

# SPI\_DAx2 基板 取扱説明書

令和 2 年 3 月

初版暫定

スパイス

改定履歴

初版暫定	新規作成

## 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 .....	3
1. ショートジャンパによるSPI選択 .....	4
2. 接続コネクタのピン配置と信号名 .....	4
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\_DAx2 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、ご了承願います。
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\_DAx2 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	J1	J2	J3
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

## 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_SELO(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		DA出力コネクタ	
ピン番号	信号名	ピン番号	信号名
1	CH0_VOUT	2	CH0_IOUT
3	CH0_GND	4	
5	CH1_VOUT	6	CH1_IOUT
7	CH1_GND	8	
9		10	

CH0とCH1のGNDは絶縁されています。接続先のGNDが共通であっても個別に配線接続が必要です。

### 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には以下があります

- ・ CSpiDaTiX760 : ドライバ本体
- ・ CspiMaster : CspiDio が使用する S P I インターフェースドライバ

アプリケーションソフトウェアは直接 DA コンバータに対して読み書きを要求できませんが、内部的には SPI インターフェースを介して DA コンバータにアクセスします。

ここでは CSpiDaTiX760 の説明のみを行います。

CspiMaster は PIC32MM CPU ボードに付属しています。PIC32MM CPU ボード以外のボードに接続される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。複数の基板を使用する場合はその基板枚数分の宣言が必要です。

デバイスの宣言時にはデバイスに関する二つの定義情報を引数として渡します。

Device/spi/フォルダにある spi\_master.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。

```
struct SpiSlaveCfg_t{
    CSpiMaster& master;
    uint8_t dev_no; //DeviceSelect()で選択するno
    SpiDevDef_t def;
};
```

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。

dev\_no は SPI バスにおける選択アドレス(0-7)を指定します。SpiDevDef\_t は mcu/spi/spi\_reg.hpp 内で以下のように定義されています。

```
typedef struct {
    uint8_t mode:2; //spi mode
    uint8_t brg;
} SpiDevDef_t;
```

mode は SPI の動作モードを、brg は SPIxBRG レジスタに設定する値を記述します。

二つ目は device/spi/spi\_da\_ti\_x7760.hpp で定義されている SpiDaTiX760Cfg\_t です。

```
typedef struct {
    uint8_t resolution; //12 or 16
```

```

uint8_t          range;
                //0:0-5V, 1:0-10V, 2:±5V, 3:2:±10V, 4:Previous,
                //5:4-20mA, 6:0-20mA, 7:0-24mA
} SpiDaTiX760Cfg_t;

```

DA コンバータとして使用している Ti 社のデバイスには 16bit と 12bit があり、resolution は 12 を指定します。Range は複数ある測定レンジを数値で指定します。

宣言文は例えば以下ようになります。

```

//SPI_DAx2
extern ::mcu::spi::CspiMaster DevSpi;    //外部で定義されている
#define kSpi3_DevSel_SpiDa6
#define kSpiBrg_12MHz 1
static const ::device::spi::SpiSlaveCfg_t SpiDa6Def = {
    DevSpi,
    kSpi3_DevSel_SpiDa6,
    {0, kSpiBrg_12MHz}
};
static const ::device::spi::spi_da::SpiDaTiX760Cfg_t SpiDa6cfg = {
    12,
    kSpiDaTiX760Range_uni_5    //0
};
//デバイスの宣言、デバイス名SpiDa6として宣言。
::device::spi::spi_da::CSpiDaTiX760 SpiDa6(SpiDa6Def, SpiDa6Cfg);

```

### 3. 2 ドライバ関数の説明

device/spi/spi\_da\_ti\_x7760.hpp にもコメントとして説明を入れてあります。そちらも参考にしてください。

SPI\_DAx2 基板のリセット直後の出力値は Hi-z です。

- bool init(uint8\_t err\_sel = 0);

DA コンバータを初期化します。初期化によって出力値は Hi-z から 0V または 0(4-20mA レンジのときは 4)mA になります。

引数 err\_sel は SPI 通信に失敗したときの動作を指定します。CSpiDaTiX760 の bool 型の関数は内部で SPI 通信が CRC エラーを起こしリトライも失敗した場合、err\_sel が 0 のとき関数の値として false を返します。err\_sel が 0 以外では fatalError() 関数を呼び出します。

fatalError() 関数はユーザー定義が必要な関数で、プログラムの継続動作が出来ない状況下で呼び出され、以後は呼び出し元の関数に戻るための関数です。

通信エラーによってシステムの動作が継続できない場合に、プログラムの見通しを良くする目的で用意されています。

- bool setOut(uint8\_t ch, uint16\_t val);

指定されたCH(0~1)に指定された電圧または電流値をセットします。

- ・ `uint16_t getOut(uint8_t ch);`  
指定されたCHにセットされている出力値を内部変数から返します。通信は行いません。
- ・ `uint8_t getAlart(uint8_t ch);`  
指定されたCHのALARM信号が出力されていれば'1'を返す。通信は行いません。
- ・ `bool getStatus(uint8_t ch, uint16_t& data);`  
指定されたCHのStatusレジスタの値を返す

以下の関数は主としてデバッグ時を想定したものです。通常は必要ありません。

- ・ `bool nop(uint8_t ch);` //通信信号デバッグ用
- ・ `bool readRegister(uint8_t ch, uint8_t reg_no, uint16_t& val);`
- ・ `bool writeRegister(uint8_t ch, uint8_t reg_no, uint16_t val);`  
//これらを使ってDACの設定を変更した場合、再度設定を変更するまでその状態。
- ・ `bool setRange(uint8_t ch, uint8_t range_code);`

#### 4. 仕様

出力ポート数	2CH
出力仕様	<ul style="list-style-type: none"> <li>・ 電圧出力レンジ 0-5V, 0-10V, ±5V, ±10V</li> <li>・ 電流出力レンジ(負荷抵抗 600Ω Max) 4-20mA, 0-20mA, 0-24mA(但し、負荷抵抗 500Ω Max)</li> <li>・ 出力分解能 : 12 ビット</li> </ul>
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°C (但し結露のないこと)

- ・ 出力ポートはCPU と絶縁されています。また、CHOとCH1間のGNDも絶縁されています。

#### 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