

NS-RX231 のソフトウェアガイド

<BLUETOOTH 編>

目次

1 プロジェクトのインポート.....	2
2 概要.....	3
3 ソースコード.....	4
3.1 送信.....	4
3.2 受信.....	5
4 デバッグ.....	6
5 実行.....	7
6 回路図.....	9

1 プロジェクトのインポート

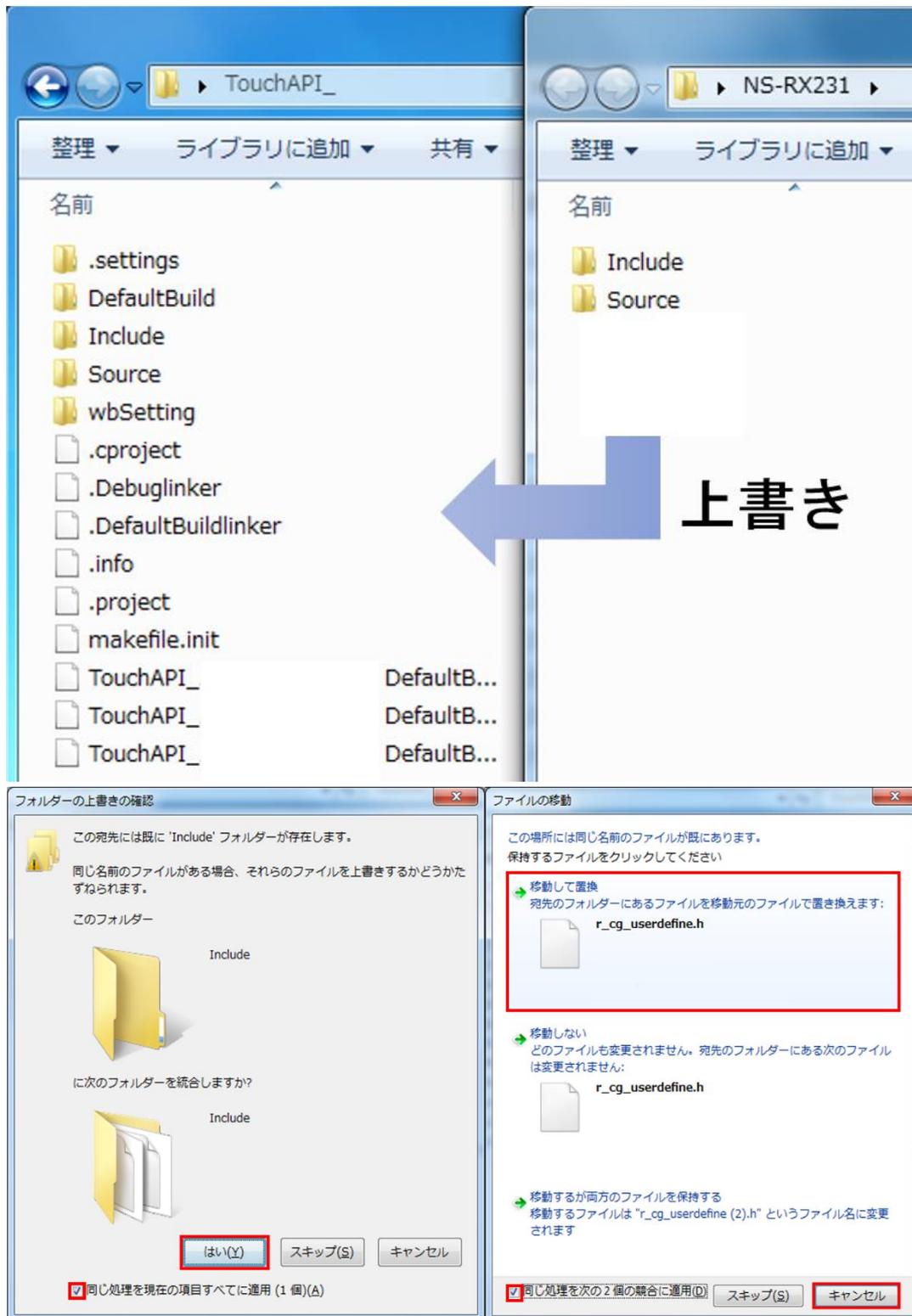


図 1-1 ソースファイルを上書き

添付されたソースファイルをWorkbench6 First step guideに従って、ウィザードで作成したプロジェクトに上書きした後、e2studioを実行します。

2 概要

Bluetoothでの通信のために、HC-06モジュールを使用しました。
 そして、他のモジュールと混乱を防ぐために、ターミナルソフトでATコマンドを利用して、Bluetooth
 hモジュールの設定を変えます。

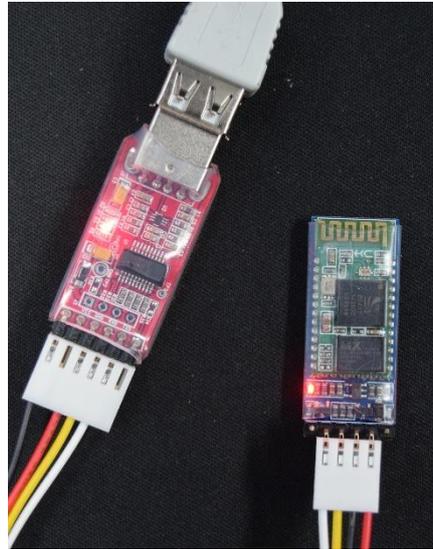


図 2-1 HC-06のBluetoothモジュールとコンピュータを接続

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

応答

Command	Feedback
Communication Test	AT OK
Change Baud rate	AT+BAUD4 OK9600
Change Bluetooth device name	AT+NAMEName Oksetname
Change Pincode(password)	AT+PINxxxx OKsetPIN

入力
コマンド

```

AT
AT+BAUD4
AT+NAMEX231
AT+PIN1234
    
```

図 2-2 ターミナルソフトの操作画面イメージ

3 ソースコード

3.1 送信

```

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;
    }
}

```

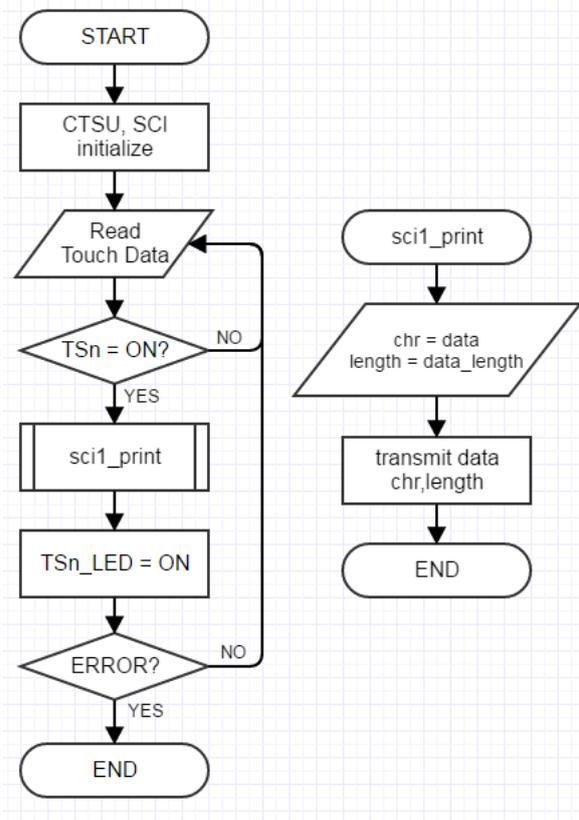


図 3-1 ソースコードとフローチャート

上記のソースコードは、条件式に基づいて検出されたタッチボタンに対応する LED を点灯し、タッチボタンの番号 (ex : タッチボタン TS15=15) を sci1 (UART 通信) 経由で送信します。

3.2 受信

```

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);
    scil_rxdone = 0;
}

```

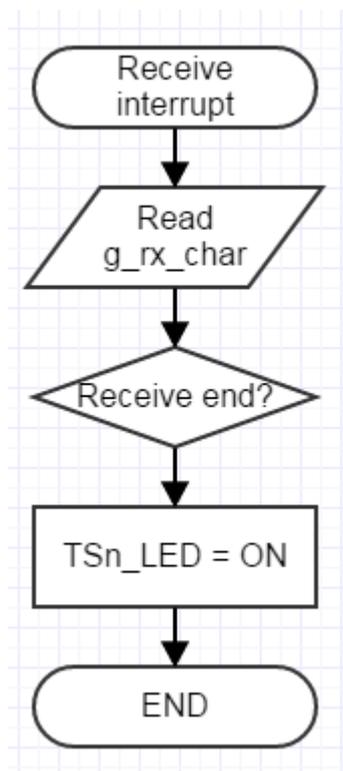


図 3-2 ソースコードとフローチャート

受信割り込みが発生して、指定した数のデータ受信が完了すると、受信したデータの処理のために、図3-2のプログラムが実行されます。

データ1を受信したときにはLEDをすべて消灯し、15を受けたときはPortBのビット0をセットしてTS15に対応するLEDのみ点灯、23を受信したときはPortBのビット5をセットしてTS23に対応するLEDのみ点灯します。

4 デバッグ



図 4-1 NS-RX231に電源アダプタとE1デバッガを接続した様子

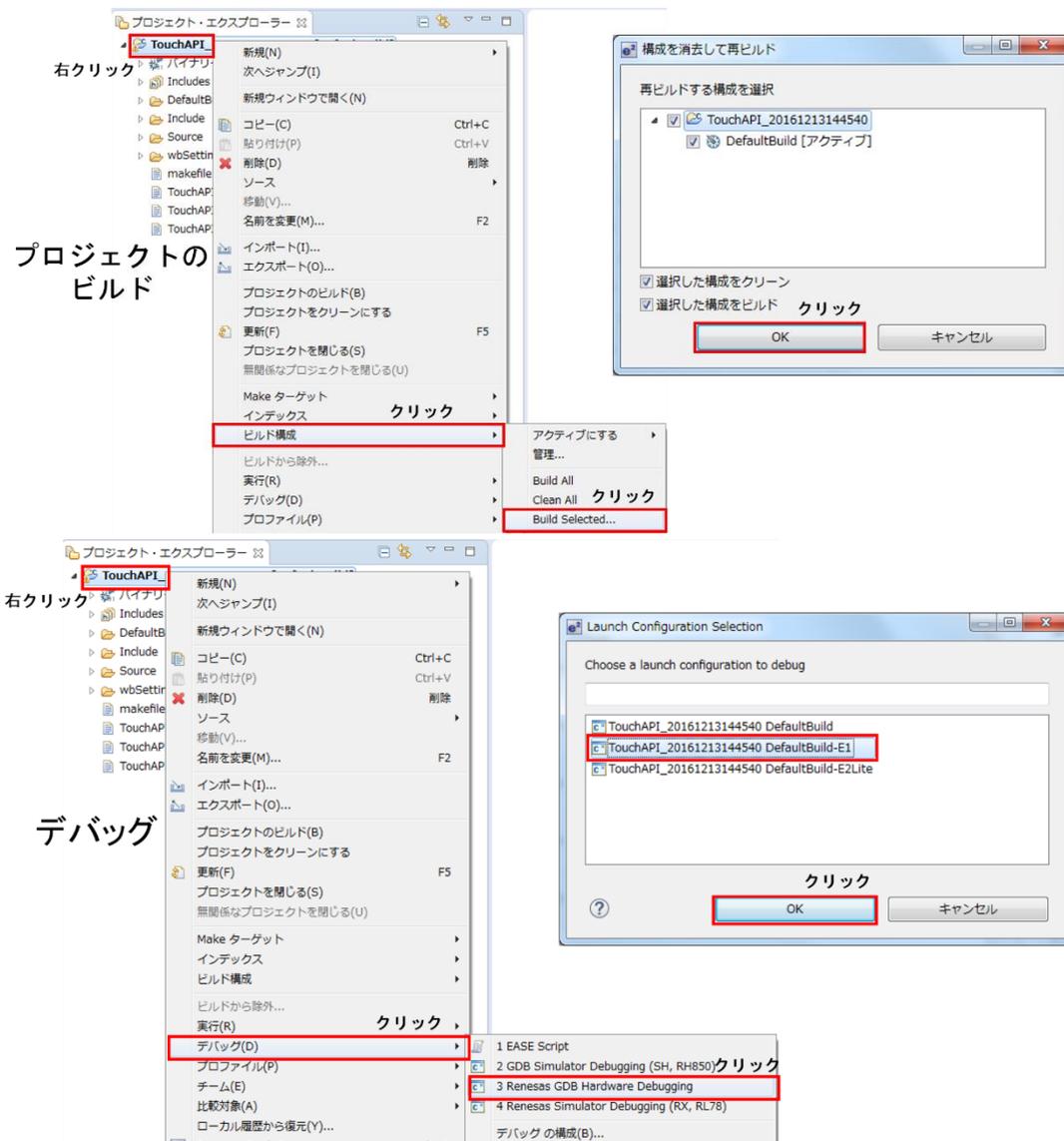


図 4-2 プロジェクトのビルドとデバッグ

5 実行

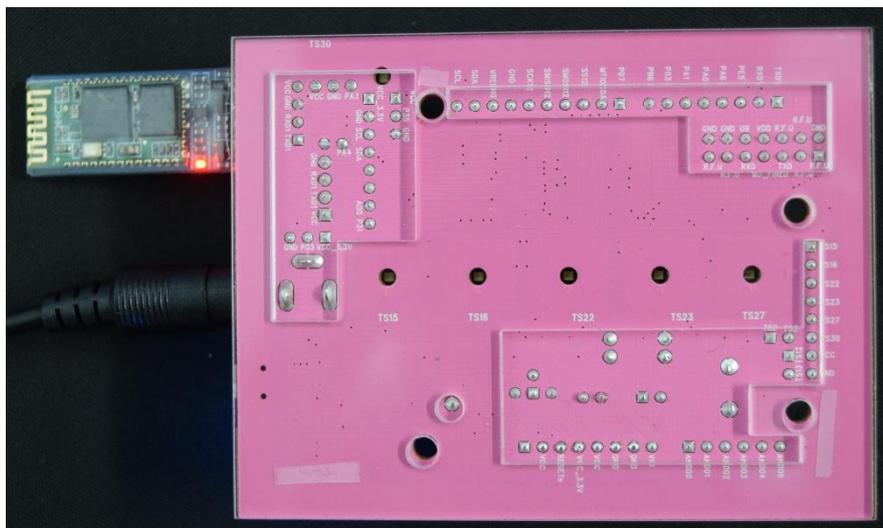


図 5-1 Bluetoothモジュールを接続したNS-RX231



図 5-2 Androidアプリケーション画面

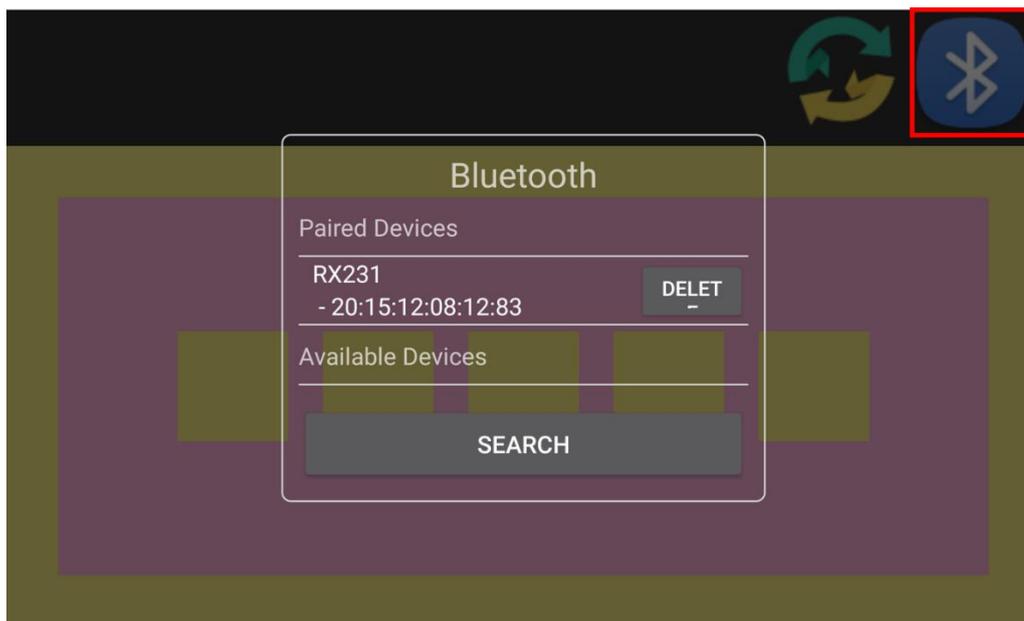


図 5-3 Androidアプリケーション

右上のボタンを押して、Bluetoothモジュールとの接続を試みます。SEARCHボタンを押してペアリングするモジュールを発見した後、パスワードを入力して接続します。

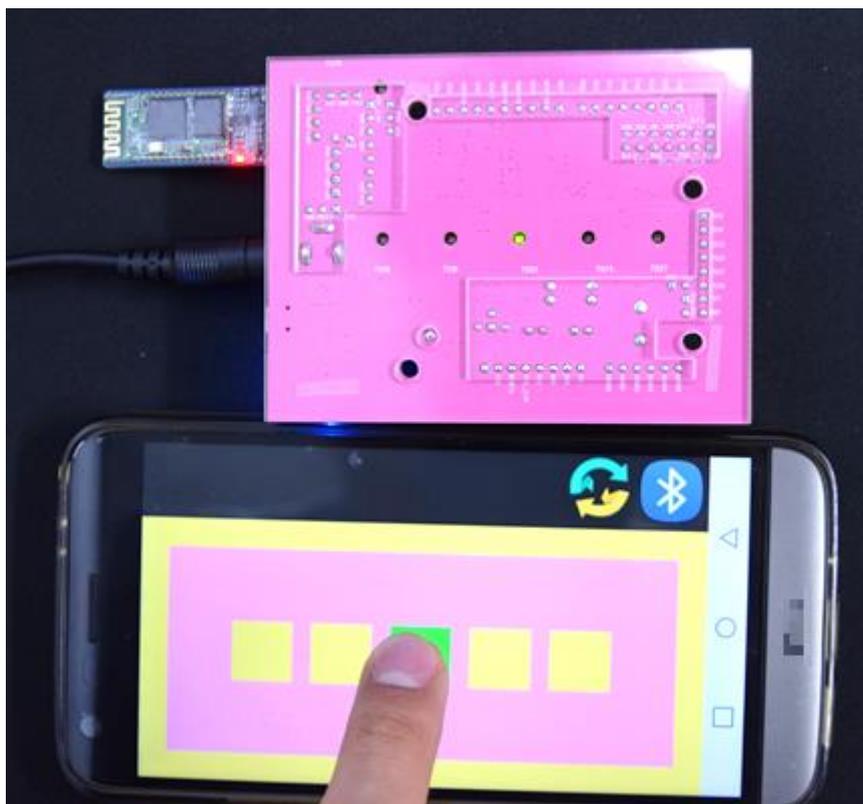


図 5-4 Androidアプリケーションのテスト

Android アプリケーションのボタンを押すと、NS-RX231 の LED が点灯し、タッチボタンを押すと、Android アプリケーションの対応するボタンが点灯します。

6 回路図

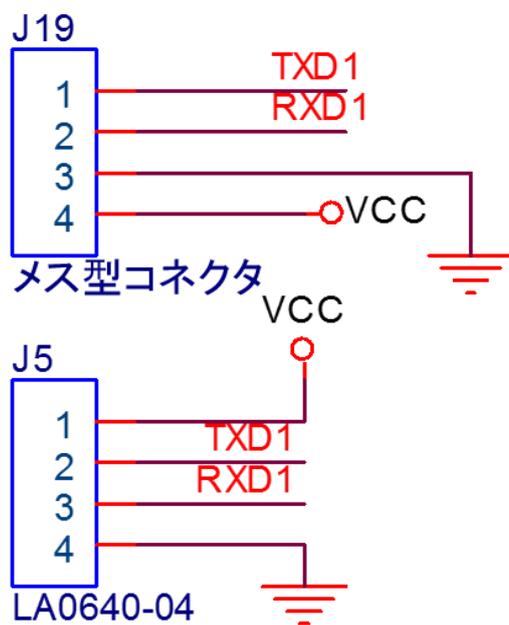


図 6-1 NS-RX231のUART通信回路図