Unit3. Lesson4. Lab3 Report

Eng. Farha Emad Mohamed

main.c

```
#include <stdio.h>
#include <stdlib.h>
#include <stdint.h>

#define SYSCTL_RCGC2_R *((volatile uint32_t*)(0x400FE108))
#define GPIO_PORTF_DIR_R *((volatile uint32_t*)(0x40025400))
#define GPIO_PORTF_DEN_R *((volatile uint32_t*)(0x4002551C))
#define GPIO_PORTF_DATA_R *((volatile uint32_t*)(0x400253FC))

int main()
{
   volatile uint32_t delay_count;
   SYSCTL_RCGC2_R = 0x20; /*enable clock for GPIO*/
   for(delay_count=0; delay_count<200; delay_count++);
   GPIO_PORTF_DIR_R |= (1<<3); /*configure pin 3 to be
output*/
   GPIO_PORTF_DEN_R |= (1<<3); /*enable pin 3*/

         while(1){
      GPIO_PORTF_DATA_R |= (1<<3);
      for(delay_count=0; delay_count<200000; delay_count++);
      GPIO_PORTF_DATA_R &= ~(1<<3);
      for(delay_count=0; delay_count<200000; delay_count++);
   }
   return 0;
}
```

startup.c

```
#include <stdint.h>
extern void main();
void Reset_handler(void);
void Default_handler(){
	Reset_handler();
}

void NMI_handler() __attribute__((weak,alias("Default_handler")));
void HardFault_handler() __attribute__((weak,alias("Default_handler")));
void MMFault_handler() __attribute__((weak,alias("Default_handler")));
void BusFault_handler() __attribute__((weak,alias("Default_handler")));
void UsageFault_handler() __attribute__((weak,alias("Default_handler")));

static uint32_t stack_top[256]; /*stack of 1K (265 element of 4 byte)*/

void(*const vector[])()__attribute__((section(".vectors")))={
	(void(*)()) ((uint32_t)stack_top+sizeof(stack_top)),
	&Reset_handler,
	&NMI_handler,
	&HardFault_handler,
	&MMFault_handler,
	&BusFault_handler,
	&UsageFault_handler
};
```

```
uint32_t i;
extern uint32_t _E_text;
extern uint32_t _S_data;
extern uint32_t _E_data;
extern uint32_t _S_bss;
extern uint32_t _E_bss;

void Reset_handler(void){
	/*copying .data from Flash to RAM*/
	uint32_t _data_size = (uint8_t*)&_E_data - (uint8_t*)&_S_data;
	uint8_t *ptr_scr = &_E_text;
	uint8_t *ptr_dest = &_S_data;
	for(i=0 ; i< _data_size ; i++)
	{
		*((uint8_t*)ptr_dest++) = *((uint8_t*)ptr_scr++);
	}
	/*create .bss section*/
	uint32_t _bss_size = (uint8_t*)&_E_bss - (uint8_t*)&_S_bss;
	ptr_dest = &_S_data;
	for(i=0 ; i< _data_size ; i++)
	{
		*((uint8_t*)ptr_dest++) = (uint8_t*)0;
	}
	/*branching to main*/
	main();
}
```

linker_script.ld

```
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
}
```

main.o sections

MINGW32:/d/Embedded/Learn_in_Depth/Unit3_embedded_c/EmbeddedC_lesson4/lab3

```
WIN 10@DESKTOP-BHGVA79 MINGW32 /d/Embedded/Learn_in_Depth/Unit3_embedded_c/Embed
dedC_lesson4/lab3
$ 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         0000008c  00000000  00000000  00000034  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, CODE
  1 .data         00000000  00000000  00000000  000000c0  2**0
                  CONTENTS, ALLOC, LOAD, DATA
  2 .bss          00000000  00000000  00000000  000000c0  2**0
                  ALLOC
  3 .debug_info   000009c5  00000000  00000000  000000c0  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  4 .debug_abbrev 0000019f  00000000  00000000  00000a85  2**0
                  CONTENTS, READONLY, DEBUGGING
  5 .debug_loc    00000038  00000000  00000000  00000c24  2**0
                  CONTENTS, READONLY, DEBUGGING
  6 .debug_aranges 00000020  00000000  00000000  00000c5c  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  7 .debug_line   00000227  00000000  00000000  00000c7c  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  8 .debug_str    00000558  00000000  00000000  00000ea3  2**0
                  CONTENTS, READONLY, DEBUGGING
  9 .comment      0000007f  00000000  00000000  000013fb  2**0
                  CONTENTS, READONLY
 10 .debug_frame  0000002c  00000000  00000000  0000147c  2**2
                  CONTENTS, RELOC, READONLY, DEBUGGING
 11 .ARM.attributes 00000033  00000000  00000000  000014a8  2**0
                  CONTENTS, READONLY
```

main.o symbols

```
WIN 10@DESKTOP-BHGVA79 MINGW32 /d/Embedded/Learn_in_Depth/Unit3_embedded_c/EmbeddedC_lesson4/lab3
$ arm-none-eabi-nm main.o
00000000 T main
```

startup.o sections

MINGW32:/d/Embedded/Learn_in_Depth/Unit3_embedded_c/EmbeddedC_lesson4/lab3

```
WIN 10@DESKTOP-BHGVA79 MINGW32 /d/Embedded/Learn_in_Depth/Unit3_embedded_c/EmbeddedC_lesson4/lab3
$ 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         000000a4  00000000  00000000  00000034  2**2
                  CONTENTS, ALLOC, LOAD, RELOC, READONLY, CODE
  1 .data         00000000  00000000  00000000  000000d8  2**0
                  CONTENTS, ALLOC, LOAD, DATA
  2 .bss          00000400  00000000  00000000  000000d8  2**2
                  ALLOC
  3 .vectors      0000001c  00000000  00000000  000000d8  2**2
                  CONTENTS, ALLOC, LOAD, RELOC, READONLY, DATA
  4 .debug_info   000001b1  00000000  00000000  000000f4  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  5 .debug_abbrev 000000d7  00000000  00000000  000002a5  2**0
                  CONTENTS, READONLY, DEBUGGING
  6 .debug_loc    0000007c  00000000  00000000  0000037c  2**0
                  CONTENTS, READONLY, DEBUGGING
  7 .debug_aranges 00000020  00000000  00000000  000003f8  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  8 .debug_line   0000013c  00000000  00000000  00000418  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  9 .debug_str    000001e0  00000000  00000000  00000554  2**0
                  CONTENTS, READONLY, DEBUGGING
 10 .comment      0000007f  00000000  00000000  00000734  2**0
                  CONTENTS, READONLY
 11 .debug_frame  00000050  00000000  00000000  000007b4  2**2
                  CONTENTS, RELOC, READONLY, DEBUGGING
 12 .ARM.attributes 00000033  00000000  00000000  00000804  2**0
                  CONTENTS, READONLY
```

startup.o symbols

```
WIN 10@DESKTOP-BHGVA79 MINGW32 /d/Embedded/Learn_in_Depth/Unit3_embedded_c/EmbeddedC_lesson4/lab3
$ arm-none-eabi-nm startup.o
         U _E_bss
         U _E_data
         U _E_text
         U _S_bss
         U _S_data
00000000 W BusFault_handler
00000000 T Default_handler
00000000 W HardFault_handler
00000004 C i
         U main
00000000 W MMFault_handler
00000000 W NMI_handler
0000000c T Reset_handler
00000000 b stack_top
00000000 W UsageFault_handler
00000000 R vector
```

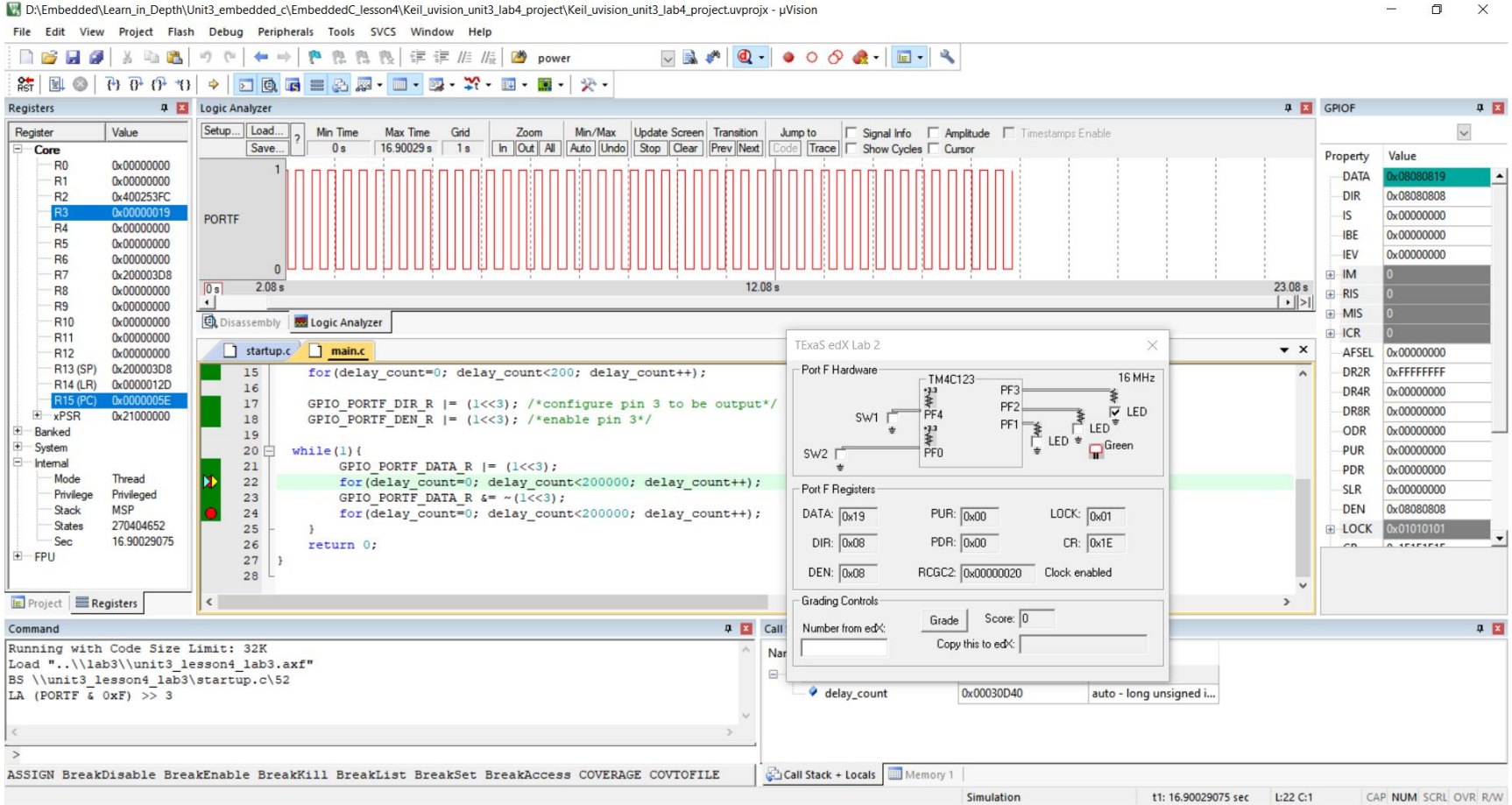
Unit3_lesson4_lab3.elf sections

```
WIN 10@DESKTOP-BHGVA79 MINGW32 /d/Embedded/Learn_in_Depth/Unit3_embedded_c/EmbeddedC_lesson4/lab3
$ arm-none-eabi-objdump.exe -h unit3_lesson4_lab3.elf

unit3_lesson4_lab3.elf:     file format elf32-littlearm

Sections:
Idx Name          Size      VMA       LMA       File off  Algn
  0 .text         0000014c  00000000  00000000  00010000  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, CODE
  1 .bss          00000404  20000000  0000014c  00020000  2**2
                  ALLOC
  2 .debug_info   00000b76  00000000  00000000  0001014c  2**0
                  CONTENTS, READONLY, DEBUGGING
  3 .debug_abbrev 00000276  00000000  00000000  00010cc2  2**0
                  CONTENTS, READONLY, DEBUGGING
  4 .debug_loc    000000b4  00000000  00000000  00010f38  2**0
                  CONTENTS, READONLY, DEBUGGING
  5 .debug_aranges 00000040  00000000  00000000  00010fec  2**0
                  CONTENTS, READONLY, DEBUGGING
  6 .debug_line   00000363  00000000  00000000  0001102c  2**0
                  CONTENTS, READONLY, DEBUGGING
  7 .debug_str    0000057a  00000000  00000000  0001138f  2**0
                  CONTENTS, READONLY, DEBUGGING
  8 .comment      0000007e  00000000  00000000  00011909  2**0
                  CONTENTS, READONLY
  9 .ARM.attributes 00000033  00000000  00000000  00011987  2**0
                  CONTENTS, READONLY
 10 .debug_frame  0000007c  00000000  00000000  000119bc  2**2
                  CONTENTS, READONLY, DEBUGGING
```

Unit3_lesson4_lab3.elf symbols

```
WIN 10@DESKTOP-BHGVA79 MINGW32 /d/Embedded/Learn_in_Depth/Unit3_embedded_c/EmbeddedC_lesson4/lab3
$ arm-none-eabi-nm unit3_lesson4_lab3.elf
20000400 B _E_bss
20000000 T _E_data
0000014c T _E_text
20000000 B _S_bss
20000000 T _S_data
000000a8 W BusFault_handler
000000a8 T Default_handler
000000a8 W HardFault_handler
20000400 B i
0000001c T main
000000a8 W MMFault_handler
000000a8 W NMI_handler
000000b4 T Reset_handler
20000000 b stack_top
000000a8 W UsageFault_handler
00000000 T vector
```

output


D:\Embedded\Learn_in_Depth\Unit3_embedded_c\EmbeddedC_lesson4\Keil_uvision_unit3_lab4_project\Keil_uvision_unit3_lab4_project.uvprojx - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Logic Analyzer
PORTF
GPIOF
startup.c
main.c
for(delay_count=0; delay_count<200; delay_count++);
GPIO_PORTF_DIR_R |= (1<<3); /*configure pin 3 to be output*/
GPIO_PORTF_DEN_R |= (1<<3); /*enable pin 3*/
while(1){
GPIO_PORTF_DATA_R |= (1<<3);
for(delay_count=0; delay_count<200000; delay_count++);
GPIO_PORTF_DATA_R &= ~(1<<3);
for(delay_count=0; delay_count<200000; delay_count++);
}
return 0;
}
TExaS edX Lab 2
Port F Hardware
TM4C123
16 MHz
SW1
SW2
PF4
PF0
PF3
PF2
PF1
LED
LED
LED
Green
Port F Registers
DATA: 0x19
PUR: 0x00
LOCK: 0x01
DIR: 0x08
PDR: 0x00
CR: 0x1E
DEN: 0x08
RCGC2: 0x00000020
Clock enabled
Grading Controls
Number from edX:
Grade
Score: 0
Copy this to edX:
Command
Running with Code Size Limit: 32K
Load "..\\lab3\\unit3_lesson4_lab3.axf"
BS \\unit3_lesson4_lab3\startup.c\52
LA (PORTF & 0xF) >> 3
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE COVTOFILE
delay_count
0x00030D40
auto - long unsigned i...
Simulation
t1: 16.90029075 sec
L:22 C:1