LAB 2

Unit3_Lesson3

PART 1

Makefile:

```
#@copyright: Aly Mustafa Enaya
CC=arm-none-eabi-
CFLAGS=-mcpu=cortex-m3 -gdwarf-2
INCS= -I .
LIBS=
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***"
%.o: %.c
    $(CC)gcc.exe -c  $(INCS) $(CFLAGS) $< -o $@

$(project_name).elf: $(OBJ) $(ASOBJ)
    $(CC)ld.exe -T linker_script_cortex_m3.ld  $(LIBS) $(ASOBJ) $(OBJ) -o $@ -Map=Map_file.map

startup.o: startup.s
    $(CC)as.exe $(CFLAGS) $< -o $@

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

clean-all:
    rm *.o *.elf *.bin *.map

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

Startup.s

```
/*Learn-in-depth
Unit3_Lesson3_Lab2
Aly mustafa Enaya-2023*/

.section .vectors

.word 0x20001000 /*stack top address*/
.word _reset     /*1 Reset*/
.word Vector_handler
.word Vector_handler
.word Vector_handler
.word Vector_handler
.word Vector_handler
.word Vector_handler
.word Vector_handler
.word Vector_handler
.word Vector_handler
.word Vector_handler
.word Vector_handler
.word Vector_handler
.word Vector_handler
.word Vector_handler
.word Vector_handler
.word Vector_handler
.word Vector_handler
.word Vector_handler
.word Vector_handler

.section .text
_reset:
    bl main
    b .

.thumb_func
Vector_handler:
    b  reset
```

```
$ make startup.o
arm-none-eabi-as.exe -mcpu=arm926ej-s -g startup.s -o startup.o
startup.s: Assembler messages:
startup.s: Warning: end of file not at end of a line; newline inserted

metro@metro MINGW32 /d/Embedded/Unit 3 - Embedded c/Lesson 3/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         0000000c  00000000  00000000  00000034  2**2
                  CONTENTS, ALLOC, LOAD, RELOC, READONLY, CODE
  1 .data         00000000  00000000  00000000  00000040  2**0
                  CONTENTS, ALLOC, LOAD, DATA
  2 .bss          00000000  00000000  00000000  00000040  2**0
                  ALLOC
  3 .vectors      00000050  00000000  00000000  00000040  2**0
                  CONTENTS, RELOC, READONLY
  4 .ARM.attributes 00000022  00000000  00000000  00000090  2**0
                  CONTENTS, READONLY
  5 .debug_line   0000003b  00000000  00000000  000000b2  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  6 .debug_info   00000061  00000000  00000000  000000ed  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  7 .debug_abbrev 00000014  00000000  00000000  0000014e  2**0
                  CONTENTS, READONLY, DEBUGGING
  8 .debug_aranges 00000020  00000000  00000000  00000168  2**3
                  CONTENTS, RELOC, READONLY, DEBUGGING
```

Linker_script.ld:

```
/*
Unit3_Lesson3_Lab2
Aly Mustafa
*/

MEMORY
{
    flash(RX)    :ORIGIN  = 0X08000000, LENGTH = 128K
    sram(RWX)    :ORIGIN  = 0X20000000, LENGTH = 20K
}

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

    .data : {
        *(.data)

    } > flash


    .bss : { *(.bss*) } > sram


}
```

Map_file.map

```
Memory Configuration

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

Linker script and memory map


.text           0x08000000      0x154
 *(.vectors*)
 .vectors       0x08000000       0x50 startup.o
 *(.text*)
 .text          0x08000050        0xc startup.o
 .text          0x0800005c       0xf4 app.o
                0x0800005c              main
 *(.rodata)
 .rodata        0x08000150        0x4 app.o
                0x08000150              const_vars

.glue_7         0x08000154        0x0
 .glue_7        0x00000000        0x0 linker stubs

.glue_7t        0x08000154        0x0
 .glue_7t       0x00000000        0x0 linker stubs

.vfp11_veneer   0x08000154        0x0
 .vfp11_veneer  0x00000000        0x0 linker stubs

.v4_bx          0x08000154        0x0
 .v4_bx         0x00000000        0x0 linker stubs

.iplt           0x08000154        0x0

 .iplt          0x00000000        0x0 startup.o

.rel.dyn        0x08000154        0x0
 .rel.iplt      0x00000000        0x0 startup.o

.data           0x08000154        0x4
 *(.data)
 .data          0x08000154        0x0 startup.o
 .data          0x08000154        0x4 app.o
                0x08000154              g_vars

.igot.plt       0x08000158        0x0
 .igot.plt      0x00000000        0x0 startup.o

.bss            0x20000000        0x4
 *(.bss*)
 .bss           0x20000000        0x0 startup.o
 .bss           0x20000000        0x4 app.o
                0x20000000              i
LOAD startup.o
LOAD app.o
OUTPUT(learn_in_depth_cortex_M3.elf elf32-littlearm)

.ARM.attributes
                0x00000000       0x2e
 .ARM.attributes
                0x00000000       0x22 startup.o
 .ARM.attributes
                0x00000022       0x32 app.o

.comment        0x00000000       0x11
 .comment       0x00000000       0x11 app.o
                                 0x12 (size before relaxing)
```

```
arm-none-eabi-gcc.exe -c  -I . -mcpu=arm926ej-s -gdwarf-2 app.c -o app.o
arm-none-eabi-as.exe -mcpu=arm926ej-s -gdwarf-2 startup.s -o startup.o
arm-none-eabi-ld.exe -T linker_script_cortex_m3.ld   startup.o app.o -o learn_in_depth_cortex_M3.elf -Map=Map_file.map
arm-none-eabi-objcopy.exe -O binary learn_in_depth_cortex_M3.elf learn_in_depth_cortex_M3.bin
C:\WinAVR-20100110\utils\bin\make.exe: Interrupt/Exception caught (code = 0xc00000fd, addr = 0x4217b3)

metro@metro MINGW32 /d/Embedded/Unit 3 - Embedded c/Lesson 3/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         00000154  08000000  08000000  00008000  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, CODE
  1 .data         00000004  08000154  08000154  00008154  2**2
                  CONTENTS, ALLOC, LOAD, DATA
  2 .bss          00000004  20000000  20000000  00010000  2**2
                  ALLOC
  3 .ARM.attributes 0000002e  00000000  00000000  00008158  2**0
                  CONTENTS, READONLY
  4 .comment      00000011  00000000  00000000  00008186  2**0
                  CONTENTS, READONLY
  5 .debug_line   00000090  00000000  00000000  00008197  2**0
                  CONTENTS, READONLY, DEBUGGING
  6 .debug_info   00000110  00000000  00000000  00008227  2**0
                  CONTENTS, READONLY, DEBUGGING
  7 .debug_abbrev 0000009e  00000000  00000000  00008337  2**0
                  CONTENTS, READONLY, DEBUGGING
  8 .debug_aranges 00000040  00000000  00000000  000083d8  2**3
                  CONTENTS, READONLY, DEBUGGING
  9 .debug_loc    0000002c  00000000  00000000  00008418  2**0
                  CONTENTS, READONLY, DEBUGGING
 10 .debug_str    00000085  00000000  00000000  00008444  2**0
                  CONTENTS, READONLY, DEBUGGING
 11 .debug_frame  00000028  00000000  00000000  000084cc  2**2
                  CONTENTS, READONLY, DEBUGGING
```

PART 2

Makefile:

```
#@copyright: Aly Mustafa Enaya
CC=arm-none-eabi-
CFLAGS=-mcpu=cortex-m3 -gdwarf-2
INCS= -I .
LIBS=
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***"
%.o: %.c
    $(CC)gcc.exe -c  $(INCS) $(CFLAGS) $< -o $@

$(project_name).elf: $(OBJ) $(ASOBJ)
    $(CC)ld.exe -T linker_script_cortex_m3.ld  $(LIBS) $(ASOBJ) $(OBJ) -o $@ -Map=Map_file.map

#startup.o: startup.s
    #$(CC)as.exe $(CFLAGS) $< -o $@

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

clean-all:
    rm *.o *.elf *.bin *.map

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

Startup.c

```c
#include "stdint.h"
void Rest_Handler(void) ;
extern int main(void);
extern unsigned int _S_DATA ;
extern unsigned int _E_DATA ;
extern unsigned int _S_bss ;
extern unsigned int _E_bss ;
extern unsigned int _E_text;
extern unsigned int _stack_top ;

void Default_Handler()
{
    Rest_Handler();
}
void NMI_Handler(void) __attribute__((weak,alias("Default_Handler")));;
void H_fault_Handler(void)__attribute__((weak,alias("Default_Handler")));;
void MM_fault_Handler(void)__attribute__((weak,alias("Default_Handler")));;
void Bus_Fault(void)__attribute__((weak,alias("Default_Handler")));;
void Usage_Fault_Handler(void)__attribute__((weak,alias("Default_Handler")));;

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_fault_Handler,
(unsigned int ) &Bus_Fault,
(unsigned int ) &Usage_Fault_Handler,
};

void Rest_Handler(void)
{
    //copy data 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_dis = (unsigned char*)&_S_DATA ;

int i=0;
    for (i; i < DATA_size ; i++ ) {

        *((unsigned char*)P_dis++) = *((unsigned char*)P_src++);
    }

    //init the .bss section
unsigned int BSS_size = (unsigned char*)&_E_bss - (unsigned char*)&_S_bss ;
P_dis = (unsigned char*)&_S_bss ;
i=0;
    for (i ; i < BSS_size ; i++ ) {

        *((unsigned char*)P_dis++) = (unsigned char)0;
    }
    //jump to main
    main();
}
```

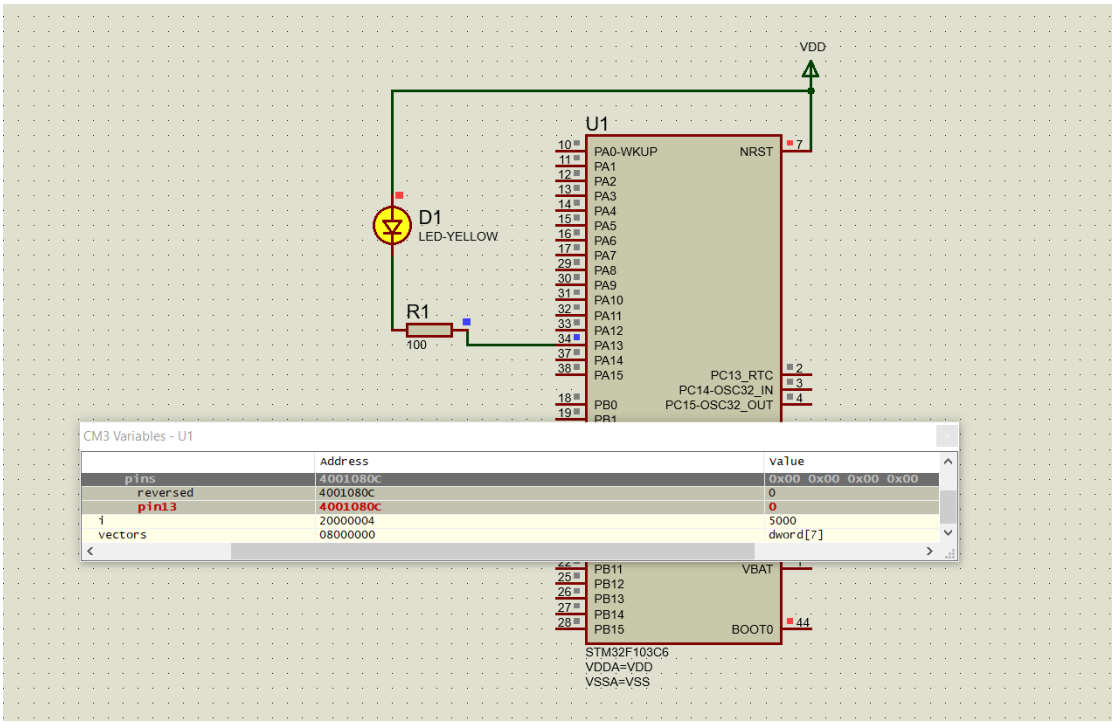
```
$ arm-none-eabi-objdump.exe -h startup.o

startup.o:     file format elf32-littlearm

Sections:
Idx Name          Size      VMA       LMA       File off  Algn
  0 .text         00000090  00000000  00000000  00000034  2**2
                  CONTENTS, ALLOC, LOAD, RELOC, READONLY, CODE
  1 .data         00000000  00000000  00000000  000000c4  2**0
                  CONTENTS, ALLOC, LOAD, DATA
  2 .bss          00000000  00000000  00000000  000000c4  2**0
                  ALLOC
  3 .vectors      0000001c  00000000  00000000  000000c4  2**2
                  CONTENTS, ALLOC, LOAD, RELOC, DATA
  4 .debug_info   00000156  00000000  00000000  000000e0  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  5 .debug_abbrev 000000b5  00000000  00000000  00000236  2**0
                  CONTENTS, READONLY, DEBUGGING
  6 .debug_loc    0000007c  00000000  00000000  000002eb  2**0
                  CONTENTS, READONLY, DEBUGGING
  7 .debug_aranges 00000020  00000000  00000000  00000367  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  8 .debug_line   0000006b  00000000  00000000  00000387  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  9 .debug_str    0000019e  00000000  00000000  000003f2  2**0
                  CONTENTS, READONLY, DEBUGGING
 10 .comment      0000007c  00000000  00000000  00000590  2**0
                  CONTENTS, READONLY
 11 .debug_frame  00000050  00000000  00000000  0000060c  2**2
                  CONTENTS, RELOC, READONLY, DEBUGGING
 12 .ARM.attributes 00000033  00000000  00000000  0000065c  2**0
                  CONTENTS, READONLY
```

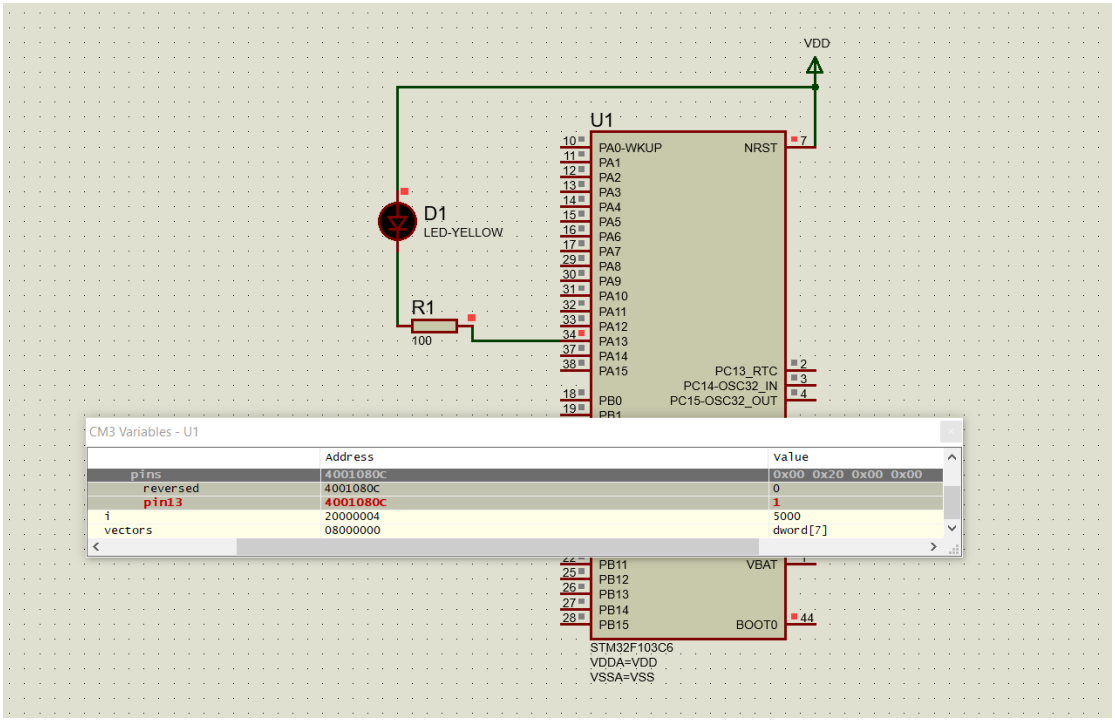
Symbols

```
20000008 B _E_bss
20000004 D _E_DATA
0800013c T _E_text
20000004 B _S_bss
20000000 D _S_DATA
20001008 B _stack_top
0800001c W Bus_Fault
0800001c T Default_Handler
0800001c W H_fault_Handler
20000004 B i
080000ac T main
0800001c W MM_fault_Handler
0800001c W NMI_Handler
20000000 D R_ODR
08000028 T Rest_Handler
0800001c W Usage_Fault_Handler
08000000 T vectors
```

Mapfile

```
text            0x0000000008000000      0x13c
*(.vectors*)
.vectors        0x0000000008000000       0x1c startup.o
                0x0000000008000000             vectors
*(.text*)
.text           0x000000000800001c       0x90 startup.o
                0x000000000800001c             MM_fault_Handler
                0x000000000800001c             Bus_Fault
                0x000000000800001c             Default_Handler
                0x000000000800001c             H_fault_Handler
                0x000000000800001c             Usage_Fault_Handler
                0x000000000800001c             NMI_Handler
                0x0000000008000028             Rest_Handler
.text           0x00000000080000ac       0x90 app.o
                0x00000000080000ac             main
*(.rodata)
                0x000000000800013c             _E_text = .

glue_7          0x000000000800013c        0x0
.glue_7         0x000000000800013c        0x0 linker stubs

glue_7t         0x000000000800013c        0x0
.glue_7t        0x000000000800013c        0x0 linker stubs

vfp11_veneer    0x000000000800013c        0x0
.vfp11_veneer   0x000000000800013c        0x0 linker stubs

v4_bx           0x000000000800013c        0x0
.v4_bx          0x000000000800013c        0x0 linker stubs

iplt            0x000000000800013c        0x0
.iplt           0x000000000800013c        0x0 startup.o

rel.dyn         0x000000000800013c        0x0
.rel.iplt       0x000000000800013c        0x0 startup.o

data            0x0000000020000000        0x4 load address 0x000000000800013c
                0x0000000020000000             _S_DATA = .
*(.data)
.data           0x0000000020000000        0x0 startup.o
.data           0x0000000020000000        0x4 app.o
                0x0000000020000000             R_ODR
                0x0000000020000004             . = ALIGN (0x4)
                0x0000000020000004             _E_DATA = .

igot.plt        0x0000000020000004        0x0 load address 0x0000000008000140
.igot.plt       0x0000000020000004        0x0 startup.o

bss             0x0000000020000004     0x1004 load address 0x0000000008000140
                0x0000000020000004             _S_bss = .
*(.bss*)
.bss            0x0000000020000004        0x0 startup.o
.bss            0x0000000020000004        0x4 app.o
                0x0000000020000004             i
                0x0000000020000008             . = ALIGN (0x4)
                0x0000000020000008             _E_bss = .
                0x0000000020000008             . = ALIGN (0x4)
                0x0000000020001008             . = (. + 0x1000)
*fill*          0x0000000020000008     0x1000
                0x0000000020001008             _stack_top = .
```


VDD
U1
D1
LED-YELLOW
R1
100
PA0-WKUP
PA1
PA2
PA3
PA4
PA5
PA6
PA7
PA8
PA9
PA10
PA11
PA12
PA13
PA14
PA15
PB0
PB1
PB11
PB12
PB13
PB14
PB15
NRST
PC13_RTC
PC14-OSC32_IN
PC15-OSC32_OUT
VBAT
BOOT0
STM32F103C6
VDDA=VDD
VSSA=VSS
CM3 Variables - U1
Address
Value
pins 4001080C 0x00 0x00 0x00 0x00
reversed 4001080C 0
pin13 4001080C 0
i 20000004 5000
vectors 08000000 dword[7]


VDD
U1
D1
LED-YELLOW
R1
100
PA0-WKUP
PA1
PA2
PA3
PA4
PA5
PA6
PA7
PA8
PA9
PA10
PA11
PA12
PA13
PA14
PA15
PB0
PB1
PB11
PB12
PB13
PB14
PB15
NRST
PC13_RTC
PC14-OSC32_IN
PC15-OSC32_OUT
VBAT
BOOT0
STM32F103C6
VDDA=VDD
VSSA=VSS
CM3 Variables - U1
Address
Value
pins 4001080C 0x00 0x20 0x00 0x00
reversed 4001080C 0
pin13 4001080C 1
i 20000004 5000
vectors 08000000 dword[7]