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い申し上げます。

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。
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て、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が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/62P
## シリアルインタフェースの特殊モード１（I²Cモード）の応用

### 要約

M16C/62Pグループは、シリアルインターフェース（UART）に特殊モード１（I²Cモード）を搭載しています。
本モードをソフトウェアと組み合わせて使用することにより、I²Cバスインターフェースの制御が可能となります。
本アプリケーションノートでは、I²Cモードの概要を説明し、M16C/62PでI²Cバスインターフェースを実現するための各機能の使い方およびプログラムをご紹介致します。
以下、シリアルインターフェース特殊モード１（I²Cモード）を簡易I²Cと記します。

### はじめに

この資料は、ルネサス 16 ビットシングルチップマイクロコンピュータM16C/62Pグループに内蔵されている簡易I²Cを制御していただくための参考資料です。なお、当資料はI²Cバスの通信動作を保証するものではありませんので、ご使用の際には十分に評価を行ってください。
M16C/62P グループの命令体系については、「M16C/60 シリーズ ソフトウエアマニュアル」を合わせてご利用ください。M16C/62P グループのハードウェアにつきましては、ご使用品種のハードウェアマニュアルを、開発サポートツールにつきましては、各ツールの操作説明書をご利用ください。
この資料で説明する応用例は、次のマイコン／条件での利用に適用されます。

・MCU　　M16C/62P グループ（M3062XFXPXP/M3062XMXP-XXXXP）

本書を使用するにあたって、電気回路／論理回路／I²Cバス仕様およびマイクロコンピュータの基本的な知識が必要です。

### 目次

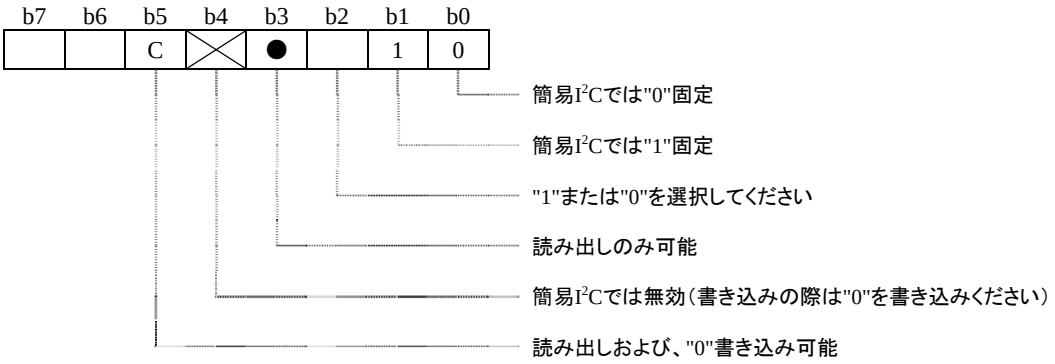


※I²C-BUSは、オランダPhilips社の登録商標です。

## 1. 簡易I²Cの各機能

本章では、M16C/62Pを使って、I²Cバスインターフェースを実現するための各ハードウェア機能の使用方法について、説明します。

［レジスタ設定の見方］



## 1.1 バイトデータの送信／受信の方法

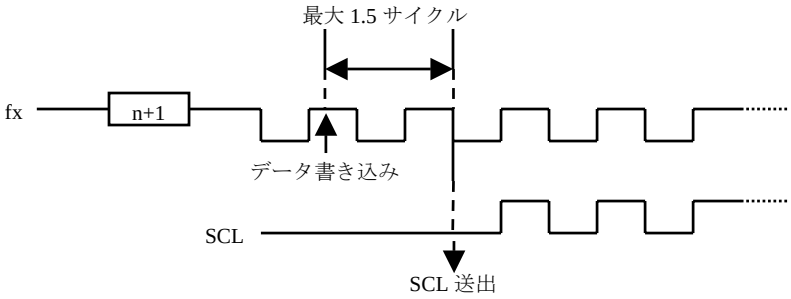
M16C/62Pをマスタとして使用し、簡易I²CのSCLを送出する際の設定方法、および 1 バイトのデータを送信／受信する場合の設定方法を説明します。

### 1.1.1 SCL 生成方法（マスタ時）

M16C/62P をマスタとして使用する場合、送出クロック(SCL)の速度を設定する必要があります。
それらは、通常のシリアル I/O 送信と同様に、以下のレジスタで設定します。
SCL は、送信バッファにデータを書き込み後、SCL の 1.5 サイクル以内に送出されます。

［SCL 送出タイミング］

fx　：BRG カウントソース
n　：iBRG 設定値(i=0～2)
SCL　：CKPH=1 の時の波形




[CKPH]の詳細については、1.6.5「クロック遅延機能」を参照してください。

［関連レジスタ］

UARTi 送受信モードレジスタ　(i=0～2)

U0MR：$03A0_{16}$番地　U1MR：$03A8_{16}$番地　U2MR：$0378_{16}$番地

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| 0 | × | × | × | 0 | 0 | 1 | 0 |

[SMD]　010：I²Cモード
[CKDIR]　0：内部クロックを選択（マスタ時）
1：外部クロックを選択（スレーブ時）
[IOPOL]　0：極性反転なし

UARTi 特殊モードレジスタ　(i=0～2)

U0SMR：$036F_{16}$番地　U1SMR：$0373_{16}$番地　U2SMR：$0377_{16}$番地

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | C | | 1 |

[IICM]　1：I²Cモード
[ABC]　0：アービトレーション・ロストをビット毎に更新
1：アービトレーション・ロストをバイト毎に更新

UARTi 送受信制御レジスタ0　(i=0～2)

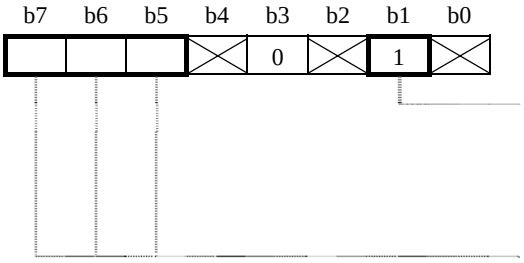
U0C0：$03A4_{16}$番地　U1C0：$03AC_{16}$番地　U2C0：$037C_{16}$番地

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 1 | ● | × | | |

[CLK]　BRGカウントソースを選択
00：$f_{1SIO}$または$f_{2SIO}$を選択
01：$f_8$を選択
10：$f_{32}$を選択
11：使用禁止
[CRD]　1:CTS／RTS 機能禁止
[NCH]　1：SCL、SDA 端子は Nch オープンドレイン出力(注 1)
[CKPOL]　0：CLK 極性は SCL 立ち下がりで送信データ出力、立ち上がりで入力
[UFORM]　1：転送フォーマット　MSB ファースト

注1　UART2 の SDA、SCL 端子は Nch オープンドレイン端子です。CMOS 出力は設定できません。
このとき、U2C0 の bit5 は"0"を設定してください。

UARTi 特殊モードレジスタ 3 (i=0～2)

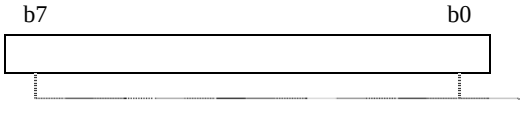



U0SMR3:036D$_{16}$番地 U1SMR3:0371$_{16}$番地 U2SMR3:0375$_{16}$番地

0:クロック遅延なし
1:クロック遅延あり
デジタル遅延値を設定
000:遅延なし
001:BRG カウントソースの 2 サイクル
010:BRG カウントソースの 3 サイクル
011:BRG カウントソースの 4 サイクル
100:BRG カウントソースの 5 サイクル
101:BRG カウントソースの 6 サイクル
110:BRG カウントソースの 7 サイクル
111:BRG カウントソースの 8 サイクル

[CKPH]の詳細については、1.6.5「クロック遅延機能」を参照してください。
[DL2][DL1][DL0]の詳細については、1.6.6「UART 出力デジタル遅延機能」を参照してください。

UARTi 転送速度レジスタ (i=0～2)




U0BRG:03A1$_{16}$番地 U1BRG:03A9$_{16}$番地 U2BRG:0379$_{16}$番地

［設定例］

源発振 10MHz で使用している場合、送信速度を 100kbps に設定する場合は、
- UiMR=00000010$_2$（I²Cモード、内部クロック選択）
- UiC0=10010000$_2$（BRGカウントソースに$f_{1SIO}$を選択）
- UiBRG=49
- PCLK1=1（PCLKRレジスタのSI/Oクロック選択ビットで$f_1$を選択）

［スレーブ時の設定］

スレーブとして使用する際には、UARTi 送受信モードレジスタ(UiMR)の bit3[CKDIR]を"1"に設定し、外部クロックを選択してください。そのとき、BRG カウントソース選択ビット[CLK0] [CLK1]および UARTi 転送速度レジスタ(UiBRG)の設定は無効となります。

### 1.1.2 バイトデータの送信方法

M16C/62P を送信装置として使用する場合、SDA 端子から 8 ビットの送信データを送出しますが、送信クロックの 9 ビット目では、アクノリッジを受信するために、M16C/62P の SDA 端子を解放する必要があります。これは、送信バッファに設定するデータで操作可能です。

送信バッファには、9 ビットのデータを設定します。I²Cバスでは、MSBファーストでデータを送信します。M16C/62Pでは、転送フォーマットをMSBファーストにして 9 ビットの設定したとき、データはb7→b6→・・・→b0→b8 の順で送出されます。

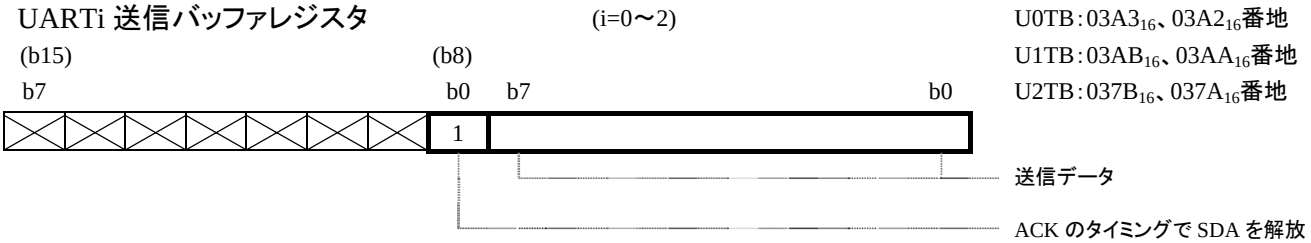
したがって、最上位ビットを送出したときが、アクノリッジを受信するタイミングとなります。

このときに、SDA 端子を解放するためには、最上位ビットに"1"を設定して、M16C/62P の SDA 端子をハイインピーダンス状態にします。以上が、バイトデータの送信方法です。

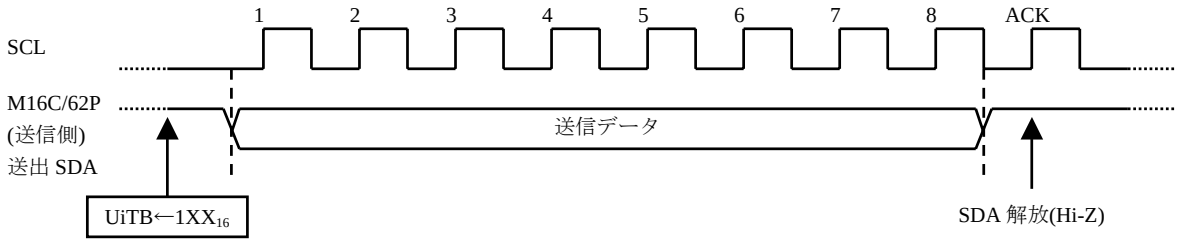
［関連レジスタ］

UARTi 送信バッファレジスタ (i=0～2)

U0TB：$03A3_{16}$、$03A2_{16}$番地
U1TB：$03AB_{16}$、$03AA_{16}$番地
U2TB：$037B_{16}$、$037A_{16}$番地




［タイミング図］



### 1.1.3 バイトデータの受信方法

M16C/62P を受信装置として使用する場合、SDA 端子から 8 ビットのデータを受信する間は、M16C/62P の SDA 端子を解放する必要があります。

また、クロックの 9 ビット目では、SDA 端子を"L"にしてアクノリッジを生成する必要があります。すでに、マスタ側が指定した受信装置のアドレス判定が終了し、自装置に対して送信が行われていることが確定している場合には、送信バッファに書き込む値でアクノリッジの送出を簡単に操作可能です。

M16C/62Pでは、データ受信時も、送信バッファに 9 ビットのデータをダミーデータ($0FF_{16}$)として設定します。下位 8 ビットを送出している間、SDA端子を解放するためには、下位 8 ビットに"1"を設定します。

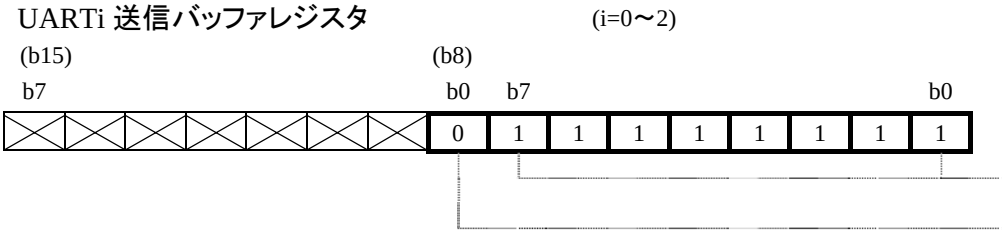
また、アクノリッジ生成のためには、最後に送出するビット(b8)に"0"を設定します。

以上が、バイトデータの受信方法です。

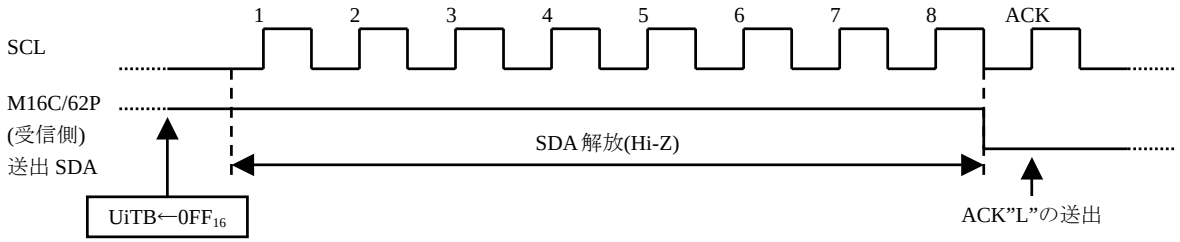
［関連レジスタ］

UARTi 送信バッファレジスタ (i=0～2)

U0TB：$03A3_{16}$、$03A2_{16}$番地
U1TB：$03AB_{16}$、$03AA_{16}$番地
U2TB：$037B_{16}$、$037A_{16}$番地




［タイミング図］



### 1.1.4 送信割り込み／受信割り込み

M16C/62P を送信装置として使用する場合、データ送出の完了は、「UARTi 送信割り込み」で検出可能です。M16C/62P を受信装置として使用する場合、データ受信の完了は、「UARTi 受信割り込み」で検出可能です。

これらの割り込みは、割り込み番号 15～20 に割り当てられており、I²Cモード選択ビット 2[IICM2]を"1"に設定することで、割り込み要因がUARTi送信およびUARTi受信になります。

送信割り込み発生タイミングは、UARTi 送信割り込み要因選択ビット[UiIRS]が"0"のとき、送信クロックの開始ビットの立ち下がりとなり、[UiIRS]が"1"のとき、次データの 1 ビット目の立ち下がり([CKPH]="1"のとき)となります。

また、受信割り込み発生タイミングは、受信クロックの最終ビットの立ち下がりとなります。なお、受信クロックの最終ビットの立ち上がり前（簡易I²Cの受信割り込み中等）に受信バッファを読み出すと、受信データはビットの位置が変化した状態で読み出されますのでご注意ください。（後述のタイミング図を参照してください。）

注） 本アプリケーションノートのサンプルプログラムでは、送信割り込みで受信データを読み出しています。

［関連レジスタ］

UARTi 特殊モードレジスタ (i=0～2)

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | C | | 1 |

U0SMR：$036F_{16}$番地 U1SMR：$0373_{16}$番地 U2SMR：$0377_{16}$番地

[IICM] 1：I²Cモード

[ABC] 0：アービトレーション・ロストをビット毎に更新
1：アービトレーション・ロストをバイト毎に更新

UARTi 特殊モードレジスタ 2 (i=0～2)

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| ☒ | | | | | | | 1 |

U0SMR2：$036E_{16}$番地 U1SMR2：$0372_{16}$番地 U2SMR2：$0376_{16}$番地

[IICM2] 1：UARTi 送信／受信割り込み有効

[CSC] 0：クロック同期化禁止
1：クロック同期化許可

[SWC] 0：SCL ウエイト出力禁止
1：SCL クロック出力許可

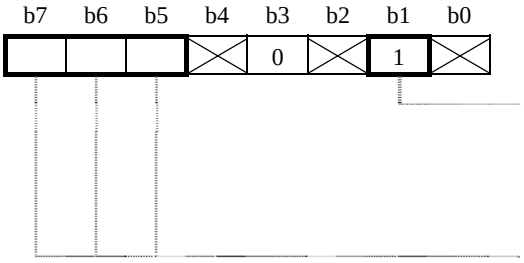
[ALS] 0：SDA 出力停止禁止
1：SDA 出力停止許可

[STAC] 0：UARTi 初期化禁止
1：UARTi 初期化許可

[SWC2] 0：SCL は UARTi クロックを出力
1：SCL から"L"を出力（SCL 端子 L 出力機能有効）

[SDHI] 0：SDA 出力許可
1：SDA 出力禁止（ハイインピーダンス）

UARTi 特殊モードレジスタ 3 (i=0～2)



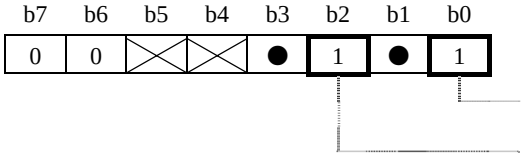

U0SMR3：036D$_{16}$番地　U1SMR3：0371$_{16}$番地　U2SMR3：0375$_{16}$番地

[CKPH]
0：クロック遅延なし
1：クロック遅延あり

[DL]
デジタル遅延値を設定
000：遅延なし
001：BRG カウントソースの 2 サイクル
010：BRG カウントソースの 3 サイクル
011：BRG カウントソースの 4 サイクル
100：BRG カウントソースの 5 サイクル
101：BRG カウントソースの 6 サイクル
110：BRG カウントソースの 7 サイクル
111：BRG カウントソースの 8 サイクル

[CKPH]の詳細については、1.6.5「クロック遅延機能」を参照してください。
[DL2][DL1][DL0]の詳細については、1.6.6「UART 出力デジタル遅延機能」を参照してください。

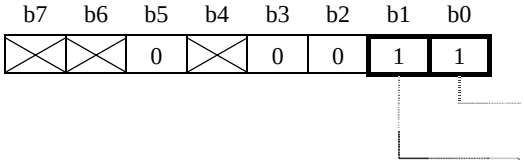
UARTi 送受信制御レジスタ 1 (i=0～1)




U0C1：03A5$_{16}$番地　U1C1：03AD$_{16}$番地

[TE] 1：送信許可
[RE] 1：受信許可

UART 送受信制御レジスタ 2

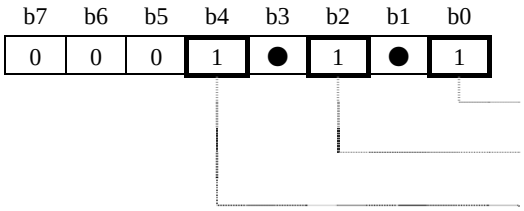



UCON：03B0$_{16}$番地

[U0IRS] UART0 送信割り込み要因選択
1：送信完了（最終ビットの立ち上がり）

[U1IRS] UART1 送信割り込み要因選択
1：送信完了（最終ビットの立ち上がり）

UART2 送受信制御レジスタ 1




U2C1：037D$_{16}$番地

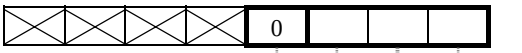
[TE] 1：送信許可
[RE] 1：受信許可

[U2IRS] UART2 送信割り込み要因選択
1：送信完了（最終ビットの立ち上がり）

UARTi 送信割り込み制御レジスタ (i=0～2)

b7 b6 b5 b4 b3 b2 b1 b0

S0TIC：$0051_{16}$番地　S1TIC：$0053_{16}$番地　S2TIC：$004F_{16}$番地



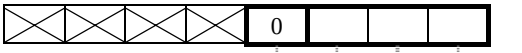

[ILVL]　割り込み優先レベル
1～7：割り込みレベル設定
（割り込みを使用しないときは"0"(禁止)にしてください）

[IR]　0：割り込み要求発生時"1"にセットされます

UARTi 受信割り込み制御レジスタ (i=0～2)

b7 b6 b5 b4 b3 b2 b1 b0

S0RIC：$0052_{16}$番地　S1RIC：$0054_{16}$番地　S2RIC：$0050_{16}$番地

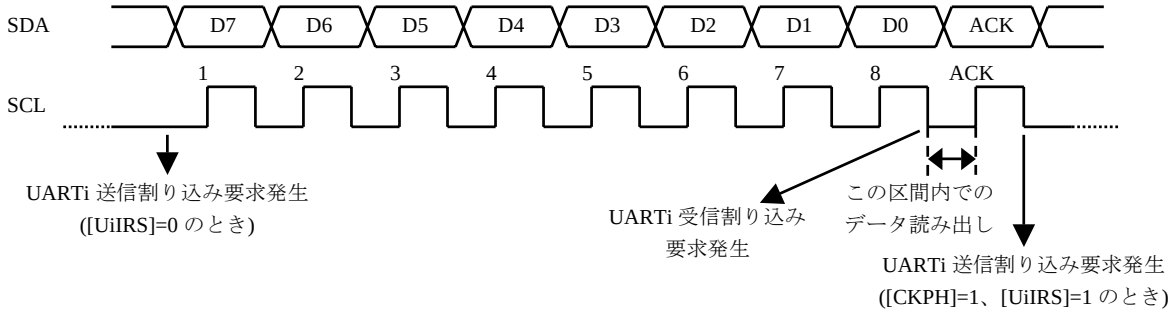



[ILVL]　割り込み優先レベル
1～7：割り込みレベル設定
（割り込みを使用しないときは"0"(禁止)にしてください）

[IR]　0：割り込み要求発生時"1"にセットされます

割り込み要求により割り込みを発生させる場合は、I フラグを"1"に設定してください。

［タイミング図］（IICM2=1 のとき）




UARTi 受信バッファレジスタ (i=0～2)

(b15) (b8)

b7 b0 b7 b0

U0RB：$03A7_{16}$、$03A6_{16}$番地
U1RB：$03AF_{16}$、$03AE_{16}$番地
U2RB：$037F_{16}$、$037E_{16}$番地


受信した 8 ビットのデータは
このように格納される。

不定

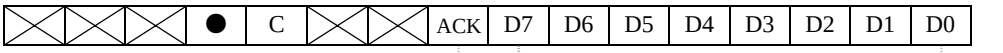
受信クロックの 8 ビット目から受信クロックの最終ビットの立ち上がり前（簡易$I^2C$の受信割り込み中等）に、データを読み出すと、図のようにD0,--,D7～D1 と格納されます。この後、受信クロックの最終ビットの立ち上がり後（簡易$I^2C$の送信割り込み中等）に、データを読み出すと、ACK、D7～D0 とデータが出力されます。

UARTi 受信バッファレジスタ(上記区間後) (i=0～2)

(b15) (b8)

b7 b0 b7 b0

U0RB：$03A7_{16}$、$03A6_{16}$番地
U1RB：$03AF_{16}$、$03AE_{16}$番地
U2RB：$037F_{16}$、$037E_{16}$番地




受信データ

## 1.2 スタートコンディション／ストップコンディションの詳細

データ送受信の開始時には、開始条件（スタートコンディション）、データ送受信の終了時には、停止条件（ストップコンディション）が発生します。

M16C/62P をスレーブとして使用する場合、マスタの生成するスタートコンディション、およびストップコンディションを検出するために、「スタートコンディション／ストップコンディション検出割り込み」機能をハードウェアで備えています。

M16C/62P をマスタとする場合は、スタートコンディションおよびストップコンディションを生成し、送出する機能をハードウェアで備えています。

スタートコンディション生成時にバスの使用状態を検知するための機能として、「バスビジー検出」機能をハードウェアで備えています。

スタートコンディション送出後から通信開始までの間に、他デバイスからクロック送出を禁止するための機能として、SCL から強制的に"L"を出力する「SCL 端子 L 出力機能 2」機能をハードウェアで備えています。

### 1.2.1 スタートコンディション／ストップコンディション検出

M16C/62P は、スタートコンディション／ストップコンディションを「スタートコンディション／ストップコンディション検出割り込み」で検出することが可能です。

本割り込みは、「ソフトウェア割り込み番号 6,7,10（注 1）」に割り当てられ、I²Cモード選択時（[IICM]="1"）、割り込み番号 6,7,10（注 1）の割り込み要因が、「スタートコンディション／ストップコンディション検出割り込み」に変化します。この割り込みを検出した場合は、バスビジーフラグ(BBS)の状態で、スタートコンディションまたはストップコンディションのどちらが発生したのかを判断してください。

なお、スタートコンディションおよびストップコンディション検出のセットアップタイム，ホールドタイムは、I²Cバス規格と異なる場合がありますので、ご注意ください。（2.1.1「スタート／ストップコンディションのセットアップタイム・ホールドタイム」参照）。

注1 割り込み番号 6,7 の要因は、それぞれ UART1 および UART0 を選択して使用します。本項「割り込み要因選択レジスタ 2」参照。

［関連レジスタ］

UARTi 特殊モードレジスタ (i=0～2)

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | C | | 1 |

U0SMR：$036F_{16}$番地 U1SMR：$0373_{16}$番地 U2SMR：$0377_{16}$番地

[IICM] 1：I²Cモード

[ABC] 0：アービトレーション・ロストをビット毎に更新
1：アービトレーション・ロストをバイト毎に更新

UARTi バス衝突検出割り込み制御レジスタ (i=0～2)

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| | | | | 0 | | | |

TB3IC：$0045_{16}$番地 TB4IC：$0046_{16}$番地 BCNIC：$004A_{16}$番地

[ILVL] 割り込み優先レベル
1～7：割り込みレベル設定
（割り込みを使用しないときは"0"(禁止)にしてください）

[IR] 0：割り込み要求発生時"1"にセットされます

割り込み要因選択レジスタ 2

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| | | | | | | | |

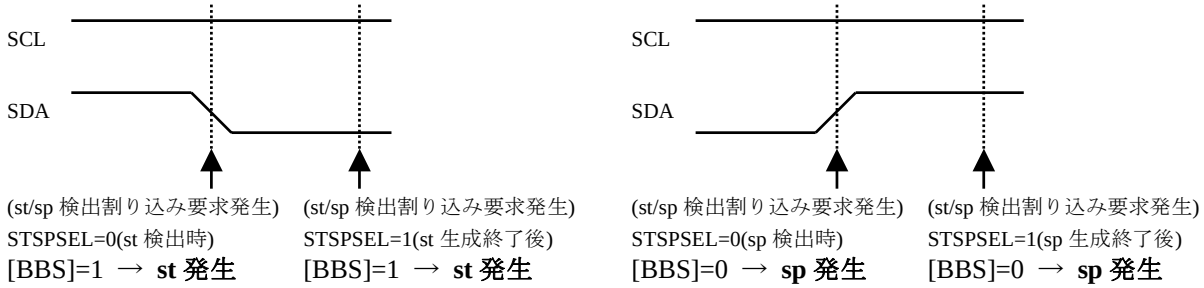
IFSR2A：$035E_{16}$番地

[IFSR26] UART0／タイマ B3 割り込み要因選択ビット
0：タイマ B3 割り込み
1：UART0 バス衝突／スタートストップ検出／障害エラー検出

[IFSR27] UART1／タイマ B4 割り込み要因選択ビット
0：タイマ B4 割り込み
1：UART1 バス衝突／スタートストップ検出／障害エラー検出

割り込み要求により割り込みを発生させる場合は、I フラグを"1"に設定してください。

［タイミング図］
st　　：スタートコンディション
sp　　：ストップコンディション



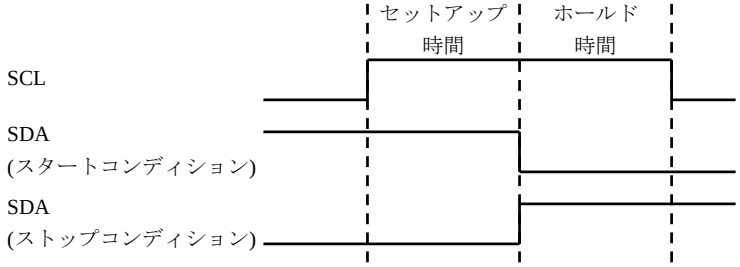

(st/sp 検出割り込み要求発生)
STSPSEL=0(st 検出時)
[BBS]=1 → **st 発生**

(st/sp 検出割り込み要求発生)
STSPSEL=1(st 生成終了後)
[BBS]=1 → **st 発生**

(st/sp 検出割り込み要求発生)
STSPSEL=0(sp 検出時)
[BBS]=0 → **sp 発生**

(st/sp 検出割り込み要求発生)
STSPSEL=1(sp 生成終了後)
[BBS]=0 → **sp 発生**

3～6 サイクル ＜ セットアップ時間（注 1）
3～6 サイクル ＜ ホールド時間（注 1）




注 1　PCLKR レジスタの PCLK1 ビットが”1”の場合は $f_{1SIO}$ の、
PCLK1 ビットが”0”の場合は $f_{2SIO}$ のサイクル数です。

### 1.2.2 スタートコンディション／ストップコンディション／リスタートコンディション送出

M16C/62P をマスタとして使用する場合のスタートコンディション、ストップコンディション、およびリスタートコンディションは、ハードウェアで生成可能です。

STAREQ ビットを"1"（スタート）にすると、スタートコンディションを生成します。

STPREQ ビットを"1"（スタート）にすると、（SCL が"L"のときは、SCL の解放を待って、）ストップコンディションを生成します。

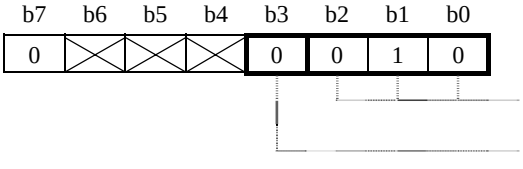
RSTREQ ビットを"1"（スタート）にすると、（SCL が"L"のときは、SCL の解放を待って、）リスタートコンディションを生成します。

STSPSEL ビットを"1"にすると、上記ビットで生成した各コンディションの出力を行います。

各コンディション生成の際は、必ず REQ ビットを"1"にしてから、STSPSEL を"1"にしてください。

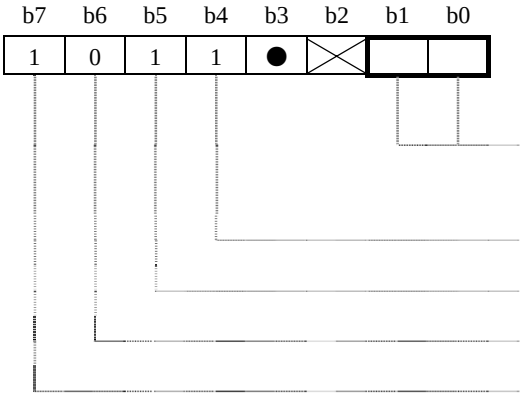
［関連レジスタ］

UARTi 送受信モードレジスタ (i=0～2)




U0MR：$03A0_{16}$番地 U1MR：$03A8_{16}$番地 U2MR：$0378_{16}$番地

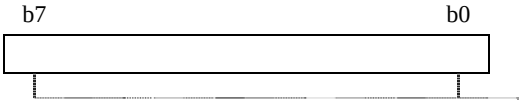
UARTi 送受信制御レジスタ0 (i=0～2)




U0C0：$03A4_{16}$番地 U1C0：$03AC_{16}$番地 U2C0：$037C_{16}$番地

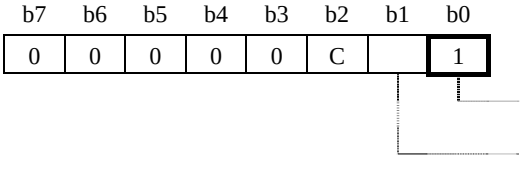
注1 UART2 の SDA、SCL 端子は Nch オープンドレイン端子です。CMOS 出力は設定できません。
このとき、U2C0 の bit5 は"0"を設定してください。

UARTi 転送速度レジスタ (i=0～2)



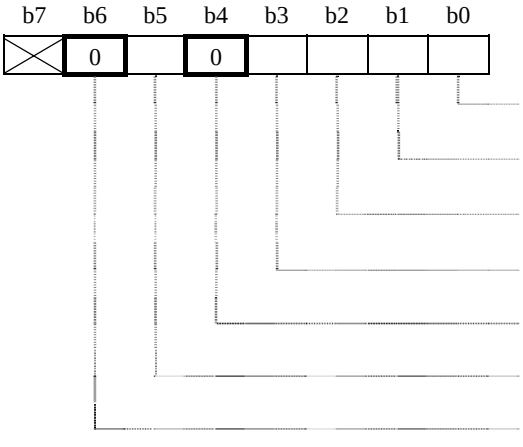

U0BRG：$03A1_{16}$番地 U1BRG：$03A9_{16}$番地 U2BRG：$0379_{16}$番地

UARTi 特殊モードレジスタ (i=0～2)




U0SMR：$036F_{16}$番地 U1SMR：$0373_{16}$番地 U2SMR：$0377_{16}$番地

UARTi 特殊モードレジスタ 2 (i=0～2)




U0SMR2：036E$_{16}$番地　U1SMR2：0372$_{16}$番地　U2SMR2：0376$_{16}$番地

[IICM2]
0：アクノリッジ検出／アクノリッジ未検出割り込み有効
1：UARTi 送信／受信割り込み有効

[CSC]
0：クロック同期化禁止
1：クロック同期化許可

[SWC]
0：SCL ウエイト出力禁止
1：SCL クロック出力許可

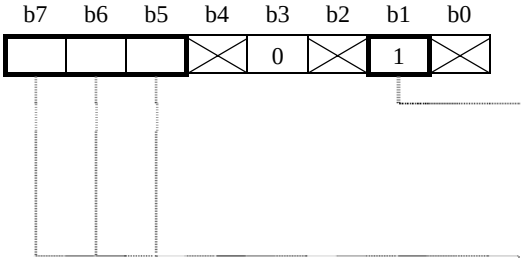
[ALS]
0：SDA 出力停止禁止
1：SDA 出力停止許可

[STAC]
0：UARTi 初期化禁止

[SWC2]
0：SCL は UARTi クロックを出力
1：SCL から"L"を出力（SCL 端子 L 出力機能有効）

[SDHI]
0：SDA 出力許可

UARTi 特殊モードレジスタ 3 (i=0～2)



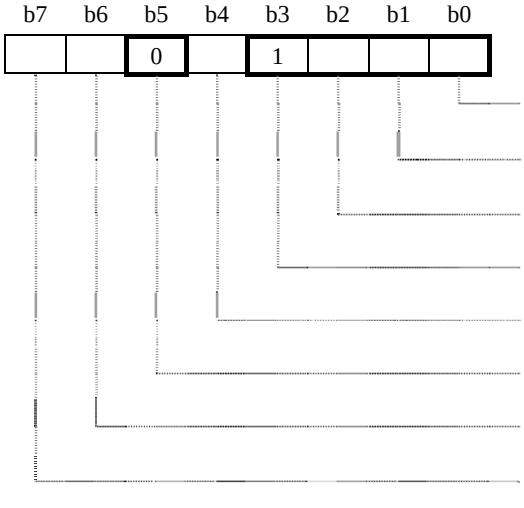

U0SMR3：036D$_{16}$番地　U1SMR3：0371$_{16}$番地　U2SMR3：0375$_{16}$番地

[CKPH]
0：クロック遅延なし
1：クロック遅延あり

[DL]
デジタル遅延値を設定
000：遅延なし
001：BRG カウントソースの 2 サイクル
010：BRG カウントソースの 3 サイクル
011：BRG カウントソースの 4 サイクル
100：BRG カウントソースの 5 サイクル
101：BRG カウントソースの 6 サイクル
110：BRG カウントソースの 7 サイクル
111：BRG カウントソースの 8 サイクル

[CKPH]の詳細については、1.6.5「クロック遅延機能」を参照してください。
[DL2][DL1][DL0]の詳細については、1.6.6「UART 出力デジタル遅延機能」を参照してください。

UARTi 特殊モードレジスタ 4 (i=0～2)




U0SMR4：036C$_{16}$番地　U1SMR4：0370$_{16}$番地　U2SMR4：0374$_{16}$番地

[STAREQ]
0：クリア
1：スタート

[RSTAREQ]
0：クリア
1：スタート

[STPREQ]
0：クリア
1：スタート

[STSPSEL]
1：スタート／ストップコンディションを出力する

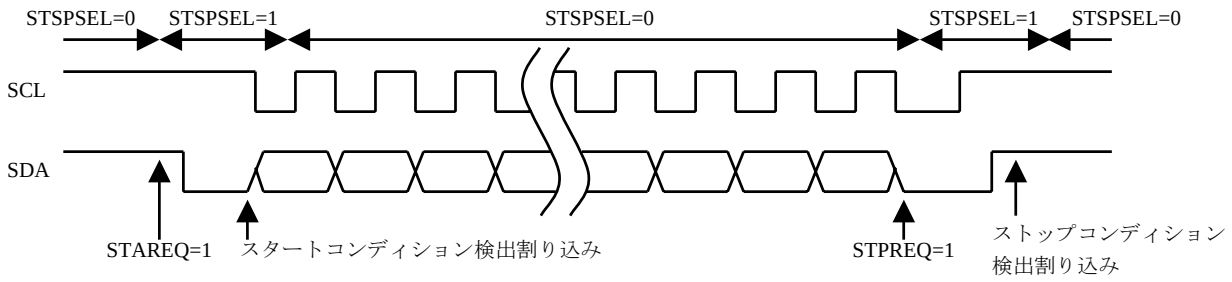
[ACKD]
0：ACK
1：NACK

[ACKC]
0：SI/O データ出力

[SCLHI]
0：禁止
1：許可

[SWC9]
0：SCL"L"ホールド禁止
1：SCL"L"ホールド許可

［タイミング図］


STSPSEL=0
STSPSEL=1
STSPSEL=0
STSPSEL=1
STSPSEL=0
SCL
SDA
STAREQ=1
スタートコンディション検出割り込み
STPREQ=1
ストップコンディション
検出割り込み

### 1.2.3 バスビジー検出

スタートコンディションを送出する前には、バスが解放されていることを確認する必要があります。

M16C/62Pの簡易I²Cは、バスの状態をバスビジーフラグ[BBS]で検出可能です。スタートコンディションを検出したとき、[BBS]は"1"にセットされ、ストップコンディションを検出したとき、[BBS]は"0"にクリアされます。

したがって、[BBS]="1"であったときにスタートコンディションを送出しようとした場合は、[BBS]="0"を待ってから、送出を開始する必要があります。

［関連レジスタ］

UARTi 特殊モードレジスタ (i=0～2)

U0SMR：$036F_{16}$番地　U1SMR：$0373_{16}$番地　U2SMR：$0377_{16}$番地

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | C | | 1 |

[IICM] 1：I²Cモード

[ABC] 0：アービトレーション・ロストをビット毎に更新
1：アービトレーション・ロストをバイト毎に更新

[BBS] 0：バスは解放状態
1：バスは使用状態 （"0"のみ書き込み可）

### 1.2.4 SCL 端子 L 出力機能 2

下記のビットずれ現象は、クロック遅延機能（1.6.5「クロック遅延機能」参照）の使用により、回避することが可能ですが、新機能を使用していない従来方式の簡易I²Cファームウェアの流用時を前提に説明します。

M16C/62P のシリアル I/O は、送信データを送信バッファに書き込んでから転送クロック（簡易 I2C では SCL）送出までに、最大で転送クロック(SCL)の 1.5 サイクルの期間を要します。

また、M16C/62P の SCL 同期化機能（1.5.3「SCL 同期化機能」参照）は、1 ビット目の SCL 送出時から有効になりますので、スタートコンディション発生後クロックライン(SCL)同期化機能が有効になるまでの期間内に、他デバイスが 1 ビット目の送出を行った場合、ビットずれを起こす可能性があります（タイミング図左図）。

このため、M16C/62P は、スタートコンディション送出後に、他デバイスからのクロック送出を禁止するための SCL 端子 L 出力機能を持っています。この機能を使用することにより、送信バッファにデータを書き込むと同時に、SCL 端子から"L"を出力し、他デバイスを待ち状態することが可能です（タイミング図右図）。本機能はウエイト出力ビット 2[SWC2]を"1"にすることで動作が許可され（SCL から"L"出力）、"0"にすることで解除されます。

注） 本アプリケーションノートのサンプルプログラムでは、クロック遅延機能を使用しています。

［関連レジスタ］

UARTi 特殊モードレジスタ 2 (i=0～2)

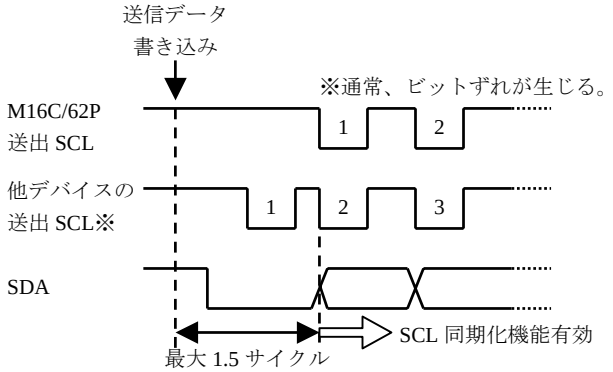
U0SMR2：$036E_{16}$番地 U1SMR2：$0372_{16}$番地 U2SMR2：$0376_{16}$番地

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| × | | 1 | | | | | |

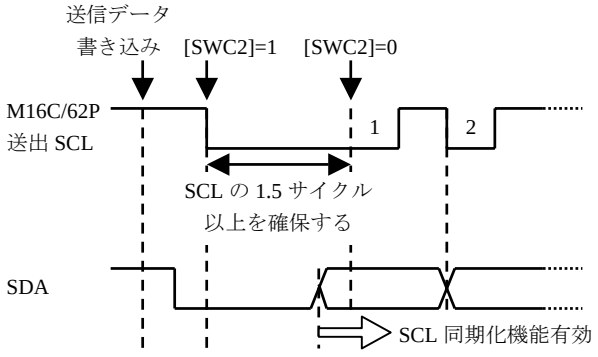
| | |
|---|---|
| [IICM2] | 0：アクノリッジ検出／アクノリッジ未検出割り込み有効<br>1：UARTi 送信／受信割り込み有効 |
| [CSC] | 0：クロック同期化禁止<br>1：クロック同期化許可 |
| [SWC] | 0：SCL ウエイト出力禁止<br>1：SCL クロック出力許可 |
| [ALS] | 0：SDA 出力停止禁止<br>1：SDA 出力停止許可 |
| [STAC] | 0：UARTi 初期化禁止<br>1：UARTi 初期化許可 |
| [SWC2] | 1：SCL から"L"を出力（SCL 端子 L 出力機能有効） |
| [SDHI] | 0：SDA 出力許可<br>1：SDA 出力禁止（ハイインピーダンス） |

［タイミング図］

SCL 端子 L 出力機能を使用しない場合




SCL 端子 L 出力機能を使用する場合



## 1.3 アクノリッジ

データの送受信には、1 バイト毎に確認応答（アクノリッジ）が付加されます。

M16C/62P を送信装置として使用する場合は、1 バイト毎に受信装置からのアクノリッジの有無を検出する必要があります。そのための機能として、「アクノリッジ検出割り込み」および「アクノリッジ未検出割り込み」をハードウェアで備えています。

また、M16C/62P を受信装置として使用し、一対一通信を行う場合は、送信データの 9 ビット目に"0"を設定することにより、容易にアクノリッジの生成が可能です。（1.1.3「バイトデータの受信方法」参照）

「アクノリッジ検出割り込み」および「アクノリッジ未検出割り込み」を使用する場合には、I²Cモード選択ビット 2(IICM2)を"0"に設定することが必要です。この設定により、割り込み番号 15,17,19 の要因、および割り込み番号 16,18,20 の要因が、それぞれ「アクノリッジ未検出割り込み」、「アクノリッジ検出割り込み」となります。その場合、UARTi受信レジスタから受信バッファレジスタへのデータ転送タイミングは、受信クロックの最終ビットの立ち上がりとなります。

注） 本アプリケーションノートのサンプルプログラムでは、アクノリッジ検出割り込み／アクノリッジ未検出割り込みを使用していません。

### 1.3.1 アクノリッジ検出

送信クロックの 9 ビット目の立ち上がり時に、送信装置側で解放している SDA のラインのレベルが、"L" になっている場合、受信装置側がアクノリッジを生成したと判断します。

M16C/62Pの場合、「アクノリッジ検出割り込み」機能で検出可能です。本割り込みは、「ソフトウェア割り込み番号 16,18,20」に割り当てられ、I²Cモード選択時（[IICM]="1"）で、かつI²Cモード選択ビット 2[IICM2]="0"のときのみ、割り込み番号 16,18,20 の割り込み要因が、「アクノリッジ検出割り込み」となります。

［関連レジスタ］

UARTi 特殊モードレジスタ (i=0～2)

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | C | | 1 |

U0SMR：$036F_{16}$番地　U1SMR：$0373_{16}$番地　U2SMR：$0377_{16}$番地

[IICM]　1：I²Cモード

[ABC]　0：アービトレーション・ロストをビット毎に更新
1：アービトレーション・ロストをバイト毎に更新

UARTi 特殊モードレジスタ 2 (i=0～2)

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| ⊠ | | | | | | | 0 |

U0SMR2：$036E_{16}$番地　U1SMR2：$0372_{16}$番地　U2SMR2：$0376_{16}$番地

[IICM2]　0：アクノリッジ検出／アクノリッジ未検出割り込み有効

[CSC]　0：クロック同期化禁止
1：クロック同期化許可

[SWC]　0：SCL ウエイト出力禁止
1：SCL クロック出力許可

[ALS]　0：SDA 出力停止禁止
1：SDA 出力停止許可

[STAC]　0：UARTi 初期化禁止
1：UARTi 初期化許可

[SWC2]　0：UARTi クロック
1："L"出力

[SDHI]　0：SDA 出力許可
1：SDA 出力禁止（ハイインピーダンス）

UARTi 受信割り込み制御レジスタ (i=0～2)　－アクノリッジ検出割り込み使用時－

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| ⊠ | ⊠ | ⊠ | ⊠ | 0 | | | |

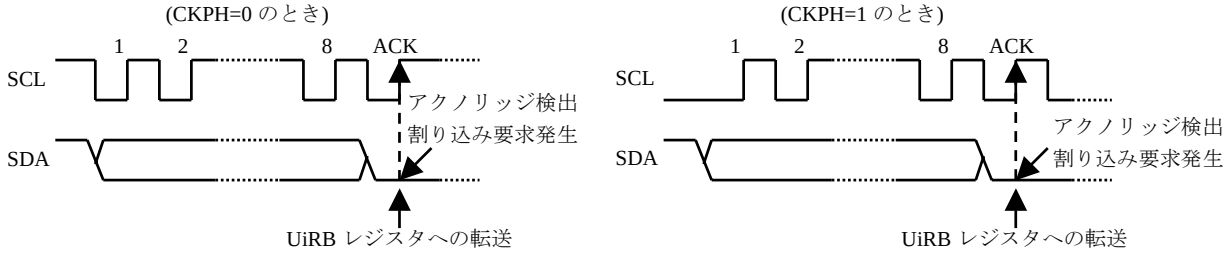
S0RIC：$0052_{16}$番地　S1RIC：$0054_{16}$番地　S2RIC：$0050_{16}$番地

[ILVL]　割り込み優先レベル
1～7：割り込みレベル設定
（割り込みを使用しないときは"0"(禁止)にしてください）

[IR]　0：割り込み要求発生時"1"にセットされます

割り込み要求により割り込みを発生させる場合は、I フラグを"1"に設定してください。

［タイミング図］



### 1.3.2 アクノリッジ未検出

送信クロックの 9 ビット目の立ち上がり時に、送信装置側で解放している SDA のラインのレベルが、"H" になっている場合、受信装置側がアクノリッジを生成していないと判断します。

M16C/62Pの場合、「アクノリッジ未検出割り込み」機能で検出可能です。本割り込みは「ソフトウェア割り込み番号 15,17,19」に割り当てられ、I²Cモード選択時（[IICM]="1"）で、かつI²Cモード選択ビット[IICM2]="0" のときのみ割り込み番号 15,17,19 の割り込み要因が、「アクノリッジ未検出割り込み」となります。

［関連レジスタ］

UARTi 特殊モードレジスタ (i=0～2)

U0SMR：$036F_{16}$番地　U1SMR：$0373_{16}$番地　U2SMR：$0377_{16}$番地

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | C | | 1 |

[IICM]　1：I²Cモード

[ABC]　0：アービトレーション・ロストをビット毎に更新
1：アービトレーション・ロストをバイト毎に更新

UARTi 特殊モードレジスタ 2 (i=0～2)

U0SMR2：$036E_{16}$番地　U1SMR2：$0372_{16}$番地　U2SMR2：$0376_{16}$番地

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| × | | | | | | | 0 |

[IICM2]　0：アクノリッジ検出／アクノリッジ未検出割り込み有効

[CSC]　0：クロック同期化禁止
1：クロック同期化許可

[SWC]　0：SCL ウエイト出力禁止
1：SCL クロック出力許可

[ALS]　0：SDA 出力停止禁止
1：SDA 出力停止許可

[STAC]　0：UARTi 初期化禁止
1：UARTi 初期化許可

[SWC2]　0：SCL は UARTi クロックを出力
1：SCL から"L"を出力（SCL 端子 L 出力機能有効）

[SDHI]　0：SDA 出力許可
1：SDA 出力禁止（ハイインピーダンス）

UARTi 送信割り込み制御レジスタ (i=0～2)　－アクノリッジ未検出割り込み使用時－

S0TIC：$0051_{16}$番地　S1TIC：$0053_{16}$番地　S2TIC：$004F_{16}$番地

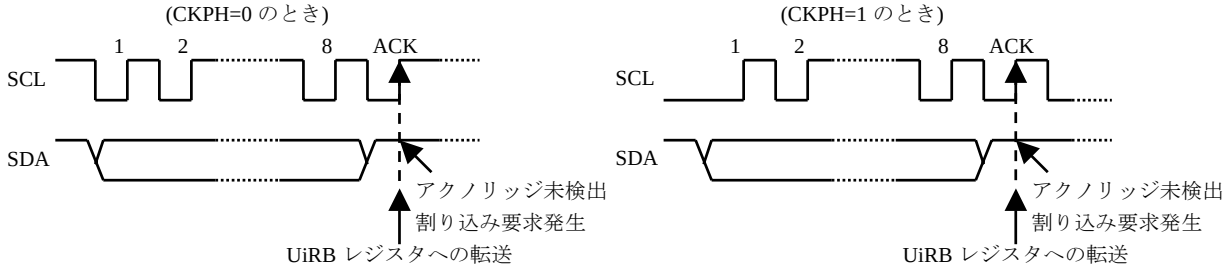
| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| × | × | × | × | 0 | | | |

[ILVL]　割り込み優先レベル
1～7：割り込みレベル設定
（割り込みを使用しないときは"0"(禁止)にしてください）

[IR]　0：割り込み要求発生時"1"にセットされます

割り込み要求により割り込みを発生させる場合は、I フラグを"1"に設定してください。

［タイミング図］



## 1.4 自己アドレス指定の判定

M16C/62P をスレーブとして使用している場合､マスタから送信されたアドレスが自己のアドレスと一致するか比較し、一致したときスレーブ(M16C/62P)は、マスタに対し、アクノリッジを送出します。

M16C/62Pの簡易I²Cでは、自己アドレスとの比較、およびアクノリッジの送出をソフトウェアで行いますが､その間､SCL端子をLホールドし､マスタを待ち状態にする「SCL端子L出力機能」､および「ACK／NACK送出機能」をハードウェアで備えています。

本サンプルソフトでは、SCL 端子の Low 中にソフトウェアでアドレスを比較する必要があるため、「SCL 端子 L 出力機能」（1.4.2「SCL 端子 L 出力機能」参照）を使用しています。

そのため、自己アドレスを指定されていない場合でもアドレス受信時は SCL 端子を L ホールドします。

### 1.4.1 ACK／NACK 送出機能

M16C/62P では、ACK／NACK の送出を送信データの 9 ビット目に設定するほかに、ACKC を"1"に設定して、ACKD ビットを制御することも可能です。

M16C/62P をスレーブとして使用している場合､マスタから送信されたアドレスが自己のアドレスと一致するか比較し、一致したときスレーブ(M16C/62P)は、マスタに対し、アクノリッジを送出します。

M16C/62Pの簡易I²Cでは、自己アドレスとの比較、およびアクノリッジの送出をソフトウェアで行いますが、その際に、ACKCを"1"に設定して、ACK／NACK送出を行います。

［関連レジスタ］

UARTi 特殊モードレジスタ (i=0～2)

U0SMR：$036F_{16}$番地 U1SMR：$0373_{16}$番地 U2SMR：$0377_{16}$番地

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | C | | 1 |

[IICM] 1：I²Cモード

[ABC] 0：アービトレーション・ロストをビット毎に更新
1：アービトレーション・ロストをバイト毎に更新

UARTi 特殊モードレジスタ 2 (i=0～2)

U0SMR2：$036E_{16}$番地 U1SMR2：$0372_{16}$番地 U2SMR2：$0376_{16}$番地

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| ☒ | | | | | | | 1 |

[IICM2] 1：UARTi 送信／受信割り込み有効

[CSC] 0：クロック同期化禁止
1：クロック同期化許可

[SWC] 0：SCL ウエイト出力禁止
1：SCL クロック出力許可

[ALS] 0：SDA 出力停止禁止
1：SDA 出力停止許可

[STAC] 0：UARTi 初期化禁止
1：UARTi 初期化許可

[SWC2] 0：SCL は UARTi クロックを出力
1：SCL から"L"を出力（SCL 端子 L 出力機能有効）

[SDHI] 0：SDA 出力許可
1：SDA 出力禁止（ハイインピーダンス）

UARTi 特殊モードレジスタ 4 (i=0～2)

U0SMR4：$036C_{16}$番地 U1SMR4：$0370_{16}$番地 U2SMR4：$0374_{16}$番地

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| | | 1 | | 0 | | | |

[STAREQ] 0：クリア
1：スタート

[RSTAREQ] 0：クリア
1：スタート

[STPREQ] 0：クリア
1：スタート

[STSPSEL] 0：スタート／ストップコンディションを出力しない

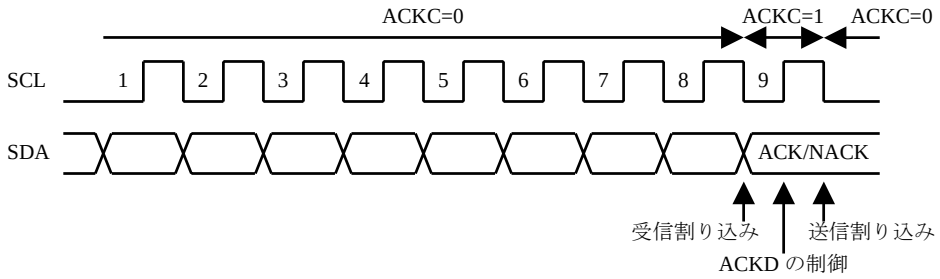
[ACKD] 0：ACK
1：NACK

[ACKC] 1：ACK データビット(ACKD)出力

[SCLHI] 0：禁止
1：許可

[SWC9] 0：SCL"L"ホールド禁止
1：SCL"L"ホールド許可

［タイミング図］


ACKC=0
ACKC=1
ACKC=0
SCL
1
2
3
4
5
6
7
8
9
SDA
ACK/NACK
受信割り込み
送信割り込み
ACKD の制御

### 1.4.2 SCL 端子 L 出力機能

I²Cバスでは、スタートコンディション検出後の第一バイトに、指定されるスレーブのアドレスが送出されます（7 ビットアドレスモード時）。

スレーブは、他のマスタから送出されるクロックのはじめ（第 1 バイト）の 7 ビットの受信データを自己アドレスと比較し、クロックの 9 ビット目に同期してアクノリッジを生成（または未生成）する処理が必要です。

本処理のための機能として、M16C/62P は、「SCL 端子 L 出力機能」を持っています。この機能を用いると、初め（第 1 バイト）の 8 ビットのデータを受信後、9 ビット目の SCL が"L"になるタイミングで、M16C/62P の SCL 端子に"L"を出力して、強制的にマスタを待ち状態にします。そして、ソフトウェアでのアドレス比較処理が終了後に、ACK／NACK 生成機能（1.4.1「ACK／NACK 送出機能」参照）でアクノリッジの生成／未生成を行うことが可能です。（一対一通信で使用している場合等は 1.1.3「バイトデータの受信方法」で説明した方法で、アドレスの受信、アクノリッジの生成が可能です。）

本機能は、ウエイト出力ビット[SWC]を"1"にすることにより、動作が許可され、"0"にすることにより、禁止されます。

また、本機能で SCL 端子が"L"レベルになったときは、[SWC]を"0"にすることで解除できます。

なお、本機能によりアドレス比較処理を行う場合、最終ビットのクロックの立ち上がり前に、受信バッファレジスタの内容を読み出すことになりますので、読み出した受信データは、ビットの位置が変化していることにご注意ください。（1.1.4「送信割り込み／受信割り込み」のタイミング図参照）

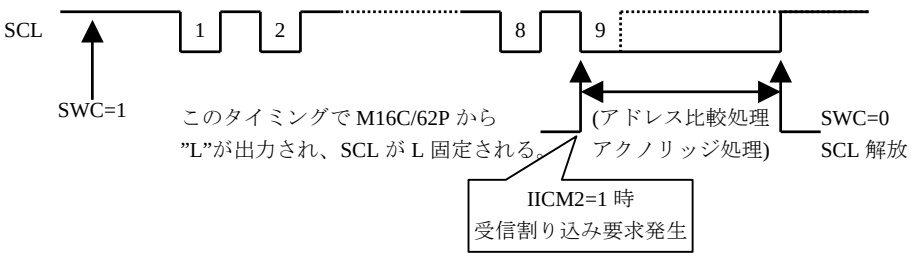
［関連レジスタ］

UARTi 特殊モードレジスタ 2 (i=0～2)

U0SMR2：036E$_{16}$番地 U1SMR2：0372$_{16}$番地 U2SMR2：0376$_{16}$番地

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 | | |
|---|---|---|---|---|---|---|---|---|---|
| × | | | | | 1 | | 1 | [IICM2] | 1：UARTi 送信／受信割り込み有効 |
| | | | | | | | | [CSC] | 0：クロック同期化禁止<br>1：クロック同期化許可 |
| | | | | | | | | [SWC] | 1：SCL クロック出力許可 |
| | | | | | | | | [ALS] | 0：SDA 出力停止禁止<br>1：SDA 出力停止許可 |
| | | | | | | | | [STAC] | 0：UARTi 初期化禁止<br>1：UARTi 初期化許可 |
| | | | | | | | | [SWC2] | 0：SCL は UARTi クロックを出力<br>1：SCL から"L"を出力（SCL 端子 L 出力機能有効） |
| | | | | | | | | [SDHI] | 0：SDA 出力許可<br>1：SDA 出力禁止（ハイインピーダンス） |

［タイミング図］

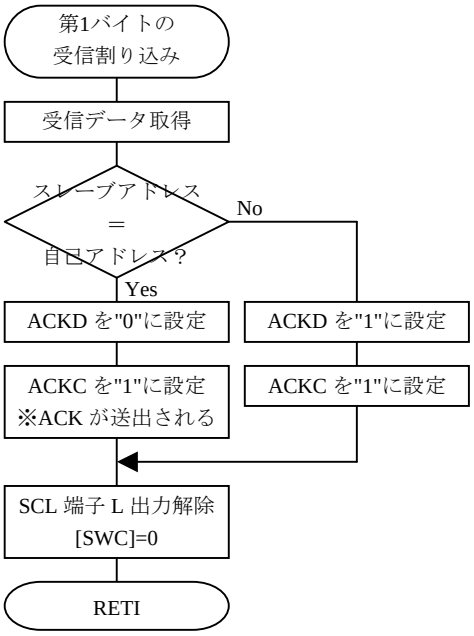


### 1.4.3 自己アドレス判定の一例

7 ビットアドレスの判定の一例をご紹介します。
スタートビット受信後、第 1 バイトの受信割り込みが発生したときの制御の一例をご紹介します。
第 1 バイト受信前に[SWC]="1"（SCL 端子 L 出力機能許可）をすでに設定しているものとし、受信割り込みルーチンの一部のみ記述します。

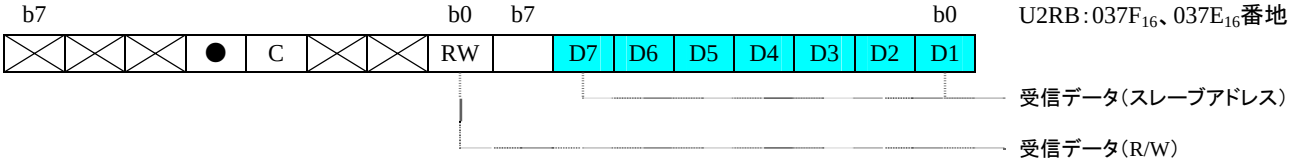
［フローチャート］

※メインルーチンでの設定（例）
- swc=1（SCL 端子 L 出力機能許可）
- IICM2=1（割り込み番号 16,18,20 の割り込み要因：受信割り込み、割り込み発生タイミングは最終ビットの立ち上がり）
- SiRIC=7(i=0～2)（受信割り込み許可）
- I フラグ=1（割り込み許可）



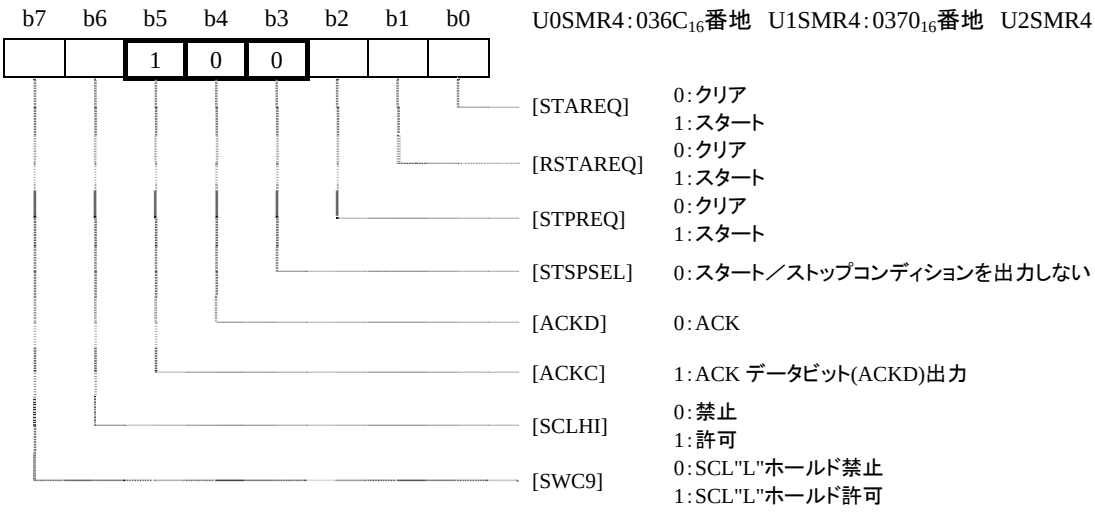

UARTi 受信バッファレジスタ (i=0～2)

U0RB：$03A7_{16}$、$03A6_{16}$番地
U1RB：$03AF_{16}$、$03AE_{16}$番地
U2RB：$037F_{16}$、$037E_{16}$番地




UARTi 特殊モードレジスタ 4 (i=0～2)

U0SMR4：$036C_{16}$番地　U1SMR4：$0370_{16}$番地　U2SMR4：$0374_{16}$番地



## 1.5 通信調整

I²Cバスをマルチマスタで使用する場合は、複数のマスタが同時にスタートコンディションを生成して、データ送信をスタートしようとすることが考えられます（アービトレーション発生）。

この場合、I²Cバスのシステムでは、複数のマスタ間で通信調整が行われます。

M16C/62Pの簡易I²Cでは、アービトレーション・ロスト発生時に通信を回復するための機能として、「アービトレーション・ロスト検出機能」、「アービトレーション・ロスト発生時のSDA出力禁止機能」をハードウェアで備えています。

また、アービトレーション・ロスト以外にも、クロック同期を利用した通信調整のひとつとして「SCL 同期化機能」も備えています。

### 1.5.1　　アービトレーション・ロスト検出機能　(i=0～2)

M16C/62Pの簡易I²Cは、アービトレーション・ロスト検出フラグ[ABT]を持っています。

この検出フラグは、UARTi 受信バッファレジスタのビット 3 に配置されており、SCL の立ち上がりのタイミングで、内部データ・レベルと SDA のレベルの不一致を検出すると、アービトレーション・ロスト検出フラグ[ABT]は"1"にセットされます。

アービトレーション・ロスト検出フラグの更新は、アービトレーション・ロスト検出フラグ制御[ABC]により、ビット毎（"0"）／バイト毎（"1"）の選択が可能です。

アクノリッジ受信のタイミングでは、出力"H"，入力"L"となり、アービトレーション・ロスト検出フラグがセットされてしまいますので、次の送信時には、アービトレーション・ロスト検出フラグを"0"にクリアしてから送信を行ってください。

注）　本アプリケーションノートのサンプルプログラムでは、[ABC]=0 に固定してください。

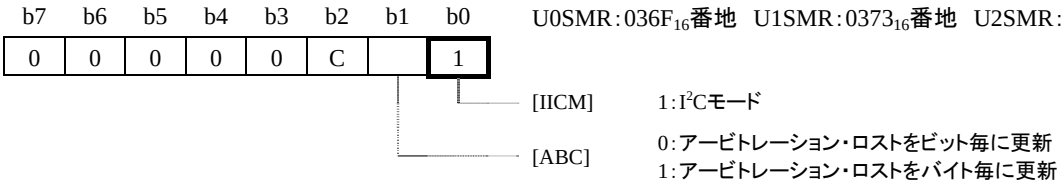
［関連レジスタ］

UARTi 受信バッファレジスタ　(i=0～2)

U0RB：$03A7_{16}$、$03A6_{16}$番地
U1RB：$03AF_{16}$、$03AE_{16}$番地
U2RB：$037F_{16}$、$037E_{16}$番地

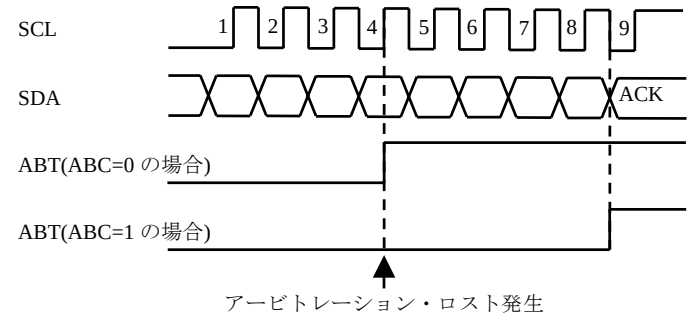

UARTi 特殊モードレジスタ　(i=0～2)

U0SMR：$036F_{16}$番地　U1SMR：$0373_{16}$番地　U2SMR：$0377_{16}$番地

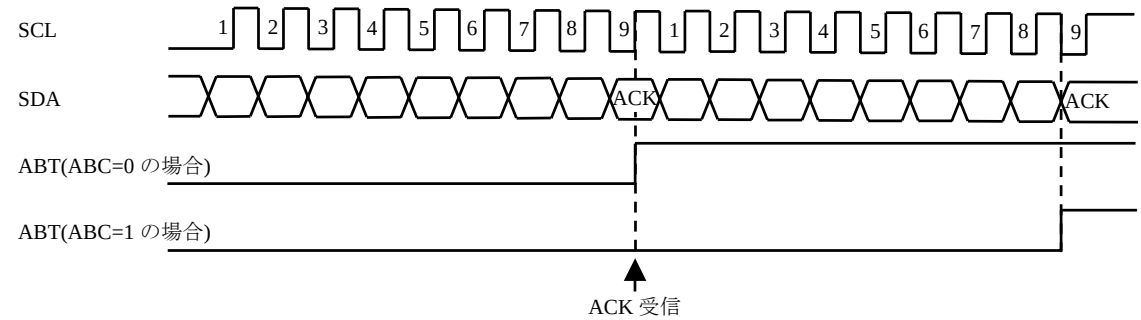



［タイミング図］

データ受信時




ACK 受信とき



### 1.5.2 アービトレーション・ロスト発生時の SDA 出力禁止機能

アービトレーション・ロストの発生を検知したマスタは、その時点で、SDA の出力をオフしなくてはなりません。

M16C/62Pの簡易I²Cは、アービトレーション・ロスト発生時に、ハードウェアで自動的にSDA出力をオフする機能を選択可能です。

本機能は、SDA 出力停止ビット[ALS]に、"1"を設定することで許可され、"0"を設定することで禁止されます。本機能により、SDA 出力がオフされた場合は、SDA 出力停止ビット[ALS]またはアービトレーション・ロスト検出フラグ[ABT]をクリアすることにより、解除されます。

ただし、本機能を許可している場合は、アクノリッジのタイミングでもアービトレーション・ロスト発生と判断して、出力をオフしてしまいますので、次のバイトデータの送出は、アービトレーション検出ビット[ABT]をクリアしてから行ってください。

また、アービトレーション検出フラグ制御[ABC]は、"0"に固定してください。

［関連レジスタ］

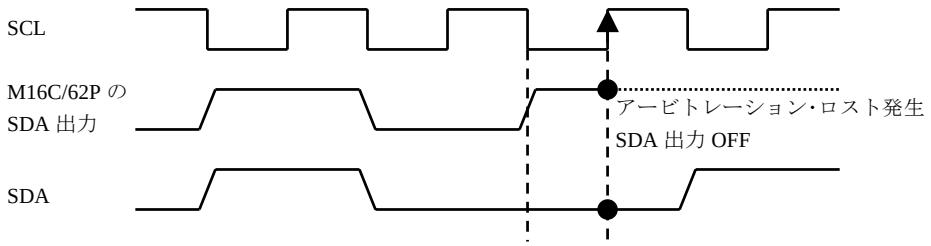
UARTi 特殊モードレジスタ 2 (i=0～2)

U0SMR2：$036E_{16}$番地　U1SMR2：$0372_{16}$番地　U2SMR2：$0376_{16}$番地

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| ☒ | | | | 1 | | | |

| | |
|---|---|
| [IICM2] | 0：アクノリッジ検出／アクノリッジ未検出割り込み有効<br>1：UARTi 送信／受信割り込み有効 |
| [CSC] | 0：クロック同期化禁止<br>1：クロック同期化許可 |
| [SWC] | 0：SCL ウエイト出力禁止<br>1：SCL クロック出力許可 |
| [ALS] | 1：SDA 出力停止許可 |
| [STAC] | 0：UARTi 初期化禁止<br>1：UARTi 初期化許可 |
| [SWC2] | 0：SCL は UARTi クロックを出力<br>1：SCL から"L"を出力（SCL 端子 L 出力機能有効） |
| [SDHI] | 0：SDA 出力許可<br>1：SDA 出力禁止（ハイインピーダンス） |

［タイミング図］



### 1.5.3 SCL 同期化機能

処理速度の遅いデバイスと接続している場合、他のデバイスが、SCL に対し L ホールドを行い、マスタから送出するクロックを強制的に待ち状態にすることがあります。

M16C/62PのI²Cでは、他デバイスからのLホールドに対し、自動的に待ち状態に入り、Lホールドの解除により待ち状態も解除するSCL同期化機能を持っています。

本機能は、クロック同期化ビット[CSC]を"1"に設定することにより動作が許可され、"0"を設定することにより禁止されます。

M16C/62P をマスタとして使用する場合（内部クロックモードで）のみ、本機能をご使用ください。

［関連レジスタ］

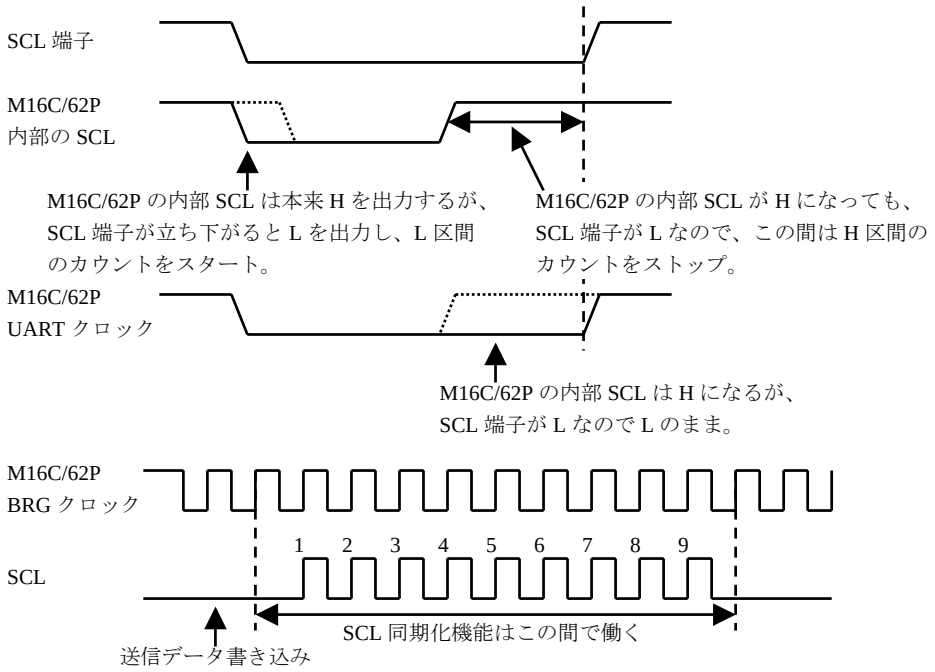
UARTi 特殊モードレジスタ 2 (i=0～2)

U0SMR2：036E$_{16}$番地 U1SMR2：0372$_{16}$番地 U2SMR2：0376$_{16}$番地

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| × | | | | | | 1 | |

| | |
|---|---|
| [IICM2] | 0：アクノリッジ検出／アクノリッジ未検出割り込み有効<br>1：UARTi 送信／受信割り込み有効 |
| [CSC] | 1：クロック同期化許可 |
| [SWC] | 0：SCL ウエイト出力禁止<br>1：SCL クロック出力許可 |
| [ALS] | 0：SDA 出力停止禁止<br>1：SDA 出力停止許可 |
| [STAC] | 0：UARTi 初期化禁止<br>1：UARTi 初期化許可 |
| [SWC2] | 0：SCL は UARTi クロックを出力<br>1：SCL から"L"を出力（SCL 端子 L 出力機能有効） |
| [SDHI] | 0：SDA 出力許可<br>1：SDA 出力禁止（ハイインピーダンス） |

［タイミング図］



## 1.6 その他の機能

M16C/62Pの簡易I²Cは、1.1～1.5 項で説明した他に、I²Cバス制御を容易にする機能として、以下の機能もハードウェアで備えています。

### 1.6.1 SDA 出力禁止機能

M16C/62P をスレーブとして使用し、スタートコンディション受信後の第 1 バイト目でアドレス判定を行う際、マスタによって指定されたアドレスが、自アドレスと異なれば M16C/62P は SDA の出力をオフ（ハイインピーダンス）にしなければなりません。

その場合、SCLの 9 クロック毎（受信割り込み要求が発生する毎）に、M16C/62Pの送信バッファレジスタに"1FF$_{16}$"を設定することにより、SDA出力をオフにします。また、M16C/62Pが備えているSDA出力禁止機能を使用することで、SDA出力をオフすることも可能です。

本機能は、SDA出力禁止ビット[SDHI]を"1"にすることで有効となり、送信バッファレジスタに"1FF$_{16}$"を設定しなくてもM16C/62PのSDA出力をハイインピーダンスにすることが可能です。[SDHI]を"0"にすることで、本機能は解除され、次のSCLの入力に同期して、送信バッファに設定した値が出力されます。

［関連レジスタ］

UARTi 特殊モードレジスタ 2 (i=0～2)

U0SMR2：036E$_{16}$番地　U1SMR2：0372$_{16}$番地　U2SMR2：0376$_{16}$番地

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 | | |
|---|---|---|---|---|---|---|---|---|---|
| × | 1 | | | | | | | [IICM2] | 0：アクノリッジ検出／アクノリッジ未検出割り込み有効<br>1：UARTi 送信／受信割り込み有効 |
| | | | | | | | | [CSC] | 0：クロック同期化禁止<br>1：クロック同期化許可 |
| | | | | | | | | [SWC] | 0：SCL ウエイト出力禁止<br>1：SCL クロック出力許可 |
| | | | | | | | | [ALS] | 0：SDA 出力停止禁止<br>1：SDA 出力停止許可 |
| | | | | | | | | [STAC] | 0：UARTi 初期化禁止<br>1：UARTi 初期化許可 |
| | | | | | | | | [SWC2] | 0：SCL は UARTi クロックを出力<br>1：SCL から"L"を出力（SCL 端子 L 出力機能有効） |
| | | | | | | | | [SDHI] | 1：SDA 出力禁止（ハイインピーダンス） |

### 1.6.2 UARTi 初期化機能(i=0～2)

M16C/62Pの簡易I²Cは、スタートコンディション検出のタイミングで自動的にUARTiを初期化する機能を備えています。本機能はM16C/62Pをスレーブとする際に使用します。

スタートコンディション初期化ビット[STAC]を"1"にすることにより機能が許可され、"0"にすることにより禁止されます。

スタートコンディションを検出すると、以下の初期化が行われます。

① 送信レジスタは初期化され、送信バッファレジスタの内容が送信レジスタに転送されます。
これにより、データ受信時に送信バッファレジスタにデータを再設定する必要はなく、次に入力されたクロックを 1 ビット目として送信が開始されます。
ただし、そのときの送信データは最後に送信していたデータと同じものとなりますので、SDA 出力禁止ビット[SDHI]を"1"として、送信データの出力を禁止してください。

② 受信レジスタは初期化され、次に入力されたクロックを 1 ビット目として、受信が開始されます。
受信バッファレジスタを読み出す前のタイミングで初期化され受信が開始しても、オーバランエラーは発生しません。

③ ウエイト出力ビット[SWC]が"1"にセットされます。これにより、SCL 端子 L 出力機能が有効となり、転送クロックの 9 ビット目の立ち下がりで、SCL 端子から L が出力されます。

本機能は外部クロック選択でのみ使用してください。また、本機能を使用し、UARTi の送受信を開始した場合、送信バッファ空フラグの値は変化しませんので、ご注意ください。

［関連レジスタ］

UARTi 特殊モードレジスタ 2 (i=0～2)

U0SMR2：$036E_{16}$番地 U1SMR2：$0372_{16}$番地 U2SMR2：$0376_{16}$番地

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| × | | | 1 | | | | |

| | |
|---|---|
| [IICM2] | 0：アクノリッジ検出／アクノリッジ未検出割り込み有効<br>1：UARTi 送信／受信割り込み有効 |
| [CSC] | 0：クロック同期化禁止<br>1：クロック同期化許可 |
| [SWC] | 0：SCL ウエイト出力禁止<br>1：SCL クロック出力許可 |
| [ALS] | 0：SDA 出力停止禁止<br>1：SDA 出力停止許可 |
| [STAC] | 1：UARTi 初期化許可 |
| [SWC2] | 0：SCL は UARTi クロックを出力<br>1：SCL から"L"を出力（SCL 端子 L 出力機能有効） |
| [SDHI] | 0：SDA 出力許可<br>1：SDA 出力禁止（ハイインピーダンス） |

### 1.6.3 SCL 出力停止機能

I²Cバスライン異常発生時に本機能を使用することで、SCLを解放することが可能となります。
本機能は、[SCLHI]を"1"に設定することにより、有効となります。

［関連レジスタ］

UARTi 特殊モードレジスタ (i=0～2)

U0SMR：$036F_{16}$番地　U1SMR：$0373_{16}$番地　U2SMR：$0377_{16}$番地

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | C | | 1 |

- [IICM] 1：I²Cモード
- [ABC] 0：アービトレーション・ロストをビット毎に更新 / 1：アービトレーション・ロストをバイト毎に更新
- [BBS] 0：バスは解放状態 / 1：バスは使用状態 （"0"のみ書き込み可）

UARTi 特殊モードレジスタ 4 (i=0～2)

U0SMR4：$036C_{16}$番地　U1SMR4：$0370_{16}$番地　U2SMR4：$0374_{16}$番地

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| | 1 | | | | | | |

- [STAREQ] 0：クリア / 1：スタート
- [RSTAREQ] 0：クリア / 1：スタート
- [STPREQ] 0：クリア / 1：スタート
- [STSPSEL] 0：スタート／ストップコンディションを出力しない / 1：スタート／ストップコンディションを出力する
- [ACKD] 0：ACK / 1：NACK
- [ACKC] 0：SI/O データ出力 / 1：ACK データビット(ACKD)出力
- [SCLHI] 1：許可
- [SWC9] 0：SCL"L"ホールド禁止 / 1：SCL"L"ホールド許可

### 1.6.4 SCL 端子 L 出力機能 3

I²Cバスでスレーブ送信を行う際、マスタは 9 ビット目に同期してアクノリッジの生成（または未生成）を行います。

このとき、スレーブは、アクノリッジ判定を行います。アクノリッジ検出時は、送信の継続（送信次データの設定）を行います。未検出時は送信を終了します。

本処理のための機能として、M16C/62P では「SCL 端子 L 出力機能 2」を持っています。この機能を用いると、初めの 9 ビット(ACK／NACK)のデータを受信後、最終ビットの SCL が"L"になるタイミングで、M16C/62P の SCL 端子に"L"を出力して、強制的にマスタを待ち状態にします。そして、ソフトウェアでのアクノリッジ判定処理が終了後に、送信継続処理、もしくは送信終了処理を行うことが可能です。

本機能は、ウエイト出力ビット 2[SWC9]を"1"にすることにより動作が許可され、"0"にすることにより禁止されます。

また、本機能で SCL 端子が"0"になったときは、[SWC9]を"0"にすることで解除できます。

［関連レジスタ］

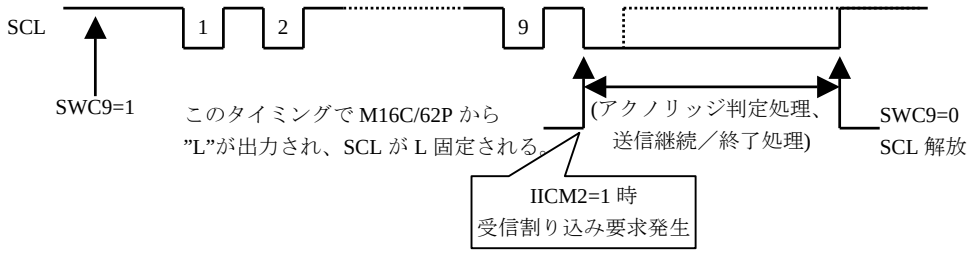
UARTi 特殊モードレジスタ 4 (i=0～2)

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| 1 | | | | | | | |

U0SMR4：$036C_{16}$番地　U1SMR4：$0370_{16}$番地　U2SMR4：$0374_{16}$番地

| | |
|---|---|
| [STAREQ] | 0：クリア<br>1：スタート |
| [RSTAREQ] | 0：クリア<br>1：スタート |
| [STPREQ] | 0：クリア<br>1：スタート |
| [STSPSEL] | 0：スタート／ストップコンディションを出力しない<br>1：スタート／ストップコンディションを出力する |
| [ACKD] | 0：ACK<br>1：NACK |
| [ACKC] | 0：SI/O データ出力<br>1：ACK データビット(ACKD)出力 |
| [SCLHI] | 0：禁止<br>1：許可 |
| [SWC9] | 1：SCL"L"ホールド許可 |

［タイミング図］



### 1.6.5 クロック遅延機能

I²CバスでM16C/62Pを使用する場合、クロック遅延機能を利用してUARTi(i=0～2)を使用します。

この機能により、SCL 初期値"L"、終了値"L"の波形出力が可能になり、各コンディションとのつなぎをスムーズに行うことが可能になります。また、SCL の 8bit 目の立ち下がりと 9bit の立ち上がりに、UiRB レジスタへの書き込みタイミングが発生するようになり、ACK／NACK ビットの受信が可能となります。さらに、IICM2="1"の場合、ACK／NACK ビットの受信後に送信割り込みが発生します。

本機能は、[IICM]を"1"に設定し、[CKPH]を"1"に設定することにより、有効となります。

［関連レジスタ］

UARTi 特殊モードレジスタ (i=0～2)

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | C | | 1 |

U0SMR：$036F_{16}$番地　U1SMR：$0373_{16}$番地　U2SMR：$0377_{16}$番地

[IICM]　1：I²Cモード

[ABC]　0：アービトレーション・ロストをビット毎に更新
1：アービトレーション・ロストをバイト毎に更新

[BBS]　0：バスは解放状態
1：バスは使用状態 （"0"のみ書き込み可）

UARTi 特殊モードレジスタ 3 (i=0～2)

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| | | | | 0 | | 1 | |

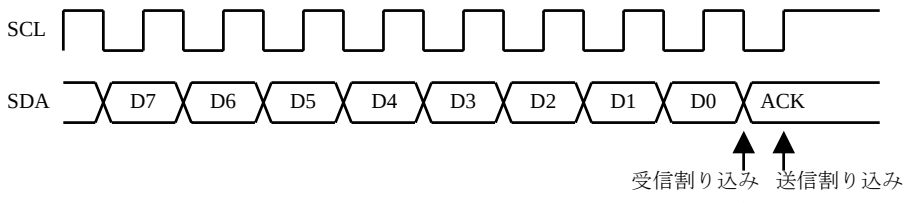
U0SMR3：$036D_{16}$番地　U1SMR3：$0371_{16}$番地　U2SMR3：$0375_{16}$番地

[CKPH]　1：クロック遅延あり

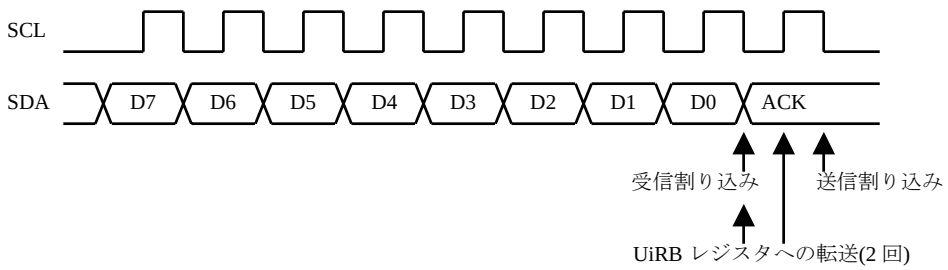
[DL]　デジタル遅延値を設定
000：遅延なし
001：BRG カウントソースの 2 サイクル
010：BRG カウントソースの 3 サイクル
011：BRG カウントソースの 4 サイクル
100：BRG カウントソースの 5 サイクル
101：BRG カウントソースの 6 サイクル
110：BRG カウントソースの 7 サイクル
111：BRG カウントソースの 8 サイクル

［タイミング図］

● CKPH="0"(クロック遅延なし)
IICM="1"(I²Cモード)
IICM2="1"(UART送受信割り込み)の場合




● CKPH="1"(クロック遅延あり)
IICM="1"(I²Cモード)
IICM2="1"(UART送受信割り込み)の場合



### 1.6.6 UART 出力デジタル遅延機能

I²Cバスで送信を行う場合、SCLが"L"区間のときに、データを切り替えする必要があります。SCLが"H"区間のとき、SDAが変化すると、各コンディションが誤検出されてしまいます。

M16C/62P では、UART 出力デジタル遅延機能を利用することにより、クロック"L"期間でのデータ変化を確定できます。

本機能は、[IICM]を 1 に設定し、[DL0～2]を"1"～"7"に設定することで、有効となります。

［関連レジスタ］

UARTi 特殊モードレジスタ (i=0～2)

U0SMR：$036F_{16}$番地　U1SMR：$0373_{16}$番地　U2SMR：$0377_{16}$番地

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | C | | 1 |

[IICM] 1：I²Cモード

[ABC] 0：アービトレーション・ロストをビット毎に更新
1：アービトレーション・ロストをバイト毎に更新

[BBS] 0：バスは解放状態
1：バスは使用状態 （"0"のみ書き込み可）

UARTi 特殊モードレジスタ 3 (i=0～2)

U0SMR3：$036D_{16}$番地　U1SMR3：$0371_{16}$番地　U2SMR3：$0375_{16}$番地

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| | | | | 0 | | 1 | |

[CKPH] 1：クロック遅延あり

[DL] デジタル遅延値を設定
000：遅延なし
001：BRG カウントソースの 2 サイクル
010：BRG カウントソースの 3 サイクル
011：BRG カウントソースの 4 サイクル
100：BRG カウントソースの 5 サイクル
101：BRG カウントソースの 6 サイクル
110：BRG カウントソースの 7 サイクル
111：BRG カウントソースの 8 サイクル

UARTi 送受信制御レジスタ0 (i=0～2)

U0C0：$03A4_{16}$番地　U1C0：$03AC_{16}$番地　U2C0：$037C_{16}$番地

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 1 | ● | | | |

[CLK] BRGカウントソースを選択
00：$f_{1SIO}$または$f_{2SIO}$を選択
01：$f_8$を選択
10：$f_{32}$を選択
11：使用禁止

[CRD] 1:CTS／RTS 機能禁止

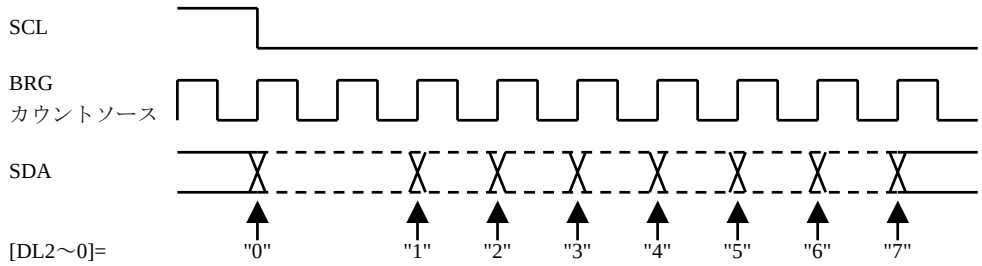
[NCH] 1：SCL、SDA 端子は Nch オープンドレイン出力(注 1)

[CKPOL] 0：CLK 極性は SCL 立ち下がりで送信データ出力、立ち上がりで入力

[UFORM] 1：転送フォーマット　MSB ファースト

注1　UART2 の SDA、SCL 端子は Nch オープンドレイン端子です。CMOS 出力は設定できません。
このとき、U2C0 の bit5 は"0"を設定してください。

［タイミング図］
各[DL2～0]設定時の SDA 出力切り替えタイミング



## 2. 簡易$I^2C$使用時の注意事項

本章では、M16C/62Pの簡易$I^2C$を使用し、$I^2C$バスのプロトコル制御を行う場合の注意事項、および制限事項を説明します。

### 2.1 電気的特性

M16C/62Pの電気的特性は、$I^2C$バス規格と一部異なる点があります。

#### 2.1.1 LOW レベル／HIGH レベル入力電圧

M16C/62P の電気的特性は、2.7V～5.5V 動作時において、

H入力電圧($V_{IH}$) = min.0.8$V_{CC}$（保証値）

L入力電圧($V_{IL}$) = max.0.2$V_{CC}$（保証値）

となります。

したがって、、$I^2C$バスの規格値

5V動作時：$V_{IH}$ = 3V、$V_{IL}$ = 1.5V

5V以外：$V_{IH}$ = 0.7$V_{CC}$、$V_{IL}$ = 0.3$V_{CC}$

とは異なります。

また、M16C/62Pの"L"出力電圧は、$V_{CC}$ = 5V、$I_{OL}$ = 5mA時に、

L出力電圧($V_{OL}$) = max.2.0V（保証値）

となります。

$I^2C$バスの規格値

L出力電圧($V_{OL}$) = max.0.6V（$I_{OL}$ = 6mA時）

とは異なります。

#### 2.1.2 スタート／ストップコンディションのセットアップタイム･ホールドタイム

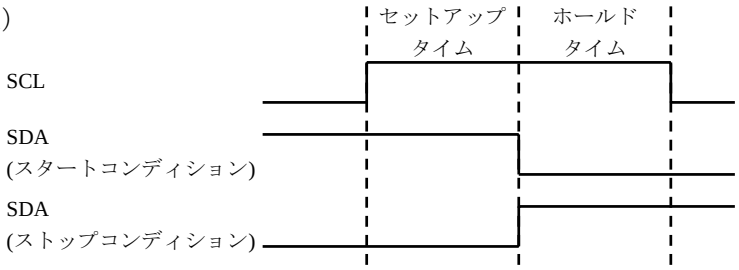
M16C/62Pのスタートコンディション／ストップコンディション検出時のセットアップタイムおよびホールドタイムは、$I^2C$バスの規格値と異なる場合があります。

（高速モード時）

M16C/62P のスタートコンディション／ストップコンディション検出時のセットアップタイムおよびホールドタイムは、以下の値となります。

セットアップ時間 ＞ 3～6 サイクル（注 1）

ホールド時間 ＞ 3～6 サイクル（注 1）




注1　サイクル数はメインクロック入力発振周波数f($X_{IN}$)のサイクル数を示します。
高速モード$I^2C$バスの規格において、スタートコンディションおよびストップコンディションのセットアップ／ホールドタイムはどちらもMin.600nsです。
それに対し、M16C/62Pのセットアップ／ホールドタイムはMin.6 サイクル（f($X_{IN}$)のサイクル数）となります。
したがって、メインクロックf($X_{IN}$)を 10MHzで使用した場合は、M16C/62Pの簡易$I^2C$のセットアップ／ホールドタイムは、Min.600nsとなり高速モード$I^2C$バス規格にも対応できますが、メインクロックを 10MHz未満で使用する場合は、セットアップ／ホールドタイムは、高速モード$I^2C$バス規格を満たすことができなくなります。

## 2.2 BRG カウントソースによる最大転送速度の制限

M16C/62PがSCLのレベルを認識するまでの時間はサンプリング周期に依存し、最大BRGカウントソースの3 クロック分を要します。したがって、動作周波数およびBRGカウントソース設定ビットで設定したBRGカウントソースの速度によって、M16C/62Pの簡易I²Cに接続可能なI²Cバスの最大転送速度が制限されます。以下の条件を満たす転送速度で使用しない場合、ビットずれを起こす可能性があります。

I²Cバスの最大転送速度(Hz) ＜ **BRGカウントソース(Hz)** / 3

（例） 源発振 10MHz、BRGカウントソースに$f_{32SIO}$を選択した場合
I²Cバスの最大転送速度(Hz) ＜ **10MHz / 32** / 3 = 104kbps
この場合のI²Cバスの最大転送速度は、104kbpsになります。

## 2.3 最大動作周波数

最大動作周波数は、M16C/62Pの最大動作周波数$f(X_{IN})$maxに依存します。

SCLのdutyが 50%であるため高速モード（400kbps）に設定したとき、SCLのL幅が 1.25usとなります。この値はI²Cの規格（$f_{LOW}$＝min1.3us）を満たしません。

また、クロック同期化機能を有効にするとカウントソースの 1～1.5 サイクル、SCL の H 幅が延びます。したがって、高速モードにおける最大転送速度は約 350kbps となります。

## 2.4 制限事項

### 2.4.1 生成するスタートコンディションに関する制限事項

- 現象
  マルチマスタ通信時にスタートコンディションが衝突した場合に、生成するスタートコンディションのホールド時間（SDAの立下りからSCLの立下り）が、I²Cの仕様(min:4us)を満たさない場合があります。min値はBCLKの 8 サイクルです。
- 問題点
  スレーブがスタートコンディションを検出できない可能性があります。
- 回避策／対策
  スレーブがスタートコンディションを検出できない場合は、NACK を返すため、バスを解放し、通信を終了します。
  送信リトライを行ってください。

### 2.4.2 ストップコンディションとスタートコンディションの間のバスフリー時間に関する制限事項

- 現象
  ストップコンディション送受信後、ストップコンディション割り込み終了まで次のスタートコンディションを受信できなくなります。
- 問題点
  スタートコンディションを検出できない可能性があります。
- 回避策／対策
  マスタとなるデバイスは、本プログラムがストップコンディション検出割り込みの処理を終了した後に、スタートコンディションを生成してください。

### 2.4.3 リスタートコンディションに関する制限事項

- 現象
  本サンプルプログラムは、リスタートコンディションに対応していません。
- 問題点
  リスタートコンディションで始まる通信を検出できません。
- 回避策／対策
  リスタートコンディションを使用しないでください。

# 3. サンプルプログラム



本プログラムは本サンプルプログラムです。通信動作を保証するものではありません。
システムへの組み込みの際には、十分検討の上ご使用ください。
また、本ソフトウェア単体ではシステムとしての評価ができないため、最終システムでの評価を実施してください。

本ソフトウェアはM16C/62Pグループに内蔵された簡易I²CのH/Wを制御し、I²Cバスの通信プロトコルを実現するものです。下記使用条件によりI²Cバスの通信プロトコルに準拠します。

＜動作条件＞
- 発信周波数：10MHz（ノンウェイト、分周なし）
  ※ 10MHz以外で使用される場合は、本アプリケーションノートを参考にUiBRG及び、DL2～0(UiSMR3)の値を選定してください。
  1.1 バイトデータの送信／受信の方法、1.6.5 その他の機能「クロック遅延機能」、2. 簡易I²C使用時の注意事項　参照

＜使用上の制限事項＞
- 7bit アドレス装置間の通信のみ対応しています。
- 特別なアドレス（ジェネラルコール等のアドレス）は使用できません。
- C-BUS、M3L-BUS等のI²Cバス互換プロトコルとの混在には対応していません。
- リスタート条件を用いて、スレーブの切り替えを行う通信フォーマットをバス上に流さないでください。(通信に異常をきたします。)
- バス衝突割り込みベクタとタイマB割り込みベクタが共用の為、I²Cバスモードでは、UART0 とタイマB3 を同時に使用できません。また同様にUART1 とタイマB4 を同時使用できません。
  ※ 割り込み要因の切り替えは、本アプリケーションノートを参考に、IFSR26～27(IFSR2A)の値を選定してください。

## 3.1 機能説明

### 3.1.1 アドレス

＜マスタ装置＞
・ 7bit アドレスを持つスレーブ装置との送受信
＜スレーブ装置＞
・ 7bit アドレスを持っています。
注） 特別なアドレス（ジェネラルコール等のアドレス）に対する送受信は対応しておりません。

### 3.1.2 転送速度

転送速度は、0～100kbps です。よって、高速モードのマスタとの通信はできません。

### 3.1.3 転送データ長

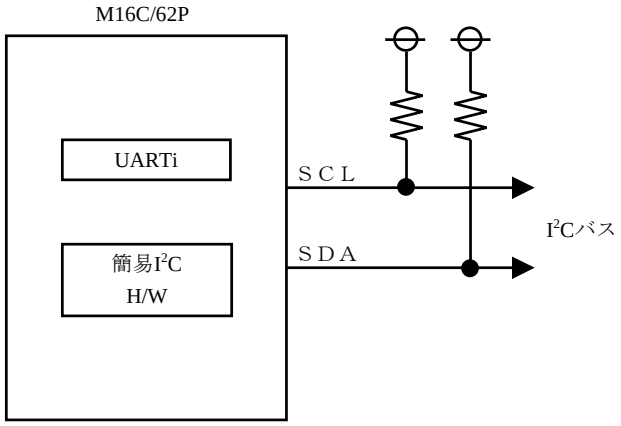
＜マスタ装置＞
・ 1～256 バイトのデータ送受信が可能です。
＜スレーブ装置＞
・ iic0_slave_end()の関数の引数である iic_index に与えられます。その範囲は 1～256 です。

### 3.1.4 マルチマスタ

I²Cバスに接続している、複数の装置が他の装置へのデータ伝送を行うことができます。

## 3.2 ハードウェア説明

I²Cバスプロトコルを実現するために、M16C/62PのUARTi(i=0～2)+内蔵簡易I²C H/Wのみを使用します。プルアップ抵抗は、システムに合わせて適切に選定してください。



## 3.3 使用方法

### 3.3.1 組み込み方法

① 使用する UART ch に従って、［intprg.c］内の下記割り込みベクタ定義をコメントアウトしてください。

・ UART0 を使用する場合

```
// TIMER B3,UART0 bus collision          (software int 7)
#pragma interrupt _timer_b3(vect=7)
void _timer_b3(void){}

// uart0 trance/NACK                     (software int 17)
#pragma interrupt _uart0_trance(vect=17)
void _uart0_trance(void){}

// uart0 receive/ACK                     (software int 18)
#pragma interrupt _uart0_receive(vect=18)
void _uart0_receive(void){}
```

・ UART1 を使用する場合

```
// TIMER B4,UART1 bus collision          (software int 6)
#pragma interrupt _timer_b4(vect=6)
void _timer_b4(void){}

// uart1 trance/NACK                     (software int 19)
#pragma interrupt _uart1_trance(vect=19)
void _uart1_trance(void){}

// uart1 receive/ACK                     (software int 20)
#pragma interrupt _uart1_receive(vect=20)
void _uart1_receive(void){}
```

・ UART2 を使用する場合

```
// UART2 Bus Collision                   (software int 10)
#pragma interrupt _uart2_bus_collision(vect=10)
void _uart2_bus_collision(void){}

// uart2 trance/NACK2                    (software int 15)
#pragma interrupt _uart2_trance(vect=15)
void _uart2_trance(void){}

// uart2 receive/ACK2                    (software int 16)
//#pragma interrupt          _uart2_receive(vect=16)
//void _uart2_receive(void){}
```

② 本プログラムを動作させる前に割り込み許可”fset I”を実行してください。

［アクセス禁止レジスタ］

以下に示すレジスタを変更しないでください。
(i=0～2　ただし、簡易I²Cとして使用しているUARTのレジスタに限る)

| レジスタ名 | ビット | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| UARTi バス衝突検出割り込み制御レジスタ | × | × | × | × | × | × | × | × |
| UARTi 送信割り込み制御レジスタ | × | × | × | × | × | × | × | × |
| UARTi 受信割り込み制御レジスタ | × | × | × | × | × | × | × | × |
| UARTi 特殊モードレジスタ | × | × | × | × | × | × | × | × |
| UARTi 特殊モードレジスタ 2 | × | × | × | × | × | × | × | × |
| UARTi 特殊モードレジスタ 3 | × | × | × | × | × | × | × | × |
| UARTi 特殊モードレジスタ 4 | × | × | × | × | × | × | × | × |
| UARTi 送受信モードレジスタ | × | × | × | × | × | × | × | × |
| UARTi 転送速度レジスタ | × | × | × | × | × | × | × | × |
| UARTi 送信バッファレジスタ | × | × | × | × | × | × | × | × |
| UARTi 送受信制御レジスタ 0 | × | × | × | × | × | × | × | × |
| UARTi 送受信制御レジスタ 1 | × | × | × | × | × | × | × | × |
| UARTi 受信バッファ | × | × | × | × | × | × | × | × |
| ポート 6 | ×注 1 | ×注 1 | ○ | ○ | ×注 1 | ×注 1 | ○ | ○ |
| ポート 6 方向レジスタ | ×注 1 | ×注 1 | ○ | ○ | ×注 1 | ×注 1 | ○ | ○ |
| ポート 7 | ○ | ○ | ○ | ○ | ○ | ○ | ×注 1 | ×注 1 |
| ポート 7 方向レジスタ | ○ | ○ | ○ | ○ | ○ | ○ | ×注 1 | ×注 1 |
| 外部割り込み要因選択レジスタ | ×注 2 | ×注 2 | ×注 2 | ×注 2 | ×注 2 | ×注 2 | ×注 2 | ×注 2 |

注1　簡易I²Cとして使用するポートのポートレジスタはライト禁止
注2　使用する外部割り込み要因選択レジスタはライト禁止

## 3.4 関数

### 3.4.1 初期化関数

| 関 数 名 | | | |
|---|---|---|---|
| char iic_ini(char SWITCH); | | | |
| 引 数 | | | |
| char | SWITCH | ; | 0：簡易I²C無効 |
| | | ; | 1：簡易I²C有効 |
| 戻 り 値 | | | |
| 0 | | ; | 失敗 |
| 1 | | ; | 成功 |
| 機 能 | | | |
| ・ I²Cバスで送受信するための初期化処理を行います。この処理が終了し、割り込み許可の状態であればスレーブ装置として動作します。<br>・ また、以下に述べるマスタ送受信を開始する関数をコールすることで、マスタ装置として動作するようになります。 | | | |
| 備 考 | | | |
| ・ 簡易I²C無効時は、次から示す関数を使用することはできません。 | | | |

### 3.4.2 Master 制御開始関数

| 関 数 名 | | | |
|---|---|---|---|
| char iic_master_start(char SLAVE, char RW, char *BUF, char LEN); | | | |
| 引 数 | | | |
| char | SLAVE | ; | 0x00～0x7F：指定するスレーブ装置アドレス |
| char | RW | ; | 0：マスタ送信動作 |
| | | ; | 1：マスタ受信動作 |
| char* | BUF | ; | 送信バッファ or 受信バッファへの pointer |
| char | LEN | ; | 0x00～0xFF：通信データ長 |
| 戻 り 値 | | | |
| 0 | | ; | Master 制御開始失敗 |
| 1 | | ; | Master 制御開始成功 |
| 機 能 | | | |
| ・ マスタ制御開始を開始します。<br>・ この関数を使用するには、iic_ini()により簡易I²Cを使用できる状態にしておく必要があります。 | | | |
| 備 考 | | | |
| ・ Master 制御開始関数の先頭で”fclr I”を実行し、割り込みを禁止しています。<br>これは、バスビジー判定後にスタートコンディションをなるべく早く出力するための処理です。 | | | |

### 3.4.3 Master EEPROM ランダムリード開始関数

| 関 数 名 | | | |
|---|---|---|---|
| char iic_master_randomread(char SLAVE, char ROM_ADR, char *BUF, char LEN); | | | |
| 引 数 | | | |
| char | SLAVE | ; | 0x00～0x7F：指定する EEPROM 装置のアドレス |
| char | ROM_ADR | ; | 読み込み対象の EEPROM 内のアドレス |
| char* | BUF | ; | 受信バッファへの pointer |
| char | LEN | ; | 0x00～0xFF：受信データ数 |
| 戻 り 値 | | | |
| 0 | | ; | Master 制御開始失敗 |
| 1 | | ; | Master 制御開始成功 |
| 機 能 | | | |
| ・ EEPROM に対するランダムリードを開始します。<br>・ この関数を使用するには、iic_ini()により簡易I²Cを使用できる状態にしておく必要があります。 | | | |
| 備 考 | | | |
| なし | | | |

## 3.5 ユーザが作成する関数

簡易I²Cでは、送受信のステータスと、そのデータ数を引数とする以下の関数を呼び出します。
この関数は、簡易I²Cに対してユーザが提供しなければなりません。

### 3.5.1 Master 制御完了関数

| | | | | |
|---|---|---|---|---|
| 関 数 名 | | | | |
| void iic_master_end(char STATUS); | | | | |
| 引 数 | | | | |
| char | STATUS | ; | 上位 4bit | 0：マスタ送信 |
| | | ; | | 1：マスタ受信 |
| | | ; | | 2：EEPROM ランダムリード |
| | | ; | 下位 4bit | 0：正常終了 |
| | | ; | | 2：バス競合負け |
| | | ; | | 3：NACK 終了 |
| 戻 り 値 | | | | |
| なし | | | | |
| 機 能 | | | | |
| ・ Master 制御完了後 F/W がコールする関数です。<br>・ マスタ通信の終了状態を上記の引数によりユーザに知らせます。 | | | | |
| 備 考 | | | | |
| ・ 簡易I²Cの割り込み処理の中からコールしています。 | | | | |

### 3.5.2 Slave check 関数

| | | | |
|---|---|---|---|
| 関 数 名 | | | |
| *char iic_id_chk(char ID, char RW); | | | |
| 引 数 | | | |
| char | ID | ; | 0x00～0x7F：マスタが指定しているスレーブ装置アドレス |
| char | RW | ; | 0：マスタは受信を要求(スレーブ受信する) |
| | | ; | 1：マスタは送信を要求(スレーブ送信する) |
| 戻 り 値 | | | |
| NULL pointer | | ; | スレーブ指定拒否 |
| pointer | | ; | 送信バッファ or 受信バッファへの Pointer |
| 機 能 | | | |
| ・ 1st Byte 受信後 F/W がコールする関数です。マスタからのスレーブへの要求内容を次に示す引数によりユーザに知らせます。戻り値に NULL ポインタを返すとスレーブ指定を拒否し、通信バッファのポインタを返せば、スレーブ動作を開始します。 | | | |
| 備 考 | | | |
| なし | | | |

### 3.5.3 Slave 制御完了関数

| 関 数 名 | | | | |
|---|---|---|---|---|
| void iic_slave_end(char STATUS, char IIC_INDEX); | | | | |
| 引 数 | | | | |
| char | STATUS | ; | 上位 4bit | 0：スレーブ受信 |
| | | ; | | 1：スレーブ送信 |
| | | ; | 下位 4bit | 0：正常終了 |
| char | IIC_INDEX | ; | 0x00～0xFF：受信データ数 | |
| 戻 り 値 | | | | |
| なし | | | | |
| 機 能 | | | | |
| ・ Slave 制御完了後 F/W がコールする関数。<br>・ スレーブ通信の終了状態を上記の引数によりユーザに知らせます。 | | | | |
| 備 考 | | | | |
| なし | | | | |

## 3.6 通信方法

### 3.6.1 準備

I²Cバスの通信を行うためには、以下の例のようにイニシャルルーチンなどのプログラム動作初期段階でiic_ini()をコールする必要があります。
iic_ini()をコールするときには、1 つの引数を渡します。
引数は、簡易I²Cの有効／無効を表します。

例)

```
(システム初期化処理)
iic_ini(1);                                          // 簡易I²C初期化
asm("fset I");                                       // Iフラグセット
```

### 3.6.2 マスタ通信

マスタ通信を開始するには、iic_master_start()をコールします。
iic_master_start()をコールするときには、4 つの引数を渡します。
第 1 引数は、送信相手のアドレスを指定します。
第 2 引数は、マスタ通信の送受信の指定を行います。0 なら送信を、1 なら受信を行います。
第 3 引数は、データ格納先先頭アドレスを指定します。このデータ格納先は、マスタ通信が終了するまでに解放されなければ near 属性の RAM 領域のどこにでも配置できます。
第 4 引数は、送信データ長を指定します。0 を指定すると最大送信データ長である、256 バイトを送信します。
また、iic_master_start()は戻り値をもっています。マスタ通信が開始された場合 0 を、開始されなかった場合 1 を返します。

以下の例では、$55_{16}$というアドレスを持つスレーブ装置にiic_ramから 5 バイトのデータを送信します。

例)

```
unsigned char      iic_ram[256];                     // 送信バッファ

if (iic_master_start(0x55, 0, &iic_ram[0], 5) != 0) {
        // マスタ通信失敗確認処理
} else {
        // マスタ通信開始確認処理
}
```

マスタ通信が終了すると、簡易I²Cはiic_master_end()をコールします。この関数は、ユーザが作成する必要があります。
iic_master_end()をコールするときには、1 つの引数を渡します。
引数は、マスタ送信終了ステータスを示しています。
ステータスの内容は 3.5.1 項に示してあります。

次に iic_master_end()の例を示します。

例)

```
// プロトタイプ
void iic_master_end(char);

// マスタ通信終了関数
void iic_master_end(char status) {
        if ((status & 0xF0) == 0x10) {                          // 送信モード
                switch (status & 0x0F){                         // 送信ステータス処理
                case 0:                 // 正常終了
                        break;
                case 1:                 // 第 1Byte に NACK
                        break;
                case 2:                 // 第 2Byte 以降に NACK
                        break;
                case 3:                 // BUS 競合負け
                        break;
                case 4:                 // 不正スタート条件
                        break;
                case 5:                 // 不正ストップ条件
                        break;
                default:
                        break;
                }
        } else if ((status & 0xF0) == 0x20) {                   // 受信モード
                // 送信ステータス処理時と同様に
                // 受信ステータス処理を行ってください。
        } else if ((status & 0xF0) == 0x30) {                   // EEPROM モード
                // 送信ステータス処理時と同様に
                // EEPROM モードステータス処理を行ってください。
        }
}
```

### 3.6.3 スレーブ通信

マスタから 1byte目のデータを受信したとき、簡易I²Cは関数iic_id_chk()をコールします。
この関数は、ユーザが作成する必要があります。
簡易I²Cは、マスタからのスレーブへの要求内容を次に示す引数に渡します。
第 1 引数はマスタが指定しているスレーブ装置のアドレスを示します。
第 2 引数はマスタの要求している通信内容を示します。
戻り値に NULL ポインタを返すとスレーブ指定を拒否し、通信バッファのポインタを返せば、スレーブ動作を開始します。

例)

```
// プロトタイプ
*char iic_id_chk(char, char);

// スレーブチェック関数
unsigned char     sw_buf[256];                    // 送信バッファ
unsigned char     sr_buf[256];                    // 受信バッファ

*char iic_id_chk(char ID, char RW){
        if (ID == 0x55) {                         // 自装置 ID が 0x55 の時
                if (RW == 1) {
                        return(&sw_buf[0]);       // スレーブ送信
                } else {
                        return(&sr_buf[0]);       // スレーブ受信
                }
        } else {
                return(0)                         // スレーブ操作無し
        }
}
```

スレーブ通信が終了すると、簡易I²Cはiic_slave_end()をコールします。この関数は、ユーザが作成する必要があります。
iic_slave_end()をコールするときには、２つの引数渡します。
第 1 引数はスレーブ通信のステータスを示します。
第 2 引数はスレーブ受信のデータ数を示します。
ステータスの内容は 3.5.3 項に示してあります。

次に iic_slave_end()の例を示します。
例)

```
// プロトタイプ
void iic_slave_end(char, char);

// スレーブチェック関数
void iic_slave_end(char status,char iic_index){
        if ((status & 0xF0) == 0x10) {                          // 送信モード
                switch (status & 0x0F) {                        // 送信ステータス処理
                case 0:                 // 正常終了
                        break;
                case 1:                 // 第 1Byte に NACK
                        break;
                case 2:                 // 第 2Byte 以降に NACK
                        break;
                case 3:                 // BUS 競合負け
                        break;
                case 4:                 // 不正スタート条件
                        break;
                case 5:                 // 不正ストップ条件
                        break;
                default:
                        break;
                }
        } else {                                                // 受信モード
                switch (status & 0x0F) {                        // 受信ステータス処理
                case 0:                 // 正常終了(iic_index 分のデータ受信あり)
                        break;
                case 1:                 // 第 1Byte に NACK
                        break;
                case 2:                 // 第 2Byte 以降に NACK
                        break;
                case 3:                 // BUS 競合負け
                        break;
                case 4:                 // 不正スタート条件
                        break;
                case 5:                 // 不正ストップ条件
                        break;
                default:
                        break;
                }
        }
}
```

## 4. 参考ドキュメント

データシート
M16C/62P グループデータシート　Rev.2.41
（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。）

ハードウェアマニュアル
M16C/62P グループハードウェアマニュアル　Rev.2.41
（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/

ルネサス製品全般に関するお問合せ先
カスタマ・サポート・センター：csc@renesas.com

## 改訂記録

| Rev. | 発行日 | 改訂内容 | |
|---|---|---|---|
| | | ページ | ポイント |
| 1.02 | 2007/07/23 | — | 初版発行 |

## 本資料ご利用に際しての留意事項

1. 本資料は、お客様に用途に応じた適切な弊社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弊社または第三者の知的財産権その他の権利の実施、使用を許諾または保証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など全ての情報の使用に起因する損害、第三者の知的財産権その他の権利に対する侵害に関し、弊社は責任を負いません。
3. 本資料に記載の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、あるいはその他軍事用途の目的で使用しないでください。また、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、それらの定めるところにより必要な手続を行ってください。
4. 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などの全ての情報は本資料発行時点のものであり、弊社は本資料に記載した製品または仕様等を予告なしに変更することがあります。弊社の半導体製品のご購入およびご使用に当たりましては、事前に弊社営業窓口で最新の情報をご確認頂きますとともに、弊社ホームページ（http://www.renesas.com）などを通じて公開される情報に常にご注意下さい。
5. 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が、万一本資料の記述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、弊社はその責任を負いません。
6. 本資料に記載の製品データ、図、表などに示す技術的な内容、プログラム、アルゴリズムその他応用回路例などの情報を流用する場合は、流用する情報を単独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下さい。弊社は、適用可否に対する責任は負いません。
7. 本資料に記載された製品は、各種安全装置や運輸・交通用、医療用、燃焼制御用、航空宇宙用、原子力、海底中継用の機器・システムなど、その故障や誤動作が直接人命を脅かしあるいは人体に危害を及ぼすおそれのあるような機器・システムや特に高度な品質・信頼性が要求される機器・システムでの使用を意図して設計、製造されたものではありません（弊社が自動車用と指定する製品を自動車に使用する場合を除きます）。これらの用途に利用されることをご検討の際には、必ず事前に弊社営業窓口へご照会下さい。なお、上記用途に使用されたことにより発生した損害等について弊社はその責任を負いかねますのでご了承願います。
8. 第7項にかかわらず、本資料に記載された製品は、下記の用途には使用しないで下さい。これらの用途に使用されたことにより発生した損害等につきましては、弊社は一切の責任を負いません。
   1）生命維持装置。
   2）人体に埋め込み使用するもの。
   3）治療行為（患部切り出し、薬剤投与等）を行なうもの。
   4）その他、直接人命に影響を与えるもの。
9. 本資料に記載された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、実装条件およびその他諸条件につきましては、弊社保証範囲内でご使用ください。弊社保証値を越えて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、弊社はその責任を負いません。
10. 弊社は製品の品質及および信頼性の向上に努めておりますが、特に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。弊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、誤動作防止設計などの安全設計（含むハードウエアおよびソフトウエア）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エアは、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致します。
11. 本資料に記載の製品は、これを搭載した製品から剥がれた場合、幼児が口に入れて誤飲する等の事故の危険性があります。お客様の製品への実装後に容易に本製品が剥がれることがなきよう、お客様の責任において十分な安全設計をお願いします。お客様の製品から剥がれた場合の事故につきましては、弊社はその責任を負いません。
12. 本資料の全部または一部を弊社の文書による事前の承諾なしに転載または複製することを固くお断り致します。
13. 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弊社営業窓口までご照会下さい。