

Project Log

Saturday 17th January

I tried desperately to get the TwitterAPI module (<https://github.com/geduldig/TwitterAPI>) to work but could not get it to install correctly on my computer so after much frustration I decided to give up and try again another time.

Saturday 24th January

Today I managed to get the python module installed by adding D:Python34/Scripts to the PATH system variables and running pip install TwitterAPI from the command line.

Then I proceeded to copy this

(https://github.com/geduldig/TwitterAPI/blob/master/examples/stream_tweets.py) example, change the track_term variable to my chosen keywords and then filter them using the if "string" in tweet: function. I then used this to create a count of each emotion based on my keywords and then used the algorithm of (Most counted emotion / Total emotion count) * 255 and all othe. Then set the RGB value corresponding to the emotion to this number. This can then be sent using serial communication to the Arduino on the 3 project days once I have my Arduino kit.

Wednesday 28th January

Testing my Arduino kit with the blink LED program worked fine but when I tried to test the RGB LED kit I bought it would not work, leuan spotted that one of my cables was not in the right place and this fixed the problem. (Example: oomlout.com/RGBA)

Then I extracted some of the example code to make my own program so that I could manipulate the example to do what I wanted it to however there was an error in converting a const byte to a byte for use in a function that I cannot figure out as it works in the example. After explaining the problem to Dr Hyde I found that this was because I didn't copy an essential part of the code to convert the const byte to byte.

I managed to copy and paste the relevant parts of the example code and get it working but ran into another issue when I tried to edit the start and end colours for the fadetocolour function I was getting errors until I realised I needed to define the colours before passing them into the function. Using this I managed to get the Arduino to fade in and out of random colours.

I took a break from coding with Arduino to setup my laptop. I decided to bring my laptop along with me as I needed the TwitterAPI module from python to run my program and I would not be able to install this on the Bath systems. I managed to connect to eduroam wifi using my Bath logon, tested the python program which worked and installed Arduino on my laptop so I could run everything from that computer and benefit from serial connections.

I then installed the pySerial library for python to make serial connections with Arduino and added some serial code to my program. Then I added it to Arduino but stuck on how to format the data from Python and how to use it. I encountered many debugging problem using the serial interface. The serial monitor for Arduino will not load as serial is open in Python so I am not even sure if anything is being written to the serial interface at all! I decided that I was getting too stressed with nothing working and I only have another 30 mins left so I decided to go and look at other's progress to look for inspiration.

Thursday 29th January

When I got in today I started by trying to get the LED to change how I wanted using random numbers as test data instead of trying to get serial connections working straight away however I found a problem with assigning variables that are arrays. After much googling, Raj and Dr Hyde's help I realised my problem could be solved by assigning each item in the variable separately.

Then with help from this (<http://stackoverflow.com/questions/11068450/arduino-c-language-parsing-string-with-delimiter-input-through-serial-interfa>) link I managed to update the program to change to and RGB input on the serial. Then I came across a problem with the LED not fading between colours just changing between them with no smooth fade. I decided to just leave it as changing as it is not noticeable

I then decided to change the algorithm of the light completely so that each of the RGB values are the amount of tweets with the emotion in the last few seconds.

However I realised I had a few problems with my program. Firstly searching for keywords and using this to determine the mood was not the best way of doing things as a tweet containing happy could be saying I am not happy or are you happy now and not actually being happy. The second problem being that every time I added a keyword to the search I was processing less tweets as Python cannot keep up, it is not the most efficient and fast programming language so I decided to fix both problems at once by changing the search to emoticons. This makes it more accurate as emoticons are more likely to show the mood of people and I am only sorting 4 emoticons (2 for angry) meaning that I can process more tweets at a time.

Then I realised there was another problem with my program. Using the tweet numbers as RGB values had 2 problems. Firstly it is not an accurate representation of the tweets and if one of the readings was more than 255 the program would break. To fix this I made an algorithm that works out the proportion of each emotion and sets that to the RGB values. The algorithm is the emotion / Total emotions x 255 to work out the proportion of tweets for each emotion.

Friday 30th January

Overnight I found a white box for everything to fit in and I cut 2 holes in the box. 1 for the power/data cable and another on the top for the LED. Then I borrowed some extension leads so the LED did not need to be plugged directly into the Arduino and could sit on the box. Then I put it together, added some stickers on the box to make it look better and then I was finished!