LAB 2

- ## Main.c code

```c
//@Copyright: Arasny

#include"stdint.h"
#define RCC_BASE            0x40021000
#define GPIOA_BASE          0x40010800
#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)

unsigned char g_variabled[3]={1,2,3};
unsigned char const const_var[3]={1,2,3};
unsigned char bss_var[3];

int main(void)
{
    int i;
    RCC_APB2ENR |= (1<<2);
    GPIOA_CRH   &= 0xFF0FFFFF;
    GPIOA_CRH   |= 0x00200000;
    while(1)
    {
        GPIOA_ODR |=1<<13;
        for(i=0;i<5000;i++);
        GPIOA_ODR &= ~(1<<13);
        for(i=0;i<5000;i++);
    }
}
```

- **Startup.c code**

```c
//startup.c
//End.Arsany
#include"stdint.h"
extern int main();
void Reset_Handler();
void Default_Handler()
{
    Reset_Handler();
}
void NMI_Handler() __attribute__((weak,alias("Default_Handler")));
void H_Fault_Handler()__attribute__((weak,alias("Default_Handler")));
void MM_Fault_Handler()__attribute__((weak,alias("Default_Handler")));
void Bus_Fault_Handler()__attribute__((weak,alias("Default_Handler")));
void Usage_Fault_Handler()__attribute__((weak,alias("Default_Handler")));

extern unsigned int _stack_top;

uint32_t vectors[] __attribute__((section(".vectors"))) = {
    (uint32_t) &_stack_top,
    (uint32_t) &Reset_Handler,
    (uint32_t) &NMI_Handler,
    (uint32_t) &H_Fault_Handler,
    (uint32_t) &MM_Fault_Handler,
    (uint32_t) &Bus_Fault_Handler,
    (uint32_t) &Usage_Fault_Handler
};


extern unsigned int _S_DATA;
extern unsigned int _E_DATA;
extern unsigned int _S_bss;
extern unsigned int _E_bss;
extern unsigned int _E_text;
int i;


void Reset_Handler()
{
    unsigned int DATA_SIZE = (unsigned char*)&_E_DATA - (unsigned char*)&_S_DATA;
    unsigned char* P_src = (unsigned char*)&_E_text;
    unsigned char* P_dst = (unsigned char*)&_S_DATA;
    for( i=0;i<DATA_SIZE;i++)
    {
        *((unsigned char*)P_dst++)=*((unsigned char*)P_src);
    }
    unsigned int bss_SIZE = (unsigned char*)&_E_bss - (unsigned char*)&_S_bss;
    P_dst=(unsigned char*)&_S_bss;
    for( i=0;i<DATA_SIZE;i++)
    {
        *((unsigned char*)P_dst++)=(unsigned char*)0;
    }
    main();
}
```

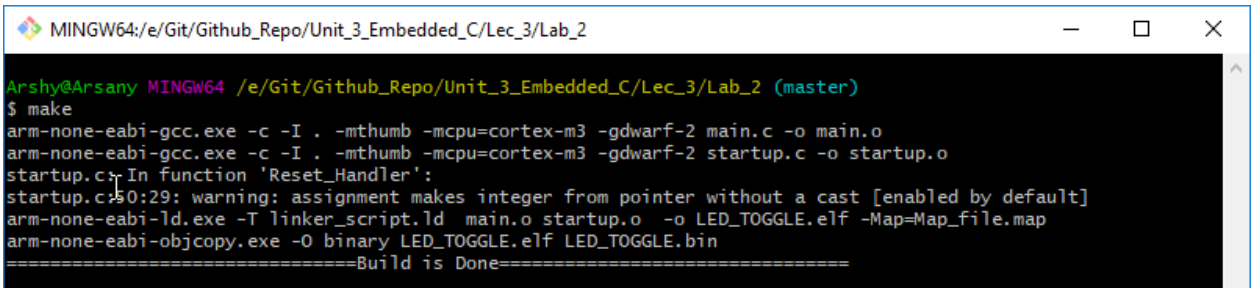
- Linker_script.ld

```
MEMORY
{
    flash(RX) : ORIGIN = 0x08000000, LENGTH = 128k
    sram(RWX) : ORIGIN = 0x20000000, LENGTH = 20k
}

SECTIONS
{
    .text : {
            *(.vectors*)
            *(.text*)
            *(.rodata*)
            _E_text = . ;
    }> flash

    .data : {
            _S_DATA = . ;
            *(.data)
            _E_DATA = . ;

    }> sram AT> flash

    .bss : {
            _S_bss = . ;
            *(.bss)
            _E_bss = . ;
            . = ALIGN(4) ;
            . = . + 0x1000 ;
            _stack_top = . ;
    }>sram
}
```

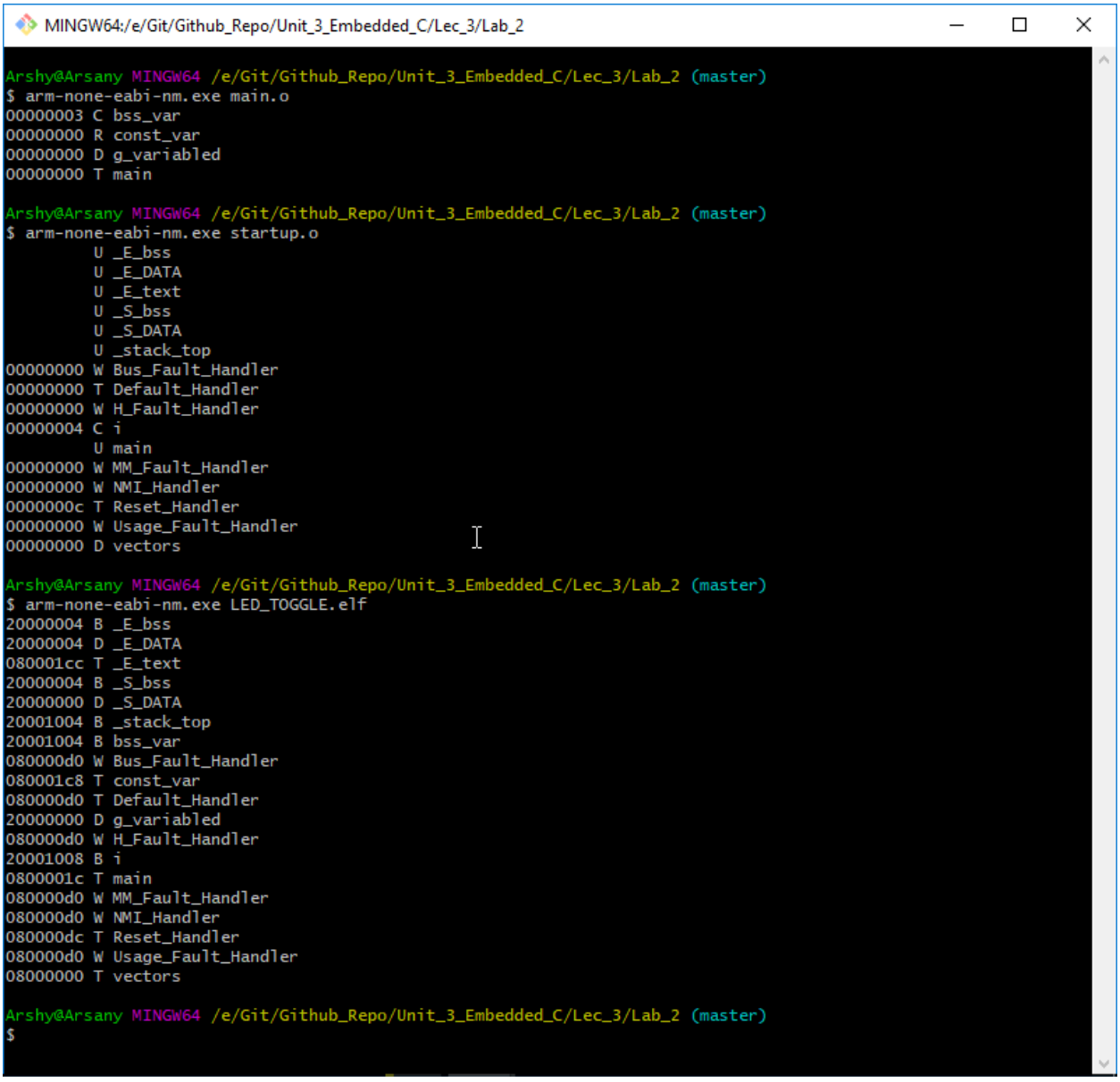
- Modifying the makefile

```
#@copyright : Arsany
CC=arm-none-eabi-
CFLAGS=-mthumb -mcpu=cortex-m3 -gdwarf-2
INCS=-I .
LIBS=
SRC = $(wildcard *.c)
OBJ = $(SRC:.c=.o)
As = $(wildcard *.s)
AsOBJ = $(As:.s=.o)

Project_name=LED_TOGGLE

all: $(Project_name).bin
    @echo "==============================Build is Done=============================="

%.o: %.c
    $(CC)gcc.exe -c $(INCS) $(CFLAGS) $< -o $@

$(Project_name).elf: $(OBJ) $(AsOBJ)
    $(CC)ld.exe -T linker_script.ld $(LIBS) $(OBJ) $(AsOBJ) -o $@ -Map=Map_file.map

$(Project_name).bin: $(Project_name).elf
    $(CC)objcopy.exe -O binary $< $@

clean_all:
    rm *.o *.elf *.bin *.map
    @echo "================================CLEAN================================"
clean:
    rm *.elf *.bin *.map
```

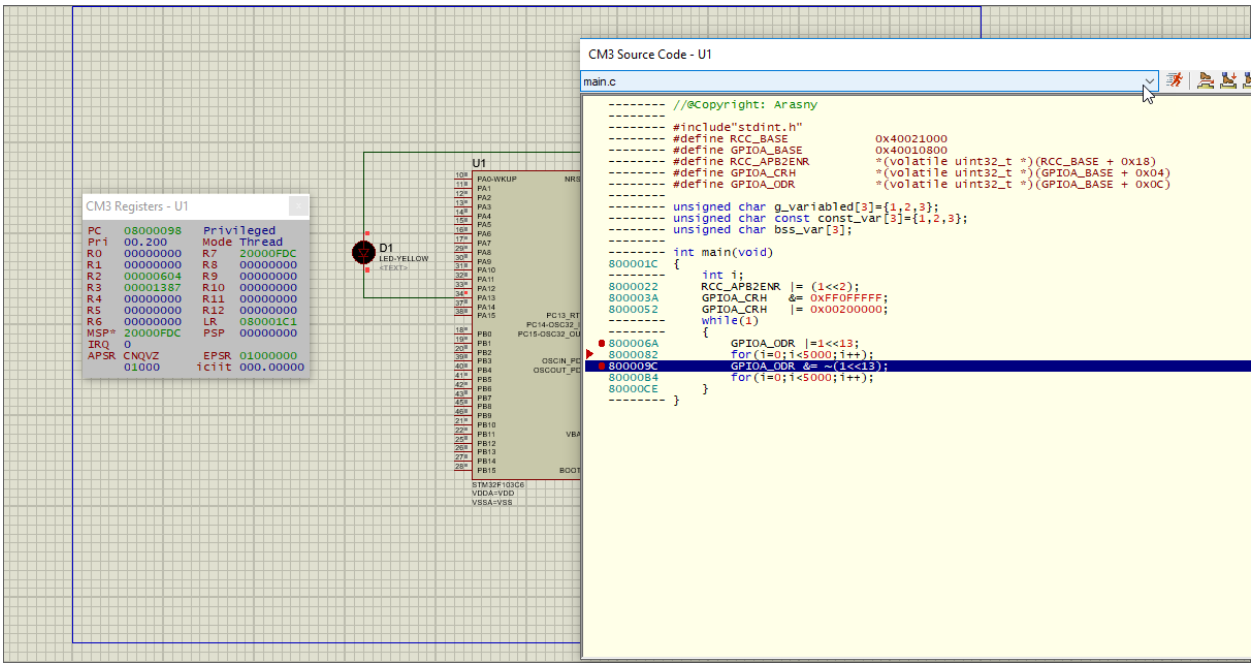
- Building Process

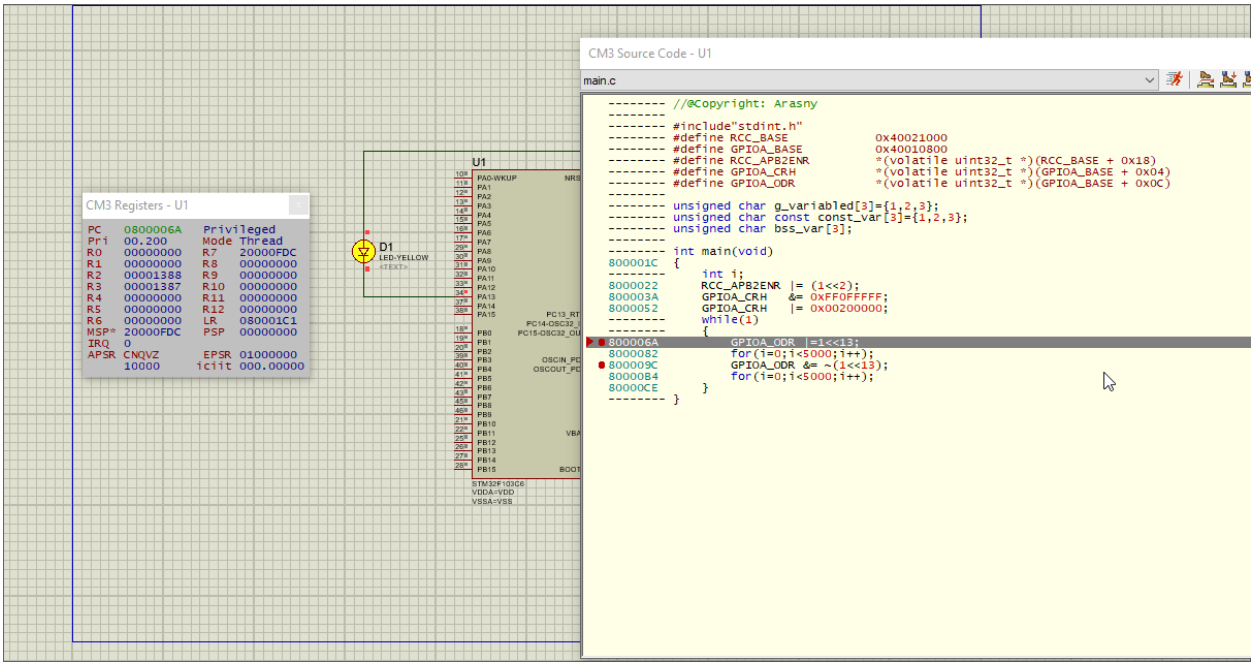


- Symbols



- Testing


CM3 Source Code - U1
main.c
CM3 Registers - U1
PC 08000098 Privileged
Pri 00.200 Mode Thread
R0 00000000 R7 20000FDC
R1 00000000 R8 00000000
R2 00000604 R9 00000000
R3 00001387 R10 00000000
R4 00000000 R11 00000000
R5 00000000 R12 00000000
R6 00000000 LR 080001C1
MSP* 20000FDC PSP 00000000
IRQ 0
APSR CNQVZ EPSR 01000000
01000 icIit 000.00000
U1
D1
LED-YELLOW
-------- //@Copyright: Arasny
-------- #include"stdint.h"
-------- #define RCC_BASE 0x40021000
-------- #define GPIOA_BASE 0x40010800
-------- #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)
-------- unsigned char g_variabled[3]={1,2,3};
-------- unsigned char const const_var[3]={1,2,3};
-------- unsigned char bss_var[3];
-------- int main(void)
800001C {
-------- int i;
8000022 RCC_APB2ENR |= (1<<2);
800003A GPIOA_CRH &= 0XFF0FFFFF;
8000052 GPIOA_CRH |= 0x00200000;
-------- while(1)
-------- {
800006A GPIOA_ODR |=1<<13;
8000082 for(i=0;i<5000;i++);
800009C GPIOA_ODR &= ~(1<<13);
80000B4 for(i=0;i<5000;i++);
80000CE }
-------- }


CM3 Source Code - U1
main.c
CM3 Registers - U1
PC 0800006A Privileged
Pri 00.200 Mode Thread
R0 00000000 R7 20000FDC
R1 00000000 R8 00000000
R2 00001388 R9 00000000
R3 00001387 R10 00000000
R4 00000000 R11 00000000
R5 00000000 R12 00000000
R6 00000000 LR 080001C1
MSP* 20000FDC PSP 00000000
IRQ 0
APSR CNQVZ EPSR 01000000
10000 icIit 000.00000
U1
D1
LED-YELLOW
-------- //@Copyright: Arasny
-------- #include"stdint.h"
-------- #define RCC_BASE 0x40021000
-------- #define GPIOA_BASE 0x40010800
-------- #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)
-------- unsigned char g_variabled[3]={1,2,3};
-------- unsigned char const const_var[3]={1,2,3};
-------- unsigned char bss_var[3];
-------- int main(void)
800001C {
-------- int i;
8000022 RCC_APB2ENR |= (1<<2);
800003A GPIOA_CRH &= 0XFF0FFFFF;
8000052 GPIOA_CRH |= 0x00200000;
-------- while(1)
-------- {
800006A GPIOA_ODR |=1<<13;
8000082 for(i=0;i<5000;i++);
800009C GPIOA_ODR &= ~(1<<13);
80000B4 for(i=0;i<5000;i++);
80000CE }
-------- }