

今回は、最初に動画をお見せします。

一応、今回の自作メトロノームが、完成しました。 今回は不可解なトラブルに 悩まされジタバタしました。 土壇場で出来たので 急遽 タイトルを「自作メトロノーム出来ました」に、変更しました。

当初 プログラムロジックのバグかと思ってましたが、どう考えても 問題ない ということで悩んでました。 で、どこで問題が 発生しているのかというと、ロータリーエンコーダーの信号読み取り部分です。

結果から報告すると 今回のメカ接点式ロータリーエンコーダですが、最初 使い始めはほぼ 理論どおりの信号が出て、非常に使いやすかったのですが、

経時変化と共に、徐々に チャタリングが 出始めるようになったのです。 チャタリングと気付くまでに、時間が かかってしまったという事です。

動画の後に、詳細を説明します。

トラブルに至った経緯、原因、処置

今回、どうせなら ロータリーエンコーダのスキャニング処理を アセンブラーのタイマー割り込み処理内に組み込もうとした事が、話がややこしくなったキッカケでした。

インターバルタイマーの割り込み周期は 1ms です。ここに、前回の C 言語で 記述された `rotary_enc_M.c` と `rotary_enc_M.h` が あり メインループ内で `rotary_m_scan(void);` を 呼び出す仕様になってました。

で、この `rotary_enc_M.c` と `rotary_enc_M.h` を元に インターバルタイマーの 割り込み処理 内にて `IOCS_R_enc_inTimer.inc` を インク ルードする事により、ロータリーエンコーダの機能を インターバルタイマー処理に 追加し ようと していました。

最初は 細々としたバグも ありましたが、殆ど取り去り、ロータリーエンコーダの信号が 取れるはずなのに取れない。

で、1秒間に 1000回呼び出されるタイマー割り込み処理内なので デバッグが 極めてやりにくいのです。細かいところを少しずつ変えてみましたが 一向に進展せず 3日過ぎてしまいました。

4日目に 方針転換で メインループから呼び出す C 言語のスキャニングルーチンに戻してみました。一応 動き出したのですが、動作が不安定です。前、ちゃんと動いてたのに何で？ で、ロータリーエンコーダの 経時変化で、チャタリングが 出ているのではないかと 推測した次第です。

アセンブラーの関数で うまく信号を取り込めなかつた原因は、処理が 早すぎて、チャタリング中の信号を 取り込んで 判定が正常に出来なかつたと思われます。

で、今回は 一つ前の A B相の信号と 今回の A B相の 変化を検出した際に 次の A相が 進んでいるのか、B相が 進んでいるのか判定する次の if 文との間に 若干の Wait処理を 入れて チャタリングが 収まるのを待つようにしました。これにより、ほぼ まともに動くようになりました。

一つ、ロータリーエンコーダで 厄介なのは Wait処理を入れ過ぎると ロータリーエンコーダの軸を 高速で回したときに 処理が 間に合わなくなる可能性もあります。 ということで Wait時間の さじ加減は 調整が必要です。

C言語の ソース [rotary_enc_M.c](#) のどこに Wait処理を入れたか、ソースを次のページでお見せします。

因みに Wait処理は 以下の関数です。

```
*****  
/** 100 μsの 固定 時間待ち **  
*****  
void delay_100micro( void )  
{  
    WORD i;  
  
    for( i=0; i<88; i++ );  
        // 20MHzで 約 100[us]  
}
```

rotary_enc_M.c 内の rotary_m_scan(void) 関数内の 一部です。赤色が Wait処理です。

```
if( Pab != Pab_e )                                // 最新の状態と 一つ前の状態が 食い違った場合
{
    delay_100micro();                            // 若干の Wait( 100us )
    delay_100micro();                            // 若干の Wait( 100us )
    if( Pab_e == AB_PTN_HI )                    // 一つ前が P0 ( A=H B=H )
    {
        delay_100micro();                            // 若干の Wait( 100us )
        if( Pab == AL_BH_PTN )                    // P0 -> P1 ( A=L B=H ) で正転
        {
            if( Cnt < MAX_CNT )      Cnt++;    // カウンタ インクリメント
        }
        if( Pab == AH_BL_PTN )                    // P0 -> P1 ( A=H B=L ) で逆転
        {
            if( Cnt > MIN_CNT )      Cnt--;    // カウンタ デクリメント
        }
    }
    if( Pab != Pab_e ) sts = 1;      // 最新の情報で 一つ前の情報と 食い違いがある事を確認。
    Pab_e = Pab;                                // 一つ前の履歴更新
```