UG447: Si117x Static HRM/SpO2 Software Demo for Wireless STK User's Guide

The Silicon Labs' Static HRM/SpO_2 software demo for Wireless Starter Kit (EFR32 MG) provides an example application to demonstrate the Silicon Labs' Static Heart Rate Monitor library (si117xhrm_static) on Silicon Labs' energy friendly EFR32™ Mighty Gecko MCU. The software demo requires a Wireless Starter Kit Mainboard (SLWSTK6020A), a EFR32 Mighty Gecko Radio Board (SLWRB4162A) and an EXP Sensor Adapter Board. In addition, the demo software works with a Si117x Wrist PPG EVB (Si117xK1-KIT) connected to the EXP Sensor Adapter board via the 6-pin ribbon cable to demonstrate HRM/SpO_2 operation on the finger-tip.

KEY POINTS

- Wireless Starter Kit (EFR32 MG) Demo
- Static Heart Rate Monitor library
- Wireless Starter Kit Mainboard (SLWSTK6020A)
- EFR32 Mighty Gecko Radio Board (SLWRB4162A)
- EXP Sensor Adapter Board

1. Introduction

Figure 1.1. Si117x Evaluation Kit Hardware with Wireless Starter Kit

The Si117x Static HRM/SpO_2 application demonstrates both heart rate monitoring (HRM) and oxygen saturation (SpO_2) on finger-tip. Red and IR LEDs are required to measure SpO_2. Therefore, it is recommended to use Si117xK1-KIT which has integrated green, red and IR LEDs for this demo. The demo software is available on Silicon Labs' website and the project can be imported into Simplicity Studio.

2. Software and Hardware Overview

2.1 Software

The software demo project, including the Si117x Static HRM/SpO_2 algorithm source code, can be downloaded from the Silicon Labs website. However, Silicon Labs will only provide limited support to the example code related questions. If you have any issue running the demo, please post the question on our forum: https://www.silabs.com/community/sensors/forum.

2.2 Hardware

Here's a list of hardware required to run the Si117x_Static_HRM demo:

- EFR32 Mighty Gecko Wireless SoC Starter Kit (SLWSTK6020A)
  - Use with Mighty Gecko Radio Board (SLWRB4162A)
- EXP Sensor Adapter Board
- Si117x PPG Wrist Evaluation Board (Si1171K1-KIT)
- 1 x Ribbon Cable

The Wireless STK and Mighty Gecko Radio Board can be ordered directly from silabs.com. However, the Si117x evaluation kits require a signed NDA first. Order it through your local sales representative.

3. Importing the Source Code Project into Simplicity Studio

The following steps will import the Si117x_Static_HRM Demo source code into Simplicity Studio resulting in a project that can be compiled, linked and debugged using the Wireless starter kit.

Simplicity Studio is available for download at https://www.silabs.com/products/development-tools/software/simplicity-studio.

1. Store the Si117x_Static_HRM software distribution uncompressed in a folder that is accessible from the PC.
2. tart Simplicity Studio and Open the Simplicity IDE by clicking on the Simplicity IDE Icon on the upper right.

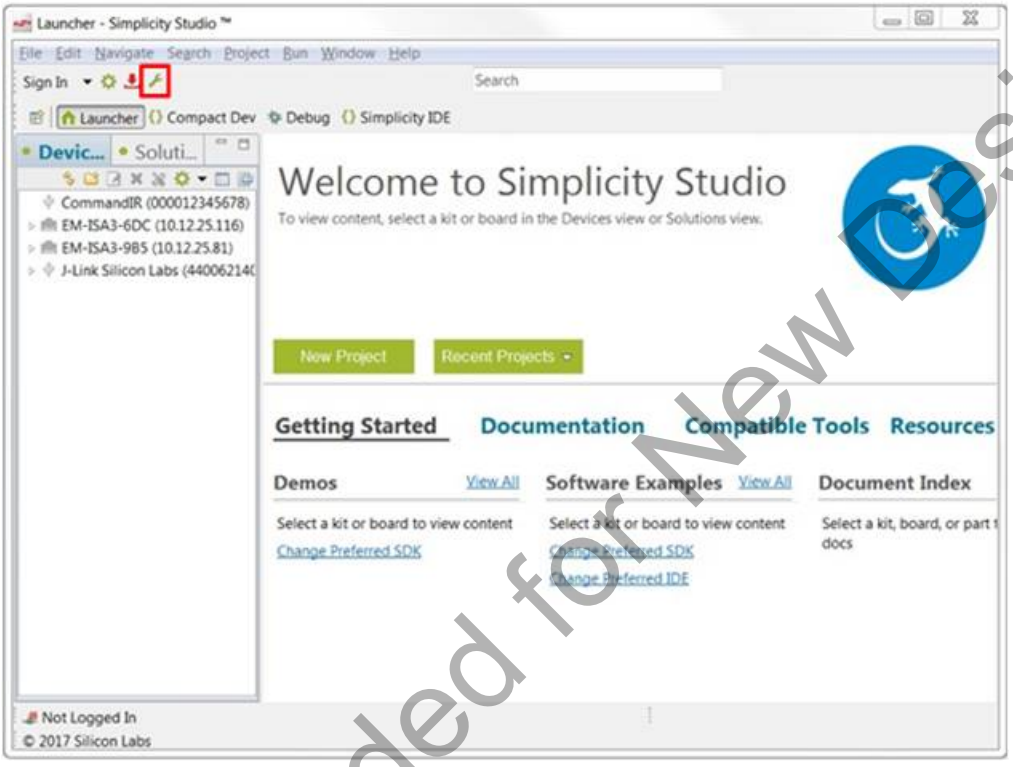

Figure 3.1. Simplicity Studio

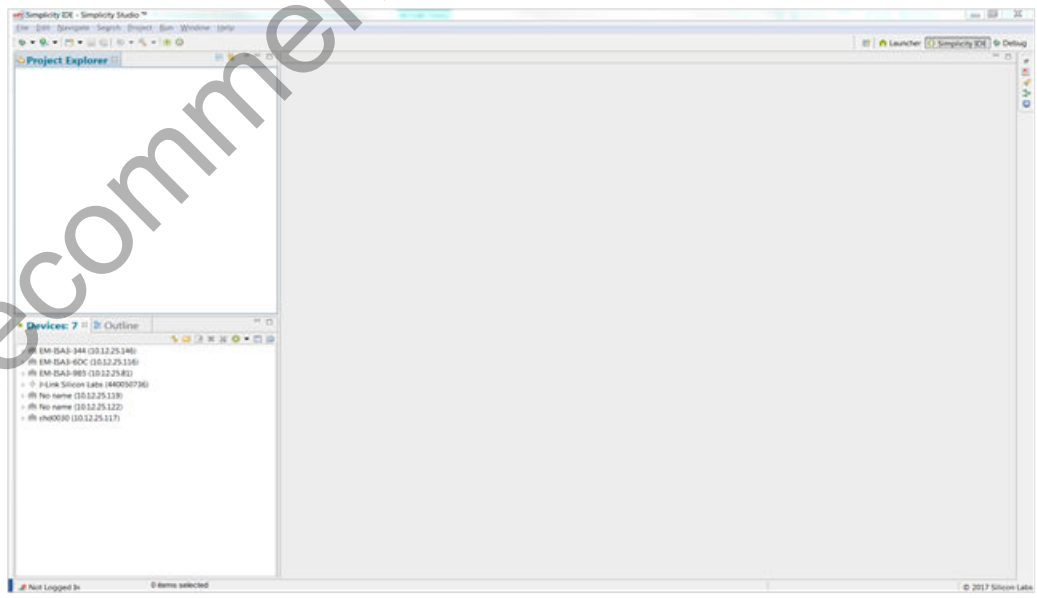

Figure 3.2. Simplicity IDE

3. In the Simplicity IDE menu, select **File->Import**.

4. Under the **General** heading, select **Existing Projects into the Workspace** then click **Next**.

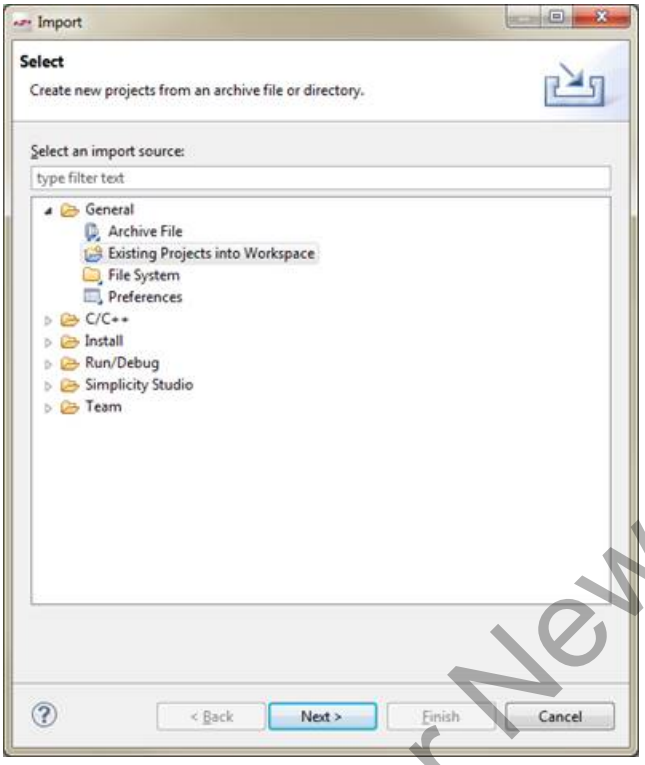

Figure 3.3. Simplicity IDE Import Dialog

5. elect **Root directory** then **Browse** to the folder Si117x_Static_HRM in the Software Demo distribution.

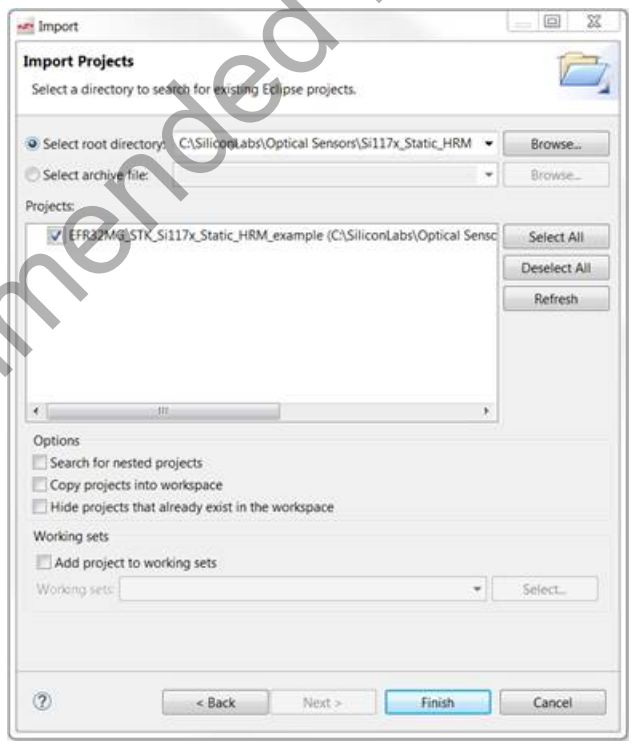

Figure 3.4. Select the Project to Import

6. Confirm that the check box for the project is checked.
7. If you wish to make a local copy within the Simplicity IDE workspace select **Copy projects into workspace**.

8. Click **Finish.**

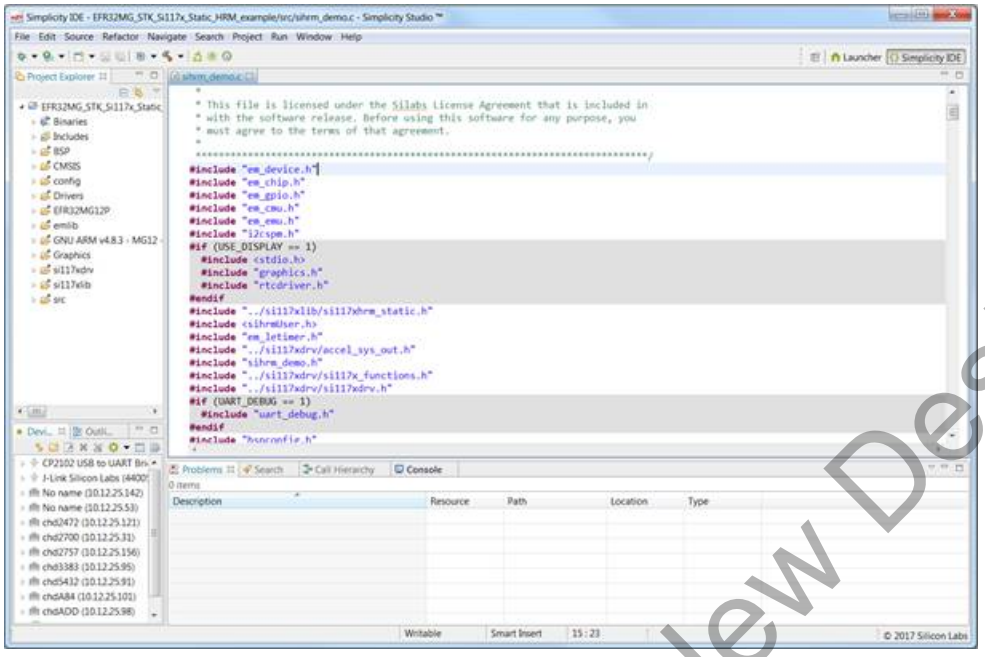

Figure 3.5. Simplicity IDE with Si117x_Static_HRM Demo Project

Upon completion of the steps above you should have a working project in Simplicity Studio that can be built and debugged.

1. To build the demo, simply click on the  icon in the Simplicity IDE. Note that there are two build configurations: Release and Debug. You can select between the two by clicking the down arrow next to the Hammer icon.

2. To debug the project, select the Debug build configuration then click on the  icon in the Simplicity IDE. A debug session will automatically launch and if a board is connected, the project will be loaded into the MCU.

4. Running the Demo

A Silicon Labs Wireless Starter Kit Mainboard (SLWSTK6020A), a EFR32 Mighty Gecko Radio Board (SLWRB4162A) and an EXP Sensor Adapter Board are needed to run the Si117x_Static_HRM Demo. To evaluate finger-tip based HRM/SpO_2, a Si117x Wrist PPG EVB (Si117xK1 module) and a 6-pin ribbon cable are needed.

The demo application uses the Wireless STK's LCD to display HRM/SpO_2 output, PB1 button to start/stop the measurement and PB0 button to cycle through the display modes. The full operation including startup sequence is illustrated in the figure below.

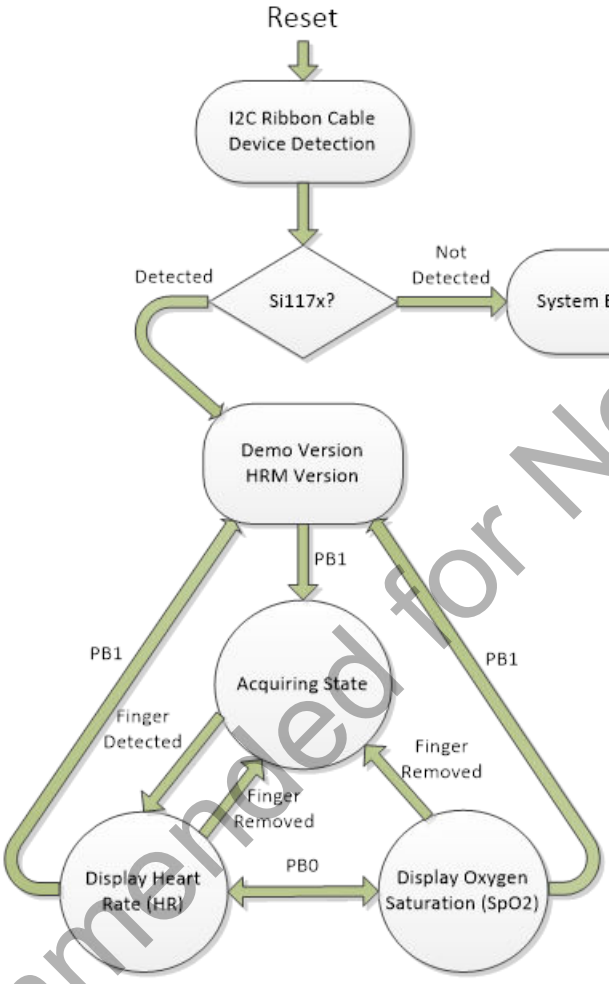

Figure 4.1. Si117x Static HRM Demo Startup Sequence and Display Modes

4.1 Demo Startup

Upon reset, the demo will first check whether Si117x sensor is detected through the ribbon cable. If a Si117x device is detected, the demo will read and display the demo version as well as the static HRM algorithm version on the LCD. If no Si117x device is detected, "System Error" message will be displayed on the LCD.

4.2 Start/Stop the HRM/SpO_2 Measurement

After successful detection of Si117x device, press PB1 button to start the static HRM/SpO_2 measurement. The user should put their finger on the sensor with proper pressure (solid contact between the finger and the sensor without optical leakage and don't press with force) and stay idle for at least 10 s. Once started, the measurement can be stopped at any time by pressing PB1 button again. The LCD display will switch to the main screen.

4.3 Acquiring State

When the measurement is started, the demo will enter the acquiring state first. While in the acquiring state, a valid finger contact is required to calculate HRM/SpO_2 result. The "Acquiring" message will be displayed on the LCD screen. During the measurement, removing the finger from the sensor or having unexpected movements will change the demo state back to acquiring state.

4.4 Switch Display

After starting the measurement, the user can press PB0 button to cycle between HRM and SpO_2 result display modes.

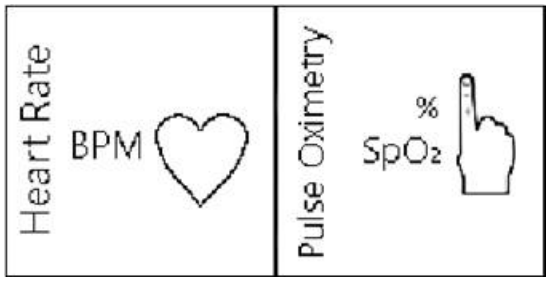

Figure 4.2. HRM and SpO_2 Display

5. USB Debug Mode

The Si117x_Static_HRM demo firmware includes a debug mode that enables HRM/SpO_2 data including raw samples from the sensor to be streamed to a host PC via the Wireless STK's USB interface.

5.1 Enabling USB Debug Mode

There is a complier option UART_DEBUG in the build configuration of the project. If the UART_DEBUG is set to 1, the USB debug mode will be enabled. The user can disable the USB debug mode to save power by clearing the UART_DEBUG option.

5.2 Connecting to USB Debug Mode on a PC

The user must first install the "JLink CDC UART" driver to use USB debug mode. The Wireless STK only has one USB type connector: a USB mini type connector labeled J-Link on the short side of the EVB. This port provides both USB connection as well as Debug capability using J-Link.

With the window driver software installed, USB debug mode enabled on the Mighty Gecko and a USB cable connecting the Wireless STK and the PC, the device will appear in Windows Device Manager as a COM port as shown in the figure below.

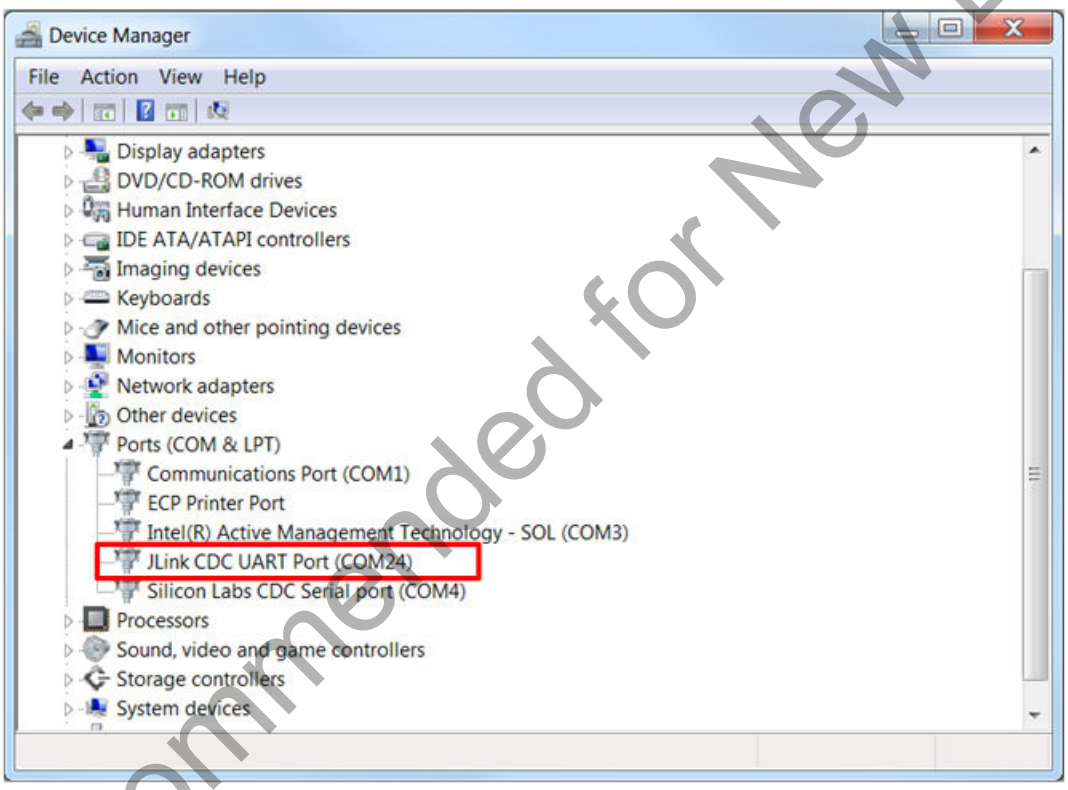

Figure 5.1. Si117x Static HRM COM Port Device in Windows Device Manager

5.3 USB Debug Output Data Format

The output data from USB debug port includes both raw samples from the sensor and the calculated HR/SpO_2 value from the algorithm. The user can use a third-party terminal tool like TeraTerm to receive the data from the USB. A screen shot of the console output is shown in the figure below.

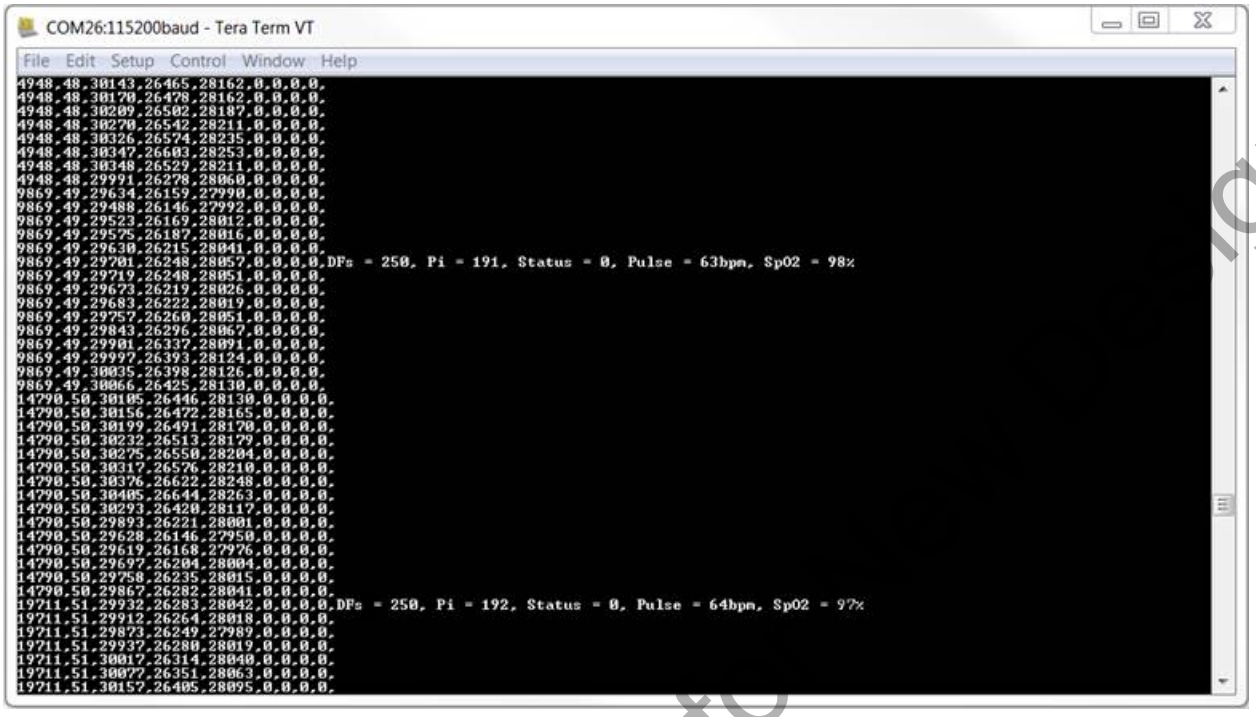

Figure 5.2. USB Debug Output Data

The raw data in each row represents "Timestamp, Sequence, PPG1, PPG2, PPG3, PPG4, ACCEL x-axis, ACCEL y-axis, ACCEL z-axis". The demo project utilizes the FIFO on Si117x to store samples. Therefore, the timestamp and sequence value are the same for raw samples from a single FIFO read. Besides, since the demo only enables PPG1, PPG2 and PPG3 channels to calculate HR/SpO_2, samples in other channels will always be 0.

Debug messages start with character "D". Heart rate and SpO_2 values are updated once a second. The user can easily log the raw samples and debug messages to a *.csv file.

6. Revision History

Revision 1.0

June, 2020

- Initial release.