31.05.2021

16:00 – 20:00

Urszula Chmielewska

300167

EMISY

Lab 3

Introduction

The goal of this laboratory was to get familiarized with LCD display and LEDS with keyboard inputs using EDSIM simulator.

Hardware – general approach

The LCD display in the task 1 is connected using 8-bit mode. Pins P3.0 and P3.1 are assigned to RS and E pin respectively. Whole P1 is assigned for the data bus. Switch button, used in this task, is connected to P2.5 pin.

For the task 2, all keypad bits are connected to P0. Due to such connection, all row bits can be modified individually at once. An LED, used in this task, is connected to P1.0 pin. The system clock is set to 12MHz.

TASK 1

Code

```
LCD_RS EQU P3.0         ;RS pin connected to P3.0
LCD_E EQU P3.1          ;E pin connected to P3.1
LCD_BUS EQU P1          ;Data bus takes pins from P1.0 to P1.7
SWITCH EQU P2.5         ;switch five


;wait for more than 30ms
mov R2, #20
    delay_30:
        lcall   ms_delay
        djnz    R2, delay_30

clr LCD_RS

mov LCD_BUS, #00111000B             ;function set
lcall   write_command
lcall   us_delay

mov LCD_BUS, #00001110B             ;display ON/OFF control
lcall   write_command
lcall   us_delay

mov LCD_BUS, #00000110B             ;entry mode set
lcall   write_command
lcall   us_delay
```

```
switch_control:
    mov LCD_BUS, #00000001B         ;display clear
    lcall   write_command
    lcall   ms_delay
    jb  SWITCH, $                   ;waits for the switch to be pressed (if switch is 1-
not pressed, routine jumps to itself)

    ;send name
    mov LCD_BUS, #'U'
    lcall   write_data
    lcall   us_delay

    mov LCD_BUS, #'R'
    lcall   write_data
    lcall   us_delay

    mov LCD_BUS, #'S'
    lcall   write_data
    lcall   us_delay

    mov LCD_BUS, #'Z'
    lcall   write_data
    lcall   us_delay

    mov LCD_BUS, #'U'
    lcall   write_data
    lcall   us_delay

    mov LCD_BUS, #'L'
    lcall   write_data
    lcall   us_delay

    mov LCD_BUS, #'A'
    lcall   write_data
    lcall   us_delay

    jnb SWITCH, $                   ;waits for the switch to be turned off
    lcall   switch_control          ;call function which waits for pressed button


write_command:
    clr LCD_RS
    setb    LCD_E
    clr LCD_E
    ret

write_data:
    setb    LCD_RS
    setb    LCD_E
    clr LCD_E
    ret
```

```
us_delay:
    mov R0, #20                         ;delay is more than 39us so 20*2 (djnz takes 2us)
    djnz    R0, $                       ;decrement and jump if R0 reaches 0
    ret                                 ;return from the subroutine

ms_delay:                               ;3*256*2 is more than 1,53ms as expected
    mov     R1, #3                      ;load 3 to R1 to make 3 loops
    delay_jump:
        mov     R0, #255                ;256 - value to decrement
        djnz    R0, $                   ;djnz takes 2 cycles - 2us
        djnz    R1, delay_jump          ;decrement R1 and loop if not yet 0
        ret
```

Code description

The goal of this task was to connect LCD display in a way, that it displays a word when the proper switch is pressed. The screen was supposed to be cleared when the switch was disabled.

This solution is based on the 8-bit program prepared by me for lab 1. The difference is that this time, I have designed *"switch_control"* subroutine which is waiting for the for switch to be pressed. If it happens, letters are loaded to LCD_BUS and become visible on the display. After that, program waits in an infinite loop, for switch to be disabled. If it happens, program goes back to the beginning of the subroutine. Clears the register and waits for pressed switch again. In my solution, pressing switch "5" executes showing word on the display.

Description about short and long delays taken from the lab 1 report:

Short - microsecond delay

The aim of this subroutine was to obtain a short break, which is required between commands with short execution time. The routine is defined by loading into a register a value and then decrementing it by using 'djnz' instruction. This particular instruction takes 2 cycles to execute itself, so to obtain a proper delay, the value which we load to the register should be two times smaller than the expected amount of microsecond delay which we want to obtain. In this task it was more than 39us delay.

Long - milliseconds delay

Some commands require higher amount of time to execute. The procedure of this delay is similar to the one described above. In this case it was more than 1,53ms delay. I separated this value into 3*255*2us. This time I needed two loops. I load 3 to the first register in the first loop and 255 to the second register in the second loop. Basically this subroutines works like two 'for' loops. The outer loop decrements from 3 to 1 and inner loop decrements from 255 to 0.

TASK 2

Code

```
LED EQU P1.0            ;first LED from the right will be turned on/off
KEYPAD EQU P0           ;keyboard row pins take from P0.0 uo to P0.3

jmp begin               ;skip interrupt handler to the beginning routine

org 0013h               ;write to specified place in the memory
cpl LED                 ;logically complements a given bit
reti

begin:
    setb    EA                  ;enable global interrupt
    setb    EX1                 ;enable INT1 interrupt
    setb    IT1                 ;set INT1 to work with falling edge

    mov KEYPAD, #01110000B      ;clear all keyboard row bits
    jmp $                       ;waiting for the interrupt
```

Code description

The goal of this task was to write a program which turns on LED every time when key on the keypad is pressed.

It the beginning, the program jumps to the subroutine which initializes all necessary setup. Then the global interrupt and INT1 external interrupt are enabled. After that INT1 is set to work with the falling edge. Then I clear all keyboard row bits by loading 0 to the proper pins. Finally, program waits for the interrupt to happen. Then I logically complement chosen LED, which causes turning on or turning off of the LED.

Declaration

I declare that this piece of work which is the basis for recognition of achieving learning outcomes in the EMISY course was completed on my own.

Urszula Chmielewska

300167

31.05.2021