

차 례

1. 리모콘 이론
2. 리모콘 수신부 회로도
3. 프로그램
4. 데이터 시트

1. 리모콘 이론

저는 Datasheet가 있는 삼성 리모콘 (BN59-00968A)를 사용했습니다.

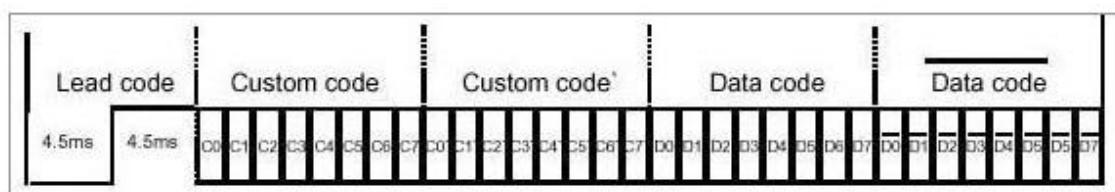


리모콘의 데이터 수신 포맷은 다음 과 같이 3개의 부분으로 나누어져 있습니다.

1. Lead Code : 리모콘의 시작을 알려줌.
2. Custom Code : 2바이트의 데이터 시트로 구성됨.
3. Data Code : 2바이트의 데이터로 구성됨.

Data Code는 두 개로 나누어 있음. **첫 번째 바이트**에 **리모콘 버튼**에 대한 **정보**를 가지고 있고, **두 번째 바이트**는 **첫 번째 바이트**에 대한 **비트 반전**을 시켜준다.

삼성리모콘(BN59-00968A)에 대한 데이터 포맷



Custom Code와 Data Code는 각각 2개의 바이트로 구성되어 있고 총 4바이트로 구성되어 있다. 각 바이트에서 처음 수신되는 비트는 최하위비트(LSB)를 나타낸다. 각 비트는 0과 1로 구성되어 있는데 비트가 0인 경우와 1인 경우의 데이터 길이가 달라진다.



비트가 0인 경우 그림의 왼쪽과 같이 Low 부분이 560usec이고 High부분이 560usec으로 전체 데이터 길이는 1.125msec이다.

비트가 1인 경우 그림의 오른쪽과 같이 Low 부분이 560usec이고 High부분이 1680usec으로 전체 데이터 길이는 2.25msec이다.

이에 수신된 데이터 비트가 0과 1을 판별하기 위해서는 데이터의 길이를 구하면 알 수 있다. 그래서 각 신호의 하강에지부분에서 다음 하강에지까지 시간을 측정하여 1.125msec이면 수신된 비트가 0이고 2.25msec이면 수신된 비트가 1인 것을 알 수 있다.

이런 방법으로 32비트(4바이트)를 모두 수신하면 된다.

다음은 삼성리모콘(BN59-00968A)에 대한 데이터 포맷이다.

리모콘 버튼	custom code0	custom code1	data0	data1
전원	0x07	0x07	0x02	0xFD
TV/외부입력	0x07	0x07	0x01	0xFE
0	0x07	0x07	0x11	0xEE
1	0x07	0x07	0x04	0xFB
2	0x07	0x07	0x05	0xFA
3	0x07	0x07	0x06	0xF9
4	0x07	0x07	0x08	0xF7
5	0x07	0x07	0x09	0xF6
6	0x07	0x07	0x0A	0xF5
7	0x07	0x07	0x0C	0xF3
8	0x07	0x07	0x0D	0xF2
9	0x07	0x07	0x0E	0xF1
회전프시	0x07	0x07	0x1F	0xE0
TV/유선	0x07	0x07	0x27	0xD8
음성다중	0x07	0x07	0x00	0xFF
회전크기	0x07	0x07	0x3E	0xC1
+100	0x07	0x07	0x23	0xDC
앞채널	0x07	0x07	0x13	0xEC
채널↑	0x07	0x07	0x12	0xED
채널↓	0x07	0x07	0x10	0xEF
음량+	0x07	0x07	0x07	0xF8
음량-	0x07	0x07	0x0B	0xF4
메뉴	0x07	0x07	0x1A	0xE5
조용히	0x07	0x07	0x0F	0xF0
비디오전원	0x05	0x05	0x02	0xFD
되감기	0x05	0x05	0x18	0xE7
재생	0x05	0x05	0x19	0xE6
정지	0x05	0x05	0x15	0xEA
빨리감기	0x05	0x05	0x1A	0xE5

리모콘 수신 방법

먼저 적외선 신호의 Lead Code를 읽어야하기 때문에 다음과 같이 프로그램을 만든다.

Lead Code의 데이터 길이가 13.5ms이기 때문에

if(edge > 87 && edge < 93) 로 프로그램을 만든다.

Lead Code 만족시 다음 Data Code를 받는다.

비트가 0인 경우 그림의 왼쪽과 같이 Low 부분이 560usec이고 High부분이 560usec으로 전체 데이터 길이는 1.125msec이고 비트가 1인 경우 그림의 오른쪽과 같이 Low 부분이 560usec이고 High부분이 1680usec으로 전체 데이터 길이는 2.25msec이기 때문에 다음과 같이 프로그램을 만든다.

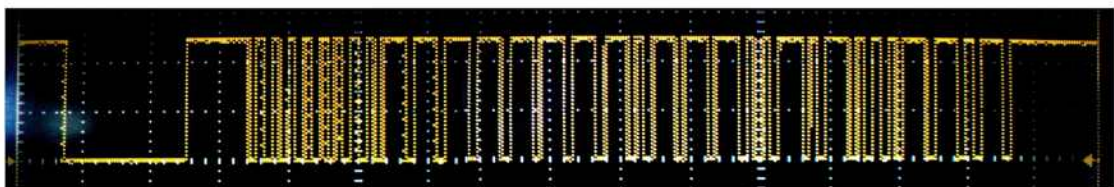
```
if(edge > 10 && edge < 13) // 0Bit인 경우 1.125ms
{
    IR_rx_data[flag_count] &= ~mask;
    mask=mask<<1;
    flag++;
}
else if(edge > 21 && edge < 24) // 10Bit인 경우 2.25ms
{
    IR_rx_data[flag_count] |= mask;
    mask=mask<<1;
    flag++;
}
```

edge는 time_count로 변수를 지정한 것으로

time_count는 time_count++로 후가산을 주어서 100us 인터럽트를 만들어줌.

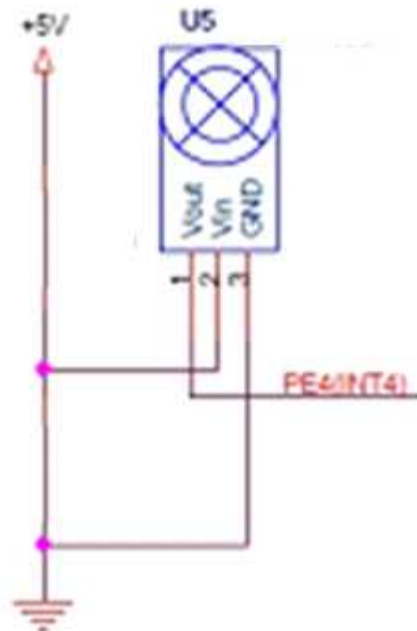
```
interrupt[TIM3_COMPA] void compa(void) // 100us 인터럽트
{
    time_count++;
}
```

※ 오실로스코프로 뽑아본 리모콘 데이터 파형



2. 리모콘 수신부 회로도

양승현 교수님께서 만드신 ATmega128 KIT의 **PORTE.4번**에 Interrupt를 하기위해서 연결 하였습니다.



3. 프로그램

```
// 20071739 김정환  
// 리모콘 수신 소스
```

```
#include <mega128.h>  
#include <delay.h>
```

```
void system_init();
```

```
unsigned int time_count = 0;  
unsigned int edge=0;  
unsigned char flag = 0;  
unsigned char flag_count = 0;  
unsigned char IR_start=0;  
unsigned char IR_rx_data[4] = { 0x00, 0x00, 0x00, 0x00 };  
unsigned char IR_data;
```

```

unsigned long mask=0x01;
unsigned char S_flag = 0, old_flag = 0;

interrupt[EXT_INT4] void input(void) // 입력 캡처 인터럽트
{
    edge = time_count;
    time_count=0;
    if(edge > 87 && edge < 93) // lead code를 읽기 위함. 13.5ms
        IR_start=1;

    else if(IR_start==1)
    {
        if(edge > 10 && edge < 13)// 0      1.125ms
        {
            IR_rx_data[flag_count] &= ~mask;
            mask=mask<<1;
            flag++;
        }
        else if(edge > 21 && edge < 24)// 1      2.25ms
        {
            IR_rx_data[flag_count] |= mask;
            mask=mask<<1;
            flag++;
        }
        if(flag == 8)
        {
            flag = 0;
            mask = 0x01;
            flag_count++;
        }
        if(flag_count == 4)// data를 모두 받으면..
        {
            if((IR_rx_data[2] ^ IR_rx_data[3]) == 0xFF)
            //data가 정확히 들어왔는가?
            {
                if(S_flag == 0)
                {
                    old_flag = S_flag;
                    S_flag = 1;
                }
                else
            }
        }
    }
}

```

```

        {
            old_flag = S_flag;
            S_flag = 0;
        }
        IR_data = IR_rx_data[2];
        //리모컨 수신 데이터 저장
        IR_start=0;
        mask=0x01;
        flag_count = 0;
        IR_rx_data[0] = 0x00;
        IR_rx_data[1] = 0x00;
        IR_rx_data[2] = 0x00;
        IR_rx_data[3] = 0x00;
        flag = 0;
    }
    else {
        IR_start=0;
        mask=0x01;
        flag_count = 0;
        IR_rx_data[0] = 0x00;
        IR_rx_data[1] = 0x00;
        IR_rx_data[2] = 0x00;
        IR_rx_data[3] = 0x00;
        flag = 0;
    }
}

}

interrupt[TIM3_COMPA] void compa(void)    // 100us 인터럽트
{
    time_count++;    // IR timer
}

void main()
{
    DDRE = 0x00;    //IR Rx Pin ( ICP3 )
    DDRA = 0xff;
    system_init();
    while (1)
    {
        switch(IR_data)

```

```

    {
        case 0x11 :    //Number 0
            PORTA=0x00;
            break;
        case 0x04 :    //Number 1
            PORTA=0x01;
            break;
        case 0x05 :    //Number 2
            PORTA=0x03;
            break;
        case 0x06 :    //Number 3
            PORTA=0x07;
            break;
        case 0x08 :    //Number 4
            PORTA=0x0f;
            break;
        case 0x09 :    //Number 5
            PORTA=0x1f;
            break;
        case 0x0A :    //Number 6
            PORTA=0x3f;
            break;
        case 0x0C :    //Number 7
            PORTA=0x7f;
            break;
        default :      //그 외 키 입력 시
            PORTA=0x00;
            delay_ms(500);
            PORTA=0xff;
            delay_ms(500);
            break;
    }
}

void system_init()
{
    TCCR3A=0x00; //CTC mode() , 프리스케일 8
    TCCR3B=0x8A; //0xCA rising edge 0x8A falling edge

    OCR3AH=0x00;
    OCR3AL=199; //16Mhz : 199 = 100u,

```



```

// f(PWM) = 8Mhz / 2 / 64 / 255 = 250Hz
TCCR1A = 0xA1; //phase corect PWM(8bit) - TOP : 0x00FF
TCCR1B = 0x03; //프리스케일 : 64
TCCR1C = 0X00;
// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x00;
ETIMSK=0x10;
EICRB = 0x02; //INT4 falling edge interrupt
EIMSK = 0x10; //INT4 interrupt enable
SREG |= 0x80;
}

```

4. 데이터 시트

DIMENSIONS

(Unit : mm)

