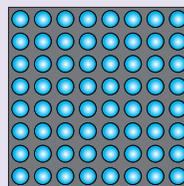


ダイナミック点灯

このSTEPでは、スタティック点灯とダイナミック点灯という点灯方法でドットマトリクスLEDを点灯させます。

課題 6-1

今までは、ドットマトリクスLEDを1つしか点灯させませんでした。ですが、せっかくLEDが64個もあるので、このSTEPでは全てのLEDを点灯させてみましょう。



待ち時間関数を使わないのでプログラム2-1の出力設定を変えるだけでもいいのですが、せっかく初期化関数を作ったので、初期化してから全点灯させましょう。

プログラム例 6-1

```
05 #include <3052f.h> // 3052F 固有の定数
06
07 /*
08  * 初期化関数
09  */
10 void initIO(void)
11 {
12     // 入出力設定
13     P4.DDR = 0xFF; // 出力 LED 横行
14     PB.DDR = 0xFF; // 出力 LED 縦行
15
16     // 出力レベル設定
17     P4.DR.BYTE = 0x00; // L アノード
18     PB.DR.BYTE = 0xFF; // H カソード
19 }
20
21 /*
22  * main 関数
23  */
24 int main(void)
25 {
26     initIO(); // 初期化関数の呼び出し
27
28     // LED 全点灯
29     P4.DR.BYTE = 0xFF; // 1111 1111
30     PB.DR.BYTE = 0x00; // 0000 0000
31
32     // 永久ループ
33     while (1)
34         ;
35
36     return 0;
37 }
```

プログラム 6-1 は、全ての LED に電流を流し続けて LED を点灯させています。このように、電流を流し続けて LED を点灯させる方法を **スタティック点灯** と言います。

しかし、電流を流し続けるということは、それだけ消費する電力も大きくなってしまいます。もっと消費する電力を抑えて LED を点灯させる方法はないのでしょうか？

6.1 ダイナミック点灯

STEP04 でドットマトリクス LED を点滅させた時を思い出してみてください。

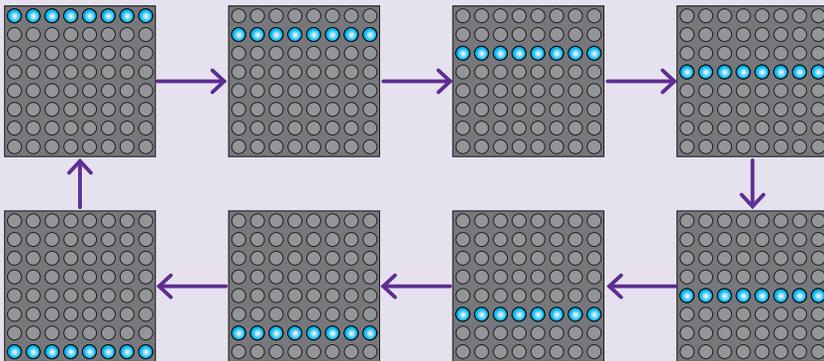
LED の点灯と消灯の周期が短いと、点灯しているように見えました。これを利用してドットマトリクス LED を全点灯すれば、スタティック点灯よりも消費電力を抑えることができます。このように、人間の目の錯覚を利用して LED が点灯しているように見せる方法を **ダイナミック点灯** といいます。

では、具体的にどのようにしてドットマトリクス LED を全点灯させるのでしょうか？

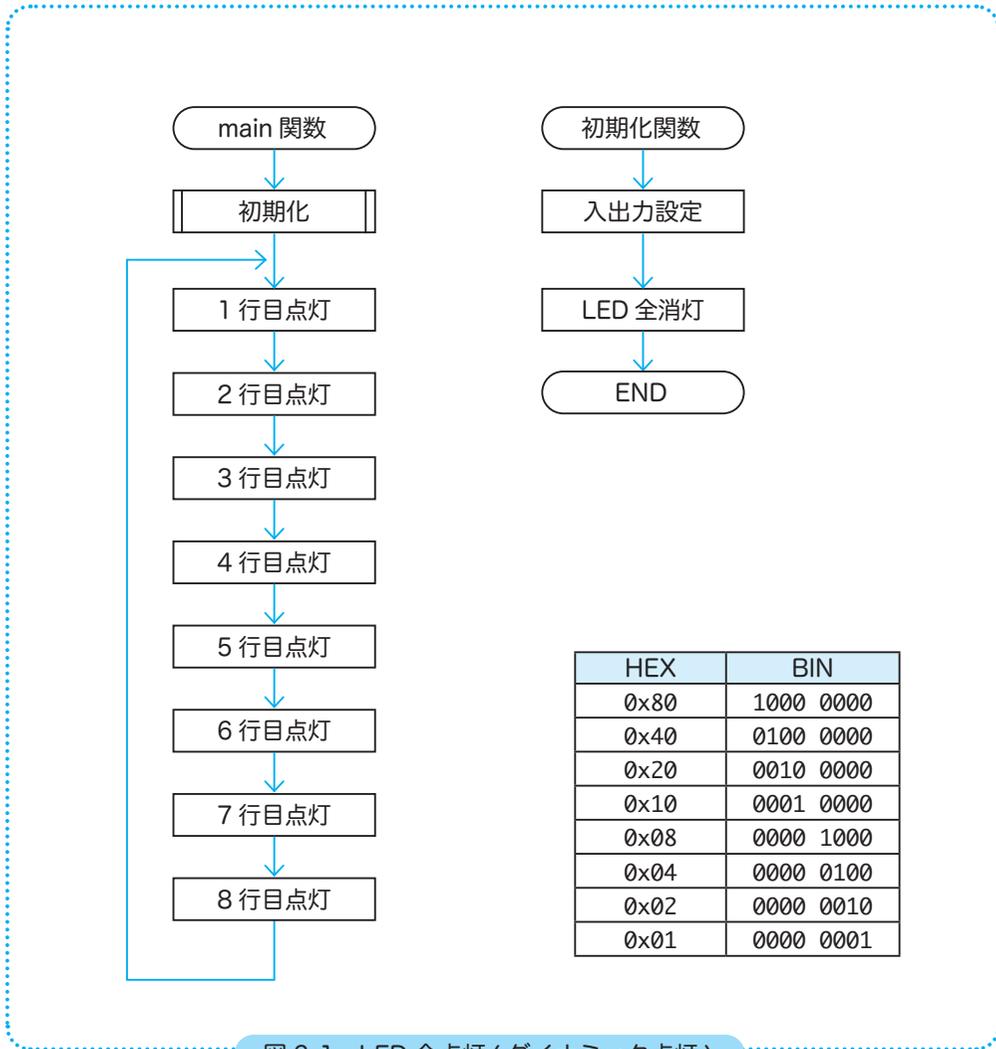
課題 6-2 のようにドットマトリクス LED を点灯する行を高速で変化させてみましょう。

課題 6-2

図のようにダイナミック点灯させ、ドットマトリクス LED を全点灯しているように見せる。



それでは、課題 6-2 を実現するフローチャートを考えてみましょう。



分岐もなく簡単なフローチャートですが、なんだかとても長くなってしまいました。これを基にプログラム 6-2 を組んでみましょう。

プログラム例 6-2

```
21  /*
22  * main 関数
23  */
24  int main(void)
25  {
26      initI0(); // 初期化関数の呼び出し
27
28      PB.DR.BYTE = 0x00; // L カソード
29
30      // 永久ループ
31      while (1)
32      {
33          P4.DR.BYTE = 0x80; // 1 行目のみ点灯
34          P4.DR.BYTE = 0x40; // 2 行目のみ点灯
35          P4.DR.BYTE = 0x20; // 3 行目のみ点灯
36          P4.DR.BYTE = 0x10; // 4 行目のみ点灯
37          P4.DR.BYTE = 0x08; // 5 行目のみ点灯
38          P4.DR.BYTE = 0x04; // 6 行目のみ点灯
39          P4.DR.BYTE = 0x02; // 7 行目のみ点灯
40          P4.DR.BYTE = 0x01; // 8 行目のみ点灯
41      }
42
43      return 0;
44  }
```

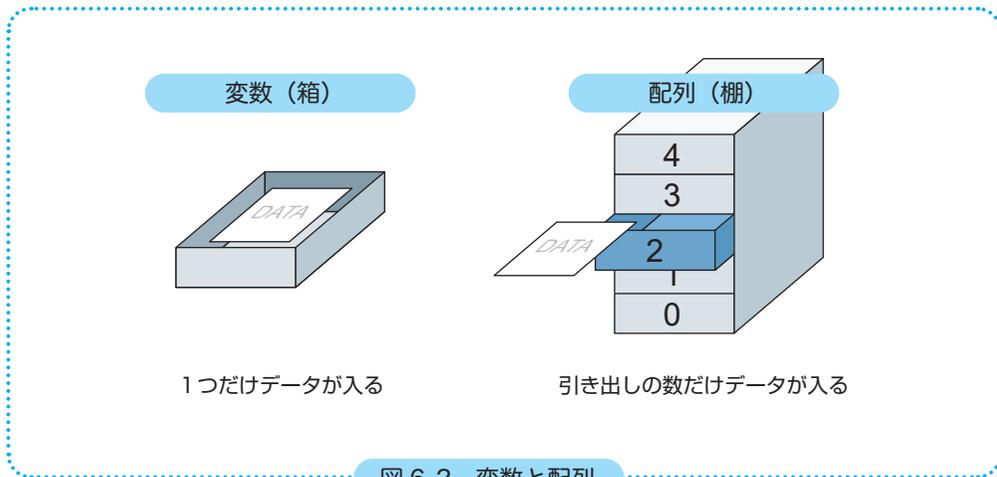
ドットマトリクス LED は点灯しましたか？

でも、各行の点灯プログラムは、ほぼ同じ形なので簡略化できそうです。こんな時は^{はいれっ}配列を使いましょう。

6.2 配列

STEP04 で、変数は箱のようなものだと話しました。

配列は、変数(箱)を並べて一塊にしたようなもので、この変数1つ1つを要素と言います。また各要素には番号がふられており、この番号を添え字と言います。下図の棚に振られた0~4が添え字です。



C 言語で配列を使うには、配列の宣言、代入を行います。

【配列の宣言・代入】

```
宣言   データ型 配列名 [要素の数];  
代入   配列名 [添え字] = 値;
```

例) データ型が「int 型」、配列名が「array」、要素数が「4」の配列を宣言する。

```
int array[4];
```

それらの要素に値を代入する時は、

```
array[0] = 0x01;  
array[1] = 0xff;  
array[2] = 0x3b;  
array[3] = 0x80;
```

ですが、このように記述するとプログラムが長くなってしまいます。
そこで、以下のような記述しましょう。

配列は変数同様、配列の宣言と代入を同時に行うことができます。

【配列の宣言と代入】

```
データ型 配列名 [要素の数] = { 値 , 値 , 値 , 値 , … } ;
```

例) データ型が「int 型」、配列名が「array」、要素数が「4」、値が 0x01、0xff、0x3b、0x80 の配列を宣言、代入を同時に行う。

```
int array[4] = { 0x01, 0xff, 0x3b, 0x80 } ;
```

※ 配列を宣言と代入を同時に行う時は、コンパイラの仕様によりプログラムの先頭で行います。

6.3 配列を使ったプログラム

それでは、配列を使ったドットマトリクス LED 全点灯のプログラム 6-3 を見てください。

プログラム例 6-3

```
05 #include <3052f.h> // 3052F 固有の定数
06
07 // 横点灯行の配列
08 int a_p4[8] = {0x80, 0x40, 0x20, 0x10, 0x08, 0x04, 0x02, 0x01};
09
10 /*
11  * 初期化関数
12  */
13 void initIO(void)
14 {
15     // 入出力設定
16     P4.DDR = 0xFF; // 出力 LED 横行
17     PB.DDR = 0xFF; // 出力 LED 縦行
18
19     // 出力レベル設定
20     P4.DR.BYTE = 0x00; // L アノード
21     PB.DR.BYTE = 0xFF; // H カソード
22 }
23
24 /*
25  * main 関数
26  */
27 int main(void)
28 {
29     int index; // 配列の添え字に使用する変数の宣言
30
31     initIO(); // 初期化関数の呼び出し
32
33     PB.DR.BYTE = 0x00; // L カソード)
34
35     // 永久ループ
36     while (1)
37     {
38         // 1 ~ 8 行目まで点灯行変更
39         for (index = 0; index < 8; index++)
40         {
41             P4.DR.BYTE = a_p4[index];
42         }
43     }
44
45     return 0;
46 }
```

プログラム 6-3 について解説します。

```
// 点灯行の配列
```

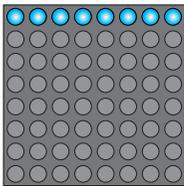
```
int a_p4[8] = {0x80, 0x40, 0x20, 0x10, 0x08, 0x04, 0x02, 0x01};
```

この記述は、ドットマトリクス LED のアノード側を上から 1 列ずつ H レベルにするように値を配列化したものです。配列名は「a_p4」、要素数は「8」、値は「0x80、0x40、0x20、0x10、0x08、0x04、0x02、0x01」です。

先程も書きましたが、[コンパイラの仕様によりプログラムの先頭で配列の宣言と代入を行っています](#)。

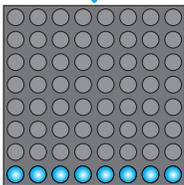
この配列の値は、プログラム 6-3 の 41 行目で P4DR に格納しています。

また、変数「index」を for 文でインクリメントすることにより添え字の値を変更し、ドットマトリクス LED の点灯行を変化させています。

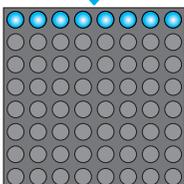


index が 0 (P4DR の bit7 が H) からループが始まります。

その後、index は 8 未満になるまでインクリメントされ、点灯行が変化します。



index が 8 になると、for 文のループが終了します。



そして再び、永久ループによって for 文ループが最初から始まります。