

サーミスタ (NTC)

サーミスタとは、熱を感知した際の抵抗値の変化を利用した電子部品のこと。温度センサーとも呼ばれ安価なので様々な用途で利用される。回路上で抵抗と分割したアナログ値を読み取り、そのアナログ値を計算式によって温度へと変換する。今回は温度が上がると抵抗値が下がる NTC タイプを使用する。



温度（気温）の求め方

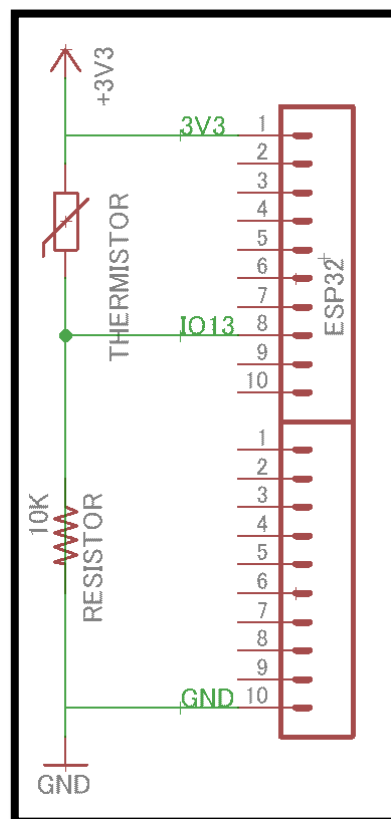
サーミスタはあくまで温度により抵抗値が変わる可変抵抗器であり、回路上にもう 1 つ抵抗を配置してその間の分圧されたアナログ値（電圧）を読み取り、その値を元に温度へと変換して取得する。

サーミスタのデータシートに記載されている温度－抵抗値の関係で温度が 25℃の時に 10kΩ の抵抗値を持つと記載されている。

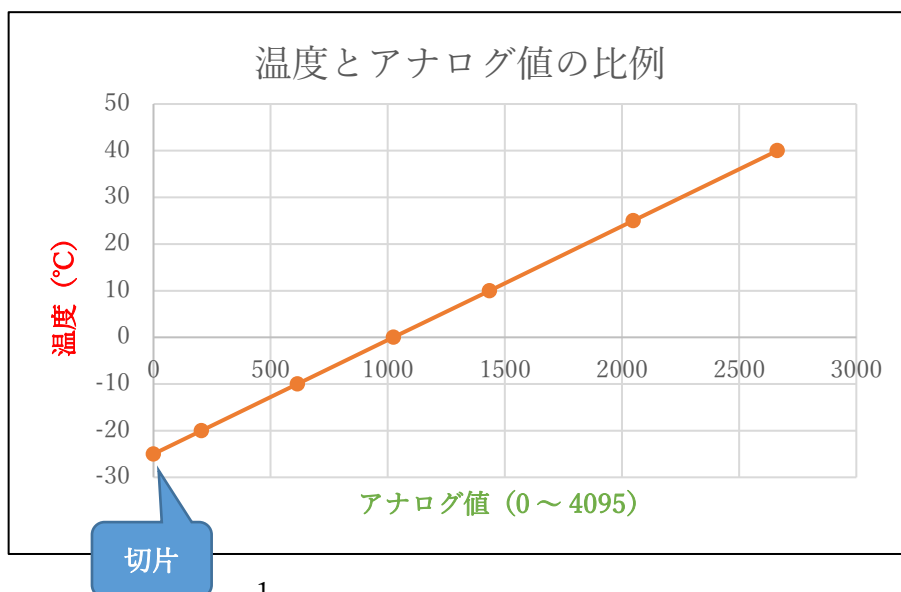
この情報から回路上のもう 1 つ抵抗の値を 10kΩ にすることで、25℃の時に抵抗値が釣り合い 1:1 で分圧することができる。

ESP32 のアナログ値は 12bit (0~4095) の範囲で取得する為、25℃のアナログ値 = 2047 (4095 を 1:1 で分圧した値) となる。

更に他の温度の数値からアナログ値を求めてグラフ化したところ、-10~40℃の範囲は直線近似で求められるほどの誤差であり気温を求めるのに差支えないので今回は直線近似で温度を求める。



単位: kΩ	
温度 (℃)	103AT
-50	329.5
-40	188.5
-30	111.3
-20	67.77
-10	42.47
0	27.28
10	17.96
20	12.09
25	10.00
30	8.313
40	5.827
50	4.160
60	3.020
70	2.228
80	1.668
85	1.451
90	1.266
100	0.9731



上記のグラフから各値を一次関数の式 ($y = ax + b$) に当てはめて温度へと変換する計算式を作成する。

y は求める温度であり、 a の傾きは 0°C と 25°C の関係から、

増加量 / (25°C のアナログ値 - 0°C のアナログ値)

の式により電卓などで求めて、**小数点第 4 位まで切り捨てた数値を a とする。**

当てはめる各数値は以下の通りである。

- ・ 増加量 → 25°C
- ・ 25°C のアナログ値 → 2047
- ・ 0°C のアナログ値 → 1024

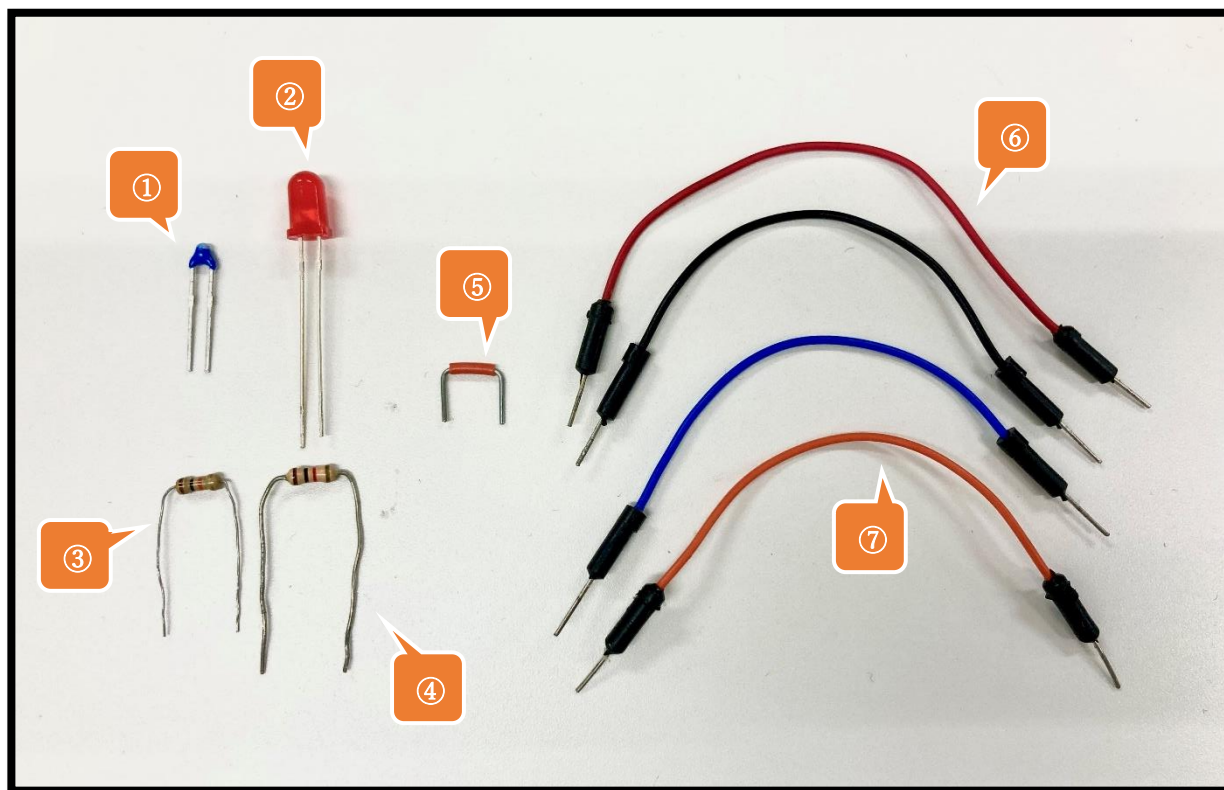
次に b は直線近似の温度の切片から**約 -25°C** とする。

残りの x に**読み取ったアナログ値**を代入すれば温度へと変換する計算式となる。

機材準備

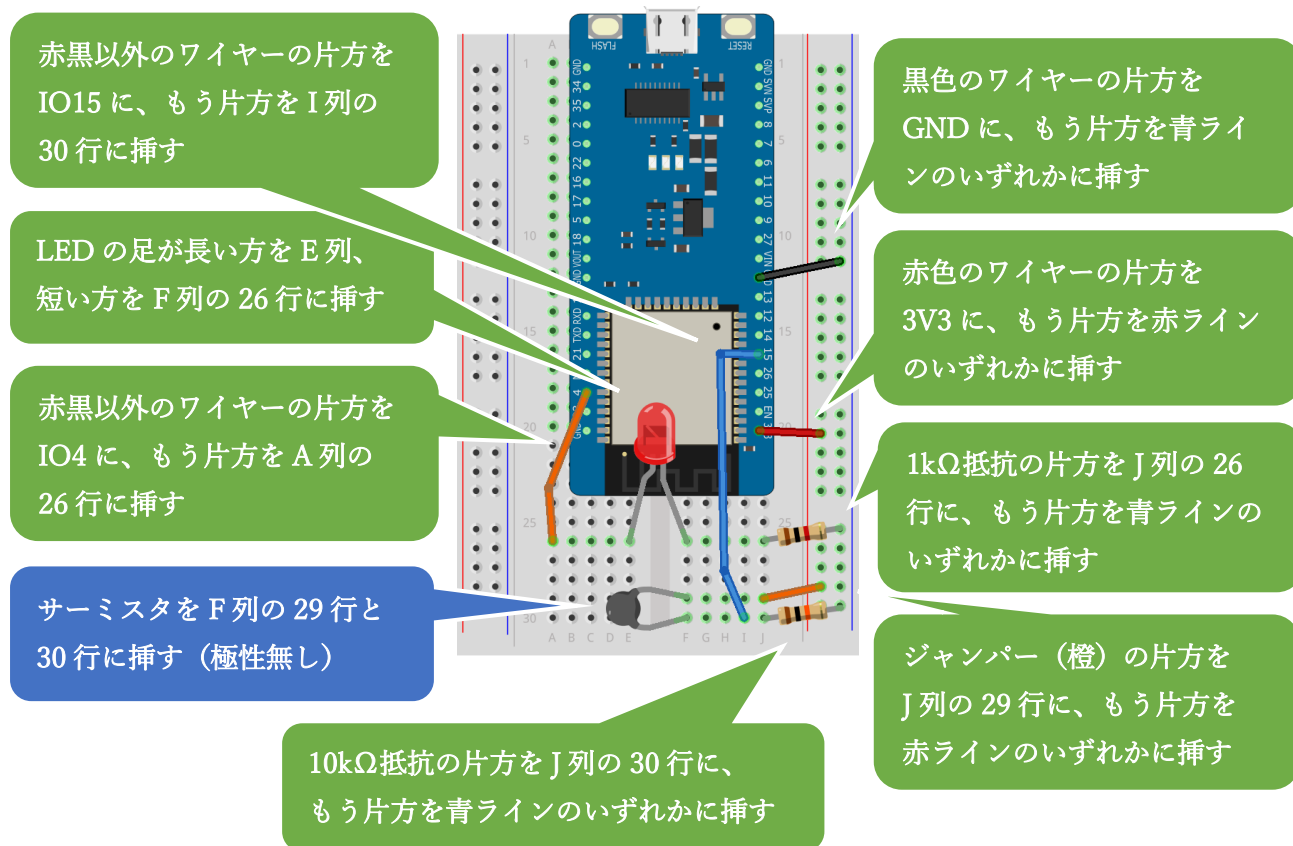
今回使用する機材を所定の位置から借りて持って来る。**※授業終了時には必ず元の場所に戻すこと。**

- ① サーミスタ
- ② LED
- ③ $10\text{k}\Omega$ 抵抗 (茶・黒・**橙**・金)
- ④ $1\text{k}\Omega$ 抵抗 (茶・黒・**赤**・金)
- ⑤ ブレッドボード用ジャンパー (**橙**)
- ⑥ ジャンパーワイヤー (赤と黒を 1 本ずつ)
- ⑦ ジャンパーワイヤー (赤・黒**以外**を 2 本)



練習 (Thermistor)

ESP32 に Micro-B USB ケーブルを差し込まずに以下の図のとおり配線を行いなさい。



配線を行った後、Micro-B USB ケーブルを ESP32 と実習機に繋ぎ通電する。

その後、以下のサンプルコードを記述し、ESP32 に書き込みを行う。

0.1 秒間隔で温度（気温）がシリアルモニタ上に表示され、**27°Cを超えた場合に LED が点灯しているか**確かめる。省略部分は前述の温度へ変換する計算式を参照し、当てはめること。

```

17:22:58.831 -> 27.19
17:22:58.955 -> 27.31
17:22:59.078 -> 27.48
17:22:59.174 -> 27.58
17:22:59.264 -> 27.66
17:22:59.358 -> 27.73
17:22:59.453 -> 27.75
17:22:59.543 -> 27.95
17:22:59.669 -> 28.05
17:22:59.764 -> 28.05
17:22:59.845 -> 28.14
17:22:59.937 -> 28.14

```

※サーミスタを指でつまむと体温により上昇し、何かしらで扇いで風を当てると下降する

☒ 自動スクロール ☒ タイムスタンプを表示 CRおよびLF 115200 bps 出力をクリア

```
/*
  Thermistor
  Date : 2022/01/01
  Author : IE1A 99 K.Murakami
*/

// ピン番号をマクロで定義
#define THER_PIN 15
#define LED_PIN 4

int analogValue = 0;    // アナログ値 (0~4095)
double temperature = 0; // 温度

// 起動時に一度だけ呼び出されるメソッド
void setup() {
  Serial.begin(115200); // シリアル通信の転送レート (bps) を設定
  // ピンの入出力設定
  pinMode(LED_PIN, OUTPUT);
}

// メインループメソッド
void loop() {
  // アナログ値取得
  analogValue = analogRead(THER_PIN);

  // アナログ値から温度を求める
  temperature = ~ 省略 ~;
  Serial.println(temperature);

  // 温度が 27℃以上の場合
  if (temperature >= 27) {
    digitalWrite(LED_PIN, HIGH); // 点灯
  } else {
    digitalWrite(LED_PIN, LOW);  // 消灯
  }

  delay(100);
}
```

課題 (ThermiStar)

前述の練習を**別名保存**し、サーミスタから 0.1 秒間隔で温度を取得した後、その温度が **27°C 以上の場合、ダイナミックスピーカーから音楽が鳴り、27°C 未満の場合は一時停止する**プログラムを追加しなさい。前回の課題と違い一時停止の為、再度温度が 27°C 以上になった場合は、**楽譜上で停止していた位置から音楽が再生される**ようにしなさい。楽譜の最後まで辿りついた場合はまた先頭から再生すること。鳴らす音はマリオのスター（無敵時間）の音楽をデフォルトとし、以下の楽譜を使用すること。

// スター楽譜

```
double star[] = { C5, NONE, C5, NONE, C5, NONE, D4, C5, NONE, C5, NONE, D4, C5, NONE, C5, NONE, B4, NONE, B4, NONE, B4, NONE, B4, NONE, C4, B4, NONE, B4, NONE, C4, B4, NONE, B4, NONE };
```

サーミスタの温度が 27°C 以上の場合、この楽譜のとおり音楽を再生する
再生中 LED は点灯し続ける

温度が 27°C 未満になり、この NONE の位置で一時停止した場合、再度 27°C 以上になった際は次の **B4** から音楽を再生する

※課題が完成した人は ESP32 にプログラムを書き込んだ状態で講師を呼び、チェックしてもらうこと。