Report

- **Lab3:**

- Board name : STM32f103c8t6 arm-cortex-m4 based .

- Flash starts with 0x00000000

- Sram starts with 0x20000000

1.Main .c

```c

//Eng.Ziad


#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))
#define PIN_SET             (1<<3)


int main()
{
    volatile unsigned long delay;

    SYSCTL_RCGC2_R |= 0x20;

    //to make sure GPIO is up and running

    for(delay=0;delay<200;delay++);


    GPIO_PORTF_DIR_R |= PIN_SET ;   //Dir is output for pin 3 port F

    GPIO_PORTF_DEN_R |= PIN_SET ;

    while(1)
    {
        GPIO_PORTF_DATA_R |= PIN_SET ;

        for(delay=0;delay<200000;delay++);

        GPIO_PORTF_DATA_R &= ~(PIN_SET) ;

        for(delay=0;delay<200000;delay++);

    }


    return 0;

}

```

2.Startup.c

```c
//startup.c
//Eng.Ziad

#include <stdint.h>
extern int main(void);

void Reset_Handler(void);
void Default_handler();


//to make it easy to overrid & all have the same handler to minimize the size

void NMI_Handler() __attribute__ ((weak,alias("Default_handler")));;
void H_fault_Handler() __attribute__ ((weak,alias("Default_handler")));;


//booking 1024 bytes located by .bss through unintialized array of int 256 elements (256*4=1024)



static unsigned long Stack_top[256] ; //static to make the scope for this file only


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 _S_DATA;
extern unsigned int _E_DATA;
extern unsigned int _S_bss;
extern unsigned int _E_bss;
extern unsigned int _E_text;

void Reset_Handler (void)
{
    volatile int i;
    //copy data from ROM 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;

    for(i=0;i<DATA_size; i++)
    {
        *((unsigned char*)P_dst++)= *((unsigned char*)P_src++);

    }

    // initialize the .bss with zero
    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 to main

    main();
}
void Default_handler()
{
    Reset_Handler();
}
```

3.Linker script

```
/* linker script CortexM4
Eng.Ziad
*/

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)
           . = ALIGN(4);
           _E_bss = . ;

    }> sram


}
```

4.Make file

```
#@ copyright : Eng.Ziad

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=unit3_lab4_cortexM4

all: $(project_name).bin
    @echo "------all build is done------"

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

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

$(project_name).elf: $(OBJ) $(ASOBJ)
    $(CC)ld.exe -T linker_script.ld -Map=map_file.map $(OBJ) $(ASOBJ) -o $@
    cp $(project_name).elf  $(project_name).axf

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


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

-Map file

```
Memory Configuration

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

Linker script and memory map


.text           0x00000000      0x1a4
 *(.Vectors*)
 .Vectors       0x00000000       0x10 startup.o
                0x00000000                g_P_fn_Vectors
 *(.text*)
 .text          0x00000010       0xd4 main.o
                0x00000010                main
 .text          0x000000e4       0xc0 startup.o
                0x000000e4                Reset_Handler
                0x00000198                H_fault_Handler
                0x00000198                Default_handler
                0x00000198                NMI_Handler
 *(.rodata)
                0x000001a4                _E_text = .

.glue_7         0x000001a4        0x0
 .glue_7        0x00000000        0x0 linker stubs

.glue_7t        0x000001a4        0x0
 .glue_7t       0x00000000        0x0 linker stubs

.vfp11_veneer   0x000001a4        0x0
 .vfp11_veneer  0x00000000        0x0 linker stubs

.v4_bx          0x000001a4        0x0
 .v4_bx         0x00000000        0x0 linker stubs

.iplt           0x000001a4        0x0
 .iplt          0x00000000        0x0 main.o

.rel.dyn        0x000001a4        0x0
 .rel.iplt      0x00000000        0x0 main.o
```

```
.data           0x20000000        0x0 load address 0x000001a4
                0x20000000                _S_DATA = .
 *(.data)
 .data          0x20000000        0x0 main.o
 .data          0x20000000        0x0 startup.o
                0x20000000                _E_DATA = .

.igot.plt       0x20000000        0x0 load address 0x000001a4
 .igot.plt      0x00000000        0x0 main.o

.bss            0x20000000      0x400 load address 0x000001a4
                0x20000000                _S_bss = .
 *(.bss)
 .bss           0x20000000        0x0 main.o
 .bss           0x20000000      0x400 startup.o
                0x20000400                . = ALIGN (0x4)
                0x20000400                _E_bss = .
LOAD main.o
LOAD startup.o
OUTPUT(unit3_lab4_cortexM4.elf elf32-littlearm)
```

400 = 1024

-Keil Uvision Simulation