Arduino字幕機自造與程式設計

主講人：洪國勝老師

主辦單位：義民高中資訊科

講義下載點：www.goodbooks.com.tw/download/ardu111.docx

日期：4/12 早上9：00～12：00

研習講義大綱

一、單晶微控板使用步驟

二、自我測試

三、資料數位化：（國中資訊科技課綱）

1.以燈號顯示12的記憶體表示。驗證是否00001100

2. 以燈號顯示-3的記憶體表示。驗證是否11111101

四、耶誕霹靂燈、紅綠燈等製作。（使用8個LED）

五、顯示文字、動畫、告白板、走馬燈。（使用8\*8點陣LED。）

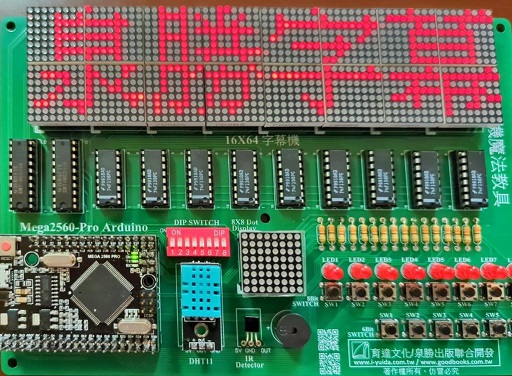
六、16\*64字幕機操作

引用教材

1. 高中生活科技實作（泉勝，洪國勝）
2. Arduino程式設計（泉勝，洪國勝）
3. Arduino字幕機自造與程式設計（泉勝，洪國勝）
4. 單晶片微處理機實習（泉勝，洪國勝）

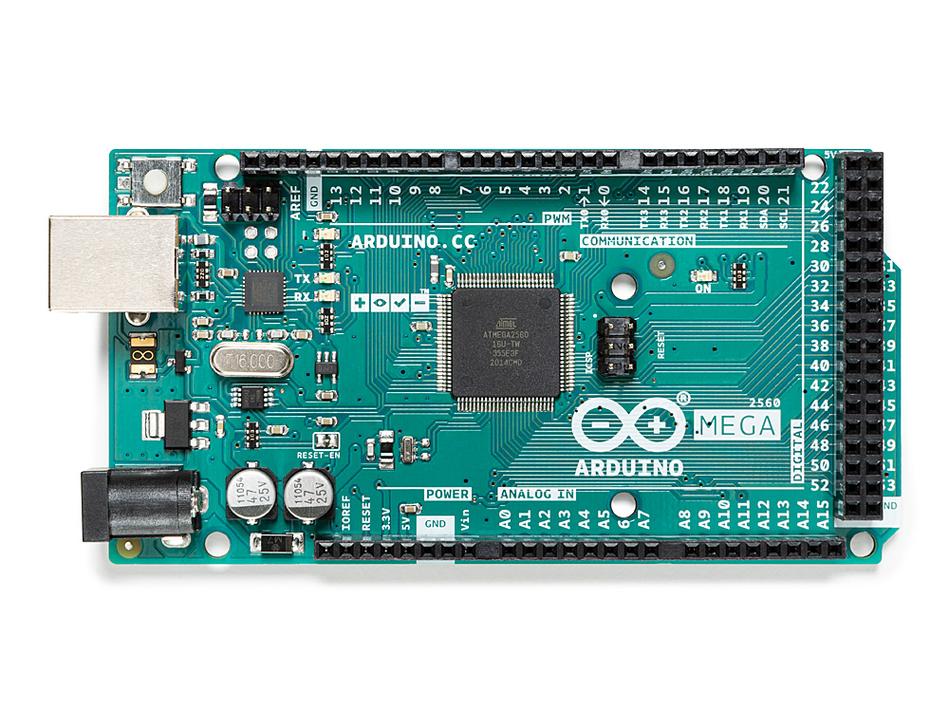
使用教具

育達泉勝「Arduino字幕機魔法教具」



講義內文

1. 單晶微控板使用步驟



Arduino的單晶微控板使用步驟分別是插入微控板、點選開發板型號、點選通訊埠編號，分別說明如下：

1、插入微控板。

請依照指示，將微控板USB插頭插入電腦USB插座。請留意微控板右上角電源指示燈是否亮起。

2、點選開發板型號。

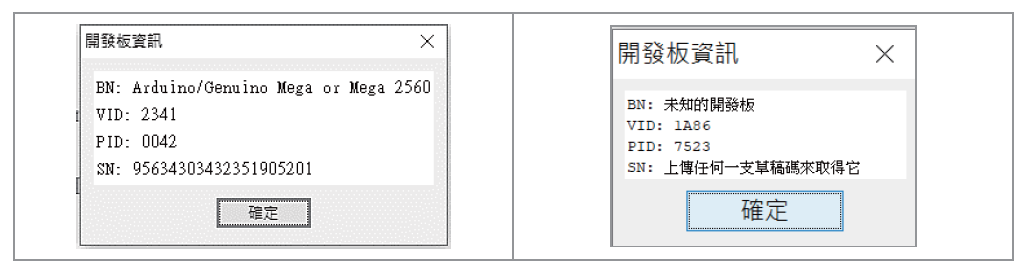
請點選功能表的『工具/開發板』即可點選您所使用的微控板型號。（請依照您的微控板型號點選，筆者是點選『Arduino/Genuino Mega or Mega 2560』，若您是UNO版，也是在此點選，因為不同版本腳位數量不一樣，選錯了就無法正確編譯程式）

3、點選通訊埠編號。

請點選功能表的『工具/序列埠』即可點選您所使用的序列埠編號（備註：系統會出現可用編號com1或com2,3,4,5等等，讓您點選，有時候會同時出現很多個編號，請按照順序點選，直到可上傳(或稱燒錄)為止。其次，有些電腦不會自動抓到通訊埠(com1、com2…)，請到網路搜尋與下載『CH34x-install-Windows』，並安裝，直到出現通訊埠。）

4、取得開發版資訊

請點選功能表的『工具/取得開發版資訊』即可取得您的微控板資訊。若出現下圖，才表示以上設定就緒，才能上傳程式。（實驗的中途，若改插入別人的微控板，那也要重覆以上步驟，直到看到下圖，才表示有抓到此微控板，才能上傳程式。其次，下圖左是原廠的微控板，若不是原廠，那可能就像下圖右，顯示『未知的開發板』。）



二、自我測試

Arduino微控板有內植一顆LED，腳位編號是13，別名是LED\_BUILTIN。其次，整合開發環境安裝後，內部也含一個自我測試程式，這樣就可以測試此微控板是否已經安裝完成。進行單晶片控制都要這樣一步一步來，這樣當問題發生時，才能一步一步除錯，逐漸縮小錯誤範圍，並排除故障。以下說明如何自我測試：

1、開啟自我測試程式。下圖是開啟測試程式Blink（請點選功能表的『檔案/範例/01.Basics/Blink』），其功能是直接使用微控板預植的LED（不管什麼板UNO、MEGA…等），都是腳位13，以常數『LED\_BUILTIN』表示），並令其明滅閃爍各一秒。

1. 資料數位化 （國中資訊科技課綱）

正數沒問題，正數直接轉為2進位，只要數值大於0，直接一直除以2，其餘數的累加，就是2進位。例如：13的轉換方式如下：

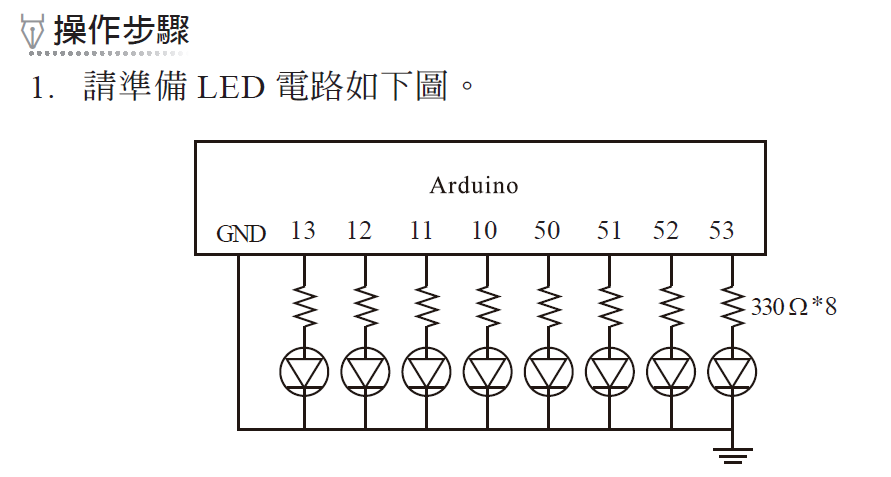
13/2=6..1

6/2=3..0

3/2=1..1

1/2=0..1

所以13的記憶體表示為00001101，我們用LED展示記憶體燈號，將會看到連續4個滅，1個亮，1個滅，再連續2個亮。



2.鍵入以下程式。

void setup() {

DDRB=B11111111;//指派PORTB全為輸出

Serial.begin(9600);//啟動序列埠

}

unsigned char i=13; //宣告i為8位元正負整數,這樣i可儲存0~255的整數

void loop() {

PORTB=i;

Serial.println((int)i);//轉為整數

}

請鍵入以下程式，觀察執行結果

void setup() {

DDRB=B11111111;//指派PORTB全為輸出

***DDRC=0xFF;*** //指派PORTC全為輸出

***PORTC=0xFE;//0xFE is B11111110，只有C8為低電位，所以可以當作8個LED***

Serial.begin(9600);//啟動序列埠

}

unsigned char i=13; //宣告i為8位元正負整數,這樣i可儲存0~255的整數

void loop() {

PORTB=i;

Serial.println((int)i);//轉為整數

}

以上斜體程式，***0xFE is B11111110，將8\*8點陣LED，***設定C1～C7皆為高電位，C8為低電位，那此8\*8點陣LED，就可當作8個LED來使用。本書往後使用任何獨立LED的程式也都這樣，直接使用軟體設定的方式。

自我練習

1. 請將112轉為2進位，並以LED驗證記憶體內容。

2、負數數位化。

負數則是以2補數表示，以-3 為例，先將3 轉為二進位

00000011

取1 補數

11111100

加1，所以是

11111101

2. 看到11111101，最高位元為1，表示此為負數，到底負多少，也是取2 補數，先取1 補數如下：

00000010

再加1，結果是

00000011

所以是-3。

3、請鍵入以下程式，並觀察-3的燈號是否為11111101（連續6個亮，1個滅，1個亮）。資料型態選char ，表示資料可接受-128～127

程式一：

void setup() {

DDRB=B11111111;

Serial.begin(9600);

}

char i=-3; //宣告i為8位元正負整數,這樣i可表示-128~127

void loop() {

PORTB=i;

Serial.println((int)i);//轉為整數

}

程式二：

void setup() {

DDRB=B11111111;

DDRC=0xFF;PORTC=0xFE;//0xFE is B11111110 將8\*8點陣LED當作8個LED使用

Serial.begin(9600);

}

char i=-3; //宣告i為8位元正負整數,這樣i可表示-128~127

void loop() {

PORTB=i;

Serial.println((int)i);//轉為整數

}

自我練習

1請將-8轉為2進位，並以LED驗證記憶體內容。

四、霹靂燈的製作

1、在自動控制的領域裡，我們會習慣以2進位或16進位來表示一些燈號或控制結果。以下是一些常用數字的16進位書寫表示方式。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 10進位 | 2進位 | 16進位 | 10進位 | 2進位 | 16進位 |
| 0 | B0 | 0x0 | 11 | B1011 | 0xb |
| 1 | B1 | 0x1 | 12 | B1100 | 0xc |
| 2 | B10 | 0x2 | 13 | B1101 | 0xd |
| 3 | B11 | 0x3 | 14 | B1110 | 0xe |
| 4 | B100 | 0x4 | 15 | B1111 | 0xf |
| 5 | B101 | 0x5 | 16 | B10000 | 0x10 |
| 6 | B110 | 0x6 | 17 | B10001 | 0x11 |
| 7 | B111 | 0x7 | 18 | B10010 | 0x12 |
| 8 | B1000 | 0x8 | 127 | B01111111 | 0x7F |
| 9 | B1001 | 0x9 | 254 | B11111110 | 0xFE |
| 10 | B1010 | 0xa | 255 | B11111111 | 0xFF |

時序圖

以上僅是一個瞬間結果的資料數位化。但有很多情況，例如，廣告燈、耶誕樹燈、紅綠燈等，還會隨著時間的移動有不同的變化，此時可以以時間為軸，取一段時間為單位，將這些單位的結果記錄下來，這些以時間為橫軸的連續資料數位化，稱為時序圖。程式設計師就按照此時序圖，將結果以陣列存放，再依序輸出此陣列，即可完成指定動作。

實例探討

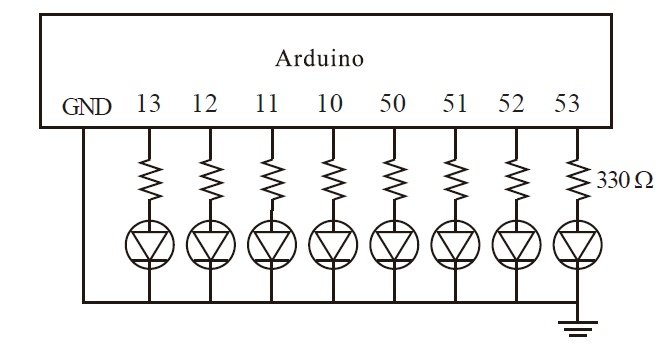
小明買了一顆耶誕樹，現在要將它裝上8顆LED，且讓其閃爍如下，請問如何規劃。

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 時序 | LED7 | LED6 | LED5 | LED4 | LED3 | LED2 | LED1 | LED0 |
| 1 | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ● |
| 2 | ○ | ○ | ○ | ○ | ○ | ○ | ● | ● |
| 3 | ○ | ○ | ○ | ○ | ○ | ● | ● | ● |
| 4 | ○ | ○ | ○ | ○ | ● | ● | ● | ● |
| 5 | ○ | ○ | ○ | ● | ● | ● | ● | ● |
| 6 | ○ | ○ | ● | ● | ● | ● | ● | ● |
| 7 | ○ | ● | ● | ● | ● | ● | ● | ● |
| 8 | ● | ● | ● | ● | ● | ● | ● | ● |

（實心代表燈亮，空心代表不亮）

操作步驟

1. 將電路接線如下：



1. 以秒為單位，規劃時序圖。本例規劃如下（1代表亮，0代表不亮）：

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 時序 | LED7 | LED6 | LED5 | LED4 | LED3 | LED2 | LED1 | LED0 | 值 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0x01 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0x03 |
| 2 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0x07 |
| 3 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0x0F |
| 4 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0x1F |
| 5 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0x3F |
| 6 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0x7F |
| 7 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0xFF |

3、一一將以上資料數位化，得到0x01, 0x03, 0x07, 0x0F, 0x1F, 0x3F, 0x7F, 0xFF，如上表最右一欄。

4、使用變數儲存以上資料。本例使用a0, a1, a2, a3,a4, a5, a6, a7表示如下：

a0=0x01;

a1=0x03;

a2=0x07;

a3=0x0F;

a4=0x1F;

a5=0x3F;

a6=0x7F;

a7=0xFF;

3 將以上資料寫入Arduino，程式如下：

void setup() {

DDRB=B11111111;

//Serial.begin(9600);

DDRC=0xFF;PORTC=0xFE;//0xFE is B11111110 將點陣LED當作8個LED使用

}

void loop() {

byte a0,a1,a2,a3,a4,a5,a6,a7;

//宣告a0,a1...是8位元正整數，可表示0~255

a0=0x01;

PORTB=a0;

delay(1000);

a1=0x03;

PORTB=a1;

delay(1000);

a2=0x07;

PORTB=a2;

delay(1000);

a3=0x0F;

PORTB=a3;

delay(1000);

a4=0x1F;

PORTB=a4;

delay(1000);

a5=0x3F;

PORTB=a5;

delay(1000);

a6=0x7F;

PORTB=a6;

delay(1000);

a7=0xFF;

PORTB=a7;

delay(1000);

}

以上程式有點冗長，對於連續的資料，請導入陣列，才能縮短程式。

霹靂燈與陣列

將以上資料直接宣告與指派初值如下：

byte a[]={0x01,0x03,0x07,0x0F,0x1F,0x3F,0x7F,0xFF};

有了陣列，往後就可以使用迴圈與陣列索引存取陣列的值，程式如下：

void setup() {

DDRB=B11111111;

//DDRC=0xFF;PORTC=0xFE;//0xFE is B11111110 將8\*8點陣LED當作8個LED使用

Serial.begin(9600);

}

void loop() {

byte a[]={0x01,0x03,0x07,0x0F,0x1F,0x3F,0x7F,0xFF};

for (int i=0 ;i<=7;i++){

PORTB=a[i];

delay(1000);

}

delay(3000);

}

也可以使用以下無窮迴圈

void setup() {

DDRF=B11111111;

Serial.begin(9600);

DDRC=0xFF;PORTC=0xFE;//0xFE is B11111110 將8\*8點陣LED當作8個LED使用

}

byte i=0;

void loop() {

byte a[]={0x01,0x03,0x07,0x0F,0x1F,0x3F,0x7F,0xFF};

PORTB=a[i];

delay(1000);

i=(i+1)%8;//i每次遞增1，但%8可保障數字i在0～7之間

}

3. 本例的  
 byte i=0;

要放在void loop() {}外面，此稱為全域變數，請移至loop()裡面，並觀察執行結果。

#### 自我練習

1. 若希望霹靂燈的變化如下，請寫程式完成。

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 時序 | LED7 | LED6 | LED5 | LED4 | LED3 | LED2 | LED1 | LED0 | 值 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |  |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 |  |
| 2 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |  |
| 3 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |  |
| 4 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |  |
| 5 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 |  |
| 6 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |  |
| 7 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  |
| 8 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  |
| 9 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |  |
| 10 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |  |
| 11 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |  |
| 12 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |  |
| 13 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |  |
| 14 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |  |
| 15 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  |
| 16 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 時序 | LED7 | LED6 | LED5 | LED4 | LED3 | LED2 | LED1 | LED0 | 值 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0x11 |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0x42 |
| 2 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0x24 |
| 3 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0x18 |
| 4 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0x18 |
| 5 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0x3c |
| 6 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0x7e |
| 7 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0x7f |
| 8 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0xff |
| 9 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0xfe |
| 10 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0xfc |
| 11 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0xf8 |
| 12 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0xf0 |
| 13 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0xe0 |
| 14 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0xc0 |
| 15 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0x80 |
| 16 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

1. 紅綠燈。若有一個單向紅綠燈時序如下，請寫程式完成。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 時序 | LED2（紅燈） | LED1（黃燈） | LED0(綠燈) | 值 |
| 0 | 0 | 0 | 1 |  |
| 1 | 0 | 0 | 1 |  |
| 2 | 0 | 0 | 1 |  |
| 3 | 0 | 0 | 1 |  |
| 4 | 0 | 1 | 0 |  |
| 5 | 0 | 1 | 0 |  |
| 6 | 1 | 0 | 0 |  |
| 7 | 1 | 0 | 0 |  |
| 8 | 1 | 0 | 0 |  |

1. 請自行觀察路口的雙向紅綠燈，並寫程式完成。

提示：將所有燈號建立時序圖。

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 時序 | LED5紅 | LED4黃 | LED3綠 | LED2紅 | LED1黃 | LED0綠 | 值 |
| 0 | 1 | 0 | 0 | 0 | 0 | 1 |  |
| 1 | 1 | 0 | 0 | 0 | 0 | 1 |  |
| 2 | 1 | 0 | 0 | 0 | 0 | 1 |  |
| 3 | 1 | 0 | 0 | 0 | 0 | 1 |  |
| 4 | 1 | 0 | 0 | 0 | 1 | 0 |  |
| 5 | 1 | 0 | 0 | 1 | 0 | 0 |  |
| 6 | 0 | 0 | 1 | 1 | 0 | 0 |  |
| 7 | 0 | 0 | 1 | 1 | 0 | 0 |  |
| 8 | 0 | 0 | 1 | 1 | 0 | 0 |  |
| 9 | 0 | 0 | 1 | 1 | 0 | 0 |  |
| 10 | 0 | 0 | 1 | 1 | 0 | 0 |  |
| 11 | 0 | 0 | 1 | 1 | 0 | 0 |  |
| 12 | 0 | 0 | 1 | 1 | 0 | 0 |  |
| 13 | 0 | 1 | 0 | 1 | 0 | 0 |  |
| 14 | 1 | 0 | 0 | 1 | 0 | 0 |  |

1. 高鐵停靠站燈號。高鐵的月台上，都有一個告示板，用來顯示即將到站列車的停靠站。例如：以下燈號表示停靠南港、台北、台中、台南、高雄。

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 列車編號 | 南港 | 台北 | 板橋 | 桃園 | 新竹 | 苗栗 | 台中 | 彰化 | 嘉義 | 台南 | 高雄 |
| 0 | ● | ● | 0 | 0 | 0 | 0 | ● | 0 | 0 | ● | ● |

請問您如何設計此系統？

1. 將PORTK的後3位元接到南港(A10)、台北(A9)、板橋(A8)，PORTF(A7~A0)接到桃園、新竹…
2. 依照列車時刻表標示將所有列車的停靠站（1代表有停靠，0代表未停靠）
3. 將以下資料數位化。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 列車編號 |  | 南港 | 台北 | 板橋 | 桃園 | 新竹 | 苗栗 | 台中 | 彰化 | 嘉義 | 台南 | 高雄 | PORTK | PORTF |
|  |  | PORTK | | | PORTF | | | | | | | |  |  |
| 0 |  | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0x6 | 0x13 |
| 1 |  | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0x7 | 0xff |
| 2 |  | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0x6 | 0x57 |
| 3 |  | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0x6 | 0x51 |

1. 捷運到站燈號。搭乘捷運時，車上都有列車停靠站顯示說明。例如：以下代表列車即將進入「紅樹林」站。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 時序 | 淡水 | 紅樹林 | 竹圍 | 關渡 | 忠義 | 復興崗 | 北投 | 奇岩 | 其哩岸 | 石牌 | 明德 | 芝山 | 士林 | 劍潭 | 圓山 | 民權西路 | 雙連 | 中山 | 台北車站 | 台大醫院 | 中正紀念堂 | 東門 | 大安 | 安和 | 101 | 象山 |
| 0 | 0 | ● | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

請問如何設計此系統。

參考解答

1.以淡水線為例，使用PORTA、PORTB、PORTC、PORTF接線如下。

2.將所有資料數位化。

byte a[]={0x80,0x40,0x20,0x10,0x08};//PORTA

byte b[]={0x0,0x0,0x0,0x0,0x0};//PORTB

byte c[]={0x0,0x0,0x0,0x0,0x0};//PORTC

byte f[]={0x0,0x0,0x0,0x0,0x0};//PORTF

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 時序 | 淡水 | 紅樹林 | 竹圍 | 關渡 | 忠義 | 復興崗 | 北投 | 奇岩 | 其哩岸 | 石牌 | 明德 | 芝山 | 士林 | 劍潭 | 圓山 | 民權西路 | 雙連 | 中山 | 台北車站 | 台大醫院 | 中正紀念堂 | 東門 | 大安 | 安和 | 101 | 象山 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | PORTA | | | | | | | | PORTB | | | | | | | | PORTC | | | | | | | | PORTF | |

6、教學機。本單元的LED您可以拿來作為順序教學的指示燈。例如，霓虹燈、鉤毛線順序、電子琴、跳舞機、CPR急救順序、葉問拳法練習樁等，也就是您可以用LED來提醒使用者要按哪裡、踩哪裡、打哪裡等等等。請自己思考身邊的應用，寫程式完成。

電子琴教學機

<https://youtu.be/CsvtgIQ7Vx0>

五、字幕機原理與製作方式

顯示文字、動畫、告白板、走馬燈。（使用8\*8點陣LED。）

字幕機是由很多LED分列與分行組成，但並不是每個LED獨立使用一條線控制，而是因為人類眼睛有視覺暫留現象，所以可分列共用資料線、分行共用位址線，進而顯示任何文字，請看以下範例說明。

1、請將點陣8\*8LED 接線如下:





硬體測試

請鍵入以下程式，觀察LED有沒有一排一排輪流亮，第1排亮01、第2排亮2（10）、第3排亮3（11）…到第8排亮8（1000）。

byte c[]={0x7f,0xbf,0xdf,0xef,0xf7,0xfb,0xfd,0xfe};

void setup() {

DDRB=0xFF;DDRC=0xFF;

}

void loop() {

for (int i=0 ;i<=7;i++){

PORTC=c[i];//位址

PORTB=i+1;//資料

delay(1000);//delay(500)

}

}

請將delay(500); ，修改為delay(1);，並觀察所有LED有沒有全亮。以上程式的c[]陣列如下：

byte c[]={0x7f,0xbf,0xdf,0xef,0xf7,0xfb,0xfd,0xfe};

時序圖如下表：



以上c[]陣列稱為位址，主要是讓C1..C8 輪流接地，資料就可以輪流傳送到C1..C8，例如，以下程式，可讓左邊C1 第一行8 個燈全亮，

PORTC=0x7f; //位址（01111111）

PORTB=0xff; //資料

以下程式，將會使點陣LED 左邊C1 最上面1 個LED 亮。

PORTC=0x7f; //位址（01111111）

PORTB=0x01; //資料

以下程式，將會使點陣LED 左邊C2 的上面兩個LED 全亮。

PORTC =0xbf;//10111111

PORTB =0x3;

只要速度夠快，因為人類眼睛有視覺暫留現象，看起來就會一起亮，例如，以下程式可以讓所有LED全亮，

byte c[]={0x7f,0xbf,0xdf,0xef,0xf7,0xfb,0xfd,0xfe};

void setup() {

DDRB=0xFF;DDRC=0xFF;//指派PORTB，PORTC為輸出

}

void loop() {

for (int i=0;i<=7;i++){

PORTC=c[i];//位址

PORTB=0xff;//資料

delay(500); //請修改為1，並比較其效果

}

}

利用人類有視覺暫留的原理，只要電腦數度夠快，就可以顯示任何文字與影像。

文字數位化

　　若要將文數字顯示在此8\*8 的點陣LED，也是要將此文字數位化，將此文數字資料數位化的步驟如下：將此文數字寫在以下方格紙，並計算每行(Column) 的值。以下是將『洪』寫在方格紙上，則C1 值二進位是B10010001（高位元在下面），以16 進位表示是0x91、C2 值是0x4A，依此類推

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 |
| R1 | **1** | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| R2 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| R3 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| R4 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| R5 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
| R6 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| R7 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| R8 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| 值 | 0x91 | 0x4a | 0x80 | 0x52 | 0x3f | 0x32 | 0x5f | 0x92 |

將以上每一行的值以陣列儲存如下：（影像變識要將文字數位化，也是此相同的原理）

byte d[]={0x91,0x4a,0x80,0x52,0x3f,0x32,0x5f,0x92};

然後快速依序傳送到對應的行，例如，0x91送到C1、0x4A送到C2…我們稱此為掃描輸出，即可顯示『洪』，程式如下：

byte c[]={0x7f,0xbf,0xdf,0xef,0xf7,0xfb,0xfd,0xfe};

byte d[]={0x91,0x4a,0x80,0x52,0x3f,0x32,0x5f,0x92};

void setup() {

DDRB=0xFF;DDRC=0xFF;

}

void loop() {

for (int i=0 ;i<=7;i++){

PORTC=c[i];//位址

PORTB=d[i];//資料

delay(500);//delay(1)

}

}

自我練習

1. 請自行找一個筆畫較少的中文字，以能填入8\*8 方格為原則，計算其陣列值，並顯示在此8\*8 LED 上。

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 |
| R1 |  |  |  |  |  |  |  |  |
| R2 |  |  |  |  |  |  |  |  |
| R3 |  |  |  |  |  |  |  |  |
| R4 |  |  |  |  |  |  |  |  |
| R5 |  |  |  |  |  |  |  |  |
| R6 |  |  |  |  |  |  |  |  |
| R7 |  |  |  |  |  |  |  |  |
| R8 |  |  |  |  |  |  |  |  |
| 值 |  |  |  |  |  |  |  |  |

### 跑馬燈

### 上一單元，我們先顯示一個字，若要顯示的文字長度超過字幕機的寬度，可以使用跑馬燈的原理，請鍵入以下程式，並觀察執行結果。

byte c[]={0x7f,0xbf,0xdf,0xef,0xf7,0xfb,0xfd,0xfe};

byte d[]={0,0,0,0,0,0,0,0,0,0x91,0x4a,0x80,0x52,0x3f,0x32,0x5f,0x92,0,0,0x47,0x45,0x45,0x39,0,0,0,0x7c,0x12,0x11,0x12,0x7c,0,0,0,0,0,0,0,0,0,0};

int slen= 41;//此數字為以上d[]的長度

byte a[8];

void setup() {

DDRB=0xFF;

DDRC=0xFF;

}

void loop() {

//以跑馬燈左旋顯示文字

for (int k=0 ;k<=slen-8;k++){

for (int i=0;i<=7;i++){//依序每次抓8個

a[i]=d[i+k] ;

}

for (int i=0;i<=50;i++){//停留時間

for (int j=0 ;j<=7;j++){

PORTC=c[j];//位址

PORTB=a[j];//資料

delay(1);

}

}

}

}

本例以跑馬燈顯示『洪5A』，先將『洪5A』數位化，將『洪5A』寫在方格紙上，如下表，分別計算每一行的值如下，例如，C1是0x91，C2是0x4a，依此類推。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 |
| R1 | 1 | 0 | 0 | 0 | **1** | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| R2 | 0 | 1 | 0 | **1** | **1** | **1** | **1** | **1** | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| R3 | 0 | 0 | 0 | 0 | **1** | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| R4 | 0 | 1 | 0 | 0 | **1** | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| R5 | 1 | 0 | 0 | 1 | **1** | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |
| R6 | 0 | 0 | 0 | 0 | **1** | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| R7 | 0 | **1** | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| R8 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0x | 91 | 4a | 80 | 52 | 3f | 32 | 5f | 92 | 0 | 0 | 47 | 45 | 45 | 39 | 0 | 0 | 0 | 7c | 12 | 11 | 12 | 7c | 0 | 0 |

將以上資料以陣列儲存如下，文字前後都要加上空白0，這樣使用者才來得及看，slen是輸出文字的行數，也是d陣列的長度。

byte d[]={0,0,0,0,0,0,0,0,0,0x91,0x4a,0x80,0x52,0x3f,0x32,0x5f,0x92,0,0,0x47,0x45,0x45,0x39,0,0,0,0x7c,0x12,0x11,0x12,0x7c,0,0,0,0,0,0,0,0,0,0};

int slen=41;

現在逐一將d[]陣列，每次複製8筆資料到a[]陣列，

for (int i=0;i<=7;i++){//依序每次抓8個

a[i]=d[i+k] ;

}

第1次抓到0,0,0,0,0,0,0,0;第2次抓到0,0,0,0,0,0,0,0x91;第3次抓到0,0,0,0,0,0,0x91,0x4a…依此類推。再將a[]陣列，同上範例掃瞄輸出。但微處理機真的太快了，每一次的掃描還要重複一點時間，如以上程式的i迴圈。請自行調整i迴圈的值，即可調整跑馬燈輪轉速度。請留意每次抓8個，所以最後1次是抓全部長度減8（如以上程式的k迴圈的slen-8），不然會造成陣列溢位。

自我練習

1.請於以下填入適當的文字，並依序每次8筆資料輸出文字。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 |
| 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 5 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 6 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 7 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 8 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 0x |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

### 動畫

### 相信大家都看過紅綠燈下的小綠人動畫，動畫製作步驟如下：

### 先設計共幾個畫面。

### 逐一填入每個畫面的亮點。

### 逐一計算每個畫面8個col的值

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  | C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | | R1 |  |  |  |  |  |  |  |  | | R2 |  |  |  |  |  |  |  |  | | R3 |  |  |  |  |  |  |  |  | | R4 |  |  |  |  |  |  |  | 1 | | R5 |  |  |  |  |  |  |  |  | | R6 |  |  |  |  |  |  |  |  | | R7 |  |  |  |  |  |  |  |  | | R8 |  |  |  |  |  |  |  |  | | 值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 08 | | |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  | C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | | R1 |  |  |  |  |  |  |  |  | | R2 |  |  |  |  |  |  |  |  | | R3 |  |  |  |  |  |  |  | 1 | | R4 |  |  |  |  |  |  | 1 | 1 | | R5 |  |  |  |  |  |  |  | 1 | | R6 |  |  |  |  |  |  |  |  | | R7 |  |  |  |  |  |  |  |  | | R8 |  |  |  |  |  |  |  |  | | 值 | 0 | 0 | 0 | 0 | 0 | 0 | 08 | 1c | |
| |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  | C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | | R1 |  |  |  |  |  |  |  |  | | R2 |  |  |  |  |  |  |  | 1 | | R3 |  |  |  |  |  |  | 1 | 1 | | R4 |  |  |  |  |  | 1 | 1 | 1 | | R5 |  |  |  |  |  |  | 1 | 1 | | R6 |  |  |  |  |  |  |  | 1 | | R7 |  |  |  |  |  |  |  |  | | R8 |  |  |  |  |  |  |  |  | | 值 |  |  |  |  |  | 08 | 1c | 3e | | |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  | C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | | R1 |  |  |  |  |  |  |  | 1 | | R2 |  |  |  |  |  |  | 1 | 1 | | R3 |  |  |  |  |  | 1 | 1 | 1 | | R4 |  |  |  |  | 1 | 1 | 1 | 1 | | R5 |  |  |  |  |  | 1 | 1 | 1 | | R6 |  |  |  |  |  |  | 1 | 1 | | R7 |  |  |  |  |  |  |  | 1 | | R8 |  |  |  |  |  |  |  |  | | 值 |  |  |  |  | 08 | 1c | 3e | 7f | |

設計步驟

1. 將以上資料以二維陣列儲存，如以下e陣列。
2. 每次從e陣列抓8個，放到d陣列。
3. 掃描輸出d陣列。

byte i=0;

byte c[]={0x7f,0xbf,0xdf,0xef,0xf7,0xfb,0xfd,0xfe};//位址

byte d[8];

byte e[4][8]={ { 0, 0, 0, 0, 0, 0, 0, 0x08},

{ 0, 0, 0, 0, 0, 0, 0x08,0x1c},

{ 0, 0, 0, 0, 0, 0x08,0x1c,0x3e},

{ 0, 0, 0, 0, 0x08,0x1c,0x3e,0x7f}};

long t,t1,t2;

void setup() {

DDRB=0xff;

DDRC=0xff;

}

void loop() {

// 每次從e陣列抓8個，放到d陣列。

for (int j=0 ;j<=7;j++)

d[j]=e[i][j];

//掃描輸出

t1= millis();

do{

for (int j=0 ;j<=7;j++){

PORTC=c[j];//位址

PORTB=d[j];//資料

delay(1);

}

t2=millis();

t=t2-t1;

}while( (t<400) );//0.4秒，根據需求調整

i=(i+1) % 4;//本例共4個畫面

}

自我練習

1. 同上範例，請製作6個畫面，且讓箭號逐一往下。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  | C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | | R1 |  |  |  |  |  |  |  |  | | R2 |  |  |  |  |  |  |  |  | | R3 |  |  |  |  |  |  |  |  | | R4 |  |  |  |  |  |  |  |  | | R5 |  |  |  |  |  |  |  |  | | R6 |  |  |  |  |  |  |  |  | | R7 |  |  |  |  |  |  |  |  | | R8 |  |  |  |  |  |  |  |  | | 值 |  |  |  |  |  |  |  |  | | |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  | C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | | R1 |  |  |  |  |  |  |  |  | | R2 |  |  |  |  |  |  |  |  | | R3 |  |  |  |  |  |  |  |  | | R4 |  |  |  |  |  |  |  |  | | R5 |  |  |  |  |  |  |  |  | | R6 |  |  |  |  |  |  |  |  | | R7 |  |  |  |  |  |  |  |  | | R8 |  |  |  |  |  |  |  |  | | 值 |  |  |  |  |  |  |  |  | | |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  | C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | | R1 |  |  |  |  |  |  |  |  | | R2 |  |  |  |  |  |  |  |  | | R3 |  |  |  |  |  |  |  |  | | R4 |  |  |  |  |  |  |  |  | | R5 |  |  |  |  |  |  |  |  | | R6 |  |  |  |  |  |  |  |  | | R7 |  |  |  |  |  |  |  |  | | R8 |  |  |  |  |  |  |  |  | | 值 |  |  |  |  |  |  |  |  | |
| |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  | C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | | R1 |  |  |  |  |  |  |  |  | | R2 |  |  |  |  |  |  |  |  | | R3 |  |  |  |  |  |  |  |  | | R4 |  |  |  |  |  |  |  |  | | R5 |  |  |  |  |  |  |  |  | | R6 |  |  |  |  |  |  |  |  | | R7 |  |  |  |  |  |  |  |  | | R8 |  |  |  |  |  |  |  |  | | 值 |  |  |  |  |  |  |  |  | | |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  | C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | | R1 |  |  |  |  |  |  |  |  | | R2 |  |  |  |  |  |  |  |  | | R3 |  |  |  |  |  |  |  |  | | R4 |  |  |  |  |  |  |  |  | | R5 |  |  |  |  |  |  |  |  | | R6 |  |  |  |  |  |  |  |  | | R7 |  |  |  |  |  |  |  |  | | R8 |  |  |  |  |  |  |  |  | | 值 |  |  |  |  |  |  |  |  | | |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  | C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | | R1 |  |  |  |  |  |  |  |  | | R2 |  |  |  |  |  |  |  |  | | R3 |  |  |  |  |  |  |  |  | | R4 |  |  |  |  |  |  |  |  | | R5 |  |  |  |  |  |  |  |  | | R6 |  |  |  |  |  |  |  |  | | R7 |  |  |  |  |  |  |  |  | | R8 |  |  |  |  |  |  |  |  | | 值 |  |  |  |  |  |  |  |  | |

2.請發揮自己的美感，製作一個動物跑、或飛、或跳的畫面。

告白板

以上是跑馬燈的輸出，輸出效果比較平淡，適用於一般公告，若讓文字逐一閃爍顯示，那會比較霸氣，這樣可用在告白板、升官、競賽、考試榜單等的顯示。例如：以下程式逐一顯示8,2,5，每1秒自動變換1個數字。

byte i=0;

byte c[]={0x7f,0xbf,0xdf,0xef,0xf7,0xfb,0xfd,0xfe};//位址

byte d[8];

byte e[6][8]={ { 0,0x36, 0x49, 0x49,0x36,0, 0, 0},

{ 0, 0xe2, 0x91, 0x89, 0x86, 0, 0, 0},

{ 0, 0x8f, 0x89, 0x89, 0xf9, 0, 0, 0}};

long t,t1,t2;

void setup() {

DDRB=0xff;

DDRC=0xff;

}

void loop() {

//放字型到d陣列，每次8個

for (int j=0 ;j<=7;j++)

d[j]=e[i][j];

//掃描輸出

t1= millis();

do{

for (int j=0 ;j<=7;j++){

PORTC=c[j];//位址

PORTB=d[j];//資料

delay(1);

}

t2=millis();

t=t2-t1;

}while( (t<1000) );//1秒，根據需求調整

i=(i+1) % 3;//本例3個字

}

以上程式如何思考呢？以下是我的思考步驟：

1. 依序將以上數字8,2,5數位化，每組8個byte，如以下d[]陣列。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  | C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | | R1 |  |  | 1 | 1 |  |  |  |  | | R2 |  | 1 |  |  | 1 |  |  |  | | R3 |  | 1 |  |  | 1 |  |  |  | | R4 |  |  | 1 | 1 |  |  |  |  | | R5 |  | 1 |  |  | 1 |  |  |  | | R6 |  | 1 |  |  | 1 |  |  |  | | R7 |  |  | 1 | 1 |  |  |  |  | | R8 |  |  |  |  |  |  |  |  | | 值 | 0 | 36 | 49 | 49 | 36 | 0 | 0 | 0 | | |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  | C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | | R1 |  |  | 1 | 1 |  |  |  |  | | R2 |  | 1 |  |  | 1 |  |  |  | | R3 |  |  |  |  | 1 |  |  |  | | R4 |  |  |  | 1 |  |  |  |  | | R5 |  |  | 1 |  |  |  |  |  | | R6 |  | 1 |  |  |  |  |  |  | | R7 |  | 1 |  |  |  |  |  |  | | R8 |  | 1 | 1 | 1 | 1 |  |  |  | | 值 | 0 | e2 | 91 | 89 | 86 | 0 | 0 | 0 | |
| |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  | C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | | R1 |  | 1 | 1 | 1 | 1 |  |  |  | | R2 |  | 1 |  |  |  |  |  |  | | R3 |  | 1 |  |  |  |  |  |  | | R4 |  | 1 | 1 | 1 | 1 |  |  |  | | R5 |  |  |  |  | 1 |  |  |  | | R6 |  |  |  |  | 1 |  |  |  | | R7 |  |  |  |  | 1 |  |  |  | | R8 |  | 1 | 1 | 1 | 1 |  |  |  | | 值 | 0 | 8f | 89 | 89 | f9 | 0 | 0 | 0 | |  |

2. 可用一維陣列儲存，則此一維陣列長度為24，如下：

byte d[]= { 0,0x36, 0x49, 0x49,0x36,0, 0, 0,

0, 0xe2, 0x91, 0x89, 0x86, 0, 0, 0,

0, 0x8f, 0x89, 0x89, 0xf9, 0, 0, 0};

3. 也可用二維陣列儲存此3組輸出碼如下：

byte e[6][8]={ { 0,0x36, 0x49, 0x49,0x36,0, 0, 0},

{ 0, 0xe2, 0x91, 0x89, 0x86, 0, 0, 0},

{ 0, 0x8f, 0x89, 0x89, 0xf9, 0, 0, 0};

4. 可用時間、也可用按鈕變換要輸出的數字。

5. 不論是一維陣列或二維陣列，每次都要取8個byte進行掃描輸出，二維陣列的寫法，如以上程式。

自我練習

1.請自行找4個筆畫少的文數字，以文字能辨識為原則，顯示此文數字。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  | C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | | R1 |  |  |  |  |  |  |  |  | | R2 |  |  |  |  |  |  |  |  | | R3 |  |  |  |  |  |  |  |  | | R4 |  |  |  |  |  |  |  |  | | R5 |  |  |  |  |  |  |  |  | | R6 |  |  |  |  |  |  |  |  | | R7 |  |  |  |  |  |  |  |  | | R8 |  |  |  |  |  |  |  |  | | 值 |  |  |  |  |  |  |  |  | | |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  | C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | | R1 |  |  |  |  |  |  |  |  | | R2 |  |  |  |  |  |  |  |  | | R3 |  |  |  |  |  |  |  |  | | R4 |  |  |  |  |  |  |  |  | | R5 |  |  |  |  |  |  |  |  | | R6 |  |  |  |  |  |  |  |  | | R7 |  |  |  |  |  |  |  |  | | R8 |  |  |  |  |  |  |  |  | | 值 |  |  |  |  |  |  |  |  | |
| |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  | C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | | R1 |  |  |  |  |  |  |  |  | | R2 |  |  |  |  |  |  |  |  | | R3 |  |  |  |  |  |  |  |  | | R4 |  |  |  |  |  |  |  |  | | R5 |  |  |  |  |  |  |  |  | | R6 |  |  |  |  |  |  |  |  | | R7 |  |  |  |  |  |  |  |  | | R8 |  |  |  |  |  |  |  |  | | 值 |  |  |  |  |  |  |  |  | | |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  | C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | | R1 |  |  |  |  |  |  |  |  | | R2 |  |  |  |  |  |  |  |  | | R3 |  |  |  |  |  |  |  |  | | R4 |  |  |  |  |  |  |  |  | | R5 |  |  |  |  |  |  |  |  | | R6 |  |  |  |  |  |  |  |  | | R7 |  |  |  |  |  |  |  |  | | R8 |  |  |  |  |  |  |  |  | | 值 |  |  |  |  |  |  |  |  | |

https://meet.google.com/cby-wmza-yvk

1. 請用按鈕控制，每按一下，可變換一個字元。

第5單元 人工智慧與字型輸出碼

前面使用人工的方式計算所要顯示的字型每行的值，只是方便讀者瞭解電腦的掃瞄原理，實用的方式當然要讓電腦算。要請電腦算，那此軟體要有圖形處理能力，目前能處理圖形的軟體都有此方法，例如VB6、VB.NET、C#、Java、Python等都有此方法。其中又以VB6編譯器的程式較輕巧，很容易取得與安裝，所以我在此推薦使用VB6（備註：若您偏愛VB.NET、C＃、Java或Python，請自行參考筆者相關著作）。我們以VB6為例，VB6的Point() 函式，可逐一檢驗哪些是亮點。Point() 函式的簡要語法如下：

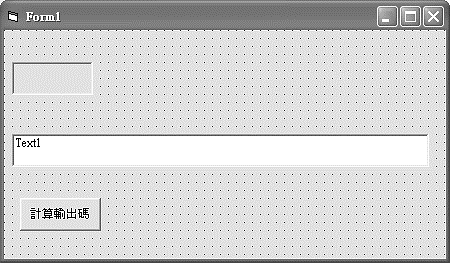
Color=Point(x,y)

也就是傳回座標(x,y)的顏色值，若該點黑色，則其值為0，其他顏色則其值不為0（備註：黑點就是完全不亮）。所以可將一個中文字寫在一個圖形控制項(Picture1)，再使用雙迴圈掃瞄，並統計有哪些亮點，將這些亮點放在d陣列，以上演算法的VB6表單配置如下：

1. 配置一個Picture控制項，且修改其Name屬性為pic，如下圖：

2. Picture控制項的座標原點在左上角，x座標向右為正，y座標向下為正。（本人Visual Basic 6程式設計，全華出版，有詳細說明。）

3. 配置一個Text控制項，其Name為Text1，以便放置LED輸出碼。



Picture控制項，本例Name改為pic

以上計算輸出碼的VB6程式如下：

Private Sub Command1\_Click()

'請先安排一個Picturebox,修改Name為 pic

pic.ScaleMode = 3 '以像素為座標單位

pic.FontSize = 8

pic.CurrentX = 0: pic.CurrentY = 0 '設定輸出位置

pic.Print "6"

'pic.Print "洪"

s = "byte d[]={"

For i = 0 To 7 '由左到右，計算每個Column

p = 0

For j = 1 To 8

If pic.Point(i, j) = 0 Then '黑點

p = p + 2 ^ (j - 1)

End If

Next

If i = 7 Then

s = s + Str(p) + "};"

Else

s = s + Str(p) + ","

End If

Next

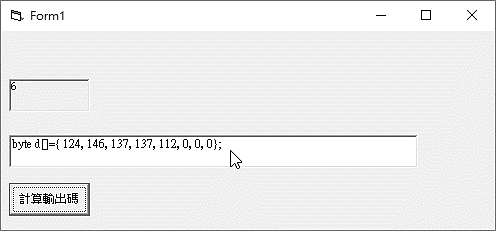
Text1.Text = s

End Sub

以上程式，請對照下表（已經填入數字6）就會懂，假如是黑點，那就依其位置（R1是權值20，R2權值是21…，R8權值是27），累加其值。外迴圈for i=0 to 7分別對照下表C1..C8，內迴圈for j=1 to 8，分別對照下表的R1..R8的黑點(1)，至於為什麼是j是1 to 8，而不是0到7，那是因為文字與文字之間上下『列距』的關係。

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | 2的次方 | 權重 |
| R1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| R2 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2 |
| R3 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 4 |
| R4 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 3 | 8 |
| R5 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 4 | 16 |
| R6 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 5 | 32 |
| R7 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 6 | 64 |
| R8 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 7 | 128 |
| 值 | 124 | 146 | 137 | 137 | 112 | 0 | 0 | 0 |  |  |

請將以下『6』的輸出碼複製，並取代前面Arduino 程式的d[]陣列即可。



以下程式，即可於共陰8\*8點陣LED顯示『6』。d陣列取自以上表單。

byte c[]={0x7f,0xbf,0xdf,0xef,0xf7,0xfb,0xfd,0xfe};

byte d[]={124,146,137,137,112,0,0,0};

void setup() {

DDRB=0xFF;

DDRC=0xFF;

}

void loop() {

for (int i=0 ;i<=7;i++){

PORTC=c[i];//位址

PORTB=d[i];//資料

delay(1);//delay(500)

}

}

#### 補充說明

1. Point是所有影像處理必用的指令，筆者30年前就使用此指令作出影像處理程式，將螢幕的圖形或文字使用銑床雕刻機雕刻，而得到教育部微電腦應用創意競賽優等的第一名。其次，Point()還可用來製作雕刻印章、寫毛筆字、印刷電路版的鑽孔；蛋糕、咖啡上面噴圖案等程式，這有待讀者進一步研究。

程式產生器

**大家耳熟能詳的Scratch和App Inventor都有程式產生器的功能，使用者只要拖拉一些物件，就會幫您產生程式碼。前面第7單元，我們已經用電腦完成計算**d[]陣列**，使用者現在就可以將這些資料套到前面的掃描輸出程式，這樣就可以輸出這些畫面。但是，這樣還是有點複雜，若要將此程式推廣到普羅大眾，還要撰寫程式產生器，將以上步驟一氣呵成，以下則繼續使用VB6寫出一段程式，將以上『資料』和『程式』連結的動作能自動完成，這樣就可讓不會程式的普羅大眾也能使用您的硬體設備。**

**範例 請將以上計算輸出碼『資料』和『程式』的連結的動作，寫程式自動完成。**

**執行結果 如圖1**

|  |  |
| --- | --- |
|  |  |
| **圖1 程式執行結果** | **圖2表單配置** |

**設計步驟**

1. **表單配置如圖2。**
2. 配置一個Text1，其Name為Text1，可以讓使用者輸入文數字。
3. 配置一個PictureBox，Name 屬性為Picture1，請改為pic，其功能是放入文數字，然後在此計算黑點。
4. 配置一個Text，其Name為Text2，MultiLine為True，可以輸出Arduino程式。
5. 配置一個CommandButton，其Name為Command1，此為按鈕功能，開始執行本程式功能。
6. 配置一個CommandButton，其Name為Command2，此為按鈕功能，將Text2內的程式複製到剪貼簿。
7. **程式設計如下：**

**Private Sub Command1\_Click()**

**s = ""**

**lf = Chr(13) + Chr(10) '跳列**

**'請依照Arduino程式，將程式先放好**

**s = s + "byte c[]={0x7f,0xbf,0xdf,0xef,0xf7,0xfb,0xfd,0xfe};" + lf**

**s = s + "byte d[]={"**

**'準備計算文字的輸出碼**

**'VB PictureBox 設定**

**pic.ScaleMode = 3 '以像素為座標單位**

**pic.FontSize = 8**

**pic.CurrentX = 0: pic.CurrentY = 0 '設定輸出位置**

**'寫入文字**

**pic.Print text1.Text**

**'pic.Print "洪"**

**For i = 0 To 7 '由左到右，計算每個Column**

**p = 0**

**For j = 1 To 8**

**If pic.Point(i, j) = 0 Then '黑點**

**p = p + 2 ^ (j - 1) '依照權重累加黑點**

**End If**

**Next**

**If i = 7 Then**

**s = s + Str(p) + "};" + lf '計算完畢**

**Else**

**s = s + Str(p) + "," '完成一個黑點**

**End If**

**Next**

**'將後段程式合併**

**s = s + "void setup() { " + lf**

**s = s + " DDRB=0xFF;" + lf**

**s = s + " DDRC=0xFF;" + lf**

**s = s + "}" + lf**

**s = s + "void loop() {" + lf**

**s = s + " for (int i=0 ;i<=7;i++){" + lf**

**s = s + " PORTC=c[i];//位址" + lf**

**s = s + " PORTB=d[i];//資料" + lf**

**s = s + " delay(1);//delay(500)" + lf**

**s = s + " }" + lf**

**s = s + "}" + lf**

**'將程式碼輸出**

**Text2.Text = s**

**End Sub**

**'複製程式碼到剪貼簿**

**Private Sub Command2\_Click()**

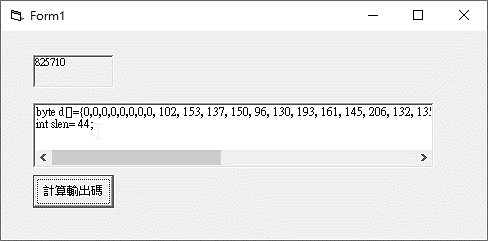
**Clipboard.Clear**

**Clipboard.SetText Text2.Text**

**End Sub**

### 跑馬燈與人工智慧

前面我們已經使用人工計算跑馬燈的輸出碼，現在我們要由電腦計算。例如，若要在本電路以跑馬燈顯示825710，則應先計算字型如下，文字前後都要加上空白0，這樣使用者才來得及看。



上圖的slen是輸出文字的行數，也是d陣列的長度，亦在VB6程式計算而得，且要傳出，以上VB6程式如下：

Private Sub Command1\_Click()

'請先安排一個Picturebox,修改Name為 pic

pic.ScaleMode = 3 '以像素為座標單位

pic.FontSize = 8

pic.CurrentX = 0: pic.CurrentY = 0 '輸出位置

s = "825710"

slen = Len(s) '字元數

pic.Print s

s = "byte a[]={0,0,0,0,0,0,0,0," '前面加一些空白

For i = 0 To 6 \* slen '文字寬度

p = 0

For j = 1 To 8

If pic.Point(i, j) = 0 Then '黑點

p = p + 2 ^ (j - 1)

End If

Next

If i = 6 \* slen Then

s = s + Str(p) + "};"

Else

s = s + Str(p) + ","

End If

Next

s = s + Chr(13) + Chr(10) '跳列

slen = 8 + 6 \* slen '行數也要傳出

s = s + "int slen=" + Str(slen) + ";"

Text1.Text = s

End Sub

Arduino的資料跑馬燈程式如下，d[]是『825710』的輸出碼，現在逐一每次將d陣列複製8筆資料到a陣列，第1次是0,0,0,0,0,0,0,0;第2次是0,0,0,0,0,0,0,102;第3次是0,0,0,0,0,0,102,153…依此類推，再將a陣列掃瞄輸出，程式如下：

int i,j,k;

byte c[]={0x7f,0xbf,0xdf,0xef,0xf7,0xfb,0xfd,0xfe};

byte d[]={0,0,0,0,0,0,0,0, 102, 153, 137, 150, 96, 130, 193, 161, 145, 206, 132, 135, 137, 113, 0, 2, 1, 225, 25, 7, 0, 130, 255, 128, 0, 126, 129, 129, 129, 126, 0, 0, 0, 0, 0, 0, 0};

int slen= 44;

byte a[8];

void setup() {

DDRB=0xFF;

DDRC=0xFF;

}

void loop() {

//以跑馬燈左旋顯示文字

for (k=0 ;k<=slen-8;k++){

for (i=0;i<=7;i++){

a[i]=d[i+k] ;

}

for (i=0;i<=50;i++){//停留時間

for (j=0 ;j<=7;j++){

PORTC=c[j];//位址

PORTB=a[j];//資料

delay(1);

}

}

}

}

查表顯示文字

前面我們已經使用人工的方式，分別計算所要顯示的文數字，但每次都要計算一次，有點浪費記憶體，所以可以先將數字0到9通通依序計算其輸出碼，通通放到陣列，如以下e陣列，然後每要輸出一個數字，則查表一次，找到對應輸出碼，程式如下：

以下是逐一顯示0～9

byte i=0;

byte c[]={0x7f,0xbf,0xdf,0xef,0xf7,0xfb,0xfd,0xfe};//位址

byte d[8];

//將數字0到9依序計算其輸出碼，放到陣列

byte e[10][8]={{ 126, 129, 129, 129, 126, 0, 0, 0},

{ 0, 130, 255, 128, 0, 0, 0, 0},

{ 130, 193, 161, 145, 206, 0, 0, 0},

{ 130, 137, 137, 118, 0, 0, 0, 0},

{ 48, 40, 38, 255, 32, 0, 0, 0},

{ 132, 135, 137, 113, 0, 0, 0, 0},

{ 124, 146, 137, 137, 112, 0, 0, 0},

{ 2, 1, 225, 25, 7, 0, 0, 0},

{ 102, 153, 137, 150, 96, 0, 0, 0},

{ 14, 145, 145, 81, 62, 0, 0, 0}};

long t,t1,t2;

void setup() {

DDRB=0xFF;

DDRC=0xFF;

Serial.begin(9600);

}

void loop() {

//將對應的輸出碼，交給d[]陣列

for (int j=0 ;j<=7;j++)

d[j]=e[i][j];

//掃描輸出

t1= millis();

do{

for (int j=0 ;j<=7;j++){

PORTC=c[j];//位址

PORTB=d[j];//資料

delay(1);

}

t2=millis();

t=t2-t1;

}while( (t<1000) );//1秒，根據需求調整

i=(i+1) % 10;//本例6個字

}

以下是計數器

byte i=0;

byte c[]={0x7f,0xbf,0xdf,0xef,0xf7,0xfb,0xfd,0xfe};//位址

byte d[8];

//將數字0到9依序計算其輸出碼，放到陣列

byte e[10][8]={{ 126, 129, 129, 129, 126, 0, 0, 0},

{ 0, 130, 255, 128, 0, 0, 0, 0},

{ 130, 193, 161, 145, 206, 0, 0, 0},

{ 130, 137, 137, 118, 0, 0, 0, 0},

{ 48, 40, 38, 255, 32, 0, 0, 0},

{ 132, 135, 137, 113, 0, 0, 0, 0},

{ 124, 146, 137, 137, 112, 0, 0, 0},

{ 2, 1, 225, 25, 7, 0, 0, 0},

{ 102, 153, 137, 150, 96, 0, 0, 0},

{ 14, 145, 145, 81, 62, 0, 0, 0}};

long t,t1,t2;

const byte pup=29;//上數

const byte pdown=28;//下數

void setup() {

DDRB=0xFF;

DDRC=0xFF;

pinMode(29,INPUT\_PULLUP);

pinMode(28,INPUT\_PULLUP);

Serial.begin(9600);

}

void loop() {

byte a=digitalRead(pup);

if (a==LOW){

while (digitalRead(pup)==LOW)

delay(10);

i++;

}

byte b=digitalRead(pdown);

if (b==LOW){

while (digitalRead(pdown)==LOW)

delay(10);

i--;

}

//將對應的輸出碼，交給d[]陣列

for (int j=0 ;j<=7;j++)

d[j]=e[i][j];

//掃描輸出

t1= millis();

do{

for (int j=0 ;j<=7;j++){

PORTC=c[j];//位址

PORTB=d[j];//資料

delay(1);

}

t2=millis();

t=t2-t1;

}while( (t<1000) );//1秒，根據需求調整

}

以下是查表顯示對應數字

byte i=0;

byte c[]={0x7f,0xbf,0xdf,0xef,0xf7,0xfb,0xfd,0xfe};//位址

byte d[8];

//將數字0到9依序計算其輸出碼，放到陣列

byte e[10][8]={{ 126, 129, 129, 129, 126, 0, 0, 0},

{ 0, 130, 255, 128, 0, 0, 0, 0},

{ 130, 193, 161, 145, 206, 0, 0, 0},

{ 130, 137, 137, 118, 0, 0, 0, 0},

{ 48, 40, 38, 255, 32, 0, 0, 0},

{ 132, 135, 137, 113, 0, 0, 0, 0},

{ 124, 146, 137, 137, 112, 0, 0, 0},

{ 2, 1, 225, 25, 7, 0, 0, 0},

{ 102, 153, 137, 150, 96, 0, 0, 0},

{ 14, 145, 145, 81, 62, 0, 0, 0}};

long t,t1,t2;

String f="825710";

void setup() {

DDRB=0xFF;

DDRC=0xFF;

Serial.begin(9600);

}

void loop() {

//每次抓一個

byte g=byte(f[i]) ; //取到的字元，是其ASCII碼

Serial.println(g);

g=g-48; //取到的字元，是其ASCII碼，所以要減48

Serial.println(g);

//將對應的輸出碼，交給d[]陣列

for (int j=0 ;j<=7;j++)

d[j]=e[g][j];

//掃描輸出

t1= millis();

do{

for (int j=0 ;j<=7;j++){

PORTC=c[j];//位址

PORTB=d[j];//資料

delay(1);

}

t2=millis();

t=t2-t1;

}while( (t<1000) );//1秒，根據需求調整

i=(i+1) % 6;//本例6個字

}

以下是用指撥開關顯示對應數字。

byte i=0;

byte c[]={0x7f,0xbf,0xdf,0xef,0xf7,0xfb,0xfd,0xfe};//位址

byte d[8];

//將數字0到9依序計算其輸出碼，放到陣列

byte e[10][8]={{ 126, 129, 129, 129, 126, 0, 0, 0},

{ 0, 130, 255, 128, 0, 0, 0, 0},

{ 130, 193, 161, 145, 206, 0, 0, 0},

{ 130, 137, 137, 118, 0, 0, 0, 0},

{ 48, 40, 38, 255, 32, 0, 0, 0},

{ 132, 135, 137, 113, 0, 0, 0, 0},

{ 124, 146, 137, 137, 112, 0, 0, 0},

{ 2, 1, 225, 25, 7, 0, 0, 0},

{ 102, 153, 137, 150, 96, 0, 0, 0},

{ 14, 145, 145, 81, 62, 0, 0, 0}};

long t,t1,t2;

void setup() {

DDRB=0xFF;

DDRC=0xFF;

pinMode(47,INPUT\_PULLUP);

pinMode(48,INPUT\_PULLUP);

pinMode(49,INPUT\_PULLUP);

Serial.begin(9600);

}

void loop() {

byte i=~PINL;

i=i & B00000111;

Serial.print(i);

//將對應的輸出碼，交給d[]陣列

for (int j=0 ;j<=7;j++)

d[j]=e[i][j];

//掃描輸出

t1= millis();

do{

for (int j=0 ;j<=7;j++){

PORTC=c[j];//位址

PORTB=d[j];//資料

delay(1);

}

t2=millis();

t=t2-t1;

}while( (t<1000) );//1秒，根據需求調整

}

#### 自我練習

1. 若使用一維陣列儲存，那程式如何寫呢？

2. 如何使用按鈕變換數字？例如，每按一下按鈕，變換下一個文字。

3、請寫一程式，可以每按一下按鈕，就產生1個1到6的亂數，且由點陣LED輸出。

4、請寫一程式，可以用8個指撥開關指派輸出1到8的數字。

5、請寫一程式，可以用電腦的鍵盤輸入1到8的數字，且顯示。

6、請寫一程式，可以輸入一串0到9的數字，且顯示。例如，輸入『825710』，那就顯示『825710』。

### 明滅

要控制點陣LED明滅，一樣要掃瞄空白碼(0x00)一個短暫的時間，測試程式如下：

byte i=0;

byte c[]={0x7f,0xbf,0xdf,0xef,0xf7,0xfb,0xfd,0xfe};//位址

byte d[8];

byte e[10][8]={{ 126, 129, 129, 129, 126, 0, 0, 0},

{ 0, 130, 255, 128, 0, 0, 0, 0},

{ 130, 193, 161, 145, 206, 0, 0, 0},

{ 130, 137, 137, 118, 0, 0, 0, 0},

{ 48, 40, 38, 255, 32, 0, 0, 0},

{ 132, 135, 137, 113, 0, 0, 0, 0},

{ 124, 146, 137, 137, 112, 0, 0, 0},

{ 2, 1, 225, 25, 7, 0, 0, 0},

{ 102, 153, 137, 150, 96, 0, 0, 0},

{ 14, 145, 145, 81, 62, 0, 0, 0}};

long t,t1,t2;

String f="825710";

void setup() {

DDRB=0xFF;

DDRC=0xFF;

Serial.begin(9600);

}

void loop() {

//放字型

byte g=byte(f[i]) ;

Serial.println(g);

g=g-48;

Serial.println(g);

for (int j=0 ;j<=7;j++)

d[j]=e[g][j];

//掃描輸出

t1= millis();

do{

for (int j=0 ;j<=7;j++){

PORTC=c[j];//位址

PORTB=d[j];//資料

delay(1);

}

t2=millis();

t=t2-t1;

}while( (t<1000) );//1秒，根據需求調整

//滅

for (int j=0;j<=50;j++){//停留時間

for (int k=0 ;k<=7;k++){

PORTC=c[k];//位址

PORTB=0;//資料

delay(1);

}

}

i=(i+1) % 6;//本例6個字

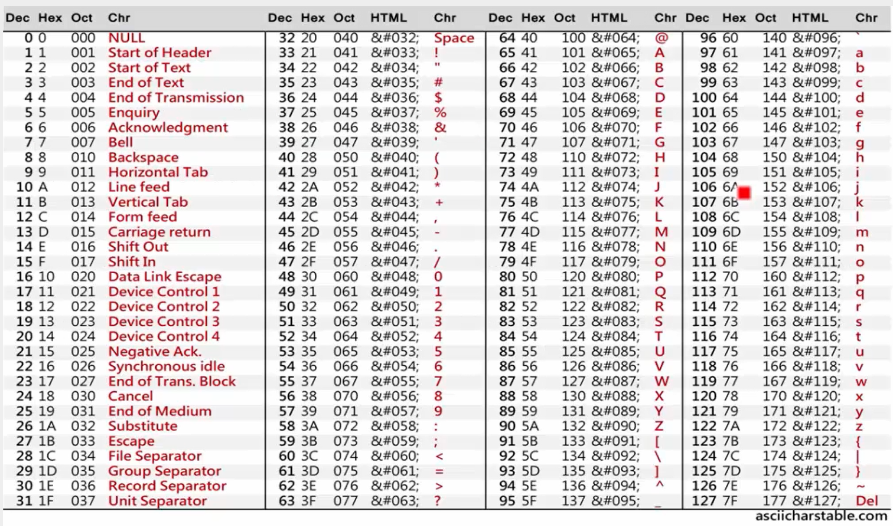
}

### 自我練習

### 1、請寫一程式，自動顯示3個1到6的亂數，每個亂數之間要先滅掉LED，再顯示下一個數字。

計算全部文數字

前面我們使用人工操作方式，由電腦分別計算0～9的輸出碼，但若要計算所有數字與大小寫字母，這樣還是耗時，以下我們寫程式計算所有數字與大小寫字母。ASCII碼如下表：



，所以是計算32～122，用電腦算所有文數字的VB程式如下：（更多的VB程式，或想要瞭解此VB程式的撰寫，請參考泉勝出版的『Ardunio字幕機自造與程式設計』）

1.撰寫VB程式如下：

Private Sub Command1\_Click()

lf = Chr(13) + Chr(10) '跳列

'請先安排一個Picturebox,修改Name為 pic

pic.ScaleMode = 3 '以像素為座標單位

pic.FontSize = 8

pic.CurrentX = 0: pic.CurrentY = 0 '輸出位置

's = "byte c[]={0x7f,0xbf,0xdf,0xef,0xf7,0xfb,0xfd,0xfe};" + lf

s = s + "byte e[][8]={ "

'當場計算e[][]

For k = 32 To 122 '計算ASCII從32到122

s = s + "{"

pic.Cls

pic.Print Chr(k)

For i = 0 To 7

p = 0

For j = 1 To 8

If pic.Point(i, j) = 0 Then '黑點

p = p + 2 ^ (j - 1)

End If

Next

If i = 7 Then

s = s + Str(p) + "}," + lf

Else

s = s + Str(p) + ","

End If

Next

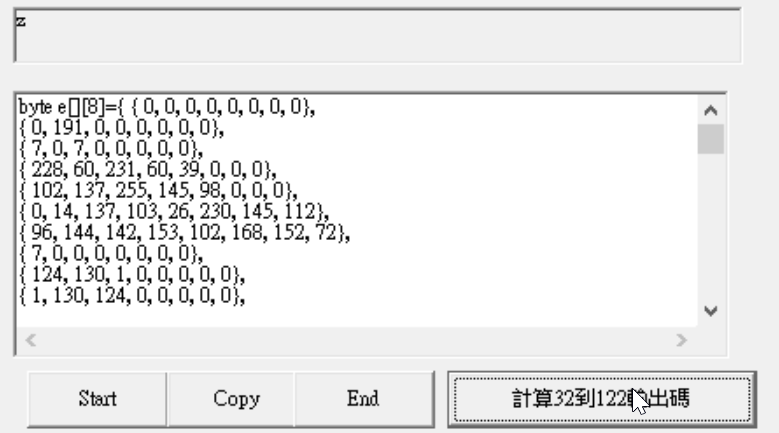
Next

s = s + "};"

Text2.Text = s

End Sub

9、以上程式執行結果如下圖：



2.將以上資料複製，配合以下Arduino程式，此程式與上一程式相同，就可顯示所有數字與大小寫字元。

byte i=0;

byte c[]={0x7f,0xbf,0xdf,0xef,0xf7,0xfb,0xfd,0xfe};//位址

byte d[8];

byte e[][8]={ { 0, 0, 0, 0, 0, 0, 0, 0},//32 is space

{ 0, 191, 0, 0, 0, 0, 0, 0},

{ 7, 0, 7, 0, 0, 0, 0, 0},

{ 228, 60, 231, 60, 39, 0, 0, 0},

{ 102, 137, 255, 145, 98, 0, 0, 0},

{ 0, 14, 137, 103, 26, 230, 145, 112},

{ 96, 144, 142, 153, 102, 168, 152, 72},

{ 7, 0, 0, 0, 0, 0, 0, 0},

{ 124, 130, 1, 0, 0, 0, 0, 0},

{ 1, 130, 124, 0, 0, 0, 0, 0},

{ 9, 6, 31, 6, 9, 0, 0, 0},

{ 16, 16, 254, 16, 16, 0, 0, 0},

{ 128, 128, 0, 0, 0, 0, 0, 0},

{ 32, 32, 32, 0, 0, 0, 0, 0},

{ 0, 128, 0, 0, 0, 0, 0, 0},

{ 224, 24, 7, 0, 0, 0, 0, 0},

{ 126, 129, 129, 129, 126, 0, 0, 0},

{ 0, 130, 255, 128, 0, 0, 0, 0},

{ 130, 193, 161, 145, 206, 0, 0, 0},

{ 130, 137, 137, 118, 0, 0, 0, 0},

{ 48, 40, 38, 255, 32, 0, 0, 0},

{ 132, 135, 137, 113, 0, 0, 0, 0},

{ 124, 146, 137, 137, 112, 0, 0, 0},

{ 2, 1, 225, 25, 7, 0, 0, 0},

{ 102, 153, 137, 150, 96, 0, 0, 0},

{ 14, 145, 145, 81, 62, 0, 0, 0},

{ 0, 136, 0, 0, 0, 0, 0, 0},

{ 0, 136, 136, 0, 0, 0, 0, 0},

{ 16, 40, 40, 68, 68, 0, 0, 0},

{ 72, 72, 72, 72, 72, 0, 0, 0},

{ 68, 68, 40, 40, 16, 0, 0, 0},

{ 2, 1, 177, 14, 0, 0, 0, 0},

{ 0, 60, 66, 153, 165, 157, 162, 92},

{ 128, 224, 156, 19, 156, 224, 128, 0},

{ 129, 255, 137, 137, 137, 118, 0, 0},

{ 60, 66, 129, 129, 130, 71, 0, 0},

{ 129, 255, 129, 129, 129, 66, 60, 0},

{ 129, 255, 137, 137, 157, 195, 0, 0},

{ 129, 255, 137, 9, 29, 3, 0, 0},

{ 60, 66, 129, 129, 146, 247, 16, 0},

{ 129, 255, 137, 8, 137, 255, 129, 0},

{ 129, 255, 129, 0, 0, 0, 0, 0},

{ 192, 129, 127, 1, 0, 0, 0, 0},

{ 129, 255, 137, 20, 163, 193, 128, 0},

{ 129, 255, 129, 128, 128, 192, 0, 0},

{ 129, 255, 134, 56, 192, 56, 134, 255},

{ 129, 255, 134, 24, 97, 255, 1, 0},

{ 60, 66, 129, 129, 129, 66, 60, 0},

{ 129, 255, 137, 9, 9, 6, 0, 0},

{ 60, 66, 129, 129, 129, 66, 60, 0},

{ 129, 255, 137, 25, 41, 198, 128, 0},

{ 230, 73, 145, 146, 103, 0, 0, 0},

{ 0, 3, 129, 255, 129, 3, 0, 0},

{ 1, 127, 129, 128, 129, 127, 1, 0},

{ 1, 7, 57, 192, 57, 7, 1, 0},

{ 1, 15, 56, 225, 31, 49, 192, 57},

{ 129, 195, 165, 24, 165, 195, 129, 0},

{ 1, 3, 141, 240, 141, 3, 1, 0},

{ 195, 161, 145, 137, 133, 195, 0, 0},

{ 0, 255, 1, 0, 0, 0, 0, 0},

{ 3, 28, 224, 0, 0, 0, 0, 0},

{ 1, 1, 255, 0, 0, 0, 0, 0},

{ 8, 6, 1, 6, 8, 0, 0, 0},

{ 0, 0, 0, 0, 0, 0, 0, 0},

{ 1, 2, 0, 0, 0, 0, 0, 0},

{ 80, 168, 168, 112, 128, 0, 0, 0},

{ 1, 127, 136, 136, 112, 0, 0, 0},

{ 112, 136, 136, 80, 0, 0, 0, 0},

{ 112, 136, 137, 127, 128, 0, 0, 0},

{ 112, 168, 168, 176, 0, 0, 0, 0},

{ 132, 254, 133, 1, 0, 0, 0, 0},

{ 144, 232, 168, 152, 8, 0, 0, 0},

{ 129, 255, 8, 240, 128, 0, 0, 0},

{ 136, 249, 128, 0, 0, 0, 0, 0},

{ 8, 249, 0, 0, 0, 0, 0, 0},

{ 129, 255, 32, 216, 136, 0, 0, 0},

{ 129, 255, 128, 0, 0, 0, 0, 0},

{ 8, 240, 8, 8, 240, 8, 8, 240},

{ 136, 248, 8, 240, 128, 0, 0, 0},

{ 112, 136, 136, 136, 112, 0, 0, 0},

{ 8, 248, 136, 136, 112, 0, 0, 0},

{ 112, 136, 136, 248, 8, 0, 0, 0},

{ 136, 240, 136, 0, 0, 0, 0, 0},

{ 144, 168, 168, 72, 0, 0, 0, 0},

{ 8, 126, 136, 0, 0, 0, 0, 0},

{ 8, 120, 128, 248, 128, 0, 0, 0},

{ 8, 56, 192, 56, 8, 0, 0, 0},

{ 8, 56, 192, 56, 192, 56, 8, 0},

{ 136, 216, 32, 216, 136, 0, 0, 0},

{ 8, 56, 192, 56, 8, 0, 0, 0},

{ 152, 200, 168, 152, 200, 0, 0, 0},

};

long t,t1,t2;

String f="This is a book 825710";

void setup() {

DDRB=0xFF;

DDRC=0xFF;

Serial.begin(9600);

}

void loop() {

//每次從f字串取一個字元

byte g=byte(f[i]) ; //取到的字元，是其ASCII碼

Serial.print(i);Serial.print(",");

Serial.println(g);

g=g-32; //取到的字元，是其ASCII碼，本例從32開始計算，所以要減32

Serial.println(g);

for (int j=0 ;j<=7;j++)

d[j]=e[g][j]; //查表，將該字元的8個輸出碼複製到d[]陣列

//掃描輸出d[]陣列

t1= millis();

do{

for (int j=0 ;j<=7;j++){

PORTC=c[j];//位址

PORTB=d[j];//資料

delay(1);

}

t2=millis();

t=t2-t1;

}while( (t<1000) );//1秒，根據需求調整

i=(i+1) % f.length();//長度

}

### 11、明滅。因為有時候連續兩個字元若相同，例如顯示『88255』，若沒有加上顯示空白碼，會變成『825』。要控制點陣LED明滅，一樣要掃瞄空白碼(0x0)一個短暫的時間，顯示空白碼的測試程式如下，請加在以上程式『掃描輸出』的後面就可以。

t1= millis();

do{

for (int j=0 ;j<=7;j++){

PORTC=c[j];//位址

PORTB=0;//資料滅掉

delay(1);

}

t2=millis();

t=t2-t1;

}while( (t<200) );//滅掉0.2秒，根據需求調整

#### 自我練習

1. 同上題，如何使用2個指撥開關調整文字輸出速度？

2. 計數器。如何使用按鈕變換數字？例如：每按一下按鈕，變換下一個文字。

3、請寫一程式，可以每按一下按鈕，就產生1個1到6的亂數，且由點陣LED輸出且停住一直顯示。

4、請寫一程式，使用8個按鈕，編號是1到8，每按一個按鈕，分別顯示對應的1到8。

5、請寫一程式，可以用8個指撥開關指派輸出0到8的數字。

6、請寫一程式，可以用3個指撥開關指派輸出0到7的數字。

### ※7、以上8\*8點陣LED僅能顯示數字、英文字母與筆畫很少的中文字，請有樂趣的同學，自行使用以上原理與演算法，自製16\*16與16\*64字幕機，這樣就可用跑馬燈或告白板顯示任何中文字。

六、16\*64字幕機操作

16\*64字幕機硬體電路設計如下圖：



Arduino的PORTA(由上而下分別是22,23,24,25,26,27,28,29)與PORTB(由上而下分別是53,52,51,50,10,11,12,13)分別連接到74244的輸入A端，再由74244的Y端連接到字幕機的R1..R16共16位元，本例稱為資料線；Arduino的PORTC (PC0,PC1..PC5由上而下分別是37,36,35,34,33,32)連接到兩層的74138，本例稱為控制線，這樣當PC給0到63時，C1..C64僅有一條線可以得到低電位，這樣就達到輪流掃描的功能。

### 74244

74244是一個電流緩衝器，由244供電給LED，這樣可以保護Arduino的控制晶片(ATmega2560)。

### 74138

本例一共有64個Column，但一顆單晶不可能有那麼多腳位可用，所以我們用74138擴充。此IC是3對8的解碼器，其真值表如下。

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| INPUTS | | | | | OUTPUTS | | | | | | | |
| ENABLE | | SELECT | | |
| G1 |  | C | B | A | Y0 | Y1 | Y2 | Y3 | Y4 | Y5 | Y6 | Y7 |
| X | H | X | X | X | H | H | H | H | H | H | H | H |
| L | X | X | X | X | H | H | H | H | H | H | H | H |
| H | L | L | L | L | L | H | H | H | H | H | H | H |
| H | L | L | L | H | H | L | H | H | H | H | H | H |
| H | L | L | H | L | H | H | L | H | H | H | H | H |
| H | L | L | H | H | H | H | H | L | H | H | H | H |
| H | L | H | L | L | H | H | H | H | L | H | H | H |
| H | L | H | L | H | H | H | H | H | H | L | H | H |
| H | L | H | H | L | H | H | H | H | H | H | L | H |
| H | L | H | H | H | H | H | H | H | H | H | H | L |

上圖=/G2A+/G2B

En=G1\*(/G2A+/G2B)

H = high level, L = low level, X = irrelevant

首先，G1接1，G2A與G2B都接0，C,B,A分別給000,001…111就可以讓Y0～Y7分別得到低電位。其次，我們用兩層74138如上圖， 第2層的PC5,PC4, PC3給000，Y0得到低電位，就選到第0個138；PC5,PC4, PC3給001， Y1得到低電位，選到第1個138；PC5,PC4, PC3給111，Y7得到低電位，選到第7個138；這樣就可以讓PC0,PC1,PC2,PC3,PC4,PC5等6個位元控制64行，使得這64行依次分別得低電位，那就可以得到掃描的效果。其次，若使用74LS138則亮度不足，為了提高亮度，請指定購買高功率74138或向本公司購買實驗板材料包。

軟體測試

以下程式可讓LED全亮。

void setup() {

DDRF=0xFF; DDRK=0xFF; DDRC=0xFF;

}

void loop() {

int i,j;

//逐行亮

for (i=0;i<=100;i++)

for (j=0 ;j<=63;j++){

PORTC=j;//位址

PORTF=0xff;//資料

PORTK=0xff;//資料

delay(1); //delay(1000);逐行亮

}

//全暗

for (i=0;i<=10;i++)

for (j=0 ;j<=63;j++){

PORTC=j;//位址

PORTF=0;//資料

PORTK=0;//資料

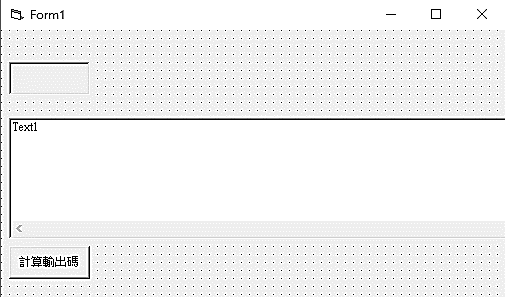
delay(1);

}

}

擷取輸出碼

擷取字型輸出碼的VB6表單配置如下：



擷取字型輸出碼的VB6程式如下：

Private Sub Command1\_Click()

'請先安排一個Picturebox,修改Name為 pic

pic.ScaleMode = 3 '以像素為座標單位

pic.FontSize = 12

pic.CurrentX = 0: pic.CurrentY = 0 '輸出位置

s = "泉勝出版"

pic.Print s

slen = 64 '行數

s = "const unsigned int a[]={"

For i = 0 To 64

p = 0

For j = 1 To 17

If pic.Point(i, j) = 0 Then '黑點

p = p + 2 ^ (j - 1)

End If

Next

If i = 64 Then

s = s + Str(p) + "};"

Else

s = s + Str(p) + ","

End If

Next

s = s + Chr(13) + Chr(10) '跳列

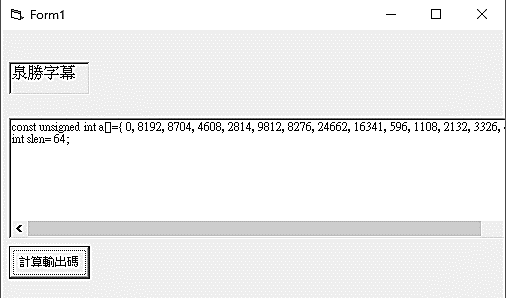
slen = 64 '行數也要傳出

s = s + "int slen=" + Str(slen) + ";"

Text1.Text = s

End Sub

以下畫面已經將輸入文字『泉勝字幕』轉為LED的輸出碼，請複製以下畫面的a[]陣列。



撰寫Arduino程式如下，其中a[]陣列從以上畫面複製而來。

//PORTF

//PORTK

//PORTC 37,36,35,34,33,32

unsigned int i;

const unsigned int a[]={ 0, 8192, 8704, 4608, 2814, 9812, 8276, 24662, 16341, 596, 1108, 2132, 3326, 4868, 12544, 4096, 0, 28672, 4095, 8466, 24850, 16383, 512, 16714, 13004, 3704, 9039, 25161, 8828, 7882, 328, 576, 0, 528, 526, 516, 548, 4644, 8740, 25125, 16294, 676, 628, 548, 516, 532, 782, 516, 0, 4610, 4610, 2562, 16123, 2734, 2986, 2730, 32424, 2730, 2730, 10926, 16123, 2562, 4867, 4610, 0};

void setup() {

DDRF=0xFF;

DDRK=0xFF;

DDRC=0xFF;

}

void loop() {

for (i=0;i<=63;i++) {

PORTC=i;

PORTK =a[i] / 256;

PORTF=a[i] % 256;

delayMicroseconds(100);

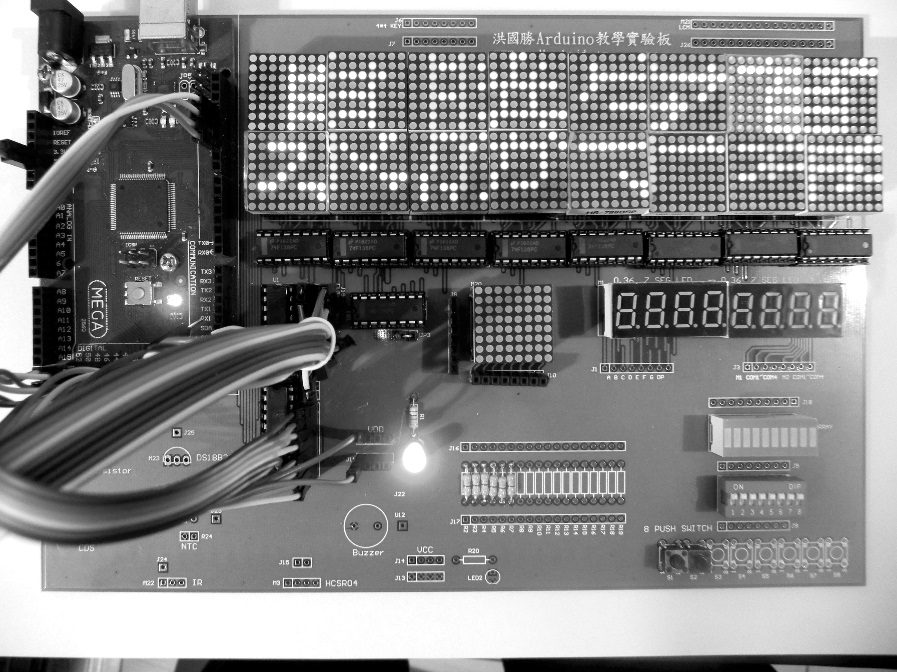
PORTA=0 ;

PORTB=0;

}

}

以上程式執行結果如下：



自我練習

1. 請自行撰寫跑馬燈程式。
2. 請自行撰寫貪食蛇遊戲。

課程編號：3552779

地點：義民高中敬業大樓5A電腦教室  
日期：111/11/17  
時間：13:00~16:00  
13:00~13:50 Arduino 晶片實驗板測試、開發環境  
14:00~14:50 資料數位化與驗證  
15:00~15:50 實作輸出程式

以下是貪食蛇程式

//8＊8點陣LED R1..R8 接PORTB 53,52,51,50,10,11,12,13

//C1..C8接PORTC 30,31,32,33,34,35,36,37,

byte c[]={0x7f,0xbf,0xdf,0xef,0xf7,0xfb,0xfd,0xfe};//位址

byte b[]={0,0,0,0,0,0,0,0};//資料

byte x[100],y[100];//蛇身位置

byte s;//蛇身長度

byte hx,hy;//蛇頭位置

const byte pleft=2;//左

const byte pup=3;//上鍵

const byte pdown=18;//下

const byte pright=19;//右

byte dir;//移動方向

byte wid=8;//螢幕大小

byte hei=8;//

//byte ndir;

byte ax,ay;//蘋果位置

void setup() {

Serial.begin(9600);//螢幕輸出

DDRB=0xFF;

DDRC=0xFF;

pinMode(pup,INPUT\_PULLUP);

pinMode(pdown,INPUT\_PULLUP);

pinMode(pleft,INPUT\_PULLUP);

pinMode(pright,INPUT\_PULLUP);

attachInterrupt(digitalPinToInterrupt(pup),up,FALLING);

attachInterrupt(digitalPinToInterrupt(pdown),down,FALLING);

attachInterrupt(digitalPinToInterrupt(pleft),left,FALLING);

attachInterrupt(digitalPinToInterrupt(pright),right,FALLING);

//起點

x[0]=2;

y[0]=1;

x[1]=1;

y[1]=1 ;

s=2;//蛇身長度

//設定記憶體

for (int i=0;i<s;i++){

bitSet(b[x[i]],y[i]);

}

//蛇頭位置

hx=x[0];

hy=y[0];

dir=4;//起始方向

//apple

ax=6;

ay=6;

bitSet(b[ax],ay);

randomSeed(analogRead(0));//亂數設定，請看全民自造與程式設計

}

void loop() {

start:

//輸出記憶體

for (int i=0;i<=100;i++){

for (int j=0;j<=7;j++){

PORTC=c[j];

PORTB=b[j];

delay(1);

}

}

//清除蛇身

for (int i=0;i<s;i++){

bitClear(b[x[i]],y[i]);

}

//依照按鍵改變蛇頭位置

if (dir==1){//上

hy=hy-1;

}else if(dir==2){//下

hy=hy+1;

}else if (dir==3){//左

hx=hx-1;

}else if (dir==4){//右

hx=hx+1;

}

//判斷是否撞牆

if (hx==-1 || hx==wid || hy==-1 ||hy==hei){

Serial.println("撞牆了");

delay(2000);

//清除蛇身

for (int i=0;i<s;i++){

bitClear(b[x[i]],y[i]);

}

//清除蘋果

bitClear(b[ax],ay);

//初始化

x[0]=2;

y[0]=1;

x[1]=1;

y[1]=1 ;

s=2;

for (int i=0;i<s;i++){

bitSet(b[x[i]],y[i]);

}

hx=x[0];

hy=y[0];

dir=4;

ax=6;

ay=6;

bitSet(b[ax],ay);

goto start;

}

//判斷是否自撞

for (byte i=1;i<s;i++){

if (hx==x[i] && hy==y[i]){

Serial.println("自撞了");

delay(2000);

//清除蛇身

for (int i=0;i<s;i++){

bitClear(b[x[i]],y[i]);

}

//清除蘋果

bitClear(b[ax],ay);

//初始化

x[0]=2;

y[0]=1;

x[1]=1;

y[1]=1 ;

s=2;

for (int i=0;i<s;i++){

bitSet(b[x[i]],y[i]);

}

hx=x[0];

hy=y[0];

dir=4;

ax=6;

ay=6;

bitSet(b[ax],ay);

goto start;

}

}

//判斷是否吃到蘋果

if(hx==ax && hy==ay){

Serial.println("吃到蘋果");

s=s+1;//蛇身長度加1

Serial.println(s);

bitClear(b[ax],ay);//清除蘋果

ax=random(0,8);

ay=random(0,8);

bitSet(b[ax],ay);//設定蘋果新位置

}

//將陣列的0放到1，1放到2

for (int i=s-1;i>=0;i--){

x[i+1]=x[i];

y[i+1]=y[i];

}

//新蛇頭的位置放到陣列的0

x[0]=hx;

y[0]=hy;

//重新設定記憶體

for (int i=0;i<s;i++){

bitSet(b[x[i]],y[i]);

}

}

//上

void up(){

if (dir==2)

return;

dir=1;

}

//下

void down(){

if (dir==1)

return;

dir=2;

}

//左

void left(){

if (dir==4)

return;

dir=3;

}

//右

void right(){

if (dir==3)

return;

dir=4;

}