

5コマ目 データシートについて



データシートとは



データシートとは

- データシートとはデバイスを使用するための必要な情報が記載されているドキュメント
- データシートには、以下の3つが記載
 - デバイス概要
 - このデバイスで出来ること
 - 仕様
 - サイズ，電流・電圧，ピン配置
 - 使用方法
 - 通信方式，値の計算方法など

DRV8830使用DCモータドライブキット
マイコンで正転・逆転・ブレーキ・速度
のコントロールが出来ます。*

*マイコン、モータは、付属していません。

■特徴■

- ★DCモータモータドライブICのDRV8830を使用しています。
- ★マイコンのI2CインターフェイスでDCモータの正転・逆転・ブレーキ・速度の切り替、速度コントロールが出来ます。当社FA-130RA-2270モータに適しています。
- ★I2Cアドレスの設定で、最大9キットまで制御が出来ます。
- ★幅広い動作電源電圧 2.75V~6.8V(マイコンの電源電圧と同じにする必要があります。)
- ★センサ抵抗による電流制限回路(1A)内蔵です。
- ★1過電流保護・短絡保護・低電圧検出防止・加熱保護機能内蔵です。
- ★内部レジスタを参照する事で、障害内容を知ることが出来ます。またモニター用 FAULT LED付きです。
- ★電池駆動などで電源電圧が変化しても、モータ速度を一定に保ちます。

■仕様■

動作電圧: 2.75V~6.8V(マイコンの電源電圧と同じ電圧にする必要があります。)

最大駆動電流: 1A

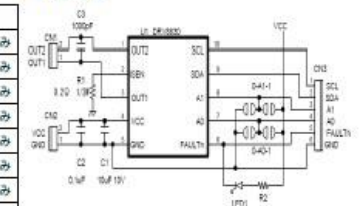
■部品表■

番号	品名	規格等	備考
U1	モータドライブIC	DRV8830	実装半田付け済み
C1	積層セラミックコンデンサ	10μF 10V	実装半田付け済み
C2	積層セラミックコンデンサ	0.1μF 50V	実装半田付け済み
C3	積層セラミックコンデンサ	1000μF 16V	実装半田付け済み
R1	チップ抵抗	0.2Ω 0.5W	実装半田付け済み
R2	チップ抵抗	1KΩ 0.1W	実装半田付け済み
LED1	チップLED	赤色	実装半田付け済み
CN1,2	端子台	小型 2ピン	
CN3	ピンヘッダ	5P	
おまけ	積層セラミックコンデンサ	1000μF 50V	モータノイズ用

■製作■

ほとんどの部品は実装半田付け済みです。
端子台とピンヘッダを半田付けするのみです。
端子台は、ケーブル接続面が基板の外側になる様に半田付けしてください。おまけの1000μFコンデンサはモータ端子間に付けてください。

■回路図■



http://akizukidenshi.com/download/ds/akizuki/AE-MOTOR8830_manual.pdf

データシートとは

- 電子回路作成時やデバイスを使う場合，
まず始めに，データシートを見て使い方を調べる
- データシート内の情報は多いので，
このコマでは必要な部分を紹介する

データシートの確認箇所

1. 特徴

- デバイスのおおざっぱな性能を説明

2. 通信方式

- デバイスへアクセスする通信規格

3. 端子又は回路

- デバイスのハードウェア的なつながり

4. アクセス方法

- 通信をするとき，デバイスの選択方法やデータのアクセス方法

5. 入出力する値

- 値の書き込み，読み込みの方法，値の意味

特徴



特徴

- そのデバイスの主な性能が記載
 - デバイスの内容：そのデバイスが何のために使われるかの情報
 - 電流・電圧：デバイスを使用するときの電流・電圧の値
電子回路を作成するときに必要

※今回の講習会では，FaBo側で考えているので考慮しない。

- **動作電源電圧範囲：使用できる電圧の範囲**
- **最大連続駆動電流：過熱することなく使用できる電流の最大値**
- 通信方式：データの通信の方法

通信方式

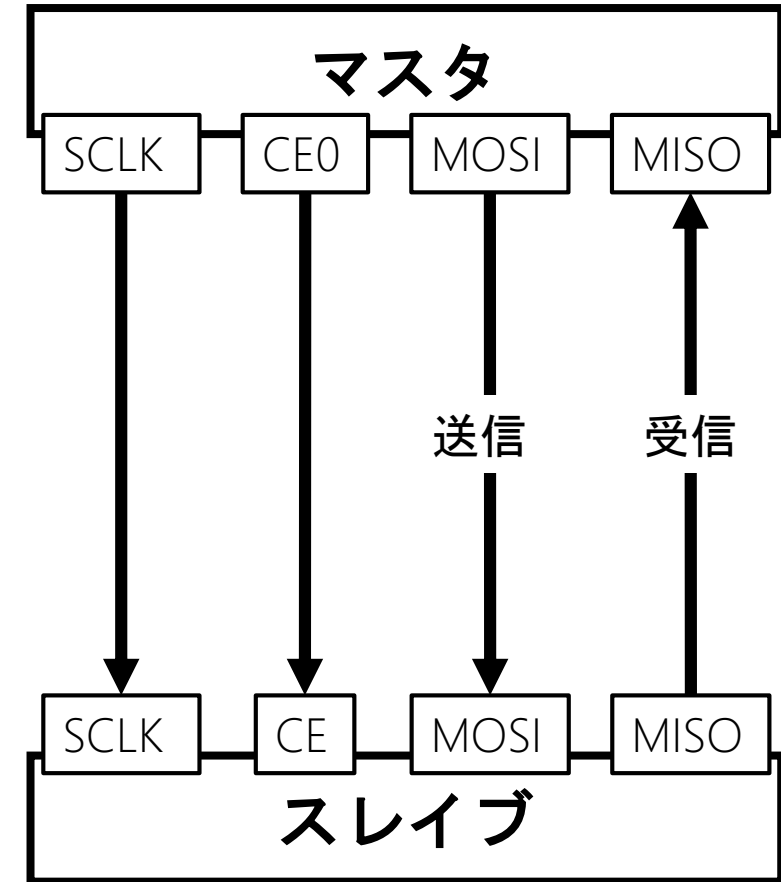


通信方式

- デバイスとの間の通信方法様々な種類があるため、通信方式により確認するものが変わるため注意する
 - SPI
 - SCLKと単方向のMOSI, MISO, CEの4本の信号線で通信する同期式のシリアル通信方法
 - I2C
 - SCLと, 双方向のSDAの2本の信号線で通信する同期式のシリアル通信方法
 - デジタル入出力
 - 値をデジタル (ON/OFF また 1/0) で入出力する方法
 - アナログ入力
 - 値をアナログ (量などの変化を連続的に表したものの) で入出力

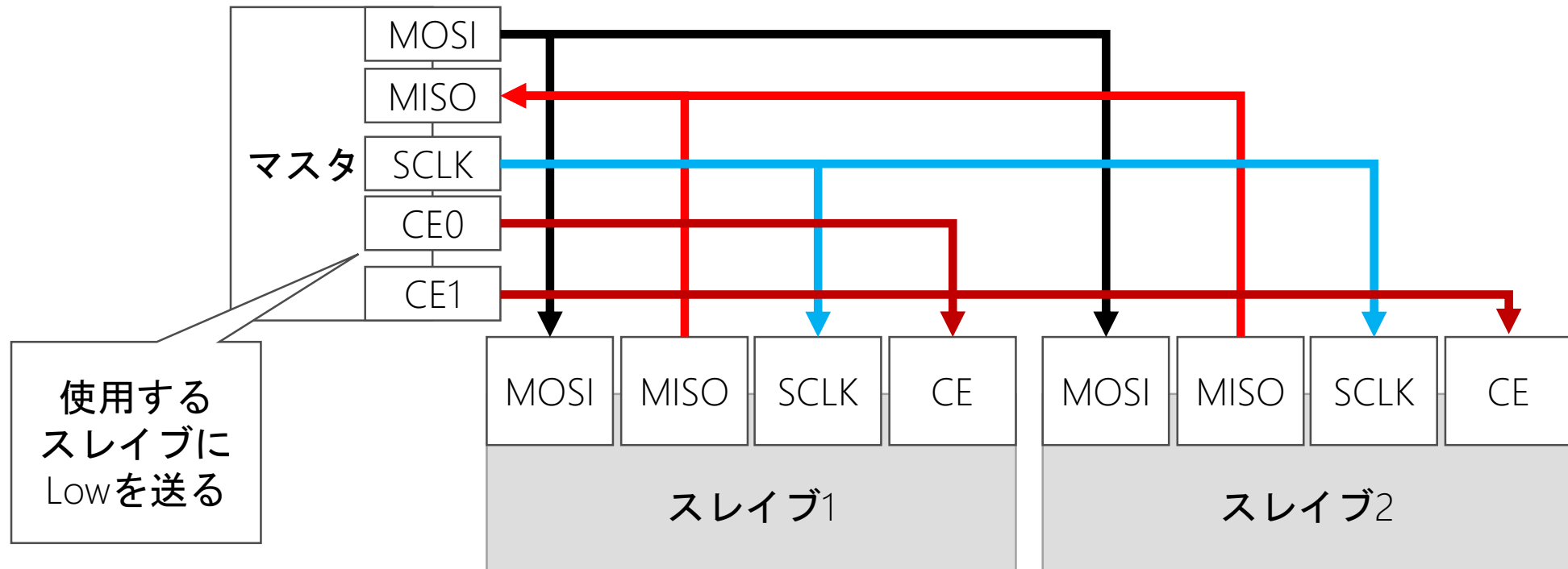
SPI

- モトローラ社が開発した通信規格
- 4本線で通信
 - MOSI：データを送信
 - MISO：データを受信
 - SCLK：デバイス間の同期
 - CE：通信するデバイスを選択
- 複数のデバイスに並列して接続可能



SPI

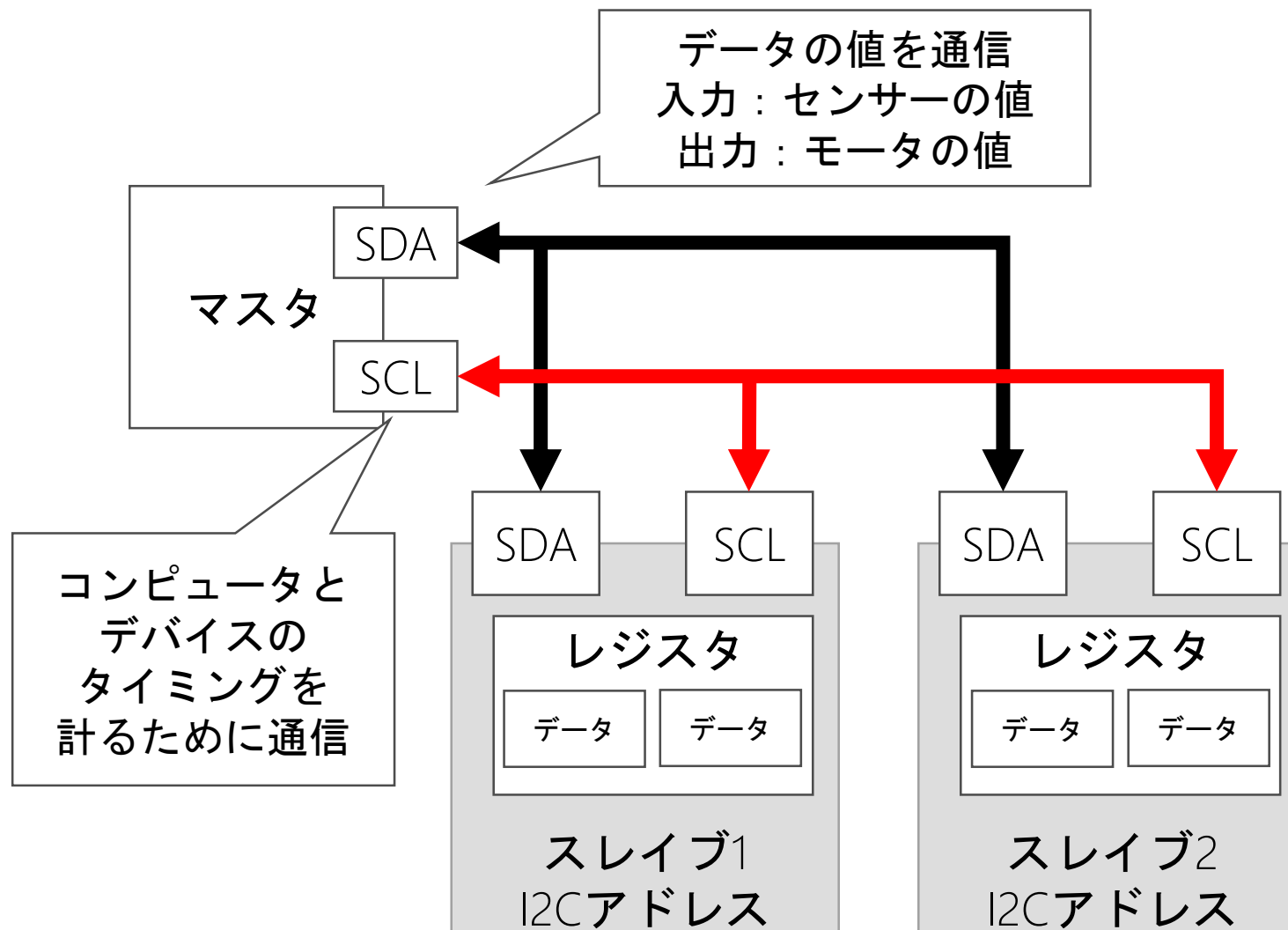
- 複数のデバイスへの接続方法



I2C

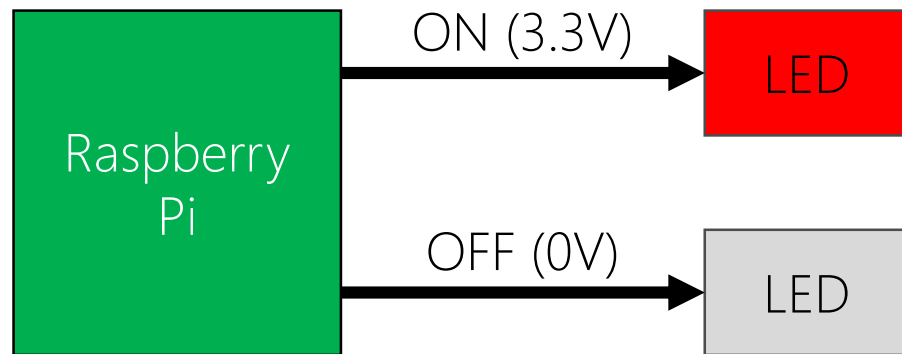
- フィリップ社が開発した通信規格
- SDA, SCLの2本信号線でデバイスを制御
 - SDA：データの通信
 - SCL：同期の通信
- 複数のデバイスに並列して接続可能

I2C

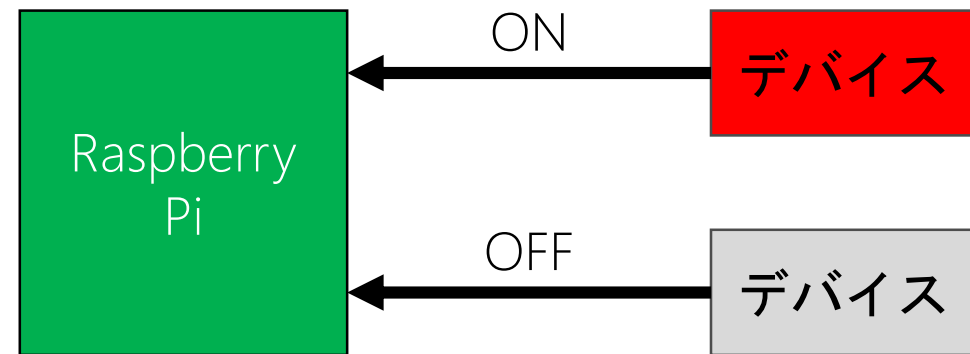


デジタル入出力

- デジタル出力
 - 値をON(1)かOFF(0)で出力
 - Raspberry Piだと電圧を「3.3V出力」「0V出力」の2種類
- デジタル入力
 - 値をON(1)かOFF(0)かを識別する



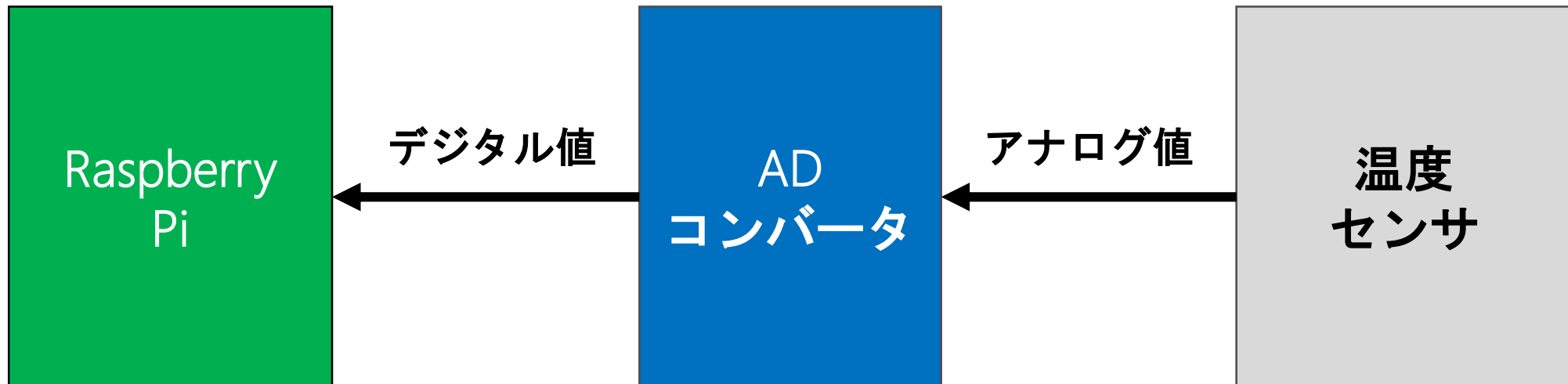
デジタル出力



デジタル入力

アナログ入力

- アナログ入力
 - Raspberry Piではデジタル入出力のみなので、ADコンバータを使用してデジタル値に変更する



端子，回路



端子

- デバイスの入出力を行う物理的な場所
 - 通信を行うための端子（SDAやMISOなど）や**アドレス設定の端子**などがあるため確認

例) DRV8330（モータドライバ）

CN1モータ接続端子

CN1	OUT1	正転時に「+」
	OUT2	正転時に「-」

CN2モータ接続端子

CN2	VCC	電源入力
	GND	GND

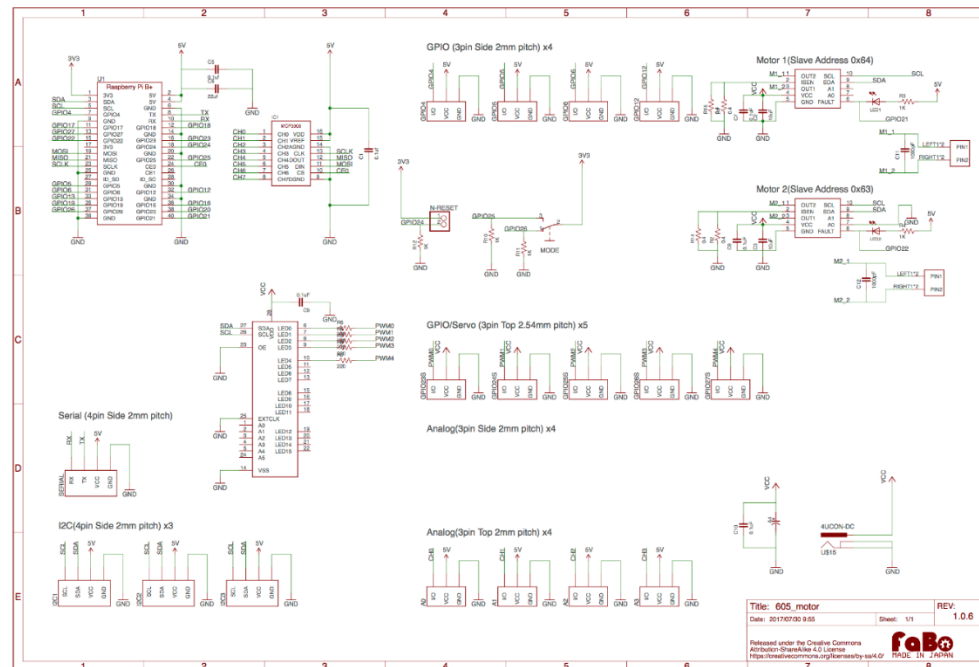
CN3マイコン接続端子

CN3	SCL	クロックライン
	SDA	データライン
	A1	アドレス設定1
	A2	アドレス設定2
	FAULTh	障害状態の出力
	GND	GND

アドレス設定の端子がある

回路

- この講習会で使用するFaBoのような拡張ボードを使用する場合、デバイスが組み込まれているので、アドレスや設定の確認のために回路を確認する必要がある



FaBo回路図：
<http://www.fabo.io/605.html>

アクセス方法

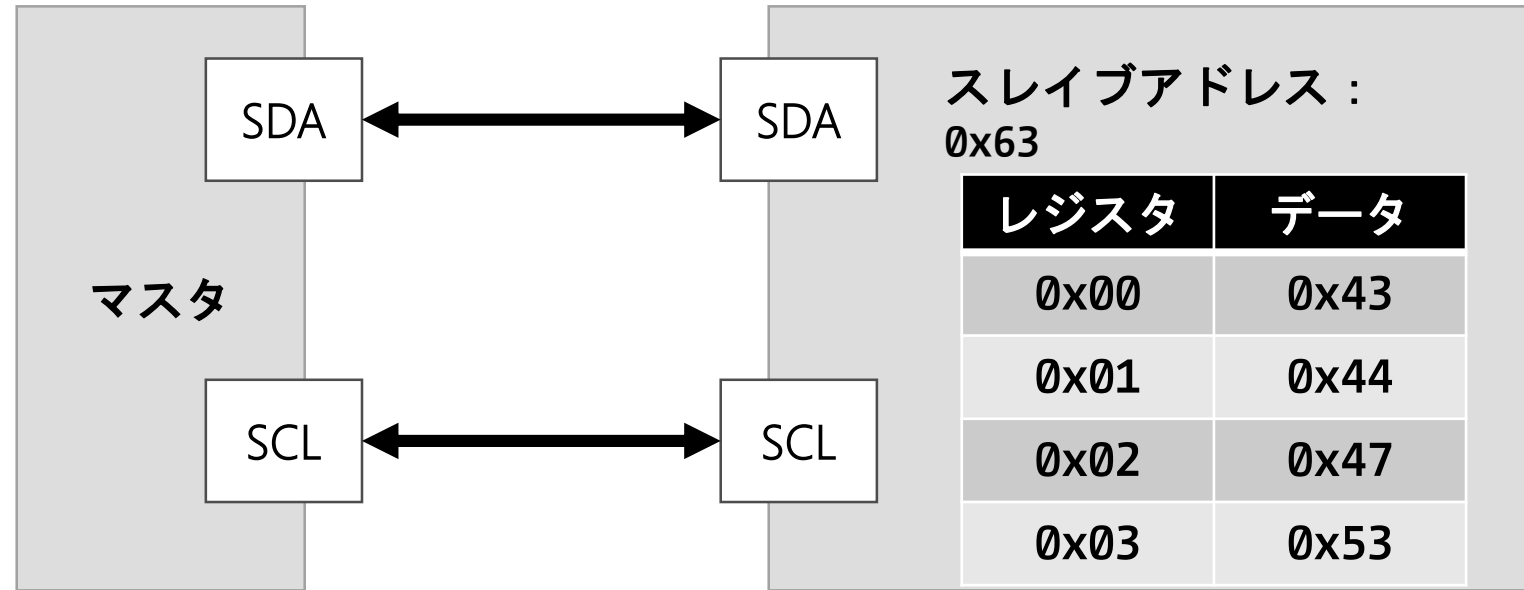


アクセス方法

- デバイスへのアクセス方法，データの入出力先の決定方法
 - 通信方式により確認項目が変わる
- I2C：スレーブアドレス，レジスタ番号
- SPI：CE番号，コンフィギュアービット，レジスタなど
- デジタル入出力：GPIOピン番号

I2Cのアクセス方法

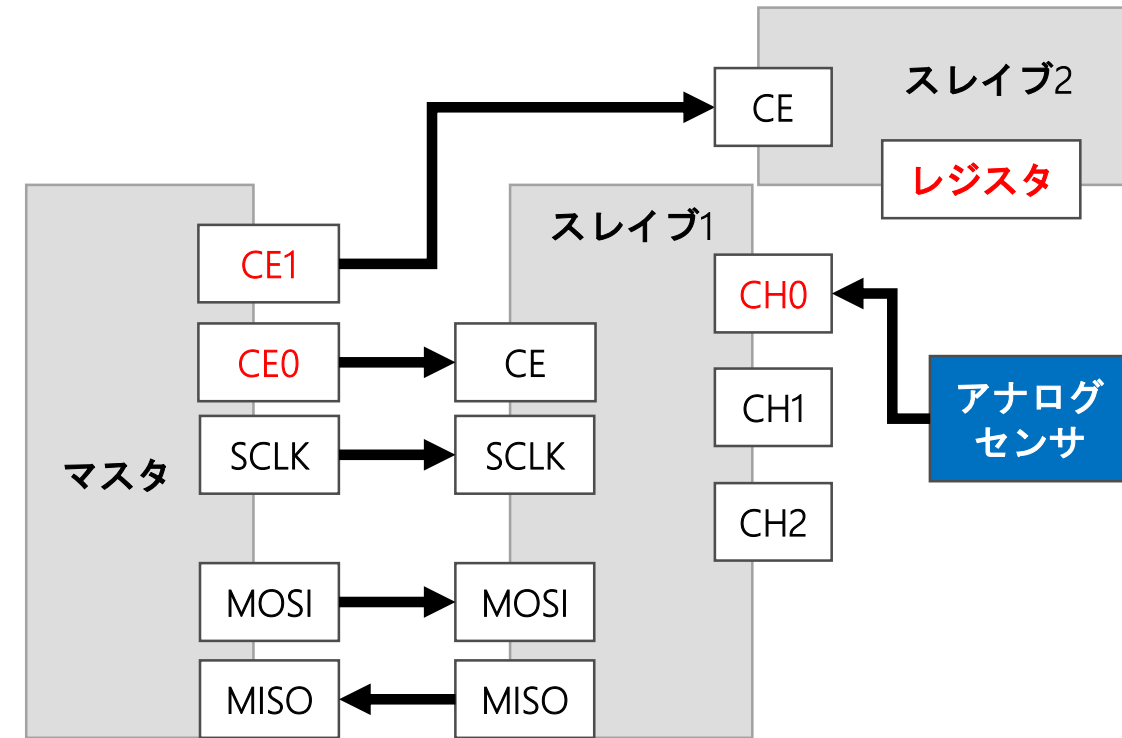
- デバイスの選択
 - スレイブアドレス
 - デバイスの番号
- データの入出力先
 - レジスタ
 - データの格納場所



- I2Cはスレイブアドレスでどのデバイスにアクセスするか決め、レジスタでどのデータを取得するか、格納するかを決める

SPIのアクセス方法

- デバイスの選択
 - CE番号
 - マスタのCE番号で使用するデバイスを選択
- データの入出力先
 - コンフィギュアビット
 - どのCHを使用しているかの値
 - レジスタ番号
 - データを格納してある場所



- SPIはCE番号でデバイスを選択，コンフィギュアビットまたはレジスタ番号で値の入出力

デジタル入出力のアクセス方法

- デバイスの選択
 - Raspberry Piでデジタル入出力を行うにはGPIOピンの指定が必要
 - GPIOピンを指定してそのピンで入出力をする
 - GPIO2～GPIO27までである
 - データの入出力先
 - GPIOピン番号
- LED ← どのGPIOピンを使用するか
- デジタル入出力はGPIOピンでデバイスの選択とデータの入出力先を決定

Pin 1	Pin 2
+3V3	+5V
GPIO2 / SDA1	+5V
GPIO3 / SCL1	GND
GPIO4	TXD0 / GPIO 14
GND	RXD0 / GPIO 15
GPIO17	GPIO 18
GPIO27	GND
GPIO22	GPIO 23
+3V3	GPIO 24
GPIO10 / MOSI	GND
GPIO9 / MISO	GPIO 25
GPIO11 / SCLK	CE0# / GPIO8
GND	CE1# / GPIO7
GPIO0 / ID_SD	ID_SC / GPIO1
GPIO5	GND
GPIO6	GPIO12
GPIO13	GND
GPIO19 / MISO	CE2# / GPIO16
GPIO26	MOSI / GPIO20
GND	SCLK / GPIO21
Pin 39	Pin 40

<https://elinux.org/File:Pi-GPIO-header.png>

入出力する値



入出力する値

- デバイスの入出力値を確認
- デバイスへ書き込む値
 - データシートを見て，デバイスへの値の書き込み方を確認する
 - 値を2進数で考え，下位ビットと上位ビットでは意味が異なることもある
- デバイスから受け取る値
 - デバイスから値を取得したとき，その値がそのまま使えないこともあるので，目的の値への変換を確認する

入力する値

- デバイスを動作させるときの値を確認

例) DRV8830 (モータドライバ) でのモータへ電圧をかける時


Controlレジスタ

D7-D2	D1	D0
VSET[5..0]	IN2	IN1

動作論理表

IN2	IN1	モータの動作
0	0	惰走
0	1	逆転
1	0	正転
1	1	停止

電圧表

値(VSET[5..0])	出力電圧 [V]
0x06h	0.48
0x07h	0.56
	
0x3Fh	5.06

- 0,1bit目 : モータの回転情報, 2~7bit目 : 電圧値

出力する値

- 受け取った値を目的の値に変換する計算式を確認

例) MCP3008の出力の値は以下の式で出力

EQUATION 4-2: DIGITAL OUTPUT CODE CALCULATION

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

Where:

V_{IN} = analog input voltage
 V_{REF} = analog input voltage

Digital OutPut Code：出力する値

V_{IN} ：CHから入力された電圧

V_{REF} ：端子VREFに接続された電圧

1024：分解能

データシート：<http://akizukidenshi.com/download/ds/microchip/mcp3008.pdf>

- 使用するには出力値を V_{IN} の値に直すことが必要