



# Application Reference Code – Q2'21 FAE Training BLE Pulse Oximeter Lab

---

This hands-on exercise will demonstrate how to import and build the BLE HRM/SpO2 example project in Simplicity Studio v5, and go over the software architecture. It will also demonstrate how to run and test the demo, including the basic operation of taking HRM/SpO2 measurements, USB debug mode, BLE connection to a mobile app and the expected result of the demo.

## KEY FEATURES

---

- Import example project in Simplicity Studio V5
- Build and Flash the application to device
- Project code walkthrough
- Take HRM/SpO2 measurements on the fingertip and view results in the EFR Connect Mobile application

## 1 Introduction

In this exercise, we will import the “Bluetooth Module Explorer Kit HRM/SpO2 Software Demo using MAXM86161 sensor” project that’s posted on the GitHub ([https://github.com/SiliconLabs/bluetooth\\_applications/tree/master/bluetooth\\_explorer\\_kit\\_i2c\\_bio\\_sensor](https://github.com/SiliconLabs/bluetooth_applications/tree/master/bluetooth_explorer_kit_i2c_bio_sensor)).

The example implements the BLE heart rate and pulse oximeter services. It enables a peer device to connect to BMG220P module and receive heart rate and pulse oximeter values via Bluetooth. The reported values are measured by a biometric sensor and calculated using an HRM/SpO2 algorithm.

### 1.1 Hardware Requirements

- [BGM220 Bluetooth Module Explorer Kit \(BGM220-EK4314A\)](#)
- [Heart Rate 2 Click Board from MikroE](#)

### 1.2 Software Requirements

- Simplicity Studio v5
  - BLE SDK v3.1.2
- EFR Mobile APP

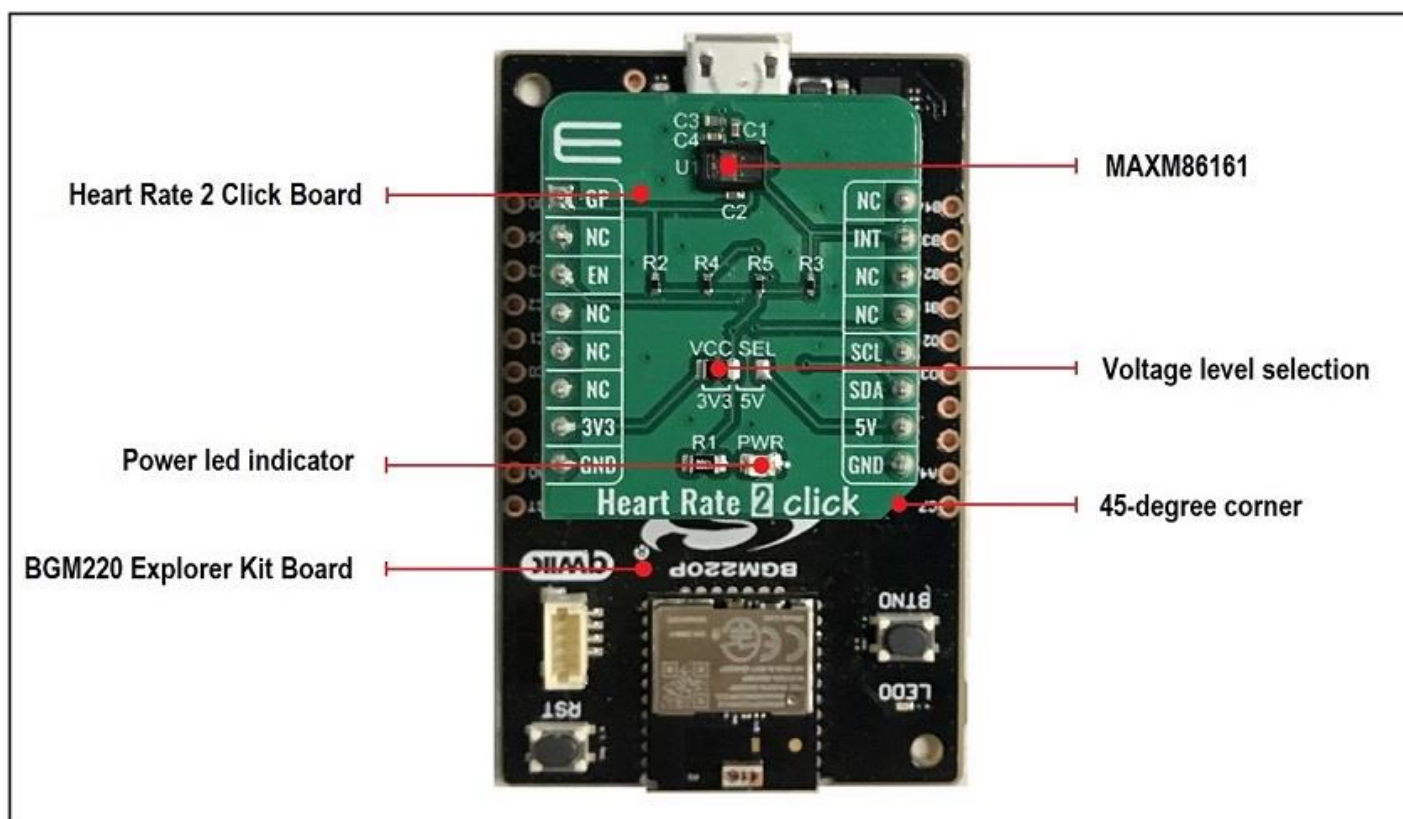


Figure 1: Board

### 1.3 Connection

Connect the Heart Rate 2 Click board to the BLE Explorer Kit directly through the socket. Make sure that the click board's 45-degree corner matches the Explorer Kit's 45-degree white line. See Figure 1. Afterwards, power up the board by connecting the kit to the PC using a microUSB cable. The Power LED on Heart Rate 2 Click board will light up when the kit is powered.

## 2 Import and Build the Example Project

The project file (bluetooth\_explorer\_kit\_i2c\_bio\_sensor.sls) can be downloaded directly from the GitHub Repo: [https://github.com/SiliconLabs/bluetooth\\_applications/tree/master/bluetooth\\_explorer\\_kit\\_i2c\\_bio\\_sensor/SimplicityStudio](https://github.com/SiliconLabs/bluetooth_applications/tree/master/bluetooth_explorer_kit_i2c_bio_sensor/SimplicityStudio).

### 2.1 Import the \*.sls project file

In Simplicity Studio v5, select File->Import... Then click "Browse" and navigate to the folder where the project file is saved. See Figure 2. Click "Next" and verify that the SDK version and the board/part information is correct. See Figure 3. Finally, click "Finish" to import the project.

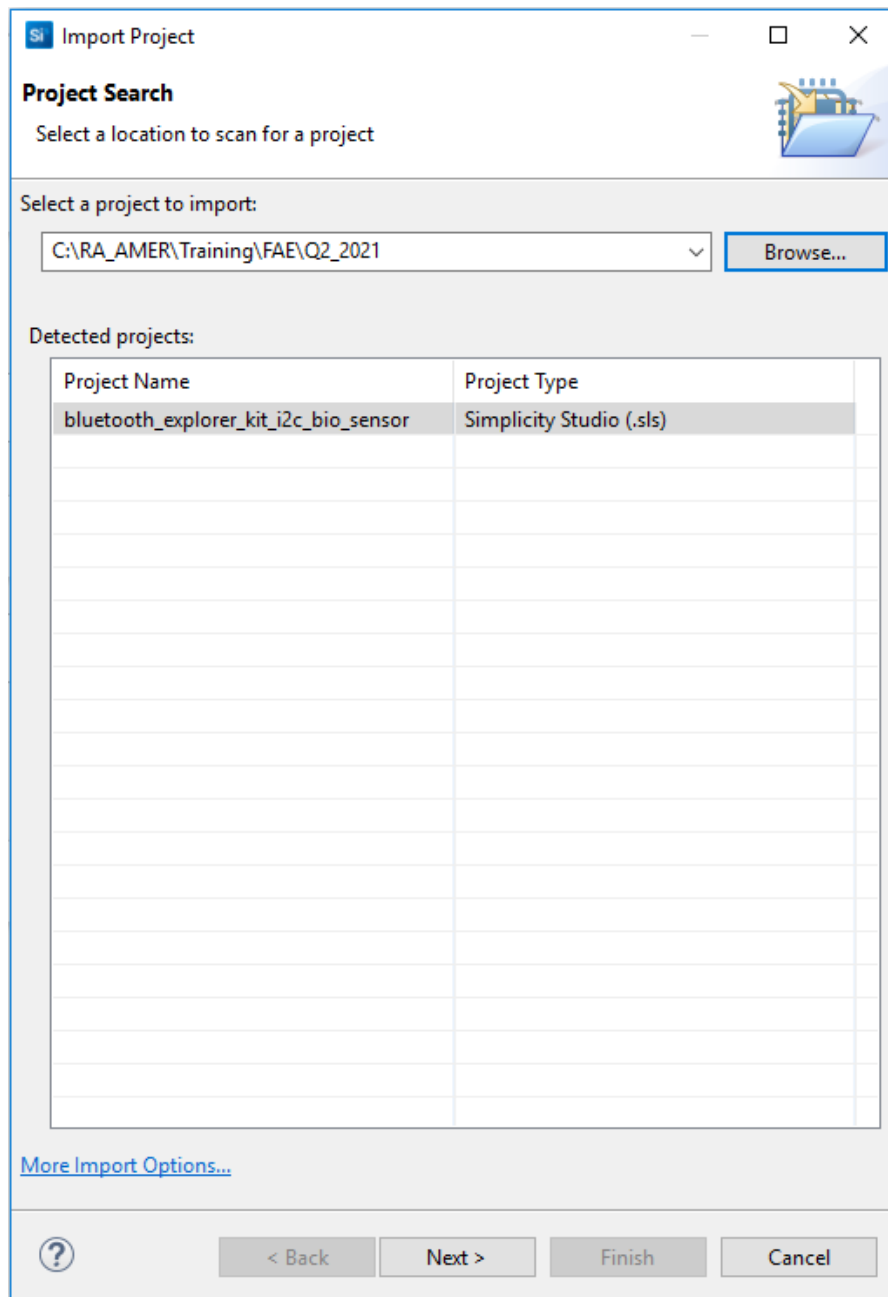


Figure 2: Select Project

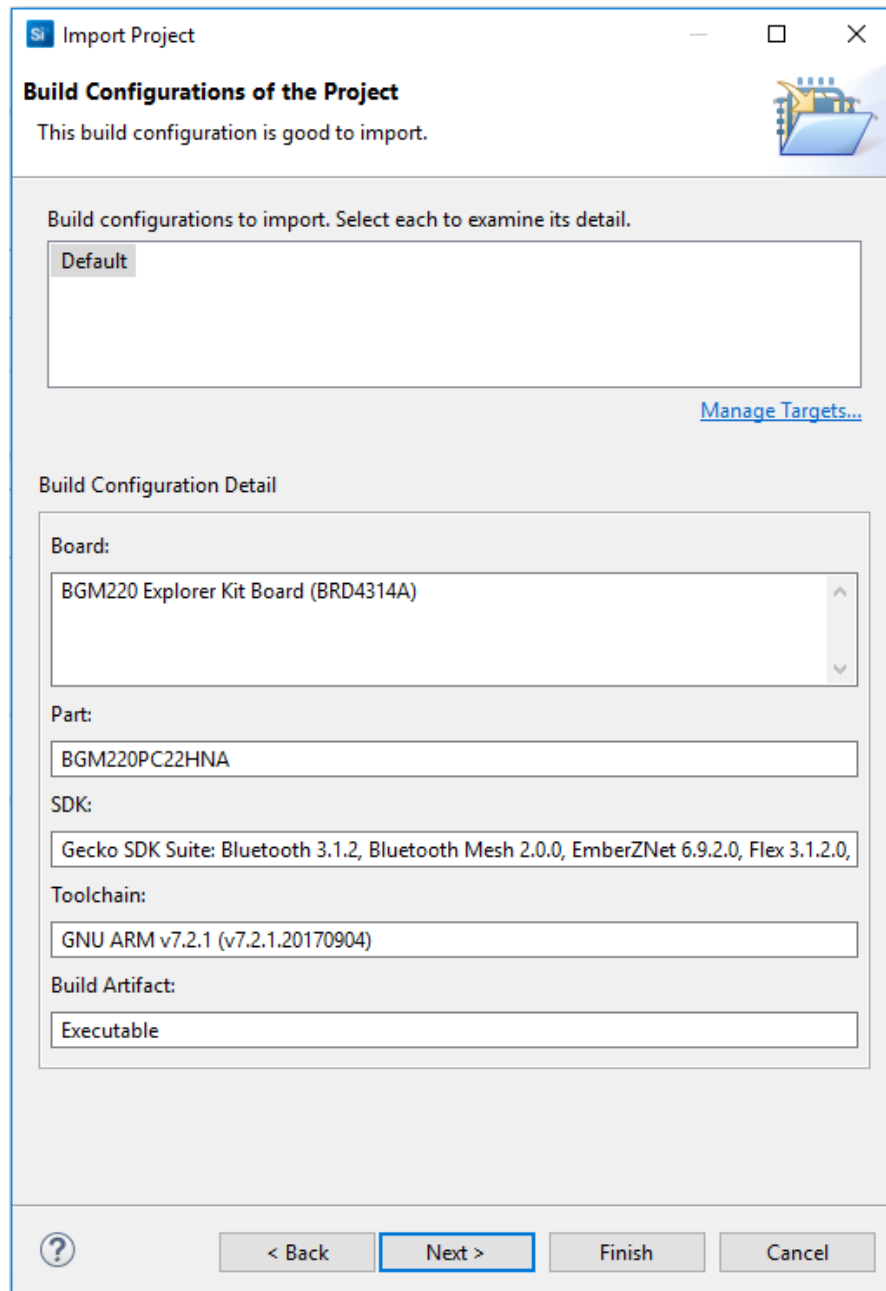



Figure 3: Build Configurations

## 2.2 Build the project

After the project is successfully imported, the Studio should automatically switch to the Simplicity IDE perspective. You can then click and open any source file under the project. Next, click the  button on the Toolbar to build the project. The project should build without any error. See Figure 4. If there's any error showing up in the console, please double check your BLE SDK version and make sure that the latest is installed.

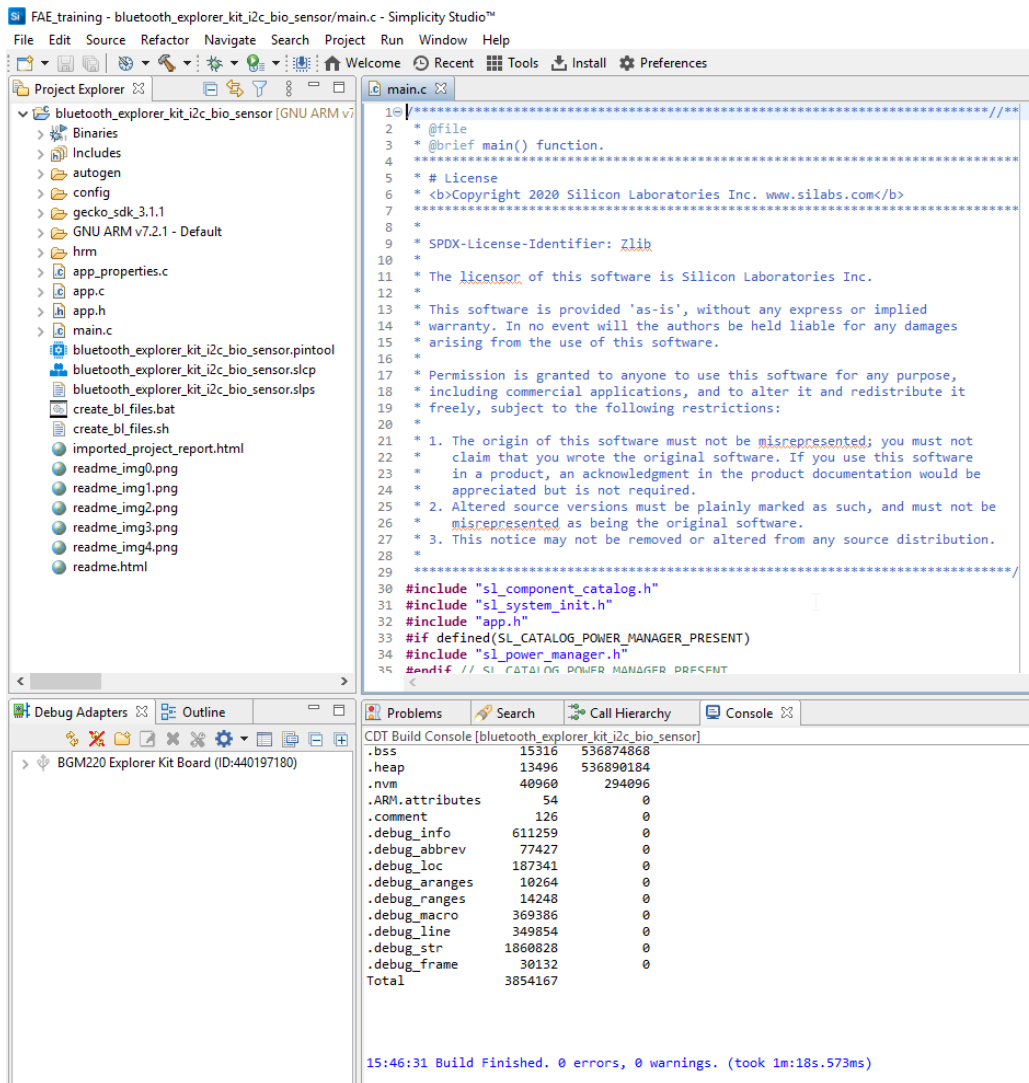


Figure 4: Build the Project

### 3 Code Walkthrough

In this section, we will briefly explain the software architecture of the project. If the customer is developing a similar application, they can follow the same architecture to develop their software. We will then break down each key component in detail and walk you through the code under the project folder.

#### 3.1 Software Architecture

Figure 5 shows an overview of the HRM/SpO2 demo software architecture. The microcontroller communicates with the biometric sensor MAXM86161 through the I2C interface. The sensor driver is implemented to configure the sensor and get the sensor data. Then, the algorithm can call the sensor driver API to initialize the sensor to take measurements and calculate HRM/SpO2 values based on measurement results. The HRM/SpO2 values will be reported to the application layer, which implements the BLE heart rate and pulse oximeter services so that connected Bluetooth devices can receive those values.

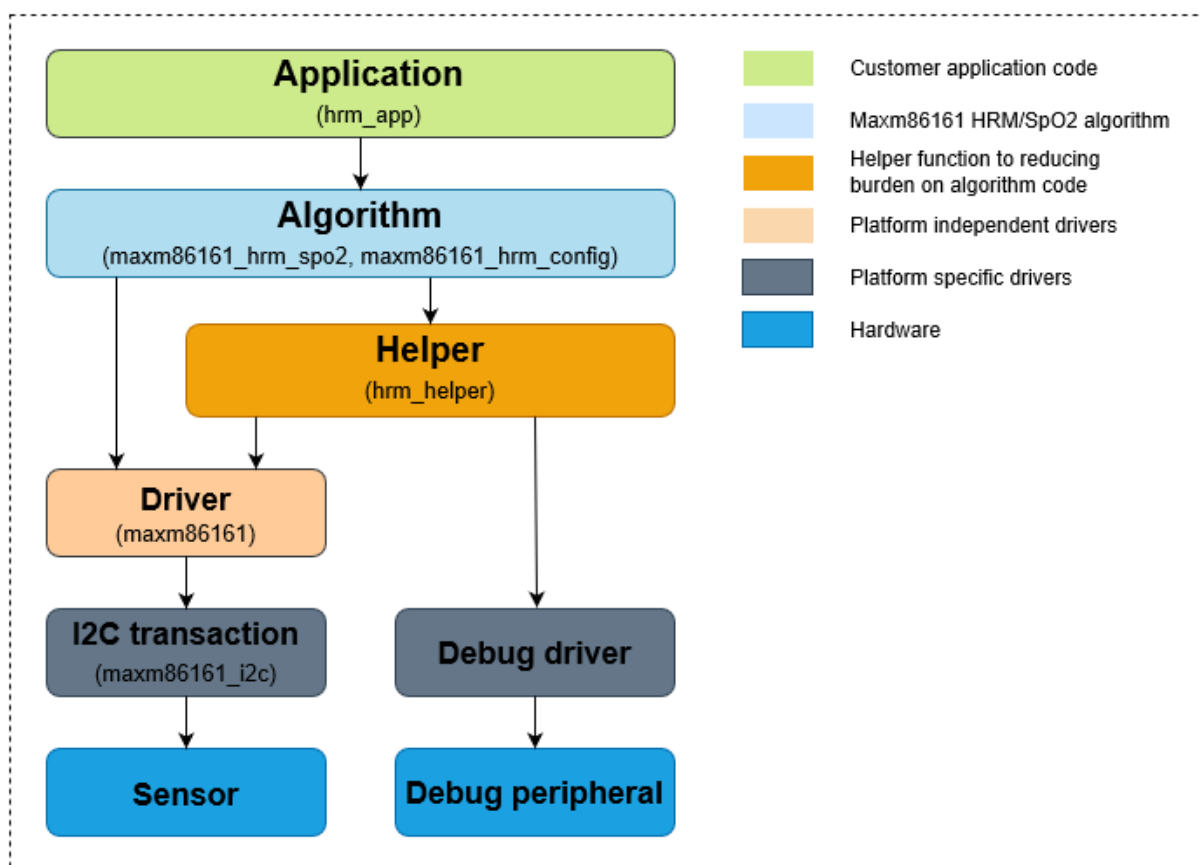


Figure 5: Software Architecture

#### 3.2 MAXM86161 Sensor Driver

The sensor driver code can be found in the /hrm/drivers folder under the project directory. See Figure 6. The code consists of two parts, the I2C driver (maxm86161\_i2c.h/c) and the sensor driver (maxm86161.h/c). The I2C driver is developed to communicate with the sensor through the I2C interface. The sensor driver code will then utilize the I2C driver to implement API functions that configures the sensor. It also buffers the sample read from the sensor in a FIFO for post processing. Both the I2C driver and sensor driver code should be reusable across all Silicon Labs Wireless SoCs.

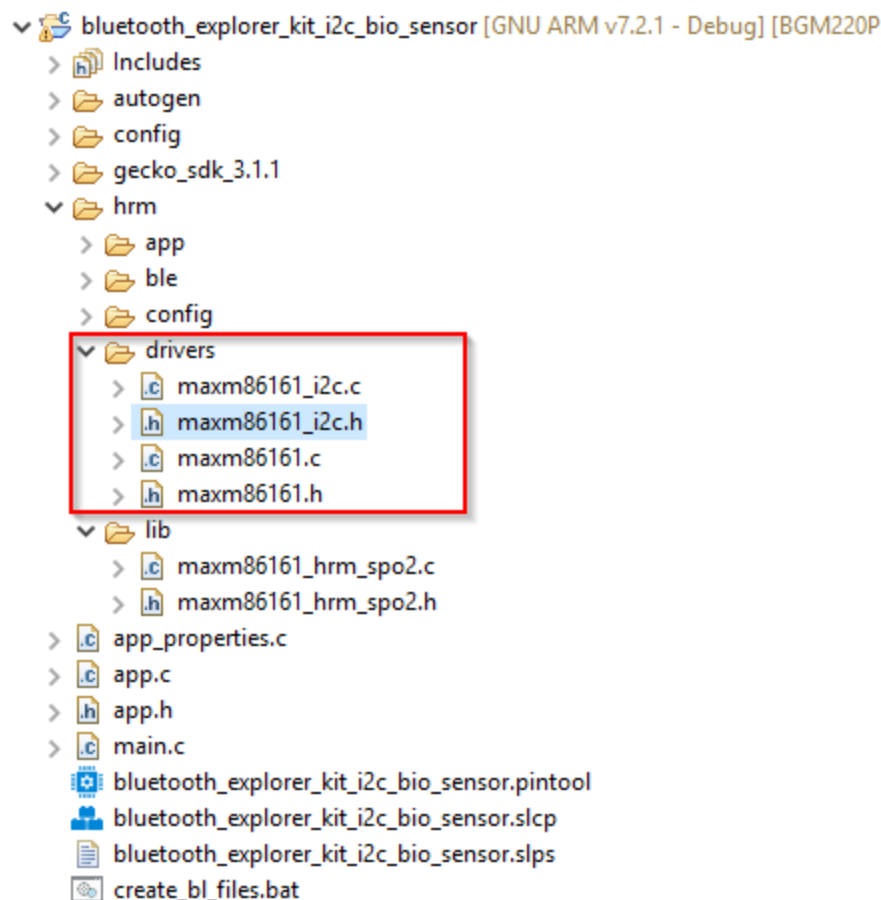


Figure 6: MAXM86161 Driver

### 3.3 HRM/SpO2 Algorithm

The algorithm code sits in the /hrm/lib folder. See Figure 7. It configures the sensor to take measurements using three different LEDs, green, red and IR LEDs. The measurement result from the green LED is used to calculate the heart rate and the measurement result from the red and IR LEDs are used to compute the SpO2. Silicon Labs developed these algorithms mainly for demonstration purpose. The performance is only compared against a commercial Pulse Oximeter. We do NOT claim the accuracy of the algorithm, nor do we plan to support the development of algorithm in the future. The customer should take it as it is or use it as a reference to develop their own algorithm.

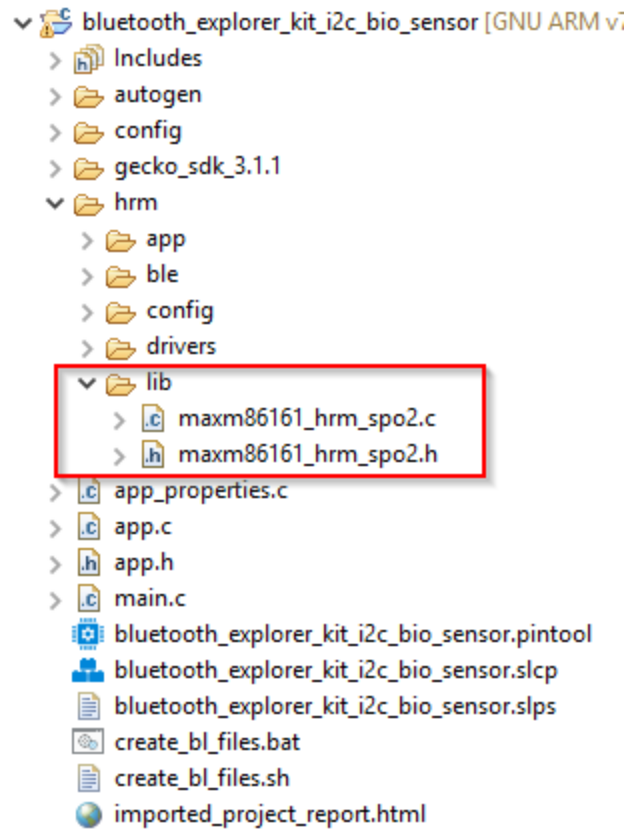


Figure 7: HRM&SpO2 Algorithm

### 3.4 BLE Application

At the application layer, we can break down the code into two parts. One implements the heart rate and pulse oximeter services. The other is the main code where the initialization of the sensor and the running of HRM/SpO2 algorithm take place.

#### 3.4.1 Heart rate and pulse oximeter services

These two services are implemented with the help of Bluetooth GATT Configurator tool under /config/btconf folder. When the BLE connection is built and the notification is enabled, it will start a 500ms timer. When the timer times out, the client will be notified about the latest heart rate and SpO2 values.

The detailed implementation can be found in the /hrm/ble/heart\_rate and /hrm/ble/pulse\_oximeter folders. See Figure 8.



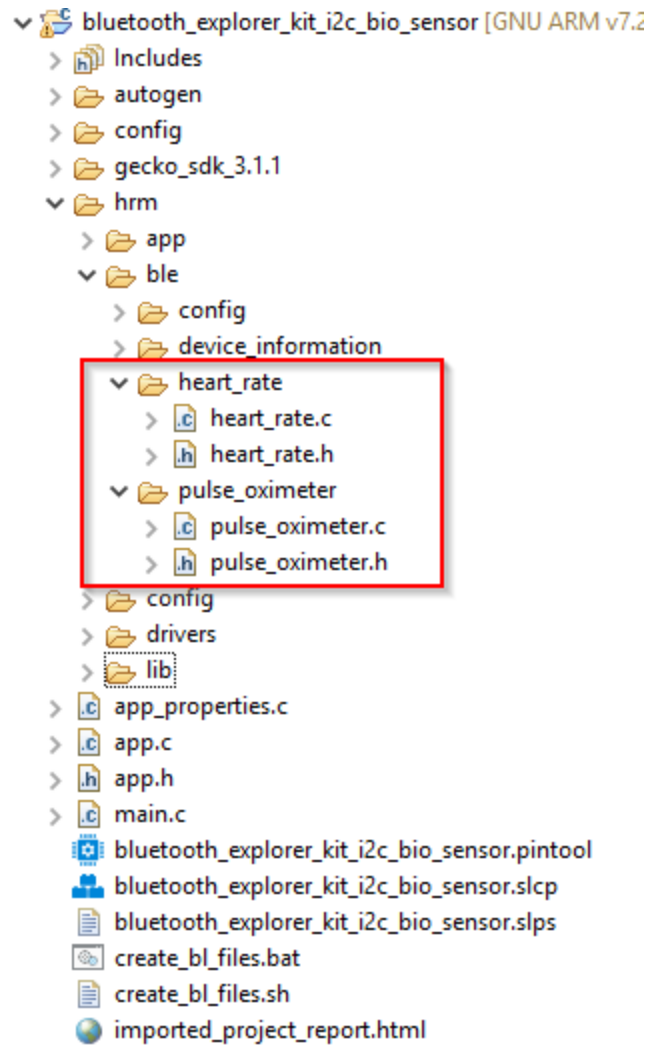


Figure 8: Heart Rate and Pulse Oximeter Services

### 3.4.2 Application and helper functions

The main logic of the application is implemented in `hrm_app.c` file. We also added another layer called `hrm_helper.c` code as a reference to show how to create sample buffers and add debugging utilities that customers may find useful. Both of those codes can be located in the `/hrm/app` folder. See Figure 9.

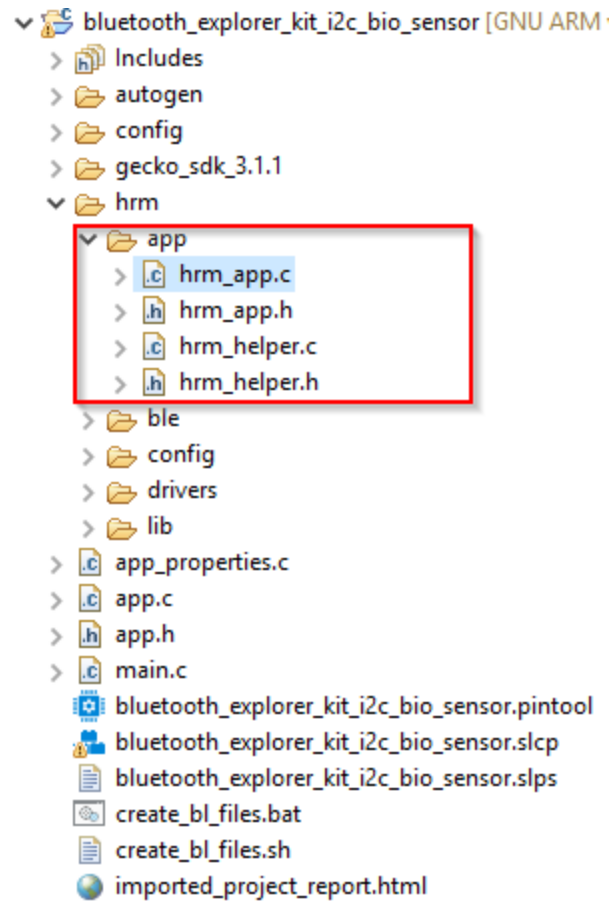


Figure 9: HRM Application Code

## 4 Flash the Demo

After developing a good understanding about the code structure of the project. We can now move on to flashing the demo to our device.

### 4.1 Flash the Bootloader

It is required to flash the bootloader first to run the application. The bootloader file is already provided on GitHub. But users can create their own bootloader in Simplicity Studio as well.

Right click at the bootloader hex file and select “Flash to Device...”. Choose the connected J-Link device which will open up the Flash Programmer as shown in Figure 10. Then, click “Program” to flash the bootloader.

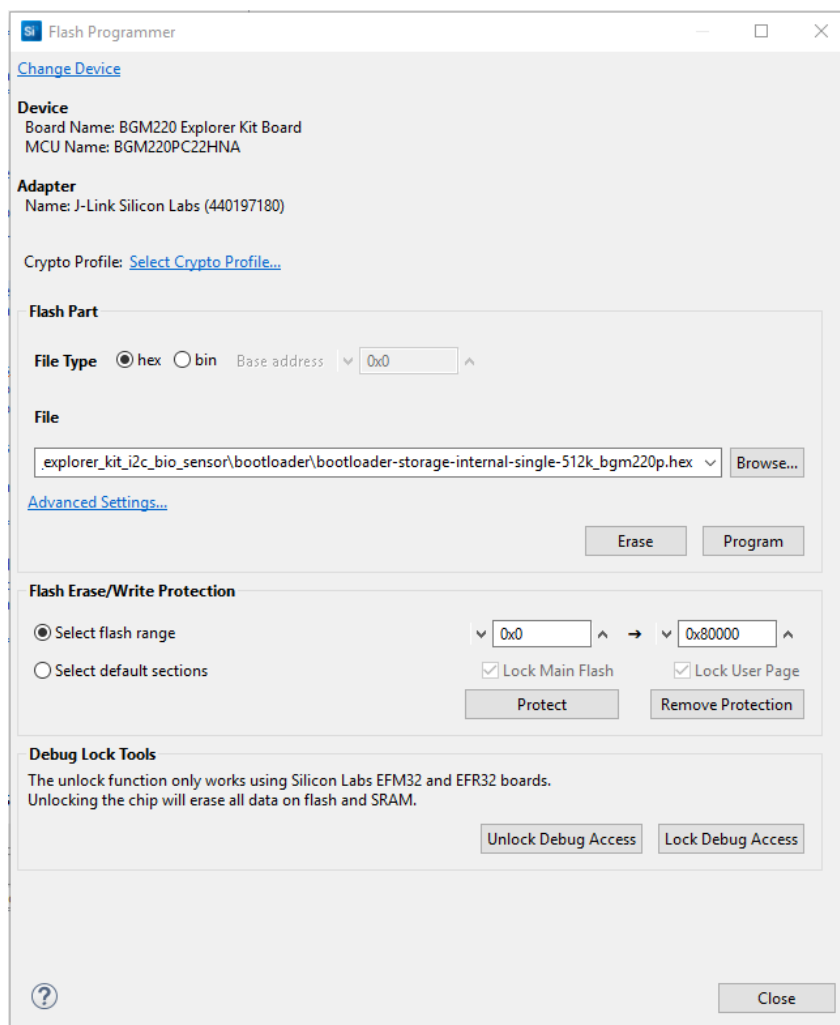


Figure 10: Flash Programmer

### 4.2 Flash the Application to the Device

Once the bootloader is flashed to the device, we can flash the application. If the project is built successful as described in section 2.2, we should have the bluetooth\_explorer\_kit\_i2c\_bio\_sensor.hex file of the application generated under /GNU ARM v7.2.1 – Default folder. Follow the same steps in section 4.1 to flash the hex file.

## 5 Run and Test the Demo

To run and test the demo, we need to use a smart phone with EFR Connect mobile application installed. Currently, the measured heart rate and SpO2 values can only be displayed on the mobile APP through Bluetooth connection. We also have developed a new project which integrates an OLED display that can show the value on the screen. Please check that out on the GitHub () if you're interested.

### 5.1 Connect to EFR Mobile Application

Open the EFR Connect application on your smartphone and allow the permission request when opened for the first time. Then click [Develop] -> [Browser] and you will see a list of nearby devices which are sending Bluetooth advertisements. Scroll down the list of devices and find the one named "Silabs HRM Example". Click the "Connect" button on the right side to connect to the device. See Figure 11. Wait for the connection to be established and GATT database to be loaded. Afterwards, you should see the Heart Rate service and the Pulse Oximeter service. Click "More Info" will expand the detailed information under those services.

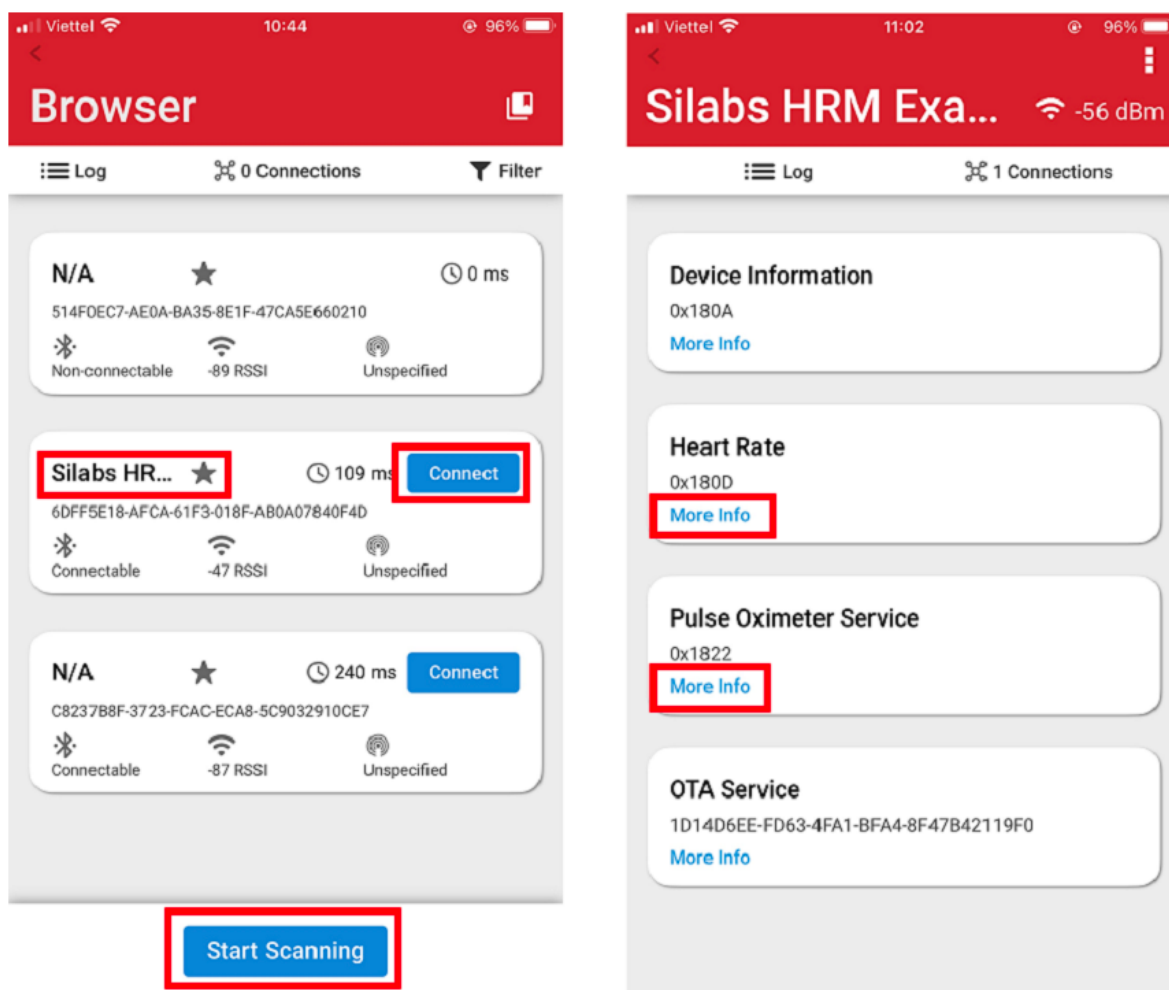


Figure 11: EFR Mobile App

### 5.2 Start HRM/SpO2 Measurements

BTN0 on the Explorer Kit is used to start/stop the HRM/SpO2 measurement. After connecting the device to the EFR Connect mobile application, press BTN0 button to start the measurement.

#### 5.2.1 Finger Detection

The algorithm has a built-in feature called finger detection. It will turn on one of the LED (Green/IR/Red) at a low sampling rate to detect the presence of the finger. If a valid finger contact is detected, it will turn on all 3 LEDs and start HRM/SpO2 measurements. If users remove their fingers during the measurement, the algorithm will switch back to finger detection mode.

In this demo, after pressing the BTN0, users should see the Green LED being turned on to detect the finger contact.

### 5.2.2 Take measurements

Users should put their finger on the sensor with a proper pressure (solid contact between the finger and the sensor without optical leakage but don't press with force) and stay idle for at least 10s. Once started, the measurement can be stopped at any time by pressing BTN0 button again.

### 5.3 View Test Results

Next, we can check the heart rate value and the status of finger (ON/OFF) in the EFR Connect App by simply turning on "Notify" in the Heart Rate Measurement characteristic of the Heart Rate service. See Figure 12. To check both the heart rate and the SpO2 values, we need to turn on "Notify" in the PLX Continuous Measurement characteristic of the Pulse Oximeter service.

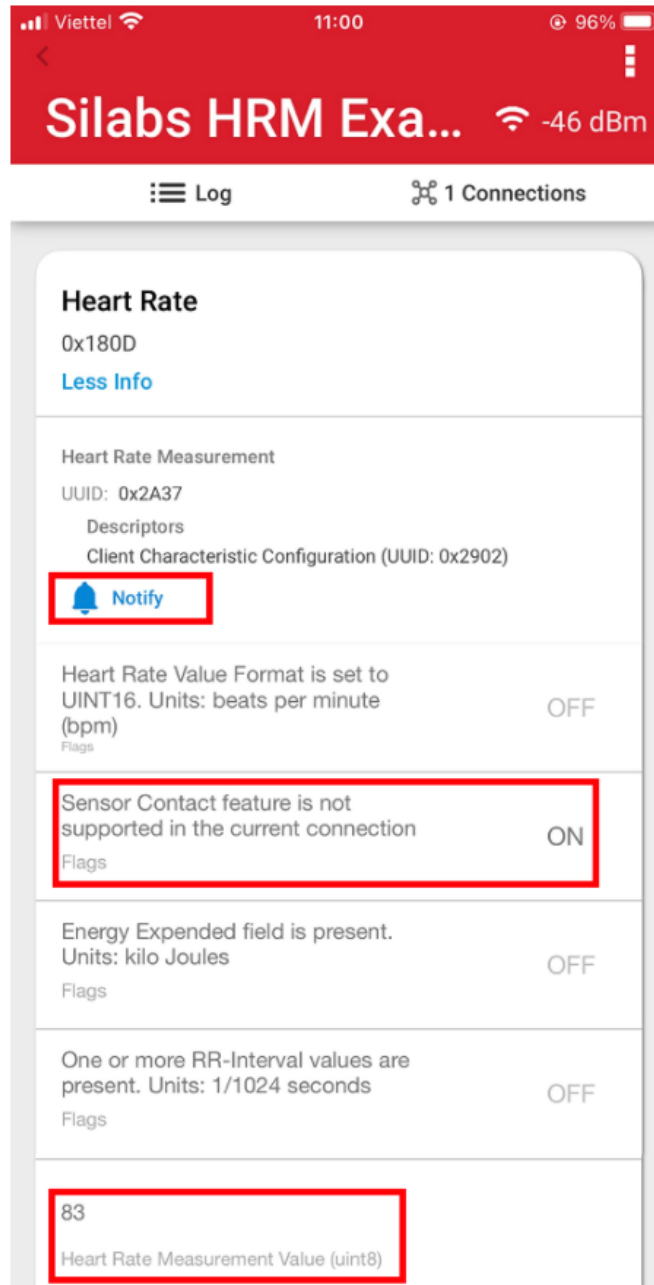


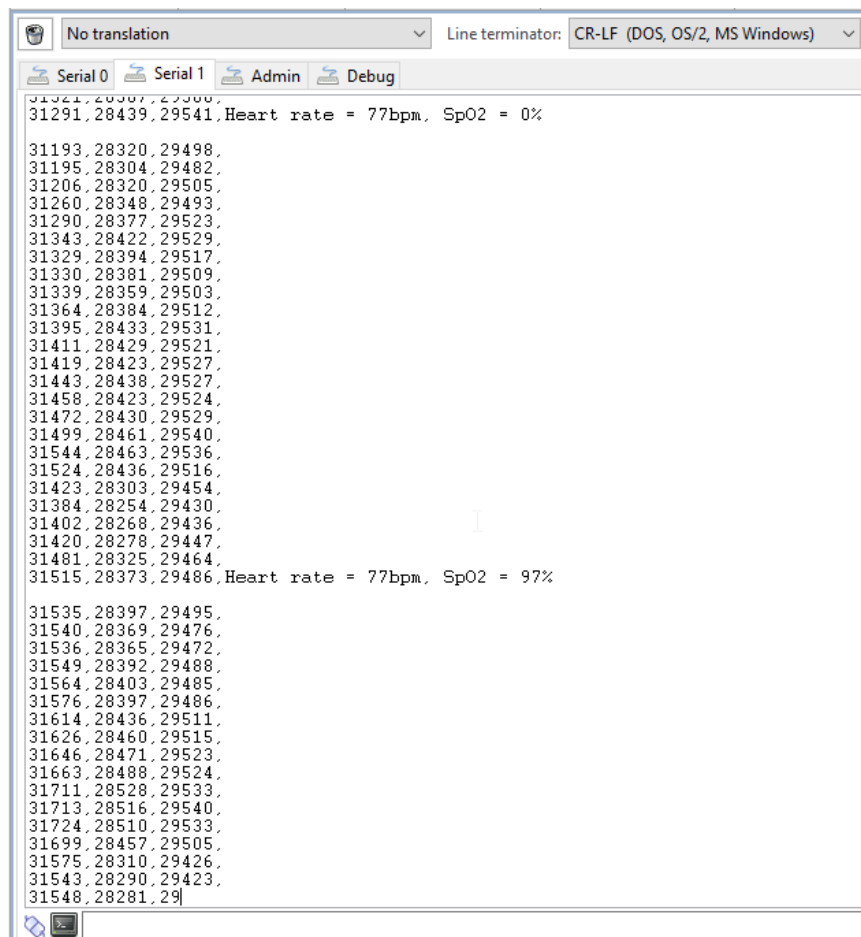
Figure 12: Heart Rate Result

## 5.4 USB Debug Mode

The demo firmware also implements a debug mode that streams HRM/SpO2 values along with sensor's raw data to a host PC via the USB interface. There's a compiler option `UART_DEBUG` with 4 different levels to choose from. If the `UART_DEBUG` is set to level 0, the USB debug mode will be disabled. Level 1 shows only raw data, level 2 shows only the Heart Rate and SpO2 values and level 3 shows both raw data, the Heart Rate and SpO2 values.

Users can launch the console in the Simplicity Studio (right click at the device under the Debug Adapters view and select "Launch Console...") or use a third-party terminal tool like TeraTerm to receive the data. A screenshot of the console output is shown in Figure 13.

The raw data in each column represents the measurement result from "green LED, IR LED, red LED". Heart rate and SpO2 values are updated once a second. The user can easily log the raw samples and debug messages to a \*.csv file for post-analysis.



The screenshot shows a serial console window with the following content:

```
No translation Line terminator: CR-LF (DOS, OS/2, MS Windows)
Serial 0 Serial 1 Admin Debug
31291,28439,29541,Heart rate = 77bpm, SpO2 = 0%
31193,28320,29498,
31195,28304,29482,
31206,28320,29505,
31260,28348,29493,
31290,28377,29523,
31343,28422,29529,
31329,28394,29517,
31330,28381,29509,
31339,28359,29503,
31364,28384,29512,
31395,28433,29531,
31411,28429,29521,
31419,28423,29527,
31443,28438,29527,
31458,28423,29524,
31472,28430,29529,
31499,28461,29540,
31544,28463,29536,
31524,28436,29516,
31423,28303,29454,
31384,28254,29430,
31402,28268,29436,
31420,28278,29447,
31481,28325,29464,
31515,28373,29486,Heart rate = 77bpm, SpO2 = 97%
31535,28397,29495,
31540,28369,29476,
31536,28365,29472,
31549,28392,29488,
31564,28403,29485,
31576,28397,29486,
31614,28436,29511,
31626,28460,29515,
31646,28471,29523,
31663,28488,29524,
31711,28528,29533,
31713,28516,29540,
31724,28510,29533,
31699,28457,29505,
31575,28310,29426,
31543,28290,29423,
31548,28281,29
```

Figure 13: Console Output

This concludes the BLE Pulse Oximeter Lab in how to flash the demo to a BMG220P device on the BLE Explorer Kit, take HRM/SpO2 measurements and view results in the EFR Connect mobile application.