

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。

↑ **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



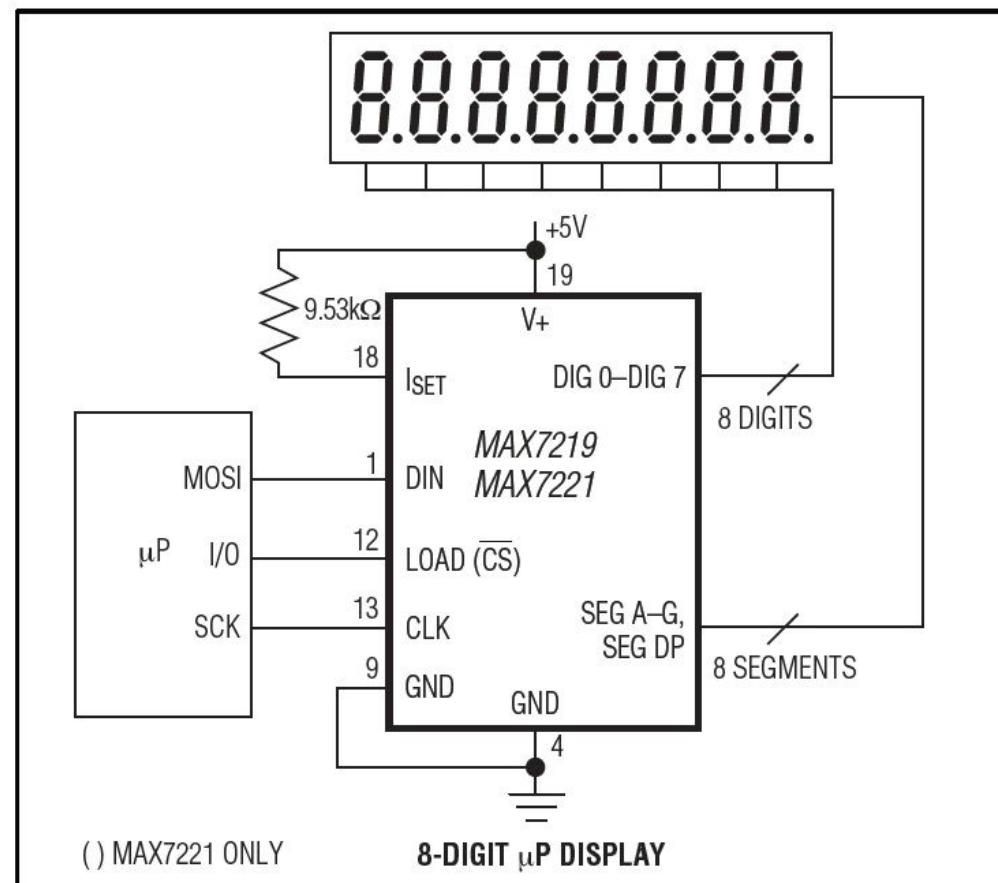
今回購入した
MAX7219 ドットマトリクス 4 in 1 について

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とのインターフェースは、連結可能なシリアルインターフェースのようです。

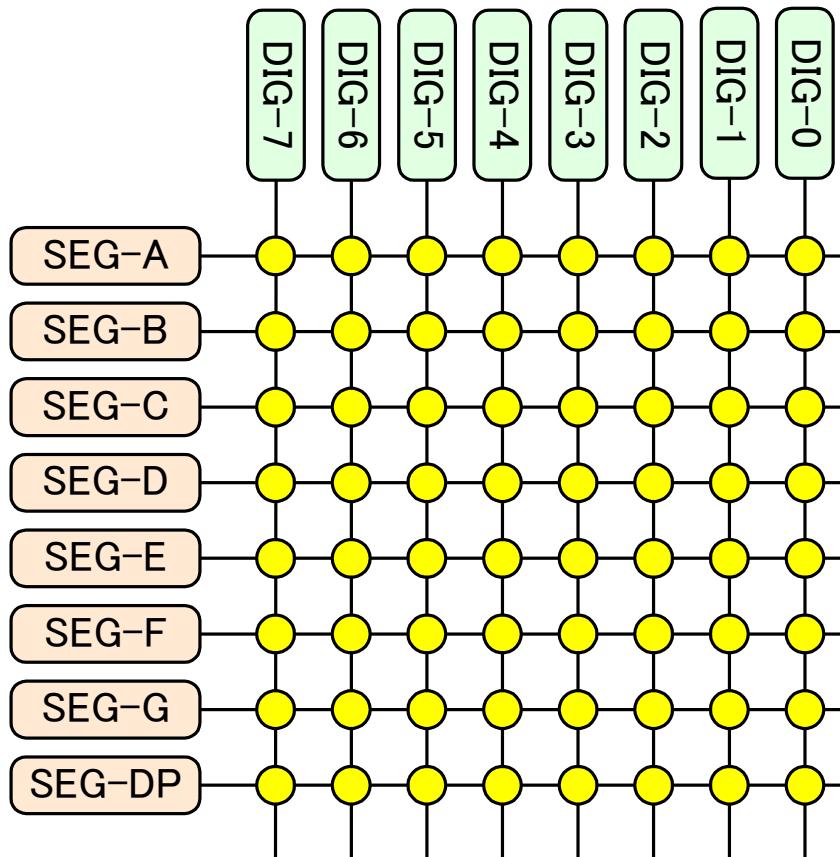
そして、②の基板に ①のLEDモジュールを
4個 ハンダ付けされています。

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、あれつ 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

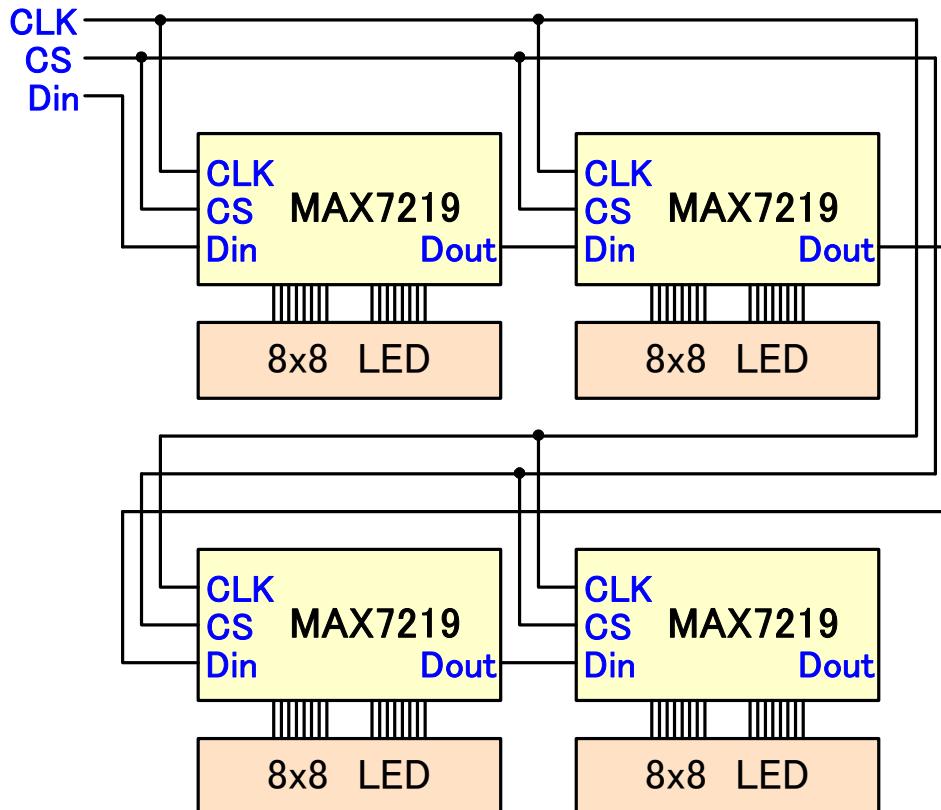
マイコンと接続する信号線は、シリアル通信で3本です。それと、5Pの

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。

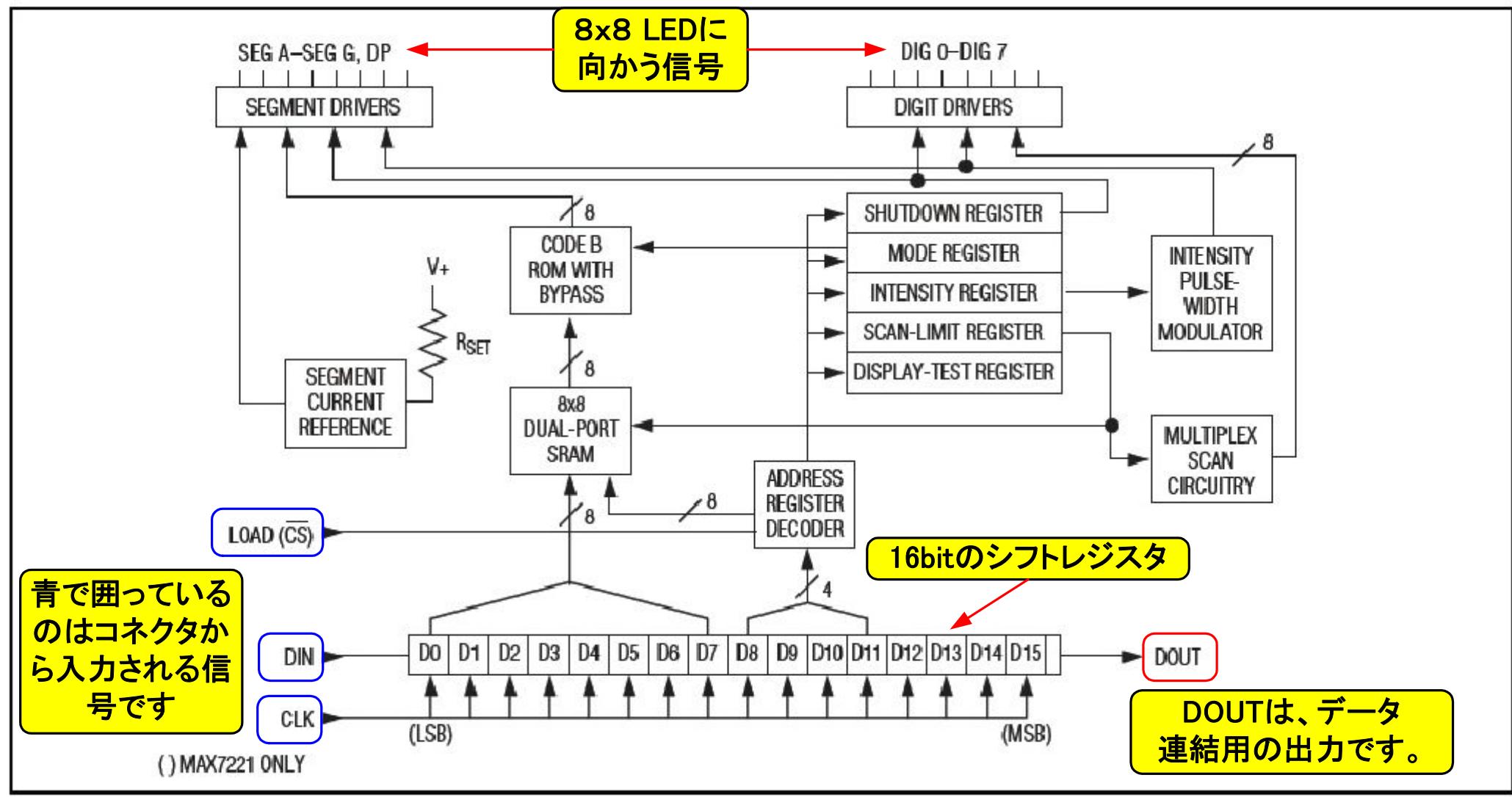
それと、5Pの
コネクタには、
出でません
が **Dout** とい
う信号線があ
ります。これ
は、先頭の
MAX7219から
2番目の

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。



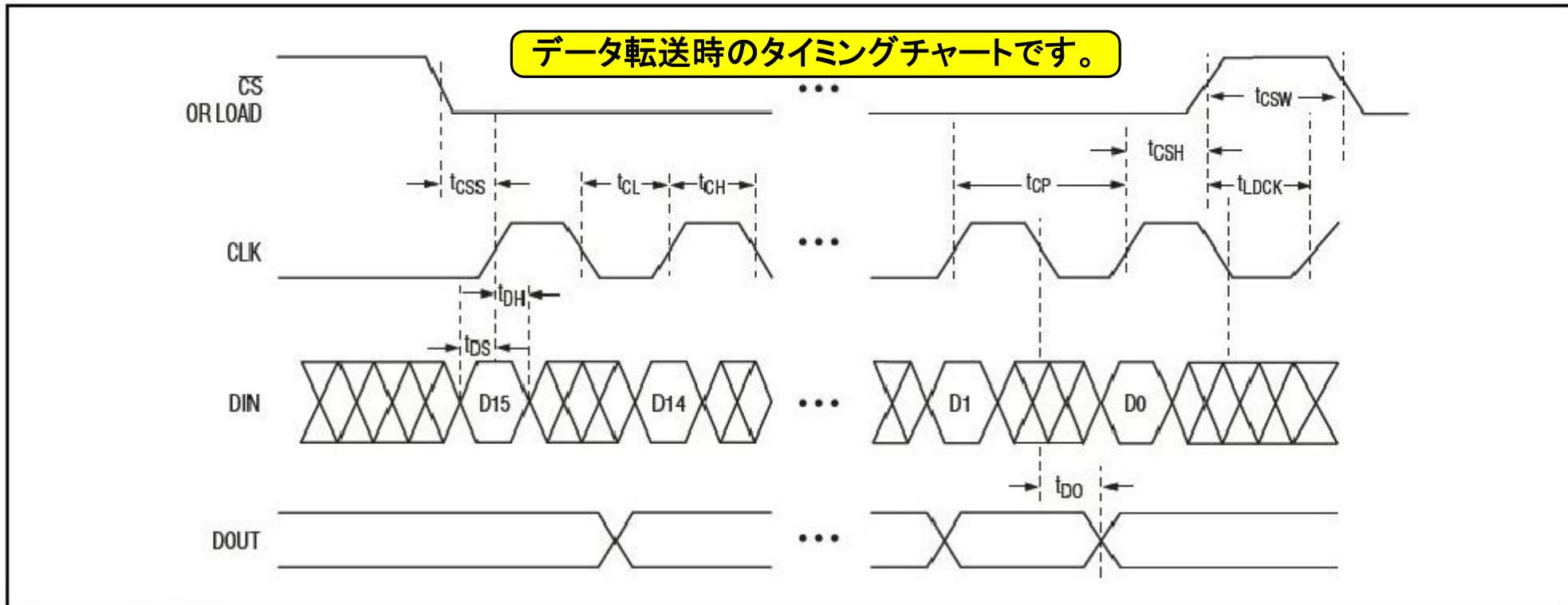


図1. タイミング図

表1. シリアルデータのフォーマット(16ビット)

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。

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

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います。

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	0XA
Scan Limit	X	1	0	1	1	0XB
Shutdown	X	1	1	0	0	0XC
Display Test	X	1	1	1	1	0XF

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						
3	5	P4_6、P4_7は 水晶で使用					

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

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。

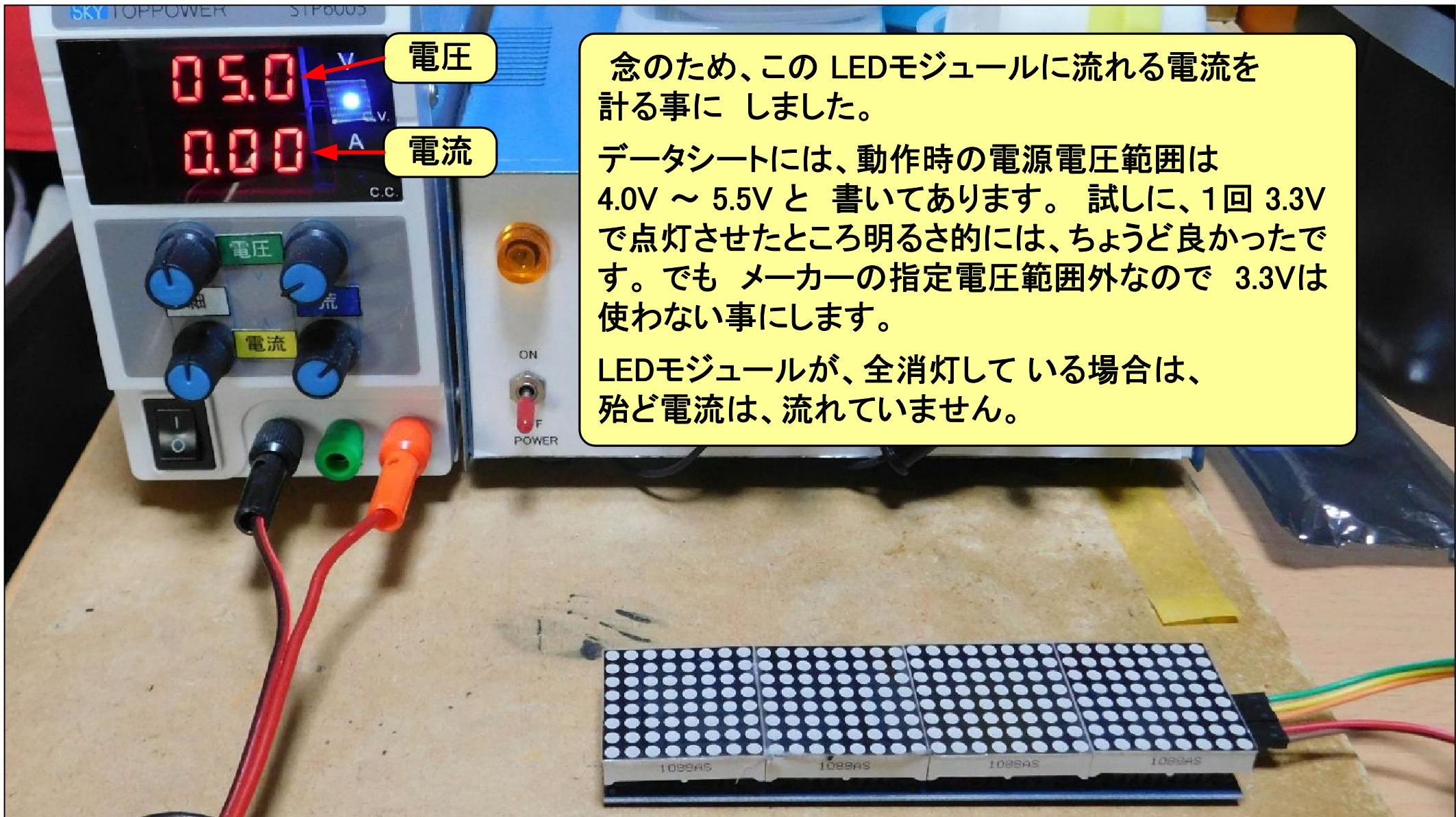
そして、ひとまとめのアクセス単位として
16ビットのシリアルデータが、あります。

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
アセンブラーで 作成します。

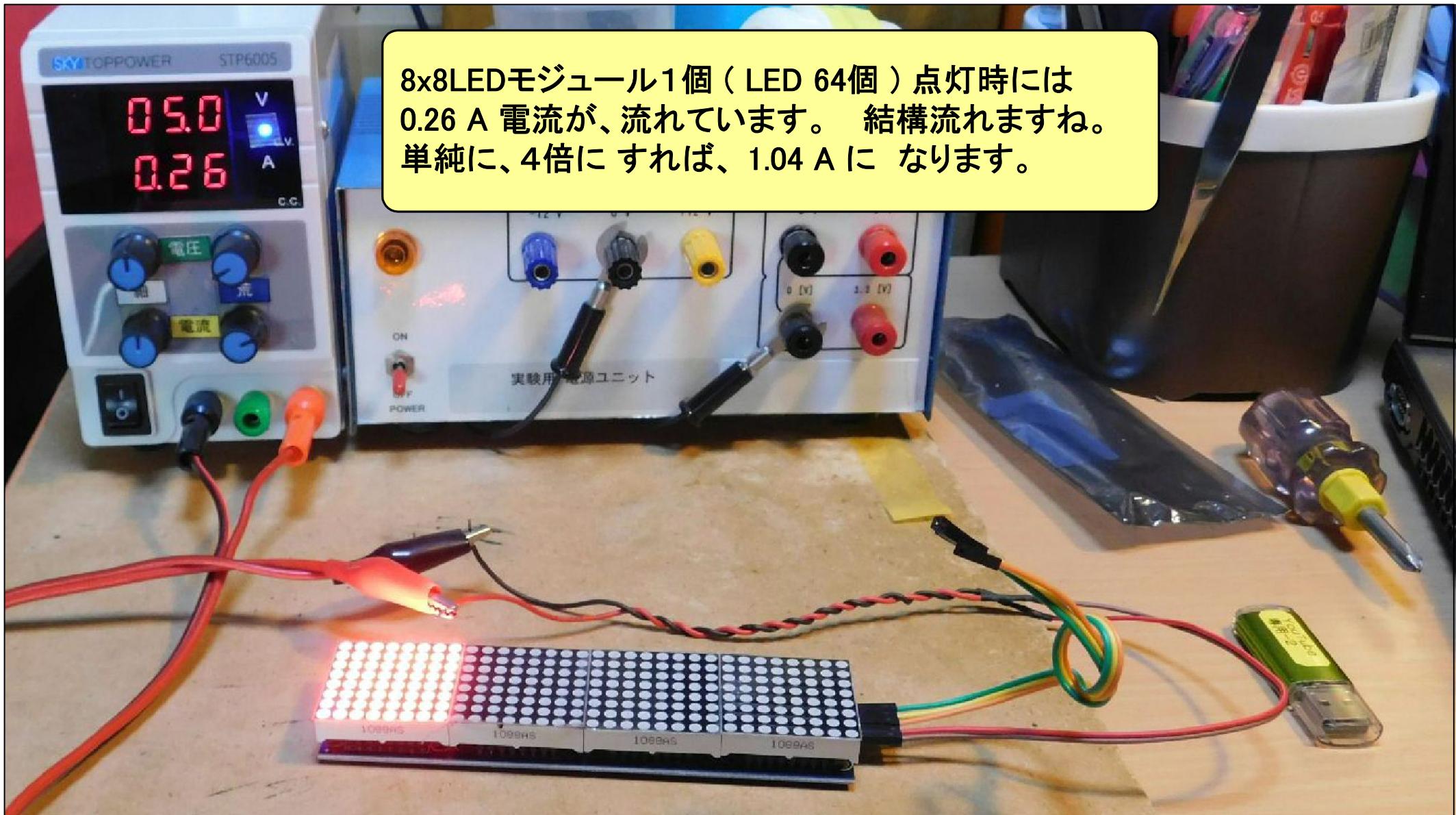
何故、アセンブラーで作成するかというと、ルネサスHEWの環境で R8Cの プログラムを作成する場合、I/O周りの こま回りする処理の場合、Cで作るより、アセンブラーで作成した方が、約 3倍ほど 実行速度が 早くなるからです。

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。

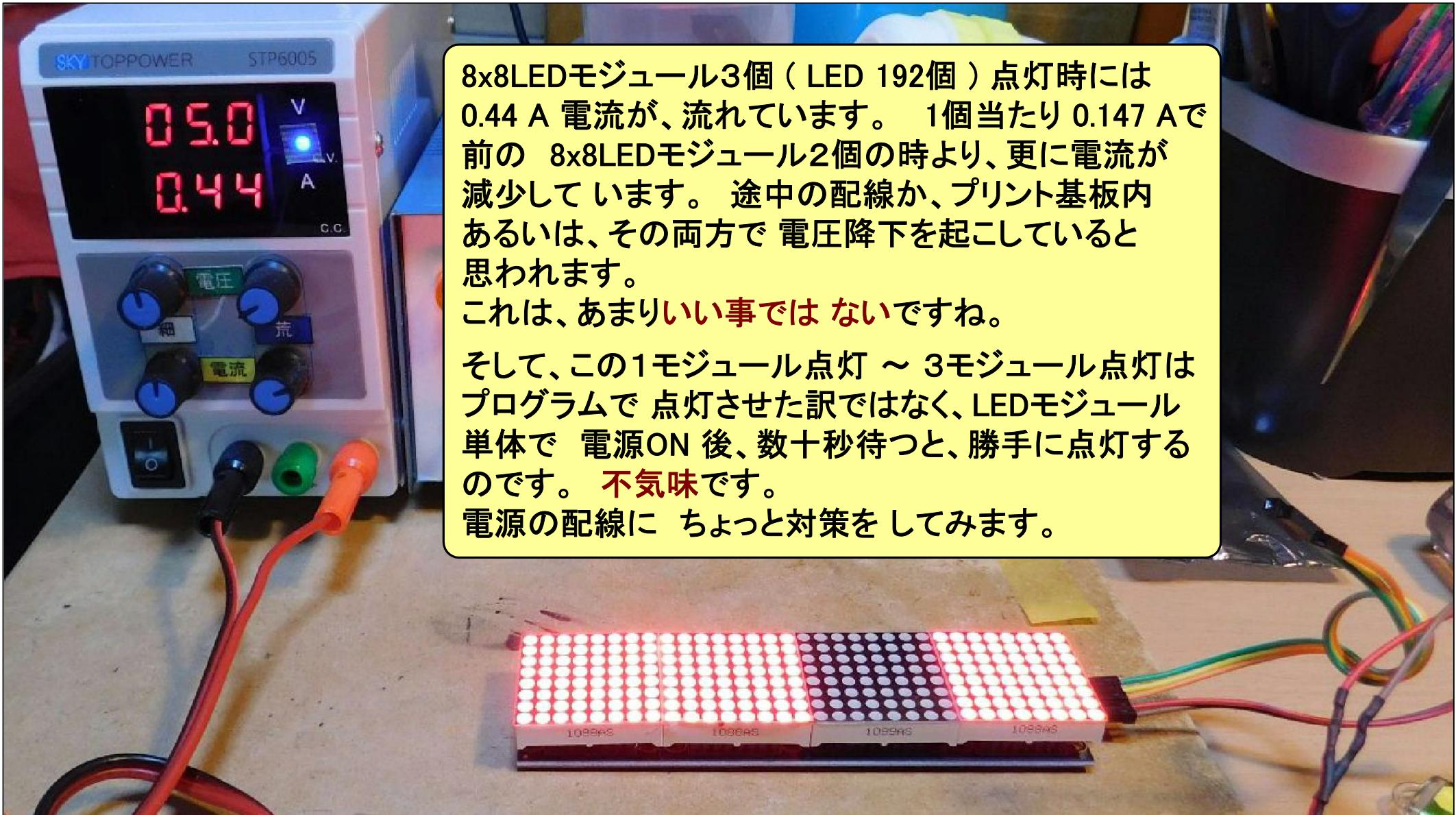
次は、実験の動画を お見せします。



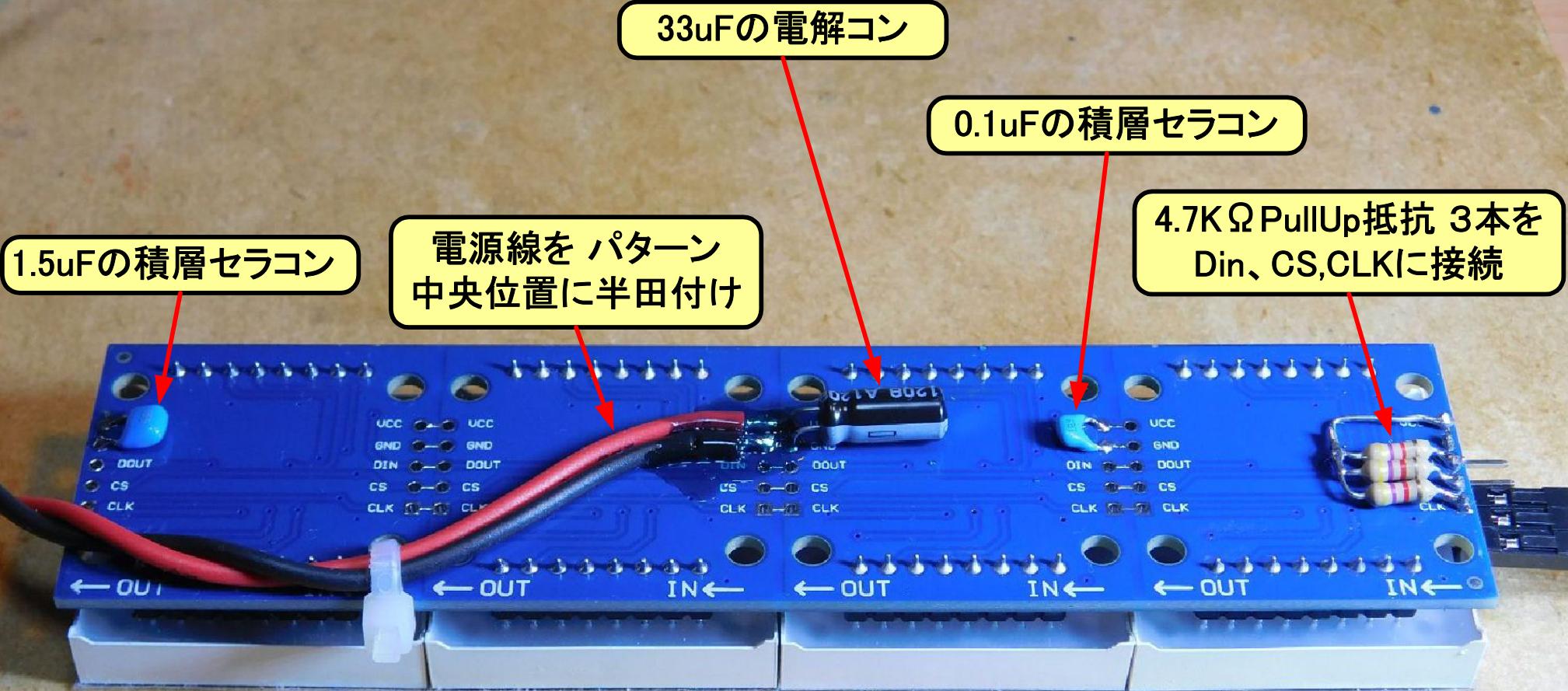
8x8LEDモジュール1個（LED 64個）点灯時には
0.26 A 電流が、流れています。結構流れますね。
単純に、4倍にすれば、1.04 A になります。

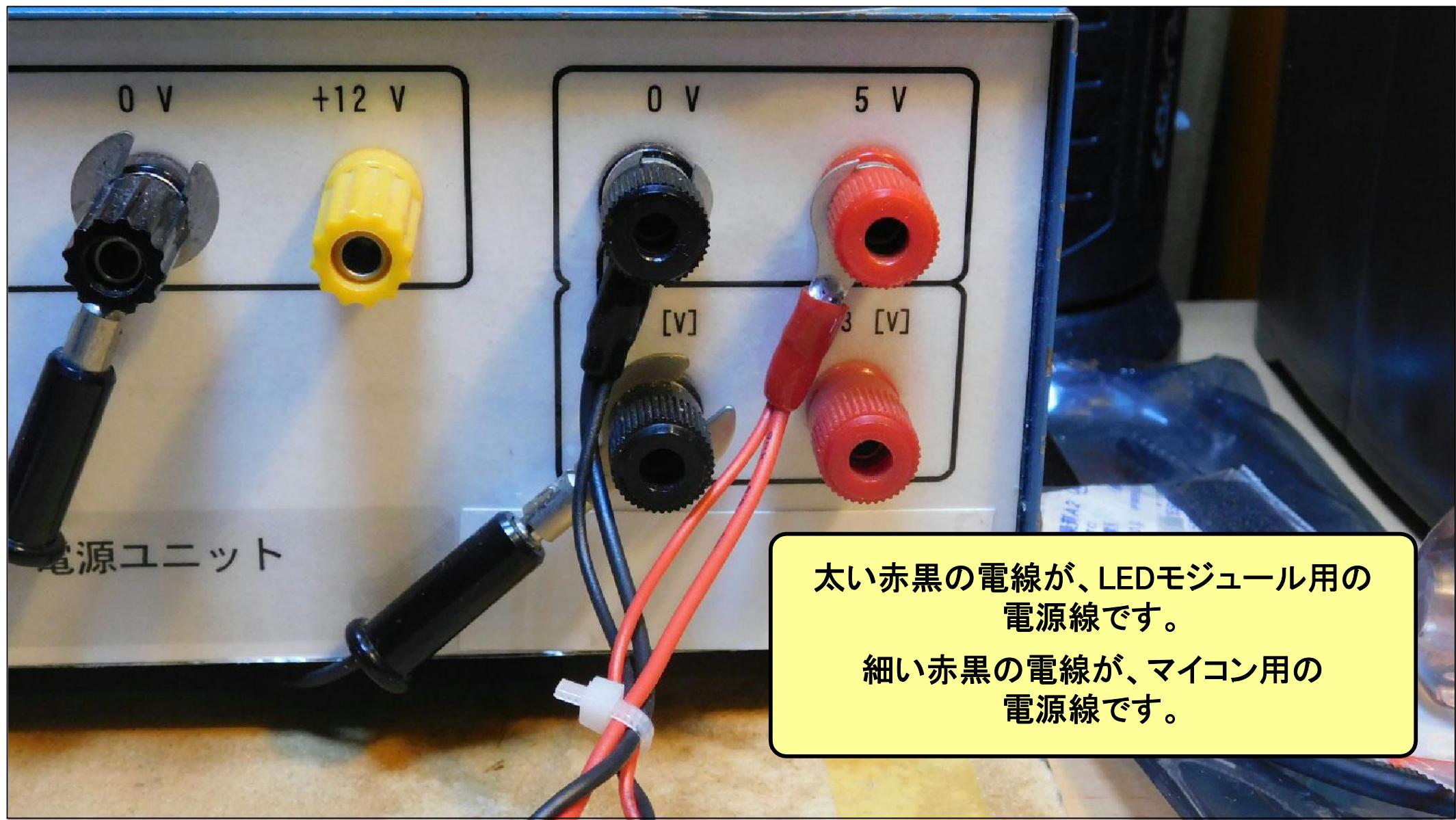




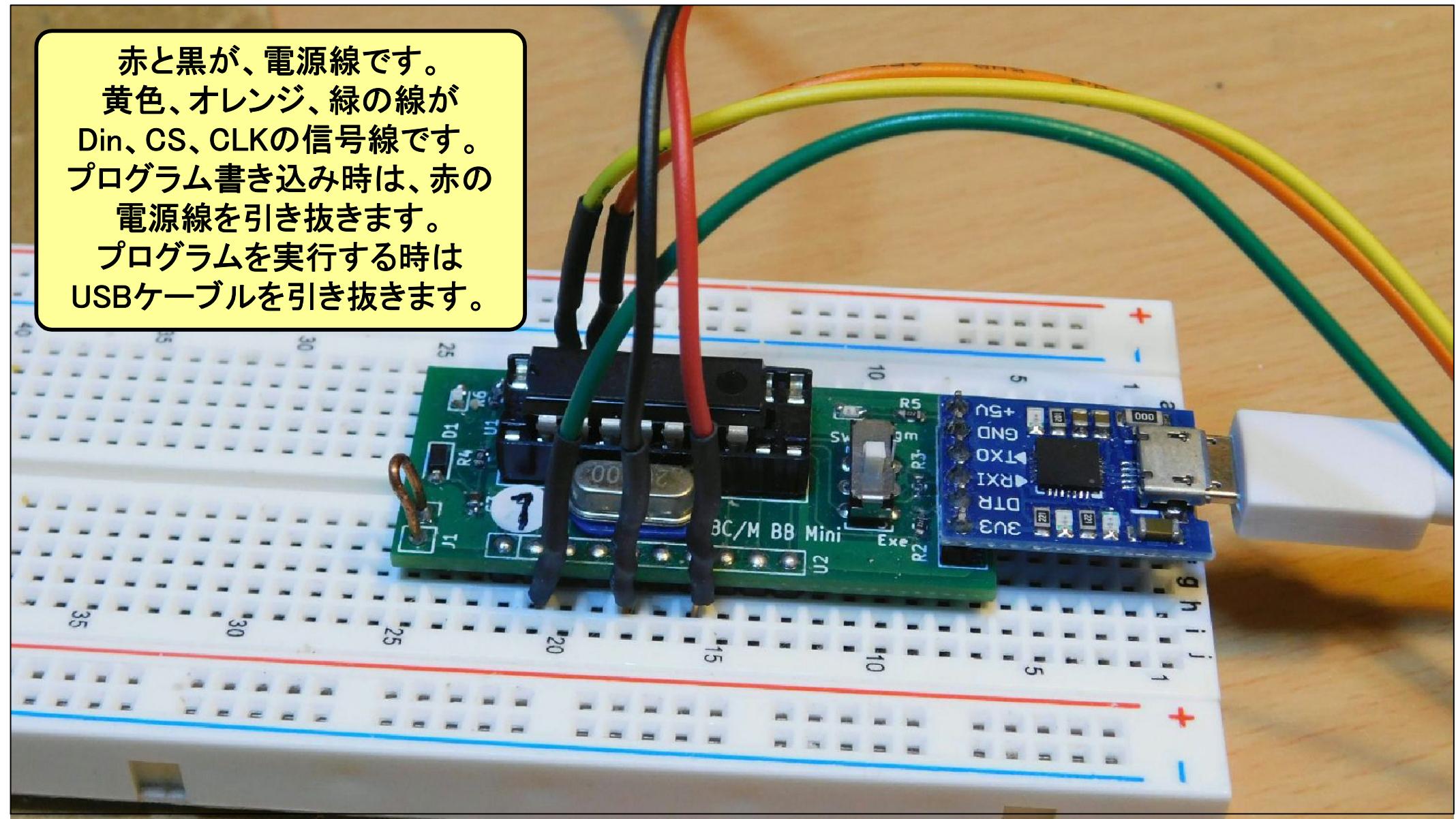


電源回り、その他改修





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。

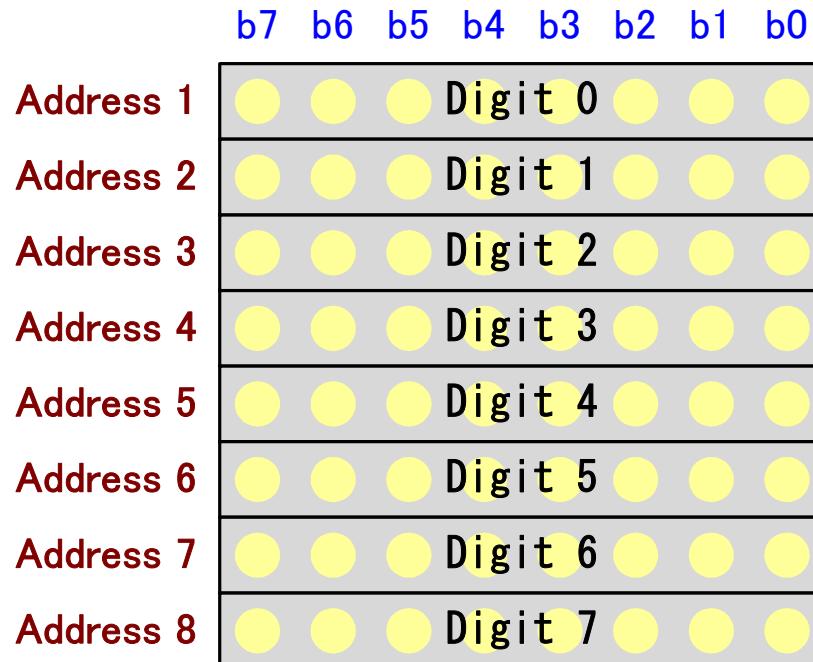


MAX7219を 4段連結している場合のアクセス

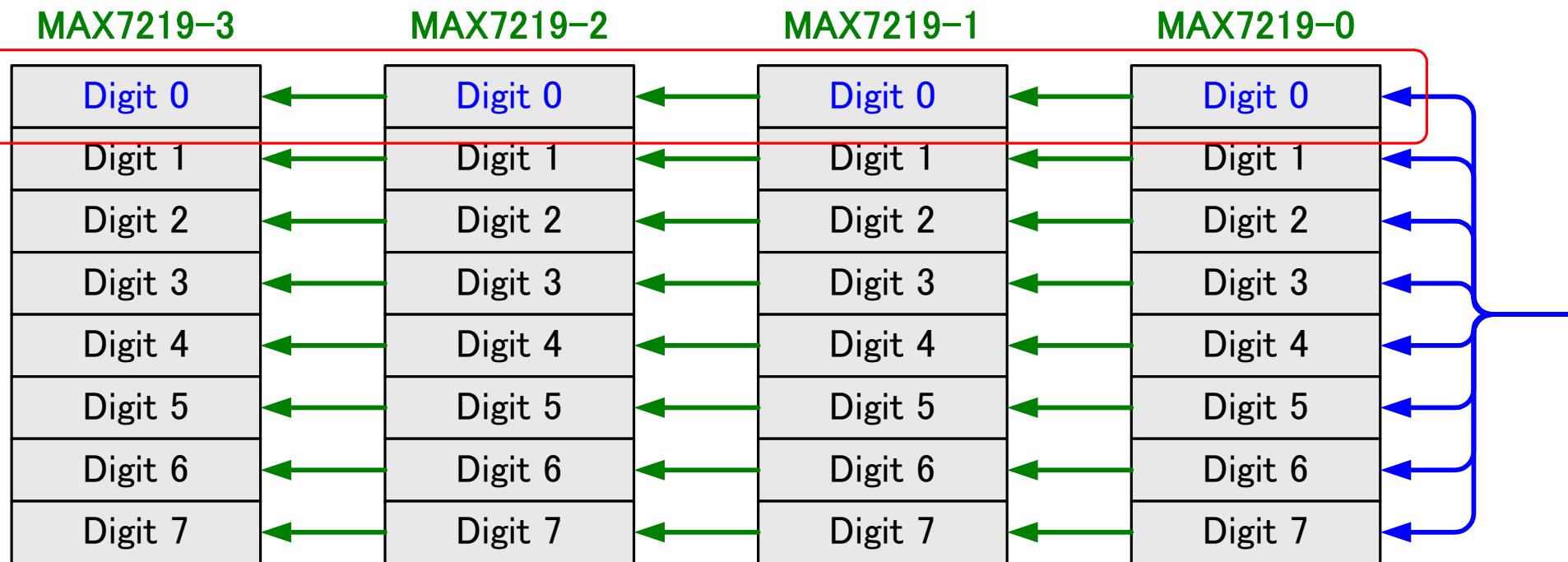
最初は、今回の4連結の 8x8 LEDモジュールの電源回りの配線によるトラブルで、悩みました。その後 また別の事で トラブルてしまいました。

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、間違いました。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、ひとまとめの電文になります。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 n 水平電文 送信

Load = Low

Digit n
16bit データ送信

Digit n
16bit データ送信

Digit n
16bit データ送信

Digit n
16bit データ送信

MAX7219-3 のデータ

MAX7219-2 のデータ

MAX7219-1 のデータ

MAX7219-0 のデータ

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。

Load = Hi

Return