
GPIOでの開始に際して

序説

著者: Alexandru Niculae, Catalin Visan, Microchip Technology Inc.

この資料の目的はMicrochip tinyAVR[®] 0及び1系とmegaAVR[®] 0系デバイスでマイクロコントローラのピンを汎用入出力(GPIO:General Purpose Input/Output)としてどう使うかを段階的に記述することです。以下の使用事例が提供されます。

- **LED点滅:**
遅延を使ってLEDを点滅、500ms毎にLEDを交互切り替え
- **長/短ボタン:**
遅延閾値によって定義される長/短のボタン間を区別するための入力としてピンを使用
- **ボタンでの起き上がり:**
ボタンで休止を抜けて、LEDをONにして休止へ戻ります。ボタン開放で、休止を抜け、LEDをOFFにして休止へ戻ります。

注: コード例はATmega4809 Xplained Pro (ATMEGA4809-XPRO)で開発され、GitHubで入手可能です。



GitHubでコード例を見てください。
貯蔵庫を閲覧するにはクリックしてください。

本書は一般の方々の便宜のため有志により作成されたもので、Microchip社とは無関係であることを御承知ください。しおりの[はじめに]での内容にご注意ください。

目次

序説	1
1. 関連デバイス	3
1.1. tinyAVR® 0系統	3
1.2. tinyAVR® 1系統	3
1.3. megaAVR® 0系統	3
2. 概要	4
3. LED点滅	4
4. 長/短釦押下	5
5. 釦押下での起き上がり	7
6. MCCでのGPIOの使い方	8
Microchipウェブサイト	10
製品変更通知サービス	10
お客様支援	10
Microchipデバイスコード保護機能	10
法的通知	10
商標	11
品質管理システム	11
世界的な販売とサービス	12

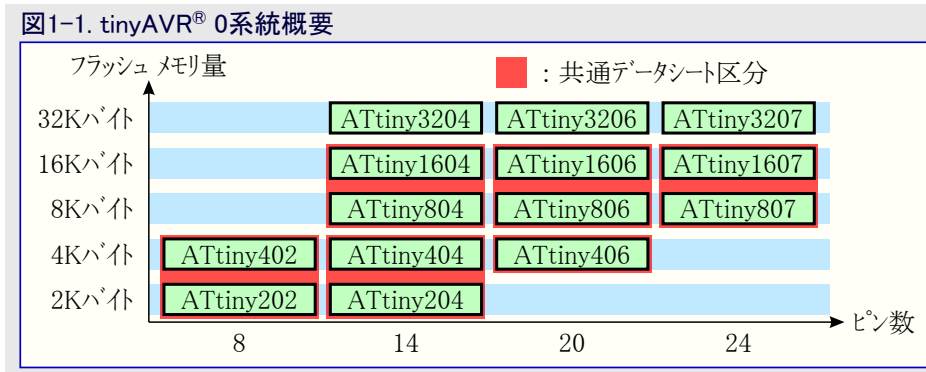
1. 関連デバイス

本章はこの資料に関連するデバイスを一覧にします。

1.1. tinyAVR[®] 0系統

下図はピン数の変種とメモリ量を展開してtinyAVR[®] 0系統デバイスを示します。

- これらのデバイスが完全にピンと機能が互換のため、垂直方向移植はコード変更なしで可能です。
- 左への水平方向移植はピン数、従って利用可能な機能を減らします。

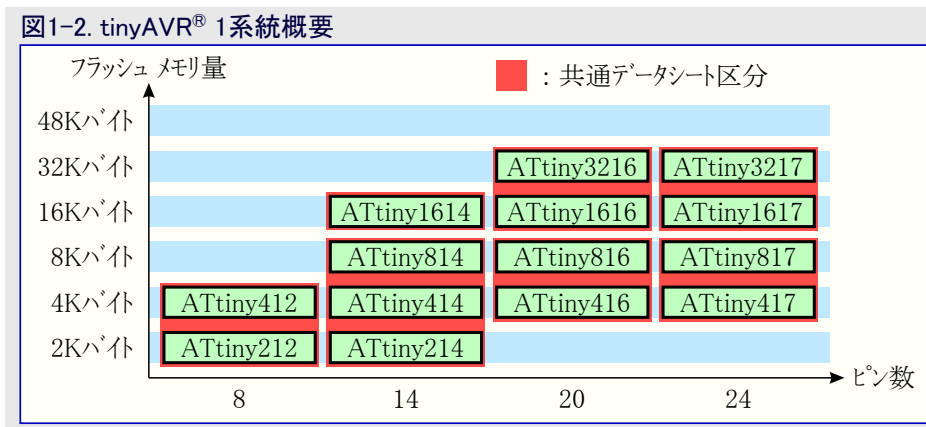


異なるフラッシュメモリ量を持つデバイスは一般的に異なるSRAMとEEPROMの量を持ちます。

1.2. tinyAVR[®] 1系統

下図はピン配置変種とメモリ量を展開してtinyAVR[®] 1系統デバイスを示します。

- これらのデバイスがピン互換で同じまたはより多くの機能を提供するため、垂直上方向移植はコード変更なしに可能です。下方向移植はより少ない利用可能ないくつかの周辺機能の実体のためにコード変更が必要かもしれません。
- 左への水平方向移植はピン数、従って利用可能な機能を減らします。

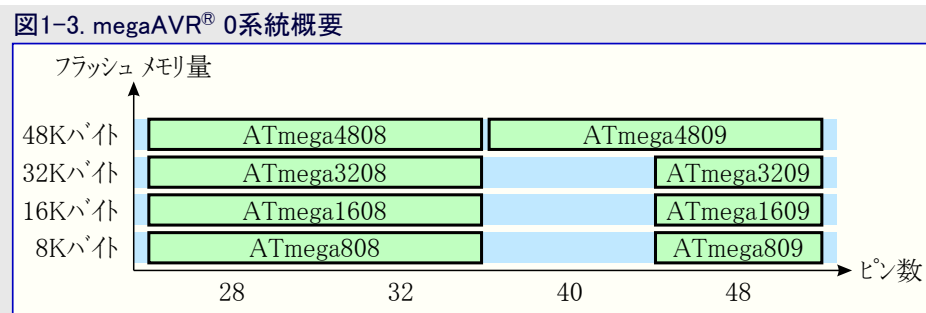


異なるフラッシュメモリ量を持つデバイスは一般的に異なるSRAMとEEPROMの量を持ちます。

1.3. megaAVR[®] 0系統

右図はピン配置変種とメモリ量を展開してmegaAVR[®] 0系統デバイスを示します。

- これらのデバイスが完全にピンと機能が互換のため、垂直方向移植はコード変更なしで可能です。
- 左への水平方向移植はピン数、従って利用可能な機能を減らします。



異なるフラッシュメモリ量を持つデバイスは一般的に異なるSRAMとEEPROMの量を持ちます。

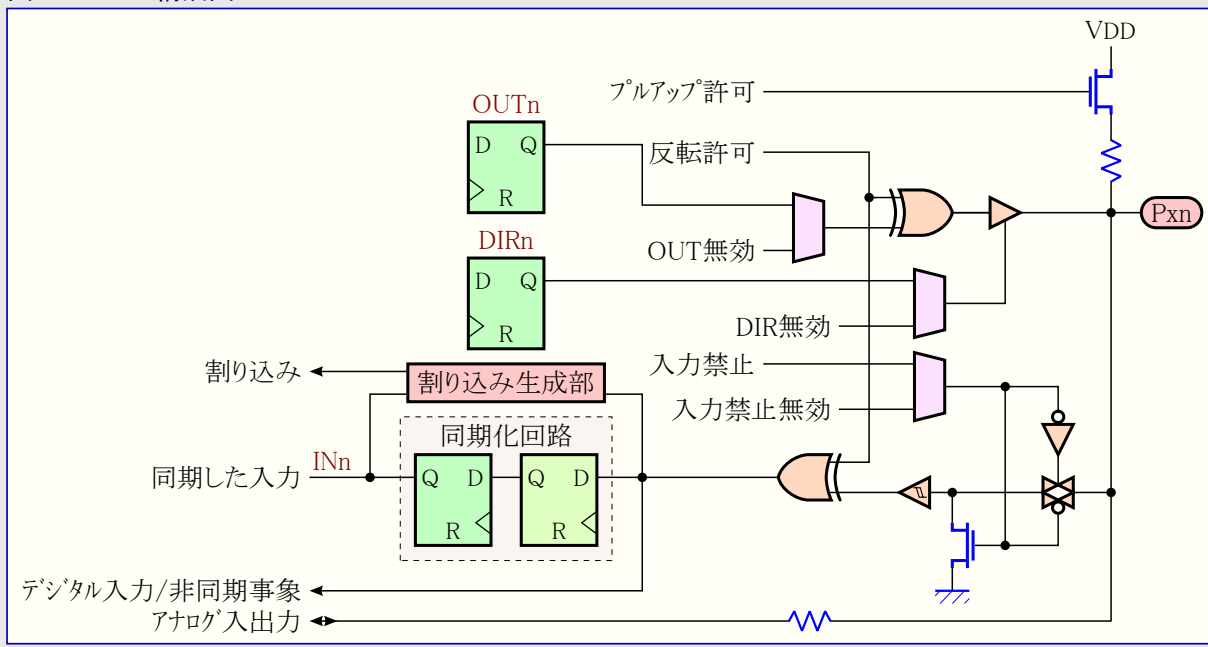
2. 概要

デバイスの入出力ピンはPORT周辺機能レジスタの実体によって制御されます。各ポートの実体は最大8つの入出力ピンを持ちます。ポートはPORTA、PORTB、PORTCなどと名付けられます。全てのピンの機能はピン毎に個別に構成設定できます。

最良の消費電力のため、未使用ピン、アナログ入力として使われるピン、出力として使われるピンの入力(緩衝部)を禁止してください。

デバッグに接続するのに使われるそれらのような特別なピンはそれら特殊機能によって必要とされるため、違うように構成設定されるかもしれません。

図2-1. PORT構成図



3. LED点滅

本章はLEDを継続的にON/OFF切り替えする方法を実演します。

この使用事例が非常に簡単とはいえ、現実の応用で広く使われ、更に、マイクロコントローラの非常に基本的な機能のいくつかを達成する方法を理解するのに役立ちます。これらの機能は次のとおりです。

- ・ピン方向を設定
- ・ピン出力値を設定

ピン方向を設定

ピンの方向はPORTx.DIRレジスタに格納され、例えば、PORTAのピンに対する方向はPORTA.DIRレジスタに格納されます。レジスタ内の各ビットは対応するピンの方向を制御し、故にPORTA.DIRのビット0は(PA0とも呼ばれる)PORTA0ピンを制御します。'1'のビット値に対して、ピンは出力として構成設定され、一方で'0'に対しては入力として構成設定されます。

殆どの場合、一度にビットを書くことだけがが必要です。この目的のため、各ビットに対して1つ、<avr/io.h>ヘッダファイルで定義される8つのビット遮蔽(PINn_bm)があります。これらのマクロはn位置で'1'のビット、残り'0'のビットを持つビット遮蔽を表します。例えば、PIN5_bmは0b00100000です。論理操作と共にこれらのビット遮蔽を使うと、対応するピンを書くだけでその他は無変化にされることを可能にします。

図3-1. データ方向レジスタビット5

ビット	7	6	5	4	3	2	1	0
	DIR7~0							
アクセス種別	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
リセット値	0	0	0	0	0	0	0	0

ピン_xを設定する('1'を書く)には以下を使ってください。

```
PORTx.DIR = PORTx.DIR | PINn_bm;
```

これは'1'との論理和(OR)操作が常に'1'の結果に終わり、一方で'0'との論理和(OR)が値を無変化のままにするからです(1|1=1, 1|0=1, 0|0=0)。

ピン_xを解除する('0'を書く)には以下を使ってください。

```
PORTx.DIR = PORTx.DIR & ~PINn_bm;
```

これは'0'との論理積(AND)操作が常に'0'の結果に終わり、一方で'1'との論理積(AND)が値を無変化のままにするからです($1 \& 1 = 1$ 、 $1 \& 0 = 0$ 、 $0 \& 0 = 0$)。

- 注: 1. 単一ビットだけを変更する同じ手法は前節で記述されるように他の全てのレジスタに使うことができます。
2. 複数のビット遮蔽を持つ論理操作は $PORTA.DIR = PORTA.DIR | PIN0_bm | PIN1_bm$ のような式で繋げることができます。
3. $PINn_bp$ はレジスタ内のビット位置を定義する類似マクロで、例えば、 $PIN2_bp$ は2の値を持ちます。ビット遮蔽マクロはこれらのマクロと移動操作を使って達成することができます。

ピンの出力値を設定

出力値は $PORTx.OUT$ レジスタによって制御され、'0'はピンがGNDに引かれ、'1'はVDDに引かれることを意味します。

注: これら'0'と'1'の値の意味は $PORTx.PINnCTRL$ レジスタを使って反転することができます。この機能を理解するにはデータシートの $PORTx.PINnCTRL$ レジスタ項をご覧ください。

図3-2. 出力値レジスタビット5

ビット	7	6	5	4	3	2	1	0
	OUT7~0							
アクセス種別	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
リセット値	0	0	0	0	0	0	0	0

前で記述されたように、単一ビットだけを変更する同じ手法は $PORTx.OUT$ レジスタを書くのにも適用することができます。

以下のコードは500ms毎にPB5ピンを交互切り替えします。ATmega4809 Xplained Pro基板のそのピンにLEDが取り付けられています。

```
#define F_CPU 3333333
#include <avr/io.h>
#include <avr/delay.h>

int main(void)
{
    PORTB.DIR |= PIN5_bm;

    while (1)
    {
        PORTB.OUT |= PIN5_bm;
        _delay_ms(500);
        PORTB.OUT &= ~PIN5_bm;
        _delay_ms(500);
    }
}
```

- 注: 1. 遅延関数を使うには $avr/delay.h$ をインクルードする前に F_CPU が定義されなければなりません。 F_CPU はCPU周波数と一致しなければなりません。
2. ピンを交互切り替えするのに $PORTx.OUTTGL$ レジスタを使うこともできます。
3. 最適化のため、方向と出力値を設定するのに $PORTx.OUTSET/PORTx.OUTCLR$ と $PORTx.DIRSET/PORTx.DIRCLR$ のレジスタを使うこともできます。これらの利点は読み-変更-書きの操作の代わりに書き込み操作だけが使われることです。

4. 長/短釘押下

本章は釘押下が長いかわりのどちらかを検出して違う反応をする方法を説明します。

GPIOインターフェースは或る決定をするために外部デジタル信号を検知するのに使うことができます。この点について、応用範囲は非常に広く、故にAVR制御器は使用者が必要かもしれない多くの使用事例に合うように沢山の構成設定をすることができます。本章では釘の押下に集中します。

最初に、データ方向解除($PORTB.DIRCLR$)レジスタのビット2に'1'を書くことによってPB2が入力として構成設定されます。

図4-1. データ方向解除レジスタビット2

ビット	7	6	5	4	3	2	1	0
	DIRCLR7~0							
アクセス種別	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
リセット値	0	0	0	0	0	0	0	0

```
PORTB.DIRCLR = PIN2_bm;
```

GPIOピンとGND間に接続された釦に対して正しく動くように、GPIOピンとVDD間にプルアップ抵抗が接続されなければなりません。この構成設定は釦が押されない時にピンが論理'1'の既定値を読むことを保証します。2番ピン制御(PORTB.PIN2CTRL)レジスタから内部プルアップ抵抗を活性(有効)にすることができます。

図4-2. PIN2制御レジスタ 内部プルアップ抵抗

ビット	7	6	5	4	3	2	1	0
	INVEN				PULLUPEN		ISC2~0	
アクセス種別	R/W	R	R	R	R/W	R/W	R/W	R/W
リセット値	0	0	0	0	0	0	0	0

```
PORTB.PIN2CTRL = PORT_PULLUPEN_bm;
```

より良いユーザー体験のため、PB5に接続されたLEDが押下の形式に基づいて異なる周波数で点滅します。従って、PB5はデータ方向設定(PORTB.DIRSET)レジスタのビット5に'1'を書くことによって出力として構成設定されます。

```
PORTB.DIRSET = PIN5_bm;
```

その後、主繰り返しの各繰り返しで、入力値(PORTB.IN)レジスタのビット2が調べられます。この値が論理'0'なら、プログラムはその値が論理'1'に戻るまで待ちます。これを行うことにより、プログラムは釦での押下と解放の活動が実行された時を感知します。待機中、計数器が数ms毎に増されます。

計数器の値が或る閾値を通過する場合、プログラムは長押しがあったと判断します。閾値を通過する前に釦が開放された場合、プログラムは短押しがあったと判断します。押下形式に応じて、LEDは違う周波数で点滅します。

図4-2. 入力値レジスタビット2

ビット	7	6	5	4	3	2	1	0
	IN7~0							
アクセス種別	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
リセット値	0	0	0	0	0	0	0	0

```
if (~PORTB.IN & PIN2_bm) /* PB2がGNDに引かれているか調査 */
{
    while (~PORTB.IN & PIN2_bm) /* PB2がVDDに引かれるまで待ち */
    {
        _delay_ms(STEP_DELAY);
        counter++;
        if (counter >= THRESHOLD)
        {
            LED_blink(LONG_DELAY);
            while (~PORTB.IN & PIN2_bm) /* PB2がVDDに引かれるまで待ち */
            {
                ;
            }
            break;
        }
    }
    if (counter < THRESHOLD)
    {
        LED_blink(SHORT_DELAY);
    }
    counter = 0;
}
```

注: 遅延マクロは<util/delay.h>ヘッダ ファイルで見つけることができます。

LED_blink関数はパラメータとして2分割されたmsでの点滅周期を取ります。_delay_msマクロがコンパイル段階で既知のパラメータを取るため、この関数は"inline"宣言されなければなりません。

```

inline void LED_blink(uint32_t time_ms)
{
    for(uint8_t i = 0; i < NUMBER_OF_BLINKS ; i++)
    {
        PORTB. OUT |= PIN5_bm;
        _delay_ms(time_ms);
        PORTB. OUT &= ~PIN5_bm;
        _delay_ms(time_ms);
    }
}

```



GitHubでコード例を見てください。
 貯蔵庫を閲覧するにはクリックしてください。

5. 釦押下での起き上がり

この例は割り込みと休止動作形態の使い方を実演します。釦押下で休止を抜け出し、LEDをONに切り替えて休止へ戻ります。釦開放で、休止を抜け出し、LEDをOFFに切り替えて休止へ戻ります。釦が押されている間、LEDはONですが、小型電子機器を休止動作にすることができます。

ピンは(上昇端と呼ばれる)'0'から'1'への遷移と(下降端と呼ばれる)'1'から'0'への遷移を検出することができます。割り込みは1つまたは両方の遷移で起動することができます。これはPORTx.PINnCTRLレジスタのISCビット領域を使って構成設定されます。

図5-1. PINn制御レジスタ 割り込み検知制御(ISC)

ビット	7	6	5	4	3	2	1	0
	INVEN				PULLUPEN		ISC2~0	
アクセス種別	R/W	R	R	R	R/W	R/W	R/W	R/W
リセット値	0	0	0	0	0	0	0	0

これらの割り込みは全ての休止動作形態から直ちにデバイスを起き上がらせます。この例では、釦押下/開放で、マイクロコントローラが休止から起き上がってLEDをON/OFFに切り替え、その後に再び休止します。このように、LEDは釦が押されている間だけONに留まります。

最初に、釦が取り付けられているPB2ピンは入力として構成設定され、その割り込みが活性(有効)にされます。

```

PORTB. DIR &= ~ PIN2_bm;
PORTB. PIN2CTRL |= PORT_PULLUPEN_bm | PORT_ISC_BOTHEDGES_gc;

```

注: 押下時、釦は回路を接地へ閉じ、ピンは'0'を読み、従って、プルアップ抵抗を使って、釦が押されない時は必ずVDDに接続されなければなりません。プルアップ抵抗はPORTx.PINnCTRLレジスタのPULLUPENビット領域を使って活性(有効)にされます。

次に、休止動作形態は<avr/sleep.h>ヘッダで定義されるマクロを使って選ばれます。休止の動作形態はアイドル、スタンバイ、パワーダウンです。この例では最も深い休止動作が使われます。

```

set_sleep_mode(SLEEP_MODE_PWR_DOWN);

```

これは使われるべき休止動作を選ぶだけです。以下のマクロがマイクロコントローラを休止に行かせます。

```

sleep_mode();

```

注: sleep_mode()マクロはCPUにSLEEP命令を送る前に休止も許可し、マイクロコントローラ起き上がり時に休止を禁止します。

休止しようとする時には全体割り込みが許可されなければならず、そうしなければ決して起き上がられません。<avr/interrupt.h>ヘッダで定義されるsei()マクロの使用が全体割り込みを許可します。この例では、これが初期化コードで行われます。

時々、割り込みなしで実行されることが必要な命令の塊部分があります。それらは非分断(Atomic)部と呼ばれます。これはその部分の始めて割り込みを禁止してその部分の最後で再許可することによって達成されます。これについては、<avr/atomic.h>内のATOMI C_BLOCKマクロが使われます。

最後に、割り込み処理ルーチン(ISR)の内側で、遷移(上昇端または下降端)発生を示すためにフラグが設定されます。これはISRを短く保つための良い習慣で、遷移形式調査とLED交互切り替えは主流れ作業コードで処理されるからです。


```
ISR(PORTB_PORT_vect)
{
    if (PB2_INTERRUPT)
    {
        pb2Ioc = 1;
        PB2_CLEAR_INTERRUPT_FLAG;
    }
}
```

同じ割り込みがポート内の全てのピンに対して使われるため、ISRコードはどのピンが割り込みを起動したかを調べなければなりません。これはPORTx.**INTFLAGS**でそのフラグを調べることによって行われます。この例では、PB2ピンが割り込みを起動したかどうかを調べるのに以下のマクロが使われます。

```
#define PB2_INTERRUPT PORTB.INTFLAGS & PIN2_bm
```

割り込み要求フラグの解除(0)はPORTx.**INTFLAGS**でその位置に'1'を書くことによって行われます。

```
#define PB2_CLEAR_INTERRUPT_FLAG PORTB.INTFLAGS &= PIN2_bm
```

注: このフラグが解除(0)されない場合、割り込みが起動し続け、故にISRを抜け出す前にフラグが常に解除(0)されることが重要です。これがISR負荷を事実上過負荷にするため、割り込み要求フラグの解除(0)に頼らないどの算法も強く制止されます。結果としてISRの責任は割り込みを処理してフラグが解除(0)されるまで始動し続けることです。これはソフトウェア設計での単一責任に違反し、バグを避けるために極度の注意が払われなければなりません。

主流作業コードはそれが上昇端または下降端のどちらだったかを判断するために割り込みの10ms後にピン値を調べます。ピンがLow('0'値)なら、それは下降端でなければならず、さもなければそれは上昇端です。10ms遅延は跳ね返り抑制(信号安定待機)法です。

```
#define PB2_LOW !(PORTB.IN & PIN2_bm)
```



GitHubでコード例を見てください。
貯蔵庫を閲覧するにはクリックしてください。

6. MCCでのGPIOの使い方

MPLAB®コード構成設定部(MCC:MPLAB Code Configurator)は画像使用者インターフェース(GUI)で行われる設定と選択に基づき、PIC®とAVR®のマイクロコントローラの周辺機能を制御して駆動するためのドライバを生成するMPLAB® XとMPLAB® Xpress IDE用の使い易いプラグインツールです。生成されたドライバはどの応用プログラムでも使うことができます。

- MCCについてのより多くの詳細に関しては[MCC使用者の手引き](#)をご覧ください。
- [MPLAB Xpress](#)はブラウザで使うことができるオンラインIDEです。

MCCではGPIOピンはPin Manager: **Grid View**(ピン管理部: 格子表示部)ウィンドウで構成設定することができます。

図6-1. Pin Manager: Grid View (ピン管理部: 格子表示部) ウィンドウ

Pin Manager: Grid View			Pin No: 44 45 46 47 48 1 2 3 4 5 6 7 8 9 10 11 12 13 16 17 18 19 20																						
Package: QFP48			Port A ▼							Port B ▼							Port C ▼								
Module	Function	Direction	0	1	2	3	4	5	6	7	0	1	2	3	4	5	0	1	2	3	4	5	6	7	0
CLKCTRL ▼	CLKI	input	<input checked="" type="checkbox"/>								<input checked="" type="checkbox"/>														
	CLKO	output																							
	TOSC1	input																							
	TOSC2	input																							
Pin Module ▼	GPIO	input	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
	GPIO	output	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
RSTCTRL	RESET	input																							

錠の1つのクリックはクリックした列に依存して対応するピンを入力または出力のどちらかとして選びます。ピンは同時に入力及び出力として働くことはできません。入力と出力の間で走行時切り替えが必要とされる場合、これは手動で実装されなければなりません。

一旦選ばれると、ピンはPin Module(ピン単位部)ウィンドウにも現れます。

図6-2. Pin Module (ピン単位部) ウィンドウ

Pin Name ▲	Module	Function	Custom Name	OUTPUT	START HIGH	INVEN	PULLUPEN	ISC
PC0	Pin Module	GPIO	LED	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Interrupt ... ▼

この資料で既に検討されたものの殆どをここで構成設定することができる様々な任意選択があります。出力チェック枠が非チェックなら、そのピンは入力です。

MCCはピン使い用マクロを生成します。マクロの接頭辞はCustom Name(独自名)領域で与えられるピン名です。これは例えば、1つのピンがLEDを制御し、そのマクロを実際のピン名の代わりにLEDを接頭辞にすることができ、抽象化に役立ちます。

図6-3. ピン使い用マクロ

```
LED_get_level()
LED_set_dir(port_dir dir)
LED_set_inverted(const _Bool inverted)
LED_set_isc(const PORT_ISC_t isc)
LED_set_level(const _Bool level)
LED_set_pull_mode(port_pull_mode pull_mode)
LED_toggle_level()
```

Microchipウェブ サイト

Microchipは<http://www.microchip.com/>で当社のウェブ サイト経由でのオンライン支援を提供します。このウェブ サイトはお客様がファイルや情報を容易に利用可能にするのに使われます。利用可能な情報のいくつかは以下を含みます。

- **製品支援** – データシートと障害情報、応用記述と試供プログラム、設計資源、使用者の手引きとハードウェア支援資料、最新ソフトウェア配布と保管されたソフトウェア
- **一般的な技術支援** – 良くある質問(FAQ)、技術支援要求、オンライン検討グループ、Microchip設計協力課程会員一覧
- **Microchipの事業** – 製品選択器と注文の手引き、最新Microchip報道発表、セミナーとイベントの一覧、Microchip営業所の一覧、代理店と代表する工場

製品変更通知サービス

Microchipの製品変更通知サービスはMicrochip製品を最新に保つのに役立ちます。加入者は指定した製品系統や興味のある開発ツールに関連する変更、更新、改訂、障害情報がある場合に必ず電子メール通知を受け取ります。

登録するには<http://www.microchip.com/pcn>へ行って登録指示に従ってください。

お客様支援

Microchip製品の使用者は以下のいくつかのチャネルを通して支援を受け取ることができます。

- 代理店または販売会社
- 最寄りの営業所
- 組み込み解決技術者(ESE:Embedded Solutions Engineer)
- 技術支援

お客様は支援に関してこれらの代理店、販売会社、またはESEに連絡を取るべきです。最寄りの営業所もお客様の手助けに利用できます。営業所と位置の一覧はこの資料の後ろに含まれます。

技術支援は<http://www.microchip.com/support>でのウェブ サイトを通して利用できます。

Microchipデバイスコード保護機能

Microchipデバイスでの以下のコード保護機能の詳細に注意してください。

- Microchip製品はそれら特定のMicrochipデータシートに含まれる仕様に合致します。
- Microchipは意図した方法と通常条件下で使われる時に、その製品系統が今日の市場でその種類の最も安全な系統の1つであると考えます。
- コード保護機能を破るのに使われる不正でおそらく違法な方法があります。当社の知る限りこれらの方法の全てはMicrochipのデータシートに含まれた動作仕様外の方法でMicrochip製品を使うことが必要です。おそらく、それを行う人は知的財産の窃盗に関与しています。
- Microchipはそれらのコードの完全性について心配されているお客様と共に働きたいと思います。
- Microchipや他のどの半導体製造業者もそれらのコードの安全を保証することはできません。コード保護は当社が製品を”破ることができない”として保証すると言うことを意味しません。

コード保護は常に進化しています。Microchipは当社製品のコード保護機能を継続的に改善することを約束します。Microchipのコード保護機能を破る試みはデジタル ミレニアム著作権法に違反するかもしれません。そのような行為があなたのソフトウェアや他の著作物に不正なアクセスを許す場合、その法律下の救済のために訴権を持つかもしれません。

法的通知

デバイス応用などに関してこの刊行物に含まれる情報は皆さまの便宜のためにだけ提供され、更新によって取り換えられるかもしれません。皆さまの応用が皆さまの仕様に合致するのを保証するのは皆さまの責任です。Microchipはその条件、品質、性能、商品性、目的適合性を含め、明示的にも黙示的にもその情報に関連して書面または表記された書面または黙示の如何なる表明や保証も**しません**。Microchipはこの情報とそれの使用から生じる全責任を否認します。生命維持や安全応用でのMicrochipデバイスの使用は完全に購入者の危険性で、購入者はそのような使用に起因する全ての損害、請求、訴訟、費用からMicrochipを擁護し、補償し、免責にすることに同意します。他に言及されない限り、Microchipのどの知的財産権下でも暗黙的または違う方法で許認可は譲渡されません。

商標

Microchipの名前とロゴ、Mmicrochipロゴ、Adaptec、AnyRate、AVR、AVRロゴ、AVR Freaks、BesTime、BitCloud、chipKIT、chipKITロゴ、CryptoMemory、CryptoRF、dsPIC、FlashFlex、flexPWR、HELDO、IGLOO、JukeBlox、KeeLoq、Kleer、LANCheck、LinkMD、maXStylus、maXTouch、MediaLB、megaAVR、Microsemi、Microsemiロゴ、MOST、MOSTロゴ、MPLAB、OptoLyzer、PackeTime、PIC、picoPower、PICSTART、PIC32ロゴ、PolarFire、Prochip Designer、QTouch、SAM-BA、SenGenuity、SpyNIC、SST、SSTロゴ、SuperFlash、Symmetricom、SyncServer、Tachyon、TempTracker、TimeSource、tinyAVR、UNI/O、Vectron、XMEGAは米国と他の国に於けるMicrochip Technology Incorporatedの登録商標です。

APT、ClockWorks、The Embedded Control Solutions Company、EtherSynch、FlashTec、Hyper Speed Control、HyperLight Load、IntelliMOS、Liberio、motorBench、mTouch、Powermite 3、Precision Edge、ProASIC、ProASIC Plus、ProASIC Plusロゴ、Quiet-Wire、SmartFusion、SyncWorld、Temux、TimeCesium、TimeHub、TimePictra、TimeProvider、Vite、WinPath、ZLは米国に於けるMicrochip Technology Incorporatedの登録商標です。

Adjacent Key Suppression、AKS、Analog-for-the-Digital Age、Any Capacitor、AnyIn、AnyOut、BlueSky、BodyCom、CodeGuard、CryptoAuthentication、CryptoCompanion、CryptoController、dsPICDEM、dsPICDEM.net、Dynamic Average Matching、DAM、ECAN、EtherGREEN、In-Circuit Serial Programming、ICSP、INICnet、Inter-Chip Connectivity、JitterBlocker、KleerNet、KleerNetロゴ、memBrain、Mindi、MiWi、MPASM、MPF、MPLAB Certifiedロゴ、MPLAB、MPLINK、MultiTRAK、NetDetach、Omniscient Code Generation、PICDEM、PICDEM.net、PICKIT、PICKITail、PowerSmart、PureSilicon、QMatrix、REALICE、Ripple Blocker、SAM-ICE、Serial Quad I/O、SMART-I.S.、SQI、SuperSwitcher、SuperSwitcher II、Total Endurance、TSHARC、USBCheck、VariSense、View Sense、WiperLock、Wireless DNA、ZENAは米国と他の国に於けるMicrochip Technology Incorporatedの商標です。

SQTPは米国に於けるMicrochip Technology Incorporatedの役務標章です。

Adaptecロゴ、Frequency on Demand、Silicon Storage Technology、Symmcomは他の国に於けるMicrochip Technology Inc.の登録商標です。

GestICは他の国に於けるMicrochip Technology Inc.の子会社であるMicrochip Technology Germany II GmbH & Co. KGの登録商標です。

ここで言及した以外の全ての商標はそれら各々の会社の所有物です。

© 2019年、Microchip Technology Incorporated、米国印刷、不許複製

品質管理システム

Microchipの品質管理システムに関する情報については<http://www.microchip.com/quality>を訪ねてください。

日本語© HERO 2019.

本技術概説はMicrochipのTB3229技術概説(DS90003229A-2019年10月)の翻訳日本語版です。日本語では不自然となる重複する形容表現は省略されている場合があります。日本語では難解となる表現は大幅に意識されている部分もあります。必要に応じて一部加筆されています。頁割の変更により、原本より頁数が少なくなっています。

必要と思われる部分には()内に英語表記や略称などを残す形で表記しています。

青字の部分はリンクとなっています。一般的に赤字の0,1は論理0,1を表します。その他の赤字は重要な部分を表します。

世界的な販売とサービス

米国	亜細亜/太平洋	亜細亜/太平洋	欧州
本社 2355 West Chandler Blvd. Chandler, AZ 85224-6199 Tel: 480-792-7200 Fax: 480-792-7277 技術支援: http://www.microchip.com/support ウェブアドレス: http://www.microchip.com	オーストラリア - シドニー Tel: 61-2-9868-6733 中国 - 北京 Tel: 86-10-8569-7000 中国 - 成都 Tel: 86-28-8665-5511 中国 - 重慶 Tel: 86-23-8980-9588 中国 - 東莞 Tel: 86-769-8702-9880 中国 - 広州 Tel: 86-20-8755-8029 中国 - 杭州 Tel: 86-571-8792-8115 中国 - 香港特别行政区 Tel: 852-2943-5100 中国 - 南京 Tel: 86-25-8473-2460 中国 - 青島 Tel: 86-532-8502-7355 中国 - 上海 Tel: 86-21-3326-8000 中国 - 瀋陽 Tel: 86-24-2334-2829 中国 - 深圳 Tel: 86-755-8864-2200 中国 - 蘇州 Tel: 86-186-6233-1526 中国 - 武漢 Tel: 86-27-5980-5300 中国 - 西安 Tel: 86-29-8833-7252 中国 - 廈門 Tel: 86-592-2388138 中国 - 珠海 Tel: 86-756-3210040	インド - ハンガロール Tel: 91-80-3090-4444 インド - ニューデリー Tel: 91-11-4160-8631 インド - フネー Tel: 91-20-4121-0141 日本 - 大阪 Tel: 81-6-6152-7160 日本 - 東京 Tel: 81-3-6880-3770 韓国 - 大邱 Tel: 82-53-744-4301 韓国 - ソウル Tel: 82-2-554-7200 マレーシア - クアラルンプール Tel: 60-3-7651-7906 マレーシア - ペナン Tel: 60-4-227-8870 フィリピン - マニラ Tel: 63-2-634-9065 シンガポール Tel: 65-6334-8870 台湾 - 新竹 Tel: 886-3-577-8366 台湾 - 高雄 Tel: 886-7-213-7830 台湾 - 台北 Tel: 886-2-2508-8600 タイ - バンコク Tel: 66-2-694-1351 ベトナム - ホーチミン Tel: 84-28-5448-2100	オーストリア - ウェルス Tel: 43-7242-2244-39 Fax: 43-7242-2244-393 デンマーク - コペンハーゲン Tel: 45-4450-2828 Fax: 45-4485-2829 フィンランド - エスポー Tel: 358-9-4520-820 フランス - パリ Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79 ドイツ - ガルピング Tel: 49-8931-9700 ドイツ - ハーン Tel: 49-2129-3766400 ドイツ - ハイムブロン Tel: 49-7131-72400 ドイツ - カールスルーエ Tel: 49-721-625370 ドイツ - ミュンヘン Tel: 49-89-627-144-0 Fax: 49-89-627-144-44 ドイツ - ローゼンハイム Tel: 49-8031-354-560 イスラエル - ラーナナ Tel: 972-9-744-7705 イタリア - ミラノ Tel: 39-0331-742611 Fax: 39-0331-466781 イタリア - ハドバ Tel: 39-049-7625286 オランダ - デルネン Tel: 31-416-690399 Fax: 31-416-690340 ノルウェー - トロンハイム Tel: 47-72884388 ポーランド - ワルシャワ Tel: 48-22-3325737 ルーマニア - ブカレスト Tel: 40-21-407-87-50 スペイン - マドリッド Tel: 34-91-708-08-90 Fax: 34-91-708-08-91 スウェーデン - イェテボリ Tel: 46-31-704-60-40 スウェーデン - ストックホルム Tel: 46-8-5090-4654 イギリス - ウォーキングム Tel: 44-118-921-5800 Fax: 44-118-921-5820
アトランタ Duluth, GA Tel: 678-957-9614 Fax: 678-957-1455 オースチン TX Tel: 512-257-3370 ホーストン Westborough, MA Tel: 774-760-0087 Fax: 774-760-0088 シカゴ Itasca, IL Tel: 630-285-0071 Fax: 630-285-0075 ダラス Addison, TX Tel: 972-818-7423 Fax: 972-818-2924 デトロイト Novi, MI Tel: 248-848-4000 ヒューストン TX Tel: 281-894-5983 インディアナポリス Noblesville, IN Tel: 317-773-8323 Fax: 317-773-5453 Tel: 317-536-2380 ロサンゼルス Mission Viejo, CA Tel: 949-462-9523 Fax: 949-462-9608 Tel: 951-273-7800 ローリー NC Tel: 919-844-7510 ニューヨーク NY Tel: 631-435-6000 サンホセ CA Tel: 408-735-9110 Tel: 408-436-4270 カナダ - トロント Tel: 905-695-1980 Fax: 905-695-2078			