Jack Carroll (T00194823)

INSTALLATION INSTRUCTIONS – FINAL YEAR PROJECT

Contents

[Raspberry Pi 2](#_Toc41926366)

[NodeRed 5](#_Toc41926367)

[Thingsboard 8](#_Toc41926368)

[App Inventor 13](#_Toc41926369)

[Visual Studio 15](#_Toc41926370)

[SQL Server Management Studio 16](#_Toc41926371)

[Power Bi 17](#_Toc41926372)

# Raspberry Pi

To install the Raspberry Pi system onto the Raspberry Pi 3 device, insert the micro SD card that is compatible with the Raspberry Pi 3 or 4 into an SD Card reader.

Insert the SD Card reader into the SD Card reader slot of any Windows computer.

Download and install the FAT32 format application which can be accessed through this link in the Ridgecrop website <http://www.ridgecrop.demon.co.uk/index.htm?guiformat.htm> for formatting the micro SD Card.

Choose the drive where the SD Card reader is stored, e.g. “E:\”. Choose the allocation unit size for the formatting of the SD Card to occur.

Write the volume label name for the SD Card, e.g. “SD CARD”.

Check the quick format option in the FAT32 format application.

Start the formatting process as seen in Figure 1.1 below.

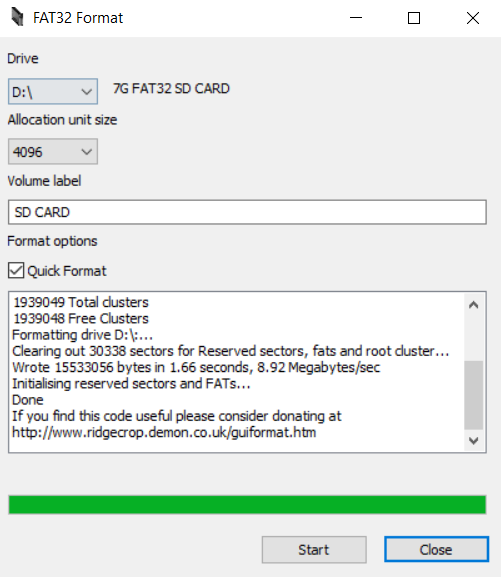


Figure 1.1 – FAT32 Format application tool which converts SD Card to FAT32 file system.

Download SD Card Formatter from the following link right here in the SD Association website <https://www.sdcard.org/downloads/formatter/> to allow the micro SD card to act as the card for the NOOBS files to be stored into that micro SD card.

Open the SD Card formatter and choose the SD Card to format, e.g. “E:\”.

Write the volume label (“SD CARD”) and choose a quick format for formatting the SD Card.

Click format to format the SDHC SD Card into a FAT32 file system for NOOBS files to be stored in that SD Card just like what Figure 1.2 shows.

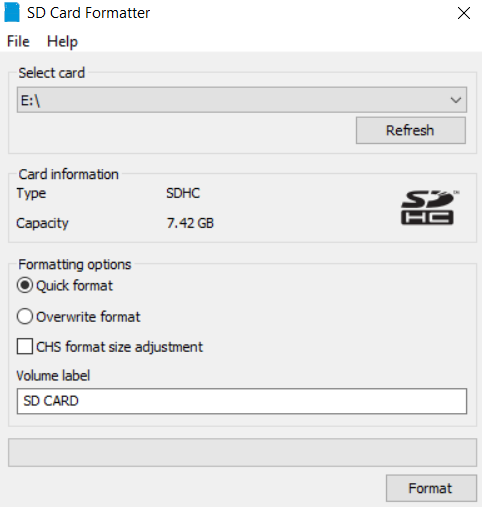


Figure 1.2 – SD Card Formatter where E Drive used for formatting to occur to turn SDHC SD Card file system into FAT32 system which deletes every file in SD Card.

Download the NOOBS Lite package from the Raspberry Pi website in the link that follows <https://www.raspberrypi.org/downloads/noobs/> to use the package for installing the Raspbian operating system (OS) onto the Raspberry Pi 3.

Download the Raspberry Pi imager for Windows from this link <https://www.raspberrypi.org/downloads/> to write the NOOBS image containing the Raspbian OS onto the micro SD Card in the SD Card reader.

Open Raspberry Pi Imager and choose the SD Card, the OS to be installed and for the image to be written to the SD Card just like in Figure 1.3 below.

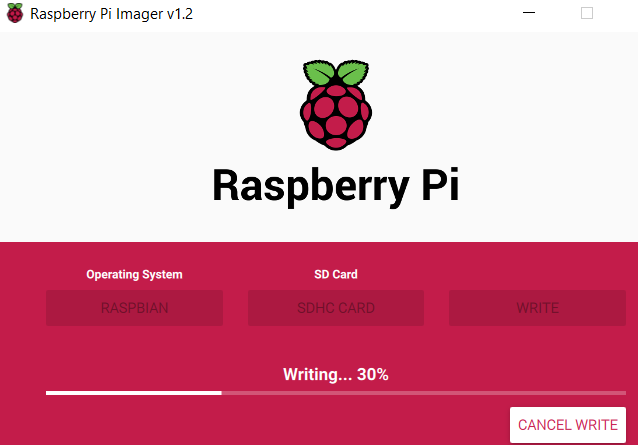


Figure 1.3 – Putting Raspbian Operating System into micro SD Card inserted into an SD Card Reader in the SD Card Reader computer slot.

Extract NOOBS Lite files from the zip folder downloaded form the Raspberry Pi website.

Copy the NOOBS Lite files from the Downloads folder onto the SD Card drive using File Explorer

Remove the SD Card reader from the SD Card reader slot on the Windows laptop.

Remove the micro SD Card from the SD Card reader.

Insert the micro SD Card into the micro SD Card slot of the Pi 3.

Insert the mouse, the power adapter USB cable off an Android phone, a wireless keyboard USB, the HDMI adapter cable, and the USB wireless adapter into the assorted attachments as labelled on the Raspberry Pi 3.

Plug in Raspberry Pi 3 by using the plug attached to the power supply cable that comes from an Android phone.

Switch on the television and switch to the HDMI screen.

Install the Raspbian OS onto the Raspberry Pi in the setup screen by choosing the preferences of how the Raspbian OS should be setup with the applications to be installed as well as the preferences for the accessibility of the Pi 3.

Head into the Preferences options of the Raspberry Pi and activate the SSH, Serial Port, I2P and the SPI interfaces.

Prior to initiating the MQTT messages for the data to be sent to Thingsboard from the NodeRed, install the Mosquitto service by inputting the command “sudo apt install -y mosquito mosquito-clients” for MQTT messages to be sent and published.

## NodeRed

If NodeRed is already installed onto the Raspberry Pi 3 after copying the NOOBS Lite files to the SD Card, there would be no need to install the NodeRed onto the Pi 3 which is the most likely option to occur based on these instructions.

Otherwise, input the “build-essential” package into the command line terminal within the Pi 3 for building modules which are needed by the npm which is “sudo apt install build-essential”.

Use the following command to install the npm, NodeRed and the Node.js packages “bash <(curl -sL https://raw.githubusercontent.com/node-red/linux-installers/master/deb/update-nodejs-and-nodered)”.

After installation, open command terminal prompt window and input “node-red-start” to start the NodeRed service or go into the main menu of the Pi and go to the Progammming applications already installed on Pi 3. Then, click on the Programming menu within the main menu and click on the NodeRed application.

To access the NodeRed application, type in the IP address of the Raspberry Pi along with the port that the NodeRed service is installed in into the search bar of the preferred browser like Google Chrome where a example of this command is this, e.g. “IP Address:Port Where NodeRed is working” to access the NodeRed interface on Chromium service on the Pi 3 or the Google Chrome service on Windows if using headless SSH (Secure Shell). The IP Address to connect to the NodeRed on the web can be seen in the command window prompt when the NodeRed service has launched.

Download the “GPSMiniProject” and “TemperatureRealTime” JSON files from Github through this link <https://github.com/jackcarroll5/Final-Year-Project-Development> in the Raspberry Pi 3 if possible by using the Chromium browser in the Pi 3.

In NodeRed, open the JSON files by pressing Import which is situated in the main menu for NodeRed at the top right corner and open the files mentioned above from the File Manager in the Raspberry Pi 3. Either copy the contents of the JSON file to the clipboard and paste that content copied from the text file or select that JSON file which will setup the layout of the project anyway.

Prior to deploying the projects, as some of the required nodes would not be installed at the start, head to Palettes in the NodeRed menu at the top right corner in the User Settings.

Head to the Install section of the Palette Manager which is beside the Nodes section of the Palette Manager.

Search for the “node-red-contrib-cpu” node for tracking temperature to allow for the temperature tracking to occur for the Temperature Sensor mini-project.

For the MQTT out node for anyone to try it out, another important point is to use the Thingsboard live demo server link which is <https://demo.thingsboard.io> as the server in the MQTT Out node options.

Set the topic in the MQTT Out node options as v1/devices/me/telemetry in both projects mentioned earlier.

In the “GPSMiniProject” NodeRed project to do with the purple MQTT out node shown below in Figure 1.4, change the username options to the access token that is used for the new device established in Thingsboard which can be copied to the clipboard and pasted to the username option in NodeRed in the MQTT out node without the need for a password to see the data being updated every five seconds.

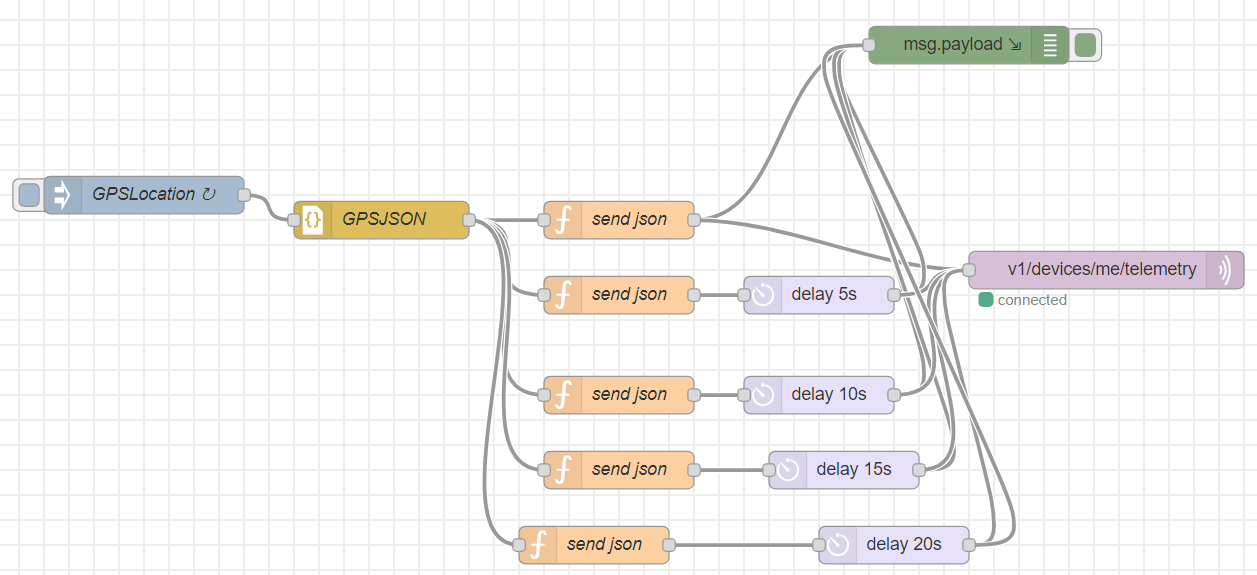


Figure 1.4 – NodeRed project involving mock coordinates data to simulate the monitoring of an asset.

For the “TemperatureRealTime” NodeRed project, change the username for the MQTT out device too to the access token of the new device created for Thingsboard representing the Pi.

Once the projects are deployed, i.e. the temperature NodeRed project, follow this example to see a demonstration of MQTT in action with the subscriber receiving messages from the publisher in Figure 1.5 below by inputting the command seen below this paragraph.



Figure 1.5 – Temperature data payload being sent to Mosquitto broker in Windows laptop from Raspberry Pi 3. Subscribed message sent back to Pi 3 as string.

# Thingsboard

Setup the Thingsboard account by accessing the live demo server version of Thingsboard through this link <https://demo.thingsboard.io/login> to see the updated data heading into Thingsboard.

Setup a device to represent the Raspberry Pi.

Establish a location asset associated with the new device added to the Thingsboard live demo server as seen in Figure 1.6 below.

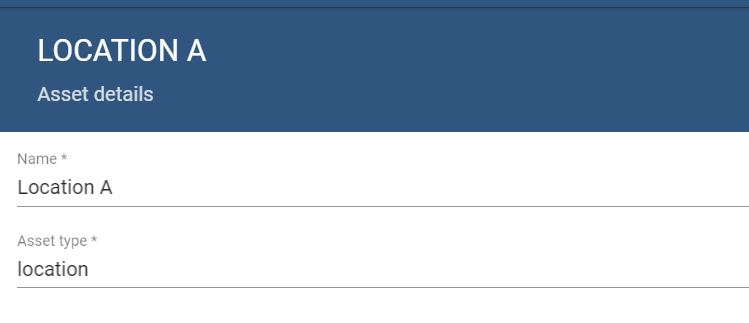


Figure 1.6 – Setting up asset in Thingsboard for new user device.

At the start of deploying the NodeRed project by pressing “Deploy”, a twenty-five second initial setup is done to allow for the tracking of the asset which uses mock data to begin the process.

Meanwhile, for the “TemperatureRealTime” NodeRed project, the same steps can be done for sending the data to Thingsboard to see the debug messages which would be sent to the Pi 3 and updated every couple of seconds as seen in Figure 1.7 below.

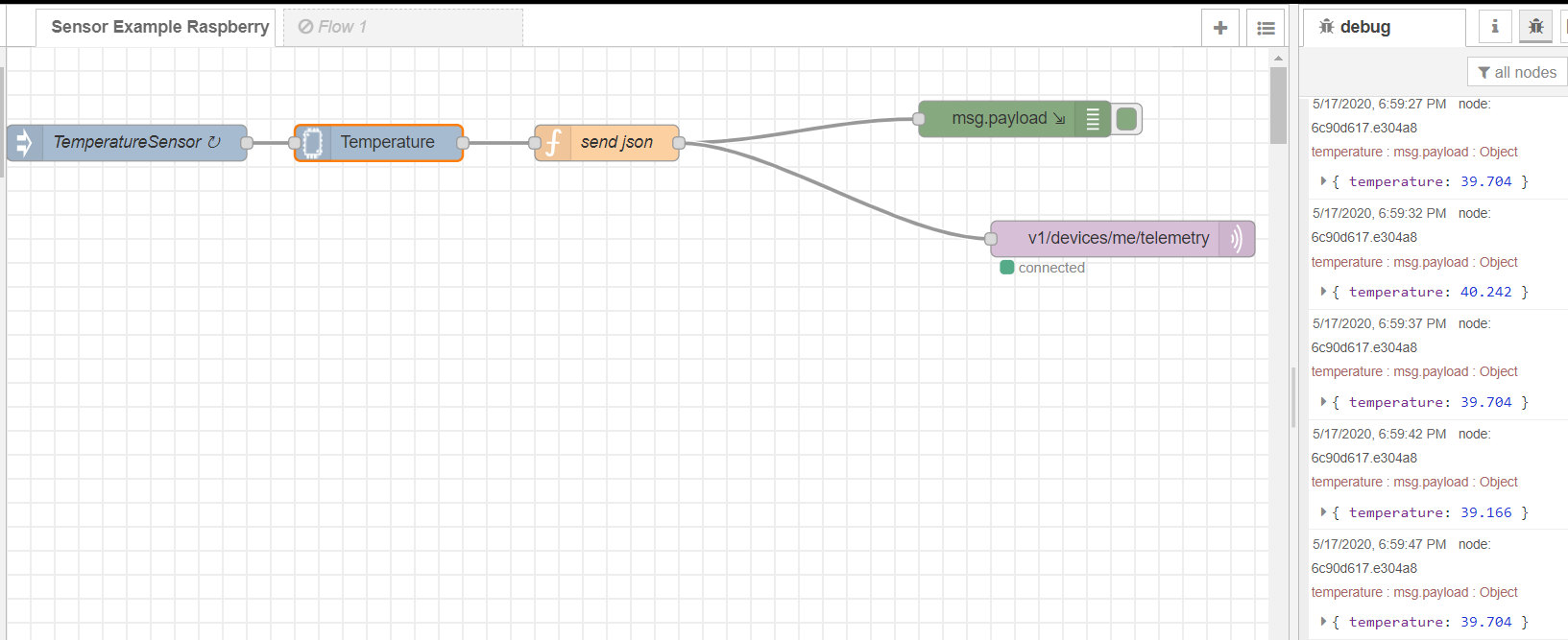


Figure 1.7 – Final NodeRed mini-project to track temperature of Raspberry Pi 3 with debug messages representing the temperature JSON object.

In Thingsboard, while the data is being tracked, head to Devices, select the new device created and head to the Telemetry section to see the data being updated while the MQTT messages are sent to Thingsboard from NodeRed as seen in Figure 1.8 underneath this paargraph.

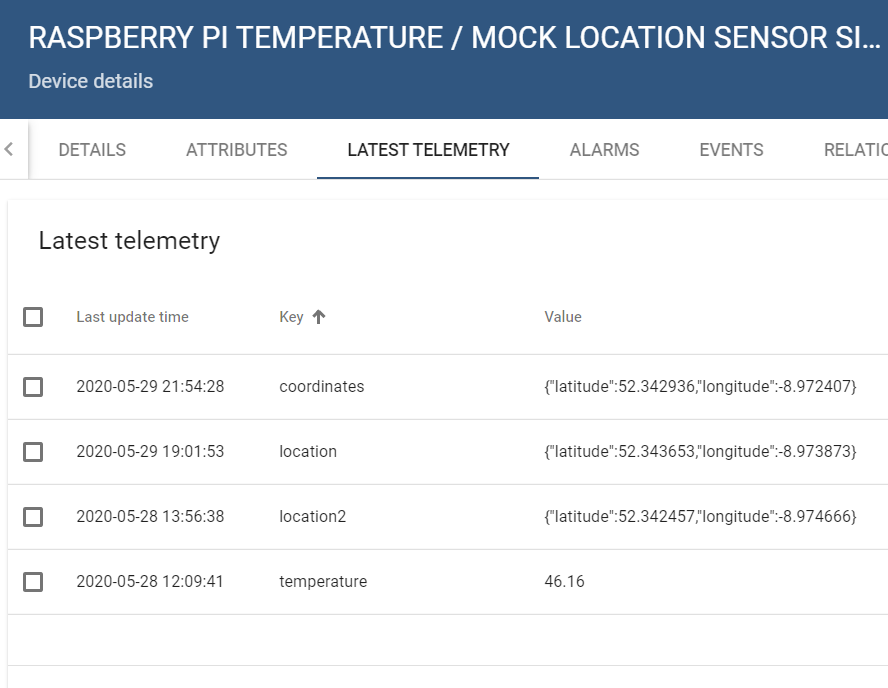


Figure 1.8 – Telemetry section of Thingsboard.

To access the dashboard regarding the tracking of the location, import the asset\_tracker\_dashboard JSON file from the Final Year Project Development folder downloaded from Github by pressing the plus symbol and press “Import Dashboard” as seen in Figure 1.9 below and importing asset\_tracker\_dashboard.

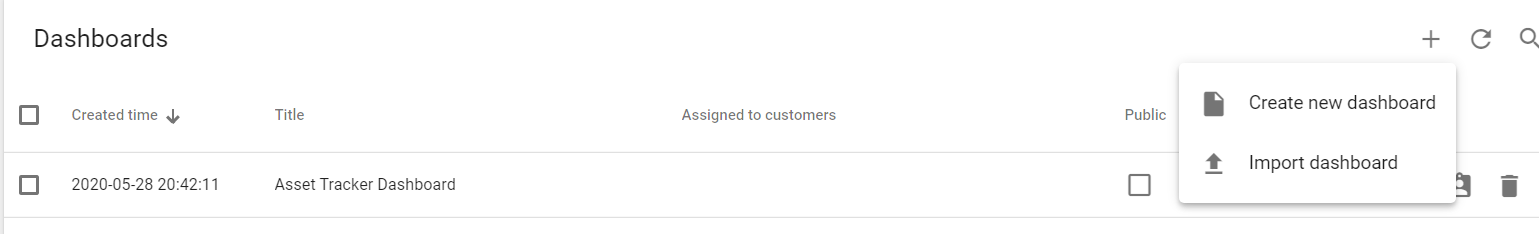


Figure 1.9 – Import Dashboard to Thingsboard.

To access the dashboard for the temperature sensor tracking on the Pi 3, import the “temperature\_raspberry\_pi\_3” JSON object into the dashboard just like what was pointed out in the previous paragraph which is also in the Final Year Project Development folder which is visible in Figure 1.10 below.

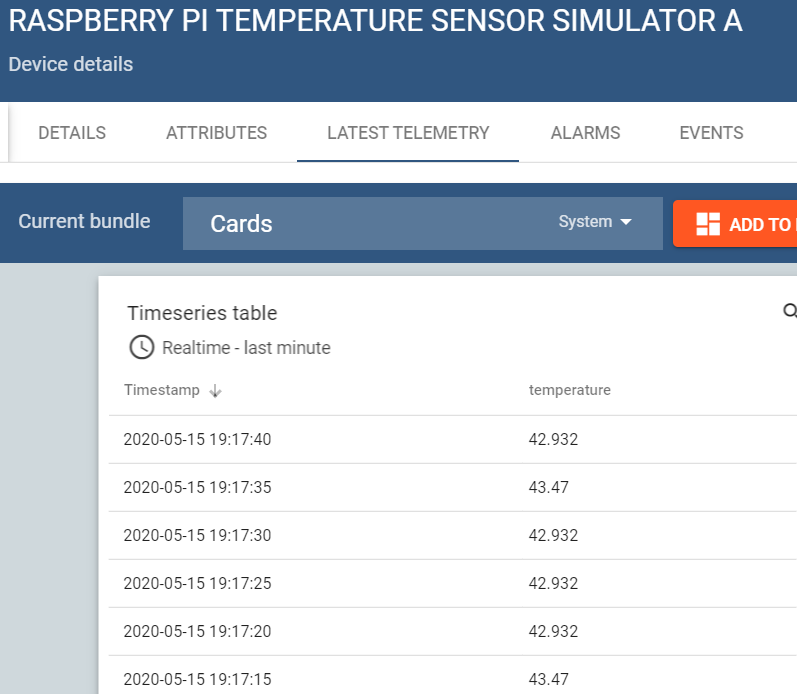


Figure 1.10 – Widget card of latest temperature of Raspberry Pi by using MQTT messages from the Temperature Raspberry Pi 3 dashboard.

Figure 1.11 shows what the asset tracker dashboard will look like once you open it in Thingsboard.

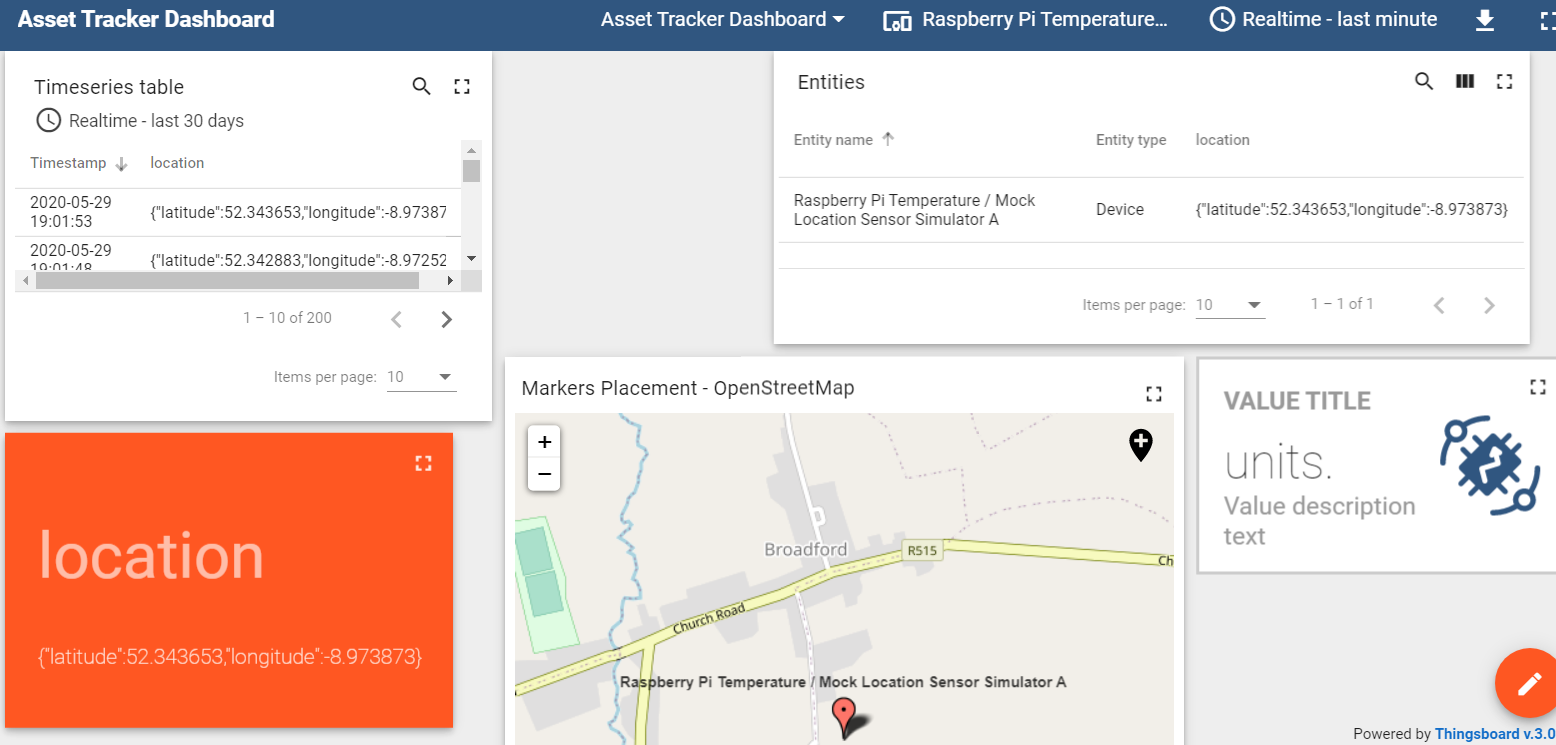


Figure 1.11 – Dashboard of asset tracker detector using mock data in Thingsboard.

# App Inventor

To use the “AssetLocationTrackerSimulation” Android application that was formed in MIT App Inventor, download the Final Year Project Development folder with the link given out earlier in this document from Github.

Copy the .apk file of the “AssetLocationTrackerSimulation” to the Downloads folder of the Android phone by transferring the apk file from the computer to the phone when the USB cable of the phone is attached to the computer to allow the transmission of the file from the USB cable attached to the phone to place into the USB port of the computer.

Find the “AssetLocationTrackerSimulation” application apk file in the Downloads folder of the Android phone and press into it.

The application will be installed onto the phone for usage of this application.

Figure 1.12 shows what the application appears like once the application is installed.

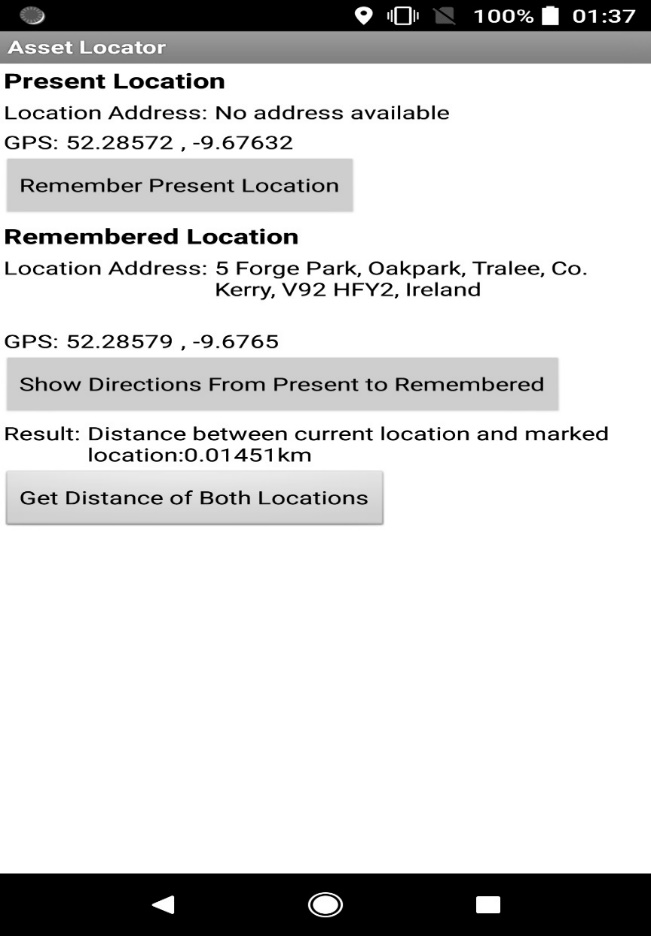


Figure 1.12 – AssetLocationTrackerSimulation application appearance.

# Visual Studio

When using Visual Studio, install the 2019 version of Visual Studio which is known as “Visual Studio 2019”. Install all the ASP.NET features through the Visual Studio Installer. Open the project which says “Geofencing Application” and specifically open the project at the “GeofencingApplicationPrototype.sln” Visual Studio Solution file.

After opening the project, press the green triangular play button to test out the application and it will launch the web application on the preferred browser, e.g. Google Chrome. The dashboard and layout of the home page to sample the home page of the asset web application can be seen below in Figure 1.13 below.

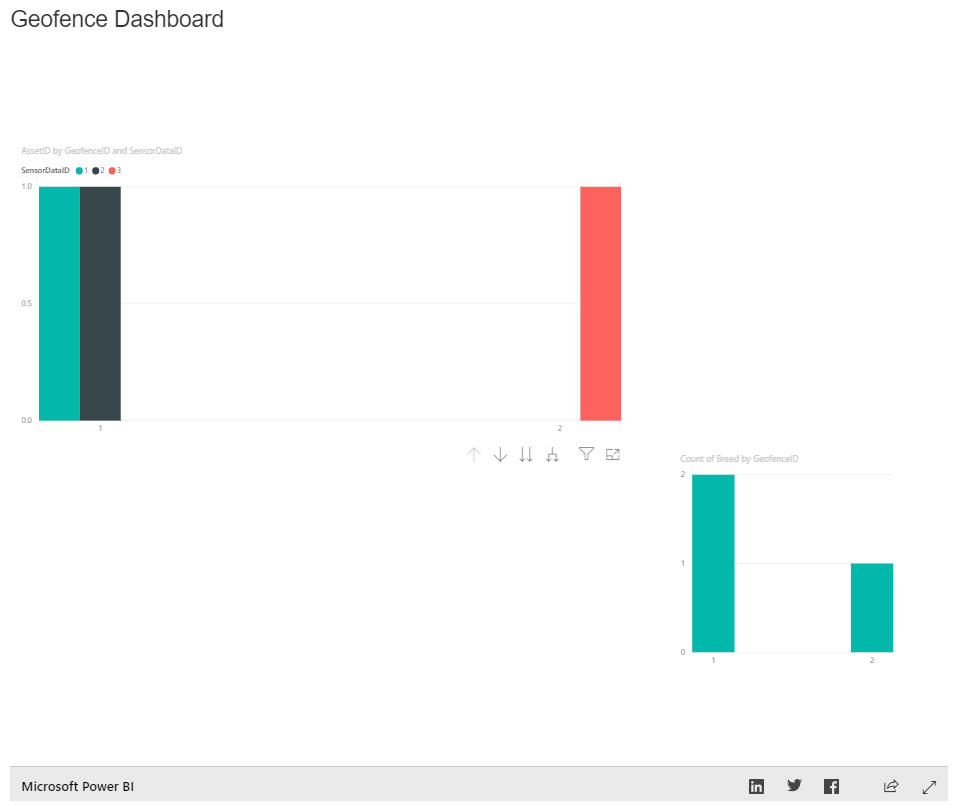


Figure 1.13 – Embedded Power Bi report in the home page of the Geofencing ASP.NET web application underneath the Leaflet Map.

# SQL Server Management Studio

Download the SQL Server Management Studio 2019 from this link <https://docs.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver15> to access the database used during the Final Year Project

To check out the database and the data used within that database, i.e. the sensor data, the geofences and the assets, open the “GeofenceDB” backup SQL file in SQL Server Management Studio.

Restore the GeofenceDB database by right clicking on the Databases folder of the SQL server which is being used in the SQL Server Management Studio version for the user.

In that Databases menu, click on “Restore Database”.

Select the device and click on the box with the three ellipsis (…).

File Explorer opens and locate and select the GeofenceDB.bak file where it is stored on the laptop.

The database automatically restores itself to the data when it was last backed up.

Press OK after choosing the backup file where the database name is automatically chosen.

# Power Bi

Download Microsoft Power Bi from this link <https://powerbi.microsoft.com/en-us/downloads/> to look at visual dashboards based on SQL databases.

Open “Geofence Report.pbix” Power Bi document in Power Bi.

The three database types of SensorData, Asset and Geofence should appear at the right -hand side of the screen.