

デュアルウェア講習会課題 4

会津大学 講習会

LM61CIZ を使い温度を取得する (FaBo 使用)

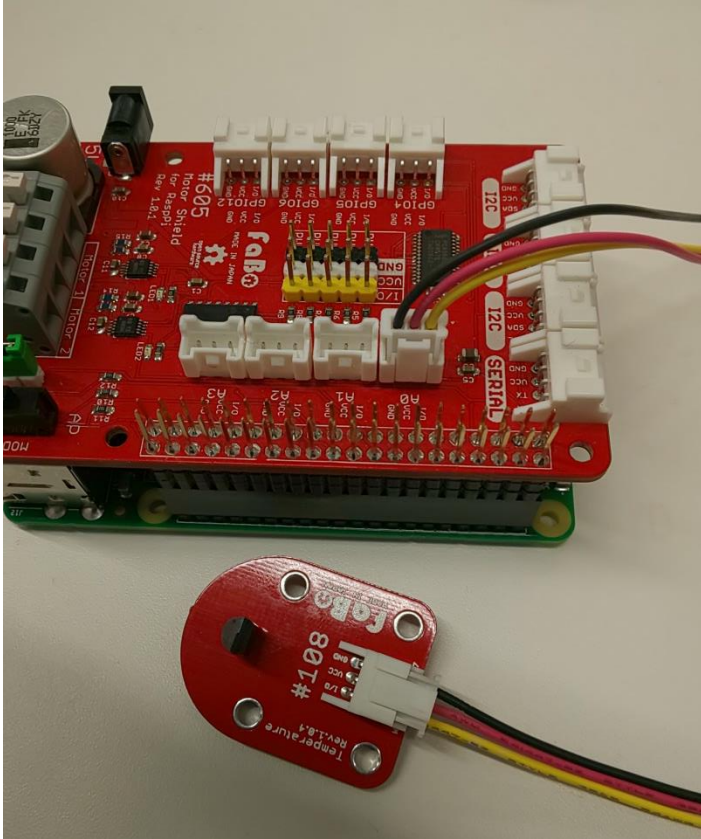
目次

1	課題.....	1
1.1	課題説明	1
2	機材説明.....	1
2.1	LM61CIZ.....	2
2.2	MCP3008	3
2.3	ジャンパー線	5
3	回路作成.....	6
3.1	Raspberry Pi に FaBo #605 を接続.....	7
3.2	FaBo #108 を接続.....	8
4	プログラム	9
4.1	温度取得のプログラム	9
4.2	プログラムの解説	10
4.2.1	AD コンバータからの値の取得.....	10
4.2.2	配列 adc の値を 1 つの変数にまとめる.....	11
4.2.3	AD コンバータから受け取った値を電圧の値に変換.....	12
4.2.4	電圧を温度に変換	12
4.3	プログラムの実行	13
4.4	追加課題	14
4.4.1	Python で CSV ファイルを使用する.....	14
4.4.2	Raspberry Pi から Windows にファイルを送信	14
4.4.3	CSV ファイルのデータからグラフを作成.....	14

1 課題

1.1 課題説明

Raspberry Pi にモーターシールドと温度センサーを接続し温度を計測します。



2 機材説明

この課題は以下の機材を使用します。手元にあることを確認してください。

機材	個数
Raspberry Pi	1
LM61C1Z (FaBo #108)	1
MCP3008 (FaBo #605)	1
ジャンパー線 (FaBo 用)	1

2.1 LM61CIZ

データシート及び参考資料															
http://akizukidenshi.com/download/ds/ti/LM61CIZ.pdf http://fabo.io/108.html															
特徴 (P. 1 : 概要, 主な仕様を参照)															
<ul style="list-style-type: none"> 温度センサー 計測範囲 : $-30^{\circ}\text{C} \sim +100^{\circ}\text{C}$ (最大$\pm 4.0^{\circ}\text{C}$) 計測した値を電圧で出力 アナログ通信の為, Raspberry Pi と通信には AD コンバータが必要 															
電圧・電流 (P. 3 : 絶対最大定格, 主な仕様を参照)															
<ul style="list-style-type: none"> 動作電圧 (+Vs) : $+2.7\text{V} \sim +10\text{V}$ ※ Raspberry Pi は 3.3V の電圧を供給可能 出力電圧 : $(+Vs+0.6\text{V}) \sim -0.6\text{V}$ 出力電流 : 10mA 															
端子 (http://fabo.io/img/108_temperature_sch.png)															
<table border="1"> <thead> <tr> <th>端子名</th> <th>説明</th> </tr> </thead> <tbody> <tr> <td>I/O</td> <td>電源からの電圧を入力</td> </tr> <tr> <td>VCC</td> <td>センサーが測定した値 (電圧) を出力</td> </tr> <tr> <td>GND</td> <td>不要な電圧を放出</td> </tr> </tbody> </table>	端子名	説明	I/O	電源からの電圧を入力	VCC	センサーが測定した値 (電圧) を出力	GND	不要な電圧を放出							
端子名	説明														
I/O	電源からの電圧を入力														
VCC	センサーが測定した値 (電圧) を出力														
GND	不要な電圧を放出														
出力値 (P. 1 : 代表的なアプリケーション参照)															
出力される電圧は以下の式で求められます。															
$V_o = (+10\text{mV}/^{\circ}\text{C} \times T^{\circ}\text{C}) + 600\text{mV}$															
<ul style="list-style-type: none"> $+10\text{mV}/^{\circ}\text{C}$: (計測値が 1°C 上昇すると出力電圧が $+10\text{mV}$ 増加) $T^{\circ}\text{C}$: 測定温度 															
<ul style="list-style-type: none"> 右表は測定温度と出力電圧の対応表 															
	<table border="1"> <thead> <tr> <th>Temperature (T)</th> <th>Typical V_o</th> </tr> </thead> <tbody> <tr> <td>$+100^{\circ}\text{C}$</td> <td>$+1600\text{ mV}$</td> </tr> <tr> <td>$+85^{\circ}\text{C}$</td> <td>$+1450\text{ mV}$</td> </tr> <tr> <td>$+25^{\circ}\text{C}$</td> <td>$+850\text{ mV}$</td> </tr> <tr> <td>0°C</td> <td>$+600\text{ mV}$</td> </tr> <tr> <td>-25°C</td> <td>$+350\text{ mV}$</td> </tr> <tr> <td>-30°C</td> <td>$+300\text{ mV}$</td> </tr> </tbody> </table>	Temperature (T)	Typical V_o	$+100^{\circ}\text{C}$	$+1600\text{ mV}$	$+85^{\circ}\text{C}$	$+1450\text{ mV}$	$+25^{\circ}\text{C}$	$+850\text{ mV}$	0°C	$+600\text{ mV}$	-25°C	$+350\text{ mV}$	-30°C	$+300\text{ mV}$
Temperature (T)	Typical V_o														
$+100^{\circ}\text{C}$	$+1600\text{ mV}$														
$+85^{\circ}\text{C}$	$+1450\text{ mV}$														
$+25^{\circ}\text{C}$	$+850\text{ mV}$														
0°C	$+600\text{ mV}$														
-25°C	$+350\text{ mV}$														
-30°C	$+300\text{ mV}$														

2.2 MCP3008

データシート及び参考資料																					
http://akizukidenshi.com/download/ds/microchip/mcp3008.pdf http://fabo.io/605.html																					
特徴 (P.1 : Features, Description を参照)																					
<ul style="list-style-type: none"> ・ FaBo #605 に組み込まれている AD コンバータ ・ 分解能 : 10bit (1024 段階) ・ チャンネル数 : 8 ・ 通信方式 : SPI 																					
電圧・電流 (P.1 : Features を参照)																					
<ul style="list-style-type: none"> ・ 動作電圧 : +2.7V ~ +5.5V ・ かける電圧の大きさによってサンプリングレートが変化 (0.75MHz ~ 2MHz) 																					
端子 (P.1 : Package Types を参照)																					
<table border="1"> <thead> <tr> <th>端子名</th> <th>説明</th> </tr> </thead> <tbody> <tr> <td>CH0~CH7</td> <td>アナログ信号を入力</td> </tr> <tr> <td>VDD</td> <td>電源端子</td> </tr> <tr> <td>VREF</td> <td>計測基準になる電圧を入力</td> </tr> <tr> <td>CLK</td> <td>SPI 関連端子 SCKL 端子に接続する</td> </tr> <tr> <td>DOUT</td> <td>SPI 関連端子 MISO 端子に接続する</td> </tr> <tr> <td>DIN</td> <td>SPI 関連端子 MOSI 端子に接続する</td> </tr> <tr> <td>CS/SHDN</td> <td>SPI 関連端子 CE0 または CE1 端子に接続する</td> </tr> <tr> <td>AGND</td> <td>アナログ入力側の GND</td> </tr> <tr> <td>DGND</td> <td>デジタル出力側の GND</td> </tr> </tbody> </table>		端子名	説明	CH0~CH7	アナログ信号を入力	VDD	電源端子	VREF	計測基準になる電圧を入力	CLK	SPI 関連端子 SCKL 端子に接続する	DOUT	SPI 関連端子 MISO 端子に接続する	DIN	SPI 関連端子 MOSI 端子に接続する	CS/SHDN	SPI 関連端子 CE0 または CE1 端子に接続する	AGND	アナログ入力側の GND	DGND	デジタル出力側の GND
端子名	説明																				
CH0~CH7	アナログ信号を入力																				
VDD	電源端子																				
VREF	計測基準になる電圧を入力																				
CLK	SPI 関連端子 SCKL 端子に接続する																				
DOUT	SPI 関連端子 MISO 端子に接続する																				
DIN	SPI 関連端子 MOSI 端子に接続する																				
CS/SHDN	SPI 関連端子 CE0 または CE1 端子に接続する																				
AGND	アナログ入力側の GND																				
DGND	デジタル出力側の GND																				

コンフィグレーションビット (P. 19 : TABLE 5-2: CONFIGURE BITS FOR THE MCP3008 を参照)

コンフィグレーションビットはどの CH に接続しているかを示す値です。

SPI 通信を行う際に必要になります。

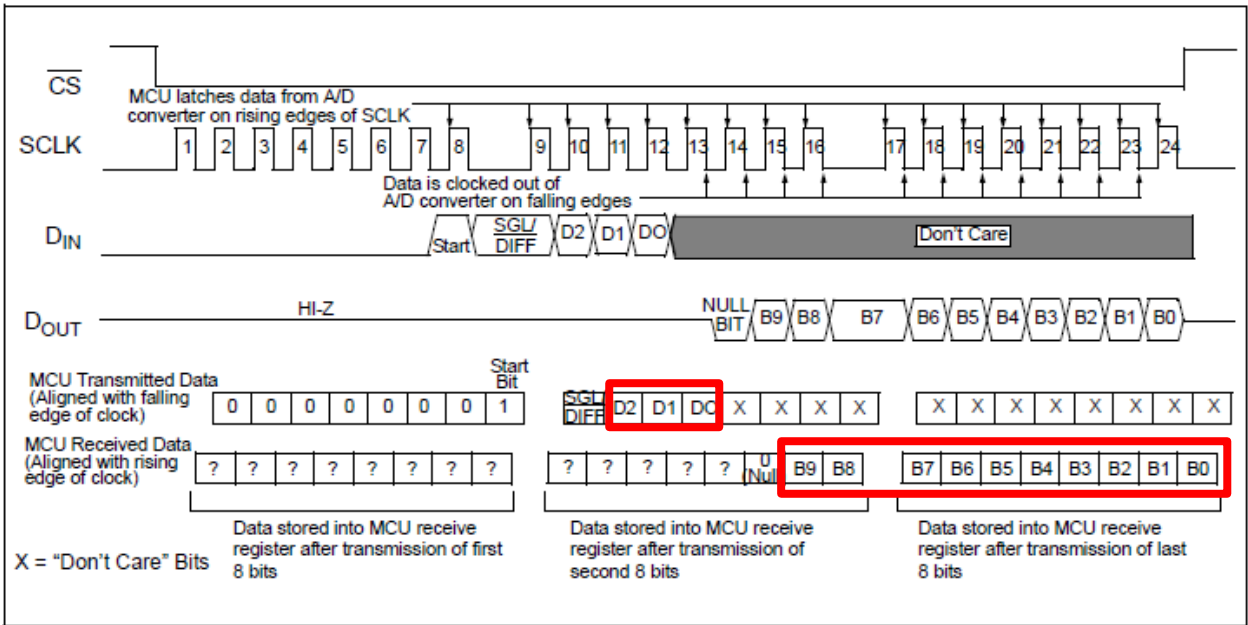
例 : CH0 に接続するとコンフィグレーションビットは[1000]になります。

TABLE 5-2: CONFIGURE BITS FOR THE MCP3008

Control Bit Selections				Input Configuration	Channel Selection
Single /Diff	D2	D1	D0		
1	0	0	0	single-ended	CH0
1	0	0	1	single-ended	CH1
1	0	1	0	single-ended	CH2
1	0	1	1	single-ended	CH3
1	1	0	0	single-ended	CH4
1	1	0	1	single-ended	CH5
1	1	1	0	single-ended	CH6
1	1	1	1	single-ended	CH7

値の取得方法 (P. 21 : FIGURE 6-1: SPI Communication using 8-bit segments (Mode 0, 0: SCLK idles low) を参照)

コンフィグレーションビットを送信して、指定したチャンネルに接続されたデバイスの値を取得します。



赤線で囲まれた所が使用する bit です。D2, D1, D0 で使用する CH を送信します。B9~B0 の 10bit の値をデバイスの出力として受信します。

出力値 (P. 17 : DIGITAL OUTPUT CODE CALCULATION を参照)

MCP3008 は入力されたアナログ値を以下の式で変換します。

$$\text{Digital Output Code} = \frac{1024 \times V_{IN}}{V_{REF}}$$

Digital Output Code : 出力する値

V_{IN} : CH から入力された電圧 (アナログ値)

V_{REF} : 端子 VREF に接続された電圧 (基準電圧)

1024 : 分解能

計算例 : 基準電圧 5V, 入力される電圧が 2V の時

1. 基準の電圧を分解能で割る : $5/1024 = 0.0048828125$
2. 入力された値を [1.] の値で割る : $2/0.0048828125 = 409.6$
3. AD コンバータの結果として 409 が出力される

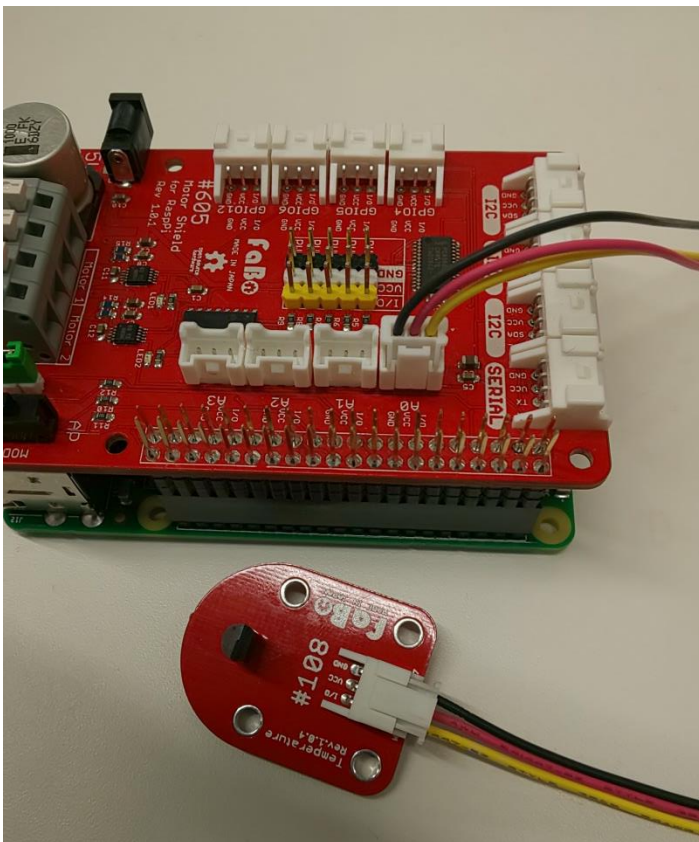
2.3 ジャンパー線

課題 1 を参照

Fabo 用の 3pin ケーブルは 3 本のジャンパー線が 1 つにまとまっています。

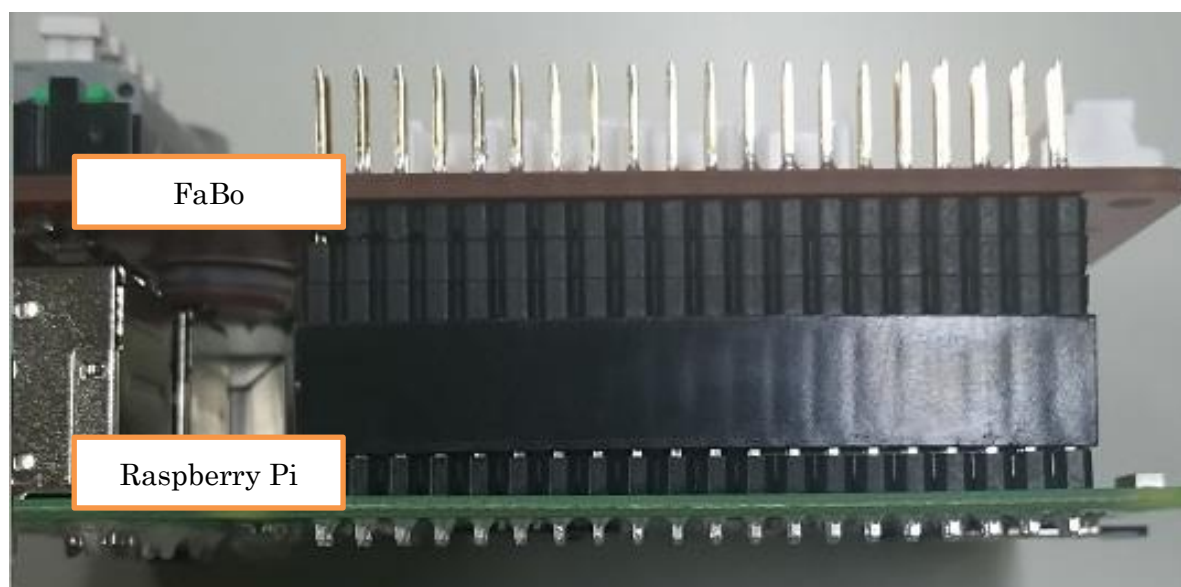
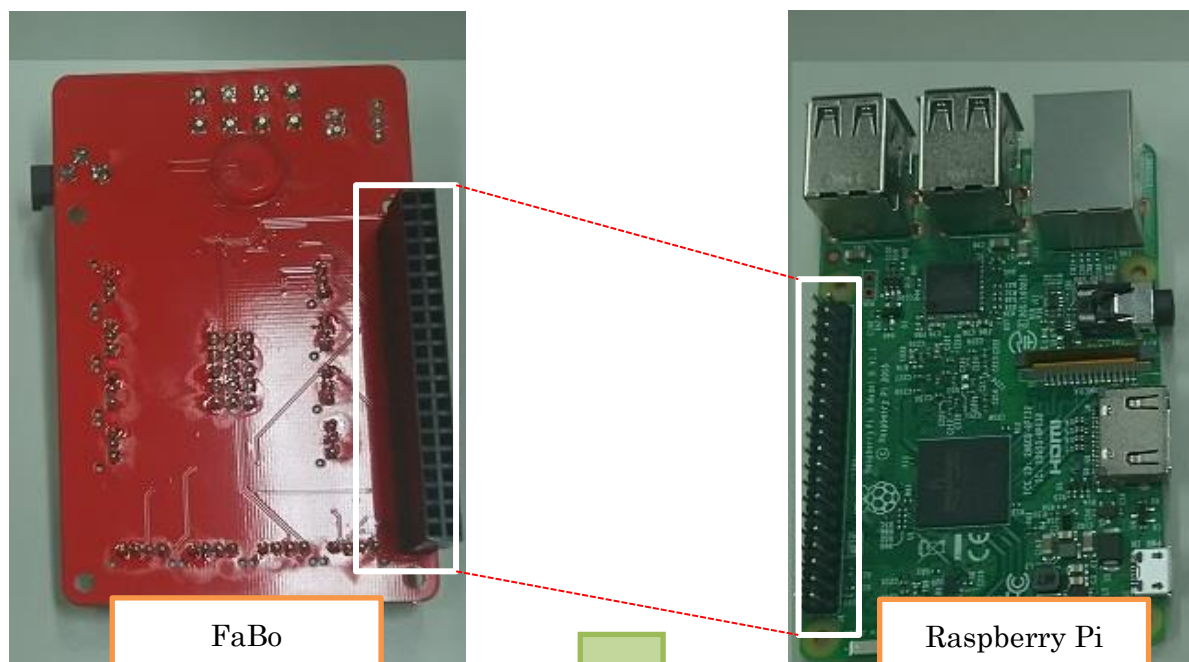
3 回路作成

Raspberry Pi に FaBo を接続し回路を作成します。
完成図は以下になります。



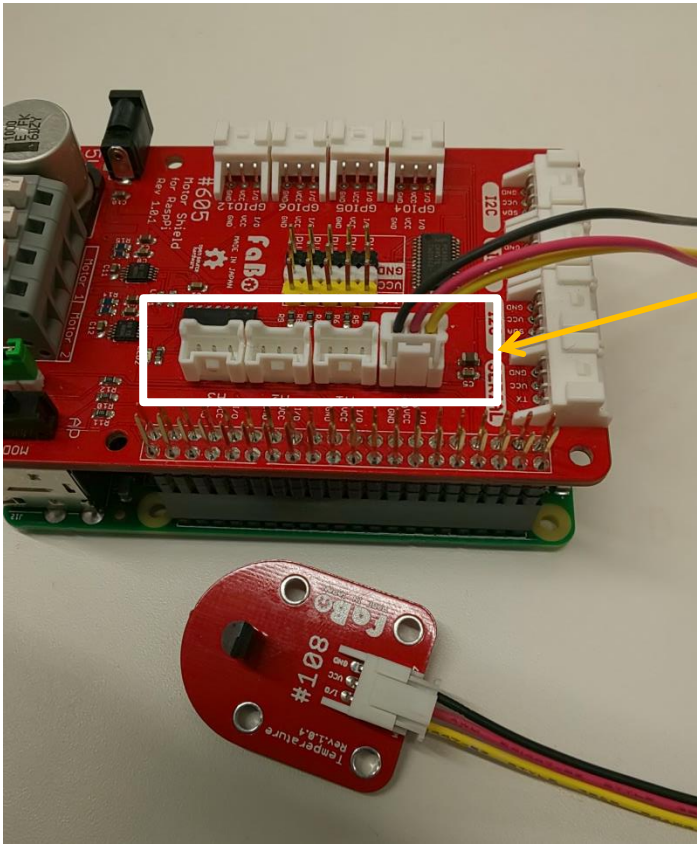
3.1 Raspberry Pi に FaBo #605 を接続

Raspberry Pi の GPIO ピン（オス）と FaBo #605 の GPIO ピン（メス）を接続します。



3.2 FaBo #108 を接続

FaBo#108 を FaBo#605 のアナログピン (A0) に接続します。

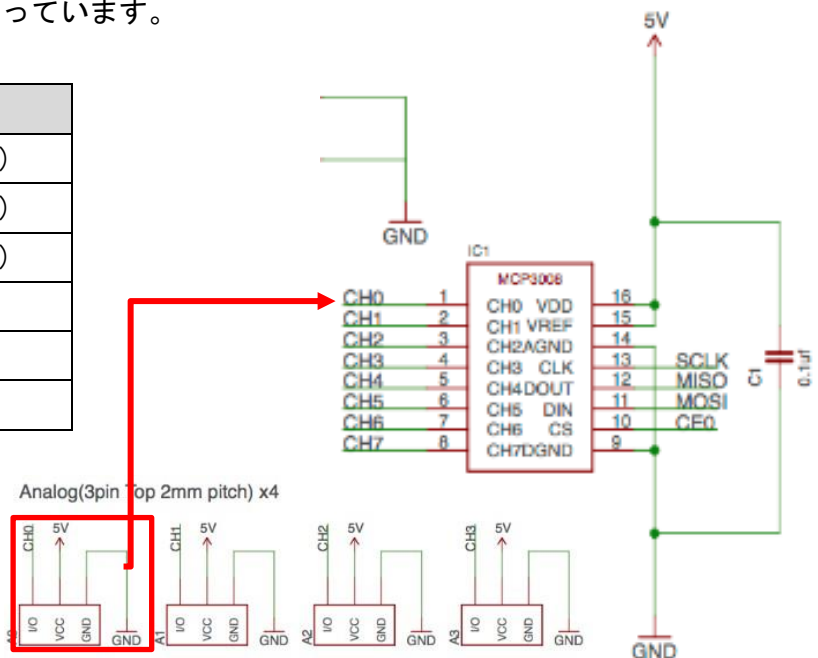


アナログピンに接続

この時内部の接続は以下の様になっています。

MCP3008	接続先
CLK	SCLK (Raspberry Pi)
DOUT	MISO (Raspberry Pi)
DIN	MOSI (Raspberry Pi)
CS	CE0 (Raspberry Pi)
VDD	5V
VREF	5V

温度センサは CH0 に接続している状態になります。



4 プログラム

温度を取得するプログラムを作成します。

4.1 温度取得のプログラム

以下のプログラムで温度を取得します。

一部「TODO:」とコメントアウトしていますので、その部分の処理を作成してください。

```
#!/usr/bin/env python

import time
import sys
import spidev

SPI_SPEED = 1000000
spi = spidev.SpiDev()
spi.open(0,0)
spi.max_speed_hz =SPI_SPEED

def readAdc(channel):
    adc = #TODO: AD コンバータによって変換された値を取得します #
    data = #TODO: 配列 adc の値を 1 つの変数にまとめます #
    return data

def convertVolts(data):
    volts = #TODO: AD コンバータから受け取った値を電圧の値に変換します #
    volts = round(volts,4)
    return volts

def convertTemp(volts):
    temp = #TODO: 電圧を温度に変換します #
    temp = round(temp,4)
    return temp

if __name__ == '__main__':
    try:
        while True:
            data = readAdc(0)
            print("adc : {:8} ".format(data))
            volts = convertVolts(data)
            temp = convertTemp(volts)
            print("volts: {:.2f}".format(volts))
            print("temp : {:.2f}".format(temp))

            time.sleep(5)
    except KeyboardInterrupt:
        spi.close()
        sys.exit(0)
```

テキストにコピーして「04_MCP3008.py」のファイル名で保存してください。
 コピーが出来ない方は以下 URL からダウンロードしてください。

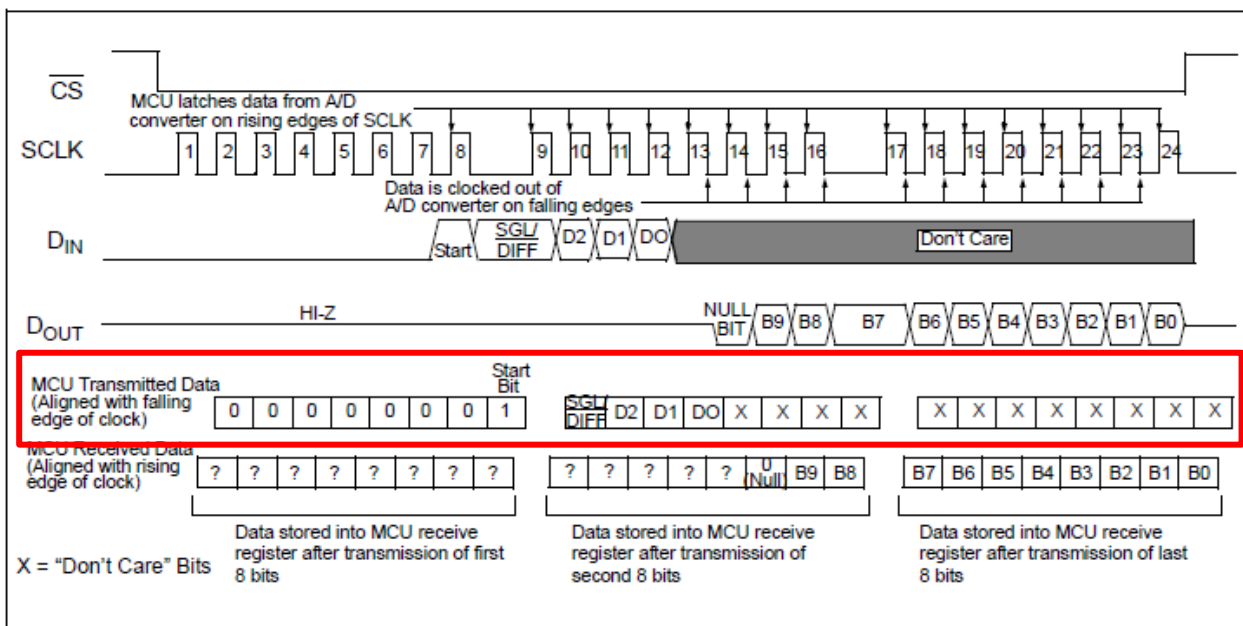
https://rtc-fukushima.jp/wp/wp-content/uploads/2019/01/04_MCP3008.zip

4.2 プログラムの解説

4.2.1 AD コンバータからの値の取得

Spidev のメソッド「xfer2()」を使用することで AD コンバータからの値を取得できます。
 送信する値はコンフィグレーションビット（使用しているチャンネルの番号）になります。

MCP3008 のコンフィグレーションビットは下図赤枠内の [Start Bit] ~ [D0] に格納されますので
 参考にして下さい。



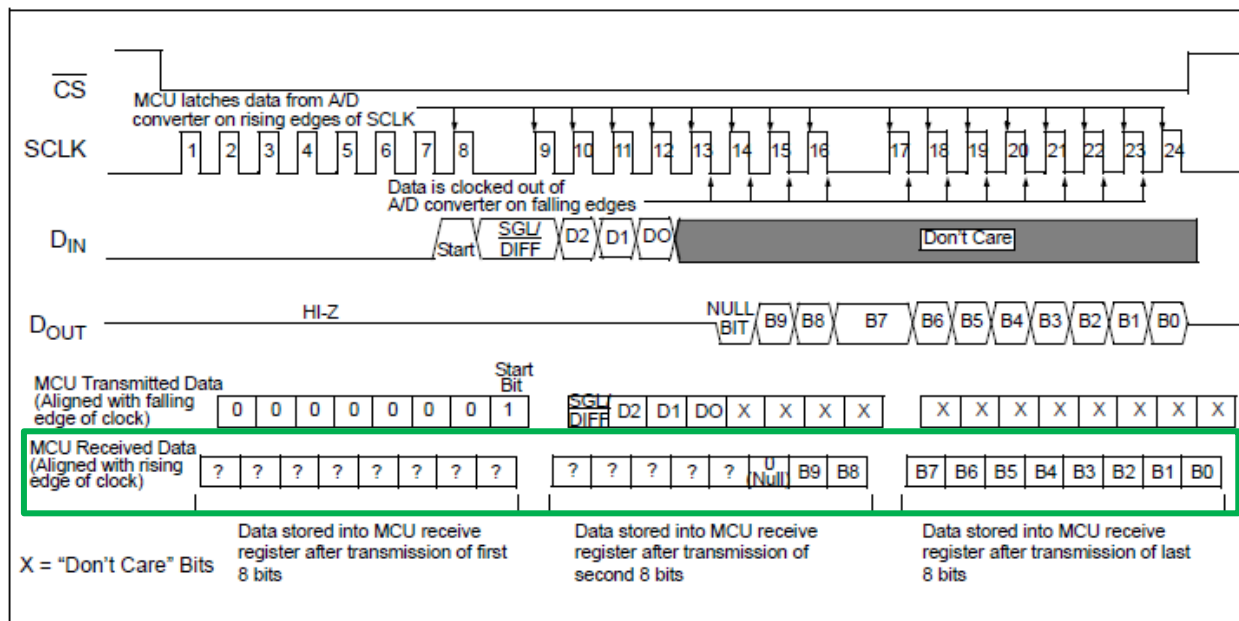
<http://akizukidenshi.com/download/ds/microchip/MCP3008.pdf>

- ・コンフィグレーションビットの格納場所は AD コンバータ毎に変わりますので、使用する際はデータシートを確認してください。
- ・コンフィグレーションビットや xfer2 メソッドの使用方法などは課題 3 を参考にして下さい。

4.2.2 配列 adc の値を1つの変数にまとめる

xfer2 メソッドを使用することで AD コンバータからの値を取得できます。

AD コンバータからの値は下図のように 3 つの配列に格納されて受信されます。



この中で必要なデータは B9~B0 の 10bit のみでほかの要素は使用しませんので、必要な要素のみ抜き出してください。

- ・要素のデータ長は AD コンバータ毎に変わりますので、使用する際はデータシートを確認してください。

4.2.3 AD コンバータから受け取った値を電圧の値に変換

AD コンバータから受け取った値は実際の電圧と値が異なるため変換する必要があります。
アナログ信号からデジタル信号への変換は下記の計算で行われています。

- ・例：基準 3.3V, 分解能 1024, 入力された値が 1V の場合
 1. 基準の電圧を分解能で割る： $3.3 / 1024 = 0.00322265625$
 2. 入力された値を[1.]の値で割る： $1 / 0.00322265625 = 310.303030303$
 3. AD コンバータの結果として 310 が出力される

この手順を参考にして電圧を求めてください。

4.2.4 電圧を温度に変換

LM61BIZ は取得した温度を電圧で出力します。測定温度と電圧は下記の表のように比例します。

Temperature (T)	Typical V_o
+100°C	+1600 mV
+85°C	+1450 mV
+25°C	+850 mV
0°C	+600 mV
-25°C	+350 mV
-30°C	+300 mV

<http://akizukidenshi.com/download/LM61biz.pdf>

出力する電圧は下記の式で求めることができます。

$$V_o = (+10\text{mV}/^\circ\text{C} \times T^\circ\text{C}) + 600\text{mV}$$

これらを参考に温度を求めてください。

4.3 プログラムの実行

プログラムを実行すると、AD コンバータからの値、電圧、温度の3つの値が表示されます。

```
ファイル(E) 編集(E) 設定(S) コントロール(O)
pi@raspA1: ~ $ python 04_MCP3008.py
adc :      175 ← AD コンバータの値
volts:     0.85 ← 電圧
temp :    25.45 ← 温度
adc :      175
volts:     0.85
temp :    25.45
```

ここまで終了した方は次ページの追加課題にも挑戦してみてください。

4.4 追加課題

取得した温度を CSV ファイルに書き込み、Excel を使用してグラフを作成します。

4.4.1 Python で CSV ファイルを使用する

Python で CSV ファイルへの値の書き込みは以下の手順で行うことができます。

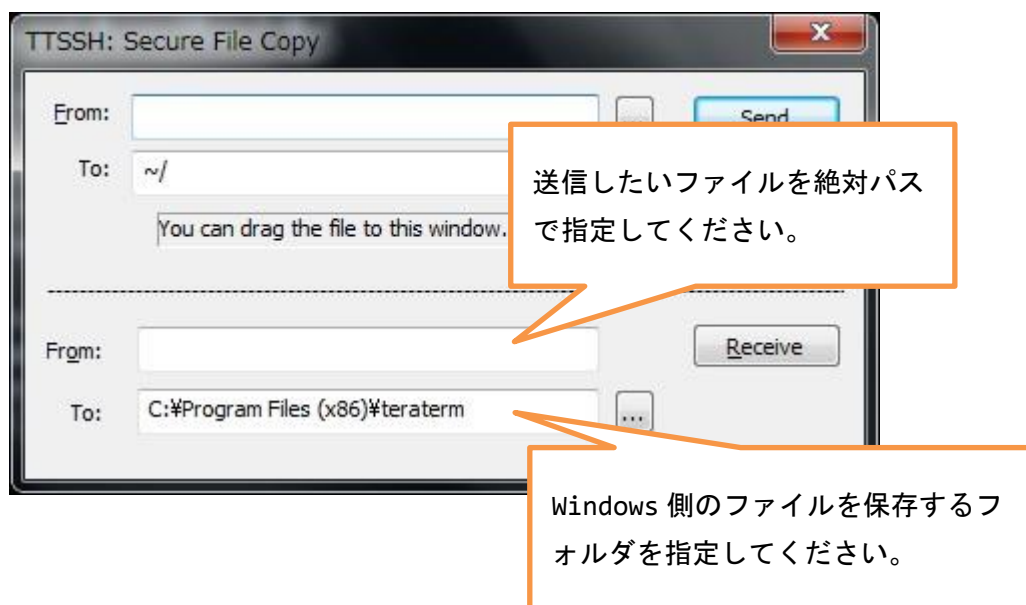
```
f = open('plot.csv', 'w')           #ファイルを開く
f.write( str("任意の変数") )       #値を書き込む
f.write("¥n")                       #改行処理
f.close()                           #ファイルを閉じる
```

※open メソッドの第 2 引数を変えると書き込み方法が変わります。

- ・ w : CSV ファイルに値を上書きします。
- ・ a : CSV ファイル内にデータが記載されているとき、その続きから値を追記します。

4.4.2 Raspberry Pi から Windows にファイルを送信

Raspberry Pi から Windows へのファイル送信は Tera Term の「SSH SCP …」を使用します。



Receive ボタンをクリックするとファイルが送信されます。

4.4.3 CSV ファイルのデータからグラフを作成

CSV ファイルをエクセルで開き、データからグラフを作成します。

作成するグラフは「折れ線グラフ」を使用してください。