motors
- ESP32 BLE module for BLE connectivity.

6.1 SWITCHING BETWEEN USB AND BLE MODE

To connect the app in USB mode, select the DIP switch SW16 with both switches 1&2 towards ON position as shown in Figure 13.



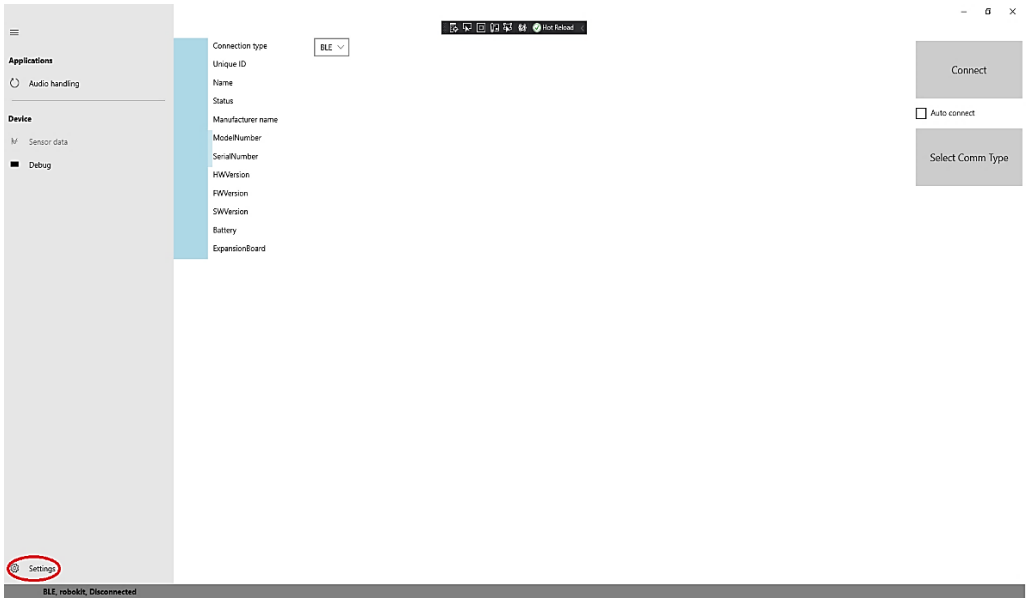
To connect in BLE mode switches 1&2 should be in OFF position, i.e. both downwards.

Caution: Disconnect the USB power prior to modifying the DIP switch settings.

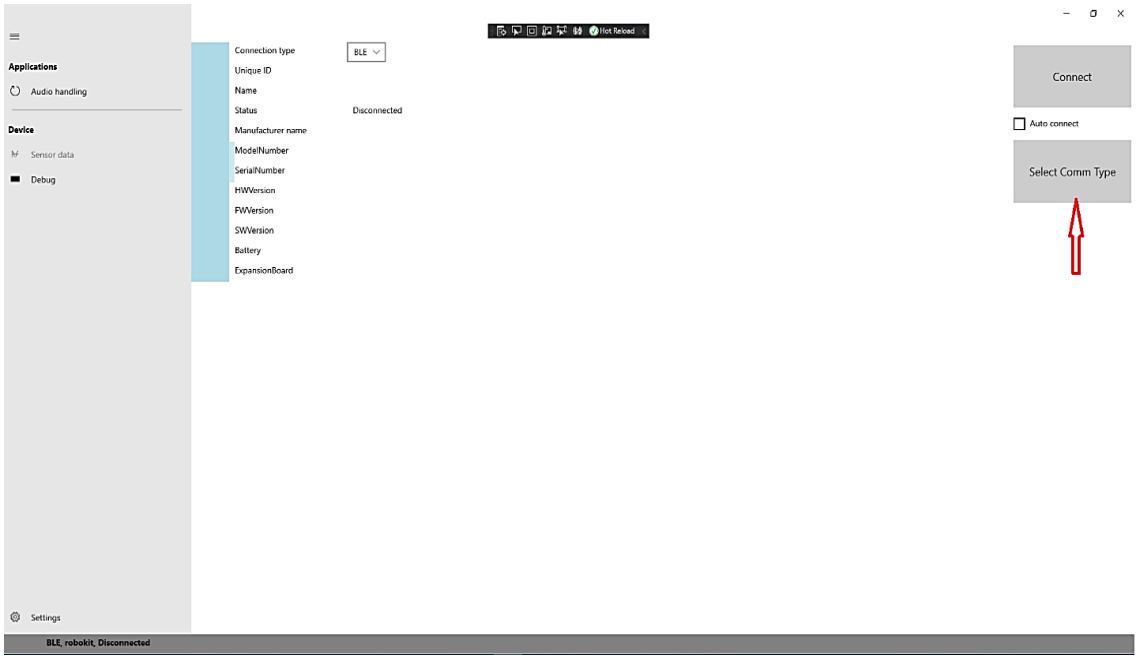
The detailed procedure to connect USB or BLE is detailed in sections 6.2 and 6.3.

6.2 USB CONNECTION

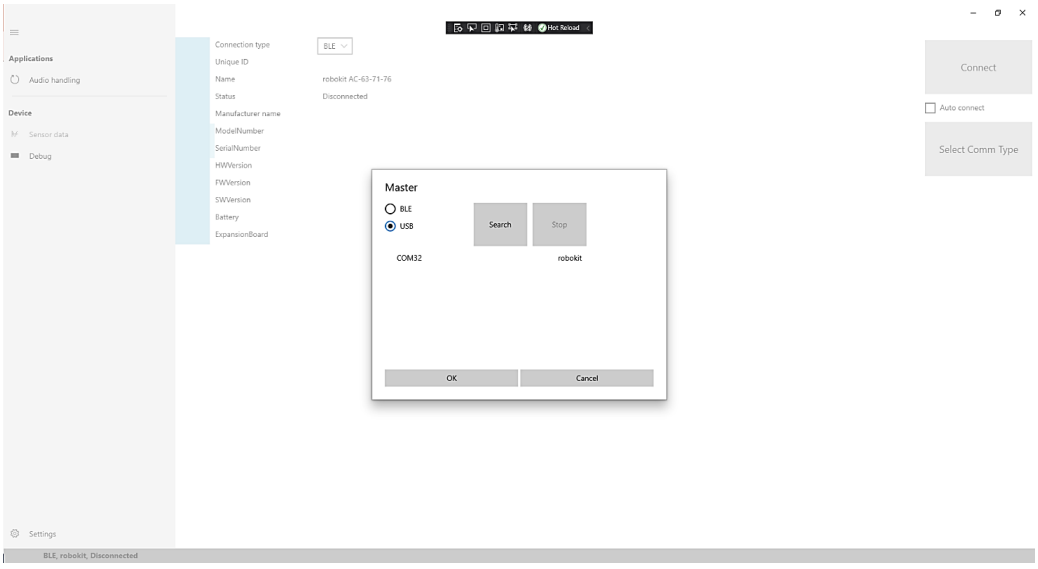
1. Click on “**Settings**.”



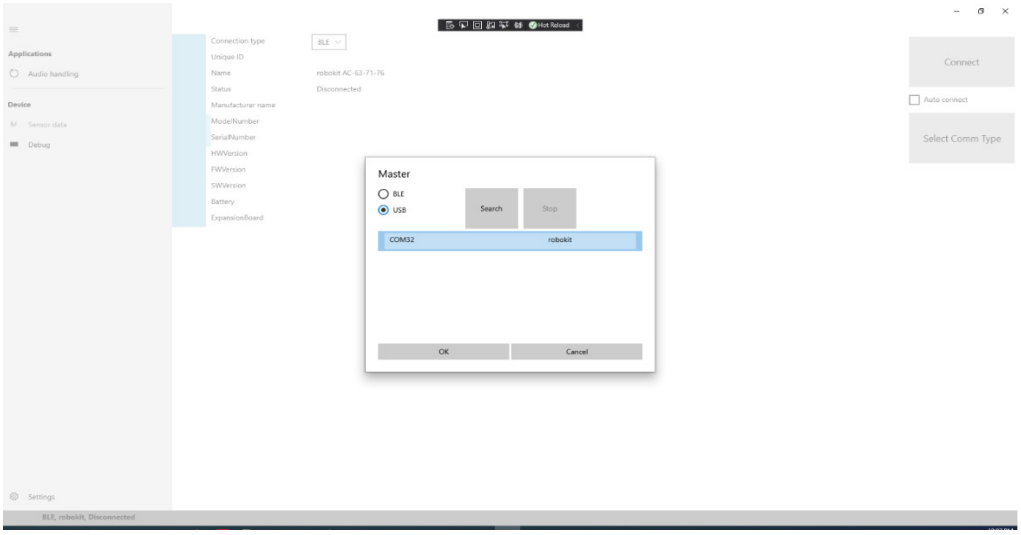
2. Click “**Select Comm Type**” to choose the communication type.



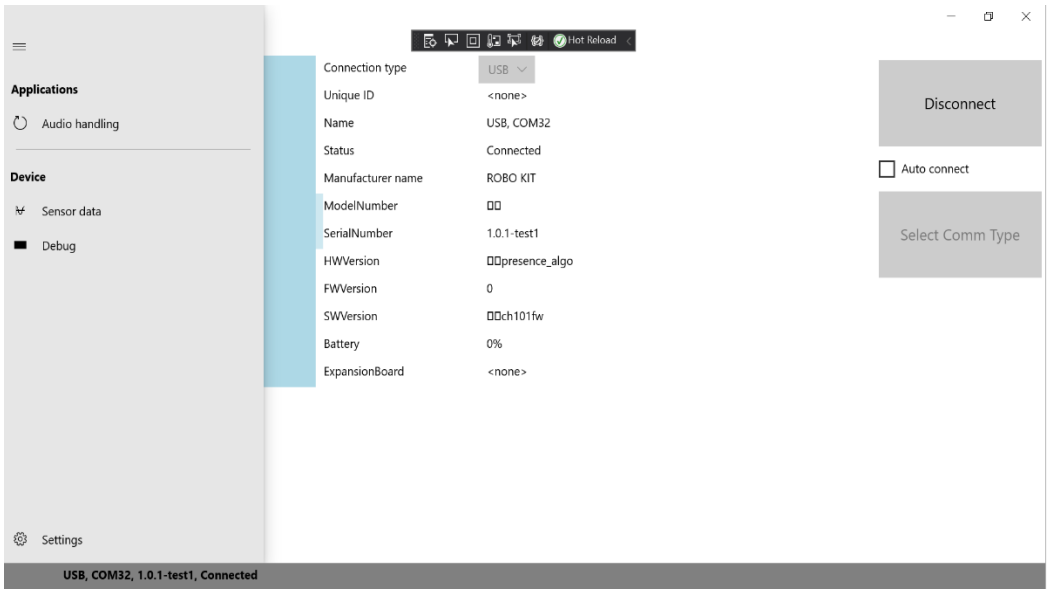
3. Click on “**Search**” to look for the available device. Upon finding matching VID and PID, device will be listed in the window.



4. Select the device and click **OK**.



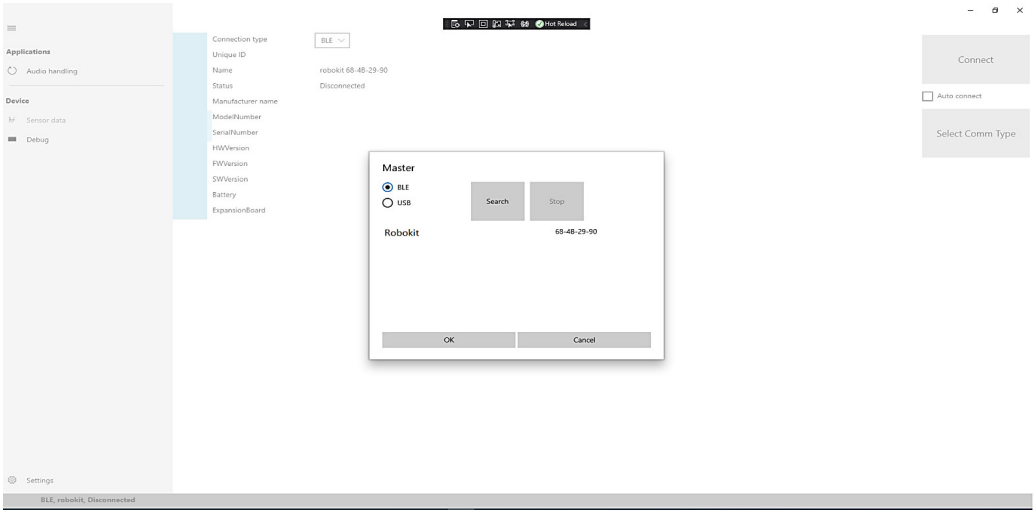
5. Once the connection is established, device parameters are populated on the UI as shown in Figure 18.



6.3 BLE CONNECTION

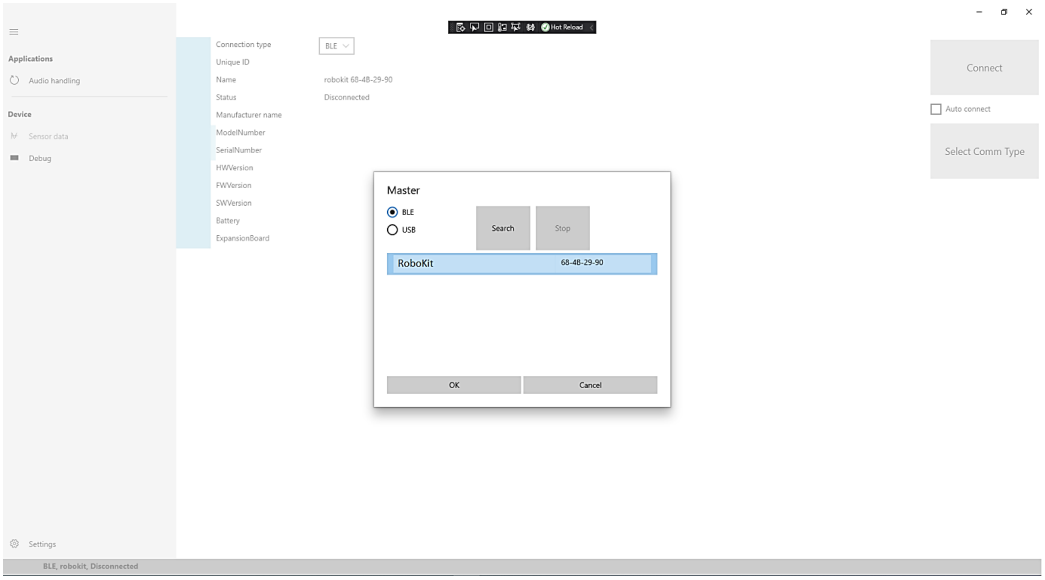
6.3.1 Bluetooth Discovery Routine:

1. Select option: “**BLE,”** and click on “**Search”** to discover the RoboKit1 module. Once a device is detected, “Robokit” will be listed for the users to choose.


Applications
Audio handling
Device
Sensor data
Debug
Settings
Connection type
Unique ID
Name
Status
Manufacturer name
ModelNumber
SerialNumber
HWVersion
FWVersion
SWVersion
Battery
ExpansionBoard
BLE
robokit 68-48-29-90
Disconnected
Hot Reload
Connect
Auto connect
Select Comm Type
Master
BLE
USB
Search
Stop
Robokit
68-48-29-90
OK
Cancel
BLE, robokit, Disconnected

7 DEVICE SELECTION

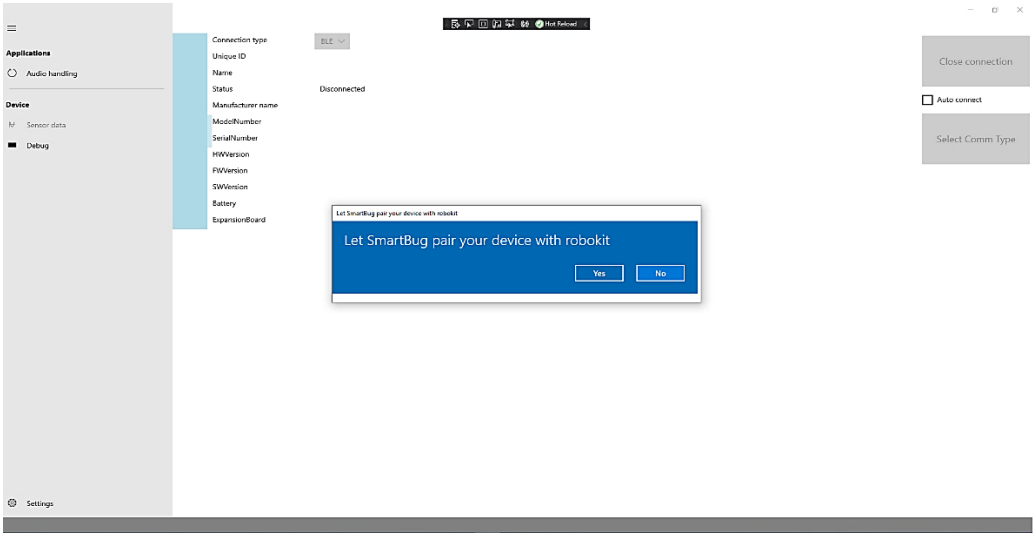
Select the device and click OK



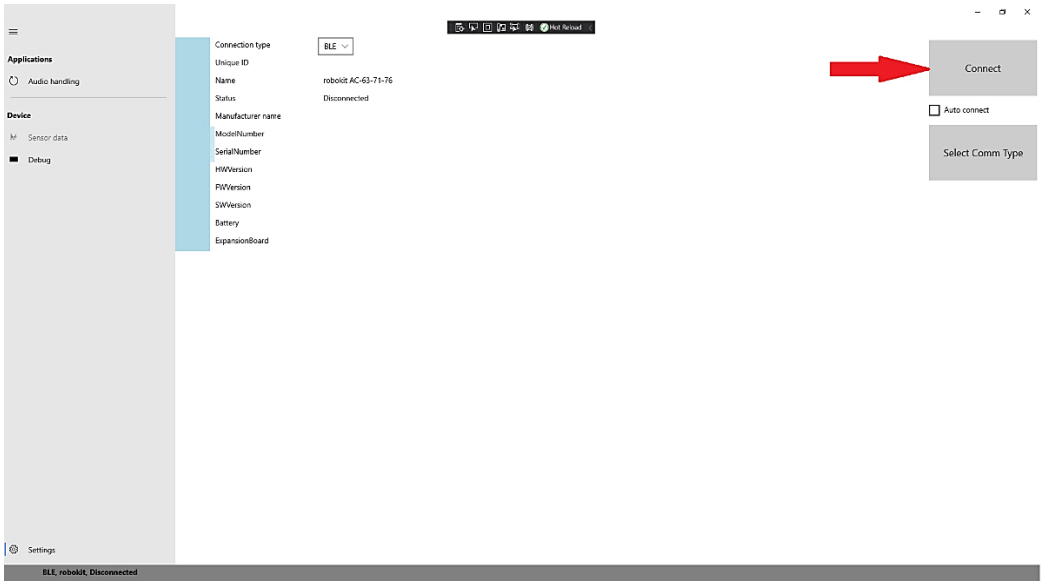
Once the BLE device is selected, press the **Connect** button to connect the RoboKit1 module to the RoboKit app. After clicking connect, wait a few seconds until the device parameters appear. If the **Auto connect** checkbox is checked, the selected device will connect automatically on the application restart.

7.1 CONNECTING BLE FOR THE FIRST TIME

When connecting BLE for the first time, the device should pair with the Windows system to begin the communication. Click "**Yes**" to proceed.



Successful device pairing takes roughly around 30s. Once it is done, press the "**Connect**" button to establish connection.

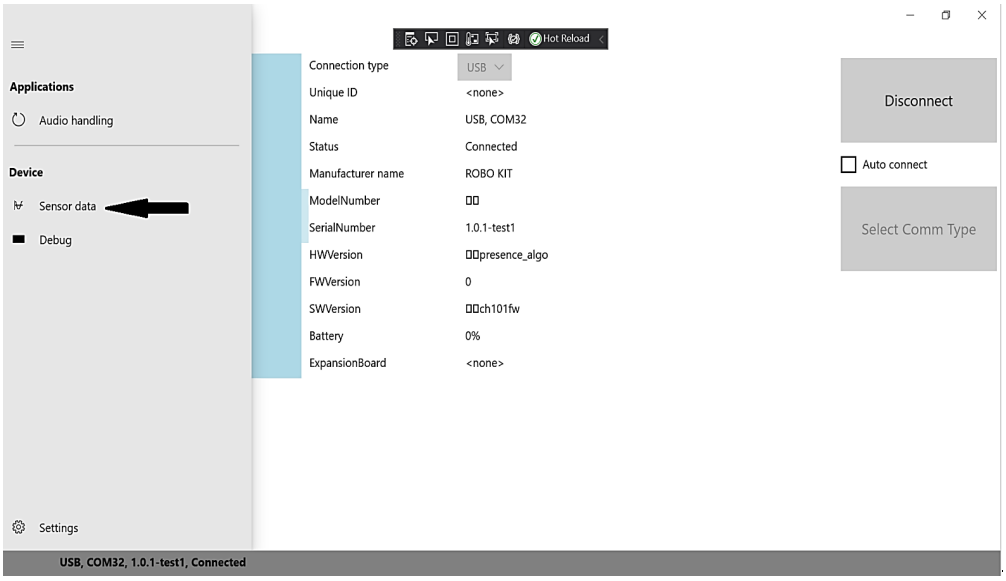


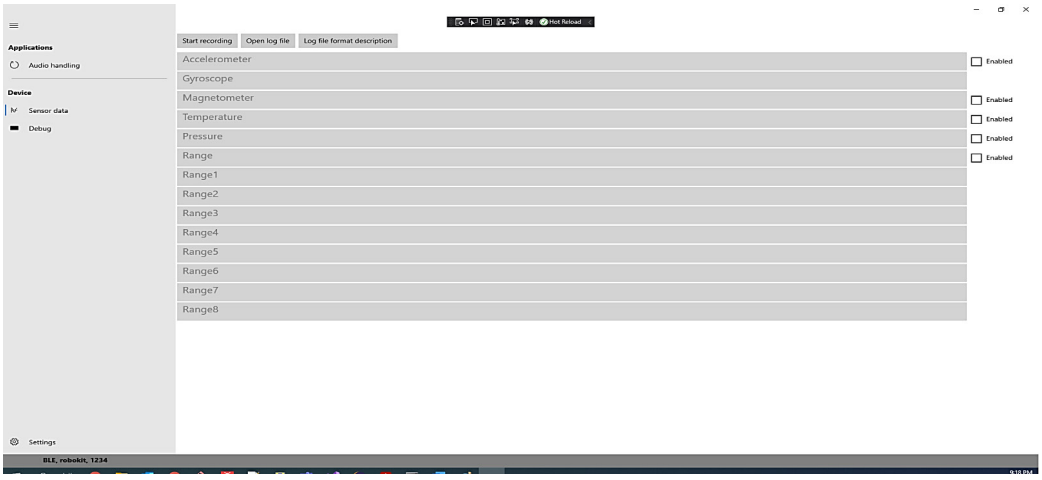
After clicking Connect, wait a few seconds until the device parameters appear. If the "**Auto connect**" checkbox is checked, the selected device will connect automatically on the application restart. Refer to Figure 18.

8 SENSOR DATA CAPTURE AND LOGGING

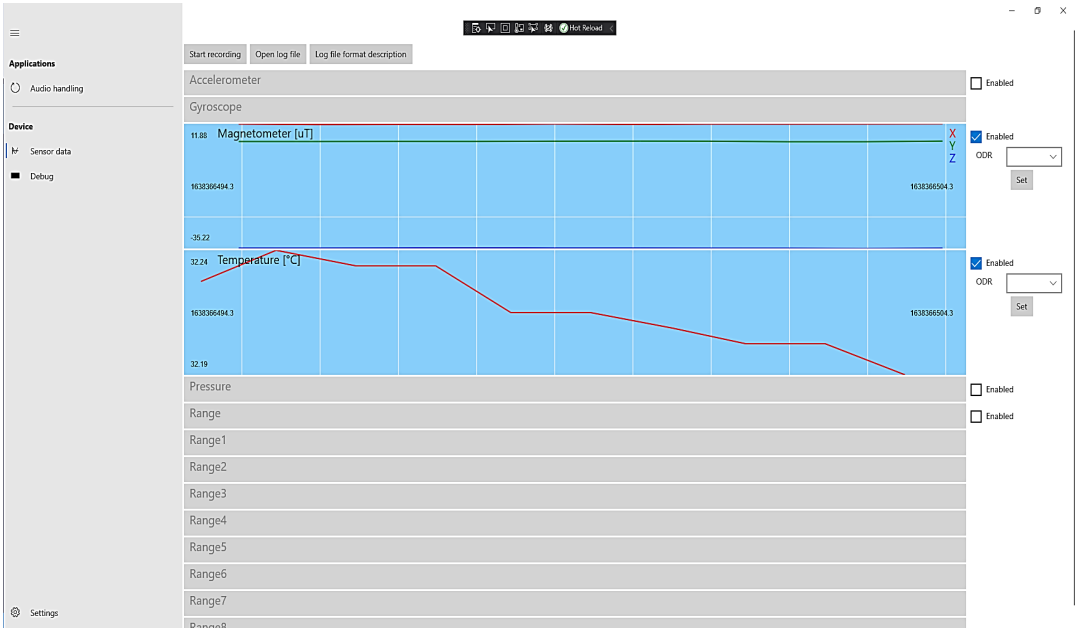
1. Click on “**Sensor data.**”

This window is the most important section of the app, as it allows users to stream multi-sensor data and record that data into log files.

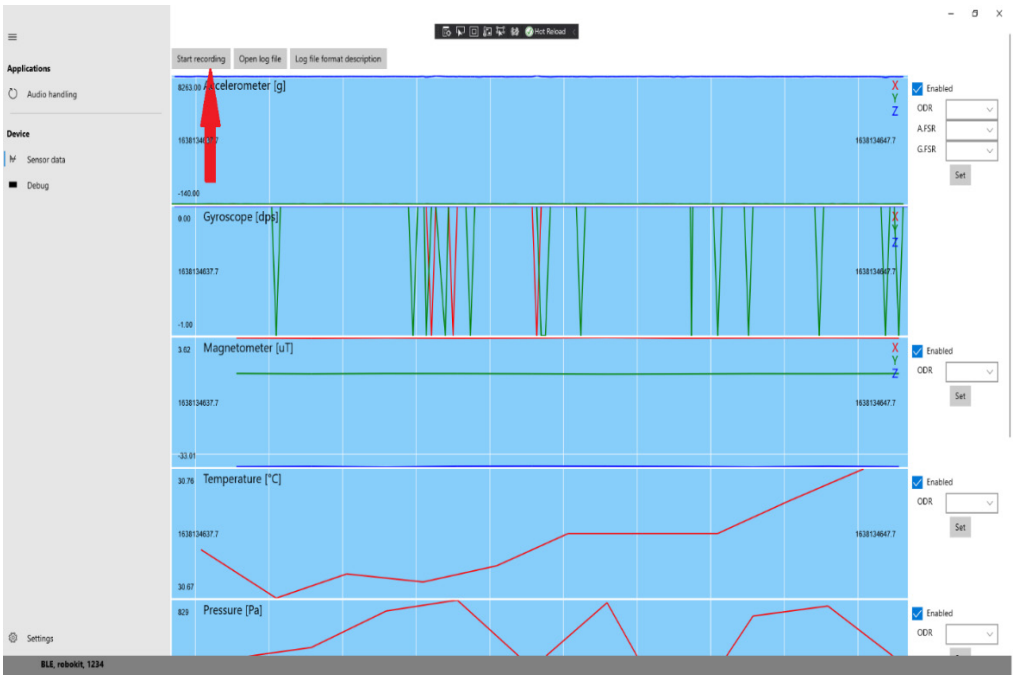




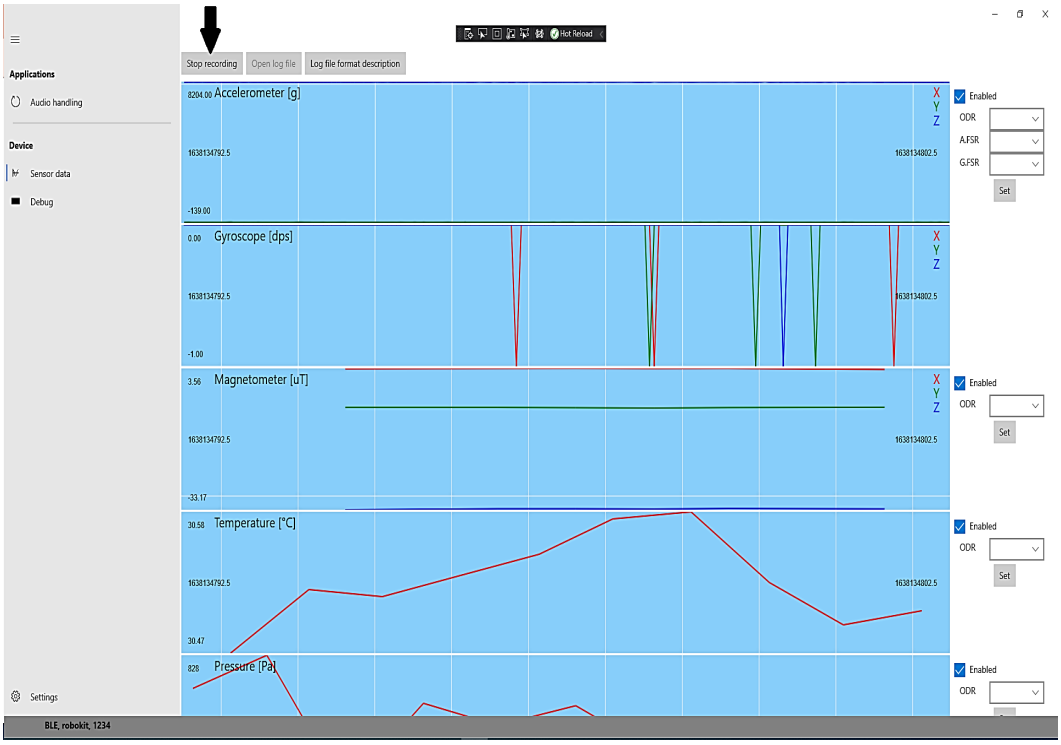
2. Select the sensor(s) of interest by checking the **Enabled** checkboxes to begin the plotting.


Hot Reload
Start recording
Open log file
Log file format description
Applications
Audio handling
Device
Sensor data
Debug
Settings
Accelerometer
Gyroscope
Magnetometer [uT]
11.88
-35.22
1638366494.3
1638366504.3
X
Y
Z
Temperature [°C]
32.24
32.19
Pressure
Range
Range1
Range2
Range3
Range4
Range5
Range6
Range7
Enabled
ODR
Set

3. Begin data recording by clicking on the "**Start recoding"** button. The app starts recording data for all sensors selected.



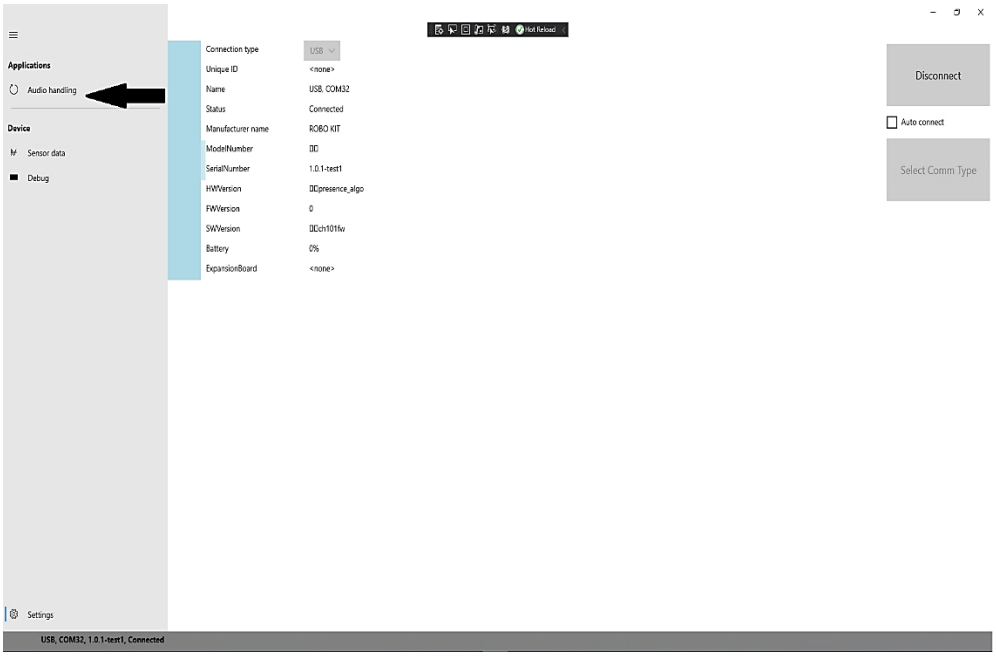
4. Stop recording the data by clicking on the “**Stop recording”** button. Recording for all sensors will stop.



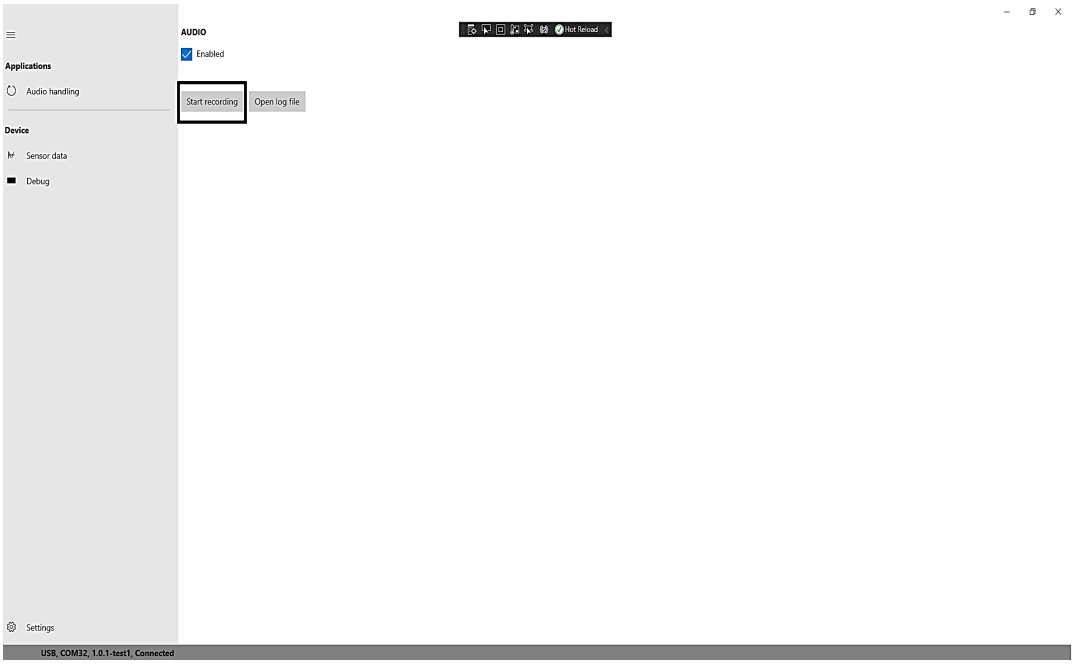
9 AUDIO DATA CAPTURE

The app also facilitates the capturing of audio data streamed by the RoboKit1 board. For audio data capture and validation follow the steps below:

1. Click on "**Audio Handling**"

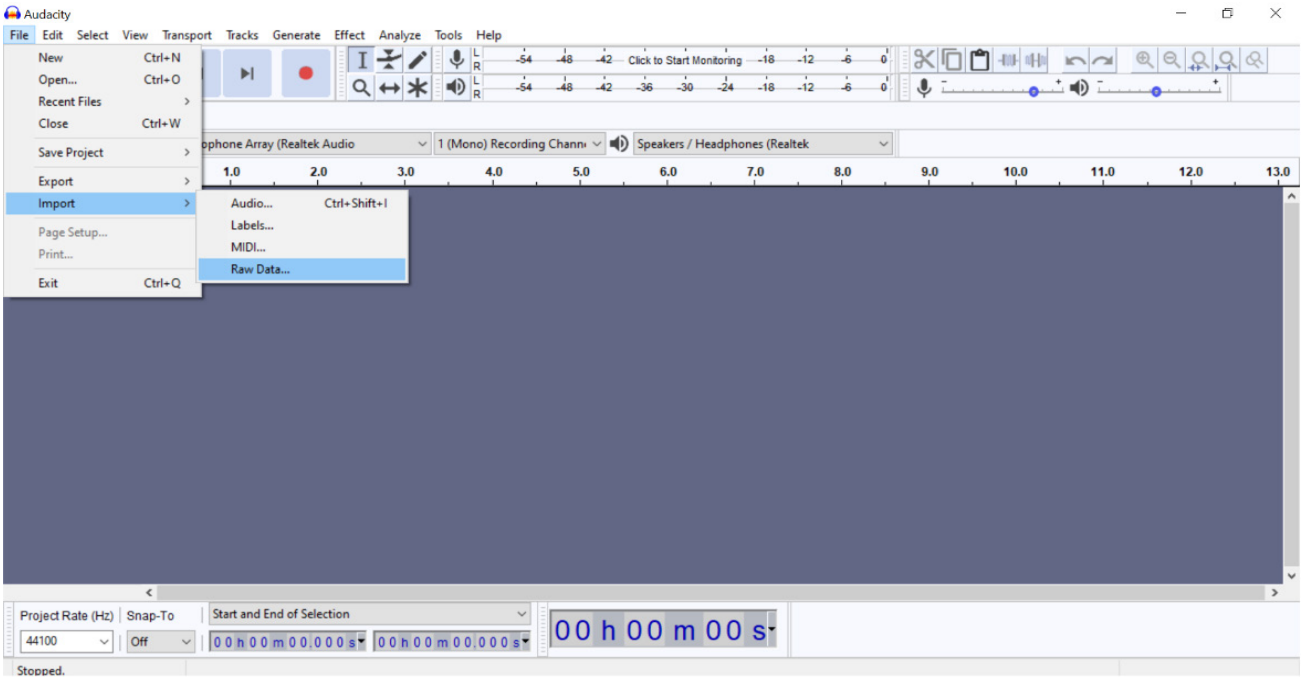


2. Select “**Enabled**” and click on “**Start Recording**”. Say “**HI TDK**” toward the board. The red LED will turn solid and around 1s of audio data will start streaming to the app and get recorded onto the file. The red LED will turn off after the streaming stops.

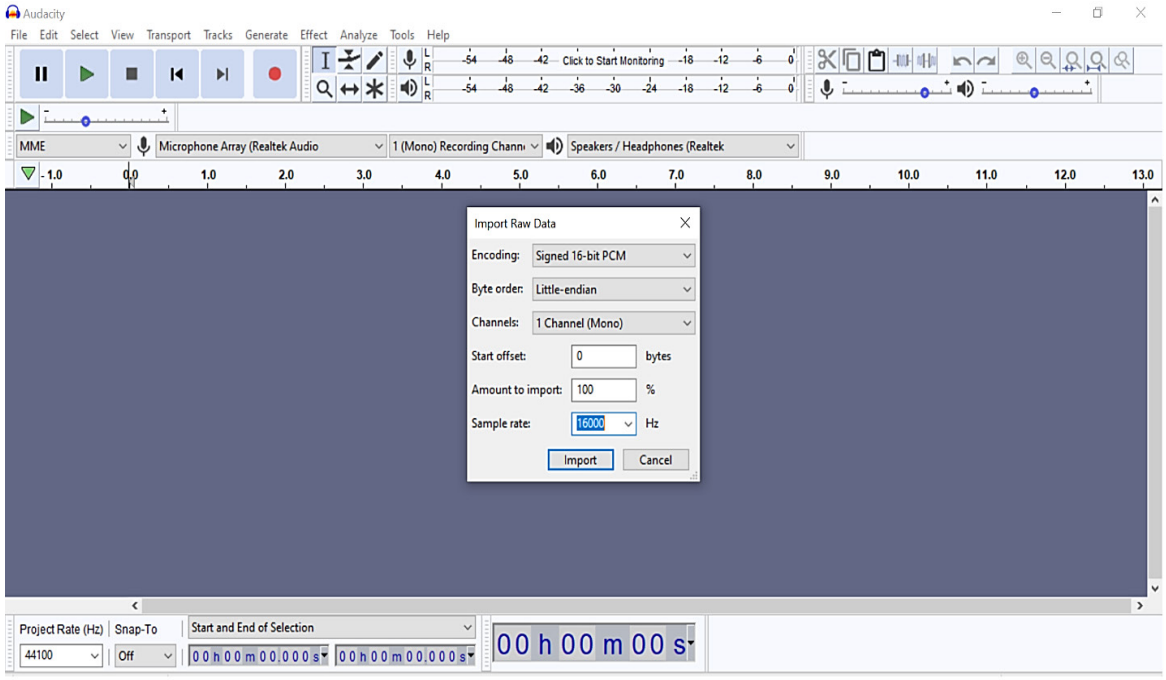


3. While the app facilitates the audio data capture, use the audio specialty tools like Audacity to verify the data validity as detailed below:

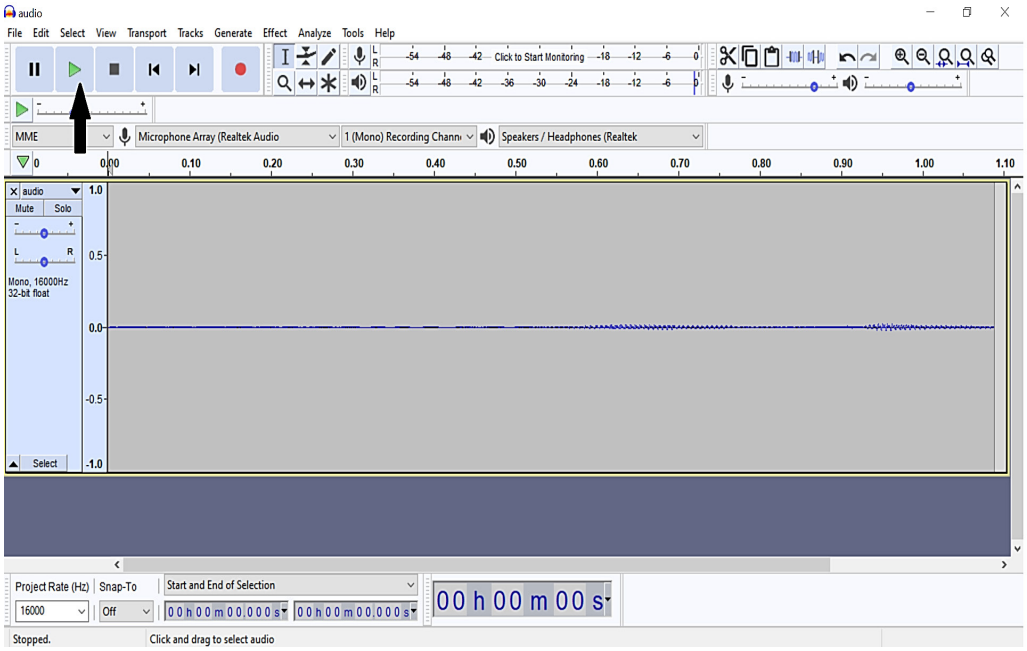
a. Launch the Audacity application on your computer, and follow the guideline as shown in Figure 30.



b. Click on “**Raw Data**” to launch a dialog to select audio file. Then choose the configuration as shown below and click “**Import.**”



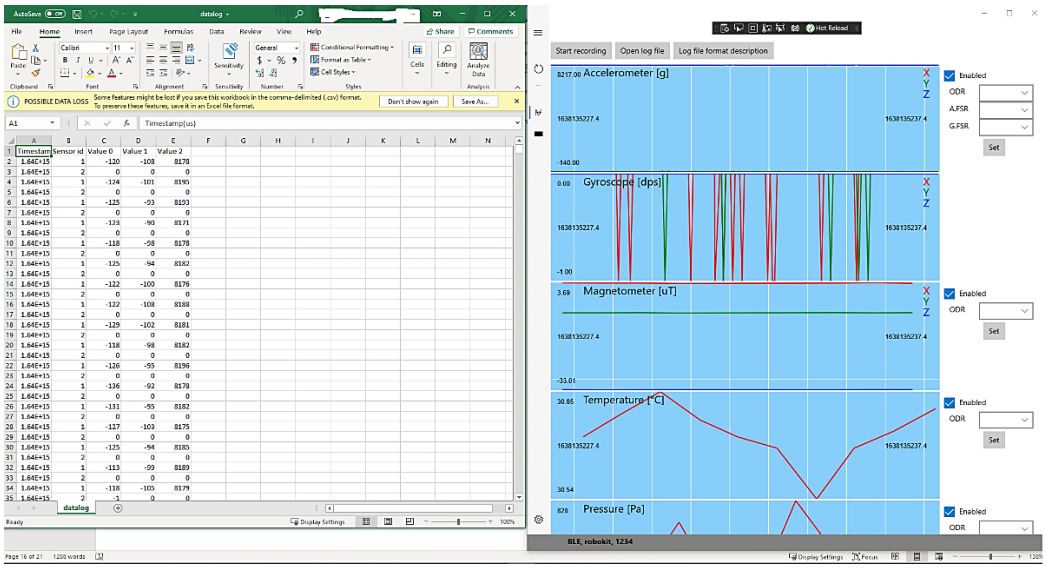
c. After importing the audio, click the **PLAY** button as indicated, and the recorded audio data will play.



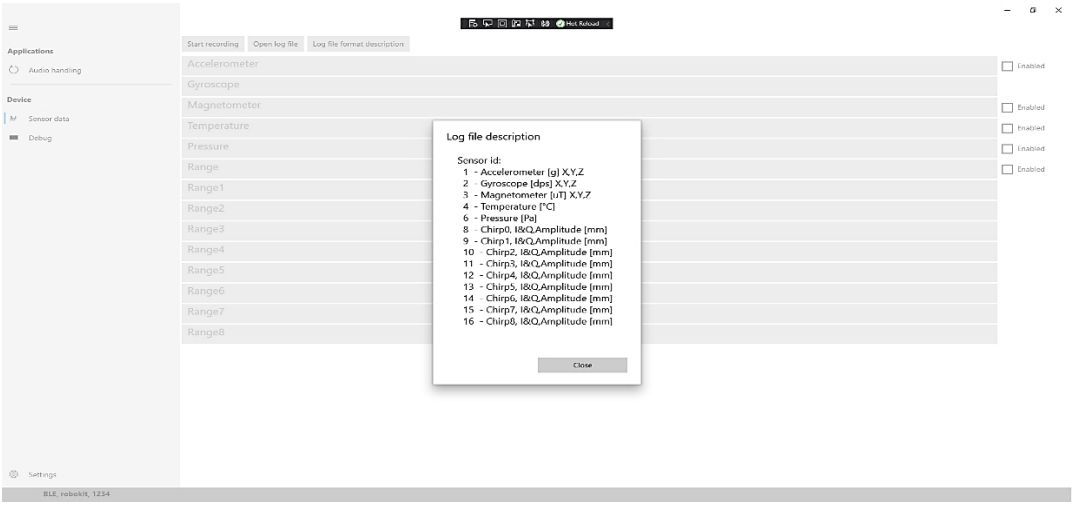
10 LOGGING

1. Click "**Open log file"** to view recorded data and a Microsoft Excel file opens automatically.

   Logs will be in the path
   C:\Users\XXXXXX\AppData\Local\Packages\InvenSenseInc.SmartBug_cg71h24es2qyy\LocalState



2. Click on the "**Log file format description**" to understand the sensor IDs and units in log file.



11 USE DEBUG MENU TO SEE MESSAGES LOG

All activities in the app are recorded in the debug window. This can help with troubleshooting and debugging the application.

Detailed app debug messages can be enabled by the "**Debug file enabled"** button. In this case, all communication messages will be logged into a file, located in the application data folder. The folder can be launched in File Explorer by clicking the "**Open application folder"** button.

12 REVISION HISTORY

REVISION DATE	REVISION	DESCRIPTION
01/16/2022	1.0	Initial release
02/28/2022	1.1	Added section 5

©2022 InvenSense. All rights reserved.