

# *RPi-GP10T*

# *RPi-GP10M*

*Raspberry Pi I2C 絶縁型デジタル入出力ボード*

## ユーザーズマニュアル

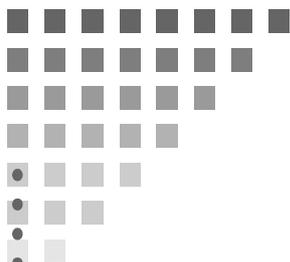
2021年7月

第2.0版



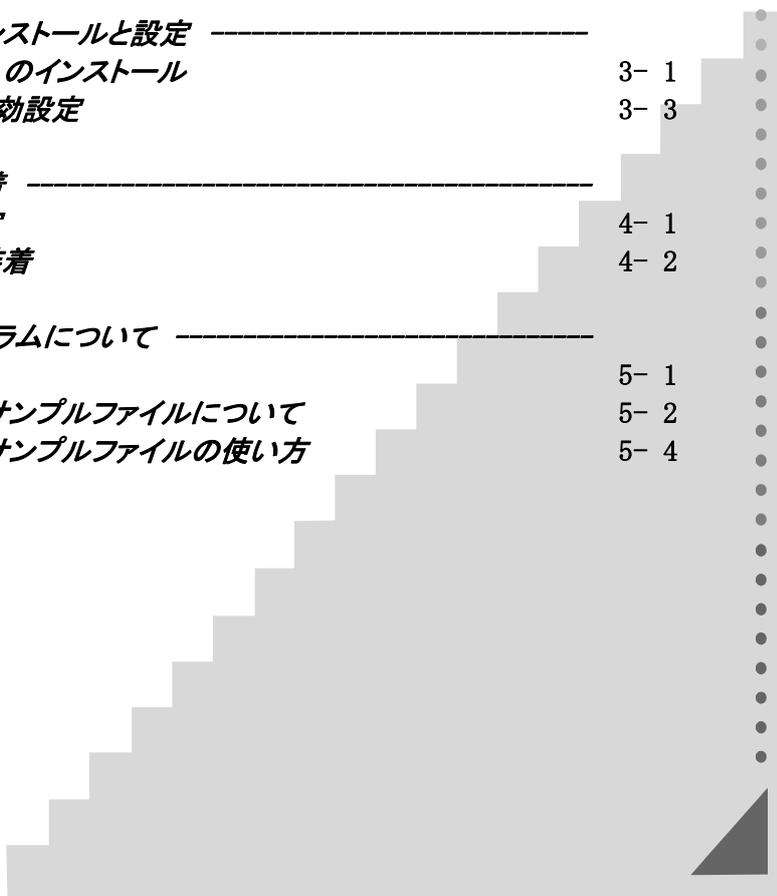
ラトックシステム株式会社

# Raspberry Pi I2C 絶縁型デジタル入出力ボード



安全にお使いいただくために

<b>第1章 はじめに</b> -----	
(1-1) 製品仕様	1- 1
(1-2) 添付品	1- 3
<b>第2章 各部名称と説明</b> -----	
(2-1) 基板構成	2- 1
(2-2) 各部説明	2- 2
(2-3) 入出力回路	2- 6
<b>第3章 Raspbian のインストールと設定</b> -----	
(3-1) Raspbian のインストール	3- 1
(3-2) I2C の有効設定	3- 3
<b>第4章 各種設定と装着</b> -----	
(4-1) 各種設定	4- 1
(4-2) 基板の装着	4- 2
<b>第5章 サンプルプログラムについて</b> -----	
(5-1) 準備	5- 1
(5-2) Python サンプルファイルについて	5- 2
(5-3) Python サンプルファイルの使い方	5- 4



## 安全にお使いいただくために

### ◆警告および注意表示◆

 <b>警告</b>	人が死亡するまたは重傷を負う可能性が想定される内容を示しています。
 <b>注意</b>	人が負傷を負う可能性が想定される内容および物的損害が想定される内容を示しています。

### **警告**

- 製品の分解や改造等は、絶対におこなわないでください。
- 無理に曲げる、落とす、傷つける、上に重いものを載せることはおこなわないでください。
- 製品が水・薬品・油等の液体によって濡れた場合、ショートによる火災や感電の恐れがあるため使用しないでください。
- 煙が出る、異臭や音がするなどの異常が発生したときは、ただちに電源を切り、すべての接続ケーブルを抜いたあと、弊社サポートセンターに連絡してください。

### **注意**

- 本製品は電子機器ですので、静電気を与えないでください。
- 高温多湿の場所、温度差の激しい場所、チリやほこりの多い場所、振動や衝撃の加わる場所、強い磁気を帯びたものの近くでの使用・保管は避けてください。
- 本製品は日本国内仕様です。日本国外で使用された場合の責任は負いかねます。
- 本製品は、医療機器、原子力機器、航空宇宙機器、輸送機器など人命に関わる設備や機器、および高度な信頼性を必要とする設備、機器での使用は意図されておりません。  
これらの設備、機器制御システムに本製品を使用し、本製品の故障により人身事故、火災事故などが発生した器制御システムに本製品を使用し、本製品の故障により人身事故、火災事故などが発生した場合、いかなる責任も負いかねます。
- 接続を誤ったことによる損失、逸失利益等が発生した場合でも、いかなる責任も負いかねます。

- 本紙の内容に関しましては、将来予告なしに変更することがあります。
- 本紙の内容につきましては万全を期して作成しておりますが、万一ご不審な点や誤りなどお気づきの点がございましたらご連絡くださいますようお願いいたします。
- 本製品は日本国内仕様となっており、海外での保守、およびサポートはおこなっておりません。
- 製品改良のため、予告なく外観または仕様の一部を変更することがあります。
- 本製品の保証や修理に関しては、本紙の保証書に記載されております。必ず内容をご確認の上、大切に保管 してください。
- 運用の結果につきましては責任を負いかねますので、予めご了承ください。
- 本製品の運用を理由とする損失、逸失利益等の請求につきましては、いかなる責任も負いかねますので、予めご了承ください。
- 本製品を廃棄するときは地方自治体の条例に従ってください。条例の内容については各地方自治体にお問い合わせください。
- 本製品および本紙に記載されている会社名および製品名は、各社商標または登録商標です。ただし本文中にはRおよびTMマークは明記しておりません。

# 第1章 はじめに

RPi-GP10T/GP10M は、Raspberry Pi の GPIO 40Pin (I2C) に接続する絶縁型のデジタル入出力拡張ボードで、以下の機能があります。

## ■ 機能概要

- ・ デジタルエキスパンダ：16bit, I2C TI TCA9535 を使用。
- ・ デジタル出力：8ch オープンドレイン方式 (最大 30V, 100mA/ch)。
- ・ デジタル入力：8ch 高耐圧ダイオード方式(最大 30V, 5V/10k $\Omega$  でプルアップ)。
- ・ ストローブ出力：1ch オープンコレクタ方式 (最大 30V, 50mA)。
- ・ トリガー入力:1ch 高耐圧ダイオード方式 (最大 30V, 5V/10k $\Omega$  でプルアップ)。
- ・ デジタル入出力およびストローブ出力、トリガー入力は、GPIO40pin 間と絶縁(耐圧 2.5kV)。

## (1-1) 製品仕様

### ハードウェア仕様

項目	仕様内容
名称	Raspberry Pi I2C 絶縁型デジタル入出力ボード 端子台モデル Raspberry Pi I2C 絶縁型デジタル入出力ボード MIL コネクタモデル
型番	RPi-GP10T：端子台モデル RPi-GP10M：MIL コネクタモデル
インターフェイス	GPIO40 ピン：I2C
I/O エクスパンダ	TI TCA9535
入力部	8ch 高耐圧ダイオード方式 VIH = 3.5V, VIL = 1.5V 最大耐圧 30V (負論理) 5V/10k $\Omega$ でプルアップ
出力部	8ch DMOS オープンドレイン方式 (負論理) 最大電流 100mA/ch (ON 時) 最大耐圧 30V (OFF 時) 標準 ON 抵抗 2 $\Omega$
トリガー入力	1ch 高耐圧ダイオード方式 VIH = 2.0V, VIL = 0.8V 最大耐圧 30V (負論理) 5V/10k $\Omega$ でプルアップ

ストローブ出力	1ch オープンコレクタ方式（負論理） 最大電流 50mA（ON時） 最大耐圧 30V（OFF時）
コモン電源	COM+ : 5~24V 入力
使用コネクタ	<p>■ RPi-GP10T Phoenix Contact PTSA 0,5/**-2,5-F もしくは互換品 適合電線 : AWG24~20 ストリップ長 9mm</p> <p>■ RPi-GP10M OMRON XG4A-2034 もしくは互換品 適合コネクタ OMRON XG5N-201-AWG26~22 フラットケーブル用 OMRON XG4M-2030</p>
絶縁耐圧	2.5kV（デジタル入出力と GPIO40 ピン間） ※デジタル入出力の各チャンネルは非絶縁
電源	+5V / +3.3V（GPIO ポートから給電）
消費電流	最大 5V/150mA, 3.3V/15mA
動作環境	温度 : 0~40℃、湿度 : 20~80%（ただし結露しないこと）
外形寸法	約 65 x 56.5 mm（突起部含まず）
重量	約 25g
生産	日本
保証期間	1年

## (1-2) 添付品

ご使用前に下記添付品が添付されているかをご確認願います。

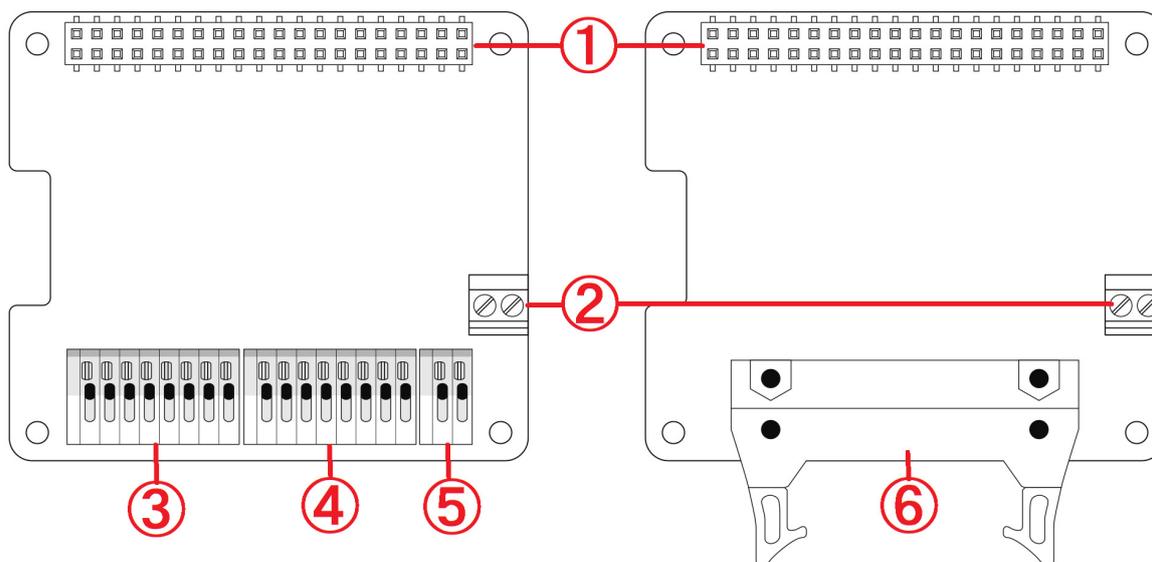
- RPi-GP10(T または M)
- GPIO 40P ピンヘッダー
- スペーサー（固定用 M2.6）x4
- ネジ（固定用 M2.6）x8
- 補足文書
- 保証書

# 第2章 各部名称と入出力回路

製品基板の各部名称とブロック図・入出力回路について説明します。

## (2-1) 基板構成

製品基板の各部名称は以下のとおりです。

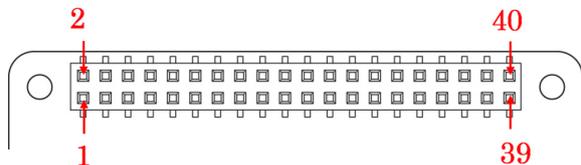


No	名称	No	名称
1	GPIO 40PIN コネクタ	2	COMMON 電源端子
3	出力端子 8ch	4	入力端子 8ch
5	ストローブ出力端子 トリガー入力端子 (各 1ch)	6	MIL コネクタ (20PIN)

## (2-2) 各部説明

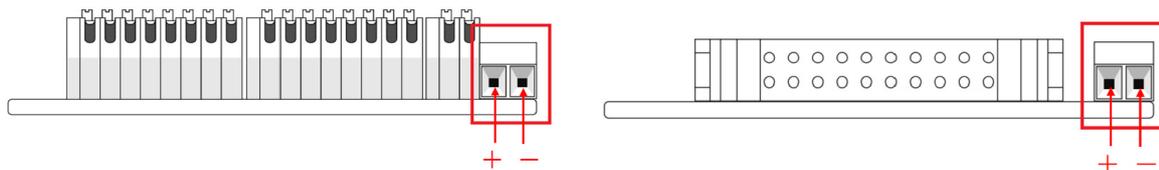
各部機能について説明します。

## 1. GPIO 40PIN コネクタ



PIN#	名称	備考	PIN#	名称	備考
1	3.3V	3.3V 電源	2	5V	5V 電源
3	I2C SDA1/GPIO 2	I2C / SDA1 TCA9535 制御用	4	5V	5V 電源
5	I2C SCL1/GPIO 3	I2C / SCL1 TCA9535 制御用	6	GND	GND
7	GPIO 4	(未使用)	8	UART TXD/GPIO 14	ストロープ出力 (デフォルト)
9	GND	GND	10	UART RXD/GPIO 15	トリガー入力 (デフォルト)
11	GPIO 17	(未使用)	12	GPIO 18	(未使用)
13	GPIO 27	絶縁電源制御	14	GND	GND
15	GPIO 22	(未使用)	16	GPIO 23	(未使用)
17	3.3V	3.3V 電源	18	GPIO 24	(未使用)
19	SPI0 MOSI/GPIO 10	(未使用)	20	GND	GND
21	SPI0 MISO/GPIO 9	(未使用)	22	GPIO 25	(未使用)
23	SPI0 SCLK/GPIO 11	(未使用)	24	SPI CE0/GPIO 8	(未使用)
25	GND	GND	26	SPI CE1/GPIO 7	(未使用)
27	I2C SDA0/GPIO 0	HAT_ID 読み込み用 I2C	28	I2C SCL0/GPIO 1	HAT_ID 読み込み用 I2C
29	GPIO 5	(未使用)	30	GND	GND
31	GPIO 6	(未使用)	32	GPIO 12	ストロープ出力 (オプション)
33	GPIO 13	トリガー入力 (オプション)	34	GND	GND
35	GPIO 19	(未使用)	36	GPIO 16	(未使用)
37	GPIO 26	(未使用)	38	GPIO 20	(未使用)
39	GND	GND	40	GPIO 21	(未使用)

## 2. COMMON 電源端子

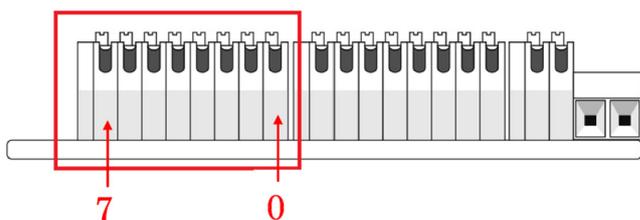


デジタル入出力用のコモン電源です。最大 30V の耐圧があります。

デジタル入出力・ストロブ出力・トリガー入力のプルアップ用電源として使用できます。

PIN#	機能
+	5~24V 入力
-	GND

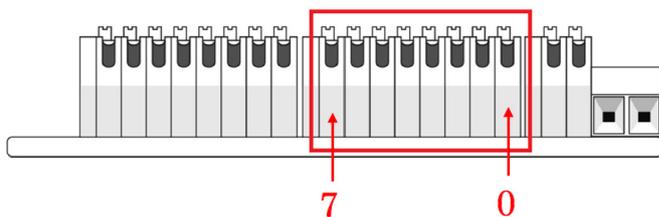
## 3. 出力端子 8ch



RPi-GP10T デジタル出力端子のピン配列と機能

PIN#	機能
0	デジタル出力 ch0
1	デジタル出力 ch1
2	デジタル出力 ch2
3	デジタル出力 ch3
4	デジタル出力 ch4
5	デジタル出力 ch5
6	デジタル出力 ch6
7	デジタル出力 ch7

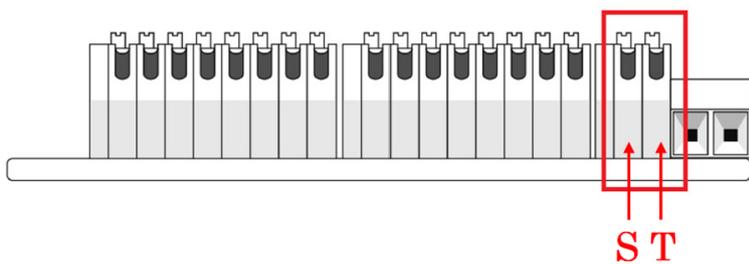
## 4. 入力端子 8ch



RPi-GP10T デジタル入力端子のピン配列と機能

PIN#	機能
0	デジタル入力 ch0
1	デジタル入力 ch1
2	デジタル入力 ch2
3	デジタル入力 ch3
4	デジタル入力 ch4
5	デジタル入力 ch5
6	デジタル入力 ch6
7	デジタル入力 ch7

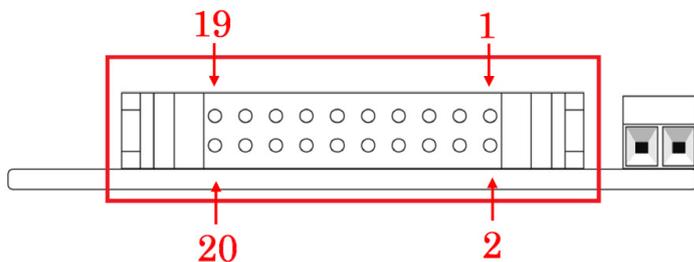
## 5. ストローブ出力端子 / トリガー入力端子 (各 1ch)



RPi-GP10T ストローブ出力/トリガー入力端子のピン配列と機能

PIN#	機能
S	ストローブ出力
T	トリガー入力

## 6. MIL コネクタ 20PIN

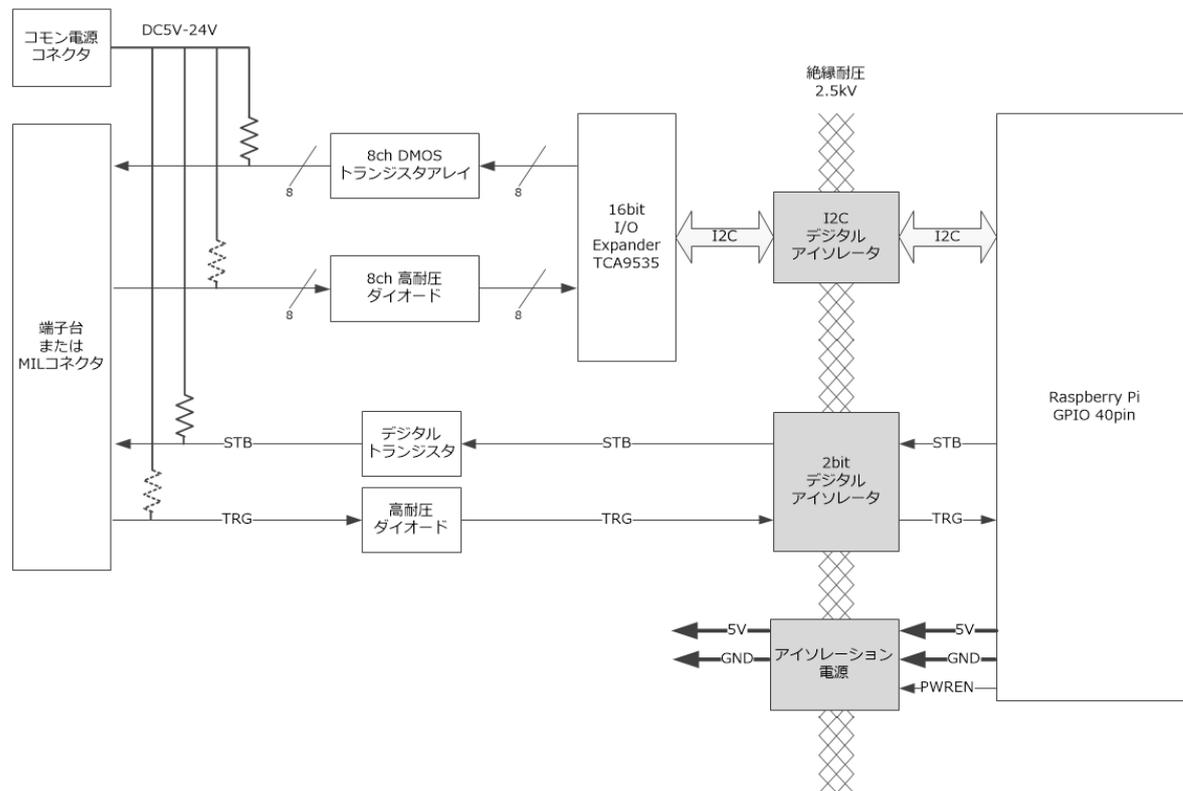


RPI-GP10M 20PIN MIL コネクタのピン配列と機能

PIN#	機能	PIN#	機能
1	5~24V 入力	11	デジタル入力 ch6
2	GND	12	デジタル入力 ch7
3	トリガー入力	13	デジタル出力 ch0
4	ストロブ出力	14	デジタル出力 ch1
5	デジタル入力 ch0	15	デジタル出力 ch2
6	デジタル入力 ch1	16	デジタル出力 ch3
7	デジタル入力 ch2	17	デジタル出力 ch4
8	デジタル入力 ch3	18	デジタル出力 ch5
9	デジタル入力 ch4	19	デジタル出力 ch6
10	デジタル入力 ch5	20	デジタル出力 ch7

## (2-3) 入出力回路

ブロック図は以下の通りとなります。



### ■ 絶縁電源制御出力について

- PWREN: 絶縁電源制御出力
- High:ON Low:OFF
- TCA9535 I/O エキスパンダなど GPIO 40pin から絶縁された回路用電源を制御します。  
(デジタル入出力・ストロブ出力・トリガー入力の制御を行う場合は、絶縁電源を ON にしてください。)

## ■ I2C について

TI 製 TCA9535 I/O エキスパンダを制御するために使用します。

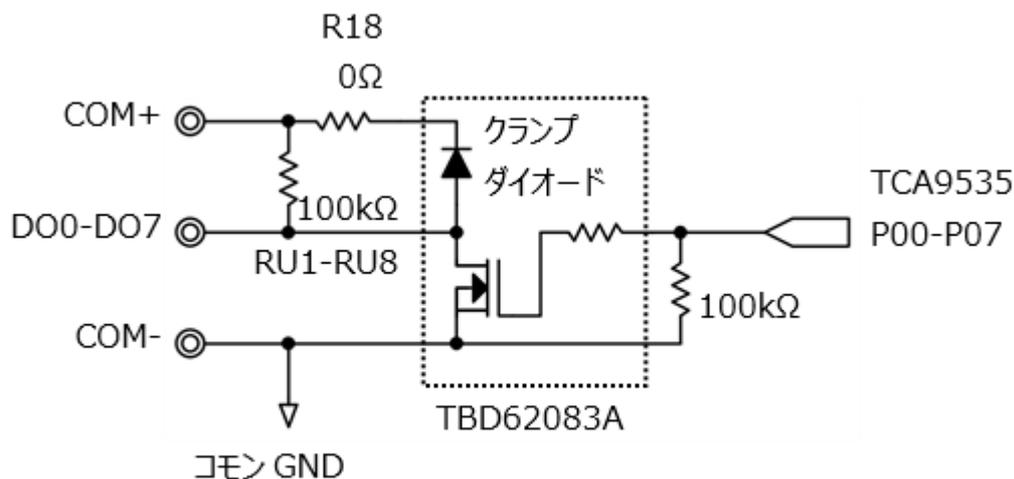
I2C アドレスは出荷時 0x20 に設定されています。  
基板裏面の抵抗 RA1-RA6 の実装を変更することで I2C アドレスを変更することができます。（「第4章 各種設定と装着」を参照）

### TCA9535 ポートレジスタ

コマンドバイト	レジスタ名	設定値	説明
0x00	ポート 0 入力データ	--	ポート 0 は出力設定なので未使用
0x01	ポート 1 入力データ	0xXX	ポート 1 入力データ bit7-0 はデジタル入力 ch7-0 に対応 極性反転設定のため '0'で'H', '1'で'L'入 力
0x02	ポート 0 出力データ	0xXX	ポート 0 出力データ bit7-0 はデジタル出力 ch7-0 に対応 出力時に反転するので '0'で'H', '1'で'L' 出力
0x03	ポート 1 出力データ	--	ポート 1 は入力設定なので未使用
0x04	ポート 0 極性反転設定	0x00	ポート 0 極性反転なし
0x05	ポート 1 極性反転設定	0xFF	ポート 1 極性反転あり
0x06	ポート 0 方向設定	0x00	ポート 0 出力に設定
0x07	ポート 1 方向設定	0xFF	ポート 1 入力に設定 ※衝突するので出力設定禁止

### ■ デジタル出力について

デジタル出力部は下図のような回路構成です。

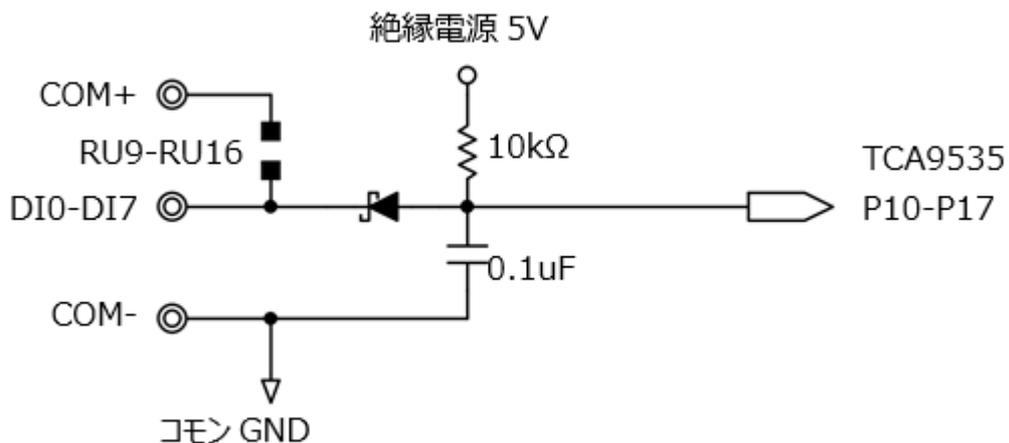


#### 【説明】

- DO0-DO7: デジタル出力チャンネル 0-7
- オープンドレイン出力方式  
(TCA9535 からのデータが反転して出力されます。)
- 最大定格は OFF 時 30V、ON 時 100mA/ch, ON 抵抗 2Ω
- DO0-DO7 は RU1-RU8(100kΩ)で COM+へプルアップ  
(プルアップが不要な場合はプルアップ抵抗 RU1-RU8 を外すか、COM+をオープンにしてください。)
- クランプダイオードは R18(0Ω)を通して COM+へ接続  
(クランプ不要な場合は R18 を外してください。)

### ■ デジタル入力について

デジタル入力部は下図のような回路構成です。

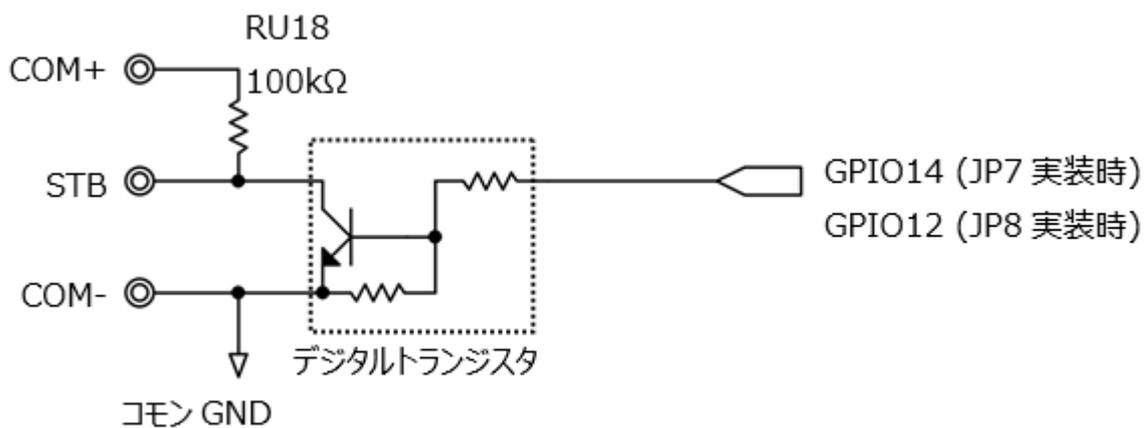


#### 【説明】

- DI0-DI7: デジタル入力チャンネル 0-7
- 高耐圧ダイオード入力方式
- $V_{IH}=3.5V$ ,  $V_{IL}=1.5V$  で耐圧 30V
- DI0-DI7 は RU9-RU16 で COM+へプルアップ可能  
(デフォルトはオープンです。  
プルアップが必要な場合は任意の 1608 サイズの抵抗を実装してください。)
- ダイオードのアノード側は絶縁電源 5V へ 10k $\Omega$  でプルアップ

### ■ ストローブ出力について

ストローブ出力部は下図のような回路構成です。

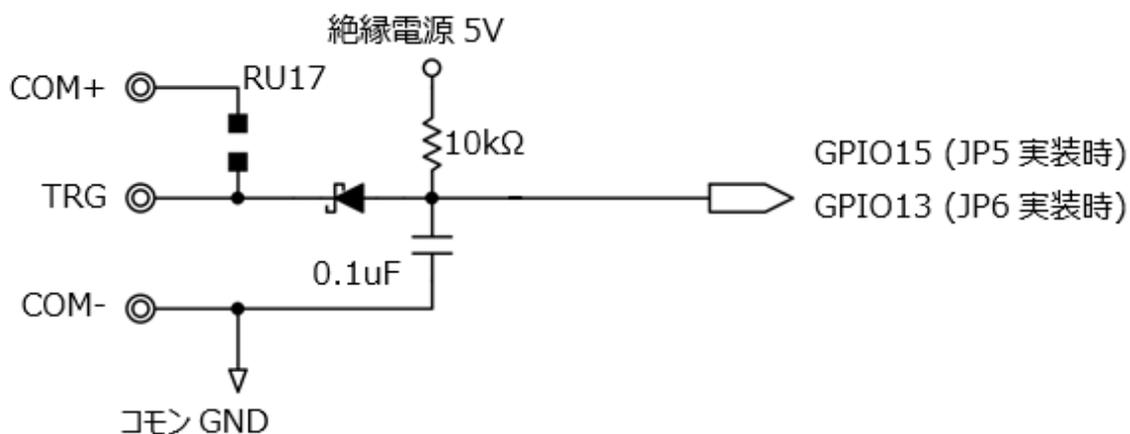


#### 【説明】

- ・ STB: ストローブ出力
- ・ オープンコレクタ出力  
(GPIO信号が反転して出力されます。)
- ・ 最大定格は OFF 時 30V、ON 時 50mA
- ・ ジャンパ抵抗 JP7/JP8 で、GPIO を 14 または 12 へ切替えることが可能  
(デフォルトは JP7 が実装されていて GPIO14 へ接続されています。)
- ・ STB は RU18(100kΩ)で COM+へプルアップ  
(不要な場合はプルアップ抵抗 RU18 を外すか、COM+をオープンにしてください。)

### ■ トリガー入力について

トリガー入力部は下図のような回路構成です。



#### 【説明】

- ・ TRG: トリガー入力
- ・ 高耐圧ダイオード入力方式
- ・  $V_{IH}=2.0V$ ,  $V_{IL}=0.8V$  で耐圧 30V
- ・ ジャンパ抵抗 JP5/JP6 で、GPIO を 15 または 13 へ切替えることが可能 (デフォルトは JP5 が実装されていて GPIO15 へ接続されています。)
- ・ TRG は RU17 で COM+へプルアップ可能 (デフォルトはオープンです。プルアップが必要な場合は任意の 1608 サイズの抵抗を実装してください。)
- ・ ダイオードのアノード側は絶縁電源 5V へ 10kΩ でプルアップ

## 第3章 Raspbianのインストールと設定

この章では、本製品を使用できるようにするための Raspbian の設定方法について説明しています。

※ Raspberry Pi は'Raspberry Pi3 ModelB'、OS は'Raspbian Stretch with Desktop (NOOBS:2018-03-14)'で説明します。

(3-1) Raspbian のインストール

(3-2) I2C の有効設定

### (3-1) Raspbian のインストール

1) Class10 の microSD(8~32G)を用意します。

64GB以上のSDカードの場合、exFATでフォーマットされます。RaspbianはexFATに対応していませんので、別のツールを使ってFAT16またはFAT32でフォーマットする必要があります。

2) SD カード用フォーマッターとユーザーマニュアルをダウンロードします。

SD アソシエーションのダウンロードページから「SD メモリカードフォーマッター」と「ユーザーマニュアル」をダウンロードします。

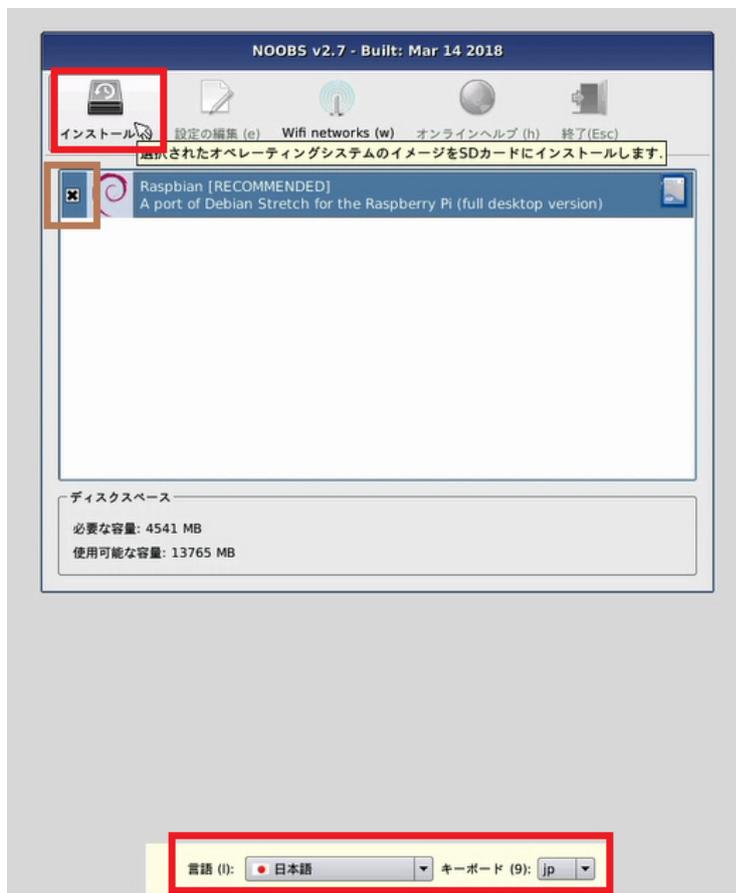
[https://www.sdcard.org/jp/downloads/formatter\\_4/index.html](https://www.sdcard.org/jp/downloads/formatter_4/index.html)

3) 「SD メモリカードフォーマッター」を使って、SD カードをフォーマットします。フォーマット方法につきましては、ダウンロードしたユーザーマニュアルをご参照ください。

4) Raspberry 財団公式ホームページ <https://www.raspberrypi.org/downloads/noobs/>

「NOOBS」(Download ZIP)を PC でダウンロードし解凍します。解凍後、フォルダー内のデータ全てを microSD にコピーします。

- 5) microSD を Raspberry Pi 基板に接続し起動します。  
[言語]に「日本語」を選択し、[Raspbian [RECOMMENDED]]にチェックを入れ「インストール」をクリックします。



- 6) 警告画面で「はい」をクリックします。

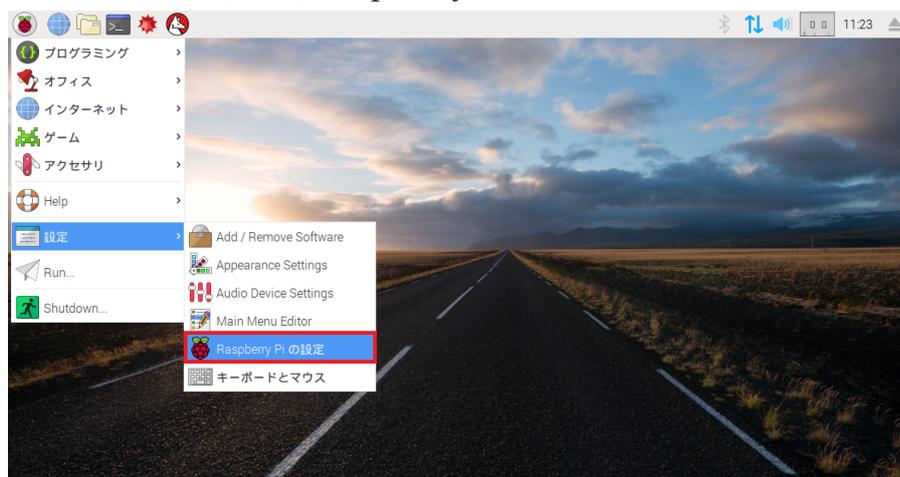


7) インストール成功の画面が表示されますので「OK」をクリックします。



### (3-2) I2C の有効設定

OS を起動し、[設定]-[Raspberry Pi の設定]をクリックします。



[インターフェース]で"I2C"を有効にし、OS を再起動します。



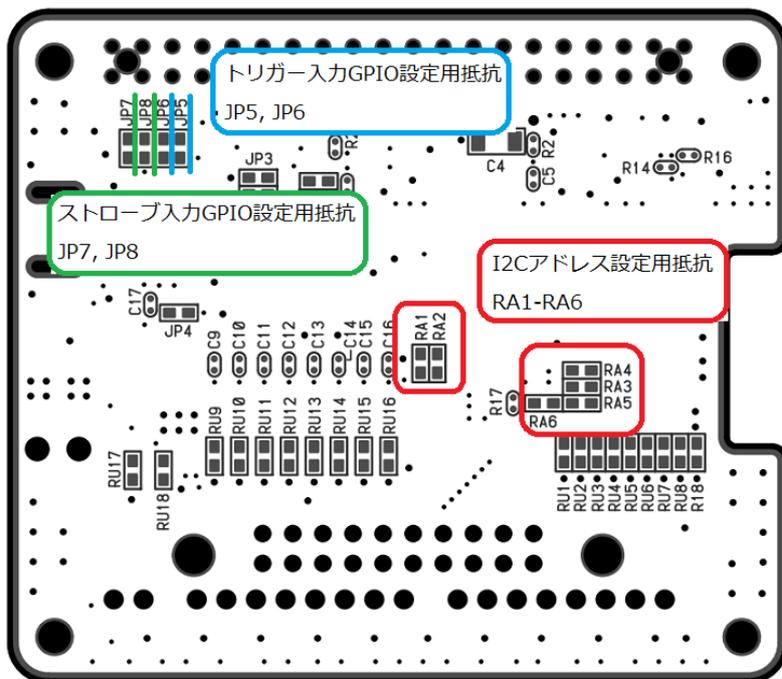
以上で Raspbian のインストールと設定は完了です。

## 第4章 各種設定と装着

各種設定と本製品を Raspberry Pi GPIO 40PIN に接続する方法を説明します。

### (4-1) 各種設定

基板裏面にある抵抗の実装を変更することにより、「I2C アドレス」「トリガー入力端子」「ストロープ出力端子」を変更することができます。



#### ■ I2C アドレスの設定

抵抗(RA1, RA2, RA3, RA4, RA5, RA6)の実装によって I2C アドレスを設定することができます。

※初期設定は 0x20(RA2, RA4, RA6 実装)です。

※抵抗は、100kohm/1608 サイズを使用してください。

RA1	RA2	RA3	RA4	RA5	RA6	I2C アドレス	
---	○	---	○	---	○	0x20	※初期設定
○	---	---	○	---	○	0x21	
---	○	○	---	---	○	0x22	
○	---	○	---	---	○	0x23	
---	○	---	○	○	---	0x24	
○	---	---	○	○	---	0x25	
---	○	○	---	○	---	0x26	
○	---	○	---	○	---	0x27	

### ■ トリガー入力用 GPIO の設定

トリガー入力用 GPIO の初期設定は **GPIO15** ですが、**JP5** を未実装にし **JP6** に 0ohm/1608 サイズの抵抗を実装することによって **GPIO13** に変更することができます。

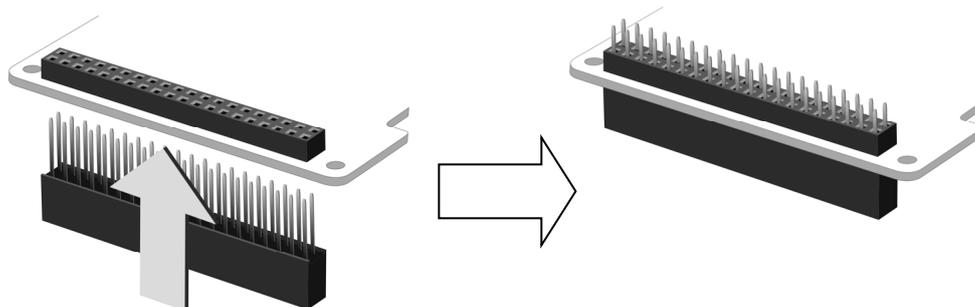
### ■ ストローブ出力制御用 GPIO の設定

ストローブ出力制御用 GPIO の初期設定は **GPIO14** ですが、**JP7** を未実装にし **JP8** に 0ohm/1608 サイズの抵抗を実装することによって **GPIO12** に変更することができます。

## (4-2) 基板の装着

製品付属の 40PIN ピンヘッダーを本製品の底面より垂直に装着します。

※40PIN のピンヘッダーの先端は尖っていますので、怪我には十分ご注意ください。



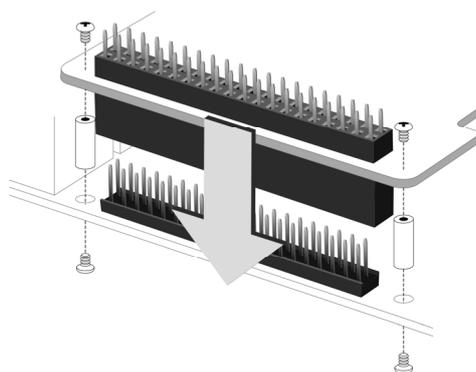
Raspberry Pi 本体の GPIO ピンヘッダーと本製品を接続します。

スペーサー (付属) を本製品本体のネジ穴にあわせ、スペーサーを付属のネジ (4 本) で固定し、そのまま RaspberryPi の GPIO ピンへ垂直に差し込みます。

スペーサーが RaspberryPi 基板に合わされば接続は完了です。

あとは RaspberryPi の背面より付属のネジ (4 本) を使用し、スペーサーを固定します。

※反対側にも同じようにスペーサーとネジを使用し本体を固定してください。



## 第5章 サンプルプログラムについて

この章ではRPI-GP10用Pythonサンプルファイル(sampleGp10.py)の使用方法について説明します。

Raspberry Pi は'Raspberry Pi3 ModelB'、OS は'Raspbian Stretch with desktop(NOOBS:2018-03-14)'で説明します。

### (5-1) 準備

#### ■ Raspberry Pi に RPi-GP10 を接続

「第4章 各種設定と装着」を参考に下記の準備をおこなってください。

- ・ 'Raspberry Pi'に'RPi-GP10'を接続
- ・ OS('Raspbian)のアップデート
- ・ GPIO40pin の I2C 設定

#### ■ Python サンプルファイルを実行するディレクトリを作成

1. 'mkdir'コマンドを使って'RPi-GP10'という名前のディレクトリを作成します。(ディレクトリ名や作成場所は自由です)

```
$ mkdir RPi-GP10
```

2. 'ls'コマンドを実行して'RPi-GP10'ディレクトリが作成されていることを確認します。

```
$ ls
```

3. 'cd'コマンドで'RPi-GP10'ディレクトリに移動します。

```
$ cd RPi-GP10
```

#### ■ Python サンプルファイルを GitHub からダウンロード

GitHub から Python サンプルファイルをダウンロードします。

1. sampleGp10.py をダウンロード

```
$ wget https://github.com/ratocsystems/rpi-gp10/raw/master/python/sampleGp10.py
```

2. ls コマンドを実行して Python サンプルファイル sampleGp10.py がダウンロードされていることを確認します。

```
$ ls  
sampleGp10.py
```

## (5-2) Python サンプルファイルについて

出力端子・入力端子・トリガー入力端子・ストロブ出力端子を制御する Python サンプルプログラムです。

サンプルプログラムでは下記の処理を行っています。

### 1. 初期値の設定

I2C と GPIO の初期値を設定します。

- ・ I2C1 を使用するためにコネクションオブジェクト取得。
- ・ RPi-GP10 を制御する I2C アドレスを設定。  
(デフォルトは 0x20 ですが、基板上の RA1-RA6 の実装を変更することで違う I2C アドレスを設定することができます。)
- ・ ストロブ出力端子用 GPIO 番号を設定。  
(デフォルトは GPIO14 ですが、基板上の JP7 を JP8 に変更することで GPIO12 に設定することができます。)
- ・ トリガー入力端子用 GPIO 番号を設定。  
(デフォルトは GPIO15 ですが、基板上の JP5 を JP6 に変更することで GPIO13 に設定することができます。)

### 2. RPi-GP10 用インターフェースの初期設定

GPIO と I2C の初期設定を行います。

※ハードウェアに依存する設定ですので変更しないでください。

- ・ GPIO を GPIO 番号で指定するように設定。
- ・ 絶縁回路用電源を ON に設定。電源 ON 後、安定するまで待ちます。
- ・ ストロブ端子を出力に設定し、初期状態を High にします。
- ・ トリガー端子を入力に設定し、初期状態を High にします。
- ・ I2C でポート 0 を出力・反転なしに設定し、出力端子として使用できるようにします。  
方向設定: Configuration レジスタ 0x06 に 0x00(全ビット出力)を書きます。  
極性 : Polarity Inversion レジスタ 0x04 に 0x00(全ビット反転なし)を書きます。
- ・ I2C でポート 1 を入力・反転ありに設定し、入力端子として使用できるようにします。  
方向設定: Configuration レジスタ 0x07 に 0xFF(全ビット入力)を書きます。  
極性 : Polarity Inversion レジスタ 0x05 に 0xFF(全ビット反転あり)を書きます。

### 3. 出力端子の制御

8ch の出力端子を制御するために、I2C で Output レジスタ 0x02 に出力 ch データを書き込みます。

書き込むデータの各ビットが出力端子の各 ch に対応していますが、反転しますので High を出力したいビットを 0 に、Low を出力したいビットを 1 に設定してください。

ch7	ch6	ch5	ch4	ch3	ch2	ch1	ch0
bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0

例 1) ch7, ch5, ch3 を High にする場合: 01010111b = 0x57

例 2) 全 ch を Low にする場合: 11111111b = 0xFF

例 3) 全 ch を High にする場合: 00000000b = 0x00

### 4. トリガー入力端子の監視

トリガー入力端子に割り当てられた GPIO の状態を監視します。

サンプルプログラムでは、GPIO.add\_event\_detect 関数と GPIO.event\_detected 関数を使ってトリガーの立ち下がり監視しています。

### 5. 入力端子の確認

8ch の入力端子の状態を確認するために、I2C で Input レジスタ 0x01 の値を読み取ります。

読み取りデータの各ビットが入力端子の各 ch に対応していますが、反転していますので各ビットが 0 なら High、1 なら Low です。

ch7	ch6	ch5	ch4	ch3	ch2	ch1	ch0
bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0

例 1) 読み取りデータが 0xA4 の場合: 0xA4 = 10100100b

ch6, ch4, ch3, ch1, ch0 が High

例 2) 読み取りデータが 0xFF の場合: 0xFF = 11111111b 全 ch が Low

例 3) 読み取りデータが 0x00 の場合: 0x00 = 00000000b 全 ch が High

### 6. ストローブ出力端子の制御

ストローブ出力端子に割り当てられた GPIO を制御することで、ストローブ信号を出力します。

制御する GPIO とストローブ出力は反転して出力します。

(GPIO が High の時はストローブ出力は Low、GPIO が Low の時はストローブ出力は High)

サンプルプログラムでは、入力端子の状態が指定データと同じだった場合 Low を出力しています。

### (5-3) Python サンプルファイルの使い方

サンプルファイルの前に、python3 をつけて実行します。

引数をつけずに Python サンプルファイル sampleGp10.py を実行すると、入力端子の状態が取得できます。

例) 入力端子の ch7, ch5, ch1 が High の場合(0x5D 01011101b)

```
python3 sampleGp10.py
入力信号: 0x5D
```

- ・ 出力端子を制御する

引数に'-o'と出力したい出力 ch データを設定することで、出力端子制御することができます。(入力端子の状態も取得します)

例) ch3, ch0 を High にしたい場合は、出力 ch データを 0xF6(11110110b)に設定

```
python3 sampleGp10.py -o 0xF6
0xF6 を出力しました
入力信号: 0xXX
```

- ・ トリガー検出時に出力端子を制御する

引数に'-t'と出力したい出力 ch データを設定すると、トリガー信号の立ち下がり検出時に出力端子制御することができます。(入力端子の状態も取得します)

トリガー信号が検出されるまで無限ループします。

例) トリガー検出時に ch7, ch5, ch3, ch1, ch0 を High にしたい場合は、出力 ch データを 0x54(01010100b)に設定

```
python3 sampleGp10.py -t 0x54
Trigger!!!
0xAB を出力しました
入力信号: 0xXX
```

- ・ 指定の入力状態であればストロブ信号を出力する

引数に'-s'と入力 ch データを設定することで、入力状態が指定の入力 ch データと同じであればストロブ信号 Low を出力します。

例) ch0 が High の場合にストロブ信号を出力したい場合は、

入力 ch データを 0xFE(11111110b)に設定

```
python3 sampleGp10.py -s 0xFE
入力信号: 0xFE
Strobe!!!
```

製品に対するお問い合わせ

RPi-GP10T/RPi-GP10M の技術的なご質問やご相談の窓口を用意していますので  
ご利用ください。

ラトックシステム株式会社

I&L サポートセンター

〒550-0015

大阪市西区南堀江 1-18-4 Osaka Metro 南堀江ビル 8F

TEL 06-7670-5064

FAX 06-7670-5066

〈サポート受付時間〉

月曜 - 金曜（祝祭日は除く）AM 10:00 - PM 1:00

PM 2:00 - PM 5:00

また、インターネットのホームページでも受け付けています。

HomePage ➡ <https://www.ratocsystems.com>



個人情報取り扱いについて

ご連絡いただいた氏名、住所、電話番号、メールアドレス、その他の個人情報は、お客様への回答など本件に関わる業務のみに利用し、他の目的では利用致しません。



©RATOC Systems, Inc. All rights reserved.