

Lab_3

- ✓ In this lab I want to toggle led on cortex m4 without using IDE just terminal, so I write startup code, linker script and main code (find them in my repo).
- ✓ After written codes should compile them using terminal. This screen defines that compilation done successfully.

```
MINGW32/d/Embedded System dip_KS/Untis_3_Embedded_C/Lesson_4_Lab_3
o
arm-none-eabi-ld.exe -T linker_script.ld Startup.o main.o -o Toggle_LED_cortex_m4.elf -Map=Map_file.map
cp Toggle_LED_cortex_m4.elf Toggle_LED_cortex_m4.axf
arm-none-eabi-objcopy.exe -O binary Toggle_LED_cortex_m4.elf Toggle_LED_cortex_m4.bin
=== Bulid is done ===

ibrahim@DESKTOP-PF9T1AH MINGW32 /d/Embedded System dip_KS/Untis_3_Embedded_C/Lesson_4_Lab_3
$ export PATH=/c/ST/STM32CubeIDE_1.4.0/STM32CubeIDE/plugins/com.st.stm32cube.ide.mcu.externaltools.gnu-tools-for-stm32.7-2018-q2-update.win32_1.4.0.202007081208/tools/bin:$PATH

ibrahim@DESKTOP-PF9T1AH MINGW32 /d/Embedded System dip_KS/Untis_3_Embedded_C/Lesson_4_Lab_3
$ mingw32-make.exe
arm-none-eabi-gcc.exe -c -I . -gdwarf-2 -mcpu=cortex-m4 -g main.c -o main.o
arm-none-eabi-ld.exe -T linker_script.ld Startup.o main.o -o Toggle_LED_cortex_m4.elf -Map=Map_file.map
cp Toggle_LED_cortex_m4.elf Toggle_LED_cortex_m4.axf
arm-none-eabi-objcopy.exe -O binary Toggle_LED_cortex_m4.elf Toggle_LED_cortex_m4.bin
=== Bulid is done ===

ibrahim@DESKTOP-PF9T1AH MINGW32 /d/Embedded System dip_KS/Untis_3_Embedded_C/Lesson_4_Lab_3
$ |
```

✓ Now I burn my code in visual board by using Keil uVision5.

Keil uVision5 interface showing the code for toggling an LED. The code is in `main.c` and includes a delay loop.

```
14 int main()
15 {
16     volatile unsigned long delay_count;
17     SYSCTL_RCGC2_R = 0x20;
18
19     // wait to make sure that GPIO is up.
20     for(delay_count=0; delay_count < 200; delay_count ++);
21
22     GPIO_PORTF_DIR_R |= 1<<3; // making DIR is output at bit 3 in PORTF.
23     GPIO_PORTF_DEN_R |= 1<<3;
24
25     while(1)
26     {
27         GPIO_PORTF_DATA_R |= 1<<3;
28         for(delay_count=0; delay_count < 200000; delay_count ++);
29         GPIO_PORTF_DATA_R ^= 1<<3;
30         for(delay_count=0; delay_count < 200000; delay_count ++);
31     }
32 }
33 return 0;
34 }
```

The code is being executed in the debugger. A breakpoint is set at line 28, after the LED is turned on. The LED is shown as ON in the hardware view.

Here I put breakpoint after set LED 1 to see it ON.

LED is ON

Non-physical board

✓

D:\Embedded System dip_KS\Embedded C\lesson_4\Keil_uvision_unit3_lab4_project\Keil_uvision_unit3_lab4_project.uvprojx - µVision [Non-Commercial Use License]

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help

Registers Disassembly

Register	Value
R0	0x00000000
R1	0x00000000
R2	0x400253FC
R3	0x00000011
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x200003C8
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x200003C8
R14 (LR)	0x00000085
R15 (PC)	0x00000102
xPSR	0x21000000

Startup.c main.c

```
14 int main()
15 {
16     volatile unsigned long delay_count;
17     SYSCTL_RCGC2_R = 0x20;
18
19     // wait to make sure that GPIO is up.
20     for(delay_count=0; delay_count <200; delay_count ++);
21
22     GPIO_PORTF_DIR_R |= 1<<3; // making DIR is output at bit 3 in PORTF.
23     GPIO_PORTF_DEN_R |= 1<<3;
24
25     while(1)
26     {
27         GPIO_PORTF_DATA_R |= 1<<3;
28         for(delay_count=0; delay_count <200000; delay_count ++);
29         GPIO_PORTF_DATA_R &= ~(1<<3);
30         for(delay_count=0; delay_count <200000; delay_count ++);
31     }
32
33     return 0;
34 }
35
36
37
38
```

Put breakpoint after clear LED 0 to can see it OFF.

TEKSA edX Lab 2

Port F Hardware

Port F Registers

Register	Value
DATA	0x11
DIR	0x08
DEN	0x08
PUR	0x00
PDR	0x00
RCGC2	0x00000020
LOCK	0x01
CR	0x1E

Grading Controls

Number from edX: Grade Score: 0

Copy this to edX:

Command

```
BS \\Toggle_LED_cortex_m4\\main.c\\29
BS \\Toggle_LED_cortex_m4\\main.c\\31
```

Call Stack + Locals

Name	Location/Value	Type
main	0x000000A0	int f()
delay_count	0x00030D40	auto - uint

Simulation

tt: 22.58913875 sec L31 C:1 CAP: NUM: SCRL: OVR: R/W

LED is OFF

