최범수 11일차 과제

1. HW_001

ADC란 아날로그 신호를 디지털 신호로 바꾸어주는 역할을 한다. ADC가 8개가 존재한다. 1024로 입력값이 나누어진다. 홀드 회로가 있어서 전압이 고정되고 8개의 단극성 입력이 가능하다. 아날로그 입력선은 짧게 하고 잡음의 영향이 없도록 회로를 구성한다. LC 필터를 통해 안정화 시키고 잡음이 심해 변동이 크면 평균치를 이용한다. 비트 7, 6 ADC는 기준전압을 설정하고 비트 5는 변환 결과를 저장하는 형식 비트 4에서 0까지는 입력채널과 입력 채널에 대한 이득을 선택한다.

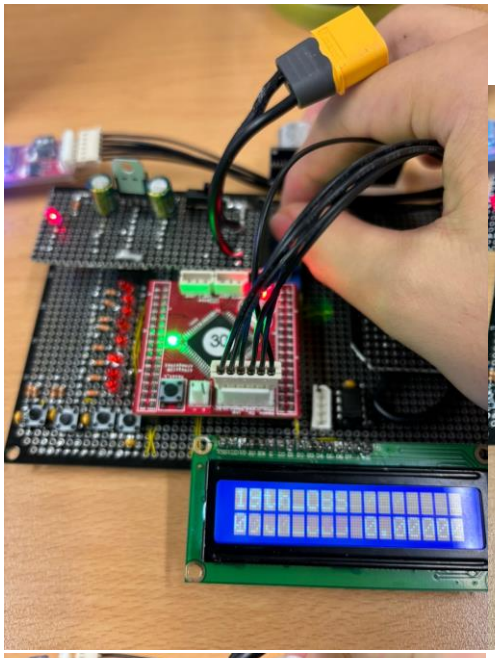
2. HW_002

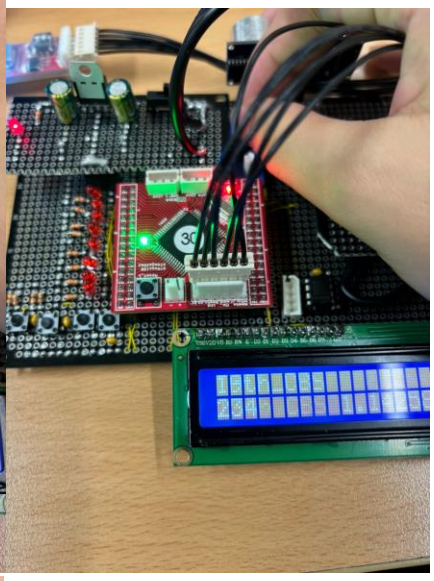
```
/*
 * test3.c
 *
 * Created: 2024-07-29 오전 2:46:38
 * Author : chlqj
 */

#include <avr/io.h>
#include "LCD_Text.h"
#include <util/delay.h>
#include <avr/interrupt.h>
#include <stdio.h>

int main(void)
{
    DDRF = 0x00;
    ADMUX = 0x40;
    ADCSRA = 0x87;
    SREG = 0x80;

    lcdInit();
```

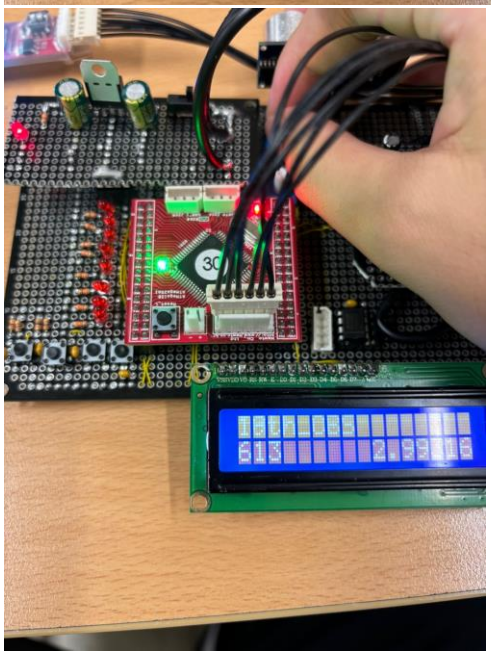
```
lcdClear();
 /* Replace with your application code */
 while (1)
 {
    unsigned int adcValue = 0;
    unsigned char channel = 0x00;
    double Voltage;

    DDRA = 0xFF;
    DDRD = 0x00;

     PORTA = 0xFF;//(LED 꺼짐)
    PORTD = 0xFF;
    char volt[10];
    ADMUX = 0x40 | channel;
    ADCSRA |= 0x40;
    while((ADCSRA&0x10)==0);
    adcValue = ADC;
    Voltage = adcValue * 5.0 / 1024.0;
    sprintf(volt, "%.5lf",Voltage);
    _delay_ms(100);
```

```
    lcdString(0, 0, "19th_CBS");
    lcdNumber(1, 0, adcValue);
    lcdString(1, 9, volt);

    // adc 값에 따라 LED 이동
    if (adcValue <= 128) PORTA = 0b11111110;
    else if (128 < adcValue && adcValue <= 256) PORTA = 0b11111101;
    else if (256 < adcValue && adcValue <= 384) PORTA = 0b11111011;
    else if (384 < adcValue && adcValue <= 512) PORTA = 0b11110111;
    else if (512 < adcValue && adcValue <= 640) PORTA = 0b11101111;
    else if (640 < adcValue && adcValue <= 768) PORTA = 0b11011111;
    else if (768 < adcValue && adcValue <= 896) PORTA = 0b10111111;
    else if (896 < adcValue) PORTA = 0b01111111;

    _delay_ms(100);

    lcdInit();
    lcdClear();

}
```

}







3. HW_003

```
#define F_CPU 16000000

#include <avr/io.h>
#include <util/delay.h>
#include <avr/interrupt.h>
#include <stdio.h>
#include "LCD_Text.h"

volatile unsigned int A = 1;
volatile unsigned int B = 1;
volatile char operator = '+';
volatile unsigned char now_operator = 0;

void printLCD() {
    lcdClear();
    char display[16];
    sprintf(display, "A: %d B: %d", A, B);
    lcdString(0, 0, display);
}

void Result()
 {
    char display[16];
    int result = 0;
```

```
    switch (operator) {
        case '+': result = A + B; break;
        case '-': result = A - B; break;
        case '*': result = A * B; break;
        case '/': result = A / B; break;
    }
    sprintf(display, "%d %c %d = %d", A, operator, B, result);
    lcdClear();
    lcdString(0, 0, display);
}

int main(void) {
    DDRA = 0xFF;
    DDRD = 0x00;

    PORTA = 0xFF;
    PORTD = 0xFF;

    lcdInit();
    lcdClear();

    while (1) {
        _delay_ms(50);

        if ((PIND & 0x01) == 0)
        {
```

```
    _delay_ms(50);
    if ((PIND & 0x01) == 0)
     {
        A++;
        printLCD();
    }
}

if ((PIND & 0x02) == 0)
{
    _delay_ms(50);
    if ((PIND & 0x02) == 0)
    {
        now_operator = (now_operator + 1) % 4;
        switch (now_operator) {
            case 0: operator = '+'; break;
            case 1: operator = '-'; break;
            case 2: operator = '*'; break;
            case 3: operator = '/'; break;
        }
        printLCD();

    }
}

if ((PIND & 0x04) == 0) {
```

```
            _delay_ms(50);
            if ((PIND & 0x04) == 0)
             {
                B++;
                printLCD();

            }
        }

        if ((PIND & 0x08) == 0)
        {
            _delay_ms(50);
            if ((PIND & 0x08) == 0)
            {
                Result();

            }
        }
    }
}
```