IoT Based Warehouse Security System

Pragadeesh S P
School of Electronics Engineering
Vellore Institute of Technology, Vellore, India
pragadeesh.sp2020@vitstudent.ac.in

Yukesh P
School of Electronics Engineering
Vellore Institute of Technology, Vellore, India
yukesh.p2020@vitstudent.ac.in

Sakkthi Praanesh S M
School of Electronics Engineering
Vellore Institute of Technology, Vellore, India
sakkthipraanesh.sm2020@vitstudent.ac.in

Abstract - **The main aim of the project is to design and build a Warehouse Security System using ARDUINO, PIR sensor and temperature sensor so that they can identify unidentified movements and also recognise and record the temperature inside the warehouse.**

I. INTRODUCTION

Safety remains a critical issue not only for houses in cities but also for warehouses to protect the goods stored. The security system based on the IoT platform has the potential of interacting real-time with the device. A proven strategy to improve and make sure security system and monitoring warehouse environment remotely is needed. This study design and develop an integrated security and monitoring system using Internet of Things (IoT) by combining Arduino Uno and ESP8266 Wi-Fi module.

The IoT based security system helps in added protection of the user/customer property. Security systems are designed to perform certain tasks when a secured zone is breached. In this paper, notification is sent to the concerned person as an alert where the user can take necessary actions. The main advantage of this system is the ability to remotely manage one's property 24/7.The monitoring system used PIR sensor to detect the intruder, LM35 sensor to detect the room temperature. We also included LED and Buzzer for indication. Using of cloud storages like Thingspeak and the software to trigger a reaction for an action like IFTTT were used to store, analyse and keep a record of the security system in the warehouse

II. METHODOLOGY

The main objective of the project is to

i. To detect unidentified movements inside a warehouse using PIR sensor.
ii. To also ensure maintaining appropriate temperature inside the warehouse using temperature sensor.
iii. Display current temperature and anonymous movements on a LCD screen for monitoring.

To design this system we need a basic block diagram (for both software and hardware) to accomplish our objective.

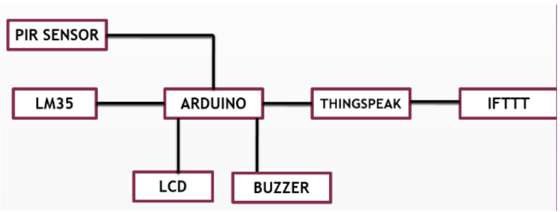

Fig. 1 Block Diagram for software

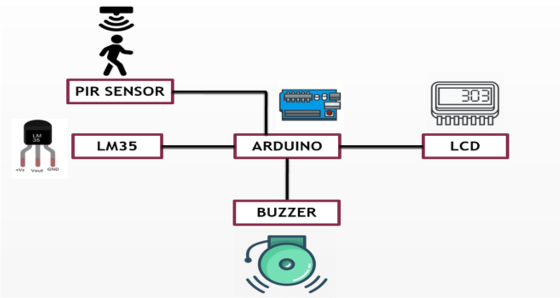

Fig. 2 Block Diagram for Hardware

III. COMPONENTS USED

A. PIR Sensor

This sensor is used so that it can identify the movements. PIR is made of a pyroelectric sensor, which is able to detect different levels of infrared radiation. The detector itself does not emit any energy but passively receives it. It uses as a sensor for converting human infrared radiation into electricity. If the human infrared radiation is directly irradiated on the detector, it will, of course, cause a temperature change to output a signal. This provides a digital output which will be sent to arduino.

Fig. 3 PIR Sensor

B. LM35

LM35 is a precession Integrated circuit Temperature sensor, whose output voltage varies, based on the temperature around it. It is a small and cheap IC which can be used to measure temperature anywhere between -55°C to 150°C. It can easily be interfaced with any Microcontroller that has ADC function or any development platform like Arduino. It is a temperature sensor that outputs an analog signal which is proportional to the instantaneous temperature. The output voltage can easily be interpreted to obtain a temperature reading in Celsius.

Fig. 4 LM35

C. Arduino

An Arduino is an open source electronic platform based on easy to use hardware and software. It's intended for anyone making interactive projects. It can sense the environment by receiving inputs from many sensors, and affects its surroundings by controlling lights, motors and other actuators.

Fig. 5 Arduino

D. Buzzer

A buzzer or beeper is an audio signalling device, which may be mechanical, electromechanical, or piezoelectric. Typical uses of buzzers and beepers include alarm devices, timers, and confirmation of user input such as a mouse click or keystroke.

Fig. 6 Buzzer

E. LCD

Liquid crystal display (LCD), electronic display device that operates by applying a varying electric voltage to a layer of liquid crystal, thereby inducing changes in its optical properties. In this project LCD is used so that we can see the changes in temperature at particular conditions.

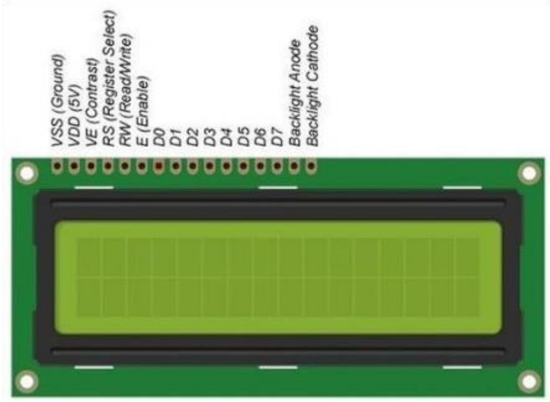

Fig. 7 LCD

F. ESP8266 Wi-Fi Module

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. Each ESP8266 module comes preprogrammed with an AT command set firmware, meaning, you can simply hook this up to your Arduino device and get about as much Wi-Fi ability as a Wi-Fi Shield offers (and that's just out of the box)! The ESP8266 module is an extremely cost effective board with a huge, and ever growing, community.

Fig. 8 SEP8266 Wi-Fi Module

G. Thingspeak
Thingspeak is an open-source software written in Ruby which allows users to communicate with internet enabled devices. It can collect the data. Analyse it and Act accordingly. It facilitates data access, retrieval and logging of data by providing an API to both the devices and social network websites.

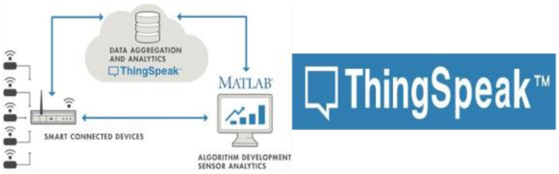

Fig. 9 Thingspeak

H. IFTTT
f This Then That is a private commercial company that runs services that allow a user to program a response to events in the world. IFTTT has partnerships with different service providers that supply event notifications to IFTTT and execute commands that implement the responses. Some event and command interfaces are simply public APIs. The programs, called applets, are simple and created graphically.

Fig. 10 IFTTT

IV. IMPLEMENTATION AND WORKING

At first, the sensors both PIR and temperature sensors are placed inside the warehouse. Both the sensors are connected to arduino to send data and receive instructions. Along with the sensors, an LCD, LED and buzzer are interfaced inside the warehouse. The temperature inside the warehouse must be continuously monitored for the safety of goods. PIR sensor inside the warehouse detects the unidentified motion of the human based on the infrared rays it receives. If there is any person inside the warehouse, it receives the rays and the received information is converted to digital data and will be transferred to the arduino stating of some motion inside the warehouse. This indicates that some one is inside the warehouse. As soon as there is detection of motion, the LED screen displays about the motion along with the buzzer beeping and LCD glowing. This makes the security alert.

The data received from the sensors by the arduino are sent to cloud for storage and analysis of the data. Thingspeak works as cloud storage for storing the data. Using ESP8266 Wi-Fi module, the data from arduino will be sent to thingspeak.

As soon as the alarm goes beeping because of some motion inside the warehouse, the use of IFTTT comes into action where it sends a mobile notification to the owner stating of the motion. This helps the owner to know about what’s going on inside the warehouse. We connect the IFTTT with thingspeak cloud using webhooks and ThingHTTP. This helps the owner to take action and can make the warehouse safe for storing the goods.

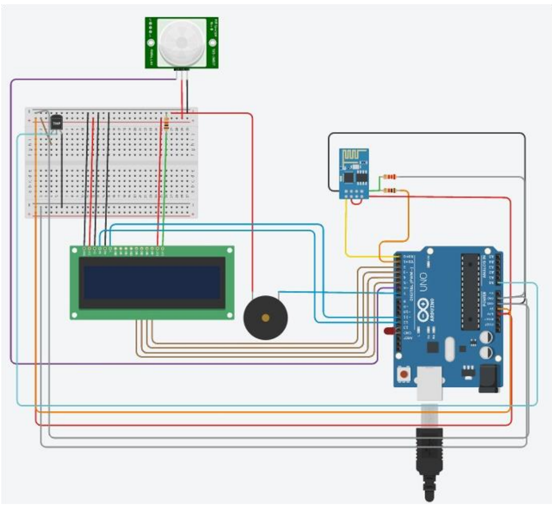
Fig. 11 Tinker Cad circuit design

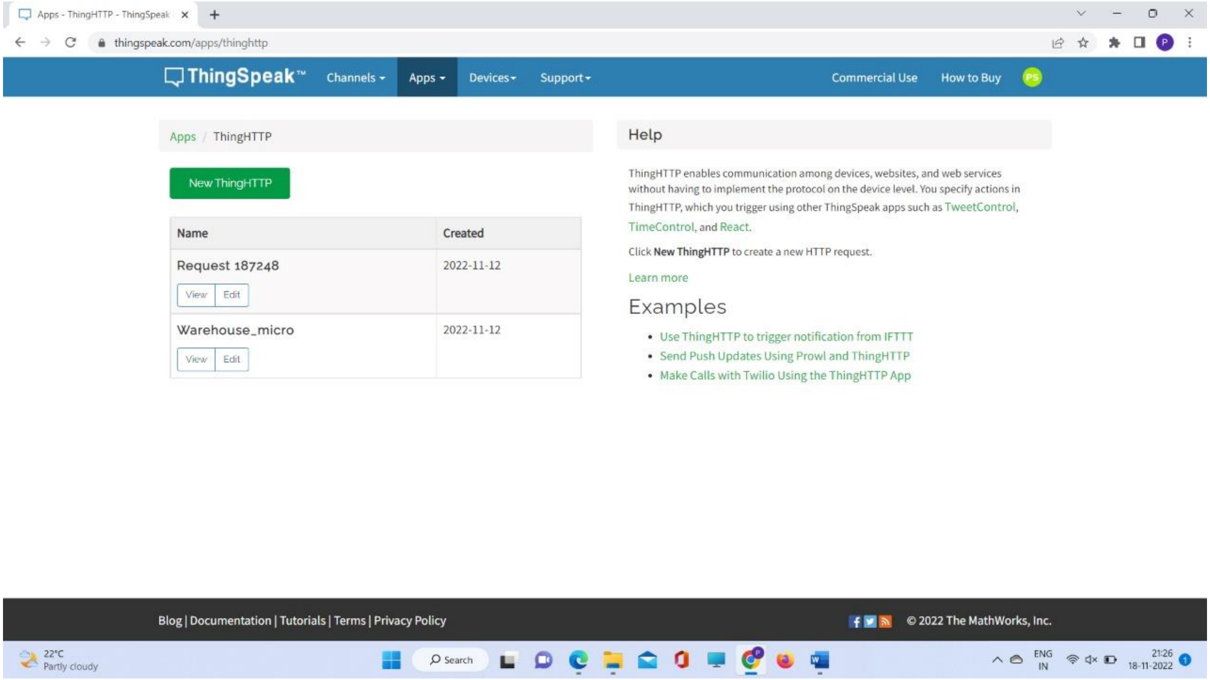

Fig. 12 New Thinkspeak Channel

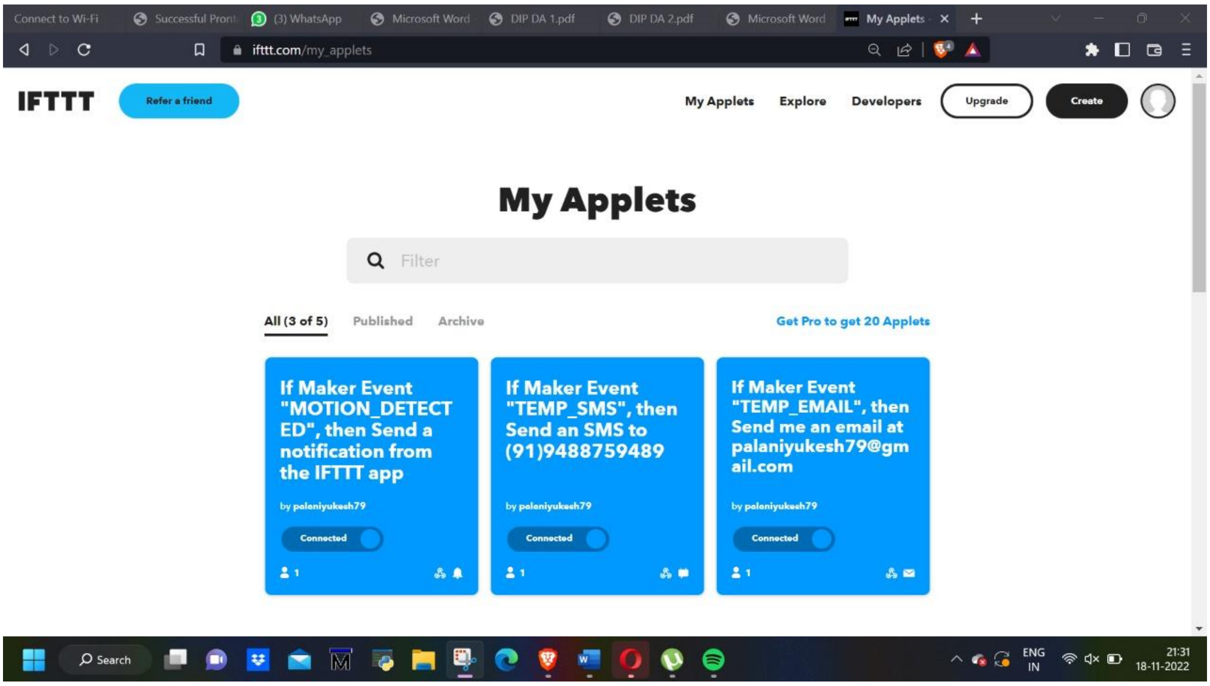

Fig. 13 IFFTT APPLETS

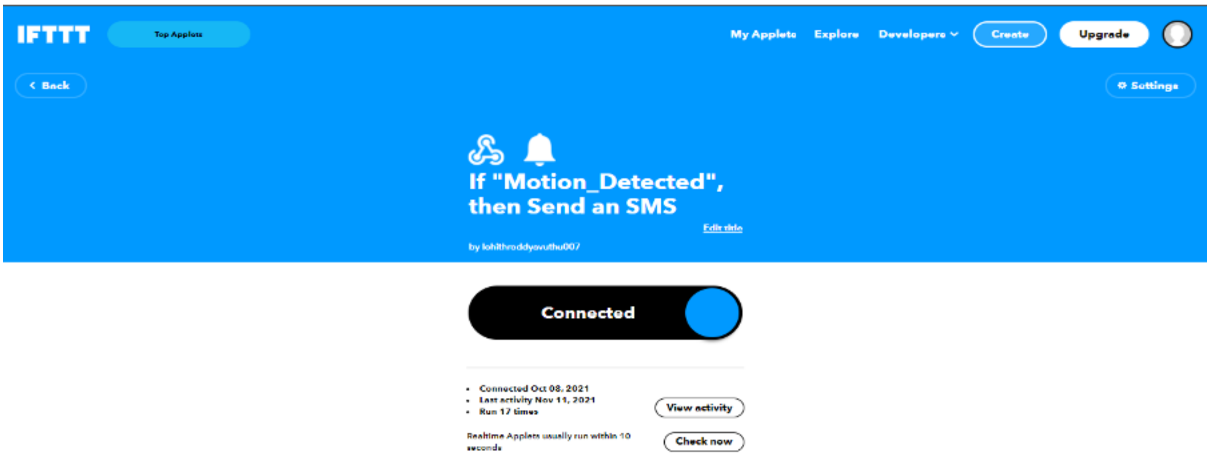

Fig. 14 Connections in IFTTT

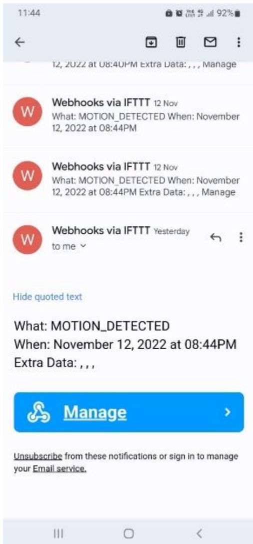

Fig. 15 Webhooks trigger

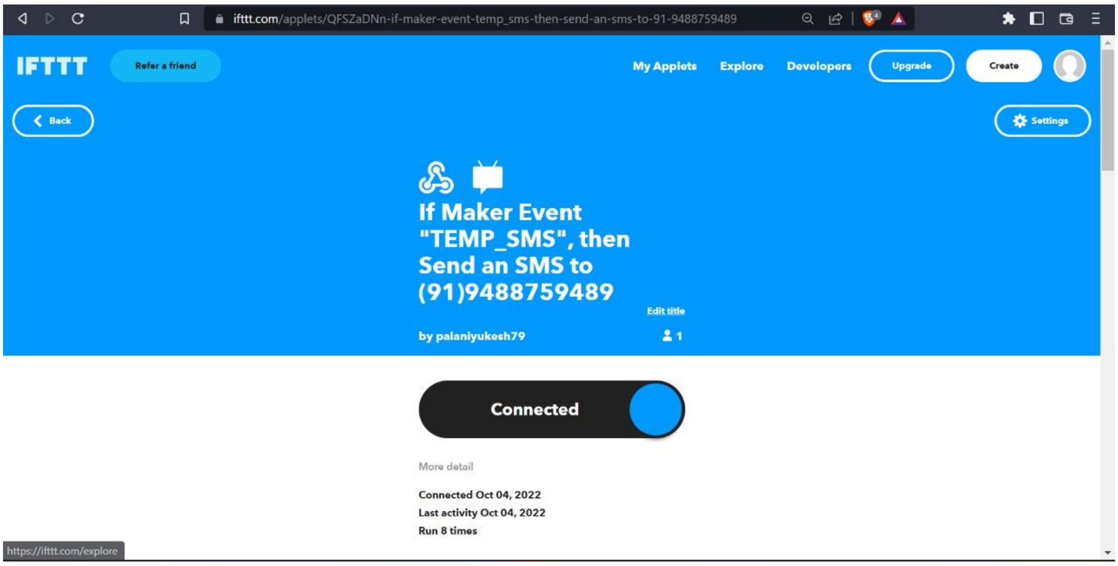

Fig. 15 Notification actions (SMS)

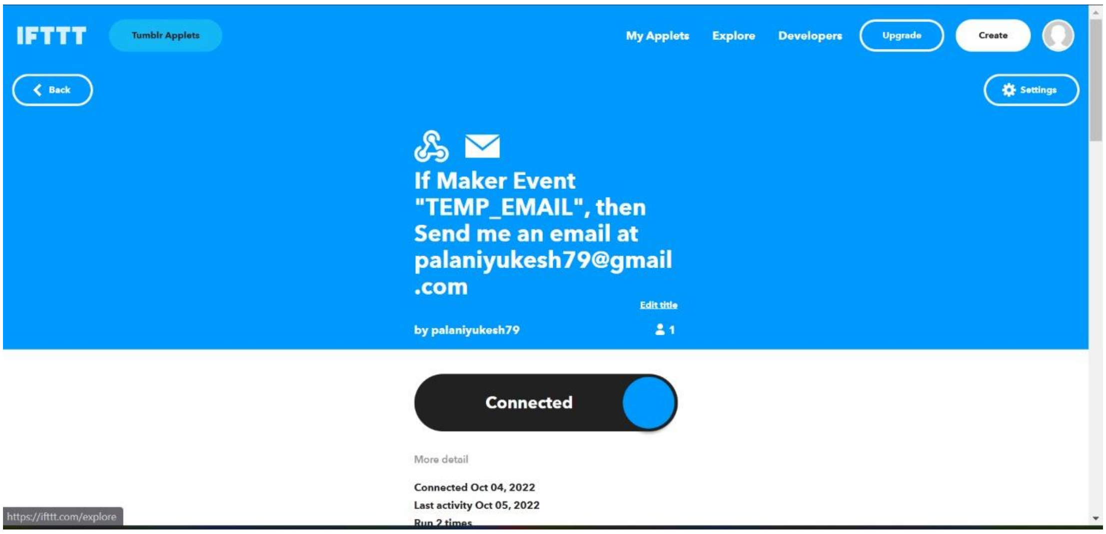

Fig. 16 Notification actions (Email)

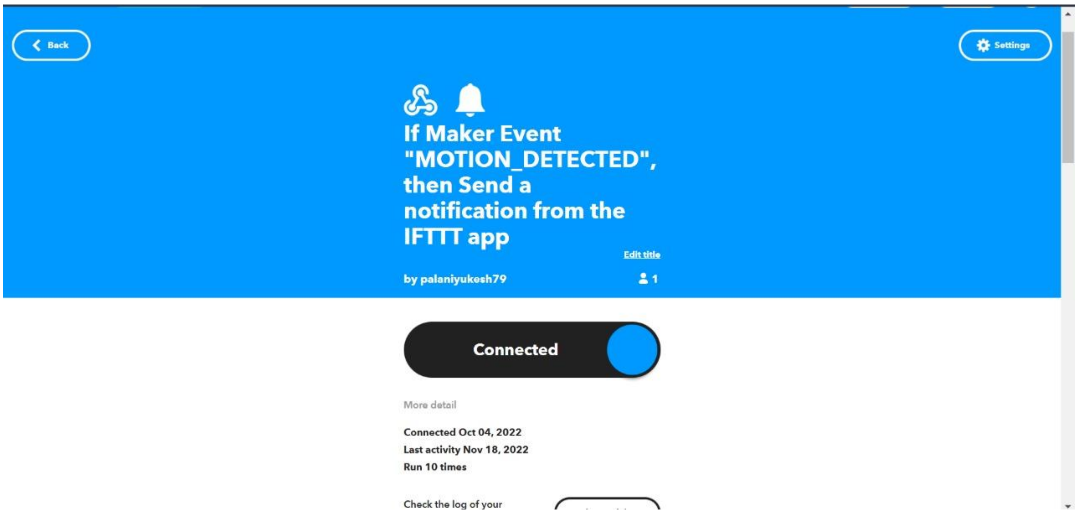

Fig. 17 Notification actions (Notification)

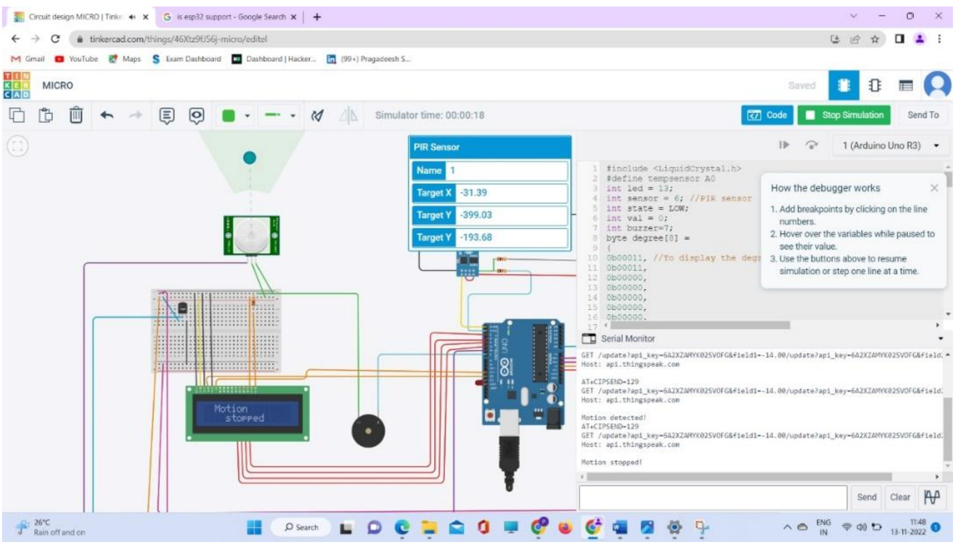

Fig. 18 Output: Giving minute to minute update to the Thingspeak and displaying the temperature continuously on LCD

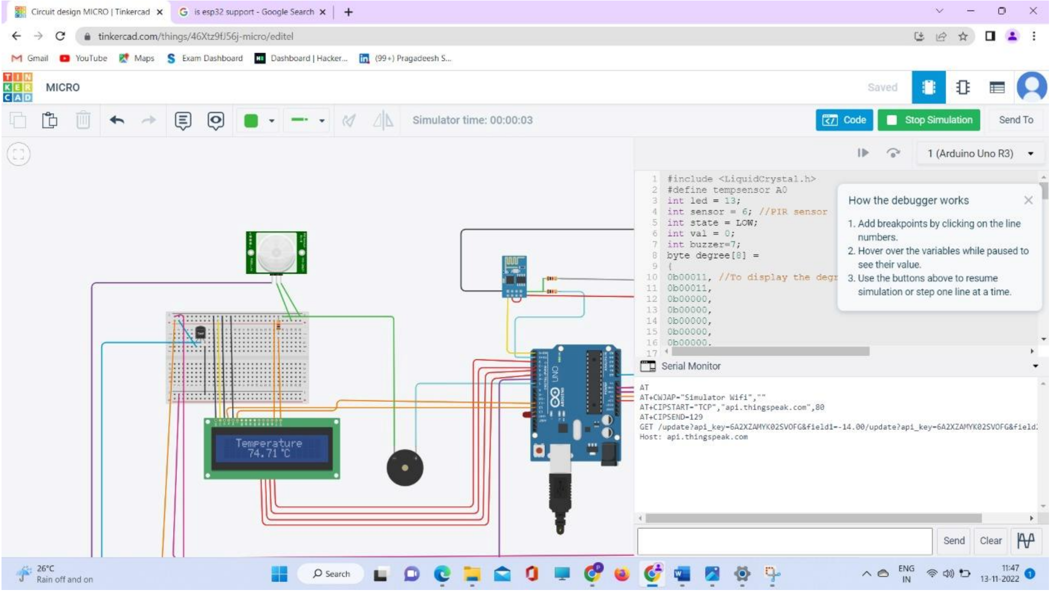

Fig. 19 Motion detection displayed on LCD

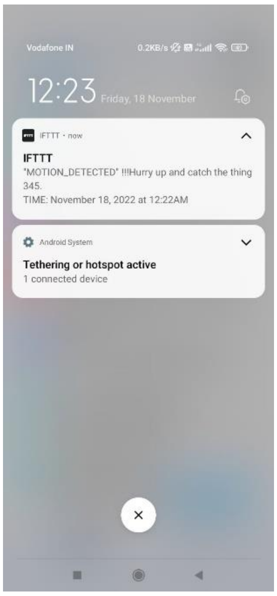

Fig. 20 Notification received in Mobile

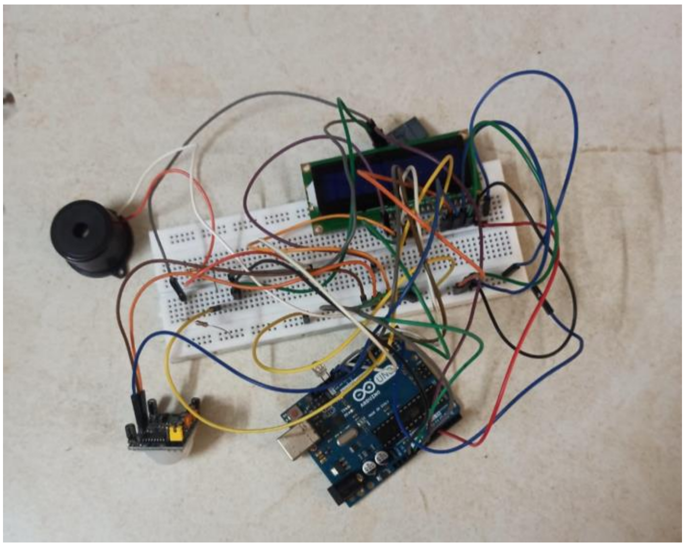

Fig. 21 Hardware Circuit

APPENDIX

```
#include <LiquidCrystal.h>
#define tempsensor A0
int led = 13;
int sensor = 6; //PIR sensor int state = LOW;
int val = 0; int buzzer=7;

byte degree[8] =
{
0b00011,    //To display the degree symbol on LCD 0b00011,
0b00000,
0b00000,
0b00000,
0b00000,
0b00000,
0b00000
};

const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2; LiquidCrystal lcd(rs, en, d4, d5, d6, d7);


String ssid = "Simulator Wifi"; // SSID to connect to String password = ""; // Our virtual wifi has no password
String host = "api.thingspeak.com"; // Open Weather Map API const int httpPort  = 80;
const int httpPort2 = 82;
String uri = "/update?api_key=05ZAWQYEQXF004T1&field1="; String uripir      =
"/update?api_key=05ZAWQYEQXF004T1&field2=";


int setupESP8266(void) {
// Start our ESP8266 Serial Communication
Serial.begin(115200);   // Serial connection over USB to computer Serial.println("AT"); // Serial connection on
Tx / Rx port to ESP8266 delay(10);  // Wait a little for the ESP to respond
if (!Serial.find("OK")) return 1;
```

```
// Connect to 123D Circuits Simulator Wifi Serial.println("AT+CWJAP=\"" + ssid + "\",\"" + password + "\"");
delay(10); // Wait a little for the ESP to respond
if (!Serial.find("OK")) return 2;

// Open TCP connection to the host: Serial.println("AT+CIPSTART=\"TCP\",\"" + host + "\"," + httpPort);
delay(50);  // Wait a little for the ESP to respond
if (!Serial.find("OK")) return 3;

return 0;

}

void anydata(void) {

int temp = analogRead(A0);  // read temperature sensor value int temp2=temp/2.048;
float temperature=map(temp2,20,358,-40,125);

int val = digitalRead(sensor);  // read PIR sensor value if (val == HIGH) {
int pirplot=map(val,0,1,0,1);
}
if (val == LOW) {
int pirplot=map(val,0,1,0,1);
}

// Construct our HTTP call

String httpPacket = "GET " + uri + float(temperature) + uripir + int(val) +" HTTP/1.1\r\nHost: " + host + "\r\n\r\n";

int length = httpPacket.length();
// Send our message length Serial.print("AT+CIPSEND="); Serial.println(length);
delay(10); // Wait a little for the ESP to respond if (!Serial.find(">")) return -1;

// Send our http request Serial.print(httpPacket);
delay(10); // Wait a little for the ESP to respond if (!Serial.find("SEND OK\r\n")) return;


}
```

Hardware Code:

```
void setup() {
   setupESP8266();
   lcd.begin(16,2);    //Start LCD lcd.createChar(1, degree); lcd.setCursor(0,0);
   lcd.print(" LCD "); lcd.setCursor(0,1); lcd.print(" switched ON "); delay(2000);
   lcd.clear();

   pinMode(led, OUTPUT);      //LED is declared as output pinMode(sensor, INPUT);        //PIR sensor value
declared input pinMode(buzzer,OUTPUT);      //Buzzer declared as output Serial.begin(115200);   //Baudrate
of 115200
}
void loop() {
```

```
    anydata();
    float reading=analogRead(A0);   //read the temperature sensor value float
temperature=reading/2.048;//converting to celcius
    delay(10);

    /*------Display Result  */
    lcd.clear(); lcd.setCursor(2,0); lcd.print("Temperature"); lcd.setCursor(4,1); lcd.print(temperature);
lcd.write(1); lcd.print("C"); delay(1000);

    val = digitalRead(sensor);  // read sensor value
    if (val == HIGH) {  // check if the sensor is HIGH digitalWrite(led, HIGH);// turn LED ON
digitalWrite(buzzer,HIGH);
    delay(500); // delay 500 milliseconds

    if (state == LOW) { Serial.println("Motion detected!"); lcd.clear();
    lcd.setCursor(2,0); lcd.print("Motion");
    lcd.setCursor(4,1); lcd.print("Detected");
    delay(5000);
    state = HIGH;    // update variable state to HIGH
    }
    }
    else {
    digitalWrite(led, LOW); // turn LED OFF digitalWrite(buzzer,LOW);
    delay(500); // delay 500 milliseconds

    if (state == HIGH){ Serial.println("Motion stopped!"); lcd.clear();
    lcd.setCursor(2,0); lcd.print("Motion");
    lcd.setCursor(4,1); lcd.print("stopped");
    delay(5000);
    state = LOW;     // update variable state to LOW
    }
    }
    delay(1000);
}
```

REFERENCES

[1] R. K. Kodali, V. Jain, S. Bose and L. Boppana, "IoT based smart security and home automation system," 2016 International Conference on Computing, Communication and Automation (ICCCA), 2016, pp. 1286- 1289, doi: 10.1109/CCAA.2016.7813916.

[2] X. Li, Z. Xuan and L. Wen, "Research on the Architecture of Trusted Security System Based on the Internet of Things," 2011 Fourth International Conference on Intelligent Computation Technology and Automation, 2011, pp. 1172-1175, doi: 10.1109/ICICTA.2011.578.

[3] Suseelan, Angel & Palaniappan, Satish & Hariharan, Naveen & Kesh, Naren & S, Vidhyalakshimi. (2015). Home Automation Systems - A Study. International Journal of Computer Applications. 116. 11-18. 10.5120/20379-2601.

[4] Abu, Mohd Azlan & Nordin, Siti Fatimah & Suboh, Mohd Zubir & Yid, Mohd & Ramli, Aizat. (2018). Design and Development of Home Security Systems based on Internet of Things Via Favoriot Platform. International Journal of Applied Engineering Research. 13.

[5] Peng, Z., Kato, T., Takahashi, H. and Kinoshita, T., 2015, October. Intelligent

home security system using agent-based IoT devices. In 2015 IEEE 4th Global Conference on Consumer Electronics (GCCE) (pp. 313- 314). IEEE.

[7] Computer Applications, 165(10), pp.33-38.

[6] Saifuzzaman, M., Khan, A.H., Moon, N.N. and Nur, F.N., 2017. Smart Security for an Organization based on IoT. International Journal of Computer Applications, 165(10), pp.33-38.