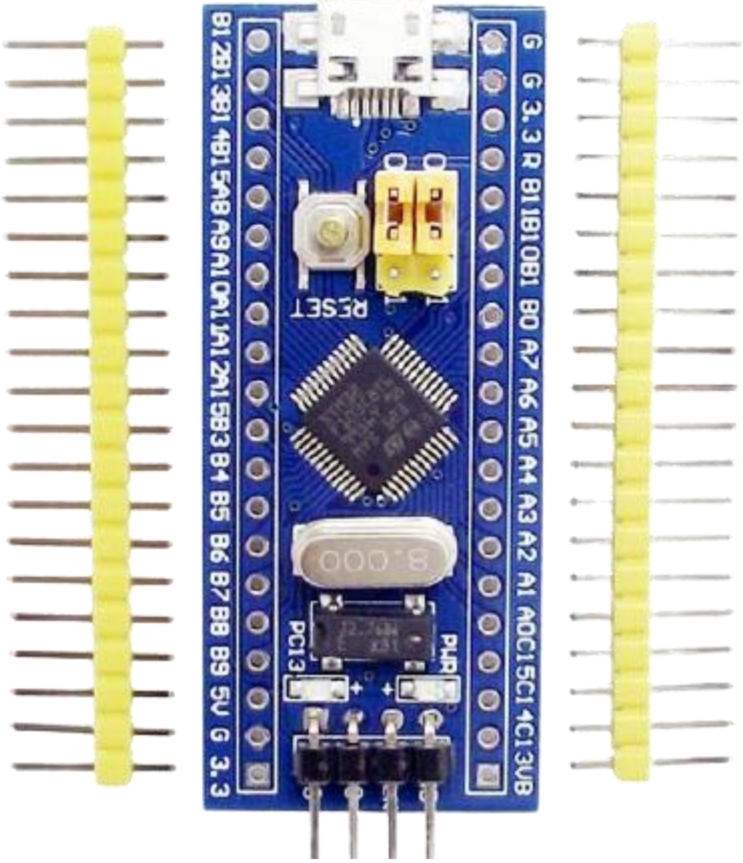
Embedded C Lab 2

*****

ARM CORTEX-M3 STM32F103


B12 B13 B14 B15 A8 A9 A10 A11 A12 A15 B3 B4 B5 B6 B7 B8 B9 5V G 3.3
G G 3.3 R B11 B10 B1 B0 A7 A6 A5 A4 A3 A2 A1 A0 C15 C14 C13 VB
RESET
8.000
PC13
PWR

Name: Mina Karam Fahmy

In this lab, I will write a bare-metal software to toggle led is connected to GPIO port C13, to make a GPIO toggling in STM32. After reading its specs I Found that:

I need to work with two peripherals:

- RCC (reset and clock control in 0x18 APB2ENR Register in IOPAEN Bit 2) is necessary because GPIO has disabled clock by default. Its base address is 0x40021000.
- GPIO (general purpose input/output) so I connected it to GPIO Port A with address 0x40010800, in GPIO_PA I have to read 2 in 0x04 CRH Register to active mode pin 13 (from bit 20 to 24) and in 0x0C ODR Register (Pin 13).

Now I have all what I need to write application file (I call it main.c).

Next step it to read also its specs because some parts of startup code dependent on the target processor.

When power is applied to the MCU the program counter (PC) value will be 0, which will map to 0x08000000 and will therefore start at address 0x080000000. This address is ten copied to the stack pointer (SP) register for later use.

The program Counter then steps to the next address which is 0x08000004 and expects the address of the reset handler at this location then the next handler from the vector table handler.

Also, I need .thumb_func, this directive specifies that the following symbol is the name of a Thumb encoded function. This information is necessary in order to allow the assembler and linker to generate correct code for interworking between arm and thumb instructions. Because there are 2 types of instructions are provided in the processor are: 16-bits instruction and 32-bit instruction.

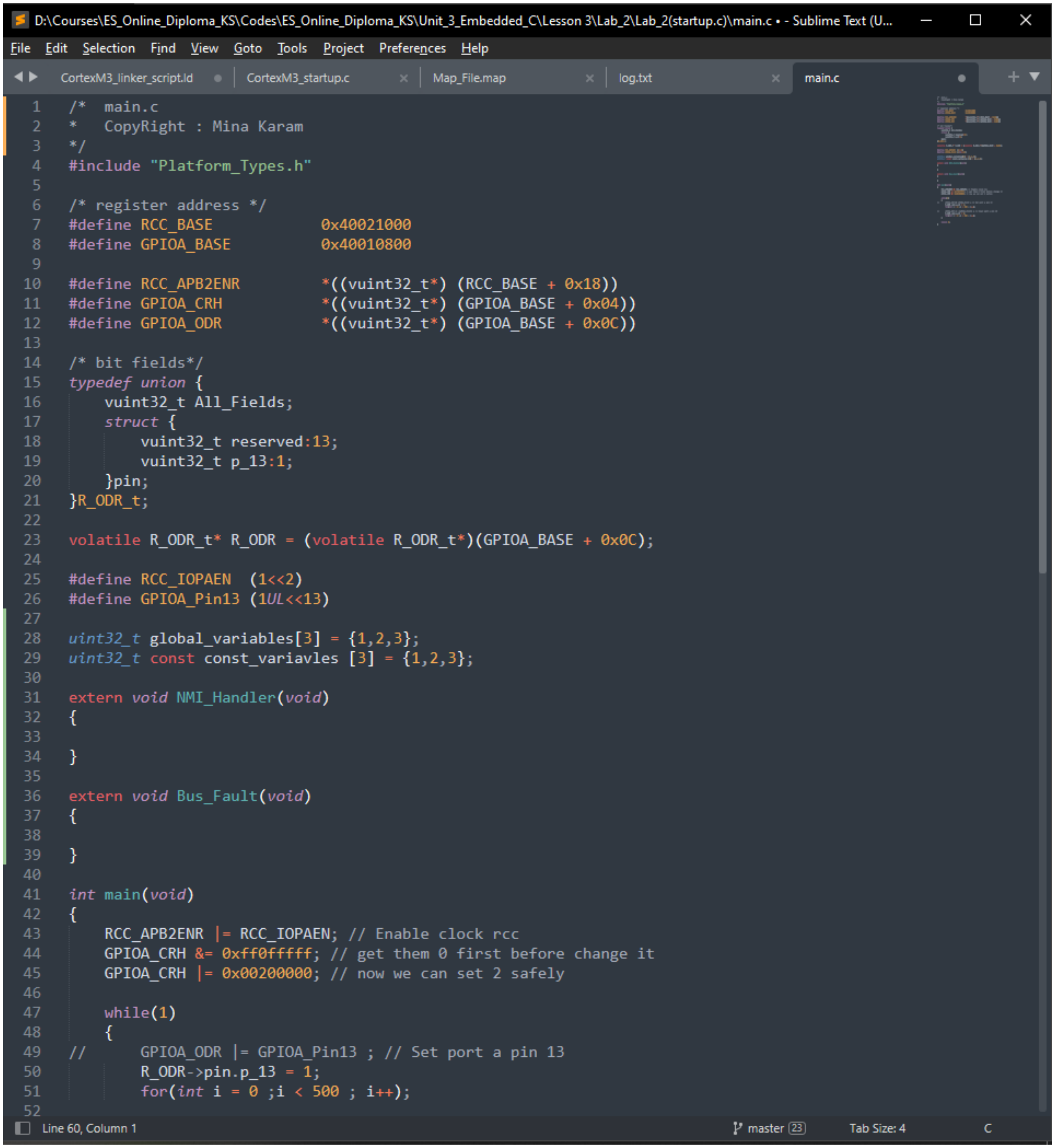
So, I created a complex startup that consists of:

1- Define Interrupt Vectors Section.
2- Copy Data form ROM to RAM.
3- Initialize Data Area.
4- Initialize Stack.
5- Create a reset section and call main function.

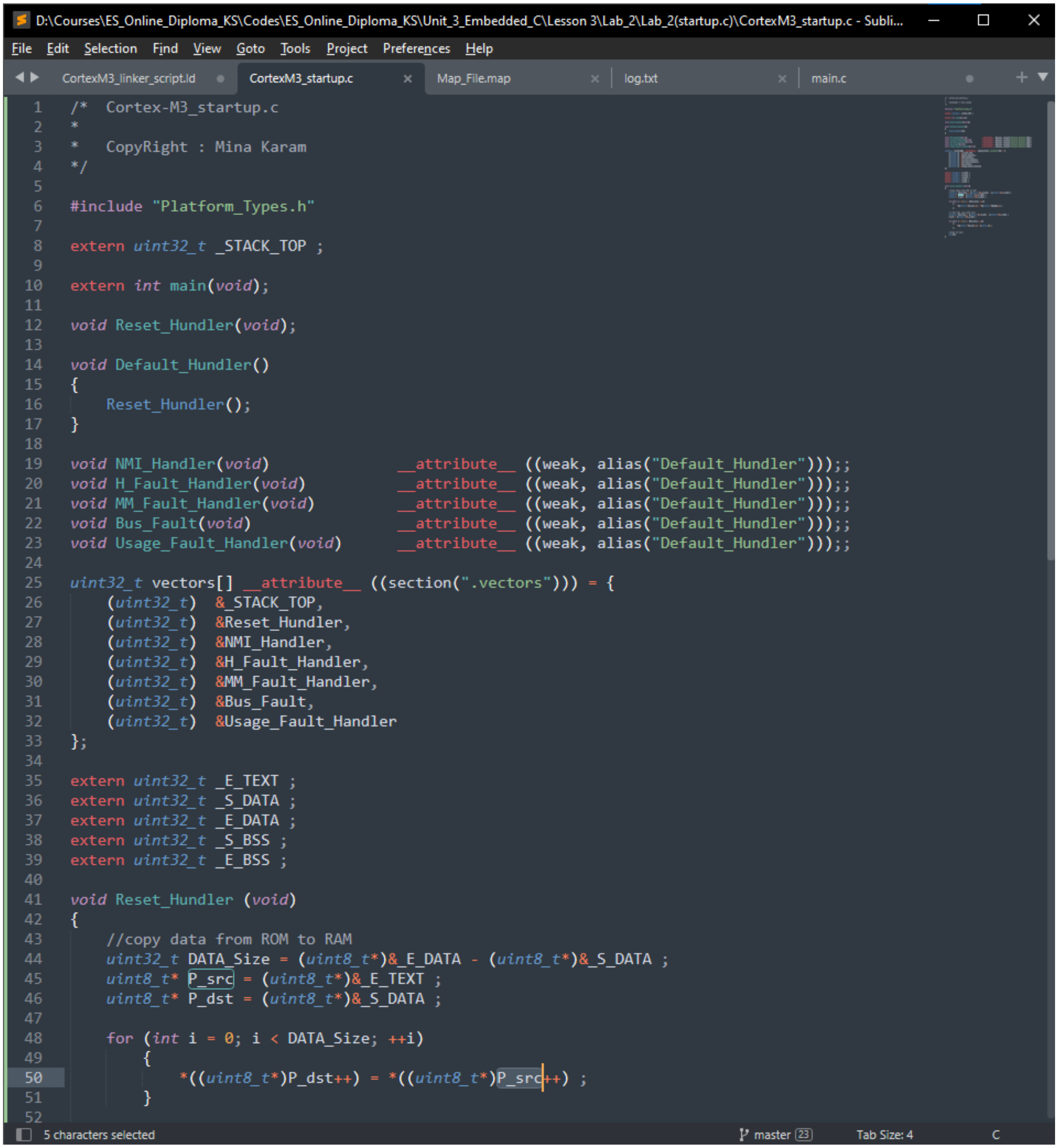
Before create it and because of SP initialized automated by the processor we can write startup.s and startup.c that feature is provided only for cortex-M3 and its family. So, I wrote the 2 files.

And in the linker file, I need to Aligned access memory to Efficiency fetch and execute so I used this command in linker script file (. = ALIGN(4)).

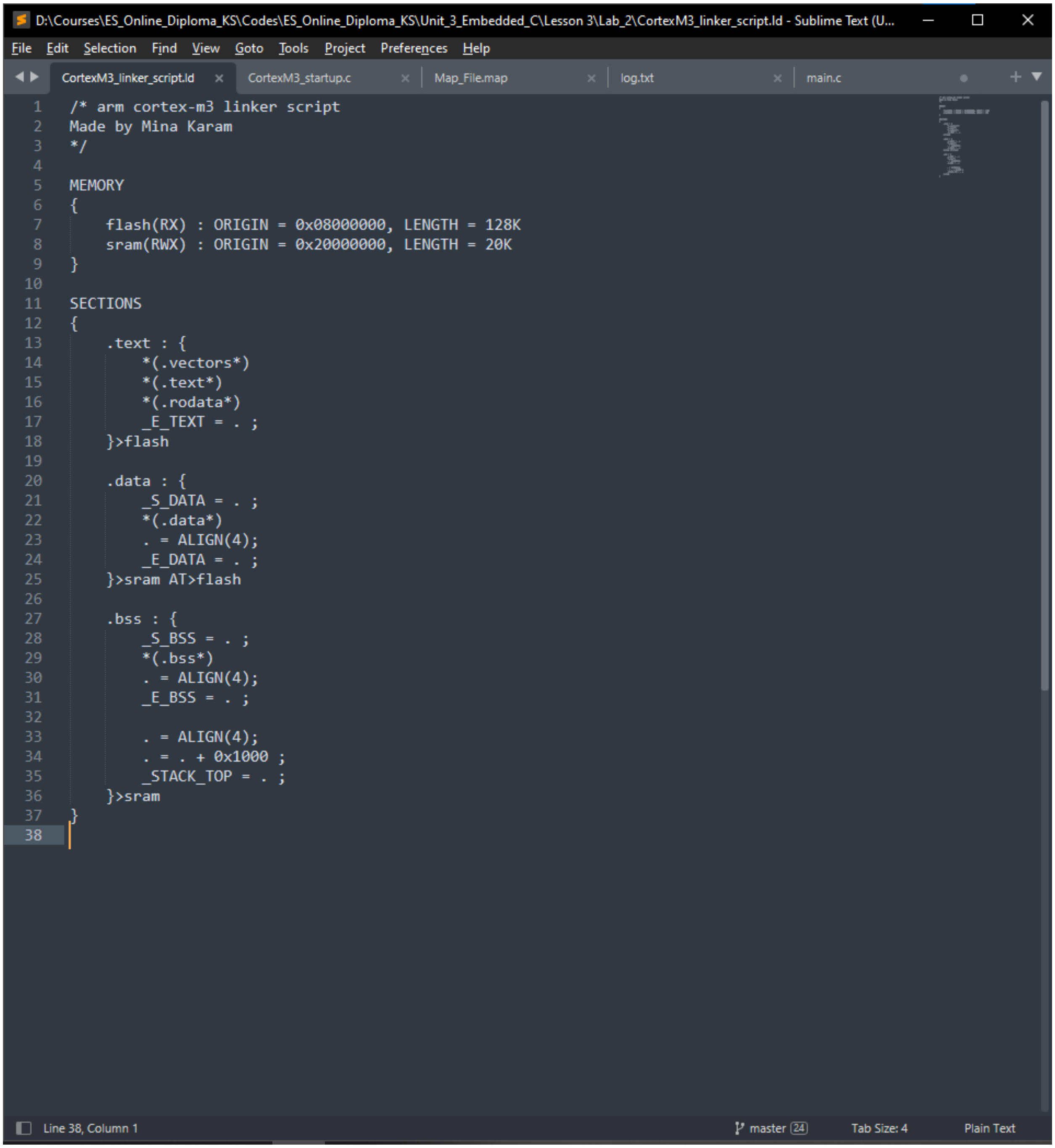
All the above information that I need it, to write a bare-metal software to toggle the led's start ….


D:\Courses\ES_Online_Diploma_KS\Codes\ES_Online_Diploma_KS\Unit_3_Embedded_C\Lesson 3\Lab_2\Lab_2(startup.c)\main.c • - Sublime Text (U...
File Edit Selection Find View Goto Tools Project Preferences Help
CortexM3_linker_script.ld
CortexM3_startup.c
Map_File.map
log.txt
main.c

```c
/*  main.c
*   CopyRight : Mina Karam
*/
#include "Platform_Types.h"

/* register address */
#define RCC_BASE            0x40021000
#define GPIOA_BASE          0x40010800

#define RCC_APB2ENR         *((vuint32_t*) (RCC_BASE + 0x18))
#define GPIOA_CRH           *((vuint32_t*) (GPIOA_BASE + 0x04))
#define GPIOA_ODR           *((vuint32_t*) (GPIOA_BASE + 0x0C))

/* bit fields*/
typedef union {
    vuint32_t All_Fields;
    struct {
        vuint32_t reserved:13;
        vuint32_t p_13:1;
    }pin;
}R_ODR_t;

volatile R_ODR_t* R_ODR = (volatile R_ODR_t*)(GPIOA_BASE + 0x0C);

#define RCC_IOPAEN  (1<<2)
#define GPIOA_Pin13 (1UL<<13)

uint32_t global_variables[3] = {1,2,3};
uint32_t const const_variavles [3] = {1,2,3};

extern void NMI_Handler(void)
{

}

extern void Bus_Fault(void)
{

}

int main(void)
{
    RCC_APB2ENR |= RCC_IOPAEN; // Enable clock rcc
    GPIOA_CRH &= 0xff0fffff; // get them 0 first before change it
    GPIOA_CRH |= 0x00200000; // now we can set 2 safely

    while(1)
    {
//      GPIOA_ODR |= GPIOA_Pin13 ; // Set port a pin 13
        R_ODR->pin.p_13 = 1;
        for(int i = 0 ;i < 500 ; i++);

```

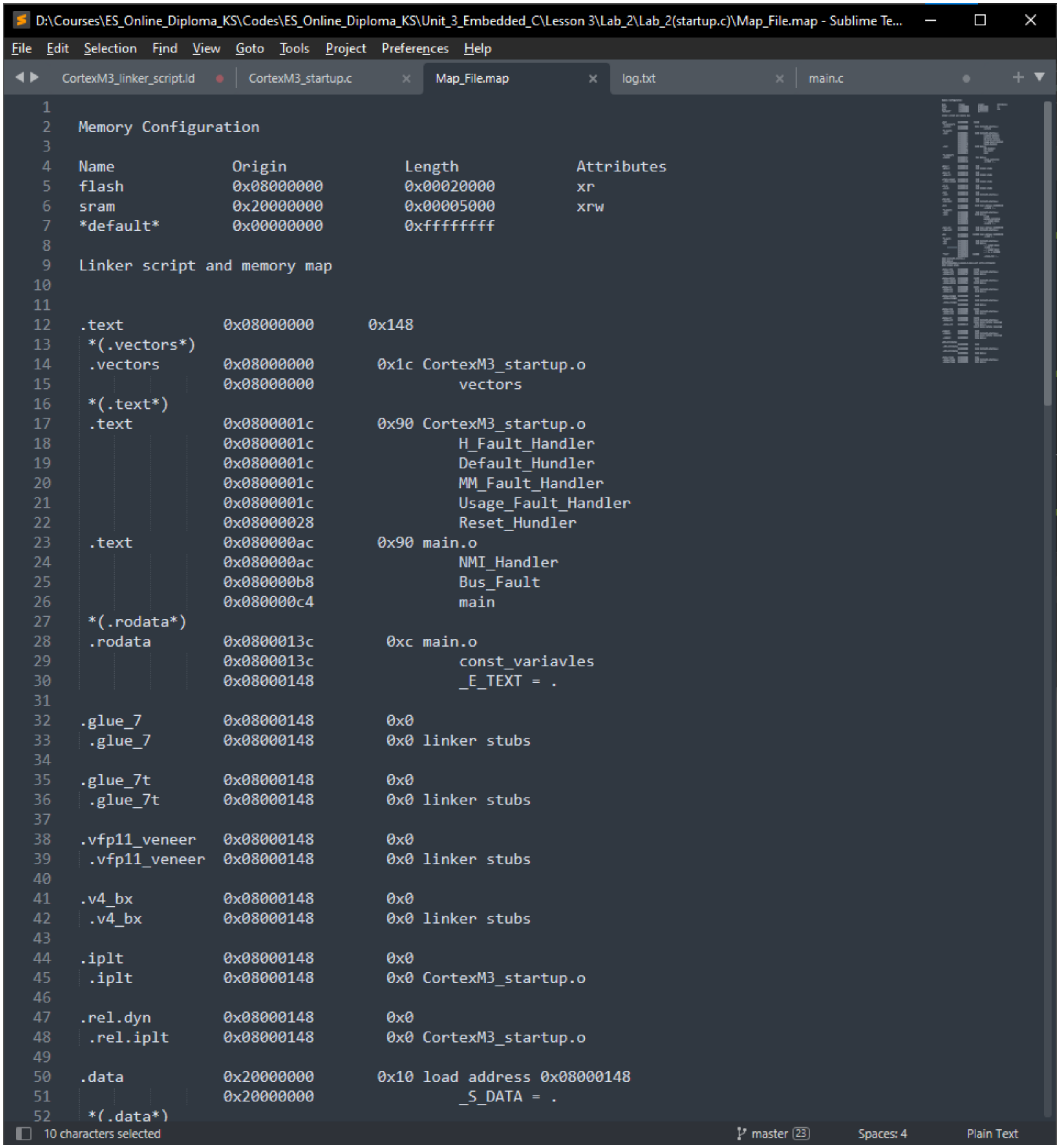

D:\Courses\ES_Online_Diploma_KS\Codes\ES_Online_Diploma_KS\Unit_3_Embedded_C\Lesson 3\Lab_2\Lab_2(startup.c)\CortexM3_startup.c - Subli...
File Edit Selection Find View Goto Tools Project Preferences Help
CortexM3_linker_script.ld
CortexM3_startup.c
Map_File.map
log.txt
main.c

```c
/*  Cortex-M3_startup.c
 *
 *  CopyRight : Mina Karam
 */

#include "Platform_Types.h"

extern uint32_t _STACK_TOP ;

extern int main(void);

void Reset_Hundler(void);

void Default_Hundler()
{
    Reset_Hundler();
}

void NMI_Handler(void)              __attribute__ ((weak, alias("Default_Hundler")));;
void H_Fault_Handler(void)          __attribute__ ((weak, alias("Default_Hundler")));;
void MM_Fault_Handler(void)         __attribute__ ((weak, alias("Default_Hundler")));;
void Bus_Fault(void)                __attribute__ ((weak, alias("Default_Hundler")));;
void Usage_Fault_Handler(void)      __attribute__ ((weak, alias("Default_Hundler")));;

uint32_t vectors[] __attribute__ ((section(".vectors"))) = {
    (uint32_t)  &_STACK_TOP,
    (uint32_t)  &Reset_Hundler,
    (uint32_t)  &NMI_Handler,
    (uint32_t)  &H_Fault_Handler,
    (uint32_t)  &MM_Fault_Handler,
    (uint32_t)  &Bus_Fault,
    (uint32_t)  &Usage_Fault_Handler
};

extern uint32_t _E_TEXT ;
extern uint32_t _S_DATA ;
extern uint32_t _E_DATA ;
extern uint32_t _S_BSS ;
extern uint32_t _E_BSS ;

void Reset_Hundler (void)
{
    //copy data from ROM to RAM
    uint32_t DATA_Size = (uint8_t*)&_E_DATA - (uint8_t*)&_S_DATA ;
    uint8_t* P_src = (uint8_t*)&_E_TEXT ;
    uint8_t* P_dst = (uint8_t*)&_S_DATA ;

    for (int i = 0; i < DATA_Size; ++i)
        {
            *((uint8_t*)P_dst++) = *((uint8_t*)P_src++) ;
        }

```



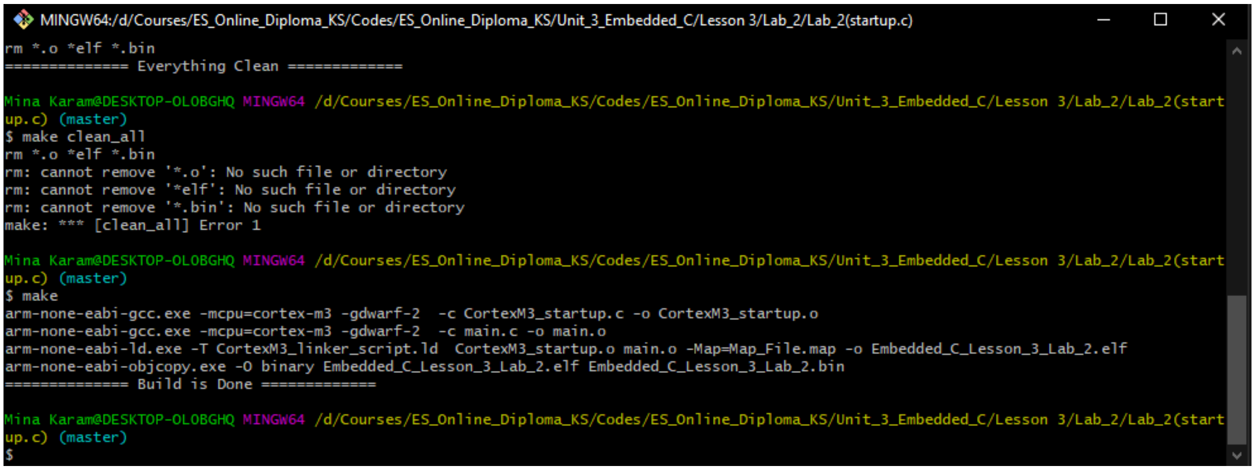
D:\Courses\ES_Online_Diploma_KS\Codes\ES_Online_Diploma_KS\Unit_3_Embedded_C\Lesson 3\Lab_2\CortexM3_linker_script.ld - Sublime Text (U...
File Edit Selection Find View Goto Tools Project Preferences Help
CortexM3_linker_script.ld
CortexM3_startup.c
Map_File.map
log.txt
main.c
Line 38, Column 1
master 24
Tab Size: 4
Plain Text

```
/* arm cortex-m3 linker script
Made by Mina Karam
*/

MEMORY
{
    flash(RX) : ORIGIN = 0x08000000, LENGTH = 128K
    sram(RWX) : ORIGIN = 0x20000000, LENGTH = 20K
}

SECTIONS
{
    .text : {
        *(.vectors*)
        *(.text*)
        *(.rodata*)
        _E_TEXT = . ;
    }>flash

    .data : {
        _S_DATA = . ;
        *(.data*)
        . = ALIGN(4);
        _E_DATA = . ;
    }>sram AT>flash

    .bss : {
        _S_BSS = . ;
        *(.bss*)
        . = ALIGN(4);
        _E_BSS = . ;

        . = ALIGN(4);
        . = . + 0x1000 ;
        _STACK_TOP = . ;
    }>sram
}
```



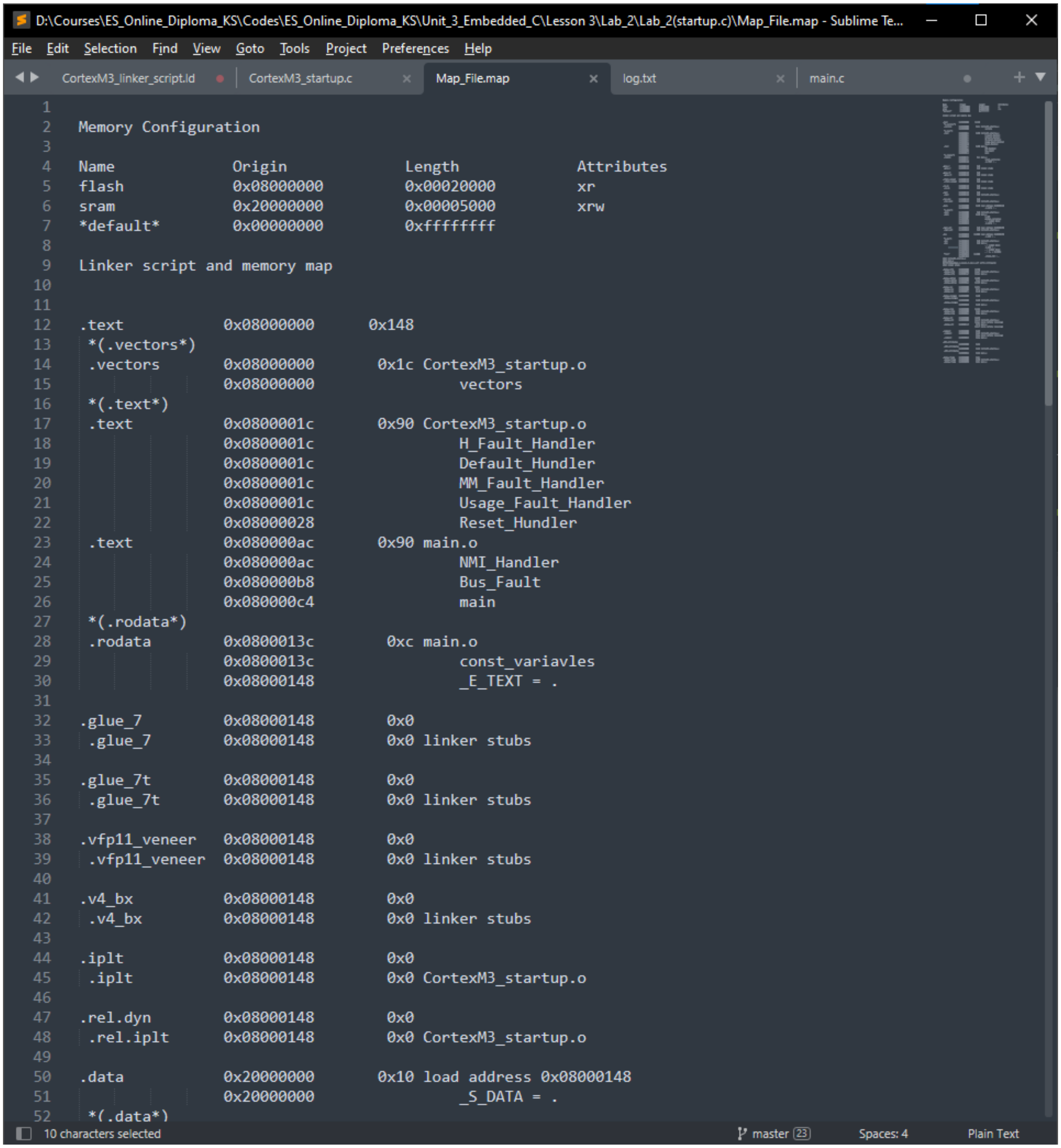
D:\Courses\ES_Online_Diploma_KS\Codes\ES_Online_Diploma_KS\Unit_3_Embedded_C\Lesson 3\Lab_2\Lab_2(startup.c)\Map_File.map - Sublime Te...
File Edit Selection Find View Goto Tools Project Preferences Help
CortexM3_linker_script.ld
CortexM3_startup.c
Map_File.map
log.txt
main.c
10 characters selected
master 23
Spaces: 4
Plain Text

```

Memory Configuration

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

Linker script and memory map


.text           0x08000000      0x148
 *(.vectors*)
 .vectors       0x08000000       0x1c CortexM3_startup.o
                0x08000000                vectors
 *(.text*)
 .text          0x0800001c       0x90 CortexM3_startup.o
                0x0800001c                H_Fault_Handler
                0x0800001c                Default_Hundler
                0x0800001c                MM_Fault_Handler
                0x0800001c                Usage_Fault_Handler
                0x08000028                Reset_Hundler
 .text          0x080000ac       0x90 main.o
                0x080000ac                NMI_Handler
                0x080000b8                Bus_Fault
                0x080000c4                main
 *(.rodata*)
 .rodata        0x0800013c        0xc main.o
                0x0800013c                const_variavles
                0x08000148                _E_TEXT = .

.glue_7         0x08000148        0x0
 .glue_7        0x08000148        0x0 linker stubs

.glue_7t        0x08000148        0x0
 .glue_7t       0x08000148        0x0 linker stubs

.vfp11_veneer   0x08000148        0x0
 .vfp11_veneer  0x08000148        0x0 linker stubs

.v4_bx          0x08000148        0x0
 .v4_bx         0x08000148        0x0 linker stubs

.iplt           0x08000148        0x0
 .iplt          0x08000148        0x0 CortexM3_startup.o

.rel.dyn        0x08000148        0x0
 .rel.iplt      0x08000148        0x0 CortexM3_startup.o

.data           0x20000000       0x10 load address 0x08000148
                0x20000000                _S_DATA = .
 *(.data*)
```



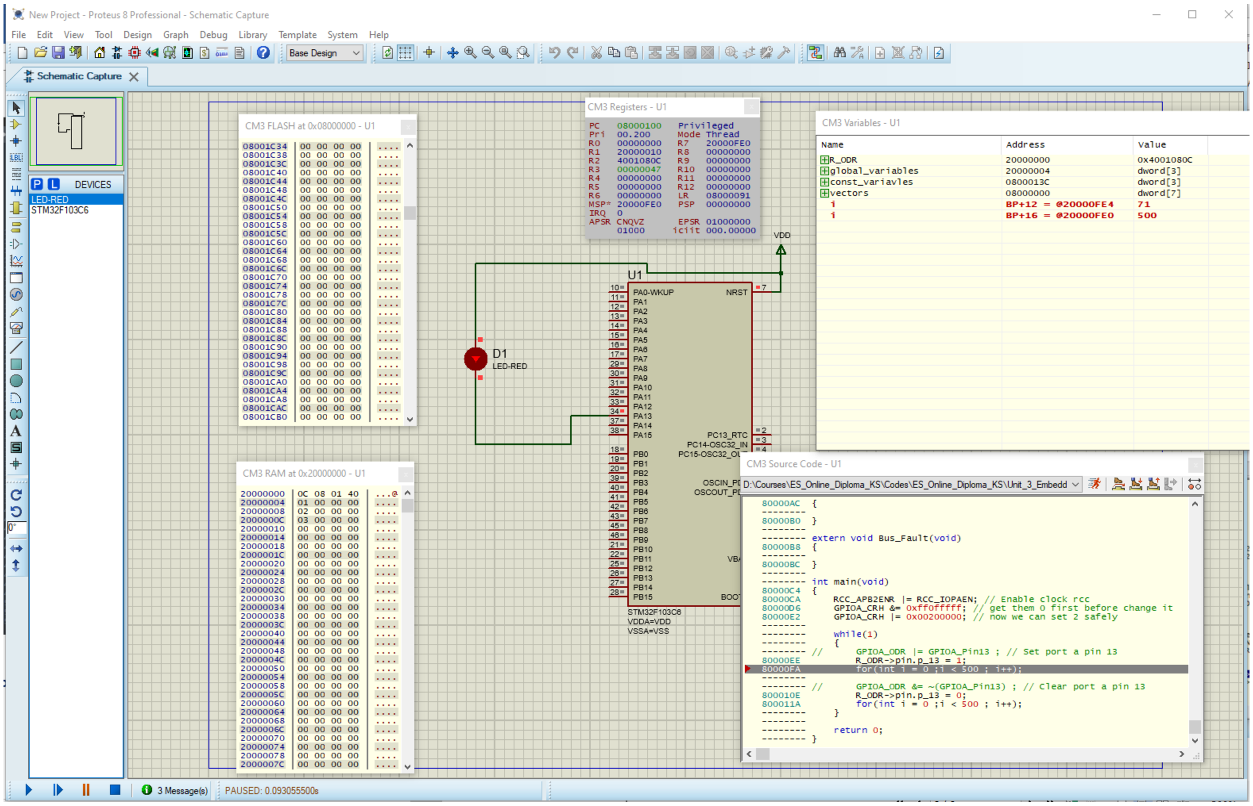
MINGW64:/d/Courses/ES_Online_Diploma_KS/Codes/ES_Online_Diploma_KS/Unit_3_Embedded_C/Lesson 3/Lab_2/Lab_2(startup.c)

```
rm *.o *elf *.bin
============== Everything Clean =============

Mina Karam@DESKTOP-OLOBGHQ MINGW64 /d/Courses/ES_Online_Diploma_KS/Codes/ES_Online_Diploma_KS/Unit_3_Embedded_C/Lesson 3/Lab_2/Lab_2(start
up.c) (master)
$ make clean_all
rm *.o *elf *.bin
rm: cannot remove '*.o': No such file or directory
rm: cannot remove '*elf': No such file or directory
rm: cannot remove '*.bin': No such file or directory
make: *** [clean_all] Error 1

Mina Karam@DESKTOP-OLOBGHQ MINGW64 /d/Courses/ES_Online_Diploma_KS/Codes/ES_Online_Diploma_KS/Unit_3_Embedded_C/Lesson 3/Lab_2/Lab_2(start
up.c) (master)
$ make
arm-none-eabi-gcc.exe -mcpu=cortex-m3 -gdwarf-2  -c CortexM3_startup.c -o CortexM3_startup.o
arm-none-eabi-gcc.exe -mcpu=cortex-m3 -gdwarf-2  -c main.c -o main.o
arm-none-eabi-ld.exe -T CortexM3_linker_script.ld  CortexM3_startup.o main.o -Map=Map_File.map -o Embedded_C_Lesson_3_Lab_2.elf
arm-none-eabi-objcopy.exe -O binary Embedded_C_Lesson_3_Lab_2.elf Embedded_C_Lesson_3_Lab_2.bin
============== Build is Done =============

Mina Karam@DESKTOP-OLOBGHQ MINGW64 /d/Courses/ES_Online_Diploma_KS/Codes/ES_Online_Diploma_KS/Unit_3_Embedded_C/Lesson 3/Lab_2/Lab_2(start
up.c) (master)
$
```


D:\Courses\ES_Online_Diploma_KS\Codes\ES_Online_Diploma_KS\Unit_3_Embedded_C\Lesson 3\Lab_2\Lab_2(startup.c)\Map_File.map - Sublime Te...
File
Edit
Selection
Find
View
Goto
Tools
Project
Preferences
Help
CortexM3_linker_script.ld
CortexM3_startup.c
Map_File.map
log.txt
main.c
10 characters selected
master 23
Spaces: 4
Plain Text

```

Memory Configuration

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

Linker script and memory map


.text           0x08000000      0x148
 *(.vectors*)
 .vectors       0x08000000       0x1c CortexM3_startup.o
                0x08000000                vectors
 *(.text*)
 .text          0x0800001c       0x90 CortexM3_startup.o
                0x0800001c                H_Fault_Handler
                0x0800001c                Default_Hundler
                0x0800001c                MM_Fault_Handler
                0x0800001c                Usage_Fault_Handler
                0x08000028                Reset_Hundler
 .text          0x080000ac       0x90 main.o
                0x080000ac                NMI_Handler
                0x080000b8                Bus_Fault
                0x080000c4                main
 *(.rodata*)
 .rodata        0x0800013c        0xc main.o
                0x0800013c                const_variavles
                0x08000148                _E_TEXT = .

.glue_7         0x08000148        0x0
 .glue_7        0x08000148        0x0 linker stubs

.glue_7t        0x08000148        0x0
 .glue_7t       0x08000148        0x0 linker stubs

.vfp11_veneer   0x08000148        0x0
 .vfp11_veneer  0x08000148        0x0 linker stubs

.v4_bx          0x08000148        0x0
 .v4_bx         0x08000148        0x0 linker stubs

.iplt           0x08000148        0x0
 .iplt          0x08000148        0x0 CortexM3_startup.o

.rel.dyn        0x08000148        0x0
 .rel.iplt      0x08000148        0x0 CortexM3_startup.o

.data           0x20000000       0x10 load address 0x08000148
                0x20000000                _S_DATA = .
 *(.data*)
```

Now using -gdwarf-2 to debug on proteus.



PAUSED: 0.093055500s

CM3 FLASH at 0x08000000 - U1

```
08001C34 | 00 00 00 00 | ....
08001C38 | 00 00 00 00 | ....
08001C3C | 00 00 00 00 | ....
08001C40 | 00 00 00 00 | ....
08001C44 | 00 00 00 00 | ....
08001C48 | 00 00 00 00 | ....
08001C4C | 00 00 00 00 | ....
08001C50 | 00 00 00 00 | ....
08001C54 | 00 00 00 00 | ....
08001C58 | 00 00 00 00 | ....
08001C5C | 00 00 00 00 | ....
08001C60 | 00 00 00 00 | ....
08001C64 | 00 00 00 00 | ....
08001C68 | 00 00 00 00 | ....
08001C6C | 00 00 00 00 | ....
08001C70 | 00 00 00 00 | ....
08001C74 | 00 00 00 00 | ....
08001C78 | 00 00 00 00 | ....
08001C7C | 00 00 00 00 | ....
08001C80 | 00 00 00 00 | ....
08001C84 | 00 00 00 00 | ....
08001C88 | 00 00 00 00 | ....
08001C8C | 00 00 00 00 | ....
08001C90 | 00 00 00 00 | ....
08001C94 | 00 00 00 00 | ....
08001C98 | 00 00 00 00 | ....
08001C9C | 00 00 00 00 | ....
08001CA0 | 00 00 00 00 | ....
08001CA4 | 00 00 00 00 | ....
08001CA8 | 00 00 00 00 | ....
08001CAC | 00 00 00 00 | ....
08001CB0 | 00 00 00 00 | ....
```

CM3 RAM at 0x20000000 - U1

```
20000000 | 0C 08 01 40 | ...@
20000004 | 01 00 00 00 | ....
20000008 | 02 00 00 00 | ....
2000000C | 03 00 00 00 | ....
20000010 | 00 00 00 00 | ....
20000014 | 00 00 00 00 | ....
20000018 | 00 00 00 00 | ....
2000001C | 00 00 00 00 | ....
20000020 | 00 00 00 00 | ....
20000024 | 00 00 00 00 | ....
20000028 | 00 00 00 00 | ....
2000002C | 00 00 00 00 | ....
20000030 | 00 00 00 00 | ....
20000034 | 00 00 00 00 | ....
20000038 | 00 00 00 00 | ....
2000003C | 00 00 00 00 | ....
20000040 | 00 00 00 00 | ....
20000044 | 00 00 00 00 | ....
20000048 | 00 00 00 00 | ....
2000004C | 00 00 00 00 | ....
20000050 | 00 00 00 00 | ....
20000054 | 00 00 00 00 | ....
20000058 | 00 00 00 00 | ....
2000005C | 00 00 00 00 | ....
20000060 | 00 00 00 00 | ....
20000064 | 00 00 00 00 | ....
20000068 | 00 00 00 00 | ....
2000006C | 00 00 00 00 | ....
20000070 | 00 00 00 00 | ....
20000074 | 00 00 00 00 | ....
20000078 | 00 00 00 00 | ....
2000007C | 00 00 00 00 | ....
```

CM3 Registers - U1

```
PC    08000100    Privileged
Pri   00.200      Mode Thread
R0    00000000    R7   20000FE0
R1    20000010    R8   00000000
R2    4001080C    R9   00000000
R3    00000047    R10  00000000
R4    00000000    R11  00000000
R5    00000000    R12  00000000
R6    00000000    LR   08000091
MSP*  20000FE0    PSP  00000000
IRQ   0
APSR  CNQVZ       EPSR 01000000
      01000      iciit 000.00000
```

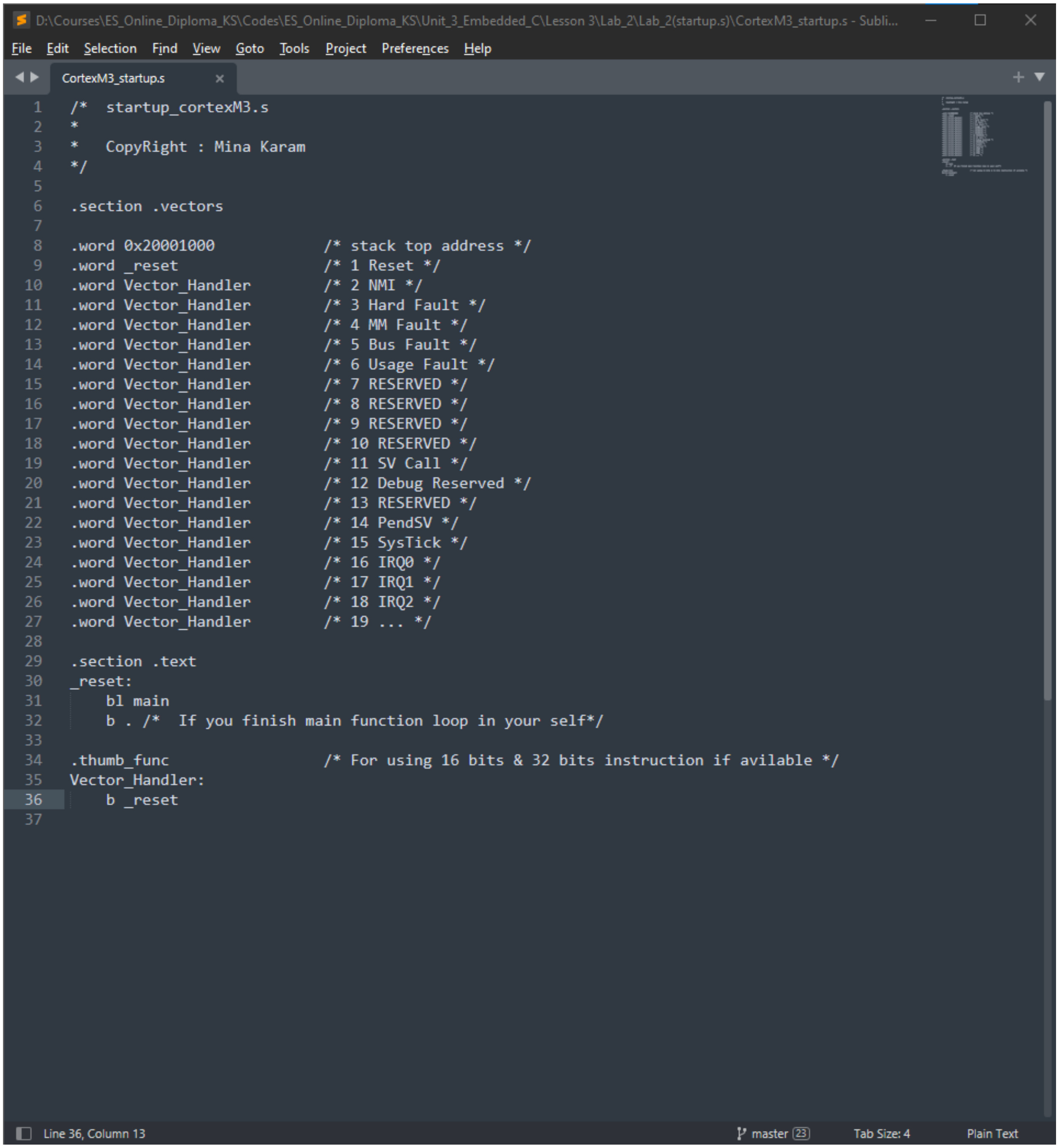
CM3 Variables - U1

Name	Address	Value
⊞R_ODR	20000000	0x4001080C
⊞global_variables	20000004	dword[3]
⊞const_variavles	0800013C	dword[3]
⊞vectors	08000000	dword[7]
i	BP+12 = @20000FE4	71
i	BP+16 = @20000FE0	500

CM3 Source Code - U1

D:\Courses\ES_Online_Diploma_KS\Codes\ES_Online_Diploma_KS\Unit_3_Embedd

```
80000AC  {
-------- 
80000B0  }
--------
-------- extern void Bus_Fault(void)
80000B8  {
--------
80000BC  }
--------
-------- int main(void)
80000C4  {
80000CA      RCC_APB2ENR |= RCC_IOPAEN; // Enable clock rcc
80000D6      GPIOA_CRH &= 0xff0fffff; // get them 0 first before change it
80000E2      GPIOA_CRH |= 0x00200000; // now we can set 2 safely
--------
--------     while(1)
--------     {
-------- //        GPIOA_ODR |= GPIOA_Pin13 ; // Set port a pin 13
80000EE          R_ODR->pin.p_13 = 1;
80000FA          for(int i = 0 ;i < 500 ; i++);
--------
-------- //        GPIOA_ODR &= ~(GPIOA_Pin13) ; // Clear port a pin 13
800010E          R_ODR->pin.p_13 = 0;
800011A          for(int i = 0 ;i < 500 ; i++);
--------     }
--------
--------     return 0;
-------- }
```


D:\Courses\ES_Online_Diploma_KS\Codes\ES_Online_Diploma_KS\Unit_3_Embedded_C\Lesson 3\Lab_2\Lab_2(startup.s)\CortexM3_startup.s - Subli...
File Edit Selection Find View Goto Tools Project Preferences Help
CortexM3_startup.s
Line 36, Column 13
master 23
Tab Size: 4
Plain Text

```
/*  startup_cortexM3.s
*
*   CopyRight : Mina Karam
*/

.section .vectors

.word 0x20001000            /* stack top address */
.word _reset                /* 1 Reset */
.word Vector_Handler        /* 2 NMI */
.word Vector_Handler        /* 3 Hard Fault */
.word Vector_Handler        /* 4 MM Fault */
.word Vector_Handler        /* 5 Bus Fault */
.word Vector_Handler        /* 6 Usage Fault */
.word Vector_Handler        /* 7 RESERVED */
.word Vector_Handler        /* 8 RESERVED */
.word Vector_Handler        /* 9 RESERVED */
.word Vector_Handler        /* 10 RESERVED */
.word Vector_Handler        /* 11 SV Call */
.word Vector_Handler        /* 12 Debug Reserved */
.word Vector_Handler        /* 13 RESERVED */
.word Vector_Handler        /* 14 PendSV */
.word Vector_Handler        /* 15 SysTick */
.word Vector_Handler        /* 16 IRQ0 */
.word Vector_Handler        /* 17 IRQ1 */
.word Vector_Handler        /* 18 IRQ2 */
.word Vector_Handler        /* 19 ... */

.section .text
_reset:
    bl main
    b . /*  If you finish main function loop in your self*/

.thumb_func                 /* For using 16 bits & 32 bits instruction if avilable */
Vector_Handler:
    b _reset

```