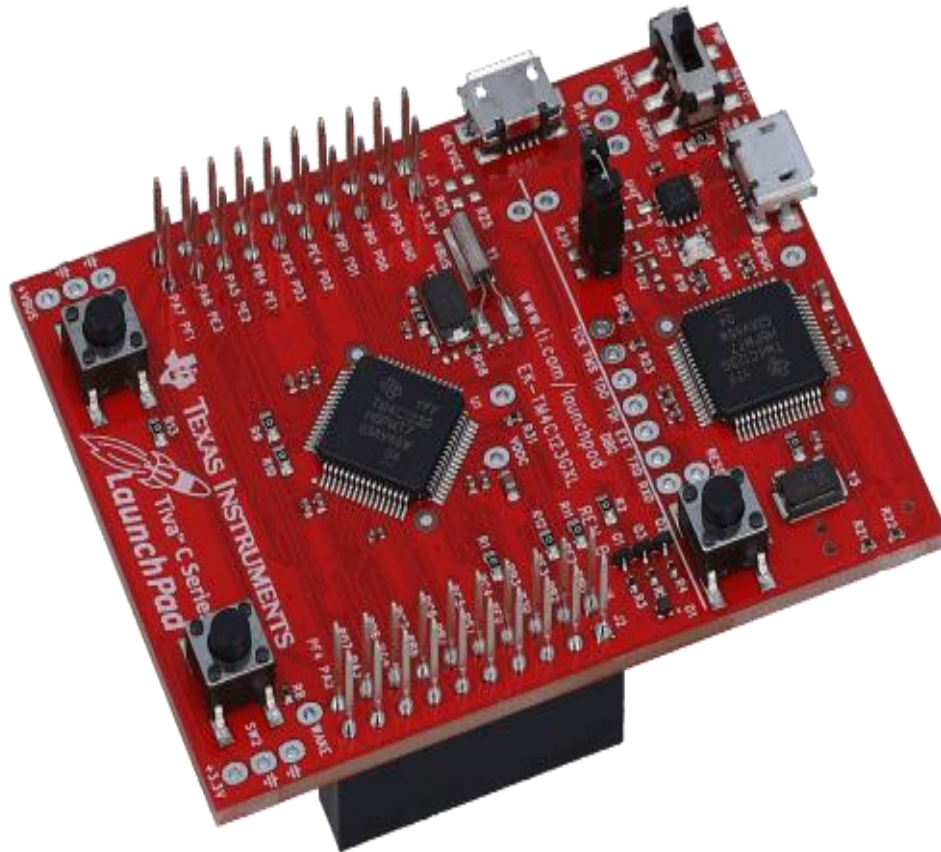


# Embedded C

## Unit 3 Lesson 4 ( LAB 3 )



**Name : Mohammed Adel Mohammed Elmasry**

# main.o

```
kf 7@DESKTOP-E11MB1Q MINGW32 /d/projects/lab 3
```

```
$ 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	00000098	00000000	00000000	000000c0	2**0
	CONTENTS, RELOC, READONLY, DEBUGGING					
4	.debug_abbrev	0000005a	00000000	00000000	00000158	2**0
	CONTENTS, READONLY, DEBUGGING					
5	.debug_loc	00000038	00000000	00000000	000001b2	2**0
	CONTENTS, READONLY, DEBUGGING					
6	.debug_aranges	00000020	00000000	00000000	000001ea	2**0
	CONTENTS, RELOC, READONLY, DEBUGGING					
7	.debug_line	00000061	00000000	00000000	0000020a	2**0
	CONTENTS, RELOC, READONLY, DEBUGGING					
8	.debug_str	00000110	00000000	00000000	0000026b	2**0
	CONTENTS, READONLY, DEBUGGING					
9	.comment	0000007c	00000000	00000000	0000037b	2**0
	CONTENTS, READONLY					
10	.debug_frame	0000002c	00000000	00000000	000003f8	2**2
	CONTENTS, RELOC, READONLY, DEBUGGING					
11	.ARM.attributes	00000033	00000000	00000000	00000424	2**0
	CONTENTS, READONLY					

# startup.o

```
kf 7@DESKTOP-E11MB1Q MINGW32 /d/projects/lab 3
$ 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	00000400	00000000	00000000	000000c4	2**2
	ALLOC					
3	.vectors	0000001c	00000000	00000000	000000c4	2**2
	CONTENTS, ALLOC, LOAD, RELOC, READONLY, DATA					
4	.debug_info	000001ca	00000000	00000000	000000e0	2**0
	CONTENTS, RELOC, READONLY, DEBUGGING					
5	.debug_abbrev	000000eb	00000000	00000000	000002aa	2**0
	CONTENTS, READONLY, DEBUGGING					
6	.debug_loc	0000007c	00000000	00000000	00000395	2**0
	CONTENTS, READONLY, DEBUGGING					
7	.debug_aranges	00000020	00000000	00000000	00000411	2**0
	CONTENTS, RELOC, READONLY, DEBUGGING					
8	.debug_line	000001f2	00000000	00000000	00000431	2**0
	CONTENTS, RELOC, READONLY, DEBUGGING					
9	.debug_str	0000019d	00000000	00000000	00000623	2**0
	CONTENTS, READONLY, DEBUGGING					
10	.comment	0000007c	00000000	00000000	000007c0	2**0
	CONTENTS, READONLY					
11	.debug_frame	00000050	00000000	00000000	0000083c	2**2
	CONTENTS, RELOC, READONLY, DEBUGGING					
12	.ARM.attributes	00000033	00000000	00000000	0000088c	2**0
	CONTENTS, READONLY					

# learn-in-depth-cortex\_m4.axf

```
kf 7@DESKTOP-E11MB1Q MINGW32 /d/projects/lab 3
$ arm-none-eabi-objdump.exe -h learn-in-depth-cortex_m4.axf

learn-in-depth-cortex_m4.axf:      file format elf32-littlearm

Sections:
Idx Name          Size      VMA       LMA       File off  Algn
  0 .text          00000138  00000000  00000000  00010000  2**2
    CONTENTS, ALLOC, LOAD, READONLY, CODE
  1 .data          00000000  20000000  00000138  00020000  2**0
    CONTENTS, ALLOC, LOAD, DATA
  2 .bss           00000400  20000000  00000138  00020000  2**2
    ALLOC
  3 .debug_info     00000262  00000000  00000000  00020000  2**0
    CONTENTS, READONLY, DEBUGGING
  4 .debug_abbrev   00000145  00000000  00000000  00020262  2**0
    CONTENTS, READONLY, DEBUGGING
  5 .debug_loc      000000b4  00000000  00000000  000203a7  2**0
    CONTENTS, READONLY, DEBUGGING
  6 .debug_aranges  00000040  00000000  00000000  0002045b  2**0
    CONTENTS, READONLY, DEBUGGING
  7 .debug_line     00000253  00000000  00000000  0002049b  2**0
    CONTENTS, READONLY, DEBUGGING
  8 .debug_str      0000016c  00000000  00000000  000206ee  2**0
    CONTENTS, READONLY, DEBUGGING
  9 .comment        0000007b  00000000  00000000  0002085a  2**0
    CONTENTS, READONLY
10 .ARM.attributes 00000033  00000000  00000000  000208d5  2**0
    CONTENTS, READONLY
11 .debug_frame     0000007c  00000000  00000000  00020908  2**2
    CONTENTS, READONLY, DEBUGGING
```

# Simulation

The screenshot displays the Texas Instruments Code Composer Studio (CCS) interface during a simulation of a Cortex-M4 microcontroller. The main window shows a logic analyzer trace of the PORTF register, which is a square wave alternating between 0 and 1. The left pane shows the register list with R15 (PC) selected. The center pane shows the C code for the main function, which configures the GPIO and toggles the PORTF register. The right pane shows the hardware schematic of the TM4C123 microcontroller, highlighting the GPIO pins and LEDs. The bottom pane shows the command window with the command 'LA (PORTF & 0xF) >> 3' and the call stack showing the main function.

**Registers**

Register	Value
R0	0x00000000
R1	0x00000000
R2	0x00004E1F
R3	0x000015BA
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x2000003C
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x2000003C
R14 (LR)	0x0000011D
R15 (PC)	0x0000009C
xPSR	0x01000000

**Code**

```
9 {
10 volatile unsigned long i;
11 SYSCTL_RCGC2_R = 0x20;
12 for(i=0; i<2000; i++);
13 GPIO_PORTF_DIR_R |= 1<<3;
14 GPIO_PORTF_DEN_R |= 1<<3;
15 while(1)
16 {
17     GPIO_PORTF_DATA_R |= 1<<3;
18     for(i=0; i<2000; i++);
19     GPIO_PORTF_DATA_R &= ~(1<<3);
20     for(i=0; i<2000; i++);
21 }
22 return 0;
23 }
24
25 }
```

**Port F Hardware**

TM4C123 16 MHz

SW1 PF4 PF3 PF2 PF1 PF0 LED LED Green

**Port F Registers**

Register	Value
DATA	0x19
DIR	0x08
DEN	0x08
RCGC2	0x00000020
LOCK	0x01
PDR	0x00
CR	0x1E

**Grading Controls**

Number from ed: Grade Score: 0

Copy this to ed:

**Command**

```
Load "D:\projects\lab 4\learn-in-depth-cortex_m4.axf"
LA (PORTF & 0xF) >> 3
```

**Call Stack - Locals**

Name	Location/Value	Type
main	0x0000001C	int f()
i	0x000015BA	auto - uint

**Simulation** t1: 15.60701288 sec L18 C1 CAP NUM SCL OVR R/W