## Table of Contents

**Acknowledgement…………….………………………………………………………………….ii**

**Company Profile………………………………………………………………………………....iii**

**Abstract……………………………………………………………………….…………………..iv**

**List of Figures ………………………………………..…………………………………………..v**

**List of Tables ……………………………………………………………………………………..vi**

**Chapter 1: Introduction** **to Internet Of Things**  **3-5**

* 1. What is Internet Of Things ? 3
  2. How does it work ? 4
  3. The Future of IoT 4

**Chapter 2: Introduction to Microcontrollers**  **6-7**

* 1. What is a Microcontroller ?6

2.2 Features Of Microcontrollers 6

**Chapter 3: Technologies Used 8-14**

* 1. Hardware 8

2.1.1 Arduino UNO Board 8

2.1.2 NODE MCU ESP8266 9

2.1.3 External Modules Used 10

2.2 Software 13

2.2.1 Arduino IDE 13

2.2.2 Cloud MQTT 14

**Chapter 4: Project Work**  **16-18**

* 1. Introduction to Project 16
     1. Criteria 16
     2. Methodology 17
     3. Block Diagram 17

2.2 Introduction to Other Projects  18

**Chapter 5: Result And Analysis**  **19-21**

**Conclusion**  22

**Future Scope**  23

**Appendix**  **24-32**

**References**  33

**CHAPTER-1**

**INTRODUCTION TO INTERNET OF THINGS**

* 1. **WHAT IS INTERNET OF THINGS ?**

The Internet of Things (IoT) is the network of physical devices, vehicles, home appliances, and other items embedded with electronics, software, sensors, actuators, and connectivity which enables these things to connect and exchange data, creating opportunities for more direct integration of the physical world into computer-based systems, resulting in efficiency improvements, economic benefits, and reduced human exertions. The number of IoT devices increased 31% year-over-year to 8.4 billion in the year 2017 and it is estimated that there will be 30 billion devices by 2020. The global market value of IoT is projected to reach $7.1 trillion by 2020. IoT involves extending Internet connectivity beyond standard devices, such as desktops, laptops, smartphones and tablets, to any range of traditionally dumb or non-internet-enabled physical devices and everyday objects. Embedded with technology, these devices can communicate and interact over the Internet, and they can be remotely monitored and controlled. With the arrival of driverless vehicles, a branch of IoT, i.e. the Internet of Vehicle starts to gain more attention. The term "Internet of things" was likely coined by Kevin Ashton of Procter & Gamble, later MIT's Auto-ID Center, in 1999, though he prefers the phrase "Internet for things". At that point, he viewed Radio-frequency identification (RFID) as essential to the Internet of things, which would allow computers to manage all individual things. A research article mentioning the Internet of Things was submitted to the conference for Nordic Researchers in Logistics, Norway, in June 2002, which was preceded by an article published in Finnish in January 2002.

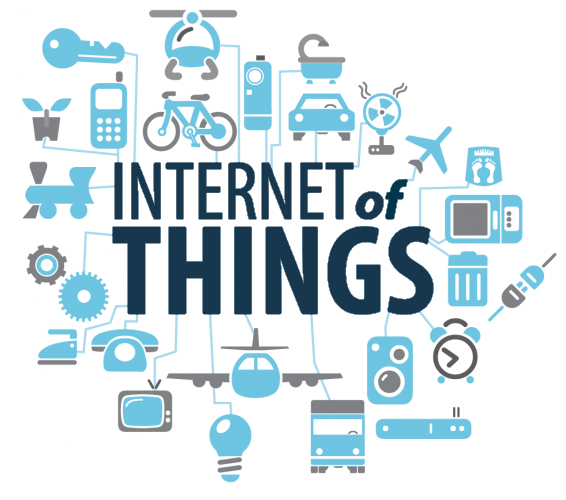


Figure 1.1 : Internet Of Things

Defining the Internet of Things as "simply the point in time when more 'things or objects' were connected to the Internet than people", Cisco Systems estimated that IoT was "born" between 2008 and 2009, with the things/people ratio growing from 0.08 in 2003 to 1.84 in 2010.

* 1. **HOW DOES IT WORK ?**

Devices and objects with built in sensors are connected to an Internet of Things platform, which integrates data from the different devices and applies analytics to share the most valuable information with applications built to address specific needs These powerful IoT platforms can pinpoint exactly what information is useful and what can safely be ignored. This information can be used to detect patterns, make recommendations, and detect possible problems before they occur.

For example, if I own a car manufacturing business, I might want to know which optional components (leather seats or alloy wheels, for example) are the most popular. Using Internet of Things technology, I can:

* Use sensors to detect which areas in a showroom are the most popular, and where customers linger longest.
* Drill down into the available sales data to identify which components are selling fastest.
* Automatically align sales data with supply, so that popular items don’t go out of stock.
* The information picked up by connected devices enables me to make smart decisions about which components to stock up on, based on real-time information, which helps me save time and money.
* With the insight provided by advanced analytics comes the power to make processes more efficient. Smart objects and systems mean you can automate certain tasks, particularly when these are repetitive, mundane, time-consuming or even dangerous.

**1.3**  **THE FUTURE OF IOT :**

Internet of Things include connected security systems, thermostats, cars, electronic appliances, lights in household and commercial environments, alarm clocks, speaker systems, vending machines and more. Businesses can leverage IoT applications to automate safety tasks (for example, notify authorities when a fire extinguisher in the building is blocked) to performing real-world A/B testing using networked cameras and sensors to detect how customers engage with products. The Internet of Things extends internet connectivity beyond traditional devices like desktop and laptop computers, smartphones and tablets to a diverse range of devices and everyday things that utilize embedded technology to communicate and interact with the external environment, all via the Internet.As far as the reach of the Internet of Things, there are more than 12 billion devices that can currently connect to the Internet, and researchers at IDC estimate that by 2020 there will be 26 times more connected things than people.

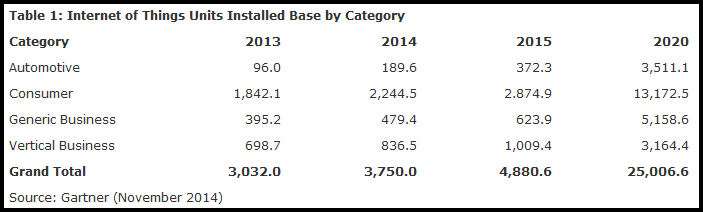


Figure 1.2 : Future Scope Of IoT

According to Gartner, consumer applications will drive the number of connected things, while enterprise will account for most of the revenue. IoT adoption is growing, with manufacturing and utilities estimated to have the largest installed base of Things by 2020.

**CHAPTER-2**

**INTRODUCTION TO MICROCONTROLLERS**

* 1. **WHAT IS A MICROCONTROLLER ?**

A microcontroller is a compact integrated circuit designed to govern a specific operation in an embedded system. A typical microcontroller includes a processor, memory and input/output (I/O) peripherals on a single chip. Sometimes referred to as an embedded controller or microcontroller unit (MCU), microcontrollers are found in vehicles, robots, office machines, medical devices, mobile radio transceivers, vending machines and home appliances among other devices. Microcontrollers are used in automatically controlled products and devices, such as automobile engine control systems, implantable medical devices, remote controls, office machines, appliances, power tools, toys and other embedded systems. By reducing the size and cost compared to a design that uses a separate microprocessor, memory, and input/output devices, microcontrollers make it economical to digitally control even more devices and processes.They will generally have the ability to retain functionality while waiting for an event such as a button press or other interrupt; power consumption while sleeping (CPU clock and most peripherals off) may be just nanowatts, making many of them well suited for long lasting battery applications. Other microcontrollers may serve performance-critical roles, where they may need to act more like a digital signal processor (DSP), with higher clock speeds and power consumption

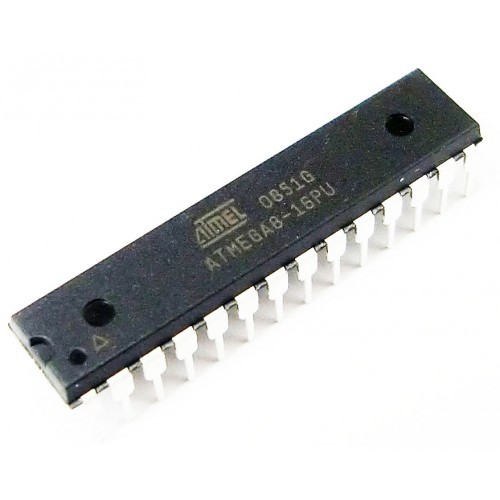


Figure 2.1 : Microcontoller

**2.2 FEATURES OF MICROCONTROLLERS :**

A microcontroller's processor will vary by application. Options range from the simple 4-bit, 8-bit or 16-bit processors to more complex 32-bit or 64-bit processors. In terms of memory, microcontrollers can use random access memory (RAM), flash memory, EPROM or EEPROM. Generally, microcontrollers are designed to be readily usable without additional computing components because they are designed with sufficient onboard memory as well as offering pins for general I/O operations, so they can directly interface with sensors and other components. Microcontroller architecture can be based on the Harvard architecture or von Neumann architecture, both offering different methods of exchanging data between the processor and memory. With a Harvard architecture, the data bus and instruction are separate, allowing for simultaneous transfers. With a Von Neumann architecture, one bus is used for both data and instructions. Microcontroller processors can be based on complex instruction set computing (CISC) or reduced instruction set computing (RISC). CISC generally has around 80 instructions while RISC has about 30, as well as more addressing modes, 12-24 compared to RISC's 3-5. RISC, which places more emphasis on software, often provides better performance than CISC processors, which place more emphasis on hardware, due to its simplified instruction set and, therefore, increased design simplicity, but because of the emphasis it places on software, software can be more complex. Which ISC is used varies depending on application.

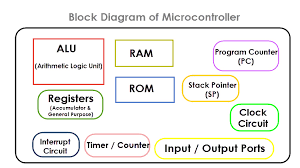


Figure 2.2

When they first became available, microcontrollers solely used assembly language. Today, the C programming language is a popular option. MCUs feature input and output pins to implement peripheral functions. Such functions include analog-to-digital converters, liquid crystal display (LCD) controllers, real-time clock (RTC), synchronous/asynchronous receiver transmitter (USART), timers, universal asynchronous receiver transmitter (UART) and universal serial bus (USB) connectivity. Sensors gathering data related to humidity and temperature among others are also often attached to microcontrollers.

**CHAPTER-3**

**TECHNOLOGIES USED**

**3.1 HARDWARE :**

There are several different kinds of programmable microcontrollers. In this project many of the most common types categorized by several parameters including Bits, Flash size, RAM size, number of input/output lines, packaging type, supply voltage and speed. Our parametric filters will allow you to refine your search results according to the required specifications Programmable microcontrollers contain general purpose input/output pins.

**3.1.1 Arduino UNO Board :**

The Arduino UNO is a widely used open-source microcontroller board based on the Microchip ATmega328P microcontroller and developed by Arduino.cc.The board is equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits. The board features 14 Digital pins and 6 Analog pins. It is programmable with the Arduino IDE (Integrated Development Environment) via a type B USB cable.[4] It can be powered by a USB cable or by an external 9 volt battery, though it accepts voltages between 7 and 20 volts. It is also similar to the Arduino Nano and Leonardo.

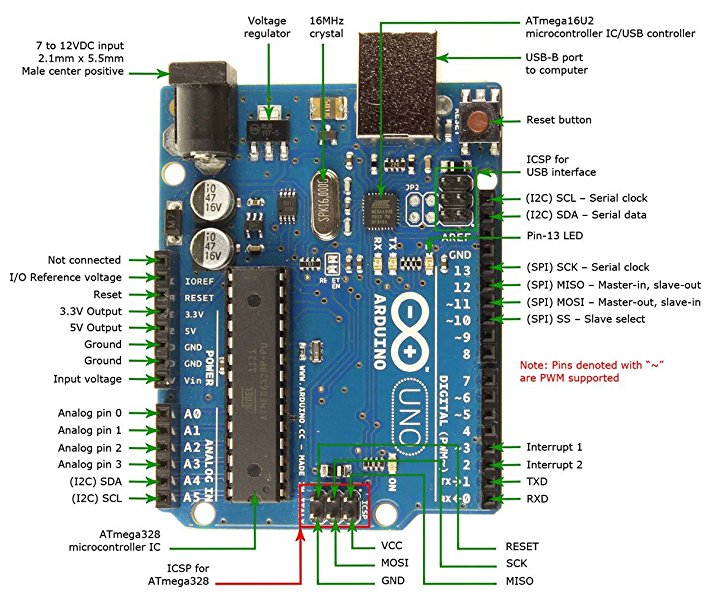


Figure 3.1 : Arduino UNO Board Pin Config

The hardware reference design is distributed under a Creative Commons Attribution Share-Alike 2.5 license and is available on the Arduino website. Layout and production files for some versions of the hardware are also available. "Uno" means one in Italian and was chosen to mark the release of Arduino Software (IDE) 1.0. The Uno board and version 1.0 of Arduino Software (IDE) were the reference versions of Arduino, now evolved to newer releases. The Uno board is the first in a series of USB Arduino boards, and the reference model for the Arduino platform. The ATmega328 on the Arduino Uno comes preprogrammed with a bootloader that allows uploading new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol. The Uno also differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter.[4] The Arduino UNO is generally considered the most user-friendly and popular board of the Arduino board series, with boards being sold worldwide for less than 25$ and clones sold for less than 5$.

**3.1.2 NODE MCU ESP 8266 :**

NodeMCU is an open source IoT platform. It includes firmware which runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which is based on the ESP-12 module. The term "NodeMCU" by default refers to the firmware rather than the development kits. The firmware uses the Lua scripting language. It is based on the eLua project, and built on the Espressif Non-OS SDK for ESP8266. It uses many open source projects, such as lua-cjson, and spiffs. NodeMCU was created shortly after the ESP8266 came out. On December 30, 2013, Espressif Systems began production of the ESP8266. The ESP8266 is a Wi-Fi SoC integrated with a Tensilica Xtensa LX106 core, widely used in IoT applications (see related projects). NodeMCU started on 13 Oct 2014, when Hong committed the first file of nodemcu-firmware to GitHub.

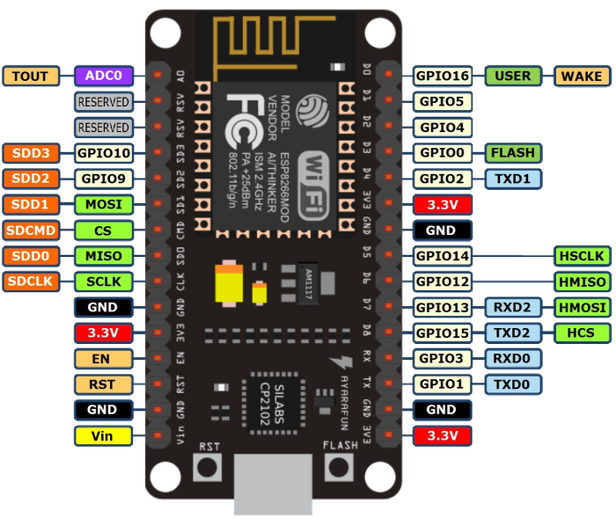


Figure 3.2 : NODE MCU ESP 8266 Pin Config

As Arduino.cc began developing new MCU boards based on non-AVR processors like the ARM/SAM MCU and used in the Arduino Due, they needed to modify the Arduino IDE so that it would be relatively easy to change the IDE to support alternate tool chains to allow Arduino C/C++ to be compiled down to these new processors. They did this with the introduction of the Board Manager and the SAM Core. A "core" is the collection of software components required by the Board Manager and the Arduino IDE to compile an Arduino C/C++ source file down to the target MCU's machine language. Some creative ESP8266 enthusiasts have developed an Arduino core for the ESP8266 WiFi SoC that is available at the GitHub ESP8266 Core webpage. This is what is popularly called the "ESP8266 Core for the Arduino IDE" and it has become one of the leading software development platforms for the various ESP8266 based modules and development boards, including NodeMCUs[4].

**3.1.3 EXTERNAL MODULES USED :**

* **SENSORS :** Sensors are devices that detects and responds to some type of input from the physical environment. The specific input could be light, heat, motion, moisture, pressure, or any one of a great number of other environmental phenomena. The output is generally a signal that is converted to human-readable display at the sensor location or transmitted electronically over a network for reading or further processing. Some of the sensor used in this projects are as follows :
* **INFRA RED SENSOR MODULE :** An infrared sensor is an electronic device, that emits in order to sense some aspects of the surroundings. An IR sensor can measure the heat of an object as well as detects the motion.These types of sensors measures only infrared radiation, rather than emitting it that is called as a passive IR sensor. Usually in the infrared spectrum, all the objects radiate some form of thermal radiations.

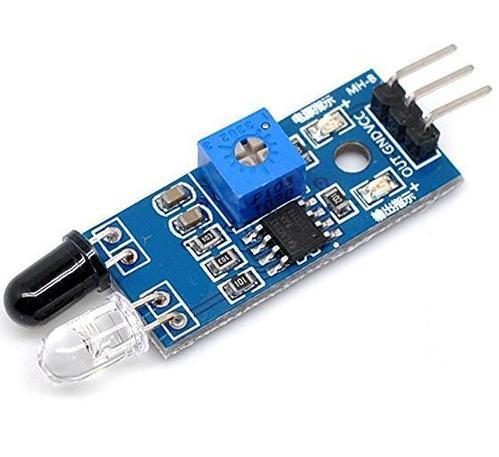
****

Figure 3.3 : IR Sensor

These types of radiations are invisible to our eyes, that can be detected by an infrared sensor.The emitter is simply an IR LED (Light Emitting Diode) and the detector is simply an IR photodiode which is sensitive to IR light of the same wavelength as that emitted by the IR LED. When IR light falls on the photodiode, The resistances and these output voltages, change in proportion to the magnitude of the IR light received**.**

* **MAGNETIC SENSOR MODULE :**

Magnetic sensors detect changes and disturbances in a magnetic field like flux, strength and direction. Other types of detection sensors work with characteristics like temperature, pressure, light. From established knowledge about the existing magnetic field and the data collected from sensors regarding changes and alterations, many things can be known. Rotation, angles, direction, presence and electrical current can all be monitored. Magnetic sensors are divided into two groups, those that measure the complete magnetic field and those that measure vector components of the field. The vector components are the individual points of the magnetic field.



Figure 3.4 : Magnetic Sensor

A recent discovery may allow improvements to magnetic sensors across the board. The NIST (National Institute of Standards And Technology) have revealed that combining layers of magnetic alloy with nano layers of silver increase magnetic sensitivity. Being able to use an extremely thin magnetic sensor (called thin films) is a necessity in applications found in medical devices, weapon detection and data storage.

* **LCD MODULE :** LCD (Liquid Crystal Display) screen is an electronic display module and find a wide range of applications. A 16x2 LCD display is very basic module and is very commonly used in various devices and circuits. These modules are preferred over seven segments and other multi segment LEDs.

The command register stores the command instructions given to the LCD. A command is an instruction given to LCD to do a predefined task like initializing it, clearing its screen, setting the cursor position, controlling display etc. The data register stores the data to be displayed on the LCD. The data is the ASCII value of the character to be displayed on the LCD. Click to learn more about internal structure of a LCD.

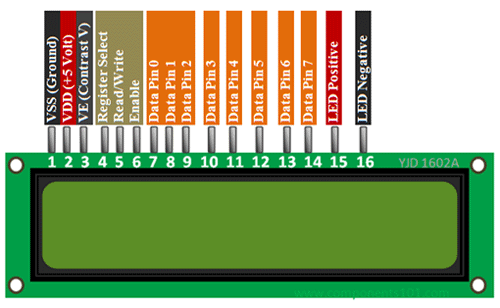


Figure 3.5 : 16 X 2 LCD Module

The reasons being: LCDs are economical; easily programmable; have no limitation of displaying special & even custom characters (unlike in seven segments), animations and so on. A 16x2 LCD means it can display 16 characters per line and there are 2 such lines. In this LCD each character is displayed in 5x7 pixel matrix. This LCD has two registers, namely, Command and Data

* **BUZZER MODULE :** A buzzer or beeper is an audio signalling device, which may be mechanical, electromechanical, or piezoelectric (piezo for short). Typical uses of buzzers and beepers include alarm devices, timers, and confirmation of user input such as a mouse click or keystroke. Piezoelectric buzzers, or piezo buzzers, as they are sometimes called, were invented by Japanese manufacturers and fitted into a wide array of products during the 1970s to 1980s. This advancement mainly came about because of cooperative efforts by Japanese manufacturing companies.

****

Figure 3.6 : Buzzer

A piezoelectric element may be driven by an oscillating electronic circuit or other audio signal source, driven with a piezoelectric audio amplifier. Sounds commonly used to indicate that a button has been pressed are a click, a ring or a beep. A piezoelectric buzzer/beeper also depends on acoustic cavity resonance or Helmholtz resonance to produce an audible beep.

**(3.2) SOFTWARE :**

* **ARDUINO IDE :** The Arduino integrated development environment (IDE) is a cross-platform application (for Windows, macOS, Linux) that is written in the programming language Java. It is used to write and upload programs to Arduino board. The source code for the IDE is released under the GNU General Public License, version 2. The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides

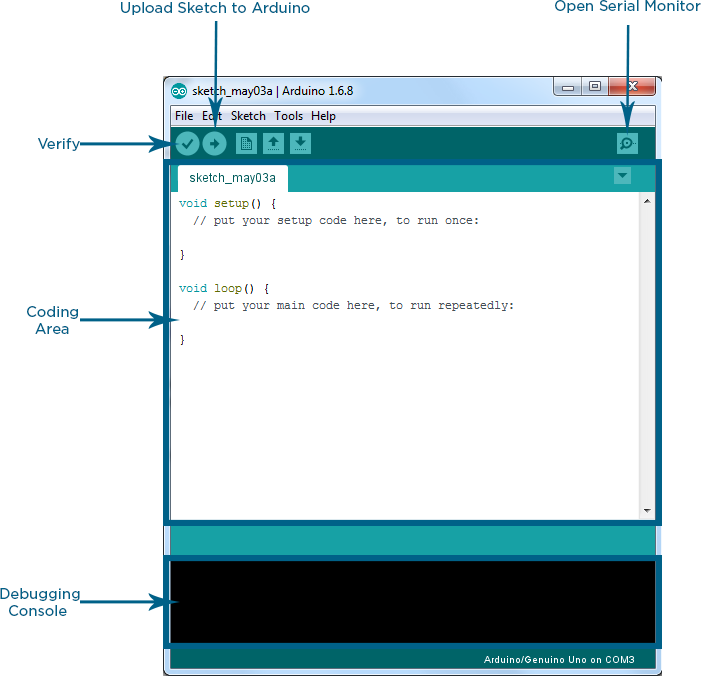


Figure 3.7 : Arduino IDE

many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub main() into an executable cyclic executive program with the GNU toolchain, also included with the IDE distribution. The Arduino IDE employs the program avrdude to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware[4]

* **CLOUD MQTT :** CloudMQTT are managed Mosquitto servers in the cloud. Mosquitto implements the MQ Telemetry Transport protocol, MQTT, which provides lightweight methods of carrying out messaging using a publish/subscribe message queueing model. MQTT is the machine-to-machine protocol of the future. It is ideal for the “Internet of Things” world of connected devices. Its minimal design makes it perfect for built-in systems, mobile phones and other memory and bandwidth sensitiveapplications. Message queues provide an asynchronous communications protocol, the sender and receiver of the message do not need to interact with the message queue at the same time. Messages placed onto the queue are stored until the recipient retrieves them or until the messages times out. MQTT and Mosquitto are for good use by bandwidth sensitive applications[5].CloudMQTT like Amazon,Azure etc provide a managed cloud based mosquitto broker.

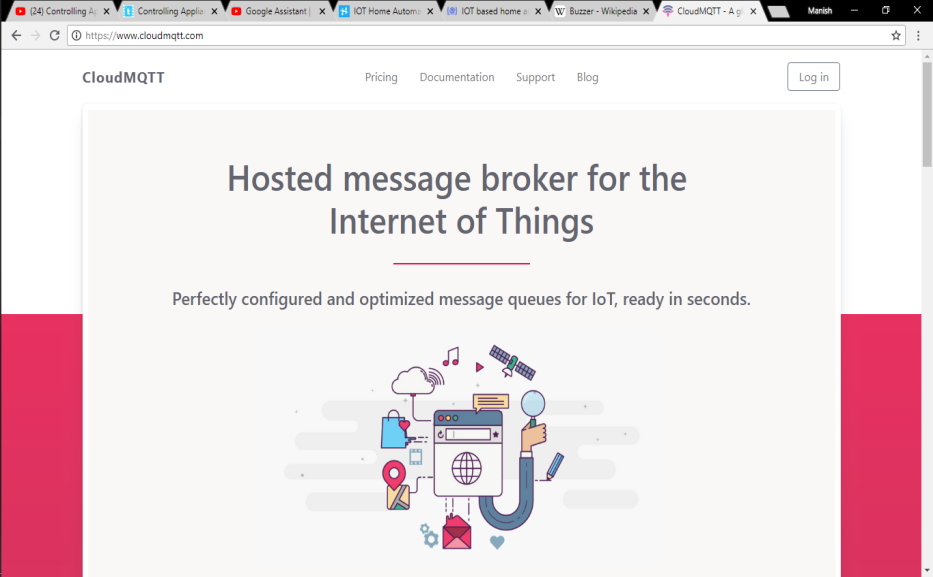


Figure 3.8 : Cloud MQTT

. CloudMQTT let you focus on the application instead of spending time on scaling the broker or patching the platform**.** Cloud based brokers are likely to become very popular in the future for organisations they operate over a wide geographic area. In addition they provide a nice user interface making it very easy to setup your own broker instance and you don’t need to have to manage your own virtual server[5] The plans on CloudMQTT are shared plans which means that several MQTT brokers run on the same hardware.

**CHAPTER-4**

**PROJECT WORK**

**4.1 INTRODUCTION TO PROJECT :**

The idea struck me when I observed that the garbage truck use to go around the town to collect solid waste twice a day. Although this system was thorough it was very inefficient. For example let's say street A is a busy street and we see that the garbage fills up really fast whereas maybe street B even after two days the bin isn't even half full[1]. This example is something that actually happens thus it lead us to the ''Eureka'' moment! 

Figure 4.1 : Garbage Bins

What our system does is it gives a real time indicator of the garbage level in a trashcan at any given time[1]. Using that data we can then optimize waste collection routes and ultimately reduce fuel consumption. It allows trash collectors to plan their daily/weekly pick up schedule.

* + 1. **CRITERIA :**

The basic Model works like this :

To start with you will first have to know that there are three sensors in the dustbin. This will help us generate the percentage of trash in the garbage bin[3]. We then have two criterias which needs to be satisfied to show that the particular bin needs to be emptied :

* The amount of trash, in other words let's say if your bin is half full you don't really need to empty it. Our thresh, or maximum amount that we permit of trash, is 100% of the bin. (You could alter the thresh according to your preference.)
* If supposing a particular garbage bin fills up 20% and then for a week doesn't change, it comes into our second criteria, time. With time even the little amount will start rotting leading to a smelly surrounding. To avoid that our tolerance level is 2 days, so if a garbage bin is less than 100 % but it is two days old it then will also need to be emptied.
  + 1. **METHODOLOGY :**
* At the Actual System there are IR sensor will be placed on the interior side of the garbage bins, the one facing the solid waste. As trash increases, then the IR sensor detects it. This live data will be sent to our micro- controller.
* Micro- controller then processes the data and sends to MQTT Cloud with the help of NODE MCU ESP 8266 Module, and also displays it to LCD module connected to it.
* At the Control Station there is a NODE MCU ESP 8266 which recieves the data from the MQTT Cloud and displays it ot the LCD module connected to it.

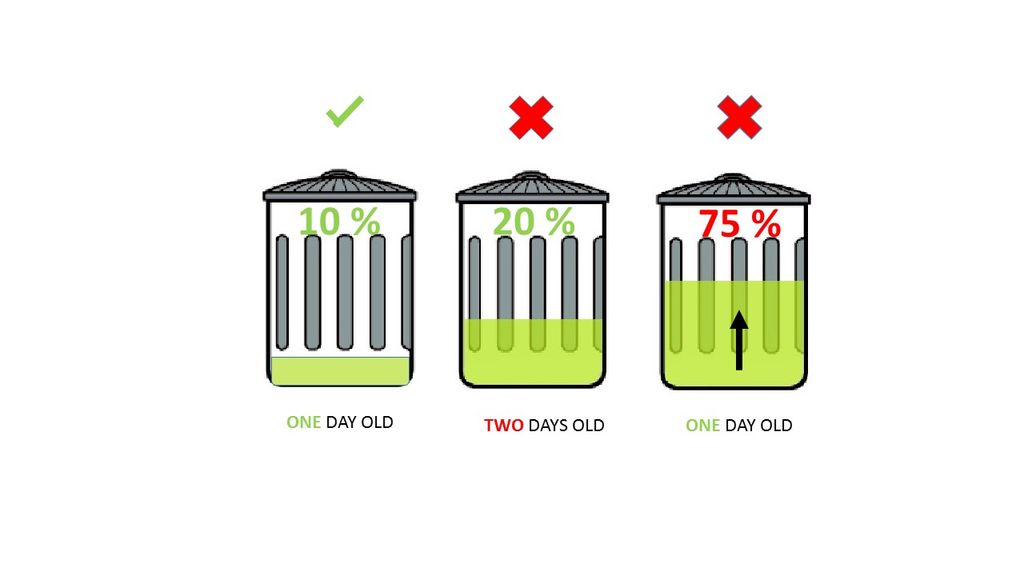


Figure 4.2 : Status of the Bins

* 1. **BLOCK DIAGRAM :**
* The Control Station :

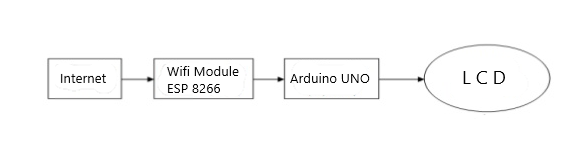


Figure 4.4 : Block Diagram of the System (a)

* The Actual System :

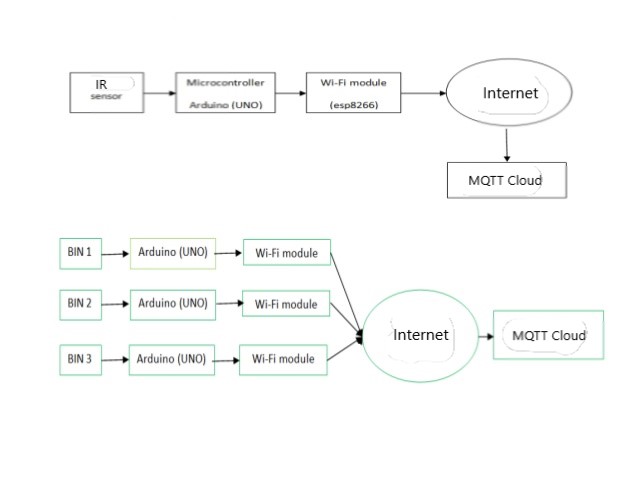


Figure 4.3 : Block Diagram of the System (b)

* 1. **INTRODUCTION TO OTHER PROJECTS :**
* **IOT BASED SENSOR MONITORING SYSTEM :**

A system consisting nodes with various sensors connected to each node which monitors and sends their data over the cloud.

Here are some of the features of the Iot Based Sesnor Monitoring System below :

* Its monitors the real time data of the sensors and displays it over the cloud.
* Its also perform certain actions acccording to the sensors data on the actual system.
* It can be easily acessed by anywhere all over the world.
* It can be remotely controlled.

The link of the project is as follows :

<https://github.com/manishkumar1080/IOT-Based-Sensor-Monitoring-System>

* **WIRELESS CONTROLLED DMD PANEL :**

A system which consist a dot matrix panel which is connected to a arduino board which and is controlled wirelessly via bluetooth.

Here are some of the features of the Wireless Controlled DMD Panel below :

* It can be easily acessed by anywhere all over the world.
* It can be remotely controlled.
* It can print any string or anyword on the Dot Matrix Panel.
* It can uses bluetooth for being wireless so it can easily use anywhere in the world

The link of the project is as follows :

<https://github.com/manishkumar1080/Bluetooth-Controlled-DMD-Panels>

**CHAPTER-5**

**RESULTS AND ANALYSIS**

**5.1 CONNECTION DIAGRAM :**

In this system we have directly connected all the sensors on to the Microcontroller pins and powered the Microcontroller with the DC Adapters.

* At the Actual System :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.No.** | **Arduino Pins**  **(as per code)** | **IR Sensor** | **Magnetic Sensor** | **LCD** | **Buzzer** |
| 1 | D3 | IR 1 (O/P) (Bin 1/100%) |  |  |  |
| 2 | D7 | IR 2(O/P) (Bin 1/50%) |  |  |  |
| 3 | A1 | IR 3 (O/P) (Bin 2/100%) |  |  |  |
| 4 | A5 | IR 4 (O/P) (Bin 2/50%) |  |  |  |
| 5 | A3 |  | Sensor (I/P)(Bin 1) |  |  |
| 6 | A0 |  | Sensor (I/P)(Bin 2) |  |  |
| 7 | A2 |  |  |  | I/P |
| 8 | D5 |  |  | D7 |  |
| 9 | D6 |  |  | D6 |  |
| 10 | D10 |  |  | D5 |  |
| 11 | D11 |  |  | D4 |  |
| 12 | D12 |  |  | EN |  |
| 13 | D13 |  |  | RS |  |
| 14 | +5V | VCC |  | VCC | VCC |
| 15 | Ground | GND | GND | GND | GND |

Table 1 : Connection Diagram (a)

* At the Control Station :

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No.** | **Arduino Pins** | **LCD** | **Buzzer** |
| 1 | D7 |  | I/P |
| 2 | D8 | D4 |  |
| 3 | D9 | D5 |  |
| 4 | D10 | D6 |  |
| 5 | D11 | D7 |  |
| 6 | D12 | EN |  |
| 7 | +5V | VCC | VCC |
| 8 | Ground | GND | GND |

Table 2 : Connection Diagram (b)

Here are some of the pictures of the Actual System is as follows :



**IR 2 (BIN 1)**

**IR 1 (BIN 1)**

Figure 5.1 : Bin 1



**IR 3 (BIN 2)**

**IR 4 (BIN 2)**

Figure 5.2 : Bin 2

The Actual System and the Control System loos like this :

****

Figure 5.3 : SYSTEM

Here are some of the screenshots of the programs programed on Arduino IDE :

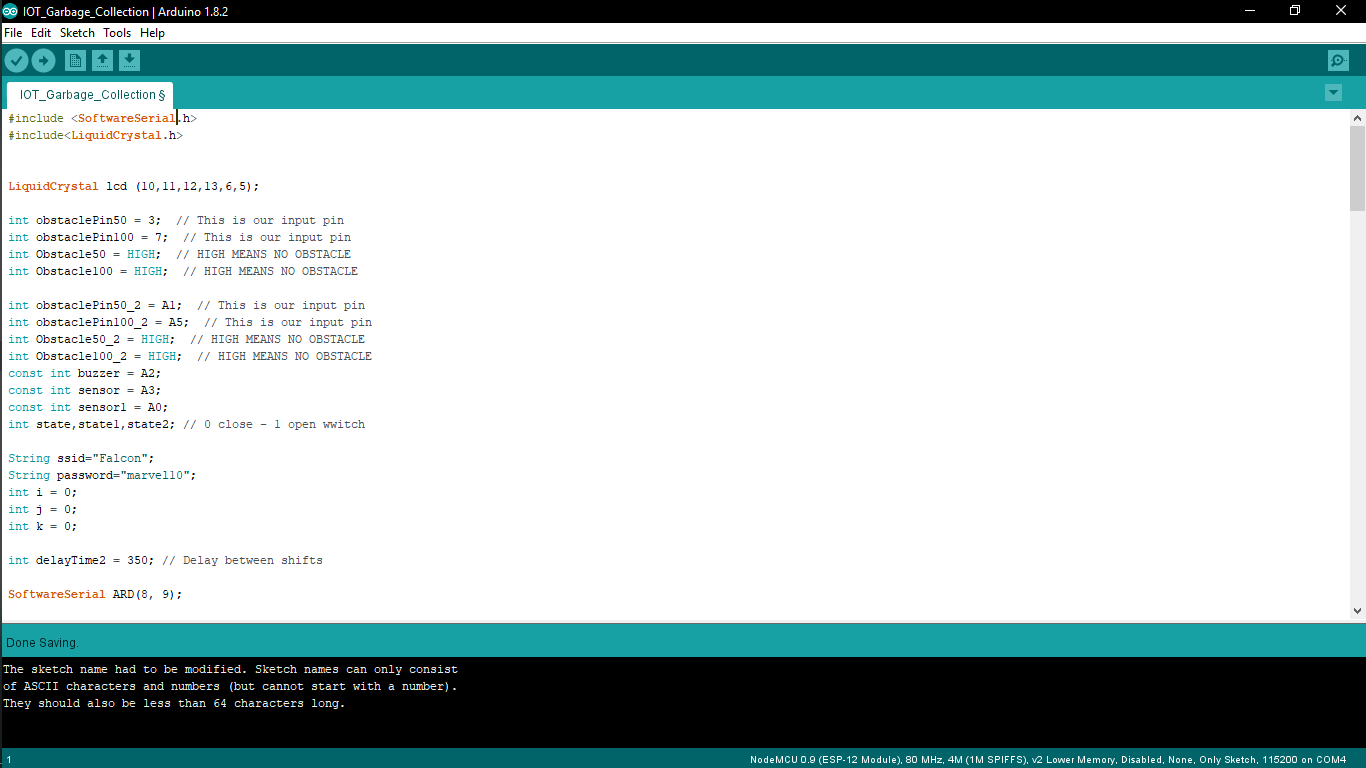


Figure 5.4 : Arduino IDE (Program)

And here is the screenshot of the Cloud MQTT cloud service used in this project :

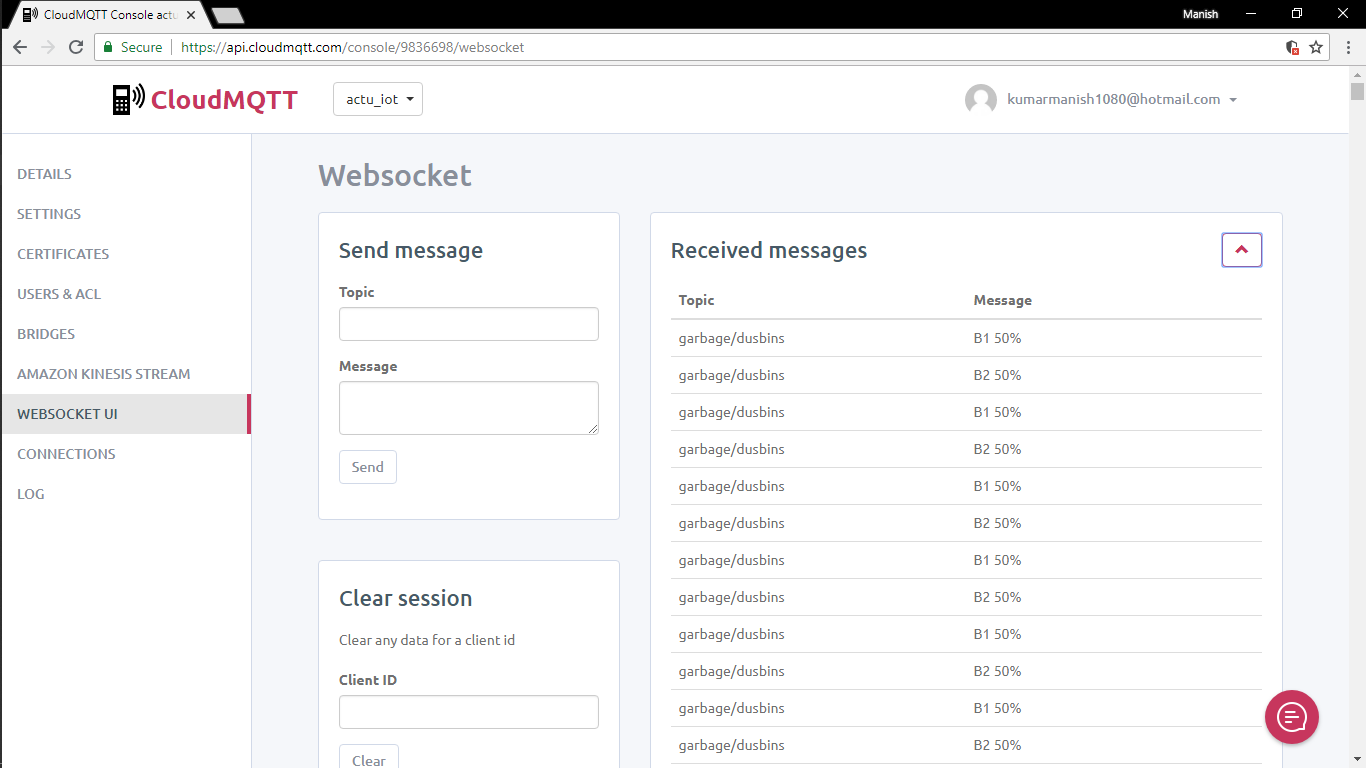


Figure 5.5 : Cloud MQTT (Cloud Status)

The link of the project is as follows :

<https://github.com/manishkumar1080/Smart-Garbage-Management-System>

**CONCLUSION**

The future of IoT is virtually unlimited due to advances in technology and consumers desire to integrate devices such as smart phones with household machines. Wi-Fi has made it possible to connect people and machines on land, in the air and at sea. It is critical that both companies and governments keep in ethics in mind as we approach the fourth Industrial Revolution. With so much data traveling from device to device, security in technology will be required to grow just as fast as connectivity in order to keep up with demands.Governments will undoubtable face tough decisions as to how far the private the sector is allowed to go in terms of robotics and information sharing. The possibilities are exciting, productivity will increase and amazing things will come by connecting the world**.** Hopefully we gained a better idea of what a embedded engieer actually does and what your life might look like if you decided to take it on as a career. This has really just been a teaser into the world of embedded systems.You will build dozens of programs, projects, and models by the time this journey is complete and will have the skills necessary to step into fulltime development.

Getting all the way there is going to be challenging. In fact, you should check out the post "Internet Of Things Changing World" so you have a good idea of what the journey ahead is like. But what worthwhile thing is truly easy?

Yes, it's going to be challenging.

But it's going to be fun.

And it might be life-changing too.

In last ,I can say that all the projects are interesting and amazing

I enjoy a lots on working on them.

**FUTURE SCOPE**

It’s the technology of today which is touching and transforming the every aspect of our real life. IOT has given a concept of Machine to-Machine (M2M) communication. Companies like Microsoft and SAP are implementing strategy to capitalize on the Internet of Things so that you can just stop your business and start making it thrive. IOT is going to have huge impact on home automation and building automation system where every convenience will be taken care of by the interconnected devices on IOT. It is also deployed on large scale for example in Songdo , South Africa , the first of its own kind fully equipped and wired smart city is near to

completion ( known as Ubiquitous City). With the personal electronics good connected to Internet will enable us to “author” our lives. In medical science field, IOT has given a privilege to devices and system to sense for coming disease and to prevent it ,for eg: It can make a person healthier with wearables that can predict heart attack and cardiovascular strokes. As per a report of Thesunsdaily, consumers will start initiating the usage of IoT in a better way during 2015 and onwards compared to past usage. It is expected that IoT products with interoperable capability will dominate the market. Awareness of IoT products is also vital for market penetration along with security features. Even very few Americans are aware of the usage of these products. As per a study of Consumer Electronics Association and Parks Associates found only 10% of the household in USA fully understood the usage of these products. Many interesting IoT products like automatic door locks, Wi-Fi connected ceiling fans, light swirches,LED bulbs, smart watches,3-D printers and smart clothes will be popular among consumers.MyBrain Technology in France has developed "Melomind". This EEG Headset can measure a human's brain waves and adjust music in a Smartphone app as they change. This product can be used

as a digital meditation aid. A smart baby pacifier can measure the temperature of a baby and transmit the same to the Smartphone of parents. IoT is proved to be an emerging technological innovation. In the current context, it is now possible that an helmet of a two wheeler can interact with a car for avoiding collision. Connected toothbrush can now monitor and make one's experience pleasurable .A three dimensional senor of the

electric brush can connect with Smartphone apps and provide real time feedback to the person . Many scopes will be created for technology companies to release offerings as per the behaviour of consumers. It may so

happen that Netflix can know when a person is sad and alone by monitoring the smart watch, smart thermostat and in-home camera. Subsequently, Netflix may offer a movie to change the mood. In a consumer electronics show in Los Vegas, Samsung informed that the company would invest 100 million dollar for progress of IoT. The company will also promote an open technology ecosystem for facilitating the usage of IoT

**APPENDIX**

**Program for the Actual System :**

#include <SoftwareSerial.h>

#include<LiquidCrystal.h>

LiquidCrystal lcd (10,11,12,13,6,5);

int obstaclePin50 = 3; // This is our input pin

int obstaclePin100 = 7; // This is our input pin

int Obstacle50 = HIGH; // HIGH MEANS NO OBSTACLE

int Obstacle100 = HIGH; // HIGH MEANS NO OBSTACLE

int obstaclePin50\_2 = A1; // This is our input pin

int obstaclePin100\_2 = A5; // This is our input pin

int Obstacle50\_2 = HIGH; // HIGH MEANS NO OBSTACLE

int Obstacle100\_2 = HIGH; // HIGH MEANS NO OBSTACLE

const int buzzer = A2;

const int sensor = A3;

const int sensor1 = A0;

int state,state1,state2; // 0 close - 1 open wwitch

String ssid="Falcon";

String password="marvel10";

int i = 0;

int j = 0;

int k = 0;

int delayTime2 = 350; // Delay between shifts

SoftwareSerial ARD(8, 9);

void scrollInFromRight (int line, char str1[]) {

i = strlen(str1);

for (j = 16; j >= 0; j--) {

lcd.setCursor(0, line);

for (j = 16; j >= 0; j--) {

lcd.setCursor(0, line);

for (k = 0; k <= 15; k++) {

lcd.print(" "); // Clear line

}

lcd.setCursor(j, line);

lcd.print(str1);

delay(delayTime2);

}

}

void scrollInFromLeft (int line, char str1[]) {

i = 40 - strlen(str1);

line = line - 1;

for (j = i; j <= i + 16; j++) {

for (k = 0; k <= 15; k++) {

lcd.print(" "); // Clear line

}

lcd.setCursor(j, line);

lcd.print(str1);

delay(delayTime2);

}

}

void setup()

{

Serial.begin(115200);

ARD.begin(115200);

lcd.begin(16, 2);

scrollInFromRight(0, " Welcome to the");

scrollInFromLeft(1, " System");

delay(100);

lcd.clear();

pinMode(sensor, INPUT);

pinMode(obstaclePin50, INPUT);

pinMode(obstaclePin100, INPUT);

delay(500);

ARD.print("u");

ARD.print(ssid);

ARD.print("\n");

delay(100);

ARD.print("p");

ARD.print(password);

ARD.print("\n");

delay(200);

}

void loop()

{

char bin1,bin2;

if(ARD.available())

{

while(ARD.read()!='\*');

bin1 = bin\_1();

bin2 = bin\_2();

lcd.setCursor(0,0);

lcd.print(" Bin1=");

lcd.print(bin1);

lcd.setCursor(0,1);

lcd.print(" Bin2=");

lcd.print(bin2);

if(bin1=='F' || bin2=='F')

{

if (state == LOW && bin1=='F' )

{

tone(buzzer, 1000);

delay(1000);

}

if (state1 == LOW && bin2=='F')

{

tone(buzzer, 1000);

delay(1000);

}

else{

noTone(buzzer);

}

}

}

}

char bin\_1()

{

char chr;

Obstacle50 = digitalRead(obstaclePin50); //Reads the output of the obstacle sensor from the 7th PIN of the Digital section of the arduino

Obstacle100 = digitalRead(obstaclePin100); //Reads the output of the obstacle sensor from the 7th PIN of the Digital section of the arduino

state = digitalRead(sensor);

if (Obstacle50 == HIGH && Obstacle100 == HIGH) //LOW means something is ahead, so illuminates the 13th Port connected LED

{

noTone(buzzer);

ARD.print("!");

ARD.print("B1 50%");

ARD.print("\a");

chr='B';

}

else if (Obstacle50 == LOW && Obstacle100 == HIGH) //LOW means something is ahead, so illuminates the 13th Port connected LED

{

noTone(buzzer);

ARD.print("!");

ARD.print("A1 50%");

ARD.print("\a");

chr='A';

}

else if (Obstacle50 == LOW && Obstacle100 == LOW) //LOW means something is ahead, so illuminates the 13th Port connected LED

{

ARD.print("!");

ARD.print("F1 100%");

ARD.print("\a");

chr='F';

}

else

{

ARD.print("!");

ARD.print("F1 100%");

ARD.print("\a");

chr='F';

}

return chr;

}

char bin\_2()

{

char chr;

Obstacle50\_2 = digitalRead(obstaclePin50\_2); //Reads the output of the obstacle sensor from the 7th PIN of the Digital section of the arduino

Obstacle100\_2 = digitalRead(obstaclePin100\_2); //Reads the output of the obstacle sensor from the 7th PIN of the Digital section of the arduino

state = digitalRead(sensor1);

if (Obstacle50\_2 == HIGH && Obstacle100\_2 == HIGH) //LOW means something is ahead, so illuminates the 13th Port connected LED

{

noTone(buzzer);

ARD.print("#");

ARD.print("B2 50%");

ARD.print("\a");

chr='B';

}

else if (Obstacle50\_2 == LOW && Obstacle100\_2 == HIGH) //LOW means something is ahead, so illuminates the 13th Port connected LED

{

noTone(buzzer);

ARD.print("#");

ARD.print("A2 50%");

ARD.print("\a");

chr='A';

}

else if (Obstacle50\_2 == LOW && Obstacle100\_2 == LOW) //LOW means something is ahead, so illuminates the 13th Port connected LED

{

ARD.print("#");

ARD.print("F2 100%");

ARD.print("\a");

chr='F';

}

else

{

ARD.print("#");

ARD.print("F2 100%");

ARD.print("\a");

chr='F';

}

return chr

}

**Program for the Control Station :**

#include <ESP8266WiFi.h>

#include <PubSubClient.h>

#include <stdlib.h>

String ssid;

String password ;

static String temp;

static String topi;

static String st,st1;

static String top;

const char \*mqtt\_server = "m20.cloudmqtt.com";

const int mqtt\_port = 17482;

const char \*mqtt\_user = "fkoxtvvc";

const char \*mqtt\_pass = "K4oJULvZgCOG";

const char \*mqtt\_client\_name = "arduinoClient";

#define BUFFER\_SIZE 100

WiFiClient wclient; //Declares a WifiClient Object using ESP8266WiFi

PubSubClient client(wclient, mqtt\_server,mqtt\_port); //instanciates client object

//Function is called when, a message is recieved in the MQTT server.

void callback(const MQTT::Publish& pub) {

topi=pub.topic();

if (pub.has\_stream()) {

uint8\_t buf[BUFFER\_SIZE];

int read;

while (read = pub.payload\_stream()->read(buf, BUFFER\_SIZE)) {

}

pub.payload\_stream()->stop();

} else

temp=pub.payload\_string();

}

void setup() {

Serial.begin(115200);

while(1)

{

if(Serial.available())

{

while(Serial.read()!='u');

ssid=Serial.readStringUntil('\n');

goto here;

}

}

here:

while(1)

{

if(Serial.available())

{

while(Serial.read()!='p');

password=Serial.readStringUntil('\n');

goto there;

}

}

there:

if (WiFi.status() != WL\_CONNECTED) {

WiFi.begin(ssid.c\_str(), password.c\_str());

if (WiFi.waitForConnectResult() != WL\_CONNECTED)

return;

}

}

void loop()

{

if (WiFi.status() == WL\_CONNECTED)

{

//client object makes connection to server

if (!client.connected()) {

//Authenticating the client object

if (client.connect(MQTT::Connect("mqtt\_client\_name").set\_auth(mqtt\_user, mqtt\_pass))) {

//Subscribe code

// client.subscribe("garbage");

} else {

}

}

if (client.connected())

client.loop();

}

client.set\_callback(callback);

Serial.print("\*");

if(Serial.available())

{

top=Serial.readStringUntil('!');

st=Serial.readStringUntil('\a');

top=Serial.readStringUntil('#');

st1=Serial.readStringUntil('\a');

client.publish("garbage/dusbins", st);

delay(100);

client.publish("garbage/dusbins", st1);

delay(100);

client.set\_callback(callback);

}

}

**REFERENCES**

1. <https://www.hackster.io/>
2. <https://github.com/>
3. <https://www.electronicshub.org/>
4. <https://www.arduino.cc/>
5. <https://www.cloudmqtt.com/>