LAB 2

Using STM32F103C8T6 Chip with ARM Cortex-M3 32-Bit Microcontroller

1. Codes :

A) Main.c

```c

#define RCC   0x40021000
#define PortA 0x40010800
#define PortA_CRH   *((volatile unsigned int*)(PortA+0x04))
#define PortA_ODR   *((volatile unsigned int*)(PortA+0x0C))
#define RCC_APB2ENR *((volatile unsigned int*)(RCC+0x18))
/** another way to access ODR **/
typedef union{
    struct{
        volatile unsigned int Reserved:13;
        volatile unsigned int Pin13:1;
    }SPIN;
    volatile unsigned int All_Pins;

}UODR_t;
volatile UODR_t* R_ODR = (volatile UODR_t*)(PortA + 0x0C);

int main(void)
{
    RCC_APB2ENR |= (1<<2);
    PortA_CRH &= 0xFF0FFFFF;
    PortA_CRH |= 0x00200000;
    int i;
    while(1)
    {
        PortA_ODR |= (1<<13);
        for(i=0 ; i<50000 ; i++);
        PortA_ODR &= ~(1<<13);
        for(i=0 ; i<5000 ; i++);
    }

}

```

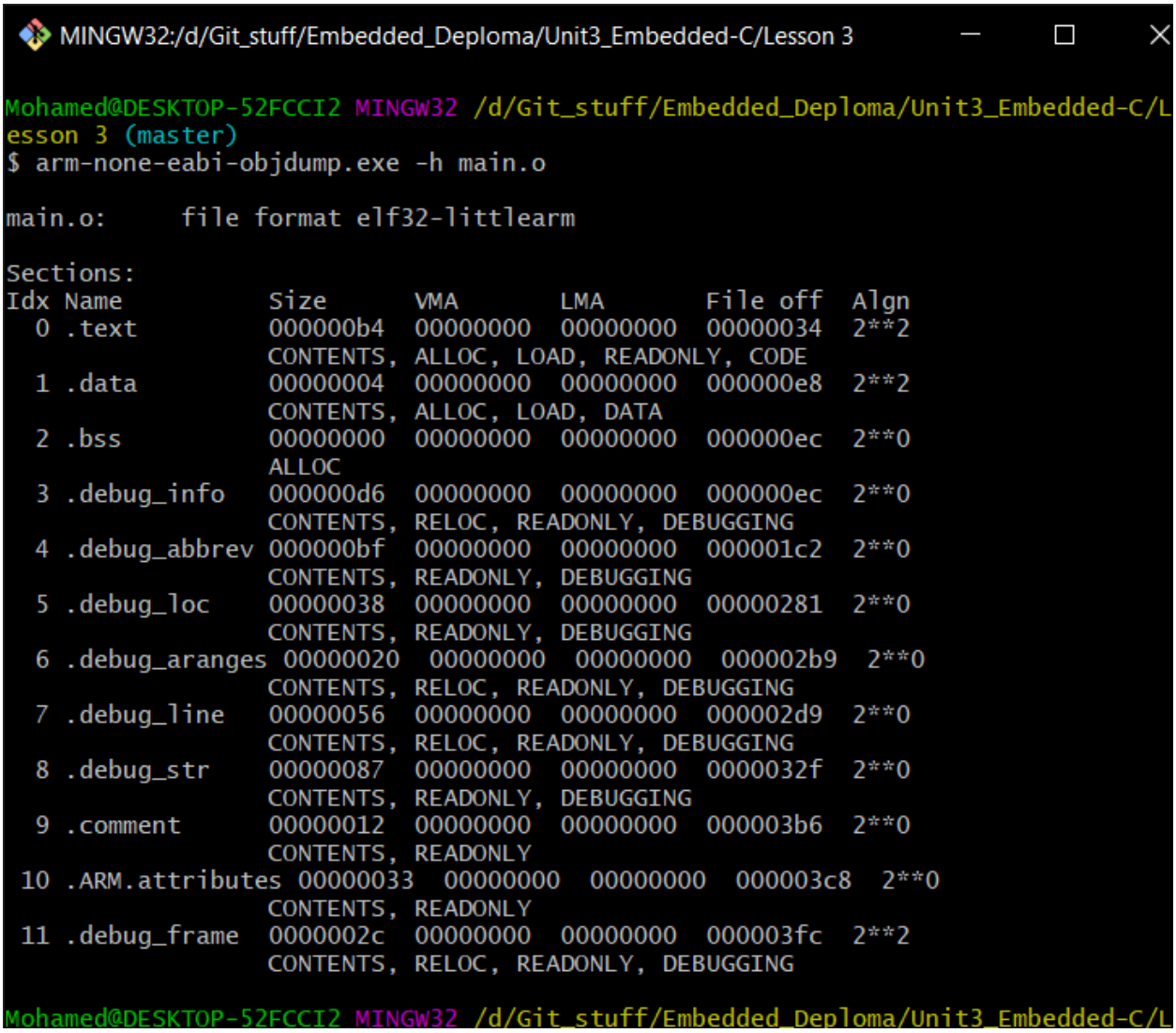
B) Startup.c

```c
extern unsigned int _STACK_TOP;
extern int main();
void default_handeler();
void reset_handeler();
void NMI_handeler() __attribute__ ((weak,alias("default_handeler")));
void HardFault_handeler() __attribute__ ((weak,alias("default_handeler")));
void MM_Fault_handeler() __attribute__ ((weak,alias("default_handeler")));
void Bus_Fault_handeler() __attribute__ ((weak,alias("default_handeler")));
void Usage_Fault_handeler() __attribute__ ((weak,alias("default_handeler")));
unsigned int  vectors [] __attribute__ ((section(".vectors"))) = {
(unsigned int ) &_STACK_TOP ,
(unsigned int ) &reset_handeler,
(unsigned int ) &NMI_handeler,
(unsigned int ) &HardFault_handeler,
(unsigned int ) &MM_Fault_handeler,
(unsigned int ) &Bus_Fault_handeler,
(unsigned int ) &Usage_Fault_handeler
};
extern unsigned int _E_text;
extern unsigned int _S_data;
extern unsigned int _E_data;
extern unsigned int _S_bss;
extern unsigned int _E_bss;
void reset_handeler(){
    /*copying .data from Flash to RAM*/
    unsigned int i;
    unsigned int _data_size = (unsigned char*)&_E_data - (unsigned char*)&_S_data;
    unsigned char *ptr_scr = &_E_text;
    unsigned char *ptr_dest = &_S_data;
    for(i=0 ; i< _data_size ; i++)
    {
        *((unsigned char*)ptr_dest++) = *((unsigned char*)ptr_scr++);
    }
    /*create .bss section*/
    unsigned int _bss_size = (unsigned char*)&_E_bss - (unsigned char*)&_S_bss;
    ptr_dest = &_S_data;
    for(i=0 ; i< _data_size ; i++)
    {
        *((unsigned char*)ptr_dest++) = (unsigned char*)0;
    }

    /*branching to main*/
    main();
}

void default_handeler(){
    reset_handeler();
}
```

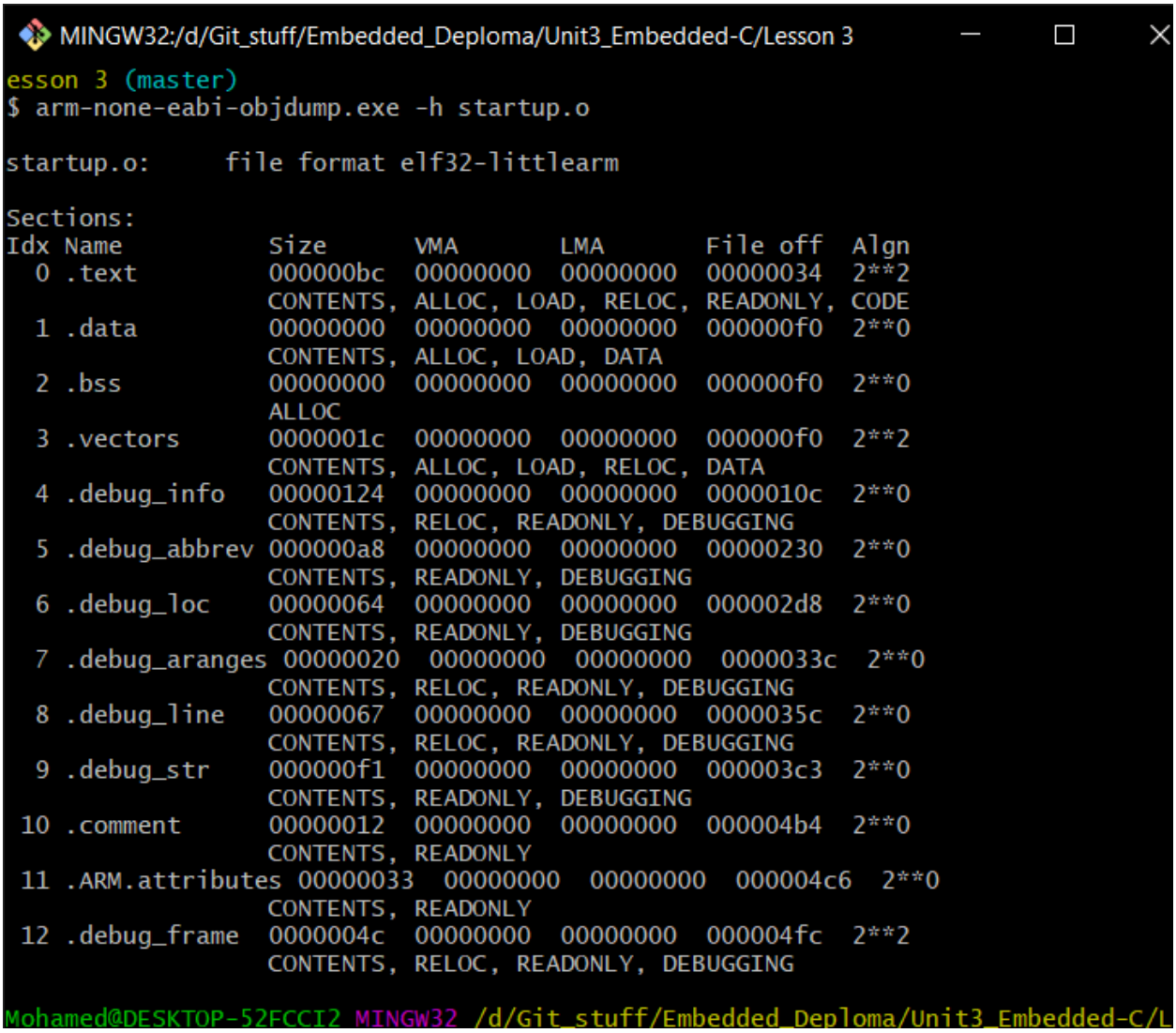
C) Linker_script.ld

```
1
2   MEMORY
3   {
4       Flash (RX) : ORIGIN = 0x08000000 ,LENGTH = 128K
5       SRAM (RWX) : ORIGIN = 0x20000000 ,LENGTH = 20K
6   }
7
8   SECTIONS
9   {
10      .text :
11      {
12          *(.vectors*)
13          *(.text*)
14          *(.rodata*)
15          _E_text = .;
16      }>Flash
17
18      .data :
19      {
20          _S_data = .;
21          *(.data*)
22          . = ALIGN(4);
23          _E_data = .;
24      }>SRAM AT> Flash
25
26      .bss :
27      {
28          _S_bss = .;
29          *(.bss*)
30          _E_bss = .;
31          . = ALIGN(4);
32          . = . + 1000;
33          _STACK_TOP = .;
34      }>SRAM
35  }
```

2. Sections :
   a) Main.o


MINGW32:/d/Git_stuff/Embedded_Deploma/Unit3_Embedded-C/Lesson 3
Mohamed@DESKTOP-52FCCI2 MINGW32 /d/Git_stuff/Embedded_Deploma/Unit3_Embedded-C/Lesson 3 (master)
$ arm-none-eabi-objdump.exe -h main.o
main.o: file format elf32-littlearm
Sections:
Idx Name Size VMA LMA File off Algn
0 .text 000000b4 00000000 00000000 00000034 2**2
CONTENTS, ALLOC, LOAD, READONLY, CODE
1 .data 00000004 00000000 00000000 000000e8 2**2
CONTENTS, ALLOC, LOAD, DATA
2 .bss 00000000 00000000 00000000 000000ec 2**0
ALLOC
3 .debug_info 000000d6 00000000 00000000 000000ec 2**0
CONTENTS, RELOC, READONLY, DEBUGGING
4 .debug_abbrev 000000bf 00000000 00000000 000001c2 2**0
CONTENTS, READONLY, DEBUGGING
5 .debug_loc 00000038 00000000 00000000 00000281 2**0
CONTENTS, READONLY, DEBUGGING
6 .debug_aranges 00000020 00000000 00000000 000002b9 2**0
CONTENTS, RELOC, READONLY, DEBUGGING
7 .debug_line 00000056 00000000 00000000 000002d9 2**0
CONTENTS, RELOC, READONLY, DEBUGGING
8 .debug_str 00000087 00000000 00000000 0000032f 2**0
CONTENTS, READONLY, DEBUGGING
9 .comment 00000012 00000000 00000000 000003b6 2**0
CONTENTS, READONLY
10 .ARM.attributes 00000033 00000000 00000000 000003c8 2**0
CONTENTS, READONLY
11 .debug_frame 0000002c 00000000 00000000 000003fc 2**2
CONTENTS, RELOC, READONLY, DEBUGGING
Mohamed@DESKTOP-52FCCI2 MINGW32 /d/Git_stuff/Embedded_Deploma/Unit3_Embedded-C/Lesson 3

b) Startup.o


MINGW32:/d/Git_stuff/Embedded_Deploma/Unit3_Embedded-C/Lesson 3
esson 3 (master)
$ arm-none-eabi-objdump.exe -h startup.o
startup.o: file format elf32-littlearm
Sections:
Idx Name Size VMA LMA File off Algn
0 .text 000000bc 00000000 00000000 00000034 2**2
CONTENTS, ALLOC, LOAD, RELOC, READONLY, CODE
1 .data 00000000 00000000 00000000 000000f0 2**0
CONTENTS, ALLOC, LOAD, DATA
2 .bss 00000000 00000000 00000000 000000f0 2**0
ALLOC
3 .vectors 0000001c 00000000 00000000 000000f0 2**2
CONTENTS, ALLOC, LOAD, RELOC, DATA
4 .debug_info 00000124 00000000 00000000 0000010c 2**0
CONTENTS, RELOC, READONLY, DEBUGGING
5 .debug_abbrev 000000a8 00000000 00000000 00000230 2**0
CONTENTS, READONLY, DEBUGGING
6 .debug_loc 00000064 00000000 00000000 000002d8 2**0
CONTENTS, READONLY, DEBUGGING
7 .debug_aranges 00000020 00000000 00000000 0000033c 2**0
CONTENTS, RELOC, READONLY, DEBUGGING
8 .debug_line 00000067 00000000 00000000 0000035c 2**0
CONTENTS, RELOC, READONLY, DEBUGGING
9 .debug_str 000000f1 00000000 00000000 000003c3 2**0
CONTENTS, READONLY, DEBUGGING
10 .comment 00000012 00000000 00000000 000004b4 2**0
CONTENTS, READONLY
11 .ARM.attributes 00000033 00000000 00000000 000004c6 2**0
CONTENTS, READONLY
12 .debug_frame 0000004c 00000000 00000000 000004fc 2**2
CONTENTS, RELOC, READONLY, DEBUGGING
Mohamed@DESKTOP-52FCCI2 MINGW32 /d/Git_stuff/Embedded_Deploma/Unit3_Embedded-C/L

c) App.elf

```
Mohamed@DESKTOP-52FCCI2 MINGW32 /d/Git_stuff/Embedded_Deploma/Unit3_Embedded-C/L
esson 3 (master)
$ arm-none-eabi-objdump.exe -h Toggling_LED.elf

Toggling_LED.elf:     file format elf32-littlearm

Sections:
Idx Name          Size      VMA       LMA       File off  Algn
  0 .text         0000018c  08000000  08000000  00008000  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, CODE
  1 .data         00000004  20000000  0800018c  00010000  2**2
                  CONTENTS, ALLOC, LOAD, DATA
  2 .bss          000003e8  20000004  08000190  00010004  2**0
                  ALLOC
  3 .debug_info   000001fa  00000000  00000000  00010004  2**0
                  CONTENTS, READONLY, DEBUGGING
  4 .debug_abbrev 00000167  00000000  00000000  000101fe  2**0
                  CONTENTS, READONLY, DEBUGGING
  5 .debug_loc    0000009c  00000000  00000000  00010365  2**0
                  CONTENTS, READONLY, DEBUGGING
  6 .debug_aranges 00000040  00000000  00000000  00010401  2**0
                  CONTENTS, READONLY, DEBUGGING
  7 .debug_line   000000bd  00000000  00000000  00010441  2**0
                  CONTENTS, READONLY, DEBUGGING
  8 .debug_str    00000127  00000000  00000000  000104fe  2**0
                  CONTENTS, READONLY, DEBUGGING
  9 .comment      00000011  00000000  00000000  00010625  2**0
                  CONTENTS, READONLY
 10 .ARM.attributes 00000033  00000000  00000000  00010636  2**0
                  CONTENTS, READONLY
 11 .debug_frame  00000078  00000000  00000000  0001066c  2**2
                  CONTENTS, READONLY, DEBUGGING
```

3. Symbols :
   a) Main.o

```
Mohamed@DESKTOP-52FCCI2 MINGW32 /d/Git_stuff/Embedded_Deploma/Unit3_Embedded-C/L
esson 3 (master)
$ arm-none-eabi-nm.exe main.o
00000000 T main
00000000 D R_ODR

Mohamed@DESKTOP-52FCCI2 MINGW32 /d/Git_stuff/Embedded_Deploma/Unit3_Embedded-C/L
```

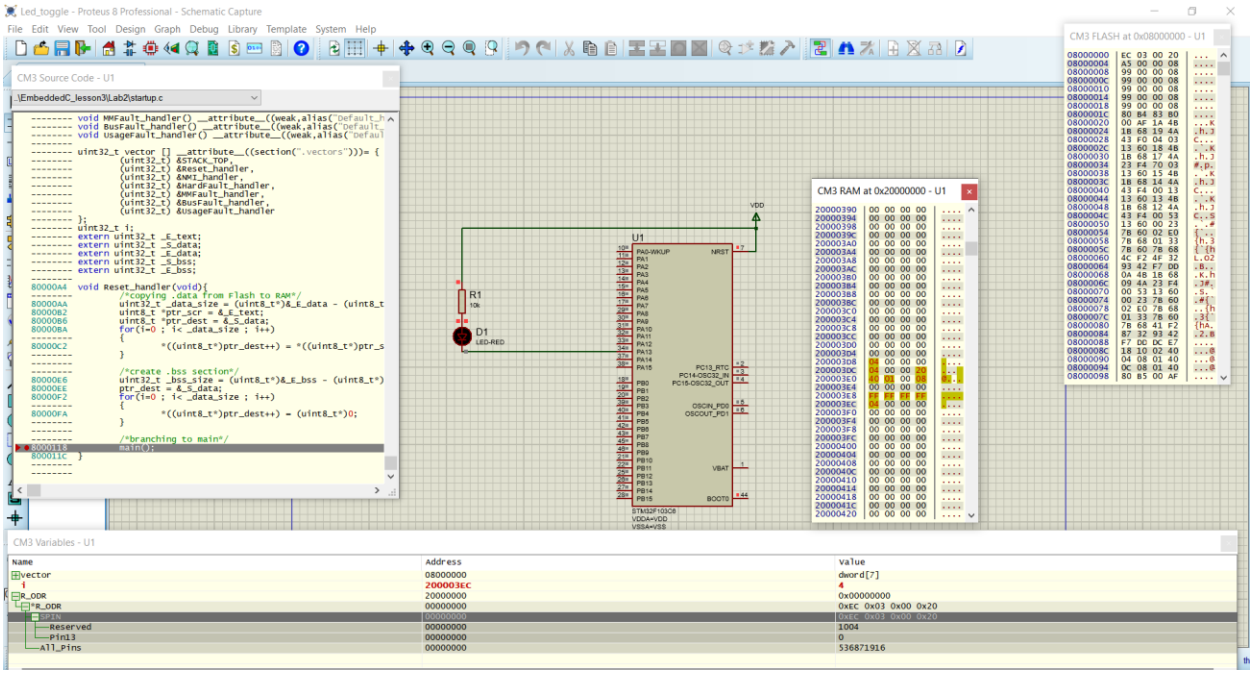
   b) Startup.o

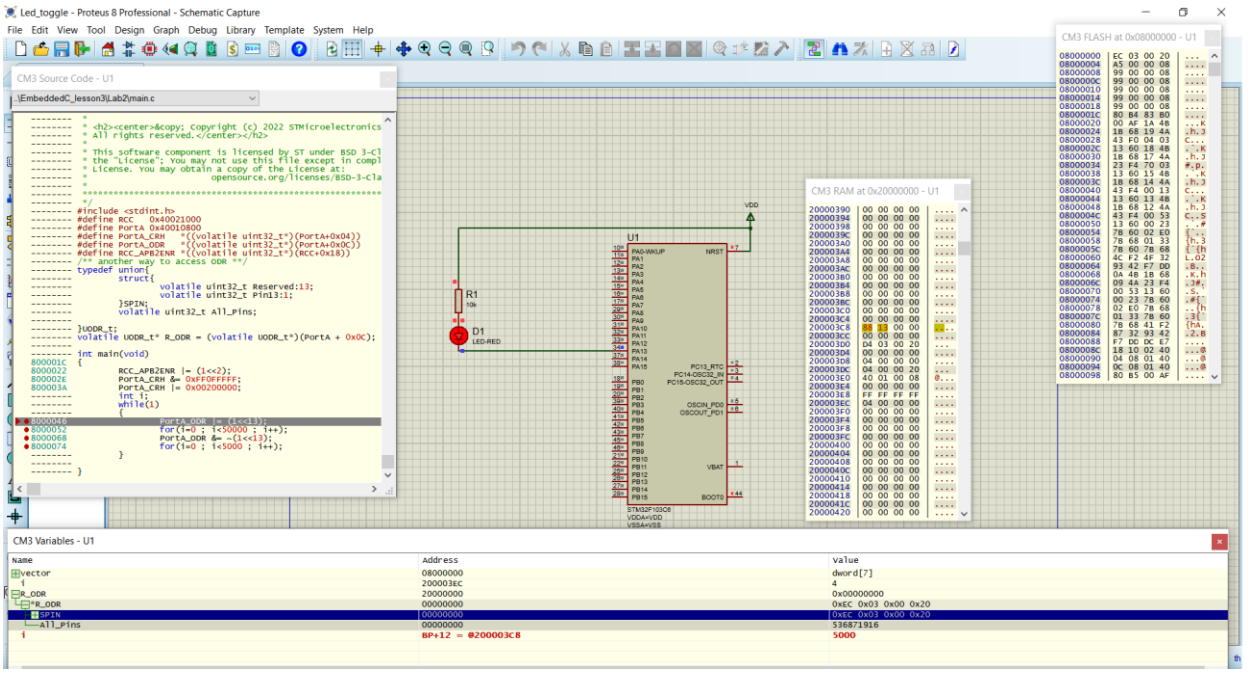
```
Mohamed@DESKTOP-52FCCI2 MINGW32 /d/Git_stuff/Embedded_Deploma/Unit3_Embedded-C/L
esson 3 (master)
$ arm-none-eabi-nm.exe startup.o
         U _E_bss
         U _E_data
         U _E_text
         U _S_bss
         U _S_data
         U _STACK_TOP
000000b0 W Bus_Fault_handeler
000000b0 T default_handeler
000000b0 W HardFault_handeler
         U main
000000b0 W MM_Fault_handeler
000000b0 W NMI_handeler
00000000 T reset_handeler
000000b0 W Usage_Fault_handeler
00000000 D vectors

Mohamed@DESKTOP-52FCCI2 MINGW32 /d/Git_stuff/Embedded_Deploma/Unit3_Embedded-C/L
```

c) App.elf

```
Mohamed@DESKTOP-52FCCI2 MINGW32 /d/Git_stuff/Embedded_Deploma/Unit3_Embedded-C/L
esson 3 (master)
$ arm-none-eabi-nm.exe Toggling_LED.elf
20000004 B _E_bss
20000004 D _E_data
0800018c T _E_text
20000004 B _S_bss
20000000 D _S_data
200003ec B _STACK_TOP
08000180 W Bus_Fault_handeler
08000180 T default_handeler
08000180 W HardFault_handeler
0800001c T main
08000180 W MM_Fault_handeler
08000180 W NMI_handeler
20000000 D R_ODR
080000d0 T reset_handeler
08000180 W Usage_Fault_handeler
08000000 T vectors

Mohamed@DESKTOP-52FCCI2 MINGW32 /d/Git_stuff/Embedded_Deploma/Unit3_Embedded-C/L
```

4. Simulation and debugging :


Led_toggle - Proteus 8 Professional - Schematic Capture
File Edit View Tool Design Graph Debug Library Template System Help
CM3 Source Code - U1
\EmbeddedC_lesson3\Lab2\startup.c
CM3 FLASH at 0x08000000 - U1
CM3 RAM at 0x20000000 - U1
CM3 Variables - U1
Name
Address
Value
vector
08000000
dword[7]
i
200003EC
4
R_ODR
20000000
0x00000000
PR_ODR
00000000
0xEC 0x03 0x00 0x20
SPIN
Reserved
00000000
1004
Pin13
00000000
0
All_Pins
00000000
536871916


Led_toggle - Proteus 8 Professional - Schematic Capture
File Edit View Tool Design Graph Debug Library Template System Help
CM3 Source Code - U1
\EmbeddedC_lesson3\Lab2\main.c
CM3 FLASH at 0x08000000 - U1
CM3 RAM at 0x20000000 - U1
CM3 Variables - U1
Name
Address
Value
vector
08000000
dword[7]
i
200003EC
4
R_ODR
20000000
0x00000000
PR_ODR
00000000
0xEC 0x03 0x00 0x20
SPIN
00000000
0xEC 0x03 0x00 0x20
All_Pins
00000000
536871916
i
BP+12 = @200003C8
5000