Code: Transmiter Section

Controller: ESP8266 NodeMcu (WiFi Module)

```
#define BLYNK_PRINT Serial    //    initialization of blynk serial communication
#include <BlynkSimpleEsp8266.h> // initialization of controller nodemcu with blynk
#include <Blynk.h>    // initialization of blynk application
#define trig1 D1        // defining ultrasonic1 trig pin is conected to D1 of nodemcu
#define echo1 D0        // defining ultrasonic1 echo pin is conected to D0 of nodemcu
#define trig2 D2        // defining ultrasonic2 trig pin is conected to D2 of nodemcu
#define echo2 D3        // defining ultrasonic2 echo pin is conected to D3 of nodemcu
#define trig3 D5        // defining ultrasonic3 trig pin is conected to D5 of nodemcu
#define echo3 D4        // defining ultrasonic3 echo pin is conected to D4 of nodemcu
float tim1=0;           // denoting a variable to store the value of time and making it 0
float distance1=0;      // denoting a variable to store the value of distance and making it 0
float tim2=0;          // denoting a variable to store the value of time and making it 0
float distance2=0;     // denoting a variable to store the value of distance and making it 0
float tim3=0;         // denoting a variable to store the value of time and making it 0
float distance3=0;     // denoting a variable to store the value of distance and making it 0

   char auth[] = "1swyZkxQfhM2AE6iU9jz5x0vU8_gA7ha"; // Auth token for connection of the
                                                                 controller to the blynk app
char ssid[] = "Blynk";    // SSID name for the connection of the controller to wifi
char pass[] = "12345678";    //pass of ssid for the connection of the controller to wifi
WidgetBridge bridge1(V2);    // connecting widget bridge to virtual pin no V2
BlynkTimer timer;                // initialization of blynk timer
BLYNK_CONNECTED()            // command for starting the connection of bridge
{
   bridge1.setAuthToken("MmLGdyarNDgtkLFkVsbKheZocXO3kH-l"); // token of device which
                                                                      is to be connected
}

/* void blynkAnotherDevice()
{
if(distance1>=10&&distance2<=10)
{
bridge1.digitalWrite(D1,HIGH);
delay(100);
}
else
{
bridge1.digitalWrite(D1,LOW);
delay(100);
}
if(distance2>=10&&distance1<=10)
{
bridge1.digitalWrite(D2,HIGH);
delay(100);
}
else
{
```

```
bridge1.digitalWrite(D2,LOW);
delay(100);
}
if(distance1<=10&&distance2<=10)
{bridge1.digitalWrite(D1,HIGH);
bridge1.digitalWrite(D2,HIGH);
delay(100);
}
else
{
bridge1.digitalWrite(D1,LOW);
bridge1.digitalWrite(D2,LOW);
delay(100);
}
if(distance3<=40)
{
bridge1.digitalWrite(D3,HIGH);
delay(100);
}
else
{
bridge1.digitalWrite(D3,LOW);
delay(100);
}
}
*/    [whole is comment from /* to*/]

void blynkAnotherDevice()    // making a void function
{
 if(distance2<=20)                          //measuring the distance value of ultrasonic for detection of obstacle
{
         bridge1.digitalWrite(D1,HIGH);      // if obstacle detected then making value of other controller high on pin D1
delay(100);       // wait for 100 millisecond
}
         else if(distance1<=20)    //measuring the distance value of ultrasonic for detection of obstacle
{
  bridge1.digitalWrite(D2,HIGH); // if obstacle detected then making value of other controller high on pin D2
delay(100);    // wait for 100 millisecond
}
  else if(distance1<=10&&distance2<=10)      //measuring the distance value of ultrasonic for detection of obstacle
{
 bridge1.digitalWrite(D1,HIGH); // if obstacle detected then making value of other controller high on pin D1 and D2
bridge1.digitalWrite(D2,HIGH);
delay(100) ;       // Wait for 100 millisecond
}
```

```
else if(distance3>=60)    //measuring the distance value of ultrasonic for detection of
obstacle
{
 bridge1.digitalWrite(D3,HIGH);   // if obstacle detected then making value of other co
                                                          ntroller high on pin D1 and D2
delay(100); // Wait for 100 millisecond
}
    else          // if none of the above conditions are true then making the value of D1,D2,D3
                                                                   low of the other controller
{
bridge1.digitalWrite(D1,LOW);
bridge1.digitalWrite(D2,LOW);
bridge1.digitalWrite(D3,LOW);
delay(100);
}
}

void setup() // void setup for defining the mode of the connected pins
{
pinMode(trig1,OUTPUT);    // defining trigger pins as output pins of ultrasonic sensor
pinMode(echo1,INPUT);      // defining echo pins as input pins of ultrasonic sensors
pinMode(trig2,OUTPUT);
pinMode(echo2,INPUT);
pinMode(trig3,OUTPUT);
pinMode(echo3,INPUT);
Serial.begin(9600);        // defining baudrate speed for serial communication as 9600 bits/sec
 Blynk.begin(auth, ssid,pass );              // start the connection of the blynk using credentials
                                                                         mentioned above
timer.setInterval(1000L,blynkAnotherDevice);    // start the timer of the blynk
}

void loop()    // void loop for running the program again and again
{
Blynk.run();    // Start the blynk process
ultra1();          // calling ultrasonic codes for executing from below
ultra2();
ultra3();
timer.run();    // timer started
}

void ultra1()    // code for ultrasonic sensor
{
digitalWrite(trig1,LOW);    // making trigger pin low
delayMicroseconds(2);      // wait 2us
digitalWrite(trig1,HIGH);    // making trigger pin high
delayMicroseconds(10);    // wait 2us
digitalWrite(trig1,LOW);    //making trigger pin low
delayMicroseconds(2);      // wait 2us this generates the pulses on the trigger pin
tim1=pulseIn(echo1,HIGH);    // counting the pulse and storing in variable
distance1=tim1*0.0340/2;    // calculating the distance by formula and storing it
}
```

```
void ultra2() //    same    as ultra 1
{
digitalWrite(trig2,LOW);
delayMicroseconds(2);
digitalWrite(trig2,HIGH);
delayMicroseconds(10);
digitalWrite(trig2,LOW);
delayMicroseconds(2);
tim2=pulseIn(echo2,HIGH);
distance2=tim2*0.0340/2;
}
void ultra3() // same as ultra1
{
digitalWrite(trig3,LOW);
delayMicroseconds(2);
digitalWrite(trig3,HIGH);
delayMicroseconds(10);
digitalWrite(trig3,LOW);
delayMicroseconds(2);
tim3=pulseIn(echo3,HIGH);
distance3=tim3*0.0340/2;
}
```

Code: Receiver Section / Remote section

Controller :ESP8266 NodeMcu (WiFi Module)
{ Note : Similar lines are note commented again }

```
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#define B D0 // defining the B as button connected on pin no D0
int val=0;      // defining a variable int to store the value

char auth[] = "MmLGdyarNDgtkLFkVsbKheZocXO3kH-l";
char ssid[] = "Blynk";
char pass[] = "12345678";
String latitude;    // defining string to store the value of latitude
String longitude; // defining string to store the value of longitude

void setup()
{
Serial.begin(9600);
pinMode(D1,INPUT);
pinMode(D2,INPUT);
pinMode(D3,INPUT);
PinMode(B,INPUT);
Blynk.begin(auth,ssid,pass);
}
BLYNK_WRITE(V6)    // connection of GPS widget on virtual pin no V6
{
```

```
GpsParam gps(param);    // initialising the GPS
 Blynk.virtualWrite(V6,  param[0].asString());  //  Making  a  pointer  [0]  to  store  data  on  s
                                                                           ame  pin  V6  of  GPS
 Blynk.virtualWrite(V6,  param[1].asString());  //  Making  a  pointer  [1]  to  store  data  on  s
                                                                           ame  pin  V6  of  GPS
latitude = param[0].asString();    // store the value of pointer[0] in variable
longitude = param[1].asString(); // store the value of pointer[1] in variable
// Print 6 decimal places for Lat, Lon
Serial.print("Lat: "); // serial print all the data on serial monitor
Serial.println(gps.getLat(), 7);
Serial.print("Lon: ");
Serial.println(gps.getLon(), 7);
}

void loop()
{
Blynk.run();
If(B==HIGH) // if condition is true then enter the loop
{
    if(digitalRead(D1)==HIGH && digitalRead(D2) == LOW)    // if D1 and D2 is high then enter
                                                                                     the loop
{
Serial.write(1); // serial writing 1 value for nano controller for text to speech conversion
Serial.println("Walk Ahead"); // print data on serial monitor
delay(100);
}
  else if(digitalRead(D2)==HIGH && digitalRead(D1) == LOW) // if condition is true then enter
                                                                                     the loop

{
Serial.write(2); // serial writing 2 value for nano controller for text to speech conversion
Serial.println("Turn Left");    // print data on serial monitor
delay(100);
}
else if(digitalRead(D3)==HIGH) // if condition is true then enter the loop

{
Serial.write(3); // serial writing 3 value for nano controller for text to speech conversion
Serial.println("Slope Ahead");    // print data on serial monitor
delay(100);
}
else if((digitalRead(D1)==HIGH && digitalRead(D2) == HIGH) || (digitalRead(D2)==HIGH &&
digitalRead(D1) == HIGH) )    // if condition is true then enter the loop

{
Serial.write(4); // serial writing 4 value for nano controller for text to speech conversion
Serial.println("Stop"); // print data on serial monitor
delay(100);
}
else // if none of the above condition is true then enter the loop
{
```

```
Serial.write(5); // serial writing 5 value for nano controller for text to speech conversion
Serial.println("All ok");    // print data on serial monitor
delay(100);
}
}
else
{
Serial.print("Help");    // print data on serial monitor
String toSend;    // defining string to store the value
toSend += " HELP !!!!! ";
toSend += ". View location on Maps: ";
toSend += "http://maps.google.com/?q=";
toSend += latitude;
toSend += ",";
toSend += longitude;    // writing the values in the string
 Blynk.email("kamblimitali9@gmail.com", "Subject: Smart Shoe", toSend);            // send mail command
}
}
```

Code: Receiver Section /Remote Section

Controller :Arduino Nano

```
#include "Talkie.h" //Header file for text to speech Module
#include "Vocab_US_Large.h" //Header file for US accent speech
#include "Vocab_Special.h"    // Header file for tts library
#include "Vocab_Soundbites.h"
int value1=0;    // defining the value of variable int to store value
Talkie voice;    // initialising tts module
void setup()
{
Serial.begin(9600);
}
void loop()
{
if(Serial.available()>0) // if serial data is available then enter the loop
{
value1=Serial.read(); // read the incoming value
if(value1 ==1) // if condition is true then enter the loop
{
// Serial.println("Walk Ahead");
voice.say(sp2_NORTH); // spell the written word (sp2,sp3,sp4 are the speed of saying word)
}
else if(value1 ==2)    // if condition is true then enter the loop
{
//Serial.println("Turn Left");
voice.say(sp2_LEFT); // spell the written word
}
else if(value1 ==3)    // if condition is true then enter the loop
{
```

```
// Serial.println("Slope Ahead");
voice.say(sp2_ALERT);    // spell the written word
}
else if(value1 ==4) // if condition is true then enter the loop
{
//Serial.println("Stop");
voice.say(sp2_STOP); // spell the written word
}
else if(value1 ==5) // if condition is true then enter the loop
{
//Serial.println("All ok");
voice.say(sp2_GO); // spell the written word
}
}
}Note : So
```