

電光掲示板

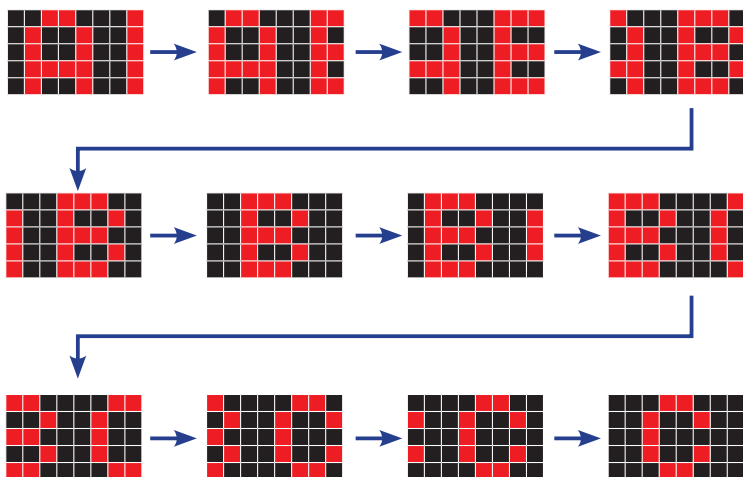
02-1

動作説明

LED DOT MATRIX を利用した電光掲示板

※ PIC16F627A は I/O ピンの数が限られるため、8 × 5 のみ利用

A ～ F のアルファベットが右から左に流れる



A ～ F の点灯パターン例

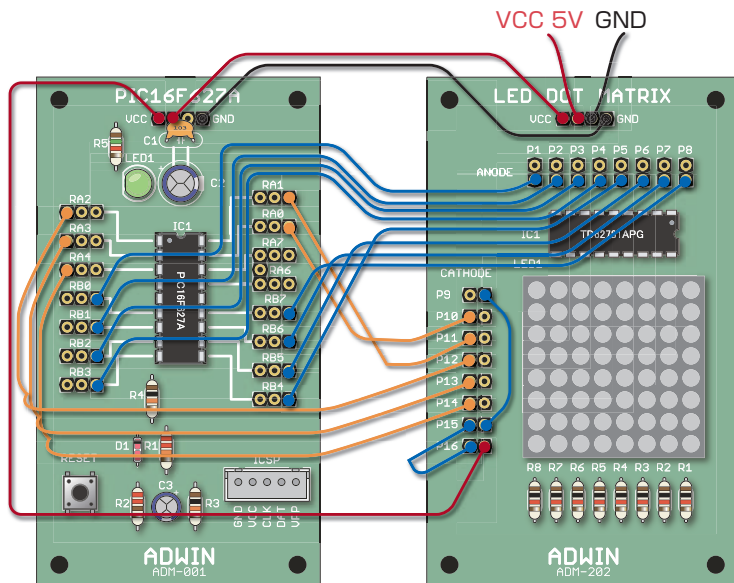


02-2

配線

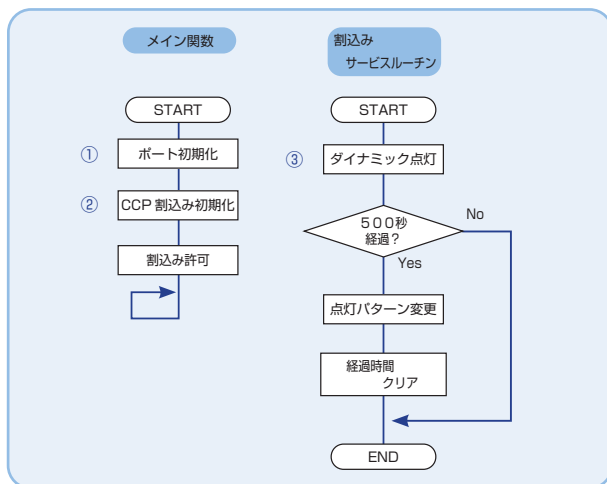
使用ボードNo.

ADM-001・ADM-202



02-3

概略フローチャート



02-4

概略フローチャート解説

① 使用する I/O ポートの設定

PORTB : 出力ポート
RA0 ~ RA4, RA6 ~ 7 : 出力ポート

② CCP1 割り込みを使い 1m 秒ごとに割り込み発生
<設定内容>

CCP モジュール

- ・モード : CCP 割り込み
- ・周期 : 1000 μ 秒

タイマ 1

- ・クロック源 : 内部クロック
- ・プリスケアラ : 1

③ ダイナミック点灯
後述

④ 500m 秒ごとに点灯パターンを変更する。



02-5

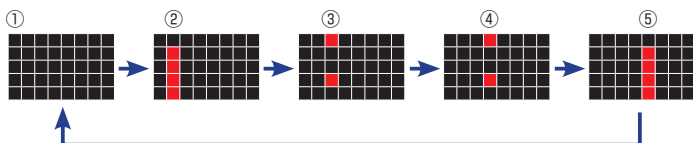
ダイナミック点灯

LED DOT MATRIX は複数の LED のアノード、カソードが共通のため、文字表示などを行う場合にはダイナミック点灯制御を行う必要がある。



LED DOT MATRIX を 1 列ずつ順に表示していく

←消灯
←点灯

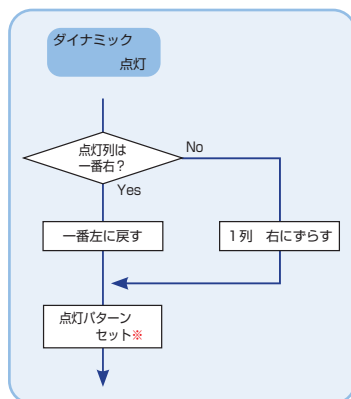


これを高速で繰り返すことにより、人間の目にはアルファベットの A と表示し続けているように見える。



ただし、あまりに速く次の列の点灯にうつると、光が薄く見えるため、時間の調整が必要である。
今回は CCP 割込みを利用し、1m 秒おきに次の列の点灯に移ることとした。

※ 上記の①～⑤のように点灯させること



02-6 ソースコード

掲載ソースは弊社 HP で公開中のエレモサンプルソースからの抜粋です。
(URL : https://www.adwin.com/image/support/ADM_SampleSource.zip)

点灯パターン配列

```
// 文字コード用配列
// LED DOT MATRIX のアノード側 ( PORTB 側 ) を一列ずつ H レベルにしなが
// RA0 ~ RA4 にこの配列の値を格納する
const unsigned char a_code[MAX_CODE] = {
    0b00000, 0b11110, 0b01001, 0b01001, 0b11110, 0b00000, // A
    0b00000, 0b11111, 0b10101, 0b10101, 0b01010, 0b00000, // B
    0b00000, 0b01110, 0b10001, 0b10001, 0b01010, 0b00000, // C
    0b00000, 0b11111, 0b10001, 0b10001, 0b01110, 0b00000, // D
    0b00000, 0b11111, 0b10101, 0b10101, 0b10001, 0b00000, // E
    0b00000, 0b11111, 0b00101, 0b00101, 0b00001, 0b00000, // F
};
```

ダイナミック点灯と点灯パターン変更

```
/*
  割込みサービスルーチン
  1m 秒ごとに CCP 割込み
*/
static void interrupt isr(void){

  static unsigned int cnt = 0;    // 500m 秒ごとに文字を流すための変数
  static unsigned int index = 0;  // LED DOT MATRIX の表示列指定用
                                   // 0(1 列目 : 一番左) ~ 7(8 列目 : 一番右)
  static unsigned int codeIndex = 0; // 流す文字列 A ~ F のどこから表示するか指定用

  /* ダイナミック点灯 */
  PORTB = ( 0x01 << index ); // ダイナミック点灯
                               // LED DOT MATRIX の P1 ~ P8 を順に H レベルに設定

  // F の次は A を表示
  if( ( codeIndex + index ) > ( MAX_CODE - 1 ) ){
    PORTA = ~a_code[codeIndex+index-MAX_CODE];
  }
  else{
    PORTA = ~a_code[codeIndex+index];
  }

  index++; // 一列右にずらす
  if( index > 7 ){
    index = 0; // 一番右まで行ったら一番左に戻す
  }
  /* 500m 秒ごとに点灯パターンの変更 */
  if ( cnt > 500 ){
    if ( ++codeIndex > ( MAX_CODE - 1 ) ){
      codeIndex = 0; // F の一番最後まで表示したら A の頭に戻る
    }
    cnt = 0;
  }
  else{
    cnt++;
  }

  CCP1IF = 0; // CCP 割込みフラグをクリア
}
}
```