【注意事項】静電容量タッチ統合開発環境 Workbench6 (R_Set_Cap_Touch_Initial_Tuning 関数の注意)

R20TS0881JJ0100
Rev.1.00
2022.12.16 号

概要

タイトルに記載している製品の使用上の注意事項を連絡します。

1. R_Set_Cap_Touch_Initial_Tuning 関数を使用する場合の注意事項

1. R_Set_Cap_Touch_Initial_Tuning 関数を使用する場合の注意事項

1.1 該当製品

静電容量タッチ統合開発環境 Workbench6 Ver1.04.00.00 以降

1.2 該当デバイス

RX ファミリ：RX113、RX130、RX230、RX231

1.3 内容

“R_Set_Cap_Touch_Initial_Tuning”関数内で呼び出される“initial_offset_tuning”関数の終了条件後の処理に誤りがあり、以下の現象が発生する可能性があります。

(1) CTSUSO0.CTSUSO[9:0]ビットが“1”ずれた値に設定されます。この場合に次のどちらかの現象が発生します。
  (1-a) 特定のタッチボタンの判定がタッチしていない状態で ON となり、そのまま ON を維持する。
  (1-b)しばらくの間タッチボタンが効かない、または感度調整時に設定した感度が変わってしまう。
    (1-b)の現象はドリフト処理が有効であれば時間経過により設定した感度に戻ります。
(2)“R_Set_Cap_Touch_Initial_Tuning”関数の調整完了までの時間が長くなる。

1.4 発生条件

1.3 項の(1-a)は以下の条件がすべて成立することにより発生する可能性があります。

1. 自己容量方式を使用している
2. “R_Set_Cap_Touch_Initial_Tuning”関数で CTSUSO0.CTSUSO[9:0]ビットが“0x010*2n-1”（n=1～31）に調整される基板定数（寄生容量）である。
3. タッチ閾値と CTSUSO0.CTSUSO[9:0]ビットの調整結果が下表に該当する。

タッチ閾値	CTSUSO0.CTSUSO[9:0]ビット調整結果
250～562 以下	0x01F, 0x21F, 0x05F, 0x25F, 0x09F, 0x29F, 0x0DF, 0x2DF, 0x11F, 0x31F, 0x15F, 0x35F, 0x1BF, 0x3BF, 0x1DF, 0x3DF
796 以下	0x03F, 0x0BF, 0x13F, 0x1BF, 0x23F, 0x2BF, 0x33F, 0x3BF
1125 以下	0x07F, 0x17F, 0x27F, 0x37F
1593 以下	0x0FF, 0x2FF
2251 以下	0x1FF

1.5　回避策

“initial_offset_tuning”関数はタッチ検出基準値がターゲット値（15510 から 15260）になるように、計測値を参照して CTSUSO0.CTSUSO[9:0]ビットを 1 ずつ増減してハードウェアによるオフセット調整を行います。10 コールごとにオフセット調整継続判定を実施しており、終了条件の 1 つとして計測値がターゲット値外にある場合に CTSUSO0.CTSUSO[9:0]ビットの増減量積算が±2 以内の範囲で振動収束していればオフセット調整完了と判定します。オフセット調整完了以降の計測は、調整完了時の CTSUSO0.CTSUSO[9:0]ビットに固定し、この設定での計測値がタッチ基準値になります。この終了条件において CTSUSO0.CTSUSO[9:0]ビットに次回の調整値がセットされたまま調整完了判定するため、次回の計測でタッチ基準値と計測値がずれてしまい、問題となる現象が発生する場合があります。回避策として“initial_offset_tuning”関数を修正してください。

修正前の”r_ctsu.c”の“initial_offset_tuning”関数

```
uint8_t initial_offset_tuning( uint8_t method, uint8_t number )
{
  (中略)
  for (loop = 0; loop < number; loop++ )
  {
  (中略)
            if (beas_val < (sensor_raw[loop] - TUNING_UPPER_LIMIT))                    /* Current over check    */
            {
                if (0x03FF != *(g_touch_tuning_info[method].ctsuso  + loop))           /* CTSUSO limit check    */
                {
                   *(g_dtc_write_data[method] + st) =  (*(g_dtc_write_data[method] + st) & 0xFC00) +
(*(g_touch_tuning_info[method].ctsuso + loop) + 1);
                   *(g_touch_tuning_info[method].result + loop)  = 0;
                   *(g_current_sign_pt[method] + loop) = *(g_current_sign_pt[method] + loop) + 1;    /* Plus          */
                }
                else
                {
                   *(g_touch_tuning_info[method].result + loop) = 1;                  /* Tuning finish flag set */
                }
            }
            else if (beas_val > (sensor_raw[loop] + TUNING_LOWER_LIMIT))               /* Current down check    */
            {
                if (0x0000 != *(g_touch_tuning_info[method].ctsuso  + loop))           /* CTSUSO limit check    */
                {
                   *(g_dtc_write_data[method] + st)  = (*(g_dtc_write_data[method] + st) & 0xFC00) +
(*(g_touch_tuning_info[method].ctsuso + loop) - 1);
                   *(g_touch_tuning_info[method].result + loop) = 0;
                   *(g_current_sign_pt[method] + loop) = *(g_current_sign_pt[method] + loop) - 1;
                }
                else
                {
                   *(g_touch_tuning_info[method].result + loop) = 1;
                }
            }
            else
            {
               *(g_touch_tuning_info[method].result + loop) = 1;
            }
        }
        if (10 == g_current_offset_count[method])
        {
            if ((OFFSET_CNT_PLUS >= (*(g_current_sign_pt[method] + loop))) && (OFFSET_CNT_MINUS <=
(*(g_current_sign_pt[method] + loop))))
            {
               *(g_touch_tuning_info[method].result + loop) = 1;
            }
            else
            {
               g_current_offset_count[method] = 0;
               *(g_current_sign_pt[method] + loop) = 0;
            }
        }

        if (0 == g_key_info[method].mode)
        {
            pt      = pt + 1;
        }
        else
        {
            pt      = pt + 3;
        }
        st = st + 3;
    }
    g_ctsu_status[method].flag.data_update = 0;
  (中略)
}    /* End of function initial_offset_tuning() */
```

有効チャネルの数だけループ

CTSUSO0.CTSUSO[9:0]ビット更新

CTSUSO0.CTSUSO[9:0]ビットの増減量積算

CTSUSO0.CTSUSO[9:0]ビットの増減量積算

10 コールごとの継続判定

問題となる終了条件
10 コールごとの継続判定カウンタと CTSUSO0.CTSUSO ビットの増減量積算値がクリアされないため、前記カウンタと積算値がオーバーフローする可能性があります

CTSUSO ビットを更新してから
計測値を確認する前に調整完了判定してしまう

有効チャネル数のループの中で判定中チャネルの結果から 10 コールごとの継続判定カウンタをクリアしてしまうため、次のチャネルは次の 10 コールまで判定されず初期化時間が長くなります

修正後の”r_ctsu.c”の“initial_offset_tuning”関数。赤文字の処理を追加してください。

```
uint8_t initial_offset_tuning( uint8_t method, uint8_t number )
{
 (中略)
    for (loop = 0; loop < number; loop++ )
    {
 (中略)
            if (beas_val < (sensor_raw[loop] - TUNING_UPPER_LIMIT))                          /* Current over check    */
            {
                if (0x03FF != *(g_touch_tuning_info[method].ctsuso  + loop))                 /* CTSUSO limit check    */
                {
                   *(g_dtc_write_data[method] + st) =  (*(g_dtc_write_data[method] + st) & 0xFC00) +
(*(g_touch_tuning_info[method].ctsuso + loop) + 1);
                   *(g_touch_tuning_info[method].result + loop)  = 0;
                   *(g_current_sign_pt[method] + loop) = *(g_current_sign_pt[method] + loop) + 1;    /* Plus          */
                }
                else
                {
                   *(g_touch_tuning_info[method].result + loop) = 1;                        /* Tuning finish flag set */
                }
            }
            else if (beas_val > (sensor_raw[loop] + TUNING_LOWER_LIMIT))                     /* Current down check    */
            {
                if (0x0000 != *(g_touch_tuning_info[method].ctsuso  + loop))                 /* CTSUSO limit check    */
                {
                   *(g_dtc_write_data[method] + st)  = (*(g_dtc_write_data[method] + st) & 0xFC00) +
(*(g_touch_tuning_info[method].ctsuso + loop) - 1);
                   *(g_touch_tuning_info[method].result + loop) = 0;
                   *(g_current_sign_pt[method] + loop) = *(g_current_sign_pt[method] + loop) - 1;
                }
                else
                {
                   *(g_touch_tuning_info[method].result + loop) = 1;
                }
            }
            else
            {
               *(g_touch_tuning_info[method].result + loop) = 1;
            }
        }

        if (10 == g_current_offset_count[method])
        {
            if ((OFFSET_CNT_PLUS >= (*(g_current_sign_pt[method] + loop))) && (OFFSET_CNT_MINUS <=
(*(g_current_sign_pt[method] + loop))))
            {
                *(g_touch_tuning_info[method].result + loop) = 1;
                *(g_dtc_write_data[method] + st)  = (*(g_dtc_write_data[method] + st) & 0xFC00) +
*(g_touch_tuning_info[method].ctsuso + loop);
            }
            else
            {
              // g_current_offset_count[method] = 0;
               *(g_current_sign_pt[method] + loop) = 0;
            }
        }

        if (0 == g_key_info[method].mode)
        {
            pt     = pt + 1;
        }
        else
        {
            pt     = pt + 3;
        }
        st = st + 3;
    }

    if (10 == g_current_offset_count[method])
    {
        g_current_offset_count[method] = 0;
    }

    g_ctsu_status[method].flag.data_update = 0;
 (中略)
}    /* End of function initial_offset_tuning() */
```

この処理は削除してください

1.6　恒久対策

2018 年 11 月より静電容量タッチ開発支援ツールは Workbench6 と QE for Capacitive Touch が平行して公開していましたが、今後 Workbench6 のバージョンアップはせず、QE for Capacitive Touch のみの公開となります。

Workbench6 のバージョンアップによる改修は行いません。Workbench6 を継続して使用する場合は回避策を適用してください。

新規開発には QE for Capacitive Touch をご使用ください。QE for Capacitive Touch では本現象は発生しません。

以上

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	Dec.16.22	-	新規発行

本資料に記載されている情報は、正確を期すため慎重に作成したものですが、誤りがないことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。

過去のニュース内容は発行当時の情報をもとにしており、現時点では変更された情報や無効な情報が含まれている場合があります。

ニュース本文中の URL を予告なしに変更または中止することがありますので、あらかじめご承知ください。

本社所在地

〒135-0061 東京都江東区豊洲 3-2-24（豊洲フォレシア）

www.renesas.com

お問合せ窓口

弊社の製品や技術、ドキュメントの最新情報、最寄の営業お問合せ窓口に関する情報などは、弊社ウェブサイトをご覧ください。

www.renesas.com/contact/

商標について

ルネサスおよびルネサスロゴはルネサス エレクトロニクス株式会社の商標です。すべての商標および登録商標は、それぞれの所有者に帰属します。

TS Colophon 4.3