

# 儀表電子能力本位訓練教材 認識與操作 EM78 模擬器

編號：PEN-IMT1009

編著者：陳宏昇

審稿者：吳經文、董勝源

主辦單位：行政院勞工委員會職業訓練局

研製單位：中華民國職業訓練研究發展中心

印製日期：九十年十二月

## 單元 PEN-IMT1009 學習指引

當你學習本單元前，你必須先瞭解下列內容：單晶片 EM78 晶片的基本架構及 EM78 的語法應用，假如自認無法勝任，則請按下列之指示進行學習：

- (1)你全部無法勝任上列之工作，請將本教材放回原位，並取出編號 PEN-IMT1006 教材開始學習，或請教你的老師。
- (2)如果你已會單晶片 EM78 的基本架構，而對 EM78 語法不熟，則請從編號 PEN-IMT1007 教材開始學，或去請教你的老師。

## 引言

本單元是介紹如何使用 EM78 的 ICE(模擬器)，一般完整的 ICE 必須具備有下列的特徵：

1. 能模擬 CPU 或單晶片微控器的動作。
2. 提供與 CPU 或單晶片微控器相同的輸出、輸入能力。
3. 提供記憶體模擬單晶片微控器的內部記憶體或模擬目標系統上的記憶體。(目標系統：target system，為要開發的微電腦系統)
4. 具有將匯流排信號儲存的能力。
5. 具有單步執行指令的能力。

ICE 配合個人電腦即相關軟體後，即成所謂的發展系統，對為微計算機系統或微控器電路的開發者而言，一套良好的發展系統可縮短開發的時間。

## 定義

ICE (In Circuit Emulator) 稱為實體線上模擬器

## 學習目標

- 一、按照操作手冊，你可以了解 EM78 ICE 模擬器的功能及正確的操作過程。

## 學習活動

目前市面上談到 EM78 ICE 模擬器參考書籍相當多種，以下列出一些常用的書籍可供參考：

- |                         |      |         |
|-------------------------|------|---------|
| 1. EM78X56 入門與實作        | 宏友出版 | 詹耀仁 胡建偉 |
| 2. 單晶片微控制 EM78X56 原理與實作 | 宏友出版 | 王宜楷     |
| 3. EM78447 入門實作         | 宏友出版 | 鄧錦城     |

本教材第一個學習目標：

了解 EM78 ICE 模擬器的功能及正確的操作過程。

E8-ICE 為義隆公司針對其單晶片微控器所設計的 ICE，這套 ICE 功能齊備且價格便宜，只要搭配上個人電腦即成一套 EM78 微控器的發展系統 (development system)。

設備需求與安裝  
面板使用說明

要組成 EM78 微控器的發展系統必須有下列的配備：

1. EGA 以上的顯示器(monitor)。
2. 1M bytes RAM。
3. 386 以上的 IBM PC 或與其相容的個人電腦。
4. E8-ICE 及其驅動程式。
5. 16.7V/350mA 的直流電源轉換器(power adaptor)。

E8-ICE 軟體安裝：

先建立一新目錄(新資料夾)，例如 C:\E8-ICE，然後將 E8-ICE 內所附的磁片檔案複製到該目錄(資料夾)內。

## 面板說明

E8-ICE 的面板上有一個 RESET、一個電源開關、三個 LED 及一個 12 孔的插座，這些元件的作用如下：

1. RESET 按鈕：用來對 target board(目標電路板)做重置動作。
2. POWER SWITCH：電源開關，在執行 E8-ICE 前，要先把電源開啓，否則會因偵測不到硬體而出現錯誤訊息。
3. 三個 LED：
  - 紅色為電源(POWER)指示 LED，用來指示電源是否開啓。紅色 LED 亮表示電源已開啓。
  - 綠色為執行程式(RUN)指示 LED，用來指示目前是否正在執行程式。綠色 LED 亮表示目前正在執行程式。
  - 黃色為目標電路板(TARGET)指示 LED，用來指示目標電路板電源是否由 E8-ICE 提供。黃色 LED 亮表示目標電路板電源是由 E8-ICE 提供。
4. 12 孔插座：這個插座是爲了不同的振盪方式而設計的。由於 EM78 系列微控器的振盪方式有二種：一爲 crystal type，另一爲 RC type。這 12 孔插座就是讓使用者自行決定要用哪一種方式。

在執行 E8-ICE 前，要先選擇適當的元件，將這個部分的電路接好，否則會出現錯誤訊息。

## EM78 ICE 驅動程式

執行檔案 ICE456.EXE，則螢幕上出現的畫面如下：

The screenshot shows the E8-ICE software interface. The main window displays a program listing for 'SAMPLE5.C'. The program includes several register definitions and constants. To the right, a 'Regs and I/Os' window shows the current values of various registers. At the bottom, a 'Bank' window displays memory addresses and their corresponding values.

```

File Modify Run Compile Debug Options
SAMPLE5.C
THIS PROGRAM IS E8-ICE-EM78456 SAMPLE PROG
DATE 5,20,1995 by K.T.M

INDIRT == 0x0 : Indirect Register def
TCC == 0x1 : Timer/Counter Register
PC == 0x2 : Program Counter define
PSW == 0x3 : Program Status Word define
RSR == 0x4 : RAM Select register define
PORT_5 == 0x5 : Port 5 define
PORT_6 == 0x6 : Port 6 define
INTF == 0x7 : Interrupt status register
FLAG == 0x10 : SYSTEM FLAG
TEMP1 == 0x11 : TEMPloare data buffer
TEMP2 == 0x12
TEMP3 == 0x13

R1/TCC:00 R10:FF
R2/PC :0FFF R11:0F
R3:00011000 R12:BF
76543210 R13:FF
R4:00000000 R14:F7
C5:0F P5:0F R15:FF
C6:FF P6:FF R16:FF
R17:FD
R18:FF
R19:FF
R1A:FF
R1B:FF
R1C:7F
R1D:FF
R1E:FD
R1F:FF

CB:FF
CC:FF
CD:FF
CE:B7
CF:B6 RF:7

Bank
R2X: BF FD FF FF EF FF FB FF DB FF FF FF F7 FF 9F FF
R3X: FF FD FF FF F7 FF FD FF FF EF F7 FF FD FF FF F6

F1-Opn F2-Log F3-Dmp F4-Goto F5-Run F6-Rst F7-Trc F8-Stp F9-Compile F10-F_Run

```

若要退出 E8-ICE 驅動程式，則利用[Esc]按鍵切換到系統功能表(system menu)，然後再用箭頭按鍵(↑, ↓, ←, →)選擇 File|Quit 或直接按 Alt+X，則結束 E8-ICE 驅動程式。

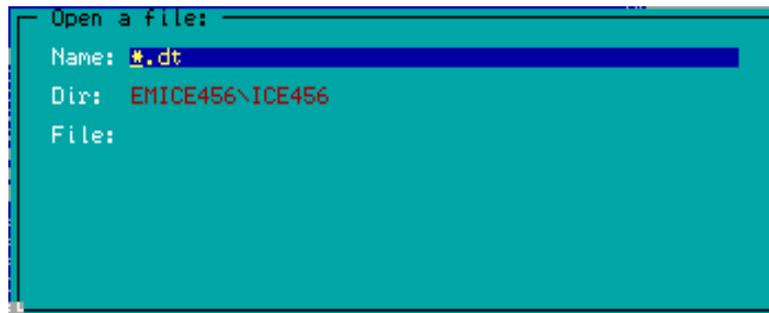
### ESC 按鍵作用說明

1. 程式視窗與系統功能表(system menu) 間的切換。
2. 取消下拉功能視窗，回到系統功能表(system menu)。
3. 當 E8-ICE 無法反應個人電腦送來的讀/寫命令時，執行程式會進入迴圈 (waiting loop)中，導致 E8-ICE 無法接受其他命令，此時可按[Esc]按鍵結束等待迴圈(waiting loop)。
4. 在程式執行中，按[Esc]按鍵可終止程式的執行。

### EM78 ICE 功能表說明 ESC 按鍵作用說明

1.File 功能：在這功能下有 6 個子功能，分別說明如下：

- 1). Open (F1)：開啓原始程式檔案，該檔案的副檔名為.DT。執行此功能時會出現如下的畫面：

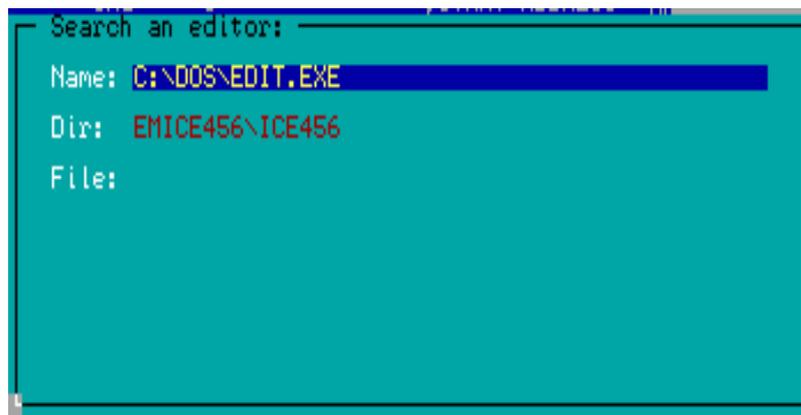


此時使用者若知道檔案名稱，則可直接鍵入原始檔案名稱，然後按↵載入檔案。若不知道檔案名稱，則直接按↵，畫面上會出現如下的視窗：



此時可用箭頭鍵(←, →, ↑, ↓)選擇原始檔案，然後按↵。

- 2). Dump to ice (F3)：將目前原始程式的機械碼傳送到 ICE 中。
- 3). Trace log (F2)：將 Trace Buffer 的內容顯示在螢幕上。
- 4). Path for editor：設定文書編輯軟體(editor)所在的路徑。執行此功能時會出現如下的畫面：



接著在 Name 方塊內填入所要用的文書編輯軟體檔名及路徑，然後按↵，即完成設定的工作。

- 5). Dos Shell：暫時離開 E8-ICE 驅動程式，操作權交給 DOS。在 DOS 中執行 EXIT 即可回到 E8-ICE 驅動程式。
  - 6). Window (Alt-W)：切換程式視窗的大小。
  - 7). Quit (Alt-X)：結束 E8-ICE 驅動程式。
2. Modify 功能：在這功能下有 3 個子功能，分別說明如下：
- 1). Program (Alt-E)：執行文書編輯程式，並自動開啓目前的原始檔案以供修改。修改完畢後，先存檔再退出文書編輯程式，則會直接回到 E8-ICE 驅動程式。此時程式視窗上的程式以及 E8-ICE 內所存的機械碼仍為修改前之舊程式，只要執行 Compile|Program 即可將二者更新。
  - 2). I/O Control Registers：修改 IOC5~IOC7 的內容，也就是設定 I/O port 為輸出模式或輸入模式。其輸入格式如下：

0xRR 0xVV, 0xRR 0xVV, ...↵

其中 0xRR 代表 I/O control register(0x05~0x07)。

0xVV 代表所要修改的數值。

例如：要將 port 6 設為輸入模式，則鍵入方式為：

0x06 0xff↵

執行本功能時會出現如下的畫面：



3). **Registers**：修改暫存器 R1~R3F 的內容。執行本功能後所出現的畫面與前一個功能相同，輸入的格式也與前一個功能相同。

例如：要將 R20 的內容設為 0x3A，則鍵入方式為：

0x20 0x3A↵

3. **Run 功能**：在這功能下有 7 個子功能，分別說明如下：

1). **Free run (F10)**：執行被載入的程式，也就是執行目前在畫面上的程式，此時設定的中斷點(break point)不會發生作用。

2). **Reset (F6)**：做重置的動作，程式計數器(PC)會被歸零。

3). **Trace into (F7)**：單步執行，執行 CALL 指令時，會進入其所呼叫的副程式單步執行。

4). **Step over (F8)**：單步執行，並將副程式視為單步，也就是執行 CALL 指令時，直接將其所呼叫的副程式一次執行完畢。

5). **Go to cursor (F4)**：由目前程式計數器(PC)所指到的位址開始執行程式，一直執行到游標所在的位址為止，游標所指的指令執行完畢後，停止執行指令的工作。

6). **Run form**：在這功能下有 4 個子功能，分別說明如下：

(1). **Initial with 8K\_step log (F5)**：

由位址 0x00 開始行指令，一直執行到中斷點(break point)條件成立為止。執行過的指令會被記錄在 trace buffer 中，最多可記錄最後被執行的 8K 個指令，這些記錄下來的指令可用 File|Trace log (F2)功能來觀察，其顯示的順序為最後被執行的指令顯示在最前面。

(2). **Current PC with 8K\_step log (Alt\_F5)**：

由目前程式計數器(PC)所指到的位址開始行指令，一直執行到中斷點(break point)條件成立為止。執行過的指令會被記錄在 trace buffer 中，最多記錄 8K 個指令。

(3). **Initial with 4K\_4K step log**：

由位址 0x00 開始行指令，一直執行到中斷點(break point)條件成立之後，再執行 4K 個指令，然後停止執行。執行過的指令會被記錄在 trace buffer 中，最多記錄 8K 個指令。其中 4K 為中斷點條件成立之後所執行的指令，其他則為中斷點條件成立之前所執行的指令。

(4). Current PC with 4K\_4K step log :

由目前程式計數器(PC)所指到的位址開始行指令，一直執行到中斷點(break point)條件成立之後，再繼續執行 4K 個指令，然後停止執行。執行過的指令會被記錄在 trace buffer 中，最多記錄 8K 個指令。其中 4K 為中斷點條件成立之後所執行的指令，其他則為中斷點條件成立之前所執行的指令。

7). Trace back (Alt\_T)：由目前的程式計數器(PC)所指到的位址開始往回追蹤 (trace)指令的執行過程，最多只能往回追蹤 8K 個指令。

4. Compile 功能：在這功能下有 2 個子功能，分別說明如下：

- 1). Program (F9)：對程式視窗內的原始程式做組譯的動作，組譯完成後會自動更新程式視窗及將機械碼載入 ICE 中。
- 2). PiggyBack (Alt\_F9)：

由於目前 EM78X56 系列的單晶片微控器只有 OTP type 可讓使用者自行燒錄程式，但 OTP type 只能燒錄一次，對正在開發的系統而言，仍有不方便的地方。因此義隆公司有一 piggy-back 電路板，這個電路板的作用與 EM78X56 單晶片微控器相同，不過它的程式記憶體是用普通的 EPROM(27C64)，因此使用者可重複燒寫程式。由於 EM78X56 單晶片微控器程式記憶體的長度為 13-bit，而普通的 EPROM 的長度為 8-bit，所以在 piggy-back 電路板上有二個 EPROM。

ICE 上 PiggyBack (Alt\_F9)的功能為將程式視窗內的原始程式做組譯的動作，組譯之後產生可燒錄至 piggy-back 電路板 EPROM 的機械碼檔案。由於 piggy-back 電路板上有二個 EPROM，所以用本功能所產生的機械碼檔案有二個，其副檔名分別為：.lo 和.hi。使用者只要分別將這二個檔案燒錄至二個 EPROM，然後插到 piggy-back 電路板上，則此 piggy-back 電路板就可插到 target board 上執行程式。

5. Debug 功能：在這功能下有 5 個子功能，分別說明如下：

- 1). Breakpoint View (Alt\_V)：檢視中斷點(breakpoint)的使用記錄。使用者在開發過程中所設定的中斷點(breakpoint)會被記錄下來，利用本功能可檢視中斷點記錄，也就是檢視使用記錄。除此之外也可由中斷點的位址表列中選擇其中一個來當目前使用的中斷點。

執行本功能後會出現如下的畫面：

```

Breakpoint View
Current Breakpoint:
A: 0x005 20

Breakpoint List:

< Note >: Function Key List:
1. TAB for more breakpoint items,
2. Line editor,
3. PageDown / PageUp,
4. DownArrow / UpArrow, and
5. Save file automatically.

Breakpoint Item: 0

```

其中 **Current Breakpoint:** 下所顯示的 **A: 0x005 20** 代表目前中斷點設在位址 **0x005**。**20** 代表通過次數，也就是執行中斷點的次數，即位址 **0x005** 必須被執行 **20** 次才會停止在中斷點位址。

2). **Breakpoint At (Alt\_A)**：將某一指定位址設為中斷點，最多可設 63 個中斷點，每個中斷點的通過次數(**pass count**)最多為 255 次。

中斷點的設定語法如下：

```
0xAAAA 0xPP, 0xAAAA 0xPP, ...
```

其中 **AAAA** 代表 16 進位的位址值，**PP** 為 16 進位的次數。例如要將位址 **0x0005** 設為中斷點，而通過次數為 **0x20** 次，則輸入方式如下：

```
0x0005 0x20
```

若一次要設多個中斷點，例如要將位址 **0x0005** 設為中斷點、通過次數為 **0x20** 次，位址 **0x0006** 設為中斷點、通過次數為 **0x10** 次，則輸入方式如下：

```
0x0005 0x20, 0x0006 0x10
```

輸入完畢按一下 ↵ 鍵，則完成中斷點設定。

中斷點的設定也可使用 10 進位，例如要將位址 **0x000A** 設為中斷點、通過次數為 10 次，則輸入方式如下：

```
0010 10
```

中斷點設定中通過次數(**pass count**)的作用如下：

當程式執行時，只要執行到中斷點位址，則所設定的通過次數值就會被減 1，一直減到變成 0 時，程式停止執行，停止的位址就是中斷點位址。

- 3). **Breakpoint Group (Alt\_G)**：將程式記憶體的一塊區域設為中斷點區域，最多可設 63 個中斷點區域，每個中斷點區域的通過次數(pass count)最多為 255 次。程式執行中只要執行到中斷點區域中的任一指令，則通過次數就會被減 1。執行本功能後會出現如下的畫面：



中斷點區域的設定語法如下：

```
0xAAAA 0xB BBBB 0xPP, 0xAAAA 0xB BBBB 0xPP, ...
```

其中 AAAA 代表 16 進位的中斷點區域起始位址值，BBBB 代表 16 進位的中斷點區域結束位址值，PP 為 16 進位的次數。例如要將位址 0x0005 到 0x0010 設為中斷點區域，而通過次數為 0x20 次，則輸入方式如下：

```
0x0005 0x0010 0x20
```

若一次要設多個中斷點區域，例如要將位址 0x0005 到 0x0010 設為中斷點區域，而通過次數為 0x20 次，位址 0x0020 到 0x0030 設為中斷點區域、通過次數為 0x10 次，則輸入方式：

```
0x0005 0x0010 0x20, 0x0020 0x0030 0x10
```

輸入完畢按一下↵鍵，則完成中斷點設定。

- 4). **Breakpoint Or (Alt\_B)**：將一些位址設為中斷點群組，最多可設 63 個中斷點群組，每個中斷點群組的通過次數(pass count)最多為 255 次。程式執行中只要執行到中斷點群組中的任一指令，則該群組的通過次數就會被減 1。

中斷點群組的設定語法如下：

```
(0xAAAA 0xAAAA ... 0xPP) (0xAAAA 0xAAAA ... 0xPP) ...
```

其中 AAAA 代表 16 進位的中斷點位址值，PP 為 16 進位的通過次數。例如要將位址 0x10, 0x20, 0x30, 0x40 設為一中斷點群組，而通過次數為 0x20 次，則輸入方式如下：

(0x10 0x20 0x30 0x40 0x20)

若一次要設多個中斷點群組，例如要將位址 0x10, 0x20, 0x30, 0x40 設為一中斷點群組，而通過次數為 0x20 次，將位址 0x15, 0x25, 0x35 設為一中斷點群組，而通過次數為 0x10 次，則輸入方式：

(0x10 0x20 0x30 0x40 0x20) (0x15 0x25 0x35 0x10)

輸入完畢按一下↵鍵，則完成中斷點設定。

5). Breakpoint Nest (Alt\_N)：將一些位址設為中斷點群組，可設多個中斷群組間，中斷群組間的關係為巢狀結構，較後面設定的中斷點群組為外層，當較外層群組(較後面設定的中斷點群組)中斷條件滿足後(也就是通過次數變 0 後)，較內層群組(較前面設定的中斷點群組)才開始作用。

6. Option 功能：在這功能下有 6 個子功能，分別說明如下：

- 1). Oscillator：本功能是用來選擇外接的振盪方式，在本功能下有二個選項：R/C circuit 及 Crystal。
- 2). Cycle：本功能是用來設定條件式跳躍指令(conditional skip instruction)的執行時間是 1 個指令週期(instruction cycle)或 2 個指令週期。在本功能下有二個選項：1 cycle 及 2 cycles。
- 3). Clock：本功能是用來設定 1 個指令週期需 2 振盪週期(clock)或 4 個振盪週期。在本功能下有二個選項：2 clocks 及 4 clocks。
- 4). Power：本功能是用來設定目標電路板(target board)電源的來源。在本功能下有二個選項：Internal 及 External。
- 5). Check Sum：本功能是用來設定程式檢查碼(check sum)的產生方式。在本功能下有二個選項：EM78156 及 EM78456。
- 6). About：本功能是顯示本軟體的版本。

### 設備需求與安裝 面板使用說明

請按下列步驟練習指令的操作，並觀察視窗的變化及指令的作用：

1. 將 E8-ICE 與個人電腦連接，並將電源接接好及電源開關撥至 ON 位置，並將 E8-ICE 的排線接到有 8 個 LED 的 EM78X56 DEMO BOARD 上。

2. 利用文書編輯軟體編輯下列程式。編輯完畢之後，將其存檔，檔名設為 LED.DT。

```

ORG 0x0
; **** Disable WDT ****
MOV A,@0x0 ;A=00H
IOW 0xE ;將 WDT 除能
; **** PORT6:輸出模式 ****
IOW 0x6 ;將 PORT 6 設為輸出模式
MOV A,@0x1 ;設定初始直
MOV 0x6,A ;輸出至 PORT6
LOOP:
RLC 0x6 ;R6 (PORT6) 向右旋轉
CALL DELAY
JMP LOOP
DELAY:
DJZ 0x10
JMP DELAY
DJZ 0x11
JMP DELAY
RET

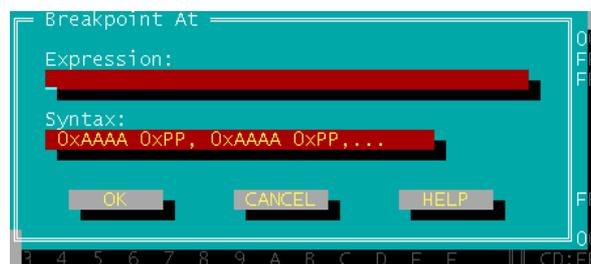
```

3. 執行檔案 ICE456.EXE。
4. 用 File|Open 或按[F1]按鍵，將 LED.DT 載入程式視窗中。
5. 用 Compile|Program 或按[F9]按鍵，將程式視窗中的原始程式組譯，並將組譯後的機械碼載入 ICE 中。
6. 用 Run|Free run 或按[F10]使程式執行，然後觀察 EM78X56 DEMO BOARD 上 LED 的變化。
7. 按[Esc]按鍵停止程式執行，此時 EM78X56 DEMO BOARD 上 LED 的不再變化。
8. 用 Run|Reset 或按[F6]按鍵執行重置 ICE 動作，做完重置動作後，PC 的內容變為 0。
- 9 單步執行：
  - (1). 用 Run|Trace into 或直接按[F7]按鍵：
 

執行 MOV A,@0x0，觀察暫存器 A 及 PC 的變化。

- (2). 用 Run|Trace into 或直接按[F7]按鍵：  
執行 IOW 0xE，觀察暫存器 IOCE(CE)及 PC 的變化。
  - (3). 用 Run|Trace into 或直接按[F7]按鍵：  
執行 IOW 0x6，觀察暫存器 IOC6(C6)及 PC 的變化。
  - (4). 用 Run|Trace into 或直接按[F7]按鍵：  
執行 MOV A,@0x1，觀察暫存器 A 及 PC 的變化。
  - (5). 用 Run|Trace into 或直接按[F7]按鍵：  
執行 MOV 0x6,A，觀察暫存器 R6(P6)、PC 及 LED 的變化。
  - (6). 用 Run|Trace into 或直接按[F7]按鍵：  
執行 RLC 0x6，觀察暫存器 R6(P6)、PC 及 LED 的變化。
  - (7). 用 Run|Trace into 或直接按[F7]按鍵：  
執行 CALL DELAY，觀察 PC 的變化。
  - (8). 用 Run|Trace into 或直接按[F7]按鍵：  
執行 DJZ 0x10，觀察暫存器 R10 及 PC 的變化。
  - (9). 用 Run|Trace into 或直接按[F7]按鍵：  
執行 JMP DELAY，觀察 PC 的變化。
10. 用 Run|Reset 或按[F6]按鍵執行重置 ICE 動作。
11. 單步執行：
- (1). 用 Run|Step over 或直接按[F8]按鍵：  
執行 MOV A,@0x0，觀察暫存器 A 及 PC 的變化。
  - (2). 用 Run|Step over 或直接按[F8]按鍵：  
執行 IOW 0xE，觀察暫存器 IOCE(CE)及 PC 的變化。
  - (3). 用 Run|Step over 或直接按[F8]按鍵：  
執行 IOW 0x6，觀察暫存器 IOC6(C6)及 PC 的變化。
  - (4). 用 Run|Step over 或直接按[F8]按鍵：  
執行 MOV A,@0x1，觀察暫存器 A 及 PC 的變化。
  - (5). 用 Run|Step over 或直接按[F8]按鍵：  
執行 MOV 0x6,A，觀察暫存器 R6(P6)、PC 及 LED 的變化。
  - (6). 用 Run|Step over 或直接按[F8]按鍵：  
執行 RLC 0x6，觀察暫存器 R6(P6)、PC 及 LED 的變化。

- (7). 用 Run|Step over 或直接按[F8]按鍵：  
執行 CALL DELAY，觀察 PC 的變化。
  - (8). 用 Run|Step over 或直接按[F8]按鍵：  
執行 JMP LOOP，觀察 PC 的變化。
  - (9). 用 Run|Step over 或直接按[F8]按鍵：  
執行 RLC 0x6，觀察暫存器 R6(P6)、PC 及 LED 的變化。
12. 用 Run|Reset 或按[F6]按鍵執行重置 ICE 動作。
  13. 用 Debug|Breakpoint At 或按 Alt\_A 按鍵，則出現如下的畫面：



14. 選用 Run|Run from|Initial with 8K\_step log 或按[F5]按鍵執行程式，則執行到中斷點時，程式會停止執行。執行時請觀察 EM78X56 DEMO BOARD 上 LED 的變化。
15. 選用 File|Trace log 或直接按[F2]按鍵將 Trace Buffer 的內容顯示出來。  
第一個執行的指令為第 5 行的 MOV A, @0x0，然後是第 4 行的 IOW 0xE，接著執行順序為第 3, 2, 1 行的指令，最後執行到被設為中斷點的指令，然後停止執行。
16. 選用 Modify|Program 或直接按 Alt\_E，則 PE2 或其他文書編輯軟體會被執行。在文書編輯軟體中將程式中指令 RLC 0x6 改為 RRC 0x6，然後關閉文書編輯軟體，並將改過的程式存檔。文書編輯軟體關閉後，會回到 ICE 軟體中，此時畫面上顯示的仍是修改前的原始程式。
17. 用 Compile|Program 或按[F9]按鍵，則修改過的原始程式會被組譯，並將組譯後的機械碼載入 ICE 中，同時畫面會顯示出修改後的原始程式。
18. 用 Run|Free run 或按[F10]使程式執行，然後觀察 EM78X56 DEMO BOARD 上 LED 的變化。
19. 按[Esc]按鍵停止程式執行，此時 EM78X56 DEMO BOARD 上 LED 的不再變化。
20. 選用 File|Quit 或按 Alt\_X 結束 ICE 程式。

## 學後評量

### 一、是非題：

- 1.( ) 所謂 ICE 它的英文全名是 In Circuit Emulator，也就是指萬用型燒錄器。
- 2.( ) EM78 ICE 可以透過 PRINTER PORT 直接與電腦連線。
- 3.( ) EM78 ICE 共有二種振盪方式，一為 crystal type，另一為 RC type。
- 4.( ) 在 EM78 ICE 的 File 檔案功能中的 Dump to ice (F3)選項，是將目前原始程式的機械碼傳送到 ICE 中。
- 5.( ) 在 EM78 ICE 的 RUN 執行功能中的 Trace into (F7)選項，是單步執行，並將副程式視為單步，也就是執行 CALL 指令時，直接將其所呼叫的副程式一次執行完畢。

### 二、選擇題：

- 1.( ) 下列何者是用來設定條件式跳躍的執行時間：(1) Clock (2) Cycle  
(3) Oscillator (4) Power
- 2.( ) 下列何者是單步執行，並執行 CALL 指令時，會進入其所呼叫的副程式單步執行：(1) Step over (2) Run from (3) Reset (4) Trace into

### 三、實作測驗：

**作業 1**：透過 EM78 ICE，自行安裝硬體與軟體，並且可以由電腦 Down load 程式到 ICE，工作時間為 30 分鐘。

在工作之前，請先填好工作計畫單，送給老師認可後才開始測驗。

### 我的工作計畫

作業名稱：\_\_\_\_\_

工作開始日期：\_\_\_\_\_ 完成日期：\_\_\_\_\_

工作時間：\_\_\_\_\_分鐘 教師認可：\_\_\_\_\_

我做上列工作時所需用之工具及機器：

1. \_\_\_\_\_ 5. \_\_\_\_\_ 9. \_\_\_\_\_

2. \_\_\_\_\_ 6. \_\_\_\_\_ 10. \_\_\_\_\_

3. \_\_\_\_\_ 7. \_\_\_\_\_ 11. \_\_\_\_\_

4. \_\_\_\_\_ 8. \_\_\_\_\_ 12. \_\_\_\_\_

### 我所需要的材料及消耗品：

名 稱	說 明	規 格	數 量	估 價

### 我計畫如何做我的作業：

工作步驟	安全注意事項	工作時注意要領

注意：1. 現在你已完成你的作業計畫，請不要馬上工作，你先檢討一下，有沒有其他更好的方法呢？有沒有遺漏呢？將你的計畫送給你的老師認可；然後再開始工作，工作時間之作業(1)：30 分鐘。

2. 當你做好了作業，請對你的作業做自我評價(Self-evaluation)，然後請老師來檢查評分。

我的作業評分 = \_\_\_\_\_ 分，屬於 \_\_\_\_\_ 等。

A = 95 分以上 B = 85 分以上 C = 75 分以上

D = 65 分以上 E = 64 分以下

B.我的工作計畫得分\_\_\_\_\_分，屬於\_\_\_\_\_等。

C.安全習慣得分\_\_\_\_\_分，屬於\_\_\_\_\_等。

D.學習態度得\_\_\_\_\_分，屬於\_\_\_\_\_等。

E.教師評分

1.作業得分\_\_\_\_\_ 3.安全習慣\_\_\_\_\_

2.工作計畫\_\_\_\_\_ 4.學習態度  
得分\_\_\_\_\_

總得分\_\_\_\_\_屬於\_\_\_\_\_等

## 參考書籍

- |                        |      |         |
|------------------------|------|---------|
| 1.EM78X56 入門與實作        | 宏友出版 | 詹耀仁 胡建偉 |
| 2.單晶片微控制 EM78X56 原理與實作 | 宏友出版 | 王宜楷     |
| 3.EM78447 入門實作         | 宏友出版 | 鄧錦城     |