

# RoboKit User Guide

## Windows Application

## TABLE OF CONTENTS

1	RoboKit1 Module .....	4
2	Installing & Running RoboKit .....	5
2.1	Installing from local directory .....	5
3	Connecting ESP32 with RoboKit .....	8
4	Updating ESP32 firmware .....	10
5	Loading RoboKit Atmel Binary .....	11
6	Connecting to the RoboKit.....	14
6.1	Switching between USB0 and Bluetooth mode .....	14
6.2	USB Connection .....	15
6.3	BLE Connection .....	17
6.3.1	Bluetooth Low Energy Discovery Routine: .....	17
6.4	Bluetooth CLASSIC(BT) Connection .....	18
6.4.1	BT Discovery Routine:.....	18
6.4.2	Device Selection.....	19
6.5	Connecting BLE for the first time.....	19
6.6	Connecting BT for the first time .....	20
7	Sensor Data capture and logging.....	22
8	Audio data capture .....	25
9	Logging.....	28
10	Use Debug menu to see log messages .....	29
11	Device firmware upgrade .....	30
12	Motor control .....	31
13	Revision History .....	32

**TABLE OF FIGURES**

Figure 1. RoboKit1 Top View ..... 4

Figure 2. RoboKit1 Bottom View ..... 4

Figure 3. Installation ..... 5

Figure 4. Certificate Authorization ..... 6

Figure 5. Developer Settings ..... 6

Figure 6. Certificate Installation Done ..... 7

Figure 7. Launch App in windows ..... 7

Figure 8. RoboKit and ESP32 Connections ..... 8

Figure 9. ESP32 Firmware Upgrade Process ..... 10

Figure 10. Atmel Project files ..... 11

Figure 11. Selecting Properties ..... 12

Figure 12. Select Start Debugging ..... 13

Figure 13. Switching between USB and BLE ..... 14

Figure 14. Application Main Screen ..... 15

Figure 15. Communication Type Selection ..... 16

Figure 16. Detecting USB device ..... 16

Figure 17. Select USB Device ..... 17

Figure 18. Connect Device ..... 17

Figure 19. BLE Search Page ..... 18

Figure 20. BT Search Page ..... 18

Figure 21. BT/BLE Device Selection ..... 19

Figure 22. BLE Pairing ..... 19

Figure 23. BLE Device Connection ..... 20

Figure 24. BT Toast screen ..... 20

Figure 25. BT pairing accept screen ..... 21

Figure 26. BT close screen ..... 21

Figure 27. Connect Device ..... 22

Figure 28. Sensor Selection ..... 22

Figure 29. Sensor Data Plot ..... 23

Figure 30. Sensor Data Recording ..... 23

Figure 31. Stop Data Recording ..... 24

Figure 32. Audio Data Handling ..... 25

Figure 33. Audio Data Recording ..... 25

Figure 34. Import Audio Data to Audacity ..... 26

Figure 35. Audacity Audio Settings ..... 26

Figure 36. Audio Data Playback in Audacity ..... 27

Figure 37. Sensor Data Logging ..... 28

Figure 38. Sensor Data Log File Format ..... 28

Figure 39. Debug screen ..... 29

Figure 40. Firmware Upgrade ..... 29

Figure 41. Firmware Upgrade Start screen ..... 29

Figure 42. Motor Control ..... 29

## 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.

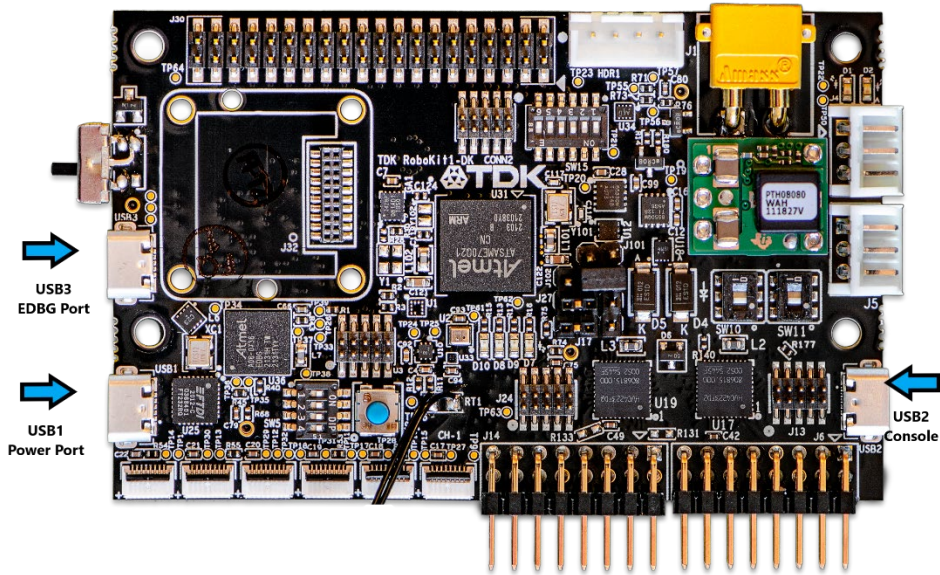


Figure 1. RoboKit1 Top View

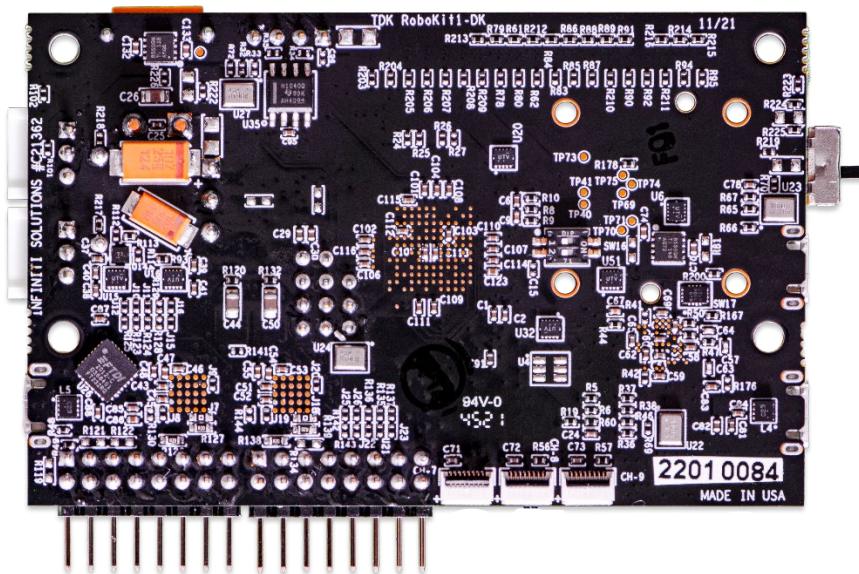


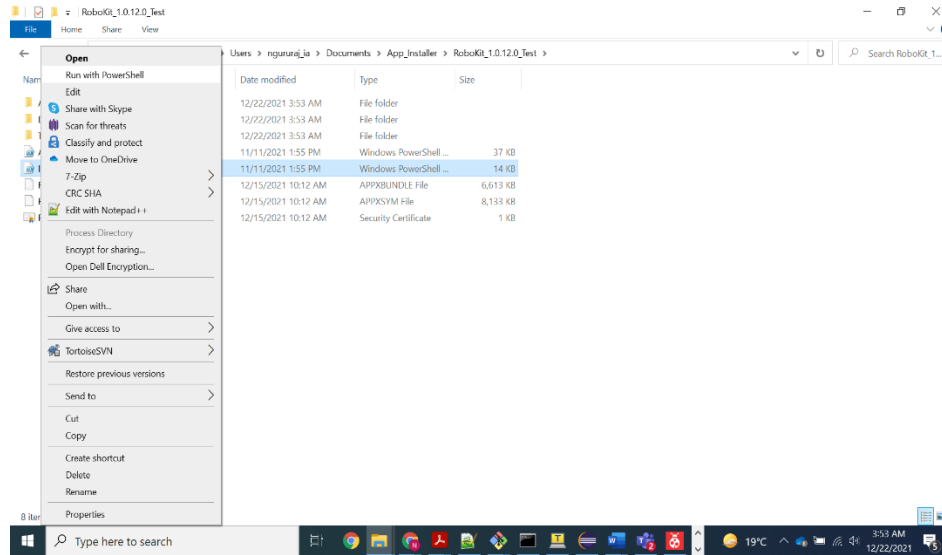
Figure 2. RoboKit1 Bottom View

## 2 INSTALLING & RUNNING ROBOKIT

### 2.1 INSTALLING FROM LOCAL DIRECTORY

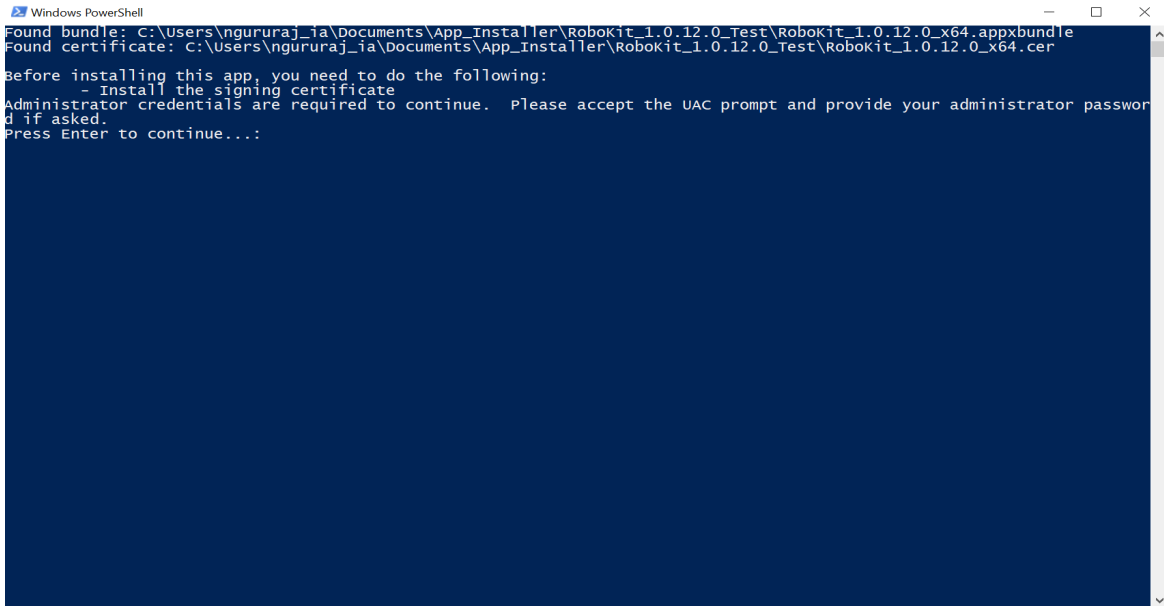
Follow the process below to install the application from the local directory:

1. Right click on the “install” file and click on “Run with PowerShell”

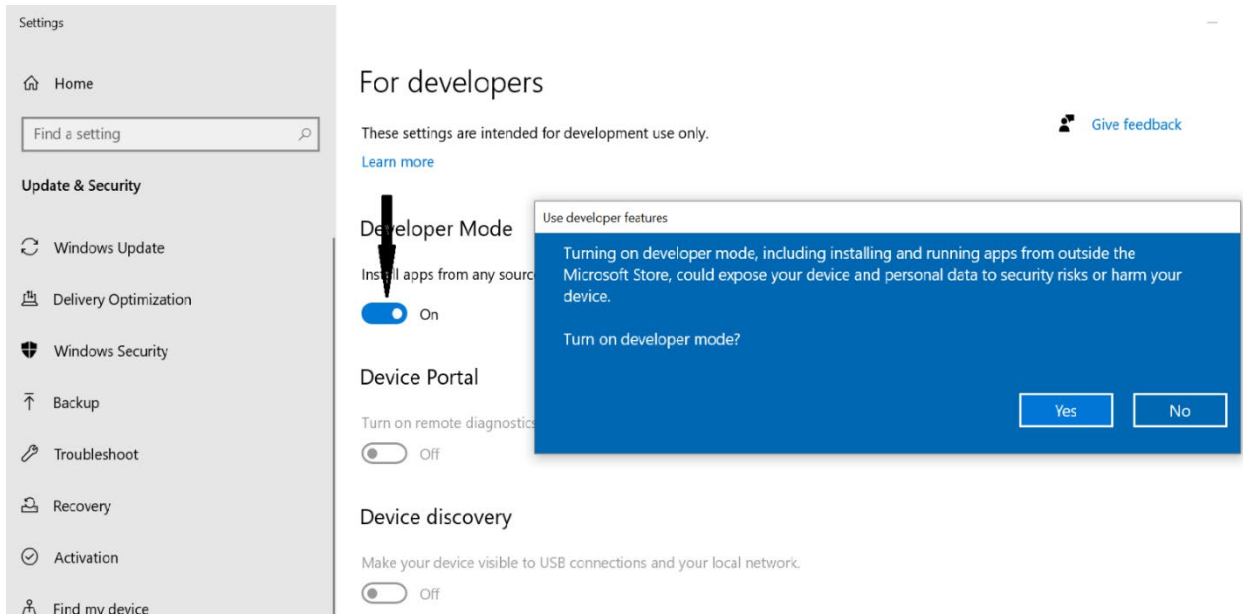


**Figure 3. Installation**

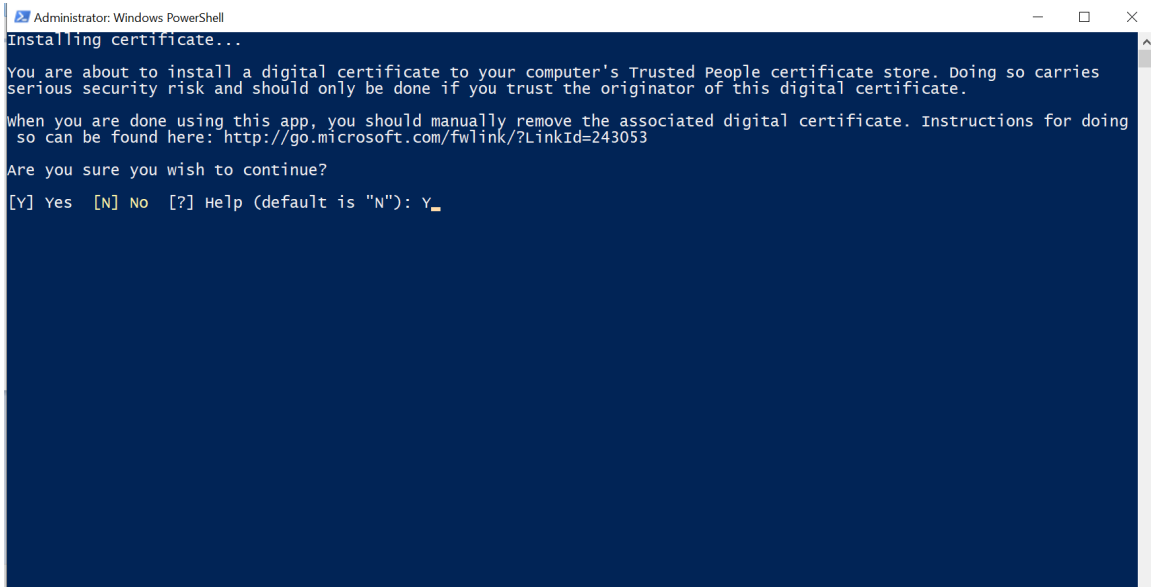
2. Press “Enter” to provide administrator access to allow installation of the certificates related to the application.



**Figure 4. Certificate Authorization**

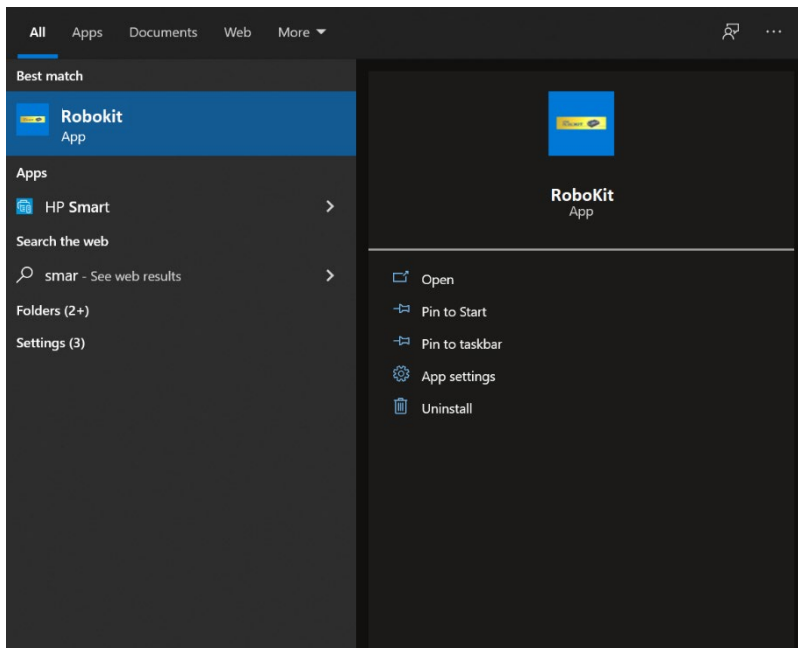


**Figure 5. Developer Settings**



**Figure 6. Certificate Installation Done**

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



**Figure 7. Launch App in windows**



### 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 or Bluetooth classic 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:

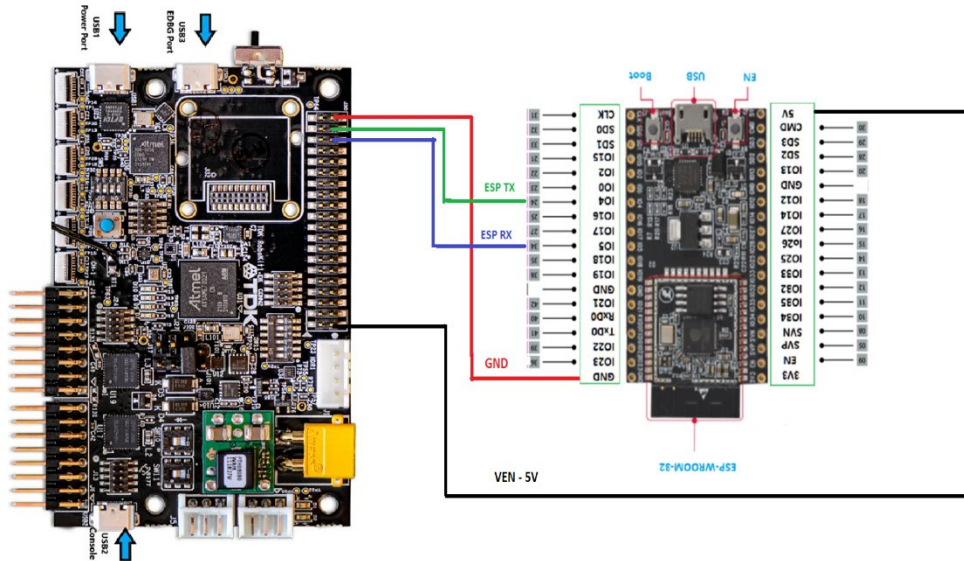


Figure 8. RoboKit and ESP32 Connections



---

**Note:** The ESP32 might need [CP210x USB to UART Bridge Virtual COM Port \(VCP\) drivers](#) which need to be installed depending on the user's OS platform. The drivers are available as part of the firmware download package on the 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 firmware for ESP32 BLE and BT are provided along with the release package folder under apps. Choose the appropriate firmware based on the connectivity needed.

Follow the steps below to update the firmware:

1. Unzip the `bt_spp_acceptor.zip` or `esp_ble_firmware.zip` file
2. Connect to ESP32 from your Windows Machine with USB. Identify the Serial Port (Ex. COM1) using Device manager.
3. Open Windows PowerShell and navigate to the folder `bt_spp_acceptor` or `esp_ble_firmware`
4. Execute the command in section a or section b below, depending on BT or BLE connectivity

a. For BT

```
./flash-binary.sh <serial-com-port> bt_spp_acceptor_vx.y.z.bin
```

Eg: `./flash-binary.sh COM2 bt_spp_acceptor_vx.y.z.bin`

b. For BLE

```
./flash-binary.sh <serial-com-port> esp_ble_firmware_vx.y.z.bin
```

Eg: `./flash-binary.sh COM2 esp_ble_firmware_vx.y.z.bin`

where,

<code>serial-com-port</code>	Serial Port connected to ESP32
<code>esp_ble_firmware_vx.y.z.bin</code>	Latest BLE firmware, x, y and z are version numbers
<code>bt_spp_acceptor_vx.y.z.bin</code>	Latest BT firmware, x, y and z are version numbers

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

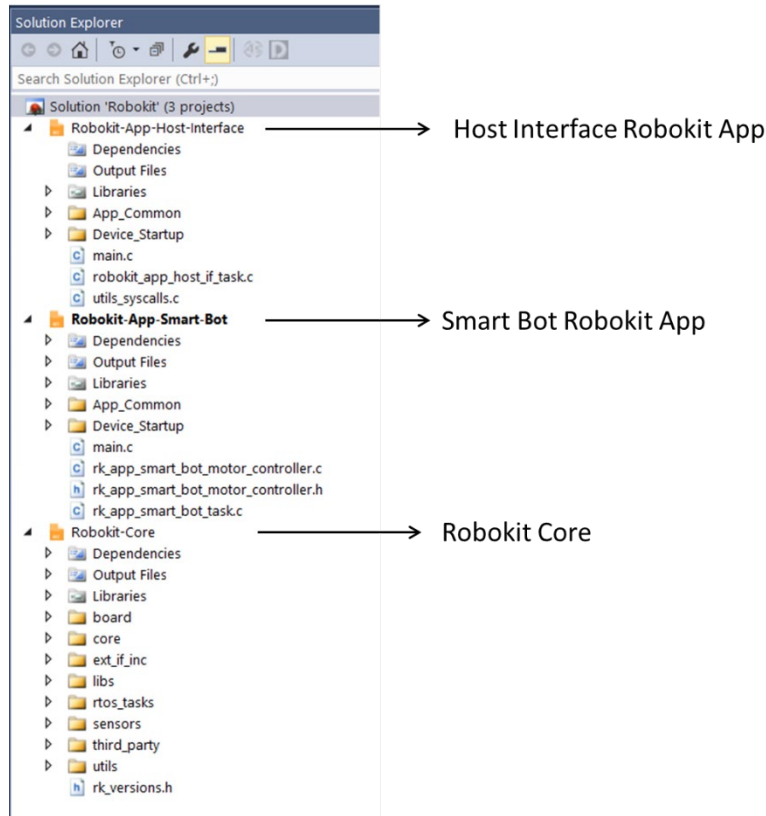
```
esptool.py v3.1-dev
Serial port COM16
Connecting...
Chip is ESP32-D0WD (revision 1)
Features: WiFi, BT, Dual Core, 240MHz, VRef calibration in efuse, Coding Scheme None
Crystal is 40MHz
MAC: 98:cd:ac:62:d3:28
Uploading stub...
Running stub...
Stub running...
Changing baud rate to 460800
Changed.
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...
```

Figure 9. ESP32 Firmware Upgrade Process

6. Restart the ESP32 with a power cycle.

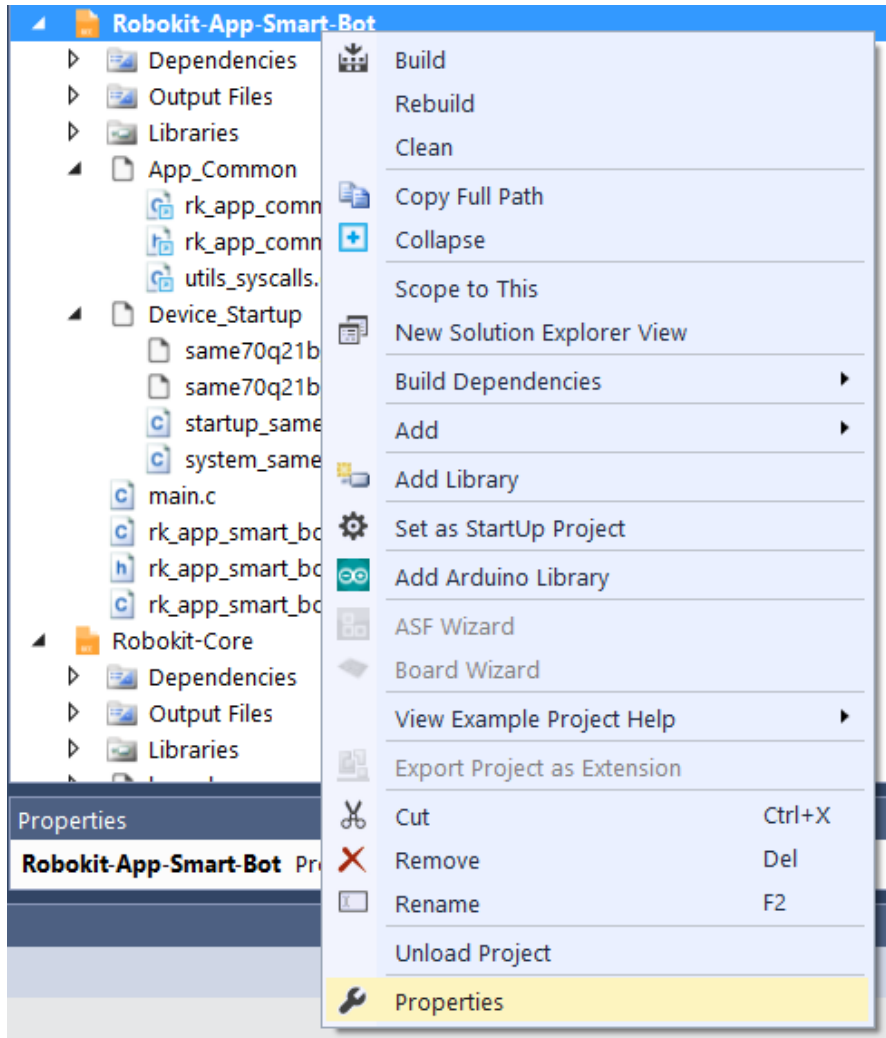
## 5 LOADING ROBOKIT ATMEL BINARY

The main solution file, “Robokit.sln”, 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 then compile it.



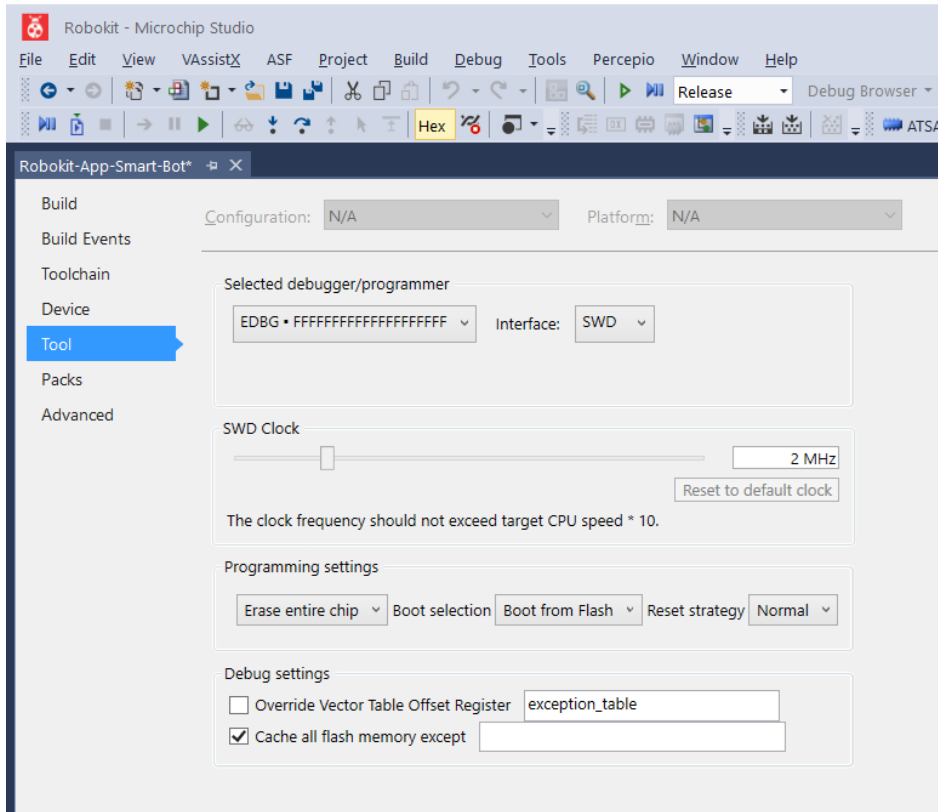
**Figure 10. Atmel Project files**

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



**Figure 11. Selecting Properties**

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



**Figure 12. Select Start Debugging**

## 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:

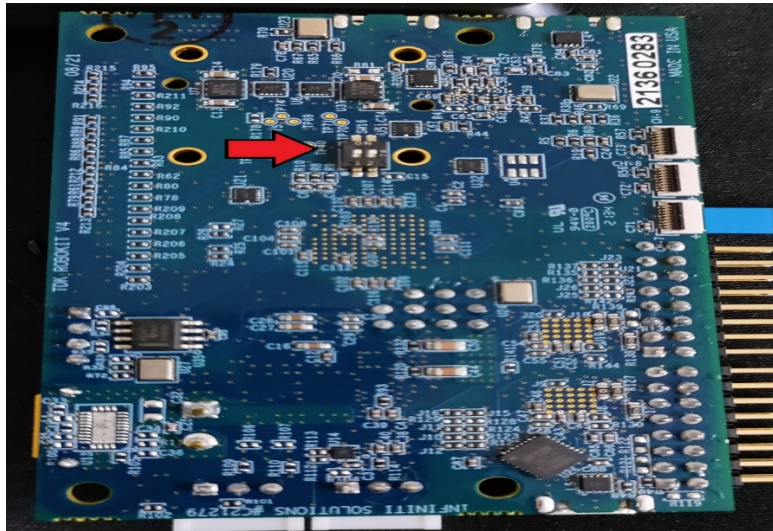
- Industrial IMU
- Chirp CH101(3x)
- Chirp Connector Cables(3x)
- Windows & Android App for Data collection
- Open-sourced based Board Design files, Firmware and ROS Drivers

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 USB0 AND BLUETOOTH 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.



**Figure 13. Switching between USB and BLE**

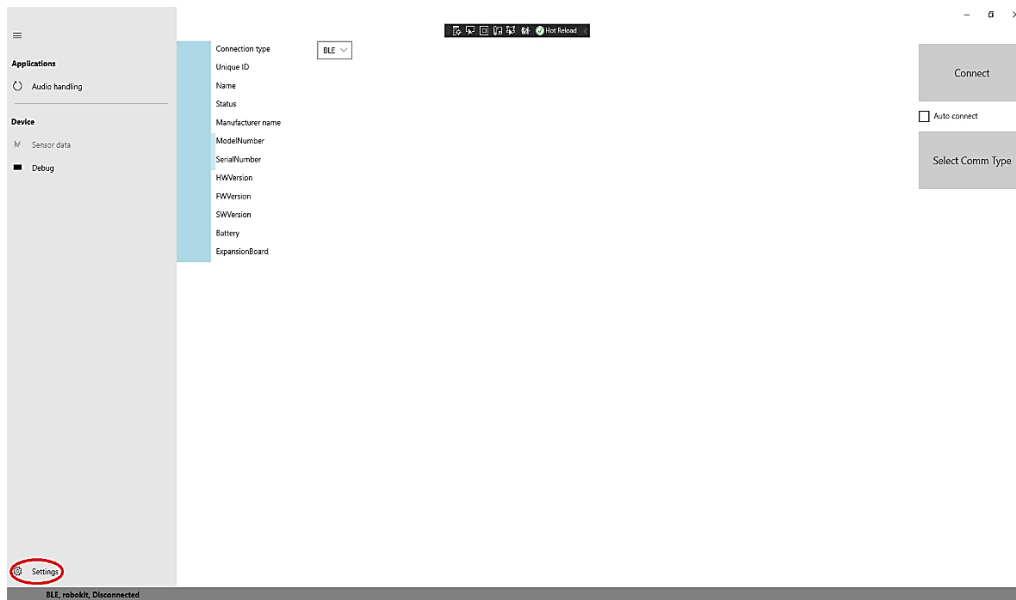
To connect in Bluetooth mode, switches 1&2 should be in OFF position.

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

The detailed procedure to connect USB or Bluetooth is detailed in sections 6.2, 6.3, and 6.4.

## 6.2 USB CONNECTION

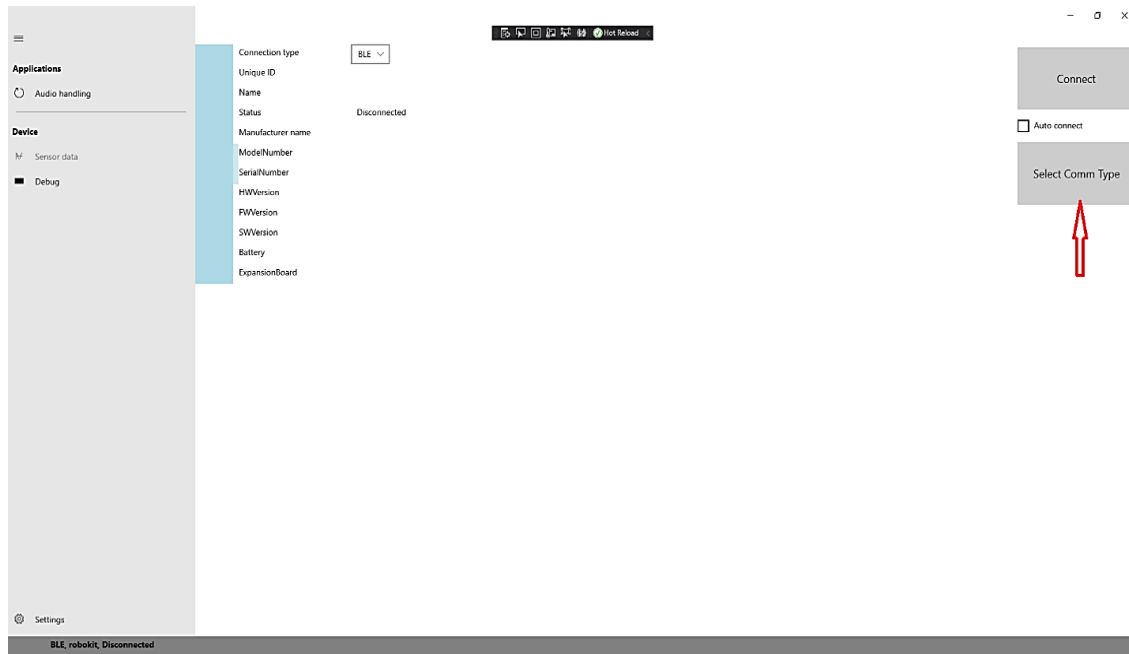
1. Click on **“Settings.”**



**Figure 14. Application Main Screen**

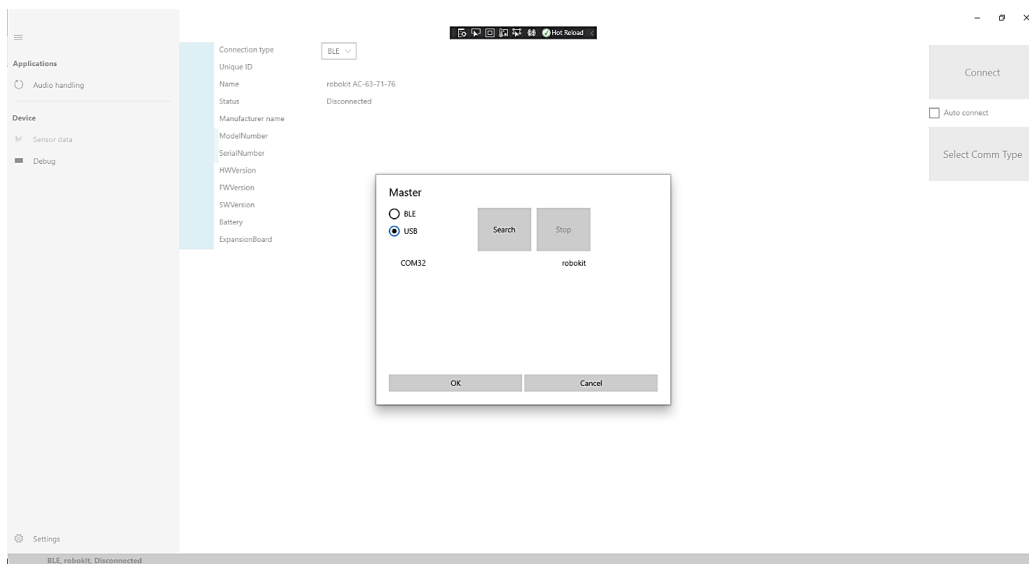


2. Click **“Select Comm Type”** to choose the communication type.



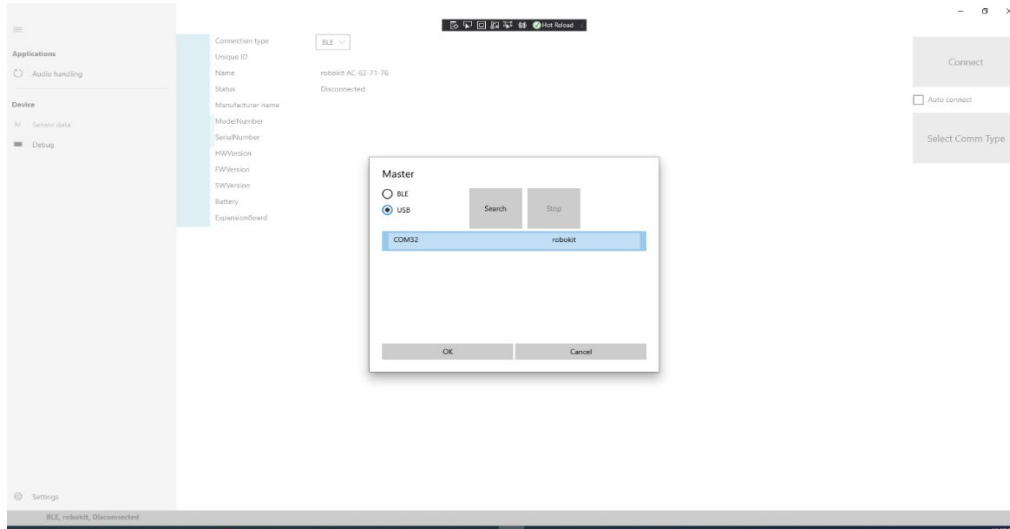
**Figure 15. Communication Type Selection**

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



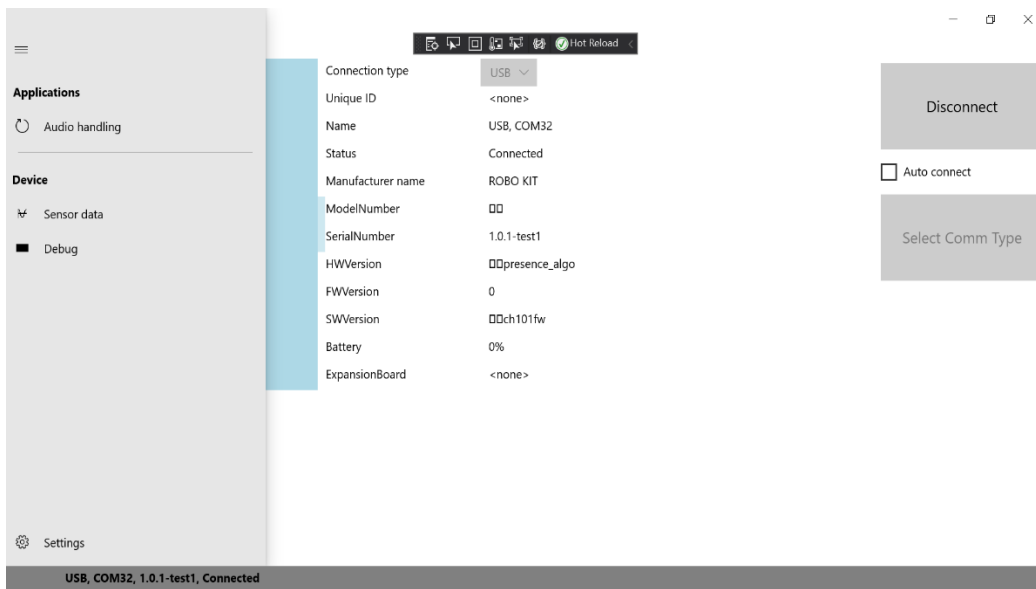
**Figure 16. Detecting USB device**

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



**Figure 17. Select USB Device**

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

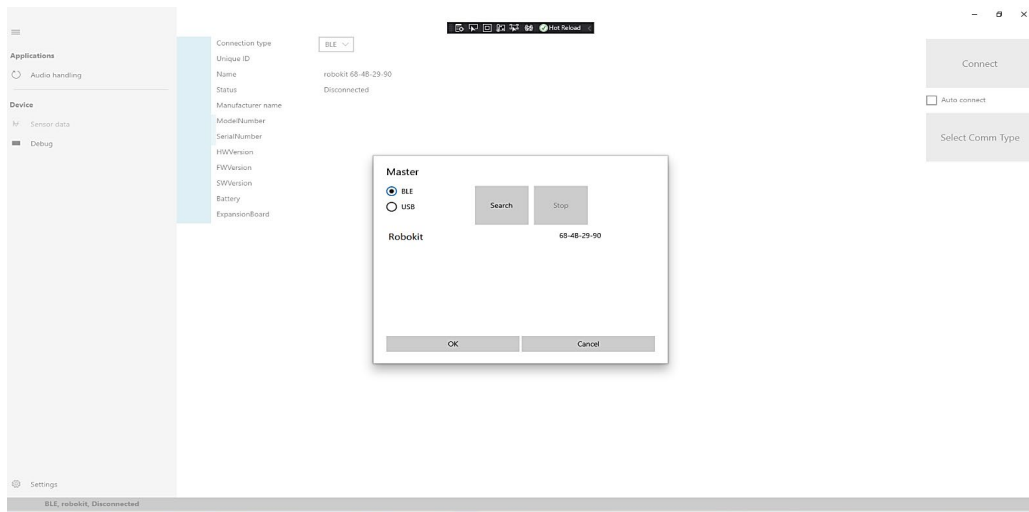


**Figure 18. Connect Device**

## 6.3 BLE CONNECTION

### 6.3.1 Bluetooth Low Energy Discovery Routine:

- Select option: **“BLE,”** as shown in Figure 16, and click on **“Search”** to discover the RoboKit1 module. Once a device is detected, **“Robokit”** will be listed for the users to choose.

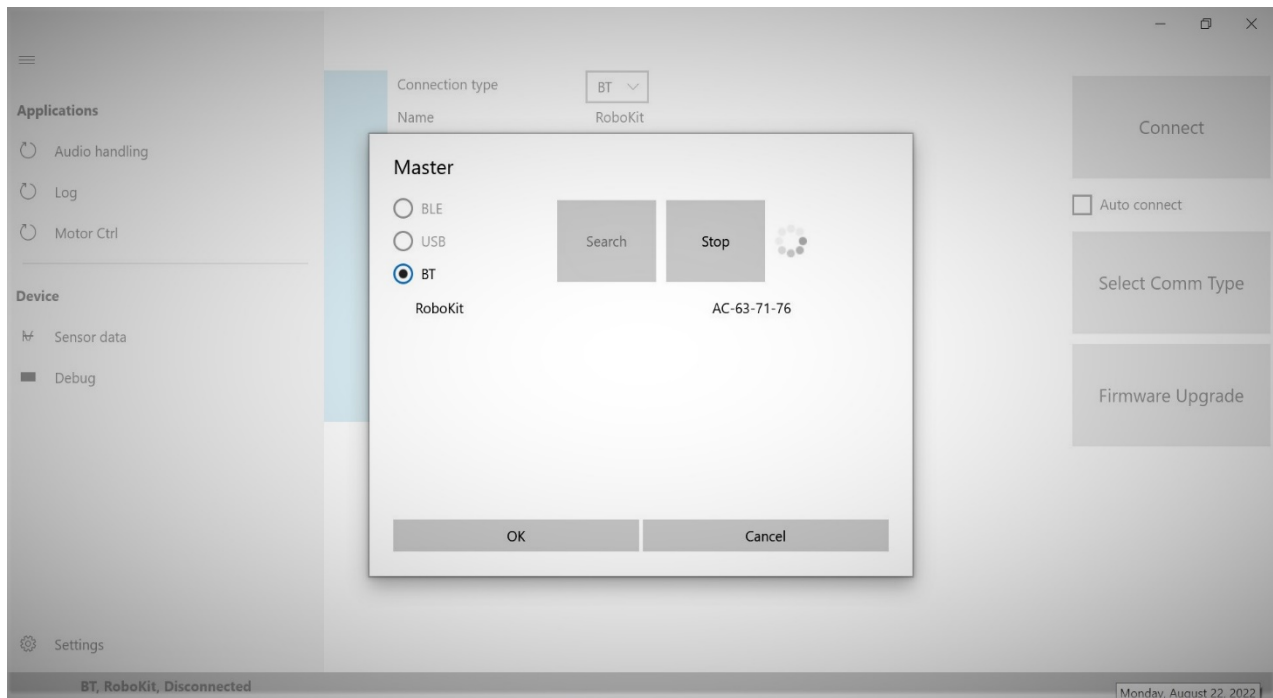


**Figure 19. BLE Search Page**

## 6.4 BLUETOOTH CLASSIC(BT) CONNECTION

### 6.4.1 BT Discovery Routine:

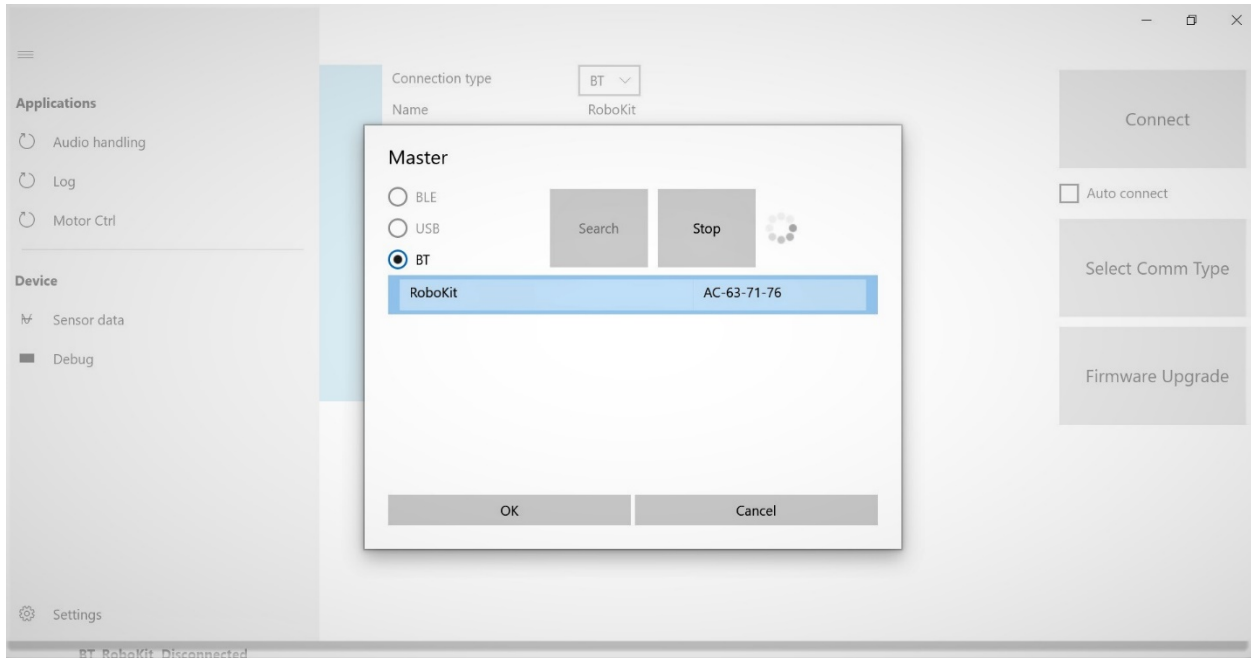
1. Select option: **“BT,”** and click on **“Search”** to discover the RoboKit1 module. Once a device is detected, **“Robokit”** will be listed for the users to choose.



**Figure 20. BT Search Page**

### 6.4.2 Device Selection

Select the device and click OK.

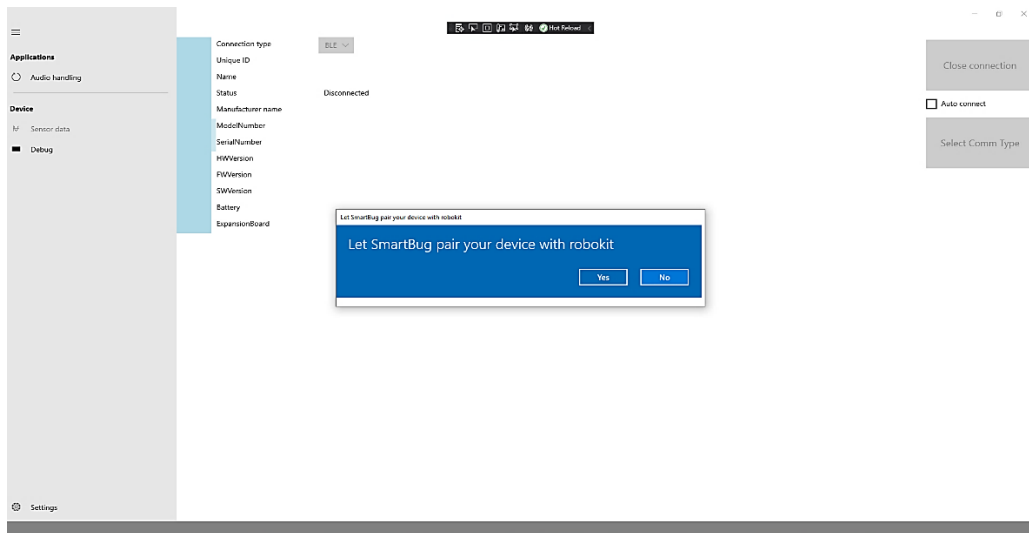


**Figure 21. BT/BLE Device Selection**

Once the BT/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.

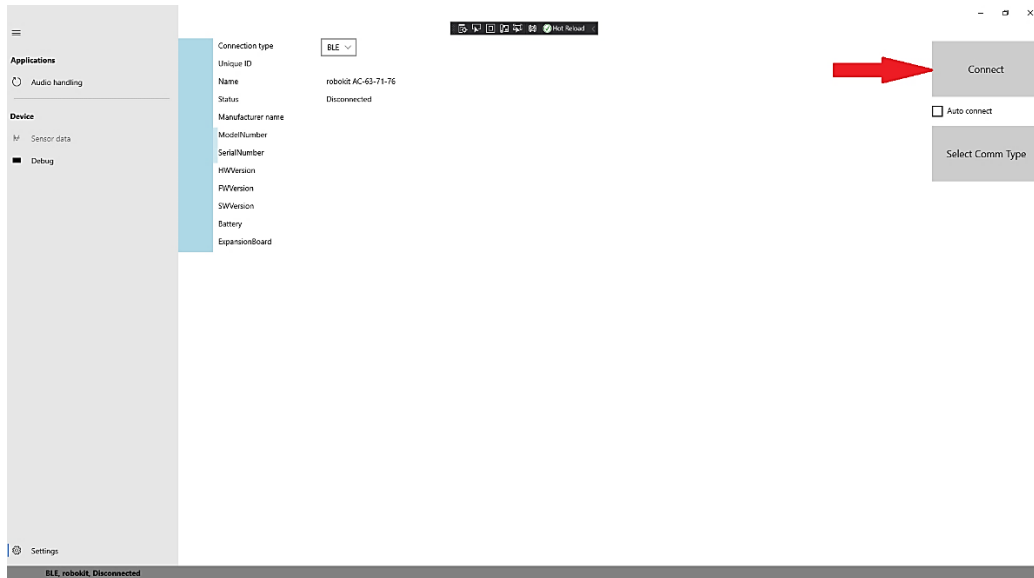
### 6.5 CONNECTING BLE FOR THE FIRST TIME

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



**Figure 22. BLE Pairing**

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

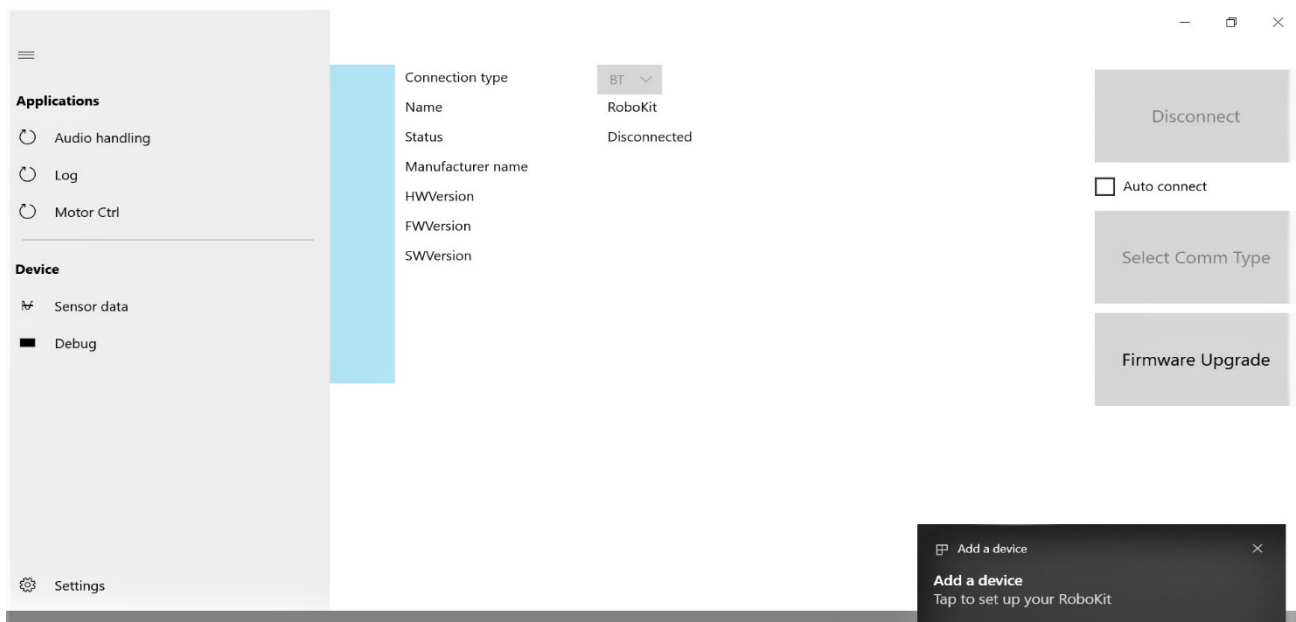


**Figure 23. BLE Device Connection**

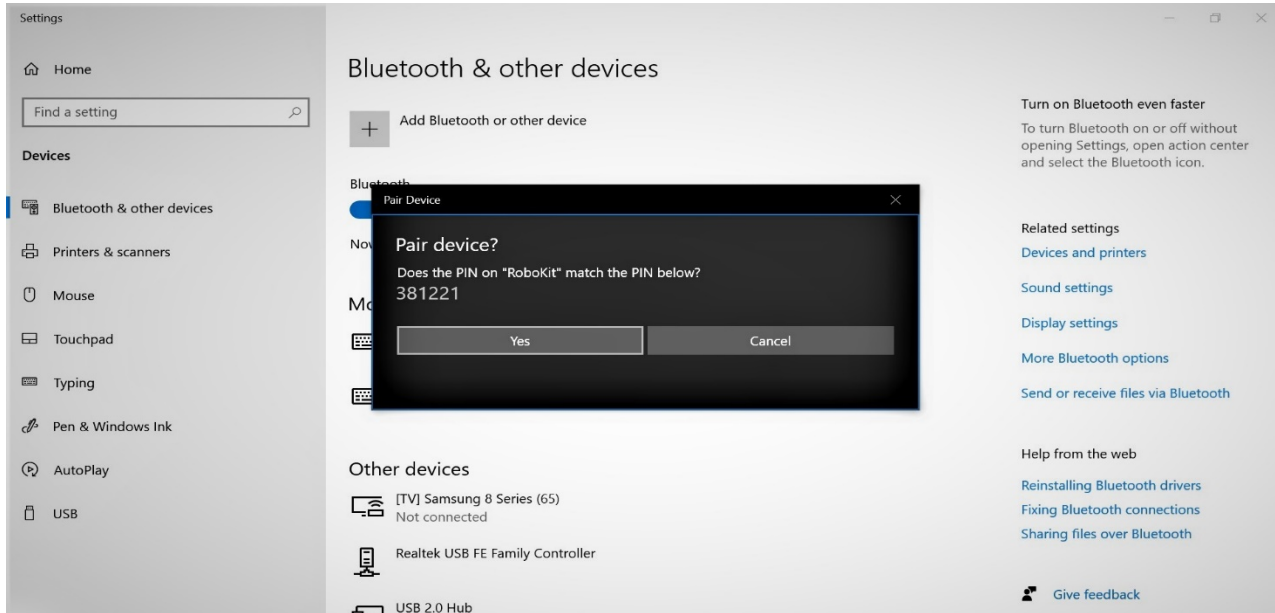
After clicking Connect, wait a few seconds until the device parameters appear. If the **“Auto connect”** checkbox is checked, the selected device will connect automatically on the application restart. Refer to Figure 18.

### 6.6 CONNECTING BT FOR THE FIRST TIME

When connecting BT for the first time, the device should pair with the Windows system to begin the communication. Press **“Connect”** button as shown in Figure 23 above. A toast message will pop up as shown below. **Note:** PIN is randomly generated so it will vary each time a pairing is established.

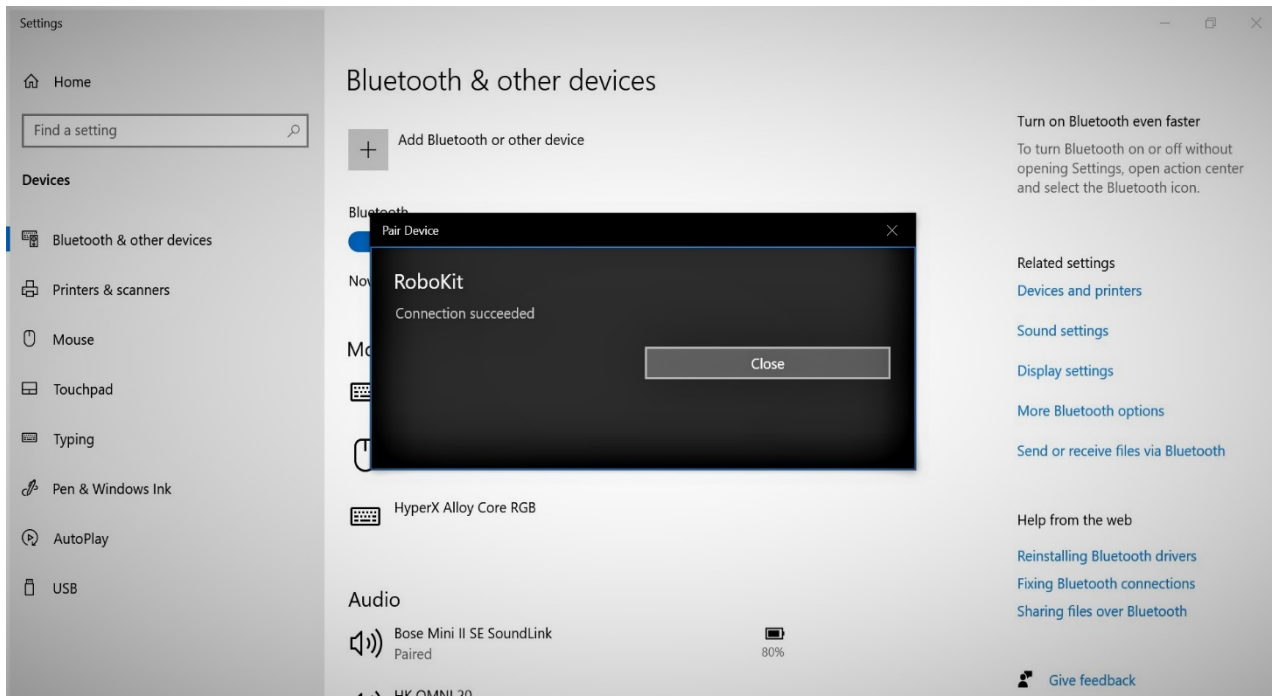


**Figure 24. BT toast screen**



**Figure 25. BT pairing accept screen**

Upon clicking “Yes” button following screen is launched indicating successful pairing between the host and server device. Click on “Close” and switch to the application screen.

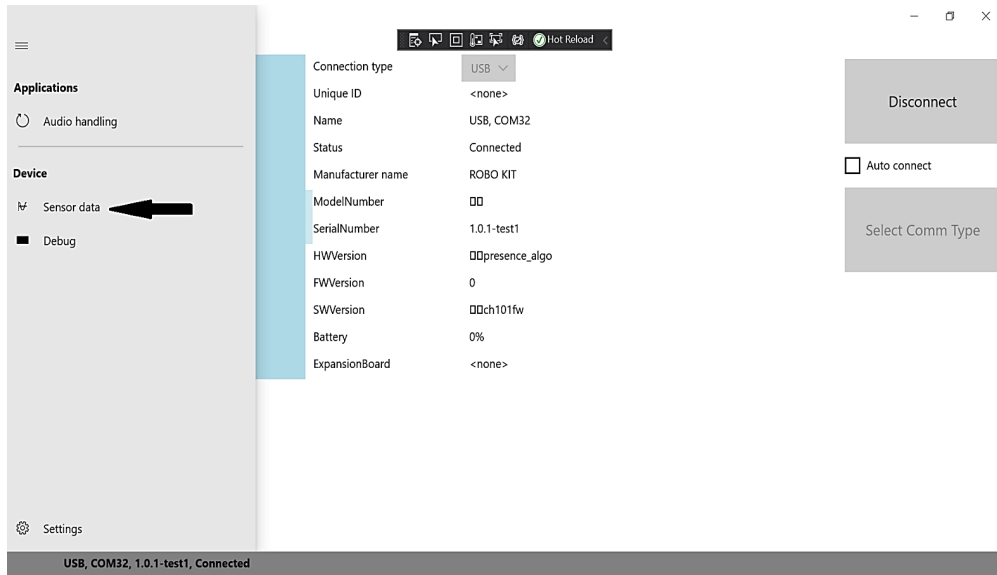


**Figure 26. BT close screen**

## 7 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.



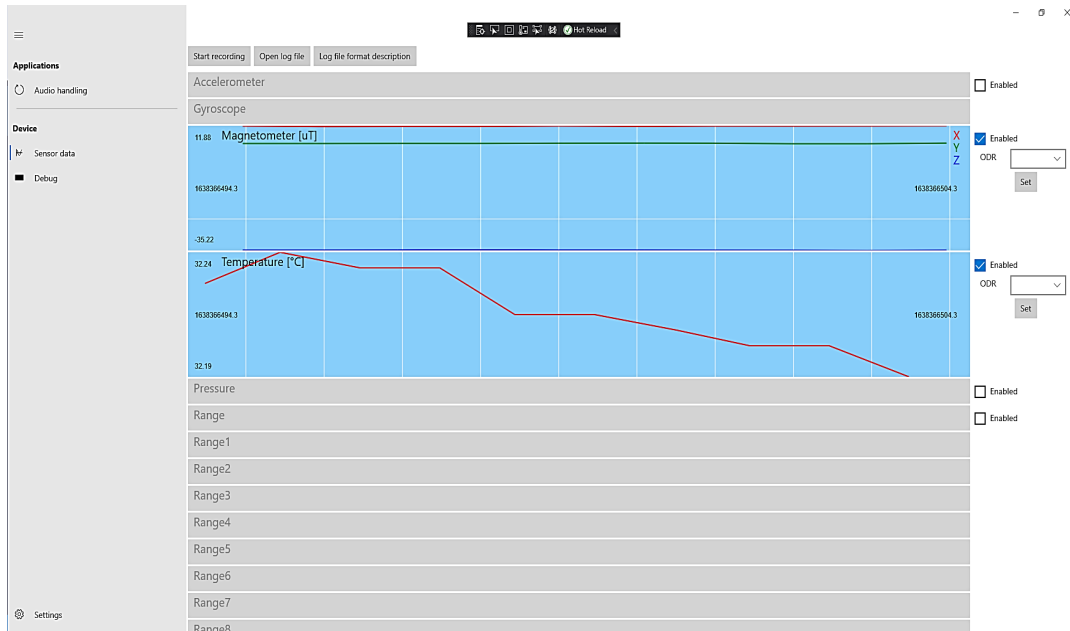
**Figure 27. Connect Device**



**Figure 28. Sensor Selection**

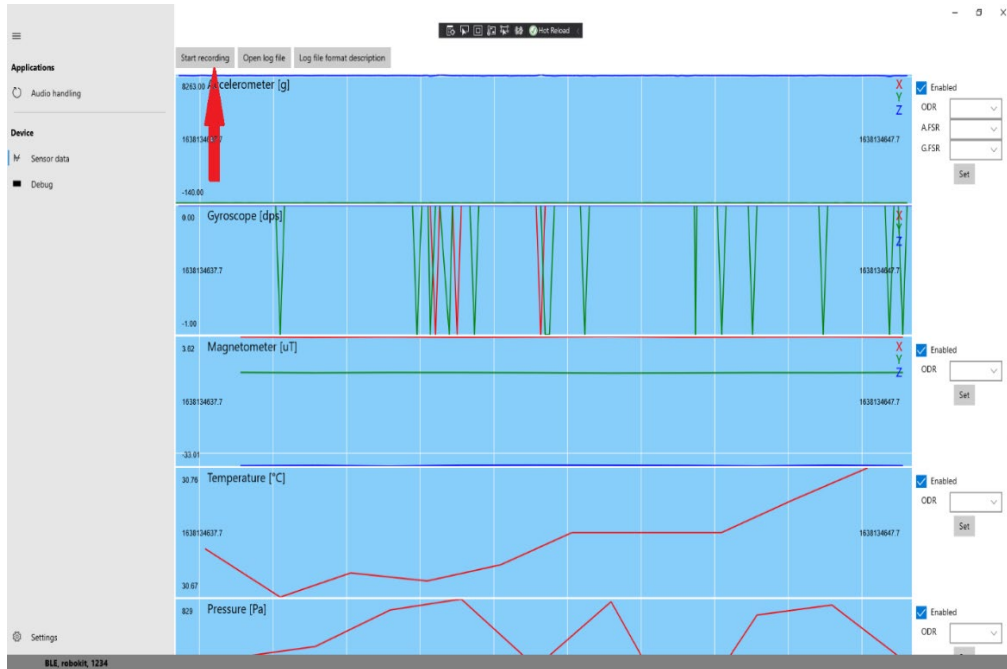


- Select the sensor(s) of interest by checking the **Enabled** checkboxes to begin the plotting.



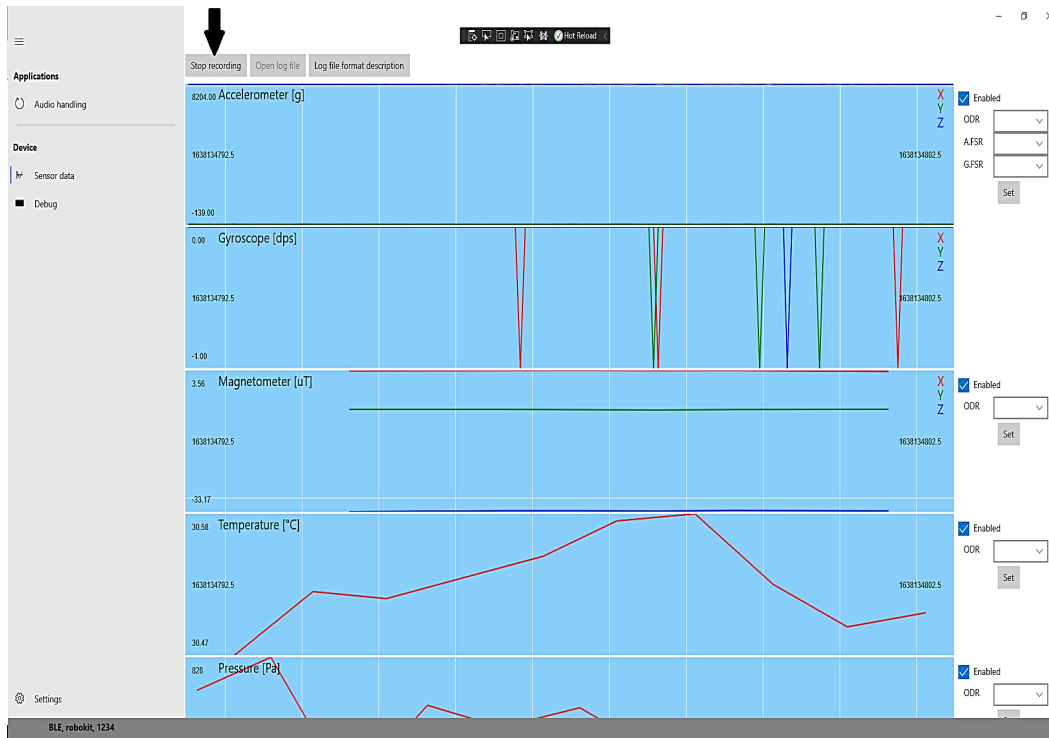
**Figure 29. Sensor Data Plot**

Begin data recording by clicking on the “Start recording” button. The app starts recording data for all sensors selected.



**Figure 30. Sensor Data Recording**

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

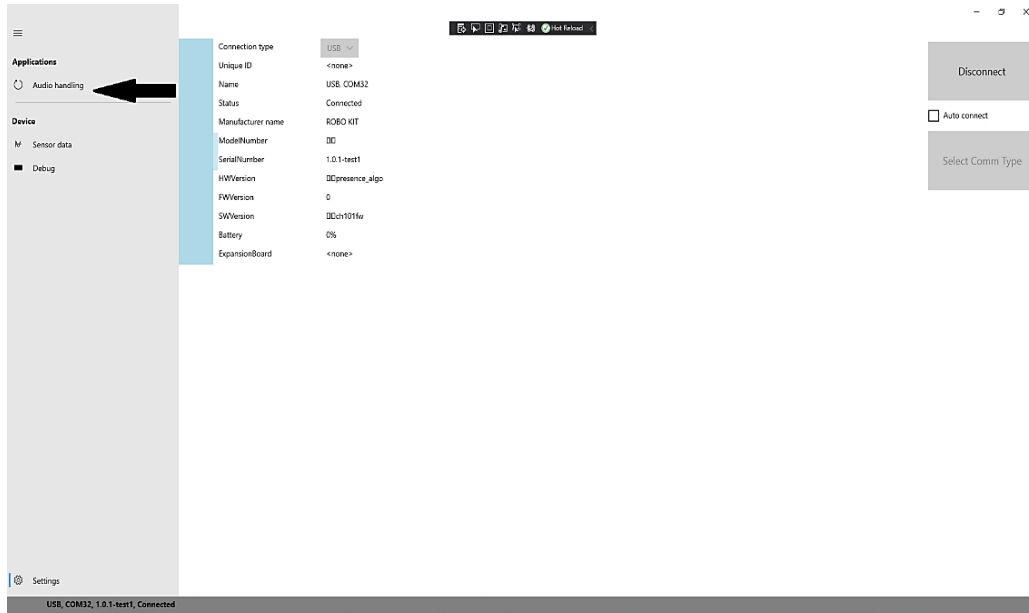


**Figure 31. Stop Data Recording**

## 8 AUDIO DATA CAPTURE

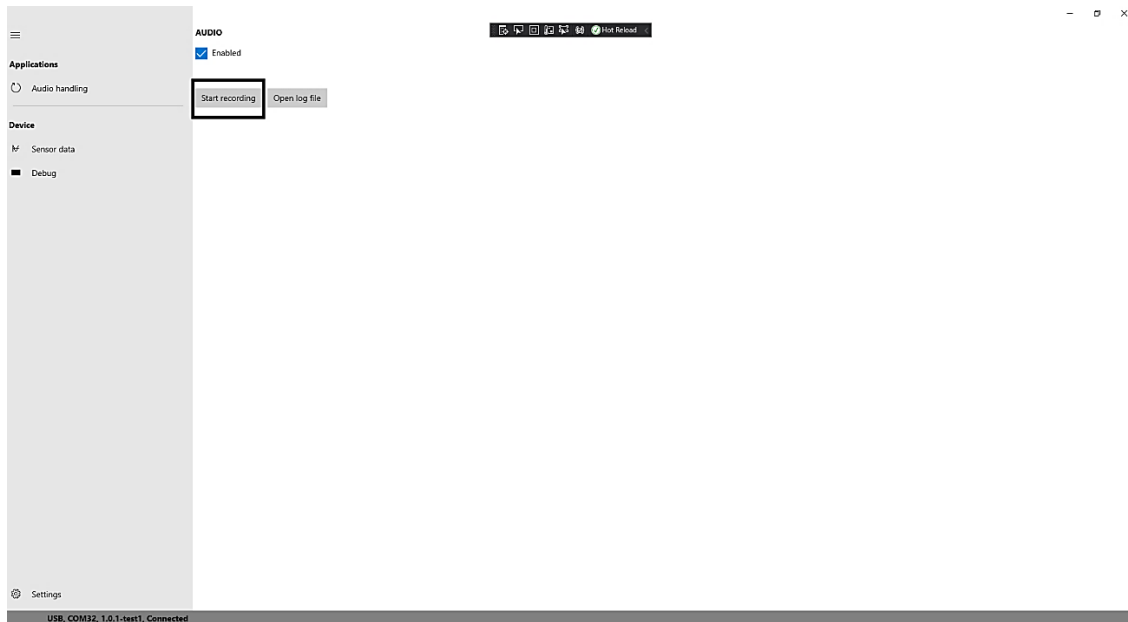
The app also facilitates the capturing of audio data streamed by the RoboKit1 board. For audio data capture and validation follow the steps below:

1. Click on “Audio Handling”



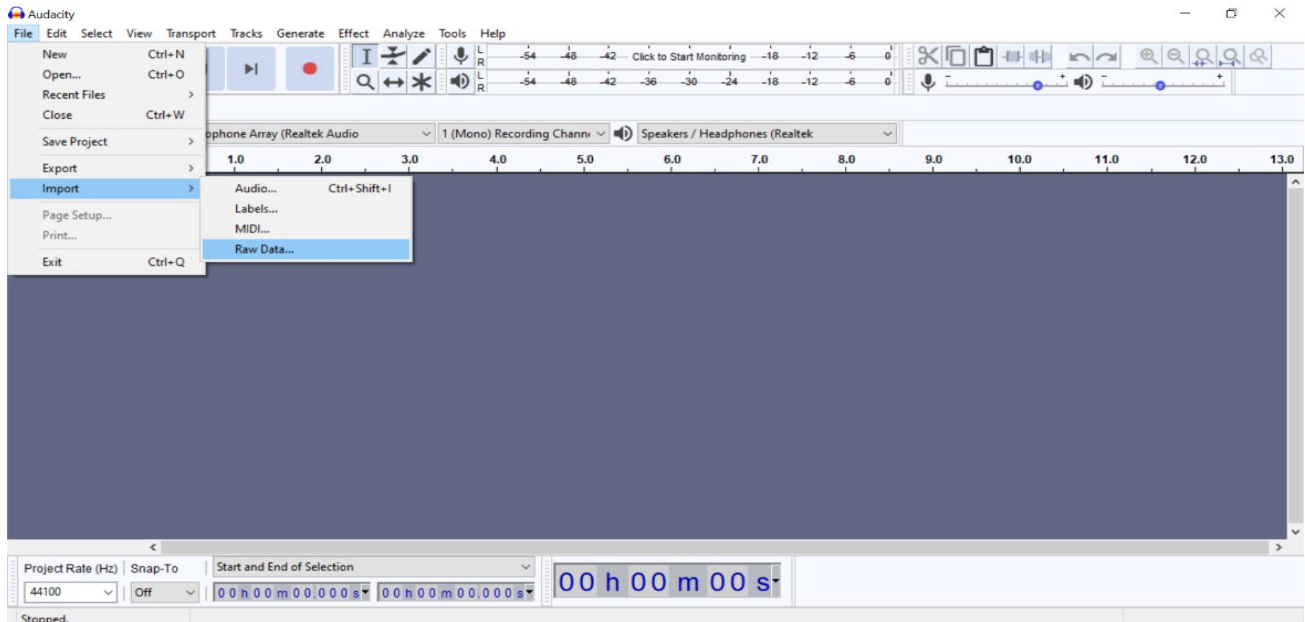
**Figure 32. Audio Data Handling**

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.



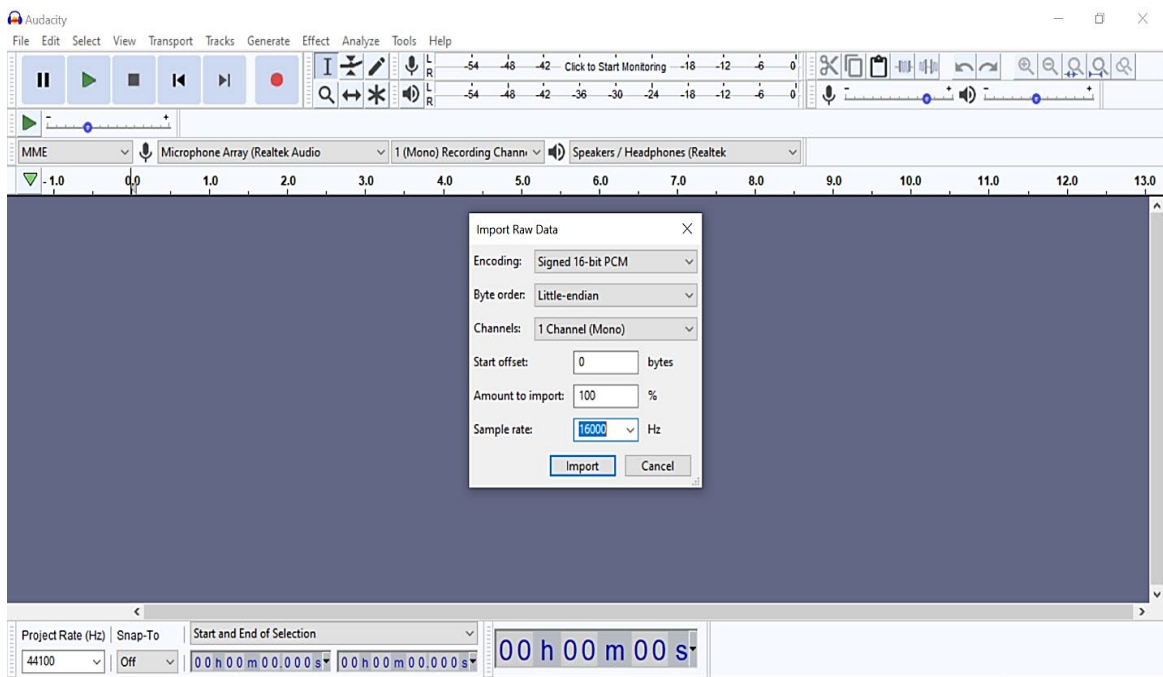
**Figure 33. Audio Data Recording**

3. While the app facilitates the audio data capture, use an audio specialty tool such as 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 34.



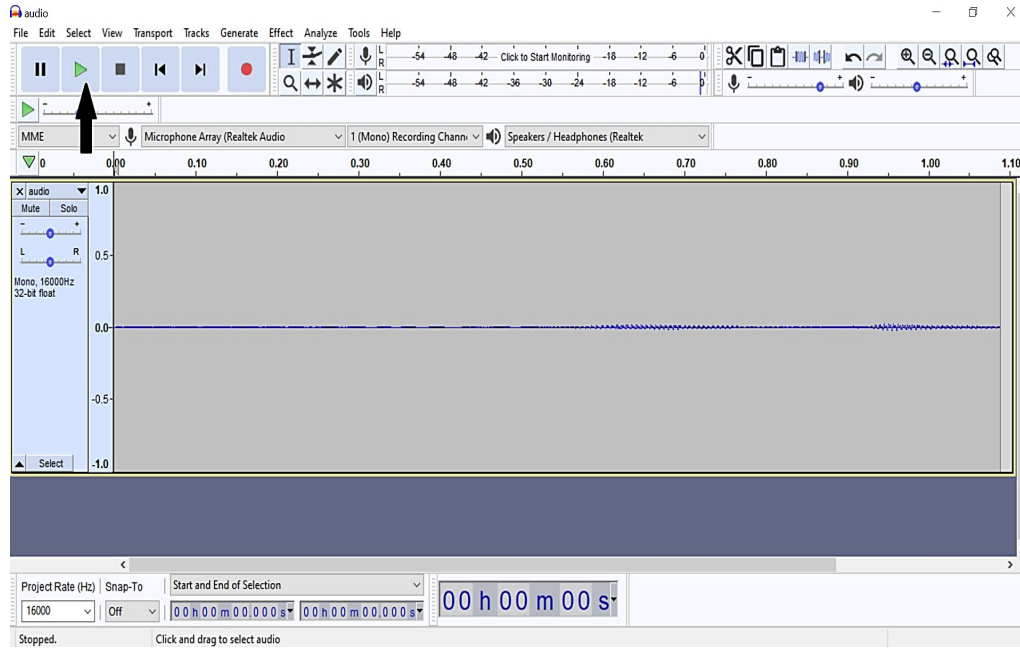
**Figure 34. Import Audio Data to Audacity**

- b. Click on **“Raw Data”** to launch a dialog to select audio file. Then choose the configuration as shown in Figure 35 and click **“Import.”**



**Figure 35. Audacity Audio Settings**

- c. After importing the audio, click the **PLAY** button as indicated in Figure 36, and the recorded audio data will play.



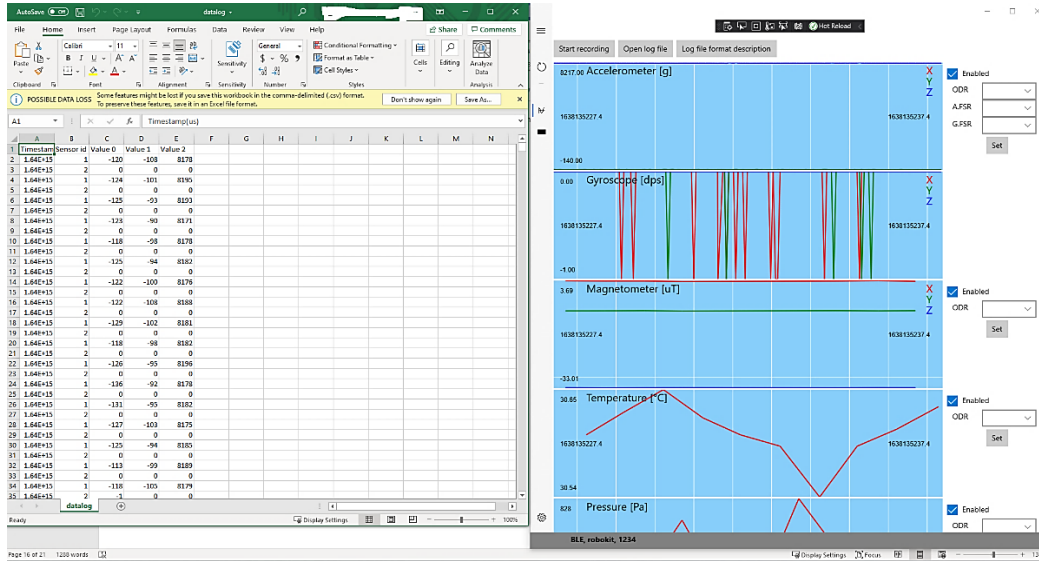
**Figure 36. Audio Data Playback in Audacity**

## 9 LOGGING

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

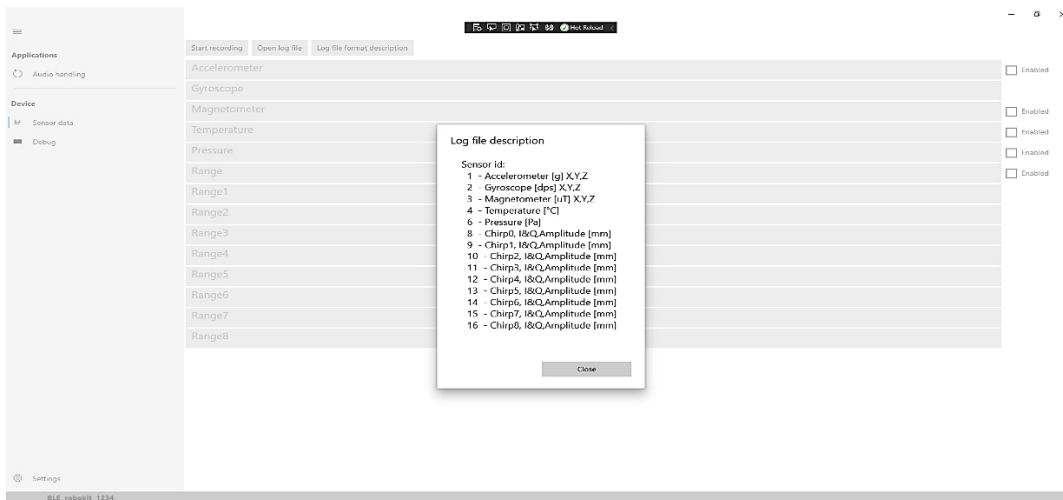
Logs will be in the path

C:\Users\XXXXXX\AppData\Local\Packages\InvenSenseInc.SmartBug\_cg71h24es2qyy\LocalState



**Figure 37. Sensor Data Logging**

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



**Figure 38. Sensor Data Log File Format**

## 10 USE DEBUG MENU TO SEE LOG MESSAGES

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.



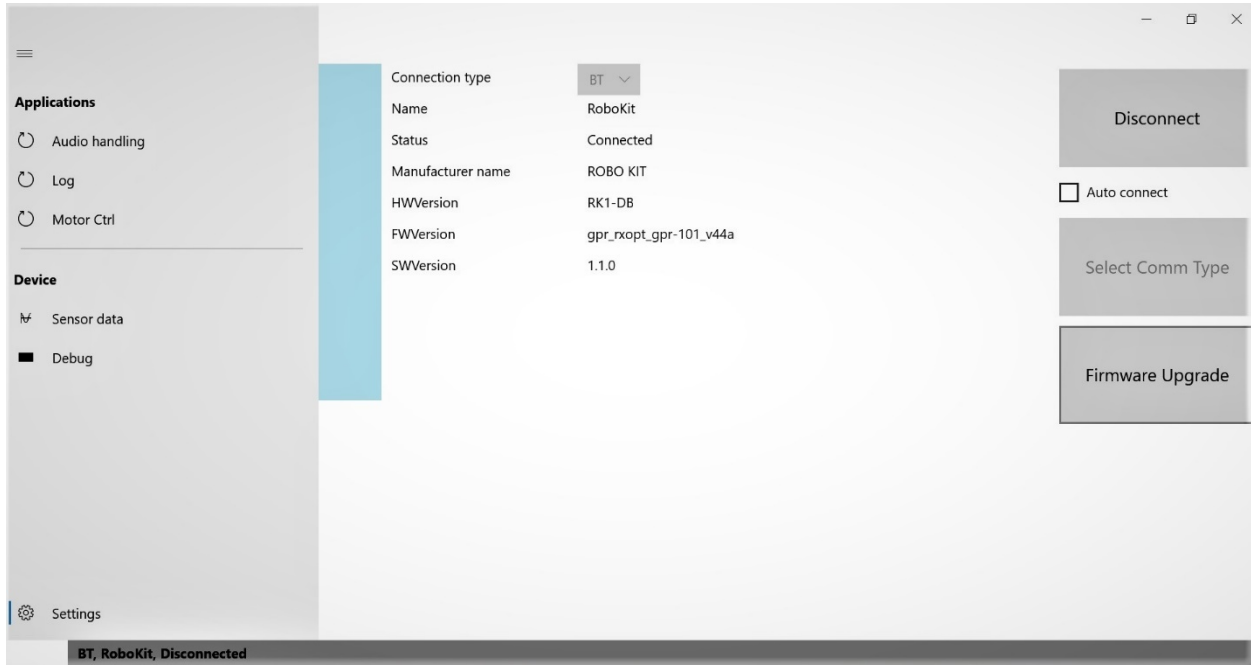
Figure 39. Debug screen



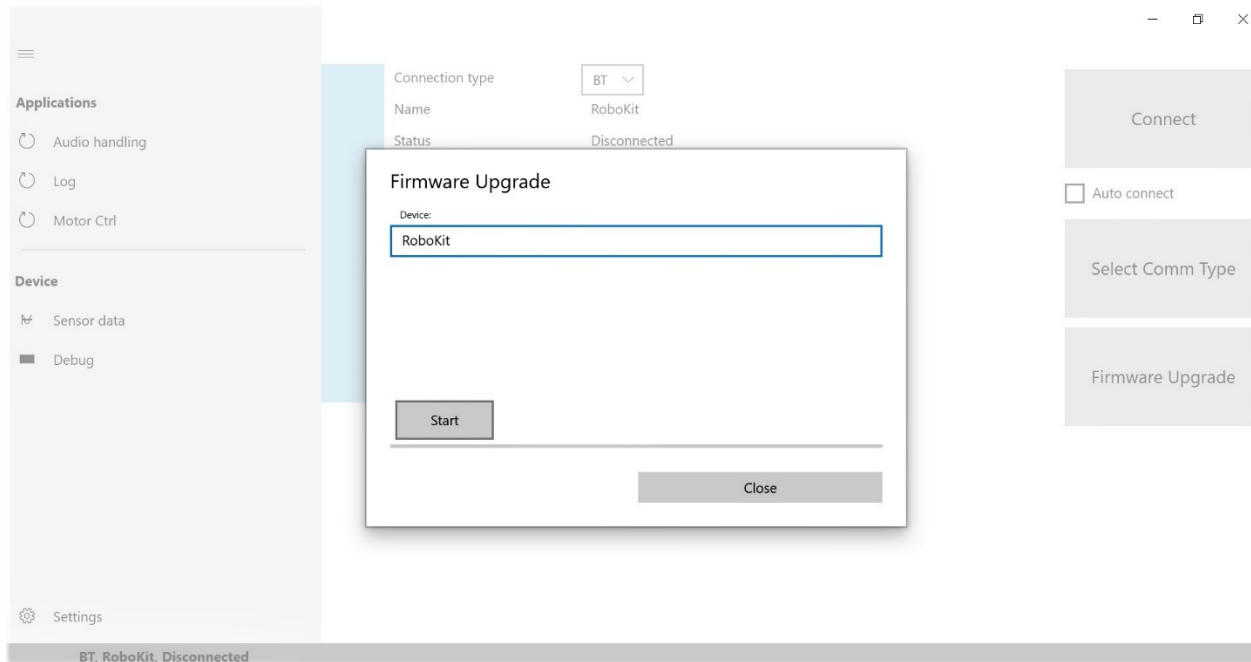
## 11 DEVICE FIRMWARE UPGRADE

Application provides a facility for the end user to upgrade the Robokit firmware.

Click on the “Firmware Upgrade” button.



**Figure 40. Firmware Upgrade**



**Figure 41. Firmware Upgrade start screen**

## 12 MOTOR CONTROL

To control the motor behavior via app click on the “Motor Ctrl” button on the extreme left of the UI resulting in launching of below screen. Options are self-explanatory. Choose the appropriate command to monitor the Motor behavior.

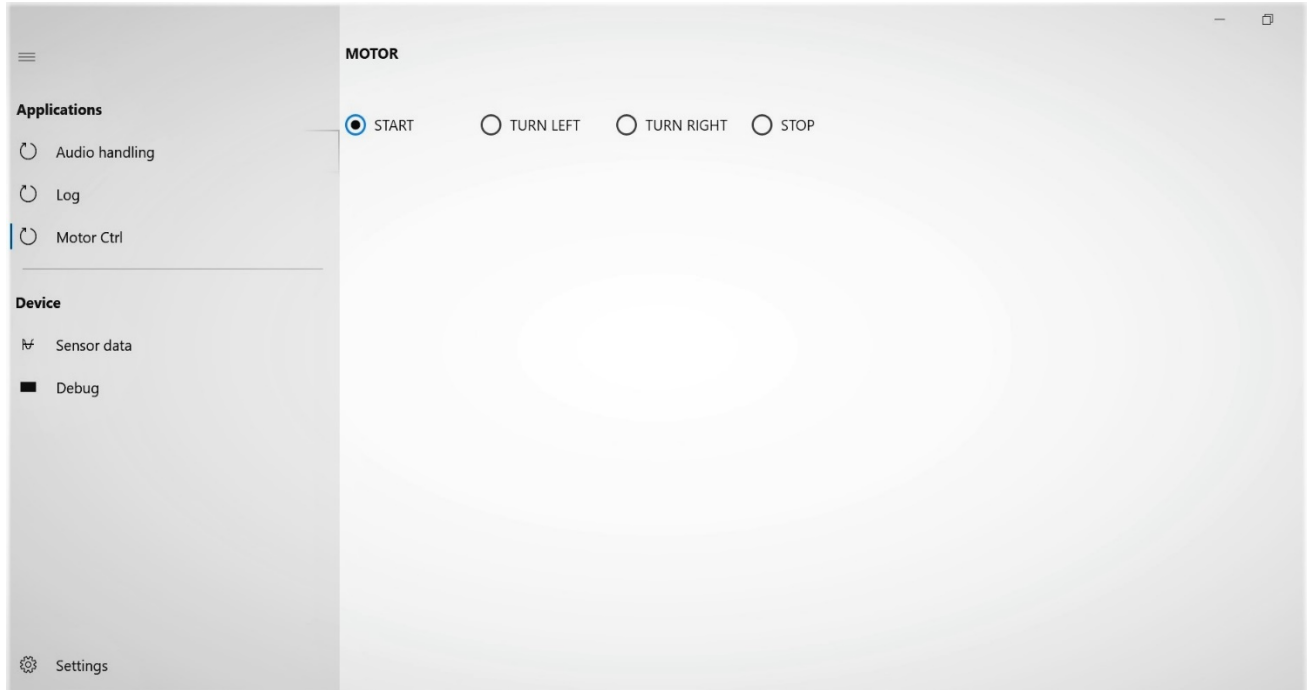


Figure 42. Motor control

### 13 REVISION HISTORY

REVISION DATE	REVISION	DESCRIPTION
01/16/2022	1.0	Initial release
02/28/2022	1.1	Added section 5
08/24/2022	1.2	Added Bluetooth section, Firmware upgrade, and Motor control.

This information furnished by InvenSense or its affiliates (“TDK InvenSense”) is believed to be accurate and reliable. However, no responsibility is assumed by TDK InvenSense for its use, or for any infringements of patents or other rights of third parties that may result from its use. Specifications are subject to change without notice. TDK InvenSense reserves the right to make changes to this product, including its circuits and software, in order to improve its design and/or performance, without prior notice. TDK InvenSense makes no warranties, neither expressed nor implied, regarding the information and specifications contained in this document. TDK InvenSense assumes no responsibility for any claims or damages arising from information contained in this document, or from the use of products and services detailed therein. This includes, but is not limited to, claims or damages based on the infringement of patents, copyrights, mask work and/or other intellectual property rights.

Certain intellectual property owned by InvenSense and described in this document is patent protected. No license is granted by implication or otherwise under any patent or patent rights of InvenSense. This publication supersedes and replaces all information previously supplied. Trademarks that are registered trademarks are the property of their respective companies. TDK InvenSense sensors should not be used or sold in the development, storage, production or utilization of any conventional or mass-destructive weapons or for any other weapons or life threatening applications, as well as in any other life critical applications such as medical equipment, transportation, aerospace and nuclear instruments, undersea equipment, power plant equipment, disaster prevention and crime prevention equipment.

©2022 InvenSense. All rights reserved. InvenSense, MotionTracking, MotionProcessing, MotionProcessor, MotionFusion, MotionApps, DMP, AAR, and the InvenSense logo are trademarks of InvenSense, Inc. The TDK logo is a trademark of TDK Corporation. Other company and product names may be trademarks of the respective companies with which they are associated.



©2022 InvenSense. All rights reserved.