

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、誤りが無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。

## SH7280 グループ

### DMAC 内蔵モジュールへの転送例

---

#### 要旨

この資料は SH7285 のダイレクトメモリアクセスコントローラ (DMAC) の内蔵モジュールへの転送例を掲載しています。

#### 動作確認デバイス

SH7285

#### 目次

1. はじめに .....	2
2. 応用例の説明 .....	3
3. 参考プログラム例 .....	9
4. 参考ドキュメント .....	14

## 1. はじめに

### 1.1 仕様

- DMAC チャンネル 1 を使用して外部メモリから FIFO 内蔵シリアルコミュニケーションインタフェース (SCIF チャンネル 3) のトランスミット FIFO データレジスタ (SCFTDR) へのデータ転送を行い、文字列の送信を行います。
- DMA 転送要求として、SCIF の送信 FIFO データエンブティ転送要求 (内蔵周辺モジュールリクエスト) を使用します。

### 1.2 使用機能

- ダイレクトメモリアクセスコントローラ (DMAC チャンネル 1)
- FIFO 内蔵シリアルコミュニケーションインタフェース (SCIF チャンネル 3)

### 1.3 適用条件

- マイコン: SH7285 / SH7286 / SH7243
- 動作周波数:
 

内部クロック	100MHz
バスクロック	50MHz
周辺クロック	50MHz
- C コンパイラ: ルネサス テクノロジ製  
SuperH RISC engine ファミリ C/C++ コンパイラパッケージ Ver.9.01 Release01
- コンパイルオプション:
 

```
-cpu = sh2a -include = "${WORKSPDIR}\inc"
-object = "${CONFIGDIR}\${FILELEAF}.obj" -debug -gbr = auto
-chgincpath -errorpath -global_volatile = 0 -opt_range = all
-infinite_loop = 0 -del_vacant_loop = 0 -struct_alloc = 1 -nologo
```

## 2. 応用例の説明

本プログラム例ではDMACを使用し、内蔵周辺モジュールリクエストにより外部メモリからSCIFへのデータ転送を行います。

### 2.1 使用機能の動作概要

DMACはDMA転送要求があると、決められたチャンネルの優先順位にしたがって転送を開始し、転送終了条件が満たされると転送を終了します。転送要求にはオートリクエスト、外部リクエスト、内蔵周辺モジュールリクエストの3種類のモードがあります。バスモードはバーストモードとサイクルスチールモードを選択することができます。

表1にDMACの概要を示します。図1にDMACの概念図を示します。

表1 DMACの概要

項目	概要
チャンネル数	8チャンネル (CH0~CH7) CH0~CH2の3チャンネルのみ、外部リクエスト受け付けが可能
アドレス空間	4Gバイト
転送データ長	バイト、ワード (2バイト)、ロングワード (4バイト)、 16バイト (ロングワード × 4)
最大転送回数	16,777,216 (24ビット) 回
アドレスモード	シングルアドレスモード、デュアルアドレスモード
転送要求	オートリクエスト、外部リクエスト、内蔵周辺モジュールリクエスト (SCIF: 2要因, IIC3: 2要因, ADC: 1要因, MTU2: 5要因, CMT: 2要因, USB: 2要因, SSU: 2要因, RCAN: 1要因)
バスモード	サイクルスチールモード、バーストモード
優先順位	チャンネル優先順位固定モード、ラウンドロビンモード
割り込み要求	データ転送 1/2 終了時またはデータ転送終了時にCPUへ割り込み要求発生
外部リクエスト検出	DREQ入力のロー/ハイレベル検出、立ち上がり/立ち下がりエッジ検出
転送要求受け付け信号/ 転送終了信号	DACK/TEND信号のアクティブレベル選択が可能

【注】 DMACについての詳細は、「SH7280 グループ ハードウェアマニュアル ディレクトメモリアクセスコントローラ」の章を参照ください。

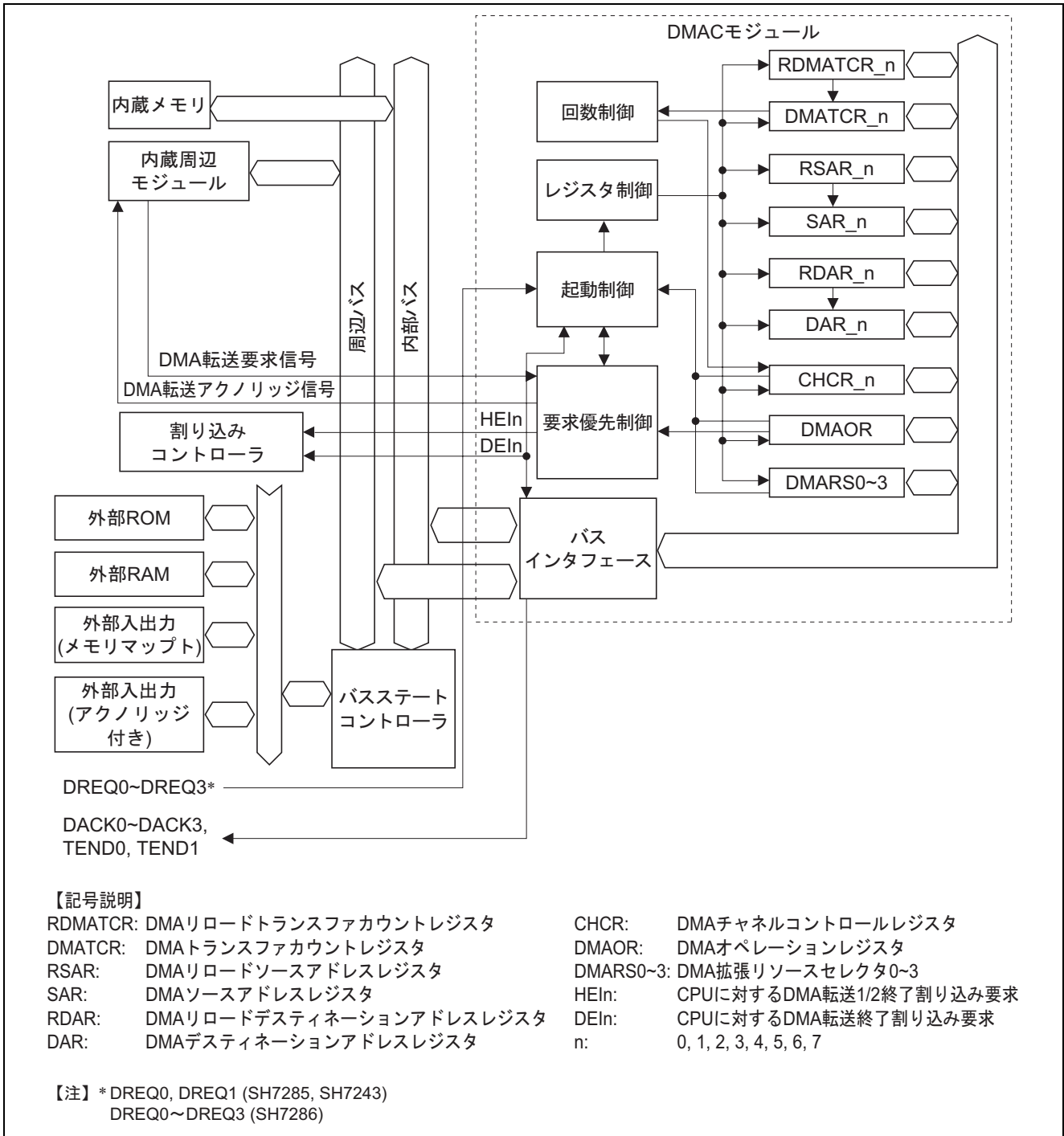


図1 DMAC の概念図

2.2 使用機能の設定手順

ここでは、DMAC によるメモリから内蔵周辺モジュールへのデータ転送を行う場合の初期設定手順について説明します。転送要求は内蔵周辺モジュールリクエストを使用します。図 2 に DMAC の初期設定フロー例を示します。なお、各レジスタ設定の詳細は、「SH7280 グループ ハードウェアマニュアル」を参照ください。

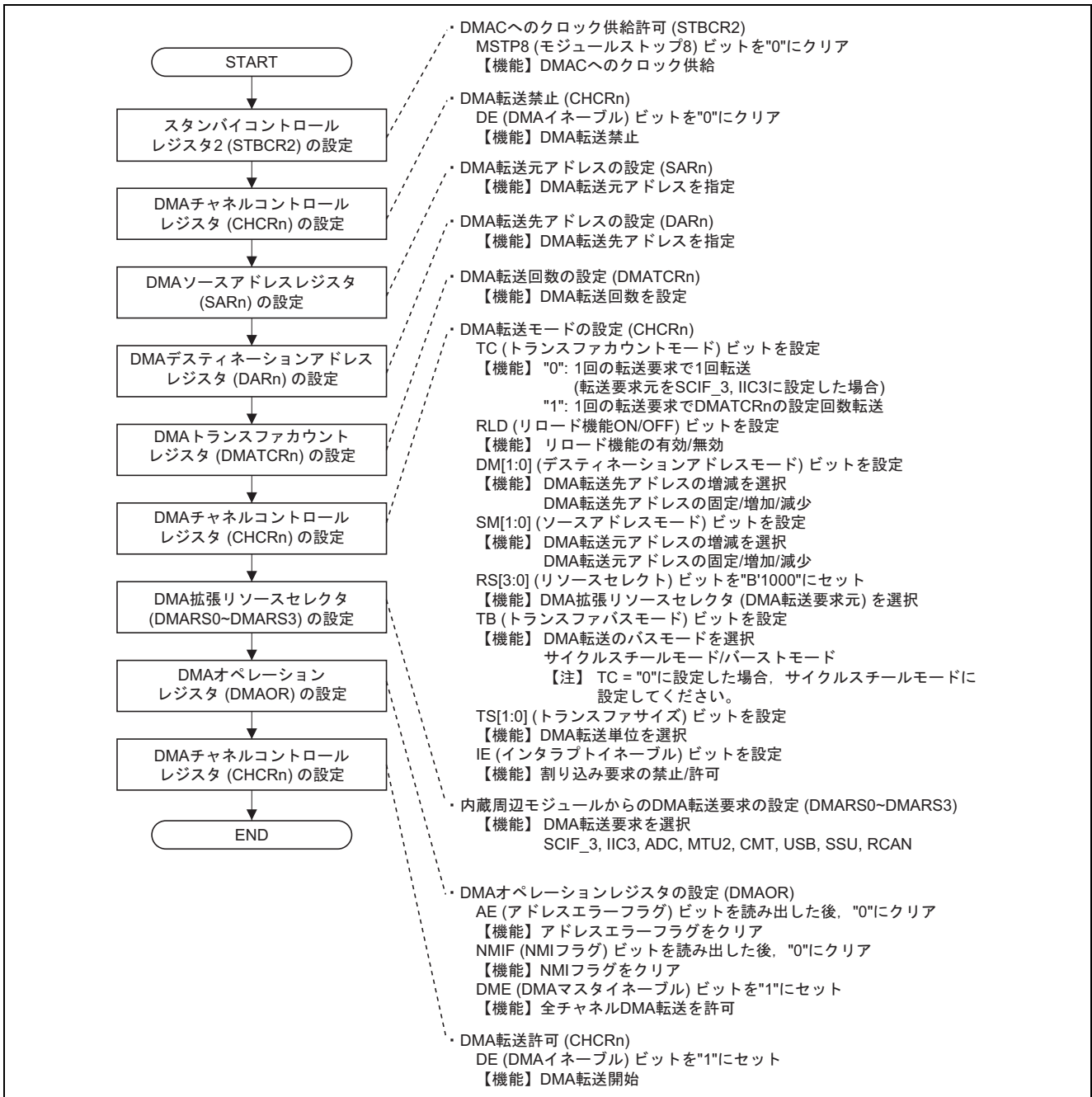


図 2 DMAC の初期設定フロー例

2.3 参考プログラムの動作

参考プログラムでは、SCIF の送信 FIFO データエンpty転送要求 (内蔵周辺モジュールリクエスト) により DMAC チャンネル 1 を起動し、外部メモリから SCIF チャンネル 3 のトランスミット FIFO データレジスタ (SCFTDR) へのデータ転送を行います。SCIF チャンネル 3 の SCFTDR に書き込まれたデータは、UART モードで送信されます。図 3 に参考プログラムの動作タイミングを示します。

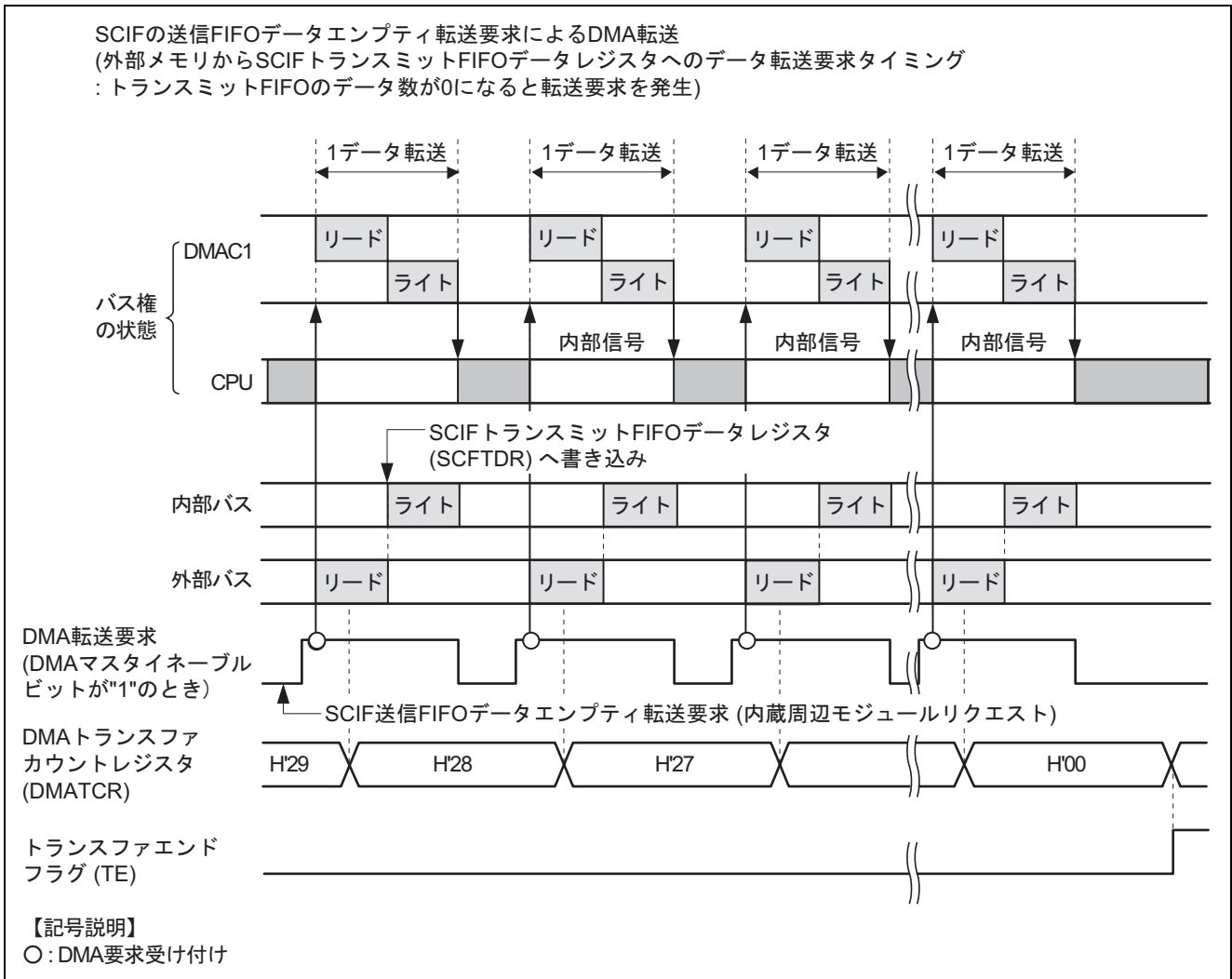


図 3 参考プログラムの動作タイミング



## 2.4 参考プログラムの処理手順

参考プログラムでは、外部メモリに格納した文字列データを SCIF チャンネル 3 のトランスミット FIFO データレジスタ (SCFTDR) に DMA 転送し、UART モードで送信します。

表 2 に参考プログラムのレジスタ設定を示します。また、表 3 に参考プログラムで使用しているマクロ定義を示します。図 4 に参考プログラムの処理フローを示します。

表 2 参考プログラムのレジスタ設定

レジスタ名	アドレス	設定値	機能
スタンバイコントロール レジスタ 2 (STBCR2)	H'FFFE 0018	H'00	MSTP8 = "0": DMAC は動作
DMA チャンネルコントロール レジスタ_1 (CHCR_1)	H'FFFE 101C	H'0000 0000	DE = "0": DMA 転送禁止
		H'0000 1800	TC = "0": 1 回の DMA 要求で 1 回転送 RLD = "0": リロード機能無効 DM = "B'00": デスティネーション アドレス固定 SM = "B'01": ソースアドレス増加 RS = "B'1000": 拡張リソースセクタ TB = "0": サイクルスチールモード TS = "B'00": バイト転送 IE = "0": 割り込み要求禁止
DMA ソースアドレス レジスタ_1 (SAR_1)	H'FFFE 1010	H'0000 1801	DE = "1": DMA 転送許可
		文字列データの 格納アドレス	転送元の開始アドレス: 外部メモリに格納した文字列の先頭 アドレス
DMA デスティネーション アドレスレジスタ_1 (DAR_1)	H'FFFE 1014	H'FFFE 980C	転送先の開始アドレス: SCIF トランスミット FIFO データ レジスタ_3 (SCFTDR_3) アドレス
DMA トランスファカウンタ レジスタ_1 (DMATCR_1)	H'FFFE 1018	文字列 データ数	転送回数: 文字列データ数
DMA オペレーション レジスタ (DMAOR)	H'FFFE 1200	H'0001	DME = "1": 全チャンネル DMA 転送許可
DMA 拡張リソースセクタ 0 (DMARS0)	H'FFFE 1300	H'8D00	MID = "B'100011" RID = "B'01": SCIF_3 の送信 FIFO データエンプティ転送 要求に設定

表 3 参考プログラムで使用しているマクロ定義

マクロ定義	設定値	機能
DMA_SIZE_BYTE	H'0000	バイト転送
DMA_SIZE_WORD	H'0001	ワード転送
DMA_SIZE_LONG	H'0002	ロングワード転送
DMA_SIZE_LONGx4	H'0003	16 バイト転送
DMA_INT_DISABLE	H'0000	DMA 転送完了割り込み未使用
DMA_INT_ENABLE	H'0010	DMA 転送完了割り込み使用

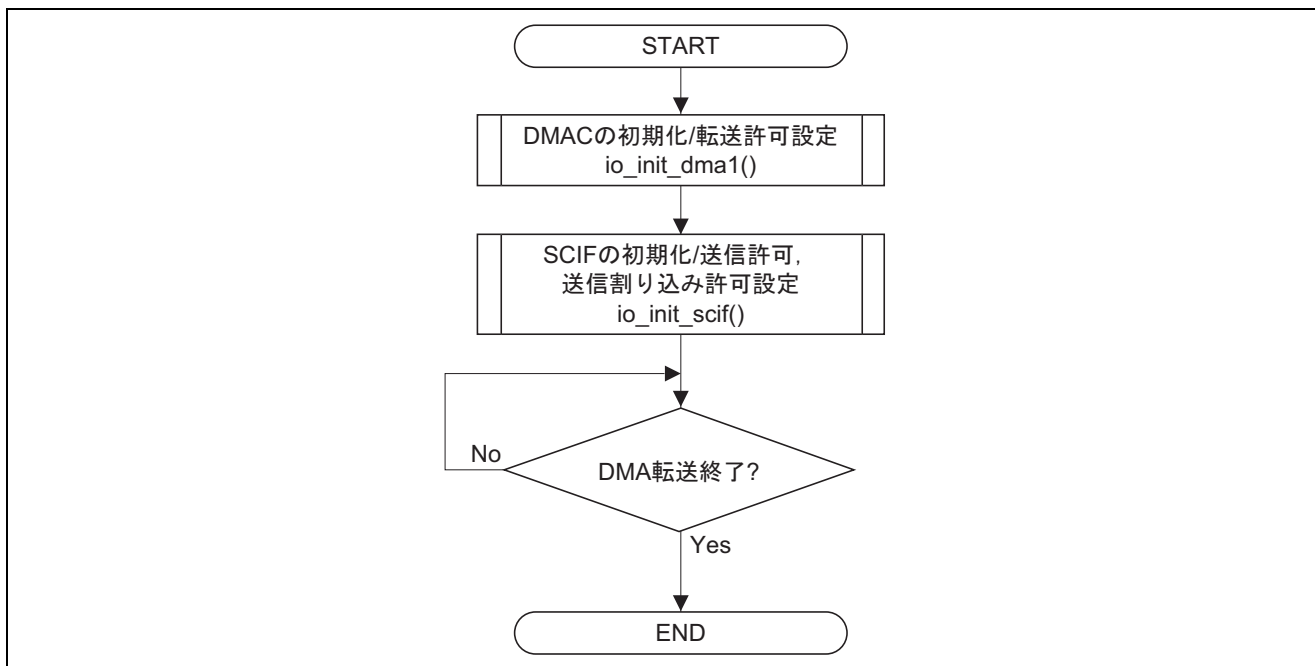


図 4 参考プログラムの処理フロー

### 3. 参考プログラム例

```

1  /*"FILE COMMENT"*****
2  *
3  *   System Name : SH7285 Sample Program
4  *   File Name   : main.c
5  *   Contents    : DMAC 内蔵モジュールへの転送
6  *   Version     : 1.00.00
7  *   Model      : M3A-HS85
8  *   CPU        : SH7285
9  *   Compiler   : SHC9.1.1.0
10 *   note       : DMAC を使用した SCIF 送信データ転送を行いません。
11 *
12 *   The information described here may contain technical inaccuracies or
13 *   typographical errors. Renesas Technology Corporation and Renesas Solutions
14 *   assume no responsibility for any damage, liability, or other loss rising
15 *   from these inaccuracies or errors.
16 *
17 *   Copyright (C) 2008 Renesas Technology Corp. All Rights Reserved
18 *   AND Renesas Solutions Corp. All Rights Reserved
19 *
20 *   history    : 2008.02.26 ver.1.00.00
21 *"FILE COMMENT END"*****/
22 #include <string.h>
23 #include "iodefine.h"      /* SH7285 iodefine */
24
25 /* ==== symbol definition ==== */
26 /* ==== DMAC 設定 ==== */
27 #define DMA_SIZE_BYTE 0x0000u
28 #define DMA_SIZE_WORD 0x0001u
29 #define DMA_SIZE_LONG 0x0002u
30 #define DMA_SIZE_LONGx4 0x0003u
31 #define DMA_INT_DISABLE 0x0000u
32 #define DMA_INT_ENABLE 0x0010u
33 #define DMA_INT (DMA_INT_ENABLE >> 4u)
34
35 /* ==== prototype declaration ==== */
36 void main(void);
37 void io_init_dmal(void *src, void *dst, size_t size, unsigned int mode);
38 void io_dmal_stop(void);
39 void io_init_scif(int);
40
41 /* ==== RAM allocation variable declaration ==== */
42 typedef struct {
43     unsigned char scbrr;
44     unsigned short scsmr;
45 } SH_BAUD_SET;
46
47 /* ---- baud rate ---- */
48 enum{
49     CBR_1200,
50     CBR_2400,
51     CBR_4800,
52     CBR_9600,
53     CBR_19200,
54     CBR_31250,
55     CBR_38400,
56     CBR_57600,
57     CBR_115200
58 };
59

```

図 5 サンプルプログラムリスト"main.c" (1)

```

60     static SH_BAUD_SET sci_baud[] = {
61         { 80u, 2u}, /* 1200bps   誤差 0.46% */
62         {162u, 1u}, /* 2400bps   誤差-0.14% */
63         { 80u, 1u}, /* 4800bps   誤差 0.46% */
64         {162u, 0u}, /* 9600bps   誤差-0.14% */
65         { 80u, 0u}, /* 19200bps  誤差 0.46% */
66         { 49u, 0u}, /* 31250bps  誤差 0.00% */
67         { 40u, 0u}, /* 38400bps  誤差-0.75% */
68         { 26u, 0u}, /* 57600bps  誤差 0.46% */
69         { 13u, 0u} /*115200bps 誤差-3.11% */
70     };
71
72     /* Transmission character string */
73     signed char data[] = "SCIF request DMAC Sample Software.\r\n";
74
75     /*"FUNC COMMENT"*****
76     * Outline      : サンプルプログラムメイン
77     *-----
78     * Include     : #include "iodefine.h"
79     *             : #include <machine.h>
80     *-----
81     * Declaration : void main(void);
82     *-----
83     * Function    : サンプルプログラムメイン
84     *-----
85     * Argument    : void
86     *-----
87     * Return Value: void
88     *-----
89     * Notice      :
90     *"FUNC COMMENT END"*****/
91     void main(void)
92     {
93         /* ==== Setting of DMAC ==== */
94         io_init_dmal(data, (void *)&SCIF3.SCFTDR ,sizeof(data),
95             DMA_SIZE_BYTE | DMA_INT_DISABLE);
96             /* Transfer requests : SCIF3 transmitter */
97             /* RAM -> SCIF transmitter */
98
99         /* ==== Setting of SCIF ==== */
100        io_init_scif(CBR_57600);
101            /* UART mode */
102            /* bit rate : 57600bps */
103
104        /* ==== DMA start ==== */
105        DMAC1.CHCR.BIT.DE = 1u1; /* DMA enable */
106
107        /* ==== DMA stop ==== */
108        io_dmal_stop();
109
110        while(1){
111            /* Program end */
112        }
113    }
114

```

図 6 サンプルプログラムリスト"main.c" (2)

```

115  /*"FUNC COMMENT"*****
116  * Outline      : DMAC 転送初期設定
117  *-----
118  * Include      : #include "iodefine.h"
119  *-----
120  * Declaration  : void io_init_dmal(void *src, void *dst, size_t size, unsigned int mode);
121  *-----
122  * Function     : ソースアドレス src からディスティネーションアドレス dst へ
123  *               : size バイト分 DMAC による転送を行います。
124  *               : 転送は SCIF3 からの要求により実行します。
125  *               : mode には、転送サイズ、割り込みの使用/未使用を指定します。
126  *-----
127  * Argument    : void *src   : ソースアドレス
128  *               : void *dst   : ディスティネーションアドレス
129  *               : size_t size : 転送サイズ (バイト)
130  *               : unsigned int mode : 転送モード、以下のモードを論理和で指定します。
131  *               :   DMA_SIZE_BYTE (0x0000) バイト転送
132  *               :   DMA_SIZE_WORD (0x0001) ワード転送
133  *               :   DMA_SIZE_LONG (0x0002) ロング転送
134  *               :   DMA_SIZE_LONGx4(0x0003) 16 バイト転送
135  *               :   DMA_INT_DISABLE(0x0000) DMA 転送完了割り込み未使用
136  *               :   DMA_INT_ENABLE (0x0010) DMA 転送完了割り込み使用
137  *-----
138  * Return Value: void
139  *-----
140  * Notice      : 転送サイズとソース/ディスティネーションアドレスの
141  *               : アライメントが一致しない場合の動作は保証しません。
142  *               : 割り込み使用時は、割り込みルーチンを登録する必要があります
143  *"FUNC COMMENT END"*****/
144  void io_init_dmal(void *src, void *dst, size_t size, unsigned int mode)
145  {
146      unsigned int ts;
147      unsigned long ie;
148
149      ts = mode & 0x3u;
150      ie = (mode & 0x00f0u ) >> 4u;
151
152      /* ==== Setting of power down mode ==== */
153      STB.CR2.BIT._DMAC = 0x0;          /* Clear the DMAC module standby mode */
154
155      /* ---- DMA Channel Control Registers(CHCR) ---- */
156      DMAC1.CHCR.BIT.DE = 0u1;          /* DMA disable */
157
158      /* ---- DMA Source Address Registers(SAR) ---- */
159      DMAC1.SAR = (void *)src;
160
161      /* ---- DMA Destination Address Registers(DAR) ---- */
162      DMAC1.DAR = (void *)dst;
163

```

図7 サンプルプログラムリスト"main.c" (3)

```

164     /* ---- DMA Transfer Count Registers(DMATCR) ---- */
165     switch(ts){
166     case DMA_SIZE_BYTE:
167         DMAC1.DMATCR = size;           /* 転送回数(1/1)を設定 */
168         DMAC1.RDMATCR = size;
169         break;
170     case DMA_SIZE_WORD:
171         DMAC1.DMATCR = size >> 1u;    /* 転送回数(1/2)を設定 */
172         DMAC1.RDMATCR = size >> 1u;
173         break;
174     case DMA_SIZE_LONG:
175         DMAC1.DMATCR = size >> 2u;    /* 転送回数(1/4)を設定 */
176         DMAC1.RDMATCR = size >> 2u;
177         break;
178     case DMA_SIZE_LONGx4:
179         DMAC1.DMATCR = size >> 4u;    /* 転送回数(1/16)を設定 */
180         DMAC1.RDMATCR = size >> 4u;
181         break;
182     default:
183         break;
184     }
185     /* ---- DMA Channel Control Registers(CHCR) ---- */
186     DMAC1.CHCR.LONG = 0x00001800ul | (ts << 3u) | (ie << 2u) ;
187                                     /* Fixed destination address      */
188                                     /* Source address is incremented    */
189                                     /* DMA extension resource selector */
190                                     /* Cycle steal mode                 */
191                                     /* Transfer Size : Byte unit        */
192     /* ---- DMA Extension Resource Selectors 0(DMARS0) ---- */
193     DMAC.DMARS0.BIT.C1MID = 0x23u; /* Transfer requests : SCIF3 transmitter */
194     DMAC.DMARS0.BIT.C1RID = 0x1;
195
196     /* ---- DMA Operation Register(DMAOR) ---- */
197     DMAC.DMAOR.WORD &= 0xffff9u; /* AE,NMIF clear */
198
199     if(DMAC.DMAOR.BIT.DME == 0ul){ /* DMA Master Enable */
200         DMAC.DMAOR.BIT.DME = 1ul;
201     }
202 }
203 /*"FUNC COMMENT"*****
204 * Outline      : DMAC 停止
205 *-----
206 * Include      : #include "iodefine.h"
207 *-----
208 * Declaration  : void io_dmal_stop(void);
209 *-----
210 * Function     : DMA 転送を停止します。
211 *-----
212 * Argument    : void
213 *-----
214 * Return Value: void
215 *-----
216 * Notice      :
217 *"FUNC COMMENT END"*****/
218 void io_dmal_stop(void)
219 {
220     /* Transmission end detection */;
221     while(DMAC1.CHCR.BIT.TE == 0ul){
222         /* wait TE bit set */
223     }
224     /* ---- DMA end ---- */
225     DMAC1.CHCR.BIT.DE = 0ul; /* DMA disable */
226 }
227

```

図 8 サンプルプログラムリスト"main.c" (4)

```

228 /*"FUNC COMMENT"*****
229 * Outline      : SCIF 設定
230 *-----
231 * Include      : #include "iodefine.h"
232 *-----
233 * Declaration  : void io_init_scif(void);
234 *-----
235 * Function     : FIFO 内蔵シリアルコミュニケーションインタフェース(SCIF)の設定
236 *-----
237 * Argument    : int bps : ボーレート指定値
238 *-----
239 * Return Value: void
240 *-----
241 * Notice      :
242 *"FUNC COMMENT END"*****/
243 void io_init_scif(int bps)
244 {
245     /* ==== Setting of power down mode ==== */
246     STB.CR4.BIT._SCIF3 = 0u;      /* Clear the SCIF3 module standby mode */
247
248     /* ==== Setting of SCIF ==== */
249     /* ---- Serial Control Register(SCSCR) ---- */
250     SCIF3.SCSCR.WORD &= 0x00u;    /* Transmitter/Receiver disabled */
251     SCIF3.SCSCR.BIT.CKE = 0x0u;  /* Internal clock */
252
253     /* ---- Serial Mode Register(SCSMR) ---- */
254     SCIF3.SCSMR.WORD = sci_baud[bps].scsmr;
255                                 /* Asynchronous mode */
256                                 /* 8-bit data */
257                                 /* Parity bit not added or checked*/
258                                 /* One stop bit */
259
260     /* ---- Bit Rate Register(SCBRR) ---- */
261     SCIF3.SCBRR = sci_baud[bps].scbrr;
262
263     /* ==== Setting of PFC ==== */
264     /* ---- port E control register L2 ---- */
265     PFC.PECRL2.BIT.PE5MD = 5u; /* Set TxD3 */
266
267     /* ---- Serial Control Register(SCSCR) ---- */
268     SCIF3.SCSCR.BIT.TIE = 1u; /* Transmit interrupt enabled */
269     SCIF3.SCSCR.BIT.TE = 1u; /* Transmitter enabled */}
270
271 /* End of File */
272

```

図9 サンプルプログラムリスト"main.c" (5)

#### 4. 参考ドキュメント

- ソフトウェアマニュアル  
SH-2A, SH2A-FPU ソフトウェアマニュアル  
(最新版をルネサス テクノロジホームページから入手してください)
- ハードウェアマニュアル  
SH7280 グループ ハードウェアマニュアル  
(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)



## ホームページとサポート窓口

ルネサス テクノロジホームページ

<http://japan.renesas.com/>

お問合せ先

<http://japan.renesas.com/inquiry>[csc@renesas.com](mailto:csc@renesas.com)

## 改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2008.10.31	—	初版発行

すべての商標および登録商標は、それぞれの所有者に帰属します。

### 本資料ご利用に際しての留意事項

1. 本資料は、お客様に用途に応じた適切な弊社製品をご購入いただくための参考資料であり、本資料中に記載の技術情報について弊社または第三者の知的財産権その他の権利の実施、使用を許諾または保証するものではありません。
2. 本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他応用回路例など全ての情報の使用に起因する損害、第三者の知的財産権その他の権利に対する侵害に関し、弊社は責任を負いません。
3. 本資料に記載の製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的、あるいはその他軍事用途の目的で使用しないでください。また、輸出に際しては、「外国為替および外国貿易法」その他輸出関連法令を遵守し、それらの定めるところにより必要な手続を行ってください。
4. 本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他応用回路例などの全ての情報は本資料発行時点のものであり、弊社は本資料に記載した製品または仕様等を予告なしに変更することがあります。弊社の半導体製品のご購入およびご使用に当たりますは、事前に弊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、弊社ホームページ (<http://www.renesas.com>) などを通じて公開される情報に常にご注意ください。
5. 本資料に記載した情報は、正確を期すため慎重に制作したものです。万一本資料の記述の誤りに起因する損害がお客様に生じた場合においても、弊社はその責任を負いません。
6. 本資料に記載の製品データ、図、表などに示す技術的な内容、プログラム、アルゴリズムその他応用回路例などの情報を流用する場合は、流用する情報を単独で評価するだけでなく、システム全体で十分に評価し、お客様の責任において適用可否を判断してください。弊社は、適用可否に対する責任を負いません。
7. 本資料に記載された製品は、各種安全装置や運輸・交通用、医療用、燃焼制御用、航空宇宙用、原子力、海底中継用の機器・システムなど、その故障や誤動作が直接人命を脅かしあるいは人体に危害を及ぼすおそれのあるような機器・システムや特に高度な品質・信頼性が要求される機器・システムでの使用を意図して設計、製造されたものではありません（弊社が自動車用と指定する製品を自動車に使用する場合を除きます）。これらの用途に利用されることをご検討の際には、必ず事前に弊社営業窓口へご照会ください。なお、上記用途に使用されたことにより発生した損害等については弊社はその責任を負いかねますのでご了承願います。
8. 第7項にかかわらず、本資料に記載された製品は、下記の用途には使用しないでください。これらの用途に使用されたことにより発生した損害等につきましては、弊社は一切の責任を負いません。
  - 1) 生命維持装置。
  - 2) 人体に埋め込み使用するもの。
  - 3) 治療行為（患部切り出し、薬剤投与等）を行うもの。
  - 4) その他、直接人命に影響を与えるもの。
9. 本資料に記載された製品のご使用につき、特に最大定格、動作電源電圧範囲、放熱特性、実装条件およびその他諸条件につきましては、弊社保証範囲内でご使用ください。弊社保証値を越えて製品をご使用された場合の故障および事故につきましては、弊社はその責任を負いません。
10. 弊社は製品の品質および信頼性の向上に努めておりますが、特に半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。弊社製品の故障または誤動作が生じた場合も人身事故、火災事故、社会的損害などを生じさせないよう、お客様の責任において冗長設計、延焼対策設計、誤動作防止設計などの安全設計（含むハードウェアおよびソフトウェア）およびエージング処理等、機器またはシステムとしての出荷保証をお願いいたします。特にマイコンソフトウェアは、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。
11. 本資料に記載の製品は、これを搭載した製品から剥がれた場合、幼児が口に入れて誤飲する等の事故の危険性があります。お客様の製品への実装後に容易に本製品が剥がれることがなきよう、お客様の責任において十分な安全設計をお願いします。お客様の製品から剥がれた場合の事故につきましては、弊社はその責任を負いません。
12. 本資料の全部または一部を弊社の文書による事前の承諾なしに転載または複製することを固くお断りいたします。
13. 本資料に関する詳細についてのお問い合わせ、その他お気付きの点等がございましたら弊社営業窓口までご照会ください。

D039444