

# Lab 2

## Description:

Create a bare-metal Software to toggle a LED on Stm32f103CX board.

## Baremetal toggle led

- ▶ Led is connected to GPIO port C13
- ▶ To make a GPIO toggling in STM32,
- ▶ you need to work with two peripherals:
  - ▶ RCC (reset and clock control) and
  - ▶ GPIOx (general purpose input/output).
- ▶ The RCC is necessary because the **GPIO has disabled clock by default.**

Bare metal SW



<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>

Eng. Keroles Shenouda

Embedded C (Mastering Embedded System online Diploma)

Eng.keroles.karam@gmail.com

## Files Created:

- main.c
- linker\_script.ld
- startup.c
- startup.s
- Makefile

## Executable Files:

- learn-in-depth\_cortex\_m3.elf
- learn-in-depth\_cortex\_m3.bin

## Analysis Files:

- main.o
- Map\_file.map
- startup.o

```
main.c  Makefile — lesson3  Map_file.map
14  Linker script and memory map
15
16
17  .text ..... 0x08000000 ..... 0xd4
18  *(.vectors*)
19  .vectors ..... 0x08000000 ..... 0x50 startup.o
20  *(.text*)
21  .text ..... 0x08000050 ..... 0x7c main.o
22  ..... 0x08000050 ..... main
23  .text ..... 0x080000cc ..... 0x8 startup.o
24
25  .rodata ..... 0x080000d4 ..... 0x3
26  .rodata ..... 0x080000d4 ..... 0x3 main.o
27  ..... 0x080000d4 ..... const_variable
28
29  .data ..... 0x080000d8 ..... 0x7
30  .data ..... 0x080000d8 ..... 0x7 main.o
31  ..... 0x080000d8 ..... R_ODR
32  ..... 0x080000dc ..... g_variables
33  .data ..... 0x080000df ..... 0x0 startup.o
34
35  .igot.plt ..... 0x080000e0 ..... 0x0
36  .igot.plt ..... 0x080000e0 ..... 0x0 main.o
37
38  .glue_7 ..... 0x080000e0 ..... 0x0
39  .glue_7 ..... 0x080000e0 ..... 0x0 linker stubs
40
41  .glue_7t ..... 0x080000e0 ..... 0x0
42  .glue_7t ..... 0x080000e0 ..... 0x0 linker stubs
43
44  .vfp11_veneer ..... 0x080000e0 ..... 0x0
45  .vfp11_veneer ..... 0x080000e0 ..... 0x0 linker stubs
46
47  .v4_bx ..... 0x080000e0 ..... 0x0
48  .v4_bx ..... 0x080000e0 ..... 0x0 linker stubs
49
50  .iplt ..... 0x080000e0 ..... 0x0
51  .iplt ..... 0x080000e0 ..... 0x0 main.o
52
53  .rel.dyn ..... 0x080000e0 ..... 0x0
54  .rel.iplt ..... 0x080000e0 ..... 0x0 main.o
55
56  .bss ..... 0x20000000 ..... 0x3
57  *(.bss*)
58  .bss ..... 0x20000000 ..... 0x0 main.o
59  .bss ..... 0x20000000 ..... 0x0 startup.o
60  COMMON ..... 0x20000000 ..... 0x3 main.o
61  ..... 0x20000000 ..... bss_var
```

## Entry Point:

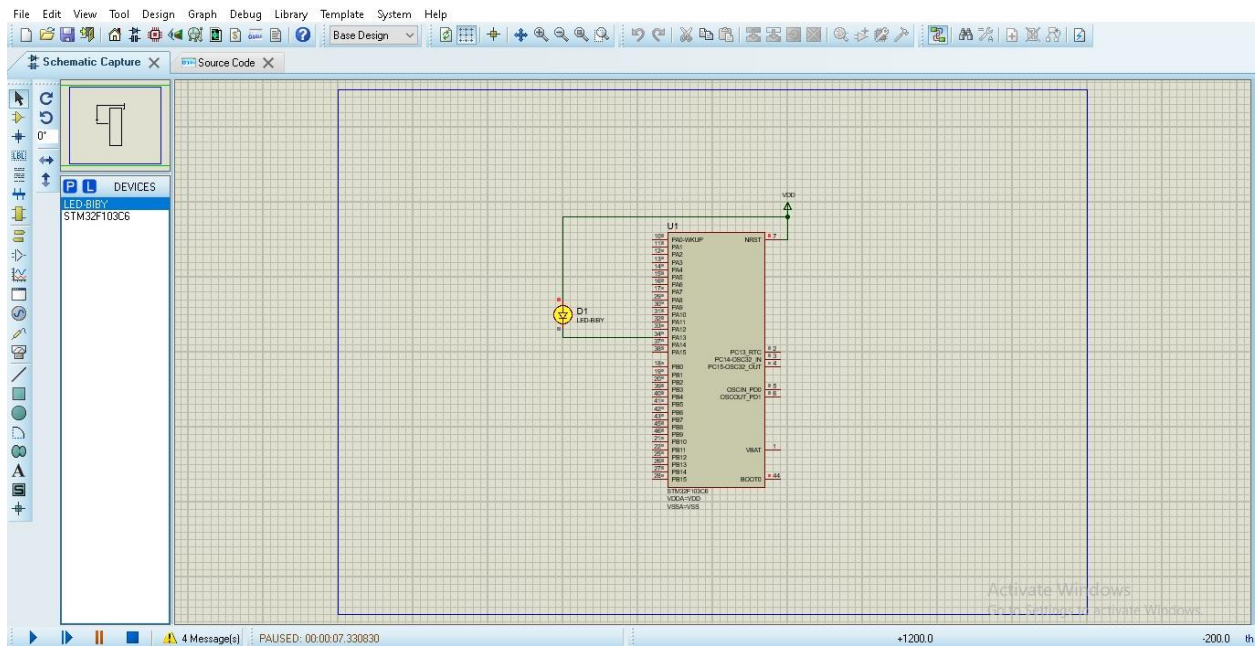
```
Sections:
Idx Name          Size      VMA       LMA       File off  Algn
  0 .text          0000007c  00000000  00000000  00000034  2**2
    CONTENTS, ALLOC, LOAD, RELOC, READONLY, CODE
  1 .data          00000007  00000000  00000000  000000b0  2**2
    CONTENTS, ALLOC, LOAD, DATA
  2 .bss           00000000  00000000  00000000  000000b7  2**0
    ALLOC
  3 .rodata        00000003  00000000  00000000  000000b8  2**2
    CONTENTS, ALLOC, LOAD, READONLY, DATA
  4 .debug_info    0000018b  00000000  00000000  000000bb  2**0
    CONTENTS, RELOC, READONLY, DEBUGGING
  5 .debug_abbrev  000000f8  00000000  00000000  00000246  2**0
    CONTENTS, READONLY, DEBUGGING
  6 .debug_loc     00000038  00000000  00000000  0000033e  2**0
    CONTENTS, READONLY, DEBUGGING
  7 .debug_aranges 00000020  00000000  00000000  00000376  2**0
    CONTENTS, RELOC, READONLY, DEBUGGING
  8 .debug_line    0000012b  00000000  00000000  00000396  2**0
    CONTENTS, RELOC, READONLY, DEBUGGING
  9 .debug_str     000001cc  00000000  00000000  000004c1  2**0
    CONTENTS, READONLY, DEBUGGING

Class:              ELF32
Data:               2's complement, little endian
Version:            1 (current)
OS/ABI:             UNIX - System V NG
ABI Version:        0               000738  2**0
Type:               EXEC (Executable file)
Machine:            ARM
Version:            0x1Downloads/Embedded Here We Go Again/Kero
Entry point address: 0x80000000      lesson3 (master)
Start of program headers: 52 (bytes into file)
Start of section headers: 68308 (bytes into file)
Flags:              0x5000200, Version5 EABI, soft-float ABI
Size of this header: 52 (bytes)      erolos Shenoda's Diploma/Code,
Size of program headers: 32 (bytes)
Number of program headers: 2
Size of section headers: 40 (bytes)00 00 00 00
Number of section headers: 17
Section header string table index: 16

Section Headers:
[Nr] Name          Type          Addr      Off      Size    ES Flg Lk Inf Al
[ 0]              NULL          00000000  000000  000000  00   0  0  0  0
[ 1] .text          PROGBITS     08000000  010000  0000d4  00   AX  0  0  4
[ 2] .rodata        PROGBITS     080000d4  0100d4  000003  00   A  0  0  4
[ 3] .data          PROGBITS     080000d8  0100d8  000007  00   WA  0  0  4
[ 4] .bss           NOBITS       20000000  020000  000003  00   WA  0  0  4
[ 5] .debug_info    PROGBITS     00000000  0100df  0001b1  00   0  0  0  1
[ 6] .debug_abbrev  PROGBITS     00000000  010290  00010c  00   0  0  0  1
```

# Proteus Simulation

Simulation working:



Debug information:

File Project Build Edit Debug System Help

Schematic Capture Source Code

### CM3 Source Code - U1

main.c

```

----- #define GPIOA_BASE      0x40010800
----- //registers
----- #define RCC_APB2ENR      *(volatile uint32_t *) (RCC_BASE + 0x18)
----- #define GPIOA_CRH        *(volatile uint32_t *) (GPIOA_BASE + 0x04)
----- #define GPIOA_ODR        *(volatile uint32_t *) (GPIOA_BASE + 0x0C)
-----
----- typedef union{
-----     uint32_t all_fields;
-----     struct{
-----         uint32_t reserved:13;
-----         uint32_t P13:1;
-----     } Pin;
----- } R_ODR_t;
-----
----- volatile R_ODR_t* R_ODR = (volatile uint32_t *) (GPIOA_BASE + 0x0C);
----- unsigned char g_variables [3] = {1,2,3}; //To increase the data section for learning purpose
----- unsigned char const const_variable [3] = {1,2,3}; //For creating the rodata section for learning purpose
----- unsigned char bss_var[3]; //bss section
-----
----- int main(void)
8000050 {
8000056     RCC_APB2ENR |= 1<<2;
8000062     GPIOA_CRH = 0xfffffff;
800006E     GPIOA_CRH = 0x00200000;
-----     int i;
-----
-----     while(1){
800007A         R_ODR->Pin.P13 = 1;
8000086         for(i = 0; i < 5000;i++);
800009C         R_ODR->Pin.P13 = 0;
80000A8         for(i = 0; i < 5000;i++);
-----     }
----- }

```

### CM3 Variables - U1

Name	Address	Value
g_variables	080000DC	byte[3]
const_variable	080000D4	byte[3]
bss_var	20000000	byte[3]
R_ODR	080000D8	0x4001080C
i	BP+12 = #20000FD4	1006

3 Message(s) PAUSED: 00:00:01.02657

## Ram and Flash Memories:

File Project Build Edit Debug System Help

Schematic Capture Source Code

### CM3 Source Code - U1

main.c

CM3 RAM at 0:20000000 - U1

CM3 FLASH at 0:08000000 - U1

Address	Value
00000000	55 31 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000010	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000020	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000030	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000040	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000050	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000060	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000070	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000080	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000090	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000A0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000B0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000C0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000D0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000E0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000F0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000100	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000110	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000120	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000130	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000140	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000150	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000160	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000170	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000180	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000190	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000001A0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000001B0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000001C0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000001D0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000001E0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000001F0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000200	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

### CM3 Variables - U1

Name	Address	Value
bss_var[2]	20000002	'\0'
R_ODR	080000D8	0x4001080C
R_ODR.all_fields	4001080C	0x00 0x00 0x00 0x00
R_ODR.Pin	4001080C	0
i	BP+12 = #20000FD4	1040

4 Message(s) PAUSED: 00:00:05.303316



## Variables:

Name	Address	Value
g_vaiables	0800000C	byte[3]
g_vaiables[0]	0800000C	0x01
g_vaiables[1]	0800000D	0x02
g_vaiables[2]	0800000E	0x03
const_variable	08000004	byte[3]
const_variable[0]	08000004	0x01
const_variable[1]	08000005	0x02
const_variable[2]	08000006	0x03
bss_var	20000000	byte[3]
bss_var[0]	20000000	'0'
bss_var[1]	20000001	'1'
bss_var[2]	20000002	'\0'
IR_ODR	08000008	0x4001080C
*IR_ODR	4001080C	0x00 0x00 0x00 0x00
all_fields	4001080C	0
Pin	4001080C	0x00 0x00 0x00 0x00
reserved	4001080C	0
P13	4001080C	0
i	BP+12 = 020000F04	1040

## Registers:

PC	080000AE	Privileged
Pri	FE.FFFFFFF0	Mode Handler
R0	00000000	R7 20000FD0
R1	00000000	R8 00000000
R2	00001387	R9 00000000
R3	00000598	R10 00000000
R4	00000000	R11 00000000
R5	00000000	R12 00000000
R6	00000000	LR 08000001
MSP*	20000FD0	PSP 00000000
IRQ	3	
APSR	CNQVZ	EPSR 01000000
	01000	iciit 000.00000