PROJECT REPORT

On

Hands Free Remote Control

By

Sanjeev Bhardwaj

TABLE OF CONTENTS

Ch1. INTRODUCTION

1.1 Introduction

Every day we use Remote, to control our TV,AC,etc. But isn’t that a bit old fashioned? In today’s world where IOT(Internet Of Things) is growing rapidly with day by day more improvements into making our life easier. So what is IOT? Internet Of Things is a system of connected devices through the internet. It involves mechanical devices, sensors,home appliances,vehicles etc apart from desktop, mobile and laptop. These devices are designed in such a way that they can share data with other devices over the internet.IOT basically provides a platform for devices to interact and collaborate with each other. Now, establishing that, our mini project is hands free remote,to give a brief introduction the user can by means of hand gesture or by typing a simple message on the Google assistant can control IR(Infrared) receiver devices such as TV,AC,Home Theater, Projectors etc,from anywhere in the world.

1.2 Aim and Objective

The reason we made this project is because the remote we use to control IR receiver devices have some disadvantages to them:
1. Batteries: We have to keep changing batteries of the remote in regular interval,using rechargeable batteries could be expensive.

2. Life-span: A normal remote has a life span of around 1 to 2 years after that we have to keep changing remote which also is costly.

3. Maintenance: We all know that it is difficult to maintain a remote, we sometimes misplace it, brake it which leads to anger and frustration.

To overcome the above disadvantages, we made our mini project which eliminates all the above disadvantages. Using our project a user can control IR devices by means of hand-gestures such as near, far etc and not only this but, the user can also control the IR devices by Google assistant, we used Google assistant due to its availibilty in almost every smart phones and its easy and interactive usage I.e. typing a simple message to giving voice commands through the phone.
Advantages:

1. No more changing of batteries, the equipment comes with a rechargeable power-bank that can power the equipment up to 5 hours on single charge, can also be plugged in the wall adapter for continuous power.

2. No more misplacing the remote since there would be no remote, the user can just control the appliance either by hand gestures or voice commands anywhere in the world.

3. The equipment has a life-span of around 4 to 5 years and also can be reprogrammed to communicate with different IR receiver appliance in the house.

Ch.2 Review of Literature

2.1 Requirements for the real world problems

- **Need at home:**

As we have established that using remote is now outdated since we have IOT to save our lives from basic tasks such as battery changing or remote repairing or to buy a new remote, hence if we replace a remote by this device, we could simply make our life much more easier, we no more have to worry about any battery changing or buying a new remote as this equipment comes with rechargeable battery and a great life-span.
Secondly the user can control his/her remote from anywhere around the world just by typing a simple message. This would surely make our life easier.

- **Need at Office and colleges**:

As we know offices have many AC’s, Projectors,TV installed which have could be of different companies and hence could lead to maintaining different remote for different devices which could lead to time waste of employees if any of the remote stops working and there's nothing they can do to use the particular appliance,secondly a office or college using our project could eliminate all of this and control all the devices with a single device, reducing the cost and maintaining the remotes of different devices.

2.2 Existing Solutions

- **Inbuilt IR blaster in Phones**:

We all are aware of that some mobile brands provide an in built IR blaster in built in their android phones by which the users can control their IR receiver devices mobile phone companies such as MI,HTC,OPPO,VIVO provide an inbuilt IR blaster in some range of their phones. Theses phones have an in built software that can control almost every device registered as they have directory with which each appliances encoding and of IR signals and is easy to set-up. But there are other phones in market which do not provide an inbuilt IR blaster example: HUAWEI,APPLE,1^{+}, do not provide it hence a user using such smart phones cannot use the functionalities.

2.3 Hardware and Software requirements with cost

Components Required

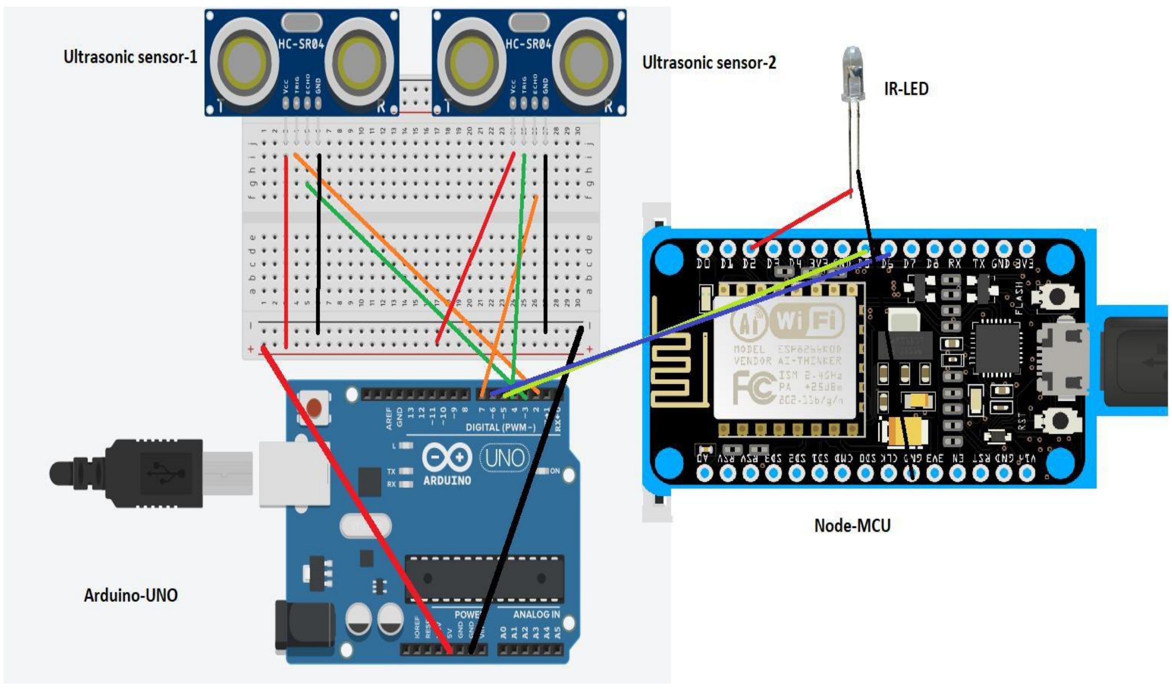
▪	1X Arduino NODEMCU(ESP8266)	-Rs. 339/-.
▪	1X Arduino UNO	-Rs. 450/-
▪	2X Ultrasonic Sensor(HCSR04)	-Rs. 270/-
▪	1X INFRARED LED	-Rs. 5/-
▪	1X 9V battery	-Rs. 25/-
▪	1X Jumper wires	-Rs. 20/-

TOTAL COST – Rs. 1109/

Ch3.DESIGN AND IMPLEMENTATION

3.1 Design Consideration

In this design we have used Node MCU(ESP8266) ,Arduino UNO, Ultrasonic Sensors, IR led.



Working

Make the connections as given in the circuit diagram then upload the node mcu code in NodeMCU using Arduino IDE and repeat the same steps for uploading the Arduino code to Arduino UNO using the Arduino IDE. The trigger and echo of the Ultrasonic Sensor-1 is connected to digital pin-2,digital pin-3 of Arduino UNO respectively. The trigger and echo of the Ultrasonic Sensor-2 is connected to digital pin-4,digital pin-4 of Arduino UNO respectively, The digital Pin-5 and Digital Pin-6 of Arduino Uno is connected to Digital pin 5 and Digital Pin 6 of NodeMCU respectively, The anode of IR led is connected to pin Digital Pin-2 of the Node MCU, The Vcc and ground pin of the ultrasonic sensors is connected in breadboard positive and ground supply.

As soon as the Ultrasonic sensor detects a gesture or the Google assistant sends a message through making a http request the IR led Glows and sends the signal to the IR receiver device and performs functions such as in case of TV, the TV will switch on or off or volume up or volume down.etc.

3.2 Design details

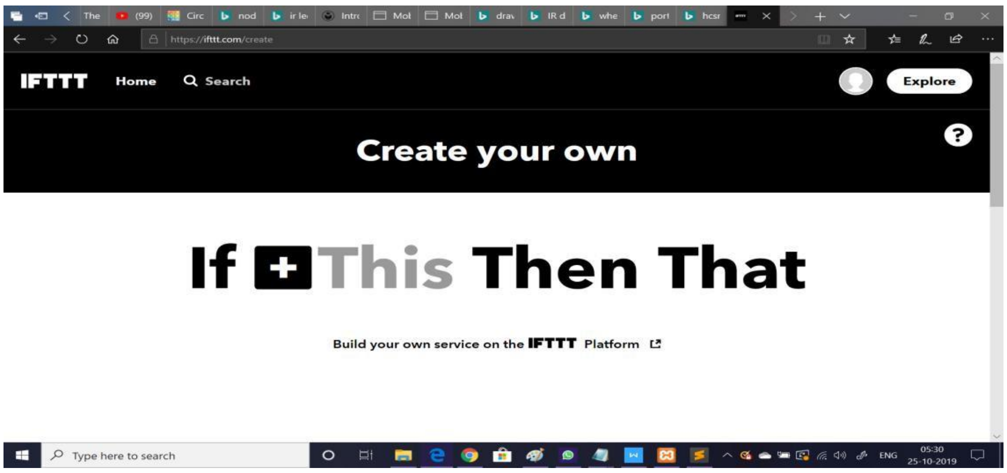
In this we are using the node mcu with IR led. As Arduino.cc began developing new MCU boards based on non-AVR processors like the ARM/SAM MCU and used in the Arduino Due, they needed to modify the Arduino IDE so that it would be relatively easy to change the IDE to support alternate toolchains to allow Arduino C/C++ to be compiled for 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 for the target MCU's machine language. Some ESP8266 enthusiasts developed an Arduino core for the ESP8266 WiFi SoC, popularly called the "ESP8266 Core for the Arduino IDE".This has become a leading software development platform for the various ESP8266-based modules and development boards, including NodeMCUs. In the nodemcu code, we have included a library to send IR signals This is done beacuse the IR signals are first encoded and then transferred through the air so as to prevent mixing or corruption of signals by other devices.On the other hand we have Arduino with the two ultrasonic sensors attached to it, the ultrasonic sensors, as we know measure the distance between its transmitter and the object placed right in front of the transmitter with a 95% accuracy hence Its easier to determine gestures using the distance values, for instance if the hand goes near the ultrasonic sensor 1, I.e. the hand is in between 5 to 10 cm the volume up command will be sent to the Node MCU and accordingly the IR led will transmit the signals.
Third and a major part in our project is the serial communication, we use the library “SoftwareSerial.h” In order to establish communication between the Node MCU and the Arduino, the digital pins 5 and 6 on both the boards work as transmitter and receiver so when the Arduino detects a gesture, we will send that to the node mcu and the node mcu will automatically fire the led.

3.3 GUI Design.

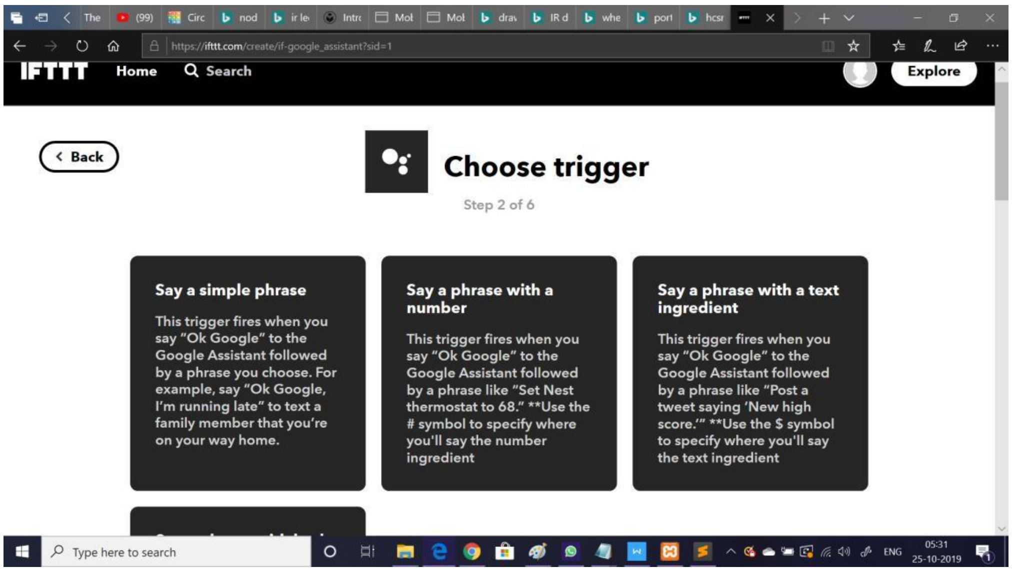
3.3.1 IFTTT:

Steps to create a trigger using IFTTT:

1. Goto create in the user drop down you will be directed to the following window:



2. Click on “this”, you will be forwarded to the following page where search google assistant and upon clicking that you will be forwarded here:



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.

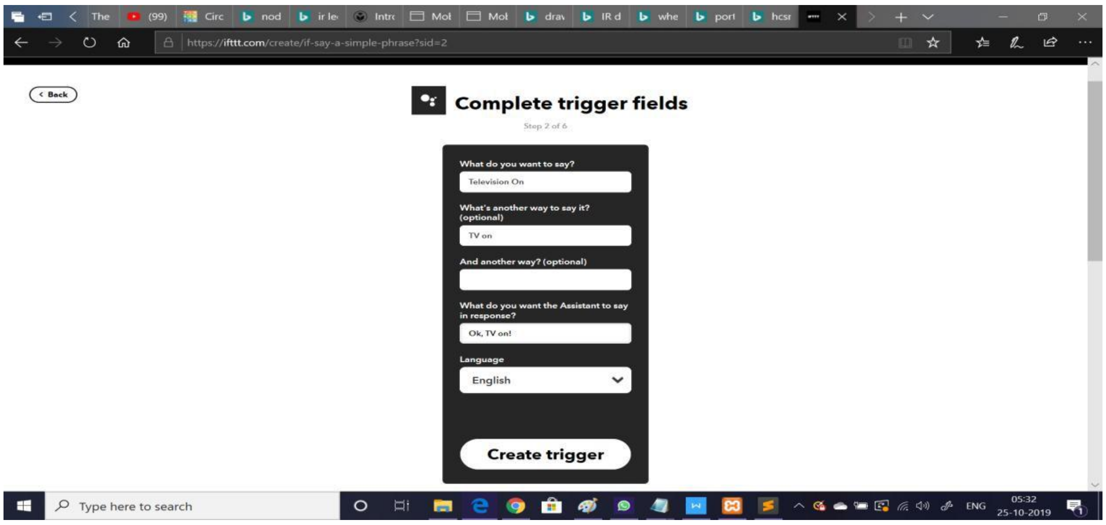
Say a phrase with a number

This trigger fires when you say “Ok Google” to the Google Assistant followed by a phrase like “Set Nest thermostat to 68.” **Use the # symbol to specify where you'll say the number ingredient

Say a phrase with a text ingredient

This trigger fires when you say “Ok Google” to the Google Assistant followed by a phrase like “Post a tweet saying ‘New high score.’” **Use the $ symbol to specify where you'll say the text ingredient

3. Click on “say a simple phrase” and fill in the following columns based on yr inputs to the Google assistant.


https://ifttt.com/create/if-say-a-simple-phrase?sid=2

Complete trigger fields

Step 2 of 6

What do you want to say?
Television On

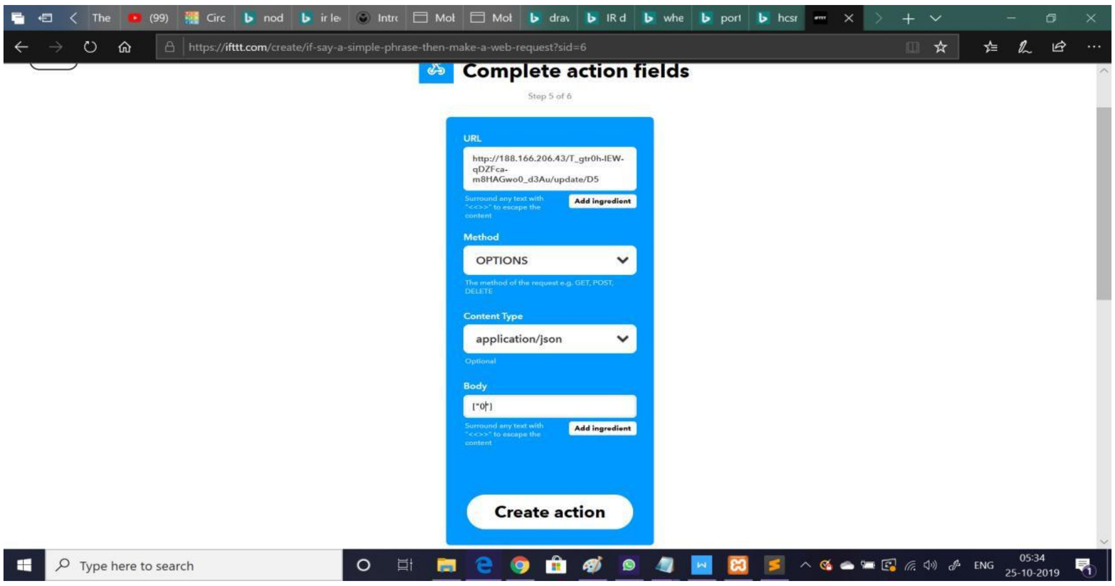
What's another way to say it? (optional)
TV on

And another way? (optional)

What do you want the Assistant to say in response?
Ok, TV on!

Language
English

4. After clicking the "Create Trigger" button you will be forwarded to the page in step-1 and the this logo would be highlighted with google assistant logo,now click on that and search webhooks and click it then click on "Make a Web Request" and fill the details in the following form keeping the ip address same but changing the pin number, authentication key of the project(BLYNK APP).


https://ifttt.com/create/if-say-a-simple-phrase-then-make-a-web-request?sid=6

Complete action fields

Step 5 of 6

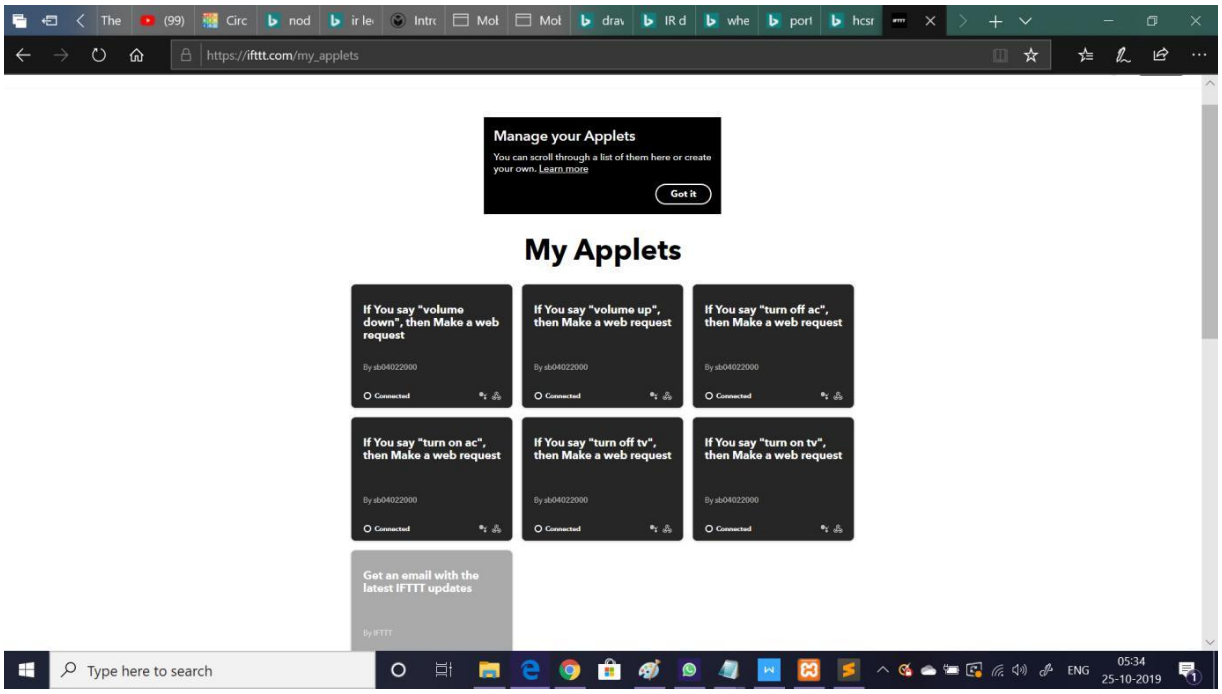
URL
http://188.166.206.43/T_gtr0h-lEW-qDZFca-m8HAGwo0_d3Au/update/D5
Surround any text with "<<>>" to escape the content

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

Content Type
application/json
Optional

Body
["0"]
Surround any text with "<<>>" to escape the content

5. Create rest of the triggers repeating the above steps,


https://ifttt.com/my_applets

Manage your Applets
You can scroll through a list of them here or create your own. Learn more
Got it

My Applets

If You say "volume down", then Make a web request
By sb04022000
Connected

If You say "volume up", then Make a web request
By sb04022000
Connected

If You say "turn off ac", then Make a web request
By sb04022000
Connected

If You say "turn on ac", then Make a web request
By sb04022000
Connected

If You say "turn off tv", then Make a web request
By sb04022000
Connected

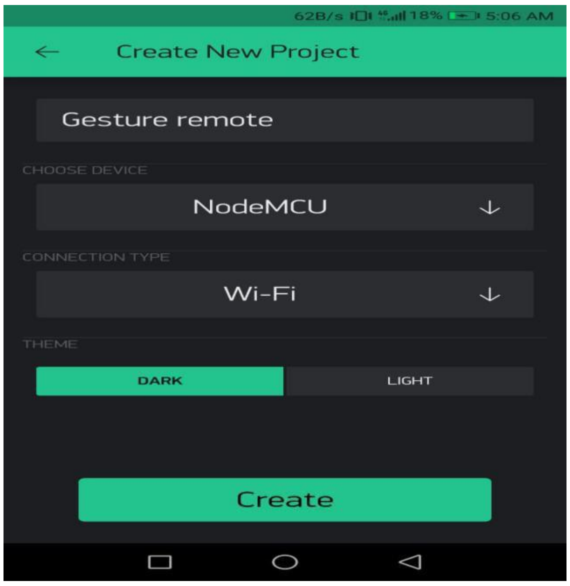
If You say "turn on tv", then Make a web request
By sb04022000
Connected

Get an email with the latest IFTTT updates
By IFTTT

3.3.2 NodeMCU:

For the Gui design we are using Blynk app. Blynk for NodeMCU – Introduction
1. Introduction:
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.
1.2 Supported Hardware
1.2BlynkApp Blynk application can be found from the following
1. Android Blynk App
2. IOS Blynk App
You'll also need to install the Blynk Arduino Library, which helps generate the firmware running on your ESP8266. Download the latest release from https://github.com/blynkkk/blynk-library/releases , and follow along with the directions there to install the required libraries.

2. Create a Blynk Project
Click the "Create New Project" in the app to create a new Blynk app. Give it any name.Blynk works with hundreds of hardware models and connection types. Select the Hardware type. After this, select connection type. In this project we have select WiFi connectivity.


Create New Project

Gesture remote

CHOOSE DEVICE: NodeMCU

CONNECTION TYPE: Wi-Fi

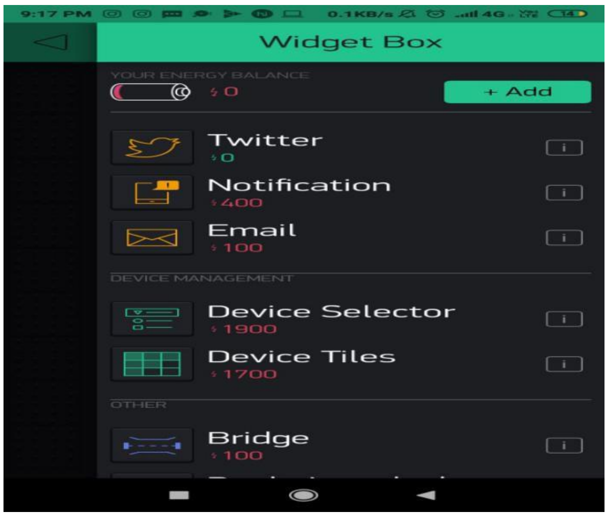
THEME: DARK | LIGHT

Create

After clicking the create button,The *Auth Token* is sent to registered email id and is very important – you'll need to stick it into your Node MCU's firmware. For now, copy it down or use the "E-mail" button to send it to yourself.

3. Add Widgets To The Project

Then you'll be presented with a blank new project. To open the widget box, click in the project window to open.
We are selecting a button to control Led connected with NodeMCU.
Click on Button.
Give name to Button say Email.
Under OUTPUT tab- Click text/html or text/plain. and also write the emailed that is connected to NodeMCU and Click continue.
Under MODE tab- Select whether you want this button as "push button" or "Switch".


Widget Box

YOUR ENERGY BALANCE	⚡0
Twitter	⚡0
Notification	⚡400
Email	⚡100
DEVICE MANAGEMENT	
Device Selector	⚡1900
Device Tiles	⚡1700
OTHER	
Bridge	⚡100

4. Upload The Code:
Now that your Blynk project is set-up, open Arduino and navigate to the ESP8266_Standalone example in the File > Examples >Blynk>Boards_WiFi> ESP8266_Standalone menu.
Code:
Before uploading, make sure to paste your authorization token into the auth [] variable. Also make sure to load your Wifi network settings into the Blynk.begin(auth, "ssid", "pass") function.
Node MCU code:

```
int data;
#define BLYNK_PRINT Serial //for serial communication
#include <ESP8266WiFi.h> //in built wifi module
#include<Arduino.h>
#include <IRremoteESP8266.h> // IRremote library for esp8266 module
#include<SoftwareSerial.h> // for serial communication
#include <IRsend.h> // to send IR signals
#include <BlynkSimpleEsp8266.h> // To communicate with Blynk app
SoftwareSerial s(D6,D5); // declaring two pins for serial communication
#define inPin1 D7 // tv on/off input pins, to be controlled by google assistant
#define inPin2 D0 // mute on/off
#define inPin3 D1 // vol up
#define inPin4 D4 // vol down
int val1,val2,val3,val4;
const uint16_t kIrLed = 4; // output pin(IR led)
IRsend irsend(kIrLed);
// You should get Auth Token in the Blynk App.
// Go to the Project Settings (nut icon).
char auth[] = "T_gtr0h-IEW-qDZFca-m8HAGwo0_d3Au";

// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "sanjeev"; //Hotspot or wifi name
char pass[] = "12345678"; // Hotspot or Wifi password

void setup()
{
  // Debug console
  irsend.begin();
  s.begin(9600); // starting serial communication with arduino
#if ESP8266
  Serial.begin(9600, SERIAL_8N1, SERIAL_TX_ONLY);
#else // ESP8266
```

```
  Serial.begin(9600, SERIAL_8N1);
#endif // ESP8266
  pinMode(inPin1, INPUT); // declaring input pins
  pinMode(inPin2, INPUT);
  pinMode(inPin3, INPUT);
  pinMode(inPin4, INPUT);
  Blynk.begin(auth, ssid, pass);
}

void loop()
{
  Blynk.run();
  val1 = digitalRead(inPin1); // reading the state of pins
  val2 = digitalRead(inPin2);
  val3 = digitalRead(inPin3);
  val4 = digitalRead(inPin4);
    data=s.read(); // reading any data from aruino
    Serial.println(data); // for debugging
  if (val1 == LOW || data == 0){ // tv on/off
     irsend.sendNEC(0x80BF3BC4, 32); // sending IR signal on condition true
     if(val1 == LOW)
    digitalWrite(inPin1,HIGH);
  }
  else if (val2 == LOW || data == 3){ // mute on/off
    irsend.sendNEC(0x80BF39C6, 32); // sending IR signal on condition true
    if(val2 == LOW)
    digitalWrite(inPin2,HIGH);
  }
  else if (val3 == LOW || data ==
    1){ Serial.println("3");// vol up
    if(val3 == LOW)
    {
    for(int i=0; i<11; i++)
    {
     irsend.sendNEC(0x80BFBB44, 32); // sending IR signal on condition true
     delay(500);
    }
    digitalWrite(inPin3,HIGH);
    }
    irsend.sendNEC(0x80BFBB44, 32); // sending IR signal on condition true
  }
  else if (val4 == LOW || data ==
    2){ Serial.println("4");// vol down
    if(val4 == LOW)
    {
    for(int i=0; i<6; i++)
    {
     irsend.sendNEC(0x80BF31CE, 32); // sending IR signal on condition true
     delay(500);
    }
   digitalWrite(inPin4,HIGH);}
   irsend.sendNEC(0x80BF31CE, 32); // sending IR signal on condition true
  }
}
```

3.3.3 Arduino:

Now, after setting up the NodeMCU, we will now set up arduino,

1. Including the <SoftwareSerial.h> library in the code, First and foremost we have to include the Software Serial library in the code in order to ensure communication between the Arduino UNO and Node MCU, we don’t need any special UI for arduino and just have to upload the code from the Arduino Ide To the Arduino UNO.

2. Code:

```
#include <SoftwareSerial.h>
SoftwareSerial s(5,6); // defining the pins for serial communication
const int trigger1 = 2; //Trigger pin of 1st Sensor
const int echo1 = 3; //Echo pin of 1st Sensor
const int trigger2 = 4; //Trigger pin of 2nd Sensor
const int echo2 = 7;//Echo pin of 2nd Sensor
long time_taken;
int data=0;
int dist,distance_L,distance_R;

void setup()
{ Serial.begin(9600);
pinMode(trigger1, OUTPUT);
pinMode(echo1, INPUT);
pinMode(trigger2, OUTPUT);
pinMode(echo2, INPUT);
s.begin(9600);
}

void calculate_distance(int trigger, int echo) // function for calculating distance
{
digitalWrite(trigger, LOW);
delayMicroseconds(2);
digitalWrite(trigger, HIGH);
delayMicroseconds(10);
digitalWrite(trigger, LOW);
time_taken = pulseIn(echo, HIGH);
dist= time_taken*0.034/2;
```

```
if (dist>50)
dist = 50;
}
void loop() { //infinite loopy
calculate_distance(trigger1,echo1);
distance_L =dist; //get distance of left sensor
calculate_distance(trigger2,echo2);
distance_R =dist; //get distance of right sensor
//Uncomment for debudding
//Serial.print("L=");
//Serial.println(distance_L);
//Serial.print("R=");
//Serial.println(distance_R);

if ((distance_L >30 && distance_R>30) && (distance_L <50 &&
distance_R<50))
{Serial.println("Play/Pause");
data = 0;
 s.write(data);
delay (500);}
calculate_distance(trigger1,echo1);
distance_L =dist;
calculate_distance(trigger2,echo2);
distance_R =dist;

if (distance_L>=10 && distance_L<=20)
{
delay(100); //Hand Hold Time
calculate_distance(trigger1,echo1);
distance_L =dist;
if (distance_L>=10 && distance_L<=20)
{
//Serial.println("Left Locked");
while(distance_L<=40)
{

calculate_distance(trigger1,echo1);
distance_L =dist;
if (distance_L<10) //Hand pushed in
{Serial.println ("Vup");
data = 1;
 s.write(data);
delay (300);
}
if (distance_L>20) //Hand pulled out
{Serial.println ("Vdown");
data = 2;
s.write(data);
delay (300);
}
} }
}

if (distance_R>=10 && distance_R<=20)
{
delay(100); //Hand Hold Time
calculate_distance(trigger2,echo2);
distance_R =dist;
if (distance_R>=10 && distance_R<=20)
{
//Serial.println("Right Locked");
while(distance_R<=40)
{
calculate_distance(trigger2,echo2);
distance_R =dist;
if (distance_R<10) //Right hand pushed in
{Serial.println ("Rewind");
data = 3;
 s.write(data);
 delay (300);}
if (distance_R>20) //Right hand pulled out
{Serial.println ("Forward");
data = 4;
 s.write(data);
delay (300);}
}
}
}
delay(200);
}
```

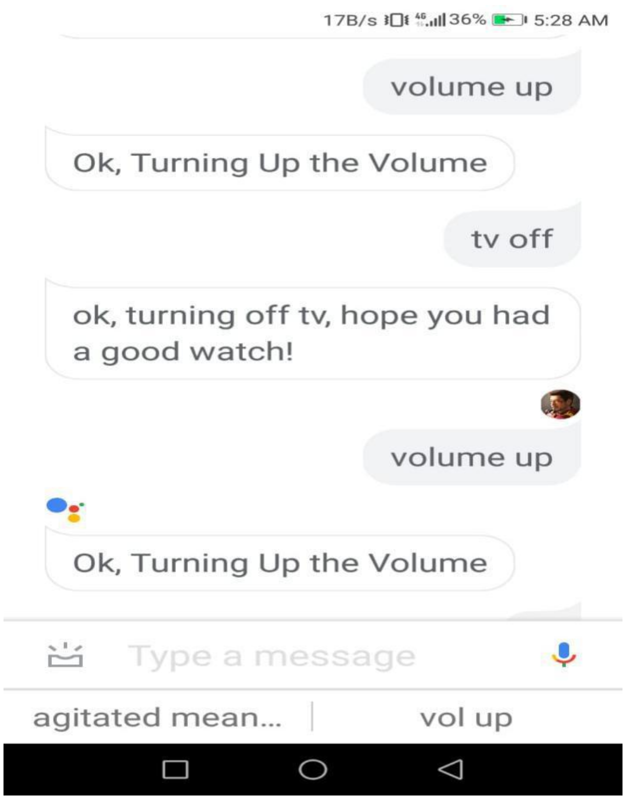
Ch4.DESIGN AND IMPLEMENTATION

4.1 Implementation Details:

After Uploading the code to the Arduino UNO and Node MCU respectively, Remove them from the transmitting cable and power them , the Arduino UNO is powered by a 9V battery and the Node MCU can be powered by a power-bank through the micro usb cable,we have to wait for a buffer time of about 5 seconds for the system to set up and establish communications among the to boards. Till then start your hotspot, the node mcu will automatically connect to the wifi,Now, we are ready to use the equipment.

4.2 Results and Evaluation:

To send IR signals to the appliances, long press the home button on your android smart phone(keeping in mind that the registered email is signed in on the particular phone)
Then type the messages that you have created the triggers with and you will get the following output with the IR led being fired.


17B/s 36% 5:28 AM

volume up

Ok, Turning Up the Volume

tv off

ok, turning off tv, hope you had a good watch!

volume up

Ok, Turning Up the Volume

Type a message

agitated mean... | vol up

Secondly, If the user performs hand gestures such as:

1. Near: keeping your hand near the ultrasonic sensor, until the your hand is removed the IR led will keep firing IR signals.

2. Near to Far: keeping your hand near the ultrasonic sensor and slowly moving your hand backwards will send another signal through the IR led.

3. Both Palms closing to wards the sensors: Facing both the palms towards the sensor and keeping them about a distance of 25cm from the sensor will send another signal.

Using these gestures, the user can communicate with the IR receiver devices and send signals.

Conclusion and Future Work

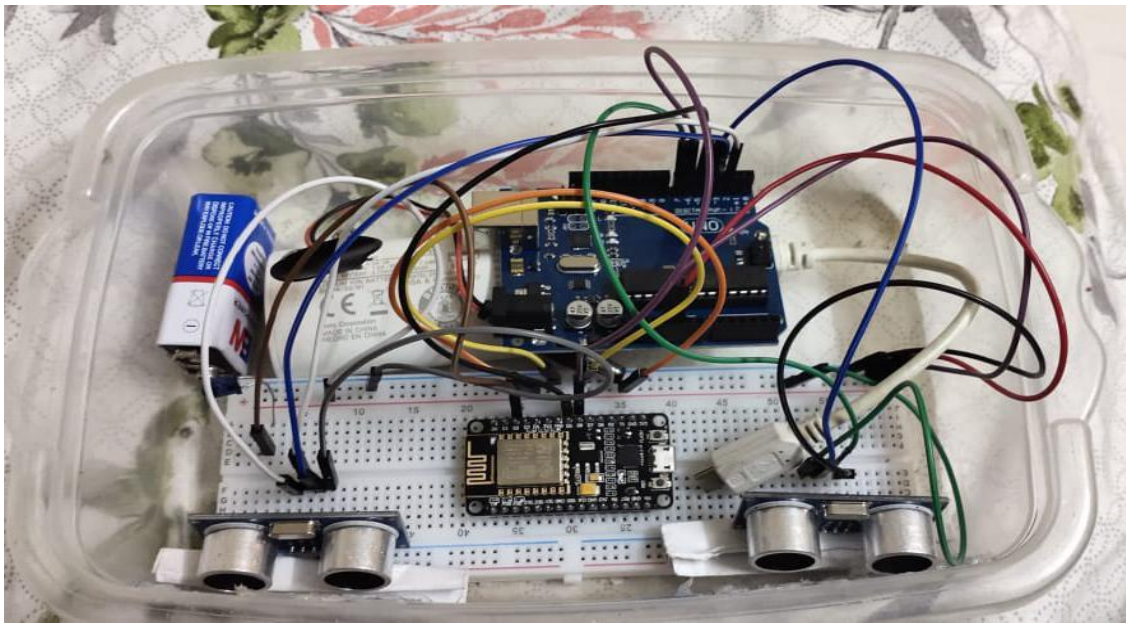
Conclusion

Our project depicts the necessity and an efficient solution for fire safety. Internet of Things was the main concept used and the project mainly builds on the techniques which are already present with an additional feature of mailing in emergency. But still there are few tweaks and remodeling required to get a more efficient and working model.To get a brief idea of the limitations of the model, The sensors could take some buffer time in between to optimize their readings the buffer time could be anywhere in between 3 to 5 seconds and after this buffer period the sensors would keep on working efficiently on the other side, the assistant is not affected during the buffer time and commands can be given.

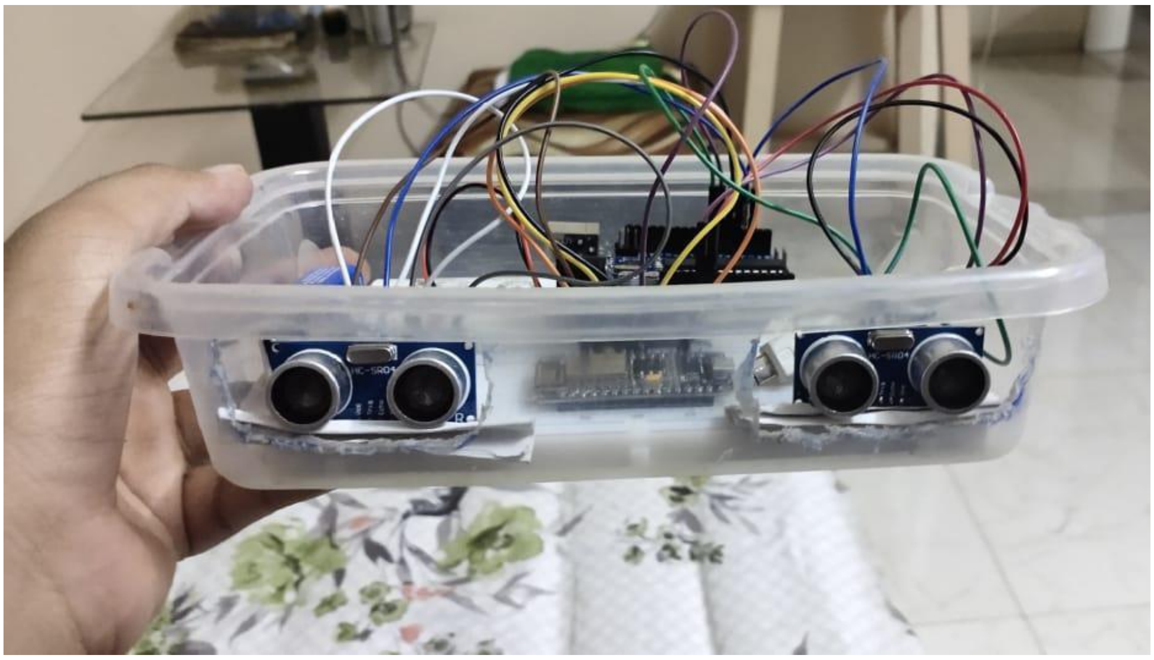
Future Work

Currently the gesture sensor is not fully optimized hence gesture could be misjudged also in future we can implement a full ir directory where IR codes with encryption style is stored and upon gesture or a message these can be dynamically used to communicate with IR receiver devices.

We can also implement more gestures with the ultrasonic sensor and optimize the code in order to eliminate the hassle of more coding we could also replace the ultrasonic sensors with the APDS-9960 gesture sensor which has its own library and could detect upto 6 gestures without anything to write complex code for, also by doing this we could eliminate the need for using the Arduino UNO board as the gesture sensor can directly be connected with Node MCU.



The above figure shows the top view of the project containing the wiring,battery connection and component body.



The above figure shows the front view of the project which shows the two ultrasonic sensors to take input from the user.

Appendix

In the most recent decade, medicinal services has drawn extensive measure of consideration. The measure of information to be followed identified with its region of application is likewise colossal.

Remembering this and review the requirement for better human services office we concocted making a hands free remote, which could also be voice controlled. There are three angles to our venture. The prime objective was to eliminate the use of remote in our day to day life and make it easier.

There are three angles to our undertaking. The user would no longer have to worry about a remote loosing its battery power or malfunctioning anymore as the product completely eliminates the possibility of these

At first, the users need to reach for remote in order to control their IR receiver devices but now we have completely erased that possibility.

We got a great amount of help in building this project with a vast amount of libraries in arduino to control the sensors, led and communicate with the servers.

Referenes:

1. Blynk Library

2. IRremoteESP8266 Library

3. SoftwareSerial Library

4. https://www.udemy.com/course/arduino-bootcamp/