ARM Cortex-M3 STM32 Lab Report

Name:	Mohamed belal
Unit:	Embedded C
Assignment:	3
Topic:	Lab2: write c code from scratch with linker, startup and make file To toggle led

1 Source Code

1.1 app.c

D:\00_Embedded System learn-in-depth\02_unit 3 Embedded C\lesson 3 - Embedded C\01_assignment & quiz\LAB_2\main.c - Sublime Text (UNREGISTERED)

main.c | Makefile | startup.c | map_file.map | linker_script.ld | Platform_Types.h | startup.s

```c
/**
 ******************************************************************************
 * @file           : main.c
 * @author         : mohamed belal
 * @brief          : Main program body
 ******************************************************************************
 */
#include <stdint.h>
#include "Platform_Types.h"

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

/*  base address + offset   */
#define RCC_APB2ENR     *(vuint32 *)(RCC_BASE + 0x18)
#define GPIOA_CRH       *(vuint32 *)(GPIOA_BASE + 0x04)
#define GPIOA_ODR       *(vuint32 *)(GPIOA_BASE + 0x0C)

typedef union
{
    vsint32 all_fields;
    struct
    {
        vsint32 reserved :13;
        vsint32 pin_13   :1;
    }Pin;
}R_ODR_t;

volatile R_ODR_t* R_ODR = (volatile R_ODR_t*)(GPIOA_BASE + 0x0C);
unsigned char g_variable[3] = {1,2,3};
unsigned char const const_varible[3] = {1,2,3};

int main(void)
{
    RCC_APB2ENR |= 1 << 2;
    GPIOA_CRH   &= 0xff0fffff;
    GPIOA_CRH   |= 0x00200000;

    while (1)
    {
//       GPIOA_ODR |= (1<<13);      //set bit
        R_ODR->Pin.pin_13 = 1;      //set bit
        for (int i = 0; i < 5000; i++);
//       GPIOA_ODR &= ~(1<<13);     //reset bit
        R_ODR->Pin.pin_13 = 0;      //reset bit
        for (int i = 0; i < 5000; i++);
    }
    return 0;
}
```

1.2 linker_script.ld

D:\00_Embedded System learn-in-depth\02_unit 3 Embedded C\lesson 3 - Embedded C\01_assignment & quiz\LAB_2\linker_script.ld - Sublime Text (UNREGISTERED)

main.c | Makefile | startup.c | map_file.map | linker_script.ld | Platform_Types.h | startup.s

```
/*
 ================================================================================
 * @author          : mohamed belal
 * @file            : linker_script.ld for unit 3 lesson 3 lab 2
 ================================================================================
 */


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*)
        _E_bss = .;
        . = ALIGN(4);
        . = . + 0x1000;
        _stack_top = .;
    } > sram
}

```

1.startup.c

D:\00_Embedded System learn-in-depth\02_unit 3 Embedded C\lesson 3 - Embedded C\01_assignment & quiz\LAB_2\startup.c • - Sublime Text (UNREGISTERED)

main.c | Makefile | startup.c | map_file.map | linker_script.ld | Platform_Types.h | startup.s

```c
/*
 ================================================================================
 * @author         : mohamed belal
 * @file           : startup.c for unit 3 lesson 3 lab 2
 ================================================================================
 */
extern int main(void);

void Default_Handler()
{
    Rest_Handler();
}
void Rest_Handler(void);
void NMI_Handler(void) __attribute__((weak,alias("Default_Handler")));;
void H_Fault_Handler(void) __attribute__((weak,alias("Default_Handler")));;
void MM_Handler(void) __attribute__((weak,alias("Default_Handler")));;
void Bus_Handler(void) __attribute__((weak,alias("Default_Handler")));;
void Usage_Handler(void) __attribute__((weak,alias("Default_Handler")));;


extern unsigned int _stack_top;
unsigned int vectors[] __attribute__((section(".vectors"))) = {
    (unsigned int) & _stack_top,
    (unsigned int) & Rest_Handler,
    (unsigned int) & NMI_Handler,
    (unsigned int) & H_Fault_Handler,
    (unsigned int) & MM_Handler,
    (unsigned int) & Bus_Handler,
    (unsigned int) & Usage_Handler,
};
```

```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 Rest_Handler(void)
{
    // copy data section from flash to sram
    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;
    for(int i=0; i<DATA_Size;i++)
    {
        *((unsigned char*)P_dst++) = *((unsigned char*)P_src++);
    }

    // int .bss section in sram ==> 0
    unsigned int Bss_Size = (unsigned char*)&_E_bss - (unsigned char*)&_S_bss ;
    P_dst = (unsigned char*)&_S_bss;
    for(int i=0;i<Bss_Size;i++)
    {
        *((unsigned char*)P_dst++) = (unsigned char)0;
    }

    // jumb main()
    main();
}
```

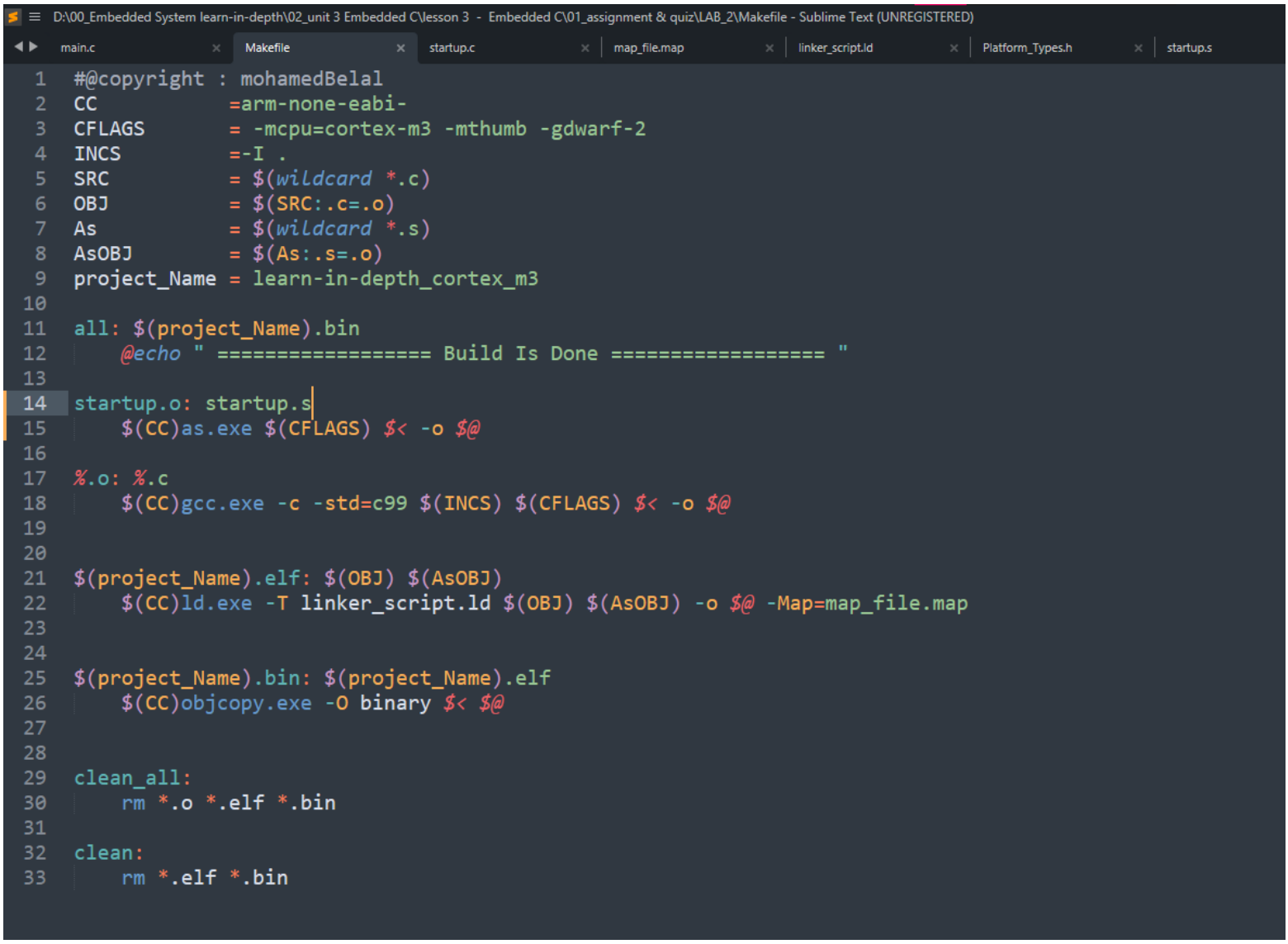
1.3 another version of startup : startup.s

D:\00_Embedded System learn-in-depth\02_unit 3 Embedded C\lesson 3 - Embedded C\01_assignment & quiz\LAB_2\startup.s • - Sublime Text (UNREGISTERED)

main.c | Makefile | startup.c | map_file.map | linker_script.ld | Platform_Types.h | startup.s

```
/*
 ================================================================================
 * @author         : mohamed belal
 * @file           : startup.s for unit 3 lesson 3 lab 2
 ================================================================================
 */

 .section .vectors

.word   0x20001000          /*  stack top adress    */
.word   _reset              /*  1 Reset */
.word   Vector_handler      /*  2 NMI   */
.word   Vector_handler      /*  3 Hart 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 ...  */
        /*  ON to IRQ67 */


 .section .text
 _reset:
    bl main
    b .

 .thumb_func
 Vector_handler:
    b _reset
```

1.4 Make File


D:\00_Embedded System learn-in-depth\02_unit 3 Embedded C\lesson 3 - Embedded C\01_assignment & quiz\LAB_2\Makefile - Sublime Text (UNREGISTERED)
main.c
Makefile
startup.c
map_file.map
linker_script.ld
Platform_Types.h
startup.s
#@copyright : mohamedBelal
CC =arm-none-eabi-
CFLAGS = -mcpu=cortex-m3 -mthumb -gdwarf-2
INCS =-I .
SRC = $(wildcard *.c)
OBJ = $(SRC:.c=.o)
As = $(wildcard *.s)
AsOBJ = $(As:.s=.o)
project_Name = learn-in-depth_cortex_m3
all: $(project_Name).bin
@echo " ================== Build Is Done ================== "
startup.o: startup.s
$(CC)as.exe $(CFLAGS) $< -o $@
%.o: %.c
$(CC)gcc.exe -c -std=c99 $(INCS) $(CFLAGS) $< -o $@
$(project_Name).elf: $(OBJ) $(AsOBJ)
$(CC)ld.exe -T linker_script.ld $(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

2 To show sections for object_file

3.1 main.o

```
MINGW64:/d/00_Embedded System learn-in-depth/02_unit 3 Embedded C/lesson 3 - Embedded C/01_assignment & quiz/LAB_2
edded C/lesson 3  -  Embedded C/01_assignment & quiz/LAB_2
$ 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         000000a8  00000000  00000000  00000034  2**2
                  CONTENTS, ALLOC, LOAD, RELOC, READONLY, CODE
  1 .data         00000008  00000000  00000000  000000dc  2**2
                  CONTENTS, ALLOC, LOAD, DATA
  2 .bss          00000000  00000000  00000000  000000e4  2**0
                  ALLOC
  3 .rodata       00000004  00000000  00000000  000000e4  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, DATA
  4 .debug_info   0000018d  00000000  00000000  000000e8  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  5 .debug_abbrev 000000f9  00000000  00000000  00000275  2**0
                  CONTENTS, READONLY, DEBUGGING
  6 .debug_loc    00000038  00000000  00000000  0000036e  2**0
                  CONTENTS, READONLY, DEBUGGING
  7 .debug_aranges 00000020  00000000  00000000  000003a6  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  8 .debug_line   0000006a  00000000  00000000  000003c6  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  9 .debug_str    00000164  00000000  00000000  00000430  2**0
                  CONTENTS, READONLY, DEBUGGING
 10 .comment      00000012  00000000  00000000  00000594  2**0
                  CONTENTS, READONLY
 11 .ARM.attributes 00000033  00000000  00000000  000005a6  2**0
                  CONTENTS, READONLY
 12 .debug_frame  0000002c  00000000  00000000  000005dc  2**2
                  CONTENTS, RELOC, READONLY, DEBUGGING

moham@DESKTOP-4ID1J68 MINGW64 /d/00_Embedded System learn-in-depth/02_unit 3 Embedded C/lesson 3  -  Embedded C/01_assignment & quiz/LAB_2
$
```

2.2 startup.o

```
MINGW64:/d/00_Embedded System learn-in-depth/02_unit 3 Embedded C/lesson 3 - Embedded C/01_assignment & quiz/LAB_2
moham@DESKTOP-4ID1J68 MINGW64 /d/00_Embedded System learn-in-depth/02_unit 3 Embedded C/lesson 3  -  Embedded C/01_assignment & quiz/LAB_2
$ 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         000000c0  00000000  00000000  00000034  2**2
                  CONTENTS, ALLOC, LOAD, RELOC, READONLY, CODE
  1 .data         00000000  00000000  00000000  000000f4  2**0
                  CONTENTS, ALLOC, LOAD, DATA
  2 .bss          00000000  00000000  00000000  000000f4  2**0
                  ALLOC
  3 .vectors      0000001c  00000000  00000000  000000f4  2**2
                  CONTENTS, ALLOC, LOAD, RELOC, DATA
  4 .debug_info   00000164  00000000  00000000  00000110  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  5 .debug_abbrev 000000e1  00000000  00000000  00000274  2**0
                  CONTENTS, READONLY, DEBUGGING
  6 .debug_loc    00000064  00000000  00000000  00000355  2**0
                  CONTENTS, READONLY, DEBUGGING
  7 .debug_aranges 00000020  00000000  00000000  000003b9  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  8 .debug_line   00000069  00000000  00000000  000003d9  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  9 .debug_str    0000011c  00000000  00000000  00000442  2**0
                  CONTENTS, READONLY, DEBUGGING
 10 .comment      00000012  00000000  00000000  0000055e  2**0
                  CONTENTS, READONLY
 11 .ARM.attributes 00000033  00000000  00000000  00000570  2**0
                  CONTENTS, READONLY
 12 .debug_frame  0000004c  00000000  00000000  000005a4  2**2
                  CONTENTS, RELOC, READONLY, DEBUGGING

moham@DESKTOP-4ID1J68 MINGW64 /d/00_Embedded System learn-in-depth/02_unit 3 Embedded C/lesson 3  -  Embedded C/01_assignment & quiz/LAB_2
$
```

2.3 lean_in_depth_cortex_m3.elf

```
MINGW64:/d/00_Embedded System learn-in-depth/02_unit 3 Embedded C/lesson 3 - Embedded C/01_assignment & quiz/LAB_2
moham@DESKTOP-4ID1J68 MINGW64 /d/00_Embedded System learn-in-depth/02_unit 3 Embedded C/lesson 3 - Embedded C/01_assignment & quiz/LAB_2
$ 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         00000188  08000000  08000000  00008000  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, CODE
  1 .data         00000008  20000000  08000188  00010000  2**2
                  CONTENTS, ALLOC, LOAD, DATA
  2 .bss          00001000  20000008  08000190  00010008  2**0
                  ALLOC
  3 .debug_info   000002f1  00000000  00000000  00010008  2**0
                  CONTENTS, READONLY, DEBUGGING
  4 .debug_abbrev 000001da  00000000  00000000  000102f9  2**0
                  CONTENTS, READONLY, DEBUGGING
  5 .debug_loc    0000009c  00000000  00000000  000104d3  2**0
                  CONTENTS, READONLY, DEBUGGING
  6 .debug_aranges 00000040  00000000  00000000  0001056f  2**0
                  CONTENTS, READONLY, DEBUGGING
  7 .debug_line   000000d3  00000000  00000000  000105af  2**0
                  CONTENTS, READONLY, DEBUGGING
  8 .debug_str    000001af  00000000  00000000  00010682  2**0
                  CONTENTS, READONLY, DEBUGGING
  9 .comment      00000011  00000000  00000000  00010831  2**0
                  CONTENTS, READONLY
 10 .ARM.attributes 00000033  00000000  00000000  00010842  2**0
                  CONTENTS, READONLY
 11 .debug_frame  00000078  00000000  00000000  00010878  2**2
                  CONTENTS, READONLY, DEBUGGING

moham@DESKTOP-4ID1J68 MINGW64 /d/00_Embedded System learn-in-depth/02_unit 3 Embedded C/lesson 3 - Embedded C/01_assignment & quiz/LAB_2
```

3 To show symbol table

4.1 main.o

```
MINGW64:/d/00_Embedded System learn-in-depth/02_unit 3 Embedded C/lesson 3 - Embedded C/01_assignment & quiz/LAB_2
 10 .ARM.attributes 00000033  00000000  00000000  00010842  2
                  CONTENTS, READONLY
 11 .debug_frame  00000078  00000000  00000000  00010878  2**
                  CONTENTS, READONLY, DEBUGGING

moham@DESKTOP-4ID1J68 MINGW64 /d/00_Embedded System learn-in-
$ arm-none-eabi-nm.exe main.o
00000000 R const_varible
00000004 D g_variable
00000000 T main
00000000 D R_ODR

moham@DESKTOP-4ID1J68 MINGW64 /d/00_Embedded System learn-in-
$
```

4.2 Startup.o

```
MINGW64:/d/00_Embedded System learn-in-depth/02_unit 3 Embedded C/lesson 3 - Embedded C/01_assignment & quiz/LAB_2

moham@DESKTOP-4ID1J68 MINGW64 /d/00_Embedded System learn-in-depth/02_unit 3 Embedded C/le
$ 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_Handler
00000000 T Default_Handler
00000000 W H_Fault_Handler
         U main
00000000 W MM_Handler
00000000 W NMI_Handler
0000000c T Rest_Handler
00000000 W Usage_Handler
00000000 D vectors

moham@DESKTOP-4ID1J68 MINGW64 /d/00_Embedded System learn-in-
```

3.2 lean_in_depth_cortex_m3.elf

```
MINGW64:/d/00_Embedded System learn-in-depth/02_unit 3 Embedded C/lesson 3 - Embedded C/01_assignment & quiz/LAB_2

moham@DESKTOP-4ID1J68 MINGW64 /d/00_Embedded System learn-in-
depth/02_unit 3 Embedded C/lesson 3  -  Embedded C/01_assignm
ent & quiz/LAB_2
$ arm-none-eabi-nm.exe learn-in-depth_cortex_m3.elf
20000008 B _E_bss
20000008 D _E_data
08000188 T _E_text
20000008 B _S_bss
20000000 D _S_data
20001008 B _stack_top
080000c4 W Bus_Handler
08000184 T const_varible
080000c4 T Default_Handler
20000004 D g_variable
080000c4 W H_Fault_Handler
0800001c T main
080000c4 W MM_Handler
080000c4 W NMI_Handler
20000000 D R_ODR
080000d0 T Rest_Handler
080000c4 W Usage_Handler
08000000 T vectors
```