

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、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/6N、M16C/1N、M16C/29、R8C/22,23 グループ

## CAN アプリケーションノート

### 1. 要約

この資料は、M16C/6N、M16C/1N、M16C/29、R8C/22,23 グループを使用して CAN 通信を行う場合の手順を説明しています。

### 2. 適用

この資料は、M16C/6N、M16C/1N、M16C/29、R8C/22,23 グループ(以下 6N、1N、29、R8C/22,23 という)に適用されます。

6N グループは 6N4、6N5、6NK、6NL、6NM、6NN グループのサブグループに分かれます。6N5、6NL、6NN、1N、29、R8C/22,23 は CAN0 のみで CAN1 に関する記述は適用されません。

<b>1. 要約</b> .....	<b>1</b>
<b>2. 適用</b> .....	<b>1</b>
<b>3. 初期設定</b> .....	<b>2</b>
3.1 CAN ビットタイミング .....	3
3.1.1 ビットタイミングの条件 .....	4
3.2 同期の取り方 .....	5
3.3 転送速度 .....	7
3.4 CAN ビットタイミングと転送速度の設定 .....	9
<b>4. CAN メッセージの送受信</b> .....	<b>10</b>
4.1 CAN コンフィグレーション .....	11
4.2 メッセージ送信.....	12
4.2.1 データフレーム送信モード.....	13
4.2.2 リモートフレーム受信/データフレーム送信モード.....	17
4.2.3 送信アボート .....	20
4.3 メッセージ受信.....	25
4.3.1 データフレーム受信モード.....	26
4.3.2 リモートフレーム送信/データフレーム受信モード.....	32
4.4 CAN オーバーランエラー .....	33
4.5 Basic CAN モード.....	34
<b>5. CAN エラー</b> .....	<b>36</b>
5.1 CAN エラー確認手順.....	37
5.2 リターンフロムバスオフ機能.....	39
<b>6. アクセプタンスフィルタの使い方</b> .....	<b>40</b>
6.1 アクセプタンスフィルタ(ACP) .....	40
6.2 アクセプタンスフィルタサポートユニット(ASU) .....	43
6.2.1 アクセプタンスフィルタサポートユニットの使用法 .....	43
<b>7. CAN スリープ動作および CAN ウェイクアップ動作</b> .....	<b>47</b>
7.1 CAN のスリープ動作 .....	47
7.2 CAN ウェイクアップ動作.....	48
<b>8. サンプルプログラムに関する注意事項</b> .....	<b>49</b>
8.1 各レジスタのシンボル表記 .....	49
8.2 while の無限ループ .....	49
<b>9. ホームページとサポート窓口</b> .....	<b>50</b>

### 3. 初期設定

CAN 通信を行う場合、以下の設定が必要です。

- CAN 動作モードの設定
- ボーレートの設定
- サンプリング回数の設定(サンプリング回数は 1 回に設定)
- ビットタイミングの設定
- アクセプタンスフィルタの設定

また、システムに応じて以下のプログラムを作成してください。

- CAN 送信完了割り込みによる送信完了処理プログラム
- ポーリングによる送信完了処理プログラム
- CAN 受信完了割り込みによる受信完了処理プログラム
- ポーリングによる受信完了処理プログラム
- CAN エラー割り込みプログラム
- CAN ウェイクアップ割り込みプログラム

### 3.1 CAN ビットタイミング

CAN プロトコルでは、通信フレームの各ビットを4つのセグメントで構成しています。

図 1にビットのセグメント構成とサンプルポイントを示します。

これらのセグメントのうち、Propagation Time Segment(以下 PTS という)、Phase Buffer Segment 1(以下 PBS1 という)、Phase Buffer Segment 2(以下 PBS2 という)は、サンプルポイントを指定するもので、これらの値を変えることでサンプリングするタイミングを変えることができます。

このタイミング設定の最小単位を 1 Time Quanta(以下 Tq という)といい、CAN モジュールに入力されるクロック周波数とボーレートプリスケアラ分周値で決められます。

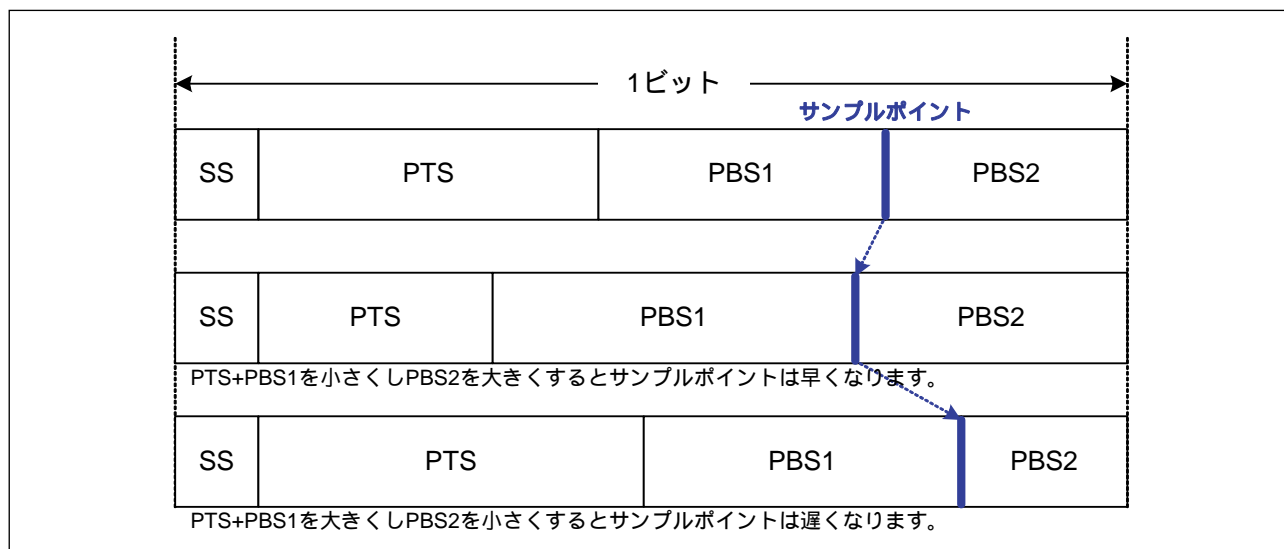


図 1. ビットのセグメント構成とサンプルポイント

- (1) SS : シンクロナイゼーションセグメント  
インターフレームスペース\*1 中に、レセシブからドミナントへのエッジをモニタして同期をとるセグメントです。
- (2) PTS : プロパゲーションタイムセグメント  
CAN ネットワーク上の物理的な遅延を吸収するセグメントです。ネットワーク上の物理的な遅延は、バスによる遅延、入力コンパレータによる遅延、出力ドライバ遅延の総和の 2 倍です。
- (3) PBS1、PBS2 : フェーズバッファセグメント 1、フェーズバッファセグメント 2  
再同期の際におきるフェーズエラー\*2 を補償するためのセグメントです。
- (4) SJW : リシンクロナイゼーションジャンプ幅  
フェーズエラーによる同期ずれを補償する最大幅です。

\*1: インターフレームスペース(Interframe Space)

インターミッション(Intermission)、サスペンドトランスミッション(Suspend Transmission)、バスアイドル(Bus Idle)で構成されます。バスアイドル中では、全ノードが送信を開始することができます。

\*2: フェーズエラー(Phase Error)

発振器周波数のずれなどで生じる同期ずれです。詳しくは3.2項(2)を参照ください。

### 3.1.1 ビットタイミングの条件

各セグメントの設定と制限事項は以下のとおりです。

(1) 各セグメントの設定

- $SS = 1Tq$  固定
- $PTS = 1 \sim 8Tq$  の範囲で設定
- $PBS1 = 2 \sim 8Tq$  の範囲で設定
- $PBS2 = 2 \sim 8Tq$  の範囲で設定
- $SJW = 1 \sim 4Tq$  の範囲で設定
- $SS+PTS+PBS1+PBS2 = 8 \sim 25Tq$

(2) PBS1、PBS2 の制限

- $PBS1 \leq PBS2$
- $PBS1 \leq SJW$
- $SJW=1$  の時、 $PBS2 \leq 2$
- $2 \leq SJW \leq 4$  の時、 $PBS2 \leq SJW$

### 3.2 同期の取り方

CAN プロトコルの通信方式は NRZ(Non-Return to Zero)方式です。各ビットの始まりや終わりに同期信号を付加しません。

- (1) ハードウェア同期(メッセージの送受信を行っていない時の同期)  
インターフレームスペース中で、レセシブからドミナントへのエッジを検出すると、その時点ビットの始まり(SS)と認識し同期を取ります。これをハードウェア同期といいます。

図 2にハードウェア同期の仕組みを示します。

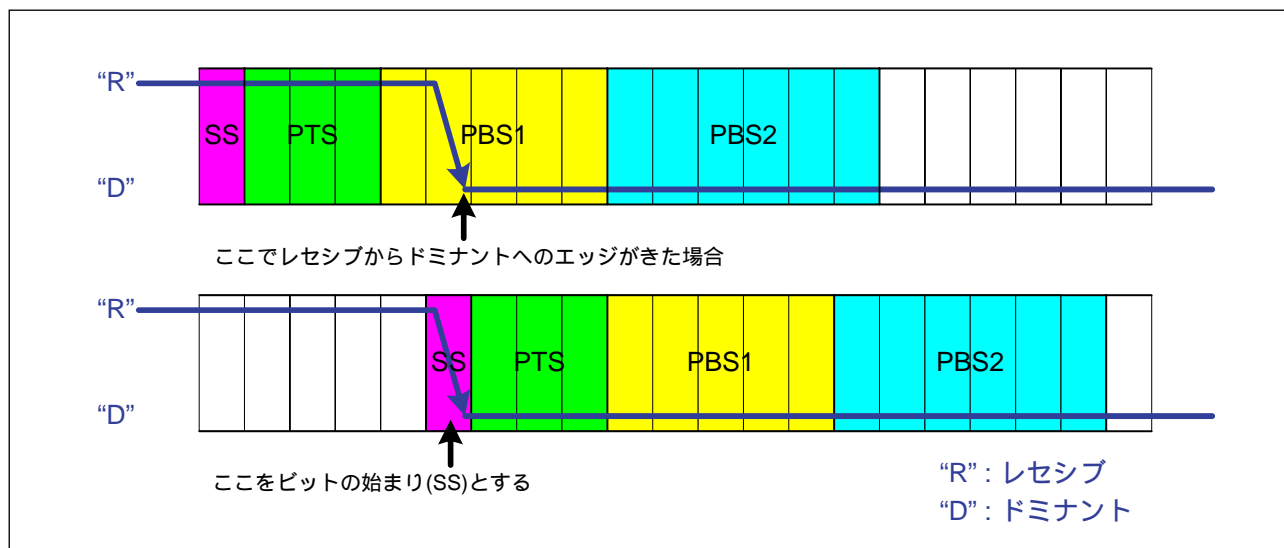


図 2 . ハードウェア同期の仕組み

(2) 再同期(メッセージ送受信中の同期)

発振器周波数のずれや伝送路の遅延などで、メッセージの送受信中に各ノード間の同期がずれる場合があります。これをフェーズエラーといいます。同期ずれが生じたとき、同期ずれに応じて SJW の値(SJW 値以上のずれが生じた場合は SJW 値)を PBS1 に加えたり PBS2 から減らしたりして 1 ビットの長さを動的に補正します。これを再同期といいます。

再同期の場合も、ハードウェア同期と同様にレセシブからドミナントへのエッジに対してのみ同期を取ります。

図 3、図 4 に再同期の仕組みを示します。

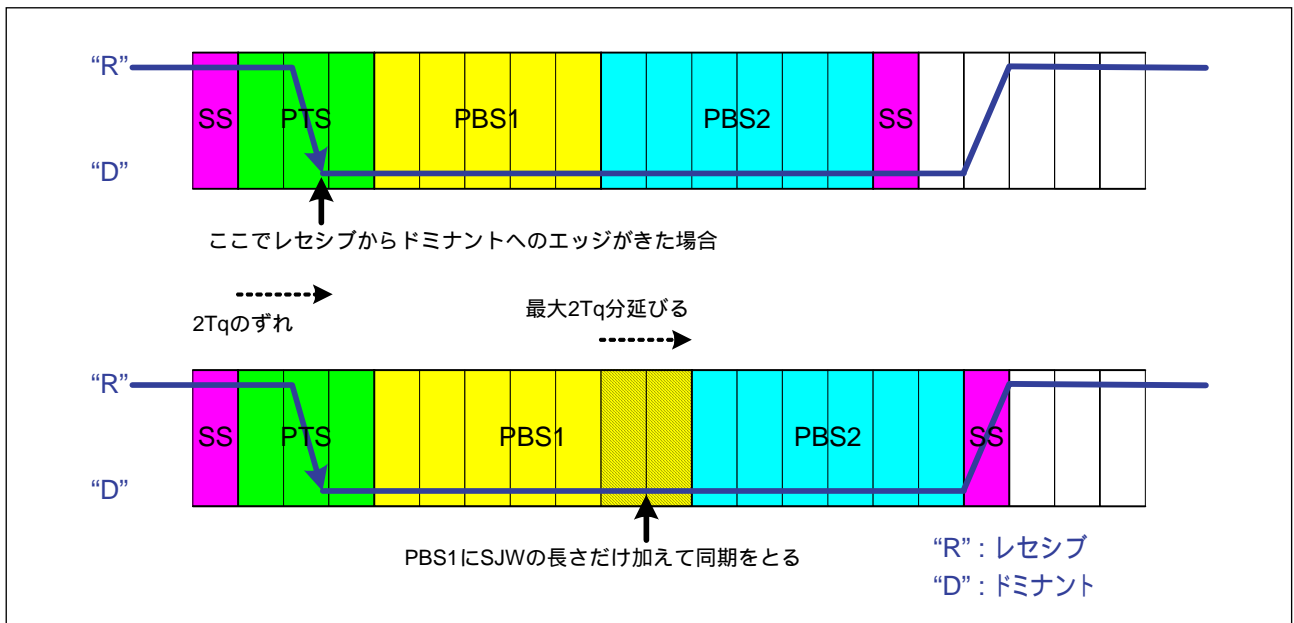


図 3 . 再同期の仕組み

R D へのエッジが PTS と PBS1 のどちらかの期間にきた場合(SJW=2 の例)

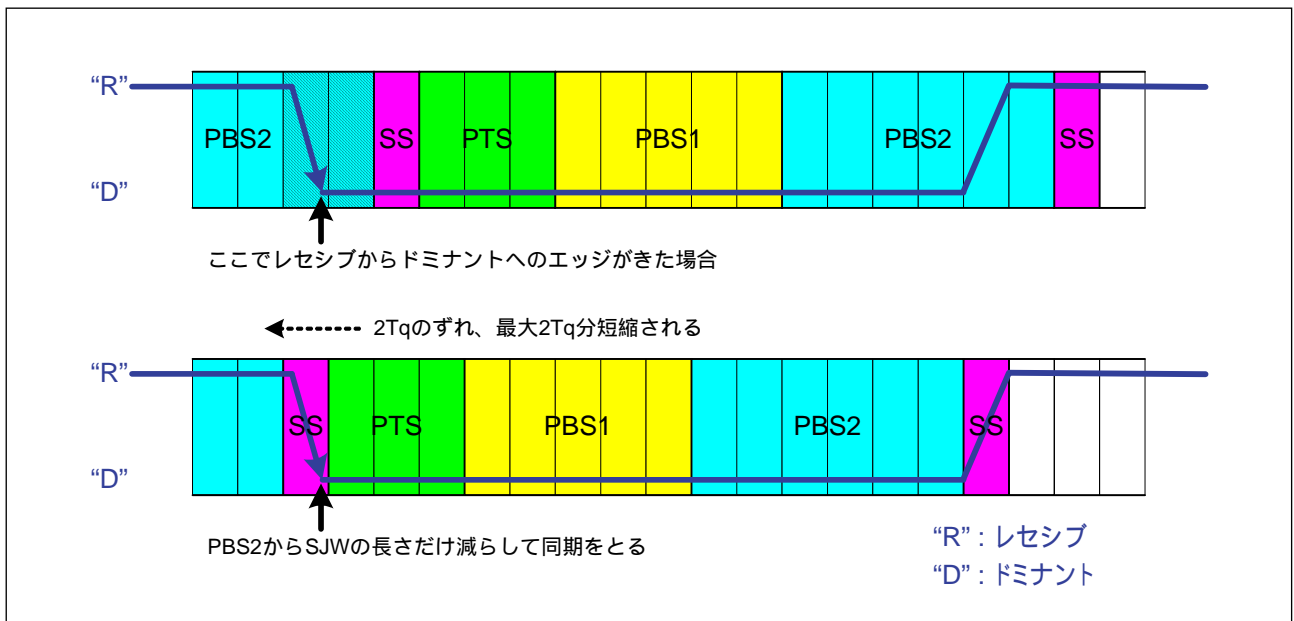


図 4 . 再同期の仕組み

R D へのエッジが PBS2 の期間にきた場合(SJW=2 の例)



3.3 転送速度

転送速度は、 $f_1$ 、CAN モジュールシステムクロック分周値、ボーレートプリスケアラ分周値、および 1 ビットの  $T_q$  数で決まります。

図 5 に CAN モジュールシステムクロックの発生回路ブロック図を示します。

表 1 に主な転送速度の算出式と実現例を、表 2 にビットタイミングの設定例を示します。

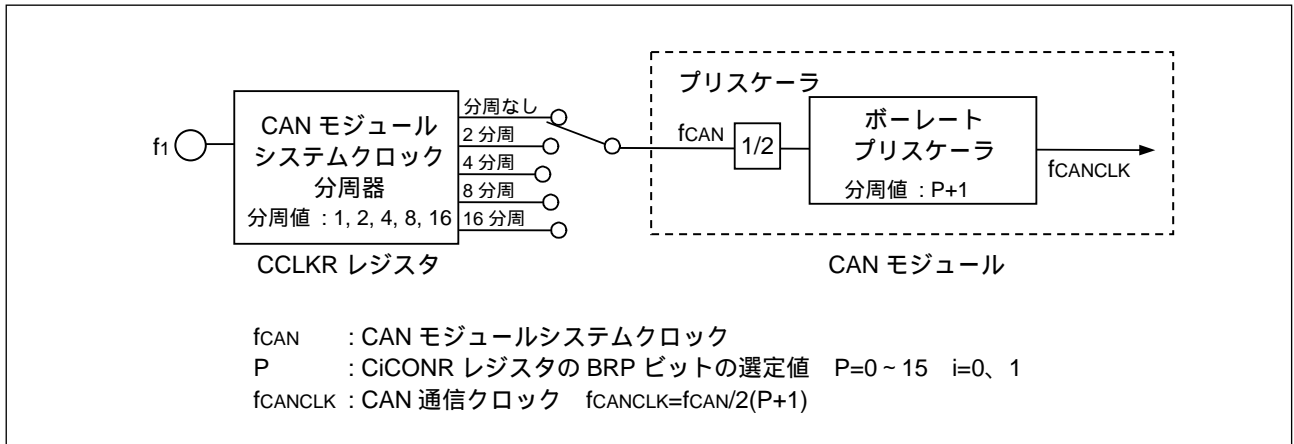


図 5 . CAN モジュールシステムクロックの発生回路ブロック図

表 1 . 転送速度の算出式と実現例

転送速度の算出式	$\frac{f_1}{2 \times f_{CAN} \text{分周値}^{*1} \times \text{ボーレートプリスケアラ分周値}^{*2} \times 1 \text{ビットの } T_q \text{ 数}}$				
	24MHz <sup>*3</sup>	20MHz <sup>*4</sup>	16MHz	10MHz	8MHz
1Mbps	12Tq (1)	10Tq (1)	8Tq (1)	-	-
500kbps	12Tq (2) 24Tq (1)	10Tq (2) 20Tq (1)	8Tq (2) 16Tq (1)	10Tq (1) -	8Tq (1) -
125kbps	12Tq (8) 16Tq (6) 24Tq (4)	10Tq (8) 20Tq (4) -	8Tq (8) 16Tq (4) -	10Tq (4) 20Tq (2) -	8Tq (4) 16Tq (2) -
83.3kbps	12Tq (12) 16Tq (9) 24Tq (6)	10Tq (12) 20Tq (6) -	8Tq (12) 16Tq (6) -	10Tq (6) 20Tq (3) -	8Tq (6) 16Tq (3) -
33.3kbps	12Tq (30) 24Tq (15)	10Tq (30) 20Tq (15)	8Tq (30) 16Tq (15)	10Tq (15) -	8Tq (15) -

\*1:  $f_{CAN}$  分周値 = 1、2、4、8、16

$f_{CAN}$  分周値:CCLKR レジスタの選定値

\*2: ボーレートプリスケアラ分周値= $P+1$ ( $P=0 \sim 15$ )

$P$ :CiCONR レジスタの BRP ビットの選定値( $i=0, 1$ )

\*3: 6NK、6NL、6NM、6NN グループのみ対応します

\*4: 1N グループは対応しません

\*5: ( )内の数字は  $f_{CAN}$  分周値 × ボーレートプリスケアラ分周値

表 2. ビットタイミングの設定例

1ビット	設定値(Tq)					サンプルポイント <sup>*1</sup> (%)
	SS	PTS	PBS1	PBS2	SJW	
8Tq	1	1	3	3	1	62.50
	1	3	2	2	1	75.00
10Tq	1	3	3	3	1	70.00
	1	5	2	2	1	80.00
16Tq	1	5	5	5	1	68.75
	1	7	4	4	1	75.00
20Tq	1	7	6	6	1	70.00
	1	5	7	7	1	65.00

\*1: サンプルポイント(サンプルポイントが75%の場合)



### 3.4 CAN ビットタイミングと転送速度の設定

図 6に CAN ビットタイミングと転送速度の設定手順を示します。  
これらの設定は必ず CAN のコンフィグレーション中に行ってください。  
CAN コンフィグレーションの手順は4.1項を参照ください。

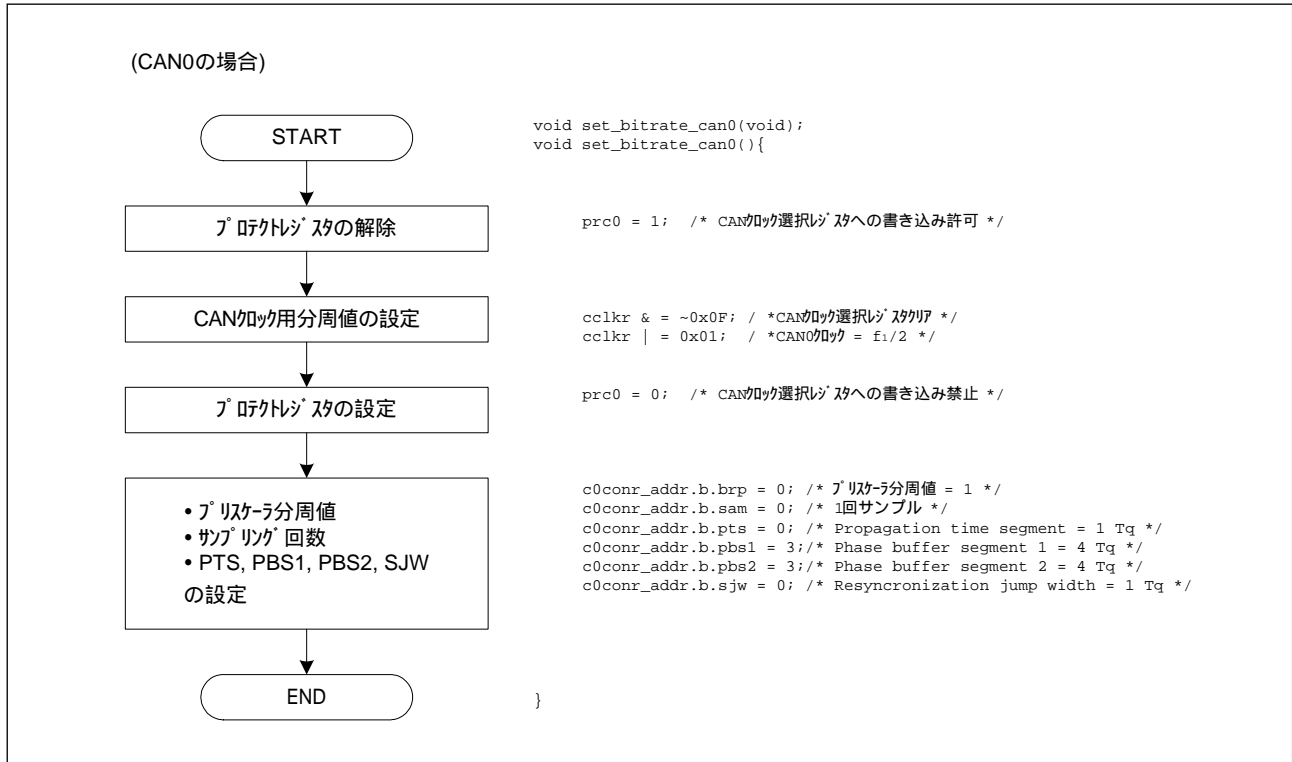


図 6 . CAN ビットタイミングと転送速度の設定手順

#### 4. CAN メッセージの送受信

CAN メッセージの送受信は次の 3 つの手順で行われます。

##### (1) CAN コンフィグレーション手順

リセット/初期化モード時<sup>\*1</sup>に CAN 制御レジスタ、CAN バスタイミング制御レジスタ、CAN クロック選択レジスタ、マスクレジスタの各設定を行います。

<sup>\*1</sup>: CAN メッセージ制御レジスタ、CAN 割り込み制御レジスタ、CAN 拡張 ID レジスタは、CAN が動作モードからリセット/初期化モードへ移行すると、自動的にクリアされます。またこれらのレジスタはリセット/初期化モードでは設定することができません。CAN を動作モードに復帰させた後、設定してください。

##### (2) スロットコンフィグレーション手順

送受信の各モードは、各スロットに対応した CAN メッセージ制御レジスタで設定します。表 3 に CAN メッセージ制御レジスタの設定と送信・受信モードの関係を示します。

##### (3) データ処理手順

メッセージ送信または受信正常終了時のメッセージ処理を行います。

表 3. CAN メッセージ制御レジスタの設定と送信・受信モードの関係

C0MCTL/C1MCTL				スロットの送信・受信モード設定内容
bit7 TrmReq	bit6 RecReq	bit5 Remote	bit4 RspLock	
0	0	-	-	送受信を行わない
0	1	0	0	データフレームを受信する(データフレーム受信モード)
1	0	1	0	リモートフレームを送信し、送信完了後、データフレームを受信する (リモートフレーム送信/データフレーム受信モード)
1	0	0	0	データフレームを送信する(データフレーム送信モード)
0	1	1	1/0 <sup>*1</sup>	リモートフレームを受信し、受信完了後、データフレームを送信する (リモートフレーム受信/データフレーム送信モード)  <sup>*1</sup> RspLock: 0: リモートフレーム受信後、自動応答でデータフレーム送信 1: リモートフレーム受信後、データフレーム送信待機 このビットをクリアすることでデータフレーム送信

### 4.1 CAN コンフィグレーション

図 7に CAN コンフィグレーション手順を示します。

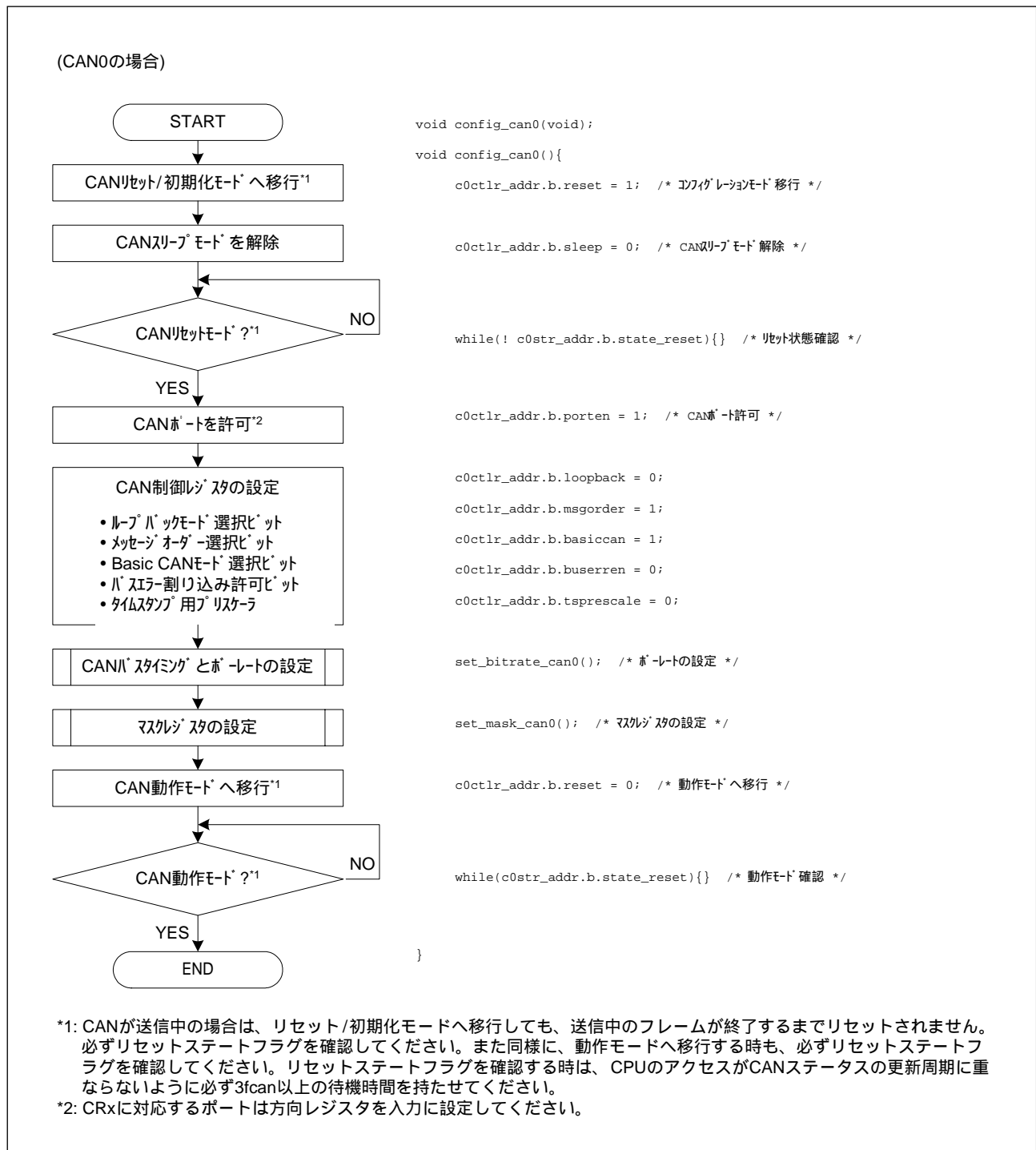


図 7 . CAN コンフィグレーション手順

## 4.2 メッセージ送信

送信モードには、以下の2つのモードがあります。

- データフレーム送信モード
- リモートフレーム受信/データフレーム送信モード

### (1) データフレーム送信モード

スロットをデータフレーム送信モードに設定すると、スロットに設定しているデータフレームを送信できます。

### (2) リモートフレーム受信/データフレーム送信モード

スロットをリモートフレーム受信後、データフレーム送信モードに設定すると、スロットに設定しているIDのリモートフレームを受信後、自動的に同じスロットに設定しているデータを送信できます。この時、送信するデータのバイト数は、受信したリモートフレームのDLC値で決まります。

4.2.1 データフレーム送信モード

(1) 送信手順

スロットをデータフレーム送信モードに設定すると、スロットに設定しているデータフレームを送信できます。

2つ以上のスロットに、同時にデータフレームの送信設定を行った場合、番号の小さいスロットから順に送信されます。

図 8にデータフレーム送信手順を示します。

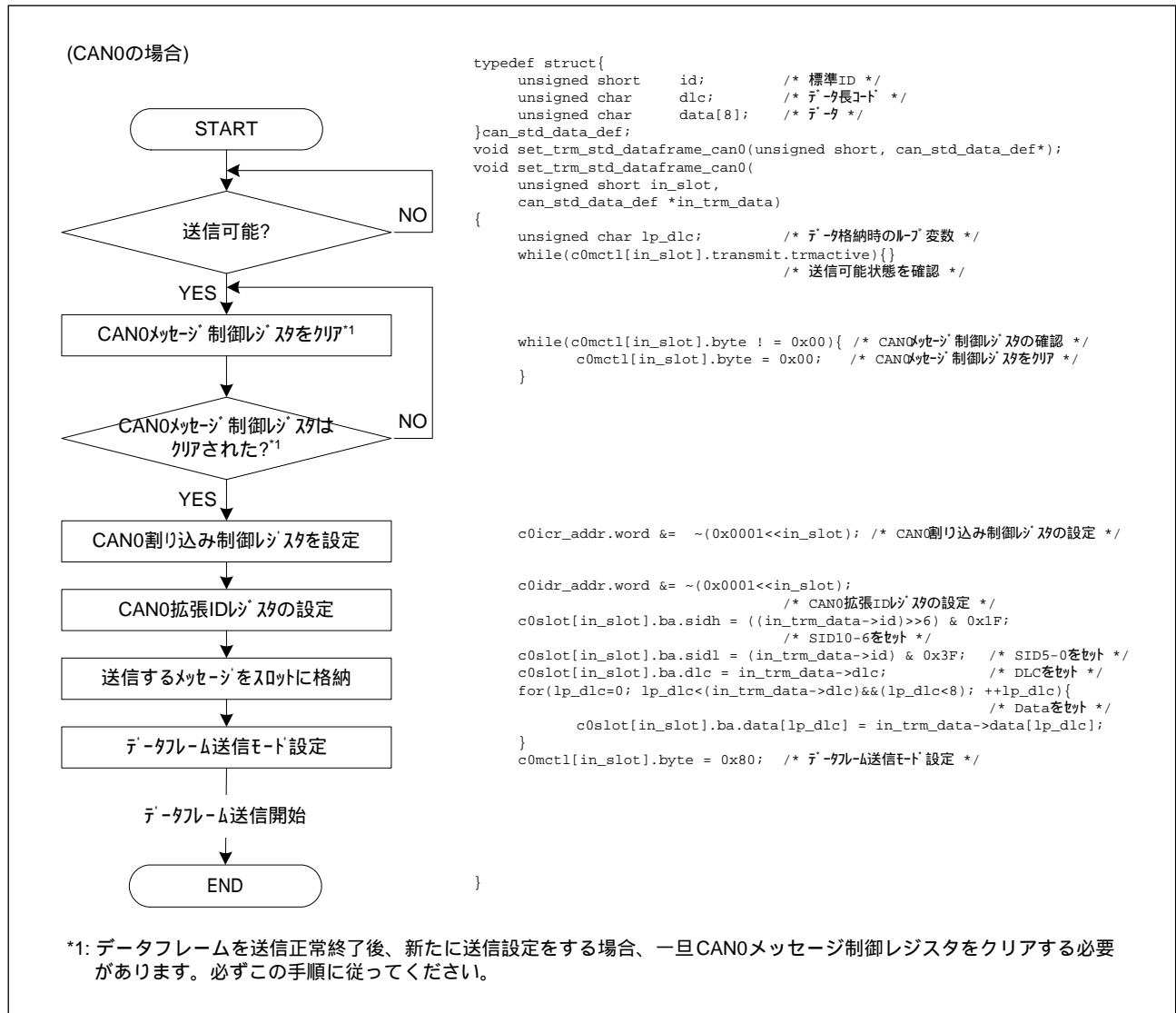


図 8. データフレーム送信手順

### (2) 送信正常終了確認手順

メッセージの送信正常終了を確認するには、ポーリングを使用する方法と、割り込みを使用する方法があります。

#### ポーリングを使用する場合

CAN メッセージ制御レジスタをポーリングすることで送信正常終了を確認できます。

図 9 にポーリングを使用する場合の送信正常終了確認手順を示します。

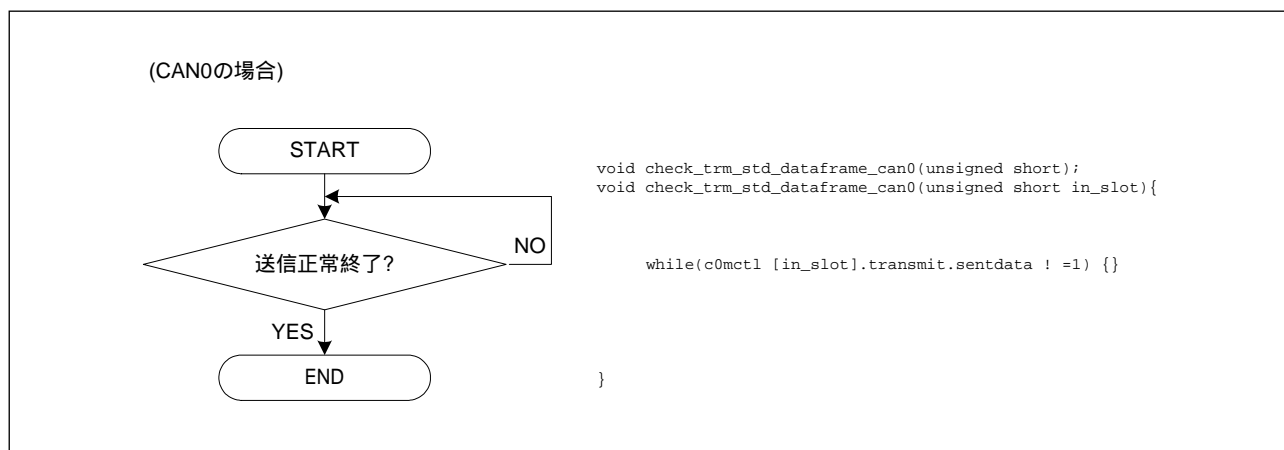


図 9 . 送信正常終了確認手順(ポーリングを使用する場合)



CAN 送信完了割り込みを使用する場合

CAN 送信完了割り込みを使用するには、CAN 送信完了割り込み制御レジスタを使用許可に設定し、その後、CAN 割り込み制御レジスタの各スロットに対応したビットを"1"にします。この CAN 割り込み制御レジスタは、CAN 送信完了割り込みと CAN 受信完了割り込みとで兼用です。CAN 割り込み制御レジスタは、CAN モジュールがリセット/初期化モードへ移行すると自動的にクリアされます。また、リセット/初期化モード中は設定できません。必ず CAN が動作モード中に設定してください。

図 10、図 11に CAN 送信完了割り込み制御レジスタ設定手順を、図 12に CAN 送信完了割り込みを使用する場合の送信正常終了確認手順を示します。

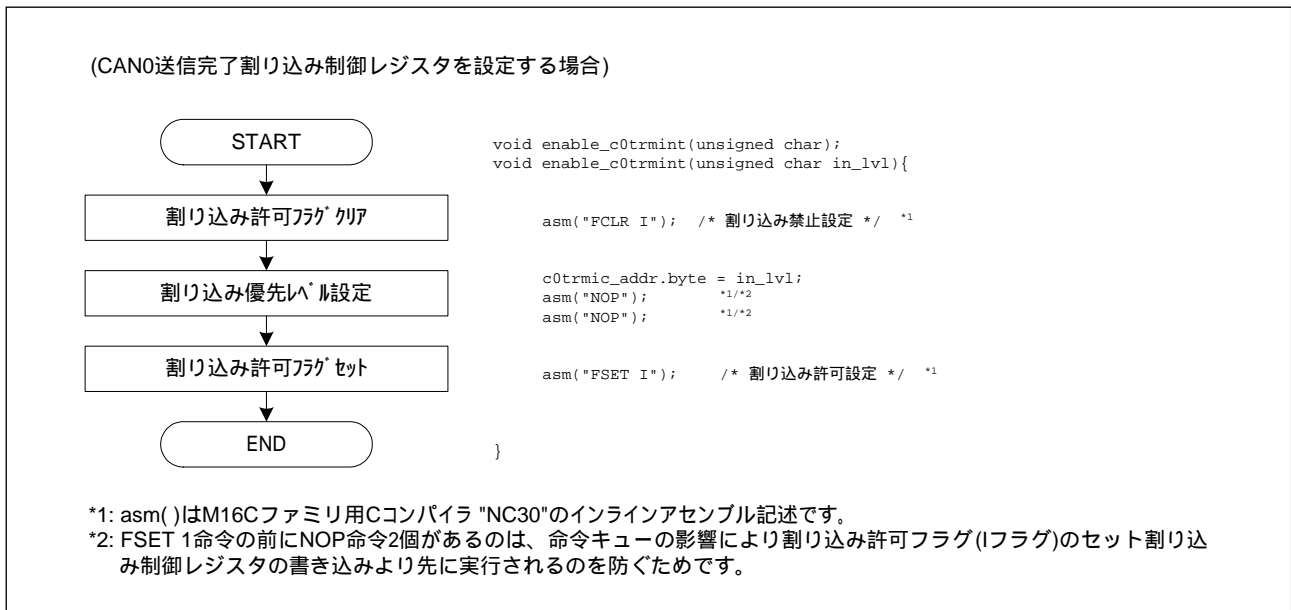


図 10 . CAN0 送信完了割り込み制御レジスタ設定手順

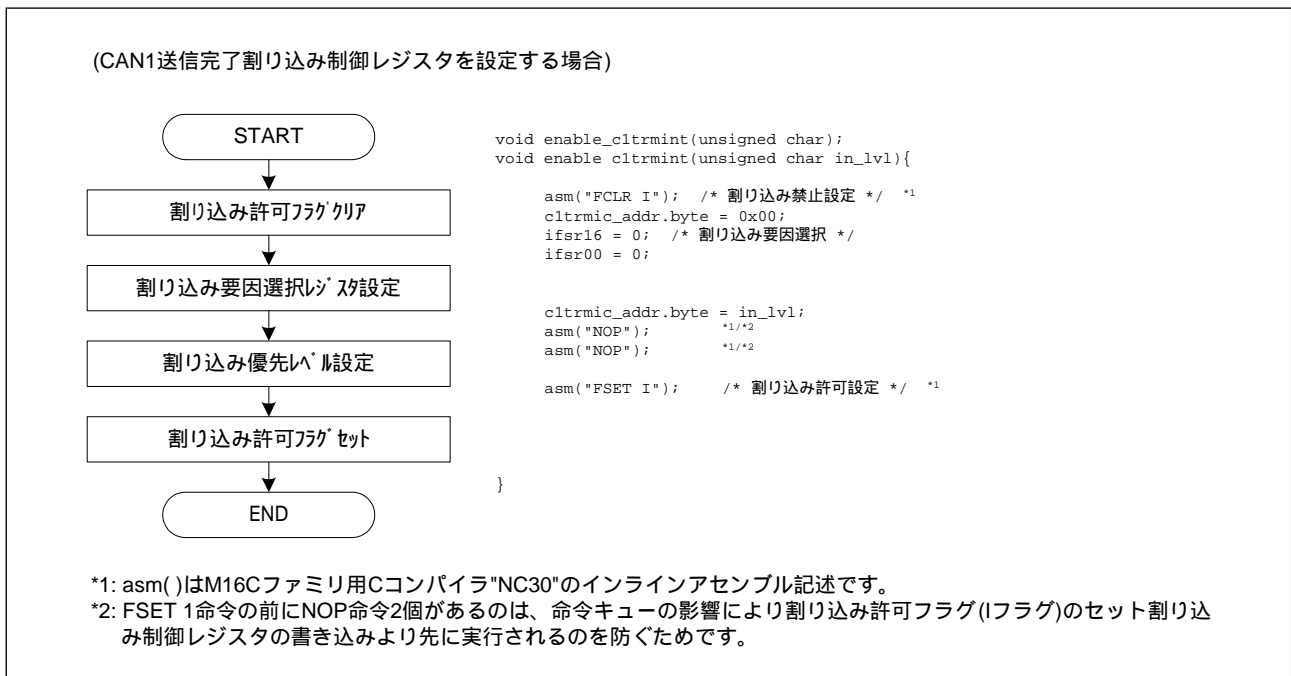


図 11 . CAN1 送信完了割り込み制御レジスタ設定手順

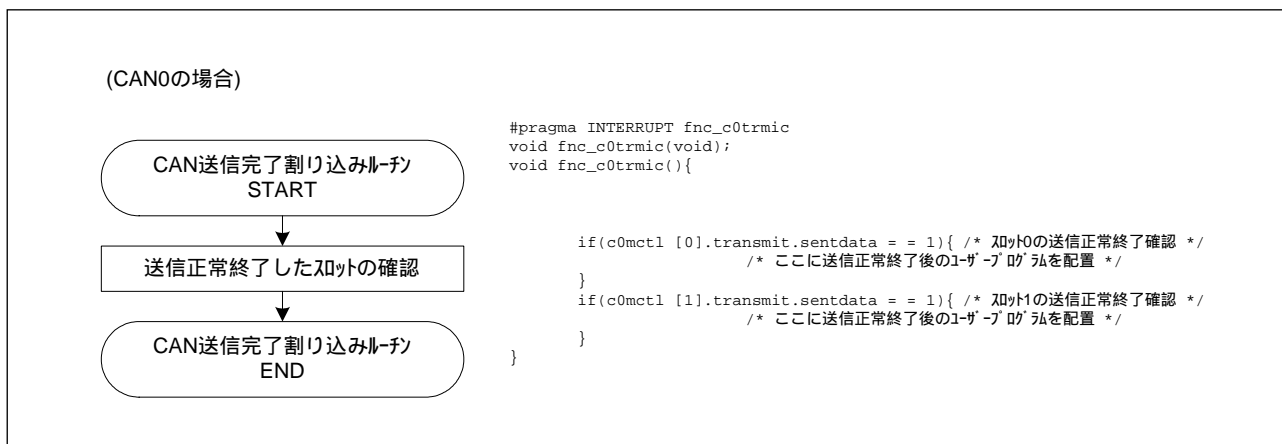


図 12 . 送信正常終了確認手順(CAN 送信完了割り込みを使用する場合)

#### 4.2.2 リモートフレーム受信/データフレーム送信モード

スロットをリモートフレーム受信/データフレーム送信モードに設定すると、スロットに設定している ID のリモートフレームを受信後、自動的に同じスロットに設定しているデータを送信できます\*1。この時、送信するデータのバイト数は、受信したリモートフレームの DLC 値で決まります。

リモートフレーム受信/データフレーム送信モードでは、CAN メッセージ制御レジスタの送受信自動応答ロックモード選択ビットの設定で、リモートフレーム受信後に自動応答するか否かを選択できます(0:自動応答する、1:自動応答しない)。

自動応答をする場合、リモートフレームを受信後、自動的にデータフレームの送信を開始します。

自動応答をしない場合、リモートフレームを受信後、このビットを"0"に設定すると、データフレームの送信を開始します。

図 13、図 14にリモートフレーム受信/データフレーム送信手順を示します。

\*1: Basic CAN モード使用時は、スロット 14、スロット 15 はリモートフレーム受信/データフレーム送信モードに設定できません。  
詳細は4.5項を参照ください。

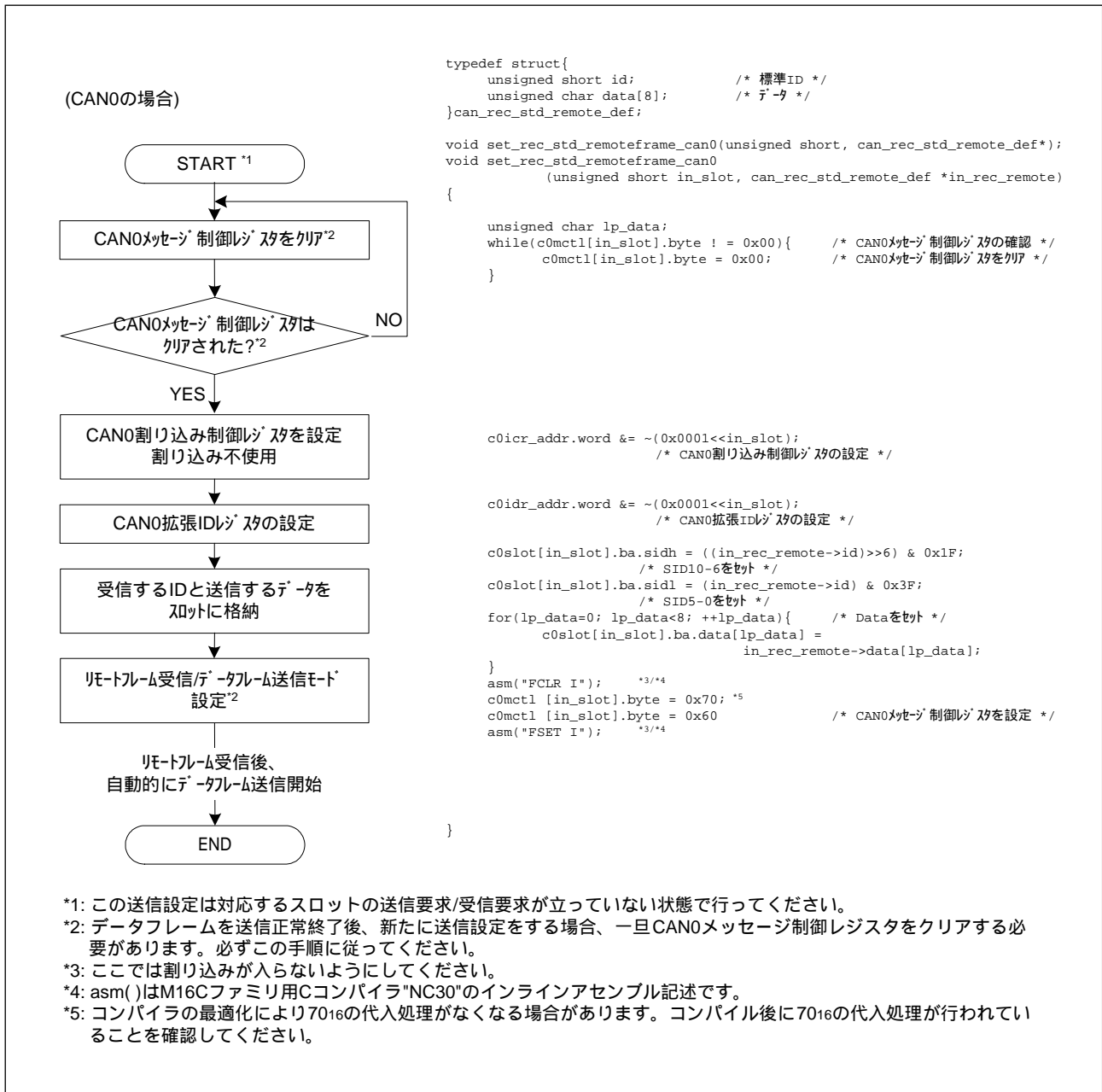


図 13 . リモートフレーム受信/データフレーム送信手順(リモートフレーム受信後、自動応答する場合)

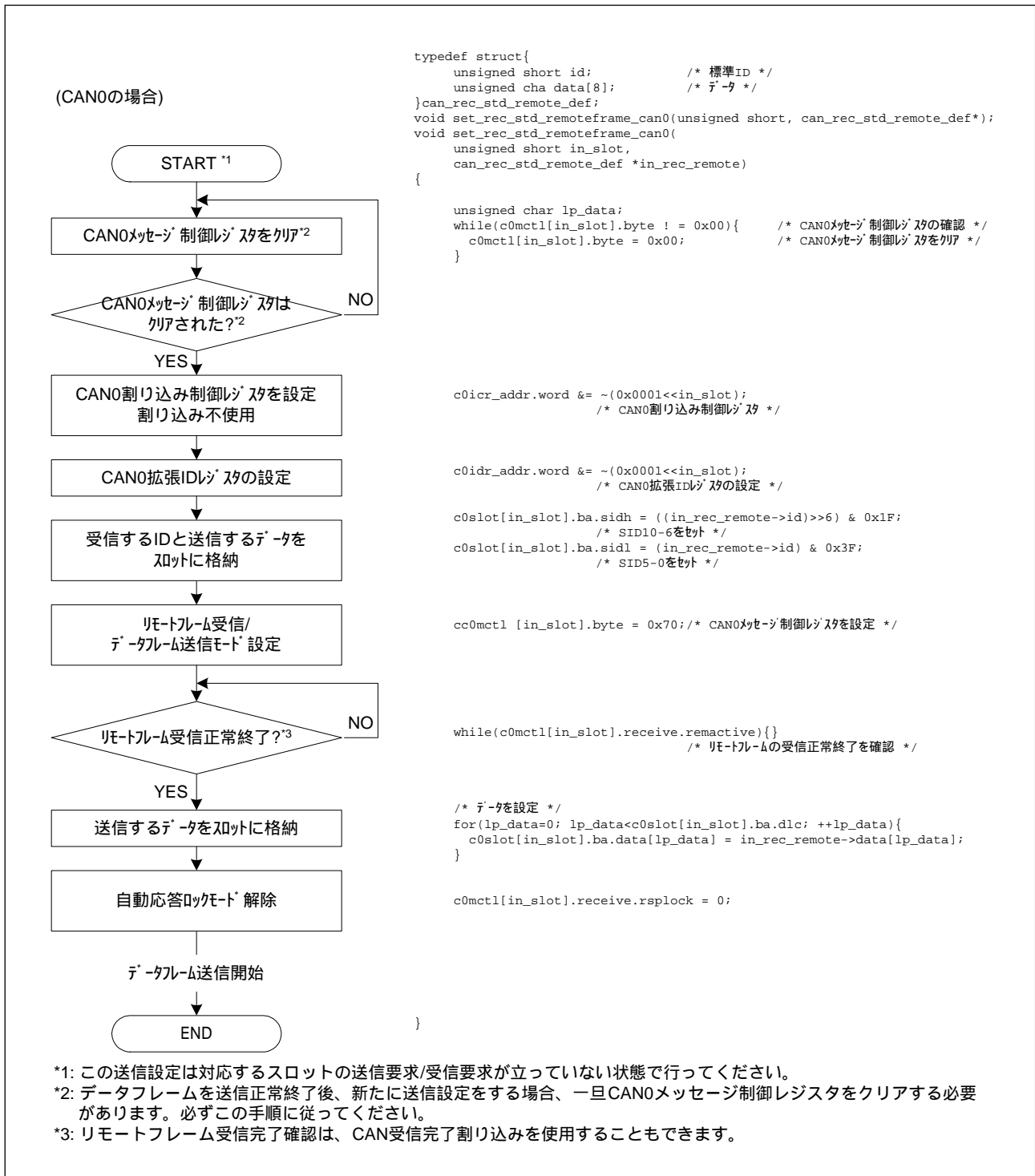


図 14 . リモートフレーム受信/データフレーム送信手順(リモートフレーム受信後、自動応答しない場合)

### 4.2.3 送信アボート

2 つ以上のノードが同時に送信を始めた場合、メッセージの優先度が低いノードはアービトレーションに負けて送信を中止します(アービトレーションに勝ったメッセージの送信終了後に再送信を行います)。アービトレーションに勝たない限りメッセージの送信が正常に終了しないため、再送信を永久に繰り返し新たなメッセージを送信できない状態になります。このような時に、再送信中のメッセージを破棄するための送信アボート機能があります。送信アボート機能は 1 つのメッセージ送信に制限時間を設けたい時や、緊急な優先順位の高いメッセージを送信する時などに有効です。

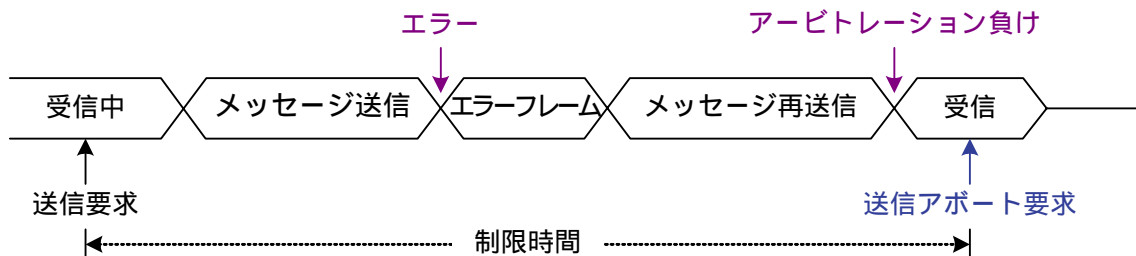
図 15 に送信アボート機能の応用例を示します。

送信アボートは、CAN メッセージ制御レジスタをクリアして実行します。

- (1) 送信アボート要求の有効となる条件
  - データフレームまたはリモートフレームを送信していた場合  
CAN メッセージ制御レジスタの送信スロット設定ビットが"1"の時のみ送信アボート有効
  - リモートフレームを受信後、データフレームを送信していた場合  
CAN メッセージ制御レジスタの受信スロット設定ビットが"1"の時のみ送信アボート有効
- (2) 送信アボートが実行される条件
  - メッセージがアービトレーションに負けた場合(図 16、図 17)
  - メッセージ送信中にエラーが発生した場合(図 16、図 17)

(1) メッセージ送信に制御時間を設けたい時

メッセージの送信開始から指定時間後、送信アポートを要求することでメッセージ送信に制限時間を設けることができます。



(2) 優先順位の高いメッセージを送信したい時

メッセージの送信中に送信アポートを実行すると、送信中のメッセージがアービトレーションに負けるかエラーを検知した後、そのメッセージが破棄され、高優先順位のメッセージを送信できます。

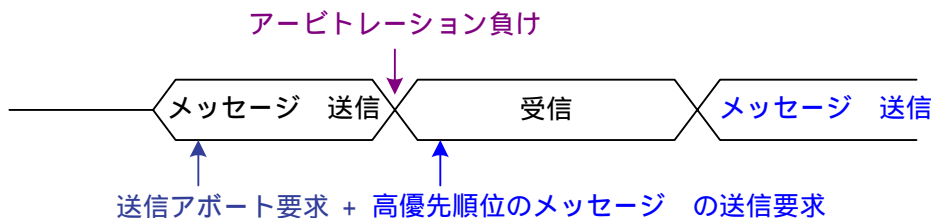
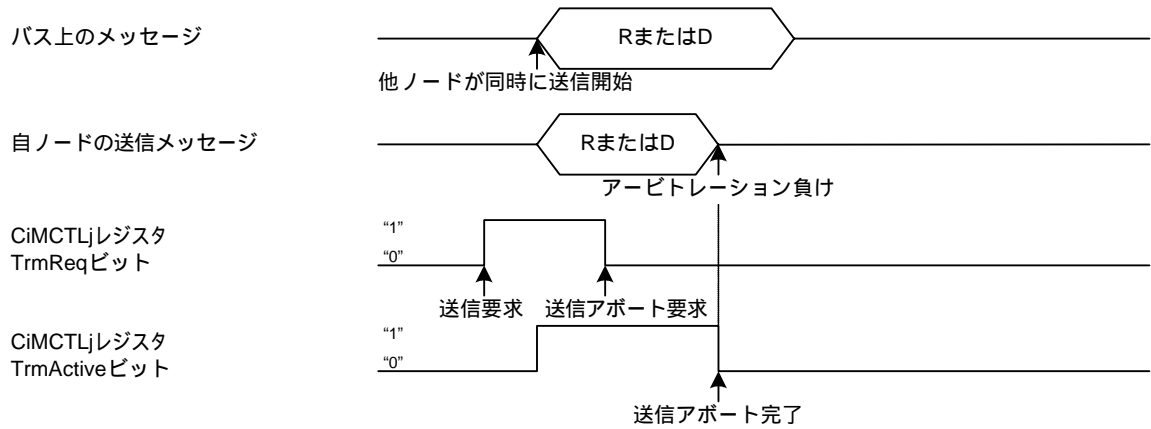
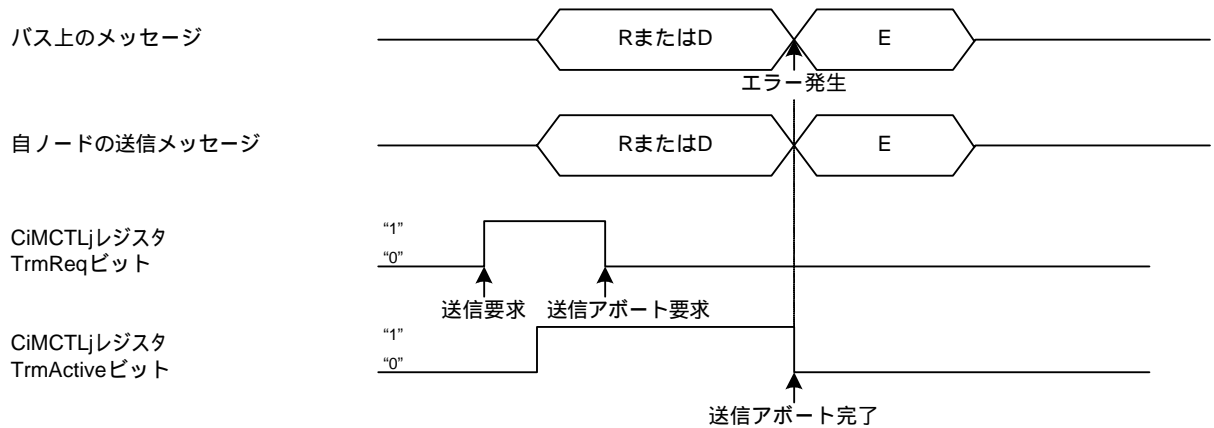


図 15 . 送信アポート機能の応用例

アービトレーションに負けた場合



エラーが発生した場合



送信アボートが無効になる場合(正常に送信が完了した場合)

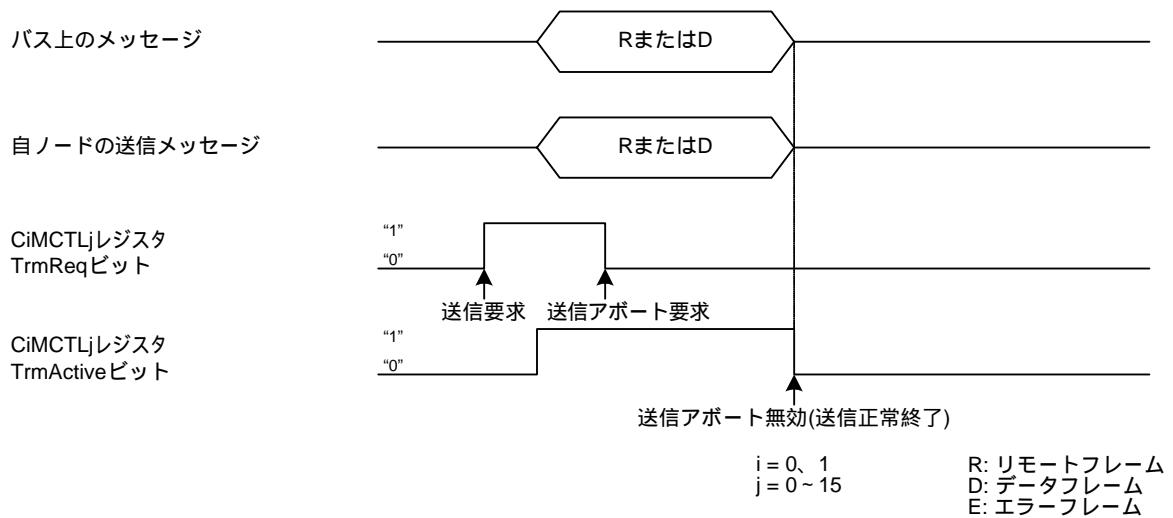
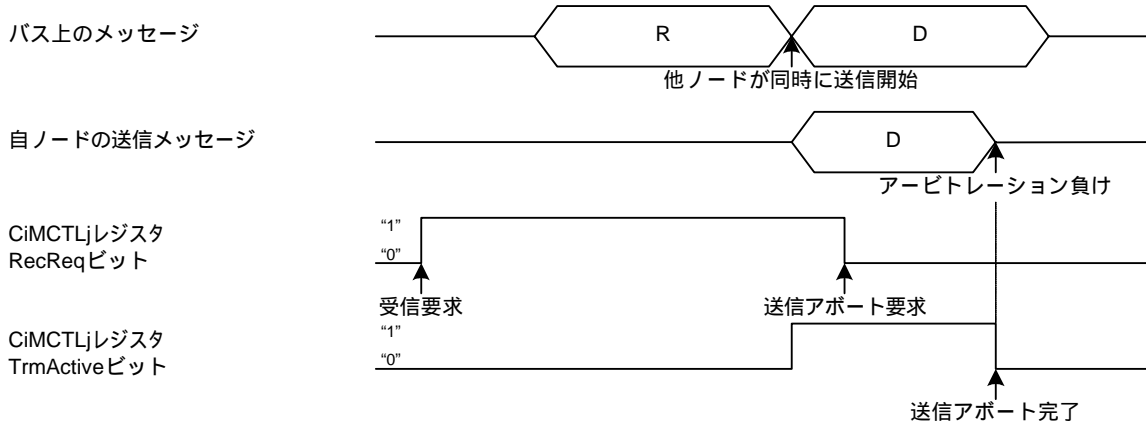


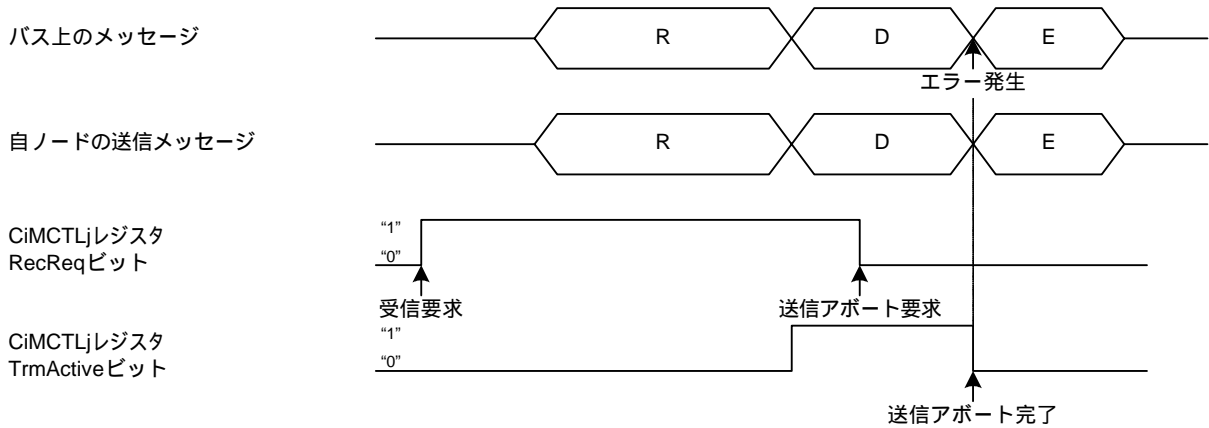
図 16 . リモートフレームまたはデータフレーム送信中での送信アボート



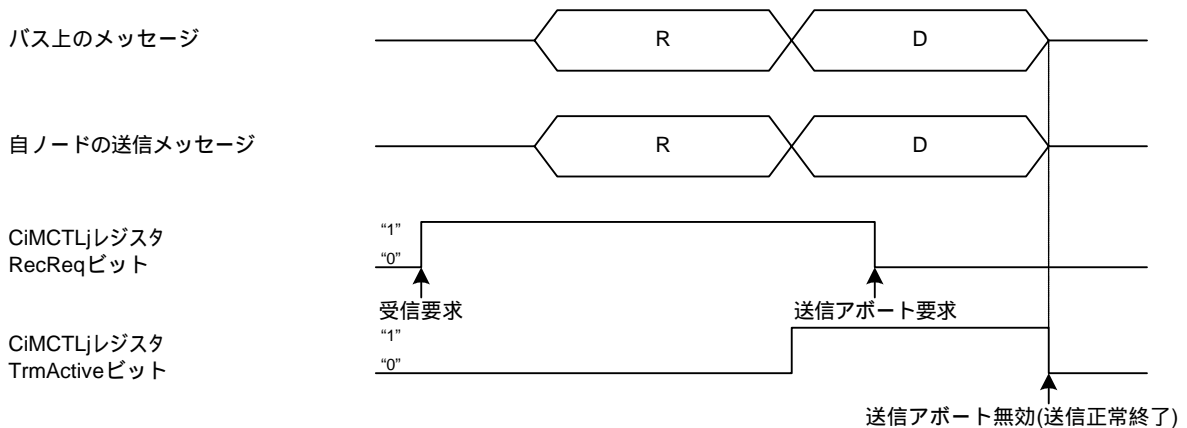
アービトレーションに負けた場合



エラーが発生した場合



送信アボートが無効になる場合(正常に送信が完了した場合)



i = 0, 1  
j = 0 ~ 15  
R: リモートフレーム  
D: データフレーム  
E: エラーフレーム

図 17. リモートフレーム受信後、データフレーム送信中での送信アボート

(3) 送信アボートの手順

図 18に送信アボート手順を示します。

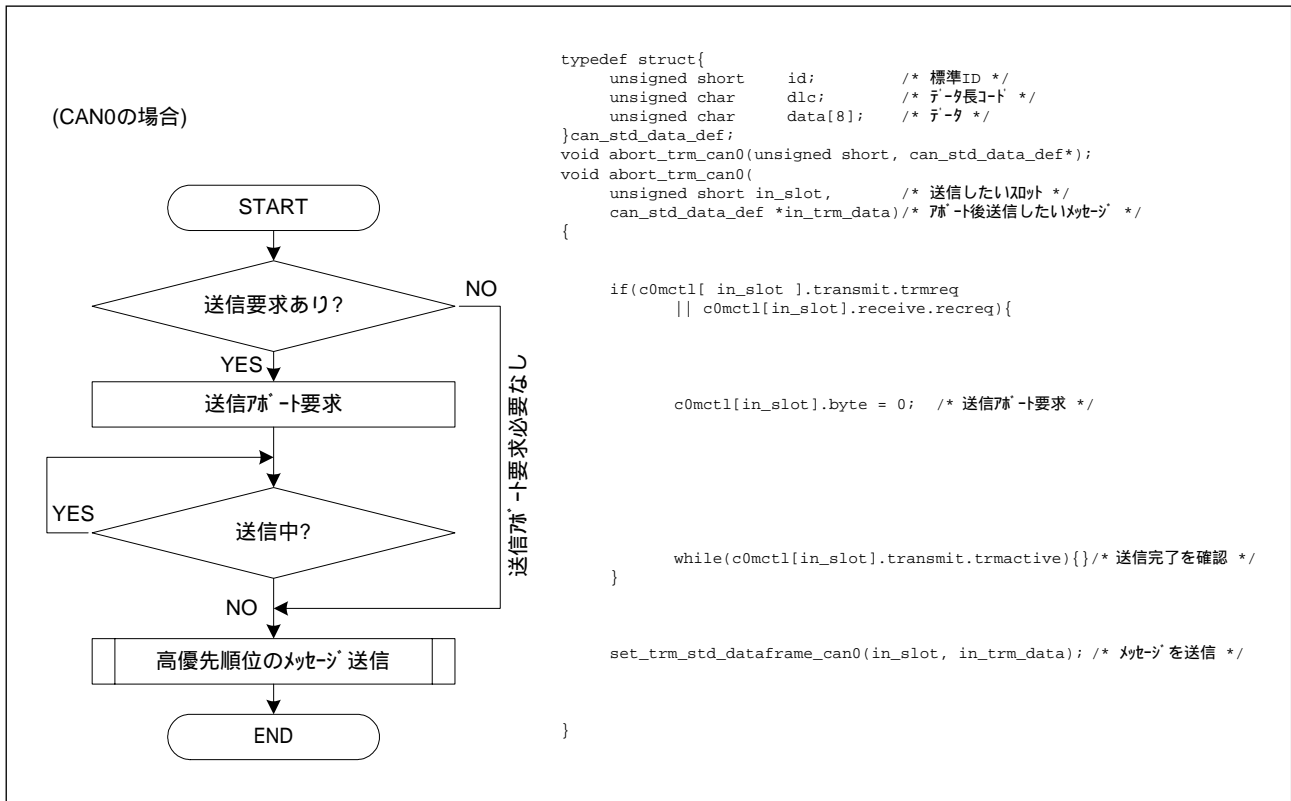


図 18 . 送信アボート手順

### 4.3 メッセージ受信

CAN1 チャンネルにつき 16 個のスロットがあり、受信メッセージはアクセプタンスフィルタを使用しない場合、受信に設定したスロットのうち常に番号の一番小さいスロットに格納されます。アクセプタンスフィルタを使用すると受信するメッセージを選択できます。アクセプタンスフィルタについては6項を参照ください。

受信モードには、以下の2つのモードがあります。

- データフレーム受信モード
- リモートフレーム送信/データフレーム受信モード

(1) データフレーム受信モード

スロットをデータフレーム受信モードに設定すると、スロットに設定している ID のデータフレームを受信できます。

(2) リモートフレーム送信/データフレーム受信モード

スロットをリモートフレーム送信/データフレーム受信に設定すると、スロットに設定している ID、DLC のリモートフレームを送信後、自動的に同じ ID のデータフレームを受信できます。

### 4.3.1 データフレーム受信モード

#### (1) 受信設定手順

2つの以上のスロットに、同じIDのデータフレームを受信する設定を行った場合、受信メッセージは常に番号の一番小さいスロットに格納されます。

図 19にデータフレーム受信設定手順を示します。

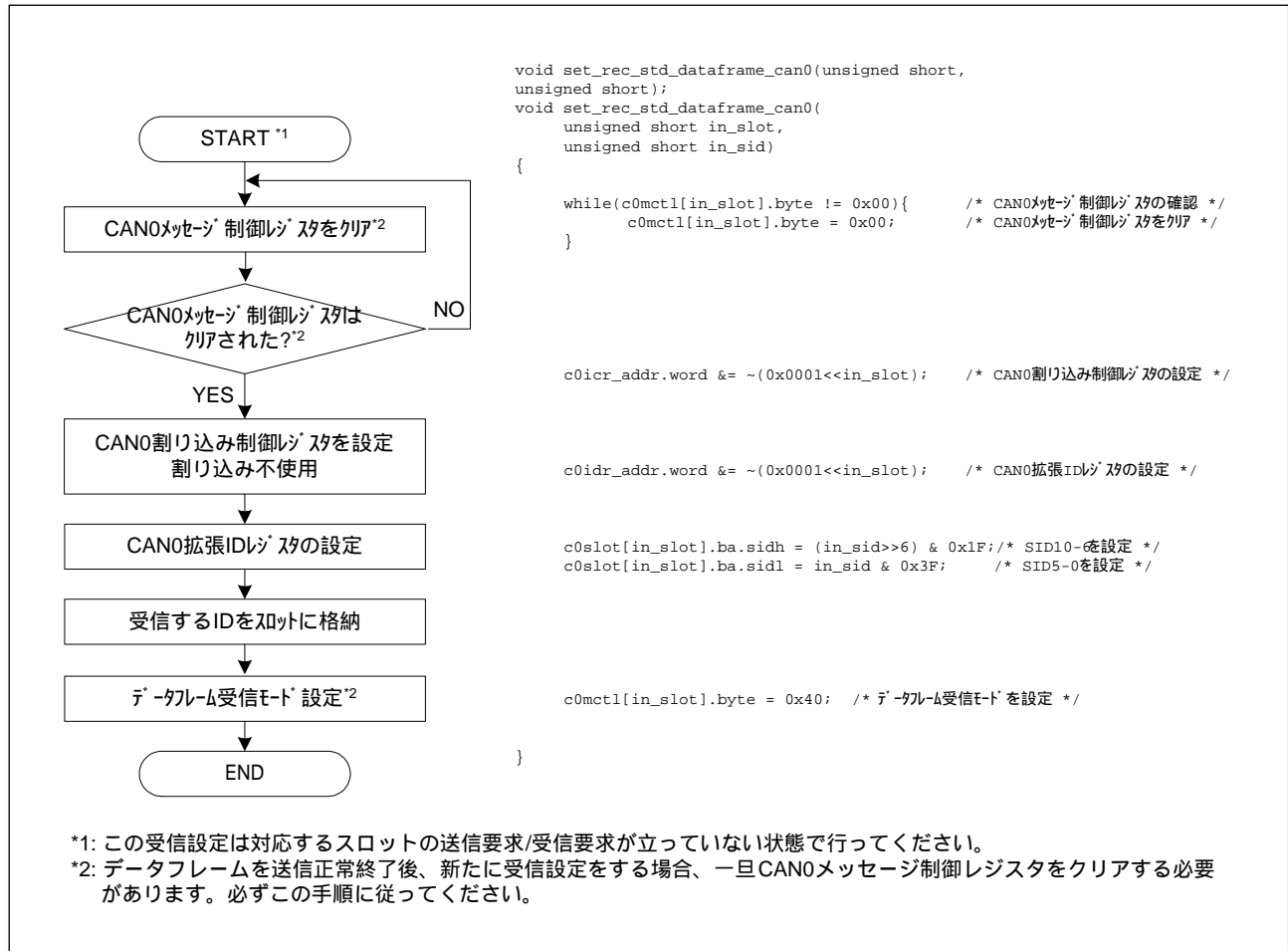


図 19 . データフレーム受信設定手順

## (2) 受信メッセージ処理手順

図 20に受信メッセージ処理手順を示します。

受信正常終了後、さらに同スロットが受信正常終了した場合、スロットは新しい受信メッセージでオーバーライトされます。詳細は4.4項を参照ください。そのため受信メッセージの読み出し完了後、読み出し中にスロットがオーバーライトされていないことを確認する必要があります。

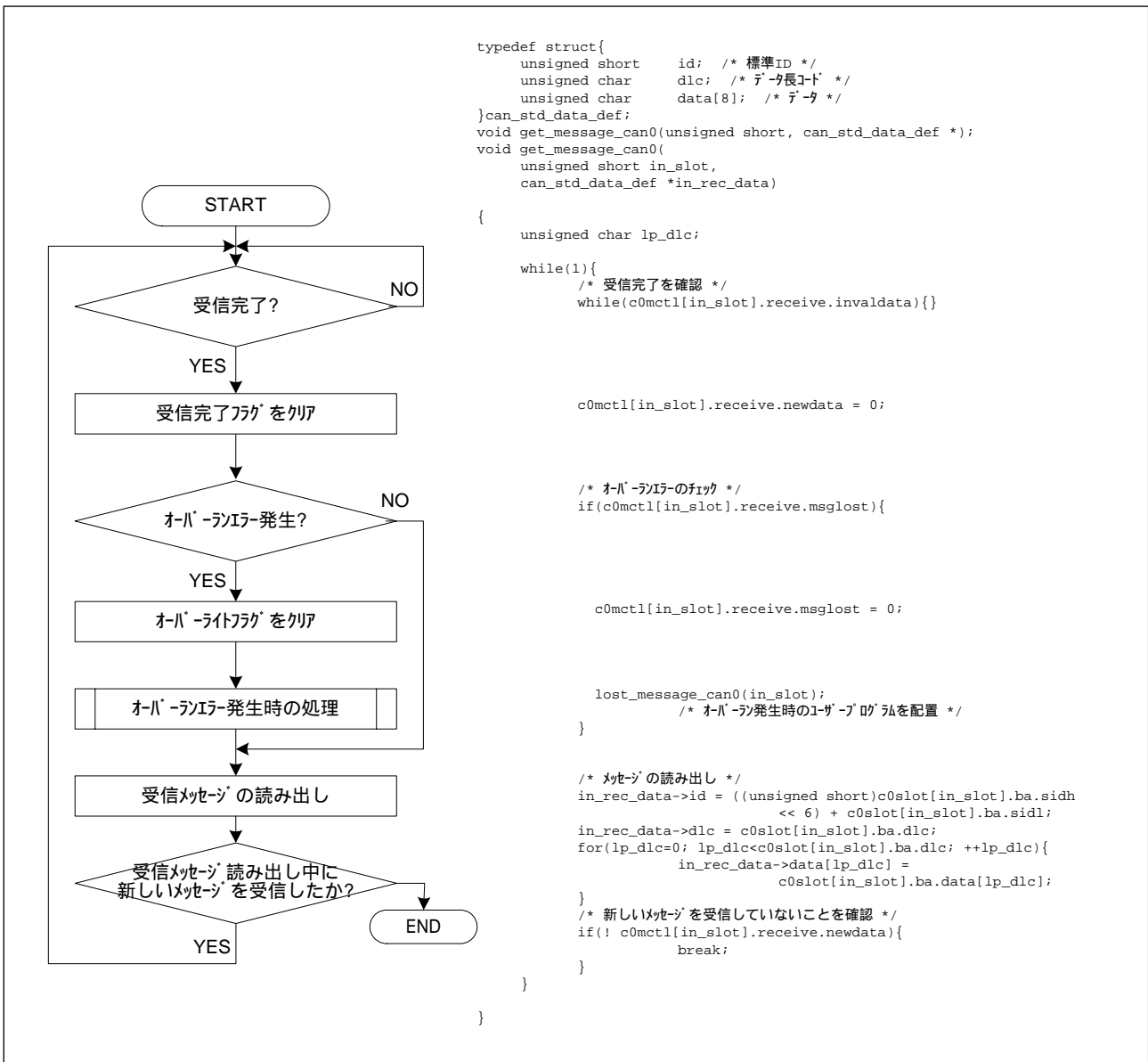


図 20 . 受信メッセージ処理手順

### (3) 受信正常終了確認手順

メッセージの受信正常終了を確認するには、ポーリングを使用する方法と、割り込みを使用する方法があります。

#### ポーリングを使用する場合

CAN メッセージ制御レジスタをポーリングすることで受信正常終了を確認することができます。

図 21 にポーリングを使用する場合の受信正常終了確認手順を示します。

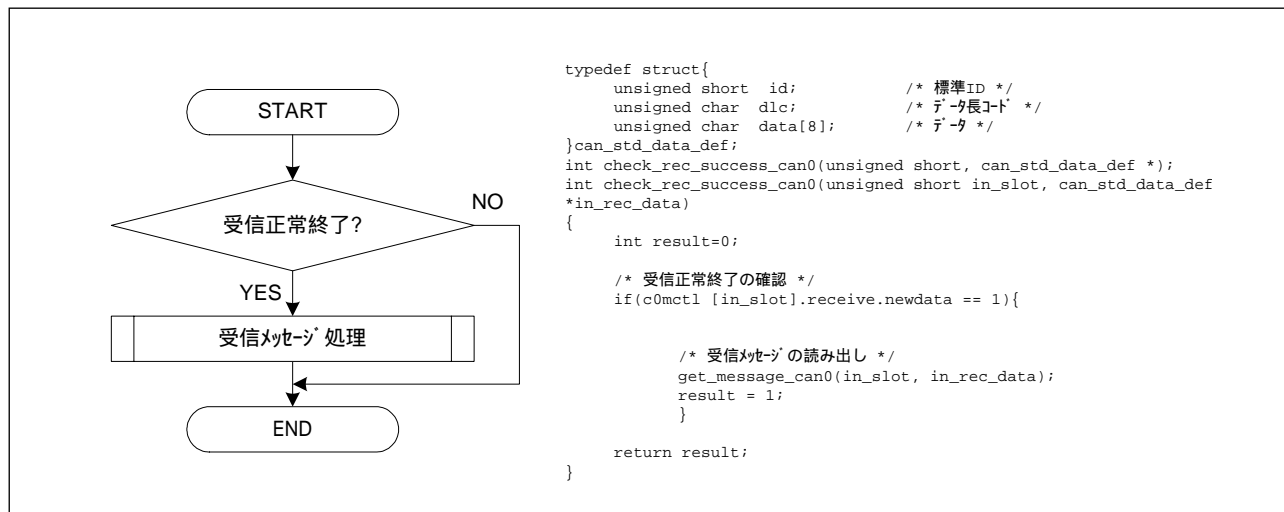


図 21 . 受信正常終了確認手順(ポーリングを使用する場合)

CAN 受信完了割り込みを使用する場合

CAN 受信完了割り込みを使用するには、CAN 受信完了割り込み制御レジスタを使用許可に設定し、その後、CAN 割り込み制御レジスタの各スロットに対応したビットを"1"にします。この CAN 割り込み制御レジスタは、CAN 受信完了割り込みと CAN 送信完了割り込みとで兼用です。CAN 割り込み制御レジスタは、CAN モジュールがリセット/初期化モードへ移行すると自動的にクリアされます。また、リセット/初期化モードでは設定することができません。必ず CAN が動作モード中に設定してください。

図 22、図 23に CAN 受信完了割り込み制御レジスタ設定手順を、図 24に CAN 受信完了割り込みを使用する場合の受信正常終了確認手順を示します。

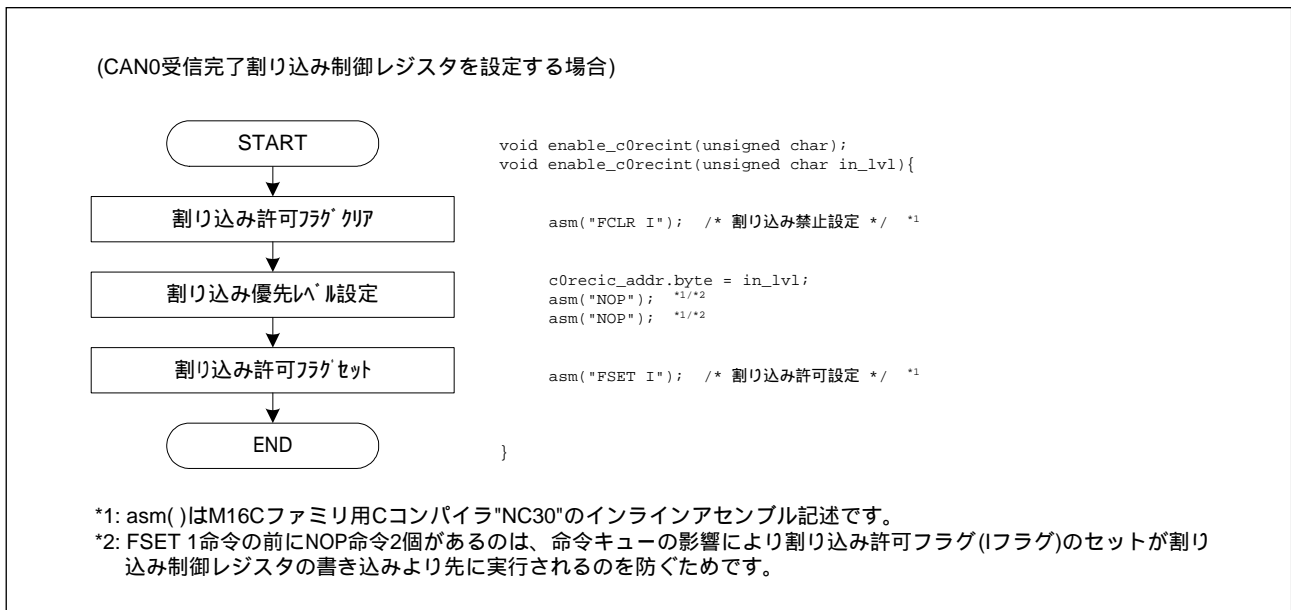


図 22 . CAN0 受信完了割り込み制御レジスタ設定手順

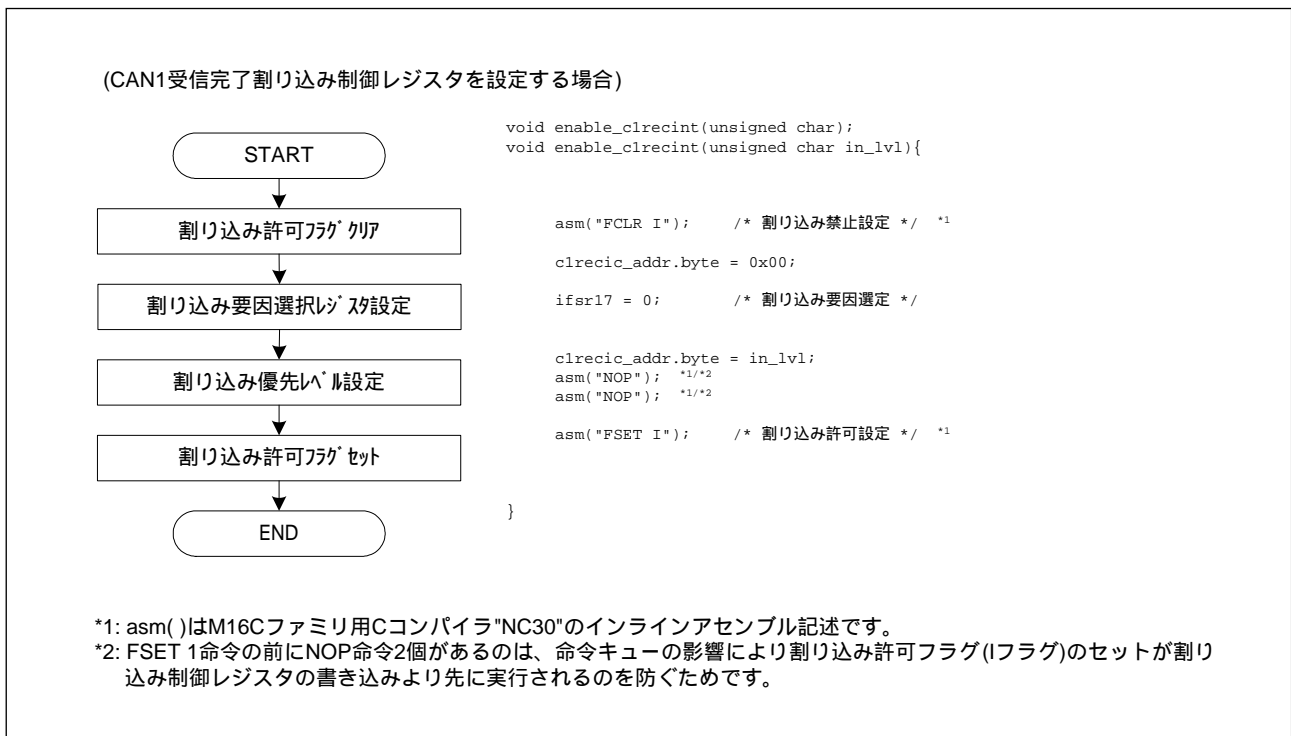


図 23 . CAN1 受信完了割り込み制御レジスタ設定手順



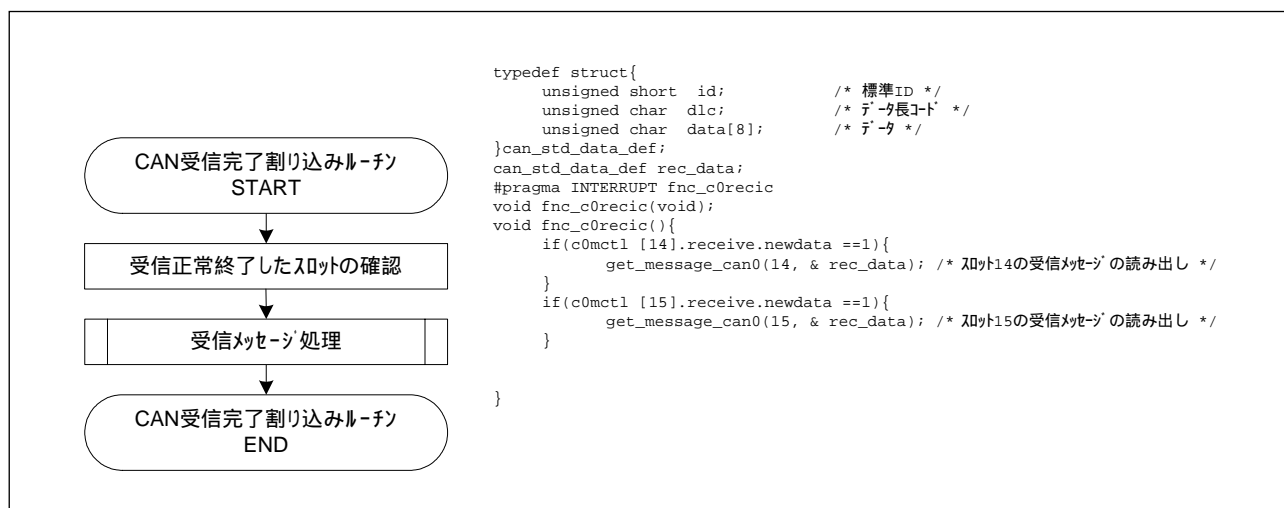


図 24 . 受信正常終了確認手順(CAN 受信完了割り込みを使用する場合)

4.3.2 リモートフレーム送信/データフレーム受信モード

スロットをリモートフレーム送信/データフレーム受信に設定すると、スロットに設定している ID、DLC のリモートフレームを送信後、自動的に同じ ID のデータフレームを受信できます。ただし、リモートフレームの送信完了前にスロットに設定している ID のデータフレームを受信した場合、リモートフレームは送信されません。

図 25 にポーリングを使用する場合のリモートフレーム送信/データフレーム受信手順を示します。

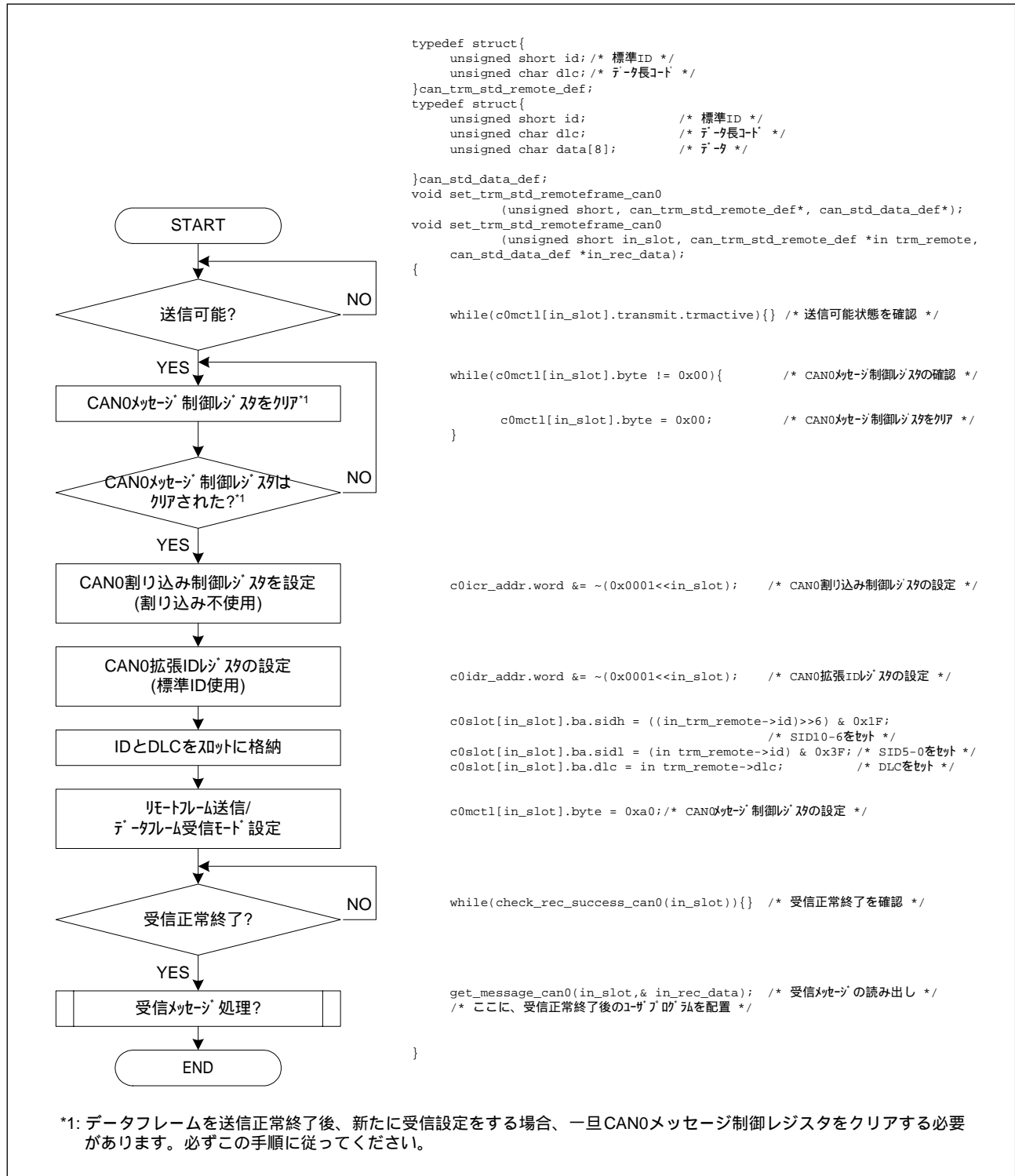


図 25 . リモートフレーム送信/データフレーム受信手順

#### 4.4 CAN オーバーランエラー

連続してメッセージを受信する場合、スロットの内容は、次の受信メッセージでオーバーライトされません。

CAN メッセージ制御レジスタの受信完了フラグをプログラムで"0"にするか、スロットへの受信要求をキャンセルする前に次のメッセージを受信した場合、オーバーライトフラグが"1"(メッセージはオーバーライトされた)になります。そして、新しく受信されたメッセージがそのスロットに格納されます。このようなときは以前のメッセージを再送信するように送信および受信側のプログラムで処理してください。

受信メッセージの処理手順については図 20を参照ください。

### 4.5 Basic CAN モード

CAN モジュール 1 チャンネルにつき 16 個のロットがあり、各ロットごとに送信または受信ロットの設定ができます。これを通常動作モードといいます。

Basic CAN モードは、ロット 14、15 が受信ロットで、ロット 0 からロット 13 が通常動作モードとして動作するモードです。このとき、受信メッセージはロット 14 からロット 15 と交互に格納されます。

通常動作モードでは、CAN メッセージ制御レジスタの設定によって、各ロットはデータフレームかリモートフレームどちらか一方のフレームタイプしか扱うことができません。しかし、Basic CAN モードでは、ロット 14、15 は両方のフレームタイプを同時に受信することができます。どちらのタイプを受信したかは、CAN メッセージ制御レジスタのリモートフレーム送受信ステータスフラグで判別でき、データフレームを受信したとき、リモートフレーム送受信ステータスフラグは"0"に、リモートフレームを受信したとき"1"になります。

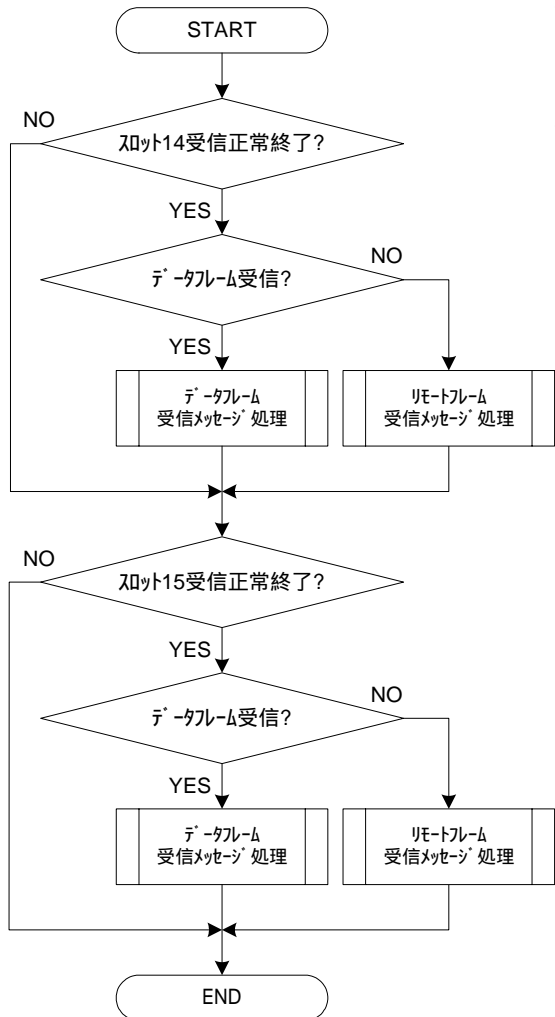
Basic CAN モードは、CAN 制御レジスタの BasicCAN モード選択ビットを"1"(Basic CAN モード)にすると選択できます。

Basic CAN モードを使用するときは、以下の点に注意してください。

- Basic CAN モードの設定はリセット/初期化モード時に行ってください。
- ロット 14 とロット 15 は受信ロットに設定してください。
- ロット 14 とロット 15 には同じ ID を設定してください。また、ローカルマスクレジスタ A、ローカルマスクレジスタ B の設定も同一にしてください。
- Basic CAN モード時も、メッセージを受信した状態で同じロットがさらにメッセージを受信すると、新しいメッセージでオーバーライトされます。

図 26 に Basic CAN モードの受信正常終了確認手順を示します。

(CAN0をBasic CANモードに設定した場合)



```

typedef struct{
    unsigned short id;           /* 標準ID */
    unsigned char dlc;          /* データ長 */
    unsigned char data[8];     /* データ */
}can_std_data_def;
can_std_data_def rec_data[2]; /* データフレーム受信 */
typedef struct{
    unsigned short id;           /* 標準ID */
    unsigned char dlc;          /* データ長 */
}can_std_remote_def;
can_std_remote_def rec_remote[2];
/* [0x0001]: received one dataframe */
/* [0x0002]: received two data frames */
/* [0x0010]: received one remoteframe */
/* [0x0020]: received two remoteframes */
int check_rec_success_basiccan_can0();
int check_rec_success_basiccan_can0(){
    static char cnt_data = 0; /* データフレーム受信のカウンタ */
    static char cnt_remote = 0; /* リモートフレーム受信のカウンタ */
    int result;
    /* Confirmation of mailbox 14 */
    if(c0mctl [14].receive.newdata == 1){
        /* データフレームを受信していた場合 */
        if(! c0mctl[14].receive.reactive){
            get_dataframe_can0(14, &rec_data[cnt_data++]);
        }
        /* リモートフレームを受信していた場合 */
        else{
            get_remoteframe_can0(14, &rec_remote[cnt_remote++]);
        }
    }
    /* mailbox 15を確認 */
    if(c0mctl [15].receive.newdata == 1){
        /* データフレームを受信していた場合 */
        if(! c0mctl[15].receive.reactive){
            get_dataframe_can0(15, &rec_data[cnt_data++]);
        }
        /* リモートフレームを受信していた場合 */
        else{
            get_remoteframe_can0(15, &rec_remote[cnt_remote++]);
        }
    }
    result = (cnt_remote<<4) + cnt_data;
    return result;
}
  
```

図 26 . Basic CAN モードの受信正常終了確認手順

## 5. CAN エラー

CAN エラーによる割り込みがあり、この CAN エラー割り込み中で通信エラーが発生したことを確認できます。

スロットが送信中または受信中に通信フレームの異常がありエラーを検知すると、送受信の状態によって送信エラーカウンタ値、または受信エラーカウンタ値が増加します。送信エラーカウンタ値、または受信エラーカウンタ値が 128 以上になった時、CAN ステータスはエラーアクティブ状態からエラーパッシブ状態になります。送信エラーカウンタ値が 256 以上になった時、バスオフ状態になります。

CAN 制御レジスタのバスエラー割り込み許可ビットが"1"(バスエラー割り込み許可)の場合、エラーを検知するごとに CAN エラー割り込みが発生します。

CAN 制御レジスタのバスエラー割り込み許可ビットが"0"(バスエラー割り込み禁止)の場合、エラーパッシブ状態となった時、およびバスオフ状態になった時に、CAN エラー割り込みが発生します。

バスエラー割り込み許可ビットの設定は、必ず CAN コンフィグレーション中に行ってください。CAN エラー割り込みを使用する場合、事前に CAN エラー割り込み制御レジスタの設定が必要です。

図 27に CAN エラー割り込み制御レジスタ設定手順を示します。

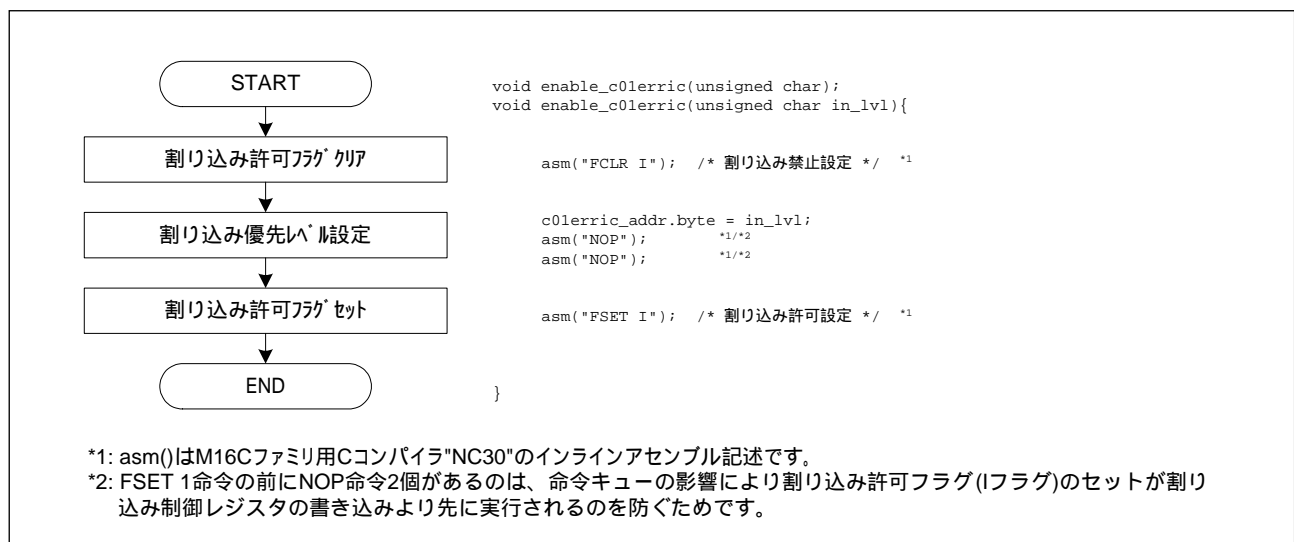


図 27 . CAN エラー割り込み制御レジスタ設定手順

5.1 CAN エラー確認手順

CAN のエラー状態を確認するには、ポーリングを使用する方法と割り込みを使用する方法があります。

ポーリングを使用する場合

CAN ステータスレジスタのバスエラー状態フラグ、エラーパッシブ状態フラグ、バスオフステートフラグをポーリングすることでエラー状態を確認できます。

図 28 にポーリングを使用する場合の CAN エラー状態確認手順を示します。

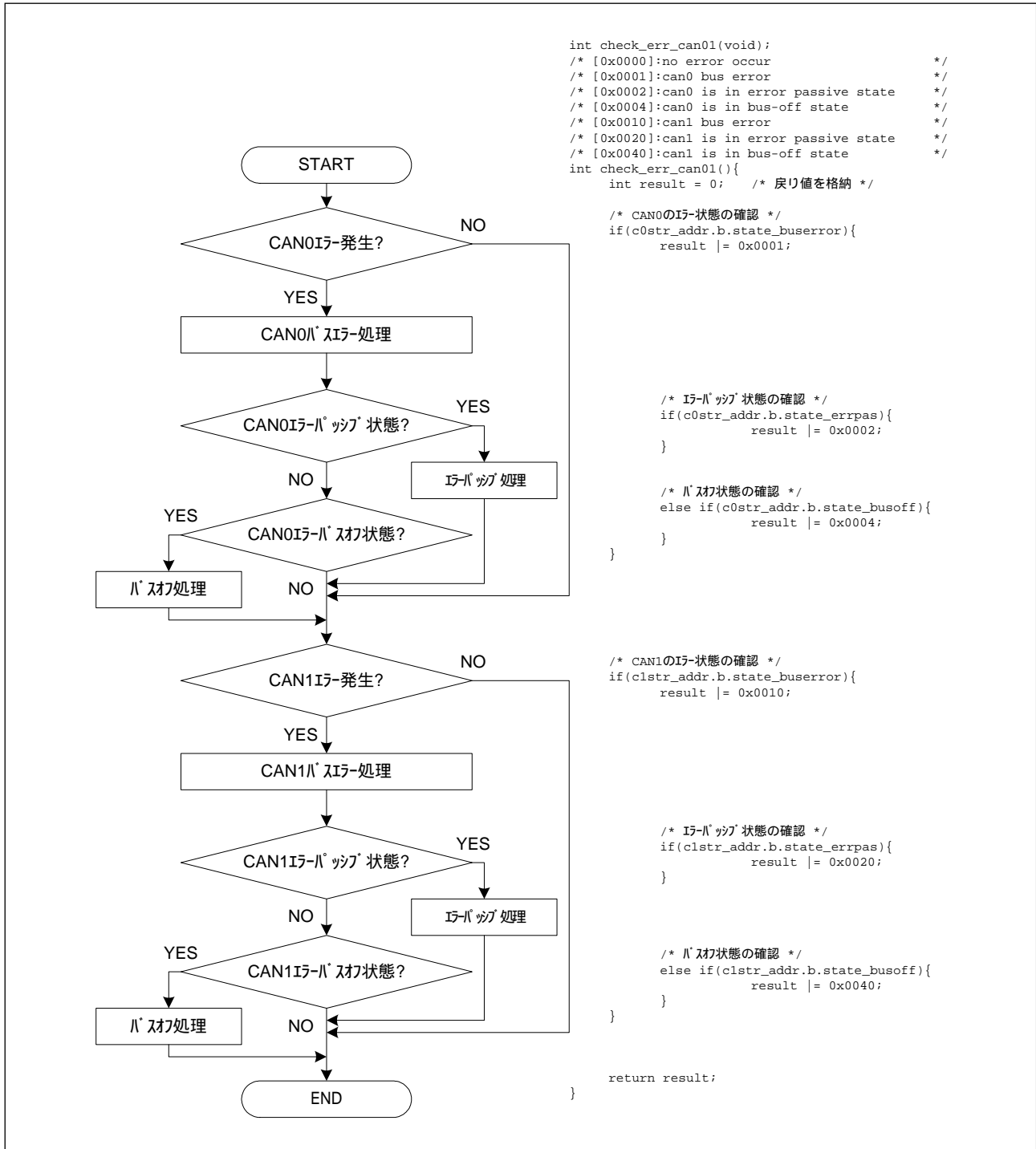


図 28 . CAN エラー状態確認手順(ポーリングを使用する場合)

CAN エラー割り込みを使用する場合

CAN エラー割り込み制御レジスタを割り込み許可にすると、CAN エラー割り込みが使用できます。この CAN エラー割り込みルーチン内で CAN エラー状態を確認できます。この CAN エラー割り込みは、CAN0 と CAN1 で兼用です。

図 29 に CAN エラー割り込みを使用する場合の CAN エラー確認手順を示します。

ここでは、CAN エラー割り込みルーチン内でエラー状態を判定し、バスオフ状態の場合にはリターンフロムバスオフを実行しています。

リターンフロムバスオフについては、5.2項を参照ください。

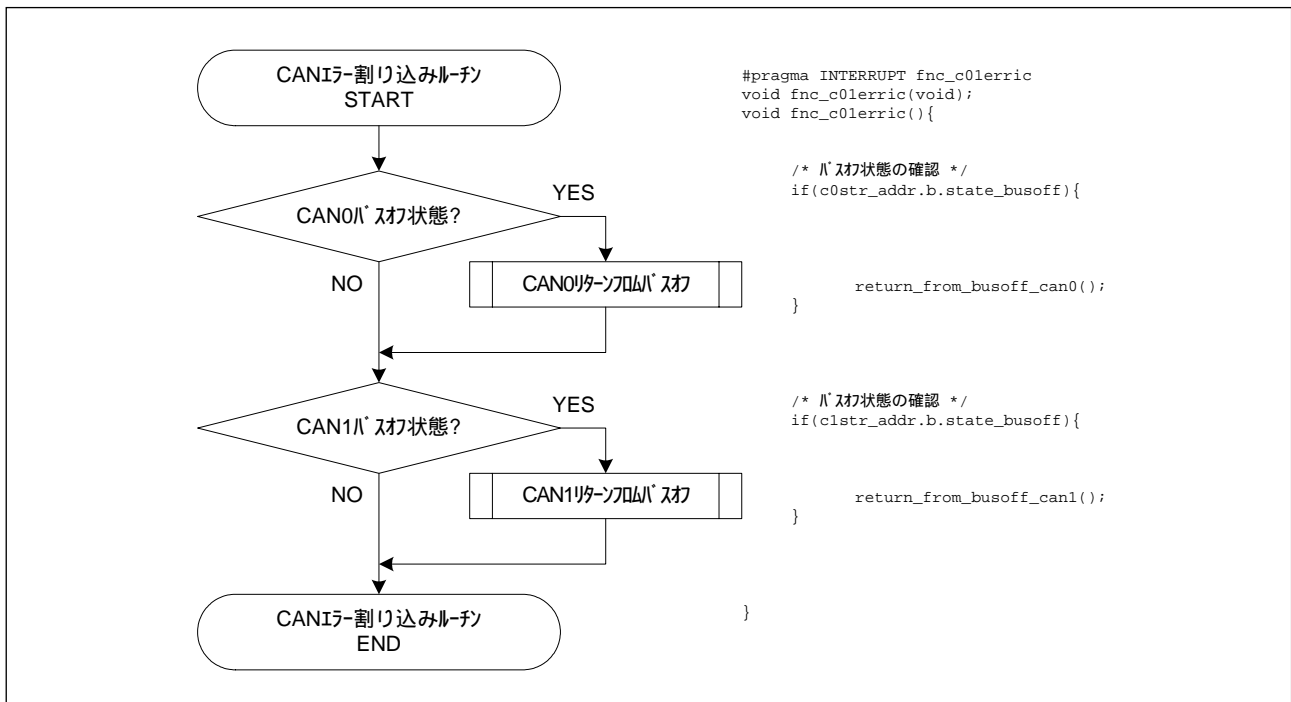


図 29 . CAN エラー確認手順(CAN エラー割り込みを使用する場合)



5.2 リターンフロムバスオフ機能

CAN ステータスがバスオフ状態になると送受信を行えなくなります。送受信を行うことができるエラーアクティブ状態になるためには、バス上に 11 ビット連続のレセシブビットを 128 回検出するまで待たなければいけません。しかし、緊急にメッセージを送信したいときなどでは、即座にエラーアクティブ状態にしたい場合があります。そこで上記の期間を待つことなく即座にエラーアクティブ状態にすることができる、リターンフロムバスオフ機能があります。

リターンフロムバスオフ機能は、CAN 制御レジスタのバスオフからの強制復帰命令ビットを"1"(バスオフからの強制復帰)にすると実行されます。バスオフからの強制復帰命令ビットは、リターンフロムバスオフが実行された後に自動的にクリアされます。このリターンフロムバスオフを実行する場合、エラーアクティブ状態になった後、CAN のコンフィグレーションを行う必要はありません。

リターンフロムバスオフ機能は、CAN がバスオフ状態の時のみ有効です。エラーアクティブ状態、エラーパッシブ状態でリターンフロムバスオフを実行した場合は無効となり、バスオフからの強制復帰命令ビットは即座にクリアされます。

図 30 にリターンフロムバスオフ機能の実行例を示します。

ここでは、CAN エラー割り込みルーチン内で CAN0 および CAN1 のエラー状態を判定し、バスオフ状態の場合にリターンフロムバスオフを実行しています。

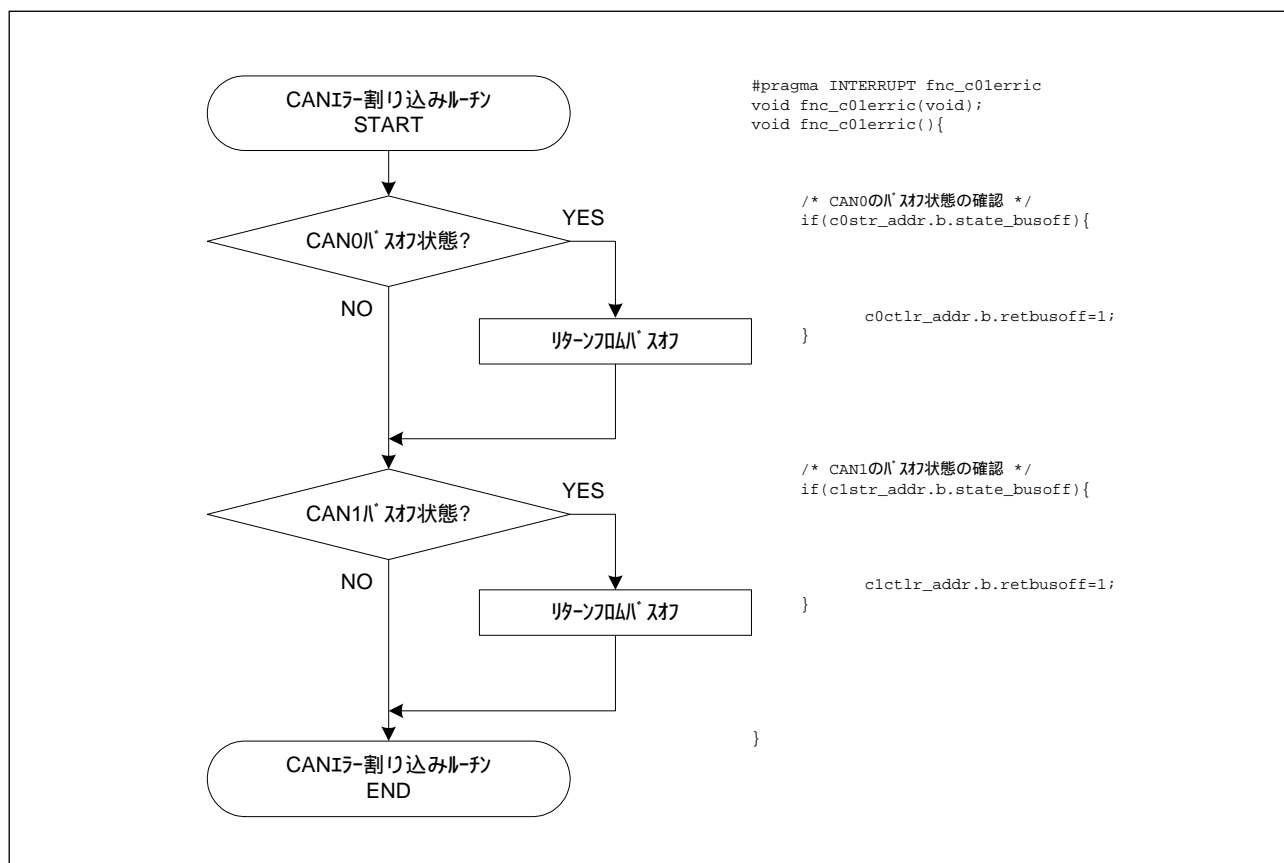


図 30 . リターンフロムバスオフ機能の実行例

## 6. アクセプタンスフィルタの使い方

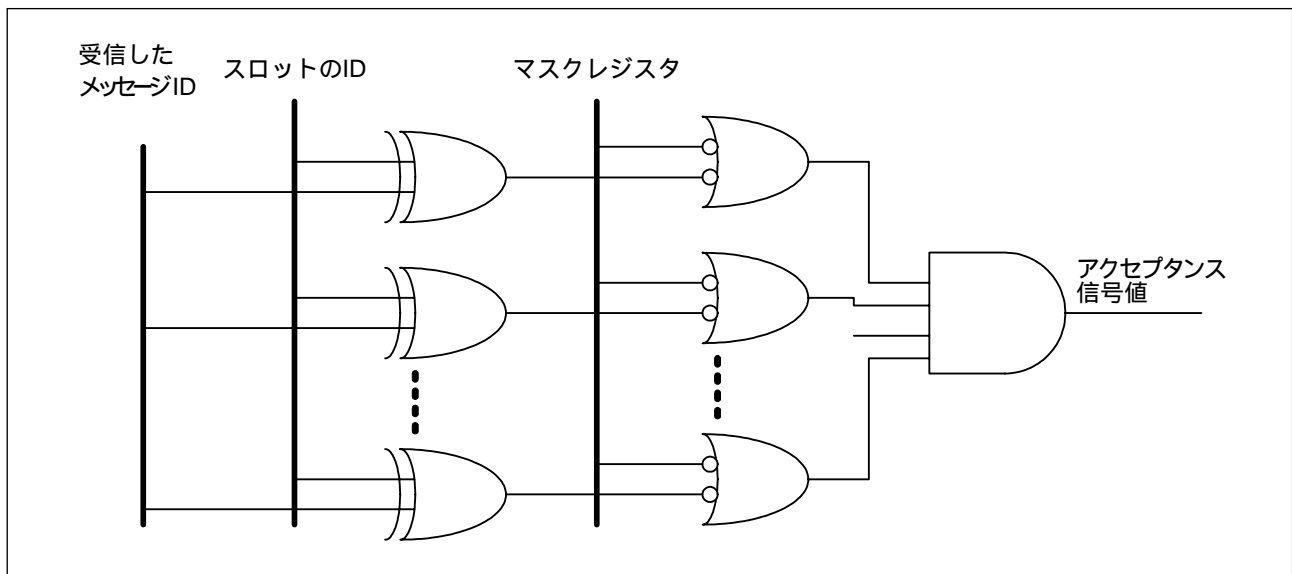
ハードウェアでメッセージの受信、破棄を行うアクセプタンスフィルタがあります。

### 6.1 アクセプタンスフィルタ(ACP)

アクセプタンスフィルタは、グローバルマスクレジスタ(スロット 0~13 用)、ローカルマスクレジスタ A(スロット 14 用)、ローカルマスクレジスタ B(スロット 15 用)を使用してフィルタリングを行います。

#### (1) アクセプタンスフィルタのレジスタ構成

図 31 に ID とマスクレジスタの構成を、図 32、図 33 にメモリ配置とビットマップを示します。



スロットの ID	受信したいメッセージ ID の値を設定
マスクレジスタ	0:受信したメッセージ ID の対応するビットを無視する 1:受信したメッセージ ID の対応するビットと、スロットの ID の対応するビットを比較する
アクセプタンス信号値	0:メッセージを破棄する 1:メッセージを受信する

図 31 . ID とマスクレジスタの構成

b7	b6	b5	b4	b3	b2	b1	b0	CAN0	CAN1 <sup>*1</sup>	
			SID10	SID9	SID8	SID7	SID6	0160 <sub>16</sub>	0360 <sub>16</sub>	C0GMR C1GMR
		SID5	SID4	SID3	SID2	SID1	SID0	0161 <sub>16</sub>	0361 <sub>16</sub>	
				EID17	EID16	EID15	EID14	0162 <sub>16</sub>	0362 <sub>16</sub>	
EID13	EID12	EID11	EID10	EID9	EID8	EID7	EID6	0163 <sub>16</sub>	0363 <sub>16</sub>	
		EID5	EID4	EID3	EID2	EID1	EID0	0164 <sub>16</sub>	0364 <sub>16</sub>	
			SID10	SID9	SID8	SID7	SID6	0166 <sub>16</sub>	0366 <sub>16</sub>	C0LMAR C1LMAR
		SID5	SID4	SID3	SID2	SID1	SID0	0167 <sub>16</sub>	0367 <sub>16</sub>	
				EID17	EID16	EID15	EID14	0168 <sub>16</sub>	0368 <sub>16</sub>	
EID13	EID12	EID11	EID10	EID9	EID8	EID7	EID6	0169 <sub>16</sub>	0369 <sub>16</sub>	
		EID5	EID4	EID3	EID2	EID1	EID0	016A <sub>16</sub>	036A <sub>16</sub>	
			SID10	SID9	SID8	SID7	SID6	016C <sub>16</sub>	036C <sub>16</sub>	C0LMBR C1LMBR
		SID5	SID4	SID3	SID2	SID1	SID0	016D <sub>16</sub>	036D <sub>16</sub>	
				EID17	EID16	EID15	EID14	016E <sub>16</sub>	036E <sub>16</sub>	
EID13	EID12	EID11	EID10	EID9	EID8	EID7	EID6	016F <sub>16</sub>	036F <sub>16</sub>	
		EID5	EID4	EID3	EID2	EID1	EID0	0170 <sub>16</sub>	0370 <sub>16</sub>	

図 32 . バイトアクセス時<sup>2</sup>のメモリ配置とビットマップ

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	CAN0	CAN1 <sup>*1</sup>	
			SID10	SID9	SID8	SID7	SID6			SID5	SID4	SID3	SID2	SID1	SID0	0160 <sub>16</sub>	0360 <sub>16</sub>	C0GMR C1GMR
				EID17	EID16	EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	EID7	EID6	0162 <sub>16</sub>	0362 <sub>16</sub>	
		EID5	EID4	EID3	EID2	EID1	EID0									0164 <sub>16</sub>	0364 <sub>16</sub>	
			SID10	SID9	SID8	SID7	SID6			SID5	SID4	SID3	SID2	SID1	SID0	0166 <sub>16</sub>	0366 <sub>16</sub>	
				EID17	EID16	EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	EID7	EID6	0168 <sub>16</sub>	0368 <sub>16</sub>	C0LMAR C1LMAR
		EID5	EID4	EID3	EID2	EID1	EID0									016A <sub>16</sub>	036A <sub>16</sub>	
			SID10	SID9	SID8	SID7	SID6			SID5	SID4	SID3	SID2	SID1	SID0	016C <sub>16</sub>	036C <sub>16</sub>	
				EID17	EID16	EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	EID7	EID6	016E <sub>16</sub>	036E <sub>16</sub>	
		EID5	EID4	EID3	EID2	EID1	EID0									0170 <sub>16</sub>	0370 <sub>16</sub>	C0LMBR C1LMBR

図 33 . ワードアクセス時<sup>3</sup>のメモリ配置とビットマップ

\*1: 1N のグローバルマスクレジスタ、ローカルマスクレジスタ A、ローカルマスクレジスタ B は CAN1 のアドレスに配置されています。

\*2: CAN 制御レジスタのメッセージオーダー選択ビットが"1"の場合

\*3: CAN 制御レジスタのメッセージオーダー選択ビットが"0"の場合

## (2) アクセプタンスフィルタの使用例

## 使用例 1

表 4にスロット 0 が ID123<sub>16</sub> の標準データフレーム、または標準リモートフレームを受信する場合の各レジスタ設定を示します。

表 4. アクセプタンスフィルタの使用例 1

		SID <sub>10-6</sub>	SID <sub>5-0</sub>	EID <sub>17-14</sub>	EID <sub>13-6</sub>	EID <sub>5-0</sub>
スロット 0		0 0 1 0 0	1 0 0 0 1 1	XXXX	XXXXXXXXXX	XXXXXX
レジスタ	COGMR	1 1 1 1 1	1 1 1 1 1 1	XXXX	XXXXXXXXXX	XXXXXX
受信メッセージ	ID123 <sub>16</sub>	0 0 1 0 0	1 0 0 0 1 1			

## 使用例 2

表 5にスロット 0 が ID122<sub>16</sub>、123<sub>16</sub> の 2 個の標準データフレーム、または標準リモートフレームを受信する場合の各レジスタの設定を示します。

表 5. アクセプタンスフィルタの使用例 2

		SID <sub>10-6</sub>	SID <sub>5-0</sub>	EID <sub>17-14</sub>	EID <sub>13-6</sub>	EID <sub>5-0</sub>
スロット 0		0 0 1 0 0	1 0 0 0 1 X	XXXX	XXXXXXXXXX	XXXXXX
レジスタ	COGMR	1 1 1 1 1	1 1 1 1 1 0	XXXX	XXXXXXXXXX	XXXXXX
受信メッセージ	ID122 <sub>16</sub>	0 0 1 0 0	1 0 0 0 1 0			
	ID123 <sub>16</sub>	0 0 1 0 0	1 0 0 0 1 1			

## 使用例 3

表 6にスロット 0 が ID12345678<sub>16</sub> の拡張データフレーム、または拡張リモートフレームを受信する場合の各レジスタの設定を示します。

表 6. アクセプタンスフィルタの使用例 3

		SID <sub>10-6</sub>	SID <sub>5-0</sub>	EID <sub>17-14</sub>	EID <sub>13-6</sub>	EID <sub>5-0</sub>
スロット 0		1 0 0 1 0	0 0 1 1 0 1	0 0 0 1	01011001	111000
レジスタ	COGMR	1 1 1 1 1	1 1 1 1 1 1	1 1 1 1	11111111	111111
受信メッセージ	ID12345678 <sub>16</sub>	1 0 0 1 0	0 0 1 1 0 1	0 0 0 1	01011001	111000

## 6.2 アクセプタンスフィルタサポートユニット(ASU)

アクセプタンスフィルタサポートユニットは、受信 ID の有効、無効をテーブル検索で判断する機能です。まず受信する ID をデータテーブルに登録します。次に受信した ID を CAN アクセプタンスフィルタサポートレジスタに格納し、デコードされた受信 ID を CAN アクセプタンスフィルタサポートレジスタから読み出しテーブル検索を行います。このアクセプタンスフィルタサポートユニットは、標準フレームの ID に対してのみ使用することができます。

アクセプタンスフィルタサポートユニットは、以下のような場合に有効です。

- アクセプタンスフィルタにて受信する ID にマスクをかけることができない場合  
(例:受信する ID 078<sub>16</sub>、087<sub>16</sub>、111<sub>16</sub>)
- 受信する ID が非常に多く、ソフトウェアでフィルタリングすると時間がかかりすぎる場合

### 6.2.1 アクセプタンスフィルタサポートユニットの使用方法

受信する ID が 000<sub>16</sub>、00D<sub>16</sub>、6F3<sub>16</sub>、6F4<sub>16</sub>、6FF<sub>16</sub> の場合のアクセプタンスフィルタサポートユニットの使用方法を示します。

#### (1) データテーブルの設定

ROM または RAM に受信する ID を登録したデータテーブルを用意します。データテーブルを配置するアドレスは任意です。

データテーブルは、縦軸に受信する ID の上位 8 ビット(SID<sub>10-3</sub>)の値を、横軸に受信する ID の下位 3 ビット(SID<sub>2-0</sub>)を 8 ビットにデコードした値をとり、対応するビットには"1"を、それ以外のビットには"0"を設定します。

#### (2) CAN アクセプタンスフィルタサポートレジスタへの書き込み

CAN0 が受信した場合は、CAN0 アクセプタンスフィルタサポートレジスタに、CAN1 が受信した場合は、CAN1 アクセプタンスフィルタサポートレジスタに書き込みます。

#### (3) CAN アクセプタンスフィルタサポートレジスタの読み出し

CAN アクセプタンスフィルタサポートレジスタより、受信 ID の上位 8 ビット(SID<sub>10-3</sub>)の値と、下位 3 ビットを 8 ビットにデコードした値を読み出します。

#### (4) 受信した ID の有効無効の判定

(3)で CAN アクセプタンスフィルタサポートレジスタから読み出した値を使用して、(1)で設定したデータテーブルを検索し、メッセージの有効、無効を判定します。

図 34 にアクセプタンスフィルタサポートレジスタ構成、図 35 にデータテーブル構成、図 36 にアクセプタンスフィルタサポートレジスタへの書き込み、読み出し時の状態を示します。

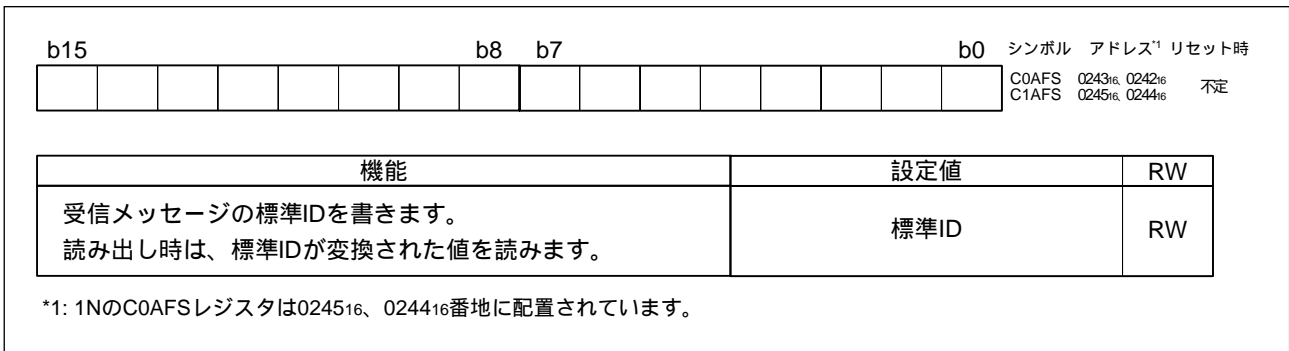


図 34 . アクセプタンスフィルタサポートレジスタ構成

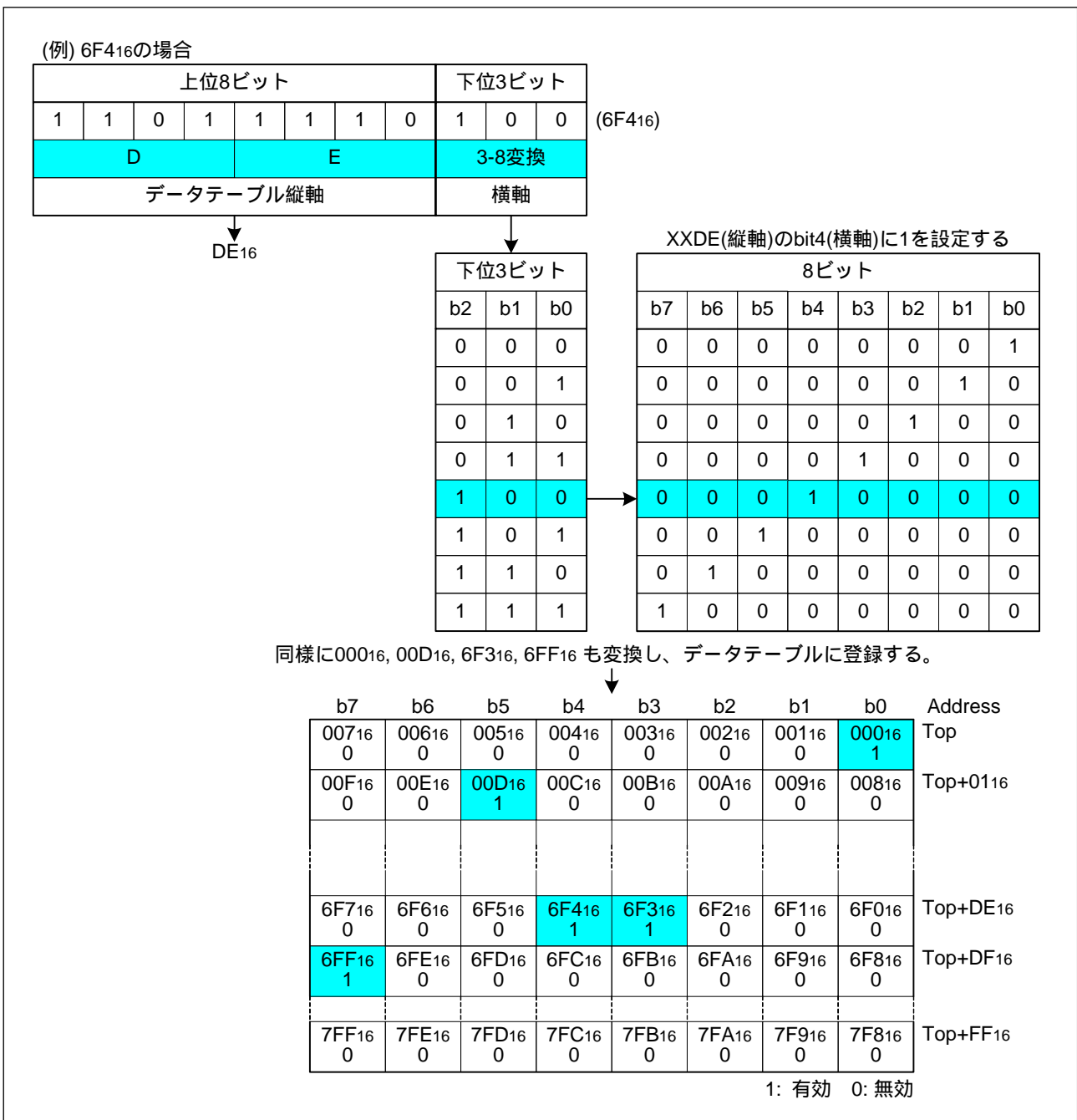


図 35 . データテーブル構成

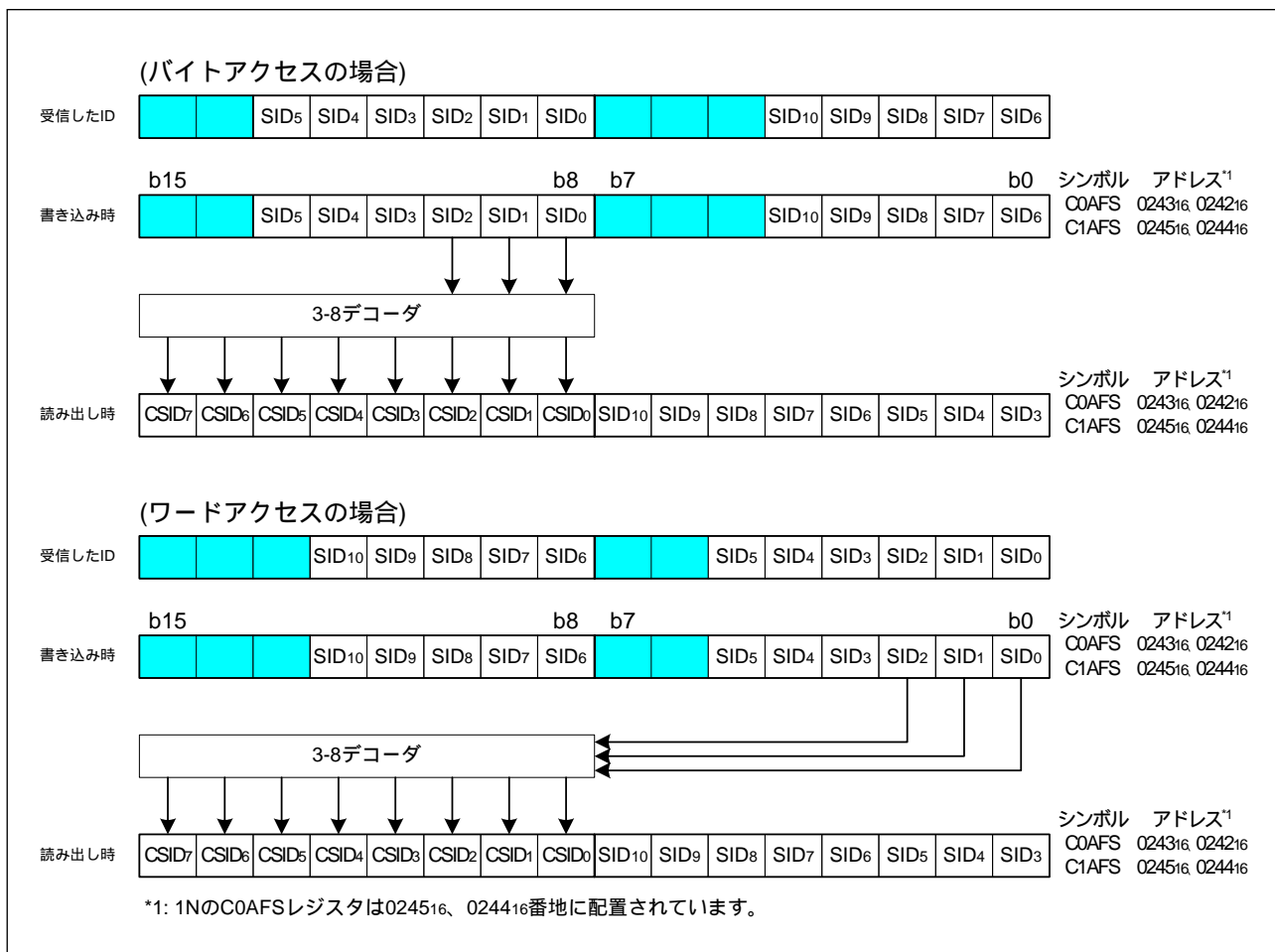


図 36 . アクセプタンスフィルタサポートレジスタへの書き込み、読み出し時の状態

図 37にアクセプタンスフィルタサポートユニット使用手順を示します。  
ここでは、CAN 受信完了割り込みルーチン内で受信 ID の有効、無効を判定しています。

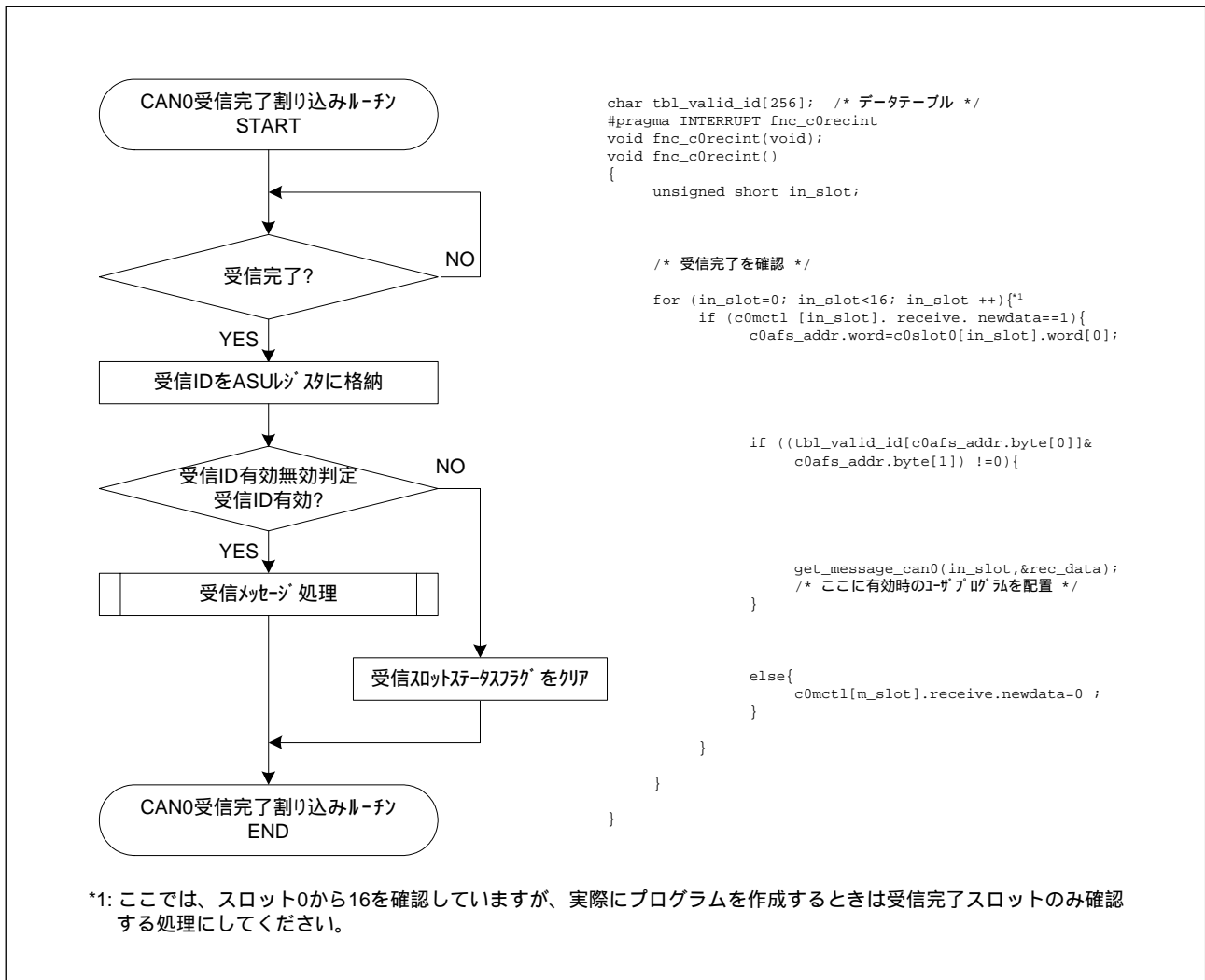


図 37 . アクセプタンスフィルタサポートユニット使用手順



## 7. CAN スリープ動作および CAN ウェイクアップ動作

### 7.1 CAN のスリープ動作

CAN モジュールがスリープモード時は、CAN モジュールへ供給されるクロックが停止するため、CAN モジュールは一切動作しません。CAN モジュールを使用しない場合、消費電流を下げるために CAN モジュールをスリープモードにすることを推奨します。

CAN モジュールをスリープモードへ移行させる前に、必ず CAN モジュールをリセットしてください。

図 38に CAN モジュールのスリープモードへの移行手順を示します。

この状態で、マイクロコンピュータをウェイトモードまたはストップモードにすると、さらに消費電流を下げるすることができます。

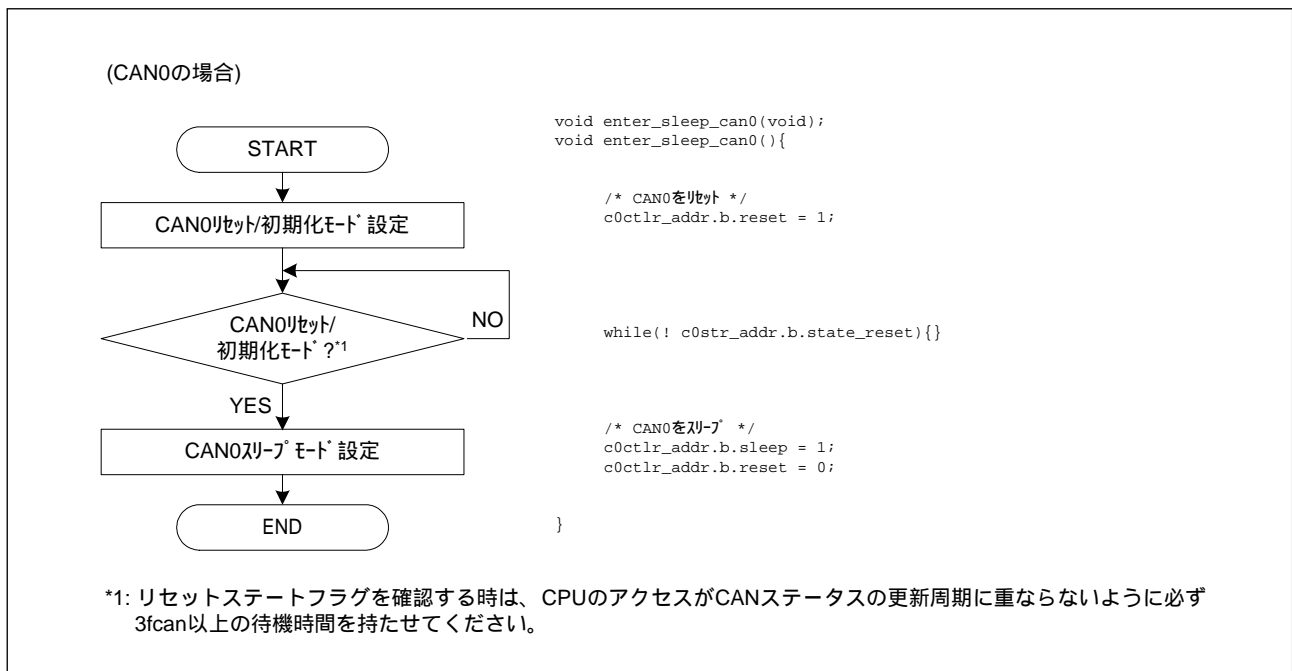


図 38 . CAN モジュールのスリープモード移行手順

## 7.2 CAN ウェイクアップ動作

CAN モジュールがスリープモード時、CAN 受信ラインの立ち下がりが発生する CAN ウェイクアップ割り込みでスリープモードから復帰することができます。CAN ウェイクアップは、CAN ウェイクアップ割り込み制御レジスタを割り込み許可にすると使用できます。

CAN ウェイクアップ割り込みは、CAN0 と CAN1 で兼用です。6N4、6NK、6NM グループは、割り込み要因選択レジスタ 0 の IFSR02 ビットを"1"にすることで CAN0 と CAN1 を個別に設定できます。CAN ウェイクアップ割り込みを使用して CAN をウェイクアップさせる場合は、この割り込みルーチン内で CAN のコンフィグレーションを行ってください。

図 39 に CAN ウェイクアップ手順を示します。

ここでは、CAN ウェイクアップ割り込みルーチンの中で、CAN0 と CAN1 の両方の CAN を動作モードへ復帰させています。

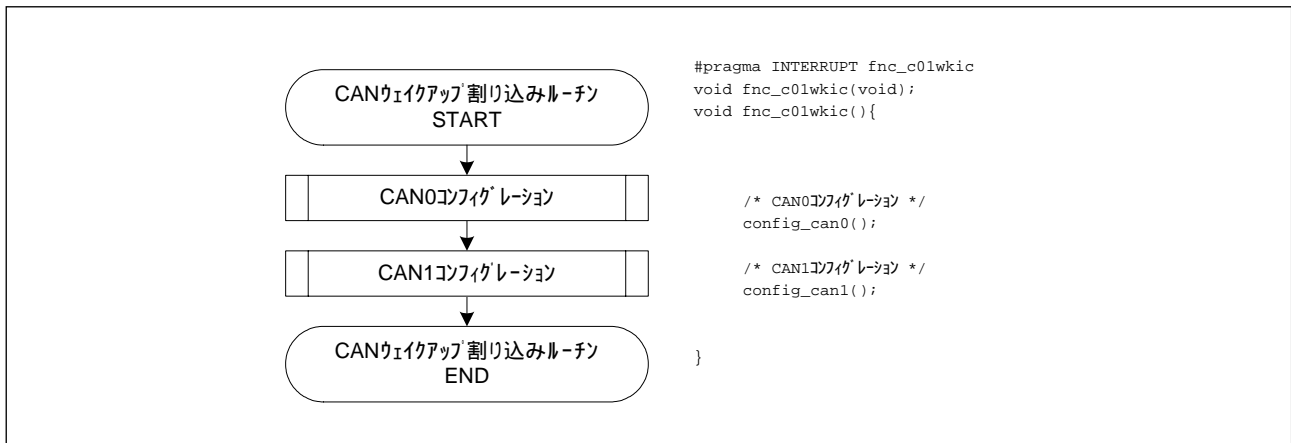


図 39 . CAN ウェイクアップ手順

8. サンプルプログラムに関する注意事項

8.1 各レジスタのシンボル表記

この資料のサンプルプログラムに記載している各レジスタのシンボルは、ルネサス標準 C 言語 SFR ヘッダファイルの表記に基づいています。

8.2 while の無限ループ

表記を簡略化するためにサンプルプログラム中に while で無限ループを組んでいる個所があります。実際にプログラムを作成するときは、各 while ループに制限時間を持たせ、オーバertime時に抜けるような処理にしてください。

図 40に while ループの制限時間を持たせた場合の処理例を示します。

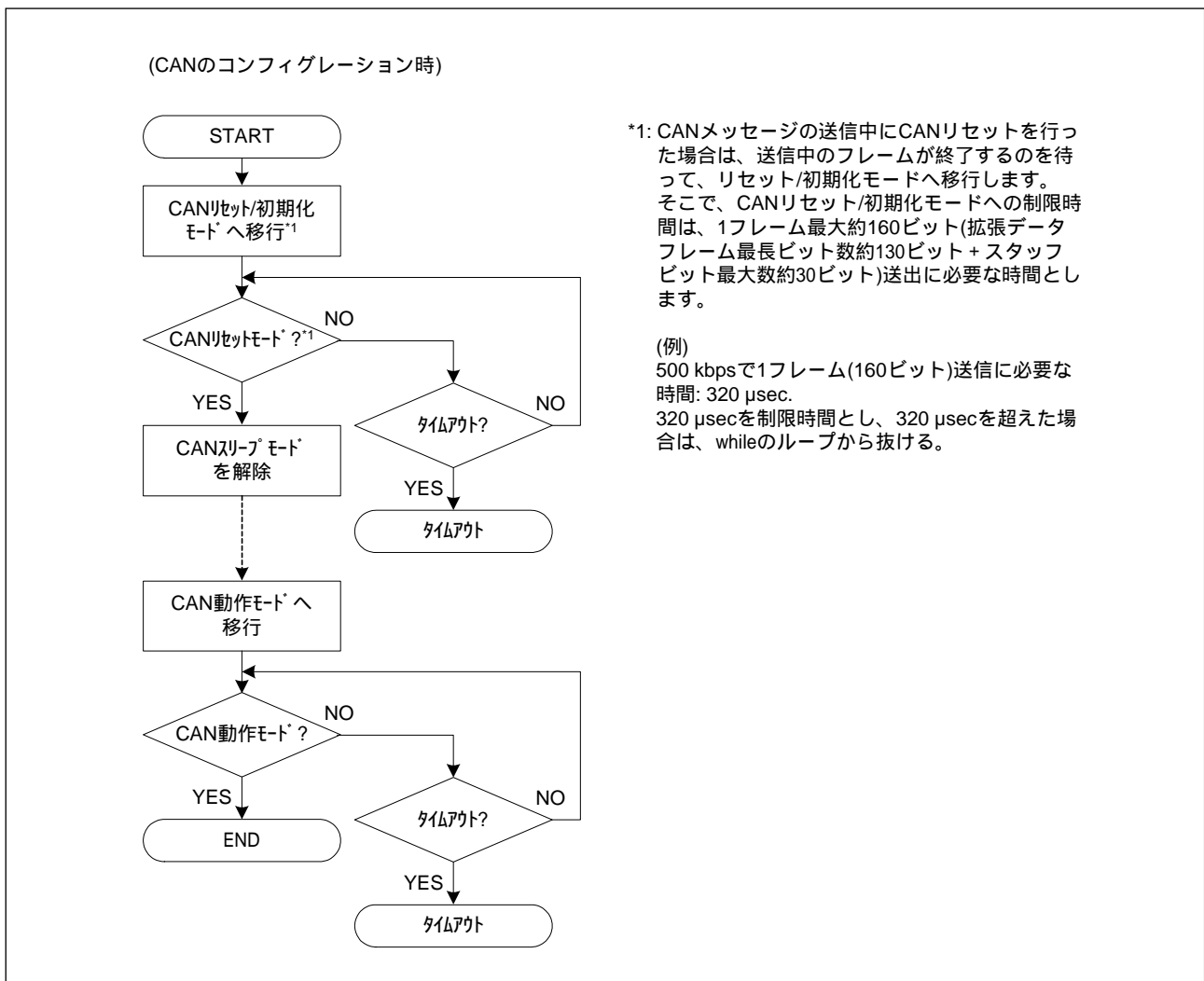


図 40 . while ループの制限時間を持たせた場合の処理例

## 9. 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://japan.renesas.com/>

CAN マイコンに関する技術的なお問い合わせ先  
カスタマ・サポート・センター：csc@renesas.com

改訂記録

M16C/6N、M16C/1N、M16C/29、R8C/22,23 グループ  
CAN アプリケーションノート

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2003.11.05	-	初版発行
2.00	2004.05.18	-	フォームの更新
		-	M16C/29 グループに対応 全文、文言および表現の統一
		1	1. 要約を一部改訂
		1	2. 適用を追加
		2	3. 初期設定を一部改訂
		3	3.1 CAN ビットタイミングを一部改訂
		5	図 2 を一部改訂
		7	3.3 項題を変更、内容を一部改訂
			図 5 を一部改訂
			表 1 を一部改訂
		9	図 6 を一部改訂
		10	表 3 を一部改訂
		11	図 7 を一部改訂
		13	図 8 を一部改訂
		22	図 16 を一部改訂
		23	図 17 を一部改訂
		28	図 20 を一部改訂
		29	図 21 を一部改訂
		30	図 23 を一部改訂
		32	図 25 を一部改訂
34	4.5 BasicCAN モードを一部改訂		
35	図 26 を一部改訂		
36	5. CAN エラーを一部改訂		
39	図 30 を一部改訂		
41	図 32、33 を一部改訂		
44	図 34 を一部改訂		
45	図 36 を一部改訂		
46	図 37 を一部改訂		
47	7.1 CAN スリープ動作を一部改訂		
50	参考ドキュメントを削除		
2.01	2004.07.22	1	M16C/6NK、6NL、6NM、6NN グループを適用品種に追加
		7	表 1 を一部改訂
		28	図 20 を一部改訂
		33	4.4 CAN オーバーランエラーを一部改訂
		44	図 34 の*1、図 35 を一部改訂
		45	図 36 を一部改訂
48	7.2 CAN ウェイクアップ動作を一部改訂		

改訂記録	M16C/6N、M16C/1N、M16C/29、R8C/22,23 グループ CAN アプリケーションノート
------	---

Rev.	発行日	改訂内容	
		ページ	ポイント
2.02	2005.06.08	18	図 13 のフロー図を変更
		19	図 14 のフロー図を変更
		26	図 19 のフロー図を変更
3.00	2005.10.26	1	R8C/22,23 グループを適用品種に追加

### 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留意ください。

### 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。