Software Guide for NS-RX231

<INFRARED RECEIVE>

Contents

1 Load project

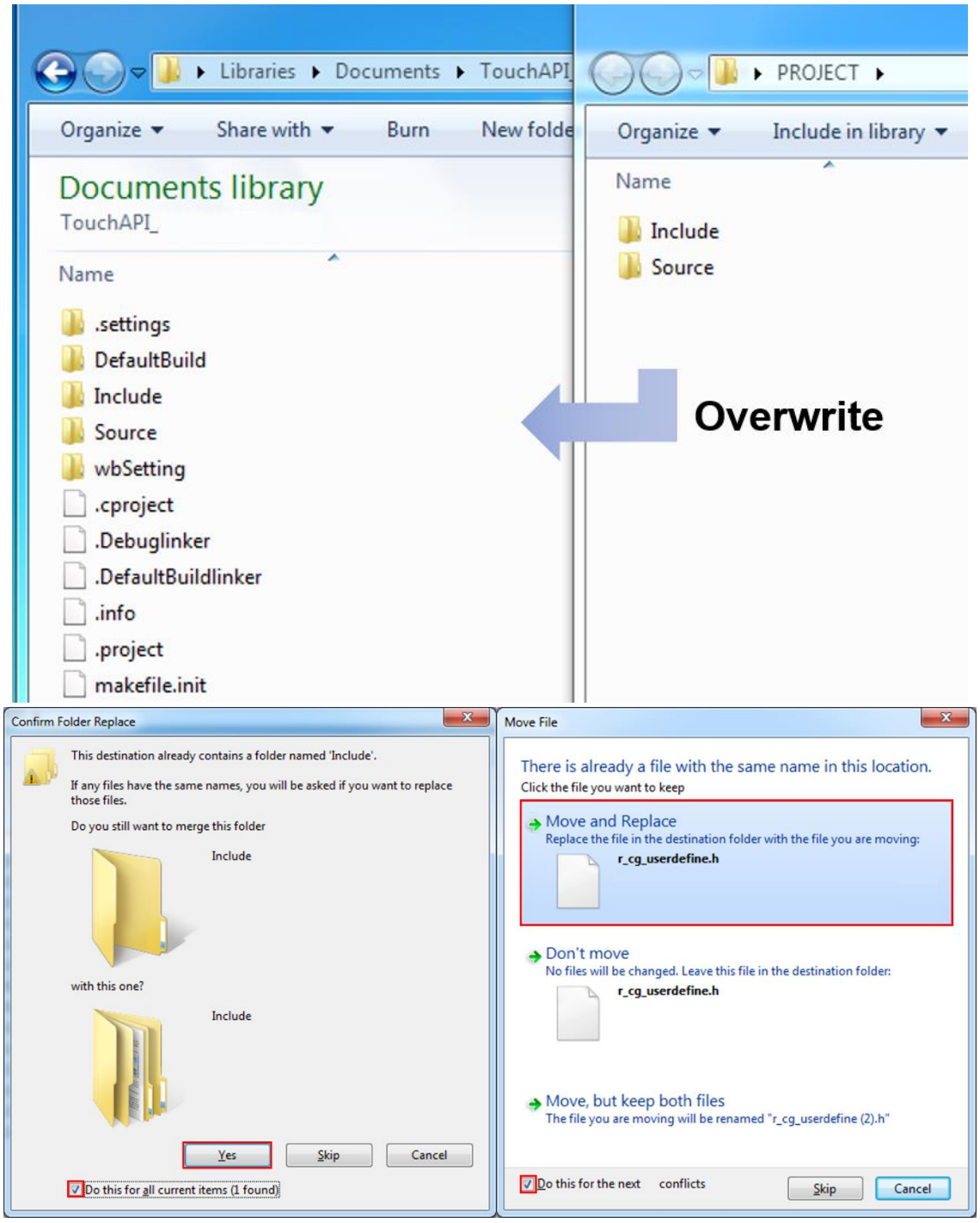

Figure 1-1 Overwrite the source file

Overwrite the attached source file with the project created by the Workbench6 First step guide wizard and run it in e2studio.

2 Summary

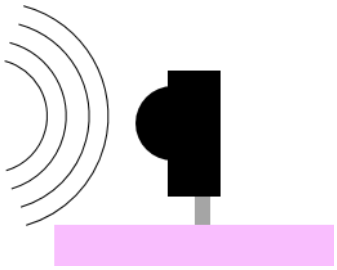

Figure 2-1 Infrared receiver

The IR receiver has the function of receiving the infrared wavelength from the IRED and recognizing it as a signal. However, due to the nature of communication using light as a medium, it is one of the parts to be careful when designing a communication system because it can be recognized as a signal by being influenced by fluorescent materials such as fluorescent lamp and sunlight.

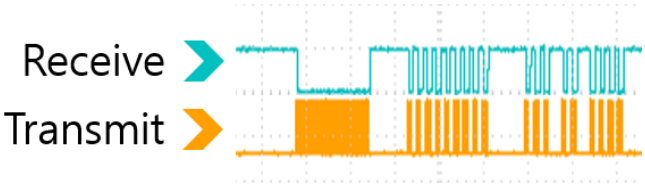

figure 2-2 적외선 통신 파형 예시

In case of receiving part, it goes HIGH normally, but goes down to LOW when receiving a signal with constant frequency, otherwise it returns to HIGH.

In the case of the receiving part, the internal circuit keeps the HIGH state at normal times, and when receiving the infrared signal with constant frequency, the power is cut off and it falls to 0 (LOW) or 1 (HIGH).

Carrier frequency	30 kHz	TSOP4130
	33 kHz	TSOP4133
	36 kHz	TSOP4136
	38 kHz	TSOP4138
	40 kHz	TSOP4140
	56 kHz	TSOP4156

Figure 2-3 Carrier frequency characteristics of TSOP4138

For the TSOP4138, an infrared receiver mounted on the NS-RX231, it is best to design this point because it has the characteristic that it can best catch the 38kHz frequency.

Refer **to 6. Schematic** Diagram for details.

In this document, we will implement simple remote control communication.

3 Project

3.1 Communication Structure

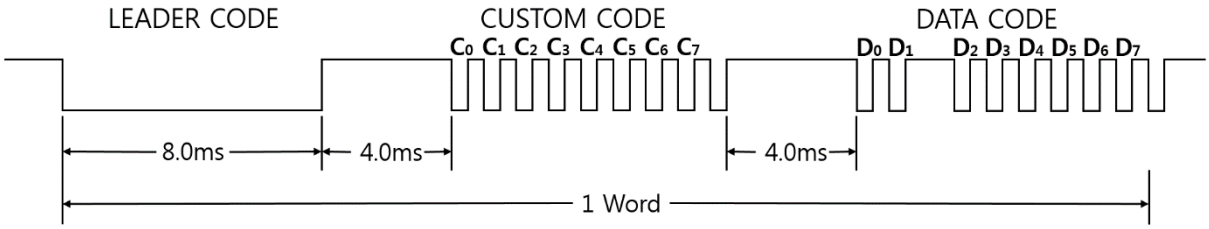

Figure 3-1 Communication Protocol

First, I made a project by taking the structure of the simplest format as an example.
The configuration consists of a reader code to indicate the start of communication, a custom code to distinguish the remote controller after that, and a data code to transmit the data.

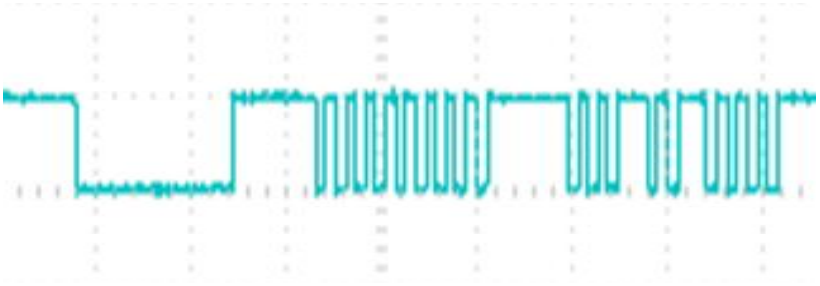

Figure 3-2 Infrared reception waveform

And the division of one bit between 0 and 1 is as follows

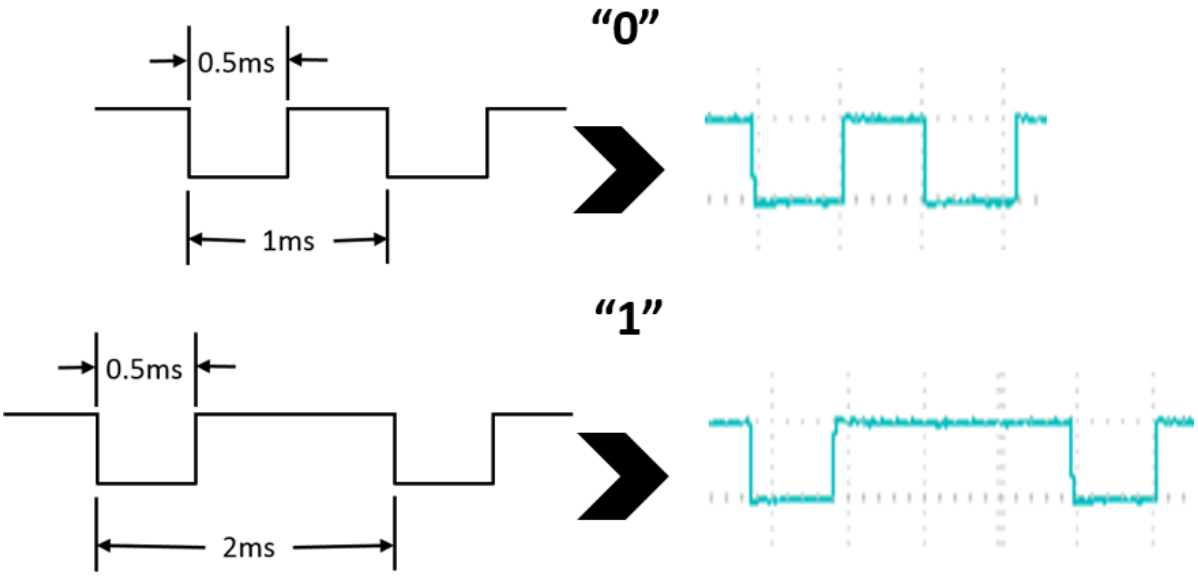

Figure 3-3 The distinction between 0 and 1 when receiving infrared communication

3.2 Source Code

```
if(IR_detectStart()){
    while(1U){
        data = IR_detectData();
        if(data == 1000){
            cnt = 0;
            break;
        }

        if(data != 0){
            buf[cnt] = data;
            cnt++;
            data = 0;
        }else if(cnt == 18){
            cnt = 0;
            data = 0;
            FallingFlg = 0;
            break;
        }
    }

    if(IR_checkStartBit(buf[0],240,260)){
        data = IR_convertData(buf,1);
        if(IR_checkCustomBit(data,0)){
            data = IR_convertData(buf,10);
            switch (data) {
                case 40:
                    index--;
                    break;
                case 24:
                    index = 3;
                    break;
                case 168:
                    index++;
                    break;
            }
            if(index < 0){
                index = 4;
            }else if(index > 4){
                index = 0;
            }
            PORTB.PODR.BYTE = ledNum[index];
        }
    }
}
```

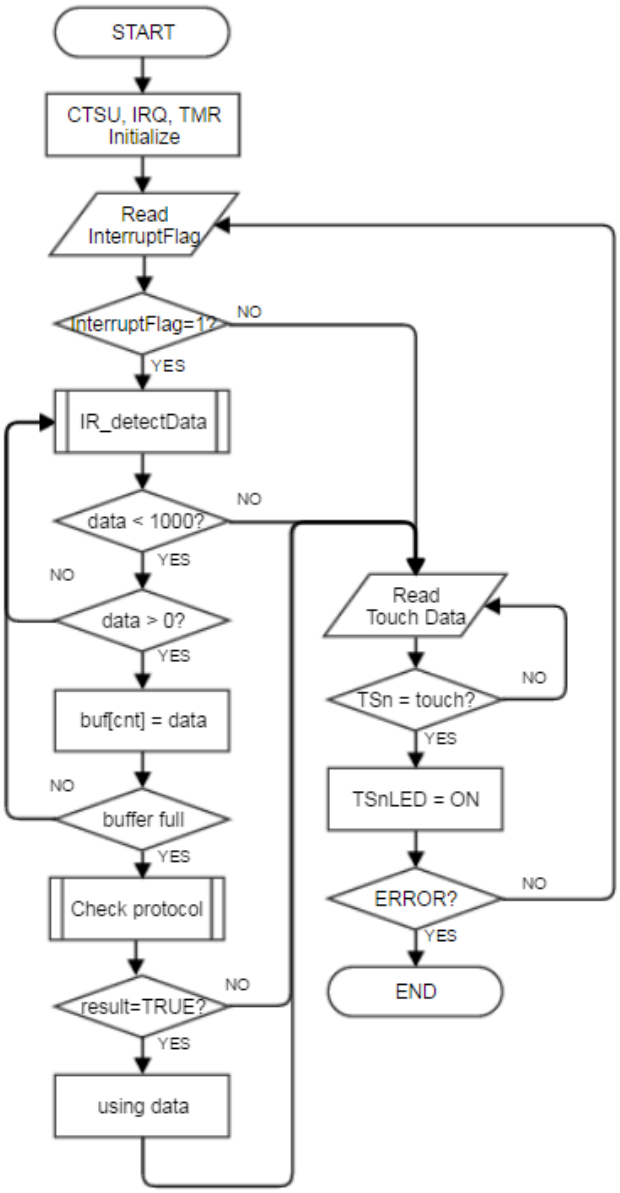

figure 3-4 소스코드 및 순서도

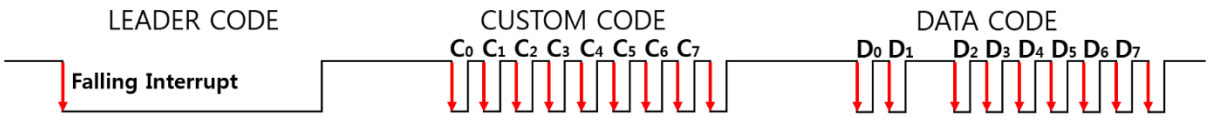

figure 3-5 Falling Interrupt

When the falling edge is detected, an interrupt is triggered. At this time, the timer count is incremented to recover the time until the interrupt again occurs. The timer is set to repeat once every 50us, and when 8ms and 4ms of the leader code are combined, the count is about 240 in 12ms.

3.3 detectIR.c

IR_convertData	
Parameters	uint16_t * buf Buffer uint8_t startIndex Start Index value
Return Value	0 ~ 255
Explanation	It calculates 8 bits of data from 8 falling edge interval data in the buffer. Falling edge interval data uses 8 from the starting index value. Ex) buf [startIndex] ~ buf [startIndex + 7] Falling edge interval data determines DISTRVAL as a threshold of 1 and 0. The default value of DISTRVAL is set to 40.

Code Example

```
if(IR_checkStartBit(buf[0],240,260)){    //check LeaderCode

    data = IR_convertData(buf,1);        //the index 1~8 buffer data convert to decimal data

}
```

IR_detectStart	
Parameters	Void
Return Value	TRUE 1 FALSE 0
Explanation	After recognizing the FallingFlg value, A value of 1 initializes the FallingFlg value and the timer count value and returns TRUE.

Code Example

```
void main(void)
{
    R_TMR0_Start();
    R_ICU_IRQ6_Start();

    while (1U)                                  /* Main loop */
    {
        if(IR_detectStart()){                   //Falling Edge detect


        }
    }
}
```

IR_detectData	
Parameters	Void
Return Value	0 ~ 65,535
Explanation	Count the time until the FallingFlg value becomes 1 again. When the value of FallingFlg becomes 1, it returns accumulated timer count value in between. If the timer count value exceeds 1000, it returns 1000.

Code Example

```
while(1U){
    data = IR_detectData();
    if(data == 1000){                       //if counter over the 1000 then; break
        cnt = 0;
        break;
    }

    if(data != 0){
        buf[cnt] = data;                    //insert data into a buffer
        cnt++;
        data = 0;
    }else if(cnt == 18){                    //buffer maximum then break
        cnt = 0;
        data = 0;
        FallingFlg = 0;
        break;
    }
}
```

IR_checkStartBit	
Parameters	uint16_t value Value to compare uint16_t compareMin Minimum comparison value uint16_t compareMax Maximum comparison value
Return Value	TRUE 1 FALSE 0
Explanation	Returns TRUE if the value to be compared is between the minimum and maximum values of the condition.

Code Example

```
if(IR_checkStartBit(buf[0],250,260)){   //check LeaderCode

}

for(i=0;i<18;i++){                      //initialize buffer
    buf[i] = 0;
}
```

IR_checkCustomBit	
Parameters	uint16_t value Value to compare uint8_t compareVal Comparison value
Return Value	TRUE 1 FALSE 0
Explanation	Returns TRUE if the value to be compared is the same as the condition comparison value.

Code Example

```
if(IR_checkStartBit(buf[0],250,260)){       //check LeaderCode
    data = binToDec(buf,1);                 //the index 1~8 buffer binary data return to decimal data
    if(IR_checkCustomBit(data,0)){          //check CustomCode
        data = binToDec(buf,10);            //the index 10~17 buffer binary data return to decimal data
        requestData(data);                  //using data
    }
}

for(i=0;i<18;i++){                          //initialize buffer
    buf[i] = 0;
}
```

4 Debugging

Figure 4-1 Connect NS-RX231 to power adapter and E1 debugger.

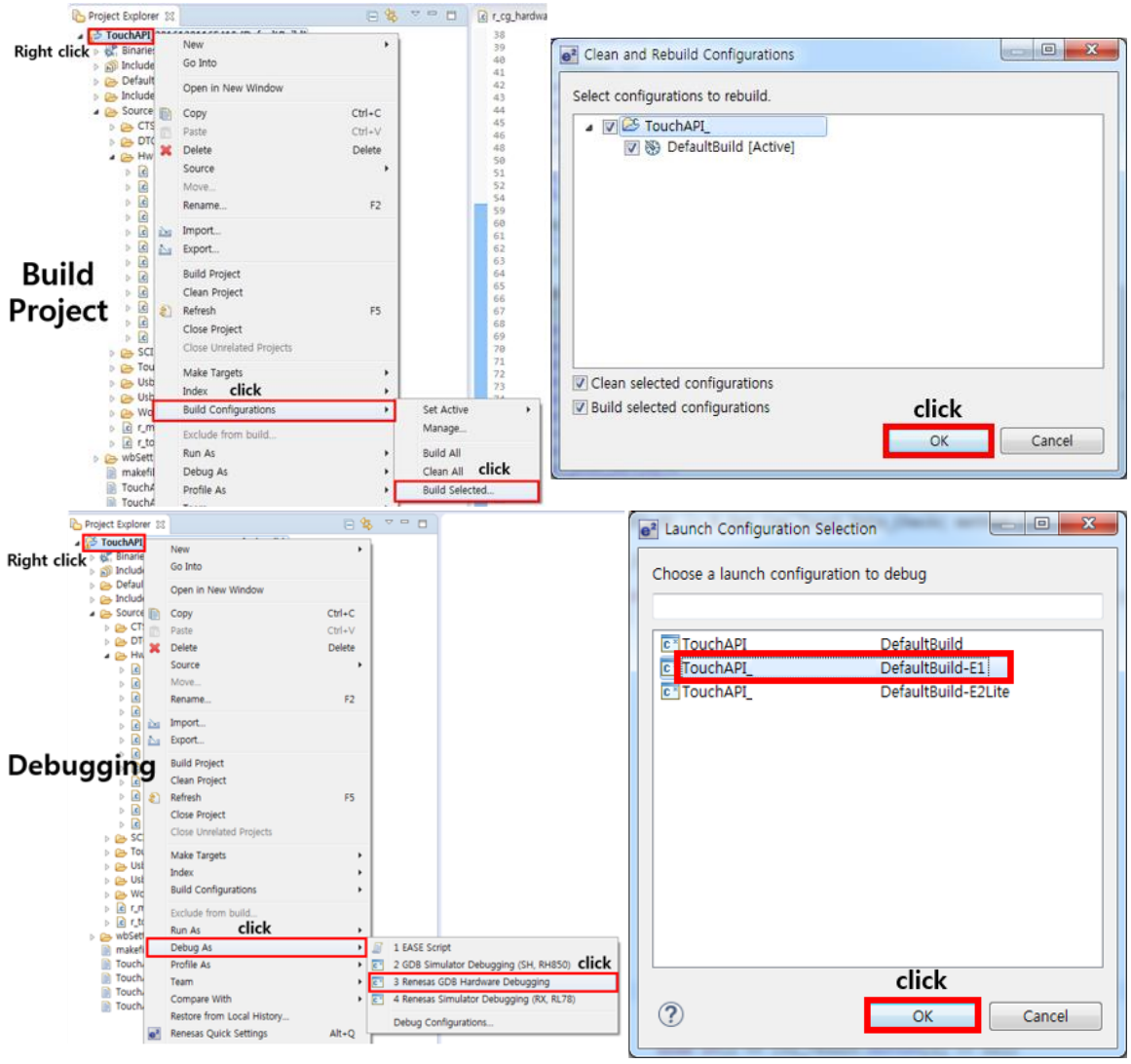

Figure 4-2 Project Build and Debugging

5 Execution

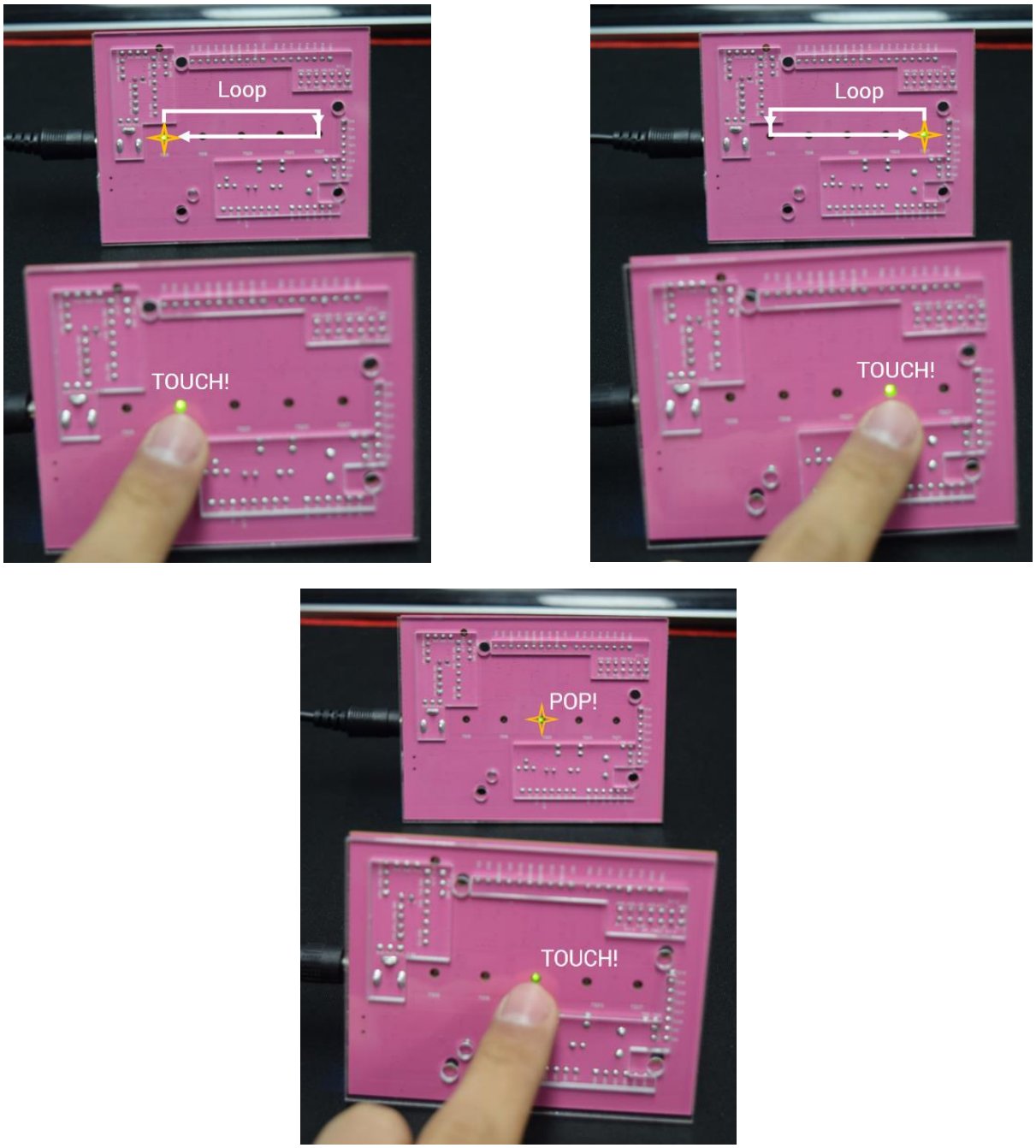

Figure 5-1 Response when receiving infrared

6 Schematic

The picture below is a circuit diagram of the infrared remote control receiver.

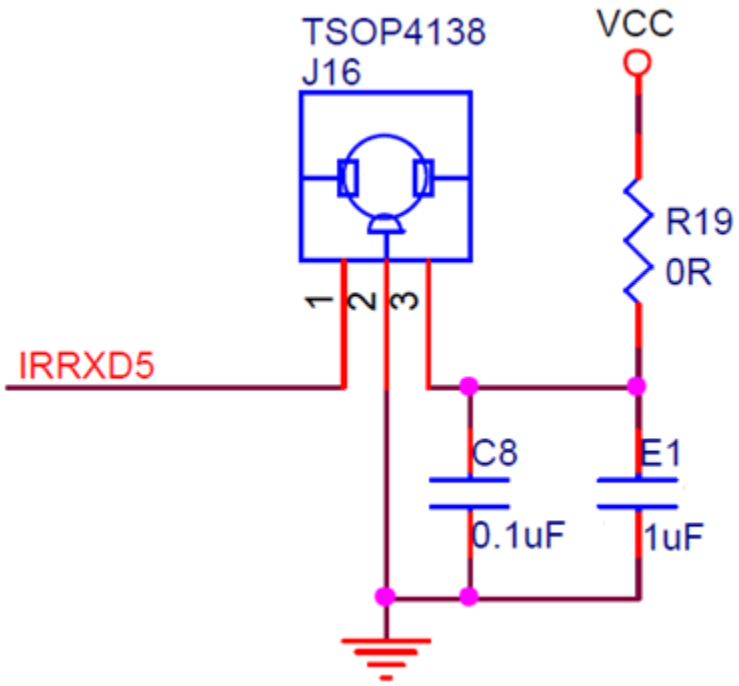

Figure 6-1 Infrared receiver circuit of NS-RX231