Software Guide for NS-RX231 <BLUETOOTH>

Contents

1 Load the Project

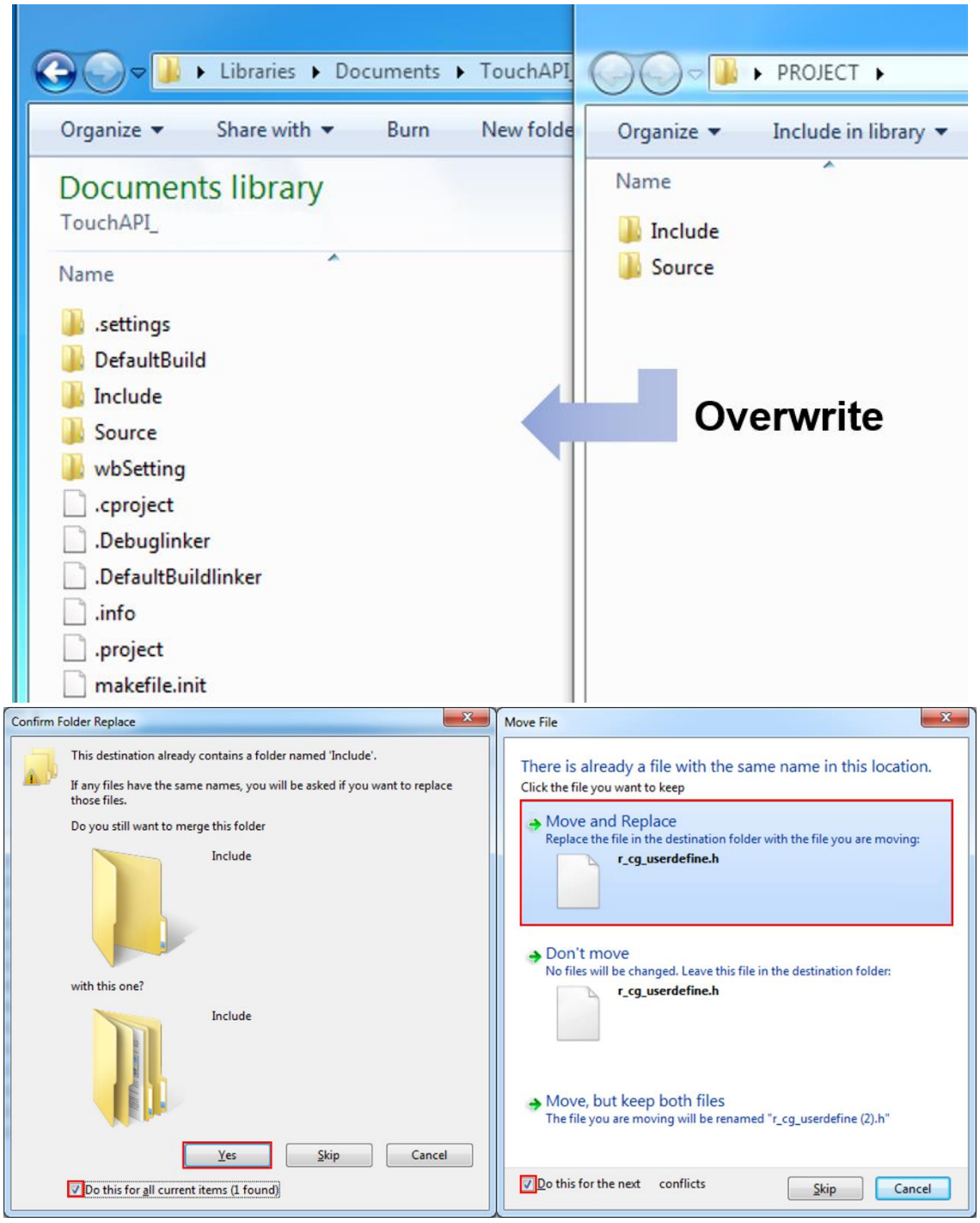

figure 1-1 Overwrite the source file

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

2 Summary

For Bluetooth communication, the HC-06 module is used.

And to prevent confusion with other modules, the configuration of the Bluetooth module using the AT command in the terminal program is changed.

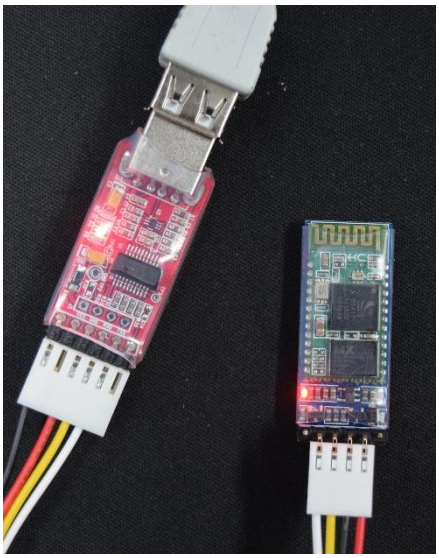

figure 2-1 HC-06 Connecting the Bluetooth Module to the Computer

https://sites.google.com/site/terminalbpp/Terminal20141030.zip?attredirects=0&d=1

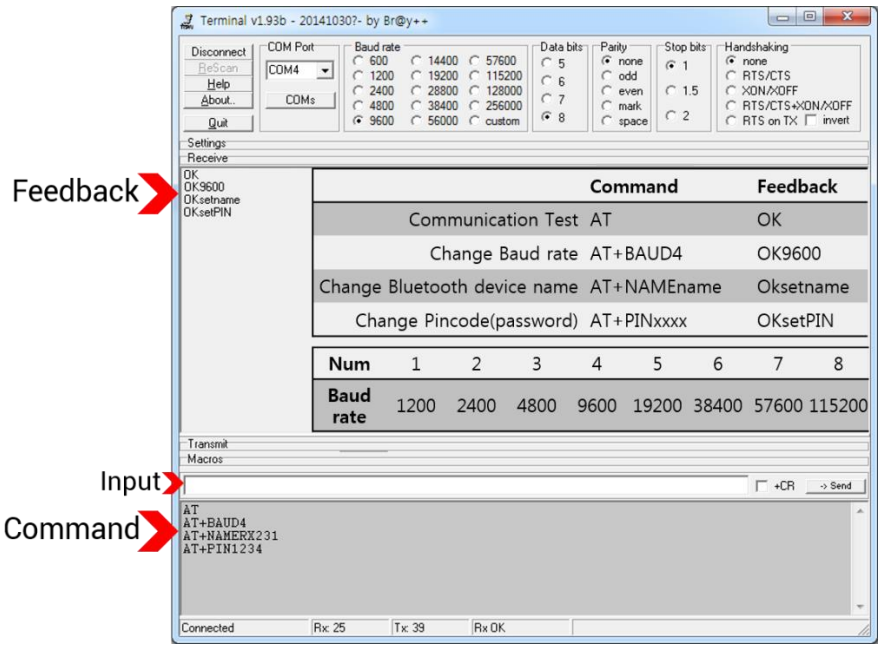

figure 2-2 Terminal Program

3 Source Code

3.1 Transmit

```
if (_0_SUCCESS == R_Set_Cap_Touch_Result_Create( method ))
{
    ts_result = R_Get_Cap_Touch_Result( method );

    if (1 == (ts_result.button[0] >> 15))
    {
        sci1_print(&send_data[0],1);
        PORTB.PODR.BYTE = 0x01;
    }
    else if(1 == (ts_result.button[1] >> 0))
    {
        sci1_print(&send_data[1],1);
        PORTB.PODR.BYTE = 0x02;
    }
    else if(1 == (ts_result.button[1] >> 6))
    {
        sci1_print(&send_data[2],1);
        PORTB.PODR.BYTE = 0x08;
    }
    else if(1 == (ts_result.button[1] >> 7))
    {
        sci1_print(&send_data[3],1);
        PORTB.PODR.BYTE = 0x20;
    }
    else if(1 == (ts_result.button[1] >> 11))
    {
        sci1_print(&send_data[4],1);
        PORTB.PODR.BYTE = 0x40;
    }
    else if(1 == (ts_result.button[1] >> 14))
    {
        sci1_print(&send_data[5],1);
        PORTB.PODR.BYTE = 0x80;
    }
}
```

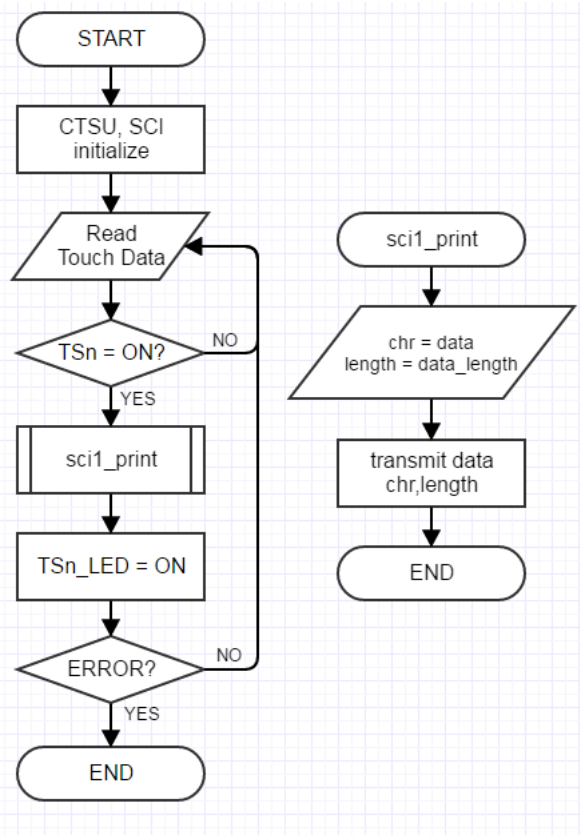

figure 3-1 Source code and flowchart

The above source code sets the LED corresponding to the button, and transmits the data n times corresponding to TSn through UART communication.

3.2 Receive

```
static void r_sci1_callback_receiveend(void) {
    /* Modifiable code [start] */
    switch (g_rx_char[0]) {
        case 1:
            PORTB.PODR.BYTE = 0x00; //reset
            break;
        case 15:
            PORTB.PODR.BYTE = 0x01; //TS15
            break;
        case 16:
            PORTB.PODR.BYTE = 0x02; //TS16
            break;
        case 22:
            PORTB.PODR.BYTE = 0x08; //TS22
            break;
        case 23:
            PORTB.PODR.BYTE = 0x20; //TS23
            break;
        case 27:
            PORTB.PODR.BYTE = 0x40; //TS27
            break;
        case 30:
            PORTB.PODR.BYTE = 0x80; //TS30
            break;
        default:
            break;
    }
    /* Modifiable Code [end] */

    R_SCI1_Serial_Receive((uint8_t *)&g_rx_char, 1);
    sci1_rxdone = 0;
}
```

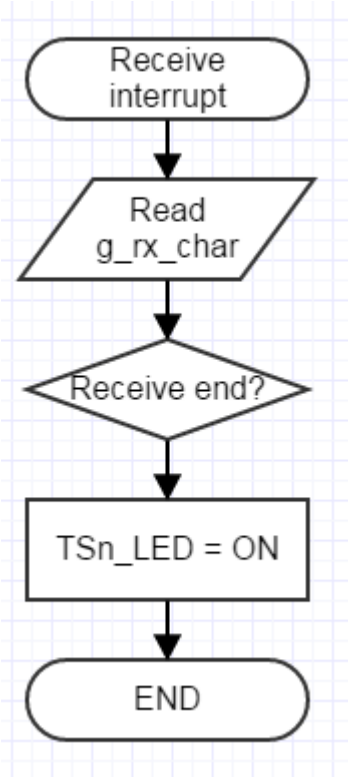

figure 3-2 Source code and flowchart

The receive interrupt occurs when data is full in the buffer that stores the data. When all data is received, to proceed the data the program shown in figure 3-2 start to run.

When receiving data 1, the LED is reset and all are turned off,

15 is received, B0 is controlled so that only the LED corresponding to TS15 is turned on,

23 controls to set B5 in port5B and turn on only the LED corresponding to TS23.

4 Debugging

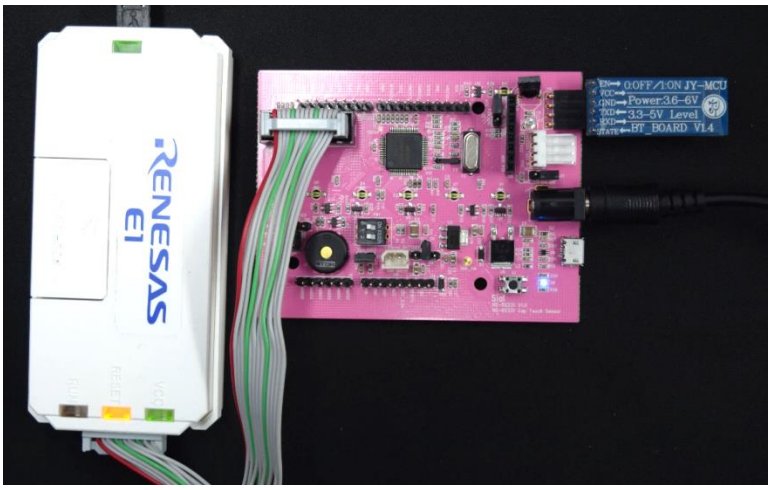

figure 4-1 Connect NS-RX231 to power adapter and E1 debugger.

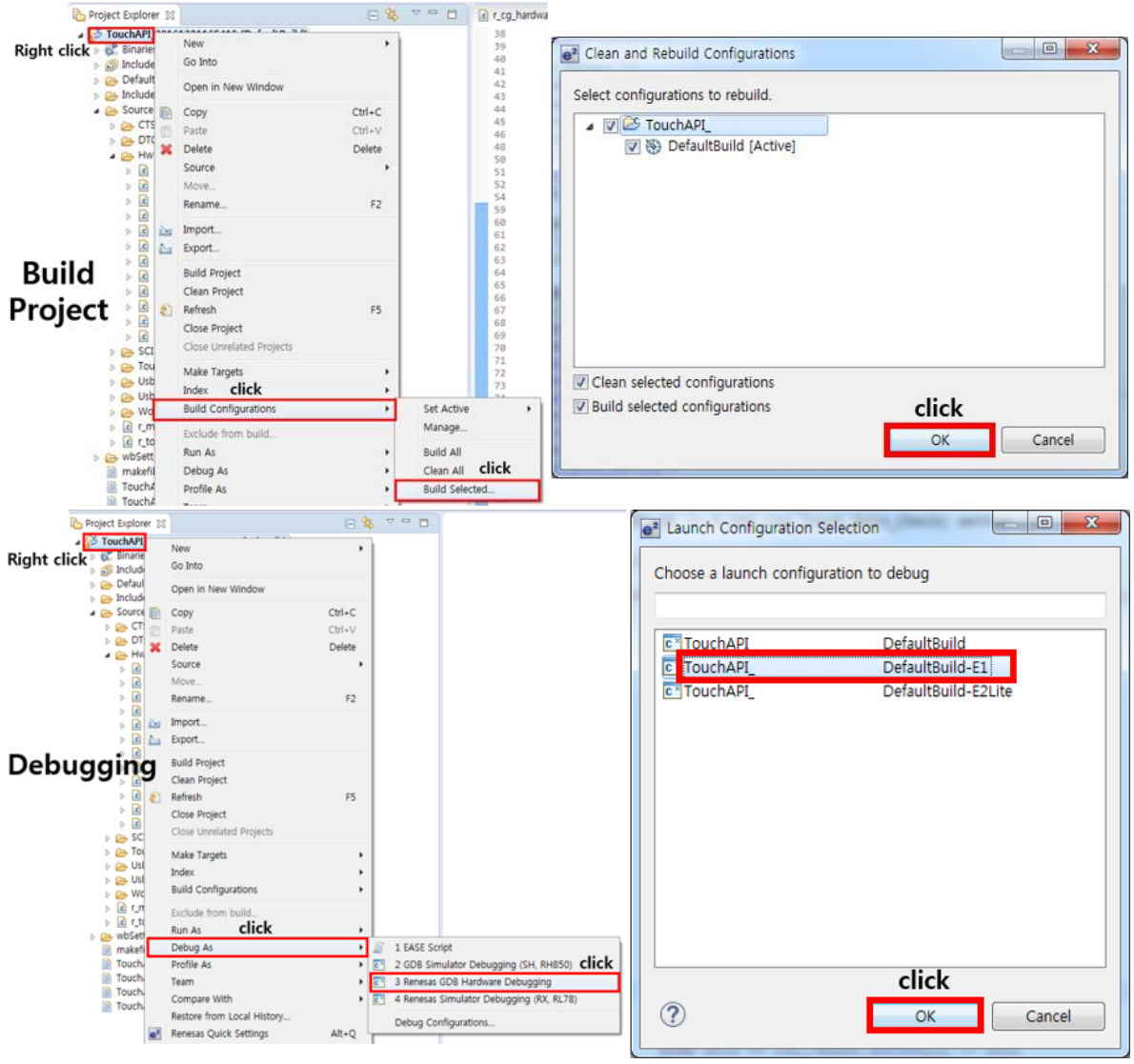

figure 4-2 Project build and debugging

5 Execution

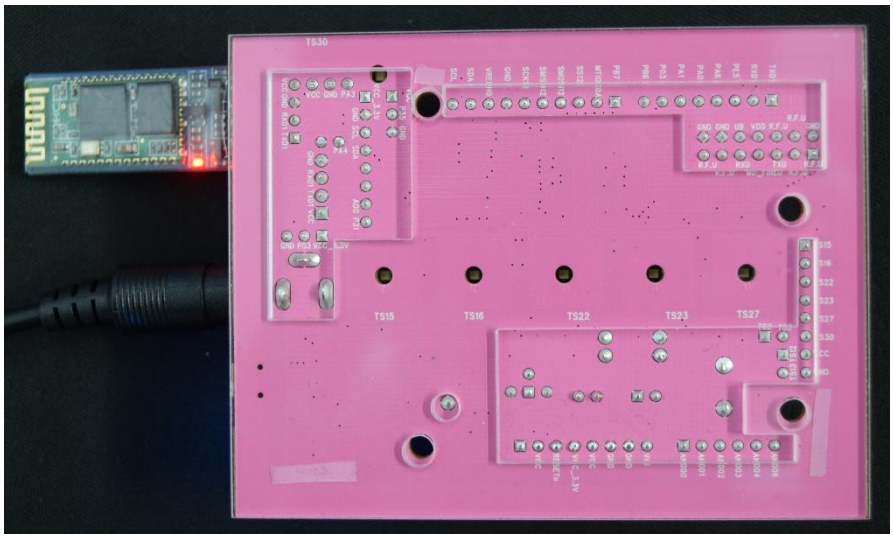

figure 5-1 NS-RX231 with Bluetooth module

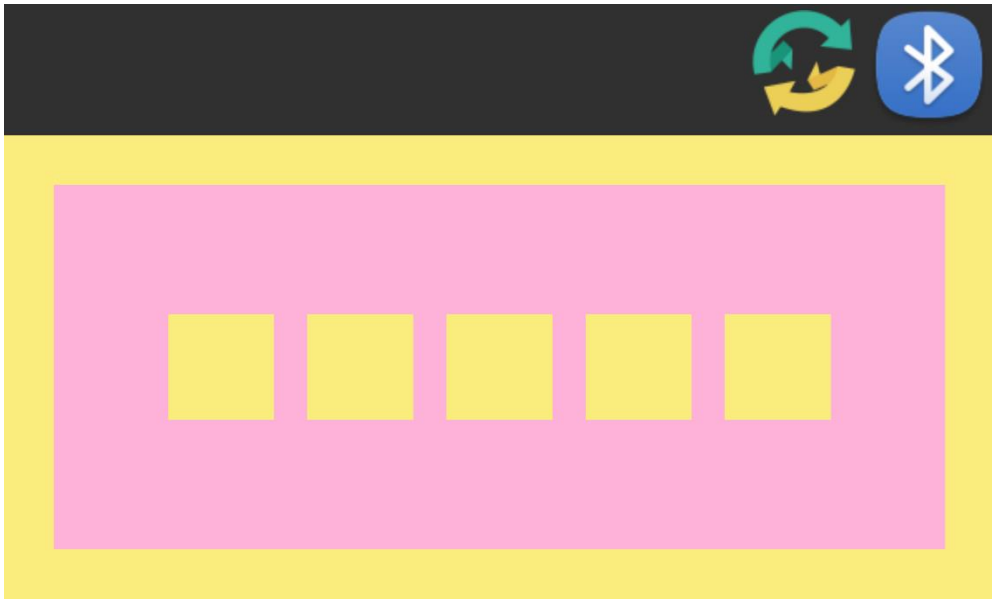

figure 5-2 Android Application

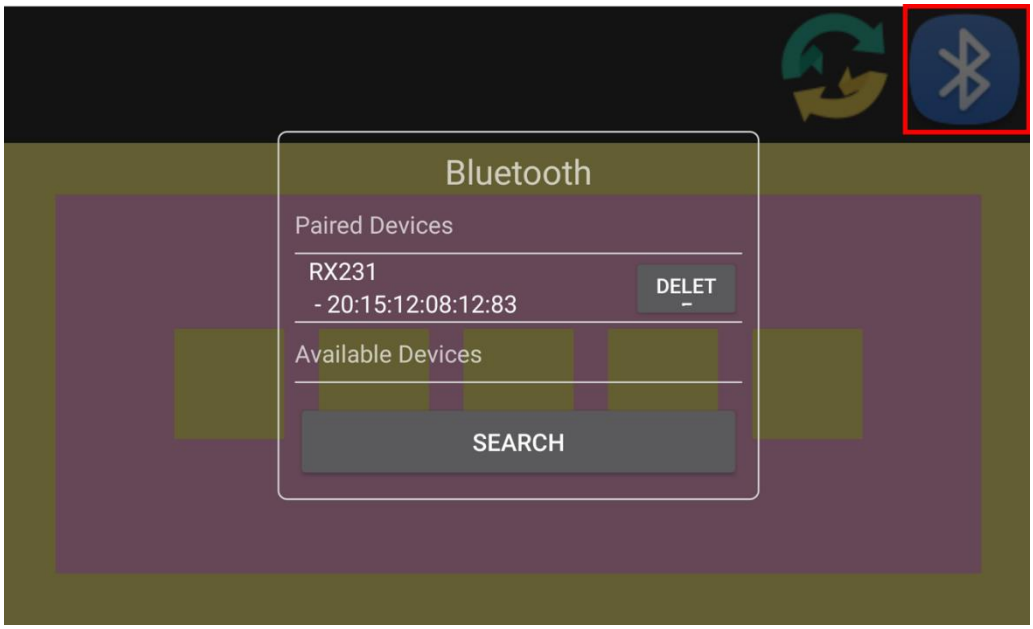

figure 5-3 Android Application

Press the top right button to connect to the Bluetooth module. Press the SEARCH button to find the module to be paired and enter the password to connect.

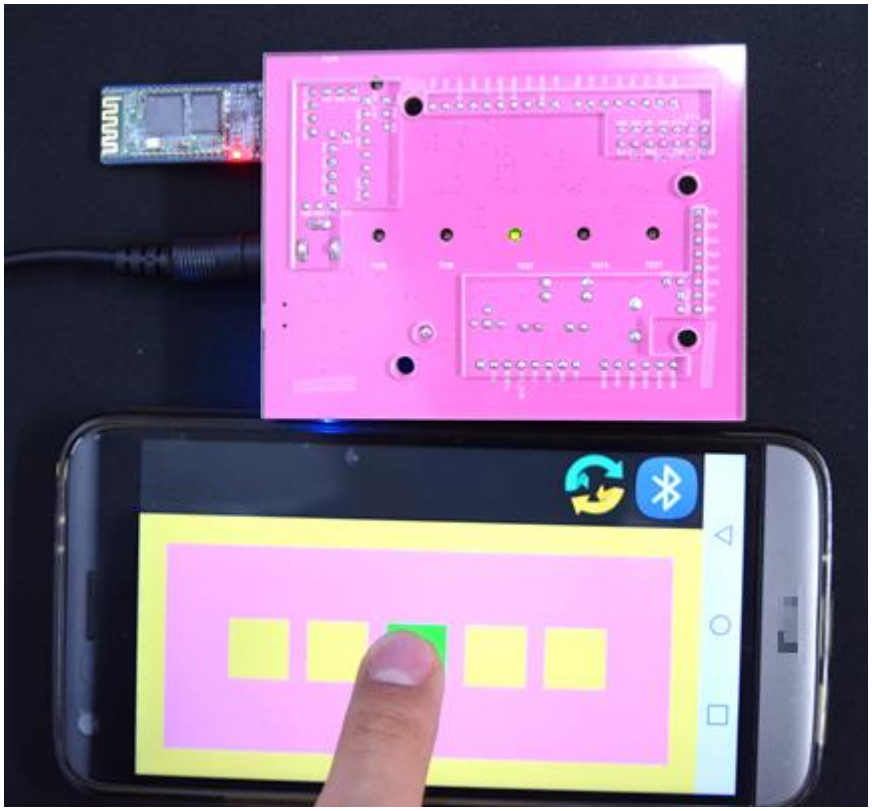

figure 5-4 Android Application test

Pressing the application's button will light the NS-RX231's LED, and pressing the touch button will illuminate the corresponding button in the application.

6 Schematic

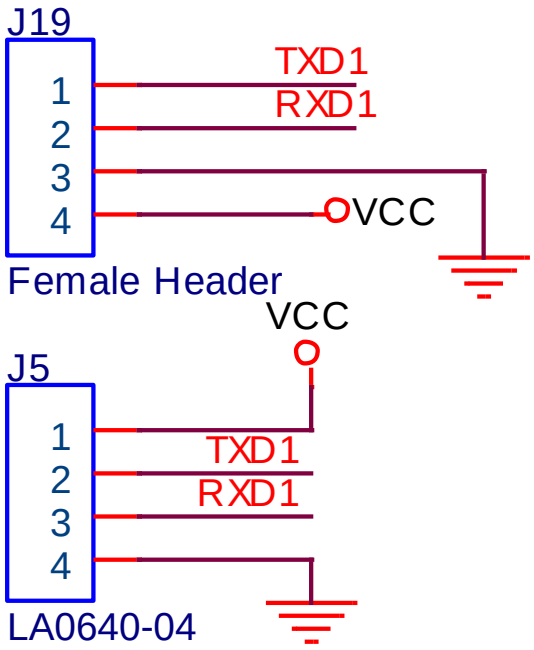

figure 6-1 UART communication circuit diagram of NS-RX231