NS-RX231 のソフトウェアガイド

〈スピーカー編〉

目次

1 プロジェクトのインポート

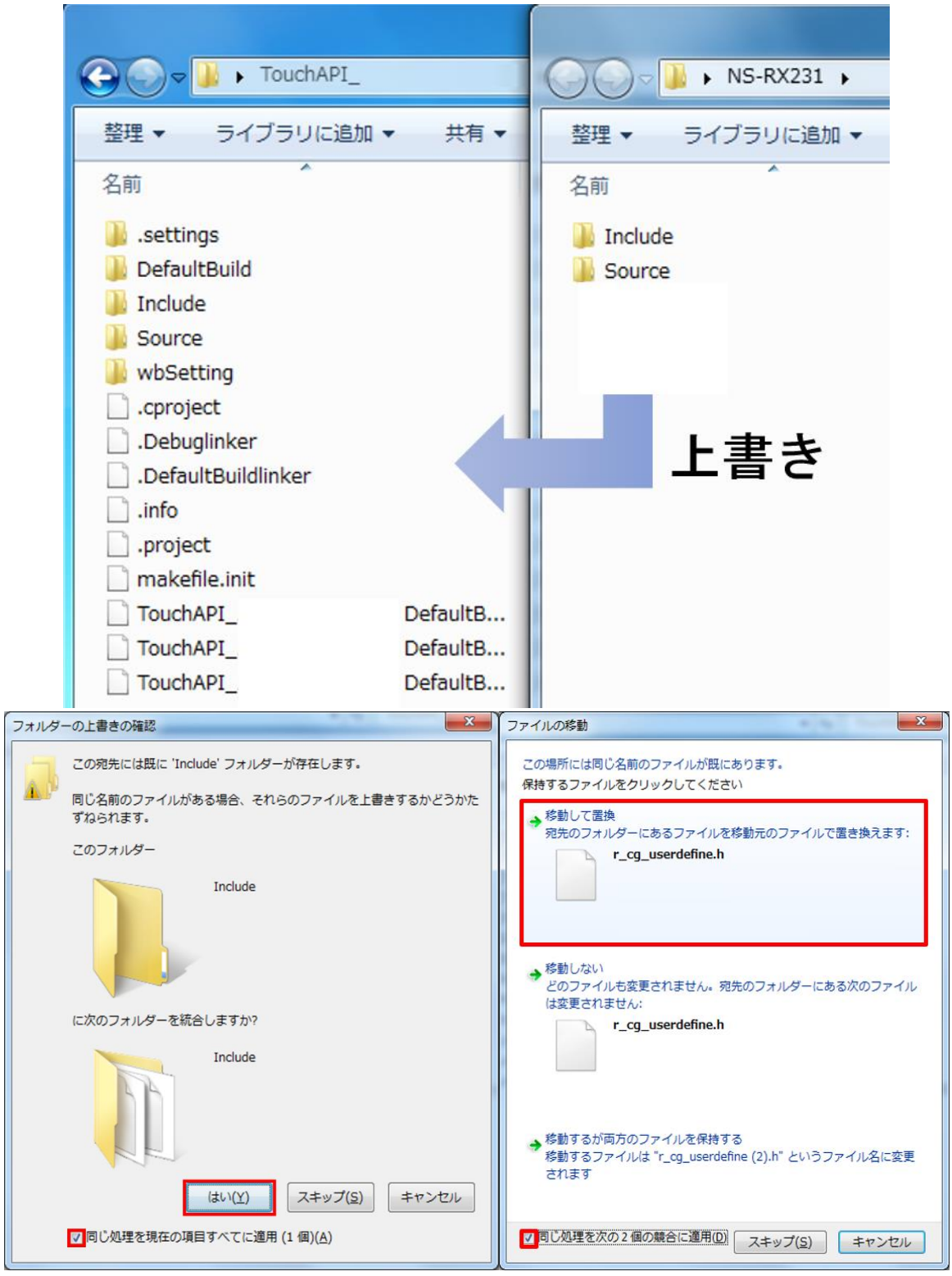

図 1-1 ソースファイルを上書き

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

8ビットタイマが4本あれば、2本ずつ束ねることで、二つの16ビットタイマとして使用することができます。16ビットタイマの最大の利点は、8ビットタイマはTCNTの値の範囲は0～255（2^8-1）であるのに対して、0～65535（2^{16}-1）になることです。指定できる範囲が広くなるため、タイマの設定時間を細かく微調整することができます。

タイマは、内部のカウンタの値を増加させ、一定の値になった時点で割り込みを発生させることができます。たとえば、オーバフロー割り込みのような場合には、カウンタの値が最大値(65535)を越えた（オーバフロー）時点で割り込みが発生します。割り込みでポートの出力を反転させるようにしてカウント値とのタイミングを見てみます。

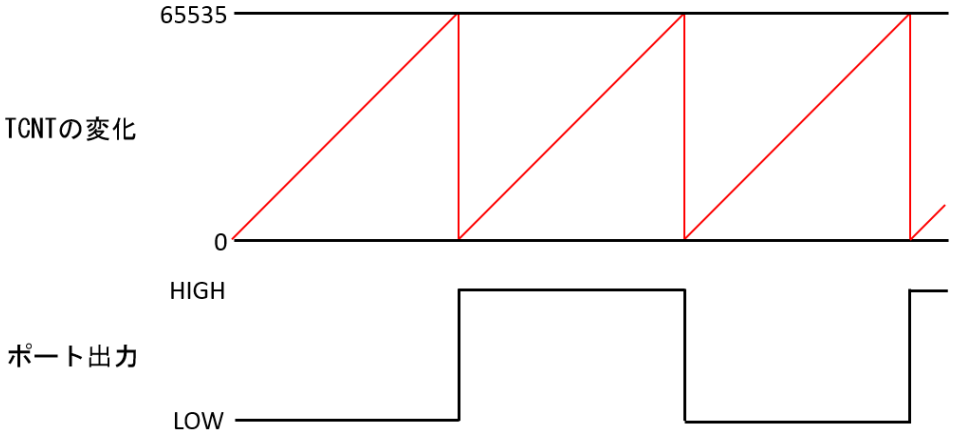

図 2-1 オーバフロー割り込み使用時の例

一致割り込みの場合、比較値を指定して、カウンタが指定の比較値と一致した時点で割り込みが発生します。

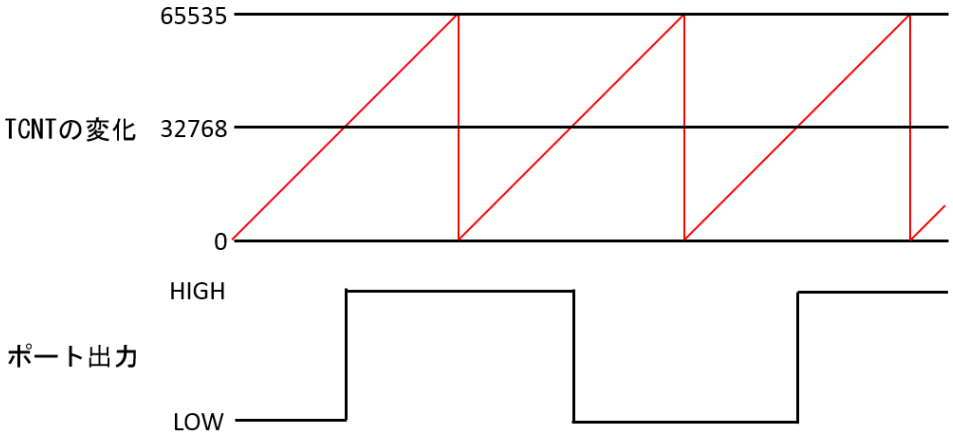

図 2-2 一致割り込み使用時の例（タイマをクリアしないとき）

一致割り込みを使用しますが、同時にカウンタをリセットしないと、図 2-2 に示すようになります。オーバーフロー割り込みに比べて発生時期が早いだけでは期待する結果がありません。

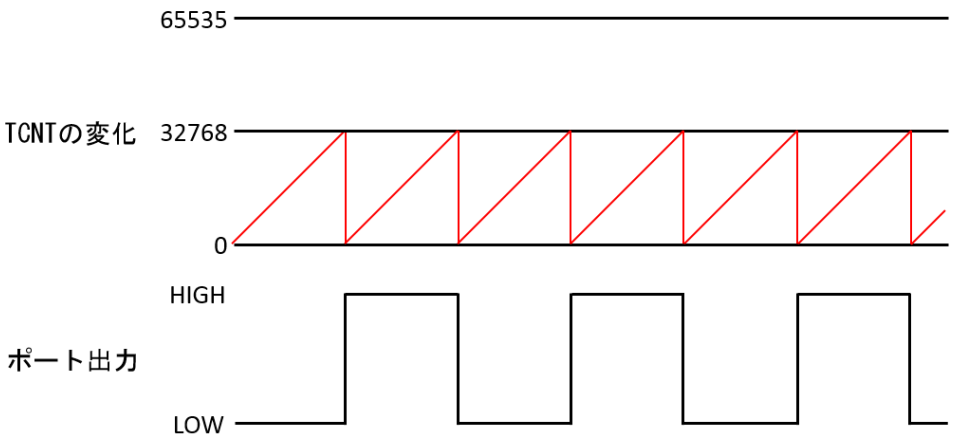

図 2-3 一致割り込みの例（カウンタをリセット）

上の図のように一致割り込みと同時にカウンタをリセットすれば，目的の間隔で割り込みを発生させることができます。

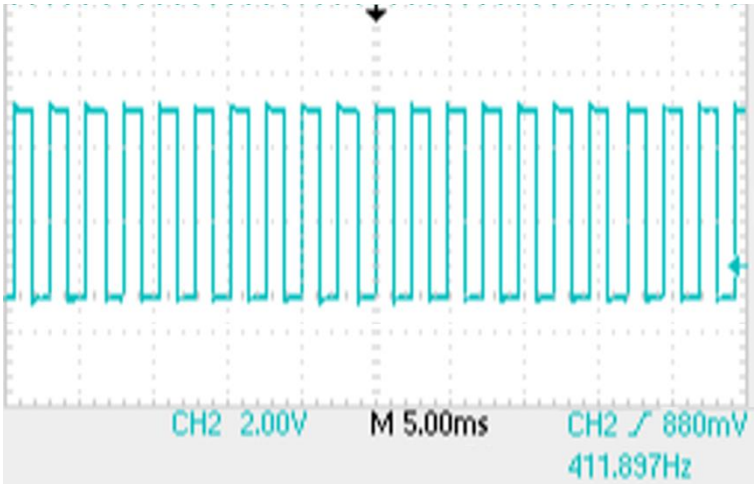

図 2-4 値32768を入れたとき、オシロスコープの波形

$$\text{TCORA 値} \times \text{目的の周波数[Hz]} = \text{タイマの動作周波数[Hz]}$$

$$7\text{FFF}(32768-1) \times 411.897[\text{Hz}] = 13497040[\text{Hz}]$$

ピアノ音階の中央ド（C）を例に挙げれば、周波数は 523Hz となります。1Hz は 1 秒に 1 回振動するという意味を持っています。つまり、ド（C）を表現するには、1 秒間に 523 回振動させなければいけません。音を表現するためには、タイマのクリア時間を調節することで実現します。

そして、次の計算式から目的の周波数を作成することができます。

$$\frac{\text{タイマの動作周波数[Hz]}}{\text{目的の周波数[Hz]}}/\text{プリスケーラ} = \text{TCORA 値}$$

$$\frac{13497040[\text{Hz}]}{523[\text{Hz}]}/1 = 25806$$

3 ソースコード

```
void request(touch_result_t ts_result){
    uint8_t i, count = 0;
    uint8_t ledValue[16] = {0,};

    ledValue[0] = 0x01;
    ledValue[1] = 0x02;
    ledValue[7] = 0x08;
    ledValue[8] = 0x20;
    ledValue[12] = 0x40;
    ledValue[15] = 0x80;

    PORTB.PODR.BYTE = 0x00;

    if(ts_result.button[0] >> 15){
        PORTB.PODR.BYTE = ledValue[0];
        count++;
    }
    for(i = 0 ; i < 16 ; i++){
        if((ts_result.button[1] >> i) & 0x01){
            PORTB.PODR.BYTE |= ledValue[i+1];
            count++;
        }
    }
    tone(count);
}

void tone(uint8_t temp){
    uint16_t scaleArr[6] = {523,587,659,699,784,880};
    if(temp != 0){
            result = 13497061/scaleArr[temp-1];
            TMR01.TCORA = result;
            tmrLength = 19+temp*3;
        }
}
```

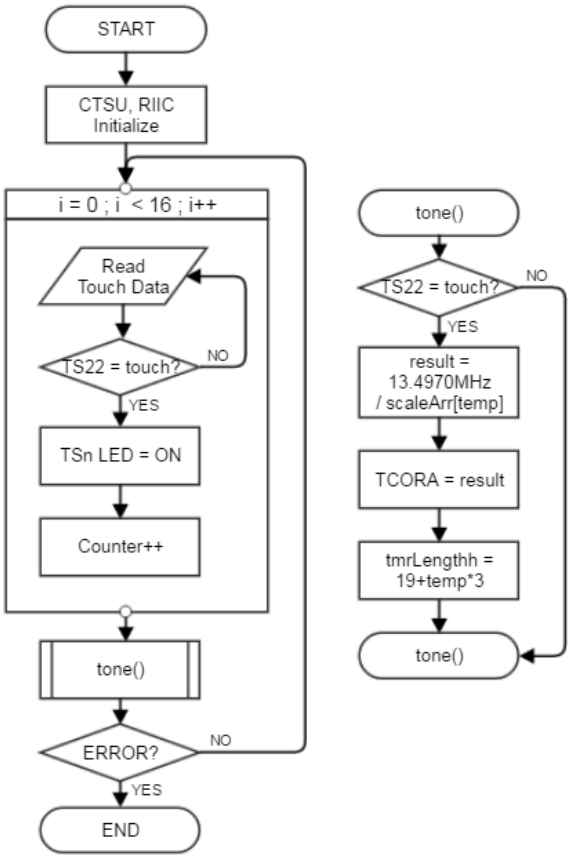

図 3-1 ソースコードとフローチャート

所望の周波数に合わせた TCORA の値の変化に基づいて tmrLength を調整します。タイマーは継続的にカウントしている状態で、tmrLength が 1 以上の場合、タイマ割り込みごとに tmrLength を 1 ずつ減算して音を出している時間を維持します。

ボタンにタッチすると、変数 count が＋1 されます。マルチタッチではタッチした数だけ count が大きくなり，それに対応してド（C）、レ（D）、ミ（E）、ファ（F）、ソ（G）、ラ（A）と、周波数が上がります。

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。

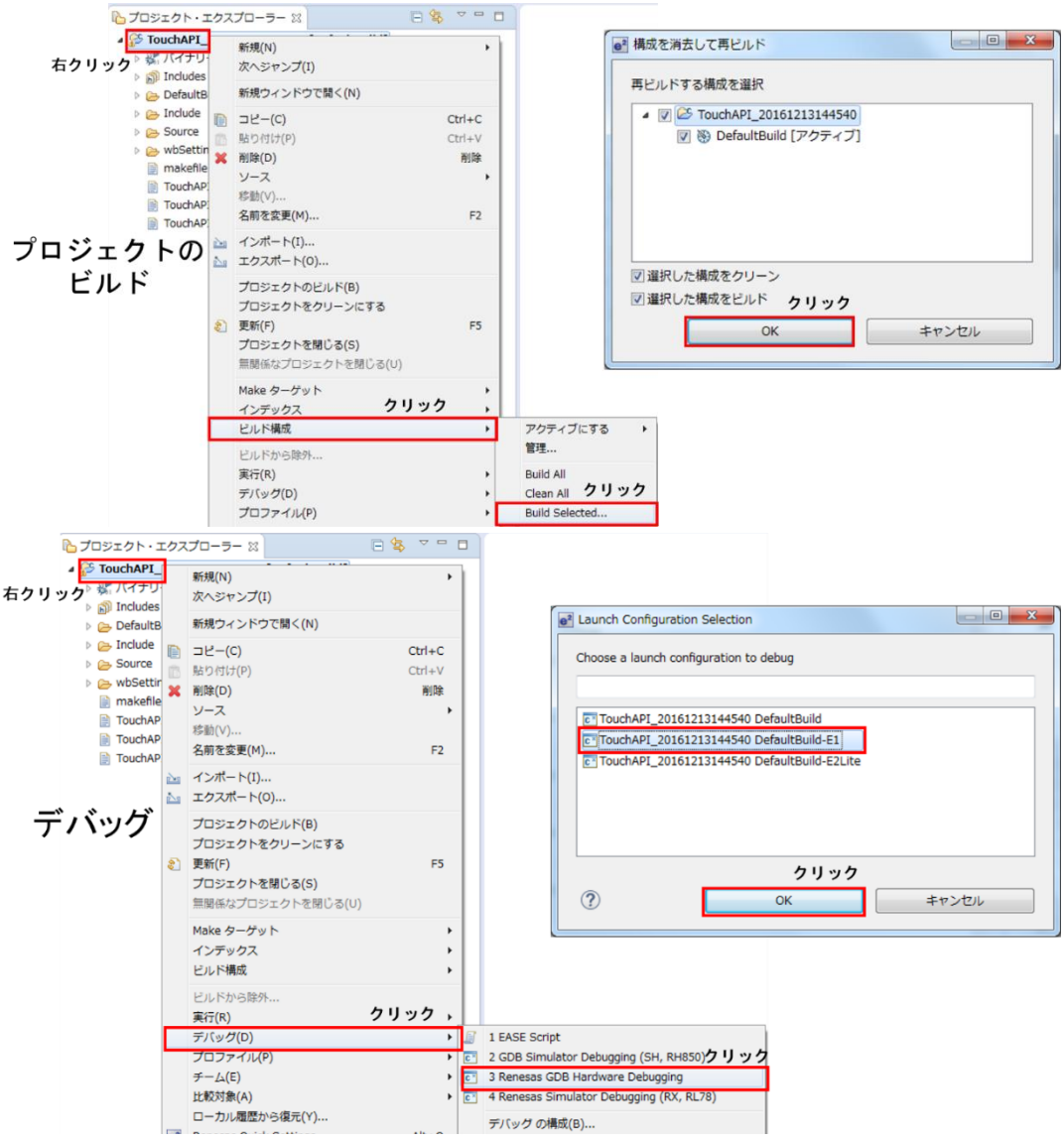

図 4-2 プロジェクトのビルドとデバッグ

5 実行

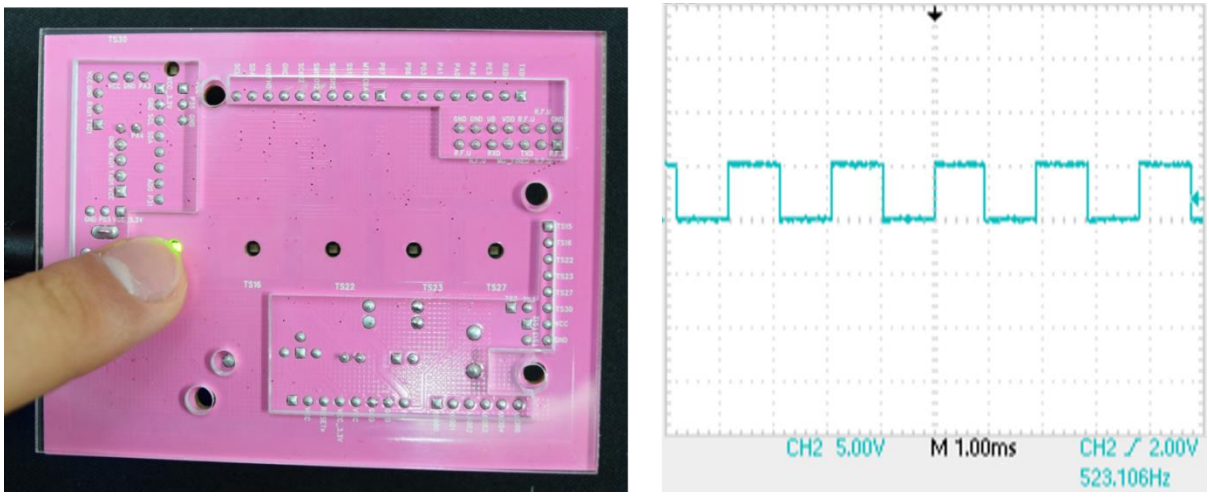

図 5-1 タッチカウントが1のとき → ド(C) 523[Hz]

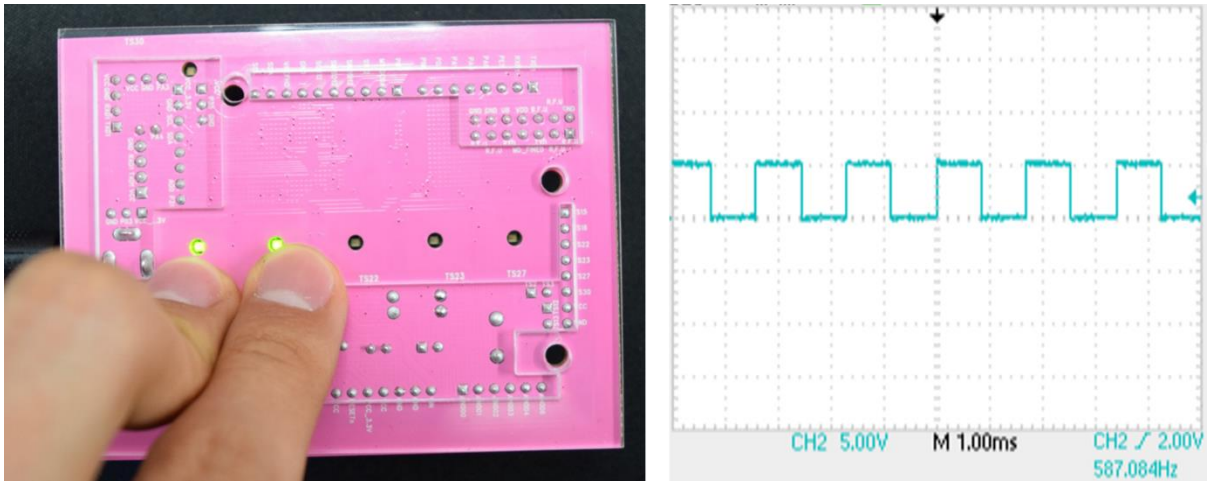

図 5-2 タッチカウントが2のとき → レ(D) 587[Hz]

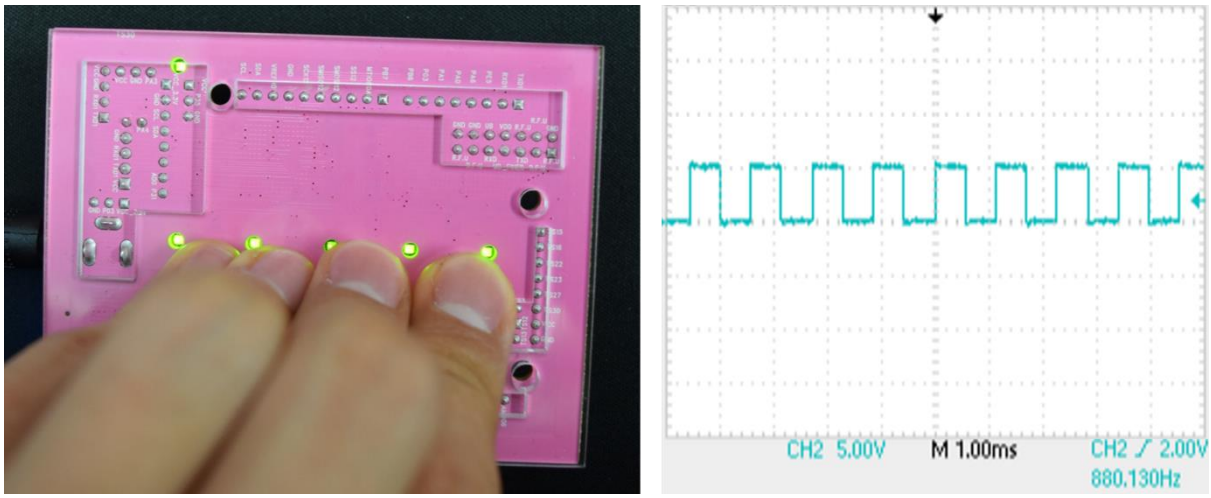

図 5-3 タッチカウントが6のとき → ラ(A) 880[Hz]

6 回路図

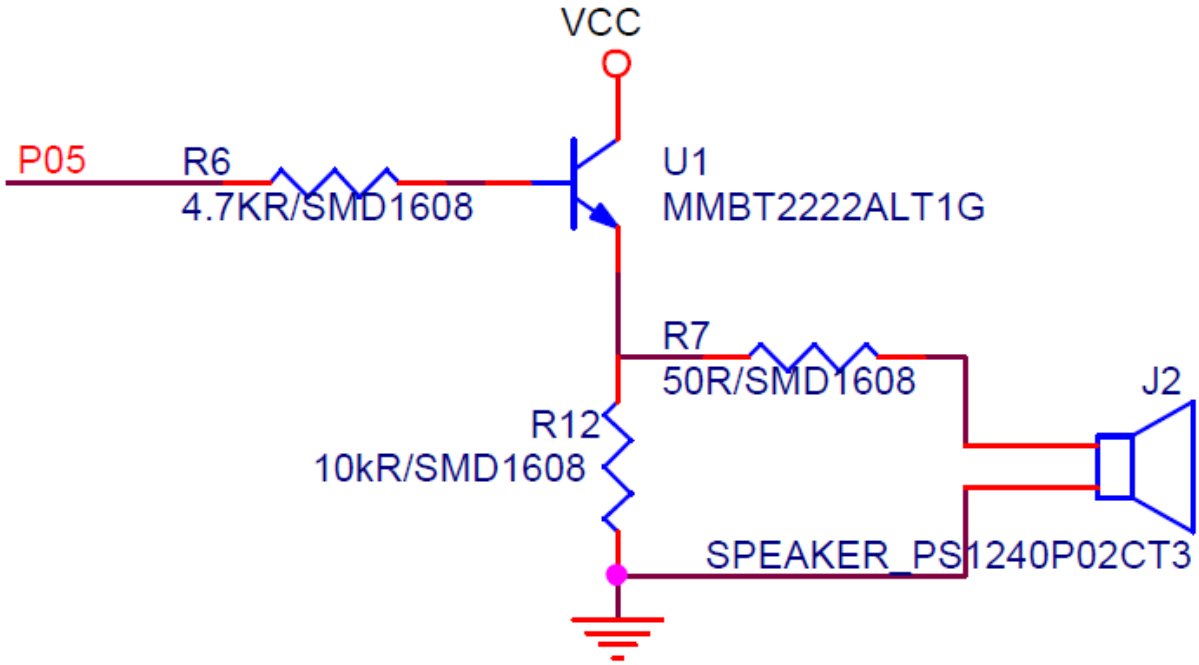

図 6-1 NS-RX231のスピーカー駆動部分の回路図