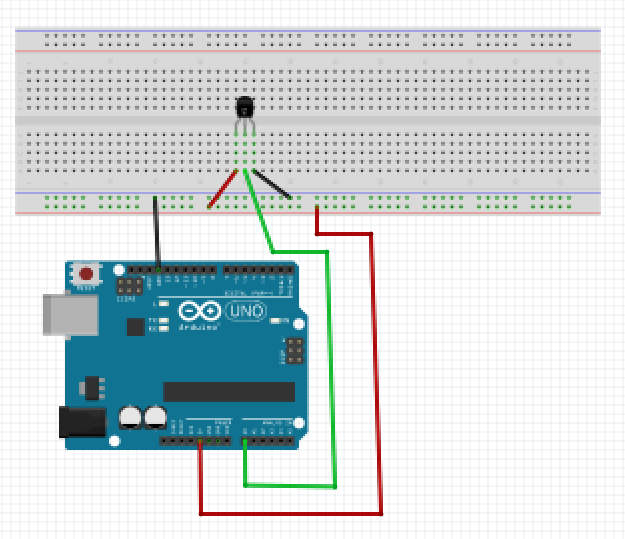
**ARDUINO LED PROJECT**

**Temperature Sensor**



For this project we will need:

* Arduino board.
* Temperature Sensor
* Breadboard.
* 3 Male to Male wires.

Circuit Design:

1. First make sure that the Arduino is powered off (no USB cable plugged to power).
2. Check the temperature sensor, identify the 3 legs on the pin. Identify the flat part of the resin. Observe orientation. Facing the flat part is the orientation used in subsequent steps.
3. Plug the center pin of the temperature sensor to a breadboard horizontal line.. Using a green male jumper wire, connect to the Arduino via pin A0.
4. Plug the left pin of the sensor to the breadboard on a separate horizontal line. Using a red jumper wire connect to the Arduino via pin 5V.
5. Plug the right pin of the sensor to the breadboard on a separate horizontal line. Using a black jumper wire connect to the Arduino via pin GND.

|  |
| --- |
| const int tempPin=A0;  float temp;  void setup()  {  // Put your setup code here, to run once:  Serial.begin(9600);  }  void loop(){  // Put your main code here, to run repeatedly:  temp=analogRead(tempPin);  temp=(5.0\*temp\*1000.0)/(1024\*10);  Serial.println(temp);  delay(500);  } |

const int tempPin = A0;

First, we create a reference for the analog pin we intend to use and define a variable for that pin number. This will enable us to reference the pin by that variable name potPin instead of the hard-coded number. In subsequent designs, if you need to use a different analog pin (for example pin A1), then you just need to change the number here and it will update it everywhere in your design program.

const shows that the value assigned doesn’t change during program execution

int shows the value is a number. Values declared without the keyword const can be modified later in the program execution.

pinMode(tempPin, INPUT);

After the execution of this line, the analog pin A0 will be set as input, and this will enable us to get information from it.

temp=analogRead(tempPin);

The pin we selected allows us to receive analog signals.

temp=(5\*temp\*1000.0)/(1024\*10);

This converts the 10 bit number received from sensor to a voltage between 0 and 5v.

This value is then multiplied by 1000 to convert the value to miliVolts

We then divide the resulting value by 10 because each degree rise results in a 10mV increase.

Serial.println(temp);

This reads the value stored in tempValue and prints to the serial monitor.

This value is checked every 500 ms/5s.

This value represents the temperature at the given moment.’

delay(500);

This determines after how long the temp measurements will be taken. The value passed is measured in milliseconds.

void setup (){}

This initializes the arduino and assigns functionality to its pins.

This also provides required resources for monitoring.

void loop(){}

After executing the void setup() function, we enter the void loop() and this function is executed continuously and repeatedly, until your Arduino is powered off.