LABORATORY WORK REPORT NO. 5

KEYPAD AND LCD

Subject: Microprocessors and their Programming
Student: Halil Ibrahim Bekli
Target Device: PIC16F1518
Variant: 1

1. Goal and Task

GOAL: To learn how to utilize a button matrix (keypad) and output information to an LCD screen.

TASK:
Program the microcontroller according to the assigned variant (Variant 1) so that:
1. The first line of the LCD displays 'Laukiu kodo' (Waiting for code).
2. The code is entered using buttons 0 to 9 and is displayed as asterisks (*) on the second line.
3. Upon entering the correct security code and confirming it with the assigned button (ok1):
   - Activate the latch (output signal) for a specific time duration (t1).
   - Display 'Atrakinta' (Unlocked) on the LCD for 2 seconds.

2. Task Variant Data

Selected Variant: 1
- Confirmation Key (ok1): #
- Latch Duration (t1): 300 ms
- Code Length: 3 digits
- Secret Code: 123
- Latch Output Connection: PORTA.0

3. Circuit Diagram Description

Based on the laboratory schematic and Variant 1 configuration:

1. MCU: PIC16F1518 (Internal Osc 16MHz).
2. LCD Display (16x2):
   - Control Lines: RS -> RB6, E -> RB7.
   - Data Lines (4-bit mode): D4..D7 -> RB2..RB5.
3. Keypad (4x4 Matrix):
   - Rows (Outputs): RC0, RC1, RC2, RC3.
   - Columns (Inputs): RC4, RC5, RC6, RC7.
4. Output Latch (Sklastis):
   - Active High Output connected to PORTA Pin 0 (RA0).

4. Code with Comments

```
/*
 * File:   LabWork5.c
 * Author: Halil Ibrahim Bekli
 *
 * Target: PIC16F1518
 * Description:
 * Variant 1 keypad-based security system using a 4x4 matrix keypad
 * and a 16x2 LCD display. No external libraries are used.
 *
 * Variant 1 Parameters:
 * - Confirmation key (ok1): '#'
 * - Secret code: "123"
 * - Code length: 3 digits
 * - Latch activation time (t1): 300 ms
 * - Latch output pin: PORTA.0
 */

#include <xc.h>
#include <string.h>

/* ================= CONFIGURATION BITS ================= */

#pragma config FOSC = INTOSC, WDTE = OFF, PWRTE = ON, MCLRE = ON
#pragma config CP = OFF, BOREN = OFF, CLKOUTEN = OFF
#pragma config IESO = OFF, FCMEN = OFF, WRT = OFF
#pragma config STVREN = ON, LVP = OFF

#define _XTAL_FREQ 16000000

/* ================= VARIANT 1 DEFINITIONS ================= */

#define SECRET_CODE "123"
#define CODE_LEN    3
#define CONFIRM_KEY '#'
#define LATCH_TIME  300                 // milliseconds
#define LATCH_PIN   LATAbits.LATA0

/* ================= LCD PIN DEFINITIONS ================= */
/* 4-bit mode, PORTB */

#define LCD_RS LATBbits.LATB6
#define LCD_EN LATBbits.LATB7
#define LCD_D4 LATBbits.LATB2
#define LCD_D5 LATBbits.LATB3
#define LCD_D6 LATBbits.LATB4
#define LCD_D7 LATBbits.LATB5

/* ================= GLOBAL VARIABLES ================= */

char entered_code[4];                   // stores entered digits
unsigned char index = 0;

/* ================= LCD FUNCTIONS ================= */

/* Generates enable pulse for LCD */
void lcd_pulse(void) {
    LCD_EN = 1;
    __delay_us(1);
    LCD_EN = 0;
}

/* Sends 4 bits of data to LCD */
void lcd_send4(unsigned char data) {
    LCD_D4 = data & 1;
    LCD_D5 = (data >> 1) & 1;
    LCD_D6 = (data >> 2) & 1;
    LCD_D7 = (data >> 3) & 1;
    lcd_pulse();
}
```

```
/* Sends command to LCD */
void lcd_cmd(unsigned char cmd) {
    LCD_RS = 0;
    lcd_send4(cmd >> 4);
    lcd_send4(cmd);
    __delay_ms(2);
}

/* Sends single character to LCD */
void lcd_putch(char c) {
    LCD_RS = 1;
    lcd_send4(c >> 4);
    lcd_send4(c);
}

/* Sends string to LCD */
void lcd_puts(const char *s) {
    while (*s) {
        lcd_putch(*s++);
    }
}

/* Sets LCD cursor position */
void lcd_goto(unsigned char pos) {
    lcd_cmd(0x80 | pos);
}

/* Initializes LCD in 4-bit mode */
void lcd_init(void) {
    LCD_RS = 0;
    LCD_EN = 0;
    __delay_ms(20);

    lcd_send4(0x03);
    __delay_ms(5);
    lcd_send4(0x02);        // 4-bit mode

    lcd_cmd(0x28);          // 2 lines, 5x8 font
    lcd_cmd(0x0C);          // display ON
    lcd_cmd(0x01);          // clear display
}

/* ================ KEYPAD FUNCTION ================ */
/*
 * Reads a key from a 4x4 matrix keypad.
 * Rows: RC0-RC3 (outputs)
 * Columns: RC4-RC7 (inputs)
 */

char read_kpad(void) {
    const char keymap[4][4] = {
        {'1','2','3','A'},
        {'4','5','6','B'},
        {'7','8','9','C'},
        {'*','0','#','D'}
    };

    for (unsigned char row = 0; row < 4; row++) {
        LATC = 0x0F;                // set all rows high
        LATC &= ~(1 << row);        // activate one row
        __delay_us(50);

        for (unsigned char col = 0; col < 4; col++) {
            if (!(PORTC & (1 << (col + 4)))) {
                while (!(PORTC & (1 << (col + 4)))); // wait for release
                return keymap[row][col];
            }
        }
    }
    return 0;                       // no key pressed
}

/* ================= MAIN PROGRAM ================= */

void main(void) {

    /* ---- System Initialization ---- */

    OSCCON = 0b01111000;             // 16 MHz internal oscillator

    ANSELA = 0;
    ANSELB = 0;
    ANSELC = 0;

    TRISA = 0x00;                    // latch output
    TRISB = 0x00;                    // LCD output
    TRISC = 0xF0;                    // keypad columns input

    LATCH_PIN = 0;                   // latch initially OFF

    lcd_init();

    /* ---- Main Loop ---- */

    while (1) {

        index = 0;
        memset(entered_code, 0, sizeof(entered_code));

        lcd_cmd(0x01);
        lcd_goto(0x00);
        lcd_puts("Waiting for code");     // Waiting for code
        lcd_goto(0x40);

        /* ---- Code Entry Loop ---- */
        while (1) {
            char key = read_kpad();
            if (key) {
                __delay_ms(150);     // debounce

                if (key == CONFIRM_KEY)
                    break;

                if (key >= '0' && key <= '9' && index < CODE_LEN) {
                    entered_code[index++] = key;
                    lcd_putch('*');  // mask input
                }
            }
        }

        /* ---- Code Verification ---- */
        if (strcmp(entered_code, SECRET_CODE) == 0) {

            lcd_cmd(0x01);
            lcd_puts("Unlocked");   // Unlocked

            LATCH_PIN = 1;
            __delay_ms(LATCH_TIME);
            LATCH_PIN = 0;

            __delay_ms(2000);

        } else {

            lcd_cmd(0x01);
            lcd_puts("Error");      // Error
            __delay_ms(1000);
        }
    }
}
```

5. Program Algorithm

```
1. Initialize System: Set Oscillator to 16MHz, configure Ports (A, B, C) as digital.
2. Initialize Peripherals: Initialize LCD and clear screen. Set Latch pin low.
3. Main Loop:
  a. Reset input buffer and index.
  b. Display 'Waiting for code' on LCD Line 1.
  c. Input Loop: Scan keypad using library.
    i. If Number (0-9): Store in buffer, print '*' on LCD.
    ii. If Confirm Key (#): Break input loop.
  d. Verification:
    i. Compare buffer with '123'.
    ii. If Match: Print 'Unlocked', Pulse Latch for 300ms, Wait 2s.
    iii. If Mismatch: Print 'Error', Wait 1s.
  e. Repeat Loop.
```

6. Conclusions

In this laboratory work, the interfacing of a 4x4 keypad and a 16x2 LCD with the PIC16F1518 was successfully implemented.

The LCD control and keypad scanning were implemented manually without using any external libraries.

The logic for Variant 1 (code '123', latch time 300 ms) was verified.

The system correctly activates the output latch only when the correct security code is entered.