

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

```
1 //Copyright: Arasny
2
3 #include"stdint.h"
4 #define SYSCTL_RCGC2_R      (*(volatile unsigned long*) (0x400FE108))
5 #define GPIO_PORTF_DATA_R   (*(volatile unsigned long*) (0x400253FC))
6 #define GPIO_PORTF_DIR_R    (*(volatile unsigned long*) (0x40025400))
7 #define GPIO_PORTF_DEN_R    (*(volatile unsigned long*) (0x4002551C))
8
9 int main(void)
10 {
11     volatile unsigned long delay_led;
12     SYSCTL_RCGC2_R = 0x00000020;
13     for(delay_led = 0;delay_led<200;delay_led++);
14     GPIO_PORTF_DIR_R |= 1<<3;
15     GPIO_PORTF_DEN_R |= 1<<3;
16     while(1)
17     {
18         GPIO_PORTF_DATA_R |= 1<<3;
19         for(delay_led = 0;delay_led<20000;delay_led++);
20         GPIO_PORTF_DATA_R &= ~(1<<3);
21         for(delay_led = 0;delay_led<20000;delay_led++);
22     }
23     return 0;
24 }
25
```

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

```
1 //startup.c
2 //End.Arsany
3 #include"stdint.h"
4 extern int main();
5 void Reset_Handler();
6 void Default_Handler()
7 {
8     Reset_Handler();
9 }
10 void NMI_Handler() __attribute__((weak,alias("Default_Handler")));
11 void H_Fault_Handler() __attribute__((weak,alias("Default_Handler")));
12 void MM_Fault_Handler() __attribute__((weak,alias("Default_Handler")));
13 void Bus_Fault_Handler() __attribute__((weak,alias("Default_Handler")));
14 void Usage_Fault_Handler() __attribute__((weak,alias("Default_Handler")));
15
16 //reserving stack size without using the linker_script
17 static unsigned long Stack_top[256];
18
19 void (*const g_p_fn_Vectors[])() __attribute__((section(".vectors"))) =
20 {
21     (void(*)()) ((unsigned long)Stack_top + sizeof(Stack_top)),
22     &Reset_Handler,
23     &NMI_Handler,
24     &H_Fault_Handler,
25     &MM_Fault_Handler,
26     &Bus_Fault_Handler,
27     &Usage_Fault_Handler
28 };
29
30
31 extern unsigned int _S_DATA;
32 extern unsigned int _E_DATA;
33 extern unsigned int _S_bss;
34 extern unsigned int _E_bss;
35 extern unsigned int _E_text;
36 int i;
37
38
39 void Reset_Handler()
40 {
41     unsigned int DATA_SIZE = (unsigned char*)&_E_DATA - (unsigned char*)&_S_DATA;
42     unsigned char* P_src = (unsigned char*)&_E_text;
43     unsigned char* P_dst = (unsigned char*)&_S_DATA;
44     for( i=0;i<DATA_SIZE;i++)
45     {
46         *((unsigned char*)P_dst++)=*((unsigned char*)P_src);
47     }
48     unsigned int bss_SIZE = (unsigned char*)&_E_bss - (unsigned char*)&_S_bss;
49     P_dst=(unsigned char*)&_S_bss;
50     for( i=0;i<DATA_SIZE;i++)
51     {
52         *((unsigned char*)P_dst++) = (unsigned char)0;
53     }
54     main();
55 }
```

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

```
1 MEMORY
2 {
3     flash(RX) : ORIGIN = 0x00000000, LENGTH = 512M
4     sram(RWX) : ORIGIN = 0x20000000, LENGTH = 512M
5 }
6
7 SECTIONS
8 {
9     .text : {
10         *(.vectors*)
11         *(.text*)
12         *(.rodata*)
13         _E_text = . ;
14     }> flash
15
16     .data : {
17         _S_DATA = . ;
18         *(.data)
19         _E_DATA = . ;
20     }> sram AT> flash
21
22     .bss : {
23         _S_bss = . ;
24         *(.bss)
25         _E_bss = . ;
26     }>sram
27 }
28
```

4. Makefile:

```
1 #!copyright : Arsany
2 CC=arm-none-eabi-
3 CFLAGS=-mthumb -mcpu=cortex-m4 -gdwarf-2 -g
4 INCS=-I .
5 LIBS=
6 SRC = $(wildcard *.c)
7 OBJ = $(SRC:.c=.o)
8 AS = $(wildcard *.s)
9 ASOBJ = $(AS:.s=.o)
10
11 Project_name=Lab3_CortexM4
12
13 all: $(Project_name).bin
14     @echo "=====Build is Done=====
15
16 %.o: %.c
17     $(CC)gcc.exe -c $(INCS) $(CFLAGS) $< -o $@
18
19 $(Project_name).elf: $(OBJ) $(ASOBJ)
20     $(CC)ld.exe -T linker_script.ld $(LIBS) $(OBJ) $(ASOBJ) -o $@ -Map=Map_file.map
21     cp $(Project_name).elf $(Project_name).axf
22
23 $(Project_name).bin: $(Project_name).elf
24     $(CC)objcopy.exe -O binary $< $@
25
26 clean_all:
27     rm *.o *.elf *.bin *.map
28     @echo "=====CLEAN=====
29
30 clean:
31     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:

7	Memory Configuration			
8				
9	Name	Origin	Length	Attributes
10	flash	0x00000000	0x20000000	xr
11	sram	0x20000000	0x20000000	xrw
12	*default*	0x00000000	0xffffffff	
13	Linker script and memory map			
14				
15				
16				
17	.text	0x00000000	0x1d4	
18	*(.vectors*)			
19	.vectors	0x00000000	0x1c startup.o	
20		0x00000000	g_pfn_Vectors	
21	*(.text*)			
22	.text	0x0000001c	0xc0 main.o	
23		0x0000001c	main	
24	.text	0x000000dc	0xf8 startup.o	
25		0x000000dc	Bus_Fault_Handler	
26		0x000000dc	H_Fault_Handler	
27		0x000000dc	MM_Fault_Handler	
28		0x000000dc	Usage_Fault_Handler	
29		0x000000dc	Default_Handler	
30		0x000000dc	NMI_Handler	
31		0x000000e8	Reset_Handler	
32	*(.rodata*)			
33		0x000001d4	_E_text = .	
34				
35	.glue_7	0x000001d4	0x0	
36	.glue_7	0x00000000	0x0 linker stubs	
37				
38	.glue_7t	0x000001d4	0x0	
39	.glue_7t	0x00000000	0x0 linker stubs	
40				
41	.vfp11_veneer	0x000001d4	0x0	
42	.vfp11_veneer	0x00000000	0x0 linker stubs	
43				
44	.v4_bx	0x000001d4	0x0	
45	.v4_bx	0x00000000	0x0 linker stubs	
46				
47				
48				
49				
50	.rel.dyn	0x000001d4	0x0	
51	.rel.plt	0x00000000	0x0 main.o	
52				
53	.data	0x20000000	0x0 load address 0x000001d4	
54		0x20000000	_S_DATA = .	
55	*(.data)			
56	.data	0x20000000	0x0 main.o	
57	.data	0x20000000	0x0 startup.o	
58		0x20000000	_E_DATA = .	
59				
60	.igot.plt	0x20000000	0x0 load address 0x000001d4	
61	.igot.plt	0x00000000	0x0 main.o	
62				
63	.bss	0x20000000	0x404 load address 0x000001d4	
64		0x20000000	_S_bss = .	
65	*(.bss)			
66	.bss	0x20000000	0x0 main.o	
67	.bss	0x20000000	0x400 startup.o	
68		0x20000400	_E_bss = .	
69	COMMON	0x20000400	0x4 startup.o	
70		0x20000400	i	
71	LOAD main.o			
72	LOAD startup.o			
73	OUTPUT(Lab3_CortexM4.elf elf32-littlearm)			

7. Sumilation:

The image displays two screenshots of the Keil uVision IDE, showing the simulation of a TM4C123 microcontroller. The top screenshot shows the initial state, and the bottom screenshot shows the state after the program execution.

Top Screenshot:

- Registers:** R0: 0x00000000, R1: 0x00000000, R2: 0x000027B3, R3: 0x000027B4, R4: 0x00000000, R5: 0x00000000, R6: 0x00000000, R7: 0x000027B3, R8: 0x00000000, R9: 0x00000000, R10: 0x00000000, R11: 0x00000000, R12: 0x00000000, R13: 0x200003D0, R14: 0x000001C0, R15: 0x00000000, xPSR: 0x81000000.
- Logic Analyzer:** Shows a square wave signal on PORTF. The time scale is 0.603643 s.
- Code:** The main.c file is open, showing the following code:

```
14 GPIO_PORTF_DIR_R |= 1<<3;
15 GPIO_PORTF_DEN_R |= 1<<3;
16 while(1)
17 {
18     GPIO_PORTF_DATA_R |= 1<<3;
19     for(delay_led = 0; delay_led < 20000; delay_led++);
20     GPIO_PORTF_DATA_R &= ~(1<<3);
21     for(delay_led = 0; delay_led < 20000; delay_led++);
22 }
23 return 0;
```
- Port F Hardware:** A schematic diagram of the TM4C123 microcontroller showing the connection of SW1, SW2, PF3, PF4, PF0, and LEDs. The clock is 16 MHz.
- Port F Registers:** DATA: 0x11, DIR: 0x08, DEN: 0x08, PUR: 0x00, PDR: 0x00, RCGC2: 0x00000020, LOCK: 0x01, CR: 0xFF, Clock enabled.
- Grading Controls:** Number from ed<, Grade, Score: 0, Copy this to ed<.

Bottom Screenshot:

- Registers:** R0: 0x00000000, R1: 0x00000000, R2: 0x00000019, R3: 0x400253FC, R4: 0x00000000, R5: 0x00000000, R6: 0x00000000, R7: 0x200003D0, R8: 0x00000000, R9: 0x00000000, R10: 0x00000000, R11: 0x00000000, R12: 0x00000000, R13: 0x200003D0, R14: 0x000001C0, R15: 0x00000000, xPSR: 0x21000000.
- Logic Analyzer:** Shows a square wave signal on PORTF. The time scale is 0.604012 s.
- Code:** The main.c file is open, showing the same code as the top screenshot.
- Port F Hardware:** A schematic diagram of the TM4C123 microcontroller showing the connection of SW1, SW2, PF3, PF4, PF0, and LEDs. The clock is 16 MHz.
- Port F Registers:** DATA: 0x19, DIR: 0x08, DEN: 0x08, PUR: 0x00, PDR: 0x00, RCGC2: 0x00000020, LOCK: 0x01, CR: 0xFF, Clock enabled.
- Grading Controls:** Number from ed<, Grade, Score: 0, Copy this to ed<.