CONTENTS

CHAPTER	TITLE	PAGE
	Declaration	
	Certificate	
	Acknowledgement	
	Contents	
	List of Abbreviations	
	Abstract	
1	Introduction	4
2	Existing and Proposed system	5
3	System Requirements	8
4	System Design	32
5	Results	37
6	Advantages	39
7	Conclusion	40
8	References	41

LIST OF ABBREVIATIONS

Abbreviation	**Expansion**
ESP	Electronic Stimulated Programming
IDE	Integrated Development environment
Ssid	Service Set Identifier

ABSTRACT

In this project we will develop a fully Automated dumpster monitoring system on ESP8266 and an IoT app, the BLYNK. Here it works on wifi hence it can be used anywhere.

Basically, it is used to check whether the garbage is empty or full. We can know the status of our trashcans from anywhere in the world with the help of internet. It will be very useful and can be installed in the trash cans at public place as well as at home.

In this IOT project an ultrasonic sensor is used for detecting whether the trash can is filled with garbage or not. Here ultrasonic sensor is installed at the top of trash can and will measure the distance of garbage from the top of trash can and we can set a threshold value according to the size of trash can. If the distance will be less than this threshold value, means the garbage is full and we will get a message that garbage is full.

CHAPTER 1

INRODUCTION

In this project we will build an IoT project to check whether the trash can is empty or full using nodemcu, ultrasonic sensor, bread board power supply, blynk app. The ultrasonic sensor is connected to node mcu and 5v is taken from bread board power supply for the working of ultrasonic sensor. The ultrasonic sensor measures the distance by sending an echo, if there is any obstacle the echo signal comes back in this way the ultrasonic sensor works. Blynk app is the platform to check our output. If the trash can is full or empty or half filled we get an email about the status of our trash can. Blynk will be connected to node mcu and as it is iot based we can know the status of our trash can from any where in the world if we have internet access. The ultrasonic sensor will be on the top of the trash can and it goes on measuring the distance of the trash in the trash can. In the code we will give some threshold value with a specific condition if the distance reaches the threshold value we get the notification through mail. By this project we can maintain cleanliness.

CHAPTER 2

EXISTING SYSTEM

At present we don’t know whether the trash can is full or not without observing it, but it is harmful for a human to go near the trash because it will have many wastes including medical wastes and many more. so it is not safe to go always check the dumpsters. According to the existing system we don’t have any automatic equipment to know the status of trash cans. As we all know that swatch bharath mission is the trending aspect of india. It is mandatory to keep our premises clean. If our premises are clean then automatically our city will be clean and if cities are clean then country will be clean and we can see our country as the country we want.

PROPOSED SYSTEM

I have already discussed how we are facing problems using the present existing system. In order to avoid all these problems faced by mankind this project named “DUMPSTER MONITORING” is very useful because we can know the status of the trash cans from any where in the world as it is connected to internet. We can get up to date status of our trashcan as an email. So that we can clear the trash cans in time and we can keep it clean. This project is of low cost and the work of it is accurate. The ultrasonic sensor used in this project calculates the amount of the trash accurately. By this we can know the status of our trash can without observing it. This will be very useful project for the cleanliness. We all know that “swatch Bharath” mission is the trending aspect of our country this project will be very useful for the mission. Just by fixing this equipment to the trash cans we can able to get the status of our trash cans. To keep our premises clean we need trash cans to know the status of our trash cans I proposed a system. So with the help of this project we can keep our premises clean. If the premises are clean the our connected to internet. We can get up to date status of our trashcan as an email. So that we can clear the trash cans in time and we can keep it clean. This project is of low cost and the work of it is accurate. The ultrasonic sensor used in this project calculates the amount of the trash accurately. By this we can know the status of our trash can without observing it. This will be very useful project for the cleanliness. We all know that “swatch Bharath” mission is the trending aspect of our country this project will be very useful for the mission. Just by fixing this equipment to the trash cans we can able to get the status of our trash cans. To keep our premises clean we need trash cans to know the status of our trash cans I proposed a system. So with the help of this project we can keep our premises clean. If the premises are clean the our city will be clean and if the cities are clean then our country will be clean. Finally, the system I proposed takes a crucial part in the cleanliness of our country.

CHAPTER 3

PROJECT REQUIREMENTS

Software Requirements:

Operating system : WINDOWS 8/10

Software :Arduino IDE

Hardware Requirements:

NodeMCU

Bread board power supply

Bread board

Ultrasonic sensor

Jump wires

Platforms:

Blynk platform

ARDUINO IDE SOFTWARE

ARDUINO:

Arduino is an open-source platform used for building electronics projects. Arduino consists of both a physical programmable circuit board (often referred to as a microcontroller) and a piece of software, or IDE (Integrated Development Environment) that runs on your computer, used to write and upload computer code to the physical board.

The Arduino platform has become quite popular with people just starting out with electronics, and for good reason. Unlike most previous programmable circuit boards, the Arduino does not need a separate piece of hardware (called a programmer) in order to load new code onto the board – you can simply use a USB cable. Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program. Finally, Arduino provides a standard form factor that breaks out the functions of the micro-controller into a more accessible package. The Uno is one of the more popular boards in the Arduino family and a great choice for beginners. We'll talk about what's on it and what it can do later in the tutorial.

ARDUNIO IDE:

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java, c and based on Processing and other open-source software.

NODEMCU

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 dev 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.

History:

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. Two months later, the project expanded to include an open-hardware platform when developer Huang R committed the gerber file of an ESP8266 board, named devkit v0.9. Later that month, Tuan PM ported MQTT client library from Contiki to the ESP8266 SoC platform, and committed to NodeMCU project, then NodeMCU was able to support the MQTT IoT protocol, using Lua to access the MQTT broker. Another important update was made on 30 Jan 2015, when Devsaurus ported the u8glib to NodeMCU .

Developer	ESP8266 Opensource Community
Type	Single-board microcontroller
Operating system	XTOS
CPU	ESP8266
Memory	128kBytes
Storage	4MBytes
Power	USB
Website	www.nodemcu.com

Back view of NodeMCU

ESP8266 Arduino Core

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. For more information on all things ESP8266, check out the ESP8266 Community Forum on GitHub.

The Button

The Button is a Wi-Fi connected push button designed by Peter R Jennings The Button is designed for single-purpose, internet-enabled functions. When the button is pressed, a connection is made to a web server which will perform the desired task. Applications include a doorbell or panic button.

NodeUSB

NodeUSB is an open IoT platform about the size of a standard USB stick. It was designed to leverage NodeMCU (Lua) for easy programming and has the extra feature of USBcapability. It is ideal for Plug-n-Play solutions, allowing easy prototyping for developers.

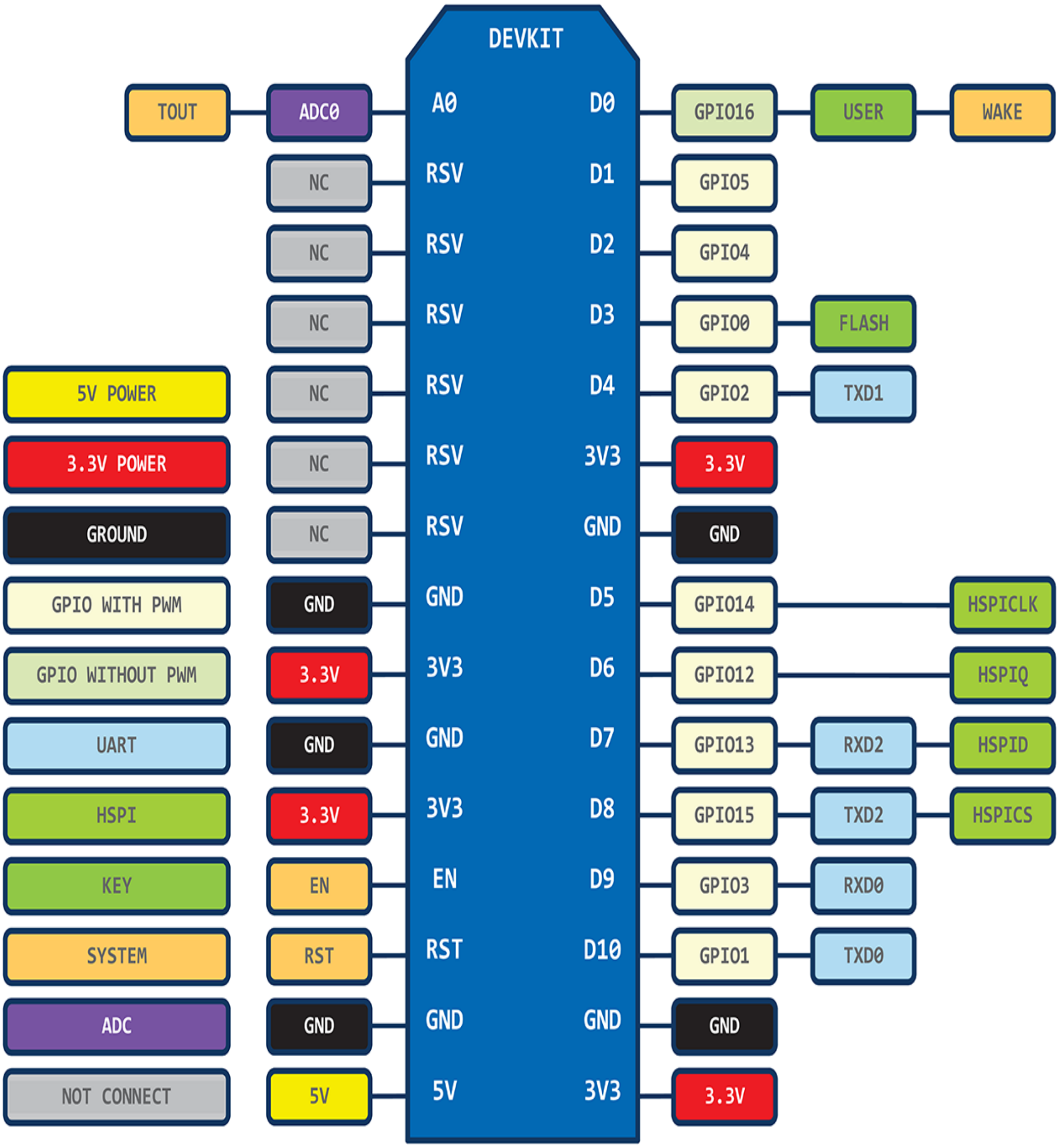
ijWatch

ijWatch is an open-hardware and open-source Wi-Fi smartwatch, using an OLED screen and running NodeMCU firmware·The author believes it may be the first smartwatch. (As in, the watch itself is fully functional without the pairing of another bluetooth device such as a smartphone.)

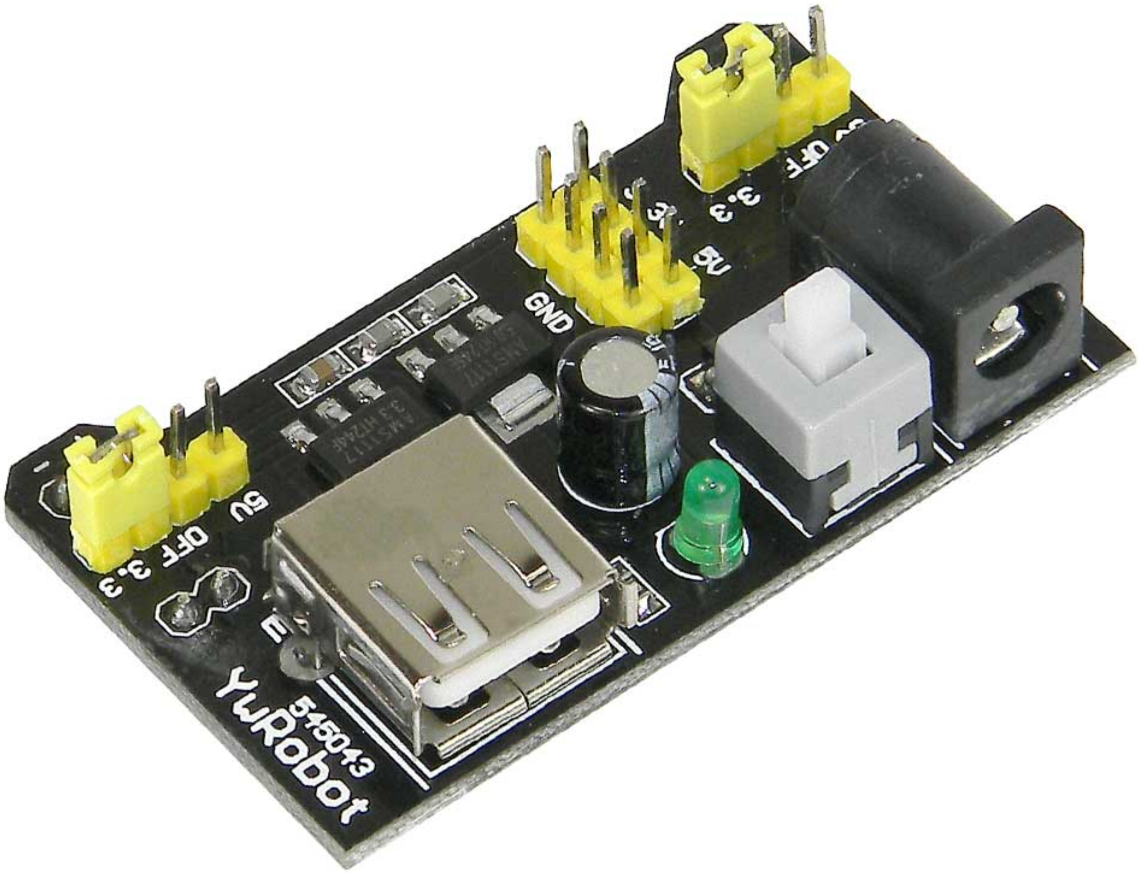
PINS OF NODEMCU:

NodeMCU provides access to the GPIO(General Purpose Input/Output) and for developing purposes below pin mapping table should be referenced.

IO index	ESP8266 pin	IO index	ESP8266 pin
0 [*]	GPIO16	7	GPIO13
1	GPIO5	8	GPIO15
2	GPIO4	9	GPIO3
3	GPIO0	10	GPIO1
4	GPIO2	11	GPIO9
5	GPIO14	12	GPIO10
6	GPIO12		

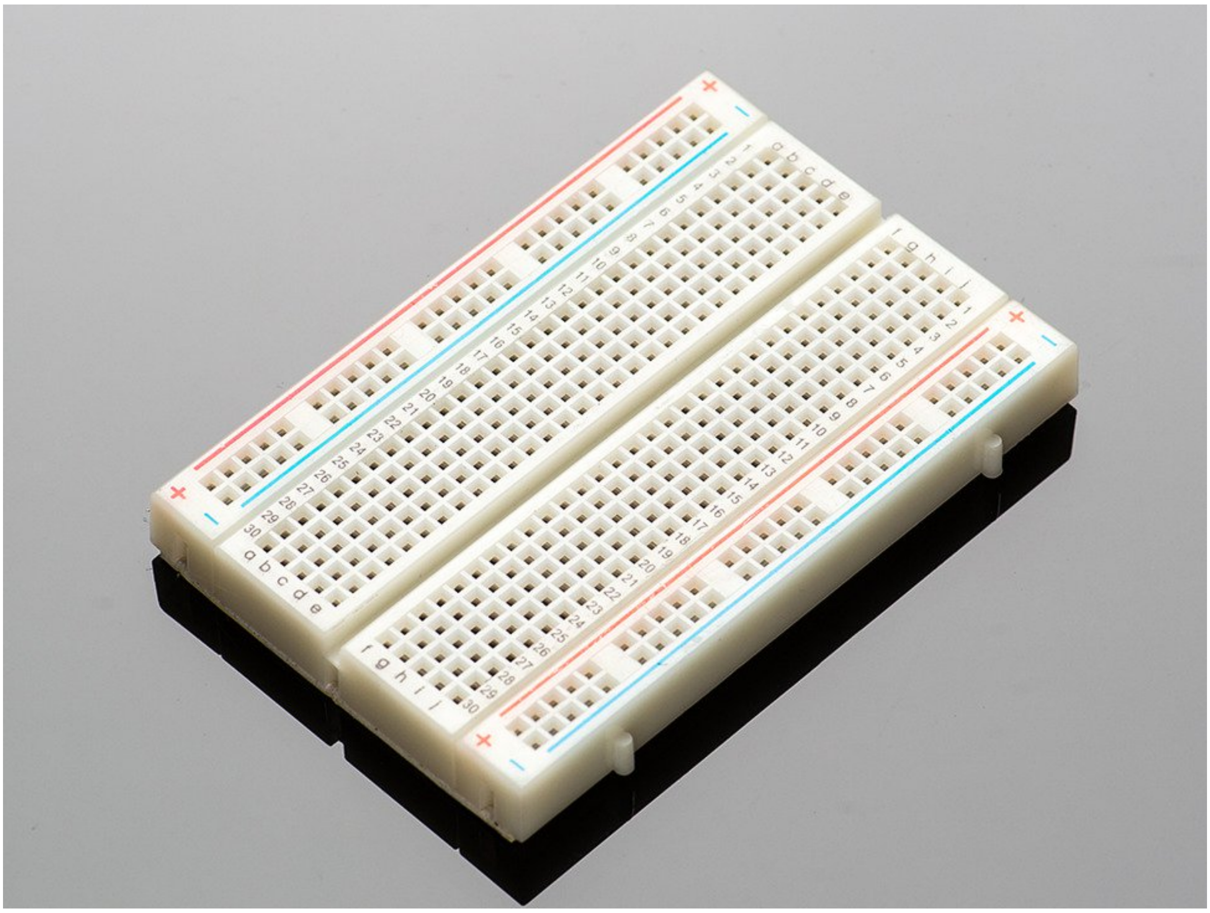

DEVKIT
TOUT
ADC0
A0
D0
GPIO16
USER
WAKE
NC
RSV
D1
GPIO5
NC
RSV
D2
GPIO4
NC
RSV
D3
GPIO0
FLASH
5V POWER
NC
RSV
D4
GPIO2
TXD1
3.3V POWER
NC
RSV
3V3
3.3V
GROUND
NC
RSV
GND
GND
GPIO WITH PWM
GND
GND
D5
GPIO14
HSPICLK
GPIO WITHOUT PWM
3.3V
3V3
D6
GPIO12
HSPIQ
UART
GND
GND
D7
GPIO13
RXD2
HSPID
HSPI
3.3V
3V3
D8
GPIO15
TXD2
HSPICS
KEY
EN
EN
D9
GPIO3
RXD0
SYSTEM
RST
RST
D10
GPIO1
TXD0
ADC
GND
GND
GND
GND
NOT CONNECT
5V
5V
3V3
3.3V

BREAD BOARD POWER SUPPLY



The bread board power supply is used to supply 5v to ultrasonic sensor as we get only 3.3v from the node mcu which is insufficient for the working of ultrasonic sensor.

BREAD BOARD


+
-
a b c d e
f g h i j
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30

A breadboard is a construction base for prototyping of electronics. Originally it was literally a bread board, a polished piece of wood used for slicing bread. In the 1970s the solderless breadboard (a.k.a. plugboard, a terminal array board) became available and nowadays the term "breadboard" is commonly used to refer to these.

Because the solderless breadboard does not require soldering, it is reusable. This makes it easy to use for creating temporary prototypes and experimenting with circuit design. For this reason, solderless breadboards are also extremely popular with students and in technological education. Older breadboard types did not have this property. A stripboard (Veroboard) and similar prototyping printed circuit boards, which are used to build semi-permanent soldered prototypes or one-offs, cannot easily be reused. A variety of electronic systems may be prototyped by using breadboards, from small analog and digital circuits to complete central processing units cpus.

Evolution

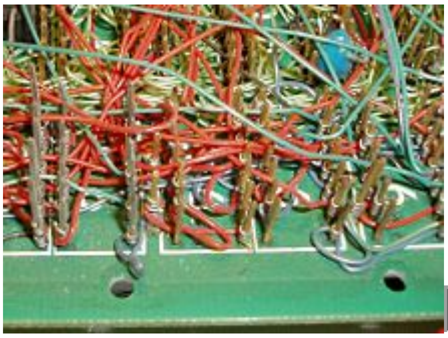
wikiUrl

This 1920s TRF radio manufactured by Signal was constructed on a wooden breadboard.

In the early days of radio, amateurs nailed bare copper wires or terminal strips to a wooden board (often literally a board to slice bread on) and soldered electronic components to them. Sometimes a paper schematic diagram was first glued to the board as a guide to placing terminals, then components and wires were installed over their symbols on the schematic. Using thumbtacks or small nails as mounting posts was also common.

Breadboards have evolved over time, with the term now being used for all kinds of prototype electronic devices. For example, US Patent 3,145,483, was filed in 1961 and describes a wooden plate breadboard with mounted springs and other facilities. US Patent 3,496,419, was filed in 1967 and refers to a particular printed circuit board layout as a Printed Circuit Breadboard. Both examples refer to and describe other types of breadboards as prior art.

The breadboard most commonly used today is usually made of white plastic and is a pluggable (solderless) breadboard. It was designed by Ronald J. Portugal in 1971.



wikiUrl

Alternative methods to create prototypes are point-to-point construction (reminiscent of the original wooden breadboards), wire wrap, wiring pencil, and boards like the stripboard. Complicated systems, such as modern computers comprising millions of transistors, diodes, and resistors, do not lend themselves to prototyping using breadboards, as their complex designs can be difficult to lay out and debug on a breadboard.

Modern circuit designs are generally developed using a schematic capture and simulation system, and tested in software simulation before the first prototype circuits are built on a printed circuit board. Integrated circuit designs are a more extreme version of the same process: since producing prototype silicon is costly, extensive software simulations are performed before fabricating the first prototypes. However, prototyping techniques are still used for some applications such as RF circuits, or where software models of components are inexact or incomplete.

You could also use a square grid of pairs of holes where one hole per pair connects to its row and the other connects to its column. This same shape can be in a circle with rows and columns each spiraling opposite clockwise/counterclockwise.

Solderless breadboard

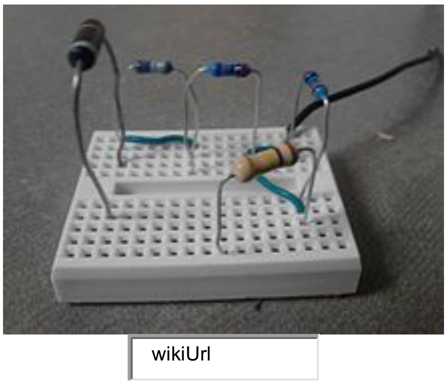
Typical specifications

A modern solderless breadboard socket consists of a perforated block of plastic with numerous tin plated phosphor bronze or nickel silver alloy spring clips under the perforations. The clips are often called tie points or contact points. The number of tie points is often given in the specification of the breadboard.

The spacing between the clips (lead pitch) is typically 0.1 in (2.54 mm). Integrated circuits (ICs) in dual in-line packages (DIPs) can be inserted to straddle the centerline of the block. Interconnecting wires and the leads of discrete components (such as capacitors, resistors, and inductors) can be inserted into the remaining free holes to complete the circuit. Where ICs are not used, discrete components and connecting wires may use any of the holes. Typically the spring clips are rated for 1 ampere at 5 volts and 0.333 amperes at 15 volts (5 watts). The edge of the board has male and female dovetail notches so boards can be clipped together to form a large breadboard.

Bus and terminal strips

Solderless breadboards are available from several different manufacturers, but most share a similar layout. The layout of a typical solderless breadboard is made up from two types of areas, called strips. Strips consist of interconnected electrical terminals.



wikiUrl

Breadboard consisting of only terminal strips but no bus strips

Terminal strips

The main areas, to hold most of the electronic components.

In the middle of a terminal strip of a breadboard, one typically finds a notch running in parallel to the long side. The notch is to mark the centerline of the terminal strip and provides limited airflow (cooling) to DIP ICs straddling the centerline. The clips on the right and left of the notch are each connected in a radial way; typically five clips (i.e., beneath five holes) in a row on each side of the notch are electrically connected. The five rows on the left of the notch are often marked as A, B, C, D, and E, while the ones on the right are marked F, G, H, I and J. When a "skinny" dual in-line pin package (DIP) integrated circuit (such as a typical DIP-14 or DIP-16, which have a 0.3-inch (7.6 mm) separation between the pin rows) is plugged into a breadboard, the pins of one side of the chip are supposed to go into row E while the pins of the other side go into row F on the other side of the notch.

Bus strips

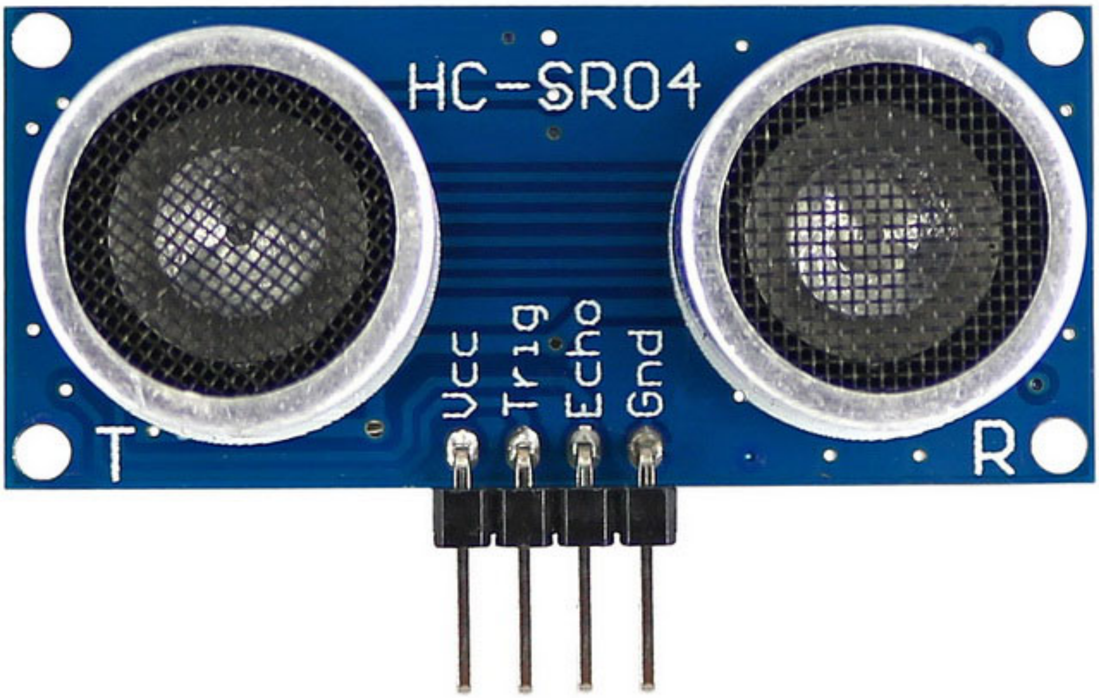
To provide power to the electronic components.

A bus strip usually contains two rows: one for ground and one for a supply voltage. However, some breadboards only provide a single-row power distributions bus strip on each long side. Typically the row intended for a supply voltage is marked in red, while the row for ground is marked in blue or black. Some manufacturers connect all terminals in a column. Others just connect groups of, for example, 25 consecutive terminals in a column. The latter design provides a circuit designer with some more control over crosstalk (inductively coupled noise) on the power supply bus. Often the groups in a bus strip are indicated by gaps in the color marking.

Bus strips typically run down one or both sides of a terminal strip or between terminal strips. On large breadboards additional bus strips can often be found on the top and bottom of terminal strips.

Note there are two different common alignments for the power bus strips. On small boards, with about 30 rows, the holes for the power bus are often aligned between the signal holes. On larger boards, about 63 rows, the power bus strip holes are often in alignment with the signal holes. This makes some accessories designed for one board type incompatible with the other. For example, some Raspberry Pi GPIO to breadboard adapters use offset aligned power pins, making them not fit breadboards with aligned power bus rows. There are no official standards, so the users need to pay extra attention to the compatibility between a specific model of breadboard and a specific accessory. Vendors of accessories and breadboards are not always clear in their specifications of which alignment they use. Seeing a close up photograph of the pin/hole arrangement can help determine compatibility.

ULTRASONIC SENSOR



An Ultrasonic sensor is a device that can measure the distance to an object by using sound waves. It measures distance by sending out a sound wave at a specific frequency and listening for that sound wave to bounce back. By recording the elapsed time between the sound wave being generated and the sound wave bouncing back, it is possible to calculate the distance between the sonar sensor and the object.

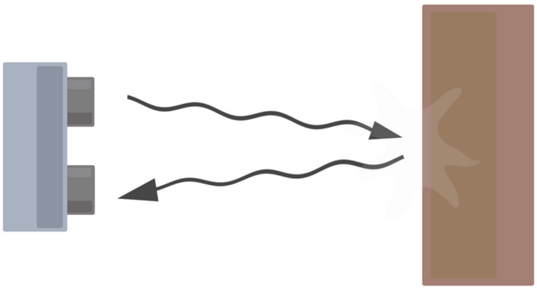


Diagram of the basic ultrasonic sensor operation

$$distance = \frac{speed\ of\ sound\ \times time\ taken}{2}$$

Since it is known that sound travels through air at about 344 m/s (1129 ft/s), you can take the time for the sound wave to return and multiply it by 344 meters (or 1129 feet) to find the total round-trip distance of the sound wave. Round-trip means that the sound wave traveled 2 times the distance to the object before it was detected by the sensor; it includes the 'trip' from the sonar sensor to the object AND the 'trip' from the object to the Ultrasonic sensor (after the sound wave bounced off the object). To find the distance to the object, simply divide the round-trip distance in half.

NOTE: The accuracy of Ultrasonic sensor can be affected by the temperature and humidity of the air it is being used in. However, for these tutorials and almost any project you will be using these sensors in, this change in accuracy will be negligible.

It is important to understand that some objects might not be detected by ultrasonic sensors. This is because some objects are shaped or positioned in such a way that the sound wave bounces off the object, but are deflected away from the Ultrasonic sensor. It is also possible for the object to be too small to reflect enough of the sound wave back to the sensor to be detected. Other objects can absorb the sound wave all together (cloth, carpeting, etc), which means that there is no way for the sensor to detect them accurately.

About Ultrasonics

Ultrasonic sensors are used around the world, indoors and outdoors in the harshest conditions, for a variety of applications. Our ultrasonic sensors, made with piezoelectric crystals, use high frequency sound waves to resonate a desired frequency and convert electric energy into acoustic energy, and vice versa. Sound waves are transmitted to and reflected from the target back to the transducer. Targets can have any reflective form, even round. Certain variables, such as target surface angle, changes in temperature and humidity, and reflective surface roughness, can affect the operation of the sensors.

There are two types of ultrasonic sensors

Proximity Detection: An object passing within the preset range will be detected and generate an output signal. The detect point is independent of target size, material or reflectivity.

Ranging Measurement: Precise distance(s) of an object moving to and from the sensor are measured via time intervals between transmitted and reflected bursts of ultrasonic sound. Distance change is continuously calculated and outputted.

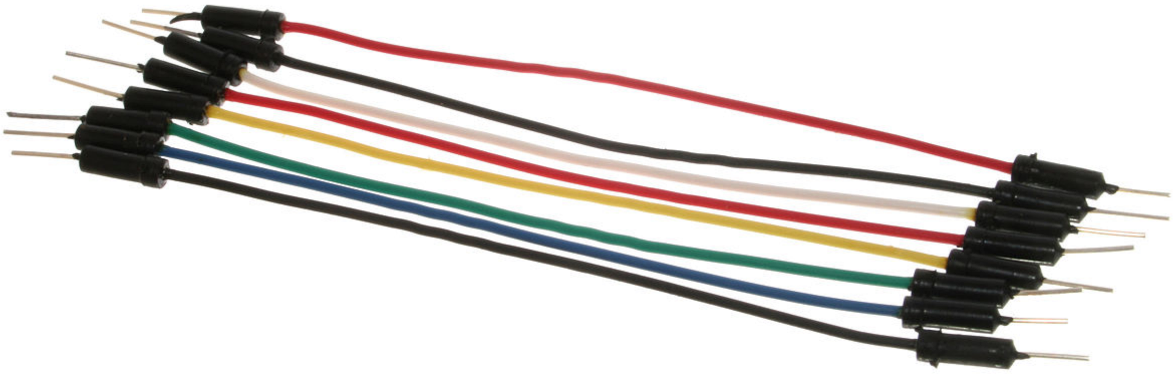
Ultrasonic Sensing Applications

Migatron has been using advanced technology to solve difficult sensing and control problems for nearly 40 years across a broad range of of industries. With Ultrasonic Sensing’s unique advantages over conventional sensors and the rapidly increasing range of applications, ultrasonic sensors are becoming widely accepted as an industry standard across the board.

JUMP WIRES

A **jump wire** (also known as jumper, jumper wire, jumper cable, DuPont wire, or DuPont cable – named for one manufacturer of them) is an electrical wire or group of them in a cable with a connector or pin at each end (or sometimes without them – simply "tinned"), which is normally used to interconnect the components of a breadboard or other prototype or test circuit, internally or with other equipment or components, without soldering.

Individual jump wires 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.



JUMPER WIRES:

There are different types of jumper wires. Some have the same type of electrical connector at both ends, while others have different connectors. Some common connectors are:

- Solid tips – are used to connect on/with a breadboard or female header connector. The arrangement of the elements and ease of insertion on a breadboard allows increasing the mounting density of both components and jump wires without fear of short-circuits. The jump wires vary in size and colour to distinguish the different working signals.
- Crocodile clips – are used, among other applications, to temporarily bridge sensors, buttons and other elements of prototypes with components or equipment that have arbitrary connectors, wires, screw terminals, etc.
- Banana connectors – are commonly used on test equipment for DC and low-frequency AC signals.
- Registered jack (RJnn) – are commonly used in telephone (RJ11) and computer networking (RJ45).
- RCA connectors – are often used for audio, low-resolution composite video signals, or other low-frequency applications requiring a shielded cable.

BLYNK

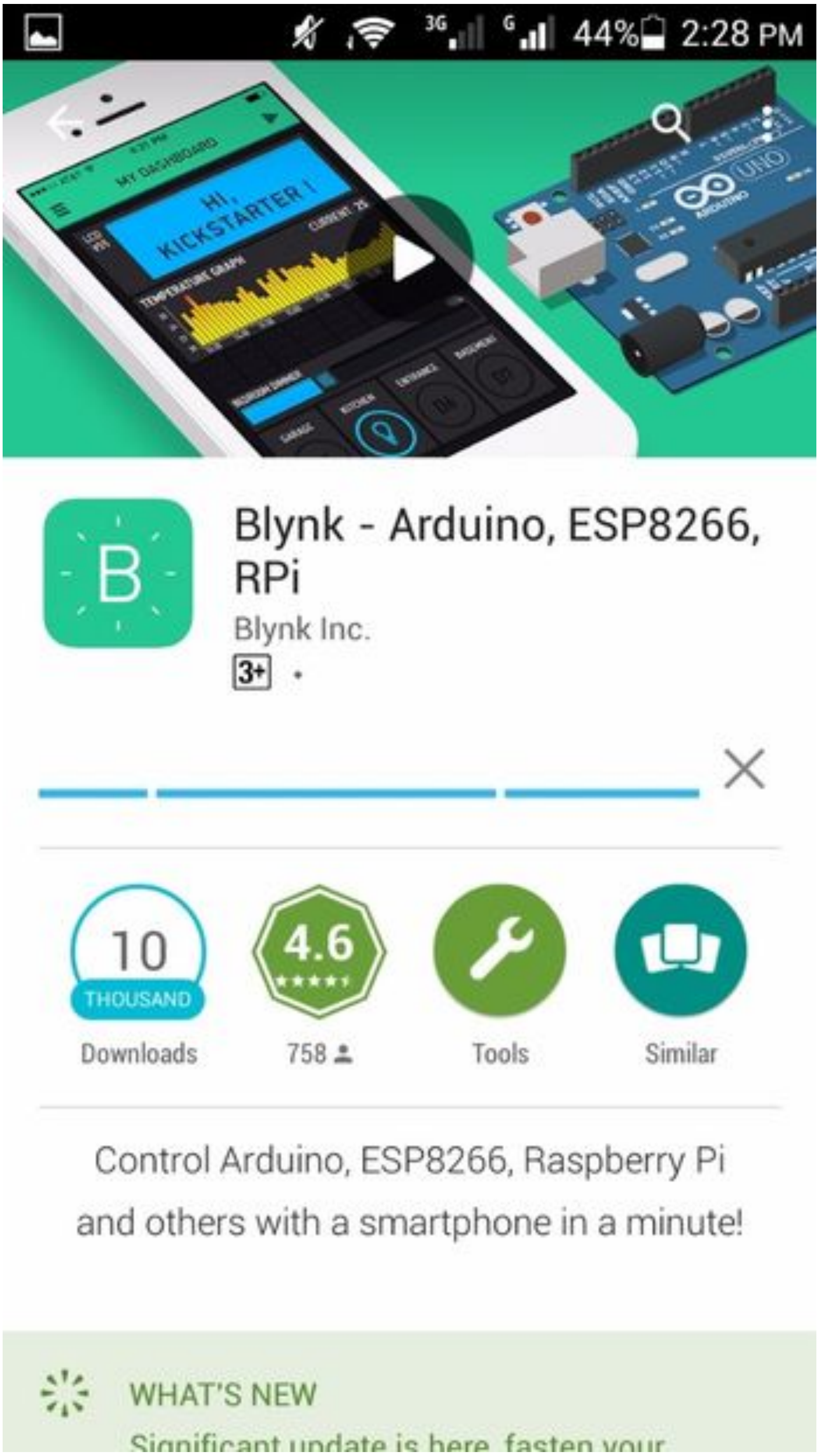
Blynk is a Platform with iOS and Android apps to control Arduino, Raspberry Pi and the likes over the Internet.

It's a digital dashboard where you can build a graphic interface for your project by simply dragging and dropping widgets.

It's really simple to set everything up and you'll start tinkering in less than 5 mins.

Blynk is not tied to some specific board or shield. Instead, it's supporting hardware of your choice. Whether your Arduino or Raspberry Pi is linked to the Internet over Wi-Fi, Ethernet or this new ESP8266 chip, Blynk will get you online and ready for the Internet Of Your Things.

INSTALLING BLYNK



44% 2:28 PM

Blynk - Arduino, ESP8266, RPi

Blynk Inc.

3+

Control Arduino, ESP8266, Raspberry Pi and others with a smartphone in a minute!

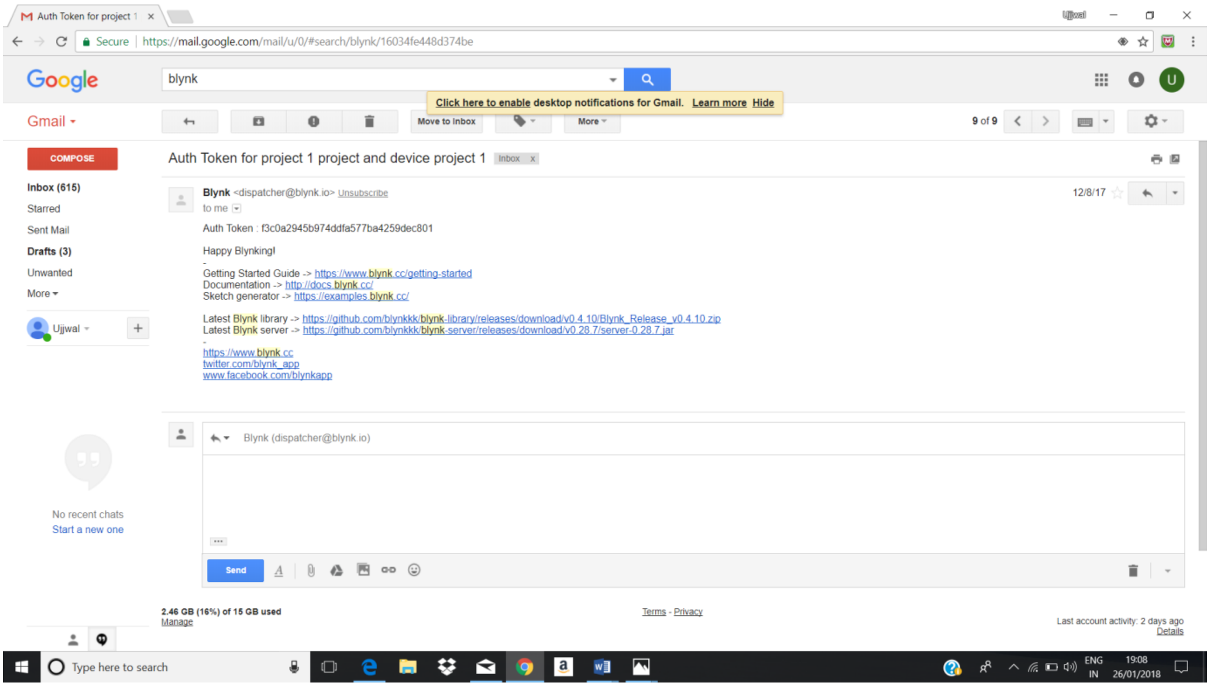
WHAT'S NEW

Significant update is here, fasten your

We can download it from google play store and it appears as shown above

CREATING ACCOUNT

First we have to create an account in blynk then an authentication code will be sent to our mail.



Auth Token for project 1 project and device project 1 Inbox

Blynk <dispatcher@blynk.io> Unsubscribe
to me
12/8/17

Auth Token : f3c0a2945b974ddfa577ba4259dec801

Happy Blynking!
-
Getting Started Guide -> https://www.blynk.cc/getting-started
Documentation -> http://docs.blynk.cc/
Sketch generator -> https://examples.blynk.cc/

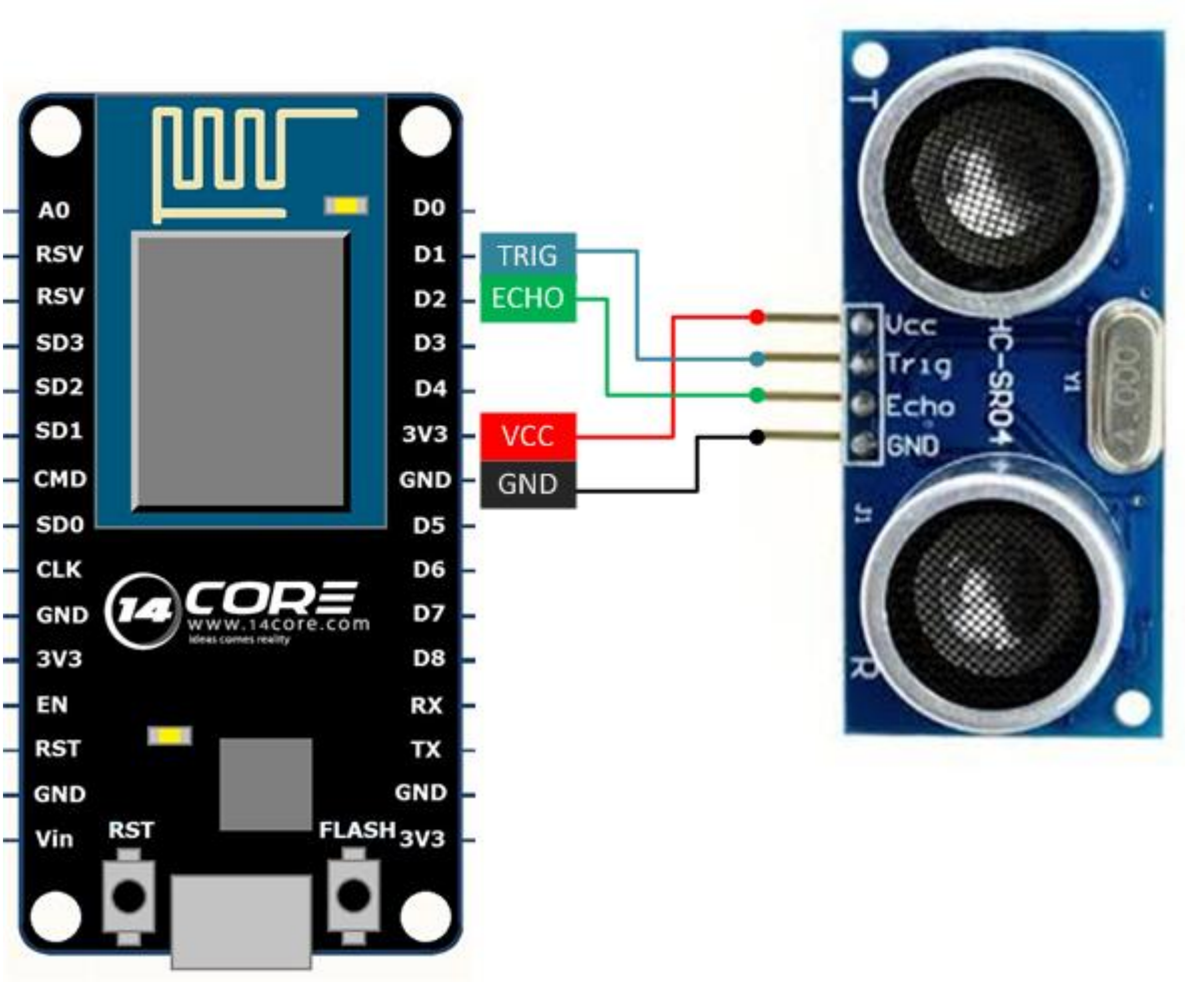
Latest Blynk library -> https://github.com/blynkkk/blynk-library/releases/download/v0.4.10/Blynk_Release_v0.4.10.zip
Latest Blynk server -> https://github.com/blynkkk/blynk-server/releases/download/v0.28.7/server-0.28.7.jar
-
https://www.blynk.cc
twitter.com/blynk_app
www.facebook.com/blynkapp

Blynk (dispatcher@blynk.io)

CHAPTER 4

PROJECT DESIGN

First we connect the node mcu board to our laptop with usb cable the node mcu board will be connected to wifi for the internet access then the ultrasonic sensor is connected to node mcu from the bread board power supply. For the working of ultrasonic sensor 5v power supply is needed so it is connected to bread board power supply for 5v power supply. These are connected with the help of jumper wires we select the type of jumper wire accordingly.



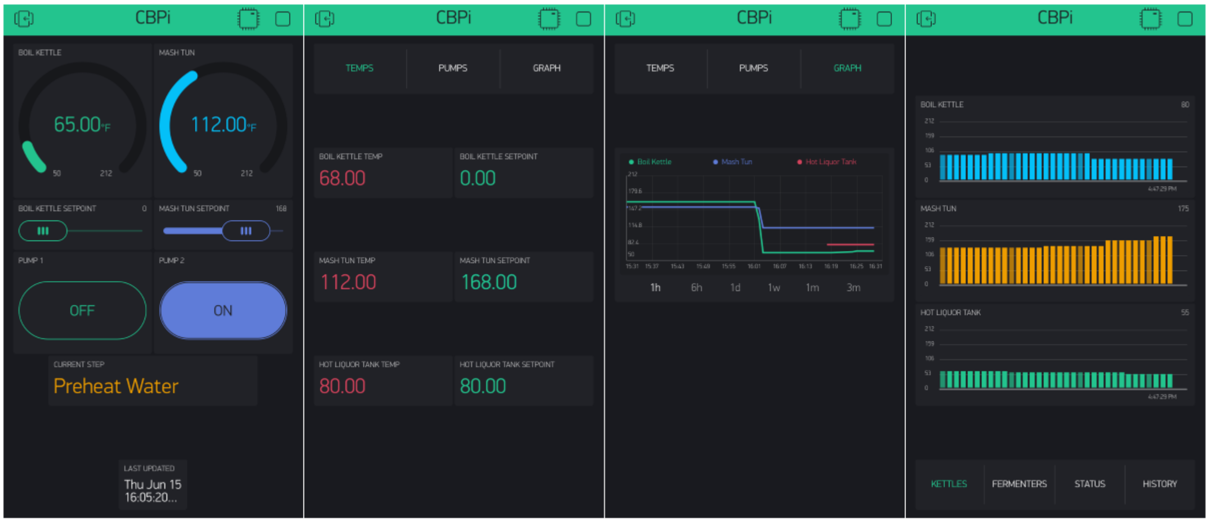
In this way the ultrasonic sensor is connected to node mcu.

CODE

For the compilation and upload of code properly we need to install some libraries and run the code.

```
#define TRIGGER 14

#define ECHO    12

// NodeMCU Pin D1 > TRIGGER | Pin D2 > ECHO

#define BLYNK_PRINT Serial    // Comment this out to disable prints and save
space

#include <ESP8266WiFi.h>

#include <BlynkSimpleEsp8266.h>

// You should get Auth Token in the Blynk App.

// Go to the Project Settings (nut icon).

char auth[] = "e6556dd0d0c1459ca08cf786f772cf97";

// Your WiFi credentials.

// Set password to "" for open networks.

char ssid[] = "stare";

char pass[] = "9652323687";

void setup() {

  Serial.begin (9600);

  Blynk.begin(auth, ssid, pass);

  pinMode(TRIGGER, OUTPUT);

  pinMode(ECHO, INPUT);

  pinMode(BUILTIN_LED, OUTPUT);

}

void loop() {
  long duration, distance;
  digitalWrite(TRIGGER, LOW);
  delayMicroseconds(2);


  digitalWrite(TRIGGER, HIGH);
  delayMicroseconds(10);
   digitalWrite(TRIGGER, LOW);
  delayMicroseconds(2);
  duration = pulseIn(ECHO, HIGH);
Serial.println(duration);
  distance = (duration/2)/29.1;
   if (distance <= 150) {
    Blynk.virtualWrite(V0, 255);
}
  else {
    Blynk.virtualWrite(V0, 0);
  }
  if (distance <= 100) {
    Blynk.virtualWrite(V1, 255);
}
  else {
    Blynk.virtualWrite(V1, 0);
  }
  if (distance <= 80) {
    Blynk.virtualWrite(V2, 255);
}
  else {
    Blynk.virtualWrite(V2, 0);
  }


   if (distance <= 40) {
    Blynk.virtualWrite(V3, 255);
}
  else {
    Blynk.virtualWrite(V3, 0);
  }
 if (distance <= 20) {
    Blynk.virtualWrite(V4, 255);
}
  else {
    Blynk.virtualWrite(V4, 0);
  }
Serial.print(distance);
  Serial.println("Centimeter:");
  Blynk.virtualWrite(V5, distance);
  delay(200);
  Blynk.run();
}
```

After the code is compiled then we have to upload it to the board after uploading we can get the distance in the blynk app



BOIL KETTLE 65.00°F

MASH TUN 112.00°F

BOIL KETTLE SETPOINT 0

MASH TUN SETPOINT 168

PUMP 1 OFF

PUMP 2 ON

CURRENT STEP
Preheat Water

LAST UPDATED
Thu Jun 15
16:05:20...

BOIL KETTLE TEMP 68.00

BOIL KETTLE SETPOINT 0.00

MASH TUN TEMP 112.00

MASH TUN SETPOINT 168.00

HOT LIQUOR TANK TEMP 80.00

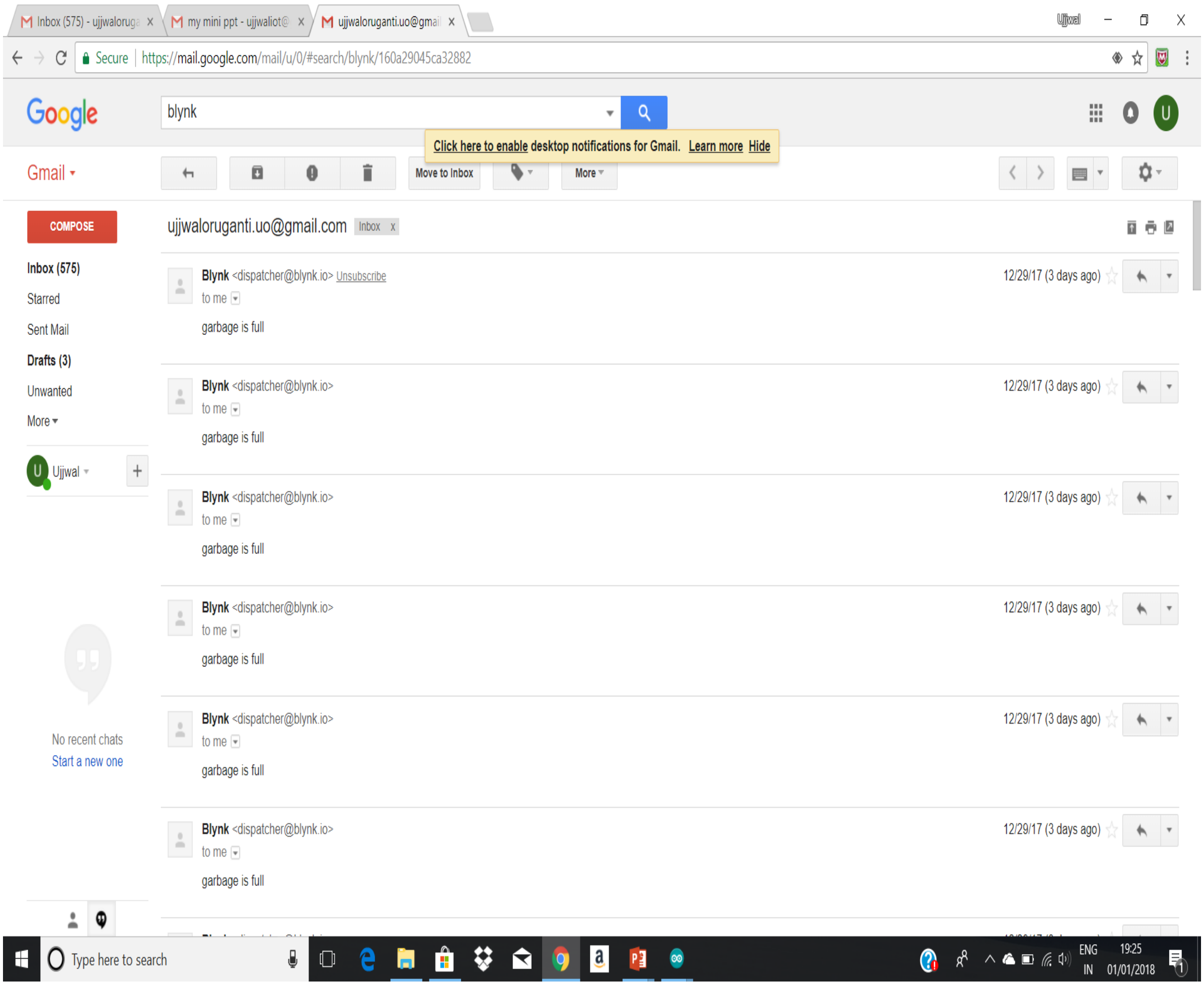
HOT LIQUOR TANK SETPOINT 80.00

We can see the distance of the trash from any of these formats with the help of blynk app.

CHAPTER-5

RESULTS

In this way we get the mail giving the condition of our trash can.



ujjwaloruganti.uo@gmail.com Inbox

Blynk <dispatcher@blynk.io> Unsubscribe 12/29/17 (3 days ago)
to me
garbage is full

Blynk <dispatcher@blynk.io> 12/29/17 (3 days ago)
to me
garbage is full

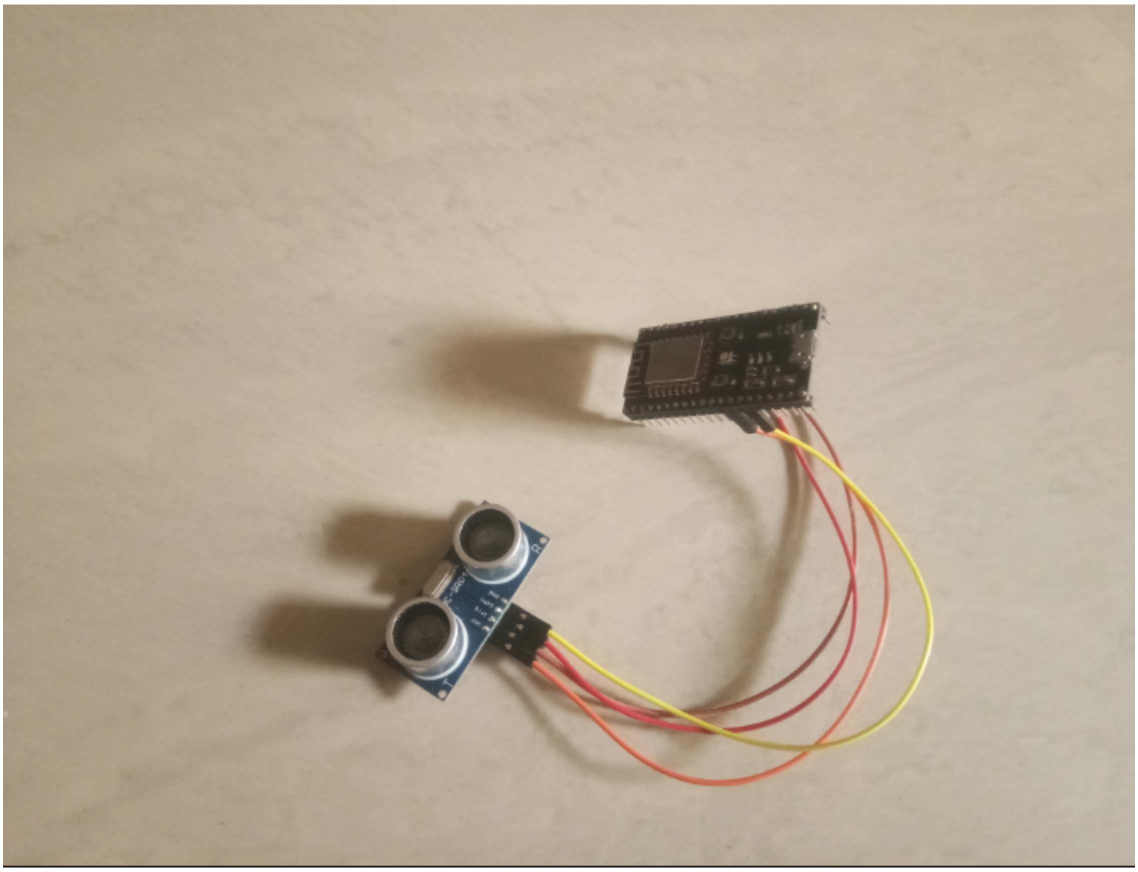
Blynk <dispatcher@blynk.io> 12/29/17 (3 days ago)
to me
garbage is full

Blynk <dispatcher@blynk.io> 12/29/17 (3 days ago)
to me
garbage is full

Blynk <dispatcher@blynk.io> 12/29/17 (3 days ago)
to me
garbage is full

Blynk <dispatcher@blynk.io> 12/29/17 (3 days ago)
to me
garbage is full

AFTER CONNECTIONS:



CHAPTER-6

ADVANTAGES

- It is time saving.
- It is very useful for keeping our surroundings clean.
- No need of constant monitoring.
- Reduced work.
- Can be used in public dustbins.
- No wastage of time.
- Eco-friendly environment.

CHAPTER-7

CONCLUSION

Thus dumpster monitoring system is made easy by this project because we are reducing the mankind labour and we are reducing all the needs and use of only some money in using this project and hence we are saving money in implementing this project and we are also using modern technology in day to day life and hence this IoT projects we will bring a revolution in our life.

CHAPTER-8

REFERENCES

TEXT BOOKS:

1. Systems,Espressif.”Espressif Systems”. EspressifWikiDevi. Retrieved 3 june 2017.
2. Jump up Brain Benchoff. “A DEV BOARD FOR THE ESP LUA INTERPRETER”. Hackday. Retrieved 2 April 2015.

WEB PORTAL:

1) https://www.youtube.com/playlist?list
2) Http://easycoding.tn/index.php/resource