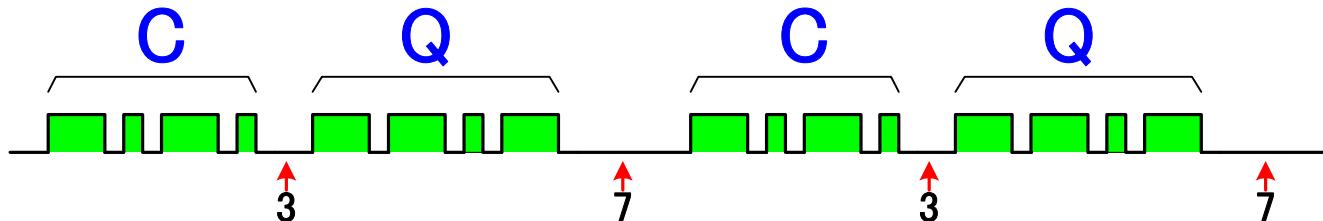


## モールス信号について

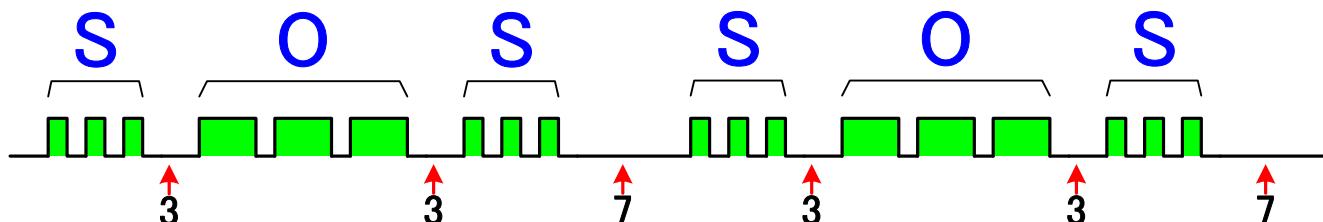
受信の話をする前に、モールス信号の復習を ちょっと やります  
ね。CQ CQ の 信号について 見てみます。



CQ は、アマチュア無線のコールサインで、通信への参加を求める呼びかけ(挨拶)の言葉として 使われているそうです。

よって CQで一つの語で、C と Q の間は、文字間を 意味する  
短点3個分の隙間を 開けます。 CQ と CQ の 間は、語と語の  
間になるので、短点7個分の隙間を 開けます。

ちなみに、 SOS SOS は、以下のようにになります。



短点の 時間的長さを 1 と  
すると、長点は 3 となりま  
す。

1つの文字内の、短点と 長点  
の 間の隙間(短点、短点間  
または、長点、長点間の 隙間  
の場合もあります。)は 短点  
1個分の 時間的長さの 隙間  
になります。

## R8Cマイコン モールス信号の受信

モールス信号 送信の場合は、送信側CPUのクロックを分周して任意の速度を作りだしますが、受信側となると 相手側（送信側）のモールス信号の タイミングに合わせて受信しなければなりません。モールス信号の短点より 数倍早いサンプリング速度で、相手のモールス信号を取り込み、短点と長点の長さを 認識します。

その前に、ちょっと サンプリングの事に関して説明します。通常 デジタルの世界では、アナログの信号を 定周期のサンプリング周波数で、その一瞬のアナログ信号の サンプルを取り込む事を サンプリングと表現します。

身近にある物として、音楽用CD コンパクトディスクが あります。

音楽用CDは、サンプリング周波数 44.1KHzで、16bitのアナログ量子化信号を L R の 2チャネルで 記録されています。

サンプリング周波数 44.1KHzは、人間の耳に聞こえる音の周波数は、凡そ 20Hz ~ 20KHzと定義されています。サンプリング定理では、サンプリングを行える上限周波数は サンプリング周波数の 1/2以下となっています。

よって、人間に聞こえる上限周波数の 20KHz の 2倍で 40KHz以上の サンプリング周波数が必要 という事で、別の事情もあって 44.1KHz になったと思われます。DATとかは、48KHzですから、何故 近い周波数に2つ周波数の規格があるのかが 分かりません。

DVDオーディオは 96KHzで 48KHzの倍ですね。また、SACDは DSDとかいう異なる 記録形式になってますよね。サンプリング周波数が決まっている物は、PCM方式とか呼ぶようです。

## 前回作成したモールス送信側の改造

今回、モールス信号の受信を行うテスト信号として、前回作成したR8C/Mマイコンによる、モールス信号送信機能を使用します。

で、前回のモールス送信機能で不自由に思えた事が、あって変更、改造を行いました。

まず、前回テラタームで、モールス送信データを、R8C/Mマイコンに転送していましたが、**1行分単位で、都度テラタームにデータを手入力しなければならない事。** これは、間違えずに手入力するのも大変ですが、**連続してデータを出し続ける事が出来ない事。**

これは、受信側で信号観測する際に、連続して信号を出し続けたいのです。

1行毎に止まって、次のデータを手入力するのは、面倒です。

逆に、ずっと連続でデータを出し続けていると、**止めた時に、すぐに停止してくれない。**

という事もあります。マイコン側に、1行分まとめてデータ転送しているので、その一行分を、吐き出さないと停止してくれません。

という事で、テラタームでは無くて、**専用の通信ユーティリティを作成する事にしました。** これが時間を食ってしまいました。作成する事により連続モールス信号を作成する事は、**可能となりました。**

次に、モールス信号出力中に、中斷する機能を付ける事。これは、**パソコン側ユーティリティとR8C/Mマイコンの両方に改造が必要でした。**

中斷は、**パソコン側から、[ctrl]-C 03h** をマイコン側に 送る事で、中斷出来るようにしました。

が、奇妙な現象が起きて、1回目は 中断する事が、出来て2回目以降が 中断出来ないという現象が 起きました。

パソコン側で、[ctrl]-C を 出している箇所で ビープ音を 鳴らしました。 受け側の R8C/M マイコンでも、[ctrl]-C を 受信した直後に ビープ音を 出すようにしました。

1回目の中断ボタンで、パソコン側と、R8C/M マイコン側で 同時にビープ音が 鳴ってモールス出力は中断出来ました。

2回目は、パソコン側だけで、ビープ音が 鳴って、R8C/M側では、ビープ音が鳴りませんでした。 当然、中断も出来ませんでした。

何で こうなるの。?

パソコン側のソースを見ても、マイコン側のソースを見ても、それらしい現象につながるところは、見つかりません。

しばらく悩みました。 パソコンの COMポート出力と、マイコンのシリアルポート入力の間にあるのは、CP2102 USB-シリアル変換モジュールと パソコン側に、CP2102用デバイスドライバが、あるだけです。

もしかして CP2102に関わる部分で、トラブルが 起きているのだろうかと思い、試しに R8C/Mマイコンでモールス出力中に、中断ボタンをクリックして、R8C/Mマイコンに [ctrl]-C を 送り、その後に、一旦 COMポートを クローズして、再オープンして、R8C/Mマイコンでモールス出力中に、再度、中断ボタンをクリックしたら、今度は 正常に 中断出来ました。

やった事は、COMポートのクローズ、再オープンです。 この、クローズ、再オープンを、行う事で、多分 CP2102を 再初期化しているのではと思われます。

では、どのような場合に、動作が、おかしくなるのか考えてみると、CP2102の通信は 通常は 行儀よく 半二重通信をやっているのです。

ところが、モールス信号出力中は、出力文字のエコーバックを パソコンに送っているので、そのタイミングで、[ctrl]-C 1byte送るのは ごく一瞬ではありますが、全二重通信をやった事になります。

最近 見なくなってきてますが 元々パソコン本体に付いている コムポートの 9pinコネクタは、SDラインと RDラインが 独立しているので 全二重通信が可能なインターフェースなのです。

USBは、パソコンが マスタになって、各周辺機器に ポーリングを行い通信を確立するプロトコルを持っているので、全二重通信という概念が、無いと思います。

USB-シリアルモジュールを作っているメーカーでは、何とか 全二重が出来るように頑張っているのかもしれません、マスタとなるパソコンがサポートしていないから 難しいでしょうね。

今回の現象は、USB-シリアルモジュールでは 全二重通信は、やらない方がいい。 という事かもしれませんね。

それと、今回のモールスの信号は、デジタルなので、A/Dコンバータは 使用しませんが、定周期で、デジタル信号を サンプリングします。

その サンプリング周波数を 凡そ見つけ出す事が 出来ました。 前回作成したモールス送信側プログラムで、出力出来る 最高速のモールス信号で、確認しました。

- ① サンプルレイト 10msで モールス短点が  
3~4 サンプルでした。
- ② サンプルレイト 5msで モールス短点が  
7~8 サンプルでした。
- ③ サンプルレイト 2msで モールス短点が  
17~18 サンプルでした。

①は、ちょっと ブレが大きい気がするので  
②か、③にしようと 思います。  
多分、②の 5ms サンプリングで 良さそうな  
気がします。

後は、今回作成した送信側ユーティリティを  
お見せします。