

AVR240 : 4×4キーパッド、キー押下での起動

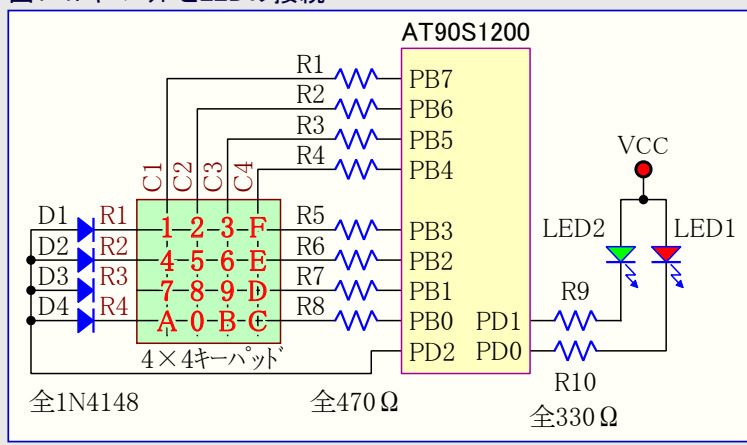
要点

- 4×4配列での16キー押下パッド
- 非常に低い電力消費
- 休止形態でのAVRとキー押下での起動
- 最小の外部部品
- 必要ならば含まれるESD保護
- 効率的なコード
- AT90S1200用の完全なプログラムを含む
- どのAVR MCUにも適合

1. 序説

この応用記述は低電力電池動作用に設計された4×4キーパッドに対する簡単なインターフェースを説明します。AVRは殆どの時をパワーダウン動作で過し、キー押下に従って2つのLEDの1つを点滅する簡単な試験プログラムを起すためにキーが押された時に起動します。'0'が押されたなら、赤LEDが10回点滅します。他の全てのキーはキーに記された回数で緑LEDを点滅します(例えば'C'が押された場合、緑LEDが12回点滅します)。

図1-1. キーパッドとLEDの接続



2. 動作の理屈

キーパッドの列はポートBの上位ニブルに接続されます。キーパッドの行は下位ニブルに接続されます。抵抗器R1～R8(これは図1-1.で示されます)はキーパッドからのESD現象に於いて入力電流を安全な水準に制限することが分かります。これらは殆どの応用に於いて省略することができます。安定状態に於いて上位ニブルは出力且つLow状態として構成設定されます。下位ニブルは許可された内部プルアップを持つ入力として構成設定され、外部プルアップ抵抗の必要をなくします。AVR初期化後は休止形態に置かれます。キーが押下されると、ダイオードD1～D4の1つがPD2外部割り込み線をLowに引っ張り、そしてこれも許可された内部プルアップを持ちます。これはAVRを起して割り込み処理ルーチンを走行させ、そしてそれはキーパッドを走査してどのキーが押されたかを算定します。

そして主プログラムに戻り、押されたキーに従ってLEDを駆動し、それが終了した時にAVRを元に戻し置きます。

抵抗器R9とR10は伝統的なLEDに対する電流制限抵抗で、供給電圧に対してどの適合値にもできます。この応用記述は5V供給で330Ωを使って検査されました。LEDは引き込み電流動作('0'=ON)で駆動され、指定値で約10mAの順方向電流を供給します。



8ビット AVR[®]
マイクロコントローラ

応用記述

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、Atmel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。

Rev. 1232D-06/06, 1232DJ3-02/21

3. 実装

ファームウェアはリセットルーチン、試験プログラム、割り込み処理ルーチンの3つの項から成り、リセットルーチンはポート、休止形態、節電と割り込みを設定します。試験プログラムは起動でLEDを点滅し、そして割り込み処理ルーチンはキー押下に応答します。

3.1. リセットルーチン

リセットルーチン用の流れ図が図3-1.で示されます。リセットでポートはそれらの開始方向で初期化されます。ポートDに於いて、外部割り込み用の入力でなければならないPD2を除いて、全てのビットが出力として固定化されます。この(PD2)ビットはPORTDのビットを設定(1)することによって許可されたプルアップを持ちます。雑音を拾う、または過大な電力消費を避けるために、未使用ビットは出力として構成設定され、さもなければ浮き状態のままの場合、それらが起き得ます。ポートBは'0'を送る出力としての上位ニブルと許可されたプルアップを持つ入力として設定された下位ニブルで始まります。

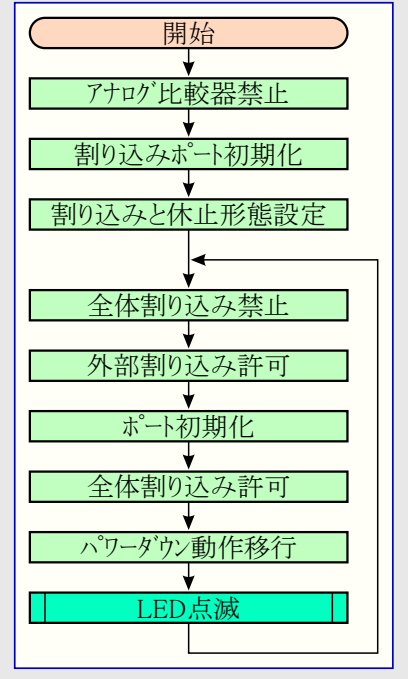
外部部品の最小使用のため、入力として設定されるそれらのビットの全てに対して内部プルアップがONされることを保証しなければなりません。これは出力に対して'1'、入力に対して'0'でデータ方向レジスタ(DDR)を構成設定し、そしてPORTレジスタに於いて入力ビットへ'1'を書くことによって達成されます。このプログラムは'0'を探し、'0'でない場合にキー押下での動きを飛ばすためにSBIS命令を用います。

MCU制御レジスタ(MCUCR)の休止命令許可(SE)と休止形態選択(SM)のビットを設定(1)することによってパワーダウン動作が選択されます。同時に割り込み条件(ISC01,0)ビットに'0'を書くことによって外部割り込みが構成設定されます。これは外部割り込み0(INT0)をLowレベルでの起動に設定します。"パワーダウン動作"使用時、AVRはLowレベル起動でだけ起きることができます。

アナログ比較器をOFFにすることが更に電力消費を減らします。これはアナログ比較器制御/状態レジスタ(ACSR)のアナログ比較器禁止(ACD)ビットを設定(1)することによって行われます。これは注意して行われなければならない、さもなければ望まれない割り込みが生成され得ます。このプログラムはプログラムが割り込まれる準備が整うまで全体割り込みを禁止して、この問題を解決します。アナログ比較器を使うつもりなら、このコードを削除できますが、ポートB(AIN1,AIN0)がキーパッドに使われているため、キーパッド用のポートを変更する必要があります。

そしてAVRは休止形態へ移行します。割り込み機能が終了されて"点滅"試験ルーチンが処理された後で休止形態へ戻ることを保証するため、これは主繰り返し内に配置されます。キー押下後にAVRが起きると、割り込みルーチンが終了された後で"点滅"ルーチンが呼ばれます。"点滅"ルーチンが終わると、外部割り込みが許可され、このために別の割り込みが起き得ます。

図3-1. リセットと主ルーチンの流れ図



3.2. 点滅試験関数

流れ図が図3-2.で示されます。

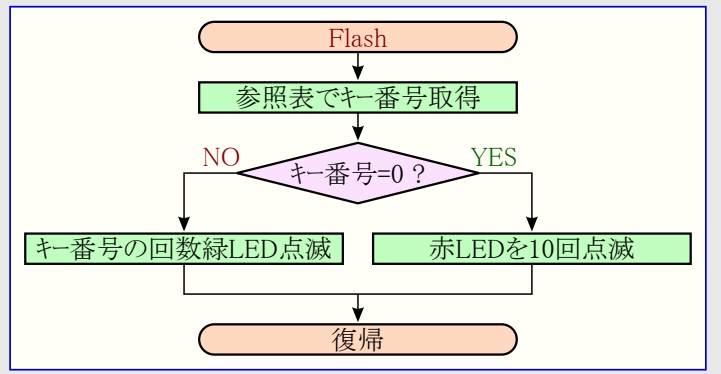
この関数は"パワーダウン"動作の外で実行されるべき使用者自身の応用によって置き換えることができます。これはキー走査ルーチンが正しく動いていることを実演するのに役立ちます。押されたキーの値は"key"変数から取られ、EEPROMに格納された16バイトの参照表をアクセスするための位置指示子として使われます。参照表は押されたキーの番号を含みます。

プログラムをずっと短くし、キー押下に対して完全なASCII符号化を提供するための容易な拡張を許す、この2つの理由のために表が使われます。より大きなAVRについては、この表をプログラム用メモリに格納し、LPM命令を使ってそれをアクセスすることが意味を持つでしょう。

そしてEEPROMから配給されたキー値はLED出力に対するON/OFF周回内側で下降計数変数として使われます。この値が0の場合は赤LEDが10回点滅されます。この値が0以外なら、緑LEDがその回数分点滅されます。例えば、"3"のキーに対しては3回、"F"のキーに対しては15回などです。AVRは周回を繰り返し、そして眠りに落ちます。

LED点滅ルーチンは"flash"関数を使用者のルーチンによって置き換えることで、使用者自身の応用へ容易に変更されます。主な考慮点はタイミングです。試験プログラムがLEDを点滅してかなりの時間を費やすので、追加の(接点)反動準備は必要ありません。使用者コードが非常に速い場合、接点に戻る間の時間を許すために短い遅延を置く必要があるかもしれません。(パワーダウン)休止形態からの起動は、より新しいデバイスに於いてこれが減らされつつあるとしても、代表的に16msまたはその程度かかります。これはいくつかの反発防止も提供します。

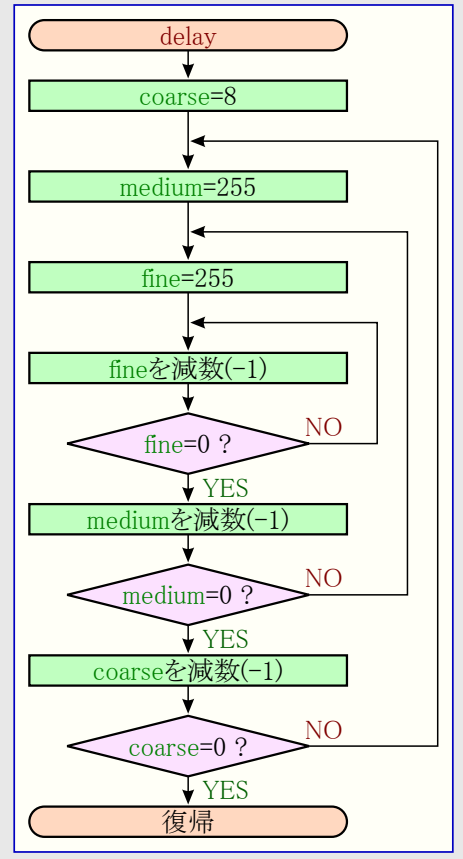
図3-2. "Flash"関数流れ図



3.3. 長時間遅延サブルーチン (delay)

LEDの点滅を見るためには最低0.25秒の遅延が必要です。これは他の作業用にタイマ/カウンタの自由を保つため、伝統的なFOR繰り返しを使って達成されます。4MHzのクロックで0.25秒以上の遅延を達成するには、3つを入れ子にした繰り返しが必要です。“fine”、“medium”、“coarse”のレジスタに含まれた3つの局所変数が繰り返し用に使われます。“fine”と“medium”のカウンタは最大の255回で動き、8に設定された“coarse”と共に約0.4秒の遅延を生成します。流れ図は図3-3.で示されます。

図3-3. “delay”サブルーチン流れ図



3.4. 割り込み処理ルーチン

行われていた主プログラムでのどの動きも不正にするのを避けるために、入口に於いてステータスレジスタ(SREG)が保存されます。この応用では望むなら最適化のためにそれを省けるかもしれませんが(訳補:本例では基本的に休止形態中に割り込まれるため)。流れ図が図3-4.で示されます。

'0'を探して各行入力を順番に検査することにより、最初に行キーが検出されます。そして0,4,8,12の基数が“key”変数に割り当てられます。そしてポートはポートBの入出力が交換されて再初期化され、それで行キーが検査されます。

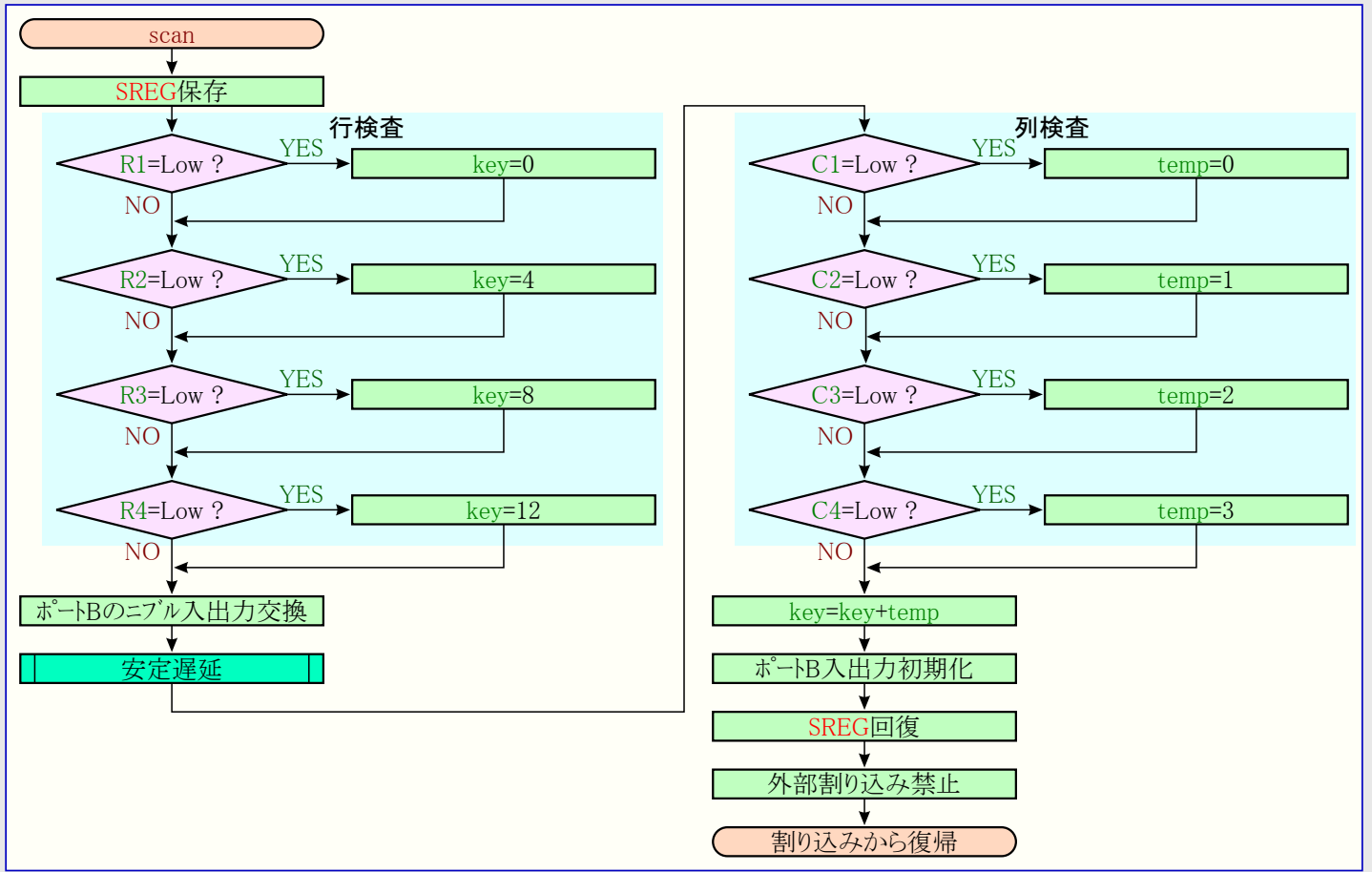
放電する時間をピンに許すために短時間遅延“settle”が使われます。これはFOR繰り返し計画を使う伝統的な時間を浪費する繰り返しの形式を取ります。

そして列キーが検出され、“temp”一時変数内に0,1,2,3で番号が割り当てられます。そして“key”と“temp”を加算することによって最終的な押下キーが算定され、“flash”関数によって使うためにその結果を“key”に置きます。この方法は本应用到に於いて伝統的な単一ビット走査よりもコードに対してより簡単です。

ステータスレジスタ(SREG)が回復されるのに先立ってポートB構成設定が交換し戻されます。これは再び安定遅延を使うのを省きます。

最後に於いて外部割り込みが禁止されます。これは割り込みルーチンが抜け出し直後に再び起動されるのを避けるために行われます。

図3-4. 割り込み処理ルーチン流れ図 (訳注: 原書の図3-4.と図3-5.は図3-4.として纏めました。)



3.5. 短時間遅延サブルーチン

この短遅延はポートBの入出力構成設定変更時に安定時間をピン値に与えるために必要とされます。ルーチンはFOR繰り返し用の単一繰り返し計数器として“temp”全域雑用レジスタを使います。これは4MHzで0.192msの遅延を提供します。この値はそれを速めるために時間が欠くことができない、または形態再設定に先立ってピンがHighに設定される場合に実験によって短くすることができます。これはこの遅延の必要を完全に取り去るかもしれませんが。

3.6. 資源

表3-1. CPUとメモリの使用

関数	コード量(語)	周期数	使用レジスタ	割り込み	説明
main	22	19	R16	-	初期化
flash	20	-	R16	-	例プログラム
scan	31	47 (代表値)	R16,R17,R21	外部割り込み0(INT0)	4×4キーパッド走査
delay	10	1,000,000	R18,R19,R20	-	例プログラムだけで使われる約0.4秒遅延
settle	4	764	R16	-	scanで使われるピン安定時間遅延
合計	85	-	R16~R21	-	

表3-2. 周辺機能の使用

周辺機能	説明	許可割り込み
外部割り込み0(INT0)	キー押下起動信号	外部割り込み0 (Lowレベル起動)
EEPROM (16バイト)	キーに対する値割り当て	-
ポートBの8つのI/Oピン	4×4キーパッド接続	-
ポートDの2つのI/Oピン	LED点滅 (例のみ)	-

```

;**** 応用記述 AVR240 ****
;*
;* 表題       : 4×4キーパッド、キー押下での起動
;* 版         : 1.2
;* 最終更新   : 2004.11.11
;* 対象       : 全AVRデバイス
;*
;* 支援Eメール : avr@atmel.com
;*
;* 説明
;*
;* この応用記述は4×4キーパッドを走査し、キー押下でAVRを起させる、休止形態を使います。この設計は最小の外部部品を使いま
;* す。含まれているのはキーが押された時にAVRを起して走査を実行し、そして2つのLEDの1つを押されたキーの数で点滅する
;* 試験プログラムです。起すのに外部割り込み線が使われます。この例はAT90S1200で動きますが、ベクタ、EEPROM、スタックポイン
;* タの適切な変更でどのAVRにもできます。タイミングは4MHzクロックを仮定します。もっと進んだプログラム、例えば表示器へのASCII
;* 出力で使われる同じ構造を可能とするため、EEPROMでの参照表が用いられます。
;*****

    .INCLUDE "1200def.inc"

;
; * 全プログラムにて使われるレジスタ *
; * 全ルーチンにて使われる全域変数 *

    .DEF      temp      =R16      ;一般雑用

;
; (ポートBピン)

    .EQU     ROW1      =3        ;キーパッド行1入力
    .EQU     ROW2      =2        ;キーパッド行2入力
    .EQU     ROW3      =1        ;キーパッド行3入力
    .EQU     ROW4      =0        ;キーパッド行4入力
    .EQU     COL1      =7        ;キーパッド列1入力
    .EQU     COL2      =6        ;キーパッド列2入力
    .EQU     COL3      =5        ;キーパッド列3入力
    .EQU     COL4      =4        ;キーパッド列4入力

;
; (ポートDピン)

    .EQU     GREEN     =0        ;緑LED出力
    .EQU     RED       =1        ;赤LED出力
    .EQU     INTR      =2        ;外部割り込み0入力

```

```

; * 割り込み処理ルーチンで使われるレジスタ *

. DEF      key      =R17      ;EEPRM参照表位置指示子
. DEF      status   =R21      ;SREG保存用

; * 遅延(delay)サブルーチンで局所変数として使われるレジスタ *

. DEF      fine     =R18      ;内側遅延計数器
. DEF      medium   =R19      ;中側遅延計数器
. DEF      coarse   =R20      ;外側遅延計数器

; * キー値変換用参照表 *

. ESEG                                ;EEPROM領域

. ORG      0

. DB       1                ; 0: 列1(C1)-行1(R1)
. DB       2                ; 1: 列2(C2)-行1(R1)
. DB       3                ; 2: 列3(C3)-行1(R1)
. DB      15                ; 3: 列4(C4)-行1(R1)
. DB       4                ; 4: 列1(C1)-行2(R2)
. DB       5                ; 5: 列2(C2)-行2(R2)
. DB       6                ; 6: 列3(C3)-行2(R2)
. DB      14                ; 7: 列4(C4)-行2(R2)
. DB       7                ; 8: 列1(C1)-行3(R3)
. DB       8                ; 9: 列2(C2)-行3(R3)
. DB       9                ;10: 列3(C3)-行3(R3)
. DB      13                ;11: 列4(C4)-行3(R3)
. DB      10                ;12: 列1(C1)-行4(R4)
. DB       0                ;13: 列2(C2)-行4(R4)
. DB      11                ;14: 列3(C3)-行4(R4)
. DB      12                ;15: 列4(C3)-行4(R4)

;***** ソースコード *****

. CSEG                                ;コード領域

. ORG      0

RJMP      reset                      ;リセットベクタ
RJMP      scan                       ;外部割り込み0ベクタ
RETI      ;[未使用] タイマ/カウンタ0溢れ割り込みベクタ
RETI      ;[未使用] アナログ比較器割り込みベクタ

; * リセット・主ルーチン *

reset:                                     ;(他の殆どのAVRに関してスタックポインタ初期化を忘れずに)
SBI       ACSR, ACD                   ;アナログ比較器禁止(節電のため)
LDI       temp, 0b11111011            ;ポートD方向指定値取得
OUT       DDRD, temp                  ;PD2(INT0)以外を出力に設定
LDI       temp, 0b00110000            ;休止と割り込み条件の指定値取得
OUT       MCUCR, temp                 ;パワーダウン動作で休止命令許可,Lowレベル割り込み設定

main:
CLI       ;全体割り込み一時禁止
LDI       temp, 0b01000000            ;INT0ビット値取得
OUT       GIMSK, temp                ;外部割り込み0(INT0)許可
LDI       temp, 0b11110000            ;キー配列初期入出力値取得
OUT       DDRB, temp                 ;列キー出力/行キー入力に設定
LDI       temp, 0b00001111            ;列キー初期出力値と行キーのプルアップ指定値を取得
OUT       PORTB, temp                ;列キーにLow出力/行キーのプルアップ許可
LDI       temp, 0b00000111            ;INT0プルアップ指定とLED消灯値を取得
OUT       PORTD, temp                ;INT0プルアップ許可,LED消灯
SEI       ;全体割り込み許可
SLEEP    ;パワーダウン休止形態へ移行(ここでキー押下による走査発生)
RCALL    flash                       ;LED点滅(使用例)
RJMP     main                         ;無限継続へ

```



```

;      * INT0割り込み処理 *
scan:  IN          status, SREG      ;現SREG値一時保存
;
;      SBIS        PINB, ROW1       ;行1(R1)=Highでスキップ
;      LDI         key, 0           ;Low(押下)で行1基準値設定
;
;      SBIS        PINB, ROW2       ;行2(R2)=Highでスキップ
;      LDI         key, 4           ;Low(押下)で行2基準値設定
;
;      SBIS        PINB, ROW3       ;行3(R3)=Highでスキップ
;      LDI         key, 8           ;Low(押下)で行3基準値設定
;
;      SBIS        PINB, ROW4       ;行4(R4)=Highでスキップ
;      LDI         key, 12          ;Low(押下)で行4基準値設定
;
;      LDI         temp, 0b00001111 ;キー配列反転入出力値取得
;      OUT         DDRB, temp        ;列キー入力/行キー出力に設定
;      LDI         temp, 0b11110000 ;列キーのプルアップ指定値と行キーのLow出力値を取得
;      OUT         PORTB, temp       ;列キーのプルアップ許可/行キーにLow出力
;      RCALL       settle           ;ポート安定時間遅延
;
;      SBIS        PINB, COL1        ;列1(C1)=Highでスキップ
;      LDI         temp, 0           ;Low(押下)で列1基準値設定
;
;      SBIS        PINB, COL2        ;列2(C2)=Highでスキップ
;      LDI         temp, 1           ;Low(押下)で列2基準値設定
;
;      SBIS        PINB, COL3        ;列3(C3)=Highでスキップ
;      LDI         temp, 2           ;Low(押下)で列3基準値設定
;
;      SBIS        PINB, COL4        ;列4(C4)=Highでスキップ
;      LDI         temp, 3           ;Low(押下)で列4基準値設定
;
;      ADD         key, temp          ;押下位置キー走査番号取得
;      LDI         temp, 0b11110000 ;キー配列初期入出力値取得
;      OUT         DDRB, temp        ;列キー出力/行キー入力に設定
;      LDI         temp, 0b00001111 ;列キー初期出力値と行キーのプルアップ指定値を取得
;      OUT         PORTB, temp       ;列キーにLow出力/行キーのプルアップ許可
;
;      LDI         temp, 0b00000000 ;INT0ビット=0値取得
;      OUT         GIMSK, temp       ;外部割り込み0(INT0)禁止(レベル割り込みによる連続起動防止用)
;
;      OUT         SREG, status      ;SREG回復
;      RETI                          ;割り込み元へ復帰
;
;      (ポート安定化用安定時間待機(約0.2ms/4MHz))
settle: LDI         temp, 255        ;計数器初期化
tagain: DEC         temp             ;計数器減数
;      BRNE        tagain           ;計数完了まで継続へ
;
;      RET                          ;呼び出し元へ復帰

```

```

;      * キー押下データを使ってLEDを点滅する試験プログラム例 *

flash:  OUT      EEAR, key      ;キー走査番号をEEPROMアドレスとして設定
        SBI      EECR, EERE     ;参照表対応位置読み込み開始
        IN       temp, EEDR     ;対応キー番号取得(点滅回数)
        TST      temp          ;キー番号=0判定
        BREQ     zero          ;0で分岐
;
green_flash:
        CBI      PORTD, GREEN   ;緑LED点灯
        RCALL    delay         ;約0.4s遅延
        SBI      PORTD, GREEN   ;緑LED消灯
        RCALL    delay         ;約0.4s遅延
        DEC      temp          ;点滅計数器減数
        BRNE     green_flash   ;点滅回数分まで継続へ
;
        RET                    ;呼び出し元へ復帰
;
zero:   LDI      temp, 10       ;点滅回数を10に変更
flash_again:
        CBI      PORTD, RED     ;赤LED点灯
        RCALL    delay         ;約0.4s遅延
        SBI      PORTD, RED     ;赤LED消灯
        RCALL    delay         ;約0.4s遅延
        DEC      temp          ;点滅計数器減数
        BRNE     flash_again   ;点滅回数分まで継続へ
;
        RET                    ;呼び出し元へ復帰
;
( LED点滅用時間遅延(約0.4s/4MHz) )
delay:  LDI      coarse, 8      ;外側計数器初期化
cagain: LDI      medium, 255    ;中側計数器初期化
magain: LDI      fine, 255     ;内側計数器初期化
fagain: DEC      fine          ;内側計数器減数
        BRNE     fagain        ;内側計数完了まで継続へ
;
        DEC      medium        ;中側計数器減数
        BRNE     magain        ;中側計数完了まで継続へ
;
        DEC      coarse        ;外側計数器減数
        BRNE     cagain        ;外側計数完了まで継続へ
;
        RET                    ;呼び出し元へ復帰

```



本社

Atmel Corporation

2325 Orchard Parkway
San Jose, CA 95131, USA
TEL 1(408) 441-0311
FAX 1(408) 487-2600

国外営業拠点

Atmel Asia

Unit 1-5 & 16, 19/F
BEA Tower, Millennium City 5
418 Kwun Tong Road
Kwun Tong, Kowloon
Hong Kong
TEL (852) 2245-6100
FAX (852) 2722-1369

Atmel Europe

Le Krebs
8, Rue Jean-Pierre Timbaud
BP 309
78054 Saint-Quentin-en-Yvelines
Cedex
France
TEL (33) 1-30-60-70-00
FAX (33) 1-30-60-71-11

Atmel Japan

104-0033 東京都中央区
新川1-24-8
東熱新川ビル 9F
アトメル ジャパン株式会社
TEL (81) 03-3523-3551
FAX (81) 03-3523-7581

製造拠点

Memory

2325 Orchard Parkway
San Jose, CA 95131, USA
TEL 1(408) 441-0311
FAX 1(408) 436-4314

Microcontrollers

2325 Orchard Parkway
San Jose, CA 95131, USA
TEL 1(408) 441-0311
FAX 1(408) 436-4314

La Chantrerie
BP 70602
44306 Nantes Cedex 3
France
TEL (33) 2-40-18-18-18
FAX (33) 2-40-18-19-60

ASIC/ASSP/Smart Cards

Zone Industrielle
13106 Rousset Cedex
France
TEL (33) 4-42-53-60-00
FAX (33) 4-42-53-60-01

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906, USA
TEL 1(719) 576-3300
FAX 1(719) 540-1759

Scottish Enterprise Technology Park
Maxwell Building
East Kilbride G75 0QR
Scotland
TEL (44) 1355-803-000
FAX (44) 1355-242-743

RF/Automotive

Theresienstrasse 2
Postfach 3535
74025 Heilbronn
Germany
TEL (49) 71-31-67-0
FAX (49) 71-31-67-2340

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906, USA
TEL 1(719) 576-3300
FAX 1(719) 540-1759

Biometrics

Avenue de Rochepleine
BP 123
38521 Saint-Egreve Cedex
France
TEL (33) 4-76-58-47-50
FAX (33) 4-76-58-47-60

文献請求

www.atmel.com/literature

お断り: 本資料内の情報はAtmel製品と関連して提供されています。本資料またはAtmel製品の販売と関連して承諾される何れの知的所有権も禁反言あるいはその逆によって明示的または暗示的に承諾されるものではありません。Atmelのウェブサイトに表示する販売の条件とAtmelの定義での詳しい説明を除いて、商品性、特定目的に関する適合性、または適法性の暗黙保証に制限せず、Atmelはそれらを含むその製品に関連する暗示的、明示的または法令による如何なる保証も否認し、何ら責任がないと認識します。たとえAtmelがそのような損害賠償の可能性を進言されたとしても、本資料を使用できない、または使用以外で発生する(情報の損失、事業中断、または利益の損失に関する制限なしの損害賠償を含み)直接、間接、必然、偶然、特別、または付随して起こる如何なる損害賠償に対しても決してAtmelに責任がないでしょう。Atmelは本資料の内容の正確さまたは完全性に関して断言または保証を行わず、予告なしでいつでも製品内容と仕様の変更を行う権利を保留します。Atmelはここに含まれた情報を更新することに対してどんな公約も行いません。特に別の方法で提供されなければ、Atmel製品は車載応用に対して適当ではなく、使用されるべきではありません。Atmel製品は延命または生命維持を意図した応用での部品としての使用に対して意図、認定、または保証されません。

© Atmel Corporation 2006. 不許複製 Atmel®、ロコとそれらの組み合わせ、AVR®とその他はAtmel Corporationの登録商標または商標またはその附属物です。他の用語と製品名は一般的に他の商標です。

© HERO 2021.

本応用記述はAtmelのAVR240応用記述(doc1232.pdf Rev.1232D-06/06)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。