

## Rs-232와 RS-485방식의 차이와 장단점

232방식은 단방향을 기본으로 한다. 데이터를 정확하게 보내기 위해 높은 전압인 -15v~15v를 사용하며 노이즈에 강하다.

485방식은 반이중 방식을 사용해 양방향 통신을 지원한다. 같은 데이터를 여러 수신기에 동시에 전송할수 있고 이를 위해 낮은 전압인 -7v~15v를 사용한다. 높은 전송 속도와 장거리 통신이 가능하다

## 232방식의 장단점

장점: 낮은 가격, 구현하기 쉬워 일반적인 pc와 주변기기와 통신에 적합함

단점: 노이즈에 취약하고 전송속도가 느리며 대역폭이 작아 대용량 데이터를 전송하지 못한다. 전송거리가 짧으며 단방향 통신이기 때문에 양방향 통신을 위해서는 두개의 전선이 필요하다

## 485방식의 장단점

장점: 전송 거리가 길다, 전송속도가 빠르고 대용량 데이터를 전송할수 있다. 여러 장치와 통신할 수 있다.

단점: 구현하기 어렵고 가격이 높다

## 과제 2 소스코드

```
#define F_CPU 16000000
```

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

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

```
#include <util/delay.h>
```

```
#include "LCD_Text.h"
```

```
volatile unsigned char RxBuffer;
```

```
void UART0_init();
```

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

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

```
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;
```

```
}
```

```
ISR(USART0_RX_vect)
```

```
{
```

```
    RxBuffer = UART0_receive();
```

```
    switch (RxBuffer)
```

```
    {
```

```
        case '0':
```

```
            PORTA = 0b01111111;
```

```
            break;
```

```
        case '1':
```

```
            PORTA = 0b10111111;
```

```
            break;
```

```
        case '2':
```

```
            PORTA = 0b11011111;
```

```
            break;
```

```
        case '3':
```

```
            PORTA = 0b11101111;
```

```
            break;
```

```
        case '4':
```

```
            PORTA = 0b11110111;
```

```
            break;
```

```
        case '5':
```

```
            PORTA = 0b11111011;
```

```
            break;
```

```
        case '6':  
            PORTA = 0b11111101;  
            break;  
        case '7':  
            PORTA = 0b11111110;  
            break;  
        default:  
            break;  
    }  
}
```

```
int main()  
{  
    sei();  
    PORTA = 0xff;  
    UART0_init();  
    DDRF = 0x00;  
    ADMUX = 0x40;  
    ADCSRA = 0x87;  
    DDRA = 0xFF;  
    lcdInit();  
    lcdClear();  
  
    while (1)  
    {
```

```

    unsigned int adcValue = 0;

    unsigned char channel = 0x00;

    ADMUX = 0x40 | channel;

    ADCSRA |= 0x40;

    while ((ADCSRA & 0x10) == 0);

    adcValue = ADC;

    _delay_ms(100);

    UART0_transmit(adcValue & 0xFF);

    lcdString(0, 0, "ADC Value :");

    lcdNumber(0, 12, adcValue);

    lcdString(1, 0, "LED Value :");

    lcdNumber(1, 12, RxBuffer-48);

    _delay_ms(100);

    lcdClear();

}

return 0;

}

```

### 과제 3 소스코드

```
#define F_CPU 16000000UL
```

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

```
#include <util/delay.h>
```

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

```
char led = 0xff;
```

```
void UART0_init();
```

```
void UART0_transmit(char data);
```

```
char UART0_receive();
```

```
void UART0_init()
```

```
{
```

```
    UBRR0H = 0x00;
```

```
    UBRR0L = 103;
```

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

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

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

```
    DDRE |= 0x02;
```

```
}
```

```
char UART0_receive()
```

```
{
```

```
    while (!(UCSR0A & (1 << RXC0)));
```

```
    return UDR0;
```

```
}
```

```
void UART0_transmit(char data)
```

```
{
```

```

        while (!(UCSR0A & (1 << UDRE0)));

        UDR0 = data;
    }

int main()
{
    UART0_init();

    DDRA = 0xff;

    DDRD = 0x00;

    EIMSK=(1<<INT0);

    EICRA=(ISC01>>1)|(ISC00>>1);

    sei();

    while (1)
    {
        switch (UART0_receive())
        {
            case '0':

                led -= 0x01;

                UART0_transmit('0');

                UART0_transmit(' ');

                UART0_transmit('l');

                UART0_transmit('e');

                UART0_transmit('d');

```

```
UART0_transmit(' ');

UART0_transmit('o');

UART0_transmit('n');

UART0_transmit(' ');

break;

case '1':

    led -= 0x02;

    UART0_transmit('1');

    UART0_transmit(' ');

    UART0_transmit('l');

    UART0_transmit('e');

    UART0_transmit('d');

    UART0_transmit(' ');

    UART0_transmit('o');

    UART0_transmit('n');

    UART0_transmit(' ');

    break;

case '2':

    led -= 0x04;

    UART0_transmit('2');

    UART0_transmit(' ');

    UART0_transmit('l');

    UART0_transmit('e');

    UART0_transmit('d');

    UART0_transmit(' ');
```



```
UART0_transmit('o');  
  
UART0_transmit('n');  
  
UART0_transmit(' ');  
  
break;  
  
case '3':  
  
led -= 0x08;  
  
UART0_transmit('3');  
  
UART0_transmit(' ');  
  
UART0_transmit('l');  
  
UART0_transmit('e');  
  
UART0_transmit('d');  
  
UART0_transmit(' ');  
  
UART0_transmit('o');  
  
UART0_transmit('n');  
  
UART0_transmit(' ');  
  
break;  
  
case '4':  
  
led -= 0x10;  
  
UART0_transmit('4');  
  
UART0_transmit(' ');  
  
UART0_transmit('l');  
  
UART0_transmit('e');  
  
UART0_transmit('d');  
  
UART0_transmit(' ');  
  
UART0_transmit('o');
```

```
UART0_transmit('n');

UART0_transmit(' ');

break;

case '5':

led -= 0x20;

UART0_transmit('5');

UART0_transmit(' ');

UART0_transmit('l');

UART0_transmit('e');

UART0_transmit('d');

UART0_transmit(' ');

UART0_transmit('o');

UART0_transmit('n');

UART0_transmit(' ');

break;

case '6':

led -= 0x40;

UART0_transmit('6');

UART0_transmit(' ');

UART0_transmit('l');

UART0_transmit('e');

UART0_transmit('d');

UART0_transmit(' ');

UART0_transmit('o');

UART0_transmit('n');
```

```
UART0_transmit(' ');

break;

case '7':

led -= 0x80;

UART0_transmit('7');

UART0_transmit(' ');

UART0_transmit('l');

UART0_transmit('e');

UART0_transmit('d');

UART0_transmit(' ');

UART0_transmit('o');

UART0_transmit('n');

UART0_transmit(' ');

break;

case '8':

UART0_transmit('L');

UART0_transmit('E');

UART0_transmit('F');

UART0_transmit('T');

UART0_transmit(' ');

led <= 1;

break;

case '9':

UART0_transmit('R');

UART0_transmit('l');
```

```

        UART0_transmit('G');

        UART0_transmit('H');

        UART0_transmit('T');

        UART0_transmit(' ');

        led >>= 1;

        break;

    }

    PORTA = led;

}

return 0;

}

```

```

ISR(INT0_vect)

{

    _delay_ms(200);

    led = 0xff;

    UART0_transmit('R');

    UART0_transmit('E');

    UART0_transmit('S');

    UART0_transmit('E');

    UART0_transmit('T');

    UART0_transmit(' ');

}

```