

組み込みシステム(HI-4)

後期期末レポート課題

HI4 45 号 山口惺司

提出日: 2024/12/17

1. 課題 自動温度計測システム

ある野菜を栽培しているビニールハウス内で温度計測を自動で行いたい。次の条件を満たすような温度計測装置を Arduino UNO R4 mini を用いて実現しなさい。

- 下記の温度センサを用いて温度計測する。
- タイマー機能を用いて 1.0 秒間隔で温度計測する。
- 計測した温度はシリアル通信(115200bps)で送信する。小数点以下 2 桁。Arduino IDE のシリアルモニタで確認できれば良い。
- Arduino IDE のシリアルプロッタでグラフ化する。

[温度センサ]

秋月電子 ADT7410 使用 高精度・高分解能 I2C・16Bit 温度センサモジュール
データシート: https://akizukidenshi.com/goodsaffix/AE-ADT7410_aw.pdf

[タイマー処理]

- 関数 delay() は使用せず、タイマー機能を使用すること。
- 今回は Arduino UNO R4 を使用するので、AGTimer_R4 ライブラリを利用する。
AGTimer_R4: https://github.com/washiyamagiken/AGTimer_R4_Library

2. I2C 通信について

2.1. I2C 通信の接続方法

I2C 通信は SDA(データ線)と SCL(クロック線)の 2 本を使用する。

I2C では通信を開始するデバイスがマスターとなり、他のデバイスはスレーブとして動作する。

2.2. オープンドレインとプルアップ抵抗

SDA と SCL の出力はオープンドレインなので、必ずプルアップ抵抗が必要である。

2.3. I2C デバイスのアドレス

各 I2C デバイスは 7 ビットか 10 ビットのアドレスを持つ。

2.4. I2C の通信手順

I2C のデータ転送はスタートコンディション、アドレス、データ、ストップコンディションの順に行われる。

スタートコンディションは SDA が HIGH から LOW に変化する際に SCL が HIGH の状態で開始される。

データ転送は 8 ビットごとに行われ、各バイトの転送後には ACK ビットが付与される。受信側がデータ

の受信に成功した場合はACKビットとしてLOWを送信し、失敗した場合はNACKビットとしてHIGHを送信する。

転送が完了すると、ストップコンディションが送信される。ストップコンディションはSCLがHIGHの状態、SDAがLOWからHIGHに変化することで示される。

3. Arduino UNO で I2C 通信を行う方法について

3.1. I2C デバイスとの接続方法

Arduino UNO R4 mini で I2C 通信を行うときは Arduino 本体の SCL, SDA 端子を使う。

また 10k Ω のプルアップ抵抗を接続する。

今回作成した回路図を図 1 に示す。

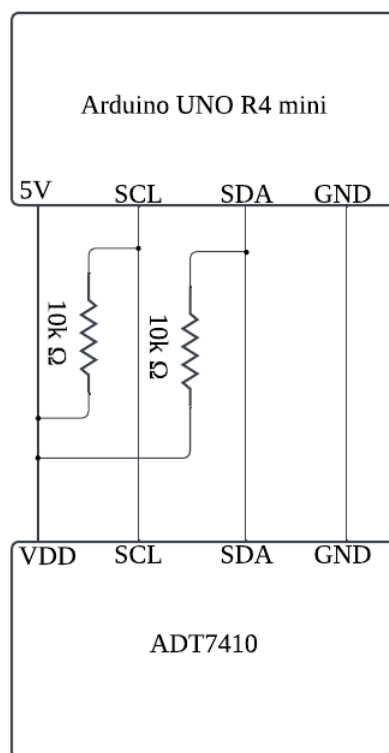


図 1 回路図

3.2. Wire クラス(I2C 通信ライブラリ)

Arduino には I2C 通信を行うための Wire ライブラリが入っている。

このライブラリを使用することで簡単にデバイスとの通信が可能になる。

3.3. I2C デバイスとの送受信方法

今回使用した Wire ライブラリの関数を以下に示す。

Wire.begin() : I2C 初期化。

Wire.requestFrom(address, quantity) : ほかの I2C デバイスにデータ要求。

Wire.read() : 受信した byte 数を調べる。

4. 温度センサモジュールについて

4.1. ANALOG DEVICES 社の温度センサ IC ADT7410 の主な技術仕様

温度範囲： $-55^{\circ}\text{C} \sim +150^{\circ}\text{C}$

電圧範囲： $2.7\text{V} \sim 5.5\text{V}$

温度精度：

- $\pm 0.5^{\circ}\text{C} @ -40^{\circ}\text{C} \sim +105^{\circ}\text{C} (2.7\text{V} \sim 3.6\text{V})$
- $\pm 0.4^{\circ}\text{C} @ -40^{\circ}\text{C} \sim +105^{\circ}\text{C} (3.0\text{V})$

16 ビット温度分解能： 0.0078°C

4.2. 秋月電子社の温度センサモジュールの主な技術仕様

温度範囲： $-55^{\circ}\text{C} \sim +150^{\circ}\text{C}$

電圧範囲： $2.7\text{V} \sim 5.5\text{V}$

温度精度：

- $\pm 0.5^{\circ}\text{C} @ -40^{\circ}\text{C} \sim +105^{\circ}\text{C} (2.7\text{V} \sim 3.6\text{V})$
- $\pm 0.4^{\circ}\text{C} @ -40^{\circ}\text{C} \sim +105^{\circ}\text{C} (3.0\text{V})$

16 ビット温度分解能： 0.0078°C

4.3. 受信データから温度を取得する方法

- ① 温度センサ IC から 16 ビットのデータを取得しする.
- ② 下位 3 ビットが不要であるため右に 3 ビットシフトし, 13 ビットにする.
- ③ 13 ビットの値を 16 で割り, 温度を取得する.

5. Arduino UNO での自動温度計測

5.1. タイマーを使って 1.0sec 毎の計測する方法

今回は AGTimerR4 ライブラリを使用する.

- ① AGTimerR4 timer で AGTimerR4 クラスのインスタンス timer を生成する.
- ② setup()関数内にて timer.init(1000000, readTemp)を実行し 1 秒ごとに readTemp()関数を実行できるようにする.
- ③ setup()関数内で timer.start()を実行しタイマーを開始する.
- ④ readTemp()関数内で計測を行うことで 1.0sec 毎の計測が可能になる.

5.2. 計測した温度を小数点以下 2 桁でシリアル通信で送信する方法

Serial.begin(115200);でシリアル通信を開始する.

Serial.println(String(fval, 2));で計測した温度を小数点以下 2 桁で出力する.

5.3. プログラムリスト

作成したプログラムを以下に示す.

```
#include <Wire.h>
#include "AGTimerR4.h"

int adt7410I2CAddress = 0x48;
AGTimerR4 timer;

void setup(void) {
    Serial.begin(115200);
    Wire.begin();
    timer.init(1000000, readTemp);
    timer.start();
}

void readTemp(void) {
    uint16_t uiVal;
    float fVal;
    int iVal;

    Wire.requestFrom(adt7410I2CAddress, 2);

    uiVal = (uint8_t)Wire.read() << 8;
    uiVal |= Wire.read();

    uiVal >>= 3;

    if (uiVal & 0x1000) {
        iVal = uiVal - 0x2000;
    }
    else {
        iVal = uiVal;
    }

    fVal = (float)iVal / 16.0;
    Serial.println(String(fVal, 2));
}

void loop(){

}
```

5.4. 実行結果とグラフ表示

30 秒間を 2 回実行し，得られた結果を図 2, 3 に示す．また，1 回目の実行はテキスト，2 回目の実行はグラフで出力した．

```
24.69
24.62
24.69
24.62
24.62
24.62
24.62
24.62
25.69
26.38
26.88
26.88
27.56
28.00
28.44
28.81
29.06
29.06
28.62
28.31
28.00
27.81
27.62
27.50
27.38
27.25
27.50
28.12
28.25
28.50
28.88
28.88
```

図 2 実行結果 1

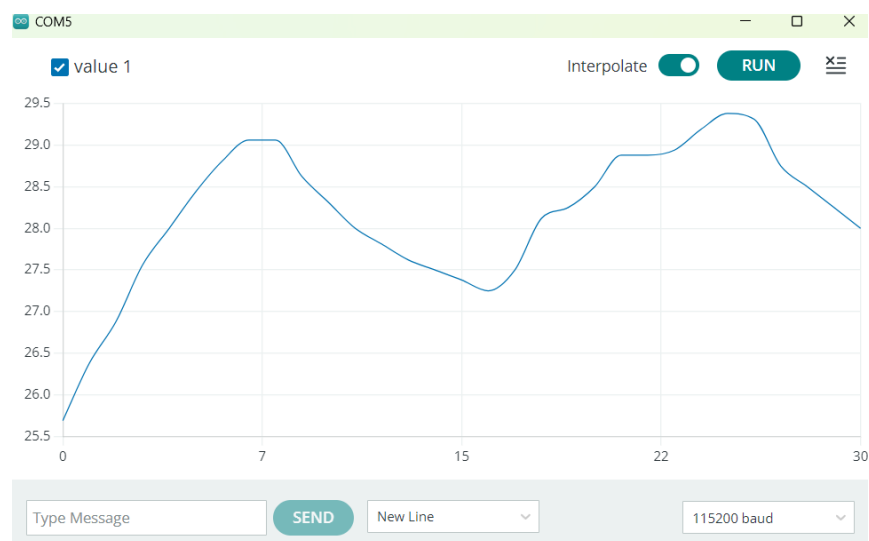


図 3 実行結果 2

6. 参考文献

XEXEQ : I2C(Inter-Integrated Circuit)とは？意味をわかりやすく簡単に解説

<https://xexeq.jp/blogs/media/it-glossary180>

CQ connect : I2Cバスのプルアップ抵抗の値 - CQ connect

<https://cc.cqpub.co.jp/system/contents/3481/>

spiceman.jp : Arduino-Wire(I2C)ライブラリの使い方 - Spiceman

https://spiceman.jp/arduino-wire-library/#index_id13

秋月電子 : ADT7410 使用 高精度・高分解能 I2C・16Bit 温度センサーモジュール

<https://akizukidenshi.com/goodsaffix/ADT7410a.pdf>

Analog Devices : ADT7410 データシートおよび製品情報

<https://www.analog.com/jp/products/adt7410.html>