GEBZE TECHNICAL UNIVERSITY

ELECTRONICS ENGINEERING

ELM335

Microprocessors Laboratory

LAB 2 Experiment Report

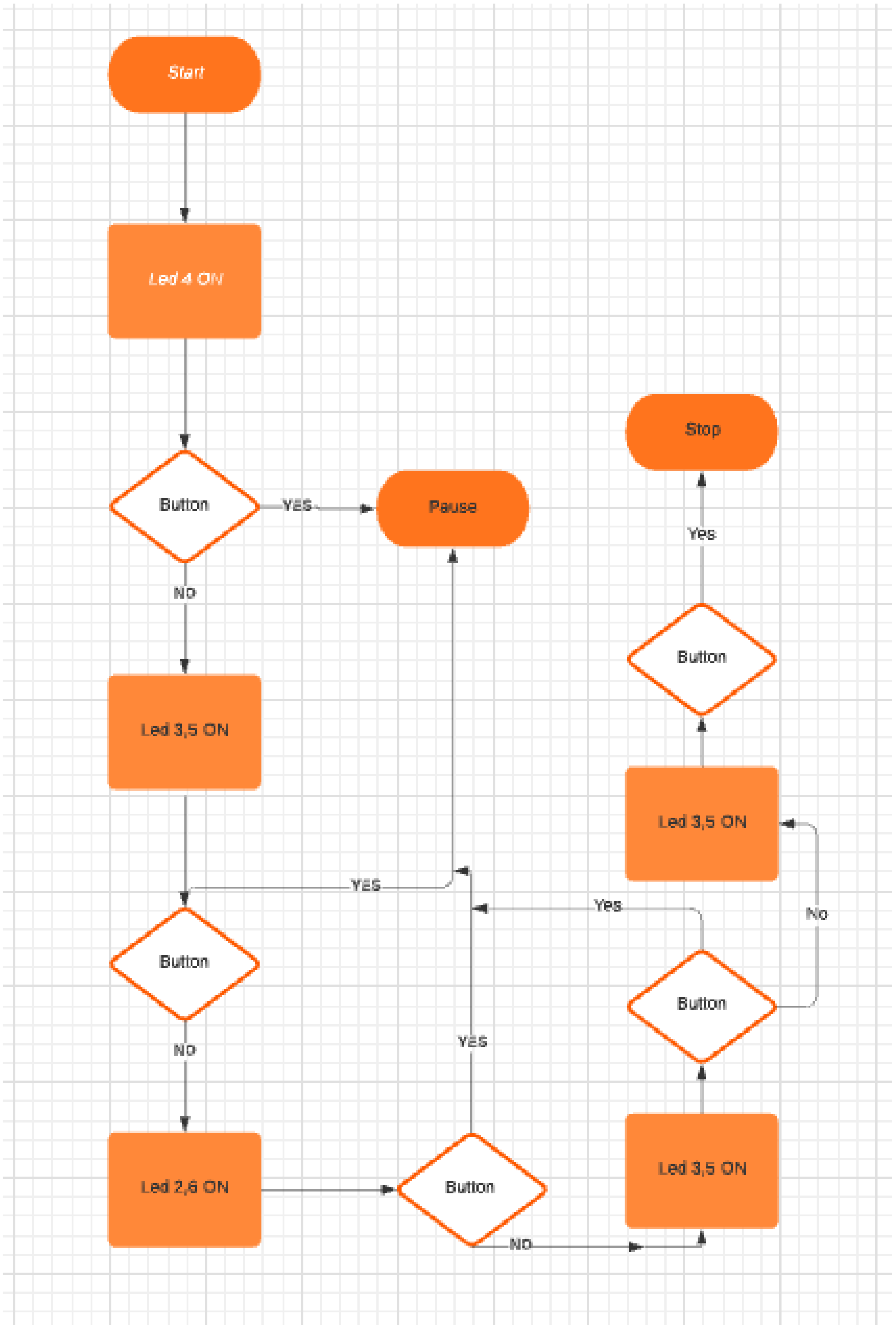
Prepared by
151024008 - Abdürrahim Deniz KUMBARACI
171024050 - Abdül Samet Karapınar
171024008 - Yasin Özbek

Introduction

It was aimed to recognize the microprocessor to be used in this experiment. And it was aimed to write led and seven segment display burning code for this microprocessor.

Problem 1

Flow Chart



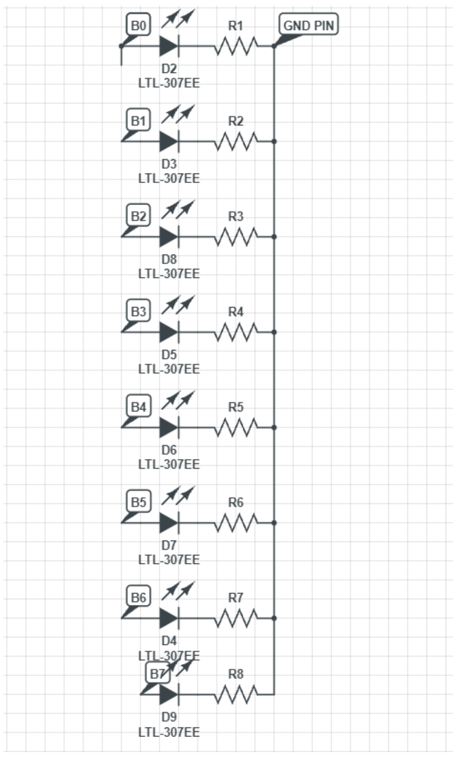
Code

```
.syntax unified
.cpu cortex-m0
.fpu softvfp
.thumb


/* make linker see this */
.global Reset_Handler

/* get these from linker script */
.word _sdata
.word _edata
.word _sbss
.word _ebss


/* define peripheral addresses from RM0444 page 57, Tables 3-4 */
.equ RCC_BASE,          (0x40021000)          // RCC base address
.equ RCC_IOPENR,        (RCC_BASE   + (0x34)) // RCC IOPENR register offset

.equ GPIOB_BASE,        (0x50000400)          // GPIOB base address
.equ GPIOB_MODER,       (GPIOB_BASE + (0x00)) // GPIOB MODER register offset
.equ GPIOB_ODR,         (GPIOB_BASE + (0x14)) // GPIOB ODR register offset

.equ GPIOA_BASE,        (0x50000000)          // GPIOA base address
.equ GPIOA_MODER,       (GPIOA_BASE + (0x00)) // GPIOA MODER register offset
.equ GPIOA_IDR,         (GPIOA_BASE + (0x10)) // GPIOA IDR register offset

/* vector table, +1 thumb mode */
.section .vectors
vector_table:
	.word _estack             /*     Stack pointer */
	.word Reset_Handler +1    /*     Reset handler */
	.word Default_Handler +1  /*       NMI handler */
	.word Default_Handler +1  /* HardFault handler */
	/* add rest of them here if needed */


/* reset handler */
.section .text
Reset_Handler:
	/* set stack pointer */
	ldr r0, =_estack
	mov sp, r0

	/* initialize data and bss
	 * not necessary for rom only code
	 * */
	bl init_data
	/* call main */
	bl main
	/* trap if returned */
	b .


/* initialize data and bss sections */
.section .text
```

```
init_data:

      /* copy rom to ram */
      ldr r0, =_sdata
      ldr r1, =_edata
      ldr r2, =_sidata
      movs r3, #0
      b LoopCopyDataInit

      CopyDataInit:
            ldr r4, [r2, r3]
            str r4, [r0, r3]
            adds r3, r3, #4

      LoopCopyDataInit:
            adds r4, r0, r3
            cmp r4, r1
            bcc CopyDataInit

      /* zero bss */
      ldr r2, =_sbss
      ldr r4, =_ebss
      movs r3, #0
      b LoopFillZerobss

      FillZerobss:
            str  r3, [r2]
            adds r2, r2, #4

      LoopFillZerobss:
            cmp r2, r4
            bcc FillZerobss

      bx lr


/* default handler */
.section .text
Default_Handler:
      b Default_Handler


/* main function */
.section .text
main:
      /* enable GPIOB clock, bits 0-7 on IOPENR */
      ldr r6, =RCC_IOPENR
      ldr r5, [r6]
      movs r4, 0x2
      orrs r5, r5, r4
      str r5, [r6]

      /* setup PB for 8 leds 01 for bits 15-0 in MODER */
      ldr r6, =GPIOB_MODER
      ldr r5, [r6]
      ldr r4, =0xFFFF  //1111_1111_1111_1111
      mvns r4, r4
      ands r5, r5, r4
      ldr r4, =0x5555 //0101_0101_0101_0101
```

```
    orrs r5, r5, r4
    str r5, [r6]

          /* setup PA0 in MODER */
    ldr r6, =GPIOA_MODER
    ldr r5, [r6]
    ldr r4, =0xFFFF     //1111_1111_1111_1111
    mvns r4, r4         //0000_0000_0000_0000
    ldr r4, =0x0000
    ands r5, r5, r4     //0000_0000_0000_0000
    str r5, [r6]

    /* FIRST turn on the first leds connected to B0 B1 B2 in ODR */
    AddFunction1:
    ldr r6, =GPIOB_ODR
    ldr r5, [r6]
    movs r4, 0x8 //0000_1000
    orrs r5, r5, r4     //xxxx 1xxx
    str r5, [r6] //0000 1000
    ldr R0,=#400000 //100ms bekleme (16mega/4cycle)=400K
    bl bekle
    bl ButtonControlFunction

    AddFunction2:
    ldr r6, =GPIOB_ODR
    ldr r5, [r6]
    movs r4, 0x1C //0001_1100
    orrs r5, r5, r4     //xxxx 1xxx
    str r5, [r6] //0001 1100
    ldr R0,=#400000 //100ms bekleme (16mega/4cycle)=400K
    bl bekle
    bl ButtonControlFunction

AddFunction3:
    ldr r6, =GPIOB_ODR
    ldr r5, [r6]
    movs r4, 0x3E //0011_1110
    orrs r5, r5, r4
    str r5, [r6] //0011 1110
    ldr R0,=#400000 //100ms bekleme (16mega/4cycle)=400K
    bl bekle
    bl ButtonControlFunction

  AddFunction4:
    ldr r6, =GPIOB_ODR
    ldr r5, [r6]
    movs r4, 0x7F //0111_1111
    orrs r5, r5, r4
    str r5, [r6] //0111 1111
    ldr R0,=#400000 //100ms bekleme (16mega/4cycle)=400K
    bl bekle
    bl ButtonControlFunction

    SubsFunction1:
    ldr r6, =GPIOB_ODR
    ldr r5, [r6]
    movs r4, 0x3E //0011_1110
    ldr r5, =0x0000
    orrs r5, r5, r4     //xx11 111x
```

```
        str r5, [r6] //0011 1110
        ldr R0,=#400000 //100ms bekleme (16mega/4cycle)=400K
        bl bekle
        bl ButtonControlFunction

        SubsFunction2:
        ldr r6, =GPIOB_ODR
        ldr r5, [r6]
        movs r4, 0x1C //0001_1100
        ldr r5, =0x0000
        orrs r5, r5, r4     //xx11 111x
        str r5, [r6] //0011 1110
        ldr R0,=#400000 //100ms bekleme (16mega/4cycle)=400K
        bl bekle
        bl ButtonControlFunction

SubsFunction3:
        ldr r6, =GPIOB_ODR
        ldr r5, [r6]
        movs r4, 0x8 //0000_1000
        ldr r5, =0x0000
        orrs r5, r5, r4     //xx11 111x
        str r5, [r6] //0011 1110
        ldr R0,=#400000 //100ms bekleme (16mega/4cycle)=400K
        bl bekle
        bl ButtonControlFunction

        SubsFunction4:
        ldr r6, =GPIOB_ODR
        ldr r5, [r6]
        movs r4, 0x0 //0000_0000
        ldr r5, =0x0000
        orrs r5, r5, r4     //xx11 111x
        str r5, [r6] //0011 1110
        ldr R0,=#400000 //100ms bekleme (16mega/4cycle)=400K
        bl bekle
        bl ButtonControlFunction

        LastControlFunction:
        bl ButtonControlFunction
        b AddFunction1

        ButtonControlFunction:
        ldr r6, =GPIOA_IDR
        ldr r5, [r6]
        ldr r7, =0x1 //0000 0001
        ands r5,r5,r7
        cmp r5,r7           //Eşit mi değil mi compare? Eşitse input vardır
        beq ButtonLoop
        bx lr

        ButtonLoop:
        ldr r6, =GPIOB_ODR
        ldr r5, [r6]
        /* turn on PB7 port */
        movs r4, 0x80
        orrs r5, r5, r4
        str r5, [r6]
        ldr R0,=#400000 //100ms bekleme (16mega/4cycle)=400K
        bl bekle
        b ButtonControlFunction


/*
        ButtonControlFunction:
        turn on led connected to A6 in ODR
        ldr r6, =GPIOA_IDR
        ldr r5, [r6]
        lsrs r5, r5, #7
        movs r4, 0x1 //0001
        ands r5, r5, r4

        cmp r5, #0x1 //0001
        bne bright;
        beq dark;
        bx lr
 */

        bekle:
        SUBS R0,R0,#1
        BNE bekle
        bx lr
        b .

        /* this should never get executed */
        nop
```



Conclusion

In conclusion in this lab we got ourselves familiarised with the assembly language in led example .This week we try to do problem 2 part but we couldnt complete it. For problem 1 we combined the leds like in the problem.