IOT EMBEDDED WATER TANK MONITORING SYSTEM

09.06.2020

Overview

Have you Ever got into situation with low or without water for getting ready to daily work as you forgot to fill your water tank?.This situation may not be a big deal if you are staying at home but what if this happens during your morning busy week days..? To overcome this we have designed a system which can automate you motor and keeps a eye on the water level.This allows you to check amount of water left in your tank and even turn off system if you don't want to fill your tank while you have been outstation . hence,The proposed system eliminates manual monitoring and controlling for your home .The system also achieves proper water management and enhances productivity from automation.

Components:

Hardware:

- Nodemcu development board
- Ultrasonic sensor HC-SR04
- 5v Relay
- BreadBoard(optional)
- 6v Battery pack(4x AA battery each 1.5v)
- Jumper cables(optional)
- Micro USB cable(Older mobile charger cable)

Software/Services:

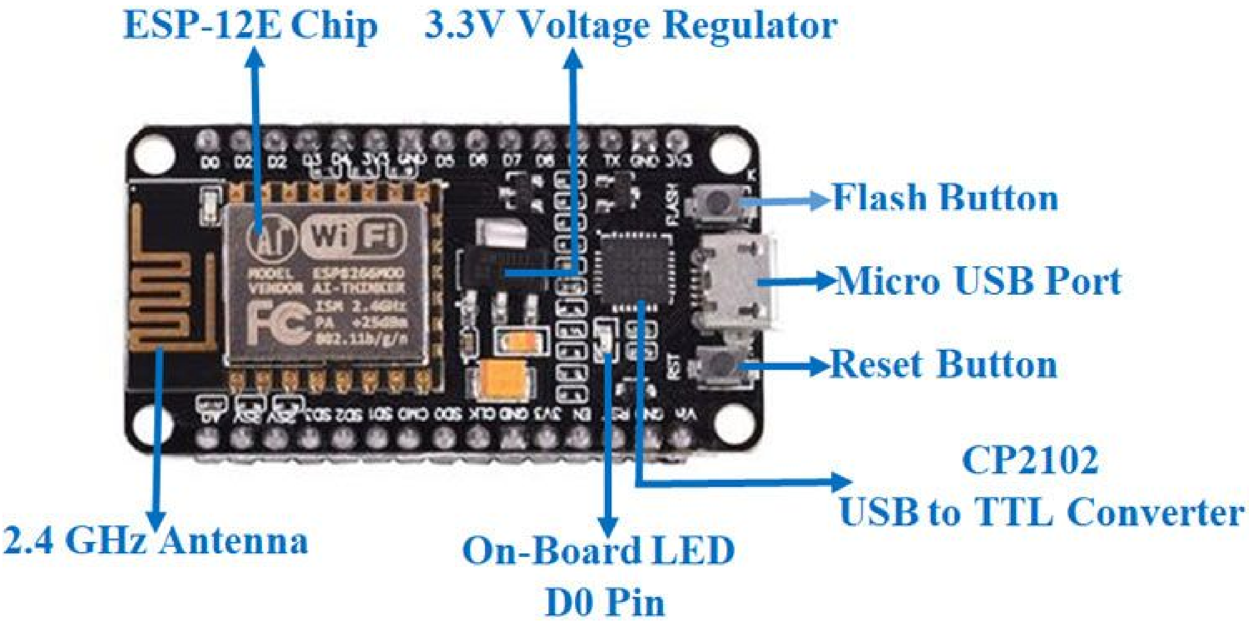
- Arduino IDE (Offline/Online version)
- Blynk (Android/IOS)
- If This Then That(IFTTT)
- Google assistant

Specifications about Components:

Nodemcu:

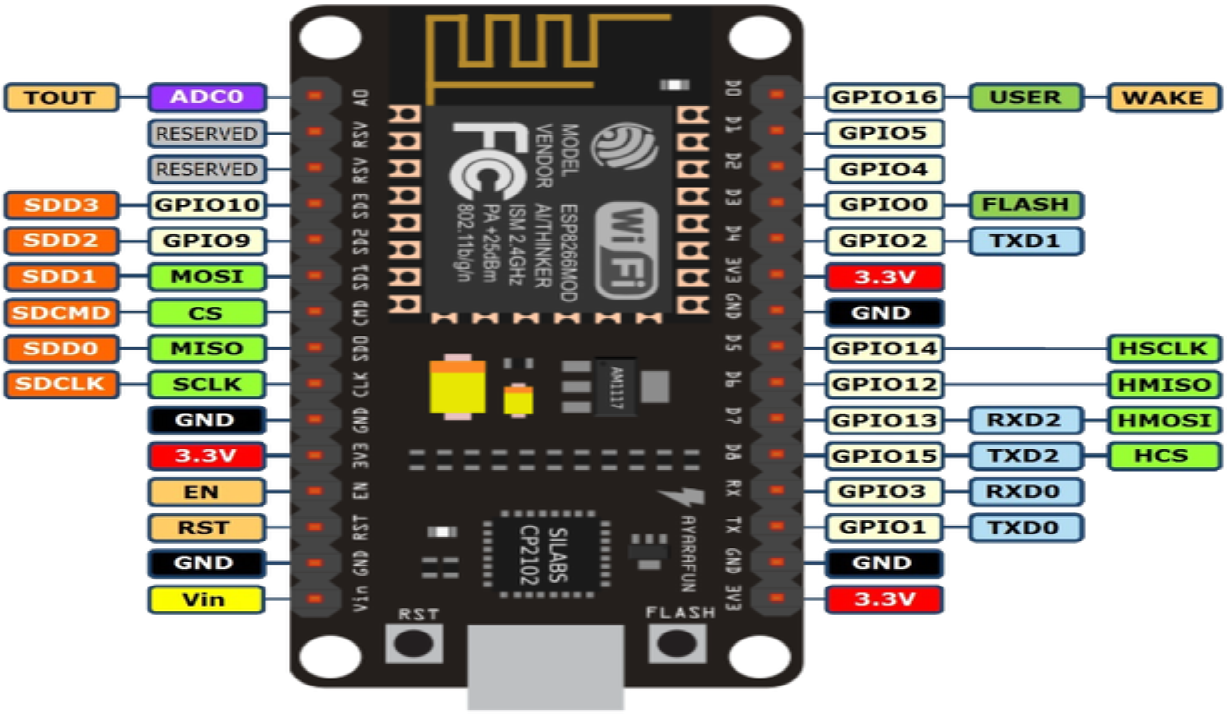
The **NodeMCU ESP8266 board** comes with the ESP-12E module(ESP8266 chip and Tensilica Xtensa 32-bit LX106 RISC microprocessor). This microprocessor supports operates at 80MHz to 160 MHz adjustable clock frequency. NodeMCU has 128 KB RAM and 4MB of Flash memory to store data and programs. Its high processing power with in-built

Wi-Fi/Bluetooth and Deep Sleep Operating features make it ideal for IoT projects.NodeMCU can be powered using Micro USB jack and VIN pin (External Supply Pin). It supports UART, SPI, and I2C interface. Nodemcu can be programmed by using Arduino IDE.



fig(1)Nodemcu

Nodemcu has 16 digital pins(GPIO) and one Analog pin. Operating Voltage of this board is 3.3v. But can receive Input voltage from 5 to 12v(via 5v usb or vin pin).

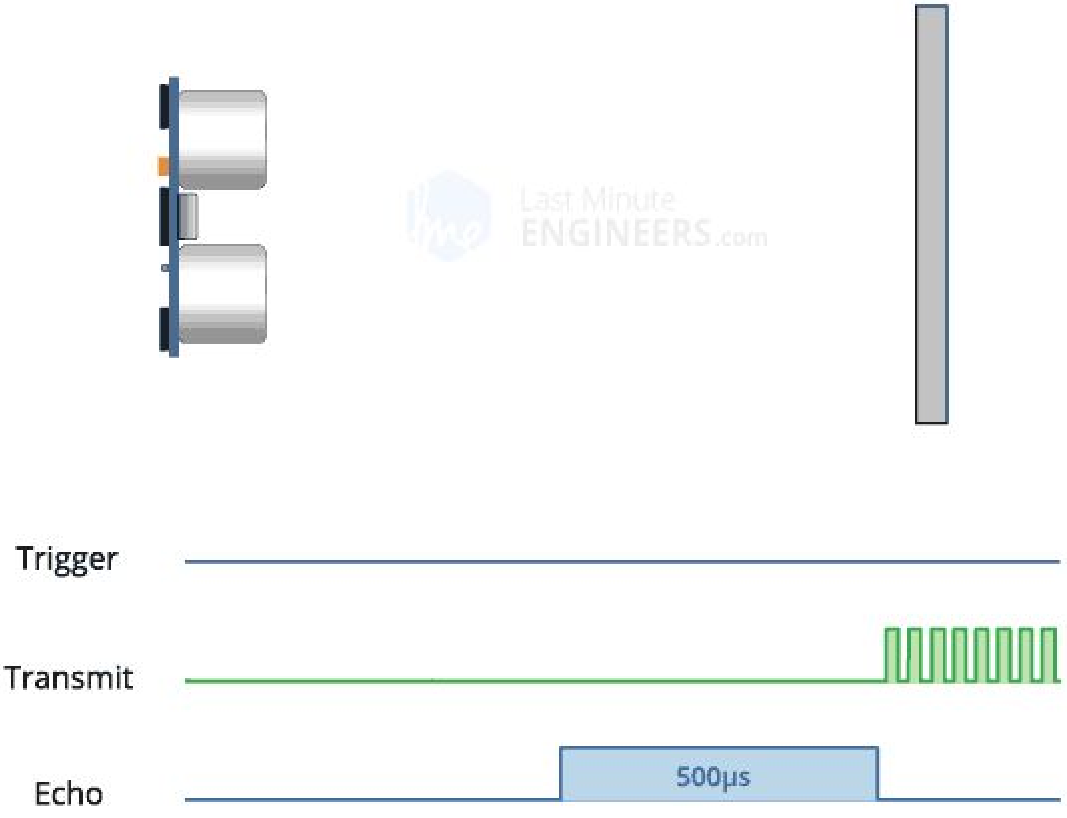


Fig(2).Nodemcu pin mapping

Note: You have use corresponding arduino pin mapping while writing a sketch.

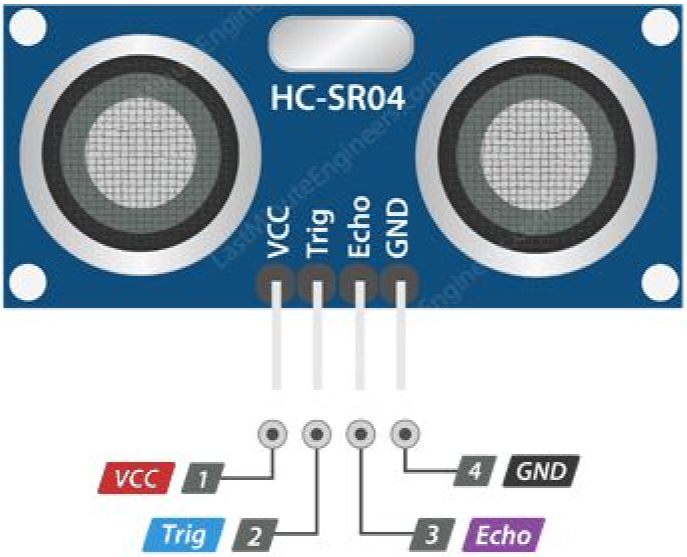
UltraSonic Sensor HC-SR04:

The HC-SR04 Ultrasonic distance sensor consists of two ultrasonic transducers. The one acts as a transmitter which converts electrical signals into 40 KHz ultrasonic sound pulses. The receiver listens for the transmitted pulses. If it receives them it produces an output pulse whose width can be used to determine the distance the pulse travelled.



Fig(3)HC-SR04

The sensor is small, easy to use in any robotics project and offers excellent non-contact range detection between 2 cm to 400 cm with an accuracy of 3mm. Since it operates on 5 volts, it can be hooked directly to an Arduino or any other 5V logic



fig(4)HC-SR04 pin mapping

Vcc: 5v power supply

Trig: It is an input pin used to trigger the ultrasonic sound pulses.

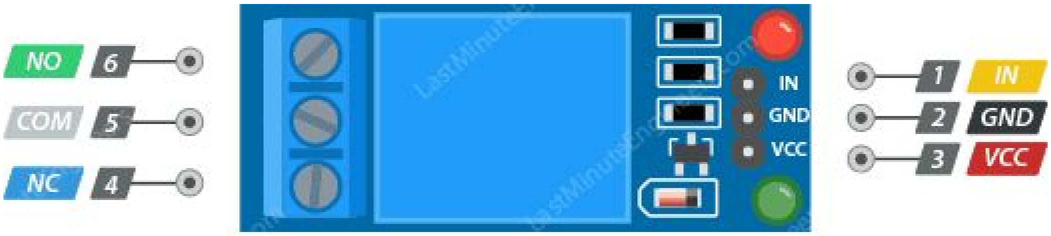
Echo: pin produces a pulse when the reflected signal is received. The length of the pulse is proportional to the time it took for the transmitted signal to be detected.

Gnd: used to connect to ground.

Note: You have to supply external 5v to this sensor as nodemcu gives only 3.3v output. You can also use vin pin from nodemcu but only if the board is connected via usb.

5v Relay:

A relay is an electromagnetic switch operated by a relatively small current that can control much larger current.



fig(5) One channel 5v relay pin mappings

In: This pin is used to control the relay mostly connected to arudino’s GPIO pin.

Vcc: 5v power supply.

GND: Should be connected to a ground of power supply.

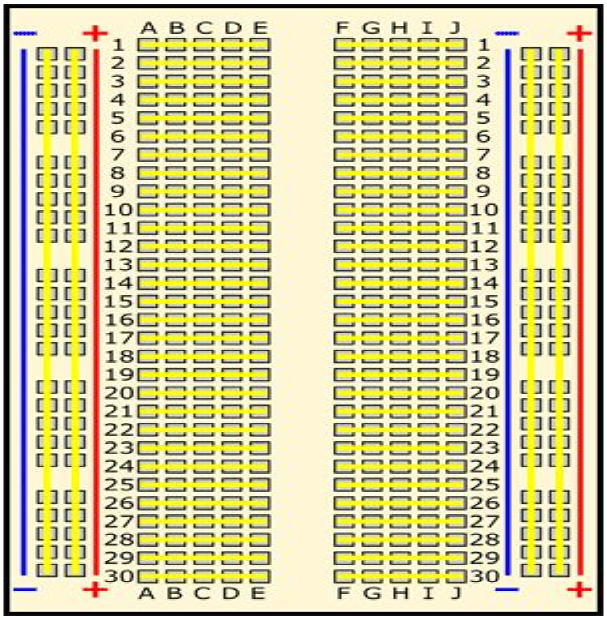
NO: Normally open(this circuit closes only when relay is triggered).

Com: Should be connected to the AC device you are trying to control.

NC: Normally closed(this circuits open when relay is triggered).

BreadBoard:

A breadboard is a base for prototyping of electronics projects.This is mainly used for learning and experiment purpose as it dont require any soldering it is reusable.



fig(6)BreadBoard

The inner strips are called Terminal strips which is where components will be inserted.The outer strips are called bus strips which used for supplying power and ground to the components.

As shown in fig(6) In a Breadboard terminal strips are connected HORIZONTALLY(row) and the Bus strips are connected VERTICALLY(column).

Battery pack:

As we are using a sensor and a relay which works on 5v but nodemcu supplies only 3.3v so we need to use external power supply. Here we are using 4x 1.5v AA batteries connected in series so they totally give 6v which will be enough to supply our sensor and relay.

Note:This cannot be used for nodemcu so we need to power nodemcu using usb or a 9v battery .

Fig(7) Battery pack case

Jumper wires:

A jumper wire is an electrical wire, or group of them in a cable, with a connector or pin at each end (Male or Female), 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

Fig(8) Jumper cables

Arduino IDE:

This software makes it easier to write sketches and upload it to your arduino or other development boards.This IDE is written in Java even though we write sketches for boards in c or c++.

To install:(Skip if you want to use web version)

Step 1:https://www.arduino.cc/en/Main/software visit this site and download zip file corresponding to your operating system.

Step 2: extract the zip file.

Online IDE:

Step 1:https://create.arduino.cc/editor/ visit this site.

Step 2 : Create an account in arduino.

Step 3: https://create.arduino.cc/getting-started/plugin/welcome You need to install this plugin in your system in order to upload code into your board.

Required Libraries:

- Blynk.h
- ESP8266WiFi.h

To install Libraries into arduino IDE:

Step 1: Open Arduino IDE

Step 2: Press ctrl+shift+I this will open a new window (Library manager).

Step 3: search the library you need to install.

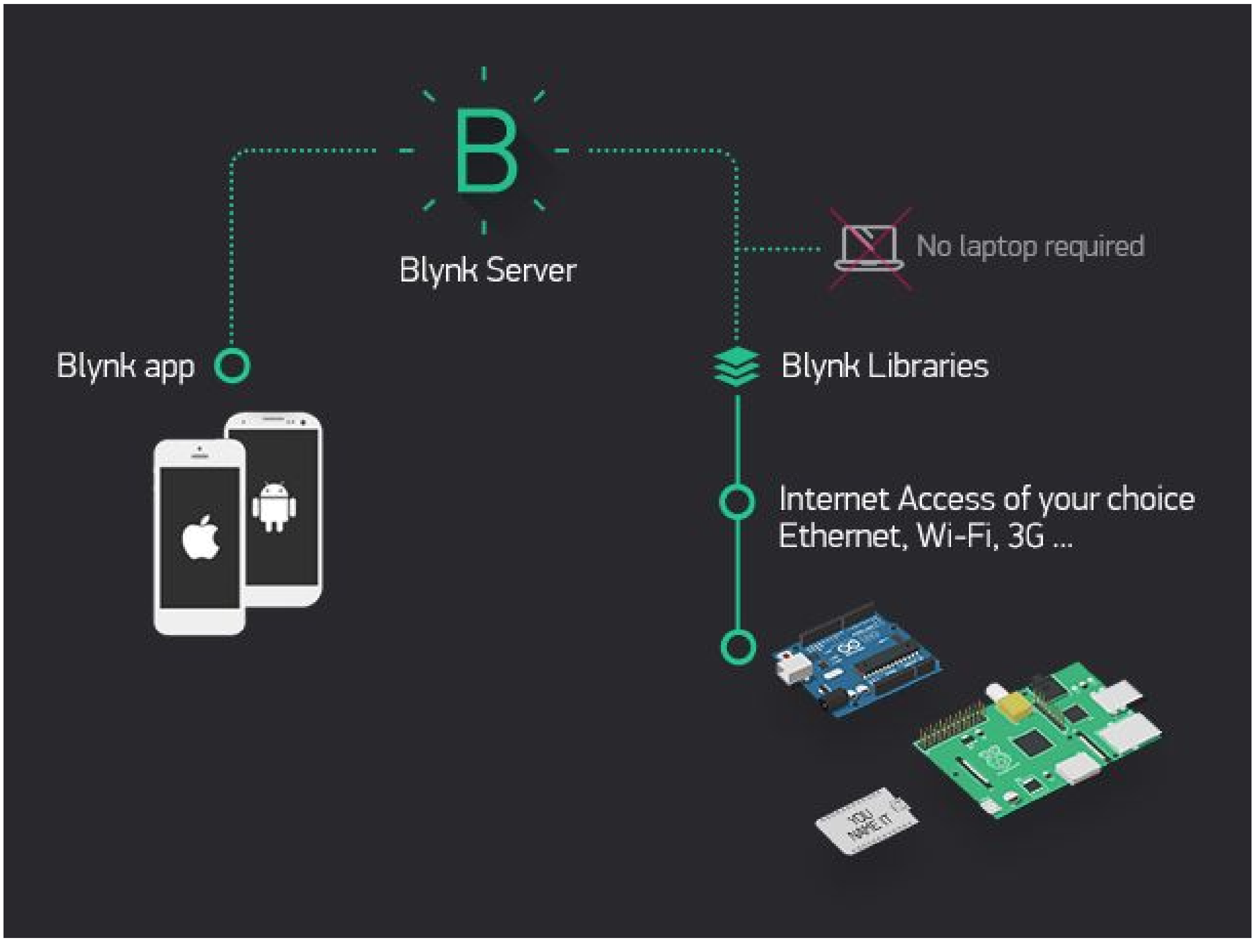
Step 4: select install.

Blynk:

Blynk was designed for the Internet of Things. It can control hardware remotely, it can display sensor data, it can store data, vizualize it and do many other cool things.

There are three major components in the platform:

- Blynk App - allows to you create amazing interfaces for your projects using various widgets we provide.
- Blynk Server - responsible for all the communications between the smartphone and hardware. You can use our Blynk Cloud or run your private Blynk server locally. It’s open-source, could easily handle thousands of devices and can even be launched on a Raspberry Pi.
- Blynk Libraries - for all the popular hardware platforms - enable communication with the server and process all the incoming and outcoming commands.



Fig(9)Working of Blynk

If This Then That(IFTTT):

If This Then That (IFTTT) is a web-based service that allows users to create chains of conditional statements triggered by changes that occur within other web services such as Ada Fruit, Facebook,instagram,google assistant,Amazon alexa,etc.

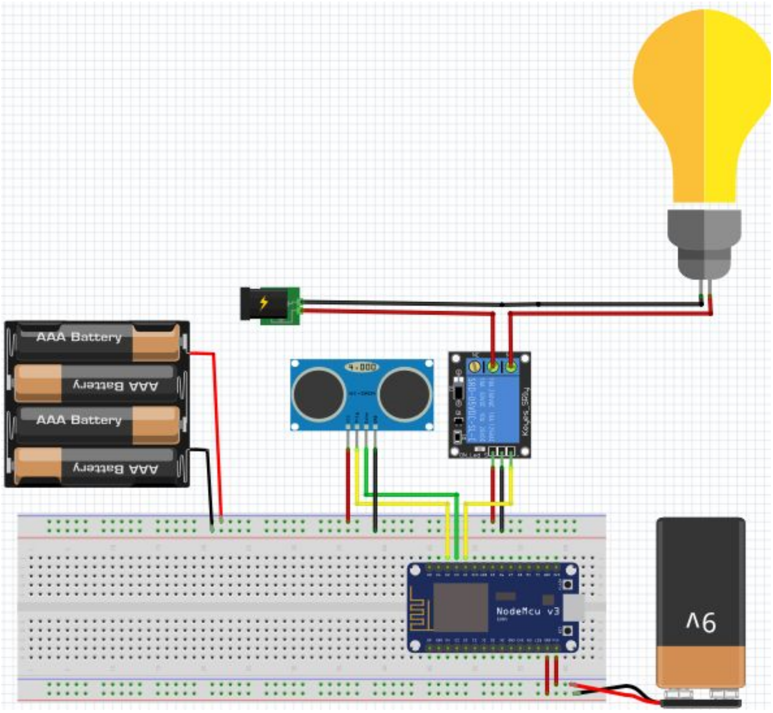
The conditional chains are called applets. An applet may for example send an e-mail message if the user tweets using a hashtag, or copy a photo on Facebook to a user's archive if someone tags a user in a photo.

Refer: https://ifttt.com

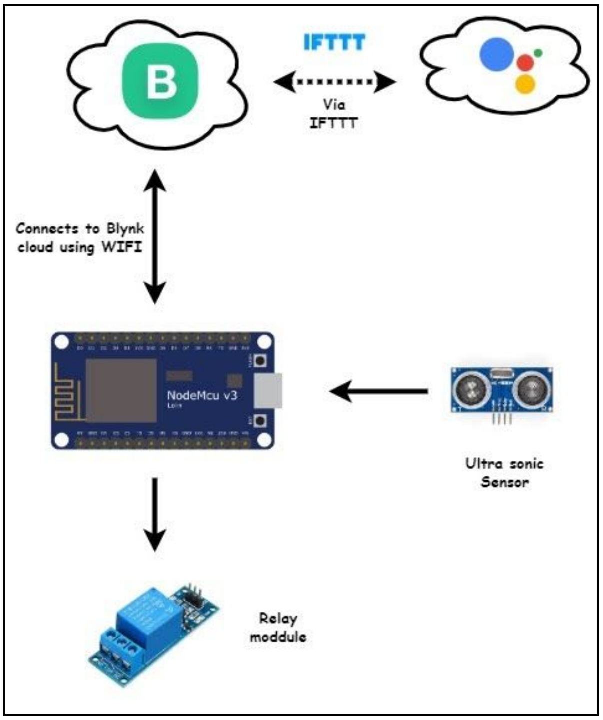
Google Assistant:

Google Assistant is a virtual assistant developed by google powered by Artificial intelligence.This can be mostly found on all android mobiles and other smart home appliances. Google assistant is capable of performing 2 way conversation. We are going to use this platform to control our system with voice command.

Circuit Diagram:



Fig(10)Circuit diagram



Fig(11)Working flow

Working and procedure:

This is an has three modules while setting up:

- Connecting Hardware components
- Uploading code into board
- Setting up blynk and IFTTT

Connecting hardware components:

Here we are using breadboard as we do it as a prototype but if you need to implement this in real life then you may need to solder things up. Below you can find brief about the components connection: (refer Fig(10))

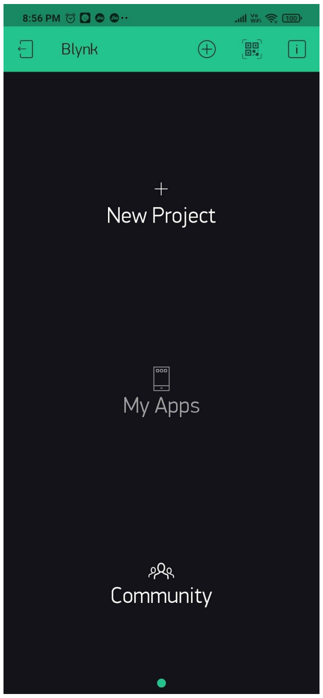
- Connect d1 pin from nodemcu to Trig pin HC-SR04
- Connect d2 pin from board to Echo pin in HC-SR04
- Connect a 6v battery pack to the HC-SR04 (i.e) battery's positive terminal to Vcc and negative terminal to Grnd.
- Now moving to Relay module connect the IN pin from relay module to d3 pin in your Nodemcu.
- Connect the battery pack to the relay module too (should Connect in parallel).
- Now for power supply for Nodemcu you can a cheap 9v battery(NOTE: You no need to supply external power if you have connected nodemcu to your computer via micro USB).

Setting up IFTTT and Blynk:

Blynk:

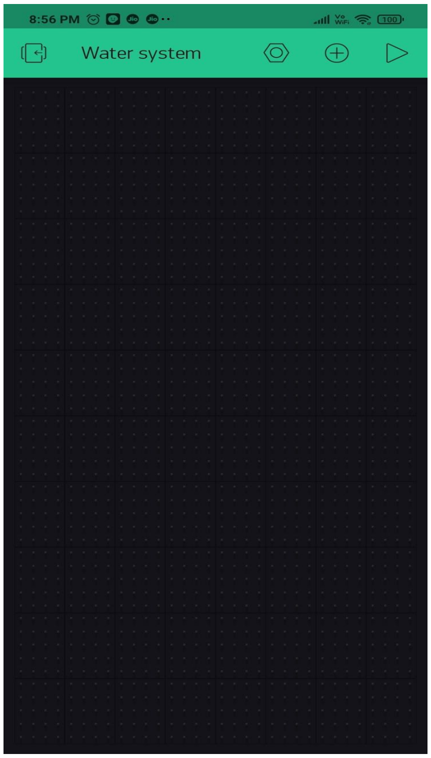
First of all you need to download and install the Blynk app on your mobile which you can do from the app store.Then you need to create an account you can also proceed by linking your existing Facebook Account.

Once you Finish setting up account you have home screen like as in fig(12) where you need to click new project.



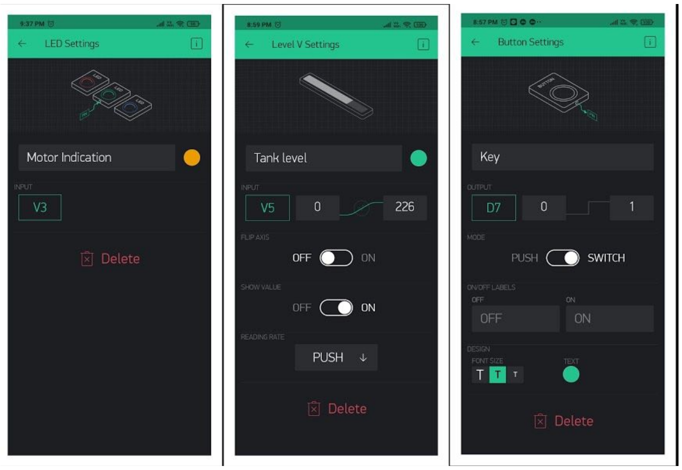
Fig(12)Blynk screen

After which you need to select Board as Nodemcu and connected type as WIFI then you will have a blank screen fig(13).



Fig(13)Blynk project screen

Now you need to place widgets like level,button and led within empty space by clicking + icon you may refer below screenshot to configure Widgets.



9:37 PM

LED Settings

Motor Indication

INPUT

V3

Delete

8:59 PM

Level V Settings

Tank level

INPUT

V5 0 226

FLIP AXIS

OFF ON

SHOW VALUE

OFF ON

READING RATE

PUSH ↓

Delete

8:57 PM

Button Settings

Key

OUTPUT

D7 0 1

MODE

PUSH SWITCH

ON/OFF LABELS

OFF OFF

ON ON

DESIGN

FONT SIZE

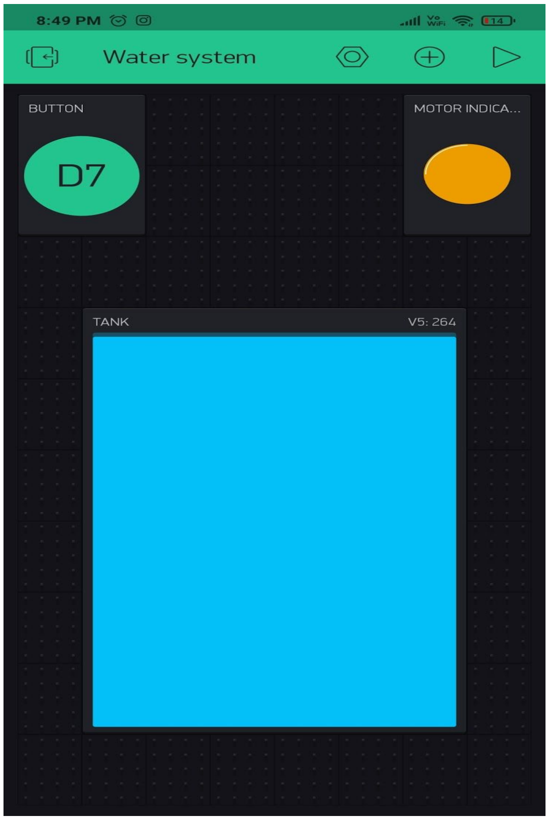
T T T

TEXT

Delete

Fig(14)Widgets Configuration

Once you finish setup you can start running your widgets by pressing RUN button on top right.



Fig(15)Blynk screen

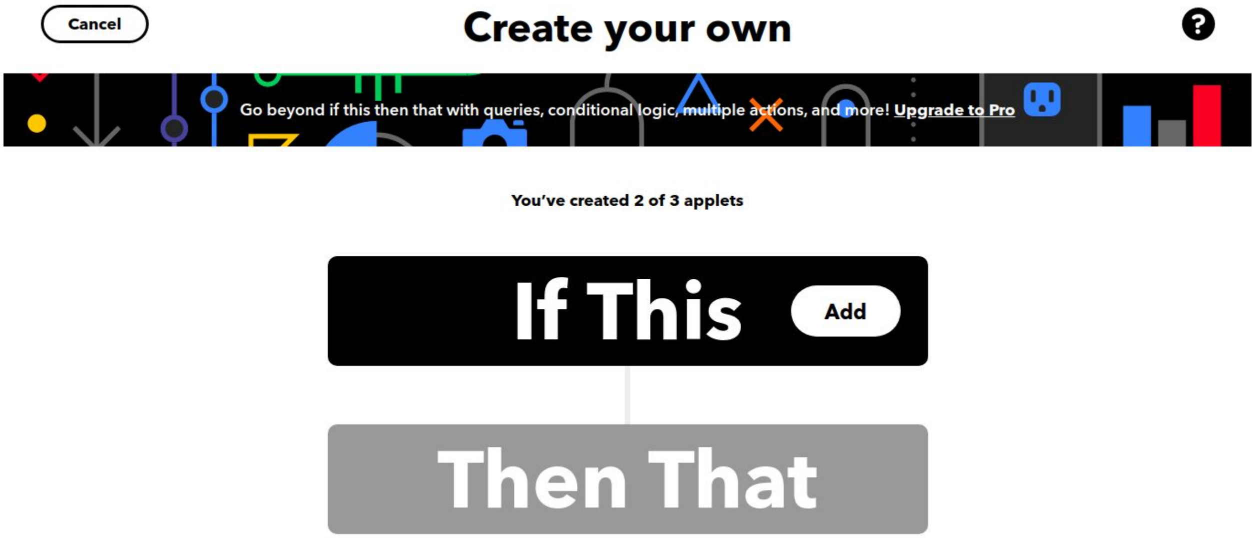
IFTT(If This Then That):

Google assistant cannot directly communicate with blynk so we are using a service called iftt which enables us to change Blynk values using google assistant.

Step 1: visit https://ifttt.com

Step 2: You may need to Create an account if you dont have one you can even use gmail/apple id.

Step 3:click on create button. Which brings you to create your own page.



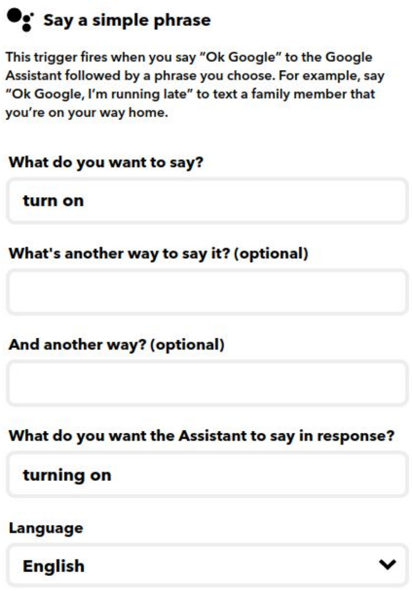
Go beyond if this then that with queries, conditional logic, multiple actions, and more! Upgrade to Pro

You've created 2 of 3 applets

Fig(16)IFTT

Step 4:Under “if this” select Google assistant.

Step 5: Select “say a simple phrase”.



Say a simple phrase

This trigger fires when you say "Ok Google" to the Google Assistant followed by a phrase you choose. For example, say "Ok Google, I'm running late" to text a family member that you're on your way home.

What do you want to say?

turn on

What's another way to say it? (optional)

And another way? (optional)

What do you want the Assistant to say in response?

turning on

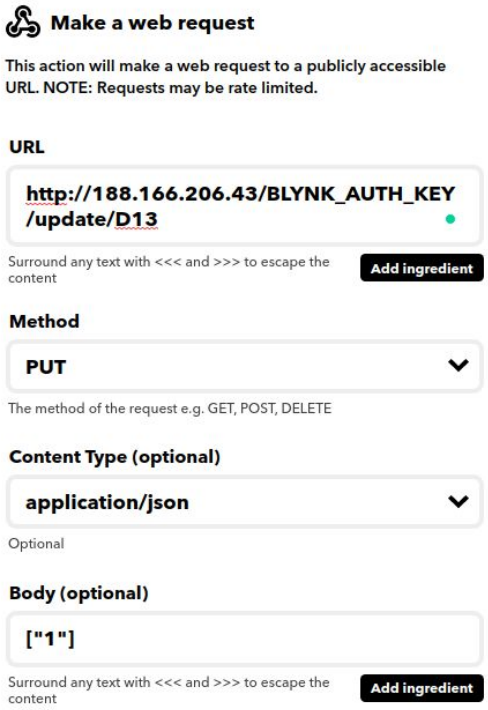
Language

English

Fig(17)IFTT"Google Assistant"

Step 6: you can fill phrases as your wish.

Step 7: you need to now select webhooks under "THEN THAT".



Make a web request

This action will make a web request to a publicly accessible URL. NOTE: Requests may be rate limited.

URL

http://188.166.206.43/BLYNK_AUTH_KEY /update/D13

Surround any text with <<< and >>> to escape the content

Method

PUT

The method of the request e.g. GET, POST, DELETE

Content Type (optional)

application/json

Optional

Body (optional)

["1"]

Surround any text with <<< and >>> to escape the content

Fig(18)IFTT "Webhooks"

Step 8: now you need to configure your webhooks as in Fig(18) NOTE: replace BLYNK_AUTH_KEY with you are BLYNK AUTH KEY which you receive to your mail after creating a new project in Blynk app.

Step 9: Click on Create action.(Now everything has been set up).

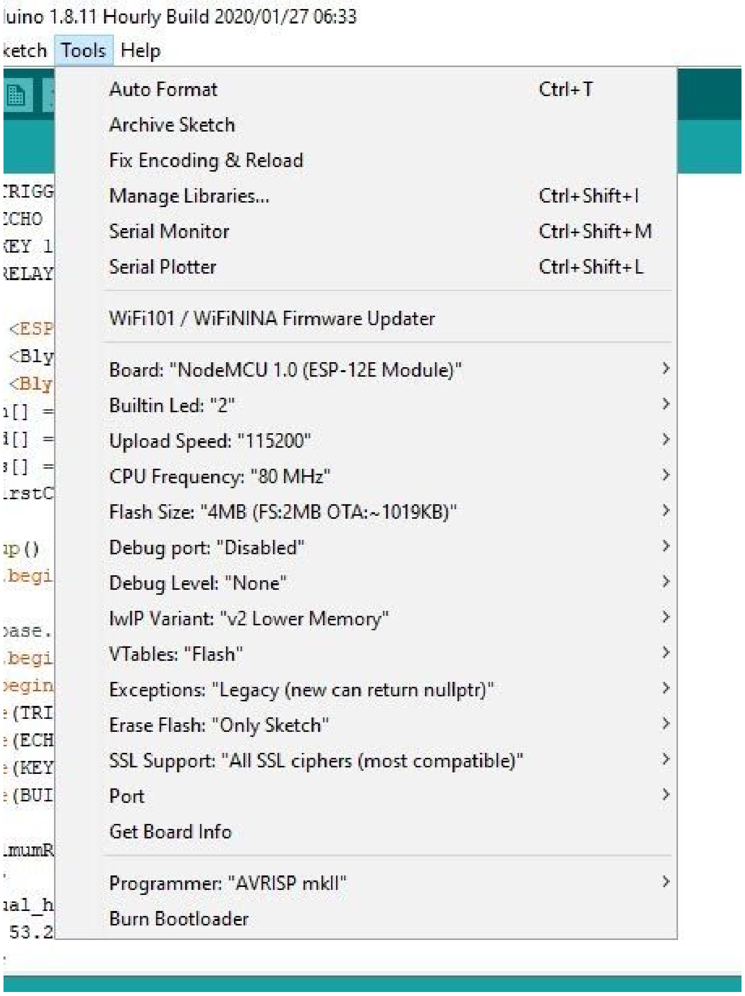
Note: you can also you IFTT mobile app.

Uploading code:

Connect your nodemcu to a computer or laptop via usb cable.Open Arduino IDE and change the settings to the same as in Fig(19).

Note:

- You have to install all the required libraries to compile the code.
- You may need to install nodemcu board from the board manager if you dont find nodemcu v1.0 in Tools->boards option.
- For the port number choose the port in which you have connected your board.



luino 1.8.11 Hourly Build 2020/01/27 06:33
ketch Tools Help
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
Port
Get Board Info
Burn Bootloader

Board: "NodeMCU 1.0 (ESP-12E Module)"
Builtin Led: "2"
Upload Speed: "115200"
CPU Frequency: "80 MHz"
Flash Size: "4MB (FS:2MB OTA:~1019KB)"
Debug port: "Disabled"
Debug Level: "None"
lwIP Variant: "v2 Lower Memory"
VTables: "Flash"
Exceptions: "Legacy (new can return nullptr)"
Erase Flash: "Only Sketch"
SSL Support: "All SSL ciphers (most compatible)"
Programmer: "AVRISP mkII"

Fig(19)Tools option

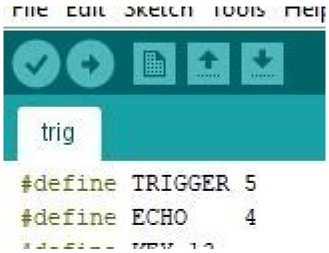
Copy the below code into Arduino IDE and enter the required details in respective places in the code.

CODE:

```
#define TRIGGER 5 //D1 pin

#define ECHO 4//D2

#define KEY 13 //D7 Controlled using blynk

#define RELAY 0 //D3


#include <ESP8266WiFi.h>

#include <BlynkSimpleEsp8266.h>

#include <Blynk.h>

char auth[] = " Enter the blynk auth id";

char ssid[] = "Enter your Wifi name here";

char pass[] = "Enter Wifi password here";


void setup() {

  Serial.begin (9600); //this command begins serial monitor

  Blynk.begin(auth, ssid, pass); //Connects to blynk server

  pinMode(TRIGGER, OUTPUT);

  pinMode(ECHO, INPUT);

  pinMode(KEY,INPUT);

  pinMode(BUILTIN_LED, OUTPUT);

}

int maximumRange = 30; //depth of the tank

float v; //volume

int actual_height; //actual height of water from bottom of the tank

int r = 53.2; //radius of the tank

int cap;//Capacity in Liters

WidgetLED led3(V3);//creates a object of widgetLED class


void loop() {
  Blynk.run(); //Begins to send and receive data from blynk server
  if(digitalRead(KEY)){
  long duration, distance;
  digitalWrite(TRIGGER, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIGGER, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIGGER, LOW);
  duration = pulseIn(ECHO, HIGH); //return time taken by the wave to reach back
*/
  distance= duration*0.034/2 //distance is calculated using speed=dis/time
formula  speed of sound 340m/s */
  Serial.print("Centimeter:"); //prints value in serial monitor
  Serial.println("distance");
  v = ((3.14*(r*r))*(maximumRange-distance));//volume calculated using
v=pi*(r^2)*h */
  cap = v/1000; //converting  into liters
  Blynk.virtualWrite(V5, cap); //returns cap to blynk server
  if(distance>=25){ //checks if the water level is low
      digitalWrite(BUILTIN_LED,LOW);
      digitalWrite(RELAY,HIGH); //turns on motor by triggering
      led3.on();
      Serial.println("low water");
      }
    if(distance<=5){ //checks if the tanks filled
      digitalWrite(BUILTIN_LED,HIGH);
      digitalWrite(RELAY,LOW); //turns off motor
      led3.off();
      Serial.println("FULL");
      }
  delay(1000);}//delays 1second
Else{
```

```
digitalWrite(RELAY,LOW);led3.off();} /*keeps the motor off if the user turns
off the system from blynk */

}
```

Once done press the upload button(Arrow button refer fig(20)). The program will be uploaded into your board if the program compiles successfully.You can also compile manually by pressing the verify button(Tick button refer fig(20)).



```
#define TRIGGER 5
#define ECHO    4
```

Fig(20)Upload and verify button

Conclusion:

This system is designed to automate water tank level maintaining processes. Usually we need to turn on motor and wait until water fills and turn it off again manually but this method is not efficient as it is time consuming and may wate a lot of water if you forgot to turn off the motor within time or even if you forgot to turn on motor you may be get locked by situation without water for necessary needs you my need to wait until water tank fills again.