

## *Embedded C Unit3 Lab (2)*



**ARM CORTEX-M3 STM32F103**

### **Unit 3 Lesson 3 Lab (2)**

**Name : Mohammed Hassan Ahmed**

# Embedded C Unit3 Lab (2)

## 1 – The .obj files of (main.o & startup.o):

### Main.o

```
mohas@DESKTOP-VP4AFMT MINGW32 ~/OneDrive/Desktop/Lab 2/startup.c
$ 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	00000094	00000000	00000000	00000034	2**2
	CONTENTS, ALLOC, LOAD, RELOC, READONLY, CODE					
1	.data	0000001c	00000000	00000000	000000c8	2**2
	CONTENTS, ALLOC, LOAD, DATA					
2	.bss	00000000	00000000	00000000	000000e4	2**0
	ALLOC					
3	.debug_info	000001cf	00000000	00000000	000000e4	2**0
	CONTENTS, RELOC, READONLY, DEBUGGING					
4	.debug_abbrev	00000111	00000000	00000000	000002b3	2**0
	CONTENTS, READONLY, DEBUGGING					
5	.debug_loc	000000c0	00000000	00000000	000003c4	2**0
	CONTENTS, READONLY, DEBUGGING					
6	.debug_aranges	00000020	00000000	00000000	00000484	2**0
	CONTENTS, RELOC, READONLY, DEBUGGING					
7	.debug_line	0000006e	00000000	00000000	000004a4	2**0
	CONTENTS, RELOC, READONLY, DEBUGGING					
8	.debug_str	00000190	00000000	00000000	00000512	2**0
	CONTENTS, READONLY, DEBUGGING					
9	.comment	0000007f	00000000	00000000	000006a2	2**0
	CONTENTS, READONLY					
10	.debug_frame	0000006c	00000000	00000000	00000724	2**2
	CONTENTS, RELOC, READONLY, DEBUGGING					
11	.ARM.attributes	00000033	00000000	00000000	00000790	2**0
	CONTENTS, READONLY					

## Embedded C Unit3 Lab (2)

### Startup.o

```
mohas@DESKTOP-VP4AFMT MINGW32 ~/OneDrive/Desktop/Lab 2/startup.c
$ 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	00000000	00000000	00000000	000000c4	2**0
	ALLOC					
3	.vectors	0000001c	00000000	00000000	000000c4	2**2
	CONTENTS, ALLOC, LOAD, RELOC, DATA					
4	.debug_info	0000019e	00000000	00000000	000000e0	2**0
	CONTENTS, RELOC, READONLY, DEBUGGING					
5	.debug_abbrev	000000d6	00000000	00000000	0000027e	2**0
	CONTENTS, READONLY, DEBUGGING					
6	.debug_loc	0000007c	00000000	00000000	00000354	2**0
	CONTENTS, READONLY, DEBUGGING					
7	.debug_aranges	00000020	00000000	00000000	000003d0	2**0
	CONTENTS, RELOC, READONLY, DEBUGGING					
8	.debug_line	0000007d	00000000	00000000	000003f0	2**0
	CONTENTS, RELOC, READONLY, DEBUGGING					
9	.debug_str	000001ab	00000000	00000000	0000046d	2**0
	CONTENTS, READONLY, DEBUGGING					
10	.comment	0000007f	00000000	00000000	00000618	2**0
	CONTENTS, READONLY					
11	.debug_frame	00000050	00000000	00000000	00000698	2**2
	CONTENTS, RELOC, READONLY, DEBUGGING					
12	.ARM.attributes	00000033	00000000	00000000	000006e8	2**0
	CONTENTS, READONLY					

## Embedded C Unit3 Lab (2)

### Alias & Weak

We can override on weak symbols

```
mohas@DESKTOP-VP4AFMT MINGW32 ~/OneDrive/Desktop/Lab 2/startup.c
$ arm-none-eabi-nm.exe learn-in-depth.elf
2000001c B _E_Bss
2000001c D _E_Data
08000140 T _E_Text
2000001c B _S_Bss
20000000 D _S_Data
2000101c B _Stack_Top
08000028 T Bus_Fault
20000010 D const_variables
080000b0 T Default_Handler
20000004 D global_variables
080000b0 W H_Fault_Handler
08000034 T main
080000b0 W MM_Fault_Handler
0800001c T NMI_Handler
20000000 D R_ODR
080000bc T Reset_Handler
080000b0 W Usage_Fault_Handler
08000000 T vectors
```

# Embedded C Unit3 Lab (2)

## Main.c

```
C:\Users\mohas\OneDrive\Desktop\Lab 2\startup.c\main.c - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
Platform_Types.h x main.c x startup.c x linker_script.ld x Map_File.map x log.txt x + ▼
1 1 /*
2 =====
3 Name      : main.c
4 Author    : Mohammed Hassan
5 Created on : 9/20/2023
6 Description : Unit 3==> Lesson 3 ==> Lab(2)
7 =====
8 */
9 #include "Platform_Types.h"
10
11 #define RCC_BASE      0x40021000
12 #define PortA_BASE    0x40010800
13
14 #define RCC_APB2ENR   *(vuint32_t *) (RCC_BASE + 0x18)
15 #define GPIO_CRH      *(vuint32_t *) (PortA_BASE + 0x04)
16 #define GPIO_ODR      *(vuint32_t *) (PortA_BASE + 0x0C)
17
18 typedef union
19 {
20     vuint32_t all_fields ;
21
22     struct
23     {
24         vuint32_t reserved : 13 ;
25         vuint32_t pin_13 : 1 ;
26     } pin;
27
28 } R_ODR_t;
29
30 volatile R_ODR_t *R_ODR = (volatile R_ODR_t *) (PortA_BASE + 0x0C) ;
31 vuint32_t global_variables[3] = {1,2,3};
32 vuint32_t const const_variavles[3] = {1,2,3};
33
34 extern void NMI_Handler(void)
35 {
36 }
37
38 extern void Bus_Fault(void)
39 {
40 }
41
42 }
43
44 int main(void)
45 {
46     RCC_APB2ENR |= (1<<2) ; //set pit 2
47     GPIO_CRH    &= 0xff0fffff ;
48     GPIO_CRH    |= 0x00200000 ; //set pits(20-24) 2
49
50     while(1)
51     {
52         R_ODR->pin.pin_13 = 1 ;
53         for(int i = 0 ; i < 5000 ; i++);
54         R_ODR->pin.pin_13 = 0 ;
55         for(int i = 0 ; i < 5000 ; i++);
56     }
57     return 0 ;
58 }
```

# Embedded C Unit3 Lab (2)

## Startup.c

```
C:\Users\mohas\OneDrive\Desktop\Lab 2\startup.c\startup.c - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
Platform_Types.h x main.c x startup.c x linker_script.ld x Map_File.map x log.txt x + v
1  /*=====
2  Name      : startup.c
3  Author    : Mohammed Hassan
4  Created on : 9/20/2023
5  Description : Unit 3==> Lesson 3 ==> Lab(2)
6  =====*/
7  #include "Platform_Types.h"
8
9  extern int main(void);
10 extern uint32_t _Stack_Top ;
11 extern uint32_t _E_Text ;
12 extern uint32_t _S_Data ;
13 extern uint32_t _E_Data ;
14 extern uint32_t _S_Bss ;
15 extern uint32_t _E_Bss ;
16
17 void Reset_Handler(void);
18
19 void Default_Handler()
20 {
21     Reset_Handler();
22 }
23
24 void NMI_Handler(void)      __attribute__((weak, alias("Default_Handler")));
25 void H_Fault_Handler(void) __attribute__((weak, alias("Default_Handler")));
26 void MM_Fault_Handler(void) __attribute__((weak, alias("Default_Handler")));
27 void Bus_Fault(void)        __attribute__((weak, alias("Default_Handler")));
28 void Usage_Fault_Handler(void) __attribute__((weak, alias("Default_Handler")));
29
30
31 uint32_t vectors[] __attribute__((section(".vectors"))) =
32 {
33     (uint32_t) &Stack_Top ,
34     (uint32_t) &Reset_Handler ,
35     (uint32_t) &NMI_Handler ,
36     (uint32_t) &H_Fault_Handler ,
37     (uint32_t) &MM_Fault_Handler ,
38     (uint32_t) &Bus_Fault ,
39     (uint32_t) &Usage_Fault_Handler
40 };
41
42 void Reset_Handler(void)
43 {
44     //Copy Data From Rom to Ram
45     uint32_t Data_Size = (uint8_t*)&E_Data - (uint8_t*)&S_Data ;
46     uint8_t* P_src = (uint8_t*)&E_Text ;
47     uint8_t* P_dst = (uint8_t*)&S_Data ;
48
49     for(int i =0 ; i < Data_Size ; i++)
50     {
51         *((uint8_t*)P_dst++) = *((uint8_t*)P_src++) ;
52     }
53
54     // init the .bss with 0
55
56     uint32_t Bss_Size = (uint8_t*)&E_Bss - (uint8_t*)&S_Bss ;
57     P_dst = (uint8_t*)&S_Bss ;
58
59     for(int i =0 ; i < Bss_Size ; i++)
60     {
61         *((uint8_t*)P_dst++) = (uint8_t)0 ;
62     }
63
64     // jump to main()
65     main();
66 }
```

Line 29, Column 1 Tab Size: 4 C

# Embedded C Unit3 Lab (2)

## Linker Script.ld

```
C:\Users\mohas\OneDrive\Desktop\Lab 2\startup.c\linker_script.ld - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
Platform_Types.h x main.c x startup.c x linker_script.ld x Map_File.map x log.txt x + v
1 /*=====
2 Name      : linker_script.ld
3 Author    : Mohammed Hassan
4 Created on : 9/20/2023
5 Description : Unit 3==>> Lesson 3 ==>> Lab(2)
6 =====*/
7 MEMORY
8 {
9     flash(RX) : ORIGIN = 0x08000000, LENGTH = 128K
10    sram(RWX)  : ORIGIN = 0x20000000, LENGTH = 20K
11 }
12
13 SECTIONS
14 {
15     .text : {
16         *(.vectors*)
17         *(.text*)
18         *(.rodata*)
19         _E_Text = . ;
20     }>flash
21
22     .data : {
23         _S_Data = . ;
24         *(.data*)
25         . = ALIGN(4);
26         _E_Data = . ;
27     }>sram AT> flash
28
29     .bss : {
30         _S_Bss = . ;
31         *(.bss*)
32         . = ALIGN(4);
33         _E_Bss = . ;
34
35         . = ALIGN(4);
36         . = . + 0x1000 ;
37         _Stack_Top = . ;
38     }>sram
39 }
40
```

Line 1, Column 1 Tab Size: 4 Plain Text

# Embedded C Unit3 Lab (2)

## Map File .map

```
C:\Users\mohas\OneDrive\Desktop\Lab 2\startup.c\Map_File.map - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
Platform_Types.h x main.c x startup.c x linker_script.ld x Map_File.map x log.txt x + v
1
2 Memory Configuration
3
4 Name Origin Length Attributes
5 flash 0x08000000 0x00020000 xrw
6 sram 0x20000000 0x00050000 xrw
7 *default* 0x00000000 0xffffffff
8
9 Linker script and memory map
10
11
12 .text 0x08000000 0x140
13 *(.vectors*)
14 .vectors 0x08000000 0x1c startup.o
15 0x08000000 vectors
16
17 *(.text*)
18 .text 0x0800001c 0x94 main.o
19 0x0800001c NMI_Handler
20 0x08000028 Bus_Fault
21 0x08000034 main
22
23 .text 0x080000b0 0x90 startup.o
24 0x080000b0 MM_Fault_Handler
25 0x080000b0 Usage_Fault_Handler
26 0x080000b0 Default_Handler
27 0x080000b0 H_Fault_Handler
28 0x080000bc Reset_Handler
29
30 *(.rodata*)
31 0x08000140 _E_Text = .
32
33 .glue_7 0x08000140 0x0
34 .glue_7 0x08000140 0x0 linker stubs
35
36 .glue_7t 0x08000140 0x0
37 .glue_7t 0x08000140 0x0 linker stubs
38
39 .vfp11_veneer 0x08000140 0x0
40 .vfp11_veneer 0x08000140 0x0 linker stubs
41
42 .v4_bx 0x08000140 0x0
43 .v4_bx 0x08000140 0x0 linker stubs
44
45 .iplt 0x08000140 0x0
46 .iplt 0x08000140 0x0 main.o
47
48 .rel.dyn 0x08000140 0x0
49 .rel.iplt 0x08000140 0x0 main.o
50
51 .data 0x20000000 0x1c load address 0x08000140
52 0x20000000 _S_Data = .
53
54 *(.data*)
55 .data 0x20000000 0x1c main.o
56 0x20000000 R_ODR
57 0x20000004 global_variables
58 0x20000010 const_variables
59
60 .data 0x2000001c 0x0 startup.o
61 0x2000001c . = ALIGN (0x4)
62 0x2000001c _E_Data = .
63
64 .igot.plt 0x2000001c 0x0 load address 0x0800015c
65 .igot.plt 0x2000001c 0x0 main.o
66
67 .bss 0x2000001c 0x1000 load address 0x0800015c
```



# Embedded C Unit3 Lab (2)

## Startup.s

```
C:\Users\mohas\OneDrive\Desktop\Lab 2\startup.s\startup.s - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
Platform_Types.h x main.c x startup.c x linker_script.ld x Map_File.map x log.txt x startup.s x + ▾

1  /*=====
2  Name      : startup.s
3  Author    : Mohammed Hassan
4  Created on : 9/20/2023
5  Description : Unit 3==>> Lesson 3 ==>> Lab(2)
6  =====*/
7
8  .section .vectors
9
10 .word 0x20001000      /* stack top address */
11 .word _reset          /* 1 Reset */
12 .word Vector_Handler /* 2 NMI */
13 .word Vector_Handler /* 3 Hard Fault */
14 .word Vector_Handler /* 4 MM Fault */
15 .word Vector_Handler /* 5 Bus Fault */
16 .word Vector_Handler /* 6 Usage Fault */
17 .word Vector_Handler /* 7 RESERVED */
18 .word Vector_Handler /* 8 RESERVED */
19 .word Vector_Handler /* 9 RESERVED */
20 .word Vector_Handler /* 10 RESERVED */
21 .word Vector_Handler /* 11 SV Call */
22 .word Vector_Handler /* 12 Debug Reserved */
23 .word Vector_Handler /* 13 RESERVED */
24 .word Vector_Handler /* 14 PendSV */
25 .word Vector_Handler /* 15 SysTick */
26 .word Vector_Handler /* 16 IRQ0 */
27 .word Vector_Handler /* 17 IRQ1 */
28 .word Vector_Handler /* 18 IRQ2 */
29 .word Vector_Handler /* 19 ... */
30
31 .section .text
32 _reset:
33     bl main
34     b .      /*After finishing main loop in your self*/
35
36 .thumb_func /*For using 16 bits & 32 bits instruction*/
37 Vector_Handler:
38     b _reset

Line 38, Column 17
Tab Size: 4
Plain Text
```

# Embedded C Unit3 Lab (2)

## Simulation

CM3 Variables - U1

Name	Address
g_variables	20000004
const_variables	08000128
ss_var	20000108
vectors	08000000
R_ODR	20000000
1	BP+12 = #20000FDC BP+16 = #20000FDB

CM3 Registers - U1

PC	0800005A	Privileged
Pr1	00_200	Mode Thread
R0	00000000	R7 20000FDB
R1	20000008	R8 00000000
R2	000013B7	R9 00000000
R3	00001240	R10 00000000
R4	00000000	R11 00000000
R5	00000000	R12 00000000
R6	00000000	LR 0800010D
NSP	20000FDB	PSP 00000000
IRQ	0	EPSR 01000000
APSR	CNVZ	1c1t 000_00000

CM3 Source Code - U1

```
main.c
/*
 * Name : main.c
 * Author : Mohamed Hassan
 * Created on : 9/20/2023
 * Description : Unit 3==> Lesson 3 ==> Lab(2)
 */
#include "Platform_Types.h"
#define RCC_BASE 0x40021000
#define PORTA_BASE 0x40010800
#define RCC_APB2ENR (*(volatile R_ODR_t *) (RCC_BASE + 0x18))
#define GPIO_CRH (*(volatile R_ODR_t *) (PORTA_BASE + 0x04))
#define GPIO_ODR (*(volatile R_ODR_t *) (PORTA_BASE + 0x0C))

typedef union
{
    uint32_t all_fields;
    struct
    {
        uint32_t reserved : 13;
        uint32_t pin13 : 1;
    } pin;
} R_ODR_t;

volatile R_ODR_t *R_ODR = (volatile R_ODR_t *) (PORTA_BASE + 0x0C);
uint32_t global_variables[] = {1,2,3};
uint32_t const_variables[] = {1,2,3};

extern void NMI_Handler(void)
{
    R_ODR->pin13 = 1;
    for(int i = 0; i < 5000; i++);
    R_ODR->pin13 = 0;
    for(int i = 0; i < 5000; i++);
}

return 0;

int main(void)
{
    RCC_APB2ENR |= (1<<2); //set pit 2
    GPIO_CRH |= 0xFF0FFFFF;
    GPIO_ODR |= 0x00200000; //set pins(20-24) 2

    while(1)
    {
        R_ODR->pin13 = 1;
        for(int i = 0; i < 5000; i++);
        R_ODR->pin13 = 0;
        for(int i = 0; i < 5000; i++);
    }
}
```

CM3 FLASH at 0x08000000 - U1

08000000	08 10 00 20	...	08 08 01 40	...
08000004	AS 00 00 08	...	01 02 03 00	...
08000008	99 00 00 08	...	00 00 00 00	...
0800000C	99 00 00 08	...	00 00 00 00	...
08000010	99 00 00 08	...	00 00 00 00	...
08000014	99 00 00 08	...	00 00 00 00	...
08000018	99 00 00 08	...	00 00 00 00	...
0800001C	80 84 83 80	...	00 00 00 00	...
08000020	00 AF 14 48	...	00 00 00 00	...
08000024	18 68 18 4A	...	00 00 00 00	...
08000028	43 F0 04 03	...	00 00 00 00	...
0800002C	13 60 18 48	...	00 00 00 00	...
08000030	18 68 17 4A	...	00 00 00 00	...
08000034	23 F4 70 03	...	00 00 00 00	...
08000038	13 60 15 48	...	00 00 00 00	...
0800003C	18 68 14 4A	...	00 00 00 00	...
08000040	43 F4 00 13	...	00 00 00 00	...
08000044	13 60 13 48	...	00 00 00 00	...
08000048	1A 68 13 88	...	00 00 00 00	...
0800004C	43 F4 00 53	...	00 00 00 00	...
08000050	13 80 00 23	...	00 00 00 00	...
08000054	78 60 00 20	...	00 00 00 00	...
08000058	78 68 01 33	...	00 00 00 00	...
0800005C	78 60 78 68	...	00 00 00 00	...
08000060	41 F2 87 32	...	00 00 00 00	...
08000064	04 48 1A 68	...	00 00 00 00	...
08000068	13 88 6F F3	...	00 00 00 00	...
0800006C	40 33 13 80	...	00 00 00 00	...
08000070	00 23 38 60	...	00 00 00 00	...
08000074	00 23 38 60	...	00 00 00 00	...

CM3 RAM at 0x20000000 - U1

20000000	0C 08 01 40	...	0C 08 01 40	...
20000004 <td>00 00 00 00</td> <td>...</td> <td>00 00 00 00</td> <td>...</td>	00 00 00 00	...	00 00 00 00	...
20000008 <td>00 00 00 00</td> <td>...</td> <td>00 00 00 00</td> <td>...</td>	00 00 00 00	...	00 00 00 00	...
2000000C <td>00 00 00 00</td> <td>...</td> <td>00 00 00 00</td> <td>...</td>	00 00 00 00	...	00 00 00 00	...
20000010 <td>00 00 00 00</td> <td>...</td> <td>00 00 00 00</td> <td>...</td>	00 00 00 00	...	00 00 00 00	...
20000014 <td>00 00 00 00</td> <td>...</td> <td>00 00 00 00</td> <td>...</td>	00 00 00 00	...	00 00 00 00	...
20000018 <td>00 00 00 00</td> <td>...</td> <td>00 00 00 00</td> <td>...</td>	00 00 00 00	...	00 00 00 00	...
2000001C <td>00 00 00 00</td> <td>...</td> <td>00 00 00 00</td> <td>...</td>	00 00 00 00	...	00 00 00 00	...
20000020 <td>00 00 00 00</td> <td>...</td> <td>00 00 00 00</td> <td>...</td>	00 00 00 00	...	00 00 00 00	...
20000024 <td>00 00 00 00</td> <td>...</td> <td>00 00 00 00</td> <td>...</td>	00 00 00 00	...	00 00 00 00	...
20000028 <td>00 00 00 00</td> <td>...</td> <td>00 00 00 00</td> <td>...</td>	00 00 00 00	...	00 00 00 00	...
2000002C <td>00 00 00 00</td> <td>...</td> <td>00 00 00 00</td> <td>...</td>	00 00 00 00	...	00 00 00 00	...
20000030 <td>00 00 00 00</td> <td>...</td> <td>00 00 00 00</td> <td>...</td>	00 00 00 00	...	00 00 00 00	...
20000034 <td>00 00 00 00</td> <td>...</td> <td>00 00 00 00</td> <td>...</td>	00 00 00 00	...	00 00 00 00	...
20000038 <td>00 00 00 00</td> <td>...</td> <td>00 00 00 00</td> <td>...</td>	00 00 00 00	...	00 00 00 00	...
2000003C <td>00 00 00 00</td> <td>...</td> <td>00 00 00 00</td> <td>...</td>	00 00 00 00	...	00 00 00 00	...
20000040 <td>00 00 00 00</td> <td>...</td> <td>00 00 00 00</td> <td>...</td>	00 00 00 00	...	00 00 00 00	...
20000044 <td>00 00 00 00</td> <td>...</td> <td>00 00 00 00</td> <td>...</td>	00 00 00 00	...	00 00 00 00	...
20000048 <td>00 00 00 00</td> <td>...</td> <td>00 00 00 00</td> <td>...</td>	00 00 00 00	...	00 00 00 00	...
2000004C <td>00 00 00 00</td> <td>...</td> <td>00 00 00 00</td> <td>...</td>	00 00 00 00	...	00 00 00 00	...
20000050 <td>00 00 00 00</td> <td>...</td> <td>00 00 00 00</td> <td>...</td>	00 00 00 00	...	00 00 00 00	...
20000054 <td>00 00 00 00</td> <td>...</td> <td>00 00 00 00</td> <td>...</td>	00 00 00 00	...	00 00 00 00	...
20000058 <td>00 00 00 00</td> <td>...</td> <td>00 00 00 00</td> <td>...</td>	00 00 00 00	...	00 00 00 00	...
2000005C <td>00 00 00 00</td> <td>...</td> <td>00 00 00 00</td> <td>...</td>	00 00 00 00	...	00 00 00 00	...
20000060 <td>00 00 00 00</td> <td>...</td> <td>00 00 00 00</td> <td>...</td>	00 00 00 00	...	00 00 00 00	...
20000064 <td>00 00 00 00</td> <td>...</td> <td>00 00 00 00</td> <td>...</td>	00 00 00 00	...	00 00 00 00	...
20000068 <td>00 00 00 00</td> <td>...</td> <td>00 00 00 00</td> <td>...</td>	00 00 00 00	...	00 00 00 00	...
2000006C <td>00 00 00 00</td> <td>...</td> <td>00 00 00 00</td> <td>...</td>	00 00 00 00	...	00 00 00 00	...
20000070 <td>00 00 00 00</td> <td>...</td> <td>00 00 00 00</td> <td>...</td>	00 00 00 00	...	00 00 00 00	...
20000074 <td>00 00 00 00</td> <td>...</td> <td>00 00 00 00</td> <td>...</td>	00 00 00 00	...	00 00 00 00	...

CM3 Source Code - U1

```
main.c
/*
 * Name : main.c
 * Author : Mohamed Hassan
 * Created on : 9/20/2023
 * Description : Unit 3==> Lesson 3 ==> Lab(2)
 */
#include "Platform_Types.h"
#define RCC_BASE 0x40021000
#define PORTA_BASE 0x40010800
#define RCC_APB2ENR (*(volatile R_ODR_t *) (RCC_BASE + 0x18))
#define GPIO_CRH (*(volatile R_ODR_t *) (PORTA_BASE + 0x04))
#define GPIO_ODR (*(volatile R_ODR_t *) (PORTA_BASE + 0x0C))

typedef union
{
    uint32_t all_fields;
    struct
    {
        uint32_t reserved : 13;
        uint32_t pin13 : 1;
    } pin;
} R_ODR_t;

volatile R_ODR_t *R_ODR = (volatile R_ODR_t *) (PORTA_BASE + 0x0C);
uint32_t global_variables[] = {1,2,3};
uint32_t const_variables[] = {1,2,3};

extern void NMI_Handler(void)
{
    R_ODR->pin13 = 1;
    for(int i = 0; i < 5000; i++);
    R_ODR->pin13 = 0;
    for(int i = 0; i < 5000; i++);
}

return 0;

int main(void)
{
    RCC_APB2ENR |= (1<<2); //set pit 2
    GPIO_CRH |= 0xFF0FFFFF;
    GPIO_ODR |= 0x00200000; //set pins(20-24) 2

    while(1)
    {
        R_ODR->pin13 = 1;
        for(int i = 0; i < 5000; i++);
        R_ODR->pin13 = 0;
        for(int i = 0; i < 5000; i++);
    }
}
```

CM3 Variables - U1

Name	Address
g_variables	20000004
const_variables	08000128
ss_var	20000108
vectors	08000000
R_ODR	20000000
1	BP+12 = #20000FDC BP+16 = #20000FDB

CM3 Registers - U1

PC	0800005A	Privileged
Pr1	00_200	Mode Thread
R0	00000000	R7 20000FDB
R1	20000008	R8 00000000
R2	000013B7	R9 00000000
R3	0000024A	R10 00000000
R4	00000000	R11 00000000
R5	00000000	R12 00000000
R6	00000000	LR 0800010D
NSP	20000FDB	PSP 00000000
IRQ	0	EPSR 01000000
APSR	CNVZ	1c1t 000_00000

CM3 Source Code - U1

```
main.c
/*
 * Name : main.c
 * Author : Mohamed Hassan
 * Created on : 9/20/2023
 * Description : Unit 3==> Lesson 3 ==> Lab(2)
 */
#include "Platform_Types.h"
#define RCC_BASE 0x40021000
#define PORTA_BASE 0x40010800
#define RCC_APB2ENR (*(volatile R_ODR_t *) (RCC_BASE + 0x18))
#define GPIO_CRH (*(volatile R_ODR_t *) (PORTA_BASE + 0x04))
#define GPIO_ODR (*(volatile R_ODR_t *) (PORTA_BASE + 0x0C))

typedef union
{
    uint32_t all_fields;
    struct
    {
        uint32_t reserved : 13;
        uint32_t pin13 : 1;
    } pin;
} R_ODR_t;

volatile R_ODR_t *R_ODR = (volatile R_ODR_t *) (PORTA_BASE + 0x0C);
uint32_t global_variables[] = {1,2,3};
uint32_t const_variables[] = {1,2,3};

extern void NMI_Handler(void)
{
    R_ODR->pin13 = 1;
    for(int i = 0; i < 5000; i++);
    R_ODR->pin13 = 0;
    for(int i = 0; i < 5000; i++);
}

return 0;

int main(void)
{
    RCC_APB2ENR |= (1<<2); //set pit 2
    GPIO_CRH |= 0xFF0FFFFF;
    GPIO_ODR |= 0x00200000; //set pins(20-24) 2

    while(1)
    {
        R_ODR->pin13 = 1;
        for(int i = 0; i < 5000; i++);
        R_ODR->pin13 = 0;
        for(int i = 0; i < 5000; i++);
    }
}
```

CM3 FLASH at 0x08000000 - U1

08000000	08 10 00 20	...	08 08 01 40	...
08000004	AS 00 00 08	...	01 02 03 00	...
08000008	99 00 00 08	...	00 00 00 00	...
0800000C	99 00 00 08	...	00 00 00 00	...
08000010	99 00 00 08	...	00 00 00 00	...
08000014	99 00 00 08	...	00 00 00 00	...
08000018	99 00 00 08	...	00 00 00 00	...
0800001C	80 84 83 80	...	00 00 00 00	...
08000020	00 AF 14 48	...	00 00 00 00	...
08000024	18 68 18 4A	...	00 00 00 00	...
08000028	43 F0 04 03	...	00 00 00 00	...
0800002C	13 60 18 48	...	00 00 00 00	...
08000030	18 68 17 4A	...	00 00 00 00	...
08000034	23 F4 70 03	...	00 00 00 00	...
08000038	13 60 15 48	...	00 00 00 00	...
0800003C	18 68 14 4A	...	00 00 00 00	...
08000040	43 F4 00 13	...	00 00 00 00	...
08000044	13 60 13 48	...	00 00 00 00	...
08000048	1A 68 13 88	...	00 00 00 00	...
0800004C	43 F4 00 53	...	00 00 00 00	...
08000050	13 80 00 23	...	00 00 00 00	...
08000054	78 60 00 20	...	00 00 00 00	...
08000058	78 68 01 33	...	00 00 00 00	...
0800005C	78 60 78 68	...	00 00 00 00	...
08000060	41 F2 87 32	...	00 00 00 00	...
08000064	04 48 1A 68	...	00 00 00 00	...
08000068	13 88 6F F3	...	00 00 00 00	...
0800006C	40 33 13 80	...	00 00 00 00	...
08000070	00 23 38 60	...	00 00 00 00	...
08000074	00 23 38 60	...	00 00 00 00	...

CM3 RAM at 0x20000000 - U1

20000000	0C 08 01 40	...	0C 08 01 40	...
20000004	00 00 00 00	...	00 00 00 00	...
20000008	00 00 00 00	...	00 00 00 00	...
2000000C	00 00 00 00	...	00 00 00 00	...
20000010	00 00 00 00	...	00 00 00 00	...
20000014	00 00 00 00	...	00 00 00 00	...
20000018	00 00 00 00	...	00 00 00 00	...
2000001C	00 00 00 00	...	00 00 00 00	...
20000020	00 00 00 00	...	00 00 00 00	...
20000024	00 00 00 00	...	00 00 00 00	...
20000028 <td>00 00 00 00</td> <td>...</td> <td>00 00 00 00</td> <td>...</td>	00 00 00 00	...	00 00 00 00	...
2000002C	00 00 00 00	...	00 00 00 00	...
20000030	00 00 00 00	...	00 00 00 00	...
20000034	00 00 00 00	...	00 00 00 00	...
20000038	00 00 00 00	...	00 00 00 00	...
2000003C	00 00 00 00	...	00 00 00 00	...
20000040	00 00 00 00	...	00 00 00 00	...
20000044	00 00 00 00	...	00 00 00 00	...
20000048	00 00 00 00	...	00 00 00 00	...
2000004C	00 00 00 00	...	00 00 00 00	...
20000050	00 00 00 00	...	00 00 00 00	...
20000054	00 00 00 00	...	00 00 00 00	...
20000058	00 00 00 00	...	00 00 00 00	...
2000005C	00 00 00 00	...	00 00 00 00	...
20000060	00 00 00 00	...	00 00 00 00	...
20000064	00 00 00 00	...	00 00 00 00	...
20000068	00 00 00 00	...	00 00 00 00	...
2000006C	00 00 00 00	...	00 00 00 00	...
20000070	00 00 00 00	...	00 00 00 00	...
20000074	00 00 00 00	...	00 00 00 00	...

CM3 Source Code - U1

```
main.c
/*
 * Name : main.c
 * Author : Mohamed Hassan
 * Created on : 9/20/2023
 * Description : Unit 3==> Lesson 3 ==> Lab(2)
 */
#include "Platform_Types.h"
#define RCC_BASE 0x40021000
#define PORTA_BASE 0x40010800
#define RCC_APB2ENR (*(volatile R_ODR_t *) (RCC_BASE + 0x18))
#define GPIO_CRH (*(volatile R_ODR_t *) (PORTA_BASE + 0x04))
#define GPIO_ODR (*(volatile R_ODR_t *) (PORTA_BASE + 0x0C))

typedef union
{
    uint32_t all_fields;
    struct
    {
        uint32_t reserved : 13;
        uint32_t pin13 : 1;
    } pin;
} R_ODR_t;

volatile R_ODR_t *R_ODR = (volatile R_ODR_t *) (PORTA_BASE + 0x0C);
uint32_t global_variables[] = {1,2,3};
uint32_t const_variables[] = {1,2,3};

extern void NMI_Handler(void)
{
    R_ODR->pin13 = 1;
    for(int i = 0; i < 5000; i++);
    R_ODR->pin13 = 0;
    for(int i = 0; i < 5000; i++);
}

return 0;

int main(void)
{
    RCC_APB2ENR |= (1<<2); //set pit 2
    GPIO_CRH |= 0xFF0FFFFF;
    GPIO_ODR |= 0x00200000; //set pins(20-24) 2

    while(1)
    {
        R_ODR->pin13 = 1;
        for(int i = 0; i < 5000; i++);
        R_ODR->pin13 = 0;
        for(int i = 0; i < 5000; i++);
    }
}
```