Thomas Barraclough Internet of Things Project

Github Link: https://github.com/YeahThat/IotProject
YouTube Presentation Link: https://www.youtube.com/watch?v=kMrMV78y2uI

Introduction:

The Internet of Things is an extremely important technology, it allows our devices to send data to each other and from this, it allows us to solve many of life's common problems. In this assignment, I have been tasked with making an Internet of Things device that solves a problem. This paper will talk about the problem I took on; how I created a device to alleviate this problem and how that device works. This paper will also talk about the parts and services I used to create this device; how I tested this device and how I have analysed the data gathered.

For this assignment, I decided to create a project that will help fix a problem experienced by many when it comes to outside lighting. My project is to make a device that detects when an outside light should be turned on and off. It does this by detecting the intensity of the light coming through the window. When it gets dark outside an email will be sent to the user suggesting them to turn the lights on. On the other hand, when it gets bright it will send an email suggesting the user to turn the lights off. The device should be placed on a window ledge for the best possible results. This device can also be used to monitor the light coming in for plant nurseries. This device can fix the problem of having to constantly monitor a plants environment for sunlight. On top of this, I have created an application for my project that allows you to connect to the Arduino through Bluetooth, with more time this would be used as an alternate way to monitor the light level the device is detecting.

Project Requirements:

Arduino Uno Wifi Rev 2:

An Arduino is a programmable circuit board used to make and program electronic devices. This device is praised for being simple to use and for being a great starting point for beginners wanting to make their own electronic devices.

I used the Arduino in my project to create my device by adding various components to it, for example, a Bluetooth component so that I could connect it to the mobile app I made..

Arduino IDE:

The Arduino IDE is a piece of software available to be downloaded for free on any PC. This piece of software allows you to add code to your Arduino using a simplified version of the C++ programming language. From here you can add functionality to your device.

I used this to code my Arduino so that it uses the light intensity gathered and stores it in a variable. I then used the IDE to upload this variable to ThingSpeak through a Wifi connection.

ThingSpeak:

ThingSpeak is a website that can receive data sent from devices. ThingSpeak can analyse this data and form graphs from the data. You can then create events on ThingSpeak that can interact with social media.

I used ThingSpeak to analyse my data, this involved the input of the light level obtained from my device. I then used ThingSpeak to find the average light levels of my data between various time periods. On top of this, I created a trigger that used IFTTT to send off an email if the light level got too low.

IFTTT:

IFTTT (AKA If This Then That) is an online tool that allows you to script actions across various apps. For example, you can make it send a text message to a user when the action is triggered. The action can be triggered from other services such as ThingSpeak.

I implemented this by creating a trigger in ThingSpeak that activates a script in IFTTT that sends an email to the user, this activates when the light intensity level goes under a certain level.

MIT App Inventor:

MIT App Inventor is a simple to use website that allows users to create applications for mobile devices. This service is great for users new to programming as it uses a simple block coding system.

In my project, I used this to create an app that can connect to my Bluetooth HC-06 module and display the connection status.

Room Light Detector App:

The Room Light Detector App is the app I created using MIT App Inventor. It is a simple app that allows the user to connect to the device using the HC-06 Bluetooth module. This application has the potential to be expanded so that it can display the current light level and can allow user interaction.

Smart Life:

Smart Life is a smart device management application that can be downloaded on most mobile phones. From here you can connect smart devices together and set up commands and interactions between those devices.

I used this to connect my Google Home speaker to my smart light bulb and created an interaction where I can give the Google Home speaker a voice command to turn on the light bulb.

Google Home Smart Speaker:

A Google Home Smart Speaker is a device that can interact with the user to do many things, such as searching the web or playing music. This device can also be used to send commands to other smart devices.

For my project, I used this to easily turn on my light bulb through a simple voice command.

Smart Light Bulb:

Smart Light Bulbs are light bulbs that can be controlled and managed through applications. This gives the user the option of being able to control their light bulbs even when they are away from their home.

I used this in combination with my Google Home smart speaker, I was able to send a voice command to my Google Home that then turned on this smart light bulb.

HALJIA HC-06 Wireless Slave Bluetooth Module:

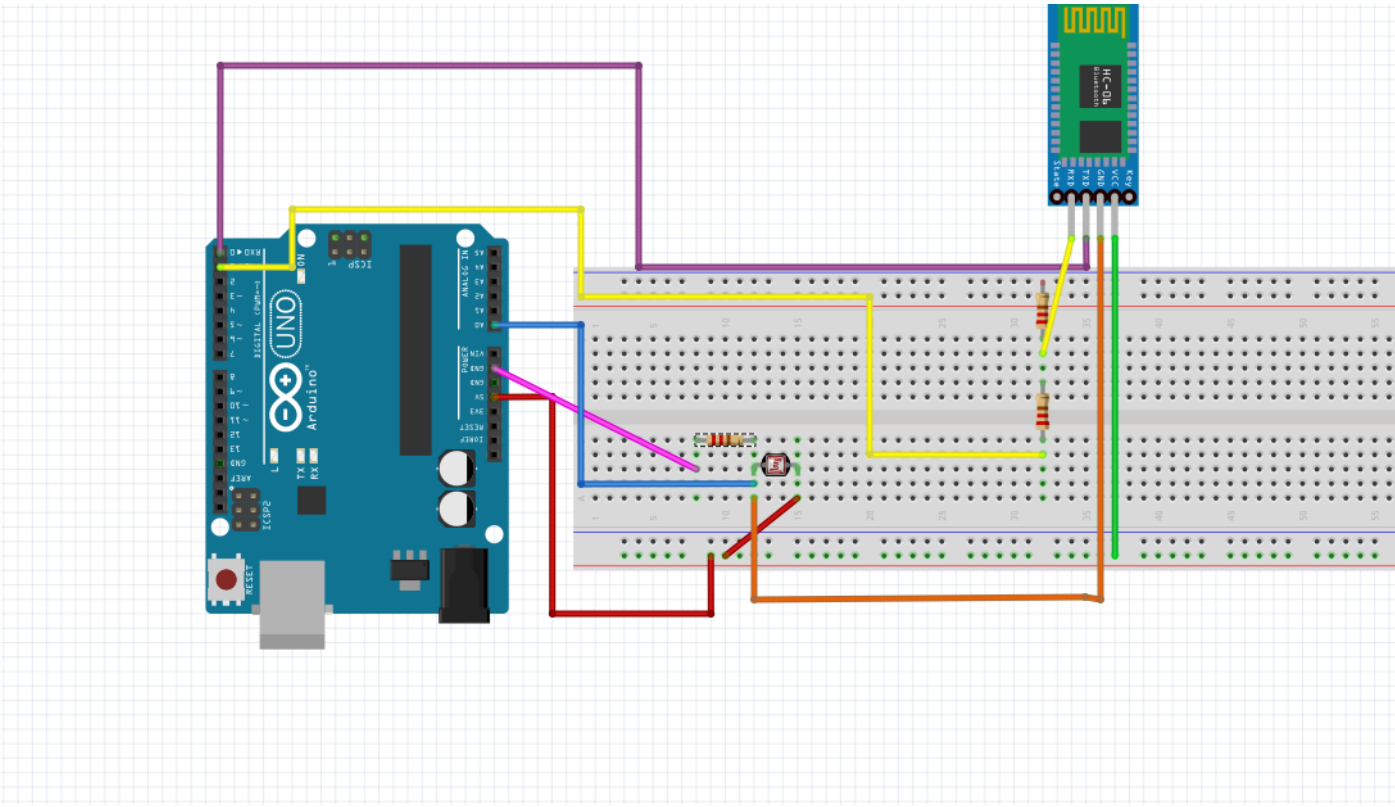
The HC-06 Bluetooth module is a slave Bluetooth module that allows for wireless communication between devices.

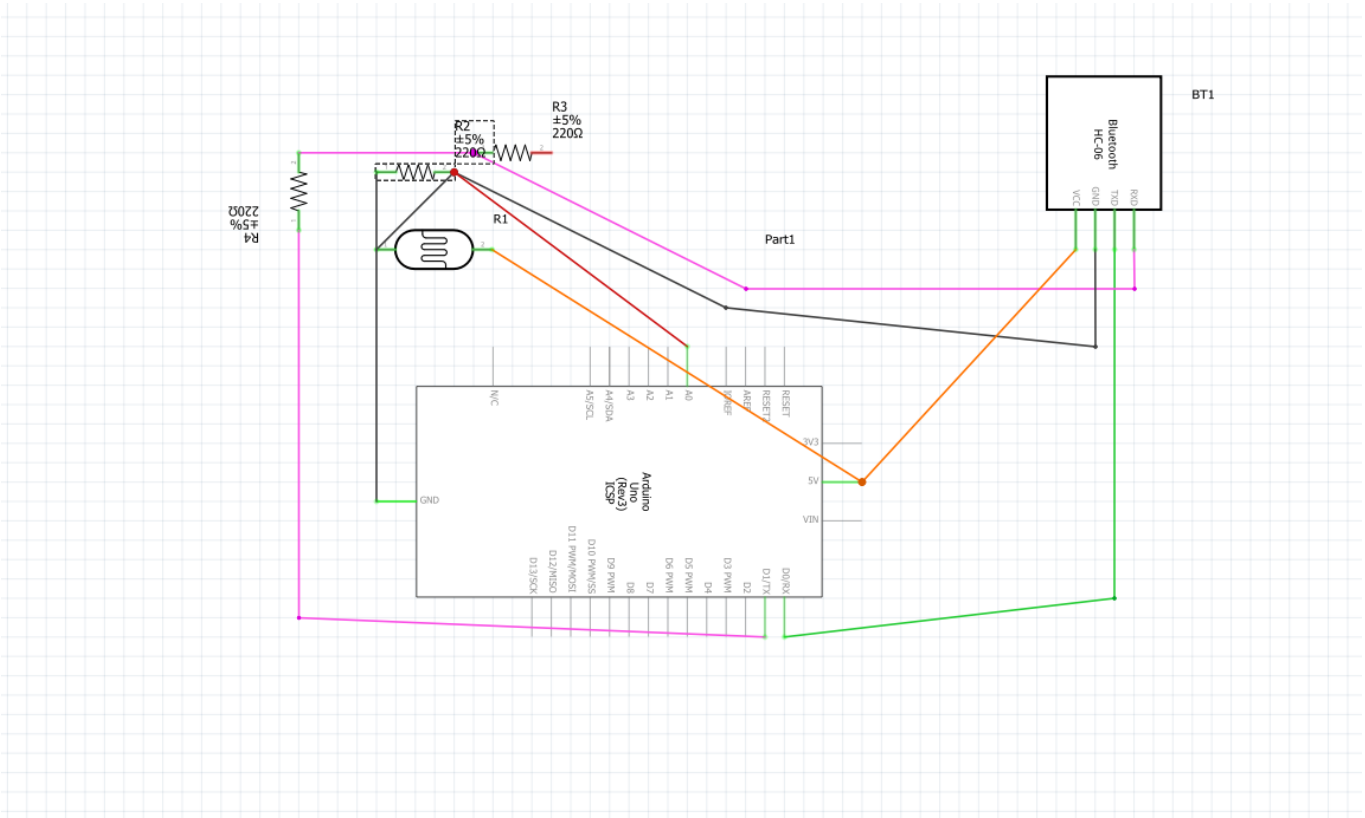
In my project, I used this to connect my device to my mobile phone using an application I created.

Circuit Design:

The circuit for my device had three main components. One was the Arduino Uno Wifi Rev 2 itself, this is where I uploaded my code from the Arduino IDE. Connected to the Arduino on the breadboard was a photocell. The photocell collected data on the light level of the room it was placed in. The last major component was the HC-06 Bluetooth module. This Bluetooth module allowed my device to interact with a mobile phone application I created.

Below here there are two schematic images showing the design of my circuit.





As you can see from my schematics I sent 5 volts to the breadboard from the Arduino. I used this to power up the photocell. After powering the photocell I sent the data it gathered to the A0 pin on the Arduino. The power was then grounded. This setup overall allowed me to

gather the light intensity from the room and send it back to the A0 pin so that it could be used later on when uploading to ThingSpeak.

As well as this, the HC-06 Bluetooth module was connected to the breadboard. Coming from the VCC pin on the HC-06 module I connected a wire to the positive bus that was being powered from the Arduinos 5 volt pin. This allowed me to turn on the HC-06 Bluetooth module.

The HC-06 Bluetooth module also needed to be grounded, therefore I connected the GND pin to the same lane as the wire connecting to the ground pin of the Arduino. This setup grounded the Bluetooth module.

Next up were the TXD and RXD pins on the HC-06 Bluetooth module. Overall the HC-06's TXD pin needed to be connected to the RX pin on the Arduino and the RXD pin needed to be connected to the TX pin on the Arduino. For the TXD pin, this was a simple procedure and it could be connected straight to the RX pin via a wire. However, for the RXD pin, there were a few extra steps. The TX pin that the RXD pin needed to be connected to had a maximum voltage count of 3.3 volts. However, the RXD pin was being supplied with 5 volts. Therefore I had to lower the voltage count of the RXD pin before it reached the Arduinos TX pin. To do this I connected a wire from the RXD pin to the breadboard, from here I connected two resistors of the same value, one of which led to the minus bus and the other led to a wire that connected to the TX pin on the Arduino. With this setup, I halved the voltage count leaving it at 2.5 volts, which was enough to power the TX pin. Overall this whole setup completed the HC-06 module and allowed it to send and receive data.

Project Testing Phase:

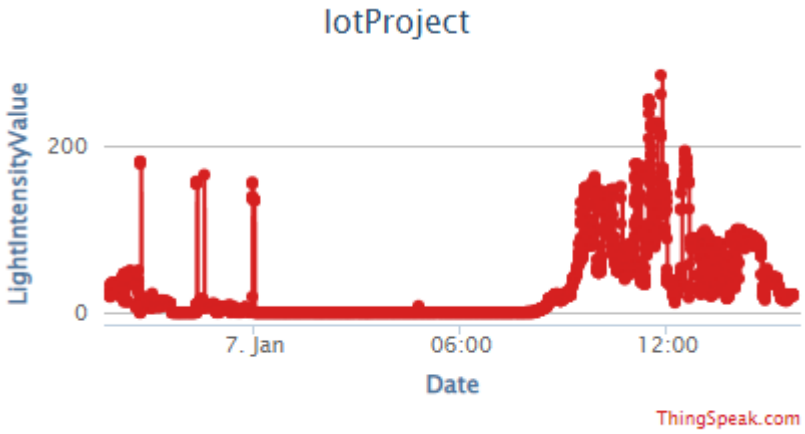
Throughout the creation process of this device, I ran many experiments to make sure each aspect of the device is functioning correctly and accurately.

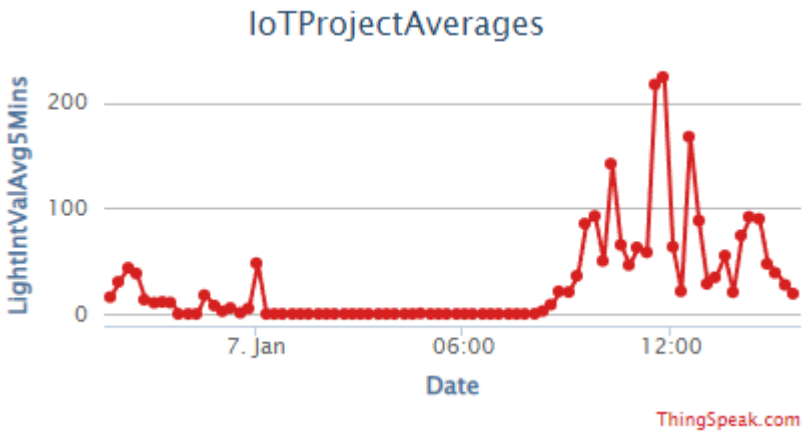
One of the first tests I ran was making sure that the light sensor was accurately gathering data. To test this I used the Arduino IDE. On the Arduino IDE I created some simple code to print out the light intensity value. I ran this code under different scenarios to find if it was correctly picking up the light intensity of the room. These tests involved covering the device with a box to see if the light intensity was drastically lowered and shining a torch in front of the device to see if the light intensity was drastically heightened.

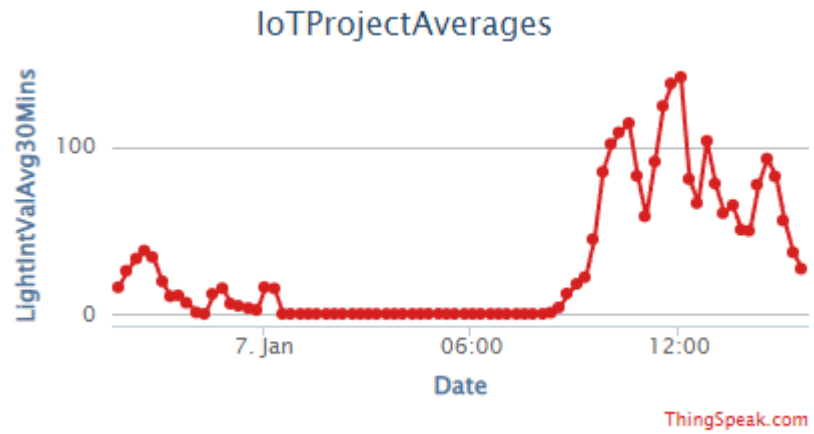
After finding that the light intensity was accurately being displayed on the IDE, I tested to see if the data being pushed to ThingSpeak was the same as the data inside of the Arduino IDE serial terminal. From this, I created another ThingSpeak channel that analyses this data and finds the average light intensity for a multitude of periods of time.

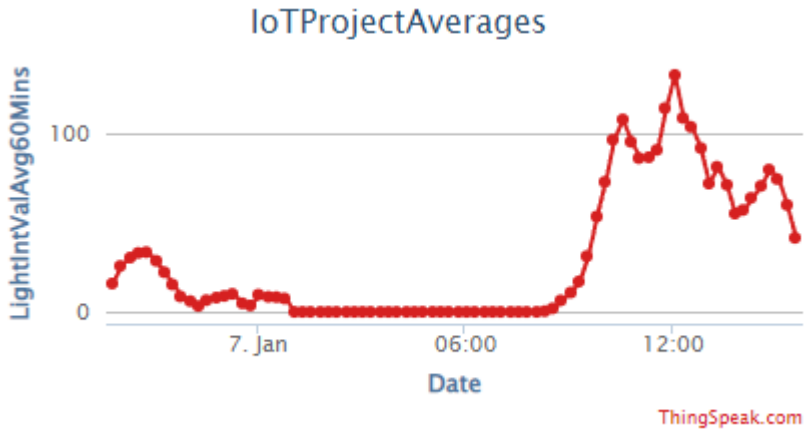
I later created an IFTTT action that triggers when the light intensity goes under a certain level. I tested this by covering the light sensor and found that it consistently worked. However, after a few tests, I found that this can be triggered by any brief anomaly in the light intensity. An example of this would have been if the light intensity was consistently at a level of 50 when the device is situated on a window ledge, but then for a brief second a cloud appears and for a few seconds it is darker than normal. This would have set off the IFTTT action because for a few brief seconds it went under the light threshold needed to trigger the event. To patch these inconsistencies I used the average light level within the past 5 minutes instead of the base numbers constantly entering from the device. Therefore this rounded out all of the inconsistencies and made the IFTTT action only trigger when the light levels are consistently low.

You can see this down below by comparing the graphs provided.







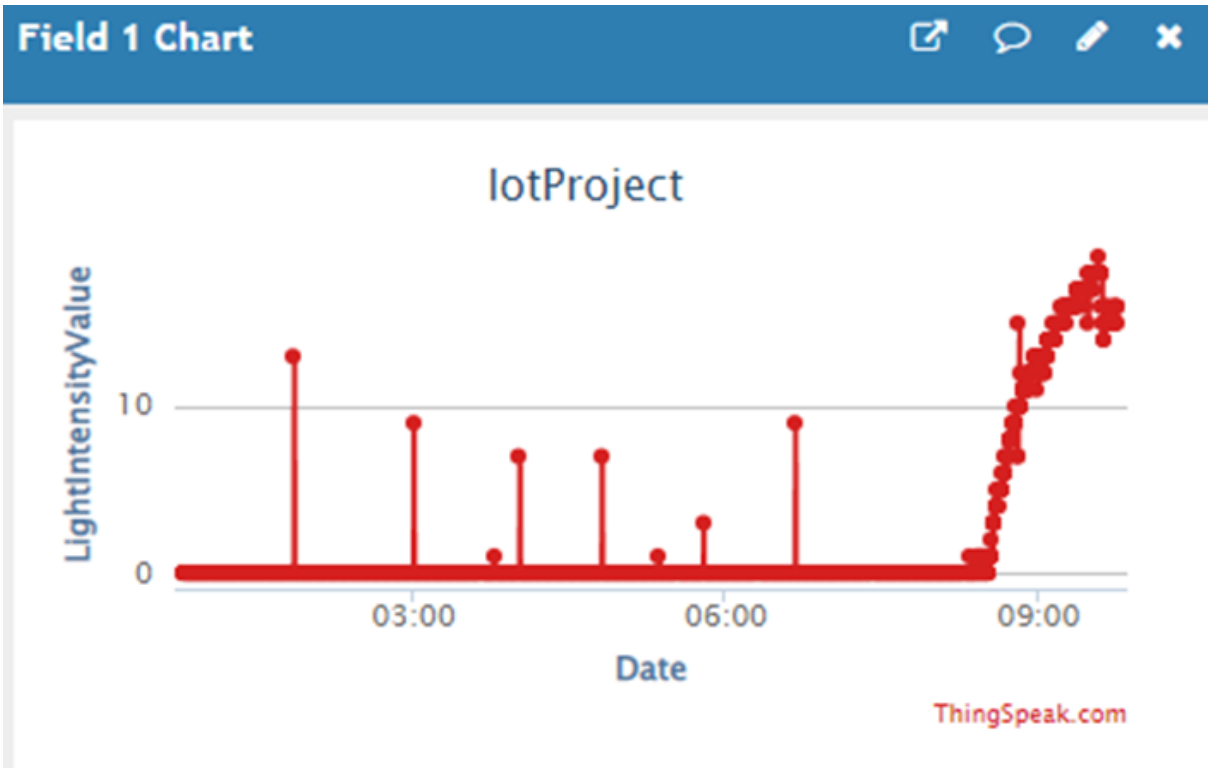


As you can see from the above graphs when the time period the averages were taken from is larger, the graph then spikes less when an inconsistent value appears.

The only other area of my project that needed testing was the Bluetooth connectivity between my phone and the Arduino. Testing this was simple, this is because when the HC-06 Bluetooth module is connected to a device a light situated on the HC-06 shine static red, however, when not connected it will flash a red light. Therefore when testing this I just made sure that when I connected the HC-06 stopped flashing and instead presented a static red light.

Data Analytics

During the development process, I was constantly running my device so that data would be uploaded to ThingSpeak. On ThingSpeak I had made four different graphs that display the data gathered in different ways. I could use the data on these graphs to measure many different things about the lighting in the room the device was situated in.



The graph above shows the raw data gathered from final test which took place from 00:00am to 9:30am. During this test the device was placed on a window ledge inside of my bedroom. From the graph we can see there are 8 inconsistent data points. During the night I turned on the light once, this was at 2:40am, this explains the first anomaly.

To find out how the other anomalies were made, I had to have a think about the different ways my window ledge can inconsistently be lit up by light. The two ways I analysed this could have happened are from the motion detected light outside my room and from headlights on the street. I can analyse that the larger inconsistencies are from the outside light, mean while the smaller ones are potentially from head lights.

We can also analyse from this graph that, it got light in my room just before nine. This makes sense on my window ledge as it is not facing the sun.

Legal and Ethical Evaluation:

This project does not break and legal laws or raise any ethical issues. The device does not take in any information from the user other than there room lighting level. No personal data is gathered and the device cannot be used in any way that would cause harm to others or the user.

On top of this the apps being used (My App and Smart Life) are also ethical. My app gathers no user data and is not a risk to anyone. Smart Life can access any device you link to it however it does not use them nor collect data from them in any ethically challenging way. Neither of these apps break any laws.

Conclusion:

In conclusion this app can be used as a simple solution for not having to be monitoring your outside lighting and instead be automatically reminded. This device can take away a lot of stress from people who want to have outside lighting but want to only have it turned on when necessary. It also offers an easy way to turn on the outside lights without physically have to flick the switch.

There are a few changes I would have loved to have made with more time, for example the app I created was originally meant to show the light level the device was picking up. However due to problems between the Arduino type and classic Bluetooth this had to be changed. Therefore if I was to do this again I would use an Arduino that supported classic Bluetooth.

Overall I am happy with the project and think it could be useful for a lot of people. I believe this successfully helped solve the problem I was tackling and made monitoring your outside lighting a lot more simple.

References:

http://iot.appinventor.mit.edu/assets/tutorials/MIT_App_Inventor_Basic_Connection.pdf

This document helped me set up my app and show me some of the basics of the app inventor website. This was the first time I had used app inventor so this was extremely helpful.

https://create.arduino.cc/projecthub/RucksikaaR/interfacing-the-hc-06-bluetooth-module-with-arduino-f9c315
This document was great for learning how to setup the HC-06 Bluetooth module with the Arduino.

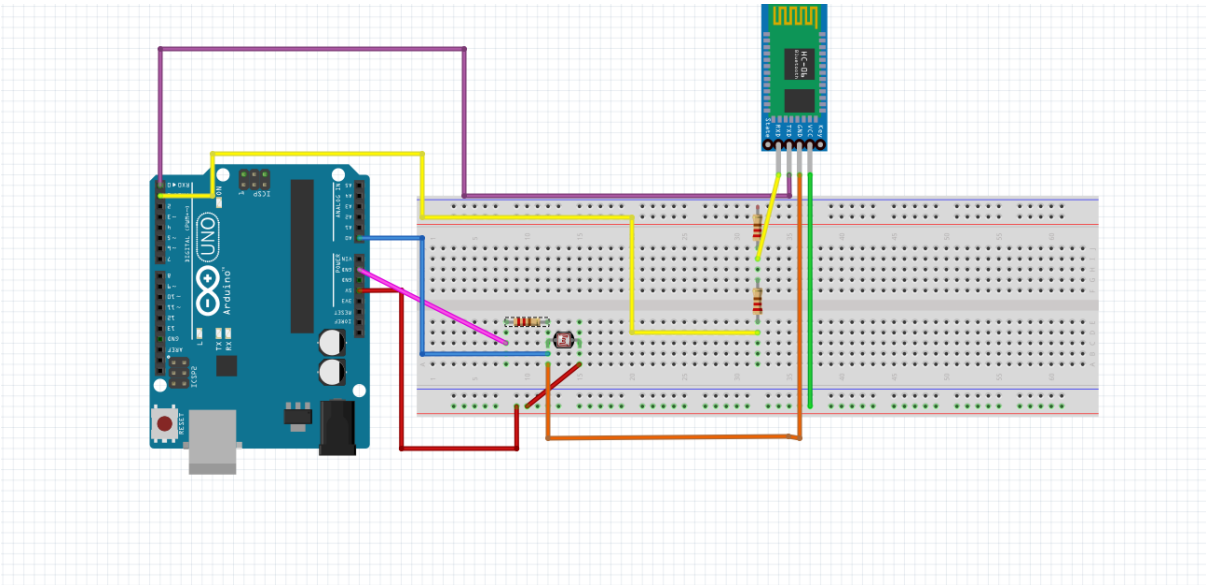
https://lifehacker.com/the-beginners-guide-to-ifttt-1819624556
This website taught me what things I could do with IFTTT and helped me decide on the way I wanted to use the website.

https://www.instructables.com/ThingSpeak-IFTTT-Temp-and-Humidity-Sensor-and-Goog/
A great document that helped me understand how to create actions that set off IFTTT events.

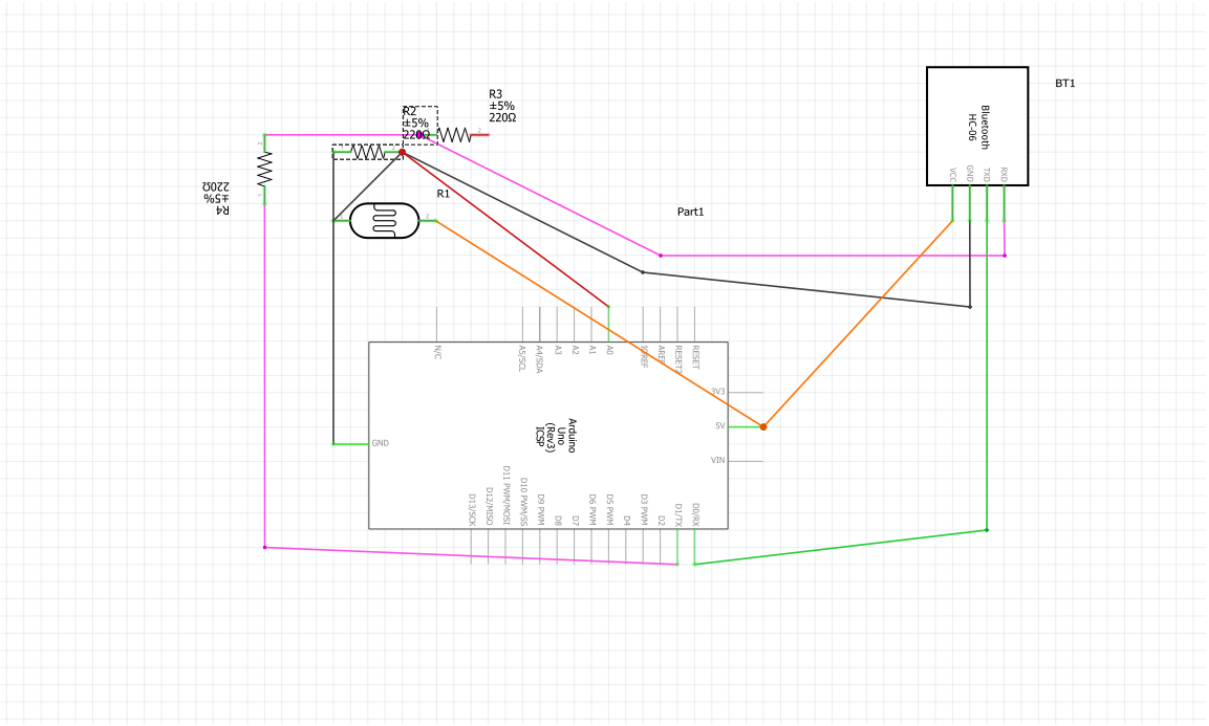
http://ai2.appinventor.mit.edu/reference/blocks/
The official documentation for the app inventor website. This website had information on all the code blocks provided which was extremely useful for setting the apps code up.

Appendix:

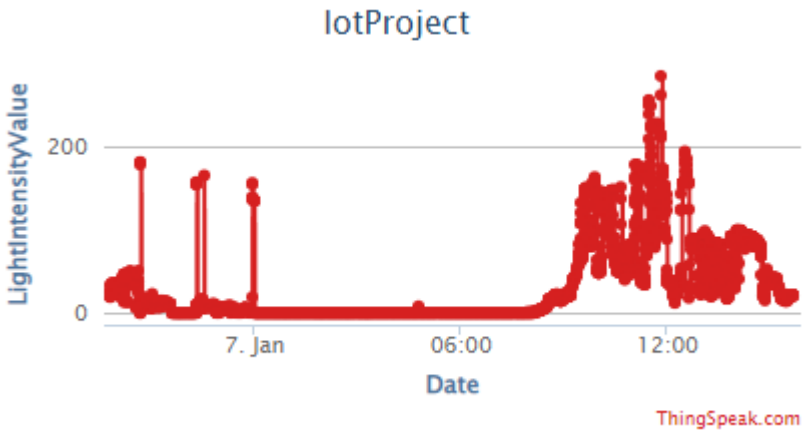
Circuit design 1:



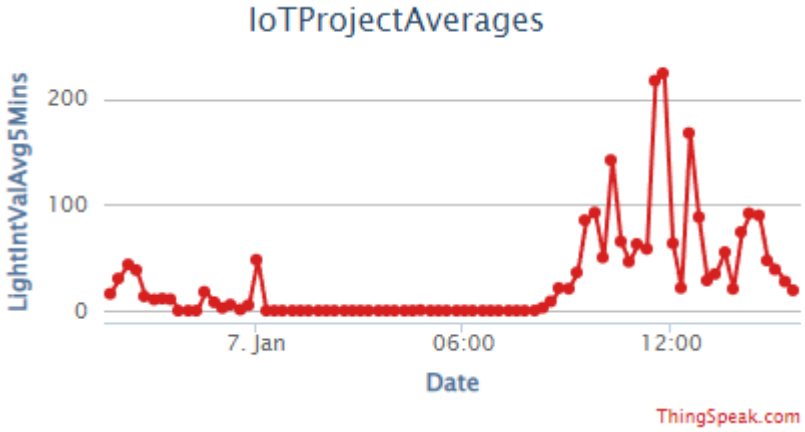
Circuit design 2:



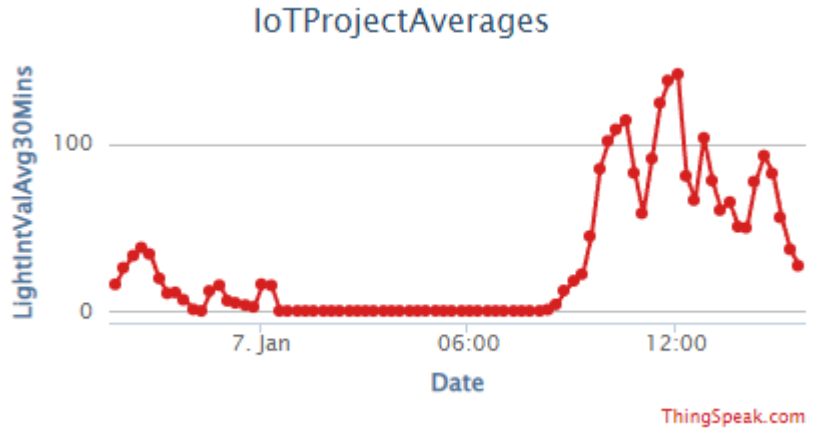
Raw Data Graph:



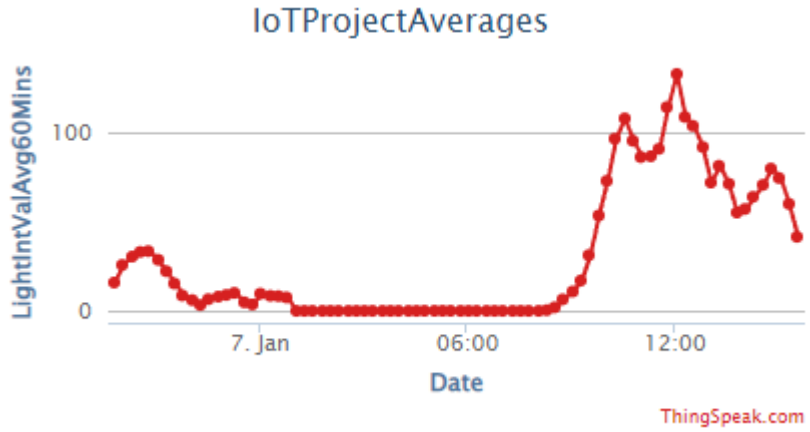
Average data over last 5 minutes graph:



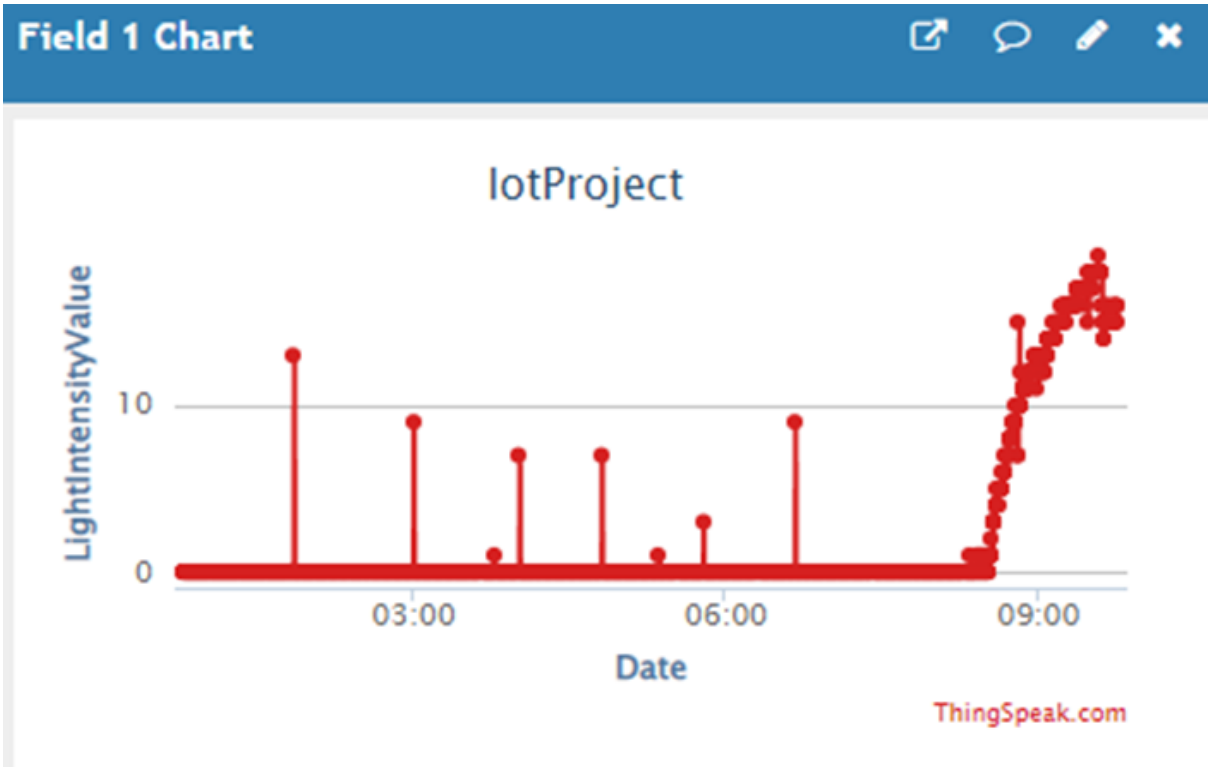
Average data over last 30 minutes graph:



Average data over last 60 minutes graph:



Raw data chart 2:


Field 1 Chart
IotProject
LightIntensityValue
10
0
03:00
06:00
09:00
Date
ThingSpeak.com