

QUANTUM LIGHT

*Sisinty Sasmita patra,
Smita Kundargi,
Jeanne Victor*

PROJECT GOAL:

Quantifying the interesting data and displaying various LED patterns based on quantification. For this project, we focus on quantifying weather data. Our main goal of the project is to design a LED pattern in such a way that it gives the insight to weather at a particular location.

RESEARCH QUESTIONS:

1. How to get the weather data and classify it. How to make it working using Arduino?
2. How to provide an input to LED to display differently for different weather condition?
3. How to seamlessly integrate multiple devices?
4. How to analyze the data for a given period?

Quantum Light Schematic:

Quantum Light allows the user to choose the type of data; one is interested for e.g. (sensor or online weather data). Then it allows getting the data from the previously mentioned choice then classifies the data and displays LED accordingly.

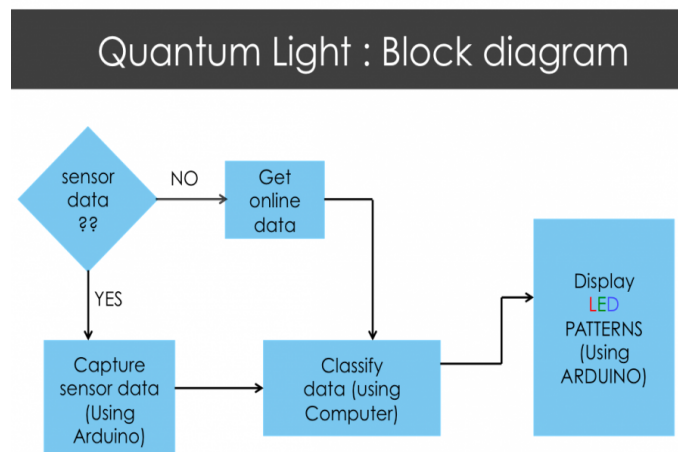


Figure 1: Represents the block diagram of Quantum Light

Components Required:

1. Arduino board
2. NeoPixel digital RGB LED Strip
<http://www.adafruit.com/products/1138>
3. Temperature Sensors (DS18B20)
<http://www.adafruit.com/products/374>
4. Resistors
5. External power supply
6. Jumper wires

DATA CLASSIFICATION:

We are classifying temperature and wind speed data into three categories. Temperature as (hot, normal, cold) and wind as (no windy, less windy, more windy). Below is the table that describes classification in detail. We have used zero('0'), one('1'), two('2') to classify each category of weather data. We have also used '0' for hot, '1' for normal and '2' for cold temperatures respectively. Similarly assigned '0', '1', and '2' for no windy, less windy and windy.

Classification Details		
WEATHER DATA	CLASSIFICATION	RANGE
TEMPERATURE	0	<=10
	1	> 10 and <= 20
	2	20 and above
WIND SPEED	0	<= 5
	1	>5 and <= 10
	2	10 and above
HUMIDITY	0	<= 50
	1	> 50 and <= 75
	2	75 and above

Figure 2: represents the weather data and its classification.

Assigning LED STRIP patterns:

For each combination of temperature and wind speed data, we assigned a particular LED pattern to display. We are using Neo pixel RGB led strip to display the LEDs. It consists of 60 consecutive LEDs in a strip. We also considered that, if the temperature is cold our LED should display blue in color, if hot; it should show red in color and if temperature is normal, to display the LED, the color which we came up is the average of red and blue i.e. (127, 0, 127), which is a shade of pink. The above classification table gives an insight of how colors are assigned to each range. In addition to this, we wanted same LED to represent the wind speed also. For that, we considered if it is no windy, the LED shouldn't blink but should represent the corresponding temperature color. If it is less windy, LEDs should blink slowly and if it is windier, the LED should blink very fast. Below is the table that explains briefly about assigning the LED patterns.

Assigning patterns based on classification

TEMPERATURE	WIND SPEED	LED PATTERN
0	0	0 => Cold No Windy
0	1	1 => Cold Less Windy
0	2	2 => Cold More Windy
1	0	3 => Normal No Windy
1	1	4 => Normal Less Windy
1	2	5 => Normal More Windy
2	0	6 => Hot No Windy
2	1	7 => Hot Less Windy
2	2	8 => Hot More Windy

Figure 3: represents which LED pattern is assigned to a combination of temperature and wind speed

ARDUINO and LED connection:

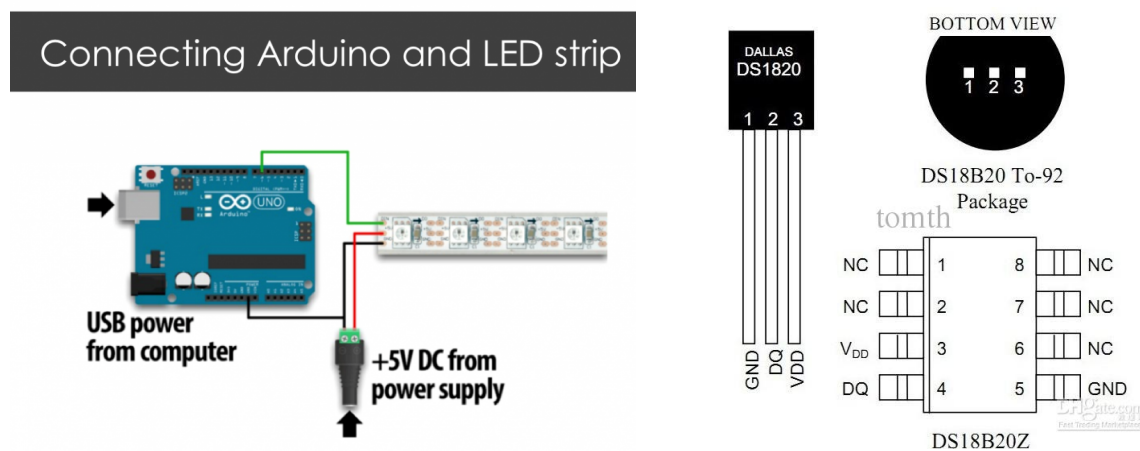


Figure 4: on the left represents schematic of how Arduino is connected to LED strip and on the right represent the temperature sensor.

IMPLEMENTATION:

Quantum Light allows user to choose either weather data from sensor or from Internet. if the data chosen is from Internet, first weather data is obtained from online. In our case we are obtaining data from Yahoo server and classifying it, assigning a pattern, communicating to Arduino and displaying LED. Below is the schematic that explains how the whole procedure works in real time scenario.

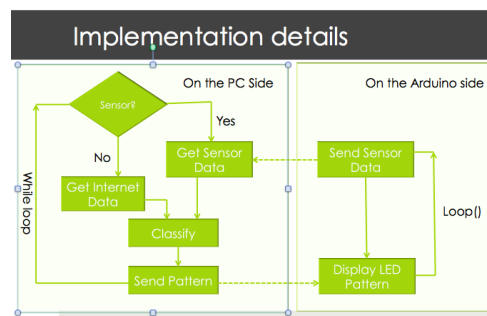
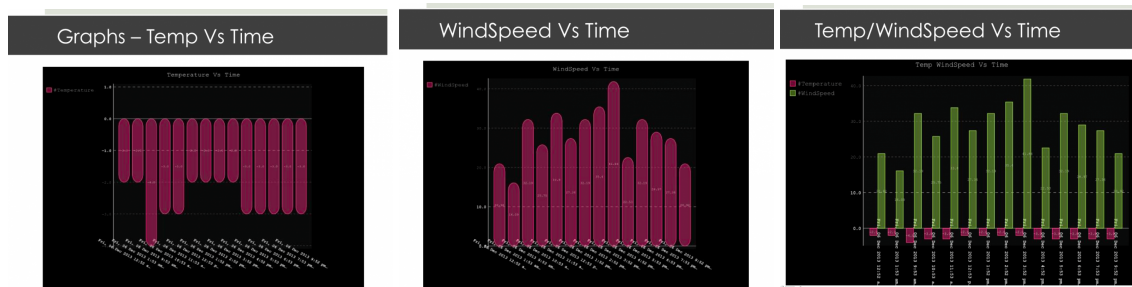


Figure 5: represents an overview of how quantum light works

Similarly, If the data chosen is from sensor then our computer communicates to Arduino via serial, a library i.e. used to communicate between computer and Arduino and retrieves the data obtained from sensor mounted on Arduino board. Here we are using only Temperature sensor. So we get temperature data. This temperature data is classified and depending upon which classification it fall into, a LED pattern is assigned as mentioned in figure 2 And figure 3. This LED pattern is communicated to Arduino via serial, which is connected to LED strip.

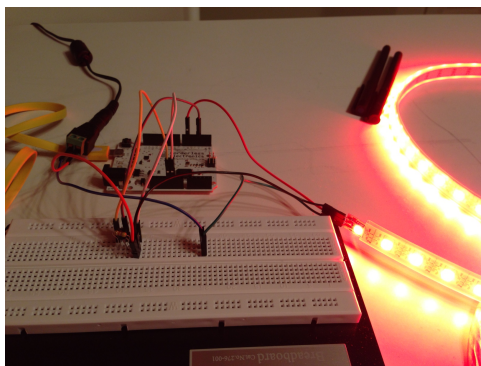
DATA ANALYSIS:

We analyzed temperature and wind speed on hourly basis. Data show that wind speed varies invariably for every hour (even seconds) of the day while sample data of temperature shows that they are quite uniform for most part of the day. From analyzing these data, we show that temperature and wind speed are not a dependent factor and thus user should make a correct choice in judging the weather.



RESULTS:

Our results can be well explained from the video snippets attached to the wiki page. The LED pattern displays differently for different weather condition. Here are few screen shots



Learning's from the Project:

- Seamlessly integrating multiple components and libraries together.
- How Arduino actually works with real time data.

- How to get temperature data using sensors.
- Getting the data, Classifying, assigning a pattern and sending that pattern to Arduino and displaying LEDs based on the pattern.
- How loops work with so many components.

Library Used:

- Weather API (pywapi)
 - To get weather data from online
- Serial
 - Python module to Communicate between computer and Arduino via serial port
- Onewire
 - To get temperature data from temperature sensors
- Adafruit_NeoPixel
 - To light led patterns using led strip

Challenges:

- Making one Arduino sketch to work for both sensor sketch and LED displaying sketch.
- Problems with loop everywhere: Changing LED patterns automatically based on current temperature.

Future work:

- Creating an App that has following features
 - Picture corresponding to weather data
 - Show temperature, wind, humidity, rain, precipitation
 - Max, Min temp in a day/week
 - Use of other sensors: humidity and wind
- Comparing indoor and outdoor sensors data and coming up with LED patterns to show the differences.
- Using Wi-Fi on Galileo
- Get the modules to work for Galileo, We had some compilation issues using Onewire library on Galileo.

References:

- Project Github link: <https://github.com/sasmita/QuantumLight>
- <http://arduino.cc/en/Tutorial/HomePage>
- https://github.com/adafruit/Adafruit_NeoPixel
- <http://playground.arduino.cc/interfacing/python>