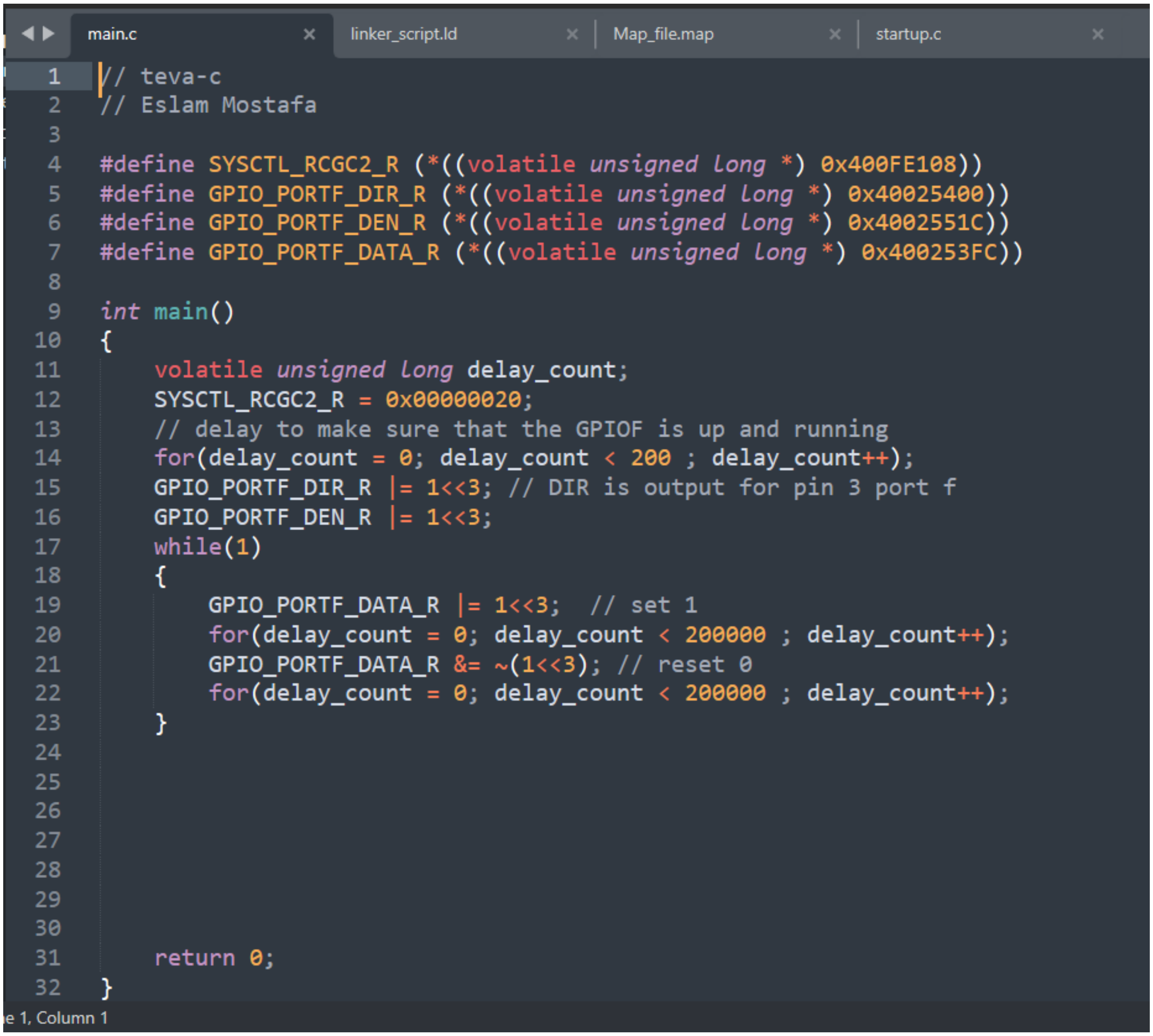
UNIT 3 LESSON 4

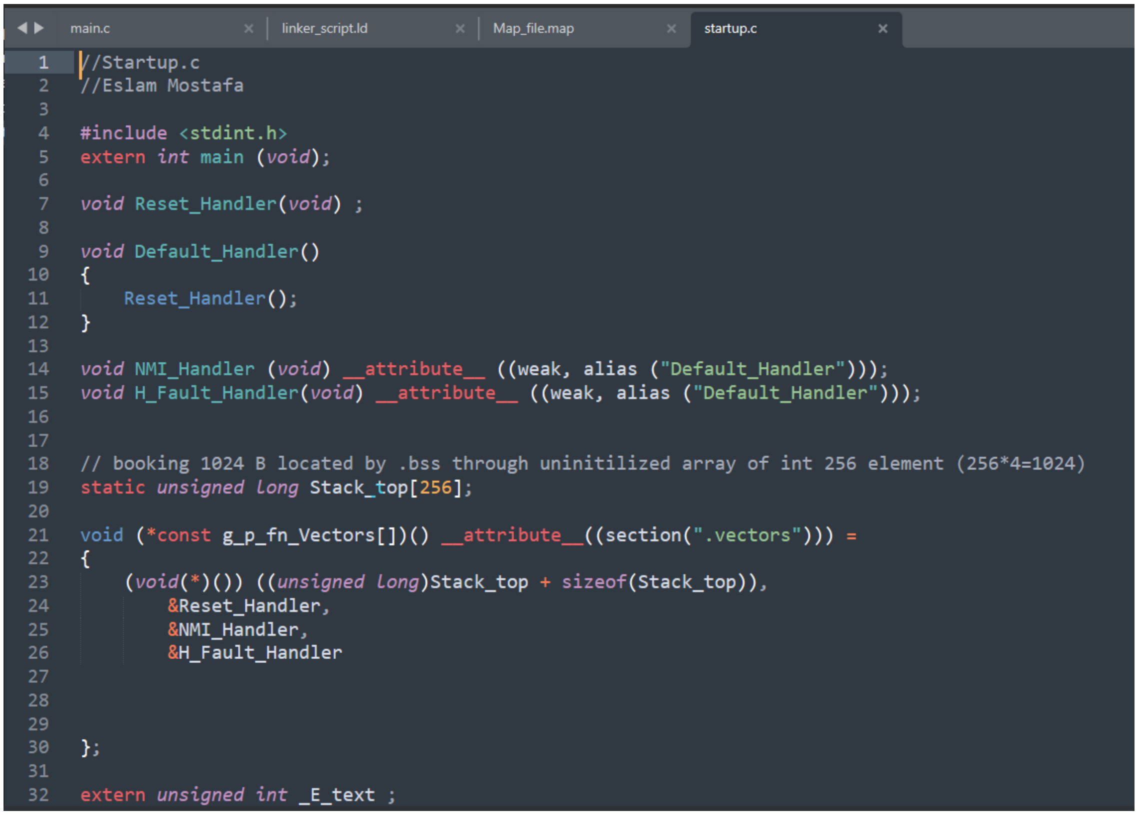
1. NAME : Eslam Mostafa

- main.c File

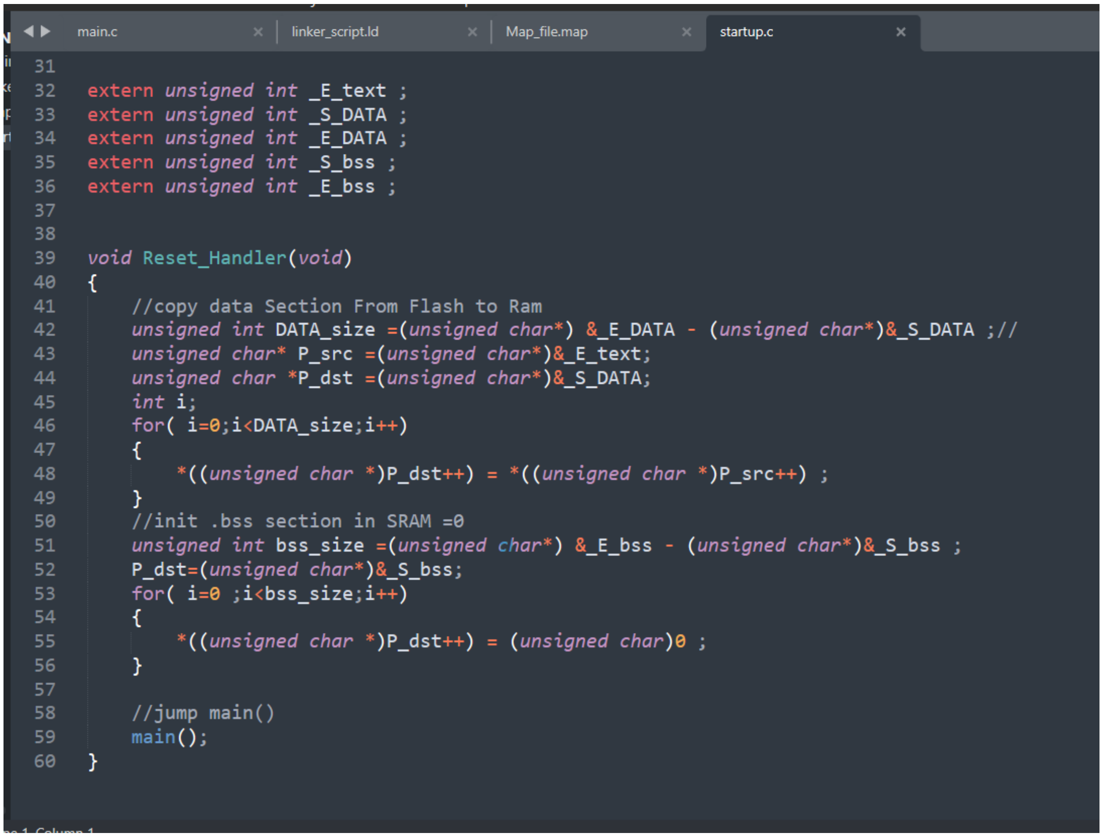


```c
// teva-c
// Eslam Mostafa

#define SYSCTL_RCGC2_R (*((volatile unsigned long *) 0x400FE108))
#define GPIO_PORTF_DIR_R (*((volatile unsigned long *) 0x40025400))
#define GPIO_PORTF_DEN_R (*((volatile unsigned long *) 0x4002551C))
#define GPIO_PORTF_DATA_R (*((volatile unsigned long *) 0x400253FC))

int main()
{
    volatile unsigned long delay_count;
    SYSCTL_RCGC2_R = 0x00000020;
    // delay to make sure that the GPIOF is up and running
    for(delay_count = 0; delay_count < 200 ; delay_count++);
    GPIO_PORTF_DIR_R |= 1<<3; // DIR is output for pin 3 port f
    GPIO_PORTF_DEN_R |= 1<<3;
    while(1)
    {
        GPIO_PORTF_DATA_R |= 1<<3;  // set 1
        for(delay_count = 0; delay_count < 200000 ; delay_count++);
        GPIO_PORTF_DATA_R &= ~(1<<3); // reset 0
        for(delay_count = 0; delay_count < 200000 ; delay_count++);
    }







    return 0;
}
```

- Startup File

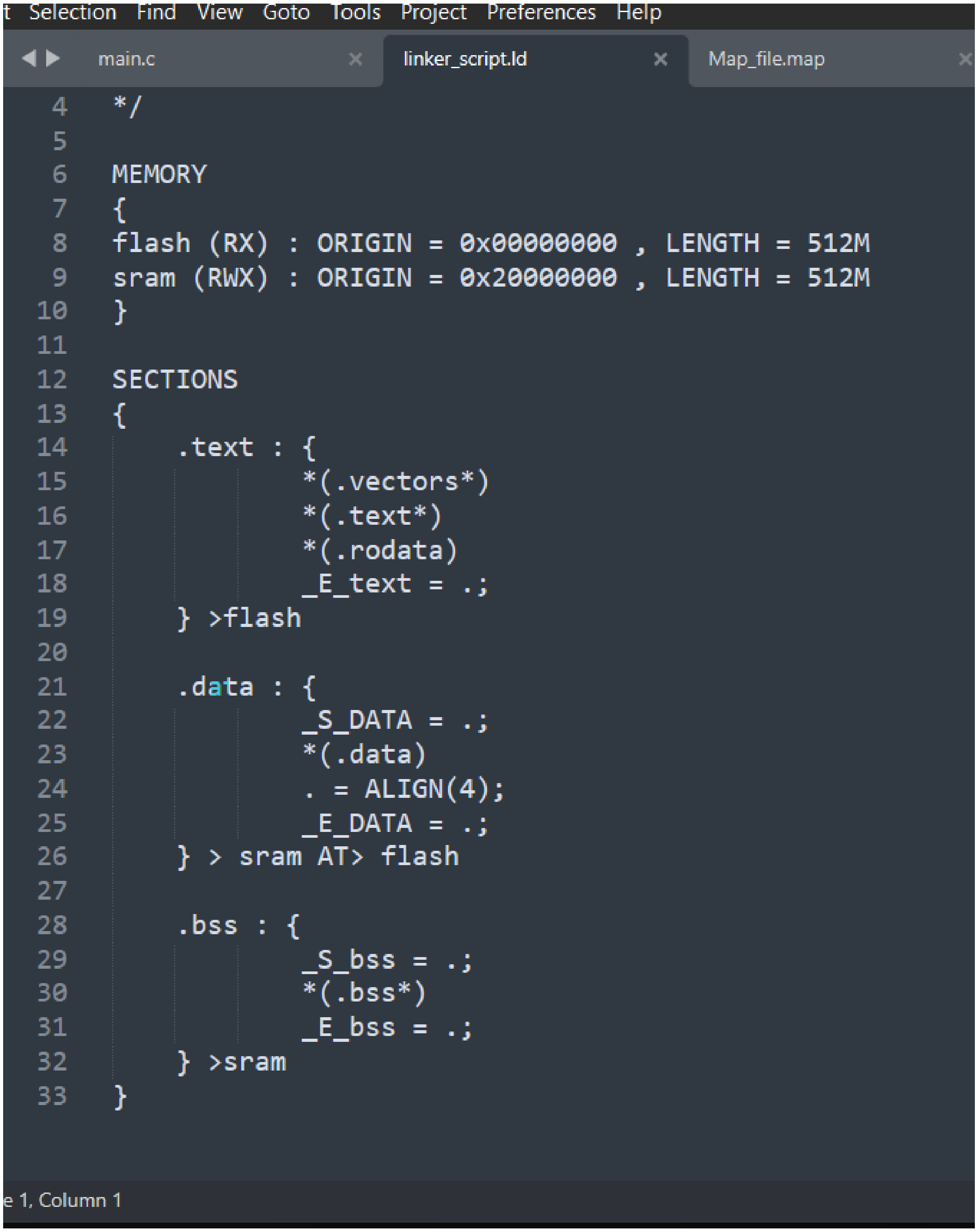


```c
//Startup.c
//Eslam Mostafa

#include <stdint.h>
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")));


// booking 1024 B located by .bss through uninitilized array of int 256 element (256*4=1024)
static unsigned long Stack_top[256];

void (*const g_p_fn_Vectors[])() __attribute__((section(".vectors"))) =
{
    (void(*)()) ((unsigned long)Stack_top + sizeof(Stack_top)),
        &Reset_Handler,
        &NMI_Handler,
        &H_Fault_Handler



};

extern unsigned int _E_text ;
```



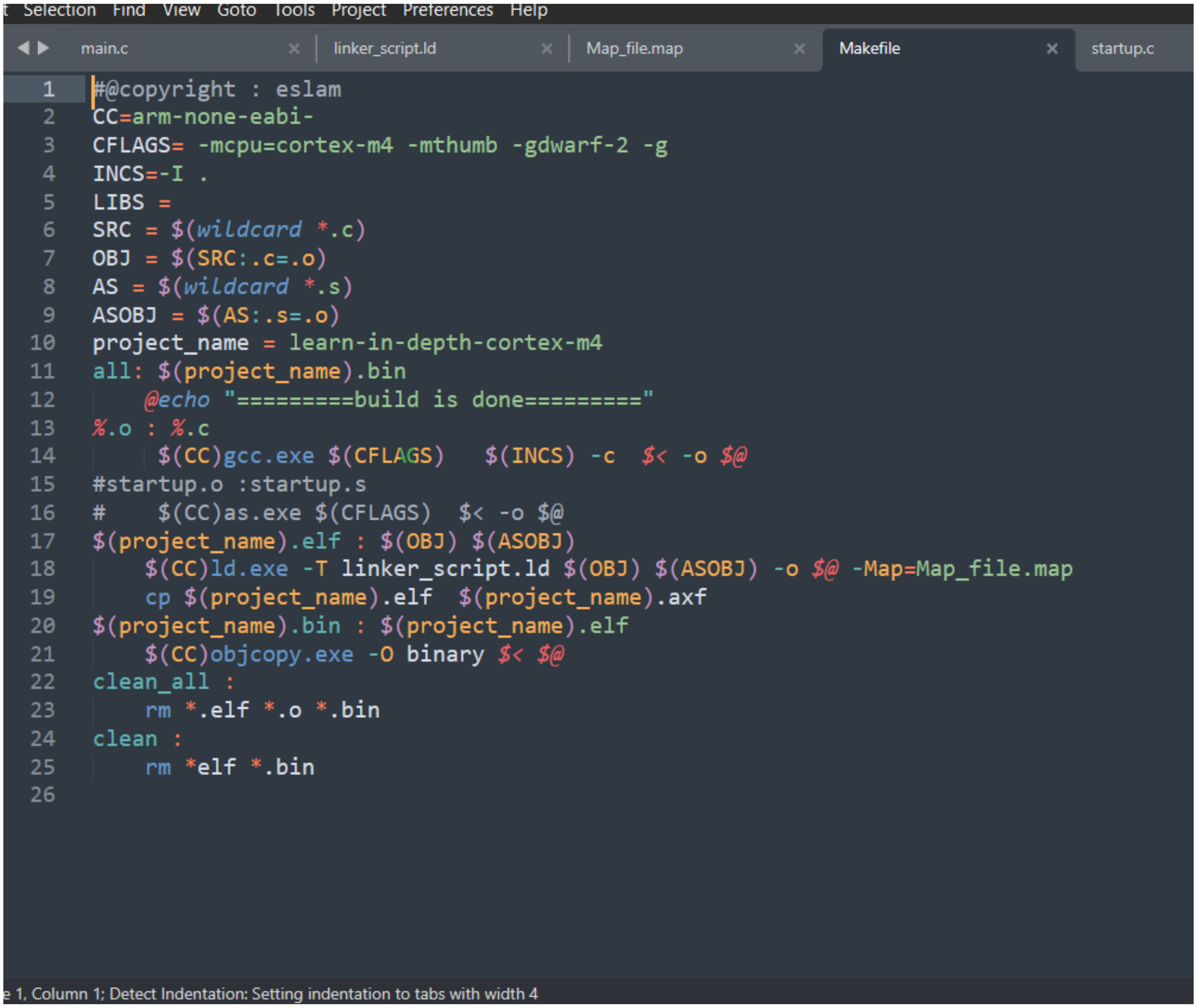
```c

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_Handler(void)
{
    //copy data Section From Flash to Ram
    unsigned int DATA_size =(unsigned char*) &_E_DATA - (unsigned char*)&_S_DATA ;//
    unsigned char* P_src =(unsigned char*)&_E_text;
    unsigned char *P_dst =(unsigned char*)&_S_DATA;
    int i;
    for( i=0;i<DATA_size;i++)
    {
        *((unsigned char *)P_dst++) = *((unsigned char *)P_src++) ;
    }
    //init .bss section in SRAM =0
    unsigned int bss_size =(unsigned char*) &_E_bss - (unsigned char*)&_S_bss ;
    P_dst=(unsigned char*)&_S_bss;
    for( i=0 ;i<bss_size;i++)
    {
        *((unsigned char *)P_dst++) = (unsigned char)0 ;
    }

    //jump main()
    main();
}
```

- LinkerScript File



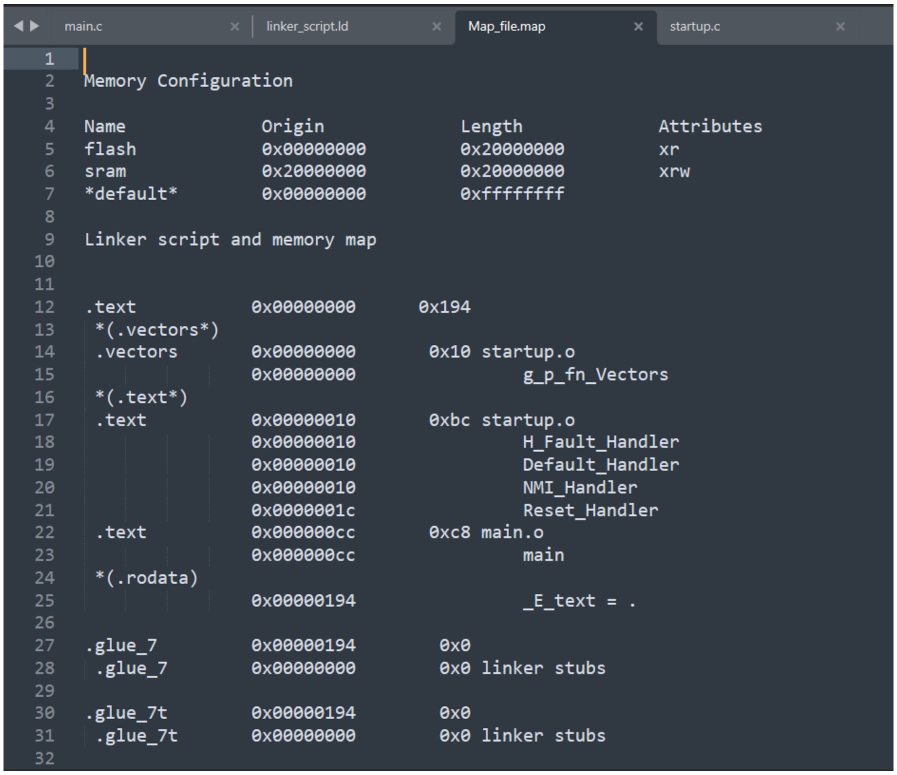
```
*/

MEMORY
{
flash (RX) : ORIGIN = 0x00000000 , LENGTH = 512M
sram (RWX) : ORIGIN = 0x20000000 , LENGTH = 512M
}

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

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

    .bss : {
            _S_bss = .;
            *(.bss*)
            _E_bss = .;
    } >sram
}
```

- MakeFile

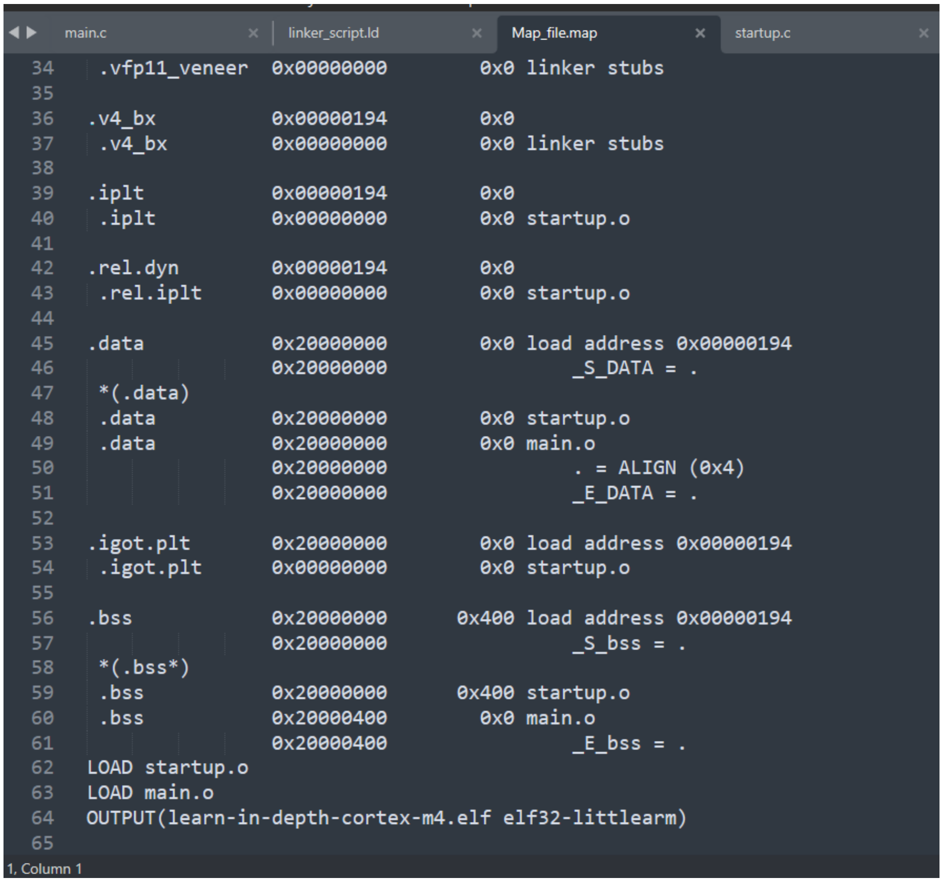

t Selection Find View Goto Tools Project Preferences Help
main.c
linker_script.ld
Map_file.map
Makefile
startup.c
e 1, Column 1; Detect Indentation: Setting indentation to tabs with width 4

```
#@copyright : eslam
CC=arm-none-eabi-
CFLAGS= -mcpu=cortex-m4 -mthumb -gdwarf-2 -g
INCS=-I .
LIBS =
SRC = $(wildcard *.c)
OBJ = $(SRC:.c=.o)
AS = $(wildcard *.s)
ASOBJ = $(AS:.s=.o)
project_name = learn-in-depth-cortex-m4
all: $(project_name).bin
	@echo "=========build is done========="
%.o : %.c
	 $(CC)gcc.exe $(CFLAGS)   $(INCS) -c  $< -o $@
#startup.o :startup.s
#    $(CC)as.exe $(CFLAGS)  $< -o $@
$(project_name).elf : $(OBJ) $(ASOBJ)
	$(CC)ld.exe -T linker_script.ld $(OBJ) $(ASOBJ) -o $@ -Map=Map_file.map
	cp $(project_name).elf  $(project_name).axf
$(project_name).bin : $(project_name).elf
	$(CC)objcopy.exe -O binary $< $@
clean_all :
	rm *.elf *.o *.bin
clean :
	rm *elf *.bin

```

- MapFile



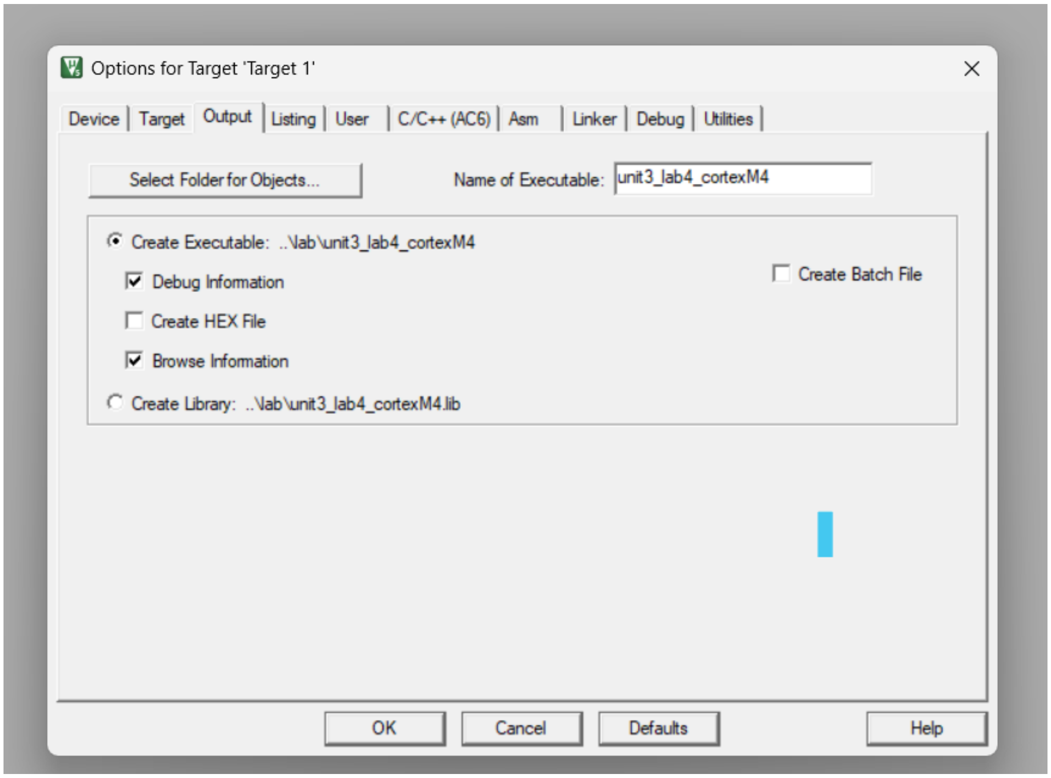
```

Memory Configuration

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

Linker script and memory map


.text           0x00000000      0x194
 *(.vectors*)
 .vectors       0x00000000       0x10 startup.o
                0x00000000                g_p_fn_Vectors
 *(.text*)
 .text          0x00000010       0xbc startup.o
                0x00000010                H_Fault_Handler
                0x00000010                Default_Handler
                0x00000010                NMI_Handler
                0x0000001c                Reset_Handler
 .text          0x000000cc       0xc8 main.o
                0x000000cc                main
 *(.rodata)
                0x00000194                _E_text = .

.glue_7         0x00000194        0x0
 .glue_7        0x00000000        0x0 linker stubs

.glue_7t        0x00000194        0x0
 .glue_7t       0x00000000        0x0 linker stubs

```



```
 .vfp11_veneer  0x00000000        0x0 linker stubs

.v4_bx          0x00000194        0x0
 .v4_bx         0x00000000        0x0 linker stubs

.iplt           0x00000194        0x0
 .iplt          0x00000000        0x0 startup.o

.rel.dyn        0x00000194        0x0
 .rel.iplt      0x00000000        0x0 startup.o

.data           0x20000000        0x0 load address 0x00000194
                0x20000000                _S_DATA = .
 *(.data)
 .data          0x20000000        0x0 startup.o
 .data          0x20000000        0x0 main.o
                0x20000000                . = ALIGN (0x4)
                0x20000000                _E_DATA = .

.igot.plt       0x20000000        0x0 load address 0x00000194
 .igot.plt      0x00000000        0x0 startup.o

.bss            0x20000000      0x400 load address 0x00000194
                0x20000000                _S_bss = .
 *(.bss*)
 .bss           0x20000000      0x400 startup.o
 .bss           0x20000400        0x0 main.o
                0x20000400                _E_bss = .
LOAD startup.o
LOAD main.o
OUTPUT(learn-in-depth-cortex-m4.elf elf32-littlearm)

```

Simulation Steps in Keil_uvisionil :

- Choose The Folder Which Has The file You Want To Simulated and Write The Executable file Name , and Choose Use Simulator



Options for Target 'Target 1'
Device | Target | Output | Listing | User | C/C++ (AC6) | Asm | Linker | Debug | Utilities
Select Folder for Objects...
OK
Cancel
Defaults
Help

Name of Executable: unit3_lab4_cortexM4

● Create Executable: ..\lab\unit3_lab4_cortexM4

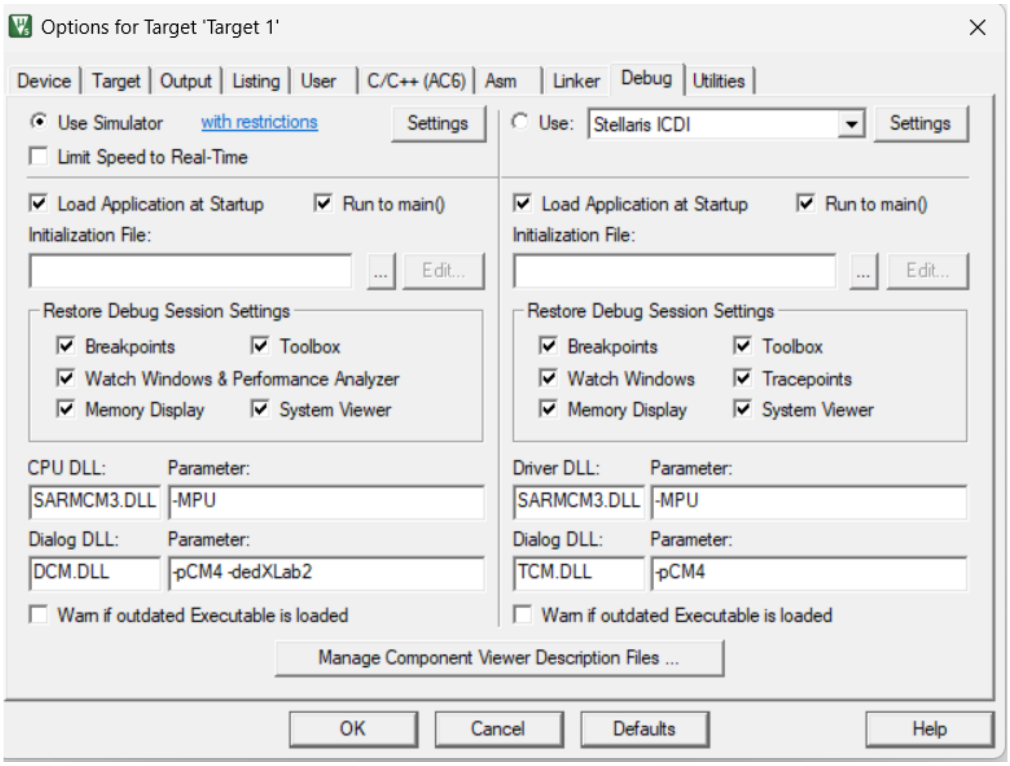
☑ Debug Information

□ Create HEX File

☑ Browse Information

□ Create Batch File

○ Create Library: ..\lab\unit3_lab4_cortexM4.lib



Options for Target 'Target 1'
Device | Target | Output | Listing | User | C/C++ (AC6) | Asm | Linker | Debug | Utilities
with restrictions
Settings
Settings
...
Edit...
...
Edit...
Manage Component Viewer Description Files ...
OK
Cancel
Defaults
Help

● Use Simulator

□ Limit Speed to Real-Time

☑ Load Application at Startup ☑ Run to main()

Initialization File:

Restore Debug Session Settings

☑ Breakpoints ☑ Toolbox

☑ Watch Windows & Performance Analyzer

☑ Memory Display ☑ System Viewer

CPU DLL: SARMCM3.DLL Parameter: -MPU

Dialog DLL: DCM.DLL Parameter: -pCM4 -dedXLab2

□ Warn if outdated Executable is loaded

○ Use: Stellaris ICDI

☑ Load Application at Startup ☑ Run to main()

Initialization File:

Restore Debug Session Settings

☑ Breakpoints ☑ Toolbox

☑ Watch Windows ☑ Tracepoints

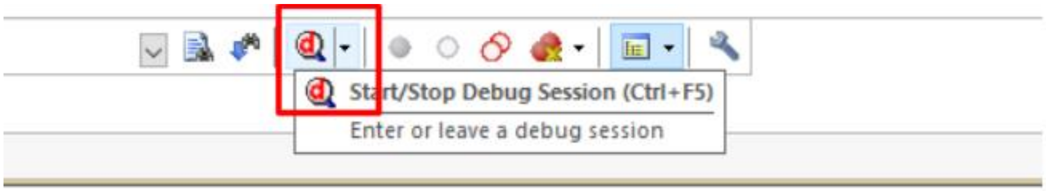
☑ Memory Display ☑ System Viewer

Driver DLL: SARMCM3.DLL Parameter: -MPU

Dialog DLL: TCM.DLL Parameter: -pCM4

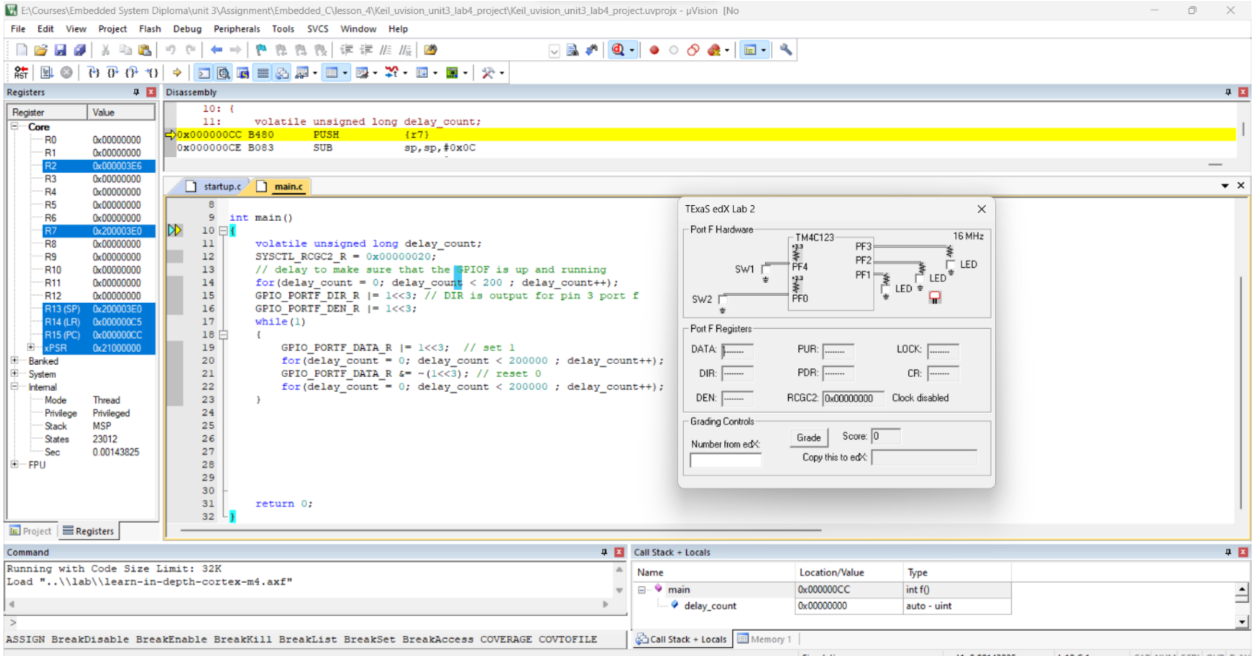
□ Warn if outdated Executable is loaded

- Start Debugging



Start/Stop Debug Session (Ctrl+F5)
Enter or leave a debug session

- Check of The output



E:\Courses\Embedded System Diploma\unit 3\Assignment\Embedded_C\lesson_4\Keil_uvision_unit3_lab4_project\Keil_uvision_unit3_lab4_project.uvprojx - µVision [No

Register	Value
Core	
R0	0x00000000
R1	0x00000000
R2	0x000003E6
R3	0x00000000
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x200003E0
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x200003E0
R14 (LR)	0x000000C5
R15 (PC)	0x000000CC
xPSR	0x21000000
Banked	
System	
Internal	
Mode	Thread
Privilege	Privileged
Stack	MSP
States	23012
Sec	0.00143825
FPU	

```
    10: {
    11:     volatile unsigned long delay_count;
0x000000CC B480      PUSH         {r7}
0x000000CE B083      SUB          sp,sp,#0x0C
```

```c

int main()
{
    volatile unsigned long delay_count;
    SYSCTL_RCGC2_R = 0x00000020;
    // delay to make sure that the GPIOF is up and running
    for(delay_count = 0; delay_count < 200 ; delay_count++);
    GPIO_PORTF_DIR_R |= 1<<3; // DIR is output for pin 3 port f
    GPIO_PORTF_DEN_R |= 1<<3;
    while(1)
    {
        GPIO_PORTF_DATA_R |= 1<<3;  // set 1
        for(delay_count = 0; delay_count < 200000 ; delay_count++);
        GPIO_PORTF_DATA_R &= ~(1<<3); // reset 0
        for(delay_count = 0; delay_count < 200000 ; delay_count++);
    }







    return 0;
}
```

TExaS edX Lab 2

Port F Registers

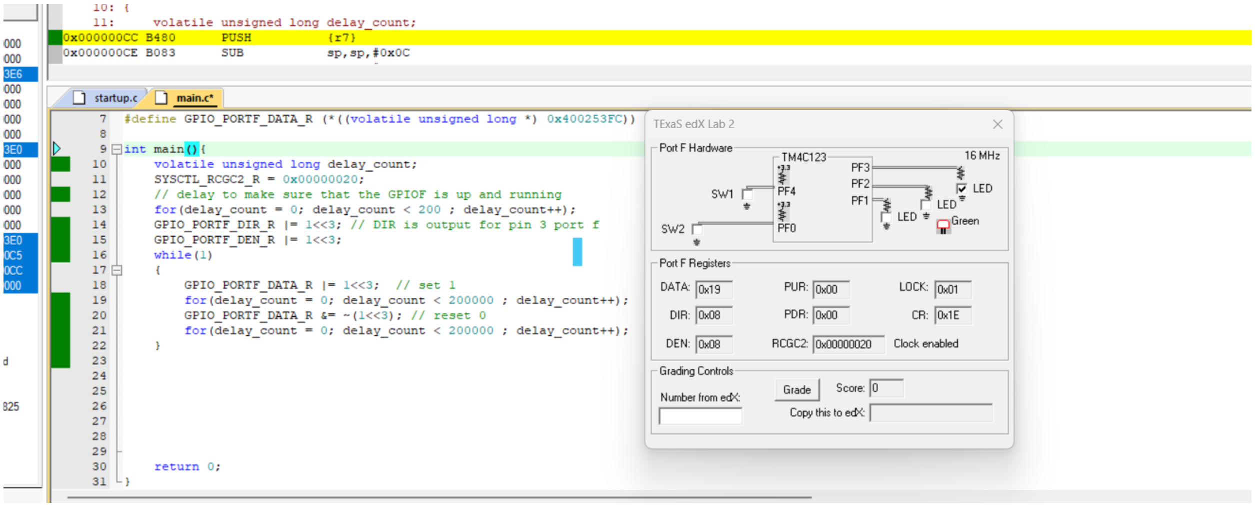
DATA: -------- PUR: -------- LOCK: --------
DIR: -------- PDR: -------- CR: --------
DEN: -------- RCGC2: 0x00000000 Clock disabled

Grading Controls

Number from edX: Grade Score: 0
Copy this to edX:

```
Running with Code Size Limit: 32K
Load "..\\lab\\learn-in-depth-cortex-m4.axf"
>
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE COVTOFILE
```

Name	Location/Value	Type
main	0x000000CC	int f()
delay_count	0x00000000	auto - uint



```
    10: {
    11:     volatile unsigned long delay_count;
0x000000CC B480      PUSH         {r7}
0x000000CE B083      SUB          sp,sp,#0x0C
```

```c
#define GPIO_PORTF_DATA_R (*((volatile unsigned long *) 0x400253FC))

int main(){
    volatile unsigned long delay_count;
    SYSCTL_RCGC2_R = 0x00000020;
    // delay to make sure that the GPIOF is up and running
    for(delay_count = 0; delay_count < 200 ; delay_count++);
    GPIO_PORTF_DIR_R |= 1<<3; // DIR is output for pin 3 port f
    GPIO_PORTF_DEN_R |= 1<<3;
    while(1)
    {
        GPIO_PORTF_DATA_R |= 1<<3;  // set 1
        for(delay_count = 0; delay_count < 200000 ; delay_count++);
        GPIO_PORTF_DATA_R &= ~(1<<3); // reset 0
        for(delay_count = 0; delay_count < 200000 ; delay_count++);
    }







    return 0;
}
```

TExaS edX Lab 2

Port F Registers

DATA: 0x19 PUR: 0x00 LOCK: 0x01
DIR: 0x08 PDR: 0x00 CR: 0x1E
DEN: 0x08 RCGC2: 0x00000020 Clock enabled

Grading Controls

Number from edX: Grade Score: 0
Copy this to edX:

Link for a video :

https://drive.google.com/file/d/1kF9j7GvACR5LkrmAtU9DCj0p7_QFanao/view?usp=sharing