

**Module Title :** Internet of Things

**Module Code :** BCC611

**Credit Points :** 20

**FHEQ Level :** 6

**Assessment :** AE2

**Student Number :** 14883899

**Student Name** : M.I.B.Farha

**Course title :** BSc (Hons) in Software Engineering

**Seminar Tutor :** Anis Saboordeen

# Table of Content

[Table of Content 2](#_Toc29579551)

[Table of Figures 4](#_Toc29579552)

[Acronyms 8](#_Toc29579553)

[1. Introduction 9](#_Toc29579554)

[2. Key aspects of the IoT model 10](#_Toc29579555)

[2.1 Sensors/ Actuators (SA) 10](#_Toc29579556)

[2.1.1 SA Set- 01 10](#_Toc29579557)

[I. Temperature Sensor (TS) 10](#_Toc29579558)

[II. Mini Fan (MF) 10](#_Toc29579559)

[III. Limitation 10](#_Toc29579560)

[2.1.2 SA Set- 02 11](#_Toc29579561)

[I. Light Intensity Sensor (Keys Photo Resister Sensor) 11](#_Toc29579562)

[II. Night LED bulbs 11](#_Toc29579563)

[III. Limitation 11](#_Toc29579564)

[2.1.3 SA Set- 03 12](#_Toc29579565)

[I. Sound sensor 12](#_Toc29579566)

[II. Speaker 12](#_Toc29579567)

[III. Limitation 12](#_Toc29579568)

[2.2 Connectivity 13](#_Toc29579569)

[2.3 Data Visualization /Analysis 15](#_Toc29579570)

[2.3.1 Data Visualization 15](#_Toc29579571)

[2.3.2 Data Analysis 16](#_Toc29579572)

[I. Temperature Analysis 16](#_Toc29579573)

[II. Light Intensity Analysis 18](#_Toc29579574)

[III. Sound Analysis 19](#_Toc29579575)

[3. Model Implementation 20](#_Toc29579576)

[4. IoT Model Evaluation 23](#_Toc29579577)

[5. Test and Validation 24](#_Toc29579578)

[5.1 Output / Results 24](#_Toc29579579)

[5.1.1 DHT11 Tester 24](#_Toc29579580)

[I. Schematic Diagram 24](#_Toc29579581)

[II. Code 24](#_Toc29579582)

[III. Testing Picture 26](#_Toc29579583)

[IV. Output 26](#_Toc29579584)

[5.1.2 Light Intensity 27](#_Toc29579585)

[I. Schematic Diagram 27](#_Toc29579586)

[II. Code 27](#_Toc29579587)

[III. Testing Picture 28](#_Toc29579588)

[IV. Output 28](#_Toc29579589)

[5.1.3 Sound Sensor 29](#_Toc29579590)

[I. Schematic Diagram 29](#_Toc29579591)

[II. Code 29](#_Toc29579592)

[III. Testing Picture 30](#_Toc29579593)

[IV. Output 30](#_Toc29579594)

[5.1.4 LED 31](#_Toc29579595)

[I. Schematic Diagram 31](#_Toc29579596)

[II. Code 31](#_Toc29579597)

[III. Testing Picture 32](#_Toc29579598)

[IV. Output 32](#_Toc29579599)

[5.1.5 Speaker 33](#_Toc29579600)

[I. Schematic Diagram 33](#_Toc29579601)

[II. Code 33](#_Toc29579602)

[III. Testing Picture 34](#_Toc29579603)

[IV. Output 34](#_Toc29579604)

[5.1.6 Mini Fan 35](#_Toc29579605)

[I. Schematic Diagram 35](#_Toc29579606)

[II. Code 36](#_Toc29579607)

[III. Testing Picture 36](#_Toc29579608)

[5.2 Limitation 37](#_Toc29579609)

[6. Future Development 38](#_Toc29579610)

[7. Conclusion 39](#_Toc29579611)

[8. References 40](#_Toc29579612)

[9. Appendix 42](#_Toc29579613)

[9.1 Code Samples 42](#_Toc29579614)

[9.1.1 Temperature Sensor Testing Code 42](#_Toc29579615)

[9.1.2 Light Intensity Sensor Testing Code 45](#_Toc29579616)

[9.1.3 Sound Sensor Testing Code 46](#_Toc29579617)

[9.1.4 LED light Testing Code 47](#_Toc29579618)

[9.1.5 Sound sensor Testing Code 48](#_Toc29579619)

[9.1.6 Mini Fan Testing Code 49](#_Toc29579620)

[9.2 Ethical and privacy consideration 50](#_Toc29579621)

[9.3 Baby Monitoring System 51](#_Toc29579622)

[9.3.1 Full System Code 51](#_Toc29579623)

[9.3.2 System working method 60](#_Toc29579624)

[9.3.3 Important Code Section/Logics 61](#_Toc29579625)

[9.3.4 System Implementing Pictures and whole testing Pictures 65](#_Toc29579626)

[9.4 Presentation Slides 70](#_Toc29579627)

# Table of Figures

[Figure 1- Temperature Sensor (electropeak.com, 2020) 9](file:///F:\Degree\IOT\14883899-Farha_Imthiyaz-IOT-AE2.docx#_Toc29579465)

[Figure 2- Mini Fan (adafruit.com, 2020) 9](file:///F:\Degree\IOT\14883899-Farha_Imthiyaz-IOT-AE2.docx#_Toc29579466)

[Figure 3- Light Intensity Sensor (amazon.com, 2020) 10](file:///F:\Degree\IOT\14883899-Farha_Imthiyaz-IOT-AE2.docx#_Toc29579467)

[Figure 4- LED bulbs (shutterstock.com, 2020) 10](file:///F:\Degree\IOT\14883899-Farha_Imthiyaz-IOT-AE2.docx#_Toc29579468)

[Figure 5- Sound Sensor (amazon.com, 2020) 11](#_Toc29579469)

[Figure 6- Speaker (banggood.in, 2020) 11](#_Toc29579470)

[Figure 7- This is Diagram which explains connectivity (Author’s work, 2019) 12](#_Toc29579471)

[Figure 8- Blynk Dashboard Visualization (Author’s work, 2019) 14](file:///F:\Degree\IOT\14883899-Farha_Imthiyaz-IOT-AE2.docx#_Toc29579472)

[Figure 9- Blynk Dashboard Visualization (Author’s work, 2019) 14](file:///F:\Degree\IOT\14883899-Farha_Imthiyaz-IOT-AE2.docx#_Toc29579473)

[Figure 10- Power BI Dashboard Analysis for temperature (Author’s work, 2019) 15](#_Toc29579474)

[Figure 11- Power BI Dashboard Analysis for temperature (Author’s work, 2019) 16](#_Toc29579475)

[Figure 12- Power BI Dashboard Analysis for light intensity (Author’s work, 2019) 17](#_Toc29579476)

[Figure 13- Power BI Dashboard Analysis for sound (Author’s work, 2019) 18](#_Toc29579477)

[Figure 14- Schematic Diagram for BMS (Author’s work, 2019) 19](#_Toc29579478)

[Figure 15- Schematic Diagram for testing DHT11 (Author’s work, 2019) 23](#_Toc29579479)

[Figure 16- Code for DHT11 Testing 23](file:///F:\Degree\IOT\14883899-Farha_Imthiyaz-IOT-AE2.docx#_Toc29579480)

[Figure 17- Code for DHT11 Testing 24](#_Toc29579481)

[Figure 18- Code for DHT11 Testing 24](#_Toc29579482)

[Figure 19- DHT11 testing picture 25](file:///F:\Degree\IOT\14883899-Farha_Imthiyaz-IOT-AE2.docx#_Toc29579483)

[Figure 20- DHT11 testing picture 25](file:///F:\Degree\IOT\14883899-Farha_Imthiyaz-IOT-AE2.docx#_Toc29579484)

[Figure 21- Output result for DHT11 25](#_Toc29579485)

[Figure 22- schematic Diagram for Light Intensity testing (Author’s work, 2019) 26](#_Toc29579486)

[Figure 23- Code for Light Intensity testing 26](#_Toc29579487)

[Figure 24- Testing picture of light intensity sensor 27](file:///F:\Degree\IOT\14883899-Farha_Imthiyaz-IOT-AE2.docx#_Toc29579488)

[Figure 25- Testing picture of light intensity sensor 27](file:///F:\Degree\IOT\14883899-Farha_Imthiyaz-IOT-AE2.docx#_Toc29579489)

[Figure 26- Output of light intensity sensor 27](#_Toc29579490)

[Figure 27- Schematic diagram for sound sensor 28](#_Toc29579491)

[Figure 28- Code for Sound sensor 28](#_Toc29579492)

[Figure 29- Testing Picture of Sound sensor 29](file:///F:\Degree\IOT\14883899-Farha_Imthiyaz-IOT-AE2.docx#_Toc29579493)

[Figure 30- Testing Picture of Sound sensor 29](file:///F:\Degree\IOT\14883899-Farha_Imthiyaz-IOT-AE2.docx#_Toc29579494)

[Figure 31- Output of Sound sensor testing 29](#_Toc29579495)

[Figure 32- Schematic diagram for LED testing 30](#_Toc29579496)

[Figure 33- Code for LED testing code 30](#_Toc29579497)

[Figure 35- Testing picture of LED 31](file:///F:\Degree\IOT\14883899-Farha_Imthiyaz-IOT-AE2.docx#_Toc29579498)

[Figure 36- Testing picture of LED 31](file:///F:\Degree\IOT\14883899-Farha_Imthiyaz-IOT-AE2.docx#_Toc29579499)

[Figure 34- Testing picture of LED 31](file:///F:\Degree\IOT\14883899-Farha_Imthiyaz-IOT-AE2.docx#_Toc29579500)

[Figure 37- Output of LED testing 31](#_Toc29579501)

[Figure 38- Schematic Diagram for Speaker (Author’s work, 2019) 32](#_Toc29579502)

[Figure 39- Code for Speaker testing 32](#_Toc29579503)

[Figure 40- Testing picture of Speaker 33](file:///F:\Degree\IOT\14883899-Farha_Imthiyaz-IOT-AE2.docx#_Toc29579504)

[Figure 41- Testing Picture of Speaker 33](#_Toc29579505)

[Figure 42- Output of Speaker testing 33](#_Toc29579506)

[Figure 43- Schematic Diagram for fan (Author’s work, 2019) 34](#_Toc29579507)

[Figure 44- Code for testing fan 35](#_Toc29579508)

[Figure 45- Testing Picture of Fan 35](#_Toc29579509)

[Figure 46- Blynk Dashboard for BMS 59](file:///F:\Degree\IOT\14883899-Farha_Imthiyaz-IOT-AE2.docx#_Toc29579510)

[Figure 47- code for temperature sensor function 60](#_Toc29579511)

[Figure 48- code for light intensity and sound sensor functions 60](#_Toc29579512)

[Figure 49- code for sending data from sensors to Blynk 61](#_Toc29579513)

[Figure 50- code for Control from Blynk to fan 61](#_Toc29579514)

[Figure 51- code for Control from Blynk to LED 62](#_Toc29579515)

[Figure 52- code for Control from Blynk to Speaker 62](file:///F:\Degree\IOT\14883899-Farha_Imthiyaz-IOT-AE2.docx#_Toc29579516)

[Figure 53- Code for Blynk setup and loop functions 63](#_Toc29579517)

[Figure 54- Picture of Testing and Implementing 64](file:///F:\Degree\IOT\14883899-Farha_Imthiyaz-IOT-AE2.docx#_Toc29579518)

[Figure 55- Picture of Testing and Implementing 64](file:///F:\Degree\IOT\14883899-Farha_Imthiyaz-IOT-AE2.docx#_Toc29579519)

[Figure 56- Picture of Implementing 64](file:///F:\Degree\IOT\14883899-Farha_Imthiyaz-IOT-AE2.docx#_Toc29579520)

[Figure 57- Picture of Testing and Implementing 64](#_Toc29579521)

[Figure 59- Picture of Implementing 65](file:///F:\Degree\IOT\14883899-Farha_Imthiyaz-IOT-AE2.docx#_Toc29579522)

[Figure 60- Picture of Implementing 65](file:///F:\Degree\IOT\14883899-Farha_Imthiyaz-IOT-AE2.docx#_Toc29579523)

[Figure 58- Picture of Implementing 65](file:///F:\Degree\IOT\14883899-Farha_Imthiyaz-IOT-AE2.docx#_Toc29579524)

[Figure 61- Picture of Implementing 65](file:///F:\Degree\IOT\14883899-Farha_Imthiyaz-IOT-AE2.docx#_Toc29579525)

[Figure 62- Picture of Implementing 66](file:///F:\Degree\IOT\14883899-Farha_Imthiyaz-IOT-AE2.docx#_Toc29579526)

[Figure 63- Picture of Testing and Implementing 66](file:///F:\Degree\IOT\14883899-Farha_Imthiyaz-IOT-AE2.docx#_Toc29579527)

[Figure 64- Picture of Implementing 66](file:///F:\Degree\IOT\14883899-Farha_Imthiyaz-IOT-AE2.docx#_Toc29579528)

[Figure 65- Picture of Implementing 66](file:///F:\Degree\IOT\14883899-Farha_Imthiyaz-IOT-AE2.docx#_Toc29579529)

[Figure 66- Picture of Implementing 67](file:///F:\Degree\IOT\14883899-Farha_Imthiyaz-IOT-AE2.docx#_Toc29579530)

[Figure 67- Picture of Implementing 67](file:///F:\Degree\IOT\14883899-Farha_Imthiyaz-IOT-AE2.docx#_Toc29579531)

[Figure 68- Picture of Testing and Implementing 67](file:///F:\Degree\IOT\14883899-Farha_Imthiyaz-IOT-AE2.docx#_Toc29579532)

[Figure 69- Picture of Testing and Implementing 67](file:///F:\Degree\IOT\14883899-Farha_Imthiyaz-IOT-AE2.docx#_Toc29579533)

[Figure 70- Picture of Testing and Implementing 68](#_Toc29579534)

[Figure 71- Picture of Testing and Implementing 68](file:///F:\Degree\IOT\14883899-Farha_Imthiyaz-IOT-AE2.docx#_Toc29579535)

[Figure 72- Picture of Testing and Implementing 68](file:///F:\Degree\IOT\14883899-Farha_Imthiyaz-IOT-AE2.docx#_Toc29579536)

[Figure 73- Beginning Slide 69](#_Toc29579537)

[Figure 74- Contents Slide 69](#_Toc29579538)

[Figure 75- Problem & Solution Slide 70](#_Toc29579539)

[Figure 76- Block Diagram for BMS 70](#_Toc29579540)

[Figure 77- Schematic Diagram for BMS 71](#_Toc29579541)

[Figure 78- Slide of System Components 71](#_Toc29579542)

[Figure 79- Slide of Artifact Pictures 72](#_Toc29579543)

[Figure 80- Slide of Artifact Pictures 72](#_Toc29579544)

[Figure 81- Slide of Blynk Dashboards 73](#_Toc29579545)

[Figure 82- Slide of System Functions 73](#_Toc29579546)

[Figure 83- Slide of Strengths & weakness 74](#_Toc29579547)

[Figure 84- Slide of Future Improvement 74](#_Toc29579548)

[Figure 85- Slide of Conclusion 75](#_Toc29579549)

[Figure 86- Ending Slide 75](#_Toc29579550)

# Acronyms

BMS – Baby Monitoring System

IoT – Internet of Things

SA - Sensors/ Actuators

SM- Servo Motor

PWM- Pulse With Module

TS- Temperature Sensor

# Introduction

I created Baby monitoring system (BMS) for IoT project because, now a days every parents are busy with office works and home works. When baby crying Parents can’t do any works. They need to care the baby. So, all the works will be remaining. From this system we can reduce the pressure of parents from a monitoring infant child through their devices and they can also control actuators manually.

As a sensors we used Temperature sensor, Light Intensity sensor and Sound sensor. As an actuators Mini fan, Night LED bulbs and Speaker. As a board we used NodeMCU. Temperature, light intensity of baby sleeping environment and sound of surrounding (cry detection) captured by sensors and sending to cloud. Cloud is connected to Blynk. From this IoT platform we can control Actuators manually.

**Temperature Sensor (DHT11)** is measure the room Temperature and we controlled **Mini fan** from our devices. **Light Intensity Sensor (Keys Photo Resister Sensor)** to measure light intensity of baby living room. According to that controlled **Night LED bulbs**. **Sound sensor** to detect strength of sound (Cry detection) of baby according to that we set music in **speaker**. These sensors work automatically also and manually control these actuators (Automation).

# Key aspects of the IoT model

## Sensors/ Actuators (SA)

### C:\Users\Dell\Desktop\am2301-dht21__3_.jpgSA Set- 01

#### Temperature Sensor (TS)

Temperature sensor is using to measure the temperature of someone or something. In here we are using DHT21 to measure the baby’s room Temperature.

Figure - Temperature Sensor (electropeak.com, 2020)

#### Image result for benefit of mini fan in IoTMini Fan (MF)

This is an Actuator that allows to cool down and get the air to an environment. In here we are using mini fan in baby cot. We are connecting MF to PWM pin.

Figure - Mini Fan (adafruit.com, 2020)

#### Limitation

* Temperature Sensor value= Temp
* As a Blynk controller we used two **value displays** to display exact value of temperature and its status. We used **Button** Controller to control fan in Blynk.
* According to TS value MF work automatically or manually we can control by our Blynk.

|  |  |  |
| --- | --- | --- |
| TS Condition  (Celsius) | Display in Blynk (Temp) | Fan Status |
| Temp<26 | Temp | Off |
| Temp==26 & Temp>26 | Temp | On |

### Image result for image of Keys Photoresistor SensorSA Set- 02

#### Light Intensity Sensor (Keys Photo Resister Sensor)

Light Intensity Sensor (LIS) is using to measure the physical quality of light rays. In here we are using to measure the light intensity of baby living room.

Figure - Light Intensity Sensor (amazon.com, 2020)

#### Related imageNight LED bulbs

This is an actuator which produce light. In here we are using to detect light intensity of baby living room. According to that **Night LED bulbs** control**.**

Figure - LED bulbs (shutterstock.com, 2020)

#### Limitation

LISstatus=Analog read of LIS

If Light Intensity low meant we will switch on LED bulbs using Blynk manually.

|  |  |  |
| --- | --- | --- |
| LISstatus | Real Status | LED bulbs |
| LISstatus<=200 | Dark | On |
| LISstatus>200 | Light | Off |

### SA Set- 03

#### Sound sensor

Sound Sensor is using to detect the sound strength of environment. In here we detect strength of sound (Cry detection) of baby according to that we set music in **speaker.**



Figure - Sound Sensor (amazon.com, 2020)

#### C:\Users\Dell\Desktop\0a2856de-9ea9-4816-8309-aef36ea34827.jpgSpeaker

This is an actuator which produce Audio output. In here we are using this to detect Cry of baby according to that music is playing.

Figure - Speaker (banggood.in, 2020)

#### Limitation

When Sound sensor (Cry) detects high meant we switched on the Lullaby Music in speaker. If not we off it manually.

|  |  |
| --- | --- |
| Sound sensor Status (Digital) | Speaker (Lullaby Music) |
| High | On |
| Low | Off |

## **Connectivity**

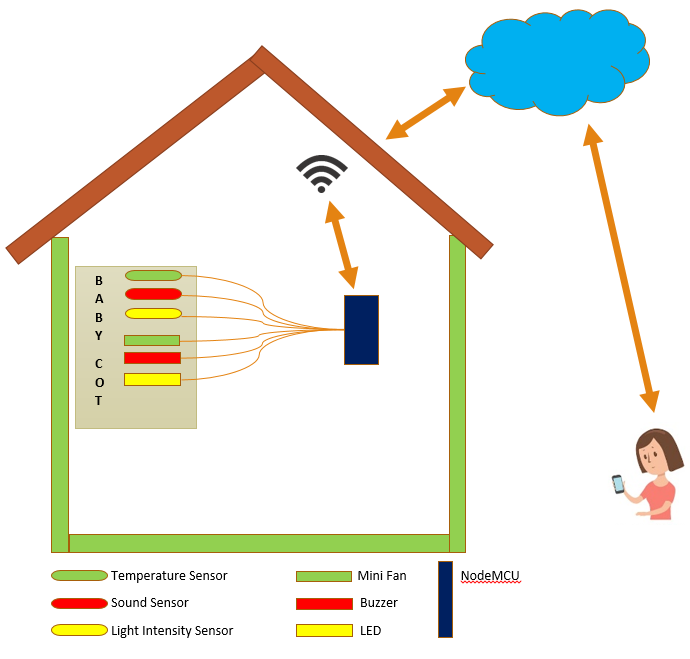


Figure - This is Diagram which explains connectivity (Author’s work, 2019)

We can divided Connectivity to three. They are communication, cloud and platform.

* As a **Communication** we used Wi-Fi and MQTT Protocol**.**
* **Wi-Fi** –This is a technology which uses radio waves to provide network connectivity. It create hotspots wirelessly and allow users to connect to it. In our system the through Wi-Fi the Sensors, cloud and actuators are connected with each other and shared the information.
* **MQTT Protocol** –This is light weight sub and pub system where we can pub and sub messages. So, In Blynk platform this protocol is using.
* We used **cloud** platform to store data high secure. Our sensors data’ send to cloud and store in the database. From our Blynk we accessed those data.
* As a **Platform** we used Blynk.
* **Blynk** - IoT platform to connect our devices to the cloud and design apps to control them. From this we monitor our system and control actuators also can analyze the data.

## Data Visualization /Analysis

### Data Visualization

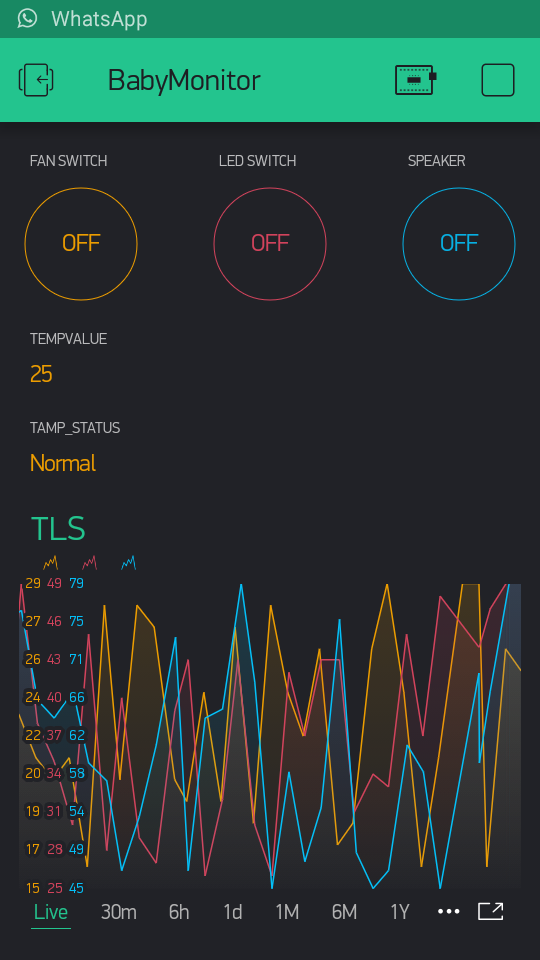
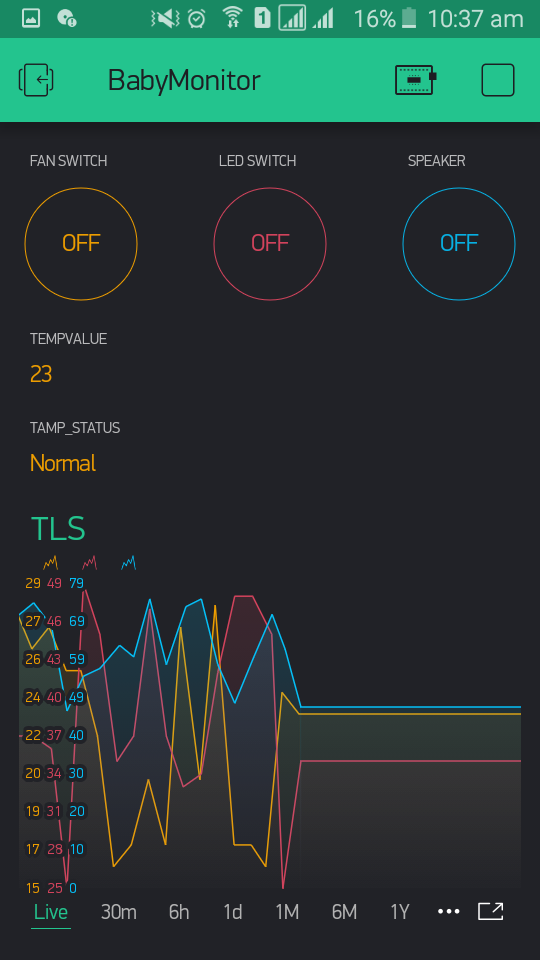
As a **Data Visualization** we used one **super chart** controller to show live sensor data of temperature sensor, Light Intensity sensor and sound sensor in Blynk platform. From this we can identify the pattern of data and its value.

Figure - Blynk Dashboard Visualization (Author’s work, 2019)

Figure - Blynk Dashboard Visualization (Author’s work, 2019)

### Data Analysis

As a **Data Analysis** I used Power BI. This is a **business intelligence** platform which provides business users with tools for aggregating, analyzing, visualizing and sharing data. So, we analyzed sensor data and we can get main decisions about baby’s cry and its environment.

#### Temperature Analysis

Temperature Sensor Analysis. We included three properties in data. They are

* Room: The room in which the temperature was recorded.
* Time: The time in YMD-HMS format at which the temperature was recorded.
* Temperature: The temperature in Fahrenheit.

In here I am considering same Room Temperature in one month.

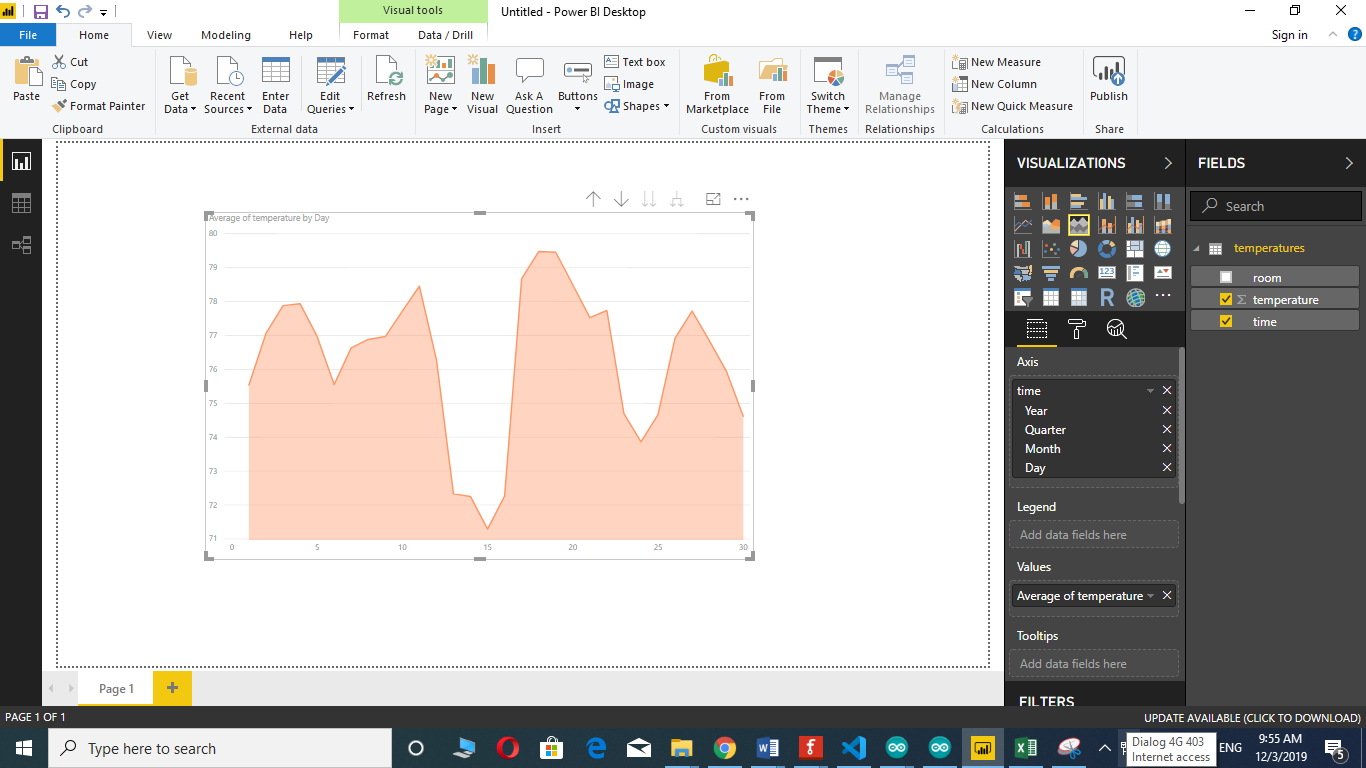


Figure - Power BI Dashboard Analysis for temperature (Author’s work, 2019)

In here I placed system in different Rooms.

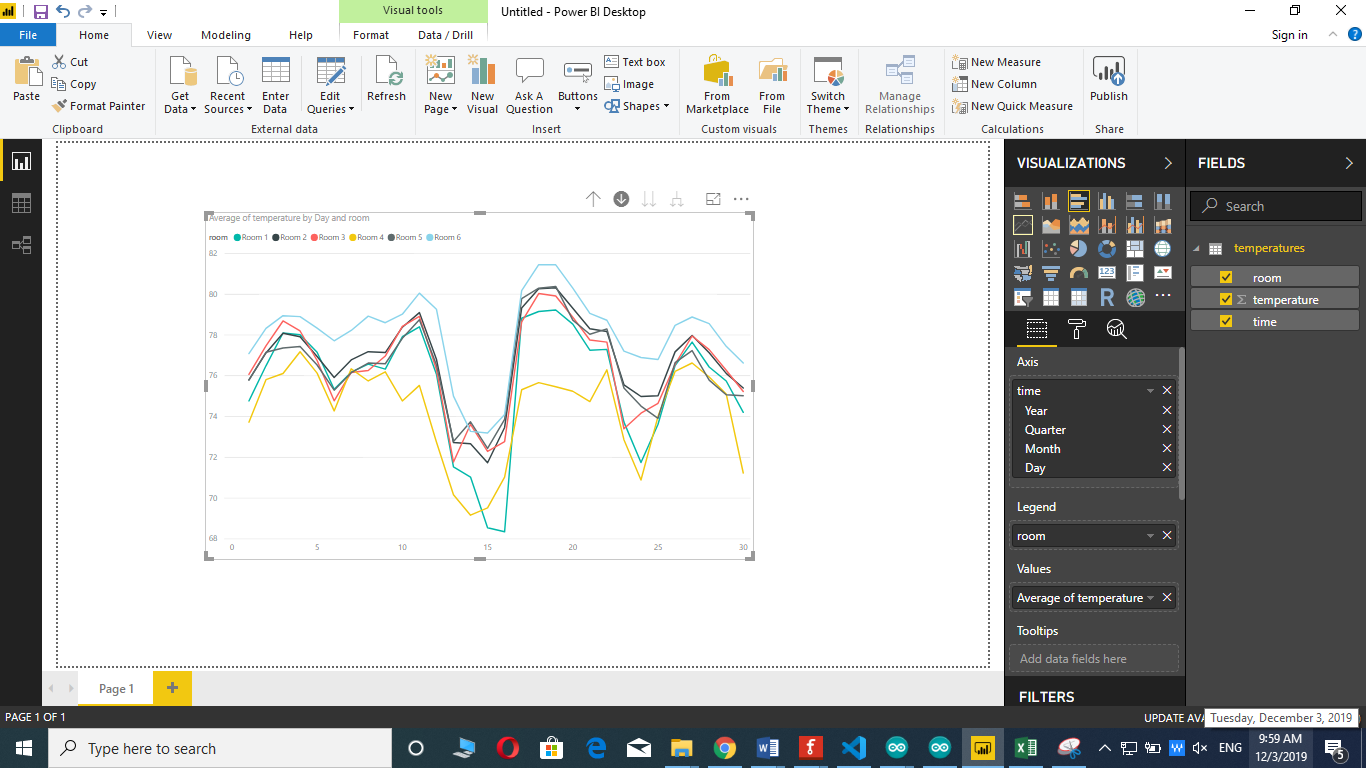


Figure - Power BI Dashboard Analysis for temperature (Author’s work, 2019)

Temperature Analysis

* According to one month temperature analysis in one room, Most temperature is below 78.8 (Fahrenheit, (26Celcius)). For us don’t need to switch on the fan.
* According to second analysis if I placed BMS in to many rooms almost temperature is different from each other. Some room I have to switch on fan some of room don’t need to on fan (According to above logic).

#### Light Intensity Analysis

We used light Intensity data to take some decisions. So, we took 30 days light intensity data for analyzing. We included below screenshots three main analysis. Line chart is to analyze average light of the day. Column chart is to analyze max light of days. Multi row card is to filter average light is less than 50.

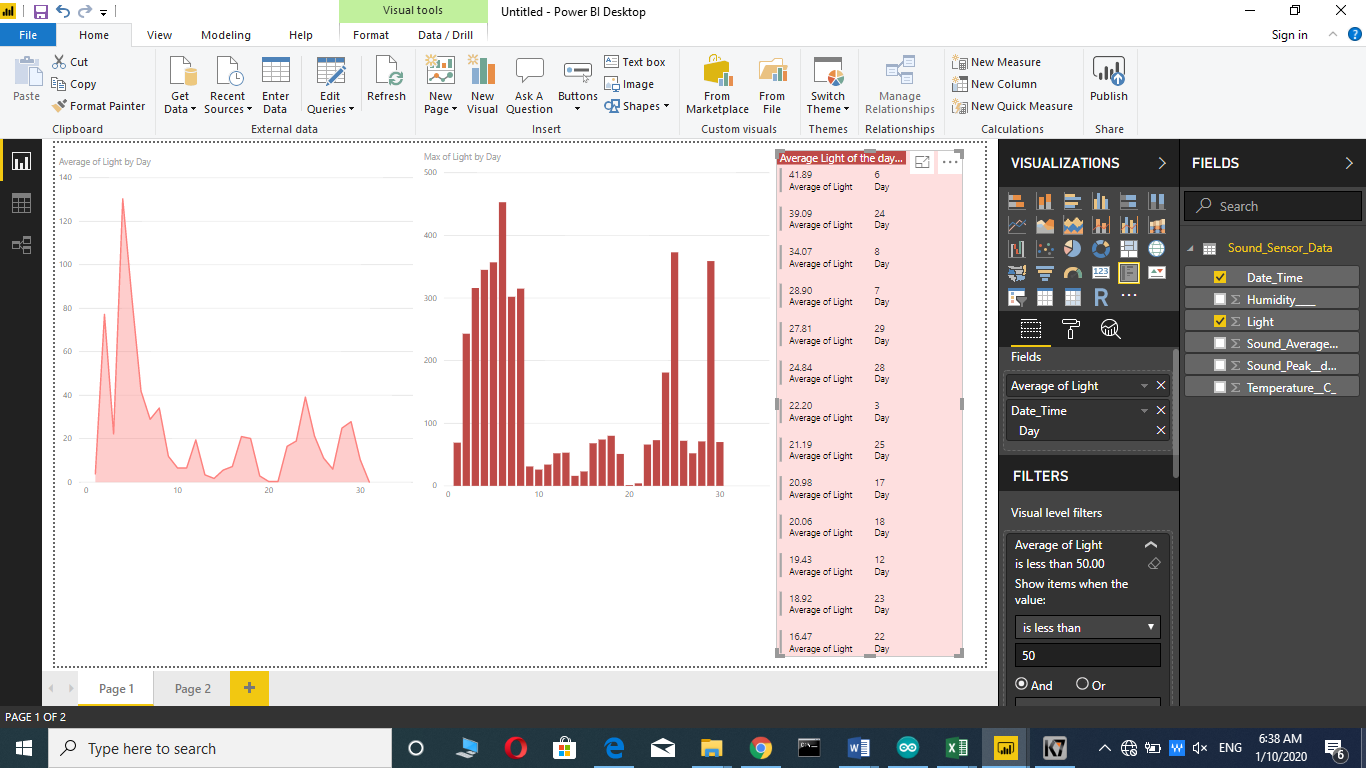


Figure - Power BI Dashboard Analysis for light intensity (Author’s work, 2019)

From this analysis, we can get in which light amount we need to on the LED bulbs. How many days light amount is more than 50 in a month? According to that we can predict future months light usage of baby living environment.

#### Sound Analysis

We used Sound data to take some decisions. So, we took 30 days sound data for analyzing. We included below screenshots two main analysis. Column chart is to analyze average sound and peak sound in average amount. Line chart is to analyze average sound of the day.

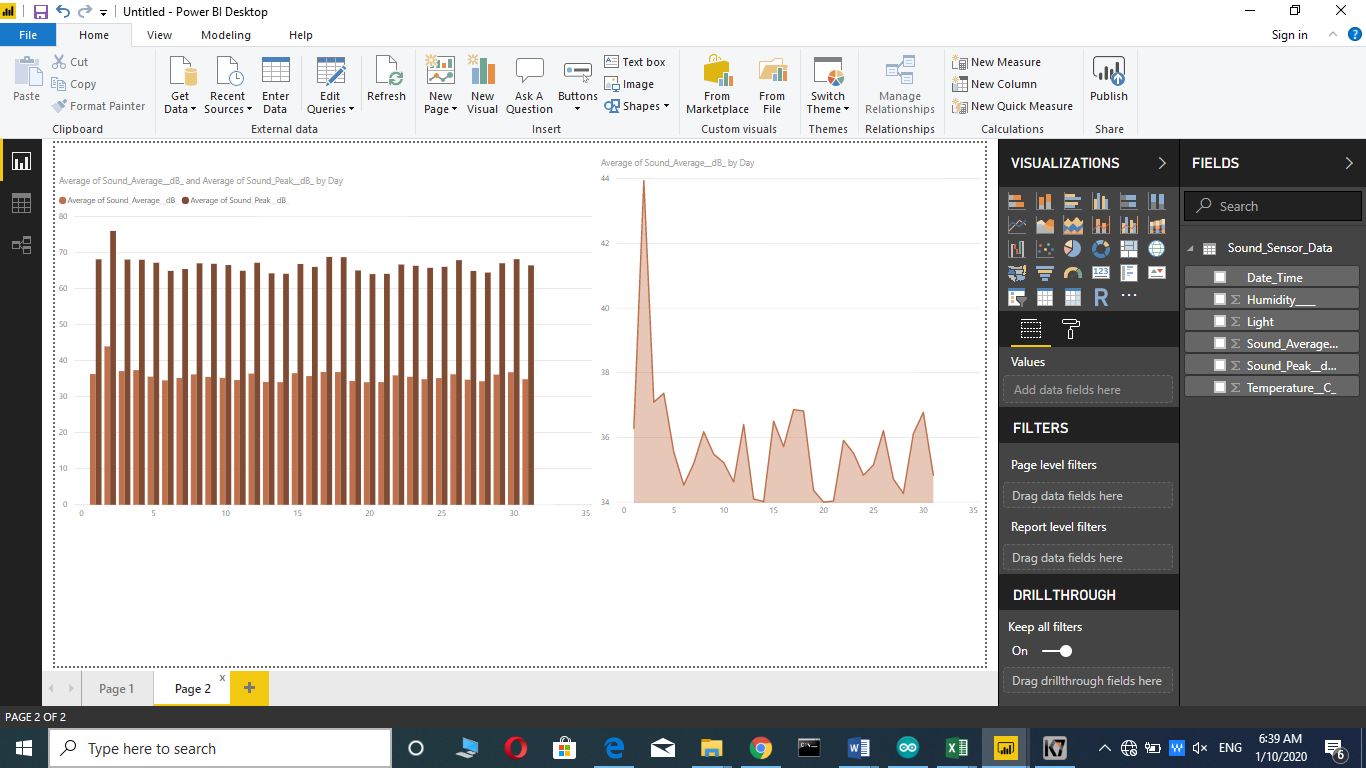


Figure - Power BI Dashboard Analysis for sound (Author’s work, 2019)

From this analysis, we can get in which sound amount we need to on the music in speaker and when was the day baby cry more? We can compare peak sound and average sound of per day.

# Model Implementation

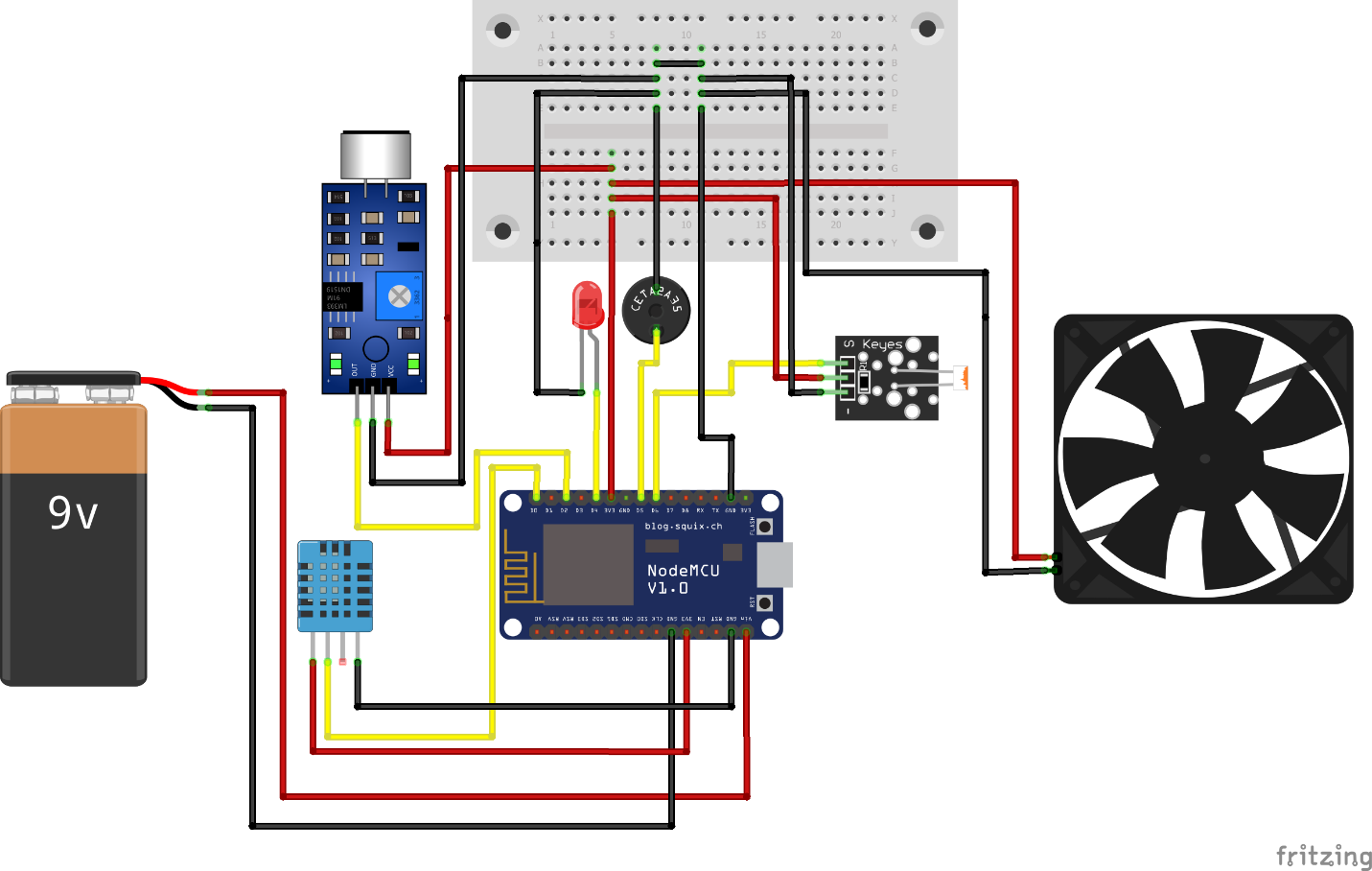


Figure - Schematic Diagram for BMS (Author’s work, 2019)

**Step 01- Planning the Components pins**

First, we planned and defined Components pins as analog and Digital.

**Step 02- Test all the Components**

We tested sensors, actuators and NodeMCU with coding’s which they work properly or not.

**Step 03- Connect all the components together**

We connected Sensors, actuators and battery with NodeMCU with jumper wire.

**Step 04- implementation of system**

Then, we implemented functions codes for all sensors and actuators using Arduino IDE.

**Step 05- Testing system (White box testing)**

We tested every each sensors and actuators functions and logics code by code.

**Step 06- Upload to Board**

Then, compiled whole code and upload to NodeMCU.

**Step 06- Create Blynk dashboard**

We designed a dashboard which has controllers to control actuators.

**Step 08- Giving Connection**

We connected our system with Home Router (HR). Then, connected HR and Blynk to cloud through internet.

**Step 09- System testing (Black box testing)**

Then, we tested from our Blynk dashboard which sensors input data correctly, from Blynk controllers’ actuators work properly or not. BMS automation work properly or not.

**Step 10- Place the system**

We placed the system in to baby cot.

In Proposal I decided to choose Anaconda. After that according to our lectures suggestion and my previous experience I choose Power BI. I thought earlier we can’t use mini fan directly as an actuator. So, I mentioned in proposal Servo Motor. But, when I am implementing system only I know we can use mini fan as actuator as well I used fan for reduce heat.

In proposal I mentioned some of sensors specification and pictures are different when it comes to real practical I change those information according to my project correctly. I mentioned in Proposal as cloud I use AWS. In real practice I used Blynk cloud.

# IoT Model Evaluation

Our BMS expense is Low in cost. BMS system is located in the home. So, it protects sensors from weather effects. We can get sensor data very much accurately. Parents’ workload will reduce. So, they can manage all the works together. I analyzed sensor data from Power BI and easily got decision about baby’s environment. We can control three main factors from this system. They are sound, light and temperature.

System work as automation because user can careless some times and we can manually control actuators by our platform. I am using Wi-Fi. So, we can access our system in any other places. So, parents can control our BMS from anywhere. BMS reduce waste of power and Time. For BMS we used NodeMCU. Because Wi-Fi is inbuilt in this board. We used Cloud. Because almost everyone using this technology to store sensing data in one place. As a platform we used Blynk. Because, this is using MQTT Protocol and can create user friendly dashboards.

Weaknesses of our BMS is we store our system’s data in third party cloud system. We can’t exactly trust their security. If there is a power battery finished meant system not work. I detect only three factors in my system. From that we can’t take exact decision. If we have to take decision more accurate meant we have to use more sensors. Monitoring System meant main part is users can view by camera. But, in our BMS we aren’t included Camera so, can’t view baby. When Wi-Fi slow down meant our system working will effect. Sound sensor detect not only baby cry detection but also it detect environment also. So, that will cause to wrong decision taking. In BMS we are mainly focusing baby’s environment (light intensity, Temperature) than baby (Sound). But, this main weakness. We need to include more factors about baby to take decisions.

# Test and Validation

## Output / Results

### DHT11 Tester

#### Schematic Diagram

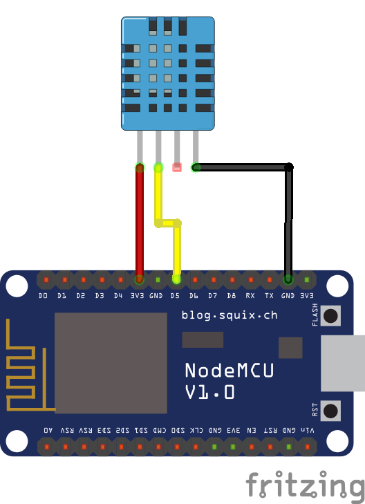


Figure - Schematic Diagram for testing DHT11 (Author’s work, 2019)

#### Code

Figure - Code for DHT11 Testing

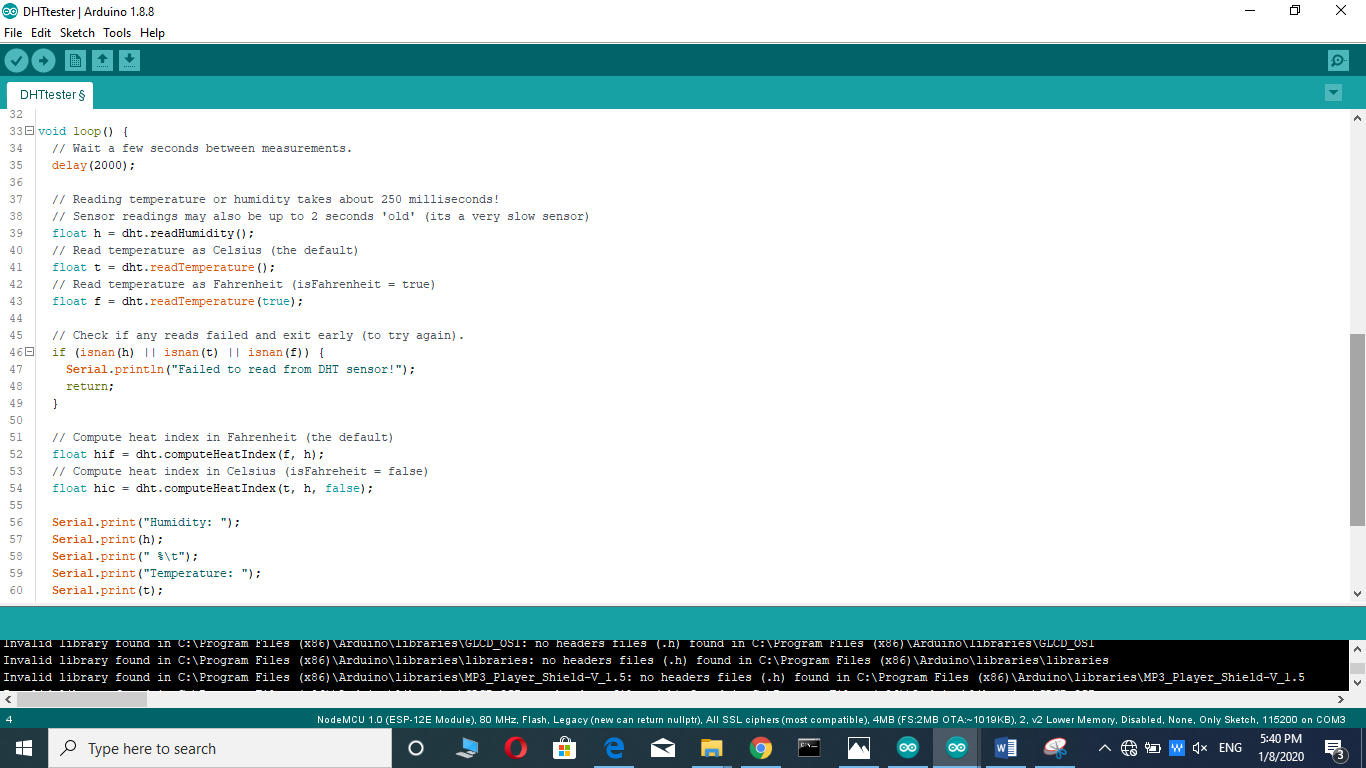


Figure - Code for DHT11 Testing

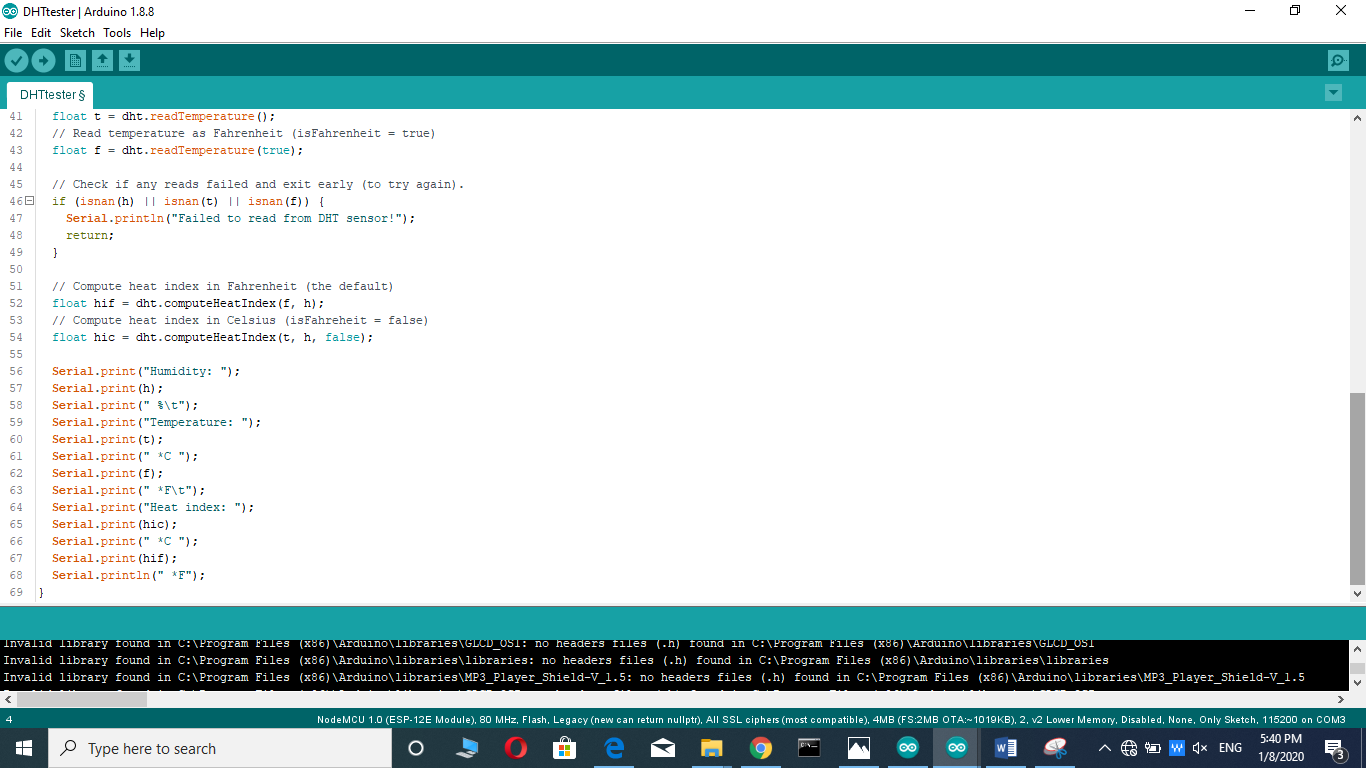


Figure - Code for DHT11 Testing

#### Testing Picture



Figure - DHT11 testing picture



Figure - DHT11 testing picture

#### Output

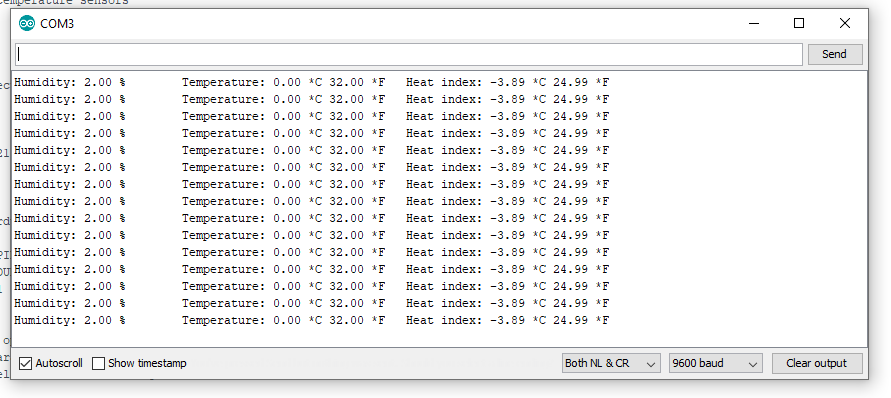


Figure - Output result for DHT11

### Light Intensity

#### Schematic Diagram

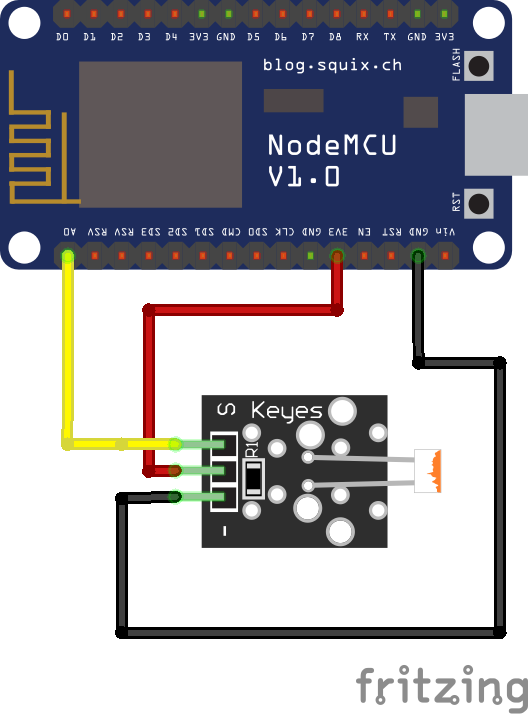


Figure - schematic Diagram for Light Intensity testing (Author’s work, 2019)

#### Code

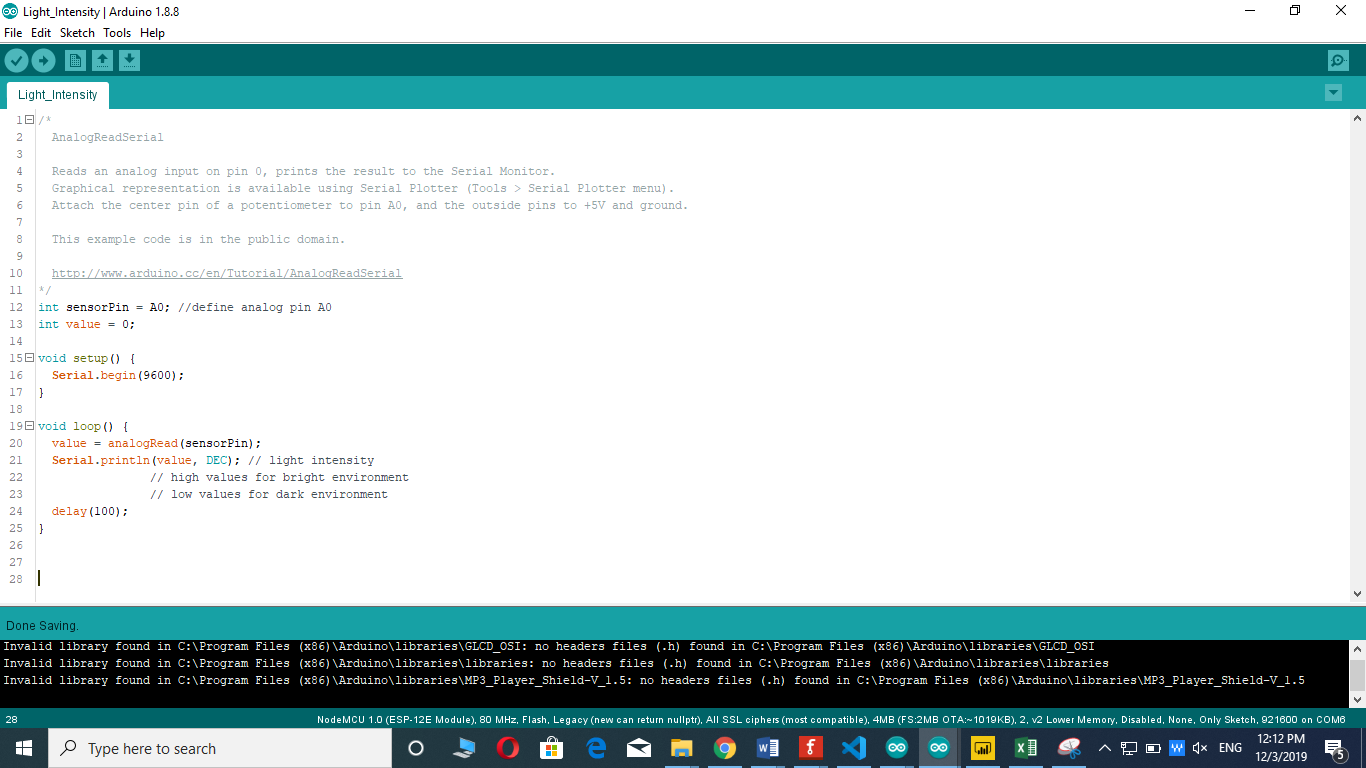


Figure - Code for Light Intensity testing

#### F:\pics\Iot\IMG-20191204-WA0030.jpgTesting Picture

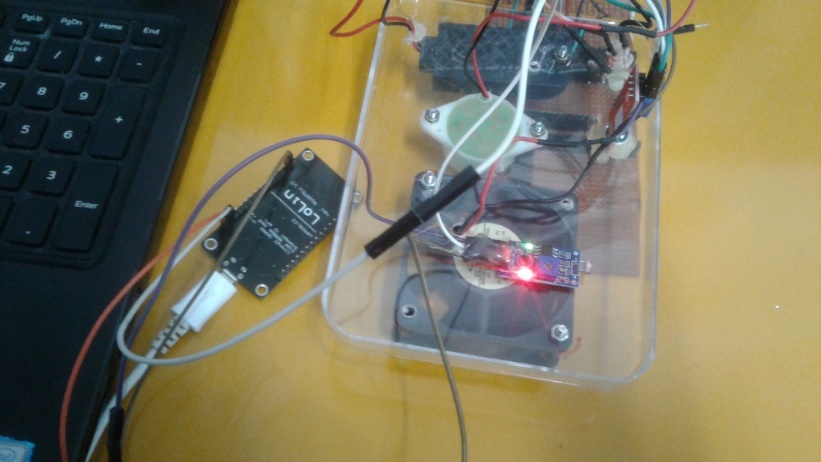


Figure - Testing picture of light intensity sensor

Figure - Testing picture of light intensity sensor

#### Output

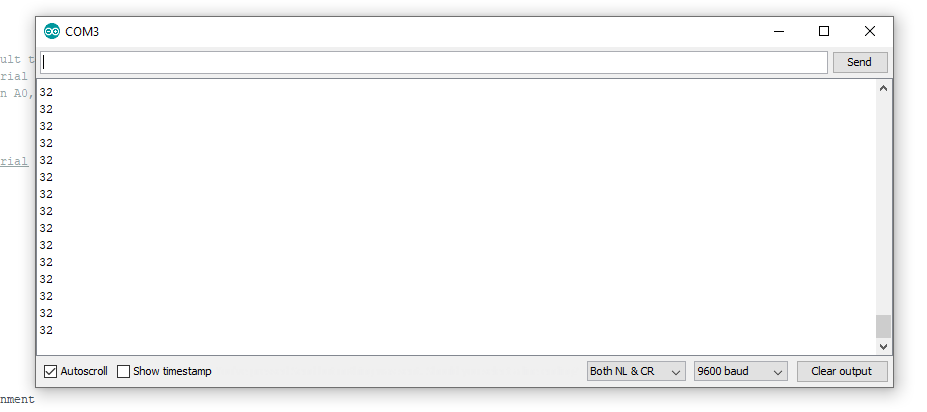


Figure - Output of light intensity sensor

### Sound Sensor

#### Schematic Diagram

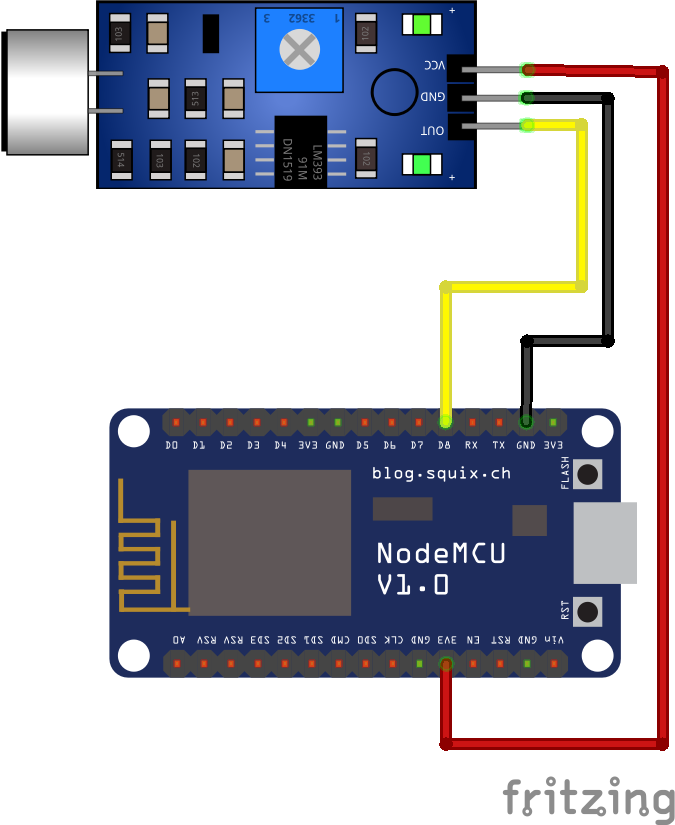


Figure - Schematic diagram for sound sensor

#### Code

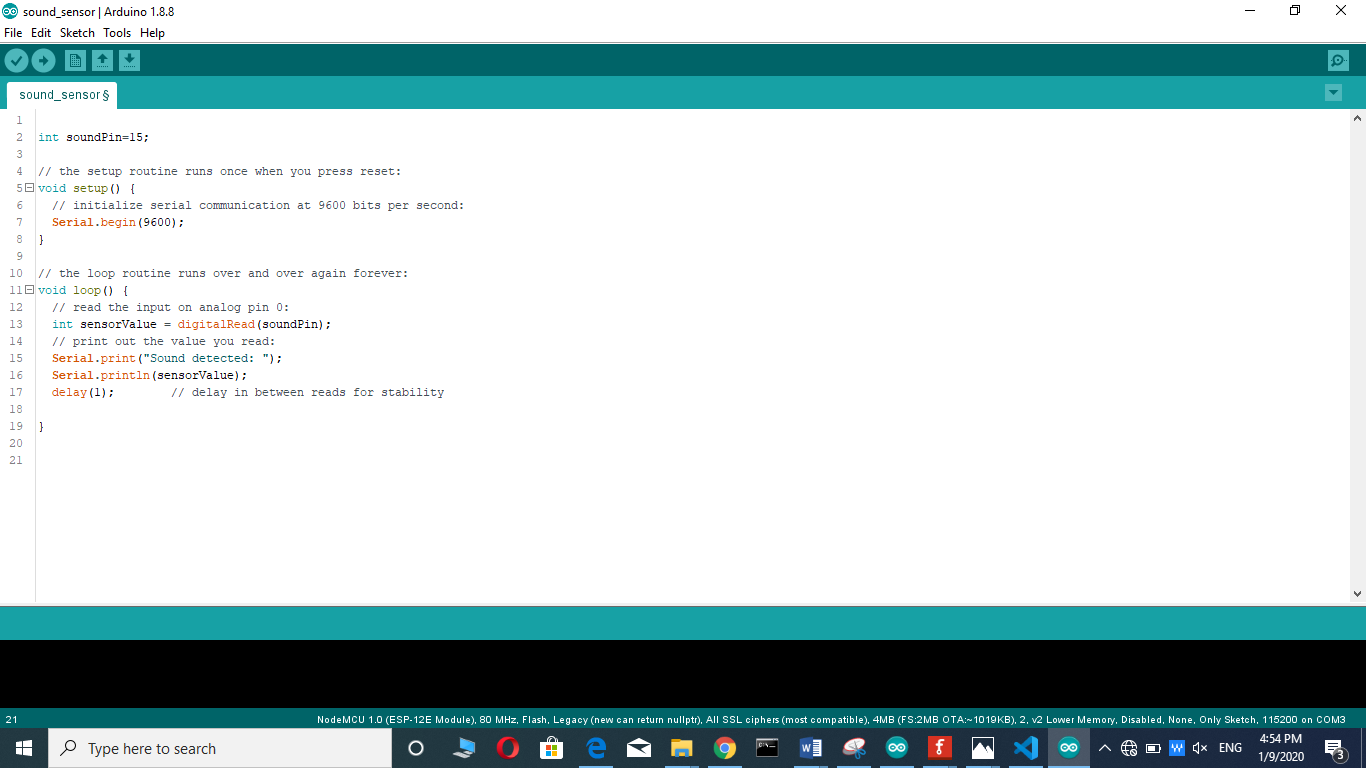


Figure - Code for Sound sensor

#### Testing Picture

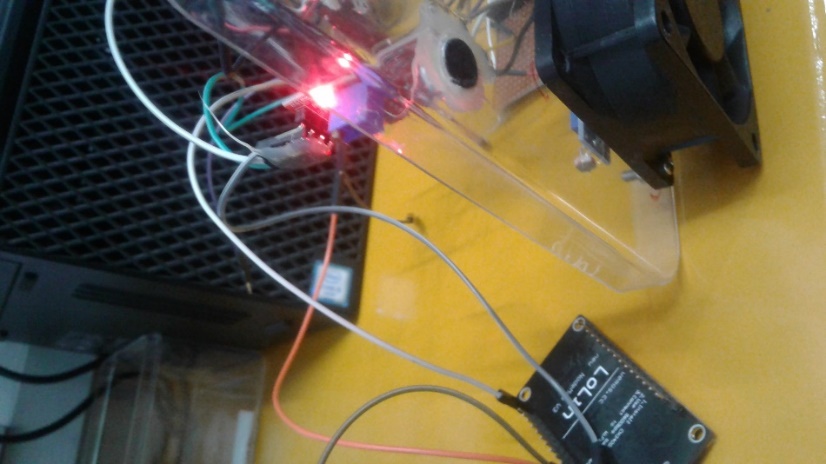
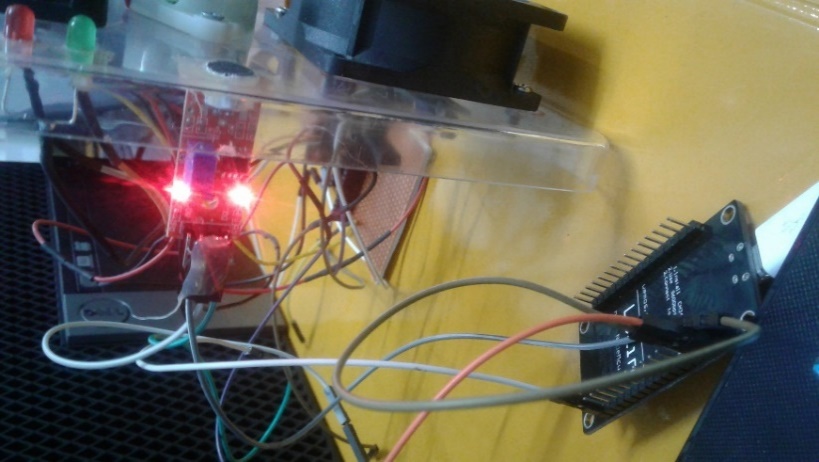


Figure - Testing Picture of Sound sensor

Figure - Testing Picture of Sound sensor

#### Output

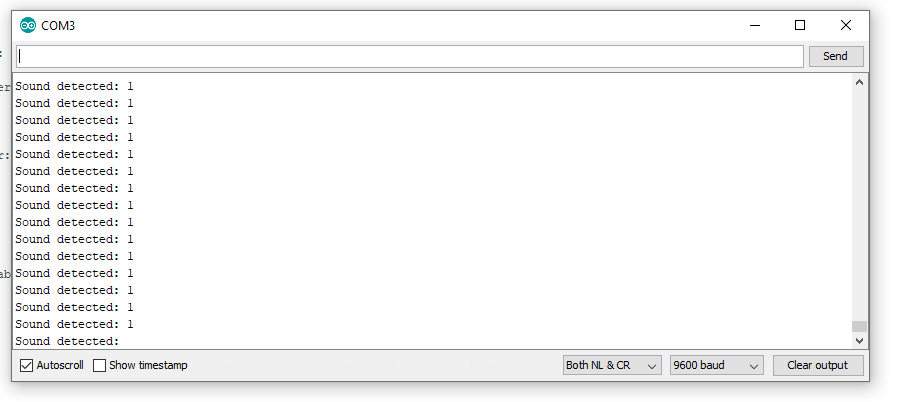


Figure - Output of Sound sensor testing

### LED

#### Schematic Diagram

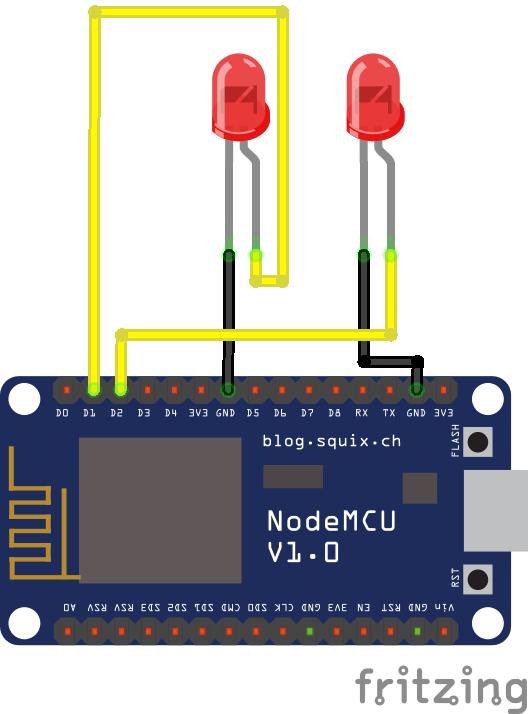


Figure - Schematic diagram for LED testing

#### Code

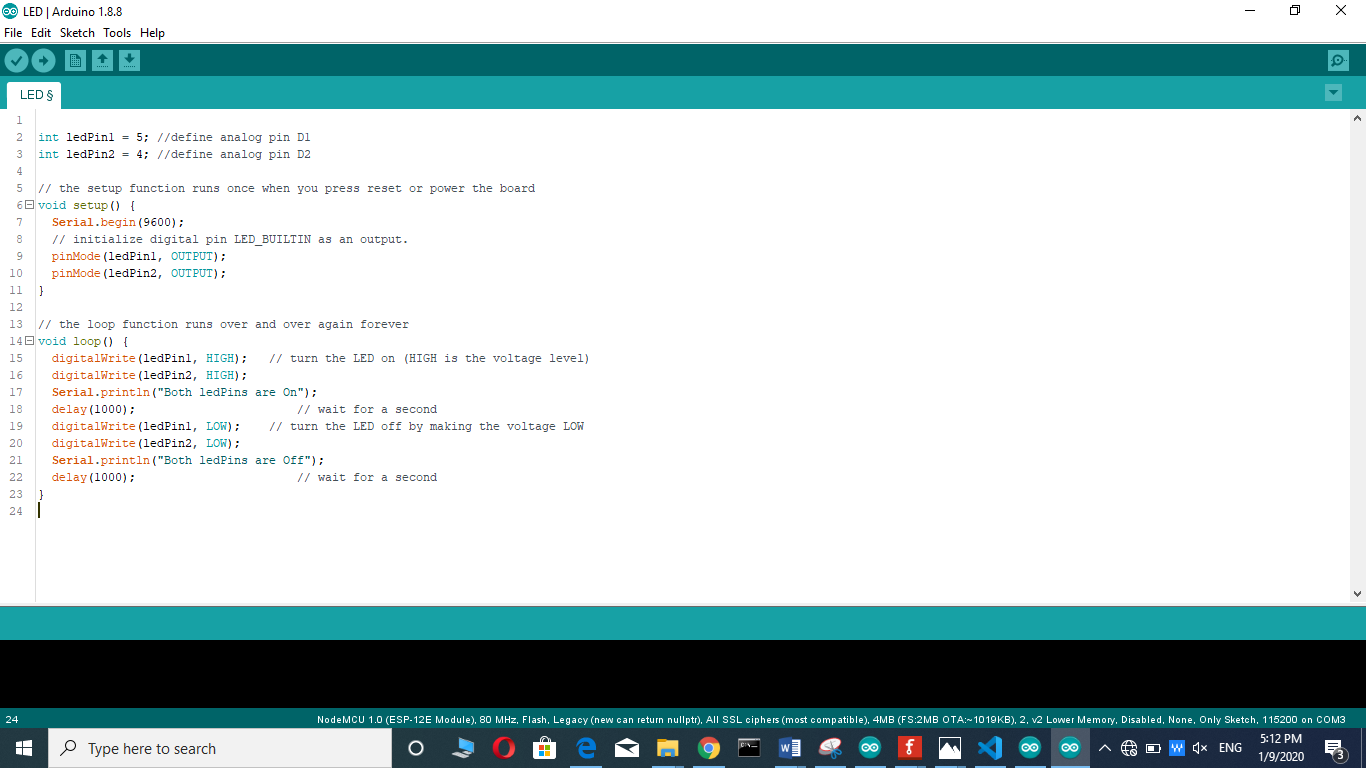


Figure - Code for LED testing code

#### F:\pics\Iot\IMG-20191203-WA0036.jpgTesting Picture

Figure - Testing picture of LED

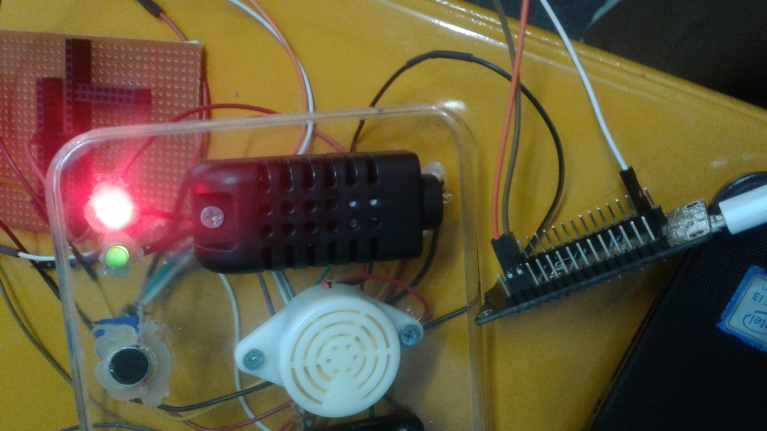


Figure - Testing picture of LED

Figure - Testing picture of LED

#### Output

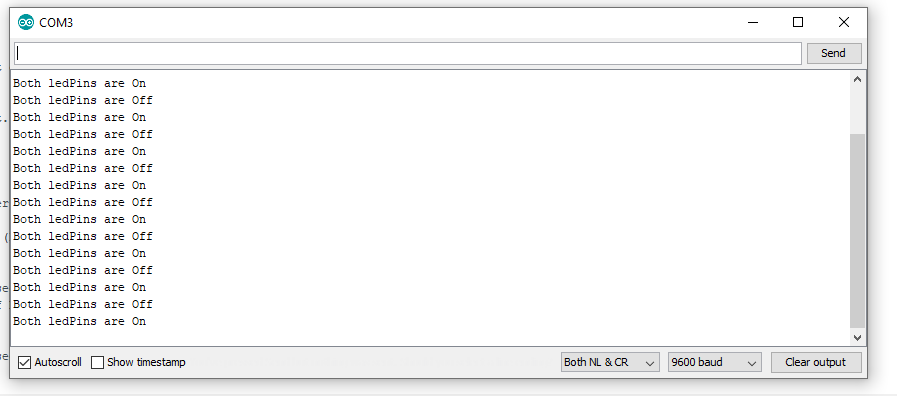


Figure - Output of LED testing

### Speaker

#### Schematic Diagram

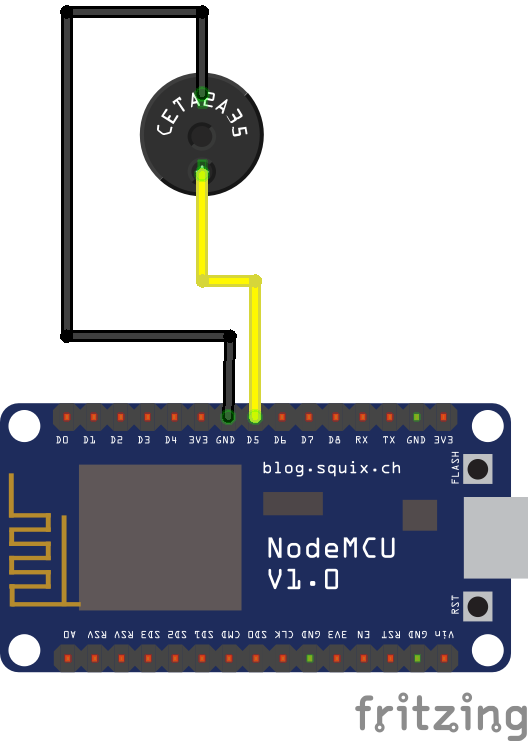


Figure - Schematic Diagram for Speaker (Author’s work, 2019)

#### Code

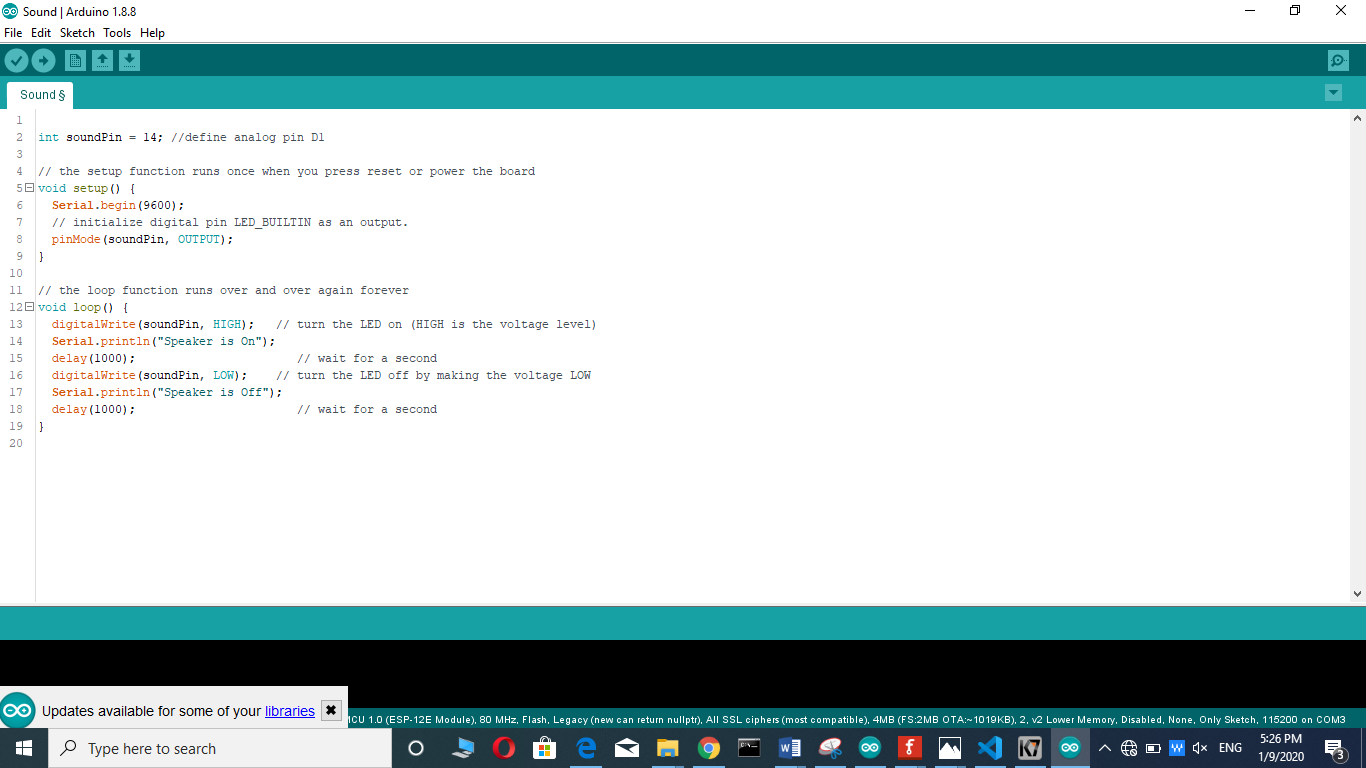


Figure - Code for Speaker testing

#### Testing Picture

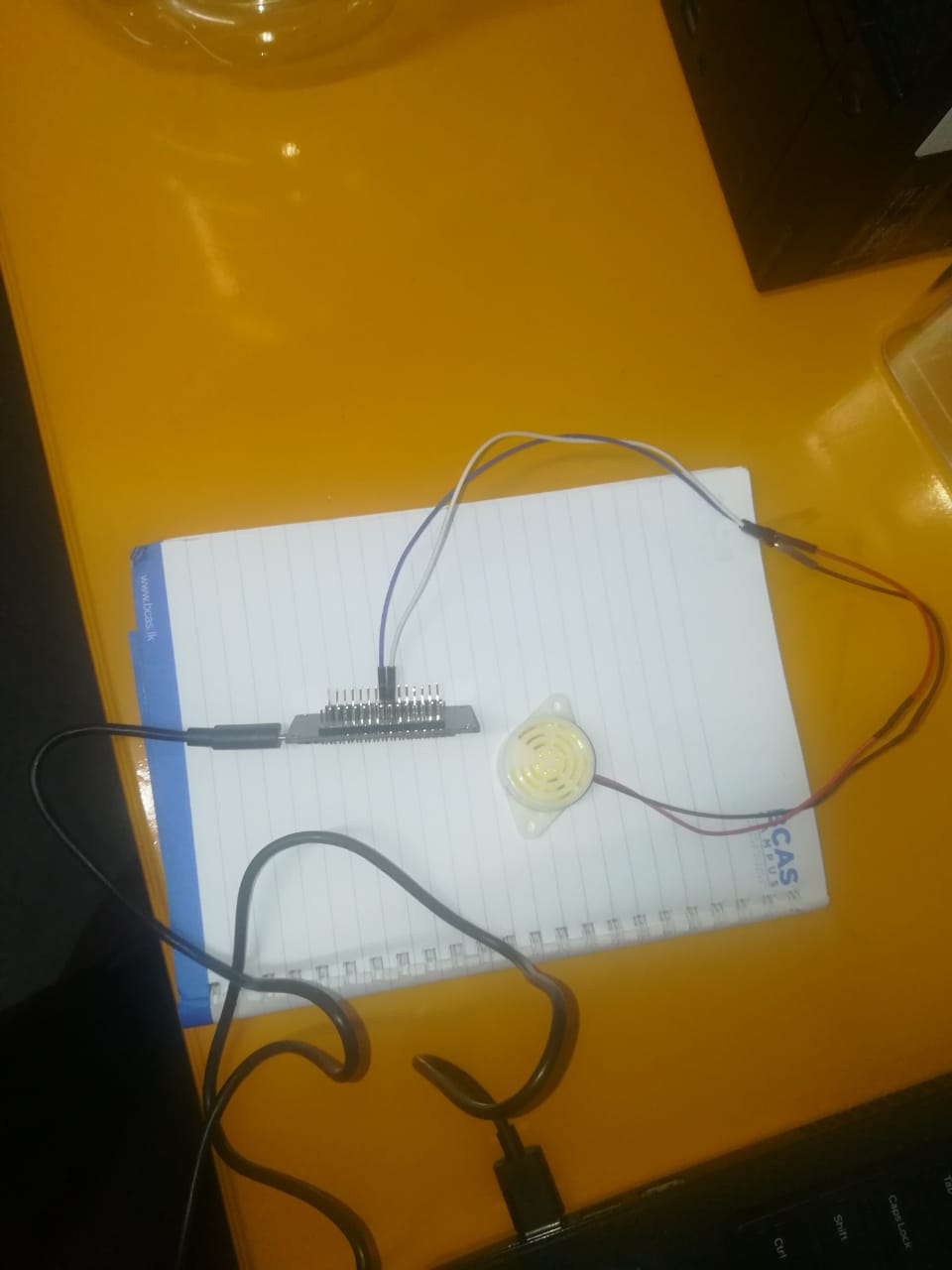


Figure - Testing picture of Speaker

Figure - Testing Picture of Speaker

#### Output

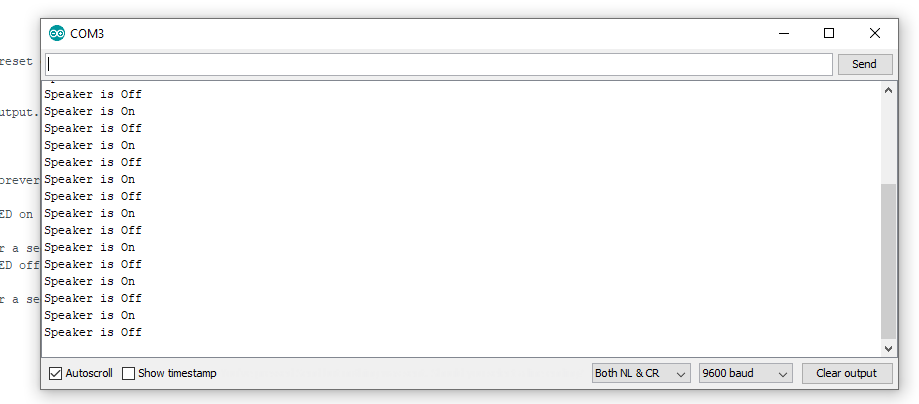


Figure - Output of Speaker testing

### Mini Fan

#### Schematic Diagram

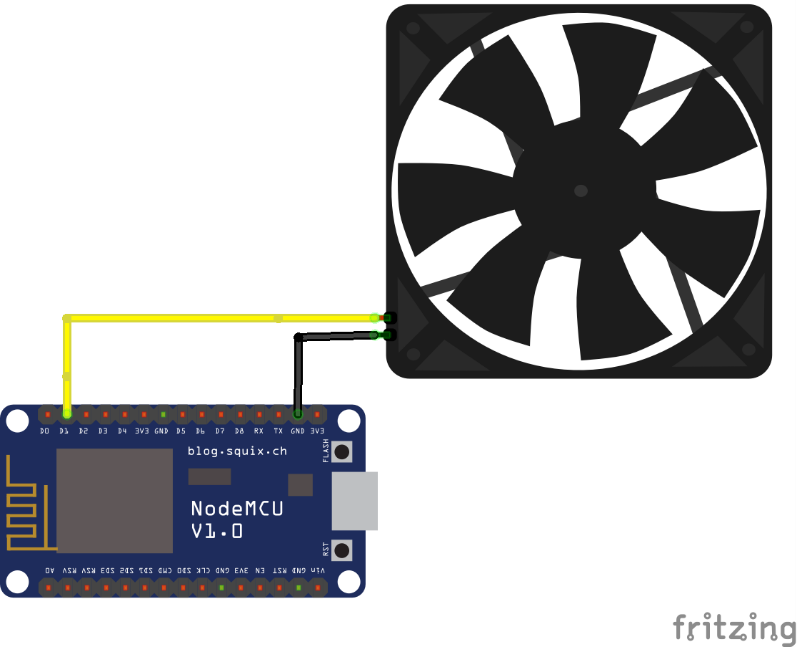


Figure - Schematic Diagram for fan (Author’s work, 2019)

#### Code

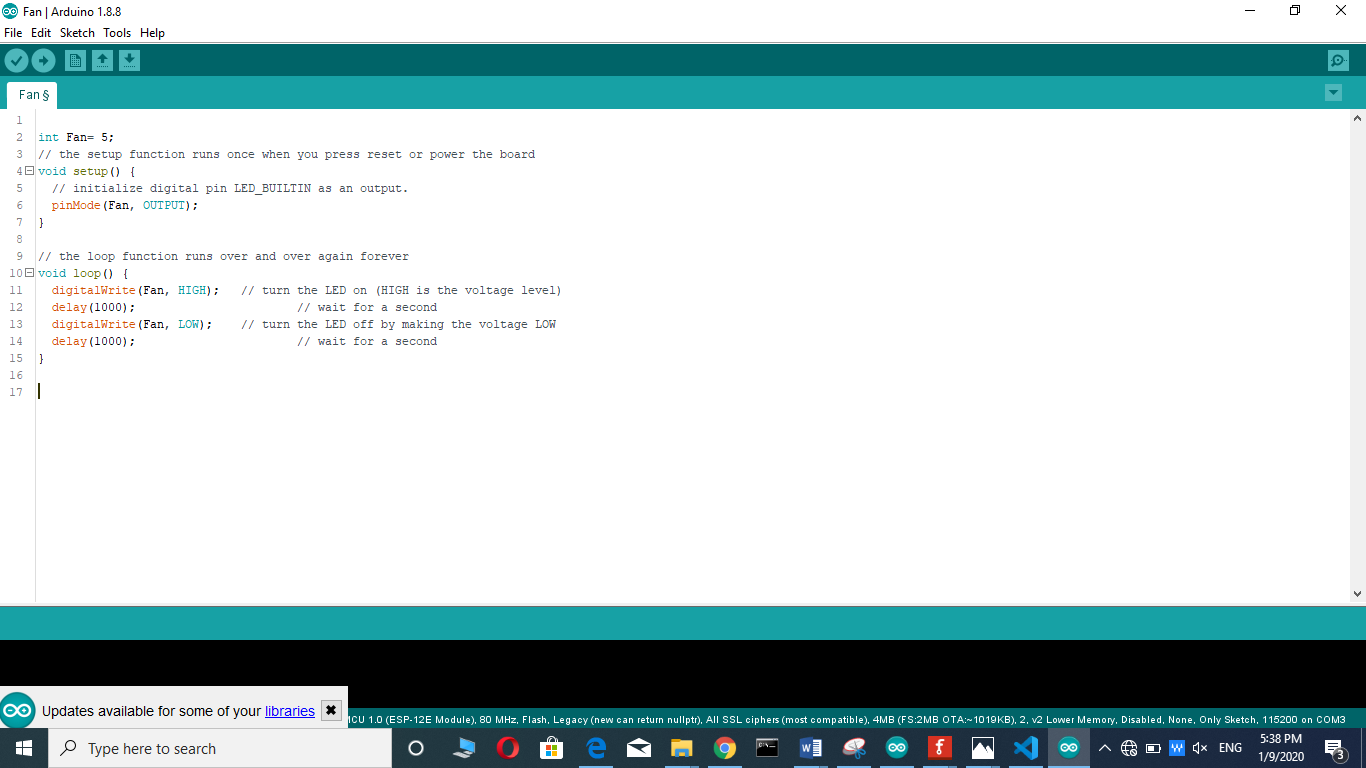


Figure - Code for testing fan

#### Testing Picture

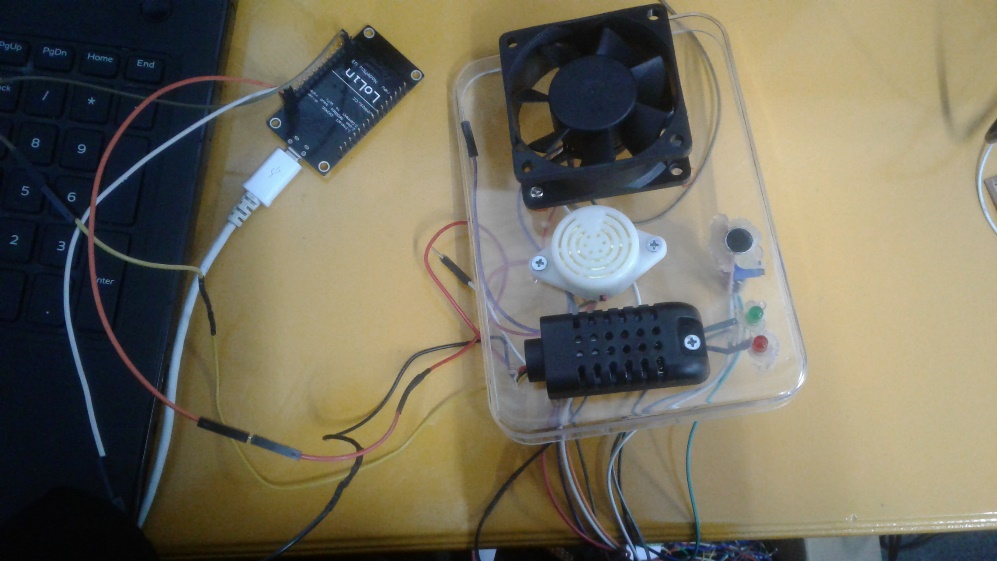


Figure - Testing Picture of Fan

## Limitation

* DHT11- “Temperature Range: 0°C to 50°C. Humidity Range: 20% to 90% Resolution: Temperature and Humidity both are 16-bit.” (components101.com, n.d.)
* Keys Photo Resister Module- “module consists of a photo resistor and a 10kΩ in-line resistor” (hobbyking.com, n.d.)
* Sound Sensor-
* Microphone sensitivity (1Khz): 52-48dB
* Microphone Impedance: 2.2KΩ
* Microphone Frequency: 16-20Khz
* Microphone S/N ratio: 54dB (robokits.co.in, n.d.)

# Future Development

In future we will include in our BMS with more exact parameters about baby. Such as viewing baby, moisture detection, baby body temperature and etc. we will set camera to capture baby from anywhere with live video. I will set as an output parents’ direct voice as live instead of playing music. Now we use mobile to control system. In future I will improve BMS to control by wrist watch and many devices. Instead of Battery I will set solar panel. In future I will set cot to swing when baby is crying. I will set AI for our BMS. From that we can get prediction about baby’s behavior from sensing data. This will give many solutions about baby with analyzing and visualizing automatically.

# Conclusion

This report is about IoT based baby monitoring system. This prototype system is develop to reduce workload of parents. First, I finalized IoT model for my BMS. In there we discussed, why we use those Sensors / Actuators, Connectivity and Data Visualization /Analysis. Then, we discussed schematic diagram for BMS. Then, we discussed about Model implementation. Then, discussed about system overall strengths, weakness and future development. Then, we discussed test and validation for every each sensors and actuators. Finally, we discussed about future development.

We learned from this IoT report applications of sensors and actuators, Boards, Gateways, Connectivity protocols, Cloud, IoT platforms and Data visualization, analysis and miming tools. From them I choose exact selection for my system. I deeply understand IoT working pattern.

# References

adafruit.com, 2020. *adafruit.com.* [Online]   
Available at: https://www.adafruit.com/product/3368

amazon.com, 2020. *amazon.com.* [Online]   
Available at: https://www.amazon.com/Phantom-YoYo-Arduino-Compatible-Sensor/dp/B00AF2GB1U

amazon.com, 2020. *amazon.com.* [Online]   
Available at: https://www.amazon.com/WINGONEER-Digital-Intensity-Resistor-Photoresistor/dp/B06XHGGG73

banggood.in, 2020. *banggood.in.* [Online]   
Available at: https://www.banggood.in/10pcs-White-High-Decibel-Alarm-SFM-27-DC6-24V-Continuous-Sounder-Buzzer-Speaker-p-1554817.html?rmmds=buy

components101.com, n.d. *components101.com.* [Online]   
Available at: https://components101.com/dht11-temperature-sensor  
[Accessed 3 12 2019].

electropeak.com, 2020. *electropeak.com.* [Online]   
Available at: https://electropeak.com/am2301-dht21-1

hobbyking.com, n.d. *hobbyking.com.* [Online]   
Available at: https://hobbyking.com/en\_us/keyes-ky-018-photo-resistor-module-for-kingduino-arduino.html  
[Accessed 03 12 2019].

robokits.co.in, n.d. *robokits.co.in.* [Online]   
Available at: https://robokits.co.in/sensors/sound/sound-sensor-module  
[Accessed 3 12 2019].

shutterstock.com, 2020. *shutterstock.com.* [Online]   
Available at: https://www.shutterstock.com/es/search/led%20bulb%20parts?image\_type=vector

# Appendix

## Code Samples

### **Temperature Sensor Testing Code**

<<#include "DHT.h"

#define DHTPIN 14 // what digital pin we're connected to

// Uncomment whatever type you're using!

//#define DHTTYPE DHT11 // DHT 11

//#define DHTTYPE DHT22 // DHT 22 (AM2302), AM2321

#define DHTTYPE DHT21 // DHT 21 (AM2301)

// Initialize DHT sensor.

// Note that older versions of this library took an optional third parameter to

// tweak the timings for faster processors. This parameter is no longer needed

// as the current DHT reading algorithm adjusts itself to work on faster procs.

DHT dht(DHTPIN, DHTTYPE);

void setup() {

Serial.begin(9600);

Serial.println("DHTxx test!");

dht.begin();

}

void loop() {

// Wait a few seconds between measurements.

delay(2000);

// Reading temperature or humidity takes about 250 milliseconds!

// Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)

float h = dht.readHumidity();

// Read temperature as Celsius (the default)

float t = dht.readTemperature();

// Read temperature as Fahrenheit (isFahrenheit = true)

float f = dht.readTemperature(true);

// Check if any reads failed and exit early (to try again).

if (isnan(h) || isnan(t) || isnan(f)) {

Serial.println("Failed to read from DHT sensor!");

return;

}

// Compute heat index in Fahrenheit (the default)

float hif = dht.computeHeatIndex(f, h);

// Compute heat index in Celsius (isFahreheit = false)

float hic = dht.computeHeatIndex(t, h, false);

Serial.print("Humidity: ");

Serial.print(h);

Serial.print(" %\t");

Serial.print("Temperature: ");

Serial.print(t);

Serial.print(" \*C ");

Serial.print(f);

Serial.print(" \*F\t");

Serial.print("Heat index: ");

Serial.print(hic);

Serial.print(" \*C ");

Serial.print(hif);

Serial.println(" \*F");

}>>

### **Light Intensity Sensor Testing Code**

<<int sensorPin = A0; //define analog pin A0

int value = 0;

void setup()

{

Serial.begin(9600);

}

void loop() {

value = analogRead(sensorPin);

Serial.println(value, DEC); // light intensity

// high values for bright environment

// low values for dark environment

delay(100);

}>>

### **Sound Sensor Testing Code**

<<int soundPin=15;

// the setup routine runs once when you press reset:

void setup() {

// initialize serial communication at 9600 bits per second:

Serial.begin(9600);

}

// the loop routine runs over and over again forever:

void loop() {

// read the input on analog pin 0:

int sensorValue = digitalRead(soundPin);

// print out the value you read:

Serial.print("Sound detected: ");

Serial.println(sensorValue);

delay(1); // delay in between reads for stability

}>>

### **LED light Testing Code**

<<int ledPin1 = 5; //define analog pin D1

int ledPin2 = 4; //define analog pin D2

// the setup function runs once when you press reset or power the board

void setup() {

Serial.begin(9600);

// initialize digital pin LED\_BUILTIN as an output.

pinMode(ledPin1, OUTPUT);

pinMode(ledPin2, OUTPUT);

}

// the loop function runs over and over again forever

void loop() {

digitalWrite(ledPin1, HIGH); // turn the LED on (HIGH is the voltage level)

digitalWrite(ledPin2, HIGH);

Serial.println("Both ledPins are On");

delay(1000); // wait for a second

digitalWrite(ledPin1, LOW); // turn the LED off by making the voltage LOW

digitalWrite(ledPin2, LOW);

Serial.println("Both ledPins are Off");

delay(1000); // wait for a second

}>>

### **Sound sensor Testing Code**

<<int soundPin = 14; //define analog pin D1

// the setup function runs once when you press reset or power the board

void setup() {

Serial.begin(9600);

// initialize digital pin LED\_BUILTIN as an output.

pinMode(soundPin, OUTPUT);

}

// the loop function runs over and over again forever

void loop() {

digitalWrite(soundPin, HIGH); // turn the LED on (HIGH is the voltage level)

Serial.println("Speaker is On");

delay(1000); // wait for a second

digitalWrite(soundPin, LOW); // turn the LED off by making the voltage LOW

Serial.println("Speaker is Off");

delay(1000); // wait for a second

}>>

### **Mini Fan Testing Code**

<<int Fan= 5;

// the setup function runs once when you press reset or power the board

void setup() {

// initialize digital pin LED\_BUILTIN as an output.

pinMode(Fan, OUTPUT);

}

// the loop function runs over and over again forever

void loop() {

digitalWrite(Fan, HIGH); // turn the LED on (HIGH is the voltage level)

delay(1000); // wait for a second

digitalWrite(Fan, LOW); // turn the LED off by making the voltage LOW

delay(1000); // wait for a second

}>>

## Ethical and privacy consideration

In my baby monitoring system I used temperature sensor, Light intensity sensor and sound sensor. From these sensors I am getting baby environment data. Such as temperature, light intensity of baby living environment and cry detection. I am not use or analyze human sensible data for my project.

## Baby Monitoring System

### Full System Code

<</\*

Title: IoT Based Baby Monitoring System

Developed by: Farha Imthiyaz

Sensors Used: DHT21, Keys photo resister (Light Intensity) and sound Sensor

Actuators Used: Mini fan, two LED bulbs and Speaker

IoT Platforms: Blynk

\*/

#include "DHT.h"

#define BLYNK\_PRINT Serial

#define EspSerial Serial3

#define ESP8266\_BAUD 9600

ESP8266 wifi(&EspSerial);

#include <ESP8266\_Lib.h>

#include <BlynkSimpleShieldEsp8266.h>

//DHT Sensor

#define pin\_DHT 2

#define DHTTYPE DHT21

float v\_TemperatureValue;

float v\_HumidityValue;

DHT dht(pin\_DHT, DHTTYPE);

//Light Intensity Sensor

#define pin\_lightintensity A0;

int v\_lightIntensityValue;

//Sound Sensor

#define pin\_sound 15;

int v\_soundValue;

//Fan

#define pin\_fan 13;

//Two LED bulbs

#define pin\_ledred 12;

#define pin\_ledgreen 14;

//Speaker

#define pin\_speaker 2;

//Blynk

char auth[] = "ZGT1tZr6p1GpNPZd-7-OojinYvAK3\_9I";

char ssid[] = "Dialog 4G 403";

char pass[] = "MyRabbRahman";

void dhtSensor()

{

float h = dht.readHumidity(); //Humidity

float t = dht.readTemperature(); //Temperature Celsius

float f = dht.readTemperature(true); //Temperature Fahrenheit

//Sensor Status Checking

if (isnan(h) || isnan(t) || isnan(f))

{

Serial.println("Failed to read from DHT sensor!");

return;

}

v\_TemperatureValue=t;

v\_HumidityValue=h;

String strTempCelsius = "Temperature: " + String(t) + "\*C";

String strHumidity= "Humidity :" + String(h);

Serial.println(strTempCelsius);

Serial.println(strHumidity);

}

void lightIntensity()

{

v\_lightIntensityValue = analogRead(pin\_lightintensity);

Serial.println(v\_lightIntensityValue, DEC); // light intensity

// high values for bright environment

// low values for dark environment

delay(1000);

}

void soundSensor()

{

v\_soundValue = digitalRead(pin\_sound);

// print out the value you read:

Serial.print("Sound detected: ");

Serial.println(v\_soundValue);

delay(1000); // delay in between reads for stability

}

BlynkTimer timer;

void myTimerEvent()

{

dhtSensor();

lightIntensity();

soundSensor();

//V1 Pin Temperature Sensor Value

Blynk.virtualWrite(V1, v\_TemperatureValue);

//V2 Light Intensity Sensor Value

Blynk.virtualWrite(V2, v\_lightIntensityValue);

//V3 Sound Sensor Value

Blynk.virtualWrite(V3, v\_soundValue);

}

//from V4 button we are controlling fan

BLYNK\_WRITE(V4)

{

int pinValueV4 = param.asInt(); // assigning incoming value from pin V4 to a variable

// You can also use:

// String i = param.asStr();

// double d = param.asDouble();

Serial.print("V4 Button value is: ");

Serial.println(pinValueV4);

if(pinValueV4==1)

{

Serial.println("Fan On");

digitalWrite(pin\_fan, HIGH);

}

if(pinValueV4==0)

{

Serial.println("Fan Off");

digitalWrite(pin\_fan, LOW);

}

}

//from V5 button we are controlling two LEDs

BLYNK\_WRITE(V5)

{

int pinValueV5 = param.asInt(); // assigning incoming value from pin V5 to a variable

// You can also use:

// String i = param.asStr();

// double d = param.asDouble();

Serial.print("V5 Button value is: ");

Serial.println(pinValueV5);

if(pinValueV5==1)

{

Serial.println("Both LED On");

digitalWrite(pin\_ledred, HIGH);

digitalWrite(pin\_ledgreen, HIGH);

}

if(pinValueV5==0)

{

Serial.println("Both LED Off");

digitalWrite(pin\_ledred, LOW);

digitalWrite(pin\_ledgreen, LOW);

}

}

//from V6 button we are controlling Speaker

BLYNK\_WRITE(V6)

{

int pinValueV6 = param.asInt(); // assigning incoming value from pin V5 to a variable

// You can also use:

// String i = param.asStr();

// double d = param.asDouble();

Serial.print("V6 Button value is: ");

Serial.println(pinValueV6);

if(pinValueV6==1)

{

Serial.println("Speaker On");

digitalWrite(pin\_speaker, HIGH);

}

if(pinValueV6==0)

{

Serial.println("Speaker Off");

digitalWrite(pin\_speaker, LOW);

}

}

void setup()

{

Serial.begin(9600);

pinMode(pin\_fan, OUTPUT);

pinMode(pin\_ledred,OUTPUT);

pinMode(pin\_ledgreen,OUTPUT);

pinMode(pin\_speaker,OUTPUT);

dht.begin();

EspSerial.begin(ESP8266\_BAUD);

delay(10);

Blynk.begin(auth, wifi, ssid, pass);

timer.setInterval(1000L, myTimerEvent);

}

void loop()

{

Blynk.run();

timer.run(); // Initiates BlynkTimer

}>>

### System working method

Above system coding we coded in Arduino IDE. We connected this coding to NodeMCU32. Then we uploaded code to NodeMCU32. In, NodeMCU32 we connected our sensors and actuators. From, Blynk platform we control our system. We got every sensor data to Blynk super chart controller. From that, we can visualize sensor data. We actuate all the actuators using Blynk buttons. Finally, we analyze all the sensor data in PowerBI.

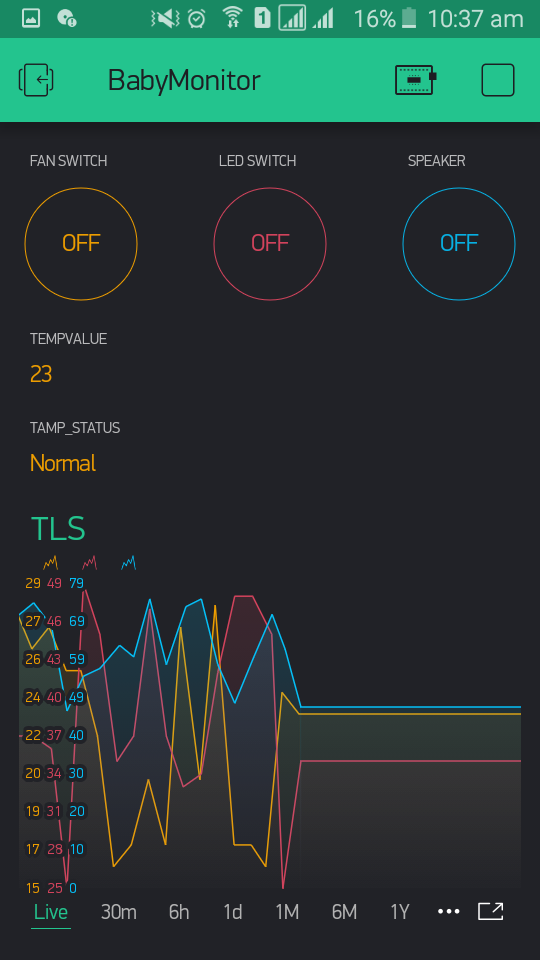
In this Blynk platform there is a super chart controller. From that, we can get temperature sensor data in yellow color, light intensity sensor data in pink color and sound sensor data in blue color. From these visualization we can get real live sensor data changes. There are two value displays. One is displaying real temperature amount and other one displaying is temperature status.

Figure - Blynk Dashboard for BMS

We can control our actuators by these buttons. So, we have three buttons. Yellow color button is fan control switch. Pink color button is LED control switch. Blue color switch is Speaker control switch.

According to Super chart data we can get data and control actuators by buttons. We get temperature data and control fan by FANSWITCH. We get light intensity data and control LED by LEDSWITCH. We get sound data and control speaker by SPEAKER switch. From this Blynk platform we can control our system in anywhere.

### Important Code Section/Logics

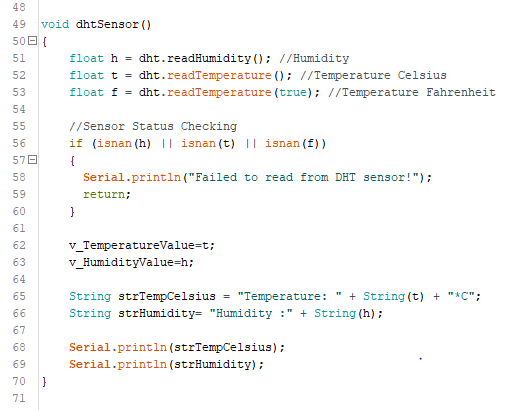
This is a temperature sensor function (dhtsensor()). In here we are getting temperature and humidity data and printing.

Figure - code for temperature sensor function

This is a light intensity sensor (lightIntensity()) and sound sensor (soundSensor()) functions. In here we are getting light intensity and sound sensor data and printing.

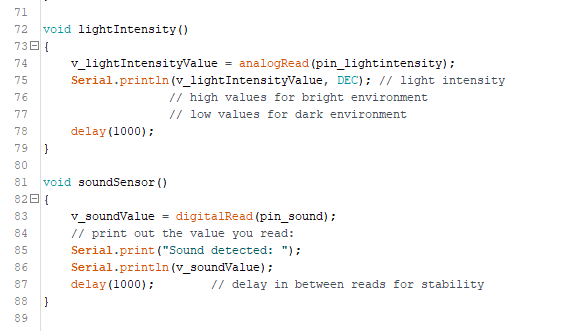


Figure - code for light intensity and sound sensor functions

In here we are sending temperature, light intensity and sound data to the Blynk platform. We can visualize line chart from these data. In here, we inserted above mentioned sensor functions also.

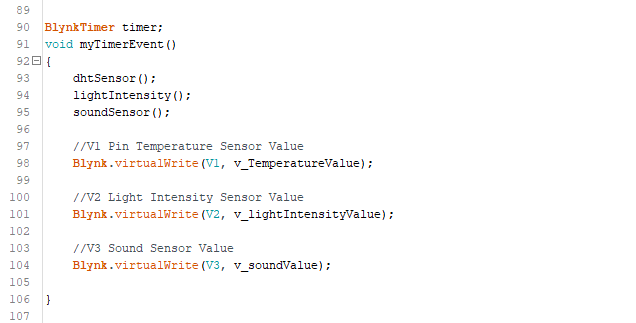


Figure - code for sending data from sensors to Blynk

In here, we are receiving data to board. From Blynk we are controlling actuators. We are controlling fan by V4 button. When V4 is pressed (High/ 1) meant fan is on. If not fan is off.

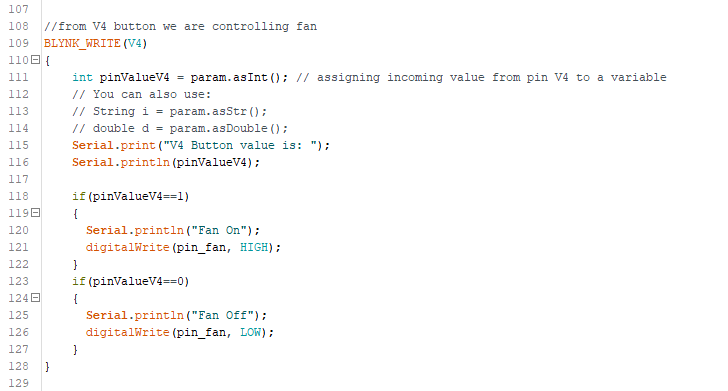


Figure - code for Control from Blynk to fan

We are controlling two LEDs by V5 button. When V5 is pressed (High/ 1) meant LEDs are on. If not LEDs are off.

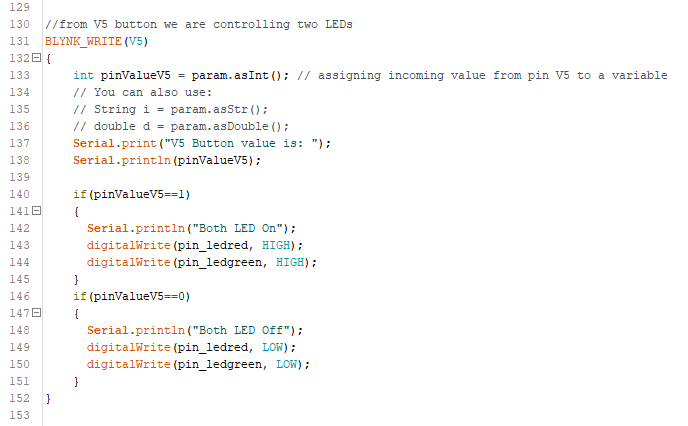


Figure - code for Control from Blynk to LED

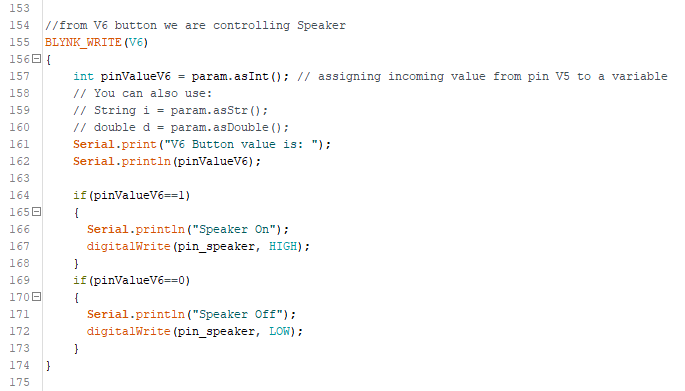
We are controlling speaker by V6 button. When V6 is pressed (High/ 1) meant Speaker is on. If not Speaker is off.

Figure - code for Control from Blynk to Speaker

In here, setup function we define pins and it’s using way that meant input or output. In here, mainly we include Blynk begin also. Then, we include loop function to run whole these functions.

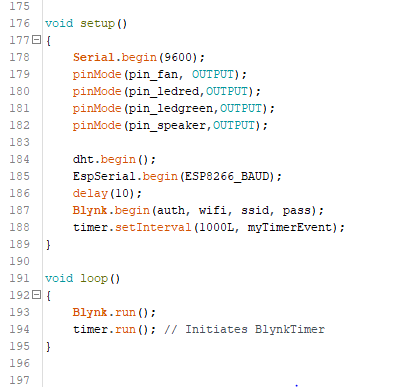


Figure - Code for Blynk setup and loop functions

### System Implementing Pictures and whole testing Pictures

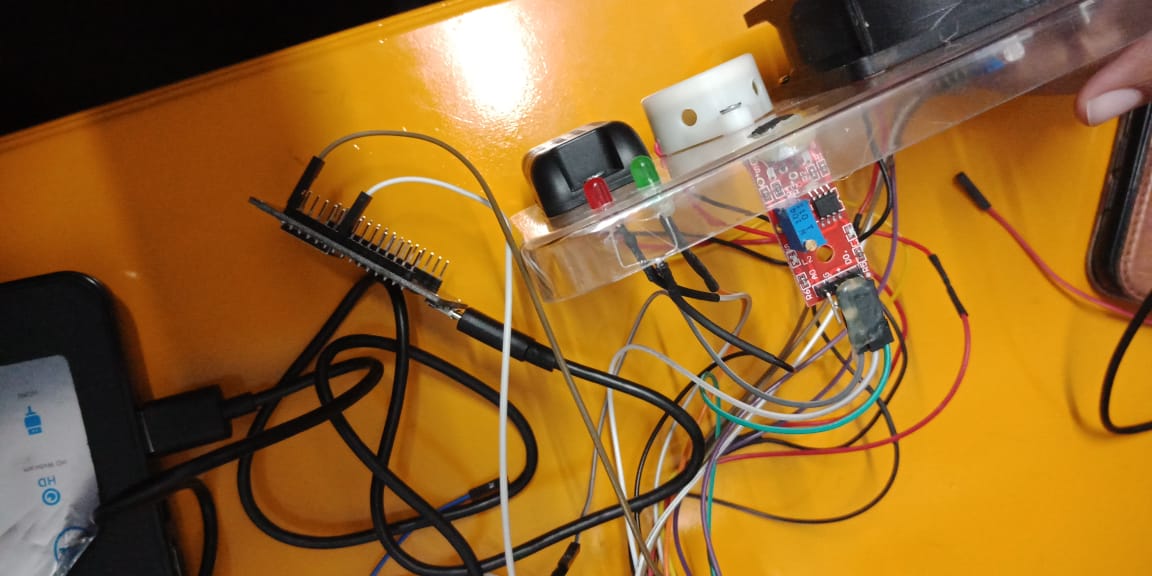
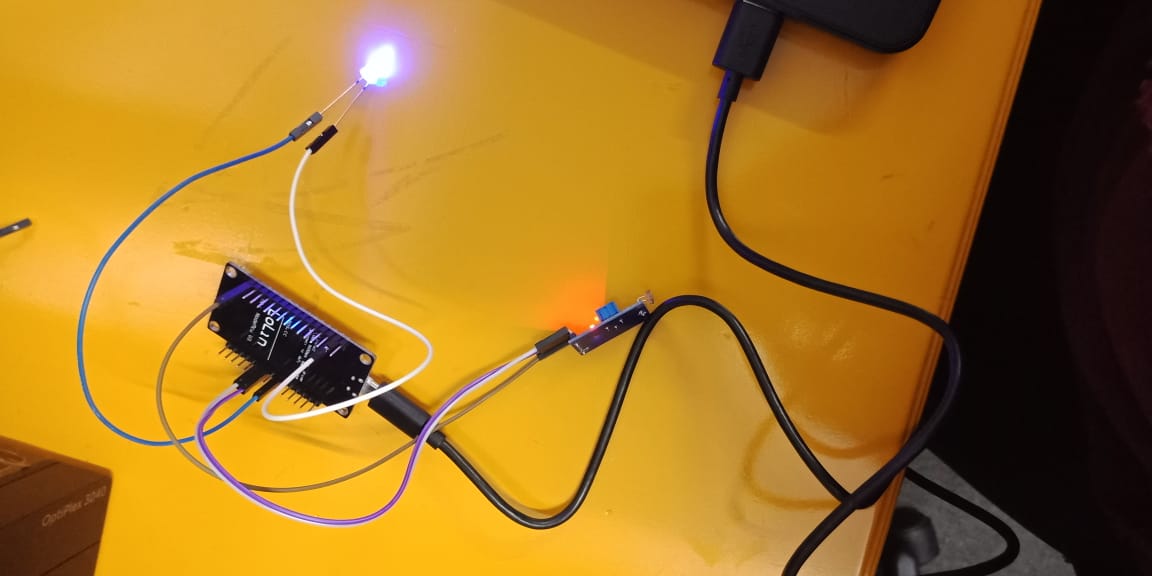


Figure - Picture of Testing and Implementing

Figure - Picture of Testing and Implementing



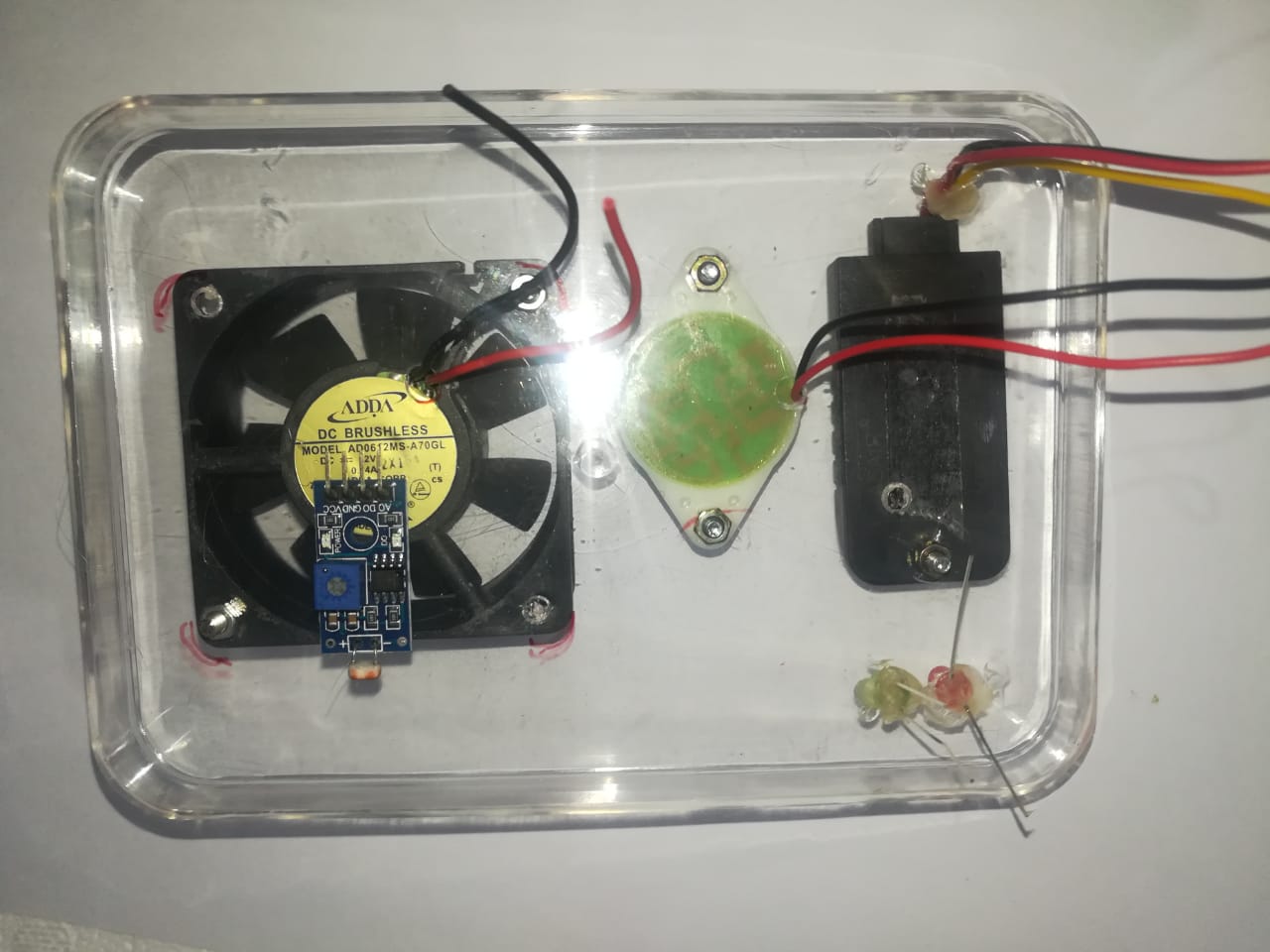


Figure - Picture of Testing and Implementing

Figure - Picture of Implementing

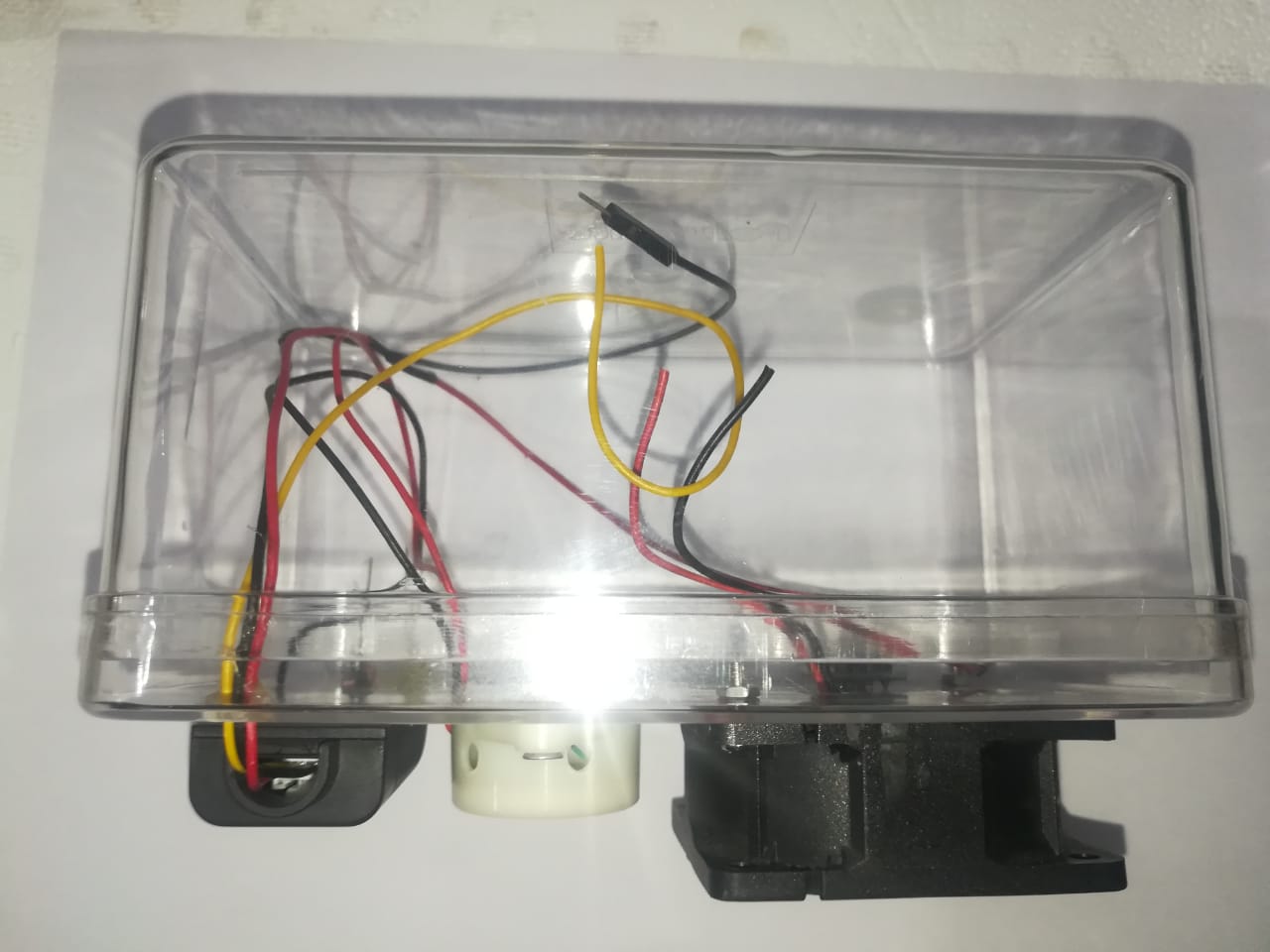


Figure - Picture of Implementing

Figure - Picture of Implementing

Figure - Picture of Implementing

Figure - Picture of Implementing



Figure - Picture of Implementing

Figure - Picture of Testing and Implementing



Figure - Picture of Implementing

Figure - Picture of Implementing



Figure - Picture of Implementing

Figure - Picture of Implementing

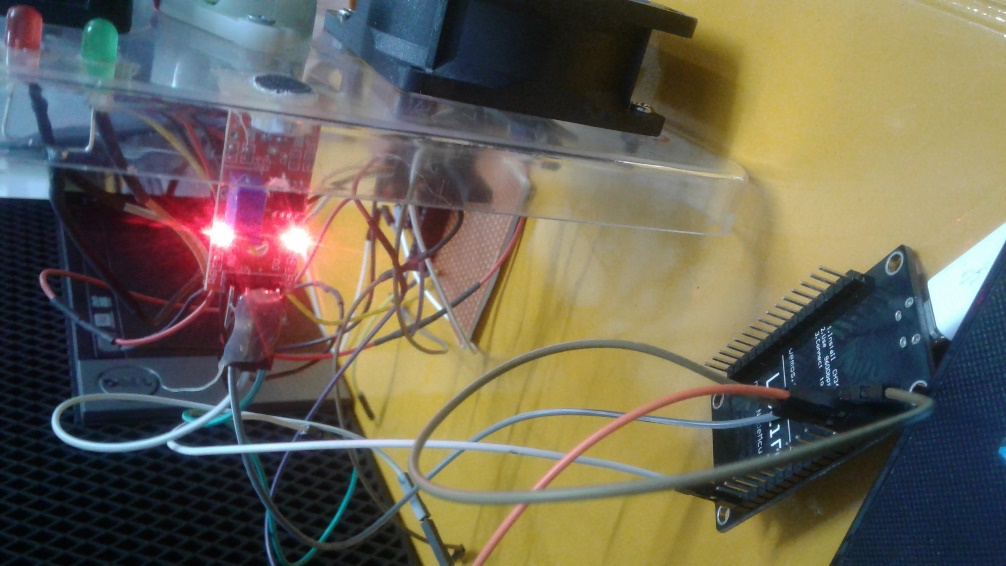




Figure - Picture of Testing and Implementing

Figure - Picture of Testing and Implementing



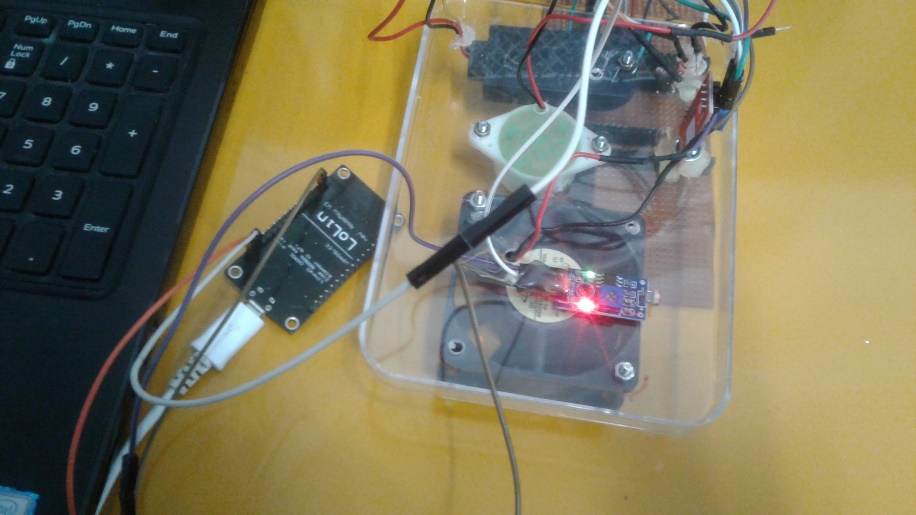
Figure - Picture of Testing and Implementing



Figure - Picture of Testing and Implementing

Figure - Picture of Testing and Implementing

## Presentation Slides

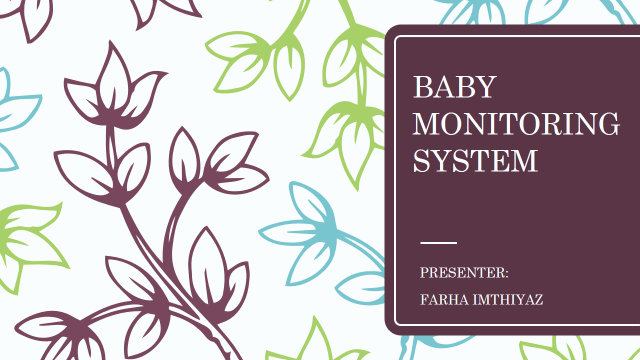


Figure - Beginning Slide

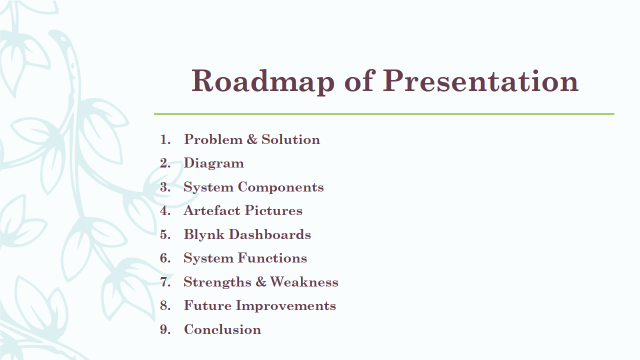


Figure - Contents Slide

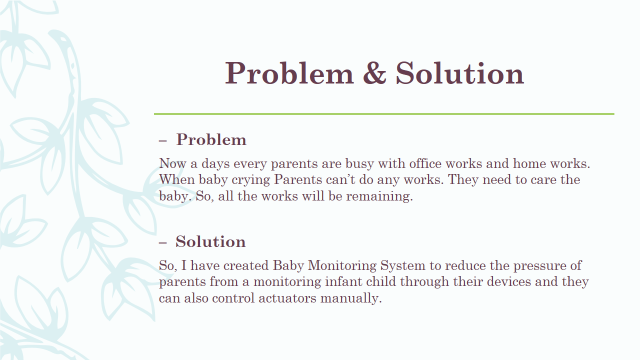


Figure - Problem & Solution Slide

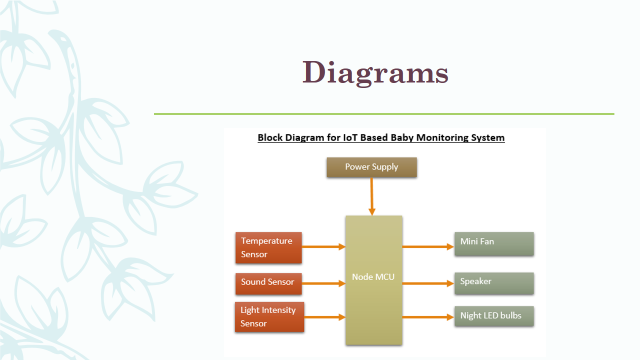


Figure - Block Diagram for BMS

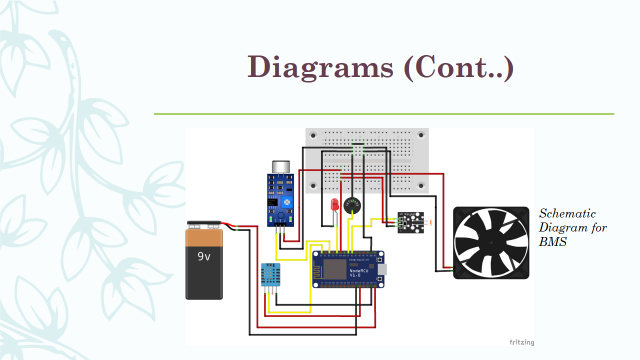


Figure - Schematic Diagram for BMS

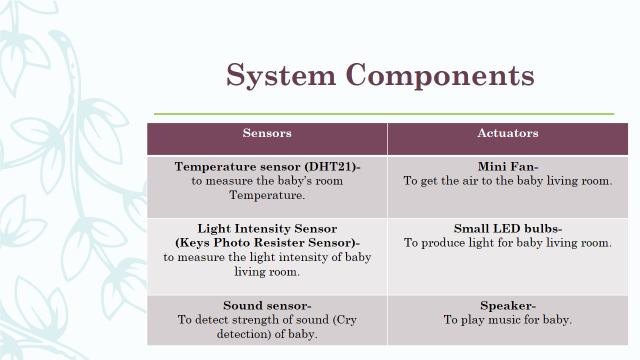


Figure - Slide of System Components

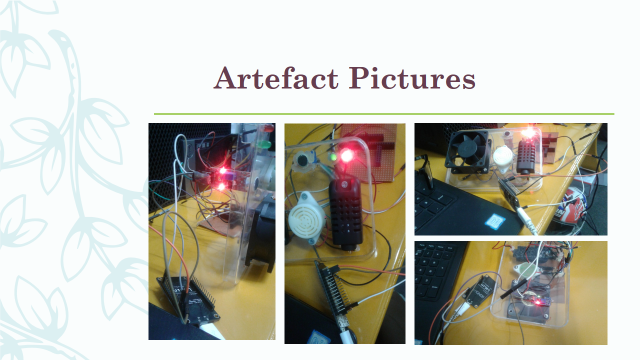


Figure - Slide of Artifact Pictures

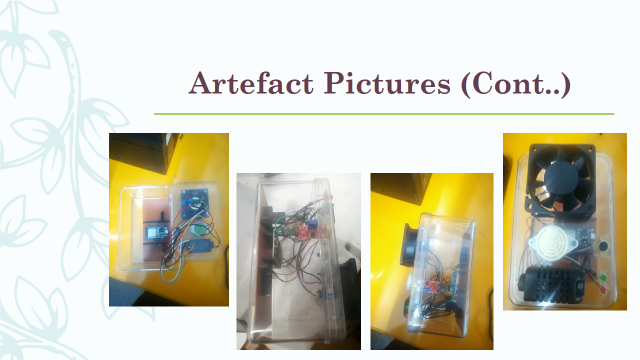


Figure - Slide of Artifact Pictures

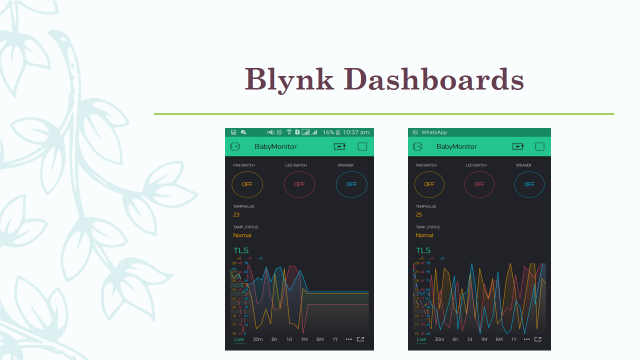


Figure - Slide of Blynk Dashboards

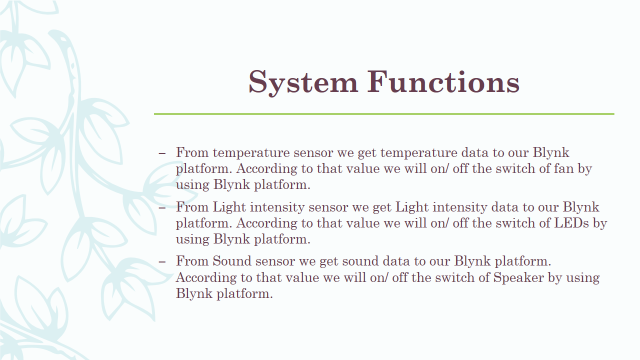


Figure - Slide of System Functions

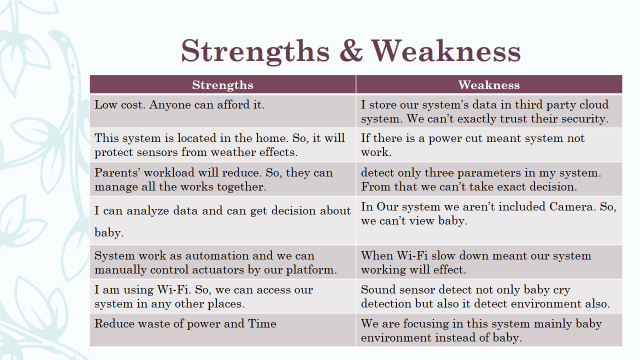


Figure - Slide of Strengths & weakness



Figure - Slide of Future Improvement

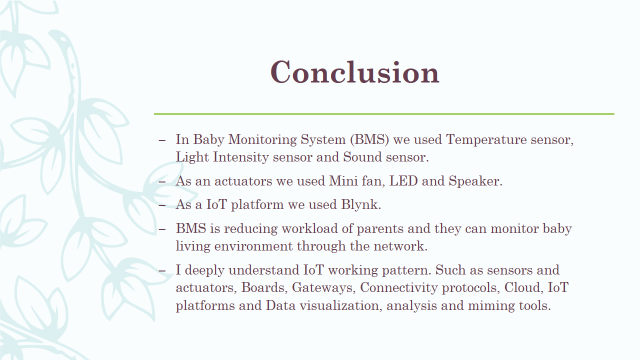


Figure - Slide of Conclusion



Figure - Ending Slide