

TRAVEO™ T2G ファミリ GPIO の使用方法

本書について

適用範囲と目的

AN220193 では、TRAVEO™ T2G ファミリ MCU の GPIO ピンを効果的に使用方法について説明します。このアプリケーションノートは、GPIO の基礎、設定オプション、割込み、および低電力動作についても説明します。

対象者

このドキュメントは TRAVEO™ T2G ファミリ MCU ファミリを使用するすべての人を対象にします。

関連製品ファミリ

TRAVEO™ T2G ファミリ CYT2/CYT3/CYT4 シリーズ

目次

	本書について	1
	目次	1
1	はじめに	3
2	GPIO ピンの基本情報	4
2.1	GPIO ピンの物理構造	4
2.2	スタートアップおよび低電力動作	6
2.3	割込み	6
2.4	GPIO 出力データ設定	7
3	GPIO 設定	8
3.1	GPIO 初期化	8
3.1.1	ドライバ部の GPIO 初期化コード例	9
3.2	ポートレベルの切替え	14
3.2.1	設定	15
3.2.2	ポートレベル切替えプログラム例	19
3.3	入力の読出し	20
3.3.1	設定	20
3.3.2	ポートレベル読出しサンプルプログラム	24
3.4	割込み	25
3.4.1	設定	25
3.5	ペリフェラル機能	32
3.5.1	SCB ポート設定	32
3.5.2	TCPWM ポート設定	39
3.5.3	アナログポート設定	44
4	Appendix A. I/O ポート設定例	48
5	用語集	54
6	関連ドキュメント	55

目次

7	その他の参考資料.....	57
	改訂履歴.....	58
	免責事項.....	59

1 はじめに

1 はじめに

TRAVEO™ T2G ファミリ MCU は、従来の MCU に比べてより多くの機能を提供する柔軟な汎用 I/O (GPIO) アーキテクチャを備えています。TRAVEO™ T2G GPIO は、従来の MCU のようにファームウェアのレジスタ設定で制御するだけでなく、カスタムデジタルロジックおよびアナログブロック信号からも駆動されます。本アプリケーションノートでは TRAVEO™ T2G GPIO ピンの基本情報および様々な機能の効果的な使用方法を説明します。本アプリケーションノートで使用されている機能と用語の詳細については、[TRAVEO™ T2G アーキテクチャテクニカルリファレンスマニュアル \(TRM\)](#) の I/O System の章を参照してください。

2 GPIO ピンの基本情報

2 GPIO ピンの基本情報

TRAVEO™ T2G GPIO ピンは次の機能を提供します。

- アナログ-デジタル入出力機能
- 様々なドライブモード
- 読出し用と書込み用に分かれたポートレジスタ
- スルーレート制御
- High-speed I/O matrix (HSIOM)
- 立上りエッジ, 立下りエッジ, または両方のエッジで割込み可能
- 状態をラッチするためのホールドモード (DeepSleep モードで I/O 状態を保持するために使用)
- CMOS, TTL, および Automotive 入力バッファモードが選択可能

GPIO 機能は TRAVEO™ T2G ファミリ MCU で使用可能なペリフェラルに依存します。

HSIOM は、CPU およびペリフェラルの信号を GPIO とやり取りする高速マルチプレクサです。HSIOM を使用すると、GPIO を複数の機能と共有し、選択したペリフェラルのピン接続を多重化できます。HSIOM 接続の詳細については、[アーキテクチャ TRM](#) を参照してください。

2.1 GPIO ピンの物理構造

[図 1](#) に TRAVEO™ T2G デバイスのリソースとピン接続を示します。

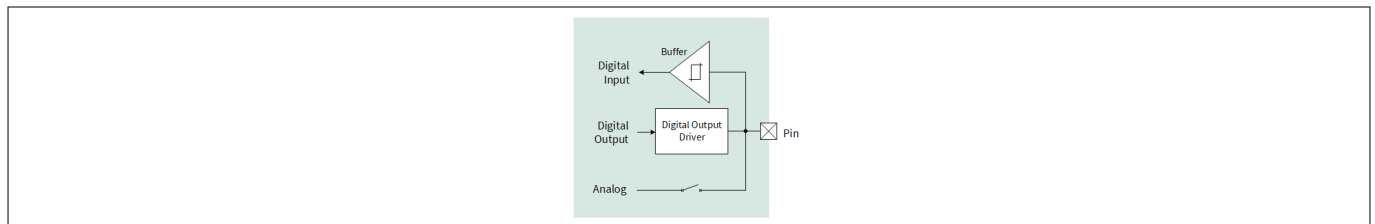


図 1 簡略化した GPIO ブロックダイアグラム

GPIO 構造の詳細なブロックダイアグラムは[アーキテクチャ TRM](#) の I/O system の章に掲載されています。各ピンは、TCPWM, SCB のデジタルペリフェラルへの入力または出力として機能します。デバイス固有の GPIO の中には、ADC で使用するためのアナログピンとして機能するものがあります。アナログピンの詳細については、[デバイスデータシート](#)を参照してください。TCPWM はタイマ, カウンタ, パルス幅変調機能を提供し、SCB はシリアル通信機能を提供します。ペリフェラル機能の詳細については、[アーキテクチャ TRM](#) を参照してください。

ピンは、デジタル入力, デジタル出力, アナログピン, またはこれら 3 つの組合せでも使用できます。例えば、デジタル出力およびデジタル入力の両方を有効にする場合、デジタル双方向ピンとなります。入力バッファは外部入力にハイインピーダンスを提供します。ピンは、CMOS, TTL, automotive レベルに設定可能です。Automotive は、CMOS よりも高い閾値レベルのトリップポイントを持ちます。CMOS と automotive のトリップレベルの違いについては、[アーキテクチャ TRM](#) を参照してください。

[図 2](#) に[図 1](#) のデジタル出力ドライバを示します。各ペリフェラルからのデジタル出力は、デジタル出力ドライバでピンを駆動します。デジタル出力ドライバは、様々なドライブモードとスルーレート制御をサポートします。

2 GPIO ピンの基本情報

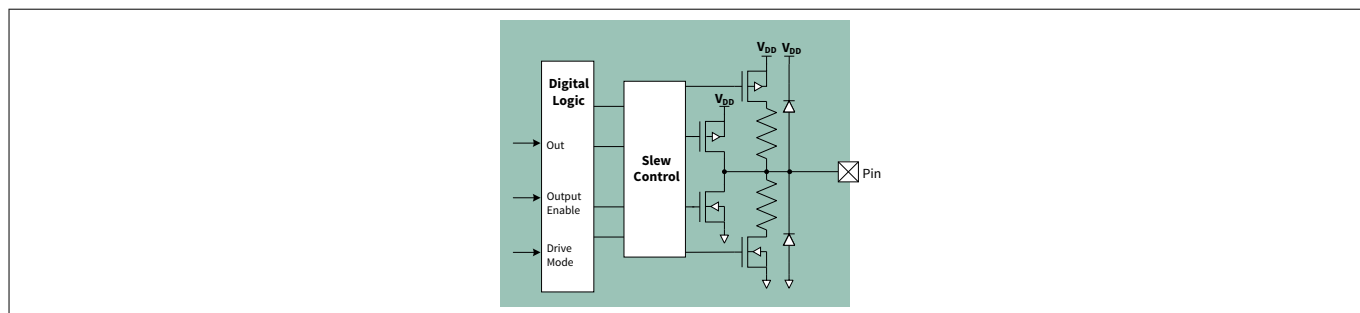


図 2 デジタル出力ドライバ

スルーレート制御は EMI およびクロストークを低減するために提供されており、ポート出力設定レジスタ (GPIO_PRTx_CFG_OUT) の SLOW ビットで設定します。高速と低速の 2 つのオプションがあります。スルーレートのデフォルト設定は高速です。速度が重要ではない場合は低速を使用してください。

TRAVERO™ T2G デバイスは様々なドライブモードをサポートします。ドライブモードはポート設定レジスタ (GPIO_PRTx_CFG) で設定します。表 1 にサポートするドライブモードと DRIVE_MODE フィールドの設定値を示します。ドライブモードに対応する出力ドライバブロックダイアグラムについては、[アーキテクチャ TRM](#) を参照してください。

表 1 ドライブモードおよびアプリケーション

DRIVE_MODE [2:0]フィールド ¹⁾	ドライブモード	アプリケーション適用例
0	ハイインピーダンス	アナログ入力とデジタル入力のインタフェース。デジタル信号の場合、入力バッファは有効にします。アナログ信号の場合、低電力設計ではクローバ電流とリーク電流を減らすために入力バッファは通常無効にします。
2	抵抗プルアップ	モータからのタコメータ出力やグラウンドに接続されているスイッチなど、オープンドレインの LOW 入力のインタフェースとして使用します。デジタル入力またはデジタル出力のどちらにも使用できます。
3	抵抗プルダウン	オープンドレインの HIGH 入力または VDD に接続されたスイッチのインタフェースとして使用します。デジタル入力またはデジタル出力のどちらにも使用できます。
4	オープンドレイン、ロードドライブ	HIGH 状態ではハイインピーダンス、LOW 状態ではストロングドライブを提供します。この設定は I2C ピンで使用します。このモードは、外部プルアップ抵抗と連動して動作します。

(続く)

2 GPIO ピンの基本情報

表 1 (続き) ドライブモードおよびアプリケーション

DRIVE_MODE [2:0] フィールド ¹⁾	ドライブモード	アプリケーション適用例
5	オープンドレイン、ハイドライブ	HIGH 状態ではストロングドライブ、LOW 状態ではハインピーダンスを提供します。このモードは外付けのプルダウン抵抗と連動して動作します。
6	ストロング	HIGH および LOW 状態の両方で CMOS 出力を提供します。
7	抵抗プルアップおよびプルダウン	HIGH および LOW 状態の両方で直列抵抗を追加します。

1) DRIVE_MODE フィールドへ 1 を設定することは禁止です。

2.2 スタートアップおよび低電力動作

リセット/電源投入時は、すべての GPIO ピンはハインピーダンス アナログ状態、すなわち、入力バッファおよび出力ドライバが無効状態で起動します。これらの GPIO ピンはリセットが解除されるまでこのモードを維持します。実行中、GPIO は関連するレジスタへの書込みで設定できます。

Sleep モードでは GPIO ピンはアクティブであり、TCPWM, SCB のようなペリフェラルで駆動可能です。このモードでは CPU だけが非アクティブです。DeepSleep モードでは DeepSleep ドメインに接続されている GPIO ピンは動作可能です。

TRAVERO™ T2G MCU には、DeepSleep および Hibernate モードで GPIO をフリーズさせる追加機能があります。低電力モードが終了すると、GPIO のフリーズは自動的に解除されます。ただし、DeepSleep ペリフェラルにより駆動される GPIO は DeepSleep モードでアクティブであり、フリーズされません。

Hibernate モードの場合、リセットを行うことでデバイスはウェイクアップします。これによって GPIO 設定およびピン状態がクリアされ、GPIO ピンがハインピーダンスのアナログ状態に初期化されます。そのため、GPIO の再設定が必要になります。

2.3 割込み

すべてのポートピンは GPIO 割込みを生成できます。図 3 に GPIO 割込み入力ブロックダイヤグラムを示します。

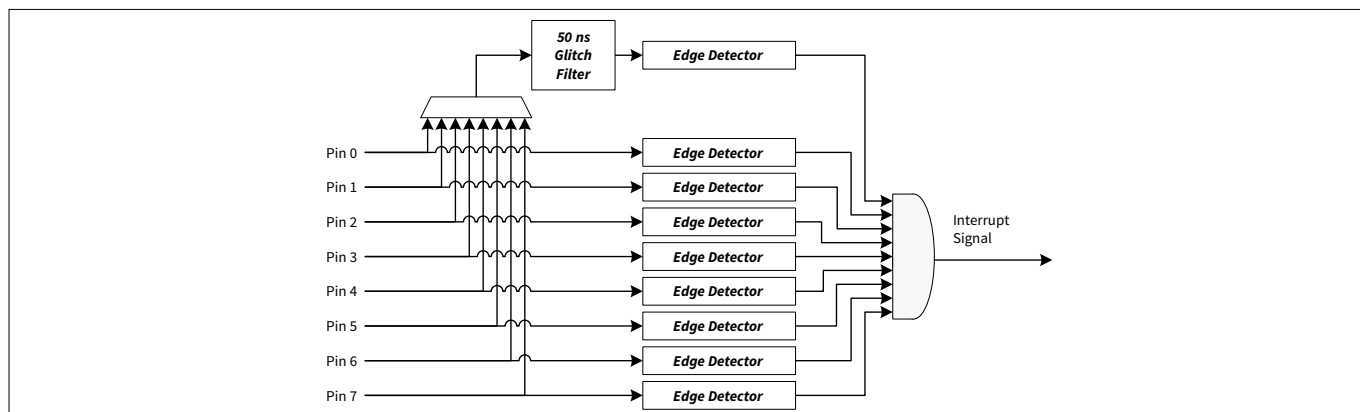


図 3 GPIO エッジ検出のブロックアーキテクチャ

ポートごとに 1 つの GPIO 割込みが生成されます。各入力ピンとグリッチフィルタ出力にはエッジ検出器があります。エッジ検出器は、入力 GPIO 信号の立上りエッジ、立下りエッジ、および両エッジのいずれかを検出できます。グリッチフィルタは、一度にポートの 1 つのピンで使用できます。エッジ検出タイプは、ポート割込み設定

2 GPIO ピンの基本情報

(GPIO_PRTx_INTR_CFG) レジスタの EDGE_x_SEL フィールドで設定されます。表 2 に、サポートされるエッジ検出タイプと EDGE_x_SEL フィールド値を示します。駆動モードに対応する出力ドライバのブロックダイアグラムについては、[アーキテクチャ TRM](#) を参照してください。

表 2 エッジ検出設定

EDGE _x _SEL フィールド	エッジタイプ
0	ディセーブル
1	立上りエッジ
2	立下りエッジ
3	両エッジ

1 つのポート内の個々の GPIO 割り込み信号は OR され、単一の割り込み要求を生成します。したがって、各ポートには 1 つの割り込みベクタがあります。割り込みをトリガしたポートを特定するために、GPIO_INTR_CAUSE_x レジスタを読み出すことができます。ソフトウェアはこのレジスタを読み出して、割り込みをアクティブにしたピンまたはグリッチフィルタ信号を特定できます。ソフトウェアは割り込み要因ルーチンフラグをクリアして、割り込みサービスルーチン (ISR) の割り込みを無効にする必要があります。

すべての I/O ピンは、Sleep および DeepSleep モードでのウェイクアップ割り込みとして使用できます。

2.4 GPIO 出力データ設定

GPIO が出力ポートとして使用されている場合、GPIO 出力データを変更するために 2 種類の設定が利用可能です。

- GPIO_PRTx_OUT

書き込みレジスタは出力データを書き込まれたデータ値に変更し、読出しは出力データ設定を反映します。レジスタの読出しは I/O ピンの現在の入力を反映しないことに注意してください。他のピンの出力データを保持するには、リードモディファイライトを使用する必要があります。

- GPIO_PRTx_OUT_CLR and GPIO_PRTx_OUT_SET

これらのレジスタは、他の I/O ピンの出力データに影響を与えることなく、対応する I/O ピンの出力データを変更できます。GPIO_PRTx_OUT_CLR レジスタに "1" を書き込むと、対応する I/O ピンは "0" にクリアされ、"0" を書き込んでもレジスタには影響しません。GPIO_PRTx_OUT_SET レジスタに "1" を書き込むと、対応する I/O ピンが "1" に設定され、"0" を書き込んでもレジスタには影響しません。

3 GPIO 設定

3 GPIO 設定

ここでは、Infineon が提供するサンプルドライバライブラリ (SDL) を使用した使用例に基づく、GPIO ピンの使用法とサンプルレジスタビット設定の実用的な例を示します。[その他の参考資料](#)を参照してください。このドキュメントに記載されている使用例は、CYT2B シリーズに基づいています。各シリーズの設定の詳細については、[関連ドキュメント](#)に記載されているデータシートを参照してください。

SDL には基本的に、設定部とドライバ部があります。設定部は、主に目的の操作のためのパラメータ値を設定します。ドライバ部は、設定部のパラメータ値に基づいて各レジスタを設定します。

ご使用のシステムに合わせて設定部を設定できます。

3.1 GPIO 初期化

[図 4](#) に、GPIO ピンを初期化するフローを示します。このフローでは、(0) が設定部で実行され、(1) から(9) がドライバ部で実行されます。サフィックス「y」は、ポートレジスタ内のピンビットフィールドを示します。詳細な設定については、使用例で説明します。

(0) システムに応じてパラメータ値を設定してください。

(1) GPIO ピンの初期出力状態を出力データ設定レジスタ (OUT_SET.OUTy) に設定してください。GPIO ピンが入力に設定されている場合、これは不要です。

(2) ピンへの HSIOM 接続を HSIOM ポート選択レジスタ (PORT_SEL0.IOy_SEL) に設定してください。HSIOM レジスタの設定により、ポートがアクティブ (ポート出力) になります。

(3) ポート入力バッファ設定レジスタ (CFG_IN.VTRIP_SELy_0) で入力バッファを選択してください。

(4) ピン出力バッファのスルーレートをポート出力バッファ設定 (CFG_OUT.SLOWy) に設定してください。

(5) 出力ドライブ強度をポート出力バッファ設定 (CFG_OUT.DRIVE_SELy) に設定してください。

(6) ドライブモードをポート設定レジスタ (CFG.DRIVE_MODE3) に設定してください。

(7) ポート割込み設定レジスタ (INTR_CFG.EDGEy_SEL) に対して、無効または有効な割込みを選択してください。

(8) ポート割込みステータスレジスタ (INTR.EDGEy) のピン割込みをクリアしてください。

(9) 割込みマスクをポート割込みマスクレジスタ (INTR_MASK.EDGEy) に設定してください。

3 GPIO 設定

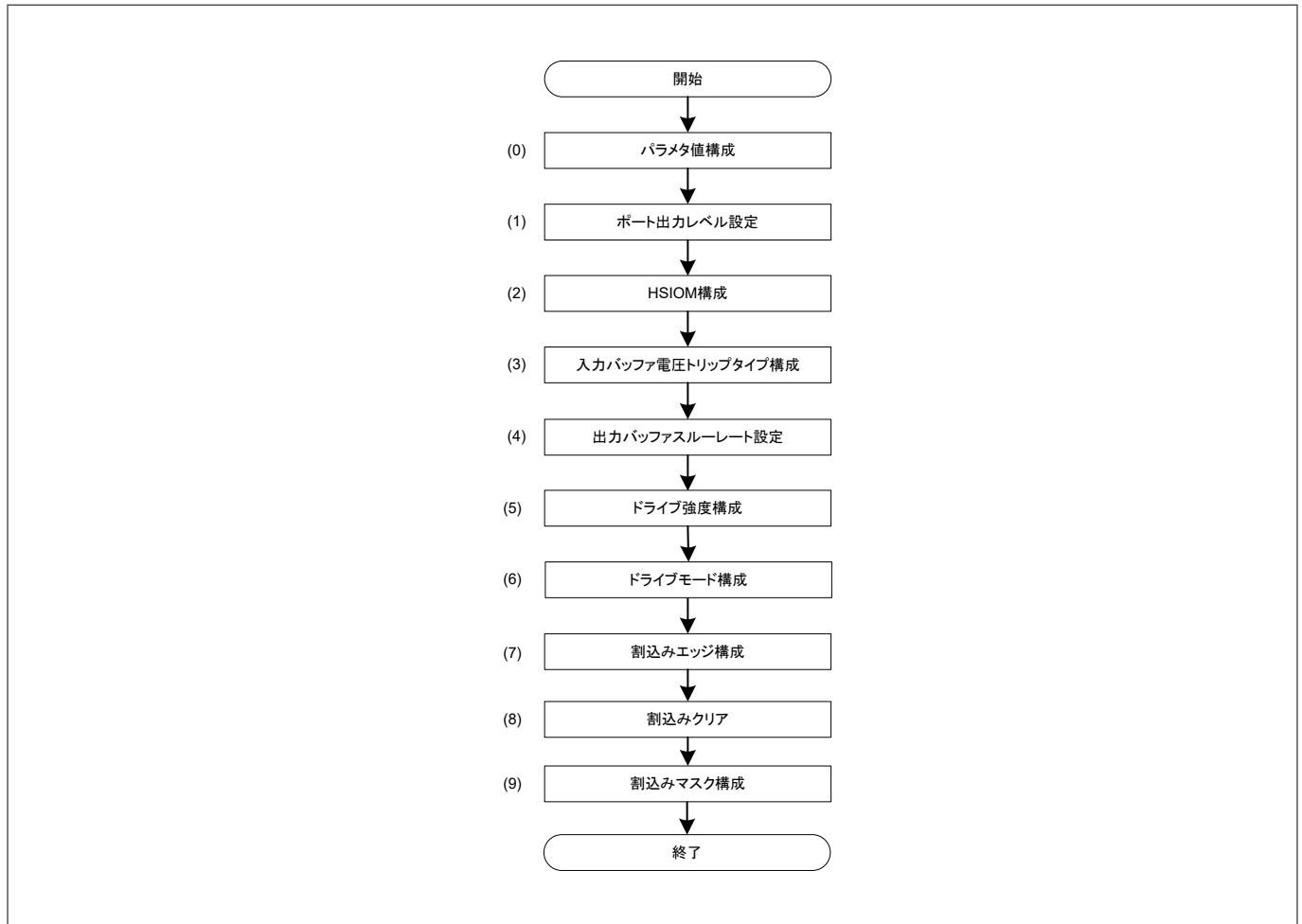


図 4 GPIO ピン初期化手順

3.1.1 ドライバ部の GPIO 初期化コード例

Code Listing 1 に、ドライバ部のポートレベルを書き込むために GPIO を初期化するサンプルプログラムを示します。

以下の説明は、SDL のドライバ部のレジスタ表記を理解するのに役立ちます。

- base は、ポートレジスタのベースアドレスへのポインタを示します。pinNum は、ポートレジスタ内のピン番号を示します。
- base->unOUT_CLR.u32Register は、[レジスタ TRM](#) で言及されている GPIO_PRTx_OUT_CLR レジスタです。「x」は GPIO ピンのポート番号を示します。
- portAddrHSIOM->unPORT_SEL0.u32Register は[レジスタ TRM](#) に記載される HSIOM_PRTx_PORT_SEL0 レジスタです。他のレジスタも同様です。
- レジスタ設定のパフォーマンス向上のため、SDL では完全な 32 ビットデータをレジスタに書き込みます。各ビットフィールドは生成後、最終的な 32 ビットデータとしてレジスタに書き込まれます。

```

tempReg = base->unCFG_IN.u32Register &
~(CY_GPIO_CFG_IN_VTRIP_SEL_MASK << pinNum);
base->unCFG_IN.u32Register = tempReg |
((value & CY_GPIO_CFG_IN_VTRIP_SEL_MASK) << pinNum);
    
```

レジスタの共用体と構造体の表現の詳細については、hdr/rev_x/ip の cyip_gpio_v2.h を参照してください。

3 GPIO 設定

Code Listing 1 ドライバ部の GPIO 初期化例

```
cy_en_gpio_status_t Cy_GPIO_Pin_Init(volatile stc_GPIO_PRT_t *base, uint32_t pinNum, const
cy_stc_gpio_pin_config_t *config)
{
    cy_en_gpio_status_t status = CY_GPIO_SUCCESS;

    if((NULL != base) && (NULL != config))
    {
        /* (1) Set port output level */
        Cy_GPIO_Write(base, pinNum, config->outVal);
        /* (2) Configure HSIOM */
        Cy_GPIO_SetHSIOM(base, pinNum, config->hsiom);
        /* (3) Configure input buffer voltage trip type */
        Cy_GPIO_SetVtrip(base, pinNum, config->vtrip);
        /* (4) Set output buffer slew rate */
        Cy_GPIO_SetSlewRate(base, pinNum, config->slewRate);
        /* (5) Configure drive strength */
        Cy_GPIO_SetDriveSel(base, pinNum, config->driveSel);
        /* (6) Configure drive Mode */
        Cy_GPIO_SetDrivemode(base, pinNum, config->driveMode);
        /* (7) Configure interrupt edge */
        Cy_GPIO_SetInterruptEdge(base, pinNum, config->intEdge);
        /* (8) Clear pin interrupt */
        Cy_GPIO_ClearInterrupt(base, pinNum);
        /* (9) Configure interrupt Mask */
        Cy_GPIO_SetInterruptMask(base, pinNum, config->intMask);
    }
    else
    {
        status = CY_GPIO_BAD_PARAM;
    }

    return(status);
}
```

次の手順は、Code Listing 1 の詳細です。

1. ポート出力レベル設定

3 GPIO 設定

Code Listing 2 ポート出力レベル設定

```
__STATIC_INLINE void Cy_GPIO_Write(volatile stc_GPIO_PRT_t* base, uint32_t pinNum,
uint32_t value)
{
    /* Thread-safe: Directly access the pin registers instead of base->OUT */
    if(CY_GPIO_ZERO == value)
    {
        /* If value is '0', output state at OUT_CLR.OUT is set to '0'. */
        base->unOUT_CLR.u32Register = CY_GPIO_OUT_MASK << pinNum;
    }
    else
    {
        /* If value is '1', output state at OUT_SET.OUT is set to '1'. */
        base->unOUT_SET.u32Register = CY_GPIO_OUT_MASK << pinNum;
    }
}
```

2. HSIOM 設定

3 GPIO 設定

Code Listing 3 HSIOM 設定

```
__STATIC_INLINE void Cy_GPIO_SetHSIOM(volatile stc_GPIO_PRT_t* base, uint32_t pinNum,
en_hsiom_sel_t value)
{
    uint8_t pinNumForHsiom;
    uint32_t portNum;
    uint32_t tempReg;
    stc_HSIOM_PRT_t* portAddrHSIOM;

    portNum = ((uint32_t)(base) - GPIO_BASE) / GPIO_PRT_SECTION_SIZE;
    portAddrHSIOM = (stc_HSIOM_PRT_t*)(HSIOM_BASE + (HSIOM_PRT_SECTION_SIZE * portNum));

    /* If pinNum is less than four, select HSIOM on port selection 0 PORT_SEL0.IO_SEL.*/
    if(pinNum < CY_GPIO_PRT_HALF)
    {
        pinNumForHsiom = pinNum;
        tempReg = portAddrHSIOM->unPORT_SEL0.u32Register & ~(CY_GPIO_HSIOM_MASK <<
(pinNumForHsiom << CY_GPIO_HSIOM_OFFSET));
        portAddrHSIOM->unPORT_SEL0.u32Register = tempReg | ((value & CY_GPIO_HSIOM_MASK)
<< (pinNumForHsiom << CY_GPIO_HSIOM_OFFSET));
    }
    else
    {
        /*If pinNum is four or greater, select HSIOM on port selection 1 PORT_SEL1.IO_SEL
        */
        pinNumForHsiom = pinNum - CY_GPIO_PRT_HALF;
        tempReg = portAddrHSIOM->unPORT_SEL1.u32Register & ~(CY_GPIO_HSIOM_MASK <<
(pinNumForHsiom << CY_GPIO_HSIOM_OFFSET));
        portAddrHSIOM->unPORT_SEL1.u32Register = tempReg | ((value & CY_GPIO_HSIOM_MASK)
<< (pinNumForHsiom << CY_GPIO_HSIOM_OFFSET));
    }
}
```

3. 入力バッファ電圧トリップ設定

Code Listing 4 入力バッファ電圧トリップ設定

```
__STATIC_INLINE void Cy_GPIO_SetVtrip(volatile stc_GPIO_PRT_t* base, uint32_t pinNum,
uint32_t value)
{
    uint32_t tempReg;

    /* Select the pin input buffer mode for the I/O pin at CFG_IN.VTRIP_SEL */
    tempReg = base->unCFG_IN.u32Register & ~(CY_GPIO_CFG_IN_VTRIP_SEL_MASK << pinNum);
    base->unCFG_IN.u32Register = tempReg | ((value & CY_GPIO_CFG_IN_VTRIP_SEL_MASK) <<
pinNum);
}
```

4. 出力バッファスルーレート設定

3 GPIO 設定

Code Listing 5 出力バッファスルーレート設定

```
__STATIC_INLINE void Cy_GPIO_SetSlewRate(volatile stc_GPIO_PRT_t* base, uint32_t pinNum,
uint32_t value)
{
    uint32_t tempReg;

    tempReg = base->unCFG_OUT.u32Register & ~(CY_GPIO_CFG_OUT_SLOW_MASK << pinNum);

    /* Select between fast or slow slew rate for the I/O pin at CFG_OUT.SLOW */
    base->unCFG_OUT.u32Register = tempReg | ((value & CY_GPIO_CFG_OUT_SLOW_MASK) <<
pinNum);
}
```

5. ドライブ強度設定

Code Listing 6 ドライブ強度設定

```
__STATIC_INLINE void Cy_GPIO_SetDriveSel(volatile stc_GPIO_PRT_t* base, uint32_t pinNum,
uint32_t value)
{
    uint32_t tempReg;
    uint32_t pinLoc;

    /* Select drive strength for the I/O pin at CFG_OUT.DRIVE_SEL */
    pinLoc = (uint32_t)(pinNum << CY_GPIO_CFG_OUT_DRIVE_OFFSET) +
CY_GPIO_CFG_OUT_DRIVE_REG_OFFSET;
    tempReg = base->unCFG_OUT.u32Register & ~(CY_GPIO_CFG_OUT_DRIVE_SEL_MASK << pinLoc);
    base->unCFG_OUT.u32Register = tempReg | ((value & CY_GPIO_CFG_OUT_DRIVE_SEL_MASK) <<
pinLoc);
}
```

6. ドライブモード設定

Code Listing 7 ドライブモード設定

```
__STATIC_INLINE void Cy_GPIO_SetDrivemode(volatile stc_GPIO_PRT_t* base, uint32_t pinNum,
uint32_t value)
{
    uint32_t tempReg;
    uint32_t pinLoc;

    pinLoc = pinNum << CY_GPIO_DRIVE_MODE_OFFSET;

    /* Select the drive mode for the I/O pin at CFG.DRIVE_MODE */
    tempReg = (base->unCFG.u32Register & ~(CY_GPIO_CFG_DM_MASK << pinLoc));

    /* Set CFG.IN_EN to '1' to enable input buffer */
    base->unCFG.u32Register = tempReg | ((value & CY_GPIO_CFG_DM_MASK) << pinLoc);
}
```

7. 割込みエッジ設定

3 GPIO 設定

Code Listing 8 割込みエッジ設定

```
__STATIC_INLINE void Cy_GPIO_SetInterruptEdge(volatile stc_GPIO_PRT_t* base, uint32_t
pinNum, uint32_t value)
{
    uint32_t tempReg;
    uint32_t pinLoc;

    /* Select interrupt edge for I/O pin at INTR_CFG.EDGE_SEL. */
    pinLoc = pinNum << CY_GPIO_INTR_CFG_OFFSET;
    tempReg = base->unINTR_CFG.u32Register & ~(CY_GPIO_INTR_EDGE_MASK << pinLoc);
    base->unINTR_CFG.u32Register = tempReg | ((value & CY_GPIO_INTR_EDGE_MASK) << pinLoc);
}
```

8. ピン割込みクリア

Code Listing 9 ピン割込みクリア

```
__STATIC_INLINE void Cy_GPIO_ClearInterrupt(volatile stc_GPIO_PRT_t* base, uint32_t
pinNum)
{
    /* Any INTR MMIO registers AHB clearing must be preceded with an AHB read access */
    (void)base->unINTR.u32Register;

    /* Clear the triggered pin interrupt at INTR.EDGE. */
    base->unINTR.u32Register = CY_GPIO_INTR_STATUS_MASK << pinNum;

    /* This read ensures that the initial write has been flushed out to the hardware */
    (void)base->unINTR.u32Register;
}
```

9. 割込みマスク設定

Code Listing 10 割込みマスク設定

```
__STATIC_INLINE void Cy_GPIO_SetInterruptMask(volatile stc_GPIO_PRT_t* base, uint32_t
pinNum, uint32_t value)
{
    uint32_t tempReg;

    tempReg = base->unINTR_MASK.u32Register & ~(CY_GPIO_INTR_EN_MASK << pinNum);

    /* Configure the pin interrupt to be forwarded to the CPU NVIC at INTR_MASK.EDGE */
    base->unINTR_MASK.u32Register = tempReg | ((value & CY_GPIO_INTR_EN_MASK) << pinNum);
}
```

3.2 ポートレベルの切替え

GPIO の簡単な使用例は、ソフトウェアでピンの出力を HIGH または LOW に設定することです。この例は、GPIO ピンを出力として構成し、CYT2B7 シリーズでポートレベルを切り替える使用例を示しています。ポートレベルを書き込むには、GPIO ピンのドライブモードをストロングに設定する必要があります。詳細については、[表 1](#) を参照してください。

3 GPIO 設定

CPU はピンを周期的に”0”または”1”に設定します。

設定例:

- ポート番号: P19.3
- 初期状態: Low
- 入出力: 出力
- ドライブモード: Strong
- 機能設定 (HSIOM): GPIO
- 割込み機能: 未使用
- スルーレート: 高速
- 出力ドライバ強度: Strong (Full drive)

3.2.1 設定

表 3 にポートレベルを書き込むための SDL の設定部のパラメータを示します。

表 3 GPIO ピンパラメータリスト

パラメータ	説明	値
.outVal	ピン出力状態選択 0: 出力状態に影響しない、1: 出力状態を”0”に設定	0ul
.driveMode	I/O ピン GPIO ドライブモード選択 0: アナログハイインピーダンス 1: 予約済み、使用できません。 2: 抵抗プルアップ 3: 抵抗プルダウン 4: オープンドレイン、ロードドライブ 5: オープンドレイン、ハイドライブ 6: 強力ドライブ 7: 抵抗プルアップ/プルダウン 8: デジタルハイインピーダンス。入力バッファオン 9: 予約済み、使用できません。 10: 抵抗プルアップ。入力バッファオン 11: 抵抗プルダウン。入力バッファオン 12: オープンドレイン、ロードドライブ。入力バッファオン 13: オープンドレイン、ハイドライブ。入力バッファオン 14: 強力ドライブ。入力バッファオン 15: 抵抗プルアップ/プルダウン。入力バッファオン	CY_GPIO_DM_STRONG_IN_OFF = 6ul SDL 説明の 1 番を参照してください。

(続く)

3 GPIO 設定

表 3 (続き) GPIO ピンパラメータリスト

パラメータ	説明	値
.hsiom	I/O ピン 0 ルート接続設定	P19_3_GPIO SDL 説明の 3 番を参照してください。
.intEdge	I/O ピン 0 割込みトリガエッジ設定 0: ディセーブル, 1: 立上りエッジ, 2: 立下りエッジ, 3: 両エッジ	0ul
.intMask	I/O ピンエッジ割込みのマスク 0: ピン割込み転送ディセーブル 1: ピン割込み転送イネーブル	0ul
.vtrip	ピン 0 入力バッファモード選択 0: CMOS, 1: TTL	0ul
.slewRate	I/O ピンスルーレート選択 0: 高速スルーレート, 1: 低速スルーレート	0ul
.driveSel	I/O ピン GPIO ドライブ強度設定 0: Full ドライブ強度 1: Full ドライブ強度 2: 1/2 ドライブ強度 3: 1/4 ドライブ強度	0ul

次の説明は、SDL でのこの例の GPIO ポート、ピン、および HSIOM の設定を理解するのに役立ちます。

3 GPIO 設定

1. ピンのドライブモードの設定に使用される定数は、common\src\drivers\gpio フォルダの GPIO ドライバヘッダ cy_gpio.h で定義されます。

```
#define CY_GPIO_DM_ANALOG                (0x00u1)
/*Analog High-Z. Input buffer off */

#define CY_GPIO_DM_PULLUP_IN_OFF          (0x02u1)
/*Resistive Pull-Up. Input buffer off */

#define CY_GPIO_DM_PULLDOWN_IN_OFF        (0x03u1)
/*Resistive Pull-Down. Input buffer off */

#define CY_GPIO_DM_OD_DRIVESLOW_IN_OFF    (0x04u1)
/*Open Drain, Drives Low. Input buffer off */

#define CY_GPIO_DM_OD_DRIVESHIGH_IN_OFF   (0x05u1)
/*Open Drain, Drives High. Input buffer off */

#define CY_GPIO_DM_STRONG_IN_OFF           (0x06u1)
/*Strong Drive. Input buffer off */

#define CY_GPIO_DM_PULLUP_DOWN_IN_OFF     (0x07u1)
/*Resistive Pull-Up/Down. Input buffer off */

#define CY_GPIO_DM_HIGHZ                  (0x08u1)
/*Digital High-Z. Input buffer on */

#define CY_GPIO_DM_PULLUP                  (0x0Au1)
/*Resistive Pull-Up. Input buffer on */

#define CY_GPIO_DM_PULLDOWN                (0x0Bu1)
/*Resistive Pull-Down. Input buffer on */

#define CY_GPIO_DM_OD_DRIVESLOW            (0x0Cu1)
/*Open Drain, Drives Low. Input buffer on */

#define CY_GPIO_DM_OD_DRIVESHIGH           (0x0Du1)
/*Open Drain, Drives High. Input buffer on */

#define CY_GPIO_DM_STRONG                  (0x0Eu1)
/*Strong Drive. Input buffer on */

#define CY_GPIO_DM_PULLUP_DOWN             (0x0Fu1)
/*Resistive Pull-Up/Down. Input buffer on */
```

2. 各ポート番号は、ヘッダフォルダのデバイスヘッダで定義されます。
例えば、MCU のリビジョンがリビジョン B の場合、デバイスヘッダは hdr/rev_b にあります。

```
#define GPIO_PRT19        ((volatile stc_GPIO_PRT_t*) &GPIO->PRT[19])
```

ポート番号とピン番号は、同じフォルダのプロジェクト固有のヘッダと一緒に定義されます。

3 GPIO 設定

例えば、MCU がリビジョン C CPU ボードの CYT2B7 シリーズの場合、bb_bsp_tviiibe1m_rev.c.h で次の定義を見つけることができます。

```
#define CY_LED0_PORT      GPIO_PRT19
#define CY_LED0_PIN      3
#define CY_LED0_PIN_MUX  P19_3_GPIO
```

3. 各 GPIO ピンには、専用の HSIOM 選択があります。特定のピンの HSIOM とそのパラメータ値は、デバイスの GPIO ヘッダにあります。

例えば、MCU が 176 ピンの CYT2B7 シリーズの場合、ヘッダファイルは gpio_cyt2b7_176_lqfp.h です。176 ピン CYT2B7 シリーズ MCU の場合、P19.3 の HSIOM 接続は次のとおりです。

```
/* P19.3 */
P19_3_GPIO = 0, /* GPIO controls 'out' */
P19_3_AMUXA = 4, /* Analog mux bus A */
P19_3_AMUXB = 5, /* Analog mux bus B */
P19_3_AMUXA_DSI = 6,
/* Analog mux bus A, DSI control */
P19_3_AMUXB_DSI = 7,
/* Analog mux bus B, DSI control */
P19_3_TCPWM0_LINE28 = 8,
/* Digital Active - tcpwm[0].line[28]:2 */
P19_3_TCPWM0_LINE_COMPL27 = 9,
/* Digital Active - tcpwm[0].line_compl[27]:2 */
P19_3_TCPWM0_TR_ONE_CNT_IN84 = 10,
/* Digital Active - tcpwm[0].tr_one_cnt_in[84]:2 */
P19_3_TCPWM0_TR_ONE_CNT_IN82 = 11,
/* Digital Active - tcpwm[0].tr_one_cnt_in[82]:2 */
P19_3_TCPWM0_TR_ONE_CNT_IN1540 = 16,
/* Digital Active - tcpwm[0].tr_one_cnt_in[1540]:0 */
P19_3_SCB2_UART_CTS = 17,
/* Digital Active - scb[2].uart_cts:1 */
P19_3_SCB2_SPI_SELECT0 = 19,
/* Digital Active - scb[2].spi_select0:1 */
P19_3_PERI_TR_IO_INPUT29 = 26,
/* Digital Active - peri.tr_io_input[29]:0 */
```

各ピンの特定の HSIOM 機能接続については、デバイスのデータシートを参照してください。

Code Listing 11 に、設定部で GPIO を初期化してポートレベルを切り替えるサンプルプログラムを示します。

3 GPIO 設定

Code Listing 11 設定部の GPIO 初期化とポートレベル切替え例

```

/* (0) Configure parameter values */
cy_stc_gpio_pin_config_t user_led_port_pin_cfg =
{
    .outVal      = 0ul,          /* Pin output state */
    .driveMode   = CY_GPIO_DM_STRONG_IN_OFF, /* Drive mode */
    .hsiom       = CY_LED0_PIN_MUX, /* HSIOM selection */
    .intEdge     = 0ul,          /* Interrupt Edge type */
    .intMask     = 0ul,          /* Interrupt enable mask */
    .vtrip       = 0ul,          /* Input buffer voltage trip type */
    .slewRate    = 0ul,          /* Output buffer slew rate */
    .driveSel    = 0ul,          /* Drive strength */
};

int main(void)
{
    SystemInit();

    __enable_irq();

    /* Place your initialization/startup code here (e.g. MyInst_Start()) */
    user_led_port_pin_cfg.hsiom = CY_LED0_PIN_MUX;

    /* CY_LED0_PORT is a Port number, CY_LED0_PIN is a Pin number and &user_led_port_pin_cfg is
    a GPIO pin structure */
    Cy_GPIO_Pin_Init(CY_LED0_PORT, CY_LED0_PIN, &user_led_port_pin_cfg);

    for(;;)
    {
        /* Wait 0.05 [s]
        Cy_SysTick_DelayInUs(50000ul);

        /* Output state is set to HIGH and LOW, successively. */
        Cy_GPIO_Inv(CY_LED0_PORT, CY_LED0_PIN);
    }
}

```

3.2.2 ポートレベル切替えプログラム例

Code Listing 12 に、ドライバ部のポートレベル切替えサンプルプログラムを示します。

Code Listing 12 ドライバ部のポートレベル切替えサンプルプログラム

```

/* Output state is set to the inverse of the current output state at OUT_INV.OUT3 */
__STATIC_INLINE void Cy_GPIO_Inv(volatile stc_GPIO_PRT_t* base, uint32_t pinNum)
{
    base->unOUT_INV.u32Register = CY_GPIO_OUT_MASK << pinNum;
}

```

3 GPIO 設定

3.3 入力の読出し

この例は、CYT2B7 シリーズの GPIO ピンからの読出しの使用例を示します。外部デジタル入力のためにデジタル入力バッファを有効にし、ポートレベルを読み出します。GPIO ピンが読出しに設定されている場合、ピンのドライブモードをデジタルハイインピーダンスに設定する必要があります。詳細については、[表 1](#) を参照してください。

設定例:

- ポート番号: P6.5
- 初期状態: Low
- 入出力: 入力
- ドライブモード: デジタルハイインピーダンス
- 機能設定 (HSIOM): 0 (GPIO)
- 割込み機能: 未使用

3.3.1 設定

[表 4](#) に入力を読み出すための SDL の設定部のパラメータを示します。

表 4 GPIO ピンパラメータリスト

パラメータ	説明	値
.outVal	ピン出力状態選択 0: 出力状態に影響しない、1: 出力状態を”0”に設定	0ul

(続く)

3 GPIO 設定

表 4 (続き) GPIO ピンパラメータリスト

パラメータ	説明	値
.driveMode	I/O ピン GPIO ドライブモード選択 0: アナログハイインピーダンス 1: 予約済み、使用できません。 2: 抵抗プルアップ 3: 抵抗プルダウン 4: オープンドレイン、ロードドライブ 5: オープンドレイン、ハイドライブ 6: 強力ドライブ 7: 抵抗プルアップ/プルダウン 8: デジタルハイインピーダンス。入力バッファオン 9: 予約済み、使用できません。 10: 抵抗プルアップ。入力バッファオン 11: 抵抗プルダウン。入力バッファオン 12: オープンドレイン、ロードドライブ。入力バッファオン 13: オープンドレイン、ハイドライブ。入力バッファオン 14: 強力ドライブ。入力バッファオン 15: 抵抗プルアップ/プルダウン。入力バッファオン	CY_GPIO_DM_HIGHZ = 8uI
.hsiom	I/O ピン 0 ルート接続設定	P6_5_GPIO SDL 説明の 2 番を参照してください。
.intEdge	I/O ピン 0 割込みトリガエッジ設定 0: ディセーブル, 1: 立上りエッジ, 2: 立下りエッジ, 3: 両エッジ	0uI
.intMask	I/O ピンエッジ割込みのマスク 0: ピン割込み転送ディセーブル 1: ピン割込み転送イネーブル	0uI
.vtrip	ピン 0 入力バッファモード選択 0: CMOS, 1: TTL	0uI
.slewRate	I/O ピンスルーレート選択 0: 高速スルーレート, 1: 低速スルーレート	0uI

(続く)

3 GPIO 設定

表 4 (続き) GPIO ピンパラメータリスト

パラメータ	説明	値
.driveSel	I/O ピン GPIO ドライブ強度設定 0: Full ドライブ強度 1: Full ドライブ強度 2: 1/2 ドライブ強度 3: 1/4 ドライブ強度	0ul

次の説明は、SDL でのこの例の GPIO ポート、ピン、および HSIOM の設定を理解するのに役立ちます。

- 各ポート番号は、header フォルダのデバイスヘッダで定義されます。
例えば、MCU シリコンのリビジョンがリビジョン B の場合、デバイスヘッダは `hdr/rev_b` にあります。

```
#define GPIO_PRT6      ((volatile stc_GPIO_PRT_t*) &GPIO->PRT[6])
```

ポート番号とピン番号は、同じフォルダのプロジェクト固有のヘッダで一緒に定義されます。

例えば、MCU がリビジョン C の CPU ボード上の CYT2B7 シリーズの場合、`bb_bsp_tviibe1m_rev_c.h` で次の定義を見つけることができます。

```
#define CY_CB_BUTTON_PORT      GPIO_PRT6
#define CY_CB_BUTTON_PIN      5
#define CY_CB_BUTTON_PIN_MUX  P6_5_GPIO
```

- 各 GPIO ピンには、専用の HSIOM 選択があります。特定のピンの HSIOM とそのパラメータ値は、デバイスの GPIO ヘッダにあります。
例えば、MCU が 176 ピンの CYT2B7 シリーズの場合、ヘッダファイルは `gpio_cyt2b7_176_lqfp.h` です。

3 GPIO 設定

176 ピン CYT2B7 シリーズ MCU の場合、P6.5 の HSIOM 接続は次のとおりです。

```
/* P6.5 */
P6_5_GPIO                = 0,      /* GPIO controls 'out' */
P6_5_AMUXA               = 4,      /* Analog mux bus A */
P6_5_AMUXB               = 5,      /* Analog mux bus B */
P6_5_AMUXA_DSI           = 6,
/* Analog mux bus A, DSI control */
P6_5_AMUXB_DSI           = 7,
/* Analog mux bus B, DSI control */
P6_5_TCPWM0_LINE2        = 8,
/* Digital Active - tcpwm[0].line[2]:0 */
P6_5_TCPWM0_LINE_COMPL258 = 9,
/* Digital Active - tcpwm[0].line_compl[258]:0 */
P6_5_TCPWM0_TR_ONE_CNT_IN6 = 10,
/* Digital Active - tcpwm[0].tr_one_cnt_in[6]:0 */
P6_5_TCPWM0_TR_ONE_CNT_IN775 = 11,
/* Digital Active - tcpwm[0].tr_one_cnt_in[775]:0 */
P6_5_SCB4_SPI_SELECT2    = 19,
/* Digital Active - scb[4].spi_select2:0 */
P6_5_LIN0_LIN_EN4        = 20,
/* Digital Active - lin[0].lin_en[4]:0 */
```

各ピンの特定の HSIOM 機能接続については、デバイスのデータシートを参照してください。

Code Listing 13 に、入力を読み出すための設定部のサンプルプログラムを示します。

3 GPIO 設定

Code Listing 13 設定部の入力読出し例

```
#define USER_BUTTON_PORT      CY_CB_BUTTON_PORT
#define USER_BUTTON_PIN      CY_CB_BUTTON_PIN
#define USER_BUTTON_PIN_MUX  CY_CB_BUTTON_PIN_MUX

cy_stc_gpio_pin_config_t user_button_port_pin_cfg =
{
    /* (0) Configure parameter values */
    .outVal      = 0ul,          /* Pin output state */
    .driveMode   = CY_GPIO_DM_HIGHZ, /* Drive mode */
    .hsiom       = USER_BUTTON_PIN_MUX, /* HSIOM selection */
    .intEdge     = 0ul,          /* Interrupt Edge type */
    .intMask     = 0ul,          /* Interrupt enable mask */
    .vtrip       = 0ul,          /* Input buffer voltage trip type */
    .slewRate    = 0ul,          /* Output buffer slew rate */
    .driveSel    = 0ul,          /* Drive strength */
};

int main(void)
{
    SystemInit();

    __enable_irq(); /* Enable global interrupts. */

    /* Place your initialization/startup code here (e.g. MyInst_Start()) */
    /* Port number, Pin number, GPIO pin structure */
    Cy_GPIO_Pin_Init(USER_BUTTON_PORT, USER_BUTTON_PIN, &user_button_port_pin_cfg);

    // Detect falling edge of the button GPIO
    uint32_t curLevel      = 0ul;

    for(;;)
    {
        // Get the current button level
        /* The current logic level on the input buffer of P6.5 is read */
        curLevel = Cy_GPIO_Read(USER_BUTTON_PORT, USER_BUTTON_PIN);
        :
    }
}
```

3.3.2 ポートレベル読出しサンプルプログラム

Code Listing 14 に、入力読出しのドライバ部のサンプルプログラムを示します。

3 GPIO 設定

Code Listing 14 ドライバ部の入力読出し例

```
__STATIC_INLINE uint32_t Cy_GPIO_Read(volatile stc_GPIO_PRT_t* base, uint32_t pinNum)
{
    /*Reads the current pin status on the input buffer of the pin from IN.IN5 */
    return (base->unIN.u32Register >> (pinNum)) & CY_GPIO_IN_MASK;
}
```

3.4 割込み

この例は、CYT2B7 シリーズのピンからの割込み生成の使用例を示します。このピンは 1 つの IRQ を使用します。そのため、割り込みソースは ISR で識別される必要があります。入力ポートでは、割込みエッジと割込みマスクを有効にします。

P6.5 が立下りエッジを検出すると、割込みが発生します。詳細については、[関連ドキュメント](#)に記載されている AN219842 の初期設定手順のセクションを参照してください。

設定例:

- ポート番号: P6.5
- 初期状態: Low
- 入出力: 入力
- ドライブモード: デジタルハイインピーダンス
- 機能設定 (HSIOM): GPIO
- 割込みエッジタイプ: 立下りエッジ
- 割込み機能: 使用
- 割込み番号: 3
- 割込み優先度: 0

3.4.1 設定

[表 5](#) に、ピン P6.5 での割込み用の SDL の設定部のパラメータを示します。

表 5 **ピン P6.5 割込みパラメータリスト**

パラメータ	説明	値
.outVal	ピン出力状態選択 0: 出力状態に影響しない、1: 出力状態を”0”に設定	0ul

(続く)

3 GPIO 設定

表 5 (続き) ピン P6.5 割込みパラメータリスト

パラメータ	説明	値
.driveMode	I/O ピン GPIO ドライブモード選択 0: アナログハイインピーダンス 1: 予約済み、使用できません。 2: 抵抗プルアップ 3: 抵抗プルダウン 4: オープンドレイン、ロードドライブ 5: オープンドレイン、ハイドライブ 6: 強力ドライブ 7: 抵抗プルアップ/プルダウン 8: デジタルハイインピーダンス。入力バッファオン 9: 予約済み、使用できません。 10: 抵抗プルアップ。入力バッファオン 11: 抵抗プルダウン。入力バッファオン 12: オープンドレイン、ロードドライブ。入力バッファオン 13: オープンドレイン、ハイドライブ。入力バッファオン 14: 強力ドライブ。入力バッファオン 15: 抵抗プルアップ/プルダウン。入力バッファオン	CY_GPIO_DM_HIGHZ = 8uI
.hsiom	I/O ピン 0 ルート接続設定	P6_5_GPIO SDL 説明の 2 番を参照してください。
.intEdge	I/O ピン 0 割込みトリガエッジ設定 0: ディセーブル, 1: 立上りエッジ, 2: 立下りエッジ, 3: 両エッジ	CY_GPIO_INTR_FALLING = 2uI
.intMask	I/O ピンエッジ割込みのマスク 0: ピン割込み転送ディセーブル 1: ピン割込み転送イネーブル	1uI
.vtrip	ピン 0 入力バッファモード選択 0: CMOS, 1: TTL	0uI
.slewRate	I/O ピンスルーレート選択 0: 高速スルーレート, 1: 低速スルーレート	0uI

(続く)

3 GPIO 設定

表 5 (続き) ピン P6.5 割り込みパラメータリスト

パラメータ	説明	値
.driveSel	I/O ピン GPIO ドライブ強度設定 0: Full ドライブ強度 1: Full ドライブ強度 2: 1/2 ドライブ強度 3: 1/4 ドライブ強度	0ul

図 5 に、ピン割り込み設定例のフローを示します。割り込みを使用して GPIO 入力設定でポートを初期化してください。

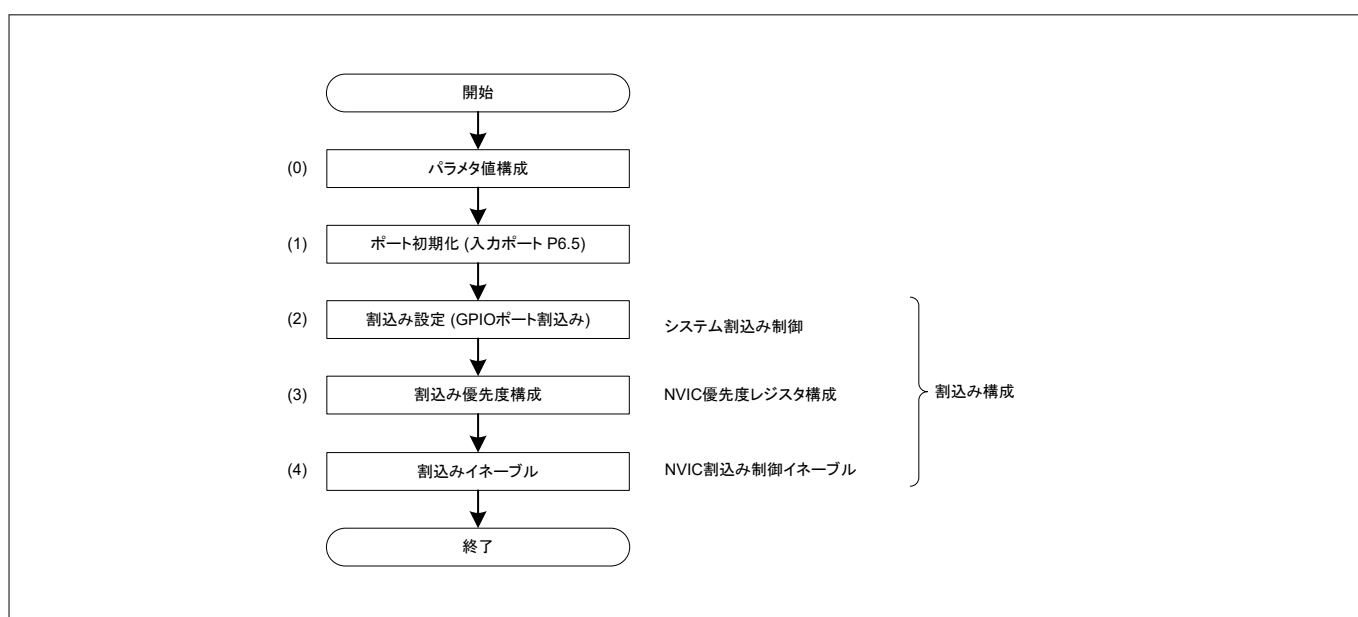


図 5 ピン割り込み生成手順

次の説明は、SDL でのこの例の GPIO ポート、ピン、および HSIOM の設定を理解するのに役立ちます。

- 各ポート番号は、header フォルダのデバイスヘッダで定義されます。
例えば、MCU シリコンのリビジョンがリビジョン B の場合、デバイスヘッダは `hdr/rev_b` にあります。

```
#define GPIO_PRT6 ((volatile stc_GPIO_PRT_t*) &GPIO->PRT[6])
```

ポート番号、ピン番号、および割り込み番号は、同じフォルダ内のプロジェクト固有のヘッダと一緒に定義されます。

例えば、MCU がリビジョン C の CPU ボード上の CYT2B7 シリーズの場合、`bb_bsp_tviibe1m_rev.c.h` で次の定義を見つけることができます。

```
#define CY_CB_BUTTON_PORT    GPIO_PRT6
#define CY_CB_BUTTON_PIN    5
#define CY_CB_BUTTON_PIN_MUX P6_5_GPIO
#define CY_CB_BUTTON_IRQN    ioss_interrupts_gpio_6_IRQn
```

- 特定のピンの HSIOM とそのパラメータ値は、デバイスの GPIO ヘッダにあります。

3 GPIO 設定

例えば、MCU が 176 ピンの CYT2B7 シリーズの場合、ヘッダファイルは `gpio_cyt2b7_176_lqfp.h` です。
176 ピン CYT2B7 シリーズ MCU の場合、P6.5 の HSIOM 接続は次のとおりです。

```
/* P6.5 */
P6_5_GPIO                = 0,    /* GPIO controls 'out' */
P6_5_AMUXA               = 4,    /* Analog mux bus A */
P6_5_AMUXB               = 5,    /* Analog mux bus B */
P6_5_AMUXA_DSI           = 6,
/* Analog mux bus A, DSI control */

P6_5_AMUXB_DSI           = 7,
/* Analog mux bus B, DSI control */

P6_5_TCPWM0_LINE2        = 8,
/* Digital Active - tcpwm[0].line[2]:0 */

P6_5_TCPWM0_LINE_COMPL258 = 9,
/* Digital Active - tcpwm[0].line_compl[258]:0 */

P6_5_TCPWM0_TR_ONE_CNT_IN6 = 10,
/* Digital Active - tcpwm[0].tr_one_cnt_in[6]:0 */

P6_5_TCPWM0_TR_ONE_CNT_IN775 = 11,
/* Digital Active - tcpwm[0].tr_one_cnt_in[775]:0 */

P6_5_SCB4_SPI_SELECT2    = 19,
/* Digital Active - scb[4].spi_select2:0 */

P6_5_LIN0_LIN_EN4        = 20,
/* Digital Active - lin[0].lin_en[4]:0 */
```

3 GPIO 設定

3. ピンの割り込みトリガタイプの設定に使用される定数は、common\src\drivers\gpio フォルダの GPIO ドライバヘッダ cy_gpio.h で定義されます。

```
#define CY_GPIO_INTR_DISABLE      (0x00u1)
/**< \brief Disable the pin interrupt generation */

#define CY_GPIO_INTR_RISING      (0x01u1)
/**< \brief Rising-Edge interrupt */

#define CY_GPIO_INTR_FALLING      (0x02u1)
/**< \brief Falling-Edge interrupt */

#define CY_GPIO_INTR_BOTH        (0x03u1)
/**< \brief Both-Edge interrupt */
```

4. 特定のピンの HSIOM とそのパラメータ値はデバイスヘッダにあります。

```
typedef enum {
    :
    ioss_interrupts_gpio_6_IRQn      = 27,
    /*!< 27 [DeepSleep] GPIO Port Interrupt #6 */
    :
} cy_en_intr_t;
```

特定の HSIOM 機能接続と割り込み割当てについては、デバイスのデータシートを参照してください。

[Code Listing 15](#) に、設定部の GPIO 割り込みを生成するサンプルプログラムを示します。

3 GPIO 設定

Code Listing 15 設定部の GPIO 割込み例

```
#define USER_BUTTON_PORT      CY_CB_USER_BUTTON_PORT
#define USER_BUTTON_PIN      CY_CB_USER_BUTTON_PIN
#define USER_BUTTON_PIN_MUX   CY_CB_USER_BUTTON_PIN_MUX
#define USER_BUTTON_IRQ      CY_CB_USER_BUTTON_IRQN

cy_stc_gpio_pin_config_t user_button_port_pin_cfg =
{
    /* (0) Configure parameter values */
    .outVal      = 0ul,          /* Pin output state */
    .driveMode   = CY_GPIO_DM_HIGHZ, /* Drive mode */
    .hsiom       = USER_BUTTON_PIN_MUX, /* HSIOM selection */
    .intEdge     = CY_GPIO_INTR_FALLING, /* Interrupt Edge type */

    .intMask     = 1ul,          /* Interrupt enable mask */
    .vtrip       = 0ul,          /* Input buffer voltage trip type */
    .slewRate    = 0ul,          /* Output buffer slew rate */
    .driveSel    = 0ul,          /* Drive strength */
};

/* Setup GPIO for BUTTON1 interrupt */
const cy_stc_sysint_irq_t irq_cfg =
{
    .sysIntSrc   = USER_BUTTON_IRQ,
    .intIdx      = CPUIntIdx3_IRQn,
    .isEnabled   = true,
};

int main(void)
{
    SystemInit();

    __enable_irq(); /* Enable global interrupts. */
    :
    /* (1) Initialize port (Input port P6.5) */
    Cy_GPIO_Pin_Init(USER_BUTTON_PORT, USER_BUTTON_PIN, &user_button_port_pin_cfg);

    /* (2) Setup interrupt (GPIO port interrupt) */
    Cy_SysInt_InitIRQ(&irq_cfg);
    Cy_SysInt_SetSystemIrqVector(irq_cfg.sysIntSrc, ButtonIntHandler);

    /* (3) Configure interrupt priority */
    NVIC_SetPriority(irq_cfg.intIdx, 0);

    /*(4) Enable interrupt */
    NVIC_EnableIRQ(irq_cfg.intIdx);

    for(;;);
}
```

3 GPIO 設定

図 6 に割り込みハンドラを示します。入力ポートが割り込みエッジを検出すると、割り込みハンドラがアクティブになります。まず、割り込みステータスを読み出し、どのポートピンが割り込みを受けているかを特定してください。次に、割り込みステータスをクリアして、次の割り込みを検出してください。Code Listing 16 に、割り込みハンドラのサンプルコードを示します。

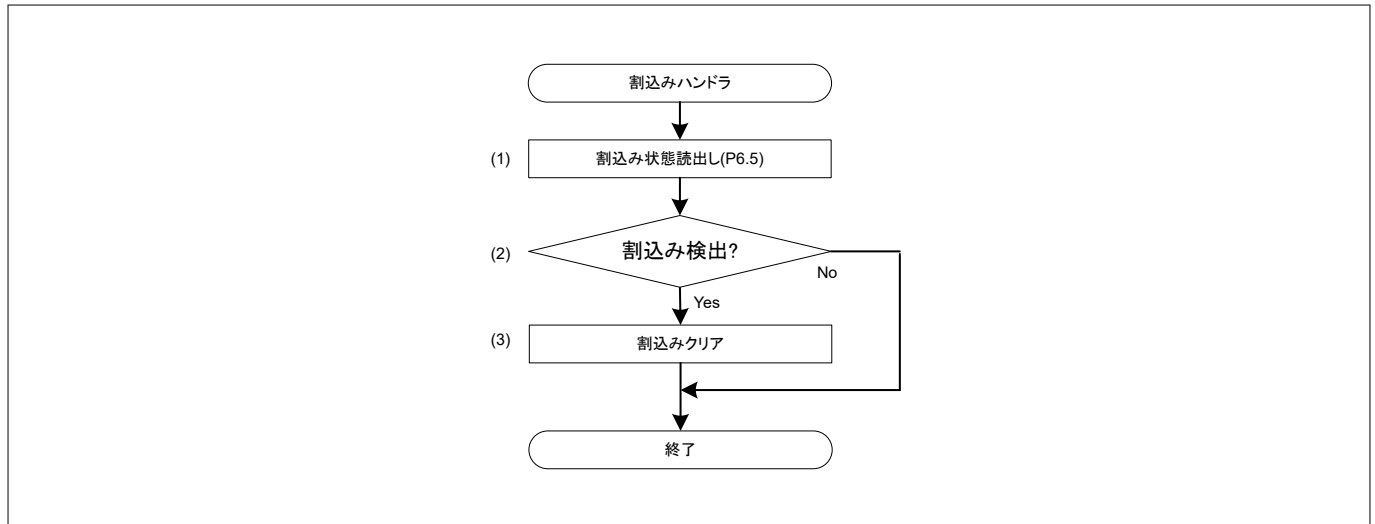


図 6 割り込みハンドラフロー例

Code Listing 16 割り込みハンドラ例

```

void IntHandler(void)
{
    uint32_t intStatus;

    /* If falling edge detected */
    /* (1) Read interrupt status (P6.5) */
    intStatus = Cy_GPIO_GetInterruptStatusMasked(USER_BUTTON_PORT, USER_BUTTON_PIN);

    /*(2) Interrupt detected? */
    if (intStatus != 0ul)
    {
        /* (3) Clear interrupt */
        Cy_GPIO_ClearInterrupt(USER_BUTTON_PORT, USER_BUTTON_PIN);
    }
}
    
```

Code Listing 17 に、(1) 割り込みステータス読出し (P6.5) ((1) Read interrupt status (P6.5))のドライバ部のサンプルプログラムを示します。

3 GPIO 設定

Code Listing 17 割込みステータス読出しドライバ部のサンプルプログラム

```
__STATIC_INLINE uint32_t Cy_GPIO_GetInterruptStatusMasked(volatile stc_GPIO_PRT_t* base,
uint32_t pinNum)
{
    /* Return the state of pin interrupt mask to be forwarded to CPU interrupt controller at
    INTR_MASKED.EDGE5 */
    return (base->unINTR_MASKED.u32Register >> pinNum) & CY_GPIO_INTR_MASKED_MASK;
}
```

3.5 ペリフェラル機能

I/O ピンへのペリフェラル機能の割当てについて説明します。ペリフェラル機能は HSIOM によって選択され、デフォルト設定は GPIO です。ピンにはいくつかの特定の機能があり、選択可能です。

各ピンで利用可能な接続については[デバイスデータシート](#)を参照してください。

アナログポート、SCB ポート TCPWM ポートの設定例について説明します。

3.5.1 SCB ポート設定

この例は、UART として設定されたシリアル通信ブロック (SCB) に関連付けられた CYT2B7 シリーズのポートの設定を示します。この例では、SCB3 (UART) ポート (P13.0/P13.1) が使用され、HSIOM ピン接続の ACT_5 に割り当てられます。Rx/Tx ポートを設定した後、UART 機能を設定してください。

各機能に適切なポートを選択する方法の詳細については、[データシート](#)の「Alternate function pin assignments」セクションを参照してください。UART 設定の詳細については、[関連ドキュメント](#)に記載されている AN220119 を参照してください。

[図 7](#) に P13.0 および P13.1 の SCB ポートとしての信号パスを示します。SCB ch3 を使用するには、HSIOM を SCB3_RX/TX に設定してください。

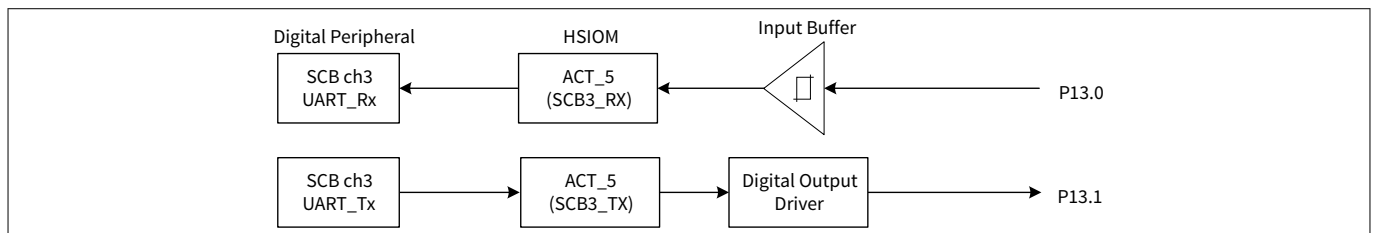


図 7 SCB ポート設定

設定例:

RX ポート

- ポート番号: P13.0
- 初期状態: Low
- 入出力: 入力
- ドライブモード: デジタルハイインピーダンス
- 機能設定 (HSIOM): SCB (UART Rx)
- 割込み機能: 未使用

TX ポート

- ポート番号: P13.1
- 初期状態: Low

3 GPIO 設定

- 入出力: 出力
- ドライブモード: Strong
- 機能設定 (HSIOM): SCB (UART Tx)
- 割込み機能: 未使用

表 6 と表 7 に、SCB RX ポートと TX ポートの SDL の設定部のパラメータをそれぞれ示します。

表 6 SCB RX ポート GPIO ピンパラメータリスト

パラメータ	説明	値
outVal	ピン出力状態選択 0: 出力状態に影響しない、 1: 出力状態を”0”に設定	0ul
driveMode	I/O ピン GPIO ドライブモード選択 0: アナログハイインピーダンス 1: 予約済み、使用できません。 2: 抵抗プルアップ 3: 抵抗プルダウン 4: オープンドレイン、ロードライブ 5: オープンドレイン、ハイドライブ 6: 強カドライブ 7: 抵抗プルアップ/プルダウン 8: デジタルハイインピーダンス。入力バッファオン 9: 予約済み、使用できません。 10: 抵抗プルアップ。入力バッファオン 11: 抵抗プルダウン。入力バッファオン 12: オープンドレイン、ロードライブ。入力バッファオン 13: オープンドレイン、ハイドライブ。入力バッファオン 14: 強カドライブ。入力バッファオン 15: 抵抗プルアップ/プルダウン。入力バッファオン	CY_GPIO_DM_HIGHZ = 8ul
.hsiom	I/O ピン 0 ルート接続設定	P13_0_SCB3_UART_RX SDL 説明の 2 番を参照してください。

(続く)

3 GPIO 設定

表 6 (続き) SCB RX ポート GPIO ピンパラメータリスト

パラメータ	説明	値
.intEdge	I/O ピン 0 割込みトリガエッジ設定 0: ディセーブル, 1: 立上リエッジ, 2: 立下リエッジ, 3: 両エッジ	0ul
.intMask	I/O ピンエッジ割込みのマスク 0: ピン割込み転送ディセーブル 1: ピン割込み転送イネーブル	0ul
.vtrip	ピン 0 入力バッファモード選択 0: CMOS, 1: TTL	0ul
.slewRate	I/O ピンスルーレート選択 0: 高速スルーレート, 1: 低速スルーレート	0ul
.driveSel	I/O ピン GPIO ドライブ強度設定 0: Full ドライブ強度 1: Full ドライブ強度 2: 1/2 ドライブ強度 3: 1/4 ドライブ強度	0ul

表 7 SCB TX ポート GPIO ピンパラメータリスト

パラメータ	説明	値
.outVal	ピン出力状態選択 0: 出力状態に影響しない、 1: 出力状態を”0”に設定	0ul

(続く)

3 GPIO 設定

表 7 (続き) SCB TX ポート GPIO ピンパラメータリスト

パラメータ	説明	値
.driveMode	I/O ピン GPIO ドライブモード選択 0: アナログハイインピーダンス 1: 予約済み、使用できません。 2: 抵抗プルアップ 3: 抵抗プルダウン 4: オープンドレイン、ロードライブ 5: オープンドレイン、ハイドライブ 6: 強カドライブ 7: 抵抗プルアップ/プルダウン 8: デジタルハイインピーダンス。入力バッファオン 9: 予約済み、使用できません。 10: 抵抗プルアップ。入力バッファオン 11: 抵抗プルダウン。入力バッファオン 12: オープンドレイン、ロードライブ。入力バッファオン 13: オープンドレイン、ハイドライブ。入力バッファオン 14: 強カドライブ。入力バッファオン 15: 抵抗プルアップ/プルダウン。入力バッファオン	CY_GPIO_DM_STRONG_IN_OFF = 6ul
.hsiom	I/O ピン 0 ルート接続設定	P13_1_SCB3_UART_TX SDL 説明の 2 番を参照してください。
.intEdge	I/O ピン 0 割込みトリガエッジ設定 0: デイセーブル, 1: 立上リエッジ, 2: 立下リエッジ, 3: 両エッジ	0ul

(続く)

3 GPIO 設定

表 7 (続き) SCB TX ポート GPIO ピンパラメータリスト

パラメータ	説明	値
.intMask	I/O ピンエッジ割込みのマスク 0: ピン割込み転送ディセーブル 1: ピン割込み転送イネーブル	0ul
.vtrip	ピン 0 入力バッファモード選択 0: CMOS, 1: TTL	0ul
.slewRate	I/O ピンスルーレート選択 0: 高速スルーレート, 1: 低速スルーレート	0ul
.driveSel	I/O ピン GPIO ドライブ強度設定 0: Full ドライブ強度 1: Full ドライブ強度 2: 1/2 ドライブ強度 3: 1/4 ドライブ強度	0ul

次の説明は、SDL でこの例の GPIO ポート、ピン、および HSIOM の設定を理解するのに役立ちます。

- 各ポート番号は、header フォルダのデバイスヘッダで定義されます。
例えば、MCU シリコンのリビジョンがリビジョン B の場合、デバイスヘッダは `hdr/rev_b` にあります。

```
#define GPIO_PRT13 ((volatile stc_GPIO_PRT_t*) &GPIO->PRT[13])
```

ポート番号とピン番号は、同じフォルダのプロジェクト固有のヘッダと一緒に定義されます。

例えば、MCU がリビジョン C CPU ボードの CYT2B7 シリーズである場合、`bb_bsp_tviibe1m_rev.c.h` で次の定義を見つけることができます。

```
#define CY_USB_SCB_UART_RX_PORT    GPIO_PRT13
#define CY_USB_SCB_UART_RX_PIN    0
#define CY_USB_SCB_UART_RX_MUX    P13_0_SCB3_UART_RX
#define CY_USB_SCB_UART_TX_PORT    GPIO_PRT13
#define CY_USB_SCB_UART_TX_PIN    1
#define CY_USB_SCB_UART_TX_MUX    P13_1_SCB3_UART_TX
```

- 特定のピンの HSIOM とそのパラメータ値は、デバイスの GPIO ヘッダにあります。
例えば、MCU が 176 ピンの CYT2B7 シリーズの場合、ヘッダファイルは `gpio_cyt2b7_176_lqfp.h` です。

3 GPIO 設定

176 ピン CYT2B7 シリーズ MCU の場合、P13.0 および P13.1 の HSIOM 接続は次のとおりです。

```

/* P13.1 */
P13_0_GPIO                = 0,    /* GPIO controls 'out' */
P13_0_AMUXA               = 4,    /* Analog mux bus A */
P13_0_AMUXB               = 5,    /* Analog mux bus B */
P13_0_AMUXA_DSI           = 6,
/* Analog mux bus A, DSI control */

P13_0_AMUXB_DSI           = 7,
/* Analog mux bus B, DSI control */

P13_0_TCPWM0_LINE264      = 8,
/* Digital Active - tcpwm[0].line[264]:0 */

P13_0_TCPWM0_LINE_COMPL43 = 9,
/* Digital Active - tcpwm[0].line_compl[43]:0 */

P13_0_TCPWM0_TR_ONE_CNT_IN792 = 10,
/* Digital Active - tcpwm[0].tr_one_cnt_in[792]:0 */

P13_0_TCPWM0_TR_ONE_CNT_IN130 = 11,
/* Digital Active - tcpwm[0].tr_one_cnt_in[130]:0 */

P13_0_PASS0_SAR_EXT_MUX_SEL6 = 16,
/* Digital Active - pass[0].sar_ext_mux_sel[6] */

P13_0_SCB3_UART_RX        = 17,
/* Digital Active - scb[3].uart_rx:0 */

P13_0_SCB3_SPI_MISO       = 19,
/* Digital Active - scb[3].spi_miso:0 */

/* P13.1 */
P13_1_GPIO                = 0,    /* GPIO controls 'out' */
P13_1_AMUXA               = 4,    /* Analog mux bus A */
P13_1_AMUXB               = 5,    /* Analog mux bus B */
P13_1_AMUXA_DSI           = 6,
/* Analog mux bus A, DSI control */

P13_1_AMUXB_DSI           = 7,
/* Analog mux bus B, DSI control */

P13_1_TCPWM0_LINE44       = 8,
/* Digital Active - tcpwm[0].line[44]:0 */

P13_1_TCPWM0_LINE_COMPL264 = 9,
/* Digital Active - tcpwm[0].line_compl[264]:0 */

P13_1_TCPWM0_TR_ONE_CNT_IN132 = 10,
/* Digital Active - tcpwm[0].tr_one_cnt_in[132]:0 */

```

3 GPIO 設定

```
P13_1_TCPWM0_TR_ONE_CNT_IN793    = 11,
/* Digital Active - tcpwm[0].tr_one_cnt_in[793]:0 */

P13_1_PASS0_SAR_EXT_MUX_SEL7      = 16,
/* Digital Active - pass[0].sar_ext_mux_sel[7] */

P13_1_SCB3_UART_TX                = 17,
/* Digital Active - scb[3].uart_tx:0 */

P13_1_SCB3_I2C_SDA                = 18,
/* Digital Active - scb[3].i2c_sda:0 */

P13_1_SCB3_SPI_MOSI               = 19,
/* Digital Active - scb[3].spi_mosi:0 */
```

各ピンの特定の HSIOM 機能接続については、デバイスのデータシートを参照してください。

[Code Listing 18](#) に、SCB RX/ TX の GPIO を初期化する設定部のサンプルプログラムを示します。ドライバ部のサンプルプログラムは [Code Listing 1](#) と同じです。

3 GPIO 設定

Code Listing 18 設定部の SCB RX/TX GPIO 初期化例

```
int main(void)
{
    SystemInit();
    :
    __enable_irq(); /* Enable global interrupts. */

    /* Initialize Port and Clock */
    /*Contains port configurations */
    Peripheral_Initialization();
    :
}

void Peripheral_Initialization(void)
{
    cy_stc_gpio_pin_config_t    stc_port_pin_cfg_uart = {0};
    :

    /*-----*/
    /* Port Configuration for UART */
    /*-----*/

    /* P13.0 -> scb[3].uart_rx */
    /* Set drive mode for P13.0 */
    stc_port_pin_cfg_uart.driveMode = CY_GPIO_DM_HIGHZ;    /*Set HSIOM to SCB UART RX*/
    stc_port_pin_cfg_uart.hsiom      = CY_USB_SCB_UART_RX_MUX;
    /* See Initializing GPIO */
    Cy_GPIO_Pin_Init(CY_USB_SCB_UART_RX_PORT, CY_USB_SCB_UART_RX_PIN, &stc_port_pin_cfg_uart);

    /* P13.1 -> scb[3].uart_tx */
    /* Set drive mode for P13.1 */
    stc_port_pin_cfg_uart.driveMode = CY_GPIO_DM_STRONG_IN_OFF;    /* Set HSIOM to SCB UART TX */
    stc_port_pin_cfg_uart.hsiom      = CY_USB_SCB_UART_TX_MUX;
    /* See Initializing GPIO */
    Cy_GPIO_Pin_Init(CY_USB_SCB_UART_TX_PORT, CY_USB_SCB_UART_TX_PIN, &stc_port_pin_cfg_uart);
    :
}
```

3.5.2 TCPWM ポート設定

この例は、CYT2B7 シリーズの TCPWM 出力用のポートの設定を示します。この例では、TCPWM ch0 ポート (P6.1) が使用され、HSIOM の ACT_0 に割り当てられています。ポートの設定後、TCPWM 機能を設定します。

各機能に適切なポートを選択する方法の詳細については、[データシートの「Alternate function pin assignments」](#) セクションを参照してください。TCPWM 設定の詳細については、[関連ドキュメント](#)に記載されている AN220224 を参照してください。

[図 8](#) に、信号パスを P6.1 の TCPWM ポートとして示します。PWM ch0 を使用するには、HSIOM を PWM_0 に設定してください。

3 GPIO 設定

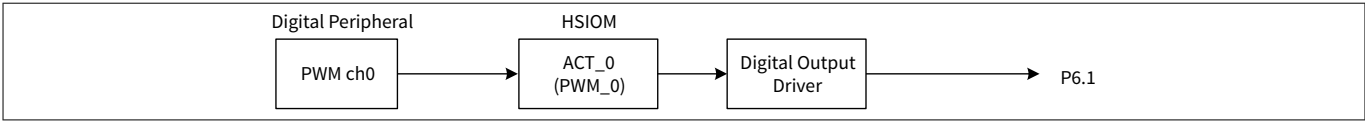


図 8 TCPWM ポート設定

設定例:

- ポート番号: P6.1
- 初期状態: Low
- 入出力: 出力
- ドライブモード: Strong
- 機能設定 (HSIOM): TCPWM
- 割込み機能: 未使用

表 8 に、TCPWM ポートの SDL の設定部のパラメータを示します。

表 8 TCPWM ポート GPIO ピンパラメータリスト

パラメータ	説明	値
.outVal	ピン出力状態選択 0: 出力状態に影響しない、1: 出力状態を”0”に設定	0ul

(続く)

3 GPIO 設定

表 8 (続き) TCPWM ポート GPIO ピンパラメータリスト

パラメータ	説明	値
.driveMode	I/O ピン GPIO ドライブモード選択 0: アナログハイインピーダンス 1: 予約済み、使用できません。 2: 抵抗プルアップ 3: 抵抗プルダウン 4: オープンドレイン、ロードドライブ 5: オープンドレイン、ハイドライブ 6: 強カドライブ 7: 抵抗プルアップ/プルダウン 8: デジタルハイインピーダンス。入力バッファオン 9: 予約済み、使用できません。 10: 抵抗プルアップ。入力バッファオン 11: 抵抗プルダウン。入力バッファオン 12: オープンドレイン、ロードドライブ。入力バッファオン 13: オープンドレイン、ハイドライブ。入力バッファオン 14: 強カドライブ。入力バッファオン 15: 抵抗プルアップ/プルダウン。入力バッファオン	CY_GPIO_DM_STRONG_IN_OFF = 6ul
.hsiom	I/O ピン 0 ルート接続設定	P6_1_TCPWM0_LINE0 SDL 説明の 2 番を参照してください。
.intEdge	I/O ピン 0 割込みトリガエッジ設定 0: ディセーブル, 1: 立上りエッジ, 2: 立下りエッジ, 3: 両エッジ	0ul
.intMask	I/O ピンエッジ割込みのマスク 0: ピン割込み転送ディセーブル 1: ピン割込み転送イネーブル	0ul
.vtrip	ピン 0 入力バッファモード選択 0: CMOS, 1: TTL	0ul

(続く)

3 GPIO 設定

表 8 (続き) TCPWM ポート GPIO ピンパラメータリスト

パラメータ	説明	値
.slewRate	I/O ピンスルーレート選択 0: 高速スルーレート, 1: 低速スルーレート	0ul
.driveSel	I/O ピン GPIO ドライブ強度設定 0: Full ドライブ強度 1: Full ドライブ強度 2: 1/2 ドライブ強度 3: 1/4 ドライブ強度	0ul

次の説明は、SDL でのこの例の GPIO ポート、ピン、および HSIOM の設定を理解するのに役立ちます。

- 各ポート番号は、header フォルダのデバイスヘッダで定義されます。
例えば、MCU シリコンのリビジョンがリビジョン B の場合、デバイスヘッダは `hdr/rev_b` にあります。

```
#define GPIO_PRT6      ((volatile stc_GPIO_PRT_t*) &GPIO->PRT[6])
```

ポート番号とピン番号は、同じフォルダのデバイス GPIO ヘッダで一緒に定義されます。

例えば、MCU が 176 ピンの CYT2B7 シリーズの場合、次の定義は `gpio_cyt2b7_176_lqfp.h` にあります。

```
#define P6_1_PORT      GPIO_PRT6
#define P6_1_PIN       1u
```

- 各 GPIO ピンには、専用の HSIOM 選択があります。特定のピンの HSIOM とそのパラメータ値については、デバイスの GPIO ヘッダを参照してください。デバイスのシリーズに応じて、ピンの HSIOM を設定します。

3 GPIO 設定

176 ピン CYT2B7 シリーズ MCU の場合、P6.1 の HSIOM 接続は次のとおりです。

```
/* P6.1 */
P6_1_GPIO                = 0,    /* GPIO controls 'out' */
P6_1_AMUXA                = 4,    /* Analog mux bus A */
P6_1_AMUXB                = 5,    /* Analog mux bus B */
P6_1_AMUXA_DSI            = 6,    /* Analog mux bus A, DSI control */
/* Analog mux bus A, DSI control */

P6_1_AMUXB_DSI            = 7,    /* Analog mux bus B, DSI control */
/* Analog mux bus B, DSI control */

P6_1_TCPWM0_LINE0         = 8,    /* Digital Active - tcpwm[0].line[0]:0 */
/* Digital Active - tcpwm[0].line[0]:0 */

P6_1_TCPWM0_LINE_COMPL256 = 9,    /* Digital Active - tcpwm[0].line_compl[256]:0 */
/* Digital Active - tcpwm[0].line_compl[256]:0 */

P6_1_TCPWM0_TR_ONE_CNT_IN0 = 10,   /* Digital Active - tcpwm[0].tr_one_cnt_in[0]:0 */
/* Digital Active - tcpwm[0].tr_one_cnt_in[0]:0 */

P6_1_TCPWM0_TR_ONE_CNT_IN769 = 11, /* Digital Active - tcpwm[0].tr_one_cnt_in[769]:0 */
/* Digital Active - tcpwm[0].tr_one_cnt_in[769]:0 */

P6_1_SCB4_UART_TX         = 17,    /* Digital Active - scb[4].uart_tx:0 */
/* Digital Active - scb[4].uart_tx:0 */

P6_1_SCB4_I2C_SDA         = 18,    /* Digital Active - scb[4].i2c_sda:0 */
/* Digital Active - scb[4].i2c_sda:0 */

P6_1_SCB4_SPI_MOSI        = 19,    /* Digital Active - scb[4].spi_mosi:0 */
/* Digital Active - scb[4].spi_mosi:0 */

P6_1_LIN0_LIN_TX3         = 20,    /* Digital Active - lin[0].lin_tx[3]:0 */
/* Digital Active - lin[0].lin_tx[3]:0 */
```

図 9 に、P6.1 の代替機能を使用したピン割当てを示します。P6_1_TCPWM0_LINE0 の「LINE0」は、TCPWM カウンタ番号 0 を示す「PWM_0」を表します。

PWM_M_0/PWM_14_N/TC_M_0_TR0/TC_14_TR1/SCB4_RX/SCB4_MISO/LIN3_RX/ADC[0]_0	P6.0	35
PWM_0/PWM_M_0_N/TC_0_TR0/TC_M_0_TR1/SCB4_TX/SCB4_SDA/SCB4_MOSI/LIN3_TX/ADC[0]_1	P6.1	36
PWM_M_1/PWM_0_N/TC_M_1_TR0/TC_0_TR1/SCB4_RTS/SCB4_SCL/SCB4_CLK/LIN3_EN/CAN0_2_TX/ADC[0]_2	P6.2	37

図 9 P6.1 CYT2B7 シリーズの代替機能によるピン割当て

各ピンの特定の HSIOM 機能接続については、デバイスのデータシートを参照してください。

Code Listing 19 に、TCPWM の GPIO を初期化する設定部のサンプルプログラムを示します。ドライバ部のサンプルプログラムは Code Listing 1 と同じです。

3 GPIO 設定

Code Listing 19 設定部の TCPWM GPIO 初期化例

```

/* TCPWM_TR_ONE_CNT_IN0 */
#define TCPWM_LINEx_PORT      GPIO_PRT6
#define TCPWM_LINEx_PIN      1u
#define TCPWM_LINEx_MUX      P6_1_TCPWM0_LINE0

cy_stc_gpio_pin_config_t pin_cfg1 =
{
    .outVal      = 0uL,          /* Pin output state */
    .driveMode   = CY_GPIO_DM_STRONG_IN_OFF, /* Drive mode */
    .hsiom       = TCPWM_LINEx_MUX, /* HSIOM selection */
    .intEdge     = 0uL,          /* Interrupt Edge type */
    .intMask     = 0uL,          /* Interrupt enable mask */
    .vtrip       = 0uL,          /* Input buffer voltage trip type */
    .slewRate    = 0uL,          /* Output buffer slew rate */
    .driveSel    = 0uL,          /* Drive strength */
};

int main(void)
{
    :
    /*-----*/
    /* Port Configuration for TCPWM */
    /*-----*/
    Cy_GPIO_Pin_Init(TCPWM_LINEx_PORT, TCPWM_LINEx_PIN, &pin_cfg1);
    :
}

```

3.5.3 アナログポート設定

この例は、CYT4BF シリーズのアナログポートの設定を示します。この例では、ADC[0]_0 がアナログ入力に割り当てられています。ポートの GPIO ドライブモードを High-Z に設定し、クローバ電流を回避するために入力バッファを無効にします。SAR ADC 設定の詳細については、[関連ドキュメント](#)に記載されている AN219755 を参照してください。

設定例:

- ポート番号: P12.6
- 初期状態: Low
- 入出力: 入力
- ドライブモード: ハイインピーダンス
- 機能設定 (HSIOM): GPIO
- 割込み機能: 未使用

表 9 に、SDL のアナログポートの設定部のパラメータを示します。

3 GPIO 設定

表 9 GPIO ピンパラメータリスト

パラメータ	説明	値
.outVal	ピン出力状態選択 0: 出力状態に影響しない、1: 出力状態を”0”に設定	0ul
.driveMode	I/O ピン GPIO ドライブモード選択 0: アナログハイインピーダンス 1: 予約済み、使用できません。 2: 抵抗プルアップ 3: 抵抗プルダウン 4: オープンドレイン、ロードドライブ 5: オープンドレイン、ハイドライブ 6: 強力ドライブ 7: 抵抗プルアップ/プルダウン 8: デジタルハイインピーダンス。入力バッファオン 9: 予約済み、使用できません。 10: 抵抗プルアップ。入力バッファオン 11: 抵抗プルダウン。入力バッファオン 12: オープンドレイン、ロードドライブ。入力バッファオン 13: オープンドレイン、ハイドライブ。入力バッファオン 14: 強力ドライブ。入力バッファオン 15: 抵抗プルアップ/プルダウン。入力バッファオン	CY_GPIO_DM_ANALOG = 0ul
.hsiom	I/O ピン 0 ルート接続設定	P12_6_GPIO SDL 説明の 2 番を参照してください。
.intEdge	I/O ピン 0 割込みトリガエッジ設定 0: ディセーブル, 1: 立上りエッジ, 2: 立下りエッジ, 3: 両エッジ	0ul
.intMask	I/O ピンエッジ割込みのマスク 0: ピン割込み転送ディセーブル 1: ピン割込み転送イネーブル	0ul
.vtrip	ピン 0 入力バッファモード選択 0: CMOS, 1: TTL	0ul

(続く)

3 GPIO 設定

表 9 (続き) GPIO ピンパラメータリスト

パラメータ	説明	値
.slewRate	I/O ピンスルーレート選択 0: 高速スルーレート, 1: 低速スルーレート	0ul
.driveSel	I/O ピン GPIO ドライブ強度設定 0: Full ドライブ強度 1: Full ドライブ強度 2: 1/2 ドライブ強度 3: 1/4 ドライブ強度	0ul

次の説明は、SDL でこの例の GPIO ポート、ピン、および HSIOM の設定を理解するのに役立ちます。

- 各ポート番号は、デバイスヘッダで定義されており、header フォルダにあります。
例えば、MCU のリビジョンがリビジョン D の場合、デバイスヘッダは `hdr/rev_d` にあります。

```
#define GPIO_PRT12 ((volatile stc_GPIO_PRT_t*) &GPIO->PRT[12])
```

ポート番号とピン番号は、同じフォルダのプロジェクト固有のヘッダと一緒に定義されます。
例えば、MCU が CYT4BF シリーズの場合、`bb_bsp_tviibh8m.h` で次の定義を見つけることができます。

```
#define CY_ADC_POT_PORT      GPIO_PRT12
#define CY_ADC_POT_PIN      6
#define CY_ADC_POT_PIN_MUX  P12_6_GPIO
```

- 特定のピンの HSIOM とそのパラメータ値は、デバイスの GPIO ヘッダにあります。
例えば、MCU が 176 ピン CYT4BF シリーズの場合、ヘッダファイルは `gpio_cyt4bf_176_teqfp.h` です。
176 ピン CYT4BF シリーズ MCU の場合、P12.6 の HSIOM 接続は次のとおりです。

```
/* P12.6 */
P12_6_GPIO           = 0,    /* N/A */
P12_6_AMUXA          = 4,    /* AMUXBUS A */
P12_6_AMUXB          = 5,    /* AMUXBUS B */
P12_6_AMUXA_DSI      = 6,    /* N/A */
P12_6_AMUXB_DSI      = 7,    /* N/A */
P12_6_TCPWM1_LINE42  = 8,
/* Digital Active - tcpwm[1].line[42]:0 */
P12_6_TCPWM1_LINE_COMPL41 = 9,
/* Digital Active - tcpwm[1].line_compl[41]:0 */
P12_6_TCPWM1_TR_ONE_CNT_IN126 = 10,
/* Digital Active - tcpwm[1].tr_one_cnt_in[126]:0 */
P12_6_TCPWM1_TR_ONE_CNT_IN124 = 11,
/* Digital Active - tcpwm[1].tr_one_cnt_in[124]:0 */
```

各ピンの特定の HSIOM 機能接続については、デバイスのデータシートを参照してください。

Code Listing 20 に、アナログポートの GPIO を初期化する設定部のサンプルプログラムを示します。ドライバ部のサンプルプログラムは Code Listing 1 と同じです。

3 GPIO 設定

Code Listing 20 設定部のアナログポート GPIO 初期化例

```
int main(void)
{
    :
    /* ADC port setting (Note default port setting after reset is just fine) */
    {
        cy_stc_gpio_pin_config_t adcPinConfig =
        {
            .outVal      = 0ul,          /* Pin output state */
            .driveMode    = CY_GPIO_DM_ANALOG, /* Drive mode */
            .hsiom        = CY_ADC_POT_PIN_MUX, /* HSIOM selection */
            .intEdge      = 0ul,          /* Interrupt Edge type */
            .intMask      = 0ul,          /* Interrupt enable mask */
            .vtrip        = 0ul,          /* Input buffer voltage trip type */
            .slewRate     = 0ul,          /* Output buffer slew rate */
            .driveSel     = 0ul,          /* Drive strength */
        };
        Cy_GPIO_Pin_Init(CY_ADC_POT_PORT, CY_ADC_POT_PIN, &adcPinConfig);
    }
    :
}
```

4 Appendix A. I/O ポート設定例

4 Appendix A. I/O ポート設定例

ここでは、さまざまな周辺機能を使用する場合の一般的な I/O ポートの設定例を説明します。以下に、SDL の CYT4BF シリーズ (BGA-320 ピンパッケージ) の設定例を示します。

表 10 I2C の I/O ポート設定例

パラメータ	SCB0_SDA(P1.1)	SCB0_SCL(P1.0)
outVal	0	0
driveMode (Master/Slave)	12	12
hsiom	14	14
intEdge	0	0
intMask	0	0
vtrip	0	0
slewRate	0	0
driveSel	0	0

表 11 SPI の I/O ポート設定例

パラメータ	SCB5_MISO(P1.0)	SCB0_MOSI(P1.1)	SCB0_CLK(P1.2)	SCB0_SEL0(P1.3)
outVal	0	0	0	0
driveMode (Master/Slave)	8/6	6/8	6/8	6/8
hsiom	30	30	30	30
intEdge	0	0	0	0
intMask	0	0	0	0
vtrip	0	0	0	0
slewRate	0	0	0	0
driveSel	0	0	0	0

表 12 LIN の I/O ポート設定例

パラメータ	LIN0_RX(P1.2)	LIN0_TX(P21.6)
outVal	0	1
driveMode	8	14
hsiom	20	20
intEdge	0	0
intMask	0	0
vtrip	0	0
slewRate	0	0
driveSel	0	0

4 Appendix A. I/O ポート設定例

表 13 CAN の I/O ポート設定例

パラメータ	CAN1_3_RX(P15.1)	CAN1_3_TX(P15.0)
outVal	0	1
driveMode	8	14
hsiom	21	21
intEdge	0	0
intMask	0	0
vtrip	0	0
slewRate	0	0
driveSel	0	0

表 14 I2S の I/O ポート設定例

パラメータ	AUDIOSS0_MCLK(P1.0)	AUDIOSS0_TX_SCK(P1.1)	AUDIOSS0_TX_WS(P11.2)	AUDIOSS0_TX_SDO(P12.0)	AUDIOSS0_CLK_I2S_IF(P12.1)	AUDIOSS0_RX_SCK(P12.2)	AUDIOS0_RX_WS(P12.3)	AUDIOS0_RX_SDI(P12.4)
outVal	0	0	0	0	0	0	0	0
driveMode	6	6	6	6	8	8	8	8
hsiom	25	25	25	25	25	25	25	25
intEdge	0	0	0	0	0	0	0	0
intMask	0	0	0	0	0	0	0	0
vtrip	0	0	0	0	0	0	0	0
slewRate	0	0	0	0	0	0	0	0
driveSel	0	0	0	0	0	0	0	0

表 15 SMIF の I/O ポート設定例

パラメータ	SPIHB_CLK (P24.1)	SPIHB_RWDS (P24.2)	SPIHB_SEL0/1 (P24.3, 4)	SPIHB_DATA0-7 (P25.0-7)
outVal	0	0	0	0
driveMode	14	14	15	14
hsiom	23	23	23	23
intEdge	0	0	0	0
intMask	0	0	0	0

(続く)

4 Appendix A. I/O ポート設定例

表 15 (続き) SMIF の I/O ポート設定例

パラメータ	SPIHB_CLK (P24.1)	SPIHB_RWDS (P24.2)	SPIHB_SEL0/1 (P24.3, 4)	SPIHB_DATA0-7 (P25.0-7)
vtrip	0	0	0	0
slewRate	0	0	0	0
driveSel	0	0	0	0
vregEn	0	0	0	0
ibufMode	0	0	0	0
vtripSel	0	0	0	0
vrefSel	0	0	0	0
vohSel	0	0	0	0

表 16 ETH インタフェースの I/O ポート設定例 (1/2)

パラメータ	ETH1_RX_ER(P34.3)	ETH1_MDIO(P27.5)	ETH1_MDC(P27.6)	ETH1_REF_CLK(P26.0)	ETH1_TX_CTL(P26.1)	ETH1_TX_ER(P33.1)
outVal	0	0	0	0	0	0
driveMode	8	14	6	8	6	6
hsiom	27	27	27	27	27	27
intEdge	0	0	0	0	0	0
intMask	0	0	0	0	0	0
vtrip	0	0	0	0	0	0
slewRate	0	0	0	0	0	0
driveSel	0	0	0	0	0	0
vregEn	0	0	0	0	0	0
ibufMode	0	0	0	0	0	0
vtripSel	0(RGMII) 3(GMII) 1(others)	1	1	0(RGMII) 3(GMII) 1(others)	0(RGMII) 3(GMII) 1(others)	0(RGMII) 3(GMII) 1(others)
vrefSel	0	0	0	0	0	0
vohSel	0	0	0	0	0	0

表 17 ETH インタフェースの I/O ポート設定例 (2/2)

パラメータ	ETH1_TX_CLK(P26.2)	ETH1_TXD(P26.3-6, 33.3-6)	ETH1_RXD(P26.7, 27.0-2, 33.7, 34.0-2)	ETH1_RX_CTL(P27.3)	ETH1_RX_CLK(P27.4)
outVal	0	0	0	0	0

(続く)

4 Appendix A. I/O ポート設定例

表 17 (続き) ETH インタフェースの I/O ポート設定例 (2/2)

パラメータ	ETH1_TX_CLK(P2 6.2)	ETH1_TXD(P26 .3-6, 33.3-6)	ETH1_RXD(P26 .7, 27.0-2, 33.7, 34.0-2)	ETH1_RX_CTL(P2 7.3)	ETH1_RX_CLK(P2 7.4)
driveMode	8(MII) 6(others)	6	8	8	8
hsiom	27	27	27	27	27
intEdge	0	0	0	0	0
intMask	0	0	0	0	0
vtrip	0	0	0	0	0
slewRate	0	0	0	0	0
driveSel	0	0	0	0	0
vregEn	0	0	0	0	0
ibufMode	0	0	0	0	0
vtripSel	0(RGMII) 3(GMII) 1(others)	0(RGMII) 3(GMII) 1(others)	0(RGMII) 3(GMII) 1(others)	0(RGMII) 3(GMII) 1(others)	0(RGMII) 3(GMII) 1(others)
vrefSel	0	0	0	0	0
vohSel	0	0	0	0	0

表 18 SDHC インタフェースの I/O ポート設定例

パラメータ	SDHC_CARD_MECH_WR ITE_PROT(P6.2)	SDHC_CARD_ CMD(P6.3)	SDHC_CLK_C ARD(P6.4)	SDHC_CARD_DET ECT_N(P6.5)	SDHC_CARD_DAT_ 3TO0_x(P7.1-4)
outVal	0	0	0	0	0
driveMode	14	14	14	14	14
hsiom	25	25	25	25	25
intEdge	0	0	0	0	0
intMask	0	0	0	0	0
vtrip	0	0	0	0	0
slewRate	0	0	0	0	0
driveSel	0	0	0	0	0
vregEn	0	0	0	0	0
ibufMode	0	0	0	0	0
vtripSel	0	0	0	0	0
vrefSel	0	0	0	0	0
vohSel	0	0	0	0	0

4 Appendix A. I/O ポート設定例

表 19 に各要素値を説明します。

表 19 GPIO ピンパラメータリスト

パラメータ	説明
outVal	ピン出力状態選択 0: 出力状態に影響しない 1: 出力状態を”0”に設定
driveMode	I/O ピン GPIO ドライブモード選択 0: アナログハイインピーダンス 1: 予約済み、使用できません。 2: 抵抗プルアップ 3: 抵抗プルダウン 4: オープンドレイン、ロードドライブ 5: オープンドレイン、ハイドライブ 6: 強カドライブ 7: 抵抗プルアップ/プルダウン 8: デジタルハイインピーダンス。入力バッファオン 9: 予約済み、使用できません。 10: 抵抗プルアップ。入力バッファオン 11: 抵抗プルダウン。入力バッファオン 12: オープンドレイン、ロードドライブ。入力バッファオン 13: オープンドレイン、ハイドライブ。入力バッファオン 14: 強カドライブ。入力バッファオン 15: 抵抗プルアップ/プルダウン。入力バッファオン
.hsiom	I/O ピン 0 ルート接続設定
.intEdge	I/O ピン 0 割込みトリガエッジ設定 0: ディセーブル, 1: 立上リエッジ, 2: 立下リエッジ, 3: 両エッジ
.intMask	I/O ピンエッジ割込みのマスク 0: ピン割込み転送ディセーブル 1: ピン割込み転送イネーブル
.vtrip	ピン 0 入力バッファモード選択 0: CMOS, 1: TTL
.slewRate	I/O ピンスルーレート選択 0: 高速スルーレート, 1: 低速スルーレート
.driveSel	I/O ピン GPIO ドライブ強度設定 0: Full ドライブ強度 1: Full ドライブ強度 2: 1/2 ドライブ強度 3: 1/4 ドライブ強度

4 Appendix A. I/O ポート設定例

注: サポートされる機能と機能のポートピン番号については、固有の**デバイスデータシート**を参照してください。

5 用語集

5 用語集

表 20 用語集

用語	説明
ADC	Analog-to-digital converter (アナログ-デジタル変換器)。詳細については、 アーキテクチャ TRM の「SAR ADC」章を参照してください。
GPIO	General-purpose I/O (汎用 I/O)
HSIOM	High-speed I/O matrix (高速 I/O マトリクス)。詳細については、 アーキテクチャ TRM の I/O System 章の High-Speed I/O Matrix を参照してください。
SCB	Serial Communications Block (シリアル通信ブロック)。詳細については、 アーキテクチャ TRM の「Serial Communications Block (SCB)」章を参照してください。
スルーレート制御 (Slew rate control)	単位時間当たりの電圧変化。詳細については、 アーキテクチャ TRM の I/O System 章の Slew Rate Control を参照してください。
TCPWM	Timer, Counter, and Pulse Width Modulator (タイマ, カウンタ, およびパルス幅変調器)。詳細については、 アーキテクチャ TRM の「Timer, Counter, and PWM」章を参照してください。

6 関連ドキュメント

6 関連ドキュメント

以下は、TRAVEO™ T2G ファミリシリーズのデータシート、テクニカルリファレンスマニュアル、アプリケーションノートです。これらドキュメントの入手については[テクニカルサポート](#)に連絡してください。

[1] デバイスデータシート

- [CYT2B6 datasheet 32-bit Arm®Cortex®-M4F microcontroller TRAVEO™ T2G family](#)
- [CYT2B7 datasheet 32-bit Arm®Cortex®-M4F microcontroller TRAVEO™ T2G family](#)
- [CYT2B9 datasheet 32-bit Arm®Cortex®-M4F microcontroller TRAVEO™ T2G family](#)
- [CYT2BL datasheet 32-bit Arm®Cortex®-M4F microcontroller TRAVEO™ T2G family](#)
- [CYT3BB/4BB datasheet 32-bit Arm®Cortex®-M7 microcontroller TRAVEO™ T2G family](#)
- [CYT4BF datasheet 32-bit Arm®Cortex®-M7 microcontroller TRAVEO™ T2G family](#)
- [CYT6BJ datasheet 32-bit Arm®Cortex®-M7 microcontroller TRAVEO™ T2G family \(Doc No. 002-33466\)](#)
- [CYT3DL datasheet 32-bit Arm®Cortex®-M7 microcontroller TRAVEO™ T2G family](#)
- [CYT4DN datasheet 32-bit Arm®Cortex®-M7 microcontroller TRAVEO™ T2G family](#)
- [CYT4EN datasheet 32-bit Arm®Cortex®-M7 microcontroller TRAVEO™ T2G family \(Doc No. 002-30842\)](#)
- [CYT2CL datasheet 32-bit Arm®Cortex®-M7 microcontroller TRAVEO™ T2G family](#)

[2] Body Controller Entry ファミリ

- [TRAVEO™ T2G automotive body controller entry family architecture technical reference manual \(TRM\)](#)
- [TRAVEO™ T2G automotive body controller entry registers technical reference manual \(TRM\) for CYT2B7](#)
- [TRAVEO™ T2G automotive body controller entry registers technical reference manual \(TRM\) for CYT2B9](#)
- [TRAVEO™ T2G automotive body controller entry registers technical reference manual \(TRM\) for CYT2BL \(Doc No. 002-29852\)](#)

[3] Body Controller High ファミリ

- [TRAVEO™ T2G automotive body controller high family architecture technical reference manual \(TRM\)](#)
- [TRAVEO™ T2G automotive body controller high registers technical reference manual \(TRM\) for CYT3BB/4BB](#)
- [TRAVEO™ T2G automotive body controller high registers technical reference manual \(TRM\) for CYT4BF](#)
- [TRAVEO™ T2G automotive body controller high registers technical reference manual \(TRM\) for CYT6BJ \(Doc No. 002-36068\)](#)

[4] Cluster 2D ファミリ

- [TRAVEO™ T2G automotive cluster 2D architecture technical reference manual \(TRM\)](#)
- [TRAVEO™ T2G automotive cluster 2D registers technical reference manual \(TRM\) for CYT3DL](#)
- [TRAVEO™ T2G automotive cluster 2D registers technical reference manual \(TRM\) for CYT4DN](#)
- [TRAVEO™ T2G automotive cluster 2D registers technical reference manual \(TRM\) for CYT4EN \(Doc No. 002-35181\)](#)

[5] クラスタ Entry ファミリ

- [TRAVEO™ T2G automotive cluster entry architecture technical reference manual \(TRM\)](#)
- [TRAVEO™ T2G automotive cluster entry registers technical reference manual \(TRM\) for CYT2CL](#)

[6] アプリケーションノート

- [AN219755 – TRAVEO™ T2G 車載マイクロコントローラーでの SAR ADC の使用](#)
- [AN219842 – TRAVEO™ T2G の割込みの使用方法](#)

6 関連ドキュメント

- [AN225401 – TRAVEO™ T2G ファミリのシリアル通信ブロック \(SCB\) 使用方法](#)
- [AN220224–TRAVEO™ T2G ファミリのタイマ, カウンタ, および PWM \(TCPWM\) の設定方法](#)

7 その他の参考資料

7 その他の参考資料

さまざまな周辺機器にアクセスするためのサンプルソフトウェアとしてのスタートアップを含むサンプルドライバライブラリ (SDL) が提供されます。SDL は、公式の AUTOSAR 製品でカバーされないドライバの顧客へのリファレンスとしても機能します。SDL は自動車用規格に適合していないため、量産用としては使用できません。このアプリケーションノートのプログラムコードは SDL の一部です。SDL の入手については、[テクニカルサポート](#)に連絡してください。

改訂履歴

改訂履歴

版数	発行日	変更内容
**	2019-06-19	これは英語版 002-20193 Rev. **を翻訳した日本語版 Rev. **です。英語版の改訂内容: New application note.
英語版*A	-	この版は英語版のみです。英語版の改訂内容: Updated Associated Part Family as “TRAVEO™ T2G Family CYT2/CYT3/CYT4 Series”. Changed target part numbers from “CYT2B/CYT4B Series” to “CYT2/CYT4 Series” in all instances across the document. Added target part numbers “CYT3 Series” related information in all instances across the document.
*A	2020-09-25	これは英語版 002-20193 Rev. *B を翻訳した日本語版 Rev. *A です。英語版の改訂内容: Updated GPIO Settings: Added flowchart and example codes in all instances.
*B	2021-07-20	これは英語版 002-20193 Rev. *C を翻訳した日本語版 Rev. *B です。英語版の改訂内容: テンプレートの変更を実施。
*C	2022-09-05	これは英語版 002-20193 Rev. *D を翻訳した日本語版 Rev. *C です。英語版の改訂内容: Updated 表 3 to 表 9 Added Appendix A. I/O ポート設定例 section
*D	2024-08-14	これは英語版 002-20193 Rev. *E を翻訳した日本語版 Rev. *D です。英語版の改訂内容: Template update; no content update.
*E	2025-01-16	これは英語版 002-20193 Rev. *F を翻訳した日本語版 Rev. *E です。英語版の改訂内容: Updated References.

Trademarks

All referenced product or service names and trademarks are the property of their respective owners.

Edition 2025-01-16

Published by

Infineon Technologies AG
81726 Munich, Germany

© 2025 Infineon Technologies AG
All Rights Reserved.

Do you have a question about any aspect of this document?

Email: erratum@infineon.com

Document reference
IFX-hlp1681384982415

重要事項

本手引書に記載された情報は、本製品の使用に関する手引きとして提供されるものであり、いかなる場合も、本製品における特定の機能性能や品質について保証するものではありません。本製品の使用前に、当該手引書の受領者は実際の使用環境の下であらゆる本製品の機能及びその他本手引書に記された一切の技術的情報について確認する義務が有ります。インフィニオンテクノロジーズはここに当該手引書内で記される情報につき、第三者の知的所有権の不侵害の保証を含むがこれに限らず、あらゆる種類の一切の保証および責任を否定いたします。

本文書に含まれるデータは、技術的訓練を受けた従業員のみを対象としています。本製品の対象用途への適合性、およびこれら用途に関連して本文書に記載された製品情報の完全性についての評価は、お客様の技術部門の責任にて実施してください。

警告事項

技術的要件に伴い、製品には危険物質が含まれる可能性があります。当該種別の詳細については、インフィニオンの最寄りの営業所までお問い合わせください。

インフィニオンの正式代表者が署名した書面を通じ、インフィニオンによる明示の承認が存在する場合を除き、インフィニオンの製品は、当該製品の障害またはその使用に関する一切の結果が、合理的に人的傷害を招く恐れのある一切の用途に使用することはできないこと予めご了承ください。