

前回の プログラムの不具合が 解決しました。

実は 前回のプログラムにて 正転、逆転の
パルス出力が 正転時 逆転側のポートから
パルスが出て、逆転時 正転側のポートから
パルスが 出ていたのです。 で、すぐに原因
が
分からなかったので さしあたり 前回は 正転
と逆転の処理を差し替える事で対応しました。

で、どこかで、極性の判定処理に不具合が
あるという事で 今回のロータリーエンコーダ
の
ソースを 印刷してじっくり見ていました。
で、不具合箇所を発見しました。

どこに不具合があったかお見せします。
ロータリーエンコーダ スキャンング処理の
ソース rotary_enc_M.h に宣言されていたポート
の ビット位置が 逆になっていました。

R8C/M120A I/O 割り付け			
内容	ポート	Dir	ピン番号
A相パルス 入力	P3_3	In	11
B相パルス 入力	P3_4	In	10
正転パルス出力	P3_5	Out	9
逆転パルス出力	P3_7	Out	2
I2C/SCL	P4_5	Out	12
I2C/SDA	P4_2	In/Out	1

前の動画で見たと思いますが I/Oポートの
一覧です。 この中で A相の パルス入力は
p3_3 です。 B相の パルス入力は P3_4
です。

よって、A相の パルス入力のビット位置は b3
です。 B相の パルス入力のビット位置は b4 で
す。 申し訳ありませんが、このビット位置を
ちょっとの間 覚えておいて下さい。

```

// 定数宣言 ( rotary_enc_M.h )
// -----
#define TAKE_OUT_AB 0x18 // A相 B相 bit 取り出し マスクパターン ( 0001 1000 )
#define AB_PTN_HI 0x18 // A相=H、B相=H のビットパターン ( 0001 1000 )

// #define AL_BH_PTN 0x08 // A相=L、B相=H のビットパターン ( 0000 1000 )
#define AL_BH_PTN 0x10 // A相=L、B相=H のビットパターン ( 0001 0000 )

// #define AH_BL_PTN 0x10 // A相=H、B相=L のビットパターン ( 0001 0000 )
#define AH_BL_PTN 0x08 // A相=H、B相=L のビットパターン ( 0000 1000 )
#define AB_PTN_LOW 0x00 // A相=L、B相=L のビットパターン ( 0000 0000 )

#define FOR_ROT 0xDF // 正転出力 負論理 ( 1101 1111 )
#define REV_ROT 0x7F // 逆転出力 負論理 ( 0111 1111 )

#define HI_ROT 0xA0 // 正転 逆転共に Hi ( 1010 0000 定常状態 )

```

間違い箇所 1

訂正箇所 1

間違い箇所 2

訂正箇所 2

よって、AL_BH_PTN と AH_BL_PTNが 入れ違った状態に なっていた事が 原因でした。
 わかってみれば、なあんだ。という感じですね。理論通り動くようになり 安心しました。
 ついで、ですから このプログラムの Cソースを お見せしましょう。

```

//*****
//**   ロータリーエンコーダ メカニカル式 スキャン処理   ** ( rotary_enc_M.c   の   一部 )
//**   -----   **
//**   関数値： 0 = A相 B相パルス   変化なし   **
//**           1 = A相 B相パルス   変化有り   **
//*****
BYTE rotary_m_scan( void )    // 使用法： 2ms周期で   呼び出し続ける
{
    BYTE sts, phn;

    sts = 0;                      // 状態変化フラグ 初期化
    P3_buf = p3;                  // Port3の状態を   byte単位で   一括取り込み
    p3 = P3_buf | HI_ROT;         // 正転 逆転共に Hi ( 1010 0000 定常状態 )に   戻す

    Pab = P3_buf & TAKE_OUT_AB;   // A相 B相の信号だけ   取り出し
    if( Sw == 0 )                 // 起動時   一つ前の履歴がないため、初回のみ処理
    {
        Sw = 1;
        Pab_e = Pab;              // 一つ前の履歴   初期設定
        return 0;
    }
}

```

このソースは 続きが あります。

ロータリーエンコーダの応用 1、メトロノーム

ロータリーエンコーダの応用例として 簡単なメトロノームを作ってみようと思います。

遥か昔、メトロノームというと右の様な四角錐の形状を思い浮かべます。今はもう売って無い。と思っていたら結構ありましたね。今は手のひらにちょこんと載せられる小さい電子機器のメトロノームが低価格で主流のようです。



であれば、作らずに買った方が安くて早いんじゃないかと言われそうですが、私は物好きな人間なので、自分で作るのが好きなのです。

それはさておき、今回作成するメトロノームの仕様を検討します。

で、中にはメトロノームは、音楽で使われる事は分かるけど、具体的に何をする物かは分からない。！という方ももしかしたらおられるかもしれません。音楽の拍子というか、ビートを正確に刻む道具です。左のメカ式の物は中央にある縦に配置された金属棒に重しが付いています。その重しを上には上げれば拍子は遅くなります。重しを下に下げると早くなります。

重しの左右に目盛が付いていますが、その数値が早さの度合です。この速さの単位は **BPM** です。 **Beats Per Minute** の略です。某楽器店の説明では1分あたりの四分音符の数を指す。との事です。で、左のメカ式のメトロノームの場合は **BPM** の設定範囲が **40 ~ 208** です。電子式の小型のものは **30 ~ 250** です。今回は **30 ~ 250** の設定範囲で作ります。

で、このBPMの設定と表示に ロータリーエンコーダと 超小型の 液晶表示器を 使う予定です。 使用するマイコンは、ディスコン品で申し訳ありませんが R8C/M120Aを使います。

R8Cマイコンは 数年前にまとめて買っていたので M120A、M110A共に まだ 20個ぐらい在庫があると思います。秋月電子で 最後 無くなるのが異様に早かったので、私以外にも 過去に まとめ買いされた方が 結構いるのではと思います。持っておられる方は 是非 R8Cマイコンで 何か作って下さい。

で、今回作成するメトロノームは 一時的な実験では無くて 百均で買ったプラケースに入れて その後も使えるようにします。あと、ボタンを押したらインクリメントする カウンターの機能も付けたいな。 と思います。

で、カウンターは 何の用途に使うかという楽器(シンセ)の上に ちょこんと載せて 練習した回数をカウントするのに使いたいのです。で、その場合 カウントアップする押しボタンでチャタリングが起こると 1回押したつもりが 数回カウント値が上がる状態では 使い物にならないので、可能な限りチャタリングが 発生しないようにしたいと思います。 それと、カウンターには リセットボタンも 必要ですね。

メトロノーム側では BPM設定の ロータリーエンコーダと スタート、ストップの 押しボタンスイッチを 付けようと思います。 という事で カウンターの押しボタンと、メトロノームの スタート、ストップの 押しボタンは 共用しようと思います。 よって モード切り替えのトグルスイッチを付けて メトロノームと カウンターの機能を 切り替えるようにしようと思います。

あと、電源は 単三乾電池3本で動かす事にします。 R8C/M120Aマイコンは 20MHz動作にて 2.7 ~ 5.5Vで 動作する事になっていますので 単三電池2本でも動作しますが 圧電ブザーが 5V動作でも音が小さいので 電圧をあまり下げたくなかったのです。という事で単三乾電池3本という事にしました。で、今回採用する百均のプラケースに 黒い単三電池3本の電池ケースも ちょうど収まりました。この電池ケースには スライド式の 電源スイッチも付いています。という事で 実装部品が ある程度 決まってきました。

百均プラケース	x1
単三3本 電池ケース SW付き	x1
メカ式ロータリーエンコーダ	x1
ツマミ	x1
I2C 8x2 LCD	x1

圧電ブザー	x1
押しボタン 3P スイッチ	x1
ボタン付き タクトスイッチ	x1
小型 3P トグルスイッチ	x1

下の画像は、説明の必要は無いと思いますが 部品の番号を 画像の部品近くに 付けました。ちょっと、間をおいて 次のページに行きます。

