

## Lab2 of lesson3 in unit 3

Firstly create linker\_script.ld and startup.s, taking the makefile from the first lab

```
MINGW64:/d/Embedded Diploma/Units/Uint_3/Lesson_2/Labs

Osama@DESKTOP-BGUJ1JP MINGW64 /d/Embedded Diploma/Units/Uint_3/Lesson_2/Labs
$ export PATH=../ARM/bin/:$PATH

Osama@DESKTOP-BGUJ1JP MINGW64 /d/Embedded Diploma/Units/Uint_3/Lesson_2/Labs
$ touch linker_script.ld startup.s

Osama@DESKTOP-BGUJ1JP MINGW64 /d/Embedded Diploma/Units/Uint_3/Lesson_2/Labs
$ |
```

This screen for the makefile file.

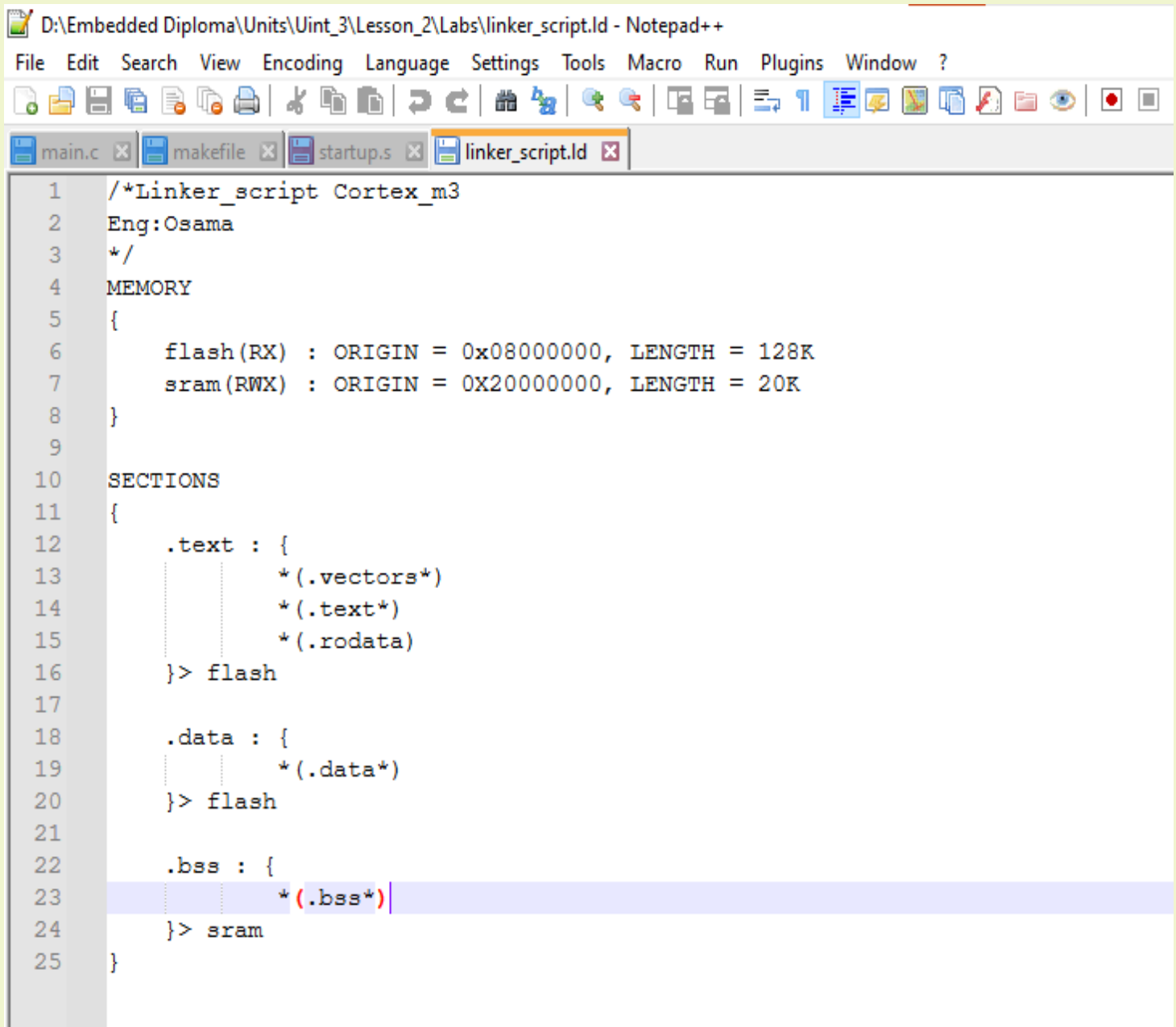
```
D:\Embedded Diploma\Units\Uint_3\Lesson_2\Labs\makefile - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

main.c makefile startup.s linker_script.ld
1  #@copyright : eng<Osama_Khallaf>
2  CC= arm-none-eabi-
3  CFLAGS= -mcpu=cortex-m3
4  INCS = -I .
5  LIBS=
6  SRC = $(wildcard *.c)
7  OBJ = $(SRC:.c=.o)
8  AS = $(wildcard *.s)
9  ASOBJ=$(AS:.s=.o)
10 Project_name=learn_in_depth_cortex_m3
11
12 all: $(Project_name).bin
13     @echo "====Build is Done====="
14
15 startup.o: startup.s
16     $(CC)as.exe $(CFLAGS) $< -o $@
17
18 %.o: %.c
19     $(CC)gcc.exe -c $(INCS) $(CFLAGS) $< -o $@
20
21 $(Project_name).elf: $(OBJ) $(ASOBJ)
22     $(CC)ld.exe -T linker_script.ld $(LIBS) $(OBJ) $(ASOBJ) -o $@
23
24 $(Project_name).bin: $(Project_name).elf
25     $(CC)objcopy.exe -O binary $< $@
26
27 clean_all:
28     rm *.o *.elf *.bin
29
30 clean:
31     rm *.elf *.bin
```

# The startup.s file

```
*D:\Embedded Diploma\Units\Uint_3\Lesson_2\Labs\startup.s - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
main.c makefile startup.s linker_script.ld
1  /*cortex_m3
2  Eng : Osama Khallaf
3  */
4
5  .section .vectors
6
7  .word 0x20001000          /*stack top address */
8  .word _reset             /*1 Reset */
9  .word Vector_handler     /*2 NMI */
10 .word Vector_handler     /*3 Hard fault */
11 .word Vector_handler     /*4 MM Fault */
12 .word Vector_handler     /*5 Bus Fault */
13 .word Vector_handler     /*6 Usage Fault */
14 .word Vector_handler     /*7 RESERVED */
15 .word Vector_handler     /*8 RESERVED */
16 .word Vector_handler     /*9 RESERVED */
17 .word Vector_handler     /*10 RESERVED */
18 .word Vector_handler     /*11 SV call */
19 .word Vector_handler     /*12 debug reserved */
20 .word Vector_handler     /*13 RESERVED */
21 .word Vector_handler     /*14 pendSV */
22 .word Vector_handler     /*15 SysTick */
23 .word Vector_handler     /*16 IRQ0 */
24 .word Vector_handler     /*17 IRQ1 */
25 .word Vector_handler     /*18 IRQ2 */
26 .word Vector_handler     /*19 ... */
27     /* on to IRQ67 */
28
29 .section .text
30 _reset:
31     bl main
32     b .
33
34 .thumb_func
35 Vector_handler:
36     b _reset
length : 1,132 lines : 37
```

## linker\_script.ld file



```
1  /*Linker_script Cortex_m3
2  Eng:Osama
3  */
4  MEMORY
5  {
6      flash(RX) : ORIGIN = 0x08000000, LENGTH = 128K
7      sram(RWX) : ORIGIN = 0x20000000, LENGTH = 20K
8  }
9
10 SECTIONS
11 {
12     .text : {
13         *(.vectors*)
14         *(.text*)
15         *(.rodata)
16     }> flash
17
18     .data : {
19         *(.data*)
20     }> flash
21
22     .bss : {
23         *(.bss*)
24     }> sram
25 }
```

Map\_file adding this command "`$(CC)ld.exe -T linker_script.ld $(LIBS)`

`$(OBJ) $(AsOBJ) -o $@ -Map=Map_file.map`"

```
D:\Embedded Diploma\Units\Uint_3\Lesson_2\Labs\Map_file.map - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
startup.s linker_script.ld makefile Map_file.map
1 Memory Configuration
2
3 Name Origin Length Attributes
4 flash 0x08000000 0x00020000 xr
5 sram 0x20000000 0x00005000 xrw
6 *default* 0x00000000 0xffffffff
7
8 Linker script and memory map
9
10 .text 0x08000000 0xd4
11 *(.vectors*) 0x08000000 0x50 startup.o
12 .vectors
13 *(.text*) 0x08000050 0x7c main.o
14 .text 0x08000050 main
15 0x08000050 startup.o
16 .text 0x080000cc 0x8 startup.o
17 *(.rodata)
18
19 .glue_7 0x080000d4 0x0
20
21 .glue_7t 0x080000d4 0x0
22
23 .vfp11_veneer 0x080000d4 0x0
24
25 .v4_bx 0x080000d4 0x0
26
27 .iplt 0x080000d4 0x0
28
29 .rel.dyn 0x080000d4 0x0
30
31 .data 0x080000d4 0xb
32
33 *(.data*)
34 .data 0x080000d4 0xb main.o
35 0x080000d4 R_ODR
36 0x080000d8 g_value
37 0x080000dc const_vaariables
38 .data 0x080000df 0x0 startup.o
39
40 .igot.plt 0x080000e0 0x0
41
42 .bss 0x20000000 0x0
43
44 *(.bss*)
45 .bss 0x20000000 0x0 main.o
46 .bss 0x20000000 0x0 startup.o
47
48 LOAD main.o
49
50
51
52
```

"`$ arm-none-eabi-objdump.exe -h learn_in_depth_cortex_m3.elf`"

```
Osama@DESKTOP-BGUJ1JP MINGW64 /d/Embedded Diploma/Units/Uint_3/Lesson_2/Labs
$ arm-none-eabi-objdump.exe -h learn_in_depth_cortex_m3.elf

learn_in_depth_cortex_m3.elf:      file format elf32-littlearm

Sections:
Idx Name          Size      VMA       LMA       File off  Algn
 0 .text          000000d4  08000000  08000000  00010000  2**2
   CONTENTS, ALLOC, LOAD, READONLY, CODE
 1 .data           0000000b  080000d4  080000d4  000100d4  2**2
   CONTENTS, ALLOC, LOAD, DATA
 2 .comment        0000007e  00000000  00000000  000100df  2**0
   CONTENTS, READONLY
 3 .ARM.attributes 00000031  00000000  00000000  0001015d  2**0
   CONTENTS, READONLY

Osama@DESKTOP-BGUJ1JP MINGW64 /d/Embedded Diploma/Units/Uint_3/Lesson_2/Labs
$ |
```

Now to burn the elf file on Proteus simulator we should make "gdwarf-2" to understand by the Proteus gdb

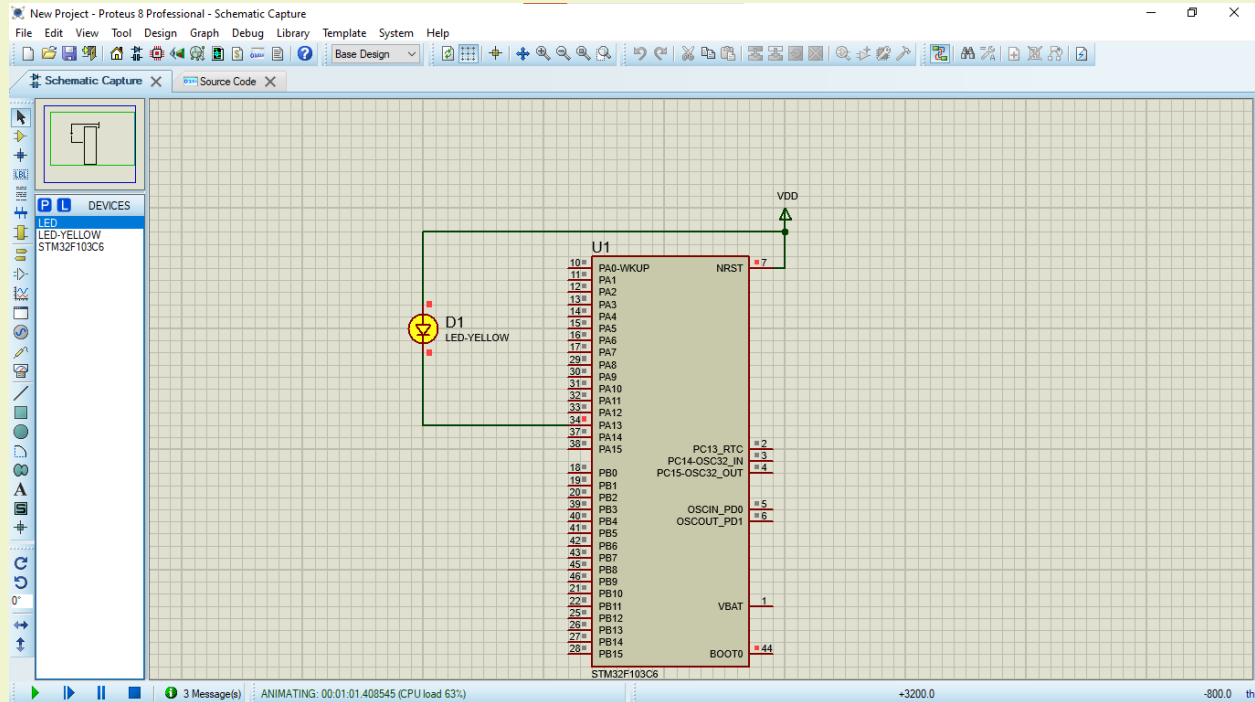
```
MINGW64:/d/Embedded Diploma/Units/Uint_3/Lesson_2/Labs
Osama@DESKTOP-BGUJ1JP MINGW64 /d/Embedded Diploma/Units/Uint_3/Lesson_2/Labs
$ make
arm-none-eabi-gcc.exe -c -I . -mcpu=cortex-m3 -gdwarf-2 main.c -o main.o
arm-none-eabi-as.exe -mcpu=cortex-m3 -gdwarf-2 startup.s -o startup.o
arm-none-eabi-ld.exe -T linker_script.ld main.o startup.o -o learn_in_depth_cortex_m3.elf -Map=Map_file.map
arm-none-eabi-objcopy.exe -O binary learn_in_depth_cortex_m3.elf learn_in_depth_cortex_m3.bin
=====Build is Done=====

Osama@DESKTOP-BGUJ1JP MINGW64 /d/Embedded Diploma/Units/Uint_3/Lesson_2/Labs
$ arm-none-eabi-objdump.exe -h learn_in_depth_cortex_m3.elf

learn_in_depth_cortex_m3.elf:      file format elf32-littlearm

Sections:
Idx Name          Size      VMA           LMA           File off  Algn
  0 .text          000000d4  08000000  08000000  00010000  2**2
    CONTENTS, ALLOC, LOAD, READONLY, CODE
  1 .data          0000000b  080000d4  080000d4  000100d4  2**2
    CONTENTS, ALLOC, LOAD, DATA
  2 .debug_info     000001a9  00000000  00000000  000100df  2**0
    CONTENTS, READONLY, DEBUGGING
  3 .debug_abbrev    00000104  00000000  00000000  00010288  2**0
    CONTENTS, READONLY, DEBUGGING
  4 .debug_loc      00000038  00000000  00000000  0001038c  2**0
    CONTENTS, READONLY, DEBUGGING
  5 .debug_aranges  00000040  00000000  00000000  000103c8  2**3
    CONTENTS, READONLY, DEBUGGING
  6 .debug_line     00000146  00000000  00000000  00010408  2**0
    CONTENTS, READONLY, DEBUGGING
  7 .debug_str      00000164  00000000  00000000  0001054e  2**0
    CONTENTS, READONLY, DEBUGGING
  8 .comment        0000007e  00000000  00000000  000106b2  2**0
    CONTENTS, READONLY
  9 .ARM.attributes 00000031  00000000  00000000  00010730  2**0
    CONTENTS, READONLY
10 .debug_frame     0000002c  00000000  00000000  00010764  2**2
    CONTENTS, READONLY, DEBUGGING
```

Now we burn the .elf file on Proteus simulator and toggle the led.



When you click in pause it display the source code for main.c file

```
#include<stdint.h>
#define Base_Rcc 0x40021000
#define Base_GPIOA 0x40010800
#define RCC_APB2ENR (*(volatile uint32_t*) (Base_Rcc + 0x18))
#define GPIO_CRH (*(volatile uint32_t*) (Base_GPIOA + 0x04))
#define GPIO_ODR (*(volatile uint32_t*) (Base_GPIOA + 0x0c))

typedef union
{
    uint32_t all_fields;
    struct
    {
        uint32_t reserved;
        uint32_t p_13;
    };
}pin;

volatile R_ODR_t * R_ODR =(volatile R_ODR_t*)(Base_GPIOA + 0x0c);

unsigned char g_value[3] = {1, 2, 3};
unsigned char const_vaariables[3] = {1, 2, 3};

int main(void)
{
    RCC_APB2ENR |= 1<<2;
    GPIO_CRH |= 0xFF0FFFFF;
    GPIO_CRH |= 0x00200000;
    while(1)
    {
        R_ODR->p_13 = 1;
        for(int i = 0; i < 5000; i++);
        R_ODR->p_13 = 0;
        for(int i = 0; i < 5000; i++);
    }
    return 0;
}
```

Name	Address	Value
g_value	08000008	byte[3]
const_vaariables	0800000C	byte[3]
R_ODR	08000004	0x4001080C
i	BP+12 = @20000FD4	2247
i	BP+16 = @20000FD0	5000

We should take a backup from the startup.s file to avoid any crash happen between startup.c and startup.s by using this command "`$ mv startup.s startup.s-orig`" we take a backup from the startup.s file

```
MINGW64:/d/Embedded Diploma/Units/Uint_3/Lesson_2/Labs

Osama@DESKTOP-BGUJ1JP MINGW64 /d/Embedded Diploma/Units/Uint_3/Lesson_2/Labs
$ mv startup.s startup.s-orig

Osama@DESKTOP-BGUJ1JP MINGW64 /d/Embedded Diploma/Units/Uint_3/Lesson_2/Labs
$ ls
Map_file.map          lab2.docx             linker_script.ld      startup.o
'New Project.pdsprj'  learn_in_depth_cortex_m3.bin  main.c               startup.s-orig
lab1/                 learn_in_depth_cortex_m3.elf  main.o               '~$lab2.docx'
lab2/                 learn_in_depth_cortex_m3.elf.asm  makefile             '~$WRL0005.tmp'

Osama@DESKTOP-BGUJ1JP MINGW64 /d/Embedded Diploma/Units/Uint_3/Lesson_2/Labs
$ |
```

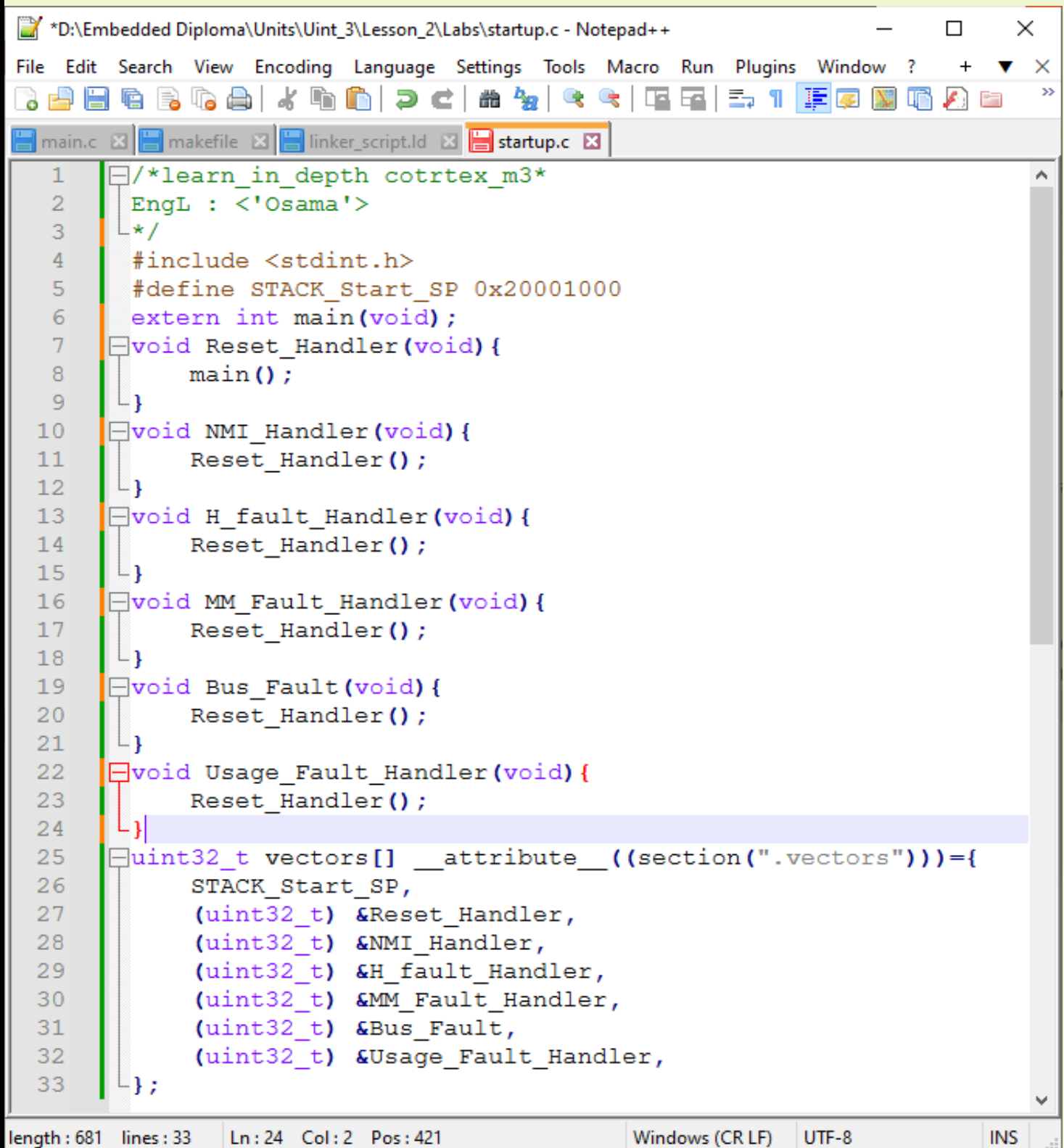
Now we create a startup.c file by using this command "`$`

`touch startup.c`"

```
Osama@DESKTOP-BGUJ1JP MINGW64 /d/Embedded Diploma/Units/Uint_3/Lesson_2/Labs
$ touch startup.c

Osama@DESKTOP-BGUJ1JP MINGW64 /d/Embedded Diploma/Units/Uint_3/Lesson_2/Labs
$ |
```

We can write the startup file with c code "startup.c"



```
1  /*learn_in_depth cotrtex_m3*
2  EngL : <'Osama'>
3  */
4  #include <stdint.h>
5  #define STACK_Start_SP 0x20001000
6  extern int main(void);
7  void Reset_Handler(void) {
8      main();
9  }
10 void NMI_Handler(void) {
11     Reset_Handler();
12 }
13 void H_fault_Handler(void) {
14     Reset_Handler();
15 }
16 void MM_Fault_Handler(void) {
17     Reset_Handler();
18 }
19 void Bus_Fault(void) {
20     Reset_Handler();
21 }
22 void Usage_Fault_Handler(void) {
23     Reset_Handler();
24 }
25 uint32_t vectors[] __attribute__((section(".vectors")))={
26     STACK_Start_SP,
27     (uint32_t) &Reset_Handler,
28     (uint32_t) &NMI_Handler,
29     (uint32_t) &H_fault_Handler,
30     (uint32_t) &MM_Fault_Handler,
31     (uint32_t) &Bus_Fault,
32     (uint32_t) &Usage_Fault_Handler,
33 };
```

length : 681 lines : 33 Ln : 24 Col : 2 Pos : 421 Windows (CR LF) UTF-8 INS

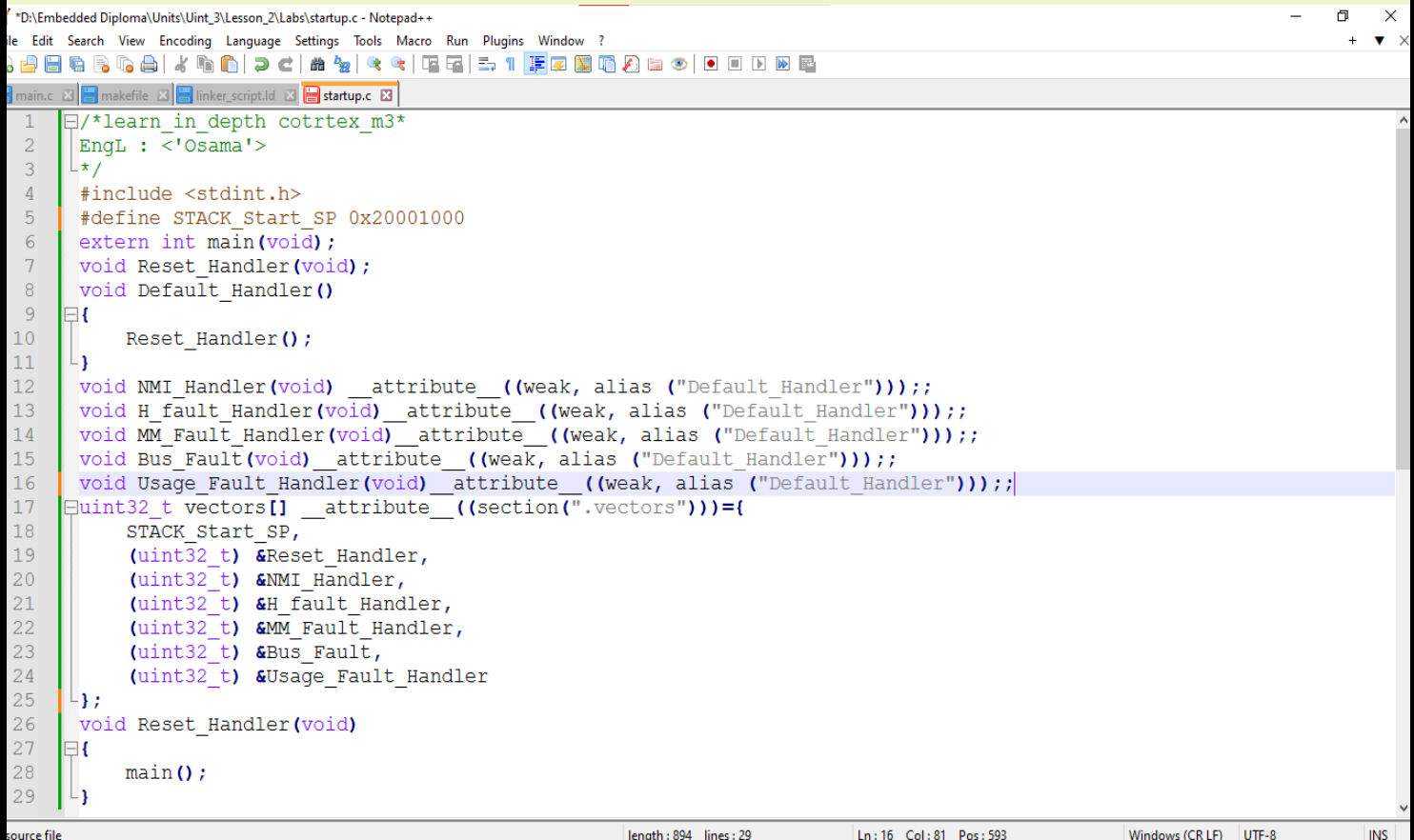


We can now show the address of each sections by using this command "\$ arm-none-eabi-nm.exe learn\_in\_depth\_cortex\_m3.elf"

```
Osama@DESKTOP-BGUJ1JP MINGW64 /d/Embedded Diploma/Units/Uint_3/Lesson_2/Labs
$ arm-none-eabi-nm.exe learn_in_depth_cortex_m3.elf
080000c8 T Bus_Fault
080000e8 D const_vaariables
080000e4 D g_value
080000b0 T H_fault_Handler
0800001c T main
080000bc T MM_Fault_Handler
080000a4 T NMI_Handler
080000e0 D R_ODR
08000098 T Reset_Handler
080000d4 T Usage_Fault_Handler
08000000 T vectors

Osama@DESKTOP-BGUJ1JP MINGW64 /d/Embedded Diploma/Units/Uint_3/Lesson_2/Labs
$ |
```

## Using weak and alias



```
*D:\Embedded Diploma\Units\Uint_3\Lesson_2\Labs\startup.c - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
main.c makefile linker_script.ld startup.c
1 /*learn_in_depth cotrtex_m3*
2 EngL : <'Osama'>
3 */
4 #include <stdint.h>
5 #define STACK_Start_SP 0x20001000
6 extern int main(void);
7 void Reset_Handler(void);
8 void Default_Handler()
9 {
10     Reset_Handler();
11 }
12 void NMI_Handler(void) __attribute__((weak, alias ("Default_Handler")));
13 void H_fault_Handler(void) __attribute__((weak, alias ("Default_Handler")));
14 void MM_Fault_Handler(void) __attribute__((weak, alias ("Default_Handler")));
15 void Bus_Fault(void) __attribute__((weak, alias ("Default_Handler")));
16 void Usage_Fault_Handler(void) __attribute__((weak, alias ("Default_Handler")));
17 uint32_t vectors[] __attribute__((section(".vectors")))={
18     STACK_Start_SP,
19     (uint32_t) &Reset_Handler,
20     (uint32_t) &NMI_Handler,
21     (uint32_t) &H_fault_Handler,
22     (uint32_t) &MM_Fault_Handler,
23     (uint32_t) &Bus_Fault,
24     (uint32_t) &Usage_Fault_Handler
25 };
26 void Reset_Handler(void)
27 {
28     main();
29 }
```

source file length: 894 lines: 29 Ln: 16 Col: 81 Pos: 593 Windows (CR LF) UTF-8 INS

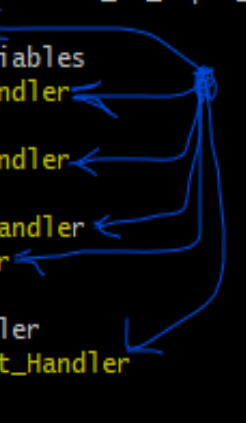
The address of each sections after adding alias and weak to our startup.c file by using this command "`$ arm-none-eabi-nm.exe learn_in_depth_cortex_m3.elf`"

```
MINGW64:/d/Embedded Diploma/Units/Uint_3/Lesson_2/Labs
Osama@DESKTOP-BGUJ1JP MINGW64 /d/Embedded Diploma/Units/Uint_3/Lesson_2/Labs
$ make clean_all
rm *.o *.elf *.bin

Osama@DESKTOP-BGUJ1JP MINGW64 /d/Embedded Diploma/Units/Uint_3/Lesson_2/Labs
$ make
arm-none-eabi-gcc.exe -c -I . -mcpu=cortex-m3 -gdwarf-2 main.c -o main.o
arm-none-eabi-gcc.exe -c -I . -mcpu=cortex-m3 -gdwarf-2 startup.c -o startup.o
arm-none-eabi-ld.exe -T linker_script.ld main.o startup.o -o learn_in_depth_cortex_m3.elf -Map=Map_file.map
arm-none-eabi-objcopy.exe -O binary learn_in_depth_cortex_m3.elf learn_in_depth_cortex_m3.bin
=====Build is Done=====

Osama@DESKTOP-BGUJ1JP MINGW64 /d/Embedded Diploma/Units/Uint_3/Lesson_2/Labs
$ arm-none-eabi-nm.exe learn_in_depth_cortex_m3.elf
08000098 W Bus_Fault
080000b8 D const_vaariables
08000098 T Default_Handler
080000b4 D g_value
08000098 W H_fault_Handler
0800001c T main
08000098 W MM_Fault_Handler
08000098 W NMI_Handler
080000b0 D R_ODR
080000a4 T Reset_Handler
08000098 W Usage_Fault_Handler
08000000 T vectors

Osama@DESKTOP-BGUJ1JP MINGW64 /d/Embedded Diploma/Units/Uint_3/Lesson_2/Labs
$ |
```



A diagram consisting of a central point with several arrows pointing to the addresses of specific sections in the nm output. The arrows point to the following lines:

- 08000098 W Bus\_Fault
- 080000b8 D const\_vaariables
- 08000098 T Default\_Handler
- 080000b4 D g\_value
- 08000098 W H\_fault\_Handler
- 0800001c T main
- 08000098 W MM\_Fault\_Handler
- 08000098 W NMI\_Handler
- 080000b0 D R\_ODR
- 080000a4 T Reset\_Handler
- 08000098 W Usage\_Fault\_Handler

## Final step: How to copy (data and create .bss sections)

```
D:\Embedded Diploma\Units\Uint_3\Lesson_2\Labs\linker_script.ld - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
main.c makefile linker_script.ld startup.c Map_file.map

8 |)
9 |
10 |SECTIONS
11 |{
12 |    .text : {
13 |        *(.vectors*)
14 |        *(.text*)
15 |        *(.rodata)
16 |        S_TEXT = .;
17 |    }> flash
18 |
19 |    .data : {
20 |        S_DATA = .;
21 |        *(.data*)
22 |        S_DATA = .;
23 |    }> flash
24 |
25 |    .bss : {
26 |        S_BSS = .;
27 |        *(.bss*)
28 |        S_BSS = .;
29 |    }> sram
30 |}
```

Normal text file      length: 405   lines: 30      Ln: 8   Col: 2   Pos: 152      Windows (CR LF)   UTF-8   INS

```
D:\Embedded Diploma\Units\Uint_3\Lesson_2\Labs\Map_file.map - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
main.c makefile linker_script.ld startup.c Map_file.map

36 |.vfp11_veneer 0x080000b0 0x0
37 |.vfp11_veneer 0x080000b0 0x0 linker stubs
38 |
39 |.v4_bx 0x080000b0 0x0
40 |.v4_bx 0x080000b0 0x0 linker stubs
41 |
42 |.iplt 0x080000b0 0x0
43 |.iplt 0x080000b0 0x0 main.o
44 |
45 |.rel.dyn 0x080000b0 0x0
46 |.rel.iplt 0x080000b0 0x0 main.o
47 |
48 |.data 0x080000b0 0xb
49 |    S_DATA = .;
50 |    *(.data*)
51 |    .data 0x080000b0 0xb main.o
52 |    0x080000b0 R_ODR
53 |    0x080000b4 g_value
54 |    0x080000b8 const_vaariables
55 |    .data 0x080000bb 0x0 startup.o
56 |    0x080000bb S_DATA = .;
57 |
58 |.igot.plt 0x080000bc 0x0
59 |.igot.plt 0x080000bc 0x0 main.o
60 |
61 |.bss 0x20000000 0x0
62 |    S_BSS = .;
63 |    *(.bss*)
64 |    .bss 0x20000000 0x0 main.o
65 |    .bss 0x20000000 0x0 startup.o
66 |    0x20000000 S_BSS = .;
67 |LOAD main.o
68 |LOAD startup.o
69 |OUTPUT(learn_in_depth_cortex_m3.elf elf32-littlearm)
70 |
71 |.debug_info 0x00000000 0x252
72 |.debug_info 0x00000000 0x183 main.o
73 |.debug_info 0x00000183 0xcf startup.o
```

Normal text file      length: 3,965   lines: 114      Ln: 1   Col: 1   Pos: 1      Windows (CR LF)   UTF-8   INS

```
MINGW64:/d/Embedded Diploma/Units/Uint_3/Lesson_2/Labs
$ make
arm-none-eabi-gcc.exe -c -I . -mcpu=cortex-m3 -gdwarf-2 main.c -o main.o
arm-none-eabi-gcc.exe -c -I . -mcpu=cortex-m3 -gdwarf-2 startup.c -o startup.o
arm-none-eabi-ld.exe -T linker_script.ld main.o startup.o -o learn_in_depth_cortex_m3.elf -Map=Map_file.map
arm-none-eabi-objcopy.exe -O binary learn_in_depth_cortex_m3.elf learn_in_depth_cortex_m3.bin
=====Build is Done=====

Osama@DESKTOP-BGUJ1JP MINGW64 /d/Embedded Diploma/Units/Uint_3/Lesson_2/Labs
$ arm-none-eabi-nm.exe learn_in_depth_cortex_m3.elf
20000000 D _E_bss
080000bb D _E_DATA
080000b0 T _E_text
20000000 D _S_bss
080000b0 D _S_DATA
08000098 W Bus_Fault
080000b8 D const_vaariables
08000098 T Default_Handler
080000b4 D g_value
08000098 W H_fault_Handler
0800001c T main
08000098 W MM_Fault_Handler
08000098 W NMI_Handler
080000b0 D R_ODR
080000a4 T Reset_Handler
08000098 W Usage_Fault_Handler
08000000 T vectors

Osama@DESKTOP-BGUJ1JP MINGW64 /d/Embedded Diploma/Units/Uint_3/Lesson_2/Labs
$ |
```

## startup.c after adding start & ends of sections

```
*D:\Embedded Diploma\Units\Uint_3\Lesson_2\Labs\startup.c - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
main.c x makefile x startup.c x
1  /*learn_in_depth cotrtex_m3*
2  EngL : <'Osama'>
3  */
4  #include <stdint.h>
5  #define STACK_Start_SP 0x20001000
6  extern int main(void);
7  void Reset_Handler(void);
8  void Default_Handler()
9  {
10     Reset_Handler();
11 }
12 void NMI_Handler(void) __attribute__((weak, alias ("Default_Handler")));
13 void H_fault_Handler(void) __attribute__((weak, alias ("Default_Handler")));
14 void MM_Fault_Handler(void) __attribute__((weak, alias ("Default_Handler")));
15 void Bus_Fault(void) __attribute__((weak, alias ("Default_Handler")));
16 void Usage_Fault_Handler(void) __attribute__((weak, alias ("Default_Handler")));
17 uint32_t vectors[] __attribute__((section(".vectors")))={
18     STACK_Start_SP,
19     (uint32_t) &Reset_Handler,
20     (uint32_t) &NMI_Handler,
21     (uint32_t) &H_fault_Handler,
22     (uint32_t) &MM_Fault_Handler,
23     (uint32_t) &Bus_Fault,
24     (uint32_t) &Usage_Fault_Handler
25 };
26 extern unsigned int _S_DATA;
27 extern unsigned int _E_DATA;
28 extern unsigned int _S_bss;
29 extern unsigned int _E_bss;
30 extern unsigned int _E_text;
31
32 void Reset_Handler(void)
33 {
34     //copy data from ROM to Ram
35     //copy bss from 0 to 0
```

Cs length: 1,610 lines: 51 Ln: 18 Col: 19 Sel: 14 | 1 Windows (CR LF) UTF-8 INS

# Linker script

```
D:\Embedded Diploma\Units\Unit_3\Lesson_2\Labs\linker_script.ld - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

linker_script.ld x makefile x Map_file.map x

1  /*Linker_script Cortex_m3
2  Eng:Osama
3  */
4  MEMORY
5  {
6      flash(RX) : ORIGIN = 0x00000000, LENGTH = 128K
7      sram(RWX) : ORIGIN = 0x20000000, LENGTH = 20K
8  }
9
10 SECTIONS
11 {
12     .text : {
13         *(.vectors*)
14         *(.text*)
15         *(.rodata)
16         _E_text = . ;
17     }> flash
18
19     .data : {
20         _S_DATA = . ;
21         *(.data*)
22         _E_DATA = . ;
23     }> sram AT> flash
24
25     .bss : {
26         _S_bss = . ;
27         *(.bss*)
28         . = ALIGN(4);
29         _E_bss = . ;
30
31         . = ALIGN(4);
32         . = . + 0x1000;
33         _stack_top = . ;
34     }> sram
35 }
```

Line 1, Column 1

Tab Size: 4 Plain Text

# Map file

D:\Embedded Diploma\Units\Uint\_3\Lesson\_2\Labs\Map\_file.map - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

linker\_script.ld x makefile x Map\_file.map

```
15      0x08000000      vectors
16      *(.text*)
17      .text          0x0800001c      0x7c main.o
18      .text          0x0800001c      main
19      .text          0x08000098      0x18 startup.o
20      .text          0x08000098      MM_Fault_Handler
21      .text          0x08000098      Usage_Fault_Handler
22      .text          0x08000098      Bus_Fault
23      .text          0x08000098      Default_Handler
24      .text          0x08000098      H_fault_Handler
25      .text          0x08000098      NMI_Handler
26      .text          0x08000098      Reset_Handler
27      *(.rodata)
28      .rodata        0x080000b0      _E_text = .
29
30      .glue_7         0x080000b0      0x0 ***
31
32      .glue_7t        0x080000b0      0x0 ***
33
34      .vfp11_veneer   0x080000b0      0x0 ***
35
36      .v4_bx          0x080000b0      0x0 ***
37
38      .iplt           0x080000b0      0x0 ***
39
40      .rel.dyn        0x080000b0      0x0 ***
41
42      .data           0x20000000      0xb load address 0x080000b0
43      .data           0x20000000      _S_DATA = .
44
45      *(.data*)
46      .data           0x20000000      0xb main.o
47      .data           0x20000000      R_ODR
48      .data           0x20000004      g_value
49      .data           0x20000008      const_vaariabls
50      .data           0x2000000b      0x0 startup.o
51      .data           0x2000000b      _E_DATA = .
52
53      .igot.plt        0x2000000c      0x0 load address 0x080000bb
54      .igot.plt        0x2000000c      0x0 main.o
55
56      .bss            0x2000000b      0x1001 load address 0x080000bb
57      .bss            0x2000000b      _S_bss = .
58
59      *(.bss*)
60      .bss            0x2000000b      0x0 main.o
61      .bss            0x2000000b      0x0 startup.o
```

Line 33, Column 5

D:\Embedded Diploma\Units\Uint\_3\Lesson\_2\Labs\Map\_file.map - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

linker\_script.ld x makefile x Map\_file.map x

```
63      *(.bss*)
64      .bss            0x2000000b      0x0 main.o
65      .bss            0x2000000b      0x0 startup.o
66      .bss            0x2000000c      . = ALIGN (0x4)
67      *fill*          0x2000000b      0x1
68      .fill            0x2000000c      _E_bss = .
69      .fill            0x2000000c      . = ALIGN (0x4)
70      .fill            0x2000000c      . = (. + 0x1000)
71      *fill*          0x2000000c      0x1000
72      .fill            0x2000000c      _stack_top = .
73      LOAD main.o
74      LOAD startup.o
75      OUTPUT(learn_in_depth_cortex_m3.elf elf32-littlearm)
76
77      .debug_info      0x00000000      0x252
78      .debug_info      0x00000000      0x183 main.o
79      .debug_info      0x000000183      0xcf startup.o
80
81      .debug_abbrev     0x00000000      0x178
82      .debug_abbrev     0x00000000      0xf0 main.o
83      .debug_abbrev     0x000000f0      0x88 startup.o
84
85      .debug_loc        0x00000000      0x90
86      .debug_loc        0x00000000      0x38 main.o
87      .debug_loc        0x00000038      0x58 startup.o
88
89      .debug_aranges    0x00000000      0x40
90      .debug_aranges    0x00000000      0x20 main.o
91      .debug_aranges    0x00000000      0x20 startup.o
92
93      .debug_line        0x00000000      0x200
94      .debug_line        0x00000000      0x10b main.o
95      .debug_line        0x00000010b      0xf5 startup.o
96
97      .debug_str         0x00000000      0x17b
98      .debug_str         0x00000000      0x14b main.o
99      .debug_str         0x00000000      0x188 (size before relaxing)
100      .debug_str         0x00000014b      0x30 startup.o
101      .debug_str         0x00000014b      0x162 (size before relaxing)
102
103      .comment           0x00000000      0x7e
104      .comment           0x00000000      0x7e main.o
105      .comment           0x00000000      0x7f (size before relaxing)
```

Line 90, Column 16

```

MINGW64:/d/Embedded Diploma/Units/Uint_3/Lesson_2/Labs
arm-none-eabi-gcc.exe -c -I . -mcpu=cortex-m3 -gdwarf-2 main.c -o main.o
arm-none-eabi-gcc.exe -c -I . -mcpu=cortex-m3 -gdwarf-2 startup.c -o startup.o
arm-none-eabi-ld.exe -T linker_script.ld main.o startup.o -o learn_in_depth_cortex_m3.elf -Map=Map_file.map
arm-none-eabi-objcopy.exe -O binary learn_in_depth_cortex_m3.elf learn_in_depth_cortex_m3.bin
=====Build is Done=====

Osama@DESKTOP-BGUJ1JP MINGW64 /d/Embedded Diploma/Units/Uint_3/Lesson_2/Labs
$ arm-none-eabi-nm.exe learn_in_depth_cortex_m3.elf
2000000c B _E_bss
2000000b D _E_DATA
080000b0 T _E_text
2000000b B _S_bss
20000000 D _S_DATA
2000100c B _stack_top
08000098 W Bus_Fault
20000008 D const_vaariables
08000098 T Default_Handler
20000004 D g_value
08000098 W H_fault_Handler
0800001c T main
08000098 W MM_Fault_Handler
08000098 W NMI_Handler
20000000 D R_ODR
080000a4 T Reset_Handler
08000098 W Usage_Fault_Handler
08000000 T vectors

Osama@DESKTOP-BGUJ1JP MINGW64 /d/Embedded Diploma/Units/Uint_3/Lesson_2/Labs
$ |

```

```

MINGW64:/d/Embedded Diploma/Units/Uint_3/Lesson_2/Labs

Osama@DESKTOP-BGUJ1JP MINGW64 /d/Embedded Diploma/Units/Uint_3/Lesson_2/Labs
$ arm-none-eabi-objdump.exe -h learn_in_depth_cortex_m3.elf

learn_in_depth_cortex_m3.elf:      file format elf32-littlearm

Sections:
Idx Name          Size      VMA       LMA       File off  Algn
  0 .text          000000b0 08000000 08000000 00010000  2**2
    CONTENTS, ALLOC, LOAD, READONLY, CODE
  1 .data          0000000b 20000000 080000b0 00020000  2**2
    CONTENTS, ALLOC, LOAD, DATA
  2 .bss           00001001 2000000b 080000bb 0002000b  2**0
    ALLOC
  3 .debug_info     00000252 00000000 00000000 0002000b  2**0
    CONTENTS, READONLY, DEBUGGING
  4 .debug_abbrev   00000178 00000000 00000000 0002025d  2**0
    CONTENTS, READONLY, DEBUGGING
  5 .debug_loc      00000090 00000000 00000000 000203d5  2**0
    CONTENTS, READONLY, DEBUGGING
  6 .debug_aranges  00000040 00000000 00000000 00020465  2**0
    CONTENTS, READONLY, DEBUGGING
  7 .debug_line     00000200 00000000 00000000 000204a5  2**0
    CONTENTS, READONLY, DEBUGGING
  8 .debug_str      0000017b 00000000 00000000 000206a5  2**0
    CONTENTS, READONLY, DEBUGGING
  9 .comment        0000007e 00000000 00000000 00020820  2**0
    CONTENTS, READONLY
10 .ARM.attributes 00000033 00000000 00000000 0002089e  2**0
    CONTENTS, READONLY
11 .debug_frame     00000074 00000000 00000000 000208d4  2**2
    CONTENTS, READONLY, DEBUGGING

Osama@DESKTOP-BGUJ1JP MINGW64 /d/Embedded Diploma/Units/Uint_3/Lesson_2/Labs
$ |

```