

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。

M16C/29 グループ

CAN デモセットプログラム

1. 要約

本資料は、M16C/29 グループの CAN デモセットのプログラムについて説明しています。

CAN プロトコルにつきましては、CAN 仕様書(Robert Bosch GmbH)および ISO11898 をご参照ください。

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 : M16C/29 グループ

M16C/29 グループと同様の SFR(周辺機能制御レジスタ)を持つ他の M16C ファミリでも本プログラムを使用することができます。ただし、一部の機能を機能追加等で変更している場合がありますのでマニュアルで確認してください。このアプリケーションノートをご使用に際しては十分な評価を行ってください。

なお、本サンプルプログラムは以下の条件で動作します。

- ・ 電源電圧 5V
- ・ メインクロック入力発振周波数 20MHz
- ・ M16C/29 スタートキット (M3A-0284)(但し、下記デバイスの追加が必要です。)
 - 追加デバイス : CAN トランシーバ (HA13721)
 - コンデンサ (スタートキット回路図上の C12、C13、C14)
 - 抵抗 (スタートキット回路図上の R8。終端抵抗。)

3. サンプルプログラム仕様概略

本サンプルプログラムは CAN 通信を行うプログラム例です。

M16C/29 スタートキット 2 台 (ボード 、ボード) を使用し、CAN 送受信を行います。

ボード :

SW 3 , SW 4 によって送信 ID、受信 ID を変更可能です。

可変抵抗 (VR) SW 2 を操作することによって CAN 送信を行います。

受信データによって LED 表示パターンを変更します。

ボード :

SW 2 ~ 4 の押下げで CAN 送信を行います。

受信データの 1byte 目を LED に表示します。

図 1 に本プログラムを使用した際の接続図を示します。また、図 2 に回路図を示します。

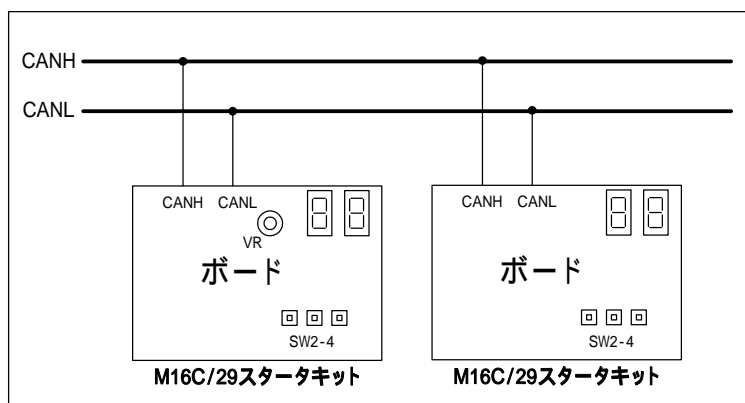


図 1 : 接続図

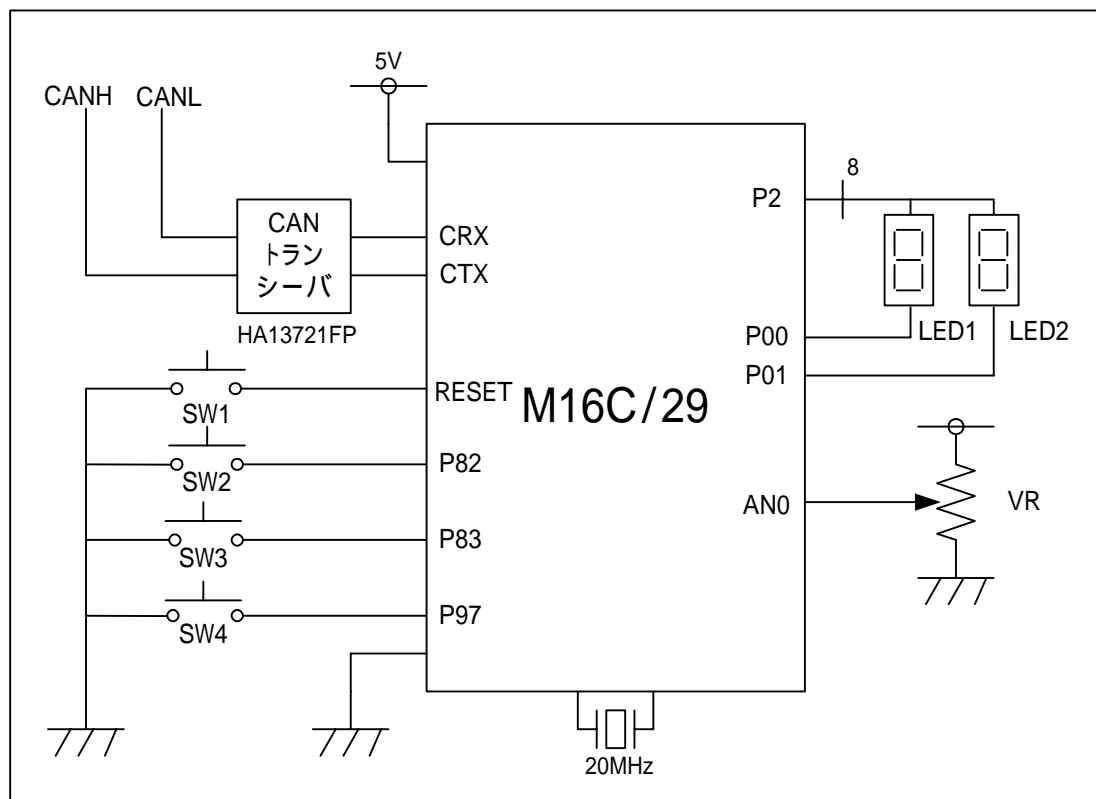


図 2 : ブロック図

使用端子を表 1 に、スイッチ (SW) の機能を表 2 にそれぞれ示します。

表 1：使用端子

端子名	入出力	機能
P00、P01	出力	LED 出力
P20 ~ P27	出力	LED 出力
P92 (CRx)	入力	CANout
P93 (CTx)	出力	CANin
P82	入力	スイッチ 2 入力
P83	入力	スイッチ 3 入力
P84	入力	スイッチ 4 入力
P100	入力	可変抵抗 ボード でのみ使用

表 2：スイッチ機能

スイッチ 名称	SW1	SW2	SW3	SW4
ボード	ハードウェア リセット	CAN 送信	受信 ID 表示 (SW4 押下げ中は 送信 ID + 1)	送信 ID 表示 (SW3 押下げ中は 受信 ID + 1)
ボード	ハードウェア リセット	CAN 送信 (DATA1)	CAN 送信 (DATA2)	CAN 送信 (DATA3)

4. サンプルプログラム仕様詳細

本サンプルプログラムは、各ボードのスイッチ、可変抵抗を操作することによって CAN 送信を行います。CAN 通信は一度に 0 ~ 8 byte のデータを送信可能ですが、本プログラムでは常に 1byte のデータを送信します。両ボードはデータを受信すると、その受信データにあわせて 7 セグメント LED の点灯パターンを変更します。

4.1 スイッチ機能

4.1.1 ボード

SW1 : リセット。

SW2 : CAN 送信を行います。

送信データは “ 01₁₆ ”、“ 02₁₆ ”、“ 03₁₆ ” の 3 種類で、SW2 を押すごとに順に送信されます。

SW3 : 受信 ID を表示します。

押下げている間、受信 ID を表示します。

表示中にSW4 を押下げると受信IDを + 1 します。設定できる受信IDは 00₁₆ ~ FF₁₆です。

SW4 : 送信 ID を表示します。

押下げている間、送信 ID を表示します。

表示中にSW3 を押下げると送信IDを + 1 します。設定できる送信IDは 00₁₆ ~ FF₁₆です。

4.1.2 ボード

SW1 : リセット。

SW2 : CAN 送信を行います。

データ “ 01₁₆ ” を送信します。

SW3 : CAN 送信を行います。

データ “ 02₁₆ ” を送信します。

SW4 : CAN 送信を行います。

データ “ 03₁₆ ” を送信します。

4.2 LED 表示

4.2.1 ボード

受信データの 1byte 目によって LED の表示パターンを変更します。また、SW3 を押下げると受信 ID を、SW4 を押下げると送信 ID をそれぞれ 16 進数で表示します。

受信データによる LED 表示パターンは 3 種類あります。表 3、図 3 に LED 表示パターンを示します。

表 3 : LED 表示パターン

	受信データ	LED1	LED2
表示パターン 1	01	反時計回り	時計回り
表示パターン 2	02	時計回り	反時計回り
表示パターン 3	03	両 LED を使って時計回り	

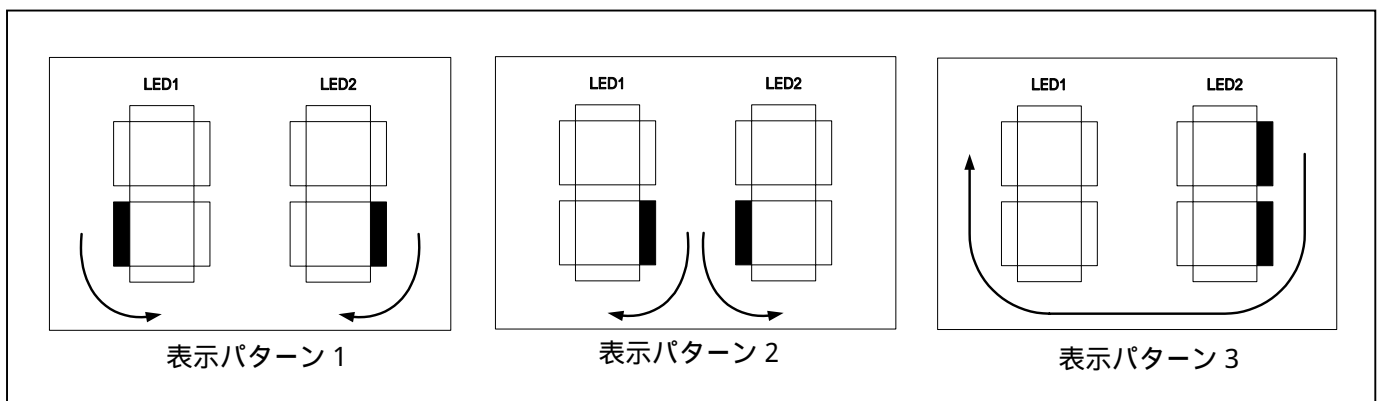


図 3 : 表示パターン

4.2.2 ボード

受信データを LED に表示します。(16 進数)

4.3 CAN 設定

ボード、ボードのCANに関連する各設定は、送信ID、受信IDを除いて同じになっています。

- ・ スロット 0：送信スロット
- ・ スロット 14：受信スロット
- ・ スロット 15：受信スロット
- ・ Basic CAN モード
- ・ CAN0 受信割り込み許可
- ・ CAN0 送信割り込み禁止
- ・ CAN0 エラー割り込み禁止
- ・ DLC (送信データ長)：1 (byte)
- ・ ボーレート：500kbps
- ・ ビットタイミング：
 - PTS = 3Tq
 - PBS1 = 3Tq
 - PBS2 = 3Tq
 - SJW = 1Tq

レセシブからドミナントへのエッジがSSの間に来た場合のビットタイミングを図4に示します。

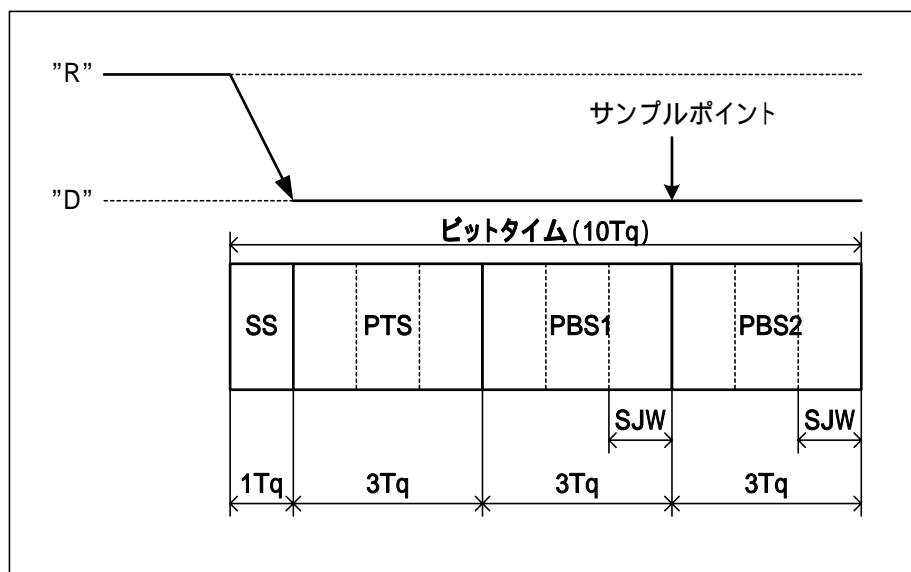


図 4: ビットタイミング図

ボード 送信ID：00₁₆ ~ FF₁₆ (初期値 02₁₆)

ボード 受信ID：00₁₆ ~ FF₁₆ (初期値 01₁₆)

ボード 送信ID：01₁₆

ボード 受信ID：02₁₆

4.4 A/D 変換

4.4.1 ボード

可変抵抗 (VR) からの入力を A/D 変換し、その結果 3 回の平均値 (2ms 周期で読み込み) に 5 以上の変化があった場合、その平均値を CAN で送信します。A/D 変換は 10 ビットモードで行いますが、送信データは上位 8 ビットを送信します。

A/D 変換設定

- ・ 繰り返しモード
- ・ 10 ビットモード
- ・ A/D 変換周波数 : fAD の 4 分周 (5MHz)
- ・ 使用端子 : P10_0

4.4.2 ボード

ボード は A/D 変換を使用しません。

4.5 エラー処理

ボード、ボード はそれぞれ、エラーが発生すると、CAN 通信、A/D 変換を停止し、エラー番号を LED に表示します。

LED1 に “E” を表示し、LED2 にエラー番号を表示します。また、両 LED を点滅させます。

ボード のエラーを表 4 に、ボード のエラーを表 5 にそれぞれ示します。

表 4：ボード エラー

エラー番号	エラー内容
1	CAN 受信バッファオーバーランエラー
2	内部エラー1
3	内部エラー2

表 5：ボード エラー

エラー番号	エラー内容
1	CAN 受信バッファオーバーランエラー
2	内部エラー

5. フローチャート

本章ではサンプルプログラムフローチャートを示します。
本章で示すフローチャートにおいて、■で表記された関数は M16C/6N、M16C/1N、M16C/29 グループの「CAN アプリケーションノート」に記載されている関数です。

5.1 プログラム フローチャート (ボード)

図 5 にメイン関数のフローを示します。

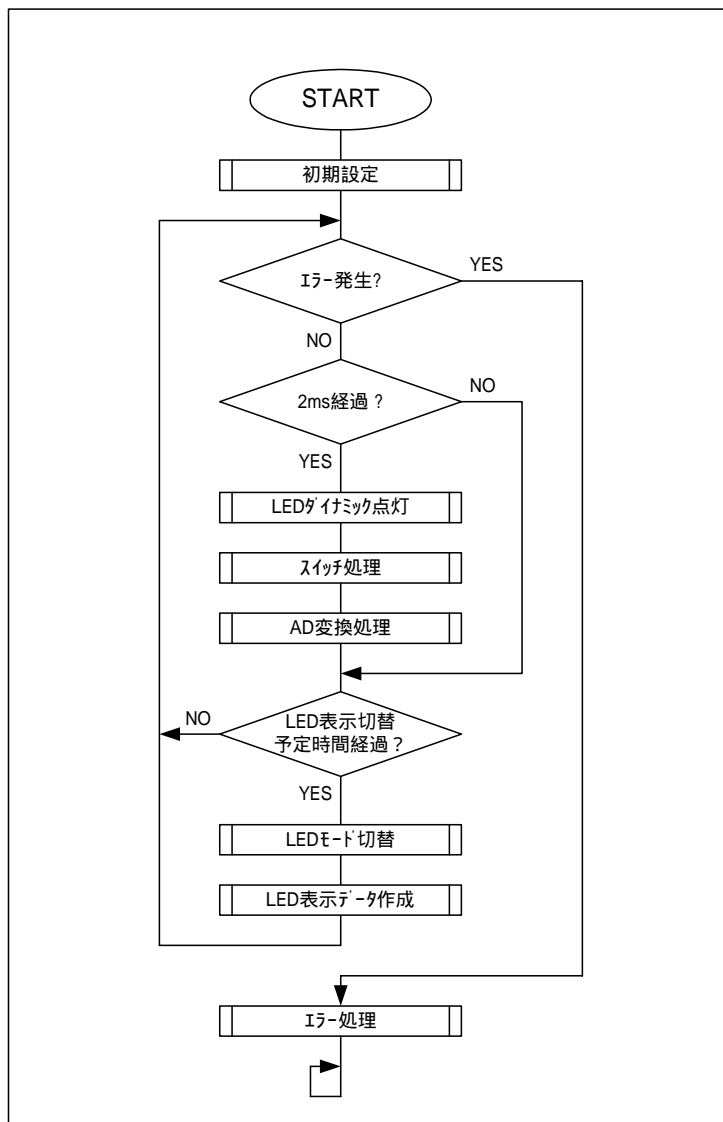


図 5 : main 関数

図 6 に初期設定を示します。

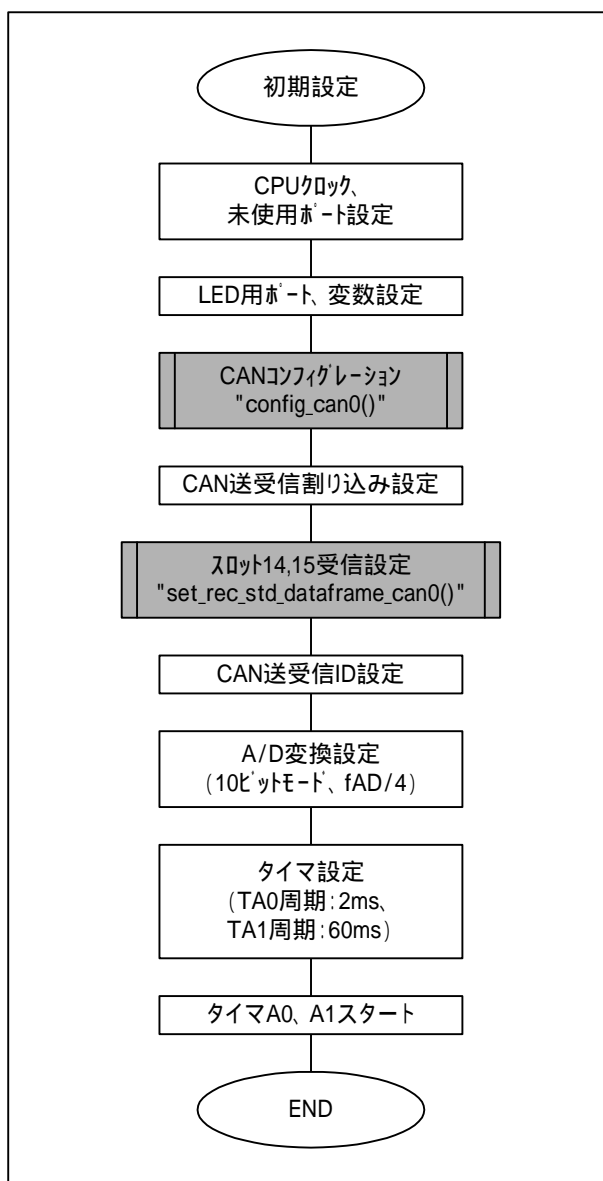


図 6 : 初期設定

図 7、図 8、図 9 にスイッチ 2~4 の立ち下げ処理を示します。

また、図 10 にはスイッチ 3 立ち下げ中にスイッチ 4 立ち下げた処理を示し、図 11 にはスイッチ 4 立ち下げ中にスイッチ 3 立ち下げた処理を示します。全スイッチが“H”の状態（いずれのスイッチも押下げられていない状態）の処理を図 12 に示します。

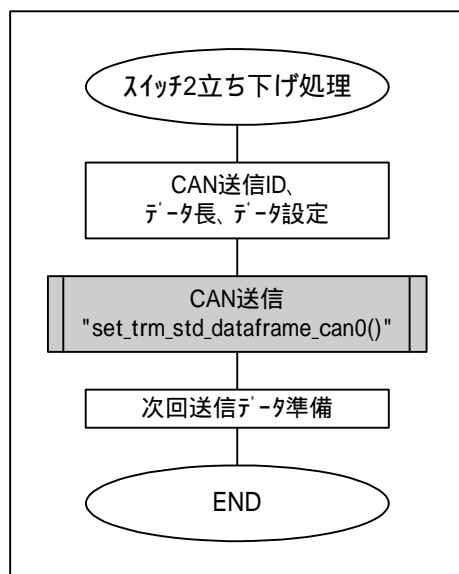


図 7：スイッチ 2 立ち下げ処理

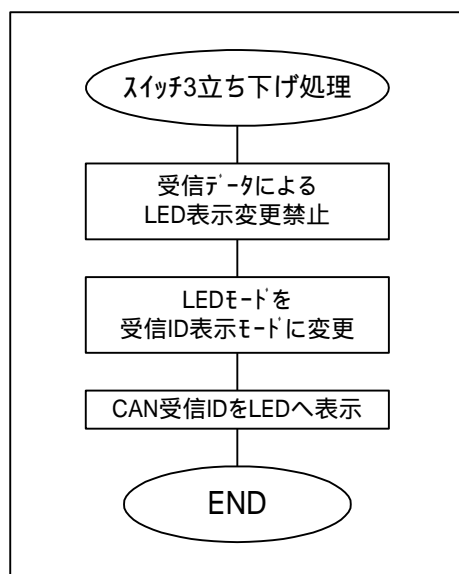


図 8：スイッチ 3 立ち下げ処理

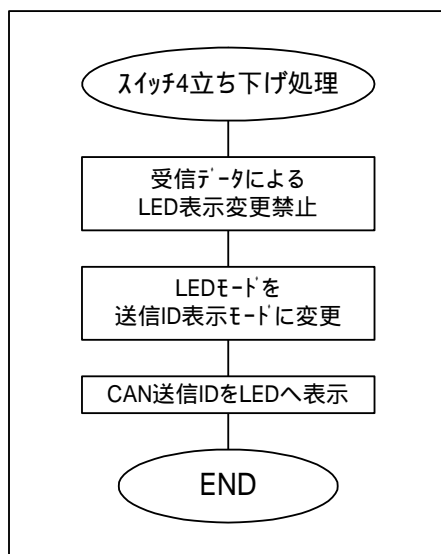


図 9 : スイッチ 4 立ち下げ処理

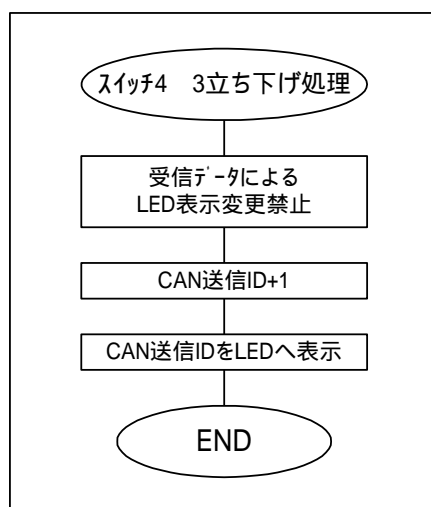


図 10 : スイッチ 4 3 立ち下げ処理

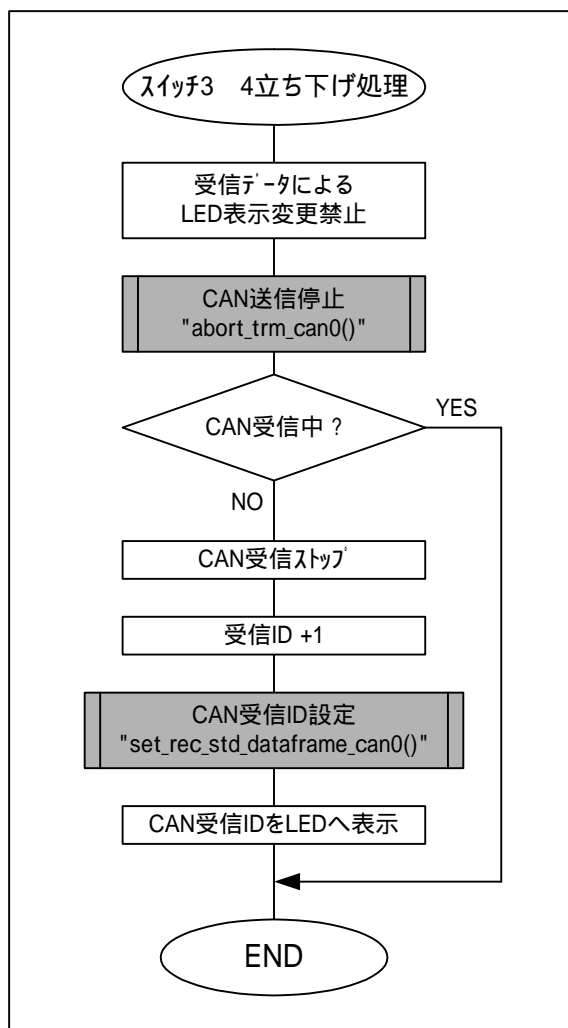


図 11 : スイッチ 3 4 立ち下げ処理

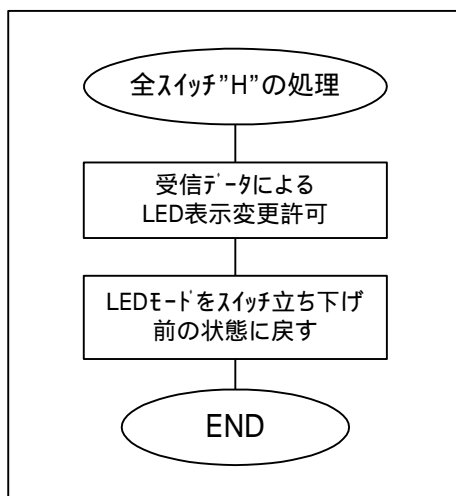


図 12 : 全スイッチ "H" の処理

図 13 に A/D 変換処理を示します。

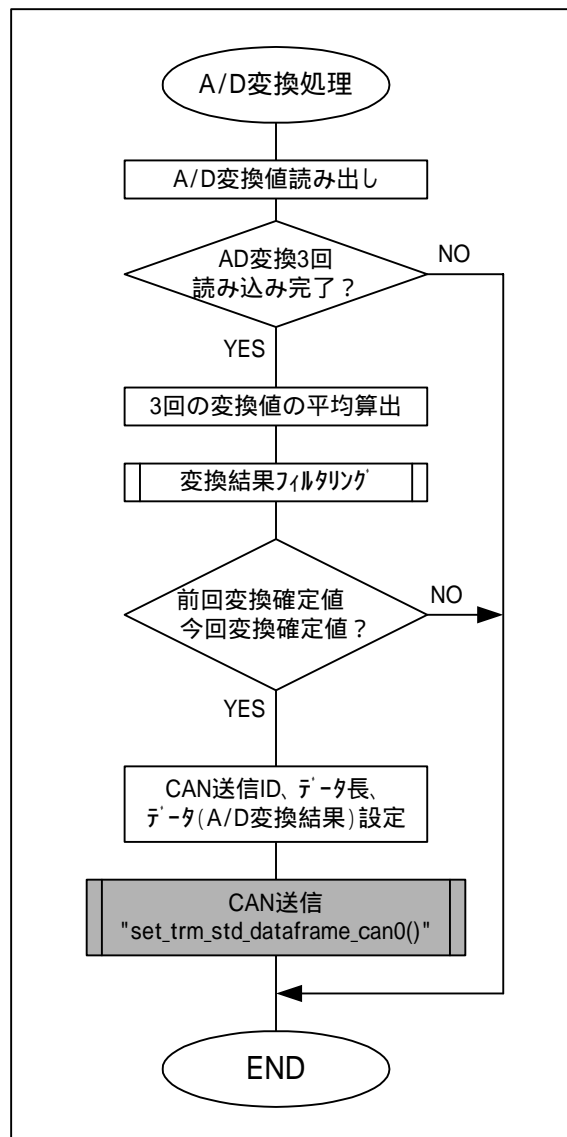


図 13 : A/D 変換処理

図 14 に CAN 受信割り込み処理を示します。

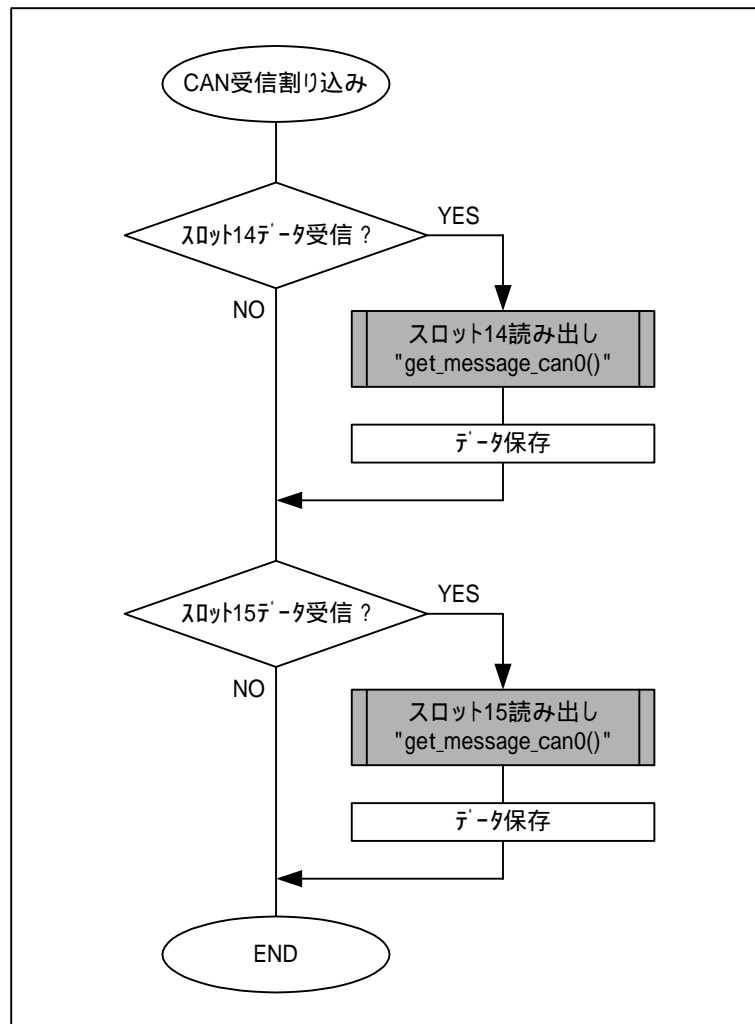


図 14 : CAN 受信割り込み

図 15 にエラー処理を示します。

エラー処理は、LED1 に “E” を表示し、LED2 にエラー番号を表示します。また、0.5 秒毎に両 LED を点滅させます。

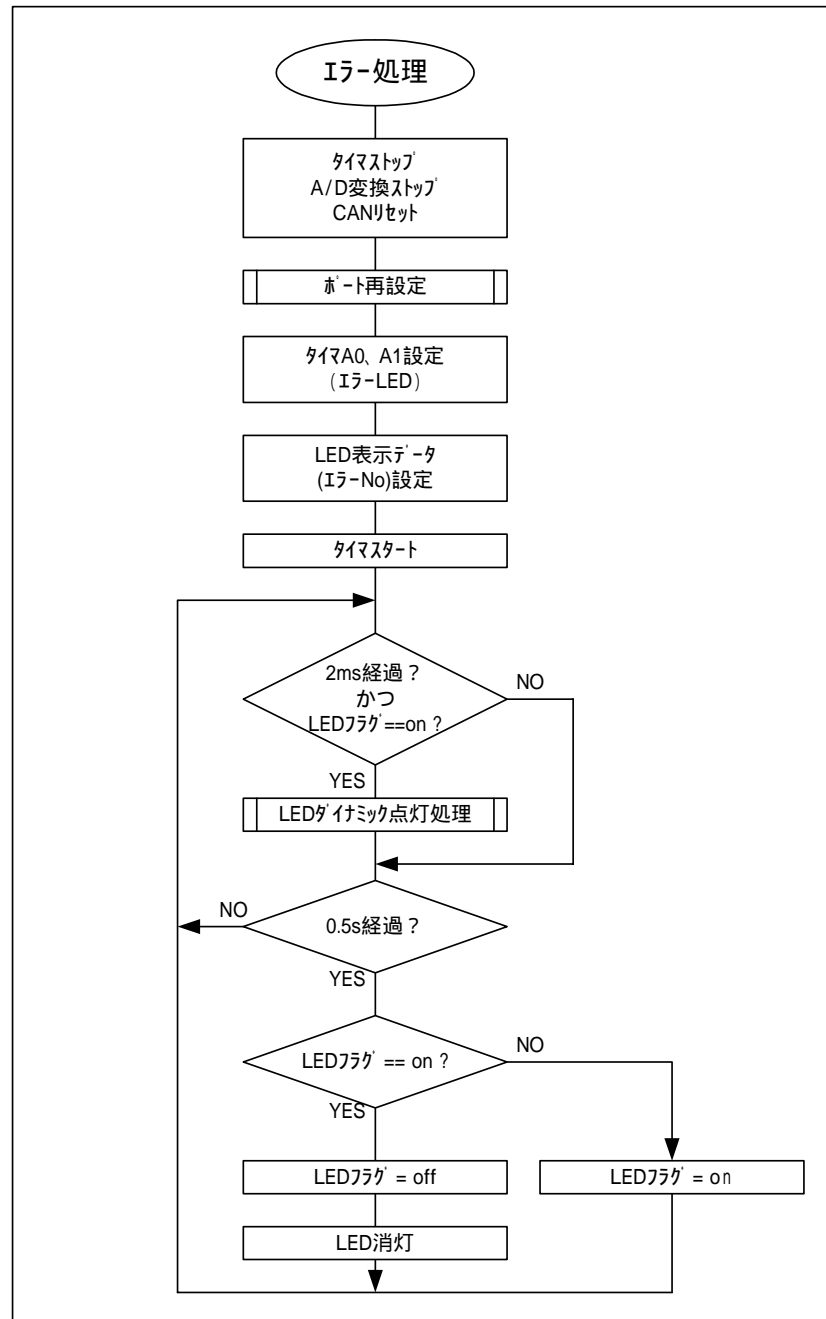


図 15 : エラー処理

プログラム フローチャート (ボード)

図 16 に main 処理を示します。

「初期設定」、「LED ダイナミック点灯」、「スイッチ処理」、「エラー処理」は下記点を除きボード と同じです。

・ボード のプログラムとの相違点

初期設定：A/D 変換に関する初期設定がありません。また、タイマ A0 を使用していません。

スイッチ処理：各スイッチの立ち下げ、立ち上げ処理が異なります。

エラー処理：A/D 変換に関するエラー処理がありません。

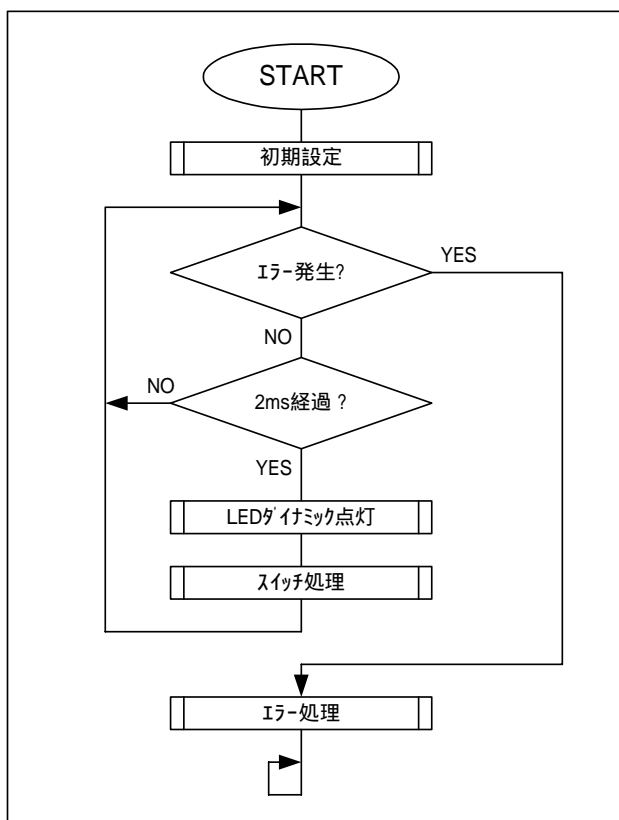


図 16 : main 関数

図 17 にスイッチ 2~4 立ち下げ処理を示します。

ボード のスイッチ 2~4 はいずれも CAN 送信を行います。CAN 送信処理は送信 ID、データ長は同一で、データのみが各スイッチで異なります。立ち上げ処理はありません。

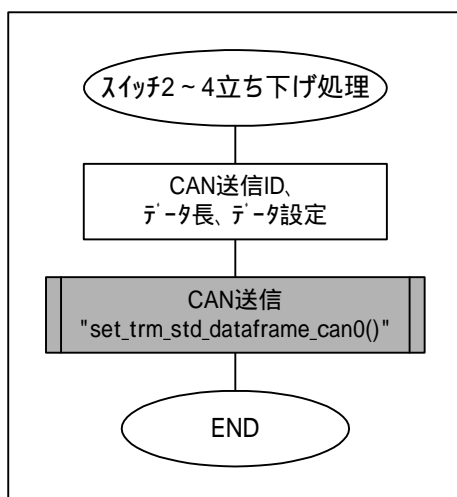


図 17 : スイッチ立ち下げ処理

図 18 に CAN 受信割り込み処理を示します。

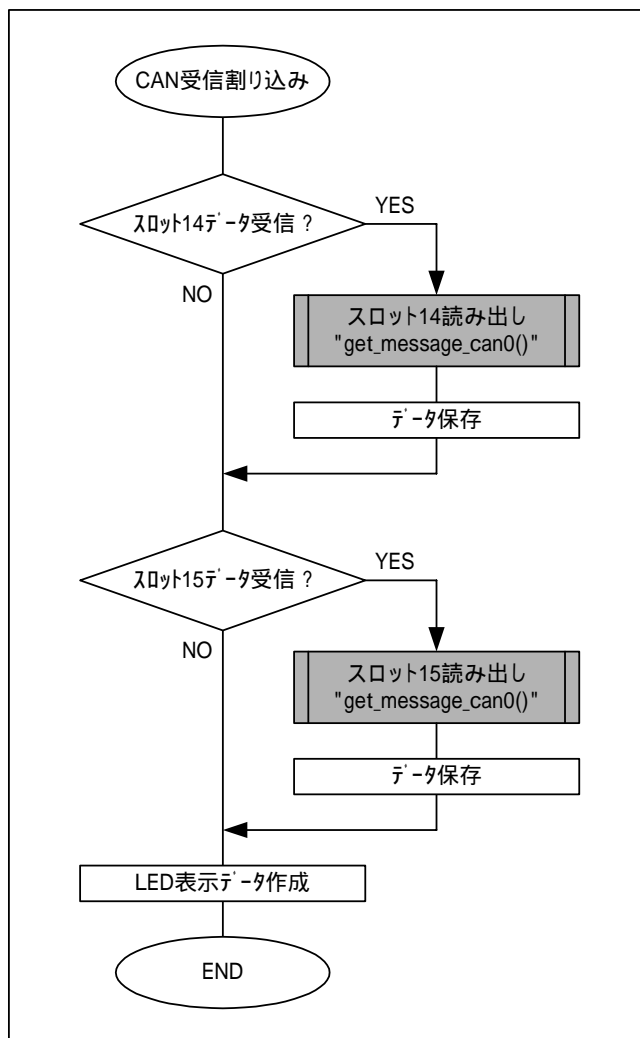


図 18 : CAN 受信割り込み

6. サンプルプログラム

ボード とボード のサンプルプログラムのファイル構成を表 6、表 7 にそれぞれ示します。

表 6：ボード プログラムファイル構成

ファイル名	内容
ncrt0.a30	NC30 スタートアッププログラム
sect30.inc	セクション定義
main1.c	メインプログラム
can1.c	CAN 関連プログラム
ad1.c	A/D 変換関連プログラム
sw_led1.c	スイッチ、LED 関連プログラム
int1.c	割り込み処理プログラム
sfr29.h	M16C/29 用 SFR レジスタヘッダ
define1.h	define 定義ヘッダ
declare1.h	関数、変数宣言ヘッダ

表 7：ボード プログラムファイル構成

ファイル名	内容
ncrt0.a30	NC30 スタートアッププログラム
sect30.inc	セクション定義
main2.c	メインプログラム
can2.c	CAN 関連プログラム
sw_led2.c	スイッチ、LED 関連プログラム
int2.c	割り込み処理プログラム
sfr29.h	M16C/29 用 SFR レジスタヘッダ
define2.h	define 定義ヘッダ
declare2.h	関数、変数宣言ヘッダ

6.1 プログラム

```

/*"FILE COMMENT"*****
*System Name : CAN DEMO PROGRAM1
*File Name   : define1.h
*Version    : 1.00
*Contens    :
*Customer   : RSO
*Model      :
*Order      :
*CPU        :
*Compiler   : NC30 Version 5.30 Release 02
*OS         :
*Programmer :
*Note       :
*****
* Copyright(C)2005, Renesas Technology Corp.
* Copyright(C)2005, Renesas Solutions Corp.
* All rights reserved.
*
*****
*History      2005.07.1   Ver 1.00
**"FILE COMMENT END"*****/

/* MODE */
#define MODE_CLWS      1          //mode num
#define MODE_ANTCLWS  2          //mode num
#define MODE_HALF_CLWS 3          //mode num
#define MODE_RCV_ID   4          //mode num
#define MODE_TRM_ID   5          //mode num

#define DATA_CLWS     1          //receive data pattern
#define DATA_ANTCLWS  2          //receive data pattern
#define DATA_HALF_CLWS 3          //receive data pattern

/* CAN */
#define MASK_14      0xff        //No mask
#define MASK_15      0xff        //

#define TRM_SLOT     0           //CAN0 trm slot

#define CAN_TRM_ID   1           //CAN0 trm initial ID
#define CAN_RCV_ID   2           //CAN0 rcv initial ID

/* AD */
#define AD_COUNT     3           //AD chatter

/* timer */
//TA0

```

```

#define DATA_CLWS_TM      30 -1 //
#define DATA_ANTCLWS_TM  30 -1 //
#define DATA_HALF_CLWS_TM 20 -1 //

//TA1
#define DATA_TA1      2000000/50 -1 //LED & SW (2ms)

/* SW & LED*/
#define CHAT_COUNT      2 //
#define SW_TBL_SIZE     7 //
#define LED_PT12_TBL_SIZE 6 //
#define LED_PT34_TBL_SIZE 6 //
#define LED_PT56_TBL_SIZE 8 //

#define PT_SW_MASK      0b00011100 //SW MASK

#define PT_ALL          0b00000000 //SW all
#define PT_SW34         0b00000100 //SW3 & SW4
#define PT_SW24         0b00001000 //SW2 & SW4
#define PT_SW4          0b00001100 //SW4
#define PT_SW23         0b00010000 //SW2 & SW3
#define PT_SW3          0b00010100 //SW3
#define PT_SW2          0b00011000 //SW2
#define PT_NO_SW        0b00011100 //no SW

/* error */
#define NO_ERROR        0 //Error number
#define CAN_RCV_BUFF_ERROR 1 //Error number
#define TABLE_ERROR    2 //Error number
#define LED_ERROR       3 //Error number

/* interrupt Lv */
#define TAO_LVL         0 //
#define TA1_LVL         0 //LED & SW
#define CAN0_LVL        5 //CAN0
#define CAN0_ERROR_LVL  0 //CAN0 error

/***** LED DEFINE *****/
#define SEG_A  0x01
#define SEG_B  0x02
#define SEG_C  0x04
#define SEG_D  0x08
#define SEG_E  0x10
#define SEG_F  0x20
#define SEG_G  0x40
#define SEG_H  0x80
#define DISP_0 (~(SEG_A|SEG_B|SEG_C|SEG_D|SEG_E|SEG_F ))
#define DISP_1 (~( SEG_B|SEG_C ))
#define DISP_2 (~(SEG_A|SEG_B| SEG_D|SEG_E| SEG_G ))
#define DISP_3 (~(SEG_A|SEG_B|SEG_C|SEG_D| SEG_G ))
#define DISP_4 (~( SEG_B|SEG_C| SEG_F|SEG_G ))

```



```

#define DISP_5  (~(SEG_A|      SEG_C|SEG_D|      SEG_F|SEG_G      ))
#define DISP_6  (~(SEG_A|      SEG_C|SEG_D|SEG_E|SEG_F|SEG_G      ))
#define DISP_7  (~(SEG_A|SEG_B|SEG_C|      SEG_F      ))
#define DISP_8  (~(SEG_A|SEG_B|SEG_C|SEG_D|SEG_E|SEG_F|SEG_G      ))
#define DISP_9  (~(SEG_A|SEG_B|SEG_C|SEG_D      |SEG_F|SEG_G      ))
#define DISP_A  (~(SEG_A|SEG_B|SEG_C|      SEG_E|SEG_F|SEG_G      ))
#define DISP_B  (~(      SEG_C|SEG_D|SEG_E|SEG_F|SEG_G      ))
#define DISP_C  (~(SEG_A|      SEG_D|SEG_E|SEG_F      ))
#define DISP_D  (~(      SEG_B|SEG_C|SEG_D|SEG_E|      SEG_G      ))
#define DISP_E  (~(SEG_A|      SEG_D|SEG_E|SEG_F|SEG_G      ))
#define DISP_F  (~(SEG_A|      SEG_E|SEG_F|SEG_G      ))

#define DISP_CL0  (~(SEG_A      ))
#define DISP_CL1  (~(      SEG_B      ))
#define DISP_CL2  (~(      SEG_C      ))
#define DISP_CL3  (~(      SEG_D      ))
#define DISP_CL4  (~(      SEG_E      ))
#define DISP_CL5  (~(      SEG_F      ))

#define DISP_CL1_0  (~(SEG_A      ))
#define DISP_CL1_1  (~(SEG_A|SEG_B      ))
#define DISP_CL1_2  (~(      SEG_B|SEG_C      ))
#define DISP_CL1_3  (~(      SEG_C|SEG_D      ))
#define DISP_CL1_4  (~(      SEG_D      ))
#define DISP_CL1_5  ((SEG_A|SEG_B|SEG_C|SEG_D|SEG_E|SEG_F|SEG_G|SEG_H))
#define DISP_CL1_6  ((SEG_A|SEG_B|SEG_C|SEG_D|SEG_E|SEG_F|SEG_G|SEG_H))
#define DISP_CL1_7  ((SEG_A|SEG_B|SEG_C|SEG_D|SEG_E|SEG_F|SEG_G|SEG_H))
#define DISP_CL2_0  (~(SEG_A      ))
#define DISP_CL2_1  ((SEG_A|SEG_B|SEG_C|SEG_D|SEG_E|SEG_F|SEG_G|SEG_H))
#define DISP_CL2_2  ((SEG_A|SEG_B|SEG_C|SEG_D|SEG_E|SEG_F|SEG_G|SEG_H))
#define DISP_CL2_3  ((SEG_A|SEG_B|SEG_C|SEG_D|SEG_E|SEG_F|SEG_G|SEG_H))
#define DISP_CL2_4  (~(      SEG_D      ))
#define DISP_CL2_5  (~(      SEG_D|SEG_E      ))
#define DISP_CL2_6  (~(      SEG_E|SEG_F      ))
#define DISP_CL2_7  (~(SEG_A      |SEG_F      ))
/*****/

typedef struct{
    unsigned short id;
    unsigned char dlc;
    union {
        unsigned char data[8];
        unsigned long long data_8;
    }data;
}can_std_data_def;

```

```

/*"FILE COMMENT"*****
*System Name : CAN DEMO PROGRAM1
*File Name   : declare1.h
*Version    : 1.00
*Contens    :
*Customer   : RS0
*Model      :
*Order      :
*CPU        :
*Compiler   : NC30 Version 5.30 Release 02
*OS         :
*Programmer :
*Note       :
*****
* Copyright(C)2005, Renesas Technology Corp.
* Copyright(C)2005, Renesas Solutions Corp.
* All rights reserved.
*
*****
*History      2005.07.1   Ver 1.00
**"FILE COMMENT END"*****/

#ifdef DECLARE_MAIN
    #define    READTASK_EXTERN
#else
    #define    READTASK_EXTERN extern
#endif

/*****
*
*          fnc
*****/

/* MAIN */
READTASK_EXTERN void main(void);
READTASK_EXTERN void cpu_initial(void);
READTASK_EXTERN void timer_initial(void);
READTASK_EXTERN void sw_led_initial(void);

/* CAN */
READTASK_EXTERN void can_initial(void);
READTASK_EXTERN void set_bitrate_can0(void);
READTASK_EXTERN void config_can0(void);
READTASK_EXTERN void set_rec_std_dataframe_can0(unsigned short, unsigned short);
READTASK_EXTERN void set_mask_can0(void);
READTASK_EXTERN void set_trm_std_dataframe_can0(unsigned short, can_std_data_def *);
READTASK_EXTERN void get_message_can0(unsigned short, can_std_data_def *);
READTASK_EXTERN void trm_can_ad(void);
READTASK_EXTERN void abort_trm_can0(unsigned short in_slot);

/* SW & LED */
READTASK_EXTERN void led_fnc(void);
READTASK_EXTERN void led_pt_fnc(void);
READTASK_EXTERN void led_mode_change(void);
READTASK_EXTERN void sw_fnc(void);

```

```

READTASK_EXTERN void sw_2(void);
READTASK_EXTERN void sw_3(void);
READTASK_EXTERN void sw_4(void);
READTASK_EXTERN void sw_3_4(void);
READTASK_EXTERN void sw_4_3(void);
READTASK_EXTERN void sw_other(void);
READTASK_EXTERN unsigned char fix_sw_fnc(unsigned char);
READTASK_EXTERN void dsp_return(void);
READTASK_EXTERN unsigned char sw_decode(void);

/* AD */
READTASK_EXTERN void ad_initial(void);
READTASK_EXTERN void ad_fnc(void);
READTASK_EXTERN unsigned char ad_read(void);
READTASK_EXTERN unsigned char ad_check(void);

/* ERROR */
READTASK_EXTERN void error_fnc(unsigned char);
READTASK_EXTERN void error_main_fnc(void);

/*****
*           interrupt fnc
*****/
#pragma INTERRUPT /B CANO_REC()
READTASK_EXTERN void CANO_REC(void);

/*****
*           variable
*****/

/* CAN */
READTASK_EXTERN can_std_data_def rec_data[2];           //CAN receive data
READTASK_EXTERN unsigned char F_RCV_DATA;             //Receive data display enable flag
READTASK_EXTERN unsigned char trm_id;                 //Transmit ID
READTASK_EXTERN unsigned char rcv_id;                 //Receive ID
READTASK_EXTERN unsigned char last_rcv_data;          //Last CAN receive data

/* SW */
READTASK_EXTERN unsigned char trm_count_sw;           //Transmit counter
READTASK_EXTERN unsigned char last_sw_data;           //Last switch data
READTASK_EXTERN unsigned char fix_sw_data;           //Fixed switch data
READTASK_EXTERN unsigned char last_fix_sw_data;       //Last fixed switch data
READTASK_EXTERN unsigned char sw_count;              //Switch counter
READTASK_EXTERN unsigned char F_SW;                  //Switch enable flag

/* LED */
READTASK_EXTERN unsigned char led_count;             //LED counter
READTASK_EXTERN unsigned char led_data[2];           //LED display data
READTASK_EXTERN unsigned char led_mode;              //LED mode
READTASK_EXTERN unsigned char last_led_mode;         //Last LED mode
READTASK_EXTERN unsigned char led_pt_count;          //LED display pattern counter

/* AD */
READTASK_EXTERN unsigned char last_ad_data;           //Last A/D data
READTASK_EXTERN unsigned short ave_ad_data;          //Average of A/D conversion data

```

```

READTASK_EXTERN unsigned char count_ad;           //A/D counter
READTASK_EXTERN unsigned short ad_data[3];       //A/D data

/* ERROR */
READTASK_EXTERN unsigned char error_num;

/*****
*           table
*****/
/* LED display table */
READTASK_EXTERN const unsigned char LED_TBL[16]
#ifdef DECLARE_MAIN
    = {DISP_0,DISP_1,DISP_2,DISP_3,
        DISP_4,DISP_5,DISP_6,DISP_7,
        DISP_8,DISP_9,DISP_A,DISP_B,
        DISP_C,DISP_D,DISP_E,DISP_F,}
#endif
;

/* LED display pattern table (mode clockwise) */
READTASK_EXTERN const unsigned char LED_PT1_TBL[LED_PT12_TBL_SIZE]
#ifdef DECLARE_MAIN
    = {DISP_C0,DISP_CL1,DISP_CL2,
        DISP_CL3,DISP_CL4,DISP_CL5}
#endif
;

/* LED display pattern table (mode clockwise) */
READTASK_EXTERN const unsigned char LED_PT2_TBL[LED_PT12_TBL_SIZE]
#ifdef DECLARE_MAIN
    = {DISP_C0,DISP_CL5,DISP_CL4,
        DISP_CL3,DISP_CL2,DISP_CL1}
#endif
;

/* LED display pattern table (mode anticlockwise) */
READTASK_EXTERN const unsigned char LED_PT3_TBL[LED_PT34_TBL_SIZE]
#ifdef DECLARE_MAIN
    = {DISP_C0,DISP_CL5,DISP_CL4,
        DISP_CL3,DISP_CL2,DISP_CL1}
#endif
;

/* LED display pattern table (mode anticlockwise) */
READTASK_EXTERN const unsigned char LED_PT4_TBL[LED_PT34_TBL_SIZE]
#ifdef DECLARE_MAIN
    = {DISP_C0,DISP_CL1,DISP_CL2,
        DISP_CL3,DISP_CL4,DISP_CL5}
#endif
;

/* LED display pattern table (mode half clockwise) */
READTASK_EXTERN const unsigned char LED_PT5_TBL[LED_PT56_TBL_SIZE]
#ifdef DECLARE_MAIN

```

```

    = {DISP_CL1_0,DISP_CL1_1,DISP_CL1_2,DISP_CL1_3,
        DISP_CL1_4,DISP_CL1_5,DISP_CL1_6,DISP_CL1_7}
#endif
;

/* LED display pattern table (mode half clockwise) */
READTASK_EXTERN const unsigned char LED_PT6_TBL[LED_PT56_TBL_SIZE]
#ifdef DECLARE_MAIN
    = {DISP_CL2_0,DISP_CL2_1,DISP_CL2_2,DISP_CL2_3,
        DISP_CL2_4,DISP_CL2_5,DISP_CL2_6,DISP_CL2_7}
#endif
;

READTASK_EXTERN const unsigned char trm_data_sw[3]
#ifdef DECLARE_MAIN
    = {DATA_CLWS,DATA_ANTCLWS,DATA_HALF_CLWS}
#endif
;

/* Jump table */
READTASK_EXTERN void (* const SW_TBL[SW_TBL_SIZE])(void)
#ifdef DECLARE_MAIN
    = {dsp_return,sw_2,sw_3,sw_4,sw_3_4,sw_4_3,sw_other}
#endif
;

```

```

/*"FILE COMMENT"*****
*System Name : CAN DEMO PROGRAM1
*File Name   : main1.c
*Version    : 1.00
*Contens    :
*Customer   : RS0
*Model      :
*Order      :
*CPU        :
*Compiler   : NC30 Version 5.30 Release 02
*OS         :
*Programmer :
*Note       :
*           :
*****
* Copyright(C)2005, Renesas Technology Corp.
* Copyright(C)2005, Renesas Solutions Corp.
* All rights reserved.
*
*****
*History      2005.07.1   Ver 1.00
*"FILE COMMENT END"*****/

```

```
#define DECLARE_MAIN
```

```
#include "sfr29.h"
#include "define1.h"
#include "declare1.h"
```

```

/*"FUNC COMMENT"*****
*ID           :1.0
*Description   :
*-----
*Include      : "sfr29.h"
*             : "define1.h"
*             : "declare1.h"
*-----
*Declaration  : void main(void)
*-----
*Function     : main
*-----
*Arguments    : Nothing
*-----
*Returns      : Nothing
*-----
*Input        : unsigned char error_num      ;error number
*Output       : unsigned char error_num      ;error number
*-----
*Call functions :cpu_initial()                ;Initialize CPU
*             :sw_led_initial()              ;Initialize switch and LED
*             :can_initial()                 ;Initialize CAN
*             :ad_initial()                  ;Initialize A/D conberter
*             :timer_initial()               ;Initialize timer

```

```

*          :led_mode_change()          ;Change LED mode
*          :led_pt_fnc()                ;Change LED lightning pattern
*          :led_fnc()                  ;LED dynamic lightning
*          :sw_fnc()                   ;Read switch port
*          :ad_fnc()                   ;A/D convert
*          :error_main_fnc()          ;Error function
*-----
*Note      :
*-----
*History   :
*          :
* "FUNC COMMENT END"*****/
void main(void)
{
    asm("fclr l");

    cpu_initial();
    sw_led_initial();
    can_initial();
    ad_initial();
    timer_initial();

    /* variable initial */
    error_num = NO_ERROR;           //set error flag "No error"

    tabsr = 0x03;                   //LED & SW timer
    asm("fset l");

    while(error_num == NO_ERROR){ //Error check
        if(ir_ta1ic == 1){         //TA1
            ta1ic = TA1_LVL;       //IR clear

            led_fnc();              //LED dynamic lightning
            sw_fnc();               //Switch function
            ad_fnc();               //A/D convert
        }

        if(ir_ta0ic == 1){         //TA0
            ta0ic = TA0_LVL;       //IR clear

            led_mode_change();      //Change LED mode
            led_pt_fnc();           //Change LED lightning pattern
        }
    }

    /* ERROR */
    error_main_fnc();              //Error function

    while(1);
}

```

```

/*"FUNC COMMENT"*****
*ID          :1.1
*Description  :Initialize CPU
*-----
*Include     : "sfr29.h"
*-----
*Declaration :void cpu_initial(void)
*-----
*Function    :Initialize main clock,processer mode and ports
*-----
*Arguments   :Nothing
*-----
*Returns     :Nothing
*-----
*Input       :Nothing
*Output      :Nothing
*-----
*Call functions :Nothing
*-----
*Note        :
*-----
*History     :
*           :
* "FUNC COMMENT END"*****/
void cpu_initial(void)
{
    while(pacr != 0x03){
        prcr = 0x04;
        pacr = 0x03;          //80pin
    }

    prcr = 0x03;

    /* clock,processer mode */
    cm1 = 0x20;
    cm2 = 0;
    pm0 = 0;
    pm1 = 0x08;
    cm0 = 0x08;
    pm2 = 0;

    /* port */
    p0 = 0x00;
    p1 = 0x00;
    p2 = 0x00;
    p3 = 0x00;
#if !_KD_
    p6 = 0x00;
#endif
    p7 = 0x00;
    p8 = 0x00;
    p9 = 0x00;
    p10 = 0x00;

```



```

    pd0 = 0x00;
    pd1 = 0x00;
    pd2 = 0x00;
    pd3 = 0x00;
#if !_KD_
    pd6 = 0x00;
#endif
    pd7 = 0x00;
    pd8 = 0x00;
    prcr = 0x07;
    pd9 = 0x00;
    pd10 = 0x00;

    prcr = 0;

    /* pull up control */
    pur0 = 0x00;
    pur1 = 0x00;
    pur2 = 0x00;

    /* port control */
    pcr = 0x00;
}

/*"FUNC COMMENT"*****
 *ID          :1.2
 *Description  :Initialize timer
 *-----
 *Include     : "sfr29.h"
 *           : "define1.h"
 *-----
 *Declaration :void timer_initial(void)
 *-----
 *Function    :Initialize TAO and TA1
 *-----
 *Arguments   :Nothing
 *-----
 *Returns     :Nothing
 *-----
 *Input       :Nothing
 *Output      :Nothing
 *-----
 *Call functions :Nothing
 *-----
 *Note        :
 *-----
 *History     :
 *           :
 *"FUNC COMMENT END"*****/
void timer_initial(void)
{
    tabsr = 0;           //all timer stop
    onsf = 0;
    trgsr = 0;
}

```

```

    ta0mr = 0x01;           //ivent
    ta0tgl = 1;            //count source:TA1
    ta0tgh = 1;           //
    ta0 = DATA_CLWS_TM;  //
    ta0ic = TAO_LVL;

    ta1mr = 0x00;         //timer mode
    ta1 = DATA_TA1;      //
    ta1ic = TA1_LVL;
}

/*"FUNC COMMENT"*****
 *ID          :1.3
 *Description  :Error process
 *-----
 *Include     : "sfr29.h"
 *-----
 *Declaration :void error_fnc(void)
 *-----
 *Function    :Timer,A/D converter and CAN module are stop.
 *            :LED off
 *-----
 *Arguments   :unsigned char in_num          ;Error number
 *-----
 *Returns     :Nothing
 *-----
 *Input       :Nothing
 *Output      :unsigned char error_num      ;Error number
 *-----
 *Call functions :Nothing
 *-----
 *Note        :
 *-----
 *History     :
 *            :
 *"FUNC COMMENT END"*****
void error_fnc(unsigned char in_num)
{
    asm("fclr l");

    tabsr = 0;           //all timer stop
    adcon0 = 0;         //AD stop
    reset_c0ctrlr = 1;  //CAN reset

    p0 = 0x03;          //LED OFF

    error_num = in_num; //Set error number
}

/*"FUNC COMMENT"*****
 *ID          :1.4
 *Description  :Error process

```

```

*-----
*Include      : "sfr29.h"
*-----
*Declaration  : void error_main_fnc(void)
*-----
*Function     : Display error number
*-----
*Arguments    : Nothing
*-----
*Returns     : Nothing
*-----
*Input       : unsigned char error_num      ; Error number
*             : unsigned char led_count    ; LED counter
*             : unsigned char led_data[]   ; LED display data
*Output      : unsigned char led_count    ; LED counter
*             : unsigned char led_data[]   ; LED display data
*-----
*Call functions : sw_led_initial()        ; Initialize switch and LED
*               : led_fnc()              ; LED dynamic lightning
*-----
*Note         :
*-----
*History      :
*             :
* "FUNC COMMENT END"*****/
void error_main_fnc(void)
{
    unsigned char in_led_on_off = 0;
    unsigned char i;

    asm("fclr l");

    tabsr = 0;           //all timer stop
    adcon0 = 0;         //AD stop
    reset_c0ctrlr = 1;  //CAN reset

    sw_led_initial();

    ta0mr = 0x01;       //ivent
    ta0tgr = 1;        //count source:TA1
    ta0tgh = 1;        //
    ta0 = 250 - 1;     //
    ta0ic = 0x00;

    ta1mr = 0x00;      //timer mode
    ta1 = 2000000/50 - 1; //2ms
    ta1ic = 0x00;

    led_count = 0;
    led_data[1] = LED_TBL[0x0e]; //Set error number
    led_data[0] = LED_TBL[error_num]; //

    tabsr = 0x03;      //TA0,TA1 start

```

```

while(1){
    if(in_led_on_off == 1){
        if(ir_ta1ic == 1){
            ta1ic = 0x00;

            led_fnc();
        }
    }

    if(ir_ta0ic == 1){ //LED flashing
        ta0ic = 0;

        if(in_led_on_off == 0){
            in_led_on_off = 1; //LED ON
        }
        else{
            in_led_on_off = 0; //LED OFF
            p0 = 0x03;
        }
    }
}
}
}

```

```

/*****"FILE COMMENT"*****/
*System Name : CAN DEMO PROGRAM1
*File Name   : can1.c
*Version     : 1.00
*Contens    :
*Customer    : RSO
*Model       :
*Order       :
*CPU         :
*Compiler    : NC30 Version 5.30 Release 02
*OS          :
*Programmer  :
*Note        :
*            :
*****
* Copyright(C)2005, Renesas Technology Corp.
* Copyright(C)2005, Renesas Solutions Corp.
* All rights reserved.
*****
*History      2005.07.1 Ver 1.00
/*****"FILE COMMENT END"*****/

```

```

#include "sfr29.h"
#include "define1.h"
#include "declare1.h"

```

```

/*****"FUNC COMMENT"*****/
*ID           :2.0
*Description   :Initialize CAN
*-----
*Include      : "sfr29.h"
*             : "define1.h"
*             : "declare1.h"
*-----
*Declaration  :void can_initial(void)
*-----
*Function     :Initialize CAN
*-----
*Arguments    :Nothing
*-----
*Returns      :Nothing
*-----
*Input        :Nothing
*Output       :unsigned char trm_id      ;transmit ID
*             :unsigned char rcv_id     ;receive ID
*-----
*Call functions :config_can0()
*             :set_rec_std_dataframe_can0
*-----
*Note         :
*-----
*History      :
*             :

```

```

    """FUNC COMMENT END"""*****/
void can_initial(void)
{
    config_can0();           //Configure can
    c0recic = CAN0_LVL;     //Interrupt lvl
    c01erric = CAN0_ERROR_LVL; //Interrupt lvl

    set_rec_std_dataframe_can0(14,CAN_RCV_ID); //Set can slot14
    set_rec_std_dataframe_can0(15,CAN_RCV_ID); //Set can slot15

    trm_id = CAN_TRM_ID;    //Set transmit ID
    rcv_id = CAN_RCV_ID;    //Set receive ID
}

/*"FUNC COMMENT"*****
 *ID          :2.1
 *Description  :Set can slot receive
 *-----
 *Include     : "sfr29.h"
 *           : "declare1.h"
 *-----
 *Declaration :void set_rec_std_dataframe_can0
 *           : (unsigned short in_slot,unsigned short in_sid)
 *-----
 *Function    :Set can slot receive
 *-----
 *Arguments   :unsigned short in_slot
 *           :unsigned short in_sid
 *-----
 *Returns     :Nothing
 *-----
 *Input       :Nothing
 *Output      :Nothing
 *-----
 *Call functions :Nothing
 *-----
 *Note        :
 *-----
 *History     :
 *           :
 *-----
 *"FUNC COMMENT END"*****/
void set_rec_std_dataframe_can0(
    unsigned short in_slot,
    unsigned short in_sid)
{
    c0icr |= (0x0001<<in_slot);
    c0idr &= ~(0x0001<<in_slot);

    c0slot[in_slot].ba.sidh = (in_sid>>6) & 0x1f; //SID10-6
    c0slot[in_slot].ba.sidl = in_sid & 0x3f; //SID5-0

    while(c0mctl[in_slot].byte != 0x00){
        c0mctl[in_slot].byte = 0x00;
    }
}

```

```

    c0mctl[in_slot].byte = 0x40;
}

/*"FUNC COMMENT"*****
 *ID          :2.0.2
 *Description  :Set can bitrate
 *-----
 *Include     : "sfr29.h"
 *           : "declare1.h"
 *-----
 *Declaration :void set_bitrate_can0(void)
 *-----
 *Function    :Set can bitrate
 *-----
 *Arguments   :Nothing
 *-----
 *Returns     :Nothing
 *-----
 *Input       :Nothing
 *Output      :Nothing
 *-----
 *Call functions :Nothing
 *-----
 *Note        :
 *-----
 *History     :
 *           :
 *-----
 *"FUNC COMMENT END"*****/
void set_bitrate_can0(void)
{
    prc0 = 1;
    cclkr = 0x01;    // CAN0 clock = f1/2
    prc0 = 0;

    /* 500kbps 10Tq */
    brp_c0conr = 0;    // fcan(10MHz)
    sam_c0conr = 0;    //
    pts_c0conr = 2;    // PTS = 3Tq
    pbs1_c0conr = 2;   // PBS1 = 3 Tq
    pbs2_c0conr = 2;   // PBS2 = 3 Tq
    sjw_c0conr = 0;    // SJW = 1 Tq
}

/*"FUNC COMMENT"*****
 *ID          :2.0.1
 *Description  :CAN configuration
 *-----
 *Include     : "sfr29.h"
 *           : "define1.h"
 *           : "declare1.h"
 *-----
 *Declaration :void config_can0() ;

```

```

*-----
*Function      :Set can slot receive
*-----
*Arguments     :Nothing
*-----
*Returns      :Nothing
*-----
*Input        :Nothing
*Output       :Nothing
*-----
*Call functions :set_bitrate_can0()
*              :set_mask_can0()
*-----
*Note         :
*-----
*History      :
*              :
* "FUNC COMMENT END"*****/
void config_can0(void)
{
    reset_c0ctlr = 1;          /* configuration */
    sleep_c0ctlr = 0;
    while(! state_reset_c0str){}

    porten_c0ctlr = 1;        /* CAN port enable */
    loopback_c0ctlr = 0;      /* normal */
    msgorder_c0ctlr = 1;      /* byte access */
    basiccan_c0ctlr = 1;      /* 0:normal mode 1:BasicCAN mode */
    buserrren_c0ctlr = 0;     /* nus error interrupt disable */
    tsprescale_c0ctlr = 0;

    set_bitrate_can0();       /* set bit rate */
    set_mask_can0();          /* mask */

    reset_c0ctlr = 0;
    while(state_reset_c0str){}
}

/* "FUNC COMMENT"*****
*ID          :2.0.3
*Description  :Set can mask
*-----
*Include     : "sfr29.h"
*            : "define1.h"
*            : "declare1.h"
*-----
*Declaration :void set_mask_can0(void)
*-----
*Function    :Set can mask
*-----
*Arguments   :Nothing
*-----
*Returns     :Nothing

```



```

*-----
*Input          :Nothing
*Output         :Nothing
*-----
*Call functions :Nothing
*-----
*Note           :
*-----
*History        :
*               :
* "FUNC COMMENT END"*****/
void set_mask_can0(void)
{
    c0lmar.ba.sidh = (MASK_14>>6) & 0x1f;          //SID10-6
    c0lmar.ba.sidl = MASK_14 & 0x3f;                //SID5-0

    c0lbr.ba.sidh = (MASK_15>>6) & 0x1f;          //SID10-6
    c0lbr.ba.sidl = MASK_15 & 0x3f;                //SID5-0
}

/* "FUNC COMMENT"*****
*ID          :2.2
*Description  :Get can data
*-----
*Include     : "sfr29.h"
*            : "define1.h"
*            : "declare1.h"
*-----
*Declaration :void get_message_can0
*             (unsigned short in_slot,can_std_data_def *in_rec_data)
*-----
*Function     :Get can data
*-----
*Arguments   :unsigned short in_slot          ;slot number
*             :can_std_data_def *in_rec_data ;receive data pointer
*-----
*Returns     :Nothing
*-----
*Input       :Nothing
*Output      :Nothing
*-----
*Call functions :Nothing
*-----
*Note        :
*-----
*History     :
*            :
* "FUNC COMMENT END"*****/
void get_message_can0(
    unsigned short in_slot,
    can_std_data_def *in_rec_data)
{
    unsigned char lp_dlc;

```

```

while(1){
    /* check receive complete */
    while(c0mctl[in_slot].receive.invaldata){}

    c0mctl[in_slot].receive.newdata = 0;

    /* overrun check */
    if(c0mctl[in_slot].receive.msglost){
        c0mctl[in_slot].receive.msglost = 0;
        error_fnc(CAN_RCV_BUFF_ERROR);
        break;
    }

    /* read message */
    in_rec_data->id = ((unsigned short)c0slot[in_slot].ba.sdh<<6)
                    + c0slot[in_slot].ba.sidl;
    in_rec_data->dlc = c0slot[in_slot].ba.dlc;
    for(lp_dlc=0; lp_dlc<c0slot[in_slot].ba.dlc; ++lp_dlc){
        in_rec_data->data.data[lp_dlc] = c0slot[in_slot].ba.data[lp_dlc];
    }

    /* check new message */
    if(! c0mctl[in_slot].receive.newdata){
        break;
    }
}
}

```

```

/*"FUNC COMMENT"*****
*ID          :2.3
*Description  :Set can slot transmit
*-----
*Include     : "sfr29.h"
*            : "define1.h"
*            : "declare1.h"
*-----
*Declaration  :void set_trm_std_dataframe_can0
*              (unsigned short in_slot,can_std_data_def *in_trm_data)
*-----
*Function     :Set can slot transmit
*-----
*Arguments   :unsigned short in_slot      ;slot number
*            :can_std_data_def *in_trm_data ;transmit data pointer
*-----
*Returns     :Nothing
*-----
*Input       :Nothing
*Output      :Nothing
*-----
*Call functions :Nothing
*-----
*Note       :
*-----

```

```

*History      :
*            :
*""FUNC COMMENT END""*****/
void set_trm_std_dataframe_can0(
    unsigned short in_slot,
    can_std_data_def *in_trm_data)
{
    unsigned char lp_dlc;

    while(c0mctl[in_slot].transmit.trmactive){}

    while(c0mctl[in_slot].byte != 0x00){
        c0mctl[in_slot].byte = 0x00;
    }

    /* transmission */
    c0icr &= ~(0x0001 << in_slot);
    c0idr &= ~(0x0001 << in_slot);

    //----- set dataframe -----
    c0slot[in_slot].ba.sidh = ((in_trm_data->id)>>6) & 0x1f;
                                     //SID10-6
    c0slot[in_slot].ba.sidl = (in_trm_data->id) & 0x3f;
                                     //SID5-0
    c0slot[in_slot].ba.dlc = in_trm_data->dlc; //DLC

    for(lp_dlc = 0;lp_dlc<(in_trm_data->dlc) && (lp_dlc<8) ; lp_dlc++){
        c0slot[in_slot].ba.data[lp_dlc] = in_trm_data->data.data[lp_dlc];
                                     //Data
    }
    //-----
    c0mctl[in_slot].byte = 0x80;
}

/*""FUNC COMMENT""*****
*ID          :2.4
*Description  :Abort can transmit
*-----
*Include     : "sfr29.h"
*            : "define1.h"
*            : "declare1.h"
*-----
*Declaration :void abort_trm_can0(unsigned short in_slot)
*-----
*Function     :Abort can transmit
*-----
*Arguments    :unsigned short in_slot      ;slot number
*-----
*Returns      :Nothing
*-----
*Input        :Nothing
*Output       :Nothing
*-----

```

```

*Call functions :Nothing
*-----
*Note :
*-----
*History :
* :
*"FUNC COMMENT END"*****/
void abort_trm_can0(unsigned short in_slot)
{
    if(c0mctl[in_slot].transmit.trmreq
        || c0mctl[in_slot].receive.recreq){

        c0mctl[in_slot].byte = 0;
        while(c0mctl[in_slot].transmit.trmactive){}
    }
}

/*"FUNC COMMENT"*****/
*ID :2.5
*Description :Transmit A/D data
*-----
*Include : "sfr29.h"
* : "define1.h"
* : "declare1.h"
*-----
*Declaration :void trm_can_ad(void)
*-----
*Function :Transmit A/D data
*-----
*Arguments :Nothing
*-----
*Returns :Nothing
*-----
*Input :trm_id ;Transmint ID
* :last_ad_data ;A/D data
*Output :Nothing
*-----
*Call functions :set_trm_std_dataframe_can0() ;Set can slot transmit
*-----
*Note :
*-----
*History :
* :
*"FUNC COMMENT END"*****/
void trm_can_ad(void)
{
    can_std_data_def in_send_data;
/* ID */
    in_send_data.id = trm_id;
/* DLC */
    in_send_data.dlc = 1;
/* data */
    in_send_data.data.data[0] = last_ad_data;
}

```

```
/* transmission */  
    set_trm_std_dataframe_can0(TRM_SLOT,&in_send_data);  
}
```

```

/*****"FILE COMMENT"*****/
*System Name : CAN DEMO PROGRAM1
*File Name   : ad1.c
*Version     : 1.00
*Contens    :
*Customer    : RSO
*Model       :
*Order       :
*CPU         :
*Compiler    : NC30 Version 5.30 Release 02
*OS          :
*Programmer  :
*Note        :
*            :
*****
* Copyright(C)2005, Renesas Technology Corp.
* Copyright(C)2005, Renesas Solutions Corp.
* All rights reserved.
*****
*History      2005.07.1 Ver 1.00
/*****"FILE COMMENT END"*****/

#include "sfr29.h"
#include "define1.h"
#include "declare1.h"

/*****"FUNC COMMENT"*****/
*ID           :3.0
*Description   :Initialize A/D converter
*-----*
*Include       : "sfr29.h"
*              : "declare1.h"
*-----*
*Declaration   :void ad_initial(void)
*-----*
*Function      :Initialize A/D converter
*-----*
*Arguments     :Nothing
*-----*
*Returns       :Nothing
*-----*
*Input         :Nothing
*Output        :Nothing
*-----*
*Call functions :Nothing
*-----*
*Note          :
*-----*
*History       :
*              :
/*****"FUNC COMMENT END"*****/
void ad_initial(void)
{

```

```

unsigned short i;
/* AD initial */
adcon0 = 0x08;      //fAD/4 (5MHz)
adcon1 = 0x28;      //
adcon2 = 0x01;      //

for(i = 0; i < 1000/50; i++);      //wait Vref(1us)

adst = 1;           //start AD

for(i = 0; i < 1000/50; i++);      //wait first AD conversion
}

/*"FUNC COMMENT"*****
*ID          :3.1
*Description  :A/D converter function
*-----
*Include     : "sfr29.h"
*           : "declare1.h"
*-----
*Declaration :void ad_fnc(void)
*-----
*Function    :A/D converter function
*-----
*Arguments   :Nothing
*-----
*Returns    :Nothing
*-----
*Input      :Nothing
*Output     :Nothing
*-----
*Call functions :ad_read()           ;Read A/D conversion value
*              :ad_check()          ;Check A/D conversion value
*-----
*Note       :
*-----
*History    :
*           :
*-----
*"FUNC COMMENT END"*****/
void ad_fnc(void)
{
    if(ad_read() != 0){
        if(ad_check() != 0){
            trm_can_ad();
        }
    }
}

/*"FUNC COMMENT"*****
*ID          :3.2
*Description  :Read A/D conversion value
*-----
*Include     : "sfr29.h"
*           : "define1.h"

```

```

*          : "declare1.h"
*-----
*Declaration : void ad_fnc(void)
*-----
*Function    : Read A/D conversion value
*-----
*Arguments   : Nothing
*-----
*Returns     : Renewal code
*             : 0; Not renew "ave_ad_data"
*             : 1; Renew "ave_ad_data"
*-----
*Input       : unsigned char count_ad      ;A/D counter
*             : unsigned char ad_data[]    ;A/D conversion data
*Output      : unsigned char count_ad      ;A/D counter
*             : unsigned short ave_ad_data ;Average of A/D conversion data
*             : unsigned char ad_data[]    ;A/D conversion data
*-----
*Call functions : Nothing
*-----
*Note        :
*-----
*History     :
*             :
* "FUNC COMMENT END"*****/
unsigned char ad_read(void)
{
    unsigned long in_ad_sum;
    unsigned short i;

    ad_data[count_ad] = ad0;
    count_ad++;

    if(count_ad >= AD_COUNT){          //Calculate A/D average
        in_ad_sum = 0;
        for(i = 0; i < (sizeof(ad_data) / 2); i++){
            in_ad_sum += ad_data[i];
        }
        ave_ad_data = in_ad_sum / (sizeof(ad_data) / 2);
        count_ad = 0;
        return 1;
    }

    return 0;
}

/* "FUNC COMMENT"*****
*ID          : 3.3
*Description  : Check A/D conversion value
*-----
*Include     : "sfr29.h"
*             : "define1.h"
*             : "declare1.h"
*-----

```



```

*Declaration      :unsigned char ad_check(void)
*-----
*Function         :Check whether A/D conversion value differ from
*                 last A/D conversion value
*-----
*Arguments        :Nothing
*-----
*Returns          : Difference code
*                 : 0; No difference
*                 : 1; Differing
*-----
*Input            :unsigned char last_ad_data      ;Last A/D countesion data
*                 :unsigned char ave_ad_data      ;Average of A/D conversion data
*Output           :unsigned char last_ad_data      ;Last A/D countesion data
*-----
*Call functions   :Nothing
*-----
*Note             :
*-----
*History          :
*                 :
* "FUNC COMMENT END"*****/
unsigned char ad_check(void)
{
    union {
        unsigned short word;
        unsigned char byte;
    }in_ad_data,in_last_ad_data;

    in_last_ad_data.byte = last_ad_data;
    in_last_ad_data.word = in_last_ad_data.word << 2;

    in_ad_data.word = ave_ad_data;

    if(in_last_ad_data.word > in_ad_data.word){
        if((in_last_ad_data.word - in_ad_data.word) > 0x0005){
            in_ad_data.word = (in_ad_data.word + 0x0004) >> 2;
            last_ad_data = in_ad_data.byte;
            return 1;
        }
        else if((in_last_ad_data.word == 0x0004) &&
            (in_ad_data.word <= 0x0001)){
//            in_ad_data.word = in_ad_data.word >> 2;
//            last_ad_data = in_ad_data.byte;
            last_ad_data = 0x00;
            return 1;
        }
    }
    else{
        if((in_ad_data.word - in_last_ad_data.word) > 0x0005){
            in_ad_data.word = in_ad_data.word >> 2;
            last_ad_data = in_ad_data.byte;
            return 1;
        }
    }
}

```

```
    }  
  
    return 0;  
}
```

```

/*"FILE COMMENT"*****
*System Name : CAN DEMO PROGRAM1
*File Name   : sw_led1.c
*Version     : 1.00
*Contens    :
*Customer    : RSO
*Model       :
*Order       :
*CPU         :
*Compiler    : NC30 Version 5.30 Release 02
*OS          :
*Programmer  :
*Note        :
*            :
*****
* Copyright(C)2005, Renesas Technology Corp.
* Copyright(C)2005, Renesas Solutions Corp.
* All rights reserved.
*
*****
*History      2005.07.1 Ver 1.00
*"FILE COMMENT END"*****

```

```

#include "sfr29.h"
#include "define1.h"
#include "declare1.h"

```

```

/*"FUNC COMMENT"*****
*ID           :4.0
*Description   :Initialize Switch and LED
*-----
*Include      : "sfr29.h"
*             : "define1.h"
*             : "declare1.h"
*-----
*Declaration  :void sw_led_initial(void)
*-----
*Function     :Initialize Switch and LED
*-----
*Arguments    :Nothing
*-----
*Returns     :Nothing
*-----
*Input       :Nothing
*Output      :unsigned char F_RCV_DATA      ;Receive data display enable flag
*             :unsigned char trm_count_sw  ;Transmit switch counter
*             :unsigned char led_mode     ;LED mode
*             :unsigned char last_led_mode ;Last LED mode
*             :unsigned char led_data[]   ;LED display data
*             :unsigned char sw_count     ;Switch counter
*             :unsigned char last_sw_data ;Last switch data
*-----
*Call functions :Nothing

```

```

*-----
*Note      :
*-----
*History   :
*         :
*         :
* "FUNC COMMENT END"*****/
void sw_led_initial(void)
{
    unsigned char i;
    /* LED port initial */
    p0 = 0x03;
    p2 = 0xff;
    pd0 = 0x03;
    p0 = 0x03;
    pd2 = 0xff;
    p2 = 0xff;

    /* SW port initial */
    pd8_2 = 0;           //SW2(INT0)
    pd8_3 = 0;           //SW3(INT1)
    pd8_4 = 0;           //SW4(INT2)

    /* variable initial */
    F_RCV_DATA = 1;      //receive data disp enable
    trm_count_sw = 0;
    led_mode = MODE_CLWS; //set LED mode
    last_led_mode = MODE_CLWS; //
    led_data[0] = LED_PT1_TBL[0]; //set initial LED dsp
    led_data[1] = LED_PT2_TBL[0]; //set initial LED dsp
    sw_count = 0;
    last_sw_data = 0;
}

/* "FUNC COMMENT"*****
*ID      :4.1
*Description :LED dynamic lightning
*-----
*Include  : "sfr29.h"
*         : "define1.h"
*         : "declare1.h"
*-----
*Declaration :void led_fnc(void)
*-----
*Function    :LED dynamic lightning
*-----
*Arguments   :Nothing
*-----
*Returns     :Nothing
*-----
*Input       :unsigned char led_data[] ;LED display data
*           :unsigned char led_count ;LED counter
*Output      :unsigned char led_count ;LED counter
*-----
*Call functions :Nothing

```

```

*-----
*Note          :
*-----
*History       :
*              :
*"FUNC COMMENT END"*****/
void led_fnc(void)
{
/* LED */
    p0 = 0x03;                //LED OFF

    led_count++;
    if(led_count > 1){
        led_count = 0;
    }

    p2 = led_data[led_count]; //set led data

    p0 = 0x01 << led_count;   //LED ON
}

/*"FUNC COMMENT"*****
*ID            :4.2
*Description    :Change LED display pattern
*-----
*Include       : "sfr29.h"
*              : "define1.h"
*              : "declare1.h"
*-----
*Declaration   :void led_pt_fnc(void)
*-----
*Function      :Change LED display pattern
*-----
*Arguments     :Nothing
*-----
*Returns       :Nothing
*-----
*Input         :unsigned char led_mode          ;LED mode
*              :unsigned char led_pt_count     ;LED display pattern counter
*Output        :unsigned char led_data[]       ;LED display data
*              :unsigned char led_pt_count     ;LED display pattern counter
*-----
*Call functions :error_fnc()                  ;Error function
*-----
*Note          :
*-----
*History       :
*              :
*"FUNC COMMENT END"*****/
void led_pt_fnc(void)
{
    switch (led_mode){
        case MODE_CLWS:
            led_pt_count++;

```

```

        if(led_pt_count >= sizeof(LED_PT1_TBL)){
            led_pt_count = 0;
        }
        led_data[0] = LED_PT1_TBL[led_pt_count];
        led_data[1] = LED_PT2_TBL[led_pt_count];
        break;

    case MODE_ANTCLWS:
        led_pt_count--;
        if(led_pt_count >= sizeof(LED_PT1_TBL)){
            led_pt_count = sizeof(LED_PT1_TBL)-1;
        }
        led_data[0] = LED_PT1_TBL[led_pt_count];
        led_data[1] = LED_PT2_TBL[led_pt_count];
        break;

    case MODE_HALF_CLWS:
        led_pt_count++;
        if(led_pt_count >= sizeof(LED_PT5_TBL)){
            led_pt_count = 0;
        }
        led_data[0] = LED_PT5_TBL[led_pt_count];
        led_data[1] = LED_PT6_TBL[led_pt_count];
        break;
    case MODE_RCV_ID:
        break;
    case MODE_TRM_ID:
        break;
    default:
        error_fnc(LED_ERROR);
        break;
    }
}

/*"FUNC COMMENT"*****
*ID          :4.3
*Description  :Change LED mode
*-----
*Include     : "sfr29.h"
*            : "define1.h"
*            : "declare1.h"
*-----
*Declaration :void led_mode_change(void)
*-----
*Function    :Change LED mode
*-----
*Arguments   :Nothing
*-----
*Returns     :Nothing
*-----
*Input       :unsigned char F_RCV_DATA      ;Receive data display enable flag
*            :unsigned char led_mode       ;LED mode
*            :unsigned char last_rcv_data   ;Last CAN receive data
*Output      :unsigned char led_data[]     ;LED display data

```

```

*          :unsigned char led_mode          ;LED mode
*          :unsigned char last_led_mode     ;Last LED mode
*-----
*Call functions :error_fnc()              ;Error function
*-----
*Note          :
*-----
*History       :
*              :
*""FUNC COMMENT END""*****/
void led_mode_change(void)
{
    unsigned char in_last_rcv_data;

    if(F_RCV_DATA == 0){                    //receive data disable?
        return;
    }

    in_last_rcv_data = last_rcv_data;

    switch (in_last_rcv_data){
        case DATA_CLWS:
            if(led_mode != MODE_CLWS){      //mode change clockwise
                led_mode = MODE_CLWS;
                last_led_mode = led_mode;
                ta0 = DATA_CLWS_TM;
            }
            break;

        case DATA_ANTCLWS:
            if(led_mode != MODE_ANTCLWS){   //mode change anticlockwise
                led_mode = MODE_ANTCLWS;
                last_led_mode = led_mode;
                ta0 = DATA_ANTCLWS_TM;
            }
            break;

        case DATA_HALF_CLWS:
            if(led_mode != MODE_HALF_CLWS){ //mode change half clockwise
                led_mode = MODE_HALF_CLWS;
                last_led_mode = led_mode;
                ta0 = DATA_HALF_CLWS_TM;
            }
            break;

        default:
            break;
    }
}

/*""FUNC COMMENT""*****
*ID          :4.4
*Description :Switch function

```

```

*-----
*Include      : "sfr29.h"
*             : "define1.h"
*             : "declare1.h"
*-----
*Declaration  : void sw_fnc(void)
*-----
*Function     : Switch function
*-----
*Arguments    : Nothing
*-----
*Returns     : Nothing
*-----
*Input       : Nothing
*Output      : Nothing
*-----
*Call functions : error_fnc()           ;Error function
*              : sw_2()                 ;Switch2 function
*              : sw_3()                 ;Switch3 function
*              : sw_4()                 ;Switch4 function
*              : sw_3_4()               ;Switch3_4 function
*              : sw_4_3()               ;Switch4_3 function
*              : sw_other()             ;Other switch function
*              : fix_sw_fnc()           ;Fix swithc function
*              : sw_decode()           ;Swithc decode function
*-----
*Note        :
*-----
*History     :
*           :
* "FUNC COMMENT END"*****/
void sw_fnc(void)
{
    unsigned char in_data,i;

    in_data = p8 & PT_SW_MASK;
    if(fix_sw_fnc(in_data)){
        i = sw_decode();
        if(i < SW_TBL_SIZE){ //table check
            (*SW_TBL[i])(); //call sw fnc
        }
        else{
            error_fnc(TABLE_ERROR);
        }
    }
}

/* "FUNC COMMENT"*****
*ID      : 4.4.1
*Description : Switch decode function
*-----
*Include  : "define1.h"
*         : "declare1.h"

```



```

*-----
*Declaration      :unsigned char sw_decode(void)
*-----
*Function         :Switch decode function
*-----
*Arguments       :Nothing
*-----
*Returns         :unsigned char in_rtn          ;Decode number
*-----
*Input           :unsigned char fix_sw_data     ;Fixed switch data
*                :unsigned char last_fix_sw_data ;Last fixed switch data
*Output         :unsigned char last_fix_sw_data ;Last fixed switch data
*-----
*Call functions  :error_fnc()                  ;Error function
*-----
*Note           :
*-----
*History        :
*                :
*""FUNC COMMENT END""*****/
unsigned char sw_decode(void)
{
    unsigned char in_rtn = 0;

    switch(fix_sw_data){
        case PT_NO_SW:      //no SW
            in_rtn = 0;
            break;
        case PT_SW2:       //SW2
            if(last_fix_sw_data == PT_NO_SW){
                in_rtn = 1;
            }
            else{
                in_rtn = 6;
            }
            break;
        case PT_SW3:       //SW3
            in_rtn = 2;
            break;
        case PT_SW4:       //SW4
            in_rtn = 3;
            break;
        case PT_SW34:      //SW3 & SW4
            if(last_fix_sw_data == PT_SW3){
                in_rtn = 4; //SW3 --> SW4
            }
            else if(last_fix_sw_data == PT_SW4){
                in_rtn = 5; //SW4 --> SW3
            }
            else{
                in_rtn = 6; //
            }
            break;
        default:

```

```

        in_rtn = 6;
        break;
    }

    last_fix_sw_data = fix_sw_data;
    return in_rtn;
}

/*"FUNC COMMENT"*****
*ID          :4.4.2
*Description  :Fix switch function
*-----
*Include     : "define1.h"
*           : "declare1.h"
*-----
*Declaration :unsigned char fix_sw_fnc(unsigned char)
*-----
*Function    :Fix switch function
*-----
*Arguments   :unsigned char in_data          ;Switch data
*-----
*Returns     :uneigned char in_rtn          ;Fixed switch data
*-----
*Input       :unsigned char fix_sw_data      ;Fixed switch data
*           :unsigned char last_sw_data     ;Last switch data
*           :unsigned char sw_count        ;Switch counter
*Output      :unsigned char trm_count_sw    ;Transmit counter
*-----
*Call functions :error_fnc()                ;Error function
*-----
*Note        :
*-----
*History     :
*           :
*-----
*"FUNC COMMENT END"*****/
unsigned char fix_sw_fnc(unsigned char in_data)
{
    unsigned char in_rtn = 0;

    if((fix_sw_data != in_data) && (last_sw_data == in_data)){
        sw_count++;
    }

    if(sw_count > CHAT_COUNT){
        fix_sw_data = in_data;
        sw_count = 0;
        in_rtn = 1;
    }

    last_sw_data = in_data;

    return in_rtn;
}

```

```

/*"FUNC COMMENT"*****
*ID          :4.4.3
*Description  :Switch2 function
*-----
*Include     : "sfr29.h"
*           : "define1.h"
*           : "declare1.h"
*-----
*Declaration  :void sw_2(void)
*-----
*Function     :Switch2 function
*-----
*Arguments    :Nothing
*-----
*Returns     :Nothing
*-----
*Input       :unsigned char trm_id          ;Transmit ID
*           :unsigned char trm_data_sw[]   ;Transmit data
*           :unsigned char trm_count_sw    ;Transmit counter
*Output      :unsigned char trm_count_sw   ;Transmit counter
*-----
*Call functions :error_fnc()              ;Error function
*-----
*Note         :
*-----
*History      :
*           :
*           :
*"FUNC COMMENT END"*****/
void sw_2(void)
{
    can_std_data_def in_send_data;

    /* ID */
    in_send_data.id = trm_id;
    /* DLC */
    in_send_data.dlc = 1;
    /* data */
    in_send_data.data.data[0] = trm_data_sw[trm_count_sw];
    /* transmission */
    set_trm_std_dataframe_can0(TRM_SLOT,&in_send_data);    //Transmit CAN

    trm_count_sw++;
    if(trm_count_sw >= sizeof(trm_data_sw)){
        trm_count_sw = 0;
    }
}

/*"FUNC COMMENT"*****
*ID          :4.4.4
*Description  :Switch3 function
*-----

```

```

*Include      : "define1.h"
*             : "declare1.h"
*-----
*Declaration  : void sw_3(void)
*-----
*Function     : Switch3 function
*-----
*Arguments    : Nothing
*-----
*Returns      : Nothing
*-----
*Input        : unsigned char rcv_id          ;Receive ID
*Output       : unsigned char F_RCV_DATA     ;Receive data display enable flag
*             : unsigned char led_data[]     ;LED display data
*             : unsigned char led_mode      ;LED mode
*-----
*Call functions :
*-----
*Note         :
*-----
*History      :
*             :
* "FUNC COMMENT END"*****/
void sw_3(void)
{
    F_RCV_DATA = 0;                //receive data disable

    led_mode = MODE_RCV_ID;        //Display rceive ID
    led_data[0] = LED_TBL[0x0f & rcv_id];
    led_data[1] = LED_TBL[rcv_id >> 4];
}

/* "FUNC COMMENT"*****
*ID           : 4.4.5
*Description   : Switch4_3 function
*-----
*Include      : "define1.h"
*             : "declare1.h"
*-----
*Declaration  : void sw_4_3(void)
*-----
*Function     : Switch4_3 function
*-----
*Arguments    : Nothing
*-----
*Returns      : Nothing
*-----
*Input        : unsigned char trm_id        ;Transmit ID
*Output       : unsigned char F_RCV_DATA     ;Receive data display enable flag
*             : unsigned char trm_id        ;Transmit ID
*             : unsigned char led_data[]     ;LED display data
*             : unsigned char led_mode      ;LED mode
*-----

```

```

*Call functions :
*-----
*Note :
*-----
*History :
* :
* "FUNC COMMENT END"*****/
void sw_4_3(void)
{
    F_RCV_DATA = 0;           //receive data disable
    trm_id++;                //Add transmit ID
    /* if(trm_id > 0x03ff){   //"trm_id" = 8bit
        trm_id = 0;
    }
*/
    led_data[0] = LED_TBL[0x0f & trm_id]; //Display transmit ID
    led_data[1] = LED_TBL[trm_id >> 4];
}

/* "FUNC COMMENT"*****
*ID :4.4.6
*Description :Switch4 function
*-----
*Include : "define1.h"
* : "declare1.h"
*-----
*Declaration :void sw_4(void)
*-----
*Function :Switch4 function
*-----
*Arguments :Nothing
*-----
*Returns :Nothing
*-----
*Input :unsigned char trm_id ;Transmit ID
*Output :unsigned char F_RCV_DATA ;Receive data display enable flag
* :unsigned char led_data[] ;LED display data
* :unsigned char led_mode ;LED mode
*-----
*Call functions :
*-----
*Note :
*-----
*History :
* :
* "FUNC COMMENT END"*****/
void sw_4(void)
{
    F_RCV_DATA = 0;           //receive data disable
    led_mode = MODE_TRM_ID;
    led_data[0] = LED_TBL[0x0f & trm_id]; //Display transmit ID
    led_data[1] = LED_TBL[trm_id >> 4];
}

```

```

/*"FUNC COMMENT"*****
*ID          :4.4.7
*Description  :Switch3_4 function
*-----
*Include     : "sfr29.h"
*           : "define1.h"
*           : "declare1.h"
*-----
*Declaration  :void sw_3_4(void)
*-----
*Function     :Switch3_4 function
*-----
*Arguments    :Nothing
*-----
*Returns     :Nothing
*-----
*Input       :unsigned char rcv_id          ;Receive ID
*Output      :unsigned char F_RCV_DATA     ;Receive data display enable flag
*           :unsigned char rcv_id          ;Receive ID
*           :unsigned char led_data[]      ;LED display data
*           :unsigned char led_mode       ;LED mode
*-----
*Call functions :abort_trm_can0()          ;Abort CAN transmit
*-----
*Note         :
*-----
*History      :
*           :
*-----
*"FUNC COMMENT END"*****/
void sw_3_4(void)
{
    unsigned short in_id;

    F_RCV_DATA = 0;                //receive data disable
    /* When transmitting the message, stop transmitting */
    abort_trm_can0(TRM_SLOT);

    if((c0mct114.receive.invaldata == 0)
        && (c0mct115.receive.invaldata == 0)){ //Check receiving

        /* Stop receiveing the message */
        c0mct114.byte = 0x00;
        c0mct115.byte = 0x00;

        rcv_id++;                  //Add receive ID
        /* if(rcv_id > 0x03ff){      //"trm_id" = 8bit
            rcv_id = 0;
        }
    */

    in_id = rcv_id;

    /* Set new ID and start rereceiving */

```

```

        set_rec_std_dataframe_can0(14, in_id);
        set_rec_std_dataframe_can0(15, in_id);

        /* Display receive ID */
        led_data[0] = LED_TBL[0x0f & rcv_id];
        led_data[1] = LED_TBL[rcv_id >> 4];
    }
}

/*"FUNC COMMENT"*****
*ID          :4.4.8
*Description  :Other switch function
*-----
*Include     : "declare1.h"
*-----
*Declaration  :void sw_other(void)
*-----
*Function     :Other switch function
*-----
*Arguments    :Nothing
*-----
*Returns     :Nothing
*-----
*Input       :Nothing
*Output      :Nothing
*-----
*Call functions :
*-----
*Note        :
*-----
*History     :
*           :
*""FUNC COMMENT END"*****/
void sw_other(void)
{
}

/*"FUNC COMMENT"*****
*ID          :4.4.9
*Description  :Return display
*-----
*Include     : "define1.h"
*           : "declare1.h"
*-----
*Declaration  :void dsp_return(void)
*-----
*Function     :Return display
*-----
*Arguments    :Nothing
*-----
*Returns     :Nothing
*-----
*Input       :unsigned char last_led_mode ;Last LED mode

```

```

*Output      :unsigned char led_mode      ;LED mode
*            :unsigned char F_RCV_DATA    ;Receive data display enable flag
*-----
*Call functions :
*-----
*Note        :
*-----
*History     :
*           :
*""FUNC COMMENT END""*****/
void dsp_return(void)
{
    F_RCV_DATA = 1;          //receive data enable
    led_mode = last_led_mode; //Return LED mode
}

```



```

/*****
* "FILE COMMENT"
* System Name : CAN DEMO PROGRAM1
* File Name   : int1.c
* Version     : 1.00
* Contens    :
* Customer    : RSO
* Model       :
* Order       :
* CPU         :
* Compiler    : NC30 Version 5.30 Release 02
* OS          :
* Programmer  :
* Note       :
*****/

* Copyright(C)2005, Renesas Technology Corp.
* Copyright(C)2005, Renesas Solutions Corp.
* All rights reserved.
*
*****/

* History      2005.07.1 Ver 1.00
* "FILE COMMENT END"
/

#include "sfr29.h"
#include "define1.h"
#include "declare1.h"

/*****
* "FUNC COMMENT"
* ID           :5.0
* Description   :CAN0 receive interrupt
* -----
* Include      : "sfr29.h"
*              : "define1.h"
*              : "declare1.h"
* -----
* Declaration   :void CAN0_REC(void)
* -----
* Function      :Read CAN0 receive data
* -----
* Arguments     :Nothing
* -----
* Returns      :Nothing
* -----
* Input        :unsigned char rec_data[]      ;CAN receive data
* Output       :unsigned char last_rcv_data   ;Last CAN receive data
* -----
* Call functions :get_message_can0()        ;Get CAN data
* -----
* Note         :
* -----
* History      :
*              :
* "FUNC COMMENT END"
/

void CAN0_REC(void)

```

```

{
  /* slot14 */
  if(c0mctl[14].receive.newdata == 1){
    get_message_can0(14, &rec_data[0]);    //Read receive data
    last_rcv_data = rec_data[0].data.data[0];
  }

  /* slot15 */
  if(c0mctl[15].receive.newdata == 1){
    get_message_can0(15, &rec_data[1]);    //Read receive data
    last_rcv_data = rec_data[1].data.data[0];
  }
}

```

6.2 プログラム

```

/*"FILE COMMENT"*****
*System Name : CAN PROGRAM
*File Name   : define2.h
*Version    : 1.00
*Contens    :
*Customer   : RSO
*Model      :
*Order      :
*CPU        :
*Compiler   : NC30 Version 5.30 Release 02
*OS         :
*Programmer :
*Note       :
*****
* Copyright(C)2005, Renesas Technology Corp.
* Copyright(C)2005, Renesas Solutions Corp.
* All rights reserved.
*
*****
*History      2005.07.1  Ver 1.00
**"FILE COMMENT END"*****/

/* CAN */
#define MASK_14 0xff //No mask
#define MASK_15 0xff //

#define TRM_SLOT 0 //CAN0 trm slot
#define TRM_DLC 1 //CAN0 trm dlc

#define CAN_TRM_ID 2 //CAN0 trm initial ID
#define CAN_RCV_ID 1 //CAN0 rcv initial ID

#define DATA_CLWS 1 //Clockwise code
#define DATA_ANTCLWS 2 //Anticlockwise code
#define DATA_HALF_CLWS 3 //Half clockwise code

/* SW & LED */
#define CHAT_COUNT 2 //

#define SW_TBL_SIZE 4 //
#define PT_SW_MASK 0b00011100 //SW MASK

#define PT_ALL 0b00000000 //SW all
#define PT_SW34 0b00000100 //SW3 & SW4
#define PT_SW24 0b00001000 //SW2 & SW4
#define PT_SW4 0b00001100 //SW4
#define PT_SW23 0b00010000 //SW2 & SW3
#define PT_SW3 0b00010100 //SW3
#define PT_SW2 0b00011000 //SW2
#define PT_NO_SW 0b00011100 //no SW

```

```

/* timer */
#define DATA_TA1 2000000/50 -1 //LED & SW (2ms)

/* error */
#define NO_ERROR 0 //Error number
#define CAN_RCV_BUFF_ERROR 1 //Error number
#define CAN_TRM_ERROR 2 //Error number
#define TABLE_ERROR 3 //Error number

/* interrupt Lv */
#define TA1_LVL 0 //LED & SW
#define CAN0_LVL 5 //CAN0
#define CAN0_ERROR_LVL 0 //CAN0 error

/***** LED DEFINE *****/
#define SEG_A 0x01
#define SEG_B 0x02
#define SEG_C 0x04
#define SEG_D 0x08
#define SEG_E 0x10
#define SEG_F 0x20
#define SEG_G 0x40
#define SEG_H 0x80
#define DISP_0 (~(SEG_A|SEG_B|SEG_C|SEG_D|SEG_E|SEG_F))
#define DISP_1 (~( SEG_B|SEG_C))
#define DISP_2 (~(SEG_A|SEG_B| SEG_D|SEG_E| SEG_G))
#define DISP_3 (~(SEG_A|SEG_B|SEG_C|SEG_D| SEG_G))
#define DISP_4 (~( SEG_B|SEG_C| SEG_F|SEG_G))
#define DISP_5 (~(SEG_A| SEG_C|SEG_D| SEG_F|SEG_G))
#define DISP_6 (~(SEG_A| SEG_C|SEG_D|SEG_E|SEG_F|SEG_G))
#define DISP_7 (~(SEG_A|SEG_B|SEG_C| SEG_F))
#define DISP_8 (~(SEG_A|SEG_B|SEG_C|SEG_D|SEG_E|SEG_F|SEG_G))
#define DISP_9 (~(SEG_A|SEG_B|SEG_C|SEG_D |SEG_F|SEG_G))
#define DISP_A (~(SEG_A|SEG_B|SEG_C| SEG_E|SEG_F|SEG_G))
#define DISP_B (~( SEG_C|SEG_D|SEG_E|SEG_F|SEG_G))
#define DISP_C (~(SEG_A| SEG_D|SEG_E|SEG_F))
#define DISP_D (~( SEG_B|SEG_C|SEG_D|SEG_E| SEG_G))
#define DISP_E (~(SEG_A| SEG_D|SEG_E|SEG_F|SEG_G))
#define DISP_F (~(SEG_A| SEG_E|SEG_F|SEG_G))
/*****

typedef struct{
    unsigned short id;
    unsigned char dlc;
    union {
        unsigned char data[8];
        unsigned long long data_8;
    }data;
}can_std_data_def;

```

```

/*****"FILE COMMENT"*****
*System Name : CAN PROGRAM
*File Name   : declare2.h
*Version    : 1.00
*Contens    :
*Customer   : RS0
*Model      :
*Order      :
*CPU        :
*Compiler   : NC30 Version 5.30 Release 02
*OS         :
*Programmer :
*Note       :
*****
* Copyright(C)2005, Renesas Technology Corp.
* Copyright(C)2005, Renesas Solutions Corp.
* All rights reserved.
*
*****
*History      2005.07.1   Ver 1.00
**"FILE COMMENT END"*****/

#ifdef DECLARE_MAIN
    #define    READTASK_EXTERN
#else
    #define    READTASK_EXTERN extern
#endif

/*****
*
*          fnc
*****/
READTASK_EXTERN void main(void);
READTASK_EXTERN void cpu_initial(void);
READTASK_EXTERN void timer_initial(void);
READTASK_EXTERN void sw_led_initial(void);

/* CAN */
READTASK_EXTERN void can_initial(void);
READTASK_EXTERN void set_bitrate_can0(void);
READTASK_EXTERN void config_can0(void);
READTASK_EXTERN void set_rec_std_dataframe_can0(unsigned short, unsigned short);
READTASK_EXTERN void set_mask_can0(void);
READTASK_EXTERN void get_message_can0(unsigned short, can_std_data_def *);
READTASK_EXTERN void set_trm_std_dataframe_can0(unsigned short, can_std_data_def *);
READTASK_EXTERN void abort_trm_can0(unsigned short in_slot);

/* SW & LED */
READTASK_EXTERN void led_fnc(void);
READTASK_EXTERN void sw_fnc(void);
READTASK_EXTERN void sw_down(unsigned char);
READTASK_EXTERN unsigned char fix_sw_fnc(unsigned char);
READTASK_EXTERN unsigned char sw_decode(void);

```

```

/* ERROR */
READTASK_EXTERN void error_fnc(unsigned char);
READTASK_EXTERN void error_main_fnc(void);

/*****
*           interrupt fnc
*****/
#pragma INTERRUPT /B CANO_REC()
READTASK_EXTERN void CANO_REC(void);

/*****
*           variable
*****/

/* CAN */
READTASK_EXTERN can_std_data_def rec_data[2];           //CAN receive data
READTASK_EXTERN unsigned char trm_id;                 //Transmit ID
READTASK_EXTERN unsigned char rcv_id;                 //Receive ID
READTASK_EXTERN unsigned char trm_led_data[3];        //Transmit data

/* LED */
READTASK_EXTERN unsigned char led_data[2];            //LED display data
READTASK_EXTERN unsigned char led_count;             //LED counter

/* SW */
READTASK_EXTERN unsigned char sw_count;              //Switch counter
READTASK_EXTERN unsigned char last_sw_data;          //Last switch data
READTASK_EXTERN unsigned char fix_sw_data;          //Fixed switch data
READTASK_EXTERN unsigned char last_fix_sw_data;      //Last fixed switch data

/* ERROR */
READTASK_EXTERN unsigned char error_num;             //Error number

/*****
*           table
*****/
/* LED display table */
READTASK_EXTERN const unsigned char LED_TBL[16]
#ifdef DECLARE_MAIN
    = {DISP_0,DISP_1,DISP_2,DISP_3,
        DISP_4,DISP_5,DISP_6,DISP_7,
        DISP_8,DISP_9,DISP_A,DISP_B,
        DISP_C,DISP_D,DISP_E,DISP_F}
#endif
;

```

```

/*"FILE COMMENT"*****
*System Name : CAN DEMO PROGRAM
*File Name   : main2.c
*Version    : 1.00
*Contens    :
*Customer   : RS0
*Model      :
*Order      :
*CPU        :
*Compiler   : NC30 Version 5.30 Release 02
*OS         :
*Programmer :
*Note       :
*           :
*****
* Copyright(C)2005, Renesas Technology Corp.
* Copyright(C)2005, Renesas Solutions Corp.
* All rights reserved.
*
*****
*History      2005.07.1   Ver 1.00
*"FILE COMMENT END"*****/

#define DECLARE_MAIN

#include "sfr29.h"
#include "define2.h"
#include "declare2.h"

/*"FUNC COMMENT"*****
*ID           :1.0
*Description   :
*-----
*Include      : "sfr29.h"
*             : "define2.h"
*             : "declare2.h"
*-----
*Declaration  :void main(void)
*-----
*Function     :main
*-----
*Arguments    :Nothing
*-----
*Returns      :Nothing
*-----
*Input        :unsigned char error_num      ;error number
*Output       :unsigned char error_num      ;error number
*-----
*Call functions :cpu_initial()              ;Initialize CPU
*               :sw_led_initial()          ;Initialize switch and LED
*               :can_initial()             ;Initialize CAN
*               :timer_initial()           ;Initialize timer
*               :led_fnc()                 ;LED dynamic lightning

```

```

*          :sw_fnc()          ;Read switch port
*-----
*Note      :
*-----
*History   :
*          :
* "FUNC COMMENT END"*****/
void main(void)
{
    unsigned int i;

    asm("fclr l");

    cpu_initial();
    sw_led_initial();
    can_initial();
    timer_initial();

/* variable initial */
    error_num = NO_ERROR;          //set error flag "No error"

    ta1s = 1;                      //LED & SW timer
    asm("fset l");

    while(error_num == NO_ERROR){
        if(ir_ta1ic == 1){
            ta1ic = TA1_LVL;        //IR clear

            led_fnc();              //LED dynamic lightning
            sw_fnc();              //Switch function
        }
    }

/* ERROR */
    error_main_fnc();

    while(1);
}

/* "FUNC COMMENT"*****
*ID          :1.1
*Description  :Initialize CPU
*-----
*Include     : "sfr29.h"
*-----
*Declaration :void cpu_initial(void)
*-----
*Function    :Initialize main clock,processer mode and ports
*-----
*Arguments   :Nothing
*-----

```


*Returns :Nothing

*

*Input :Nothing

*Output :Nothing

*

*Call functions :Nothing

*

*Note :

*

*History :

*

"FUNC COMMENT END"***/

void cpu_initial(void)

{

while(pacr != 0x03){

prcr = 0x04;

pacr = 0x03; //80pin

}

prcr = 0x03;

/* clock */

cm1 = 0x20;

cm2 = 0;

pm0 = 0;

pm1 = 0x08;

cm0 = 0x08;

pm2 = 0;

/* port */

p0 = 0x00;

p1 = 0x00;

p2 = 0x00;

p3 = 0x00;

#if !_KD_

p6 = 0x00;

#endif

p7 = 0x00;

p8 = 0x00;

p9 = 0x00;

p10 = 0x00;

pd0 = 0x00;

pd1 = 0x00;

pd2 = 0x00;

pd3 = 0x00;

#if !_KD_

pd6 = 0x00;

#endif

pd7 = 0x00;

pd8 = 0x00;

prcr = 0x07;

pd9 = 0x00;

pd10 = 0x00;

```

prcr = 0;

/* pull up control */
pur0 = 0x00;
pur1 = 0x00;
pur2 = 0x00;

/* port control */
pcr = 0x00;
}

/*"FUNC COMMENT"*****
*ID          :1.2
*Description  :Initialize timer
*-----
*Include     : "sfr29.h"
*           : "define2.h"
*-----
*Declaration :void timer_initial(void)
*-----
*Function     :Initialize TA1
*-----
*Arguments   :Nothing
*-----
*Returns     :Nothing
*-----
*Input       :Nothing
*Output      :Nothing
*-----
*Call functions :Nothing
*-----
*Note        :
*-----
*History     :
*           :
* "FUNC COMMENT END"*****/
void timer_initial(void)
{
    tabsr = 0;           //all timer stop
    onsf = 0;
    trgsr = 0;

    ta1mr = 0x00;       //timer mode
    ta1 = DATA_TA1;    //
    ta1ic = TA1_LVL;
}

/*"FUNC COMMENT"*****
*ID          :1.3
*Description  :Error process
*-----

```

```

*Include      : "sfr29.h"
*-----
*Declaration  : void error_fnc(void)
*-----
*Function     : Timer, A/D converter and CAN module are stop.
*             : LED off
*-----
*Arguments    : unsigned char in_num          ; Error number
*-----
*Returns      : Nothing
*-----
*Input        : Nothing
*Output       : unsigned char error_num      ; Error number
*-----
*Call functions : Nothing
*-----
*Note         :
*-----
*History      :
*             :
*             :
**"FUNC COMMENT"*****/
void error_fnc(unsigned char in_num)
{
    asm("fclr l");

    tabsr = 0;           //all timer stop
    adcon0 = 0;         //AD stop
    reset_c0ctrlr = 1;  //CAN reset

    p0 = 0x03;          //LED OFF

    error_num = in_num;
}

/*"FUNC COMMENT"*****
*ID           : 1.4
*Description   : Error process
*-----
*Include      : "sfr29.h"
*-----
*Declaration  : void error_main_fnc(void)
*-----
*Function     : Display error number
*-----
*Arguments    : Nothing
*-----
*Returns      : Nothing
*-----
*Input        : unsigned char error_num      ; Error number
*             : unsigned char led_count      ; LED counter
*             : unsigned char led_data[]    ; LED display data
*Output       : unsigned char led_count      ; LED counter
*             : unsigned char led_data[]    ; LED display data
*-----

```

```

*Call functions :sw_led_initial() ;Initialize switch and LED
*              :led_fnc() ;LED dynamic lightning
*-----
*Note          :
*-----
*History       :
*              :
*""FUNC COMMENT END""*****/
void error_main_fnc(void)
{
    unsigned char in_led_on_off = 0;
    unsigned char i;

    asm("fclr l");

    tabsr = 0; ;//all timer stop
    reset_c0ctrlr = 1; ;//CAN reset

    sw_led_initial();

    ta0mr = 0x01; ;//ivent
    ta0tgl = 1; ;//count source:TA1
    ta0tgh = 1; ;//
    ta0 = 250 - 1; ;//
    ta0ic = 0x00;

    ta1mr = 0x00; ;//timer mode
    ta1 = 2000000/50 -1; ;//2ms
    ta1ic = 0x00;

    led_count = 0;
    led_data[1] = LED_TBL[0x0e]; ;//Set error number
    led_data[0] = LED_TBL[error_num]; ;//

    tabsr = 0x03; ;//TA0,TA1 start

    while(1){
        if(in_led_on_off == 1){
            if(ir_ta1ic == 1){
                ta1ic = 0x00;

                led_fnc();
            }
        }

        if(ir_ta0ic == 1){ ;//LED flashing
            ta0ic = 0;

            if(in_led_on_off == 0){
                in_led_on_off = 1; ;//LED ON
            }
            else{
                in_led_on_off = 0; ;//LED OFF
                p0 = 0x03;
            }
        }
    }
}

```

```
}  
}  
}
```

```

/*****"FILE COMMENT"*****/
*System Name : CAN DEMO PROGRAM
*File Name   : can2.c
*Version     : 1.00
*Contens    :
*Customer    : RSO
*Model       :
*Order       :
*CPU         :
*Compiler    : NC30 Version 5.30 Release 02
*OS          :
*Programmer  :
*Note        :
*            :
*****
* Copyright(C)2005, Renesas Technology Corp.
* Copyright(C)2005, Renesas Solutions Corp.
* All rights reserved.
*****
*History      2005.07.1 Ver 1.00
/*****"FILE COMMENT END"*****/

```

```

#include "sfr29.h"
#include "define2.h"
#include "declare2.h"

```

```

/*****"FUNC COMMENT"*****/
*ID           :2.0
*Description   :Initialize CAN
*-----
*Include      : "sfr29.h"
*             : "define2.h"
*             : "declare2.h"
*-----
*Declaration  :void can_initial(void)
*-----
*Function     :Initialize CAN
*-----
*Arguments    :Nothing
*-----
*Returns      :Nothing
*-----
*Input        :Nothing
*Output       :unsigned char trm_id      ;transmit ID
*             :unsigned char rcv_id     ;receive ID
*-----
*Call functions :config_can0()
*             :set_rec_std_dataframe_can0
*-----
*Note         :
*-----
*History      :
*             :

```

```

    /*"FUNC COMMENT END"******/
void can_initial(void)
{
    config_can0();                //Configure can
    c0recic = CAN0_LVL;           //Interrupt lvl
    c01erric = CAN0_ERROR_LVL;   //Interrupt lvl

    set_rec_std_dataframe_can0(14,CAN_RCV_ID); //Set can slot14
    set_rec_std_dataframe_can0(15,CAN_RCV_ID); //Set can slot15

    trm_id = CAN_TRM_ID;         //Set transmit ID
    rcv_id = CAN_RCV_ID;        //Set receive ID
}

```

```

/*"FUNC COMMENT"******/
*ID          :2.1
*Description  :Set can slot receive
*-----*
*Include     : "sfr29.h"
*           : "declare2.h"
*-----*
*Declaration :void set_rec_std_dataframe_can0
*           : (unsigned short in_slot,unsigned short in_sid)
*-----*
*Function    :Set can slot receive
*-----*
*Arguments   :unsigned short in_slot
*           :unsigned short in_sid
*-----*
*Returns     :Nothing
*-----*
*Input       :Nothing
*Output      :Nothing
*-----*
*Call functions :Nothing
*-----*
*Note        :
*-----*
*History     :
*           :
*-----*
/*"FUNC COMMENT END"******/

```

```

void set_rec_std_dataframe_can0(
    unsigned short in_slot,
    unsigned short in_sid)
{
    c0icr |= (0x0001<<in_slot);
    c0idr &= ~(0x0001<<in_slot);

    c0slot[in_slot].ba.sidh = (in_sid>>6) & 0x1f; //SID10-6
    c0slot[in_slot].ba.sidl = in_sid & 0x3f;      //SID5-0

    while(c0mctl[in_slot].byte != 0x00){
        c0mctl[in_slot].byte = 0x00;
    }
}

```

```

    }
    c0mctl[in_slot].byte = 0x40;
}

```

```

/*"FUNC COMMENT"*****
*ID          :2.0.2
*Description  :Set can bitrate
*-----
*Include     : "sfr29.h"
*           : "declare2.h"
*-----
*Declaration  :void set_bitrate_can0(void)
*-----
*Function     :Set can bitrate
*-----
*Arguments   :Nothing
*-----
*Returns     :Nothing
*-----
*Input       :Nothing
*Output      :Nothing
*-----
*Call functions :Nothing
*-----
*Note        :
*-----
*History     :
*           :
*           :
*-----
/*"FUNC COMMENT END"*****

```

```

void set_bitrate_can0(void)
{
    prc0 = 1;
    cclkr = 0x01;    // CAN0 clock = f1/2
    prc0 = 0;

    /* 500kbps 10Tq */
    brp_c0conr = 0;    // fcan(10MHz)
    sam_c0conr = 0;    //
    pts_c0conr = 2;    // PTS = 3Tq
    pbs1_c0conr = 2;   // PBS1 = 3 Tq
    pbs2_c0conr = 2;   // PBS2 = 3 Tq
    sjw_c0conr = 0;    // SJW = 1 Tq
}

```

```

/*"FUNC COMMENT"*****
*ID          :2.0.1
*Description  :CAN configuration
*-----
*Include     : "sfr29.h"
*           : "define2.h"
*           : "declare2.h"
*-----

```



```

*Declaration      :void set_rec_std_dataframe_can0
*                  (unsigned short in_slot,unsigned short in_sid)
*-----
*Function         :Set can slot receive
*-----
*Arguments        :Nothing
*-----
*Returns          :Nothing
*-----
*Input            :Nothing
*Output           :Nothing
*-----
*Call functions   :set_bitrate_can0()
*                  :set_mask_can0()
*-----
*Note             :
*-----
*History          :
*                  :
* "FUNC COMMENT END"*****/
void config_can0(void)
{
    reset_c0ctlr = 1;          /* configuration */
    sleep_c0ctlr = 0;
    while(! state_reset_c0str){}

    porten_c0ctlr = 1;        /* CAN port enable */
    loopback_c0ctlr = 0;      /* normal */
    msgorder_c0ctlr = 1;      /* byte access */
    basiccan_c0ctlr = 1;      /* 0:normal mode 1:BasicCAN mode */
    buserrren_c0ctlr = 0;     /* nus error interrupt disable */
    tsprescale_c0ctlr = 0;

    set_bitrate_can0();       /* set bit rate */
    set_mask_can0();          /* mask */

    reset_c0ctlr = 0;
    while(state_reset_c0str){}
}

/* "FUNC COMMENT"*****
*ID                :2.0.3
*Description        :Set can mask
*-----
*Include           : "sfr29.h"
*                  : "define2.h"
*                  : "declare2.h"
*-----
*Declaration       :void set_mask_can0(void)
*-----
*Function          :Set can mask
*-----
*Arguments         :Nothing

```

```

*-----
*Returns      :Nothing
*-----
*Input        :Nothing
*Output       :Nothing
*-----
*Call functions :Nothing
*-----
*Note         :
*-----
*History      :
*             :
* "FUNC COMMENT END"*****/
void set_mask_can0(void)
{
    c0lmar.ba.sidh = (MASK_14>>6) & 0x1f;          //SID10-6
    c0lmar.ba.sidl = MASK_14 & 0x3f;                //SID5-0

    c0lbr.ba.sidh = (MASK_15>>6) & 0x1f;          //SID10-6
    c0lbr.ba.sidl = MASK_15 & 0x3f;                //SID5-0
}

/* "FUNC COMMENT"*****
*ID          :2.2
*Description  :Get can data
*-----
*Include     : "sfr29.h"
*            : "define2.h"
*            : "declare2.h"
*-----
*Declaration :void get_message_can0
*             (unsigned short in_slot,can_std_data_def *in_rec_data)
*-----
*Function     :Get can data
*-----
*Arguments    :unsigned short in_slot          ;slot number
*             :can_std_data_def *in_rec_data ;receive data pointer
*-----
*Returns      :Nothing
*-----
*Input        :Nothing
*Output       :Nothing
*-----
*Call functions :Nothing
*-----
*Note         :
*-----
*History      :
*             :
* "FUNC COMMENT END"*****/
void get_message_can0(
    unsigned short in_slot,
    can_std_data_def *in_rec_data)

```

```

{
    unsigned char lp_dlc;

    while(1){
        /* check receive complete */
        while(c0mctl[in_slot].receive.invaldata){}

        c0mctl[in_slot].receive.newdata = 0;

        /* overrun check */
        if(c0mctl[in_slot].receive.msglost){
            c0mctl[in_slot].receive.msglost = 0;
            error_fnc(CAN_RCV_BUFF_ERROR);
            break;
        }

        /* read message */
        in_rec_data->id = ((unsigned short)c0slot[in_slot].ba.sdh<<6)
            + c0slot[in_slot].ba.sidl;
        in_rec_data->dlc = c0slot[in_slot].ba.dlc;
        for(lp_dlc=0; lp_dlc<c0slot[in_slot].ba.dlc; ++lp_dlc){
            in_rec_data->data.data[lp_dlc] = c0slot[in_slot].ba.data[lp_dlc];
        }

        /* check new message */
        if(! c0mctl[in_slot].receive.newdata){
            break;
        }
    }
}

```

```

/*"FUNC COMMENT"*****
*ID                :2.3
*Description       :Set can slot transmit
*-----
*Include           : "sfr29.h"
*                  : "define2.h"
*                  : "declare2.h"
*-----
*Declaration       :void set_trm_std_dataframe_can0
*                  : (unsigned short in_slot,can_std_data_def *in_trm_data)
*-----
*Function          :Set can slot transmit
*-----
*Arguments         :unsigned short in_slot          ;slot number
*                  :can_std_data_def *in_trm_data ;transmit data pointer
*-----
*Returns           :Nothing
*-----
*Input             :Nothing
*Output            :Nothing
*-----
*Call functions    :Nothing

```

```

*-----
*Note          :
*-----
*History       :
*              :
* "FUNC COMMENT END"*****/
void set_trm_std_dataframe_can0(
    unsigned short in_slot,
    can_std_data_def *in_trm_data)
{
    unsigned char lp_dlc;

    while(c0mctl[in_slot].transmit.trmactive){}

    while(c0mctl[in_slot].byte != 0x00){
        c0mctl[in_slot].byte = 0x00;
    }

    /* transmission */
    c0icr &= ~(0x0001 << in_slot);
    c0idr &= ~(0x0001 << in_slot);

//----- set dataframe -----
    c0slot[in_slot].ba.sidh = ((in_trm_data->id)>>6) & 0x1f;
                                //SID10-6
    c0slot[in_slot].ba.sidl = (in_trm_data->id) & 0x3f;
                                //SID5-0
    c0slot[in_slot].ba.dlc = in_trm_data->dlc;    //DLC

    for(lp_dlc = 0;lp_dlc<(in_trm_data->dlc) && (lp_dlc<8) ; lp_dlc++){
        c0slot[in_slot].ba.data[lp_dlc] = in_trm_data->data.data[lp_dlc];
                                //Data
    }
//-----
    c0mctl[in_slot].byte = 0x80;
}

/* "FUNC COMMENT"*****
*ID          :2.4
*Description  :Abort can transmit
*-----
*Include     : "sfr29.h"
*            : "define2.h"
*            : "declare2.h"
*-----
*Declaration  :void abort_trm_can0(unsigned short in_slot)
*-----
*Function     :Abort can transmit
*-----
*Arguments    :unsigned short in_slot          ;slot number
*-----
*Returns     :Nothing
*-----

```

```

*Input          :Nothing
*Output         :Nothing
*-----
*Call functions :Nothing
*-----
*Note           :
*-----
*History        :
*               :
*""FUNC COMMENT END""*****/
void abort_trm_can0(unsigned short in_slot)
{
    if(c0mctl[in_slot].transmit.trmreq
        || c0mctl[in_slot].receive.recreq){

        c0mctl[in_slot].byte = 0;
        while(c0mctl[in_slot].transmit.trmactive){}
    }
}

```

```

/*"FILE COMMENT"*****
*System Name : CAN DEMO PROGRAM2
*File Name   : sw_led2.c
*Version    : 1.00
*Contens    :
*Customer   : RSO
*Model      :
*Order      :
*CPU        :
*Compiler   : NC30 Version 5.30 Release 02
*OS         :
*Programmer :
*Note       :
*           :
*****
* Copyright(C)2005, Renesas Technology Corp.
* Copyright(C)2005, Renesas Solutions Corp.
* All rights reserved.
*
*****
*History      2005.07.1 Ver 1.00
* "FILE COMMENT END"*****

```

```

#include "sfr29.h"
#include "define2.h"
#include "declare2.h"

```

```

/*"FUNC COMMENT"*****
*ID           :3.0
*Description  :Initialize Switch and LED
*-----
*Include      : "sfr29.h"
*             : "define2.h"
*             : "declare2.h"
*-----
*Declaration  :void sw_led_initial(void)
*-----
*Function     :Initialize Switch and LED
*-----
*Arguments    :Nothing
*-----
*Returns     :Nothing
*-----
*Input       :Nothing
*Output      :unsigned char led_data[]      ;LED display data
*             :unsigned char trm_led_data[] ;Transmit data
*-----
*Call functions :Nothing
*-----
*Note        :
*-----
*History     :
*           :

```

```

    """FUNC COMMENT END"""*****
void sw_led_initial(void)
{
    /* LED initial */
    p0 = 0x03;
    p2 = 0xff;
    pd0 = 0x03;
    pd2 = 0xff;

    led_data[0] = LED_TBL[0x00];    //initial LED dsp
    led_data[1] = LED_TBL[0x00];    //

    /* SW initial */
    pd8_2 = 0;                      //SW2(INT0)
    pd8_3 = 0;                      //SW3(INT1)
    pd8_4 = 0;                      //SW4(INT2)

    /* variable initial */
    trm_led_data[0] = DATA_CLWS;    //set transmit data
    trm_led_data[1] = DATA_ANTCLWS; //
    trm_led_data[2] = DATA_HALF_CLWS; //
}

/*"FUNC COMMENT"*****
*ID          :3.1
*Description  :LED dynamic lightning
*-----
*Include     : "sfr29.h"
*           : "define2.h"
*           : "declare2.h"
*-----
*Declaration :void led_fnc(void)
*-----
*Function    :LED dynamic lightning
*-----
*Arguments   :Nothing
*-----
*Returns     :Nothing
*-----
*Input       :unsigned char led_data[] ;LED display data
*           :unsigned char led_count ;LED counter
*Output     :unsigned char led_count ;LED counter
*-----
*Call functions :Nothing
*-----
*Note        :
*-----
*History     :
*           :
    """FUNC COMMENT END"""*****
void led_fnc(void)
{
    p0 = 0x03;                      //LED OFF

```

```

led_count++;
if(led_count > 1){
    led_count = 0;
}
//set led data
p2 = led_data[led_count];

p0 = 0x01 << led_count;    //LED ON
}

```

```

/*"FUNC COMMENT"*****
*ID          :3.2
*Description  :Switch function
*-----
*Include     : "sfr29.h"
*            : "define2.h"
*            : "declare2.h"
*-----
*Declaration :void sw_fnc(void)
*-----
*Function    :Switch function
*-----
*Arguments   :Nothing
*-----
*Returns     :Nothing
*-----
*Input       :Nothing
*Output      :Nothing
*-----
*Call functions :sw_down()           ;Switch down function
*              :fix_sw_fnc()        ;Fix switch function
*              :sw_decode()         ;Switch decode function
*-----
*Note        :
*-----
*History     :
*            :
*-----
/*"FUNC COMMENT END"*****

```

```

void sw_fnc(void)
{
    unsigned char in_data,i;

    in_data = p8 & PT_SW_MASK;
    if(fix_sw_fnc(in_data)){
        i = sw_decode();
        if(i){
            sw_down(i-1);
        }
    }
}

```

```

/*"FUNC COMMENT"*****

```



```

*ID          :3.2.1
*Description  :Switch decode function
*-----
*Include     : "define2.h"
*           : "declare2.h"
*-----
*Declaration : unsigned char sw_decode(void)
*-----
*Function    : Switch decode function
*-----
*Arguments   : Nothing
*-----
*Returns     : unsigned char in_rtn          ;Decode number
*-----
*Input       : unsigned char fix_sw_data     ;Fixed switch data
*           : unsigned char last_fix_sw_data ;Last fixed switch data
*Output      : unsigned char last_fix_sw_data ;Last fixed switch data
*-----
*Call functions : error_fnc()                ;Error function
*-----
*Note        :
*-----
*History     :
*           :
* "FUNC COMMENT END"*****/
unsigned char sw_decode(void)
{
    unsigned char in_rtn = 0;

    if(last_fix_sw_data != PT_NO_SW){
        in_rtn = 0;
    }
    else{
        switch(fix_sw_data){
            case PT_NO_SW:    //no SW
                in_rtn = 0;
                break;
            case PT_SW2:     //SW2
                in_rtn = 1;
                break;
            case PT_SW3:     //SW3
                in_rtn = 2;
                break;
            case PT_SW4:     //SW4
                in_rtn = 3;
                break;
            default:
                in_rtn = 0;
                break;
        }
    }

    last_fix_sw_data = fix_sw_data;
    return in_rtn;
}

```

}

```

/*"FUNC COMMENT"*****
 *ID          :3.2.2
 *Description  :Fix switch function
 *-----
 *Include     : "sfr29.h"
 *            : "define2.h"
 *            : "declare2.h"
 *-----
 *Declaration  :unsigned char fix_sw_fnc(unsigned char)
 *-----
 *Function     :Fix switch function
 *-----
 *Arguments    :unsigned char in_data          ;Switch data
 *-----
 *Returns      :uneigned char in_rtn          ;Fixed switch data
 *-----
 *Input        :unsigned char fix_sw_data      ;Fixed switch data
 *            :unsigned char last_sw_data     ;Last switch data
 *            :unsigned char sw_count         ;Switch counter
 *Output       :unsigned char trm_count_sw     ;Transmit counter
 *-----
 *Call functions :error_fnc()                 ;Error function
 *-----
 *Note         :
 *-----
 *History      :
 *            :
 *            :
 *"FUNC COMMENT END"*****

```

```

unsigned char fix_sw_fnc(unsigned char in_data)
{
    unsigned char in_rtn = 0;

    if((fix_sw_data != in_data) && (last_sw_data == in_data)){
        sw_count++;
    }

    if(sw_count > CHAT_COUNT){
        fix_sw_data = in_data;
        sw_count = 0;
        in_rtn = 1;
    }

    last_sw_data = in_data;

    return in_rtn;
}

```

```

/*"FUNC COMMENT"*****
 *ID          :3.2.3
 *Description  :Switch down function

```

```

*-----
*Include      : "sfr29.h"
*             : "define2.h"
*             : "declare2.h"
*-----
*Declaration  : void sw_down(unsigned char in_trm_num)
*-----
*Function     : Switch2 down function
*-----
*Arguments    : Nothing
*-----
*Returns     : Nothing
*-----
*Input       : unsigned char trm_id          ; Transmit ID
*             : unsigned char trm_led_data[] ; Transmit data
*Output      : Nothing
*-----
*Call functions : error_fnc()                ; Error function
*-----
*Note         :
*-----
*History      :
*             :
* "FUNC COMMENT END"*****/
void sw_down(unsigned char in_trm_num)
{
    can_std_data_def in_trm_data;
    /* ID */
    in_trm_data.id = trm_id;
    /* DLC */
    in_trm_data.dlc = TRM_DLC;
    /* data */
    if(in_trm_num >= sizeof(trm_led_data)){
        error_fnc(CAN_TRM_ERROR);
    }

    in_trm_data.data.data[0] = trm_led_data[in_trm_num];

    /* transmission */
    set_trm_std_dataframe_can0(TRM_SLOT, &in_trm_data);
}

```

```

/*****
*System Name : CAN DEMO PROGRAM
*File Name   : int2.c
*Version    : 1.00
*Contens    :
*Customer   : RSO
*Model      :
*Order      :
*CPU        :
*Compiler   : NC30 Version 5.30 Release 02
*OS         :
*Programmer :
*Note       :
*****/

* Copyright(C)2005, Renesas Technology Corp.
* Copyright(C)2005, Renesas Solutions Corp.
* All rights reserved.
*
*****/

*History      2005.07.1 Ver 1.00
***/FILE COMMENT END"*****/

```

```

#include "sfr29.h"
#include "define2.h"
#include "declare2.h"

```

```

/*****
*ID          :4.0
*Description  :CAN0 receive interrupt
*-----
*Include     : "sfr29.h"
*            : "define2.h"
*            : "declare2.h"
*-----
*Declaration  :void CAN0_REC(void)
*-----
*Function     :Read CAN0 receive data
*-----
*Arguments    :Nothing
*-----
*Returns     :Nothing
*-----
*Input       :unsigned char rec_data[]      ;CAN receive data
*Output      :unsigned char led_data[]     ;LED display data
*-----
*Call functions :get_message_can0()      ;Get CAN data
*-----
*Note        :
*-----
*History     :
*            :
***/FUNC COMMENT END"*****/
void CAN0_REC(void)

```

```

{
    unsigned char in_rcv_data = 0;

    /* slot14 */
    if(c0mctI[14].receive.newdata == 1){
        get_message_can0(14, &rec_data[0]);
        in_rcv_data = rec_data[0].data.data[0];
    }

    /* slot15 */
    if(c0mctI[15].receive.newdata == 1){
        get_message_can0(15, &rec_data[1]);
        in_rcv_data = rec_data[1].data.data[0];
    }

    led_data[0] = LED_TBL[0x0f & in_rcv_data];
    led_data[1] = LED_TBL[in_rcv_data >> 4];
}

```

7. 参考ドキュメント

ハードウェアマニュアル
M16C/29 グループハードウェアマニュアル Rev.1.0
(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。)

テクニカルアップデート/テクニカルニュース
(最新の情報をルネサス テクノロジホームページから入手してください。)

8. 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/m16c>
M16C ファミリ MCU 技術サポート窓口
E-mail: csc@renesas.com

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2005.01.25	-	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しては、事前にルネサス テクノロジ、ルネサス販売または特約店へ最新の情報をご確認頂きますとともに、ルネサス テクノロジホームページ(<http://www.renesas.com>)などを通じて公開される情報に常にご注意ください。
4. 本資料に記載した情報は、正確を期すため、慎重に制作したのですが万一本資料の記述誤りに起因する損害がお客様に生じた場合には、ルネサス テクノロジはその責任を負いません。
5. 本資料に記載の製品データ、図、表に示す技術的な内容、プログラム及びアルゴリズムを流用する場合は、技術内容、プログラム、アルゴリズム単位で評価するだけでなく、システム全体で十分に評価し、お客様の責任において適用可否を判断してください。ルネサス テクノロジは、適用可否に対する責任を負いません。
6. 本資料に記載された製品は、人命にかかわるような状況の下で使用される機器あるいはシステムに用いられることを目的として設計、製造されたものではありません。本資料に記載の製品を運輸、移動体用、医療用、航空宇宙用、原子力制御用、海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。