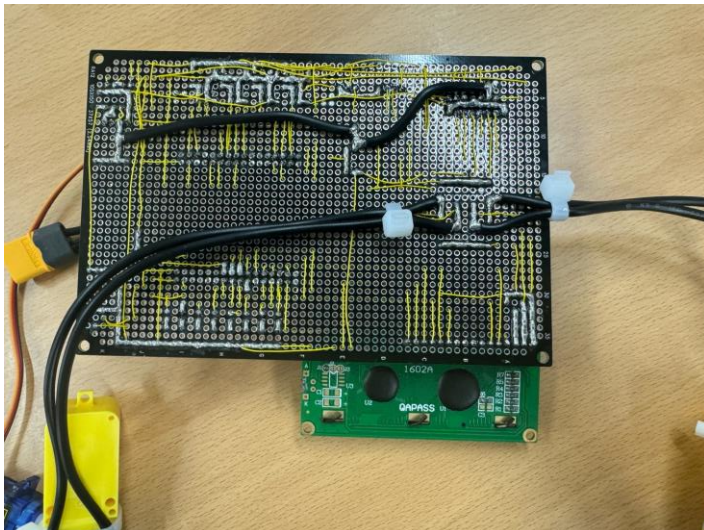
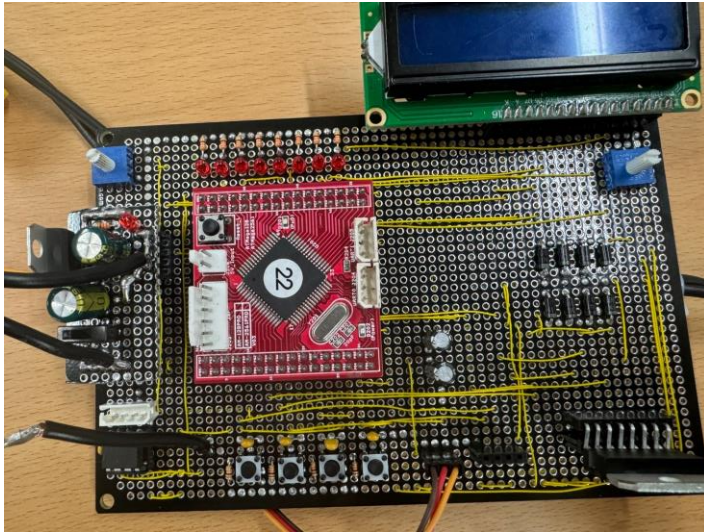


납땜 상황



과제 1

타이머는 일정한 시간 간격으로 0이나 1의 값을 주는 클럭과 클럭을 세는 카운터로 구성된다. 과제에서 사용한 overflow 타이머는 분주비마다 증가하는 tcnt0값이 지정한 값보다 커지는 오버플로우가 발생하면 인터럽트가 발생하는 구조이다. 사용법으로는 TIMSK로 인터럽트를 활성화 시키고 TCNT와 TCCR값을 정해준다 TCCR은 분주비로 시스템 클럭이 16000000 일때 TCCR로 나눈 주기마다 TCNT가 증가한다. TCNT 레지스터로는 값이 얼마나 커져야 오버플로우가 발생할지 지정한다. ISR(TIMER0_OVF_vect)로 오버플로우가 발생했을 때 무엇을 할지 정할수 있다.

과제 2 소스코드

```
#define F_CPU 16000000
```

```
#include <avr/io.h>

#include <avr/interrupt.h>

#include <util/delay.h>

#include "LCD_Text.h"
```

```
unsigned int count50us;
```

```
int timer;
```

```
char led = 0x00;
```

```
int main()
```

```
{
```

```
    DDRA = 0xFF;
```

```
    DDRF = 0x00;
```

```
    TCNT0 = 156;
```

```
    TCCR0 = 0x02;
```

```
    TIMSK = 0x01;
```

```
    ADMUX = 0x40;
```

```
    ADCSRA = 0x87;
```

```
    lcdInit();
```

```
    lcdClear();
```

```
    PORTA = led;
```

```
    sei();
```

```

while(1)
{
    ADCSRA |= (1 << ADSC);
    while (ADCSRA & (1 << ADSC));

    unsigned int adcValue = ADC;
    timer = ((adcValue + 1) / 128) * 200 + 700;

    lcdNumber(0, 0, timer);
    lcdString(0, 6, "us");
    _delay_ms(100);
    lcdClear();
}
}

```

```

ISR(TIMERO_OVF_vect)
{
    count50us++;
    if (count50us * 100 >= timer - 200)
    {
        if (led == 0x00)
        {
            led = 0xFF;
        }
    }
}

```

```

        else
        {
            led = 0x00;
        }

        PORTA = led;

        count50us = 0;
    }
}

```

과제 3 소스코드

```

#define F_CPU 16000000

#include <avr/io.h>

#include <avr/interrupt.h>

#include <util/delay.h>

#include "LCD_Text.h"

unsigned int count1ms = 0;

int flow = 0;

int year = 0;

int month = 1, day = 1, hour = 0, min = 0, sec = 0;

int mode = 0;

void count(int *month, int *day, int *hour, int *min, int *sec);

int main()
{

```

```
DDRD = 0x00;
```

```
lcdInit();
```

```
lcdClear();
```

```
TCCR0 = 0x04;
```

```
TCNT0 = 0x06;
```

```
TIMSK = 1;
```

```
ADMUX = 0x40;
```

```
ADCSRA = 0x87;
```

```
sei();
```

```
while(1)
```

```
{
```

```
    ADCSRA |= (1 << ADSC);
```

```
    while (ADCSRA & (1 << ADSC));
```

```
    unsigned int adcValue = ADC;
```

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

```
    {
```

```
        mode++;
```

```
        mode = mode % 6;
```

```
        _delay_ms(200);
```

```

}

else if((PIND & 0x02) == 0)
{
    _delay_ms(200);

    if(flow == 0)

        flow = 1;

    else if(flow == 1)

        flow = 0;

}

if (flow == 0)

    {

        switch (mode)

        {

            case 0:

                year = adcValue+1000;

                break;

            case 1:

                month = adcValue * 12 / 1023;

                break;

            case 2:

                day = adcValue * 31 / 1023;

                break;

            case 3:

                hour = adcValue * 23 / 1023;

                break;

```

case 4:

min = adcValue * 59 / 1023;

break;

case 5:

sec = adcValue * 59 / 1023;

break;

}

}

count(&month, &day, &hour, &min, &sec);

lcdNumber(0, 0, year);

lcdNumber(0, 4, month);

lcdNumber(0, 6, day);

lcdNumber(1, 0, hour);

lcdString(1, 2, ":");

lcdNumber(1, 3, min);

lcdString(1, 5, ":");

lcdNumber(1, 6, sec);

lcdString(1, 8, ".");

lcdNumber(1, 9, count1ms);

lcdNumber(0,10,mode);

_delay_ms(50);

lcdClear();

}

}

```
ISR(TIMER0_OVF_vect)
```

```
{
```

```
    if (flow == 1)
```

```
    {
```

```
        count1ms++;
```

```
        if (count1ms >= 1000)
```

```
        {
```

```
            count1ms = 0;
```

```
            sec++;
```

```
        }
```

```
    }
```

```
    TCNT0 = 6;
```

```
}
```

```
void count(int *month, int *day, int *hour, int *min, int *sec)
```

```
{
```

```
    if (*sec >= 60)
```

```
    {
```

```
        (*min)++;
```

```
        *sec = 0;
```

```
    }
```

```
    if (*min >= 60)
```

```
    {
```

```
        (*hour)++;
```



```

        *min = 0;
    }
    if (*hour >= 24)
    {
        (*day)++;
        *hour = 0;
    }
    if (*day > 31)
    {
        (*month)++;
        *day = 1;
    }
}

```

과제 4 소스코드

```

#define F_CPU 16000000UL

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

unsigned int count1ms;
unsigned int time = 0;
char led = 0x02;
unsigned int ms;
unsigned int sec;

```

```
unsigned int min;
```

```
void UART0_init();
```

```
void UART0_transmit(unsigned int data);
```

```
unsigned char UART0_receive();
```

```
void ledOn(char led);
```

```
void updateLCD(unsigned int time);
```

```
void ledOn(char led)
```

```
{
```

```
    char _led = 0xff;
```

```
    _led = _led - led;
```

```
    PORTA = _led;
```

```
}
```

```
void UART0_init()
```

```
{
```

```
    UBRR0H = 0x00;
```

```
    UBRR0L = 103;
```

```
    UCSR0A |= 0x20;
```

```
    UCSR0B |= 0x18;
```

```
    UCSR0C |= 0x06;
```

```
    DDRE = 0x02;
```

```
    UCSR0B |= (1 << RXCIE0);
```

```
}
```

```
unsigned char UART0_receive()
{
    while (!(UCSR0A & (1 << RXC0)));
    return UDR0;
}
```

```
void UART0_transmit(unsigned int data)
{
    while (!(UCSR0A & (1 << UDRE0)));
    UDR0 = data;
}
```

```
int main()
{
    DDRA = 0xff;
    TCNT0 = 0x04;
    TCCR0 = 0x06;
    TIMSK = 0x01;
    lcdInit();
    lcdClear();

    UART0_init();
    sei();
}
```

```

while(1)
{
    for (int i = 0; i < 5; i++)
    {
        led <<= 1;

        _delay_ms(1000);

        ledOn(led);

        led <<= 1;

        _delay_ms(1000);

        ledOn(led);

        led >>= 1;

        _delay_ms(1000);

        ledOn(led);

        if (led == 0x00)
        {
            led = 0b000000001;

            break;
        }
    }
}

for (int i = 0; i < 5; i++)
{
    led >>= 1;

    _delay_ms(1000);

    ledOn(led);

    led >>= 1;
}

```

```

        _delay_ms(1000);

        ledOn(led);

        led <= 1;

        _delay_ms(1000);

        ledOn(led);

        if (led == 0x00)

        {

                led = 0b10000000;

                break;

        }

    }

}

```

```

ISR(USART0_RX_vect)

{

    if (USART0_receive() == 48)

    {

        led = 0x01;

    }

    else if (USART0_receive() == 49)

    {

        led = 0x02;

    }

    else if (USART0_receive() == 50)

```

```
{  
  
    led = 0x04;  
  
}  
  
else if (UART0_receive() == 51)  
  
{  
  
    led = 0x08;  
  
}  
  
else if (UART0_receive() == 52)  
  
{  
  
    led = 0x10;  
  
}  
  
else if (UART0_receive() == 53)  
  
{  
  
    led = 0x20;  
  
}  
  
else if (UART0_receive() == 54)  
  
{  
  
    led = 0x40;  
  
}  
  
else if (UART0_receive() == 55)  
  
{  
  
    led = 0x80;  
  
}  
  
}
```

```

ISR(TIMERO0_OVF_vect)
{
    count1ms++;
    if (count1ms >= 2)
    {
        count1ms = 0;
        time++;
        if (time % 50 == 0)
        {
            UART0_transmit(led);
        }
        ms = time % 100;
        sec = (time / 100) % 60;
        min = (time / 6000) % 60;
        lcdNumber(0, 0, min);
        lcdString(0, 2, ":");
        lcdNumber(0, 3, sec);
        lcdString(0, 5, ":");
        lcdNumber(0, 6, ms);
    }
}

```

uart통신이 망가져서 아쉽게도 코드4 실행결과는 확인하지 못했습니다