

Міністерство освіти і науки України

Національний технічний університет України

«Київський політехнічний інститут ім. Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

ЛАБОРАТОРНА РОБОТА №1.2

З дисципліни «Архітектура комп'ютерів-3. Мікропроцесорні засоби»

ВИКОНАВ:

Студент III курсу ФІОТ

Групи ІВ-81

Болотов Г. В.

ПЕРЕВІРИВ:

Нікольський С. С.

Київ-2021

Вміст директорії

```
(base) heorhii@heorhii:~/AK3/lab2$ ls
firmware.bin  lab1.o  lscript.ld  start.o
firmware.elf  lab1.S  Makefile    start.S
```

lscript.ld

```
MEMORY
{
    FLASH ( rx )          : ORIGIN = 0x08000000, LENGTH = 1M
    RAM ( rxw )           : ORIGIN = 0x20000000, LENGTH = 128K
}
```

Makefile

```
(base) heorhii@heorhii:~/AK3/lab2$ cat Makefile
SDK_PREFIX?=arm-none-eabi-
CC = $(SDK_PREFIX)gcc
LD = $(SDK_PREFIX)ld
SIZE = $(SDK_PREFIX)size
OBJCOPY = $(SDK_PREFIX)objcopy
QEMU = qemu-system-gnuarmelipse
BOARD ?= STM32F4-Discovery
MCU=STM32F407VG
TARGET=firmware
CPU_CC=cortex-m4
TCP_ADDR=1234
deps = \
    start.S \
    lscript.ld
all: target
target:
    $(CC) -x assembler-with-cpp -c -O0 -g3 -mcpu=$(CPU_CC) -Wall start.S -o start.o
    $(CC) -x assembler-with-cpp -c -O0 -g3 -mcpu=$(CPU_CC) -Wall lab1.S -o lab1.o
    $(CC) start.o lab1.o -mcpu=$(CPU_CC) -Wall --specs=nosys.specs -nostdlib -l
gcc -T./lscript.ld -o $(TARGET).elf
    $(OBJCOPY) -O binary -F elf32-littlearm $(TARGET).elf $(TARGET).bin
qemu:
    $(QEMU) --verbose --verbose --board $(BOARD) --mcu $(MCU) -d unimp,guest_er
rors --image $(TARGET).bin --semihosting-config enable=on,target=native -gdb tcp::$
(TCP_ADDR) -S
clean:
    -rm *.o
    -rm *.elf
    -rm *.bin
flash:
    st-flash write $(TARGET).bin 0x08000000
```

start.S

```
(base) heorhii@heorhii:~/AK3/lab2$ cat start.S
.syntax unified
.cpu cortex-m4
//.fpu softvfp
.thumb

// Global memory locations.
.global vtable
.global __hard_reset__

/*
 * vector table
 */
.type vtable, %object
.type __hard_reset__, %function
vtable:
    .word __stack_start
    .word __hard_reset__+1
    .size vtable, .-vtable
__hard_reset__:
// initialize stack here
// if not initialized yet
    bl lab1
    _loop: b _loop
    .size __hard_reset__, .-__hard_reset__
```

lab1.S

```
.global lab1
.syntax unified

#define A #1
#define B #3
#define C #4

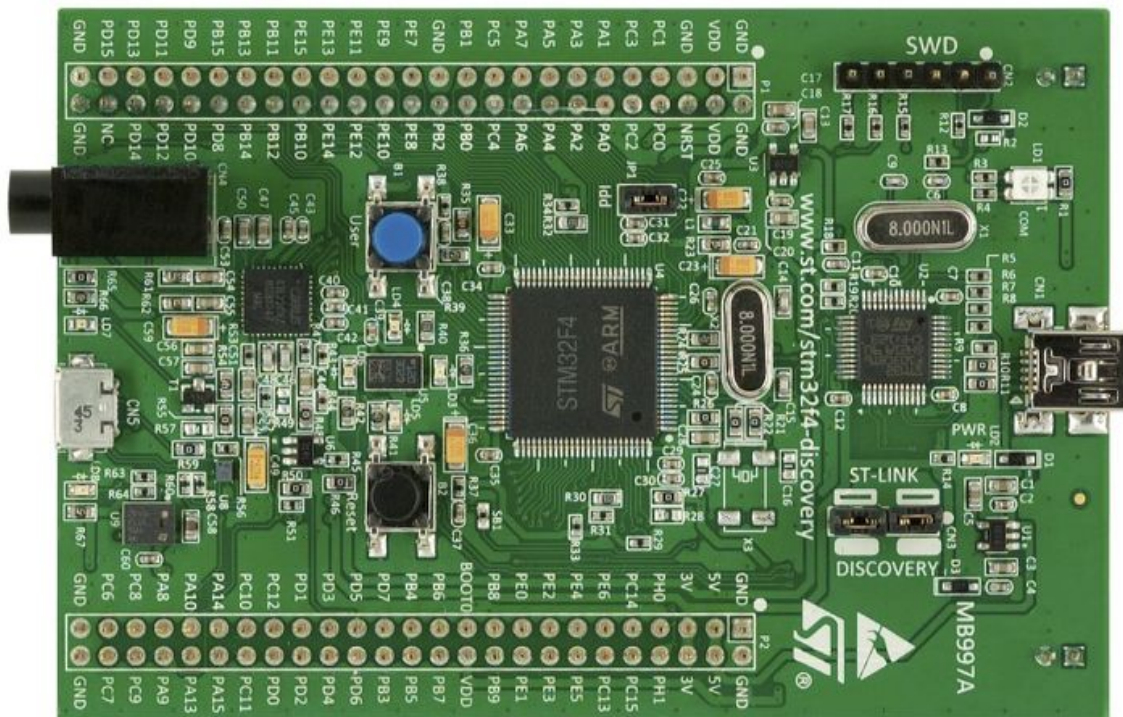
lab1:
    push {lr}
    mov r0, A
    add r0, B
    lsr r1, r0, #1 // r1 = (a+b)/2
    mov r0, #1
    mov r2, C

    .factorial:
        mul r0, r2
        subs r2, #1
        bne .factorial // r0 = c!
    add r0, r1
    pop {pc}
```

Збирання проекту

```
(base) heorhii@heorhii:~/AK3/lab2$ make qemu
qemu-system-gnuarmelipse --verbose --verbose --board STM32F4-Discovery --mcu STM32F407VG -d unimp,guest_errors --image firmware.bin --semihosting-config enable=on,target=native -gdb tcp::1234 -S
```

ST Discovery kit for STM32F407/417 lines



```
['/machine/mcu/stm32/USART6', address: 0x40011400, size: 0x0400
```

gdb

```
heorhii@heorhii: ~/AK3/lab2
Register group: general
r0      0x1a      26
r1      0x2       2
r2      0x0       0
r3      0x0       0
r4      0x0       0
r5      0x0       0

0x800002a <.factorial+6>    bne.n    0x8000024 <.factorial>
0x800002c <.factorial+8>    add      r0, r1
>0x800002e <.factorial+10>  pop      {pc}
0x8000030                movs     r0, r0
0x8000032                movs     r0, r0
0x8000034                movs     r0, r0

extended-r Thread 1 In: factorial      L21    PC: 0x800002e
(gdb) step
(gdb) step
(gdb) step
(gdb) step
(gdb) step
(gdb) step
(gdb) step
(gdb) step
(gdb) 
```