
AVR® デバイスでの配列キーパッドの使い方

要点

- 配列キーパッド動作の基本的な概要と例
- 他の応用への容易な実装
- 高度な低電力実装例
 - 完全な割り込み駆動動作
 - パワーダウン休止動作
 - 電池給電応用に適合
- ソフトウェア鉤跳ね返り制御
- 簡単な暗証番号検査実装
- LED暗証番号確認表示器

序説

著者: Amund Aune, Microchip Technology Inc.

この応用記述はtinyAVR®とmegaAVR®のデバイスで一般的なキーパッド応用がどう実装され得るかを示します。配列キーパッドの動作の概念的な概要と2つの実演応用が提示されます。1つの実演がキーパッドの簡単な実装を示す一方で、2つ目の実演は少ない電力しか使わずに応用をもっと効率的にするのにより高度な機能を使います。

この応用記述の例はより小さなまたはより大きな配列キーパッドを接続するためにおそらく容易に変更され、別の応用への実装が容易です。この応用はアクセス制御キーパッド、キーボード、遠隔制御のような配列キーパッドを使う全ての実装に使うことができますでしょう。

コード例はAtmel STARTを通して入手可能です。

- AVRデバイスでの配列キーパッドの使い方 - 基本 (Using Matrix Keypad with AVR Devices - Basic)
 - https://start.atmel.com/#example/Atmel%3AApplication_AVR_Examples%3A1.0.0%3A%3AApplication%3AUsing_Matrix_Keypad_with_AVR_Devices_-_Basic%3A
- AVRデバイスでの配列キーパッドの使い方 - 高度 (Using Matrix Keypad with AVR Devices - Advanced)
 - https://start.atmel.com/#example/Atmel%3AApplication_AVR_Examples%3A1.0.0%3A%3AApplication%3AUsing_Matrix_Keypad_with_AVR_Devices_-_Advanced%3A

コード例はGitHubを通しても入手可能です。



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。

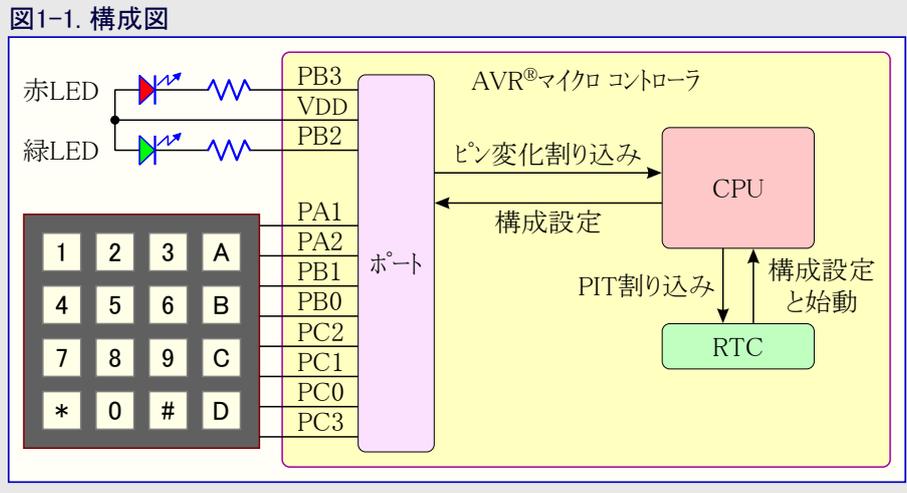
本書は一般の方々の便宜のため有志により作成されたもので、Microchip社とは無関係であることを御承知ください。しおりの[はじめに]での内容にご注意ください。

目次

要点	1
序説	1
1. 構成図	3
2. 動作の原理	3
3. 実演操作	4
3.1. ハードウェア事前要件	5
3.2. ソフトウェア事前要件	5
3.3. 例走行	5
4. ソースコード概要	5
4.1. 基本的な動作	5
4.2. 高度な動作	7
5. 消費電力	9
5.1. 基本的な動作	9
5.2. 高度な動作	10
5.3. 電流データ作図	10
6. Atmel STARTからのコード例取得	12
7. GitHubからのコード例取得	12
8. 改訂履歴	12
Microchipウェブ サイト	13
製品変更通知サービス	13
お客様支援	13
Microchipデバイス コード保護機能	13
法的通知	13
商標	14
品質管理システム	14
世界的な販売とサービス	15

1. 構成図

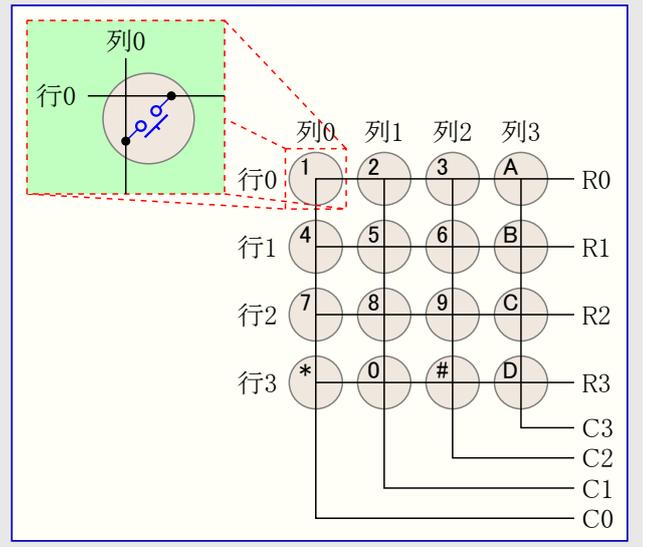
下の構成図はATtiny1627 Curiosity Nano開発基板上で4×4キーパッドを使う高度な応用例の概要を示します。これは応用がtinyAVR[®] 2系デバイスの周辺機能とCPUとどう互いに作用し合うかを示します。キーパッドとLEDはおそらく汎用入出力(GPIO:General Purpose Input/Output)ピンのどれかに接続されるでしょうが、この例では接続を簡単化するため、それらがATtiny1627 Curiosity Nano上の列に物理的に接続されます。



2. 動作の原理

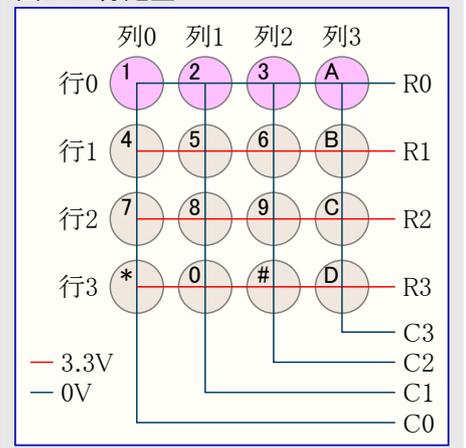
配列キーパッドはマイクロコントローラによって読まれて解釈され得る対応する線を持つ釘の格子から成ります。右図で示されるように、マイクロコントローラによって必要とされるピン数は行毎と列毎に1つの線が必要とされる釘格子の行と列の数によって決められます。釘が押されると、対応する行と列間の接続が作られます。この接続がキーパッドにマイクロコントローラによって解釈されることを許します。

図2-1. 配列キーパッド回路図



配列キーパッドの走査には多数の方法があります。例応用で使われる簡単な方法は一度に1つの完全な軸(即ち、行または列)の走査です。これは軸の1つに既知の電圧水準を出力するように設定してどの線がその電圧に引かれたかを見つけることによって行われます。図2-2は列出力0Vで行がプルアップ抵抗を通して3.3Vに接続される場合を示します。列への接続のため、行0が0Vに引き下げられることが示され、従って、押された釘が行0であると推測され得ます。

図2-2. 行走査



最初の軸が走査された後、他の軸を読むために設定がひっくり返されなければなりません。これは図2-3.で示され、ここでは行への接続のために列1が0Vに引き下げられることが示され、従って、押された釦が列1であると推測され得ます。

両軸が走査されると、格子内で押された釦の座標が知られ、釦押下が記録されて反応されます。この例では行0と列1が押された釦を含むことが知られ、釦2が押されたと判断されます。

この方法は簡単ですが、これでの問題は同時に1つの釦押下だけしか処理できないことです。図2-4.はそれぞれの軸がLowに引かれる時にどの線がLowに引かれるかを示します。図で示されるように、行1と2、列1と2が押され、実際に2つの釦だけが押されている間、4つの釦が記録されていることに至ります。この影響は虚像発生と呼ばれます。

図2-3. 列走査

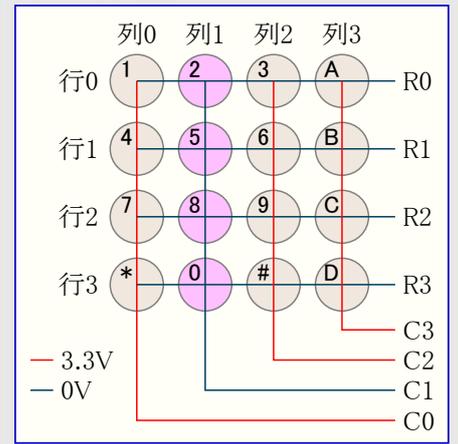
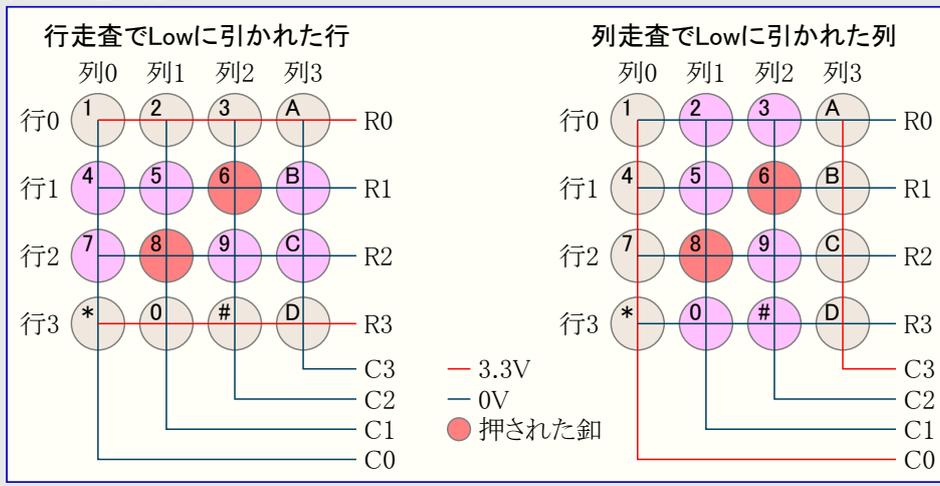
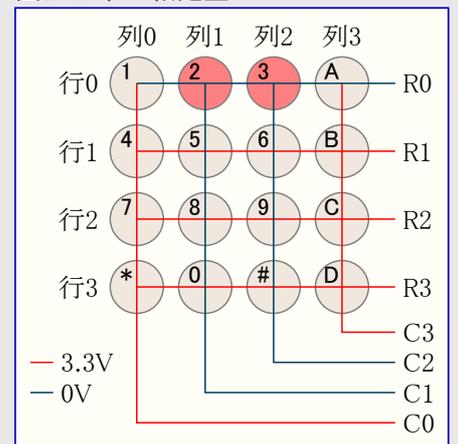


図2-4. 虚像発生



別の方法は一度に1つの線で軸をLowに引き、交差を見つけるために毎回ごとに他の軸で全てのピンを読むことです。これは右図で示され、ここで行0がLowに駆動され、釦2と3が押されていることによって列1と列2がLowに引かれます。この方法の利点は同時に2つの釦押下を読む能力である一方で、欠点は応用に複雑さを加え得ることです。

図2-5. 単一軸走査



3. 実演操作

これらの実演応用では、暗証番号がどう読まれて正当性が検査されるかを示すため、キーパッドがATtiny1627 Curiosity Nano開発基板に接続されます。PINコードはキーパッド上で英数釦を押すことによって書かれ、アスタリスク(*)を押すことによってリセットされ、シャープ(#)を押すことによって正当性が検査されます。PINが有効の時に緑LEDが点滅し、コードが不正な場合に赤LEDが点滅します。赤LEDは最大文字数(20)に達した場合にも点滅します。

2つの実演が提示され、その1つは基本的なキーパッド例で、可能な最も簡単な方法で上の機能を実装します。これはキーパッドがどう動いてそれでどう接続し得るかの理解を生じるために提示されます。他の実演は高度なキーパッド例で、様々な方法で応用の効率性を改善するためにtinyAVR®デバイスのもとで高度な機能を実装します。

3.1. ハードウェア事前要件

- Microchip ATtiny1627 Curiosity Nano評価基板
 - <https://www.microchip.com/developmenttools/ProductDetails/DM080104>
- 4×4配列キーパッド
- 2つのLED
- 2つの抵抗器
- マイクロUSBケーブル (A型/マイクロB)

3.2. ソフトウェア事前要件

- Atmel Studio 7 (7.0.2397またはそれ以降版)
- Atmel Studio ATtiny_DFP 1.4.308またはそれ以降版

3.3. 例走行

- 以下のようにキーパッドをATtiny1627 Curiosity Nanoに接続してください。
 - 行0 : PB0
 - 行1 : PB1
 - 行2 : PA2
 - 行3 : PA1
 - 列0 : PC3
 - 列1 : PC0
 - 列2 : PC1
 - 列3 : PC2
- 赤LEDをPB3、緑LEDをPB2に接続してください。
- USBケーブルを使ってATtiny1627 Curiosity Nanoをコンピュータに接続してください。
- 「6. Atmel STARTからのソースコード取得」章で説明されるように応用をダウンロードしてそれをATtiny1627 Curiosity Nanoに書き込んでください。
- 暗証番号としてPINコード”123ABC”が書かれています。緑LED点滅を観察するため、このコードに続けてシャープ(#)を押してみてください。赤LED点滅を観察するため、違うコードで試みてください。

4. ソースコード概要

4.1. 基本的な動作

本項は最低限の周辺機能と機能が使われた基本的なキーパッド応用を提示します。これはAVR®マイクロコントローラとキーパッドを接続する基本を示します。

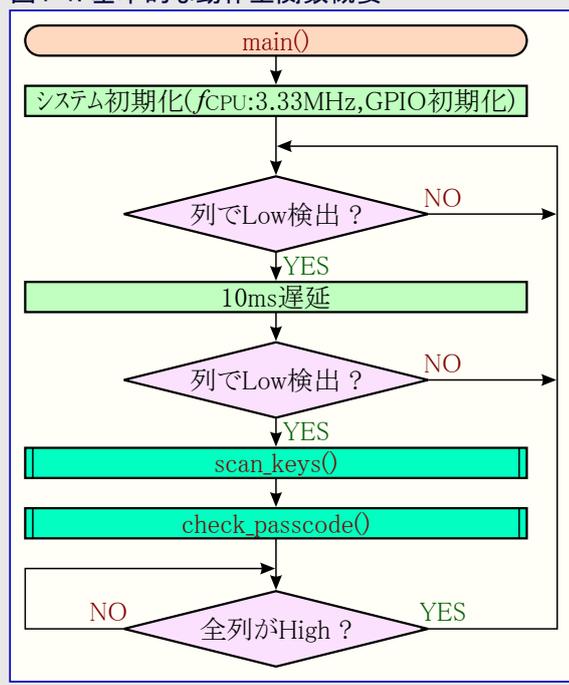
右図は応用主関数を示します。これはマイクロコントローラを既定主クロック設定に初期化し、LEDとキーパッドへ接続された汎用入出力(GPIO)を構成設定します。キーパッドピンはLow出力値を持つ出力としての行と許可された内部プルアップを持つ入力としての列で初期化されます。

図で示されるように、デバイスが行への接続のためにそれらの何れかがLowに引かれたかどうかを知るために列ピンをポーリングします。接続が検出された場合、デバイスは再び列ピンを調べる前に10ms間占有待機をします。列ピンの1つが未だ行電圧に引かれているなら、キー押下が有効と見做されます。この手順は簡単な釦跳ね返し制御技法の実装です。

有効なキー押下が検出されると、`scan_kys()`と`check_passcode()`の関数が走行します。この2つの関数は以降の部分で検討されます。

この2つの関数によってキー押下が正しく処理されると、デバイスは解放されるべき釦に対してポーリングします。これは1回の押下が複数回記録されるのを防ぐためです。釦が解放されると、デバイスは新しいキー押下をポーリングするために戻ります。

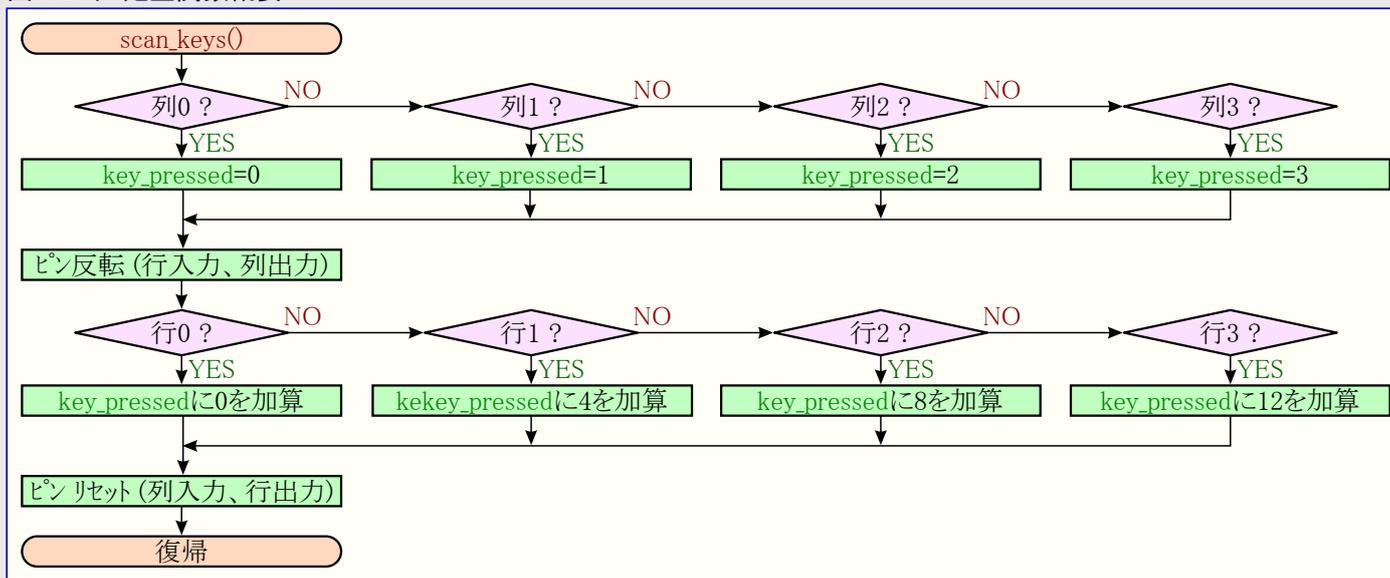
図4-1. 基本的な動作主関数概要



4.1.1. キー走査

scan_keys()関数はどの列が押された鍵を含むかを記録することによって開始します。各列は下図で示されるようにkey_pressed変数で格納される値を割り当てられます。列が記録されると、行と列のピンは行が入力になり、列が出力になると言う意味に於いて反転されます。

図4-2. キー走査関数概要



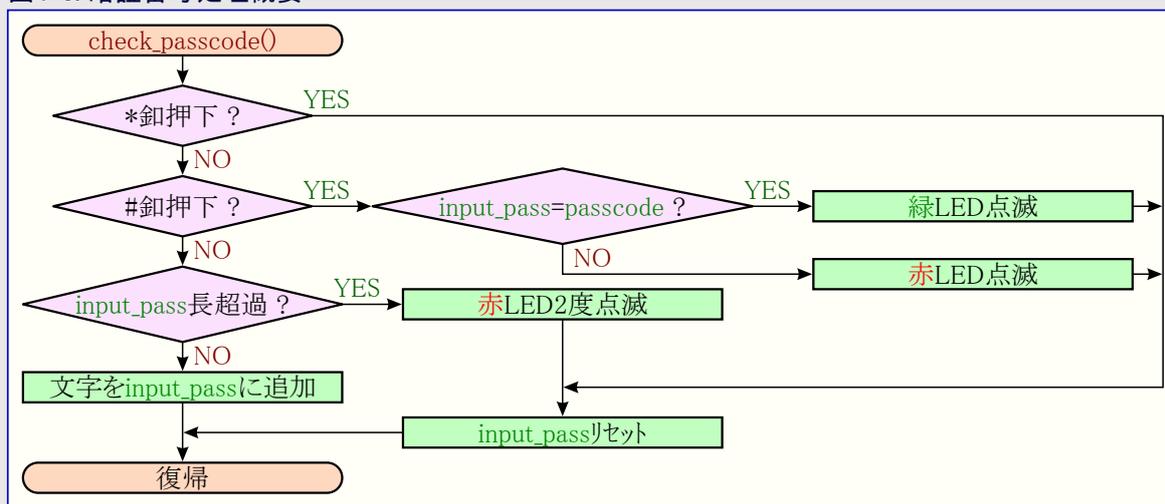
key_pressed変数は押された鍵を含むことを検出された行に応じた値によって増やされます。例えば、B鍵が押された場合、列3が最初に記録され、3の値でkey_pressed変数に格納されています。行1が記録され、変数に4が加算され、合計7を変数に与えます。キーパッドの鍵を左から右、上から下へ数えて、B鍵は(0基準番号付けの仕組みで)鍵番号7です。

鍵押下記録後、ピンは列が入力で行が出力の初期状態にリセットされます。

4.1.2. 暗証番号検査

この例応用では入力コードをリセットするのにアスタリスク(*)キーが使われる一方で、入力コードが暗証番号と合うかどうかを検査するのにシャープ(#)キーが使われます。従って、この関数はこれら特殊文字の1つが押されたかどうかを調べることによって始まります。そうでなければ、押されたアルファベット文字はPINコード入力文字列に追加されます。

図4-3. 暗証番号処理概要



シャープ(#)キーが押されて入力コードが暗証番号と合った場合、入力コードがリセットされる前に緑LEDが点滅されます。一方で、入力コードが合わなかった場合、代わりに赤LEDが点滅されます。アスタリスク(*)キーが押された場合、単に入力コードがリセットされます。

応用は20文字の固定長入力コードを実装します。長さを超えた場合、赤の状態LEDを2度点滅して入力コードをリセットします。

4.2. 高度な動作

tinyAVR®とmegaAVR®のマイクロコントローラシステムは消費電力とCPU利用の両面で応用をもっと効率的にすることができる高度な機能を含みます。以下の項は高度な応用例で使われる各機能を説明します。

4.2.1. 休止

高度なキーパッド応用で実装される最も重要な低電力機能はMCUを休止に置くことです。tinyAVRとmegaAVRのデバイスは利用可能な3つの休止動作を持ちます。休止動作は節電のためにマイクロコントローラの各種部分用クロックを各々OFFにします。

休止動作を使う応用の設計時、CPUが休止の間に起きて(動いて)いるのが必要な周辺機能とクロックがどれかを考慮しなければなりません。例えば、実時間計数器(RTC:Real-Time Counter)はCPUが休止の間に時間の経緯を保つべきで、それによって休止動作を選ばなければなりません。また、休止動作はCPUが休止からどう起きるべきかに基づいて選ばなければなりません。いくつかの割り込み形式は或る休止動作に於いてデバイスを起こすことが不可能です。各種休止動作での周辺機能クロックと割り込みに関する情報はデバイスのデータシートの「休止制御器」章で見つけることができます。

高度なキーパッド応用では鈕を押す間に周辺機能は走行を続けることを必要とされません。従って、パワーダウン休止動作が適切です。この休止動作の起き上がり元を見ると、ピン変化割り込みがCPUを起こすことができるのを知ることができます。これは応用が待っているものがピン変化(即ち、鈕押下)なので応用に対しても適切です。いくつかのピン変化割り込み形式が起動するのにクロック信号に頼ることに注意することが重要です。関連クロックがパワーダウン休止動作でOFFにされるため、ピン変化割り込みに対して非同期(クロック信号と無関係)の起動が選ばなければなりません。この応用では(「4.2.2. 割り込み動作」項で記述される)両端(BOTHEDGES)起動が選ばれます。

パワーダウン休止動作初期化が以下のコード断片で示されます。

```
/* パワーダウン休止動作と休止許可を設定 */
SLPCTRL.CTRLA = SLPCTRL_SMODE_PDOWN_gc | SLPCTRL_SEN_bm;
```

4.2.2. 割り込み動作

基本的なキーパッド応用は鈕が押されたかを知るため、CPUに列ピンを継続的にポーリングさせます。CPUでの鈕のポーリングは電力とCPU資源の両方で非効率な使用です。

この問題を解決するのに割り込みを使うことができます。tinyAVR®とmegaAVR®のデバイスはそれらの汎用入出力(GPIO)ピンの全てに対するピン感知割り込みが特徴です。許可されたピン感知割り込みで、PORT周辺機能はピン上を見て変化が検出され場合、自動的にCPUへ通知します。これはキー押下を待つ間に違う作業を実行する、または節電のために休止へ行くためにCPUを自由にします。

ピン感知割り込みは以下の起動が特徴です。

- **FALLING** : ピンの論理入力水準がHighからLowに落とされた時に起動
- **RISING** : ピンの論理入力水準がLowからHighに上昇された時に起動
- **BOTHEDGES** : ピンの論理入力水準がHighからLowに落とされた、またはLowからHighに上昇された時に起動
- **LEVEL** : ピンがLowを検出された時に必ず起動

ピン感知割り込みを休止動作と組み合わせる時に、更なる考慮が行われなければなりません。全てのピン感知割り込み起動が全ての休止動作に適合するとは限りません。低電力キーパッド応用では可能な最低消費電力が望まれ、従って、パワーダウン休止動作からデバイスを起こすことができる起動元が使われなければなりません。**BOTHEDGES**と**LEVEL**の起動が最も深い休止動作でデバイスを起こすことができます。この応用では**BOTHEDGES**起動が選ばれました。

キーパッドの列に対する**BOTHEDGES**割り込み起動の構成設定が以下のコード断片で示されます。

```
/* 列用BOTHEDGES割り込み許可 */
PORTC.PIN0CTRL = PORT_ISC_BOTHEDGES_gc;
PORTC.PIN1CTRL = PORT_ISC_BOTHEDGES_gc;
PORTC.PIN2CTRL = PORT_ISC_BOTHEDGES_gc;
PORTC.PIN3CTRL = PORT_ISC_BOTHEDGES_gc;
```

4.2.3. 計時器遅延

基本的なキーパッド例と同様にCPUでクロック周期を数えることによって応用の流れを遅らせるのは非効率です。クロック周期計数のような何かつまらないことにCPUを拘束する代わりに、節電のために休止へ行くか、または他の有用な作業を行うことができます。

この応用記述例で使われるATtiny1627マイクロコントローラはCPUと無関係に遅延の経緯を保つことができる実時間計数器(RTC)が特徴です。これは内部32.768kHz RC発振器で動くことができ、これはどの休止動作でも動くことを意味します。RTCはRTCクロック周期数設定後、どの休止動作からもCPUに割り込むことができる周期割り込み計時器(PIT:Periodic Interrupt Timer)が特徴です。

PIT初期化は以下のコード断片で示されます。

```
/* 32 RTCクロック周期毎に始動するように割り込みを設定してPITを許可 */
RTC.PITCTRLA = RTC_PERIOD_CYC32_gc | RTC_PITEN_bm;
/* PIT割り込みを許可 */
RTC.PITINTCTRL = RTC_PI_bm;
```

4.2.4. ピン設定

応用がマイクロコントローラのいくつかのピンを未使用のままにする時に消費電力低下を維持するために特別な考慮が行われるべきです。デバイスが休止動作の時に未使用と不正に構成設定したピンからの消費電力は重要で有り得ます。

考慮の面で最も重要なのは浮いているピンを避けることです。浮きピンは定義された電圧水準に繋がれていないピンです。これらは外側と内部の干渉に基づいて散発的に行動するかもしれません。この散発的な動きは動的消費電力を引き起こすトランジスタの電圧閾値の交差を引き起こすかもしれません。

浮きピンはそのピンに対する内部プルアップを許可することによって避けられます。これは内部プルアップ抵抗を通して電圧を一定の論理High水準に引っ張ります。そのピンのデジタル入力緩衝部も禁止できるかもしれません。これはデジタル入力回路を切断し、さらに消費電力を低下します。

Atmel STARTを使って構成された全ての応用では未使用ピンが自動的に許可された内部プルアップを持つ入力として初期化されます。高度なキーパッド応用ではまた未使用ピンがデジタル入力回路を禁止するようにSTARTで初期化されます。

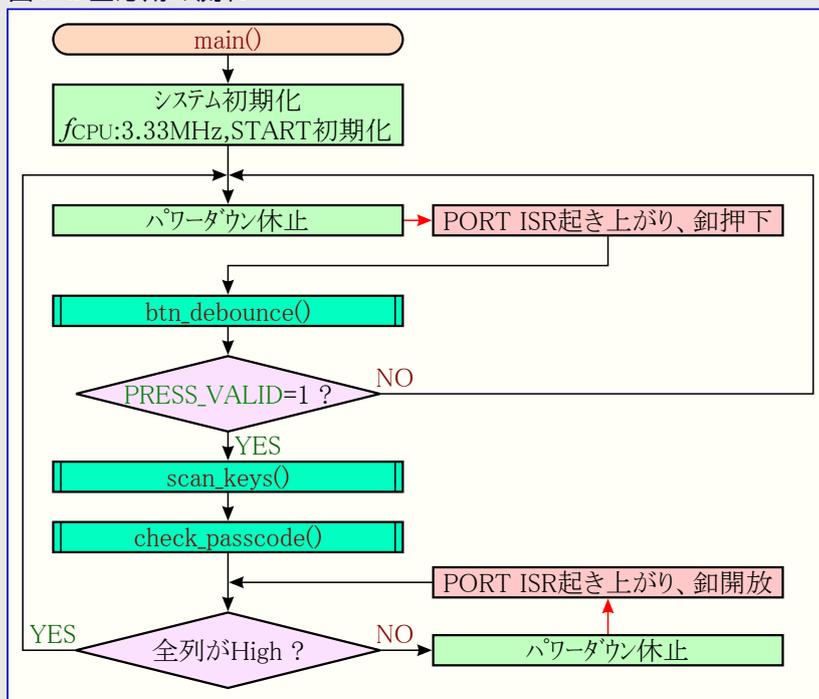
以下のコード断片はPA0ピンを許可されたプルアップと禁止されたデジタル入力回路を持つ入力に構成設定します。

```
/* ピンを入力として設定 */
PORTA.DIRCLR = PIN0_bm;
/* プルアップ許可とデジタル入力緩衝部禁止 */
PORTA.PINCTRL = PORT_PULLUPEN_bm | PORT_ISC_INPUT_DISABLE_gc;
```

4.2.5. 高度なキーパッド実装

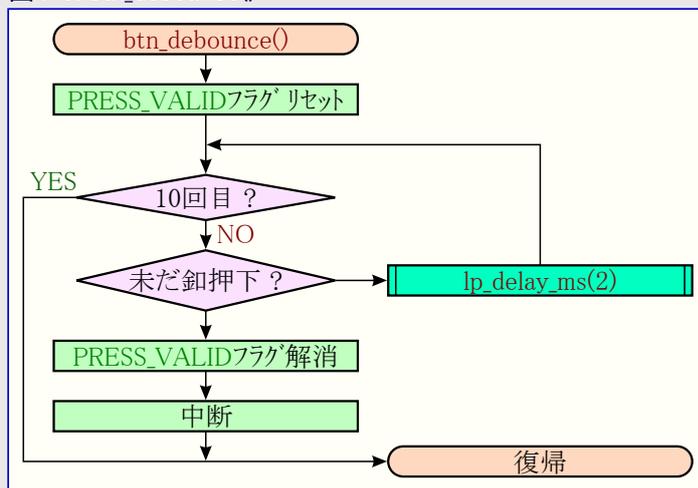
高度なキーパッド応用の主関数の流れが右図で示されます。これはキー押下待ちの時にデバイスが休止に設定されることを示します。ピン変化割り込みがCPUを起き上がらせて押された釦の走査をさせます。

図4-4. 主応用の流れ



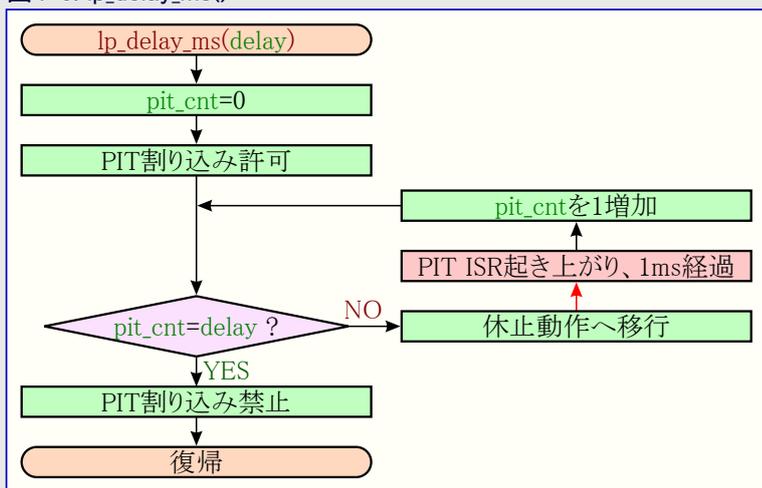
休止から起き上がり後、釦押下は跳ね返り制御(btn_debounce)をされます。これは右図で示されるように釦が押されたことが調べられる繰り返しの10回走行することによって実行されます。10回全てで釦が押されたことを見つげられた場合、釦押下は有効と見做されます。繰り返しの通して反復中に釦が押下として読まれな場合、確認(PRESS_VALID)フラグが解消されてCPUは繰り返しの抜け出します。

図4-5. btn_debounce()



跳ね返り制御(`btn_debounce`)は繰り返しの反復間に2msの遅延を使います。この遅延は右図で示されるような計時器遅延を使って実装されます。周期割り込み計時器(PIT)はms毎に1回割り込みを起動するように計画されます。割り込みが起動されると、計数器(`pit_cnt`)が増やされ、この計数器が望むms数に達すると、CPUはこの関数から戻ります。

`scan_keys()`と`check_passcode()`の関数はそれらがLED表示器のタイミング時に低電力計時器遅延関数を使うことを除いて基本的なキーパッド例のそれらと同じです。

図4-6. `lp_delay_ms()`

5. 消費電力

5.1. 基本的な動作

基本的な応用例でキーパッドを走査する時に、CPUは入出力線のポーリングと遅延用占有待機を継続的に行います。これは下図で見ることができるように相対的な高消費電力で反映されます。

図5-1. 電力分析、キー押下なしでの基本的な動作

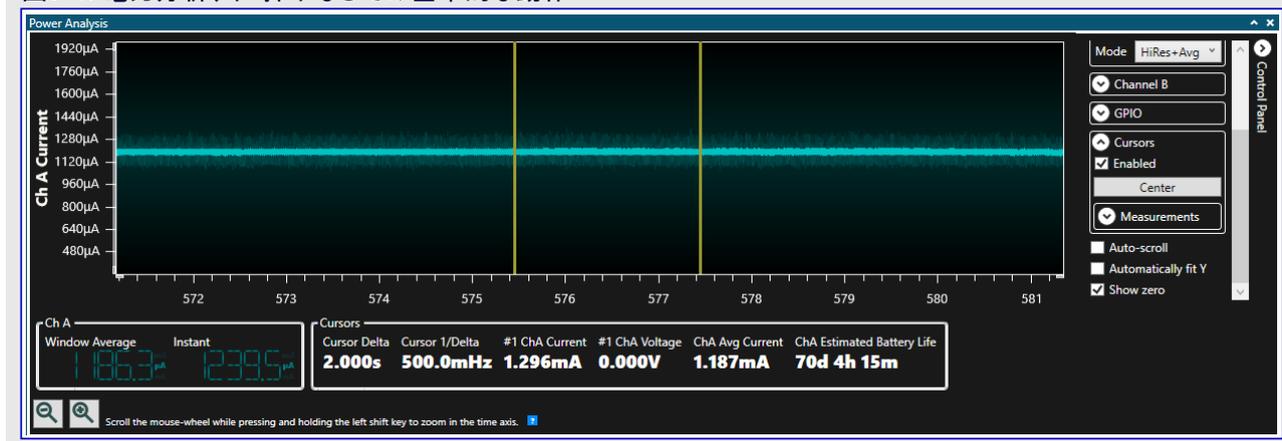
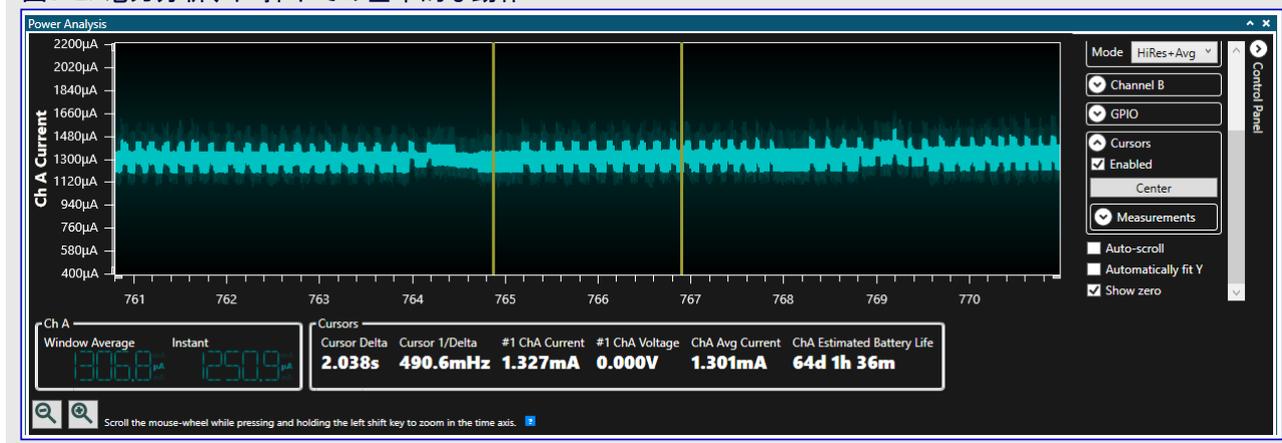


図5-2. 電力分析、キー押下での基本的な動作



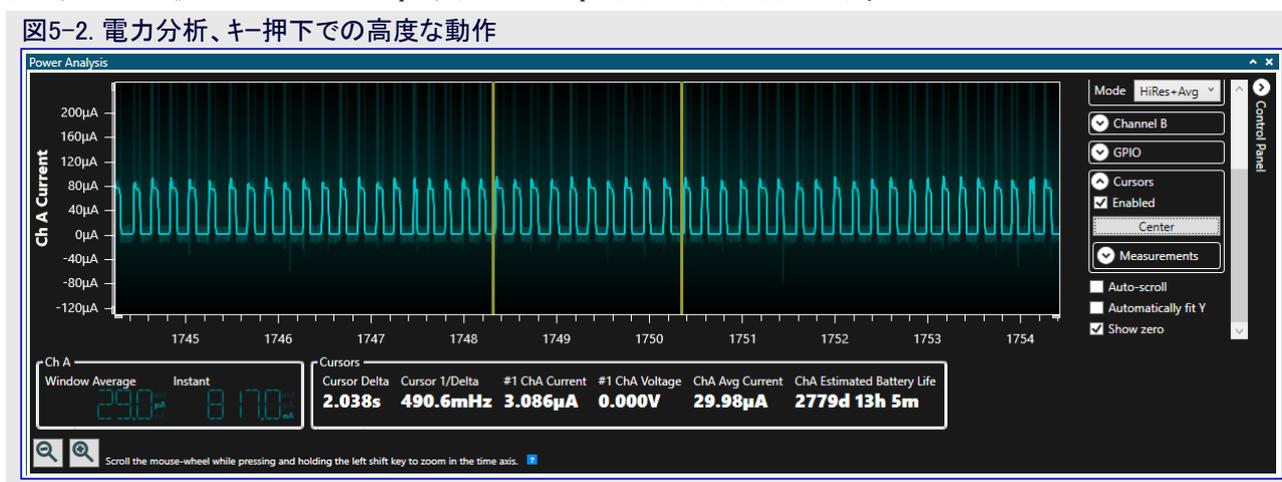
多くの電力を使うことに加えて、未使用ピンが内部プルアップを許可することによって正しく処理されず、これは不安定な消費電力で見えるかもしれません。電流測定時にデバイスの他のピンを触ると、結果が影響を及ぼされることを明確に見ることができます。

5.2. 高度な動作

「4.2. 高度な動作」項で記述された機能を使うと、下図で示されるように消費電力は非常に低くなります。デバイスが休止して鉤が押されていない時は約0.8 μ Aしか使いません。



2つ目の図で見られるように鉤が繰り返して押されてデバイスによって走査される時に消費電力はCPUが休止している時よりもかなり高くなりますが、鉤を押し続けている時に約80 μ A、平均で約30 μ Aと未だ非常に低いです。



5.3. 電流データ作図

以下の指示はPower Debuggerとデータ可視器(Data Visualizer)を使って消費電力を分析する方法を示します。

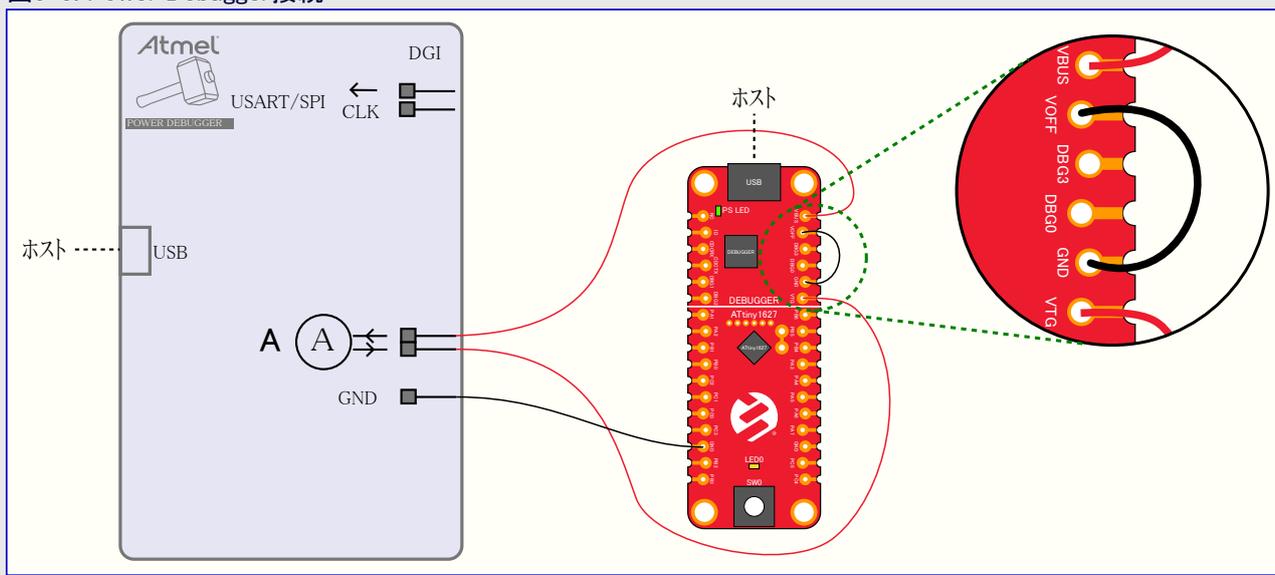
- 注:**
- Power Debuggerの詳細な情報については[Power Debugger使用者の手引き](#)を参照してください。
 - データ可視器の詳細な情報については[データ可視器使用者の手引き](#)を参照してください。

Power Debugger接続

- Power Debuggerの接地参照基準ピンの1つをCuriosity Nano基板の接地ピンに接続してください。
- USB電源とtinyAVRデバイス間の接続をOFFにするためにCuriosity Nano基板上のVOFFピンを接地に接続してください。
- Curiosity Nano基板上のVBUSピンをPower Debugger上の測定回路Aの入力電流ピンに接続してください。
- Power Debugger上の測定回路Aの出力電流ピンをCuriosity Nano基板のVTGピンに接続してください。

Power Debuggerの測定回路Aは今やUSB電源とtinyAVRマイクロコントローラ間に直列で接続され、使われる総電流を測定することができます。

図5-5. Power Debugger接続



データ可視器(Data Visualizer)設定

1. データ可視器を開いてください。
2. データ可視器でConfiguration(構成設定)⇒Visualization(可視化)⇒Power Debugging(電力デバッグ)を開いてください。
3. DGI制御盤(DGI Control Panel)区画でPower Debugger Data Gateway(Power Debuggerデータ中継器)⇒Connect(接続)を選んでください。
4. Power(電力)チェック枠をチェックしてください。
5. 電力分析(Power Analysis)区画の制御盤(Control Panel)でCannel A(チャンネルA)引き落としを展開してください。
6. DGI制御盤(DGI Control Panel)区画のA Current(A電流)からのプラグを電力分析(Power Analysis)区画のCurrent(電流)ソケットにプラグしてください。
7. DGI制御盤(DGI Control Panel)区画でStart(開始)を押してください。
8. 任意選択: Cursors(カーソル)引き落としでEnabled(許可)チェック枠をチェックすることによってカーソルを許可してください。

図5-6. データ可視器での電力デバッグ図



6. Atmel STARTからのコード例取得

コード例は画像ユーザーインターフェース(GUI)を通して応用コードの構成設定を許すウェブに基づくAtmel STARTを通して利用可能です。コード例は下の直接コード例リンクまたはAtmel START先頭頁の**BROWSE EXAMPLES**(例検索)鈕経由でAtmel Studio/MPLAB XとIAR Embedded Workbench®用をダウンロードすることができます。

Atmel STARTウェブ ページ : start.atmel.com/

コード例

- AVRデバイスでの配列キーパッドの使い方 - 基本 (Using Matrix Keypad with AVR Devices - Basic)
 - https://start.atmel.com/#example/Atmel%3AApplication_AVR_Examples%3A1.0.0%3A%3AApplication%3AUsing_Matrix_Keypad_with_AVR_Devices_-_Basic%3A
- AVRデバイスでの配列キーパッドの使い方 - 高度 (Using Matrix Keypad with AVR Devices - Advanced)
 - https://start.atmel.com/#example/Atmel%3AApplication_AVR_Examples%3A1.0.0%3A%3AApplication%3AUsing_Matrix_Keypad_with_AVR_Devices_-_Advanced%3A

例プロジェクトについての詳細と情報に関してはAtmel STARTで**User guide**(使用者の手引き)をクリックしてください。 **User guide**鈕はAtmel STARTプロジェクト形態設定部内の一覧画面でプロジェクト名をクリックすることにより、例閲覧部で見つけることができます。

Atmel Studio

DOWNLOAD SELECTED EXAMPLE(選んだ例をダウンロード)をクリックすることにより、Atmel STARTで例閲覧部からAtmel Studio用 **.atzip**ファイルとしてコードをダウンロードしてください。Atmel START内からファイルをダウンロードするには、**EXPORT PROJECT**(プロジェクトをエクスポート)に続いて**DOWNLOAD PACK**(一括ダウンロード)をクリックしてください。

ダウンロードした **.atzip**ファイルをダブルクリックしてください。プロジェクトがAtmel Studio 7.0にインポートされます。

MPLAB X

EXPORT PROJECT(プロジェクトをエクスポート)に続いて**DOWNLOAD PACK**(一括ダウンロード)をクリックすることによってAtmel START内でMPLAB X IDE用 **.atzip**ファイルとしてコードをダウンロードしてください。

MPLAB XでAtmel START例を開くにはMPLAB Xでメニューから**File**(ファイル)⇒**Import**(インポート)⇒**START MPLAB Project**(START MPLABプロジェクト)を選んで **.atzip**ファイルに誘導してください。

IAR Embedded Workbench

IAR Embedded Workbenchでプロジェクトをインポートする方法の情報については**Atmel START使用者の手引き**を開き、**Using Atmel Start Output in External Tools**(外部ツールでAtmel START使用)と**IAR Embedded Workbench**を選んでください。Atmel START使用者の手引きへのリンクは共に頁の右上隅に置かれたAtmel START先頭頁から**Help**(手助け)またはプロジェクト形態設定部内の**Help And Support**(手助けと支援)をクリックすることによって見つけることができます。

7. GitHubからのコード例取得

コード例は画像ユーザーインターフェース(GUI)を通して応用コードを提供するウェブに基づくサーバであるGitHubを通して入手可能です。コード例はAtmel StudioとMPLAB Xの両方で開くことができます。MPLAB XでAtmel Studioプロジェクトを開くにはMPLAB Xでメニューから**File**(ファイル)⇒**Import**(インポート)⇒**START MPLAB Project**(START MPLABプロジェクト)を選んで **.cproj**ファイルに誘導してください。

GitHubウェブ 頁 : [GitHub](https://github.com)

コード例



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。

Clone(複製)または**download**(ダウンロード)鈕をクリックすることによってGitHub上の例頁から **.zip**ファイルとしてコードをダウンロードしてください。

8. 改訂履歴

資料改訂	日付	注釈
A	2020年6月	初版文書公開

Microchipウェブ サイト

Microchipは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
- **Microship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、PICtail、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。

© 2020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www.microchip.com/qualityを訪ねてください。

日本語© HERO 2020.

本応用記述はMicrochipのAN3407応用記述(DS00003407A-2020年6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。



MICROCHIP

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: www.microchip.com/support ウェブアドレス: www.microchip.com 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 ボストン 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	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 - 香港特別行政区 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: 45-4485-5910 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 - ガルヒング 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 - パドバ Tel: 39-049-7625286 オランダ - デルフト 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 - ウォーキングハム Tel: 44-118-921-5800 Fax: 44-118-921-5820