



MINI PROJECT REPORT
ON
MiniWeather Station

NESARA K V
PES1UG19CS180

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.	ARDUINO CODE	5-11
3.	SCREENSHOTS OF THE OUTPUT	12-15
4.	REFERENCES	16

2

ABSTRACT OF THE PROJECT:

As the title suggests our project is based on building a simulation of Weather reports, which include temperature, pressure, gas in atmosphere, UV radiation, and also a security sensor as an add-on.

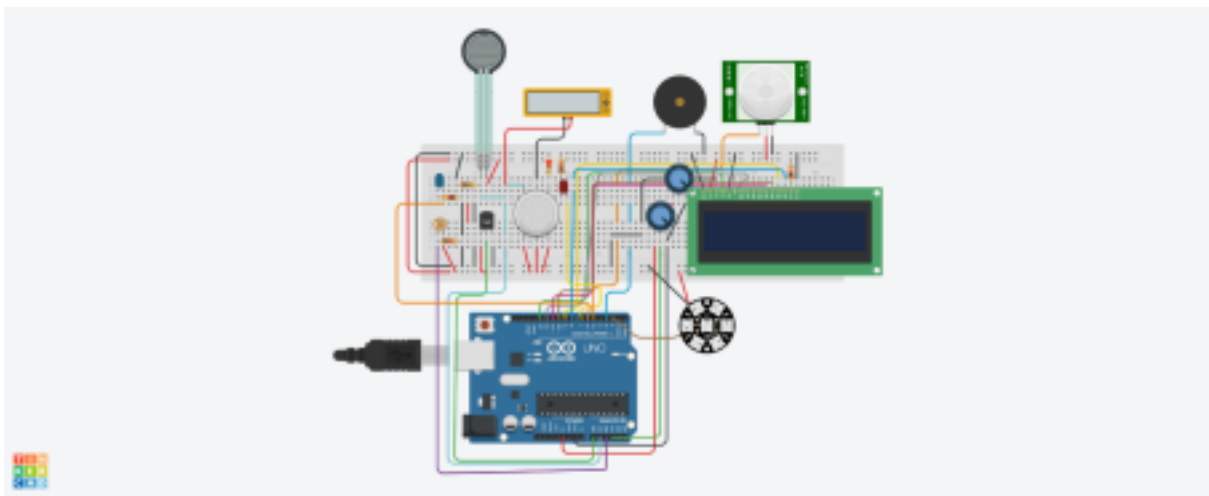
The weather station collects the above data related to the weather and environment using different sensors:

- Temperature
- Force

- Gas
- Photoresistor
- PIV

The aim is to make a small and simple weather station, using a tinkercad.

Here's an Abstract View of our Project:



3

ARDUINO CODE:

```
#include <LiquidCrystal.h>
#include <Adafruit_NeoPixel.h>

#define PIN 2 // pin Neopixel is attached to
#define NUMPIXELS 1 // number of neopixels in strip
#define SENSOR A3 //input pin for Potentiometer
#define dryThreshold 50 //below this value, begin alerting dry, turn red; #define
wetThreshold 200 //above this value, begin alerting wet,turn blue; #define
thresholdCenter (dryThreshold + wetThreshold)/2 //Brightest Green point #define
crossFade 20 //how much blue and red should fade in to green

#define PIR_In 2
#define buzzer 3
#define red 7
#define blue 6
```

```

Adafruit_NeoPixel pixels = Adafruit_NeoPixel(NUMPIXELS, PIN, NEO_GRB +
NEO_KHZ800);
int delayval = 100;
int redColor = 0;
int greenColor = 0;
int blueColor = 0;

int sensorValue = 0;
int transitionValue = 0;

LiquidCrystal lcd(13, 12, 11, 10, 9, 8);

int sensor_Input;
int sensorPin;
float temp;
int Forcevalue=0;
int value;
float Lb=0.0;
float N;

```

5

```

void setup(void)
{

  pixels.begin();
  pinMode(SENSOR,INPUT);

  Serial.begin(9600);
  lcd.begin(16, 2);

  pinMode(buzzer, OUTPUT);
  pinMode(PIR_In, INPUT);

  pinMode(sensorPin,INPUT);
  pinMode(red,OUTPUT);

  pinMode(sensorPin,INPUT);
  pinMode(blue,OUTPUT);

```

```

}

void loop()
{
  lcd.setCursor(0,0);
  delay(100);
  lcd.print("ALLCLEAR :)");

  check_For_Intruder();
  delay(3000);

  measure_Force();

  measure_Temp();
  display_Temp();

  check_For_Gas();

  display_Photo();
}

```

6

//CodeforPhotoresistor

```

void display_Photo()
{
  value=analogRead(A2);
  Serial.print("Intensityof light : ");
  Serial.println(value);
  lcd.setCursor(0,0);

  if(value < 500)
  {
    digitalWrite(blue,HIGH);
    lcd.print("Low intensity");
    delay(3000);
  }
}

```

```

else
{
    digitalWrite(blue,LOW);
    lcd.print("High intensity");
    delay(3000);
}
lcd.clear();
}

```

```
//CodeforForceSensor
```

```

voidmeasure_Force()
{
    Forcevalue=analogRead(A1);
    P=(Forcevalue/1.0);
    //pressure=Force/unitarea

    if(P > 40.0)
    {
        digitalWrite(red, HIGH);
    }
    else
    {
        digitalWrite(red, LOW);
    }

    lcd.setCursor(0, 0);
    Serial.print("Pressure(N/m2):");

```

7

```

Serial.print(P);
Serial.println();
lcd.setCursor(0,0);
lcd.print("Pressure(N/m2):");
lcd.setCursor(5, 1);
lcd.print(P);
delay(3000);
lcd.clear();
}

```

```
//CodeforGasSensor
```

```
voidcheck_For_Gas()
```

```

{
  intval =analogRead(A1);
  Serial.print("Gasvalue: ");
  Serial.println(val);
  lcd.clear();
    if (val>0 && val <=6){
      lcd.print("Normal");
      delay(3000);
    }
  else if(val >6&&val <=9){
      lcd.print("Gas Detected");
      delay(3000);
    }
  else if(val >9&&val <=12){
      lcd.print("DenseGasDet");
      delay(3000);
    }
  else if(val >12){
      lcd.print("EMERGENCY!");
      delay(3000);
    }
  lcd.clear();
}

```

//CodeforTemperatureSensor

```

voidmeasure_Temp()
{
  sensor_Input=analogRead(A0);
  temp=(float)sensor_Input/1024;

```

```

  temp=temp*5;
  temp=temp-0.5;
  temp=temp*100;
}
voiddisplay_Temp()
{
  lcd.setCursor(0,0);
  Serial.print("Temp. inCelsius: ");
  Serial.print(temp);
  Serial.println();
  lcd.print("Temp. inCelsius: ");

```

```
    lcd.setCursor(5, 1);  
    lcd.print(temp);  
    delay(3000);  
}
```

```
//CodeforPIRSensor
```

```
void check_For_Intruder()  
{  
    boolean sensorvalue = digitalRead(PIR_In);  
    if(sensorvalue == 1)  
    {  
        digitalWrite(buzzer, HIGH);  
        tone(buzzer, 1000, 10000);  
        lcd.setCursor(0, 0);  
        lcd.print("Intruder in the ");  
        lcd.setCursor(0, 1);  
        lcd.print("Station : ( ");  
        delay(3000);  
        lcd.clear();  
    }  
    else  
    {  
        noTone(buzzer);  
        digitalWrite(buzzer, LOW);  
    }  
    delay(2);  
}
```

```
//NeoPixel Jewel
```

```
void Rain_Pixel()
```

```
{  
    sensorValue = analogRead(SENSOR);  
    transitionValue = map(sensorValue, 0, 1023, 0, 255);  
    setColor();  
    pixels.setPixelColor(0, redColor, greenColor, blueColor);  
    pixels.show();  
}
```

```
int value = analogRead(A3);  
Serial.print("Rain Sensor value: ");
```



```

Serial.println(value);

if (value <= 10) {
  lcd.setCursor(0, 0);
  lcd.print(" Dry Weather ");
  Serial.println("Dry Weather");
  lcd.setCursor(0, 1);
  lcd.print(" No Rain ");
}

if ((value > 10) && (value <= 17)) {
  Serial.println("Heavy Mist ");
  lcd.setCursor(0, 0);
  lcd.print(" ");
  lcd.setCursor(0, 1);
  lcd.print(" Heavy Mist ");
}

if ((value > 328) && (value <= 511)) {
  Serial.println(" Drizzle ");
  lcd.setCursor(0, 0);
  lcd.print(" ");
  lcd.setCursor(0, 1);
  lcd.print(" Drizzle ");
}

if ((value > 512) && (value <= 700)) {
  Serial.println("Average Rainfall");
  lcd.setCursor(0, 0);
  lcd.print(" ");
  lcd.setCursor(0, 1);
  lcd.print("Average Rainfall ");
}

if ((value > 700) && (value <= 800)) {
  Serial.println("Heavy Rain");
  lcd.setCursor(0, 0);
  lcd.print(" ");
  lcd.setCursor(0, 1);
  lcd.print(" Heavy Rain ");
}

```

```

if (value > 800) {
  Serial.println("Extreme Rain");
  lcd.setCursor(0, 0);
  lcd.print(" Very ");
  lcd.setCursor(0, 1);
  lcd.print(" Extreme Rain ");
}
delay(3000);
lcd.clear();
}

void setColor()
{

  redColor = ((transitionValue <= dryThreshold + crossFade) && (transitionValue >= 0
))? map(transitionValue,0,dryThreshold + crossFade,255,0) : 0;
  blueColor = (transitionValue >= wetThreshold - crossFade && transitionValue <=
255)? map(transitionValue,wetThreshold - crossFade,255,0,255):0;

  if(transitionValue >= dryThreshold && transitionValue <=
thresholdCenter) {
    greenColor = map(transitionValue,dryThreshold,thresholdCenter,0,255);
  }
  else if(transitionValue > thresholdCenter && transitionValue < wetThreshold)
  {
    greenColor = map(transitionValue,dryThreshold,thresholdCenter,255,0);
  }
  else{
    greenColor = 0;
  }
}
}

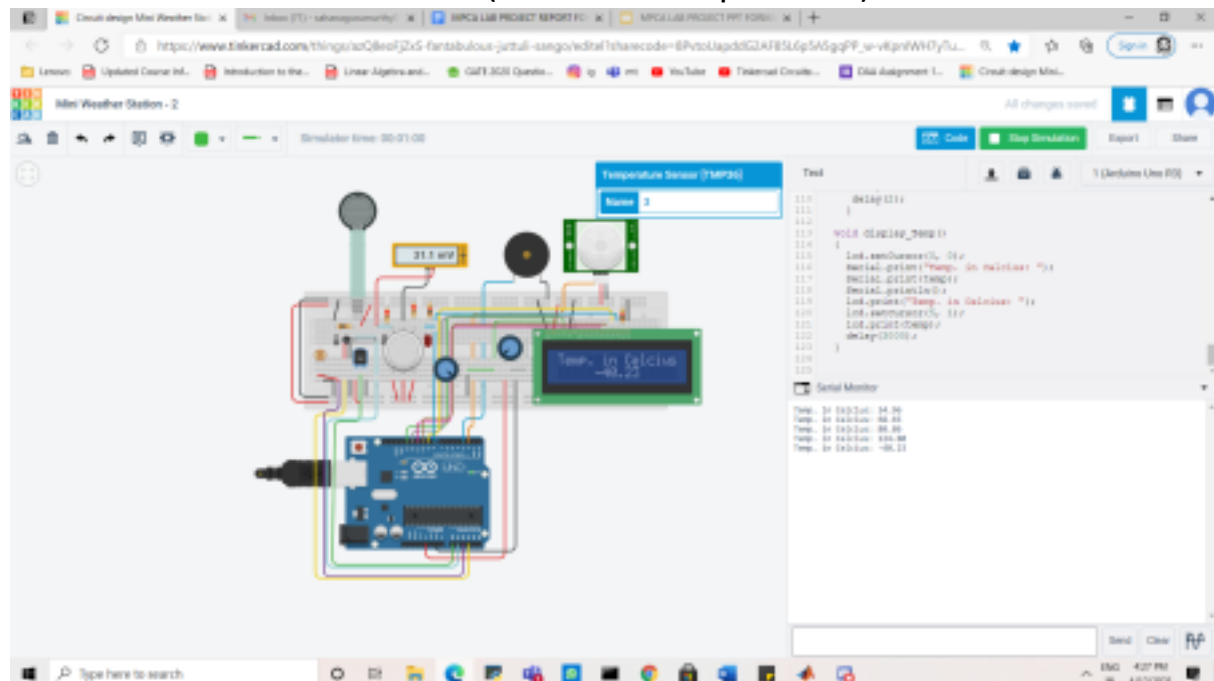
```

SCREENSHOTS OF THE OUTPUT:

I.TEMPERATURE SENSOR

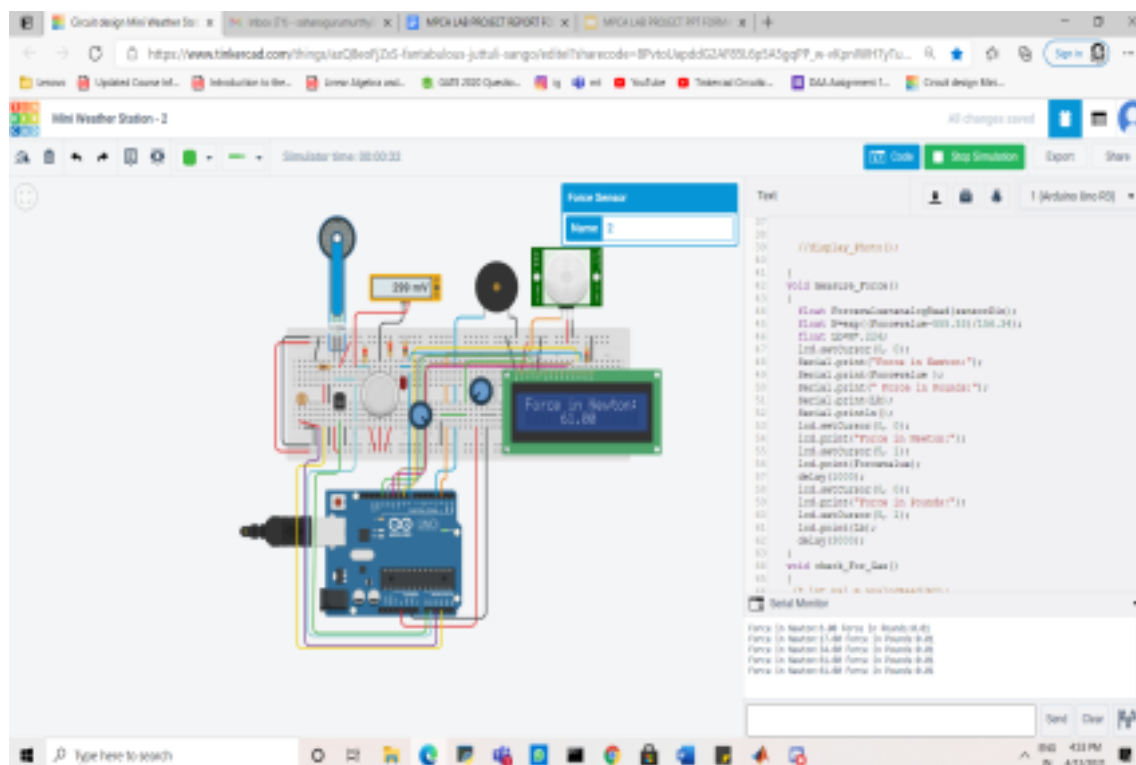
A Temperature sensor is an electronic device that measures the

temperature of its environment and analog output from the temperature sensor is converted to celsius (SI unit of Temperature).



II.FORCE SENSOR

Force sensors are devices that are designed to translate applied mechanical forces , such as tensile and compressive forces, and pressure into output signals and this analog output from force sensor i.e, force in



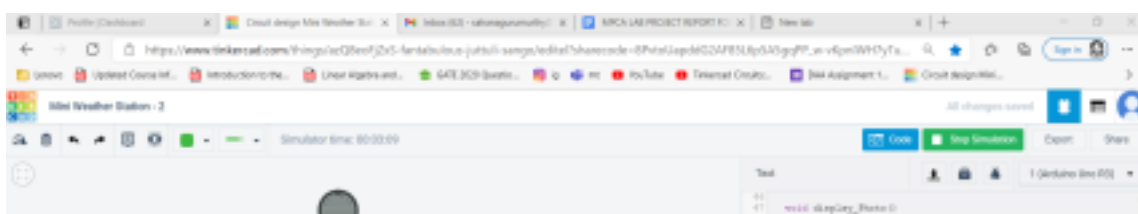
newton (SI unit of Force) is converted to pressure (F/A) in Nm^2 and value is displayed in led.

The screenshot shows the Arduino IDE interface for a project titled "Mini Weather Station - 2". The main workspace displays a breadboard circuit. An Arduino Uno is connected to a gas sensor module (labeled "Gas Sensor" with "Name 1" in a text box). A 100k potentiometer is also connected. An LCD screen displays the text "EMERGENCY!". The code in the editor is as follows:

```

104 //
105
106 void check_Far_Gas()
107 {
108   int val = analogRead(A0);
109   Serial.print("Gas value: ");
110   Serial.println(val);
111   delay(1000);
112   if (val == 0) {
113     Serial.print("Gas Detected!");
114     delay(1000);
115   }
116   else if (val > 0 && val <= 10) {
117     Serial.print("Concent Gas Sen!");
118     delay(1000);
119   }
120   else if (val > 10 && val <= 20) {
121     Serial.print("GAS DANGER!");
122     delay(1000);
123   }
124   else if (val > 20) {
125     Serial.print("GAS DANGER!");
126     delay(1000);
127   }
128 }
129
130 void setup() {
131   //
132   //
133   //
134   //
135   //
136   //
137   //
138   //
139   //
140   //
141   //
142   //
143   //
144   //
145   //
146   //
147   //
148   //
149   //
150   //
151   //
152   //
153   //
154   //
155   //
156   //
157   //
158   //
159   //
160   //
161   //
162   //
163   //
164   //
165   //
166   //
167   //
168   //
169   //
170   //
171   //
172   //
173   //
174   //
175   //
176   //
177   //
178   //
179   //
180   //
181   //
182   //
183   //
184   //
185   //
186   //
187   //
188   //
189   //
190   //
191   //
192   //
193   //
194   //
195   //
196   //
197   //
198   //
199   //
200   //
201   //
202   //
203   //
204   //
205   //
206   //
207   //
208   //
209   //
210   //
211   //
212   //
213   //
214   //
215   //
216   //
217   //
218   //
219   //
220   //
221   //
222   //
223   //
224   //
225   //
226   //
227   //
228   //
229   //
230   //
231   //
232   //
233   //
234   //
235   //
236   //
237   //
238   //
239   //
240   //
241   //
242   //
243   //
244   //
245   //
246   //
247   //
248   //
249   //
250   //
251   //
252   //
253   //
254   //
255   //
256   //
257   //
258   //
259   //
260   //
261   //
262   //
263   //
264   //
265   //
266   //
267   //
268   //
269   //
270   //
271   //
272   //
273   //
274   //
275   //
276   //
277   //
278   //
279   //
280   //
281   //
282   //
283   //
284   //
285   //
286   //
287   //
288   //
289   //
290   //
291   //
292   //
293   //
294   //
295   //
296   //
297   //
298   //
299   //
300   //
301   //
302   //
303   //
304   //
305   //
306   //
307   //
308   //
309   //
310   //
311   //
312   //
313   //
314   //
315   //
316   //
317   //
318   //
319   //
320   //
321   //
322   //
323   //
324   //
325   //
326   //
327   //
328   //
329   //
330   //
331   //
332   //
333   //
334   //
335   //
336   //
337   //
338   //
339   //
340   //
341   //
342   //
343   //
344   //
345   //
346   //
347   //
348   //
349   //
350   //
351   //
352   //
353   //
354   //
355   //
356   //
357   //
358   //
359   //
360   //
361   //
362   //
363   //
364   //
365   //
366   //
367   //
368   //
369   //
370   //
371   //
372   //
373   //
374   //
375   //
376   //
377   //
378   //
379   //
380   //
381   //
382   //
383   //
384   //
385   //
386   //
387   //
388   //
389   //
390   //
391   //
392   //
393   //
394   //
395   //
396   //
397   //
398   //
399   //
400   //
401   //
402   //
403   //
404   //
405   //
406   //
407   //
408   //
409   //
410   //
411   //
412   //
413   //
414   //
415   //
416   //
417   //
418   //
419   //
420   //
421   //
422   //
423   //
424   //
425   //
426   //
427   //
428   //
429   //
430   //
431   //
432   //
433   //
434   //
435   //
436   //
437   //
438   //
439   //
440   //
441   //
442   //
443   //
444   //
445   //
446   //
447   //
448   //
449   //
450   //
451   //
452   //
453   //
454   //
455   //
456   //
457   //
458   //
459   //
460   //
461   //
462   //
463   //
464   //
465   //
466   //
467   //
468   //
469   //
470   //
471   //
472   //
473   //
474   //
475   //
476   //
477   //
478   //
479   //
480   //
481   //
482   //
483   //
484   //
485   //
486   //
487   //
488   //
489   //
490   //
491   //
492   //
493   //
494   //
495   //
496   //
497   //
498   //
499   //
500   //
501   //
502   //
503   //
504   //
505   //
506   //
507   //
508   //
509   //
510   //
511   //
512   //
513   //
514   //
515   //
516   //
517   //
518   //
519   //
520   //
521   //
522   //
523   //
524   //
525   //
526   //
527   //
528   //
529   //
530   //
531   //
532   //
533   //
534   //
535   //
536   //
537   //
538   //
539   //
540   //
541   //
542   //
543   //
544   //
545   //
546   //
547   //
548   //
549   //
550   //
551   //
552   //
553   //
554   //
555   //
556   //
557   //
558   //
559   //
560   //
561   //
562   //
563   //
564   //
565   //
566   //
567   //
568   //
569   //
570   //
571   //
572   //
573   //
574   //
575   //
576   //
577   //
578   //
579   //
580   //
581   //
582   //
583   //
584   //
585   //
586   //
587   //
588   //
589   //
590   //
591   //
592   //
593   //
594   //
595   //
596   //
597   //
598   //
599   //
600   //
601   //
602   //
603   //
604   //
605   //
606   //
607   //
608   //
609   //
610   //
611   //
612   //
613   //
614   //
615   //
616   //
617   //
618   //
619   //
620   //
621   //
622   //
623   //
624   //
625   //
626   //
627   //
628   //
629   //
630   //
631   //
632   //
633   //
634   //
635   //
636   //
637   //
638   //
639   //
640   //
641   //
642   //
643   //
644   //
645   //
646   //
647   //
648   //
649   //
650   //
651   //
652   //
653   //
654   //
655   //
656   //
657   //
658   //
659   //
660   //
661   //
662   //
663   //
664   //
665   //
666   //
667   //
668   //
669   //
670   //
671   //
672   //
673   //
674   //
675   //
676   //
677   //
678   //
679   //
680   //
681   //
682   //
683   //
684   //
685   //
686   //
687   //
688   //
689   //
690   //
691   //
692   //
693   //
694   //
695   //
696   //
697   //
698   //
699   //
700   //
701   //
702   //
703   //
704   //
705   //
706   //
707   //
708   //
709   //
710   //
711   //
712   //
713   //
714   //
715   //
716   //
717   //
718   //
719   //
720   //
721   //
```

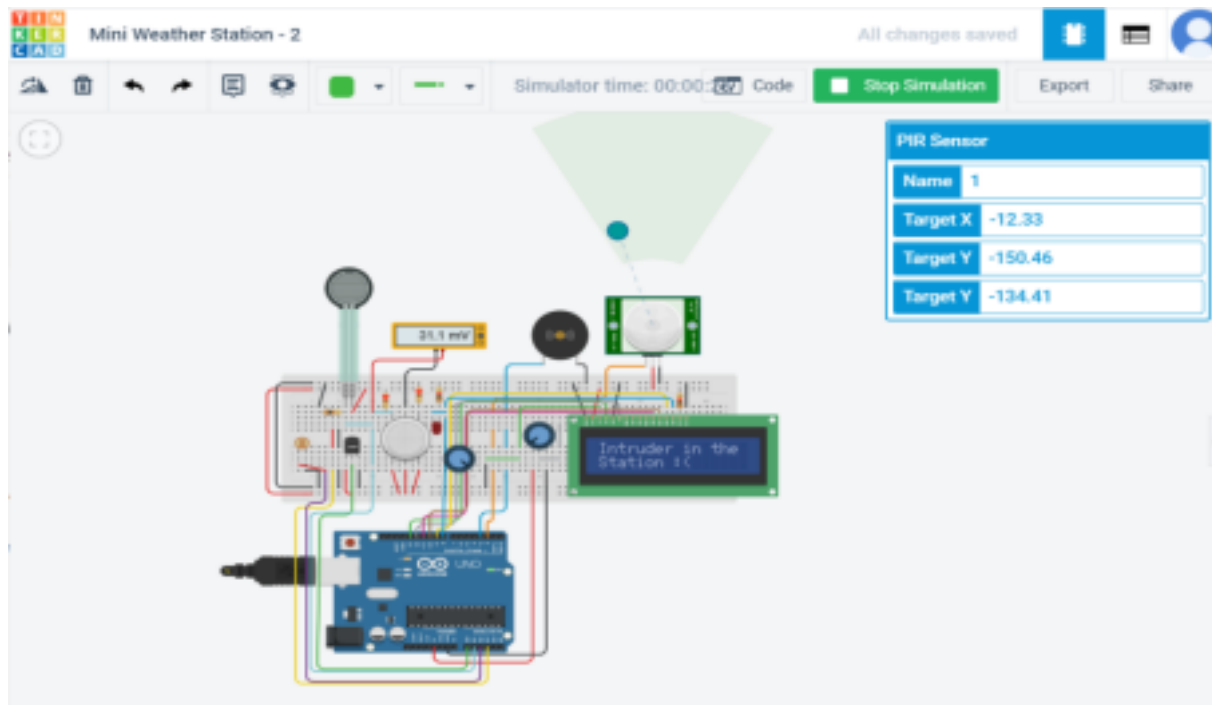
Photoresistor sensors are light sensitive devices most often used to indicate the presence or absence of light or to measure light intensity, output from a photoresistor sensor is measured (i.e, intensity of light)



13

PIR sensors allow you to sense motion, almost always used to detect whether a human has moved in or out of the sensor's range. The PIR acts as a digital output so all you need to do is listen for the pin to flip high or low. The motion can be detected by checking for a high signal on a single I/O pin.



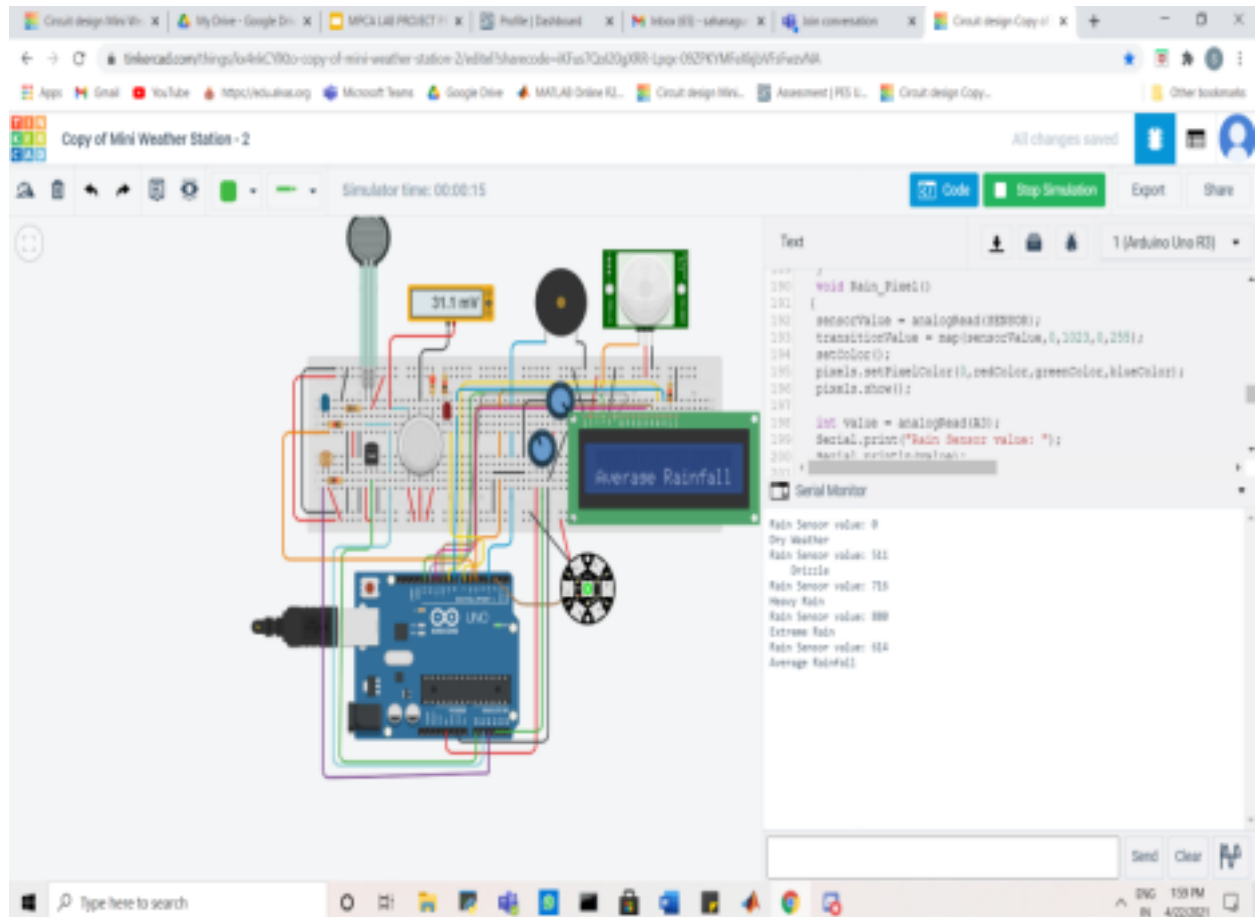


- In presence of Intruder(Buzzer beeps)

14

VI. NeoPixel Jewel:

A NeoPixel is just a name that Adafruit uses to refer to a bunch of different RGB (and RGBW) individually addressable LEDs. We're using a NeoPixel that is using a WS2812 RGB individually addressable LED. The LED itself has three inputs and three outputs. Power, ground, and data are the three inputs, and they are passed through to the next LED in the strip, allowing you to set the Red, Green, and Blue intensities for each LED in the strip individually. The data signal to control the LEDs brightness is timing dependent, in this implementation the RGB values vary according to the potentiometer value, and depending on the range the LCD will display the information regarding rainfall.



15

REFERENCES

These wonderful websites helped us build this project:

1. www.arduino.cc
2. www.instructortables.com
3. www.bluedot.space

4. www.scientiffic.medium.com
5. www.youtube.com
6. [LucidCharts](#) to draw the circuit