

お客様各位

カタログ等資料中の旧社名の扱いについて

2010年4月1日を以ってNECエレクトロニクス株式会社及び株式会社ルネサステクノロジが合併し、両社の全ての事業が当社に承継されております。従いまして、本資料中には旧社名での表記が残っておりますが、当社の資料として有効ですので、ご理解の程宜しくお願ひ申し上げます。

ルネサスエレクトロニクス ホームページ (<http://www.renesas.com>)

2010年4月1日

ルネサスエレクトロニクス株式会社

【発行】ルネサスエレクトロニクス株式会社 (<http://www.renesas.com>)

【問い合わせ先】 <http://japan.renesas.com/inquiry>

ご注意書き

1. 本資料に記載されている内容は本資料発行時点のものであり、予告なく変更することがあります。当社製品のご購入およびご使用にあたりましては、事前に当社営業窓口で最新の情報をご確認いただきますとともに、当社ホームページなどを通じて公開される情報に常にご注意ください。
2. 本資料に記載された当社製品および技術情報の使用に関連し発生した第三者の特許権、著作権その他の知的財産権の侵害等に関し、当社は、一切その責任を負いません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
3. 当社製品を改造、改変、複製等しないでください。
4. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器の設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因しお客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
5. 輸出に際しては、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。本資料に記載されている当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事用途の目的で使用しないでください。また、当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器に使用することができません。
6. 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のですが、誤りが無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。
7. 当社は、当社製品の品質水準を「標準水準」、「高品質水準」および「特定水準」に分類しております。また、各品質水準は、以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認ください。お客様は、当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途に当社製品を使用することができません。また、お客様は、当社の文書による事前の承諾を得ることなく、意図されていない用途に当社製品を使用することができません。当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途または意図されていない用途に当社製品を使用したことによりお客様または第三者に生じた損害等に関し、当社は、一切その責任を負いません。なお、当社製品のデータ・シート、データ・ブック等の資料で特に品質水準の表示がない場合は、標準水準製品であることを表します。
標準水準： コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、産業用ロボット
高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、防災・防犯装置、各種安全装置、生命維持を目的として設計されていない医療機器（厚生労働省定義の管理医療機器に相当）
特定水準： 航空機器、航空宇宙機器、海底中継機器、原子力制御システム、生命維持のための医療機器（生命維持装置、人体に埋め込み使用するもの、治療行為（患部切り出し等）を行うもの、その他直接人命に影響を与えるもの）（厚生労働省定義の高度管理医療機器に相当）またはシステム等
8. 本資料に記載された当社製品のご使用につき、特に、最大定格、動作電源電圧範囲、放熱特性、実装条件その他諸条件につきましては、当社保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
9. 当社は、当社製品の品質および信頼性の向上に努めておりますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害などを生じさせないようお客様の責任において冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、機器またはシステムとしての出荷保証をお願いいたします。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様が製造された最終の機器・システムとしての安全検証をお願いいたします。
10. 当社製品の環境適合性等、詳細につきましては製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関し、当社は、一切その責任を負いません。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを固くお断りいたします。
12. 本資料に関する詳細についてのお問い合わせその他お気付きの点等がございましたら当社営業窓口までご照会ください。

注 1. 本資料において使用されている「当社」とは、ルネサスエレクトロニクス株式会社およびルネサスエレクトロニクス株式会社とその総株主の議決権の過半数を直接または間接に保有する会社をいいます。

注 2. 本資料において使用されている「当社製品」とは、注 1 において定義された当社の開発、製造製品をいいます。

M16C/26 グループ

キー入力割り込みの使い方

1. 要約

この資料では M16C/26 グループのキー入力割り込みの使用方法について説明します。

2. はじめに

キー入力割り込みは、M16C/26 グループのウェイトモード、またはストップモードの解除に使用できます。また、ポーリングによる入力端子のチェックを割り込みに置き換えるなど、入力端子の監視にも使用できます。M16C/26 グループではキー入力割り込みに加え、5本の端子を外部要因割り込み端子として使用できます。

3.0 キー入力割り込み端子

M16C/26 グループではポート 10 の上位 4 ビット (P10_7, P10_6, P10_5, P10_4) がキー入力割り込み端子となります。これらの 4 本の端子は、プログラマブル入出力ポートと A-D 変換器のアナログ入力端子としての機能も備えています。

3.1 キー入力割り込みの設定

ポート P10 の上位 4 ビットを使用目的に合わせてレジスタに設定する必要があります。キー入力割り込み端子として使用する場合、以下の設定が必要です。

- ポート P10 方向レジスタに、キー入力割り込み端子として使用する端子を入力に設定する。
- キー入力割り込み（およびすべての割り込み）を有効にする。

キー入力割り込みを有効にするには、割り込み優先レベル (1~7) をキー入力割り込み制御レジスタ (kupic) に設定する必要があります。また全ての割り込みを許可するために FSET I (I フラグを“1”に設定する) 命令を実行します。リセット後は、割り込みは禁止になっていることに注意してください。

3.2 キー入力割り込み使用時の注意事項

キー入力割り込みの設定と使用方法は簡単ですが、使用する際には以下の注意が必要です。

キー入力割り込みは、入力端子の立ち下りエッジ (“H” 状態から “L” 状態へ遷移する) によって発生します。キー入力端子のいずれかに “L” が入力されている場合、他の端子の入力によるキー入力割り込みは発生しません。

他の割り込みと同様、キー入力割り込み発生時にマイクロコンピュータが対応する割り込み処理を行うため、ベクタアドレスを設定する必要があります。割り込みベクタは通常のプログラム開発では、sect30.inc ファイルに設定します。

4. キー入力割り込み参考プログラムについて

参考プログラムは MSV30262 (評価用基板) で動作します。スイッチ S2 が押される等すると、キー入力割り込みが発生し、LED が点灯します。スイッチを離すと LED は消灯します。

5. 参考プログラムコード

以下にキー入力割り込み参考プログラムコードを示します。

```

/*****
 * Mini 26 Rev. B Key Input Irq Program *
 * main.c *
 * v1.0 03/06/2003 *
 *****/

/*****
 * This Key Input Irq Program shows how to setup key input interrupts *
 * on the M16C/26. The program was written for the Mini 26 board. *
 * *
 * The program is simple: blink LEDs every time an interrupt occurs. *
 *****/

/* Include the required header files */
#include "..\¥common¥sfr262.h" // M16C/26 special function register definitions

/* Interrupt routines used for this demo must be defined with #pragma INTERRUPT as
shown below. Vectors must also be set in sect30_keyinirq.inc startup file to
point to the interrupt routine. */

#pragma INTERRUPT ki_irq

/* Function prototypes */
void ki_irq(void);

/* General definitions */
#define ON 0
#define OFF 1

```

```

/*****
Name:          Main - main program loop
Parameters:    None
Returns:       None
Description:    Handles processing and contains the infinite loop while waiting for
                key input interrupt.
*****/
main(){

    /* To use key input interrupts, the port direction where the key inputs
       are must be configured as inputs. The M16C/26 has 4 key inputs which can
       be found on the upper 4 bits of port 10 (P10_7, P10_6, P10_5, P10_4).
       We will set the direction for these 4 pins as inputs by setting it to 0.

       On the Mini 26 Board, only P10_7 is used. P10_7 is connected to
       pushbutton S2. The other 3 pins (P10_6, P10_5, & P10_4) are unused.

       On the MSV30262 SKP Board, P10_7, P10_6, & P10_5 are
       connected to S4, S3, & S2 respectively. P10_4 is not used.

       The lower 4 bits are don't cares and can be set to inputs or outputs
       depending on your hardware connection to these 4 pins. In our example,
       P10_0 & P10_1 are set as inputs because on the Mini 26 and MSV30262
       SKP Boards, these are used as ADC inputs. Ports P10_2 & P10_3 are unused
       and we set these as outputs. */

    pd10 = 0x0C;
    /* 00001100;          Ports 10_7, 10_6, 10_5, 10_4 as inputs, where the
       | | | | | | | |      key inputs are. Ports 10_3, 10_2, 10_1, 10_0 are
       | | | | | | | |      don't cares. In our example, we set these to outputs.
       | | | | | | | |
       | | | | | | | |_____P10_0, don't care - set to 0 (input)
       | | | | | | | |_____P10_1, don't care - set to 0 (input)
       | | | | | | | |_____P10_2, don't care - set to 1 (output)
       | | | | | | | |_____P10_3, don't care - set to 1 (output)
       | | | | | | | |_____P10_4 (key input 0), set to 0 (input)
       | | | | | | | |_____P10_5 (key input 1), set to 0 (input)
       | | | | | | | |_____P10_6 (key input 2), set to 0 (input)
       | | | | | | | |_____P10_7 (key input 3), set to 0 (input) */

    /* The next program step is not necessary as long as the pins are held at
       a high level when not generating a key input irq.

       On the MSV30262-SKP, the 3 inputs are held high with external pull-up
       resistors and so, the step can be omitted.

       On the Mini 26 Board, there is no external pull-up for pushbutton S2, and
       so enable the internal pull-up for the upper bits of port 10. Again this
       is only for the Mini 26 Board.          */

    pu25 = 1;    // enable internal pull-up for upper 4 bits of P10

    /* We need to enable the key input interrupt. To accomplish this, we set

```

the key input interrupt control register to a non-zero value ranging from 1 to 7. The value you set controls the software interrupt priority level - 1 (lowest) and 7 (highest). */

```

/* Initialize LED ports so we can use it for this demo */
pd7 |= 0x07;      // initialize LED's ports to outputs
p7  |= 0x07;      // turn off LED's

asm("FCLR I");    // disable interrupts before changing irq registers
kupic = 2;        // enable key input irq by setting to non-zero value,
                  // 2 (interrupt priority level 2)
asm("FSET I");    // enable interrupts

while(1){        // infinite loop
    p7 |= 0x07;   // turn off LED's while waiting for key input irq
}
}

/*****
Name:      ki_irq
Parameters: None
Returns: None
Description: This is an key input interrupt routine - processing when a key
             interrupt is generated. A key input interrupt is generated when
             one of the 4 pins goes to a low level.

NOTE: If one of the 4 inputs is always low, a key input interrupt
      is NOT generated. It is generated by the falling edge (high->low)
      of one of the key inputs.

For this demo on the Mini 26 Board, the key input interrupt is
generated when pushbutton S2 is pressed.

The routine turns Red and Green LEDs.
*****/
void ki_irq(void){
    p7 = 0x02;      // turn on Red and Green LEDs
    while(p10_7==0); // wait here until pushbutton is released
}

```

6. 参考ドキュメント

ハードウェアマニュアル

- M16C/26 グループ ハードウェアマニュアル

7. ホームページとサポート窓口

ルネサス テクノロジホームページ
<http://www.renesas.com/>

M16C ファミリ MCU 技術サポート窓口
E-mail: support_apl@renesas.com

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2004.02.31	-	初版発行

安全設計に関するお願い

1. 弊社は品質、信頼性の向上に努めておりますが、半導体製品は故障が発生したり、誤動作する場合があります。弊社の半導体製品の故障又は誤動作によって結果として、人身事故、火災事故、社会的損害などを生じさせないような安全性を考慮した冗長設計、延焼対策設計、誤動作防止設計などの安全設計に十分ご注意ください。

本資料ご利用に際しての留意事項

1. 本資料は、お客様が用途に応じた適切なルネサス テクノロジ製品をご購入いただくための参考資料であり、本資料中に記載の技術情報についてルネサス テクノロジが所有する知的財産権その他の権利の実施、使用を許諾するものではありません。
2. 本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他応用回路例の使用に起因する損害、第三者所有の権利に対する侵害に関し、ルネサス テクノロジは責任を負いません。
3. 本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他全ての情報は本資料発行時点のものであり、ルネサス テクノロジは、予告なしに、本資料に記載した製品または仕様を変更することがあります。ルネサス テクノロジ半導体製品のご購入に当たりますは、事前にルネサス テクノロジ、ルネサス販売または特約店へ最新の情報をご確認頂きますとともに、ルネサス テクノロジホームページ(<http://www.renesas.com>)などを通じて公開される情報に常にご注意ください。
4. 本資料に記載した情報は、正確を期すため、慎重に制作したものです。万一本資料の記述誤りに起因する損害がお客様に生じた場合には、ルネサス テクノロジはその責任を負いません。
5. 本資料に記載の製品データ、図、表に示す技術的な内容、プログラム及びアルゴリズムを流用する場合は、技術内容、プログラム、アルゴリズム単位で評価するだけでなく、システム全体で十分に評価し、お客様の責任において適用可否を判断してください。ルネサス テクノロジは、適用可否に対する責任を負いません。
6. 本資料に記載された製品は、人命にかかわるような状況の下で使用される機器あるいはシステムに用いられることを目的として設計、製造されたものではありません。本資料に記載の製品を運輸、移動体用、医療用、航空宇宙用、原子力制御用、海底中継用機器あるいはシステムなど、特殊用途へのご利用をご検討の際には、ルネサス テクノロジ、ルネサス販売または特約店へご照会ください。
7. 本資料の転載、複製については、文書によるルネサス テクノロジの事前の承諾が必要です。
8. 本資料に関し詳細についてのお問い合わせ、その他お気づきの点がございましたらルネサス テクノロジ、ルネサス販売または特約店までご照会ください。