IAP library note

SQ7617

Rev 1.0

內容

參照圖例:

圖表 1-1 版本記錄

Version	Approved date	Description
R1.0	2022/4/8	First release

1 函式庫

1.1 函式庫說明

A. **IAP** 是 **In Application Programming** 的首字母縮寫，**IAP** 是用戶自己的程序在運行過程中對 **User Flash** 的部分區域進行編寫，目的是為了在產品發布後可以方便地通過預留的通信口對產品中的程序進行更新升級。

B. 設計者需要實現 **IAP** 功能時，需要在程序運行時執行更新程序，設計者只需使用此 **IAP Library** 即可實現 **IAP** 功能。

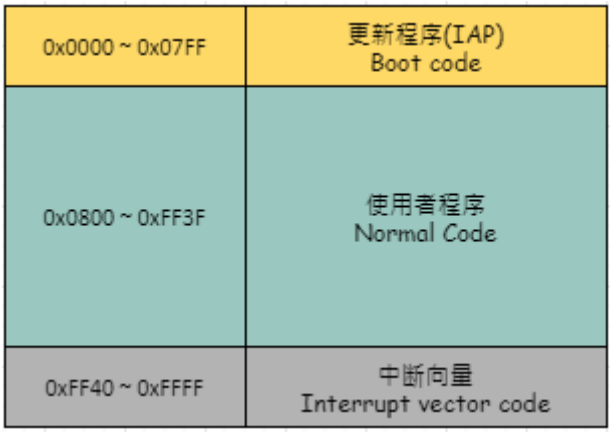
C. **IAP Library** 是使用通用異步接收器**(UART)**接收程序碼或數據，執行對原始代碼的更新；正常啟動代碼**(normal mode)**代碼才是真正的功能代碼。這兩部分的程序碼都同時燒錄在 **User Flash** 中，當晶片上電後，首先是 **IAP Library** 程序開始運行，如下操作：

步驟 **1:** 檢查是否需要對原代碼進行更新
步驟 **2:** 如果不需要更新則轉到步驟 **4**
步驟 **3:** 執行更新操作
步驟 **4:** 轉到正常啟動代碼執行

D. **User Flash** 編寫位置

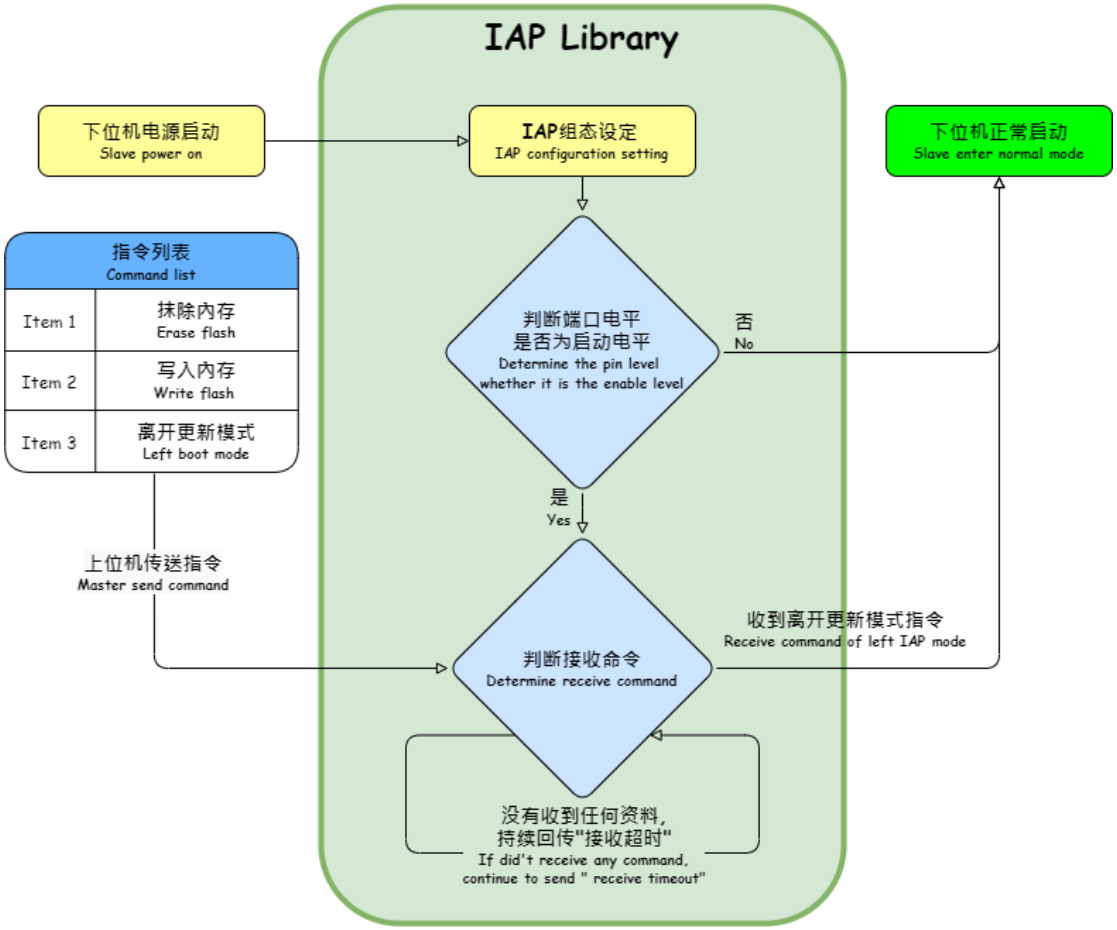
I. **IAP boot library** 編碼位置 **: 0x0000 ~ 0x07FF**
II. **Normal mode** 編碼位置 **: 0x0800 ~ 0xFF3F**
III. **Interrupt vector** 中斷向量編碼位置 **: 0xFF40 ~ 0xFFFF**

圖表 1-1 User Flash 編寫位置圖



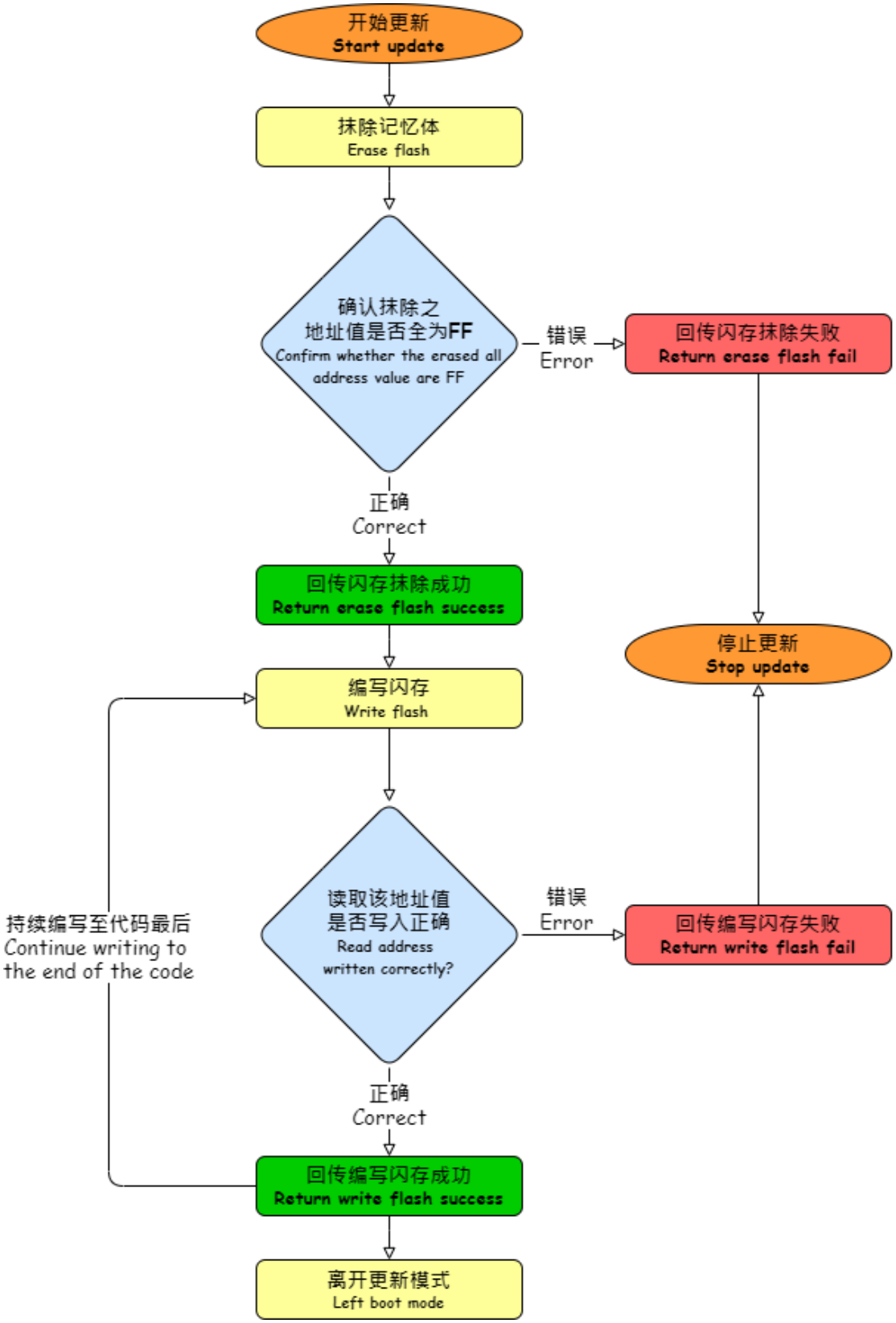
1.2 架構說明

圖表 1-2 IAP 架構圖



1.3 更新流程

圖表 1-3 更新流程圖



1.4 IAP 指令內容

圖表 1-4 指令表

	指令 Command	执行正确回传值 Execute correct feedback value
抹除记忆体 Erase flash	0x00 0x00 0x00 0x80 0x80	0x03 0x02 0xFB
离开更新模式 Left boot mode	0x00 0x00 0x00 0x01 0xFF	0x03 0x04 0xF9
记忆体参数写入 Write flash		0x03 0x03 0xFA

档案资料转换成传送形式说明 H16 file data format (Ex: 02E50000345E00)					
flash data length	flash address (9~16 bit)	Flash address (1~8 bit)	Write command	Write value	check sum
1 byte	1 byte	1 byte	1 byte	Same as flash data length	1 byte
0x02	0xE5	0x00	0x00	0x34,0x5E	0x00

1.5 IAP 回覆格式及內容

圖表 1-5 回覆格式

回传值格式表 feedback format		
长度 Data length	回传值 return value	校验和 check sum
0x03	0x05	0xF8

校驗和公式: **0xFF - (**信息長度**) - (**回傳值**) + 1**。

範例: **0xF8 = 0xFF – 0x03 – 0x05 + 1**

圖表 1-6 回傳值內容說明

回传值定义表 Return value definition table	
0x00	离开开机模式 Leaving boot mode
0x01	UART等待接收资料 UART waiting receive data
0x02	抹除闪存成功 Slave erase flash success
0x03	编写闪存成功 Slave write flash success
0x04	离开更新模式 Slave out of boot mode
0x05	串口接收超时,自动重新接收进程 UART receive time out,auto renew the receive process
0x06	串口接收到未知指令 UART receive unknown command
0x07	串口接收到不合法地址 UART receive illegal address
0x08	奇偶校验错误 UART error by parity
0x09	逐帧错误 UART error by frame
0x0A	溢出错误 UART error by overflow
0x0B	串口接收到超过 21 byte的资料(因 H16 档资料列最长只有 21 byte) UART receive data over than 21 byte(each H16 data, Max is 21 byte)
0x0C	记忆体长度错误(记忆体编程长度最长只有 16 byte 的资料) UART receive program data over than 16 byte(each flash data, Max is 16 byte)
0x0D	UART 接收端索引溢位 UART receive index overflow
0x0E	校验和错误 UART receive data, check sum error
0x0F	抹除闪存失败 Slave erase flash fail
0x10	编写闪存失败 Slave write flash fail

2 專案設置

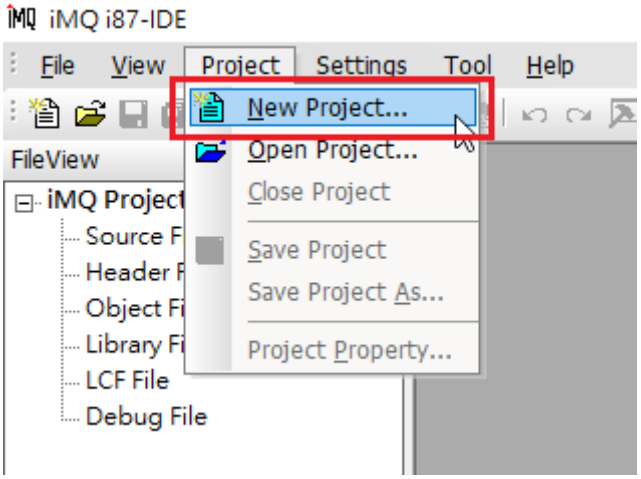
2.1 建立新專案

A. 開啟 **iMQ i87 IDE** 建立 **SQ7617** 開發專案。

圖表 2-1 點選 iMQ i87-IDE

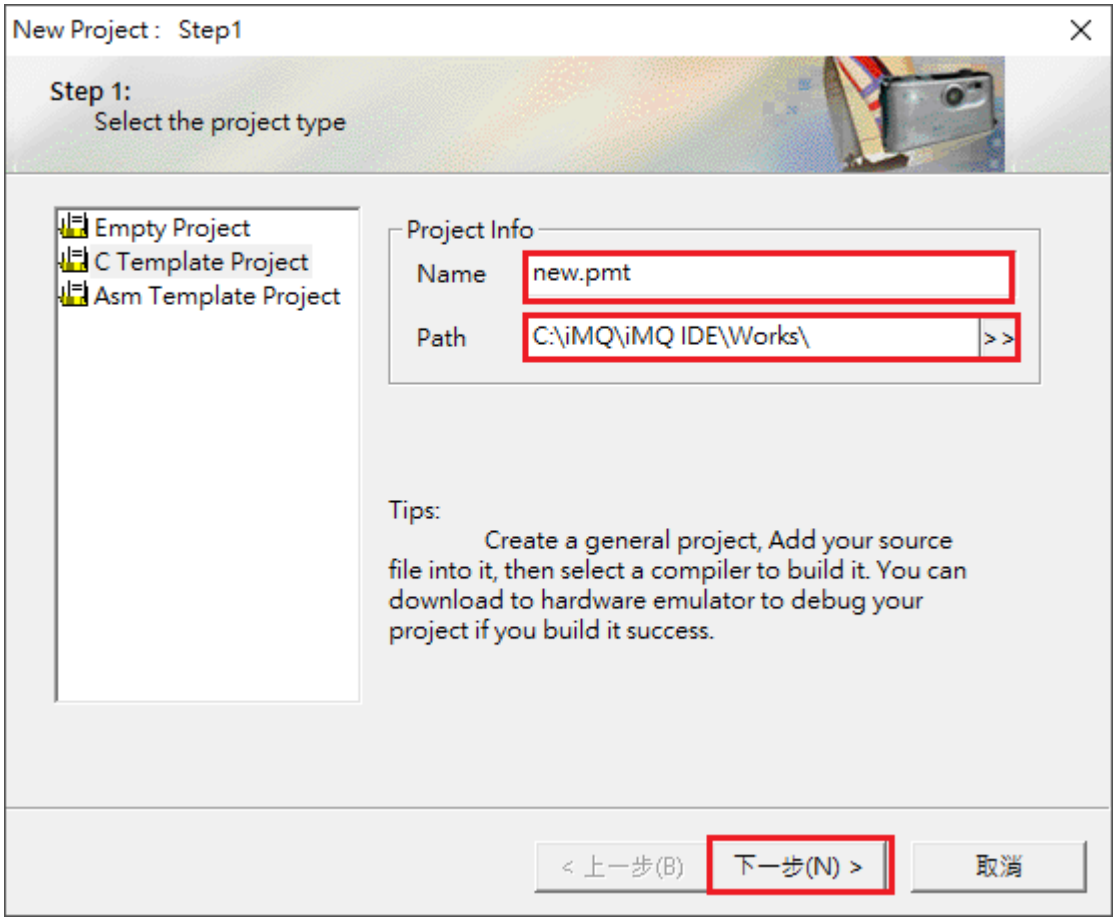
B. 點選專案**(Project)** -> 新專案**(New Project)**。

圖表 2-2 建立新專案



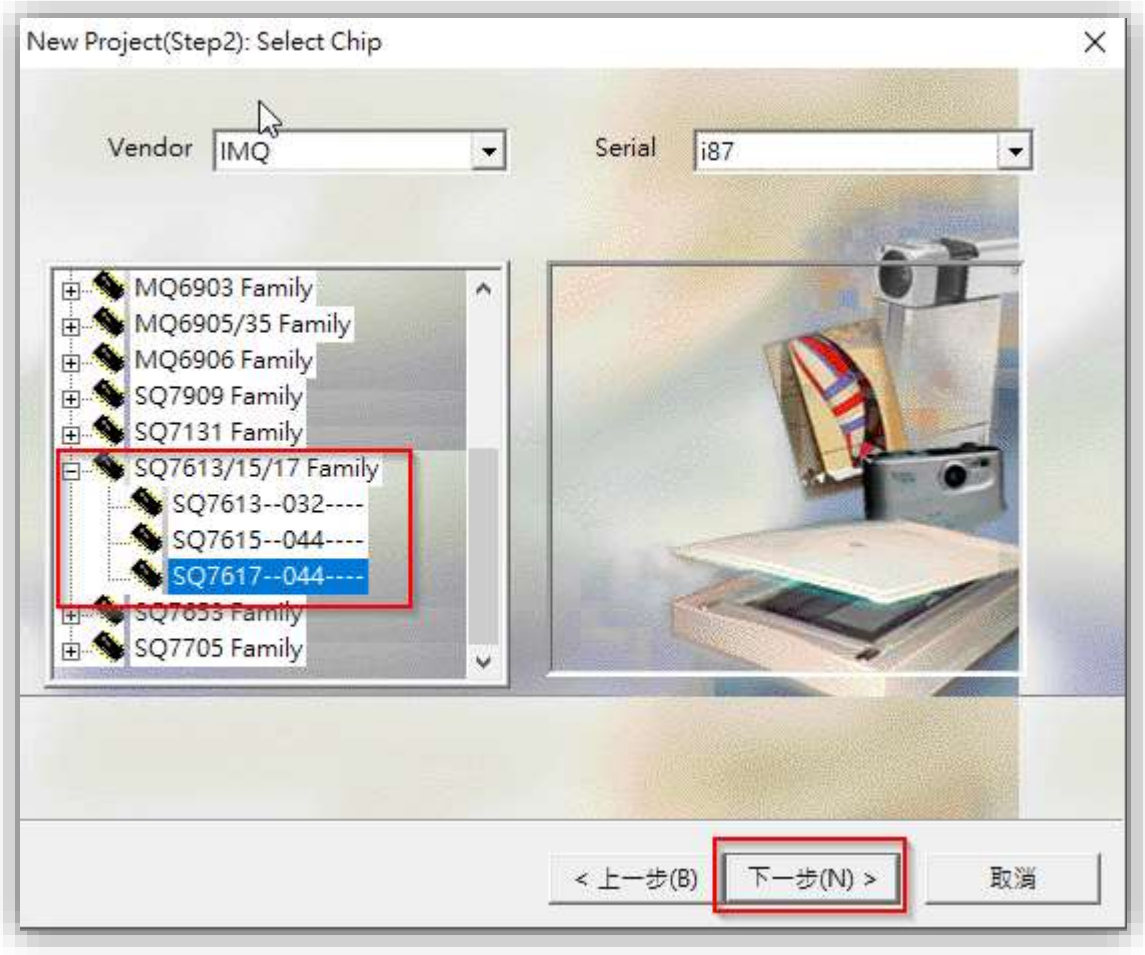
C. 編輯專案名稱(Name)及專案路徑(Path)，編輯完成點選下一步。

圖表 2-3 編輯專案名稱及路徑



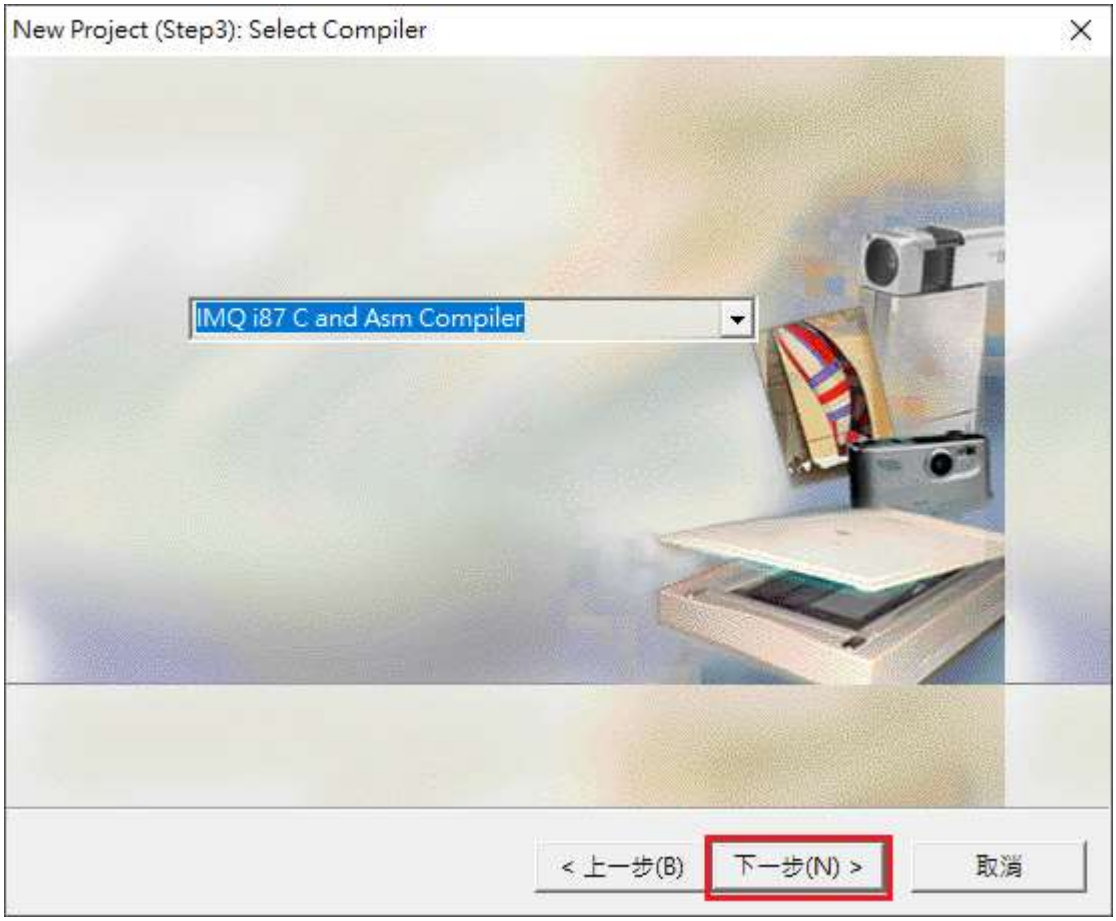
D. 選擇相對應晶片 **SQ7617—044—**，點選下一步。

圖表 2-4 選擇對應晶片



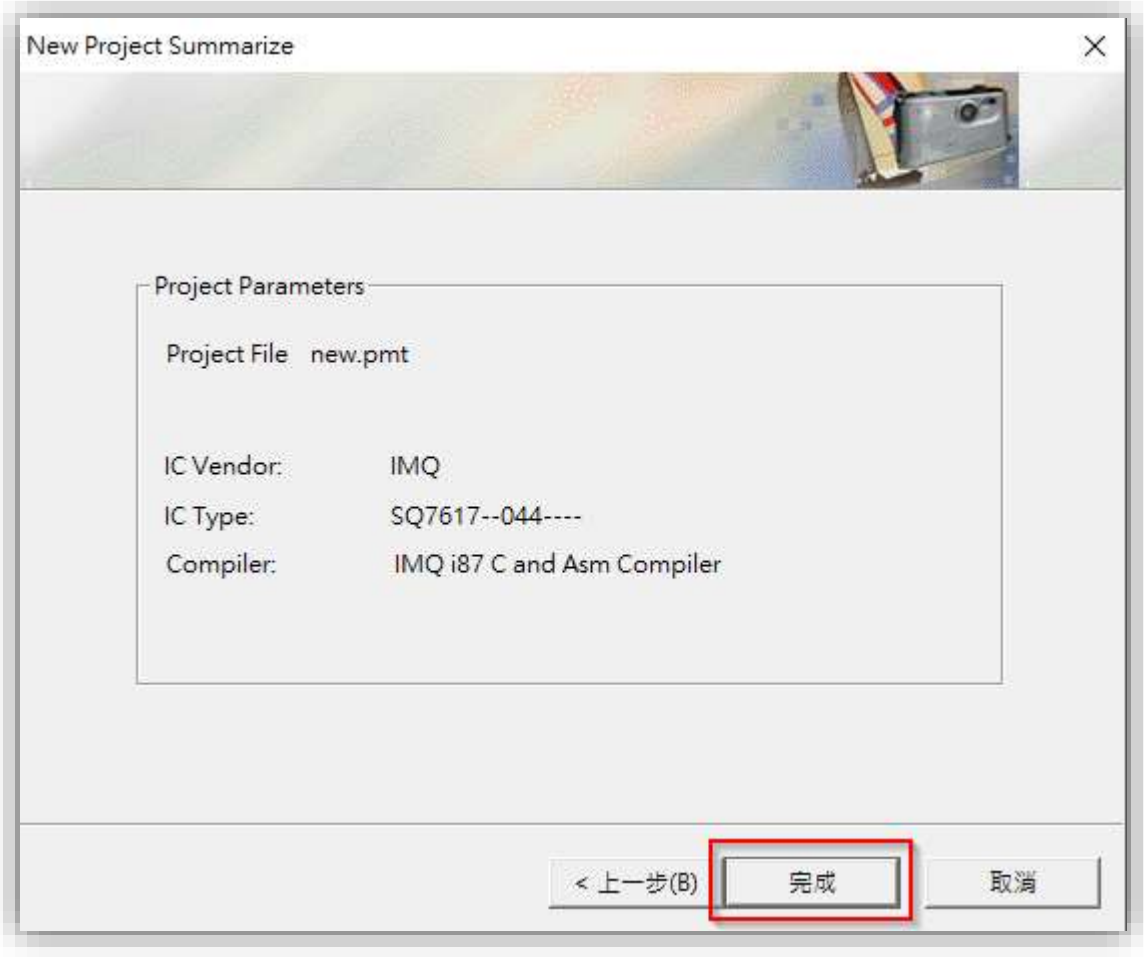
E. 選擇編譯器**(Compiler)** ，點選下一步。

圖表 2-5 選擇編譯器(Compiler)



F. 點選完成即建立完成專案。

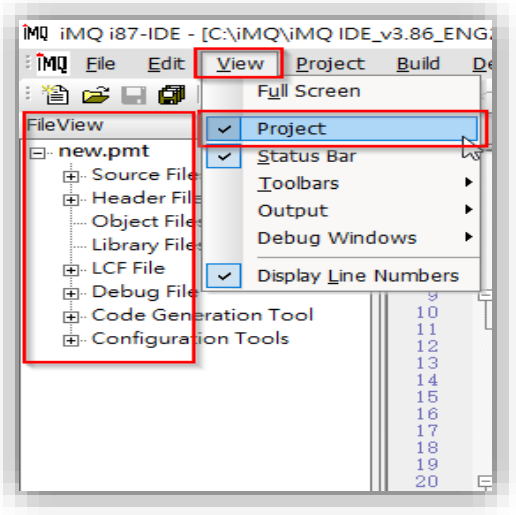
圖表 2-6 新專案建立完成



2.2 專案加入函式庫檔案

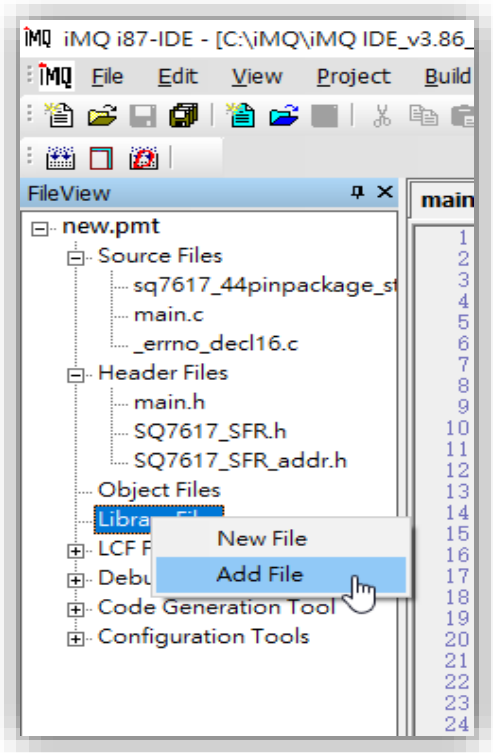
A. ***iMQ IDE*** 選單欄位**(Menu bar) ->** 視圖**(View) ->** 專案**(project)**選取後會出現檔案視圖**(File view)**，如下圖：

圖表 2-7 檔案視圖顯示



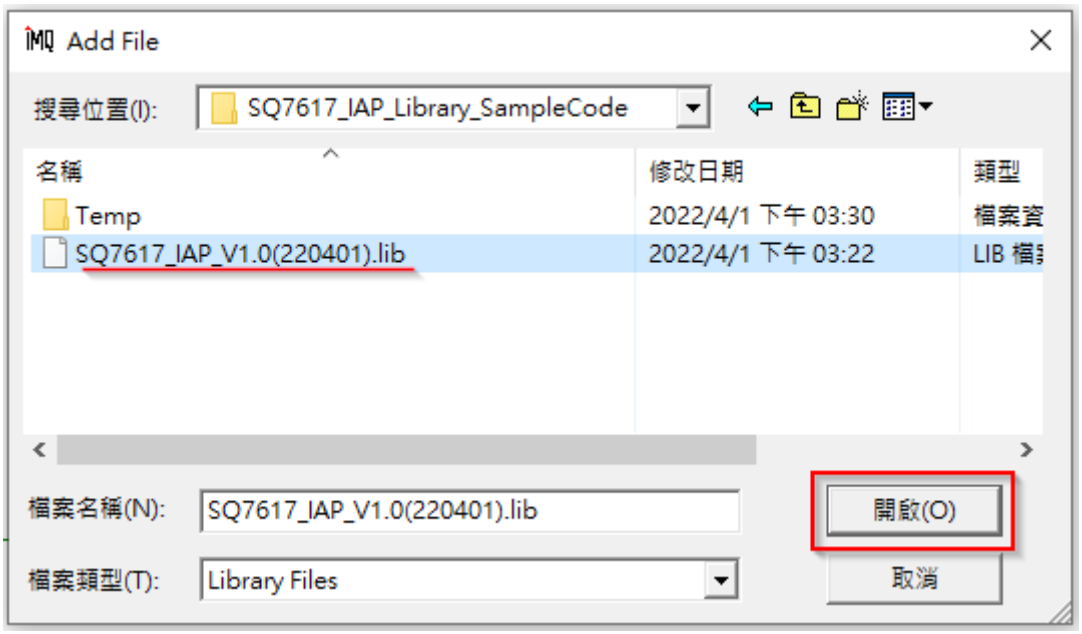
B. 在檔案視圖中**(File view)**對 **Library File** 點擊滑鼠右鍵 **->** 增加檔案**(Add File)**，如下圖：

圖表 2-8 加入函式庫檔案



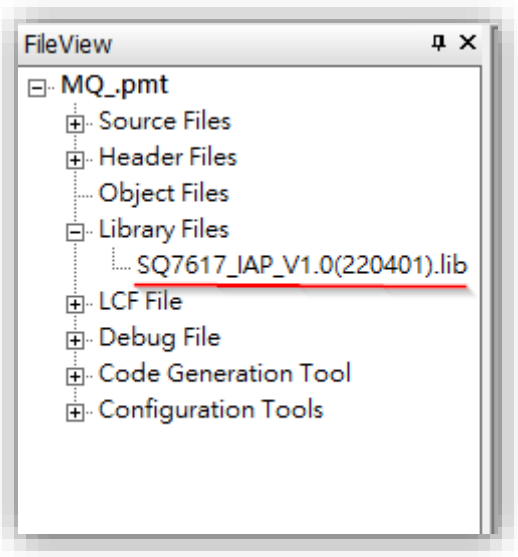
C. 選擇與 **SQ7617** 相對應之函式庫檔案**(.lib)**並開啟，如下圖：

圖表 2-9 選擇函式庫檔案



D. 確認函式庫檔案加入專案成功，如下圖：

圖表 2-10 確認專案內已加入函式庫檔案



2.3 主程式增加編碼及修改編碼

A. 增加下列編碼至主程式**(main.c)**內，作為 **IAP** 函式庫變數參考的位址及設定值並將此編譯後的代碼編譯在關聯指令檔**(lcf file)**指定的區域位址。

圖表 2-11 變數參考設定

```
/* ---------------------- IAP相关(不要修改此处 code) -------------------------- */
#pragma section code    c_value
void IAP_Init()
{
    // Memory address related
    IAP_Dev_define.norBootAddr = NORMAL_BOOT_ADDRESS;

    // BOOT PIN select
    IAP_Dev_define.ptrBootPinFC = IAP_BOOT_PIN_FUNCTION_ADR;
    IAP_Dev_define.ptrBootPinPU = IAP_BOOT_PIN_PULL_UP_ADR;
    IAP_Dev_define.ptrBootPinPD = IAP_BOOT_PIN_PULL_DOWN_ADR;
    IAP_Dev_define.ptrBootPinCR = IAP_BOOT_PIN_CONTROL_ADR;
    IAP_Dev_define.ptrBootPinPRD    = IAP_BOOT_PIN_INPUT_ADR;
    IAP_Dev_define.valBootPinBit    = IAP_BOOT_PIN_SELECT;
    IAP_Dev_define.valBootDefaultHL = IAP_BOOT_GPIO_DEFAULT_HIGH_LOW;
    IAP_Dev_define.valBootEnableHL  = IAP_BOOT_GPIO_ENABLE_HIGH_LOW;

    // UART select
    IAP_Dev_define.ptrUartPinCR = IAP_UART_CR_ADR;
    IAP_Dev_define.ptrUartPinFC = IAP_UART_FC_ADR;
    IAP_Dev_define.valUartTxPin = IAP_UART_TX_PIN;
    IAP_Dev_define.valUartRxPin = IAP_UART_RX_PIN;

    IAP_Dev_define.ptrUartTxBuf = IAP_UART_TDBUF_ADR;
    IAP_Dev_define.ptrUartRxBuf = IAP_UART_RDBUF_ADR;
    IAP_Dev_define.ptrUartSR    = IAP_UART_SR_ADR;
    IAP_Dev_define.ptrUartDR    = IAP_UART_DR_ADR;
    IAP_Dev_define.ptrUartCR1   = IAP_UART_CR1_ADR;
    IAP_Dev_define.ptrUartCR2   = IAP_UART_CR2_ADR;

    IAP_Dev_define.ptrPxPU      = IAP_PxPU_ADR;
}
#pragma section code
/* ---------------------- IAP相关 End -------------------------- */
```

B. 修改主程式(main.c)內中斷向量表定義 **STARTUP**，修改為 **iap_boot**，目的為系統啟動時必須先執行 **IAP** 流程，確認是否執行更新流程。

圖表 2-12 中斷向量修改

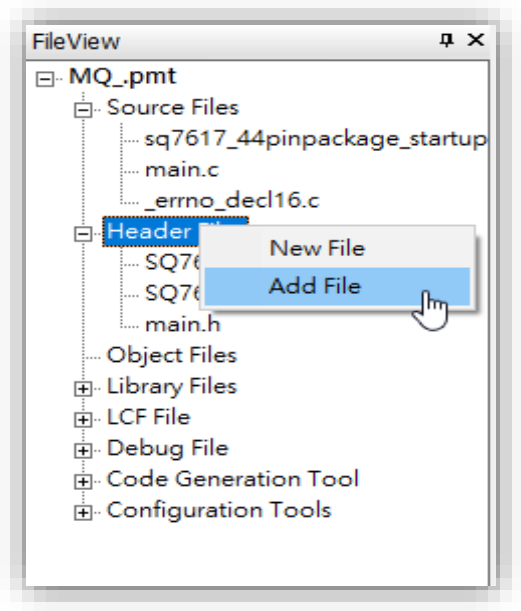
```
#pragma section const INT_VECTOR1 0xff40
void * const IntTbl1[] = {
    OnlyReti,               /* 0xff40 : Reserved        */
    OnlyReti,               /* 0xff42 : Reserved        */
    OnlyReti,               /* 0xff44 : Reserved        */
    OnlyReti,               /* 0xff46 : Reserved        */
    OnlyReti,               /* 0xff48 : Reserved        */
    OnlyReti,               /* 0xff4a : Reserved        */
    OnlyReti,               /* 0xff4c : Reserved        */
    OnlyReti,               /* 0xff4e : Reserved        */
    OnlyReti,               /* 0xff50 : Reserved        */
    OnlyReti,               /* 0xff52 : Reserved        */
    OnlyReti,               /* 0xff54 : Reserved        */
    OnlyReti,               /* 0xff56 : Reserved        */
    OnlyReti,               /* 0xff58 : IntTCA7         */  //中断源:TCA7 16位定时器
    OnlyReti,               /* 0xff5a : IntTCA6         */  //中断源:TCA6 16位定时器
    OnlyReti,               /* 0xff5c : Reserved        */
    OnlyReti,               /* 0xff5e : Reserved        */
    OnlyReti,               /* 0xff60 : Reserved        */
    OnlyReti,               /* 0xff62 : Reserved        */
    OnlyReti,               /* 0xff64 : Reserved        */
    OnlyReti,               /* 0xff66 : Reserved        */
    OnlyReti,               /* 0xff68 : Reserved        */
    OnlyReti,               /* 0xff6a : Reserved        */
    OnlyReti,               /* 0xff6c : IntTX2          */  //中断源:UART2 TX2
    OnlyReti,               /* 0xff6e : IntRX2          */  //中断源:UART2 RX2
    OnlyReti,               /* 0xff70 : Reserved        */
    OnlyReti,               /* 0xff72 : Reserved        */
    OnlyReti,               /* 0xff74 : Reserved        */
    OnlyReti,               /* 0xff76 : IntTCA5         */  //中断源:TCA5 16位定时器
    OnlyReti,               /* 0xff78 : IntTCA4         */  //中断源:TCA4 16位定时器
    OnlyReti,               /* 0xff7a : Reserved        */
    OnlyReti,               /* 0xff7c : Reserved        */
    OnlyReti,               /* 0xff7e : Reserved        */
```

```
    OnlyReti,               /* 0xffe0 : Reserved        */
    OnlyReti,               /* 0xffe2 : Reserved        */
    OnlyReti,               /* 0xffe4 : Reserved        */
    OnlyReti,               /* 0xffe6 : IntTCA1         */  //中断源:TCA1 16位定时器
    OnlyReti,               /* 0xffe8 : IntTCA0         */  //中断源:TCA0 16位定时器
    OnlyReti,               /* 0xffea : Reserved        */
    OnlyReti,               /* 0xffec : Reserved        */
    OnlyReti,               /* 0xffee : IntRTC          */  //中断源:实时时钟RTC定时器
    OnlyReti,               /* 0xfff0 : IntTBT          */  //中断源:时基TBT定时器
    OnlyReti,               /* 0xfff2 : Reserved        */
    OnlyReti,               /* 0xfff4 : IntCFD          */  //中断源:时钟失效侦测
    OnlyReti,               /* 0xfff6 : IntLVD          */  //中断源:低电压侦测LVD
    IntWDT,                 /* 0xfff8 : IntWDT          */  //non-maskable 中断源:看门狗定时器
    (void *)0xffff,         /* 0xfffa : Reserved        */
    IntSWI,                 /* 0xfffc : IntSWI/INTUNDEF */  //non-maskable

    // Boot mark start
    iap_boot,               /* 0xfffe : RESET           */  //non-maskable
    // Boot mark end
#pragma section const
/* ------------------- 结束 SQ7615中断向量表定义 ---------------------- */
```

2.4 專案加入組態標頭檔

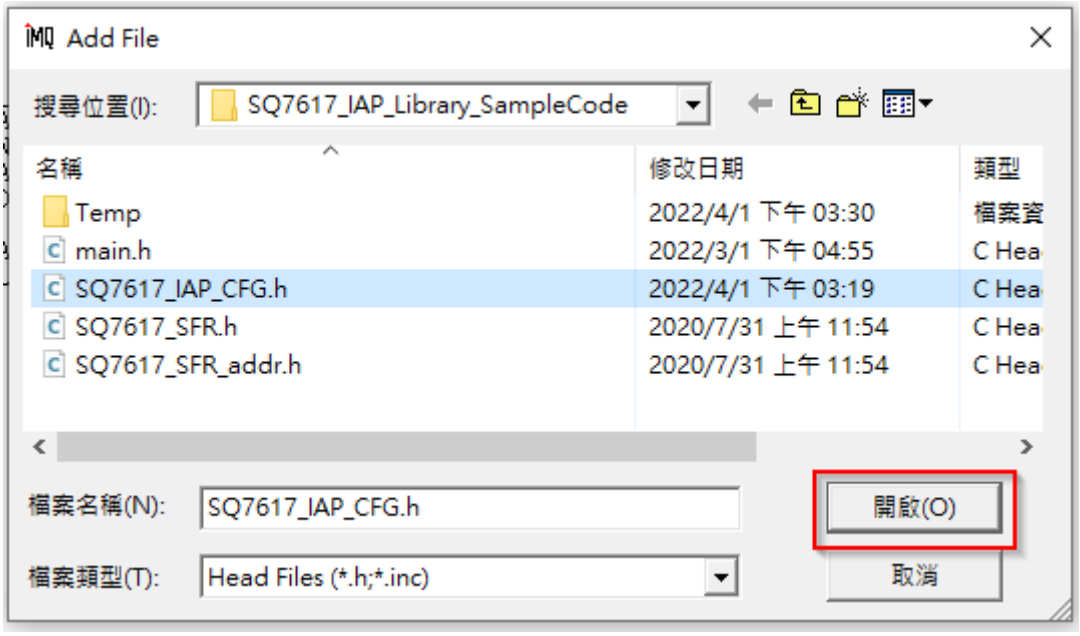
A. 在檔案視圖中**(File view)**對 **Header File** 點擊滑鼠右鍵 **->** 增加檔案**(Add File)**，如下圖：

圖表 2-13 加入標頭檔



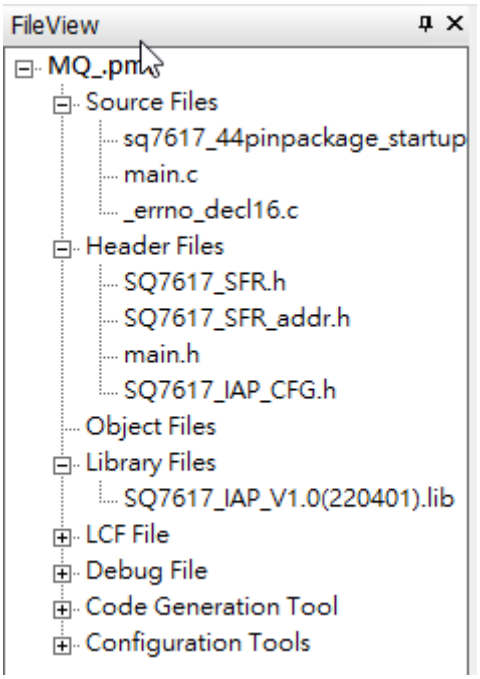
B. 選擇與 **SQ7617** 相對應之 **IAP** 組態標頭檔**(.h)**並開啟，如下圖：

圖表 2-14 選擇 IAP CFG 標頭檔



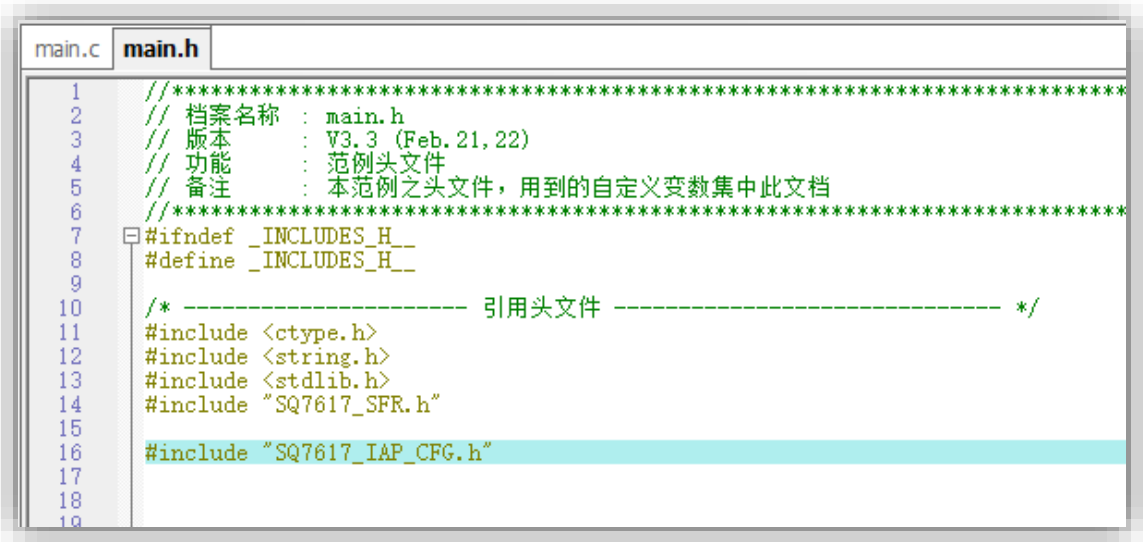
C. 確認 **IAP** 組態標頭檔**(MQ7617_IAP_CFG.h)**已被加入至專案：

圖表 2-15 確認標頭檔



D. 增加下列代碼至主程式標頭檔**(main.h)**才可引用**(#include)IAP** 組態標頭檔

圖表 2-16 引用 IAP CFG 組態標頭檔

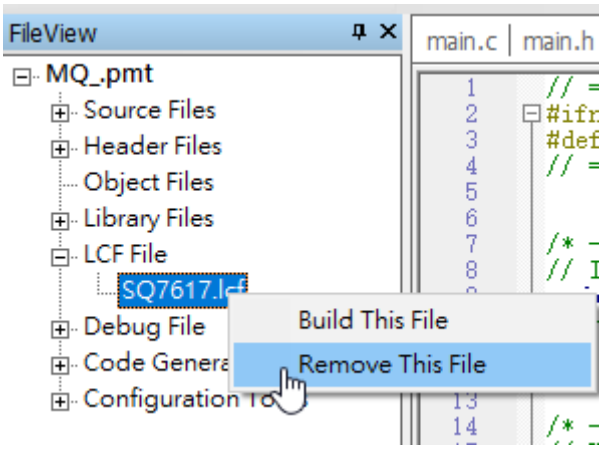


2.5 關聯指令檔修改(LCF file)

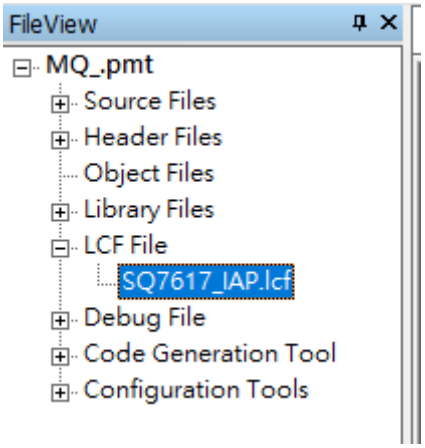
編譯關聯指令檔(.lcf file)目的為使 **i87** 編譯器正確編譯 **IAP** 函式庫代碼至指定位址。下列兩種方式擇一即可

方式一：移除初始專案內”SQ7617.lcf”檔案，新增”SQ7617_IAP.lcf”至專案內。

圖表 2-17 移除 SQ7617.lcf 檔案

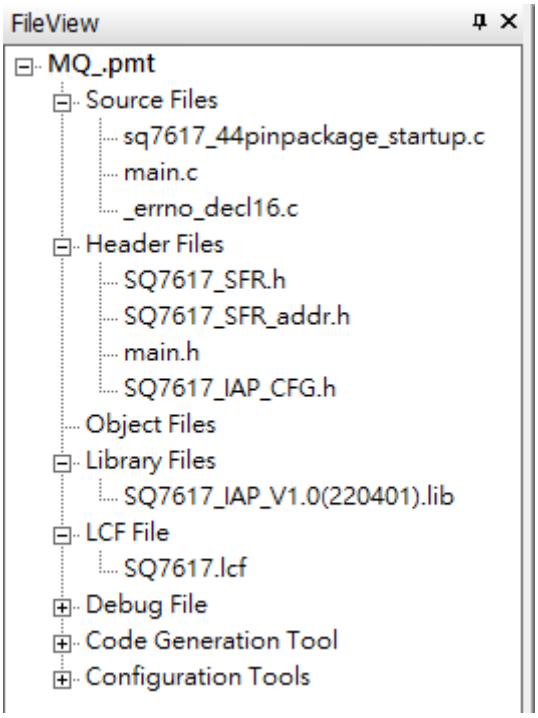


圖表 2-18 新增 SQ7617_IAP.lcf 檔案



方式二：修改專案內”SQ7617.lcf”檔案 Sections 內的代碼

圖表 2-19 關聯指令檔



1. 增加 **boot_code org = 0x0000 (OVERLAY) : {*(c_boot)}**
   說明：設置 **IAP** 編碼起始位址

2. 增加 **boot_value org = 0x0780 (OVERLAY) : {*(c_value)}**
   說明：設置 **IAP** 變數儲存區域位址

3. 增加 **boot_table org = 0x0800 (OVERLAY) : {*(t_boot)}**
   說明：設置當 **IAP** 返回正常模式中斷向量**(STARTUP)**之位址

4. 修改 **near_code org = 0x0802**
   addr = org(tiny_area) + sizeof(tiny_area) : {*(t_data)}

圖表 2-20 初始關聯檔 Sections 代碼

```
sections
{
    tiny_area     org=0x1000 : {*(t_area)}
    near_code     org=0x0000 (OVERLAY): {*(StartUp)*(n_code)*(Main)*(Interrupt_Handler)}
                          x_cal = org(near_code)+sizeof(near_code);
    tiny_data     org=x_cal*((x_cal & 0x8000)>>15) + 0x8000 *(((x_cal + 0x8000)&0x8000)>>15)
                  addr=org(tiny_area)+sizeof(tiny_area)(OVERLAY) : {*(t_data)}
    near_area     org=addr(tiny_data)+sizeof(tiny_data) : {*(n_area)}
    near_data     org=org(tiny_data)+sizeof(tiny_data)
                  addr=org(near_area)+sizeof(near_area)(OVERLAY): {*(n_data)}
    near_const    org=org(near_data)+sizeof(near_data): {*(n_const)}
}
```

圖表 2-21 修改後關聯檔 Sections 代碼

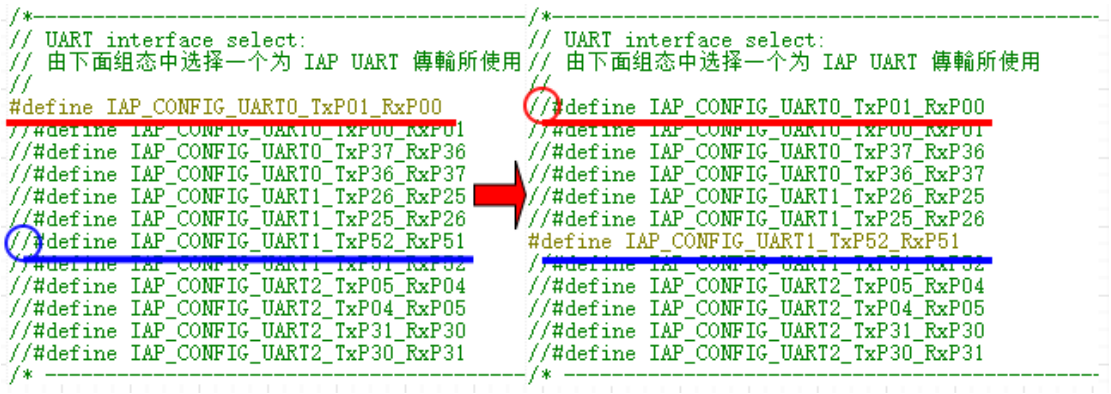
```
sections
{
    tiny_area     org=0x1000 : {*(t_area)}

                  boot_code       org = 0x0000 (OVERLAY) : {*(c_boot)}
                  boot_value      org = 0x0780 (OVERLAY) : {*(c_value)}
                  boot_table      org = 0x0800 (OVERLAY) : {*(t_boot)}

    near_code     org=0x0802 (OVERLAY): {*(StartUp)*(n_code)*(Main)*(Interrupt_Handler)}
                  x_cal = org(near_code)+sizeof(near_code);

    tiny_data     org=x_cal*((x_cal & 0x8000)>>15) + 0x8000 *(((x_cal + 0x8000)&0x8000)>>15)
                  addr=org(tiny_area)+sizeof(tiny_area) (OVERLAY): {*(t_data)}
    near_area     org=addr(tiny_data)+sizeof(tiny_data) : {*(n_area)}
    near_data     org=org(tiny_data)+sizeof(tiny_data)
                  addr=org(near_area)+sizeof(near_area)(OVERLAY): {*(n_data)}
    near_const    org=org(near_data)+sizeof(near_data): {*(n_const)}
}
```

2.6 IAP 組態檔設定

A. 設定 “SQ7617_IAP_CFG.h” 標頭檔內之定義，設置 IAP 相關設定。

B. 組態檔已經協助使用者設定 IAP 使用之 UART、GPIO 等設定細項，使用者僅需選擇對應的設定即可設定完成。

C. 如何變更組態：
若要變更組態需要在原來使用的組態最前頭編寫雙斜線(//)使原來使用的組態失效，然後刪除需要使用的組態最前頭的雙斜線(//)。
只能選擇一種組態，否則編譯將會出現錯誤，舉例說明如下圖：



D. 以下組態必須完整設定，否則編譯將出現錯誤或無法編譯完整功能。

1. Normal boot start address:

圖表 2-22 正常模式起始位置

```
// Normal boot start address:
// 设定更新起始位置
//
#define NORMAL_BOOT_ADDRESS      (0x0800)
// 定义说明:
//       因IAP Library资料大小约使用2048Bytes(0x0800(hex))的大小,
//       使用者编码起始位置设定从位址开始更新,不可小於此位址
//       LCF(Sample Link Command File)档案已经设定相同位址
```

2. UART interface select:

由以下 **12** 種組態中選擇一組 **UART** 組態當作更新傳輸媒介。

圖表 2-23 UART 端口組態設定

```
/*---------------------------------------------------------------*/
// UART interface select:
// 由下面组态中选择一个为 IAP UART 傳輸所使用
//
//#define IAP_CONFIG_UART0_TxP01_RxP00
//#define IAP_CONFIG_UART0_TxP00_RxP01
//#define IAP_CONFIG_UART0_TxP37_RxP36
//#define IAP_CONFIG_UART0_TxP36_RxP37
//#define IAP_CONFIG_UART1_TxP26_RxP25
//#define IAP_CONFIG_UART1_TxP25_RxP26
#define IAP_CONFIG_UART1_TxP52_RxP51
//#define IAP_CONFIG_UART1_TxP51_RxP52
//#define IAP_CONFIG_UART2_TxP05_RxP04
//#define IAP_CONFIG_UART2_TxP04_RxP05
//#define IAP_CONFIG_UART2_TxP31_RxP30
//#define IAP_CONFIG_UART2_TxP30_RxP31
/* ---------------------------------------------------------------*/
// 定义说明:
//       IAP_CONFIG_UART0_TxP01_RxP00:  使用UART0传送, P0.1作发送端口,P0.0作接收端口
//       IAP_CONFIG_UART0_TxP00_RxP01:  使用UART0传送, P0.0作发送端口,P0.1作接收端口
//       IAP_CONFIG_UART0_TxP37_RxP36:  使用UART0传送, P3.7作发送端口,P3.6作接收端口
//       IAP_CONFIG_UART0_TxP36_RxP37:  使用UART0传送, P3.6作发送端口,P3.7作接收端口
//       IAP_CONFIG_UART1_TxP26_RxP25:  使用UART1传送, P2.6作发送端口,P2.5作接收端口
//       IAP_CONFIG_UART1_TxP25_RxP26:  使用UART1传送, P2.5作发送端口,P2.6作接收端口
//       IAP_CONFIG_UART1_TxP52_RxP51:  使用UART1传送, P5.2作发送端口,P5.1作接收端口
//       IAP_CONFIG_UART1_TxP51_RxP52:  使用UART1传送, P5.1作发送端口,P5.2作接收端口
//       IAP_CONFIG_UART2_TxP05_RxP04:  使用UART2传送, P0.5作发送端口,P0.4作接收端口
//       IAP_CONFIG_UART2_TxP04_RxP05:  使用UART2传送, P0.4作发送端口,P0.5作接收端口
//       IAP_CONFIG_UART2_TxP31_RxP30:  使用UART2传送, P3.1作发送端口,P3.0作接收端口
//       IAP_CONFIG_UART2_TxP30_RxP31:  使用UART2传送, P3.0作发送端口,P3.1作接收端口
/* ---------------------------------------------------------------*/
```

3. Baud rate select:

由以下 **4** 種組態中選擇一組波特率使用。

圖表 2-24 波特率組態設定

```
/* ---------------------------------------------------------------*/
// Baud rate select:
// 由下面组态中选择一个为 IAP UART Baudrate Speed所使用
// -- Baud Rate Setting------
//
#define Operat24MBaud_115200
//#define Operat24MBaud_57600
//#define Operat24MBaud_19200
//#define Operat24MBaud_9600
/* ---------------------------------------------------------------*/
```

4. UART control setting (parity / stop bit / even):

設定 **1.** (有/無)奇偶校驗。
2. (奇數/偶數)奇偶校驗。
3. 發送端結束位長度。
4. 接收端結束位長度。

建議使用初始值，即範例 1。

圖表 2-25 UART 收發相關設定

```
#define UART_CR1_VALUE        (0x00)
#define UART_CR2_VALUE        (0x00)
/*-------------------------------------------------------------*/
// 範例1:無奇偶校验,发送端结束位长度1位,接收端结束位长度1位
//       UART_CR1_VALUE = 0x00
//       UART_CR2_VALUE = 0x00
//
// 範例2:無奇偶校验,发送端结束位长度2位,接收端结束位长度2位
//       UART_CR1_VALUE = 0x20
//       UART_CR2_VALUE = 0x01
//
// 範例3:有奇偶校验,奇数奇偶校验,发送端结束位长度1位,接收端结束位长度1位
//       UART_CR1_VALUE = 0x08
//       UART_CR2_VALUE = 0x00
//
// 範例4:有奇偶校验,奇数奇偶校验,发送端结束位长度2位,接收端结束位长度2位
//       UART_CR1_VALUE = 0x28
//       UART_CR2_VALUE = 0x01
//
// 範例5:有奇偶校验,偶数奇偶校验,发送端结束位长度1位,接收端结束位长度1位
//       UART_CR1_VALUE = 0x18
//       UART_CR2_VALUE = 0x00
//
// 範例6:有奇偶校验,偶数奇偶校验,发送端结束位长度2位,接收端结束位长度2位
//       UART_CR1_VALUE = 0x38
//       UART_CR2_VALUE = 0x01
/* -------------------------------------------------------------*/
```

5. Enable boot PIN select:

由 **44** 種組態中選一個 **GPIO** 端口當作 **IAP** 啟動判斷端口。

圖表 2-26 IAP 啟動端口設定內容

```
/* ---------------------------------------------------------------*/
// Enable Boot PIN Setting:
// 由下面组态中选择一个为 IAP Boot 启动所使用

// #define IAP_BOOT_PIN_P00
// #define IAP_BOOT_PIN_P01
// #define IAP_BOOT_PIN_P02
// #define IAP_BOOT_PIN_P04
// #define IAP_BOOT_PIN_P05
// #define IAP_BOOT_PIN_P06
// #define IAP_BOOT_PIN_P10
// #define IAP_BOOT_PIN_P11
// #define IAP_BOOT_PIN_P12
// #define IAP_BOOT_PIN_P13
// #define IAP_BOOT_PIN_P14
// #define IAP_BOOT_PIN_P15
// #define IAP_BOOT_PIN_P16
// #define IAP_BOOT_PIN_P17
// #define IAP_BOOT_PIN_P20
// #define IAP_BOOT_PIN_P21
// #define IAP_BOOT_PIN_P22
// #define IAP_BOOT_PIN_P24
// #define IAP_BOOT_PIN_P25
// #define IAP_BOOT_PIN_P26
// #define IAP_BOOT_PIN_P30
// #define IAP_BOOT_PIN_P31
// #define IAP_BOOT_PIN_P32
// #define IAP_BOOT_PIN_P33
// #define IAP_BOOT_PIN_P35
// #define IAP_BOOT_PIN_P43
#define IAP_BOOT_PIN_P50
// #define IAP_BOOT_PIN_P51
// #define IAP_BOOT_PIN_P52
/* ---------------------------------------------------------------*/
// 定义说明:
//     请參考datasheet选择适合可用的Boot PIN
//     但建议避开 P3.4、P4.2(OCD仿真佔用)、P4.0、P4.1(LX佔用)、P4.4、P4.5(HX佔用)
//     建议可从 PORT_0/PORT_1/PORT_2/PORT_3/PORT_4/PORT_5 的其他未被使用脚位挑选
/* ---------------------------------------------------------------*/
```

```
/* ---------------------------------------------------------------*/
// 定義說明:
//       各啟動端口組態包含下列詳細設定:
//       1.IAP_BOOT_PIN_FC_REG:端口功能控制寄存器位址(PxFC)
//       2.IAP_BOOT_PIN_PU_REG:端口內置上拉电阻控制寄存器位址(PxPU)
//       3.IAP_BOOT_PIN_PD_REG:端口內置下拉电阻控制寄存器位址(PxPD)
//       4.IAP_BOOT_PIN_CR_REG:端口輸入輸出控制寄存器位址(PxCR)
//       5.IAP_BOOT_PIN_PRD_REG:端口輸入数据寄存器位址(PxPRD)
//       6.IAP_BOOT_PIN_BIT:端口Bit位數值
/* ---------------------------------------------------------------*/
```

6. IAP boot pin default level:

設定啟動 **GPIO** 端口在上電時初始的電平高低。

圖表 2-27 IAP 啟動端口初始電平設定

設定 **IAP** 啟動端口"初始"電平。

```
/* ------------------------------------------------------------*/
// IAP boot pin default level:
// 设定 IAP Boot 启动端口(初始)电位
// 初始(Default) 設為高電位(High) = 0x01;
//               設為低電位(Low)  = 0x00;
#define IAP_BOOT_GPIO_DEFAULT_HIGH_LOW (0x00)
/* ------------------------------------------------------------*/
```

7. IAP boot pin enable level

設定啟動 **GPIO** 端口判斷是否進入 **IAP** 更新模式時的電平高低。

圖表 2-28 IAP 啟動端口啟動電平設定

設定 **IAP** 啟動端口"啟動"電平。

```
/* ------------------------------------------------------------*/
// IAP boot pin enable level
// 设定 IAP Boot 启动端口(启动)电位
// 啟動(Enable)  設為高電位(High) = 0x01;
//               設為低電位(Low)  = 0x00;
#define IAP_BOOT_GPIO_ENABLE_HIGH_LOW  (0x01)
/* ------------------------------------------------------------*/
```

3 更新實作

3.1 實作需求

A. 硬體：**RS232** 轉 **UART** 通訊模組**(**此實作範例使用 **FT232 USB UART Board)**

B. 軟體：

I. **COM Port to UART Communication Tool.exe**

II. **iMQ i87-IDE.exe**

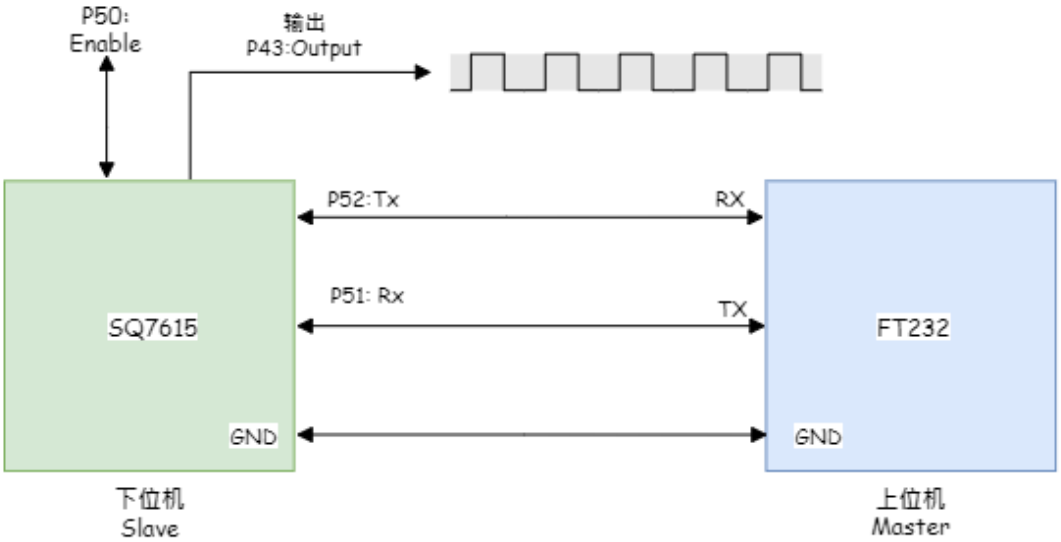
C. 開發板:**MQ7617 EV Board v1.1**

3.2 實作目的

將下位機代碼更新。

3.3 接線設定

圖表 3-1 實作接線設定



3.4 實作專案建立

3.4.1 原 SQ7617 專案建立

A. 依據第 2 章節全部說明建立使用 IAP 函式庫之專案。
B. 編譯 **SQ7617_IAP_CFG.h** 標頭檔案的下列組態設定。

I. **NORMAL_BOOT_ADDRESS**：**(0x0800)**
說明：設定更新起始位址為 **0x0800**。

II. **UART interface select**：
IAP_CONFIG_UART1_TxP52_RxP51
說明：設定 **P52** 為發送端口、**P51** 為接收端口。

III. **Baud rate select**：
BAUDRATE_115200
說明：設定波特率為 **115200**。

IV. **UART control setting (parity / stop bit / even)**：
UART_CR1_VALUE (0x00)
UART_CR2_VALUE (0x00)
說明：設定 **UART_CR1** 特殊功能寄存器值為 **0x00**。
設定 **UART_CR2** 特殊功能寄存器值為 **0x00**。

V. **Enable boot PIN setting**
IAP_BOOT_PIN_P50
說明：設定 **P50** 端口為判斷是否啟動 **IAP** 更新程序之端口。

VI. **IAP boot pin default level**
IAP_BOOT_GPIO_DEFAULT_HIGH_LOW (0x00)
說明：設定 **P50** 端口為低電平。

VII. **IAP boot pin enable level**
IAP_BOOT_GPIO_ENABLE_HIGH_LOW (0x01)
說明：設定當 **P50** 端口為高電平時啟動 **IAP** 更新程序。

C. 主程序**(main.c)**主函式**(main)** 編寫 **P43** 高低電平輸出，高低電平切換頻率為 **100(ms)**，編碼如下圖。

圖表 3-2 原專案主程序主函數編碼

```
/* ---------------------- 主程序 -------------------------- */
void main()
{
    DI;
    {
        P4DO_P3 = 0;
        P4PU_P3 = 1;
        P4PD_P3 = 0;
        P4OE_P3 = 1;     // 1 = set p43 been output mode
        P4FC1_P3 = 0;    // 0,0 = set p43 been GPIO type
        P4FC2_P3 = 0;
    }
    EI;

    while(1)
    {
        P4DO_P3 = !P4DO_P3;
        delay_times(0,100);
        NOP;
    }
}
```

[1]**delay_times** 函數為建議延遲時間 100ms。

D. 將編譯完成的代碼載入至下位機。當 **P50** 端口為低電平時，**P43** 端口會輸出頻率為 **100(ms)**之高低電平切換信號。

3.5 軟體說明及更新操作

A. 開啟 **COM Port to UART Communication Tool.exe**

B. 將下位機斷電後，**P50** 端口接上 **3.3V** 拉高電平後上電。
此時下位機會啟動 **IAP** 更新模式。

C. 序列埠連線設定

I. 序列埠設定**(COM Port)**

II. 波特率設定**(Baud Rate)**

III. 有無奇偶校驗**(Parity) 3** 種選項**:**

None -> 無

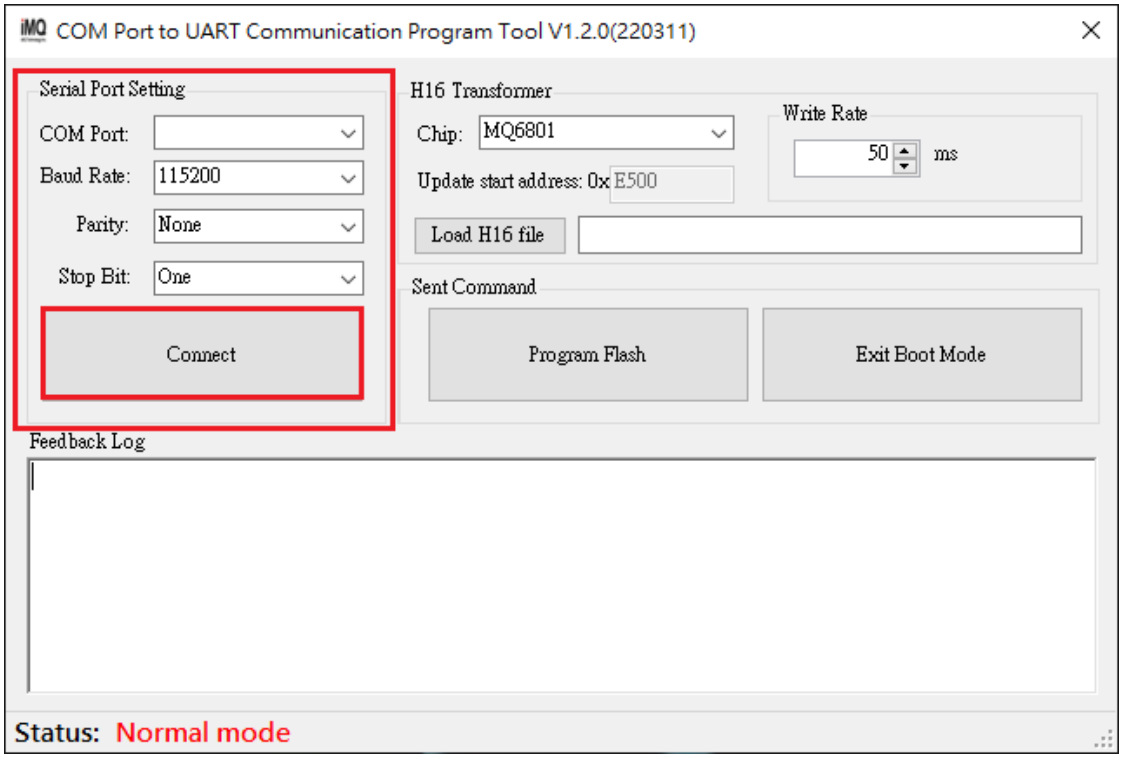
Odd -> 奇校驗

Even -> 偶校驗

IV. 收發結束長度設定

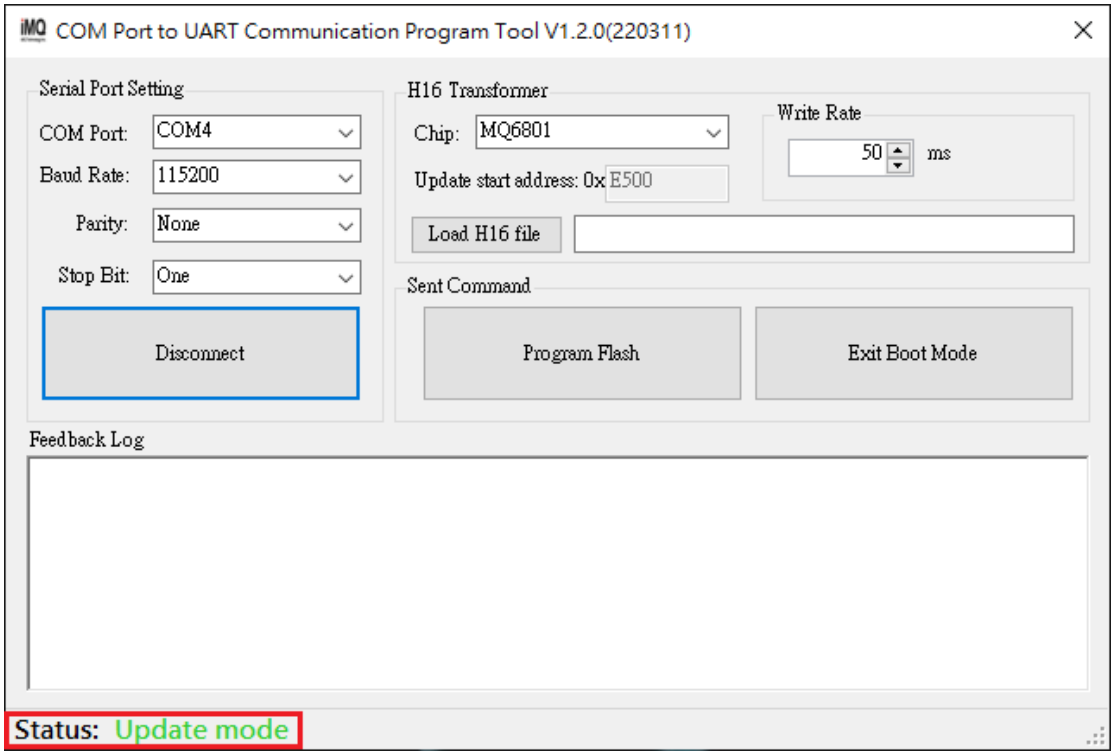
設定完成後點擊連線按鈕**(Connect)**。

圖表 3-3 UART tool 介面-序列埠連線設定說明



序列埠連線設定正確且啟動端口電平為啟動電平時，左下方狀態會顯示更新模式**(Update mode)**，若非啟動電平則為正常模式**(Normal mode)**。

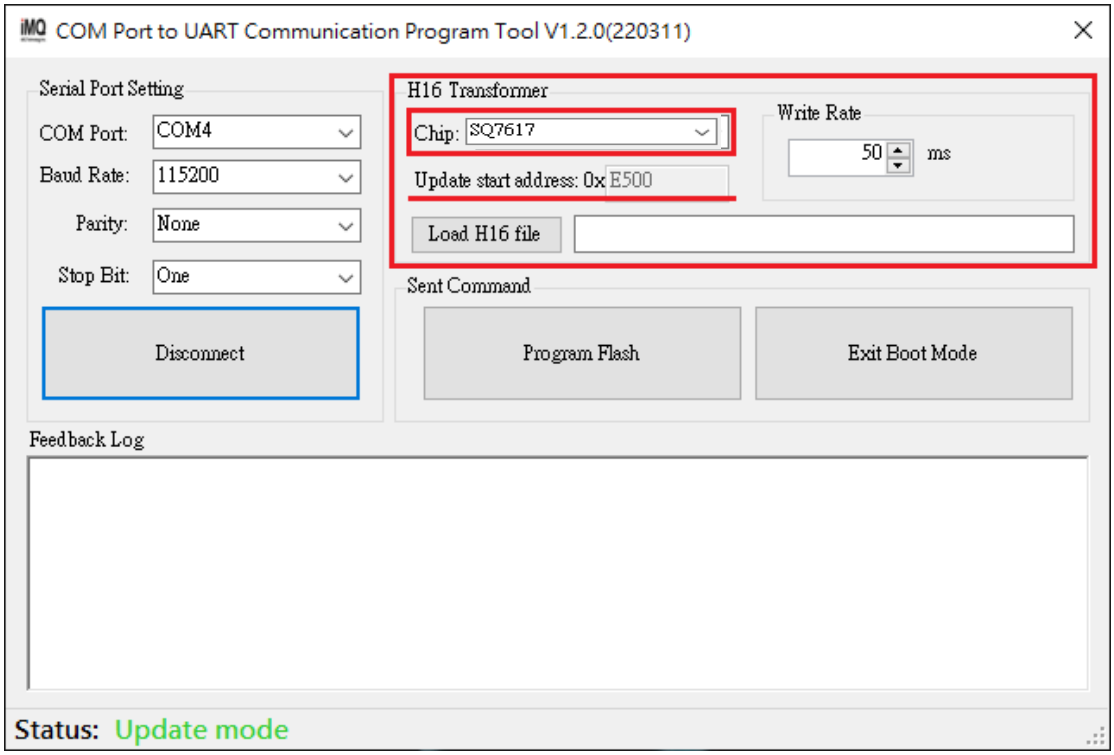
圖表 3-4 UART tool 介面進入更新模式狀態列顯示信息



D. hex 檔案(.h16)轉換器

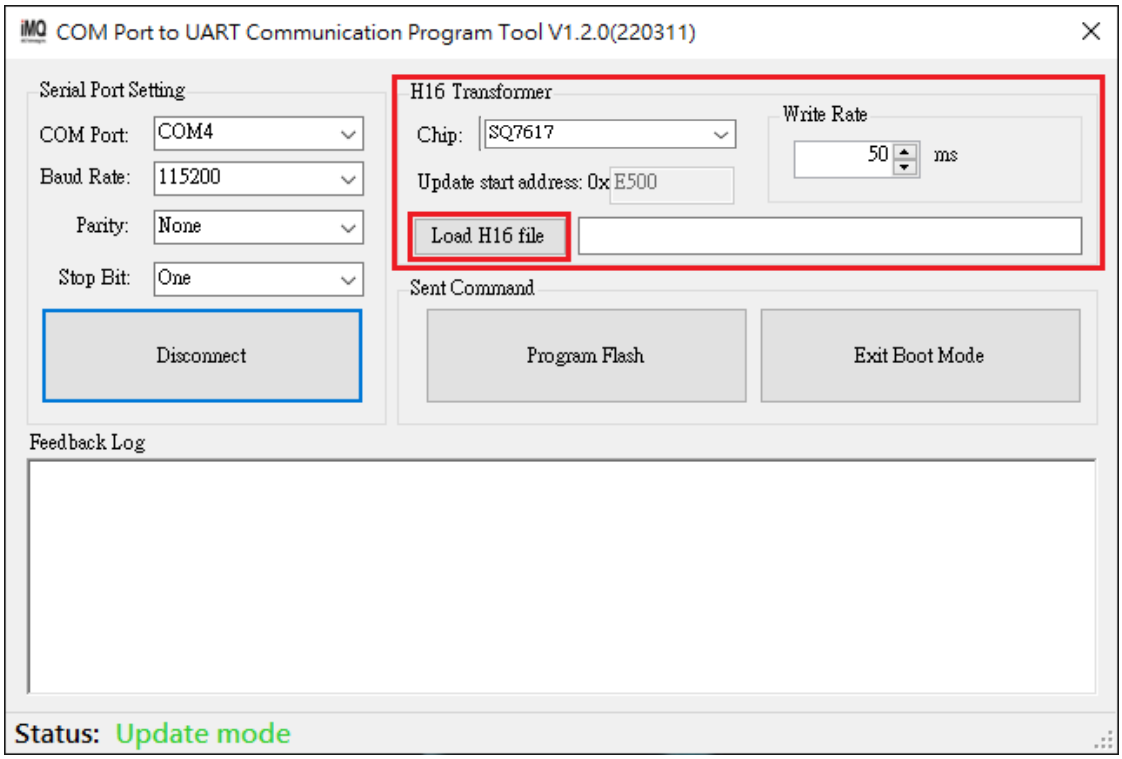
I. 編譯記憶體前需要先設定晶片規格，更新起始位址欄位會自動填入，也可以選擇其他(Other)讓使用者自行填入位址。

圖表 3-5 UART tool 介面-選擇對應晶片



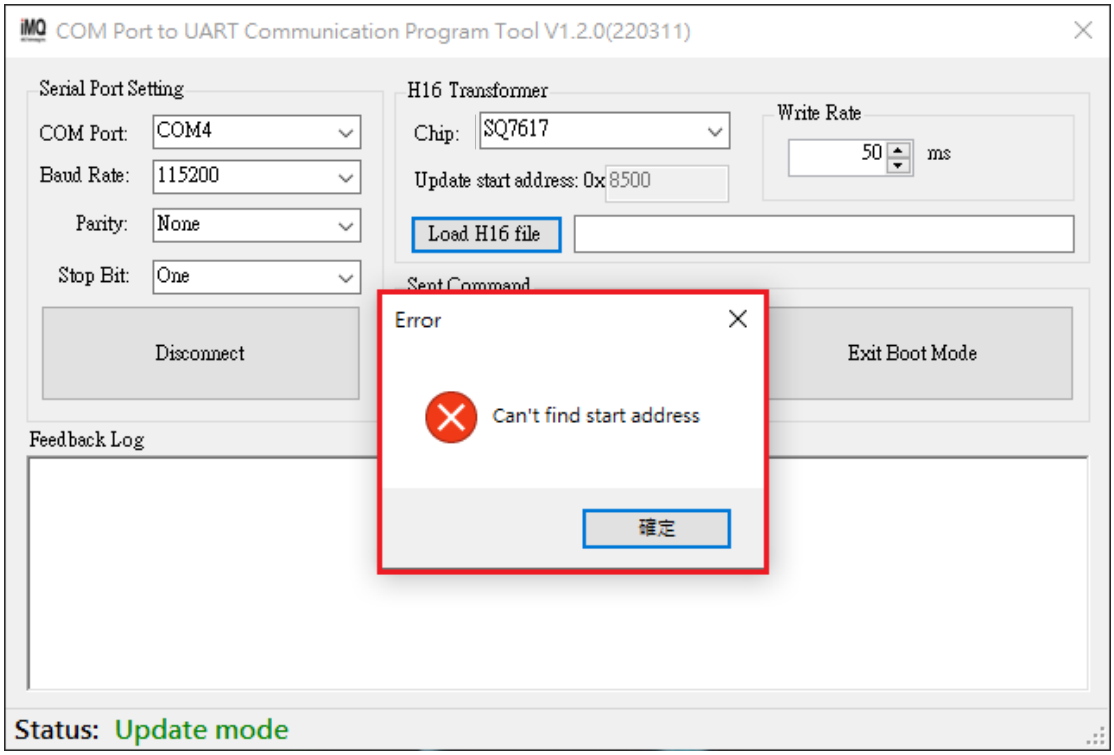
II. 選擇要更新的新 **H16** 檔案。

圖表 3-6 UART tool 介面-載入 H16 檔案



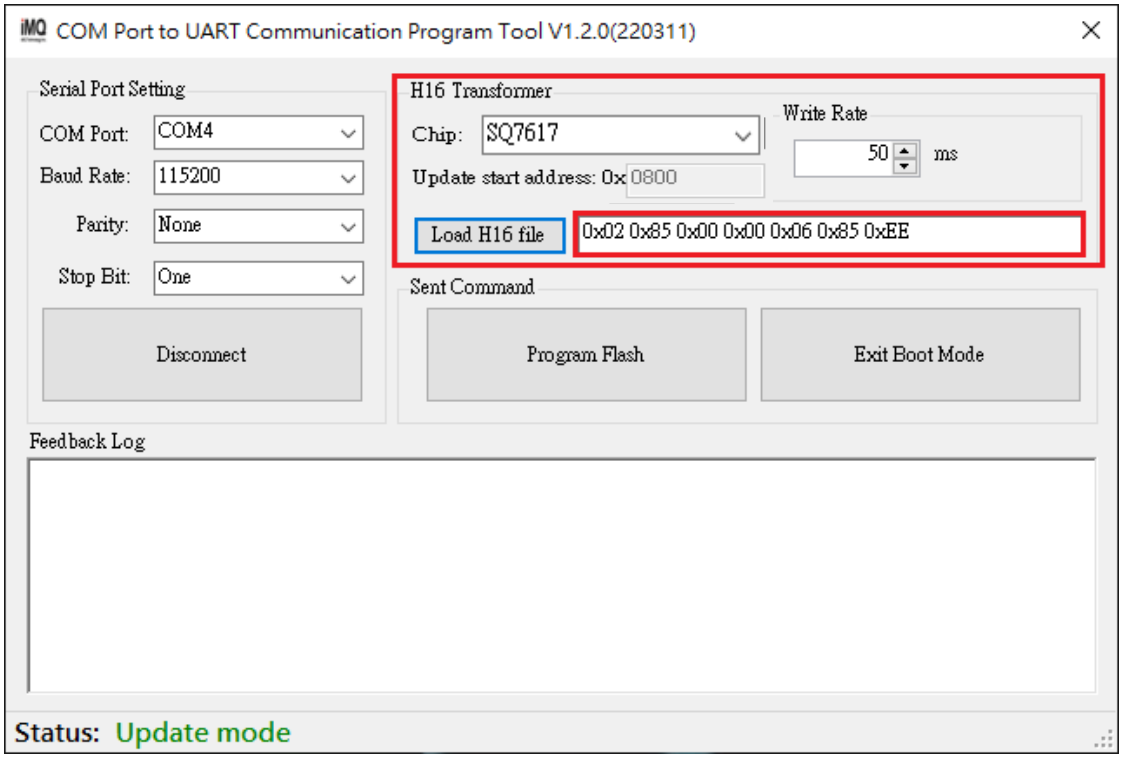
III. 若選擇檔案起始編譯位址不相符，軟體會出現錯誤警示。

圖表 3-7 UART tool 介面-載入 H16 檔案錯誤



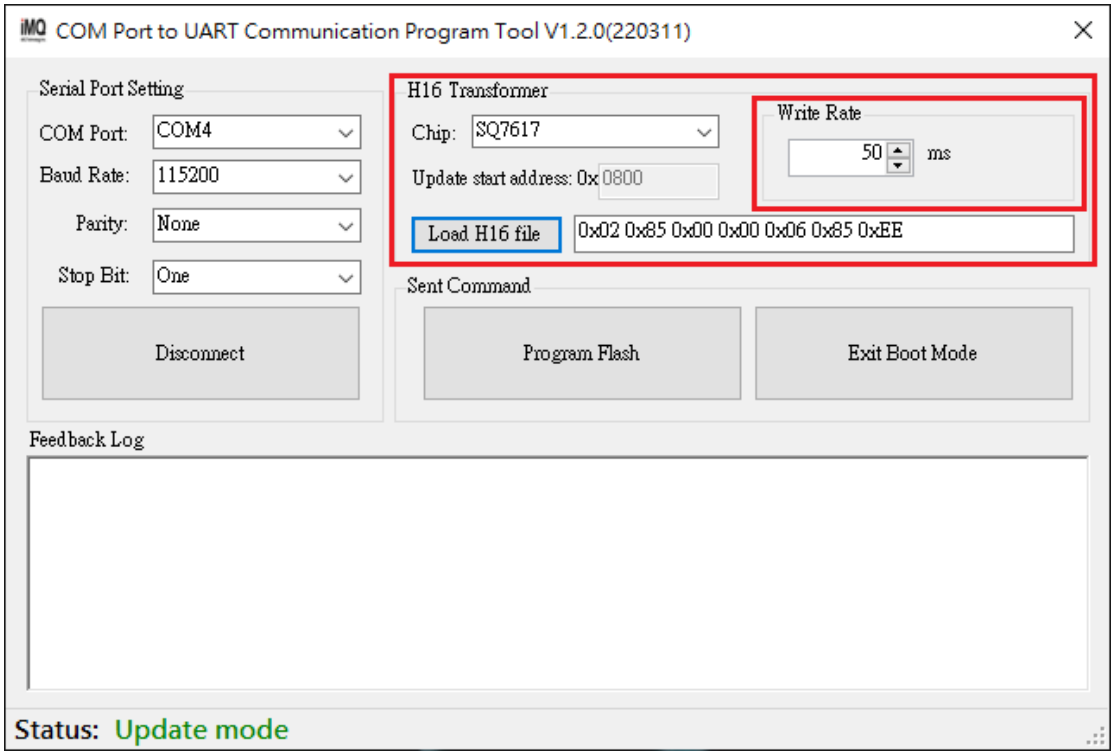
IV. 當檔案選擇正確時，代碼列欄位會出現待發送的代碼第一列。

圖表 3-8 UART tool 介面-H16 檔案代碼顯示



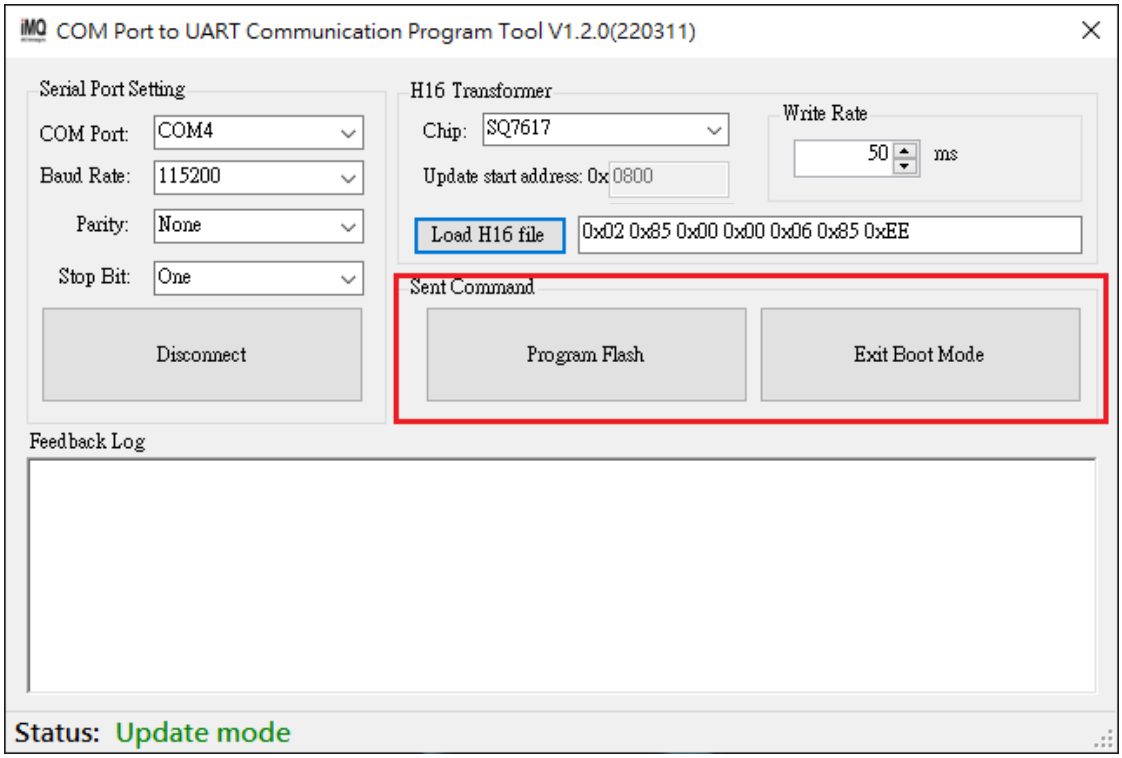
V. 寫入速率設定 50ms~1000ms

圖表 3-9 UART tool 介面-寫入速率設定



E. 指令發送

圖表 3-10 UART tool 介面-指令發送區塊



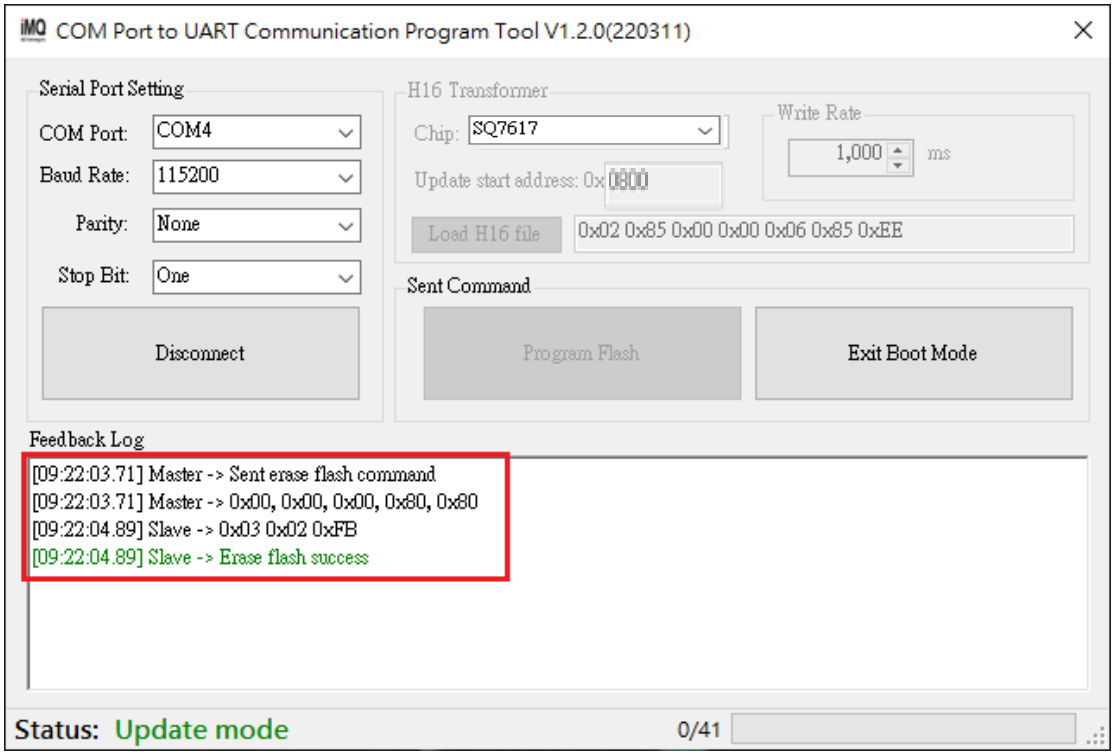
I. 編譯快閃記憶體**(Program Flash)**

i. 按下編譯記憶體按鈕**(Program Flash)**，在執行編譯前會先執行抹除指令。

發送指令**:0x00,0x00,0x00,0x80,0x80**

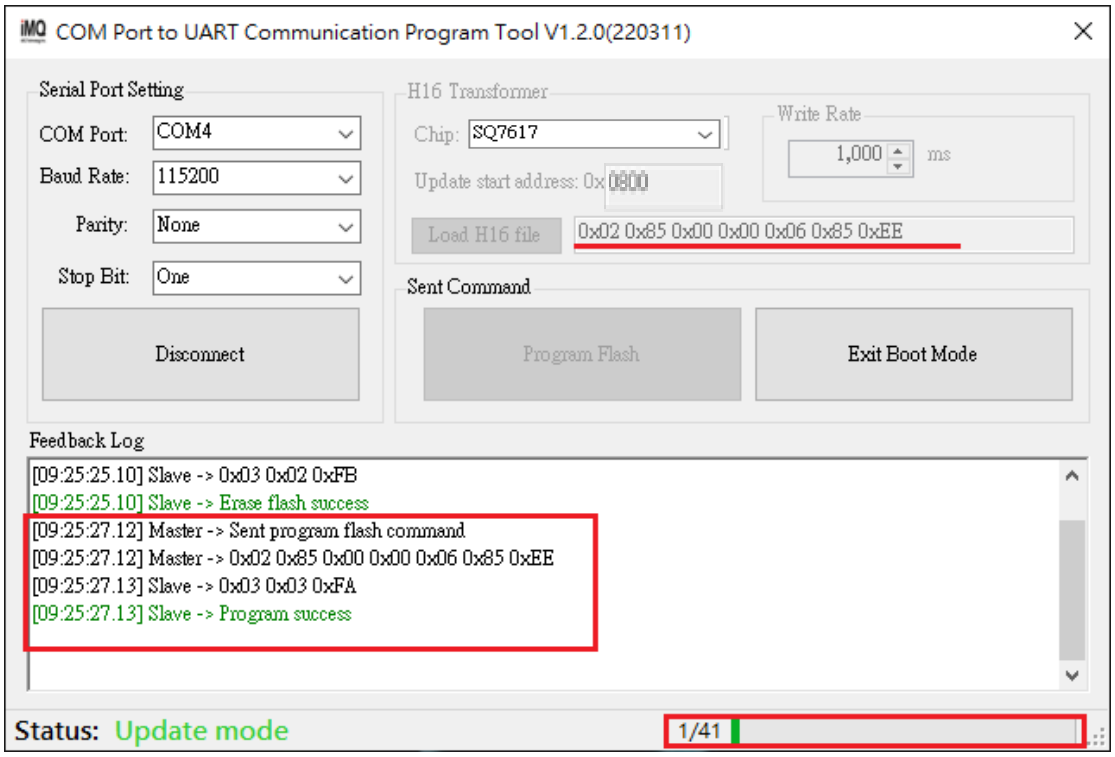
抹除指令執行成功將會回傳信息**:0x03, 0x02, 0xFB** ，如圖圖表 1-6 回傳值內容說明。

圖表 3-11 UART tool 介面-抹除記憶體



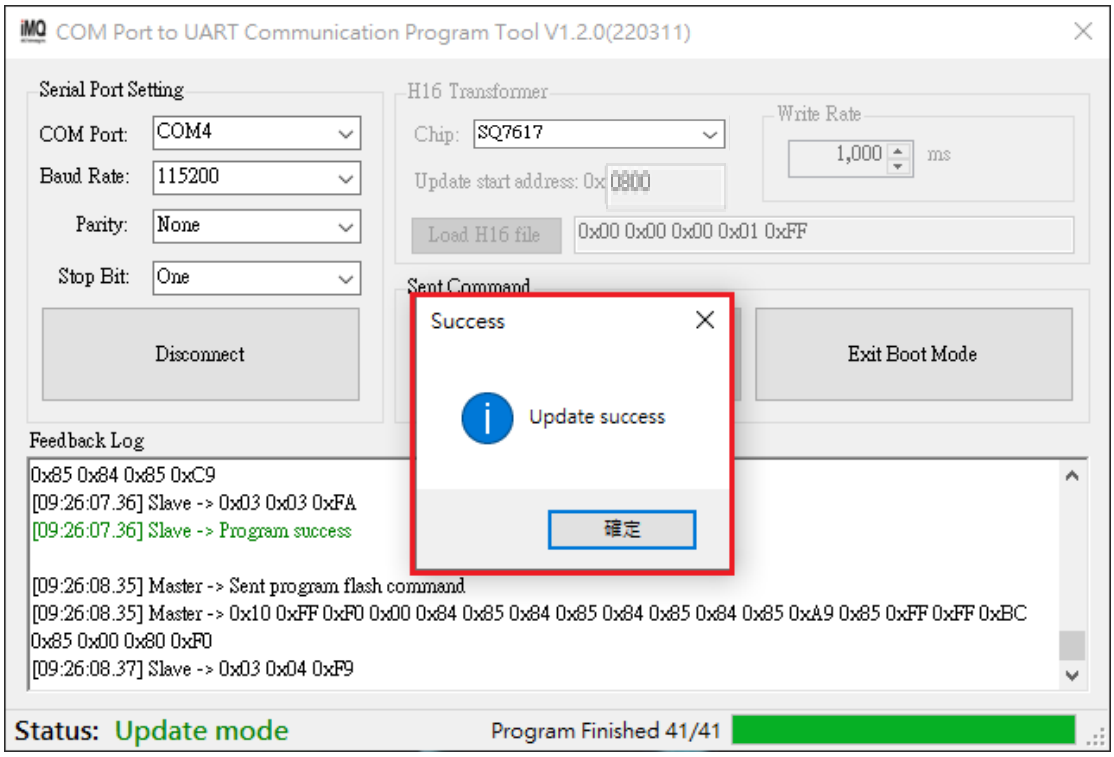
ii. 抹除執行成功後，軟件將會發送代碼欄位內的代碼至下位機，下位機寫入成功會回傳**:0x03, 0x03, 0xFA**，如圖圖表 1-6 回傳值內容說明。
軟件也會計數發送總數及記錄寫入成功總數。

圖表 3-12 UART tool 介面-寫入記憶體



iii. 待更新完成後，軟體會跳出更新成功**(Update success)**。

圖表 3-13 UART tool 介面-更新完成



II. 離開 IAP 更新模式**(Exit Boot Mode)**，下位機寫入成功會回傳**:0x03, 0x04, 0xF9**，如圖圖表 1-6 回傳值內容說明。

圖表 3-14 UART tool 介面-退出更新模式