EXPERIMENT No: 01

TITLE: LED BLINKING.

AIM: Interface Light Emitting Diode (LED) to Arduino UNO board & write the program for blinking LED with a specified delay.

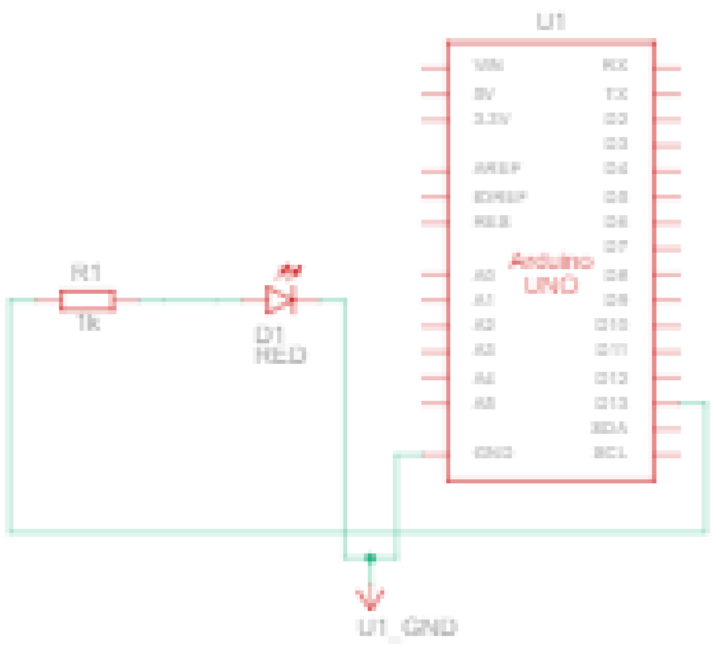
PREREQUISITES:

1. Knowledge of Arduino Uno Board interfaces.
2. Tool- Arduino IDE.

OBJECTIVES:

1. To connect & operate LED connected to digital outputs of an Arduino.
2. To understand concept of Interfacing with microcontroller.
3. To understand requirement of microcontroller for interfacing any external devices.

SCHEMATIC DIAGRAM:

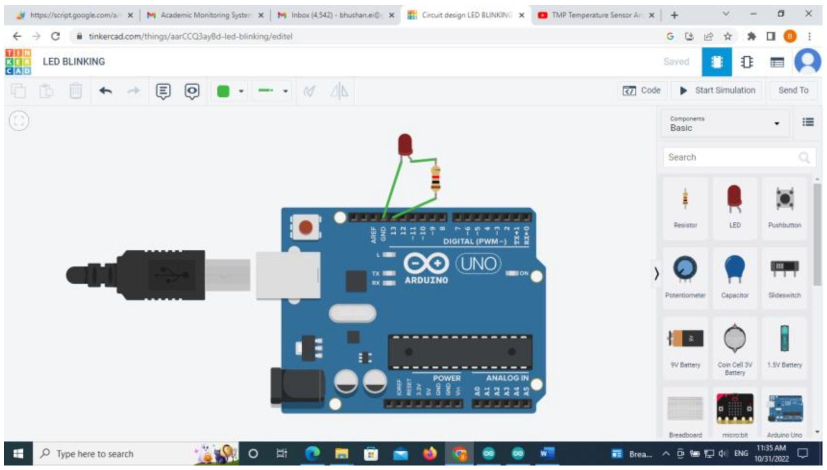


PROGRAM:

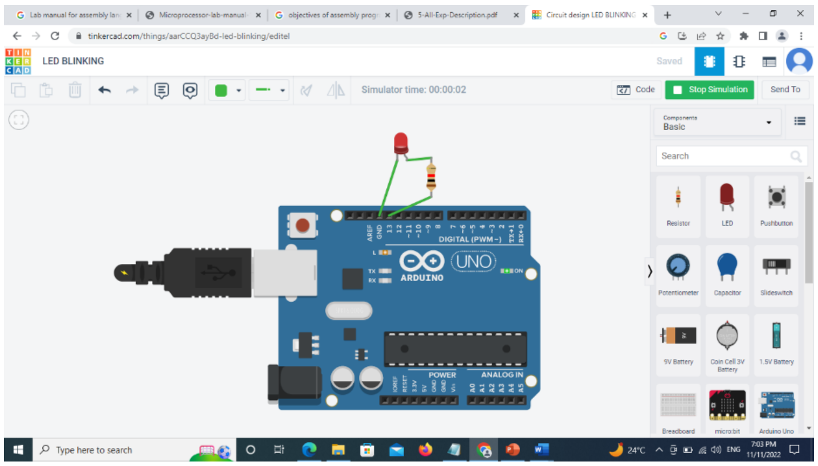
```
int led=13;
void setup()
{
  pinMode(led, OUTPUT);
}

void loop() {
  digitalWrite(led, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);               // wait for a second
  digitalWrite(led, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);               // wait for a second
}
```

OUTPUT: LED OFF



OUTPUT: LED ON



CONCLUSION:

EXPERIMENT No: 02

TITLE: PIR SENSOR INTERFACING.

AIM: Interface the PIR sensor with Arduino UNO board and write the program to control the LED (ON/OFF) on motion and play the buzzer on detection of object.

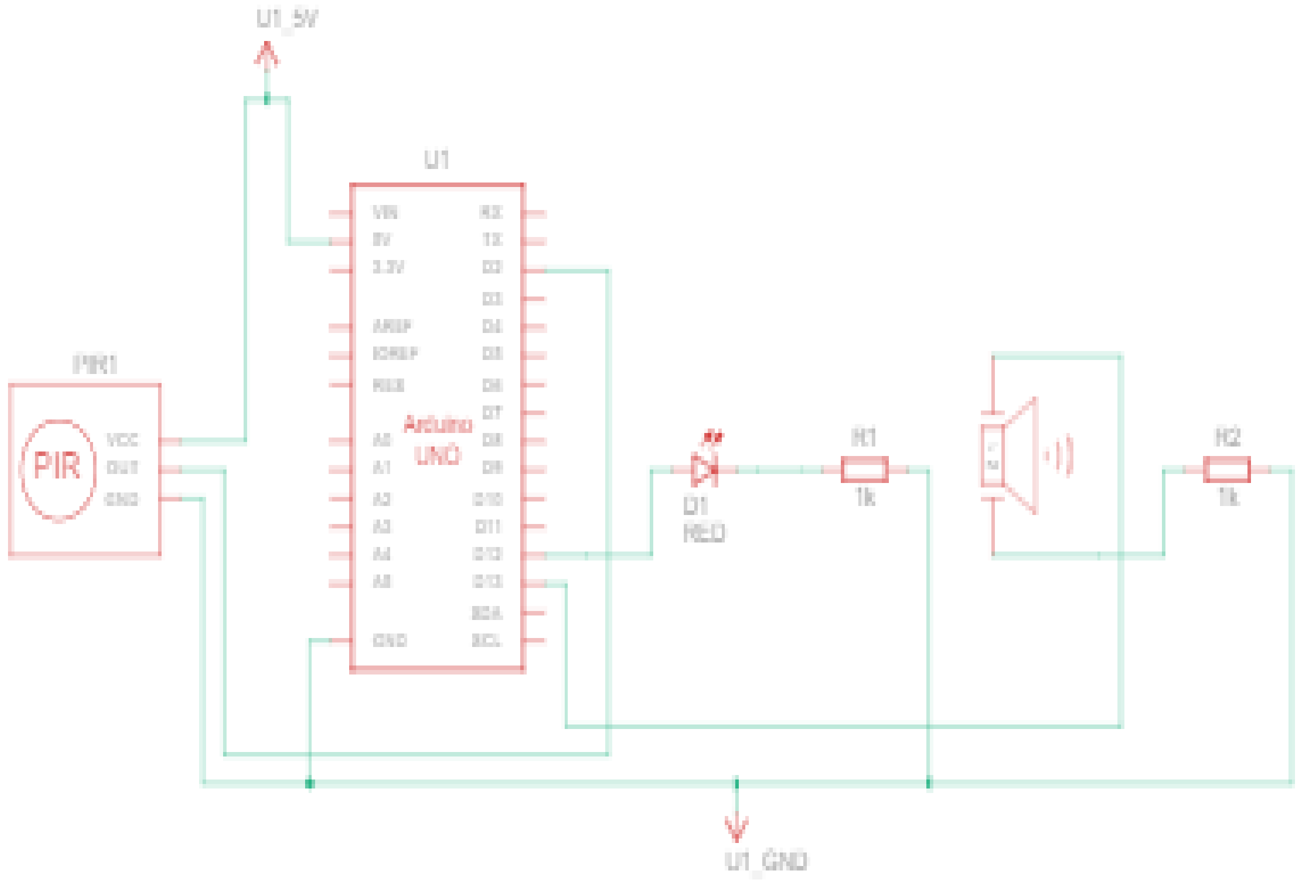
PREREQUISITES:

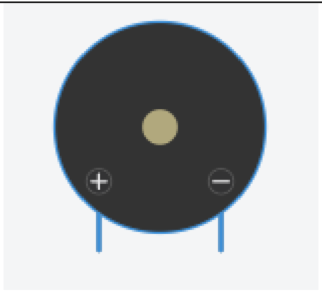
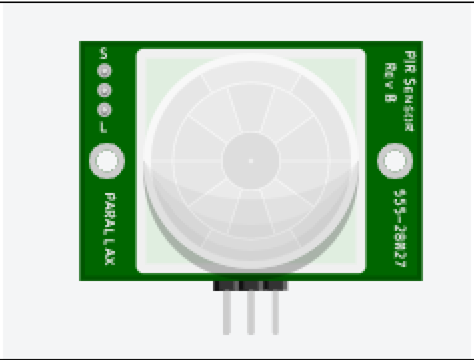
1. Knowledge of Arduino Uno Board interfaces.
2. Tool- Arduino IDE.

OBJECTIVES:

1. To connect & operate PIR sensor connected to pins of an Arduino.
2. To understand concept of Interfacing with microcontroller.
3. To understand requirement of microcontroller for interfacing any external devices.

SCHEMATIC DIAGRAM:



	
Buzzer + pin connected to Pin 13 - Pin connected through resistor to GND	PIR Sensor Signal pin connected to pin 12 Power pin connected to 5V GND pin connected to GND

PROGRAM:

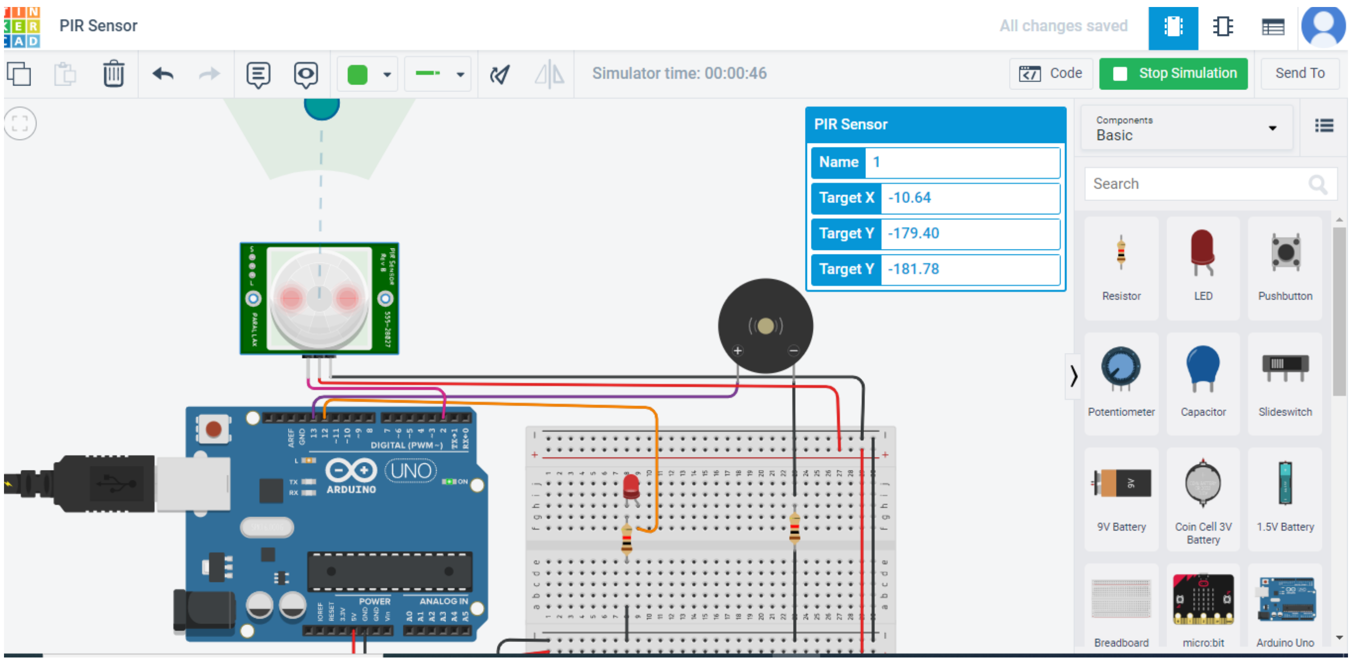
```
int pinsensor = 2;
int pinled = 12;
int pinbuzzer = 13;
int pirsensor=0;

void setup()
{
  pinMode(pinsensor, INPUT);
  pinMode(pinled, OUTPUT);
  pinMode(pinbuzzer, OUTPUT);
}

void loop()
{
    pirsensor = digitalRead(pinsensor);
  if (pirsensor == HIGH)
  {
    digitalWrite(pinled, HIGH);
    tone(pinbuzzer,1000,500);
  }

else
   {
       digitalWrite(pinled, LOW);
  }
delay(10);
}
```

OUTPUT: When Motion is detected.



PIR Sensor

Name	1
Target X	-10.64
Target Y	-179.40
Target Y	-181.78

CONCLUSION:

EXPERIMENT No: 03

TITLE: ULTRASONIC SENSOR INTERFAICNG.

AIM: Interface ultrasonic sensor with Arduino UNO board & write the program to measure and display the distance on serial monitor.

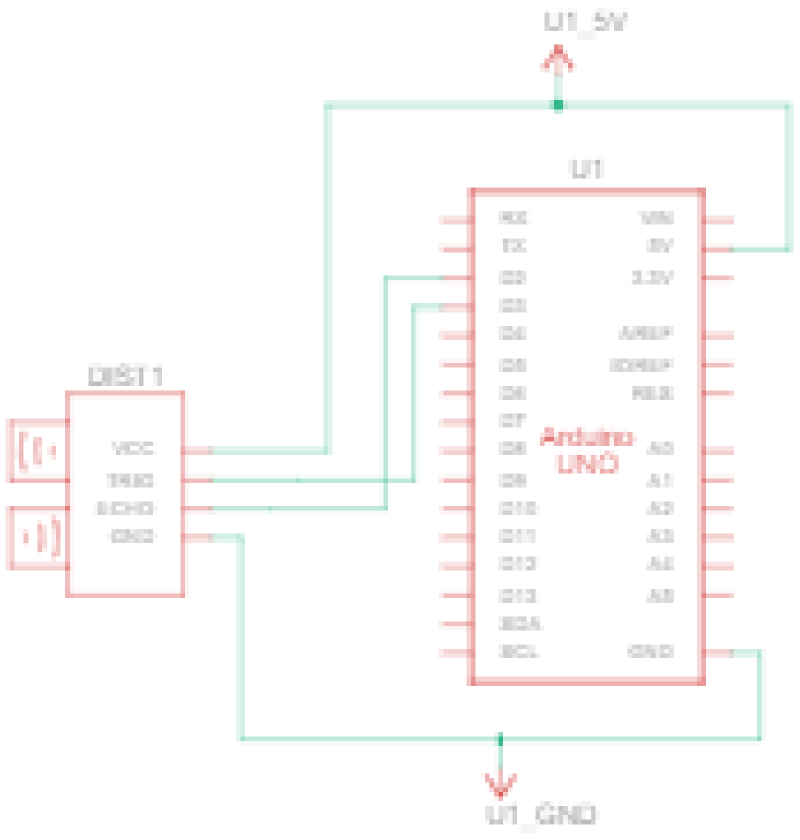
PREREQUISITES:

1. Knowledge of Arduino Uno Board interfaces.
2. Tool- Arduino IDE.

OBJECTIVES:

1. To connect & operate Ultrasonic sensor connected to pins of an Arduino.
2. To understand concept of Interfacing with microcontroller.
3. To understand requirement of microcontroller for interfacing any external devices.

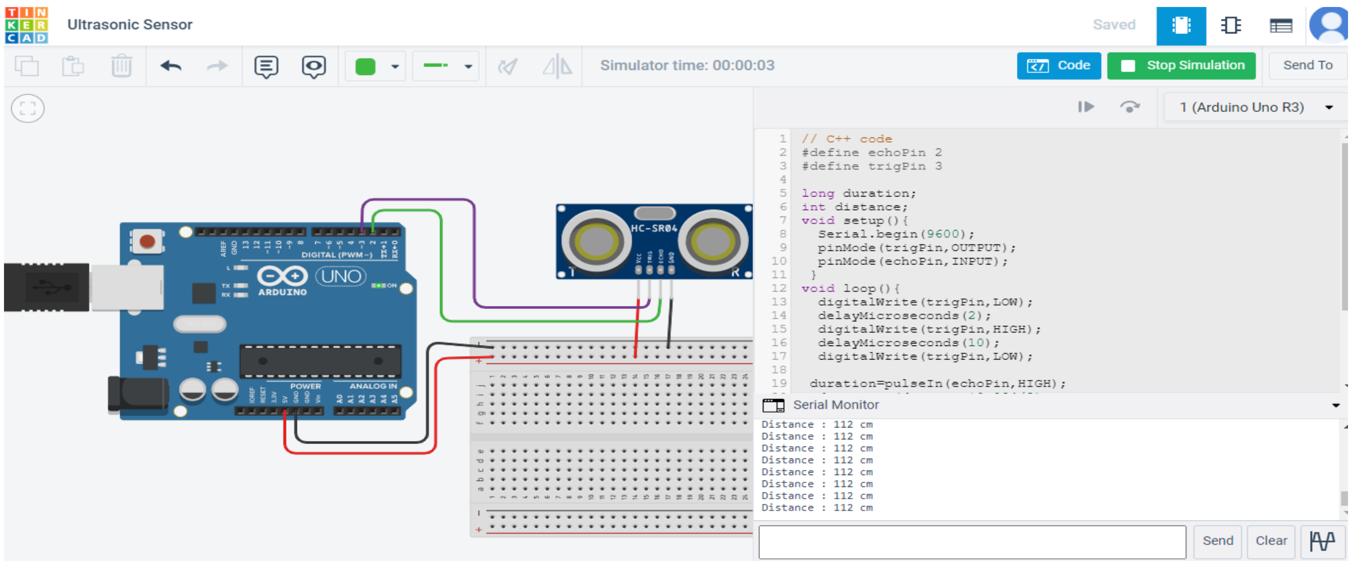
SCHEMATIC DIAGRAM:



PROGRAM:

```
#define echoPin 2
#define trigPin 3
long duration;
int distance;
void setup()
{
  Serial.begin(9600);
  pinMode(trigPin,OUTPUT);
  pinMode(echoPin,INPUT);
 }
void loop()
{
  digitalWrite(trigPin,LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin,HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin,LOW);
    duration=pulseIn(echoPin,HIGH);
  distance=(duration*0.034/2);
  Serial.print("Distance : ");
  Serial.print(distance);
  Serial.println(" cm ");
  delay(1000);
  }
```

OUTPUT:



```
// C++ code
#define echoPin 2
#define trigPin 3

long duration;
int distance;
void setup(){
  Serial.begin(9600);
  pinMode(trigPin,OUTPUT);
  pinMode(echoPin,INPUT);
 }
void loop(){
  digitalWrite(trigPin,LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin,HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin,LOW);

 duration=pulseIn(echoPin,HIGH);
```

```
Distance : 112 cm
Distance : 112 cm
Distance : 112 cm
Distance : 112 cm
Distance : 112 cm
Distance : 112 cm
Distance : 112 cm
Distance : 112 cm
```

CONCLUSION:

EXPERIMENT No: 04

TITLE: TEMPERATURE SENSOR INTERFAICNG.

AIM: Interface Temperature sensor with Arduino UNO board & write the program to measure temperature and display it on serial monitor.

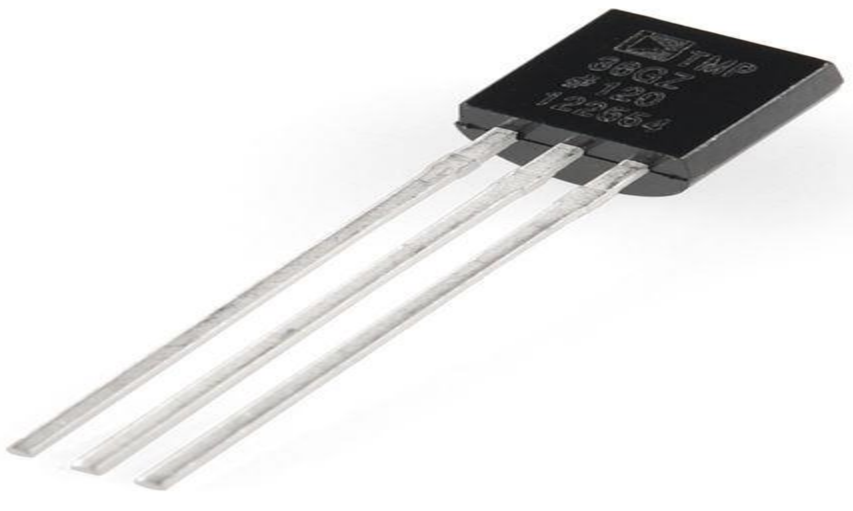
PREREQUISITES:

1. Knowledge of Arduino Uno Board interfaces.
2. Tool- Arduino IDE.

OBJECTIVES:

1. To connect & operate Ultrasonic sensor connected to pins of an Arduino.
2. To understand concept of Interfacing with microcontroller.
3. To understand requirement of microcontroller for interfacing any external devices.

THEORY:

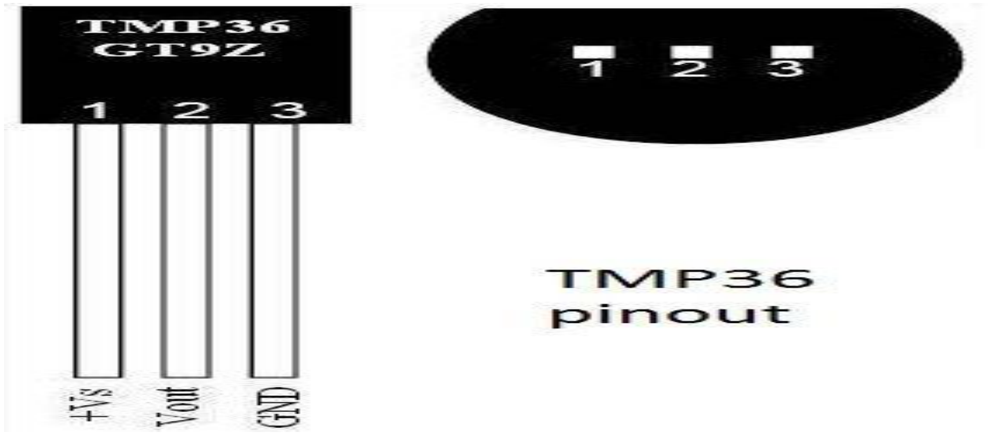


The TMP35/TMP36/TMP37 are low voltage, precision centigrade temperature sensors. They provide a voltage output that is linearly proportional to the Celsius (centigrade) temperature. The TMP35/ TMP36/TMP37 do not require any external calibration to provide typical accuracies of ±1°C at +25°C and ±2°C over the −40°C to +125°C temperature range.

In this tutorial you will learn how to use the TMP36 sensor with Arduinouno. The room temperature will be printed to serial monitor.

Before we start let's see more information about this sensor.

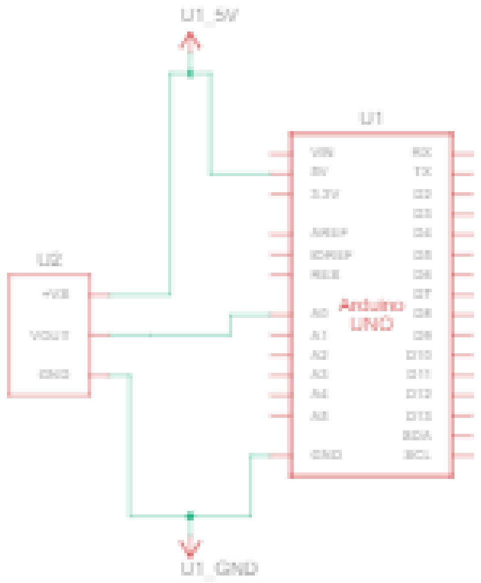
About TMP36 Sensor



Features:

- Low Voltage Operation (+2.7 V to+5.5 V)
- Calibrated Directly in °C
- 10 mV/8°C Scale Factor (20 mV/8°C on TMP37)
- ±2°C Accuracy OverTemperature (typ)
- ±0.5°C Linearity (typ)
- Stable with Large Capacitive Loads
- Specified -40 °C to +125 °C, Operation to +150 °C
- Less than 50 µA Quiescent Current
- Shutdown Current 0.5 µA max

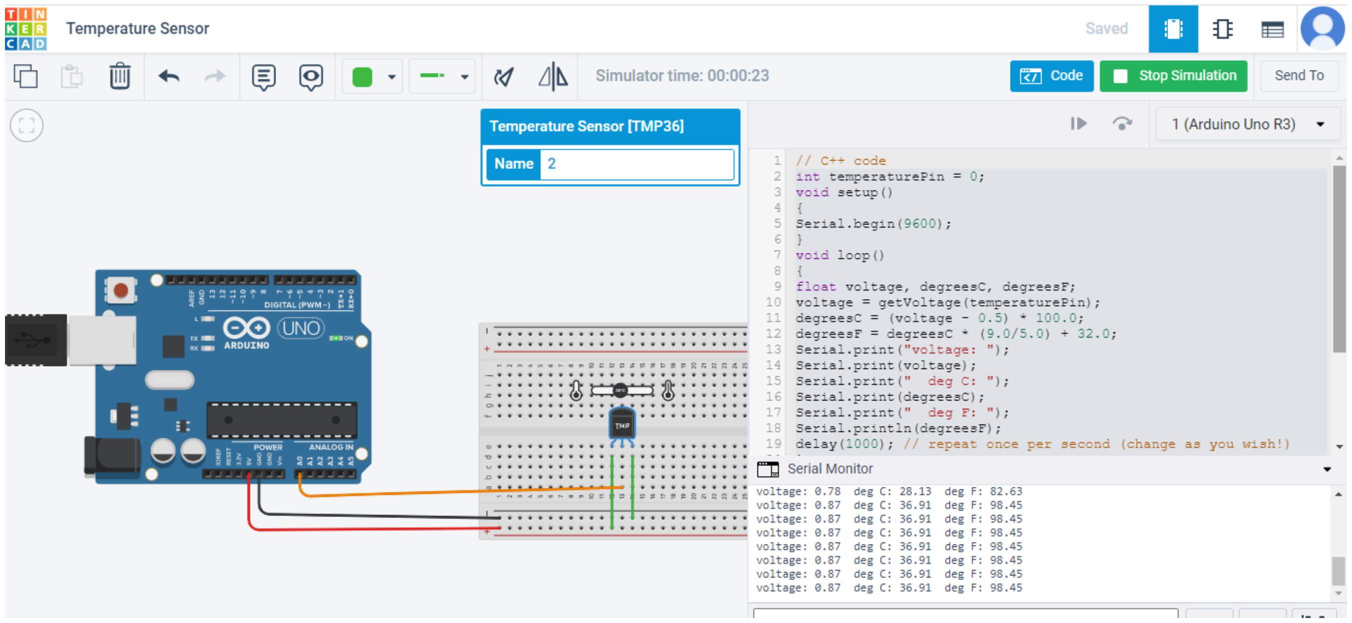
SCHEMATIC DIAGRAM:



PROGRAM:

```
int temperaturePin = 0;
void setup()
{
Serial.begin(9600);
}
void loop()
{
float voltage, degreesC, degreesF;
voltage = getVoltage(temperaturePin);
degreesC = (voltage - 0.5) * 100.0;
degreesF = degreesC * (9.0/5.0) + 32.0;
Serial.print("voltage: ");
Serial.print(voltage);
Serial.print("  deg C: ");
Serial.print(degreesC);
Serial.print("  deg F: ");
Serial.println(degreesF);
delay(1000); // repeat once per second (change as you wish!)
}
float getVoltage(int pin)
{
return (analogRead(pin) * 0.004882814);
 }
```

OUTPUT:



```
// C++ code
int temperaturePin = 0;
void setup()
{
Serial.begin(9600);
}
void loop()
{
float voltage, degreesC, degreesF;
voltage = getVoltage(temperaturePin);
degreesC = (voltage - 0.5) * 100.0;
degreesF = degreesC * (9.0/5.0) + 32.0;
Serial.print("voltage: ");
Serial.print(voltage);
Serial.print("  deg C: ");
Serial.print(degreesC);
Serial.print("  deg F: ");
Serial.println(degreesF);
delay(1000); // repeat once per second (change as you wish!)
```

```
voltage: 0.78  deg C: 28.13  deg F: 82.63
voltage: 0.87  deg C: 36.91  deg F: 98.45
voltage: 0.87  deg C: 36.91  deg F: 98.45
voltage: 0.87  deg C: 36.91  deg F: 98.45
voltage: 0.87  deg C: 36.91  deg F: 98.45
voltage: 0.87  deg C: 36.91  deg F: 98.45
voltage: 0.87  deg C: 36.91  deg F: 98.45
voltage: 0.87  deg C: 36.91  deg F: 98.45
```

CONCLUSION: