SNS COLLEGE OF TECHNOLOGY

Coimbatore-35

An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with ‘A++’ Grade

Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

19ECT213- IoT SYSTEM ARCHITECTURE

II ECE / IV SEMESTER

UNIT 2 – MICROCONTROLLER AND INTERFACING TECHNIQUES FOR IoT DEVICES

TOPIC 6 –Analog Sensor Interfacing

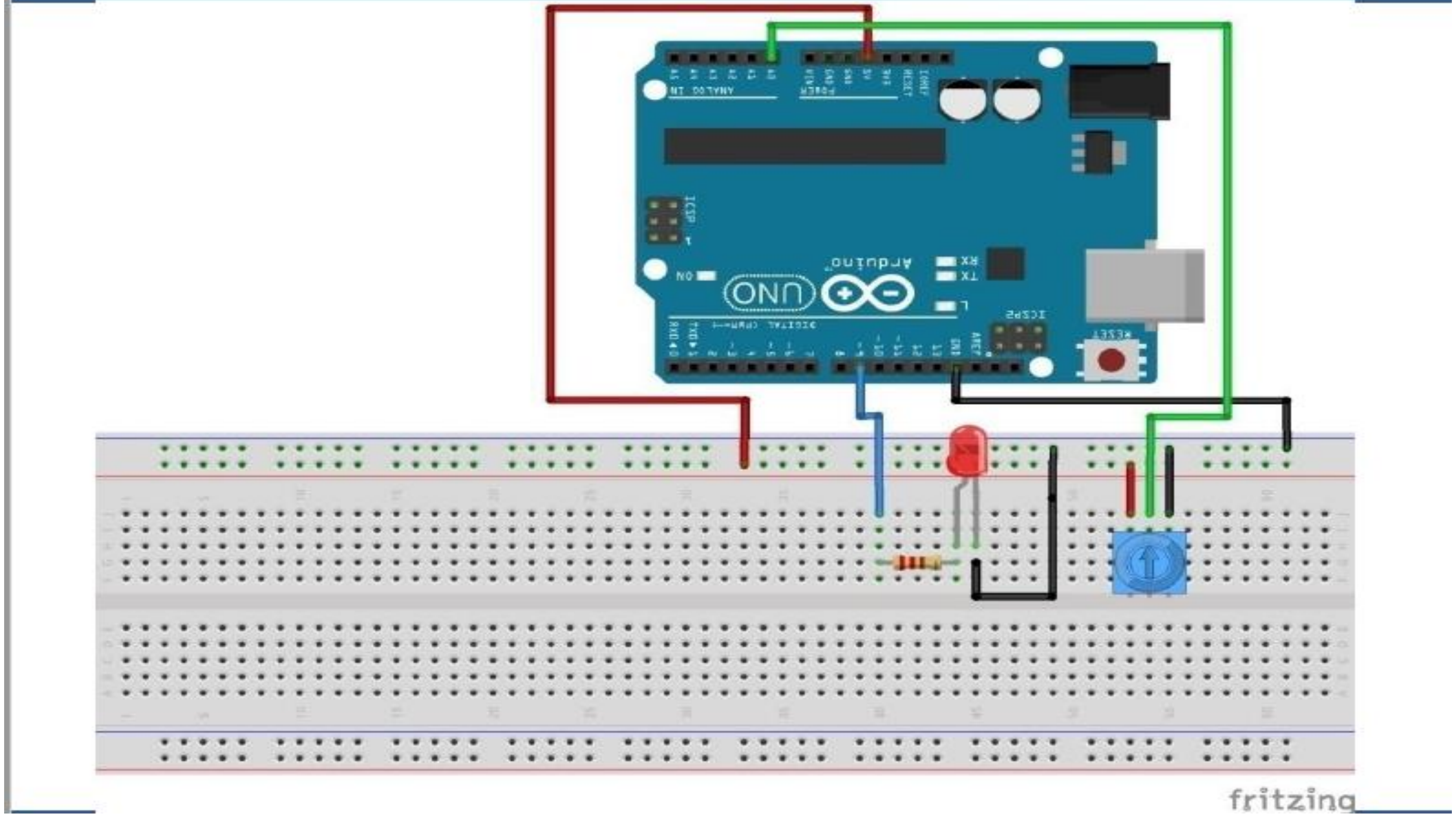
Creating a Dimmable LED using Potentiometer

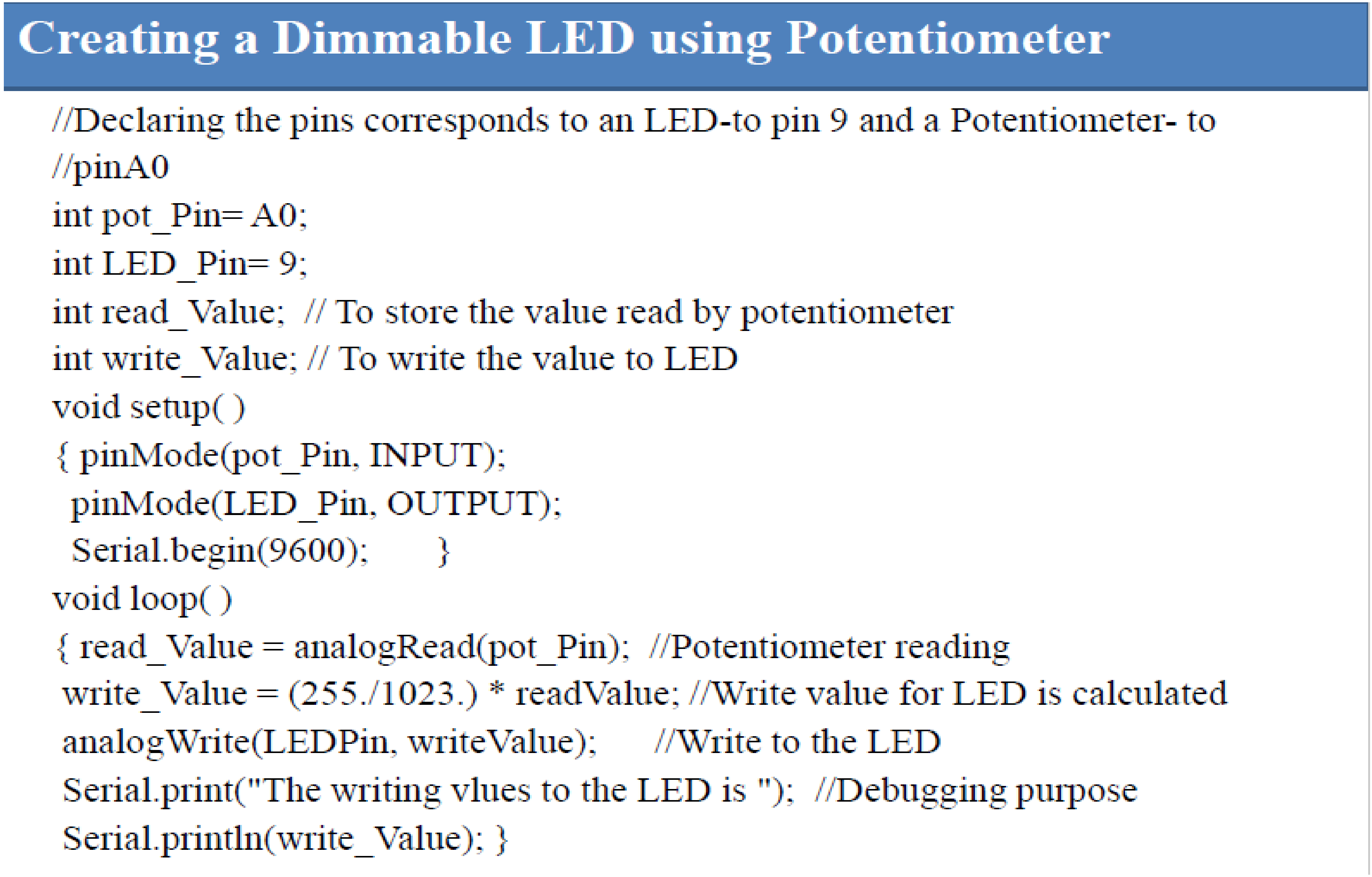
Components Required	1-LED, 220Ω resistor, 1-Potentiometer, Jumper wires, Breadboard

In this program we dim the LED based on the value read from the potentiometer. A "0" value from potentiometer is a **"0V"** and a value **"1023"** from potentiometer is a **"5V"**, which means we need to write a value of **255**. Hence we need to scale our read values from the potentiometer which falls between 0 to 1023 to suitable write values to be between 0 to 255 using the below given formulae.

write value=(255/1023)* read_value

Creating a Dimmable LED using Potentiometer





Creating a Dimmable LED using Potentiometer

```
//Declaring the pins corresponds to an LED-to pin 9 and a Potentiometer- to
//pinA0
int pot_Pin= A0;
int LED_Pin= 9;
int read_Value;  // To store the value read by potentiometer
int write_Value; // To write the value to LED
void setup( )
{ pinMode(pot_Pin, INPUT);
  pinMode(LED_Pin, OUTPUT);
  Serial.begin(9600);      }
void loop( )
{ read_Value = analogRead(pot_Pin);  //Potentiometer reading
 write_Value = (255./1023.) * readValue; //Write value for LED is calculated
 analogWrite(LEDPin, writeValue);      //Write to the LED
 Serial.print("The writing vlues to the LED is ");  //Debugging purpose
 Serial.println(write_Value); }
```

Interfacing Sensors to the Arduino

- **Temperature Sensor**
- **Light Sensor**
- **Ultrasonic distance sensor**
- **Line sensor** (infrared).

Interfacing Temperature Sensor

Components Required	**Buzzer, LM35 Temperature Sensor, Jumper wires, Breadboard**

LM35 Temperature Sensor:

The LM35 series are the gadgets with precision integrated circuit temperature whose yield voltage falls directly corresponding to the Centigrade temperature.

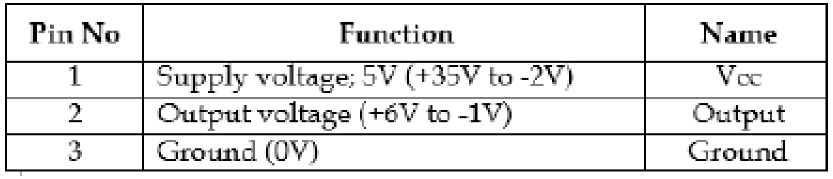
□ Calibrated Directly in Celsius (Centigrade)

□ Operates from 4 V to 30 V

□ Ranges are evaluated from Full −55°C to 150°C.

□ Suitable for Remote Applications

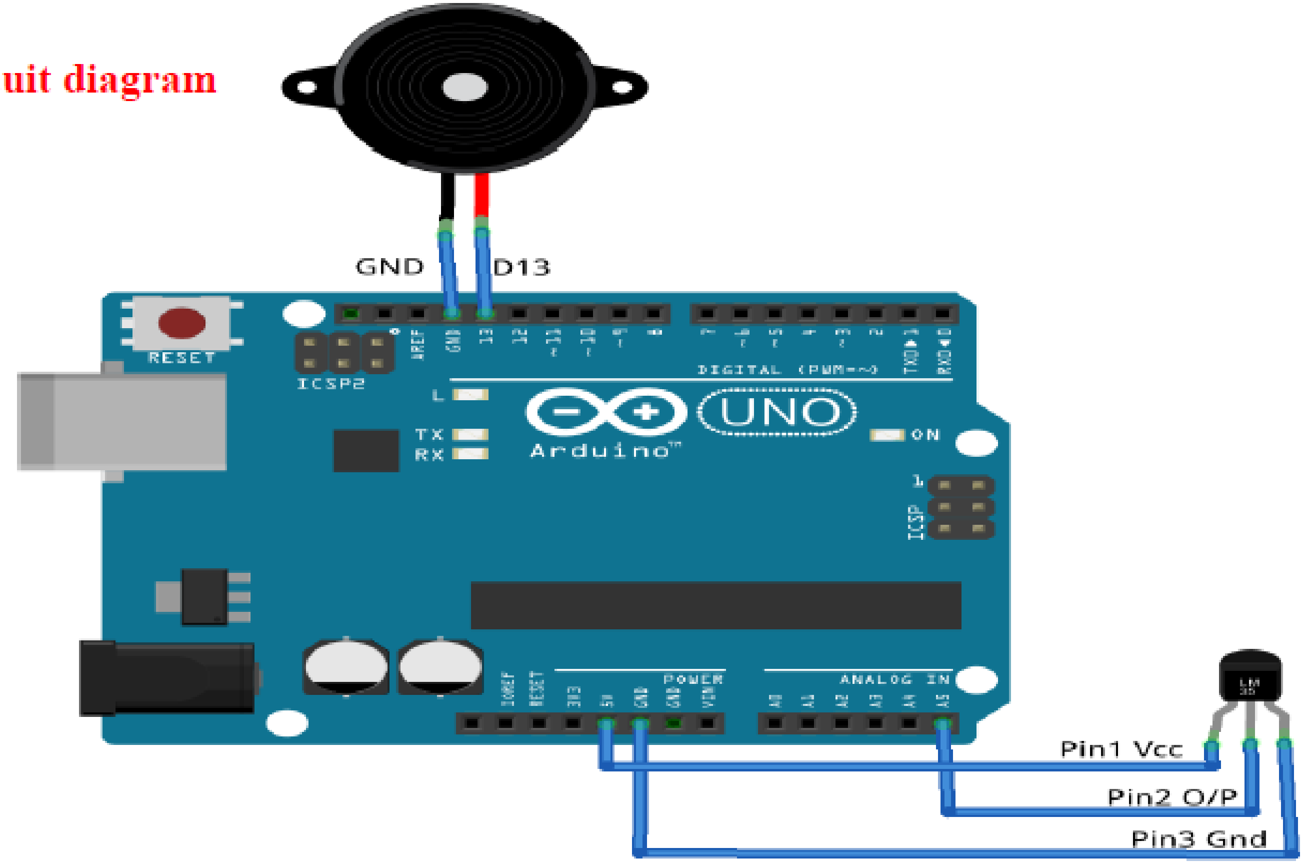
□ Used in Battery Management



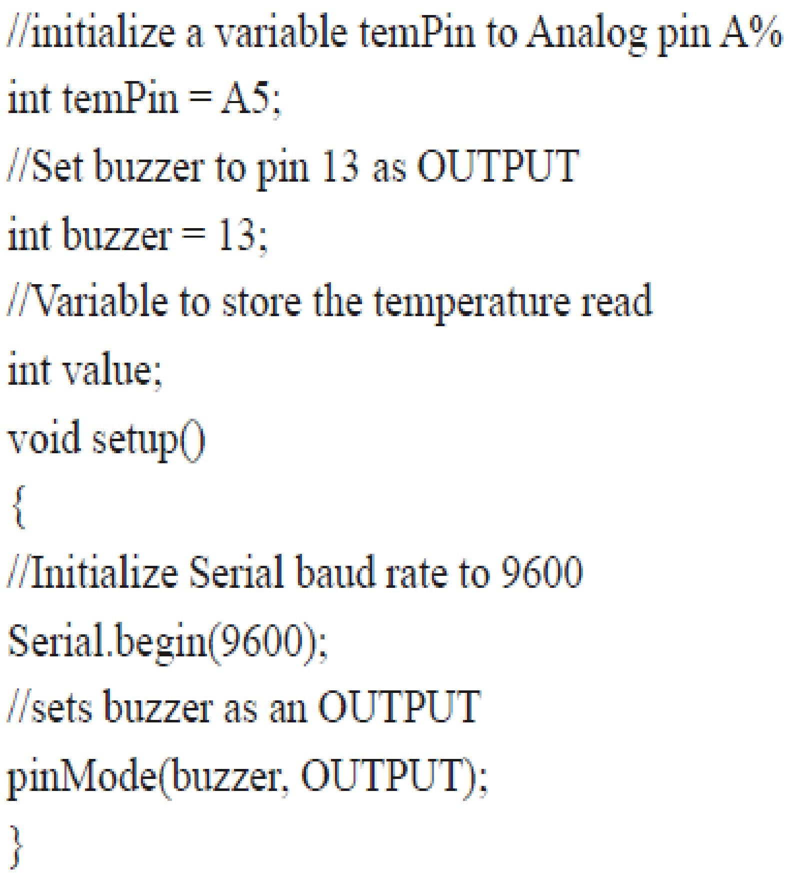
Pin No	Function	Name
1	Supply voltage; 5V (+35V to -2V)	Vcc
2	Output voltage (+6V to -1V)	Output
3	Ground (0V)	Ground

Interfacing Temperature Sensor

Circuit diagram

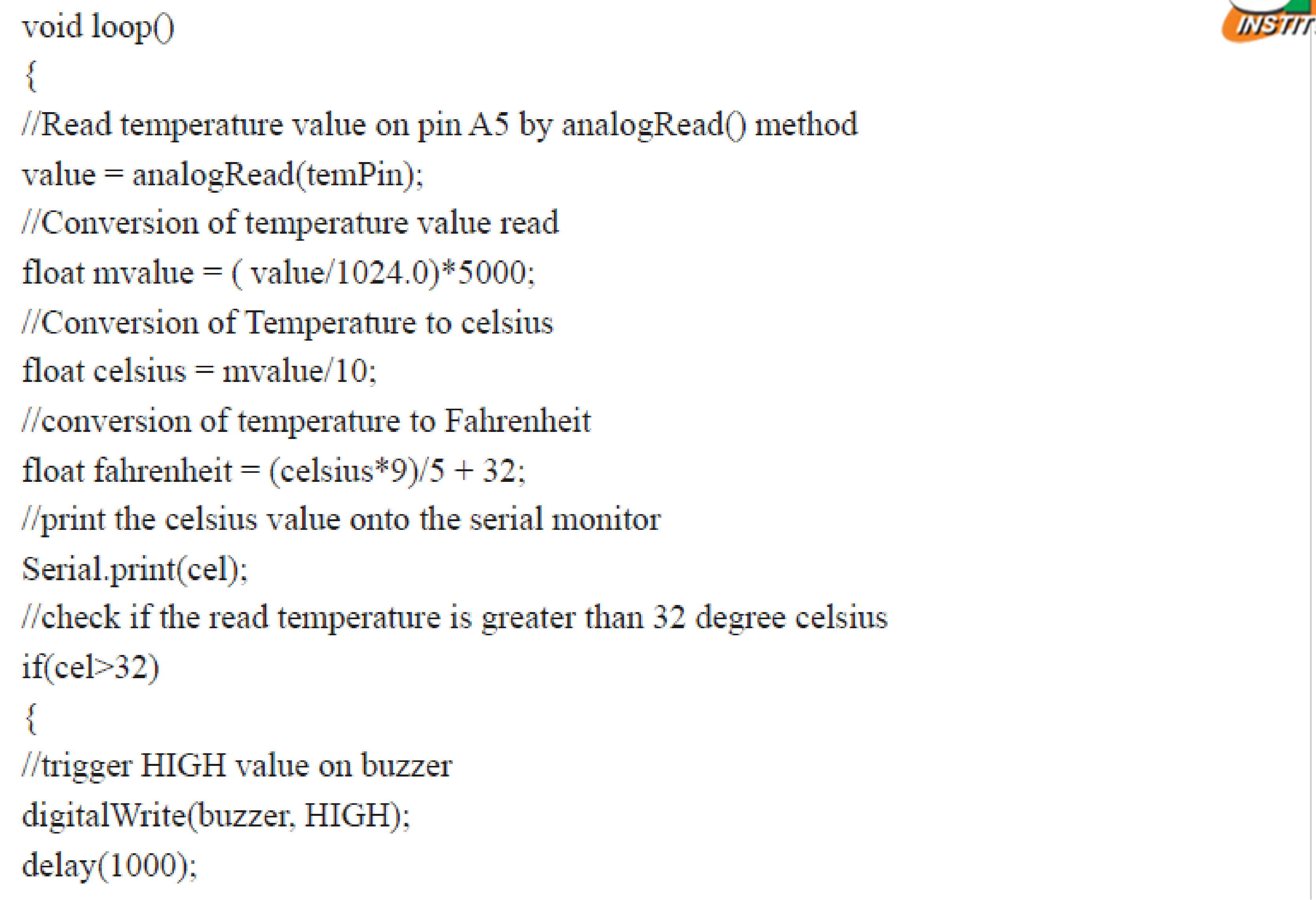


Interfacing Temperature Sensor



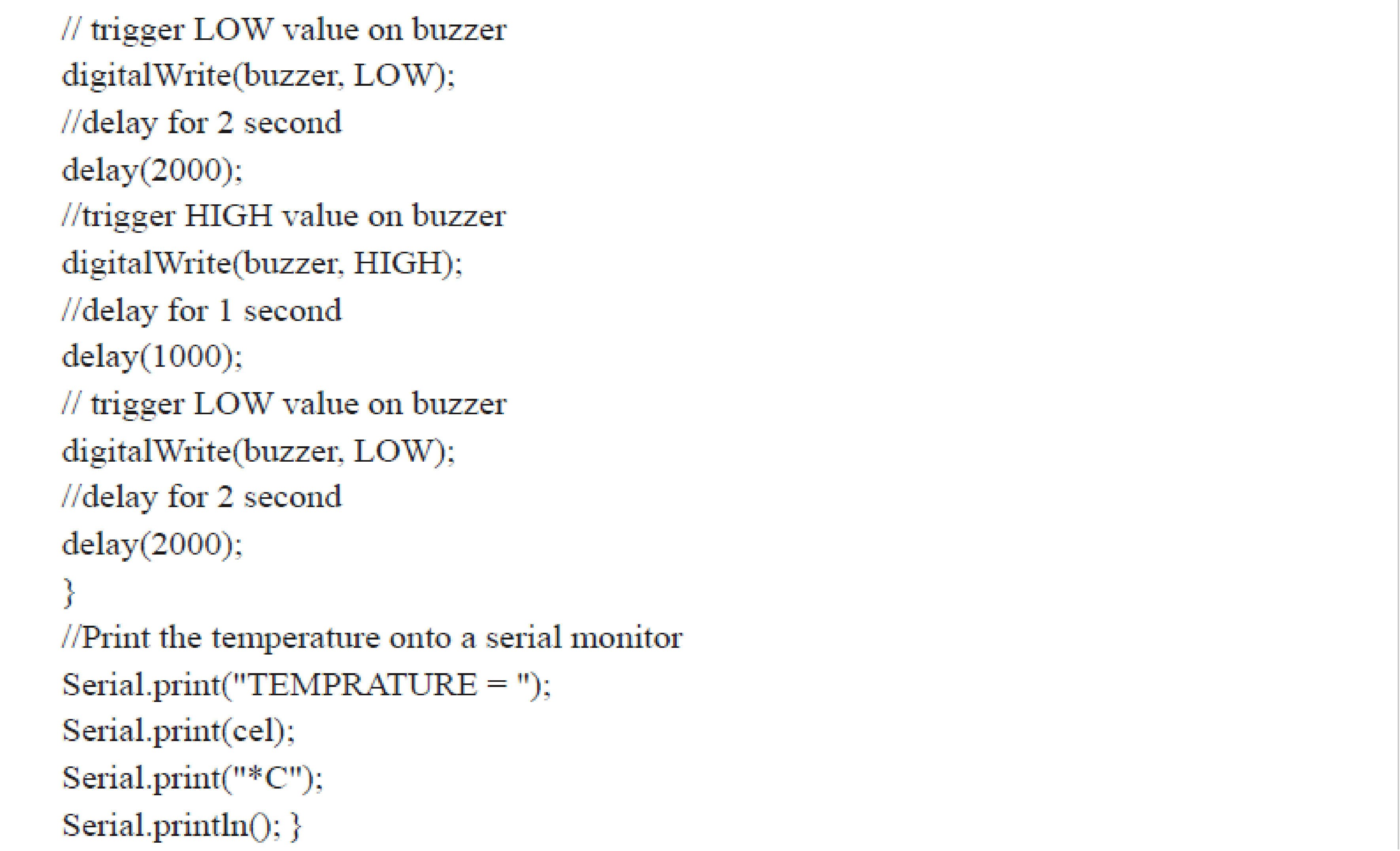
```
//initialize a variable temPin to Analog pin A%
int temPin = A5;
//Set buzzer to pin 13 as OUTPUT
int buzzer = 13;
//Variable to store the temperature read
int value;
void setup()
{
//Initialize Serial baud rate to 9600
Serial.begin(9600);
//sets buzzer as an OUTPUT
pinMode(buzzer, OUTPUT);
}
```

Interfacing Temperature Sensor



```
void loop()
{
//Read temperature value on pin A5 by analogRead() method
value = analogRead(temPin);
//Conversion of temperature value read
float mvalue = ( value/1024.0)*5000;
//Conversion of Temperature to celsius
float celsius = mvalue/10;
//conversion of temperature to Fahrenheit
float fahrenheit = (celsius*9)/5 + 32;
//print the celsius value onto the serial monitor
Serial.print(cel);
//check if the read temperature is greater than 32 degree celsius
if(cel>32)
{
//trigger HIGH value on buzzer
digitalWrite(buzzer, HIGH);
delay(1000);
```

Interfacing Temperature Sensor



```
// trigger LOW value on buzzer
digitalWrite(buzzer, LOW);
//delay for 2 second
delay(2000);
//trigger HIGH value on buzzer
digitalWrite(buzzer, HIGH);
//delay for 1 second
delay(1000);
// trigger LOW value on buzzer
digitalWrite(buzzer, LOW);
//delay for 2 second
delay(2000);
}
//Print the temperature onto a serial monitor
Serial.print("TEMPRATURE = ");
Serial.print(cel);
Serial.print("*C");
Serial.println(); }
```

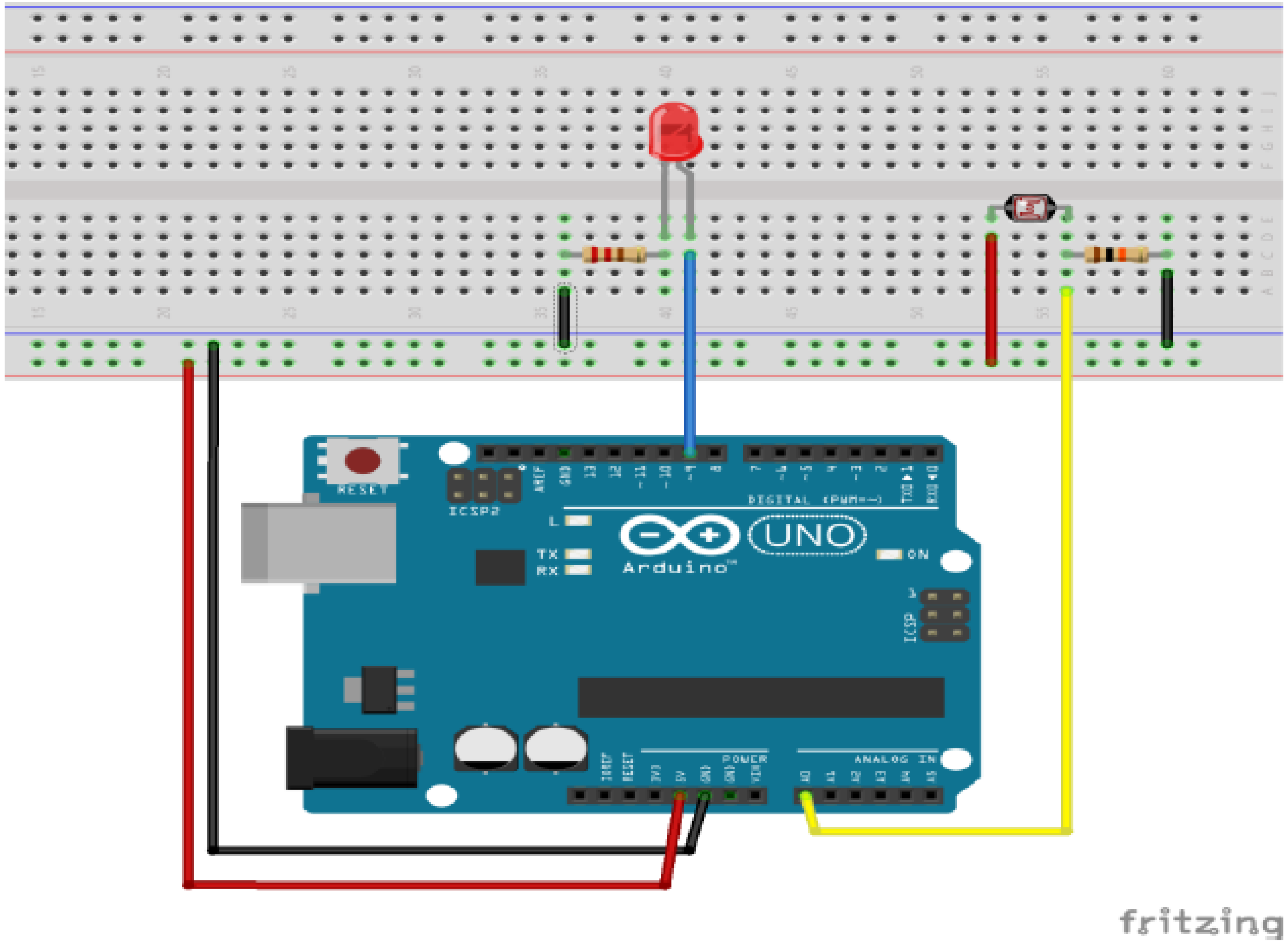
Automatic lights with light sensor

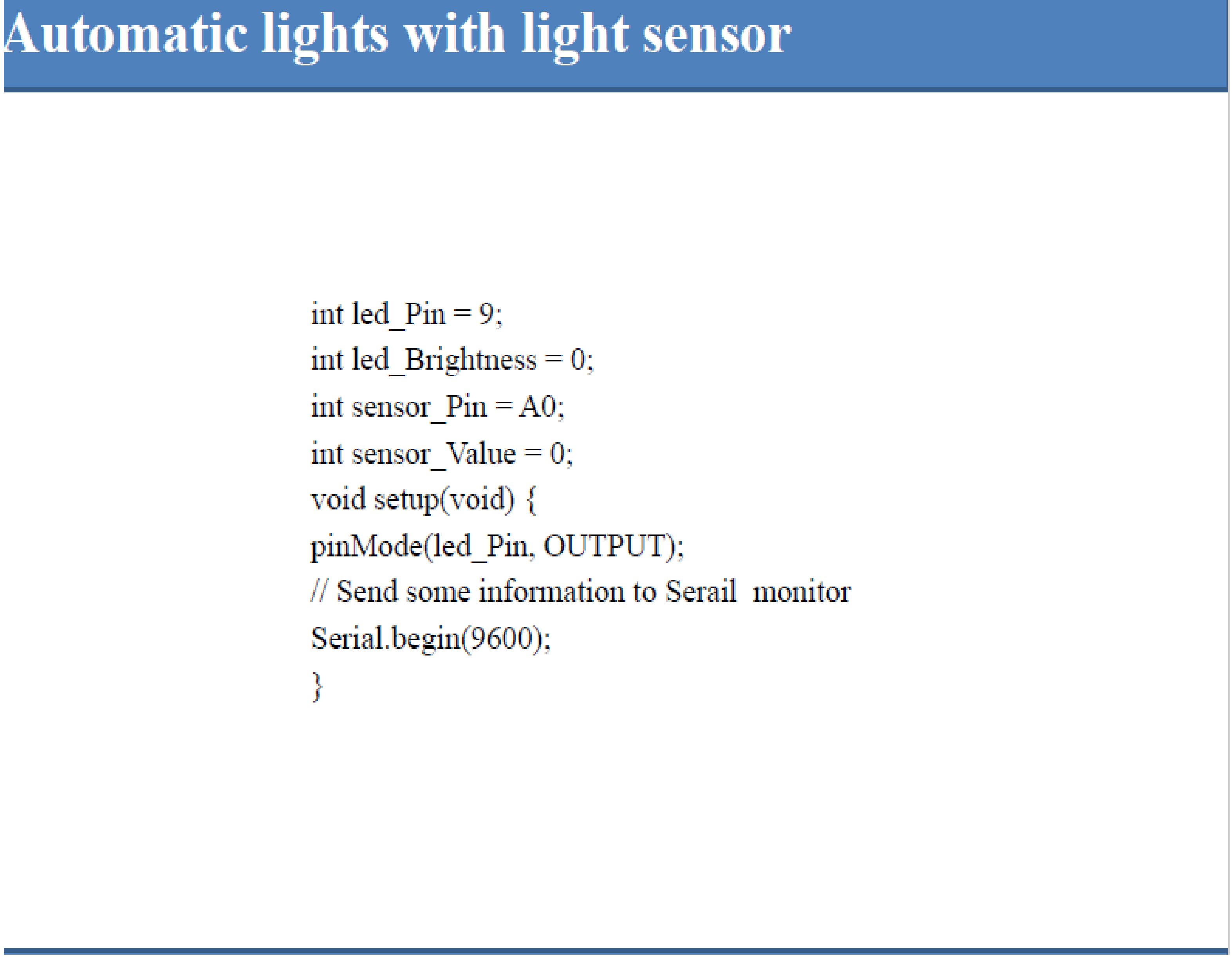
Components Required	**1x LED , 1x 220Ω resistor , 1x photoresistor , 1x 10kΩ resistor, Jumper wires, Breadboard**

A **photoresistor** is a light-dependent resistor. The resistance of a photoresistor decreases with increasing of light intensity. So:

- When there is light, the resistance decreases, we will have more current flowing.
- When there is no light, the resistor increases, we will have less current flowing.

Automatic lights with light sensor

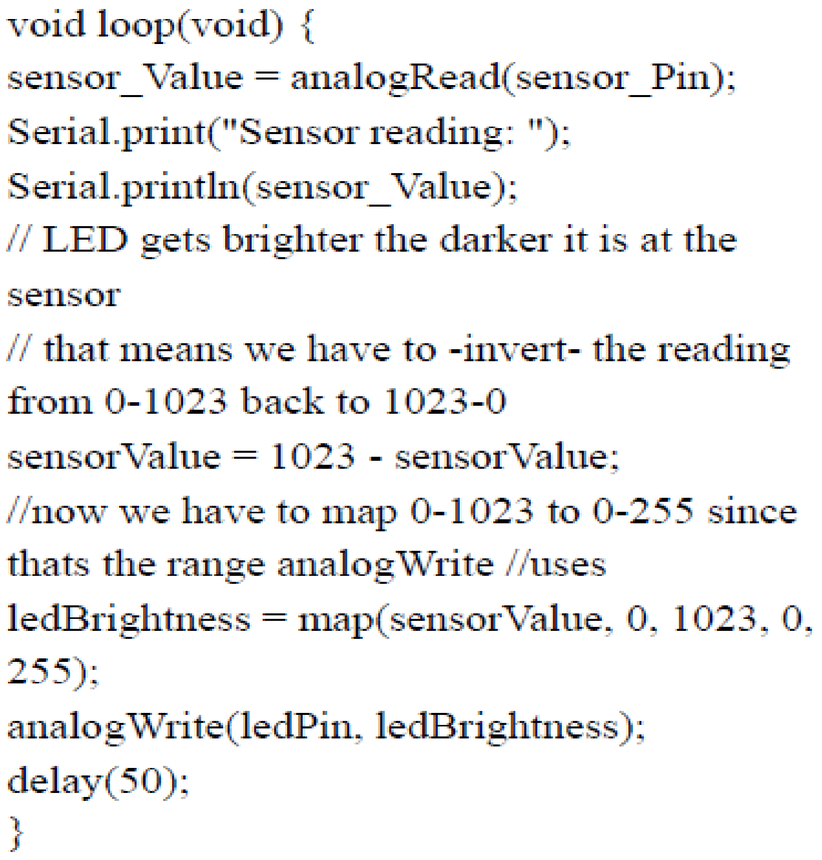




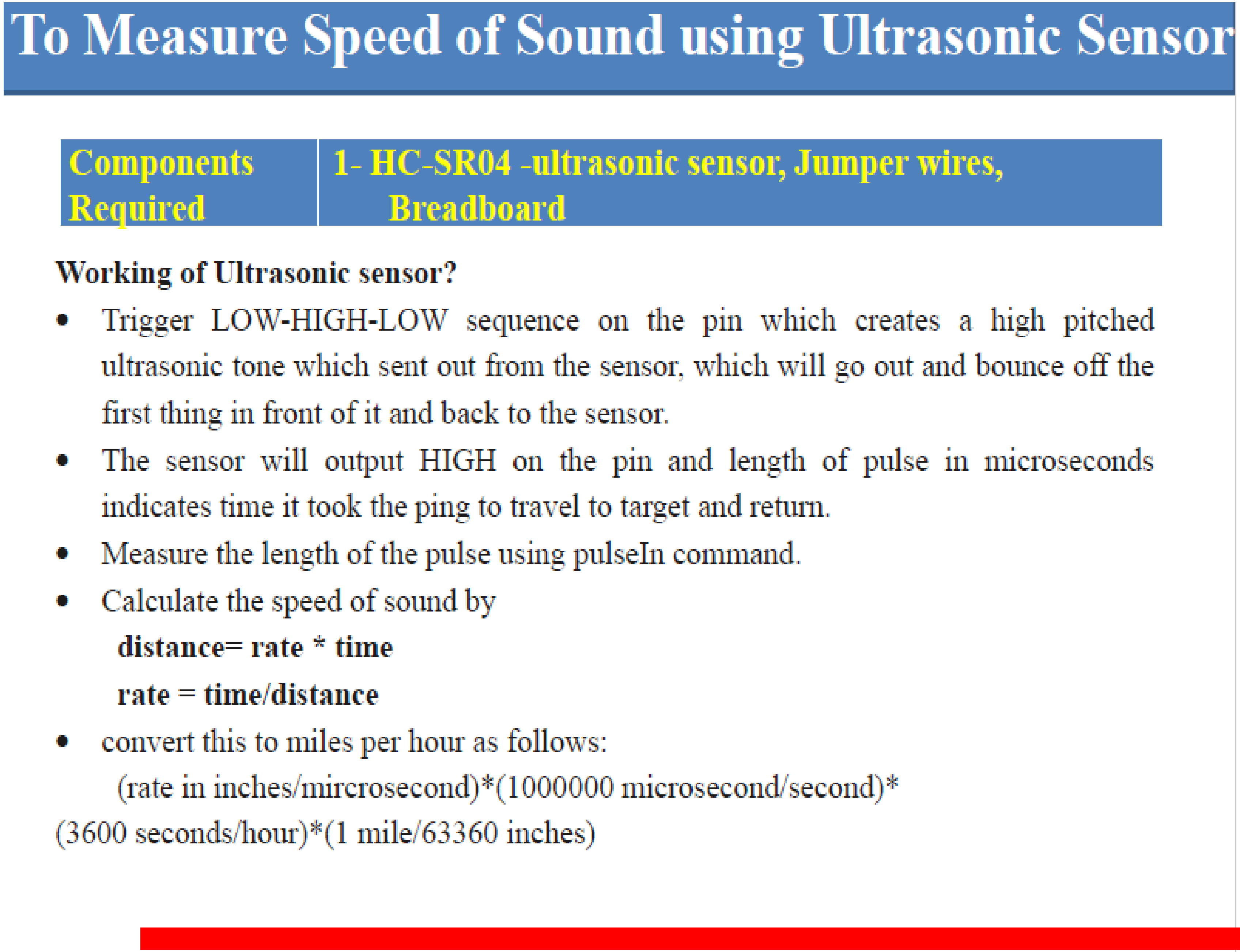
Automatic lights with light sensor

```
int led_Pin = 9;
int led_Brightness = 0;
int sensor_Pin = A0;
int sensor_Value = 0;
void setup(void) {
pinMode(led_Pin, OUTPUT);
// Send some information to Serail  monitor
Serial.begin(9600);
}
```

Automatic lights with light sensor



```
void loop(void) {
sensor_Value = analogRead(sensor_Pin);
Serial.print("Sensor reading: ");
Serial.println(sensor_Value);
// LED gets brighter the darker it is at the
sensor
// that means we have to -invert- the reading
from 0-1023 back to 1023-0
sensorValue = 1023 - sensorValue;
//now we have to map 0-1023 to 0-255 since
thats the range analogWrite //uses
ledBrightness = map(sensorValue, 0, 1023, 0,
255);
analogWrite(ledPin, ledBrightness);
delay(50);
}
```



To Measure Speed of Sound using Ultrasonic Sensor

Components Required	**1- HC-SR04 -ultrasonic sensor, Jumper wires, Breadboard**

Working of Ultrasonic sensor?

- Trigger LOW-HIGH-LOW sequence on the pin which creates a high pitched ultrasonic tone which sent out from the sensor, which will go out and bounce off the first thing in front of it and back to the sensor.
- The sensor will output HIGH on the pin and length of pulse in microseconds indicates time it took the ping to travel to target and return.
- Measure the length of the pulse using pulseIn command.
- Calculate the speed of sound by

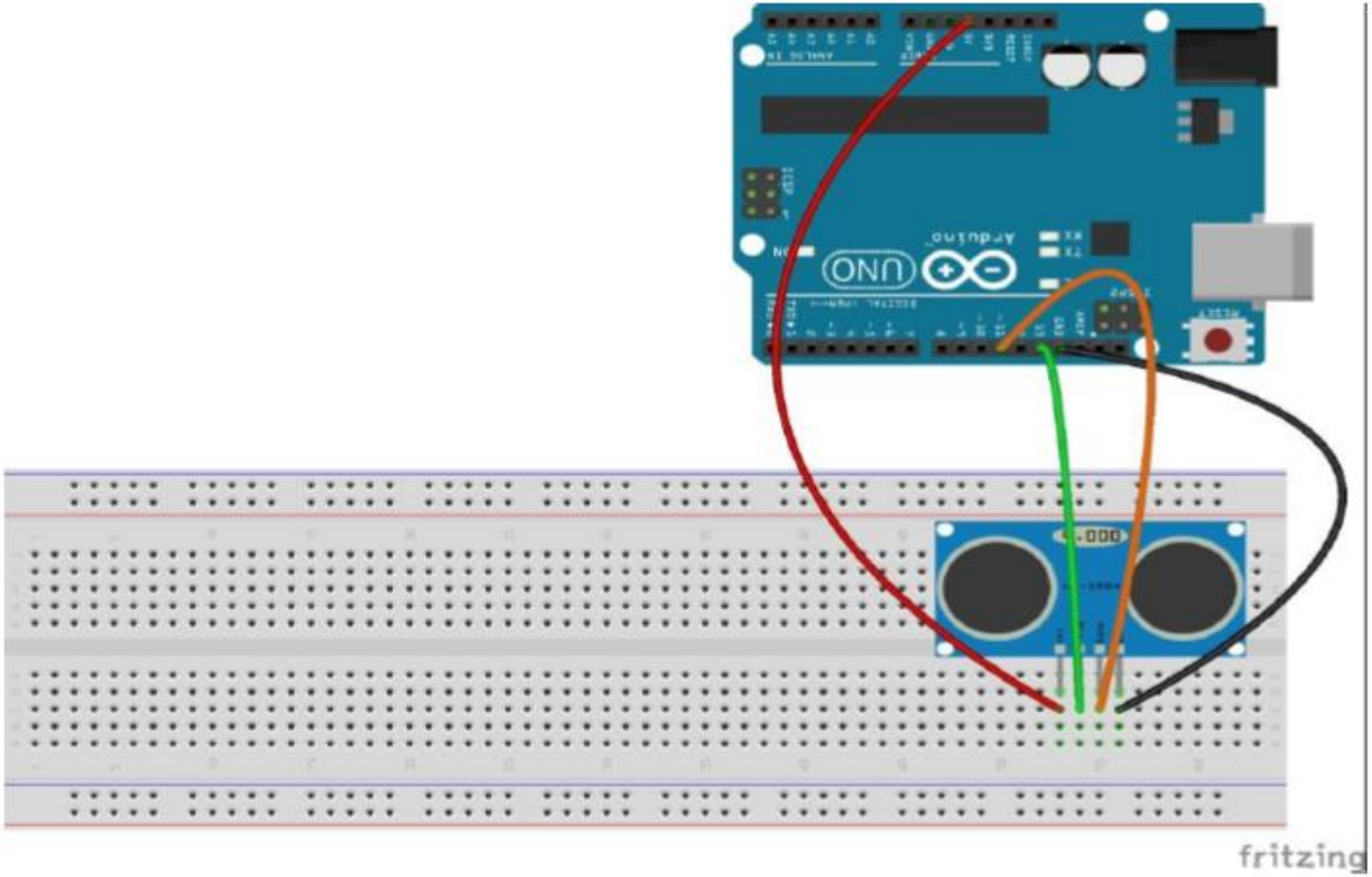
  distance= rate * time

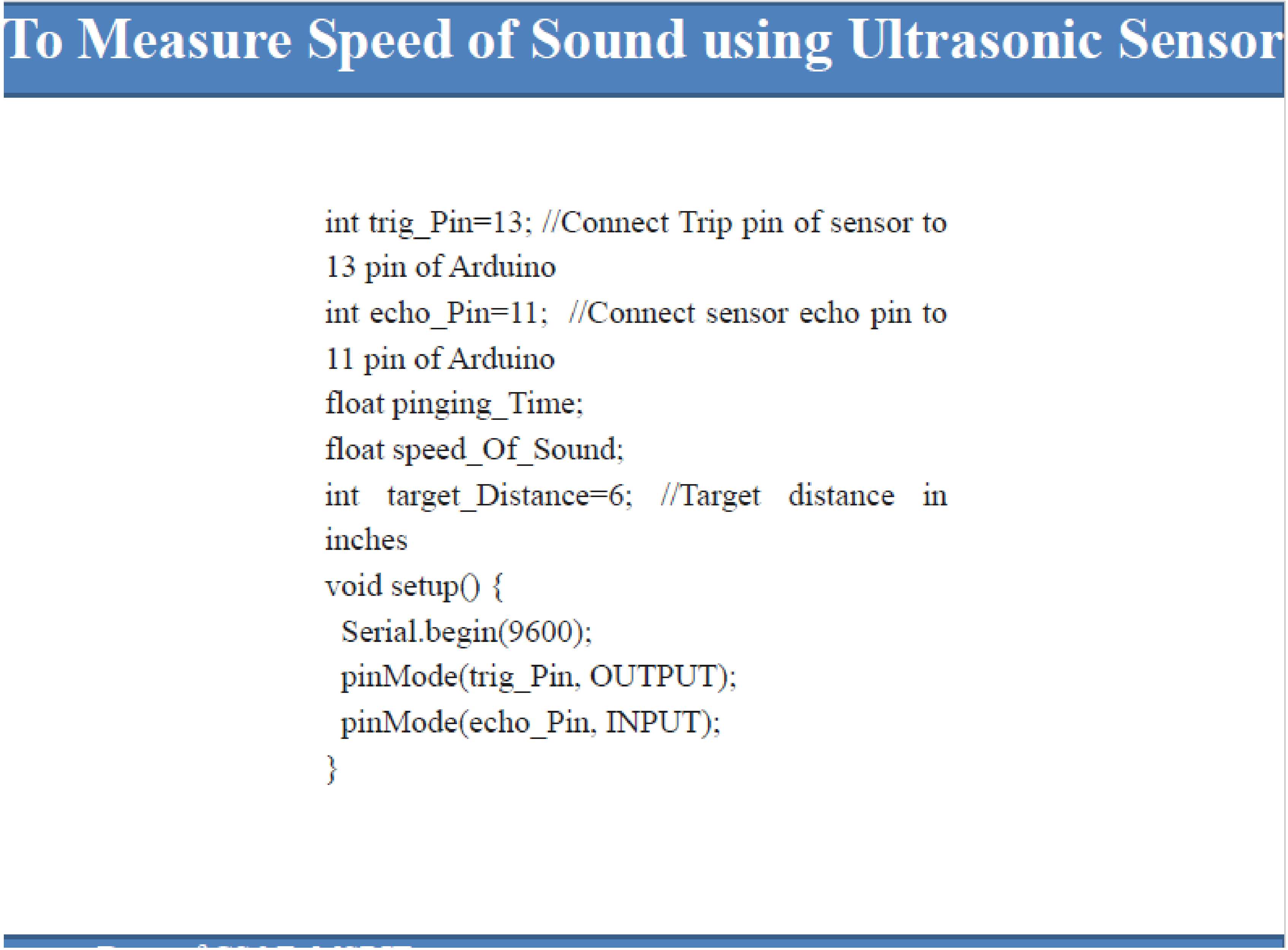
  rate = time/distance
- convert this to miles per hour as follows:

  (rate in inches/mircrosecond)*(1000000 microsecond/second)*

(3600 seconds/hour)*(1 mile/63360 inches)

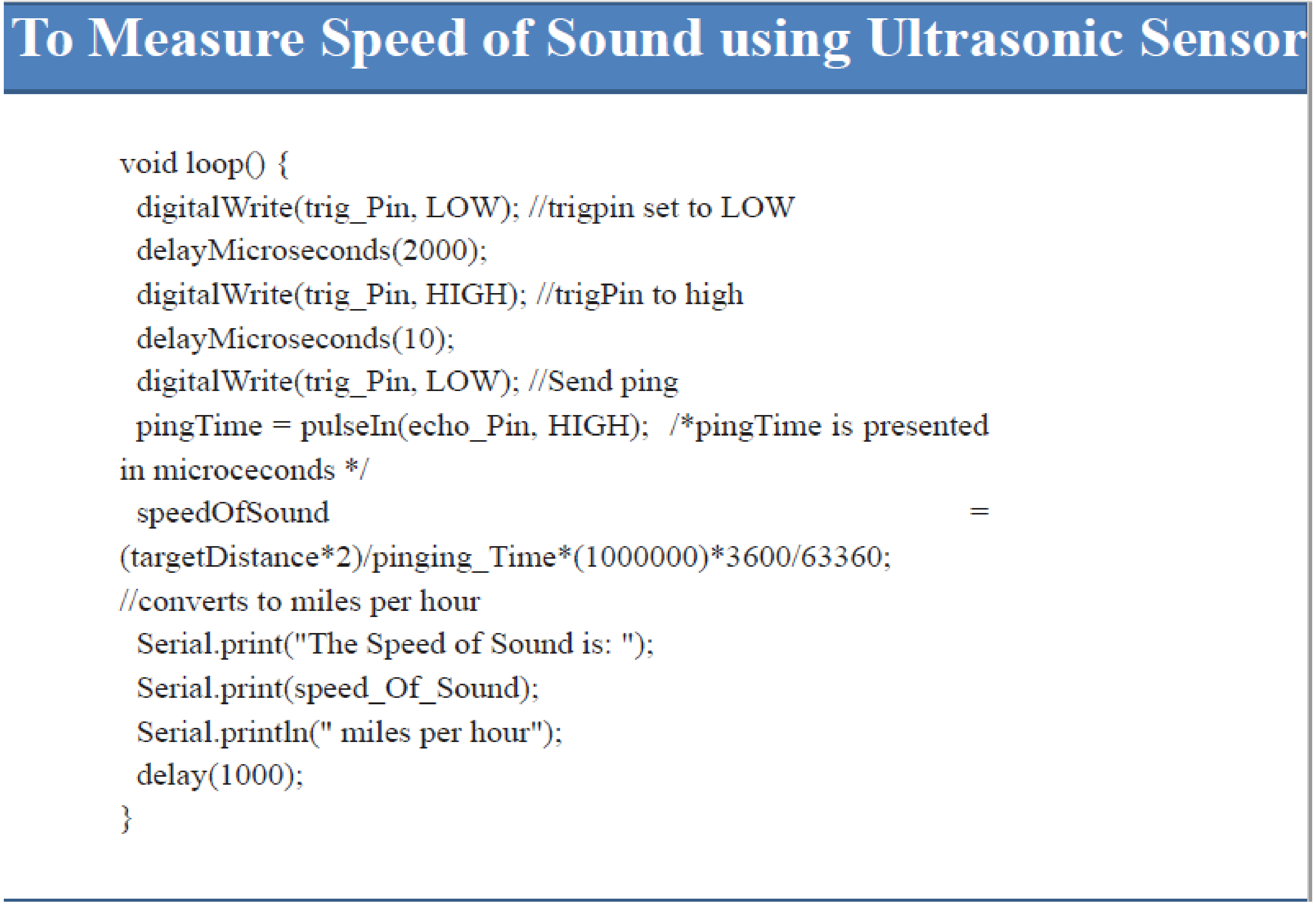
To Measure Speed of Sound using Ultrasonic Sensor





To Measure Speed of Sound using Ultrasonic Sensor

```
int trig_Pin=13; //Connect Trip pin of sensor to
13 pin of Arduino
int echo_Pin=11;  //Connect sensor echo pin to
11 pin of Arduino
float pinging_Time;
float speed_Of_Sound;
int  target_Distance=6;  //Target  distance  in
inches
void setup() {
 Serial.begin(9600);
 pinMode(trig_Pin, OUTPUT);
 pinMode(echo_Pin, INPUT);
}
```



To Measure Speed of Sound using Ultrasonic Sensor

```
void loop() {
  digitalWrite(trig_Pin, LOW); //trigpin set to LOW
  delayMicroseconds(2000);
  digitalWrite(trig_Pin, HIGH); //trigPin to high
  delayMicroseconds(10);
  digitalWrite(trig_Pin, LOW); //Send ping
  pingTime = pulseIn(echo_Pin, HIGH);  /*pingTime is presented
in microceconds */
  speedOfSound                                        =
(targetDistance*2)/pinging_Time*(1000000)*3600/63360;
//converts to miles per hour
  Serial.print("The Speed of Sound is: ");
  Serial.print(speed_Of_Sound);
  Serial.println(" miles per hour");
  delay(1000);
}
```