PROJECT NO: 22R16P00973 SOW FOR C1-C26 **ERDF PROJECT: LCR 4.0**

LCR4.0

European Union
European Regional
Development Fund

Stickx Limited Sensor Project

Created by: Eamon Magdoubi
Prepared by: Magomed Muradov
Approved by: John Marsland

LCR4.0

European Union
European Regional
Development Fund

EXECUTIVE SUMMARY

Introduction

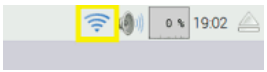
This project is created with the Raspberry Pi 3 B+ and uses a range of SunFounder sensors to measure temperature, humidity, sound, vibration and the tilt of the device. As an IOT device, the machine will connect to a cloud database and output the data every hour for the average or counted values of each sensor.

Instructions

1. For the application, the Raspberry Pi IOT device must be plugged into a micro USB to a power outlet. The power given to the device must be a 5.1V @ 2.5A to ensure that the device can boot. A power supply is provide on behalf of LCR that will have a sufficient amount of voltage to the device.

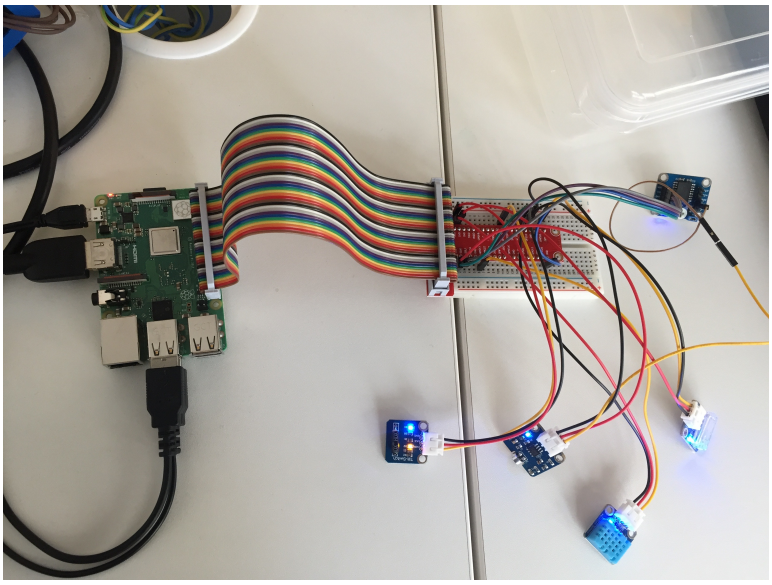
2.A. If the device would be connected via ethernet, plug the cable into the ethernet port onto the pi to give it internet access. This method does not has mobile as connecting to WiFi but saves the required mouse, keyboard and monitor for initial communication.

2.B.Plug in a HDMI cable into the device leading to a monitor that can accept an HDMI (or use a conversion adaptor) cable. Use a USB Keyboard and Mouse which can easily be plugged into the raspberry pi and use any two of the four USB ports. When everything is plugged in, switch the power supply on. The device will boot and after going to the home screen successfully the device is almost ready to deploy data.



2.B.. The Raspberry Pi has Bluetooth 4.1 and 802.11n WiFi and the device must be connected to a local WiFi source to send data. Go to the WiFi Icon and select the WiFi network you wish the Pi to be connected to. Join, save the data and test the connection with the browser. After the

device is connected to the internet, switch the power switch off from the power supply. Detach



the HDMI, Keyboard and Mouse from the device and switch the power back on.

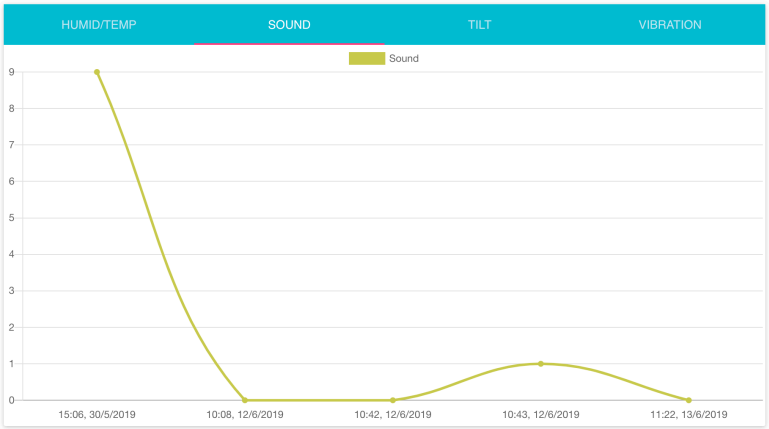
3. Ensure all the sensor lights are lit and the Raspberry Pi power light (the red one) is constantly on while the ACT light is flickering green at a pulse.

DASHBOARD INSTRUCTIONS

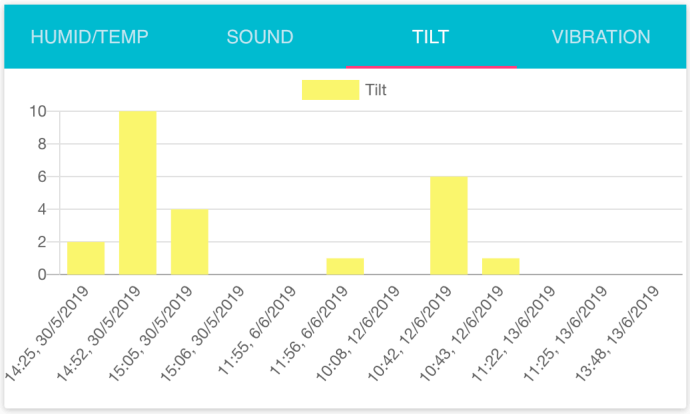
The codebase requires an internet connection to output the data onto the cloud. After one hour the data readings will average on the device and send to the server which can be easily seen in a friendly format for the user.

The dashboard has four tabs and has the data from each sensor on each of the graphs. There are two graph types, a line graph for Temperature, Humidity and Sound, whilst the Tilt and Vibration use

a horizontal bar chart. Each table has its own colour and is dynamically filtered on even if the tab



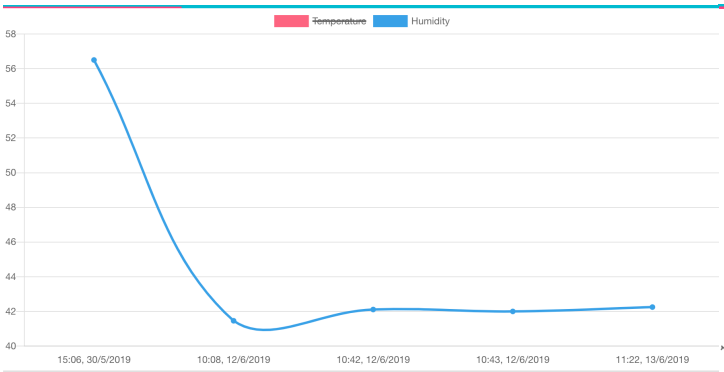
Temperature	Humidity	Sound Triggers	Tilt Triggers	Vibration Triggers	Date Time
23					
23	56.5	9	0	0	15:06, 30/5/2019
23	41.46	0	0	0	10:08, 12/6/2019
23	42.11	0	6	0	10:42, 12/6/2019
23	42	1	1	0	10:43, 12/6/2019
23.75	42.25	0	0	0	11:22, 13/6/2019

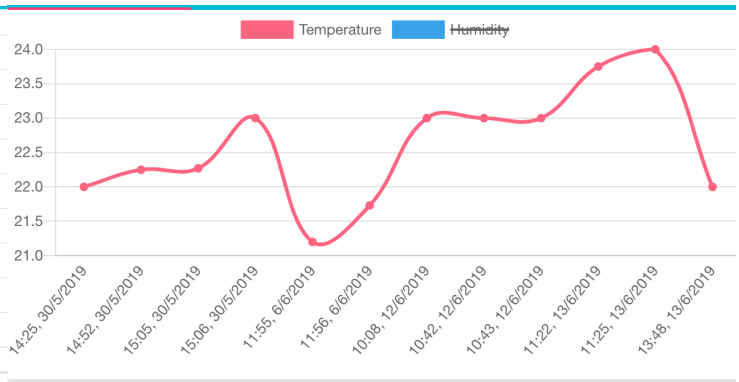


Temperat...	Humidity	Sound Tri...	Tilt Triggers	Vibration ...
22	57	3	2	11
22.25	57.42	2	10	3
22.27	57	11	4	3

isn't selected.

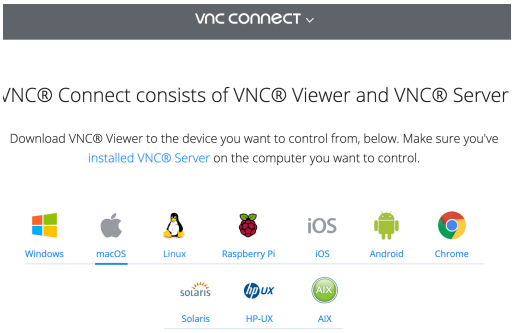
These dashboards are animated and while on the humidity and temperature the keys can be clicked to remove one to only show a single graph at a time. The reason for humidity and temperature to be on the same graph is that they are both recorded using the same sensor so any radical change





should be closely reflected onto both graphs.

Links to Sensor Dashboard:



- https://pi-iot-project-240909.firebaseapp.com/
- https://pi-iot-project-240909.web.app/

MODIFYING DEVICE SETTINGS AFTER WIFI CONNECTION

An alternative way to way to connect to the IOT device is through VNC connect. VNC connect is a virtual screen sharing software which allows the IOT device to be controlled from another computer. Go to the VNC connect website and download the VNC viewer.

VNC Viewer Link: https://www.realvnc.com/en/connect/download/viewer/

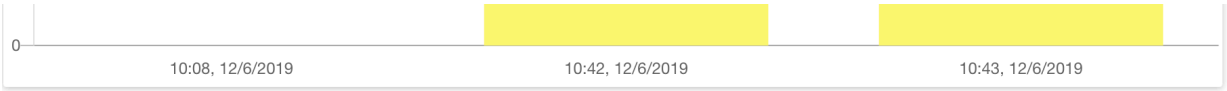
Launch the connector and type the IP address associated with the Pi in the input box, which is **172.20.194.99** and click enter. For the launcher to connect the device must have a connecting to a WiFi source to allow for a connection to establish. After connecting the device can be controlled and edited from the original pc and can be an alternative way to set-up a WiFi connection from an initial ethernet port to not require the extra

pieces of hardware.

Guidance

Due to the potential scale of the data sets the graphs can only show up to 27 points at any given time. The graph can show all the variables in a paginated format. When typing into the filter the graph will filter the top 27 results based on the filter but if filtered down enough will show all variables associated with the data inserted. The data is given over a period of an hour with the date time in

BST. After a piece of data is found the temperature, humidity and sound will sleep for a second before listening again. The tilt and vibration are independent of these triggers as are triggered when the



Temperature	Humidity	Sound Triggers	Tilt Triggers	Vibration Triggers	Date Time
23					10
23	41.46	0	0	0	10:08, 12/6/2019
23	42.11	0	6	0	10:42, 12/6/2019
23	42	1	1	0	10:43, 12/6/2019

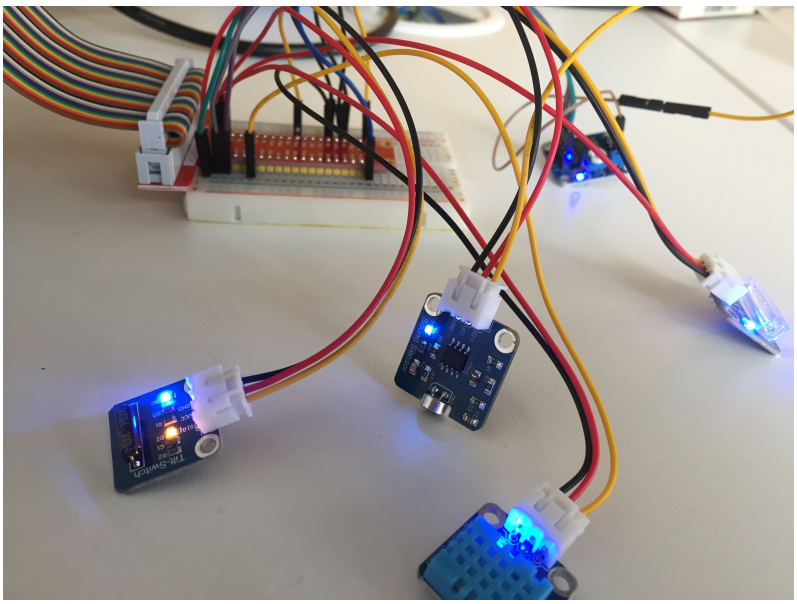
Previous Page 1 of 1 5 rows Next

sensor is active.

Objectives Achieved

A major objective for the project was to find a small factor computer device that could easily connect to the internet and read data coming from various sensors. The set up off the device will need to be simple and effective for all users to be on a simple on and off switch after the WiFi is set-up. GPIO pins communication was achieved as it does not require any additional boards to read or attach onto the Pi which reduces its price significantly. The end goal for the sensor information is to be sent

to a cloud backend and choosing a server-less database was the way forward to make the cost of



production to be as minimal as possible.

Communication to the board

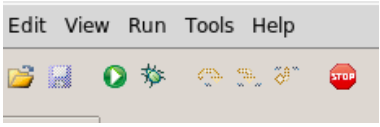
The GPIO import library in Python was the choice of a communication platform for the project with the raspberry pi. This was chosen as the GPIO pins can be moved and others can be added in the future with small hierarchy modifications to the python code base. The init.py file is the initialisation of the entire project and any new sensors must be included in the setup function in that file.

Sensor	GPIO Pin	Measurement
Temperature	11	Degrees Celsius
Humidity	11	Percentage
Sound PCF8591	5	MCU value over 70
Tilt	22	Triggers when tilted
Vibration	40	Triggers when hit

RUNNING THE PROGRAM MANUALLY ON THE RASPBERRY PI

If the project is required to run manually the root LRC Pi Project folder on the desktop folder will provide the code where the **init.py** can be found. This requires a monitor and keyboard/mouse set-up or using VNC server to modify the device.

If on the file explorer locating the **init.py** and double-clicking will show the code in a Thonny IDE. Click the Green run button and the code will run manually to observe any possible errors that could occur. Additional alterations can be made on the pi if you double click the README file which will assist in showing the data on the screen for local viewing.



An alternative to running the code manually is using the LX terminal and type **cd ~/Desktop/LCR Pi Project** then type **sudo python3 init.py**

Back-End Service choices and database restrictions

With the time frame in mind, the most effective route to save the data on a cloud database was a development platform with database and server-side capabilities. This inherit set-up allowed for more development time and allowed a sensor data dashboard to be shown and utilised on as many platforms as possible. Firebase and FireStore, which are cloud SQL-Less databases allow for dynamic fields to be integrated so if other sensors were added an entire overall of the project would not be required. As the scale of the project is revolved around a single sensor data entry a free tier with Firebase fits the requirements perfectly for the stage and web hosting. Storage has a maximum of one Gigabyte but every measure has been made to reduce the amount of data that the IOT device will be sent at any given time. Instead of a constant feed, an hourly measure will be sent from each sensor with the average or total values being displayed. With 50,000 reads per day, it is very hard for the amount of possible daily users to go over the limit within the 24-hour time frame.

Free Tier	Quota for Project
Stored data	1 GiB
Document reads	50,000 per day
Document writes	20,000 per day
Document deletes	20,000 per day
Network egress	10 GiB per month

Key	Dataset
H	Humidity
S	Sound Trigger Count
Te	Temperature Celsius
Ti	Tilt Count
V	Vibration Count

Users can easily be added to the server-side dashboard to view viewing trends and the total amount of traffic received from the dashboard. From this view the NO-SQL database can be seen under Database and then Cloud Firestore. There are two tables one being the keys for each variable in Database keys and sensorData which is the data being sent from the IOT device. The key and values they represent can also be shown with the table below. Keys are used to reduce the amount of data

being send per hour so the IOT device can sent three times the expected amount of data.

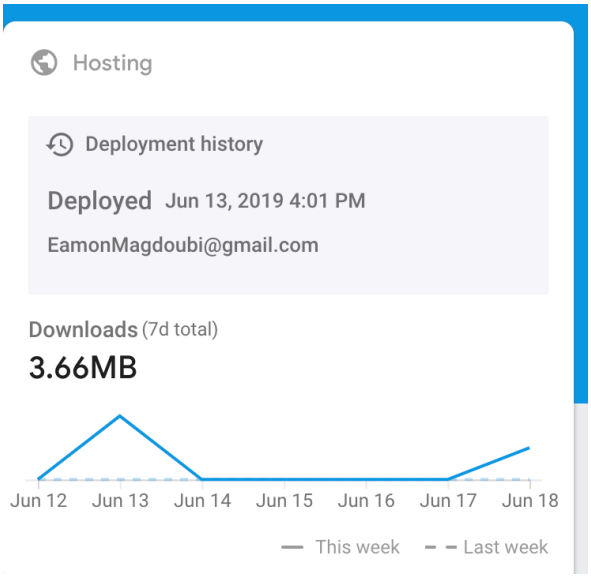
The sensor data has the number variables and the date-time stamp of when the data was sent from the Raspberry Pi. To change the date time format to be on a different time zone it will haft to be manually edited on the pi using VNC or a Monitor Keyboard/Mouse set-up.

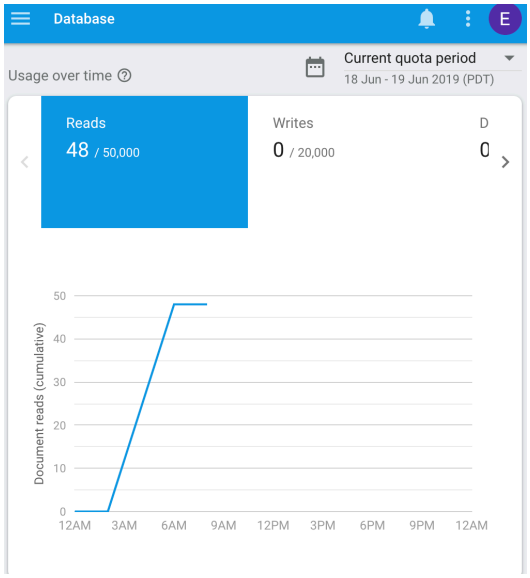
SERVER-SIDE DASHBOARD

As said above there is a server-side dashboard displaying the amount of data that was used by users accessing the web dashboard. Data download and uptime can be monitored to inform users that are trying to access the site to restrict any usage for a certain amount of time. The server-side

functionality can be updated in the future to allow for user authentication so users are not so restricted on the amount of information they are able to see.

The hosted web dashboard is provided through the Firebase hosting service which allows for sites to be hosted through firebase or even a custom





domain if the client has one.

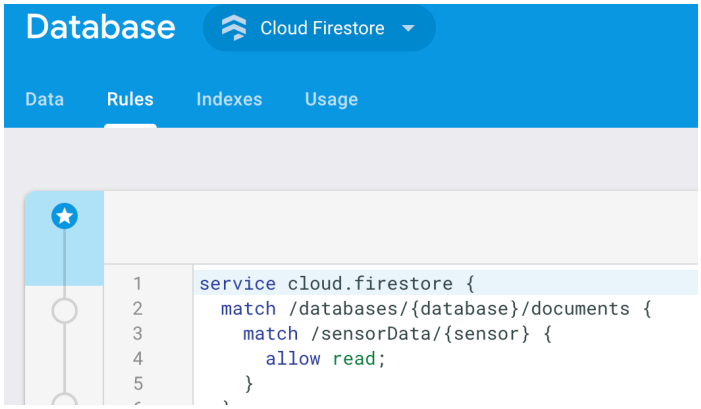
Sensor Dashboard Architecture

The project is built using ReactJS to display and create all the User Interface elements on the web application. When any data is added to the database, the ReactJS application uses Redux to dynamically edit the data on screen so no refresh is required to receive new data. These dynamic values are saved through value changed and subscribing variables, which receives the new datasets

on real-time. If the inherit structure or the database name is altered the Chart.JS and Table plugin must also change to fit the new table rows and information. The animation is provided via the

LCR4.0
European Union
European Regional Development Fund

ChartJs plugin for React and React Tables that allow for the inherently dynamic nature. The React Firebase plugin assists with the database and reading connection.



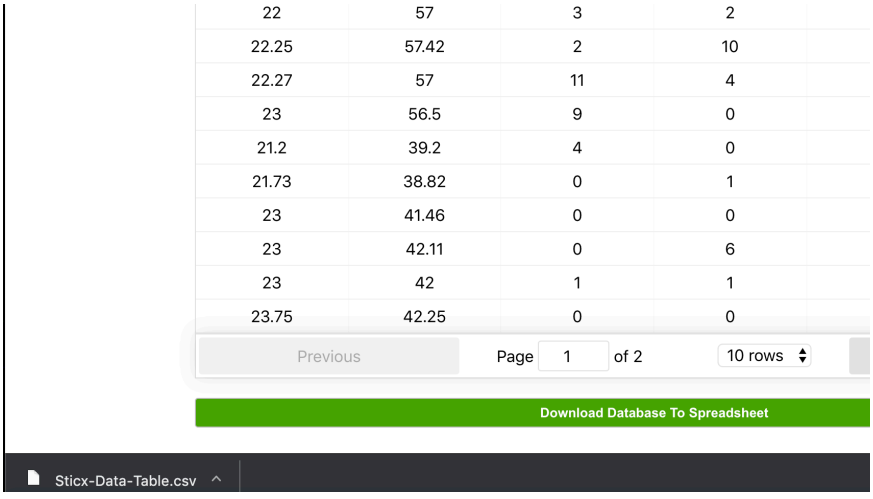
The database is secure only allowing read privileges on the select data set from any web plugins the code can not be altered by another user to delete data from a device.If extra security measures are required on the data they can be done under the rules tab to further enforce extra features that the user will need to fulfil to view the data set.

DOWNLOAD DATABASE TO SPREAD SHEET

At the bottom of the page there is a Download Database To Spreadsheet. This screen button takes the current filtered data that is being displayed to be put into a Comma Separated Values list. This list can be used on spreadsheet softwares such as Numbers or Microsoft Excel. This database can now be used by external users to monitor or add further Business Intelligent searches for extended patterns.

Sticx-data-table

Temperature	Humidity	Sound Triggers	Tilt Triggers	Vibration Triggers	Date Time
22	57	3	2	11	14:25, 30/5/2019
22.25	57.42	2	10	3	14:52, 30/5/2019
22.27	57	11	4	3	15:05, 30/5/2019
23	56.5	9	0	0	15:06, 30/5/2019
21.2	39.2	4	0	1	11:55, 6/6/2019
21.73	38.82	0	1	6	11:56, 6/6/2019
23	41.46	0	0	0	10:08, 12/6/2019
23	42.11	0	6	0	10:42, 12/6/2019
23	42	1	1	0	10:43, 12/6/2019
23.75	42.25	0	0	0	11:22, 13/6/2019
24	42	0	0	0	11:25, 13/6/2019
22	54.33	4	0	0	13:48, 13/6/2019
22.86	40.79	0	1	0	10:31, 20/6/2019
22	66.5	4	0	0	9:09, 25/6/2019



LCR4.0

European Union
European Regional
Development Fund

BACK UP ON PEN-DRIVE

With the dashboard a back-up CSV is saved on the first pen drive that is plugged in. Please only use one pen drive at any given time but any pen-drive that is writable can be used with the writable permissions checked. If there is no plugged in pen-drive the code will run but will not save the database on a hard copy. Each piece of data being saved on the pen drive is also saved on the cloud database.