VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

GESTURE BASED DOORBELL SYSTEM - ADVANCED MICROCONTROLLERS ECE4002, REVIEW - III

18BIS0043 – Allen Ben Philipose
18BIS0057 – Mutha Aakar Jayesh
18BIS0086 – D Chandini

Submitted to Prof. Vidhyapathi C.M.

INDEX

This project report, titled **Gesture Based Doorbell System** is submitted to **Prof. Vidhyapathi CM,** Assistant Professor Sr., School of Electronics Engineering, VIT, after completing the requirements of **Review-3,** in the J-component of **ECE4002: Advanced Microcontrollers** conducted during **slots: B2+TB2**

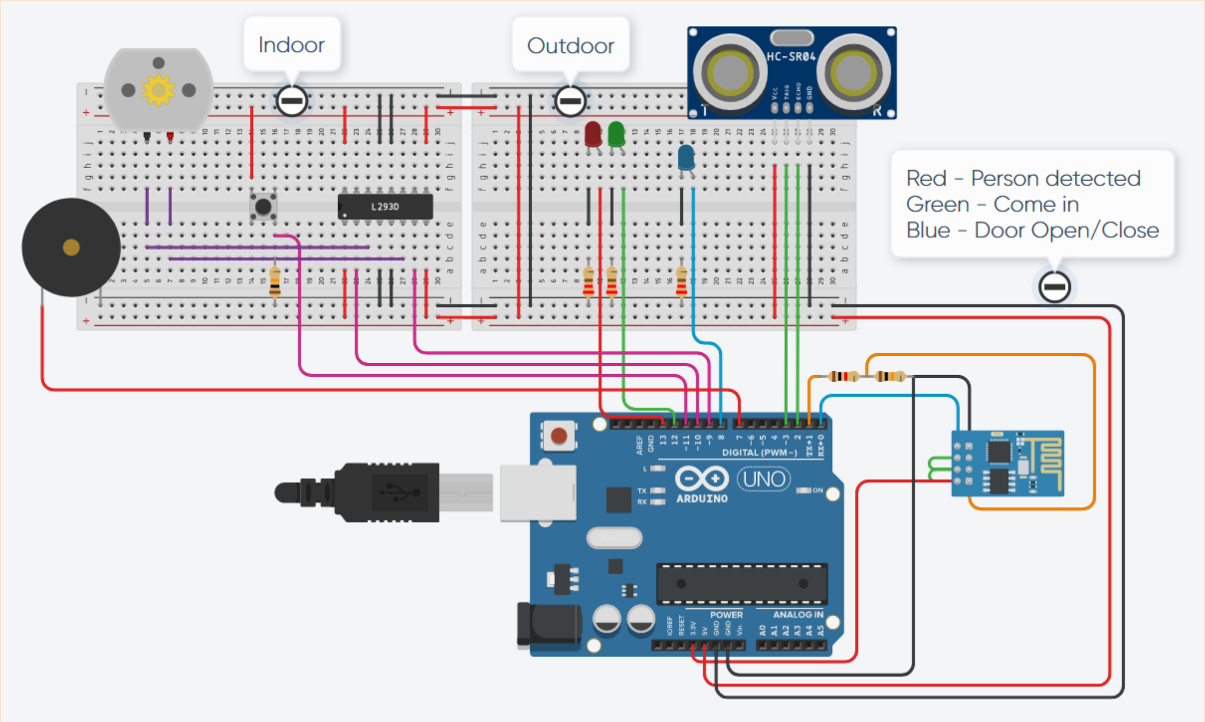
AIM

Coronavirus has prompted endless deaths and long periods of lockdown. Yet, the main thing that can stop this is social distancing and cleanliness. For this, we must touch regular surfaces as less as possible to avoid danger. One such regular surface is our doorbell switch which comes in contact by numerous individuals and consequently can be a shared belief for infection to spread. Let us replace this doorbell with our **GESTURE BASED DOORBELL SYSTEM** to keep corona out of our way.

ABSTRACT

To avoid unnecessary interaction between human with his surroundings we have designed a GESTURE BASED DOORBELL SYSTEM which senses the presence of a human using an ultrasonic sensor. when the presence of a person is sensed, the doorbell rings automatically using a relay module. This will help us stay safe during this pandemic by reducing the need to touch objects near you.

CIRCUIT DIAGRAM



TOOLS

1. Arduino Uno

The Arduino Uno is an open-source microcontroller board based on the Microchip ATmega328P microcontroller and developed by Arduino. The board is equipped with sets of digital and analogue input/output pins that may be interfaced to various expansion boards and other circuits.

2. Ultrasonic sensor

Ultrasonic sensor is an electronic system that operates by emitting sound waves at a frequency that is too high for humans to detect. It then waits for the sound to be expressed in the measurement of the distance depending on the time required to reflect the wave back. Here we use the Hc-Sr04 Ultrasonic Sensor Module.

3. Relay module

The relay is a switch that is electrically operated by an electromagnet. The electromagnet is triggered with a low voltage, e.g. 5 volts from a microcontroller, and pulls a contact to make or break a high voltage circuit. Not added in this experiment.

4. Connecting wires

Connecting wires provides a medium to an electrical current that an allow it to flow from one point to another with less loss.

5. Resistors

A resistor is a passive two-terminal electrical component that implements electrical resistance as a circuit element.

6. Servo Motor

The servo motor is in integral part of our system. It will be used to open or close the door when we get a trigger from the microcontroller.

7. Servo Motor Driver

L293D IC is a typical Motor Driver IC which allows the DC motor to drive on any direction. This IC consists of 16-pins which are used to control a set of two DC motors instantaneously in any direction. It means, by using a L293D IC we can control two DC motors.

8. Wi-Fi Module ESP8266

The ESP8266 Wi-Fi Module is a self-contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to your Wi-Fi network. The ESP8266 is capable of either hosting an application or offloading all Wi-Fi networking functions from another application processor.

9. Push button

A push button switch is a small, sealed mechanism that completes an electric circuit when you press on it. When it's on, a small metal spring inside contacts two wires, allowing electricity to flow.

10. Buzzer

A buzzer or beeper is an audio signalling device, which may be mechanical, electromechanical, or piezoelectric.

11. LED indicators

A light-emitting diode (LED) is a semiconductor light source that emits light when current flows through it.

12. TinkerCad

Tinkercad is a free, online 3D modelling program that runs in a web browser, known for its simplicity and ease of use.

13. Arduino IDE

The Arduino Integrated Development Environment is a cross-platform application that is written in functions from C and C++.

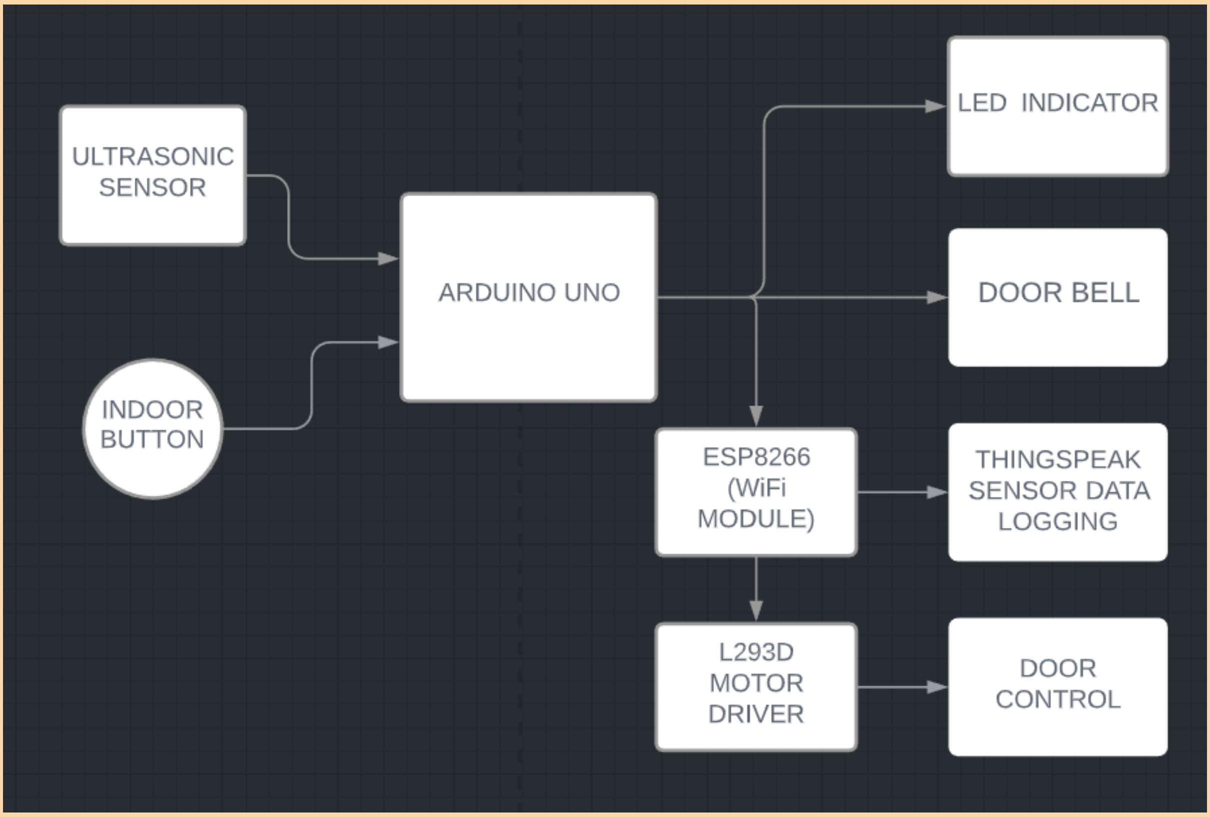
14. ThingSpeak

ThingSpeak is an open-source Internet of Things application and API to store and retrieve data from things using the HTTP and MQTT protocol over the Internet or via a Local Area Network.

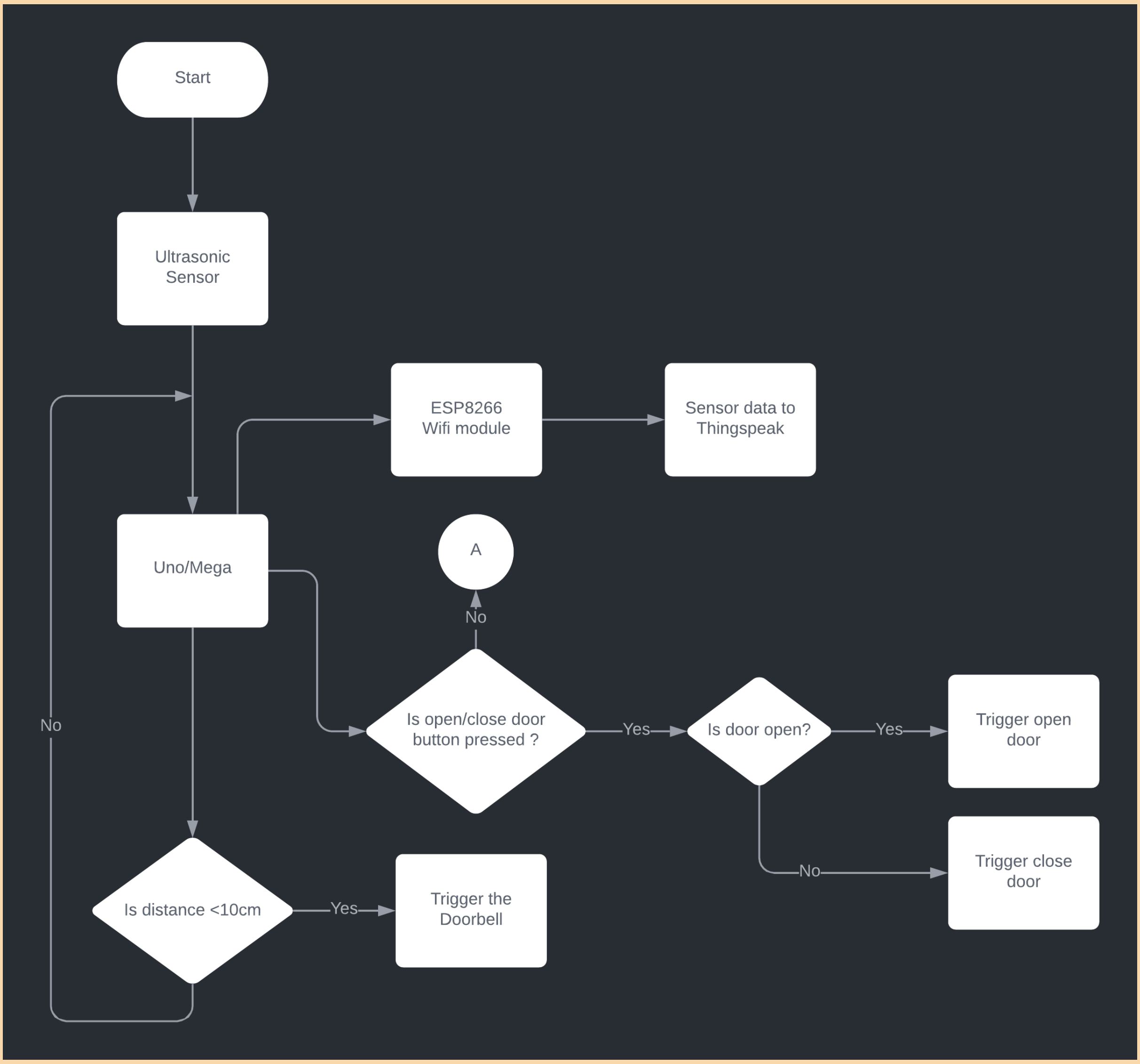
METHODOLOGY

An ultrasonic sensor is attached near the entrance, this will help us to detect people around it. When ultrasonic sensor detects someone, it sends a signal to Arduino which is then processed and gives an indication to the owner. Owner has a button throw which he can open the door. As soon as the button is switched on a buzzer rings and green LED is turned on. The door gets open using the motor and the guest gets in. After that the door gets closed. In this duration blue LED will glow. After the door shuts down red LED will glow again.

BLOCK DIAGRAM



FLOW CHART


Start
Ultrasonic Sensor
ESP8266 Wifi module
Sensor data to Thingspeak
Uno/Mega
A
No
No
Is open/close door button pressed ?
Yes
Is door open?
Yes
Trigger open door
No
Trigger close door
Is distance <10cm
Yes
Trigger the Doorbell

CODE - ARDUINO

```
String ssid     = "Simulator Wifi";
String password = "";
String host     = "api.thingspeak.com";
const int httpPort   = 80;
String     uri                           =
"/update?api_key=RDRD4KQJS6ECPD7V&field1=
";
int Distance = 0;
int Door = 0; // Closed Initially
long           readUltrasonicDistance(int
triggerPin, int echoPin)
{
  pinMode(triggerPin, OUTPUT);
  digitalWrite(triggerPin, LOW);
  delayMicroseconds(2);
  digitalWrite(triggerPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(triggerPin, LOW);
  pinMode(echoPin, INPUT);
  return pulseIn(echoPin, HIGH);
}

int setupESP8266(void) {
  Serial.begin(115200);
  Serial.println("AT");
  delay(10);
  if (!Serial.find("OK")) return 1;
  Serial.println("AT+CWJAP=\""  +  ssid  +
"\",\"" + password + "\"");
  delay(10);
  if (!Serial.find("OK")) return 2;
  Serial.println("AT+CIPSTART=\"TCP\",\""
+ host + "\"," + httpPort);
  delay(50);
  if (!Serial.find("OK")) return 3;
  return 0;
}
```

```
void anydata(int t1, int t2) {
  int temp1 = map(t1,0,1000,0,1000);
  int temp2 = map(t2,0,1000,0,1000);
  String httpPacket = "GET " + uri +
String(temp1) + "&field2=" + String(temp2)
+"&amp" + " HTTP/1.1\r\nHost: " + host +
"\r\n\r\n";
  int length = httpPacket.length();
  Serial.print("AT+CIPSEND=");
  Serial.println(length);
  delay(10);
  Serial.print(httpPacket);
  delay(10);
  if (!Serial.find("SEND OK\r\n")) return;
}
void ring(){
  tone(7, 1000, 200);
  delay(100);
  noTone(7);
}
```

```
void blink(){
  for(int i=0;i<5;i++){
    digitalWrite(8, HIGH);
    delay(200);
    digitalWrite(8, LOW);
    delay(200);
  }
}

void setup() {
  pinMode(13, OUTPUT);
  pinMode(12, OUTPUT);
  pinMode(11, INPUT);
  pinMode(10, INPUT);
  pinMode(9, INPUT);
  pinMode(8, OUTPUT);
  setupESP8266();
}
```

```
void loop() {
  Distance = 0.01723 *
readUltrasonicDistance(3, 2);
  int m = 0;
  if (Distance >= 100) {
    digitalWrite(13, LOW);
    digitalWrite(12, LOW);
    m=0;
    if (Door==1) {
       digitalWrite(13, HIGH);
        digitalWrite(10, HIGH);
        digitalWrite(9, LOW);
        blink();
        digitalWrite(13, HIGH);
        digitalWrite(10, LOW);
        Door = 0;
    }
  }
  else {
    if (Door==0){
      digitalWrite(12, LOW);
      digitalWrite(13, HIGH);
      ring();
    }
    else if (Door==1){
      digitalWrite(12, HIGH);
      digitalWrite(13, LOW);
    }
    m=1;
    // 11 is the close/open button
    if(digitalRead(11)==HIGH) {
      if (Door==0) {
        digitalWrite(12, HIGH);
        digitalWrite(13, LOW);
        digitalWrite(9, HIGH);
        digitalWrite(10, LOW);
        blink();
        digitalWrite(9, LOW);
        Door = 1;
      }
      else if (Door==1) {
        digitalWrite(13, HIGH);
        digitalWrite(12, LOW);
        digitalWrite(10, HIGH);
        digitalWrite(9, LOW);
        blink();
        digitalWrite(10, LOW);
        Door = 0;
      }
    }
  }
  anydata(m, Door);
}
```

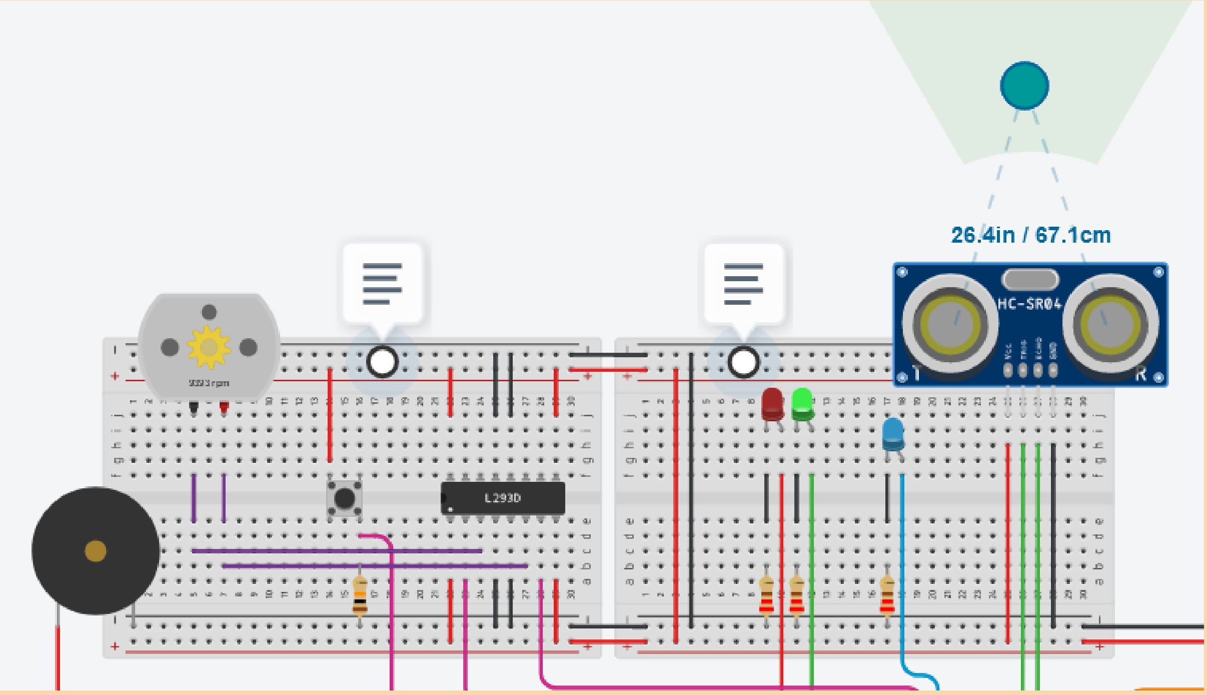
CODE – ATMEGA32 + ULTRASONIC SENSOR

```
#include <avr/io.h>
#include <avr/interrupt.h>
#include <util/delay.h>
#include <string.h>
#include <stdlib.h>

int TimerOverflow = 0;

ISR(TIMER1_OVF_vect)
{
    TimerOverflow++;
}
int main(void)
{
    char string[10];
    long count;
    double distance;
    DDRA = (1<<PA0);
    PORTD = (1<<PD6);
    sei();
    TIMSK = (1 << TOIE1);
/* Enable Timer1 overflow interrupts */
    TCCR1A = 0;      /* Set all bit to zero
Normal operation */

    while(1)
    {
        /* Give 10us trigger pulse on trig.
pin to HC-SR04 */
        PORTA |= (1 << PA0);
        _delay_us(10);
        PORTA &= (~(1 << PA0));

        TCNT1 = 0;/* Clear Timer counter */
        TCCR1B   =   (1<<ICES1)||(1<<CS10);
    /* Capture on rising edge, No scaler*/
        TIFR = 1<<ICF1;
        TIFR = 1<<TOV1;
        /*Calculate width of Echo by Input
Capture (ICP) */
        while ((TIFR & (1 << ICF1)) == 0);/*
Wait for rising edge */
        TCNT1 = 0;/* Clear Timer counter */
        TCCR1B = 0x01;   /*   Capture   on
falling edge, No scaler */
        TIFR = 1<<ICF1;  /* Clear ICP flag
(Input Capture flag) */
        TIFR = 1<<TOV1;  /*  Clear  Timer
Overflow flag */
        TimerOverflow = 0;/* Clear Timer
overflow count */


        while ((TIFR & (1 << ICF1)) == 0);/*
Wait for falling edge */
        count  =  ICR1  +  (65535  *
TimerOverflow);  /* Take count */
        /* 8MHz Timer freq, sound speed =343
m/s */
        distance = (double)count / 466.47;
        _delay_ms(200);
    }
}
```

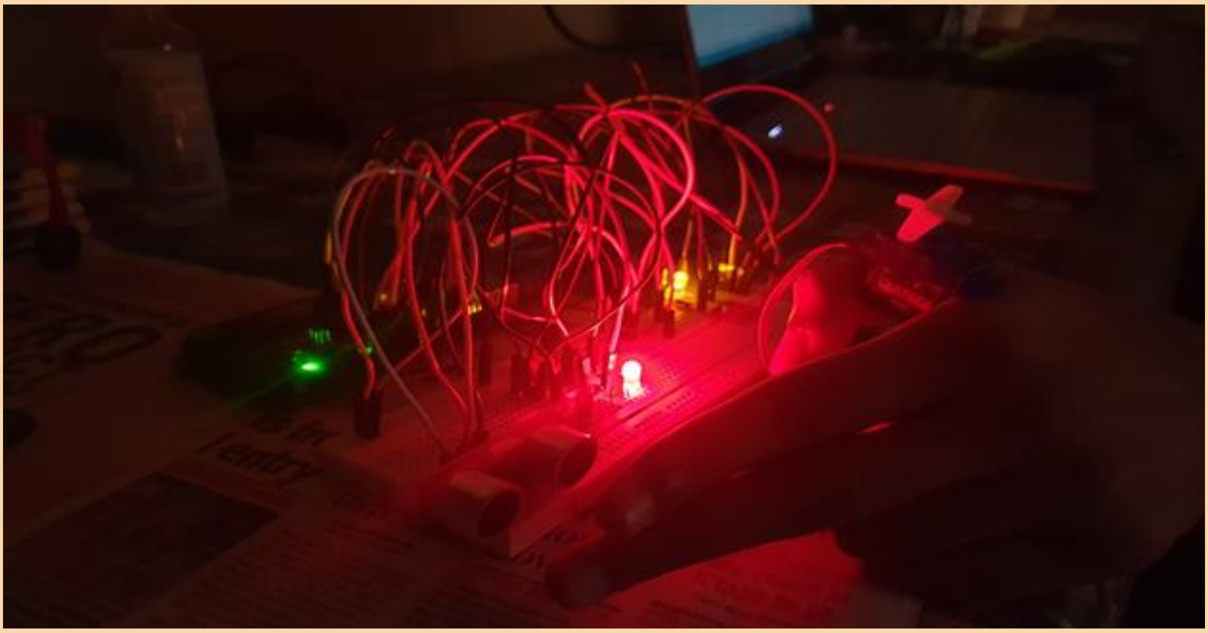
CODE – ATMEGA32 + MOTOR

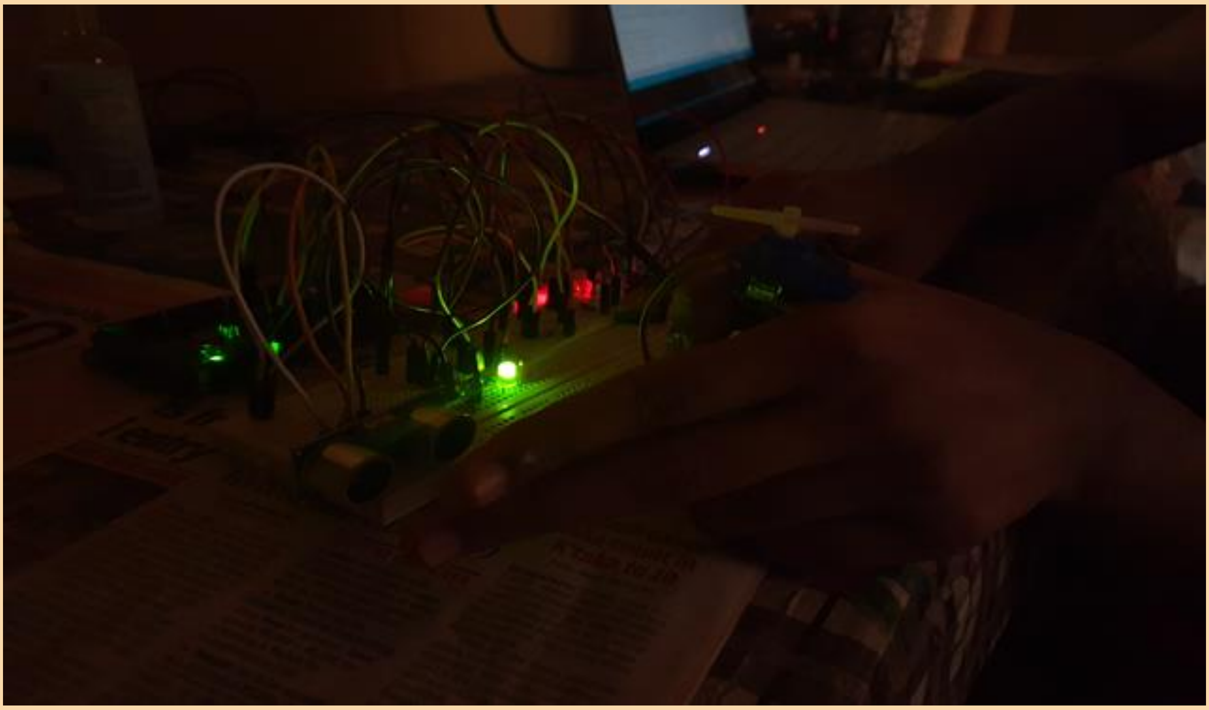
```
#include <avr/io.h>
#include <util/delay.h>
int main(void)
{
   DDRA = (1<<PA0)|(1<<PA1);
   DDRB = 0x00;
    while (1) {
       if(PINB & 1<<PB0){
          PORTA = (1<<PA0);
          _delay_ms(5000);
          PORTA = 0x00;
        }
        else if(PINB & 1<<PB1){
           PORTA = (1<<PA1);
           _delay_ms(5000);
           PORTA = 0x00;
        }
    }
}
```

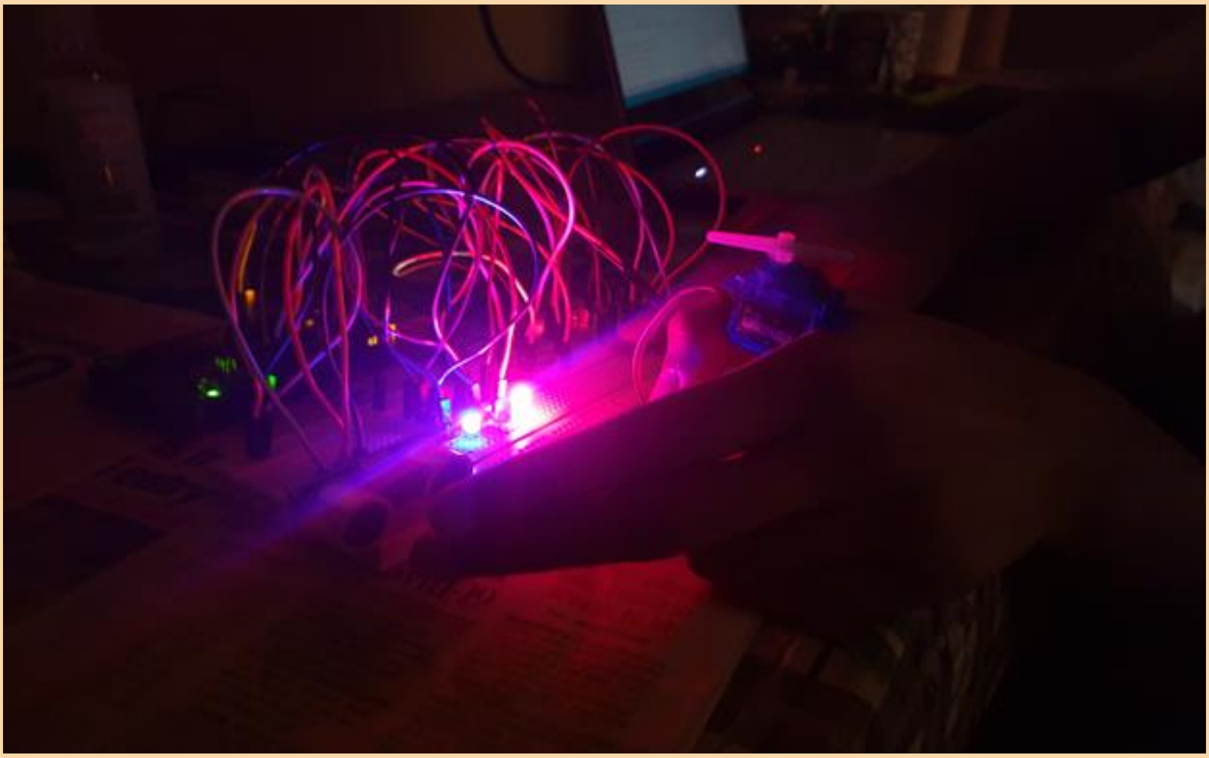
OUTPUT – TINKERCAD



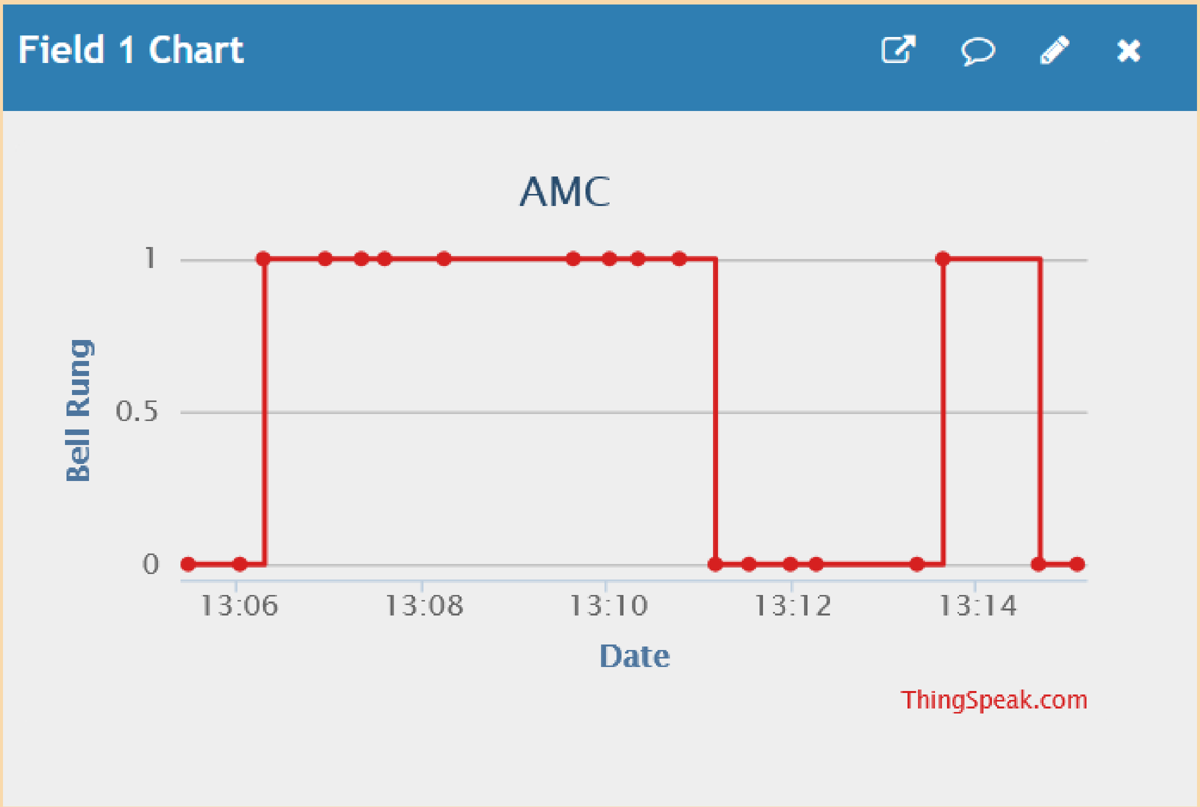
OUTPUT – HARDWARE

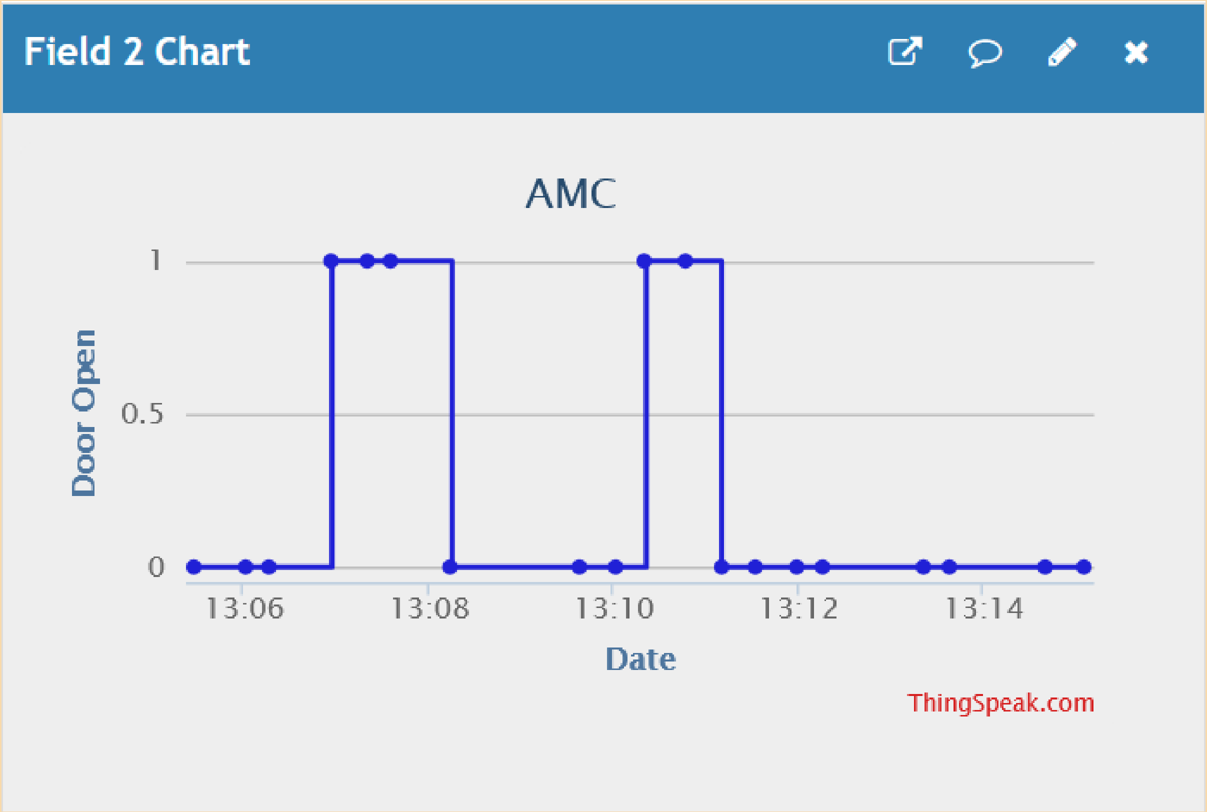






OUTPUT – THINGSPEAK





OBSERVATIONS

Whenever someone will be within the predefined distance of the ultrasonic sensor, the owner is alerted, either by a bell or by light indicators and the door system [motor] can be triggered using a push button which will stay open till the ultrasonic sensor value is cleared or until it is manually overridden.

CONCLUSIONS

Hence by using Arduino we have simulated a gesture-based doorbell system using ultrasonic sensor to detect human presence, a button for owner to open the door, LEDs for indication and L293D IC and relay module to open and close the door.

FUTURE EXPANSION

This project can be applied directly and easily, in a real-world scenario, which replaces our common doorbell and by having a strong motor, capable enough to move the door. So, this can make our task easier and to keep infections away.

CONTRIBUTION & OUR WORK

- https://prezi.com/view/gnjJOdP4ttBT2l4LVuV3/
- https://drive.google.com/drive/folders/1uiXENMvr9m6iH6oJX00r9sITFM4loqlg?usp=sharing

1. Allen Ben Philipose (18BIS0043)
   i) Hardware implementation
   ii) ThingSpeak integration with Arduino code using ESP
   iii) Prezi presentation

2. Mutha Aakar Jayesh (18BIS0057)
   i) ATMEGA coding
   ii) Arduino coding for materials
   iii) PowerPoint presentation

3. Dommaraju Chandini (18BIS0086)
   i) Literature Survey
   ii) Logical analysis of the program
   iii) Final Report