EE443 - Embedded Systems

Experiment **4** Laboratory Report
LCD Module and Time Markers

Name: M. Serdar Karaman

Number: 190206038

Date: 23 - November - 2016

Objective

Writing string to LCD module and measuring task time with markers. Optimization to avoid unnecessary updates.

Experimental Work

Experiment code

```
#include <avr/io.h>
#include <stdio.h>
#include <util/delay.h>
#include "LCDmodule.h"
#define _NOP() do { __asm__ __volatile__ ("nop"); } while (0)



int main(){
    DDRB  |= 0b11111111;// all output
    DDRD  |= 0b11111100;//0-1 input, others  output
    PORTD |= 0b00000011;//Pull-up resistor for inputs
    LCD_Init();
    LCD_Clear();

    unsigned char PDsave = 0x00;
    unsigned char PDsave_1 = 0x00;

    char LCDtext[16];
    unsigned char Atten;
    unsigned char Atten_save;

    Atten = 4;

    PrintByte(LCDtext, "Atten=");
    LCD_MoveCursor(1,1);
    LCD_WriteString(LCDtext);

    while(1){

        PDsave_1=PDsave;
        PDsave = PIND;
        Atten_save=Atten;

            if(((PIND & 0x01) == 0x00) && ((PDsave_1&0x01) == 0x01)){//SW0
pressed
                _NOP();
                _NOP();
                if(Atten > 1){
                    _NOP();
                    Atten--;
                }
                else{
                    ;
                }

            }
            else if(((PIND & 0x02) == 0x00) && ((PDsave_1&0x02) == 0x02)){//SW1
pressed
                _NOP();
                if(Atten < 11){
                    _NOP();
                    Atten++;
                }
                else{
```

```
                    ;
                }
            }
            else{//Both pressed or not pressed
                ;
            }

            if(Atten!=Atten_save){//When change occur
                _NOP();
                PORTD |= 0b00010000;//PD4 is 1
                PrintByte(LCDtext, "", Atten);
                PORTD &= 0b11101111;//PD4 is 0

                PORTD |= 0b00100000;//PD5 is 1
                LCD_MoveCursor(1,7);
                LCD_WriteString(LCDtext);
                PORTD &= 0b11011111;//PD5 is 0
            }
            else{
                ;
            }
            _delay_us(500);
    };

    return(0);
}
```

sprint function didn’t worked for me.

While loop time, no transition on inputs;
SW0 pressed, 1,67 us
SW1 pressed, 1,61 us
None pressed, 1,39 us

PrintByte function time, 5,52 us
LCD_MoveCursor and LCD_WriteString function time, 208 us

Code is fully optimized. We are iniating LCD screan only once and updating it only when there is chage.

By adding NOP function after if statement we try to equalize time spent on else and if part of code.

Due to the NOPs there is no execution time difference between counting up and counting down on Atten.

LSS file

```
000003f4 <main>:
 3f4:   ef 92           push    r14
 3f6:   ff 92   push    r15
 3f8:   0f 93           push    r16
 3fa:   1f 93           push    r17
 3fc:   df 93           push    r29
```

```
3fe:   cf 93          push   r28
400:   cd b7          in     r28, 0x3d      ; 61
402:   de b7          in     r29, 0x3e      ; 62
404:   60 97          sbiw   r28, 0x10      ; 16
406:   0f b6          in     r0, 0x3f  ; 63
408:   f8 94          cli
40a:   de bf          out    0x3e, r29      ; 62
40c:   0f be          out    0x3f, r0  ; 63
40e:   cd bf          out    0x3d, r28      ; 61
410:   84 b1          in     r24, 0x04      ; 4
412:   8f ef      ldi r24, 0xFF      ; 255
414:   84 b9          out    0x04, r24      ; 4
416:   8a b1          in     r24, 0x0a      ; 10
418:   8c 6f          ori    r24, 0xFC      ; 252
41a:   8a b9          out    0x0a, r24      ; 10
41c:   8b b1          in     r24, 0x0b      ; 11
41e:   83 60          ori    r24, 0x03      ; 3
420:   8b b9          out    0x0b, r24      ; 11
422:   0e 94 56 01    call   0x2ac    ; 0x2ac <LCD_Init>
426:   0e 94 ec 00    call   0x1d8    ; 0x1d8 <LCD_Clear>
42a:   7e 01          movw   r14, r28
42c:   08 94          sec
42e:   e1 1c          adc    r14, r1
430:   f1 1c          adc    r15, r1
432:   c7 01          movw   r24, r14
434:   60 e0          ldi    r22, 0x00      ; 0
436:   71 e0          ldi    r23, 0x01      ; 1
438:   0e 94 5b 02    call   0x4b6    ; 0x4b6 <PrintByte>
43c:   81 e0          ldi    r24, 0x01      ; 1
43e:   61 e0          ldi    r22, 0x01      ; 1
440:   0e 94 2d 01    call   0x25a    ; 0x25a <LCD_MoveCursor>
444:   c7 01          movw   r24, r14
446:   0e 94 a2 00    call   0x144    ; 0x144 <LCD_WriteString>
44a:   80 e0          ldi    r24, 0x00      ; 0
44c:   94 e0          ldi    r25, 0x04      ; 4
44e:   09 b1          in     r16, 0x09      ; 9
450:   48 99          sbic   0x09, 0  ; 9
452:   22 c0          rjmp   .+68     ; 0x498 <main+0xa4>
454:   80 ff      sbrs r24, 0
456:   20 c0          rjmp   .+64     ; 0x498 <main+0xa4>
458:   00 00          nop
45a:   00 00          nop
45c:   92 30          cpi    r25, 0x02      ; 2
45e:   40 f1          brcs   .+80     ; 0x4b0 <main+0xbc>
460:   00 00          nop
462:   19 2f          mov    r17, r25
464:   11 50          subi   r17, 0x01      ; 1
466:   91 17          cp     r25, r17
468:   19 f1          breq   .+70     ; 0x4b0 <main+0xbc>
46a:   00 00          nop
46c:   5c 9a          sbi    0x0b, 4  ; 11
46e:   c7 01          movw   r24, r14
470:   67 e0          ldi    r22, 0x07      ; 7
472:   71 e0          ldi    r23, 0x01      ; 1
474:   41 2f          mov    r20, r17
476:   50 e0          ldi    r21, 0x00      ; 0
478:   0e 94 5b 02    call   0x4b6    ; 0x4b6 <PrintByte>
47c:   5c 98          cbi    0x0b, 4  ; 11
47e:   5d 9a          sbi    0x0b, 5  ; 11
480:   81 e0          ldi    r24, 0x01      ; 1
482:   67 e0          ldi    r22, 0x07      ; 7
484:   0e 94 2d 01    call   0x25a    ; 0x25a <LCD_MoveCursor>
488:   c7 01          movw   r24, r14
48a:   0e 94 a2 00    call   0x144    ; 0x144 <LCD_WriteString>
48e:   5d 98          cbi    0x0b, 5  ; 11
490:   c8 01          movw   r24, r16
492:   09 b1          in     r16, 0x09      ; 9
494:   48 9b          sbis   0x09, 0  ; 9
496:   de cf          rjmp   .-68     ; 0x454 <main+0x60>
498:   49 99          sbic   0x09, 1  ; 9
49a:   0a c0          rjmp   .+20     ; 0x4b0 <main+0xbc>
49c:   81 ff      sbrs r24, 1
49e:   08 c0          rjmp   .+16     ; 0x4b0 <main+0xbc>
4a0:   00 00          nop
4a2:   9b 30          cpi    r25, 0x0B      ; 11
4a4:   28 f4          brcc   .+10     ; 0x4b0 <main+0xbc>
4a6:   00 00          nop
4a8:   19 2f          mov    r17, r25
4aa:   1f 5f      subi r17, 0xFF      ; 255
4ac:   91 17          cp     r25, r17
4ae:   e9 f6          brne   .-70     ; 0x46a <main+0x76>
4b0:   19 2f          mov    r17, r25
4b2:   c8 01          movw   r24, r16
4b4:   ee cf          rjmp   .-36     ; 0x492 <main+0x9e>
```