

ロータリーエンコーダー変換表示基板の製作

ロータリーエンコーダーと8ピンのワンチップマイコン(AT90S2343)を組み合わせた基板を作ってみました。今回は7セグ表示用のCMOS IC 4511とカウント用IC74HC192を使用しました。表示用IC2つをつなげていくことにより、任意に桁数を増やすことができます。構成図は次のようになります。



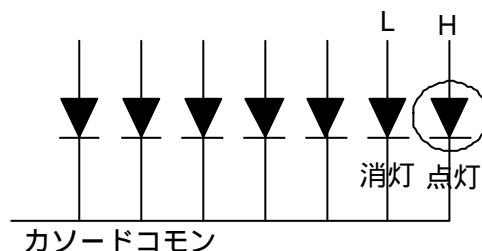
ロータリーエンコーダーからの信号で回転方向を判定して1クリックごとに1パルスを表示部に送るようにしてあります。カウント数は表示部のIC 74HC192が記憶していて、カウントUP(またはDOWN)の信号に応じてIC内で記憶している値を増減していきます。それを2進数のコードで表現(A,B,C,D)で出力します。

$$0 \sim 9 \text{ までの数} = D \times 2^3 + C \times 2^2 + B \times 2^1 + A$$

この上の関係に従った出力となるので、例えば5であれば

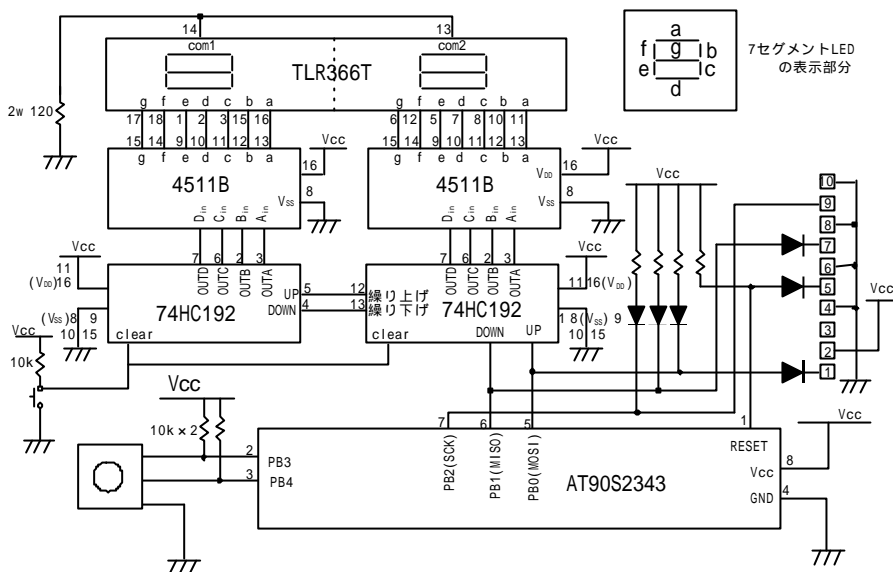
$$5 = 0 \times 2^3 + 2^2 + 0 \times 2^1 + 1$$

(A,B,C,D) = (1, 0, 1, 0)となります。この出力を4511の入力端子A,B,C,Dに入れると対応した7セグコード(a~g)が出力されます。右図のように、7セグメントLEDの入力がHレベル(ここでは5V)のときLEDが点灯し、Lレベル(ここでは0V)で消灯します。

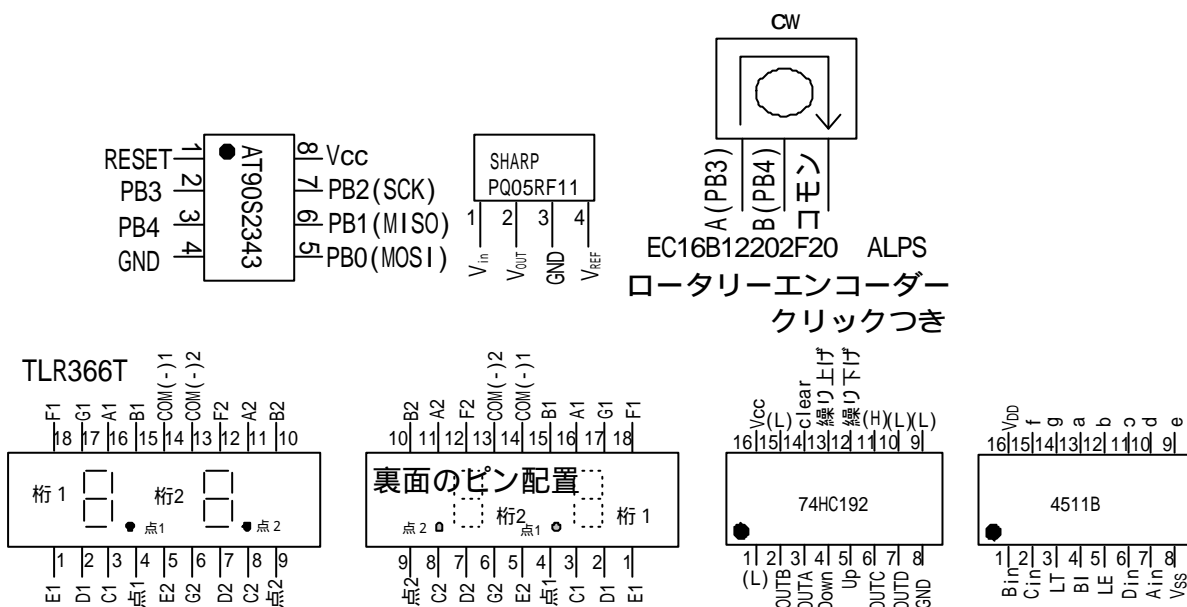


LEDにはアノードコモン(+), カソードコモン(-)があり点灯状態が逆(Lレベルで点灯)になりますので、カソードコモン(LED)かを規格表で確認してください。

この基板の回路図を示します。

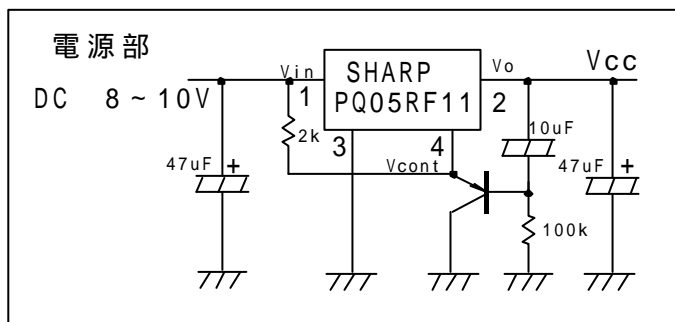


対応する部品のピン配置は以下の通りです。

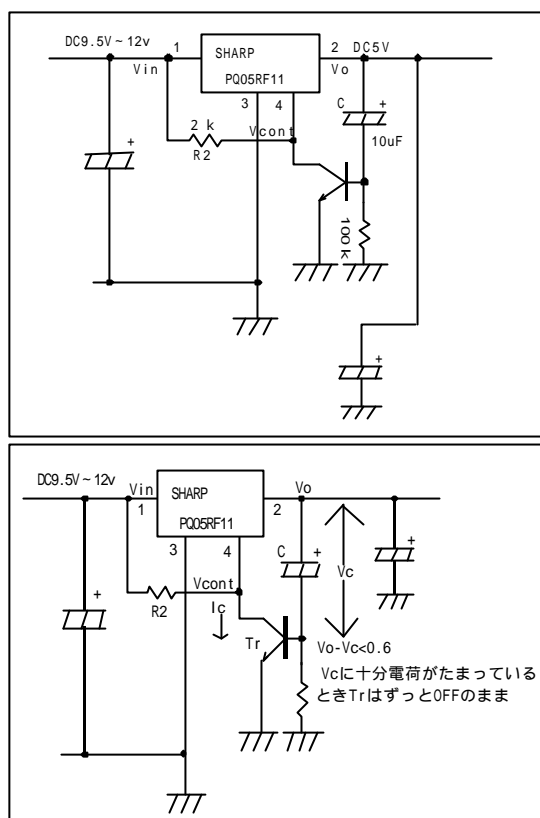


電源部について

AT90S2343 の誤作動を防ぐため 4 端子レギュレータをつけています。
AT90S2343 (デジッ 5/25 現 店 頭 値 5 5 0 円) を 内 部 ク ロ ッ ク モ ー ド で 使 う と き 、 電 池 な ど の 立 ち 上 が り の 早 い 電 源 電 圧 を 使 う 場 合 を 除 い て、



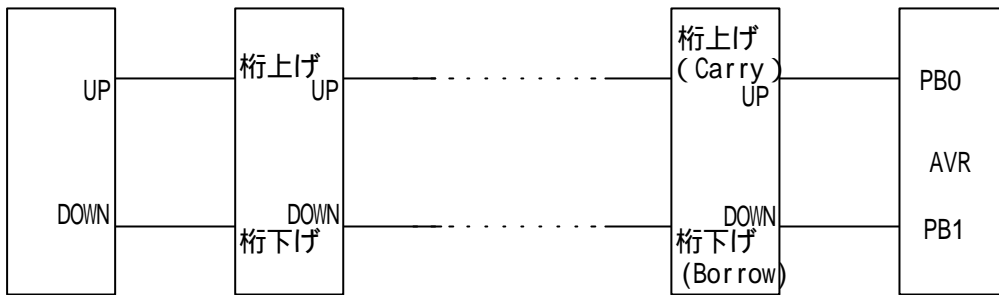
この回路が必要です。立ち上がりが遅い電源を使うと、CPU が動かなくなるだけでなく、リセット自体が効かなくなるようです。しかし、基板のスイッチを入れなおすと、正常に動きますし、水晶を外付けしている場合は正常に動きます。立ち上がりの遅い電源（例えば、AC アダプタでも、中のコンデンサに電荷が入っていない場合）を使うと、CPU が外部クロック動作で動こうとするようです。そのため外付け水晶のない本基板では CPU が動かなくなります。これは、AC アダプタを一旦ショート（100V 電源から抜いていないとアダプタが壊れます。）させて、電源を入れなおすと、この動かなくなる状態が再現できます。根本的にこの問題を取り除くことはできませんでしたが、手動でスイッチを入れなおす操作をすれば正常に動かすことができることから、それを自動的にやってしまう回路を入れてみました。



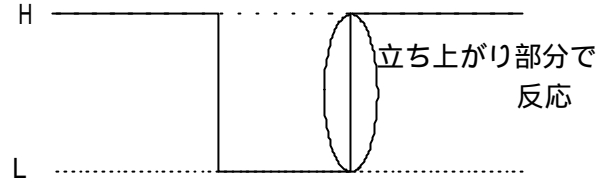
レファレンス端子のついた 5V 4 端子レギュレータ SHARP PQ05RF11（デジット 5/25 現在店頭値 150 円）を使っています。レギュレータへの入力電源がゆっくり立ち上がって、 V_{in} （ $= V_{cont}$ ）が 2V 以上になったとき ON 状態になります。レギュレータが ON 状態になると、コンデンサを通してトランジスタに電流が流れはじめ、 V_{cont} （ $= V_{in} - R_2 * I_c$ ）が下がって（ R_2 の電圧降下 $R_2 * I_c$ が大きくなります。）いきます。 V_{cont} が 2V をきると、コンデンサにいくらか電荷がたまった状態で、レギュレータの出力が OFF（ $= 0V$ ）になります。このとき、トランジスタに逆バイアスがかかるため完全に OFF になります。トランジスタが OFF になると V_{cont} （ $= V_{in}$ ）は立ち上がっている電源と等しくなって、4 端子レギュレータの出力が出始めます。ここで、コンデンサ C に十分電荷がたまっている（上図）と、トランジスタが OFF のままになりトランジスタの ON-OFF 切り替えが起こらなく（図では $V_o - V_c < 0.6$ のとき）なります。トランジスタの切り替え動作はコンデンサにかかる電圧が $(5 - V_{BE})V$ 近くになるまで続きコンデンサにかかる電圧が増えていきます。スイッチングが起こらなくなるころには、電源の電圧が十分上がっている状態で（ $V_{in} > 8v$ ）、最後の切り替えで出力からすばやく 0 から 5V に電源が立ち上がらせることができます。

表示部について

この基板は 2 桁表示にしてあるので、TLR366（5/31 現在 デジット店頭値 150 円）を使いましたが、74HC192 を図のように直列につなげていくことで、桁を目的に応じて増やすことができます。

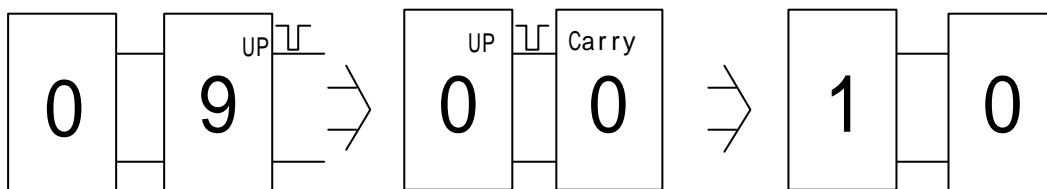


図のように、UP と繰り上げ（一般的には Carry と書いてあります）、DOWN と繰り下げ（一般的には Borrow）をつなぎます。ここで UP(または DOWN)は図のように、常時は H レベルで、L レベル、H レベルの信号を加えて動作させます。74 シリーズ規格表によると、図のパルスの立ち上がり部分でカウントアップ（ダウン）が起こります。



繰り上げ・繰り下げについて

メモリは、値が9になった時、次のカウントで値が0に戻ります。そして、繰り上げのしるしとして、Carry からパルスが生じます。これを次の桁を数えるため次の IC の UP 端子で受けて、上位で繰り上げが起こって、下位の値は0に戻ります。

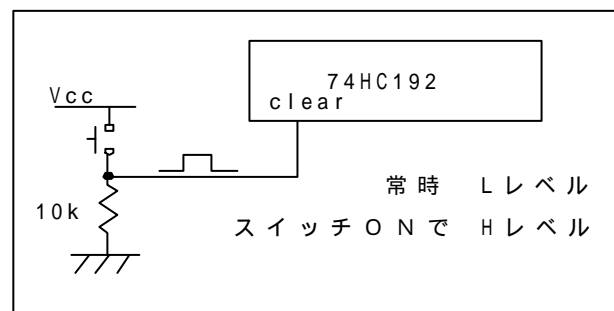


この桁上げ（Carry）端子に次々と UP 端子をつないでいくと、いくらでも桁を増やしていくことができます。桁下げの場合は、IC の値が0になって、次のカウントで DOWN にパルスが加わると、値が0から9へ戻ります。このときに、繰り下げを示すパルスが、Borrow 生じます。それを次の桁の DOWN につなぐと、繰り下げを行うことができます。

7セグメント LED、74HC192、4511 をまとめた基板を企画中です。これを使えば桁数を目的に応じて増やしていくことができます。（発売日は未定）

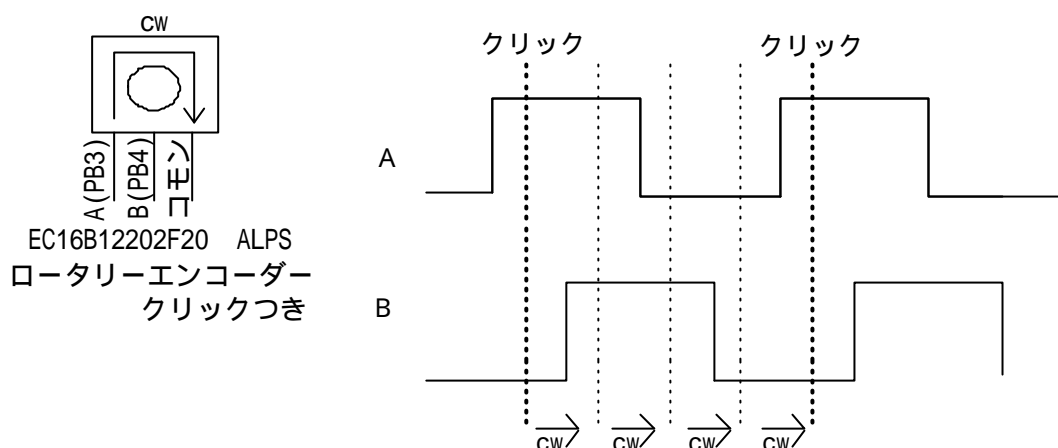
クリア回路について

74HC192 にカウントをリセットするためのクリア回路がついています。図のように、スイッチをONにするとclearにかかる電圧が0Vから5Vになり、スイッチを離すと7セグ LED の表示がすべて0になります。



ロータリーエンコーダーについて

現在使用しているものはクリックつきメカニカルロータリーエンコーダー（EC16B12202F20 ALPS 4パルス/1クリック、接点式：デジットの5/31現在の店頭値150円）です。右に回すと下のようなタイミングでON-OFFを繰り返します。12クリックで1回転します。ABの状態は(A,B) = (H,H) (H,L) (L,H) (L,L)の4とおりありこの変化のパターンを読み取っていきます。

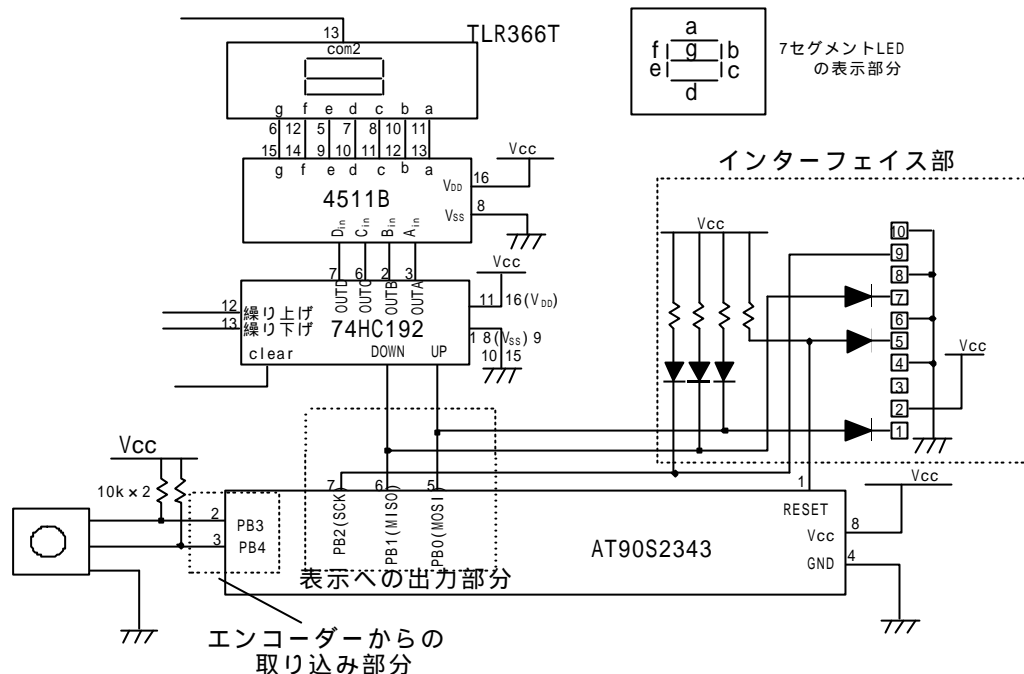


この4パルスを1カウントとして調整してあります。その他には、次のページの表1のような商品も扱っていますので、目的に応じて使い分けてください。例えば、速い回転を扱う場合、今回使ったロータリーエンコーダーは接点式なので、接点式特有のバウンドおよび接点の磨耗があります。バウンドロータリーエンコーダーの回転速度が速いと正しいデータが出せなくなりますので、回転計やメカの回転部分に使う場合に、誤動作のもとになることがあります。接点磨耗に関しては、使用回数がデータブックに載っており、繰り返し回転する所に使うと意外と速く不良になることがあります。しかし、光学式に比べて極端に価格が安い（約1/100程度）ので、それを踏まえて消耗品だと割りきって使うのも一つの方法ではあります。本来接点式は、メカトロに使うというよりは、データを外部からCPUへ取り込む接点としての使用が普通です。本格的メカトロに使い場合は、光学式のロータリーエンコーダー（当店で扱っているエンコーダーは6000rpmまで応答可能）を使うことをお勧めします。現在、当店では取り扱っていませんが、磁気を用いて絶対位置がわかるロータリーエンコーダーがあるという噂もあります。これを使うと、いわゆる角度や位置決めのポテンションメーター（抵抗式）と同じ用途と同時に、角度まで分割できるロータリーエンコーダー等への応用が考えられます。新しい部品を絶えずリサーチすることが、メカトロ制御を成功させる一つの鍵になります。光学式では、バイナリ信号を出力できるロータリーエンコーダーがすでに販売されています。

(表1)

方式	名称	パルス/回転数	特長	5/31 現在デジット 店頭値
接点式	メカニカルロータリーエンコーダ ー(ALPS) EC16B24104F20	24 パルス/回転	縦型・クリックなし	150 円
接点式	メカニカルロータリーエンコーダ ー(ALPS) EC16B12202F20	12 パルス/回転	横型・クリックなし	150 円
接点式	メカニカルロータリーエンコーダ ー(ALPS) EC16B12104F20	12 パルス/回転	縦型・クリックあり	150 円
接点式	メカニカルロータリーエンコーダ ー(ALPS) EC16B12202F20	12 パルス/回転	横型・クリックなし	150 円
接点式	ロータリーエンコーダ RK10J12PO	4 パルス/回転	クリックなし	200 円
接点式	ロータリーエンコーダ RK10J12PO	4 パルス/回転	クリックあり	200 円
光学式	インクリメンタルロータリーエン コーダー(NPNオープンコレク タ) TRD-SH100A	100 パルス/回転	2 相 + 原点付き 6 ホローシャフト 6000rpm まで	11000 円
光学式	インクリメンタルロータリーエン コーダー(NPNオープンコレク タ) TRD-SH10A	10 パルス/回転	2 相 + 原点つき 6 ホローシャフト 6000rpm まで	11000 円
光学式	インクリメンタルロータリーエン コーダー(NPNオープンコレク タ) TRD-S100A	100 パルス/回転	2 相 + 原点付き 6 6000rpm まで	10000 円

ワンチップマイクロコントローラ AT90S2343(5/31 現在 デジット店頭値 550 円)について。
この CPU を使って、回転方向の判定と正転逆転パルスの発生をしています。



周辺との接続は、図のように、エンコーダーからの取り込み部分、74HC192 への出力部分、パソコンとのインターフェイス用回路からなっています。

パソコンから dongle 基板 (別売: dongle 基板 セット特別価格 1000 円 (デジット 5/31 現在)) とこのインターフェイス回路をとおして、AVR にプログラムを入れていきます。

プログラムソースはデジットのホームページ上でダウンロードできます。(rot2343cl.asm) これらを実際には AVRwav.asm,ISP 等のソフトが必要です。(インターネット <http://www.atmel.com> または 2000 年トランジスタ技術 7 月号の付属 CD-ROM から入手できます。)

ソースは以下のようになっています。

```

;*****
;
;*Atmel 1チップマイクロCPU (AT90s2343) によるロータリーエンコーダー表示基板 *
;*ロータリーエンコーダーからのパルス変化を検知して、方向によってUPサインDOWN *
;*サインを出力し、外部のカウンタ74HC192に送る動作をします。 *
;*****
.equ dirc=$60 ;アドレス$60のメモリをパルス方向記憶設定メモリを設定
.equ unit=dirc+1 ;アドレス$61のメモリを分周メモリ設定
.equ un=dirc+2 ;アドレス$62のメモリをカウント出力判定メモリに設定

```

```

.include "2343def.inc";2343のインクルードファイルの呼び出し
.org $000          ;プログラムメモリ$000にrjmp iniをおく
rjmp ini          ;リセット条件が満たされるとrjmp iniが実行される。
                  ;(リセットベクトルの設定)
ini:  ldi r16,$df   ;スタックポインタを内蔵データメモリの一番後ろ
                  ;($df)に設定

out spl,r16
ldi r16,$e7      ;入力(PB3,4),出力(PB0,1)PB2は未使用
out ddrb,r16
ldi r16,$ff     ;カウンタ出力レベルをH(UP)H(DOWN)にしておく
out portb,r16
ldi r16,$01     ;分周カウンタunit初期値$01に設定。
                  ;UpでもDownでも誤動作しない値

sts unit,r16
ldi r16,$00     ;回転方向初期値を(LL-LL)に設定
sts dirc,r16    ;ロータリーエンコーダーがLLから動いていない状態
main: in r16,pinb ;ポートBからのエンコーダーの値を読み取り
andi r16,$18   ;エンコーダーからの入力以外をマスク
lsr r16        ;下位方向への1回シフト
lsr r16        ;下位方向への1回シフト
lsr r16        ;この操作を合わせて(PB3,PB4)->(PB0,PB1)に移される
lds r17,dirc   ;方向記憶の呼び出し
andi r17,$03   ;PB0,PB1以外は0にクリア(前回のパルスだけを残す)
cp r16,r17     ;前回と同じか?
breq main     ;違うパルスが来るまでmainとの間を回りつづける
lsl r17        ;上位へ1回シフト
lsl r17        ;この動作を合わせて、データがdirc2,3ビットへ移動
or r17,r16     ;取り込んだ新しいデータ(dirc0,1ビットに入る予定)
                  ;と合わせる
sts dirc,r17   ;合わせた結果をdircメモリに蓄える。2ビット1組でデー

;タが入っている dirc= (00、00、昔1、今)、
; (昔1、今)のデータを方向判定に使います
rcall rdte    ;方向判定ルーチンの呼び出し
rcall pr4d    ;4分周ルーチンの呼び出し
lds r16,un    ;カウント判定結果を呼び出し$01 = cw(*pr4)

```



```

        cpi r16,$02          ;カウントダウンで$02まできたか？
        brne pccw3         ;来ていなかったら、カウントさせないようにpccw3へ
pccw2:  ldi r16,$02        ;来ていたら、前のルーチンを通して、カウントダウン
        sts un,r16         ;させるサインを出す。
        ret                ;このルーチン呼び出した次の行へ(ここでは*pr4)
pccw3:  ldi r16,$00        ;カウントしないための印un=0をいれる
        sts un,r16         ;unにこの値で戻す
        ret                ;ここをとおるとそのまま何もしないでmainへ戻る
prin2:  ldi r16,$04        ;カウントダウンを循環させる部分。$00になると$04に戻す
        sts unit,r16
        rjmp pccw3         ;そのまま何もしないで(un=0をいれて戻るため)pccw3へ
prcw:   lds r16,unit       ;*****正転カウント部分*****メモリの呼び出し
        inc r16            ;カウントを1増やす
        sts unit,r16       ;unitにその値を戻す
        cpi r16,$05        ;カウント値の端まできたか？
        breq prini        ;来ていたら、priniへ
        cpi r16,$03        ;カウントアップで$03まできたか？
        brne prcw3        ;来ていなかったら、カウントさせないようにprcw3へ
prcw2:  ldi r16,$01        ;来ていたら、前のルーチンを通して、カウントアップ
        sts un,r16         ;させるサインを出す。
        ret                ;このルーチン呼び出した次の行へ(ここでは*pr4)
prcw3:  ldi r16,$00        ;カウントしないための印un=0をいれる
        sts un,r16         ;unにこの値で戻す
        ret                ;ここをとおるとそのまま何もしないでmainへ戻る
prini:  ldi r16,$01        ;カウントダウンを循環させる部分。$05になると$01に戻す
        sts unit,r16
        rjmp prcw3        ;そのまま何もしないで(un=0をいれて戻るため)pccw3へ
;*****ロータリーエンコーダ解読ルーチン*****
;ここでロータリーエンコーダのデータを比較して正転か逆転かを判定します。それ *
;以外のコードが出た場合は、何もしないで帰っていきます。 *
;*****
rdte:   andi r17,$0f       ;上位4ビットを消去
        cpi r17,$01        ;前00 -> 新01
        breq cwe           ;このパターンだったらcw判定へ
        cpi r17,$02        ;前00 -> 新10
        breq ccw          ;このパターンだったらccw判定へ

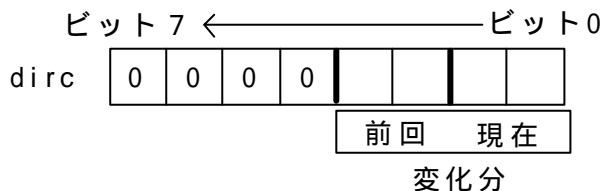
```

```

cpi r17,$07          ;前01 -> 新11
breq cwe             ;このパターンだったらcw判定へ
cpi r17,$04          ;前01 -> 新00
breq ccwe           ;このパターンだったらccw判定へ
cpi r17,$08          ;前10 -> 新00
breq cwe             ;このパターンだったらcw判定へ
cpi r17,$0b         ;前10 -> 新11
breq ccwe           ;このパターンだったらccw判定へ
cpi r17,$0e         ;前11 -> 新10
breq cwe             ;このパターンだったらcw判定へ
cpi r17,$0d         ;前11 -> 新01
breq ccwe           ;このパターンだったらccw判定へ
ret                  ;呼び出した次の行へ戻る (rcall pr4dへ)
cwe: ldi r17,$01     ;cw判定をr17にいれる
ret                  ;呼び出した次の行へ戻る (rcall pr4dへ)
ccwe: ldi r17,$02    ;ccw判定をr17にいれる
ret                  ;呼び出した次の行へ戻る (rcall pr4dへ)
;*****
;

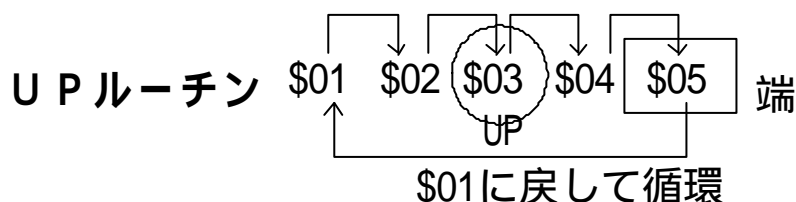
```

構成は、エンコーダ読み取り部 rdte,分周部 pr4d,出力部からなります。
読み取りのために dirc メモリに変化したパターンを図のように蓄えています。

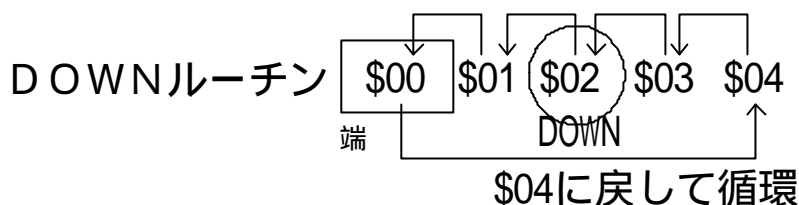


図の右側から新しいデータが入ってきて、全 4 ビットで変化分を表すことができます。cw で 4 とおり、ccw で 4 とおりの全部で 8 とおりを比べて回転方向を判定しています。分周部では r17 に出力された値に応じて 4 分周していきます。

4 分周の方法ですが unit メモリを r17(cw)が来るたびに増やしていきます。これが図のように端まで来ると、\$01 に値が戻され、循環しながらカウントされていきます。



このカウントが4つ分すすめば、カウントUPのためにメモリ un に\$01 をが入ります。ここでは、カウンタ unit=\$03 になるたびごとに、PB0 からカウントアップ信号が出力されます。端 (\$00) にくると図のように\$00 に戻って、カウンタ値が循環します。



カウントダウンの場合ですが r17(ccw)が来るたびに共通の unit メモリが減っていきます。カウンタ unit が\$02 にくるたびごとに、PB1 からカウントダウン信号が出力されます。

出力について

74HC192 の仕様に合わせて出力します。74 シリーズ規格表よると、カウントアップ信号は常時 H 状態になっており、カウントアップ時にパルス (L レベル、H レベル) の信号を検出してカウントが増やすようにできます。プログラム上では、次の命令が対応しています。

```
cwo: (
    ldi r16,$03
    out portb,r16 )
    ldi r16,$02          ;UP端子へLレベル出力
    out portb,r16
    ldi r16,$03          ;ここで、UP端子へ立ち上がり出力を加えたことになる
    out portb,r16
    ret                  ;呼び出された次の行へ戻る。ここでは、(*)に戻るしかない
```

なお、(~) の部分は すでに出力済みなので、必要なく、ソース上では省略してあります。

あとがき

ホンダやソニーが本格的な歩行ができるメカトロニクスが目立ってきたこのごろ、メカトロという次元ではアマチュアの我々が踏み込めそうもないような領域にも感じるかもしれません。あれを見ていると、自作でロボットをつくる意義がなくなるようにも思われますが、このような簡単な技術の積みかさねこそが新しいメカトロ技術のアイデアを、作り出すヒントになっています。また、簡単な装置でも、自分で原理を理解し、実際に作って動いたときの感激はひとしおです。この感動を忘れずに頑張りましょう。

AVR, PIC, メカトロ、電子部品の

ことなら...

〒556-0005 大阪市浪速区日本橋 4-6-7

デジット営業所 (電子部品店)

TEL 06-6644-4555

FAX 06-6644-1744

In need of AVR, PIC, Mechatronics,
Electronic Parts.....

Digit sales department

(an Electronic parts shop)

