GEBZE TECHNICAL UNIVERSITY

ELECTRONIC ENGINEERING

ELM335

MICROPROCESSORS LAB

LAB 1 Report

Prepared by
171024086 Berat KIZILARMUT

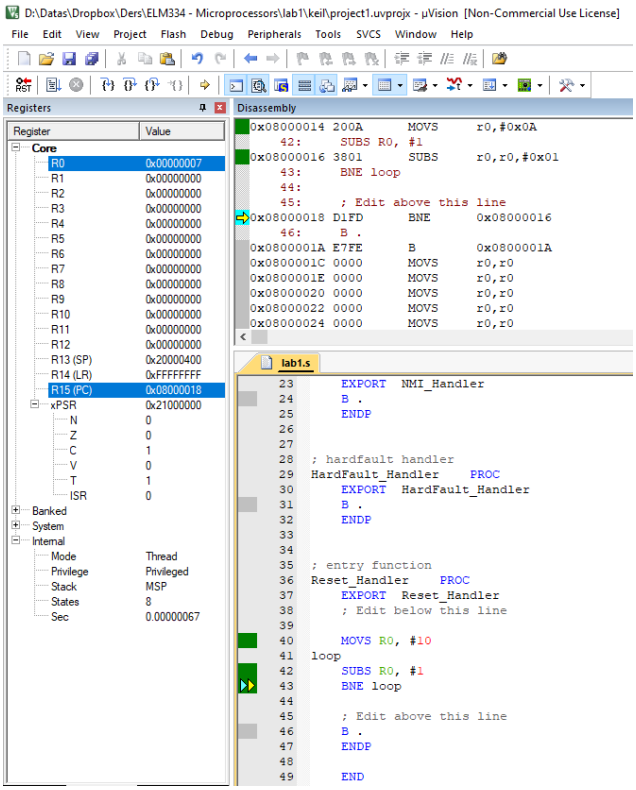


Figure 1

1. Introduction

Goal of this lab is to get familiarized with the new environment of the developer board. All the necessary steps have been followed to install the necessary software and test the provided example code. Debugging exercise is completed, given Figure 1.

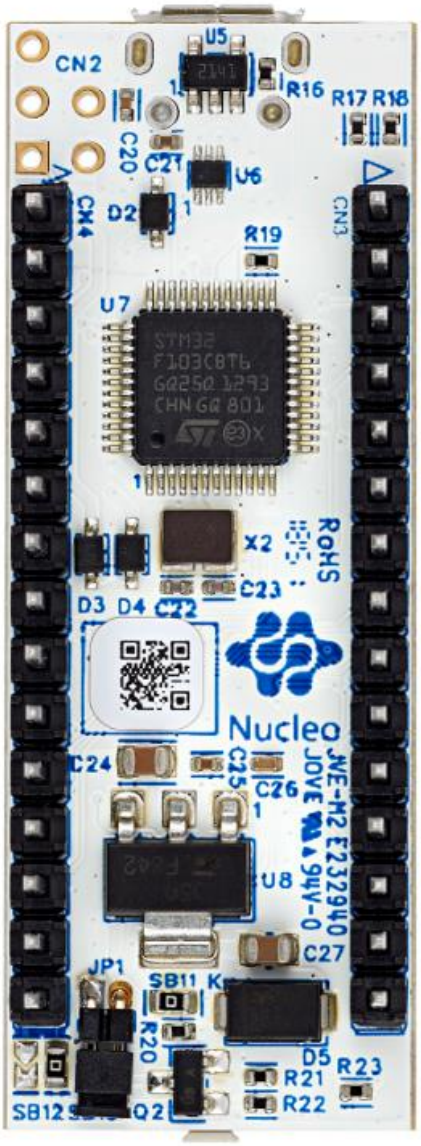
2. Problem 1

Since I don't have the board itself at this very moment, I resorted to the "STM32G0 Nucleo-32 board User Manual", and 3D PCBDOC.

Analysing the top side of the board it can be seen that largest visible IC has "ES32G031" written on top of it, this is the main microcontroller on board which does the calculating. Full name of this microcontroller is "STM32G031K8T6U" Whole purpose of this board is to let customers of ST Microelectronics test the microcontrollers to their own use cases.

Going bottom right from the main microcontroller we have an IC named "Y820 3905 33". This is an voltage regulator

Going top right from the main microcontroller we have an IC named "LDK120M33R", which is a low-dropout regulator.



Analysing the bottom side of the board it can be seen that the largest IC has markings that say “STM32 F103CBT6 GQ25Q 1293” this is the ST F103CBT6 microcontroller, used as the ST-LINK/V2-1. This microcontroller is used as a debugger/programmer.

On the very top side, there is a IC marked “2141” which is a component named STMPS2141STR, this component is a enhanced single channel power switch with active low enable pin

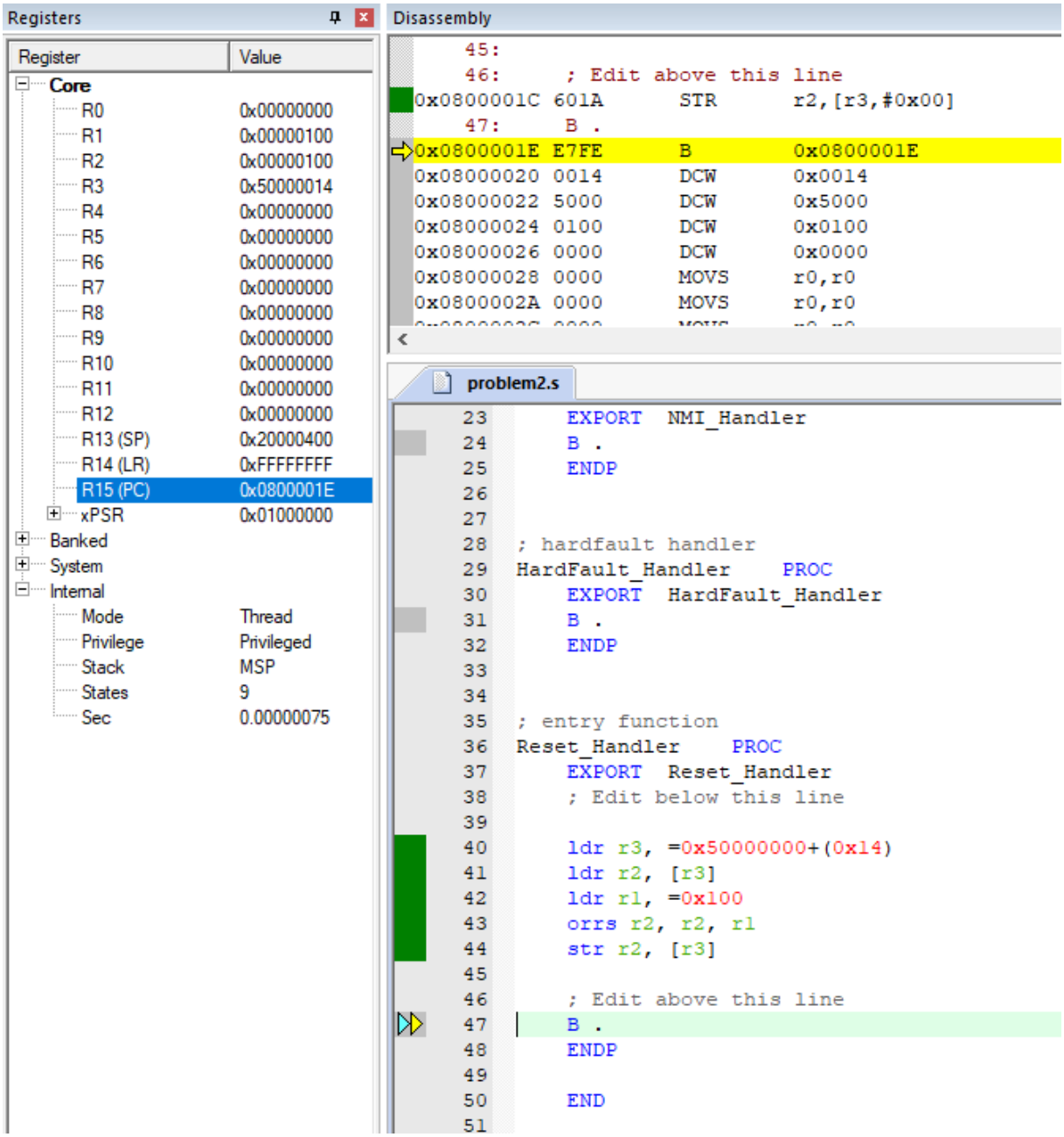
Under “2141”, there is a IC named “USBLC6-2P6”, which is an USB OTG FS Electro Static Decharge protection controller.

Going lower left from the nucleo branding, we have an IC numbered U8, named LD1117S59TR. This is a REG Low-dropout regulator, working on 5V, 0.8 amps.

3. Problem 2

It is requested that code will light up a LED connected to pin PA8. To achieve this GPIOA address is required. From the "STM32G0x1 Peripheral Register Boundary Addresses" in RM0444, it is learned that boundary address of GPIOA is "0x5000 0000 – 0x5000 03FF". Offset for the ODR register is also required, looking up to the "GPIO Register Map and Reset Values" table on Section 6 information is gathered that offset value is "0x14". To turn the led on the 8th pin, 9th bit on the register has to be 1. This equates to decimal 2^8 which is 0x100 hexadecimal. After gathering this information, registers are loaded with the addresses and values and whole code is a added to provided "basic assembly template for keil ARM simulator", only added code given left to prevent mess.

```
ldr r3, =0x50000000+(0x14)
ldr r2, [r3]
ldr  r1, =0x100
orrs r2, r2, r1
str r2, [r3]
```



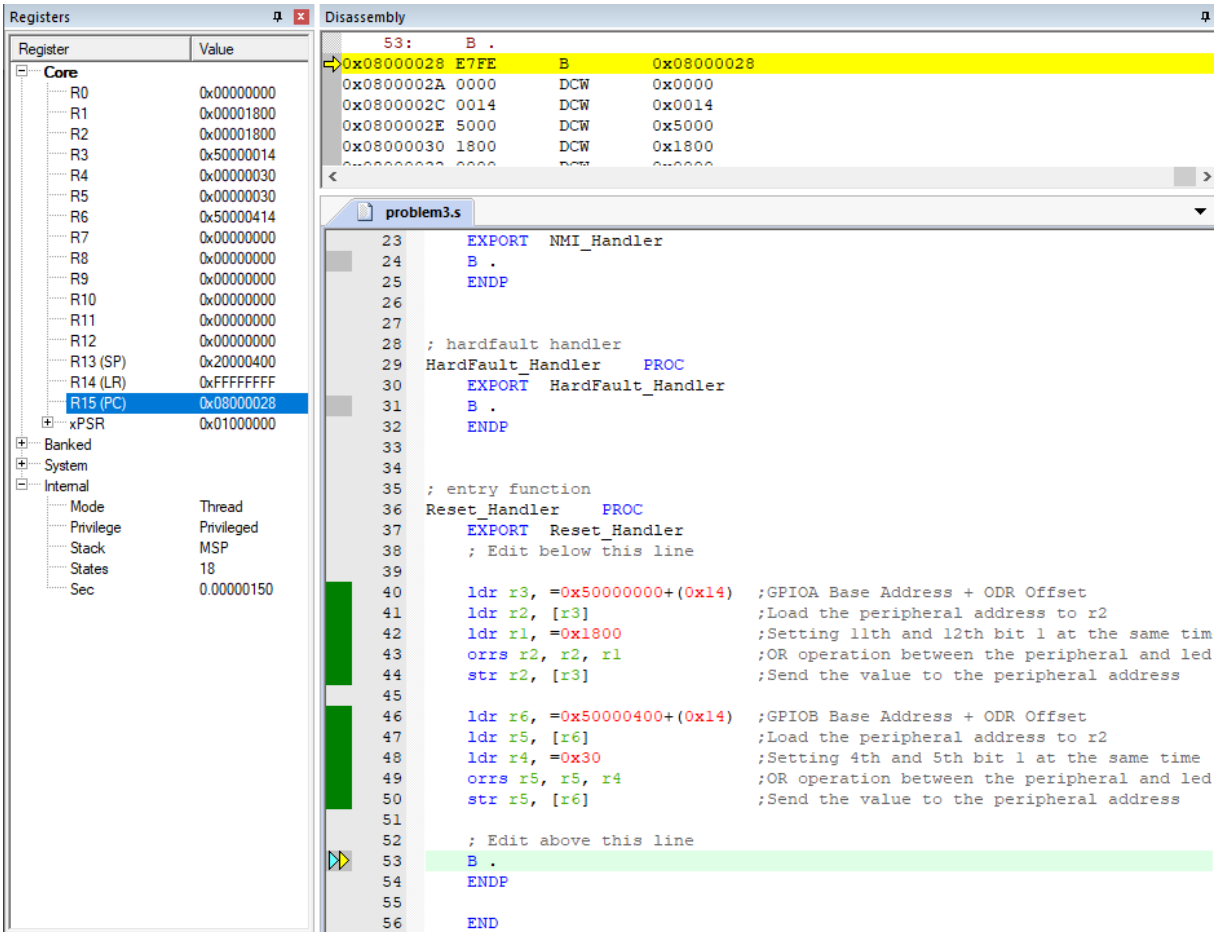
4. Problem 3

```
ldr r3, =0x50000000+(0x14)  ;GPIOA Base Address + ODR Offset
ldr r2, [r3]                ;Load the peripheral address to r2
ldr r1, =0x1800             ;Setting 11th and 12th pin 1 at the same time
orrs r2, r2, r1             ;OR operation between the peripheral and led encoding
str r2, [r3]                ;Send the value to the peripheral address

ldr r6, =0x50000400+(0x14)  ;GPIOB Base Address + ODR Offset
ldr r5, [r6]                ;Load the peripheral address to r4
ldr r4, =0x30               ;Setting 4th and 5th pin 1 at the same time
orrs r5, r5, r4             ;OR operation between the peripheral and led encoding
str r5, [r6]                ;Send the value to the peripheral address
```

It is requested that the new assembly code will light up LEDs connected to pin PA11, PA12, PB4 and PB5. Firstly starting from PA11 and PA12, GPIOA base address is required. As it is known from the previous problem, boundary address of GPIOA is “0x5000 0000 – 0x5000 03FF” and offset for the ODR register is “0x14”. To turn the led on the 11th pin, 12th bit on the register has to be 1, to turn on the 12th pin, 13th. This equates to binary “1100000000000”, hex 0x1800.

For pin PB4 and PB5, different base address, GPIOB boundary address is required, which is “0x5000 0400 - 0x5000 07FF”. Same exact procedure is followed. New value for the 4th and 5th pin, which is “110000” binary, 0x30 hex. After gathering this information, registers are loaded with the addresses and values and whole code is a added to provided “basic assembly template for keil ARM simulator”, only added code given above to prevent mess.

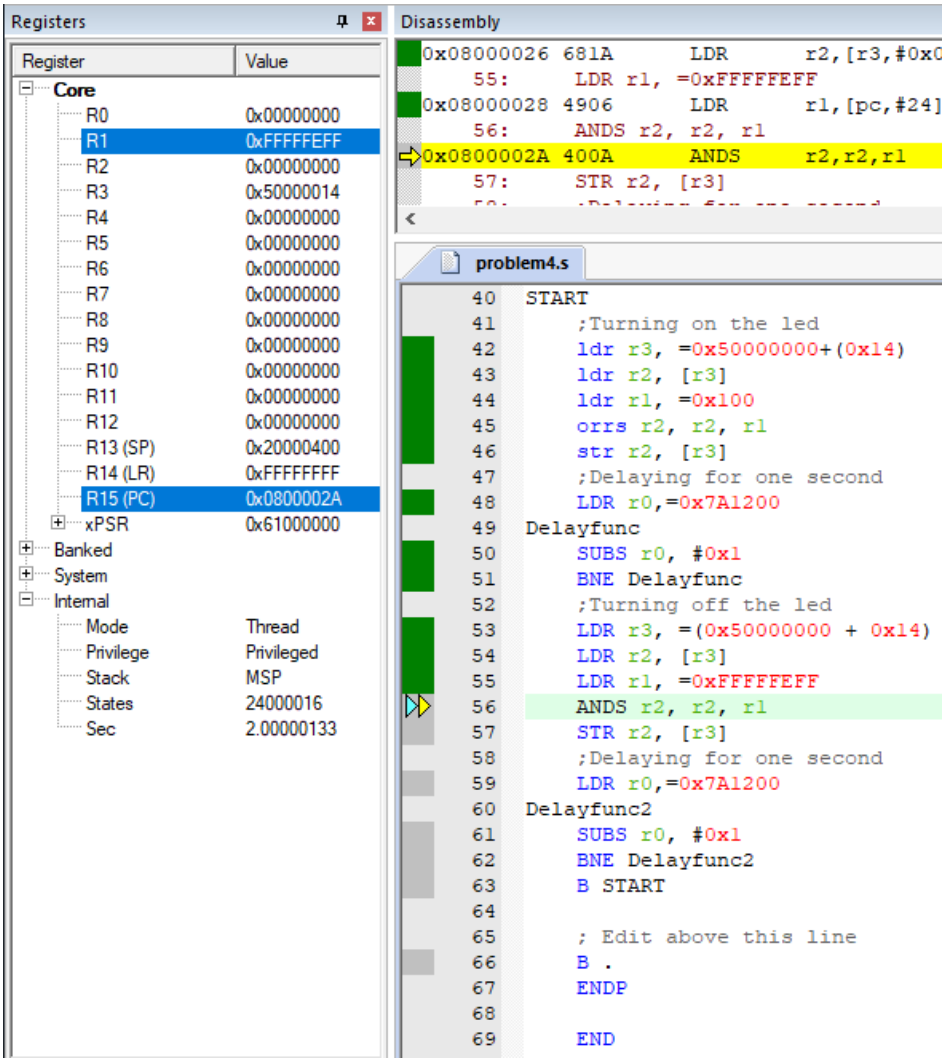


5. Problem 4

```
START
    ;Turning on the led
    ldr r3, =0x50000000+(0x14)
    ldr r2, [r3]
    ldr r1, =0x100
    orrs r2, r2, r1
    str r2, [r3]
    ;Delaying for one second
    LDR r0,=0x7A1200
Delayfunc
    SUBS r0, #0x1
    BNE Delayfunc
    ;Turning off the led
    LDR r3, =(0x50000000 + 0x14)
    LDR r2, [r3]
    LDR r1, =0xFFFFFEFF
    ANDS r2, r2, r1
    STR r2, [r3]
    ;Delaying for one second
    LDR r0,=0x7A1200
Delayfunc2
    SUBS r0, #0x1
    BNE Delayfunc2
    B START
```

I've combined my written codes from ELEC334 Homework 2 Problem 6 and ELEC 334 Lab 1 Problem 2 together to complete this task. I quote my HW2 report as I explain:

Chose an appropriate number to load to r0 for 16 MHz processor. Subtraction and branch not equal operations take 2 cycles. Divided 16 million by 2 for 2 cycles per whole operation, gathered 8 million per second. Set r0 to "0x7A1200" to get approximately 1 second delay. Then gathered the led lighting algorithm from Problem 2, set the appropriate GPIOA address, set the correct pin. Arranged the correct turning of pin. Then add the whole code to provided "basic assembly template for keil ARM simulator", only added code given left to prevent mess.



6. References

1. RM0444 Reference manual, ST Microelectronics, https://www.st.com/resource/en/reference_manual/dm00371828-stm32g0x1-advanced-armbased-32bit-mcus-stmicroelectronics.pdf
2. Online PCB Viewer, Altium, https://www.altium.com/viewer/
3. ELEC334 Homework 2 Report, Berat Kızılarmut