

Team ID	PNT2022TMID44448
Submitted by	T.Priyadharsini
Topic	Signs With smart connectivity for better road safety
Assignment 2	Built a python code , assume you get temperature and humidity values (generated with random function to a variable) and write a condition to continuously detect a alarm in case of high temperature

CODING :

```

#define ADC_RESOLUTION 1024.0
#define PIN_LM35      A0
const int TRIG_PIN    = 6; // Arduino pin connected to Ultrasonic Sensor's
TRIG pin
const int ECHO_PIN    = 7; // Arduino pin connected to Ultrasonic Sensor's
ECHO pin
const int BUZZER_PIN = 3; // Arduino pin connected to Piezo Buzzer's pin
const int DISTANCE_THRESHOLD = 50; // centimeters
float duration_us, distance_cm;
void setup() {
    Serial.begin(9600);
    // switch to Internal 1.1V Reference
    pinMode(TRIG_PIN, OUTPUT); // set arduino pin to output mode
    pinMode(ECHO_PIN, INPUT);  // set arduino pin to input mode
    pinMode(BUZZER_PIN, OUTPUT); // set arduino pin to output mode
    analogReference(INTERNAL);
    pinMode(LED_BUILTIN, OUTPUT);
}
void loop() {

```

```

// get the ADC value from the temperature sensor
int adcVal = analogRead(PIN_LM35);
// convert the ADC value to voltage in millivolt
float milliVolt = adcVal * (ADC_VREF_mV / ADC_RESOLUTION); //
ADC_VREF_mV = 1100 mV
// convert the voltage to the temperature in Celsius
float tempC = milliVolt / 10;
// convert the Celsius to Fahrenheit
float tempF = tempC * 9 / 5 + 32;
// print the temperature in the Serial Monitor:
Serial.print("Temperature: ");
Serial.print(tempC); // print the temperature in Celsius
Serial.print("°C");
Serial.print(" ~ "); // separator between Celsius and Fahrenheit
Serial.print(tempF); // print the temperature in Fahrenheit
Serial.println("°F");
digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the
voltage level)
delay(1000); // wait for a second
digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the
voltage LOW
delay(1000);
delay(1000);
digitalWrite(TRIG_PIN, HIGH);
delayMicroseconds(10);
digitalWrite(TRIG_PIN, LOW);
// measure duration of pulse from ECHO pin
duration_us = pulseIn(ECHO_PIN, HIGH);
// calculate the distance
distance_cm = 0.017 * duration_us;
if(distance_cm < DISTANCE_THRESHOLD)
    digitalWrite(BUZZER_PIN, HIGH); // turn on Piezo Buzzer
else
    digitalWrite(BUZZER_PIN, LOW); // turn off Piezo Buzzer
// print the value to Serial Monitor
Serial.print("distance: ");
Serial.print(distance_cm);

```

```
Serial.println(" cm");  
delay(500);
```

```
}
```

Footer

© 2022 GitHub, Inc.

Footer navigation

T