

Spectral Data Viewer Manual

Sam Foucart
sf241616@ohio.edu

Spring 2020

Contents

1	Spectral Data Viewer	5
1.1	List of Prerequisite Software	5
1.2	Downloading Prerequisite Software	5
1.3	Building the Program From Source	5
1.3.1	Setting Up Android Studio	5
1.3.2	Opening SpectralDataViewer in Android Studio	7
1.4	Installing the Application on the Phone	10
1.5	Using the Application	10
2	Data Collector and Bluetooth Server	13
2.1	List of Prerequisite Software	13
2.2	Downloading Prerequisite Software	13
2.3	Building the Program From Source	14
2.4	Installing the Program as a Service	15
2.5	Using the Application	15

Chapter 1

Spectral Data Viewer

1.1 List of Prerequisite Software

- Android Studio
- git

1.2 Downloading Prerequisite Software

Android Studio can be downloaded from <https://developer.android.com/studio>. On MacOS and Linux, git will be already installed, but Windows users will need to download it from <https://git-scm.com/download/win>

1.3 Building the Program From Source

1.3.1 Setting Up Android Studio

Open the Android Studio application. The first time you open it, you will see a dialog that says "Import Android Studio Settings From...". It will not matter which option you pick. On the Android Studio Setup Wizard, click next until you reach the end. You can choose either the Standard or Custom install. Once you reach the end, click Finish. The application will then finish downloading the most recent APIs. Once everything finishes, click Finish.

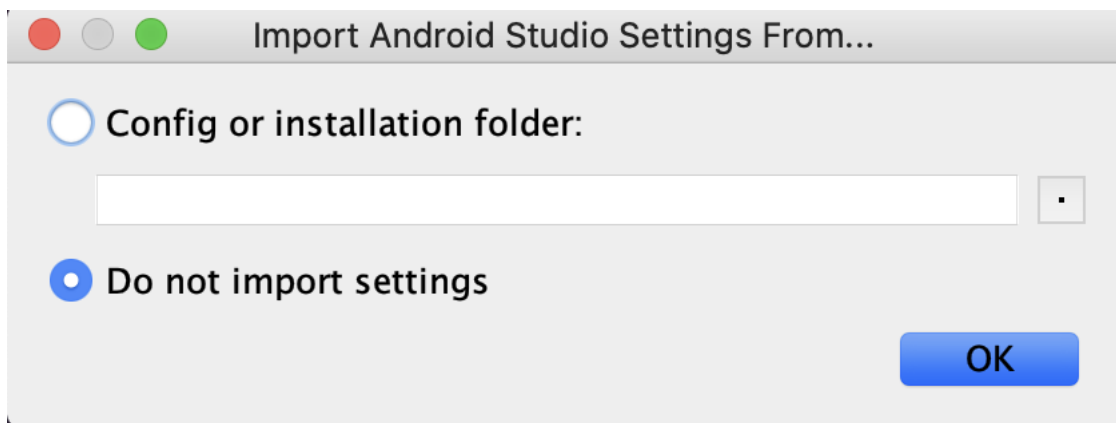


Figure 1.1: The Import Popup

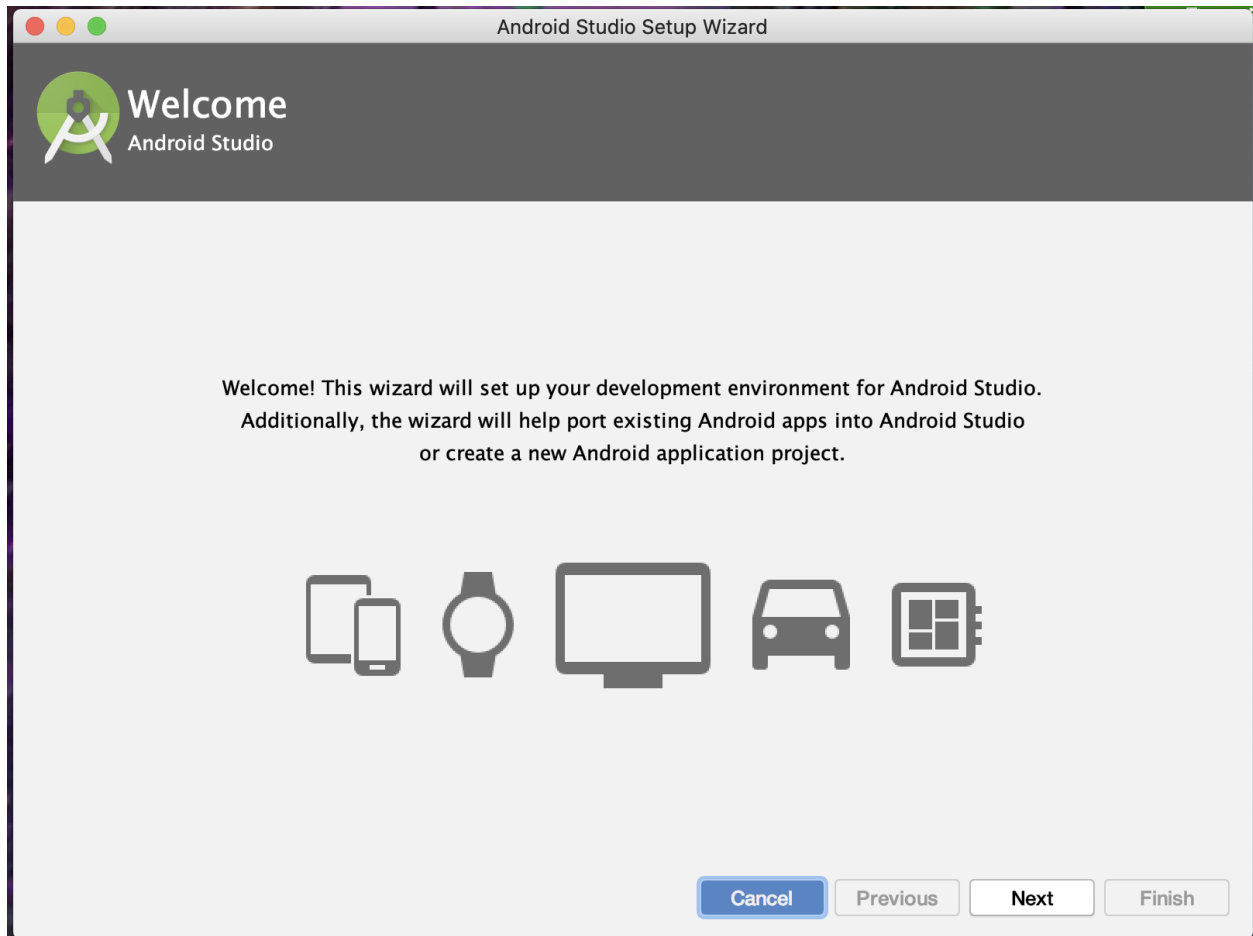


Figure 1.2: The Android Studio Wizard

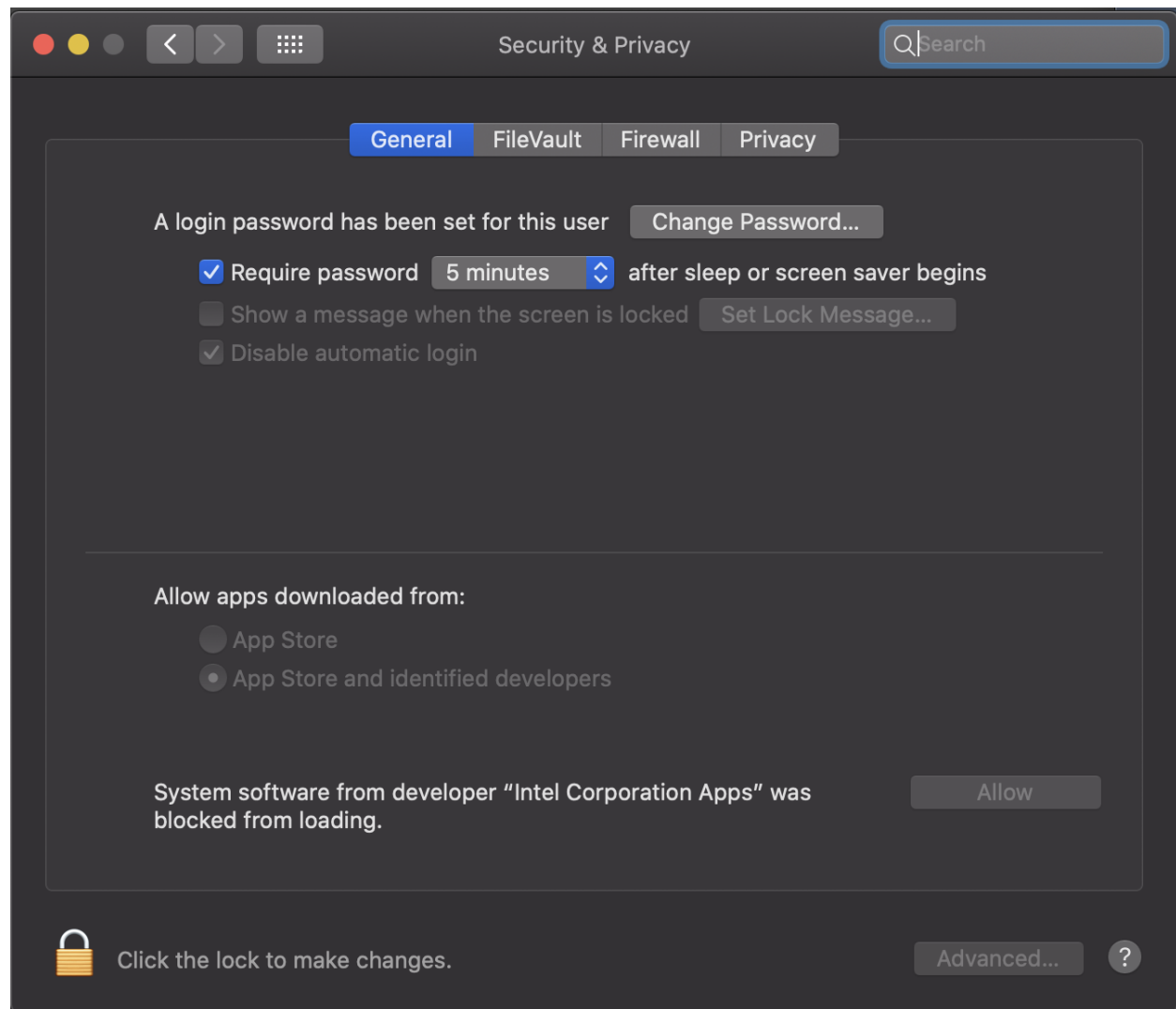


Figure 1.3: Screen to enable the system extension

NOTE: If on MacOS, you will see a warning, "System Extension Blocked". Click "Open Security Preferences", then click the lock at the bottom left to allow changes. After that, click "Allow".

1.3.2 Opening SpectralDataViewer in Android Studio

Now that Android Studio is setup, you should meet a launch wizard. To open the SpectralDataViewer project, click "Check out project from Version Control", then click "Git". In the URL text field, enter, https://github.com/samfoucart/oueecs_SpectralDataViewer, then click "Clone".

The project will now open, and you will be in the Android Studio IDE. The project will also start to build automatically. You can see the status of the build in the bottom left of the screen. It will take a couple minutes to build the first time opening Android Studio.

NOTE: If the Android SDK is missing the build will fail. To fix this, click "Install missing SDK package(s)". Then click Accept and Next. The build will then complete.

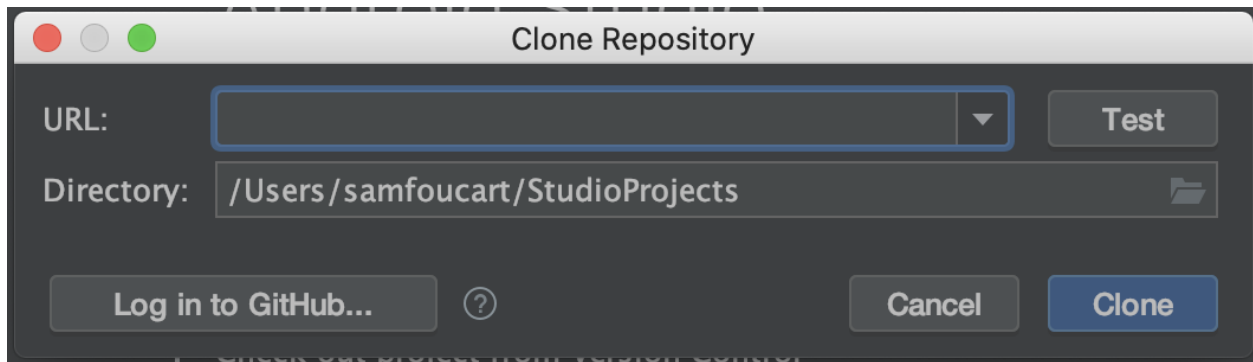


Figure 1.4: Cloning the Project From GitHub

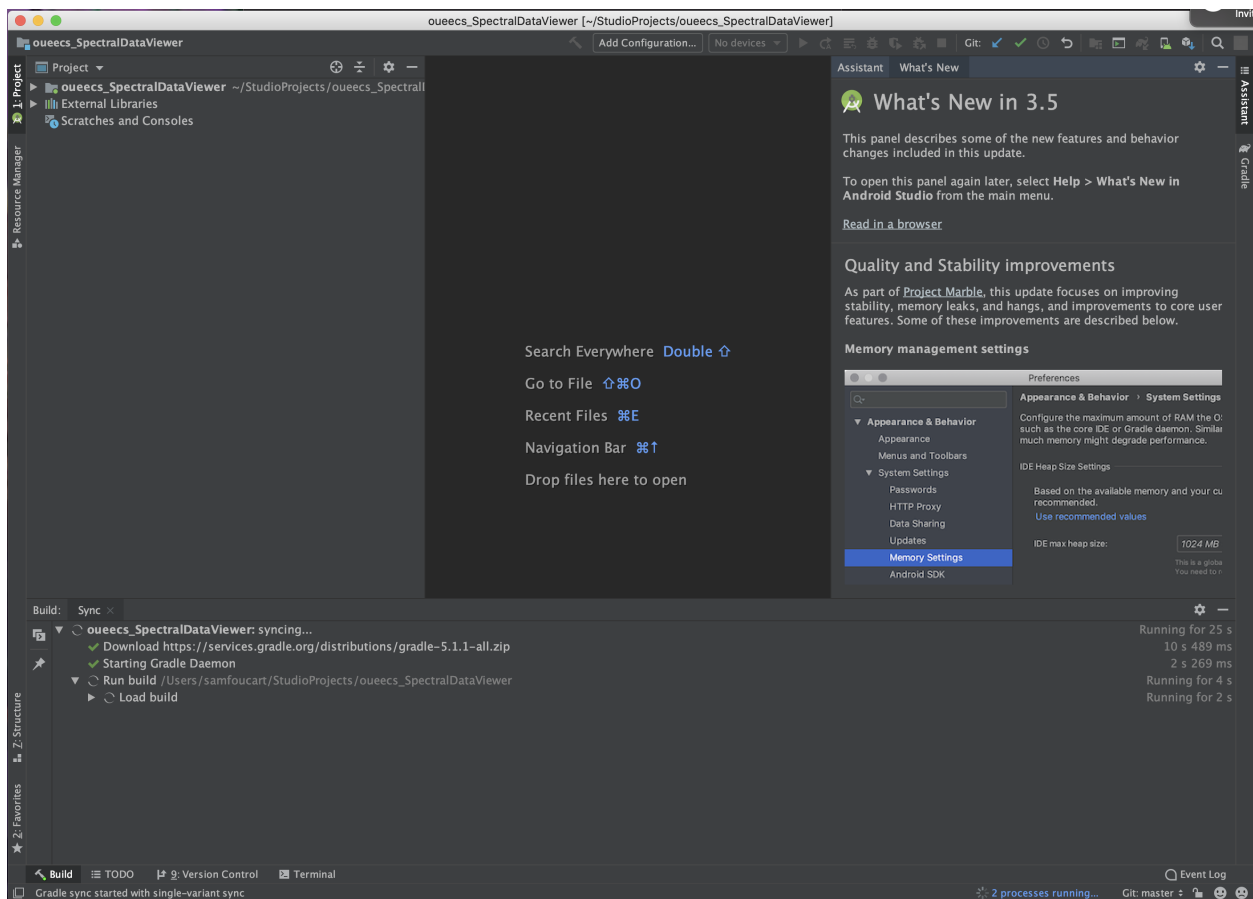


Figure 1.5: The Android Studio IDE


```
ERROR: Failed to install the following Android SDK packages as some licences have not been accepted.
  build-tools;28.0.3 Android SDK Build-Tools 28.0.3
  platforms;android-28 Android SDK Platform 28
To build this project, accept the SDK license agreements and install the missing components using the Android Studio SDK
Alternatively, to transfer the license agreements from one workstation to another, see http://d.android.com/r/studio-ui/

Using Android SDK: /Users/samfoucart/Library/Android/sdk
Install missing SDK package(s)
```

Figure 1.6: The error when the SDK packages are not installed

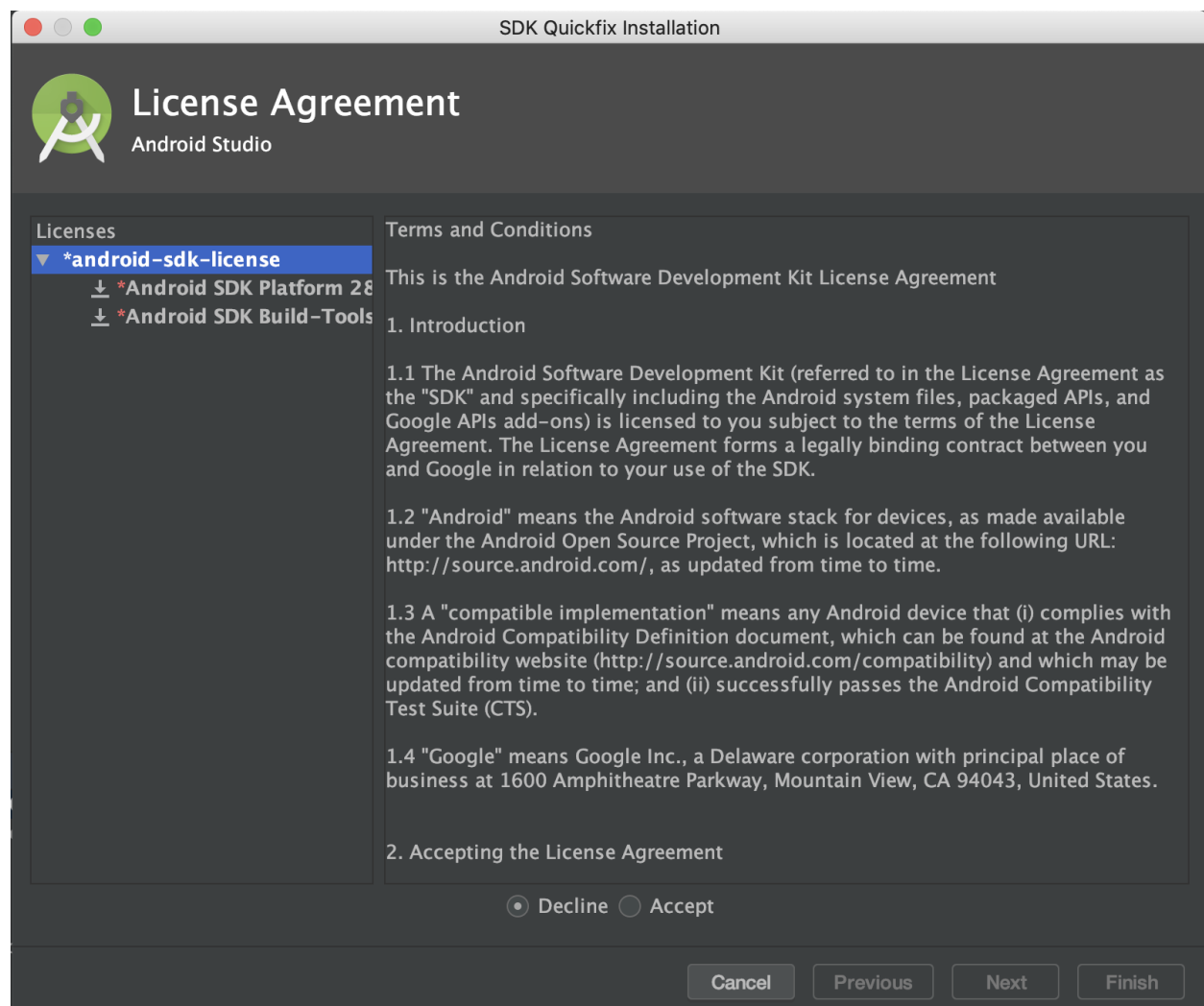


Figure 1.7: The SDK Setup Screen

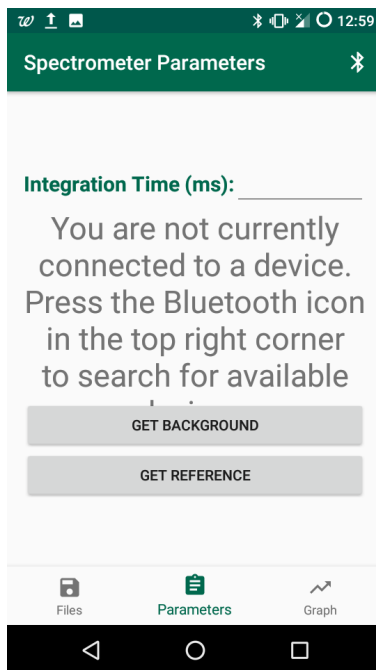


Figure 1.8: The Home Screen indicating that the user is not connected

1.4 Installing the Application on the Phone

Plug in the phone to the computer, and put the device in development mode. This varies from phone to phone, so the device instructions will need to be referenced. Then USB debugging will need to be enabled in the developer options of the phone.

Then the user can open Android Studio and push the green Play button in the top center of the screen. It will compile the application install the application to the device that is selected.

1.5 Using the Application

When the device is first opened, text will be in the center of the screen indicating that you are not currently connected to the raspberry pi. Connect to the device that you want by pushing the bluetooth icon in the top right corner. The text should then disappear from the center of the screen. Now select an integration time, then push get background. The background will then be viewable in the graph tab. To save purely the background, give the file a name and push save. Then push get reference. The app will subtract the background from the reference automatically. The reference will then be able to be viewed in the graph tab, and can be saved and opened later in the saved tab.

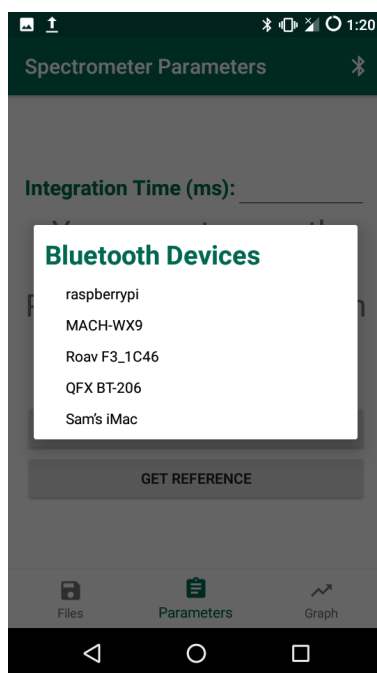


Figure 1.9: The popup showing the available connections

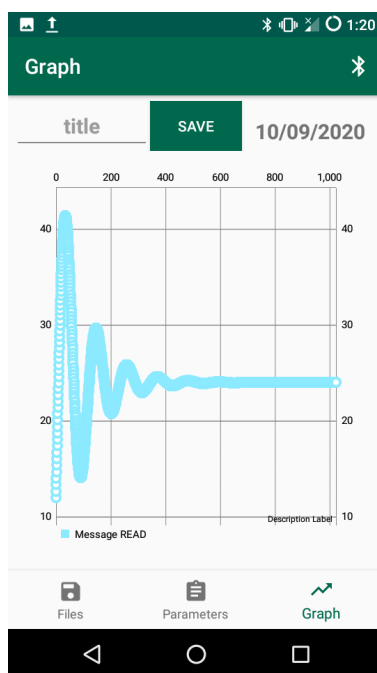


Figure 1.10: A normal background collected using the test collector endpoint

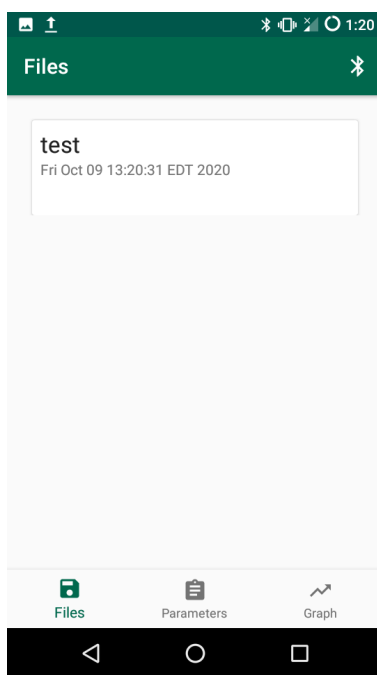


Figure 1.11: The screen indicating a file has been saved

Chapter 2

Data Collector and Bluetooth Server

2.1 List of Prerequisite Software

- Python 2.7
- git
- make
- SeaBreeze
- libusb-dev
- libbluetooth-dev
- blueman
- bluez
- pybluez version 0.22

2.2 Downloading Prerequisite Software

- Download Raspberry Pi OS <https://www.raspberrypi.org/downloads/raspberry-pi-os/>
- Use balena etcher to transfer the .img file to the sd Card
- Follow the prompts to set up the operating system and update software
- Open Terminal and type:

```
mkdir PACE
cd PACE
git clone https://www.github.com/samfoucart/oueecs_SpectralDataCollector
```

- Then change directory back to the PACE root:

```
cd ..
```

- Download the seabreeze API from Ocean Optics: <https://sourceforge.net/projects/seabreeze/>
- Unzip the project into /PACE/seabreeze-3.0.11

- Then download the dependencies:

```
sudo apt-get install libusb-dev
sudo apt-get install libbluetooth-dev
sudo apt-get install blueman
sudo pip install pybluez==0.22
```

2.3 Building the Program From Source

- Then change directory into the seabreeze root:

```
cd ~/PACE/seabreeze-3.0.11/SeaBreeze/
```

- Then modify the makefile for building on linux by typing:

```
sudo nano common.mk
```

- Then delete line 9 which says:

```
-Werror \
```

- Now to build type:

```
sudo make
sudo cp lib/libseabreeze.so /usr/lib/libseabreeze.so
```

- Now to build the data collector we change into the data-collector root:

```
cd ~/PACE/oueecs_spectraldatacollector/data-collector/
```

- Then to compile the program, type in all as one line:

```
g++ src/data-collector.cc -I /home/pi/PACE/seabreeze-3.0.11/SeaBreeze/include/
-L /home/pi/PACE/seabreeze-3.0.11/SeaBreeze/lib -lseabreeze -lpthread
-lstdc++ -lusb -o build/data-collector
```

- To build a test endpoint, replace data-collector.cc with test-collector.cc in the preceding line.

```
g++ src/test-collector.cc -I /home/pi/PACE/seabreeze-3.0.11/SeaBreeze/include/
-L /home/pi/PACE/seabreeze-3.0.11/SeaBreeze/lib -lseabreeze -lpthread
-lstdc++ -lusb -o build/test-collector
```

- Now the we must modify a system file to run bluetooth in compatibility mode. To do this type:

```
sudo nano /etc/systemd/system/dbus-org.bluez.service
```

- Then find the line that says:

```
ExecStart=/usr/lib/bluetooth/bluetoothd
```

and add a -C to the end of it so that the line now reads:

```
ExecStart=/usr/lib/bluetooth/bluetoothd -C
```

- Now to restart the bluetooth of the device type:

```
sudo systemctl daemon-reload
sudo systemctl restart bluetooth
```

and restart the raspberry Pi. Nothing will happen if you do not restart the device.

- Now type:

```
sudo chmod 777 /var/run/sdp
sudo sdptool add SP
```

and restart the device again.

- Now the last step is to make the device discoverable. Click the bluetooth icon in the top right corner of the screen and select “Make Discoverable”

2.4 Installing the Program as a Service

This step used to work in 2019, but when reinstalled in 2020 it was unable to be made work again. This feature allows the device to run the data-collector as soon as it receives power from a battery or the wall. To install the device as a service, type:

```
cd ~/PACE/oueecs_SpectralDataCollector
sudo cp service/data-collector.service /lib/systemd/system/data-collector.service
sudo systemctl start data-collector.service
```

2.5 Using the Application

To use the application, type:

```
cd ~/PACE/oueecs_spectraldatacollector/bluetooth-socket-server/
sudo python bluetooth-socket-server.py
```

To point the bluetooth socket server towards a test endpoint that doesn't check for the spectrometer, recompile the test-collector program but name it data-collector.