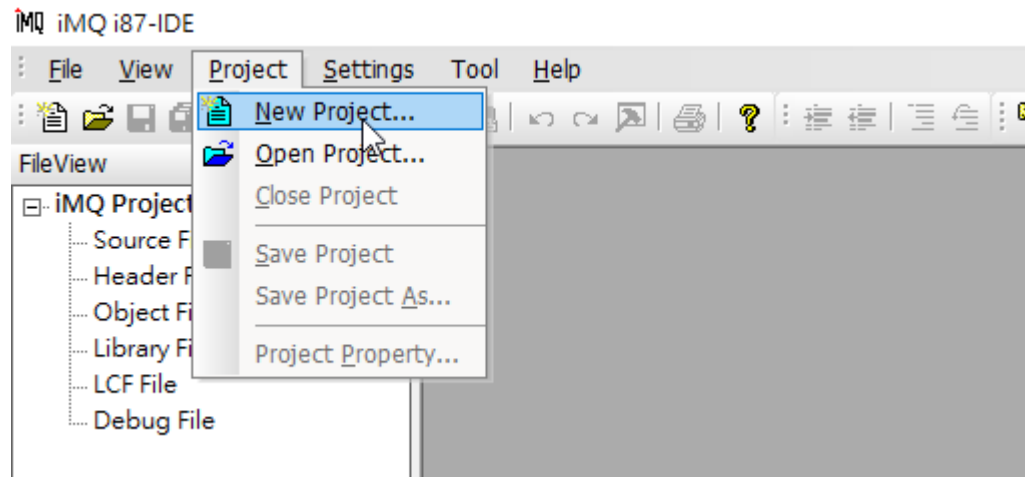


# **SQ7705 U2A sample code**

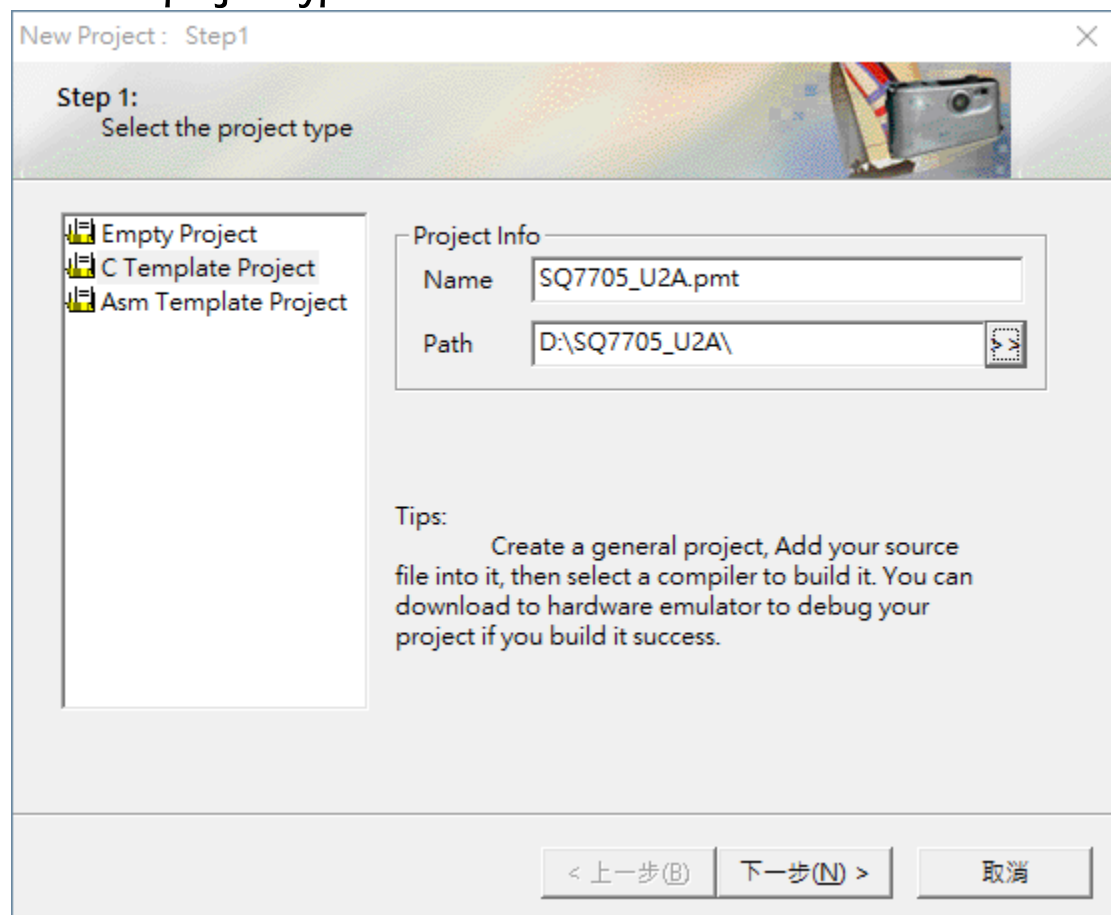
## **introduction**

# 1. Create SQ7705 SIMU Project

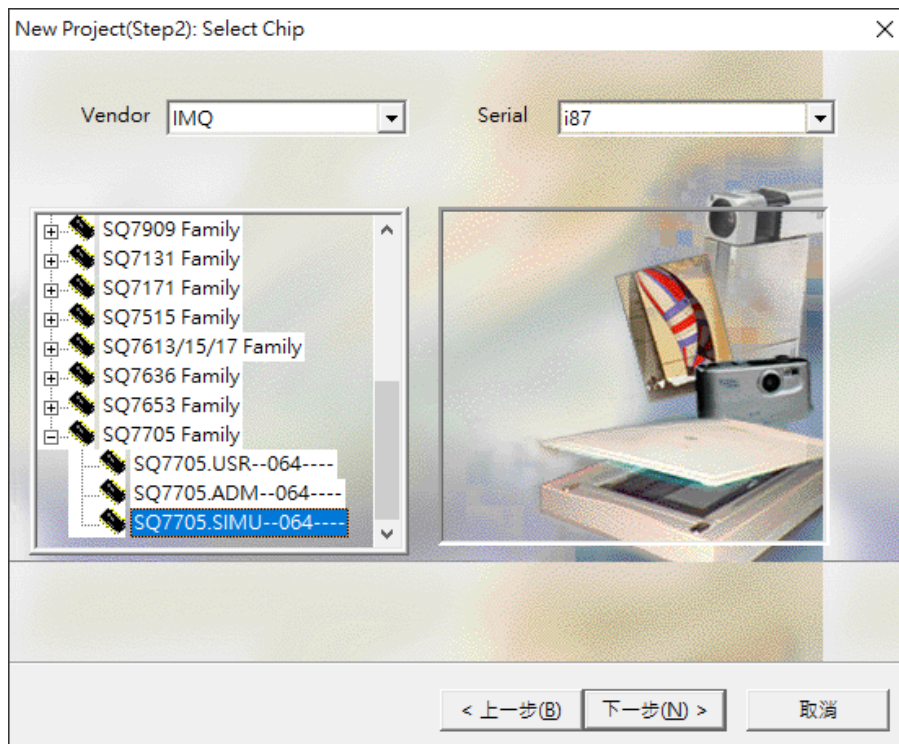
## a. Open IDE and then create New Project



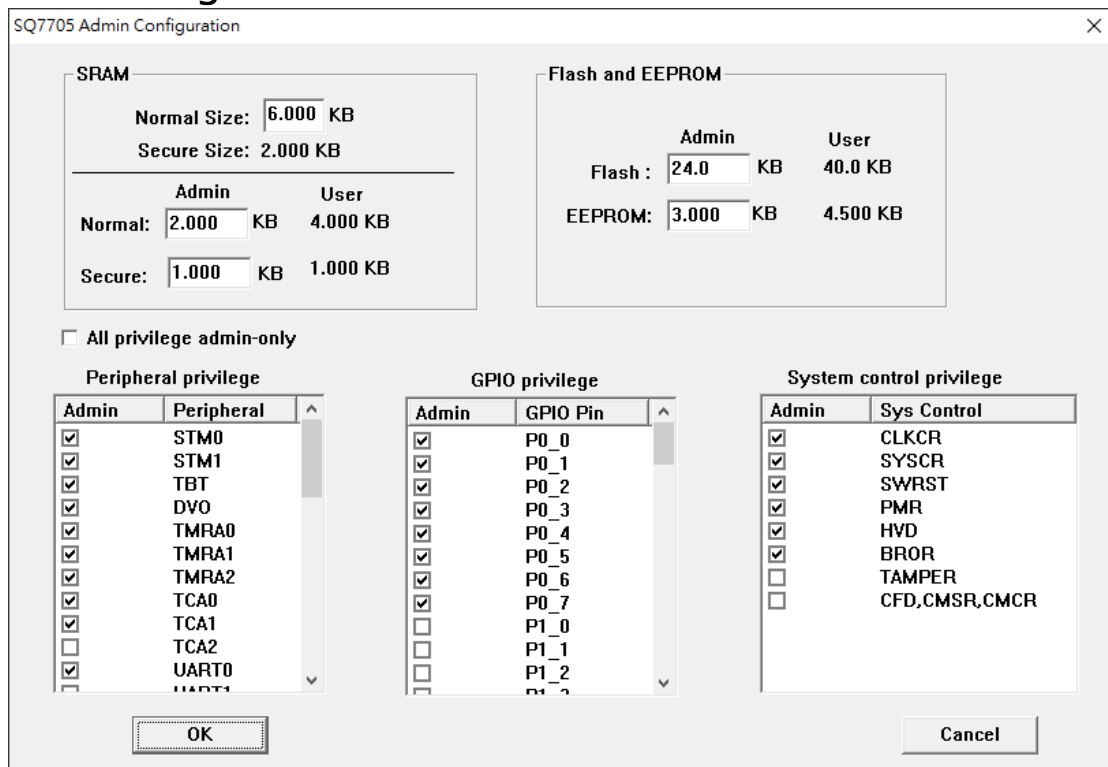
## b. Select project type



### c. Select SQ7705.SIMU project

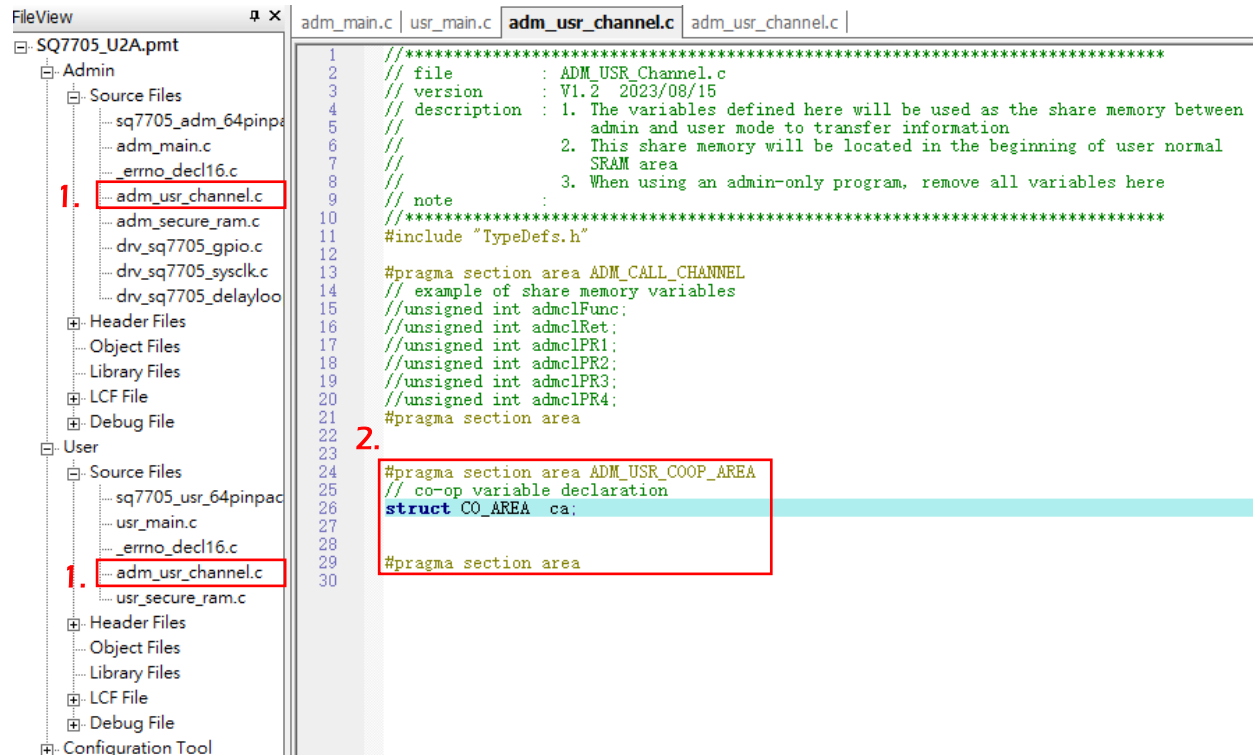


### d. Set configuration



## 2. Share memory

Declare a variable named “ca” that uses the structure named “CO\_AREA” that will be used as the share memory between admin mode and user mode to transfer parameters.



- (1) A variable are declared in the files “adm\_usr\_channel.c” in admin mode and user mode, and must be declared with the same name.
- (2) The variable must be declared within the section of #pragma section area ADM\_USR\_COOP\_AREA.

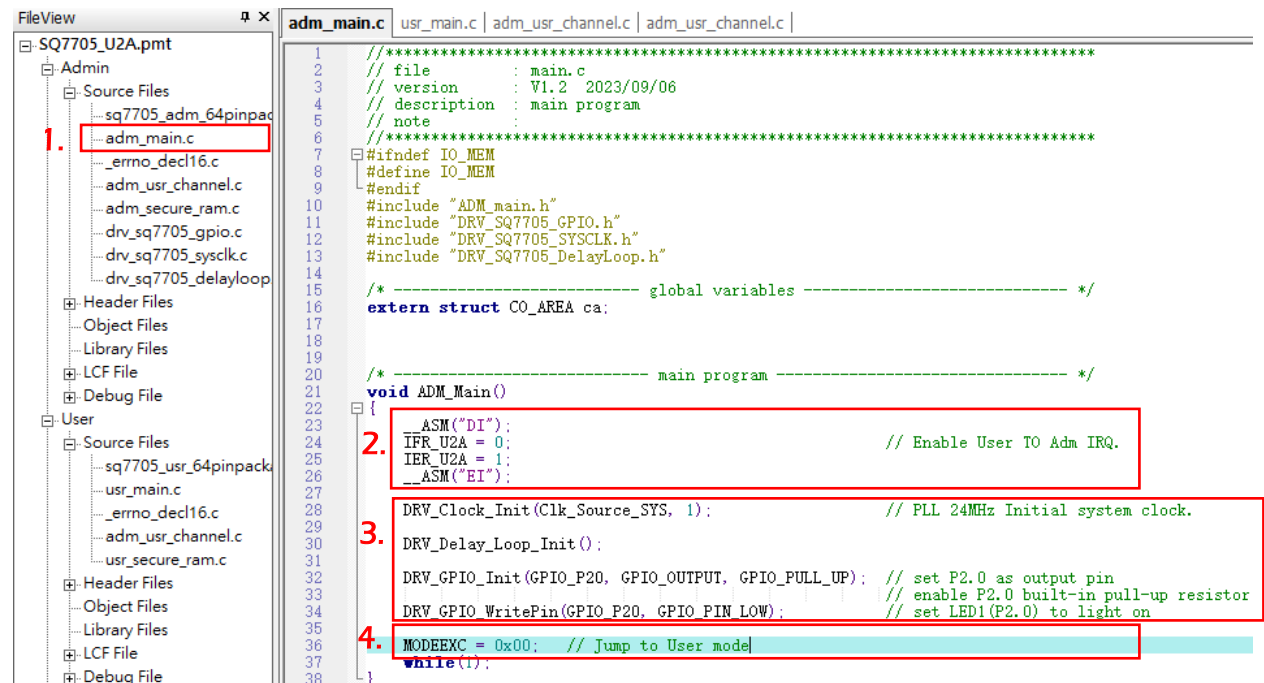
In this example, the declared variable “ca” is a variable of struct “CO\_AREA”. The Structure ‘CO\_AREA’ contains 2 variables “fid” and “para”. “fid” is used to transfer the function ID and “para” is used to transfer parameter of function.

```
struct CO_AREA {
    uint8_t  fid;
    uint16_t para;
};

enum FID {
    FID_LED_ON,
    FID_LED_OFF,
    FID_DELAYLOOP
};
```

### 3. Mode switch between Admin mode and User mode

#### a. Admin mode setting



```
1 //*****
2 // file      : main.c
3 // version   : V1.2 2023/09/06
4 // description : main program
5 // note
6 //*****
7 #ifndef IO_MEM
8 #define IO_MEM
9 #endif
10 #include "ADM_main.h"
11 #include "DRV_SQ7705_GPIO.h"
12 #include "DRV_SQ7705_SYSClk.h"
13 #include "DRV_SQ7705_DelayLoop.h"
14
15 /* ----- global variables ----- */
16 extern struct CO_AREA ca;
17
18 /* ----- main program ----- */
19
20 void ADM_Main()
21 {
22     __ASM("DI");
23     IIR_U2A = 0; // Enable User TO Adm IRQ.
24     IER_U2A = 1;
25     __ASM("EI");
26
27     DRV_Clock_Init(Clk_Source_SYS, 1); // PLL 24MHz Initial system clock.
28
29     DRV_Delay_Loop_Init();
30
31     DRV_GPIO_Init(GPIO_P20, GPIO_OUTPUT, GPIO_PULL_UP); // set P2.0 as output pin
32                                                         // enable P2.0 built-in pull-up resistor
33     DRV_GPIO_WritePin(GPIO_P20, GPIO_PIN_LOW); // set LED1 (P2.0) to light on
34
35     MODEEXC = 0x00; // Jump to User mode
36     while(1);
37 }
38
```

- (1) Open the "adm\_main.c" file in admin mode.
- (2) Enable the interrupt of user mode to admin mode (U2A\_IRQ).
- (3) Actions performed in Admin\_Main. The actions in this example are to initialize system clock, delay loop function and GPIO P2.0.
- (4) Jump to user mode (USR\_Main).

```

61 | SWRST = 0x3c;
62 | }
63 |
64 |
65 |
66 | void __interrupt IntADM_Req(void)           // Admin U2A IRQ
67 | {
68 |     switch (ca.fid) {
69 |         case FID_LED_ON:
70 |             DRV_GPIO_WritePin(GPIO_P20, GPIO_PIN_LOW);    // set LED1(P2.0) to light on
71 |             break;
72 |         case FID_LED_OFF:
73 |             DRV_GPIO_WritePin(GPIO_P20, GPIO_PIN_HIGH);   // set LED1(P2.0) to light off
74 |             break;
75 |         case FID_DELAYLOOP:
76 |             DRV_Delay_ms(ca.para);                        // call delay millisecond function
77 |             break;
78 |         default:
79 |             break;
80 |     }
81 | }
82 |
83 | /* ----- end of interrupt service routine ----- */
84 |
85 | /* ----- SQ7705 interrupt vector ----- */
86 | #pragma section const ADM_INT_VECTOR
87 | void * const IntTbl1[] = {
88 |     ADM_STARTUP, /* 0x0000 : RESET */ /*non-maskable
89 |     SWI_IRQ,     /* 0x0002 : SWI_IRQ/INTUNDEF */ /*non-maskable
90 |     (void *)0xffff, /* 0x0004 : Reserved */
91 |     WDT_IRQ,     /* 0x0006 : WDT_IRQ */ /*non-maskable interrupt source:watchdog timer
92 |     OnlyReti,    /* 0x0008 : TAMPER_IRQ */ /*interrupt source:tamper detection
93 |     OnlyReti,    /* 0x000A : CFD_IRQ */ /*interrupt source:clock faildetection
94 |     OnlyReti,    /* 0x000C : Reserved */
95 |     IntADM_Req,  /* 0x000E : U2A_IRQ/IntADM_Req */ /*interrupt source:user mode to admin mode
96 |     OnlyReti,    /* 0x0010 : RTC_IRQ */ /*interrupt source:real time clock timer
97 |     OnlyReti,    /* 0x0012 : TMRA00_IRQ */ /*interrupt source:TMRA00 8-bit timer
98 |     OnlyReti,    /* 0x0014 : TMRA01_IRQ */ /*interrupt source:TMRA01 8-bit timer
99 |     OnlyReti,    /* 0x0016 : TMRA10_IRQ */ /*interrupt source:TMRA10 8-bit timer
100 |     OnlyReti,    /* 0x0018 : TMRA11_IRQ */ /*interrupt source:TMRA11 8-bit timer

```

(5) IntADM\_Req (U2A\_IRQ): This interrupt is occurred when user mode jumps to admin mode. After executing the interrupt action, it will return to the position before the user mode jump and continue execution.

In this example, the interrupt will perform corresponding actions based on the transmitted “fid” and “para”:

- If “fid” is FID\_LED\_ON, set LED1 (P2.0) to light on(low).
- If “fid” is FID\_LED\_OFF, set LED1 (P2.0) to light off(high).
- If “fid” is FID\_DELAYLOOP, call delay function and delay time is determined by “para”.

(6) Set IntADM\_Req (U2A\_IRQ) in the interrupt vector table.

## b. User mode setting

```
FileView
SQ7705_U2A.pmt
  Admin
    Source Files
      sq7705_adm_64pinpack
      adm_main.c
      _errno_decl16.c
      adm_usr_channel.c
      adm_secure_ram.c
      drv_sq7705_gpio.c
      drv_sq7705_sysclk.c
      drv_sq7705_delayloop
    Header Files
    Object Files
    Library Files
    LCF File
    Debug File
  User
    Source Files
      sq7705_usr_64pinpack
      1. usr_main.c
      _errno_decl16.c
      adm_usr_channel.c
      usr_secure_ram.c
    Header Files
    Object Files
    Library Files
    LCF File
    Debug File
  Configuration Tool

16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75

/* ----- main program ----- */
void USR_Main()
{
    while (TRUE) {
        2. ca.fid = FID_LED_ON;
           usrRequestAdm();           // Jump to Admin U2A IRQ

        ca.fid = FID_DELAYLOOP;
        ca.para = 1000;
        usrRequestAdm();           // Jump to Admin U2A IRQ

        ca.fid = FID_LED_OFF;
        usrRequestAdm();           // Jump to Admin U2A IRQ

        ca.fid = FID_DELAYLOOP;
        ca.para = 500;
        usrRequestAdm();           // Jump to Admin U2A IRQ
    }
}

/* ----- interrupt service routine ----- */
void __interrupt OnlyReti(void)
{
    __ASM("NOP");           // no operation
}

//*****
// name      : usrRequestAdm()
// description : Jump to Admin U2A IRQ
// input param :
// output param :
// retval :
// note :
//*****
3. static void usrRequestAdm(void)
{
    MODEEXC = 0x5A;
    MODEEXC = 0x33;
    MODEEXC = 0x69;
    MODEEXC = 0xF3;
}
```

(1) Open the “usr\_main.c” file in user mode.

(2) USR\_Main is the entry point for admin mode to user mode.

In this example, the following actions are performed in the while loop:

- Set “fid” to FID\_LED\_ON and jump to admin mode to execute.
- Set “fid” to FID\_DELAYLOOP and set “para” to 1000 for 1000 ms delay time, then jump to admin mode to execute.
- Set “fid” to FID\_LED\_OFF and jump to admin mode to execute.
- Set “fid” to FID\_DELAYLOOP and set “para” to 500 for 500 ms delay time, then jump to admin mode to execute.

(3) Jump to admin mode IntADM\_Req (U2A\_IRQ).