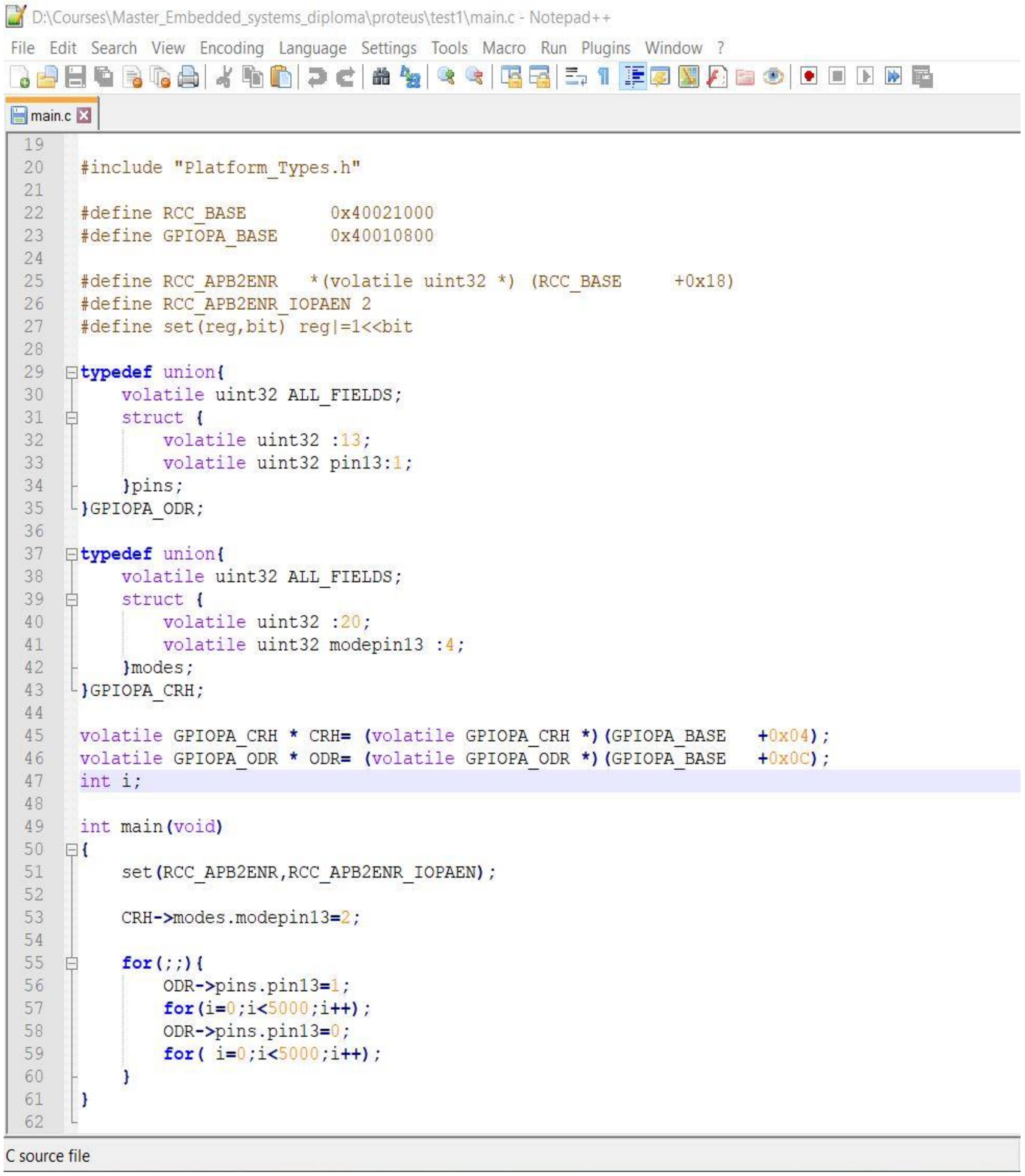
# LAB3:

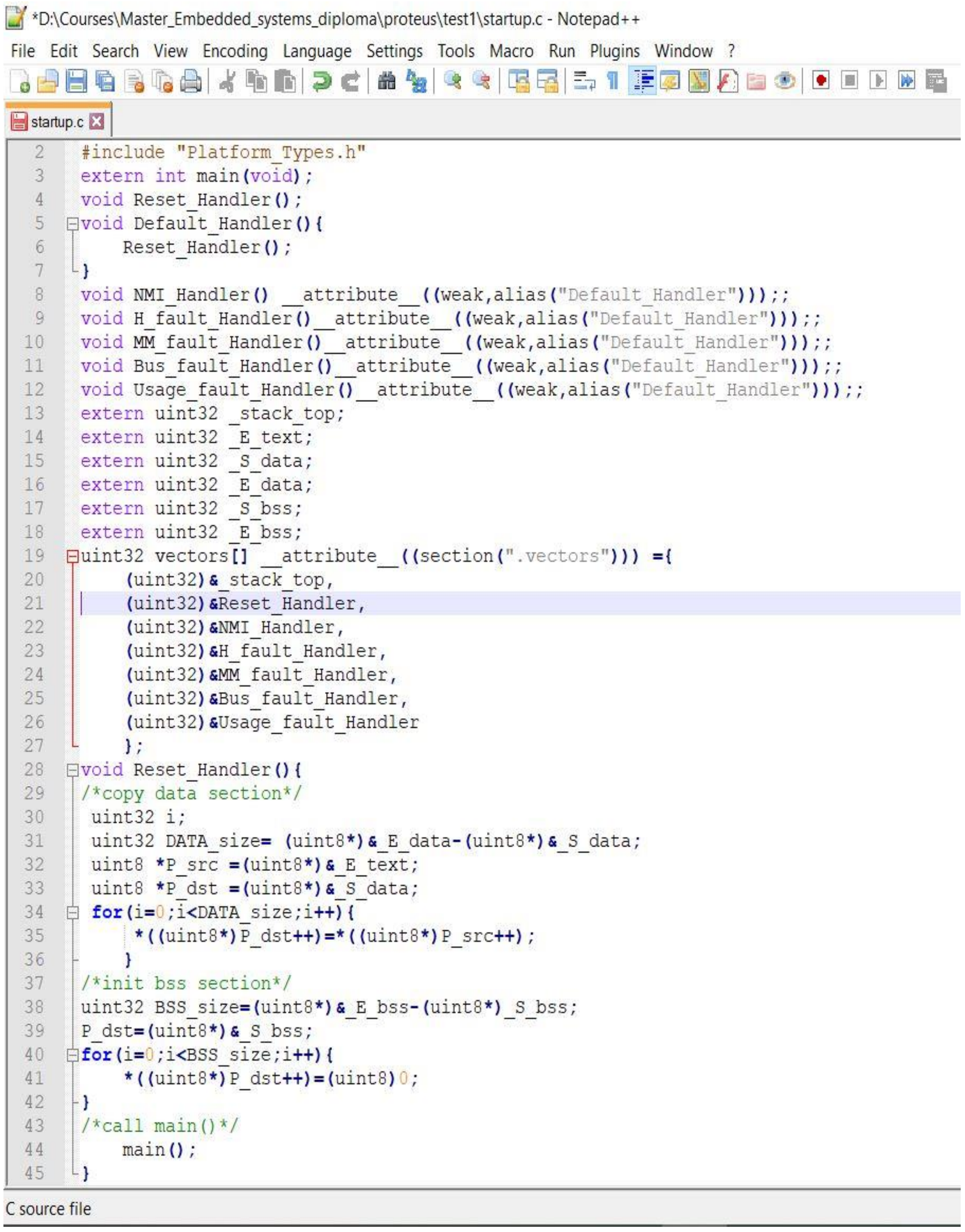
- First writing the (main.c and startup.c) code files:-

  -main.c


D:\Courses\Master_Embedded_systems_diploma\proteus\test1\main.c - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
main.c
C source file

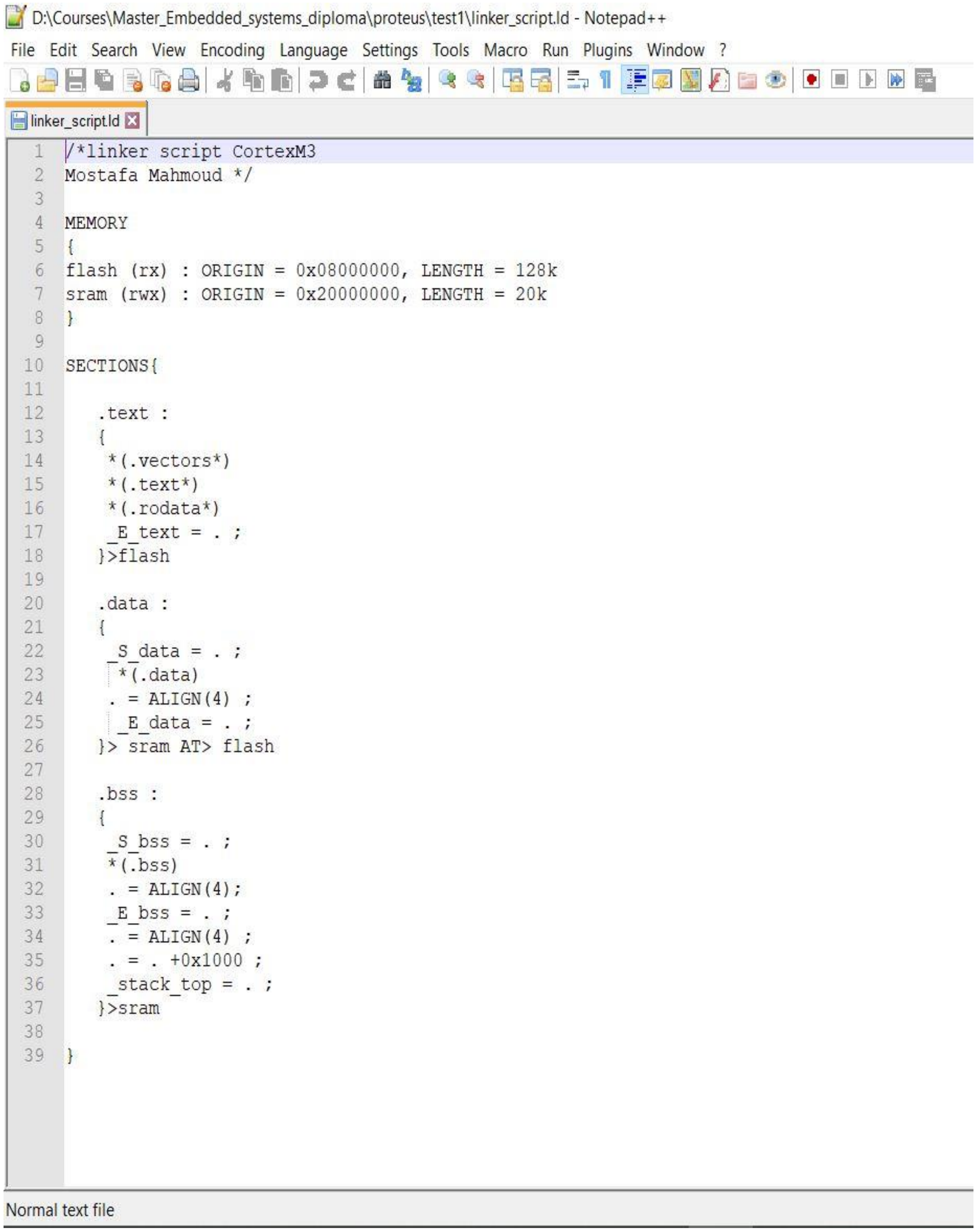
```c

#include "Platform_Types.h"

#define RCC_BASE        0x40021000
#define GPIOPA_BASE     0x40010800

#define RCC_APB2ENR   *(volatile uint32 *) (RCC_BASE     +0x18)
#define RCC_APB2ENR_IOPAEN 2
#define set(reg,bit) reg|=1<<bit

typedef union{
    volatile uint32 ALL_FIELDS;
    struct {
        volatile uint32 :13;
        volatile uint32 pin13:1;
    }pins;
}GPIOPA_ODR;

typedef union{
    volatile uint32 ALL_FIELDS;
    struct {
        volatile uint32 :20;
        volatile uint32 modepin13 :4;
    }modes;
}GPIOPA_CRH;

volatile GPIOPA_CRH * CRH= (volatile GPIOPA_CRH *)(GPIOPA_BASE   +0x04);
volatile GPIOPA_ODR * ODR= (volatile GPIOPA_ODR *)(GPIOPA_BASE   +0x0C);
int i;

int main(void)
{
    set(RCC_APB2ENR,RCC_APB2ENR_IOPAEN);

    CRH->modes.modepin13=2;

    for(;;){
        ODR->pins.pin13=1;
        for(i=0;i<5000;i++);
        ODR->pins.pin13=0;
        for( i=0;i<5000;i++);
    }
}

```

-startup.c:


*D:\Courses\Master_Embedded_systems_diploma\proteus\test1\startup.c - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
startup.c
C source file

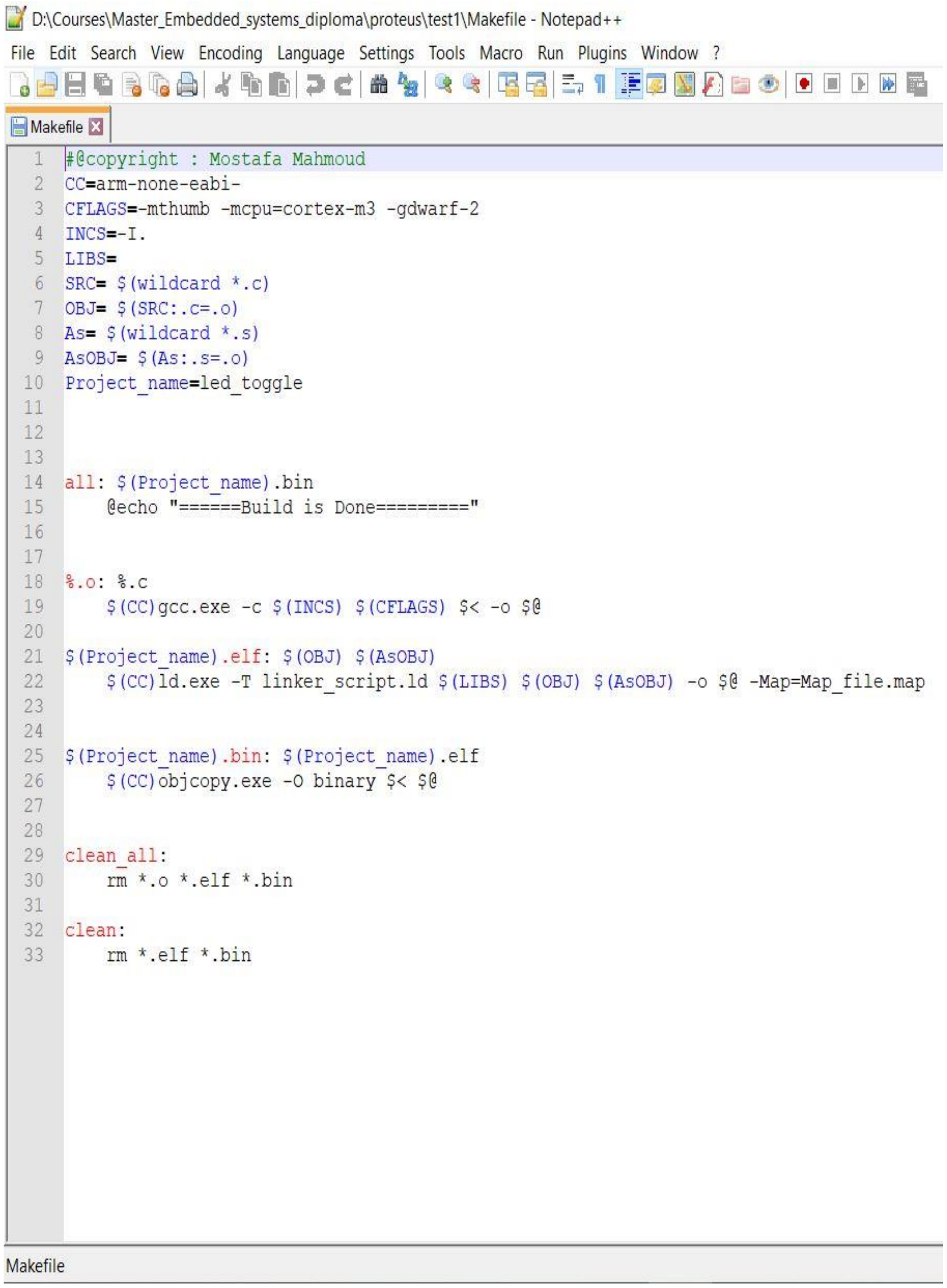
```c
#include "Platform_Types.h"
extern int main(void);
void Reset_Handler();
void Default_Handler(){
    Reset_Handler();
}
void NMI_Handler()  __attribute__((weak,alias("Default_Handler")));;
void H_fault_Handler()__attribute__((weak,alias("Default_Handler")));;
void MM_fault_Handler()__attribute__((weak,alias("Default_Handler")));;
void Bus_fault_Handler()__attribute__((weak,alias("Default_Handler")));;
void Usage_fault_Handler()__attribute__((weak,alias("Default_Handler")));;
extern uint32 _stack_top;
extern uint32 _E_text;
extern uint32 _S_data;
extern uint32 _E_data;
extern uint32 _S_bss;
extern uint32 _E_bss;
uint32 vectors[] __attribute__((section(".vectors"))) ={
    (uint32)&_stack_top,
    (uint32)&Reset_Handler,
    (uint32)&NMI_Handler,
    (uint32)&H_fault_Handler,
    (uint32)&MM_fault_Handler,
    (uint32)&Bus_fault_Handler,
    (uint32)&Usage_fault_Handler
    };
void Reset_Handler(){
/*copy data section*/
 uint32 i;
 uint32 DATA_size= (uint8*)&_E_data-(uint8*)&_S_data;
 uint8 *P_src =(uint8*)&_E_text;
 uint8 *P_dst =(uint8*)&_S_data;
 for(i=0;i<DATA_size;i++){
     *((uint8*)P_dst++)=*((uint8*)P_src++);
    }
/*init bss section*/
uint32 BSS_size=(uint8*)&_E_bss-(uint8*)_S_bss;
P_dst=(uint8*)&_S_bss;
for(i=0;i<BSS_size;i++){
    *((uint8*)P_dst++)=(uint8)0;
}
/*call main()*/
    main();
}
```

- Then writing linker script:
  linker_script.ld:


D:\Courses\Master_Embedded_systems_diploma\proteus\test1\linker_script.ld - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
linker_script.ld
Normal text file

```
/*linker script CortexM3
Mostafa Mahmoud */

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

}
```

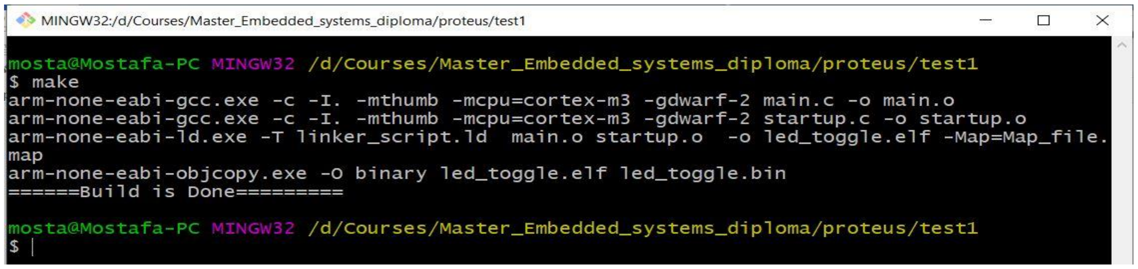
- Now Adjusting Make file to build the project:
  Makefile:



D:\Courses\Master_Embedded_systems_diploma\proteus\test1\Makefile - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
Makefile

```
#@copyright : Mostafa Mahmoud
CC=arm-none-eabi-
CFLAGS=-mthumb -mcpu=cortex-m3 -gdwarf-2
INCS=-I.
LIBS=
SRC= $(wildcard *.c)
OBJ= $(SRC:.c=.o)
As= $(wildcard *.s)
AsOBJ= $(As:.s=.o)
Project_name=led_toggle



all: $(Project_name).bin
	@echo "======Build is Done========="


%.o: %.c
	$(CC)gcc.exe -c $(INCS) $(CFLAGS) $< -o $@

$(Project_name).elf: $(OBJ) $(AsOBJ)
	$(CC)ld.exe -T linker_script.ld $(LIBS) $(OBJ) $(AsOBJ) -o $@ -Map=Map_file.map


$(Project_name).bin: $(Project_name).elf
	$(CC)objcopy.exe -O binary $< $@


clean_all:
	rm *.o *.elf *.bin

clean:
	rm *.elf *.bin
```

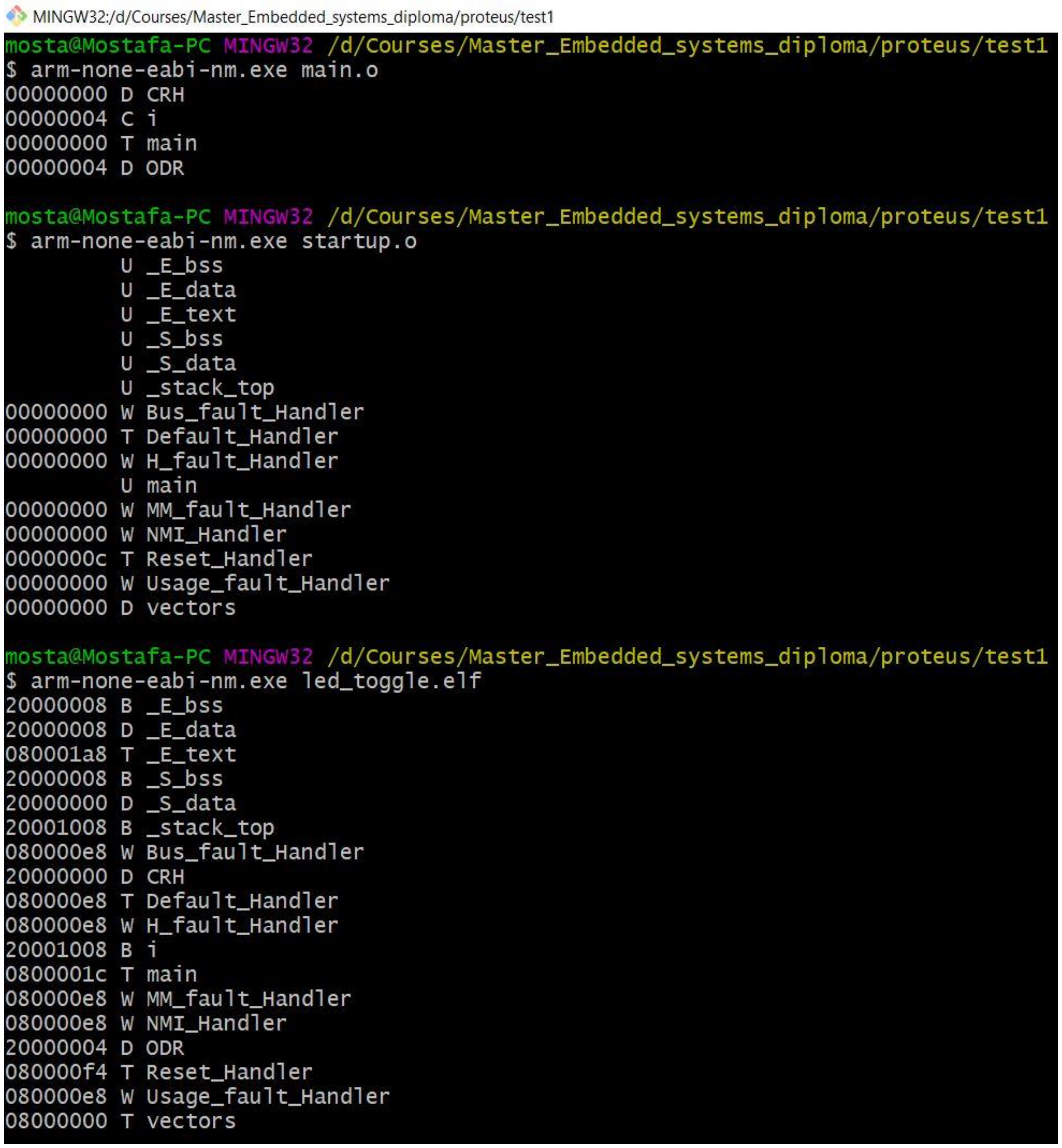
- Building the project:



MINGW32:/d/Courses/Master_Embedded_systems_diploma/proteus/test1

```
mosta@Mostafa-PC MINGW32 /d/Courses/Master_Embedded_systems_diploma/proteus/test1
$ make
arm-none-eabi-gcc.exe -c -I. -mthumb -mcpu=cortex-m3 -gdwarf-2 main.c -o main.o
arm-none-eabi-gcc.exe -c -I. -mthumb -mcpu=cortex-m3 -gdwarf-2 startup.c -o startup.o
arm-none-eabi-ld.exe -T linker_script.ld  main.o startup.o  -o led_toggle.elf -Map=Map_file.
map
arm-none-eabi-objcopy.exe -O binary led_toggle.elf led_toggle.bin
======Build is Done=========

mosta@Mostafa-PC MINGW32 /d/Courses/Master_Embedded_systems_diploma/proteus/test1
$ |
```

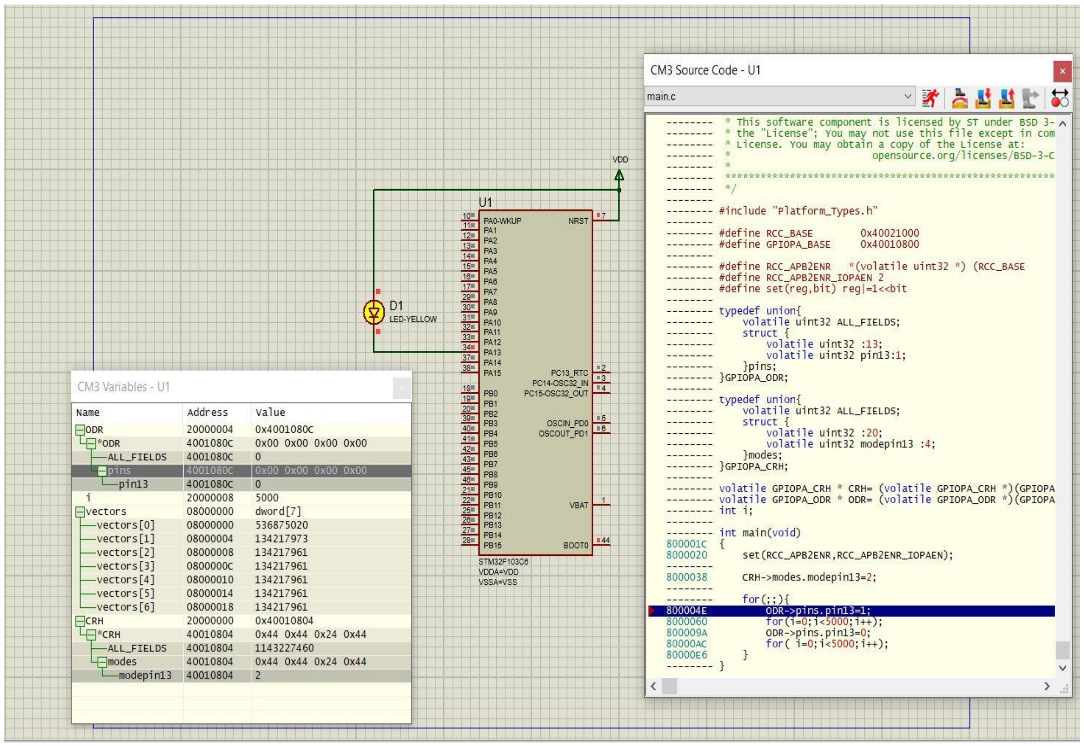
- Viewing symbols in main.o, startup.o and led_toggle.elf:



MINGW32:/d/Courses/Master_Embedded_systems_diploma/proteus/test1

```
mosta@Mostafa-PC MINGW32 /d/Courses/Master_Embedded_systems_diploma/proteus/test1
$ arm-none-eabi-nm.exe main.o
00000000 D CRH
00000004 C i
00000000 T main
00000004 D ODR

mosta@Mostafa-PC MINGW32 /d/Courses/Master_Embedded_systems_diploma/proteus/test1
$ 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
00000000 W Bus_fault_Handler
00000000 T Default_Handler
00000000 W H_fault_Handler
         U main
00000000 W MM_fault_Handler
00000000 W NMI_Handler
0000000c T Reset_Handler
00000000 W Usage_fault_Handler
00000000 D vectors

mosta@Mostafa-PC MINGW32 /d/Courses/Master_Embedded_systems_diploma/proteus/test1
$ arm-none-eabi-nm.exe led_toggle.elf
20000008 B _E_bss
20000008 D _E_data
080001a8 T _E_text
20000008 B _S_bss
20000000 D _S_data
20001008 B _stack_top
080000e8 W Bus_fault_Handler
20000000 D CRH
080000e8 T Default_Handler
080000e8 W H_fault_Handler
20001008 B i
0800001c T main
080000e8 W MM_fault_Handler
080000e8 W NMI_Handler
20000004 D ODR
080000f4 T Reset_Handler
080000e8 W Usage_fault_Handler
08000000 T vectors
```

- Running simulation and debugging on proteus:



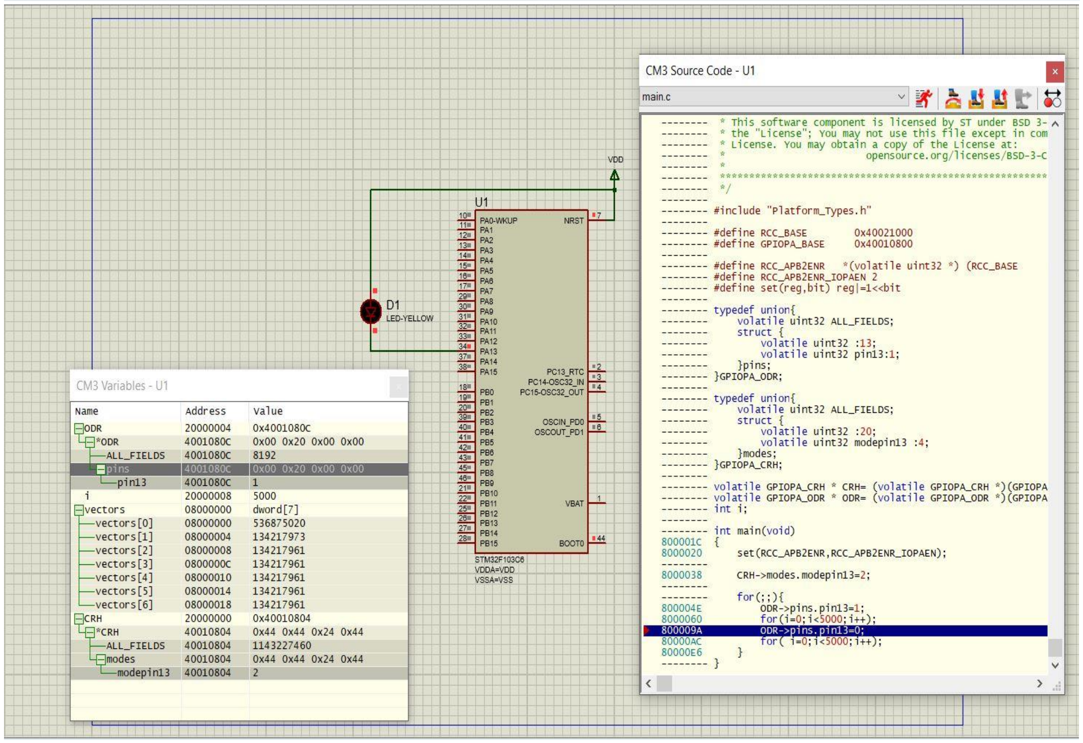
CM3 Source Code - U1

main.c

```
--------  * This software component is licensed by ST under BSD 3-
--------  * the "License"; You may not use this file except in com
--------  * License. You may obtain a copy of the License at:
--------  *                        opensource.org/licenses/BSD-3-C
--------  *
--------  ******************************************************************
--------  */
--------
-------- #include "Platform_Types.h"
--------
-------- #define RCC_BASE        0x40021000
-------- #define GPIOPA_BASE     0x40010800
--------
-------- #define RCC_APB2ENR   *(volatile uint32 *) (RCC_BASE
-------- #define RCC_APB2ENR_IOPAEN 2
-------- #define set(reg,bit) reg|=1<<bit
--------
-------- typedef union{
--------     volatile uint32 ALL_FIELDS;
--------     struct {
--------         volatile uint32 :13;
--------         volatile uint32 pin13:1;
--------     }pins;
-------- }GPIOPA_ODR;
--------
-------- typedef union{
--------     volatile uint32 ALL_FIELDS;
--------     struct {
--------         volatile uint32 :20;
--------         volatile uint32 modepin13 :4;
--------     }modes;
-------- }GPIOPA_CRH;
--------
-------- volatile GPIOPA_CRH * CRH= (volatile GPIOPA_CRH *)(GPIOPA
-------- volatile GPIOPA_ODR * ODR= (volatile GPIOPA_ODR *)(GPIOPA
-------- int i;
--------
-------- int main(void)
800001C  {
8000020      set(RCC_APB2ENR,RCC_APB2ENR_IOPAEN);
--------
8000038      CRH->modes.modepin13=2;
--------
--------     for(;;){
800004E          ODR->pins.pin13=1;
8000060          for(i=0;i<5000;i++);
800009A          ODR->pins.pin13=0;
80000AC          for( i=0;i<5000;i++);
80000E6      }
-------- }
```

CM3 Variables - U1

| Name | Address | Value |
|---|---|---|
| ODR | 20000004 | 0x4001080C |
| *ODR | 4001080C | 0x00 0x00 0x00 0x00 |
| ALL_FIELDS | 4001080C | 0 |
| pins | 4001080C | 0x00 0x00 0x00 0x00 |
| pin13 | 4001080C | 0 |
| i | 20000008 | 5000 |
| vectors | 08000000 | dword[7] |
| vectors[0] | 08000000 | 536875020 |
| vectors[1] | 08000004 | 134217973 |
| vectors[2] | 08000008 | 134217961 |
| vectors[3] | 0800000C | 134217961 |
| vectors[4] | 08000010 | 134217961 |
| vectors[5] | 08000014 | 134217961 |
| vectors[6] | 08000018 | 134217961 |
| CRH | 20000000 | 0x40010804 |
| *CRH | 40010804 | 0x44 0x44 0x24 0x44 |
| ALL_FIELDS | 40010804 | 1143227460 |
| modes | 40010804 | 0x44 0x44 0x24 0x44 |
| modepin13 | 40010804 | 2 |



CM3 Source Code - U1

main.c

```
--------  * This software component is licensed by ST under BSD 3-
--------  * the "License"; You may not use this file except in com
--------  * License. You may obtain a copy of the License at:
--------  *                        opensource.org/licenses/BSD-3-C
--------  *
--------  ******************************************************************
--------  */
--------
-------- #include "Platform_Types.h"
--------
-------- #define RCC_BASE        0x40021000
-------- #define GPIOPA_BASE     0x40010800
--------
-------- #define RCC_APB2ENR   *(volatile uint32 *) (RCC_BASE
-------- #define RCC_APB2ENR_IOPAEN 2
-------- #define set(reg,bit) reg|=1<<bit
--------
-------- typedef union{
--------     volatile uint32 ALL_FIELDS;
--------     struct {
--------         volatile uint32 :13;
--------         volatile uint32 pin13:1;
--------     }pins;
-------- }GPIOPA_ODR;
--------
-------- typedef union{
--------     volatile uint32 ALL_FIELDS;
--------     struct {
--------         volatile uint32 :20;
--------         volatile uint32 modepin13 :4;
--------     }modes;
-------- }GPIOPA_CRH;
--------
-------- volatile GPIOPA_CRH * CRH= (volatile GPIOPA_CRH *)(GPIOPA
-------- volatile GPIOPA_ODR * ODR= (volatile GPIOPA_ODR *)(GPIOPA
-------- int i;
--------
-------- int main(void)
800001C  {
8000020      set(RCC_APB2ENR,RCC_APB2ENR_IOPAEN);
--------
8000038      CRH->modes.modepin13=2;
--------
--------     for(;;){
800004E          ODR->pins.pin13=1;
8000060          for(i=0;i<5000;i++);
800009A          ODR->pins.pin13=0;
80000AC          for( i=0;i<5000;i++);
80000E6      }
-------- }
```

CM3 Variables - U1

| Name | Address | Value |
|---|---|---|
| ODR | 20000004 | 0x4001080C |
| *ODR | 4001080C | 0x00 0x20 0x00 0x00 |
| ALL_FIELDS | 4001080C | 8192 |
| pins | 4001080C | 0x00 0x20 0x00 0x00 |
| pin13 | 4001080C | 1 |
| i | 20000008 | 5000 |
| vectors | 08000000 | dword[7] |
| vectors[0] | 08000000 | 536875020 |
| vectors[1] | 08000004 | 134217973 |
| vectors[2] | 08000008 | 134217961 |
| vectors[3] | 0800000C | 134217961 |
| vectors[4] | 08000010 | 134217961 |
| vectors[5] | 08000014 | 134217961 |
| vectors[6] | 08000018 | 134217961 |
| CRH | 20000000 | 0x40010804 |
| *CRH | 40010804 | 0x44 0x44 0x24 0x44 |
| ALL_FIELDS | 40010804 | 1143227460 |
| modes | 40010804 | 0x44 0x44 0x24 0x44 |
| modepin13 | 40010804 | 2 |