



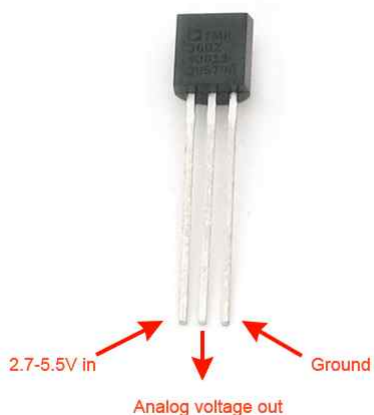
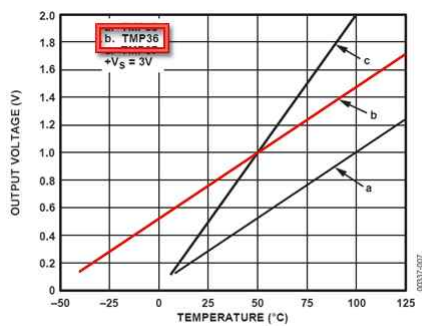
바. 온도 센서

일기예보에서는 내일의 날씨를 나타낼 때 대부분 온도 정보를 함께 보여준다. 온도는 물질의 뜨겁고 찬 정도를 나타내는 것으로서, 일상생활에서 많이 활용이 된다. 온도는 켈시우스 온도계나 디지털 온도계를 통해 손쉽게 측정이 가능하다. 아두이노에서 편리하게 사용할 수 있는 아날로그 온도 센서를 이용하여 외부의 온도를 측정하고 그 결과를 시리얼 모니터를 통해서 확인해 보자. 이를 통해 연속된 데이터인 아날로그 값을 아두이노가 처리 하는 방식과 온도를 나타내기 위하여 변환 과정을 거치는 것을 이해할 수 있다.

켈시우스 온도계	디지털 온도계
	

1) 온도 센서

온도 센서(TMP36)는 다음과 같은 구조를 가지며 전원 인가를 위한 핀과, 접지를 위한 핀 그리고 온도를 알 수 있도록 아날로그 전압을 출력해주는 3개의 핀으로 구성되어 있다.

TMP36	온도와 전압 특성
	 Figure 6. Output Voltage vs. Temperature

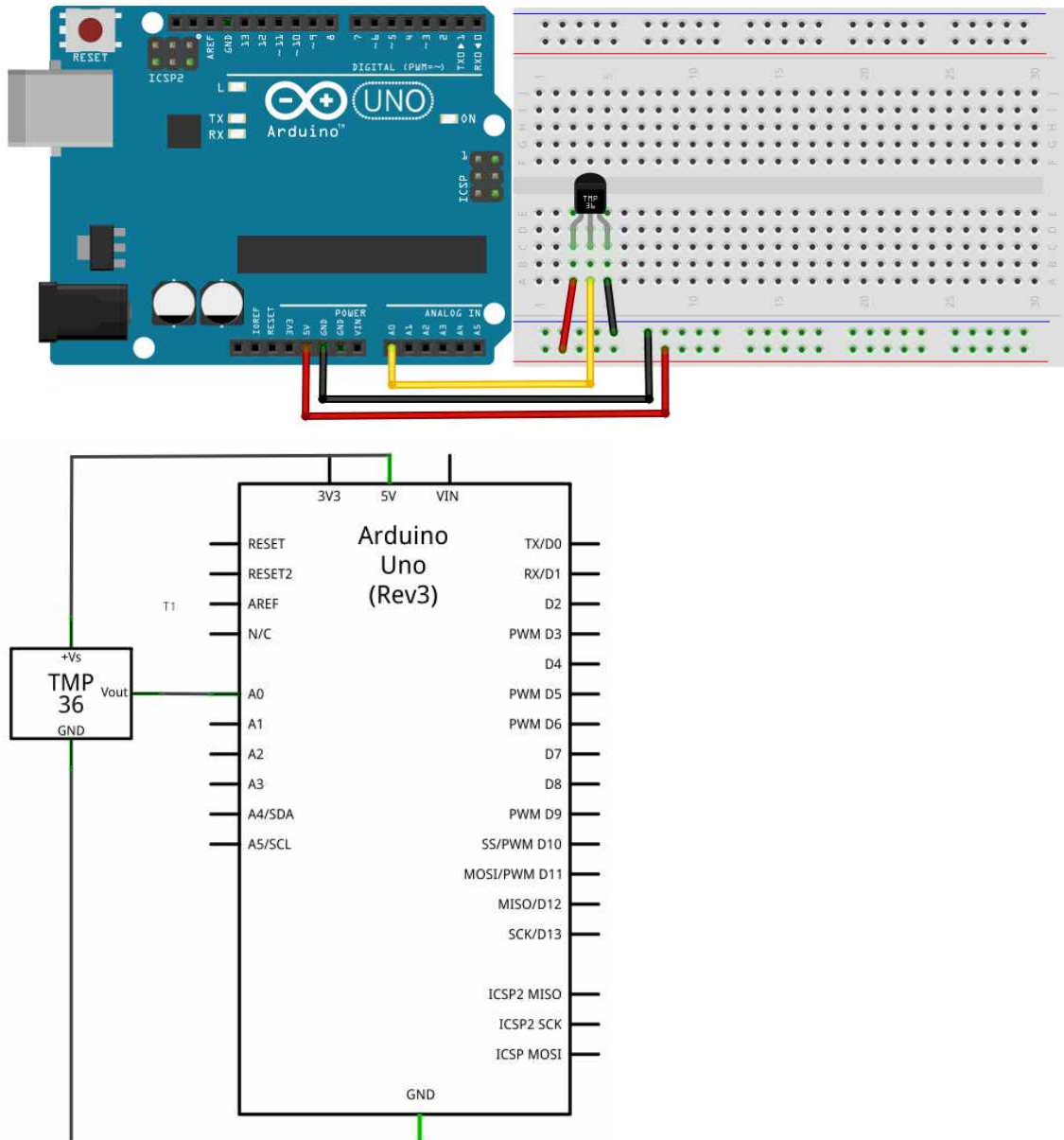
TMP36은 비교적 저렴한 아날로그 온도 센서로서 추가적인 회로를 구성하지 않고도 간편하게 온도를 측정할 수 있다. TMP36은 섭씨 -40°C ~ 150°C 까지의 온도를 측정할 수

있으며 분해능은 $10\text{mV}/^{\circ}\text{C}$ 이며 편리하게 온도 측정이 가능하다. 오차는($\pm 2^{\circ}\text{C}$) 정도가 발생한다. TMP36은 아래와 같은 특성을 지니며, 온도가 올라갈수록 전압이 증가하며, 섭씨 50도에서 1.0V 전압이 측정된다.

연속된 아날로그 신호를 아두이노 내부로 입력하는 과정에서 아날로그 데이터는 디지털로 변환이 일어나며, 아두이노의 아날로그 입력은 입력 값을 0~1023의 디지털 값으로 변환해서 보여주는 10bit 분해능을 갖추고 있다. 따라서, 입력 값을 기준 전압인 5V의 상대적인 위치값으로 변환을 한 다음 데이터시트상에서 온도를 측정하기 위해 Offset 값의 조정을 통해 원하는 온도 값을 알아낼 수 있게 된다.

2) 브레드 보드

VCC(왼쪽)는 아두이노의 5V에, GND(오른쪽)는 아두이노의 GND에 그리고 신호선(가운데))은 아두이노의 아날로그 0번 핀에 연결한다.



3) 프로그램 1

스케치를 아두이노 IDE에서 작성하고 Upload 한다.

```
01 int sensorPin = 0;
02 void setup()
03 {
04     Serial.begin(9600);
05 }
06 void loop()
07 {
08     int reading = analogRead(sensorPin);
09     float voltage = reading * 5.0;
10     voltage /= 1024.0;
11     float temperatureC = (voltage - 0.5) * 100 ;
12     Serial.println(temperatureC);
13     delay(1000);
14 }
```

[스케치 설명]

01 센서가 연결되어 있는 아날로그 포트인 0번 포트를 위한 변수 생성

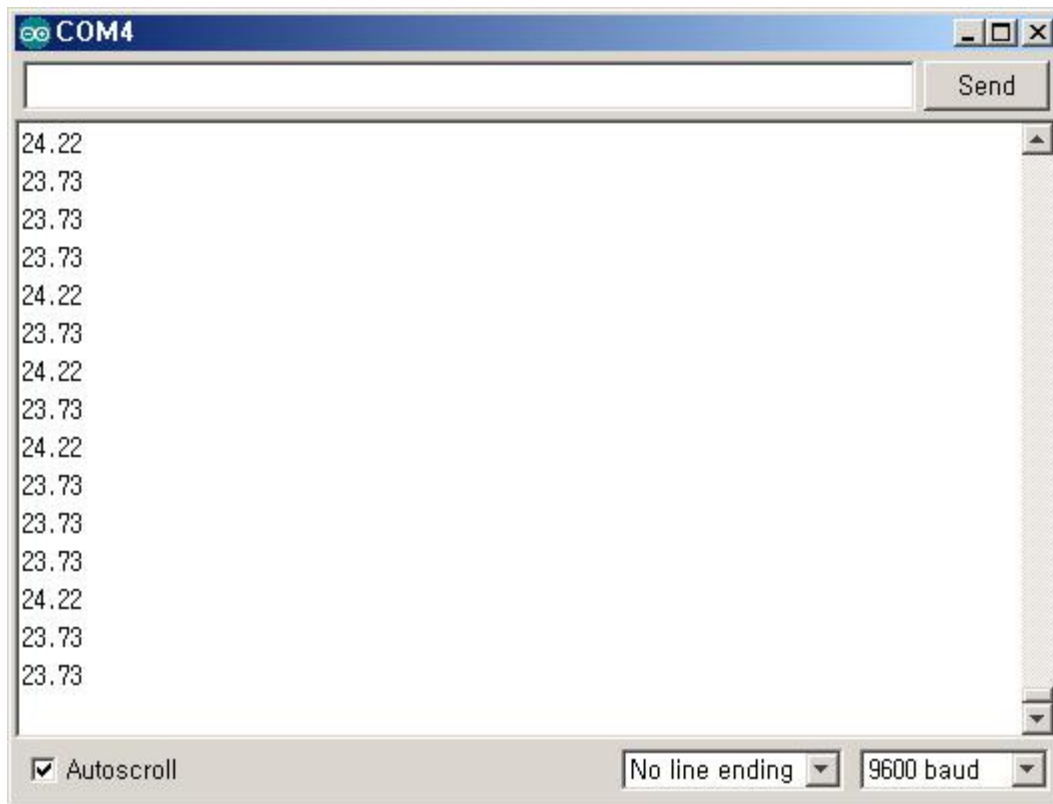
04 시리얼 통신을 위한 기본 속도를 설정해 준다.

08 A0에 들어오는 아날로그 값을 읽기 위한 함수

09-11 측정된 전압을 온도로 변환시키는 작업

12 측정된 온도를 시리얼을 통해 컴퓨터로 전송

13 1초간 지연 시킨 후에 다시 loop 함수가 시작되도록 해준다.



4) 프로그램 2

시리얼을 통한 온도 입력자료를 시각적으로 표현하기 위해 프로세싱을 이용하여 그래프로 표현한다.



```

import processing.serial.*;

Serial myPort;
int xPos = 1;
void setup () {
    size(400, 300);
    println(Serial.list());
    myPort = new Serial(this, Serial.list()[1], 9600);
    myPort.bufferUntil('\n');
    background(0);
}
void draw () {

void serialEvent (Serial myPort) {
    String inString = myPort.readStringUntil('\n');

    if (inString != null) {
        inString = trim(inString);
        float inByte = float(inString);
        fill(0);
        noStroke();
        rect(0, 0, 70, 50);
        fill(127, 34, 255);
        textFont(font12);
        textAlign(RIGHT);
        text(inString + " C" , 65, 25);
        stroke(127,34,255);
        line(xPos, height, xPos, height - inByte*7);
        if (xPos >= width) {
            xPos = 0;
            background(0);
        }
        else {
            xPos++;
        }
    }
}
}

```