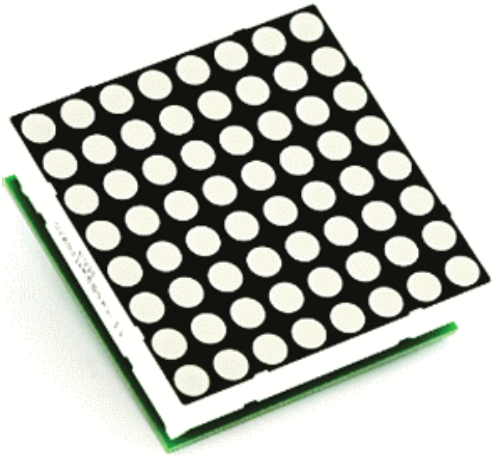
01076001
วิศวกรรมคอมพิวเตอร์เบื้องต้น
Introduction to Computer Engineering

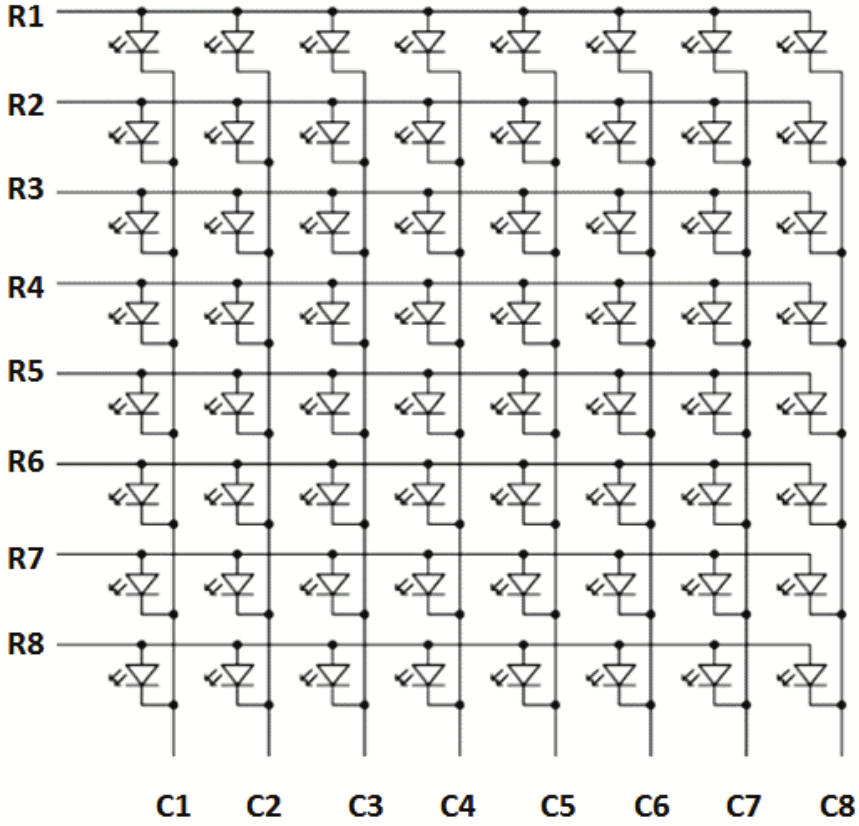
Arduino #4

LED Dot Matrix

LED Dot Matrix

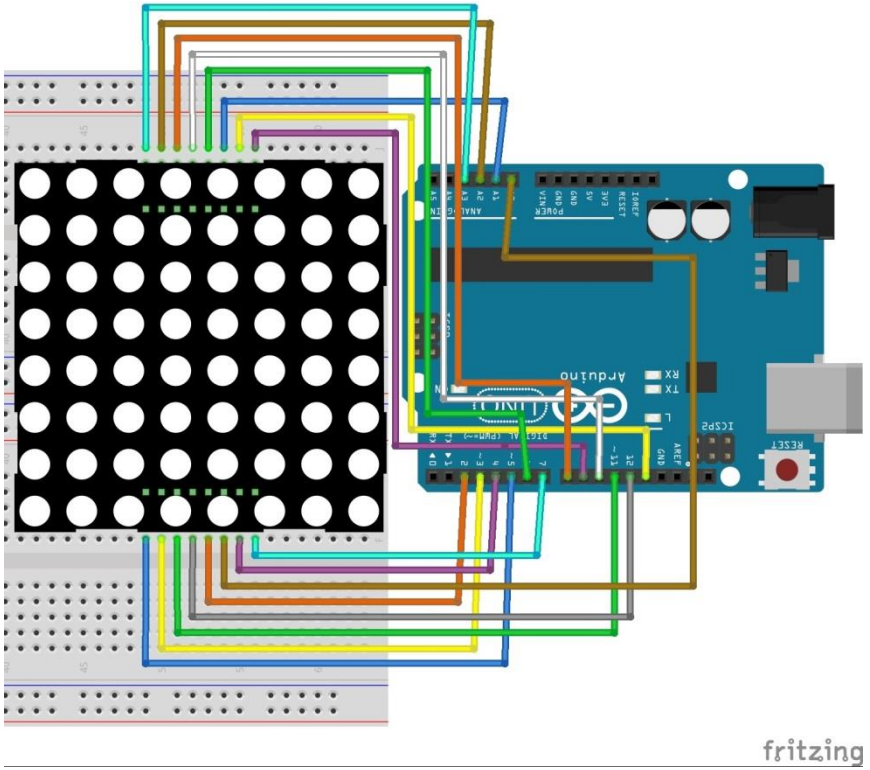
- เป็นอุปกรณ์ที่นำเอา LED จำนวนมากมาไว้ในชิ้นเดียวกัน มีวงจรตามรูป
- ถ้าป้อน R1=High, C1=Low ไฟดวงไหนจะติด





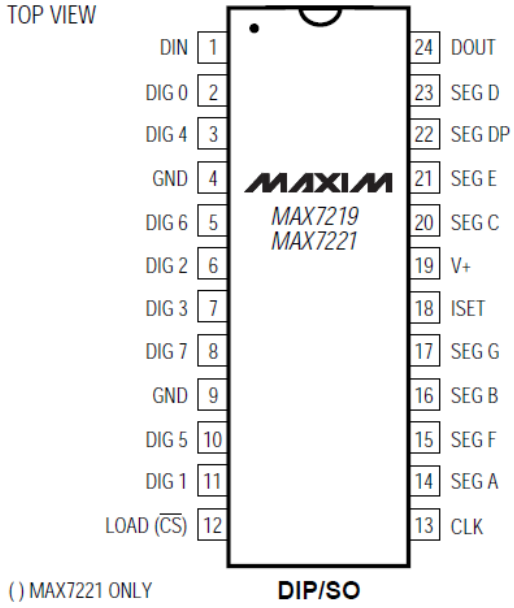
การเชื่อมต่อ Arduino กับ LED Dot Matrix

- เราสามารถเชื่อมต่อ LED Dot Matrix กับ Arduino ได้ ตามวงจรตัวอย่างในรูป
- จะเห็นว่ามีการใช้สายไฟจำนวนมาก
- ทำให้ไม่สะดวก
- ในรูปไม่ได้ต่อ R เพื่อให้ดูง่าย



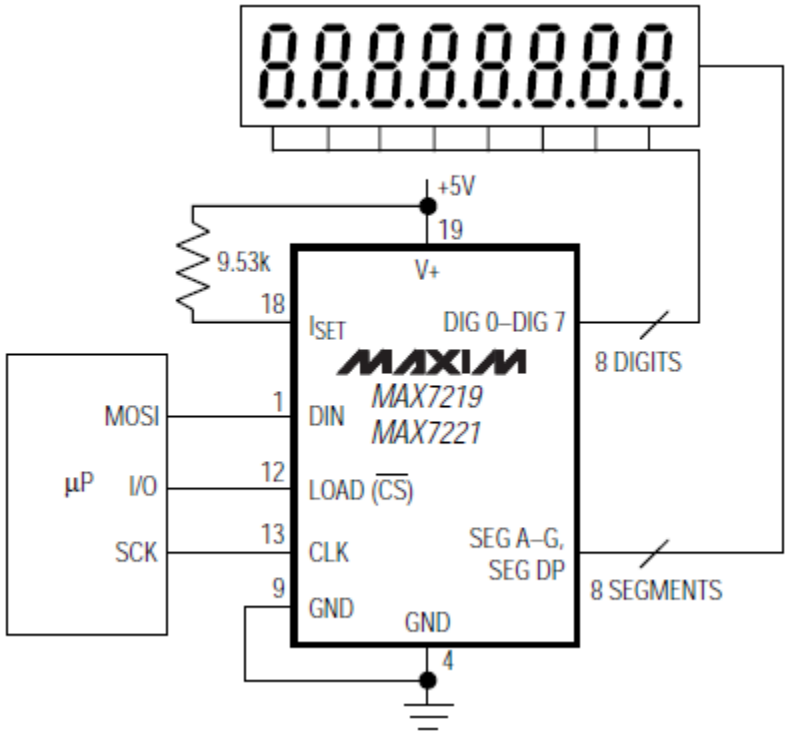
การเชื่อมต่อ Arduino กับ LED Dot Matrix

- ปัญหาข้างต้นเกิดขึ้นกับการต่อใช้งาน LED 7 Segment ที่มีหลายหลักเช่นกัน
- จึงได้มีผู้สร้างชิป IC สำหรับใช้ต่อกับ 7 Segment หลายหลัก โดยมีชื่อว่า MAX7219
- MAX7219 สามารถควบคุม LED 7 Segment ได้ 8 หลัก หรือใช้กับ LED Dot Matrix ขนาด 8x8 ได้



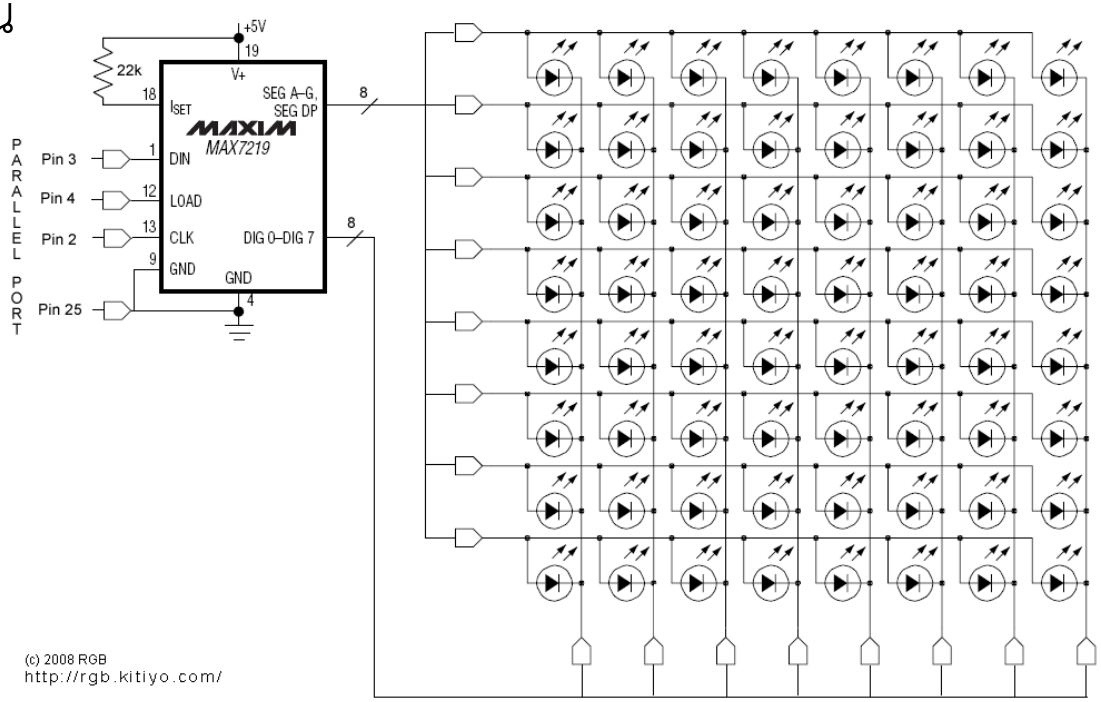
การเชื่อมต่อ Arduino กับ LED Dot Matrix

- ในการต่อ MAX7219 เข้ากับ LED 7 Segment จะต่อขา SEG A-G,DP เข้ากับขา a-g ของแต่ละหลัก และต่อ DIG 0-7 เข้ากับแต่ละหลัก (ตามรูป)
- ในการทำงาน MAX7219 จะส่งข้อมูลของแต่ละหลักไปทีละครั้ง เช่น ส่งข้อมูลของหลักที่ 1 ในขณะที่ส่ง DIG 0 ออกไป ซึ่งจะทำให้หลักที่ 1 ติด จากนั้นทิ้งไว้ระยะหนึ่งแล้วจึงส่งหลักที่ 2-8 ในทำนองเดียวกัน
- โดยใช้ความเร็วที่เหมาะสม ตาของมนุษย์จะเห็นทุกหลักติดหมด วิธีนี้เรียกว่าการ Scan



การเชื่อมต่อ Arduino กับ LED Dot Matrix

- สำหรับการเชื่อมต่อกับ Arduino จะใช้ต่อผ่านขาจำนวน 3 ขา คือ
  - Din เป็นขาสำหรับข้อมูล
  - CLK เป็นขาคล็อกสำหรับ Sync ข้อมูล โดยคล็อก 1 สัญญาณ จะหมายถึงข้อมูล 1 บิต
  - LOAD/CS จะใช้บอกกับ MAX7219 ว่า จะโหลดข้อมูลแล้ว



การเชื่อมต่อ Arduino กับ LED Dot Matrix

- การส่งข้อมูลจะส่งเป็นชุด ชุดละ 16 บิต ขั้นตอนจะเริ่มจาก 1) CS เป็น LOW 2) ส่งสัญญาณ CLK 3) ส่งข้อมูลทุกครั้งที่ขอบขาขึ้น จาก D15-> D0 (เส้นแดง)

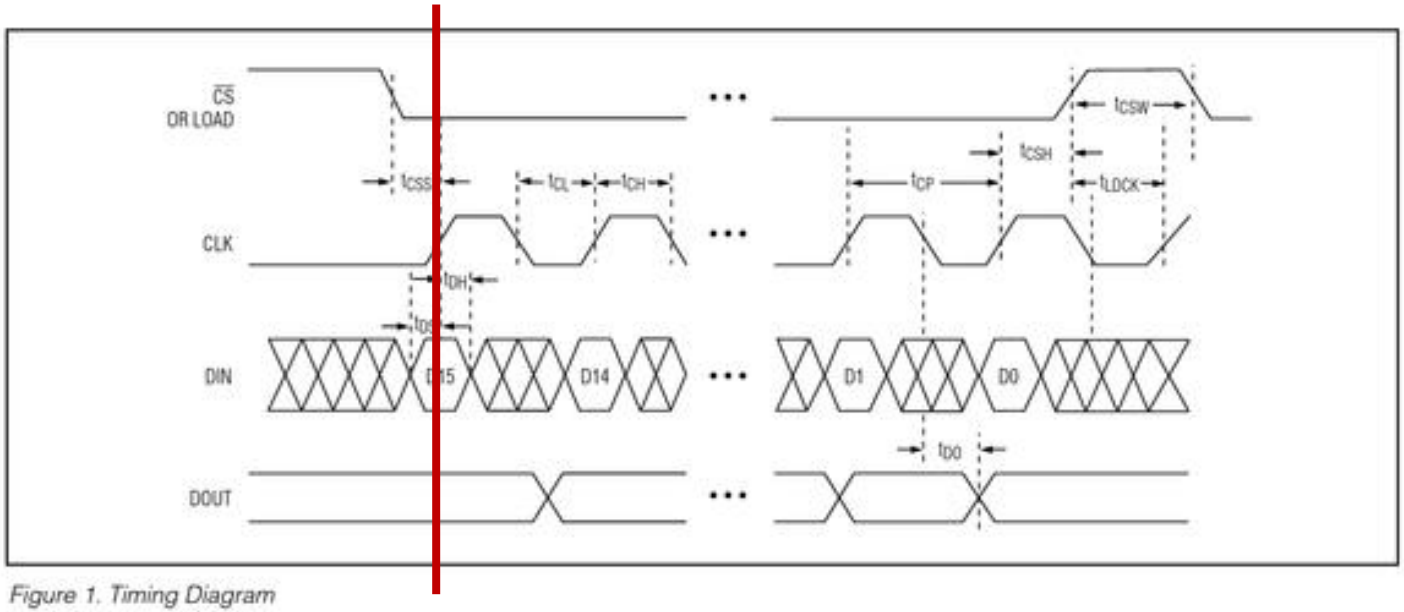


Figure 1. Timing Diagram

Table 1. Serial-Data Format (16 Bits)

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
X	X	X	X	ADDRESS				MSB	DATA						LSB

การเชื่อมต่อ Arduino กับ LED Dot Matrix

- ในชิป MAX7219 จะมีที่เก็บข้อมูลที่เรียกว่า รีจิสเตอร์ (Register) ขนาด 8 บิตอยู่ 14 ตัว (อาจมองได้ว่าเป็นชุดข้อมูลที่ใช้โดย MAX7219)
- รีจิสเตอร์แต่ละตัวจะเก็บข้อมูลต่างกัน บางตัวเก็บค่าข้อมูลที่จะแสดง บางตัวเก็บความสว่าง เป็นต้น ชิปจะใช้ข้อมูลจากรีจิสเตอร์เหล่านี้ไปทำงาน หรืออาจจะบอกว่าสามารถสั่งงานชิปผ่านรีจิสเตอร์ก็ได้
- รูปแบบข้อมูลที่ส่งจะเป็นไปตามรูปด้านล่าง โดย D15-D12 จะเป็นอะไรก็ได้ (ไม่สนใจ แต่โดยทั่วไปจะกำหนดเป็น 0)
- ในการระบุว่าการส่งข้อมูลแต่ละครั้ง จะส่งเข้าไปที่รีจิสเตอร์ใด จะระบุใน Address (D11-D8) และตามด้วยค่าข้อมูล 8 บิต

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
X	X	X	X	ADDRESS				MSB	DATA						LSB

การเชื่อมต่อ Arduino กับ LED Dot Matrix

- Reg. 1-8 เก็บข้อมูลที่จะแสดงแต่ละหลัก
- Reg. 9 บอกว่าแต่ละหลักจะใช้ decode แบบ BCD หรือไม่ (ค่าปกติคือ ไม่)
- Reg. A ใช้กำหนดความสว่างโดยมี 15 ระดับ (00h-0Fh)
- Reg. B ใช้กำหนดว่าแสดงหลักใดบ้าง (00h-07h)
- Reg. C ค่า 1=ทำงานปกติ 0=หยุดทำงาน

REGISTER	ADDRESS					HEX CODE
	D15–D12	D11	D10	D9	D8	
No-Op	X	0	0	0	0	X0
Digit 0	X	0	0	0	1	X1
Digit 1	X	0	0	1	0	X2
Digit 2	X	0	0	1	1	X3
Digit 3	X	0	1	0	0	X4
Digit 4	X	0	1	0	1	X5
Digit 5	X	0	1	1	0	X6
Digit 6	X	0	1	1	1	X7
Digit 7	X	1	0	0	0	X8
Decode Mode	X	1	0	0	1	X9
Intensity	X	1	0	1	0	XA
Scan Limit	X	1	0	1	1	XB
Shutdown	X	1	1	0	0	XC
Display Test	X	1	1	1	1	XF

การเชื่อมต่อ Arduino กับ LED Dot Matrix

```
#include <SPI.h>

const int CS_PIN   = 10;  // SPI /SS
const int CLK_PIN  = 13;  // SPI SCK
const int DIN_PIN  = 11;  // SPI MOSI

void MAX7219_write_reg( uint8_t addr, uint8_t data ) {
    digitalWrite( CS_PIN, LOW );
    SPI.transfer( addr );
    SPI.transfer( data );
    digitalWrite( CS_PIN, HIGH );
}

#define REG_DIGIT(x)       (0x1+(x))
#define REG_DECODE_MODE    (0x9)
#define REG_INTENSITY      (0xA)
#define REG_SCAN_LIMIT     (0xB)
#define REG_SHUTDOWN       (0xC)
#define REG_DISPLAY_TEST   (0xF)

void MAX7219_init(void) {
  MAX7219_write_reg( REG_DECODE_MODE, 0x00 );  // decode mode: no decode for digits 0-7
  MAX7219_write_reg( REG_INTENSITY, 0x07 );    // set intensity: 0x07=15/32
  MAX7219_write_reg( REG_SCAN_LIMIT, 0x07 );   // scan limit: display digits 0-7
  MAX7219_write_reg( REG_SHUTDOWN, 0x01 );     // shutdown: normal operation
  MAX7219_write_reg( REG_DISPLAY_TEST, 0x00 ); // display test: no display test
}
```

การเชื่อมต่อ Arduino กับ LED Dot Matrix

```
void setup() {
  SPI.begin();
  SPI.setBitOrder( MSBFIRST );
  SPI.setClockDivider( SPI_CLOCK_DIV16 );// 16MHz/16 -> 1MHz SCK frequency
  SPI.setDataMode( SPI_MODE0 ); // use SPI mode 0
  pinMode( CS_PIN, OUTPUT );
  digitalWrite( CS_PIN, HIGH );
  MAX7219_init();
}

void flashing() {
  MAX7219_write_reg( REG_SHUTDOWN, 0x01 );      // normal operation
  MAX7219_write_reg( REG_DISPLAY_TEST, 0x01 ); // enter display test mode
  delay(100);
  MAX7219_write_reg( REG_DISPLAY_TEST, 0x00 ); // exit display test mode
  MAX7219_write_reg( REG_SHUTDOWN, 0x00 );      // shutdown operation
  delay(900);
}
```

การเชื่อมต่อ Arduino กับ LED Dot Matrix

```
const byte char_data[][8] = { // 'C', 'E'
  { B00000000,
    B10000001,
    B10000001,
    B10000001,
    B10000001,
    B10000001,
    B11111111,
    B00000000 },
  { B00000000,
    B10000001,
    B10000001,
    B10010001,
    B10010001,
    B10010001,
    B11111111,
    B00000000 }};


void show_ce() {
  static uint8_t ch=0;
  for (uint8_t i=0; i < 8; i++) {
     MAX7219_write_reg( REG_DIGIT(i), char_data[ch][i] );
  }
  delay(500);
  ch = (ch+1) % 2;
}
```

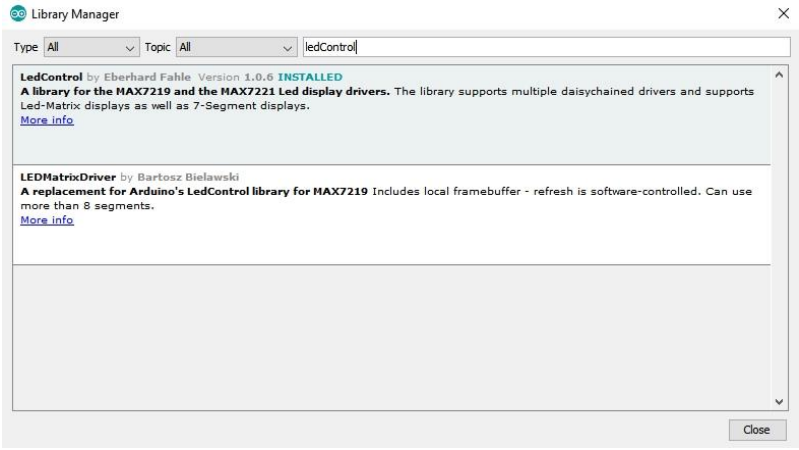
การเชื่อมต่อ Arduino กับ LED Dot Matrix

```
void loop() {
  for (uint8_t i=0; i < 3; i++) {
    flashing();
  }
  MAX7219_write_reg( REG_SHUTDOWN, 0x01 );     // normal operation
  while (1) {
    show_ce();
  }
}
```

Activity

- ให้นำ LED Dot Matrix ขนาด 8x8 มาต่อกับบอร์ด Arduino โดยต่อดังนี้
  - Vcc ต่อกับ 5V
  - GND ต่อกับ Ground
  - CS_PIN ต่อกับขา 10
  - CLK_PIN ต่อกับขา 13
  - DIN_PIN ต่อกับขา 11
- นำโปรแกรมข้างต้นโหลดและตรวจสอบการทำงาน
- ทดลองเปลี่ยนเป็นข้อความอื่น
- หมายเหตุ สามารถต่อผ่าน Joy Stick Shield ได้

การเชื่อมต่อ Arduino กับ LED Dot Matrix



- จากวิธีการที่กล่าวมาข้างต้น แม้จะแสดงผลบนจอ LED Dot Matrix ได้ แต่การเอาไปใช้ก็ยังยากอยู่
- จึงได้มีผู้สร้าง Library ขึ้นมาใช้งาน ซึ่งก็มีจำนวนมาก Library ตัวหนึ่งที่นิยมใช้กันมีชื่อว่า LedControl
- การติดตั้ง
  - ไปที่ Sketch -> Include Library -> Manage Libraries
  - จะปรากฏหน้าต่าง Library Manager ให้ป้อน LedControl แล้วเลือกติดตั้ง

การเชื่อมต่อ Arduino กับ LED Dot Matrix

```
#include "LedControl.h"

LedControl lc=LedControl(11,13,10,4);  // CLK,DIN,CS,Number of LED Module

unsigned long delaytime=500;  // time to updates of the display

void setup() {
  int devices=lc.getDeviceCount();  // find no of LED modules
  //we have to init all devices in a loop
  for(int address=0;address<devices;address++) {  // set up each device
    lc.shutdown(address,false);
    lc.setIntensity(address,8);
    lc.clearDisplay(address);
  }
}

void loop() {
  int devices=lc.getDeviceCount();  // find no of LED modules

  for(int row=0;row<8;row++) {
    for(int col=0;col<8;col++) {
      for(int address=0;address<devices;address++) {
        delay(delaytime);
        lc.setLed(address,row,col,true);
        delay(delaytime);
        lc.setLed(address,row,col,false);
      }
    }
  }
}
```

การเชื่อมต่อ Arduino กับ LED Dot Matrix

- ฟังก์ชัน **`setLed(addr,row,col,T/F)`** จะรับข้อมูลประกอบด้วย addr คือ โมดูล row,col ค่าแถวและคอลัมน์ และ T/F คือ ให้ติดหรือดับ
- ฟังก์ชัน **`setRow(addr,row,value)`** จะรับข้อมูลประกอบด้วย addr คือ โมดูล row คือ แถวที่จะแสดง และ value คือค่า 8 บิตที่จะให้แสดง
- ฟังก์ชัน **`setColumn(int addr, int col, byte value)`** ใช้งานเช่นเดียวกับ setRow

การเชื่อมต่อ Arduino กับ LED Dot Matrix

- สำหรับการแสดงผลตัวอักษร มีผู้ที่ทำข้อมูล 8x8 เอาไว้

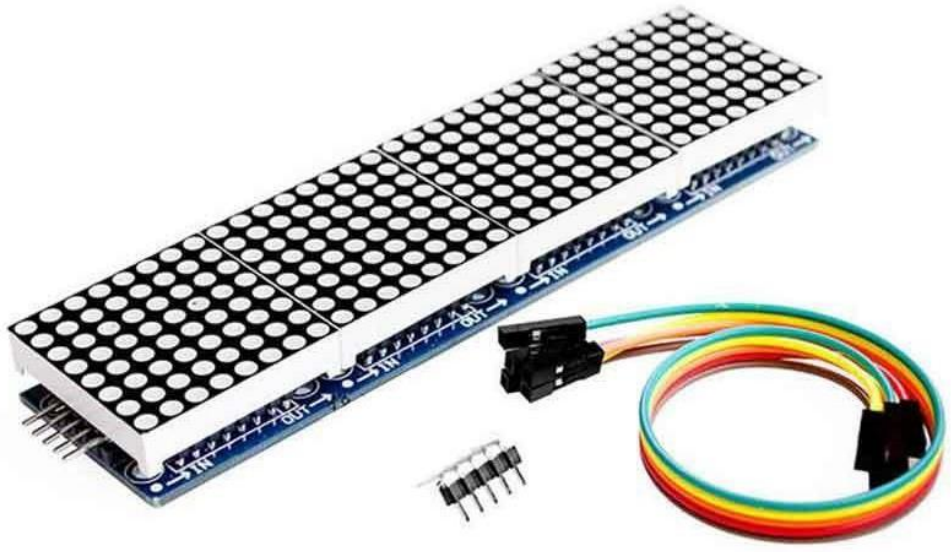
http://www.gammon.com.au/forum/?id=11516

Activity

- ให้นำโปรแกรมใน Slide 16 มาทดลองรัน
- ให้แก้ไขโปรแกรมให้จุดวิ่งเป็นรูป 4 เหลี่ยม

การเชื่อมต่อ Arduino กับ LED Dot Matrix

- กรณีที่ต้องการใช้ LED Dot Matrix มากกว่า 1 โมดูล
- ยังคงสามารถใช้ไลบรารี LedControl ในการควบคุมได้ เนื่องจากฟังก์ชันสามารถระบุโมดูลที่ต้องการแสดงผลได้ เช่น **`setLed(addr,row,col,T/F)`** จะใช้ตัวแปร addr ในการระบุโมดูล แต่เนื่องจากใช้งานไม่สะดวก จึงได้ทำฟังก์ชันขึ้นมาครอบ



การเชื่อมต่อ Arduino กับ LED Dot Matrix

- ฟังก์ชันแรกที่สร้าง คือ plot โดยจะทำหน้าที่แปลงตำแหน่ง x,y ที่เป็น 32x8 ลงไปยังโมดูล เพื่อให้ง่ายต่อการเขียนโปรแกรมมากขึ้น
- ฟังก์ชัน clear_display ใช้สำหรับเคลียร์หน้าจอ ทั้ง 4 หน้าจอ
- ฟังก์ชัน fede_down ทำหน้าที่ค่อยๆ หรี่จอลงจนดับ ทั้ง 4 หน้าจอเช่นกัน
- **หมายเหตุ** โปรแกรมที่เขียนใหม่ได้ upload ลงใน FB แล้ว
- จากนั้นจะใช้ฟอนต์จาก https://github.com/mrnick1234567/miniclock/blob/master/libraries/FontLEDClock/FontLEDClock.h
- และสร้างฟังก์ชันสำหรับแสดงตัวอักษรโดยมี 2 รูปแบบ คือ อักษรตัวใหญ่ (5x7) และตัวเล็ก (3x5)

Assignment #4 : game

- ให้สร้างโปรแกรมเกม space invader บน LED 8x8 หรือ เกมอื่น
- เริ่มต้นโดยการสร้างฐานปืนที่ด้านล่าง ควบคุมการเลื่อนซ้ายขวาโดย Joy Stick
- เมื่อกดปุ่มยิง ให้ยิงกระสุนออกจากฐานปืน โดยให้กระสุนวิ่งไปเรื่อยๆ
- สร้างยานอวกาศ โดยให้สุ่มแสดงผล และสุ่มการปล่อยระเบิด
- ถ้ายิงโดนระเบิดให้ระเบิดหายไป กำหนดจำนวนระเบิด ถ้ายิงได้ครบ ชนะ
- ถ้ามีระเบิดมาถึงด้านล่าง ก็ตาย (อาจกำหนดชีวิตก็ได้)
- ให้เขียนเอง ถ้าพบว่าลอกมาจากที่อื่น จะได้ 0
- การส่ง 1) รายงาน อธิบายแนวคิด พร้อมไฟล์ source code และคำอธิบาย 2)วิดีโอ อธิบายและสาธิตการเล่นเกม
- คะแนน 5 คะแนน 1) โครงสร้าง รูปแบบรายงาน 2) Clean and Organize Code

For your attention