Marie Christelle Medino
Robin Morales
Dan Gutmann
CPE329

# Assignment 4

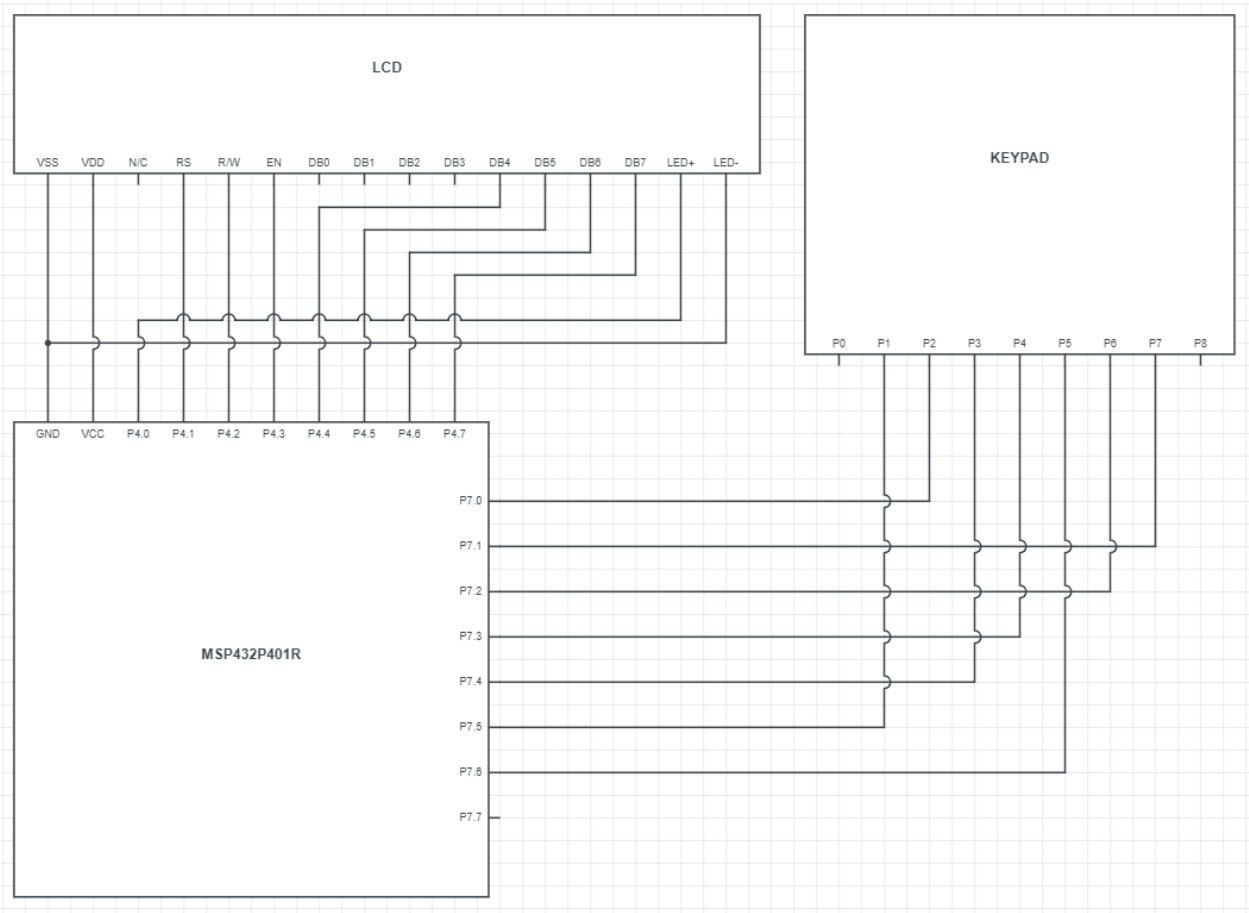


*Figure 1: LCD and Keypad Wiring Diagram*

Video Demonstration:
https://drive.google.com/open?id=1ObqiScB6N5JNWj5PyoMRPHQDprUD-Fv1

Marie Christelle Medino
Robin Morales
CPE329

```c
#include "utils.h"
#include "msp.h"
#include <stdarg.h>
#include "lcd.h"
#include "keypad.h"

#define LED_PORT P2

void main(void){
    WDT_A->CTL = WDT_A_CTL_PW | WDT_A_CTL_HOLD;     // stop watchdog timer

    setDCO(freq3Mhz);
    setGPIO_even(LED_PORT, BIT2, OUTPUT);

    char keypadChar;

    lcd.init();
    lcd.backlight(HIGH);
    lcd.clear();
    lcd.putstr("You pressed the");
    lcd.line_2();

    keypad.init();

    while(1){
        keypadChar = keypad.poll();
        if(keypadChar){
            lcd.line_2();
            lcd.putchar(keypadChar);
            lcd.putstr(" key.");
        }
        LED_PORT->OUT ^= BIT2;                      //Blink LED when finished.
        delay_us(100000);
    }
}
```

Marie Christelle Medino
Robin Morales
CPE329

```c
//KEYPAD.C
#include "keypad.h"
#include "msp.h"

#define KPD_PORT P7
#define KPD_ROW1 BIT0  //PIN2
#define KPD_ROW2 BIT1  //PIN7
#define KPD_ROW3 BIT2  //PIN6
#define KPD_ROW4 BIT3  //PIN4

#define KPD_COL1 BIT4  //PIN3
#define KPD_COL2 BIT5  //PIN1
#define KPD_COL3 BIT6  //PIN5

void KPD_init(void);
char poll(void);

uint8_t ROW = 0;
uint8_t COL = 0;

void KPD_init(void){
    setGPIO_odd(KPD_PORT, KPD_ROW1, OUTPUT);
    setGPIO_odd(KPD_PORT, KPD_ROW2, OUTPUT);
    setGPIO_odd(KPD_PORT, KPD_ROW3, OUTPUT);
    setGPIO_odd(KPD_PORT, KPD_ROW4, OUTPUT);

    setGPIO_odd(KPD_PORT, KPD_COL1, INPUT_PD);
    setGPIO_odd(KPD_PORT, KPD_COL2, INPUT_PD);
    setGPIO_odd(KPD_PORT, KPD_COL3, INPUT_PD);
}

uint8_t keyPress(void){ //Quickly check if a key is pressed.
    KPD_PORT->OUT |= (KPD_ROW4 | KPD_ROW3 | KPD_ROW2 | KPD_ROW1);
    return KPD_PORT->IN & (KPD_COL3 | KPD_COL2 | KPD_COL1);
}

char poll(void){    //Determine which key is pressed.
    uint8_t CURRENT_ROW = (uint8_t)KPD_ROW1;
    uint8_t ACTIVE_COL = 0x00;
    ROW = 0;
    COL = 0;
    char KPD_LOOKUP[4][3] = {{'1','2','3'},{'4','5','6'},{'7','8','9'},{'*','0','#'}};
    for(ROW = 0; ROW < 4; ROW++){
        KPD_PORT->OUT &= ~(KPD_ROW4 | KPD_ROW3 | KPD_ROW2 | KPD_ROW1);
        KPD_PORT->OUT = CURRENT_ROW;
        delay_us(100);
        ACTIVE_COL = (KPD_PORT->IN & (KPD_COL3 | KPD_COL2 | KPD_COL1)) >> 4;
        for(COL = 0; COL < 3; COL++){
            if(ACTIVE_COL & BIT0)
                return KPD_LOOKUP[ROW][COL];
            else
                ACTIVE_COL = ACTIVE_COL >> 1;
        }
        CURRENT_ROW = CURRENT_ROW << 1;
    }
    return '\0';
}

const struct keypadInterface keypad = {
     .init = KPD_init,
     .poll = poll
};
```

Marie Christelle Medino
Robin Morales
CPE329

```c
//KEYPAD.h
#ifndef KEYPAD_H_
#define KEYPAD_H_

#include "utils.h"

struct keypadInterface {
    void (*init)(void);
    char (*poll)(void);
};

extern const struct keypadInterface keypad;
#endif /* KEYPAD_H_ */
```