IOT BASED WATER TANK OVERFLOW DETECTION SYSTEM

Submitted in partial fulfillment of the requirements

of the degree of

Bachelor of Engineering

MINI PROJECT REPORT
(BE)

Name of Student	Roll no
Fullar Soniya	2102613
Sakhare Pankaj	2102633
Yadav Omkar	2102649

Project Guide

(PROF. R.S. MESHRAM)

DEPARTMENT OF INFORMATION TECHNOLOGY

KONKAN GYANPEETH COLLEGE OF ENGINEERING

UNIVERSITY OF MUMBAI
(2023-24)

CERTIFICATE

This is to certify that the Mini Project- **BE-17** entitled "**IOT BASED WATER TANK OVERFLOW DETECTION SYSTEM**" is a bonafide work of **"Soniya Fullar (13), Sakhare Pankaj (33) and Yadav Omkar (49)"** Submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of **"Bachelor of Engineering"** in **"Information Technology".**

(Prof. R.S. MESHRAM)	**(Prof A.S. KUNTE)**	**(Dr. V.J. PILLEWAN)**
Guide	Head Of department	Principal

KONKAN GYANPEETH COLLEGE OF ENGINEERING, KARJAT

University of Mumbai

(AY 2023-24)

PROJECT REPORT APPROVAL

This Mini Project- **BE-17** on "**IOT BASED WATER TANK OVERFLOW DETECTION SYSTEM**" by **"Soniya Fullar (13), Sakhare Pankaj (33) and Yadav Omkar (49)"**is approved for the degree of **Bachelor of Engineering** in **Information Technology.**

Examiners

1)……

(Internal Examiner Name & Sign)

2)……

(External Examiner Name & Sign)

Date: 07/11/2023

Place: KARJAT

DECLARATION

I declare that this written submission represents my ideas in my own word and where others'. ideas or words have been included. I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the institute and can also evoke panel action from the source which have thus not been properly cited or from whom proper permission has not been taken when needed

Name of Student	**Roll no**	**Signature**
Soniya Fullar	2102613	……..
Sakhare Pankaj	2102633	……..
Yadav Omkar	2102649	……

Date: 07/11/2023

Place: KARJAT

ACKNOWLEDGEMENT

This Project would not have been completed without the encouragement and support of many people who gave their precious time and encouragement throughout this period. First and foremost we would like to express our sincerest gratitude to our project guide Prof. Rupendra S. Meshram for his invaluable support, guidance, motivation and encouragement throughout the period this work was carried out.

We are also grateful to prof. Anup Kunte Head of Department, Information Technology Engineering who gave us permission to use the project lab and give us all the necessary information for completion of project.

Name of Student	**Roll no**	**Signature**
Soniya Fullar	2102613	……..
Sakhare Pankaj	2102633	……..
Yadav Omkar	2102649	……

INDEX

ABSTRACT

This project gives us a proper insight into water conservation through the application of a water level monitoring system. Industries and homes where water is used to a large extent can implement the given model to get the information about the water level in real-time. An ultrasonic sensor and a NodeMCU microcontroller are used to achieve the given results. Using the power of internet, simplification can be further achieved for the users as they will be able to view the data from anywhere and thereby giving the users the power to monitor it from anywhere and giving them the power to achieve the control of water loss from everywhere. The need for this control arises due to the fact that water loss is happening at a critical rate and if not controlled the situation will worsen further. This work is presenting a water level control and monitoring system in a tank, by using the development boards NodeMCU ESP8266 and the ultrasonic sensor HC-SR04. We had issued a control algorithm for controlling the water level and a program in the development environment Arduino IDE that has been implemented and validated on the experimental stand. For monitoring the development board NodeMCU ESP8266 has been used that takes the information from the ultrasonic sensor HC-SR04 and sends it through Wi-Fi to the platform BLYNK App. The measured value of the water level is shown on the LCD 1602 display. The advantage of this water level detection and monitoring system in the tank is given by the very low costs in comparison to the cost of an industrial type control system.

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1 INTRODUCTION

The topic of our project is Water Level Monitoring. We intend to design a system which can monitor the level of water from water tanks. We know that the resources of water are continuously depleting, so we need to keep a check on our usage so that no extra water goes unutilized. If we monitor the water level in tanks we ensure that the tank never overflows and hence no extra water loss would happen. Water is one in all the foremost vital basic desires for all living beings. However sadly an enormous quantity of water is being wasted by uncontrolled use[1]. The major losses happen either in homes or industries. So the system can be applied there to fill the void. Our main aim is to design a system which will be versatile, economical and simply configurable which will be able to solve water losing issues. The water level data can be used for various purposes for better management of water source. Monitoring water level from remote location may be very useful when it is not possible to visit location physically every time. The system uses a NodeMCU microcontroller and a HC- SR04 Ultrasonic sensor to do the work. The sensor sends a pulse of ultrasonic waves and the waves after hitting the water source, again are reflected back towards the sensor. The NodeMCU microcontroller computes the time taken for the journey and also computes the distance. After that a pump is used to always help to maintain the water level between 30-70 % so that the users can always have adequate amount of water supply at their home[2].

1.2 EXISTING SYSTEM

The working principle of a water level indicator is actually quite simple. Water level indicators work by **using sensor probes** to indicate water levels in a storage tank. These probes send information back to the indicator LCD display. As mentioned above, the HD44780 LCD 1602 display can be programmed to show the level of water in five different modes which is: Very Low, Low, Medium, High & Full according to that by considering that information we can refill the water again. Ultrasonic sensor measures the depth of water in the tank. The program written will receive the data from the sensor and supplies the collected data to NodeMCU ESP8266 and based on the data received, we can manually switch the Button ON or OFF[1].

This work is presenting a water level detecting and monitoring system in a tank, by using the development boards NodeMCU ESP8266 and the ultrasonic sensor HC-SR04. We had issued a control algorithm for detecting the water level and a program in the development environment Arduino IDE that has been implemented and validated on the experimental stand. For monitoring the development board NodeMCU ESP8266 has been used that takes the information from the ultrasonic sensor HC-SR04 and sends it through Wi-Fi to the platform BLYNK App. The measured value of the water level is shown on the HD44780 LCD 1602 display. The advantage of this water level detection and monitoring system in the tank is given by the very low costs in comparison to the cost of an industrial type control system[3].

1.3 OBJECTIVE

The objective of this project is to notify the user the amount of water that is present in the overhead water tank. This project can be further enhanced to control the water level in the tank by turning it on, when the water level is low, and turning it off when the water level is high. thus, the NODE MCU water level indicator helps in preventing wastage of water in overhead tank. Ultrasonic sensor to measure the water level in terms of distance.[2] The primary objective of the proposed model is to monitor the water present in the tank and sump visually when the water exceeds or recedes the threshold set by the user. The main aim of this system is to monitor the water level at rural areas so that they help in detecting the wastage of water and measures can be taken to avoid unnecessary overflowing of water in the areas where monitoring is a difficult task. The LED switches on when the particular water level is detected[3].

1.4 PROPOSED SYSTEM

In this proposed application architecture of the system, the physical layer consists of the physical environment such as water source and to sense the water level in a water source with required network connectivity. The sensor data is uploaded to Blynk cloud at service layer of the architecture which is the second layer. The Blynk is a cloud application platform for IoT. Blynk has many advantages. Blynk IoT platform offers a full suite of software allowing to prototype, deploy, and remotely manage connected electronic devices at any scale: from small

IoT projects to millions of commercial connected products.[1] Blynk has a no-code approach to IoT app building and also a mobile app editor. Blynk is a cloud-based, hardware-agnostic internet of things (IoT) platform which allows users to connect more than 400 hardware Save Power: In an era of energy conservation, these devices are very beneficial to save energy. Hence electrical power wastage can be reduced. These sensor detect water levels and minimize the usage of electricity. These are low cost and easy to install devices. The design is compact and with low maintenance and cam clearly indicate the water levels in the overhead tank. As they can be fully automatic, they save time and avoid seepage of roofs and walls due to overflowing tanks "Water Tank Monitoring System" is introduced to overcome the drawbacks in the existing system. In this system, as soon the tank becomes empty, pump is automatically turned on. And when the tank is full, pump is automatically turned off. Raspberry Pi can be used to automate the system and to monitor the water level.[1]

CHAPTER 2
LITERATURE SURVEY

2.1 LITERATURE REVIEW

L.A. Gama-Moreno, A.Corralejo, A.Ramirez-Molina [1], implemented a system to monitor water tanks. The implemented system consisted of instrumentation system, an application, which managed the water levels and a mobile user interface. The system was called as Monitoring Water Tanks (MWT). Ultrasonic sensor were put into use along with NODE MCU ESP8266 Microcontroller Board, which is connected to the application service.

BezaNegashGetu, HussainA.Attia [2], have developed a system which initially tests the availability of water in the tank with the help of a level detector and then adjusts the state of the water pump according to the information collected through the level detector. The proposed system consists of water level sensor The proposed system eliminates manually controlling of water requirements in home and agricultural fields.

PriyenP.Shah, Anjali A.Patil [3], introduced a project which uses Android application and IoT for monitoring of water in tanks. This project uses ESP 8266 as microcontroller. Maximum and minimum levels of water are obtained through ESP from blynk cloud. When the water level is between Maximum and minimum status can be controlled by the user. This project overcomes the short-comings of conventional tanks which can neither monitor nor control the water level in tank.

Gama-Moreno, L. A., A. Corralejo, A. Ramirez-Molina, J. A. Torres-Rangel, C. Martinez-Hernandez, and M. A. Juarez [4]. "A design of a water tanks monitoring system based on mobile devices." In 2016 International Conference on Mechatronics, Electronics and Automotive Engineering (ICMEAE), pp. 133-138. IEEE, 2016.

Malche, Timothy, and Priti Maheshwary [5]. "Internet of things (IoT) based water level monitoring system for smart village." In Proceedings of International Conference on Communication and Networks: ComNet 2016, pp. 305-312. Springer Singapore, 2017.

CHAPTER 3
REQUIREMENTS ANALYSIS

3.1 BLOCK DIAGRAM:

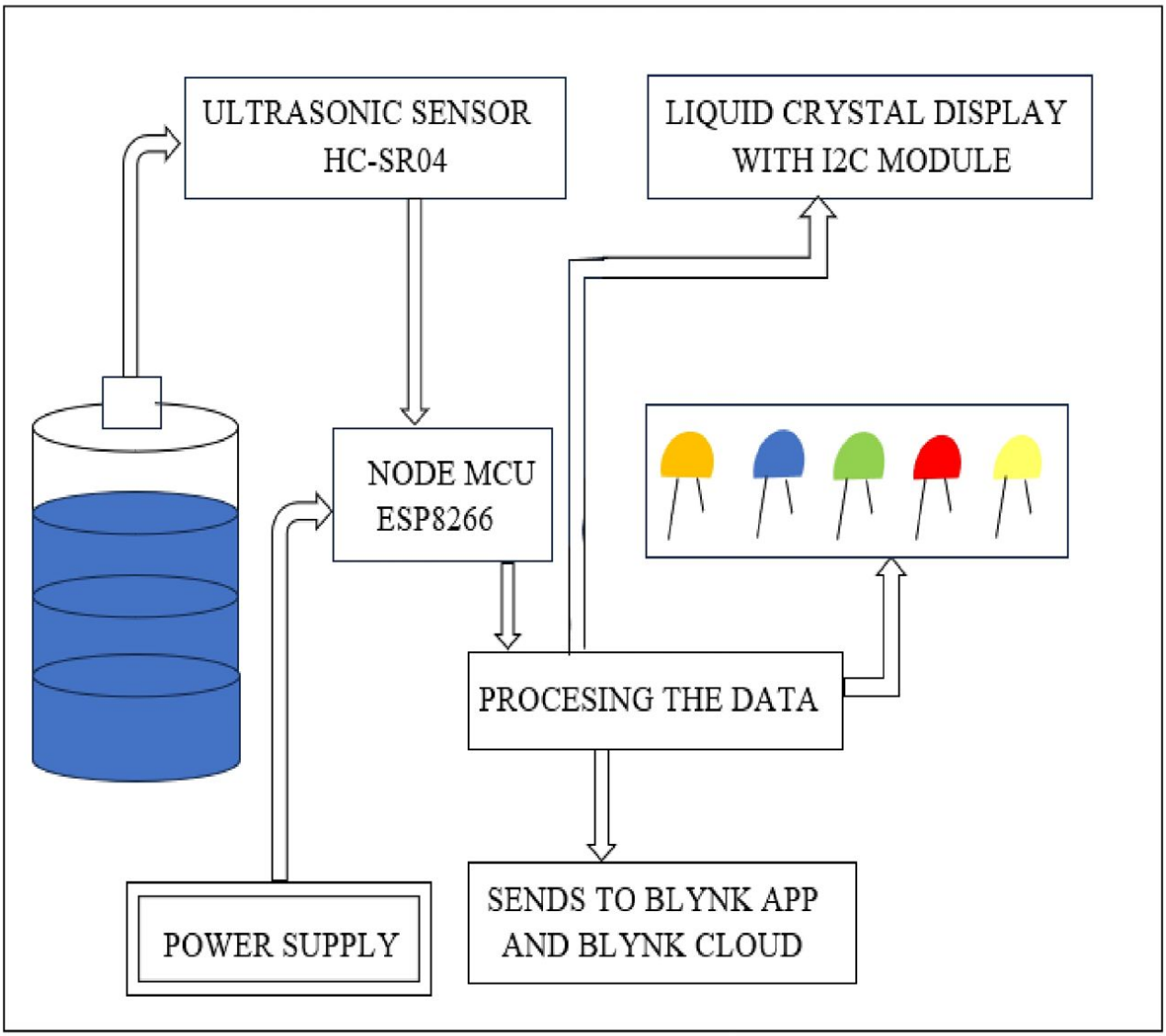

Figure no. 3.1: Block Diagram.

3.2 CIRCUIT DIAGRAM & CONNECTIONS:

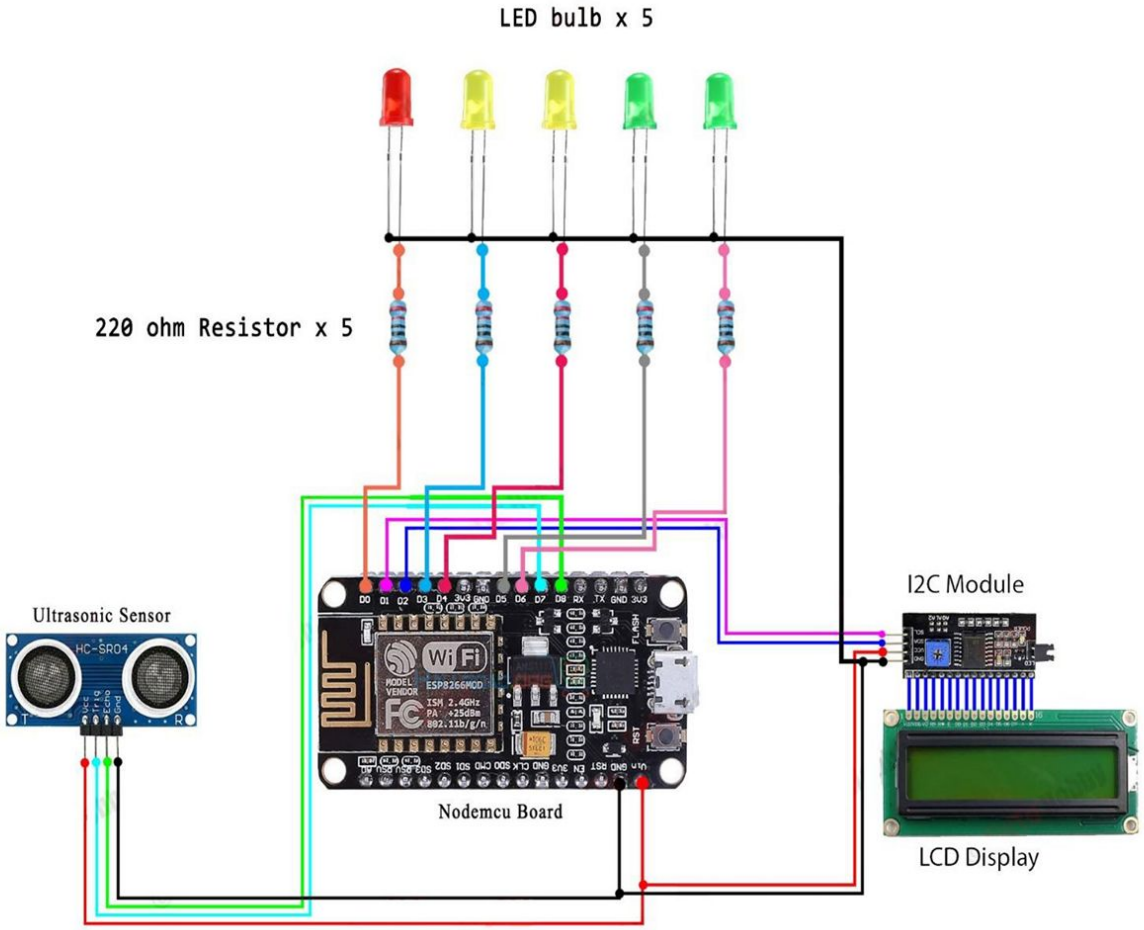

Figure no. 3.2: Circuit Diagram Of System

3.3 HARDWARE IMPLEMENTATION

3.3.1 NODEMCU ESP8266 :

The NodeMCU (Node MicroController Unit) is an open-source software and hardware development environment built around an inexpensive System-on-a-Chip (SoC) called the ESP8266. The ESP8266, designed and manufactured by Espressif Systems, contains the crucial elements of a computer: CPU, RAM, networking (WiFi), and even a modern operating system and SDK. That makes it an excellent choice for Internet of Things (IoT) projects of all kinds.

The NodeMCU ESP8266 development board comes with the ESP-12E module containing the ESP8266 chip having Tensilica Xtensa 32-bit LX106 RISC microprocessor. This microprocessor supports RTOS and operates at 80MHz to 160 MHz adjustable clock frequency. NodeMCU has 128 KB RAM and 4MB of Flash memory to store data and programs. Its high processing power with in-built Wi-Fi / Bluetooth and Deep Sleep Operating features make it ideal for IoT projects.

NodeMCU can be powered using a Micro USB jack and VIN pin (External Supply Pin). It supports UART, SPI, and I2C interface.

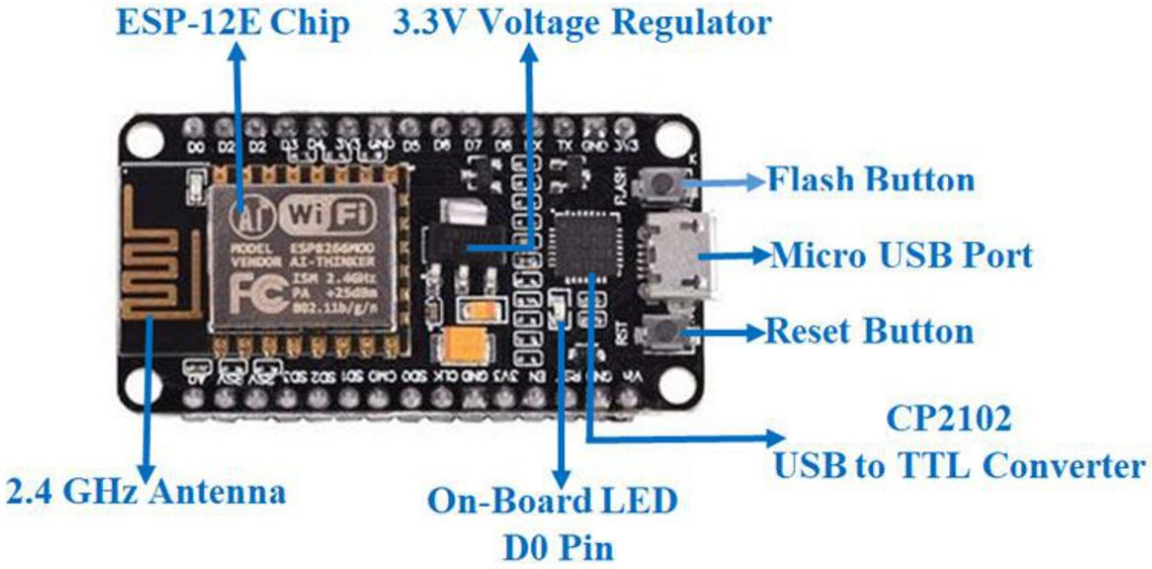

Figure no. 3.3: Nodemcu Esp8266 Board

Programming NodeMCU ESP8266 with Arduino IDE

The NodeMCU Development Board can be easily programmed with Arduino IDE since it is easy to use.

Programming NodeMCU with the Arduino IDE will hardly take 5-10 minutes. All you need is the Arduino IDE, a USB cable and the NodeMCU board itself. You can check this Getting Started Tutorial for NodeMCU to prepare your Arduino IDE for NodeMCU.

Uploading your first program

Once Arduino IDE is installed on the computer, connect the board with the computer using the USB cable. Now open the Arduino IDE and choose the correct board by selecting Tools>Boards>NodeMCU1.0 (ESP-12E Module), and choose the correct Port by selecting Tools>Port. To get it started with the NodeMCU board and blink the built-in LED, load the example code by selecting Files>Examples>Basics>Blink. Once the example code is loaded into your IDE, click on the 'upload' button given on the top bar. Once the upload is finished, you should see the built-in LED of the board blinking.

NodeMCU ESP8266 Specifications & Features

- Microcontroller: Tensilica 32-bit RISC CPU Xtensa LX106
- Operating Voltage: 3.3V
- Input Voltage: 7-12V
- Digital I/O Pins (DIO): 16
- Analog Input Pins (ADC): 1
- UARTs: 1
- SPIs: 1
- I2Cs: 1
- Flash Memory: 4 MB
- SRAM: 64 KB
- Clock Speed: 80 MHz
- USB-TTL based on CP2102 is included onboard, Enabling Plug n Play
- PCB Antenna

- Small Sized module to fit smartly inside your IoT projects.

3.3.1.1 ESP8266 12-E CHIP PINOUT DIAGRAM :

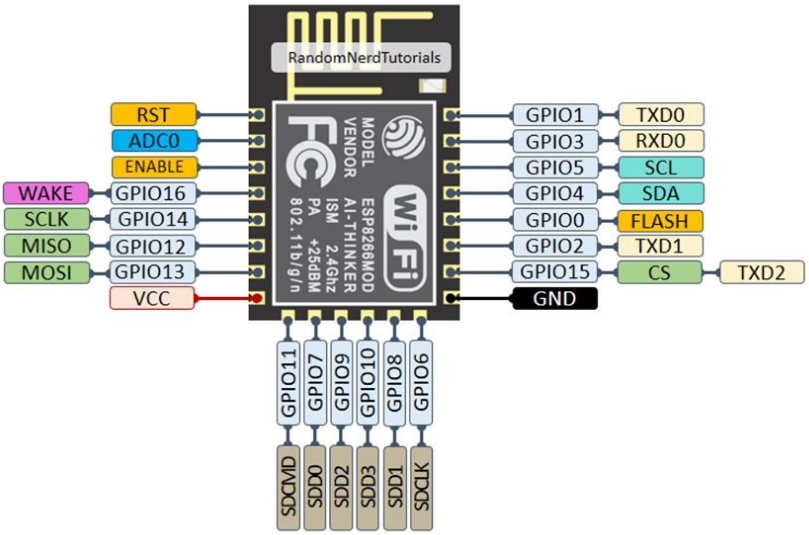

Figure no. 3.4: ESP8266 12-E Chip Pinout Diagram

3.3.1.2 ESP8266 12-E NODEMCU KIT PINOUT DIAGRAM :

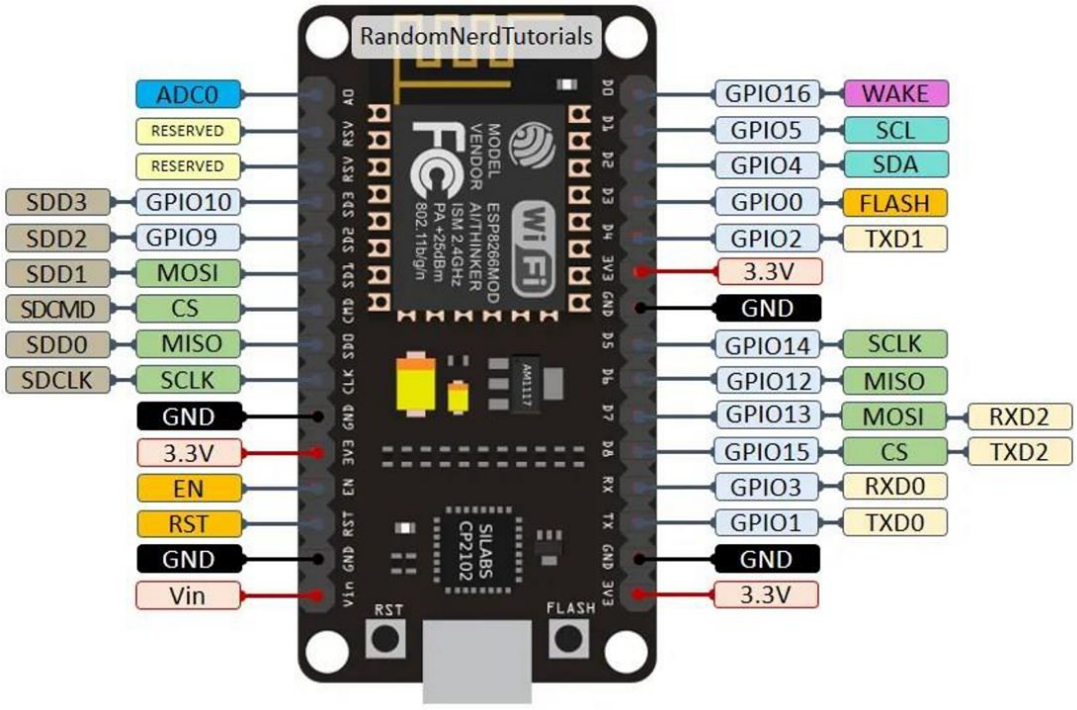

Figure no.3.5: ESP8266 12-E NodeMCU Kit Pinout Diagram

Pins to Use – ESP8266 :

1. 3.3V (3.3V Power): This pin provides a regulated 3.3V power supply, which is used to power the ESP8266 module and other connected components.

2. GND (Ground): These are ground pins, used as the reference voltage for all other components connected to the ESP8266.

3. Vin (Voltage In): You can connect an external power source (typically 5V) to this pin if you want to power the ESP8266 module via an external supply. However, the onboard voltage regulator can handle this automatically when powered via USB or the onboard power supply.

4. RST (Reset): This pin is used to reset the ESP8266 module, similar to the reset button on your computer.

5. EN (Enable/Chip Enable): This pin is connected to the Enable pin on the ESP8266. It's used to enable or disable the module.

6. TX (Transmit) and RX (Receive): These pins are used for serial communication. You can use them to connect to other serial devices like USB-to-Serial converters or other microcontrollers.

7. GPIO Pins (General-Purpose I/O): These pins, labeled as GPIO0 to GPIO16, are general-purpose digital input/output pins. You can use them to connect sensors, LEDs, or other digital components.

8. ADC (Analog-to-Digital Converter): The ESP8266 has one analog input pin (usually labeled as A0 or ADC) that can be used for analog voltage measurements.

9. SCL and SDA (I2C Pins): These pins are used for I2C communication, a two-wire serial communication protocol. They are often used to connect various sensors and displays.

10. SCK, MISO, MOSI, and SS (SPI Pins): These pins are used for SPI (Serial Peripheral Interface) communication, which is commonly used to interface with devices like displays, SD cards, and other microcontrollers.

11. LED_BUILTIN (Onboard LED): This pin is connected to the onboard LED, which you can control using code.

12. Antenna Connection: The ESP8266 module may have an external antenna connector to improve Wi-Fi range. Some modules have an internal PCB antenna.

3.3.2: HC-SR04 ULTRASONIC SENSOR :

HC-SR-04 has an ultrasonic transmitter, receiver and control circuit. In the ultrasonic module HCSR04, we have to give trigger pulse, so that it will generate ultrasound of frequency 40 kHz. After generating ultrasound i.e. 8 pulses of 40 kHz, it makes echo pin high. Echo pin remains high until it does not get the echo sound back. So the width of echo pin will be the time for sound to travel to the object and return back. Once we get the time we can calculate distance, as we know the speed of sound. Ultrasonic Sensor HC-SR04 can measure up to range from 2 cm - 400 cm.

An ultrasonic sensor is an electronic device that measures the distance of a target object by emitting ultrasonic sound waves, and converts the reflected sound into an electrical signal. Ultrasonic waves travel faster than the speed of audible sound (i.e. the sound that humans can hear). Ultrasonic sensors have two main components: the transmitter (which emits the sound using piezoelectric crystals) and the receiver (which encounters the sound after it has travelled to and from the target).

As the name indicates, ultrasonic / level sensors measure distance by using ultrasonic waves. The sensor head emits an ultrasonic wave and receives the wave reflected back from the target. ultrasonic / level sensors measure the distance to the target by measuring the time between the emission and reception. Ultrasonic devices are used to detect objects and measure distances. Ultrasound imaging or sonography is often used in medicine. In the nondestructive testing of products and structures, ultrasound is used to detect invisible flaws. An ultrasonic sensor generates an analog signal internally. However, the output(s) of an ultrasonic sensor chip could be in multiple formats, including digital and analog.

The following picture shows the HC-SR04 ultrasonic sensor.

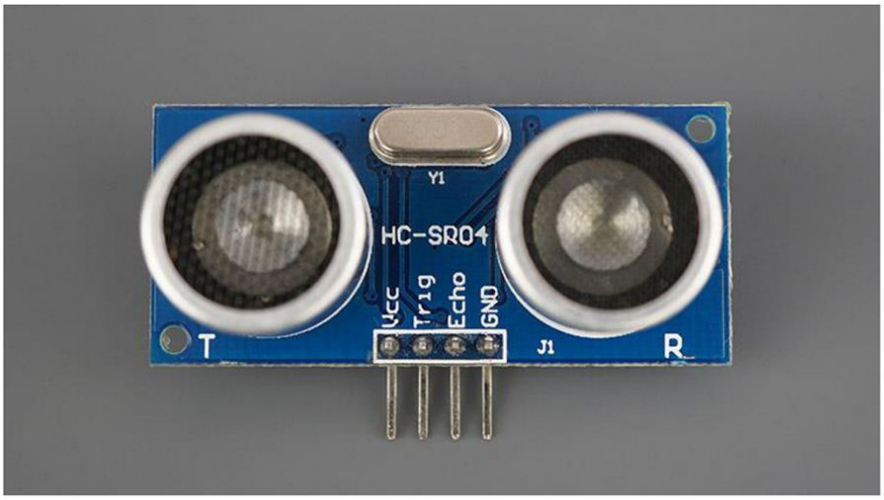

Figure no.3.6: HC-SR04 ultrasonic sensor.

The next picture shows the other side of the sensor.

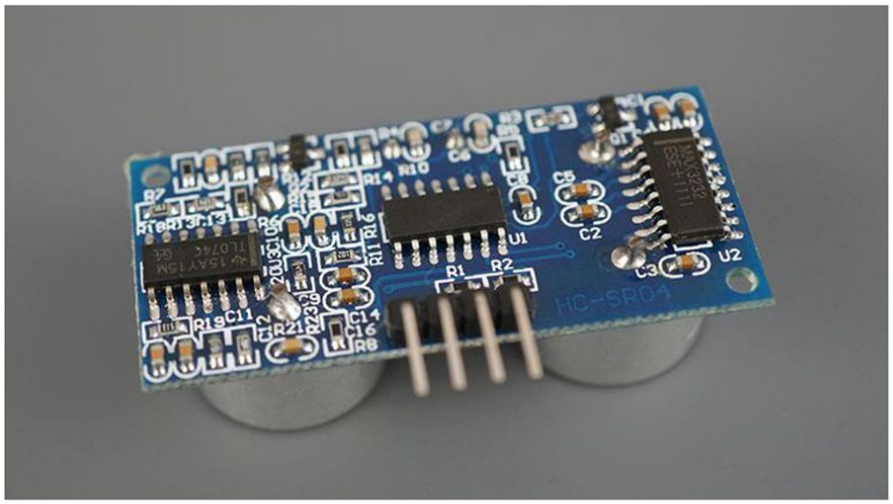

Figure no.3.7: Other side of the HC-SR04 ultrasonic sensor.

FEATURES

- Power Supply :+5V DC
- Quiescent Current : <2mA
- Working Current: 15mA
- Effectual Angle: <15°
- Ranging Distance : 2cm – 400 cm/1″ – 13ft
- Resolution : 0.3 cm

- Measuring Angle: 30 degree
- Trigger Input Pulse width: 10uS TTL pulse
- Echo Output Signal: TTL pulse proportional to the distance range
- Dimension: 45mm x 20mm x 15mm.

3.3.2.1 HC-SR04 ULTRASONIC SENSOR PINOUT CONFIGURATION :

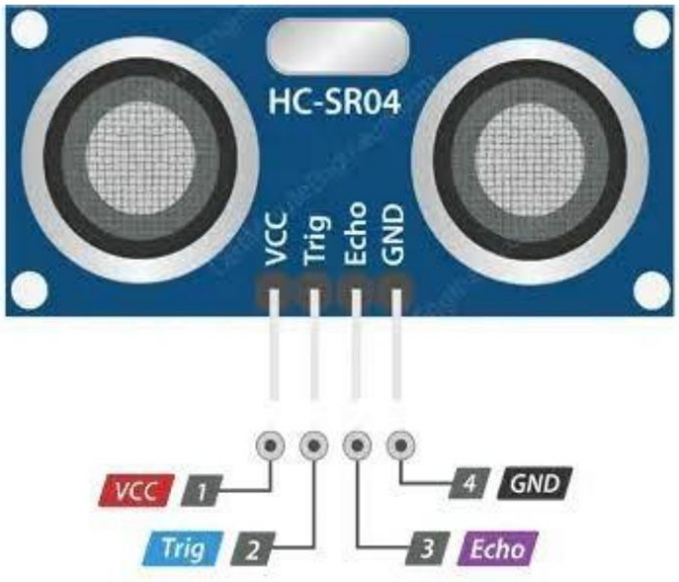

Figure no.3.8: HC-SR04 Ultrasonic Sensor Pinout Configuration

Pin Number	Pin Name	Description
1	Vcc	The Vcc pin powers the sensor, typically with +5V
2	Trigger	Trigger pin is an Input pin. This pin has to be kept high for 10us to initialize measurement by sending US wave.
3	Echo	Echo pin is an Output pin. This pin goes high for a period of time which will be equal to the time taken for the US wave to return back to the sensor.
4	Ground	This pin is connected to the Ground of the system.

3.3.2.2 HC-SR04 Ultrasonic Sensor – Working :

HC-SR04 Ultrasonic (US) sensor is a 4 pin module, whose pin names are Vcc, Trigger, Echo and Ground respectively. This sensor is a very popular sensor used in many applications where measuring distance or sensing objects are required. The module has two eyes like projects in the front which forms the Ultrasonic transmitter and Receiver. The sensor works with the simple high school formula that

Distance = Speed × Time

The Ultrasonic transmitter transmits an ultrasonic wave, this wave travels in air and when it gets objected by any material it gets reflected back toward the sensor this reflected wave is observed by the Ultrasonic receiver module as shown in the picture below

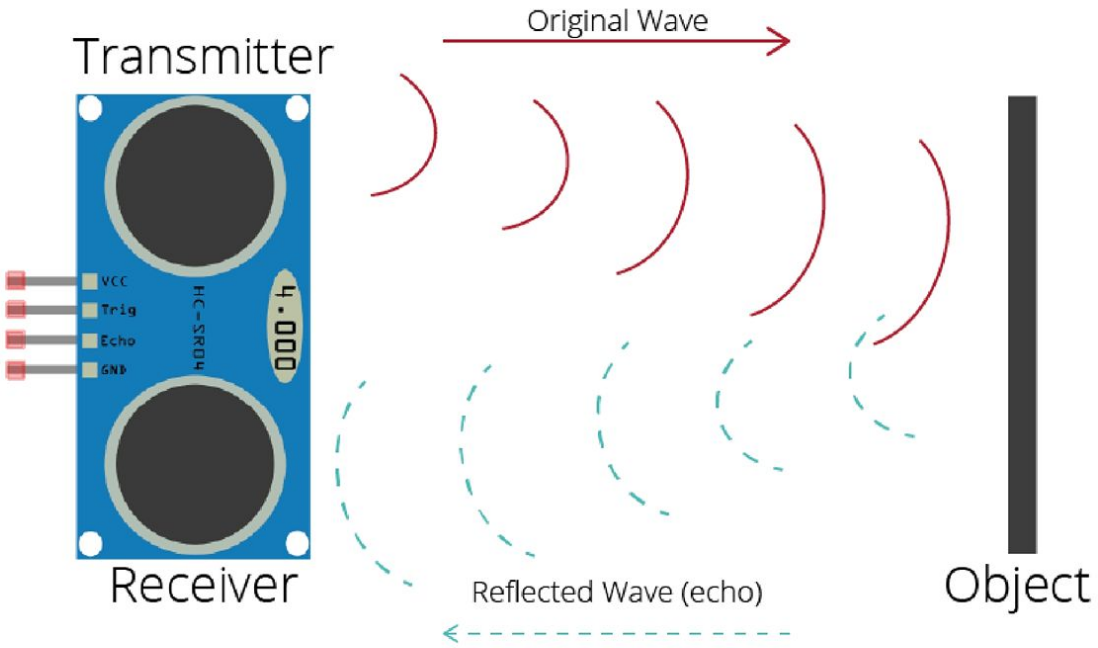

Figure no.3.9: Working of ultrasonic sensor

The ultrasonic sensor uses sonar to determine the distance to an object. Here's what happens:

1. The ultrasound transmitter (trig pin) emits a high-frequency sound (40 kHz).
2. The sound travels through the air. If it finds an object, it bounces back to the module.
3. The ultrasound receiver (echo pin) receives the reflected sound (echo).

The time between the transmission and reception of the signal allows us to calculate the distance to an object. This is possible because we know the sound's velocity in the air. Speed of sound in the air at 20ºC (68ºF) = 343m/s

3.3.4 LCD Display With I2C module :

Figure no.3.10: LCD Display With I2C module.

LCD stands for Liquid Crystal Display. LCD is a flat-paneled display. It uses liquid crystals combined with polarized to display the content. LCD uses the light modulation property of LCD. LCD is available both in Monochrome and Multicolor. It cannot emit light directly without a backlight. In some LCDs, It displays the content only with the help of a backlight in a dark place.

I2C communication: I2C or IIC stands for Inter-Integrated Communication. I2C is a serial communication interface to communicate with other I2C devices. I2C uses multi-master / multi slave method. I2C uses 2 lines named SCL and SDA for transmission/reception and another 2 lines for power supply and ground. Each and every I2C device has I2C address to identify. I2C addresses of multiple devices may have the same address. The address is in the format of "0x20" (Example address). **I2C LCD Adapter:** At the heart of the adapter is an 8-bit I/O expander chip – PCF8574. This chip converts the I2C data from an Arduino into the parallel data required for an LCD display.

Figure no.3.11: PCF8574

The board also includes a tiny trimpot for making precise adjustments to the display's contrast.

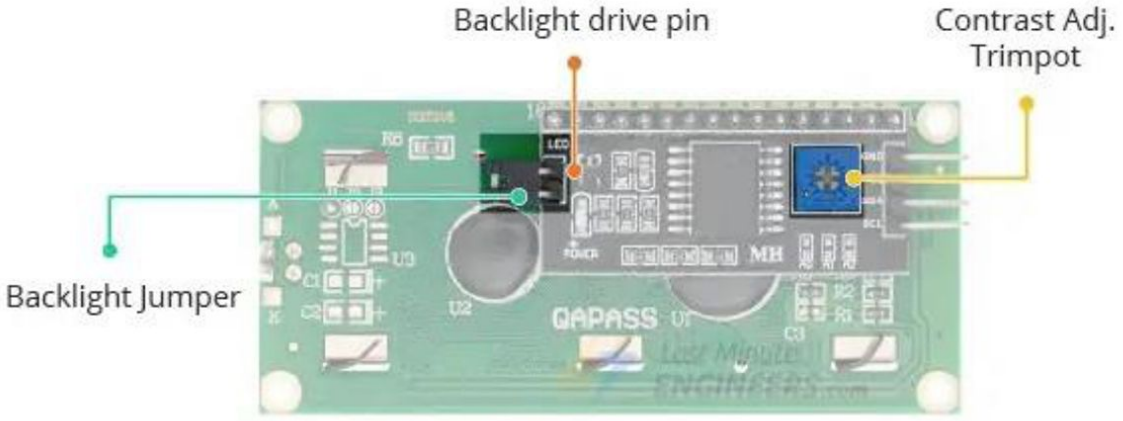

Figure no.3.12: Ports And Backlight

There is a jumper on the board that provides power to the backlight. To control the intensity of the backlight, you can remove the jumper and apply external voltage to the header pin labeled 'LED'.

3.3.4.1.1 I2C LCD Display Pinout Diagram :

The I2C LCD Display has only four pins. The following is the pinout:

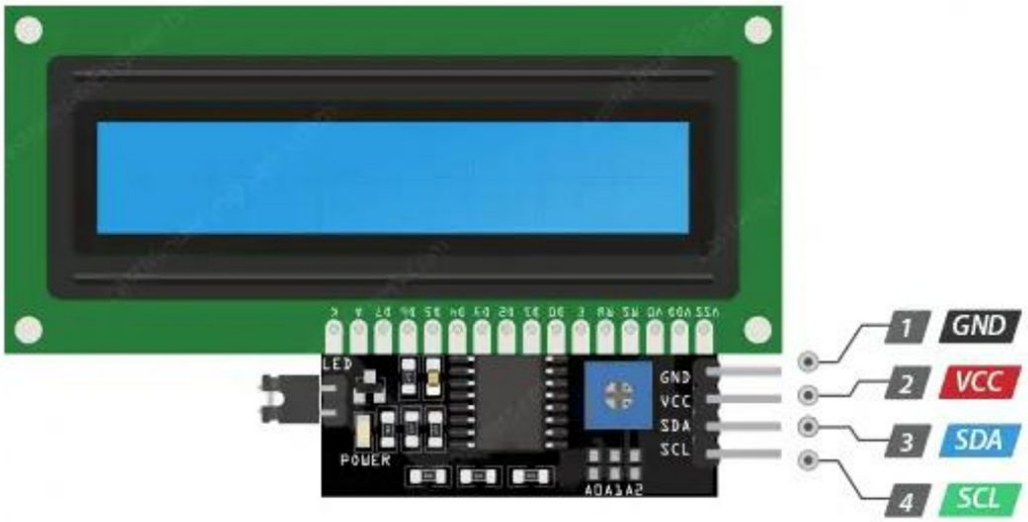

Figure no.3.13: I2C LCD Display Pinout Diagram

1. **GND** is a ground pin.
2. **VCC** is the power supply pin. Connect it to the 5V output of the Arduino or an external 5V power supply.
3. **SDA** is the I2C data pin.
4. **SCL** is the I2C clock pin.

3.3.4.1.2 I2C Serial Interface Module Pinout Diagram:

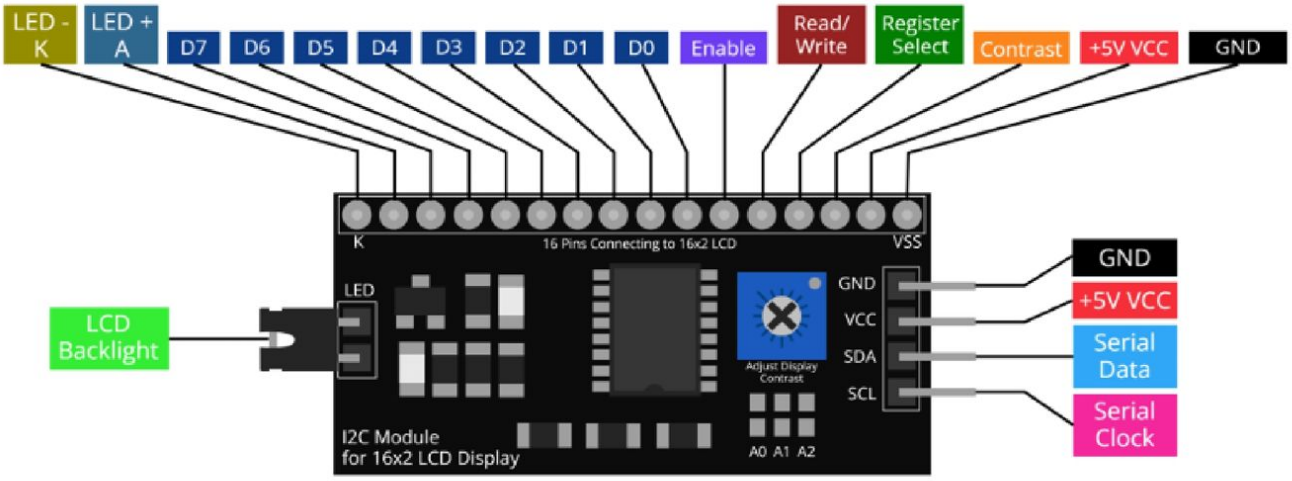

Figure no.3.14: I2C Serial Interface Module Pinout Diagram

3.3.5 Resistor (220 ohm) :

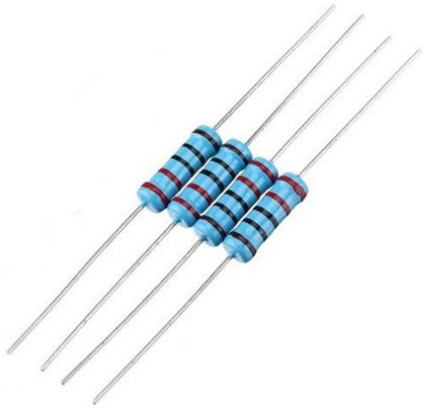

Figure no.3.15: Resistor

A 220-ohm resistor is an electronic component that is used to resist the flow of electricity in a circuit. Resistors are used in a wide variety of electronic circuits to control the flow of current and protect other components from damage. 220-ohm resistors are a commonly used resistance value in electronic circuits.

A 220 Ω resistor has a silver tolerance band. Tolerance = value of resistor x value of tolerance band = 220 Ω x 10% = 22 Ω 220 Ω stated resistance +/- 22 Ω tolerance means that the resistor could range in actual value from as much as 242 Ω to as little as 198 Ω. Variable resistors can be used to adjust circuit elements (such as a volume control or a lamp dimmer), or as sensing devices for heat, light, humidity, force.

3.3.6 light emitting diode (LED Bulb) :

Figure no.3 16: light emitting diode (LED Bulb)

A light-emitting diode (LED) is a semiconductor device that emits light when current flows through it. Electrons in the semiconductor recombine with electron holes, releasing energy in the form of photons. The color of the light (corresponding to the energy of the photons) is determined by the energy required for electrons to cross the band gap of the semiconductor.[5] White light is obtained by using multiple semiconductors or a layer of light-emitting phosphor on the semiconductor device.

A Light Emitting Diode (LED) is a semiconductor device, which can emit light when an electric current passes through it. To do this, holes from p-type semiconductors recombine with electrons from n-type semiconductors to produce light. The major uses of LED (Light Emitting Diodes) are to illuminate objects and even places. Its application is everywhere due to its compact size, low consumption of energy, extended lifetime, and flexibility in terms of use in various applications.

3.3.7 CONNECTING WIRES(JUMPER WIRES)

3.3.7.1 Male To Male

Figure no.3.17: Jumper Wires - Male To Male

This is male to male Dupont/Jumper Wire 40 Pin 40cm. A very Flexible and easily detachable cable to the no. of wires according to your requirement. It has 1Pin male to the 1pin male header with both ends. Also, it is compatible with 2.54 mm mil spacing pin headers.

This cable is an electrical wire or group of them in a cable with a connector or pins at each end, which is normally for interconnecting the components of a breadboard or other prototype or test circuit, internally or with other equipment or components, without soldering.

Individual Dupont Cables are fitted by inserting their “end connectors” into the slots provided in a breadboard, the header connector of a circuit board, or a piece of test equipment.

Mostly it is useful with Orange Pi, Banana Pi, Raspberry Pi, Arduino, and other mini pc and development board. It is very useful in the PCB project, pc motherboard, as well as Breadboard connections. Additionally, it allows you to plug and unplug easily for prototyping and can be used over and over again.

Features :

- Compatible with 2.54mm spacing pin headers
- 40pcs chromatic color jump wire
- High quality and in good working condition
- Durable and reusable
- Easy to install and use
- A popular choice for construction or repair
- Be used for the electronic project and Genuine Arduino product
- Flexible Breadboard Jumper Cable Wire allows you to plug and unplug easily for prototyping.

3.3.7.2 Male To Female

Figure no.3.18: Jumper Wires - Male To female

These are male to female jumper wires used in connecting the female header pin of any development board to other development boards having a male connector. They are simple wires that have connector pins at each end allowing them to be used to connect two points to each other. Jumper wires are typically used with breadboards and other prototyping tools in order to make it easy to change a circuit as needed.

40 strip Male to female jumper wire each cable length about 20cm or 8-inch.

Features

- It connects two points to each other without soldering
- It is reusable
- It is inexpensive and easy to use.

Applications

It is used to interconnect the components of a breadboard or other prototype or test circuit internally or with other equipment or components without soldering.

3.3.8 BREADBOARD

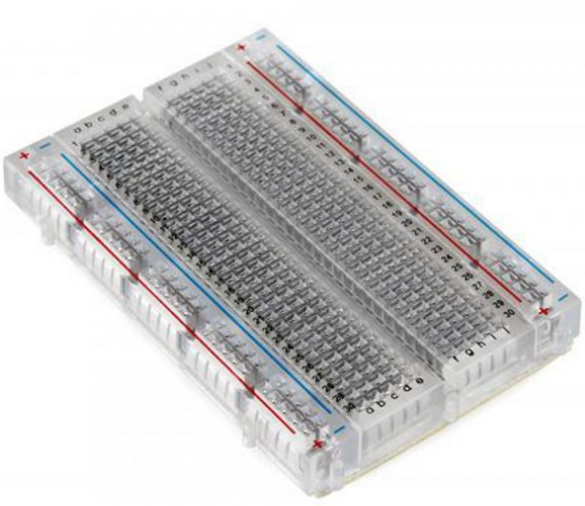

Figure no.3.19: Breadboard

A breadboard (sometimes called a plugblock) is used for **building temporary circuits**. It is useful to designers because it allows components to be removed and replaced easily. It is useful to the person who wants to build a circuit to demonstrate its action, then to reuse the components in another circuit.

Breadboards are one of the most fundamental pieces when learning how to build circuits. In this tutorial, you will learn a little bit about what breadboards are, why they are called breadboards, and how to use one. Once you are done you should have a basic understanding of how breadboards work and be able to build a basic circuit on a breadboard.

3.4 SOFTWARE IMPLEMENTATION

3.4.1 Arduino IDE

The Arduino IDE is an open-source software, which is used to write and upload code to the Arduino boards. The IDE application is suitable for different operating systems such as Windows, Mac OS X, and Linux. It supports the programming languages C and C++. Here, IDE stands for Integrated Development Environment.

The program or code written in the Arduino IDE is often called as sketching. We need to connect the Genuino and Arduino board with the IDE to upload the sketch written in the Arduino IDE software. The sketch is saved with the extension '.ino.'

The Arduino IDE will appear as:

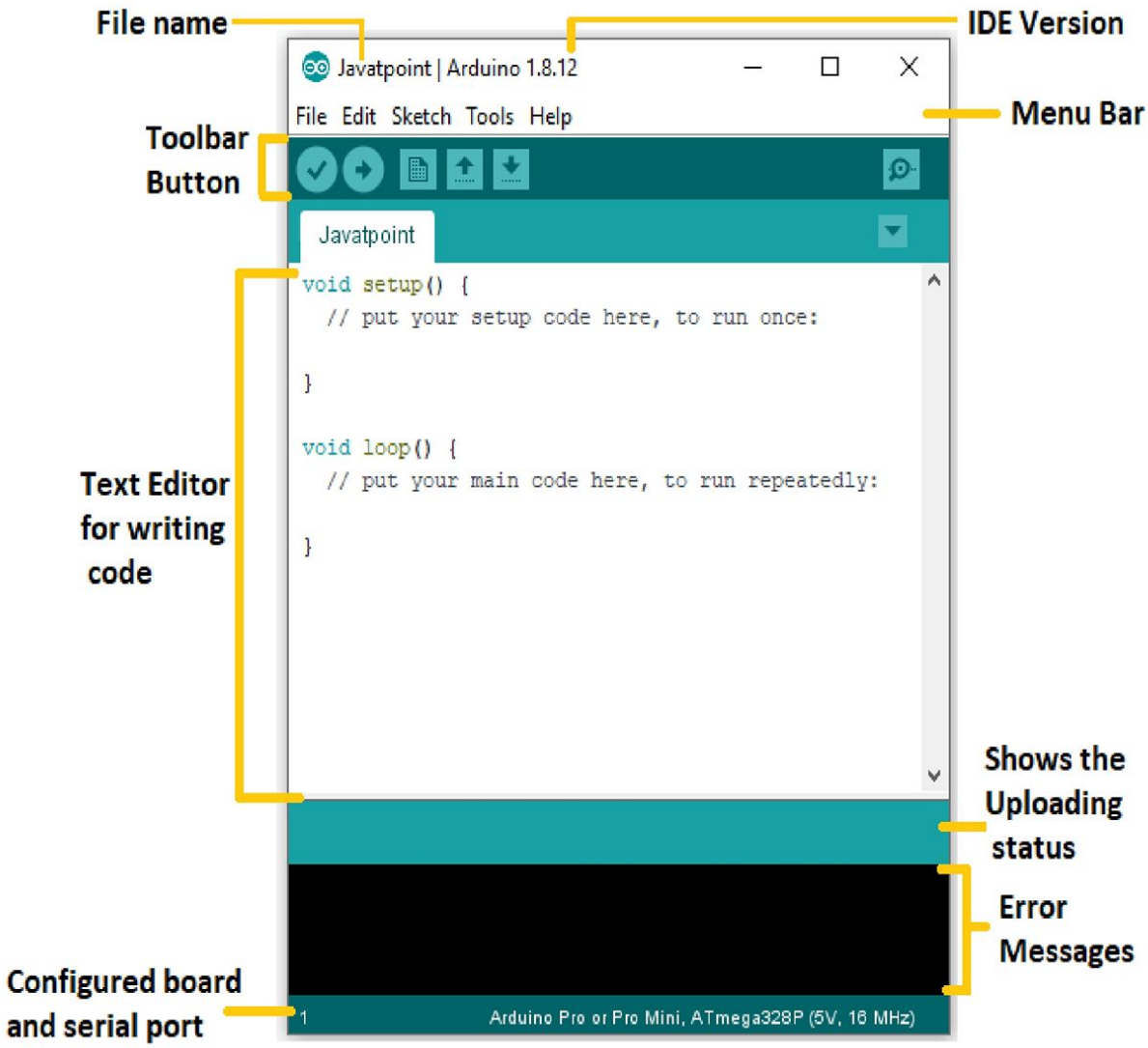

Figure no.3.20: Arduino IDE

Let's discuss each section of the Arduino IDE display in detail.

Play Video

Toolbar Button

The icons displayed on the toolbar are New, Open, Save, Upload, and Verify.

It is shown below:

Figure no.3.21: Toolbar

Upload

The Upload button compiles and runs our code written on the screen. It further uploads the code to the connected board. Before uploading the sketch, we need to make sure that the correct board and ports are selected.

We also need a USB connection to connect the board and the computer. Once all the above measures are done, click on the Upload button present on the toolbar.

3.4.2 WIFI LIBRARY ESP8266:

ESP8266 library for Arduino that facilitates the implementation of WiFi communication via user sketches. ESP8266 is a very powerful and 802.11b/g/n protocol based low cost WiFi module. It is contained with a sufficient size of EEPROM and a 32-bit MPU necessary to TCP/IP protocol stack built-in. You can easily build a WiFi device with a serial communication from physical computing boards such as Arduino. ESP8266 WiFi Library for Arduino provides a function for easily WiFi communication using ESP8266 from your sketch via the serial on such as Arduino UNO, Leonardo and MEGA. Also this library has a debug output facility can monitor the transmitted and received data. The ESP8266WiFi library provides a wide collection of C++ methods (functions) and properties to configure and operate an ESP8266 module in station and / or soft access point mode.

3.4.3 I2C LIBRARY :

The library allows to control I2C displays with functions extremely similar to LiquidCrystal library. The LiquidCrystal_I2C library allows you to program an Arduino to print messages to an LCD screen using an I2C backpack. I2C is a communication protocol that allows you to communicate with multiple devices using only a few pins.

3.4.4 BLYNK LIBRARY:

Blynk Library is an extension that runs on your hardware. It handles connectivity, device authentication in the cloud, and commands processing between Blynk app, Cloud, and hardware. Blynk is an IoT platform for iOS or Android smartphones that is used to control Arduino, Raspberry Pi and NodeMCU via the Internet. This application is used to create a graphical interface or human machine interface (HMI) by compiling and providing the appropriate address on the available widgets.

3.4.5 ADAFRUIT GFX LIBRARY

This is the core graphics library for all our displays, providing a common set of graphics primitives (points, lines, circles, etc.). It needs to be paired with a hardware-specific library for each display device we carry (to handle the lower-level functions).
Adafruit invests time and resources providing this open source code, please support Adafruit and open-source hardware by purchasing products from Adafruit!

CHAPTER 4
METHODOLOGY

4.1 CODE:

```
/*Water level monitoring system with the New Blynk app
   https://srituhobby.com
*/
//Include the library files
#include <LiquidCrystal_I2C.h>
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

//Initialize the LCD display
LiquidCrystal_I2C lcd(0x27, 16, 2);

char auth[] = "";//Enter your Auth token
char ssid[] = "";//Enter your WIFI name
char pass[] = "";//Enter your WIFI password

BlynkTimer timer;

// Define the component pins
#define trig D7
#define echo D8
#define LED1 D0
#define LED2 D3
#define LED3 D4
#define LED4 D5
#define LED5 D6
#define relay 3

//Enter your tank max value(CM)
int MaxLevel = 20;

int Level1 = (MaxLevel * 75) / 100;
int Level2 = (MaxLevel * 65) / 100;
int Level3 = (MaxLevel * 55) / 100;
int Level4 = (MaxLevel * 45) / 100;
int Level5 = (MaxLevel * 35) / 100;

void setup() {
  Serial.begin(9600);
  lcd.init();
  lcd.backlight();
  pinMode(trig, OUTPUT);
  pinMode(echo, INPUT);
  pinMode(LED1, OUTPUT);
  pinMode(LED2, OUTPUT);
```

```
  pinMode(LED3, OUTPUT);
  pinMode(LED4, OUTPUT);
  pinMode(LED5, OUTPUT);
  pinMode(relay, OUTPUT);
  digitalWrite(relay, HIGH);
  Blynk.begin(auth, ssid, pass, "blynk.cloud", 80);

  lcd.setCursor(0, 0);
  lcd.print("Water level");
  lcd.setCursor(4, 1);
  lcd.print("Monitoring");
  delay(4000);
  lcd.clear();

  //Call the functions
  timer.setInterval(100L, ultrasonic);
}
//Get the ultrasonic sensor values
void ultrasonic() {
  digitalWrite(trig, LOW);
  delayMicroseconds(4);
  digitalWrite(trig, HIGH);
  delayMicroseconds(10);
  digitalWrite(trig, LOW);
  long t = pulseIn(echo, HIGH);
  int distance = t / 29 / 2;

  int blynkDistance = (distance - MaxLevel) * -1;
  if (distance <= MaxLevel) {
    Blynk.virtualWrite(V0, blynkDistance);
  } else {
    Blynk.virtualWrite(V0, 0);
  }
  lcd.setCursor(0, 0);
  lcd.print("WLevel:");
  if (Level1 <= distance) {
    lcd.setCursor(8, 0);
    lcd.print("Very Low");
    digitalWrite(LED1, HIGH);
    digitalWrite(LED2, LOW);
    digitalWrite(LED3, LOW);
    digitalWrite(LED4, LOW);
    digitalWrite(LED5, LOW);
  } else if (Level2 <= distance && Level1 > distance) {
    lcd.setCursor(8, 0);
    lcd.print("Low");
    lcd.print("      ");
    digitalWrite(LED1, HIGH);
    digitalWrite(LED2, HIGH);
    digitalWrite(LED3, LOW);
```

```
    digitalWrite(LED4, LOW);
    digitalWrite(LED5, LOW);
  } else if (Level3 <= distance && Level2 > distance) {
    lcd.setCursor(8, 0);
    lcd.print("Medium");
    lcd.print("      ");
    digitalWrite(LED1, HIGH);
    digitalWrite(LED2, HIGH);
    digitalWrite(LED3, HIGH);
    digitalWrite(LED4, LOW);
    digitalWrite(LED5, LOW);
  } else if (Level4 <= distance && Level3 > distance) {
    lcd.setCursor(8, 0);
    lcd.print("High");
    lcd.print("      ");
    digitalWrite(LED1, HIGH);
    digitalWrite(LED2, HIGH);
    digitalWrite(LED3, HIGH);
    digitalWrite(LED4, HIGH);
    digitalWrite(LED5, LOW);
  } else if (Level5 >= distance) {
    lcd.setCursor(8, 0);
    lcd.print("Full");
    lcd.print("      ");
    digitalWrite(LED1, HIGH);
    digitalWrite(LED2, HIGH);
    digitalWrite(LED3, HIGH);
    digitalWrite(LED4, HIGH);
    digitalWrite(LED5, HIGH);
  }
}

//Get the button value
BLYNK_WRITE(V1) {
  bool Relay = param.asInt();
  if (Relay == 1) {
    digitalWrite(relay, LOW);
    lcd.setCursor(0, 1);
    lcd.print("Motor is ON ");
  } else {
    digitalWrite(relay, HIGH);
    lcd.setCursor(0, 1);
    lcd.print("Motor is OFF");
  }
}

void loop() {
  Blynk.run();//Run the Blynk library
  timer.run();//Run the Blynk timer
}
```

CHAPTER 5

IMPLEMENTATION AND RESULTS

5.1 ANDROID SETUP :

Step 1

First, download and install the Blynk app on your smartphone. Then, click the template you created on the web dashboard.

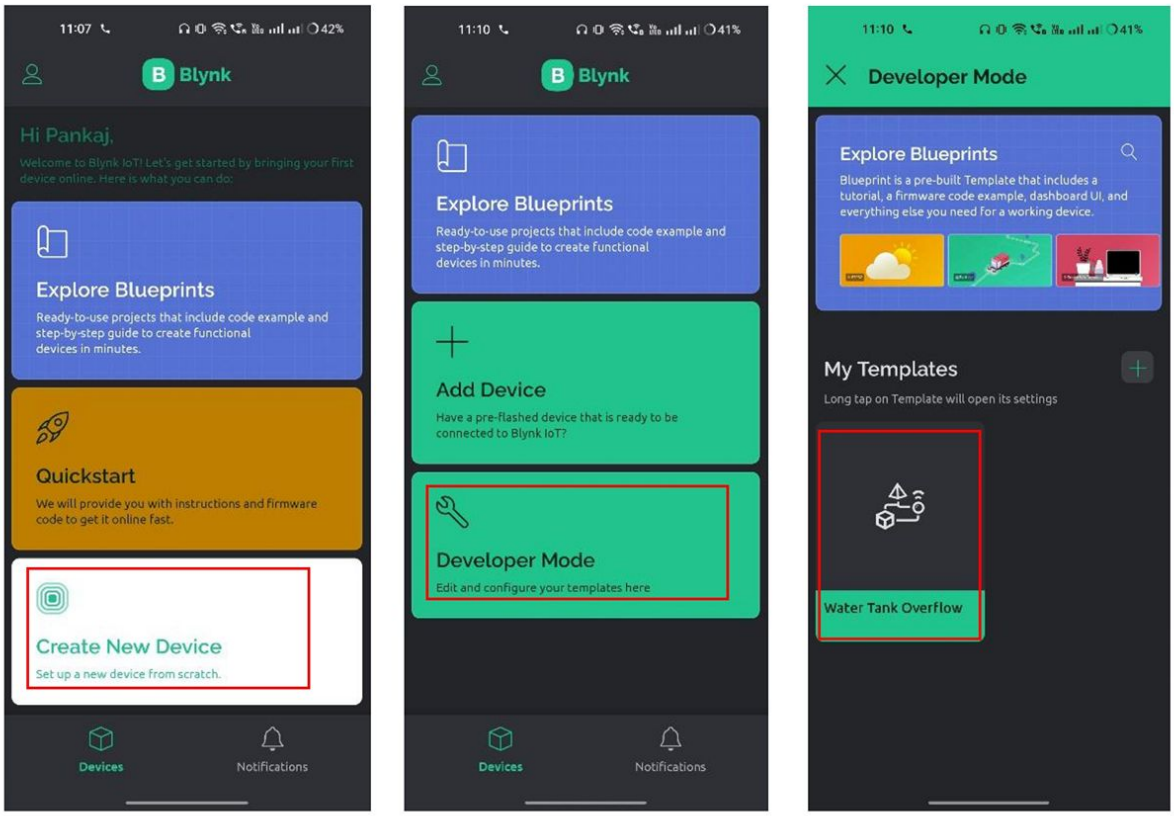

Figure no.5.1: Create New Template

Step 2:

Next, add a button widget and one gauge widget to the dashboard. And then, customize these widgets as you like.

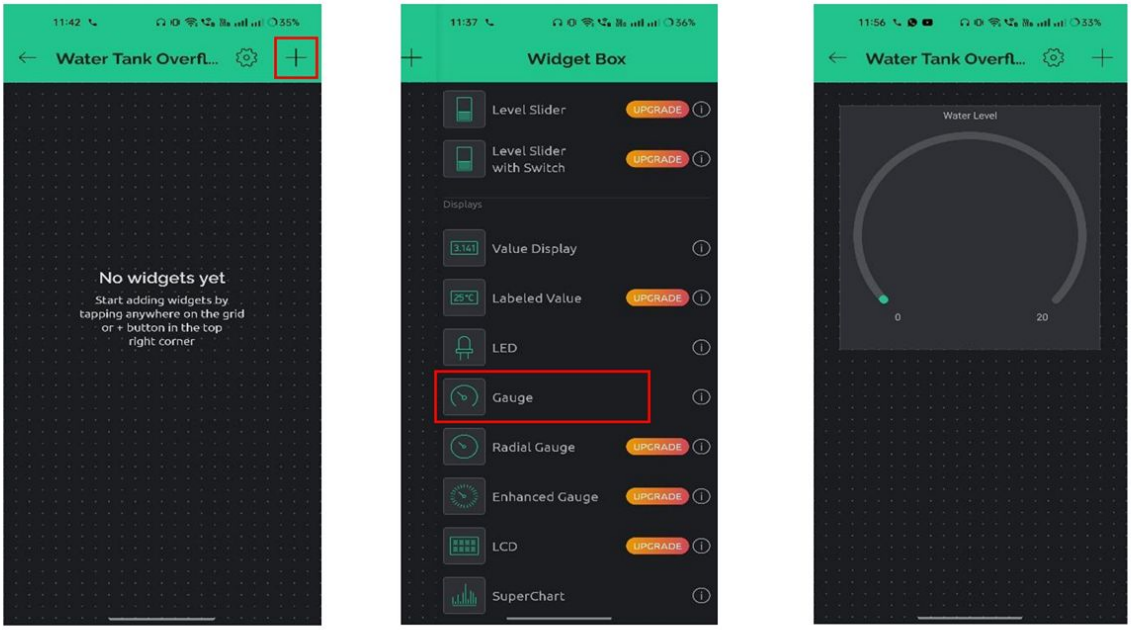

Figure no.5.2: Adding Gauge

Step 3:

Now, select the "water level" data stream for the gauge widget and then, name these widgets as you like.

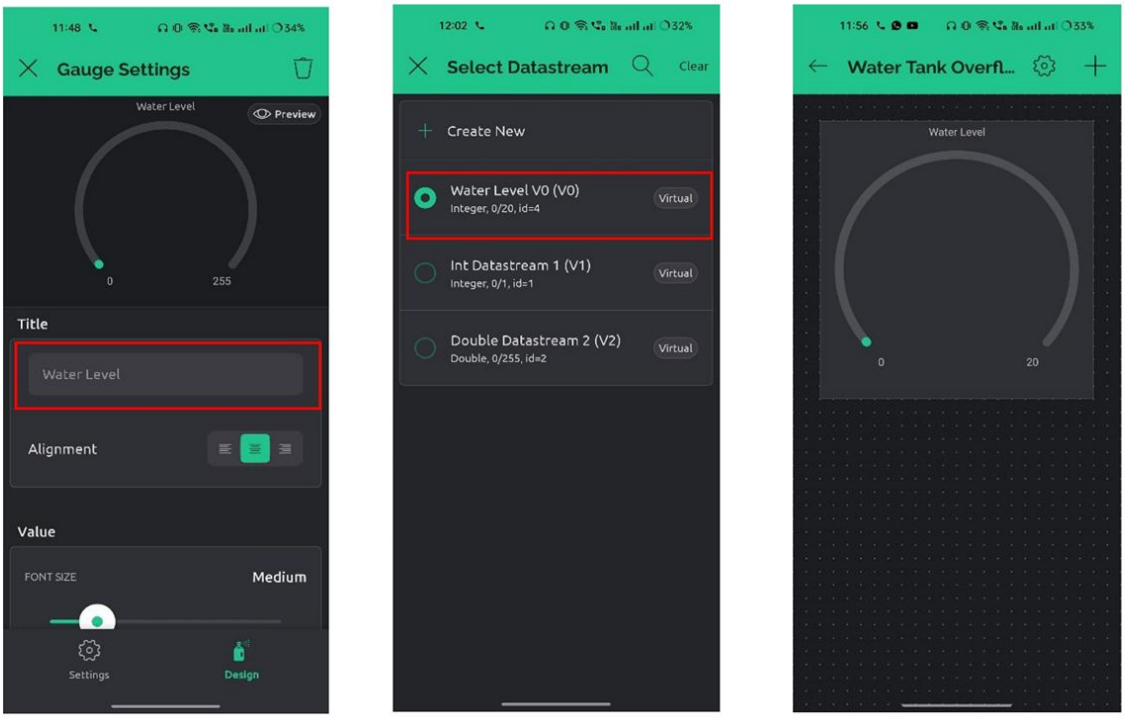

Figure no.5.3: Adding Title And Datastream

5.2 WEB BASED SETUP :

For setting up Blynk web app, follow the instructions below.

Step 1 :
First, go to the Blynk website and create a new account using your email address. And then, log in to your account and create a new template for this project.

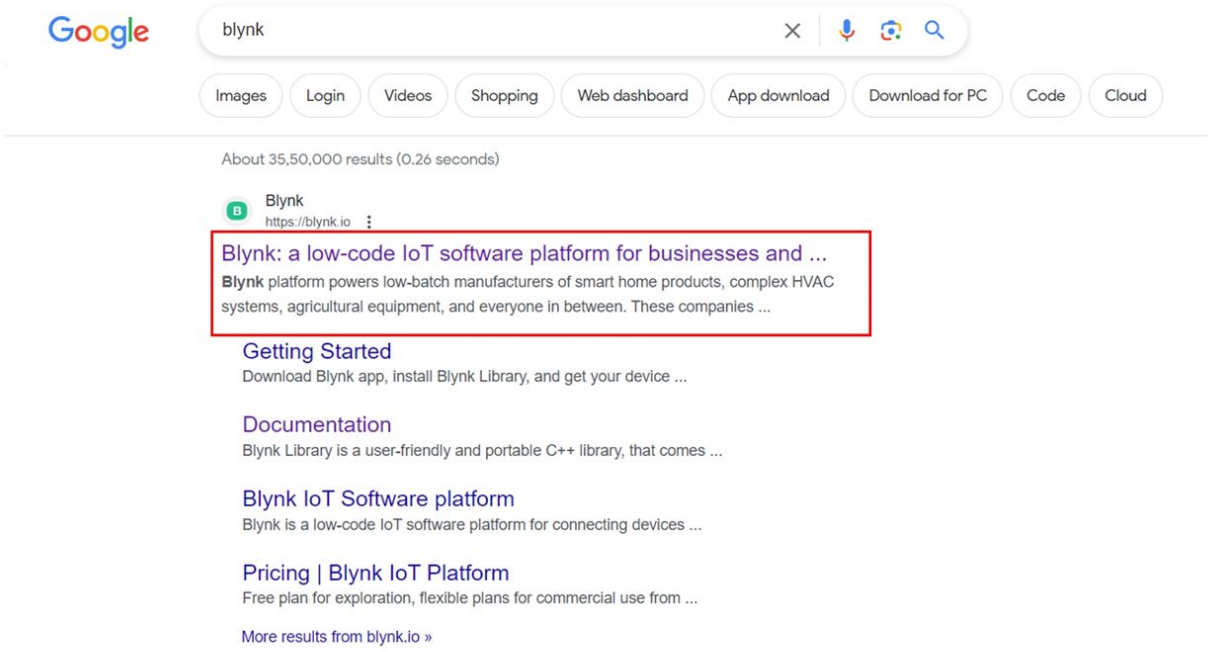

Figure no.5.4: Link To Website

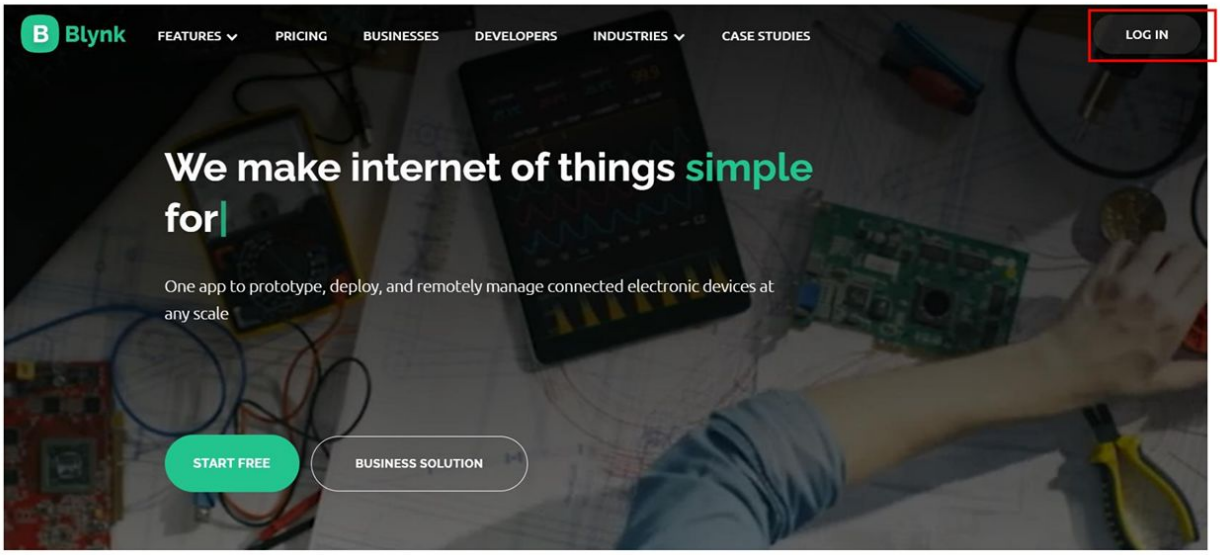

Figure no.5.5: Login To BLYNK

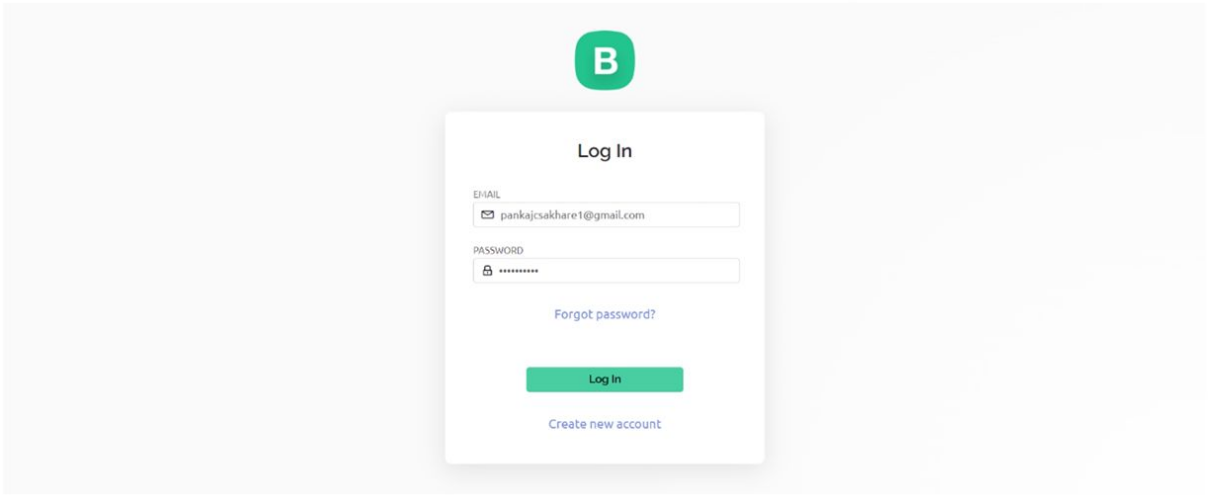

Figure no.5.6: Use Email To Login

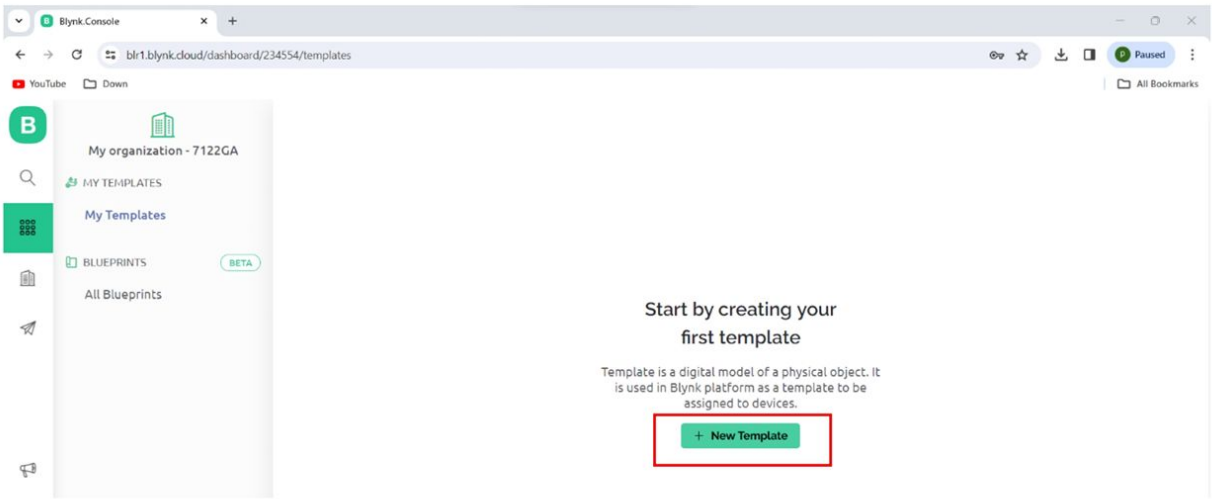

Figure no.5.7: Create A New Template

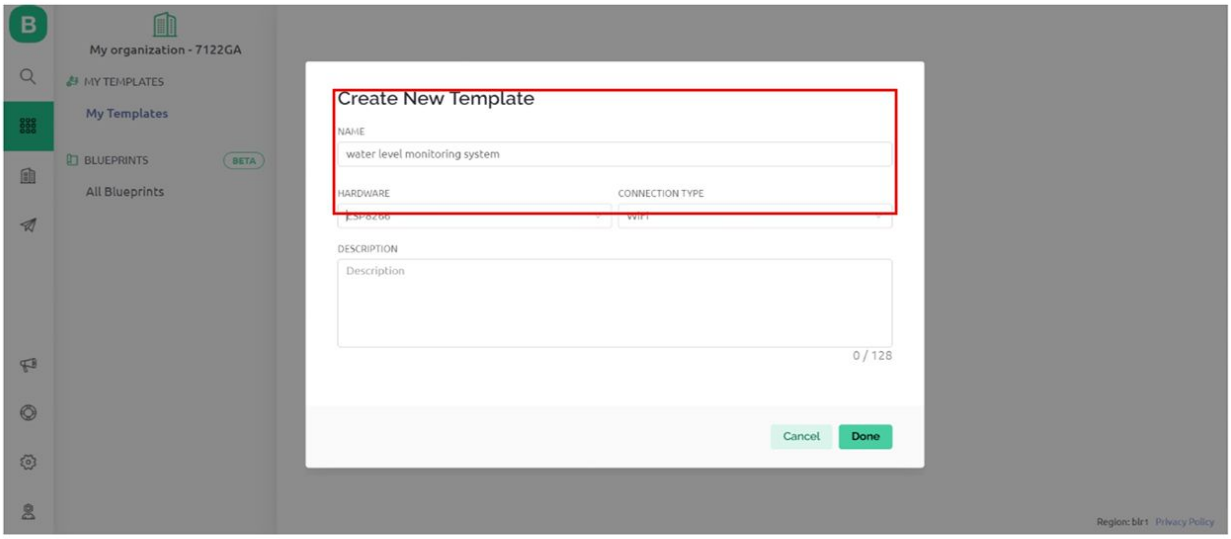

Figure no.5.8: Add Template Name

Now, click the datastream tab and create two data streams for that.
Virtual PIN / Name — Water level / PIN — V0 / MIN — 0 / MAX — 20 (The height of the water tank)
Virtual PIN / Name — Water pump / PIN — V1 / MIN — 0 / MAX — 1

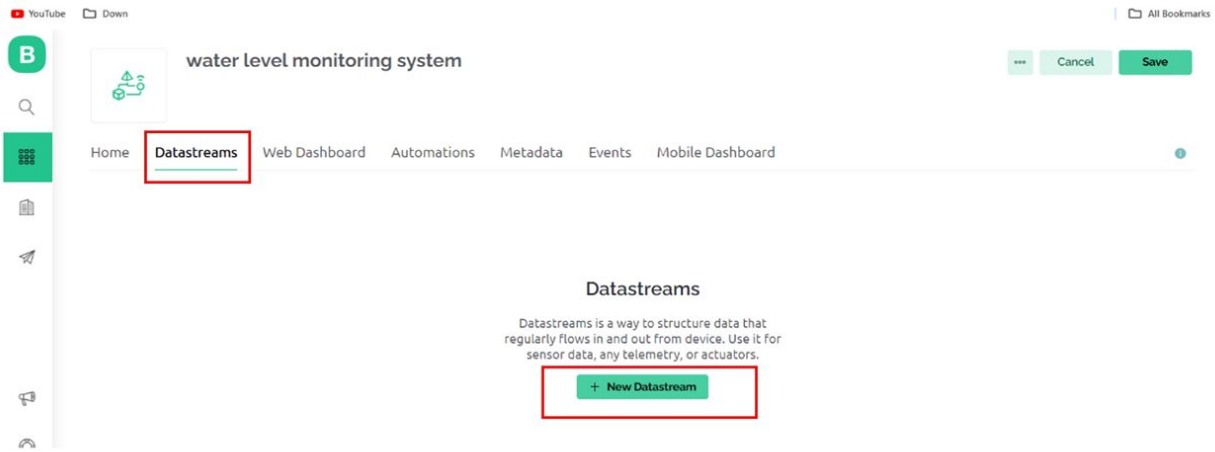

Figure no.5.9: Create Datastream

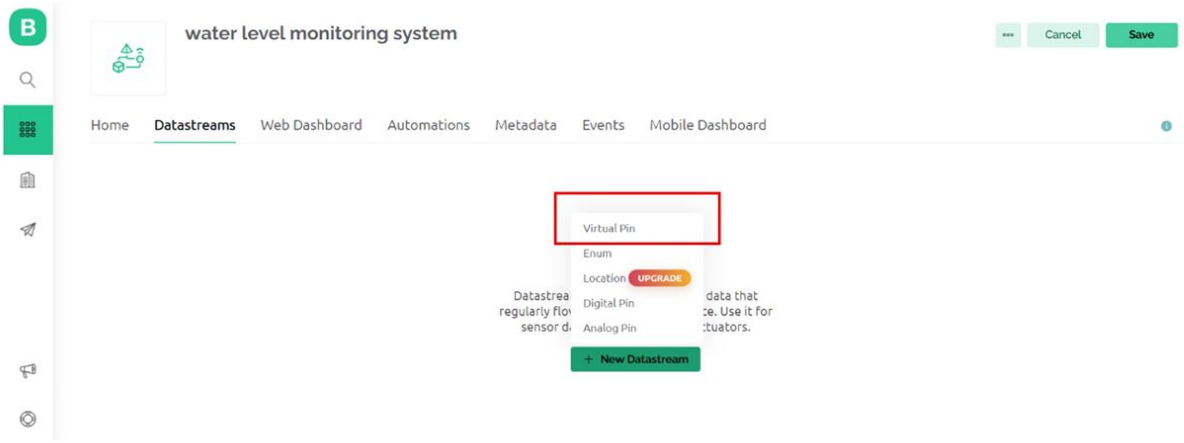

Figure no.5.10: Select Virtual Pin

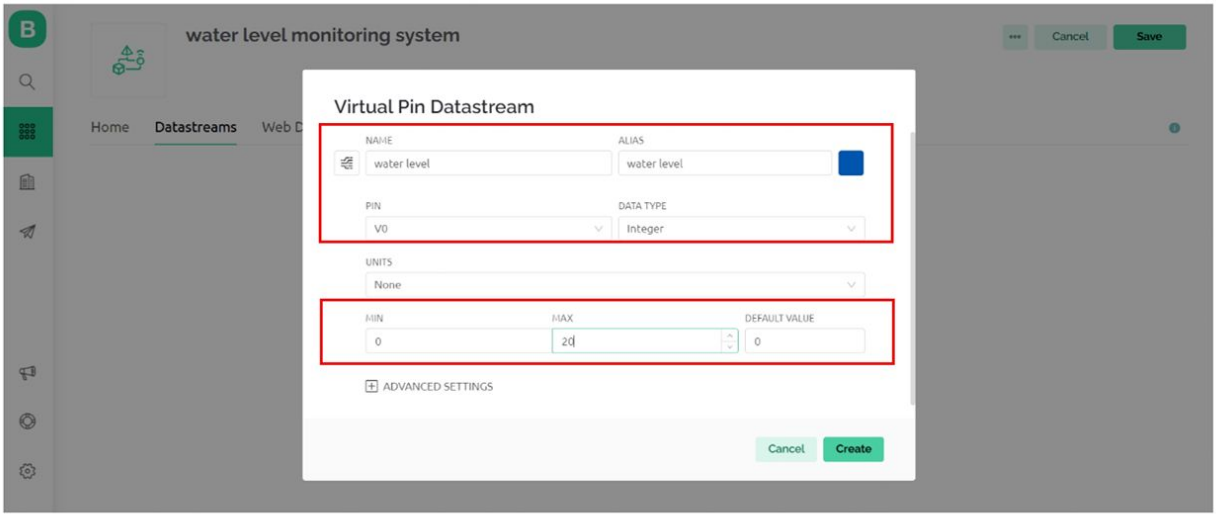

Figure no.5.11: Set Virtual Pin

After, click the web dashboard tab and create a suitable web dashboard for this project. For that, I used a button and one gauge widget.

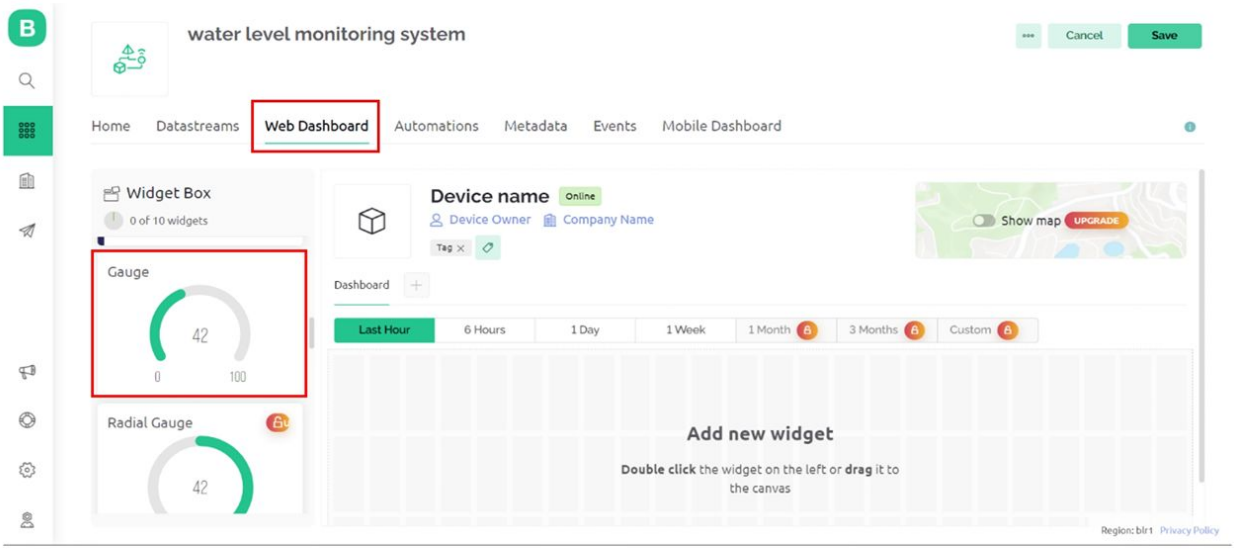

Figure no.5.12: Go To Webdashboard And Add Gauge

Now, click the one-by-one setting buttons on the widgets and select the water level data stream for the gauge widget and the water pump data stream for the button widget. Finally, click the save button.

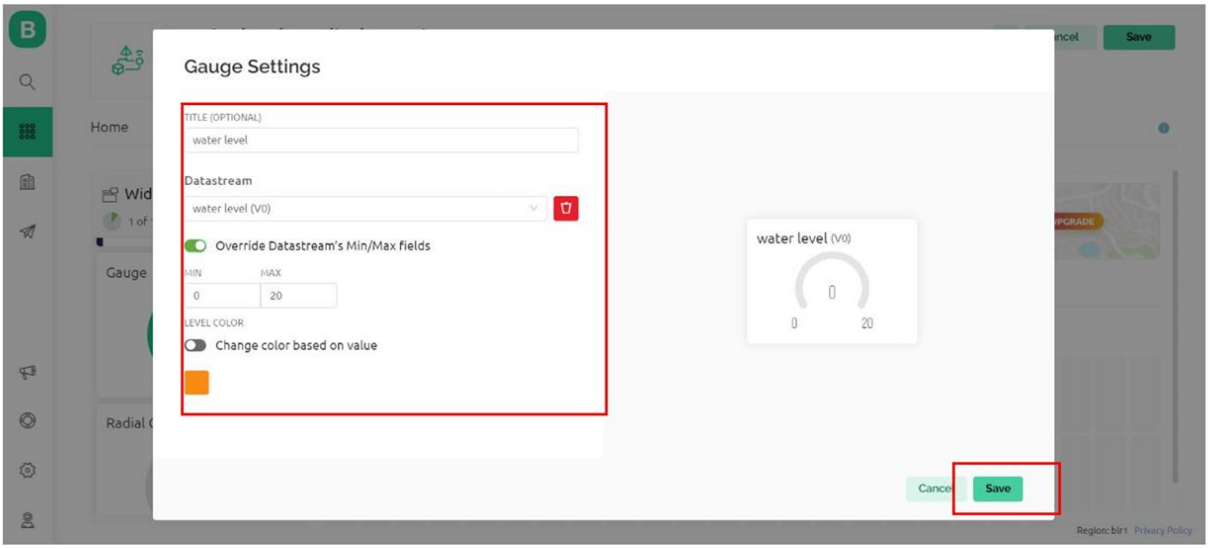

Figure no.5.13: Set Gauge Setting

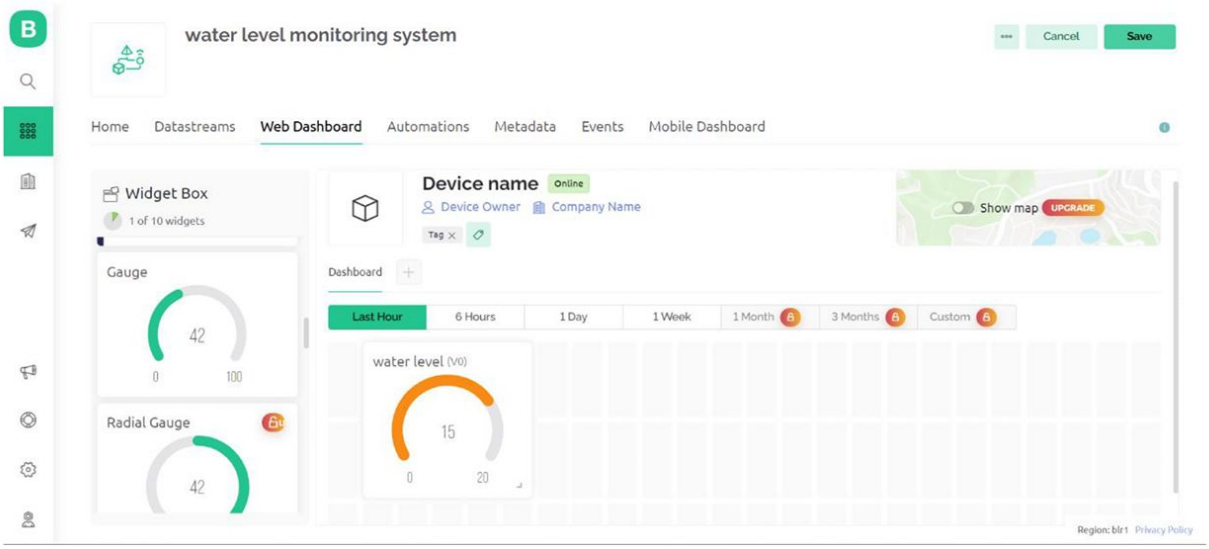

Figure no.5.14: Screen Of Web Dashboard

Next, click the search icon button and create a new device. For that, select the template you created earlier.

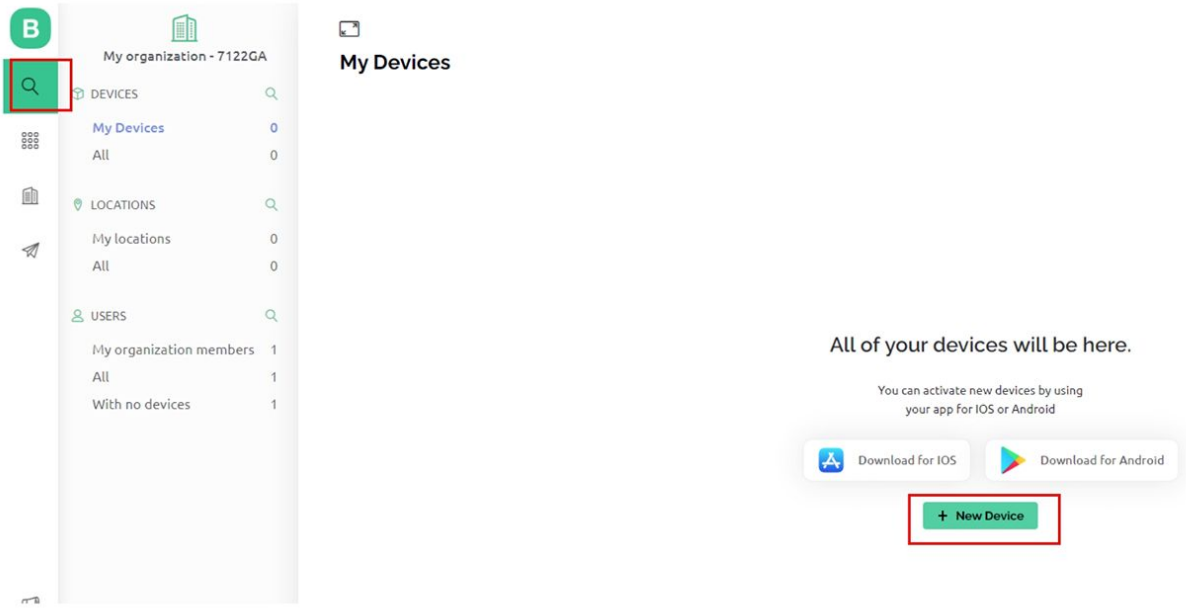

Figure no.5.15: Create New Device

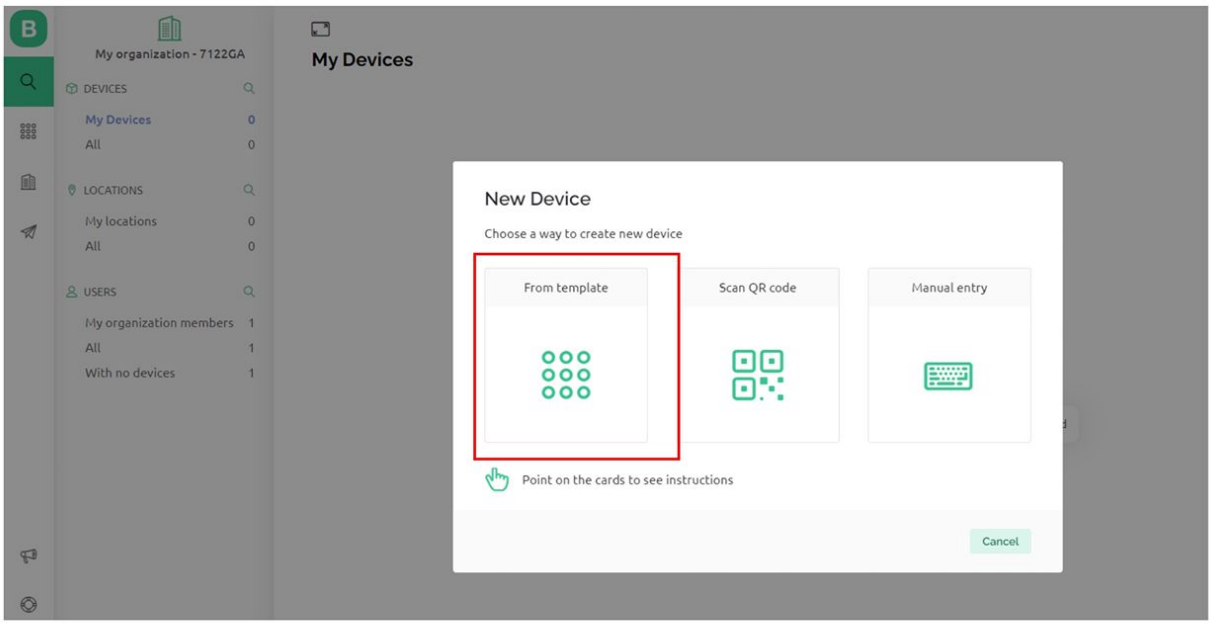

Figure no.5.16: Way To Create New Device

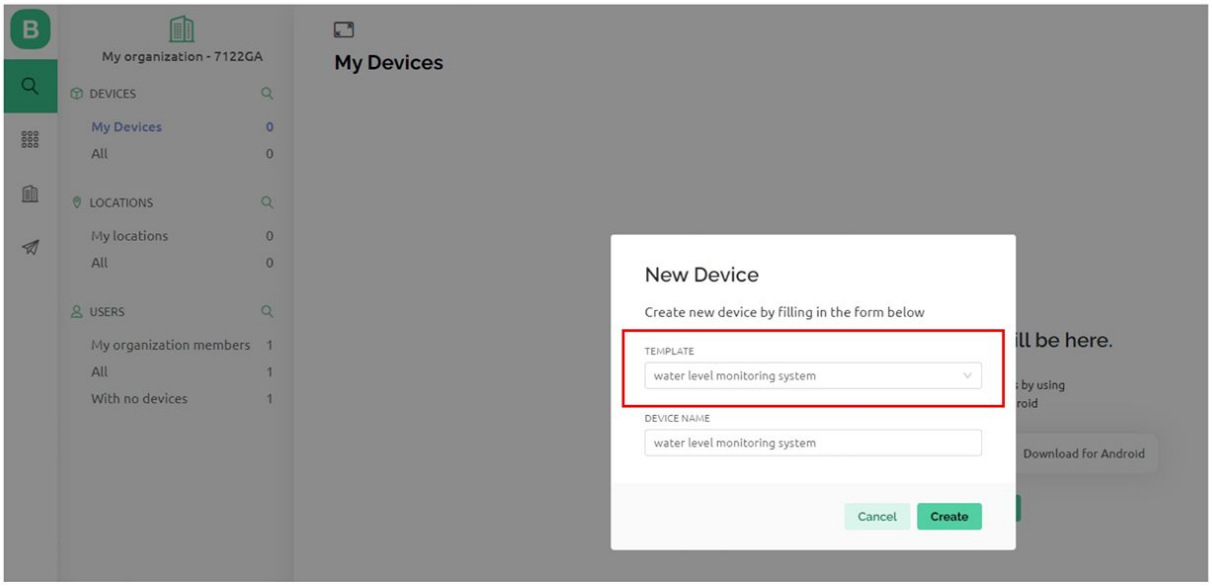

Figure no.5.17: Select Template

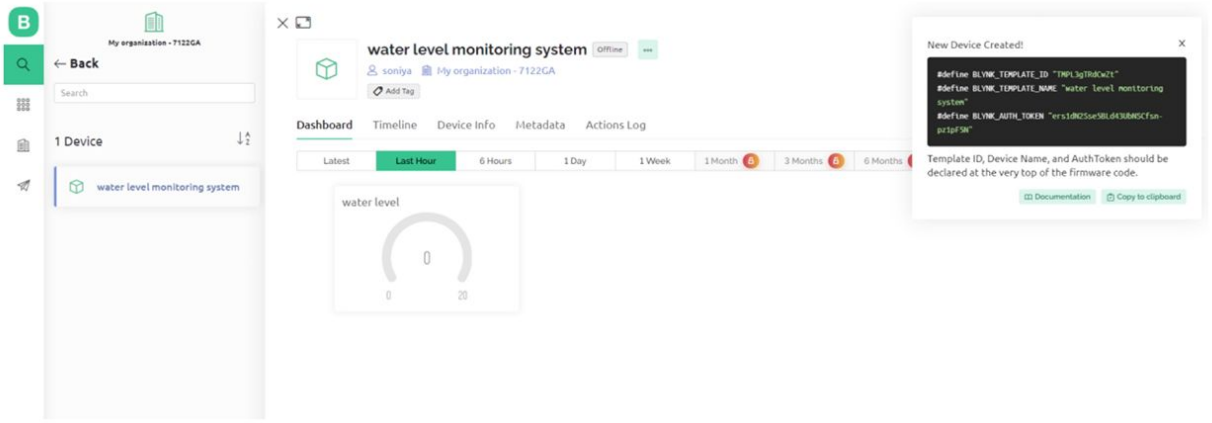

Figure no.5.18: For Auth Token And Id's

5.3 Output:

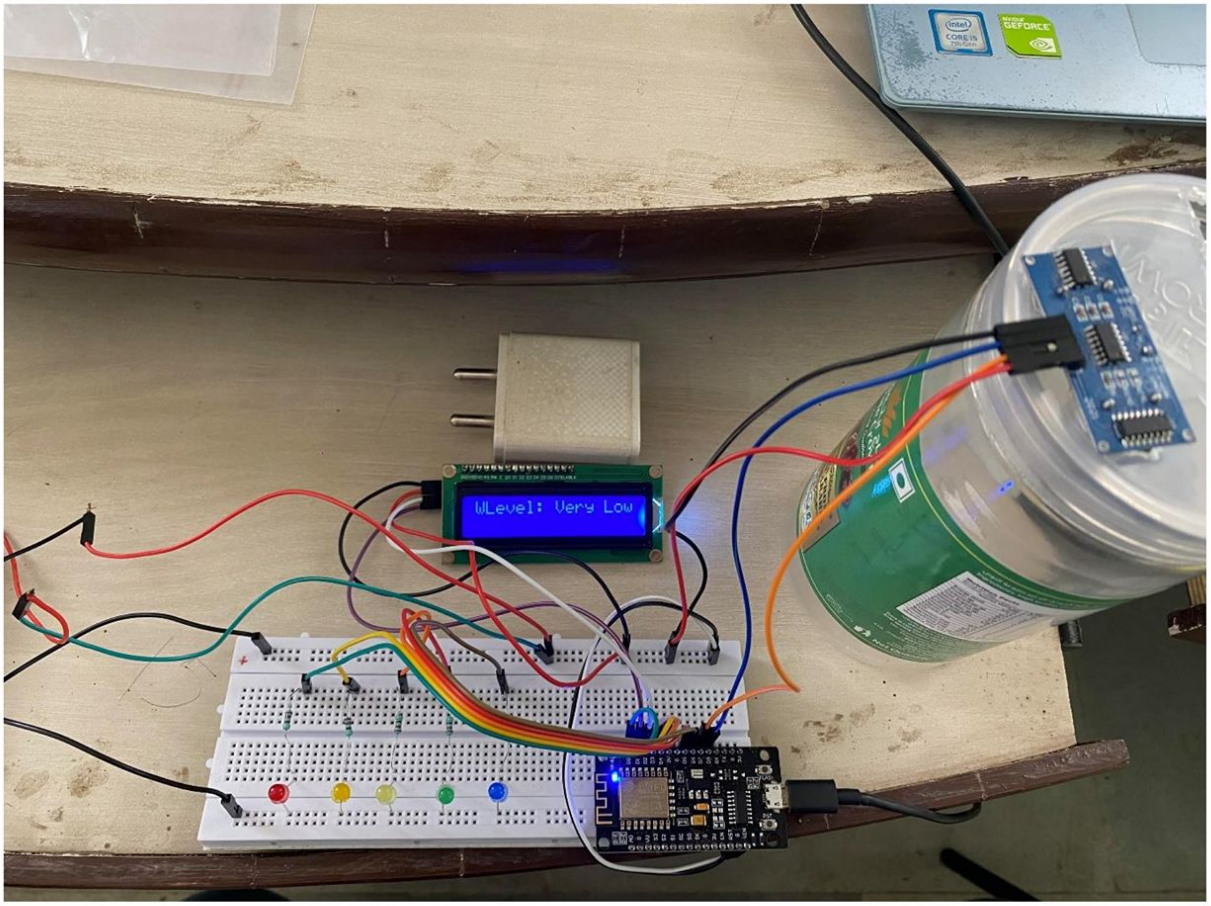

Figure no.5.19: Output

CHAPTER 7

CONCLUSION

The project enables us to observe the level of water from a distant location and helps us to track it and protect it from overflowing and there by enabling the user to ensure that no extra water gets used and there is no excess loss of water. We know that the major place where water gets wasted is industries and homes. So using this proposed system will help to minimize the water loss to a large extent. If the user knows about the water level in real-timehe/she has the power to maintain the water loss to an extent by maintaining the water at a sustainable height, i.e., between 30-70% of the height of the original tank to ensure no overflowing.

CHAPTER 8

REFERENCES

[1] L.A. Gama-Moreno, A.Corralejo, A.Ramirez-Molina [1], implemented a system to monitor water tanks. The implemented system consisted of instrumentation system, an application, which managed the water levels and a mobile user interface. The system was called as Monitoring Water Tanks (MWT). Ultrasonic sensor were put into use along with NODE MCU ESP8266 Microcontroller Board, which is connected to the application service.

[2] BezaNegashGetu, HussainA.Attia [2], have developed a system which initially tests the availability of water in the tank with the help of a level detector and then adjusts the state of the water pump according to the information collected through the level detector. The proposed system consists of water level sensor The proposed system eliminates manually controlling of water requirements in home and agricultural fields.

[3] PriyenP.Shah, Anjali A.Patil [3], introduced a project which uses Android application and IoT for monitoring of water in tanks. This project uses ESP 8266 as microcontroller. Maximum and minimum levels of water are obtained through ESP from blynk cloud. When the water level is between Maximum and minimum status can be controlled by the user. This project overcomes the short-comings of conventional tanks which can neither monitor nor control the water level in tank.

[4] Gama-Moreno, L. A., A. Corralejo, A. Ramirez-Molina, J. A. Torres-Rangel, C. Martinez-Hernandez, and M. A. Juarez. "A design of a water tanks monitoring system based on mobile devices." In 2016 International Conference on Mechatronics, Electronics and Automotive Engineering (ICMEAE), pp. 133-138. IEEE, 2016.

[5] Malche, Timothy, and Priti Maheshwary. "Internet of things (IoT) based water level monitoring system for smart village." In Proceedings of International Conference on Communication and Networks: ComNet 2016, pp. 305-312. Springer Singapore, 2017.