

# SPI\_DIO 基板 取扱説明書

令和 元年 7月

初版暫定

スパイス

## 重要 必ずお読みください

本製品を安全にお使いいただくために、以下に示す注意事項を必ずお守りください。

万が一、誤った使い方をされますと、お客様の開発ターゲットの破損、怪我、火災の原因となるおそれがあります。

1. 通電状態の機器に触れる際には、破損や感電、怪我などに十分ご注意ください。
2. 本製品を誤った方向に差し込むと、ハードウェアが破損することがあります。また、本製品の挿抜は必ず電源断の状態で行ってください。
3. 本製品に強い振動や衝撃、熱を与えないで下さい。
4. 万が一、異常を感じた場合は速やかに電源を OFF にし状況を確認してください。

本製品は、家電機器や工作機械、通信機器、計測機器などの一般的な産業機器に使われることを意図しております。同じ産業機器であっても特に高い信頼性を要求される用途には向いていません。

半導体部品を使用した製品は、外来ノイズやサージにより誤作動したり故障したりする可能性がありますので、ご使用になる場合は万一の誤作動や故障した場合においても生命・身体・財産等が侵害されることのないよう、装置としての安全設計（リミットスイッチやヒューズ・ブレーカ等の保護回路の設置、装置の多重化等）に万全を期されますようお願い申し上げます。

また、書面による事前の許諾なしに次に掲げるハイリスク用途に使用することはできません。

1. 人命に関わる機器（輸送機器、航空機器、医療機器などを含む）
2. 誤動作により、人体、財産または自然環境に影響を及ぼす可能性のある機器
3. 誤動作により、火災の発生を起こさせる可能性のある機器
4. 航空・宇宙機器およびナビゲーションシステム
5. 兵器システムあるいは軍事目的の機器を製造または製造の支援をするための機器
6. 海底中継機器、交通用信号機器、防災・防犯装置、各種安全装置
7. 原子力関連機器
8. 電動工具

# 目次

はじめに .....	2
1. ショートジャンパによるSPI選択.....	3
2. 接続コネクタのピン配置と信号名 .....	3
3. ドライバソフトウェアの使い方.....	5
3. 1 デバイスの宣言.....	5
3. 2 ドライバ関数の説明 .....	6
4. 仕様 .....	7
5. サポート .....	7
6. 保障について .....	7

## はじめに

このたびは SPI\_DIO ボードをお買い上げいただきまして誠にありがとうございます。

本製品は別売の PIC32MM CPU ボードおよびシステムソフトウェア P-System と一緒に使用することを想定して製作してあります。

本製品が皆様のお役にたてば幸いです。

## ご注意

1. 本書の内容および製品の仕様は、改良のため将来予告無しに変更することがありますので、ご了承ください。
2. 本書の内容については万全を期して作成しておりますが、万一お気づきの点がございましたらご連絡いただければ幸いです。

## 付属品

本製品の梱包物は以下の通りです。

- SPI\_DIO 基板本体 1 個
- M3x16mm スペーサ(オスメス型) 4 個

## 1. ショートジャンパによるSPI選択

SPI デバイスのアドレス選択はショートジャンパによって行います。

SPIアドレス	J2	J3	J4
0	OFF	OFF	OFF
1	ON	OFF	OFF
2	OFF	ON	OFF
3	ON	ON	OFF
4	OFF	OFF	ON
5	ON	OFF	ON
6	OFF	ON	ON
7	ON	ON	ON

ON : ジャンパショート

OFF : ジャンパ無し

### [重要な注意事項]

Rev.1 の基板のみシルク印刷にミスがあります。ジャンパの並びを下記のように読み替えてください。

Rev 番号は基板上の CN3 IO ポートコネクタの近くに型式と共に記載があります。

誤 : J2 J4 J3

正 : J2 J3 J4

## 2. 接続コネクタのピン配置と信号名

CN1		JTAGコネクタ	
ピン番号	信号名	ピン番号	信号名
1	TCK	2	GND
3	TDO	4	3.3V
5	TMS	6	
7		8	
9	TDI	10	GND

CN2 SPIコネクタ1

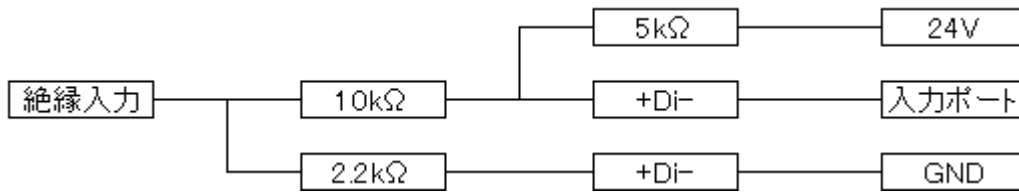
ピン番号	信号名	ピン番号	信号名
1	+5V	2	+3.3V
3	DO0(RC15)	4	GND
5	DO1(RC11)	6	nDIO(RC0)
7	GND	8	nDI(RC1)
9	nINT	10	GND
11	RP9	12	RP19
13	RP8	14	GND
15	RP7	16	RP3
17	RP6	18	GND
19	RP2	20	RP4
21	RP1	22	GND
23	RP17	24	RP10
25	RP16	26	GND
27	DEV_SEL0(RA11)	28	DEV_SEL1(RA12)
29	GND	30	DEV_SEL2(RA13)
31	*nDEV_EN(RC14)	32	GND
33	SDI	34	GND
35	SCK	36	GND
37	SDO	38	GND
39	*RESET(RC5)	40	空き

CN2 SPI コネクタ1の詳細はPIC32MM\_CPU\_取扱説明書を参照してください。

CN3 IOポートコネクタ

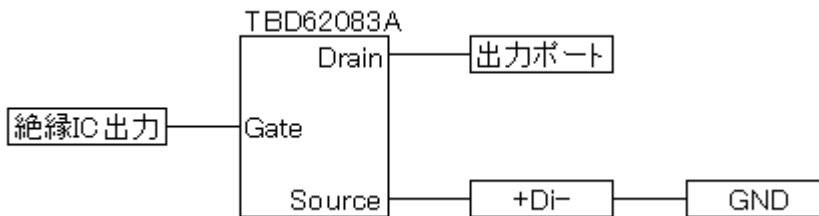
ピン番号	信号名	ピン番号	信号名
1	IN0	2	IN1
3	IN2	4	IN3
5	IN4	6	IN5
7	IN6	8	IN7
9	24V	10	GND
11	IN8	12	IN9
13	IN10	14	IN11
15	IN12	16	IN13
17	IN14	18	IN15
19	24V	20	GND
21	OUT0	22	OUT1
23	OUT2	24	OUT3
25	OUT4	26	OUT5
27	OUT6	28	OUT7
29	GND	30	GND
31	OUT8	32	OUT9
33	OUT10	34	OUT11
35	OUT12	36	OUT13
37	OUT14	38	OUT15
39	GND	40	GND

## 入力ポートの模式図



- ・ Di はダイオードで '+' 側がアノード, '-' 側がカソード
- ・ 絶縁入力から CPU の入力ポートに接続される間にはチャタリング除去用のLPFとシュミットトリガ入力ICが接続されています。
- ・ 添付のライブラリソフトウェアによる入力値は上図の入力ポートにおける値を示します。

## 出力ポートの模式図



- ・ Di はダイオードで '+' 側がアノード, '-' 側がカソード
- ・ 添付のライブラリソフトウェアによる出力値は上図の出力ポートにおける値を示します。ただし、Hi-z は '1' として扱います。

## 3. ドライバソフトウェアの使い方

本製品用のドライバソフトウェアは別売の P-System を使用することを前提として動作します。このためドライバソフトウェア自体も P-System に含まれています。もし、ご自分でドライバソフトウェアを作成するという方は別途ご相談ください。

### 3. 1 デバイスの宣言

まず、最初の作業はシステムにデバイスを宣言します。

デバイスの宣言時にはデバイスに関する定義情報を引数として渡します。

device::spi::spi\_dio フォルダにある spi\_dio.hpp を参照してください。下記の構造体を定義してあります。

```
typedef struct {  
    ::mcu::spi::CSpiMaster& master;  
    uint8_t dev_no; //DeviceSelect()で選択する no  
    uint8_t brg;  
} SpiDioBoardDef_t;
```

CSpiMaster&は本基板が接続される SPI インターフェースへの参照を記述します。PIC32MM\_CPU 基板では DevSpi になります。

dev\_no は SPI バスにおける選択アドレス(0-7), brg は SPIxBRG レジスタに設定する値を記述しますが、これらは global\_const.hpp というファイルで別途定義し、その定義値を記述します。

宣言文は以下のようになります。

```
//SPI_DIO
static const ::device::spi::spi_dio::SpiDioBoardDef_t SpiDioBoard7Def = { DevSpi,
kSpi3_DevSel_SpiDio7, kSpiBrg_24MHz
};
::device::spi::spi_dio::CSpiDio SpiDio7(SpiDioBoard7Def);
```

### 3. 2 ドライバ関数の説明

SPI\_DIO 基板のリセット直後の出力値は全て Hi-z です。通常は外部に接続される負荷によって'1'と判断されます。

- uint16\_t getLastRead();  
直前に読み出した値を返します。まだ、一度も読み出ししていない時には 0xffffを返します。
- uint16\_t getLastWrite();  
直前に出力した値を返します。まだ、一度も書き出ししていない時には 0xffff を返します。
- void read(uint16\_t& val);  
読み出し動作を行ない、その読み値を val に返します。この処理には十数～数十 us 程度の時間が掛かります。
- void write(uint16\_t val);  
指定された値で出力動作を行ないます。この処理には十数～数十 us 程度の時間 が掛かります。
- void setVerify();  
この関数はread(), write()の動作に影響を与えます。  
これらの関数では動作時にSPI 通信を使用しますが、SPI 通信は高速である反面、エラーの検出や訂正の機能がないため、通信の確実性を保障するためにread()では二度読みをwrite()ではリードアフタライト動作を行うことでこれを担保しています。setVerify()はこの確認動作を許可します。default 設定では許可になっています。  
確認処理で不一致が見られた場合はリトライを行いますが二度とも不一致なら致命的なエラーとしてfatalErro()を呼び出し処理を中断します。
- void clearVerify();  
setVerify()で説明した処理を禁止します。確認動作の分処理時間が短縮されます。  
が、SPI 通信は他の通信方式に比べると非常に高速です。  
また、これらの処理を省いても得られる処理時間はごく僅かでありませぬ。  
特段の問題がない限りは禁止にしないことを推奨します。

#### 4. 仕様

外部供給電圧	DC24±4V
入力ポート数	16CH
出力ポート数	16CH(定格電流 100mA/CH)
複数使用枚数	最大で8枚まで(他の SPI バス接続基板を含めて)
基板サイズ	10x10x1.5cm (突起部を除く)
電源電圧	DC5V、DC3.3V (PIC32MM CPU ボードより供給)
動作環境温度	0~60℃ (但し結露のないこと)

- ・ 入力ポート, 出力ポート共にCPU とは絶縁されています。
- ・ 出力ポートの仕様はトランジスタ出力です。電源投入時の出力値はHi-z ですが、通常は外部負荷を接続することで'1'と判断されます。
- ・ 標準仕様では本基板からCPU 基板に割り込みを要求することは出来ません。
- ・ 出力ポートに大電流が流れると一瞬で故障します。誤って24V電源のプラス線を接続すると瞬時に故障しますのでご注意ください。

#### 5. サポート

本製品に関する修理・サポートはメールにてご連絡ください。

メールアドレス : [info@spice-elec.com](mailto:info@spice-elec.com)

なお、本製品に関する最新の情報はHP に記載します。

<http://www.spice-elec.com>

#### 6. 保障について

商品の無償保障期間は購入後1 年間です。この期間に正常な使用状態で故障した場合は、無料で修理いたします。

ただし、商品の返送費用はお客様のご負担となります。

保証期間内でも、天災や無理な使用による故障, お客様による改造などが原因の場合は有料での修理となります。

保障の範囲はあくまで製品本体が対象であり、不具合に伴い発生した損害については保証対象外となります。

スパイス

〒781-5241

高知県香南市吉川町吉原144-6