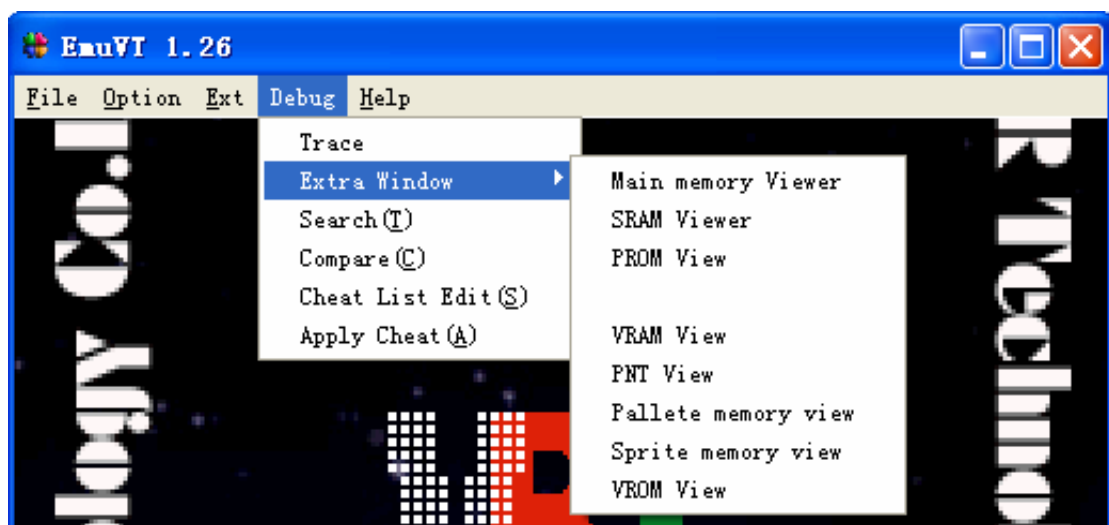


Debug Windows users note:

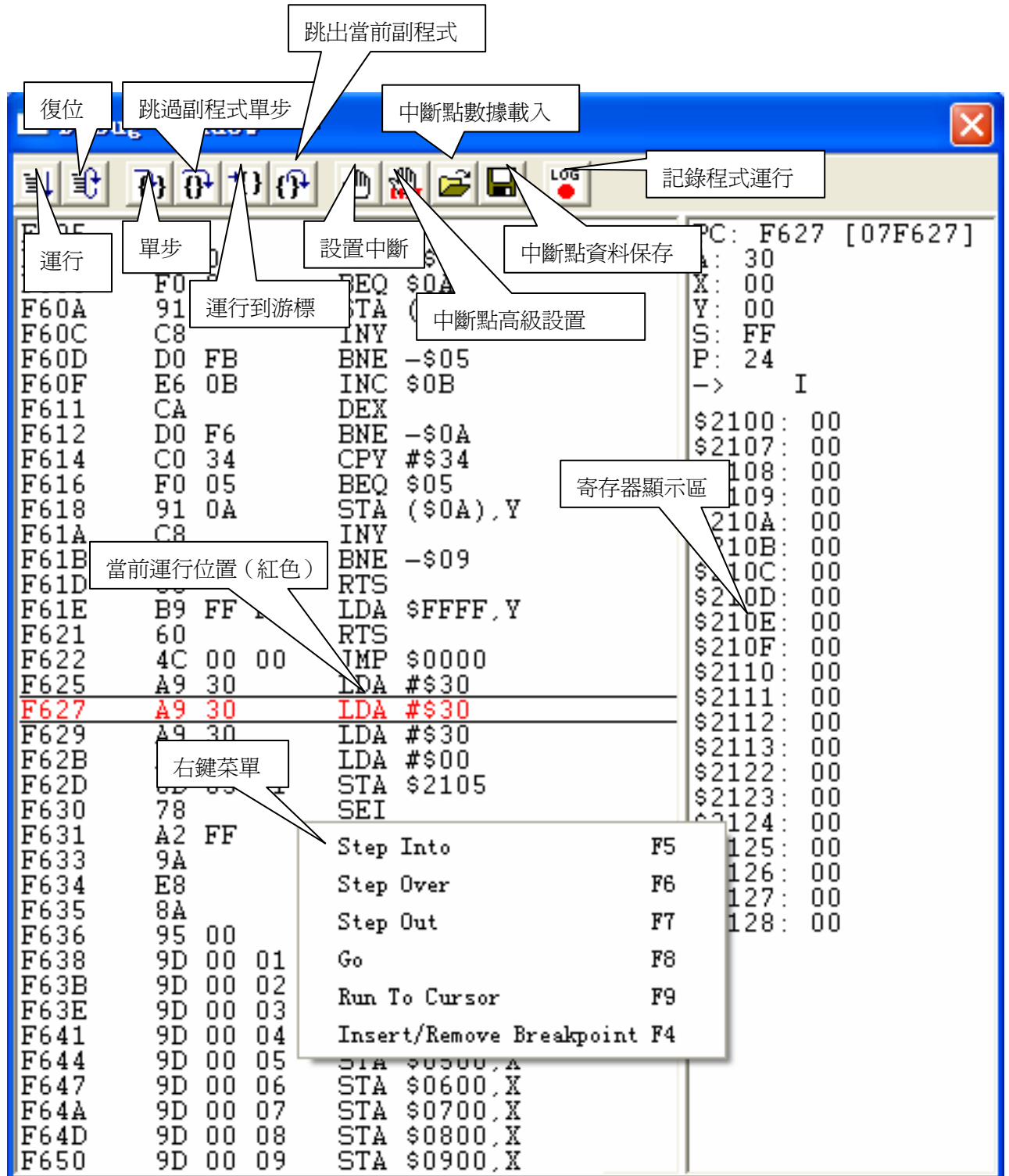
設置完之後，Trace Set 視窗可以關閉也可以不關閉，點運行那個按鈕（Debug Windows 左起第一個按鈕）才能跟蹤，或者點重設那個按鈕（Debug Windows 左起第一個按鈕）然後點運行那個按鈕（Debug Windows 左起第二個按鈕）。在 Debug 的過程中不要把 Debug Windows 關閉，否則不能跟蹤。Trace Set 視窗中的“Enable”選項只是顯示這個 break point 是否使能,而且是反的,沒勾的時候是使能，有勾的時候是禁止。

Debug 菜單



Trace	跟蹤調試視窗
Extra Window	記憶體查看器
Search	主記憶體搜索查找
Compare	主記憶體比較
Cheat List Edit	金手指設置器
Apply Cheat	金手指致能

Trace 跟蹤調試視窗



The screenshot shows the Trace window of the VT1682 Emulator. The window title is "Trace 跟蹤調試視窗". The interface includes a toolbar with various icons and a main display area showing assembly code and registers.

Annotations:

- 復位 (Reset)
- 跳過副程式單步 (Skip Subroutine Single Step)
- 跳出當前副程式 (Exit Current Subroutine)
- 中斷點數據載入 (Load Breakpoint Data)
- 記錄程式運行 (Record Program Execution)
- 運行 (Run)
- 單步 (Single Step)
- 設置中斷 (Set Breakpoint)
- 中斷點資料保存 (Save Breakpoint Data)
- 運行到游標 (Run to Cursor)
- 中斷點高級設置 (Advanced Breakpoint Settings)
- 當前運行位置 (紅色) (Current execution position (red))
- 右鍵菜單 (Right-click menu)
- 寄存器顯示區 (Register display area)

Assembly Code Display:

Address	Hex	Hex	Instruction
F60A	91		BEQ \$0A
F60C	C8		STA (
F60D	D0 FB		INY
F60F	E6 0B		BNE -\$05
F611	CA		INC \$0B
F612	D0 F6		DEX
F614	C0 34		BNE -\$0A
F616	F0 05		CPY #\$34
F618	91 0A		BEQ \$05
F61A	C8		STA (\$0A),Y
F61B			INY
F61D			BNE -\$09
F61E	B9 FF		RTS
F621	60		LDA \$FFFF,Y
F622	4C 00 00		RTS
F625	A9 30		JMP \$0000
F627	A9 30		LDA #\$30
F629	A9 30		LDA #\$30
F62B			LDA #\$00
F62D			STA \$2105
F630	78		SEI
F631	A2 FF		
F633	9A		
F634	E8		
F635	8A		
F636	95 00		
F638	9D 00 01		
F63B	9D 00 02		
F63E	9D 00 03		
F641	9D 00 04		
F644	9D 00 05		
F647	9D 00 06		
F64A	9D 00 07		
F64D	9D 00 08		
F650	9D 00 09		

Register Display:

```

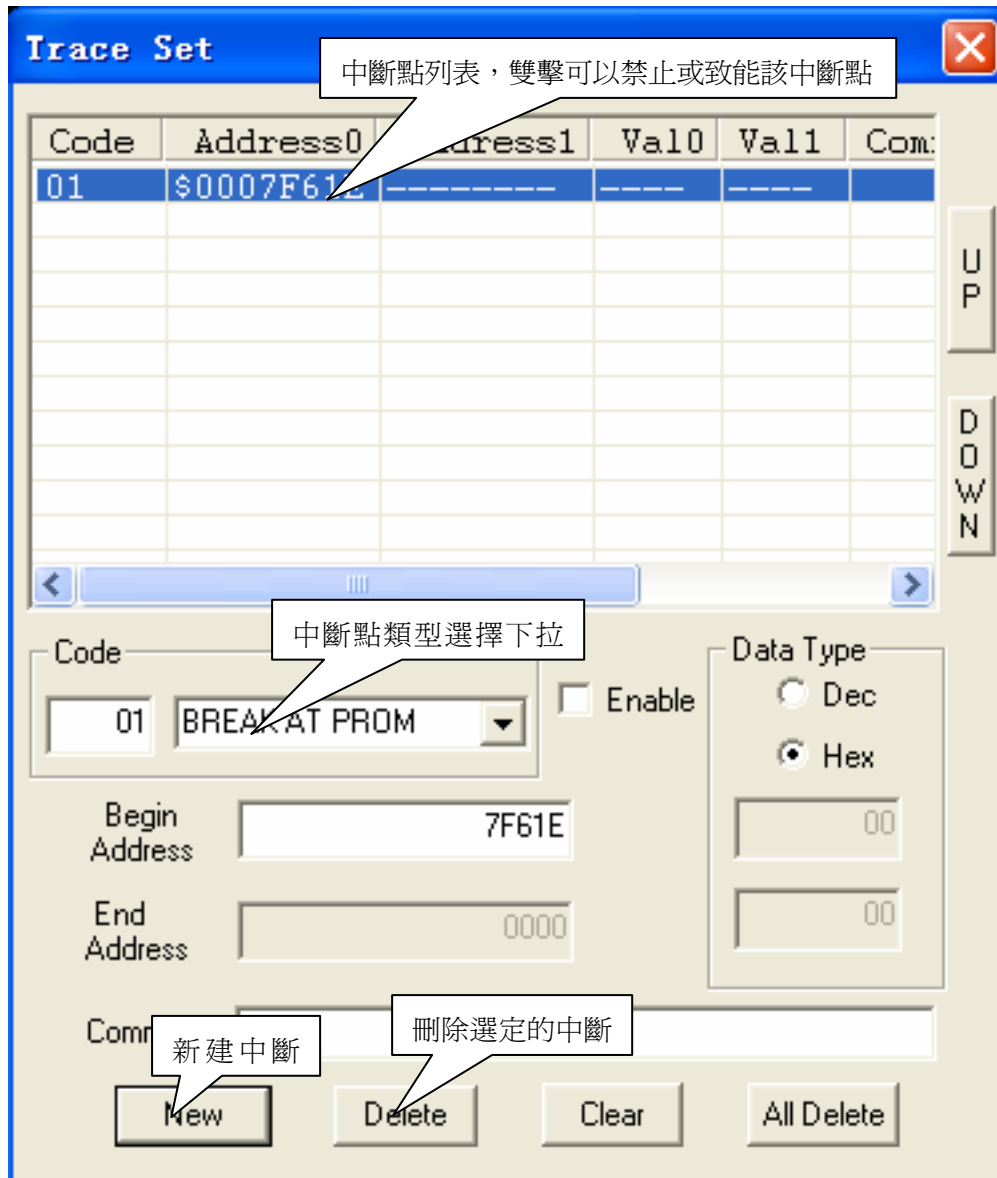
PC: F627 [07F627]
A: 30
X: 00
Y: 00
S: FF
P: 24
-> I
$2100: 00
$2107: 00
$2108: 00
$2109: 00
$210A: 00
$210B: 00
$210C: 00
$210D: 00
$210E: 00
$210F: 00
$2110: 00
$2111: 00
$2112: 00
$2113: 00
$2122: 00
$2123: 00
$2124: 00
$2125: 00
$2126: 00
$2127: 00
$2128: 00

```

Right-click menu:

- Step Into (F5)
- Step Over (F6)
- Step Out (F7)
- Go (F8)
- Run To Cursor (F9)
- Insert/Remove Breakpoint (F4)

按下“中斷點高級設置”：



中斷點類型：

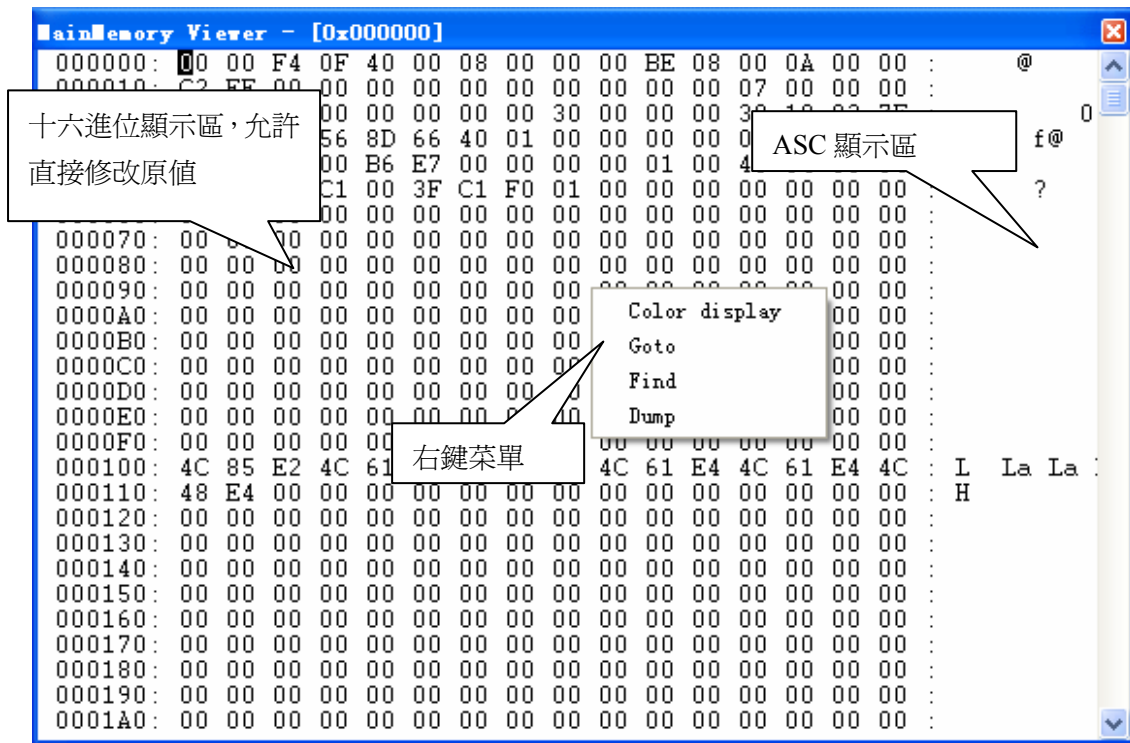
BREAK AT PROM	運行到物理位址中斷
BREAK AT ADDRESS	運行到邏輯位址中斷
BREAK IN ADDRESS	運行到邏輯位址範圍中斷
BREAK IN PROM	運行到物理位址範圍中斷
BREAK AT WRITE	寫指定位址（寄存器）中斷
BREAK IN WRITE	寫指定位址（寄存器）範圍中斷
BREAK AT WRITE AT	以指定值寫指定位址（寄存器）中斷
BREAK IN WRITE AT	以指定值寫指定位址（寄存器）範圍中斷
BREAK AT WRITE IN	以指定範圍值寫指定位址（寄存器）中斷

BREAK IN WRITE IN	以指定範圍值寫指定位址（寄存器）範圍中斷
BREAK AT READ	讀指定位址（寄存器）中斷
BREAK IN READ	讀指定位址（寄存器）範圍中斷
BREAK AT READ AT	讀指定位址（寄存器）得到指定值中斷
BREAK IN READ AT	讀指定位址（寄存器）範圍得到指定值中斷
BREAK AT READ IN	讀指定位址（寄存器）得到指定範圍值中斷
BREAK IN READ IN	讀指定位址（寄存器）範圍得到指定範圍值中斷
EXT IRQ BREAK	設定擴展 IRQ 中斷中斷點
NMI BREAK	設定 NMI 中斷中斷點
IRQ BREAK	設定 IRQ 中斷中斷點
SCPU IRQ BREAK	設定 SCPU IRQ 中斷中斷點
TIMER IRQ BREAK	設定 TIMER IRQ 中斷中斷點

舉例：如果欲設置這樣一個中斷點，當對\$200 到\$20F 範圍有寫操作，且寫入的值為\$00 到\$7F 之間的任意值時。首先我們點 New 按鈕，然後在下拉式列示方塊選擇“BREAK IN WRITE IN”，代碼是 25，在 Begin Address 中輸入 200，在 End Address 中輸入 210，選擇 Hex 模式，在 Date 靠上的一個輸入框輸入 00，靠下一個輸入框輸入 80，這樣就設置好了一個中斷點。

Extra Window 記憶體查看器

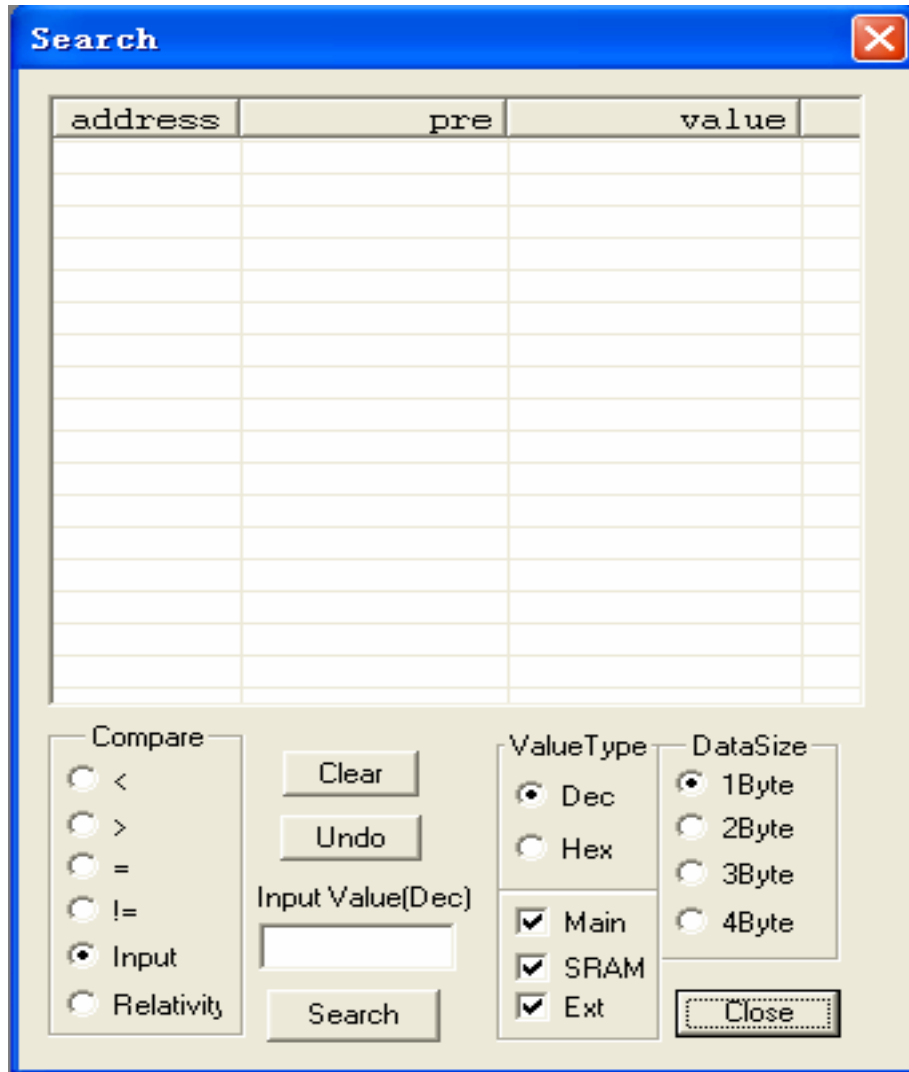
Main memory Viewer	主記憶體查看器
SRAM Viewer	延伸記憶體查看器
PROM Viewer	程式空間查看器
VRAM Viewer	PPU RAM 記憶體查看器
PNT Viewer	PPU PNT 記憶體查看器
Pallete memory Viewer	調色板記憶體查看器
Sprite memory Viewer	卡通記憶體查看器
VROM Viewer	PPU 點陣區查看器



右鍵菜單說明：

- Color Display 以特殊顏色標示特殊的內容
- Goto 跳到指定的記憶體偏移位址
- Find 在當前記憶體區域查找內容
- Dump 將當前記憶體區域內容以二進位或文本方式存到指定檔

Search 主記憶體搜索查找



這個功能類似于“金山遊俠”或“FPE”之類的遊戲工具，用來搜索找出特定記憶體單元的含義，比如做一些遊戲的 HACK，我們需要找出遊戲中“命”的數量是存在哪個記憶體單元，以便我們做出不死“命”的遊戲版本來。