LAB3

- In this lab we will write a code to run on Tiva C board with ARM-CortexM4 chip

1. **Main.c:** defining GPIO Port F Registers

```c
//@Copyright: Arasny

#include"stdint.h"
#define SYSCTL_RCGC2_R          (*(volatile unsigned long*)(0x400FE108))
#define GPIO_PORTF_DATA_R       (*(volatile unsigned long*)(0x400253FC))
#define GPIO_PORTF_DIR_R        (*(volatile unsigned long*)(0x40025400))
#define GPIO_PORTF_DEN_R        (*(volatile unsigned long*)(0x4002551C))

int main(void)
{
    volatile unsigned long delay_led;
    SYSCTL_RCGC2_R = 0x00000020;
    for(delay_led = 0;delay_led<200;delay_led++);
    GPIO_PORTF_DIR_R |= 1<<3;
    GPIO_PORTF_DEN_R |= 1<<3;
    while(1)
    {
        GPIO_PORTF_DATA_R |= 1<<3;
        for(delay_led = 0;delay_led<20000;delay_led++);
        GPIO_PORTF_DATA_R &= ~(1<<3);
        for(delay_led = 0;delay_led<20000;delay_led++);
    }
    return 0;
}
```

2. **Startup.c** : defining the stack_top without extern definition from linker_script using array to function”

```c
//startup.c
//End.Arsany
#include"stdint.h"
extern int main();
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")));

//reserving stack size without using the linker_script
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,
    &MM_Fault_Handler,
    &Bus_Fault_Handler,
    &Usage_Fault_Handler
};


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;
int i;


void Reset_Handler()
{
    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( i=0;i<DATA_SIZE;i++)
    {
        *((unsigned char*)P_dst++)=*((unsigned char*)P_src);
    }
    unsigned int bss_SIZE = (unsigned char*)&_E_bss - (unsigned char*)&_S_bss;
    P_dst=(unsigned char*)&_S_bss;
    for( i=0;i<DATA_SIZE;i++)
    {
        *((unsigned char*)P_dst++) = (unsigned char)0;
    }
    main();
}
```

3. **Linker_script.ld :** removing the Stack_top symbol and adjusting address/size of flash and SRAM

```
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)
            _E_DATA = . ;

    }> sram AT> flash

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

4. Makefile:

```
#@copyright : Arsany
CC=arm-none-eabi-
CFLAGS=-mthumb -mcpu=cortex-m4 -gdwarf-2 -g
INCS=-I .
LIBS=
SRC = $(wildcard *.c)
OBJ = $(SRC:.c=.o)
As = $(wildcard *.s)
AsOBJ = $(As:.s=.o)

Project_name=Lab3_CortexM4

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
    cp $(Project_name).elf $(Project_name).axf

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

clean_all:
    rm *.o *.elf *.bin *.map
    @echo "================================CLEAN================================"
clean:
    rm *.elf *.bin *.map
```

5. Building:

```
Arshy@Arsany MINGW64 /e/EmbeddedSystemKS/Unit_3_Embedded_C/Lesson_4/Assignment/Lab3
$ make
arm-none-eabi-gcc.exe -c -I . -mthumb -mcpu=cortex-m4 -gdwarf-2 -g main.c -o main.o
arm-none-eabi-gcc.exe -c -I . -mthumb -mcpu=cortex-m4 -gdwarf-2 -g startup.c -o startup.o
arm-none-eabi-ld.exe -T linker_script.ld  main.o startup.o  -o Lab3_CortexM4.elf -Map=Map_file.map
cp Lab3_CortexM4.elf Lab3_CortexM4.axf
arm-none-eabi-objcopy.exe -O binary Lab3_CortexM4.elf Lab3_CortexM4.bin
==============================Build is Done==============================
```

6. 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      0x1d4
 *(.vectors*)
 .vectors       0x00000000       0x1c startup.o
                0x00000000                g_p_fn_Vectors
 *(.text*)
 .text          0x0000001c       0xc0 main.o
                0x0000001c                main
 .text          0x000000dc       0xf8 startup.o
                0x000000dc                Bus_Fault_Handler
                0x000000dc                H_Fault_Handler
                0x000000dc                MM_Fault_Handler
                0x000000dc                Usage_Fault_Handler
                0x000000dc                Default_Handler
                0x000000dc                NMI_Handler
                0x000000e8                Reset_Handler
 *(.rodata*)
                0x000001d4                _E_text = .

.glue_7         0x000001d4        0x0
 .glue_7        0x00000000        0x0 linker stubs

.glue_7t        0x000001d4        0x0
 .glue_7t       0x00000000        0x0 linker stubs

.vfp11_veneer   0x000001d4        0x0
 .vfp11_veneer  0x00000000        0x0 linker stubs

.v4_bx          0x000001d4        0x0
 .v4_bx         0x00000000        0x0 linker stubs
```

```
.glue_7         0x000001d4        0x0
 .glue_7        0x00000000        0x0 linker stubs

.glue_7t        0x000001d4        0x0
 .glue_7t       0x00000000        0x0 linker stubs

.vfp11_veneer   0x000001d4        0x0
 .vfp11_veneer  0x00000000        0x0 linker stubs

.v4_bx          0x000001d4        0x0
 .v4_bx         0x00000000        0x0 linker stubs

.iplt           0x000001d4        0x0
 .iplt          0x00000000        0x0 main.o

.rel.dyn        0x000001d4        0x0
 .rel.iplt      0x00000000        0x0 main.o

.data           0x20000000        0x0 load address 0x000001d4
                0x20000000                _S_DATA = .
 *(.data)
 .data          0x20000000        0x0 main.o
 .data          0x20000000        0x0 startup.o
                0x20000000                _E_DATA = .

.igot.plt       0x20000000        0x0 load address 0x000001d4
 .igot.plt      0x00000000        0x0 main.o

.bss            0x20000000      0x404 load address 0x000001d4
                0x20000000                _S_bss = .
 *(.bss)
 .bss           0x20000000        0x0 main.o
 .bss           0x20000000      0x400 startup.o
                0x20000400                _E_bss = .
 COMMON         0x20000400        0x4 startup.o
                0x20000400                i
LOAD main.o
LOAD startup.o
OUTPUT(Lab3_CortexM4.elf elf32-littlearm)
```

7. Sumilation: