

建構簡單 Arduino 自動控制系統(二) (輸出部份 MX7219 八位數碼管)

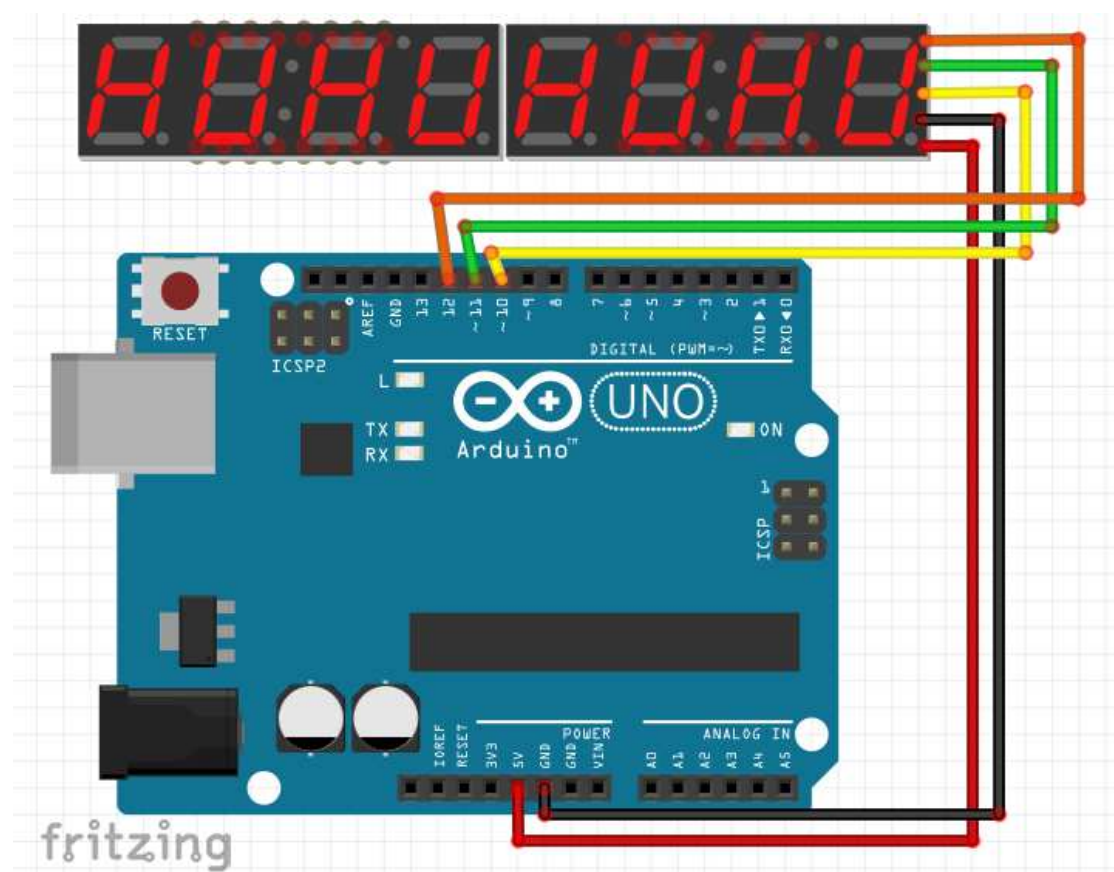
Author:spuggy0919@gmail.com

學習等級：已會操作 Arduino 開發介面寫程式。

Arduino 有一個 LedControl 的模組，這次 MX7219 使用這個模組來顯示。
在此教學我們主要在說明如何在八位管以 printf 輸出。

零件表

品名	數量	說明
Arduino UNO 3	1	也可以用其它版的 Arduino
MX7219 八位管	1	

第一步 弄清楚 MX7219 模組，並進行硬體連接腳位定義

MX7219 模組用時序控制的方式來顯示每一個 7 段字，這些顯示的字是由 Arduino 透過 SPI 序列方式一個位元(Bit)一個位元用 CLOCK 當車廂運送進去的，當 MX7219 收了一個 8 位元的訊息就是一個字節。這個原理同樣也要看 MX7219D 規格書才能懂的。在這我們買的模組已經將 8 個字節和 MX7219 IC 組成一個 5Pins 的模組，扣

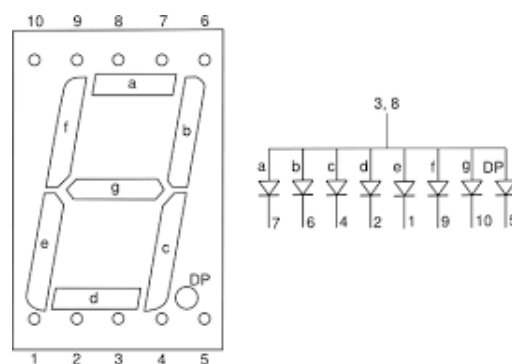
除 5V 及 GND，有三隻控制腳連接方式如下所述
電源正負千萬別弄錯，程式碼也要對應定義腳位。

	Arduino	MX7219 八位數碼管
VCC	+5V	VCC
GND	GND	GND
MAX7219DataIn	10	DIN
MAX7219LOAD	11	LOAD
MAX7219CLK	12	CLK

第二步 修改 LedControl 程式庫

開啓文字編輯器(TEXT editor)可用 Word Pad，打開檔案 LedControl.h Library 中，若沒有此程式庫可至

<http://playground.arduino.cc/Main/LedControl> 網站下載，在 LedControl.h 內有一個表，這個表是將 ASCII 碼對應到要顯示的 7 段符號，例如 '1' 的 ASCII 碼為 0x31，所以在第七列第二個 B00110000，(第一個為 0x30)，此兩個 1 分別點亮 b、c，第八列為 '8' ASCII 為 0x38，7 段是 B01111111 表是除了 dp 以外的 abcdefg 全點亮。



依此我們可以將 0~9a~z 這 36 個數字及英文字母以示意的方式將之全部點畫如下：

```

B01111110,B00110000,B01101101,B01111001,B00110011,B01011011,B01011111,B01110000, // 01234567
B01111111,B01111011,B00000000,B00000000,B00000000,B00000000,B00000000,B00000000, // 89
B00000000,B01110111,B00011111,B00001101,B00111101,B01001111,B01000111,B01011110, // AbcdEFG
B00010111,B00010000,B00111100,B01010111,B00001110,B01010100,B00010101,B00011101, // hiJKLMno
B01100111,B01110011,B00000101,B01011011,B00001111,B00011100,B00111110,B00101010, // PqrStuVW
B00110111,B00111011,B01101101,B00000000,B00000000,B00000000,B00000000,B00001000, // XyzStuVW

```

這樣回存，接下來便可以將數字英文字母顯示於七段字節上。

第三步 創建 LedControl 物件，透過此物件才能將字母轉換成顯示燈號
在程式中輸入

```

// 定義 MAX7219 顯示模組 I/O 怎麼接
#define MAX7219DataIn 10 //DIN
#define MAX7219LOAD 11 //CS
#define MAX7219CLK 12 //CLK
// 建立"LedControl"物件
// 以下 numDevices 是指要串接幾組 MAX7219 顯示模組,目前只用 1 組做測試
LedControl
lc=LedControl(MAX7219DataIn,MAX7219CLK,MAX7219LOAD,1);

```

```
//(DataIn, CLK1, LOAD, numDevices)
```

第四步 在 init()中, 初始化 LedControl 物件, 設亮度清除等

```
// MAX7219 啟動時預設為省電模式(shutdown mode),必須初始化
// 設定正常運作模式
lc.shutdown(0,false); //模組 0, true = shutdown mode ; false
= normal operation
// 設定亮度
lc.setIntensity(0,8); //模組 0
// 清除 DSP
lc.clearDisplay(0); //模組 0
```

第五步 在 loop()中, 顯示字節字元, 經此步驟便能完整達到輸出功能了, 但只能一個字元一個字元顯示。

```
void loop() {
lc.setChar(0,1,'8',false); //(模組 0, 第 1 位置, 顯示'8', 不亮點
}
```

第六步 移植 Printf 函數,便可以用 printf 將字串輸出, 將原來的 **printchare** 中的 PutChar 修改用 lc.setChar 丟字元。這樣便可以使用 Printf 當然字串長度依然只有 8 位數。

```
//----- 以下副程式在 7 段顯示器 printf -----

/*=====
// 在 7 段顯示器 printf
// 用到函式 setChar(int addr=模組#, int digit=DSP#, char c, boolean dp);
=====*/

static int pos = 7;
static int upos = 7;
static int uc = 0;
static boolean ub = 7;
static void printchare(char **str, int c, boolean b)
{
    if (str) {
        **str = c;
        ++(*str);
    }
    else {
        if (c=='\n') {
            pos = 7;
            lc.clearDisplay(0); //模組#0 DSP off
        }else if (c=='\r') {
            pos = 7;
        } else if (toupper(c) == '.' && !ub){
            lc.setChar(0,upos, uc,true); upos = pos;
        }
    }
}
```

```

        }else {
            lc.setChar(0,pos, toupper(c),b);
            upos = pos; uc = toupper(c); ub= b;
            pos= (pos-1+8) % 8;
        }
    }
}

#define PAD_RIGHT 1
#define PAD_ZERO 2

static int prints(char **out, const char *string, int width, int pad)
{
    register int pc = 0, padchar = ' ';
    if (width > 0) {
        register int len = 0;
        register const char *ptr;
        for (ptr = string; *ptr; ++ptr) ++len;
        if (len >= width) width = 0;
        else width -= len;
        if (pad & PAD_ZERO) padchar = '0';
    }
    if (!(pad & PAD_RIGHT)) {
        for ( ; width > 0; --width) {
            printchare (out, padchar,false);
            ++pc;
        }
    }
    for ( ; *string ; ++string) {
        printchare (out, *string,false);
        ++pc;
    }
    for ( ; width > 0; --width) {
        printchare (out, padchar,false);
        ++pc;
    }

    return pc;
}

/* the following should be enough for 32 bit int */
#define PRINT_BUF_LEN 12
static int printi(char **out, int i, int b, int sg, int width, int pad, int
letbase)
{
    char print_buf[PRINT_BUF_LEN];
    register char *s;
    register int t, neg = 0, pc = 0;
    register unsigned int u = i;

    if (i == 0) {
        print_buf[0] = '0';
        print_buf[1] = '\\0';
        return prints (out, print_buf, width, pad);
    }

    if (sg && b == 10 && i < 0) {
        neg = 1;

```

```

        u = -i;
    }

    s = print_buf + PRINT_BUF_LEN-1;
    *s = '\0';

    while (u) {
        t = u % b;
        if( t >= 10 )
            t += letbase - '0' - 10;
        *--s = t + '0';
        u /= b;
    }

    if (neg) {
        if( width && (pad & PAD_ZERO) ) {
            printchare (out, '-', false);
            ++pc;
            --width;
        }
        else {
            *--s = '-';
        }
    }

    return pc + prints (out, s, width, pad);
}

static int print(char **out, const char *format, va_list args )
{
    register int width, pad;
    register int pc = 0;
    char scr[2];

    for (; *format != 0; ++format) {
        if (*format == '%') {
            ++format;
            width = pad = 0;
            if (*format == '\\0') break;
            if (*format == '%') goto out;
            if (*format == '-') {
                ++format;
                pad = PAD_RIGHT;
            }
            while (*format == '0') {
                ++format;
                pad |= PAD_ZERO;
            }
            for ( ; *format >= '0' && *format <= '9'; ++format) {
                width *= 10;
                width += *format - '0';
            }
            if( *format == 's' ) {
                register char *s = (char *)va_arg( args, int );

                pc += prints (out, s?s:"(null)", width, pad);
                continue;
            }

```

```

        if( *format == 'd' ) {
            pc += printi (out, va_arg( args, int ), 10, 1, width, pad,
'a');
            continue;
        }
        if( *format == 'x' ) {
            pc += printi (out, va_arg( args, int ), 16, 0, width, pad,
'a');
            continue;
        }
        if( *format == 'X' ) {
            pc += printi (out, va_arg( args, int ), 16, 0, width, pad,
'A');
            continue;
        }
        if( *format == 'u' ) {
            pc += printi (out, va_arg( args, int ), 10, 0, width, pad,
'a');
            continue;
        }
        if( *format == 'c' ) {
            /* char are converted to int then pushed on the stack */
            scr[0] = (char)va_arg( args, int );
            scr[1] = '\0';
            pc += prints (out, scr, width, pad);
            continue;
        }
    }
    else {
        out:
        printchare (out, *format,false);
        ++pc;
    }
}
if (out) **out = '\0';
va_end( args );
return pc;
}
int setcursor(int position)
{
    pos = rangecheck(position,0,7);
    return pos;
}
int printf(const char *format, ...)
{
    va_list args;
    va_start( args, format );
    return print( 0, format, args );
}

```

第七步 可以在 loop()中使用 printf 試試

```

void loop(){
    int state=0x2A;
    printf("\n%x",state);
    delay(1000);
    printf("\n012345678.");
}

```