



**END SEMESTER ASSESSMENT (ESA)
B.TECH. (CSE)
IV SEMESTER**

**UE19CS256 – MICROPROCESSOR AND COMPUTER
ARCHITECTURE LABORATORY**

**Mini Project on
Server Protect**

SUBMITTED BY

NAME	SRN
Abhishek Aditya BS	PES1UG19CS019
A Sai Chaithanya	PES1UG19CS002

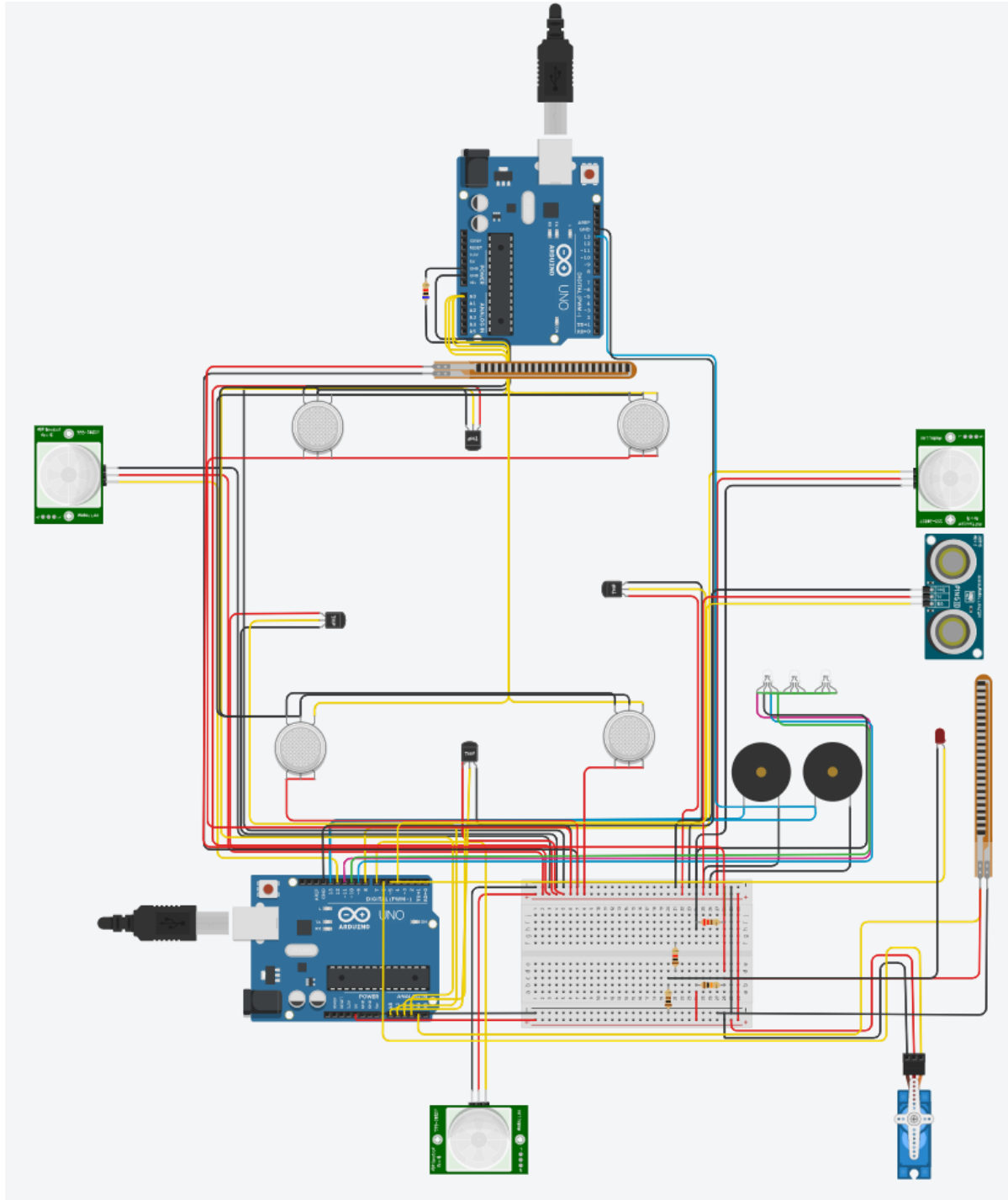
JANUARY – MAY 2021
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
RR CAMPUS,
BENGALURU – 560100, KARNATAKA, INDIA

TABLE OF CONTENTS		
Sl.No	TOPIC	PAGE No
1.	ABSTRACT OF THE PROJECT	3
2.	CIRCUIT DIAGRAM	4
3.	ARDUINO CODE	5-7
4.	SCREEN SHOTS OF THE OUTPUT	8-11
5.	REFERENCES	11

ABSTRACT OF THE PROJECT:

- Most of today's high computing services offered by different companies as different or on different platforms is responsible for a fast and efficient internet.
- For data to be processed at such high speeds specialized server farms are established across many countries. These server farms have computing devices running around the clock all the time. Hence, it is necessary to monitor them for any errors or issues.
- One Potential attention needed would be towards the cooling ability in the server farm. Hence, temperature sensors need to be used to measure the temperature and to keep all the devices running at optimal temperature for maximum performance.
- The server farms usually have the air composition consisting of hydrogen gas for certain extreme measures, now in case the control to the valve was hacked and all the gas has been released it could cause an explosion, hence gas sensors are required to keep a check on the air composition.
- In case some intruders try to access the mainframe of a certain set of servers then, some proximity sensors are required to detect physical movements.
- Early detection of failure is essential as these units will be housing petabytes of data and handling terabytes of traffic, hence backup of data and balancing of the traffic among units can be done gracefully to reduce the downtime.
- To enable this sensitive flex sensors placed at strategic points like structural point of failures can detect the failure of servers a precious few minutes before the actual failure occurs.

CIRCUIT DIAGRAM:



ARDUINO CODE:

Arduino 1:

```
#include <Servo.h>
#include <math.h>
int pirsensor1 = 0;
int pirsensor2 = 0;
int pirsensor3 = 0;

int celsius1 = 0;
int celsius2 = 0;
int celsius3 = 0;
int celsius4 = 0;
int fahrenheit1 = 0;
int fahrenheit2 = 0;
int fahrenheit3 = 0;
int fahrenheit4 = 0;
int baselinetemp = 5;

Servo my_servo;
double angle = 0.0;

int flex_value1;
int distance;

void setup()
{
    pinMode(13, OUTPUT);
    pinMode(7, INPUT);
    pinMode(11, OUTPUT);
    pinMode(10, OUTPUT);
    pinMode(9, OUTPUT);
    pinMode(5, OUTPUT);
    my_servo.attach(6);
    Serial.begin(9600);
}

void loop()
{
    pirsensor1 = digitalRead(7);
    pirsensor2 = digitalRead(12);
    pirsensor3 = digitalRead(8);

    pinMode(4, OUTPUT);
    digitalWrite(4, LOW);
    delayMicroseconds(2);
    digitalWrite(4, HIGH);
    delayMicroseconds(5);
    digitalWrite(4, LOW);

    pinMode(4, INPUT);
    distance = pulseIn(4, HIGH);
    distance = distance/29/2;

    celsius1 = map(((analogRead(A0) - 20) * 3.04), 0, 1023, -40, 125);
    celsius2 = map(((analogRead(A1) - 20) * 3.04), 0, 1023, -40, 125);
```

```

celsius3 = map(((analogRead(A2) - 20) * 3.04), 0, 1023, -40, 125);
celsius4 = map(((analogRead(A3) - 20) * 3.04), 0, 1023, -40, 125);
fahrenheit1 = ((celsius1 * 9) / 5 + 32);
fahrenheit2 = ((celsius2 * 9) / 5 + 32);
fahrenheit3 = ((celsius3 * 9) / 5 + 32);
fahrenheit4 = ((celsius4 * 9) / 5 + 32);

Serial.println(celsius1);
Serial.println(celsius2);
Serial.println(celsius3);
Serial.println(celsius4);

flex_value1 = map(analogRead(A4),600,920,0,100);

if(flex_value1 >= 30)
{
    digitalWrite(5,HIGH);
}
else
{
    digitalWrite(5,LOW);
}

if (abs(celsius1) < baselinetemp && abs(celsius2) < baselinetemp && \
    abs(celsius3) < baselinetemp && abs(celsius4) < baselinetemp)
{
    analogWrite(11, 0);
    analogWrite(10, 255);
    analogWrite(9, 0);
}

else if((abs(celsius1) >= baselinetemp && abs(celsius1) < baselinetemp + 10) || \
    abs(celsius2) >= baselinetemp && abs(celsius2) < baselinetemp + 10 || \
    abs(celsius3) >= baselinetemp && abs(celsius3) < baselinetemp + 10 || \
    abs(celsius4) >= baselinetemp && abs(celsius4) < baselinetemp + 10)

{
    analogWrite(11, 254);
    analogWrite(10, 216);
    analogWrite(9, 177);
}

else if(abs(celsius1)>=baselinetemp + 10 && abs(celsius1) < baselinetemp + 20 || \
    abs(celsius2) >= baselinetemp + 10 && abs(celsius2) < baselinetemp + 20 || \
    abs(celsius3) >= baselinetemp + 10 && abs(celsius3) < baselinetemp + 20 || \
    abs(celsius4) >= baselinetemp + 10 && abs(celsius4) < baselinetemp + 20)
{
    analogWrite(11, 100);
    analogWrite(10, 55);
    analogWrite(9, 0);
}

else if(abs(celsius1)>=baselinetemp + 20 && abs(celsius1) < baselinetemp + 30 || \
    abs(celsius2) >= baselinetemp + 20 && abs(celsius2) < baselinetemp + 30 || \
    abs(celsius3) >= baselinetemp + 20 && abs(celsius3) < baselinetemp + 30 || \
    abs(celsius4) >= baselinetemp + 20 && abs(celsius4) < baselinetemp + 30)
{
    analogWrite(11, 255);
    analogWrite(10, 99);
}

```

```

        analogWrite(9, 71);
    }
else if(abs(celsius1)>=baselinetemp + 30 || abs(celsius2) >= baselinetemp + 30 ||\
abs(celsius3) >= baselinetemp + 30 || abs(celsius4) >= baselinetemp + 30)
{
    analogWrite(11, 255);
    analogWrite(10, 0);
    analogWrite(9, 0);
}
if(pirsensor1 == HIGH || pirsensor2 == HIGH || pirsensor3==HIGH)
{
    Serial.println("if");
    int freq = 2000;
    if(distance>=25 && distance<=350)
    {
        freq = distance * 100;
    }
    tone(13,freq,200);
    if (pirsensor1 == HIGH)
    {
        my_servo.write(20);
    }
    if (pirsensor2== HIGH)
    {
        my_servo.write(60);
    }

    if (pirsensor3== HIGH)
    {
        my_servo.write(120);
    }
}
else
{
    noTone(13);
}
delay(100);
}

```

Arduino 2:

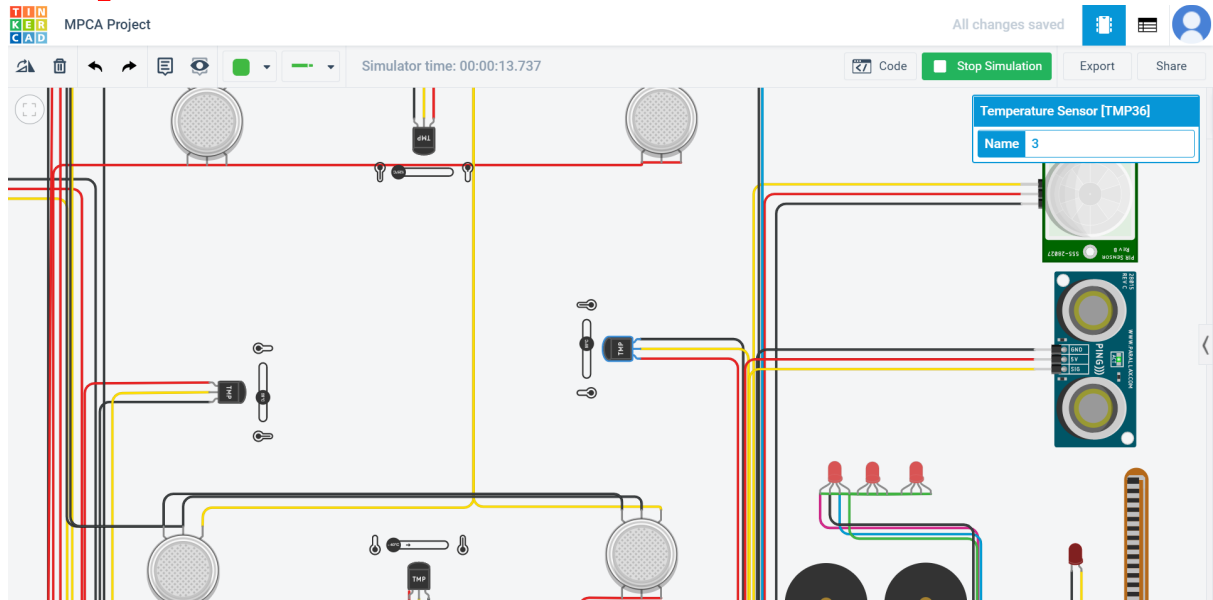
```

int gas1 = A0;
void setup() {
    pinMode(13, OUTPUT);
    pinMode(gas1,INPUT);
    pinMode(3,OUTPUT);
    Serial.begin(9600);
}
void loop() {
    int gasSensor = analogRead(gas1);
    if ((gasSensor > 750)) {
        tone(13,1000,200);
    }
    else
    { noTone(13); }
}

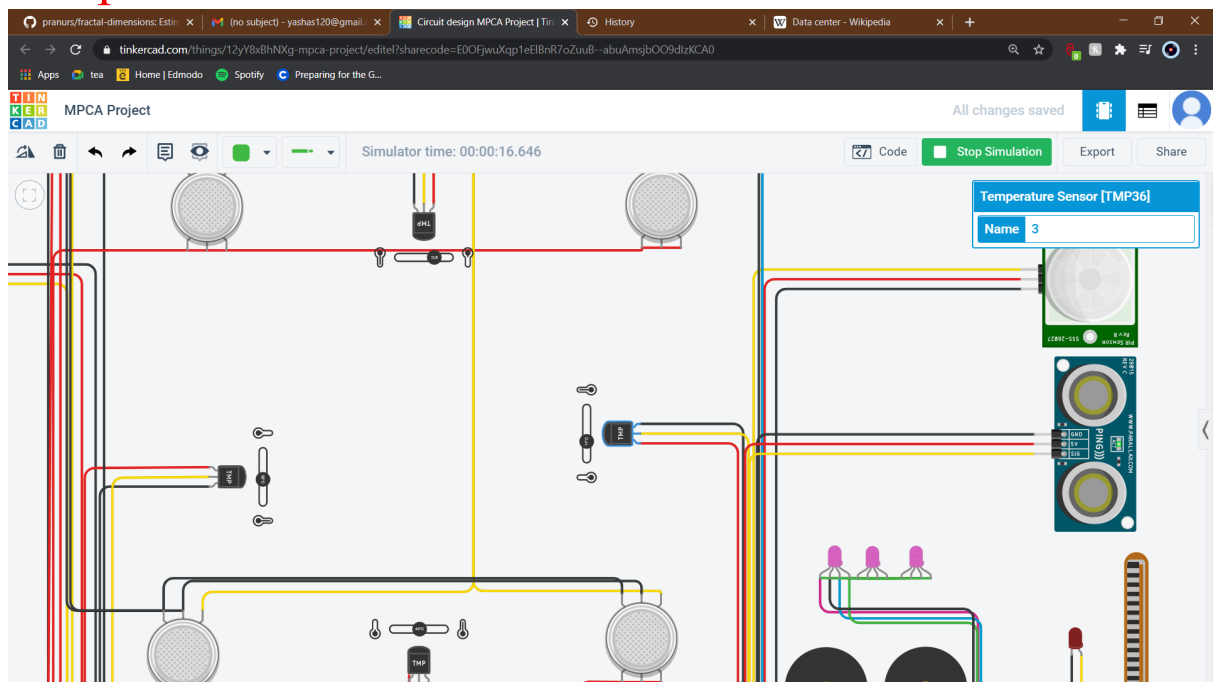
```

SCREEN SHOTS OF THE OUTPUT:

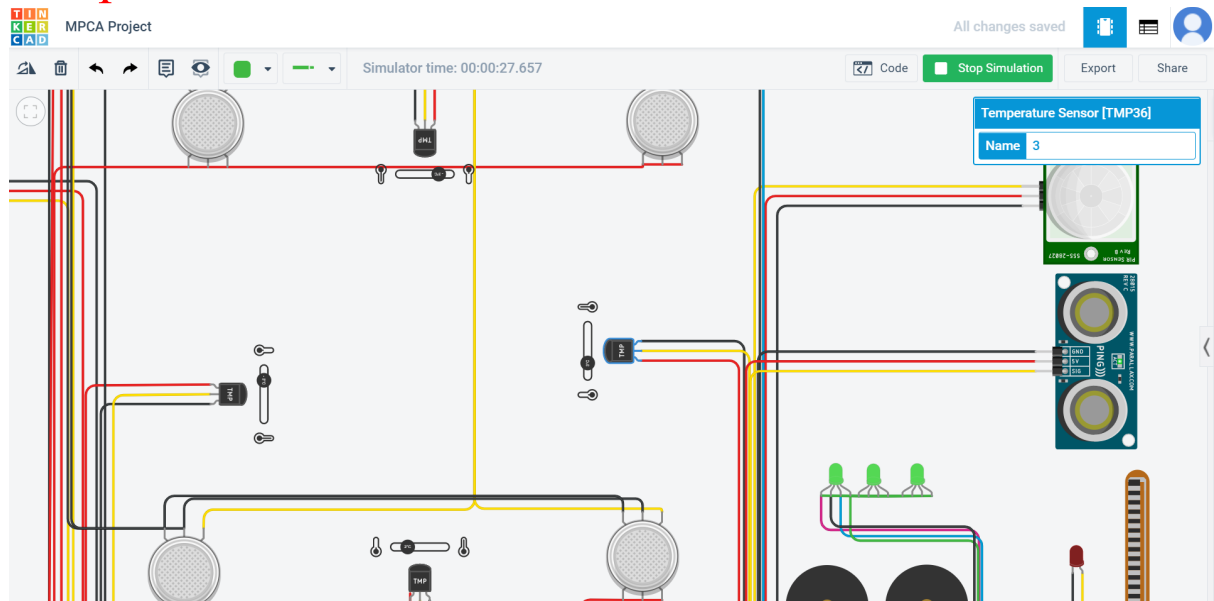
Temperature Critical:



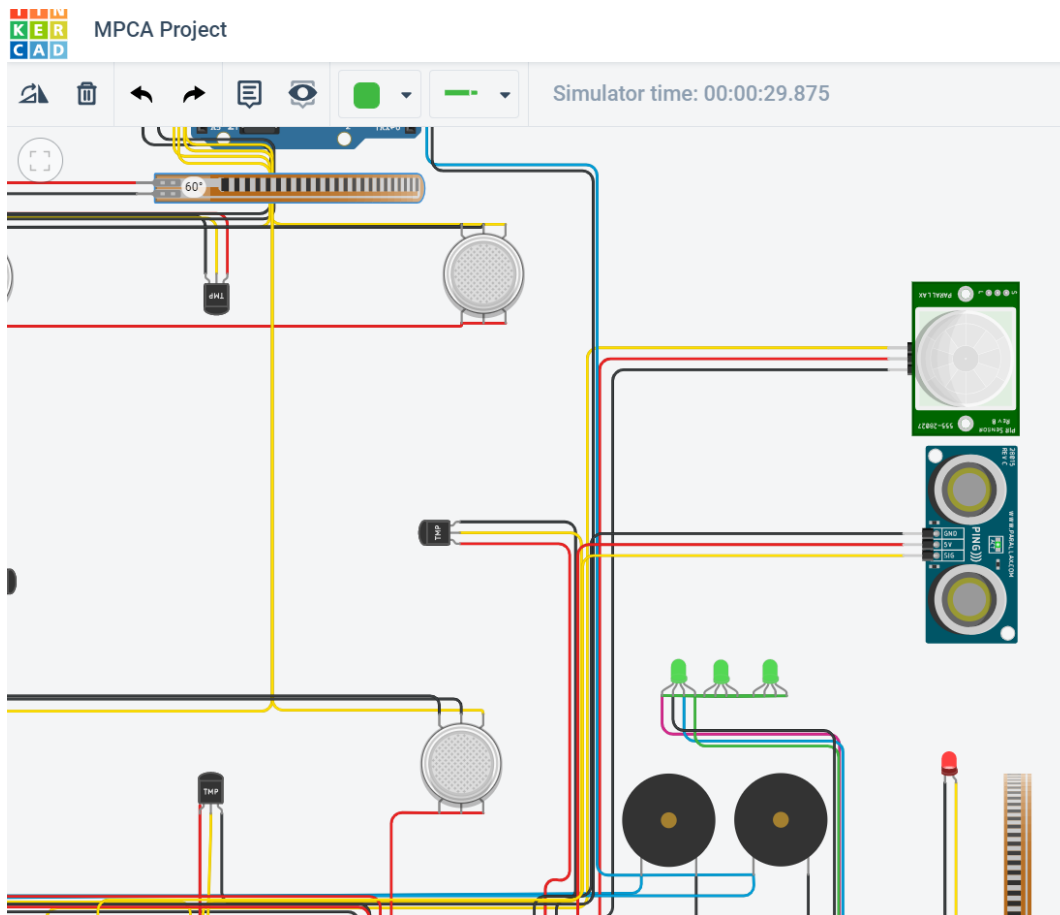
Temperature Abnormal:



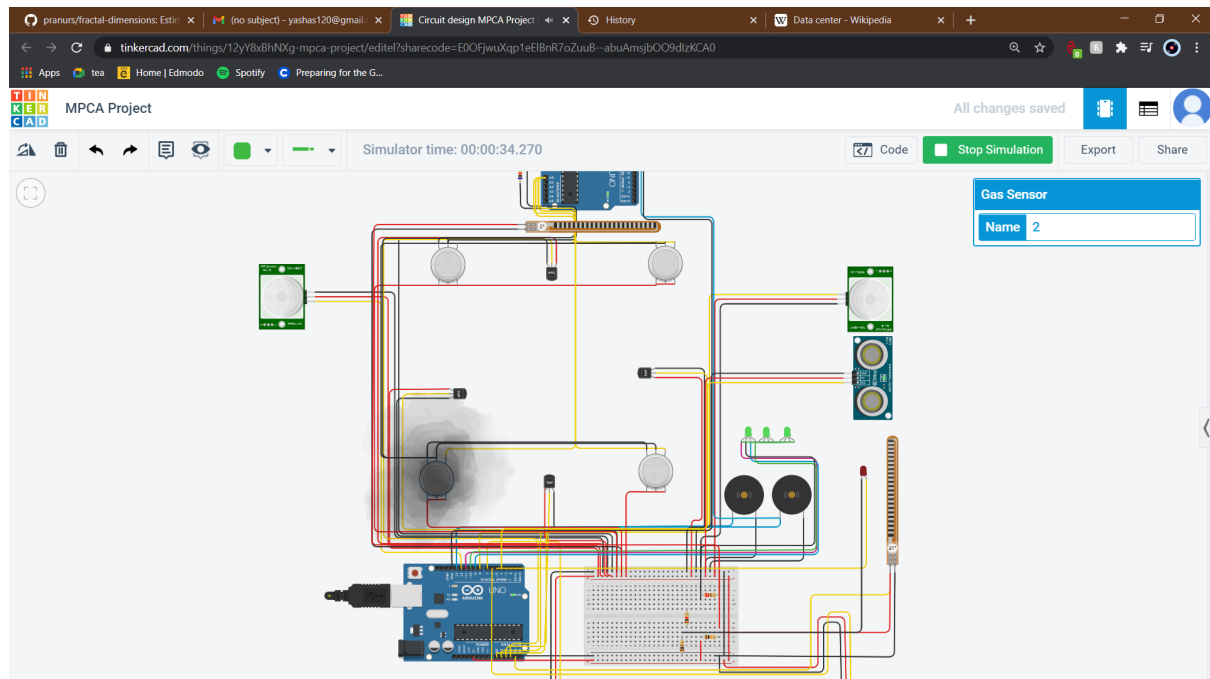
Temperature Nominal:



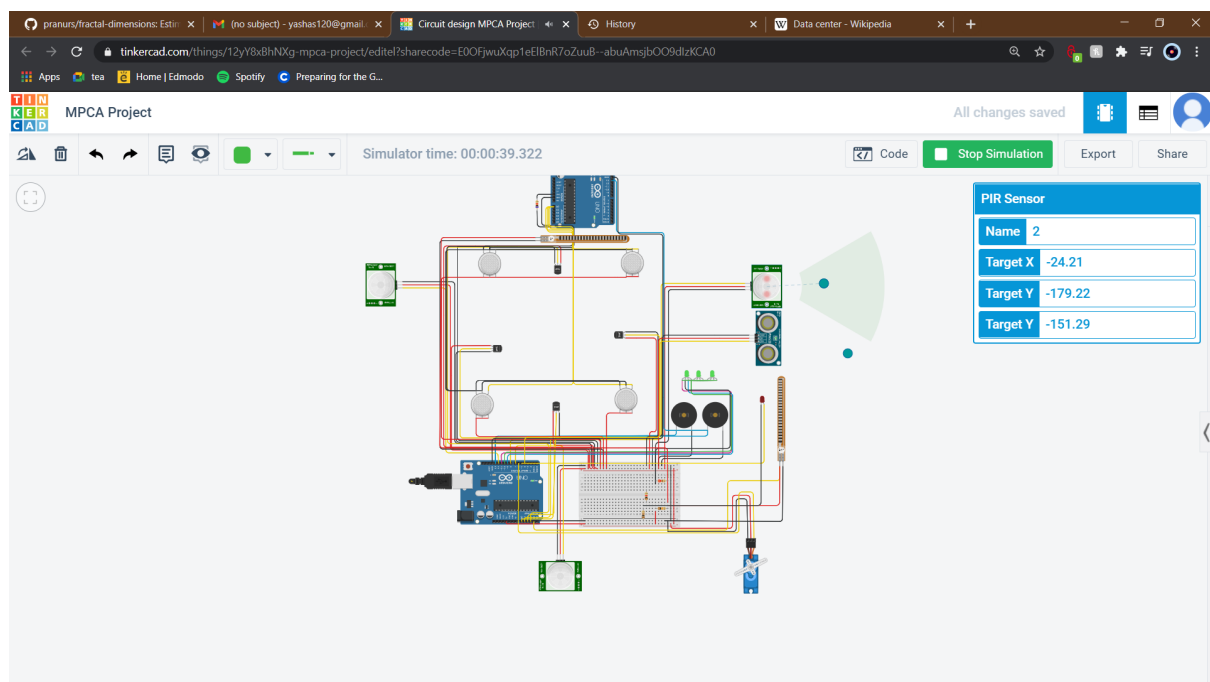
Flex Sensor:

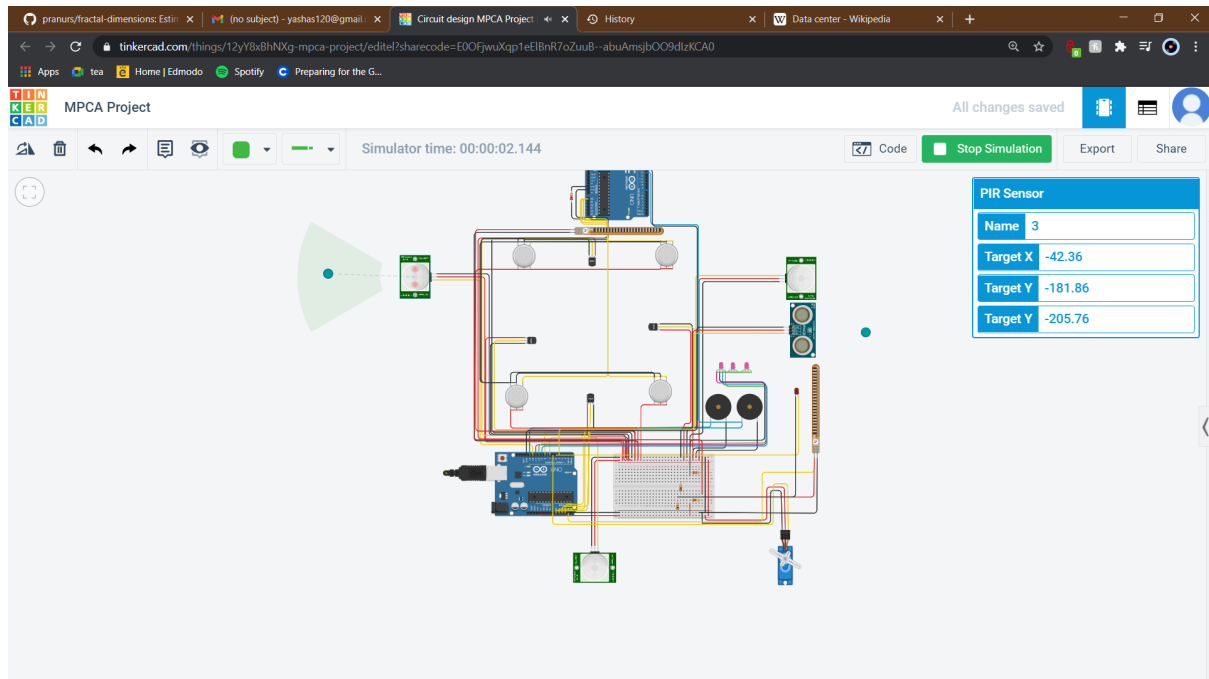


Smoke Sensor:



Motion Detection:





REFERENCES

- <https://www.tinkercad.com/things/gmlVPAIOwXt-flex-sensor>
- <https://www.tinkercad.com/things/1Sg0hWGQjiU-gas-sensor>
- <https://www.tinkercad.com/things/fzViPd8sSge-servo-motor>
- <https://www.tinkercad.com/things/2ovnobxW8bN-copy-of-rgb-led/editel?tenant=circuits>