

前回 紹介しきれなかった物

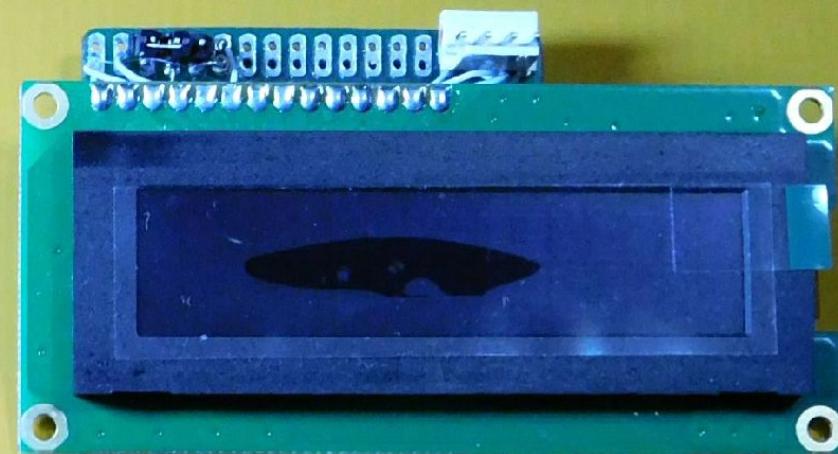
LCDと、OLEDです。

LCDは、バックライトが無いので、明るくないと見づらいです。しかし、今回のタイマーの用途では、そう頻繁に時刻を確認する物では



LCD 液晶ディスプレイ
バックライト無し

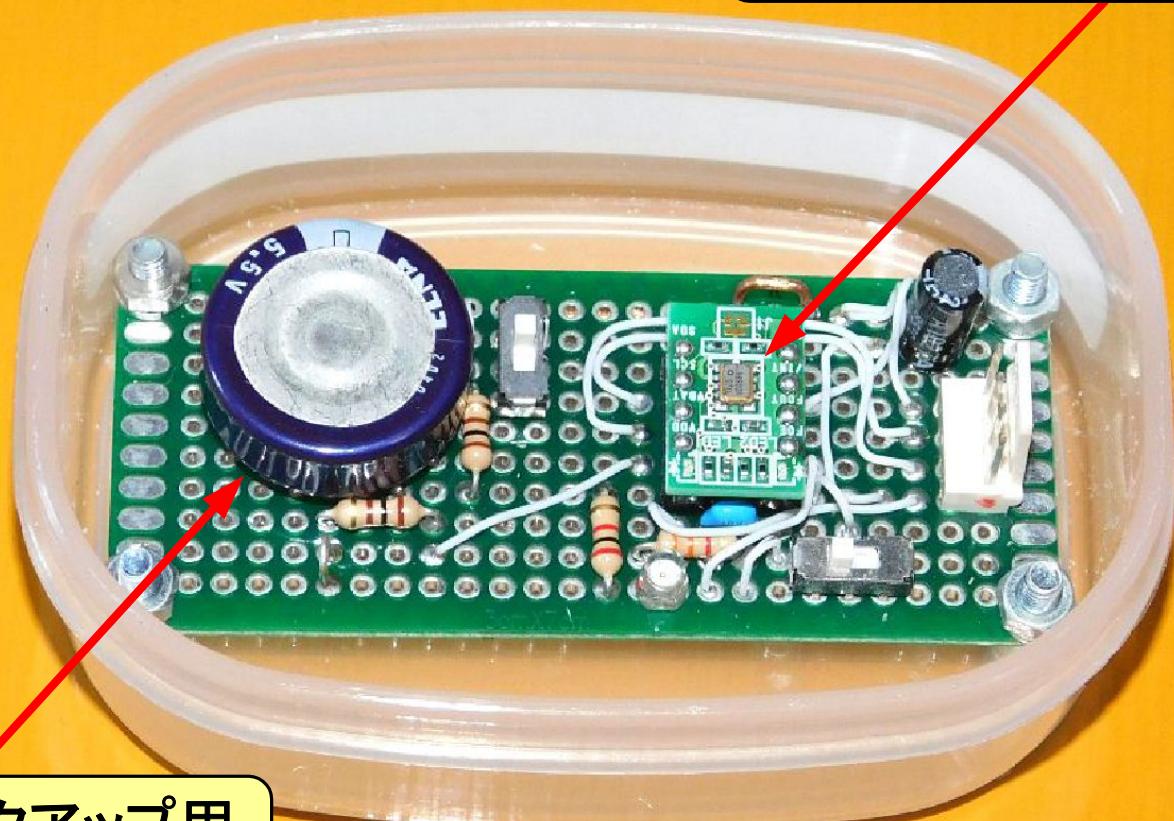
ないので、省エネ性重視で、左のLCDを使う事にしました。それと、OLEDは、3.3V動作で、5Vでは、使えません。LCDは、3.3Vでも5Vでも使用できます。表示文字数は、どちらも同じASCII文字 16文字 × 2行です。



OLED 有機ELディスプレイ
文字そのものが光る

リアルタイムクロック RX8900

秋月電子 RX8900 小基板



停電時バックアップ用
スーパーキャパシタ

4芯シールドケーブル 2.5m／外形：6mm



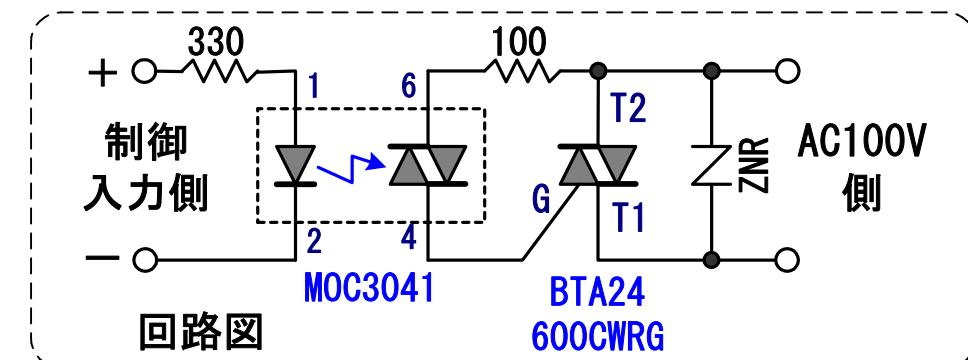
秋月電子のソリッドステートリレー・キット

初心者の方に、ソリッドステートリレーを、理解してもらいたいので、秋月電子のソリッドステートリレー・キットの 説明書の一部を 転載します。

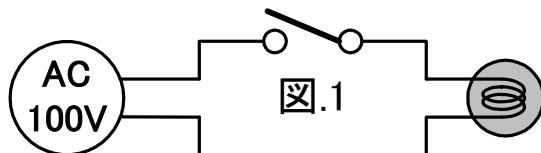
BTA24 600V 25A スナバーレストライアック使用
フォトトライアック使用 max 20A (25A) タイプ
ゼロクロス(交流電圧が 0Vの時 ON/OFFする)
スイッチ内蔵型フォトトライアック使用。
AC100V (40 ~ 220V) 50/60Hz

- ★ 本キットは、入力(制御)電圧 DC3V～DC8Vで作動する半導体(ソリッドステート)リレー SSRです。
- ★ AC100V(40V～220V)の交流で 20Aまで ON/OFF のコントロールができます。(放熱を全くしない時の出力は 最大 2Aまでです。)
- ★ フォトトライアック使用で超シンプル、部品点数も5個で完成。
(ZNRは サージアブソーバと同様の部品です。)

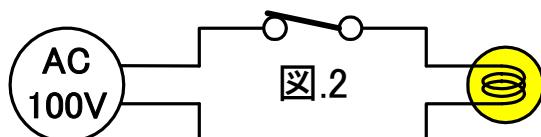
- ★ 制御入力と出力側は、フォトトライアックによって完全にアイソレートされていますので、TTL、CMOSトランジスタなどの制御回路から安全にACを制御できます。
- ★ 制御入力側は、フォトトライアック内部のLEDを点灯させるだけなので、TTL、CMOSロジックで簡単に直接制御が可能です。
- ★ 制御入力側の 消費電流は、5mA～30mA です。
- ★ 基板がガラスエポキシ基板になり、より小型化しました。
- ★ トライアック BTA24の 放熱部は、G、T1、T2 から絶縁されています。



ソリッドステートリレー 基礎の基



左の 図.1は、スイッチが、開いているので、電球は点灯しません。

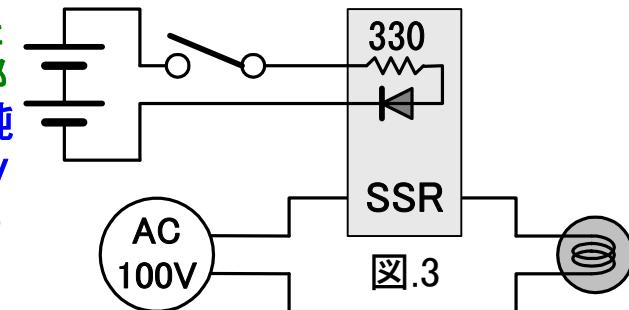


左の 図.2は、スイッチが、閉じているので、電球は点灯します。

上の例は、極めて当たり前のことがですが、上の図.1は、スイッチが開いています。図.2のスイッチは、閉じています。この2つの状態を半導体回路で実現した物が、SSR ソリッドステートリレーです。

スイッチの場合は、人間がスイッチ接点を開けたり閉じたりしますが、SSRの場合は、人間が手で操作する代わりに、SSR制御入力側の LEDに電流を流したり、切ったりする事により、AC100Vを ON、OFF 制御します。

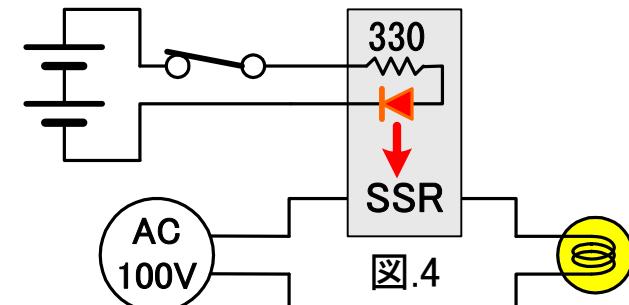
最後に書きたかったのですが 紙面の都合で... SSRは単純に AC100V、200V の交流のスイッチと考えて下さい



上の 図.3は、スイッチの代わりに、SSRを 入れた図です。 SSR制御側の LEDを スイッチと電池で 点灯消灯する事は、あり得ない事ですが、原理を 分かりやすくするために、電池とスイッチで表現しました。

上の図の場合、スイッチが 開いているので、LEDは 消灯してます。 よってAC側の 電球も点灯しません。

右の図.4は、スイッチを閉じて LEDを 点灯させて います。 SSRが、ONして AC側の電球も点灯します。



スレーブCPUボードの 作成

パワーユニット内の CPUは、表示制御ユニットから送られてくる単純なコマンドにより、8チャネルの SSRを、ON、OFF制御します。（今回は、SSRが6個しか無いので、6チャネルの制御となります。）

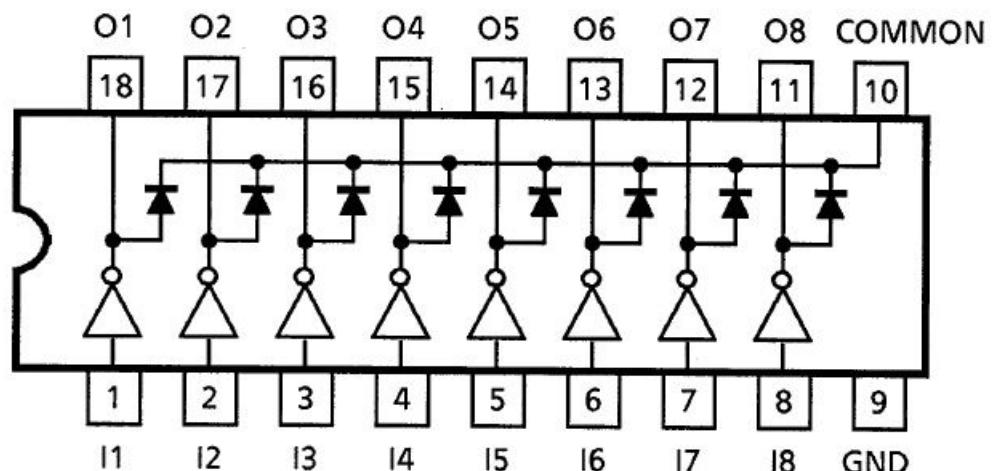
SSRの ON、OFFコマンドを送りだす表示制御ユニット側の CPUを マスタCPUと呼ぶ事にします。

コマンドを受けてSSRを駆動するパワーユニット内の CPUを スレーブCPUと呼ぶ事にします。

前回の動画のブロック図で、スレーブCPUに R8C/M120Aを使用する。そして SSR制御入力の LED信号と、スレーブCPUの 8本の出力ポートの間に、8bit オープンコレクタアレイを 入れる話をしてました。

これは、電源 5Vで LEDの電圧降下 1.6~1.8Vとトランジスタ ONの電圧降下を 0.3~0.4V を引くと3Vぐらいになります。

LEDの電流制限抵抗が 330Ω なので、 $3 / 330 = 0.009$ という事で 9mAぐらいになります。R8Cマイコンの出力ポートのドライブ能力は、10mAぐらいで、余力がありません。最悪の状態で 8ポート全て LEDの電流を引き込む場合は、 $0.009 * 8 = 72mA$ になります。CPUのグランドラインが、1本足の R8Cマイコンにおいて、グランドラインに 72mA も 流す事は、厳しい状況になると思ったので、8bit オープンコレクタアレイを入れる事にしました。今回使用する 8bit オープンコレクタアレイは、下図の TD62083APを 使用します。



TD62083APは、

出力耐圧: 50V

出力 1チャネルのみの出力: 最大 500mA

8チャネル同時出力時の デューティ50の
パルス出力時に

1チャネルに流せる最大電流: 123mA

これは、発熱の問題で 流せる電流が、減少するものと思われます。連続 流し続ける場合は 60mAぐらいに 減少するものと思われます。

今回は、1チャネル当たり 9mAで 使用する
予定なので 最大定格の 1/6 で使用する事に
なります。

入力電圧: 2.5 ~ 30V

入力電流: 標準:0.93mA 最大 1.35mA

Turn On 時間: 0.1us 、 Turn Off 時間: 0.2us

Turn On 時間は、トランジスタが、OFF状態から
ON状態に移行するのにかかる時間です。

Turn Off 時間は、トランジスタが、ON状態から
OFF状態に移行するのにかかる時間です。

Turn On 時間: 0.1us 、 Turn Off 時間: 0.2us

は、この手のオープンコレクタアレイとしては、早い方と
思います。

もう一つ、PC817C という

DIP 4ピンの

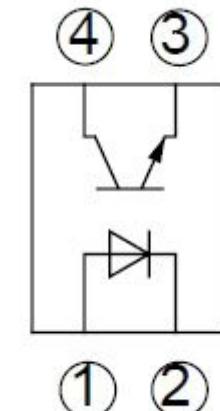
フォトカプラを使用します。

①、② 間が 入力側の
LEDです。

③、④ 間が 出力側の
フォトトランジスタです。

入力側と、出力側の間は
光で、絶縁されるため、5KVrmsの 耐圧が あります。

R8C/M120Aマイコンに関しては、過去に何度か取り
上げているので、今回はピンアサイン表のみ表示しま
す。 R8Cマイコンの詳細は [過去の私の動画、タイト
ル先頭の番号が、018'、019、020、021、022 等を、](#)
参照して下さい。



R8C/M120A CPUの Pinアサイン

P4_2/TRB0/TXDO//K13	1	20	P1_0/AN0/TRC10D//K10
P3_7//ADTRG/TRJ0/TRC10D	2	19	P1_1/AN1/TRC10A/TRCTRG/
RESET/PA_0	3	18	P1_2/AN2/TRC10B//K12
P4_7/XOUT//INT2	4	17	P1_3/AN3/TRC10C//K13/TRB0
VSS/AVSS	5	16	P1_4/AN4/TXDO/RXDO//INT0/
P4_6/XIN/RxD0/TxD0//INT1/	6	15	P1_5/RXDO/TRJ10//INT1/
VCC/AVCC	7	14	P1_6/VREF1/CLK0/TRJ0/
MODE	8	13	P1_7/AN7/CMP1/IVCMP1//INT1/
P3_5/TRC10D//K12/VCOUT3	9	12	P4_5//INT0/ADTRG
P3_4/VREF3/TRC10C//INT2	10	11	P3_3/VCMP3/TRCCLK//INT3

赤のPin (5, 7, 8) は、他の用途では使えない。

柿色のPin (3, 4, 6) は、通常は、RESET、XOUT、XIN に使用するが、
Pinが足りない時は、条件付きで別の用途にも使える。

R8C/M120Aマイコンの周辺回路で 今までと異なる箇所

一つ説明を忘れてましたが、今まで **ICSP** というか
イン サーキット プログラミング の機能を、**ターゲット
マイコン基板** に付けていました。

しかし **今回の基板には ICSP機能を付けていません**。初心者の方には、**イン サーキット プログラミング** という言葉が、ピンとこないという方も おられるかもしれませんので、簡単に説明しておきます。

ICSP とは、**実際の運用に使用するターゲット基板** 上においてマイコンにプログラムを **書き込む事**、または **書き込む機能** の事です。

特に **表面実装型マイコン** のように **プリント基板** に直接CPUが **半田付け** されている場合は、CPUを **外して** プログラムを **書き込む事** は **出来ないので** **ICSP機能** は、**必須** となります。

今回の **R8C/M120Aマイコン** は **DIP20ピン** なので **ICソケット** にて、**抜き差し** する事が **可能** です。

よって今回は、CPUに **書き込み器** でプログラムを書き込んで、ターゲットボードに、CPUを **挿入** する事にしました。一つには、今回のプログラムの動作が、非常に単純なので、**デバッグ** に、そう手間どる事は、なかろうと判断したからです。

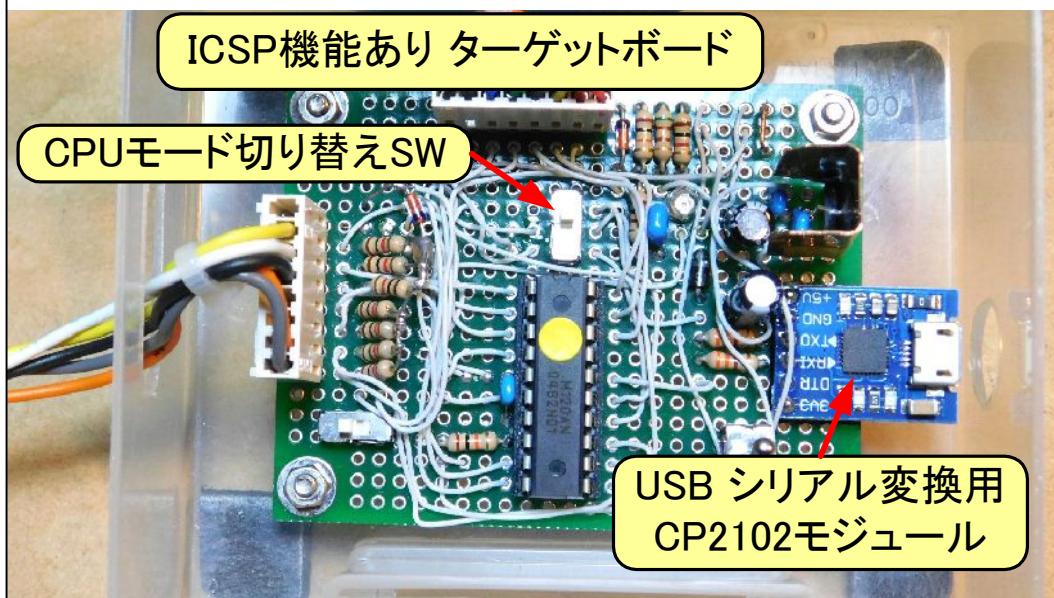
逆に、**デバッグ** に手間どる可能性のあるプログラム開発では、**書き込み器** と、ターゲットボード間で、CPUの **抜き差し** を、**頻繁** に行うことになるので、**ICSP** の方が **便利** です。

今回、**ICSP** を使用しなかったのは、**プログラム書き込み** 時に使用するピンが、**ポート1** の **14ピン** と **16ピン** で、**ポート1** は、**8bit** 全てを **SSR出力** に割り当てており **使用するピン** が、重なっていたからです。

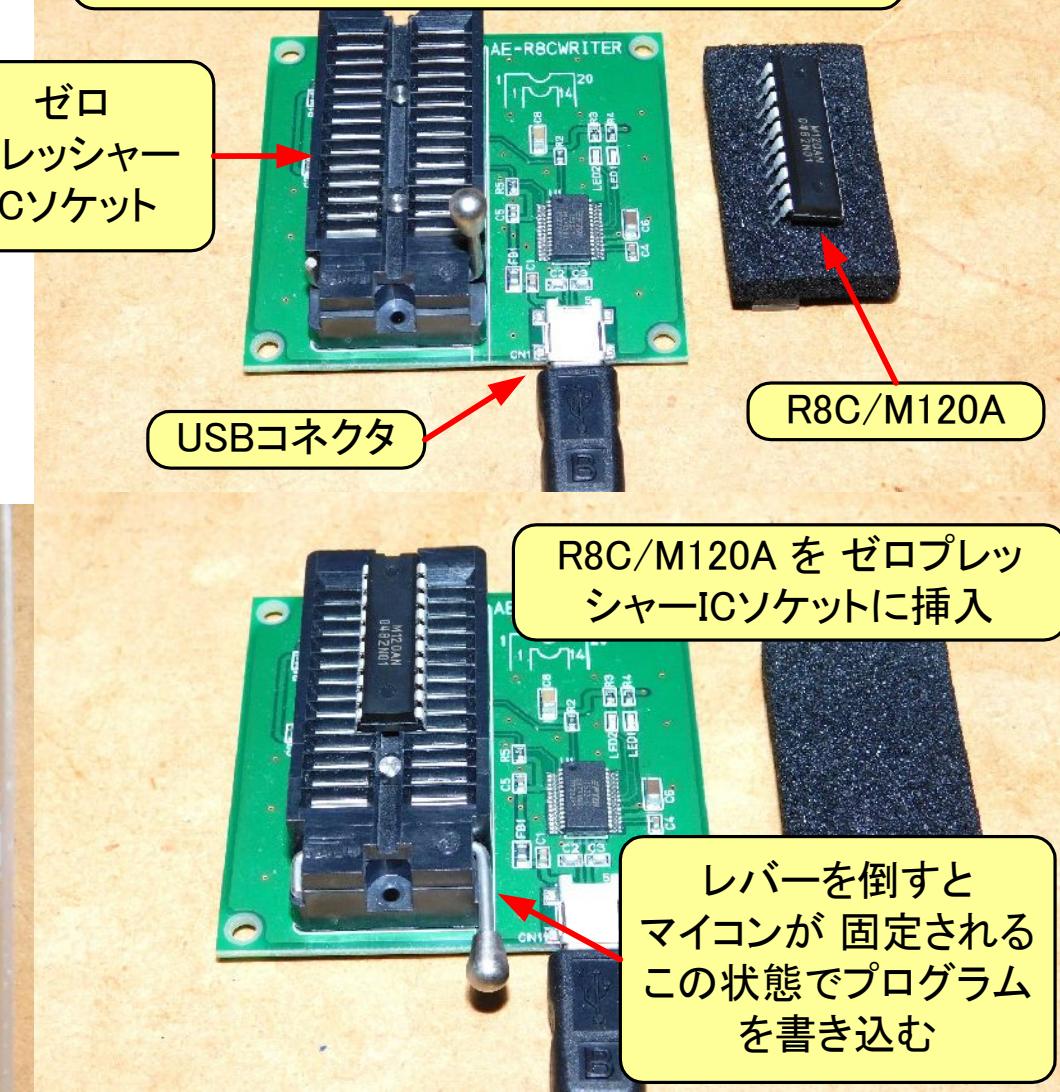
それと、**ICSP** に **関わる** 余分な部品を付けなくて済む。という事もあります。文面ばかりでは イメージが掴みにくいと思いますので、次ページで画像で説明します。

ICSP機能 有無の 作業の違い

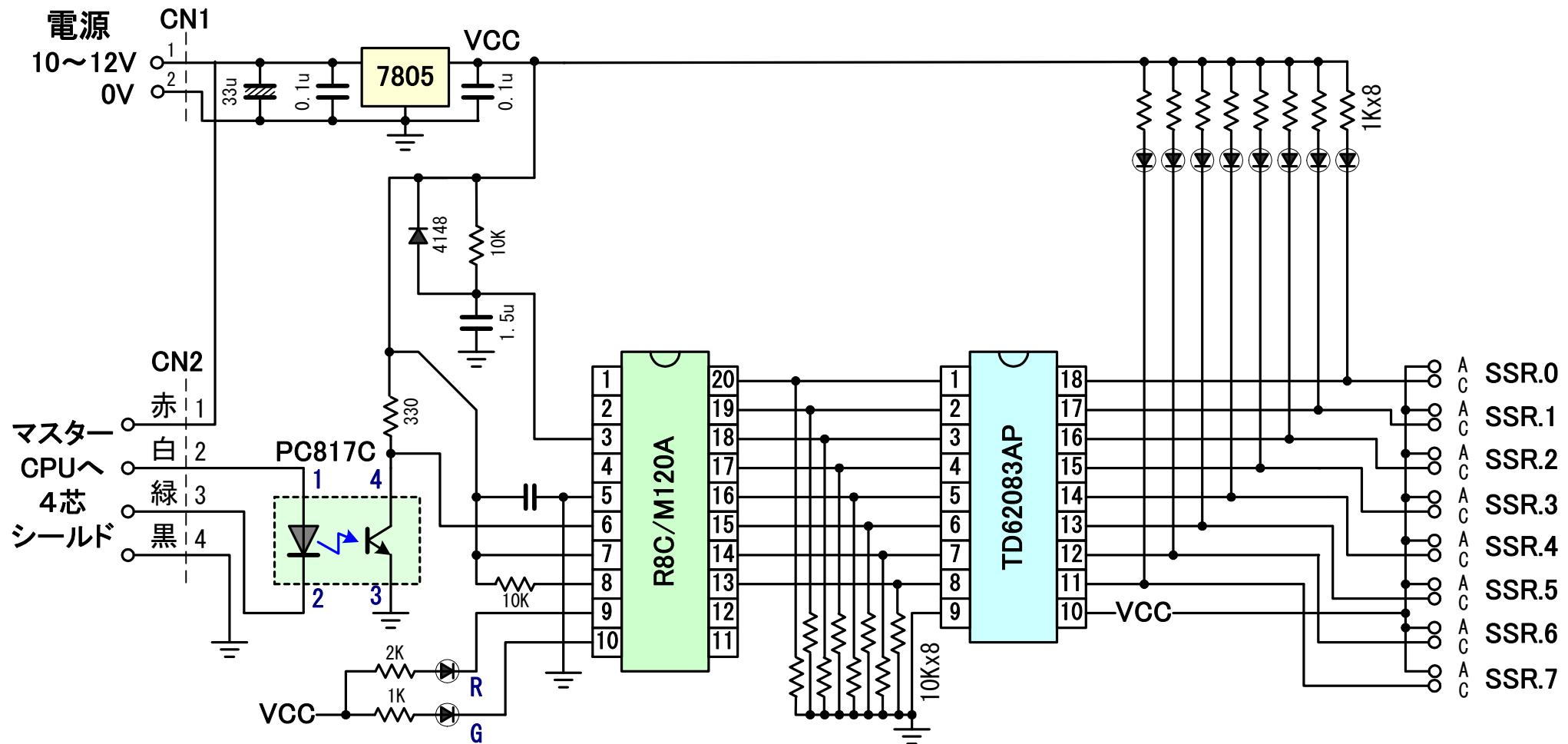
左下の画像は、ICSP機能の付いたマイコンボードです。そのICSP機能に関わる部品としてUSBシリアル変換用 CP2102モジュールとCPUの書き込みモードと、実行モードを切り替えるスイッチが、あります。右は、秋月電子製 R8C/M1用書き込み器です。書き込み器側で、プログラムを、書き込み その CPUをターゲットボードに挿入します。



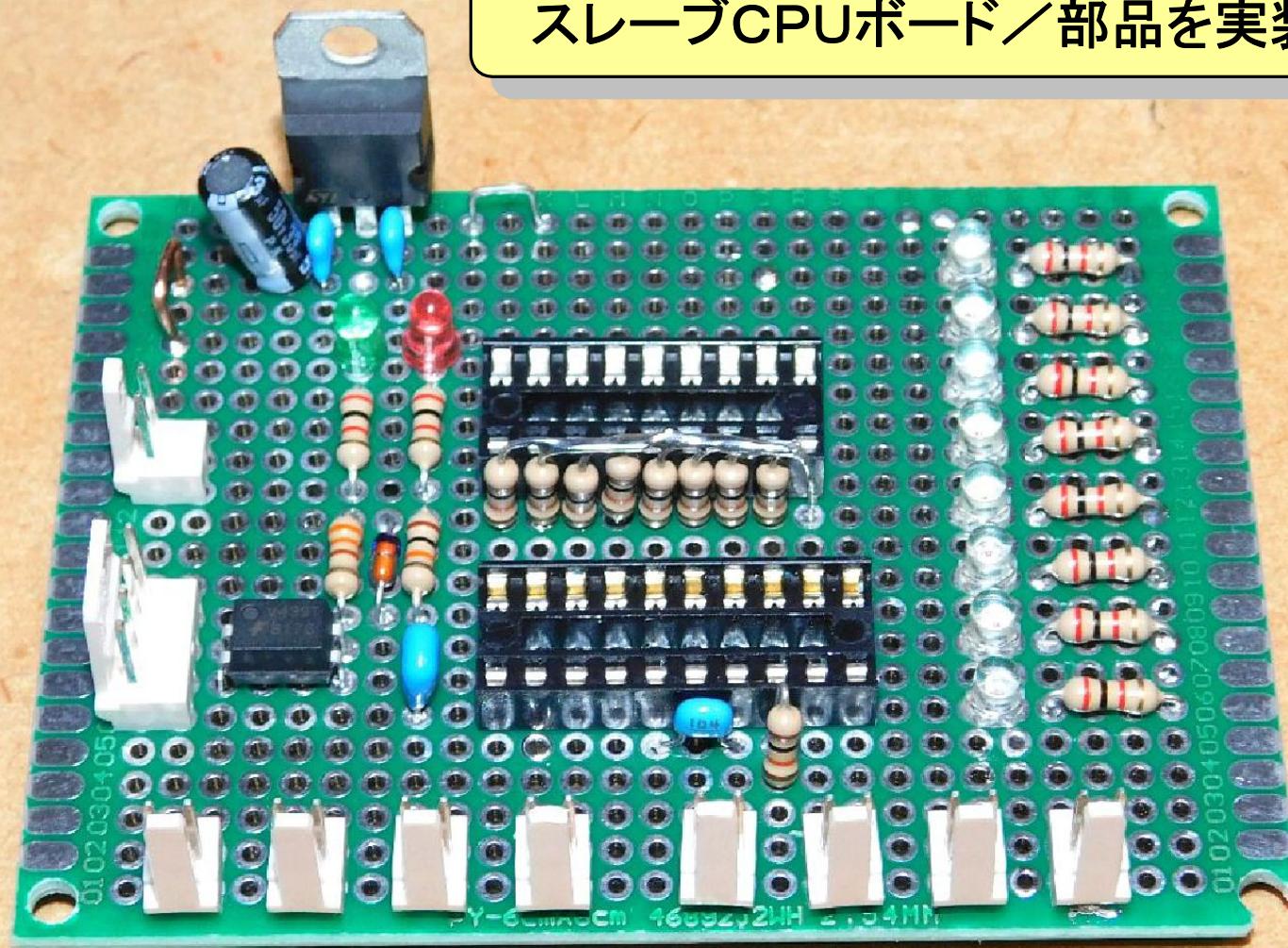
秋月電子 R8C/M1シリーズ用 書き込み器



スレーブCPUボード回路図



スレーブCPUボード／部品を実装した画像



後は、部品間の配線作業を行います。