

概要

以前、R8Cマイコンで モールス受信側を作成した際に、受信文字表示を、8×8DotのLED表示モジュールで表示したい。と コメントに

書き込まれた Jさんがいたので、その時 どのようなものか知らなかったので、探したところ、Amazonで それらしい物を見つけました。



ブランド: VKLSVAN

VKLSVAN MAX7219 ドットマトリクスモジュール 4 In 1 MCU 制御 LED ディスプレイモジュール Arduino対応

4.3 ★★★★★ 8個の評価

¥890 税込

ポイント: 9pt (1%) 詳細はこちら

法人価格で最大4%割引となります。無料のビジネスアカウントに登録する

あと払い (ペイディ) で買い物するたび毎回最大10%ポイント還元。 ※毎月エントリーの必要あり

画像にマウスを合わせると拡大されます

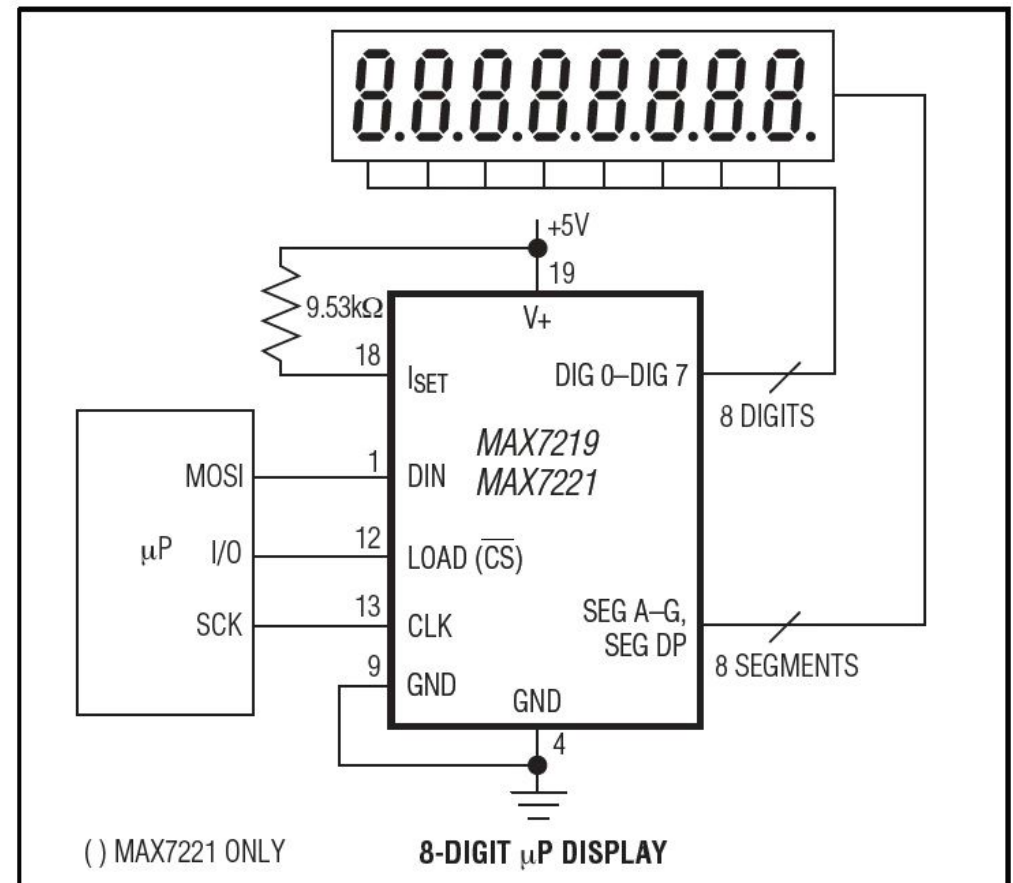
今回購入した
MAX7219 ドットマトリクス 4 in 1 に ついて

という事で、1個 Amazonから 試しに購入しました。商品の構成は、

- ① 8×8 LED ドットマトリクスモジュールが 4個
- ② LED ドットマトリクスモジュール4個分の横長のプリント基板 1枚、そして基板には **MAX7219**という ダイナミック点灯の、ICが **4個実装**されています。**8×8 LEDドットマトリクスモジュール1個に対し、1個の MAX7219が対応**します。CPUとのインタフェースは、連結可能なシリアルインタフェースのようです。

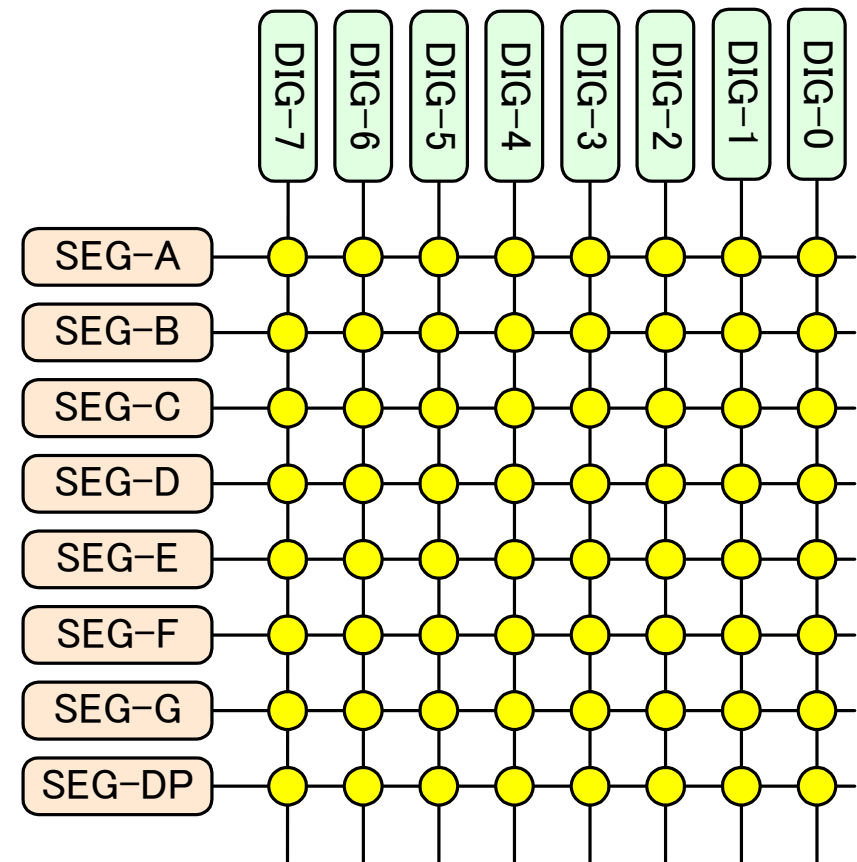
そして、②の基板に ①のLEDモジュールを 4個 ハンダ付けされています。

MAX7219 標準アプリケーション回路



前のページの右側の標準アプリケーション回路図を見て、あれっ LED表示器は **8桁の 7セグメント表示器**だ。と思われた方も いると思います。元々は 8桁の7セグメント表示器を想定して作られたものではないかと思われま
す。7セグメントは、**A ~ G の 7個のセグメント**と、少数点の **Dotセグメント**表示で、**計 8セグメント**(SEGMENT)に、なります。そして**8桁**(DIGIT)の表示桁数です。これを、8×8ドットLEDモジュールに そのまま置き換えた。という事です。

右は、結線のイメージ図です。小さい黄色の丸が、LEDです。右側の結線は、購入した基板のプリント配線で 既に配線済みになっています。よって、電源とアース、CPUとの信号線3本を 接続すれば、ハード的には 使えるようになります。電源電圧は、Amazonの 商品説明欄には **5V**と書いてあります。



CPUと接続する信号線について

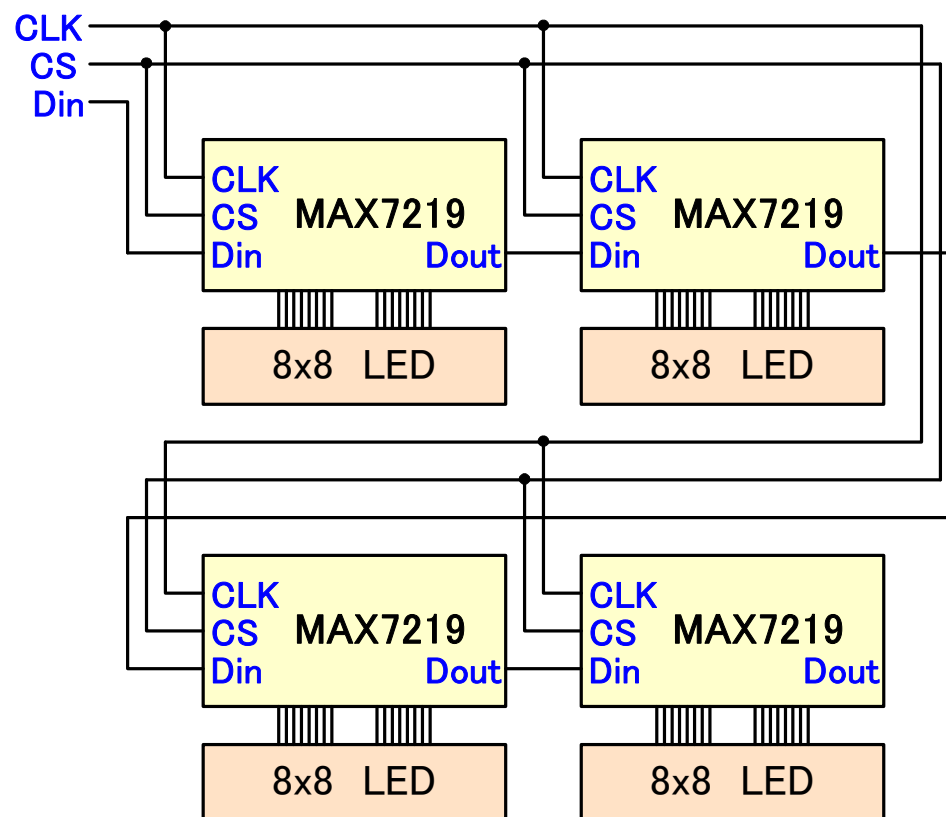
マイコンと接続する信号線は、シリアル通信で3本です。

MAX7219 ドットマトリクス 4 in 1 コネクタ端子	
信号名	内 容
VCC	+5V 電源
GND	0V グランド
DIN	シリアルデータ入力
CS	チップセレクト
CLK	同期用クロック

MAX7219に シリアルデータを渡すためのデータ出力信号です。この信号は、MAX7219 内で、データを表示するための 16bit の シフトレジスタの 出力です。よって 先頭、2番目、3番目、4番目の MAX7219に 順次 16bitの遅延を伴いながら シリアルデータが、渡されて行きます。

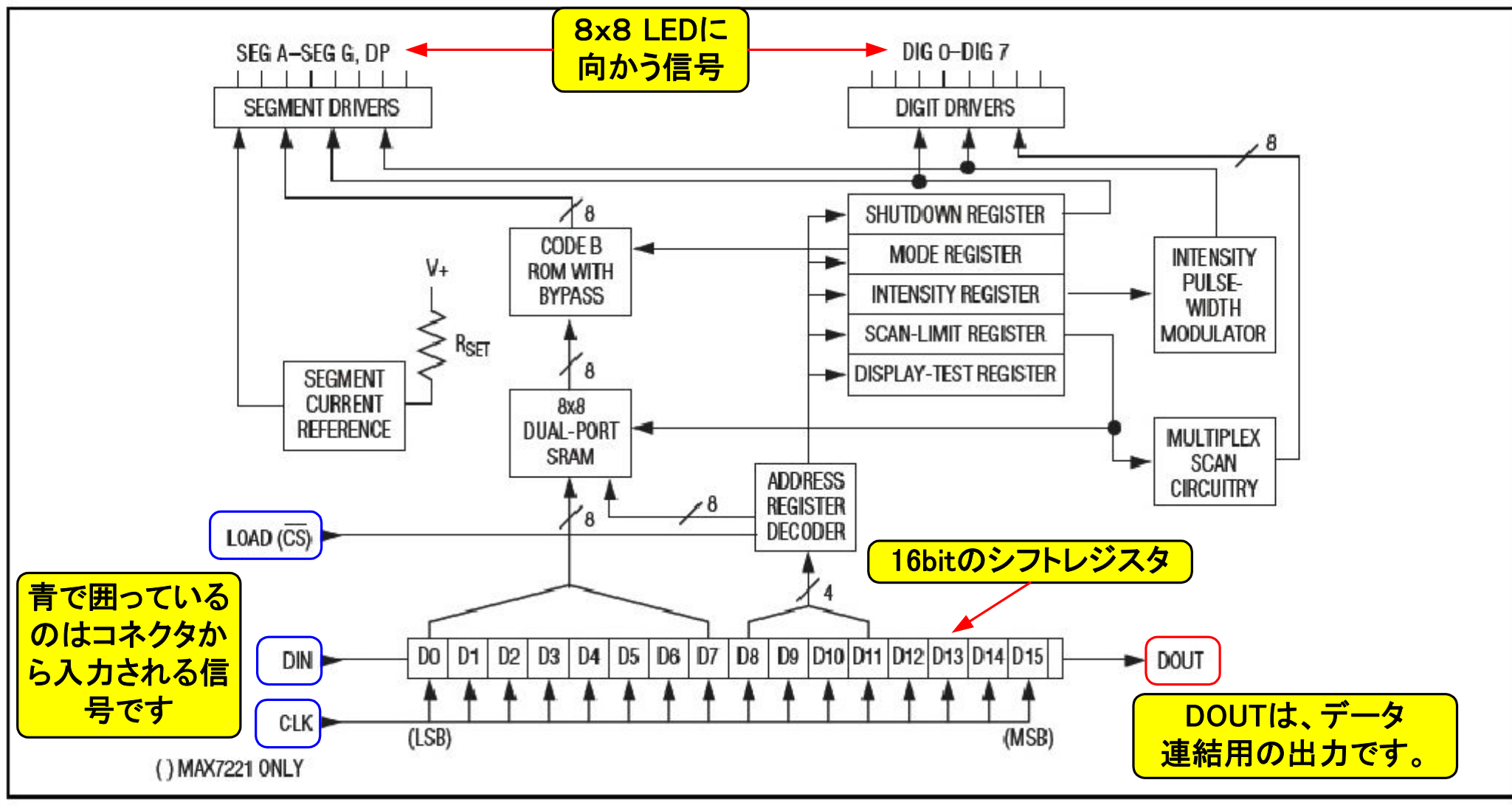
それと、5Pのコネクタには、出ていませんが **Dout** という信号線があります。これは、先頭のMAX7219から2番目の

言葉だけで、説明するのは意味が、分かりにくいので、図にしてみます。



ファンクションダイアグラム

線が細くて見づらいところがあるのは、ご容赦下さい。



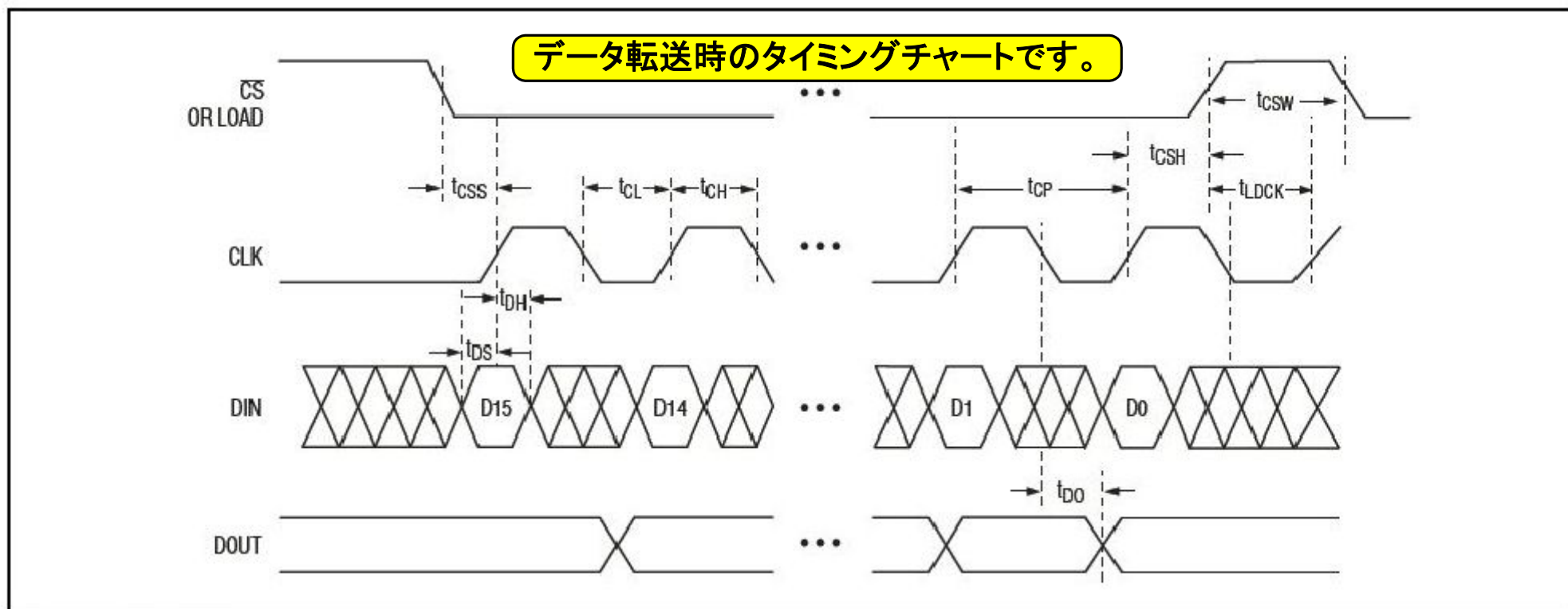


図1. タイミング図

表1. シリアルデータのフォーマット(16ビット)

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
X	X	X	X	ADDRESS				MSB	DATA						LSB

シリアルデータの転送ですが、このタイミングチャートでは、D15 ~ D0 の順に転送します。

シリアルデータのフォーマット

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
無効Data				アドレス				データ							

シリアルデータのフォーマットは、上位4 bit は、無効データです。さしあたり 0 とします。
 アドレスというのは、右側の表のレジスタ選択のアドレスです。例えば、アドレス = 0h は Non Operation で 何もしない事になります。アドレス = 1h は セブンセグメントであれば 最下行の1桁のセグメント データを書き込む事になります。但し、表示するLEDは 8x8 ドットマトリクスなので 実際に表示してみないと、どこが点灯するのか分かりません。多分 常識的に考えると 一番右の縦一列ではないかと思えます。

アドレス = 2h は 上の仮定で考えると、右から2番目の 縦一列のデータと 思われます。

アドレス = 8h は 上の仮定で考えると、8x8の一番左の、縦一列ではないかと思われます。

表2. レジスタアドレスマップ

REGISTER	ADDRESS					HEX CODE
	D15-D12	D11	D10	D9	D8	
No-Op	X	0	0	0	0	0xX0
Digit 0	X	0	0	0	1	0xX1
Digit 1	X	0	0	1	0	0xX2
Digit 2	X	0	0	1	1	0xX3
Digit 3	X	0	1	0	0	0xX4
Digit 4	X	0	1	0	1	0xX5
Digit 5	X	0	1	1	0	0xX6
Digit 6	X	0	1	1	1	0xX7
Digit 7	X	1	0	0	0	0xX8
Decode Mode	X	1	0	0	1	0xX9
Intensity	X	1	0	1	0	0xXA
Scan Limit	X	1	0	1	1	0xXB
Shutdown	X	1	1	0	0	0xXC
Display Test	X	1	1	1	1	0xFF

レジスタアドレスマップの 9h 以降ですが
いろんな設定が出来るようです。

アドレス 9h アドレスモードレジスタです。
これは、BCDコードを表示する時のデコード
モードをいくつか持っている様ですが、8x8ドット
LEDモジュールの場合は、デコード処理を行
わず、そのまま 8 bitデータを LEDに出力しま
す。よって、アドレス 9h = 00h になります。

アドレス Ah 輝度調整です。
データ設定範囲は 00h ~ 0Fhで、明るさを
段階的に設定可能で 0Fhが、一番明るい設定
と 思われます。

アドレス Bh スキャン制限レジスタです。
これも、7セグメントLEDに特化した機能なので
アドレス Bh = 07h を 設定して下さい。

アドレス Ch シャットダウンレジスタです。
これは、LEDを消灯して、省エネ状態にする機
能です。

アドレス Ch = 01h LED表示状態になります。

アドレス Ch = 00h LED消灯状態になります。

アドレス Dh ~ Eh は レジスタが ありません。

アドレス Fh ディスプレイ表示テストレジスタ
です。全ての LEDが 点灯します。

アドレス Fh = 00h 通常状態。

アドレス Fh = 01h LEDテスト状態。

後は、マイコン側にて、今回の3本の足ピンを
決める事にします。次のページに R8C/M110
と M120の ピン割り付け表を 示します。

信号出力ピン(Din、/CS、CLK)を 決めたら
ソフトを作ります。

R8C/M110AN

Port. 1							
b7	b6	b5	b4	b3	b2	b1	b0
P1_7	P1_6	P1_5	P1_4	P1_3	P1_2	P1_1	
8	9	10	11	12	13	14	

Port. 3							
b7	b6	b5	b4	b3	b2	b1	b0
P3_7							
1							

Port. 4							
b7	b6	b5	b4	b3	b2	b1	b0
P4_7	P4_6	P4_6、P4_7は 水晶で使用					
3	5						

Port. A							
b7	b6	b5	b4	b3	b2	b1	b0
PA_0は /RESET で 使用							PA_0
							2

P1_4 ~ P1_6は 通信 PGM書き込みで使用

R8C/M120AN

Port. 1							
b7	b6	b5	b4	b3	b2	b1	b0
P1_7	P1_6	P1_5	P1_4	P1_3	P1_2	P1_1	P1_0
13	14	15	16	17	18	19	20

Port. 3							
b7	b6	b5	b4	b3	b2	b1	b0
P3_7		P3_5	P3_4	P3_3			
2		9	10	11			

Port. 4							
b7	b6	b5	b4	b3	b2	b1	b0
P4_7	P4_6	P4_5			P4_2		
4	6	12			1		

Port. A							
b7	b6	b5	b4	b3	b2	b1	b0
							PA_0
							3

百円 R8Cマイコンの 足ピン取り決め

R8C/M110AN でピン数は、足りてるので R8C/M110AN(14ピン)を使用します。

CLK: P3_7(1ピン)

/CS: P1_1(14ピン)

Din: P1_2(13ピン)

とします。

この、3本の信号線で、一番頻繁にアクセスするのは、CLK信号線です。 よってこの信号線を 1つのポートで単独で使える方が、I/Oポートアクセスの 速度面で有利です。

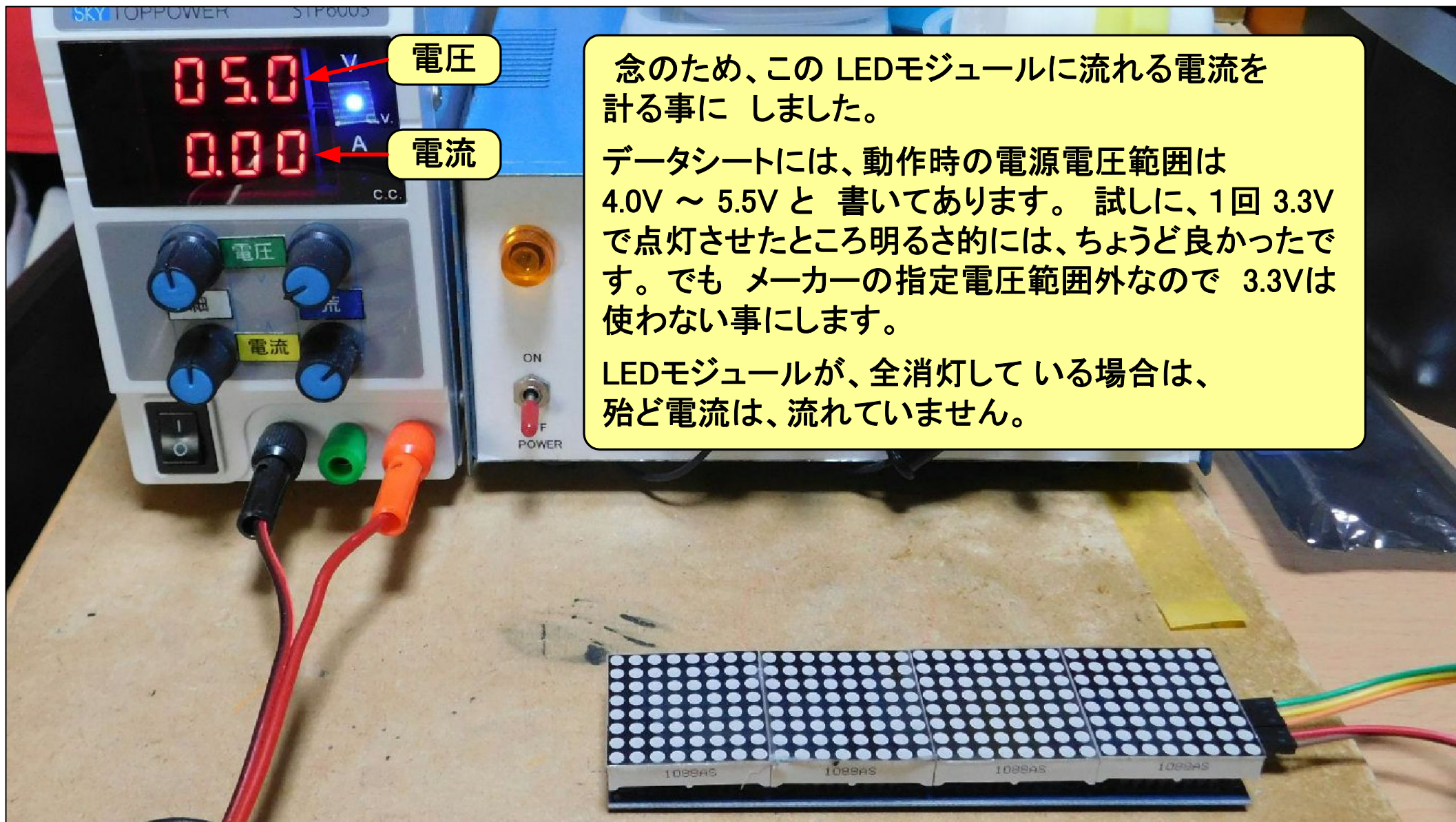
そして、ひとまとまりのアクセス単位として 16ビットのシリアルデータが、あります。

この MAX7219 ポートアクセス処理を、一つの サブルーチンとします。 引数として WORD データを渡して、それを 16bit シリアルデータとして転送します。 このサブルーチンを アセンブラで 作成します。

何故、アセンブラで作成するかというと、ルネサスHEWの環境で R8Cの プログラムを作成する場合、I/O周りの こま回りの処理の場合、Cで作るより、アセンブラで作成した方が、約 3倍ほど 実行速度が 早くなるからです。

ここまで、仕様を決めれば後は、プログラム開発と なります。 プログラム開発の細かい話は今回は、しません。 今回も HEWの Projectはダウンロード出来るようにしておきますので、興味のある方は ダウンロードして見てください。

次は、実験の動画を お見せします。



電圧

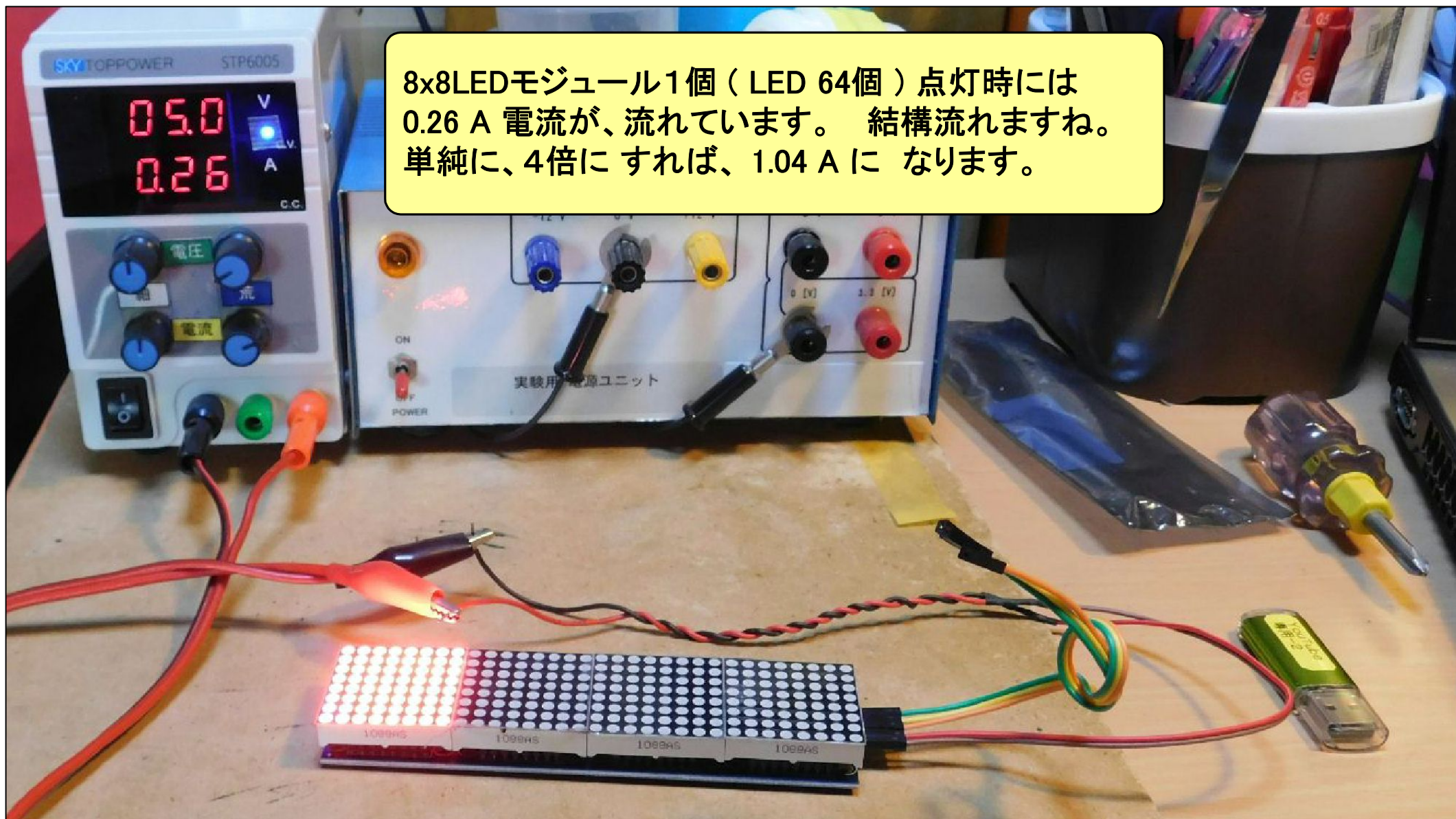
電流

念のため、この LEDモジュールに流れる電流を計る事に しました。

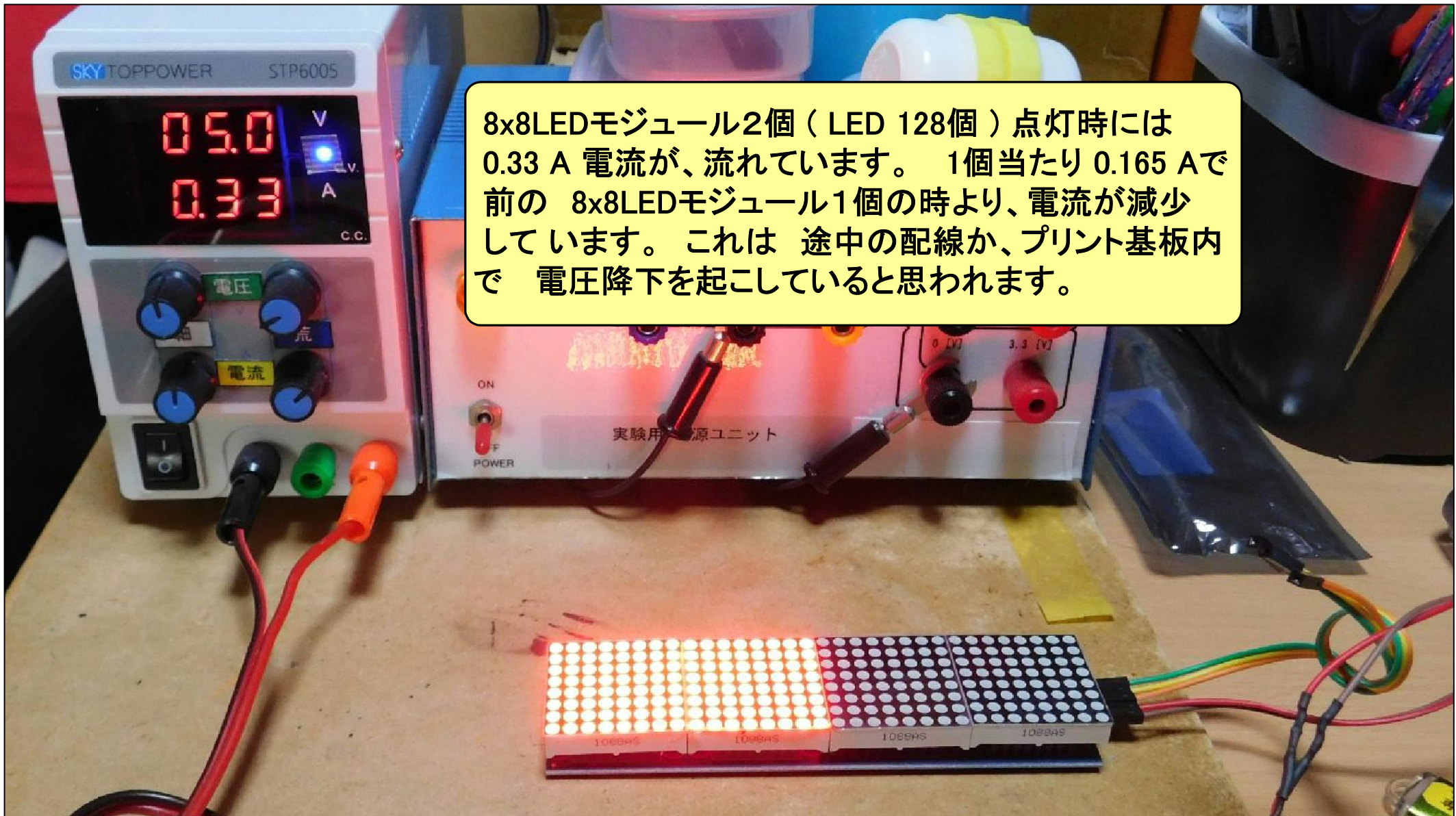
データシートには、動作時の電源電圧範囲は 4.0V ~ 5.5V と 書いてあります。 試しに、1回 3.3V で点灯させたところ明るさ的には、ちょうど良かったです。でも メーカーの指定電圧範囲外なので 3.3Vは使わない事にします。

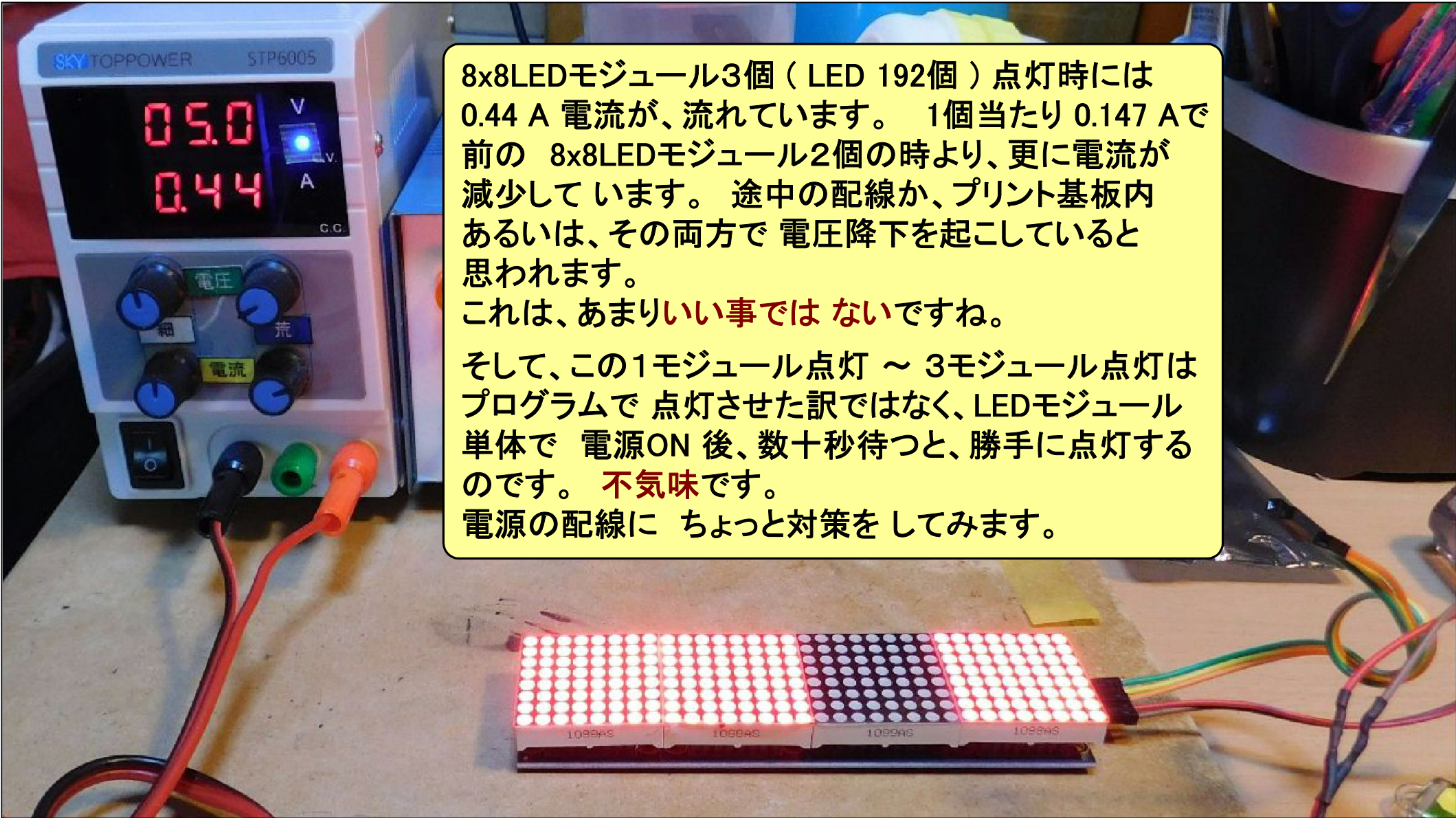
LEDモジュールが、全消灯している場合は、殆ど電流は、流れていません。

8x8LEDモジュール1個（LED 64個）点灯時には
0.26 A 電流が、流れています。結構流れますね。
単純に、4倍にすれば、1.04 A になります。



8x8LEDモジュール2個（LED 128個）点灯時には
0.33 A 電流が、流れています。 1個当たり 0.165 Aで
前の 8x8LEDモジュール1個の時より、電流が減少
しています。これは 途中の配線か、プリント基板内
で 電圧降下を起こしていると思われます。





8x8LEDモジュール3個（LED 192個）点灯時には
0.44 A 電流が、流れています。 1個当たり 0.147 Aで
前の 8x8LEDモジュール2個の時より、更に電流が
減少しています。 途中の配線か、プリント基板内
あるいは、その両方で 電圧降下を起こしていると思
われます。

これは、あまりいい事では ないですね。

そして、この1モジュール点灯 ～ 3モジュール点灯は
プログラムで 点灯させた訳ではなく、LEDモジュール
単体で 電源ON 後、数十秒待つと、勝手に点灯する
のです。 **不気味**です。

電源の配線に ちょっと対策を してみます。

電源回り、その他改修

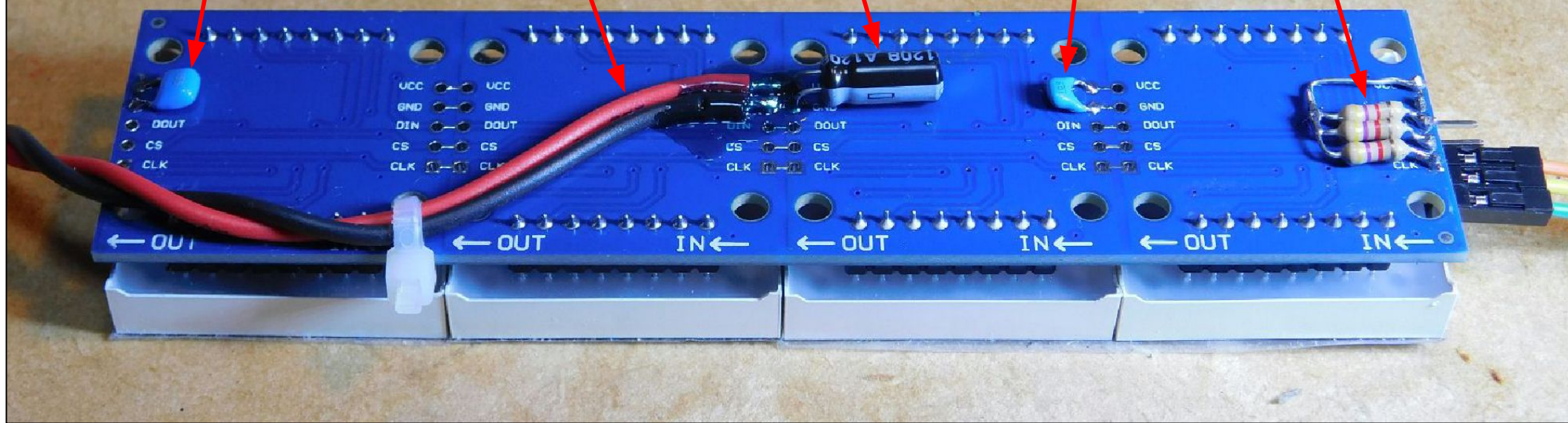
33 μ Fの電解コン

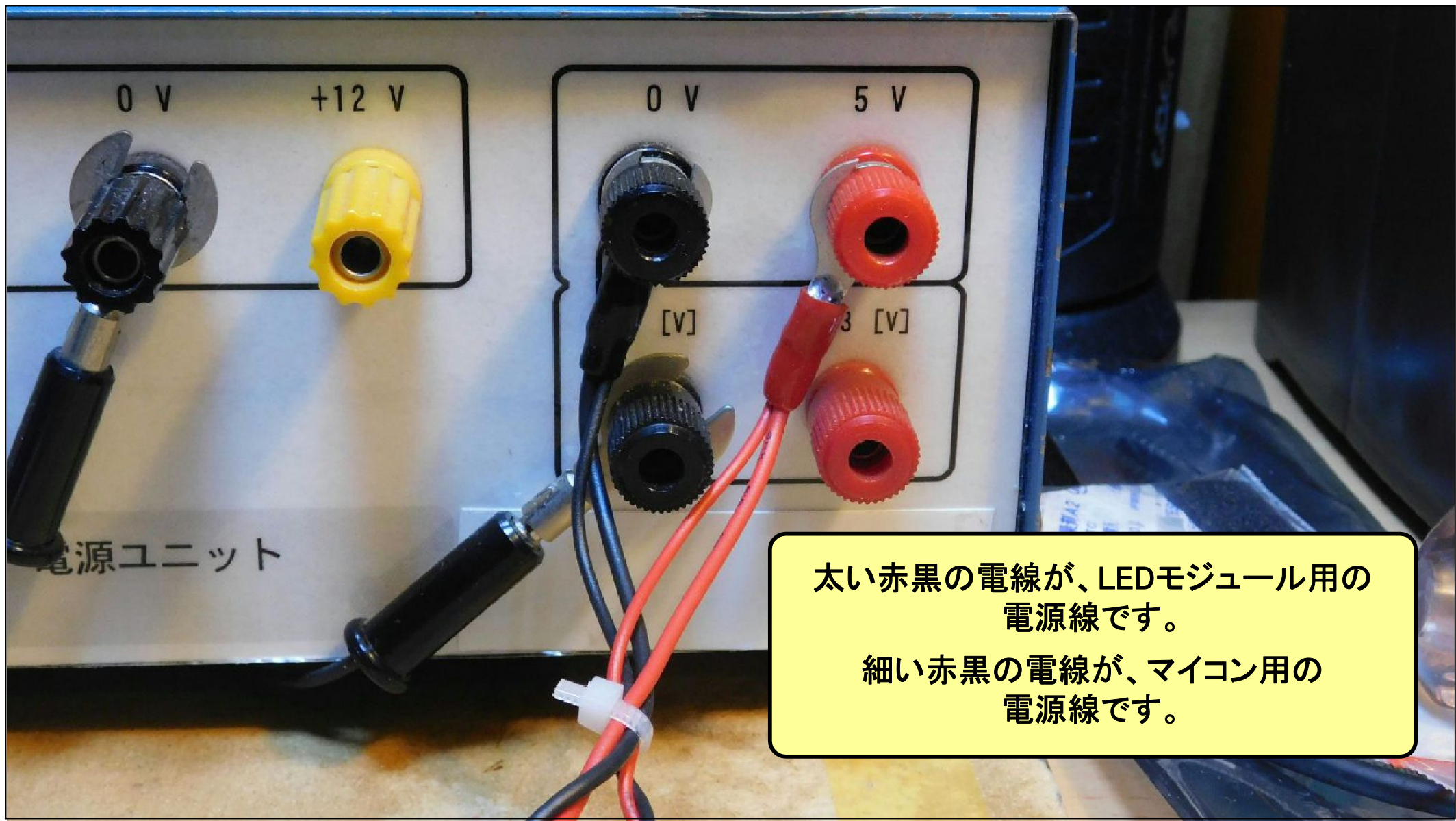
0.1 μ Fの積層セラコン

1.5 μ Fの積層セラコン

電源線をパターン
中央位置に半田付け

4.7K Ω PullUp抵抗 3本を
Din、CS,CLKに接続

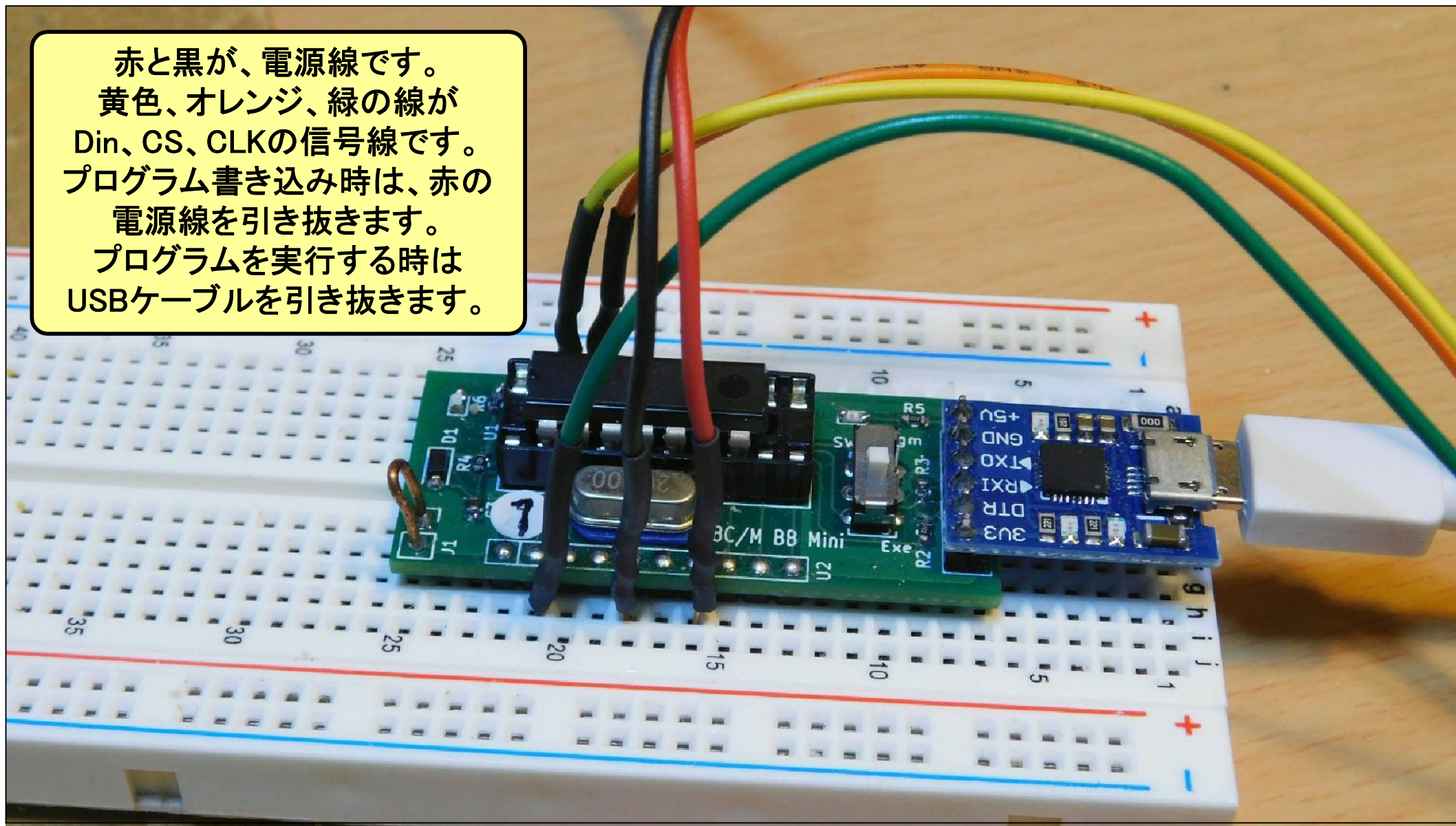




太い赤黒の電線が、LEDモジュール用の
電源線です。

細い赤黒の電線が、マイコン用の
電源線です。

赤と黒が、電源線です。
黄色、オレンジ、緑の線が
Din、CS、CLKの信号線です。
プログラム書き込み時は、赤の
電源線を引き抜きます。
プログラムを実行する時は
USBケーブルを引き抜きます。

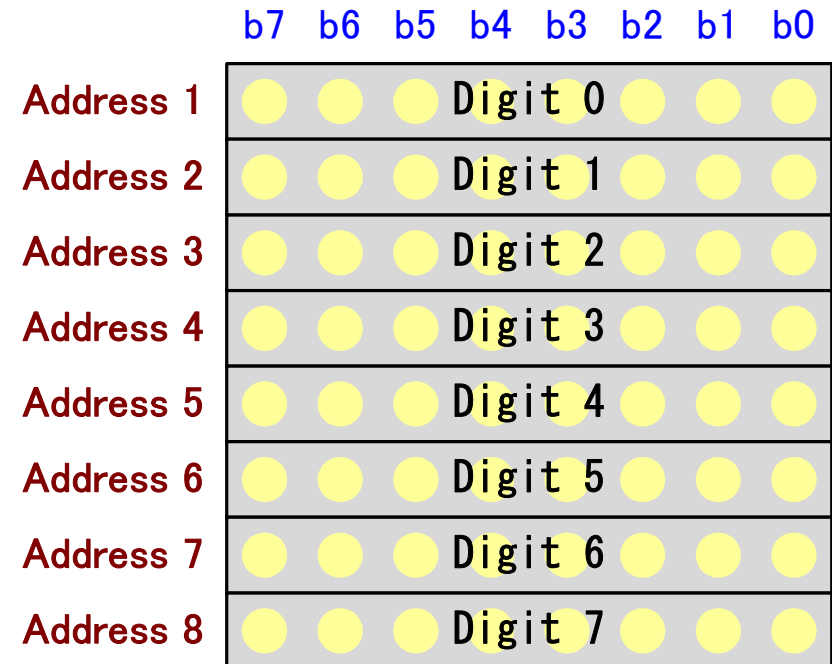


MAX7219を 4段連結している場合のアクセス

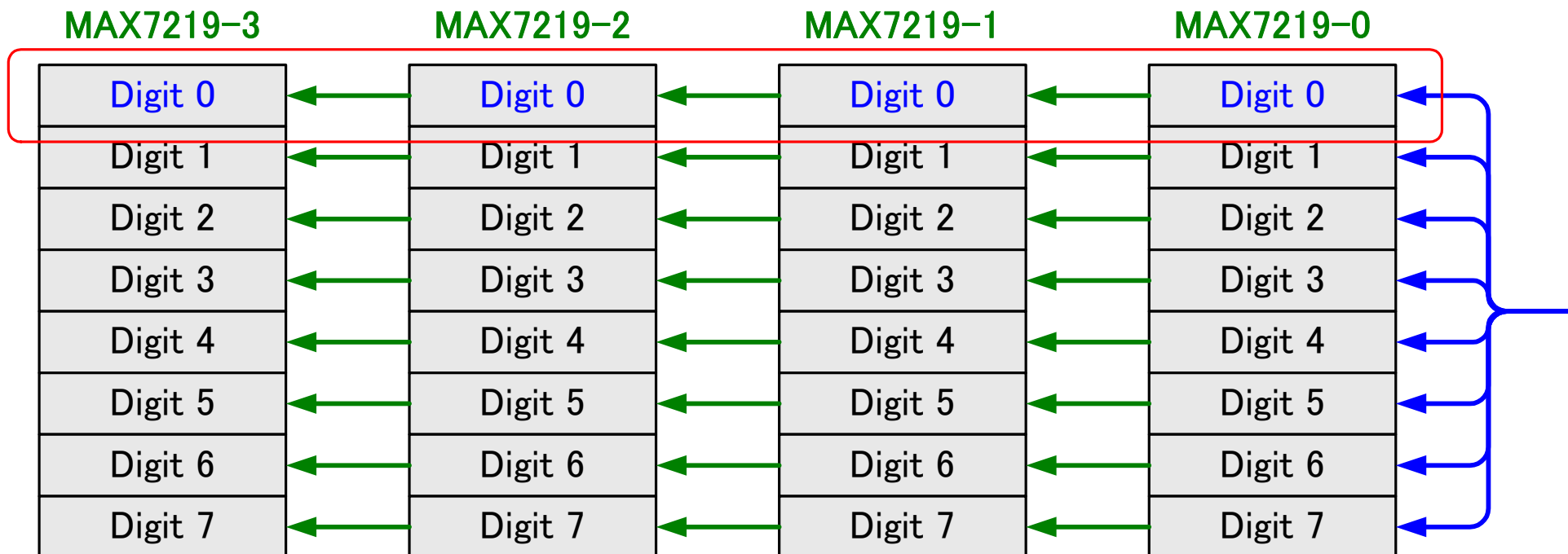
最初は、今回の4連結の 8x8 LEDモジュールの電源回りの配線によるトラブルで、悩みましたが、その後 また別の事で トラブってしまいました。

それは、何かというと、MAX7219を 1個 と、8x8 LEDモジュール1個を組みにしたモジュールであれば、この MAX7219は あっさり動かせると思います。 が、MAX7219を 4段連結されていると、コマンドの出し方、Load信号の出し方で、うまく動作しない場合があります。

それと、最初の説明の 8x8 ドット LEDモジュールの バイト単位で、一度にアクセス出来る範囲は 仮定で、通信コネクタを 右に置いて右端の 縦一列 8ドットが、アクセス単位ではないかと、説明してましたが、間違いでした。一番上の、横一列が、MAX7219の Digit 0 レジスタです。



1個のMAX7219内の データレジスタと、8x8 LED表示器の ドット位置の関係は、上の図のようになります。 次のページに MAX7219を 4段接続したイメージを 図示します。



アクセスの手順は、Loadを Low にする → MAX7219-0/Digit 0 → MAX7219-1/Digit 0 → MAX7219-2/Digit 0 → MAX7219-3/Digit 0 → Loadを Hi にする。という手順が、ひとまとまりの電文になります。上の図では Loadについては 書いて無いですが、赤で 囲った 右から左への 流れになります。次に、Loadを Low にする → MAX7219-0/Digit 1 → MAX7219-1/Digit 1 → MAX7219-2/Digit 1 → MAX7219-3/Digit 1 → Loadを Hi にする。

以上、横一列の 電文を、Digit 7 まで、繰り返す事になります。

MAX7219に 表示データを 転送する手順のフロー

1フレームデータ転送

Digit 0
水平電文 送信

Digit 1
水平電文 送信

Digit 2
水平電文 送信

Digit 3
水平電文 送信

Digit 4
水平電文 送信

Digit 5
水平電文 送信

Digit 6
水平電文 送信

Digit 7
水平電文 送信

Return

Digit n 水平電文 送信

Load = Low

Digit n
16bit データ送信

MAX7219-3 のデータ

Digit n
16bit データ送信

MAX7219-2 のデータ

Digit n
16bit データ送信

MAX7219-1 のデータ

Digit n
16bit データ送信

MAX7219-0 のデータ

Load = Hi

Return

MAX7219の 順番が
-3、-2、-1、-0 の順に
送るのは シフトレジス
タの都合です。