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>

# I2C通信について

* 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.

* 1. 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.

* 1. 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.

* 1. 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の状態でSDAがLOWからHIGHに変化することで示される.

1. Arduino UNOでI2C通信を行う方法について
   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10kΩ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.

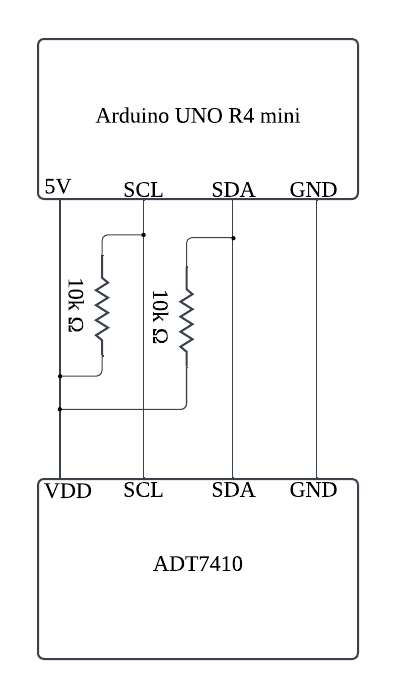


図1 回路図

* 1. 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.

* 1. 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.

1. 温度センサモジュールについて
   1. ANALOG DEVICES社の温度センサIC ADT7410の主な技術仕様

温度範囲： -55°C～+150°C

電圧範囲： 2.7V～5.5V

温度精度：

* ±0.5℃@-40℃～+105℃(2.7V～3.6V)
* ±0.4℃@-40℃～+105℃(3.0V)

16ビット温度分解能：0.0078℃

* 1. 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°C～+150°C

電圧範囲： 2.7V～5.5V

温度精度：

* ±0.5℃@-40℃～+105℃(2.7V～3.6V)
* ±0.4℃@-40℃～+105℃(3.0V)

16ビット温度分解能：0.0078℃

* 1. 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.
4. Arduino UNOでの自動温度計測
   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.
   1. 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.

* 1. 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(){

}

* 1. 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.

屋外, 吊るす, 異なる, 民衆 が含まれている画像

自動的に生成された説明

図2 実行結果1

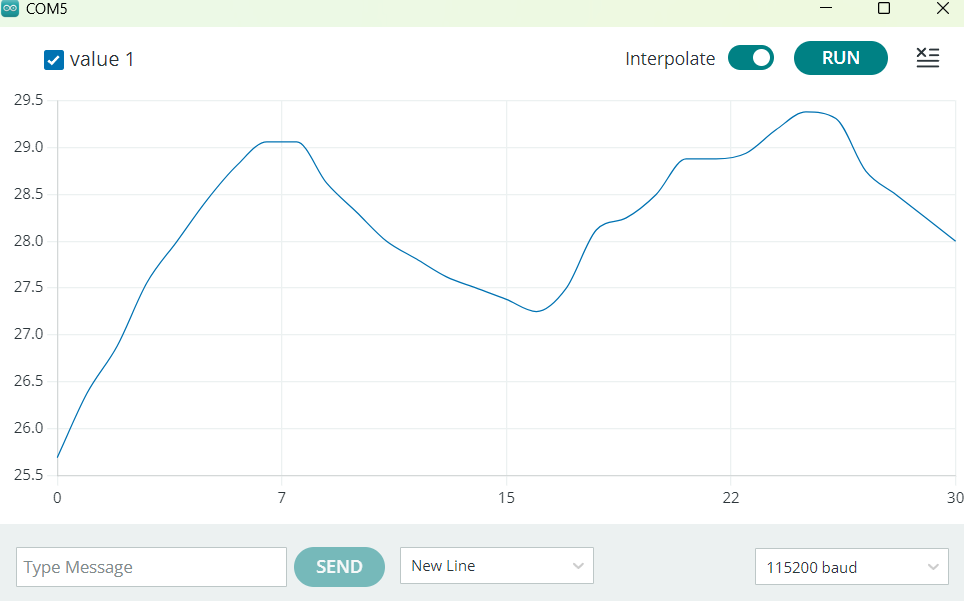


図3 実行結果2

# 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：I 2 C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>