MANUAK

EMI-Toolkit App

Kiet Hoang (21256877)
Clariza Look (22860721)
Arjun Panicker (22954837)
Deepakraj Sugumaran(22789952)
Harper Wu (23052765)

Introduction

Figure 1: EMI-Toolkit App

What is EMI-Toolkit?

1. EMI-Toolkit is an open-source application to collect raw sensor values from a DUALEM Sensor and a compatible NMEA format GPS Sensor.
2. It is designed to run smoothly on a RaspberryPi 3/4.
3. It can run on: Debian based Linux, macOS and Windows 10.

We thank you for using our application and would love to receive feedback (Positive or Otherwise). We also request you to report bugs found (if any) on our GitHub Repository.

KIET HOANG (21256877)
CLARIZA LOOK (22860721)
ARJUN PANICKER (22954837)
DEEPAKRAJ SUGUMARAN(22789952)
HARPER WU (23052765)

Installation

System Requirements

The EMI Toolkit software comes with a package file that can be installed to any computing or microcomputer e.g., Raspberry Pi or a laptop computer.

- **Raspberry Pi**
    1. Raspberry Pi (model 3 and above)
    2. SD Card with pre-installed Raspbian OS (Linux OS)
    3. Monitor/ Screen to plug into Raspberry Pi
    4. Keyboard to control the Raspberry Pi
- **Windows PC/Laptop**
    1. Windows 10
    2. Memory: 2GB RAM +
    3. Storage: 120 GB SSD+
- **Macintosh**
    1. macOS 10.0 and above
    2. Memory: 2GB RAM +
    3. Storage: 120 GB SSD+

KIET HOANG (21256877)
CLARIZA LOOK (22860721)
ARJUN PANICKER (22954837)
DEEPAKRAJ SUGUMARAN(22789952)
HARPER WU (23052765)

GETTING CORE FILES

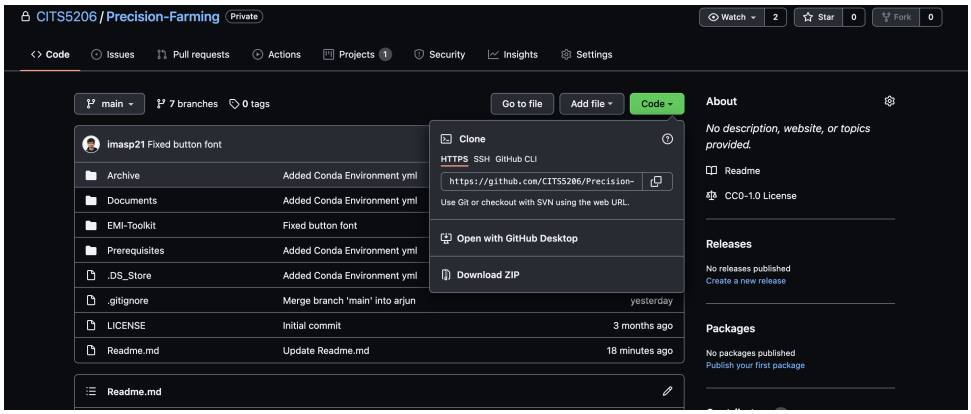

Figure 2: Project's GitHub Repo

Downloading the files

1. Download the zip file from GitHub to your computer.
2. To download from GitHub, your email should have been added to a GitHub project CITS5206/Precision-Farming

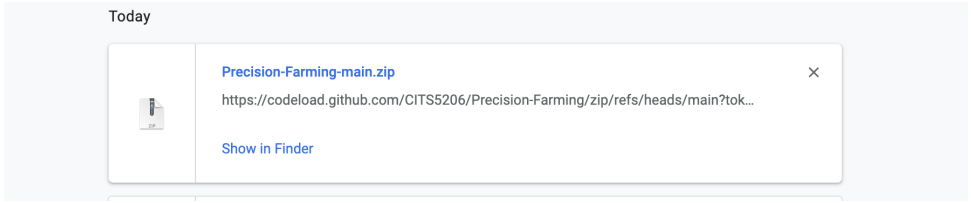

Figure 3: Downloaded Files

Extracting the downloaded files

1. After downloading the zip file to your computer, right-click the ZIP file.
2. In the drop-down menu, click "Extract All. . . " The zip wizard will appear.
3. If you want to unzip the files to a different folder, click "Browse. . . " and choose a location.
4. Click "Extract" and the files will be unzipped and copied to the folder you chose

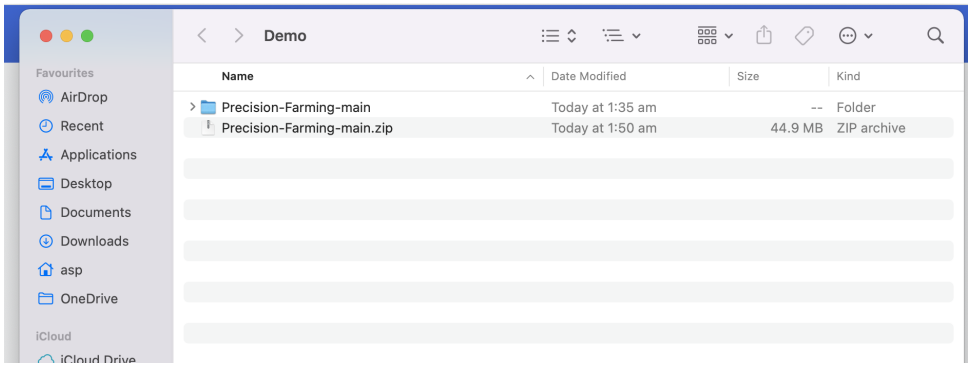

Figure 4: Extracted zip file

KIET HOANG (21256877)
CLARIZA LOOK (22860721)
ARJUN PANICKER (22954837)
DEEPAKRAJ SUGUMARAN(22789952)
HARPER WU (23052765)

SOFTWARE PACKAGES

Following dependent packages are are required to run the software.

- Anaconda (Virtual Environment) : Download Anaconda

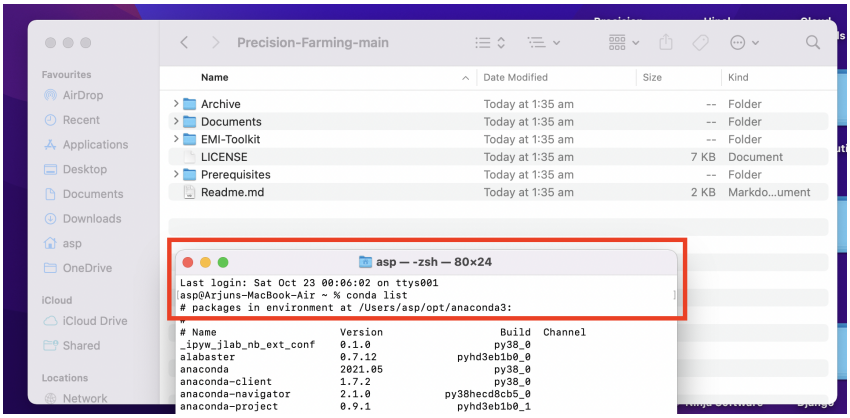

Figure 5: Open Anaconda Terminal [Windows] or Terminal [Others] and type conda list to verify conda has been succesfully installed

- Open terminal [Linux/MacOS] or Open Anaconda Terminal [Windows]
- Navigate PrecisionFarming-main/Prerequisites/

  `cd PrecisionFarming-main/Prerequisites/`
- Use the following command to create a conda environment [Windows]

  `conda env create -f windows-precision-farming-emitoolkit.yml`
- Use the following command to create a conda environment [MacOS]

  `conda env create -f mac-precision-farming-emitoolkit.yml`

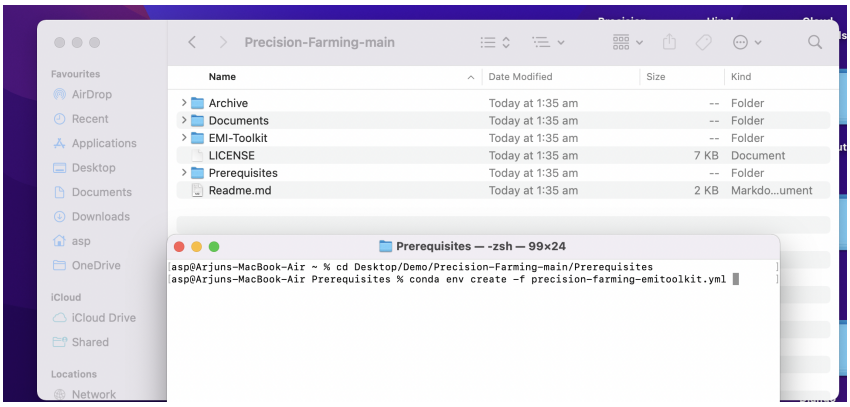

Figure 6: Running .yml file to create custom environment in conda

- Activate the environment by

  `conda activate emitoolkit`

KIET HOANG (21256877)
CLARIZA LOOK (22860721)
ARJUN PANICKER (22954837)
DEEPAKRAJ SUGUMARAN(22789952)
HARPER WU (23052765)

DOWNLOADING MAP DATA (QGIS)

Qtiles Plugin is needed to download tile-data from QGIS

- QGIS : Download QGIS

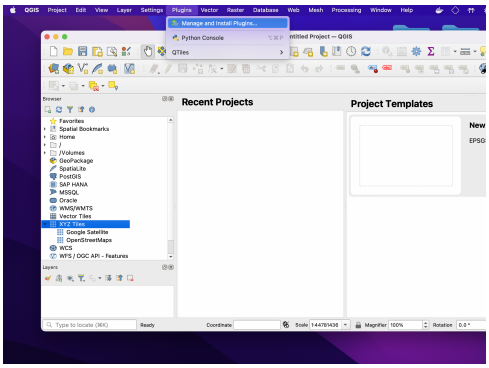

Figure 7: Open QGIS and Click on Plugins – Manage Plugins

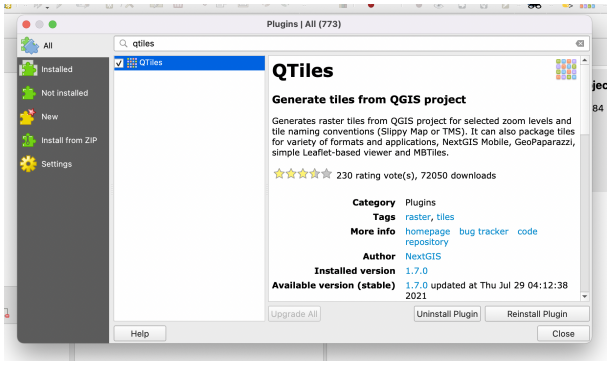

Figure 8: Search Qtiles and install pluging

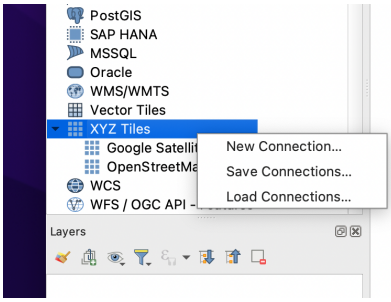

Figure 9: From the left side panel add new connection under XYZ Tiles

KIET HOANG (21256877)
CLARIZA LOOK (22860721)
ARJUN PANICKER (22954837)
DEEPAKRAJ SUGUMARAN(22789952)
HARPER WU (23052765)

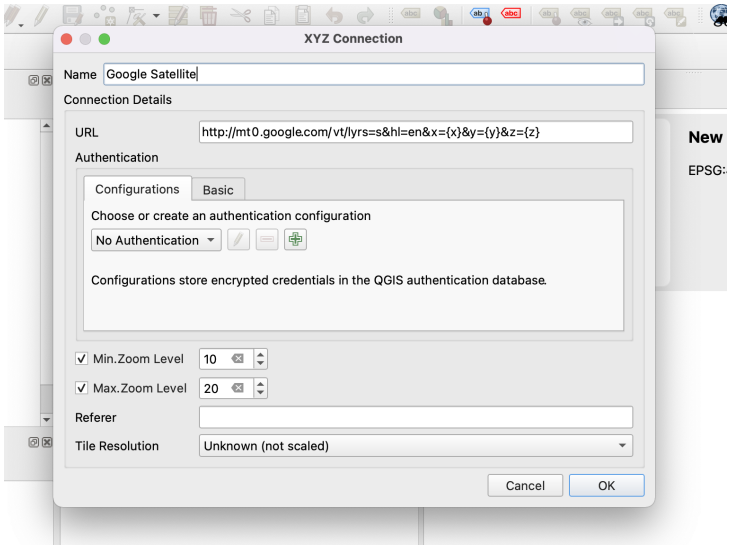

Figure 10: Add the following details in the new connection Google Satellite Link

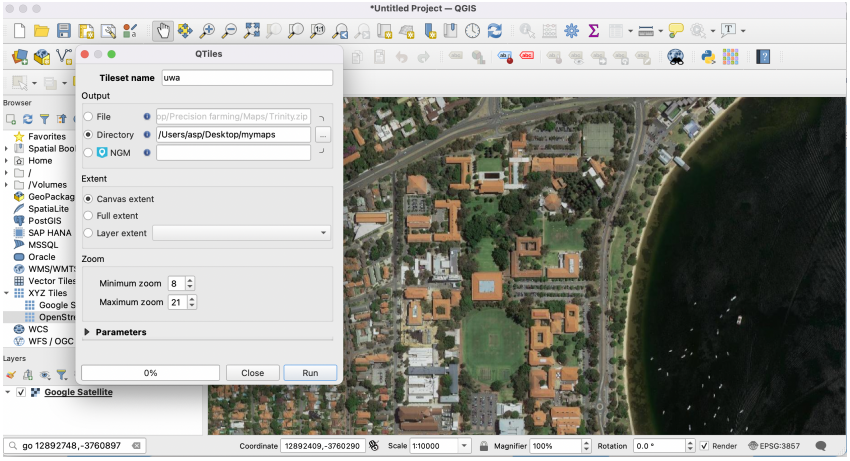

Figure 11: Set Map name and output folder and a zoom level between 8 to 21

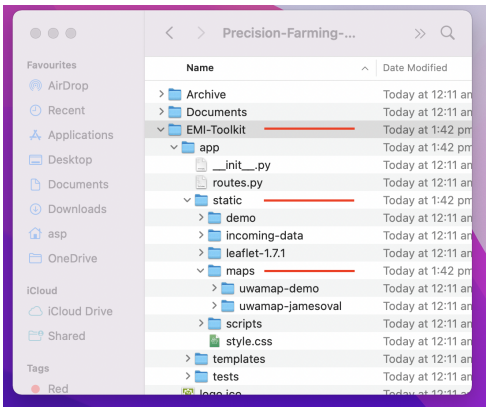

Figure 12: Copy the newly generated map to ”app/static/maps/” folder

KIET HOANG (21256877)
CLARIZA LOOK (22860721)
ARJUN PANICKER (22954837)
DEEPAKRAJ SUGUMARAN(22789952)
HARPER WU (23052765)

Running the Program

ON DESKTOPS

Executing the program on desktop environment

1. Navigate to EMI-Toolkit Folder [Anaconda Terminal for Windows], [Terminal for others]

```
cd ..
cd EMI-Toolkit/
```

2. activate the conda virtual environment by,

```
conda activate emitoolkit
```

3. start the program by,

```
python3 main.py
```

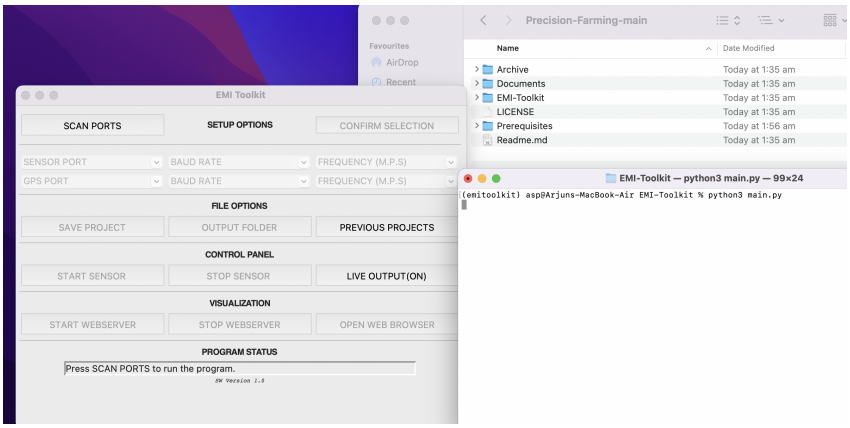

Figure 13: Program running successfully inside conda environment

KIET HOANG (21256877)
CLARIZA LOOK (22860721)
ARJUN PANICKER (22954837)
DEEPAKRAJ SUGUMARAN(22789952)
HARPER WU (23052765)

ON RASPBERRYPI 3/4

Executing the program on raspberry pi

> **Note**
>
> *EMI Toolkit App is not included in GitHub Repository as it needs to be recompiled for each new version)*

1. Update the system

```
sudo apt-get update
sudo apt-get upgrade
```

2. Navigate to the Prerequisites Folder and install dependent packages

```
python3 -m pip install -r *raspberry-pi-requirements.txt*
```

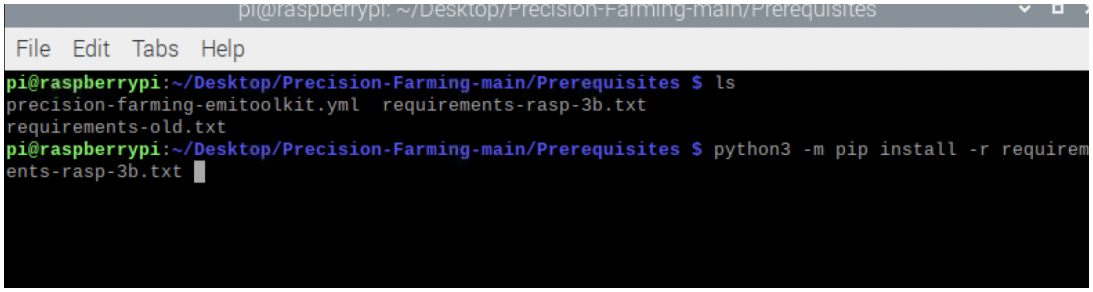

Figure 14: Dependent packages installation on Pi 3/4

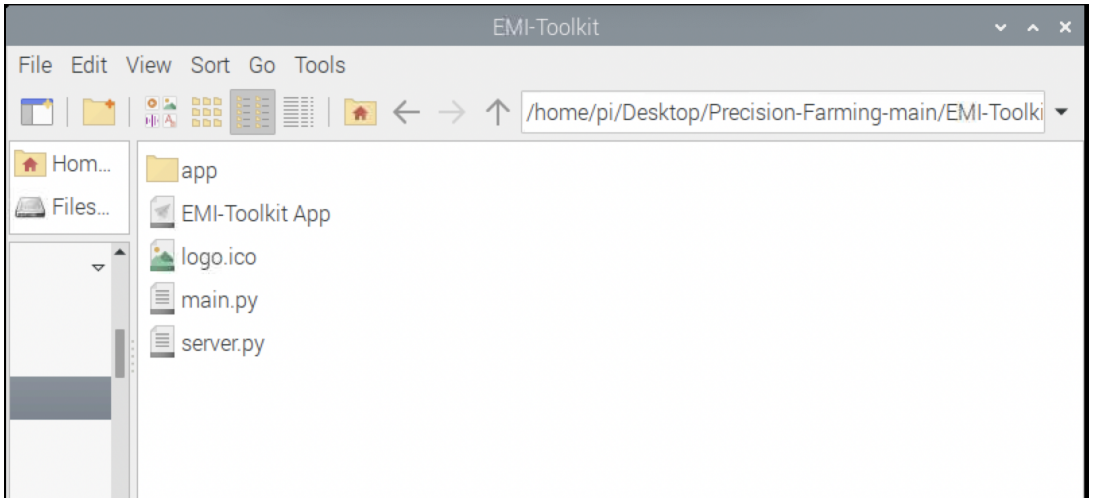

Figure 15: Open EMI-Toolkit Folder, and run EMI-Toolkit App

KIET HOANG (21256877)
CLARIZA LOOK (22860721)
ARJUN PANICKER (22954837)
DEEPAKRAJ SUGUMARAN(22789952)
HARPER WU (23052765)

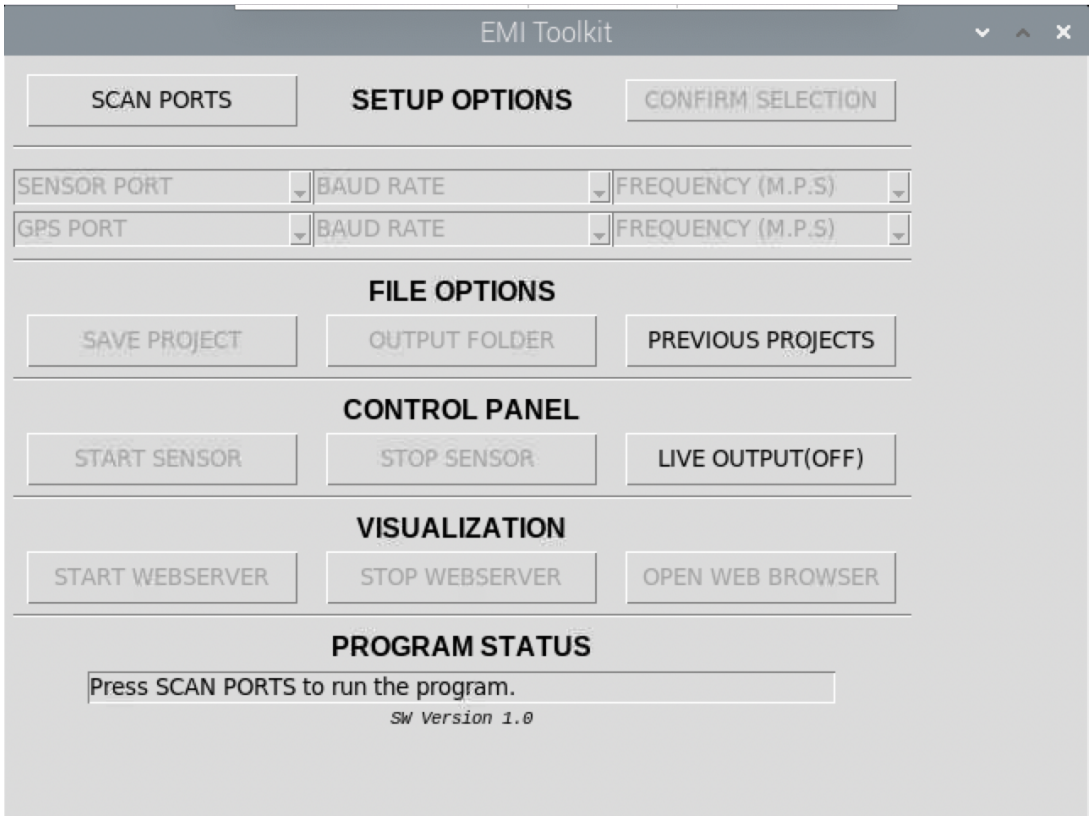

Figure 16: Program running successfully on Raspberry Pi

EMI-Toolkit App

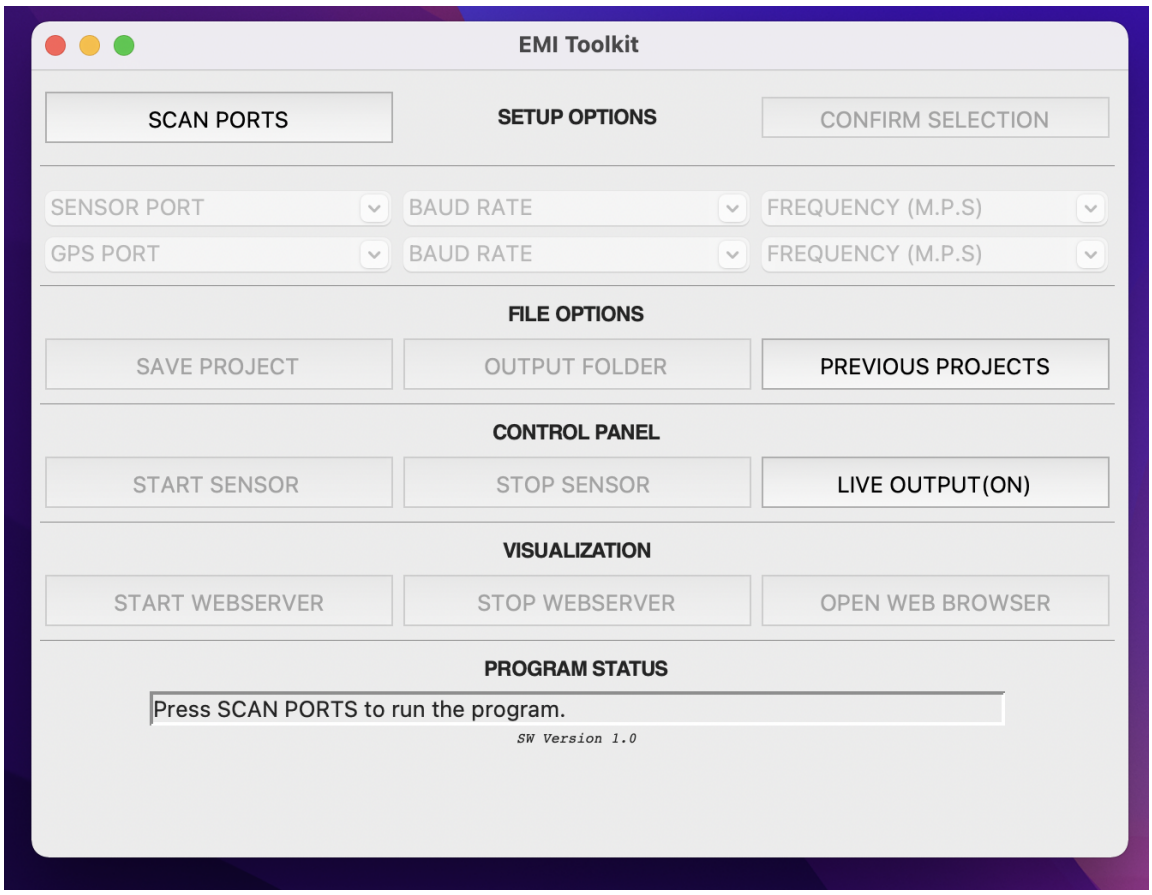

Figure 17: EMI-Toolkit App (v1.0.0) - Note, it will look different on Windows/Linux

EMI-Toolkit [Alert]

Make sure you are in the EMI-Toolkit Folder before running the program

EMI-Toolkit [Alert]

Make sure the program is running without any graphical glitches

The software provided to you has conveniently been implemented into a package with user interface. Following section will explain each button's and menu's functionality and expected results.

KIET HOANG (21256877)
CLARIZA LOOK (22860721)
ARJUN PANICKER (22954837)
DEEPAKRAJ SUGUMARAN(22789952)
HARPER WU (23052765)

BUTTONS

The App has 5 main sections namely Setup, File, Control, Visualization, Program Status.

Setup Options

1. SCAN PORTS: This feature will check if the sensor and GPS are connected
2. CONFIRM SELECTION: Confirms the selected settings for the current project
3. SENSOR PORT: Once the sensor is plugged in, this part will give a drop-down list of the connected sensor. The user should choose the right sensor name)
4. GPS PORT: Once the GPS module is plugged in, this part will give a drop down list of the connected GPS module. The user should choose the right GPS device
5. BAUD RATE: This part will give a drop-down list of the options for the baud Rate for sensor and GPS devices. The user can choose the baud rate.
6. FREQUENCY: This part will give a drop-down list of the options for the frequency of tracking the sensor and GPS devices. The user can choose the frequency.

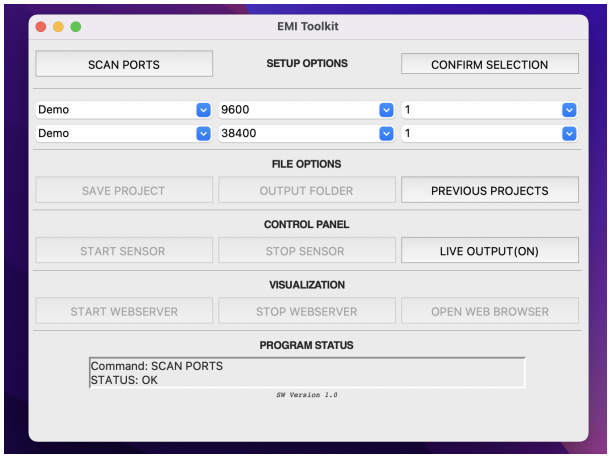

Figure 18: SETUP OPTIONS – CMD: SCAN PORTS

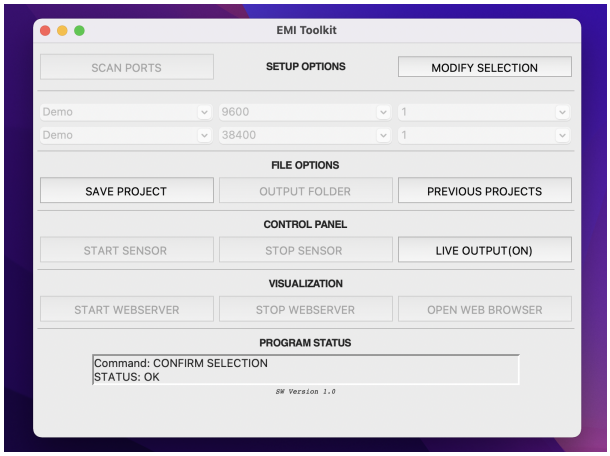

Figure 19: SETUP OPTIONS – CMD: CONFIRM SELECTION

KIET HOANG (21256877)
CLARIZA LOOK (22860721)
ARJUN PANICKER (22954837)
DEEPAKRAJ SUGUMARAN(22789952)
HARPER WU (23052765)

File Options

1. SAVE PROJECT: This will let the user create or save a project into the laptop or Raspberry Pi
2. OUTPUT FOLDER: Open selected project's output folder.
3. PREVIOUS PROJECTS: Open past projects folder defaults to "/Documents/EMI-Toolkit/"

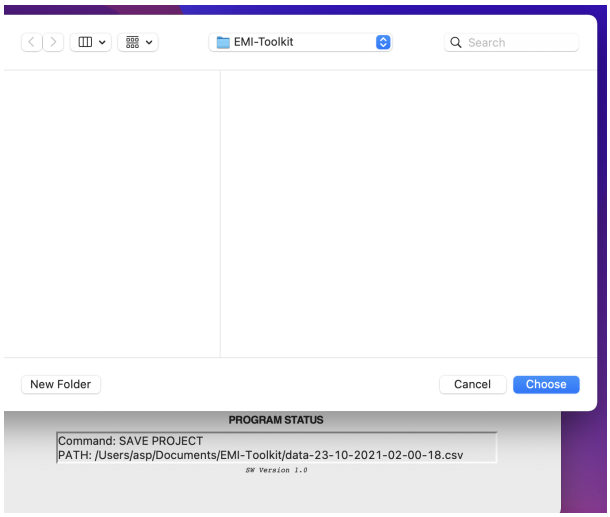

Figure 20: FILE OPTIONS – CMD: SAVE PROJECT

Control Options

1. START SENSOR: [Requires Confirm Selection] Starts the sensor reading.
2. STOP SENSOR: [Requires Confirm Selection] Stops the sensor reading.
3. LIVE OUTPUT (ON/OFF): Toggles Live output stream on to the Program Status Bar.

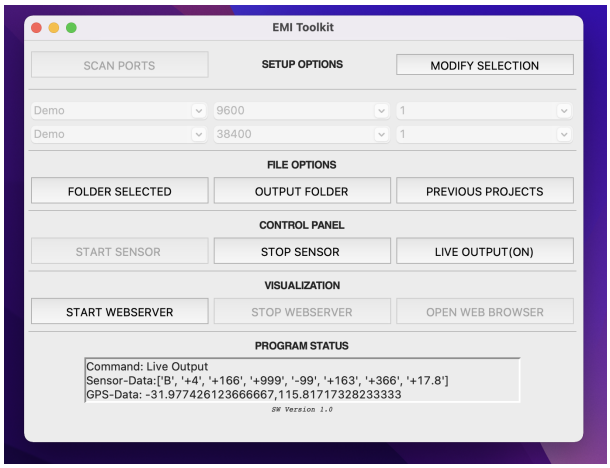

Figure 21: CONTROL PANEL – CMD: START SENSOR [Note: DEMO MODE]

KIET HOANG (21256877)
CLARIZA LOOK (22860721)
ARJUN PANICKER (22954837)
DEEPAKRAJ SUGUMARAN(22789952)
HARPER WU (23052765)

Visualization

1. START WEBSERVER: [Requires START SENSOR Selection] Starts the Flask web server.
2. STOP WEBSERVER: [Requires START WEBSERVER Selection] Stops the Flask web server.
3. OPEN BROWSER: Open's user's default web browser pointing to localhost:3152 .

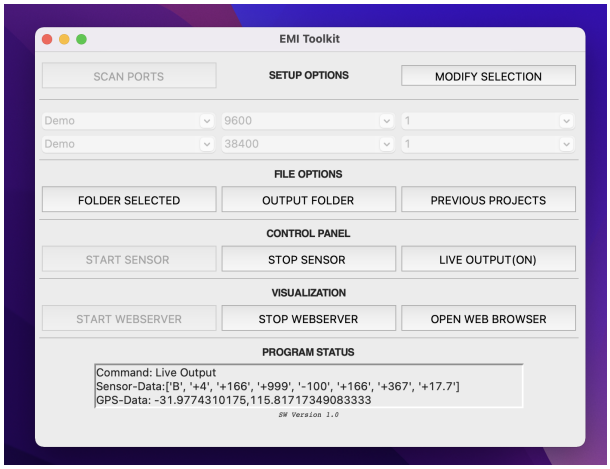

Figure 22: VISUALIZATION – CMD: START WEBSERVER [Note: DEMO MODE]

PROGRAM STATUS

Program Status

1. Program Status Bar displays information sent by the background processes.
2. Any error or issues will be first displayed here.
3. The outputs are in the following format

```
CMD|COMMAND : "MOST RECENT BUTTON PRESS"
STATUS: OK | ERROR | WARNING | FAILURE
```


KIET HOANG (21256877)
CLARIZA LOOK (22860721)
ARJUN PANICKER (22954837)
DEEPAKRAJ SUGUMARAN(22789952)
HARPER WU (23052765)

GUI - SOURCE CODE

Please refer to the table below for complete list of Classes and it's methods present in the current version. Note this is a top level view, check GitHub (Click here to Visit) for in-depth review of the code.

```
class GUI():
        # EMI Toolkit GUI Class
```

Methods	Description
def __init__(self):	Initialize a GUI Window and returns nothing
def GUIWindow(self)	Initializes the main window
def GUIMainloop(self)	Initializes the MAINLOOP
def GUIFrame(self)	Initializes the frame settings
def GUIVariables(self):	Initializes the variables requires in the GUI
def GUISetupOptions(self)	Initializes the setup options and buttons
def GUIControlOPtions(self)	Initializes the control panel options and buttons
def GUIFileOptions(self)	Initializes the File options and buttons
def GUIVisualOptions(self)	Initializes the Visual options and buttons
def GUIProgramStatus(self)	Indicates the status of the program
def GUISoftwareVersion(self)	Indicates the software version
def GUIMasterGrid(self)	Defines the layout for the grid
def GUISanityCheck(self)	Check the validity of the each settings in the GUI
def GUISelectionToggle(self)	Manage the toggle between the function buttons
def scanPorts(self)	Helps to identify the available ports
def saveProject(self)	Saves the CSV file in the selected path
def readSensor(self)	Main function to read the sensor values
def readserial_Demo(self,path=self.project_path)	Function to read demo values
def readserial_Prod(self,path=self.project_path)	Function to read actual sensor values
def stopSensor(self)	Function that stops reading the sensor values
def prevProjects(self)	Function that opens the saved folder
def projectOutput(self)	Function that opens the path to save CSV file
def toggleOutput(self)	To enable and disable the live output
def startWebserver(self)	To start the webserver at localhost:3152
def stopWebserver(self)	To stop the running web server
def openBrowser(self)	To open web browser in using webserver to view the map

```
if __name__ == "__main__":
try:
    gui = GUI()
    gui.GUIMainloop()
except Exception as e:
    print("Unable to start the app, check files")
    pass
```


KIET HOANG (21256877)
CLARIZA LOOK (22860721)
ARJUN PANICKER (22954837)
DEEPAKRAJ SUGUMARAN(22789952)
HARPER WU (23052765)

EMI-Toolkit Webserver

EMI-TOOLKIT - MAP

Map Page Options

1. FIELD: [Requires Web Server Running and Map File insides maps/ folder] Name of the Maps.
2. OPERATION: [Requires Web Server Running and Map File insides maps/ folder] Plot Tracking data on the map.

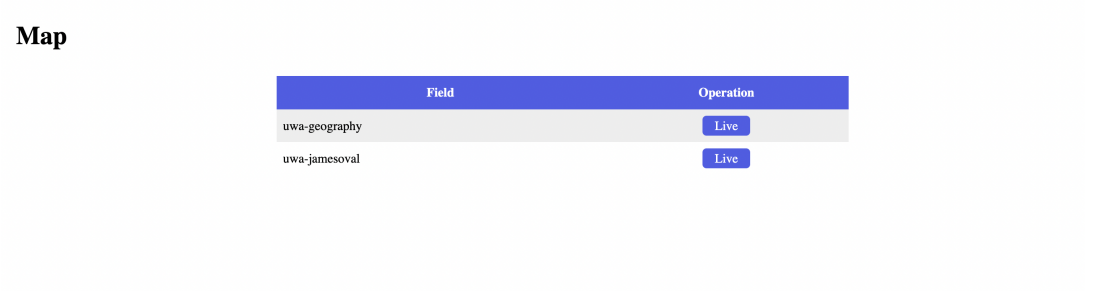

Figure 23: EMI-Toolkit Web server – MAP Page

EMI-TOOLKIT - TRACKING

Tracking Options

1. MAP: Go back to Maps Page
2. STOP: Stop and save tracking information into an image.
3. Current Location: Shows the current position of the GPS

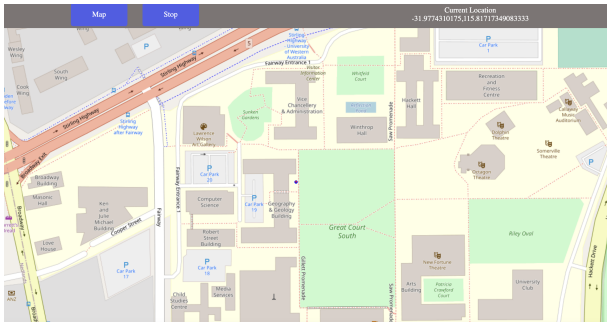

Figure 24: EMI-Toolkit Web server – Tracking Page

KIET HOANG (21256877)
CLARIZA LOOK (22860721)
ARJUN PANICKER (22954837)
DEEPAKRAJ SUGUMARAN(22789952)
HARPER WU (23052765)

SERVER - SOURCE CODE

Please refer to the table below for complete list of Classes and it's methods present in the current version. Note this is a top level view, check GitHub (Click here to Visit) for in-depth review of the code.

Methods	Description
map()	Serves the map.html to list available maps on web server. Provide operation buttons view current live tracking.
tracking()	Serves the tracking.html with the map tiles inside the folder with the name of the variable 'mapName'. Current position is also returned.
getJson()	Loads the local json containing GPS data and returns it to the web browser
getMapNames()	Search dir, use folder name as map name, return folder names in list
draw()	It shows the GPS data as poly line on top of map tiles, current position as circle marker and updates current position text on tracking page.
draw_Interval	Execute draw function in a set time interval
$('#saveImage').click(async function ())	Save the map and other map layers as image.
var map = L.map('map').setView(currPos, 18)	Initialize map instance and set central point and zoom level.
var baseLayer = L.tileLayer().addTo(map)	Initialize map base layer on top of map instance to show map tiles.
var markerLayer = L.circle().addTo(map)	Initialize marker layer on top of map instance to represents current position.
var getJson = async (callback))	Call getJson() function in 'routes.py' to get GPS data.

```
# server.py code
-------------------------------------------------------------------------
from app import app
if __name__ == '__main__':
    app.run(debug=False,host="0.0.0.0", port=3152)

# __init__.py code
-------------------------------------------------------------------------
from flask              import Flask
from flask_jsglue       import JSGlue
app = Flask(__name__,
    static_folder='static',
    template_folder='templates')
jsglue = JSGlue(app)
from app import routes

# routes.py code
-------------------------------------------------------------------------
{EXPANDED ABOVE IN THE TABLE}
 # map.js code
-------------------------------------------------------------------------
{EXPANDED ABOVE IN THE TABLE}
```


KIET HOANG (21256877)
CLARIZA LOOK (22860721)
ARJUN PANICKER (22954837)
DEEPAKRAJ SUGUMARAN(22789952)
HARPER WU (23052765)

Disclaimer

SAFETY

Even though the app is tried and tested, the functioning of dependent sensors with the provided software is still at user's own risk. The team takes no responsibility for damage or injuries it may cause. The software has many inbuilt safety features and this API improves upon it by accommodating errors and will try it's best to not crash unless a fatal error occurs. Still, it is highly important to take the following precautions to ensure your safety and the safety of others around you:

- Always make sure the external battery used with Sensor and Raspberry Pi is in good health and away from direct heat.
- Ensure that you are at a safe distance when running the program during survey.
- Please follow other general safety guidelines.

SOFTWARE QUALITY

This software comes with absolutely no guarantee or warranty. This software is provided to you free of cost. You are allowed to modify this software in any way deemed fit. The methods can be improved easily by editing the source code file. Although the software is capable of impressive feats, it must be noted that it is by no means a comprehensive package for Dualem sensor software. For advanced users, further exploration and development is recommended, and the documentation can be found here:

- DUALEM MANUAL: Click here to read the manual

KIET HOANG (21256877)
CLARIZA LOOK (22860721)
ARJUN PANICKER (22954837)
DEEPAKRAJ SUGUMARAN(22789952)
HARPER WU (23052765)