

# DMAを使ったLEDディスプレイ表示実験

田仲 健治

2018.4.29

# きっかけ

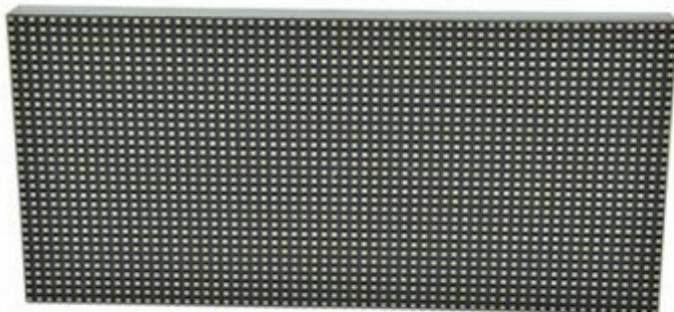
- 最近、街中を歩くと店の前で宣伝用のLEDディスプレイを見かけることが多くなった。
- しかし、自分が気になるのは宣伝内容ではなく、LEDディスプレイのほうで、とにかく気になって仕方がない。
- 何度も自分で作ろうとしたのだが、縦横のLED数がちょっと増えるだけで、かなりコストがアップするので躊躇。
- ところが、Amazonで完成品が安かったのでとうとう買ってしまった。64×32ドット、RGBカラーでなんと税込み3,299円、しかも送料無料、10%ポイント還元！

# 購入LEDディスプレイ

P3 RGBピクセルパネルHDビデオディスプレイ64×32ドットマトリックスSMD LEDディスプレイモジュール192×96mm

<https://www.amazon.co.jp/gp/product/B071VJFB86>

 お客様は、2018/4/7にこの商品を注文しました。  
この注文を表示



P3 RGBピクセルパネルHDビデオディスプレイ64×32ドットマトリックス  
SMD LEDディスプレイモジュール192×96mm

YANGTAO

★★★★★ 2件のカスタマーレビュー

価格: ¥ 3,299 通常配送無料 詳細

ポイント: 330pt (10%) 詳細はこちら

在庫あり。在庫状況について

4/30 月曜日にお届けするには、今から21 時間 9 分以内にお急ぎ便を選択して注文を確定してください  
(Amazonプライム会員は無料)

この商品は、YANGTAOが販売し、Amazon.co.jp が発送します。この出品商品にはコンビニ・ATM・ネット  
バンキング・電子マネー払いが利用できます。ギフトラッピングを利用できます。

新品の出品: 1 ¥ 3,299より

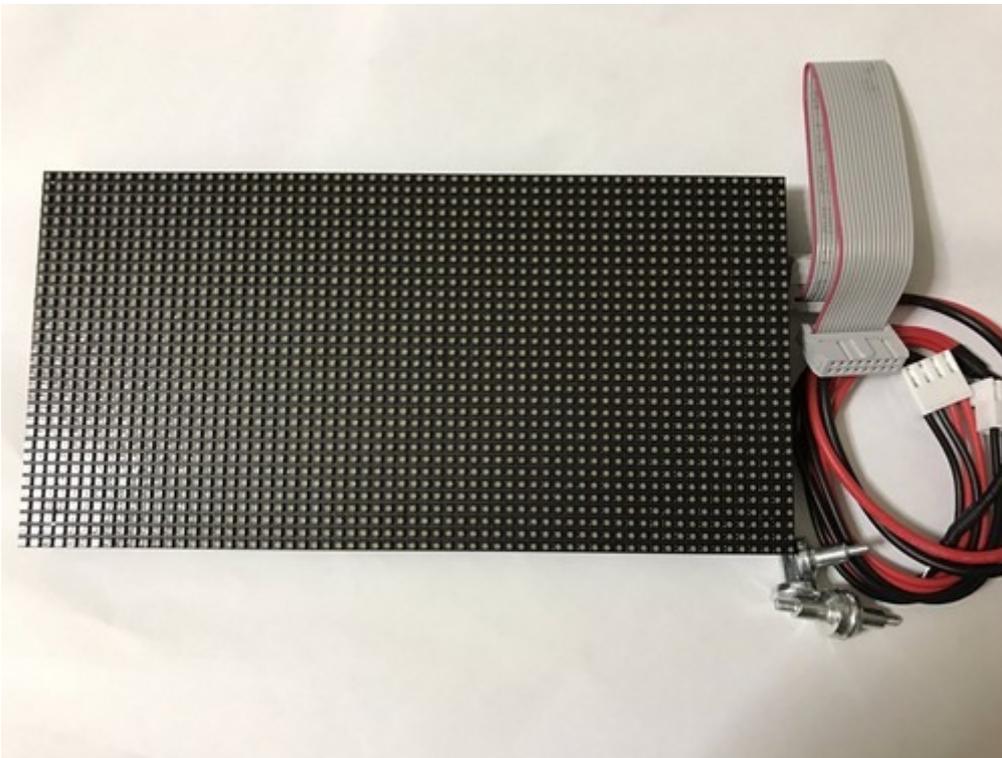
- サイズ: 192x96mm
- 解像度: 64 \* 32
- スキャン: 1/16
- 明るさ: 2200CD /平方メートル
- ハブのタイプ: 75

> もっと見る

画像にマウスを合わせると拡大されます

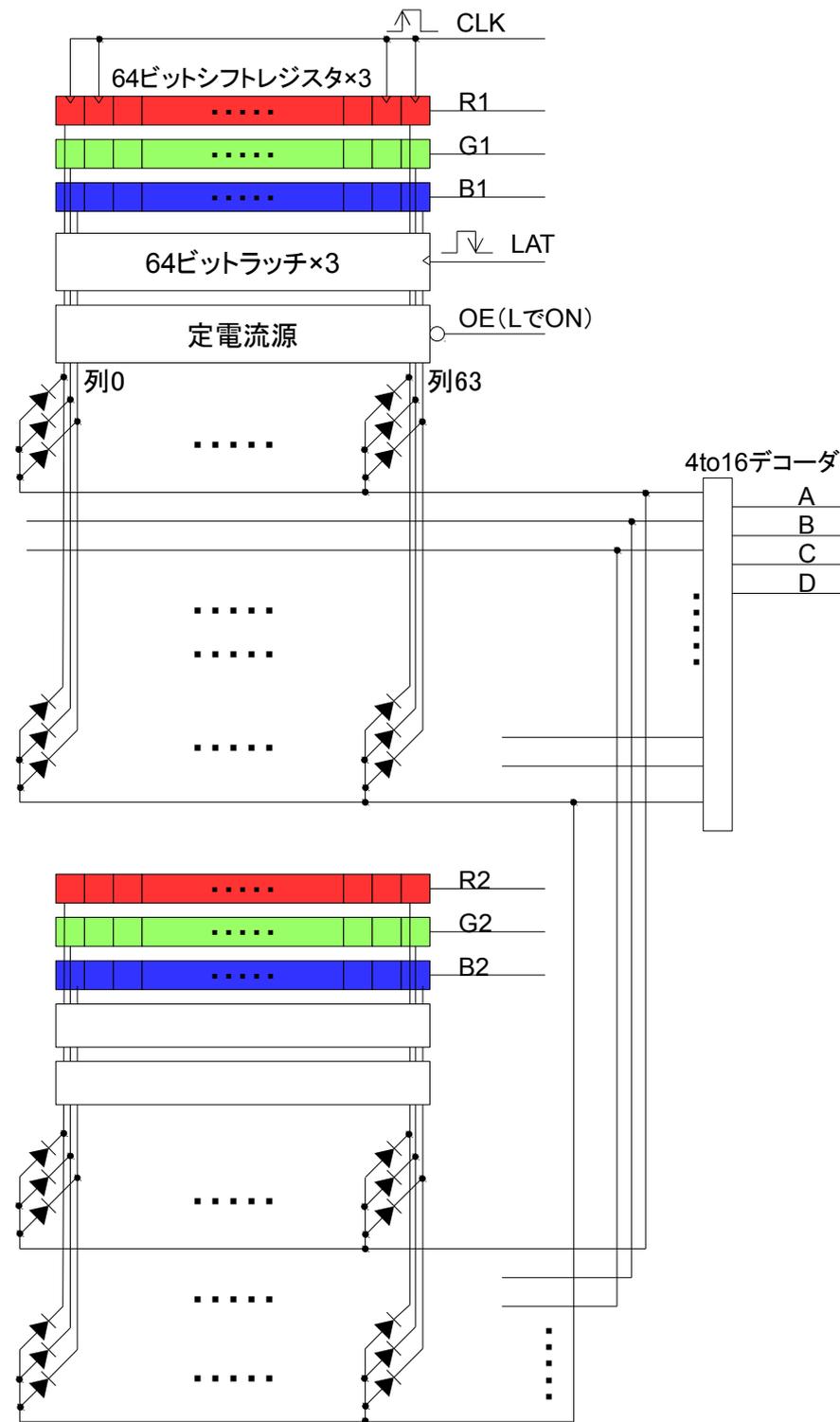
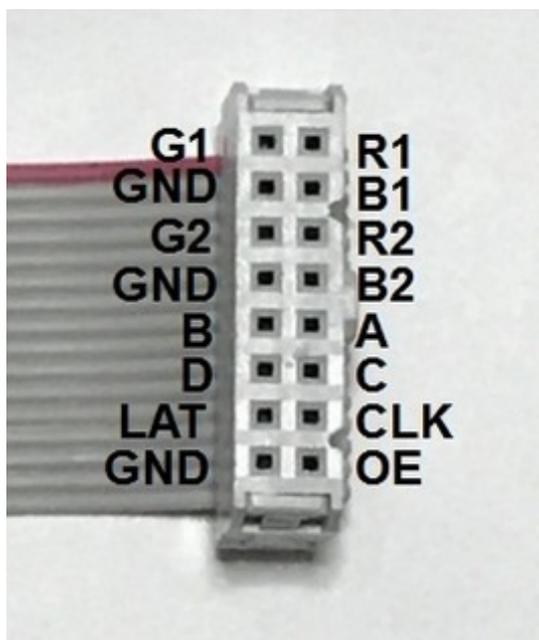
# LEDディスプレイ到着

予想されたことだが、説明書は一切なし。  
Webに豊富に情報あるが、64×32の解説は見当たらず。  
とりあえず32×16ドット用のArduinoサンプルで点灯することを確認できたので、あとは自力で解析。



# 購入したLEDディスプレイの調査結果

上下16行ずつ分割されている。  
 横方向にシフトレジスタ、縦方向にデコーダ。  
 マイコンは内蔵されておらず、外付け制御回路で  
 ダイナミック点灯しなければならない。



# LEDディスプレイ点灯方法

## (1)シフトレジスタにデータを送信

R1/G1/B1、R2/G2/B2の6ビットの信号を出力し、CLKにパルスを与えることで1番目の色を決定。  
R1/G1/B1が上半分、R2/G2/B2は下半分の表示用信号。CLKは立ち上がり時にシフトするよう。  
これを64回繰り返し、全てのシフトレジスタに任意の値を書き込み。

## (2)OEをHIにしてLEDを消灯

瞬間的な表示の乱れを防ぐため、一瞬全てのLEDを消灯。

## (3)A/B/C/Dの4ビット信号で表示行を決定

Aが最下位、Dが最上位の4ビット信号を出力し、表示する行(セクション)を決定。  
セクションは上半分と下半分の同じ行が同時に指定される。

## (4)シフトレジスタからラッチに転送

LATにパルスを与えることで上下それぞれのラッチにシフトレジスタからデータが転送される。  
LATは立ち下がり時に転送されるよう。

## (5)OEをLにしてLEDを点灯

これで上下各1行分のLED表示が行われる。

## (6)一定の時間を待つ

人間の目に残像として残すため、LEDを点灯させてから少し時間をおく。

以上の(1)から(6)を繰り返し、シフトレジスタへの送信信号と表示行を高速で変更していくことで、画面全体に対し任意の表示をさせることができる。

# 濃淡の表現方法

濃淡を表現するには、全ての行のLEDを高速で点灯、消灯を繰り返す必要あり。

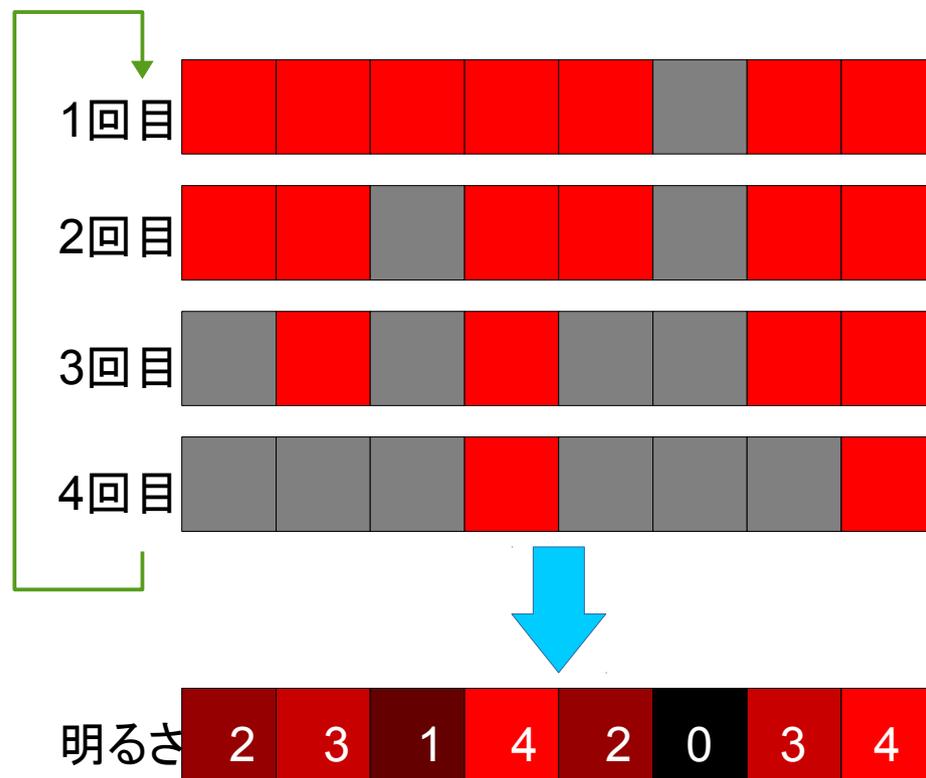
しかし、マイコンのPWM出力機能も使えず、色数を増やすと結構な力技となり、CPUパワーをかなり消費することになる。

目標

RGB各16色 = 4096色相当

60フレーム/秒

右図でいうと15回目まで必要！



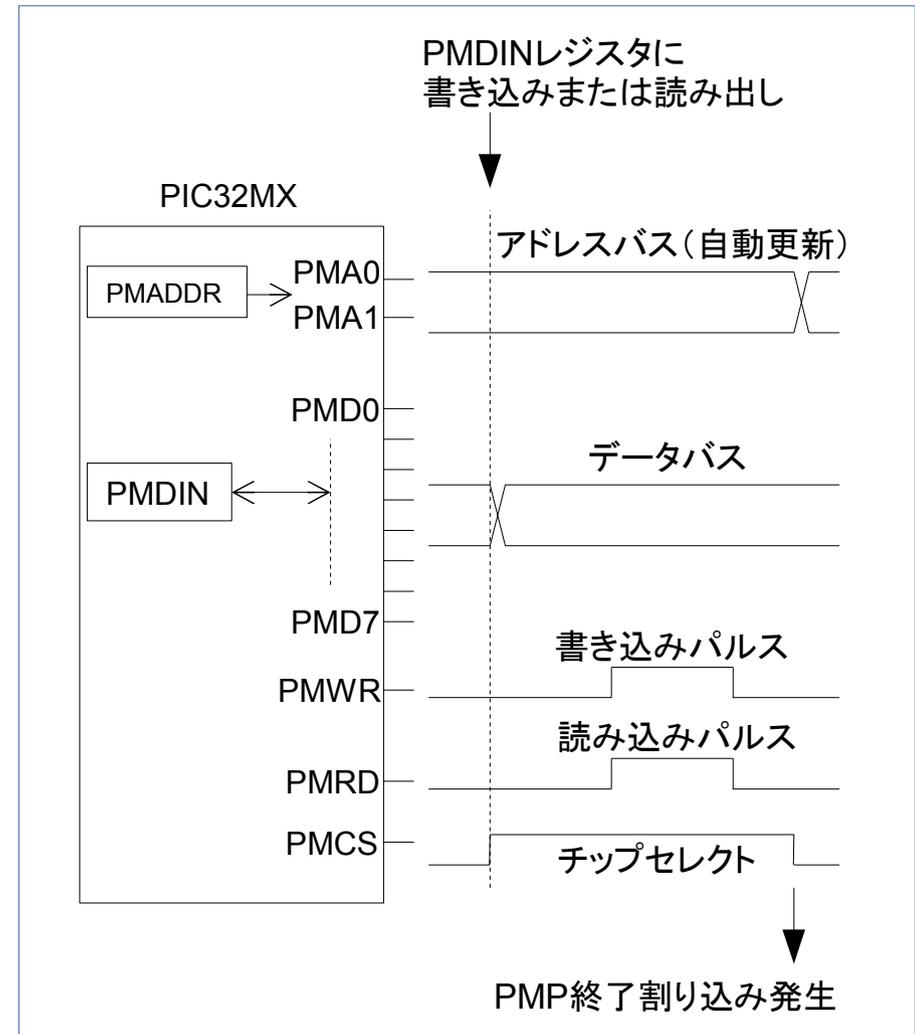
# DMAとPMP

CPUパワーをなるべく使わないようにするために、DMAの使用を検討。

本ディスプレイのインターフェイスは、共通CLKのシフトレジスタが6本もあり、シリアルのようにシリアルではないので、パラレルマスターポート(PMP)をDMAで駆動することにした。

PMPのWR信号をCLKにつなぎ、パラレルデータ転送。

DMA転送するため、通常のビデオメモリのほか、全画面分の出力データをRAM上にあらかじめ展開。



# DMAとPMP

タイマー周期

15回/行 × 16行 × 60フレーム/s = 14.4KHz

## DMAチャンネル初期設定

転送先アドレス : PMDIN

転送先サイズ : 1

転送元アドレス : LED用バッファ  
(後で設定)

転送元サイズ : 64(横1行分)

転送セルサイズ : 1

(1バイトごとに転送終了)

転送開始イベント : PMP割り込み

ブロック転送終了で割り込み発生  
(64バイト転送終了時)

DMAによる転送開始と  
終了の繰り返し  
この間CPUの介在なし

タイマー割り込み

転送元アドレス設定

DMAチャンネル有効化

1バイト目転送  
(PMDINに書き込み)

タイマー割り込み終了

PMP終了割り込み⇒  
DMA転送開始イベント

1バイト転送  
(PMDINに書き込み)

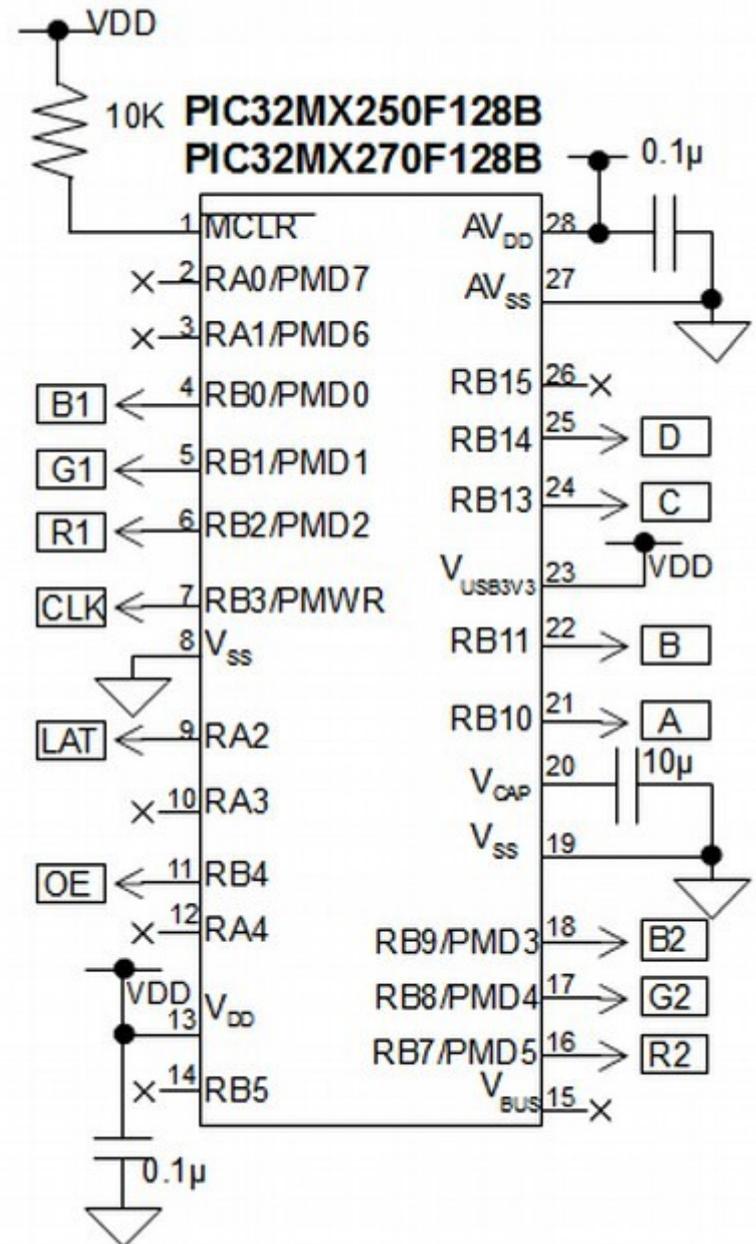
ブロック転送終了割り込  
みでラッチに取り込み  
LED点灯

# 回路図

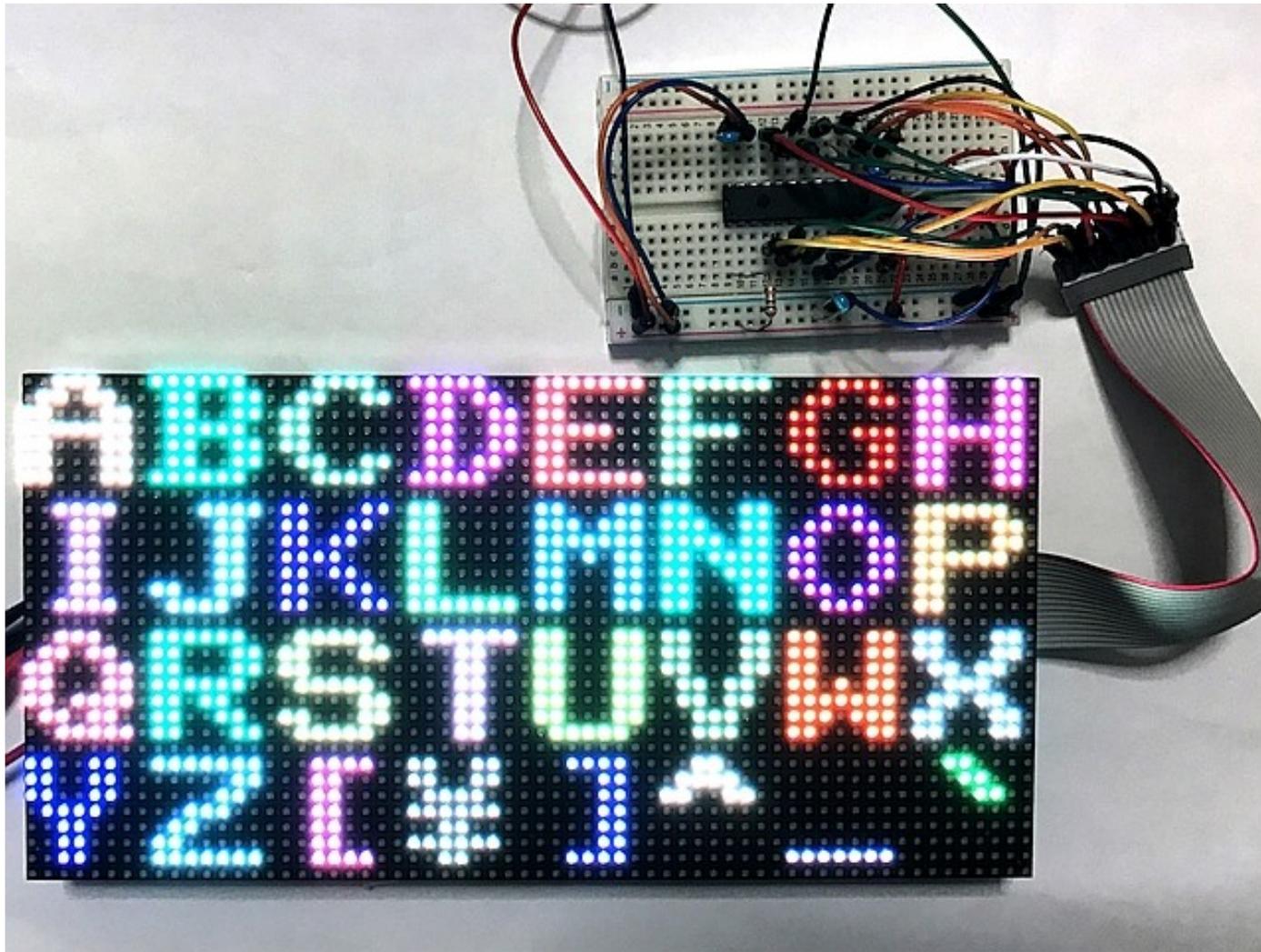
マイコンはPIC32MX250F128B  
を使用。

RAM容量32Kバイト

内蔵クロックで48MHz動作



# 実験結果



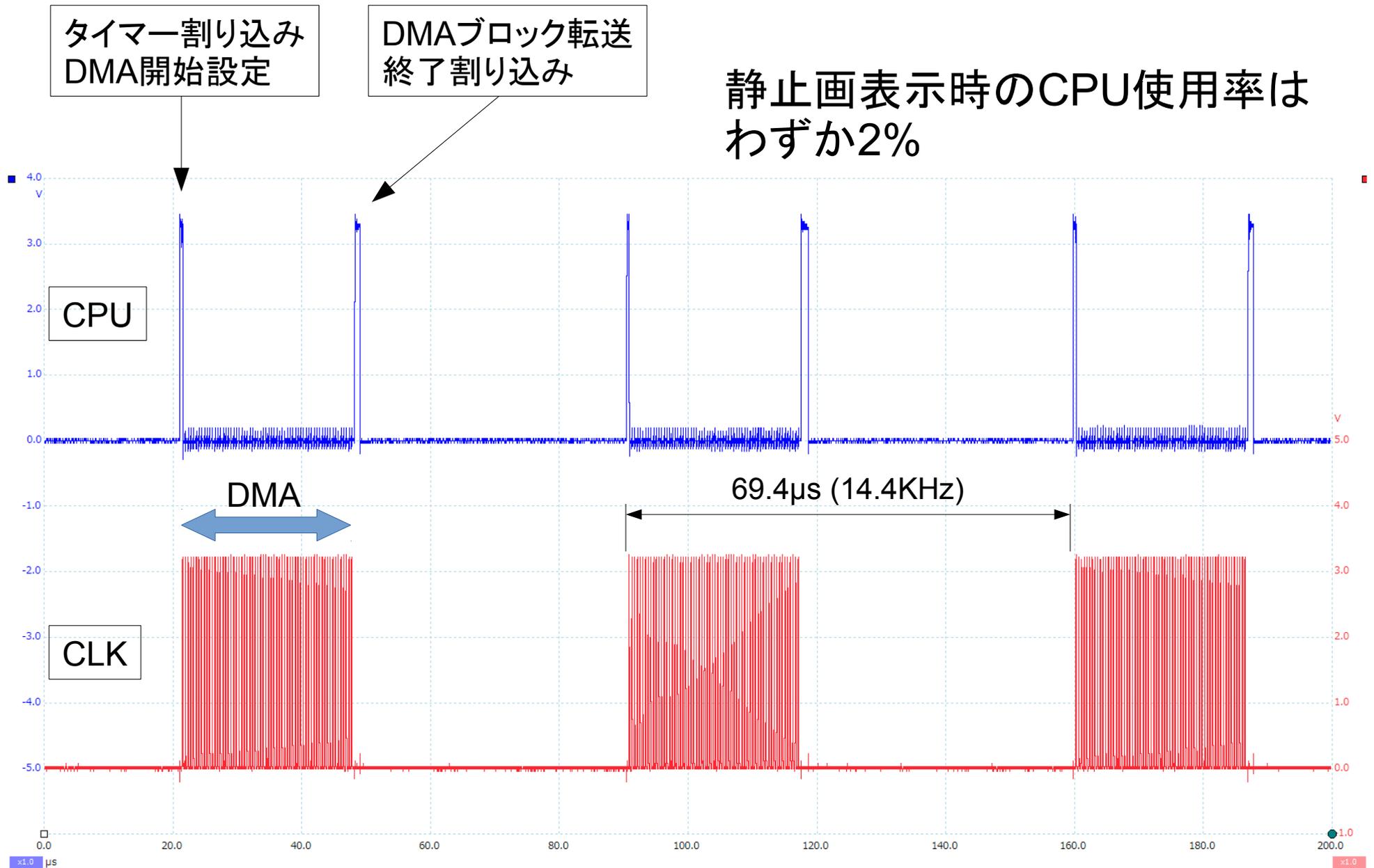
60フレーム/秒のおかげで、チラつきもなし

# 実験結果



写真ではわからないけど、かなり眩しい

# 静止画表示時のCPU使用率



趣味の電子工作とプログラミング

ケンケンのホームページ

<http://www.ze.em-net.ne.jp/~kenken/index.html>

ツイッターアカウント @KenKenMkIISR