

LED 7-segment interface

ถ้าต่อ LED 7-segment 1 หลักร ให้แสดงผลเป็นตัวเลขฐานสิบหกตั้งแต่ 0 จนถึง F

หมายเหตุ ตอน Download ต้องเอาสายที่ต่อขา D0 D1 ออกก่อน

โปรแกรมที่ 1

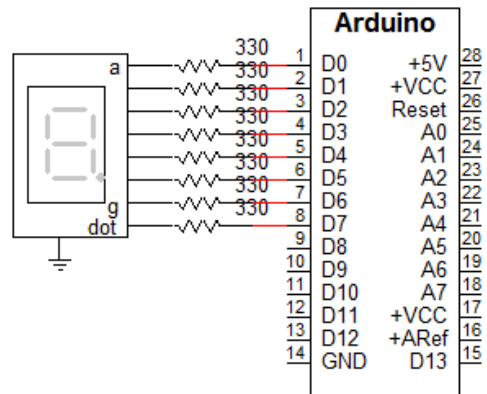
```

unsigned char col;
const unsigned char pattern[] =
{0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,0x7f,0x6f,0x77,0x
7c,0x39,0x5e,0x79,0x71};

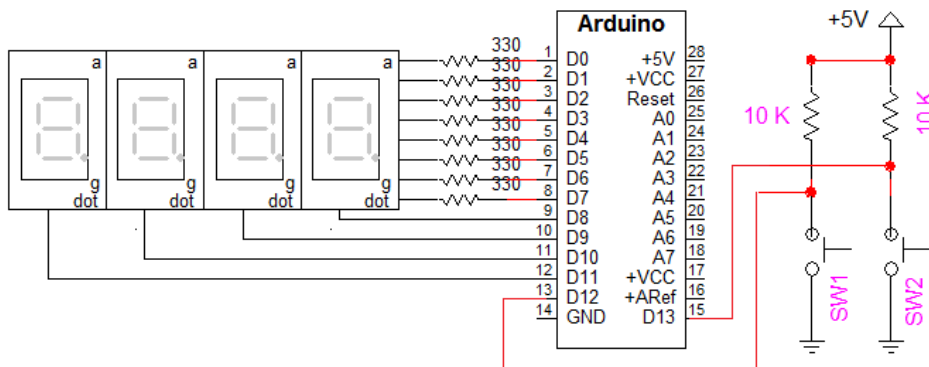
void setup() {
  DDRD = B11111111; //Port D as output
}
void loop() {
  for (int i=0;i<16;i++)

  {
    PORTD = pattern[i];
    delay(400);
  }
}

```



ถ้าต่อ LED 7-segment 4 หลักร ค่าที่ต้องการให้แสดงเก็บอยู่ในตัวแปรแบบอะเรย์ขนาด 4 ไบต์



โปรแกรมที่ 2

```

unsigned char col;
const unsigned char pattern[] = {0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,0x7f,0x6f};
const unsigned char digit[] = {0b11111110,0b11111101,0b11111011,0b11110111};
unsigned char dis_buf[4] = {5,7,2,8}; //Display number
void setup() {
  DDRD = B11111111; //Port D as output
  DDRB = B11111111; //Port B as output
}
void loop() {
  for (int i=0;i<4;i++)
  {
    PORTB = 0b11111111; //Blank display
    PORTD = pattern[dis_buf[i]]; //Out segment
    PORTB = digit[i]; //Out Digit
    delay(5); //Delay
  }
}

```

ตัวอย่างแสดงการใช้งาน Constant array และตัวแปรตัวอย่างแสดง

เทคนิคการเขียนอีกแบบ เปลี่ยนการสแกน display ในฟังก์ชัน loop มาเป็นการทำในฟังก์ชันการอินเทอร์รัพท์

โปรแกรมที่ 3

```
#include <TimerOne.h>
unsigned char col;
const unsigned char pattern[] = {0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,0x7f,0x6f};
unsigned char dis_buf[4] = {5,7,2,8};
unsigned char digit;
void setup() {
  DDRD = B11111111; //Port D as output
  DDRB = B11111111; //Port B as output
  Timer1.initialize(5000); // set a timer of length 1000 microseconds
  Timer1.attachInterrupt( timerIsr ); // attach the service routine here
  digit = 0;
}
void loop() {
}
// -----
// Scan display
// -----
void timerIsr()
{
  PORTB = 0b11111111; //Blank display
  switch (digit) {
  case 0:
    PORTD = pattern[dis_buf[digit]]; //Out segment
    PORTB = 0B11111110;
    break;
  case 1:
    PORTD = pattern[dis_buf[digit]]; //Out segment
    PORTB = 0B11111101;

    break;
  case 2:
    PORTD = pattern[dis_buf[digit]]; //Out segment
    PORTB = 0B11111011;

    break;
  case 3:
    PORTD = pattern[dis_buf[digit]]; //Out segment
    PORTB = 0B11110111;
    break;
  }
  ++digit;
  if (digit > 3) digit = 0;
}
```

โปรแกรมที่ 3 และ 4 อยู่ที่ <https://sites.google.com/site/eplearn/arduino-project/24-led-7seg>

โปรแกรมที่ 4 แสดงตัวอย่างแสดงผลเลขจำนวนเต็ม ตั้งแต่ 0 – 9999

โดยการคำนวณหาค่าเลขแต่ละหลัก หลักหน่วย หลักสิบ หลักร้อย และหลักพัน แล้วนำไปใส่ในอะเรย์ของแต่ละหลัก

โปรแกรมที่ 4

```
#include <TimerOne.h>
unsigned char col;
const unsigned char pattern[] =
{0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,0x7f,0x6f,0x77,0x7c,0x39,0x5e,0x79,0x71};
unsigned char dis_buf[4] = {0,0,0,0};
unsigned char digit;
unsigned int count = 0;
unsigned int c0,c1,c2,c3;
unsigned char tmp;
void setup() {
  DDRD = B11111111; //Port D as output
  DDRB = B11111111; //Port C as output
  Timer1.initialize(5000); // set a timer of length 1000 microseconds
  Timer1.attachInterrupt( timerIsr ); // attach the service routine here
  digit = 0;
}
void loop() {
  c0 = count/1000;
  c1 = (count%1000)/100;
  c2 = ((count%1000)-c1*100)/10;
  c3 = count%10;
  dis_buf[0] = c0;
  dis_buf[1] = c1;
  dis_buf[2] = c2;
  dis_buf[3] = c3;
  delay(10);
  count++;
  if(count>9999) count = 0;
}

/// Scan display
void timerIsr()
{
  PORTB = 0b11111111; //Blank display
  switch (digit) {
    case 0:
      PORTD = pattern[dis_buf[digit]]; //Out segment
      PORTB = 0B11111110;
      break;
    case 1:
      PORTD = pattern[dis_buf[digit]]; //Out segment
      PORTB = 0B11111101;
      break;
    case 2:
      PORTD = pattern[dis_buf[digit]]; //Out segment
      PORTB = 0B11111011;
      break;
    case 3:
      PORTD = pattern[dis_buf[digit]]; //Out segment
      PORTB = 0B11110111;
      break;
  }
  ++digit;
  if (digit > 3) digit = 0;
}
```

งานมอบหมาย ให้เขียนโปรแกรม เมื่อกดสวิทช์ SW1 ให้เพิ่มค่าหลักหน่วยและหลักสิบ ถ้ากด SW2 ให้เพิ่มค่าหลักร้อยและหลักพัน