

デュアルウェア講習会課題 3

会津大学 講習会

LM61BIZ を使い温度を取得する

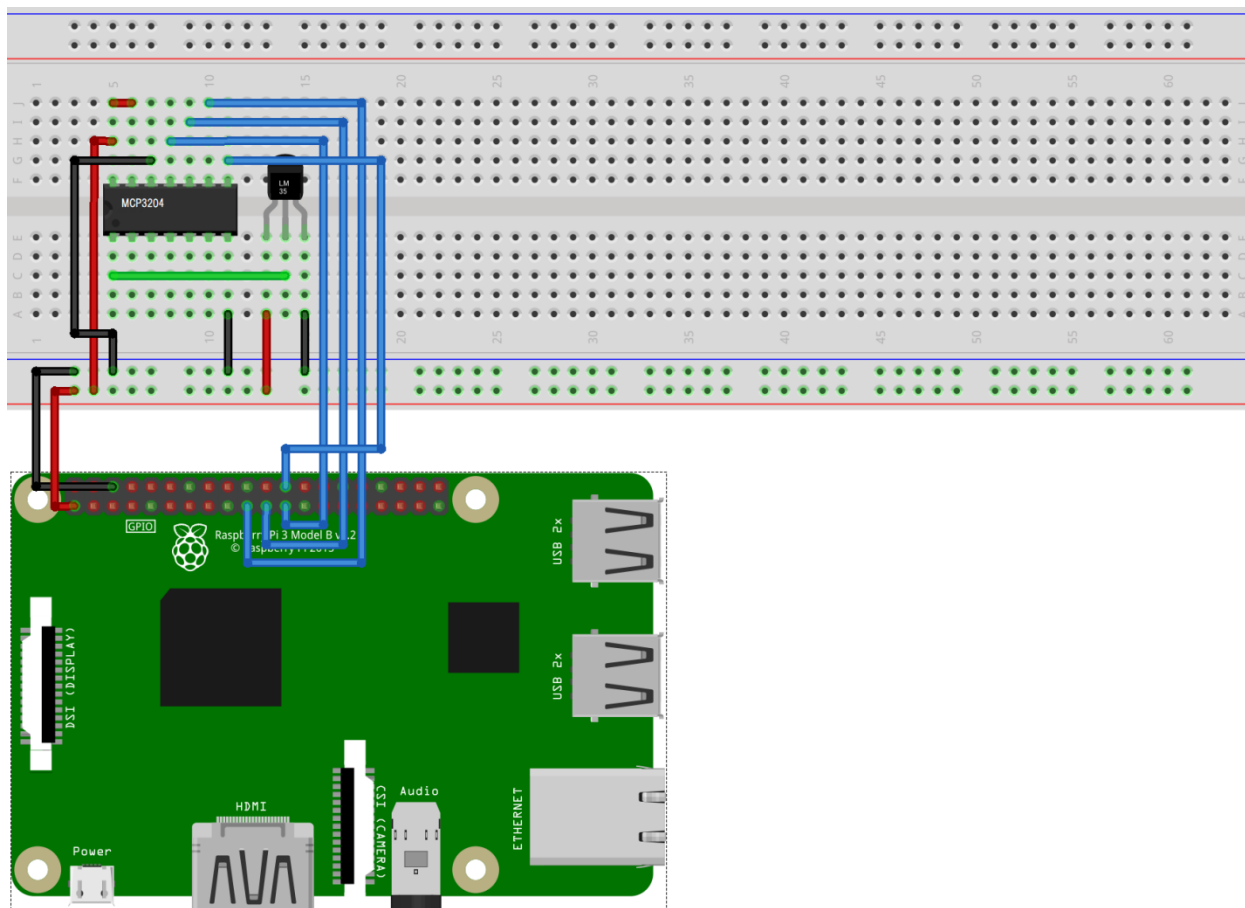
目次

1	課題.....	1
1.1	課題説明	1
2	機材説明.....	1
2.1	ブレッドボード.....	2
2.2	LM61BIZ.....	2
2.3	MCP3204	3
2.4	ジャンパー線	5
3	SPI	6
3.1	SPI（シリアル・ペリフェラル・インタフェース）について	6
3.2	SPI の使い方	6
3.2.1	Python での使い方	6
4	回路作成.....	8
4.1	LM61BIZ を差し込む	9
4.2	MCP3204 を差し込む.....	9
4.3	MCP3204 と LM61BIZ を接続.....	10
4.4	Raspberry Pi と MCP3204 と LM61BIZ を接続	11
5	プログラム	12
5.1	温度取得のプログラム	12
5.2	プログラムの解説	13
5.2.1	配列 <code>adc</code> の値を 1 つの変数にまとめる	13
5.2.2	AD コンバータから受け取った値を電圧の値に変換	14
5.2.3	電圧を温度に変換	14
5.3	プログラムの実行	15

1 課題

1.1 課題説明

ブレッドボードに LM61BIZ（温度センサー）と MCP3204（AD コンバータ）を差し込み、Raspberry Pi と接続して温度を取得します。



fritzing

2 機材説明

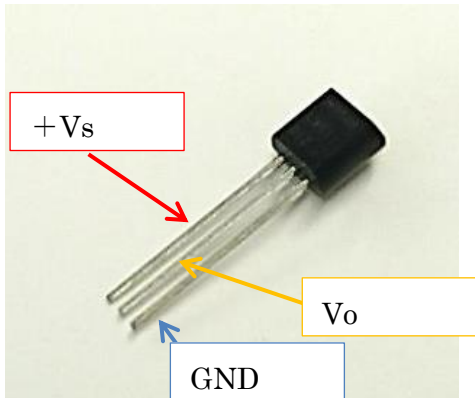
この課題は以下の機材を使用します。手元にあることを確認してください。

機材	個数
Raspberry Pi	1
ブレッドボード	1
LM61BIZ（温度センサー）	1
MCP3204（AD コンバータ）	1
ジャンパー線（オスーオス）	7
ジャンパー線（オスーメス）	6

2.1 ブレッドボード

課題 1 を参照

2.2 LM61BIZ

データシート															
http://akizukidenshi.com/download/LM61biz.pdf															
特徴 (P. 1 : 概要, 主な仕様を参照)															
- 温度センサー - 計測範囲 : $-25^{\circ}\text{C} \sim +85^{\circ}\text{C}$ (最大 $\pm 3.0^{\circ}\text{C}$) - 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)															
- 動作電圧 ($+V_s$) : $+2.7\text{V} \sim +10\text{V}$ ※ Raspberry Pi は 3.3V の電圧を供給可能 - 出力電圧 : $(+V_s + 0.6\text{V}) \sim -0.6\text{V}$ - 出力電流 : 10mA															
端子 (P. 2 : ピン配置図)															
センサーの平な面を上と見ます。															
<table border="1"> <thead> <tr> <th>端子名</th><th>説明</th></tr> </thead> <tbody> <tr> <td>$+V_s$</td><td>電源からの電圧を入力</td></tr> <tr> <td>V_o</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	$+V_s$	電源からの電圧を入力	V_o	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														
$+V_s$	電源からの電圧を入力														
V_o	センサーが測定した値 (電圧) を出力														
GND	不要な電圧を放出														
※接続時, 端子の向きに注意!! やけどするくらい熱くなります。															
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。 $V_o = (+10\text{mV}/^{\circ}\text{C} \times T^{\circ}\text{C}) + 600\text{mV}$ $+10\text{mV}/^{\circ}\text{C}$: (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 - 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	$+300\text{ mV}$														

2.3 MCP3204

データシート																							
http://akizukidenshi.com/download/ds/microchip/MCP3204.pdf																							
特徴 (P.1 : Features, Description を参照)																							
・ AD コンバータ ・ 分解能 : 12bit (4096 段階) ・ チャンネル数 : 4 ・ 通信方式 : SPI																							
電圧・電流 (P.1 : Features を参照)																							
・ 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 (0.5MHz ~ 1MHz)																							
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 ~ CH3</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> <tr> <td>NC</td><td>使用しません</td></tr> </tbody> </table>	端子名	説明	CH0 ~ CH3	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	NC	使用しません	
端子名	説明																						
CH0 ~ CH3	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																						
NC	使用しません																						

コンフィグレーションビット (P. 19 : TABLE 5-1: CONFIGURATION BITS FOR THE MCP3204 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。

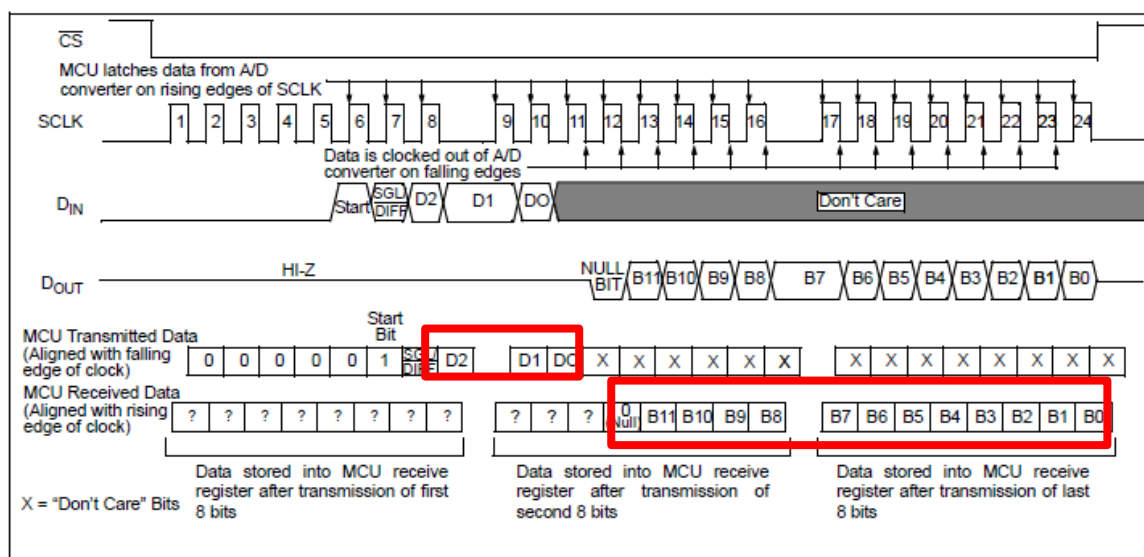
TABLE 5-1: CONFIGURATION BITS FOR THE MCP3204

Control Bit Selections				Input Configuration	Channel Selection
Single/Diff	D2*	D1	D0		
1	X	0	0	single-ended	CH0
1	X	0	1	single-ended	CH1
1	X	1	0	single-ended	CH2
1	X	1	1	single-ended	CH3
0	X	0	0	differential	CH0 = IN+ CH1 = IN-
0	X	0	1	differential	CH0 = IN- CH1 = IN+
0	X	1	0	differential	CH2 = IN+ CH3 = IN-
0	X	1	1	differential	CH2 = IN- CH3 = IN+

* D2 is a "don't care" for MCP3204

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。B11~B0 の 12bit の値をデバイスの出力として受信します。

出力値 (P. 17 : Reference Input を参照)

MCP3204 は入力されたアナログ値を以下の式で変換します。

$$\text{Digital Output Code} = \frac{4096 \times V_{IN}}{V_{REF}}$$

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)

4096 : 分解能

計算例 : 基準電圧 3.3V, 入力される電圧が 2V の時

1. 基準の電圧を分解能で割る : $3.3/4096 = 0.0008056640625$
2. 入力された電圧を[1.]の値で割る : $2/ 0.0008056640625 = 2482.4$
3. AD コンバータの結果として 2482 が出力される

2.4 ジャンパー線

課題 1 を参照

3 SPI

3.1 SPI（シリアル・ペリフェラル・インタフェース）について

旧モトローラ社が開発したシリアル通信規格です。

使用する信号線は MOSI, MISO, SCLK, CE の 4 本必要ですが、I2C と比べて通信速度が速いことが特徴です。

3.2 SPI の使い方

3.2.1 Python での使い方

- import

Python プログラムで SPI を使用するには [spidev] を import します。

```
import spidev # spidev をインポート
```

- spidev の初期宣言

```
spi = spidev.SpiDev() # spi の初期宣言
spi.open(0, 0)        # spi のデバイスへの接続
```

※spi.open(spi の番号, CE 番号)

- spi の番号はデフォルトでは 0 になります。
- CE 番号は接続する AD コンバータに接続するコンピュータの CE 番号によってかわります。

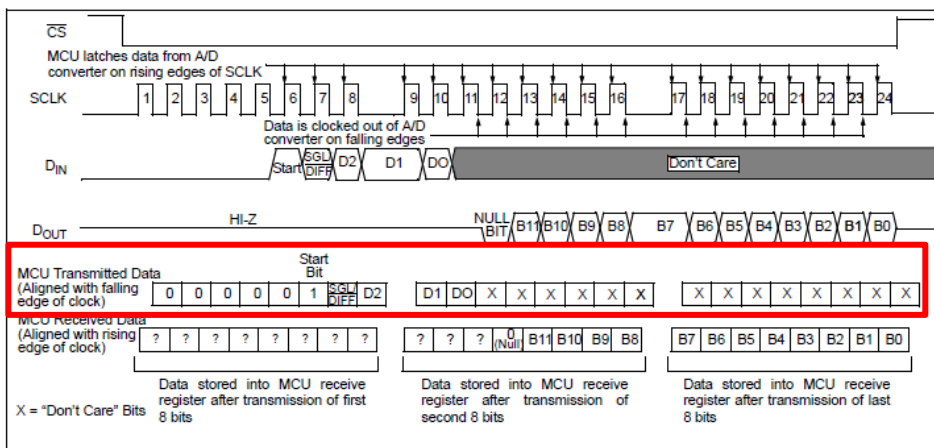
今回の講習会では以下の関数を使用して値のやり取りを行います。

- xfer2（コンフィグレーションビット）

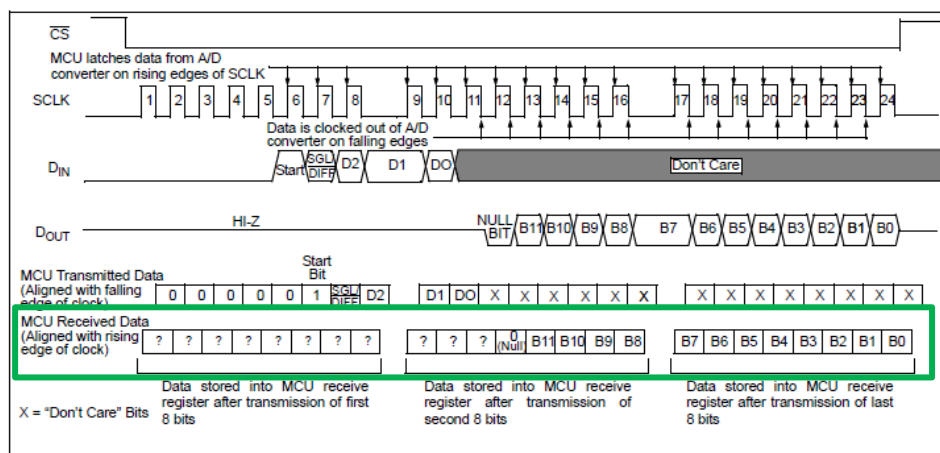
引数に指定したコンフィグレーションビットを参照し AD コンバータからの値を受け取る。

コンフィグレーションビット	接続している AD コンバータのコンフィグレーションビット
---------------	-------------------------------

MCP3204 のコンフィグレーションビットは下図赤枠内の [Start Bit] ~ [D0] に格納されています。

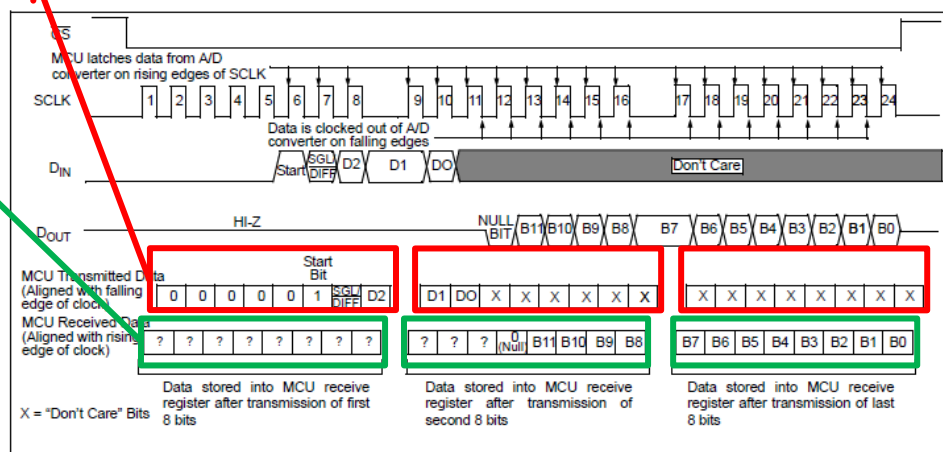


AD コンバータから返される値は下記の下図緑枠の [B11] ~ [B0] に格納されています。



使用例 (CH0 に接続)

`adc = spi.xfer2([6, 0, 0])` # (adc には値が 3 要素のリストで格納されます。)



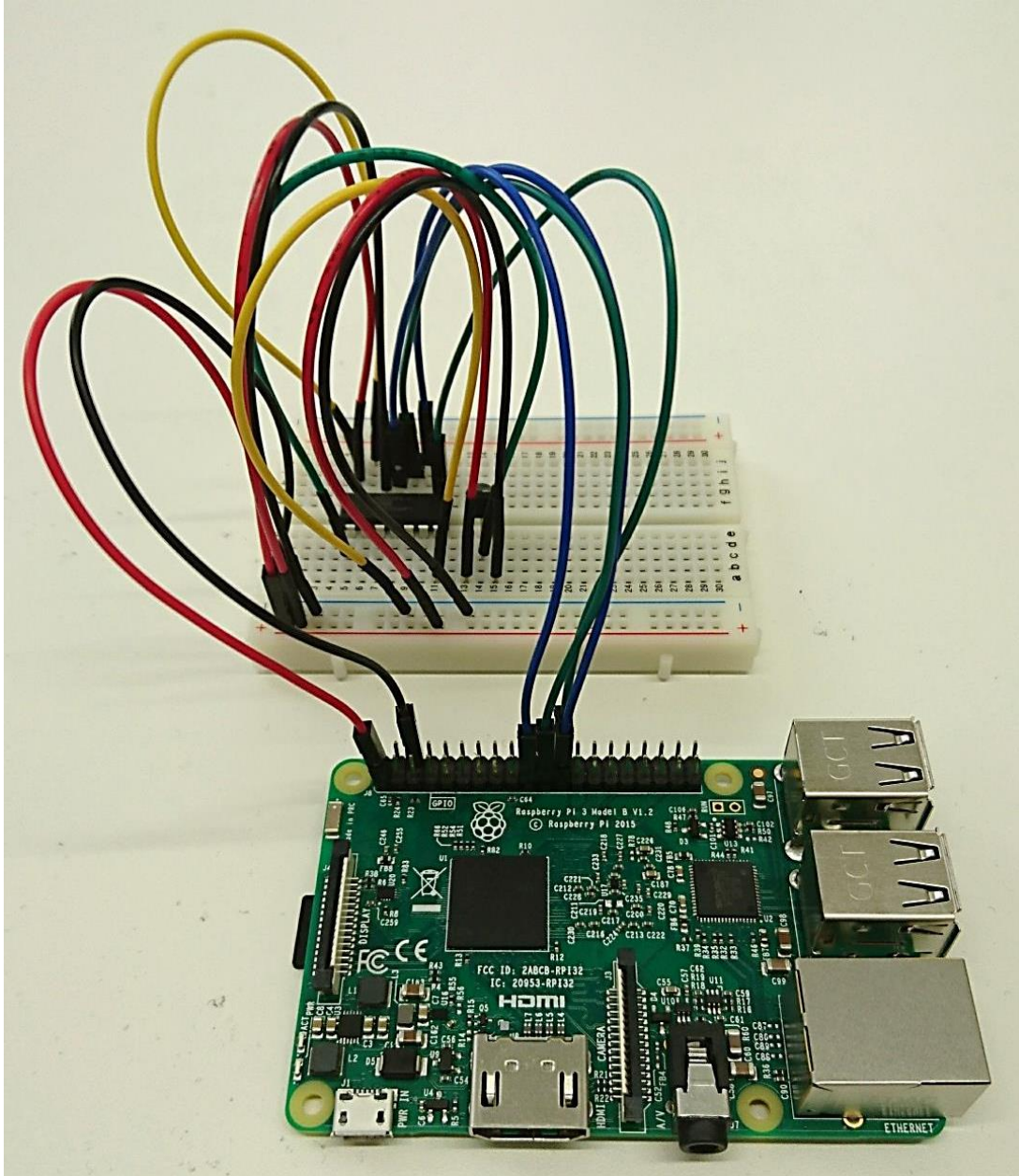
※コンフィグレーションビット、AD コンバータからの返り値の格納場所は AD コンバータごとに変わります。使用する際はデータシートを確認してください。

4 回路作成

ブレッドボードに LM61BIZ と MCP3204 を差し回路を作成します。

完成図は以下になります。

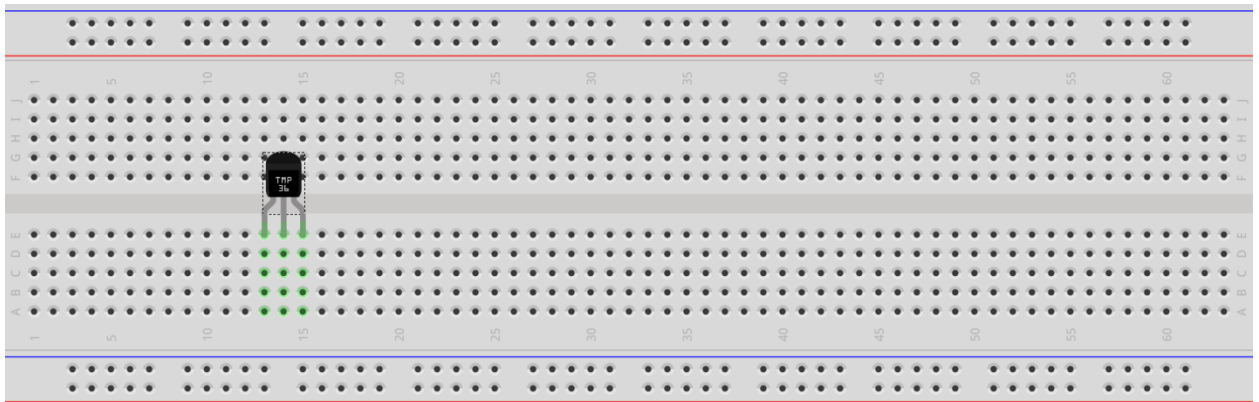
※接続作業は必ず Raspberry Pi の電源を切ってから行ってください



4.1 LM61BIZ を差し込む

LM61BIZ をブレッドボードに差し込みます。

Vs 側を E13, GND 側を E15 に差し込んでください。



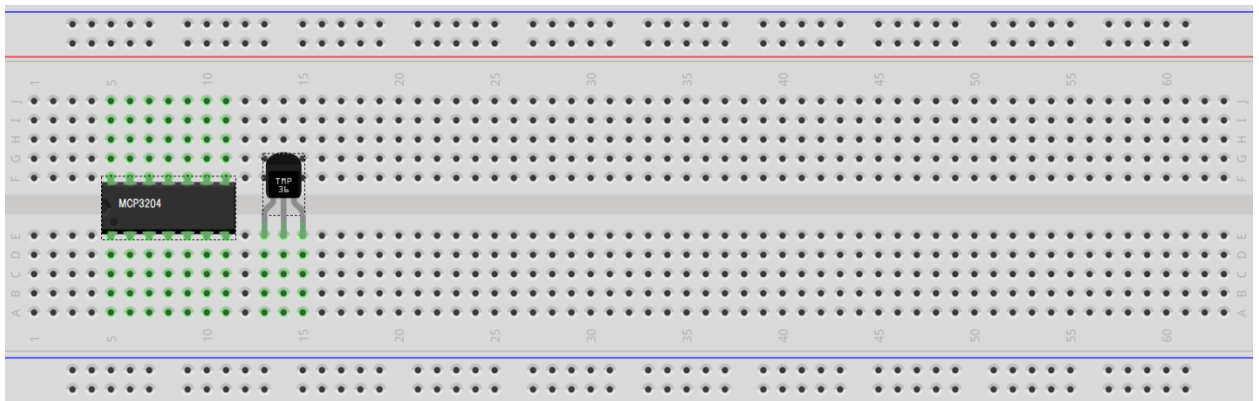
fritzing

4.2 MCP3204 を差し込む

MCP3204 をブレッドボードに差し込みます。

左下ピンが E5, 右上ピンが F11 になるように差し込んでください。

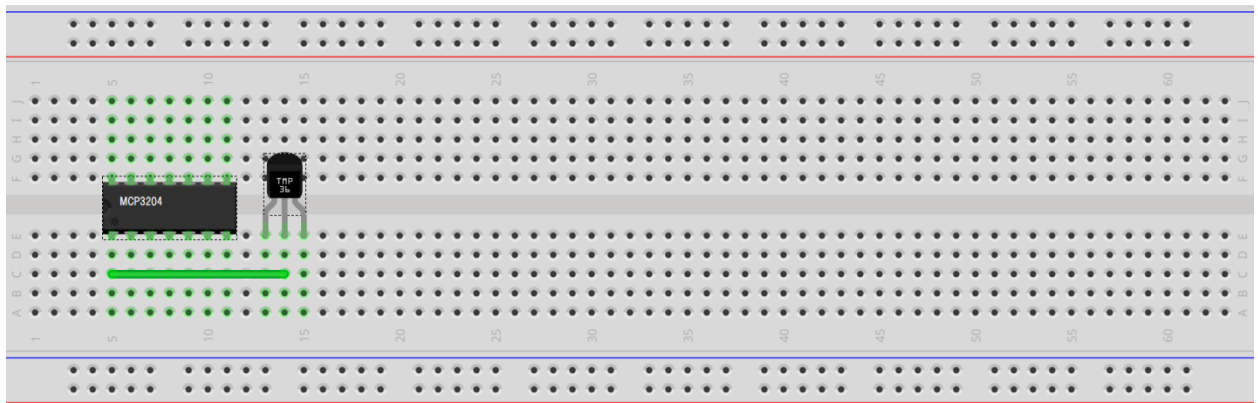
くぼみがあるほうが左側になります。



fritzing

4.3 MCP3204 と LM61BIZ を接続

MCP3204 「CH0 ピン」と LM61BIZ 「Vout ピン」を接続します。



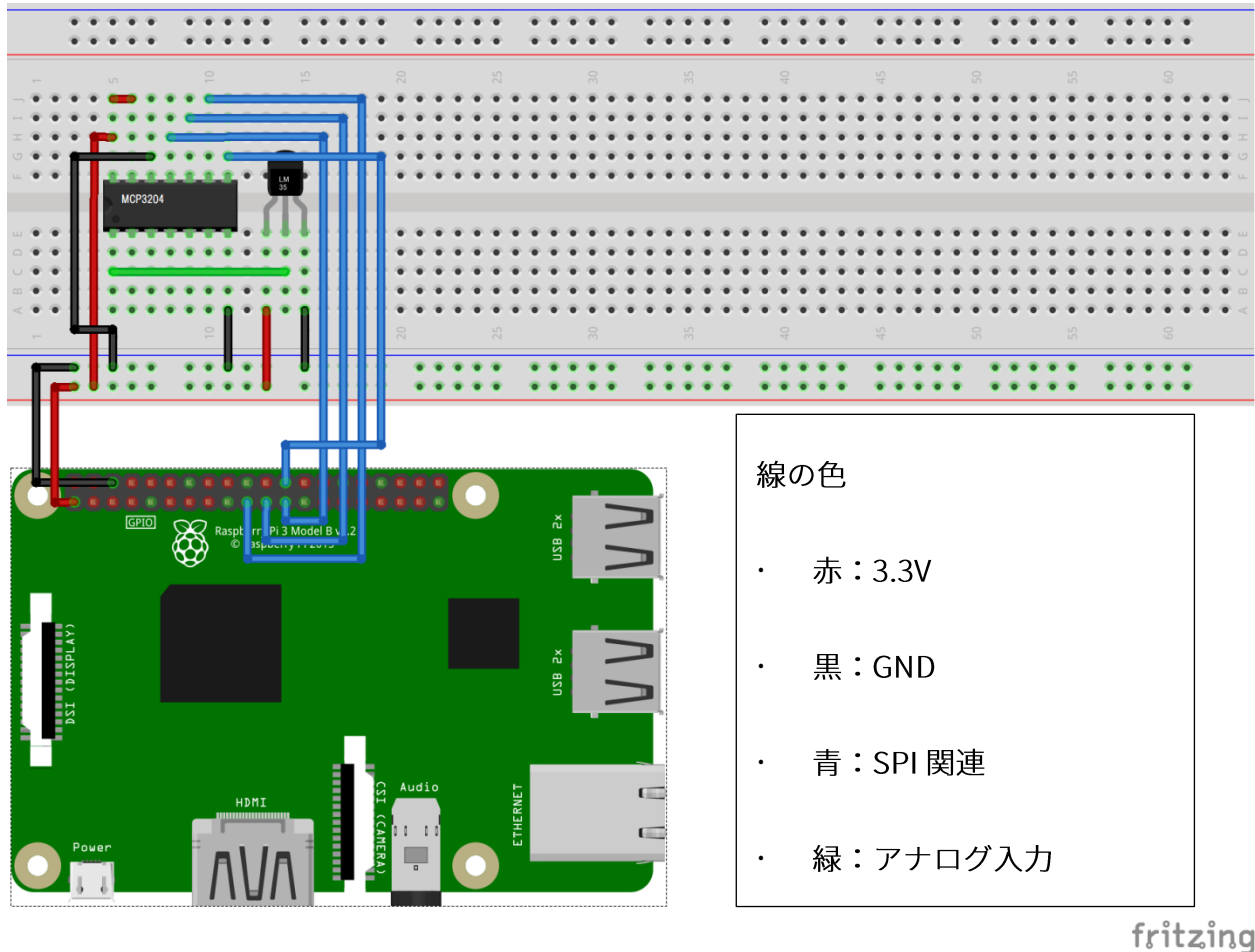
fritzing

4.4 Raspberry Pi と MCP3204 と LM61BIZ を接続

Raspberry Pi と MCP3204, LM61BIZ を接続します。

※Raspberry Pi の 3.3V と GND を直接接続しないでください。

※ブレッドボードと Raspberry Pi の接続は最後に行ってください。



各ピンの接続は以下になります。

MCP3204	Raspberry Pi
CLK	SCLK (GPIO11) (PIN 番号 23)
DOUT	MISO (GPIO9) (PIN 番号 21)
DIN	MOSI (GPIO10) (PIN 番号 19)
CS/SHDN	CE0 (GPIO8) (PIN 番号 24)
VDD, VREF	3.3V (PIN 番号 1)
AGND, DGND	GND (PIN 番号 6)

LM61BIZ	Raspberry Pi
Vs	3.3V (PIN 番号 1)
GND	GND (PIN 番号 6)

5 プログラム

温度を取得するプログラムを作成します。

5.1 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「03_MCP3024.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。

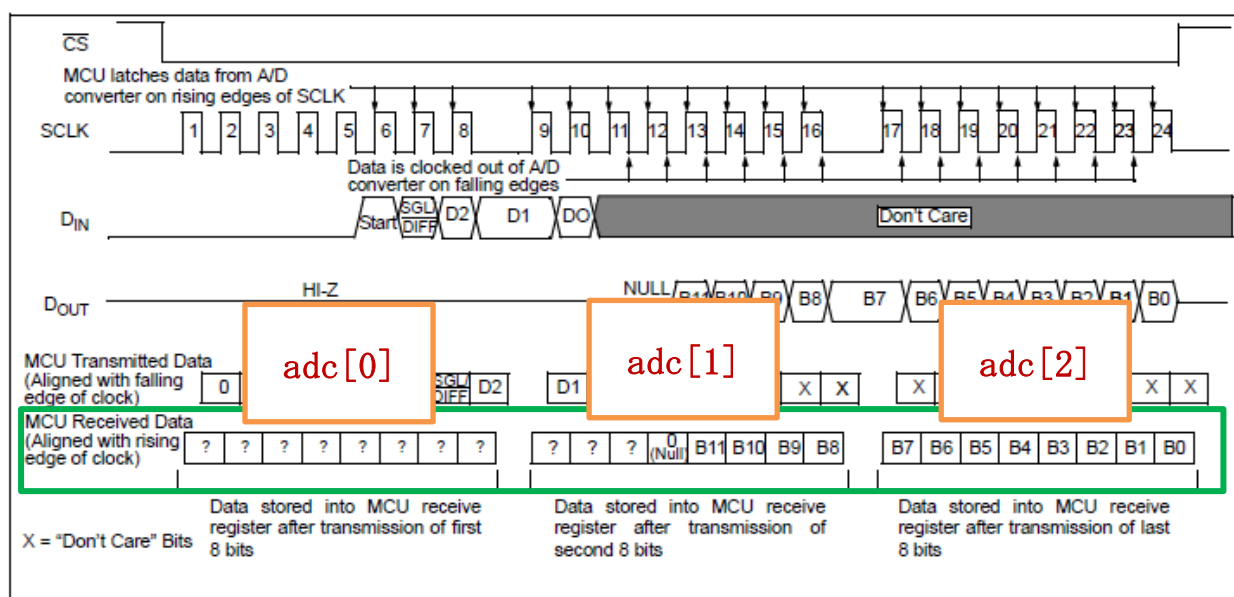
https://rtc-fukushima.jp/wp/wp-content/uploads/2019/01/03_MCP3024.zip

5.2 プログラムの解説

5.2.1 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 (adc[0], adc[1], adc[2]) 3 つの配列に格納されて受信されます。

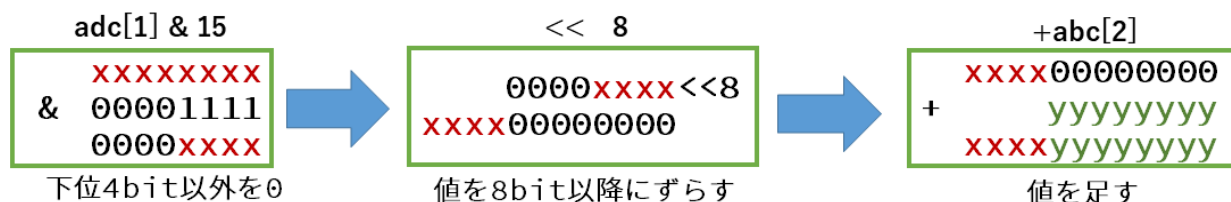


この中で必要なデータは B11~B0 の 12bit のみでほかの要素は使用しませんので、必要な要素のみ抜き出します。

受け取った値は以下の計算で一つの変数に格納されます。

$$((adc[1] \& 15) \ll 8) + adc[2]$$

処理の内容は下図になります。



- ・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。

5.2.2 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 4096，入力された値が 1V の場合
 1. 基準の電圧を分解能で割る： $3.3 / 4096 = 0.000805664063$
 2. 入力された値を[1.]の値で割る： $1 / 0.000805664063 = 1241.21212$
 3. ADコンバータの結果として 1241 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。

5.2.3 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>

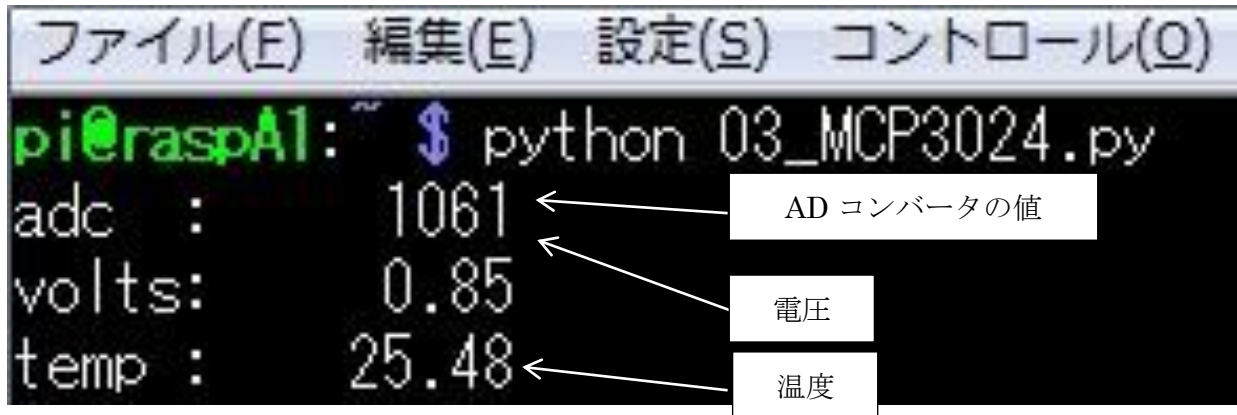
出力する電圧は下記の式で求めることができます。

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

これらを参考に温度を求めてください。

5.3 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 03_MCP3024.py
adc : 1061
volts: 0.85
temp : 25.48
```

AD コンバータの値

電圧

温度