LCD文字型顯示器基本實驗

下面的LCD文字型顯示器基本實驗，我們透過KeilC組譯我們完成編輯的程式碼，並且，透過燒錄程式(progisp)，將程式燒錄到單晶片。

A.完成硬體接線

B.建立新的專案並加入新的原始碼檔

後鍵入原始碼

C. 撰寫程式碼後組譯之

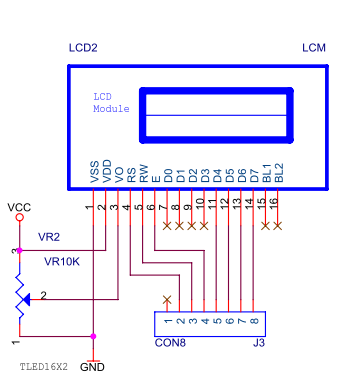
D. 將HEX檔燒錄至晶片

A.完成硬體接線

這次的實驗，將從P2埠輸出到文字型LCD顯示器

第一步， 我們要找出文字型LCD顯示器的跳線位置

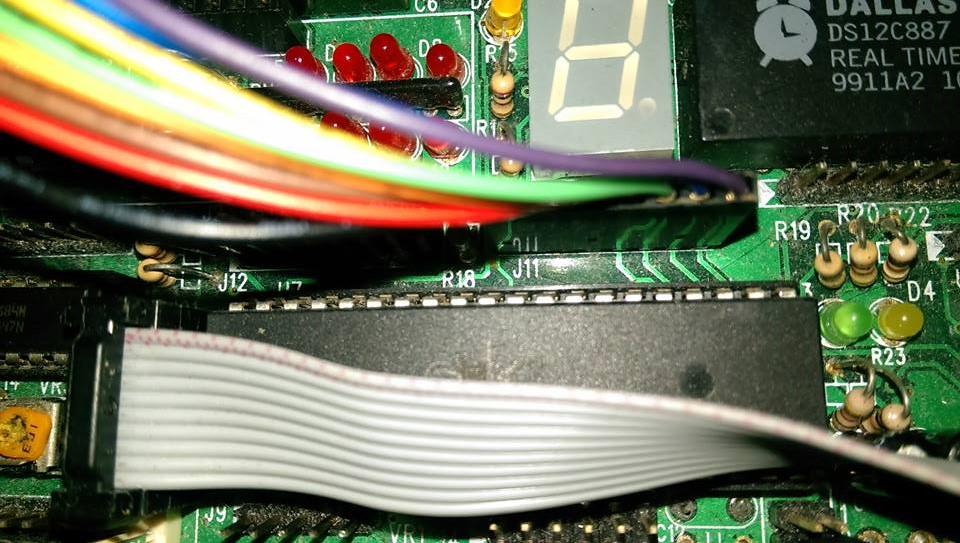
從下圖，我們可以的得知，文字型LCD顯示器跳線位置位於J3

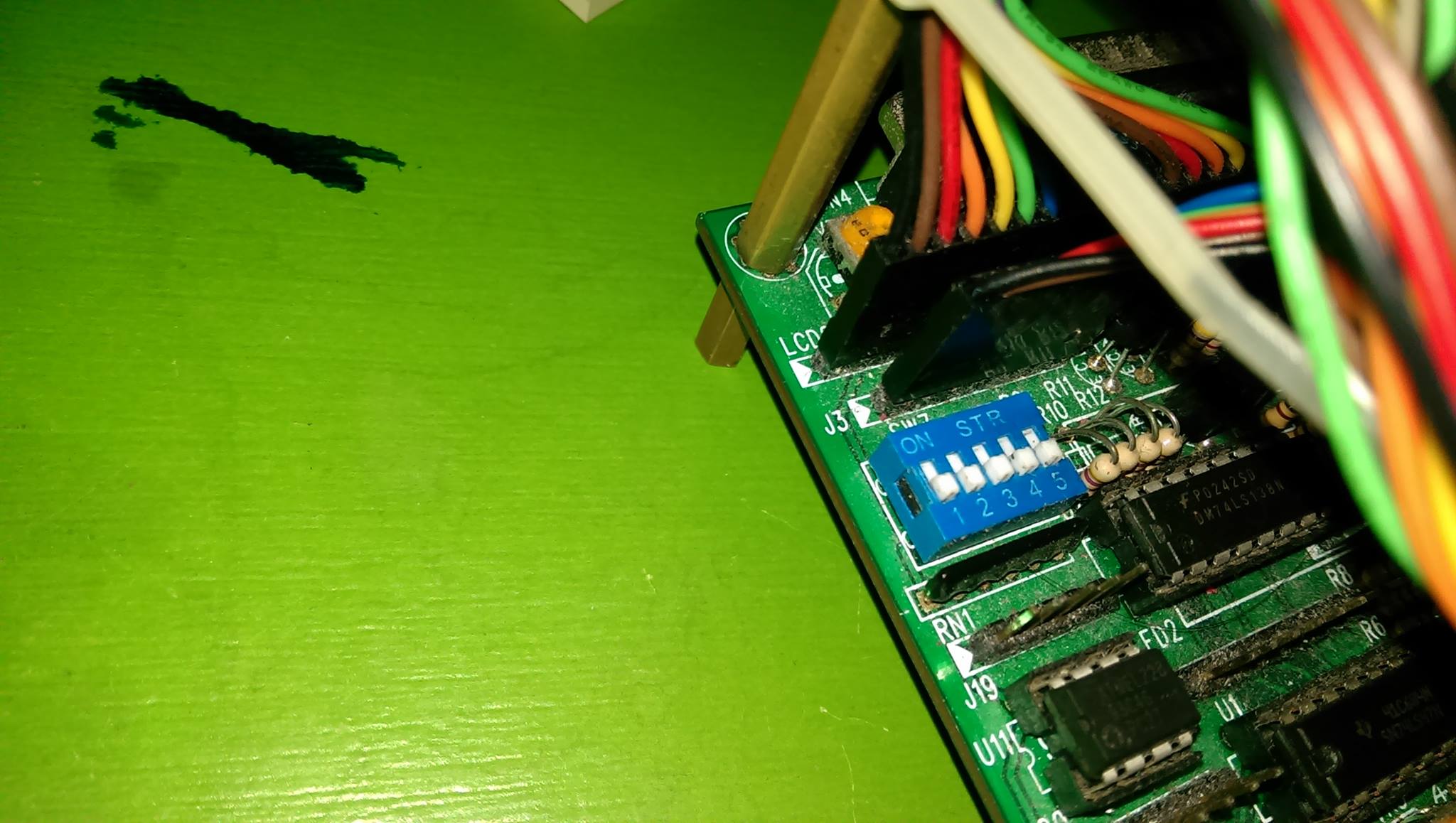


第二步，根據之前的實驗我們已經得知P2埠跳線位置位於J11

所以，我們將J3(文字型LCD顯示器)接到J11(P2埠)

特別注意，黑色的線都在左邊



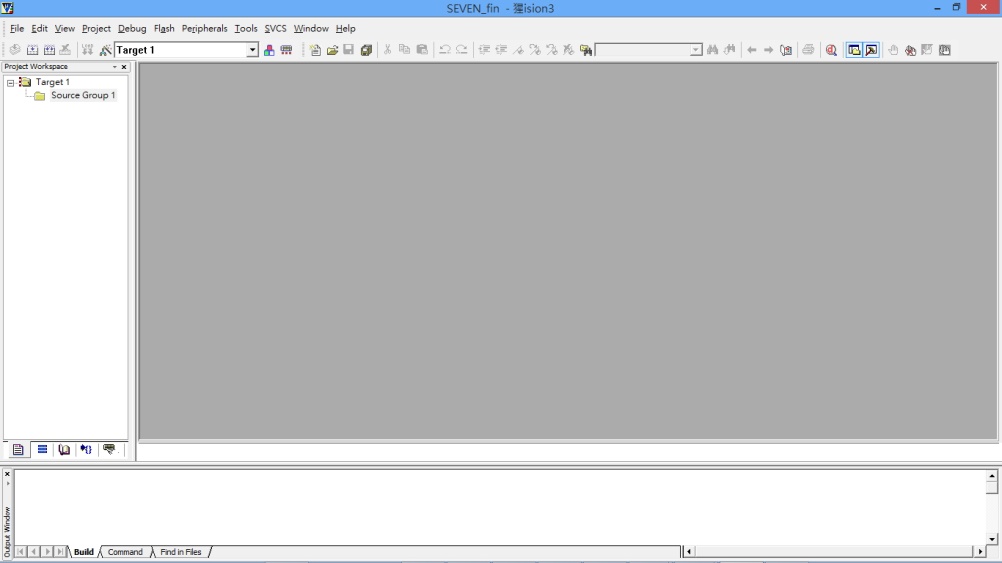


B. 建立新的專案並加入新的原始碼檔後鍵入原始碼

第一步，建立新的專案

這次我們將專案放置位置，指定放置於桌面

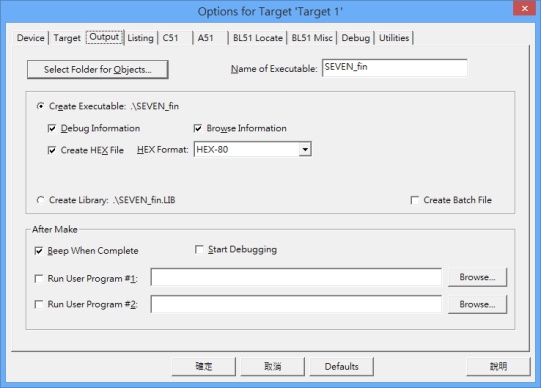
由於是指播開關與七段顯示器的實驗。所以，我們設專案名稱為LCD\_TEXT



第二步，設定專案，使組譯後將產生HEX檔

進到Flash Configure Flash Tools Output

將Create HEX files打勾



第三步，加入新的原始碼檔

這一步，我們要建立六個個原始檔，分別是:

LCD\_TEXT.ASM(主程式)

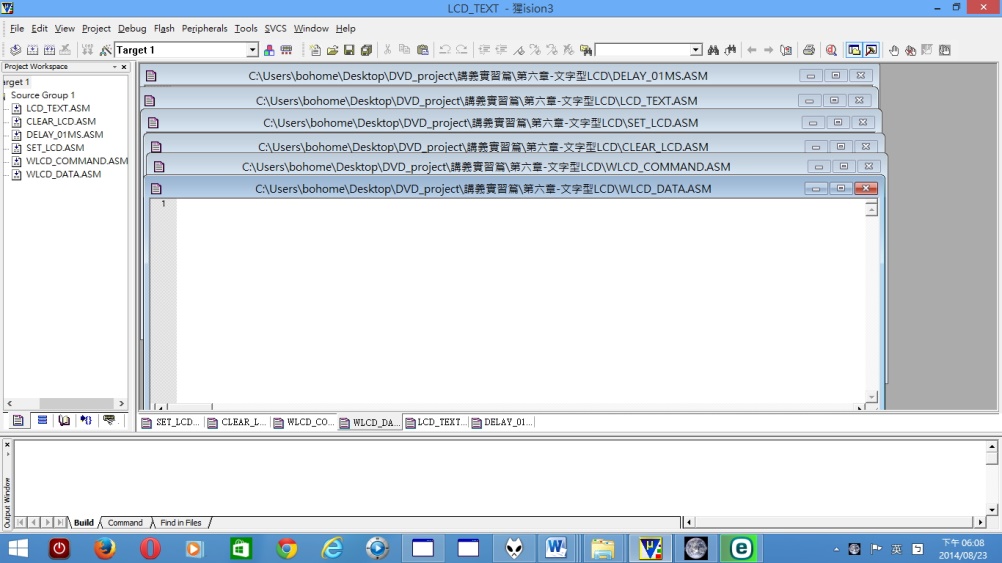
CLEAR\_LCD.ASM(清除螢幕)

DELAY\_01MS.ASM(DELAY 1ms)

SET\_LCD.ASM(LCD初始化)

WLCD\_COMMAND.ASM(輸入LCD命令)

WLCD\_DATA.ASM(輸入LCD資料)



C. 撰寫程式碼後組譯之

首先，先將下列的程式碼鍵入到主程式(LCD\_TEXT.ASM)

**EXTRN CODE**

**(CLEAR\_LCD,SET\_LCD,WLCD\_COMMAND,WLCD\_DATA)**

**ORG 00H**

**JMP START**

**//呼叫SET\_LCD，設置LCD**

**START: CALL SET\_LCD**

**//呼叫 CLEAR\_LCD，清除LCD**

**CALL CLEAR\_LCD**

**//設置LCD的位置為80H，第一行第一個位置**

**MOV R2,#80H**

**//呼叫WLCD\_COMMAND，設置LCD位置**

**CALL WLCD\_COMMAND**

**//設置LCD ASCII的41H位置資料"A"**

**MOV R2,#41H**

**//呼叫WLCD\_DATA，顯示資料於LCD上**

**CALL WLCD\_DATA**

**//調整時序**

**CALL DELAY\_01MS**

**//設置LCD的位置為C0H，第二行第一個位置**

**MOV R2,#0C0H**

**//呼叫WLCD\_COMMAND，設置LCD位置**

**CALL WLCD\_COMMAND**

**//設置LCD ASCII的46H位置資料"F"**

**MOV R2,#46H**

**//呼叫WLCD\_DATA，顯示資料於LCD上**

**CALL WLCD\_DATA**

**END**

第二步，鍵入第一個副程式(CLEAR\_LCD.ASM)

**;------------------------------------------------------------------------**

**;CLEAR\_LCD**

**;(1) FUNCTION: DISPLAY CLEAR / CURSOR HOME**

**;(2) INPUT : None**

**;(3) OUTPUT : None**

**;(4) USE REG. : R2,R3,R4**

**;-------------------------------------------------------------------------**

**EXTRN CODE(WLCD\_COMMAND)**

**PUBLIC CLEAR\_LCD**

**ROM SEGMENT CODE**

**RSEG ROM**

**CLEAR\_LCD:**

**MOV R2,#00000001B**

**CALL WLCD\_COMMAND**

**MOV R2,#00000010B**

**CALL WLCD\_COMMAND**

**RET**

**END**

第三步，鍵入第二個副程式(DELAY\_01MS.ASM)

**;-------------------------------------------------------**

**; DELAY\_01MS**

**;(1) FUNCTION: Delay R4\*0.1ms time**

**;(2) INPUT : R4**

**;(3) OUTPUT : None?**

**;(4) USE REG.: R3,R4**

**;--------------------------------------------------------**

**PUBLIC DELAY\_01MS**

**ROM SEGMENT CODE**

**RSEG ROM**

**DELAY\_01MS:**

**MOV R3,#1**

**RET**

**END**

第四步，鍵入第三個副程式(SET\_LCD.ASM)

**;-----------------------------------------------**

**;SET\_LCD**

**;(1) FUNCTION: Initialize LCD module**

**; \* FUNCTION SET: DL=0, N=1**

**; \* DISPLAY ON/OFF: D=1, C=0, B=0**

**; \* ENTRY MODE SET: I/D=1, S=0**

**;(2) INPUT : None**

**;(3) OUTPUT : None**

**;(4) USE REG. : R2,R3,R4**

**;------------------------------------------------**

**EXTRN CODE(WLCD\_COMMAND)**

**EXTRN CODE(DELAY\_01MS)**

**PUBLIC SET\_LCD**

**ROM SEGMENT CODE**

**RSEG ROM**

**LCD\_BUS EQU P2**

**LCD\_E EQU P2.3**

**LCD\_RW EQU P2.2**

**LCD\_RS EQU P2.1**

**SET\_LCD: MOV P0,#0FFH**

**MOV R4,#150**

**CALL DELAY\_01MS**

**;(1)**

**MOV R2,#00111000B**

**CALL SET\_COMMAND**

**MOV R4,#41**

**CALL DELAY\_01MS**

**;(2)**

**MOV R2,#00111000B**

**CALL SET\_COMMAND**

**MOV R4,#10**

**CALL DELAY\_01MS**

**;(3)**

**MOV R2,#00111000B**

**CALL SET\_COMMAND**

**;(4)**

**MOV R2,#00101000B**

**CALL SET\_COMMAND**

**;(5)**

**MOV R2,#00101000B**

**CALL SET\_COMMAND**

**;(6)**

**MOV R2,#10001000B**

**CALL SET\_COMMAND**

**;(7)**

**MOV R2,#00001000B**

**CALL SET\_COMMAND**

**;(8)**

**MOV R2,#10001000B**

**CALL SET\_COMMAND**

**;(9)**

**MOV R2,#00001000B**

**CALL SET\_COMMAND**

**;(10)**

**MOV R2,#00011000B**

**CALL SET\_COMMAND**

**;(11)**

**MOV R2,#00001000B**

**CALL SET\_COMMAND**

**;(12)**

**MOV R2,#01101000B**

**CALL SET\_COMMAND**

**;(13)**

**MOV R2,#00001000B**

**CALL SET\_COMMAND**

**;(14)**

**MOV R2,#11001000B**

**CALL SET\_COMMAND**

**RET**

**SET\_COMMAND:**

**CLR LCD\_RW**

**CLR LCD\_RS**

**SETB LCD\_E**

**MOV LCD\_BUS,R2**

**CLR LCD\_E**

**SETB LCD\_RS**

**SETB LCD\_RW**

**MOV R4,#20**

**CALL DELAY\_01MS**

**RET**

**END**

第五步，鍵入第四個副程式(WLCD\_COMMAND.ASM)

**;-------------------------------------------------------------**

**;WLCD\_COMMAND**

**;(1) FUNCTION: Write LCD command to IR**

**; \* E=1, R/W=0, RS=0**

**; \* Command -> LCD\_BUS**

**; \* E=0, R/W=1, RS=1**

**;(2) INPUT : R2 .... 82H**

**;(3) OUTPUT : None**

**;(4) USE REG.: R2,R3,R4**

**;---------------------------------------------------------------**

**EXTRN CODE(DELAY\_01MS)**

**PUBLIC WLCD\_COMMAND**

**ROM SEGMENT CODE**

**LCD\_BUS EQU P2**

**LCD\_E EQU P2.3**

**LCD\_RW EQU P2.2**

**LCD\_RS EQU P2.1**

**RSEG ROM**

**WLCD\_COMMAND:**

**; High 4 bits**

**CLR LCD\_RW**

**CLR LCD\_RS**

**SETB LCD\_E**

**MOV A,R2**

**ANL A,#0F0H ;A=80H**

**ADD A,#08H ;A=88H**

**MOV LCD\_BUS,A**

**CLR LCD\_E**

**SETB LCD\_RS**

**SETB LCD\_RW**

**MOV R4,#20**

**CALL DELAY\_01MS**

**; Low 4 bits**

**CLR LCD\_RW**

**CLR LCD\_RS**

**SETB LCD\_E**

**MOV A,R2 ;82H**

**ANL A,#0FH ;A=02H**

**SWAP A ;A=20H**

**ADD A,#08H ;A=28H**

**MOV LCD\_BUS,A**

**CLR LCD\_E**

**SETB LCD\_RS**

**SETB LCD\_RW**

**MOV R4,#20**

**CALL DELAY\_01MS**

**RET**

**END**

第六步，鍵入第五個副程式(WLCD\_DATA.ASM)

**;-----------------------------------------------**

**;WLCD\_DATA**

**;(1) FUNCTION: Write LCD DATA to DR**

**; \* E=1, R/W=0, RS=1**

**; \* Data -> LCD\_BUS**

**; \* E=0, R/W=1, RS=1**

**;(2) INPUT : R2 : 41H**

**;(3) OUTPUT : None**

**;(4) USE REG. : R2,R3,R4**

**;------------------------------------------------**

**EXTRN CODE(DELAY\_01MS)**

**PUBLIC WLCD\_DATA**

**ROM SEGMENT CODE**

**LCD\_BUS EQU P2**

**LCD\_E EQU P2.3**

**LCD\_RW EQU P2.2**

**LCD\_RS EQU P2.1**

**RSEG ROM**

**WLCD\_DATA:**

**; High 4 bits**

**CLR LCD\_RW**

**SETB LCD\_RS**

**SETB LCD\_E**

**MOV A,R2**

**ANL A,#0F0H**

**ADD A,#0AH**

**MOV LCD\_BUS,A**

**CLR LCD\_E**

**SETB LCD\_RS**

**SETB LCD\_RW**

**MOV R4,#20**

**CALL DELAY\_01MS**

**; Low 4 bits**

**CLR LCD\_RW**

**SETB LCD\_RS**

**SETB LCD\_E**

**MOV A,R2**

**ANL A,#0FH**

**SWAP A**

**ADD A,#0AH**

**MOV LCD\_BUS,A**

**CLR LCD\_E**

**SETB LCD\_RS**

**SETB LCD\_RW**

**MOV R4,#20**

**CALL DELAY\_01MS**

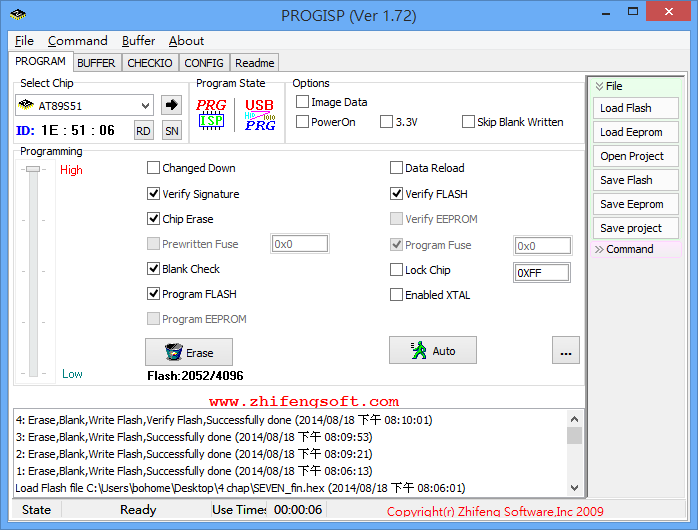
**RET**

**END**

D. 將HEX檔燒錄至晶片

第二步，燒錄器根據下圖設定後，按下下圖中的AUTO鍵

在這裡，我們將速度調到High(高速)。如果，跳出錯誤訊息，可以將速度調到中間，再試一次。



做到這裡，恭喜你了，LCD文字型顯示器實驗已經完成了

**觀念提醒:**

**A. 文字LCD 顯示器腳位與PORT2埠對照**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 腳位 | P2.7 | P2.6 | P2.5 | P2.4 | P2.3 | P2.2 | P2.1 | P2.0 |
| 對照開發版PIN腳 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
| 對照開發版LCD顯示器腳位 | D7 | D6 | D5 | D4 | E | RW | RS | X |

**B. LCD文字顯示器使用方式:**

第一步: 初始化設定

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 指令 | | | | | | | | 指令說明 | 執行次數 |
| 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 功能設定(先)  註a | 3 |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 功能設定(後) | 2 |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 顯示OFF | 1 |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |  | 1 |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 顯示OFF | 1 |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |  | 1 |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 清除顯示 | 1 |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |  | 1 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 輸入模式設定 | 1 |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |  | 1 |
| 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 顯示開關設定 | 1 |

PS: 因為時序的關係，每次設定前 要先暫停 0.1MS(delay 0.1ms)，在進行設定

第二步: 設定位置

80H 81H…………………..8FH LCD文字顯示器上排

C0H C1H…………………..CFH LCD文字顯示器下排

例如:

//設置LCD的位置為80H，第一行第一個位置

MOV R2,#80H

//呼叫WLCD\_COMMAND，設置LCD位置

CALL WLCD\_COMMAND

3. 編碼: FIG1

編碼範例:

|  |  |  |
| --- | --- | --- |
| 數字 | 編碼(B) | 編碼(H) |
| 1 | 00110001 | 31H |
| 2 | 00110010 | 32H |
| 3 | 00110011 | 33H |
| 4 | 00110100 | 34H |
| 5 | 00110101 | 35H |
| 6 | 00110110 | 36H |
| 7 | 00110111 | 37H |
| 8 | 00111000 | 38H |
| 9 | 00111001 | 39H |
| 0 | 00110000 | 30H |

**Q&A:**

**Q1.為什麼每次設定(WLCD\_COMMAND)要送兩次碼?**

**ANS1.因為我們是使用四位元的方式去操作文字型LCD顯示器。**

**所以，必須先送高四位元給文字型LCD顯示器，再送低四位元給文字型LCD顯示器。**

LCD文字型顯示器造字實驗

經過LCD文字型顯示器基本實驗，我們已經知道操作LCD文字型顯示器，方法與過程。

由於，這次實驗跟LCD文字型顯示器基本實驗的程式碼，除了，主程式與查表程式碼外，都雷同，是故本篇就以主程式與查表程式碼來論述。

A. 本實驗將用到以下七個原始碼檔

LCD\_TEXT.ASM(主程式)

FLOOK\_TABLE(查表)

CLEAR\_LCD.ASM(清除螢幕)

DELAY\_01MS.ASM(DELAY 1ms)

SET\_LCD.ASM(LCD初始化)

WLCD\_COMMAND.ASM(輸入LCD命令)

WLCD\_DATA.ASM(輸入LCD資料)

B. 將以下的原始碼，鍵入到主程式中(LCD\_TEXT.ASM)

**EXTRN CODE(CLEAR\_LCD,SET\_LCD,WLCD\_COMMAND,WLCD\_DATA,DELAY\_01MS,FLOOK\_TABLE)**

**ORG 00H**

**JMP START**

**START: CALL SET\_LCD**

**CALL CLEAR\_LCD**

**MOV R0,#0**

**MOV DPTR,#TABLE**

**//將文字寫入CGRAM**

**MOV R2,#01000000B**

**CALL WLCD\_COMMAND**

**LOOP: MOV A,R0**

**MOV R1,A**

**CALL FLOOK\_TABLE**

**CALL WLCD\_DATA**

**INC R0**

**//8行x2筆=16**

**CJNE R0,#16,LOOP**

**//寫入到LCD顯示器**第一**行**第一個位置

**MOV R2,#080H**

**CALL WLCD\_COMMAND**

**//呼叫CGRAM第0個位址的資料**

**MOV R2,#000H**

**CALL WLCD\_DATA**

**//寫入到LCD顯示器的**第一**行**第二個位置

**MOV R2,#081H**

**CALL WLCD\_COMMAND**

**//呼叫CGRAM第1個位址的資料**

**MOV R2,#001H**

**CALL WLCD\_DATA**

**JMP $**

**TABLE: DB 00AH,00AH,011H,004H,008H,012H,01FH,000H ;公**

**DB 00EH,000H,01FH,00AH,00AH,00AH,011H,000H ;元**

**END**

C. 將以下的原始碼，鍵入到查表程式碼中 (**FLOOK\_TABLE**)

**;順向查表 : index -> code**

**;輸入 : index(R1),table\_name(DPTR)**

**;輸出 : code(R2)**

**;使用 : R1,R2**

**;查表在R1,結果放R2**

**;----------------------------------------------- --------------------------------**

**PUBLIC FLOOK\_TABLE ;模組名稱**

**ROM SEGMENT CODE ;共享ROM空間**

**RSEG ROM ;選取可重置的 ROM空間**

**FLOOK\_TABLE: ;模組名稱**

**MOV A,R1 ;將R1位置的值放在A**

**MOVC A,@A+DPTR ;將表格內第A(R1)個的數字放於A**

**MOV R2,A ;將A的值放在R2**

**RET ;從模組返回**

**END ;程式結束**

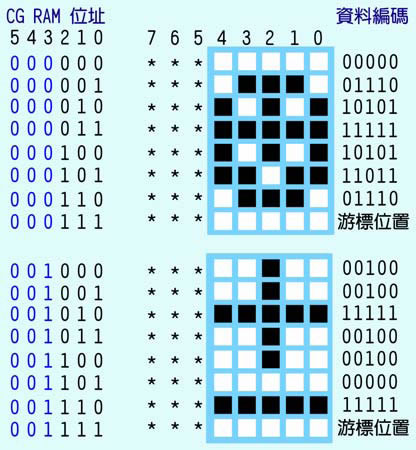
**觀念提醒:**

**A. 如何造字??**

CGRAM在造字的時候一共保留6個位元

後三個位置(CGRAM位址2,1,0)則表示第幾列，由CGRAM內部自行遞增

前三個(CGRAM位址5,4,3)位置表示第幾個字，由使用者鍵入，也可以放入000讓CGRAM自行遞增



編碼範例: 田

|  |  |
| --- | --- |
| CGRAM位址 | 資料位元 |
| 000 | 00000B |
| 000 | 01110B |
| 000 | 10101B |
| 000 | 11111B |
| 000 | 10101B |
| 000 | 11011B |
| 000 | 01110B |
| 000 | 00000B |

LCD文字型顯示器字串實驗

經過LCD文字型顯示器造字實驗，我們已經知道操作LCD文字型顯示器，方法與過程。

由於，這次實驗跟LCD文字型顯示器造字實驗的程式碼，除了，主程式程式碼外，都雷同，是故本篇就以主程式程式碼來論述。

LCD\_TEXT.ASM(主程式)

FLOOK\_TABLE(查表)

CLEAR\_LCD.ASM(清除螢幕)

DELAY\_01MS.ASM(DELAY 1ms)

SET\_LCD.ASM(LCD初始化)

WLCD\_COMMAND.ASM(輸入LCD命令)

WLCD\_DATA.ASM(輸入LCD資料)

A.將以下的原始碼，鍵入到主程式中(LCD\_TEXT.ASM)

EXTRN CODE(CLEAR\_LCD,SET\_LCD,WLCD\_COMMAND,WLCD\_DATA,DELAY\_01MS,FLOOK\_TABLE)

ORG 00H

JMP START

START: CALL SET\_LCD

CALL CLEAR\_LCD

MOV R1,#0

**//設定初始位置在第一行第一個位置**

MOV R2,#080H

CALL WLCD\_COMMAND

MOV DPTR,#TABLE01

LOOP: CALL FLOOK\_TABLE

CALL WLCD\_DATA

INC R1

CJNE R1,#20,LOOP

MOV R1,#0

**//設定初始位置在第二行第一個位置**

MOV R2,#0C0H

CALL WLCD\_COMMAND

MOV DPTR,#TABLE02

LOOP1: CALL FLOOK\_TABLE

CALL WLCD\_DATA

INC R1

CJNE R1,#20,LOOP1

JMP $

TABLE01: DB "MCUEE is famous "

TABLE02: DB "department

END

**觀念提醒:**

**A.如何顯示字串在LCD文字型顯示器上??**

先設定字串的初始位置，由於AC會自行遞增，所以接下來，將欲顯示的字一個一個透過查表法丟到DDRAM裡面即可。

**特別注意，字串若未滿16的字(一列)，請用空白補滿至16個字**