

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。

お客様各位

資料中の「日立製作所」、「日立XX」等名称の株式会社ルネサス テクノロジへの変更について

2003年4月1日を以って三菱電機株式会社及び株式会社日立製作所のマイコン、ロジック、アナログ、ディスクリート半導体、及びDRAMを除くメモリ(フラッシュメモリ・SRAM等)を含む半導体事業は株式会社ルネサス テクノロジに承継されました。従いまして、本資料中には「日立製作所」、「株式会社日立製作所」、「日立半導体」、「日立XX」といった表記が残っておりますが、これらの表記は全て「株式会社ルネサス テクノロジ」に変更されておりますのでご理解の程お願い致します。尚、会社商標・ロゴ・コーポレートステートメント以外の内容については一切変更しておりませんので資料としての内容更新ではありません。

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

2003年4月1日
株式会社ルネサス テクノロジ
カスタマサポート部

ご注意

安全設計に関するお願い

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

本資料ご利用に際しての留意事項

1. 本資料は、お客様が用途に応じた適切なルネサス テクノロジ製品をご購入いただくための参考資料であり、本資料中に記載の技術情報についてルネサス テクノロジが所有する知的財産権その他の権利の実施、使用を許諾するものではありません。
2. 本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他応用回路例の使用に起因する損害、第三者所有の権利に対する侵害に関し、ルネサス テクノロジは責任を負いません。
3. 本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他全ての情報は本資料発行時点のものであり、ルネサス テクノロジは、予告なしに、本資料に記載した製品または仕様を変更することがあります。ルネサス テクノロジ半導体製品のご購入に当たりましては、事前にルネサス テクノロジ、ルネサス販売または特約店へ最新の情報をご確認頂きますとともに、ルネサス テクノロジホームページ (<http://www.renesas.com>)などを通じて公開される情報に常にご注意ください。
4. 本資料に記載した情報は、正確を期すため、慎重に制作したのですが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、海中継用機器あるいはシステムなど、特殊用途へのご利用をご検討の際には、ルネサス テクノロジ、ルネサス販売または特約店へご照会ください。
7. 本資料の転載、複製については、文書によるルネサス テクノロジの事前の承諾が必要です。
8. 本資料に関し詳細についてのお問い合わせ、その他お気付きの点がございましたらルネサス テクノロジ、ルネサス販売または特約店までご照会ください。

H8Sシリーズ内蔵HCAN

アプリケーションノート

ルネサスシングルチップマイクロコンピュータ

ご注意

- 1 本書に記載の製品及び技術のうち「外国為替及び外国貿易法」に基づき安全保障貿易管理関連貨物・技術に該当するものを輸出する場合、または国外に持ち出す場合は日本国政府の許可が必要です。
- 2 本書に記載された情報の使用に際して、弊社もしくは第三者の特許権、著作権、商標権、その他の知的所有権等の権利に対する保証または実施権の許諾を行うものではありません。また本書に記載された情報を使用した事により第三者の知的所有権等の権利に関わる問題が生じた場合、弊社はその責を負いませんので予めご了承ください。
- 3 製品及び製品仕様は予告無く変更する場合がありますので、最終的な設計、ご購入、ご使用に際しましては、事前に最新の製品規格または仕様書をお求めになりご確認ください。
- 4 弊社は品質・信頼性の向上に努めておりますが、宇宙、航空、原子力、燃焼制御、運輸、交通、各種安全装置、ライフサポート関連の医療機器等のように、特別な品質・信頼性が要求され、その故障や誤動作が直接人命を脅かしたり、人体に危害を及ぼす恐れのある用途にご使用をお考えのお客様は、事前に弊社営業担当迄ご相談をお願い致します。
- 5 設計に際しては、特に最大定格、動作電源電圧範囲、放熱特性、実装条件及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いただきますようお願い致します。保証値を越えてご使用された場合の故障及び事故につきましては、弊社はその責を負いません。
また保証値内のご使用であっても半導体製品について通常予測される故障発生率、故障モードをご考慮の上、弊社製品の動作が原因でご使用機器が人身事故、火災事故、その他の拡大損害を生じないようにフェールセーフ等のシステム上の対策を講じて頂きますようお願い致します。
- 6 本製品は耐放射線設計をしておりません。
- 7 本書の一部または全部を弊社の文書による承認なしに転載または複製することを堅くお断り致します。
- 8 本書をはじめ弊社半導体についてのお問い合わせ、ご相談は弊社営業担当迄お願い致します。

はじめに

本アプリケーションノートでは、H8Sシリーズ内蔵HCAN (日立コントローラエリアネットワーク) を使用した場合の動作例を説明しています。

(ただし、本文中のプログラム例は、H8S/2623内蔵のHCANについて記述しています。)

本アプリケーションノートに掲載されているプログラム、回路例等の動作は確認しておりますが、実際にご使用になる場合は、改めて動作確認のうえ、ご使用くださいますようお願いいたします。

- 目 次 -

1. HCAN送受信 (例題 1.):スタンダードフォーマット、1バイトデータ	
1.1 仕様	3
1.2 使用機能説明	5
1.3 動作説明	6
1.4 ソフトウェア説明	8
1.5 フローチャート	9
1.6 プログラムリスト	11
1.7 備考	13
2. HCAN送受信 (例題 2.):スタンダードフォーマット、8バイトデータ、DTC使用	
2.1 仕様	19
2.2 使用機能説明	21
2.3 動作説明	22
2.4 ソフトウェア説明	24
2.5 フローチャート	26
2.6 プログラムリスト	30
2.7 備考	33
3. HCAN送受信 (例題 3.):エクステンドフォーマット、1バイトデータ	
3.1 仕様	39
3.2 使用機能説明	41
3.3 動作説明	42
3.4 ソフトウェア説明	44
3.5 フローチャート	46
3.6 プログラムリスト	50
3.7 備考	53
4. HCAN送受信 (例題 4.):スタンダードフォーマット、8バイトデータ、優先順位付き	
4.1 仕様	59
4.2 使用機能説明	61
4.3 動作説明	63
4.4 ソフトウェア説明	65
4.5 フローチャート	70
4.6 プログラムリスト	74
4.7 備考	81
5. HCAN送受信 (例題 5.):レポートフレーム	
5.1 仕様	87
5.2 使用機能説明	89
5.3 動作説明	90
5.4 ソフトウェア説明	92
5.5 フローチャート	93
5.6 プログラムリスト	95
5.7 備考	97

1.HCAN送受信 (例題 1.):スタンダードフォーマット、1バイトデータ

第1章 目次

1.1 仕様	3
1.2 使用機能説明	5
1.3 動作説明	6
1.4 ソフトウェア説明	8
1.5 フローチャート	9
1.6 プログラムリスト	11
1.7 備考	13

1 HCAN送受信(例題 1 .)	対象	H8S/2623F	使用機能	HCAN
-------------------	----	-----------	------	------

1.1 仕様

- (1) データフレーム (* 1) の送受信を行います。(H8S/2623Fを2個使用)
 図1.1にデータフレームの仕様を示します。
 (a) SOF : データフレームの開始を示します。
 (b) アービトレーションフィールド (* 2) : " 101010101010 " とします。
 ・ RTR = " 0 " : データフレームの選択
 (c) コントロールフィールド (* 3) : " 000001 " とします。
 ・ IDE = " 0 " : スタンダードフォーマットの選択
 ・ RO = " 0 " : リザーブビット
 ・ DLC = " 0001 " : 1バイトのデータ長の設定
 (d) データフィールド (* 4) : " 10101010 " とします。
 (e) CRCフィールド (* 5) : HCAN内部でCRCが自動生成されます。
 (f) ACKフィールド (* 6) : 送信側では " 11 "、受信側では " 01 (正常時)" が出力されます。
 (g) EOF : 送信/受信データフレームの完了を示します。
- (2) 通信速度は250Kbps (20MHz動作時) とします。
 (3) データ長は、1バイトとします。
 (4) メッセージの送信はメールボックス1を用います。
 (5) メッセージの受信はメールボックス0を用います。メッセージの受信方式としてはIdentifierをマスクして一致すれば受信とします。
 (6) 受信データは内蔵RAMに格納します。
 (7) 図1.2にCANバスの接続例を示します。

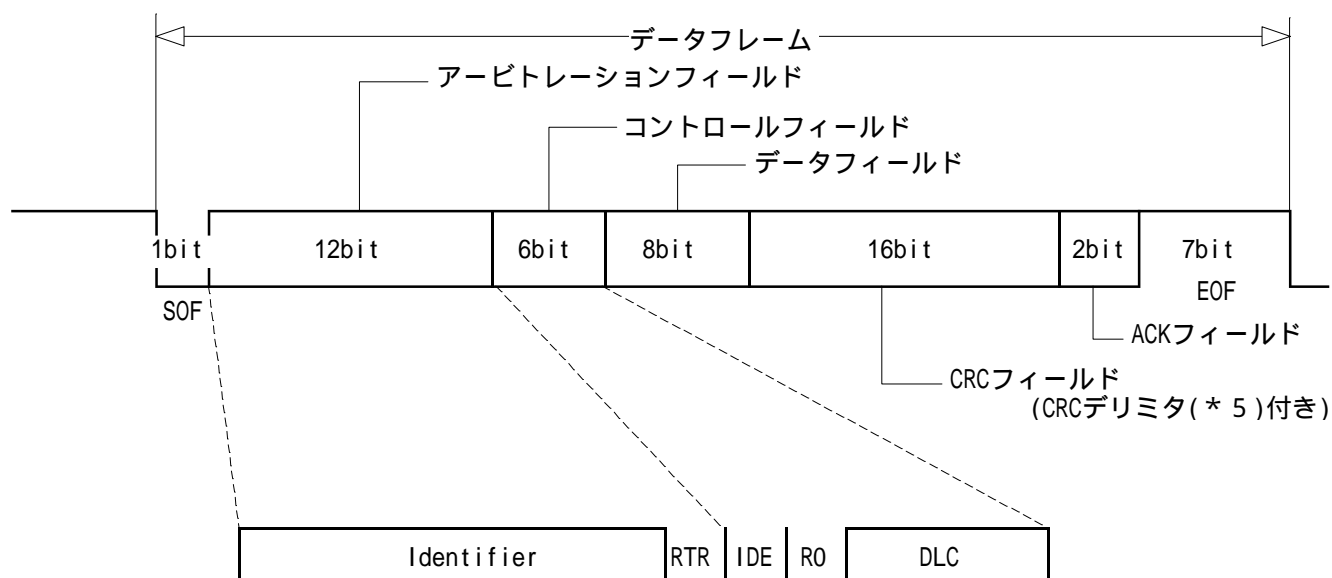
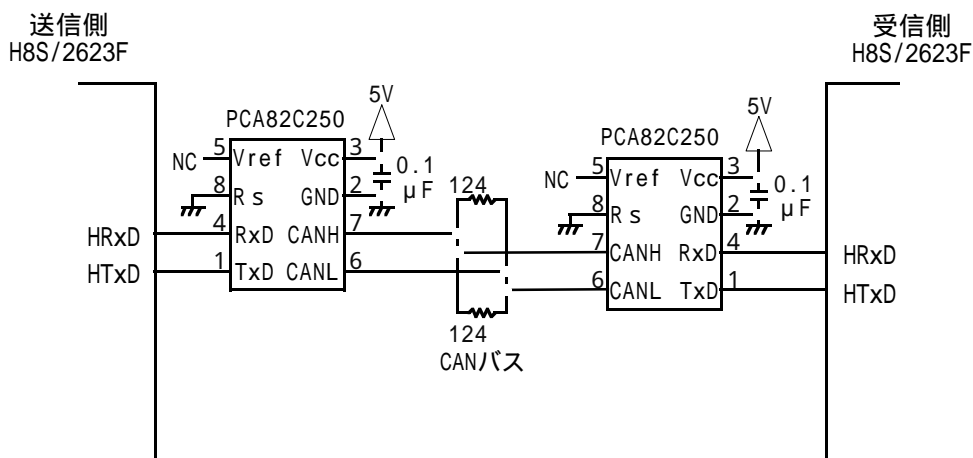


図1.1 データフレームの仕様

(*) は「1.7 備考」を参照してください。

1 HCAN送受信(例題 1 .)	対象	H8S/2623F	使用機能	HCAN
-------------------	----	-----------	------	------

1.1 仕様



本LSIとCANバスを接続するためにはバストランシーバICが必要になります。Philips社製PCA82C50とコンパチブルなものを推奨します。

図1.2 H8S/2623Fを用いたCANインタフェース

1 HCAN送受信(例題 1 .)	対象	H8S/2623F	使用機能	HCAN
-------------------	----	-----------	------	------

1.2 使用機能説明

表1.1、表1.2に本タスク例の機能割り付けを示します。本タスク例は表1.1、表1.2に示すようにH8S/2623Fに内蔵しているHCANの機能を割り付け、HCANの送受信を行います。

表1.1 HCAN機能割り付け

HCANレジスタ		機能
端子	HTxD	メッセージの送信を行います。
	HRxD	メッセージの受信を行います。
送受信 共通 レジスタ	IRR	各割り込み要因のステータスを示します。
	BCR	CANのボーレートプリスケアラ、ビットタイミングパラメータを設定します。
	MBCR	メールボックスの送信/受信を設定します。
	MCR	CANインタフェースの制御を行います。
	MCO_1 § MC15_8	アービトラションフィールド、コントロールフィールドの設定を行います。
	MDO_1 § MD15_8	データフィールドの設定を行います。
送信用 レジスタ	TXPR	送信メッセージをメールボックスに格納後送信待ち状態を設定します。
	TXACK	対応するメールボックスの送信メッセージが正常に送信されたことを示します。
受信用 レジスタ	LAFMH	受信用メールボックス0のIdentifier用フィルタマスクの設定を行います。
	RXPR	対応するメールボックスにデータが正常に受信されたことを示します。

表1.2 MSTPCR機能割り付け

MSTPCRレジスタ	機能
MSTPCRC	モジュールストップモードの制御を行います。

1.3 動作説明(送信時)

図1.3に送信時の動作原理を示します。図1.3に示すようにH8S/2623Fのハードウェア処理およびソフトウェア処理によりHCANの送信を行います。

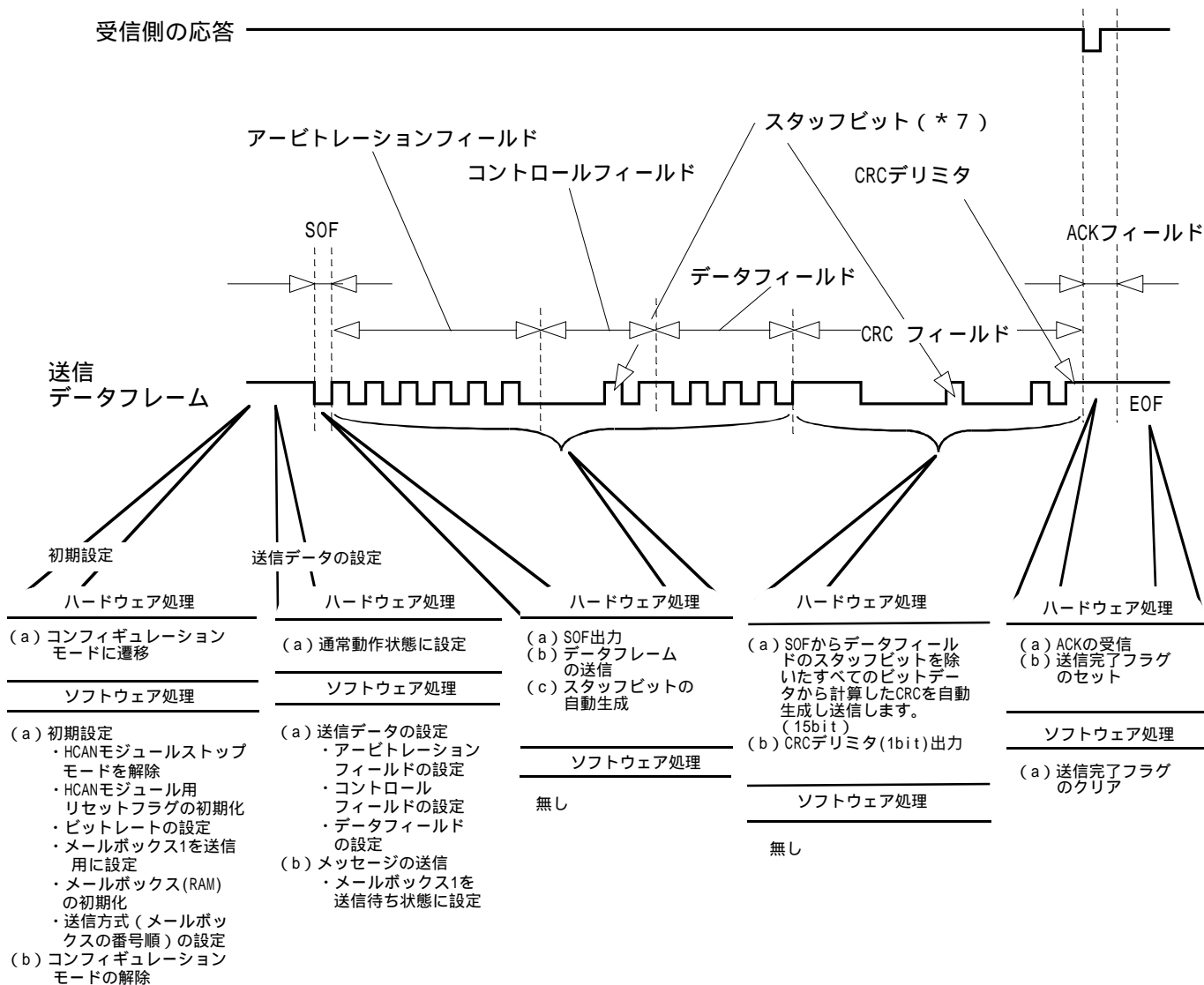


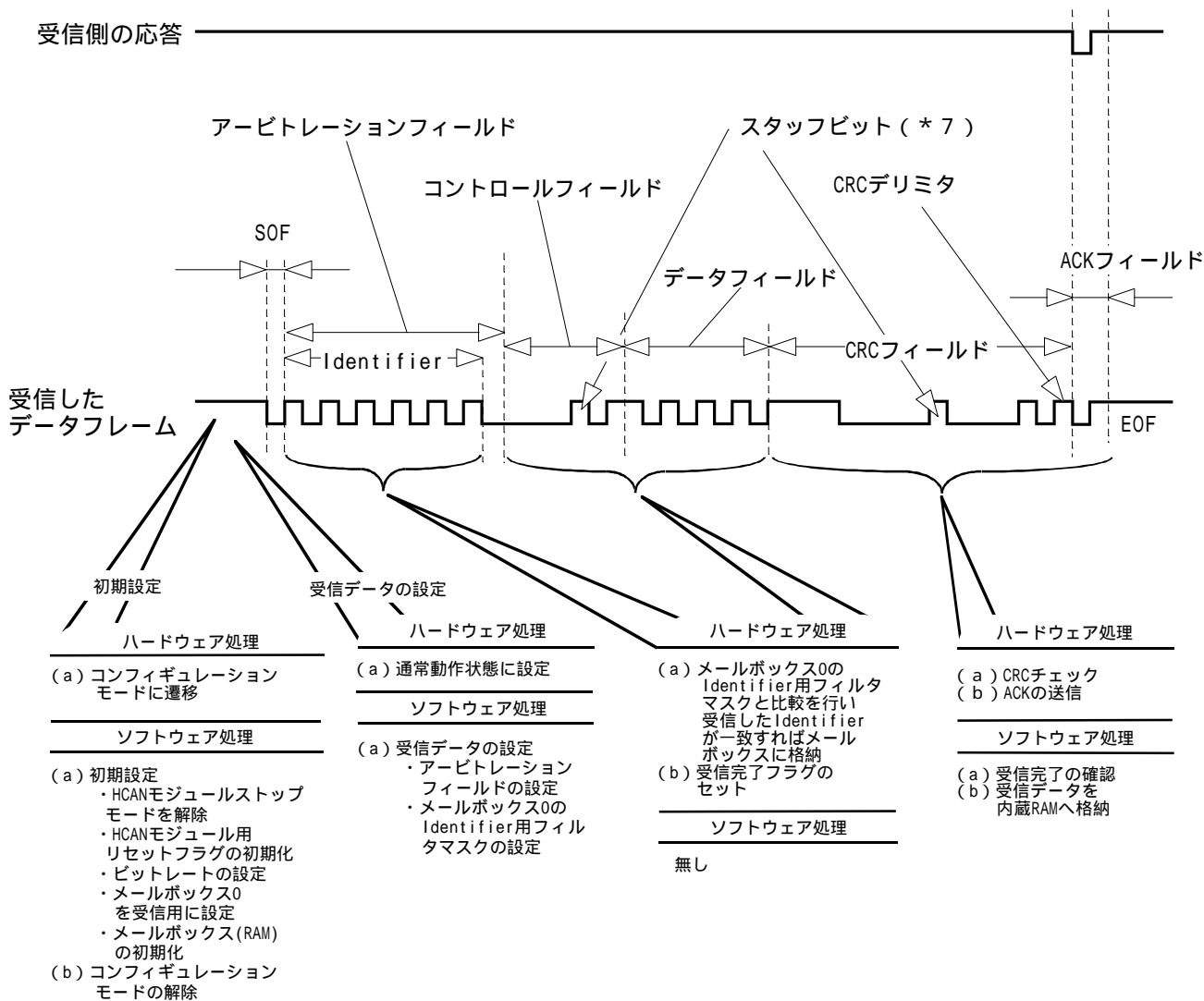
図1.3 HCAN送信時の動作

(*)は「1.7 備考」を参照してください。

1 HCAN送受信(例題 1 . .)	対象	H8S/2623F	使用機能	HCAN
----------------------	----	-----------	------	------

1.3 動作説明(受信時)

図1.4に受信時の動作原理を示します。図1.4に示すようにH8S/2623Fのハードウェア処理およびソフトウェア処理によりHCANの受信を行います。



(*) は「1.7 備考」を参照してください。

1 HCAN送受信(例題 1 . .)	対象	H8S/2623F	使用機能	HCAN
----------------------	----	-----------	------	------

1.4 ソフトウェア説明

(1) モジュール説明

モジュール名	ラベル名	機能
メインルーチン	main	HCANの初期設定および送信、受信の設定を行います。

(2) 使用変数の説明

ラベル名	機能	データ長	使用モジュール名
COUNT	HCAN_MCO_1 ~ MC15_8, HCAN_MDO_1 ~ MD15_8を初期化します。	unsigned short	メインルーチン

(3) 仕様内部レジスタの説明

レジスタ名	機能	設定値	使用モジュール名	
送受信共通設定				
MSTPCRRC	HCANモジュールストップモードを解除します。	0xF7	メインルーチン	
HCAN_IRR	HCANモジュール用リセットフラグを初期化します。	0x0100		
HCAN_BCR	HCANのビットレートを250Kbpsに設定します。	0x0334		
送信時の設定				
HCAN_MBCR	メールボックス1を送信用に設定します。	0xFDFF		
HCAN_MCR	メールボックスの番号順で送信するように設定および、リセットリクエストビットをクリアします。	0x04		
HCAN_MC1_1	1バイトのデータ長を設定します。	0x01		
HCAN_MC1_5	メールボックス1のデータフレーム、スタンダードフォーマットを選択および、Identifierを設定します。	0xA0		
HCAN_MC1_6	メールボックス1のIdentifierを設定します。	0xAA		
HCAN_MD1_1	メールボックス1の送信データを設定します。	0xAA		
HCAN_TXPR	メールボックス1を送信待ち状態に設定します。	0x0200		
HCAN_TXACK	データフレーム送信完了フラグをクリアします。	0x0200		
受信時の設定				
HCAN_MBCR	メールボックス0を受信用に設定します。	0x0100		
HCAN_MCR	リセットリクエストビットをクリアします。	0xFE		
HCAN_MCO_5	メールボックス0のデータフレーム、スタンダードフォーマットを選択および、Identifierを設定します。	0xA0		
HCAN_MCO_6	メールボックス0のIdentifierを設定します。	0xAA		
HCAN_LAFMH	メールボックス0のIdentifier用フィルタマスクを設定します。	0x0000		

(4) 使用RAM

シンボル名	機能	アドレス	使用モジュール名
MAIL_BOX0	HCAN_MDO_1のデータを格納します。	H'FFC100	メインルーチン

1.5 送信フローチャート

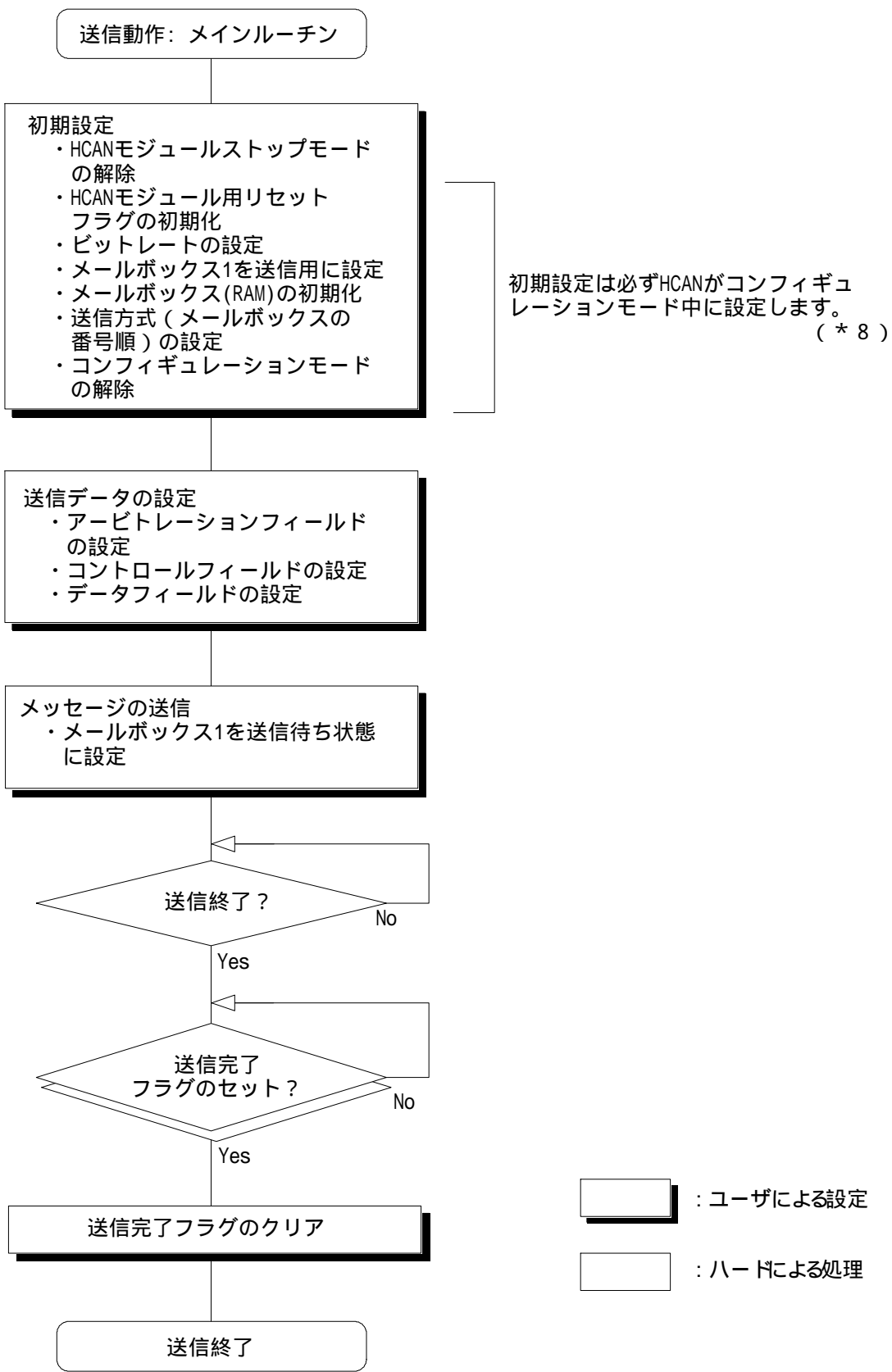
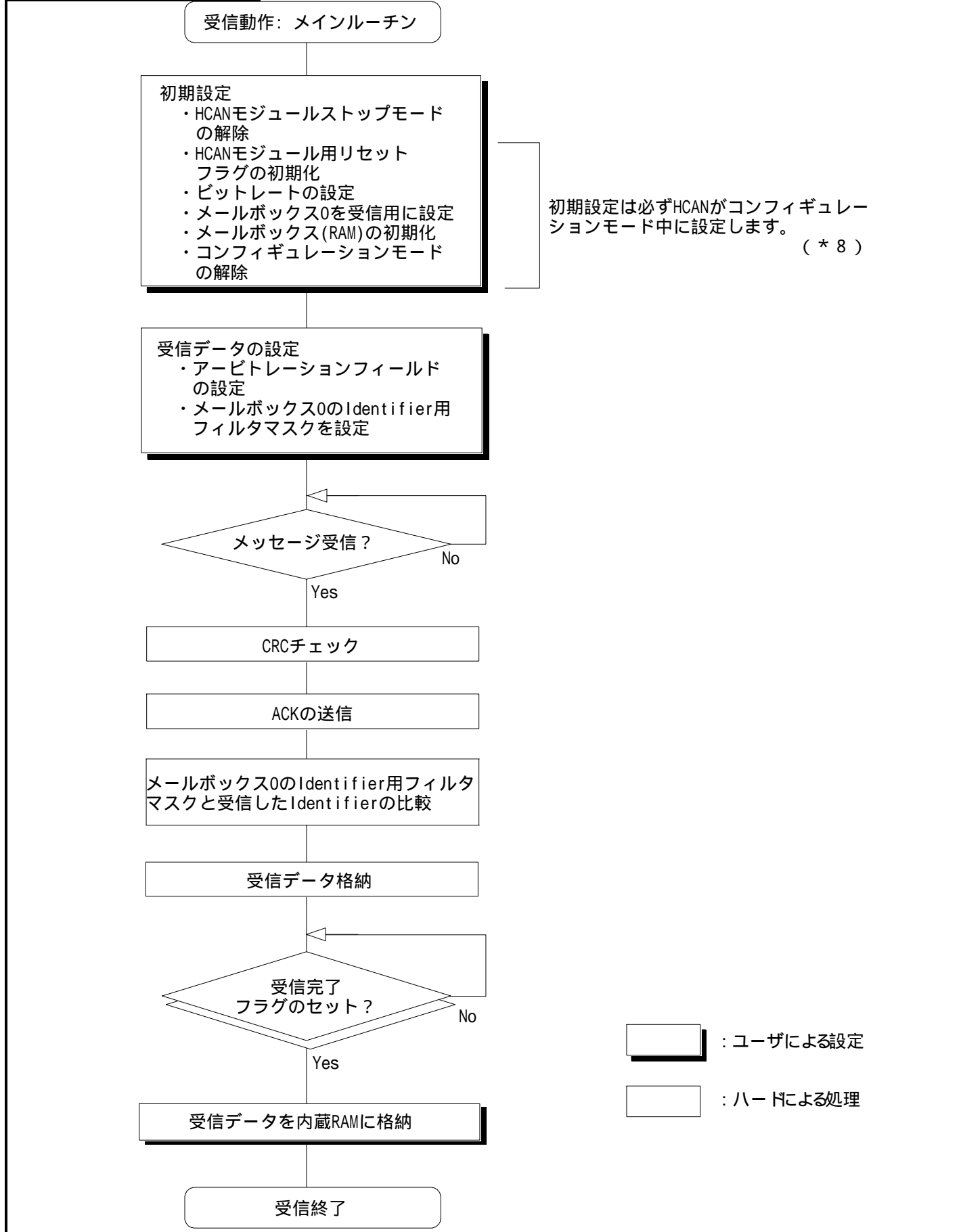


図1.5 送信時のフローチャート

1.5 受信フローチャート



初期設定は必ずHCANがコンフィギュレーションモード中に設定します。 (* 8)

: ユーザによる設定
 : ハードによる処理

図1.6 受信時のフローチャート

1 HCAN送受信(例題 1 .)	対象	H8S/2623F	使用機能	HCAN
-------------------	----	-----------	------	------

1.6 送信プログラムリスト

```

/*****
/*          HCAN 送信プログラム          */
/*****
#include <stdio.h>          /* ライブ 関数用ヘッダ ファイル          */
#include <machine.h>       /* ライブ 関数用ヘッダ ファイル          */
#include "2623.h"         /* 周辺レジスタ定義ヘッダ ファイル          */
/*****
/*          関数プロトコル宣言          */
/*****
void main( void );
/*****
/*          定数定義          */
/*****
#define COUNT (*(unsigned short *)0xFFC000)
/*****
/*          メインルーチン          */
/*****
void main(void)
{
/* 初期設定 */
MSTPCR = 0xF7;          /* HCANモジュールストップモードの解除          */
HCAN_IRR = 0x0100;     /* HCANモジュールリセットフラグの初期化          */
HCAN_BCR = 0x0334;     /* ビットレート 250Kbps          */
HCAN_MBCR = 0xFDFF;    /* メールボックス1を送信用に設定          */
for( COUNT = 0; COUNT < 128; COUNT++ ) /* メールボックス(RAM)の初期化          */
{
*(char*)&HCAN_MCO_1 + COUNT) = 0x00;
}
for( COUNT = 0; COUNT < 128; COUNT++ ) /* メールボックス(RAM)の初期化          */
{
*(char*)&HCAN_MDO_1 + COUNT) = 0x00;
}
HCAN_MCR = 0x04;      /* メールボックスの番号順で送信、コンフィグレーションモードの解除          */
/* 送信データの設定 */
HCAN_MC1_5 = 0xA0;    /* データフレーム、スタンダードフォーマットの選択、Identifierの設定          */
HCAN_MC1_6 = 0xAA;    /* Identifierの設定          */
HCAN_MC1_1 = 0x01;    /* データ長：1バイト          */
HCAN_MD1_1 = 0xAA;    /* 送信データ：10101010          */
/* メッセージの送信 */
HCAN_TXPR = 0x0200;   /* メールボックス1を送信待ち状態に設定          */
while((HCAN_TXACK & 0x0200) != 0x0200); /* 送信完了待ち          */
/* 送信完了フラグのクリア */
HCAN_TXACK &= 0x0200; /* 送信完了フラグのクリア          */
while(1);
}

```

1 HCAN送受信(例題 1 .)	対象	H8S/2623F	使用機能	HCAN
-------------------	----	-----------	------	------

1.6 受信プログラムリスト

```

/*****
/*
/*          HCAN 受信プログラム          */
/*****
#include <stdio.h>          /* ライブ 関数用ヘッダ ファイル          */
#include <machine.h>       /* ライブ 関数用ヘッダ ファイル          */
#include "2623.h"         /* 周辺レジスタ定義ヘッダ ファイル          */
/*****
/*          関数プロトコル宣言          */
/*****
void main( void );
/*****
/*          定数定義          */
/*****
#define COUNT          (*(unsigned short *)0xFFC000)
#define MAIL_BOX0     (*(unsigned char *) 0xFFC100) /* メールボックスの受信データ格納          */
/*****
/*          メインルーチン          */
/*****
void main(void)
{
/* 初期設定 */
MSTPCRC = 0xF7;          /* HCANシミュレーションモードの解除          */
HCAN_IRR = 0x0100;      /* HCANシミュレーション用リセットフラグの初期化          */
HCAN_BCR = 0x0334;      /* ビットレート 250Kbps          */
HCAN_MBCR = 0x0100;     /* メールボックスを受信に設定          */
for( COUNT = 0; COUNT < 128; COUNT++ ) /* メールボックス(RAM)の初期化          */
{
*(char*)&HCAN_MCO_1 + COUNT) = 0x00;
}
for( COUNT = 0; COUNT < 128; COUNT++ ) /* メールボックス (RAM)の初期化          */
{
*(char*)&HCAN_MDO_1 + COUNT) = 0x00;
}
HCAN_MCR &= 0xFE;      /* コンフィギュレーションモードの解除          */
/* 受信データの設定 */
HCAN_MCO_5 = 0xA0;     /* データフレーム、スタンダードフォーマット、Identifierの設定          */
HCAN_MCO_6 = 0xAA;     /* Identifierの設定          */
HCAN_LAFMH = 0x0000;   /* メールボックスのIdentifier用のフィルタマスクの設定          */
while((HCAN_RXPR & 0x0100) != 0x0100); /* 受信完了待ち          */
/* 受信データを内蔵RAMに格納 */
MAIL_BOX0 = HCAN_MDO_1; /* 受信データの格納          */
while(1);
}

```

1 HCAN送受信(例題 1 .)	対象	H8S/2623F	使用機能	HCAN
-------------------	----	-----------	------	------

1.7 備考

- (* 1) データフレーム : 送信元から送信先へ転送したいデータです。
- (* 2) アービトレーションフィールド : メッセージ固有のIDおよびデータフレームまたはリモートフレームの設定をします。
- (* 3) コントロールフィールド : 転送するデータ長およびスタンダードフォーマットまたはエクステンデッドフォーマットの設定をします。
- (* 4) データフィールド : メッセージの内容(送信したいデータ)を設定します。
- (* 5) CRCフィールド : HCAN内部でSOFからデータフィールドのスタッフビットを除いたすべてのビットデータからCRCが自動生成され、送信メッセージのエラーを検出します。15bitのCRCと1bitのCRCデリミタから成ります。CRCデリミタはCRC後に必ず " 1 " が出力されます。

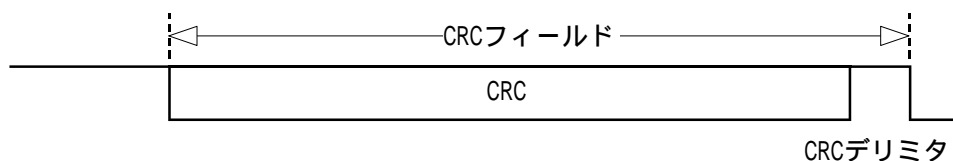


図1.7 CRCフィールド

CRCについて

伝送しようとするデータ多項式 $P(X)$ に X^R を乗じ、さらにこの $X^R \cdot P(X)$ を生成多項式 $G(X)$ で除し、余りである $R(X)$ を求めます。情報を送り出す側では、 $X^R \cdot P(X)$ に求められた $R(X)$ をチェックビットとして付加し電信情報 $Tx(X)$ として送りだします。

また、情報を受ける側では受けた受信情報 $Rx(X)$ を生成多項式 $G(X)$ で除し、余りを求めます。この余りがゼロとなった場合は、情報伝送は正しく行われたとみなし、余りが出た場合は伝送された情報に誤りがあったと判断します。

$P(X) = \{ \text{スタッフビットを除いたSOFからデータフィールドまで} \}$

$G(X) = X^{15} + X^{14} + X^{10} + X^8 + X^7 + X^4 + X^3 + 1$

$R = 15$

$G(X)$ はCRCを生成する多項式としてCANプロトコルにて規定されています。

例として以下の設定において送信されたデータフレームについて説明します。

設定値

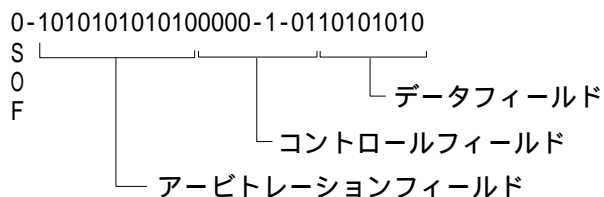
SOF : 0

アービトレーションフィールド : 101010101010

コントロールフィールド : 000001

データフィールド : 10101010

送信されるデータのSOFからデータフィールドは(スタッフビット)



で、対象となるデータはスタッフビットを除いた

01010101010100000110101010

すなわち、 $P(X) = X^{25} + X^{23} + X^{21} + X^{19} + X^{17} + X^{15} + X^8 + X^7 + X^5 + X^3 + X^1$ となります。

1.7 備考

ここで、 $P(X)$ に X^{15} を掛け、
 $X^{15} \cdot P(X) = X^{40} + X^{38} + X^{36} + X^{34} + X^{32} + X^{30} + X^{23} + X^{22} + X^{20} + X^{18} + X^{16}$
とし、この値を
 $G(X) = X^{15} + X^{14} + X^{10} + X^8 + X^7 + X^4 + X^3 + 1$
で除します。

$$\left(\begin{array}{l} X^{15} \cdot P(X) = X^{40} + X^{38} + X^{36} + X^{34} + X^{32} + X^{30} + X^{23} + X^{22} + X^{20} + X^{18} + X^{16} \\ P[40:0] = 1010101010100000011010101000000000000000 \\ G(X) = X^{15} + X^{14} + X^{10} + X^8 + X^7 + X^4 + X^3 + 1 \\ G[15:0] = 1100010110011001 \end{array} \right)$$

商

	1010101010100000011010101000000000000000	
1100010110011001	1100010110011001	
	110111100110010	
	1100010110011001	
	1101111101011110	
	1100010110011001	
	1101011000111101	
	1100010110011001	
	1001110100100010	
	1100010110011001	
	1011000101110110	
	1100010110011001	
	1110100111011110	
	1100010110011001	
	1011000100011100	
	1100010110011001	
	1110100100001010	
	1100010110011001	
	1011001001001100	
	1100010110011001	
	111011110101010	
	1100010110011001	
	1010100011001100	
	1100010110011001	
	1101101010101010	
	1100010110011001	
	1111100110011000	
	1100010110011001	
	111100000000010	
	111100000000010	← 余り : R[14:0]

計算の結果、次の値がえられます。
 $R[14:0] = 111100000000010$
つまり、これがCRCの値となり、データフィールド後に付加され電信情報Tx(X)として送信されます。
実際にはスタッフビット()とCRCデリミタ()が付加され、
11110000010000101がCRCフィールド上で送信されます。

1 HCAN送受信(例題 1 . .)	対象	H8S/2623F	使用機能	HCAN
----------------------	----	-----------	------	------

1.7 備考

次に、受信した受信情報Rx(X)に誤りがあるか計算してみます。
 Rx(X) とは、前記の計算により求めた余りを $X^{15} \cdot P(X)$ に加算した値であり、この値をG(X)で除します。

$$\left(\begin{array}{r}
 P [40:0]=1010101010100000011010101000000000000000 \\
 + R [14:0]= 11110000000010 \\
 \hline
 1010101010100000011010101011110000000010 \\
 \end{array} \right)$$

この値をG[15:0]で除します。

	商	
1100010110011001	$ \begin{array}{r} 1010101010100000011010101011110000000010 \\ 1100010110011001 \\ \hline 1101111001110010 \\ 1100010110011001 \\ \hline 110111101011110 \\ 1100010110011001 \\ \hline 1101011000111101 \\ 1100010110011001 \\ \hline 1001110100100010 \\ 1100010110011001 \\ \hline 1011000101110111 \\ 1100010110011001 \\ \hline 1110100111011101 \\ 1100010110011001 \\ \hline 1011000100010011 \\ 1100010110011001 \\ \hline 1110100100010100 \\ 1100010110011001 \\ \hline 1011001000110100 \\ 1100010110011001 \\ \hline 1110111101011010 \\ 1100010110011001 \\ \hline 1010101100001100 \\ 1100010110011001 \\ \hline 1101110100101010 \\ 1100010110011001 \\ \hline 1100010110011001 \\ 1100010110011001 \\ \hline 1100010110011001 \\ 1100010110011001 \\ \hline \boxed{0} \end{array} $	← 余り

除数と非除数でEORをとります。

以上の計算から余りはゼロとなり情報伝達は正しいとみなされます。

1 HCAN送受信(例題 1 .)	対象	H8S/2623F	使用機能	HCAN
-------------------	----	-----------	------	------

1.7 備考

(* 6) ACKフィールド : 正常受信確認用です。
1bitのACKスロット、1bitのACKデリミタから成ります。

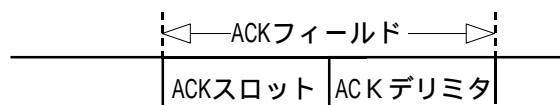


図1.8 ACKフィールド

・ H8S/2623F受信側がCRCチェックでエラーを検出した場合はACKスロット=Highを、検出しなかった場合はACKスロット=Lowを出力します。

(* 7) スタッフビット : データフレーム上でLowが5bit続いた場合、次の1bitは必ずHighが出力されます。同様にHighが5bit続いた場合、次の1bitは必ずLowが出力されます。これらの様にスタッフビットが出力される場合データフレームのビット長は、スタッフビットの分だけ長くなります。

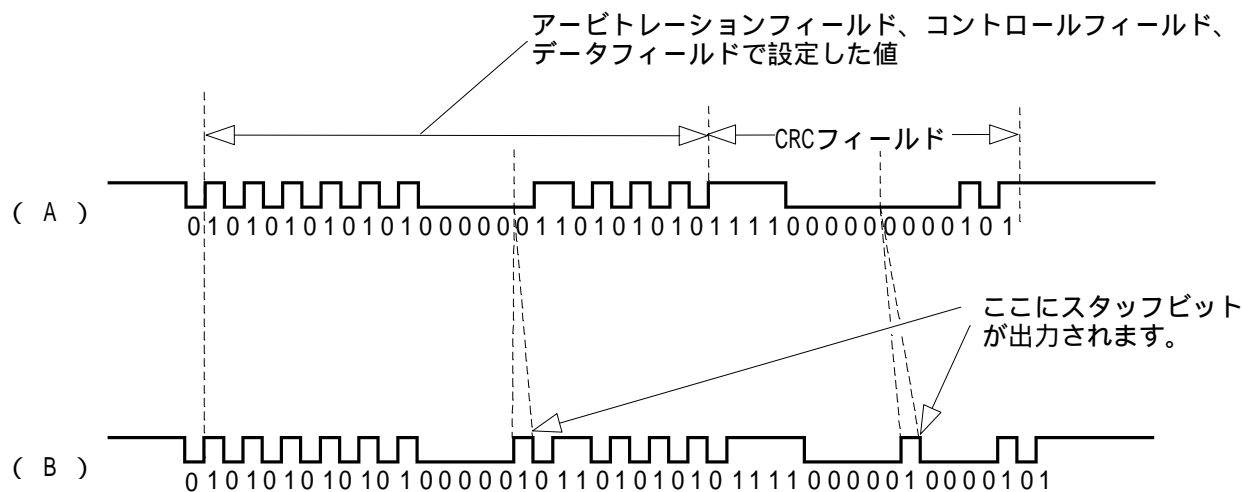
例として、設定値を

アービトレーションフィールド : " 101010101010 "

コントロールフィールド : " 000001 "

とすると、" 101010101010000001 " となり、最下位から2bit目 (0が5つ続いた後) にスタッフビット () が出力されます。

よって、CANバス上に送信される値は " 1010101010100000101 " となり、スタッフビットの1bit分だけ長くなります。



(A)の送信したいデータフレームに対して、CANバス上ではスタッフビットの乗ったデータフレーム (B)が送信されます。また、(B)ではデータフレームの長さもスタッフビットの分だけ長くなっています。

図1.9 スタッフビット

(* 8) コンフィギュレーションモード : HCANモジュールがリセット状態のことであり、解除はマスタコントロールレジスタ (MCR) のリセットリクエストビット (MCRO) をクリアすることで行います。

2.HCAN送受信 (例題 2.): スタンダードフォーマット、8バイトデータ、DTC使用

第2章 目次

2.1 仕様	19
2.2 使用機能説明	21
2.3 動作説明	22
2.4 ソフトウェア説明	24
2.5 フローチャート	26
2.6 プログラムリスト	30
2.7 備考	33

2 HCAN送受信(例題 2 .)	対象	H8S/2623F	使用機能	HCAN, DTC
-------------------	----	-----------	------	-----------

2.1 仕様

- (1) データフレーム (* 1) の送受信を行います。(H8S/2623Fを2個使用)
 図2.1にデータフレームの仕様を示します。
 (a) SOF : データフレームの開始を示します。
 (b) アービトレーションフィールド (* 2) : " 101010101010 " とします。
 ・ RTR = " 0 " : データフレームの選択
 (c) コントロールフィールド (* 3) : " 001000 " とします。
 ・ IDE = " 0 " : スタンダードフォーマットの選択
 ・ RO = " 0 " : リザーブビット
 ・ DLC = " 1000 " : 8バイトのデータ長の設定
 (d) データフィールド (* 4)
 ・ 1バイト目 : " 01010101 (H'55) "
 ・ 2バイト目 : " 01100110 (H'66) "
 ・ 3バイト目 : " 01110111 (H'77) "
 ・ 4バイト目 : " 10001000 (H'88) "
 ・ 5バイト目 : " 10011001 (H'99) "
 ・ 6バイト目 : " 10101010 (H'AA) "
 ・ 7バイト目 : " 10111011 (H'BB) "
 ・ 8バイト目 : " 11111111 (H'FF) " とします。
 (e) CRCフィールド (* 5) : HCAN内部でCRCが自動生成されます。
 (f) ACKフィールド (* 6) : 送信側では " 11 " 、受信側では " 01 (正常時) " が出力されます。
 (g) EOF : 送信/受信データフレームの完了を示します。
- (2) 通信速度は250Kbps (20MHz動作時) とします。
 (3) データ長は、8バイトとします。
 (4) メッセージの送信はメールボックス1を用います。
 (5) メッセージの受信はメールボックス0を用います。メッセージの受信方式としてはIdentifierをマスクして一致すれば受信とします。
 (6) 割り込み制御モード2を選択し、
 (a) バス動作割り込み(OVRO)を使用します。
 (b) メッセージ受信によるDTC割り込み、メッセージ受信割り込み(RMO)を使用します。
 (7) 受信メッセージは、DTCを使用し、内蔵RAMに格納します。
 (a) メッセージ受信で、DTCを起動します。
 (b) DTCの仕様を下記に示します。
 ・ ブロック転送モードを使用します。
 ・ 転送終了後、メッセージ受信割り込み(RMO)を使用します。
 (8) HCANスリープモードを使用します。
 (a) 初期設定後、HCANスリープモードへ遷移します。(CANバス動作により解除。)
 (b) 受信処理後、HCANスリープモードへ遷移します。(CANバス動作により解除。)
 (9) 図2.2にCANバスの接続例を示します。

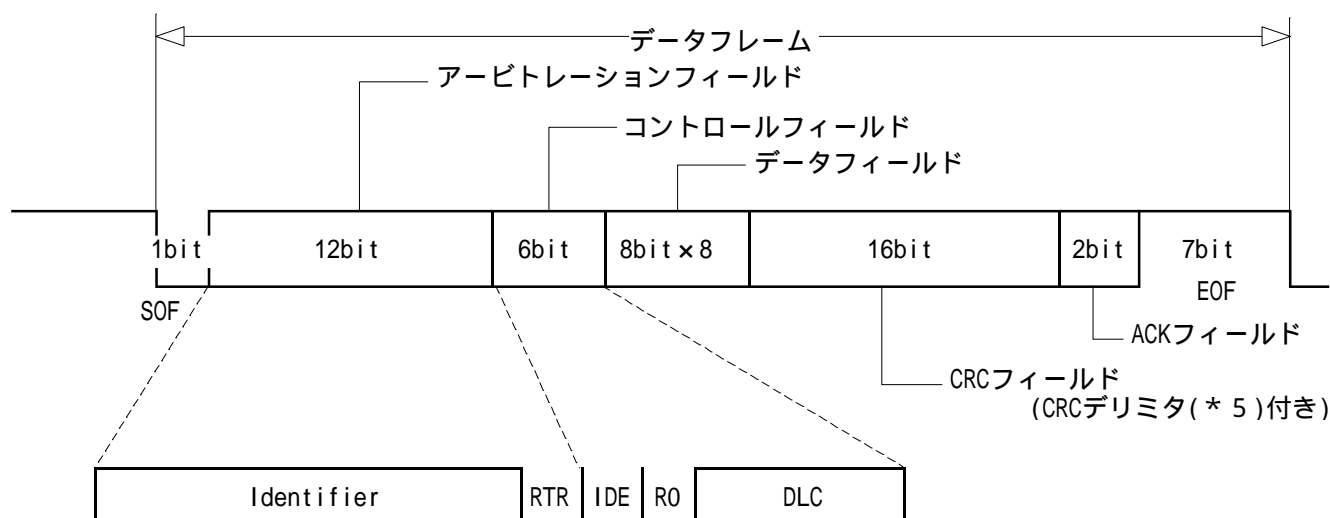
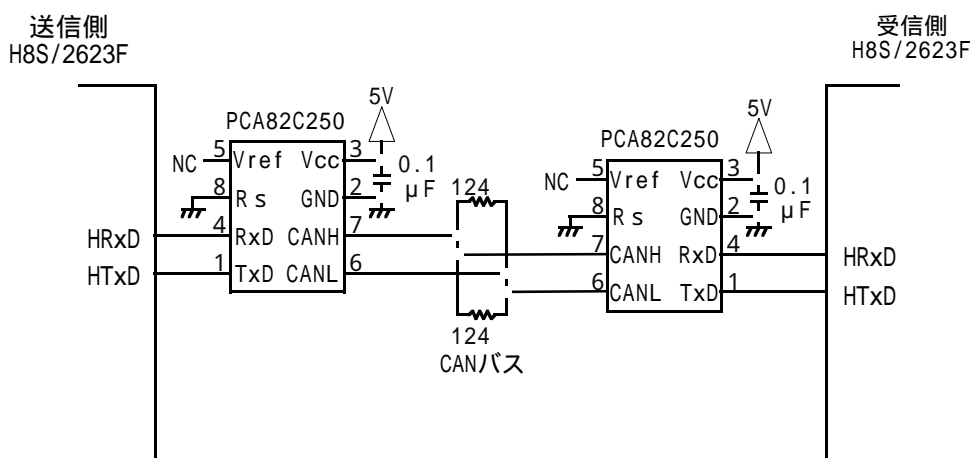


図2.1 データフレームの仕様

2 HCAN送受信(例題 2 .)	対象	H8S/2623F	使用機能	HCAN, DTC
-------------------	----	-----------	------	-----------

2.1 仕様



本LSIとCANバスを接続するためにはバストランシーバICが必要になります。Philips社製PCA82C50とコンパチブルなものを推奨します。

図2.2 H8S/2623Fを用いたCANインタフェース

2 HCAN送受信(例題 2 .)	対象	H8S/2623F	使用機能	HCAN, DTC
-------------------	----	-----------	------	-----------

2.2 使用機能説明

表2.1~2.4に本タスク例の機能割り付けを示します。本タスク例は表2.1~2.4に示すようにH8S/2623Fに内蔵しているHCANの機能を割り付け、HCANの送受信を行います。

表2.1 HCAN機能割り付け

HCANレジスタ		機能
端子	HTxD	メッセージの送信を行います。
	HRxD	メッセージの受信を行います。
送受信 共通 レジスタ	IRR	各割り込み要因のステータスを示します。
	BCR	CANのポーレートプリスケラ、ビットタイミングパラメータを設定します。
	MBCR	メールボックスの送信/受信を設定します。
	MCR	CANインタフェースの制御を行います。
	MC0_1 ┆ MC15_8	アービトラションフィールド、コントロールフィールドの設定を行います。
	MDO_1 ┆ MD15_8	データフィールドの設定を行います。
	送信用 レジスタ	TXPR
TXACK		対応するメールボックスの送信メッセージが正常に送信されたことを示します。
受信用 レジスタ	MBIMR	各メールボックスの割り込み要求を許可/禁止します。
	IMR	各割り込み要因の要求を許可/禁止します。
	LAFMH	受信用メールボックス0のIdentifier用フィルタマスクの設定を行います。

表2.2 MSTPCR機能割り付け

MSTPCRレジスタ	機能
MSTPCRC	モジュールストップモードの制御を行います。
MSTPCRA	

表2.3 割り込みコントローラ割り付け

IPRレジスタ	機能
IPRM	割り込み優先レベルの制御を行います。
SYSCR	割り込み制御モードの設定を行います。
exr	割り込み要求マスクレベルの指定を行います。

表2.4 DTC機能割り付け

DTCレジスタ	機能
SAR (内蔵RAM)	転送元アドレス(受信メッセージの格納元)を設定します。
DAR (内蔵RAM)	転送先アドレス(受信メッセージの格納先)を設定します。
MAR (内蔵RAM)	ブロック転送モード、バイトサイズ転送等の設定をします。
MRB (内蔵RAM)	DTCデータ転送終了後、CPUへの割り込みを設定します。
CRA (内蔵RAM)	DTCのデータ転送の転送回数を設定します。
CRB (内蔵RAM)	DTCのブロックデータ転送の転送回数を設定します。
DTCERG (レジスタ)	割り込みによるDTC起動を設定します。

2.3 動作説明(送信時)

図2.3に送信時の動作原理を示します。図2.3に示すようにH8S/2623Fのハードウェア処理およびソフトウェア処理によりHCANの送信を行います。

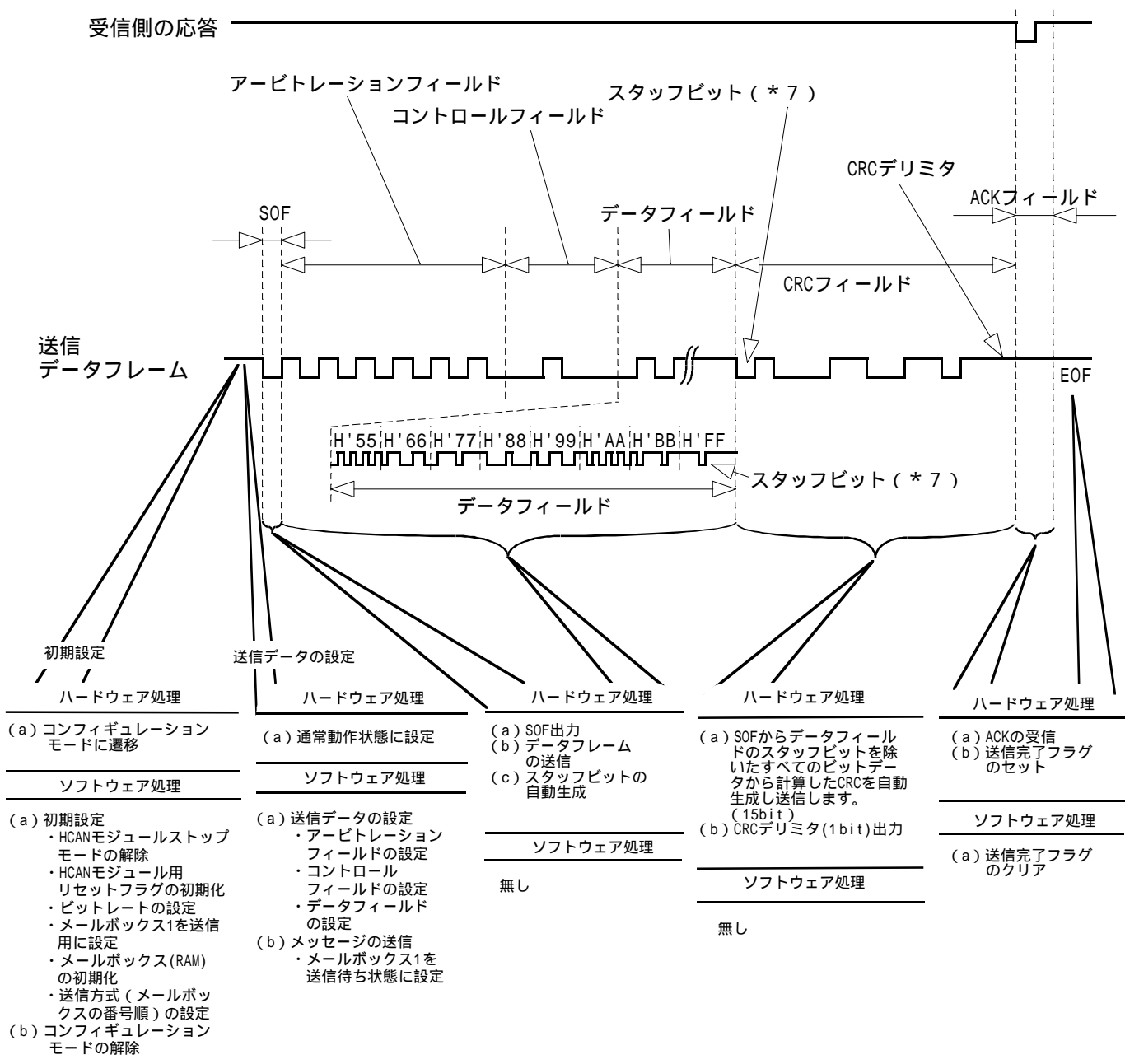
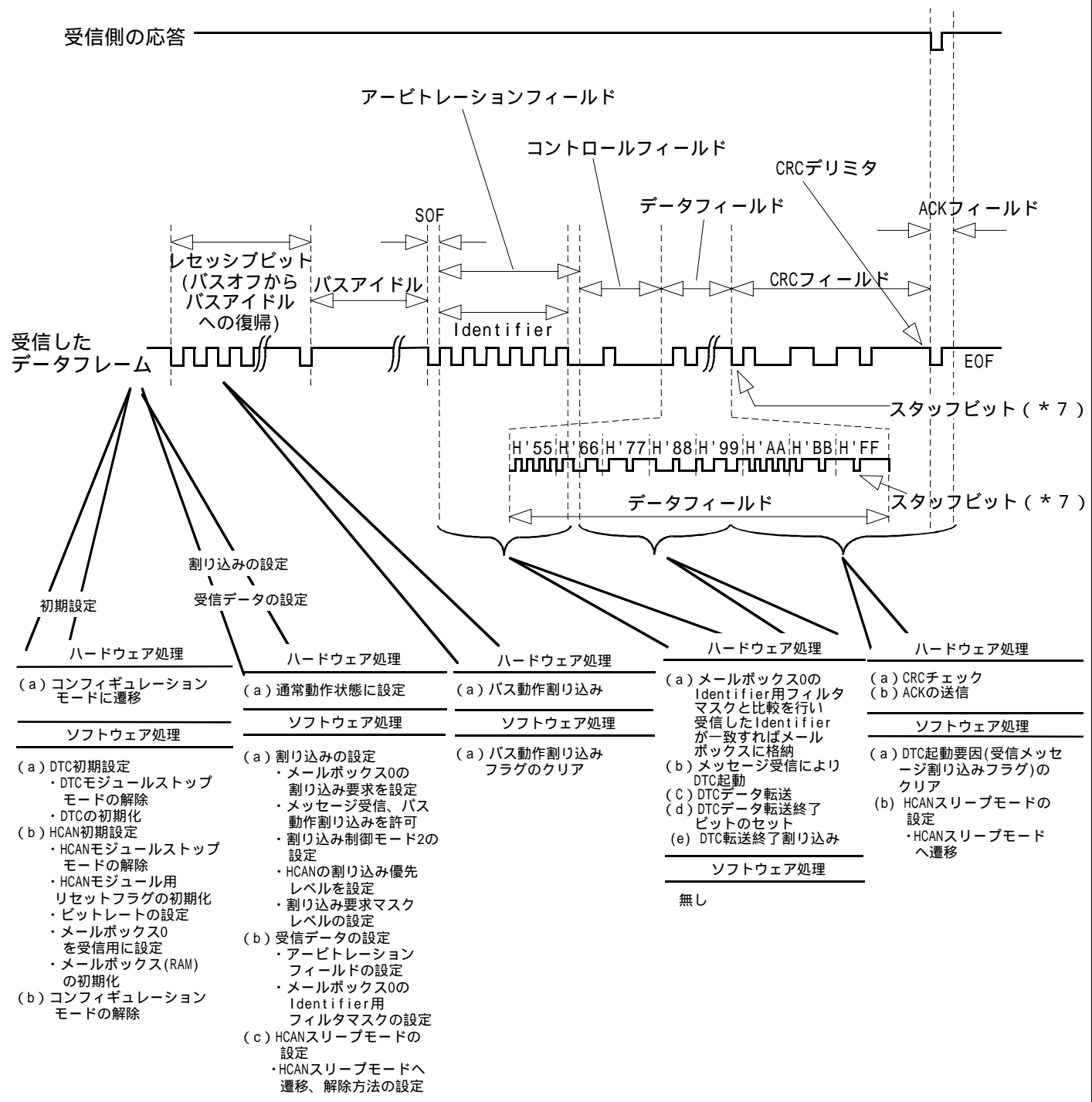


図2.3 HCAN送信時の動作

(*) は「2.7 備考」を参照してください。

2.3 動作説明(受信時)

図2.4に受信時の動作原理を示します。図2.4に示すようにH8S/2623Fのハードウェア処理およびソフトウェア処理によりHCANの受信を行います。



(*) は「2.7 備考」を参照してください。

図2.4 HCAN受信時の動作

2 HCAN送受信(例題 2 .)	対象	H8S/2623F	使用機能	HCAN, DTC
-------------------	----	-----------	------	-----------

2.4 ソフトウェア説明

(1) モジュール説明

モジュール名	ラベル名	機能
メインルーチン	main	HCANの初期設定および送信、受信の設定を行います。
バス動作割り込みルーチン	OVR0_IRR12	バス動作割り込みフラグのクリアを行います。
DTC転送終了割り込みルーチン	DTCend_RMO	DTC起動要因をクリアし、HCANスリープモードへ遷移します。

(2) 使用変数の説明

ラベル名	機能	データ長	使用モジュール名
COUNT	HCAN_MCO_1 ~ MC15_8, HCAN_MDO_1 ~ MD15_8を初期化します。	unsigned short	メインルーチン

(3) 仕様内部レジスタの説明

レジスタ名	機能	設定値	使用モジュール名	
送受信共通設定				
MSTPCRC	HCANモジュールストップモードを解除します。	0xF7	メインルーチン	
HCAN_BCR	HCANのビットレートを250Kbpsに設定します。	0x0334		
送信時の設定				
HCAN_IRR	HCANモジュール用リセットフラグを初期化します。	0x0100		
HCAN_MBCR	メールボックス1を送信用に設定します。	0xFDFE		
HCAN_MCR	メールボックスの番号順で送信するように設定および、リセットリクエストビットをクリアします。	0x04		
HCAN_MC1_1	8バイトのデータ長を設定します。	0x08		
HCAN_MC1_5	メールボックス1のデータフレーム、スタンダードフォーマットを選択および、Identifierを設定します。	0xA0		
HCAN_MC1_6	メールボックス1のIdentifierを設定します。	0xAA		
HCAN_MD1_1	メールボックス1の1バイト目の送信データを設定します。	0x55		
HCAN_MD1_2	メールボックス1の2バイト目の送信データを設定します。	0x66		
HCAN_MD1_3	メールボックス1の3バイト目の送信データを設定します。	0x77		
HCAN_MD1_4	メールボックス1の4バイト目の送信データを設定します。	0x88		
HCAN_MD1_5	メールボックス1の5バイト目の送信データを設定します。	0x99		
HCAN_MD1_6	メールボックス1の6バイト目の送信データを設定します。	0xAA		
HCAN_MD1_7	メールボックス1の7バイト目の送信データを設定します。	0xBB		
HCAN_MD1_8	メールボックス1の8バイト目の送信データを設定します。	0xFF		
HCAN_TXPR	メールボックス1を送信待ち状態に設定します。	0x0200		
HCAN_TXACK	データフレーム送信完了フラグをクリアします。	0x0200		

2 HCAN送受信(例題 2 .)	対象	H8S/2623F	使用機能	HCAN, DTC
-------------------	----	-----------	------	-----------

2.4 ソフトウェア説明

レジスタ名	機能	設定値	使用モジュール名
受信時の設定			
HCAN_IRR	HCANモジュール用リセットフラグを初期化します。	0x0100	メイン
	バス動作割り込みフラグをクリアします。	0x0010	バス動作
HCAN_MCR	リセットリクエストビットをクリアします。	0xFE	メイン
	HCANスリープモードへ遷移、バス動作による解除を設定します。	0xA0	
	HCANスリープモードへ遷移します。	0x20	DTC終了
HCAN_RXPR	受信メッセージ割り込みをクリアします。	0xFFFF	メイン
HCAN_MBCR	メールボックス0を受信用に設定します。	0x0100	
HCAN_MCO_5	メールボックス0のデータフレーム、スタンダードフォーマットを選択および、Identifierを設定します。	0xA0	
HCAN_MCO_6	メールボックス0のIdentifierを設定します。	0xAA	
HCAN_LAFMH	メールボックス0のIdentifier用フィルタマスクを設定します。	0x0000	
HCAN_MBIMR	メールボックス0の割り込み要求許可に設定します。	0xFEFF	
HCAN_IMR	メッセージ受信、バス動作割り込み要求許可に設定します。	0xFCEF	
INTC.IPRM	HCANの割り込み優先レベルを7に設定します。	0x70	
SYSCR	割り込み制御モード2に設定します。	0x20	
exr	割り込み要求マスクレベルを指定します。	0x00	
MSTPCRA	DTCモジュールストップモードを解除します。	0x3F	
DTC_DTCERG	HCAN受信メッセージによるDTC起動を設定します。	0x20	

(4) 使用RAM

シンボル名	機能	アドレス	使用モジュール名
受信時の設定			
Message_Box1 ~ 8	メールボックス0用データ(8バイト)の格納先アドレス。	0xFFC000 ~ 7 (不定)	メイン
SAR	転送元アドレス(受信メッセージの格納元)を設定します。	0xFFEC01 ~ 3 (0xFFFF8B0)	
MRA	ブロック転送モード、バイトサイズ転送等の設定をします。	0xFFEC00 (0xA8)	
DAR	転送先アドレス(受信メッセージの格納先)を設定します。	0xFFEC05 ~ 7 (0xFFC000)	
MRB	DTCデータ転送終了後、メッセージ受信割り込み(RM0)を許可します。	0xFFEC04 (0x00)	
CRA	DTCのデータ転送の転送ブロックサイズ保持値(上位バイト)と転送ブロックサイズカウンタ(下位バイト)を設定します。	0xFFEC08, 9 (0x0808)	
CRB	DTCのブロック転送の転送回数を設定します。	0xFFEC0A, B (0x0001)	

(注意)

- ・メイン：メインルーチン
- ・バス動作：バス動作割り込みルーチン
- ・DTC終了：DTC転送終了割り込みルーチン

()内は設定値

2.5 送信フローチャート

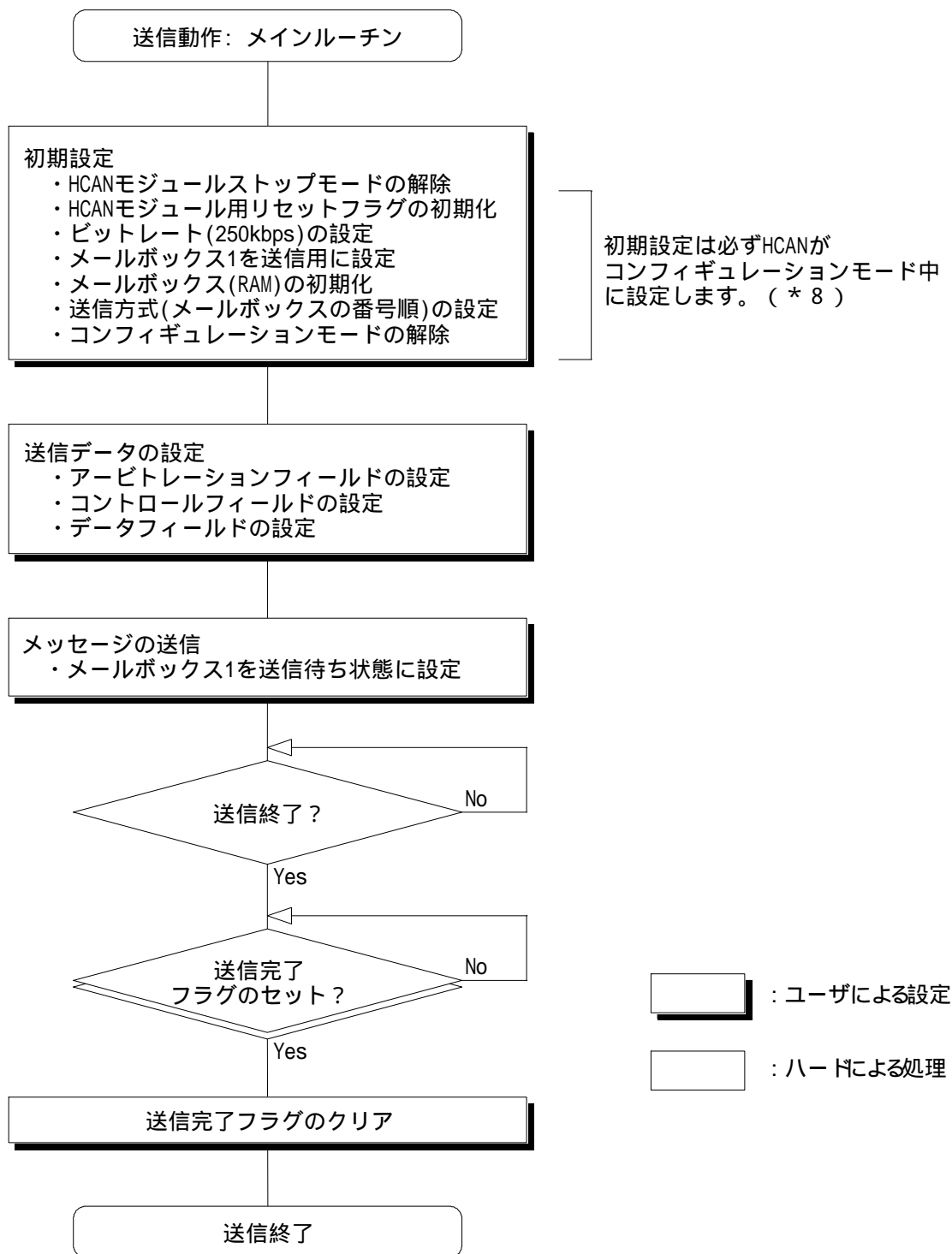


図2.5 送信時のフローチャート

(*) は「2.7 備考」を参照してください。

2.5 受信フローチャート

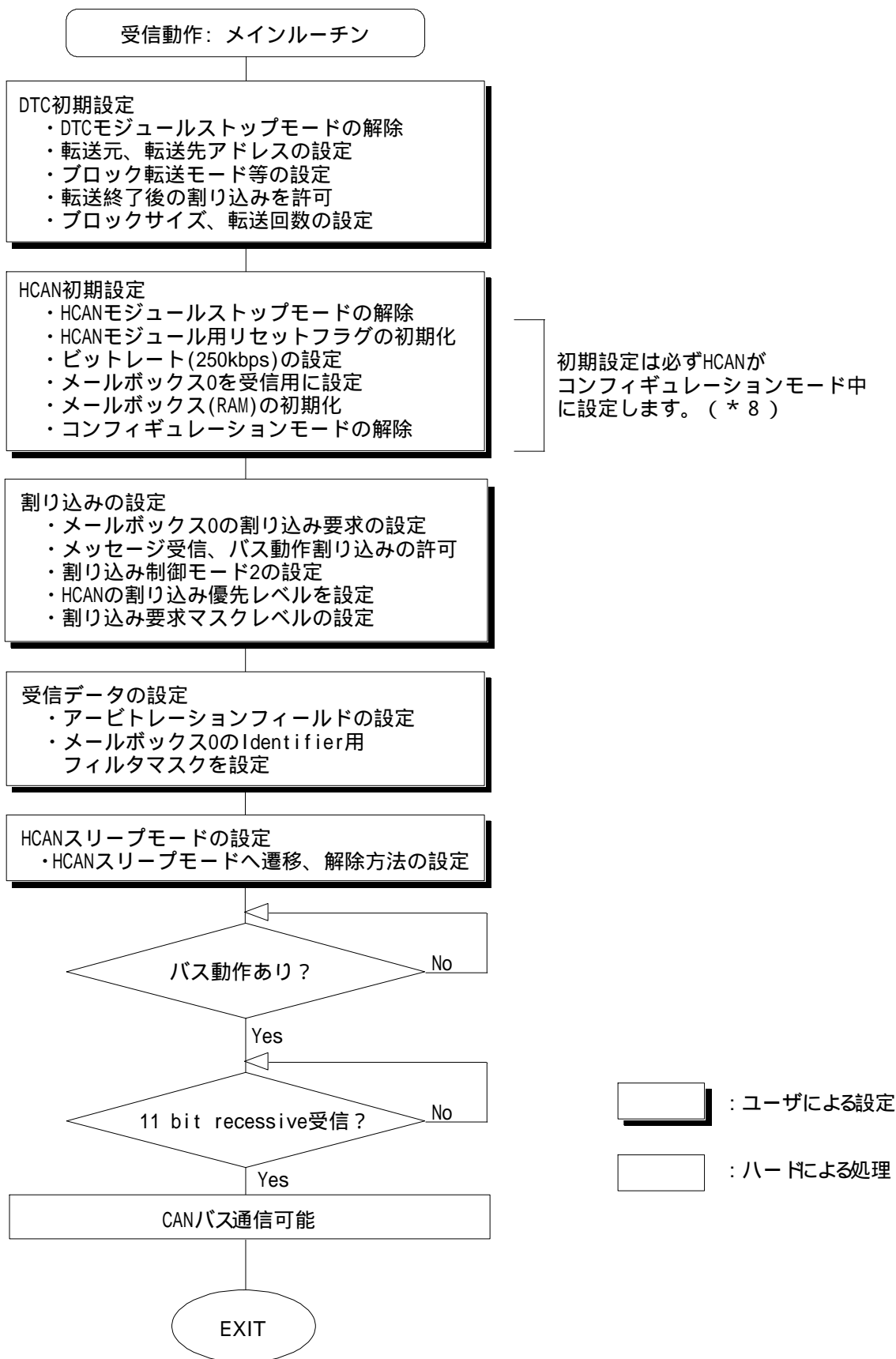


図2.6 受信時のフローチャート(1)

2.5 受信フローチャート

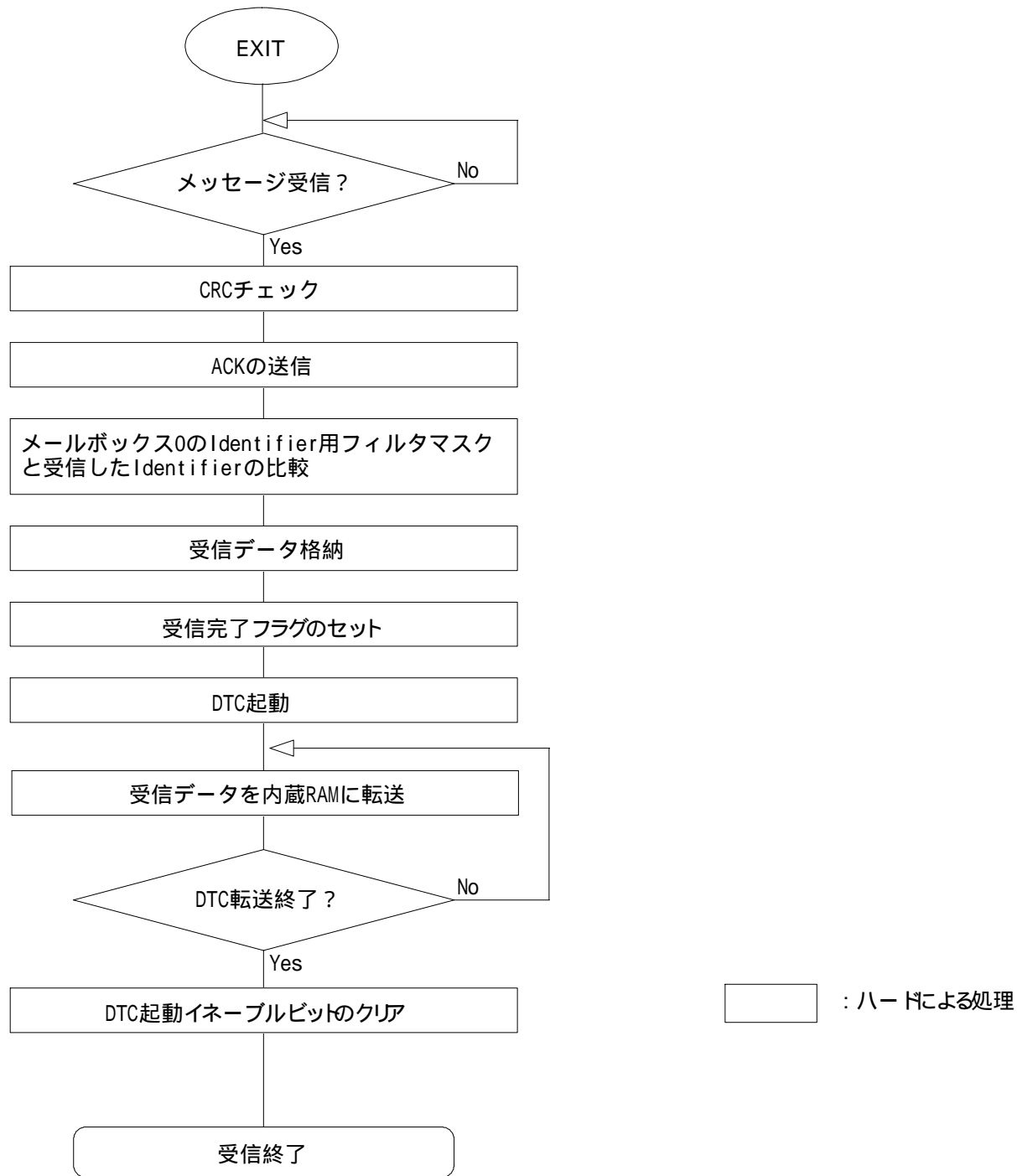


図2.7 受信時のフローチャート(2)

2.5 受信フローチャート



図2.8 バス動作割り込みのフローチャート

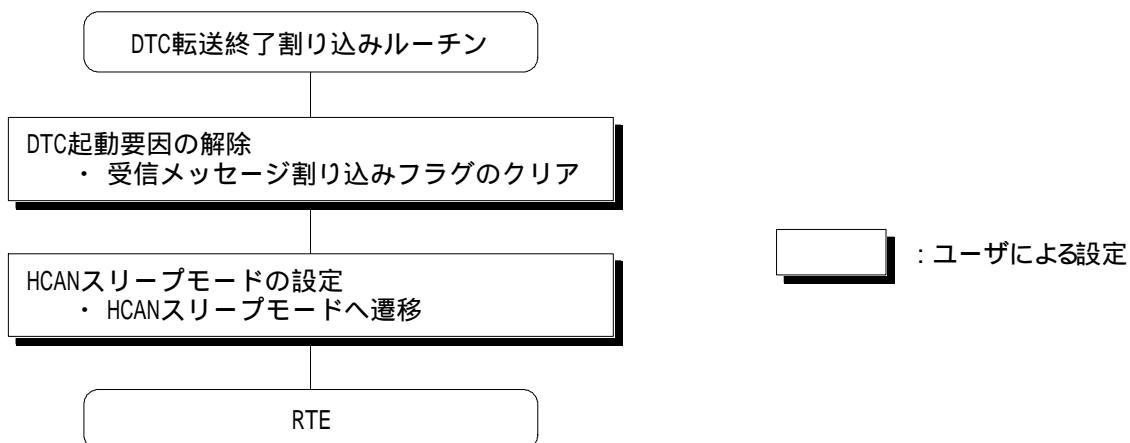


図2.9 DTC転送終了割り込みのフローチャート

2 HCAN送受信(例題 2 .)	対象	H8S/2623F	使用機能	HCAN, DTC
-------------------	----	-----------	------	-----------

2.6 送信プログラムリスト

```

/*****
/*          HCAN 送信プログラム          */
/*****
#include <stdio.h>                /* ライブ 関数用ヘッダ ファイル */
#include <machine.h>              /* ライブ 関数用ヘッダ ファイル */
#include "2623.h"                 /* 周辺レジスタ定義ヘッダ ファイル */
/*****
/*          関数プロトコル宣言          */
/*****
void main( void );
/*****
/*          定数定義          */
/*****
#define COUNT (*(unsigned long *)0xFFC000)
/*****
/*          メインルーチン          */
/*****
void main(void)
{
/* 初期設定 */
MSTPCRC = 0xF7;                 /* HCANレジスタストップモードの解除 */
HCAN_IRR = 0x0100;              /* HCANレジスタ用リセットフラグの初期化 */
HCAN_BCR = 0x0334;              /* ビットレート 250kbps */
HCAN_MBCR = 0xFDFF;             /* メールボックス1を送信用に設定 */
for( COUNT = 0; COUNT < 128; COUNT++ ) /* メールボックス(RAM)の初期設定 */
{
(char*)&HCAN_MCO_1 + COUNT) = 0x00;
}
for( COUNT = 0; COUNT < 128; COUNT++ ) /*メールボックス(RAM)の初期設定 */
{
(char*)&HCAN_MDO_1 + COUNT) = 0x00;
}
HCAN_MCR = 0x04; /* メールボックスの番号順で送信、コンフィギュレーションモードの解除 */
/* 送信データの設定 */
HCAN_MC1_5 = 0xA0; /* スタンドフォーマット、データレム、Identifierの設定 */
HCAN_MC1_6 = 0xAA; /* Identifierの設定 */
HCAN_MC1_1 = 0x08; /* データ長：8ビット */
HCAN_MD1_1 = 0x55; /* メッセージの内容：01010101 */
HCAN_MD1_2 = 0x66; /* メッセージの内容：01100110 */
HCAN_MD1_3 = 0x77; /* メッセージの内容：01110111 */
HCAN_MD1_4 = 0x88; /* メッセージの内容：10001000 */
HCAN_MD1_5 = 0x99; /* メッセージの内容：10011001 */
HCAN_MD1_6 = 0xAA; /* メッセージの内容：10101010 */
HCAN_MD1_7 = 0xBB; /* メッセージの内容：10111011 */
HCAN_MD1_8 = 0xFF; /* メッセージの内容：11111111 */
/* メッセージの送信 */
HCAN_TXPR = 0x0200; /* メールボックス1を送信待ち状態に設定 */
while((HCAN_TXACK & 0x0200) != 0x0200);
/* 送信完了フラグのクリア */
HCAN_TXACK &= 0x0200; /* 送信完了フラグのクリア */
while(1);
}

```

2 HCAN送受信(例題 2 .)	対象	H8S/2623F	使用機能	HCAN, DTC
-------------------	----	-----------	------	-----------

2.6 受信プログラムリスト

```

/*****
/*          HCAN 受信プログラム          */
/*****
#include <stdio.h>                                /* ライブリ関数用ヘッダファイル */
#include <machine.h>                              /* ライブリ関数用ヘッダファイル */
#include "2623.h"                                /* 周辺レジスタ定義ヘッダファイル */
/*****
/*          関数プロトコル宣言          */
/*****
void main( void );
/*****
/*          定数定義          */
/*****
#define COUNT      (*(unsigned short *)0xFFC100)
#define Message_Box1 (*(volatile unsigned char *)0xFFC000) /* 受信データ1格納 */
#define Message_Box2 (*(volatile unsigned char *)0xFFC001) /* 受信データ2格納 */
#define Message_Box3 (*(volatile unsigned char *)0xFFC002) /* 受信データ3格納 */
#define Message_Box4 (*(volatile unsigned char *)0xFFC003) /* 受信データ4格納 */
#define Message_Box5 (*(volatile unsigned char *)0xFFC004) /* 受信データ5格納 */
#define Message_Box6 (*(volatile unsigned char *)0xFFC005) /* 受信データ6格納 */
#define Message_Box7 (*(volatile unsigned char *)0xFFC006) /* 受信データ7格納 */
#define Message_Box8 (*(volatile unsigned char *)0xFFC007) /* 受信データ8格納 */
#define SAR          (*(volatile unsigned long *)0xFFEC00) /* DTCLレジスタ情報設定 */
#define MRA          (*(volatile unsigned char *)0xFFEC00) /* DTCLレジスタ情報設定 */
#define DAR          (*(volatile unsigned long *)0xFFEC04) /* DTCLレジスタ情報設定 */
#define MRB          (*(volatile unsigned char *)0xFFEC04) /* DTCLレジスタ情報設定 */
#define CRA          (*(volatile unsigned short *)0xFFEC08) /* DTCLレジスタ情報設定 */
#define CRB          (*(volatile unsigned short *)0xFFECA0) /* DTCLレジスタ情報設定 */
/*****
/*          メインルーチン          */
/*****
void main(void)
{
/* DTC初期設定 */
MSTPCRA = 0x3F;                                /* DTCレジスタストップモードの解除 */
SAR = (long)&HCAN_MDO_1;                       /* 転送元アドレスの設定 */
MRA = 0xA8;                                    /* SAR, DARは転送後インクリメント、ブロック転送モードの設定 */
DAR = (long)&Message_Box1;                     /* 転送先アドレス(内蔵RAM)の設定 */
MRB = 0x00;                                    /* DTCによるデータ転送終了後、割り込みを許可 */
CRA = 0x0808;                                  /* ブロック転送は8バイト単位に設定 */
CRB = 0x0001;                                  /* ブロック転送回数: 1回の設定 */
DTC_DTCERG |= 0x20;                            /* HCAN割り込み(RMO)によるDTC起動の許可 */
/* HCAN初期設定 */
MSTPCRC = 0xF7                                /* HCANレジスタストップモードの解除 */
HCAN_IRR = 0x0100;                            /* HCANレジスタ用リセットフラグの初期化 */
HCAN_BCR = 0x0334;                            /* ビットレート 250Kbps */
HCAN_MBCR = 0x0100;                           /* メールボックス0を受信に設定 */
for( COUNT = 0; COUNT < 128; COUNT++ )      /* メールボックス(RAM)の初期設定 */
{
(char*)&HCAN_MCO_1 + COUNT) = 0x00;
}
for( COUNT = 0; COUNT < 128; COUNT++ )      /* メールボックス(RAM)の初期設定 */
{
(char*)&HCAN_MDO_1 + COUNT) = 0x00;
}
HCAN_MCR &= 0xFE;                             /* コンフィグレーションモードの解除 */

```

2 HCAN送受信(例題 2 .)	対象	H8S/2623F	使用機能	HCAN, DTC
-------------------	----	-----------	------	-----------

2.6 受信プログラムリスト

```

/* 割り込みの設定 */
HCAN_MBIMR = 0xFEFF;          /* メルホックス0の割り込み要求許可 */
HCAN_IMR = 0xFCEF;          /* メッセージ受信、バス動作割り込みを許可 */
SYSCR |= 0x20;              /* 割り込み制御モード2の設定 */
INTC.IPRM = 0x70;          /* HCAN割り込みの優先レベルを7に設定 */
set_imask_exr(0);          /* 割り込み要求マスクレベルの指定 */
/* 受信データの設定 */
HCAN_MCO_5 = 0xA0;          /* スタンドフォーマット、データフレーム、Identifier設定 */
HCAN_MCO_6 = 0xAA;          /* Identifierの設定 */
HCAN_LAFMH = 0x0000;        /* メルホックス0はbitの一致でデータを格納 */
/* HCANスリープモードの設定 */
HCAN_MCR |= 0xA0;          /* HCANスリープモードへ遷移、バス動作による解除を許可 */
while(1);
}
/*****
/*          バス動作割り込みルーチン
*****/
#pragma interrupt(OVRO_IRR12)
void OVRO_IRR12(void)
{
    HCAN_IRR &= 0x0010;      /* IRR12(バス動作割り込みフラグ)のクリア */
}
/*****
/*          DTC転送終了割り込みルーチン
*****/
#pragma interrupt(DTCend_RM0)
void DTCend_RM0(void)
{
    HCAN_RXPR &= 0xFFFF;    /* IRR1(受信メッセージ割り込みフラグ)のクリア */
/* HCANスリープモードの設定 */
    HCAN_MCR |= 0x20;        /* HCANスリープモードへ遷移 */
}

```

2 HCAN送受信(例題 2 .)	対象	H8S/2623F	使用機能	HCAN, DTC
-------------------	----	-----------	------	-----------

2.7 備考

- (* 1) データフレーム : 送信元から送信先へ転送したいデータです。
- (* 2) アービトレーションフィールド : メッセージ固有のIDおよびデータフレームまたはリモートフレームの設定をします。
- (* 3) コントロールフィールド : 転送するデータ長およびスタンダードフォーマットまたはエクステンデッドフォーマットの設定をします。
- (* 4) データフィールド : メッセージの内容(送信したいデータ)を設定します。
- (* 5) CRCフィールド : HCAN内部でSOFからデータフィールドのスタンプビットを除いたすべてのビットデータからCRCが自動生成され、送信メッセージのエラーを検出します。
15bitのCRCと1bitのCRCデリミタから成ります。
CRCデリミタはCRC後に必ず " 1 " が出力されます。

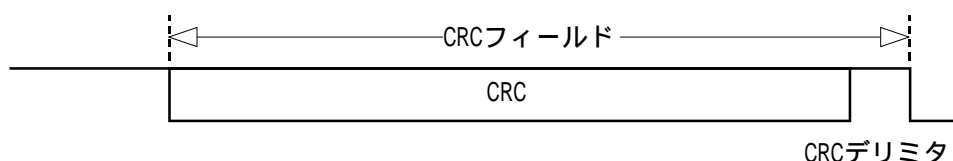


図2.10 CRCフィールド

CRCについて

伝送しようとするデータ多項式 $P(X)$ に X^R を乗じ、さらにこの $X^R \cdot P(X)$ を生成多項式 $G(X)$ で除し、余りである $R(X)$ を求めます。情報を送り出す側では、 $X^R \cdot P(X)$ に求められた $R(X)$ をチェックビットとして付加し電信情報 $Tx(X)$ として送りだします。

また、情報を受ける側では受けた受信情報 $Rx(X)$ を生成多項式 $G(X)$ で除し、余りを求めます。この余りがゼロとなった場合は、情報伝送は正しく行われたとみなし、余りが出た場合は伝送された情報に誤りがあったと判断します。

$P(X) = \{ \text{スタンプビットを除いたSOFからデータフィールドまで} \}$

$G(X) = X^{15} + X^{14} + X^{10} + X^8 + X^7 + X^4 + X^3 + 1$

$R = 15$

$G(X)$ はCRCを生成する多項式としてCANプロトコルにて規定されています。

例として以下の設定において送信されたデータフレームについて説明します。

設定値

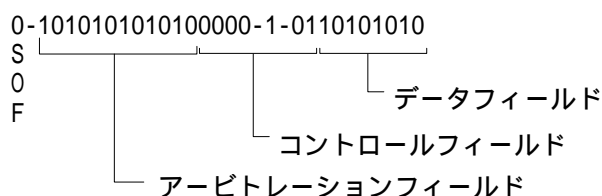
SOF : 0

アービトレーションフィールド : 101010101010

コントロールフィールド : 000001

データフィールド : 10101010

送信されるデータのSOFからデータフィールドは(スタンプビット)



で、対象となるデータはスタンプビットを除いた

010101010101000000110101010

すなわち、 $P(X) = X^{25} + X^{23} + X^{21} + X^{19} + X^{17} + X^{15} + X^8 + X^7 + X^5 + X^3 + X^1$ となります。

2.7 備考

ここで、P(X) に X^{15} を掛け、
 $X^{15} \cdot P(X) = X^{40} + X^{38} + X^{36} + X^{34} + X^{32} + X^{30} + X^{23} + X^{22} + X^{20} + X^{18} + X^{16}$
とし、この値を
 $G(X) = X^{15} + X^{14} + X^{10} + X^8 + X^7 + X^4 + X^3 + 1$
で除します。

$$\begin{aligned}
 &X^{15} \cdot P(X) = X^{40} + X^{38} + X^{36} + X^{34} + X^{32} + X^{30} + X^{23} + X^{22} + X^{20} + X^{18} + X^{16} \\
 &P[40:0] = 10101010101000000110101010000000000000000 \\
 &G(X) = X^{15} + X^{14} + X^{10} + X^8 + X^7 + X^4 + X^3 + 1 \\
 &G[15:0] = 1100010110011001
 \end{aligned}$$

商

	10101010101000000110101010000000000000000
1100010110011001	1100010110011001
	1101111001110010
	1100010110011001
	1101111101011110
	1100010110011001
	1101011000111101
	1100010110011001
	1001110100100010
	1100010110011001
	1011000101110110
	1100010110011001
	1110100111011110
	1100010110011001
	1011000100011100
	1100010110011001
	1110100100001010
	1100010110011001
	1011001001001100
	1100010110011001
	1101101010101010
	1100010110011001
	1111100110011000
	1100010110011001
	111100000000010

← 余り : R[14:0]

除数と非除数で
EORをとります。

計算の結果、次の値がえられます。
 $R[14:0] = 111100000000010$
つまり、これがCRCの値となり、データフィールド後に付加され電信情報Tx(X)として送信されます。
実際にはスタッフビット()とCRCデリミタ()が付加され、
11110000010000101がCRCフィールド上で送信されます。

2.7 備考

次に、受信した受信情報Rx(X)に誤りがあるか計算してみます。
 Rx(X)とは、前記の計算により求めた余りを $X^{15} \cdot P(X)$ に加算した値であり、この値をG(X)で除します。

$$\left(\begin{array}{l} P [40:0]=1010101010100000011010101000000000000000 \\ \oplus R[14:0]= \hspace{18em} 11110000000010 \\ \hline 1010101010100000011010101011110000000010 \end{array} \right)$$

↙

この値をG[15:0]で除します。

商

除数と非除数で
EORをとります。

↗

	1010101010100000011010101011110000000010	1100010110011001
	1101111001110010	
	1100010110011001	
	110111101011110	
	1100010110011001	
	1101011000111101	
	1100010110011001	
	1001110100100010	
	1100010110011001	
	1011000101110111	
	1100010110011001	
	1110100111011101	
	1100010110011001	
	1011000100010011	
	1100010110011001	
	1110100100010100	
	1100010110011001	
	1011001000110100	
	1100010110011001	
	1110111101011010	
	1100010110011001	
	1010101100001100	
	1100010110011001	
	1101110100101010	
	1100010110011001	
	1100010110011001	
	1100010110011001	
	1100010110011001	

0	←	余り
---	---	----

以上の計算から余りはゼロとなり情報伝達は正しいとみなされます。

2 HCAN送受信(例題 2 .)	対象	H8S/2623F	使用機能	HCAN, DTC
-------------------	----	-----------	------	-----------

2.7 備考

- (* 6) ACKフィールド : 正常受信確認用です。
1bitのACKスロット、1bitのACKデリミタから成ります。

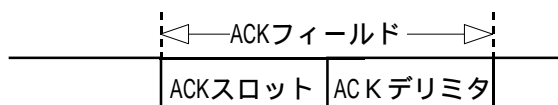


図2.11 ACKフィールド

・H8S/2623F受信側がCRCチェックでエラーを検出した場合はACKスロット=Highを、検出しなかった場合はACKスロット=Lowを出力します。

- (* 7) スタッフビット : データフレーム上でLowが5bit続いた場合、次の1bitは必ずHighが出力されます。同様にHighが5bit続いた場合、次の1bitは必ずLowが出力されます。これらの様にスタッフビットが出力される場合データフレームのビット長は、スタッフビットの分だけ長くなります。

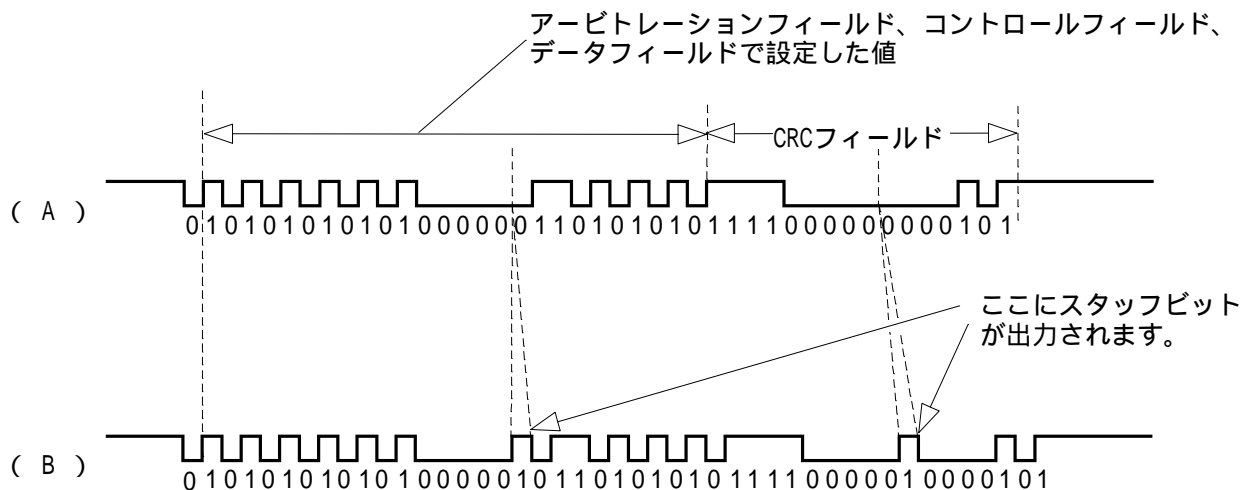
例として、設定値を

アービトレーションフィールド : " 101010101010 "

コントロールフィールド : " 000001 "

とすると、" 101010101010000001 110101010111100000000000101 " となり、最下位から2bit目 (0が5つ続いた後) にスタッフビット () が出力されます。

よって、CANバス上に送信される値は " 1010101010100000101 " となり、スタッフビットの1bit分だけ長くなります。



(A)の送信したいデータフレームに対して、CANバス上ではスタッフビットの乗ったデータフレーム (B)が送信されます。また、(B)ではデータフレームの長さもスタッフビットの分だけ長くなっています。

図2.12 スタッフビット

- (* 8) コンフィギュレーションモード : HCANモジュールがリセット状態のことであり、解除はマスタコントロールレジスタ (MCR) のリセットリクエストビット (MCR0) をクリアすることで行います。

3.HCAN送受信 (例題 3.) :エクステンッドフォーマット、1バイトデータ

第3章 目次

3.1 仕様	39
3.2 使用機能説明	41
3.3 動作説明	42
3.4 ソフトウェア説明	44
3.5 フローチャート	46
3.6 プログラムリスト	50
3.7 備考	53

3.1 仕様

- (1) データフレーム (* 1) の送受信を行います。(H8S/2623Fを2個使用)
 図3.1にデータフレームの仕様を示します。
 (a) SOF : データフレームの開始を示します。
 (b) アービトレーションフィールド (* 2)
 ・ 11ビット Identifier = " 10101010101 "
 ・ SRR (Substitute Remote Request) = " 1 " : エクステンデッドフォーマットの選択
 ・ IDE (ID Extension) = " 1 " : エクステンデッドフォーマットの選択
 ・ 18ビット Identifier = " 1010101010101010 "
 ・ RTR = " 0 " : データフレームの選択
 (c) コントロールフィールド (* 3) : " 000001 " とします。
 ・ R0 = " 0 " : リザーブビット
 ・ R1 = " 0 " : リザーブビット
 ・ DLC = " 0001 " : 1バイトのデータ長の設定
 (d) データフィールド (* 4) : " 10101010 " とします。
 (e) CRCフィールド (* 5) : HCAN内部でCRCが自動生成されます。
 (f) ACKフィールド (* 6) : 送信側では " 11 " 、受信側では " 01 (正常時) " が出力されます。
 (g) EOF : 送信/受信データフレームの完了を示します。
- (2) 通信速度は1Mbps (20MHz動作時) とします。
 (3) データ長は、1バイトとします。
 (4) メッセージの送信はメールボックス1を用います。
 (5) メッセージの受信はメールボックス0を用います。メッセージの受信方式としてはIdentifierをマスクして一致すれば受信とします。
 (6) 割り込み制御モード2を選択し、
 (a) バス動作割り込み(OVRO)を使用します。
 (b) メッセージ受信割り込み(RMO)を使用します。
 (7) 受信メッセージは、内蔵RAMに格納します。
 (8) HCANスリープモードを使用します。
 (a) 初期設定後、HCANスリープモードへ遷移します。(CANバス動作により解除。)
 (b) 受信処理後、HCANスリープモードへ遷移します。(CANバス動作により解除。)
 (9) 図3.2にCANバスの接続例を示します。

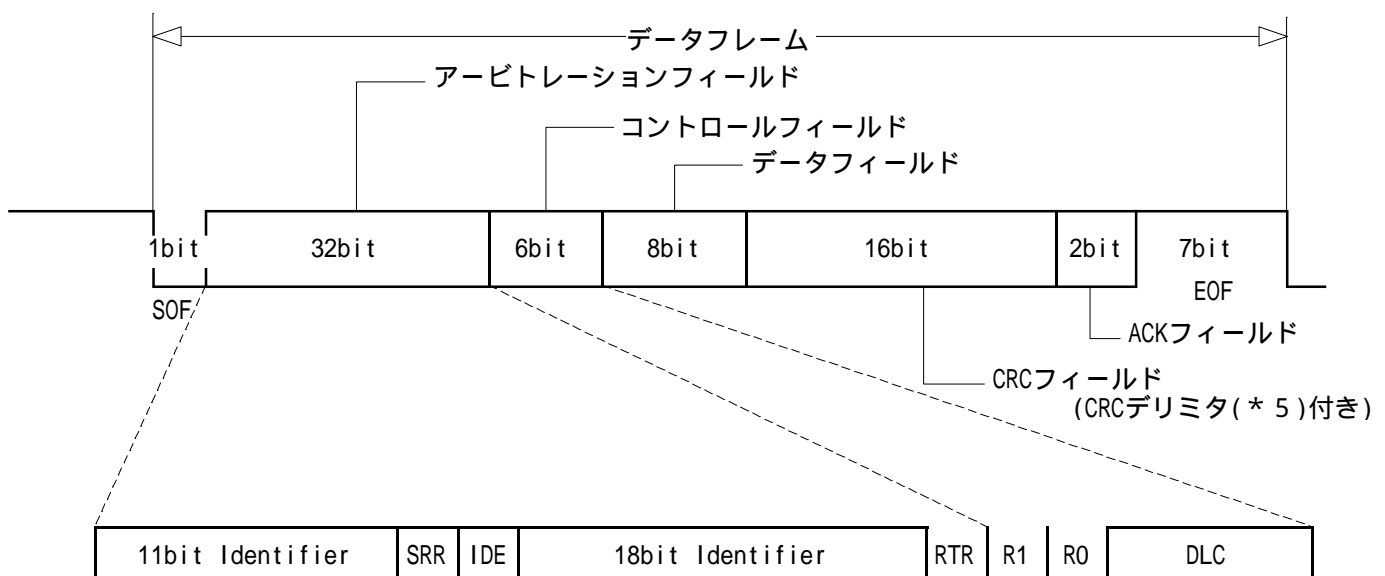
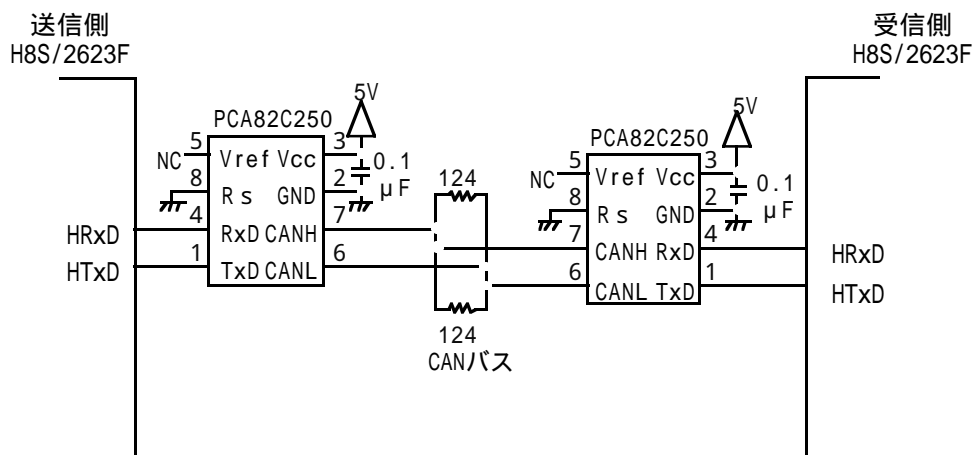


図3.1 データフレームの仕様

3 HCAN送受信(例題 3 .)	対象	H8S/2623F	使用機能	HCAN
-------------------	----	-----------	------	------

3.1 仕様



本LSIとCANバスを接続するためにはバストランシーバICが必要になります。Philips社製PCA82C50とコンパチブルなものを推奨します。

図3.2 H8S/2623Fを用いたCANインタフェース

3 HCAN送受信(例題 3 .)	対象	H8S/2623F	使用機能	HCAN
-------------------	----	-----------	------	------

3.2 使用機能説明

表3.1~3.3に本タスク例の機能割り付けを示します。本タスク例は表3.1~3.3に示すようにH8S/2623Fに内蔵しているHCANの機能を割り付け、HCANの送受信を行います。

表3.1 HCAN機能割り付け

HCANレジスタ		機能
端子	HTxD	メッセージの送信を行います。
	HRxD	メッセージの受信を行います。
送受信 共通 レジスタ	IRR	各割り込み要因のステータスを示します。
	BCR	CANのボーレートプリスケアラ、ビットタイミングパラメータを設定します。
	MBCR	メールボックスの送信/受信を設定します。
	MCR	CANインタフェースの制御を行います。
	MCO_1 s MC15_8	アービトレーションフィールド、コントロールフィールドの設定を行います。
	MDO_1 s MD15_8	データフィールドの設定を行います。
送信用 レジスタ	TXPR	送信メッセージをメールボックスに格納後送信待ち状態を設定します。
	TXACK	対応するメールボックスの送信メッセージが正常に送信されたことを示します。
受信用 レジスタ	MBIMR	メールボックス0を割り込み要求許可に設定します。
	IMR	受信メッセージ、バス動作割り込み許可の設定を行います。
	LAFMH,L	受信用メールボックス0のIdentifier用フィルタマスクの設定を行います。
	RXPR	受信メッセージ割り込みフラグをクリアします。

表3.2 MSTPCR機能割り付け

MSTPCRレジスタ	機能
MSTPCRC	モジュールストップモードの制御を行います。

表3.3 割り込みコントローラ割り付け

IPRレジスタ	機能
IPRM	割り込み優先レベルの制御を行います。
SYSCR	割り込み制御モードの設定を行います。
exr	割り込み要求マスクレベルの指定を行います。

3.3 動作説明(送信時)

図3.3に送信時の動作原理を示します。図3.3に示すようにH8S/2623Fのハードウェア処理およびソフトウェア処理によりHCANの送信を行います。

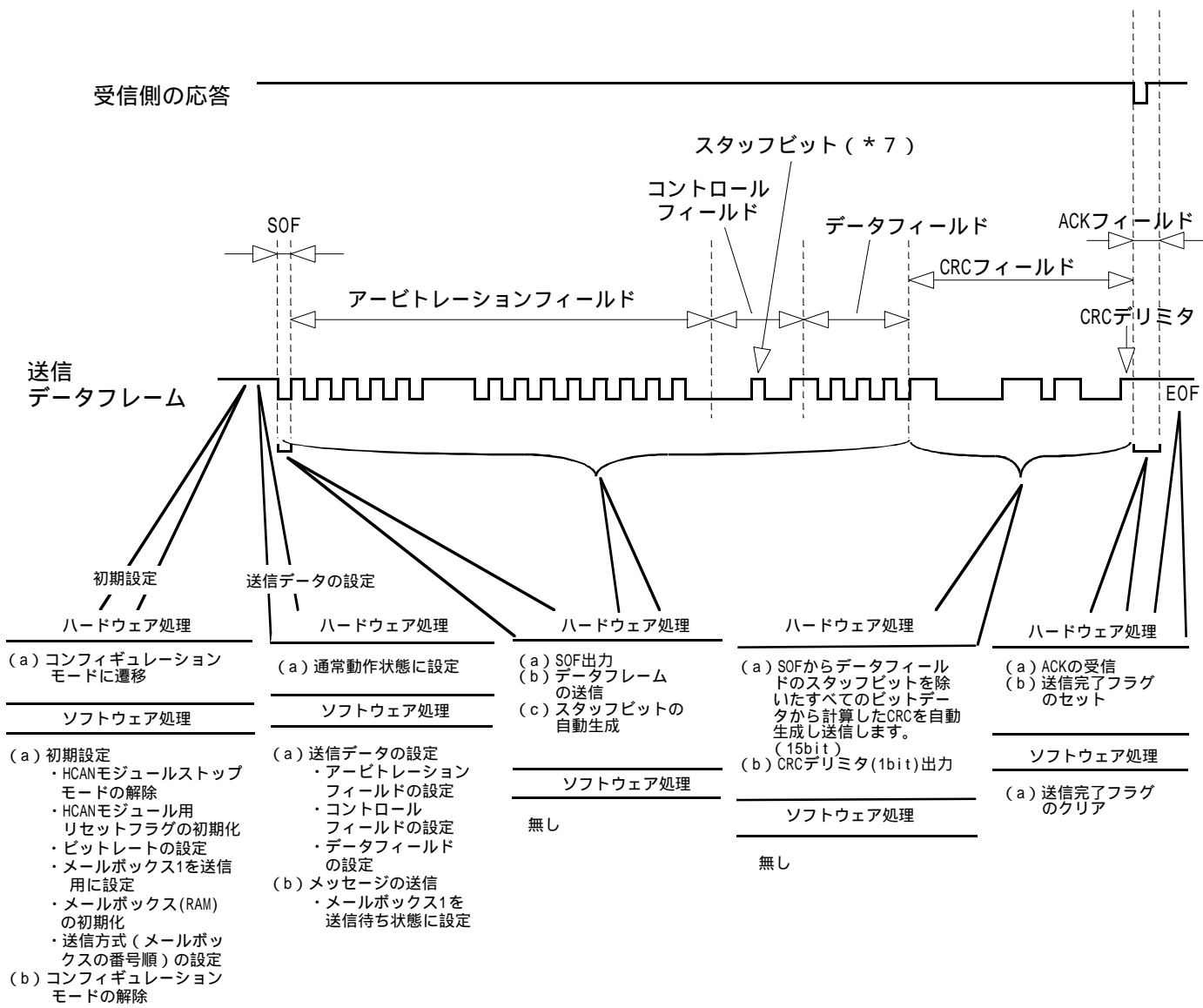


図3.3 HCAN送信時の動作

(*) は「3.7 備考」を参照してください。

3.3 動作説明(受信時)

図3.4に受信時の動作原理を示します。図3.4に示すようにH8S/2623Fのハードウェア処理およびソフトウェア処理によりHCANの受信を行います。

受信側の応答

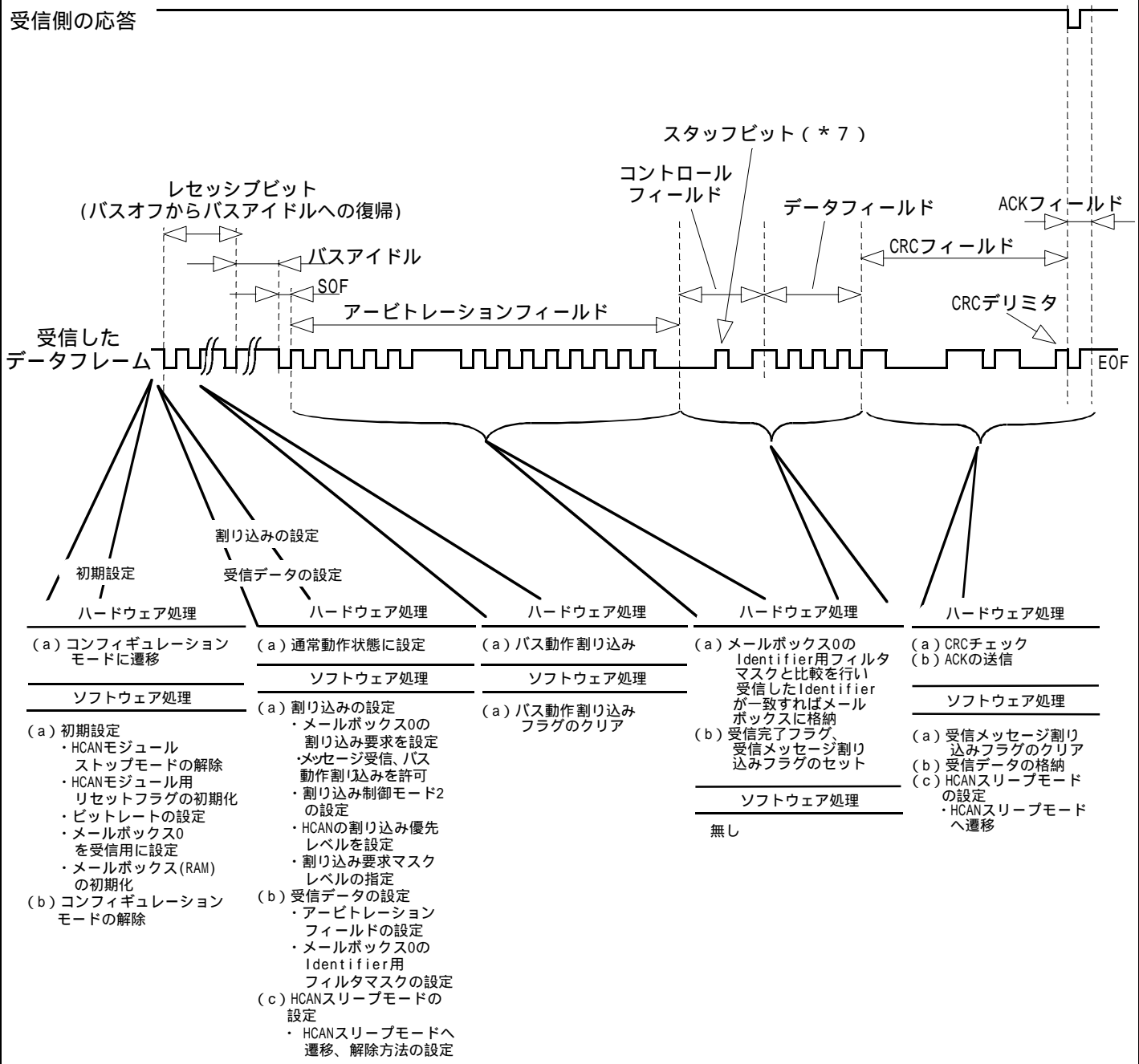


図3.4 HCAN受信時の動作

(*) は「3.7 備考」を参照してください。

3 HCAN送受信(例題3.)	対象	H8S/2623F	使用機能	HCAN
------------------	----	-----------	------	------

3.4 ソフトウェア説明

(1) モジュール説明

モジュール名	ラベル名	機能
メインルーチン	main	HCANの初期設定および送信、受信の設定を行います。
バス動作 割り込みルーチン	OVR0_IRR12	バス動作割り込みフラグのクリアを行います。
メッセージ受信 割り込みルーチン	RMO	メールボックス0に格納したメッセージを内蔵RAMに転送します。

(2) 使用変数の説明

ラベル名	機能	データ長	使用モジュール名
COUNT	HCAN_MCO_1 ~ MC15_8, HCAN_MDO_1 ~ MD15_8を初期化します。	unsigned_short	メインルーチン

(3) 仕様内部レジスタの説明

レジスタ名	機能	設定値	使用モジュール名	
送受信共通設定				
MSTPCRC	HCANモジュールストップモードを解除します。	0xF7	メインルーチン	
HCAN_BCR	HCANのビットレートを1Mbpsに設定します。	0x0025		
送信時の設定				
HCAN_IRR	HCANモジュール用リセットフラグを初期化します。	0x0100		
HCAN_MBCR	メールボックス1を送信用に設定します。	0xFDFF		
HCAN_MCR	メールボックスの番号順で送信するように設定および、リセットリクエストビットをクリアします。	0x04		
HCAN_MC1_1	1バイトのデータ長を設定します。	0x01		
HCAN_MC1_5	メールボックス1のデータフレーム、エクステンデッドフォーマットを選択および、Identifierを設定します。	0xAA		
HCAN_MC1_6	メールボックス1のIdentifierを設定します。	0xAA		
HCAN_MC1_7	メールボックス1のIdentifierを設定します。	0xAA		
HCAN_MC1_8	メールボックス1のIdentifierを設定します。	0xAA		
HCAN_MD1_1	メールボックス1の送信データを設定します。	0xAA		
HCAN_TXPR	メールボックス1を送信待ち状態に設定します。	0x0200		
HCAN_TXACK	データフレーム送信完了フラグをクリアします。	0x0200		

3 HCAN送受信(例題3. .)	対象	H8S/2623F	使用機能	HCAN
-------------------	----	-----------	------	------

3.4 ソフトウェア説明

レジスタ名	機能	設定値	使用モジュール名	
受信時の設定				
HCAN_IRR	HCANモジュール用リセットフラグを初期化します。	0x0100	メインルーチン	
	バス動作割り込みフラグをクリアします。	0x0010	バス動作割り込み	
HCAN_MCR	リセットリクエストビットをクリアします。	0xFE	メインルーチン	
	HCANスリープモードへ遷移、バス動作による解除を設定します。	0xA0		
	HCANスリープモードへ遷移を設定します。	0x20	受信割り込み	
HCAN_MBCR	メールボックス0を受信用に設定します。	0x0100	メインルーチン	
HCAN_MCO_5	メールボックス0のデータフレーム、エクステンデッドフォーマットを選択および、Identifierを設定します。	0xAA		
HCAN_MCO_6	メールボックス0のIdentifierを設定します。	0xAA		
HCAN_MCO_7	メールボックス0のIdentifierを設定します。	0xAA		
HCAN_MCO_8	メールボックス0のIdentifierを設定します。	0xAA		
HCAN_LAFMH	メールボックス0のIdentifier用フィルタマスクを設定します。	0x0000		
HCAN_LAFML	メールボックス0のIdentifier用フィルタマスクを設定します。	0x0000		
HCAN_MBIMR	メールボックス0の割り込み要求許可に設定します。	0xFEFF		
HCAN_IMR	メッセージ受信、バス動作割り込み要求許可に設定します。	0xFCEF		
INTC.IPRM	HCANの割り込み優先レベルを7に設定します。	0x70		
SYSCR	割り込み制御モード2に設定します。	0x20		
exr	割り込み要求マスクレベルを指定します。	0x00		
HCAN_RXPR	受信メッセージ割り込みフラグをクリアします。	0xFFFF		受信割り込み

(4) 使用RAM

シンボル名	機能	アドレス	使用モジュール名
受信時の設定			
Message_DATA	メールボックス0用データの格納先アドレス。	0xFFC100	メインルーチン

(注意)

受信割り込み : メッセージ受信割り込みルーチン

バス動作割り込み : バス動作割り込みルーチン

3.5 送信フローチャート

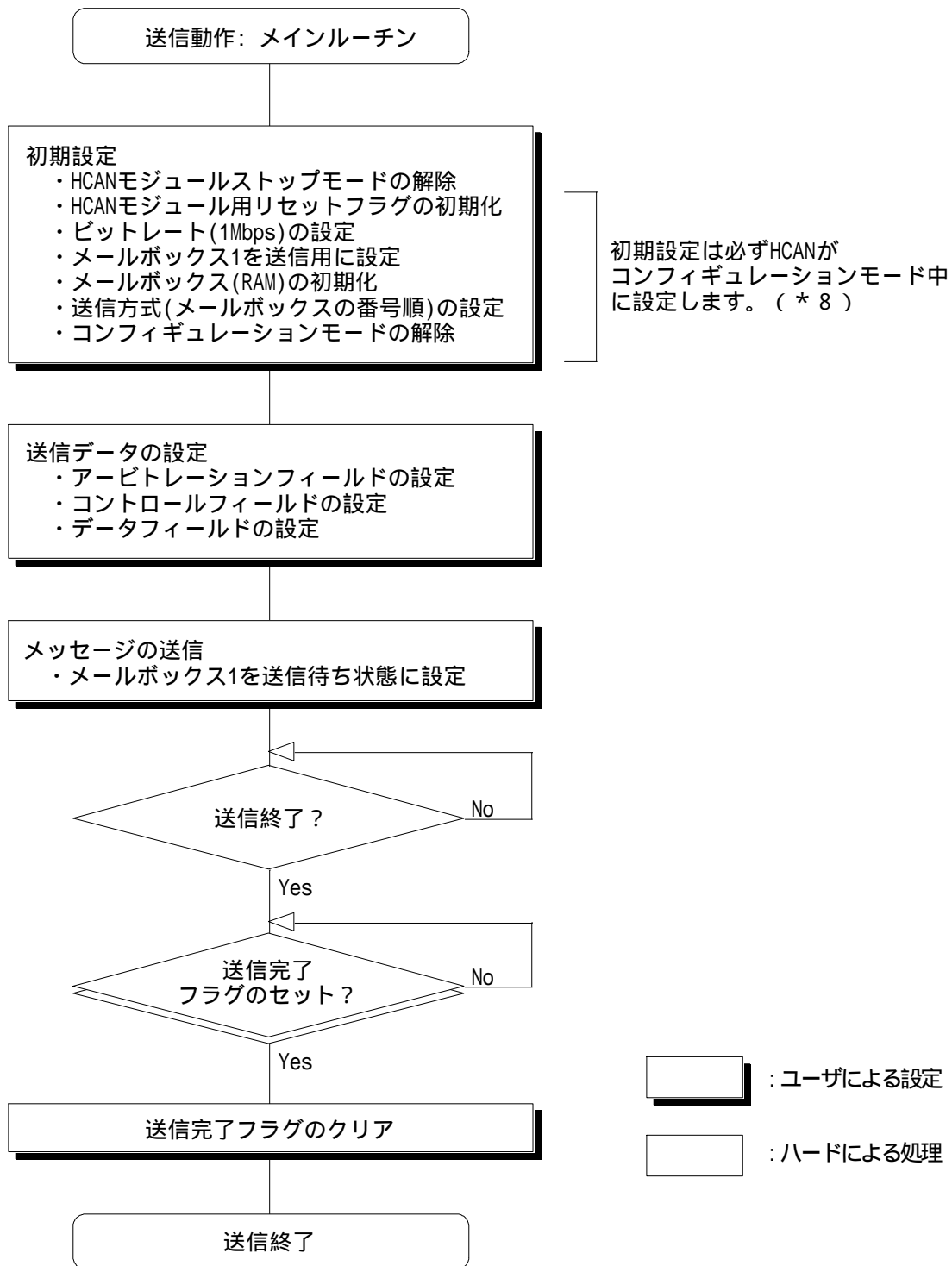


図3.5 送信時のフローチャート

(*) は「3.7 備考」を参照してください。

3.5 受信フローチャート

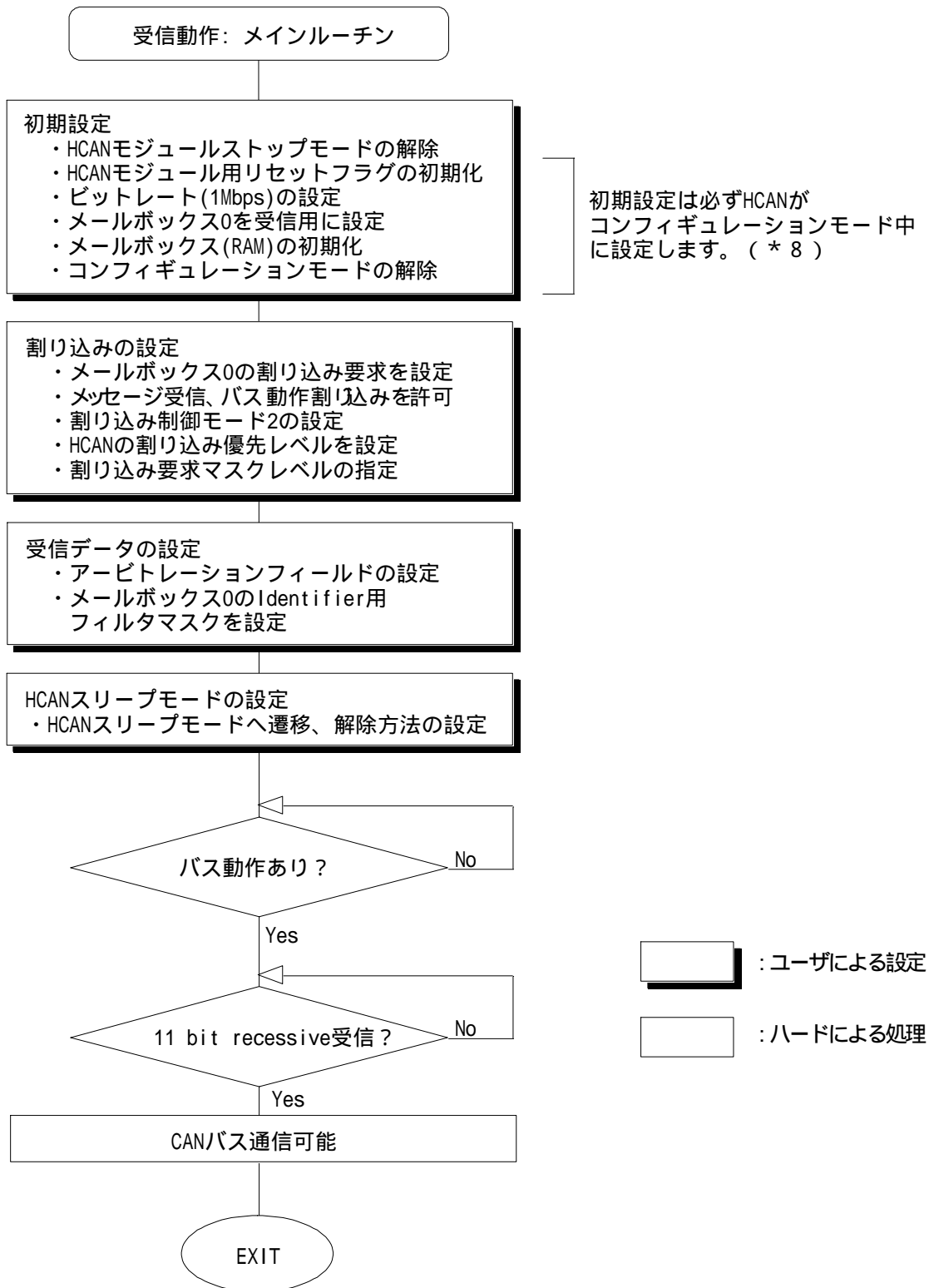


図3.6 受信時のフローチャート(1)

(*)は「3.7 備考」を参照してください。

3.5 受信フローチャート

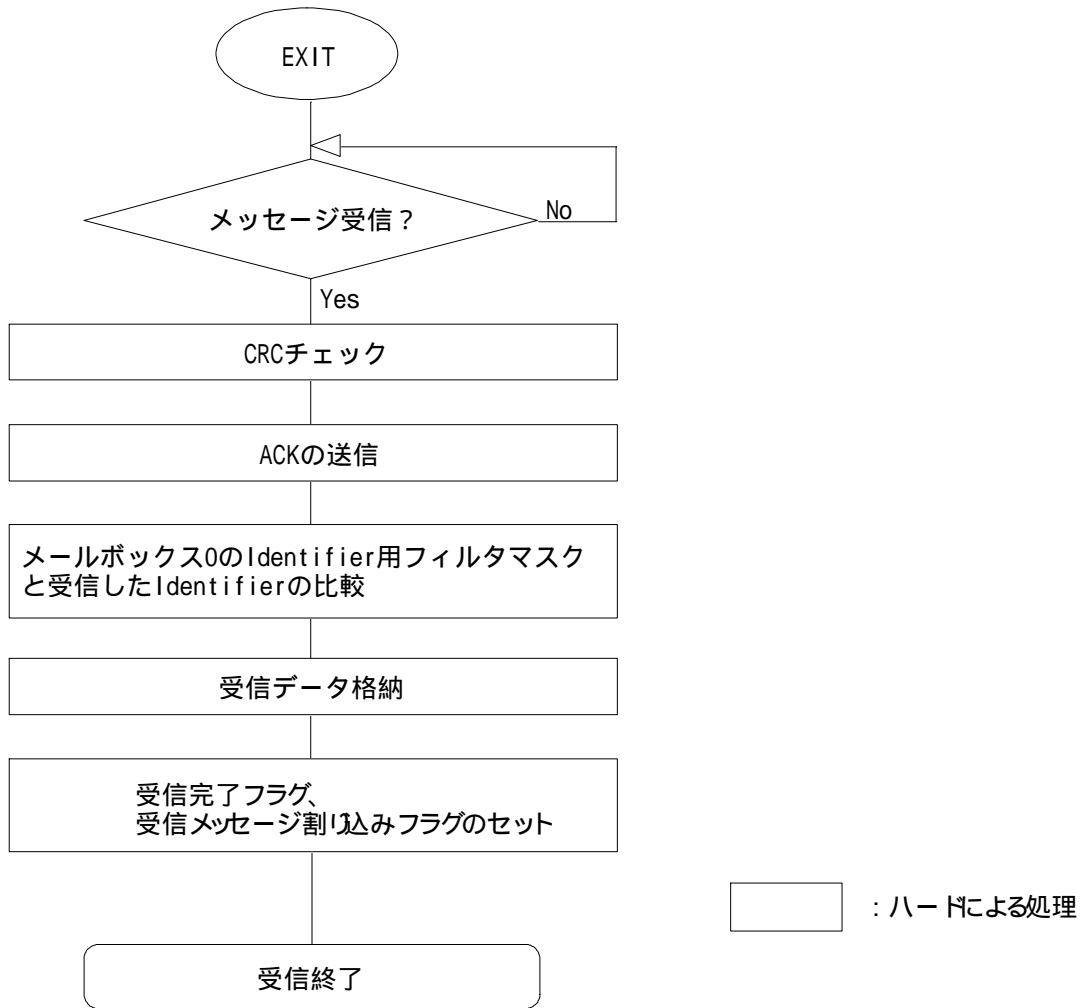


図3.7 受信時のフローチャート(2)

3.5 受信フローチャート

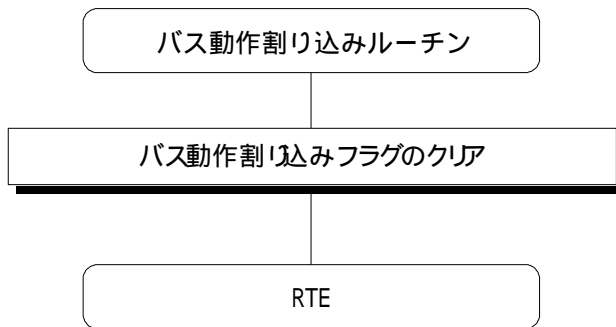


図3.8 バス動作割り込みのフローチャート


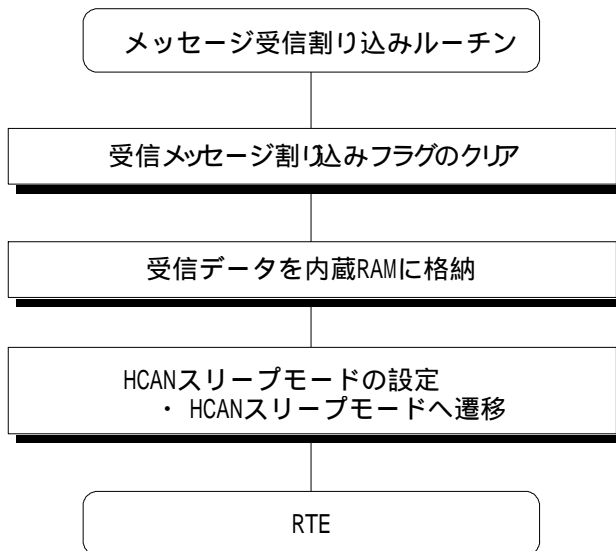
 : ユーザによる設定

図3.9 メッセージ受信割り込みのフローチャート

3 HCAN送受信(例題 3 .)	対象	H8S/2623F	使用機能	HCAN
-------------------	----	-----------	------	------

3.6 送信プログラムリスト

```

/*****
/*          HCAN 送信プログラム          */
/*****
#include <stdio.h>                /* ライブ 関数用ヘッダ ファイル */
#include <machine.h>              /* ライブ 関数用ヘッダ ファイル */
#include "2623.h"                 /* 周辺レジスタ定義ヘッダ ファイル */
/*****
/*          関数プロトコル宣言          */
/*****
void main( void );
/*****
/*          定数定義          */
/*****
#define COUNT (*(unsigned short *)0xFFC000)
/*****
/*          メインルーチン          */
/*****
void main(void)
{
/* 初期設定 */
MSTPCRC = 0xF7;                  /* HCANシユールストップ モード の解除 */
HCAN_IRR = 0x0100;              /* HCANシユール用リセットフラグ の初期化 */
HCAN_BCR = 0x0025;              /* ビットレート 1Mbps */
HCAN_MBCR = 0xFDFF;             /* メールボックス1を送信用に設定 */
for( COUNT = 0; COUNT < 128; COUNT++ ) /* メールボックス(RAM)の初期設定 */
{
*(char*)&HCAN_MCO_1 + COUNT) = 0x00;
}
for( COUNT = 0; COUNT < 128; COUNT++ ) /* メールボックス(RAM)の初期設定 */
{
*(char*)&HCAN_MDO_1 + COUNT) = 0x00;
}
HCAN_MCR = 0x04; /* メールボックスの番号順で送信、コンフィグレーションモード の解除 */
/* 送信データの設定 */
HCAN_MC1_5 = 0xAA; /* エクステンデッドフォーマット、データフレーム、Identifierの設定 */
HCAN_MC1_6 = 0xAA; /* Identifierの設定 */
HCAN_MC1_7 = 0xAA; /* Identifierの設定 */
HCAN_MC1_8 = 0xAA; /* Identifierの設定 */
HCAN_MC1_1 = 0x01; /* データ長：1バイト */
HCAN_MD1_1 = 0xAA; /* メッセージの内容：10101010 */
/* メッセージの送信 */
HCAN_TXPR = 0x0200; /* メールボックス1を送信待ち状態に設定 */
while((HCAN_TXACK & 0x0200) != 0x0200);
/* 送信完了フラグのクリア */
HCAN_TXACK &= 0x0200; /* 送信完了フラグ のクリア */
while(1);
}

```

3.6 受信プログラムリスト

```

/*****
/*          HCAN 受信プログラム          */
/*****
#include <stdio.h>                /* ライブ 関数用ヘッダ ファイル */
#include <machine.h>              /* ライブ 関数用ヘッダ ファイル */
#include "2623.h"                 /* 周辺レジスタ定義ヘッダ ファイル */
/*****
/*          関数プロトコル宣言          */
/*****
void main( void );
/*****
/*          定数定義          */
/*****
#define COUNT          (*(unsigned short *)0xFFC000)
#define Message_DATA  (*(unsigned char *)0xFFC100)    /* 受信データ格納 */
/*****
/*          メインルーチン          */
/*****
void main(void)
{
/* 初期設定 */
MSTPCR = 0xF7;                    /* HCANレジスタストップモードの解除 */
HCAN_IRR = 0x0100;                /* HCANレジスタ用リセットフラグの初期化 */
HCAN_BCR = 0x0025;                /* ビットレート 1Mbps */
HCAN_MBCR = 0x0100;               /* メールボックス0を受信に設定 */
for( COUNT = 0; COUNT < 128; COUNT++ ) /* メールボックス(RAM)の初期設定 */
{
*(char*)&HCAN_MCO_1 + COUNT) = 0x00;
}
for( COUNT = 0; COUNT < 128; COUNT++ ) /* メールボックス(RAM)の初期設定 */
{
*(char*)&HCAN_MDO_1 + COUNT) = 0x00;
}
HCAN_MCR &= 0xFE;                 /* コンフィグレーションモードの解除 */
/* 割り込みの設定 */
HCAN_MBIMR = 0xFEFF;              /* メールボックス0の割り込み要求許可 */
HCAN_IMR = 0xFCEF;               /* メッセージ受信割り込み、バス動作割り込みを許可 */
SYSCR |= 0x20;                    /* 割り込み制御モード2の設定 */
INTC.IPRM = 0x70;                 /* HCAN割り込み優先レベルを7に設定 */
set_imask_exr(0);                 /* 割り込み要求マスクレベルの指定 */
/* 受信データの設定 */
HCAN_MCO_5 = 0xAA;                /* エクステンデッドフォーマット、データフレーム、Identifierの設定 */
HCAN_MCO_6 = 0xAA;                /* Identifierの設定 */
HCAN_MCO_7 = 0xAA;                /* Identifierの設定 */
HCAN_MCO_8 = 0xAA;                /* Identifierの設定 */
HCAN_LAFMH = 0x0000;              /* メールボックス0は全bitの一致でデータを格納 */
HCAN_LAFML = 0x0000;              /* メールボックス0は全bitの一致でデータを格納 */
/* HCANスリープモードの設定 */
HCAN_MCR |= 0xA0; /* HCANスリープモードへの遷移、バス動作による解除を許可 */
while(1);
}

```


3.6 受信プログラムリスト

```

/*****
/*          バス動作割り込みルーチン          */
/*****
#pragma interrupt(OVR0_IRR12)
void OVR0_IRR12(void)
{
    HCAN_IRR &= 0x0010;      /* IRR12(バス動作割り込みフラグ)のクリア */
}
/*****
/*          メッセージ受信割り込みルーチン          */
/*****
#pragma interrupt(RM0)
void RM0(void)
{
    HCAN_RXPR &= 0xFFFF;    /* IRR1(受信メッセージ割り込みフラグ)のクリア */
    Message_DATA = HCAN_MDO_1; /* 受信データの格納 */
/* HCANスリープモードの設定 */
    HCAN_MCR |= 0x20;      /* HCANスリープモードへ遷移 */
}

```

3 HCAN送受信(例題 3 .)	対象	H8S/2623F	使用機能	HCAN
-------------------	----	-----------	------	------

3.7 備考

- (* 1) データフレーム : 送信元から送信先へ転送したいデータです。
- (* 2) アービトレーションフィールド : メッセージ固有のIDおよびデータフレームまたはリモートフレームの設定をします。
- (* 3) コントロールフィールド : 転送するデータ長およびスタンダードフォーマットまたはエクステンデッドフォーマットの設定をします。
- (* 4) データフィールド : メッセージの内容(送信したいデータ)を設定します。
- (* 5) CRCフィールド : HCAN内部でSOFからデータフィールドのスタンプビットを除いたすべてのビットデータからCRCが自動生成され、送信メッセージのエラーを検出します。15bitのCRCと1bitのCRCデリミタから成ります。CRCデリミタはCRC後に必ず " 1 " が出力されます。

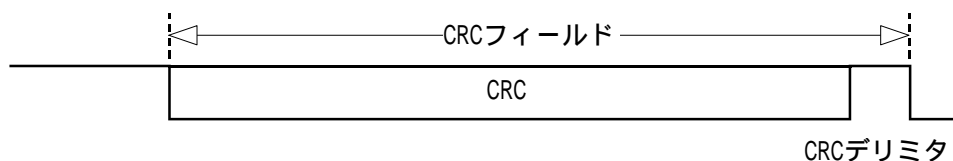


図3.10 CRCフィールド

CRCについて

伝送しようとするデータ多項式 $P(X)$ に X^R を乗じ、さらにこの $X^R \cdot P(X)$ を生成多項式 $G(X)$ で除し、余りである $R(X)$ を求めます。情報を送り出す側では、 $X^R \cdot P(X)$ に求められた $R(X)$ をチェックビットとして付加し電信情報 $Tx(X)$ として送りだします。

また、情報を受ける側では受けた受信情報 $Rx(X)$ を生成多項式 $G(X)$ で除し、余りを求めます。この余りがゼロとなった場合は、情報伝送は正しく行われたとみなし、余りが出た場合は伝送された情報に誤りがあったと判断します。

$$P(X) = \{ \text{スタンプビットを除いたSOFからデータフィールドまで} \}$$

$$G(X) = X^{15} + X^{14} + X^{10} + X^8 + X^7 + X^4 + X^3 + 1$$

$$R = 15$$

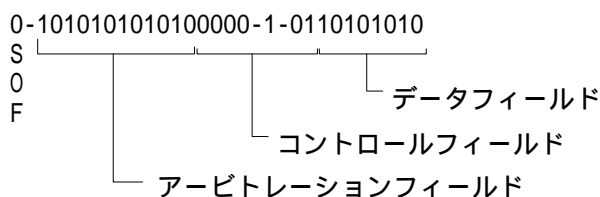
$G(X)$ はCRCを生成する多項式としてCANプロトコルにて規定されています。

例として以下の設定において送信されたデータフレームについて説明します。

設定値

SOF : 0
 アービトレーションフィールド : 101010101010
 コントロールフィールド : 000001
 データフィールド : 10101010

送信されるデータのSOFからデータフィールドは(スタンプビット)



で、対象となるデータはスタンプビットを除いた

010101010101000000110101010

すなわち、 $P(X) = X^{25} + X^{23} + X^{21} + X^{19} + X^{17} + X^{15} + X^8 + X^7 + X^5 + X^3 + X^1$ となります。

3.7 備考

ここで、P(X) に X^{15} を掛け、
 $X^{15} \cdot P(X) = X^{40} + X^{38} + X^{36} + X^{34} + X^{32} + X^{30} + X^{23} + X^{22} + X^{20} + X^{18} + X^{16}$
 とし、この値を
 $G(X) = X^{15} + X^{14} + X^{10} + X^8 + X^7 + X^4 + X^3 + 1$
 で除します。

$$\left(\begin{array}{l} X^{15} \cdot P(X) = X^{40} + X^{38} + X^{36} + X^{34} + X^{32} + X^{30} + X^{23} + X^{22} + X^{20} + X^{18} + X^{16} \\ P[40:0] = 1010101010100000011010101000000000000000 \\ G(X) = X^{15} + X^{14} + X^{10} + X^8 + X^7 + X^4 + X^3 + 1 \\ G[15:0] = 1100010110011001 \end{array} \right)$$

商

	1010101010100000011010101000000000000000
1100010110011001	1100010110011001
1101111001110010	1101111001110010
1100010110011001	1100010110011001
1101111101011110	1101111101011110
1100010110011001	1100010110011001
1101011000111101	1101011000111101
1100010110011001	1100010110011001
1001110100100010	1001110100100010
1100010110011001	1100010110011001
1011000101110110	1011000101110110
1100010110011001	1100010110011001
1110100111011110	1110100111011110
1100010110011001	1100010110011001
1011000100011100	1011000100011100
1100010110011001	1100010110011001
1110100100001010	1110100100001010
1100010110011001	1100010110011001
1011001001001100	1011001001001100
1100010110011001	1100010110011001
1110111110101010	1110111110101010
1100010110011001	1100010110011001
1010100011001100	1010100011001100
1100010110011001	1100010110011001
1101101010101010	1101101010101010
1100010110011001	1100010110011001
1111100110011000	1111100110011000
1100010110011001	1100010110011001
111100000000010	111100000000010

← 余り : R[14:0]

除数と非除数で
EORをとります。

計算の結果、次の値がえられます。

$R[14:0] = 111100000000010$

つまり、これがCRCの値となり、データフィールド後に付加され電信情報Tx(X)として送信されます。

実際にはスタッフビット()とCRCデリミタ()が付加され、

11110000010000101がCRCフィールド上で送信されます。

3.7 備考

次に、受信した受信情報Rx(X)に誤りがあるか計算してみます。
 Rx(X)とは、前記の計算により求めた余りを $X^{15} \cdot P(X)$ に加算した値であり、この値をG(X)で除します。

$$\begin{array}{r}
 P [40:0]=1010101010100000011010101000000000000000 \\
 + R [14:0]=11110000000010 \\
 \hline
 1010101010100000011010101011110000000010
 \end{array}$$

この値をG[15:0]で除します。

	商
1100010110011001	1010101010100000011010101011110000000010
	1100010110011001
	1101111001110010
	1100010110011001
	110111101011110
	1100010110011001
	1101011000111101
	1100010110011001
	1001110100100010
	1100010110011001
	1011000101110111
	1100010110011001
	1110100111011101
	1100010110011001
	1011000100010011
	1100010110011001
	1110100100010100
	1100010110011001
	1011001000110100
	1100010110011001
	1110111101011010
	1100010110011001
	1010101100001100
	1100010110011001
	1101110100101010
	1100010110011001
	1100010110011001
	1100010110011001
	1100010110011001
	1100010110011001
	0

← 余り

以上の計算から余りはゼロとなり情報伝達は正しいとみなされます。

3 HCAN送受信(例題3.)	対象	H8S/2623F	使用機能	HCAN
-----------------	----	-----------	------	------

3.7 備考

(* 6) ACKフィールド : 正常受信確認用です。
1bitのACKスロット、1bitのACKデリミタから成ります。

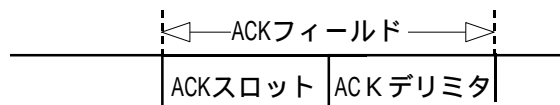


図3.11 ACKフィールド

・H8S/2623F受信側がCRCチェックでエラーを検出した場合はACKスロット=Highを、検出なかった場合はACKスロット=Lowを出力します。

(* 7) スタッフビット : データフレーム上でLowが5bit続いた場合、次の1bitは必ずHighが出力されます。同様にHighが5bit続いた場合、次の1bitは必ずLowが出力されます。これらの様にスタッフビットが出力される場合データフレームのビット長は、スタッフビットの分だけ長くなります。

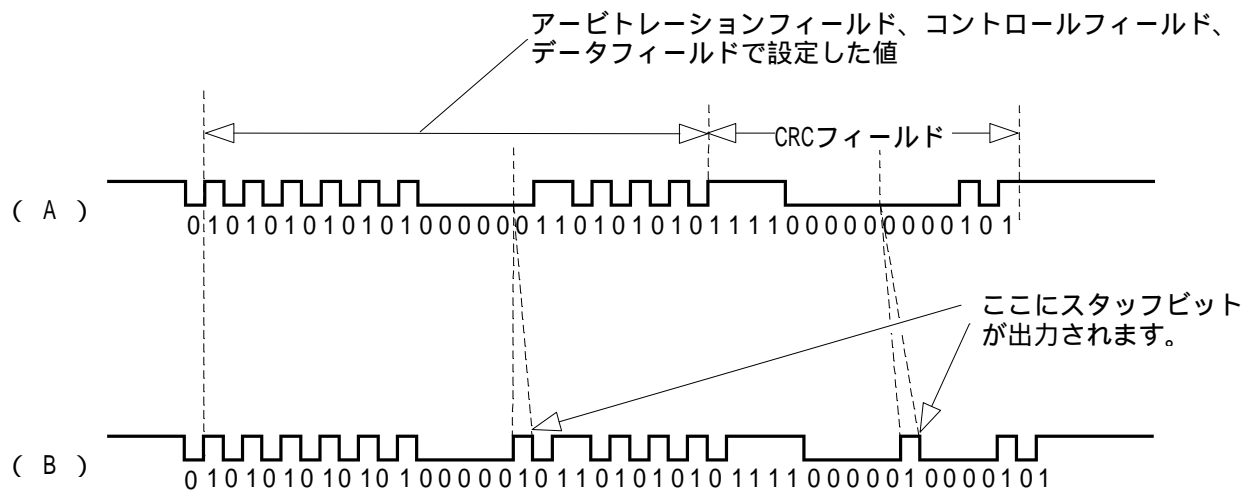
例として、設定値を

アービトレーションフィールド: " 101010101010 "

コントロールフィールド: " 000001 "

とすると、" 101010101010000001 1101010101011110000000000101 " となり、最下位から2bit目(0が5つ続いた後)にスタッフビット()が出力されます。

よって、CANバス上に送信される値は " 1010101010100000101 " となり、スタッフビットの1bit分だけ長くなります。



(A)の送信したいデータフレームに対して、CANバス上ではスタッフビットの乗ったデータフレーム(B)が送信されます。また、(B)ではデータフレームの長さもスタッフビットの分だけ長くなっています。

図3.12 スタッフビット

(* 8) コンフィギュレーションモード : HCANモジュールがリセット状態のことであり、解除はマスタコントロールレジスタ(MCR)のリセットリクエストビット(MCRO)をクリアすることで行います。

4.HCAN送受信 (例題 4.): スタンドフォーマット、8バイトデータ、優先順位付き

第4章 目次

4.1 仕様	59
4.2 使用機能説明	61
4.3 動作説明	63
4.4 ソフトウェア説明	65
4.5 フローチャート	70
4.6 プログラムリスト	74
4.7 備考	81

4 HCAN送受信(例題4. .)	対象	H8S/2623F	使用機能	HCAN, DTC
-------------------	----	-----------	------	-----------

4.1 仕様

- (1) データフレーム(*1)の送受信を行います。(H8S/2623Fを2個使用)
 図4.1にデータフレームの仕様を示します。
- (a) SOF: データフレームの開始を示します。(メールボックス 1~15)
- (b) アービトラージョンフィールド(*2)
- | | |
|--------------------------------------|---------------|
| ・メールボックス 1: " 110011001100 (H'CCC) " | 優先順位が12番目に高い。 |
| ・メールボックス 2: " 010101010100 (H'554) " | 優先順位が 5番目に高い。 |
| ・メールボックス 3: " 111011101110 (H'EEE) " | 優先順位が14番目に高い。 |
| ・メールボックス 4: " 011001100110 (H'666) " | 優先順位が 6番目に高い。 |
| ・メールボックス 5: " 000100010000 (H'110) " | 優先順位が 1番目に高い。 |
| ・メールボックス 6: " 100110011000 (H'998) " | 優先順位が 9番目に高い。 |
| ・メールボックス 7: " 001100110010 (H'332) " | 優先順位が 3番目に高い。 |
| ・メールボックス 8: " 111111111110 (H'FFE) " | 優先順位が 1番低い。 |
| ・メールボックス 9: " 001000100010 (H'222) " | 優先順位が 2番目に高い。 |
| ・メールボックス10: " 100010001000 (H'888) " | 優先順位が 8番目に高い。 |
| ・メールボックス11: " 101010101010 (H'AAA) " | 優先順位が10番目に高い。 |
| ・メールボックス12: " 110111011100 (H'DDC) " | 優先順位が13番目に高い。 |
| ・メールボックス13: " 011101110110 (H'776) " | 優先順位が 7番目に高い。 |
| ・メールボックス14: " 010001000100 (H'444) " | 優先順位が 4番目に高い。 |
| ・メールボックス15: " 101110111010 (H'BBA) " | 優先順位が11番目に高い。 |
- (c) コントロールフィールド(*3) : " 001000 " とします。(メールボックス 1~15)
- ・IDE = " 0 " : スタンダードフォーマットの選択
 - ・RO = " 0 " : リザーブビット
 - ・DLC = " 1000 " : 8バイトのデータ長の設定
- (d) データフィールド(*4)
- | |
|--------------------------------------|
| ・メールボックス 1: " H'1111111111111111 " " |
| ・メールボックス 2: " H'2222222222222222 " " |
| ・メールボックス 3: " H'3333333333333333 " " |
| ・メールボックス 4: " H'4444444444444444 " " |
| ・メールボックス 5: " H'5555555555555555 " " |
| ・メールボックス 6: " H'6666666666666666 " " |
| ・メールボックス 7: " H'7777777777777777 " " |
| ・メールボックス 8: " H'8888888888888888 " " |
| ・メールボックス 9: " H'9999999999999999 " " |
| ・メールボックス10: " H'AAAAAAAAAAAAAAAA " " |
| ・メールボックス11: " H'BBBBBBBBBBBBBBBB " " |
| ・メールボックス12: " H'CCCCCCCCCCCCCCCC " " |
| ・メールボックス13: " H'DDDDDDDDDDDDDDDD " " |
| ・メールボックス14: " H'EEEEEEEEEEEEEEEE " " |
| ・メールボックス15: " H'FFFFFFFFFFFFFFFF " " |
- (e) CRCフィールド(*5) : HCAN内部でCRCが自動生成されます。(メールボックス 1~15)
- (f) ACKフィールド(*6) : 送信側では " 11 "、受信側では " 01 (正常時) " が出力されます。
(メールボックス 1~15)
- (g) EOF: 送信/受信データフレームの完了を示します。(メールボックス 1~15)
- (2) 通信速度は1Mbps (20MHz動作時) とします。
- (3) データ長は、8バイトとします。(メールボックス 1~15)
- (4) メッセージの送信はメールボックス1~15を用います。
- (5) メッセージの受信はメールボックス0~15を用います。メールボックス0のメッセージの受信方式としては Identifierをマスクして一致すれば受信とします。
- (6) メッセージIdentifierの優先順位により送信します。
- (7) 受信メッセージは、DTCを使用し、内蔵RAMに格納します。
- (a) すべてのメッセージを受信後、DTCをソフトウェア起動します。
 - (b) ブロック転送モードを使用します。
- (8) DTC転送終了割り込み後、HCANスリープモードへ遷移します。
- (9) 図4.2にCANバスの接続例を示します。

4 HCAN送受信(例題4. .)	対象	H8S/2623F	使用機能	HCAN, DTC
-------------------	----	-----------	------	-----------

4.1 仕様

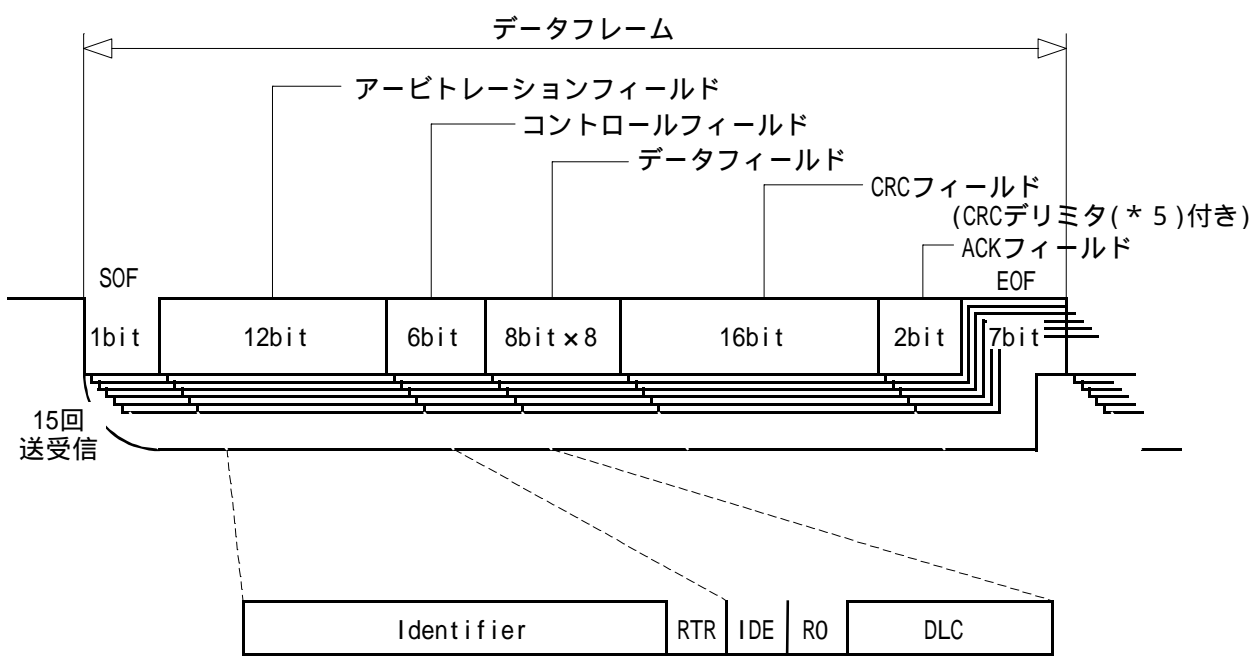
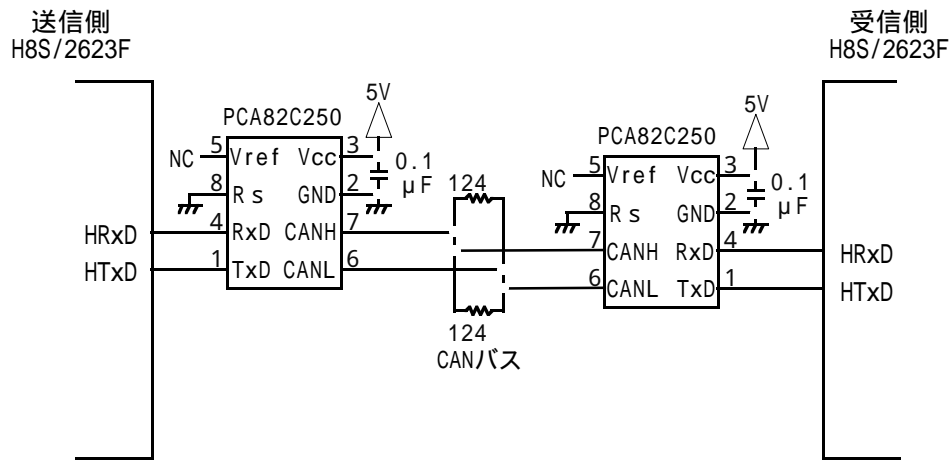


図4.1 データフレームの仕様

(*)は「4.7 備考」を参照してください。



本LSIとCANバスを接続するためにはバストランシーバICが必要になります。Philips社製PCA82C50とコンパチブルなものを推奨します。

図4.2 H8S/2623Fを用いたCANインターフェース

4 HCAN送受信(例題4. .)	対象	H8S/2623F	使用機能	HCAN, DTC
-------------------	----	-----------	------	-----------

4.2 使用機能説明

表4.1~4.4に本タスク例の機能割り付けを示します。本タスク例は表4.1~4.4に示すようにH8S/2623Fに内蔵しているHCANの機能を割り付け、HCANの送受信を行います。

表4.1 HCAN機能割り付け

HCANレジスタ		機能
端子	HTxD	メッセージの送信を行います。
	HRxD	メッセージの受信を行います。
送受信 共通 レジスタ	IRR	各割り込み要因のステータスを示します。
	BCR	CANのボーレートプリスケアラ、ビットタイミングパラメータを設定します。
	MBCR	メールボックスの送信/受信を設定します。
	MCR	CANインタフェースの制御を行います。
	MC0_1 ┆ MC15_8	アービトレーションフィールド、コントロールフィールドの設定を行います。
	MD0_1 ┆ MD15_8	データフィールドの設定を行います。
送信用 レジスタ	TXPR	送信メッセージをメールボックスに格納後送信待ち状態を設定します。
	TXACK	対応するメールボックスの送信メッセージが正常に送信されたことを示します。
受信用 レジスタ	LAFMH	受信用メールボックス0のIdentifier用フィルタマスクの設定を行います。
	RXPR	対応するメールボックスにデータが正常に受信されたことを示します。

表4.2 MSTPCR機能割り付け

MSTPCRレジスタ	機能
MSTPCRA	モジュールストップモードの制御を行います。
MSTPCRC	

表4.3 割り込みコントローラ割り付け

IPRレジスタ	機能
IPRC	割り込み優先レベルの制御を行います。
SYSCR	割り込み制御モードの設定を行います。
exr	割り込み要求マスクレベルの指定を行います。

4 HCAN送受信(例題4. .)	対象	H8S/2623F	使用機能	HCAN, DTC
-------------------	----	-----------	------	-----------

4.2 使用機能説明

表4.4 DTC機能割り付け

DTC	機能
SAR (内蔵RAM)	転送元アドレス(受信メッセージの格納元)を設定します。
DAR (内蔵RAM)	転送先アドレス(受信メッセージの格納先)を設定します。
MAR (内蔵RAM)	ブロック転送モード、バイトサイズ転送等の設定をします。
MRB (内蔵RAM)	DTCデータ転送終了後、CPUへの割り込みを設定します。
CRA (内蔵RAM)	DTCのデータ転送の転送回数を設定します。
CRB (内蔵RAM)	DTCのブロックデータ転送の転送回数を設定します。
DTVECR(レジスタ)	ソフトウェア起動によるDTC起動および、ベクタ番号を設定します。

4.3 動作説明(送信時)

図4.3に送信時の動作原理を示します。図4.3に示すようにH8S/2623Fのハードウェア処理およびソフトウェア処理によりHCANの送信を行います。

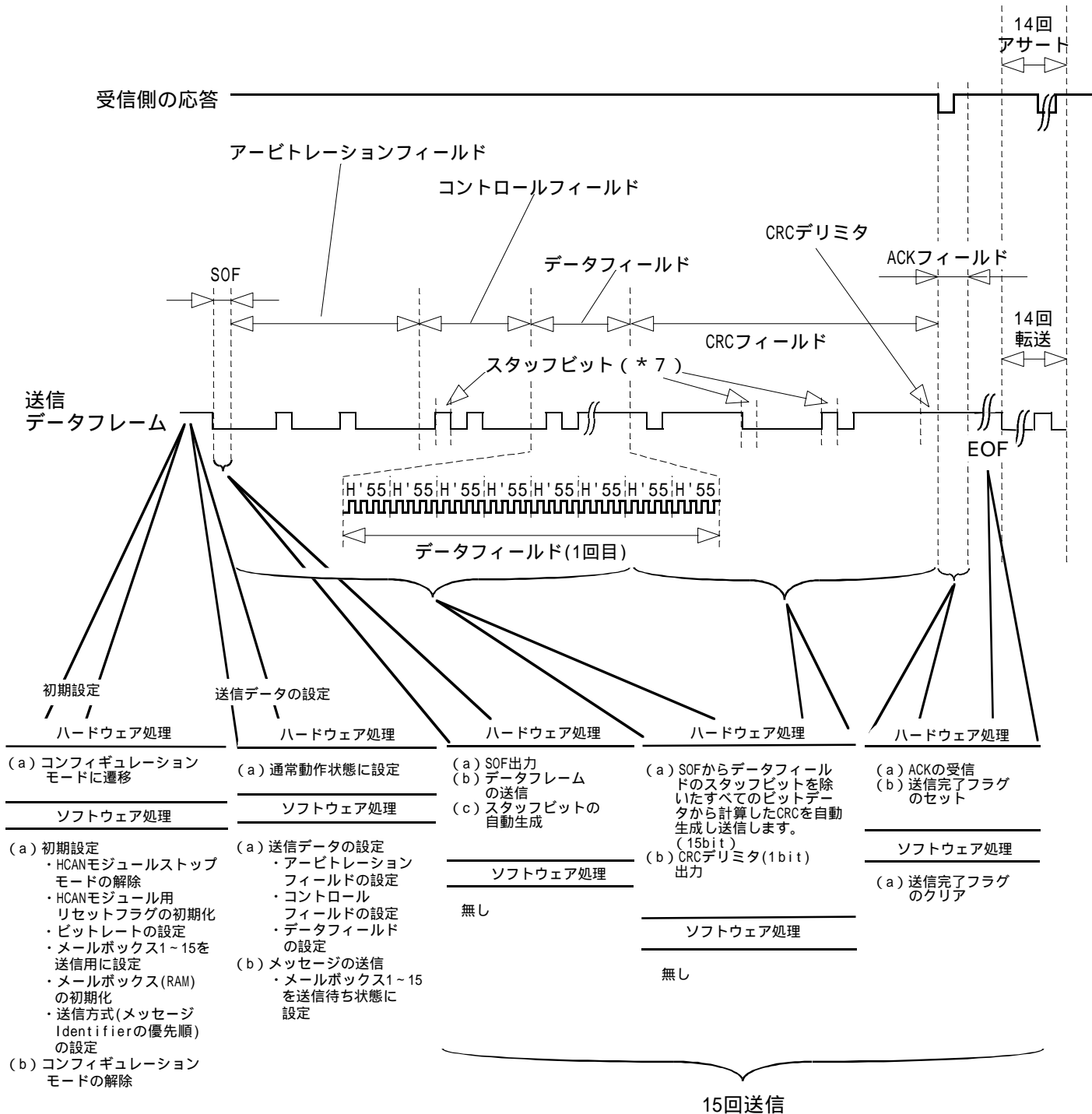


図4.3 HCAN送信時の動作

(*)は「4.7 備考」を参照してください。

4.3 動作説明(受信時)

図4.4に受信時の動作原理を示します。図4.4に示すようにH8S/2623Fのハードウェア処理およびソフトウェア処理によりHCANの受信を行います。

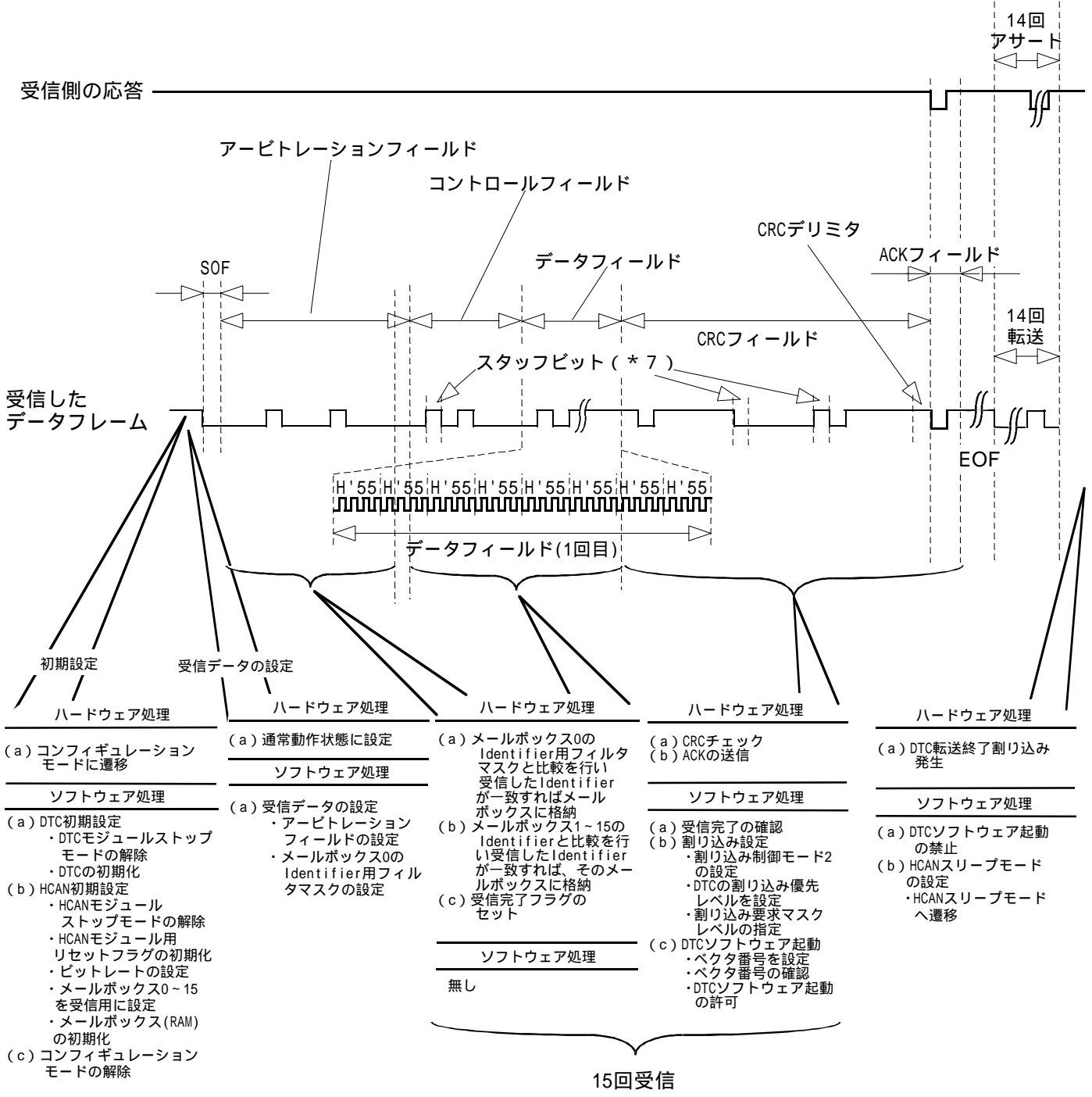


図4.4 HCAN受信時の動作

(*)は「4.7 備考」を参照してください。

4 HCAN送受信(例題4. .)	対象	H8S/2623F	使用機能	HCAN, DTC
-------------------	----	-----------	------	-----------

4.4 ソフトウェア説明

(1) モジュール説明

モジュール名	ラベル名	機能
メインルーチン	main	HCANの初期設定および送信、受信の設定を行います。
DTC転送終了 割り込みルーチン	SWDTEND	DTCソフトウェア起動を禁止し、HCANスリープモードへ遷移します。

(2) 使用変数の説明

ラベル名	機能	データ長	使用モジュール名
COUNT	HCAN_MCO_1 ~ MC15_8, HCAN_MDO_1 ~ MD15_8を初期化します。	unsigned_short	メインルーチン

4 HCAN送受信(例題4. .)	対象	H8S/2623F	使用機能	HCAN, DTC
-------------------	----	-----------	------	-----------

4.4 ソフトウェア説明

(3) 仕様内部レジスタの説明

レジスタ名	機能	設定値	使用 モジュール名	
送受信共通設定				
MSTPCRC	HCANモジュールストップモードを解除します。	0xF7	メインルーチン	
HCAN_BCR	HCANのビットレートを1Mbpsに設定します。	0x0025		
HCAN_IRR	HCANモジュール用リセットフラグを初期化します。	0x0100		
送信時の設定				
HCAN_MBCR	メールボックス1~15を送信用に設定します。	0x0100		
HCAN_MCR	メッセージIdentifierの優先順位で送信するように設定および、リセットリクエストビットをクリアします。	0x00		
HCAN_MC1~15_1	8バイトのデータ長を設定します。	すべて0x08		
HCAN_MC1_5	メールボックス1のデータフレーム、スタンダードフォーマットを選択および、Identifierを設定します。	0xC0		
HCAN_MC1_6		0xCC		
HCAN_MC2_5	メールボックス2のデータフレーム、スタンダードフォーマットを選択および、Identifierを設定します。	0x40		
HCAN_MC2_6		0x55		
HCAN_MC3_5	メールボックス3のデータフレーム、スタンダードフォーマットを選択および、Identifierを設定します。	0xE0		
HCAN_MC3_6		0xEE		
HCAN_MC4_5	メールボックス4のデータフレーム、スタンダードフォーマットを選択および、Identifierを設定します。	0x60		
HCAN_MC4_6		0x66		
HCAN_MC5_5	メールボックス5のデータフレーム、スタンダードフォーマットを選択および、Identifierを設定します。	0x00		
HCAN_MC5_6		0x11		
HCAN_MC6_5	メールボックス6のデータフレーム、スタンダードフォーマットを選択および、Identifierを設定します。	0x80		
HCAN_MC6_6		0x99		
HCAN_MC7_5	メールボックス7のデータフレーム、スタンダードフォーマットを選択および、Identifierを設定します。	0x20		
HCAN_MC7_6		0x33		
HCAN_MC8_5	メールボックス8のデータフレーム、スタンダードフォーマットを選択および、Identifierを設定します。	0xE0		
HCAN_MC8_6		0xFF		
HCAN_MC9_5	メールボックス9のデータフレーム、スタンダードフォーマットを選択および、Identifierを設定します。	0x20		
HCAN_MC9_6		0x22		
HCAN_MC10_5	メールボックス10のデータフレーム、スタンダードフォーマットを選択および、Identifierを設定します。	0x80		
HCAN_MC10_6		0x88		
HCAN_MC11_5	メールボックス11のデータフレーム、スタンダードフォーマットを選択および、Identifierを設定します。	0xA0		
HCAN_MC11_6		0xAA		
HCAN_MC12_5	メールボックス12のデータフレーム、スタンダードフォーマットを選択および、Identifierを設定します。	0xC0		
HCAN_MC12_6		0xDD		
HCAN_MC13_5	メールボックス13のデータフレーム、スタンダードフォーマットを選択および、Identifierを設定します。	0x60		
HCAN_MC13_6		0x77		
HCAN_MC14_5	メールボックス14のデータフレーム、スタンダードフォーマットを選択および、Identifierを設定します。	0x40		
HCAN_MC14_6		0x44		
HCAN_MC15_5	メールボックス15のデータフレーム、スタンダードフォーマットを選択および、Identifierを設定します。	0xA0		
HCAN_MC15_6		0xBB		

4 HCAN送受信(例題4. .)	対象	H8S/2623F	使用機能	HCAN, DTC
-------------------	----	-----------	------	-----------

4.4 ソフトウェア説明

レジスタ名	機能	設定値	使用 モジュール名
送信時の設定			
HCAN_MD1_1~8	メールボックス1の1~8バイト目の送信データを設定します。	すべて0x11	メインルーチン
HCAN_MD2_1~8	メールボックス2の1~8バイト目の送信データを設定します。	すべて0x22	
HCAN_MD3_1~8	メールボックス3の1~8バイト目の送信データを設定します。	すべて0x33	
HCAN_MD4_1~8	メールボックス4の1~8バイト目の送信データを設定します。	すべて0x44	
HCAN_MD5_1~8	メールボックス5の1~8バイト目の送信データを設定します。	すべて0x55	
HCAN_MD6_1~8	メールボックス6の1~8バイト目の送信データを設定します。	すべて0x66	
HCAN_MD7_1~8	メールボックス7の1~8バイト目の送信データを設定します。	すべて0x77	
HCAN_MD8_1~8	メールボックス8の1~8バイト目の送信データを設定します。	すべて0x88	
HCAN_MD9_1~8	メールボックス9の1~8バイト目の送信データを設定します。	すべて0x99	
HCAN_MD10_1~8	メールボックス10の1~8バイト目の送信データを設定します。	すべて0xAA	
HCAN_MD11_1~8	メールボックス11の1~8バイト目の送信データを設定します。	すべて0xBB	
HCAN_MD12_1~8	メールボックス12の1~8バイト目の送信データを設定します。	すべて0xCC	
HCAN_MD13_1~8	メールボックス13の1~8バイト目の送信データを設定します。	すべて0xDD	
HCAN_MD14_1~8	メールボックス14の1~8バイト目の送信データを設定します。	すべて0xEE	
HCAN_MD15_1~8	メールボックス15の1~8バイト目の送信データを設定します。	すべて0xFF	
HCAN_TXPR	メールボックス1~15を送信待ち状態に設定します。	0xFEFF	
HCAN_TXACK	データフレーム送信完了フラグをクリアします。	0xFEFF	
受信時の設定			
MSTPCRA	DTCのモジュールストップモードを解除します。	0x3F	メインルーチン
HCAN_MCR	リセットリクエストビットをクリアします。	0xFE	
	HCANスリープモードへ遷移します。	0x20	
HCAN_MBCR	メールボックス0~15を受信用に設定します。	0xFFFF	
HCAN_LAFMH	メールボックス0のIdentifier用フィルタマスクを設定します。	0x0000	
HCAN_RXPR	メッセージ受信完了フラグをクリアします。	0xFFFF	
SYSCR	割り込み制御モード2に設定します。	0x20	
INTC.IPRC	DTCの割り込み優先レベルを7に設定します。	0x07	
exr	割り込み要求マスクレベルを指定します。	0x00	
DTC_DTVECR	ベクタ番号を設定します。	0x60	
	DTCソフトウェア起動を許可します。	0x80	
	DTCソフトウェア起動を禁止します。	0x7F	終了割り込み

(注意)

終了割り込み : DTC転送終了割り込みルーチン

4 HCAN送受信(例題4. .)	対象	H8S/2623F	使用機能	HCAN, DTC
-------------------	----	-----------	------	-----------

4.4 ソフトウェア説明

レジスタ名	機能	設定値	使用モジュール名
受信時の設定			
HCAN_MC0_5	メールボックス0のデータフレーム、スタンダードフォーマットを選択および、Identifierを設定します。	0xA0	メインルーチン
HCAN_MC0_6		0xAA	
HCAN_MC1_5	メールボックス1のデータフレーム、スタンダードフォーマットを選択および、Identifierを設定します。	0x00	
HCAN_MC1_6		0x11	
HCAN_MC2_5	メールボックス2のデータフレーム、スタンダードフォーマットを選択および、Identifierを設定します。	0x20	
HCAN_MC2_6		0x22	
HCAN_MC3_5	メールボックス3のデータフレーム、スタンダードフォーマットを選択および、Identifierを設定します。	0x20	
HCAN_MC3_6		0x33	
HCAN_MC4_5	メールボックス4のデータフレーム、スタンダードフォーマットを選択および、Identifierを設定します。	0x40	
HCAN_MC4_6		0x44	
HCAN_MC5_5	メールボックス5のデータフレーム、スタンダードフォーマットを選択および、Identifierを設定します。	0x40	
HCAN_MC5_6		0x55	
HCAN_MC6_5	メールボックス6のデータフレーム、スタンダードフォーマットを選択および、Identifierを設定します。	0x60	
HCAN_MC6_6		0x66	
HCAN_MC7_5	メールボックス7のデータフレーム、スタンダードフォーマットを選択および、Identifierを設定します。	0x60	
HCAN_MC7_6		0x77	
HCAN_MC8_5	メールボックス8のデータフレーム、スタンダードフォーマットを選択および、Identifierを設定します。	0x80	
HCAN_MC8_6		0x88	
HCAN_MC9_5	メールボックス9のデータフレーム、スタンダードフォーマットを選択および、Identifierを設定します。	0x80	
HCAN_MC9_6		0x99	
HCAN_MC10_5	メールボックス10のデータフレーム、スタンダードフォーマットを選択および、Identifierを設定します。	0xA0	
HCAN_MC10_6		0xAA	
HCAN_MC11_5	メールボックス11のデータフレーム、スタンダードフォーマットを選択および、Identifierを設定します。	0xA0	
HCAN_MC11_6		0xBB	
HCAN_MC12_5	メールボックス12のデータフレーム、スタンダードフォーマットを選択および、Identifierを設定します。	0xC0	
HCAN_MC12_6		0xCC	
HCAN_MC13_5	メールボックス13のデータフレーム、スタンダードフォーマットを選択および、Identifierを設定します。	0xC0	
HCAN_MC13_6		0xDD	
HCAN_MC14_5	メールボックス14のデータフレーム、スタンダードフォーマットを選択および、Identifierを設定します。	0xE0	
HCAN_MC14_6		0xEE	
HCAN_MC15_5	メールボックス15のデータフレーム、スタンダードフォーマットを選択および、Identifierを設定します。	0xE0	
HCAN_MC15_6		0xFF	

4 HCAN送受信(例題4. .)	対象	H8S/2623F	使用機能	HCAN, DTC
-------------------	----	-----------	------	-----------

4.4 ソフトウェア説明

(4) 使用RAM

シンボル名	機能	アドレス	使用モジュール名
受信時の設定			
MAILBOX0.Message_DATA1~8	HCAN_MD0_1~8のデータを格納します。	0xFFC100~7	メインルーチン
MAILBOX1.Message_DATA1~8	HCAN_MD1_1~8のデータを格納します。	0xFFC108~F	
MAILBOX2.Message_DATA1~8	HCAN_MD2_1~8のデータを格納します。	0xFFC110~7	
MAILBOX3.Message_DATA1~8	HCAN_MD3_1~8のデータを格納します。	0xFFC118~F	
MAILBOX4.Message_DATA1~8	HCAN_MD4_1~8のデータを格納します。	0xFFC120~7	
MAILBOX5.Message_DATA1~8	HCAN_MD5_1~8のデータを格納します。	0xFFC128~F	
MAILBOX6.Message_DATA1~8	HCAN_MD6_1~8のデータを格納します。	0xFFC130~7	
MAILBOX7.Message_DATA1~8	HCAN_MD7_1~8のデータを格納します。	0xFFC138~F	
MAILBOX8.Message_DATA1~8	HCAN_MD8_1~8のデータを格納します。	0xFFC140~7	
MAILBOX9.Message_DATA1~8	HCAN_MD9_1~8のデータを格納します。	0xFFC148~F	
MAILBOX10.Message_DATA1~8	HCAN_MD10_1~8のデータを格納します。	0xFFC150~7	
MAILBOX11.Message_DATA1~8	HCAN_MD11_1~8のデータを格納します。	0xFFC158~F	
MAILBOX12.Message_DATA1~8	HCAN_MD12_1~8のデータを格納します。	0xFFC160~7	
MAILBOX13.Message_DATA1~8	HCAN_MD13_1~8のデータを格納します。	0xFFC168~F	
MAILBOX14.Message_DATA1~8	HCAN_MD14_1~8のデータを格納します。	0xFFC170~7	
MAILBOX15.Message_DATA1~8	HCAN_MD15_1~8のデータを格納します。	0xFFC178~F	
SAR	転送元アドレス(受信メッセージの格納元)を設定します。	0xFFEC01~3 (0xFFFF8B0)	
MRA	ブロック転送モード、バイトサイズ転送等の設定をします。	0xFFEC00 (0xA8)	
DAR	転送先アドレス(受信メッセージの格納先)を設定します。	0xFFEC05~7 (0xFFC100)	
MRB	DTCデータ転送終了後、CPUへの割り込みを許可します。	0xFFEC04 (0x00)	
CRA	DTCのデータ転送の転送ブロックサイズ保持値(上位バイト)と転送ブロックサイズカウンタ(下位バイト)を設定します。	0xFFEC08、9 (0x8080)	
CRB	DTCのブロック転送の転送回数を設定します。	0xFFEC0A、B (0x0001)	

()は設定値

4.5 送信フローチャート

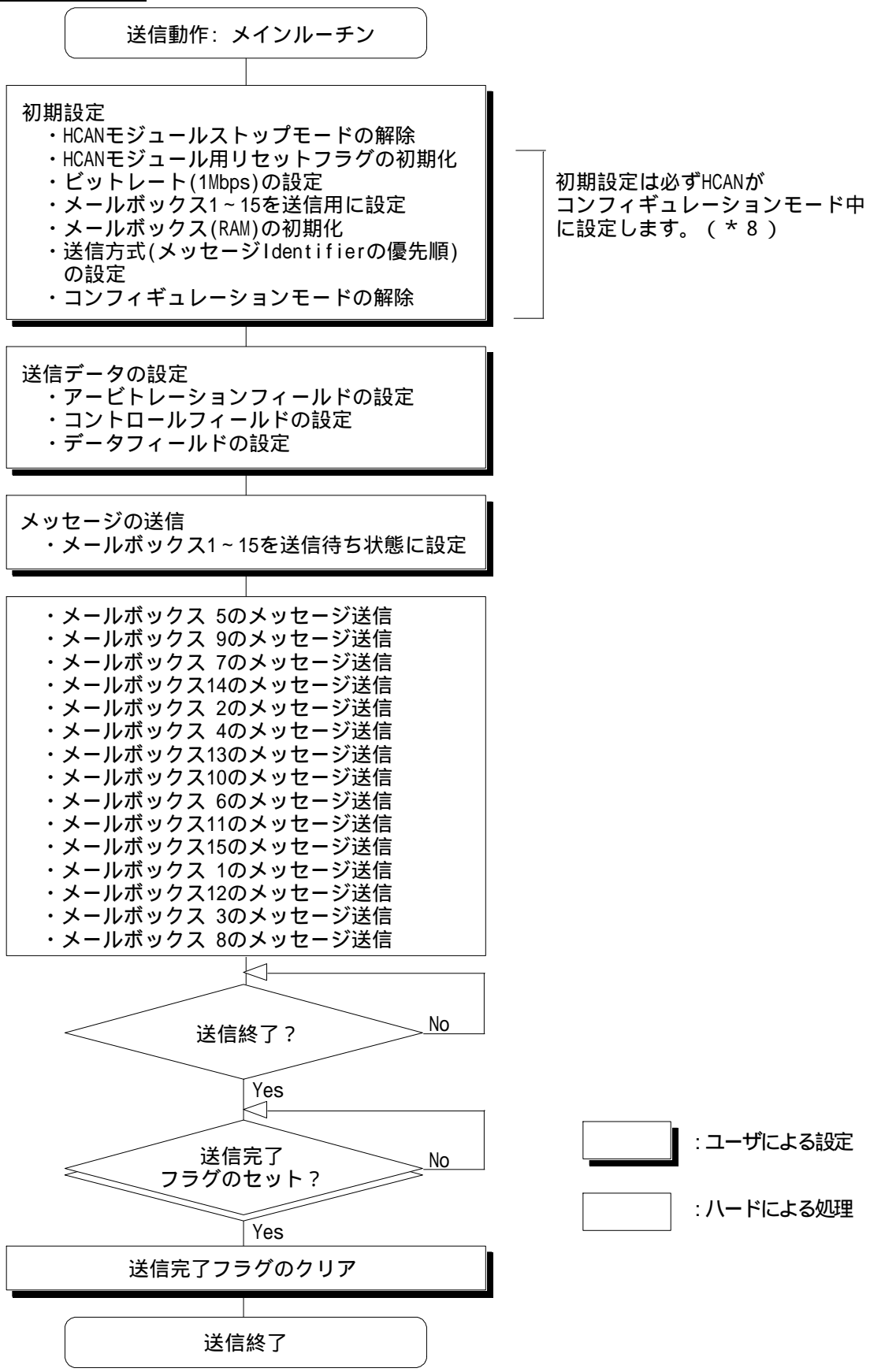


図4.5 送信時のフローチャート

4.5 受信フローチャート

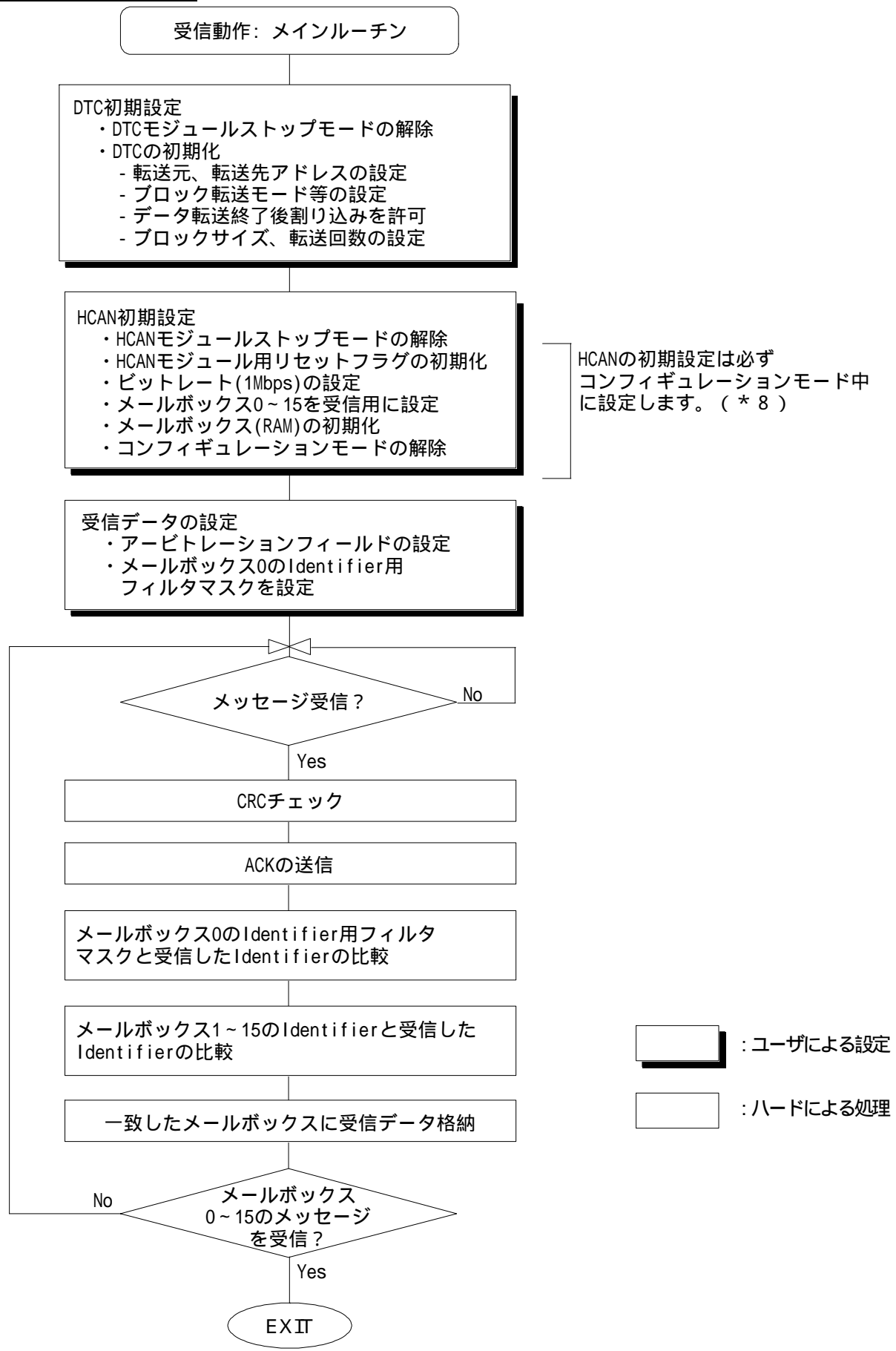


図4.6 受信時のフローチャート(1)

4.5 受信フローチャート

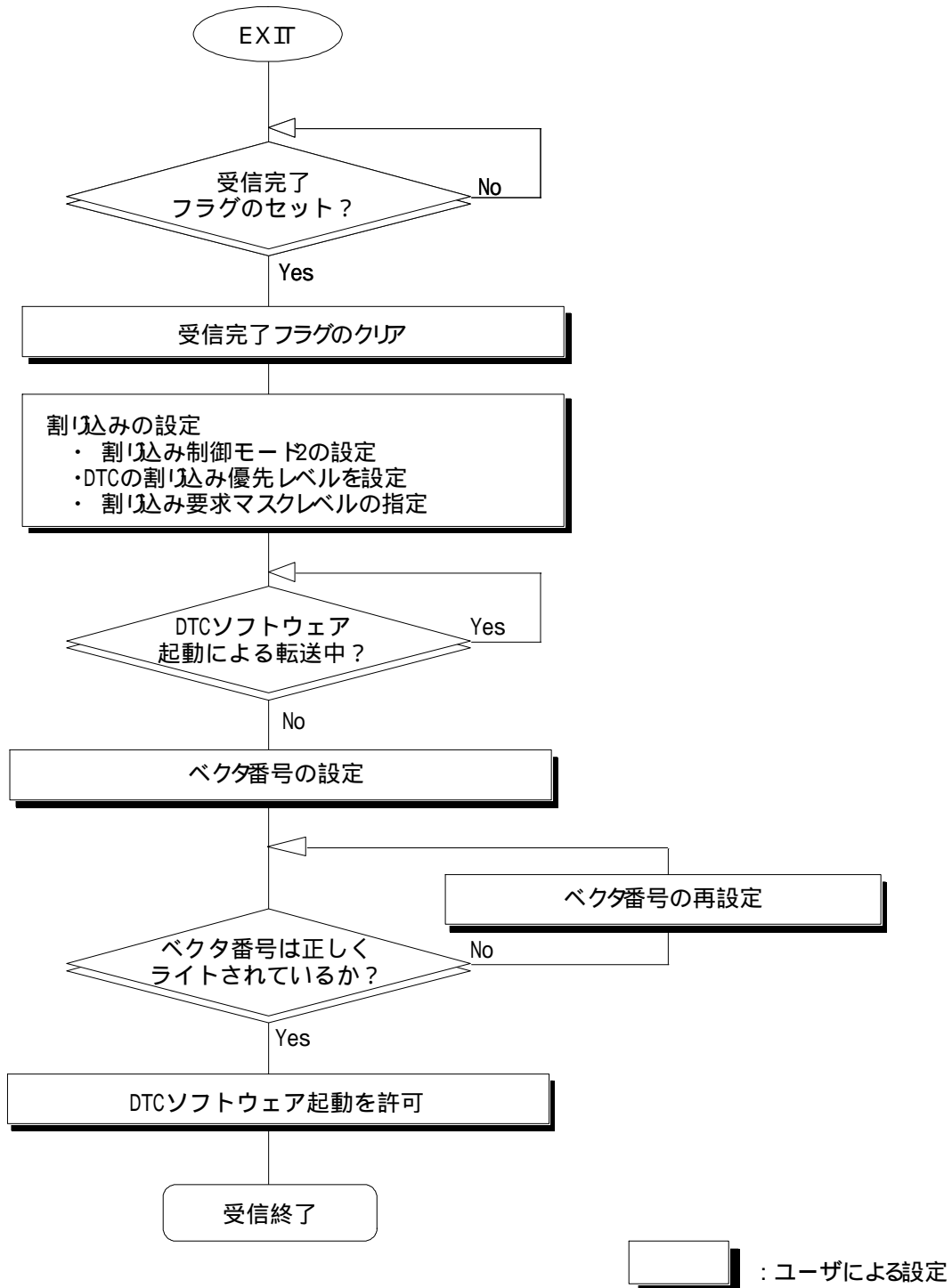


図4.7 受信時のフローチャート(2)

4.5 受信フローチャート

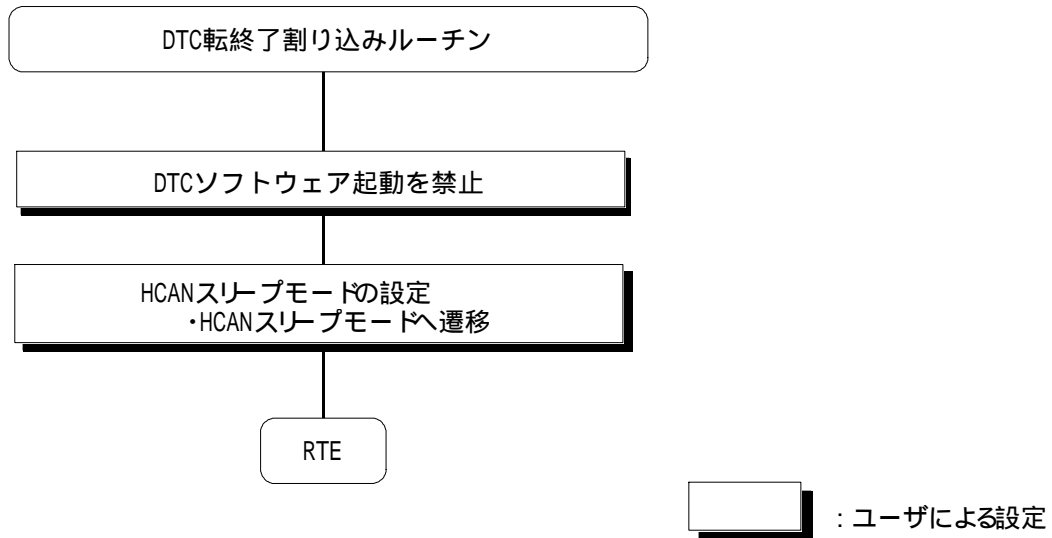


図4.8 DTC転送終了割り込みルーチンのフローチャート

4.6 送信プログラムリスト

```

/*****
/*          HCAN 送信プログラム          */
/*****
#include <stdio.h>          /* ライブ 関数用ヘッダ ファイル */
#include <machine.h>       /* ライブ 関数用ヘッダ ファイル */
#include "2623.h"         /* 周辺レジスタ定義ヘッダ ファイル */
/*****
/*          関数プロトコル宣言          */
/*****
void main( void );
/*****
/*          定数定義          */
/*****
#define COUNT (*(unsigned short *)0xFFC000)
/*****
/*          メインルーチン          */
/*****
void main(void)
{
/* 初期設定 */
MSTPCRC = 0xF7;          /* HCANレジスタストップモードの解除 */
HCAN_IRR = 0x0100;      /* HCANレジスタ用リセットフラグの初期化 */
HCAN_BCR = 0x0025;      /* ヒットレート 1Mbps */
HCAN_MBCR = 0x0100;     /* メールボックス1~15を送信用に設定 */
for( COUNT = 0; COUNT < 128; COUNT++ ) /* メールボックス(RAM)の初期設定 */
{
    *(char*)&HCAN_MCO_1 + COUNT) = 0x00;
}
for( COUNT = 0; COUNT < 128; COUNT++ ) /* メールボックス(RAM)の初期設定 */
{
    *(char*)&HCAN_MDO_1 + COUNT) = 0x00;
}
HCAN_MCR = 0x00;        /* メッセージ Identifierの優先順位で送信 */
/* 送信データの設定 */
/***** Mail Box 1 *****/
HCAN_MC1_5 = 0xC0;      /* スタンドフォーマット、データフレーム、Identifierの設定 */
HCAN_MC1_6 = 0xCC;      /* Identifierの設定 */
HCAN_MC1_1 = 0x08;      /* データ長: 8バイト */
HCAN_MD1_1 = 0x11;      /* メッセージの内容: 00010001 */
HCAN_MD1_2 = 0x11;      /* メッセージの内容: 00010001 */
HCAN_MD1_3 = 0x11;      /* メッセージの内容: 00010001 */
HCAN_MD1_4 = 0x11;      /* メッセージの内容: 00010001 */
HCAN_MD1_5 = 0x11;      /* メッセージの内容: 00010001 */
HCAN_MD1_6 = 0x11;      /* メッセージの内容: 00010001 */
HCAN_MD1_7 = 0x11;      /* メッセージの内容: 00010001 */
HCAN_MD1_8 = 0x11;      /* メッセージの内容: 00010001 */
/***** Mail Box 2 *****/
HCAN_MC2_5 = 0x40;      /* スタンドフォーマット、データフレーム、Identifierの設定 */
HCAN_MC2_6 = 0x55;      /* Identifierの設定 */
HCAN_MC2_1 = 0x08;      /* データ長: 8バイト */
HCAN_MD2_1 = 0x22;      /* メッセージの内容: 00100010 */
HCAN_MD2_2 = 0x22;      /* メッセージの内容: 00100010 */
HCAN_MD2_3 = 0x22;      /* メッセージの内容: 00100010 */
HCAN_MD2_4 = 0x22;      /* メッセージの内容: 00100010 */
HCAN_MD2_5 = 0x22;      /* メッセージの内容: 00100010 */
HCAN_MD2_6 = 0x22;      /* メッセージの内容: 00100010 */
HCAN_MD2_7 = 0x22;      /* メッセージの内容: 00100010 */
HCAN_MD2_8 = 0x22;      /* メッセージの内容: 00100010 */

```

4 HCAN送受信(例題4. .)	対象	H8S/2623F	使用機能	HCAN, DTC
-------------------	----	-----------	------	-----------

4.6 送信プログラムリスト

```

/***** Mail Box 3 *****/
HCAN_MC3_5 = 0xE0; /* スタンドフォーマット、データレム、Identifierの設定 */
HCAN_MC3_6 = 0xEE; /* Identifierの設定 */
HCAN_MC3_1 = 0x08; /* データ長：8ビット */
HCAN_MD3_1 = 0x33; /* メッセージの内容：00110011 */
HCAN_MD3_2 = 0x33; /* メッセージの内容：00110011 */
HCAN_MD3_3 = 0x33; /* メッセージの内容：00110011 */
HCAN_MD3_4 = 0x33; /* メッセージの内容：00110011 */
HCAN_MD3_5 = 0x33; /* メッセージの内容：00110011 */
HCAN_MD3_6 = 0x33; /* メッセージの内容：00110011 */
HCAN_MD3_7 = 0x33; /* メッセージの内容：00110011 */
HCAN_MD3_8 = 0x33; /* メッセージの内容：00110011 */
/***** Mail Box 4 *****/
HCAN_MC4_5 = 0x60; /* スタンドフォーマット、データレム、Identifierの設定 */
HCAN_MC4_6 = 0x66; /* Identifierの設定 */
HCAN_MC4_1 = 0x08; /* データ長：8ビット */
HCAN_MD4_1 = 0x44; /* メッセージの内容：01000100 */
HCAN_MD4_2 = 0x44; /* メッセージの内容：01000100 */
HCAN_MD4_3 = 0x44; /* メッセージの内容：01000100 */
HCAN_MD4_4 = 0x44; /* メッセージの内容：01000100 */
HCAN_MD4_5 = 0x44; /* メッセージの内容：01000100 */
HCAN_MD4_6 = 0x44; /* メッセージの内容：01000100 */
HCAN_MD4_7 = 0x44; /* メッセージの内容：01000100 */
HCAN_MD4_8 = 0x44; /* メッセージの内容：01000100 */
/***** Mail Box 5 *****/
HCAN_MC5_5 = 0x00; /* スタンドフォーマット、データレム、Identifierの設定 */
HCAN_MC5_6 = 0x11; /* Identifierの設定 */
HCAN_MC5_1 = 0x08; /* データ長：8ビット */
HCAN_MD5_1 = 0x55; /* メッセージの内容：01010101 */
HCAN_MD5_2 = 0x55; /* メッセージの内容：01010101 */
HCAN_MD5_3 = 0x55; /* メッセージの内容：01010101 */
HCAN_MD5_4 = 0x55; /* メッセージの内容：01010101 */
HCAN_MD5_5 = 0x55; /* メッセージの内容：01010101 */
HCAN_MD5_6 = 0x55; /* メッセージの内容：01010101 */
HCAN_MD5_7 = 0x55; /* メッセージの内容：01010101 */
HCAN_MD5_8 = 0x55; /* メッセージの内容：01010101 */
/***** Mail Box 6 *****/
HCAN_MC6_5 = 0x80; /* スタンドフォーマット、データレム、Identifierの設定 */
HCAN_MC6_6 = 0x99; /* Identifierの設定 */
HCAN_MC6_1 = 0x08; /* データ長：8ビット */
HCAN_MD6_1 = 0x66; /* メッセージの内容：01100110 */
HCAN_MD6_2 = 0x66; /* メッセージの内容：01100110 */
HCAN_MD6_3 = 0x66; /* メッセージの内容：01100110 */
HCAN_MD6_4 = 0x66; /* メッセージの内容：01100110 */
HCAN_MD6_5 = 0x66; /* メッセージの内容：01100110 */
HCAN_MD6_6 = 0x66; /* メッセージの内容：01100110 */
HCAN_MD6_7 = 0x66; /* メッセージの内容：01100110 */
HCAN_MD6_8 = 0x66; /* メッセージの内容：01100110 */
/***** Mail Box 7 *****/
HCAN_MC7_5 = 0x20; /* スタンドフォーマット、データレム、Identifierの設定 */
HCAN_MC7_6 = 0x33; /* Identifierの設定 */
HCAN_MC7_1 = 0x08; /* データ長：8ビット */
HCAN_MD7_1 = 0x77; /* メッセージの内容：01110111 */
HCAN_MD7_2 = 0x77; /* メッセージの内容：01110111 */
HCAN_MD7_3 = 0x77; /* メッセージの内容：01110111 */
HCAN_MD7_4 = 0x77; /* メッセージの内容：01110111 */
HCAN_MD7_5 = 0x77; /* メッセージの内容：01110111 */
HCAN_MD7_6 = 0x77; /* メッセージの内容：01110111 */
HCAN_MD7_7 = 0x77; /* メッセージの内容：01110111 */
HCAN_MD7_8 = 0x77; /* メッセージの内容：01110111 */

```

4 HCAN送受信(例題4. .)	対象	H8S/2623F	使用機能	HCAN, DTC
-------------------	----	-----------	------	-----------

4.6 送信プログラムリスト

```

/***** Mail Box 8 *****/
HCAN_MC8_5 = 0xE0; /* スタンドフォーマット、データフレーム、Identifierの設定 */
HCAN_MC8_6 = 0xFF; /* Identifierの設定 */
HCAN_MC8_1 = 0x08; /* データ長：8バイト */
HCAN_MD8_1 = 0x88; /* メッセージの内容：10001000 */
HCAN_MD8_2 = 0x88; /* メッセージの内容：10001000 */
HCAN_MD8_3 = 0x88; /* メッセージの内容：10001000 */
HCAN_MD8_4 = 0x88; /* メッセージの内容：10001000 */
HCAN_MD8_5 = 0x88; /* メッセージの内容：10001000 */
HCAN_MD8_6 = 0x88; /* メッセージの内容：10001000 */
HCAN_MD8_7 = 0x88; /* メッセージの内容：10001000 */
HCAN_MD8_8 = 0x88; /* メッセージの内容：10001000 */

/***** Mail Box 9 *****/
HCAN_MC9_5 = 0x20; /* スタンドフォーマット、データフレーム、Identifierの設定 */
HCAN_MC9_6 = 0x22; /* Identifierの設定 */
HCAN_MC9_1 = 0x08; /* データ長：8バイト */
HCAN_MD9_1 = 0x99; /* メッセージの内容：10011001 */
HCAN_MD9_2 = 0x99; /* メッセージの内容：10011001 */
HCAN_MD9_3 = 0x99; /* メッセージの内容：10011001 */
HCAN_MD9_4 = 0x99; /* メッセージの内容：10011001 */
HCAN_MD9_5 = 0x99; /* メッセージの内容：10011001 */
HCAN_MD9_6 = 0x99; /* メッセージの内容：10011001 */
HCAN_MD9_7 = 0x99; /* メッセージの内容：10011001 */
HCAN_MD9_8 = 0x99; /* メッセージの内容：10011001 */

/***** Mail Box 10 *****/
HCAN_MC10_5 = 0x80; /* スタンドフォーマット、データフレーム、Identifierの設定 */
HCAN_MC10_6 = 0x88; /* Identifierの設定 */
HCAN_MC10_1 = 0x08; /* データ長：8バイト */
HCAN_MD10_1 = 0xAA; /* メッセージの内容：10101010 */
HCAN_MD10_2 = 0xAA; /* メッセージの内容：10101010 */
HCAN_MD10_3 = 0xAA; /* メッセージの内容：10101010 */
HCAN_MD10_4 = 0xAA; /* メッセージの内容：10101010 */
HCAN_MD10_5 = 0xAA; /* メッセージの内容：10101010 */
HCAN_MD10_6 = 0xAA; /* メッセージの内容：10101010 */
HCAN_MD10_7 = 0xAA; /* メッセージの内容：10101010 */
HCAN_MD10_8 = 0xAA; /* メッセージの内容：10101010 */

/***** Mail Box 11 *****/
HCAN_MC11_5 = 0xA0; /* スタンドフォーマット、データフレーム、Identifierの設定 */
HCAN_MC11_6 = 0xAA; /* Identifierの設定 */
HCAN_MC11_1 = 0x08; /* データ長：8バイト */
HCAN_MD11_1 = 0xBB; /* メッセージの内容：10111011 */
HCAN_MD11_2 = 0xBB; /* メッセージの内容：10111011 */
HCAN_MD11_3 = 0xBB; /* メッセージの内容：10111011 */
HCAN_MD11_4 = 0xBB; /* メッセージの内容：10111011 */
HCAN_MD11_5 = 0xBB; /* メッセージの内容：10111011 */
HCAN_MD11_6 = 0xBB; /* メッセージの内容：10111011 */
HCAN_MD11_7 = 0xBB; /* メッセージの内容：10111011 */
HCAN_MD11_8 = 0xBB; /* メッセージの内容：10111011 */

/***** Mail Box 12 *****/
HCAN_MC12_5 = 0xC0; /* スタンドフォーマット、データフレーム、Identifierの設定 */
HCAN_MC12_6 = 0xDD; /* Identifierの設定 */
HCAN_MC12_1 = 0x08; /* データ長：8バイト */
HCAN_MD12_1 = 0xCC; /* メッセージの内容：11001100 */
HCAN_MD12_2 = 0xCC; /* メッセージの内容：11001100 */
HCAN_MD12_3 = 0xCC; /* メッセージの内容：11001100 */
HCAN_MD12_4 = 0xCC; /* メッセージの内容：11001100 */
HCAN_MD12_5 = 0xCC; /* メッセージの内容：11001100 */
HCAN_MD12_6 = 0xCC; /* メッセージの内容：11001100 */
HCAN_MD12_7 = 0xCC; /* メッセージの内容：11001100 */
HCAN_MD12_8 = 0xCC; /* メッセージの内容：11001100 */

```


4 HCAN送受信(例題4. .)	対象	H8S/2623F	使用機能	HCAN, DTC
-------------------	----	-----------	------	-----------

4.6 送信プログラムリスト

```

/***** Mail Box 13 *****/
    HCAN_MC13_5 = 0x60;      /* スタンドフォーマット、データフレーム、Identifierの設定 */
    HCAN_MC13_6 = 0x77;      /* Identifierの設定 */
    HCAN_MC13_1 = 0x08;      /* データ長：8バイト */
    HCAN_MD13_1 = 0xDD;      /* メッセージの内容：11011101 */
    HCAN_MD13_2 = 0xDD;      /* メッセージの内容：11011101 */
    HCAN_MD13_3 = 0xDD;      /* メッセージの内容：11011101 */
    HCAN_MD13_4 = 0xDD;      /* メッセージの内容：11011101 */
    HCAN_MD13_5 = 0xDD;      /* メッセージの内容：11011101 */
    HCAN_MD13_6 = 0xDD;      /* メッセージの内容：11011101 */
    HCAN_MD13_7 = 0xDD;      /* メッセージの内容：11011101 */
    HCAN_MD13_8 = 0xDD;      /* メッセージの内容：11011101 */
/***** Mail Box 14 *****/
    HCAN_MC14_5 = 0x40;      /* スタンドフォーマット、データフレーム、Identifierの設定 */
    HCAN_MC14_6 = 0x44;      /* Identifierの設定 */
    HCAN_MC14_1 = 0x08;      /* データ長：8バイト */
    HCAN_MD14_1 = 0xEE;      /* メッセージの内容：11101110 */
    HCAN_MD14_2 = 0xEE;      /* メッセージの内容：11101110 */
    HCAN_MD14_3 = 0xEE;      /* メッセージの内容：11101110 */
    HCAN_MD14_4 = 0xEE;      /* メッセージの内容：11101110 */
    HCAN_MD14_5 = 0xEE;      /* メッセージの内容：11101110 */
    HCAN_MD14_6 = 0xEE;      /* メッセージの内容：11101110 */
    HCAN_MD14_7 = 0xEE;      /* メッセージの内容：11101110 */
    HCAN_MD14_8 = 0xEE;      /* メッセージの内容：11101110 */
/***** Mail Box 15 *****/
    HCAN_MC15_5 = 0xA0;      /* スタンドフォーマット、データフレーム、Identifierの設定 */
    HCAN_MC15_6 = 0xBB;      /* Identifierの設定 */
    HCAN_MC15_1 = 0x08;      /* データ長：8バイト */
    HCAN_MD15_1 = 0xFF;      /* メッセージの内容：11111111 */
    HCAN_MD15_2 = 0xFF;      /* メッセージの内容：11111111 */
    HCAN_MD15_3 = 0xFF;      /* メッセージの内容：11111111 */
    HCAN_MD15_4 = 0xFF;      /* メッセージの内容：11111111 */
    HCAN_MD15_5 = 0xFF;      /* メッセージの内容：11111111 */
    HCAN_MD15_6 = 0xFF;      /* メッセージの内容：11111111 */
    HCAN_MD15_7 = 0xFF;      /* メッセージの内容：11111111 */
    HCAN_MD15_8 = 0xFF;      /* メッセージの内容：11111111 */
/* メッセージの送信 */
    HCAN_TXPR = 0xFEFF;      /* メールボックス1~15を送信待ち状態に設定 */
    while((HCAN_TXACK & 0xFEFF) != 0xFEFF); /* 送信完了待ち */
/* 送信完了フラグのクリア */
    HCAN_TXACK &= 0xFEFF;    /* 送信完了フラグのクリア */
    while(1);
}

```

4.6 受信プログラムリスト

```

/*****
/*
/*          HCAN 受信プログラム          */
/*
/*****
#include <stdio.h>          /* ライフ 関数用ヘッダ ファイル */
#include <machine.h>       /* ライフ 関数用ヘッダ ファイル */
#include "2623.h"         /* 周辺レジスタ定義ヘッダ ファイル */
/*****
/*          関数プロトコル宣言          */
/*
/*****
void main( void );
/*****
/*          定数定義          */
/*
/*****
#define COUNT      (*(unsigned short *)0xFFC000)
volatile struct MB /* struct MAILBOX0 ~ 15 */
{
    unsigned char Message_DATA1; /* 受信データ1格納 */
    unsigned char Message_DATA2; /* 受信データ2格納 */
    unsigned char Message_DATA3; /* 受信データ3格納 */
    unsigned char Message_DATA4; /* 受信データ4格納 */
    unsigned char Message_DATA5; /* 受信データ5格納 */
    unsigned char Message_DATA6; /* 受信データ6格納 */
    unsigned char Message_DATA7; /* 受信データ7格納 */
    unsigned char Message_DATA8; /* 受信データ8格納 */
};
#define MAILBOX0  (*(volatile struct MB *)0xFFC100) /* メールボックス0 アドレス */
#define MAILBOX1  (*(volatile struct MB *)0xFFC108) /* メールボックス1 アドレス */
#define MAILBOX2  (*(volatile struct MB *)0xFFC110) /* メールボックス2 アドレス */
#define MAILBOX3  (*(volatile struct MB *)0xFFC118) /* メールボックス3 アドレス */
#define MAILBOX4  (*(volatile struct MB *)0xFFC120) /* メールボックス4 アドレス */
#define MAILBOX5  (*(volatile struct MB *)0xFFC128) /* メールボックス5 アドレス */
#define MAILBOX6  (*(volatile struct MB *)0xFFC130) /* メールボックス6 アドレス */
#define MAILBOX7  (*(volatile struct MB *)0xFFC138) /* メールボックス7 アドレス */
#define MAILBOX8  (*(volatile struct MB *)0xFFC140) /* メールボックス8 アドレス */
#define MAILBOX9  (*(volatile struct MB *)0xFFC148) /* メールボックス9 アドレス */
#define MAILBOX10 (*(volatile struct MB *)0xFFC150) /* メールボックス10 アドレス */
#define MAILBOX11 (*(volatile struct MB *)0xFFC158) /* メールボックス11 アドレス */
#define MAILBOX12 (*(volatile struct MB *)0xFFC160) /* メールボックス12 アドレス */
#define MAILBOX13 (*(volatile struct MB *)0xFFC168) /* メールボックス13 アドレス */
#define MAILBOX14 (*(volatile struct MB *)0xFFC170) /* メールボックス14 アドレス */
#define MAILBOX15 (*(volatile struct MB *)0xFFC178) /* メールボックス15 アドレス */
#define SAR      (*(volatile unsigned long *)0xFFEC00) /* レジスタ情報設定 */
#define MRA      (*(volatile unsigned char *)0xFFEC00) /* レジスタ情報設定 */
#define DAR      (*(volatile unsigned long *)0xFFEC04) /* レジスタ情報設定 */
#define MRB      (*(volatile unsigned char *)0xFFEC04) /* レジスタ情報設定 */
#define CRA      (*(volatile unsigned short *)0xFFEC08) /* レジスタ情報設定 */
#define CRB      (*(volatile unsigned short *)0xFFEC0A) /* レジスタ情報設定 */

```

4 HCAN送受信(例題4. .)	対象	H8S/2623F	使用機能	HCAN, DTC
-------------------	----	-----------	------	-----------

4.6 受信プログラムリスト

```

/*****
/*
/*                               メインルーチン                               */
/*****
void main(void)
{
/* DTC初期設定 */
MSTPCRA = 0x3F; /* DTCエンジンストップモードの解除 */
SAR = (long)(&HCAN_MDO_1); /* 転送元アドレスの設定 */
MRA = 0xA8; /* SAR, DARは転送後インクリメント、ブロック転送モードの設定 */
DAR = (long)(&MAILBOX0.Message_DATA1); /* 転送先アドレス(内蔵RAM)の設定 */
MRB = 0x00; /* DTCによるデータ転送終了後、割り込みを許可 */
CRA = 0x8080; /* ブロック転送は128バイト単位に設定 */
CRB = 0x0001; /* ブロック転送回数: 1回の設定 */

/* HCAN初期設定 */
MSTPCRC = 0xF7; /* HCANエンジンストップモードの解除 */
HCAN_IRR = 0x0100; /* HCANエンジン用リセットフラグの初期化 */
HCAN_BCR = 0x0025; /* ビットレート 1Mbps */
HCAN_MBCR = 0xFFFF; /* メールボックス0~15を受信に設定 */
for( COUNT = 0; COUNT < 128; COUNT++ ) /* メールボックス(RAM)の初期設定 */
{
(char*)&HCAN_MCO_1 + COUNT) = 0x00;
}
for( COUNT = 0; COUNT < 128; COUNT++ ) /*メールボックス(RAM)の初期設定 */
{
(char*)&HCAN_MDO_1 + COUNT) = 0x00;
}
HCAN_MCR &= 0xFE; /* コンフィグレーションモードの解除 */

/* 受信データの設定 */
/* Mail Box 0 */
HCAN_MC0_5 = 0xA0; /* スタンドフォーマット、データフレーム、Identifierの設定 */
HCAN_MC0_6 = 0xAA; /* Identifierの設定 */
/* Mail Box 1 */
HCAN_MC1_5 = 0x00; /* スタンドフォーマット、データフレーム、Identifierの設定 */
HCAN_MC1_6 = 0x11; /* Identifierの設定 */
/* Mail Box 2 */
HCAN_MC2_5 = 0x20; /* スタンドフォーマット、データフレーム、Identifierの設定 */
HCAN_MC2_6 = 0x22; /* Identifierの設定 */
/* Mail Box 3 */
HCAN_MC3_5 = 0x20; /* スタンドフォーマット、データフレーム、Identifierの設定 */
HCAN_MC3_6 = 0x33; /* Identifierの設定 */
/* Mail Box 4 */
HCAN_MC4_5 = 0x40; /* スタンドフォーマット、データフレーム、Identifierの設定 */
HCAN_MC4_6 = 0x44; /* Identifierの設定 */
/* Mail Box 5 */
HCAN_MC5_5 = 0x40; /* スタンドフォーマット、データフレーム、Identifierの設定 */
HCAN_MC5_6 = 0x55; /* Identifierの設定 */
/* Mail Box 6 */
HCAN_MC6_5 = 0x60; /* スタンドフォーマット、データフレーム、Identifierの設定 */
HCAN_MC6_6 = 0x66; /* Identifierの設定 */
/* Mail Box 7 */
HCAN_MC7_5 = 0x60; /* スタンドフォーマット、データフレーム、Identifierの設定 */
HCAN_MC7_6 = 0x77; /* Identifierの設定 */
/* Mail Box 8 */
HCAN_MC8_5 = 0x80; /* スタンドフォーマット、データフレーム、Identifierの設定 */
HCAN_MC8_6 = 0x88; /* Identifierの設定 */

```

4 HCAN送受信(例題4. .)	対象	H8S/2623F	使用機能	HCAN, DTC
-------------------	----	-----------	------	-----------

4.6 受信プログラムリスト

```

/* Mail Box 9 */
  HCAN_MC9_5 = 0x80; /* スタンドフォーマット、データフレーム、Identifierの設定 */
  HCAN_MC9_6 = 0x99; /* Identifierの設定 */
/* Mail Box 10 */
  HCAN_MC10_5 = 0xA0; /* スタンドフォーマット、データフレーム、Identifierの設定 */
  HCAN_MC10_6 = 0xAA; /* Identifierの設定 */
/* Mail Box 11 */
  HCAN_MC11_5 = 0xA0; /* スタンドフォーマット、データフレーム、Identifierの設定 */
  HCAN_MC11_6 = 0xBB; /* Identifierの設定 */
/* Mail Box 12 */
  HCAN_MC12_5 = 0xC0; /* スタンドフォーマット、データフレーム、Identifierの設定 */
  HCAN_MC12_6 = 0xCC; /* Identifierの設定 */
/* Mail Box 13 */
  HCAN_MC13_5 = 0xC0; /* スタンドフォーマット、データフレーム、Identifierの設定 */
  HCAN_MC13_6 = 0xDD; /* Identifierの設定 */
/* Mail Box 14 */
  HCAN_MC14_5 = 0xE0; /* スタンドフォーマット、データフレーム、Identifierの設定 */
  HCAN_MC14_6 = 0xEE; /* Identifierの設定 */
/* Mail Box 15 */
  HCAN_MC15_5 = 0xE0; /* スタンドフォーマット、データフレーム、Identifierの設定 */
  HCAN_MC15_6 = 0xFF; /* Identifierの設定 */

  HCAN_LAFMH = 0x0000; /* メールボックス0はbitの一致でデータを格納 */

/* 受信完了待ち */
  while((HCAN_RXPR & 0xFFFF) != 0xFFFF); /* 受信完了待ち */
  HCAN_RXPR &= 0xFFFF; /* 受信完了フラグのクリア */
/* 割り込み設定 */
  SYSCR |= 0x20; /* 割り込み制御モード2の設定 */
  INTC_IPRC = 0x07; /* DTCの割り込みの優先レベルを7に設定 */
  set_imask_exr(0); /* 割り込み要求マスクレベルの指定 */
/* DTCソフトウェア起動 */
  while(DTC_DTVECR & 0x80); /* DTCソフトウェア起動による転送中か確認 */
  DTC_DTVECR |= 0x60; /* ハッシュ番号(H'4C0)を設定 */
  while((DTC_DTVECR & 0x7F) != 0x60){ /* DTVECR にライトしたハッシュ番号の確認 */
    DTC_DTVECR |= 0x60; /* ハッシュ番号(H'4C0)を再設定 */
  }
  DTC_DTVECR |= 0x80; /* DTCソフトウェア起動の許可 */
  while(1);
}
/*****
/*
          DTC転送終了割り込みルーチン
*****/
#pragma interrupt(SWDTEND)
void SWDTEND(void)
{
  DTC_DTVECR &= 0x7F; /* DTCソフトウェア起動を禁止 */
/* HCANスリープモードの設定 */
  HCAN_MCR |= 0x20; /* HCANスリープモードへ遷移 */
}

```

4 HCAN送受信(例題 4 .)	対象	H8S/2623F	使用機能	HCAN, DTC
-------------------	----	-----------	------	-----------

4.7 備考

- (* 1) データフレーム : 送信元から送信先へ転送したいデータです。
- (* 2) アービトレーションフィールド : メッセージ固有のIDおよびデータフレームまたはリモートフレームの設定をします。
- (* 3) コントロールフィールド : 転送するデータ長およびスタンダードフォーマットまたはエクステンデッドフォーマットの設定をします。
- (* 4) データフィールド : メッセージの内容(送信したいデータ)を設定します。
- (* 5) CRCフィールド : HCAN内部でSOFからデータフィールドのスタッフビットを除いたすべてのビットデータからCRCが自動生成され、送信メッセージのエラーを検出します。15bitのCRCと1bitのCRCデリミタから成ります。CRCデリミタはCRC後に必ず " 1 " が出力されます。

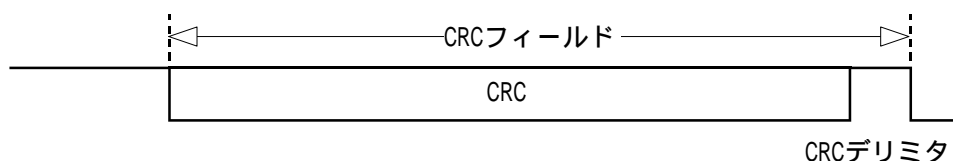


図4.9 CRCフィールド

CRCについて

伝送しようとするデータ多項式 $P(X)$ に X^R を乗じ、さらにこの $X^R \cdot P(X)$ を生成多項式 $G(X)$ で除し、余りである $R(X)$ を求めます。情報を送り出す側では、 $X^R \cdot P(X)$ に求められた $R(X)$ をチェックビットとして付加し電信情報 $Tx(X)$ として送りだします。

また、情報を受ける側では受けた受信情報 $Rx(X)$ を生成多項式 $G(X)$ で除し、余りを求めます。この余りがゼロとなった場合は、情報伝送は正しく行われたとみなし、余りが出た場合は伝送された情報に誤りがあったと判断します。

$$P(X) = \{ \text{スタッフビットを除いたSOFからデータフィールドまで} \}$$

$$G(X) = X^{15} + X^{14} + X^{10} + X^8 + X^7 + X^4 + X^3 + 1$$

$$R = 15$$

$G(X)$ はCRCを生成する多項式としてCANプロトコルにて規定されています。

例として以下の設定において送信されたデータフレームについて説明します。

設定値

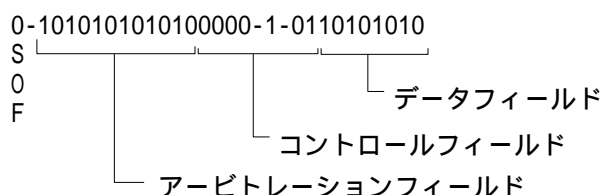
SOF : 0

アービトレーションフィールド : 101010101010

コントロールフィールド : 000001

データフィールド : 10101010

送信されるデータのSOFからデータフィールドは (スタッフビット)



で、対象となるデータはスタッフビットを除いた

010101010101000000110101010

すなわち、 $P(X) = X^{25} + X^{23} + X^{21} + X^{19} + X^{17} + X^{15} + X^8 + X^7 + X^5 + X^3 + X^1$ となります。

4.7 備考

ここで、 $P(X)$ に X^{15} を掛け、
 $X^{15} \cdot P(X) = X^{40} + X^{38} + X^{36} + X^{34} + X^{32} + X^{30} + X^{23} + X^{22} + X^{20} + X^{18} + X^{16}$
 とし、この値を
 $G(X) = X^{15} + X^{14} + X^{10} + X^8 + X^7 + X^4 + X^3 + 1$
 で除します。

$$\left(\begin{array}{l} X^{15} \cdot P(X) = X^{40} + X^{38} + X^{36} + X^{34} + X^{32} + X^{30} + X^{23} + X^{22} + X^{20} + X^{18} + X^{16} \\ P[40:0] = 1010101010100000011010101000000000000000 \\ G(X) = X^{15} + X^{14} + X^{10} + X^8 + X^7 + X^4 + X^3 + 1 \\ G[15:0] = 1100010110011001 \end{array} \right)$$

商

1100010110011001	1010101010100000011010101000000000000000	-----
	1100010110011001	-----
	1101111001110010	-----
	1100010110011001	-----
	1101111101011110	-----
	1100010110011001	-----
	1101011000111101	-----
	1100010110011001	-----
	1001110100100010	-----
	1100010110011001	-----
	1011000101110110	-----
	1100010110011001	-----
	1110100111011110	-----
	1100010110011001	-----
	1011000100011100	-----
	1100010110011001	-----
	1110100100001010	-----
	1100010110011001	-----
	1011001001001100	-----
	1100010110011001	-----
	1110111110101010	-----
	1100010110011001	-----
	1010100011001100	-----
	1100010110011001	-----
	1101101010101010	-----
	1100010110011001	-----
	1111100110011000	-----
	1100010110011001	-----
	111100000000010	-----

← 余り : R[14:0]

除数と非除数で
EORをとります。

計算の結果、次の値がえられます。

$$R[14:0] = 111100000000010$$

つまり、これがCRCの値となり、データフィールド後に付加され電信情報Tx(X)として送信されます。
 実際にはスタッフビット()とCRCデリミタ()が付加され、
 11110000010000101がCRCフィールド上で送信されます。

4.7 備考

次に、受信した受信情報Rx(X)に誤りがあるか計算してみます。
 Rx(X)とは、前記の計算により求めた余りを $X^{15} \cdot P(X)$ に加算した値であり、この値をG(X)で除します。

$$\begin{array}{r}
 P[40:0]=1010101010100000011010101000000000000000 \\
 + R[14:0]=11110000000010 \\
 \hline
 1010101010100000011010101011110000000010
 \end{array}$$

この値をG[15:0]で除します。

	商	
1100010110011001	1010101010100000011010101011110000000010	
	1100010110011001	
	1101111001110010	
	1100010110011001	
	110111101011110	
	1100010110011001	
	1101011000111101	
	1100010110011001	
	1001110100100010	
	1100010110011001	
	1011000101110111	
	1100010110011001	
	1110100111011101	
	1100010110011001	
	1011000100010011	
	1100010110011001	
	1110100100010100	
	1100010110011001	
	1011001000110100	
	1100010110011001	
	1110111101011010	
	1100010110011001	
	1010101100001100	
	1100010110011001	
	1101110100101010	
	1100010110011001	
	1100010110011001	
	1100010110011001	
	1100010110011001	
	0	← 余り

除数と非除数で
EORをとります。

以上の計算から余りはゼロとなり情報伝達は正しいとみなされます。

4 HCAN送受信(例題4. .)	対象	H8S/2623F	使用機能	HCAN, DTC
-------------------	----	-----------	------	-----------

4.7 備考

(* 6) ACKフィールド : 正常受信確認用です。
1bitのACKスロット、1bitのACKデリミタから成ります。

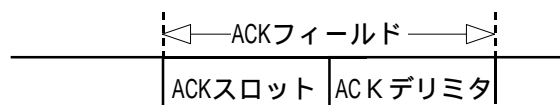


図4.10 ACKフィールド

・ H8S/2623F受信側がCRCチェックでエラーを検出した場合はACKスロット=Highを、検出しなかった場合はACKスロット=Lowを出力します。

(* 7) スタッフビット : データフレーム上でLowが5bit続いた場合、次の1bitは必ずHighが出力されます。同様にHighが5bit続いた場合、次の1bitは必ずLowが出力されます。これらの様にスタッフビットが出力される場合データフレームのビット長は、スタッフビットの分だけ長くなります。

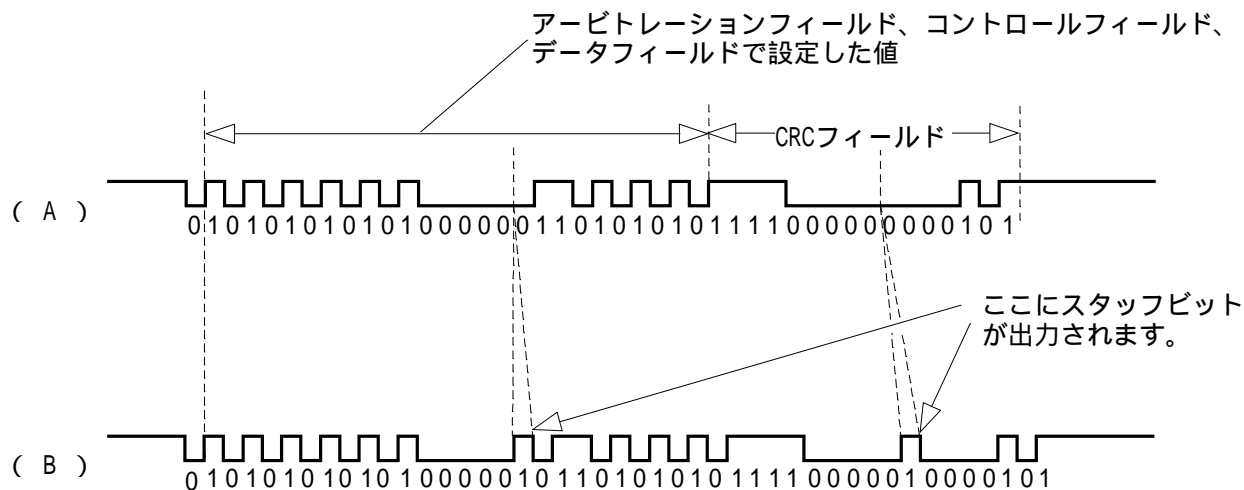
例として、設定値を

アービトレーションフィールド : " 101010101010 "

コントロールフィールド : " 000001 "

とすると、" 10101010101000000111010101011110000000000101 " となり、最下位から2bit目 (0が5つ続いた後) にスタッフビット () が出力されます。

よって、CANバス上に送信される値は " 10101010101000000101 " となり、スタッフビットの1bit分だけ長くなります。



(A)の送信したいデータフレームに対して、CANバス上ではスタッフビットの乗ったデータフレーム (B)が送信されます。また、(B)ではデータフレームの長さもスタッフビットの分だけ長くなっています。

図4.11 スタッフビット

(* 8) コンフィギュレーションモード : HCANモジュールがリセット状態のことであり、解除はマスタコントロールレジスタ (MCR) のリセットリクエストビット (MCR0) をクリアすることで行います。

5.HCAN送受信 (例題 5.):リモートフレーム

第5章 目次

5.1 仕様	87
5.2 使用機能説明	89
5.3 動作説明	90
5.4 ソフトウェア説明	92
5.5 フローチャート	93
5.6 プログラムリスト	95
5.7 備考	97

5.1 仕様

- (1) リモートフレーム(*1)の送受信を行います。(H8S/2623Fを2個使用)
 図5.1にリモートフレームの仕様を示します。
 (a) SOF: リモートフレームの開始を示します。
 (b) アービトレーションフィールド(*2): "101010101011" とします。
 • RTR = "1": リモートフレームの選択
 (c) コントロールフィールド(*3): "000001" とします。
 • IDE = "0": スタンダードフォーマットの選択
 • RO = "0": リザーブビット
 • DLC = "0001": データ長の設定(リモートフレームのため、送信側からデータの転送(データフィールド(*4)はなし。)は行われません。)
 (d) CRCフィールド(*5): HCAN内部でCRCが自動生成されます。
 (e) ACKフィールド(*6): 送信側では"11"、受信側では"01(正常時)"が出力されます。
 (f) EOF: 送信/受信データフレームの完了を示します。
 (2) 通信速度は250Kbps(20MHz動作時)とします。
 (3) メッセージの送信はメールボックス1を用います。
 (4) メッセージの受信はメールボックス0を用います。メッセージの受信方式としてはIdentifierをマスクして一致すれば受信とします。
 (5) 図5.2にCANバスの接続例を示します。

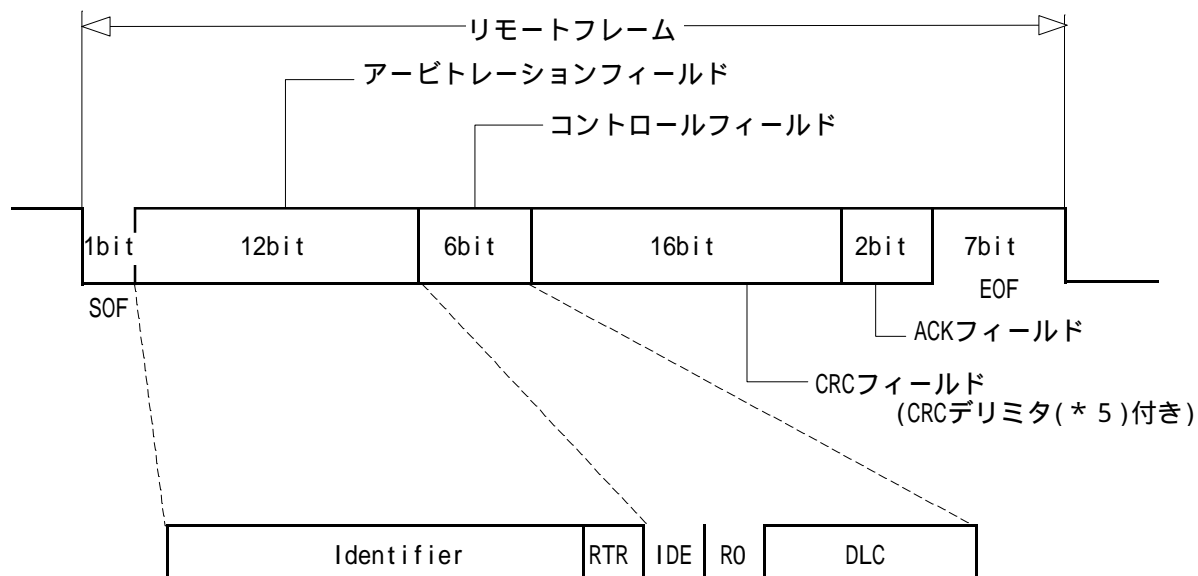
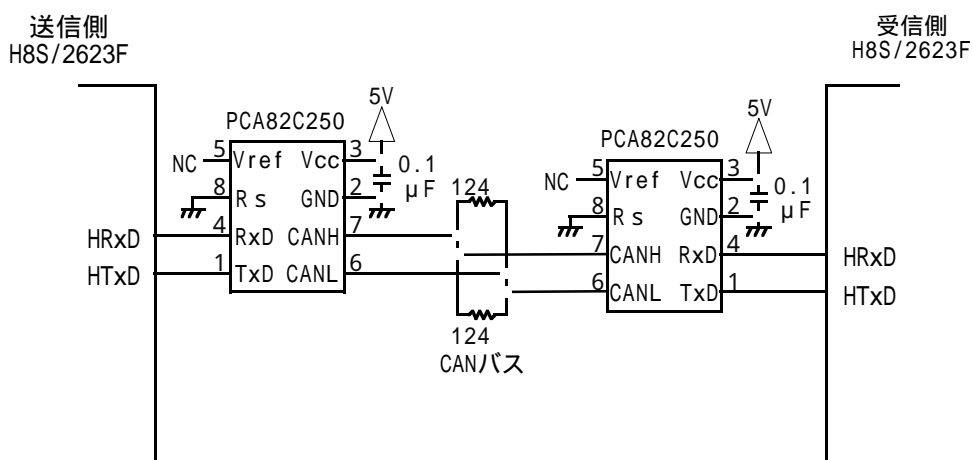


図5.1 リモートフレームの仕様

(*)は「5.7 備考」を参照してください。

5 HCAN送受信(例題5 .)	対象	H8S/2623F	使用機能	HCAN
------------------	----	-----------	------	------

5.1 仕様



本LSIとCANバスを接続するためにはバストランシーバICが必要になります。Philips社製PCA82C50とコンパチブルなものを推奨します。

図5.2 H8S/2623Fを用いたCANインタフェース

5 HCAN送受信(例題5. .)	対象	H8S/2623F	使用機能	HCAN
-------------------	----	-----------	------	------

5.2 使用機能説明

表5.1、表5.2に本タスク例の機能割り付けを示します。本タスク例は表5.1、表5.2に示すようにH8S/2623Fに内蔵しているHCANの機能を割り付け、HCANの送受信を行います。

表5.1 HCAN機能割り付け

HCANレジスタ		機能
端子	HTxD	メッセージの送信を行います。
	HRxD	メッセージの受信を行います。
送受信 共通 レジスタ	IRR	各割り込み要因のステータスを示します。
	BCR	CANのボーレートプリスケアラ、ビットタイミングパラメータを設定します。
	MBCR	メールボックスの送信/受信を設定します。
	MCR	CANインタフェースの制御を行います。
	MC0_1 ┆ MC15_8	アービトレーションフィールド、コントロールフィールドの設定を行います。
	MD0_1 ┆ MD15_8	データフィールドの設定を行います。
送信用 レジスタ	TXPR	送信メッセージをメールボックスに格納後送信待ち状態を設定します。
	TXACK	対応するメールボックスの送信メッセージが正常に送信されたことを示します。
受信用 レジスタ	LAFMH	受信用メールボックス0のIdentifier用フィルタマスクの設定を行います。
	RXPR	対応するメールボックスにデータが正常に受信されたことを示します。
	RFPR	対応するメールボックスにリモートフレームが正常に受信されたことを示します。

表5.2 MSTPCR機能割り付け

MSTPCRレジスタ	機能
MSTPCRC	モジュールストップモードの制御を行います。

5.3 動作説明(送信時)

図5.3に送信時の動作原理を示します。図5.3に示すようにH8S/2623Fのハードウェア処理およびソフトウェア処理によりHCANの送信を行います。

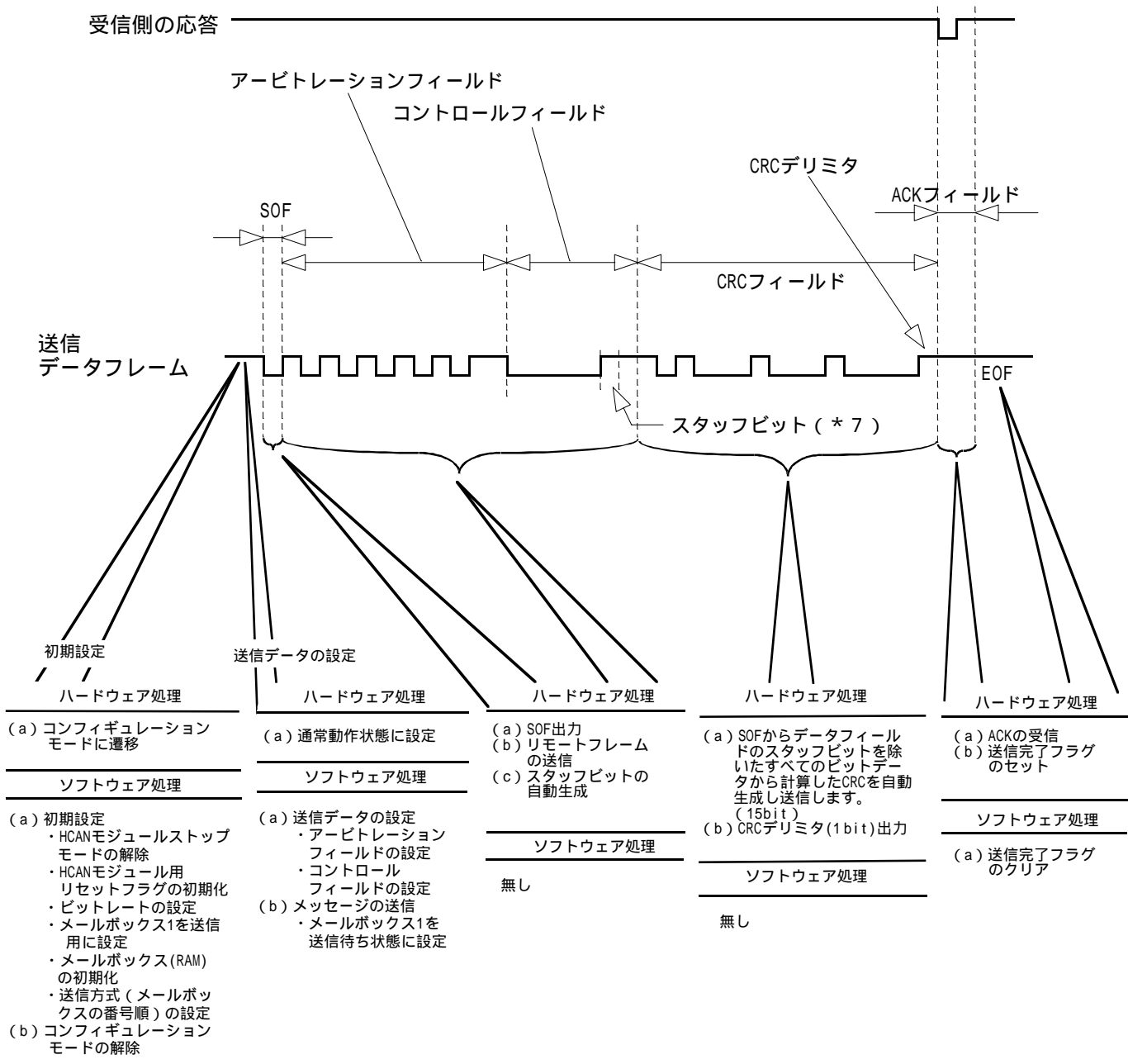


図5.3 HCAN送信時の動作

(*)は「5.7 備考」を参照してください。

5.3 動作説明(受信時)

図5.4に受信時の動作原理を示します。図5.4に示すようにH8S/2623Fのハードウェア処理およびソフトウェア処理によりHCANの受信を行います。

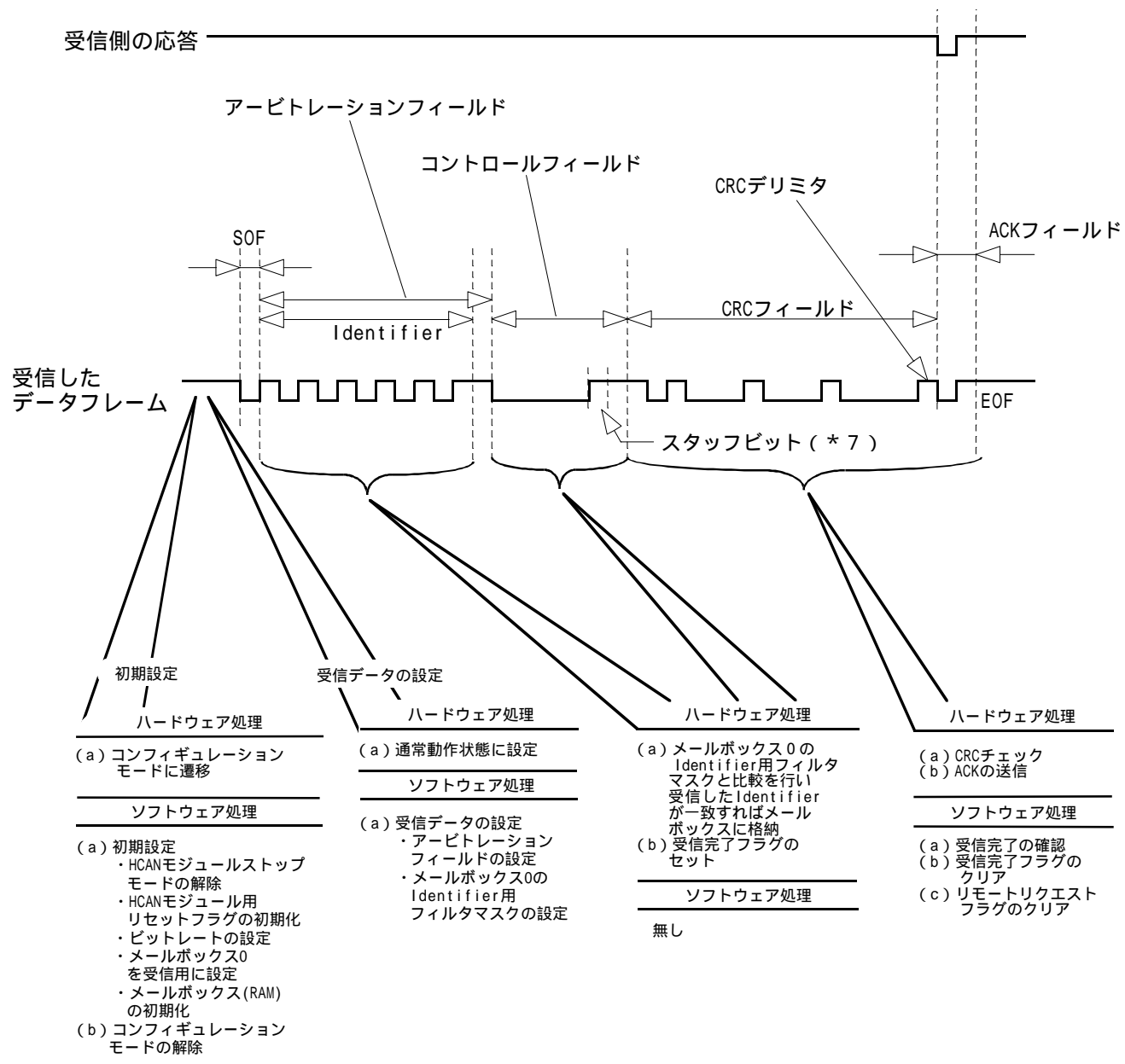


図5.4 HCAN受信時の動作

(*)は「5.7 備考」を参照してください。

5 HCAN送受信(例題5. .)	対象	H8S/2623F	使用機能	HCAN
-------------------	----	-----------	------	------

5.4 ソフトウェア説明

(1) モジュール説明

モジュール名	ラベル名	機能
メインルーチン	main	HCANの初期設定および送信、受信の設定を行います。

(2) 使用変数の説明

ラベル名	機能	データ長	使用モジュール名
COUNT	HCAN_MCO_1 ~ MC15_8, HCAN_MDO_1 ~ MD15_8を初期化します。	unsigned short	メインルーチン

(3) 仕様内部レジスタの説明

レジスタ名	機能	設定値	使用モジュール名	
送受信共通設定				
MSTPCRRC	HCANモジュールストップモードを解除します。	0xF7	メインルーチン	
HCAN_IRR	HCANモジュール用リセットフラグを初期化します。	0x0100		
HCAN_BCR	HCANのビットレートを250Kbpsに設定します。	0x0334		
送信時の設定				
HCAN_MBCR	メールボックス1を送信用に設定します。	0xFDFF		
HCAN_MCR	メールボックスの番号順で送信するように設定および、リセットリクエストビットをクリアします。	0x04		
HCAN_MC1_1	1バイトのデータ長を設定します。	0x01		
HCAN_MC1_5	メールボックス1のリモートフレーム、スタンダードフォーマットを選択および、Identifierを設定します。	0xB0		
HCAN_MC1_6	メールボックス1のIdentifierを設定します。	0xAA		
HCAN_TXPR	メールボックス1を送信待ち状態に設定します。	0x0200		
HCAN_TXACK	リモートフレーム送信完了フラグをクリアします。	0x0200		
受信時の設定				
HCAN_MBCR	メールボックス0を受信用に設定します。	0x0100		
HCAN_MCR	リセットリクエストビットをクリアします。	0xFE		
HCAN_MCO_5	メールボックス0のリモートフレーム、スタンダードフォーマットを選択および、Identifierを設定します。	0xB0		
HCAN_MCO_6	メールボックス0のIdentifierを設定します。	0xAA		
HCAN_LAFMH	メールボックス0のIdentifier用フィルタマスクを設定します。	0x0000		
HCAN_RXPR	メッセージ受信完了フラグをクリアします。	0x0100		
HCAN_RFPR	リモートフレーム受信完了フラグをクリアします。	0x0100		

5.5 送信フローチャート

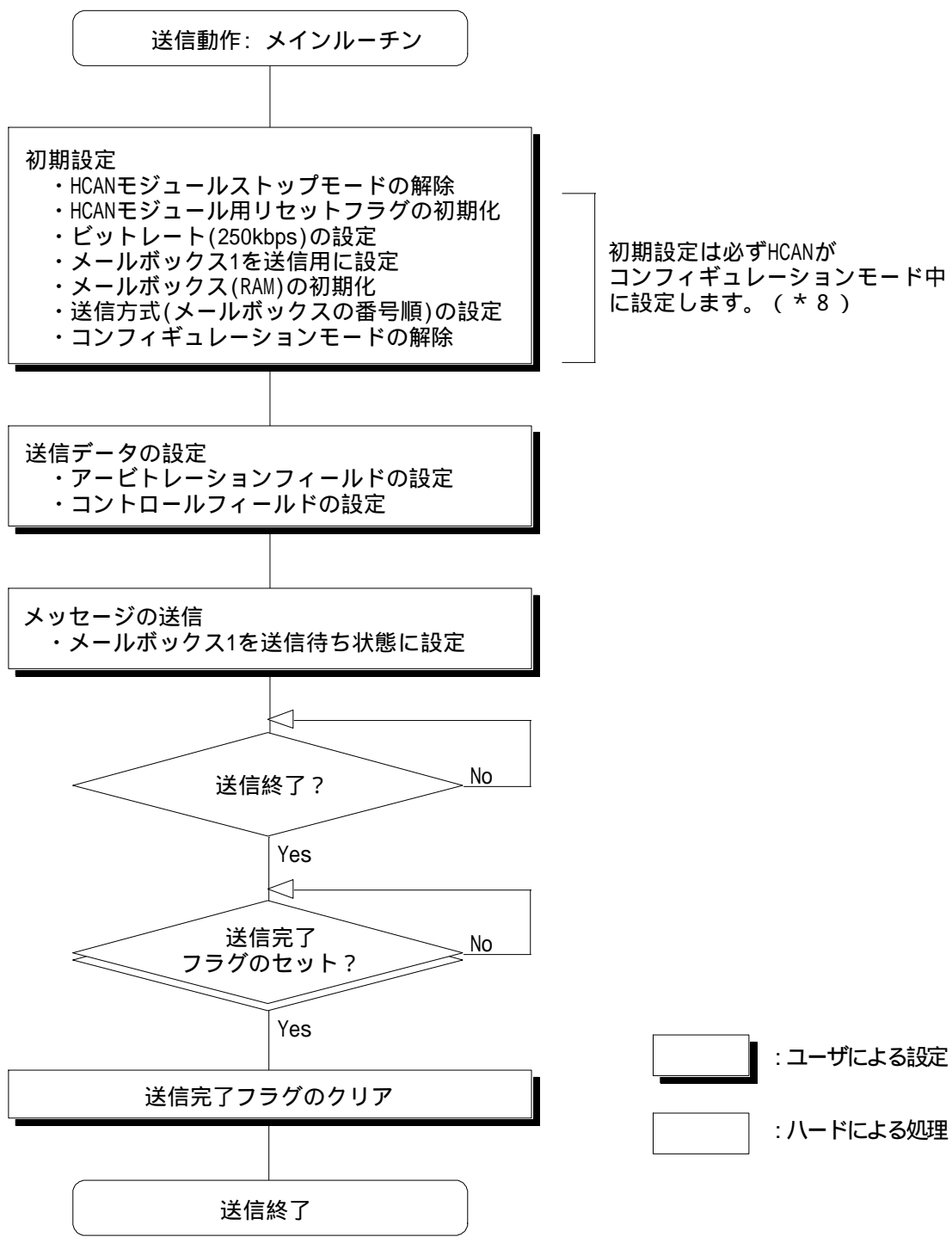


図5.5 送信時のフローチャート

(*) は「5.7 備考」を参照してください。

5.5 受信フローチャート

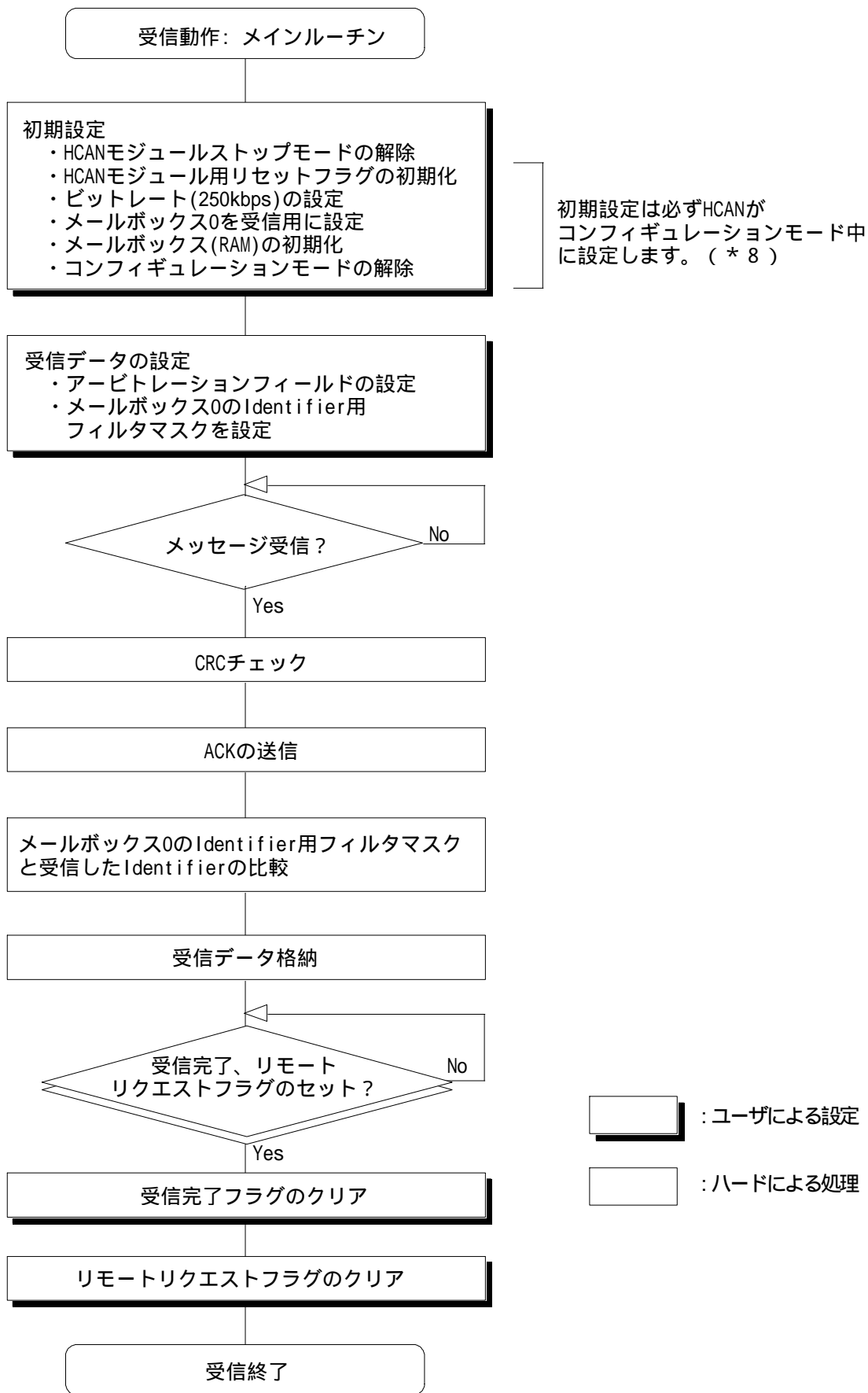


図5.6 受信時のフローチャート

5 HCAN送受信(例題5 .)	対象	H8S/2623F	使用機能	HCAN
------------------	----	-----------	------	------

5.6 送信プログラムリスト

```

/*****
/*          HCAN 送信プログラム          */
/*****
#include <stdio.h>                /* ライブ 関数用ヘッダ ファイル */
#include <machine.h>              /* ライブ 関数用ヘッダ ファイル */
#include "2623.h"                 /* 周辺レジスタ定義ヘッダ ファイル */
/*****
/*          関数プロトコル宣言          */
/*****
void main( void );
/*****
/*          定数定義          */
/*****
#define COUNT (*(unsigned short *)0xFFC000)
/*****
/*          メインルーチン          */
/*****
void main(void)
{
/* 初期設定 */
MSTPCRC = 0xF7;                  /* HCANモジュールストップモードの解除 */
HCAN_IRR = 0x0100;              /* HCANモジュール用リセットフラグの初期化 */
HCAN_BCR = 0x0334;              /* ヒットレート 250kbps */
HCAN_MBCR = 0xFDFF;            /* メールボックス1を送信用に設定 */
for( COUNT = 0; COUNT < 128; COUNT++ ) /* メールボックス(RAM)の初期設定 */
{
*(char*)&HCAN_MCO_1 + COUNT) = 0x00;
}
for( COUNT = 0; COUNT < 128; COUNT++ ) /* メールボックス(RAM)の初期設定 */
{
*(char*)&HCAN_MDO_1 + COUNT) = 0x00;
}
HCAN_MCR = 0x04; /* メールボックスの番号順で送信、コンフィグレーションモードの解除 */
/* 送信データの設定 */
HCAN_MC1_5 = 0xB0; /* スタンダードフォーマット、リモートフレーム、Identifierの設定 */
HCAN_MC1_6 = 0xAA; /* Identifierの設定 */
HCAN_MC1_1 = 0x01; /* データ長:1バイト */
/* メッセージの送信 */
HCAN_TXPR = 0x0200; /* メールボックス1を送信待ち状態に設定 */
while((HCAN_TXACK & 0x0200) != 0x0200); /* 送信完了待ち */
/* 送信完了フラグのクリア */
HCAN_TXACK &= 0x0200; /* 送信完了フラグのクリア */
while(1);
}

```

5 HCAN送受信(例題5. .)	対象	H8S/2623F	使用機能	HCAN
-------------------	----	-----------	------	------

5.6 受信プログラムリスト

```

/*****
/*          HCAN 受信プログラム          */
/*****
#include <stdio.h>                /* ライブ 関数用ヘッダ ファイル  */
#include <machine.h>             /* ライブ 関数用ヘッダ ファイル  */
#include "2623.h"                /* 周辺レジスタ定義ヘッダ ファイル */
/*****
/*          関数プロトコル宣言          */
/*****
void main( void );
/*****
/*          定数定義          */
/*****
#define COUNT (*(unsigned short *)0xFFC000)
/*****
/*          メインルーチン          */
/*****
void main(void)
{
/* 初期設定 */
MSTPCRC = 0xF7;                /* HCANモジュールストップモードの解除 */
HCAN_IRR = 0x0100;            /* HCANモジュール用リセットフラグの初期化 */
HCAN_BCR = 0x0334;            /* ビットレート 250Kbps */
HCAN_MBCR = 0x0100;           /* メールボックス0を受信に設定 */
for( COUNT = 0; COUNT < 128; COUNT++ ) /* メールボックス(RAM)の初期設定 */
{
*(char*)&HCAN_MCO_1 + COUNT) = 0x00;
}
for( COUNT = 0; COUNT < 128; COUNT++ ) /* メールボックス(RAM)の初期設定 */
{
*(char*)&HCAN_MDO_1 + COUNT) = 0x00;
}
HCAN_MCR &= 0xFE;            /* コンフィグレーションモードの解除 */
/* 受信データの設定 */
HCAN_MCO_5 = 0xB0;           /* スタンドフォーマット、リモートフレーム、Identifierの設定 */
HCAN_MCO_6 = 0xAA;           /* Identifierの設定 */
HCAN_LAFMH = 0x0000;         /* メールボックス0はbitの一致でデータを格納 */
/* 受信待ち */
while(((HCAN_RXPR & 0x0100) != 0x0100) /* リモートフレーム受信完了待ち */
&&((HCAN_RFPR & 0x0100) != 0x0100));
HCAN_RXPR &= 0x0100;         /* 受信完了フラグのクリア */
HCAN_RFPR &= 0x0100;         /* リモートクエストフラグのクリア */
while(1);
}

```

5 HCAN送受信(例題5. .)	対象	H8S/2623F	使用機能	HCAN
-------------------	----	-----------	------	------

5.7 備考

- (* 1) リモートフレーム : 送信元ヘデータフレームを要求します。
- (* 2) アービトレーションフィールド : メッセージ固有のIDおよびデータフレームまたはリモートフレームの設定をします。
- (* 3) コントロールフィールド : 転送するデータ長およびスタンダードフォーマットまたはエクステンデッドフォーマットの設定をします。
- (* 4) データフィールド : リモートフレームにデータフィールドは存在しません。ただし、送信側のコントロールフィールドのデータ長コード(DLC)には、データフレームで返信されるべきデータ長が格納されなければなりません。
- (* 5) CRCフィールド : HCAN内部でSOFからデータフィールドのスタッフビットを除いたすべてのビットデータからCRCが自動生成され、送信メッセージのエラーを検出します。15bitのCRCと1bitのCRCデリミタから成ります。CRCデリミタはCRC後に必ず " 1 " が出力されます。

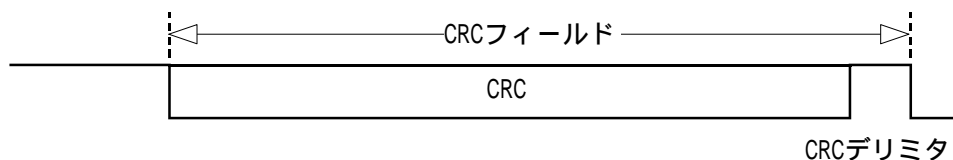


図5.7 CRCフィールド

- (* 6) ACKフィールド : 正常受信確認用です。1bitのACKスロット、1bitのACKデリミタから成ります。

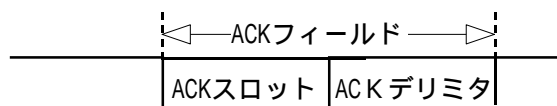


図5.8 ACKフィールド

・ H8S/2623F受信側がCRCチェックでエラーを検出した場合はACKスロット=Highを、検出なかった場合はACKスロット=Lowを出力します。

- (* 7) スタッフビット : データフレーム上でLowが5bit続いた場合、次の1bitは必ずHighが出力されます。同様にHighが5bit続いた場合、次の1bitは必ずLowが出力されます。これらの様にスタッフビットが出力される場合データフレームのビット長は、スタッフビットの分だけ長くなります。

例として、設定値を

アービトレーションフィールド : " 101010101010 "

コントロールフィールド : " 000001 "

とすると、" 101010101010000001 " となり、最下位から2bit目 (0が5つ続いた後) にスタッフビット () が出力されます。

よって、CANバス上に送信される値は " 1010101010100000101 " となり、スタッフビットの1bit分だけ長くなります。

- (* 8) コンフィギュレーションモード : HCANモジュールがリセット状態のことであり、解除はマスターコントロールレジスタ(MCR)のリセットリクエストビット(MCRO)をクリアすることで行います。

H8Sシリーズ内蔵HCAN アプリケーションノート

発行年月日 平成 11年 9月 第 1版

発行 株式会社 日立製作所
半導体グループ 電子統括営業本部

編集 株式会社 超Lメディア
技術ドキュメントグループ

©株式会社 日立製作所 1999

H8S シリーズ内蔵 HCAN アプリケーションノート



ルネサスエレクトロニクス株式会社
神奈川県川崎市中原区下沼部1753 〒211-8668

ADJ-502-080