ＧＫＲＸ７＿１Ｍ　マイコン学習セット　応用編

初版　２０１８．１．２６

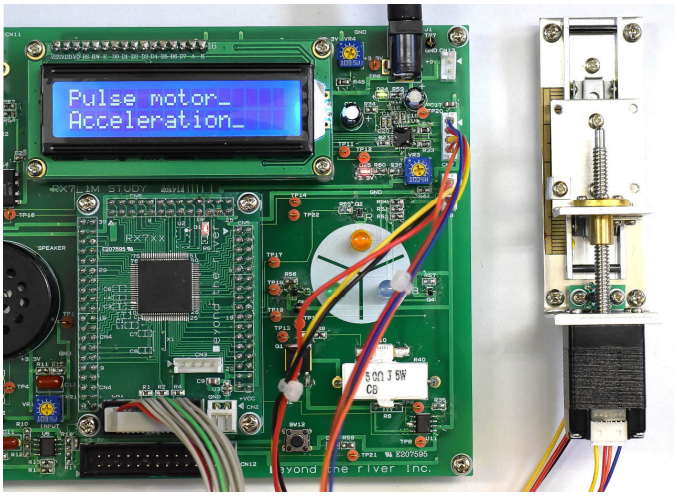
【　製品概要　】

本マニュアルはマイコンの学習を行うためのハードウエア、ソフトウエアのセットの応用編です。

応用編では入門編で学習した基礎を基に実用的なプログラム例を用意しました。

ソフトウエアはルネサス社の無償統合開発環境ＣＳ＋　ｆｏｒ　ＣＣです。サンプルプログラムはＣ言語で書かれています。めんどくさい初期設定は「コード生成」機能で簡単に行え、２４０ＭＨｚクロック、ＦＰＵ内蔵で驚異の演算速度を体験出来ます。

※本学習ボード開発にはルネサスエレクトロニクス社製Ｅ１が別途必要です。

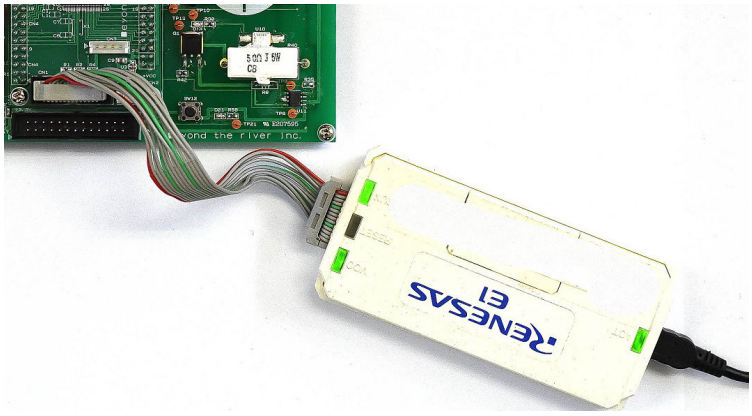


３．サンプルプログラム

３－１．　ｓａｍｐｌｅ３１　声の大きさでＬＥＤの明るさを変える
３－２．　ｓａｍｐｌｅ３２　比例帯温度制御
３－３．　ｓａｍｐｌｅ３３　メモリ使用音声録音再生器
３－４．　ｓａｍｐｌｅ３４　声でパルスモータを動かす　ヒューマンインターフェイス
３－５．　ｓａｍｐｌｅ３５　加速させてパルスモータを動作させる

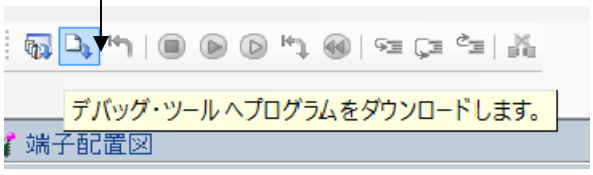
３．サンプルプログラム

初めに、ＣＮ１にＥ１のケーブルを差して下さい。

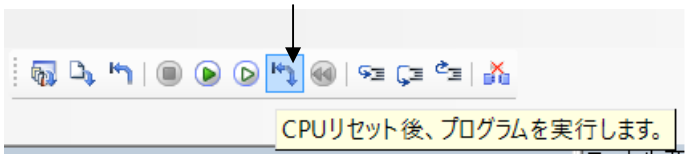


次に、電源の添付のＡＣアダプターをＪ１に差して下さい。ＡＣアダプタをＡＣ１００Ｖ　コンセントに挿入し、ＬＥＤ　Ｄ２４　が緑（９Ｖ）、ＬＥＤ　Ｄ２５が赤（３．３Ｖ）に点灯すれば正常です。

サンプルプログラムは全てコンパイル、動作確認済みです。、「デバックツールへプログラムをダウンロード」後



ＣＰＵリセット後、実行で動作します。

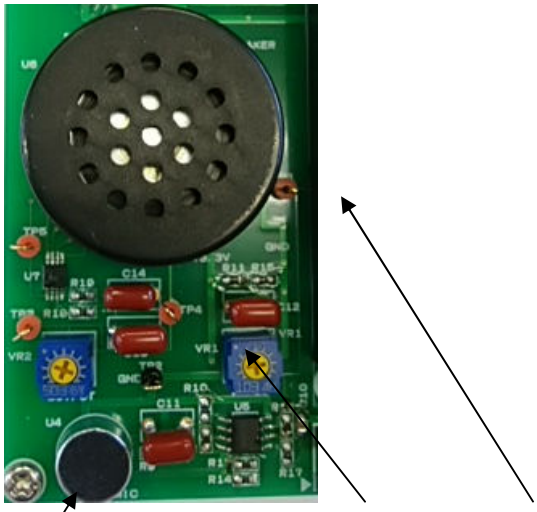


コネクタの抜き差しを行うときは必ずＡＣアダプタのＡＣプラグを抜いて、ＬＥＤ　Ｄ２４，Ｄ２５が消灯してから行って下さい。

３－１　ｓａｍｐｌｅ３１　声の大きさでＬＥＤの明るさを変える

【　概要　】

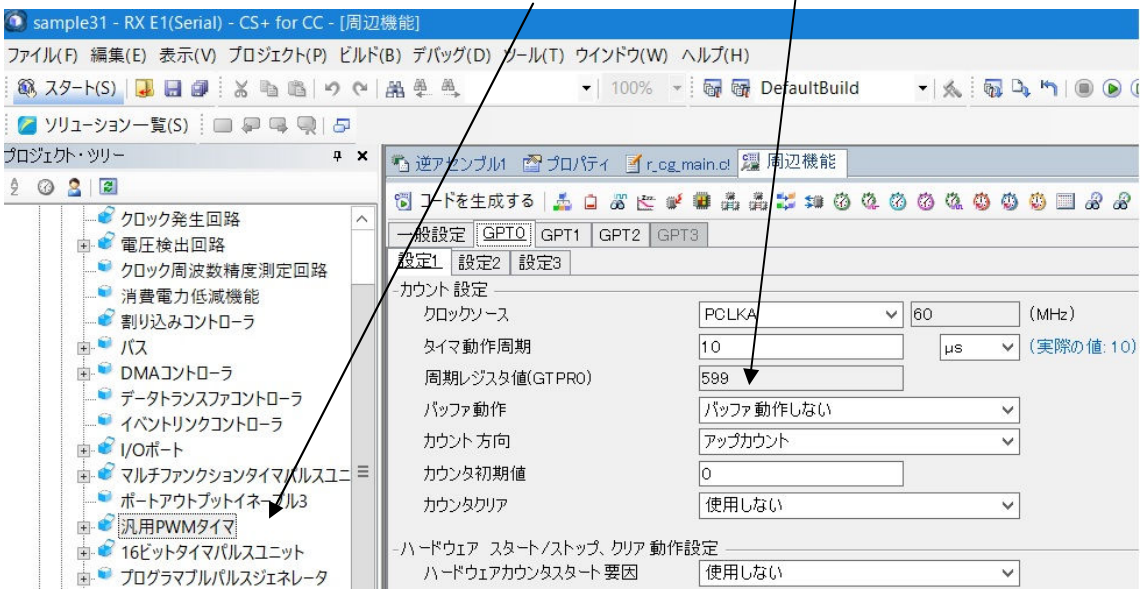
ＭＩＣに向かって話すことにより、ＬＥＤの明るさを変えるプログラムです。



マイク　　音量調整ボリウムＶＲ１　　波形観測用　ＴＰ１

【周辺機能の説明】

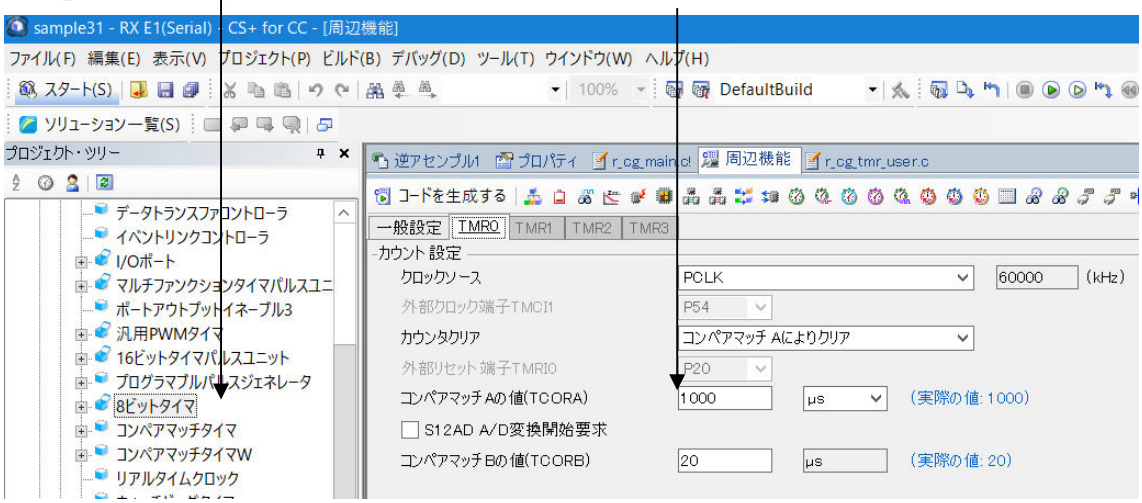
ＬＥＤの明るさを変えるのに５９９分解能のＰＷＭを使用しています。



声をコンデンサマイクで電気信号に変え、オペアンプで１１倍×１１倍＝１２１倍拡大しています。それをＡ／Ｄコンバータに出力し、Ａ／Ｄ値を得ています。

声が無いときは３．３Ｖ／２　≒　１．６５Ｖ前後の電圧が液晶に表示されます。声を上げて行くと最大３．３Ｖ付近まで電圧が上がりＬＥＤが明るく点灯します。５秒間ごとに保持電圧がクリアされ、新たに声を待ち受ける状態になります。

時間を見るために８ビットタイマ（１６ビットモード）で１ｍｓｅｃ（１０００ｕｓｅ）毎に定周期割り込みを入れています。



【　主要プログラム　】

省略

【　解説　】

①　　int_timer = 10;　　//1msec*10=10msec毎にこのプログラムを通る

ｉｎｉ＿ｔｉｍｅｒに１０をセットし、それが０になるたびに１回プログラムが動作します。

②　　S12AD.ADCSR.BIT.ADST = 1;　　//ADスタート

while(S12AD.ADCSR.BIT.ADST)　　//変換終了待ち

;

ad0 = S12AD.ADDR1;　　//AD1値

１０ｍｓｅｃごとにＡ／Ｄデータをサンプリングしています。

③　　fdata1 = ad0/(4095/3.3);　　//データ→電圧Ｖ換算

Ａ／Ｄ値を電圧値に変換しています。電圧値＝ａｄ０／（４０９５／３．３Ｖ）

④　　if(fdata1 > fdata2)　　{

今回のデータが前のデータより大きいときだけ以下のプログラムを実行します。

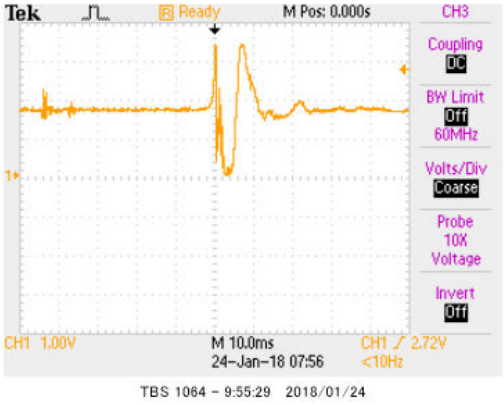
⑤　　fdata3 = fdata1 - fdata2;　　//差を見る　計算上1.65V～3.3Vまで動く

差をｆａｄａｔ３にセット。

//周期レジスタ599/1.65V≒363

fdata3 *= 363;　　//

電圧として変動する幅は３．３Ｖ－１．６５Ｖ＝１．６５Ｖあります。その変動分をＰＷＭの周期レジスタ最大５９９に換算します。一番電圧が大きい＝一番声が大きい＝ＬＥＤが一番明るくなる、となります。



ＴＰ１の波形観測例　大きな声だとＡＤ００１の入力は０Ｖ～３．３Ｖまで変動します。

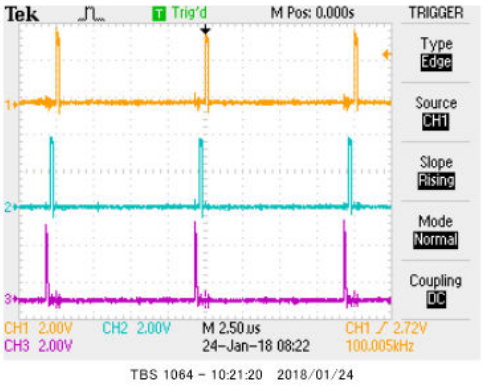
if(fdata3 < 10){fdata3 = 10;}　　//下限設定

if(fdata3 > 599){fdata3 = 10;}　　//大きすぎるデータは使わない

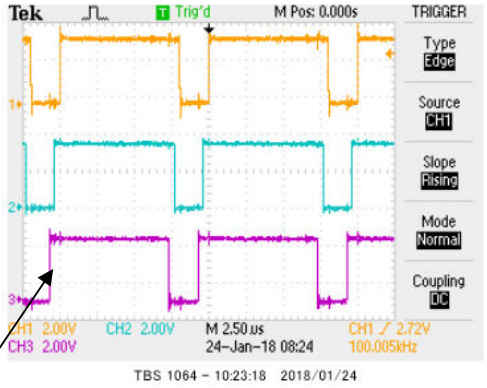
上下限のリミットを付けています。

アウトプットコンペアレジスタに換算値を設定しています。数が大きいほどＬＥＤは明るくなります。

```
GPT0.GTCCRA = fdata3;                    //演算値設定
GPT1.GTCCRA = fdata3;                    //
GPT2.GTCCRA = fdata3;                    //

fdata2 = fdata1;                         //大きい方を残す
timer = 0;                               //残した時にタイマークリア
}
```



ＧＴＣＣＲＡが小さいときのＰＷＭ出力波形
ＴＰ１７，１８，１９



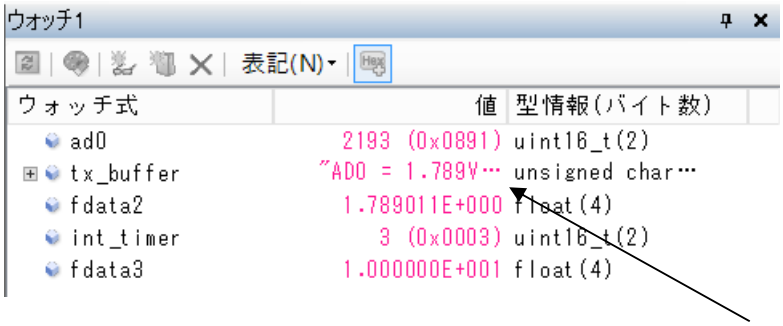
ＧＴＣＣＲＡが大きいときのＰＷＭ出力波形

トランジスタのベースドライブ時間が長いので、ＬＥＤが明るくなります。

⑥

```
sprintf(tx_buffer,"AD0 = %.3fV    Timer = %4d¥n¥r",fdata2,timer);
                                         //液晶表示の為の変換
lcd_disp_0();                            //液晶表示
```

液晶に電圧値とタイマーを表示させています。



ｔｘ＿ｂｕｆｆｅｒに変換した値はウオッチ窓でも確認できます。

```
⑦                       if(timer > 500)                                 //10msec*500＝5S
                        {
                                timer = 0;
                                fdata2 = 0;
                                fdata3 = 10;
                                GPT0.GTCCRA = fdata3;                   //初期化
                                GPT1.GTCCRA = fdata3;                   //
                                GPT2.GTCCRA = fdata3;                   //
                        }
```

５秒経過するとデータを初期化しています。

定周期割り込み　r_cg_tmr_user.c　で動作しているプログラム。１ｍｓｅｃに１回走ります。

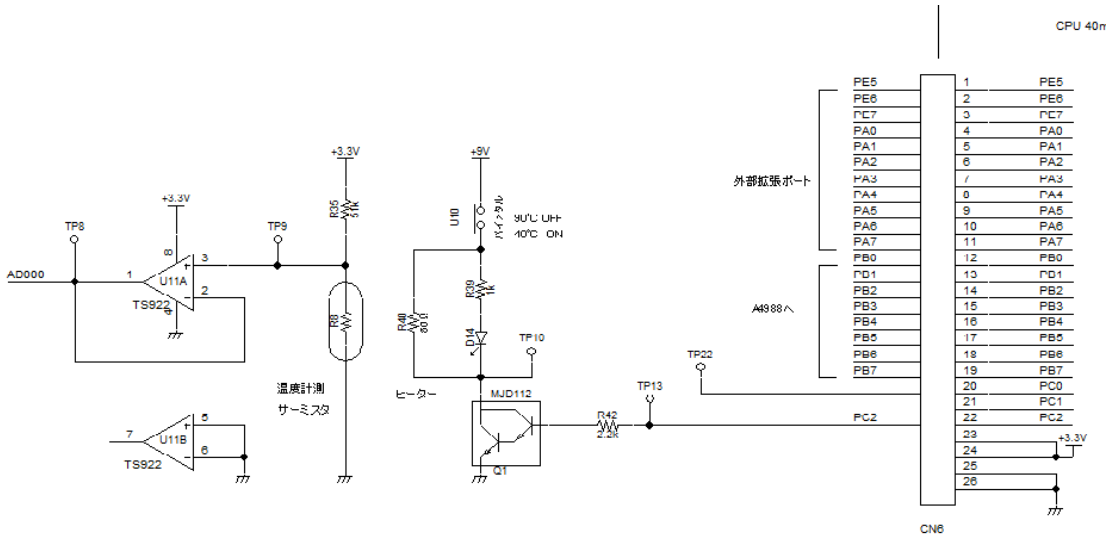
```
static void r_tmr_cmia0_interrupt(void)
{
    /* Start user code. Do not edit comment generated here */

//          PORTC.PODR.BIT.B0 = 1;      //LED1 ON

⑧           if(int_timer != 0){int_timer--;}

//          PORTC.PODR.BIT.B0 = 0;      //LED1 OFF


    /* End user code. Do not edit comment generated here */
}
```

これも骨格はコード生成で生成されますので、１ｍｓｅｃに１回、ｉｎｔ＿ｔｉｍｅｒが０でない場合、－１するというプログラムだけ書き込んであります。

３－２　ｓａｍｐｌｅ３２　比例帯温度調節　サーミスタリニアライズ付き

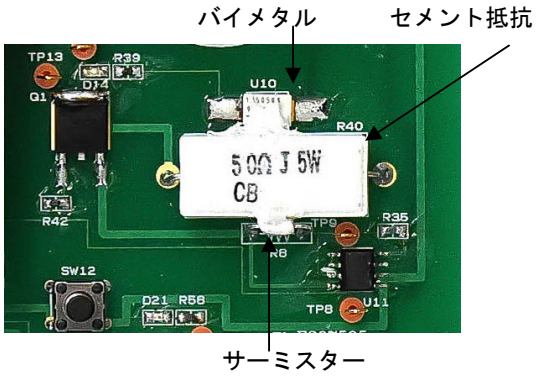
【　概要　】

ｓａｍｐｌｅ３２は比例帯温度調節ソフトサンプル、ｓａｍｐｌｅ３２ｂはＯＮ／ＯＦＦ制御ソフトサンプルになっています。ここではｓａｍｐｌｅ３２　比例帯温度調節について説明します。



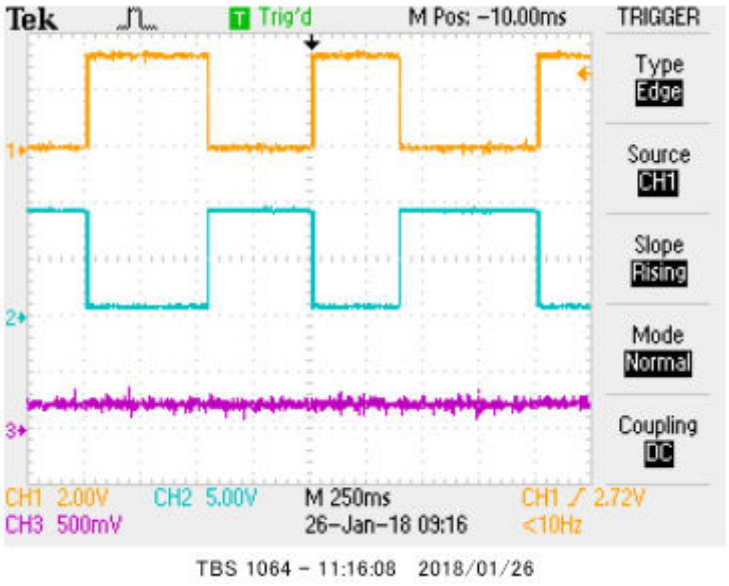
回路的にはＲＸ７１ＭのＰＣ２ポートでトランジスタＭＪＤ１１２をＯＮ／ＯＦＦさせます。ＯＮ時にはセメント抵抗Ｒ４０に電流が９Ｖ／５０Ω＝１８０ｍＡ流れて抵抗を温めます。バイメタルＵ１０とサーミスタＲ８はＲ４０とシリコンゴムで熱結合されています。バイメタルはヒステリシスを持っていて９０℃でオープンになりますが、４０℃まで元に戻ります。ずっと通電していても９０℃にはなりませんが、安全のために入っています。室温＋２０℃くらいが上限温度です。

サーミスタは温度上昇とともに抵抗値が減少する非線形素子です。温度を計測することが出来ますが、リニアライズが必要です。



比例帯温度制御、別名Ｐ制御は設定温度の上下に比例帯を設け、この中で単位時間内で負荷ＯＮ／ＯＦＦの比率を変えていく制御です。ＯＮ／ＯＦＦ制御（ｓａｍｐｌｅ３２ｂ）に比べてオーバーシュートが少ない特徴があります。欠点は設定値に対してオフセットが生じることで、これを取り除くために積分動作Ｉ、外乱からの戻りを迅速にする微分動作Ｄを加えた制御を一般にＰＩＤ制御といい、様々なものに使用されています。

本プログラムでは設定値を３５℃、比例帯１０℃で設定されています。



波形は上から
ＴＰ１３　ＰＣ２　トランジスタＱ１のベースに抵抗Ｒ４２を介して接続。
ＴＰ１０　トランジスタＱ１のコレクタ　ベースとは位相が反転します。９Ｖ制御。
ＴＰ９　　サーミスタ電圧　温度が上がると電圧は下がります。

【　主要プログラム　】

省略

【　解説　】

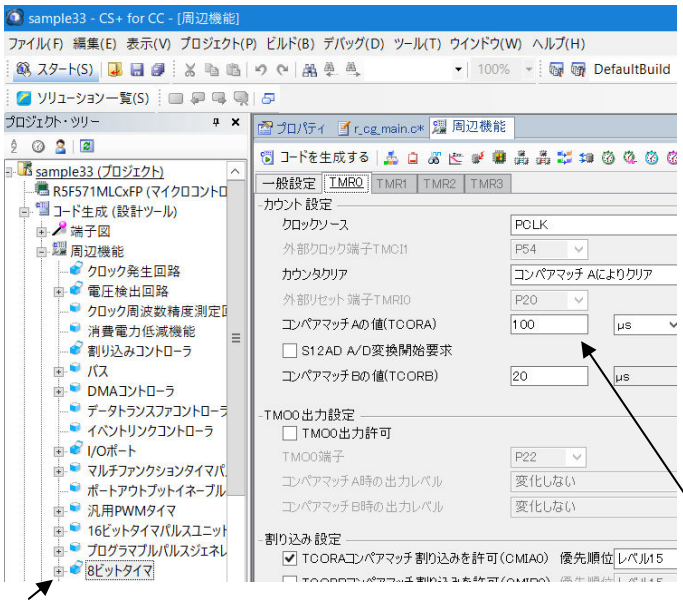
省略

３－３　ｓａｍｐｌｅ３３　メモリ使用音声録音再生器

【　動作概要　】

マイクで話した言葉をＡ／Ｄ変換、マイコンの内蔵メモリに記録し、スピーカーからＤ／Ａ変換で再生します。録音のボリウムは大きめ（右回りで大きくなります）、再生のボリウムは大きくし過ぎると音が割れてしまいます。

【周辺機能の説明】



８ビットタイマーは１００μｓｅｃ＝１０ＫＨｚにしてサンプリングします。

【　主要プログラム　】

省略

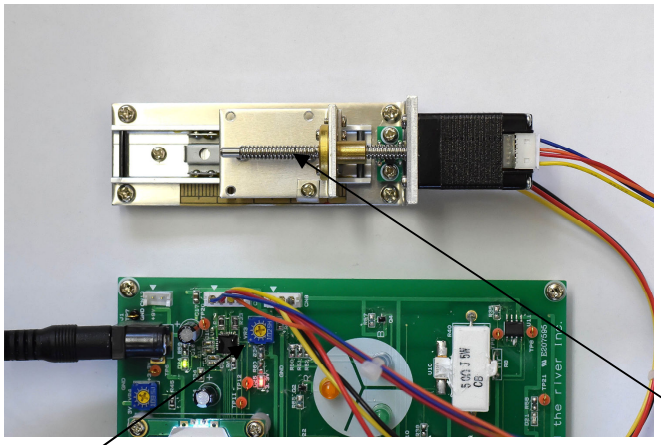
【　解説　】

省略

３－４　ｓａｍｐｌｅ３４　声でパルスモータを動かす　ヒューマンインターフェイス

【　動作概要　】

声でパルスモータを１mm動作させます。２０mmを超えると自動的に原点に戻ります。



※ステージのねじの部分にゴミが入ると回転がスムースに行かなくなりますの保存は必ず付属の　箱、またはゴミの無い環境でお願いします。

※パルスモータが脱調（空回りして大きな音がする現象）している場合、ＶＲ３で電流を増減し、正常になる場合があります。右回りで増加です。発熱があるので、増加させ過ぎないで下さい。

【周辺機能の説明】

特にありません。

【　主要プログラム　】

省略

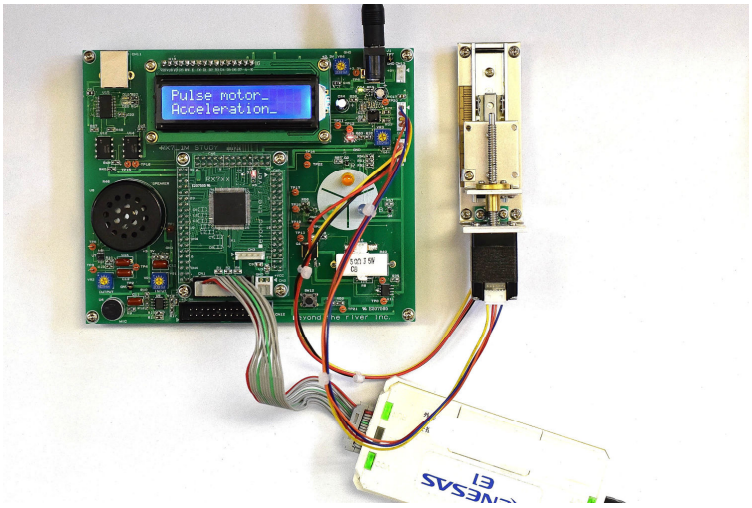
【　解説　】

省略

３－５　ｓａｍｐｌｅ３５　加速させてパルスモータを動作させる

【　動作　】

パルスモータを動作させるときに起動周波数を上げて高速域で動作させる場合があります。そのサンプルです。



動画はｙｏｕｔｕｂｅで見ることが出来ます。

https://youtu.be/OKBoqBwwaPw

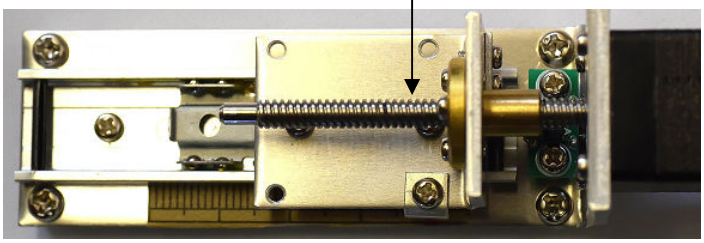
【　プログラム　】

省略

【　解説　】

省略

※ステージの回転ねじの部分にゴミが入ると回転がスムースに行かなくなりますの保存は必ず付属の箱、またはゴミの無い環境でお願いします。



お問い合わせ先：

〒３５０－１２１３　埼玉県日高市高萩１１４１－１
TEL　０４２（９８５）６９８２
FAX　０４２（９８５）６７２０
Homepage：http//beriver.co.jp
e-mail：info@beriver.co.jp
有限会社ビーリバーエレクトロニクス　