

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。

32192/32196 グループ

CAN モジュールの応用（リモートフレーム送信）

1. 要約

この資料は 32192/32196 グループの CAN（Controller Area Network）を使用した参考プログラム例を掲載しています。

2. はじめに

この資料で説明する応用例は次のマイコン、条件での利用に適用されます。

- ・マイコン : 32192 グループ (M32192F8VFP、M32192F8UFP、M32192F8TFP)
32196 グループ (M32196F8VFP、M32196F8UFP、M32196F8TFP)
- ・動作周波数 : 128～160MHz (参考プログラムは 160MHz を想定して作成しています)
- ・動作ボード : 32192 グループ用スタータキット

3. 応用技術の説明

3.1 CAN モジュールの概要

32192/32196 は CAN Specification V2.0B active 準拠の Full CAN モジュールを 2 系統内蔵しています。32 本のメッセージスロットと 4 本のマスクレジスタを有効に活用することにより、データ処理に伴う CPU の負荷を低減することができます。

CAN 機能の詳細は、32192/32196 グループのハードウェアマニュアルと 32192/32196 グループ CAN モジュールの概要を参照してください。

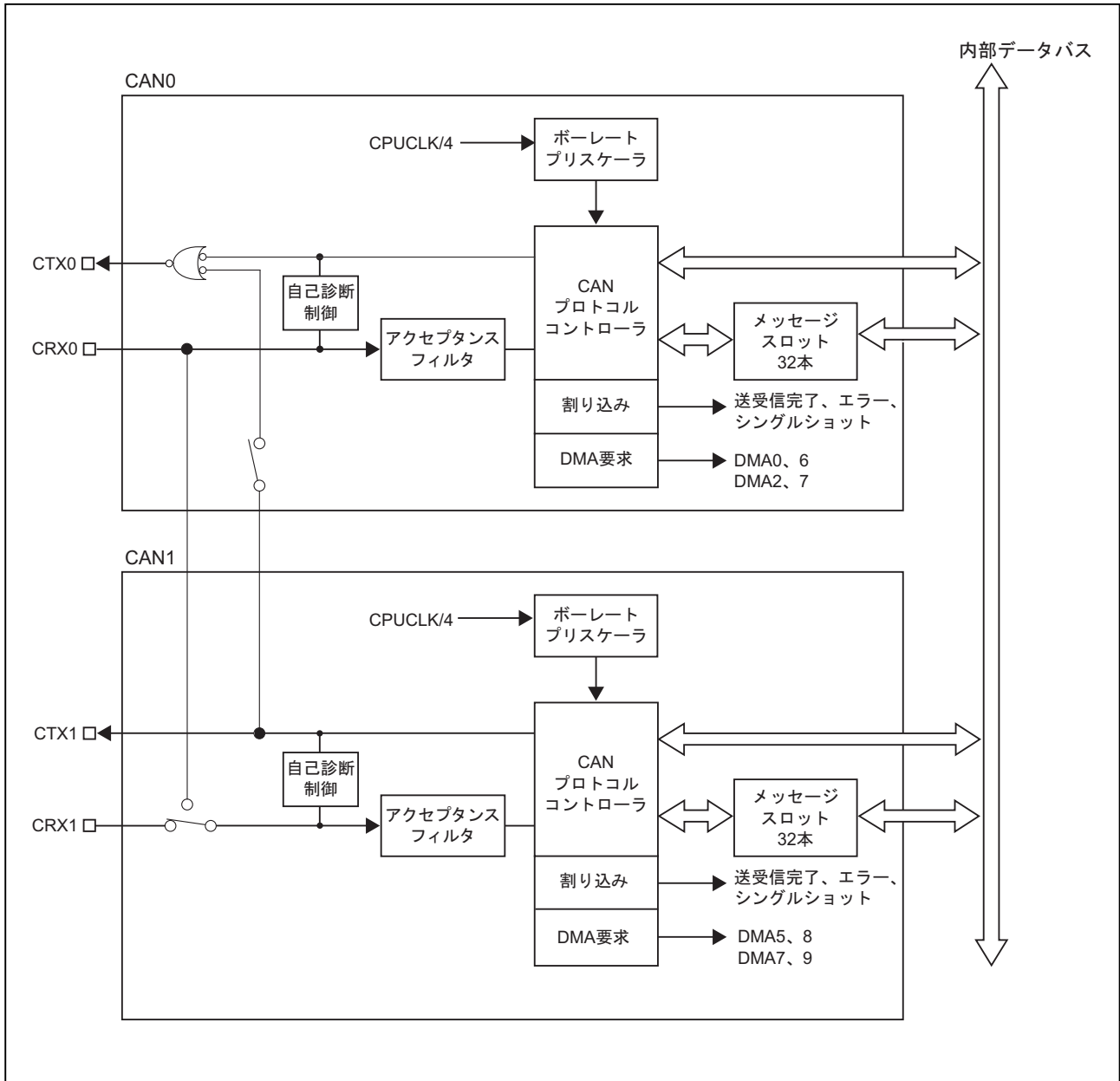


図 3.1.1 CAN モジュールブロック図

4. CAN リモートフレーム送信参考プログラム

4.1 参考プログラムの概要

本参考プログラムでは、CAN バス速度 125kbps、CAN0 のスロット 0 から ID : 0、DLC : 1 の標準フォーマットのリモートフレームを一回送信し、他ノードから送り返されたデータフレームを受信します。データフレームの受信後、データを RAM 上に読み出します。割り込みと DMA 転送は使用していません。

4.2 初期設定処理

CAN モジュールの初期設定のフローを図 4.2.1 に示します。

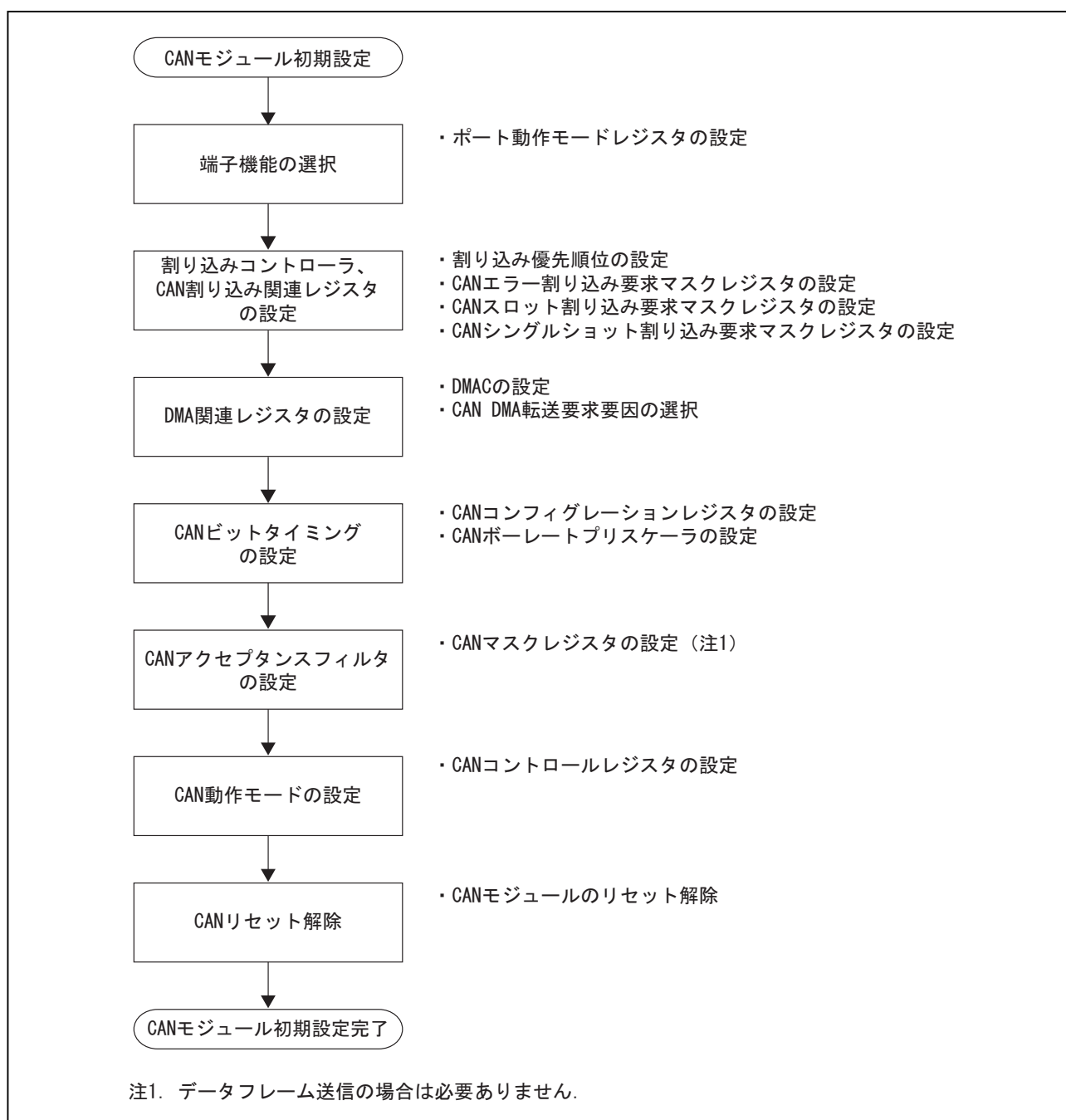


図 4.2.1 CAN モジュール初期設定

4.3 送信処理

リモートフレーム送信処理手順のフローを図 4.3.1 に示します。

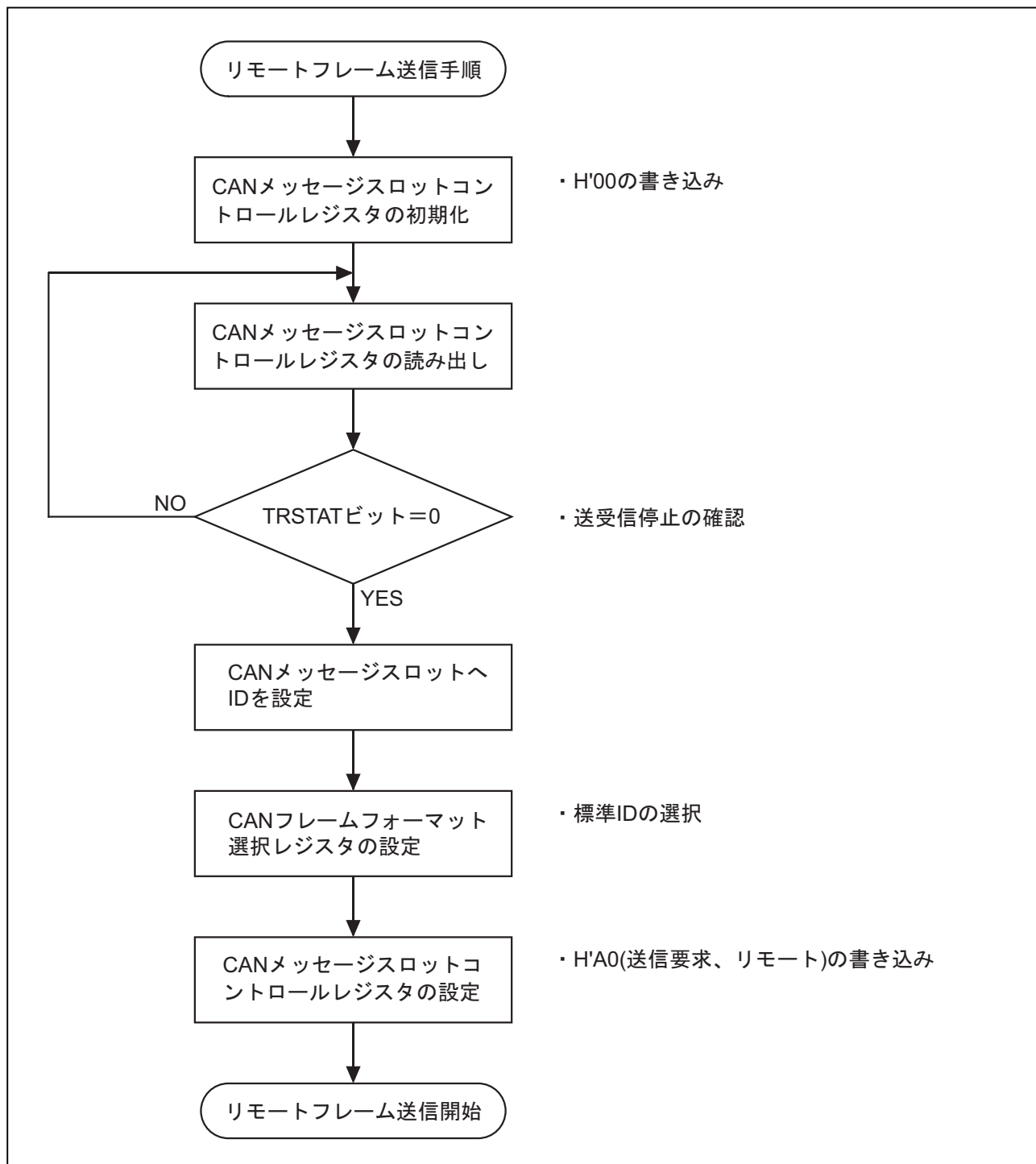


図 4.3.1 リモートフレーム送信手順

4.4 CAN メッセージスロットコントロールレジスタの状態

リモートフレーム送信時の CAN メッセージスロットコントロールレジスタの状態遷移図を図 4.4.1 に示します。

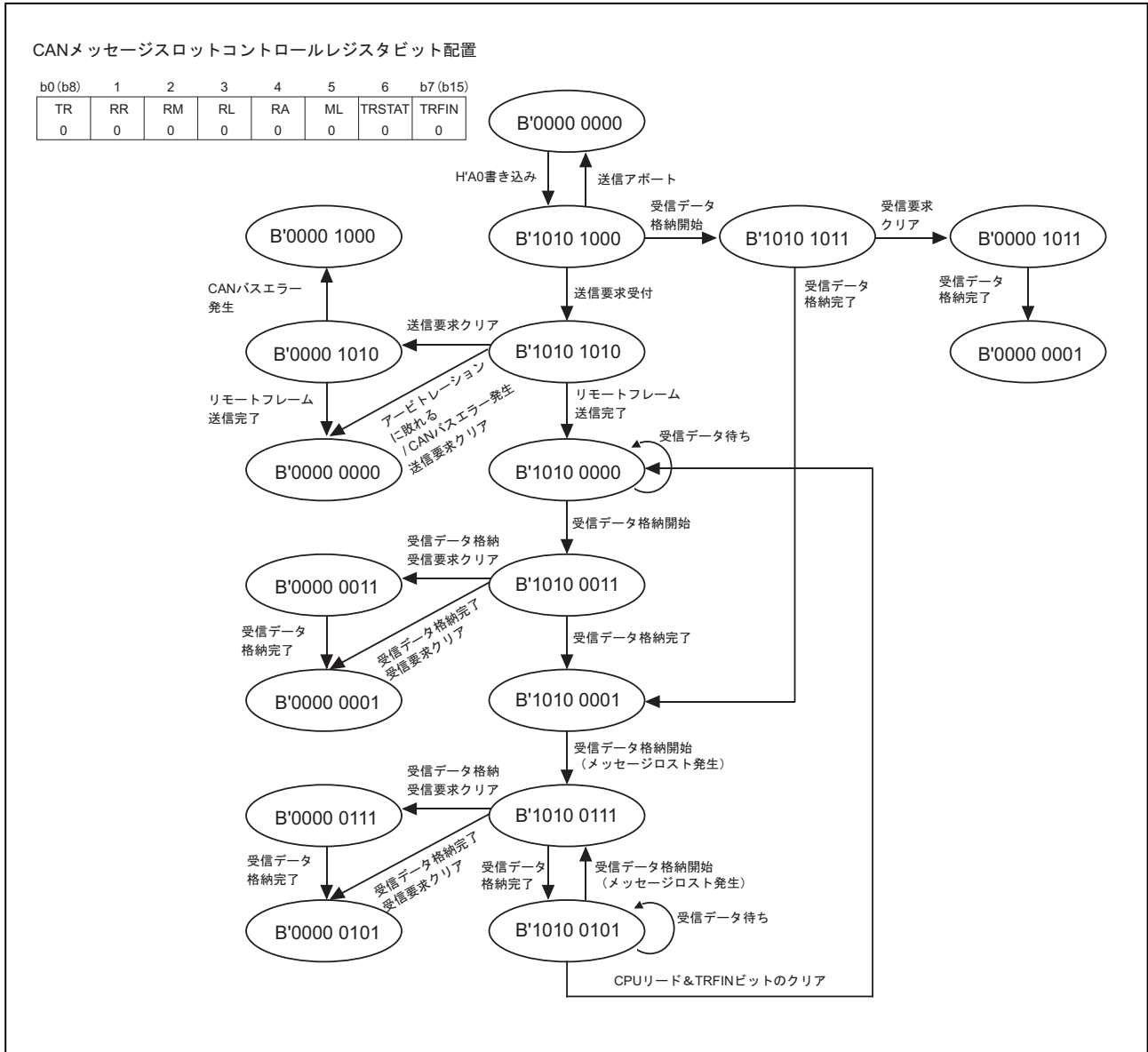


図 4.4.1 リモートフレーム送信時の CAN メッセージスロットコントロールレジスタの状態

4.5 参考プログラムの解説

注. 使用しているレジスタを (レジスタ名 : ビット名) と表記しています。

4.5.1 CAN モジュール初期化関数 (can_init())

- (1) CAN バスモードの設定
 - ・ CAN バスモード制御レジスタを CAN バス独立に設定 (CANBUSCR : CBUSSEL)
 - ・ CAN0 コントロールレジスタをリセット状態に設定 (CAN0CNT : FRST、RST)
- (2) 端子機能の設定
 - ・ P22 動作モードレジスタと P22 周辺機能選択レジスタの設定 (P22MOD、P22SMOD)
ポート P220 を CTX0 に、ポート P221 を CRX0 に設定

注. ・ CAN1 を使用するときには、ポート入力特別機能制御レジスタのポート入力許可ビット (PICNT : PIEN0) を "1" (入力許可) に設定する処理を追加してください。
- (3) 割り込み設定
 - ・ CAN0 送受信&エラー割り込み制御レジスタを割り込み禁止に設定 (ICAN0CR : ILEVEL)
- (4) CAN0 割り込み関連レジスタの設定
 - ・ CAN0 スロット割り込み要求ステータスレジスタをクリア (CAN0SLISTW)
 - ・ CAN0 エラー割り込み要求ステータスレジスタをクリア (CAN0ERIST)
 - ・ CAN0 スロット割り込み要求マスクレジスタを割り込み要求禁止に設定 (CAN0SLIMKW)
 - ・ CAN0 エラー割り込み要求マスクレジスタを CAN バスエラー割り込み禁止、エラーパッシブ割り込み禁止、バスオフ割り込み禁止に設定 (CAN0ERIMK : EIM、PIM、OIM)
- (5) CAN0 コンフィグレーションレジスタの設定 (CAN0CONF : SJW、PH2、PH1、PRB、SAM)
 - ・ Propagation Segment (PRB) を 5 Tq に設定
 - ・ Phase Segment1 (PH1) を 7 Tq に設定
 - ・ Phase Segment2 (PH2) を 7 Tq に設定
 - ・ reSynchronization Jump Width (SJW : 再同期幅) を 1Tq に設定
 - ・ サンプルング数を 1 回に設定

上記設定で 1 ビット内の Tq 数は 20、サンプルングポイントは 65% になります。
ビットタイミングを図 4.5.1 に示します。

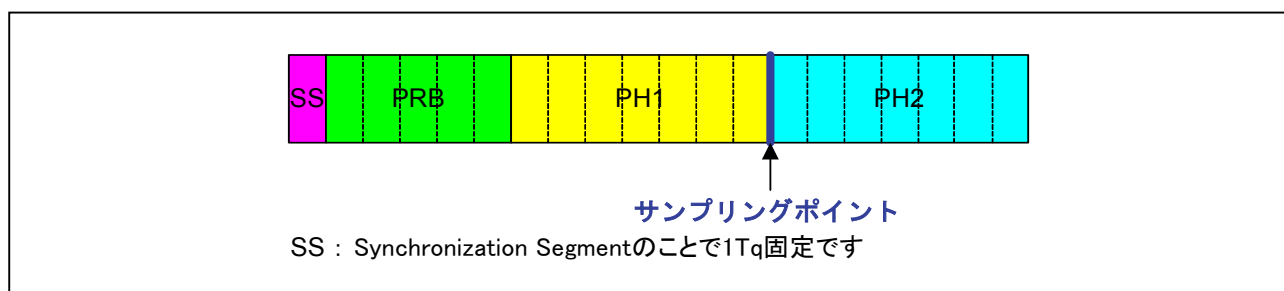


図 4.5.1 ビットタイミング

- (6) CAN0 ボーレートプリスケアラの設定 (CAN0BRP)
 - ・ ボーレートプリスケアラ値に "15" を設定 ((160 MHz / 4) / (125 kbps × 20 Tq) - 1 = 15)

ボーレートプリスケアラ (BRP) 設定値の計算式

$$\text{BRP 設定値} = \frac{\text{CPUCLK}/4}{\text{ボーレート}[\text{bps}] \times 1 \text{ ビット内の Tq 数}} - 1$$

(7) CAN0 アクセプタンスフィルタの設定

下記すべてのレジスタを ID チェックありに設定

- ・ CAN0 グローバルマスクレジスタ A 標準 ID (C0GMSKAS0、C0GMSKAS1)
- ・ CAN0 グローバルマスクレジスタ A 拡張 ID (C0GMSKAE0、C0GMSKAE1、C0GMSKAE2)
- ・ CAN0 グローバルマスクレジスタ B 標準 ID (C0GMSKBS0、C0GMSKBS1)
- ・ CAN0 グローバルマスクレジスタ B 拡張 ID (C0GMSKBE0、C0GMSKBE1、C0GMSKBE2)
- ・ CAN0 ローカルマスクレジスタ A 標準 ID (C0LMSKAS0、C0LMSKAS1)
- ・ CAN0 ローカルマスクレジスタ A 拡張 ID (C0LMSKAE0、C0LMSKAE1、C0LMSKAE2)
- ・ CAN0 ローカルマスクレジスタ B 標準 ID (C0LMSKBS0、C0LMSKBS1)
- ・ CAN0 ローカルマスクレジスタ B 拡張 ID (C0LMSKBE0、C0LMSKBE1、C0LMSKBE2)

(8) CAN0 拡張 ID レジスタの設定

- ・ すべてのスロットを標準 ID フォーマットに設定 (CAN0FFSW)

(9) CAN0 コントロールレジスタの設定 (CAN0CNT : TSP、FRST、BCM、LBM、RST)

- ・ タイムスタンププリスケアラに CAN バスビットクロックを選択
- ・ 強制リセットを解除
- ・ BasicCAN 機能を無効に設定
- ・ ループバック機能を無効に設定
- ・ CAN リセットを解除

4.5.2 メイン関数 (main())

(1) CAN モジュール初期化関数の呼び出し

(2) リモートフレーム送信処理関数の呼び出し

(3) リモートフレームの送信完了を確認

- ・ CAN0 スロット 0 割り込み要求ビットを確認 (CAN0SLISTW : SSB0)

(4) CAN0 スロット 0 割り込み要求ビットをクリア (CAN0SLISTW : SSB0)

(5) データフレームの受信完了を確認

- ・ CAN0 スロット 0 割り込み要求ビットを確認 (CAN0SLISTW : SSB0)

(6) データ読み出し処理関数の呼び出し

4.5.3 リモートフレーム送信処理関数 (remote_send())

(1) CAN0 メッセージスロット 0 コントロールレジスタの初期化

- ・ すべてのフラグをクリアし、送受信動作を停止 (COMSL0CNT)

(2) 送受信動作の停止を確認

- ・ 送受信ステータスビットを確認 (COMSL0CNT : TRSTAT)

(3) スロット 0 の送信データ作成

- ・ ID を”0”に設定 (COMSL0SID0、COMSL0SID1)
- ・ データ長を”1”に設定 (COMSL0DLC)

(4) CAN0 拡張 ID レジスタの設定

- ・ すべてのスロットを標準 ID フォーマットに設定 (CAN0FFSW)

(5) CAN0 メッセージスロット 0 コントロールレジスタの設定

- ・ リモートフレーム送信要求に設定 (COMSL0CNT : TR、RM)

4.5.4 データ読み出し処理関数（remote_send_read_data()）

(1) 送受信完了ビットをクリア（COMSL0CNT : TRFIN）

(2) 受信したデータの読み出し（COMSL0DT0～COMSL0DT7）

(3) 送受信完了ビットの確認（COMSL0CNT : TRFIN）

送受信完了ビットがセットされていた場合、受信データの読み出し中に新しい受信データが格納されたことを示します。この場合、読み出した値の中に不定値が含まれているので、送受信完了ビットのクリアからやり直します。

4.6 参考プログラム例

下記に CAN0 リモートフレーム送信の参考プログラム例を示します。

尚、下記の参考プログラム例では、SFR 定義ファイルが必要です。最新の SFR 定義ファイルはホームページよりダウンロードできます。SFR 定義ファイル使用時は、お客様の環境に合わせてパスの設定をおこなってください。

4.6.1 init.c

```

1  /*"FILE COMMENT"*****
2  *      M32R C Programming          Rev. 1.00
3  *      < Sample Program for 32192 >
4  *      < CAN init >
5  *
6  *      Copyright (c) 2005 Renesas Technology Corporation
7  *      All Rights Reserved
8  *      *****/
9
10 /*****
11 /*      Include file                */
12 /*****
13
14 #include          "..\inc\sfr32192_pragma.h"
15
16 /*****
17 /*      Function prototype declaration */
18 /*****
19
20 void          can_init(void);
21
22 /*"FUNC COMMENT"*****
23 * Function name: void can_init(void)
24 *-----
25 * Description   : Initializes CAN module
26 *-----
27 * Argument     : -
28 *-----
29 * Returns      : -
30 *-----
31 * Notes       :
32 *"FUNC COMMENT END"*****
33 void can_init(void)
34 {
35     CANBUSCR = 0x02;
36     CANBUSCR = 0x00;          /* CAN Bus mode setting */
37     CANOCNT = 0x0011;        /* CAN control register */
38
39     /* Setting input/output port operation mode register */
40     P22MOD  &= ~0xC0u;        /* P220,P221 clear */
41     P22SMOD &= ~0xC0u;        /* CTX, CRX select */
42     P22MOD |= 0xC0u;         /* P220 used as CTX, P221 used as CRX */
43
44     /* To use CAN1, set it up here */
45     /*
46     * - P7MOD  &= ~0x03u
47     * - P7SMOD |= 0x03u
48     * - P7MOD  |= 0x03u
49     * - PICNT  |= 0x01u
50     */
51
52     /* Setting interrupt controller */
53     ICANOCR = 0x07;          /* CAN0 interrupt priority level 7 (interrupt disabled) */
54
55
56     /* Setting CAN0 related interrupt mask register */
57     CANOSLISTW = 0x0000000ul; /* Clear CAN0 slot transmit/receive-finished interrupt request */
58     CANOERIST  = 0x00;        /* Clear CAN0 error interrupt request */
59     CANOSLIMKW = 0x0000000ul; /* Disable CAN0 slot transmit/receive-finished interrupt */
60     CANOERIMK  = 0x00;        /* Disable CAN0 error interrupt */
61
62
63     /* Setting CAN configuration register */

```

```

64     CANOCONF = 0x3680;                               /* SJW=1, Sync(1)+Prop(5)+PH1(7)+PH2(7), sampling count = 1 */
65     CANOBRP = (16 - 1);                               /* Baud rate: 160/4=40 MHz / divided by 16 / 20 Tq -> 125 Kbps */
66
67     /* Setting ID mask register */
68     COGMSKAS0 = 0xff;                                 /* Global mask register A */
69     COGMSKAS1 = 0xff;
70     COGMSKAE0 = 0xff;
71     COGMSKAE1 = 0xff;
72     COGMSKAE2 = 0xff;
73     COGMSKBS0 = 0xff;                                 /* Global mask register B */
74     COGMSKBS1 = 0xff;
75     COGMSKBE0 = 0xff;
76     COGMSKBE1 = 0xff;
77     COGMSKBE2 = 0xff;
78
79     COILMSKAS0 = 0xff;                                 /* Local mask register A */
80     COILMSKAS1 = 0xff;
81     COILMSKAE0 = 0xff;
82     COILMSKAE1 = 0xff;
83     COILMSKAE2 = 0xff;
84     COILMSKBS0 = 0xff;                                 /* Local mask register B */
85     COILMSKBS1 = 0xff;
86     COILMSKBE0 = 0xff;
87     COILMSKBE1 = 0xff;
88     COILMSKBE2 = 0xff;
89
90     /* To use in BasicCAN mode, set it up here. */
91     /*
92     * - Set IDE30/31 of CANOFFSW
93     * - Set ID of slots 30/31
94     * - Set local mask registers A/B
95     * - Set slots 30/31 for data frame reception
96     */
97
98     /* Setting CAN operation mode */
99     CANOFFSW = 0x00000000ul;                           /* Select standard format frame */
100
101     /* Negating CAN reset */
102     CANOCNT = 0x0000;                                 /* Clear FRST and RST bits and disable BasicCAN function */
103     /* Disable loopback function and select timestamp divide-by-1 */
104 }

```

4.6.2 remote_send.c

```

1  /*"FILE COMMENT"*****
2  *      M32R C Programming          Rev. 1.00
3  *      < Sample Program for 32192 >
4  *      < CAN remote_send >
5  *
6  *      Copyright (c) 2005 Renesas Technology Corporation
7  *      All Rights Reserved
8  *      *****/
9
10 /******
11 /*      Include file          */
12 /******
13
14 #include          "..\inc\sfr32192_pragma.h"
15
16 /******
17 /*      Function prototype declaration          */
18 /******
19
20     void          main(void);
21     void          remote_send(void);
22     void          remote_send_read_data(void);
23 extern void          can_init(void);
24
25 /******
26 /*      Externally referenced variable          */
27 /******
28
29     UCHAR          CAN_DATA[8];          /* Used to store received data */
30
31 /*"FUNC COMMENT"*****
32 * Function name: void main(void)
33 *-----
34 * Description : Remote frame transmission sample program
35 *-----
36 * Argument : -
37 *-----
38 * Returns : -
39 *-----
40 * Notes :
41 /*"FUNC COMMENT END"*****
42 void main(void)
43 {
44     /* Initialize CAN module */
45     can_init();
46
47     /* CAN module operation */
48     remote_send();
49
50     /* Wait until remote frame is sent */
51     while( ( CANOSLISTW & SSB0) == 0x00000000ul){
52         ;
53     }
54     CANOSLISTW = 0x00000000ul;
55
56     /* Wait until data frame is received */
57     while( ( CANOSLISTW & SSB0) == 0x00000000ul){
58         ;
59     }
60
61     /* Read out received data frame */
62     remote_send_read_data();
63
64
65     while( 1 ){
66         ;
67     }
68 }
69
70 /*"FUNC COMMENT"*****
71 * Function name: void remote_send(void)
72 *-----
73 * Description : Sends remote frame from slot 0
74 *-----
75 * Argument : -

```

```

76  *-----
77  * Returns      : -
78  *-----
79  * Notes        :
80  *"FUNC COMMENT END"*****
81  void remote_send(void)
82  {
83      COMSLOCNT = 0x00;                               /* Initialize CAN message slot control
register */
84      while ( ( COMSLOCNT & TRSTAT) != 0u){           /* Verify that transmit operation is idle */
85          ;
86      }
87
88      /* Set ID and DLC in message slot 0 */
89      COMSLOSID0 = 0x00;                               /* ID : 0 */
90      COMSLOSID1 = 0x00;
91      COMSLODLC = 0x01;                               /* DLC : 1 */
92
93      /* Set extended ID register */
94      CANOFFSW = 0x00000000ul;                       /* Select standard format */
95
96      /* Set CAN message slot control register */
97      COMSLOCNT = 0xA0;                               /* Request transmission of remote frame */
98  }
99
100 /*"FUNC COMMENT"*****
101 * Function name: void remote_send_read_data(void)
102 *-----
103 * Description   : Reads out received data frame when CAN module has been set for remote frame
transmission
104 *-----
105 * Argument      : -
106 *-----
107 * Returns       : -
108 *-----
109 * Notes         :
110 *"FUNC COMMENT END"*****
111 void remote_send_read_data(void)
112 {
113     do {
114         COMSLOCNT = 0xAE;                             /* Clear TRFIN bit */
115
116         /* Read out message slot 0 */
117         CAN_DATA[0] = COMSLODT0;
118         CAN_DATA[1] = COMSLODT1;
119         CAN_DATA[2] = COMSLODT2;
120         CAN_DATA[3] = COMSLODT3;
121         CAN_DATA[4] = COMSLODT4;
122         CAN_DATA[5] = COMSLODT5;
123         CAN_DATA[6] = COMSLODT6;
124         CAN_DATA[7] = COMSLODT7;
125
126     } while( ( COMSLOCNT & TRFIN) != 0u );           /* Redo if TRFIN is set */
127
128     /* If necessary, check ML bit to see if message is lost
129     *
130     * if( (COMSLOCNT & ML) != 0)
131     *     (Processing for messages lost);
132     */
133 }
134 }

```

5. 参考ドキュメント

- 32192 グループ データシート Rev.1.01
- 32196 グループ データシート Rev.1.00
- 32192/32196 グループ ハードウェアマニュアル Rev.1.00
- 32192/32196 グループ CAN モジュールの概要 Rev.1.00
- M3T-CC32R V.4.30 ユーザーズマニュアル（C コンパイラ編）
- M3T-AS32R V.4.30 ユーザーズマニュアル（アセンブラ編）
- M32R-FPU ソフトウェアマニュアル Rev.1.01
（最新版をルネサス テクノロジホームページから入手してください。）
- Robert Bosch GmbH/Bosch's CAN : <http://www.can.bosch.com/>
- CiA: CAN in Automation : <http://www.can-cia.de/>

6. ホームページとサポート窓口

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

ルネサス製品全般に関するお問合せと M32R ファミリに関する技術的なお問合せ先
カスタマサポートセンタ : csc@renesas.com

改訂記録	32192/32196 グループ CAN モジュールの応用（リモートフレーム送信） アプリケーションノート
------	---

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2005.06.16	-	初版発行

安全設計に関するお願い

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

本資料ご利用に際しての留意事項

1. 本資料は、お客様が用途に応じた適切なルネサス テクノロジ製品をご購入いただくための参考資料であり、本資料中に記載の技術情報についてルネサス テクノロジが所有する知的財産権その他の権利の実施、使用を許諾するものではありません。
2. 本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他応用回路例の使用に起因する損害、第三者所有の権利に対する侵害に関し、ルネサス テクノロジは責任を負いません。
3. 本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他全ての情報は本資料発行時点のものであり、ルネサス テクノロジは、予告なしに、本資料に記載した製品または仕様を変更することがあります。ルネサス テクノロジ半導体製品のご購入に当たりますは、事前にルネサス テクノロジ、ルネサス販売または特約店へ最新の情報をご確認頂きますとともに、ルネサス テクノロジホームページ(<http://www.renesas.com>)などを通じて公開される情報に常にご注意ください。
4. 本資料に記載した情報は、正確を期すため、慎重に制作したものです。万一本資料の記述誤りに起因する損害がお客様に生じた場合には、ルネサス テクノロジはその責任を負いません。
5. 本資料に記載の製品データ、図、表に示す技術的な内容、プログラム及びアルゴリズムを流用する場合は、技術内容、プログラム、アルゴリズム単位で評価するだけでなく、システム全体で十分に評価し、お客様の責任において適用可否を判断してください。ルネサス テクノロジは、適用可否に対する責任を負いません。
6. 本資料に記載された製品は、人命にかかわるような状況の下で使用される機器あるいはシステムに用いられることを目的として設計、製造されたものではありません。本資料に記載の製品を運輸、移動体用、医療用、航空宇宙用、原子力制御用、海底中継用機器あるいはシステムなど、特殊用途へのご利用をご検討の際には、ルネサス テクノロジ、ルネサス販売または特約店へご照会ください。
7. 本資料の転載、複製については、文書によるルネサス テクノロジの事前の承諾が必要です。
8. 本資料に関し詳細についてのお問い合わせ、その他お気づきの点がございましたらルネサス テクノロジ、ルネサス販売または特約店までご照会ください。