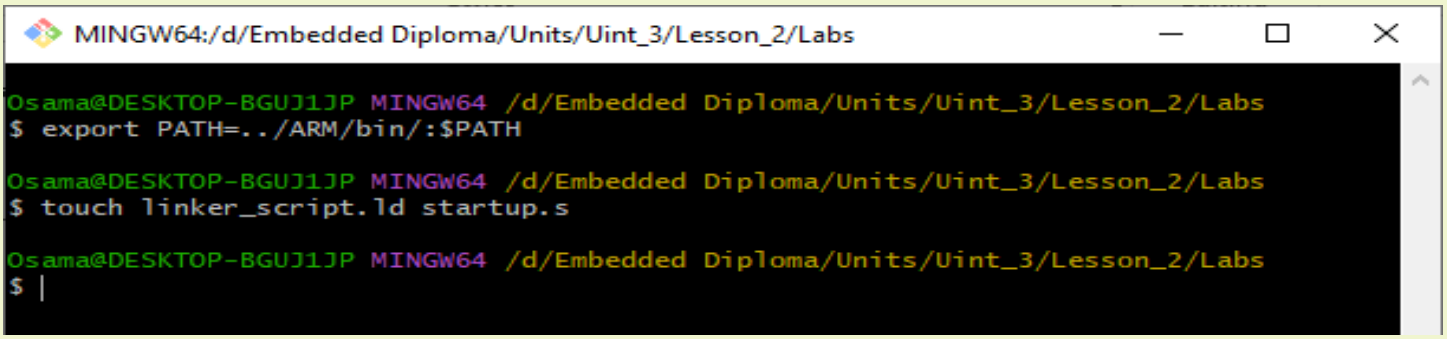
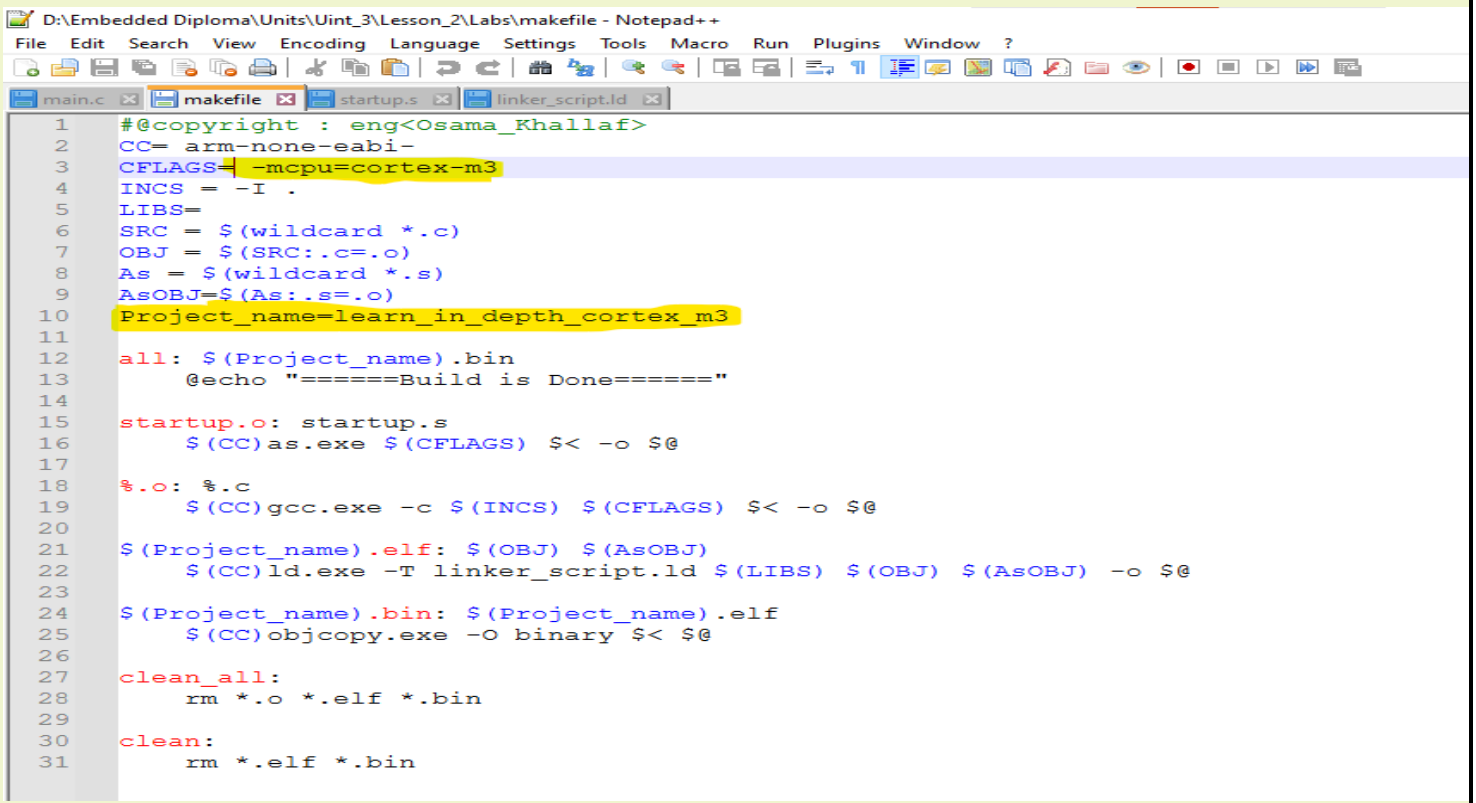
Lab2 of lesson3 in unit 3

Firstly create linker_script.ld and startup.s, taking the makefile form the first lab



This screen for the makefile file.



The startup.s file

*D:\Embedded Diploma\Units\Uint_3\Lesson_2\Labs\startup.s - Notepad++

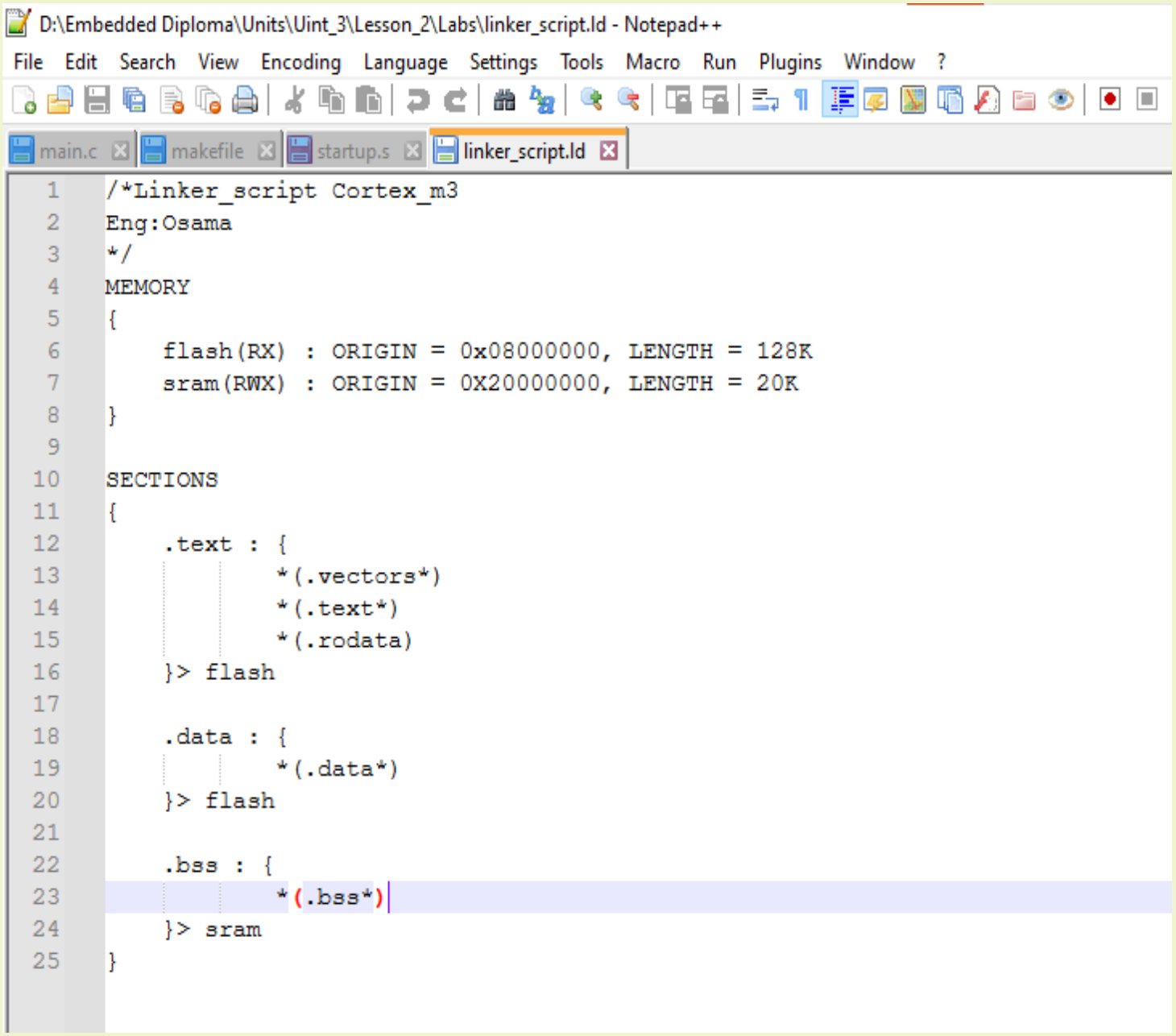
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

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
```

R programming language | length : 1,132 lines : 37

linker_script.ld file



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

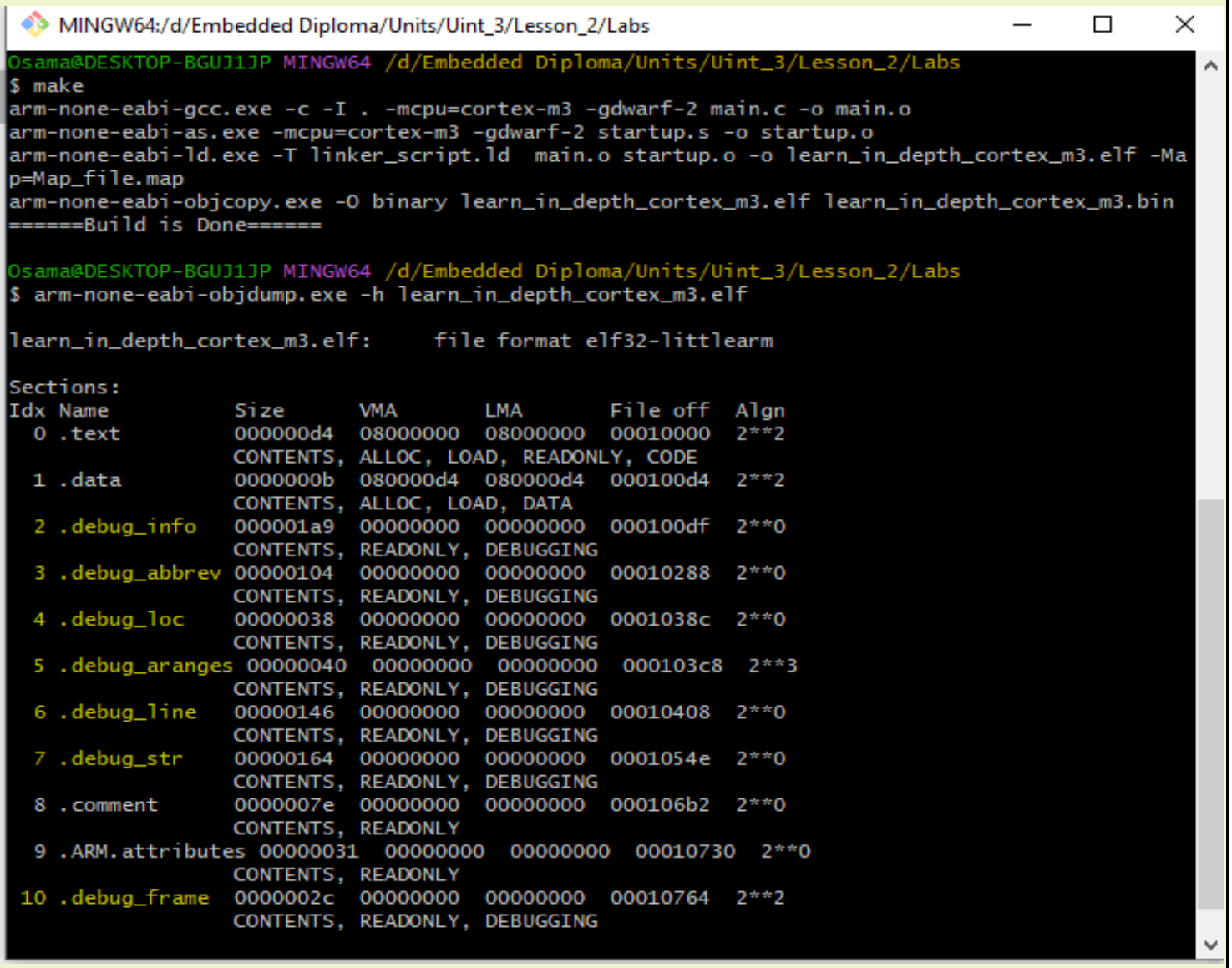
```
Memory Configuration

Name             Origin             Length             Attributes
flash            0x08000000         0x00020000         xr
sram             0x20000000         0x00005000         xrw
*default*        0x00000000         0xffffffff

Linker script and memory map

.text           0x08000000        0xd4
 *(.vectors*)
 .vectors       0x08000000        0x50 startup.o
 *(.text*)
 .text          0x08000050        0x7c main.o
                0x08000050                main
 .text          0x080000cc         0x8 startup.o
 *(.rodata)

.glue_7         0x080000d4         0x0 ...

.glue_7t        0x080000d4         0x0 ...

.vfp11_veneer   0x080000d4         0x0 ...

.v4_bx          0x080000d4         0x0 ...

.iplt           0x080000d4         0x0 ...

.rel.dyn        0x080000d4         0x0 ...

.data           0x080000d4         0xb
 *(.data*)
 .data          0x080000d4         0xb main.o
                0x080000d4                R_ODR
                0x080000d8                g_value
                0x080000dc                const_vaariables
 .data          0x080000df         0x0 startup.o

.igot.plt       0x080000e0         0x0 ...

.bss            0x20000000         0x0
 *(.bss*)
 .bss           0x20000000         0x0 main.o
 .bss           0x20000000         0x0 startup.o
```

Line 1, Column 1

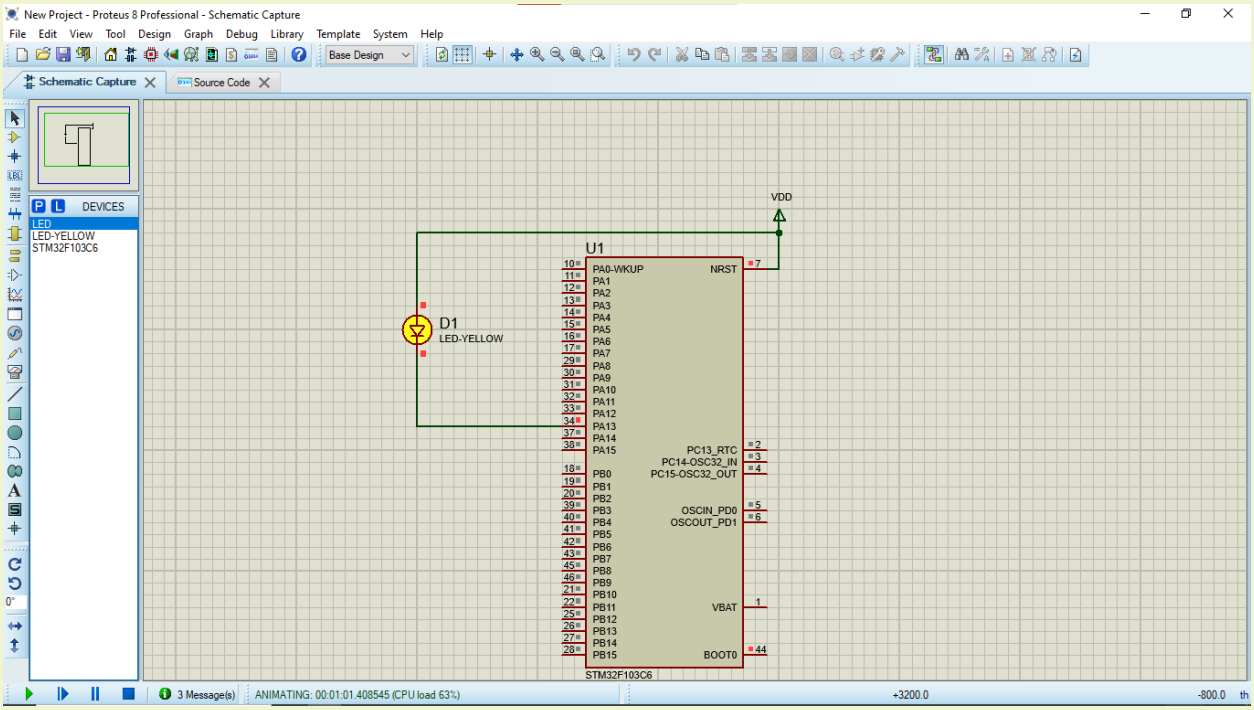
" $ arm-none-eabi-objdump.exe -h learn_in_depth_cortex_m3.elf"

```
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
$
```

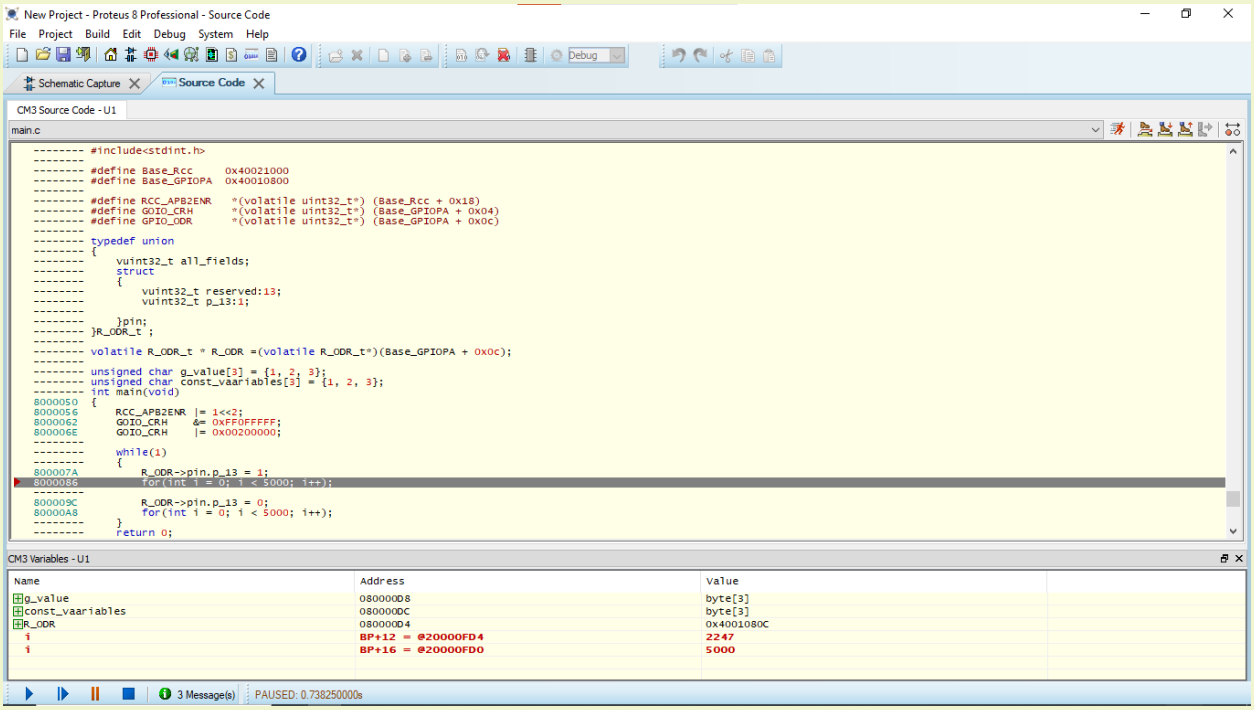
Now to burn the elf file on Proteus simulator we should make "gdwarf-2" to understand by the Proteus gdb



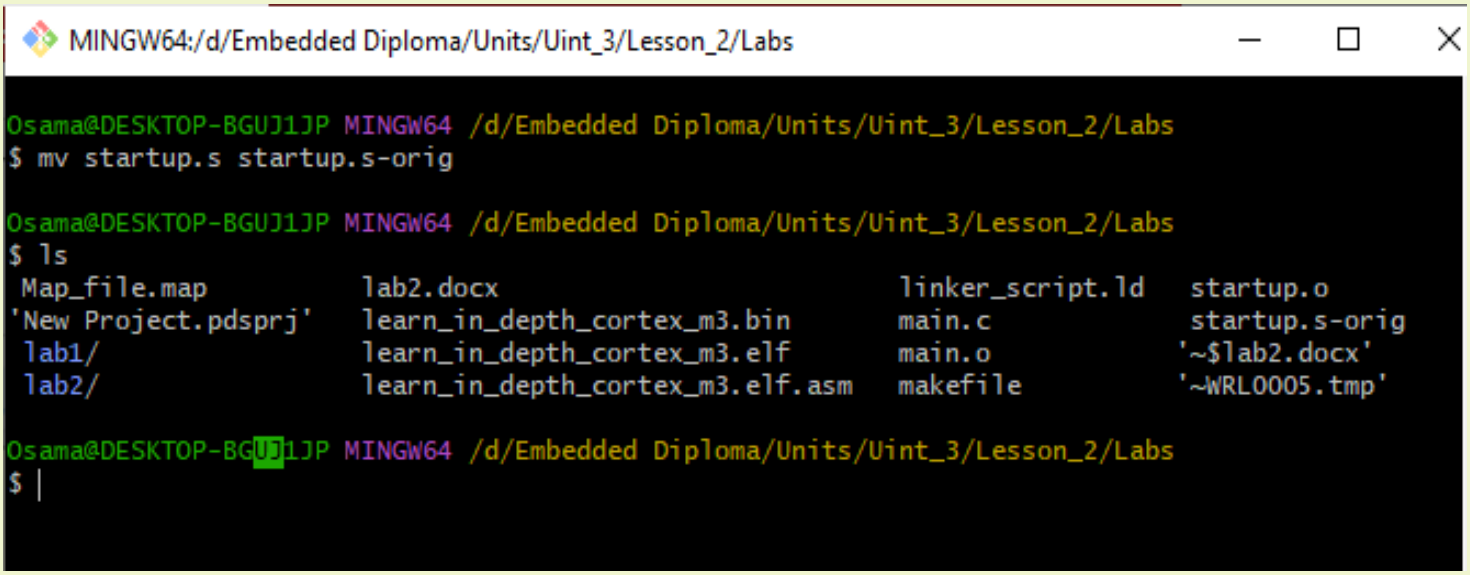
Now we burn the .elf file on Proteus simulator and toggle the led.



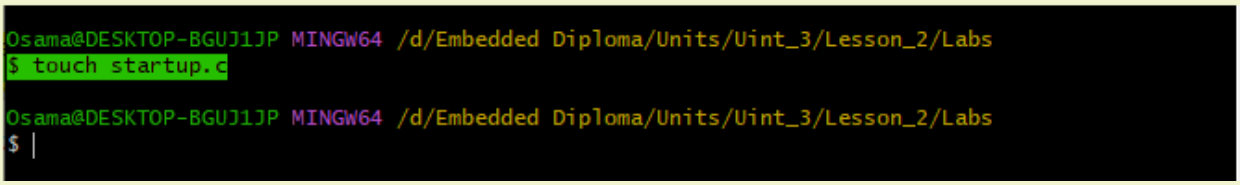
When you click in pause it display the source code for main.c file



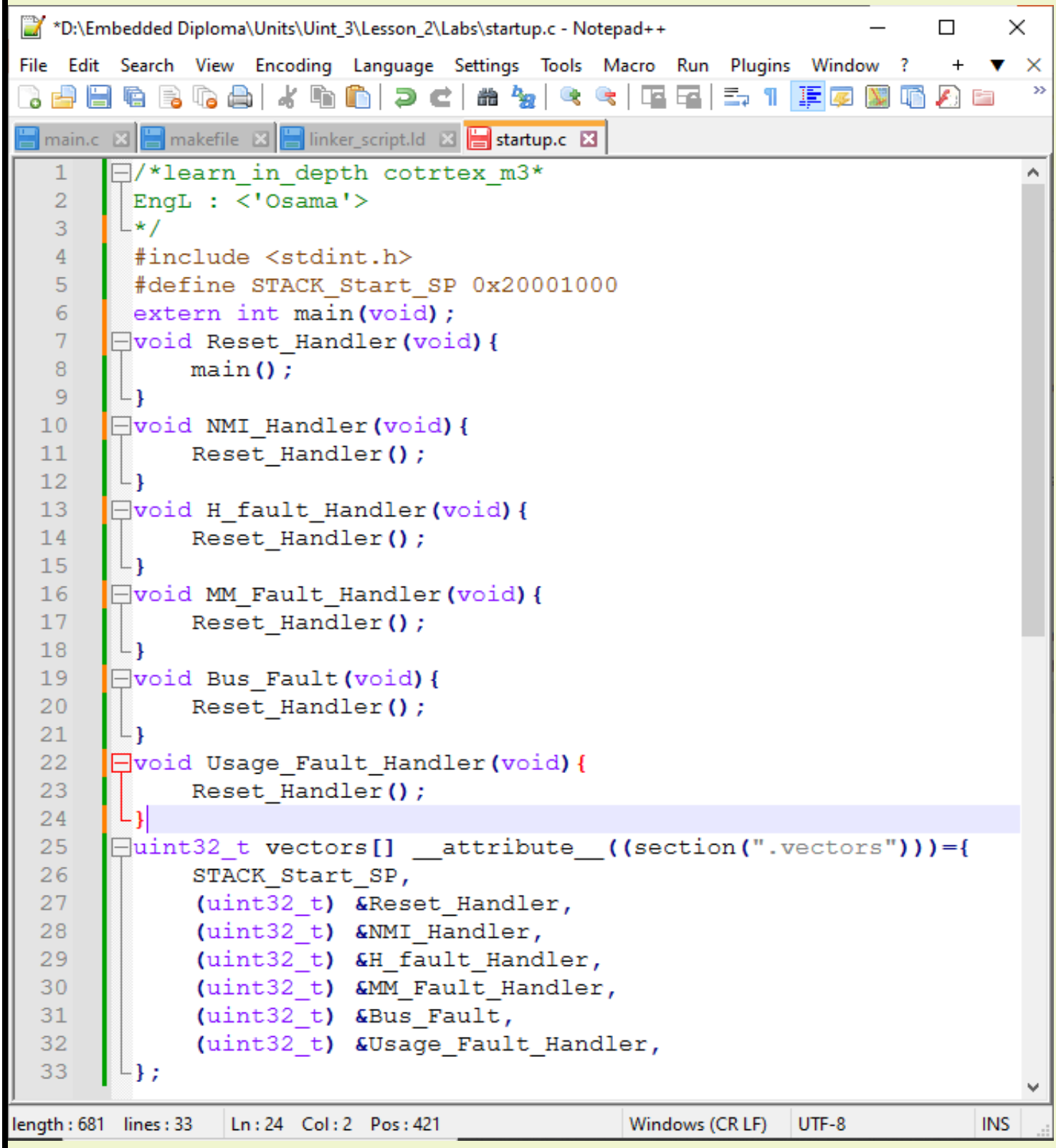
We should take a backup from the startup.s file to avoid any crash happen between startup.c and startup.s by using this command "`$ mv startup.s startup.s-orig`" we take a backup from the startup.s file



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



We can write the startup file with c code “startup.c”



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++

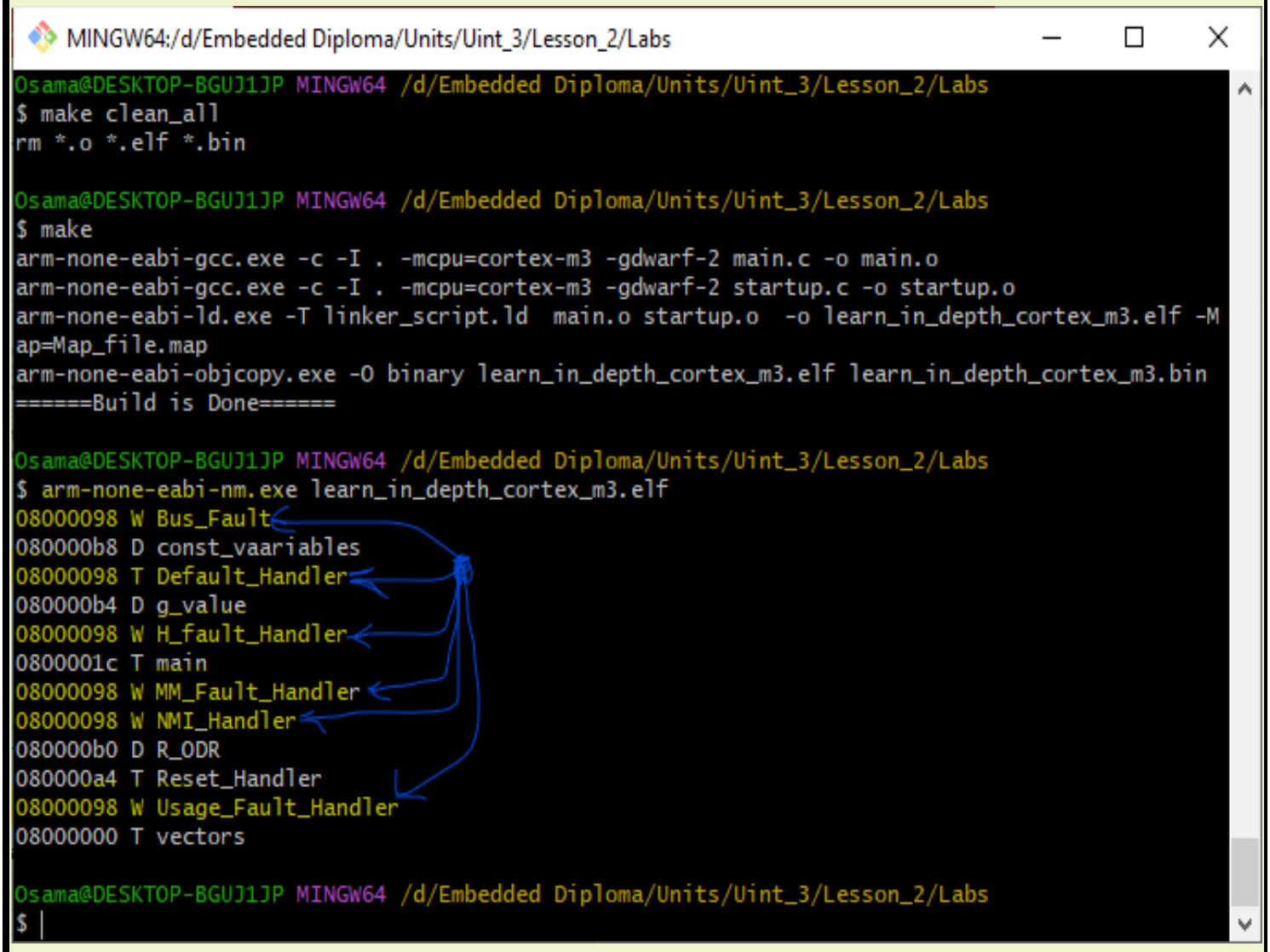
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

main.c | makefile | linker_script.ld | startup.c

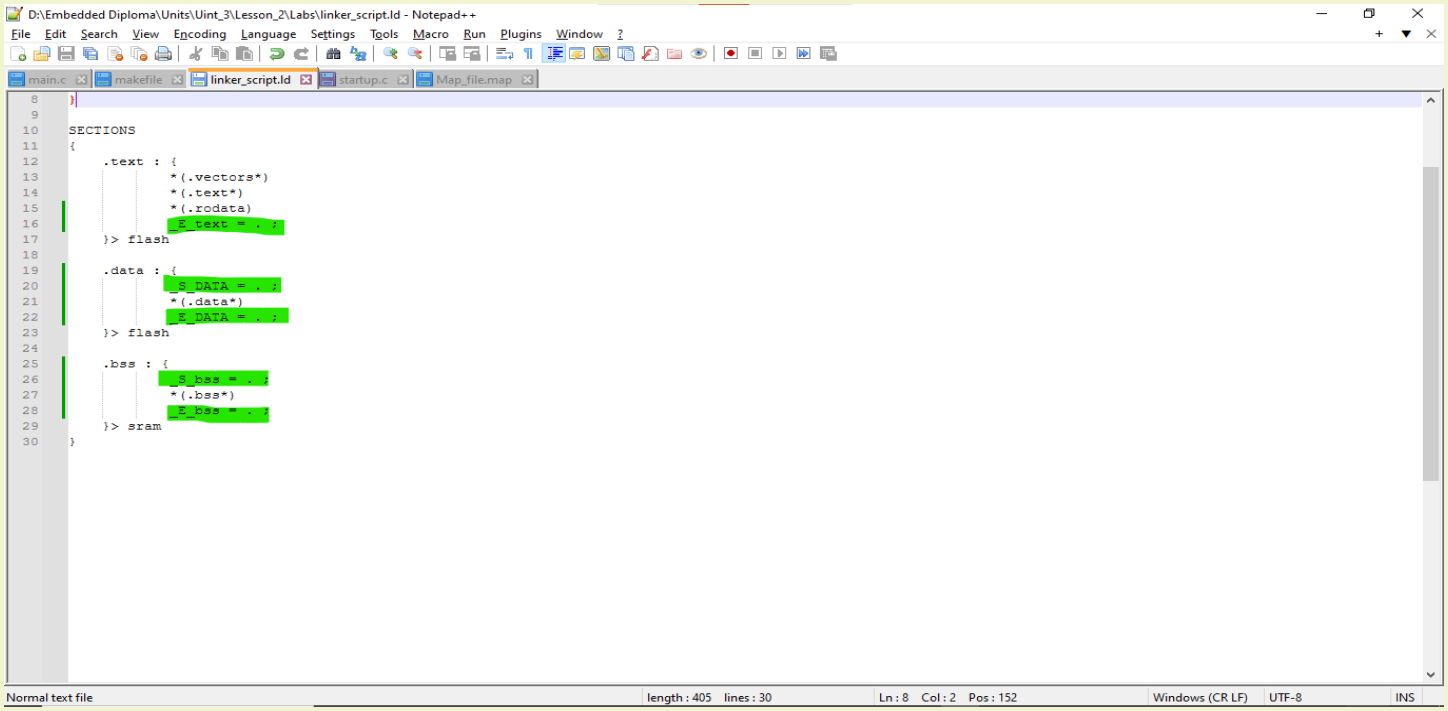
```c
/*learn_in_depth cotrtex_m3*
EngL : <'Osama'>
*/
#include <stdint.h>
#define STACK_Start_SP 0x20001000
extern int main(void);
void Reset_Handler(void);
void Default_Handler()
{
    Reset_Handler();
}
void NMI_Handler(void) __attribute__((weak, alias ("Default_Handler")));;
void H_fault_Handler(void)__attribute__((weak, alias ("Default_Handler")));;
void MM_Fault_Handler(void)__attribute__((weak, alias ("Default_Handler")));;
void Bus_Fault(void)__attribute__((weak, alias ("Default_Handler")));;
void Usage_Fault_Handler(void)__attribute__((weak, alias ("Default_Handler")));;
uint32_t vectors[] __attribute__((section(".vectors")))={
    STACK_Start_SP,
    (uint32_t) &Reset_Handler,
    (uint32_t) &NMI_Handler,
    (uint32_t) &H_fault_Handler,
    (uint32_t) &MM_Fault_Handler,
    (uint32_t) &Bus_Fault,
    (uint32_t) &Usage_Fault_Handler
};
void Reset_Handler(void)
{
    main();
}
```

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

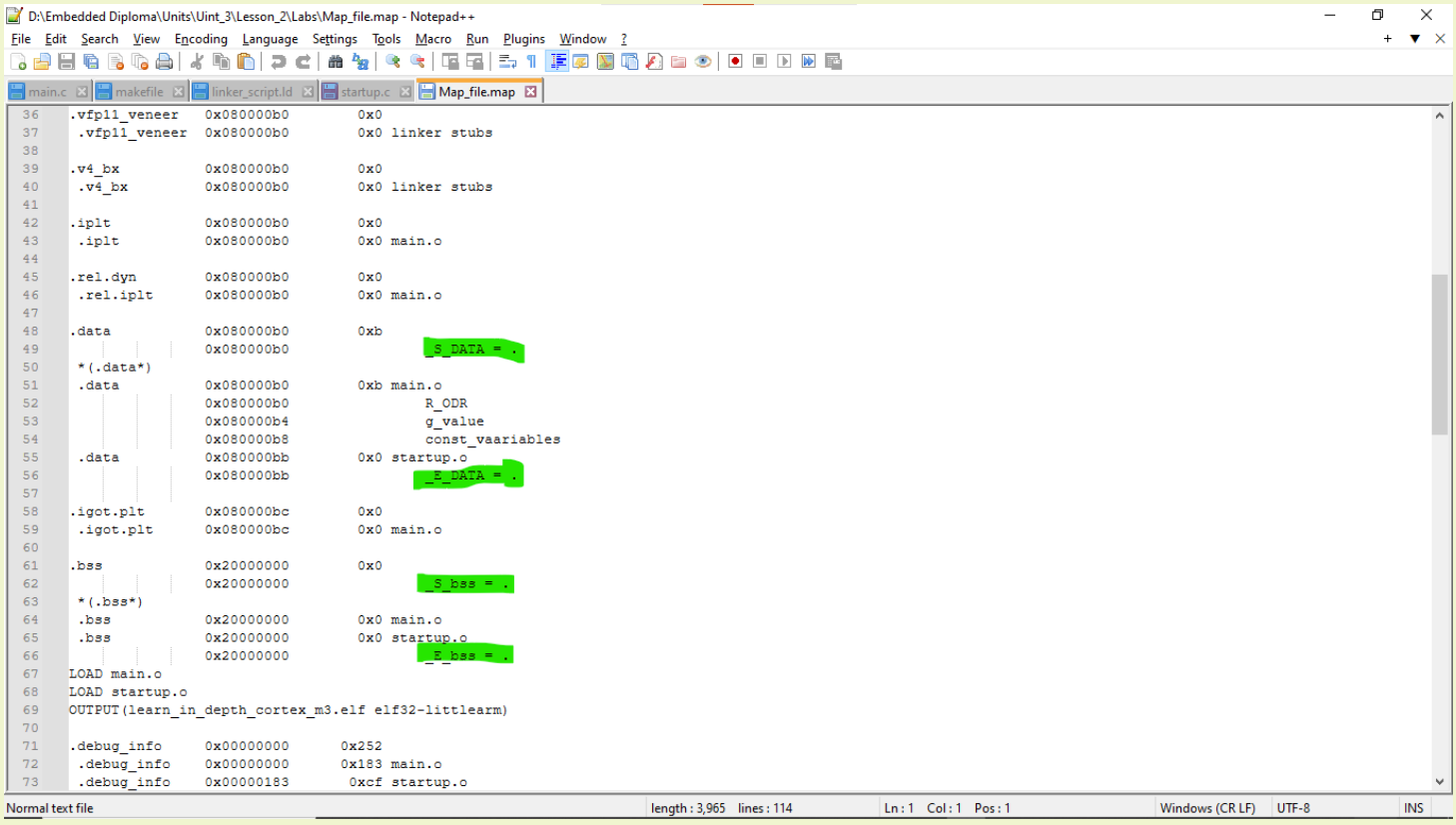
The address of each sections after adding alias and weak to our startup.c file by using this command "`$ arm-none-eabi-nm.exe learn_in_depth_cortex_m3.elf`"



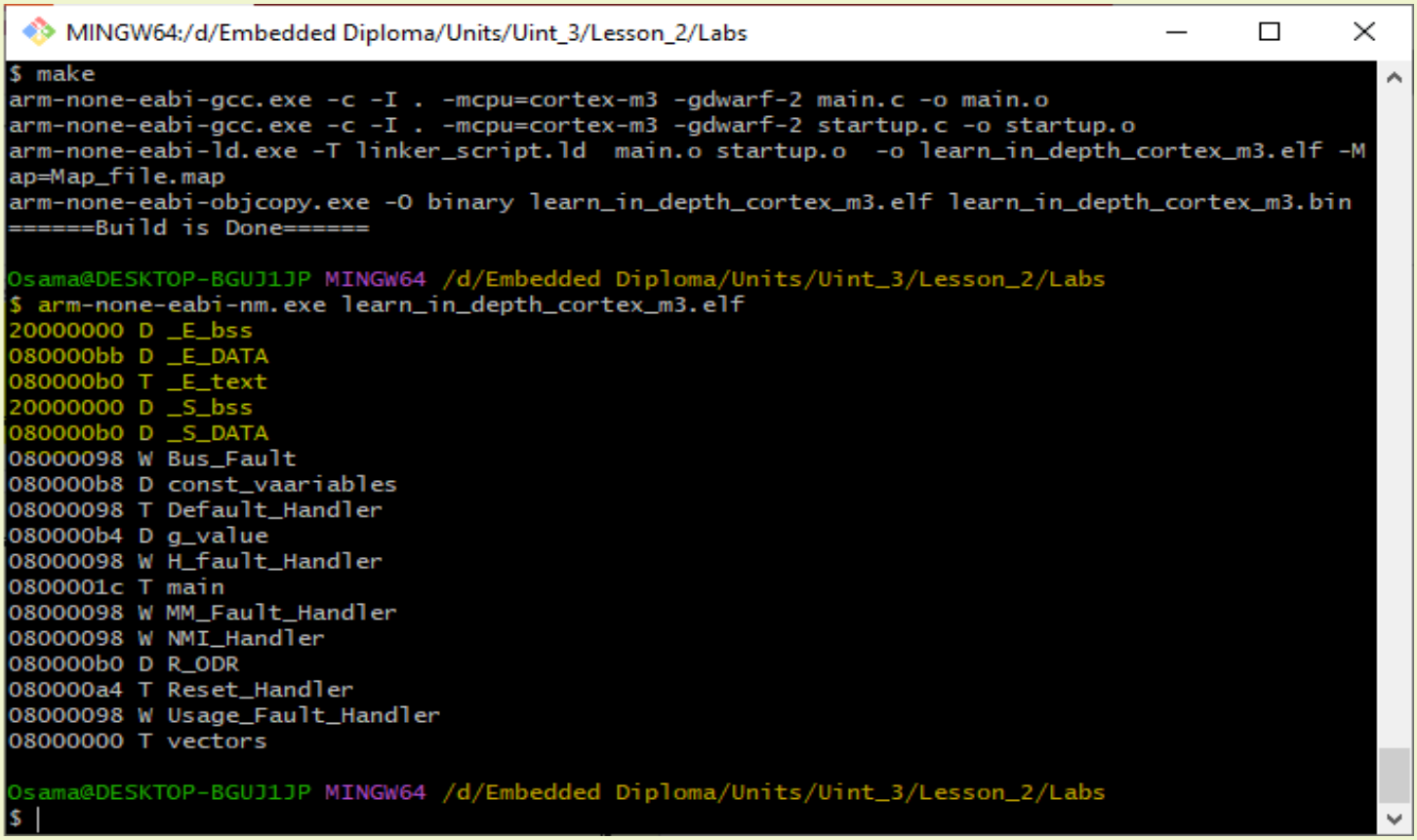
Final step: How to copy (data and create .bss sections)



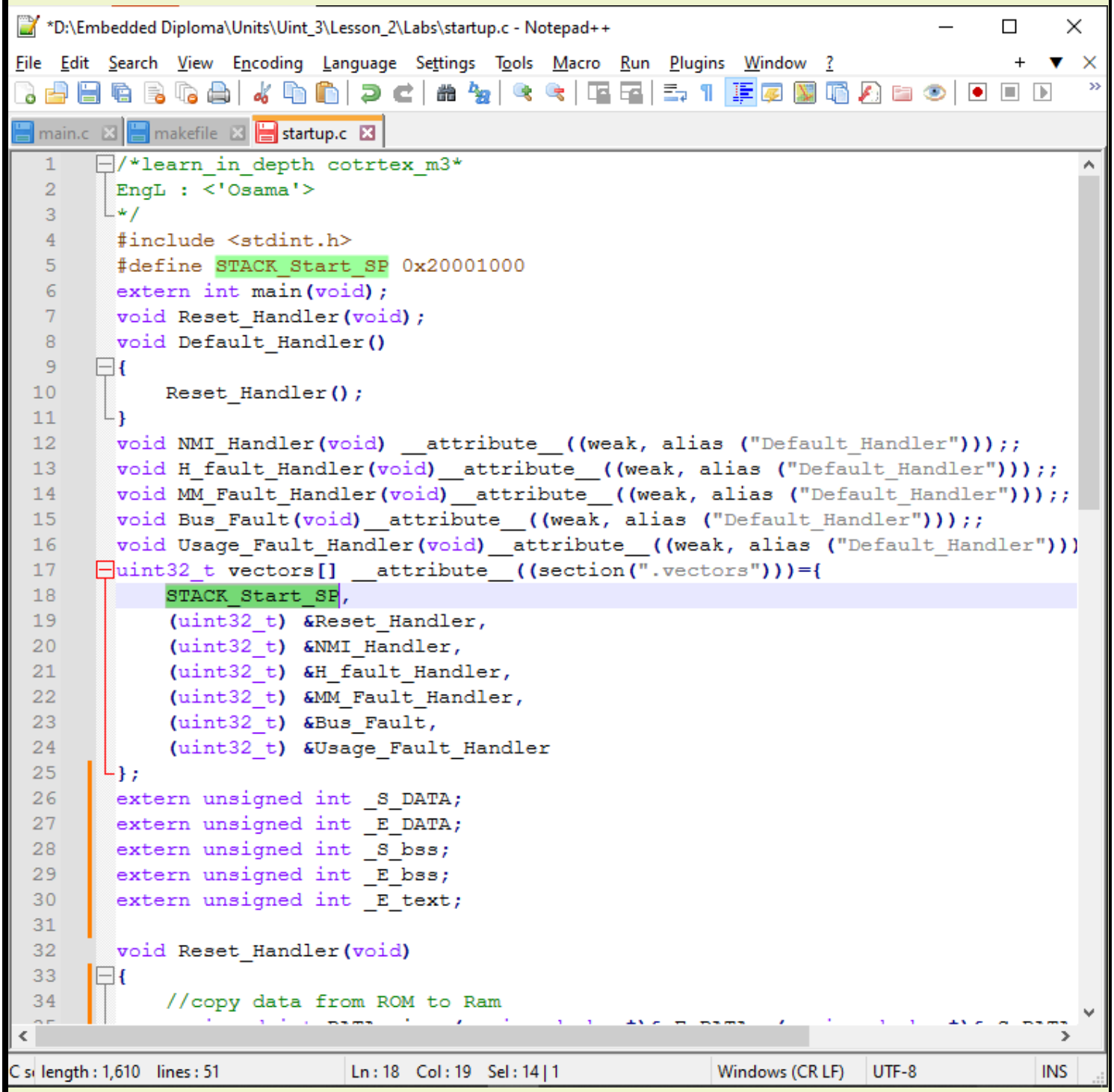
```
D:\Embedded Diploma\Units\Uint_3\Lesson_2\Labs\linker_script.ld - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
main.c | makefile | linker_script.ld | startup.c | Map_file.map
 8  }
 9
10  SECTIONS
11  {
12      .text : {
13              *(.vectors*)
14              *(.text*)
15              *(.rodata)
16              _E_text = . ;
17      }> flash
18
19      .data : {
20              _S_DATA = . ;
21              *(.data*)
22              _E_DATA = . ;
23      }> flash
24
25      .bss : {
26              _S_bss = . ;
27              *(.bss*)
28              _E_bss = . ;
29      }> sram
30  }
Normal text file | length : 405 lines : 30 | Ln : 8 Col : 2 Pos : 152 | Windows (CR LF) | UTF-8 | INS
```



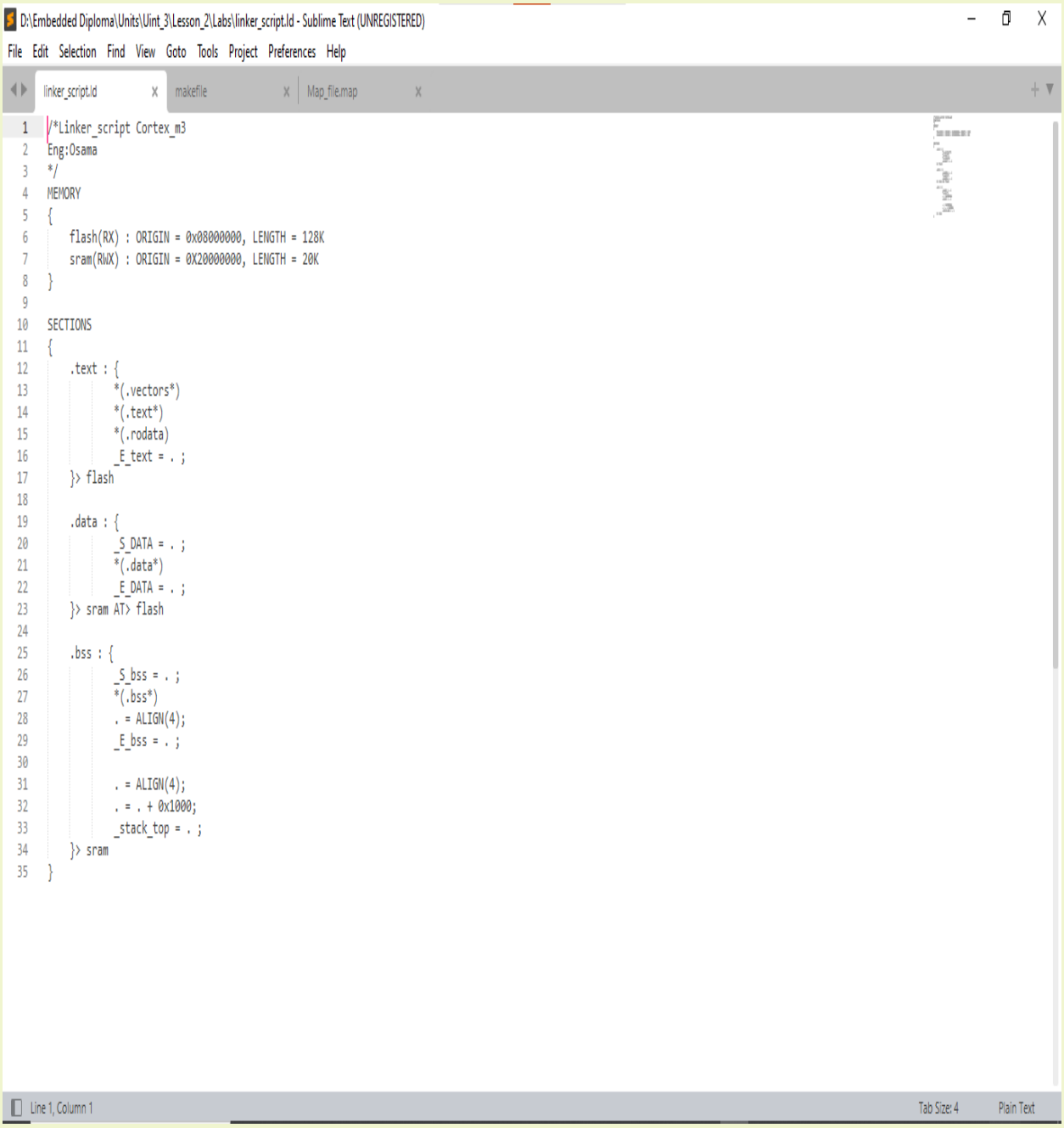
```
D:\Embedded Diploma\Units\Uint_3\Lesson_2\Labs\Map_file.map - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
main.c | makefile | linker_script.ld | startup.c | Map_file.map
36  .vfp11_veneer   0x080000b0        0x0
37   .vfp11_veneer  0x080000b0        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                  0x080000b0                _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                _E_DATA = .
57
58  .igot.plt       0x080000bc        0x0
59   .igot.plt      0x080000bc        0x0 main.o
60
61  .bss            0x20000000        0x0
62                  0x20000000                _S_bss = .
63   *(.bss*)
64   .bss           0x20000000        0x0 main.o
65   .bss           0x20000000        0x0 startup.o
66                  0x20000000                _E_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
$

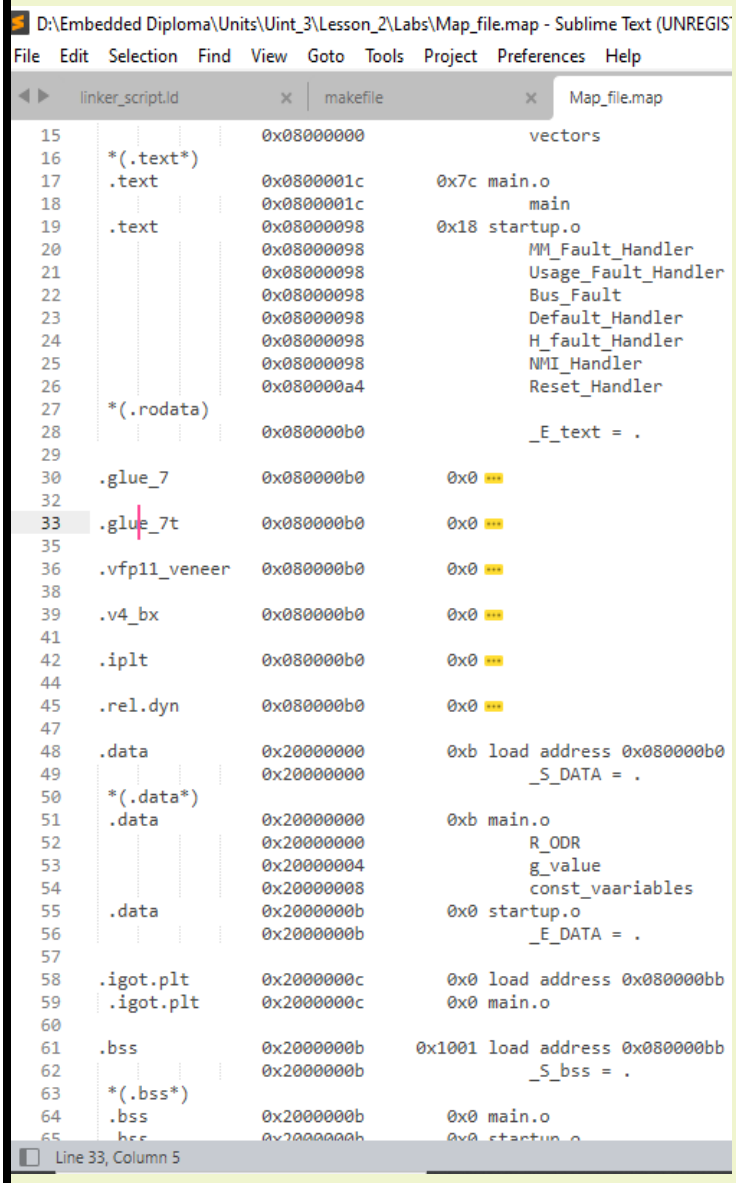
startup.c after adding start && ends of sections

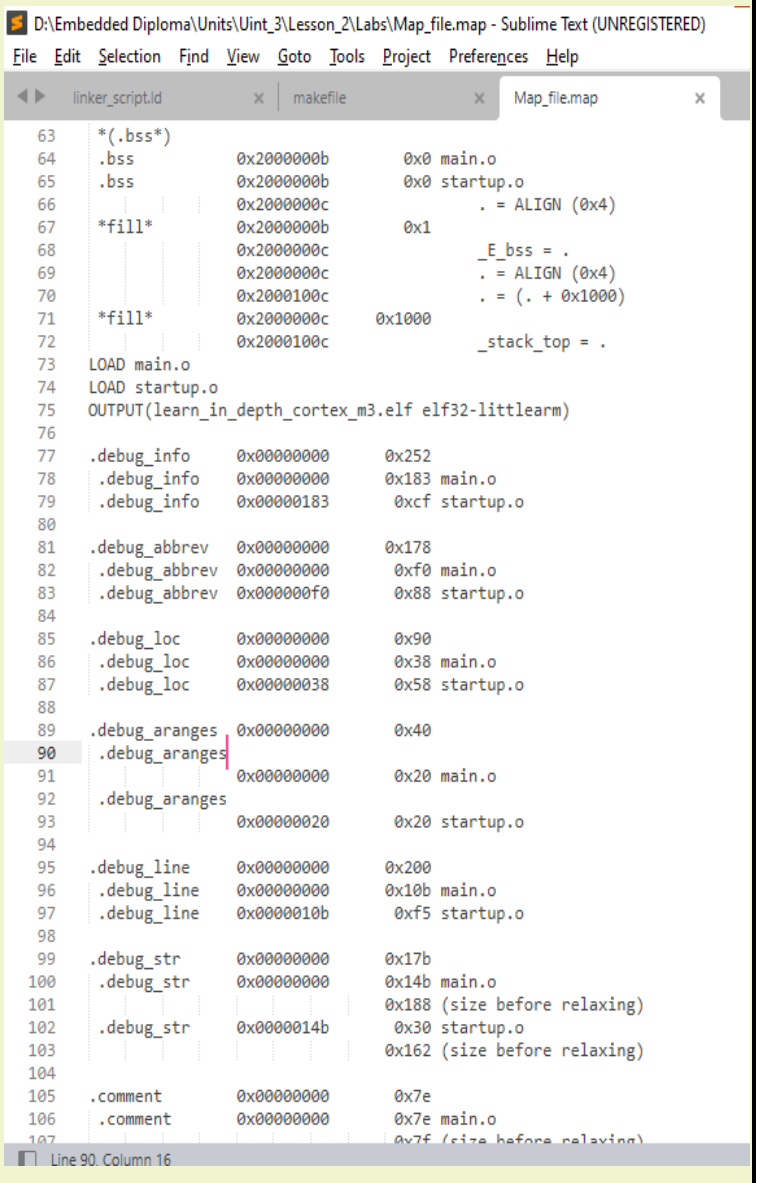


Linker script



Map file


D:\Embedded Diploma\Units\Uint_3\Lesson_2\Labs\Map_file.map - Sublime Text (UNREGIS
File Edit Selection Find View Goto Tools Project Preferences Help
linker_script.ld
makefile
Map_file.map
15 0x08000000 vectors
16 *(.text*)
17 .text 0x0800001c 0x7c main.o
18 0x0800001c main
19 .text 0x08000098 0x18 startup.o
20 0x08000098 MM_Fault_Handler
21 0x08000098 Usage_Fault_Handler
22 0x08000098 Bus_Fault
23 0x08000098 Default_Handler
24 0x08000098 H_fault_Handler
25 0x08000098 NMI_Handler
26 0x080000a4 Reset_Handler
27 *(.rodata)
28 0x080000b0 _E_text = .
29
30 .glue_7 0x080000b0 0x0 ...
32
33 .glue_7t 0x080000b0 0x0 ...
35
36 .vfp11_veneer 0x080000b0 0x0 ...
38
39 .v4_bx 0x080000b0 0x0 ...
41
42 .iplt 0x080000b0 0x0 ...
44
45 .rel.dyn 0x080000b0 0x0 ...
47
48 .data 0x20000000 0xb load address 0x080000b0
49 0x20000000 _S_DATA = .
50 *(.data*)
51 .data 0x20000000 0xb main.o
52 0x20000000 R_ODR
53 0x20000004 g_value
54 0x20000008 const_vaariables
55 .data 0x2000000b 0x0 startup.o
56 0x2000000b _E_DATA = .
57
58 .igot.plt 0x2000000c 0x0 load address 0x080000bb
59 .igot.plt 0x2000000c 0x0 main.o
60
61 .bss 0x2000000b 0x1001 load address 0x080000bb
62 0x2000000b _S_bss = .
63 *(.bss*)
64 .bss 0x2000000b 0x0 main.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
makefile
Map_file.map
63 *(.bss*)
64 .bss 0x2000000b 0x0 main.o
65 .bss 0x2000000b 0x0 startup.o
66 0x2000000c . = ALIGN (0x4)
67 *fill* 0x2000000b 0x1
68 0x2000000c _E_bss = .
69 0x2000000c . = ALIGN (0x4)
70 0x2000100c . = (. + 0x1000)
71 *fill* 0x2000000c 0x1000
72 0x2000100c _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 0x00000183 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
91 0x00000000 0x20 main.o
92 .debug_aranges
93 0x00000020 0x20 startup.o
94
95 .debug_line 0x00000000 0x200
96 .debug_line 0x00000000 0x10b main.o
97 .debug_line 0x0000010b 0xf5 startup.o
98
99 .debug_str 0x00000000 0x17b
100 .debug_str 0x00000000 0x14b main.o
101 0x188 (size before relaxing)
102 .debug_str 0x0000014b 0x30 startup.o
103 0x162 (size before relaxing)
104
105 .comment 0x00000000 0x7e
106 .comment 0x00000000 0x7e main.o
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 -M
ap=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
$
```