IoT Application Development CA1 – MyPiControl Functionality Report

Android Application

* Application consists of 7 views; main view allows the user to enter the remaining 6 views.
* Buzzer Control view allows the user to tap on/off button, which will send a command to dweet through volley, the values behind the button are simple true/false. The URL looks like this (“https://dweet.io/dweet/for/mypicontrolboardBuzzer?BuzzerStatus=false&" + "SampleRate=0") When the buzzer is turned on we also send other sensor information (temp, hum, distance, lightlevel) over to dweet, We can change the sample rate that the data gets pushed to dweet by using a seek bar which allows the user to change the sample rate 0 to 30 seconds and then tap a button to update to that sample rate.
* LED Control view enables the user to turn on and off the LED light using buttons, we can also change the brightness of the LED light by using a seek bar, the value can be changed from 0 to 1000 and is updated by a button in the view.
* Dweet Dashboard view allows the user to see sensor data(temp,hum,distance,lightlevel) that’s been pull from dweet using volley, The user gets last values from dweet when button is clicked, the user can get five requests on the screen and on the sixth one the text view gets cleared.
* Last three views are done using Firebase Realtime database, first view allows us to turn on/off the buzzer through firebase, the next view allows us to turn on/off the LED light through Firebase and the last view allows us to view live sensor readings from firebase and change the sample rate of the data being push to Firebase. A seek bar is being used with a button to update the sample rate.

Java Code

* The java code is modified for my dweet things, the code allows us to get the last 5 latest dweets, it also allows us to publisher a value/s to dweet and we can also listen for new values from dweet using the java code.

Python Code

* The project has two python scripts. Once is for dweet controls etc and one is for Firebase controls, I have separated these out as it made it easier to maintain and also makes it a lighter program as you won’t be running both at the same time due it not making much sense to do so.
* The first python script, has functions that gathers sensor data from each sensor and then another function that gets all the functions into one place using dict.
* Then multithreading is being used to provide a listener and a publisher, the listener, listens for certain changes on dweet. For example, the buzzer, we are waiting for dweet to see True under BuzzerStatus. Once it’s true, then the script turns on the buzzer and starts sending sensor data to dweet. We also have sample rate which again is listened by script once the value has been detected a time.sleep is used to change the sample rate. The publisher is the function that starts the buzzer and send sensor data to dweet.
* The same sort of control is provided for the LED lights, the brightness is controlled by a value that the user sends through the Android application to dweet.
* The second python script is for firebase control. The script is very similar to dweets one. It has functions that gathers sensor data, then a function that used dict to put all the values into one method and then we send them to Firebase Realtime database.
* There’re two additional functions that allow us to control the LED and Buzzer, the functions are simple, we create a variable that pulls data from the Firebase database and then assign it to the line of code that control the buzzer or Led, it’s a simple 0 or 1.

Degree of Control over the Raspberry Pi Sensors

* Control sample rate of how quick data from sensors(temp,hum,distance,lightlevel) level is sent, works for both Dweet or Firebase
* Control LED light and buzzer using Dweet or Firebase (on/off)
* Control the brightness of the LED using Dweet.
* Pull information about sensors from Dweet to the Android application and display it to the user
* Pull information about sensors from Firebase to the Android application and display it to the user.

Error Handing

* If internet connection is off volley will give an error telling the user to check their internet connection.
* If the script somehow crashes, and for example the buzzer is left on or the light, running the script again and pressing the off button will allow us to turn the sensor off, before this we had to turn on and the off to be able to turn off the sensors.

Use of different communication mechanism

* Firebase and Dweet were used