Bootloader Demo setup -- USB DFU

FRDM-KL25Z

MCU FAE David Chen

Oct.,28 , 2014

Confidential and Proprietary

USB DFU example

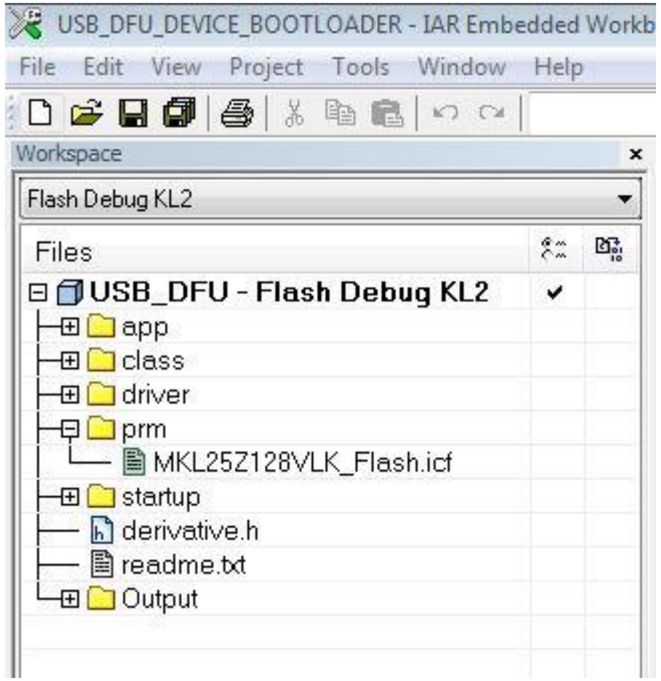
- The USB DFU example can be found from two example source .
  - USB Stack V4.1.1⊗
    - http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=MEDICALUSB&fpsp=1&tab=Design_Tools_Tab#
    - C:\Freescale\Freescale USB Stack v4.1.1\Source\Device\app\dfu
  - AN4370 document and software
    - http://cache.freescale.com/files/microcontrollers/doc/app_note/AN4370.pdf
    - http://www.freescale.com/webapp/sps/download/license.jsp?colCode=AN4370SW&location=null&Parent_nodeId=&Parent_pageType=&Parent_nodeId=&Parent_pageType=

- But USB Stack V4.1.1 USB DFU example not working .
  - There are some limitation or bug that only 264 bytes available for firmware download .

- AN4370 working well .
  - But IAR EWARM is required on KL series .

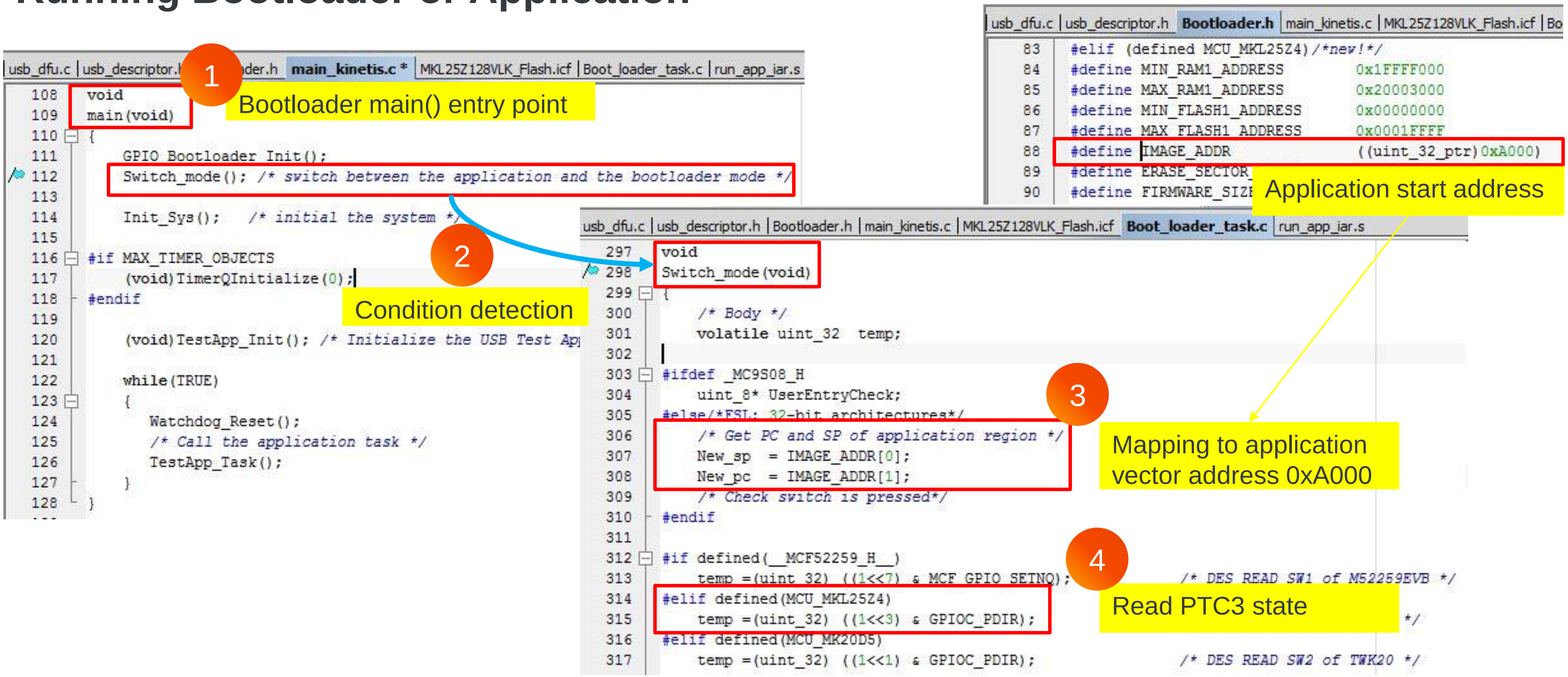
Open USB DFU project

- Project path
  - D:\AN4370SW\Source\Device\app\dfu_bootloader\iar_ew\kinetis_l25



USB_DFU_DEVICE_BOOTLOADER - IAR Embedded Workb
File Edit View Project Tools Window Help
Workspace
Flash Debug KL2
Files
USB_DFU - Flash Debug KL2 ✔
app
class
driver
prm
MKL25Z128VLK_Flash.icf
startup
derivative.h
readme.txt
Output

Running Bootloader or Application


usb_dfu.c | usb_descriptor.h | Bootloader.h | main_kinetis.c | MKL25Z128VLK_Flash.icf | Bo
usb_dfu.c | usb_descriptor.h | Bootloader.h | main_kinetis.c * | MKL25Z128VLK_Flash.icf | Boot_loader_task.c | run_app_iar.s
usb_dfu.c | usb_descriptor.h | Bootloader.h | main_kinetis.c | MKL25Z128VLK_Flash.icf | Boot_loader_task.c | run_app_iar.s
1
2
3
4

```
void
main(void)
{
    GPIO_Bootloader_Init();
    Switch_mode(); /* switch between the application and the bootloader mode */

    Init_Sys();   /* initial the system */

#if MAX_TIMER_OBJECTS
    (void)TimerQInitialize(0);
#endif

    (void)TestApp_Init(); /* Initialize the USB Test Ap

    while(TRUE)
    {
       Watchdog_Reset();
       /* Call the application task */
       TestApp_Task();
    }
}
```

1 Bootloader main() entry point

2 Condition detection

```
#elif  (defined MCU_MKL25Z4)/*new!*/
#define MIN_RAM1_ADDRESS            0x1FFFF000
#define MAX_RAM1_ADDRESS            0x20003000
#define MIN_FLASH1_ADDRESS          0x00000000
#define MAX_FLASH1_ADDRESS          0x0001FFFF
#define IMAGE_ADDR                  ((uint_32_ptr)0xA000)
#define ERASE_SECTOR
#define FIRMWARE_SIZE
```

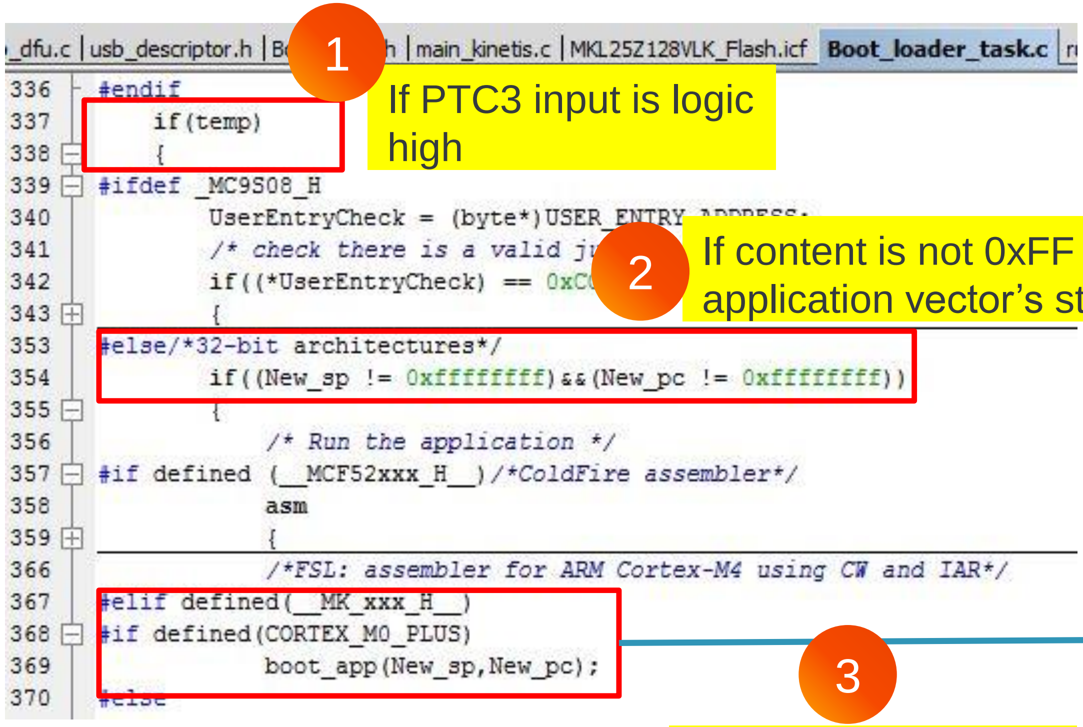
Application start address

```
void
Switch_mode(void)
{
    /* Body */
    volatile uint_32  temp;

#ifdef _MC9S08_H
    uint_8* UserEntryCheck;
#else/*FSL: 32-bit architectures*/
    /* Get PC and SP of application region */
    New_sp  = IMAGE_ADDR[0];
    New_pc  = IMAGE_ADDR[1];
    /* Check switch is pressed*/
#endif

#if defined(__MCF52259_H__)
    temp =(uint_32) ((1<<7) & MCF_GPIO_SETNQ);       /* DES READ SW1 of M52259EVB */
#elif defined(MCU_MKL25Z4)
    temp =(uint_32) ((1<<3) & GPIOC_PDIR);                                       */
#elif defined(MCU_MK20D5)
    temp =(uint_32) ((1<<1) & GPIOC_PDIR);           /* DES READ SW2 of TWK20 */
```

3 Mapping to application vector address 0xA000

4 Read PTC3 state

Bootloader or Application -- Continues

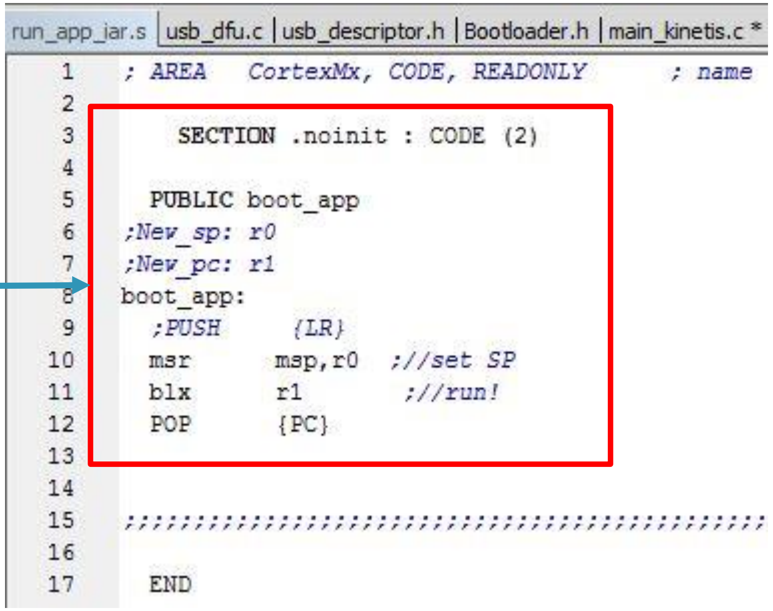

_dfu.c | usb_descriptor.h | Bo h | main_kinetis.c | MKL25Z128VLK_Flash.icf | Boot_loader_task.c | r

```
#endif
    if(temp)
    {
#ifdef _MC9S08_H
        UserEntryCheck = (byte*)USER_ENTRY_ADDRESS;
        /* check there is a valid ju
        if((*UserEntryCheck) == 0xC
        {
#else/*32-bit architectures*/
        if((New_sp != 0xffffffff)&&(New_pc != 0xffffffff))
        {
            /* Run the application */
#if defined (__MCF52xxx_H__)/*ColdFire assembler*/
            asm
            {
            /*FSL: assembler for ARM Cortex-M4 using CW and IAR*/
#elif defined(__MK_xxx_H__)
#if defined(CORTEX_M0_PLUS)
            boot_app(New_sp,New_pc);
#else
```

1 If PTC3 input is logic high

2 If content is not 0xFF in application vector's start address

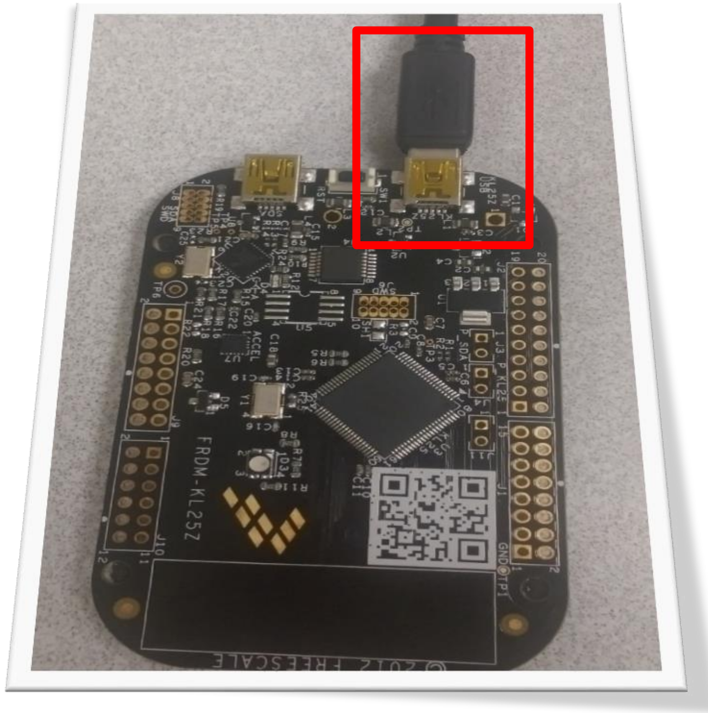
3 Configure PC to jump to application

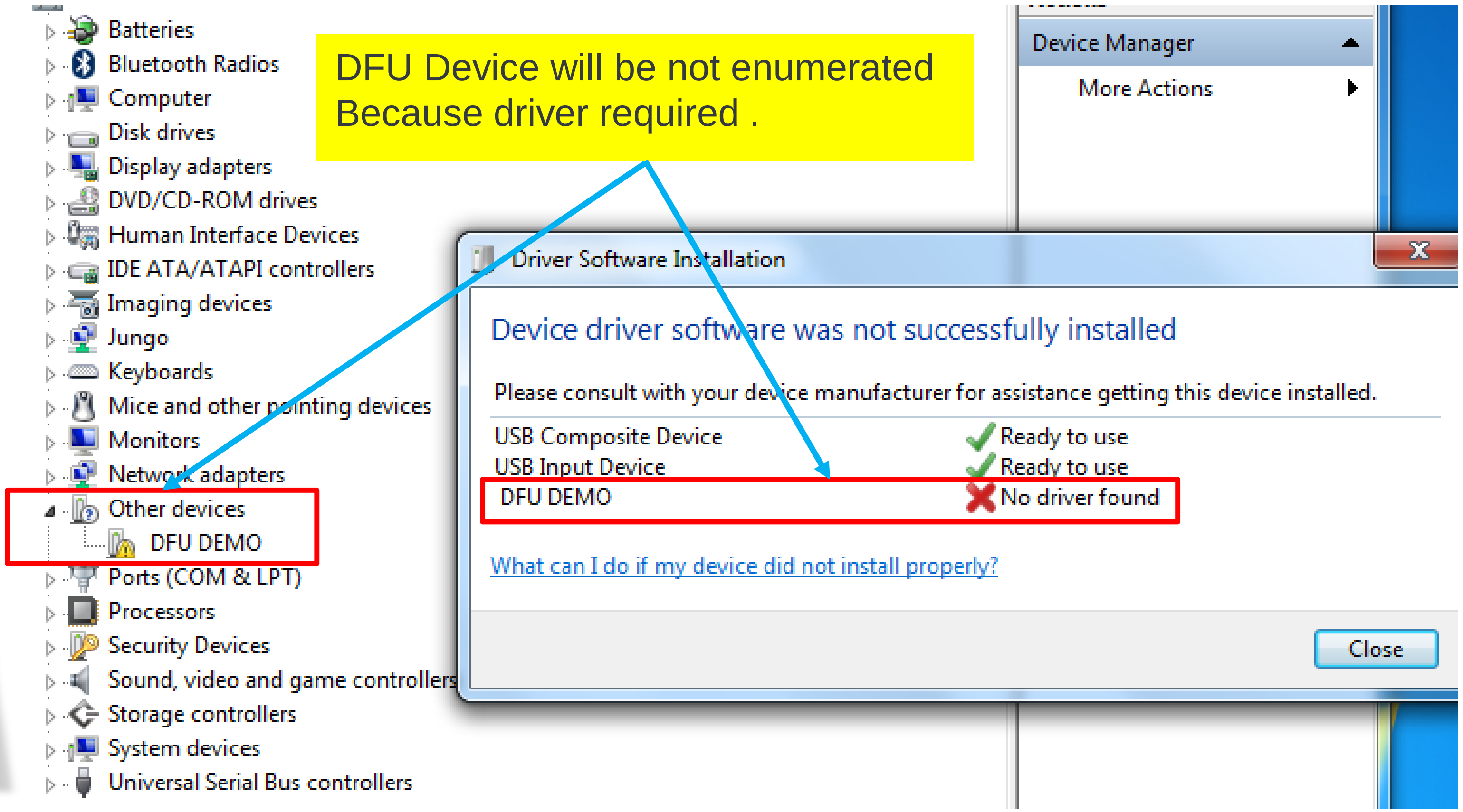

run_app_iar.s | usb_dfu.c | usb_descriptor.h | Bootloader.h | main_kinetis.c *

```
; AREA    CortexMx, CODE, READONLY        ; name

    SECTION .noinit : CODE (2)

  PUBLIC boot_app
;New_sp: r0
;New_pc: r1
boot_app:
  ;PUSH      {LR}
  msr       msp,r0    ;//set SP
  blx       r1         ;//run!
  POP       {PC}


;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

  END
```

Windows USB Device enumeration

- After download DFU FW , please plug in USB to PC .
- USB3.0 Host is not supported .




DFU Device will be not enumerated
Because driver required .

- Batteries
- Bluetooth Radios
- Computer
- Disk drives
- Display adapters
- DVD/CD-ROM drives
- Human Interface Devices
- IDE ATA/ATAPI controllers
- Imaging devices
- Jungo
- Keyboards
- Mice and other pointing devices
- Monitors
- Network adapters
- Other devices
  - DFU DEMO
- Ports (COM & LPT)
- Processors
- Security Devices
- Sound, video and game controllers
- Storage controllers
- System devices
- Universal Serial Bus controllers

Driver Software Installation

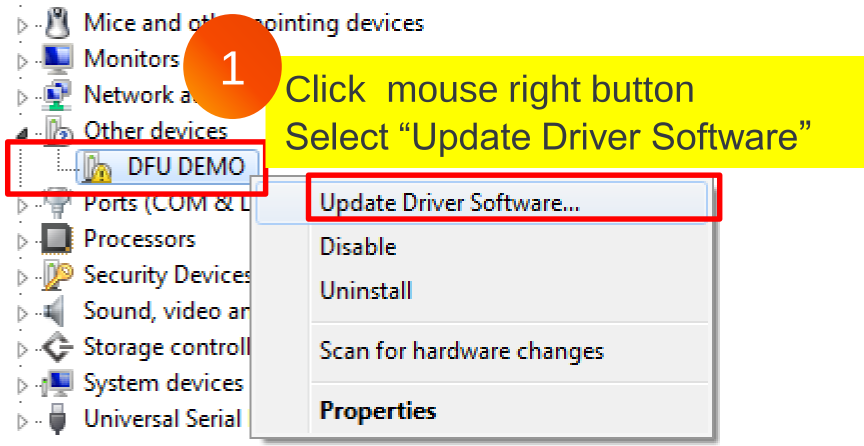
Device driver software was not successfully installed

Please consult with your device manufacturer for assistance getting this device installed.

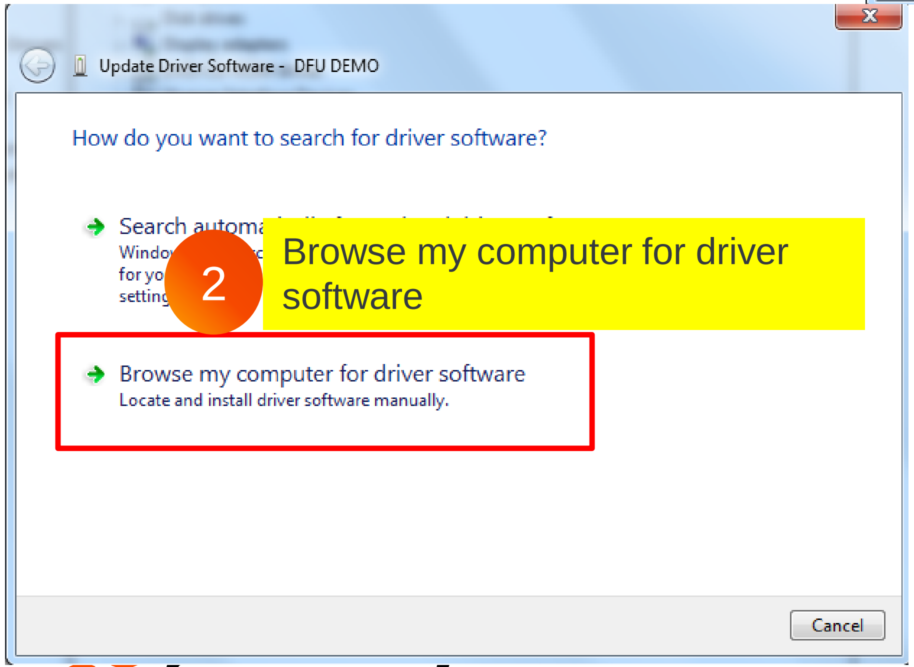
USB Composite Device	✓ Ready to use
USB Input Device	✓ Ready to use
DFU DEMO	✗ No driver found

What can I do if my device did not install properly?

Update Driver



1 Click mouse right button
Select “Update Driver Software”

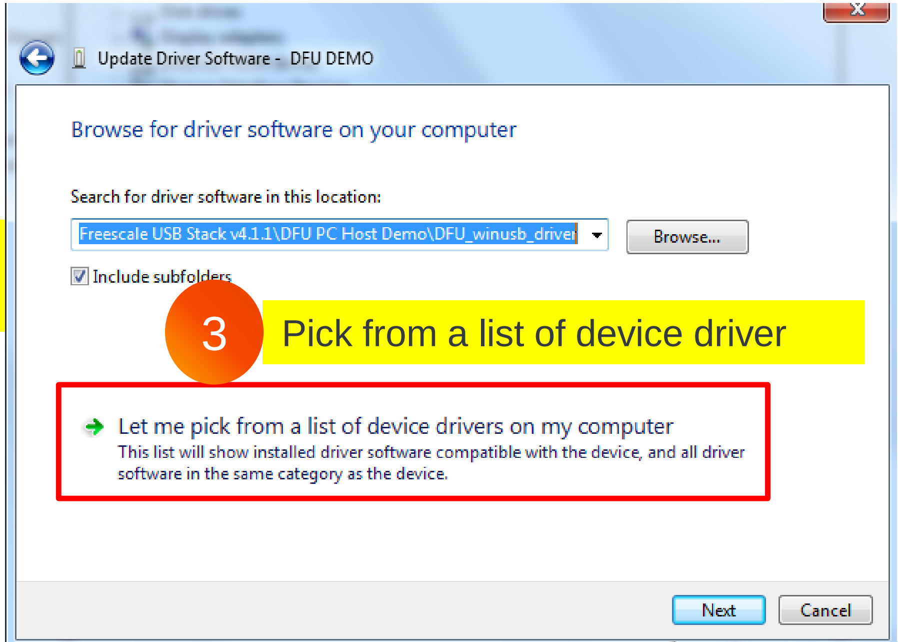


How do you want to search for driver software?

Search automa
Windo
for yo
setting

Browse my computer for driver software
Locate and install driver software manually.

2 Browse my computer for driver software



Browse for driver software on your computer

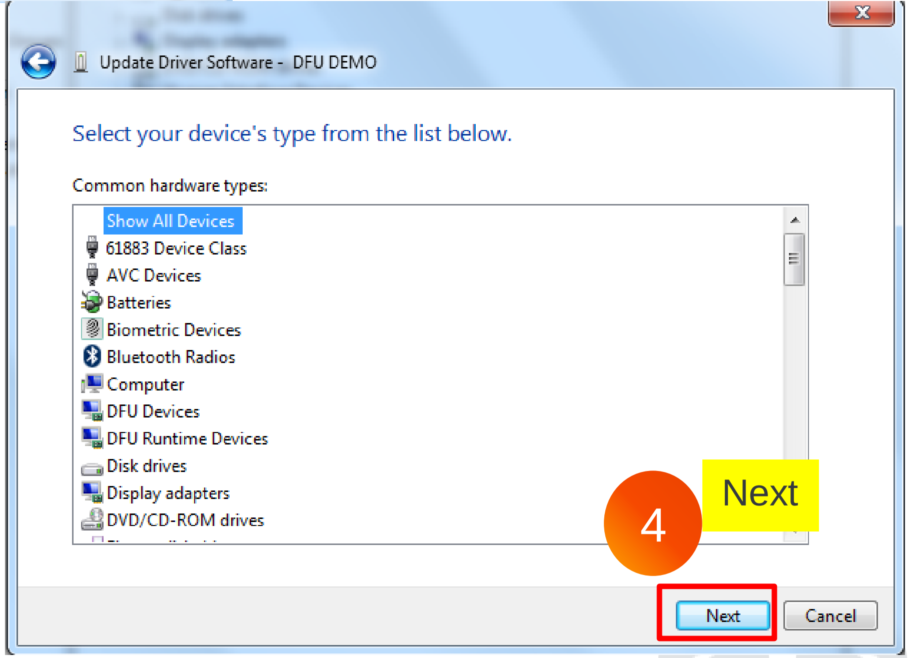
Search for driver software in this location:

Freescale USB Stack v4.1.1\DFU PC Host Demo\DFU_winusb_driver

☑ Include subfolders

Let me pick from a list of device drivers on my computer
This list will show installed driver software compatible with the device, and all driver software in the same category as the device.

3 Pick from a list of device driver



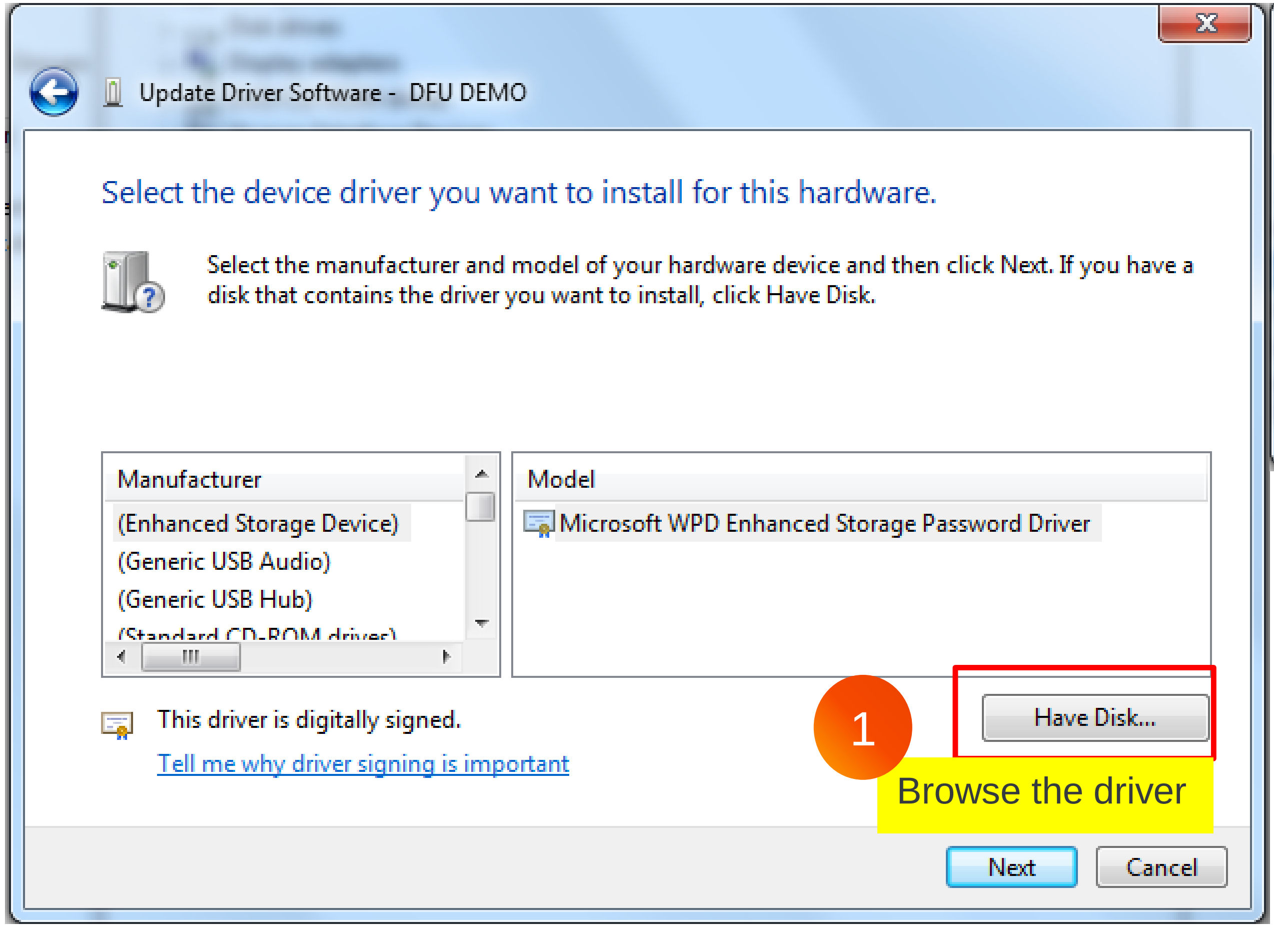
Select your device's type from the list below.

Common hardware types:

- Show All Devices
- 61883 Device Class
- AVC Devices
- Batteries
- Biometric Devices
- Bluetooth Radios
- Computer
- DFU Devices
- DFU Runtime Devices
- Disk drives
- Display adapters
- DVD/CD-ROM drives

4 Next

Update Driver – Continues



Update Driver Software - DFU DEMO

Select the device driver you want to install for this hardware.

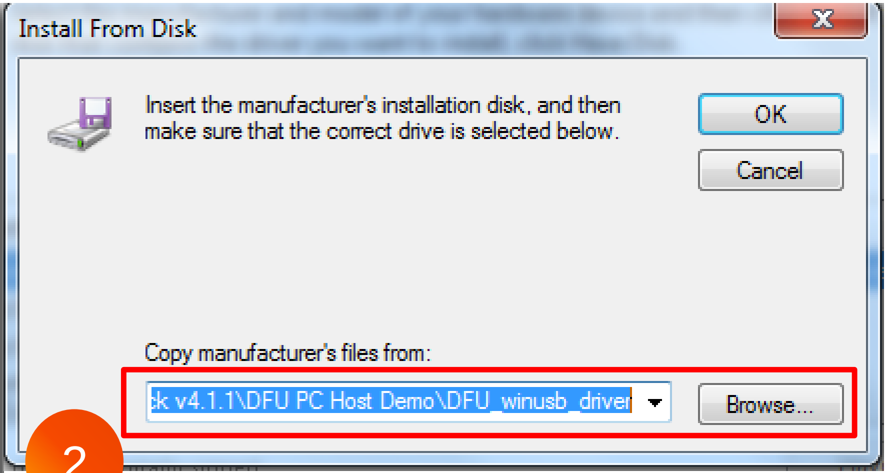
Select the manufacturer and model of your hardware device and then click Next. If you have a disk that contains the driver you want to install, click Have Disk.

Manufacturer	Model
(Enhanced Storage Device)	Microsoft WPD Enhanced Storage Password Driver
(Generic USB Audio)	
(Generic USB Hub)	
(Standard CD-ROM drives)	

This driver is digitally signed.

Tell me why driver signing is important

1 Browse the driver



Install From Disk

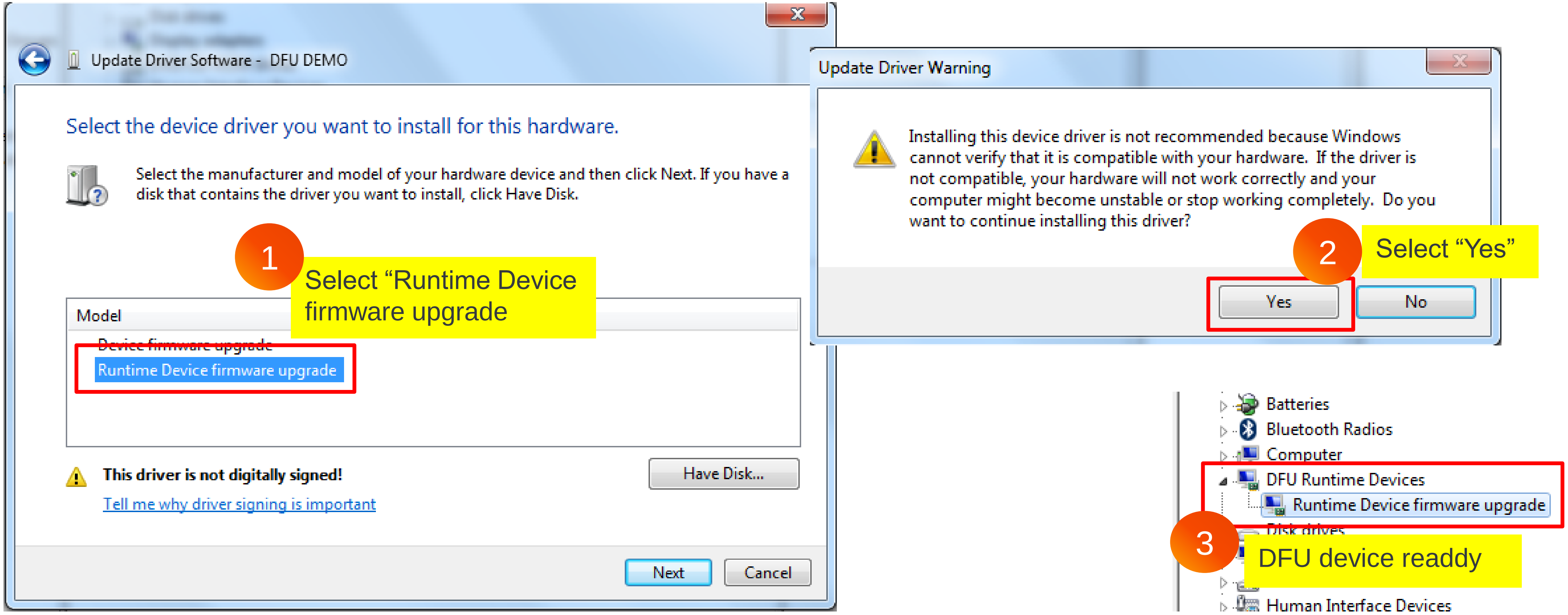
Insert the manufacturer's installation disk, and then make sure that the correct drive is selected below.

Copy manufacturer's files from:

:k v4.1.1\DFU PC Host Demo\DFU_winusb_driver

2 The driver path D:\AN4370SW_20121007\AN4370SW\DFU_winusb_driver

Update Driver – Continues



Update Driver Software - DFU DEMO

Select the device driver you want to install for this hardware.

Select the manufacturer and model of your hardware device and then click Next. If you have a disk that contains the driver you want to install, click Have Disk.

Model

Device firmware upgrade

Runtime Device firmware upgrade

This driver is not digitally signed!

Tell me why driver signing is important

1 Select “Runtime Device firmware upgrade

Update Driver Warning

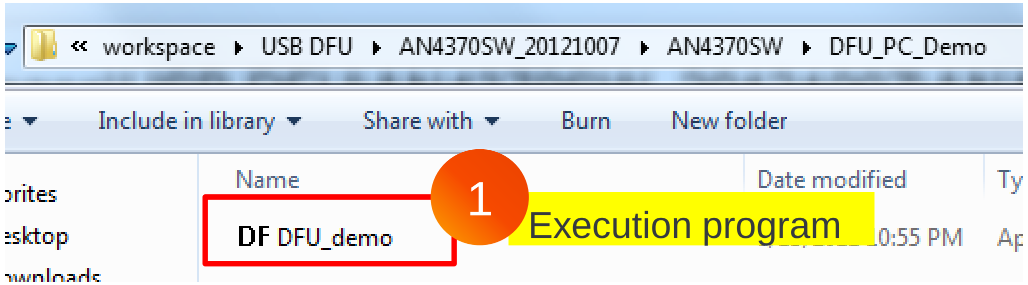
Installing this device driver is not recommended because Windows cannot verify that it is compatible with your hardware. If the driver is not compatible, your hardware will not work correctly and your computer might become unstable or stop working completely. Do you want to continue installing this driver?

2 Select “Yes”

3 DFU device readdy

Open DFU PC program & into DFU mode

- D:\AN4370SW\DFU_PC_Demo

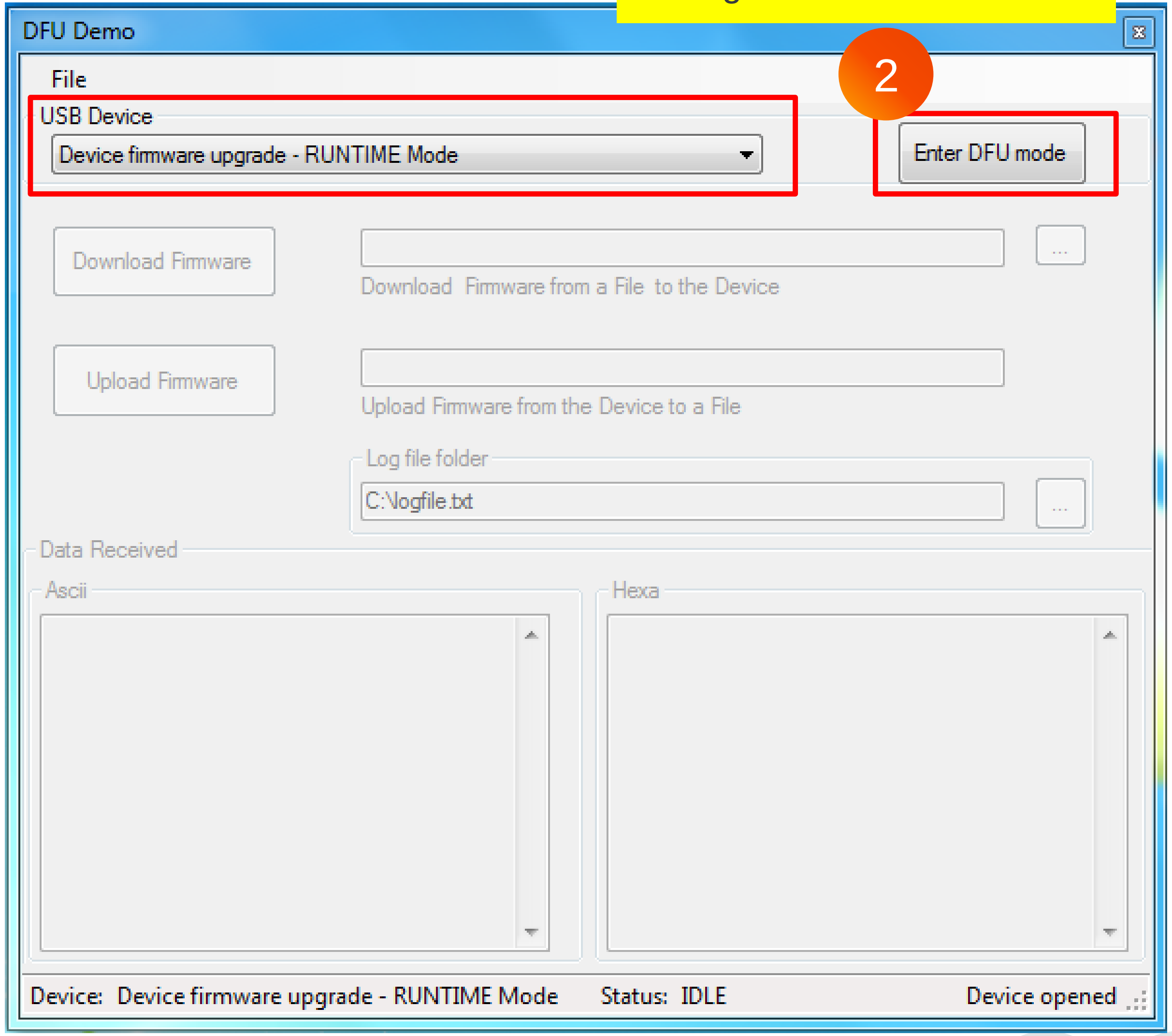


« workspace ▸ USB DFU ▸ AN4370SW_20121007 ▸ AN4370SW ▸ DFU_PC_Demo

Name	Date modified	Ty
DFU_demo	0:55 PM	Ap

1 Execution program

Change USB to DFU mode



DFU Demo
File
Download Firmware
Download Firmware from a File to the Device
Upload Firmware
Upload Firmware from the Device to a File
Data Received
Ascii
Hexa

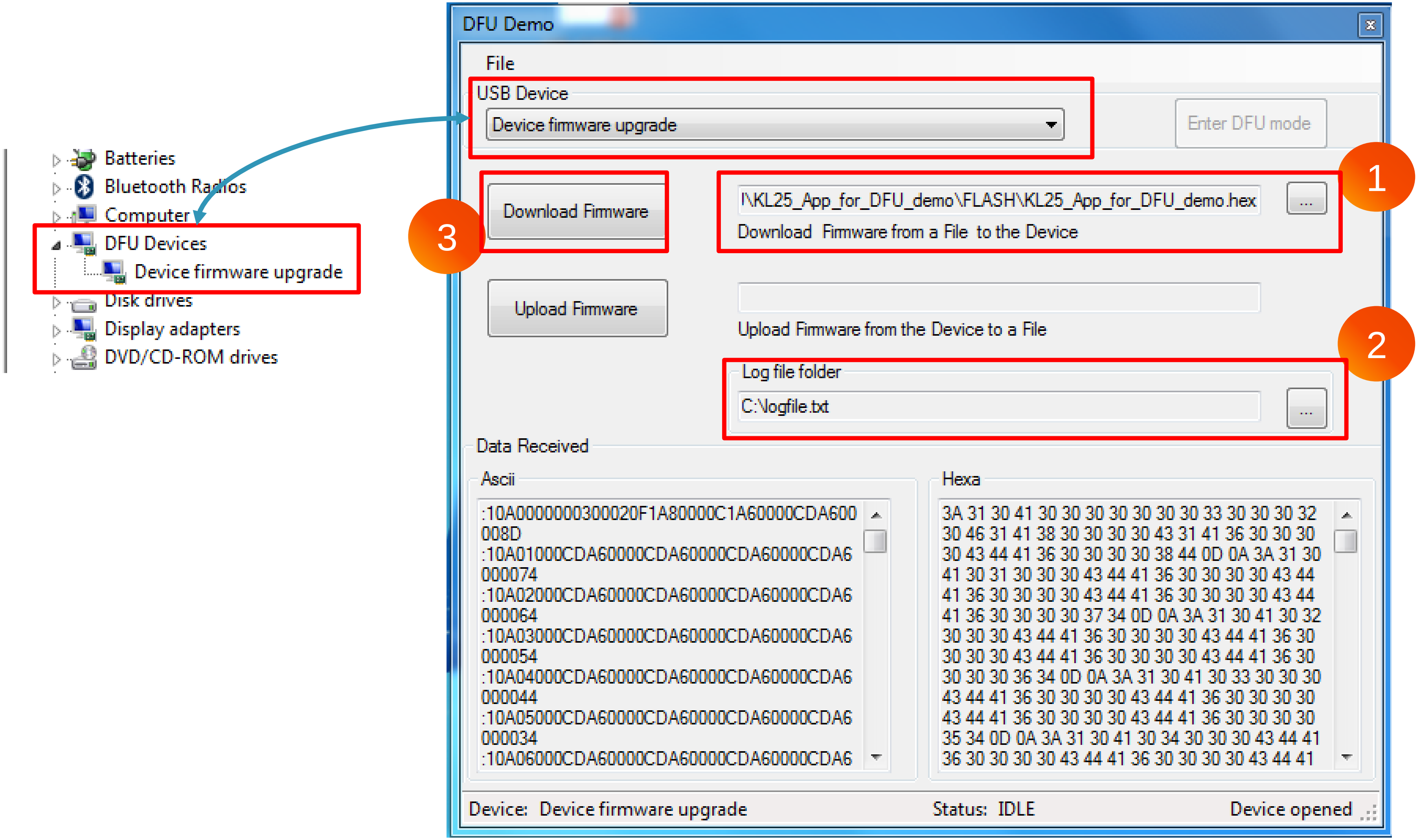
USB Device: Device firmware upgrade - RUNTIME Mode

2 Enter DFU mode

Log file folder: C:\logfile.txt

Device: Device firmware upgrade - RUNTIME Mode Status: IDLE Device opened

Programming application image



I:\KL25_App_for_DFU_demo\FLASH\KL25_App_for_DFU_demo.hex

Download Firmware from a File to the Device

Upload Firmware from the Device to a File

Log file folder: C:\logfile.txt

```
:10A0000000300020F1A80000C1A60000CDA600
008D
:10A01000CDA60000CDA60000CDA60000CDA6
000074
:10A02000CDA60000CDA60000CDA60000CDA6
000064
:10A03000CDA60000CDA60000CDA60000CDA6
000054
:10A04000CDA60000CDA60000CDA60000CDA6
000044
:10A05000CDA60000CDA60000CDA60000CDA6
000034
:10A06000CDA60000CDA60000CDA60000CDA6
```

```
3A 31 30 41 30 30 30 30 30 30 30 33 30 30 30 32
30 46 31 41 38 30 30 30 30 43 31 41 36 30 30 30
30 43 44 41 36 30 30 30 30 38 44 0D 0A 3A 31 30
41 30 31 30 30 30 43 44 41 36 30 30 30 30 43 44
41 36 30 30 30 30 43 44 41 36 30 30 30 30 43 44
41 36 30 30 30 30 37 34 0D 0A 3A 31 30 41 30 32
30 30 30 43 44 41 36 30 30 30 30 43 44 41 36 30
30 30 30 43 44 41 36 30 30 30 30 43 44 41 36 30
30 30 30 36 34 0D 0A 3A 31 30 41 30 33 30 30 30
43 44 41 36 30 30 30 30 43 44 41 36 30 30 30 30
43 44 41 36 30 30 30 30 43 44 41 36 30 30 30 30
35 34 0D 0A 3A 31 30 41 30 34 30 30 30 43 44 41
36 30 30 30 30 43 44 41 36 30 30 30 30 43 44 41
```

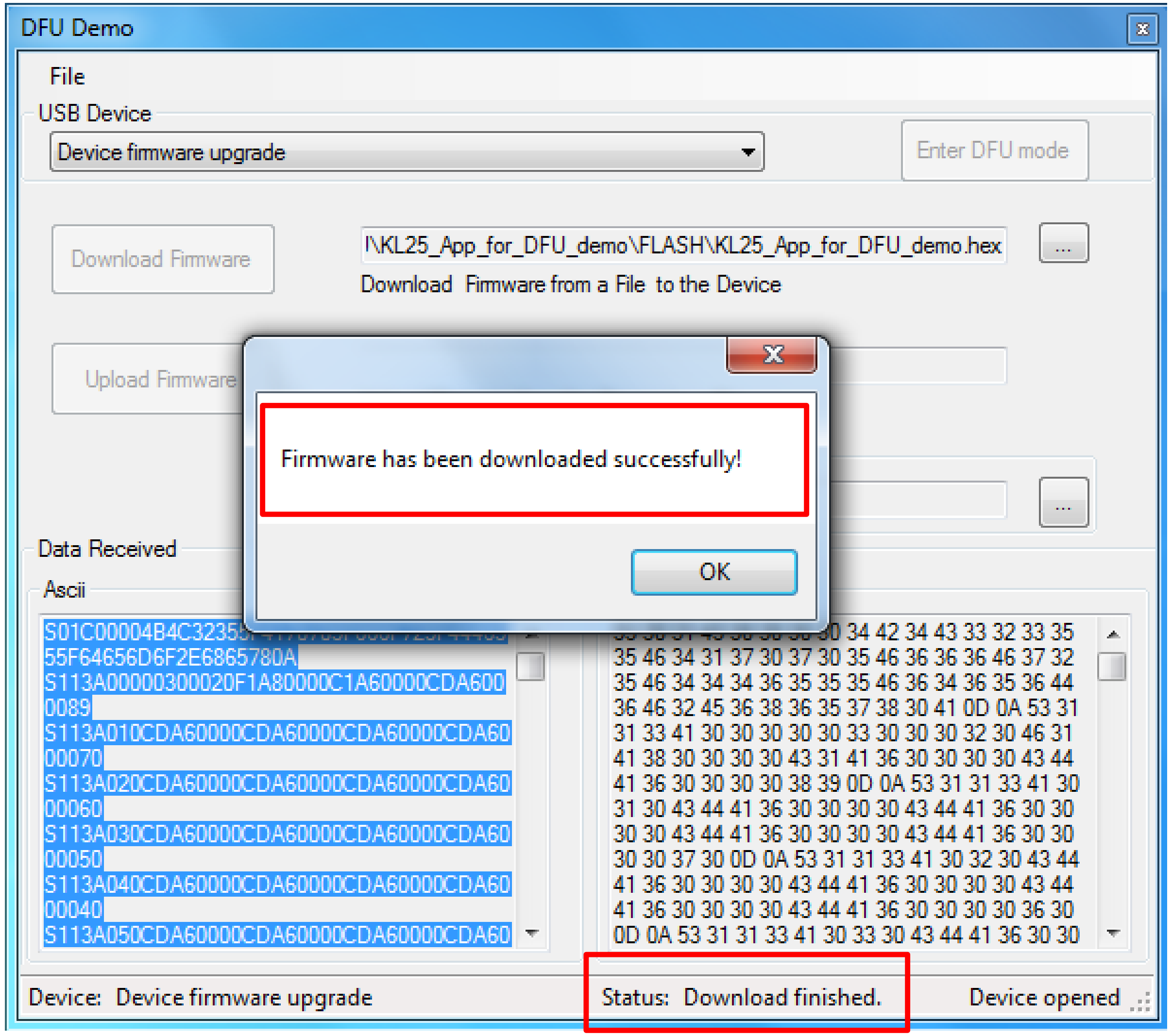
Device: Device firmware upgrade   Status: IDLE   Device opened

Browse your application file , it support s19 and bin files .

Log file to record programming state

- Bootloader driver: Parses firmware image files and flash them to flash memory. The bootloader driver supports parsing image files in CodeWarrior binary, S19, and raw binary file formats.

Programming successfully



DFU Demo

File

USB Device

Device firmware upgrade

Enter DFU mode

Download Firmware

l\KL25_App_for_DFU_demo\FLASH\KL25_App_for_DFU_demo.hex

Download Firmware from a File to the Device

Upload Firmware

Firmware has been downloaded successfully!

OK

Data Received

Ascii

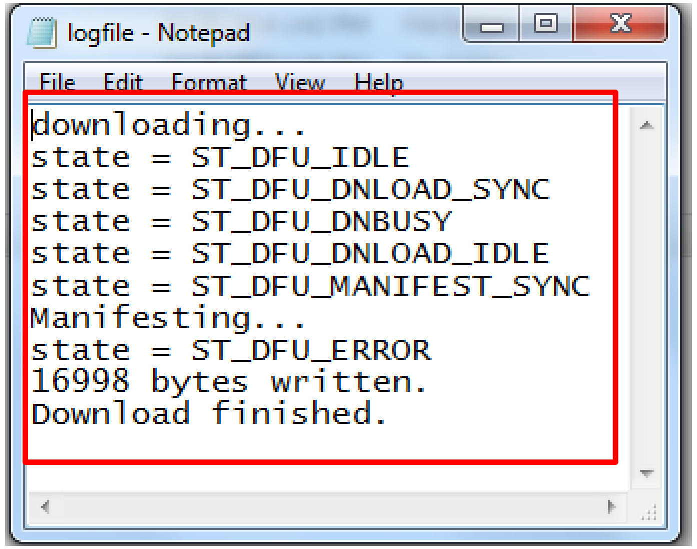
```
S01C00004B4C32355
55F64656D6F2E6865780A
S113A0000300020F1A80000C1A60000CDA600
0089
S113A010CDA60000CDA60000CDA60000CDA60
00070
S113A020CDA60000CDA60000CDA60000CDA60
00060
S113A030CDA60000CDA60000CDA60000CDA60
00050
S113A040CDA60000CDA60000CDA60000CDA60
00040
S113A050CDA60000CDA60000CDA60000CDA60
```

```
30 34 42 34 43 33 32 33 35
35 46 34 31 37 30 37 30 35 46 36 36 36 46 37 32
35 46 34 34 34 36 35 35 35 46 36 34 36 35 36 44
36 46 32 45 36 38 36 35 37 38 30 41 0D 0A 53 31
31 33 41 30 30 30 30 30 33 30 30 30 32 30 46 31
41 38 30 30 30 30 43 31 41 36 30 30 30 30 43 44
41 36 30 30 30 30 38 39 0D 0A 53 31 31 33 41 30
31 30 43 44 41 36 30 30 30 30 43 44 41 36 30 30
30 30 43 44 41 36 30 30 30 30 43 44 41 36 30 30
30 30 37 30 0D 0A 53 31 31 33 41 30 32 30 43 44
41 36 30 30 30 30 43 44 41 36 30 30 30 30 43 44
41 36 30 30 30 30 43 44 41 36 30 30 30 30 36 30
0D 0A 53 31 31 33 41 30 33 30 43 44 41 36 30 30
```

Device: Device firmware upgrade

Status: Download finished.

Device opened



logfile - Notepad

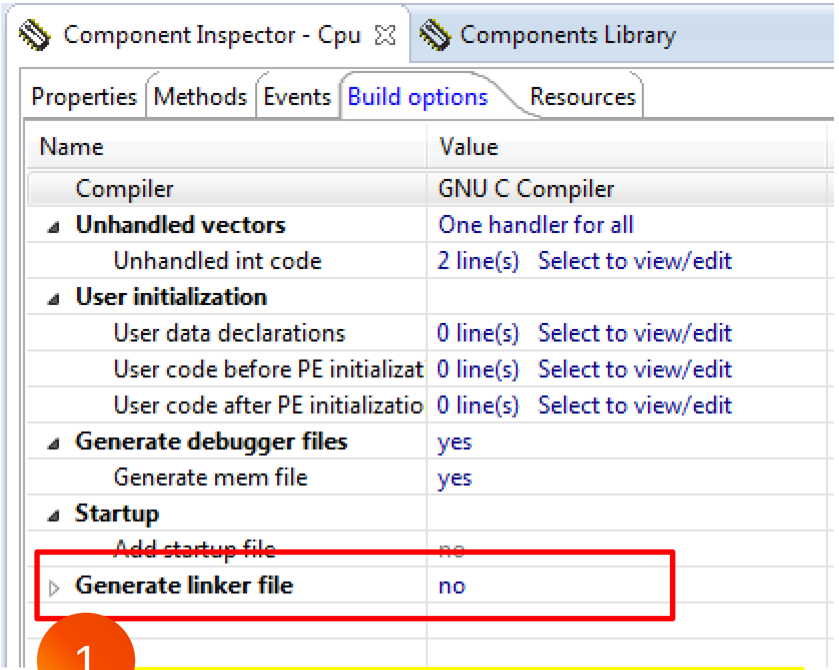
File Edit Format View Help

```
downloading...
state = ST_DFU_IDLE
state = ST_DFU_DNLOAD_SYNC
state = ST_DFU_DNBUSY
state = ST_DFU_DNLOAD_IDLE
state = ST_DFU_MANIFEST_SYNC
Manifesting...
state = ST_DFU_ERROR
16998 bytes written.
Download finished.
```

Log file to record programming state

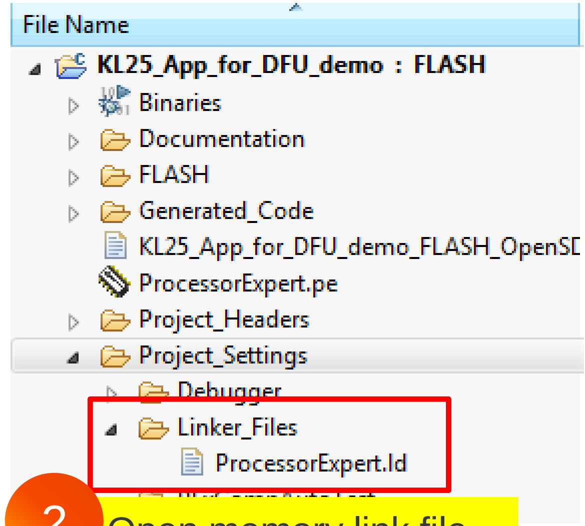
Application code

Arrange memory



Name	Value
Compiler	GNU C Compiler
Unhandled vectors	One handler for all
Unhandled int code	2 line(s) Select to view/edit
User initialization	
User data declarations	0 line(s) Select to view/edit
User code before PE initializat	0 line(s) Select to view/edit
User code after PE initializatio	0 line(s) Select to view/edit
Generate debugger files	yes
Generate mem file	yes
Startup	
Add startup file	no
Generate linker file	no

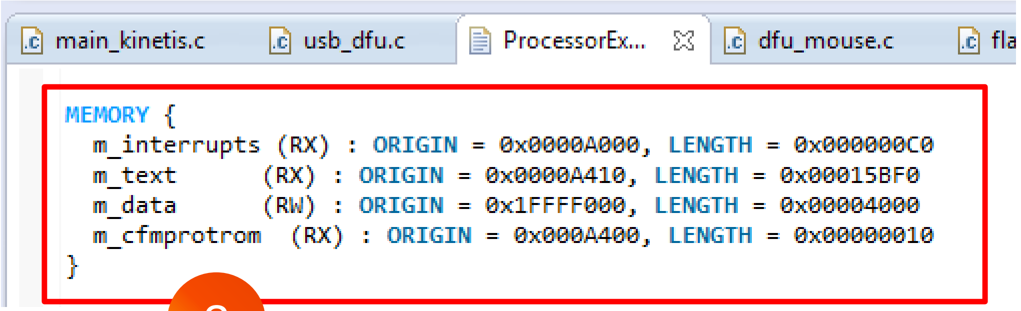
1 If Processor Expert is engaged
Please disable “Generate linker file”



```
File Name
KL25_App_for_DFU_demo : FLASH
  Binaries
  Documentation
  FLASH
  Generated_Code
  KL25_App_for_DFU_demo_FLASH_OpenSD
  ProcessorExpert.pe
  Project_Headers
  Project_Settings
    Debugger
    Linker_Files
      ProcessorExpert.ld
```

2 Open memory link file

- m_interrupts start from 0xA000
- m_text start from 0xA410
- m_cfmprotrom start from 0xA400

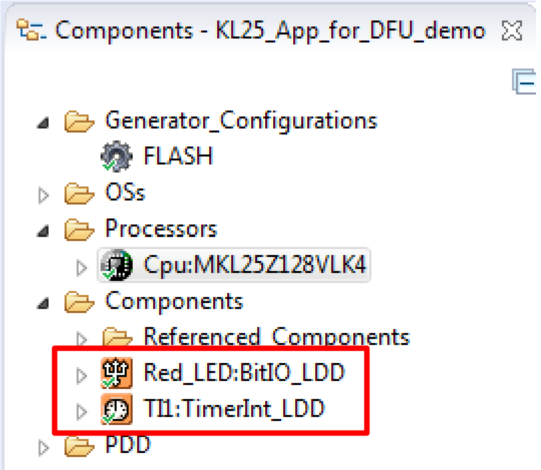


```
MEMORY {
  m_interrupts (RX) : ORIGIN = 0x0000A000, LENGTH = 0x000000C0
  m_text      (RX) : ORIGIN = 0x0000A410, LENGTH = 0x00015BF0
  m_data      (RW) : ORIGIN = 0x1FFFF000, LENGTH = 0x00004000
  m_cfmprotrom  (RX) : ORIGIN = 0x000A400, LENGTH = 0x00000010
}
```

3 Shift flash start address to 0xA000 .

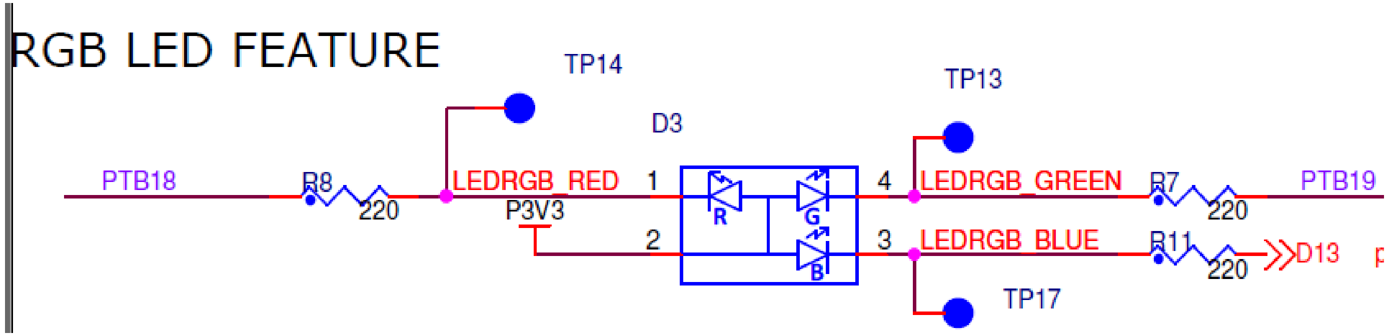
Coding application

- GPIO control LED – prove the application is working
- A periodic interrupt – prove interrupt function is working



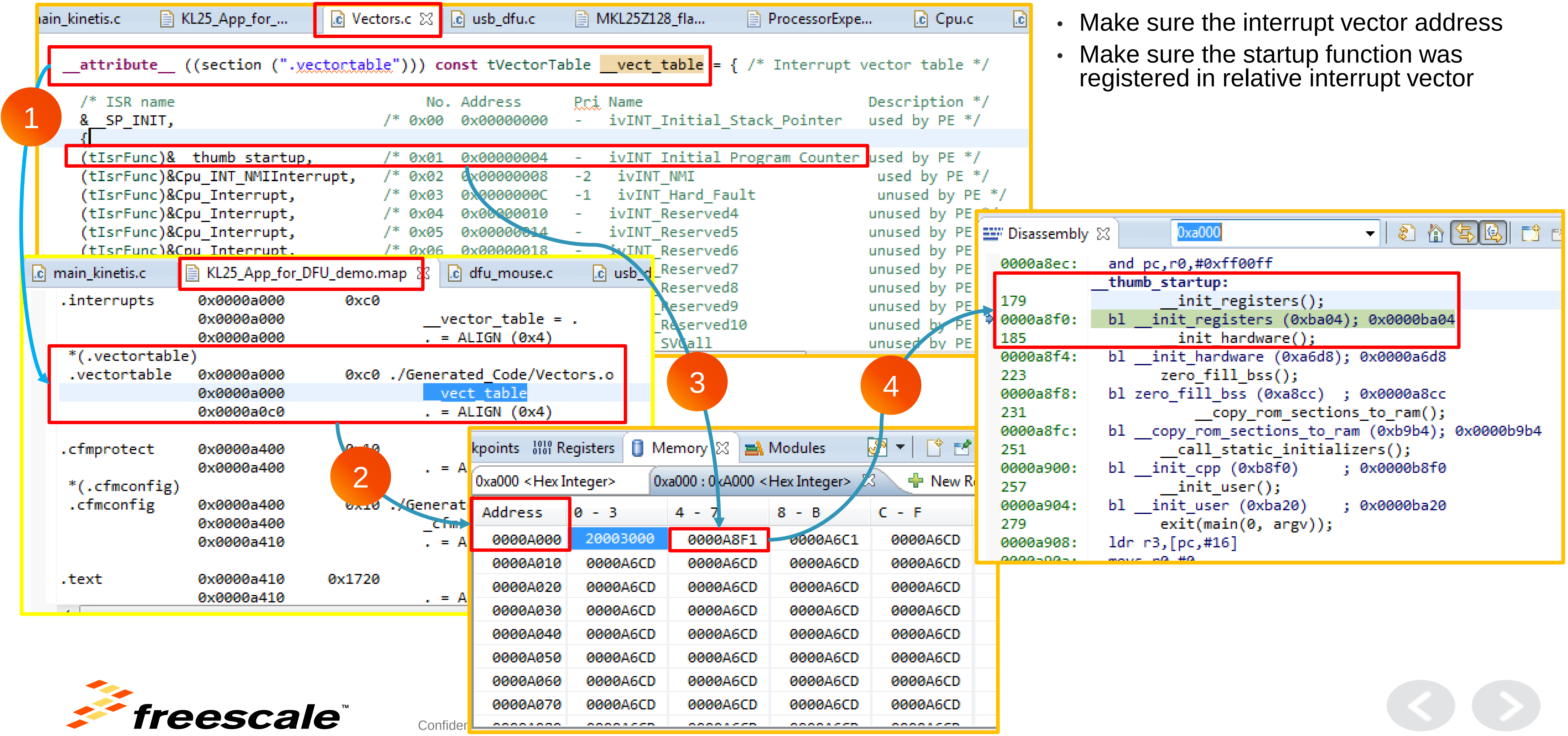
Components - KL25_App_for_DFU_demo

- Generator_Configurations
  - FLASH
- OSs
- Processors
  - Cpu:MKL25Z128VLK4
- Components
  - Referenced_Components
  - Red_LED:BitIO_LDD
  - TI1:TimerInt_LDD
- PDD



Verify memory mapping – start up

- Make sure the interrupt vector address
- Make sure the startup function was registered in relative interrupt vector



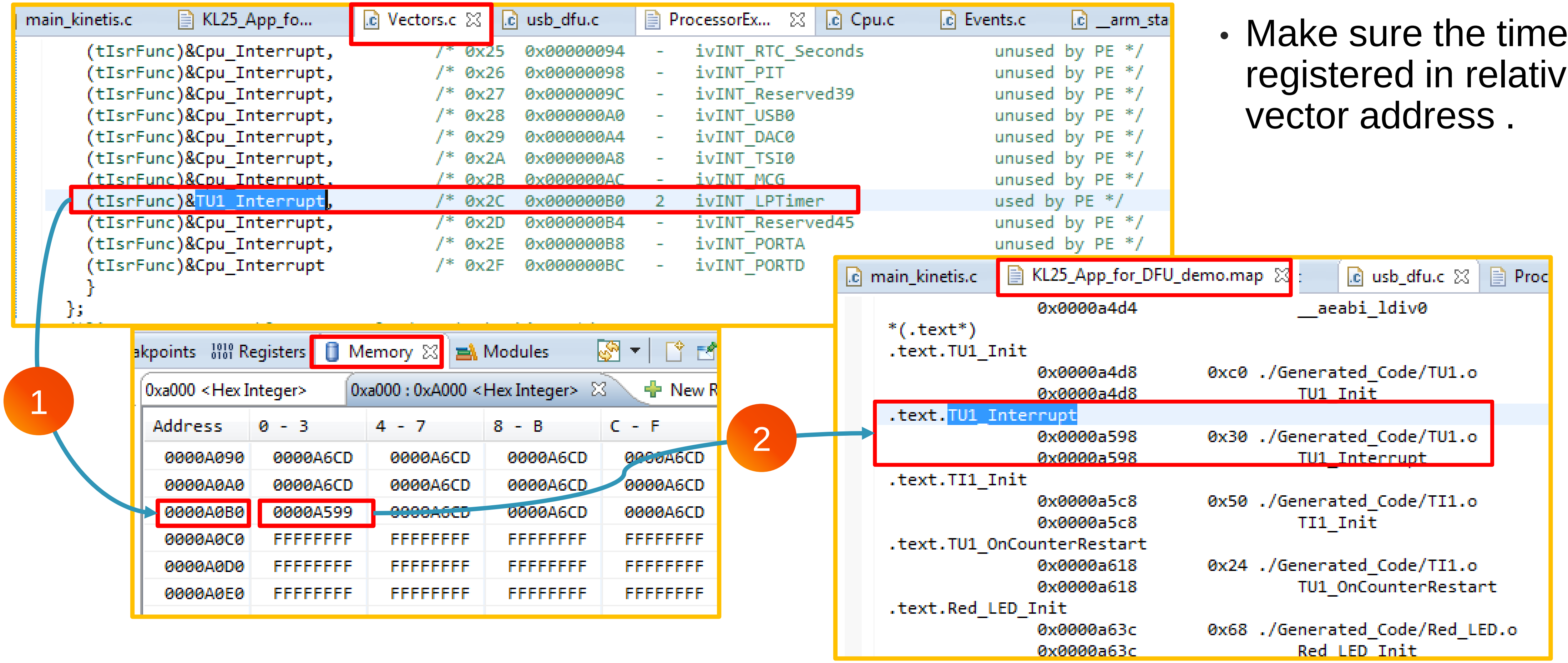
```
__attribute__ ((section (".vectortable"))) const tVectorTable __vect_table = { /* Interrupt vector table */

  /* ISR name                             No.  Address     Pri Name                           Description */
  &__SP_INIT,                          /* 0x00  0x00000000  -   ivINT_Initial_Stack_Pointer    used by PE */
  {
  (tIsrFunc)&__thumb_startup,          /* 0x01  0x00000004  -   ivINT_Initial_Program_Counter  used by PE */
  (tIsrFunc)&Cpu_INT_NMIInterrupt,     /* 0x02  0x00000008  -2  ivINT_NMI                       used by PE */
  (tIsrFunc)&Cpu_Interrupt,            /* 0x03  0x0000000C  -1  ivINT_Hard_Fault                unused by PE */
  (tIsrFunc)&Cpu_Interrupt,            /* 0x04  0x00000010  -   ivINT_Reserved4                unused by PE */
  (tIsrFunc)&Cpu_Interrupt,            /* 0x05  0x00000014  -   ivINT_Reserved5                unused by PE
  (tIsrFunc)&Cpu_Interrupt,            /* 0x06  0x00000018  -   ivINT_Reserved6                unused by PE
                                                                 Reserved7                      unused by PE
                                                                 Reserved8                      unused by PE
                                                                 Reserved9                      unused by PE
                                                                 Reserved10                     unused by PE
                                                                 SVCall                         unused by PE
```

```
 .interrupts     0x0000a000      0xc0
                 0x0000a000                __vector_table = .
                 0x0000a000                . = ALIGN (0x4)
  *(.vectortable)
  .vectortable   0x0000a000      0xc0 ./Generated_Code/Vectors.o
                 0x0000a000                __vect_table
                 0x0000a0c0                . = ALIGN (0x4)

 .cfmprotect     0x0000a400      0x10
                 0x0000a400                . = A
  *(.cfmconfig)
  .cfmconfig     0x0000a400      0x10 ./Generat
                 0x0000a400                _cfm
                 0x0000a410                . = A

 .text           0x0000a410      0x1720
                 0x0000a410                . = A
```

Address	0 - 3	4 - 7	8 - B	C - F
0000A000	20003000	0000A8F1	0000A6C1	0000A6CD
0000A010	0000A6CD	0000A6CD	0000A6CD	0000A6CD
0000A020	0000A6CD	0000A6CD	0000A6CD	0000A6CD
0000A030	0000A6CD	0000A6CD	0000A6CD	0000A6CD
0000A040	0000A6CD	0000A6CD	0000A6CD	0000A6CD
0000A050	0000A6CD	0000A6CD	0000A6CD	0000A6CD
0000A060	0000A6CD	0000A6CD	0000A6CD	0000A6CD
0000A070	0000A6CD	0000A6CD	0000A6CD	0000A6CD

```
0000a8ec:   and pc,r0,#0xff00ff
          __thumb_startup:
                 __init_registers();
0000a8f0:   bl __init_registers (0xba04); 0x0000ba04
                 init_hardware();
0000a8f4:   bl __init_hardware (0xa6d8); 0x0000a6d8
               zero_fill_bss();
0000a8f8:   bl zero_fill_bss (0xa8cc)   ; 0x0000a8cc
                    __copy_rom_sections_to_ram();
0000a8fc:   bl __copy_rom_sections_to_ram (0xb9b4); 0x0000b9b4
                 __call_static_initializers();
0000a900:   bl __init_cpp (0xb8f0)      ; 0x0000b8f0
                 __init_user();
0000a904:   bl __init_user (0xba20)     ; 0x0000ba20
               exit(main(0, argv));
0000a908:   ldr r3,[pc,#16]
```

Verify memory mapping – interrupt ISR



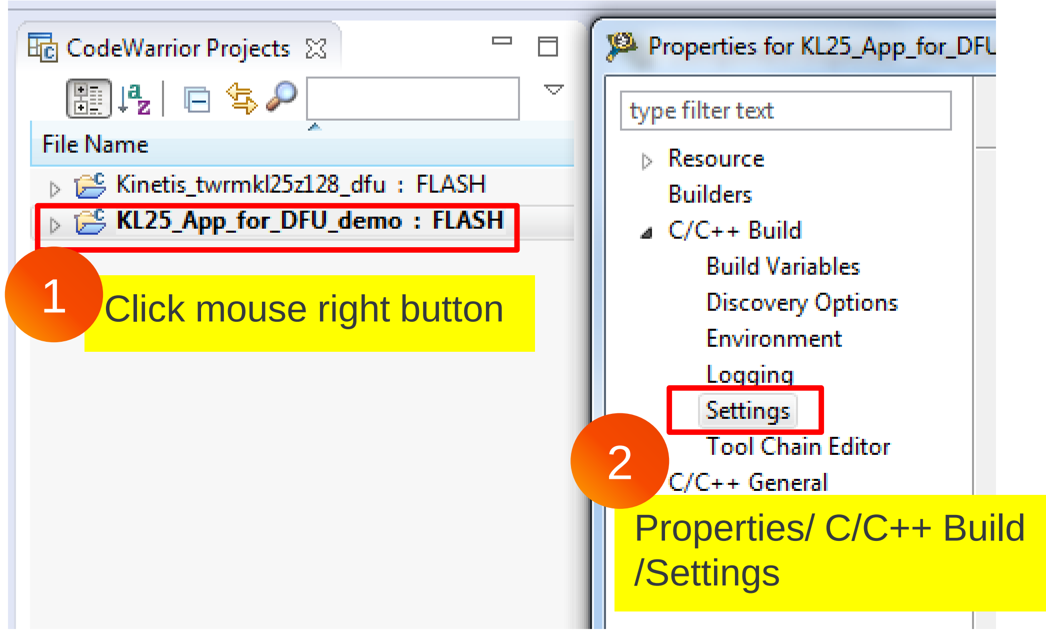
```
    (tIsrFunc)&Cpu_Interrupt,          /* 0x25  0x00000094   -   ivINT_RTC_Seconds                 unused by PE */
    (tIsrFunc)&Cpu_Interrupt,          /* 0x26  0x00000098   -   ivINT_PIT                         unused by PE */
    (tIsrFunc)&Cpu_Interrupt,          /* 0x27  0x0000009C   -   ivINT_Reserved39                  unused by PE */
    (tIsrFunc)&Cpu_Interrupt,          /* 0x28  0x000000A0   -   ivINT_USB0                        unused by PE */
    (tIsrFunc)&Cpu_Interrupt,          /* 0x29  0x000000A4   -   ivINT_DAC0                        unused by PE */
    (tIsrFunc)&Cpu_Interrupt,          /* 0x2A  0x000000A8   -   ivINT_TSI0                        unused by PE */
    (tIsrFunc)&Cpu_Interrupt,          /* 0x2B  0x000000AC   -   ivINT_MCG                         unused by PE */
    (tIsrFunc)&TU1_Interrupt,          /* 0x2C  0x000000B0   2   ivINT_LPTimer                     used by PE */
    (tIsrFunc)&Cpu_Interrupt,          /* 0x2D  0x000000B4   -   ivINT_Reserved45                  unused by PE */
    (tIsrFunc)&Cpu_Interrupt,          /* 0x2E  0x000000B8   -   ivINT_PORTA                       unused by PE */
    (tIsrFunc)&Cpu_Interrupt           /* 0x2F  0x000000BC   -   ivINT_PORTD                       unused by PE */
    }
  };
```

```
Address    0 - 3      4 - 7      8 - B      C - F
0000A090   0000A6CD   0000A6CD   0000A6CD   0000A6CD
0000A0A0   0000A6CD   0000A6CD   0000A6CD   0000A6CD
0000A0B0   0000A599   0000A6CD   0000A6CD   0000A6CD
0000A0C0   FFFFFFFF   FFFFFFFF   FFFFFFFF   FFFFFFFF
0000A0D0   FFFFFFFF   FFFFFFFF   FFFFFFFF   FFFFFFFF
0000A0E0   FFFFFFFF   FFFFFFFF   FFFFFFFF   FFFFFFFF
```

```
                0x0000a4d4                __aeabi_ldiv0
 *(.text*)
 .text.TU1_Init
                0x0000a4d8       0xc0 ./Generated_Code/TU1.o
                0x0000a4d8                TU1_Init
 .text.TU1_Interrupt
                0x0000a598       0x30 ./Generated_Code/TU1.o
                0x0000a598                TU1_Interrupt
 .text.TI1_Init
                0x0000a5c8       0x50 ./Generated_Code/TI1.o
                0x0000a5c8                TI1_Init
 .text.TU1_OnCounterRestart
                0x0000a618       0x24 ./Generated_Code/TI1.o
                0x0000a618                TU1_OnCounterRestart
 .text.Red_LED_Init
                0x0000a63c       0x68 ./Generated_Code/Red_LED.o
                0x0000a63c                Red_LED_Init
```

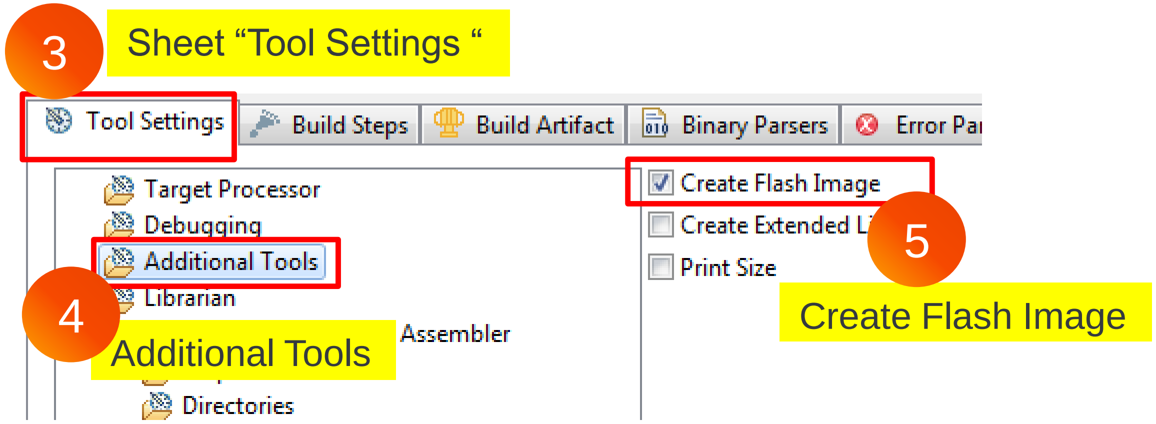
- Make sure the timer ISR was registered in relative interrupt vector address .

Generate S-record or binary image file



1 Click mouse right button

2 Properties/ C/C++ Build /Settings



☑ Create Flash Image

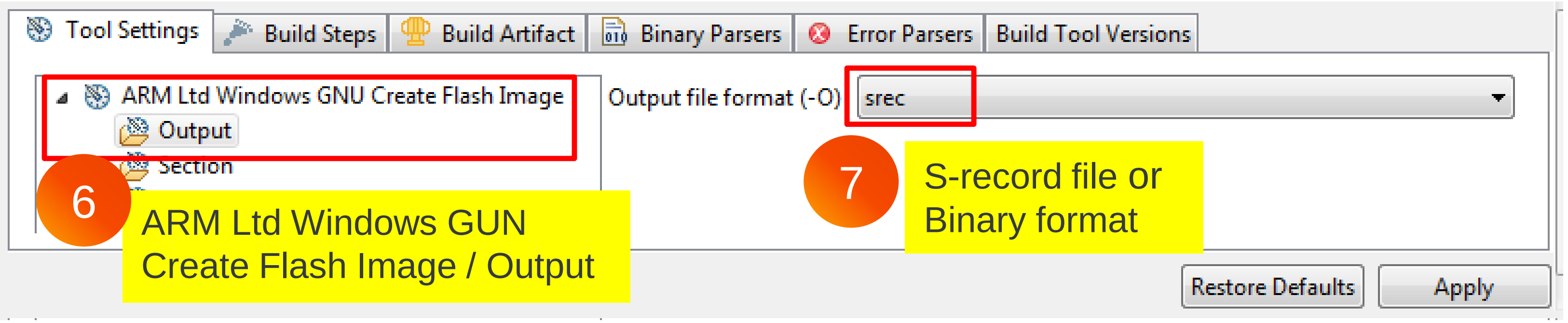
□ Create Extended Li

□ Print Size

3 Sheet “Tool Settings “

4 Additional Tools

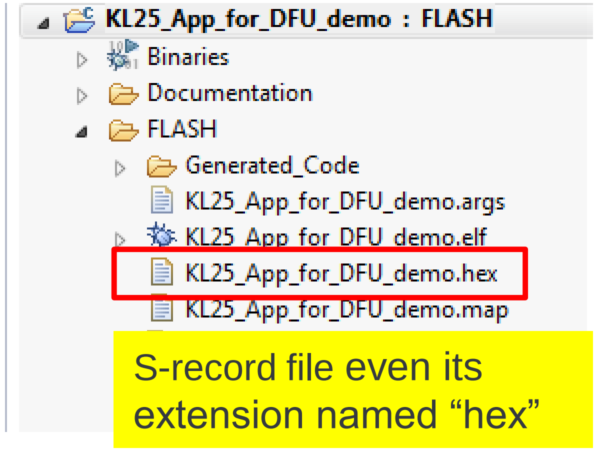
5 Create Flash Image



Output file format (-O) srec

6 ARM Ltd Windows GUN Create Flash Image / Output

7 S-record file or Binary format


KL25_App_for_DFU_demo : FLASH
Binaries
Documentation
FLASH
Generated_Code
KL25_App_for_DFU_demo.args
KL25_App_for_DFU_demo.elf
KL25_App_for_DFU_demo.hex
KL25_App_for_DFU_demo.map

S-record file even its extension named “hex” freescale™

www.Freescale.com