

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/62 グループ

A-D 変換器 オペアンプゲイン調整接続例の説明

1. 要約

この資料では、オペアンプを使用してゲイン調整を行った A-D 変換方法の紹介と使用例を掲載しています。

2. はじめに

この資料で説明する応用例は次のマイコンに適用されます。

- ・マイコン : M16C/62 グループ

M16C/62 グループと同様の SFR(周辺装置制御レジスタ)を持つ他の M16C ファミリでも本プログラムを使用することができます。ただし、一部の機能を機能追加等で変更している場合がありますのでマニュアルで確認してください。

このアプリケーションノートをご使用に際しては十分な評価を行ってください。

3. 応用例の説明

M16C/62 グループで、外部オペアンプを使用してゲイン調整を行った場合の A-D 変換例を説明します。

3.1 接続例

外部オペアンプを使用してゲイン調整を行う場合の接続例を以下に示します。

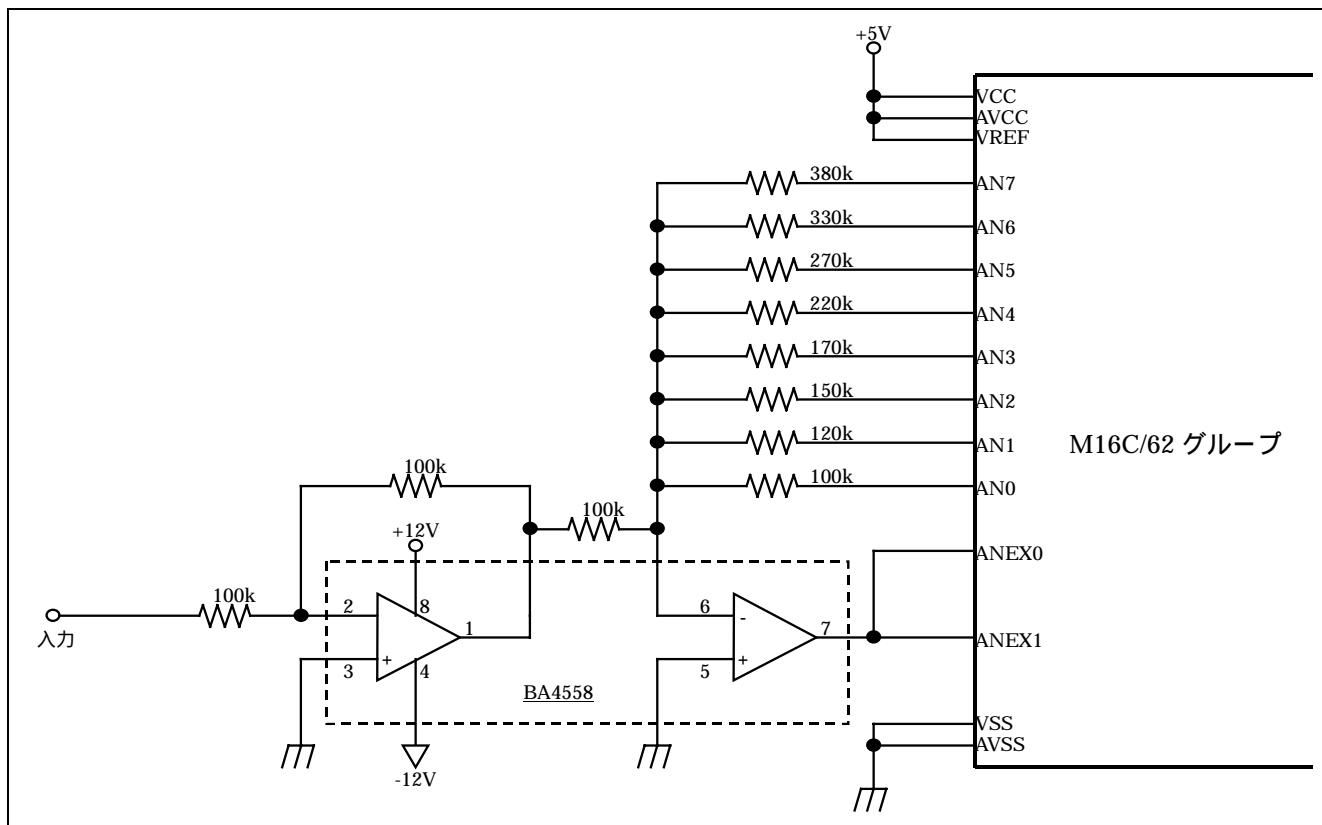


図 1. 接続例

注意事項

- (1) 上記の接続例では、反転増幅回路を 2 回路使用して、正相に戻しています。
非反転増幅回路を使用する場合は、上記の接続例とは異なります。
- (2) 上記の接続例では、AN0 ~ AN7 へ入力される値は抵抗値の比率により、下記の通り増幅されます。

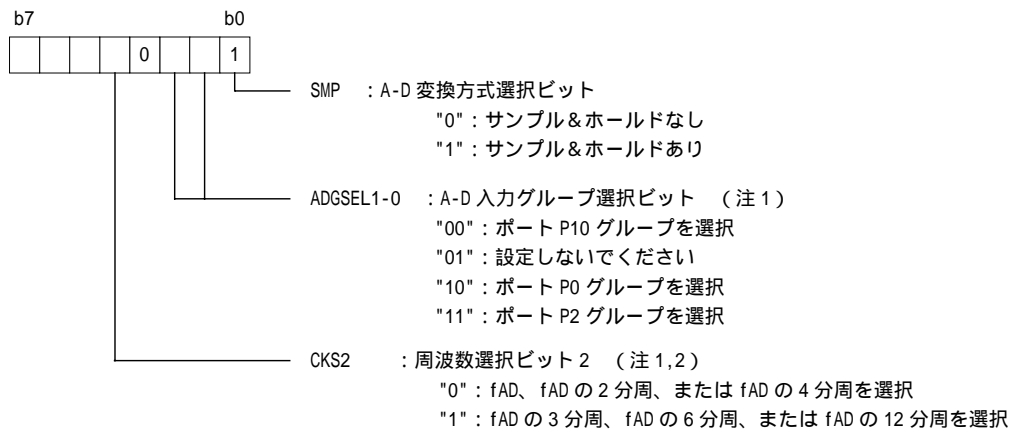
AN0 : 100k/100k=1	(1 倍増幅)
AN1 : 120k/100k=1.2	(1.2 倍増幅)
AN2 : 150k/100k=1.5	(1.5 倍増幅)
AN3 : 170k/100k=1.7	(1.7 倍増幅)
AN4 : 220k/100k=2.2	(2.2 倍増幅)
AN5 : 270k/100k=2.7	(2.7 倍増幅)
AN6 : 330k/100k=3.3	(3.3 倍増幅)
AN7 : 380k/100k=3.8	(3.8 倍増幅)

3.2 設定方法

外部オペアンプを使用してゲイン調整を行った結果を A-D 変換する場合の設定方法を以下に示します。
ここでは、単発モード、10 ビットモード、サンプル&ホールドありで A-D 変換を行う例を示します。

(1) ADCON2 レジスタ (A-D 制御レジスタ 2) を設定する。

- A-D 変換方式、周波数選択ビットを設定する。



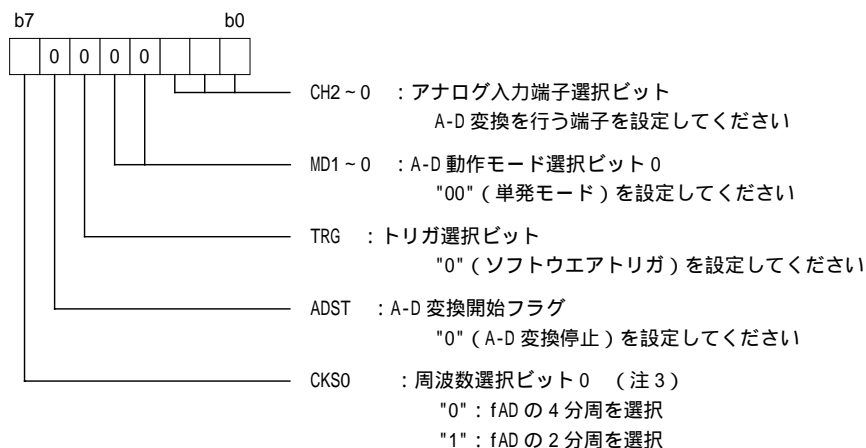
注1. M16C/62P グループでのみ有効です。その他のグループでは"00"にしてください。

注2. M16C/62P グループでは下記の組み合わせで AD の周波数を選択してください。

CKS2	CKS1	CKS0	AD
0	0	0	fAD の 4 分周
0	0	1	fAD の 2 分周
0	1	0	fAD
0	1	1	
1	0	0	fAD の 12 分周
1	0	1	fAD の 6 分周
1	1	0	fAD の 3 分周
1	1	1	

(2) ADCON0 レジスタ (A-D 制御レジスタ 0) を設定する。

- アナログ入力端子、A-D 動作モード、周波数選択ビットなどを設定する。



注3. M16C/62P グループの設定例は、「3.2(1)注2」を参照してください。

(7) A-D 変換完了待ち

ADIC レジスタの IR ビットが"1" (割り込み要求あり) になるまで待つ。

(8) A-D 変換結果の読み出し

アナログ入力端子選択ビットで選択した端子に該当する AD レジスタ i ($i=0\sim 7$) より変換結果を読み出す。

4. 参考ドキュメント

ハードウェアマニュアル

M16C/62 グループ (M16C/62P) ハードウェアマニュアル Rev.1.11

M16C/62 グループデータシート Rev.H2

M16C/62A グループデータシート Rev.C1

M16C/62N グループデータシート Rev.1.1

(最新版をルネサス テクノロジホームページから入手してください。)

ユーザーズマニュアル

M16C/62 グループユーザーズマニュアル Rev.C3

M16C/62A グループユーザーズマニュアル Rev.1.0

(最新版をルネサス テクノロジホームページから入手してください。)

5. 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/jpn/>

M16C ファミリ MCU 技術サポート窓口

E-mail: support_apl@renesas.com

6. 参考プログラム例

AN0 ~ AN7 から外部オペアンプを接続してゲイン増幅をした値を順次読み出していくプログラム例を以下に示します。

```

;*****
;
;
; M16C/62 Group Program Collection
;
; FILE NAME : rjj05b0441_src.a30
; CPU : M16C/62 Group
; FUNCTION : The example of A-D conversion at the time of external
;           operational amplifier gain adjustment
; HISTORY : 2004.01.15 Ver 1.00
;
; Copyright (C) 2004. Renesas Technology Corp.
; Copyright (C) 2004. Renesas Solutions Corp.
; All right reserved.
;*****
;
; ----- include define -----
; .list off
; .include sfr62p.inc
; .list on
;
; ----- Symbol define -----
vstack .equ 0002b00h ; Stack Pointer
vram .equ 0000400h ; Internal RAM area
vram_end .equ 0002c00h ;
vpro .equ 00fc000h ; Program Start address
vval_vec .equ 00ffd00h ; Variable vector address
vvector .equ 00ffdch ; Non-maskable vector address
;
; ----- Internal RAM Area -----
; .section ramdata,data
; .org vram
;
; ----- Program Area -----
; .section program,code
; .org vpro
reset:
;
; ----- Initial setting -----
; ldc #vstack,sp ; Set stack-pointer address
; ldintb #vval_vec ; Set variable vector table address
;
; mov.b #003h, prcr
; mov.b #008h, cm0
; mov.b #020h, cm1 ; main-clock divid by 0 mode
; mov.b #000h, prcr
;
; mov.w #00000h, p0
; mov.w #0ffffh, pd0 ; Port0/1 output select
; mov.w #00000h, p2
; mov.w #0ffffh, pd2 ; Port2/3 output select
; mov.w #00000h, p4
; mov.w #0ffffh, pd4 ; Port4/5 output select
; mov.w #00000h, p6
; mov.w #0ffffh, pd6 ; Port6/7 output select
;
; bset prc2
; mov.b #000h, pd9 ; P9_5(ANEX0) & P9_6(ANEXq) is input port
; mov.b #000h, pd10 ; P10_0(AN0) to P10_7(AN7) is input port
;
;

```

```

;----- Evaluation start -----
start:
  mov.b  #0000001b, adcon2
;          | ||+----- conversion mode select   : sample&hold
;          | ++----- input group select       : select P10 group
;          +----- Freq select bit2           : fAD/2

  mov.b  #1000000b, adcon0
;          ||||+----- input select            : AN0 select
;          |||++----- mode select bit0        : single mode
;          ||+----- trigger select           : software
;          |+----- AD start flag             : stop
;          +----- Freq select bit0           : fAD/2

  mov.b  #1110000b, adcon1
;          |||||+----- sweep pin select       : none
;          ||||+----- mode select bit1       : 0 fix
;          |||+----- 8/10 bit select         : 8bit
;          ||+----- Freq select bit1        : fAD/2
;          |+----- VREF connect bit        : VREF connect
;          ++----- Ext ope-amp connect mode : Ext ope-amp connect

  mov.b  #007h, adic          ; set AD interrupt priority level

main_loop:
;
; ----- AN0 conversion -----

  bclr   adst                ; AD conversion stop
  mov.b  #1000000b, adcon0

  jsr    AD_wait

  bset   adst                ; AD conversion start
an0_wait:
;          ; wait for AN0 conversion complete
  btst   ir_adic
  jnc    an0_wait
  bclr   ir_adic            ; AD interrupt req clear

  mov.b  ad0, p0            ; conversion result display

;
; ----- AN1 conversion -----

  bclr   adst                ; AD conversion stop
  mov.b  #1000001b, adcon0
;          +++----- input select            : AN1 select

  jsr    AD_wait

  bset   adst                ; AD conversion start
an1_wait:
;          ; wait for AN1 conversion complete
  btst   ir_adic
  jnc    an1_wait
  bclr   ir_adic            ; AD interrupt req clear

  mov.b  ad1, p1            ; conversion result display

;
; ----- AN2 conversion -----

  bclr   adst                ; AD conversion stop
  mov.b  #10000010b, adcon0
;          +++----- input select            : AN2 select

  jsr    AD_wait

  bset   adst                ; AD conversion start

```

```

an2_wait:                                     ; wait for AN2 conversion complete
    btst    ir_adic
    jnc     an2_wait
    bclr    ir_adic                           ; AD interrupt req clear

    mov.b   ad2, p2                           ; conversion result display

;
; ----- AN3 conversion -----

    bclr    adst                               ; AD conversion stop
    mov.b   #10000011b, adcon0
;         +----- input select      : AN3 select

    jsr     AD_wait

    bset    adst                               ; AD conversion start
an3_wait:                                     ; wait for AN3 conversion complete
    btst    ir_adic
    jnc     an3_wait
    bclr    ir_adic                           ; AD interrupt req clear

    mov.b   ad3, p3                           ; conversion result display

;
; ----- AN4 conversion -----

    bclr    adst                               ; AD conversion stop
    mov.b   #10000100b, adcon0
;         +----- input select      : AN4 select

    jsr     AD_wait

    bset    adst                               ; AD conversion start
an4_wait:                                     ; wait for AN4 conversion complete
    btst    ir_adic
    jnc     an4_wait
    bclr    ir_adic                           ; AD interrupt req clear

    mov.b   ad4, p4                           ; conversion result display

;
; ----- AN5 conversion -----

    bclr    adst                               ; AD conversion stop
    mov.b   #10000101b, adcon0
;         +----- input select      : AN5 select

    jsr     AD_wait

    bset    adst                               ; AD conversion start
an5_wait:                                     ; wait for AN5 conversion complete
    btst    ir_adic
    jnc     an5_wait
    bclr    ir_adic                           ; AD interrupt req clear

    mov.b   ad5, p5                           ; conversion result display

;
; ----- AN6 conversion -----

    bclr    adst                               ; AD conversion stop
    mov.b   #10000110b, adcon0
;         +----- input select      : AN6 select

    jsr     AD_wait

    bset    adst                               ; AD conversion start

```

```

an6_wait:                                     ; wait for AN6 conversion complete
    btst    ir_adic
    jnc     an6_wait
    bclr    ir_adic                           ; AD interrupt req clear

    mov.b   ad6, p6                           ; conversion result display

;
; ----- AN7 conversion -----
;
    bclr    adst                               ; AD conversion stop
    mov.b   #10000111b, adcon0
;
;         +++----- input select      : AN7 select
;
    jsr     AD_wait

    bset    adst                               ; AD conversion start
an7_wait:                                     ; wait for AN7 conversion complete
    btst    ir_adic
    jnc     an7_wait
    bclr    ir_adic                           ; AD interrupt req clear

    mov.b   ad7, p7                           ; conversion result display

; ----- AD conversion completed -----
    mov.b   #0ffh, pd8
    mov.b   #0ffh, p8
end_loop:
    jmp     end_loop                           ; Infinity loop

;
; ----- Opeamp wakeup wait routine -----
AD_wait:
    mov.b   #000b, ta0mr                       ; for Opeamp wakeup wait
    mov.w   #1600-1, ta0                       ; 100us(16MHz,f1)
    bset    ta0s                                ; TA0 start
ta0_wait:
    btst    ir_ta0ic                           ; TA0 overflow wait
    jnc     ta0_wait
    bclr    ir_ta0ic

    rts

;
;----- Evaluation end -----
;
;//////////////////////////////////////
; interrupt routine
;//////////////////////////////////////
dummyi:                                     ; Dummy interrupt routine
;
    nop
    nop
    nop
    nop
    reit

;
;//////////////////////////////////////
; Non-maskable interrupt routine
;//////////////////////////////////////
undi:                                       ; Undefined instruction interrupt
ovfli:                                       ; Overflow interrupt
brki:                                       ; BRK instruction interrupt
addr:                                       ; Address match interrupt
wdti:                                       ; Watch-dog timer interrupt
nmii:                                       ; NMI interrupt
;
    nop
    nop

```

```

nop
nop
reit

;
;////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
; Variabile vector table
;////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
.section    val_vector,romdata
.org       vval_vec
;
.lword     dummyi           ; 0=BRK instruction interrupt
.lword     dummyi           ; 1=
.lword     dummyi           ; 2=
.lword     dummyi           ; 3=
.lword     dummyi           ; 4=^INT3 interrupt
.lword     dummyi           ; 5=TB5 interrupt
.lword     dummyi           ; 6=TB4/Uart1 bus collision interrupt
.lword     dummyi           ; 7=TB3/Uart0 bus collision interrupt
.lword     dummyi           ; 8=SI04/^INT5 interrupt
.lword     dummyi           ; 9=SI03/^INT4 interrupt
.lword     dummyi           ;10=Uart2 bus collision interrupt
.lword     dummyi           ;11=DMA0 interrupt
.lword     dummyi           ;12=DMA1 interrupt
.lword     dummyi           ;13=KEY input interrutt
.lword     dummyi           ;14=AD interrupt
.lword     dummyi           ;15=Uart2 transmit/NACK2 interrupt
.lword     dummyi           ;16=Uart2 receive/ACK2 interrupt
.lword     dummyi           ;17=Uart0 transmit/NACK0 interrupt
.lword     dummyi           ;18=Uart0 receive/ACK0 interrupt
.lword     dummyi           ;19=Uart1 transmit/NACK1 interrupt
.lword     dummyi           ;20=Uart1 receive/ACK1 interrupt
.lword     dummyi           ;21=TA0 interrupt
.lword     dummyi           ;22=TA1 interrupt
.lword     dummyi           ;23=TA2 interrupt
.lword     dummyi           ;24=TA3 interrupt
.lword     dummyi           ;25=TA4 interrupt
.lword     dummyi           ;26=TB0 interrupt
.lword     dummyi           ;27=TB1 interrupt
.lword     dummyi           ;28=TB2 interrupt
.lword     dummyi           ;29=^INT0 interrupt
.lword     dummyi           ;30=^INT1 interrupt
.lword     dummyi           ;31=^INT2 interrupt
.lword     dummyi           ;32=
.lword     dummyi           ;33=
.lword     dummyi           ;34=
.lword     dummyi           ;35=
.lword     dummyi           ;36=
.lword     dummyi           ;37=
.lword     dummyi           ;38=
.lword     dummyi           ;39=
.lword     dummyi           ;40=
.lword     dummyi           ;41=
.lword     dummyi           ;42=
.lword     dummyi           ;43=
.lword     dummyi           ;44=
.lword     dummyi           ;45=
.lword     dummyi           ;46=
.lword     dummyi           ;47=
.lword     dummyi           ;48=
.lword     dummyi           ;49=
.lword     dummyi           ;50=
.lword     dummyi           ;51=
.lword     dummyi           ;52=
.lword     dummyi           ;53=
.lword     dummyi           ;54=
.lword     dummyi           ;55=
.lword     dummyi           ;56=

```

```

.lword    dummy            ;57=
.lword    dummy            ;58=
.lword    dummy            ;59=
.lword    dummy            ;60=
.lword    dummy            ;61=
.lword    dummy            ;62=
.lword    dummy            ;63=

;
;////////////////////////////////////
; Non-Maskable interrupt vector table
;////////////////////////////////////
.section   vector,romdata
.org      vvector
;
.lword    undi              ; fffffc to f Undefined instruction interrupt
.lword    ovfli             ; fffffe0 to 3 Overflow interrupt
.lword    brki              ; fffffe4 to 7 BRK instruction interrupt
.lword    addri             ; fffffe8 to b Address match interrupt
.lword    dummy            ; fffffec to f
.lword    wdti              ; ffffff0 to 3 Watch-dog timer interrupt
.lword    dummy            ; ffffff4 to 7
.lword    nmii              ; ffffff8 to b NMI interrupt
.lword    reset             ; ffffffc to f RESET
;
.end

```

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2004.1.15	-	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しては、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、その他お気付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。