

## R8Cマイコンで、モールス信号の受信処理

R8Cマイコンで、モールス信号の受信解読を行い ASCIIコードにして、シリアル通信でパソコンに出力します。

と、前回も 公言してたのですが、**最終的に出来たのだろうか。?** と 気になっておられる方も いると思うので、最初に結果を発表します。

**一応、出来ました。** ということです。  
**ほっとした、** というのが正直なところです。

**一応**という言葉が付いているのは、**電源ON直後の、最初の 1文字受信を失敗します。**

これは、モールス信号の長点または 短点を識別する判定値が、確立していない状態で 入つてくるので、**判定出来ない**のです。最初の 1文字が入って来る事で、その文字のパルス幅を計測して 入力信号に同期する形で 判定値が、確立します。その関係で、2文字目以降が 正常に受信出来るようになります。

中には何故、最初から長点、短点の 判定値を入れておかないので。と思われる方も おられるかもしれません。モールス信号の長点、短点とは、パルス信号の 短い時間幅が、3対1の比率で、長点、短点を識別出来るのです。これは、あくまで**比率**であって、**絶対的な時間幅は、規定されていません**。実際の運用を考えるとオペレータが 電鍵を使って手動でモールスを 送信していた訳で、**人により速めだったり、遅かったり**すると思います。よって**送信側の、モールス信号の速度に追従する形で、受信する必要がある**という事です。 という事で送信側のモールス信号速度が、速かったり 遅かったりするので、長点、短点の 判定値も 受信する信号に合わせて調整する必要が、あるのです。データを受信しながら、判定値も、毎回更新し続けているので途中で モールス信号が、だんだん早くなったり遅くなったりしても、追従します。

## 今回の実験環境の構成と概要

左側: 送信用PC 専用アプリにて 文字をシリアル通信でマイコンに転送

右側: 受信用PC テラタームにて シリアル受信文字を 表示

パソコン本体は  
机の左下に  
置いてます。

左送信側と 右受信側の  
マイコンを載せたブレッドボード

送信側マイコンは、PCから  
受けた文字データを モール  
ス信号で 送信する。

受信側マイコンは、送られてきた  
モールス信号を 解読した文字を  
シリアル通信で PCに転送する。

## ブレッドボードの拡大



左送信側と 右受信側の  
マイコンを載せたブレッドボード

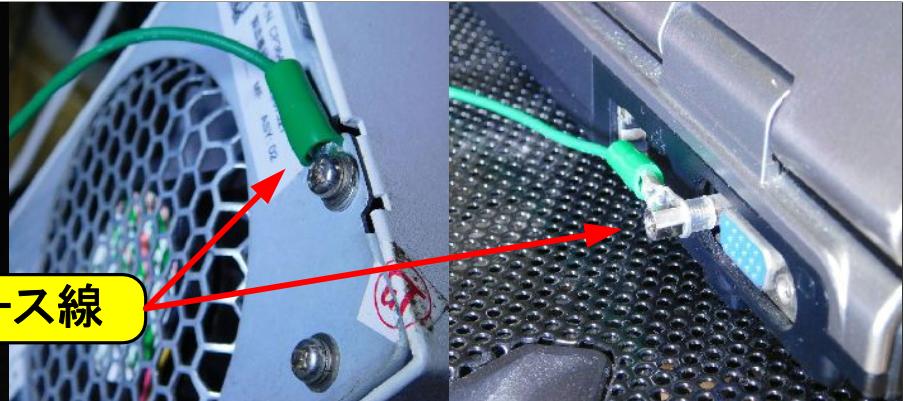
アース線

左の小型デスクトップPCと 右のノートPC それぞれの USB ケーブルから電源をもらい、2つの R8Cマイコンが、ブレッドボード上で **グランドラインを 共有接続**しています。

2台のPC間に電位差がある事も考えられるので、安全のため それぞれのPCに **アース線を 最初に接続**して下さい。

送信側

受信側



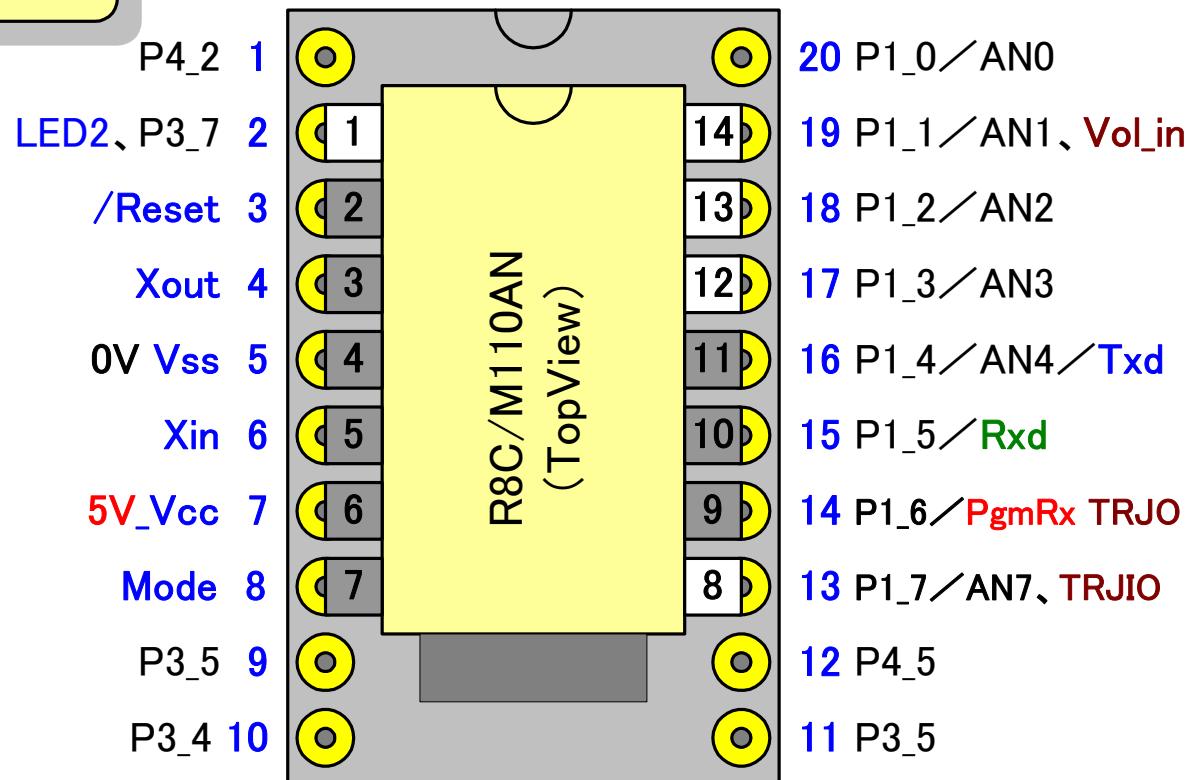
## 2枚の R8C/Mマイコン間の接続

R8C/Mシリーズのマイコンの小基板を使用していますので、接続は、小基板のブレッドボードに挿す 20ピンの番号で表示します。(右図の 20ピンICソケット外側の 青い番号です。)

ちなみに、今回送信側に R8C/M110A を 使用して、受信側に R8C/M120Aを 使用しています。R8C/M120Aマイコンは、20ピンなので 小基板のピン番号と一致します。

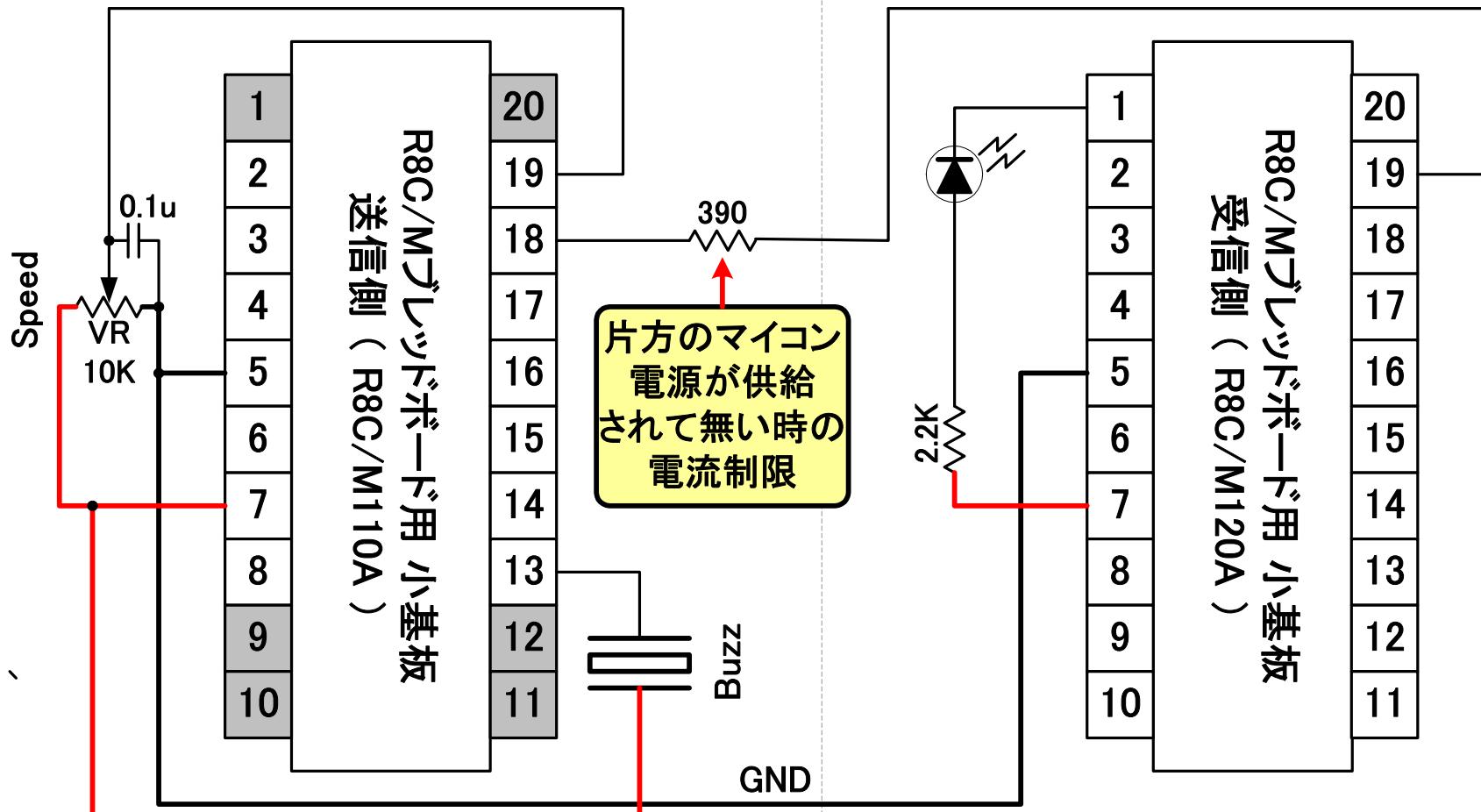
R8C/M110Aの場合は、14ピンなので、ピン番号が、小基板のピン番号と一致しませんので 気を付けて下さい。

尚、CP2102基板モジュールを使用したUSB接続なので、電源は パソコンから供給されます。シリアル通信も 小基板上で、CP2102と接続されています。



よって、それ以外の部分を ブレッドボード上で、配線すれば、いい事になります。

## R8C/Mマイコン間の 結線図



こうやって見ると、配線 結構少ないですね。  
小基板内に、定型的な回路を入れ込んでもるから  
でしょうね。

## モールス信号解読 ASCII文字変換の概要

この処理は、細かく話をすると長くなるので概要の説明を 箇条書き的に示します。

- ① モールス信号を、取り込むサンプルレイトは、5ms 間隔、200Hzサンプリングです。
- ② 入力端子から読み出したモールス信号を連続 3bit一致による デジタルフィルターを構成します。
- ③ 次に、2bitの信号履歴を持ち 01で 立ち上がり検出、10 で 立ち下がり検出を行います。 11、00 は 同じレベルが継続しているので、エッジ検出するまで、カウンタでサンプルの カウントをします。 エッジ検出したら、カウント値を、次の別の変数に保存し、この値を、次の処理に渡します。これが、モールスパルスの、Hi側、Low側のパルス幅となります。 ここまで処理は 5ms タイマー割り込み処理内で行います。

③で、作成したパルス幅データですが、R8C/Mマイコンは、RAM容量が小さいため、Byte値で表現しています。 そしてパルス幅のカウント値だけでなく、Hiレベル側のパルス幅か、Lowレベル側の パルス幅かを、判別する機能も 持たせる事にしました。 最上位 bitを 符号ビットにして正の値の場合 Hiレベル側のパルス幅、負の値の場合 Lowレベル側のパルス幅としました。

Byte変数という事で、-128 ~ 127の値を取る整数値となります。 0 は 無効データとして扱います。 モールスの 短点1と、長点3は、正のデータのパルス幅で、判断します。 負のパルス幅は、1文字内の 点と 点の間の隙間は、短点1個幅で、文字間の 隙間は、短点3つ分の長さで、語と 語の間の隙間は 短点7つ分の 隙間となります。 よって隙間のカウントで、文字間、語間を 判断する事が 出来ます。

説明は、このあたりにして、実験の動画をお見せします。