

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。

SH7047F

コントローラエリアネットワーク(HCAN2)

アプリケーションノート

要旨

本アプリケーションノートは、HCAN2 を使用した通信動作例から構成されており、ユーザにてソフトウェア設計およびハードウェア設計の際、ご参考として役立てていただけるようにまとめたものです。

尚、本アプリケーションノートに掲載されているタスク例及びアプリケーション例は動作確認しておりますが、実際にご使用になる場合には、必ず動作確認の上ご使用くださいますようお願いいたします。

動作確認デバイス

SH7047

目次

1.	例題1(スタンダードフォーマット,1バイトデータ)	3
1.1	送受信仕様.....	3
1.2	送受信仕様機能説明.....	4
1.3	送信側フロー.....	5
1.4	ソフトウェア説明(送信側).....	6
1.5	送信プログラム.....	7
1.6	受信側フロー.....	8
1.7	ソフトウェア説明(受信側).....	9
1.8	受信プログラム.....	10
1.9	動作波形(送受信).....	11
2.	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)	12
2.1	送受信仕様.....	12
2.2	送受信仕様機能説明.....	13
2.3	送信側フロー.....	15
2.4	ソフトウェア説明(送信側).....	17
2.5	送信プログラム.....	18
2.6	受信側フロー.....	20
2.7	ソフトウェア説明(受信側).....	22
2.8	受信プログラム.....	23
2.9	動作波形(送受信).....	25

3.	例題3(エクステンデッドフォーマット,1バイトデータ)	26
3.1	送受信仕様.....	26
3.2	送受信仕様機能説明.....	27
3.3	送信側フロー.....	28
3.4	ソフトウェア説明(送信側).....	30
3.5	送信プログラム.....	31
3.6	受信側フロー.....	33
3.7	ソフトウェア説明(受信側).....	35
3.8	受信プログラム.....	36
3.9	動作波形(送受信).....	38
4.	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)	39
4.1	送受信仕様.....	39
4.2	送受信仕様機能説明.....	41
4.3	送信側フロー.....	43
4.4	ソフトウェア説明(送信側).....	45
4.5	送信プログラム.....	46
4.6	受信側フロー.....	50
4.7	ソフトウェア説明(受信側).....	52
4.8	受信プログラム.....	53
4.9	動作波形(送受信).....	55
5.	例題5(リモートフレーム)	56
5.1	送受信仕様.....	56
5.2	送受信仕様機能説明.....	57
5.3	送信側フロー.....	59
5.4	ソフトウェア説明(送信側).....	61
5.5	送信プログラム.....	62
5.6	受信側フロー.....	64
5.7	ソフトウェア説明(受信側).....	66
5.8	受信プログラム.....	67
5.9	動作波形(送受信).....	69

1. 例題1(スタンダードフォーマット,1バイトデータ)

1.1 送受信仕様

SH7047F を 2 個使用した、スタンダードフォーマットによる 1 バイトデータ送受信。

送受信共通仕様

- 通信速度は 250Kbps(50MHz 動作時)とします。
- Identifier は H 555 とします。

送信側仕様

- メールボックス 1 を使用します。
- データ長は 1 バイトとし、送信データは H AA とします。
- 送信中は送信完了フラグをポーリングします。
- 送信完了フラグセットを確認後、送信完了フラグをクリアし終了します。

受信側仕様

- メールボックス 0 を使用します。
- Identifier をマスクして一致すれば受信します。
- 受信データは内蔵 RAM に格納し、終了します。

1.2 送受信仕様機能説明

- 表 1.1 に使用端子及び関連レジスタの機能割り付けを示します。

表 1.1 HCAN2 機能割り付け

使用端子		機能	
端子	HTxD1	HCAN2 によるメッセージの送信を行ないます。(57Pin)	
	HRxD1	HCAN2 によるメッセージの受信を行ないます。(56Pin)	
関連レジスタ		機能	
送受信 共通レジスタ	PBCR1	ポート B コントロールレジスタ HTxD1、HRxD1 端子の機能に設定します。	
	PBCR2		
	MCR	マスタコントロールレジスタ HCAN2 の動作を制御します。	
	BCR0	ビットタイミングコンフィギュレーションレジスタ 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。	
	BCR1		
	IRR	インタラプトリクエストレジスタ 各割り込み要因のステータスを示します。	
	MBx	MC0	メッセージコントロールレジスタ Identifier 設定、データフレームかりモートフレームの設定を行ないます。
		MC4	メッセージコントロールレジスタ 送信/受信の設定を行います。
		MC5	メッセージコントロールレジスタ データ長の設定を行います。
		MD7	メッセージデータレジスタ 送受信された CAN のメッセージデータを格納します。
		LAFM15	ローカルアクセプタンスフィルタマスク 受信用メールボックスの Identifier 用フィルタマスクの設定を行ないます。
送信用 レジスタ	TXPR	送信待ちレジスタ 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	送信アクノレッジレジスタ 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	RXPR	受信完了レジスタ 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。	

1.3 送信側フロー

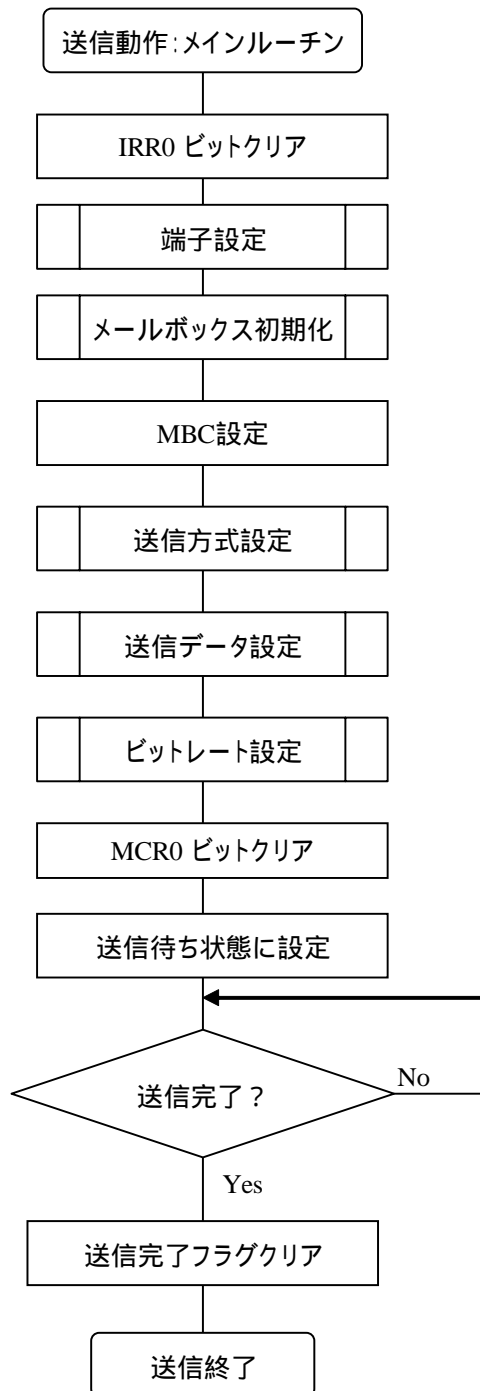


図 1.1 送信側フローチャート

1.4 ソフトウェア説明(送信側)

- 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	t_main	HCAN2 の初期設定及び送信の設定を行います。

- 使用レジスタの説明

レジスタ名	機能	設定値	モジュール名
PB.PBCR1	PB1,0 端子 (56,57pin) を HRxD1、HTxD1 に設定。	0x0000	メインルーチン
PB.PBCR2		0x000F	
work	MB 初期化に使用するワークレジスタ。		
HCAN_MCR	リセットリクエストビットクリア。	0x0000	
HCAN_IRR	リセット/ホールド/スリープ割込みフラグクリア。 (クリア条件: 1 ライト)	0x0001	
HCAN_BCR0	ビットレートを =50MHz 時 250Kbps に設定。	0x0009	
HCAN_BCR1		0x4300	
HCAN_MB1.MC0	MB1 をデータフレーム、スタンダードフォーマットに設定。また、Identifier (H 555) を設定。	0x5550	
HCAN_MB1.MC4	MB1 を送信用に設定。	0x01	
HCAN_MB1.MC5	MB1 の送信サイズを 1 バイトのデータ長に設定。	0x01	
HCAN_MB1.MD7	MB1 の送信データ (H AA) を設定。	0xAA	
HCAN_TXPRO	MB1 を送信待ち状態に設定。	0x0002	
HCAN_TXACK0	MB1 送信完了フラグのチェックとクリア。 (クリア条件: 1 ライト)	0x0002	

1.5 送信プログラム

```

/*****
/* Filename      : t_main.c                               */
/* Written       : '06/07/24 Ver.4.0                       */
/* Purpose       : HCAN2 for SH7047                       */
/*****
/*****
/* HCAN2 送信プログラム(例題 1)                           */
/*****
#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 "SH7047.h"        /* 周辺レジスタ定義ヘッダ ファイル */
/*****
/* 関数プロトタイプ宣言                                   */
/*****
void t_main(void );
/*****
/* メインルーチン                                       */
/*****
void t_main(void){
    unsigned short *work;

/* IRRO ビットクリア */
HCAN_IRR = 0x0001;      /* リセット割り込みフラグのクリア(クリア条件:1ライト) */
/* 端子設定 */
PB.PBCR1 = 0x0000;     /* PB HCAN2 設定 */
PB.PBCR2 = 0x000F;     /* PB HCAN2 設定 */
/* メールボックス初期化 */
work = (unsigned short *)0xFFFFB100;
do {
    *work = 0x0000;
    work++;
} while(work < (unsigned short *)0xFFFFB4F4);
/* MBC 設定 */
HCAN_MB1.MC4 = 0x01;   /* メールボックス1を送信用に設定 */
/* 送信方式設定 */
HCAN_MB1.MC0 = 0x5550; /* データレム、スタンダートフォーマットの選択、Identifier の設定 */
HCAN_MB1.MC5 = 0x01;   /* データ長:1バイト設定 */
/* 送信データ設定 */
HCAN_MB1.MD7 = 0xAA;   /* 送信データ:10101010 設定 */
/* ビットレート(BCR)設定: =50MHz 時ビットレート 250Kbps */
HCAN_BCR0 = 0x0009;   /* BRP=9(10システムクロック) */
HCAN_BCR1 = 0x4300;   /* TSEG1=4(5tq),TSEG2=3(4tq) */
/* MCRO ビットクリア */
HCAN_MCR = 0x0000;    /* リセットリクエストビットのクリア */
/* メッセージの送信待ち状態に設定 */
HCAN_TXPRO = 0x0002;   /* メールボックス1を送信待ち状態に設定 */
/* 送信完了待ち */
while((HCAN_TXACK0 & 0x0002) != 0x0002);
/* 送信完了フラグのクリア */
HCAN_TXACK0 = 0x0002;   /* 送信完了フラグのクリア(クリア条件:1ライト) */
while(1);
}
    
```

1.6 受信側フロー

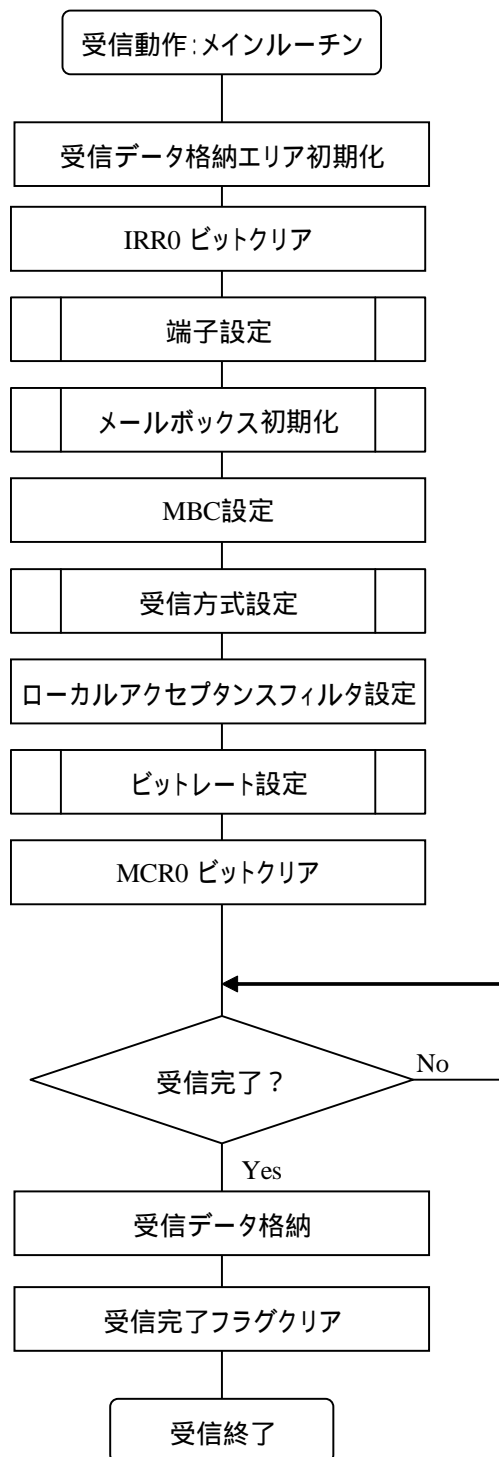


図 1.2 受信側フロ - チャート

1.7 ソフトウェア説明(受信側)

- 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	r_main	HCAN2 の初期設定及び受信の設定を行います。

- 使用レジスタの説明

レジスタ名	機能	設定値	モジュール名
MBbuff	受信データ格納エリア(アドレス : H FFFFD100)		メインルーチン
work	MB 初期化に使用するワークレジスタ。		
PB.PBCR1	PB1,0 端子 (56,57pin) を HRxD1、HTxD1 に設定。	0x0000	
PB.PBCR2		0x000F	
HCAN_MCR	リセットリクエストビットクリア。	0x0000	
HCAN_IRR	リセット/ホールド/スリープ割込みフラグクリア。 (クリア条件 : 1 ライト)	0x0001	
HCAN_MBO.MC4	MB0 を受信用に設定。	0x02	
HCAN_BCRO	ビットレートを =50MHz 時 250Kbps に設定。	0x0009	
HCAN_BCR1		0x4300	
HCAN_MBO.MC0	MB0 をデータフレーム、スタンダードフォーマットに設定。また、Identifier (H 555) を設定。	0x5550	
HCAN_MBO.LAFM15	MB0 の Identifier 用フィルタマスクを設定。	0x0000	
HCAN_RXPRO	MB0 受信完了フラグのチェックとクリア。 (クリア条件 : 1 ライト)	0x0001	

1.8 受信プログラム

```

/*****
/* Filename      : r_main.c
/* Written       : '06/07/24 Ver.4.0
/* Purpose       : HCAN2 for SH7047
/*****
/*****
/* HCAN2 受信プログラム(例題 1)
/*****
#include <stdio.h>          /*ライブラリ関数用ヘッダファイル */
#include <machine.h>        /*ライブラリ関数用ヘッダファイル */
#include "SH7047.h"         /*周辺レジスタ定義ヘッダファイル */
/*****
/* 関数プロトタイプ宣言
/*****
void r_main(void);
/*****
/* 定数定義
/*****
#define MBbuff (*(unsigned char *)0xFFFFD100) /*メールボックス0の受信データ格納 */
/*****
/* メインルーチン
/*****
void r_main(void){
    unsigned short *work;

/* 受信データ格納エリア初期化 */
    MBbuff = 0x00;
/* IRRO ビットクリア */
    HCAN_IRR = 0x0001;          /* リセット割り込みフラグのクリア(クリア条件:1ライト) */
/* 端子設定 */
    PB.PBCR1 = 0x0000;         /* PB HCAN2 設定 */
    PB.PBCR2 = 0x000F;         /* PB HCAN2 設定 */
/* メールボックス初期化 */
    work = (unsigned short *)0xFFFFB100;
    do {
        *work = 0x0000;
        work++;
    } while(work < (unsigned short *)0xFFFFB4F4);
/* MBC 設定 */
    HCAN_MBO.MC4 = 0x02;        /* メールボックス0を受信用に設定 */
/* 受信方式設定 */
    HCAN_MBO.MC0 = 0x5550;      /* データレム、スタンダードフォーマットの選択、Identifier の設定 */
/* ローカルアクセプタンスフィルタの設定 */
    HCAN_MBO.LAFM15 = 0x0000;   /* メールボックス0のIdentifier用のフィルタマスクの設定 */
/* ビットレート(BCR)設定: =50MHz時ビットレート 250Kbps */
    HCAN_BCR0 = 0x0009;         /* BRP=9(10システムクロック) */
    HCAN_BCR1 = 0x4300;         /* TSEG1=4(5tq),TSEG2=3(4tq) */
/* MCRO ビットクリア */
    HCAN_MCR = 0x0000;          /* リセットレジスタビットのクリア */
/* 受信完了待ち */
    while((HCAN_RXPRO & 0x0001) != 0x0001); /* 受信完了待ち */
/* 受信データ格納 */
    MBbuff = HCAN_MBO.MD7;      /* 受信データを内蔵RAMに格納 */
/* 受信完了フラグクリア */
    HCAN_RXPRO = 0x0001;        /* 受信完了フラグクリア(クリア条件:1ライト) */
    while(1);
}

```

1.9 動作波形(送受信)

図 1.3 に本アプリケーション動作時の波形を示します。

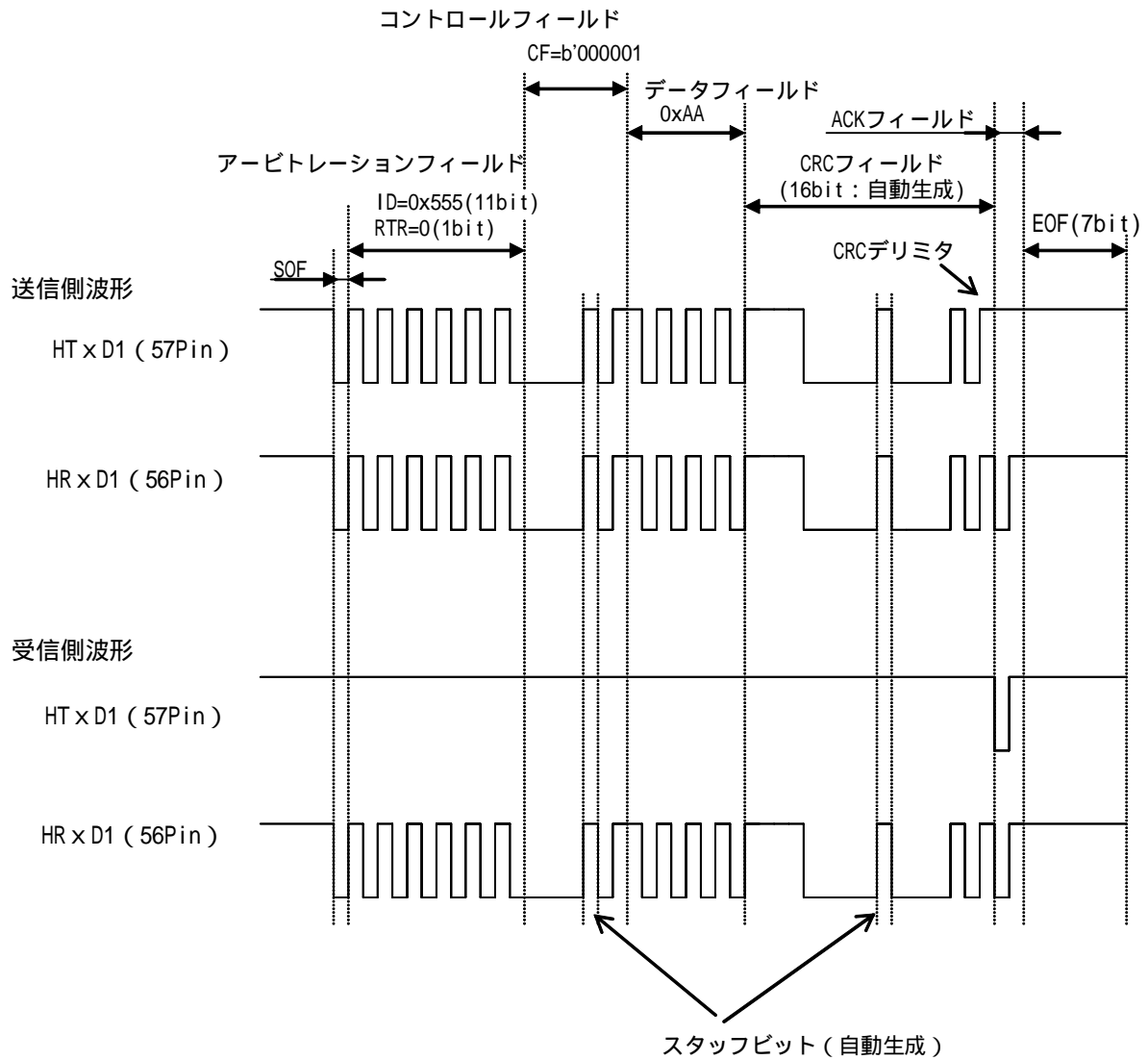


図 1.3 動作波形

2. 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 送受信仕様

SH7047F を 2 個使用した、スタンダードフォーマットによる 8 バイトデータ送受信及び DTC を使用した受信データ格納。

送受信共通仕様

- 通信速度は 250Kbps(50MHz 動作時)とします。
- Identifier は H 555 とします。

送信側仕様

- メールボックス 1 を使用します。
- データ長は 8 バイトとし、送信データは H 55, H 66, H 77, H 88, H 99, H AA, H BB, H FF とします。
- メールボックス空き割り込み(IRR8)を使用します。
メッセージの送信待ちに設定後、メールボックス空き割り込みルーチン内で送信完了フラグをクリアとメールボックス空き割り込みを禁止し、終了します。

受信側仕様

- メールボックス 0 を使用します。
- Identifier をマスクして一致すれば受信します。
- 受信メッセージ割り込み(IRR1)を使用します。
 - (a) 受信データは受信メッセージ割り込みで DTC を起動し、内蔵 RAM に格納します。
 - (b) DTC 転送はブロック転送モードを使用し、8 バイトを 1 ブロックとして 1 ブロック転送します。
 - (c) DTC 転送終了後の受信メッセージ割り込みルーチン内で受信完了フラグをクリアと受信メッセージ割り込みを禁止し、終了します。

2.2 送受信仕様機能説明

- 表 2.1~2.2 に関連レジスタの機能割り付けを示します。
 (使用端子及びポートレジスタは例題 1 参照)

表 2.1 HCAN2 機能割り付け

関連レジスタ		機能	
送受信共通 レジスタ	MCR	マスタコントロールレジスタ HCAN2 の動作を制御します。	
	BCR0	ビットタイミングコンフィギュレーションレジスタ CAN のポ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。	
	BCR1		
	IRR	インタラプトリクエストレジスタ 各割り込み要因のステータスを示します。	
	MBx	MC0	メッセージコントロールレジスタ Identifier 設定、データフレームカリモートフレームの設定を 行ないます。
		MC4	メッセージコントロールレジスタ 送信/受信の設定を行います。
		MC5	メッセージコントロールレジスタ データ長の設定を行います。
MD7 ~ MD14		メッセージデータレジスタ 送受信された CAN のメッセージデータを格納します。	
LAFM15	ローカルアクセプタンスフィルタマスク 受信用メールボックスの Identifier 用フィルタマスクの設定を 行ないます。		
送信用 レジスタ	TXPR	送信待ちレジスタ 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	送信アクノレッジレジスタ 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	RXPR	受信完了レジスタ 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。	
割り込み関連 レジスタ	MBIMR	メールボックスインタラプトマスクレジスタ 各メールボックスの割り込み要求をイネーブルにします。	
	IMR	インタラプトマスクレジスタ IRR の割り込みフラグによる割り込み要求をイネーブルにします。	
	IPRK	インタラプトプライオリティレジスタ HCAN2 の割り込み要求の優先順位を設定します。	

表 2.2 DTC 機能割り付け

関連レジスタ	機能
DTMR	DTC モードレジスタ DTC の動作モードの制御を行います。
DTSAR	DTC ソースアドレスレジスタ DTC の転送するデータの転送元アドレスを指定します。
DTDAR	DTC デスティネーションアドレスレジスタ DTC の転送するデータの転送先アドレスを指定します。
DTCRA	DTC 転送カウントレジスタ A DTC のデータ転送の転送回数を指定します。
DTCRB	DTC 転送カウントレジスタ B ブロック転送モードのとき、ブロック長を指定します。
DTBR	DTC 情報ベースレジスタ DTC 転送情報を格納するメモリアドレスの上位 16 ビットを指定します。
DTEF	DTC イネーブルレジスタ DTC を起動する割り込み要因 (HCAN2 の RM1) を選択します。

2.3 送信側フロー



図 2.1 送信側フロ - チャート(1)

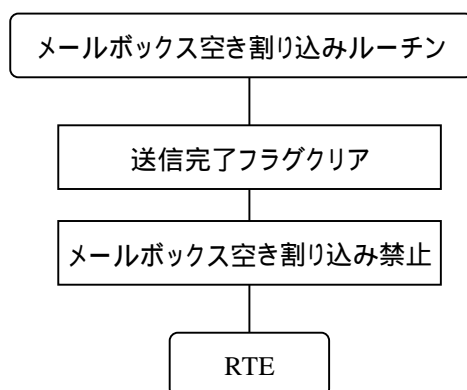


図 2.2 送信側フロ - チャート(2)

2.4 ソフトウェア説明(送信側)

- 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	t_main	HCAN2 の初期設定及び送信の設定を行います。
メールボックス空き 割り込みルーチン	SLE1_IRR8	送信完了フラグクリアと割り込み禁止の設定を行います。

- 使用レジスタの説明(使用端子及びポートレジスタは例題 1 参照)

レジスタ名	機能	設定値	モジュール名
work	MB 初期化に使用するワークレジスタ。		メインルーチン
HCAN_MCR	リセットリクエストビットクリア。	0x0000	
HCAN_IRR	リセット/ホールド/スリープ割り込みフラグクリア。 (クリア条件: 1 ライト)	0x0001	
HCAN_BCRO	ビットレートを =50MHz 時 250Kbps に設定。	0x0009	
HCAN_BCR1		0x4300	
HCAN_MB1.MC0	MB1 をデータフレーム、スタンダードフォーマットに 設定。また、Identifier (H 555) を設定。	0x5550	
HCAN_MB1.MC4	MB1 を送信用に設定。	0x01	
HCAN_MB1.MC5	MB1 の送信サイズを 8 バイトのデータ長に設定。	0x08	
HCAN_MB1.MD7	MB1 の 1 バイト目送信データ (H 55) を設定。	0x55	
HCAN_MB1.MD8	MB1 の 2 バイト目送信データ (H 66) を設定。	0x66	
HCAN_MB1.MD9	MB1 の 3 バイト目送信データ (H 77) を設定。	0x77	
HCAN_MB1.MD10	MB1 の 4 バイト目送信データ (H 88) を設定。	0x88	
HCAN_MB1.MD11	MB1 の 5 バイト目送信データ (H 99) を設定。	0x99	
HCAN_MB1.MD12	MB1 の 6 バイト目送信データ (H AA) を設定。	0xAA	
HCAN_MB1.MD13	MB1 の 7 バイト目送信データ (H BB) を設定。	0xBB	
HCAN_MB1.MD14	MB1 の 8 バイト目送信データ (H FF) を設定。	0xFF	
HCAN_IMR	メールボックス空き割り込み許可。	0xFEFF	
HCAN_MB1IMR0	MB1 の割り込み要求を許可。	0xFFFFD	
INTC.IPRK	HCAN2 の割り込み要求の優先順位を設定。	0x00F0	
HCAN_TXPRO	MB1 を送信待ち状態に設定。	0x0002	
HCAN_TXACK0	MB1 送信完了フラグのクリア。 (クリア条件: 1 ライト)	0x0002	メールボックス 空き割り込み ルーチン
HCAN_IMR	メールボックス空き割り込み禁止。	0xFFFF	

2.5 送信プログラム

```

/*****
/* Filename      : t_main.c                               */
/* Written       : '06/07/24 Ver.4.0                       */
/* Purpose       : HCAN2 for SH7047                       */
/*****
/*****
/* HCAN2 送信プログラム(例題2)                             */
/*****
#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 "SH7047.h"        /* 周辺レジスタ定義ヘッダ ファイル */
/*****
/* 関数プロトタイプ宣言                                     */
/*****
void t_main(void);
void SLE1_IRR8(void);
/*****
/* メインルーチン                                         */
/*****
void t_main(void){
    unsigned short *work;

/* IRR0 ビットクリア */
    HCAN_IRR = 0x0001;      /* リセット割り込みフラグのクリア(クリア条件:1ライト) */
/* 端子設定 */
    PB.PBCR1 = 0x0000;     /* PB HCAN2 設定 */
    PB.PBCR2 = 0x000F;     /* PB HCAN2 設定 */
/* メールボックス初期化 */
    work = (unsigned short *)0xFFFFB100;
    do {
        *work = 0x0000;
        work++;
    } while(work < (unsigned short *)0xFFFFB4F4);
/* MBC 設定 */
    HCAN_MB1.MC4 = 0x01;   /* メールボックス1を送信用に設定 */
/* 送信方式設定 */
    HCAN_MB1.MC0 = 0x5550; /* データフレーム、スタンダートフォーマットの選択、Identifier の設定 */
    HCAN_MB1.MC5 = 0x08;   /* データ長:8ビット設定 */
/* 送信データ設定 */
    HCAN_MB1.MD7 = 0x55;   /* 送信データ:01010101 設定 */
    HCAN_MB1.MD8 = 0x66;   /* 送信データ:01100110 設定 */
    HCAN_MB1.MD9 = 0x77;   /* 送信データ:01110111 設定 */
    HCAN_MB1.MD10 = 0x88;  /* 送信データ:10001000 設定 */
    HCAN_MB1.MD11 = 0x99;  /* 送信データ:10011001 設定 */
    HCAN_MB1.MD12 = 0xAA;  /* 送信データ:10101010 設定 */
    HCAN_MB1.MD13 = 0xBB;  /* 送信データ:10111011 設定 */
    HCAN_MB1.MD14 = 0xFF;  /* 送信データ:11111111 設定 */
/* 割り込みの設定 */
    HCAN_IMR = 0xFEFF;     /* IRR8 割り込み許可 */
    HCAN_MB1.MR0 = 0xFFFF; /* メールボックス1 割り込み要求を許可 */
    INTC.IPRK = 0x00F0;    /* SLE1 優先順位設定 */
/* ビットレート(BCR)設定: =50MHz 時ビットレート 250Kbps */
    HCAN_BCR0 = 0x0009;    /* BRP=9(10システムクロック) */
    HCAN_BCR1 = 0x4300;    /* TSEG1=4(5tq), TSEG2=3(4tq) */
/* MCR0 ビットクリア */
    HCAN_MCR = 0x0000;     /* リセットリクエストのクリア */
/* メッセージの送信待ち状態に設定 */
    HCAN_TXPRO = 0x0002;   /* メールボックス1を送信待ち状態に設定 */
    set_imask(0);
    while(1);
}

```

```
/*
 * メールボックス空き割り込みルーチン
 */
#pragma interrupt(SLE1_IRR8)
void SLE1_IRR8(void){
/* 送信完了フラグクリア */
    HCAN_TXACK0 = 0x0002; /* 送信完了フラグのクリア(クリア条件:1バイト) */
/* メールボックス空き割り込み禁止 */
    HCAN_IMR = 0xFFFF; /* IRR8 割り込み禁止 */
}
```

2.6 受信側フロー



図 2.3 受信側フローチャート(1)

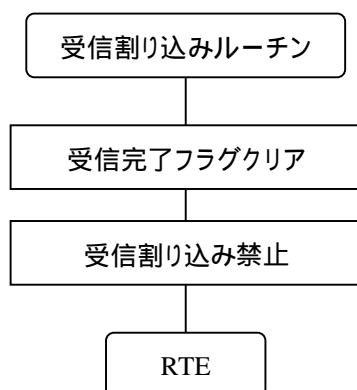


図 2.4 受信側フローチャート(2)

2.7 ソフトウェア説明(受信側)

- 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	r_main	HCAN2 の初期設定及び受信の設定を行います。
受信割り込みルーチン	RM1_IRR1	受信完了フラグクリアと割り込み禁止の設定を行います。

- 使用レジスタの説明(使用端子及びポートレジスタは例題 1 参照)

レジスタ名	機能	設定値	モジュール名
MBbuff	受信データ格納エリア。 (アドレス : H FFFFD100 ~ H FFFFD107)		メインルーチン
work	MB 初期化に使用するワークレジスタ。		
DTMR	DTSAR, DTDAR 共に転送後インクリメント、 ブロック転送モード、バイト転送に設定。	0xA890	
DTSAR	転送元アドレスに MB0 のメッセージデータ領域を 設定。	0xFFFFB108	
DTDAR	転送先アドレスに受信データ格納エリアを設定。	0xFFFFD100	
DTCRA	ブロック転送回数を設定。(1 回)	0x0001	
DTCRB	ブロック長を設定。(8 バイト)	0x0008	
DTC.DTBR	DTC 転送情報を格納するメモリアドレスの上位 16 ビット を指定。	0xFFFF	
DTC.DTEF	DTC を起動する割り込み要因(HCAN2 の RM1)を選択。	0x04	
HCAN_MCR	リセットリクエストビットクリア。	0x0000	
HCAN_IRR	リセット/ホールド/スリープ割り込みフラグクリア。 (クリア条件 : 1 ライト)	0x0001	
HCAN_BCRO	ビットレートを =50MHz 時 250Kbps に設定。	0x0009	
HCAN_BCR1		0x4300	
HCAN_MB0.MC0	MB0 をデータフレーム、スタンダードフォーマットに 設定。また、Identifier (H 555) を設定。	0x5550	
HCAN_MB0.MC4	MB0 を受信用に設定。	0x02	
HCAN_MB0.LAFM15	MB0 の Identifier 用フィルタマスクを設定。	0x0000	
HCAN_IMR	受信メッセージ割り込み許可。	0xFFFFD	
HCAN_MBIMR0	MB0 の割り込み要求を許可。	0xFFFFE	
INTC.IPRK	HCAN2 の割り込み要求の優先順位を設定。	0x00F0	
HCAN_RXPRO	MB0 受信完了フラグのクリア。 (クリア条件 : 1 ライト)	0x0001	
HCAN_IMR	受信メッセージ割り込み禁止。	0xFFFF	

2.8 受信プログラム

```

/*****
/* Filename      :   r_main.c                */
/* Written       :   '06/07/24 Ver.4.0      */
/* Purpose       :   HCAN2 for SH7047      */
/*****
/*****
/* HCAN2 受信プログラム(例題2)              */
/*****
#include <stdio.h>           /*ライブラリ関数用ヘッダファイル */
#include <machine.h>        /*ライブラリ関数用ヘッダファイル */
#include "SH7047.h"         /*周辺レジスタ定義ヘッダファイル */
/*****
/* 関数プロトタイプ宣言                    */
/*****
void r_main(void);
void RM1_IRR1(void);
/*****
/* 定数定義                                */
/*****
#define DTMR (* (unsigned short *)0xFFFFD080) /* DTC レジスタ情報 */
#define DTCRA (* (unsigned short *)0xFFFFD082) /* DTC レジスタ情報 */
#define DTCRB (* (unsigned short *)0xFFFFD086) /* DTC レジスタ情報 */
#define DTSAR (* (unsigned long *)0xFFFFD088) /* DTC レジスタ情報 */
#define DTDAR (* (unsigned long *)0xFFFFD08C) /* DTC レジスタ情報 */
#define MBuff (* (unsigned char *)0xFFFFD100) /* 受信データ格納エリア */
/*****
/* メインルーチン                          */
/*****
void r_main(void){
    unsigned short *work;

/* 受信データ格納エリア初期化 */
    work = (unsigned short *)0xFFFFD100;
    do {
        *work = 0x0000;
        work++;
    } while(work < (unsigned short *)0xFFFFD108);
/* DTC 設定 */
    DTMR = 0xA890; /* DTSAR,DTDAR 共に転送後インクリメント,ブロック転送モード,バイト転送 */
    DTSAR = (unsigned long)&HCAN_MBO.MD7; /* 転送元アドレス設定 */
    DTDAR = (unsigned long)&MBuff; /* 転送先アドレス設定 */
    DTCRA = 0x0001; /* ブロック転送 1 回 */
    DTCRB = 0x0008; /* ブロック長 8 バイト */
    DTC.DTBR = 0xFFFF; /* レジスタ情報先頭アドレス */
    DTC.DTEF |= 0x04; /* HCAN2 割込み(RM1) */
/* IRRO ビットクリア */
    HCAN_IRR = 0x0001; /* レジスタ割り込みフラグのクリア(クリア条件:1 ライト)
/* 端子設定 */
    PB.PBCR1 = 0x0000; /* PB HCAN2 設定 */
    PB.PBCR2 = 0x000F; /* PB HCAN2 設定 */
/* メールボックス初期化 */
    work = (unsigned short *)0xFFFFB100;
    do {
        *work = 0x0000;
        work++;
    } while(work < (unsigned short *)0xFFFFB4F4);
/* MBC 設定 */
    HCAN_MBO.MC4 = 0x02; /* メールボックス0を受信用に設定 */
/* 割り込みの設定 */
    HCAN_IMR = 0xFFFF; /* 受信メッセージ割り込み許可 */
    HCAN_MBIMRO = 0xFFFE; /* メールボックス0受信割り込み要求許可 */
    INTC.IPRK = 0x00F0; /* RM1 優先順位設定 */
/* 受信方式設定 */
    HCAN_MBO.MCO = 0x5550; /* データレーム、スタンダードフォーマットの選択、Identifier の設定 */

```

```

/* ローカルアクセプタンスフィルタの設定 */
HCAN_MB0.LAFM15 = 0x0000; /* メルプッス0のIdentifier用のフィルタマスクの設定 */
/* ビットレート(BCR)設定: =50MHz時ビットレート 250Kbps */
HCAN_BCRO = 0x0009; /* BRP=9(10システムクロック) */
HCAN_BCR1 = 0x4300; /* TSEG1=4(5tq),TSEG2=3(4tq)*/
/* MCRO ビットクリア */
HCAN_MCR = 0x0000; /* リセットリクエストビットのクリア

set_imask(0);
while(1);
}

/*****
/* 受信割り込みルーチン */
/*****
#pragma interrupt(RM1_IRR1)
void RM1_IRR1(void){

/* 受信完了レジスタクリア */
HCAN_RXPRO = 0x0001; /* 受信完了フラグクリア(クリア条件:1バイト) */
/* 受信割り込み禁止 */
HCAN_IMR = 0xFFFF; /* 受信メッセージ割り込み禁止

}
    
```

2.9 動作波形(送受信)

図 2.5 に本アプリケーション動作時の波形を示します。

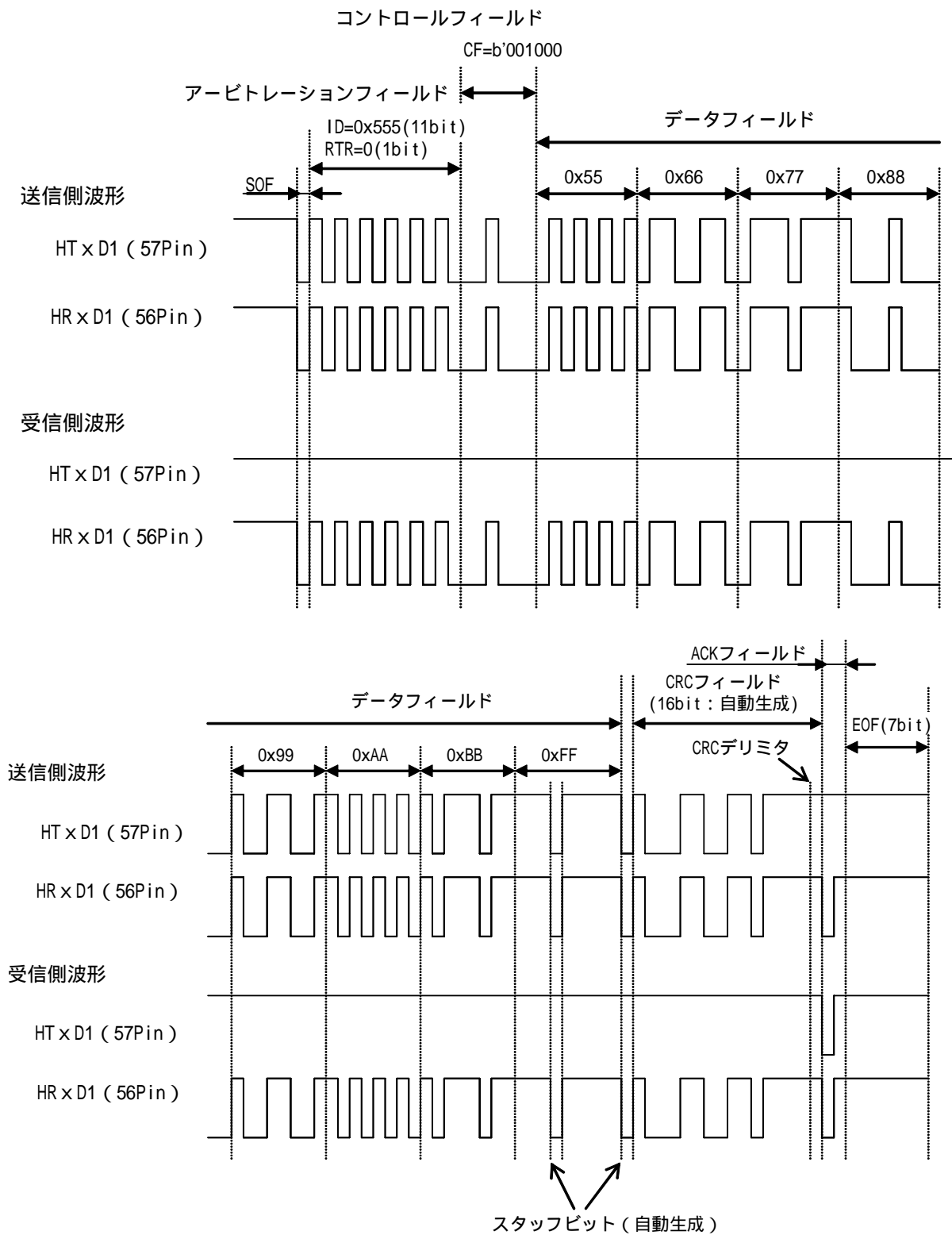


図 2.5 動作波形

3. 例題3(エクステンデッドフォーマット,1バイトデータ)

3.1 送受信仕様

SH7047F を 2 個使用したエクステンデッドフォーマットによる 1 バイトデータ送受信。

送受信共通仕様

- 通信速度は 250Kbps(50MHz 動作時)とします。
- スタンダード Identifier は H 555、エクステンデッド Identifier は H 2AAAA とします。

送信側仕様

- メールボックス 1 を使用します。
- データ長は 1 バイトとし、送信データは H AA とします。
- メールボックス空き割り込み (IRR8) を使用します。
 メッセージの送信待ちに設定後、メールボックス空き割り込みルーチン内で送信完了フラグをクリアとメールボックス空き割り込みを禁止し、終了します。

受信側仕様

- メールボックス 0 を使用します。
- Identifier をマスクして一致すれば受信します。
- 受信メッセージ割り込み (IRR1) を使用します。
 受信メッセージ割り込みルーチン内で受信データを内蔵 RAM に格納し、受信完了フラグをクリアと受信メッセージ割り込みを禁止し、終了します。

3.2 送受信仕様機能説明

- 表 3.1 に関連レジスタの機能割り付けを示します。
 (使用端子及びポートレジスタは例題 1 参照)

表 3.1 HCAN2 機能割り付け

関連レジスタ		機能	
送受信共通 レジスタ	MCR	マスタコントロールレジスタ HCAN2 の動作を制御します。	
	BCR0	ビットタイミングコンフィギュレーションレジスタ CAN のポ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。	
	BCR1		
	IRR	インタラプトリクエストレジスタ 各割り込み要因のステータスを示します。	
	MBx	MC0 ~ MC2	メッセージコントロールレジスタ Identifier 設定、データフレームかりリモートフレームの設定を 行ないます。
		MC4	メッセージコントロールレジスタ 送信/受信の設定を行います。
		MC5	メッセージコントロールレジスタ データ長の設定を行います。
		MD7	メッセージデータレジスタ 送受信された CAN のメッセージデータを格納します。
LAFM15		ローカルアクセプタンスフィルタマスク 受信用メールボックスの Identifier 用フィルタマスクの設定を 行ないます。	
LAFM17			
送信用 レジスタ	TXPR	送信待ちレジスタ 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	送信アクノレッジレジスタ 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	RXPR	受信完了レジスタ 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。	
割り込み関連 レジスタ	MBIMR	メールボックスインタラプトマスクレジスタ 各メールボックスの割り込み要求をイネーブルにします。	
	IMR	インタラプトマスクレジスタ IRR の割り込みフラグによる割り込み要求をイネーブルにします。	
	IPRK	インタラプトプライオリティレジスタ HCAN2 の割り込み要求の優先順位を設定します。	

3.3 送信側フロー

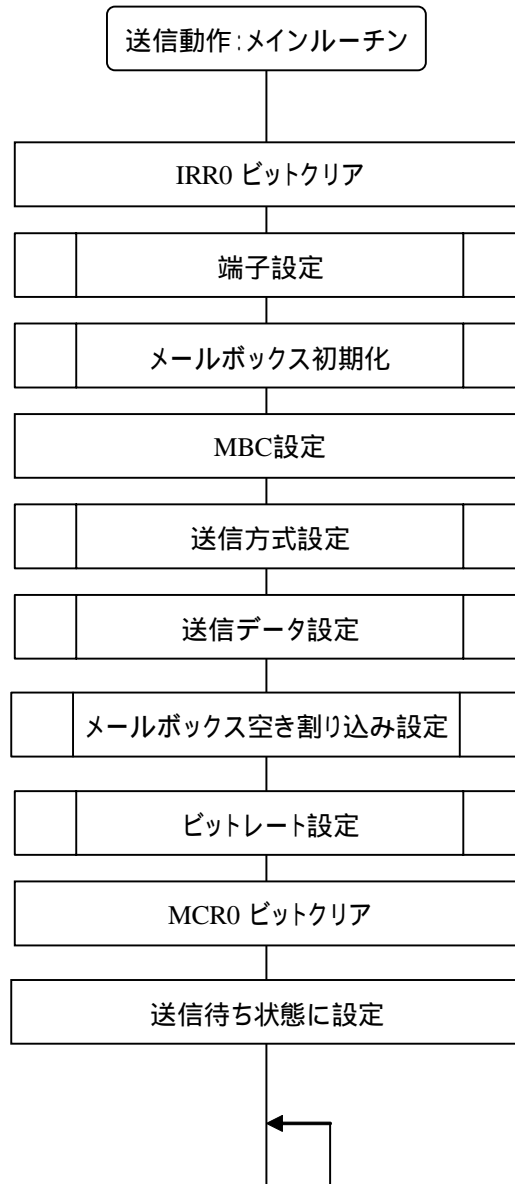


図 3.1 送信側フローチャート(1)

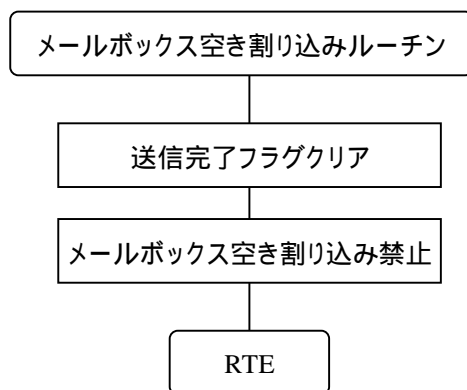


図 3.2 送信側フローチャート(2)

3.4 ソフトウェア説明(送信側)

- 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	t_main	HCAN2 の初期設定及び送信の設定を行います。
メールボックス空き 割り込みルーチン	SLE1_IRR8	送信完了フラグクリアと割り込み禁止の設定を行います

- 使用レジスタの説明(使用端子及びポートレジスタは例題 1 参照)

レジスタ名	機能	設定値	モジュール名
work	MB 初期化に使用するワークレジスタ。		メインルーチン
HCAN_MCR	リセットリクエストビットクリア。	0x0000	
HCAN_IRR	リセット/ホールド/スリープ割り込みフラグクリア。 (クリア条件:1 ライト)	0x0001	
HCAN_BCR0	ビットレートを =50MHz 時 250Kbps に設定。	0x0009	
HCAN_BCR1		0x4300	
HCAN_MB1.MC0	MB1 をデータフレーム、エクステンデッドフォーマット に設定。また、スタンダード Identifier (H 555)、拡張 Identifier(H 2AAAA)を設定。	0x5556	
HCAN_MB1.MC2		0xAAAA	
HCAN_MB1.MC4	MB1 を送信用に設定。	0x01	
HCAN_MB1.MC5	MB1 の送信サイズを 1 バイトのデータ長に設定。	0x01	
HCAN_MB1.MD7	MB1 の送信データ (H AA) を設定。	0xAA	
HCAN_IMR	メールボックス空き割り込み許可。	0xFEFF	
HCAN_MB1MR0	MB1 の割り込み要求を許可。	0xFFFD	
INTC.IPRK	HCAN2 の割り込み要求の優先順位を設定。	0x00F0	
HCAN_TXPRO	MB1 を送信待ち状態に設定。	0x0002	
HCAN_TXACK0	MB1 送信完了フラグのクリア。 (クリア条件:1 ライト)	0x0002	メールボックス 空き割り込み ルーチン
HCAN_IMR	メールボックス空き割り込み禁止。	0xFFFF	

3.5 送信プログラム

```

/*****
/* Filename      : t_main.c                               */
/* Written       : '06/07/24 Ver.4.0                     */
/* Purpose       : HCAN2 for SH7047                       */
/*****
/*****
/* HCAN2 送信プログラム(例題3)                            */
/*****
#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 "SH7047.h"         /* 周辺レジスタ定義ヘッダ ファイル */
/*****
/* 関数プロトタイプ宣言                                    */
/*****
void t_main(void);
void SLE1_IRR8(void);
/*****
/* メインルーチン                                        */
/*****
void t_main(void){
    unsigned short *work;

/* IRR0 ビットクリア */
    HCAN_IRR = 0x0001;          /* レット割り込みフラグのクリア(クリア条件:1バイト)*/
/* 端子設定 */
    PB.PBCR1 = 0x0000;         /* PB HCAN2 設定 */
    PB.PBCR2 = 0x000F;         /* PB HCAN2 設定 */
/* メールボックス初期化 */
    work = (unsigned short *)0xFFFFB100;
    do {
        *work = 0x0000;
        work++;
    } while(work < (unsigned short *)0xFFFFB4F4);
/* MBC 設定 */
    HCAN_MB1.MC4 = 0x01;       /* メールボックス1を送信用に設定 */
/* 送信方式設定 */
    HCAN_MB1.MC0 = 0x5556;     /* データレーム、エクステンデッドフォーマットの選択、Identifier の設定 */
    HCAN_MB1.MC2 = 0xAAAA;     /* Identifier(拡張)の設定 */
    HCAN_MB1.MC5 = 0x01;       /* データ長:1バイト設定 */
/* 送信データ設定 */
    HCAN_MB1.MD7 = 0xAA;       /* 送信データ:10101010 設定 */
/* 割り込みの設定 */
    HCAN_IMR = 0xFEFF;         /* IRR8 割り込み許可 */
    HCAN_MB1MRO = 0xFFFFD;     /* メールボックス1 割り込み要求を許可 */
    INTC.IPRK = 0x00F0;        /* SLE1 優先順位設定 */
/* ビットレート(BCR)設定: =50MHz 時ビットレート 250Kbps */
    HCAN_BCRO = 0x0009;        /* BRP=9(10システムクロック) */
    HCAN_BCR1 = 0x4300;        /* TSEG1=4(5tq),TSEG2=3(4tq) */
/* MCRO ビットクリア */
    HCAN_MCR = 0x0000;         /* レットリクエストビットのクリア */
/* 送信待ち状態に設定 */
    HCAN_TXPRO = 0x0002;       /* メールボックス1を送信待ち状態に設定 */

    set_imask(0);
    while(1);
}

```

```

/*****/
/* メールボックス空き割り込みルーチン */
/*****/
#pragma interrupt(SLE1_IRR8)
void SLE1_IRR8(void){
/* 送信完了フラグクリア */
    HCAN_TXACK0 = 0x0002; /* 送信完了フラグのクリア(クリア条件:1バイト) */
/* メールボックス空き割り込み禁止 */
    HCAN_IMR = 0xFFFF; /* IRR8 割り込み禁止 */
}
    
```

3.6 受信側フロー



図 3.3 受信側フロ - チャート(1)

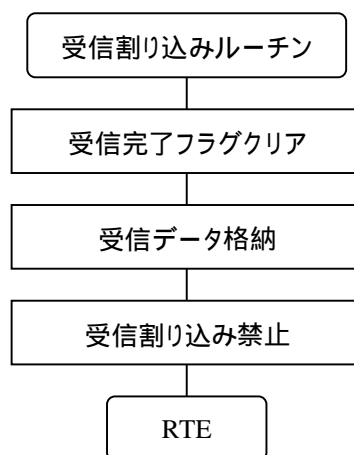


図 3.4 受信側フロ - チャート(2)

3.7 ソフトウェア説明(受信側)

- 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	r_main	HCAN2 の初期設定及び受信の設定を行います。
受信割り込みルーチン	RM1_IRR1	受信完了フラグクリアと割り込み禁止の設定を行います

- 使用レジスタの説明(使用端子及びポートレジスタは例題 1 参照)

レジスタ名	機能	設定値	モジュール名
MBbuff	受信データ格納エリア。(アドレス : H FFFFD100)		メインルーチン
work	MB 初期化に使用するワークレジスタ。		
HCAN_MCR	リセットリクエストビットクリア。	0x0000	
HCAN_IRR	リセット/ホールド/スリープ割り込みフラグクリア。 (クリア条件 : 1 ライト)	0x0001	
HCAN_BCRO	ビットレートを =50MHz 時 250Kbps に設定。	0x0009	
HCAN_BCR1		0x4300	
HCAN_MBO.MC0	MB1 をデータフレーム、エクステンデッドフォーマットに設定。また、スタンダード Identifier (H 555)、拡張 Identifier (H 2AAAA) を設定。	0x5556	
HCAN_MBO.MC2		0xAAAA	
HCAN_MBO.MC4	MB0 を受信用に設定。	0x02	
HCAN_MBO.LAFM15	MB0 の Identifier 用フィルタマスクを設定。	0x0000	
HCAN_MBO.LAFM17	MB0 の Identifier 用フィルタマスクを設定。	0x0000	
HCAN_IMR	受信メッセージ割り込み許可。	0xFFFF	
HCAN_MBIMR0	MB0 の割り込み要求を許可。	0xFFFE	
INTC.IPRK	HCAN2 の割り込み要求の優先順位を設定。	0x00F0	
HCAN_RXPRO	MB0 受信完了フラグのクリア。 (クリア条件 : 1 ライト)	0x0001	受信割り込みルーチン
HCAN_IMR	受信メッセージ割り込み禁止。	0xFFFF	

3.8 受信プログラム

```

/*****
/* Filename      :  r_main.c                */
/* Written       :  '06/07/24 Ver.4.0       */
/* Purpose       :  HCAN2 for SH7047        */
/*****
/*****
/* HCAN2 受信プログラム(例題 3)             */
/*****
#include <stdio.h>          /*ライブラリ関数用ヘッダファイル */
#include <machine.h>        /*ライブラリ関数用ヘッダファイル */
#include "SH7047.h"         /*周辺レジスタ定義ヘッダファイル */
/*****
/* 関数プロトタイプ宣言                     */
/*****
void r_main(void);
void RM1_IRR1(void);
/*****
/* 定数定義                                 */
/*****
#define MBbuff (*(unsigned char *)0xFFFFD100) /* 受信データ格納エリア */
/*****
/* メインルーチン                           */
/*****
void r_main(void){
    unsigned short *work;

/* 受信データ格納エリア初期化 */
    MBbuff = 0x00;
/* IRRO ビットクリア */
    HCAN_IRR = 0x0001; /* リセット割り込みフラグのクリア(クリア条件:1ライト)*/
/* 端子設定 */
    PB.PBCR1 = 0x0000; /* PB HCAN2 設定 */
    PB.PBCR2 = 0x000F; /* PB HCAN2 設定 */
/* メールボックス初期化 */
    work = (unsigned short *)0xFFFFB100;
    do {
        *work = 0x0000;
        work++;
    } while(work < (unsigned short *)0xFFFFB4F4);
/* MBC 設定 */
    HCAN_MBO.MC4 = 0x02; /* メールボックス0を受信用に設定 */
/* 割り込みの設定 */
    HCAN_IMR = 0xFFFFD; /* 受信メッセージ割り込み許可 */
    HCAN_MBIMRO = 0xFFFE; /* メールボックス0受信割り込み要求許可 */
    INTC.IPRK = 0x00F0; /* RM1優先順位設定 */
/* 受信方式設定 */
    HCAN_MBO.MC0 = 0x5556; /* データフレーム、エクステンデッドフォーマットの選択、Identifierの設定 */
    HCAN_MBO.MC2 = 0xAAAA; /* Identifier(拡張)の設定 */
/* ローカルアクセプタンスフィルタの設定 */
    HCAN_MBO.LAFM15 = 0x0000; /* メールボックス0のIdentifier用のフィルタマスクの設定 */
    HCAN_MBO.LAFM17 = 0x0000; /* メールボックス0のIdentifier用のフィルタマスクの設定 */
/* ビットレート(BCR)設定: =50MHz時ビットレート 250Kbps */
    HCAN_BCR0 = 0x0009; /* BRP=9(10システムクロック) */
    HCAN_BCR1 = 0x4300; /* TSEG1=4(5tq),TSEG2=3(4tq) */
/* MCRO ビットクリア */
    HCAN_MCR = 0x0000; /* リセットリクエストビットのクリア */

    set_imask(0);
    while(1);
}

```

```

/*****
/* 受信割り込みルーチン */
/*****
#pragma interrupt(RM1_IRR1)
void RM1_IRR1(void){
/* 受信完了レジスタクリア */
    HCAN_RXPRO = 0x0001; /* 受信完了フラグクリア(クリア条件:1バイト) */
/* 受信データ格納 */
    MBbuff = HCAN_MB0.MD7;
/* 受信割り込み禁止 */
    HCAN_IMR = 0xFFFF; /* IRR1 割り込み禁止 */

}
    
```

3.9 動作波形(送受信)

図 3.5 に本アプリケーション動作時の波形を示します。

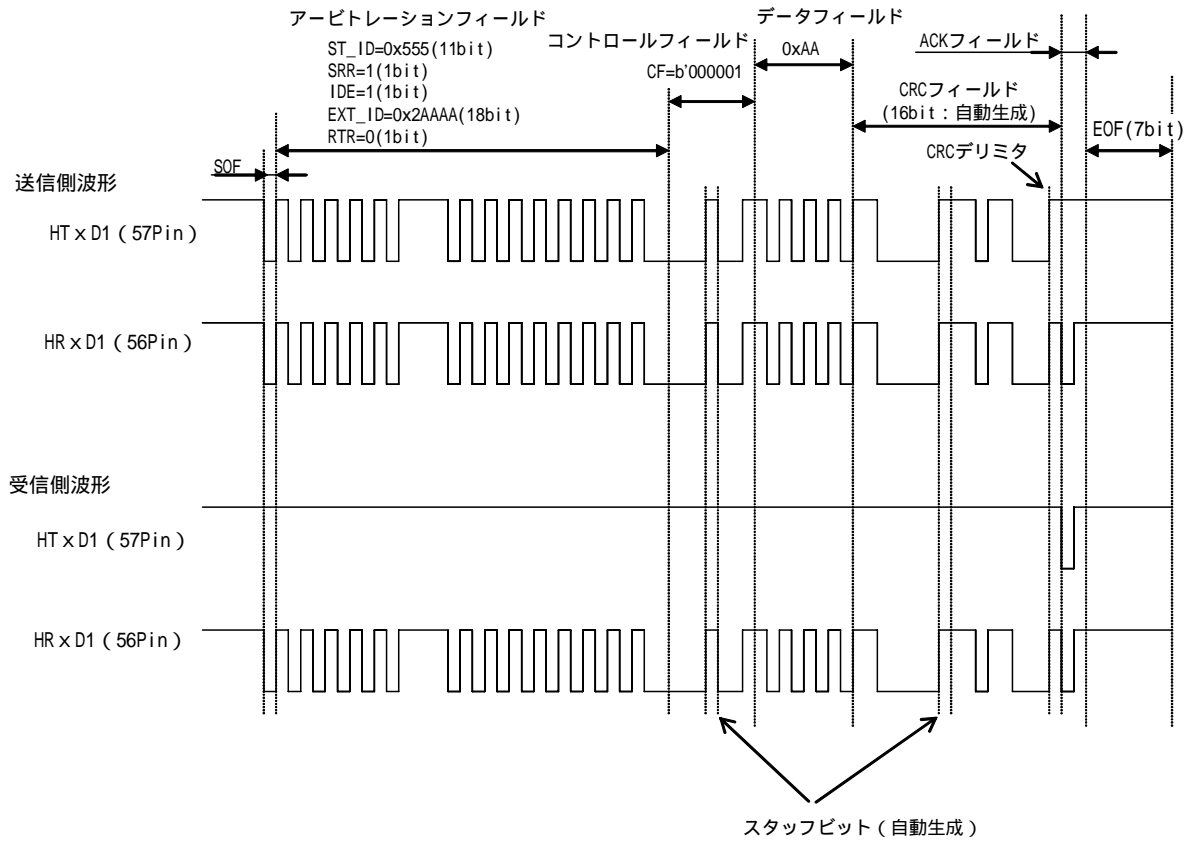


図 3.5 動作波形

4. 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 送受信仕様

SH7047F を 2 個使用した、スタンダードフォーマットの Identifier 優先順位による複数メールボックス送信及びメールボックス 0 のみでの受信。

送受信共通仕様

- 通信速度は 250Kbps(50MHz 動作時)とします。

送信側仕様

- メールボックス 1~ 15 を使用します。
- メッセージの送信は Identifier の優先順位とします。
- 各メールボックスのデータ長は 8 バイトとし、各メールボックスの Identifier 及びデータは表 4.1 の通りとします。

表 4.1 メールボックスの設定値

MBx	Identifier(11bit)	データ(8Byte)	優先順位
1	0x666	0x1111 1111 1111 1111	12
2	0x2AA	0x2222 2222 2222 2222	5
3	0x777	0x3333 3333 3333 3333	14
4	0x333	0x4444 4444 4444 4444	6
5	0x088	0x5555 5555 5555 5555	1
6	0x4CC	0x6666 6666 6666 6666	9
7	0x199	0x7777 7777 7777 7777	3
8	0x7FF	0x8888 8888 8888 8888	15
9	0x111	0x9999 9999 9999 9999	2
10	0x444	0xAAAA AAAA AAAA AAAA	8
11	0x555	0xBBBB BBBB BBBB BBBB	10
12	0x6EE	0xCCCC CCCC CCCC CCCC	13
13	0x3BB	0xDDDD DDDD DDDD DDDD	7
14	0x222	0xEEEE EEEE EEEE EEEE	4
15	0x5DD	0xFFFF FFFF FFFF FFFF	11

- メールボックス 1~ 15 を一括送信します。
- 送信中は送信完了フラグをポーリングします。
- 送信完了フラグセットを確認後、送信完了フラグをクリアし終了します。

受信側仕様

- メールボックス 0 を使用します。
- Identifier をマスクせず全て受信します。
- 受信メッセージ割り込み (IRR1) を使用します。
 - (a) 受信データは受信メッセージ割り込みで DTC を起動し、内蔵 RAM に格納します。
 - (b) DTC 転送はブロック転送モードを使用し、8 バイトを 1 ブロックとして 15 ブロック転送します。
 - (c) DTC 転送終了後の受信メッセージ割り込みルーチン内で受信完了フラグをクリアと受信メッセージ割り込みを禁止し、終了します。

4.2 送受信仕様機能説明

- 表 4.2~4.3 に関連レジスタの機能割り付けを示します。
 (使用端子及びポートレジスタは例題 1 参照)

表 4.2 HCAN2 機能割り付け

関連レジスタ		機能	
送受信共通 レジスタ	MCR	マスタコントロールレジスタ HCAN2 の動作を制御します。	
	BCR0	ビットタイミングコンフィギュレーションレジスタ CAN のポーレートプリスケール、ビットタイミングパラメータを 設定します。	
	BCR1		
	IRR	インタラプトリクエストレジスタ 各割り込み要因のステータスを示します。	
	MBx	MC0	メッセージコントロールレジスタ Identifier 設定、データフレームカリモートフレームの設定を 行ないます。
		MC4	メッセージコントロールレジスタ 送信/受信の設定を行います。
		MC5	メッセージコントロールレジスタ データ長の設定を行います。
MD7 ~ MD14		メッセージデータレジスタ 送受信された CAN のメッセージデータを格納します。	
LAFM15	ローカルアクセプタンスフィルタマスク 受信用メールボックスの Identifier 用フィルタマスクの設定を 行ないます。		
送信用 レジスタ	TXPR	送信待ちレジスタ 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	送信アクノレッジレジスタ 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	RXPR	受信完了レジスタ 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。	
割り込み関連 レジスタ	MBIMR	メールボックスインタラプトマスクレジスタ 各メールボックスの割り込み要求をイネーブルにします。	
	IMR	インタラプトマスクレジスタ IRR の割り込みフラグによる割り込み要求をイネーブルにします。	
	IPRK	インタラプトプライオリティレジスタ HCAN2 の割り込み要求の優先順位を設定します。	

表 4.3 DTC 機能割り付け

関連レジスタ	機能
DTMR	DTC モードレジスタ DTC の動作モードの制御を行います。
DTSAR	DTC ソースアドレスレジスタ DTC の転送するデータの転送元アドレスを指定します。
DTDAR	DTC デスティネーションアドレスレジスタ DTC の転送するデータの転送先アドレスを指定します。
DTCRA	DTC 転送カウントレジスタ A DTC のデータ転送の転送回数を指定します。
DTCRB	DTC 転送カウントレジスタ B ブロック転送モードのとき、ブロック長を指定します。
DTBR	DTC 情報ベースレジスタ DTC 転送情報を格納するメモリアドレスの上位 16 ビットを指定します。
DTEF	DTC イネーブルレジスタ DTC を起動する割り込み要因 (HCAN2 の RM1) を選択します。

4.3 送信側フロー

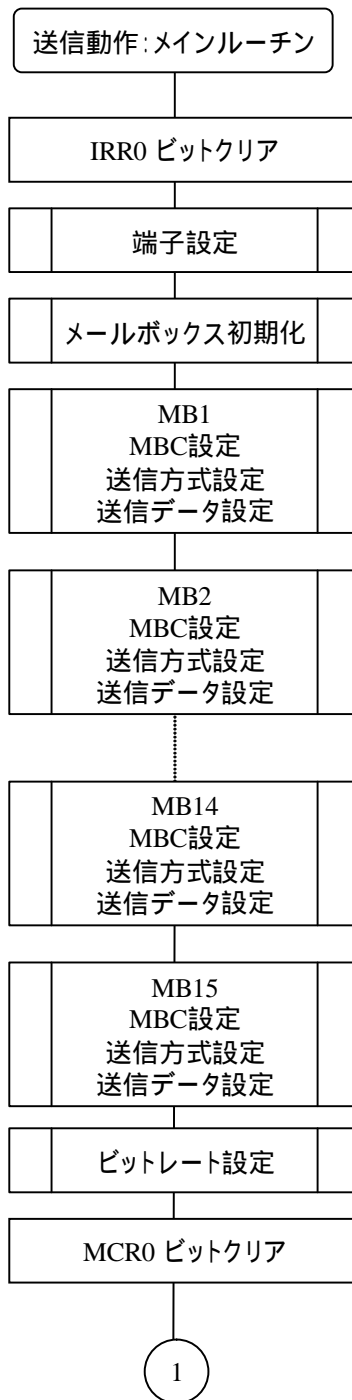


図 4.1 送信側フローチャート(1)

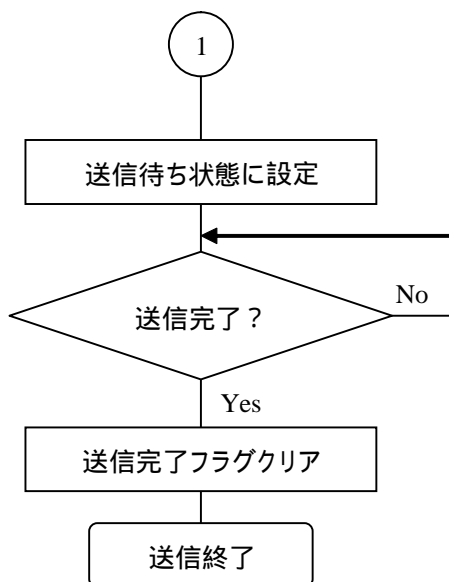


図 4.2 送信側フロ-チャート(2)

4.4 ソフトウェア説明(送信側)

- 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	t_main	HCAN2 の初期設定及び送信の設定を行います。

- 使用レジスタの説明(使用端子及びポートレジスタは例題 1 参照)

レジスタ名	機能	設定値	モジュール名
work	MB 初期化に使用するワークレジスタ。		メインルーチン
HCAN_MCR	リセットリクエストビットのクリア。また、メールボックスの Identifier 優先順位に従って送信に設定。	0x0000	
HCAN_IRR	リセット/ホールド/スリープ割込みフラグクリア。 (クリア条件: 1 ライト)	0x0001	
HCAN_BCRO	ビットレートを =50MHz 時 250Kbps に設定。	0x0009	
HCAN_BCR1		0x4300	
HCAN_MBx.MC0	MB1 ~ 15 をデータフレーム、スタンダードフォーマットに設定。また、Identifier を設定。(設定値は表 4.1 を参照)		
HCAN_MBx.MC4	MB1 ~ 15 を送信用に設定。	0x01	
HCAN_MBx.MC5	MB1 ~ 15 の送信サイズを 8 バイトのデータ長に設定。	0x08	
HCAN_MBx.MD7 ~ 14	MB1 ~ 15 の送信データを設定。 (設定値は表 4.1 を参照)		
HCAN_TXPRO	MB1 ~ 15 を送信待ち状態に設定。	0xFFFFE	
HCAN_TXACK0	MB1 ~ 15 の送信完了フラグのチェックとクリア。 (クリア条件: 1 ライト)	0xFFFFE	

4.5 送信プログラム

```

/*****
/* Filename      : t_main.c                               */
/* Written       : '06/07/24 Ver.4.0                       */
/* Purpose       : HCAN2 for SH7047                         */
/*****
/*****
/* HCAN2 送信プログラム(例題 4)                             */
/*****
#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 "SH7047.h"        /* 周辺レジスタ定義ヘッダ ファイル */
/*****
/* 関数プロトタイプ宣言                                     */
/*****
void t_main(void);
/*****
/* メインルーチン                                         */
/*****
void t_main(void){
    unsigned short *work;

/* IRRO ビットクリア */
    HCAN_IRR = 0x0001;          /* リセット割り込みフラグのクリア(クリア条件:1バイト)*/
/* 端子設定 */
    PB.PBCR1 = 0x0000;         /* PB HCAN2 設定 */
    PB.PBCR2 = 0x000F;         /* PB HCAN2 設定 */
/* メールボックス初期化 */
    work = (unsigned short *)0xFFFFB100;
    do {
        *work = 0x0000;
        work++;
    } while(work < (unsigned short *)0xFFFFB4F4);
/* 送信データ設定 */
/****  MB1  ****/
    HCAN_MB1.MC4 = 0x01;       /* メールボックス1を送信用に設定 */
    HCAN_MB1.MC0 = 0x6660;     /* データフレーム、スタンダートフォーマットの選択、Identifier の設定 */
    HCAN_MB1.MC5 = 0x08;       /* データ長: 8 バイト設定 */
    HCAN_MB1.MD7 = 0x11;       /* 送信データ: 00010001 設定 */
    HCAN_MB1.MD8 = 0x11;       /* 送信データ: 00010001 設定 */
    HCAN_MB1.MD9 = 0x11;       /* 送信データ: 00010001 設定 */
    HCAN_MB1.MD10 = 0x11;      /* 送信データ: 00010001 設定 */
    HCAN_MB1.MD11 = 0x11;      /* 送信データ: 00010001 設定 */
    HCAN_MB1.MD12 = 0x11;      /* 送信データ: 00010001 設定 */
    HCAN_MB1.MD13 = 0x11;      /* 送信データ: 00010001 設定 */
    HCAN_MB1.MD14 = 0x11;      /* 送信データ: 00010001 設定 */

/****  MB2  ****/
    HCAN_MB2.MC4 = 0x01;       /* メールボックス2を送信用に設定 */
    HCAN_MB2.MC0 = 0x2AA0;     /* データフレーム、スタンダートフォーマットの選択、Identifier の設定 */
    HCAN_MB2.MC5 = 0x08;       /* データ長: 8 バイト設定 */
    HCAN_MB2.MD7 = 0x22;       /* 送信データ: 00100010 設定 */
    HCAN_MB2.MD8 = 0x22;       /* 送信データ: 00100010 設定 */
    HCAN_MB2.MD9 = 0x22;       /* 送信データ: 00100010 設定 */
    HCAN_MB2.MD10 = 0x22;      /* 送信データ: 00100010 設定 */
    HCAN_MB2.MD11 = 0x22;      /* 送信データ: 00100010 設定 */
    HCAN_MB2.MD12 = 0x22;      /* 送信データ: 00100010 設定 */
    HCAN_MB2.MD13 = 0x22;      /* 送信データ: 00100010 設定 */
    HCAN_MB2.MD14 = 0x22;      /* 送信データ: 00100010 設定 */

```



```

/**** MB3 ****/
HCAN_MB3.MC4 = 0x01; /* メーブルックス3を送信用に設定 */
HCAN_MB3.MC0 = 0x7770; /* 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_MB3.MC5 = 0x08; /* データ長：8ビット設定 */
HCAN_MB3.MD7 = 0x33; /* 送信データ：00110011設定 */
HCAN_MB3.MD8 = 0x33; /* 送信データ：00110011設定 */
HCAN_MB3.MD9 = 0x33; /* 送信データ：00110011設定 */
HCAN_MB3.MD10 = 0x33; /* 送信データ：00110011設定 */
HCAN_MB3.MD11 = 0x33; /* 送信データ：00110011設定 */
HCAN_MB3.MD12 = 0x33; /* 送信データ：00110011設定 */
HCAN_MB3.MD13 = 0x33; /* 送信データ：00110011設定 */
HCAN_MB3.MD14 = 0x33; /* 送信データ：00110011設定 */

/**** MB4 ****/
HCAN_MB4.MC4 = 0x01; /* メーブルックス4を送信用に設定 */
HCAN_MB4.MC0 = 0x3330; /* 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_MB4.MC5 = 0x08; /* データ長：8ビット設定 */
HCAN_MB4.MD7 = 0x44; /* 送信データ：01000100設定 */
HCAN_MB4.MD8 = 0x44; /* 送信データ：01000100設定 */
HCAN_MB4.MD9 = 0x44; /* 送信データ：01000100設定 */
HCAN_MB4.MD10 = 0x44; /* 送信データ：01000100設定 */
HCAN_MB4.MD11 = 0x44; /* 送信データ：01000100設定 */
HCAN_MB4.MD12 = 0x44; /* 送信データ：01000100設定 */
HCAN_MB4.MD13 = 0x44; /* 送信データ：01000100設定 */
HCAN_MB4.MD14 = 0x44; /* 送信データ：01000100設定 */

/**** MB5 ****/
HCAN_MB5.MC4 = 0x01; /* メーブルックス5を送信用に設定 */
HCAN_MB5.MC0 = 0x0880; /* 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_MB5.MC5 = 0x08; /* データ長：8ビット設定 */
HCAN_MB5.MD7 = 0x55; /* 送信データ：01010101設定 */
HCAN_MB5.MD8 = 0x55; /* 送信データ：01010101設定 */
HCAN_MB5.MD9 = 0x55; /* 送信データ：01010101設定 */
HCAN_MB5.MD10 = 0x55; /* 送信データ：01010101設定 */
HCAN_MB5.MD11 = 0x55; /* 送信データ：01010101設定 */
HCAN_MB5.MD12 = 0x55; /* 送信データ：01010101設定 */
HCAN_MB5.MD13 = 0x55; /* 送信データ：01010101設定 */
HCAN_MB5.MD14 = 0x55; /* 送信データ：01010101設定 */

/**** MB6 ****/
HCAN_MB6.MC4 = 0x01; /* メーブルックス6を送信用に設定 */
HCAN_MB6.MC0 = 0x4CC0; /* 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_MB6.MC5 = 0x08; /* データ長：8ビット設定 */
HCAN_MB6.MD7 = 0x66; /* 送信データ：01100110設定 */
HCAN_MB6.MD8 = 0x66; /* 送信データ：01100110設定 */
HCAN_MB6.MD9 = 0x66; /* 送信データ：01100110設定 */
HCAN_MB6.MD10 = 0x66; /* 送信データ：01100110設定 */
HCAN_MB6.MD11 = 0x66; /* 送信データ：01100110設定 */
HCAN_MB6.MD12 = 0x66; /* 送信データ：01100110設定 */
HCAN_MB6.MD13 = 0x66; /* 送信データ：01100110設定 */
HCAN_MB6.MD14 = 0x66; /* 送信データ：01100110設定 */

/**** MB7 ****/
HCAN_MB7.MC4 = 0x01; /* メーブルックス7を送信用に設定 */
HCAN_MB7.MC0 = 0x1990; /* 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_MB7.MC5 = 0x08; /* データ長：8ビット設定 */
HCAN_MB7.MD7 = 0x77; /* 送信データ：01110111設定 */
HCAN_MB7.MD8 = 0x77; /* 送信データ：01110111設定 */
HCAN_MB7.MD9 = 0x77; /* 送信データ：01110111設定 */
HCAN_MB7.MD10 = 0x77; /* 送信データ：01110111設定 */
HCAN_MB7.MD11 = 0x77; /* 送信データ：01110111設定 */
HCAN_MB7.MD12 = 0x77; /* 送信データ：01110111設定 */
HCAN_MB7.MD13 = 0x77; /* 送信データ：01110111設定 */
HCAN_MB7.MD14 = 0x77; /* 送信データ：01110111設定 */
    
```

```

/**** MB8 ****/
HCAN_MB8.MC4 = 0x01; /* メッセージ 8 を送信用に設定 */
HCAN_MB8.MC0 = 0x7FF0; /* データレム、スタンダートフォーマットの選択、Identifier の設定 */
HCAN_MB8.MC5 = 0x08; /* データ長：8 バイト設定 */
HCAN_MB8.MD7 = 0x88; /* 送信データ：10001000 設定 */
HCAN_MB8.MD8 = 0x88; /* 送信データ：10001000 設定 */
HCAN_MB8.MD9 = 0x88; /* 送信データ：10001000 設定 */
HCAN_MB8.MD10 = 0x88; /* 送信データ：10001000 設定 */
HCAN_MB8.MD11 = 0x88; /* 送信データ：10001000 設定 */
HCAN_MB8.MD12 = 0x88; /* 送信データ：10001000 設定 */
HCAN_MB8.MD13 = 0x88; /* 送信データ：10001000 設定 */
HCAN_MB8.MD14 = 0x88; /* 送信データ：10001000 設定 */

/**** MB9 ****/
HCAN_MB9.MC4 = 0x01; /* メッセージ 9 を送信用に設定 */
HCAN_MB9.MC0 = 0x1110; /* データレム、スタンダートフォーマットの選択、Identifier の設定 */
HCAN_MB9.MC5 = 0x08; /* データ長：8 バイト設定 */
HCAN_MB9.MD7 = 0x99; /* 送信データ：10011001 設定 */
HCAN_MB9.MD8 = 0x99; /* 送信データ：10011001 設定 */
HCAN_MB9.MD9 = 0x99; /* 送信データ：10011001 設定 */
HCAN_MB9.MD10 = 0x99; /* 送信データ：10011001 設定 */
HCAN_MB9.MD11 = 0x99; /* 送信データ：10011001 設定 */
HCAN_MB9.MD12 = 0x99; /* 送信データ：10011001 設定 */
HCAN_MB9.MD13 = 0x99; /* 送信データ：10011001 設定 */
HCAN_MB9.MD14 = 0x99; /* 送信データ：10011001 設定 */

/**** MB10 ****/
HCAN_MB10.MC4 = 0x01; /* メッセージ 10 を送信用に設定 */
HCAN_MB10.MC0 = 0x4440; /* データレム、スタンダートフォーマットの選択、Identifier の設定 */
HCAN_MB10.MC5 = 0x08; /* データ長：8 バイト設定 */
HCAN_MB10.MD7 = 0xAA; /* 送信データ：10101010 設定 */
HCAN_MB10.MD8 = 0xAA; /* 送信データ：10101010 設定 */
HCAN_MB10.MD9 = 0xAA; /* 送信データ：10101010 設定 */
HCAN_MB10.MD10 = 0xAA; /* 送信データ：10101010 設定 */
HCAN_MB10.MD11 = 0xAA; /* 送信データ：10101010 設定 */
HCAN_MB10.MD12 = 0xAA; /* 送信データ：10101010 設定 */
HCAN_MB10.MD13 = 0xAA; /* 送信データ：10101010 設定 */
HCAN_MB10.MD14 = 0xAA; /* 送信データ：10101010 設定 */

/**** MB11 ****/
HCAN_MB11.MC4 = 0x01; /* メッセージ 11 を送信用に設定 */
HCAN_MB11.MC0 = 0x5550; /* データレム、スタンダートフォーマットの選択、Identifier の設定 */
HCAN_MB11.MC5 = 0x08; /* データ長：8 バイト設定 */
HCAN_MB11.MD7 = 0xBB; /* 送信データ：10111011 設定 */
HCAN_MB11.MD8 = 0xBB; /* 送信データ：10111011 設定 */
HCAN_MB11.MD9 = 0xBB; /* 送信データ：10111011 設定 */
HCAN_MB11.MD10 = 0xBB; /* 送信データ：10111011 設定 */
HCAN_MB11.MD11 = 0xBB; /* 送信データ：10111011 設定 */
HCAN_MB11.MD12 = 0xBB; /* 送信データ：10111011 設定 */
HCAN_MB11.MD13 = 0xBB; /* 送信データ：10111011 設定 */
HCAN_MB11.MD14 = 0xBB; /* 送信データ：10111011 設定 */

/**** MB12 ****/
HCAN_MB12.MC4 = 0x01; /* メッセージ 12 を送信用に設定 */
HCAN_MB12.MC0 = 0x6EE0; /* データレム、スタンダートフォーマットの選択、Identifier の設定 */
HCAN_MB12.MC5 = 0x08; /* データ長：8 バイト設定 */
HCAN_MB12.MD7 = 0xCC; /* 送信データ：11001100 設定 */
HCAN_MB12.MD8 = 0xCC; /* 送信データ：11001100 設定 */
HCAN_MB12.MD9 = 0xCC; /* 送信データ：11001100 設定 */
HCAN_MB12.MD10 = 0xCC; /* 送信データ：11001100 設定 */
HCAN_MB12.MD11 = 0xCC; /* 送信データ：11001100 設定 */
HCAN_MB12.MD12 = 0xCC; /* 送信データ：11001100 設定 */
HCAN_MB12.MD13 = 0xCC; /* 送信データ：11001100 設定 */
HCAN_MB12.MD14 = 0xCC; /* 送信データ：11001100 設定 */
    
```

```

/**** MB13 ****/
HCAN_MB13.MC4 = 0x01; /* メッセージ 13 を送信用に設定 */
HCAN_MB13.MC0 = 0x3BB0; /* 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_MB13.MC5 = 0x08; /* データ長：8 バイト設定 */
HCAN_MB13.MD7 = 0xDD; /* 送信データ：11011101 設定 */
HCAN_MB13.MD8 = 0xDD; /* 送信データ：11011101 設定 */
HCAN_MB13.MD9 = 0xDD; /* 送信データ：11011101 設定 */
HCAN_MB13.MD10 = 0xDD; /* 送信データ：11011101 設定 */
HCAN_MB13.MD11 = 0xDD; /* 送信データ：11011101 設定 */
HCAN_MB13.MD12 = 0xDD; /* 送信データ：11011101 設定 */
HCAN_MB13.MD13 = 0xDD; /* 送信データ：11011101 設定 */
HCAN_MB13.MD14 = 0xDD; /* 送信データ：11011101 設定 */

/**** MB14 ****/
HCAN_MB14.MC4 = 0x01; /* メッセージ 14 を送信用に設定 */
HCAN_MB14.MC0 = 0x2220; /* 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_MB14.MC5 = 0x08; /* データ長：8 バイト設定 */
HCAN_MB14.MD7 = 0xEE; /* 送信データ：11101110 設定 */
HCAN_MB14.MD8 = 0xEE; /* 送信データ：11101110 設定 */
HCAN_MB14.MD9 = 0xEE; /* 送信データ：11101110 設定 */
HCAN_MB14.MD10 = 0xEE; /* 送信データ：11101110 設定 */
HCAN_MB14.MD11 = 0xEE; /* 送信データ：11101110 設定 */
HCAN_MB14.MD12 = 0xEE; /* 送信データ：11101110 設定 */
HCAN_MB14.MD13 = 0xEE; /* 送信データ：11101110 設定 */
HCAN_MB14.MD14 = 0xEE; /* 送信データ：11101110 設定 */

/**** MB15 ****/
HCAN_MB15.MC4 = 0x01; /* メッセージ 15 を送信用に設定 */
HCAN_MB15.MC0 = 0x5DD0; /* 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_MB15.MC5 = 0x08; /* データ長：8 バイト設定 */
HCAN_MB15.MD7 = 0xFF; /* 送信データ：11111111 設定 */
HCAN_MB15.MD8 = 0xFF; /* 送信データ：11111111 設定 */
HCAN_MB15.MD9 = 0xFF; /* 送信データ：11111111 設定 */
HCAN_MB15.MD10 = 0xFF; /* 送信データ：11111111 設定 */
HCAN_MB15.MD11 = 0xFF; /* 送信データ：11111111 設定 */
HCAN_MB15.MD12 = 0xFF; /* 送信データ：11111111 設定 */
HCAN_MB15.MD13 = 0xFF; /* 送信データ：11111111 設定 */
HCAN_MB15.MD14 = 0xFF; /* 送信データ：11111111 設定 */

/* ビットレート(BCR)設定： =50MHz 時ビットレート 250Kbps */
HCAN_BCRO = 0x0009; /* BRP=9(10 システムクロック) */
HCAN_BCR1 = 0x4300; /* TSEG1=4(5tq),TSEG2=3(4tq) */

/* MCRO ビットクリア */
HCAN_MCR = 0x0000; /* リセットマスクビットのクリア、Identifier 優先順位送信 */

/* メッセージの送信待ち状態に設定 */
HCAN_TXPRO = 0xFFFF; /* メッセージ 1 ~ 15 を送信待ち状態に設定 */

/* 送信完了待ち */
while((HCAN_TXACK0 & 0xFFFF) != 0xFFFF);

/* 送信完了フラグのクリア */
HCAN_TXACK0 = 0xFFFF; /* 送信完了フラグのクリア(クリア条件：1 ライト) */
while(1);

}
    
```

4.6 受信側フロー



図 4.3 受信側フローチャート(1)

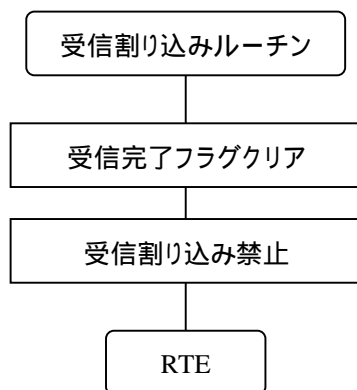


図 4.4 受信側フローチャート(2)

4.7 ソフトウェア説明(受信側)

- 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	r_main	HCAN2 の初期設定及び受信の設定を行います。
受信割り込みルーチン	RM1_IRR1	受信完了フラグクリアと割り込み禁止の設定を行います。

- 使用レジスタの説明(使用端子及びポートレジスタは例題 1 参照)

レジスタ名	機能	設定値	モジュール名
MBbuff	受信データ格納先頭アドレス。(H FFFFD100)		メインルーチン
work	MB 初期化に使用するワークレジスタ。		
DTMR	DTSAR, DTDAR 共に転送後インクリメント、ブロック転送モード、バイト転送に設定。	0xA890	
DTSAR	転送元アドレスに MB0 のメッセージデータ領域を設定。	0xFFFFB108	
DTDAR	転送先アドレスに受信データ格納エリアを設定。	0xFFFFD100	
DTCRA	ブロック転送回数を設定。(15 回)	0x000F	
DTCRB	ブロック長を設定。(8 バイト)	0x0008	
DTC.DTBR	DTC 転送情報を格納するメモリアドレスの上位 16 ビットを指定。	0xFFFF	
DTC.DTEF	DTC を起動する割り込み要因(HCAN2 の RM1)を選択。	0x04	
HCAN_MCR	リセットリクエストビットクリア。	0x0000	
HCAN_IRR	リセット/ホールド/スリープ割り込みフラグクリア。 (クリア条件: 1 ライト)	0x0001	
HCAN_BCRO	ビットレートを =50MHz 時 250Kbps に設定。	0x0009	
HCAN_BCR1		0x4300	
HCAN_MB0.MC0	MB0 をデータフレーム、スタンダードフォーマットに設定。	0x0000	
HCAN_MB0.MC4	MB0 を受信用に設定。	0x02	
HCAN_MB0.LAFM15	MB0 の Identifier 用フィルタマスクを設定。	0x7FF0	
HCAN_IMR	受信メッセージ割り込み許可。	0xFFFFD	
HCAN_MBIMRO	MB0 の割り込み要求を許可。	0xFFFFE	
INTC.IPRK	HCAN2 の割り込み要求の優先順位を設定。	0x00F0	
HCAN_RXPRO	MB0 受信完了フラグのクリア。 (クリア条件: 1 ライト)	0x0001	
HCAN_IMR	受信メッセージ割り込み禁止。	0xFFFF	

4.8 受信プログラム

```

/*****/
/* Filename      :  r_main.c                               */
/* Written       :  '06/07/24 Ver.4.0                     */
/* Purpose       :  HCAN2 for SH7047                       */
/*****/
/*****/
/* HCAN2 受信プログラム(例題 4)                           */
/*****/
#include <stdio.h>          /*ライブラリ関数用ヘッダファイル */
#include <machine.h>        /*ライブラリ関数用ヘッダファイル */
#include "SH7047.h"         /*周辺レジスタ定義ヘッダファイル */
/*****/
/* 関数プロトタイプ宣言                                    */
/*****/
void r_main(void);
void RM1_IRR1(void);
/*****/
/* 定数定義                                              */
/*****/
#define DTMR (* (unsigned short *)0xFFFFD080) /* DTC レジスタ情報 */
#define DTCRA (* (unsigned short *)0xFFFFD082) /* DTC レジスタ情報 */
#define DTCRB (* (unsigned short *)0xFFFFD086) /* DTC レジスタ情報 */
#define DTSAR (* (unsigned long *)0xFFFFD088) /* DTC レジスタ情報 */
#define DTDAR (* (unsigned long *)0xFFFFD08C) /* DTC レジスタ情報 */
#define MBuffer (* (unsigned char *)0xFFFFD100) /* 受信データ格納エリア */
/*****/
/* メインルーチン                                        */
/*****/
void r_main(void){
    unsigned short *work;

/* 受信データ格納エリア初期化 */
    work = (unsigned short *)0xFFFFD100;
    do {
        *work = 0x0000;
        work++;
    } while(work < (unsigned short *)0xFFFFD180);
/* DTC 設定 */
    DTMR = 0xA890; /* DTSAR,DTDAR 共に転送後インクリメント,ブロック転送モード,ビット転送 */
    DTSAR = (unsigned long)&HCAN_MBO.MD7; /* 転送元アドレス設定 */
    DTDAR = (unsigned long)&MBuffer; /* 転送先アドレス設定 */
    DTCRA = 0x000F; /* ブロック転送 15 回 */
    DTCRB = 0x0008; /* ブロック長 8 ビット */
    DTC.DTBR = 0xFFFF; /* レジスタ情報先頭アドレス */
    DTC.DTEF |= 0x04; /* CAN2 割込み(RM1) */
/* IRRO ビットクリア */
    HCAN_IRR = 0x0001; /* レジスタ割り込みフラグのクリア(クリア条件:1 ライト) */
/* 端子設定 */
    PB.PBCR1 = 0x0000; /* PB HCAN2 設定 */
    PB.PBCR2 = 0x000F; /* PB HCAN2 設定 */
/* メールボックス初期化 */
    work = (unsigned short *)0xFFFFB100;
    do {
        *work = 0x0000;
        work++;
    } while(work < (unsigned short *)0xFFFFB4F4);
/* MBC 設定 */
    HCAN_MBO.MC4 = 0x02; /* メールボックス 0 を受信用に設定 */
/* 割り込みの設定 */
    HCAN_IMR = 0xFFFF; /* 受信メッセージ割り込み許可 */
    HCAN_MBIMRO = 0xFFFE; /* メールボックス 0 受信割り込み要求許可 */
    INTC.IPRK = 0x00F0; /* RM1 優先順位設定 */
/* 受信方式設定 */
    HCAN_MBO.MCO = 0x0000; /* データレーム、スタンダードフォーマットの選択、Identifier の設定 */

```

```

/* ローカルアクセプタンスフィルタの設定 */
HCAN_MB0.LAFM15 = 0x7FF0; /* メルプックス0のIdentifier用のフィルタマスクの設定 */
/* ビットレート(BCR)設定: =50MHz時ビットレート 250Kbps */
HCAN_BCRO = 0x0009; /* BRP=9(10システムクロック) */
HCAN_BCR1 = 0x4300; /* TSEG1=4(5tq),TSEG2=3(4tq)*/
/* MCRO ビットクリア */
HCAN_MCR = 0x0000; /* リセットリクエストビットのクリア */

set_imask(0);
while(1);
}

/*****
/* 受信割り込みルーチン */
/*****
#pragma interrupt(RM1_IRR1)
void RM1_IRR1(void){

/* 受信完了レジスタクリア */
HCAN_RXPRO = 0x0001; /* 受信完了フラグのクリア(クリア条件:1バイト) */
/* 受信割り込み禁止 */
HCAN_IMR = 0xFFFF; /* 受信メッセージ割り込み禁止 */

}
    
```


4.9 動作波形(送受信)

本アプリケーション動作時の先頭波形及び最終波形を図 4.5 ~ 図 4.6 に示します。

先頭波形

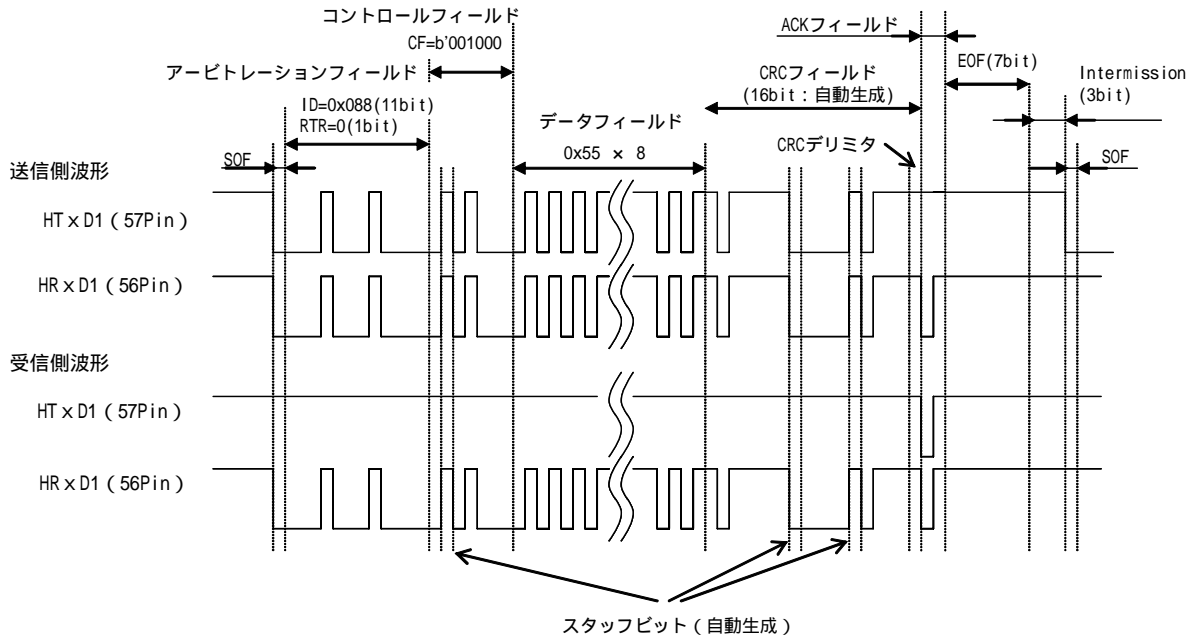


図 4.5 動作波形(先頭波形)

最終波形

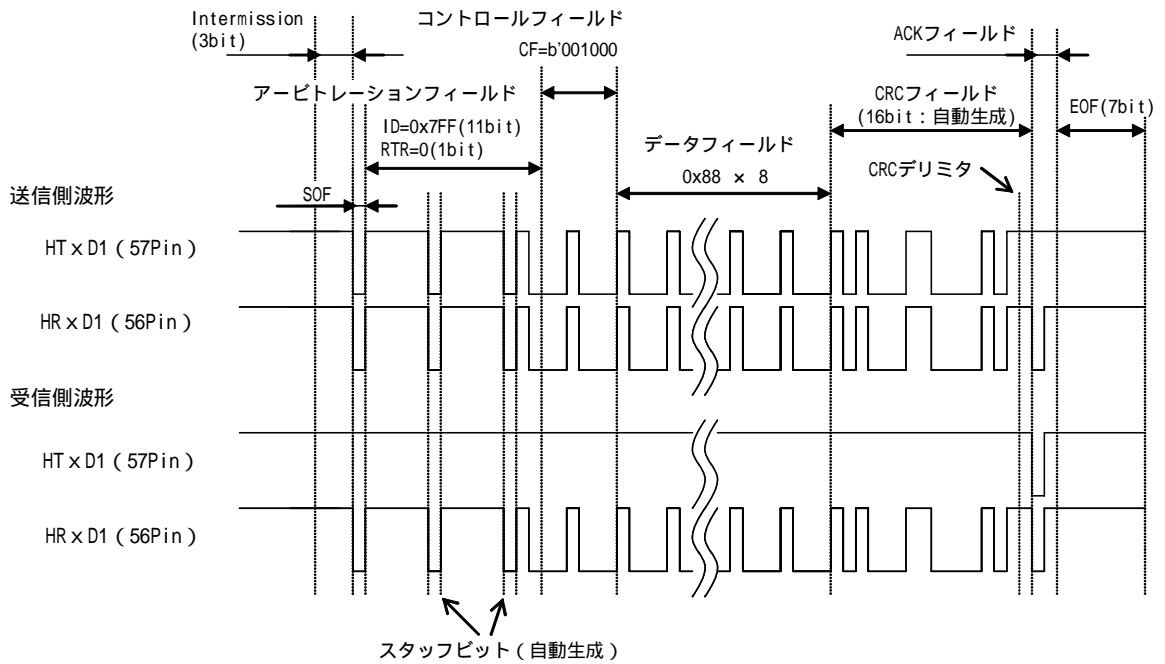


図 4.6 動作波形(最終波形)

5. 例題5(リモートフレーム)

5.1 送受信仕様

SH7047F を 2 個使用した、スタンダードフォーマットによるリモートフレーム送受信。

送受信共通仕様

- 通信速度は 250Kbps(50MHz 動作時)とします。
- Identifier は H 555 とします。
- Identifier をマスクして一致すれば受信します。

リモートフレーム送信側仕様

- メールボックス 1 をリモートフレーム送信及びデータフレーム受信に使用します。
- データ長は 8 バイトを要求します。
- 受信メッセージ割り込み (IRR1) を使用します。
 - (a) リモートフレーム送信後、受信メッセージ割り込み待ち状態にします。
 - (b) 受信メッセージ割り込みルーチン内で受信完了フラグをクリアと受信メッセージ割り込みを禁止します。
 - (c) 受信データは受信メッセージ割り込みルーチン内で DTC を設定及びソフトウェア起動して、内蔵 RAM に格納します。
 - (d) DTC 転送はブロック転送モードを使用し、8 バイトを 1 ブロックとして 1 ブロック転送します。
- DTC 転送終了割り込みを使用します。
 - DTC 転送終了割り込みルーチン内で DTC のソフトウェア起動を禁止し、終了します。

リモートフレーム受信側仕様

- メールボックス 1 をリモートフレーム受信及びデータフレーム送信に使用します。
- データフレームを予め準備し、自動送信機能 (ATX) を使用し送信します。
- 送信データは H 55、H 66、H 77、H 88、H 99、H AA、H BB、H FF とします。
- リモートフレームリクエスト割り込み (IRR2) を使用します。
 - (a) リモートフレームリクエスト割り込みルーチン内でリモートフレーム受信完了フラグをクリア及び受信割り込みを禁止し、データフレーム送信完了フラグをポーリングします。
 - (b) 送信完了フラグセットを確認後、送信完了フラグをクリアし終了します。

5.2 送受信仕様機能説明

- 表 5.1~5.2 に関連レジスタの機能割り付けを示します。
 (使用端子及びポートレジスタは例題 1 参照)

表 5.1 HCAN2 機能割り付け

関連レジスタ		機能	
送受信共通 レジスタ	MCR	マスタコントロールレジスタ HCAN2 の動作を制御します。	
	BCR0	ビットタイミングコンフィギュレーションレジスタ CAN のポ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。	
	BCR1		
	IRR	インタラプトリクエストレジスタ 各割り込み要因のステータスを示します。	
	MBx	MC0	メッセージコントロールレジスタ Identifier 設定、データフレームかりリモートフレームの設定を 行ないます。
		MC4	メッセージコントロールレジスタ 送信/受信の設定を行います。
		MC5	メッセージコントロールレジスタ データ長の設定を行います。
		MD7 ~ MD14	メッセージデータレジスタ 送受信された CAN のメッセージデータを格納します。
LAFM15		ローカルアクセプタンスフィルタマスク 受信用メールボックスの Identifier 用フィルタマスクの設定を 行ないます。	
送信用 レジスタ	TXPR	送信待ちレジスタ 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	送信アクノレッジレジスタ 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	RXPR	受信完了レジスタ 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	リモートリクエストレジスタ 対応するメールボックスがリモートフレームを受信したことを示します。	
割り込み関連 レジスタ	MBIMR	メールボックスインタラプトマスクレジスタ 各メールボックスの割り込み要求をイネーブルにします。	
	IMR	インタラプトマスクレジスタ IRR の割り込みフラグによる割り込み要求をイネーブルにします。	
	IPRK	インタラプトプライオリティレジスタ HCAN2 の割り込み要求の優先順位を設定します。	

表 5.2 DTC 機能割り付け

関連レジスタ	機能
DTMR	DTC モードレジスタ DTC の動作モードの制御を行います。
DTSAR	DTC ソースアドレスレジスタ DTC の転送するデータの転送元アドレスを指定します。
DTDAR	DTC デスティネーションアドレスレジスタ DTC の転送するデータの転送先アドレスを指定します。
DTCRA	DTC 転送カウントレジスタ A DTC のデータ転送の転送回数を指定します。
DTCRB	DTC 転送カウントレジスタ B ブロック転送モードのとき、ブロック長を指定します。
DTBR	DTC 情報ベースレジスタ DTC 転送情報を格納するメモリアドレスの上位 16 ビットを指定します。
DTCSR	DTC コントロール/ステータスレジスタ ソフトウェアによる DTC 起動の許可/禁止の設定、及びソフトウェア起動による DTC ベクタアドレスを設定します。
IPRG	インタラプトプライオリティレジスタ DTC の割り込み要求の優先順位を設定します。

5.3 送信側フロー



図 5.1 送信側フローチャート(1)



図 5.2 送信側フローチャート(2)

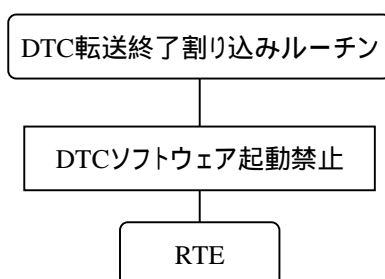


図 5.3 送信側フローチャート(3)

5.4 ソフトウェア説明(送信側)

- 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	t_main	HCAN2 の初期設定及び送信の設定を行います。
受信割り込みルーチン	RM1_IRR1	受信完了フラグクリアと割り込み禁止の設定、及び DTC 起動により受信データを格納します。
DTC 転送終了割り込みルーチン	DTC_SWDTEND	ソフトウェアによる DTC 起動を禁止に設定します。

- 使用レジスタの説明(使用端子及びポートレジスタは例題 1 参照)

レジスタ名	機能	設定値	モジュール名
MBbuff	受信データ格納エリア。(H FFFFD100~H FFFFD108)		メインルーチン
work	MB 初期化に使用するワークレジスタ。		
HCAN_MCR	リセットリクエストビットのクリア。	0x0000	
HCAN_IRR	リセット/ホールド/スリープ割り込みフラグクリア。 (クリア条件: 1 ライト)	0x0001	
HCAN_BCR0	ビットレートを =50MHz 時 250Kbps に設定。	0x0009	
HCAN_BCR1		0x4300	
HCAN_MB1.MC0	MB1 をリモートフレーム、スタンダードフォーマットに設定。また、Identifier(H 555)を設定。	0x5558	
HCAN_MB1.MC4	MB1 をリモートフレーム送信可、データフレーム受信可に設定。	0x05	
HCAN_MB1.MC5	MB1 の送信サイズを 8 バイトのデータ長に設定。	0x08	
HCAN_MB1.LAFM15	MB1 の Identifier 用フィルタマスクを設定。	0x0000	
HCAN_IMR	受信メッセージ割り込み許可。	0xFFFF	
HCAN_MB1MRO	MB1 の割り込み要求を許可。	0xFFFF	
INTC.IPRK	HCAN2 の割り込み要求の優先順位を設定。	0x00F0	
INTC.IPRG	DTC の割り込み要求の優先順位を設定します。	0x0E00	
HCAN_TXPRO	MB1 を送信待ち状態に設定。	0x0002	
HCAN_TXACK0	MB1 の送信完了フラグのチェックとクリア。 (クリア条件: 1 ライト)	0x0002	受信割り込みルーチン
HCAN_RXPRO	MB1 の受信完了フラグのクリア。 (クリア条件: 1 ライト)	0x0002	
HCAN_IMR	受信メッセージ割り込み禁止。	0xFFFF	
DTMR	DTSAR, DTDAR 共に転送後インクリメント、ブロック転送モード、バイト転送に設定。	0xA890	
DTSAR	転送元アドレスに MB1 のメッセージデータ領域を設定。	0xFFFFB128	
DTDAR	転送先アドレスに受信データ格納エリアを設定。	0xFFFFD100	
DTCRA	ブロック転送回数を設定。(1 回)	0x0001	
DTCRB	ブロック長を設定。(8 バイト)	0x0008	
DTC.DTBR	DTC 転送情報を格納するメモリアドレスの上位 16 ビットを指定。	0xFFFF	
DTC.DTC SR	DTC のソフトウェア起動及び DTC ベクタアドレス (H 0000045A)を設定。	0x015A	
DTC.DTC SR	DTC のソフトウェア起動を禁止に設定。	0x06FF	DTC 転送終了 割り込みルーチン

5.5 送信プログラム

```

/*****
/* Filename      :  t_main.c                               */
/* Written       :  '06/07/24 Ver.4.0                       */
/* Purpose       :  HCAN2 for SH7047                       */
/*****
/*****
/* HCAN2 送信プログラム(例題 5)                             */
/*****
#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 "SH7047.h"     /* 周辺レジスタ定義ヘッダ ファイル */
/*****
/* 関数プロトタイプ宣言                                     */
/*****
void t_main(void);
void RM1_IRR1(void);
void DTC_SWDTEND(void);
/*****
/* 定数定義                                               */
/*****
#define DTMR (*(unsigned short *)0xFFFFD080) /* DTC レジスタ情報 */
#define DTCRA (*(unsigned short *)0xFFFFD082) /* DTC レジスタ情報 */
#define DTCRB (*(unsigned short *)0xFFFFD086) /* DTC レジスタ情報 */
#define DTSAR (*(unsigned long *)0xFFFFD088) /* DTC レジスタ情報 */
#define DTDAR (*(unsigned long *)0xFFFFD08C) /* DTC レジスタ情報 */
#define MBbuff (*(unsigned char *)0xFFFFD100) /* 受信データ格納エリア */
/*****
/* メインルーチン                                         */
/*****
void t_main(void){
    unsigned short *work;

/* 受信データ格納エリア初期化 */
    work = (unsigned short *)0xFFFFD100;
    do {
        *work = 0x0000;
        work++;
    } while(work < (unsigned short *)0xFFFFD108);

/* IRRO ビットクリア */
    HCAN_IRR = 0x0001; /* リセット割り込みフラグのクリア(クリア条件:1ライト)*/
/* 端子設定 */
    PB.PBCR1 = 0x0000; /* PB HCAN2 設定 */
    PB.PBCR2 = 0x000F; /* PB HCAN2 設定 */
/* メールボックス初期化 */
    work = (unsigned short *)0xFFFFB100;
    do {
        *work = 0x0000;
        work++;
    } while(work < (unsigned short *)0xFFFFB4F4);
/* MBC 設定 */
    HCAN_MB1.MC4 = 0x05; /* メールボックス1 をリモートフレーム送信可,データフレーム受信可に設定 */
/* 受信割り込み設定 */
    HCAN_IMR = 0xFFFFD; /* 受信メッセージ割り込み許可 */
    HCAN_MB1MRO = 0xFFFFD; /* メールボックス1 割り込み要求許可 */
    INTC.IPRK = 0x00F0; /* RM1 優先順位設定 */
    INTC.IPRG = 0x0E00; /* DTC 割り込み優先順位設定 */
/* リモートフレーム送信設定 */
    HCAN_MB1.MC0 = 0x5558; /* リモートフレーム、スタンダートフォーマットの選択、Identifier の設定 */
    HCAN_MB1.MC5 = 0x08; /* データ長:8ビット設定 */
/* ローカルアクセプタンスフィルタの設定 */
    HCAN_MB1.LAFM15 = 0x0000; /* メールボックス1 の Identifier 用フィルタマスクの設定 */

```



```

/* ビットレート(BCR)設定 : =50MHz 時ビットレート 250Kbps */
HCAN_BCR0 = 0x0009; /* BRP=9(10 システムクロック) */
HCAN_BCR1 = 0x4300; /* TSEG1=4(5tq), TSEG2=3(4tq) */
/* MCR0 ビットクリア */
HCAN_MCR = 0x0000; /* リセットリクエストビットのクリア */
/* メッセージの送信待ち状態に設定 */
HCAN_TXPRO = 0x0002; /* マスターックス1 を送信待ち状態に設定 */
/* 送信完了待ち */
while((HCAN_TXACKO & 0x0002) != 0x0002);
/* 送信完了フラグのクリア */
HCAN_TXACKO = 0x0002; /* 送信完了フラグのクリア(クリア条件: 1 ライト) */

set_imask(0);
while(1);
}

/*****
/* 受信割り込みルーチン */
/*****
#pragma interrupt(RM1_IRR1)
void RM1_IRR1(void){

/* 受信完了フラグのクリア */
HCAN_RXPRO = 0x0002; /* 受信完了フラグのクリア(クリア条件: 1 ライト) */
/* 受信割り込み禁止 */
HCAN_IMR = 0xFFFF; /* 受信メッセージ割り込み禁止 */
/* DTC 設定及び起動 */
DTMR = 0xA890; /* DTSAR, DTDAR 共に転送後インクリメント, ブロック転送モード, 8ビット転送 */
DTSAR = (unsigned long)&HCAN_MB1.MD7; /* 転送元アドレス設定 */
DTDAR = (unsigned long)&MBbuff; /* 転送先アドレス設定 */
DTCRA = 0x0001; /* ブロック転送 1 回 */
DTCRB = 0x0008; /* ブロック長 8 ビット */
DTC.DTBR = 0xFFFF; /* レジスタ情報先頭アドレス */
while((DTC.DTCSCR & 0x0100) != 0x0000); /* DTC ソフトウェア起動が禁止状態であることを確認 */
DTC.DTCSCR = 0x015A; /* DTC ソフトウェア起動許可 */
while((DTC.DTCSCR & 0x00FF) != 0x005A); /* DTC 起動ヘッダアドレスを確認 */

}

/*****
/* DTC 転送終了割り込みルーチン */
/*****
#pragma interrupt(DTC_SWDTEND)
void DTC_SWDTEND(void){
    DTC.DTCSCR &= 0x06FF; /* DTC ソフトウェア起動禁止 */
}

```

5.6 受信側フロー



図 5.4 受信側フローチャート(1)

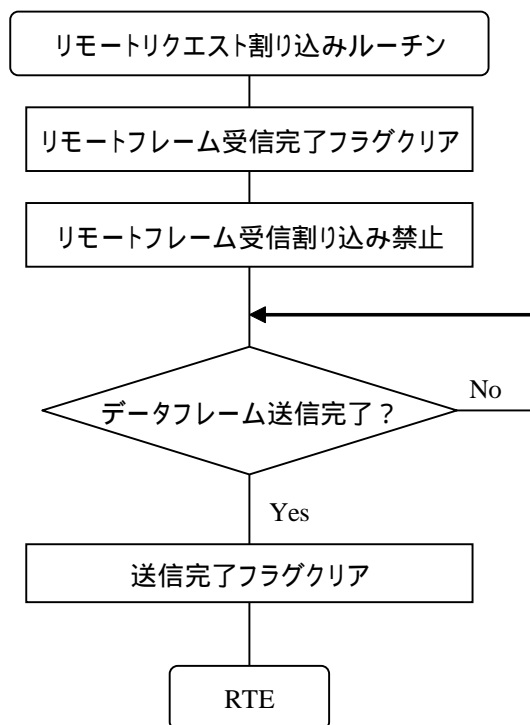


図 5.5 受信側フローチャート(2)

5.7 ソフトウェア説明(受信側)

- 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	r_main	HCAN2 の初期設定及び受信の設定を行います。
リモートリクエスト 割り込みルーチン	RM1_IRR2	リモートフレーム受信完了フラグクリアと割り込み禁止の設定及び送信完了フラグのチェックとクリアを行います。

- 使用レジスタの説明(使用端子及びポートレジスタは例題 1 参照)

レジスタ名	機能	設定値	モジュール名
work	MB 初期化に使用するワークレジスタ。		メインルーチン
HCAN_MCR	リセットリクエストビットのクリア。	0x0000	
HCAN_IRR	リセット/ホールド/スリープ割り込みフラグクリア。 (クリア条件: 1 ライト)	0x0001	
HCAN_BCR0	ビットレートを =50MHz 時 250Kbps に設定。	0x0009	
HCAN_BCR1		0x4300	
HCAN_MB1.MC0	MB1 をデータフレーム、スタンダードフォーマットに設定。また、Identifier(H 555)を設定。	0x5550	
HCAN_MB1.MC4	MB1 をリモートフレーム受信可、データフレーム送信可に設定。また、ATX(データフレームの自動送信)を設定。	0x11	
HCAN_MB1.MC5	MB1 の送信サイズを 8 バイトのデータ長に設定。	0x08	
HCAN_MB1.MC7	MB1 の 1 バイト目送信データ (H 55) を設定。	0x55	
HCAN_MB1.MC8	MB1 の 2 バイト目送信データ (H 66) を設定。	0x66	
HCAN_MB1.MC9	MB1 の 3 バイト目送信データ (H 77) を設定。	0x77	
HCAN_MB1.MC10	MB1 の 4 バイト目送信データ (H 88) を設定。	0x88	
HCAN_MB1.MC11	MB1 の 5 バイト目送信データ (H 99) を設定。	0x99	
HCAN_MB1.MC12	MB1 の 6 バイト目送信データ (H AA) を設定。	0xAA	
HCAN_MB1.MC13	MB1 の 7 バイト目送信データ (H BB) を設定。	0xBB	
HCAN_MB1.MC14	MB1 の 8 バイト目送信データ (H FF) を設定。	0xFF	
HCAN_MB1.LAFM15	MB1 の Identifier 用フィルタマスクを設定。	0x0000	
HCAN_IMR	リモートリクエスト割り込み許可。	0xFFFFB	
HCAN_MB1MRO	MB1 の割り込み要求を許可。	0xFFFFD	
INTC.IPRK	HCAN2 の割り込み要求の優先順位を設定。	0x00F0	
HCAN_RFPRO	MB1 のリモートフレーム受信完了フラグのクリア。 (クリア条件: 1 ライト)	0x0002	リモートリクエスト 割り込みルーチン
HCAN_IMR	リモートリクエスト割り込み禁止。	0xFFFF	
HCAN_TXACK0	MB1 の送信完了フラグのチェックとクリア。 (クリア条件: 1 ライト)	0x0002	

5.8 受信プログラム

```

/*****/
/* Filename      :  r_main.c                */
/* Written       :  '06/07/24 Ver.4.0       */
/* Purpose       :  HCAN2 for SH7047        */
/*****/
/*****/
/* HCAN2 受信プログラム(例題 5)            */
/*****/
#include <stdio.h>          /*ライブラリ関数用ヘッダファイル */
#include <machine.h>        /*ライブラリ関数用ヘッダファイル */
#include "SH7047.h"        /*周辺レジスタ定義ヘッダファイル */
/*****/
/* 関数プロトタイプ宣言                    */
/*****/
void r_main(void);
void RM1_IRR2(void);
/*****/
/* メインルーチン                          */
/*****/
void r_main(void){
    unsigned short *work;

/* IRRO ビットクリア */
    HCAN_IRR = 0x0001;      /* リセット割り込みフラグのクリア(クリア条件:1バイト)*/
/* 端子設定 */
    PB.PBCR1 = 0x0000;     /* PB HCAN2 設定 */
    PB.PBCR2 = 0x000F;     /* PB HCAN2 設定 */
/* メールボックス初期化 */
    work = (unsigned short *)0xFFFFB100;
    do {
        *work = 0x0000;
        work++;
    } while(work < (unsigned short *)0xFFFFB4F4);
/* MBC 設定 */
    HCAN_MB1.MC4 = 0x11;   /* メールボックス1をATX使用,リモートフレーム受信可,データフレーム送信可に設定 */
/* リモートリクエスト割り込み設定 */
    HCAN_IMR = 0xFFFF;    /* リモートリクエスト割り込み許可 */
    HCAN_MB1.MR0 = 0xFFFF; /* メールボックス1割り込み要求許可 */
    INTC.IPRK = 0x00F0;   /* RM1 優先順位設定 */
/* 送信データ設定 */
    HCAN_MB1.MC0 = 0x5550; /* 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_MB1.MD7 = 0x55;   /* 送信データ:01010101 設定 */
    HCAN_MB1.MD8 = 0x66;   /* 送信データ:01100110 設定 */
    HCAN_MB1.MD9 = 0x77;   /* 送信データ:01110111 設定 */
    HCAN_MB1.MD10 = 0x88;  /* 送信データ:10001000 設定 */
    HCAN_MB1.MD11 = 0x99;  /* 送信データ:10011001 設定 */
    HCAN_MB1.MD12 = 0xAA;  /* 送信データ:10101010 設定 */
    HCAN_MB1.MD13 = 0xBB;  /* 送信データ:10111011 設定 */
    HCAN_MB1.MD14 = 0xFF;  /* 送信データ:11111111 設定 */
/* ローカルアクセプタンスフィルタの設定 */
    HCAN_MB1.LAFM15 = 0x0000; /* メールボックス1のIdentifier用フィルタマスクの設定 */
/* ビットレート(BCR)設定: =50MHz時ビットレート250Kbps */
    HCAN_BCR0 = 0x0009;    /* BRP=9(10システムクロック) */
    HCAN_BCR1 = 0x4300;    /* TSEG1=4(5tq),TSEG2=3(4tq) */
/* MCRO ビットクリア */
    HCAN_MCR = 0x0000;    /* リセットリクエストビットのクリア */

    set_imask(0);
    while(1);
}

```

```

/*****/
/* リモートリクエスト受信割り込みルーチン */
/*****/
#pragma interrupt(RM1_IRR2)
void RM1_IRR2(void){

/* リモートリクエストレジスタクリア */
HCAN_RFPRO = 0x0002; /* リモート受信完了フラグクリア(クリア条件:1バイト) */
/* 受信割り込み禁止 */
HCAN_IMR = 0xFFFF; /* リモート受信割り込み禁止 */
/* 送信完了待ち */
while((HCAN_TXACK0 & 0x0002) != 0x0002);
/* 送信完了フラグのクリア */
HCAN_TXACK0 = 0x0002; /* 送信完了フラグのクリア(クリア条件:1バイト) */

}
    
```

5.9 動作波形(送受信)

図 5.6 に本アプリケーション動作時の波形を示します。

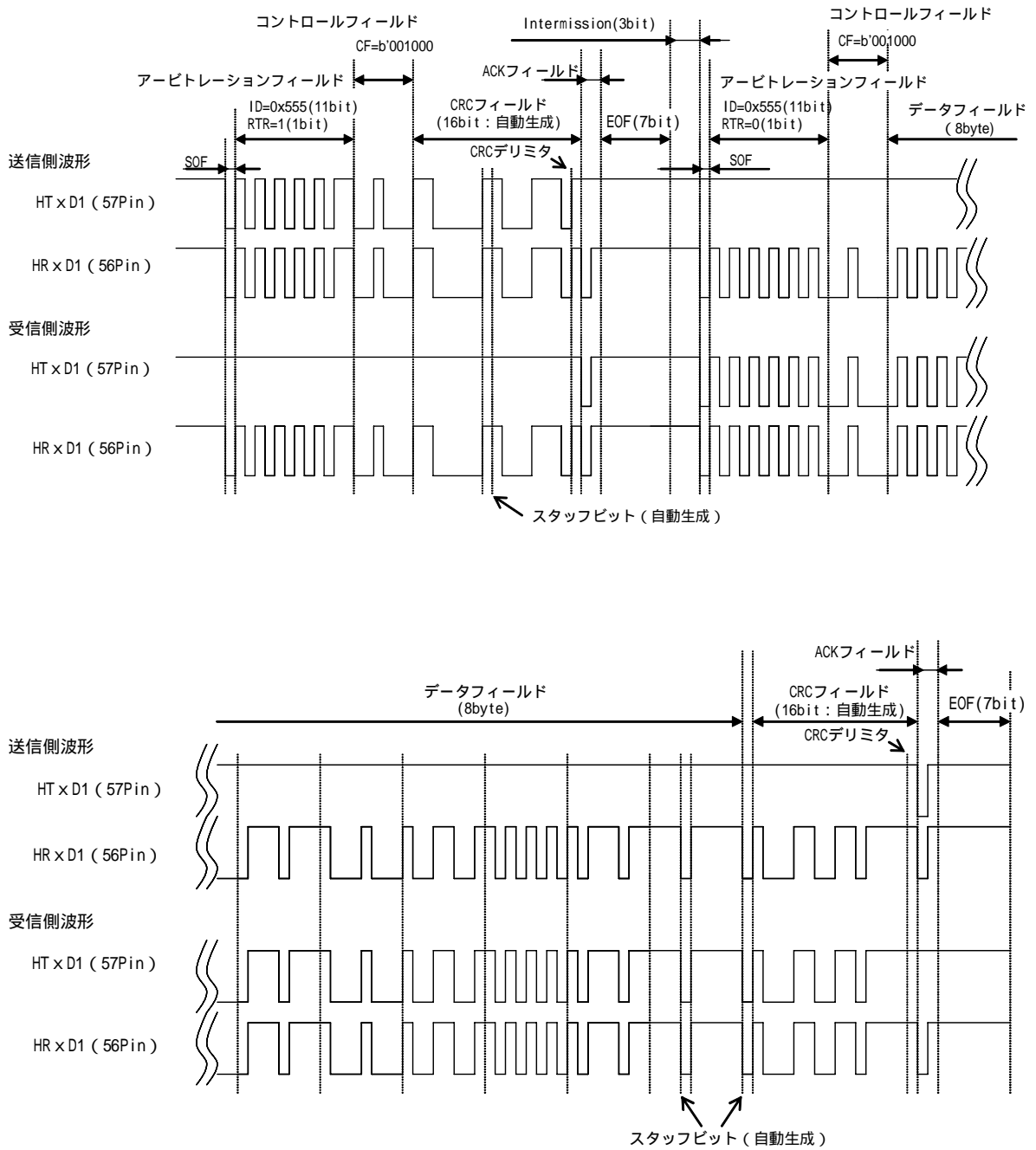


図 5.6 動作波形

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2003.8.27	—	新規作成
2.00	2006.8.25	—	ハードウェアマニュアル改訂による修正 (SH7047 グループハードウェアマニュアル第2版)

安全設計に関するお願い

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

本資料ご利用に際しての留意事項

1. 本資料は、お客様が用途に応じた適切なルネサス テクノロジ製品をご購入いただくための参考資料であり、本資料中に記載の技術情報についてルネサス テクノロジが所有する知的財産権その他の権利の実施、使用を許諾するものではありません。
2. 本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他応用回路例の使用に起因する損害、第三者所有の権利に対する侵害に関し、ルネサス テクノロジは責任を負いません。
3. 本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他全ての情報は本資料発行時点のものであり、ルネサス テクノロジは、予告なしに、本資料に記載した製品または仕様を変更することがあります。ルネサス テクノロジ半導体製品のご購入に当たりますとは、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、海底中継用機器あるいはシステムなど、特殊用途へのご利用をご検討の際は、ルネサス テクノロジ、ルネサス販売または特約店へご照会ください。
7. 本資料の転載、複製については、文書によるルネサス テクノロジの事前の承諾が必要です。
8. 本資料に関し詳細についてのお問い合わせ、その他お気づきの点がございましたらルネサス テクノロジ、ルネサス販売または特約店までご照会ください。