[INTERNET OF THINGS]

Practical-4

-:AIM:-

Arduino programming with serial monitor, Temperature sensor.

Submitted By: Dharmay Sureja

Enrollment No:17012011056

GANPAT UNIVERSITY

U. V. Patel College of Engineering

Computer Engineering Department

AIM:- Arduino programming with serial monitor, Temperature sensor.

Auduino function

- **Serial.begin() :**
    - Sets the data rate in bits per second (baud) for serial data transmission.
    - For communicating with Serial Monitor, make sure to use one of the baud rates listed in the menu at the bottom right corner of its screen.
    - An optional second argument configures the data, parity, and stop bits. The default is 8 data bits, no parity, one stop bit.
    - Syntax - Serial.begin(speed)
      - Serial.begin(speed, config)
    - Parameters
      speed: in bits per second (baud). (Allowed data types : long.)
      config: sets data, parity, and stop bits.
    - Returns - Nothing

- **Serial.end() :**
    - Reads the value from a specified digital pin, either HIGH or LOW.
    - Disables serial communication, allowing the RX and TX pins to be used for general input and output.
    - Syntax - Serial.end()
    - Parameters
      Serial: serial port object. See the list of available serial ports for each board on the Serial main page.
    - Returns - Nothing

- **Serial.read() :**
    - Reads incoming serial data.
    - Syntax - Serial.read()
    - Parameters
      Serial: serial port object. See the list of available serial ports for each board on the Serial main page.
    - Returns - The first byte of incoming serial data available (or -1 if no data is available).
      Data type: int.

- **Serial.write() :**
  - Writes binary data to the serial port. This data is sent as a byte or series of bytes.
  - Syntax - Serial.write(val)
    - Serial.write(str)
    - Serial.write(buf, len)
  - Parameters

    Serial: serial port object. See the list of available serial ports for each board on the Serial main page.

    val: a value to send as a single byte.

    str: a string to send as a series of bytes.

    buf: an array to send as a series of bytes.

    len: the number of bytes to be sent from the array.
  - Returns - the number of bytes written, though reading that number is optional.

- **Serial.print() :**
  - Prints data to the serial port as human-readable ASCII text. This command can take many forms. Numbers are printed using an ASCII character for each digit. Floats are similarly printed as ASCII digits, defaulting to two decimal places. Bytes are sent as a single character. Characters and strings are sent as is.
  - An optional second parameter specifies the base (format) to use; permitted values are BIN(binary, or base 2), OCT(octal, or base 8), DEC(decimal, or base 10), HEX(hexadecimal, or base 16). For floating point numbers, this parameter specifies the number of decimal places to use. For example-
  - You can pass flash-memory based strings to Serial.print() by wrapping them with F().
  - Syntax - Serial.print(val)
    - Serial.print(val, format)
  - Parameters

    Serial: serial port object. See the list of available serial ports for each board on the Serial main page.

    val: the value to print. Allowed data types: any data type.
  - Returns - print() returns the number of bytes written, though reading that number is optional.

- **Serial.println() :**
  - Prints data to the serial port as human-readable ASCII text followed by a carriage return character (ASCII 13, or '\r') and a newline character (ASCII 10, or '\n'). This command takes the same forms as Serial.print().
  - Syntax - Serial.println(val)
    - Serial.println(val, format)
  - Parameters

    Serial: serial port object. See the list of available serial ports for each board on the Serial main page.

    val: the value to print. Allowed data types: any data type.

    format: specifies the number base (for integral data types) or number of decimal places (for floating point types).
  - Returns - println() returns the number of bytes written, though reading that number is optional.

- **Serial.available() :**
  - Get the number of bytes (characters) available for reading from the serial port. This is data that's already arrived and stored in the serial receive buffer (which holds 64 bytes).
  - Serial.available() inherits from the Stream utility class.
  - Syntax - Serial.available()
  - Parameters

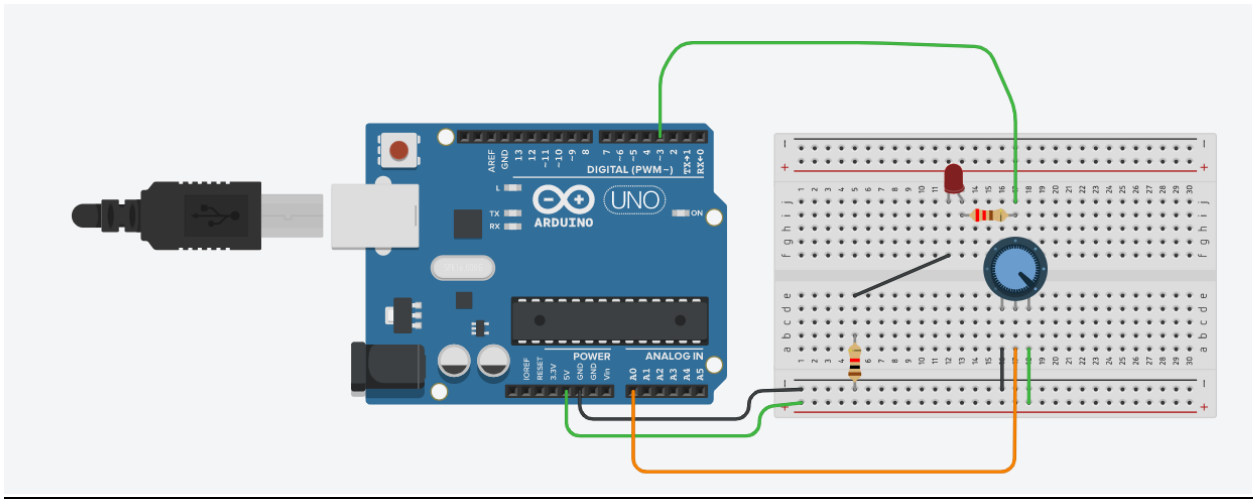
    Serial: serial port object. See the list of available serial ports for each board on the Serial main page.
  - Returns - The number of bytes available to read.

Experiment

1. **Circuit for Increase and decrease the brightness of LED using Potentiometer and display in serial monitor**

Components used : Arduino Uno R3, LED, Potentiometer, Resistor

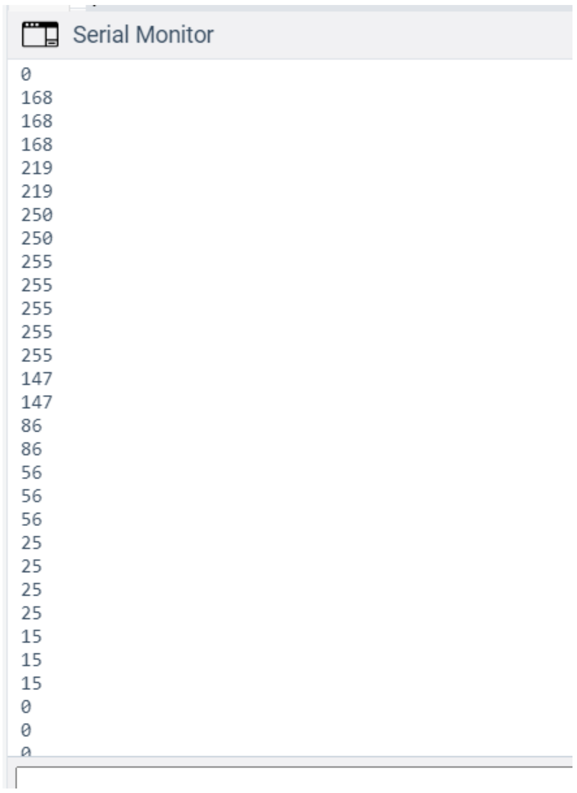
Circuit:



Code:

```
int v=0;

void setup()
{
  pinMode(A0, INPUT);
  pinMode(3, OUTPUT);
  Serial.begin(9600);
}

void loop()
{
  int p_val=analogRead(A0);
  v=map(p_val,0,1023,0,255);
  analogWrite(3,v);
  Serial.println(v);
}
```

Output :

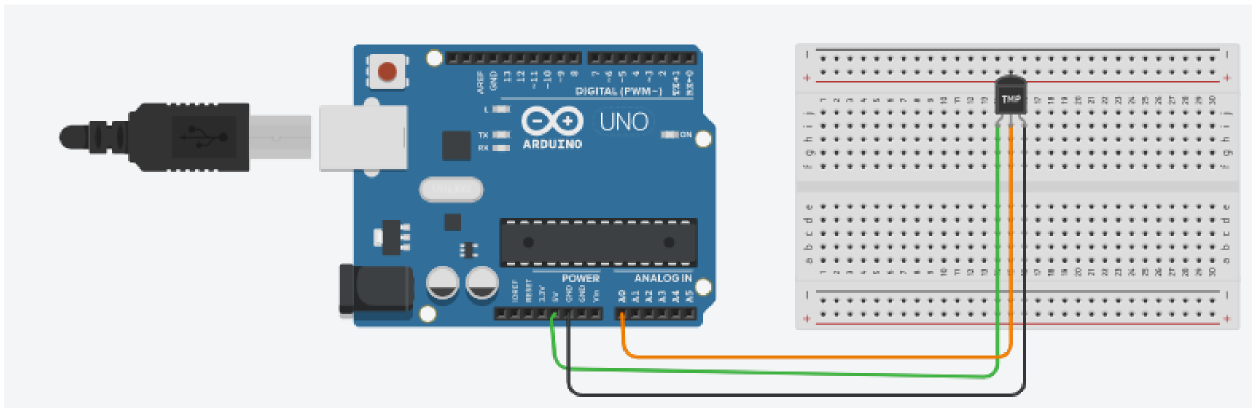


```
0
168
168
168
219
219
250
250
255
255
255
255
255
147
147
86
86
56
56
56
25
25
25
25
15
15
15
0
0
0
```

2. **Read the current temperature of room and display it on serial monitor.**

Components used: Arduino Uno R3, Temperature Sensor(TMP36)

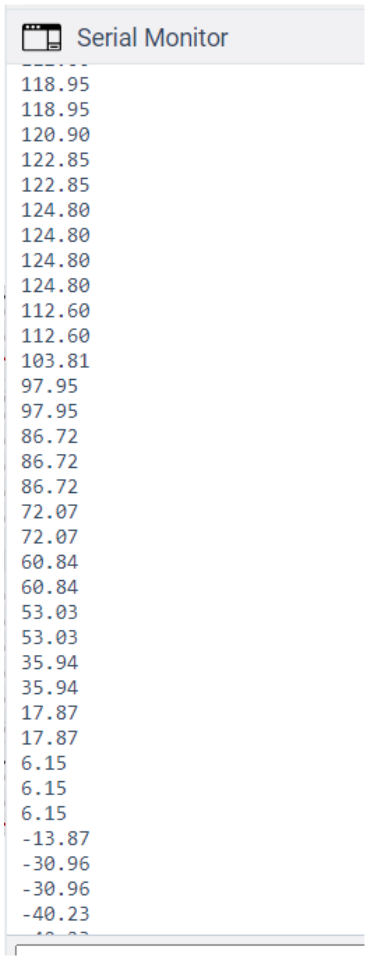
Circuit:



Code:

```
void setup()
{
  pinMode(A0, INPUT);
  Serial.begin(9600);
}

void loop()
{
  float tmp=analogRead(A0);
  tmp=tmp*5000/1024;
  tmp=tmp/10;
  tmp=tmp-50;
  Serial.println(tmp);
}
```

Output :

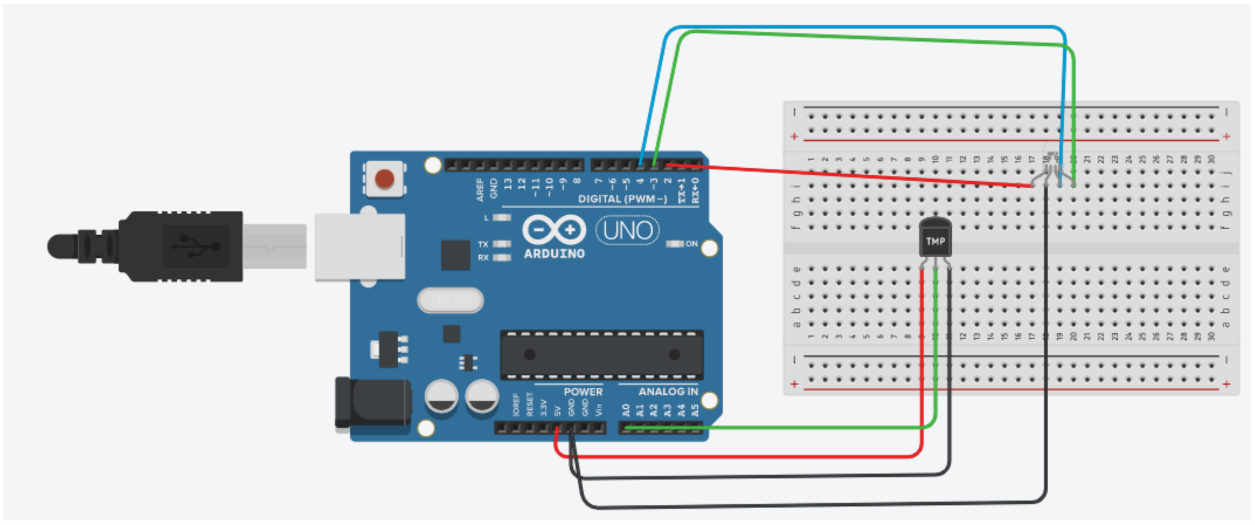


```
118.95
118.95
120.90
122.85
122.85
124.80
124.80
124.80
124.80
112.60
112.60
103.81
97.95
97.95
86.72
86.72
86.72
72.07
72.07
60.84
60.84
53.03
53.03
35.94
35.94
17.87
17.87
6.15
6.15
6.15
-13.87
-30.96
-30.96
-40.23
```

3. **Read the current temperature of room and turn on RGB Led with specific colour according to current temperature value**
   a. If temperature more than 100 Co then turn on RGB LED with RED colour
   b. If temperature between 40 Co to 100 Co then turn on RGB LED with GREEN colour
   c. If temperature less than 40 Co then turn on RGB LED with BLUE colour

Components used : Arduino Uno R3, RGB LED, Temperature Sensor(TMP36)

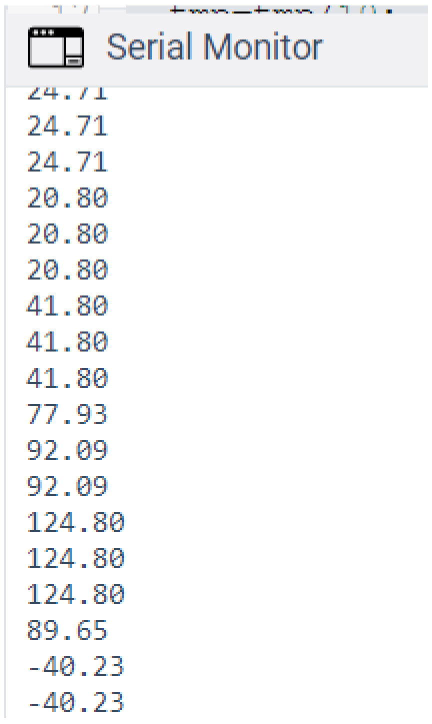
Circuit:



Code:

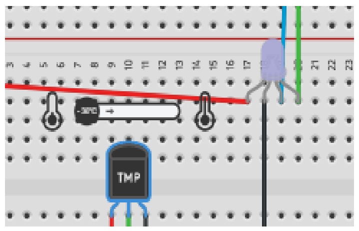
```
void setup()
{

   pinMode(A0, INPUT);
   Serial.begin(9600);
}

void loop()
{
 float tmp=analogRead(A0);
  tmp=tmp*5000/1024;
  tmp=tmp/10;
  tmp=tmp-50;
  Serial.println(tmp);
  if(tmp>=-40 && tmp<=40)
  {
  digitalWrite(2, LOW);
  digitalWrite(3, LOW);
    digitalWrite(4, HIGH);
  }
  else if (tmp>40 && tmp<=100)
  {
   digitalWrite(2, LOW);
  digitalWrite(3, HIGH);
    digitalWrite(4, LOW);
  }
  else if (tmp>100 )
  {
   digitalWrite(2, HIGH);
  digitalWrite(3, LOW);
    digitalWrite(4, LOW);
  }
}
```

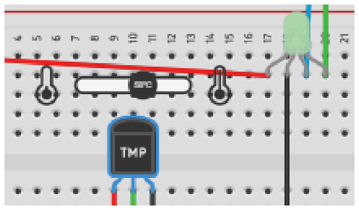
Output :

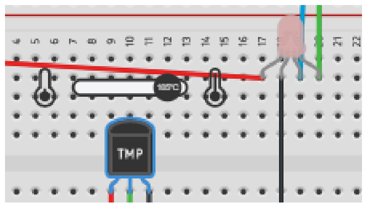


Serial Monitor

```
24.71
24.71
24.71
20.80
20.80
20.80
41.80
41.80
41.80
77.93
92.09
92.09
124.80
124.80
124.80
89.65
-40.23
-40.23
```


TMP


TMP


TMP