



SyncMOS MSM9066 使用手冊

適用微控制器型號：SM59XX 系列、SM59RXX 系列、SM39RXX 系列

內 容

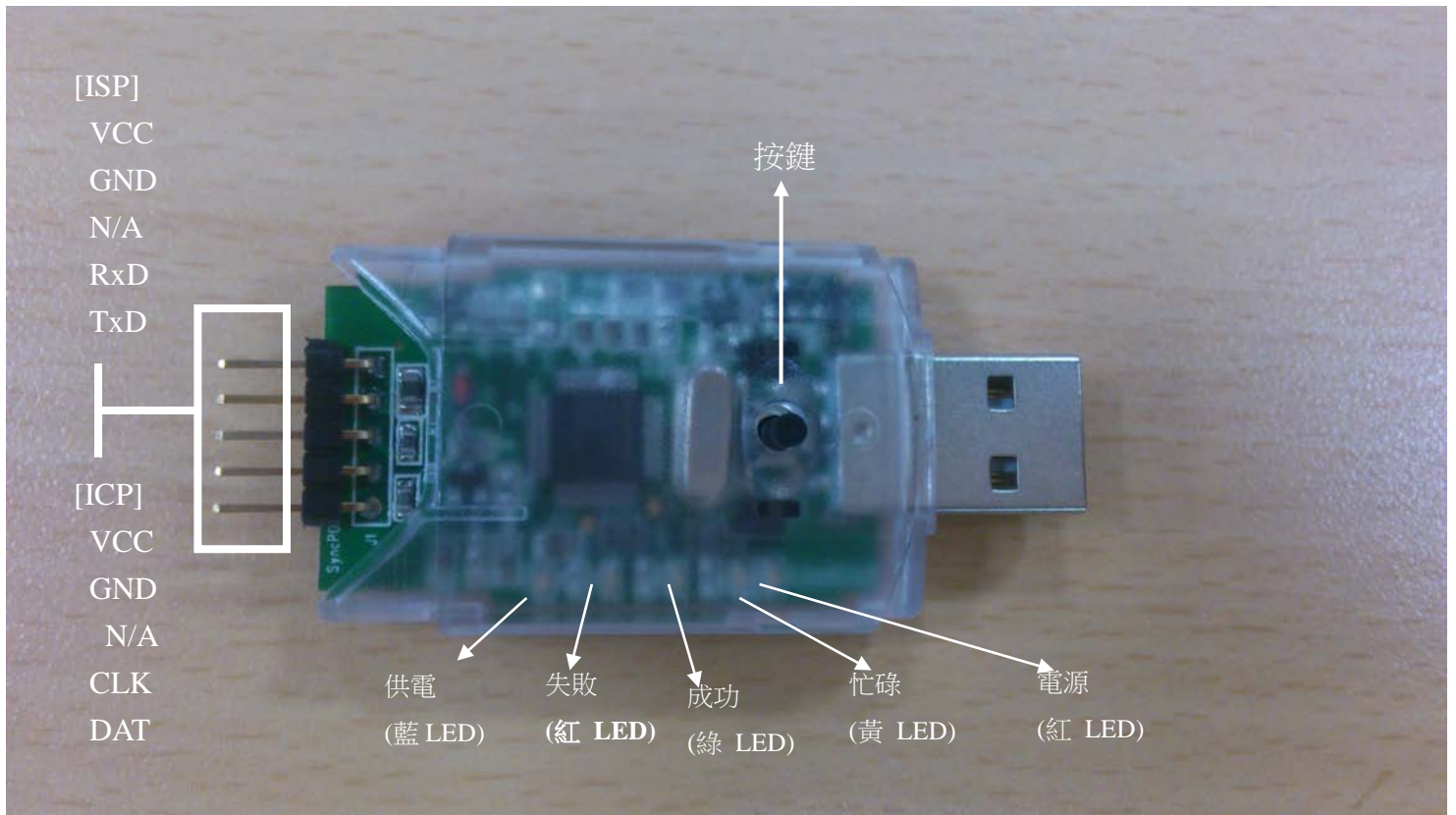
1 前言	2
2 MSM9066 外觀圖	2
3 ISP 簡介	3
4 使用 ISP 功能之 Configuration 設定	4
5 進入 ISP 機制說明	6
6 ICP 簡介	9
7 ICP 與 ISP 差異比較表	10
8 SMAP 設定說明	11
9 SMAP 操作介面簡介	15
10 ISP 燒錄教學	19
11 ICP 燒錄教學	29
12 離線燒錄教學	30



1 前言

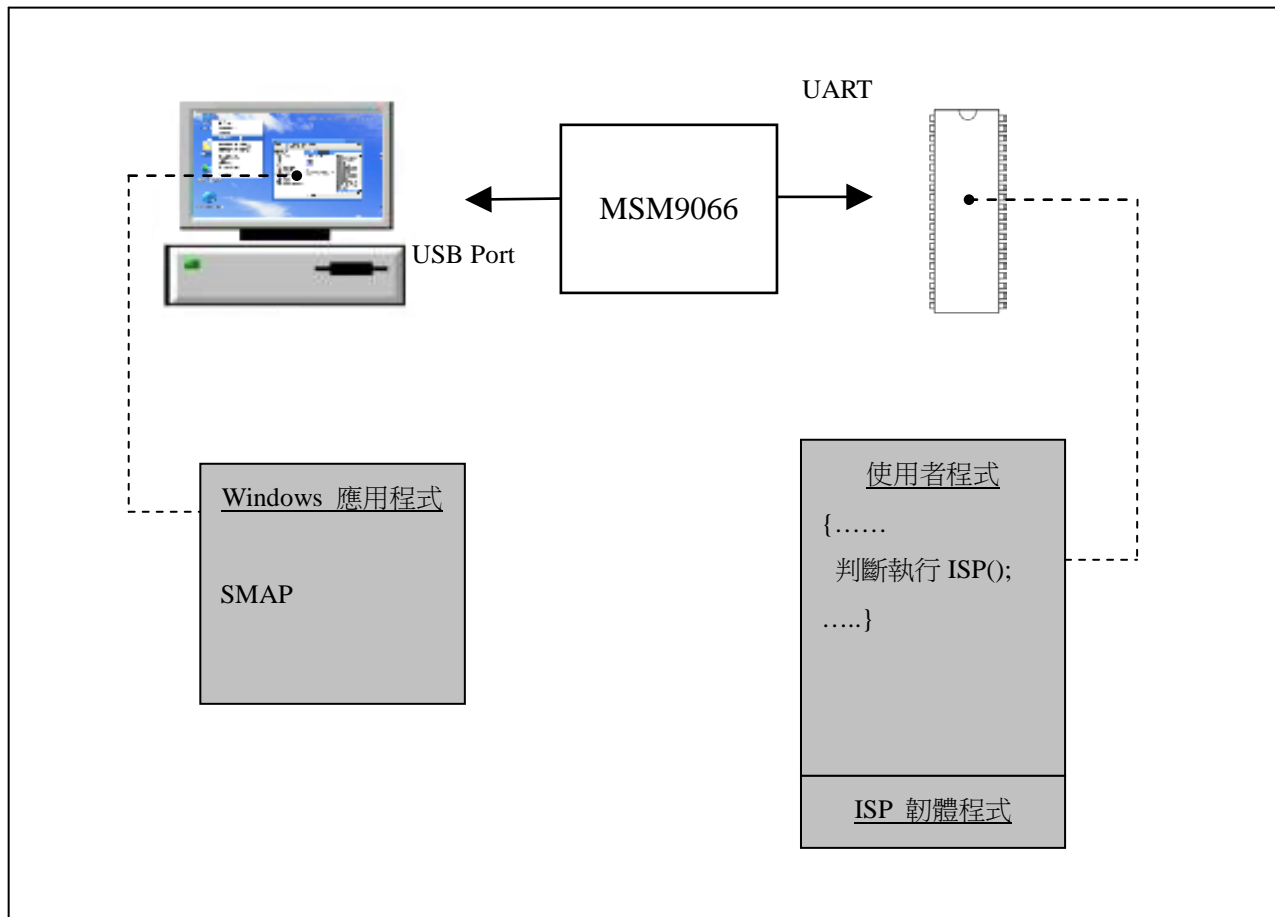
- 1.1 MSM9066 支援 ICP，使用者不需要額外燒錄 ISP Code 於 IC 上，即可將所要燒錄的程式燒錄於 IC 中，也可支援 ISP 功能。
- 1.2 可支援之傳輸介面：USB。

2 MSM9066 外觀圖



3 ISP 簡介

ISP(In System Programming)，它是一個可線上燒錄的功能，何謂可線上燒錄，就是使用者不需將 IC 從機器中取下，放在特定的燒錄器進行燒錄的動作，它可直接透過特定的傳輸介面，例如 UART，搭配特定的傳輸協定即可進行晶片抹除、寫入，為一便利的功能。但在執行此功能前須預先將 ISP 韌體程式透過燒錄器燒進晶片內，這樣才可以跟 PC 端的應用程式進行連結與溝通，這種方式我們稱之為 ISP。下圖所示為 ISP 系統圖。



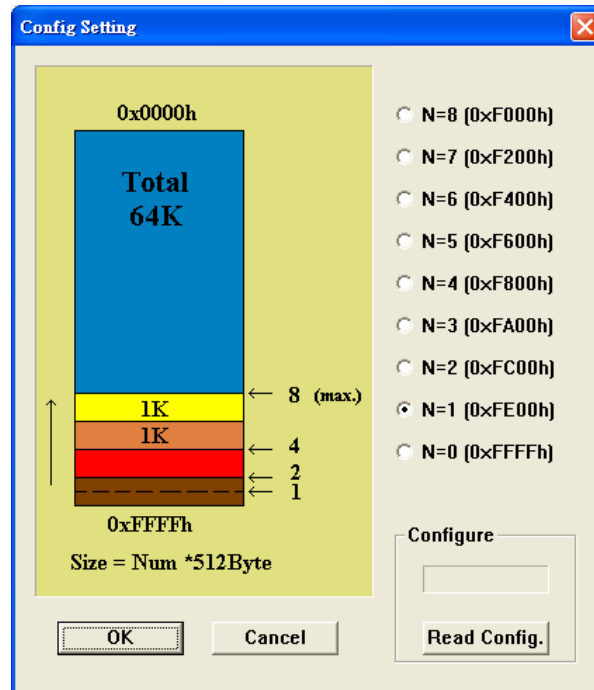
ISP 系統圖

新茂提供上圖中 Windows 應用程式、ISP 韌體程式的 hex/bin 和原始碼供客戶使用，以縮短客戶開發時間。為方便開發者使用，出廠前已將 ISP 韌體程式燒入 IC。

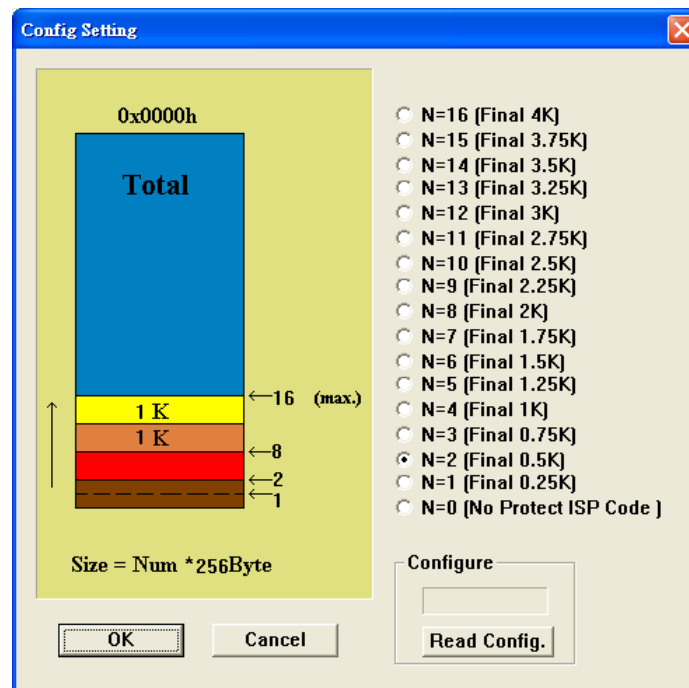
4 使用 ISP 功能之 Configuration 設定

若 IC 出廠的 ISP code 已經被通用燒錄器抹除過，需重新燒錄 ISP code 才能正常使用 ISP 功能，請依照下列步驟燒錄 ISP code：

4.1 在 SM59XX 系列 IC，請使用通用型燒錄器設定 N=1。

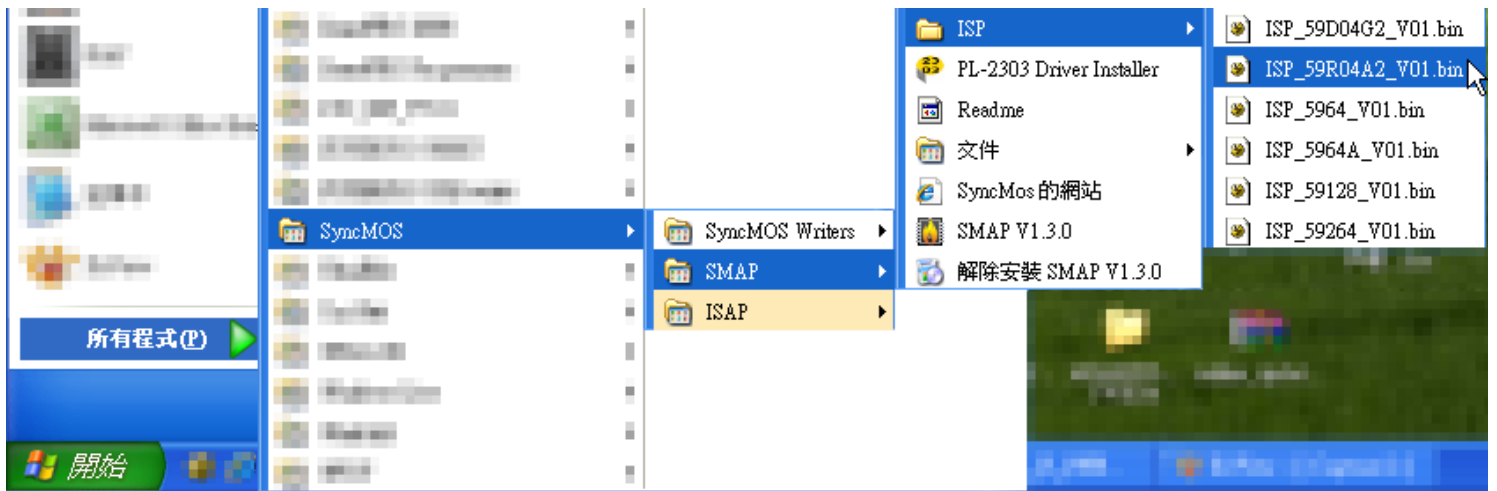


4.2 在 SM59RXX、SM39RXX 系列 IC，請使用通用型燒錄器設定 N=2。





4.3 依所使用的 IC 型號載入 ISP code，出廠 ISP code 在程式集中的 SMAP 路徑找到。





5 進入 ISP 機制說明

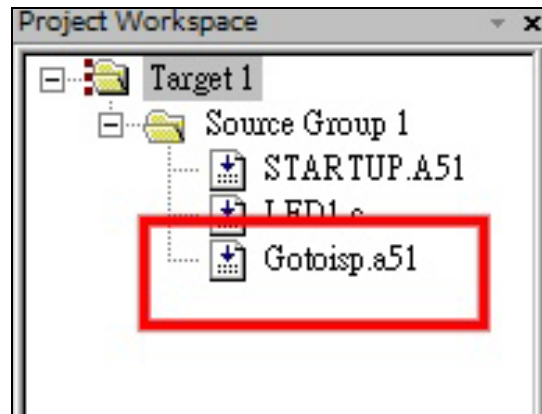
5.1 硬體 ISP：使用 IC 內部的硬體機制，使 PC Counter 可跳躍到 ISP code 執行 ISP 功能。未列出之 IC 請參考 Datasheet。

IC 型號	第一個位置的值 \$0000=FFH	P2.6/P2.7 = 0	P4.3 = 0	P3.0 輸入兩個時脈
Standard package				
SM5964	Yes	No	No	No
SM5964A	Yes	Yes	Yes	No
SM59R02/03/04A1	Yes	Yes	Yes	Yes
SM59R04A2	Yes	Yes	Yes	Yes
SM59R05/09/16A3	Yes	Yes	Yes	Yes
SM59R05/09/16A5	Yes	Yes	Yes	Yes
SM59R05/09/16G6	Yes	Yes	Yes	Yes
SM59R01/02G1	Yes	Yes	Yes	Yes
Low pin count package				
SM39R20/4051	Yes	No	P1.5 = 0	Yes
SM39R02/04G1	Yes	No	P1.5 = 0	Yes
SM39R08/12/16A2	Yes	Yes	P1.6 = 0	Yes

各 IC 型號支援硬體 ISP 機制一覽表

- 5.1.1 “第一個位置的值 \$0000=FFH”：當 IC 為空片時，PC Counter 可自行跳躍到 ISP code。
- 5.1.2 “P2.6/P2.7 = 0”：當 Power-on 或 Reset 發生時，若 P2.6 與 P2.7 同時為低電位，PC Counter 可跳躍到 ISP code。
- 5.1.3 “P4.3 = 0”：當 Power-on 或 Reset 發生時，若 P4.3 為低電位，PC Counter 可跳躍到 ISP code。因 DIP 無 Port 4，所以必須要 QFP 或 PLCC 封裝 IC 才可支援。
- 5.1.4 “P3.0 輸入兩個時脈”：當 Power-on 或 Reset 發生時，若 P3.0 收到兩個時脈，PC Counter 可跳躍到 ISP code。

5.2 軟體 ISP：以 Keil C 為例，要讓使用者程式能夠進入 ISP 韌體程式執行，須在 Project 內加入 Gotoisp.a51 檔案一起編譯，並在使用者程式加入是否執行 ISP 的判斷。如下所示：



Gotoisp.a51 檔案：

```
; GOTOISP.A51

NAME GOTOISP
?PR?GotoIsp?GOTOISP SEGMENT CODE
    PUBLIC GotoIsp
    RSEG ?PR?GotoIsp?GOTOISP
    USING 0
GotoIsp:
    LJMP 03E00H ;isp start address
    END
```

使用者程式：

```
#include <...h>
...
extern void GoToIsp(void); // 使用者須在呼叫此函式先定義
...
void main(void)
{
    if(P4.3 == 0) // 是否執行ISP的判斷，此 Pin 4.3 使用者可自行定義。
    {
        GoToIsp();
    }
    ...
    ...
}
```

備註：

※ Code Size 為 16KB 以下的 IC(如 SM59R04A2)，其 ISP 位址在 3E00H；Code Size 為 64KB 以上



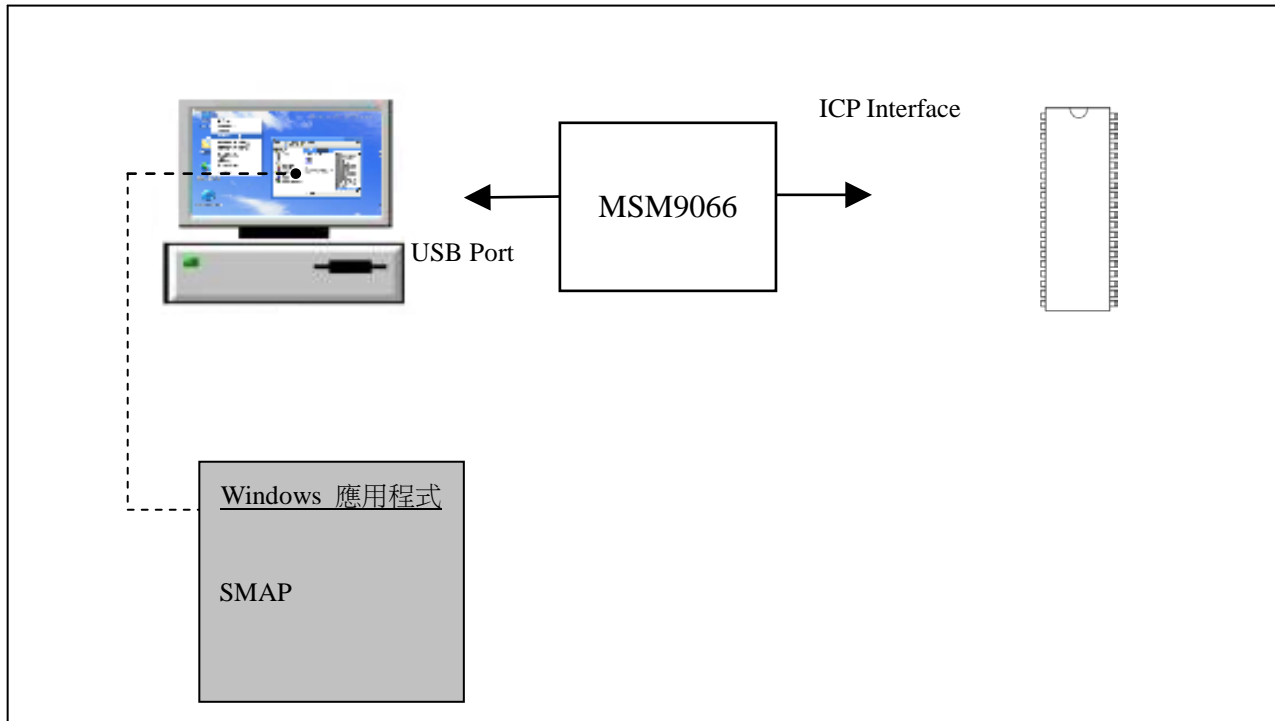
的 IC(如 SM5964A)，其 ISP 位址在 FE00H，請依使用的 IC 型號修改需跳躍的 ISP 位址。

- ※ SM59XX 系列 ISP code 使用 Timer 1 當鮑率產生器，因此若客戶在主程式中使用 Timer2 作為鮑率產生器，請必須在跳躍到 ISP code 前將 T2CON 給 00H，避免 Timer2 佔用 UART 串口。



6 ICP 簡介

ICP(In Circuit Programming)，它是一個可線上燒錄的功能，與 ISP 相較，由於 ISP 須預先將 ISP 韌體程式透過燒錄器燒進晶片內，必須佔用 IC 的 code size，而 ICP 因內建在晶片硬體架構，不需內含 ISP boot code 即可進行晶片抹除、寫入，這種方式我們稱之為 ICP。下圖所示為 ICP 系統圖。



ICP 系統圖



7 ICP 與 ISP 差異比較表

上線燒錄方式	ISP	ICP(2-Wires interface)
支援 IC	SM5964 SM5964A SM59R02/03/04A1 SM59R04A2 SM59R05/09/16A3 SM59R05/09/16A5 SM59R05/09/16G6 SM59R01/02G1 SM39R20/4051 SM39R02/04G1 SM39R08/12/16A2	SM59R02/03/04A1 SM59R04A2 SM59R05/09/16A3 SM59R05/09/16A5 SM59R05/09/16G6 SM39R20/4051 SM39R02/04G1 SM39R08/12/16A2
燒錄 IC 組態	僅 SM59RXX 系列可	可
IC 加密	可	可
介面	GND,RX,TX	GND,OCI_SCL,OCI_SDA
備註	1. 需內含 0.5k ISP code	無
優點	可透過 MAX232 或 USB-to-UART IC 直接與 PC 連線更新	不需內含 ISP code

未列出之 IC 請參考 Datasheet。

8 SMAP 設定說明

8.1 設定燒錄器：

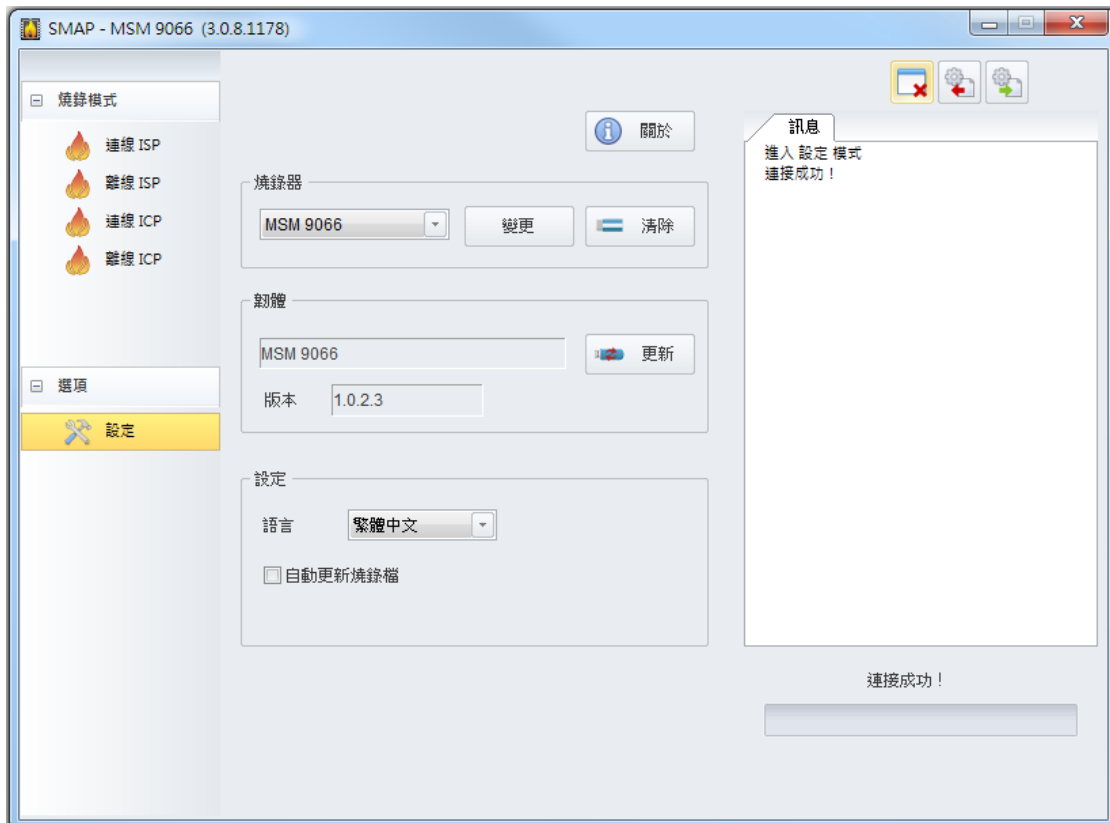


8.1.1 首先將 SyncMOS SMAP 安裝於您的電腦，安裝完成之後在您的桌面會產生一個捷徑。

8.1.2 第一次啟動 SMAP 時，會跳出設定燒錄器視窗，有 MSM9059、MSM9066、MSM9171 可供選擇，請選擇 MSM9066，然後按<確定>。



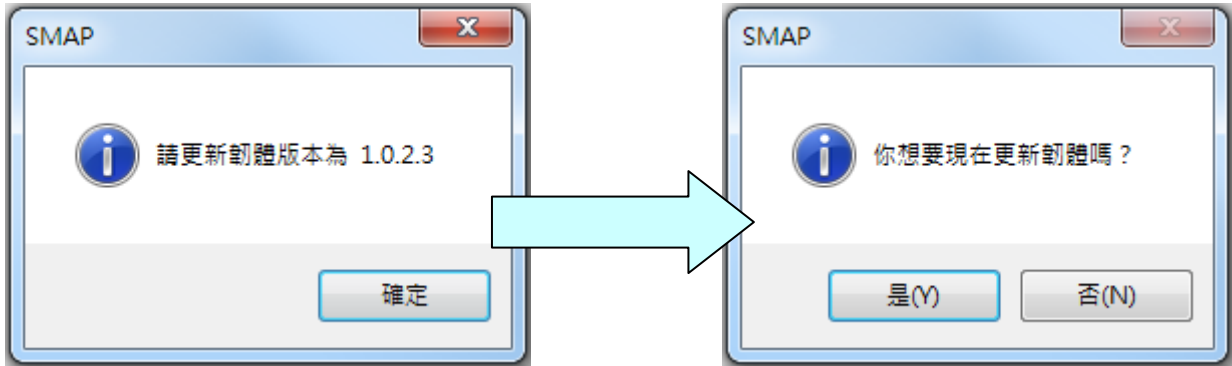
8.1.3 接著跳出 SMAP 主視窗，可在[設定]畫面中變更燒錄器型號，更新韌體以及改變語言設定。



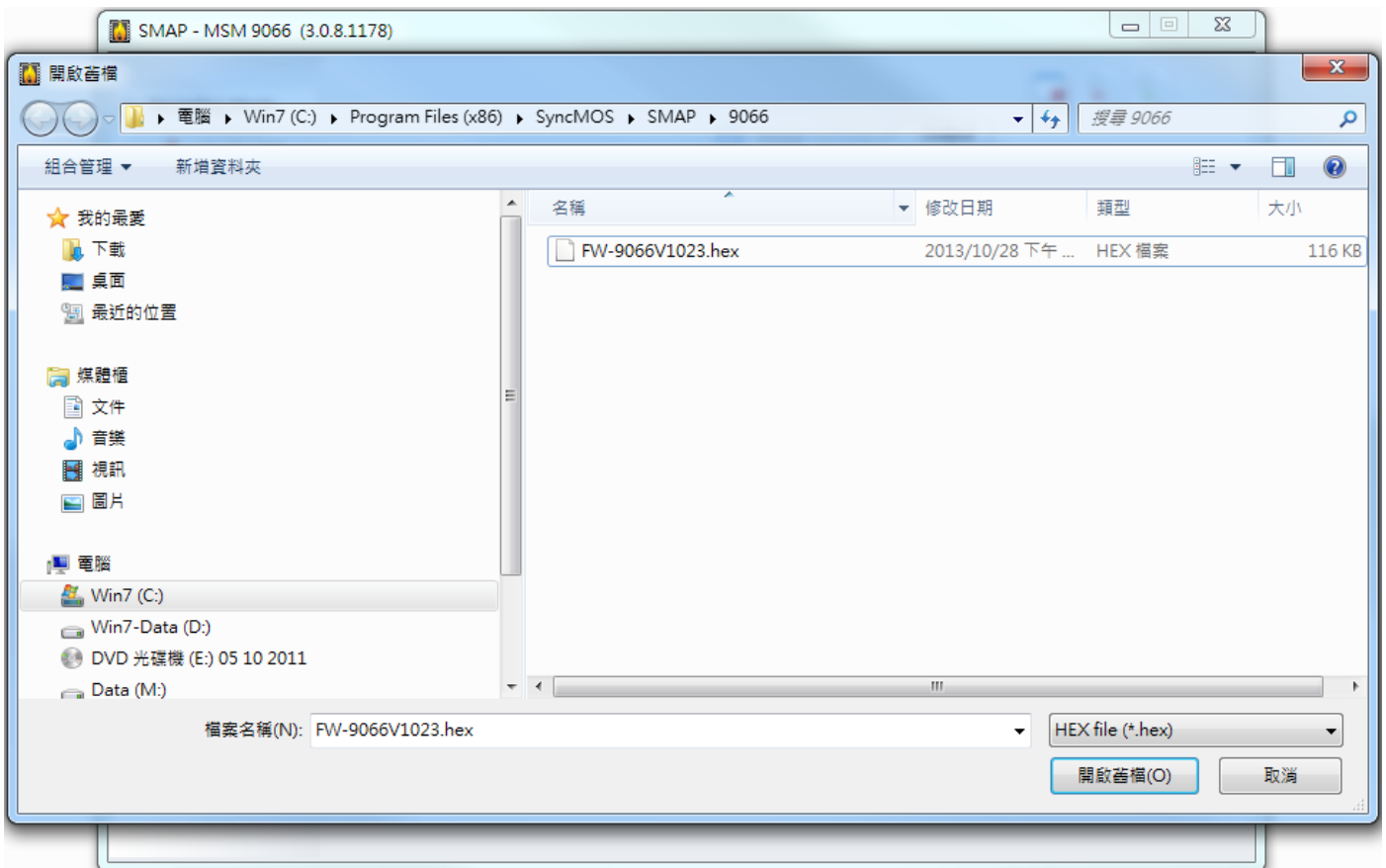


8.2 自動更新韌體：

8.2.1 進入**[燒錄模式]**並點選**<連接>**，若您的 MSM9066 韌體與目前的 SMAP 版本不符，會跳出視窗請您更新韌體。依照下圖流程點選，跳出**[更新的方式]**視窗時請選**<自動>**並點選**<確定>**鍵。

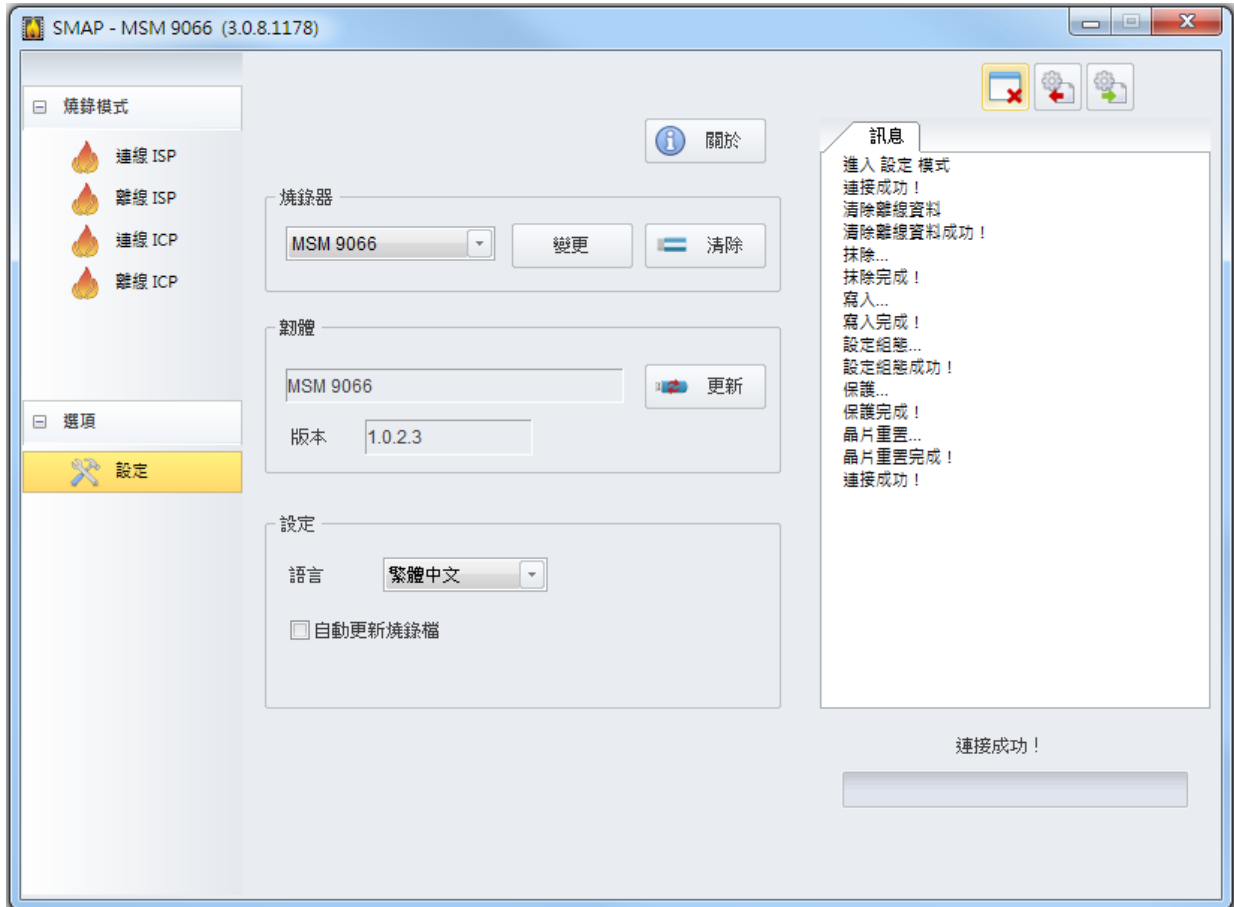


8.2.2 選擇 FW-9066VXXX.hex，並點選**<開啟(O)>**鍵，即開始更新。





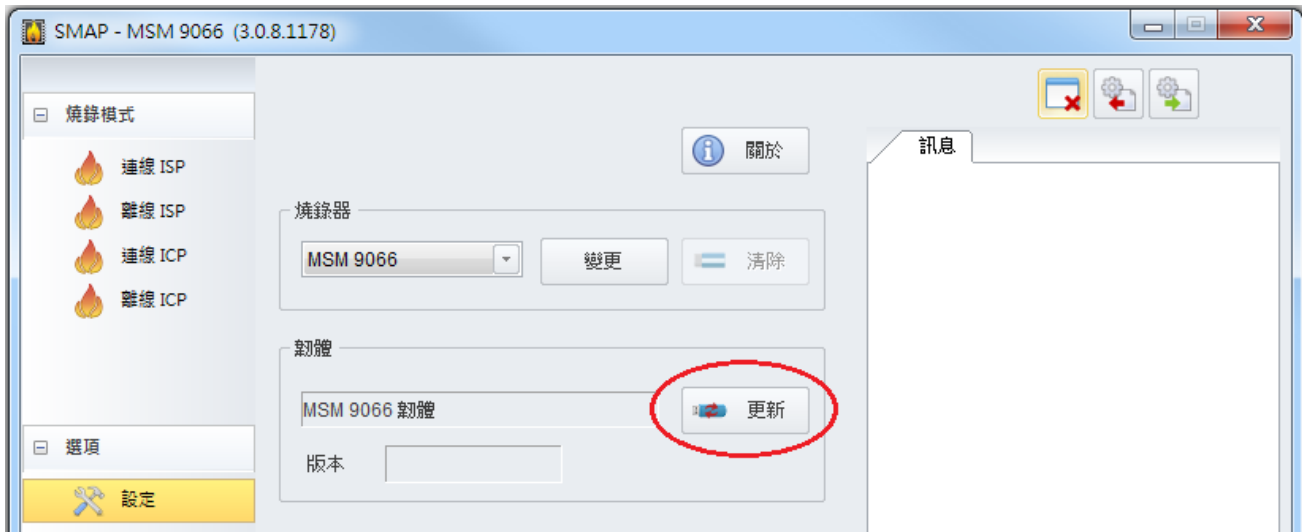
8.2.3 訊息視窗會依序顯示”掃描完成”、”抹除完成！”、”寫入完成！”、”保護完成！”、”晶片重置完成！”、”連接成功！”，即代表完成更新。



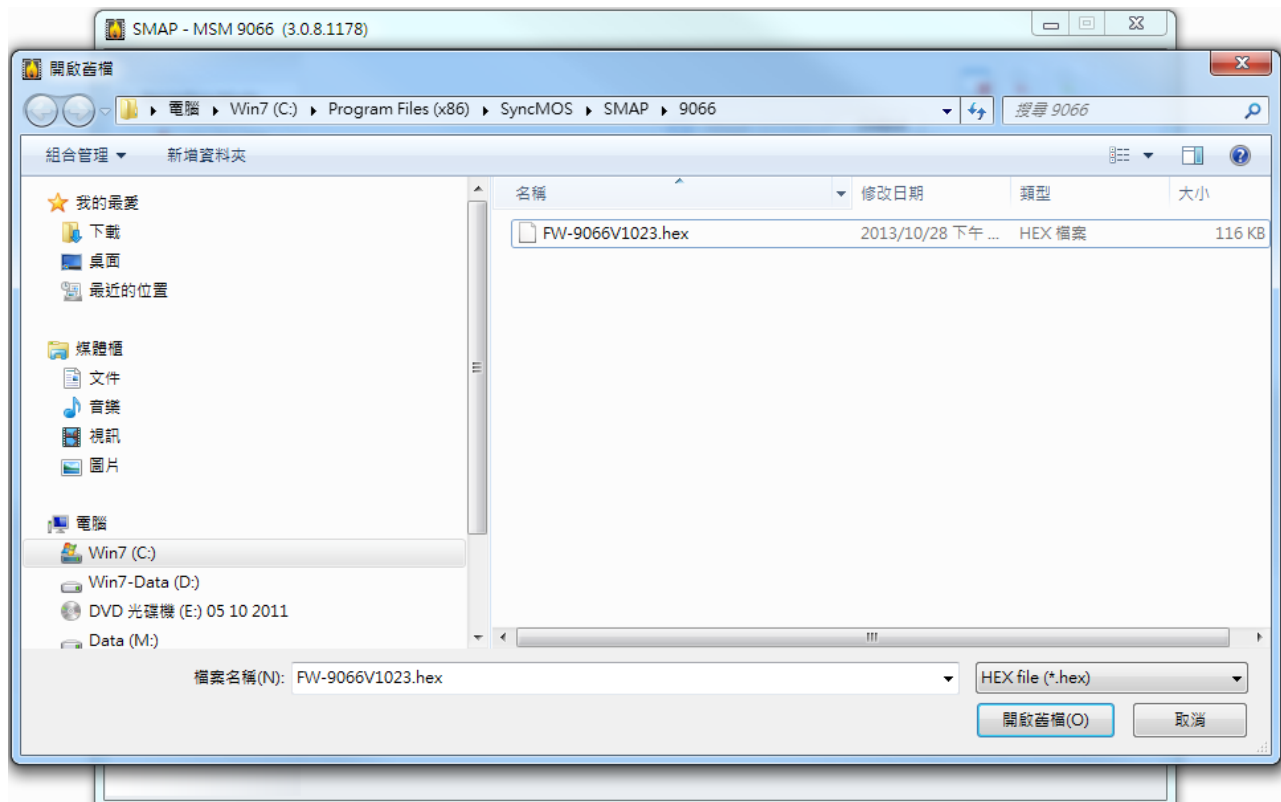


8.3 手動更新韌體：

8.3.1 當使用自動更新韌體失敗時，也可以使用手動更新方式讓韌體能順利更新。先壓住按鈕同時將 MSM9066 插入 USB 插孔，在[設定]頁面，COM Port 的設定必須為正確的連接串口，再按<更新>按鈕，會跳出[更新方式]視窗。



8.3.2 按<確定>後跳出[開啟]視窗，選擇 FW-9066VXXX.hex，然後按<開啟(O)>，即可開始更新韌體。



9 SMAP 操作介面簡介

9.1 燒錄模式：有連線 ISP、離線 ISP、連線 ICP 與離線 ICP 四種模式可供選擇。



9.1.1 **連線 ISP**：MSM9066 需連接 PC，並透過 Rx、Tx 連接 MCU 燒錄。需設定晶振頻率，請依照目標板所用的晶振頻率設定。



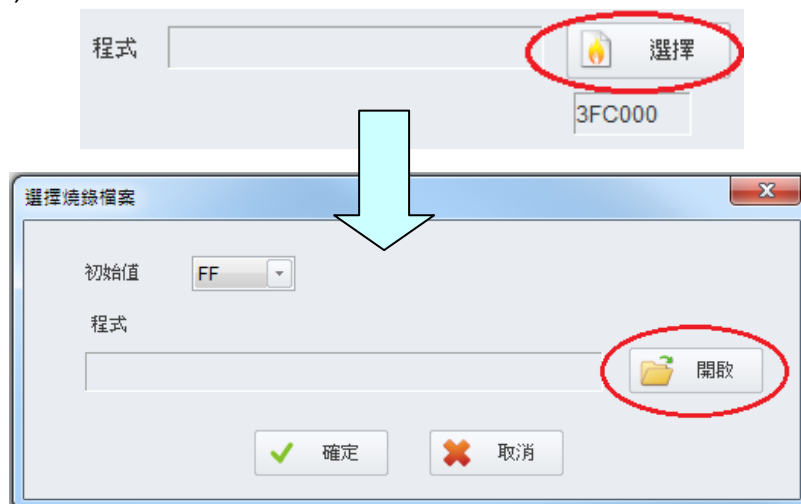
9.1.2 **離線 ISP**：可將待燒錄的 code 先傳到 MSM9066 上，便可不須連線電腦，透過 Rx、Tx 燒錄連接 MCU 進行燒錄。設定注意事項如同連線 ISP。

9.1.3 **連線 ICP**：MSM9066 需連接 PC，並透過 OCI_SCL、OCI_SDA 連接 MCU 燒錄。

9.1.4 **離線 ICP**：可將待燒錄的 code 先傳到 MSM9066 上，便可不須連線電腦，透過 OCI_SCL、OCI_SDA 連接 MCU 進行燒錄。

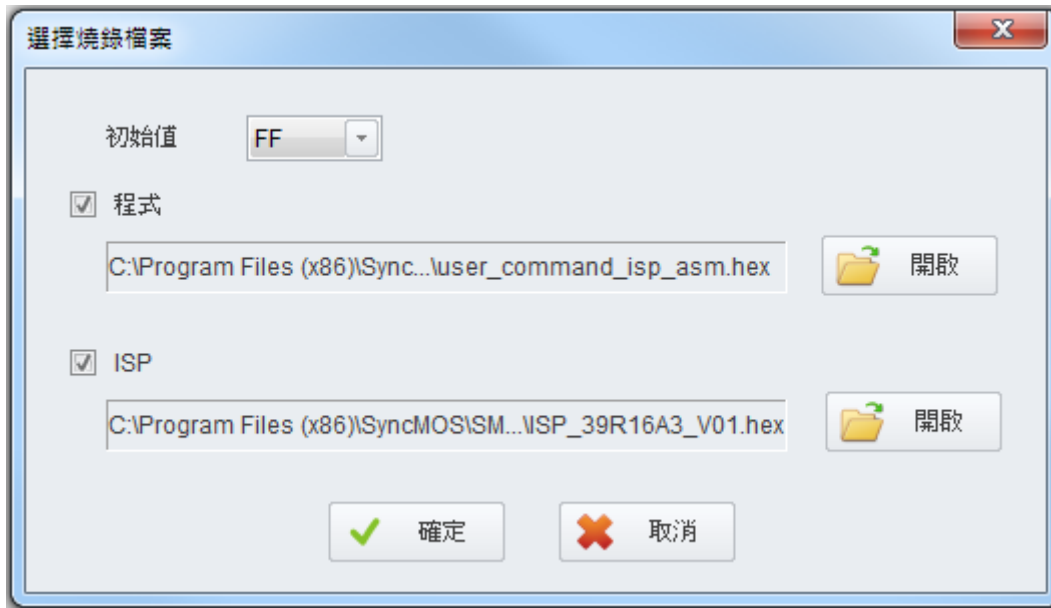
9.2 選擇燒錄檔案：

9.2.1 點按<選擇>，會跳出"選擇燒錄檔案"視窗，按<開啟>選擇您要燒錄的檔案(XXX.HEX or XXX.BIN)。



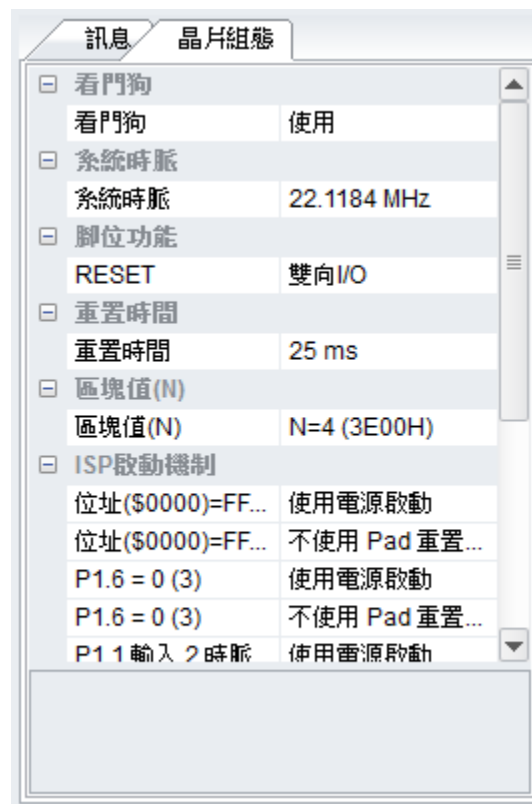


9.2.2 當使用 ICP 模式時如需同時載入 main code 與 ISP code 時，可以同時在“程式”與“ISP”各別載入。



9.3 晶片組態：

9.3.1 晶片組態介面：選擇您要設定的項目，會再燒錄時將組態設定一起燒錄進 IC 中。













9.3.2 讀取組態：選取讀取組態圖示時，可讀出晶片組態設定。

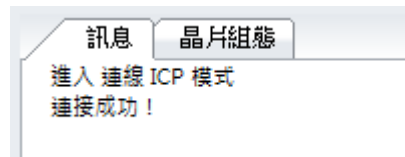


9.4 燒錄選單：

- 9.4.1  掃描 ISP 使用 ISP 模式時，會掃描出適合使用的傳輸速率。
- 9.4.2  抹除 清除 IC 的所有資料。
- 9.4.3  空白檢查 檢查 IC 的資料是否為 0xFF。
- 9.4.4  寫入 將您載入的燒錄檔案與所設定的 config 選項燒錄到 IC 裡。
- 9.4.5  比對 確認您所載入的燒錄檔案與燒錄到 IC 的資料是否正確。
- 9.4.6  保護 將 IC 的資料加密，使其無法使用讀出正確的資料。
- 9.4.7  晶片重置 當使用 ISP 時，燒錄完成後讓 IC 內部自行 Reset。
- 9.4.8  自動執行 勾選您要執行的項目，如<抹除>、<寫入>.....等。點按<自動執行>則會執行剛剛所選擇的項目。

9.5 訊息視窗：






9.5.1 點選<晶片組態>左邊的<訊息>可顯示出訊息視窗。

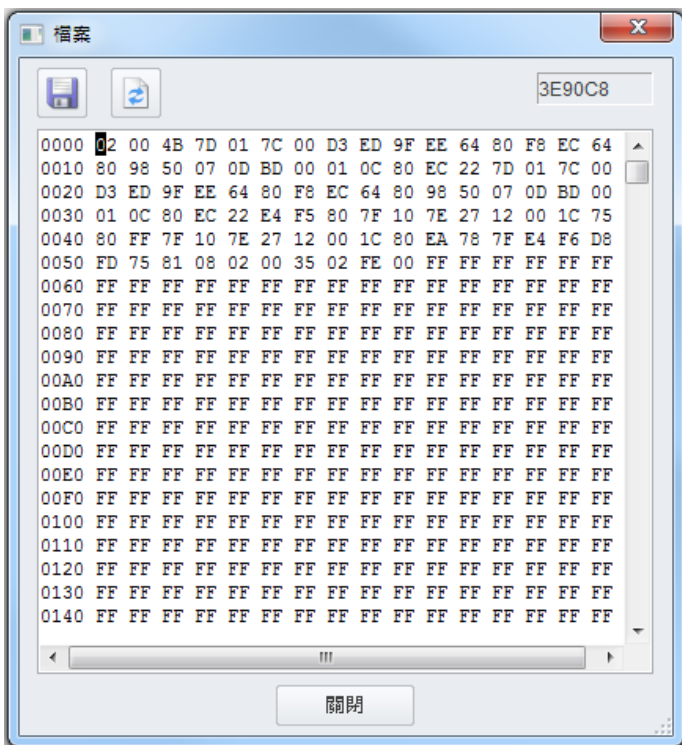


9.5.2 若要清除訊息視窗的資訊，按<清除訊息>即可清除。

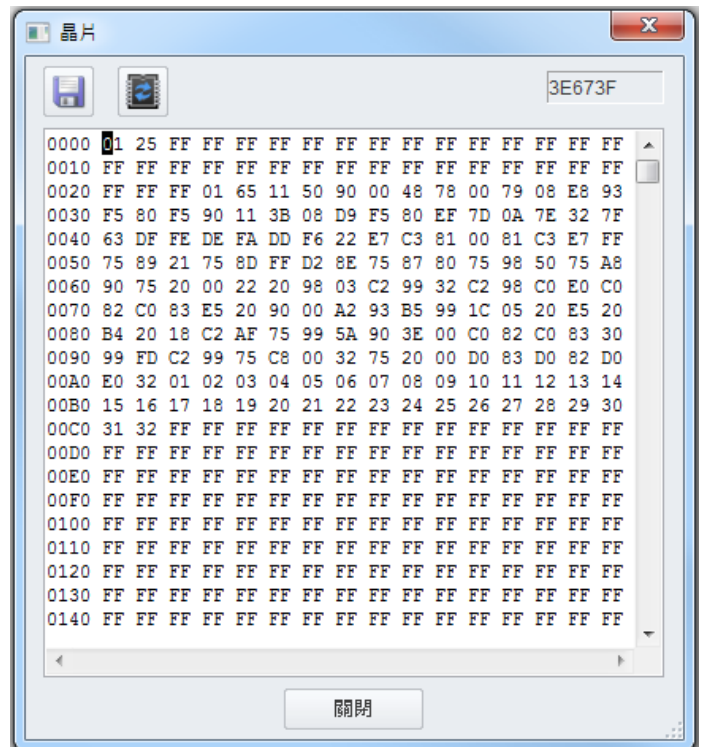


9.6 緩存區：

- 9.6.1  檔案內容：可開啟檔案緩存區視窗，可查看所載入檔案的程式碼。
- 9.6.2  晶片內容：可開啟晶片緩存區視窗，可查看所載入檔案的程式碼。
- 9.6.3  存檔：可將緩存區中的資料另存 bin 檔。
- 9.6.4  更新：更新檔案中的程式碼到緩存區視窗中。
- 9.6.5  更新：讀取 IC 中的程式碼到緩存區視窗中。



檔案緩存區視窗

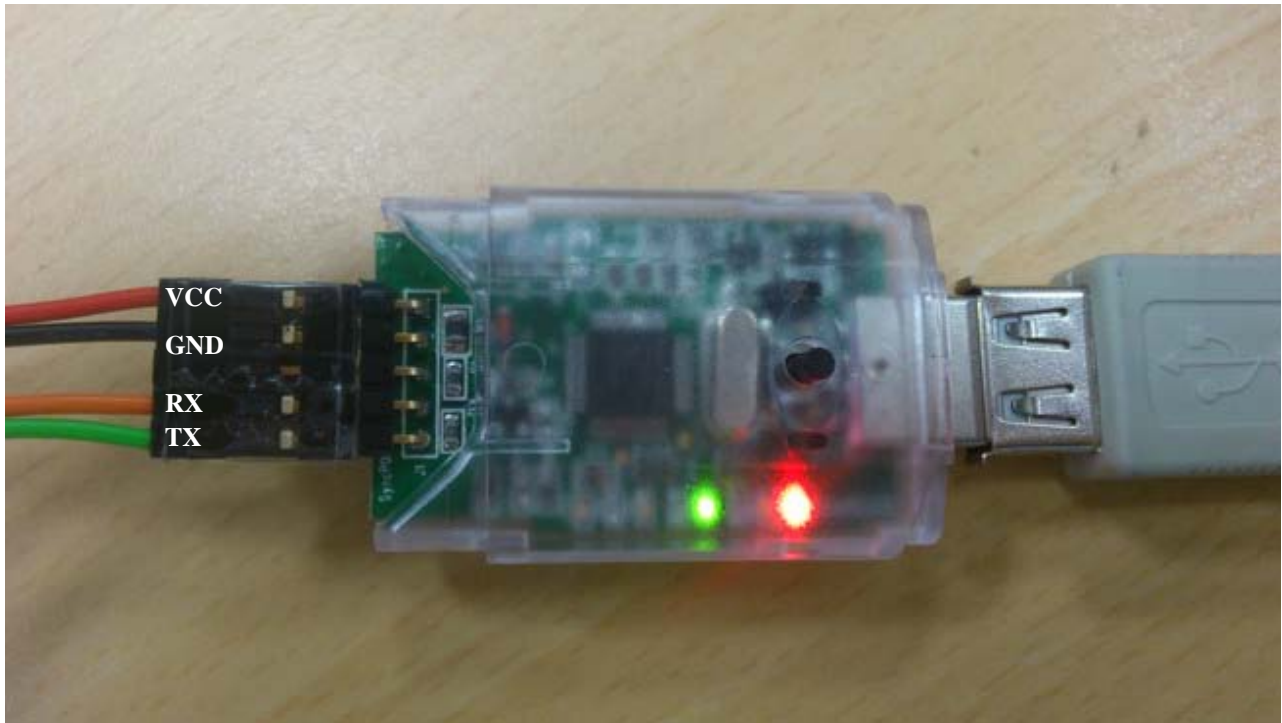


晶片緩存區視窗

10 ISP 燒錄教學

10.1 硬體連接方式：

10.1.1 當使用 ISP 連目標板時，MSM9066 需透過 Rx、Tx 與目標板連線，請將目標板的 Rx、Tx、VCC 與 GND 連接至 MSM9066。



10.1.2 ISP 燒錄需設定晶振頻率，如果連接目標板，請依照目標板所用的晶振頻率設定，若列表中無貴公司所使用的頻率，請反應給我們新茂，我們將協助加入 SMAP 支援。



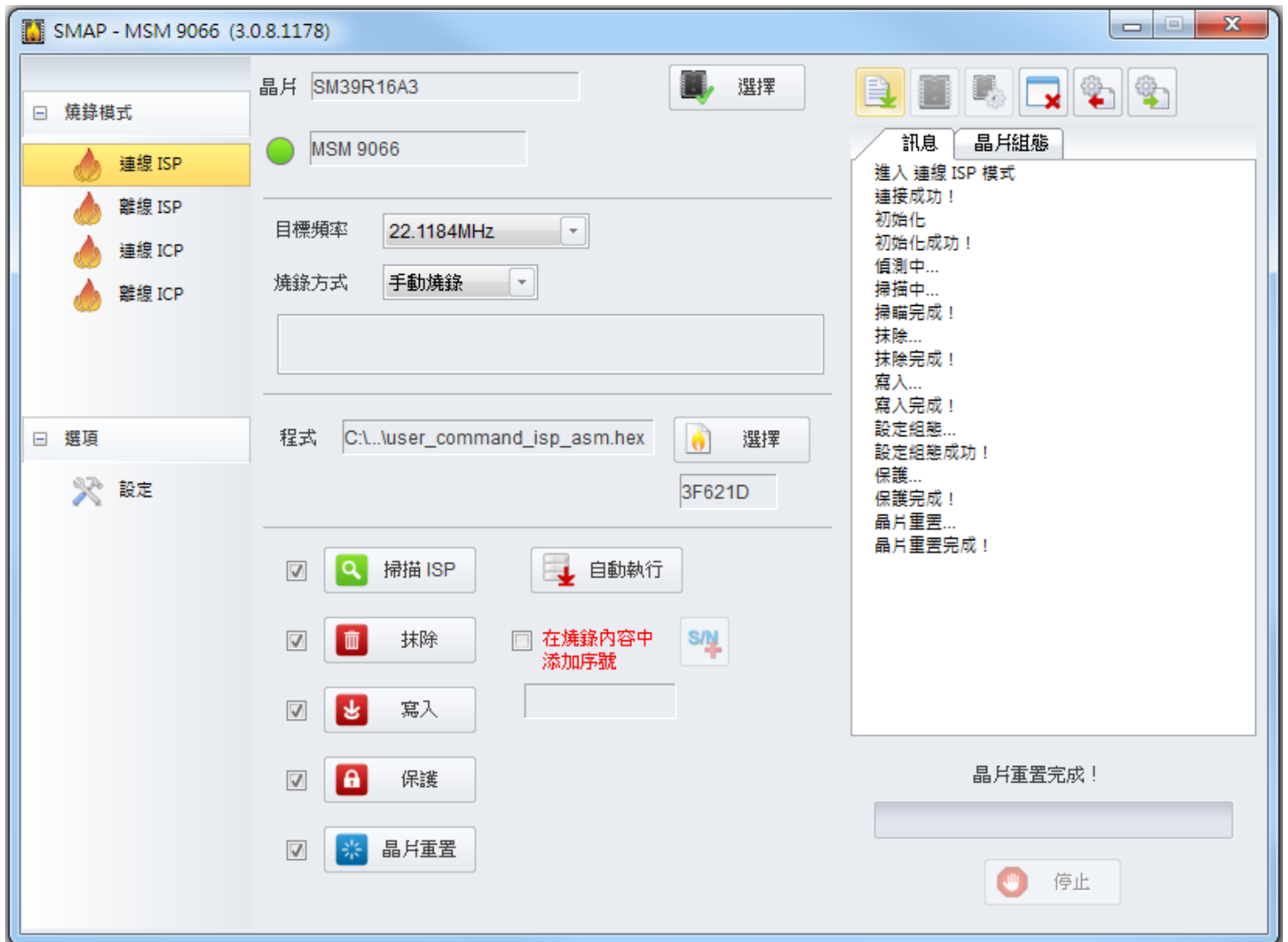


10.2 手動燒錄：

10.2.1 請自行先讓 IC 利用硬體 ISP 或軟體 ISP 方式進入 ISP 狀態。

10.2.2 設定欲燒錄的晶片、頻率，燒錄方式選擇<手動燒錄>。

10.2.3 選擇所需燒錄之檔案，然後在燒錄功能圖塊中選取要執行之功能，按下<自動執行>即可，也可以執行單步動作，直接按下所需功能之圖塊即可。



10.3 上電燒錄：只要 Power on 即可重新燒錄。SM59R04A2 與 SM59R02/03/04A1 有 P3.0 收兩個時脈的 ISP 硬體機制，則跳躍到 ISP code 進行 ISP 燒錄。操作步驟如下：

(1) 將 MCU 放置於目標板上，先不要上電。

(2) 設定欲燒錄的晶片、頻率，燒錄方式選擇<上電燒錄>。

(3) 按下<自動執行>按鈕，在停止鍵上方會顯示”偵測中”，此時再將系統板上電，即可開始燒錄。

(4) 按下<自動執行>按鈕，在停止鍵上方會顯示”偵測中”，此時再將系統板上電，即可開始燒錄。



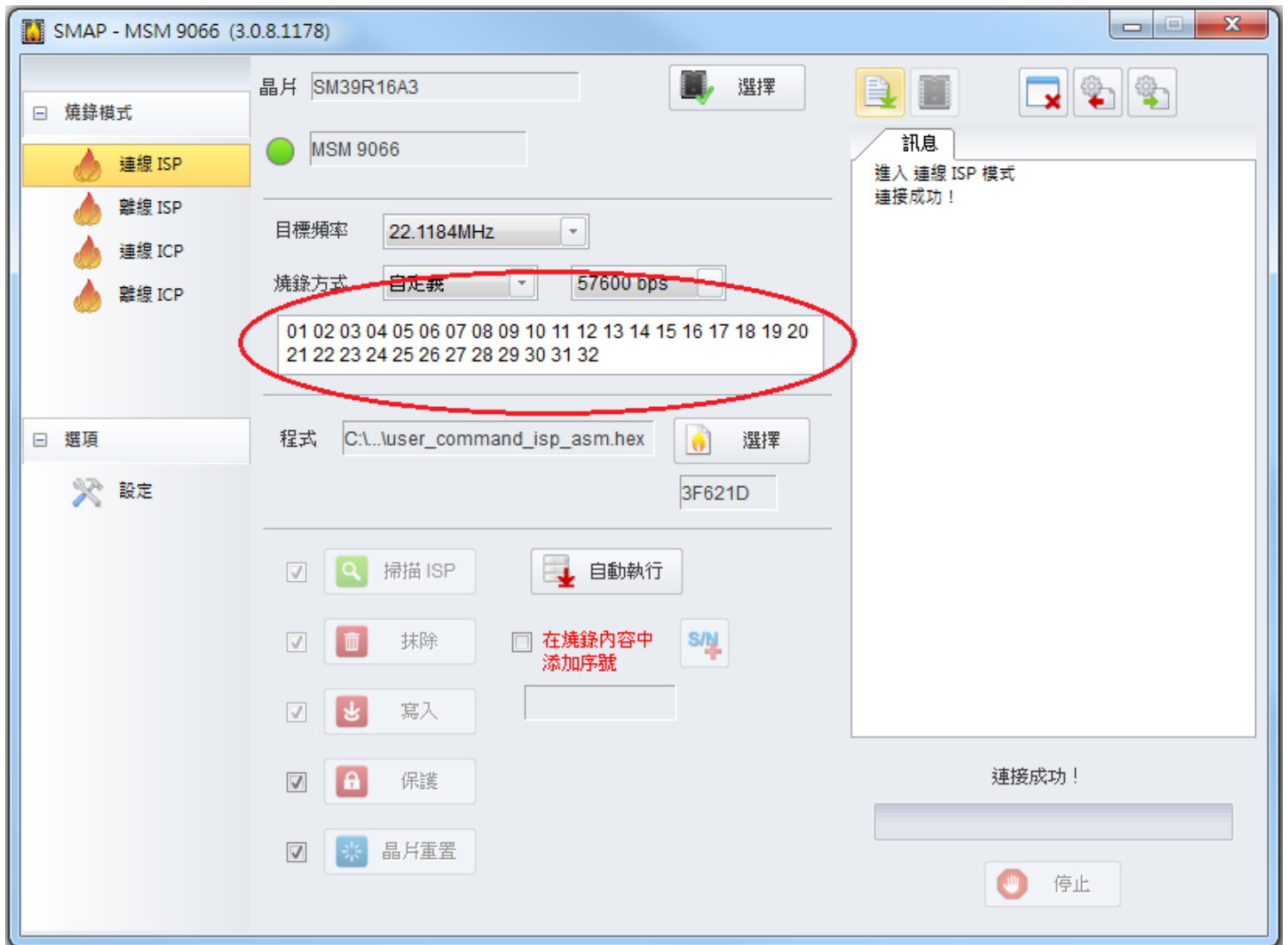
(5) 燒錄完成後在停止鍵上方會顯示”晶片重置完成！”。



10.4 自定義下載：此燒錄方式僅適用在目標板燒錄，可以讓使用者自定義 command 作為進入 ISP 燒錄的通關指令，透過 UART 連接 SMAP 軟體做連線更新。

10.4.1 操作步驟：

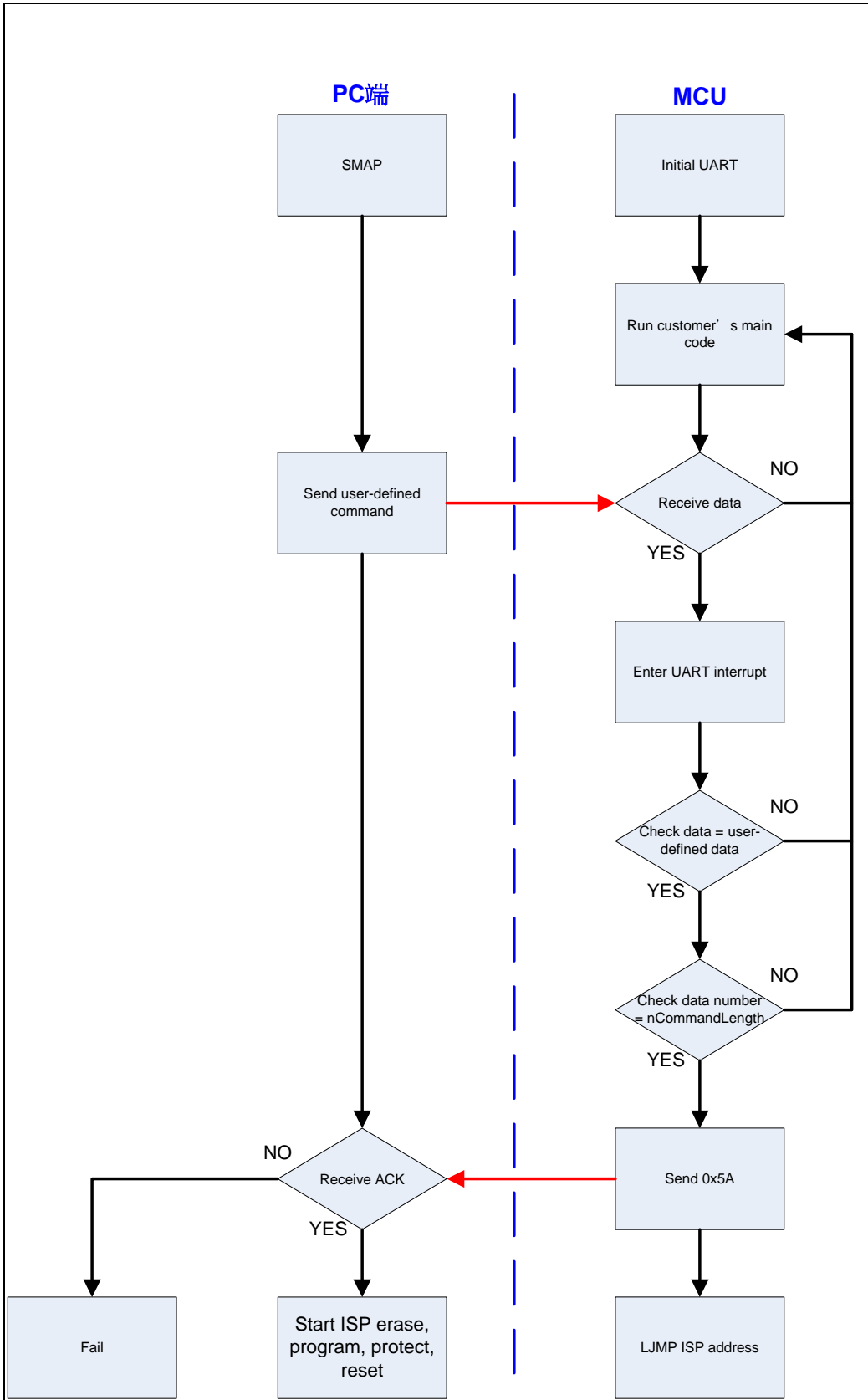
- (1) 先將範例程式及 ISP 服務程式燒進 MCU 中。
- (2) 將 MCU 放至上，與 MSM9066 連接並上電。
- (3) 選擇<自定義>，並依據客戶主程式所使用的傳輸速度設定鮑率值，因範例程式使用 115200bps，因此需選擇 115200bps。
- (4) 接著將自定義的指令輸入至視窗中，自定義的指令需配合客戶主程式中 UART 中斷的接收副程式所接受之自定義的指令，客戶可自定義的指令長度最多達 32 個位元組，資料型態為 16 進制。



(5) 按下”自動執行”按鈕，即開始燒錄，燒錄完成後在停止鍵上方會顯示”晶片重置完成！”。



10.4.2 動作流程圖：





10.4.3 範例程式(使用 SM59R04G2 為例)：

(1) Keil C 範例程式：

Description	//此範例應用包括 main.c、GoToISP_Low.A51
main.c (Main program)	<pre> #include "SM59D04G2.h" #define nCommandLength 32 #define nAckCommand 0x5A unsigned char UartCmdCount; unsigned char code CommandArray[nCommandLength]= {0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x10, 0x11, 0x12, 0x13, 0x14, 0x15, 0x16, 0x17, 0x18, 0x19, 0x20, 0x21, 0x22, 0x23, 0x24, 0x25, 0x26, 0x27, 0x28, 0x29, 0x30, 0x31, 0x32}; extern GoToISP_Low(); void delays(unsigned int mscnt) { unsigned int i; while(mscnt--) { for(i=0;i<250;i++); } } void init_UART_timer1(unsigned char BR) { SCON = 0x50; //SCON: serial mode 1, 8-bit UART, enable receive TMOD = 0x22; //TMOD: timer 1, mode 2, 8-bit reload PCON = 0x80; //SMOD = 1; TH1 = BR; //Baud:57600 fosc=22.1184MHz TR1 = 1; //timer 1 run EA = 1; //all interrupt enable ES = 1; //UART enable } void init_UART_timer2(unsigned int T2value) { RCAP2L = (char)T2value; RCAP2H = (char)(T2value>>8); T2CON = 0x34; SCON = 0x50; EA = 1; //all interrupt enable ES = 1; //UART enable } void main(void) { init_UART_timer2(0xFFFA);//choise Timer 1 or Timer 2 as baud rate generator //init_UART_timer1(0xFF); //SyncMOS_GoToISP via UART //TH1 = 0xFF; //12T, 3.579MHz(19200), 11.0592MHz(57600) // 6T, 3.579MHz(38400), 11.0592MHz(115200) </pre>



	<pre>//TH1 = 0xFE; //12T, 14.31818MHz(38400), 22.1184MHz(57600), 14.7456MHz(38400) // 6T, 22.1184MHz(115200) //TH1 = 0xFD; //12T, 4MHz(7200), 8MHz(14400), 8.192MHz(14400), 16MHz(28800) // 6T, 4MHz(14400),8MHz(28800), 8.192MHz(28800), 16MHz(57600) //TH1 = 0xFC; // // 6T, 14.31818MHz(38400), 14.7456MHz(38400) //TH1 = 0xFB; //12T, 18.432MHz(19200) // 6T, 18.432MHz(38400) //TH1 = 0xF7; //12T, 24.576MHz(14400), 25MHz(14400) // 6T, 24.576MHz(28800), 25MHz(28800) //TH1 = 0xF5; //12T, 20MHz(9600) // 6T, 20MHz(19200) //TH1 = 0xF3; //12T, 6MHz(2400), 12MHz(4800), 24MHz(9600) // 6T, 6MHz(4800), 12MHz(9600), 24MHz(19200) while(1) { P0 = 0xF0; delayms(300); P0 = 0x0F; delayms(300); // P2 = 0xF0; delayms(300); // P2 = 0x0F; delayms(300); } void serial(void) interrupt 4 { if(RI) { if(SBUF == CommandArray[UartCmdCount]) UartCmdCount++; else UartCmdCount=0; RI = 0 ; if(UartCmdCount==nCommandLength) { IE = 0x00; //disabling interrupt SBUF = nAckCommand; while(!TI); TI = 0; T2CON = 0x00; //disabling T2 because using T1 baud rate generator in ISP code GoToISP_Low(); //LJMP 3E00H ISP code address } } else TI=0; } }</pre>
<p>GoToISP_Low. A51</p>	<pre>; GoToISP_Low.A51 NAME GoToISP_Low ?PR?GoToISP_Low?GOTOISP_LOW SEGMENT CODE PUBLIC GoToISP_Low RSEG ?PR?GoToISP_Low?GOTOISP_LOW USING 0 GoToISP_Low: MOV DPTR,#3E00H ;LJMP 3E00H ISP code address</pre>



	PUSH DPL PUSH DPH RETI END
--	-------------------------------------

(2) 組合語言範例程式：

Description	
Main program	<pre> nCommandLength EQU 32 ;setting command length by user UARTCmdCount EQU 20H T2CON EQU 0C8H RCAP2L EQU 0CAH RCAP2H EQU 0CBH ORG 0000H AJMP ISP_Entry ORG 0023H AJMP UART_INTERRUPT ISP_Entry: ACALL SyncMOS_GoToISP ;initial setting ;;; ;;; main code start ;;; /* LED display demo */ MOV DPTR,#TABLE_01 ;DPTR point to data area START: MOV R0,#0 ; MOV R1,#8 ;8 datas LOOP: MOV A,R0 ;put R0 connect to A MOVC A,@A+DPTR ;use indirect address get data, and put into A MOV P0,A ;put ACC to Port 0 MOV P1,A ;put ACC to Port 1 ACALL DELAY100MS ;delay 0.1 second INC R0 ;R0 point to next data DJNZ R1,LOOP ;if R1 != 0 then jump to LOOP SJMP START ;if R1=0 then jump to START ;; dealy subroutine DELAY100MS: MOV R5,#10 ;Dealy 10x10ms=100ms DELAY10MS: MOV R6,#50 ;delay time 10ms @ 11.0592MHz DELAY01: MOV R7,#99 DJNZ R7,\$; Inner loop delay DJNZ R6,DELAY01 ; DJNZ R5,DELAY10MS RET ;;; TABLE_01: DB 11100111B DB 11000011B DB 10000001B </pre>



```

DB 0000000B
DB 10000001B
DB 11000011B
DB 11100111B
DB 11111111B
...
;;;
;;; main code End
...
;;;
...
;;; SyncMOS_GoToISP Subroutine
...
;;;
SyncMOS_GoToISP:

;;;=====UART Timer1 setting=====
;MOV TMOD,#00100001B ;Timer 1 is mode1 is mode 2, Timer 0 is mode 1
...
;MOV TH1,#0FFH ;baud rate, 12T, 3.579MHz(19200), 11.0592MHz(57600)
; ; 6T, 3.579MHz(38400), 11.0592MHz(115200)
;MOV TH1,#0FEH ;baud rate, 12T, 14.31818MHz(38400), 22.1184MHz(57600),
14.7456MHz(38400)
; ; 6T, 22.1184MHz(115200)
;MOV TH1,#0FDH ;baud rate, 12T, 4MHz(7200), 8MHz(14400), 8.192MHz(14400),
16MHz(28800)
; ; 6T, 4MHz(14400),8MHz(28800), 8.192MHz(28800),
16MHz(57600)
;MOV TH1,#0FCH ;baud rate,
; ; 6T, 14.31818MHz(38400), 14.7456MHz(38400)
;MOV TH1,#0FBH ;baud rate, 12T, 18.432MHz(19200)
; ; 6T, 18.432MHz(38400)
;MOV TH1,#0F7H ;baud rate, 12T, 24.576MHz(14400), 25MHz(14400)
; ; 6T, 24.576MHz(28800), 25MHz(28800)
;MOV TH1,#0F5H ;baud rate, 12T, 20MHz(9600)
; ; 6T, 20MHz(19200)
;MOV TH1,#0F3H ;baud rate, 12T, 6MHz(2400), 12MHz(4800), 24MHz(9600)
; ; 6T, 6MHz(4800), 12MHz(9600), 24MHz(19200)
;SETB TR1 ;enable Timer 1
;MOV PCON,#1000000B ;set SMOD = 1
...
;;;=====
...
;;;
;;;=====UART Timer2 setting=====
;choise Timer 1 or Timer 2 as baud rate generator
MOV RCAP2H,#0FFH
MOV RCAP2L,#0FAH
MOV T2CON,#34H
...
;;;=====
MOV SCON,#01010000B ;serial mode 1, REN=1, TI=0, RI=0
MOV IE,#90H ;enable UART interrupt
MOV UARTCmdCount,#00H
RET
UART_INTERRUPT:
JB RI,RX ; if RI=1 jump RX,else clear TI
CLR TI
RETI
RX:
CLR RI

```



	<pre>PUSH ACC PUSH DPL PUSH DPH MOV A,UARTCmdCount MOV DPTR,#COMMAND_TABLE MOVC A,@A+DPTR CJNE A,SBUF,CLR_UARTCmdCount INC UARTCmdCount MOV A,UARTCmdCount CJNE A,#nCommandLength,RETURN CLR EA MOV SBUF,#5AH ;transmit ack MOV DPTR,#3E00H ;LJMP 3E00H ISP code address PUSH DPL PUSH DPH JNB TI,\$ CLR TI MOV T2CON,#00H ;disabling T2 because using T1 baud rate generator in ISP code RETI CLR_UARTCmdCount: MOV UARTCmdCount,#00H RETURN: POP DPH POP DPL POP ACC RETI COMMAND_TABLE: DB 01H,02H,03H,04H,05H,06H,07H,08H ;to define command by user DB 09H,10H,11H,12H,13H,14H,15H,16H DB 17H,18H,19H,20H,21H,22H,23H,24H DB 25H,26H,27H,28H,29H,30H,31H,32H END</pre>
--	---

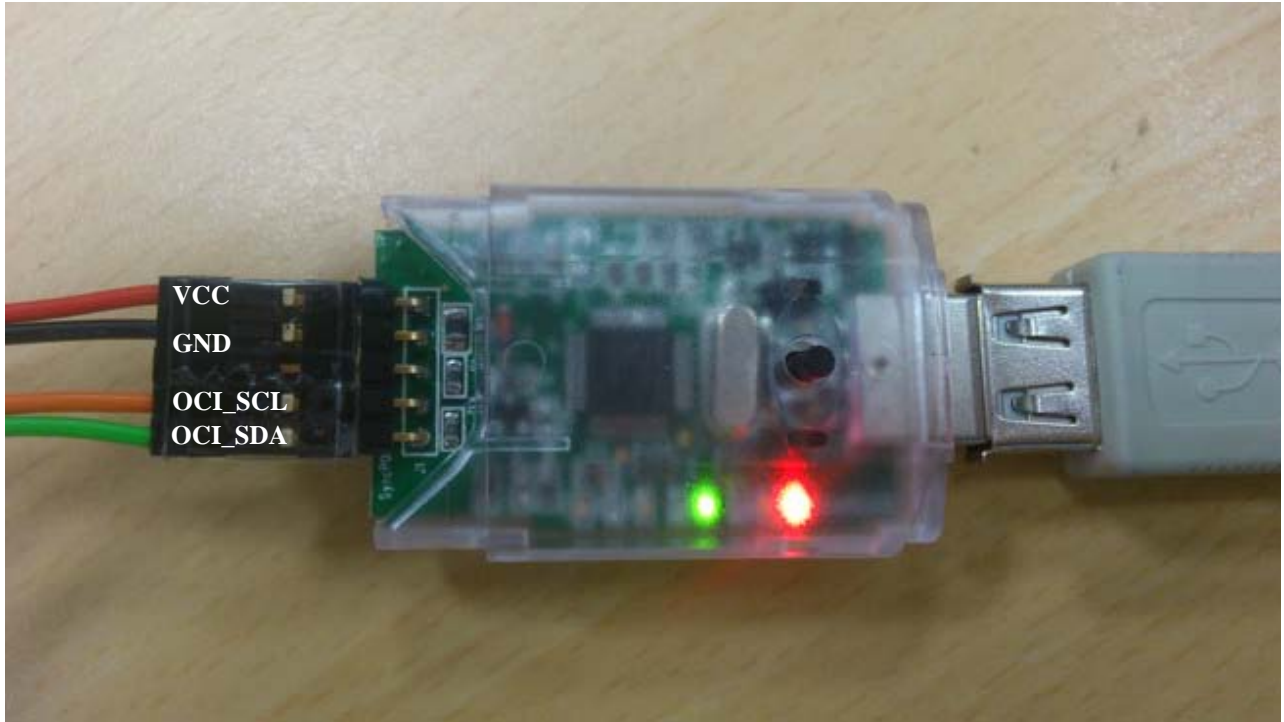
備註：

- ※ 相關範例程式已附在 SMAP 安裝目錄下 C:\Program Files\SyncMOS\SMAP\OnlineUpdate\User-defined



11 ICP 燒錄教學

11.1 MSM9066 需透過 OCI_SCL、OCI_SDA 與目標板連線。請將目標板的 OCI_SCL、OCI_SDA、VCC 與 GND 連接至 MSM9066。設定好所需的項目，點選<自動執行>，即可使用 ICP 燒錄。



11.2 當 OCI_SCL、OCI_SDA 與 Reset pin 皆被組態為 GPIO，會影響 ICP 燒錄時無法入 ICP mode，只有 IC 剛 Power on 可進入 ICP mode。在此情況下使用 MSM9066 請依照下列操作步驟：

- (1) 先將目標板電源關閉，並將目標板的 OCI_SCL、OCI_SDA、VCC 與 GND 連接至 MSM9066。
- (2) 按下<自動執行>時會顯示”偵測中”。



- (3) 這時候再開啟目標板電源，燒錄完成後會顯示完成資訊。





12 離線燒錄教學

12.1 離線 ICP

12.1.1 設定好所需的項目，點選<自動執行>，MSM9066 便會將所需燒錄的 code 存至 MSM9066 上的 flash 中，當執行成功後，停止鍵上方會顯示”傳送成功!”。



12.1.2 拔掉 USB 線，連接目標板並按下按鈕，便可使用離線燒錄的功能，燒錄過程中黃燈 LED 會不斷閃爍，當黃燈熄滅且綠燈亮時，即代表離線燒錄作業完成。若您的 MCU 已把 OCL_SCL 與 OCL_SDA 設定為 GPIO，則必須要用 ICP 上電燒錄方式，先接好連接線，目標板上電即可開始燒錄。

12.2 離線 ISP

12.2.1 設定好所需的項目，點選<自動執行>，MSM9066 便會將所需燒錄的 code 存至 MSM9066 上的 flash 中，當執行成功後，停止鍵上方會顯示”傳送成功!”。



12.2.2 拔掉 USB 線，連接目標板並按下按鈕，便可使用離線燒錄的功能，燒錄過程中黃燈 LED 會不斷閃爍，當黃燈熄滅且綠燈亮時，即代表離線燒錄作業完成。

12.3 離線 ISP 上電燒錄：

12.3.1 燒錄方式需設定為"上電燒錄"，將欲燒錄 main code 以離線 ISP 方式將 code 儲存至 MSM9066 中。

12.3.2 先關掉目標板電源，拔掉 USB 線，如同一般 ISP 燒錄在 MCU 目標板連接方式連接 Rx,Tx,GND,VCC。

12.3.3 開啟電源，便會自動執行離線燒錄，燒錄過程中黃燈 LED 會不斷閃爍，當黃燈熄滅且綠燈亮時，即代表離線燒錄作業完成。