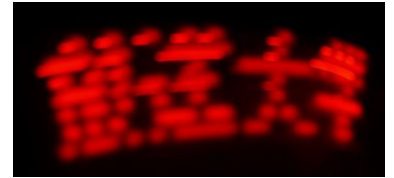


・ LED POV をやってみよう。

LED POV(Persistent Of Vision=残像)は、右の写真のように LED を点滅させ、その残像で文字を書くしくみです。写真では手でブレッドボードを振っています。前回の digitalWrite を8ビット一度に行う digitalWriteByte や、振り初めを検知する水銀スイッチを読む digitalRead などを使います。



・「放送大学」の文字パターン

(8ドット美咲フォント:

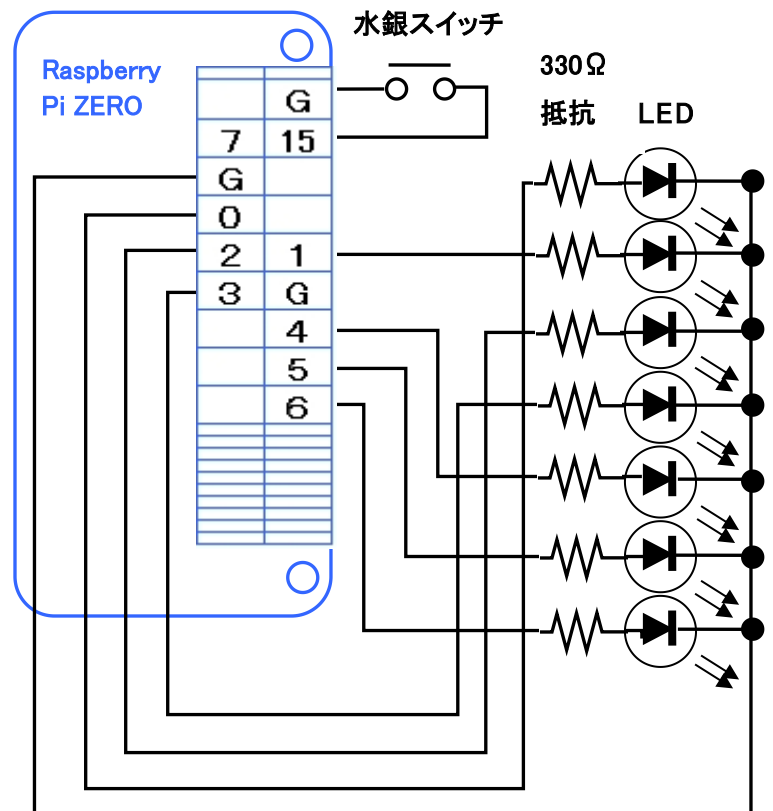
漢字を擬似的に8ドットで表すフォント 使用)

・回路図

文字パターン → 2進数であらわす → 16進にする

7	6	5	4	3	2	1	0	8	4	2	1	8	4	2	1	0x	4	2
0								0	1	0	0	0	0	1	0	0x	3	E
1								0	0	1	1	1	1	1	0	0x	4	B
2								0	1	0	0	1	0	1	1	0x	7	A
3								0	1	1	1	1	0	1	0	0x	5	C
4								0	1	0	1	1	1	0	0	0x	2	3
5								0	0	1	0	0	0	1	1	0x	5	E
6								0	1	0	1	1	1	1	0	0x	0	0
7								0	1	0	0	1	0	0	0	0x	4	9
8								0	0	0	0	0	0	0	0	0x	3	8
9								0	1	0	0	1	0	0	0	0x	6	2
10								0	1	1	0	0	0	1	1	0x	6	B
11								0	1	0	1	1	1	1	0	0x	5	E
12								0	1	1	0	1	0	1	1	0x	6	B
13								0	1	1	0	0	0	1	0	0x	6	2
14								0	0	0	0	0	0	0	0	0x	0	0
15								0	1	0	0	0	1	0	0	0x	4	4
16								0	1	0	0	0	1	0	0	0x	4	4
17								0	0	1	0	0	1	0	0	0x	2	4
18								0	0	0	1	1	1	1	1	0x	1	F
19								0	0	1	0	0	1	0	0	0x	2	4
20								0	1	0	0	0	1	0	0	0x	4	4
21								0	1	0	0	0	1	0	0	0x	4	4
22								0	1	0	0	0	1	0	0	0x	4	4
23								0	0	0	0	0	0	0	0	0x	0	0
24								0	0	0	1	0	1	1	0	0x	1	6
25								0	0	0	1	0	0	1	1	0x	1	3
26								0	1	0	1	0	1	1	0	0x	5	6
27								0	1	1	1	0	1	1	1	0x	7	7
28								0	0	0	1	1	1	1	0	0x	1	E
29								0	0	0	1	0	0	1	1	0x	1	3
30								0	0	0	1	0	1	1	0	0x	1	6
31								0	0	0	0	0	0	0	0	0x	0	0

このデータをプログラムに書く



・ Wiring Pi の関数(その3)

digitalWriteByte(8ビット出力データ) ピン番号 0~7 に 8ビットの出力データを一度に出力。

digitalRead(ピン番号) ピン番号のデータを読んで戻り値を返す。

★ プルアップ設定でスイッチが接続されている時、スイッチがONなら”0” OFFなら”1” が返る。

pullUpDnControl(ピン番号,プルアップ設定) Raspberry Pi に内蔵されているプルアップ抵抗を設定する。

★ 外付けで抵抗をつけるのと同じことがソフトウェアの指定で可能。スイッチをつけるときによく使う。

プルアップ設定 PUD_UP :+ 3.3V 50K Ω でプルアップ PUD_DWN :0V 50K Ω でプルダウン PUD_OFF 無し
setPadDrive(グループ番号,設定値) ピンのグループに流すことができる最大電流を設定。

グループ番号:グループ番号0 がピン番号 0~27 (それ以外のピンは普通ない)なので、設定は通常 0

設定値の計算: 最大電流(mA)=(設定値+1)×2 設定値の最大値は7 (最大電流16mA) 省略値は3

・ POV プログラム pov.c

```
/*
  空中に字を書く
  POV(Persistent Of Vision=残像)プログラム
  pov.c
*/

#include <wiringPi.h>          /* WiringPi の ライブラリを読み込む */

int main()
{
    /* 「放送大学」の文字パターンデータ */
    char patern [32] = { 0x42,0x3e,0x4b,0x7a,0x5c,0x23,0x5e,0x00,
                          0x49,0x38,0x62,0x6b,0x5e,0x6b,0x62,0x00,
                          0x44,0x44,0x24,0x1f,0x24,0x44,0x44,0x00,
                          0x16,0x13,0x56,0x77,0x1e,0x13,0x16,0x00 };

    int i , j;
    char sw;          /* 水銀スイッチ on off を表す変数 */

    wiringPiSetup();          /* WiringPi の 初期化 */
    setPadDrive(0,7);        /* ピングループ 0(0~27)の最大電流を 16mA に */
    pinMode (0,OUTPUT);      /* ***** */
    pinMode (1,OUTPUT);      /* */
    pinMode (2,OUTPUT);      /* */
    pinMode (3,OUTPUT);      /* ピン番号0~7を出力モードにする */
    pinMode (4,OUTPUT);      /* */
    pinMode (5,OUTPUT);      /* */
    pinMode (6,OUTPUT);      /* ***** */
    pinMode (15,INPUT);      /* 15 番ピン(水銀 sw 用)を入力モードにする */
    pullUpDnControl(15,PUD_UP); /* 15 番ピンに内部抵抗を設定 */

    for (j=0;j<1000;j++)     /* 無限ループの代わりに 1000 回 回す */
    {
        sw = digitalRead(15); /* 水銀スイッチの状態を読む */
        if (sw ==0)          /* 水銀スイッチが on なら以下を実行 */
        {
            delay (100);      /* 開始前にまず 100ms 待つ */
            for (i=0;i<32;i++) /* 文字パターンを 32 列表示するループ*/
            {
                digitalWriteByte(patern[i]); /* 文字パターンを 1 列表示 */
                delay (4);          /* 列を 4ms の間表示 */
                digitalWriteByte(0x00); /* LED を一旦全部消す */
            }
            delay(100);        /* 100ms 待って水銀スイッチの状態を見に行く */
        }
        digitalWriteByte(0x00); /* 最後に LED をすべて消灯する */
    }
    return 0;
}
```

コンパイル gcc -o pov pov.c -lwiringPi 実行 sudo ./pov (sudo:システム権限で実行)