

## 要旨

本アプリケーションノートでは、シリアル通信の送受信を想定し、CRC 演算器 (CRC) を用いて CRC データの作成とチェックを行うサンプルプログラムについて説明します。

サンプルプログラムの特長を以下に示します。

- CRC 生成多項式は 16-CCITT ( $X^{16} + X^{12} + X^5 + 1$ ) を使用します。
- シリアル通信の送信側と受信側の CRC 結果が一致したとき、LED0 を点灯させます。

## 動作確認デバイス

RZ/T1 グループ

本アプリケーションノートを他のマイコンへ適用する場合、そのマイコンの仕様にあわせて変更し、十分評価してください。

# 目次

1.	仕様	3
2.	動作環境	4
3.	関連アプリケーションノート	5
4.	周辺機能説明	6
5.	ハードウェア説明	7
5.1	ハードウェア構成例	7
5.2	使用端子一覧	7
6.	ソフトウェア説明	8
6.1	動作概要	8
6.1.1	プロジェクト設定	9
6.2	メモリマップ	9
6.2.1	サンプルプログラムのセクション配置	9
6.2.2	MPU の設定	9
6.2.3	例外処理ベクタテーブル	9
6.3	固定幅整数一覧	9
6.4	定数／エラーコード一覧	10
6.5	構造体／共用体／列挙型一覧	11
6.6	大域変数一覧	11
6.7	関数一覧	12
6.8	関数仕様	12
6.8.1	main	12
6.8.2	crc_init	12
6.8.3	set_send_data	13
6.8.4	check_receive_data	13
6.8.5	R_CRC_Open	14
6.8.6	R_CRC_Set1stValue	15
6.8.7	R_CRC_GetData	15
6.9	フローチャート	16
6.9.1	メイン処理	16
6.9.2	CRC 初期化処理	16
6.9.3	シリアル送信データ設定処理	17
6.9.4	シリアル受信データ確認処理	17
6.9.5	CRC オープン関数	18
6.9.6	CRC データ出力レジスタ初期開始値設定関数	19
6.9.7	CRC データ取得関数	20
7.	サンプルプログラム	21
8.	参考ドキュメント	22

# 1. 仕様

表 1.1 に使用する周辺機能と用途を、図 1.1 に動作環境を示します。

表 1.1 使用する周辺機能と用途

周辺機能	用途
クロック発生回路 (CPG)	CPUクロックおよび低速オンチップオシレータで使用
コンペアマッチタイマ (CMT)	コンペアマッチタイマの周期カウント動作で使用
CRC演算器 (CRC)	シリアル送信データに設定するCRCデータの算出、シリアル受信データを確認する際に用いるCRCデータの算出に使用
エラーコントロールモジュール (ECM)	ERROROUT#端子の初期化
汎用入出力ポート	LEDの点灯および消灯のための端子制御に使用

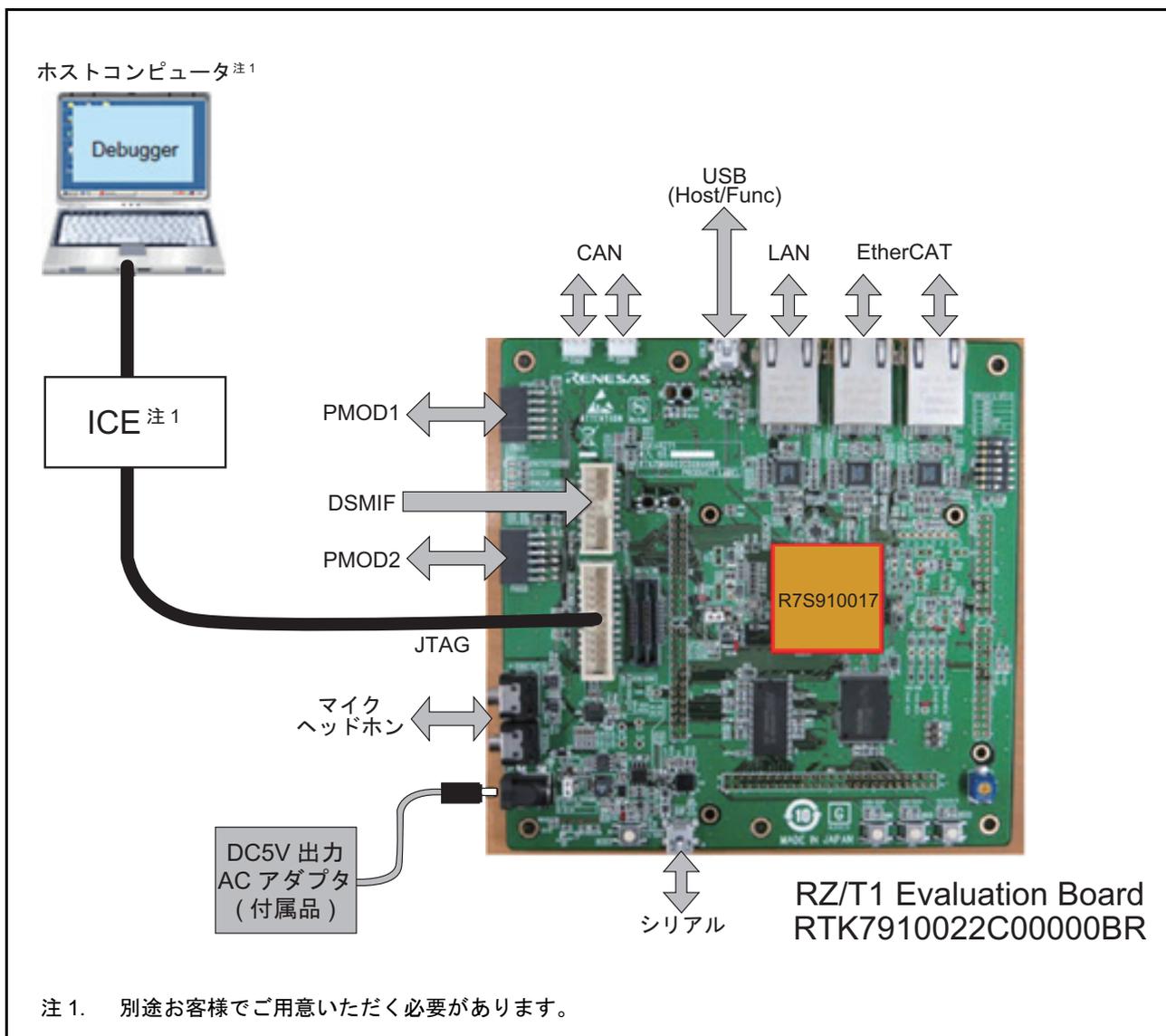


図 1.1 動作環境

## 2. 動作環境

本アプリケーションのサンプルプログラムは、下記の環境を想定しています。

表2.1 動作環境

項目	内容
使用マイコン	RZ/T1グループ
動作周波数	CPUCLK = 450MHz
動作電圧	3.3V
統合開発環境	IARシステムズ製 Embedded Workbench® for Arm Version 8.20.2 Arm製 DS-5™ 5.26.2 RENESAS製 e2studio 6.1.0
動作モード	SPIブートモード 16ビットバスブートモード
使用ボード	RZ/T1 Evaluation Board (RTK7910022C00000BR)
使用デバイス (ボード上で使用する機能)	<ul style="list-style-type: none"><li>NORフラッシュメモリ (CS0、CS1空間に接続) メーカー名 : Macronix International Co., 型名 : MX29GL512FLT2I-10Q</li><li>SDRAM (CS2、CS3空間に接続) メーカー名 : Integrated Silicon Solution Inc、型名 : IS42S16320D-7TL</li><li>シリアルフラッシュメモリ メーカー名 : Macronix International Co., 型名 : MX25L51245G</li></ul>

### 3. 関連アプリケーションノート

本アプリケーションノートに関連するアプリケーションノートを以下に示します。併せて参照してください。

- RZ/T1 グループ初期設定

## 4. 周辺機能説明

クロック発生回路 (CPG)、エラーコントロールモジュール (ECM)、CRC 演算器 (CRC)、汎用入出力ポートについての基本的な内容は、RZ/T1 グループ・ユーザーズマニュアルハードウェア編を参照してください。

## 5. ハードウェア説明

### 5.1 ハードウェア構成例

図 5.1 にハードウェア構成例を示します。

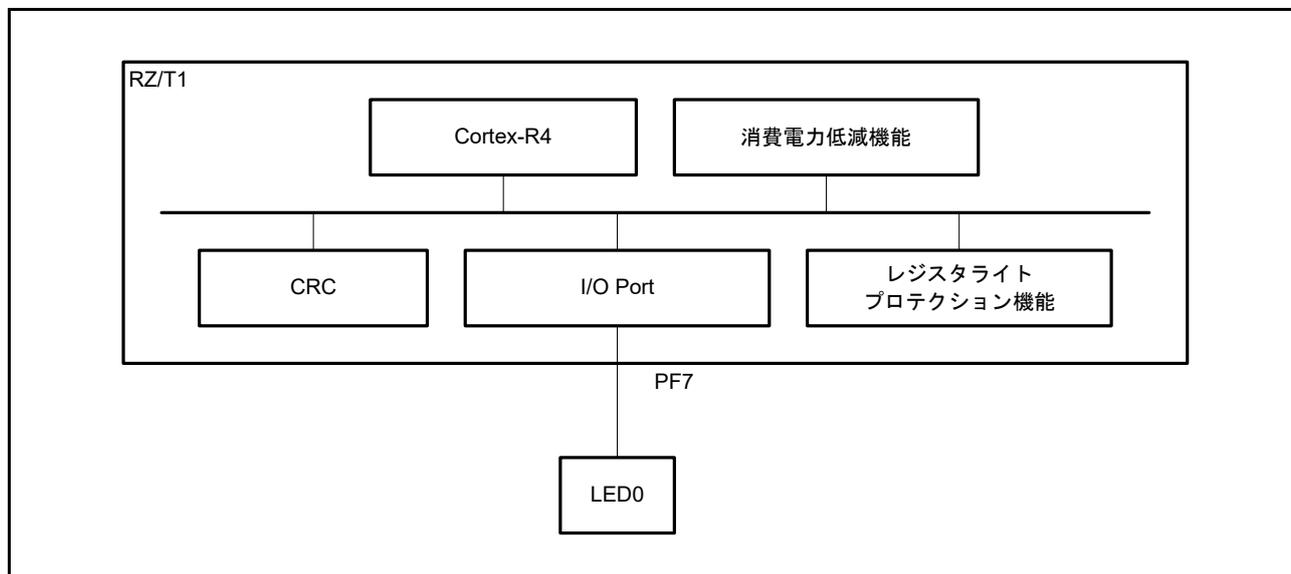


図 5.1 ハードウェア構成例

### 5.2 使用端子一覧

表 5.1 に使用端子と機能を示します。

表 5.1 使用端子と機能

端子名	入出力	内容
MD0	入力	動作モードの選択
MD1	入力	MD0 = "L"、MD1 = "L"、MD2 = "L" (SPI ブートモード)
MD2	入力	MD0 = "L"、MD1 = "H"、MD2 = "L" (16ビットバスブートモード)
PF7	出力	LED0の点灯および消灯

## 6. ソフトウェア説明

### 6.1 動作概要

本サンプルプログラムでは、CRC 演算器 (CRC) の初期設定を行い、シリアル通信の元送信データ 12345678H に対して CRC 演算結果を付与した送信データを作成します。

その後、シリアル受信を想定し、受信データから計算した CRC データと受信した CRC データとを比較します。比較の結果、一致していたとき、LED0 を点灯させます。

本サンプルプログラムの機能概要を表 6.1 動作概要に示します。また、図 6.1 に動作イメージを示します。

表 6.1 動作概要

機能	概要
CRC 演算対象データ	8 ビット
CRC 生成多項式	16 ビット CCITT CRC (16-CCITT)

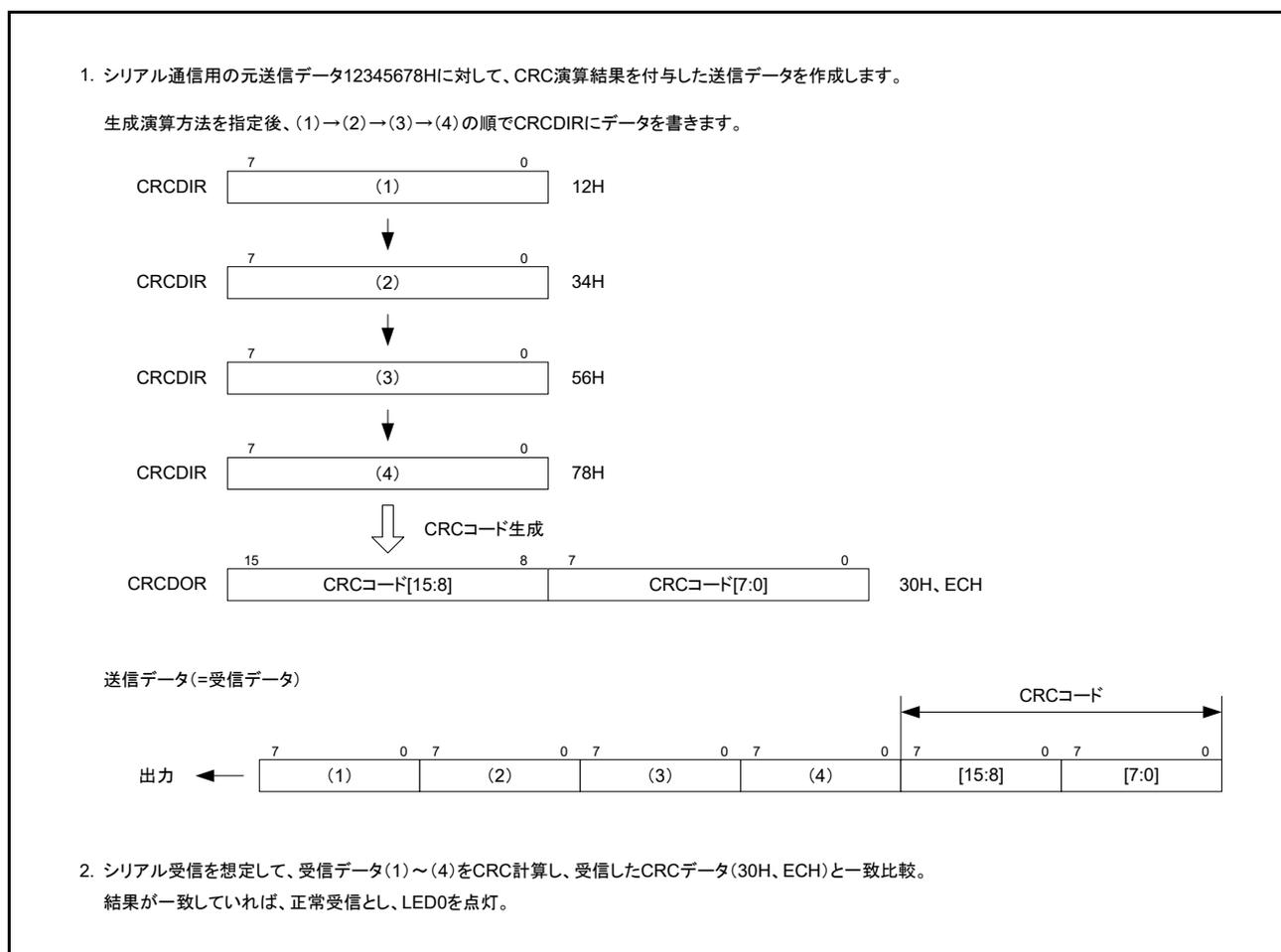


図 6.1 動作イメージ

### 6.1.1 プロジェクト設定

開発環境となる EWARM 上で使用されるプロジェクト設定については、アプリケーションノート「RZ/T1グループ初期設定」に記載しています。

## 6.2 メモリマップ

RZ/T1 グループのアドレス空間と RZ/T1 評価ボードのメモリマッピングについては、アプリケーションノート「RZ/T1 グループ初期設定」に記載しています。

### 6.2.1 サンプルプログラムのセクション配置

サンプルプログラムで使用するセクションおよびサンプルプログラムの初期状態のセクション配置（ロードビュー）、スキッタローディング機能を使用後のセクション配置（実行ビュー）は、アプリケーションノート「RZ/T1 グループ初期設定」に記載しています。

### 6.2.2 MPU の設定

MPU の設定は、アプリケーションノート「RZ/T1 グループ初期設定」に記載しています。

### 6.2.3 例外処理ベクタテーブル

例外処理のベクタテーブルは、アプリケーションノート「RZ/T1 グループ初期設定」に記載しています。

## 6.3 固定幅整数一覧

表 6.2 にサンプルプログラムで使用する固定幅整数を示します。

表 6.2 サンプルプログラムで使用する固定幅整数

シンボル	内容
int8_t	8ビット整数、符号あり（標準ライブラリにて定義）
int16_t	16ビット整数、符号あり（標準ライブラリにて定義）
int32_t	32ビット整数、符号あり（標準ライブラリにて定義）
int64_t	64ビット整数、符号あり（標準ライブラリにて定義）
uint8_t	8ビット整数、符号なし（標準ライブラリにて定義）
uint16_t	16ビット整数、符号なし（標準ライブラリにて定義）
uint32_t	32ビット整数、符号なし（標準ライブラリにて定義）
uint64_t	64ビット整数、符号なし（標準ライブラリにて定義）

## 6.4 定数／エラーコード一覧

表 6.4 にサンプルプログラムで使用する定数を示します。

表6.3 サンプルプログラムで使用する定数

定数名	設定値	内容
CRC_1ST_32_ETHERNET	(0xFFFFFFFF)	CRC生成多項式が32-EthernetのときにCRCデータ出力レジスタ (CRCDOR) を初期化するための定数です。
CRC_1ST_16_CCITT	(0x0000FFFF)	CRC生成多項式が16-CCITTのときにCRCデータ出力レジスタ (CRCDOR) を初期化するための定数です。
CRC_1ST_8_SAE_J1850	(0x000000FF)	CRC生成多項式が8-SAE-J1850のときにCRCデータ出力レジスタ (CRCDOR) を初期化するための定数です。
CRC_1ST_8_0x2F	(0x000000FF)	CRC生成多項式が8-0x2FのときにCRCデータ出力レジスタ (CRCDOR) を初期化するための定数です。
CRC_MASK_16_CCITT	(0x0000FFFF)	CRC生成多項式が16-CCITTのときに読み出したCRCデータ出力レジスタ (CRCDOR) をマスクするための定数です。
CRC_MASK_8_SAE_J1850	(0x000000FF)	CRC生成多項式が8-SAE-J1850のときに読み出したCRCデータ出力レジスタ (CRCDOR) をマスクするための定数です。
CRC_MASK_8_0x2F	(0x000000FF)	CRC生成多項式が8-0x2Fのときに読み出したCRCデータ出力レジスタ (CRCDOR) をマスクするための定数です。
CRC_CFG_PARAM_CHECKING_ENABLE	(1)	CRCのAPI関数にてパラメータチェック許可の有効 (1) /無効 (0) を表します。
CRC_SERIAL_DATA_NUM	(4)	シリアル送受信データの数を表す定数です。

## 6.5 構造体／共用体／列挙型一覧

図 6.2 にサンプルプログラムで使用する構造体／共用体／列挙体を示します。

```

/* API ERROR RETURN CODES */
typedef enum e_crc_err          // CRC API error codes
{
    CRC_SUCCESS=0,
    CRC_ERR_OPEN_IGNORED,      // The module has already been Open()ed
    CRC_ERR_INVALID_ARG,      // Argument is not valid for parameter
    CRC_ERR_NULL_PTR,         // Received null pointer or missing required argument
    CRC_ERR_NOT_OPENED        // Open function has not yet been called
} crc_err_t;

/* Open() and GetData() DEFINITIONS */
typedef enum e_crc_mode        // CRC generation mode
{
    CRC_32_ETHERNET = 0x00u,    // 32-Ethernet
    CRC_16_CCITT    = 0x01u,    // 16-CCITT
    CRC_8_SAE_J1850 = 0x02u,    // 8-SAE J1850
    CRC_8_0x2F     = 0x03u,    // 8-0x2F
    CRC_PROHIBITED
} crc_mode_t;

typedef enum e_crc_width      // CRC input bit width
{
    CRC_32_BIT    = 0x00u,      // 32 bits
    CRC_16_BIT    = 0x10u,      // 16 bits
    CRC_8_BIT     = 0x20u       // 8 bits
} crc_width_t;

typedef struct st_crc_config   // CRC configuration options used in Open function
{
    crc_mode_t      mode;        // CRC generation mode
    crc_width_t     width;       // CRC input bit width
} crc_config_t;

```

図 6.2 サンプルプログラムで使用する構造体／共用体／列挙体

## 6.6 大域変数一覧

表 6.4 に大域変数一覧を示します。

表 6.4 大域変数一覧

型	変数名	内容	使用関数
uint8_t	g_serial_data [CRC_SERIAL_DATA_NUM + 2]	シリアル送受信データ	set_send_data check_receive_data

## 6.7 関数一覧

表 6.5 に関数一覧を示します。

表 6.5 関数一覧

関数名	ページ番号
main	12
crc_init	12
set_send_data	13
check_receive_data	13
R_CRC_Open	14
R_CRC_Set1stValue	15
R_CRC_GetData	15

## 6.8 関数仕様

### 6.8.1 main

#### main

概 要	メイン処理
宣 言	int main(void)
説 明	ポート、ECM、CRC の初期設定を行います。その後、シリアル送信データを作成し、作成した送信データを受信データと見立てて、受信データの CRC 比較を行います。
引 数	なし
リターン値	なし
補 足	なし

### 6.8.2 crc\_init

#### crc\_init

概 要	CRC 初期化処理
ヘッダ	void crc_init(void)
説 明	CRC オープン関数、CRC データ出力レジスタ初期開始値設定関数を呼び出し、CRC の初期化を行います。
引 数	なし
リターン値	なし
補 足	なし

---

### 6.8.3 set\_send\_data

---

#### set\_send\_data

---

概要	シリアル送信データ設定処理
宣言	void set_send_data(void)
説明	シリアル送信するデータの後に続く CRC データを送信データから算出し、シリアル送信データを設定します。
引数	なし
リターン値	なし
補足	なし

### 6.8.4 check\_receive\_data

---

#### check\_receive\_data

---

概要	シリアル受信データ確認処理
宣言	void check_receive_data(void)
説明	CRC データ出力レジスタの初期化をした後、シリアル受信データ中の CRC データと算出した CRC データとを比較します。比較の結果、CRC が一致したとき LED0 を点灯させます。
引数	なし
リターン値	なし
補足	なし

## 6.8.5 R\_CRC\_Open

## R\_CRC\_Open

概 要	CRC オープン関数
ヘッダ	r_crc_if.h
宣 言	crc_err_t R_CRC_Open (void * const p_cfg)
説 明	CRC 関連レジスタを初期化します。
引 数	void * const p_cfg      CRC コントロールレジスタに設定するデータ群を格納した ポインタです。 CRC 生成方式指定 CRC_32_ETHERNET CRC_16_CCITT CRC_8_SAE_J1850 CRC_8_0x2F CRC 入力ビット幅指定 CRC_32_BIT CRC_16_BIT CRC_8_BIT
リターン値	オープン関数の実行結果を返します。 CRC_SUCCESS : CRC が初期化されました CRC_ERR_OPEN_IGNORED : モジュールはすでに開かれています CRC_ERR_INVALID_ARG : p_cfg 構造体の要素に無効な値が含まれています CRC_ERR_NULL_PTR : p_cfg ポインタが NULL です
補足	r_crc_config.h で定義される CRC_CFG_PARAM_CHECKING_ENABLE を 1 とすること で、引数パラメータのチェック処理を有効にします。

## 6.8.6 R\_CRC\_Set1stValue

## R\_CRC\_Set1stValue

概要	CRC データ出力レジスタ初期開始値設定関数
ヘッダ	r_crc_if.h
宣言	crc_err_t R_CRC_Set1stValue(crc_mode_t const mode)
説明	CRC データ出力レジスタに初期開始値を設定します。
引数	crc_mode_t const mode    CRC 生成方法を指定します。 CRC_32_ETHERNET CRC_16_CCITT CRC_8_SAE_J1850 CRC_8_0x2F
リターン値	CRC データ出力レジスタ初期開始値設定関数の実行結果を返します。 CRC_SUCCESS : 正常に完了しました CRC_ERR_INVALID_ARG : 引数の値が無効です CRC_ERR_NOT_OPENED : Open が読み出されていません
補足	r_crc_config.h で定義される CRC_CFG_PARAM_CHECKING_ENABLE を 1 とすることで、引数パラメータのチェック処理を有効にします。

## 6.8.7 R\_CRC\_GetData

## R\_CRC\_GetData

概要	CRC データ取得関数
ヘッダ	r_crc_if.h
宣言	crc_err_t R_CRC_GetData(crc_mode_t const mode, uint32_t* p_result)
説明	CRC 演算結果を取得します。
引数	crc_mode_t const mode    CRC 生成方法を指定します。 CRC_32_ETHERNET CRC_16_CCITT CRC_8_SAE_J1850 CRC_8_0x2F  uint32_t* p_result        CRC 演算結果の格納位置のポインタです。
リターン値	CRC データ取得関数の実行結果を返します。 CRC_SUCCESS : 正常に完了しました CRC_ERR_INVALID_ARG : 引数の値が無効です CRC_ERR_NULL_PTR : p_result ポインタが NULL です CRC_ERR_NOT_OPENED : Open が読み出されていません
補足	r_crc_config.h で定義される CRC_CFG_PARAM_CHECKING_ENABLE を 1 とすることで、引数パラメータのチェック処理を有効にします。

## 6.9 フローチャート

### 6.9.1 メイン処理

図 6.3 にメイン処理のフローチャートを示します。

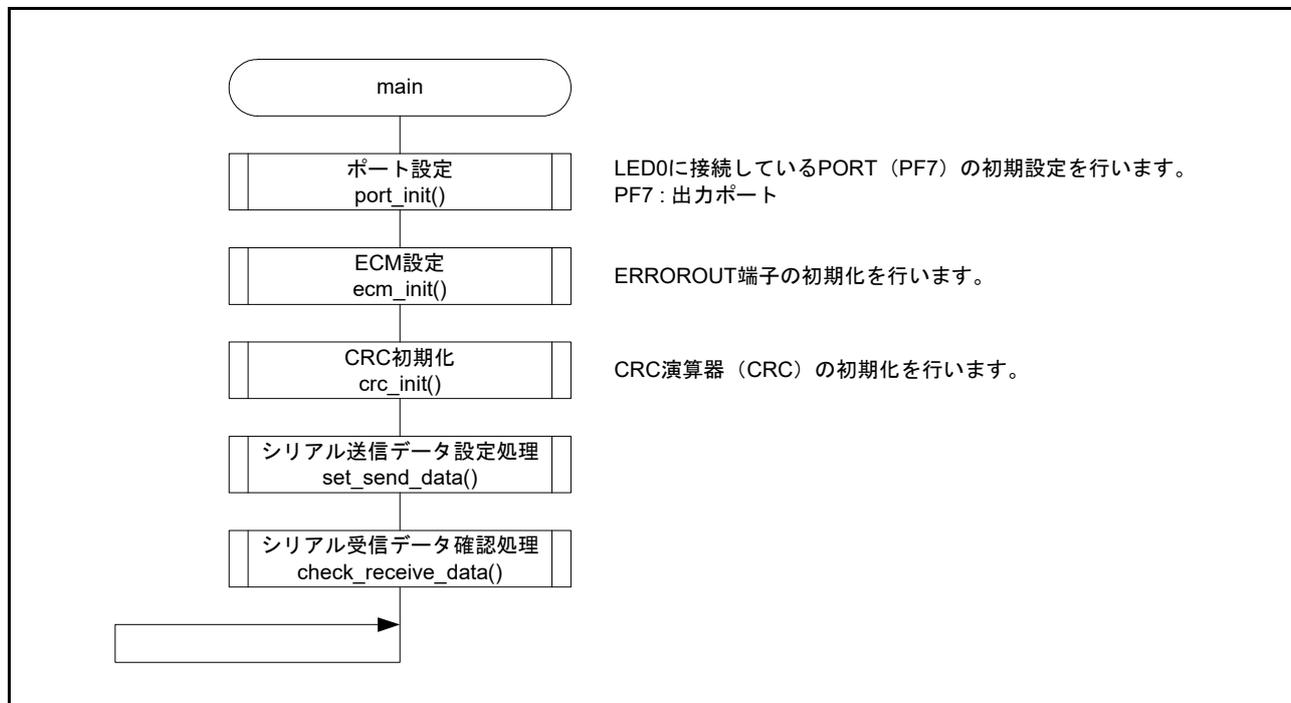


図 6.3 メイン処理

### 6.9.2 CRC 初期化処理

図 6.4 に CRC 初期化処理のフローチャートを示します。

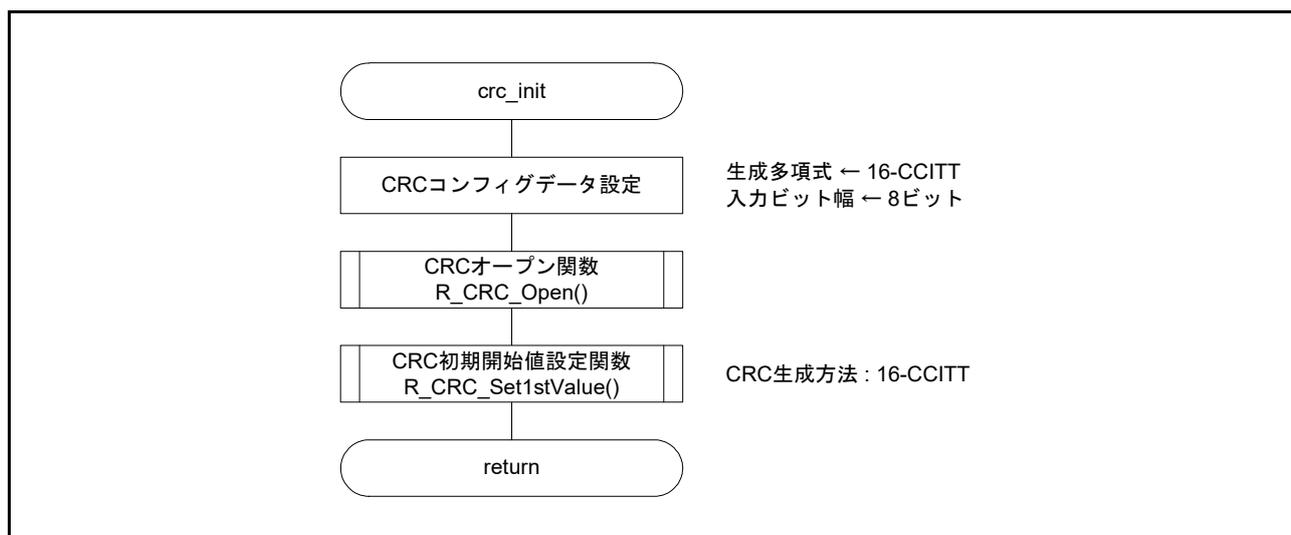


図 6.4 CRC 初期化処理

### 6.9.3 シリアル送信データ設定処理

図 6.5 にシリアル送信データ設定処理のフローチャートを示します。

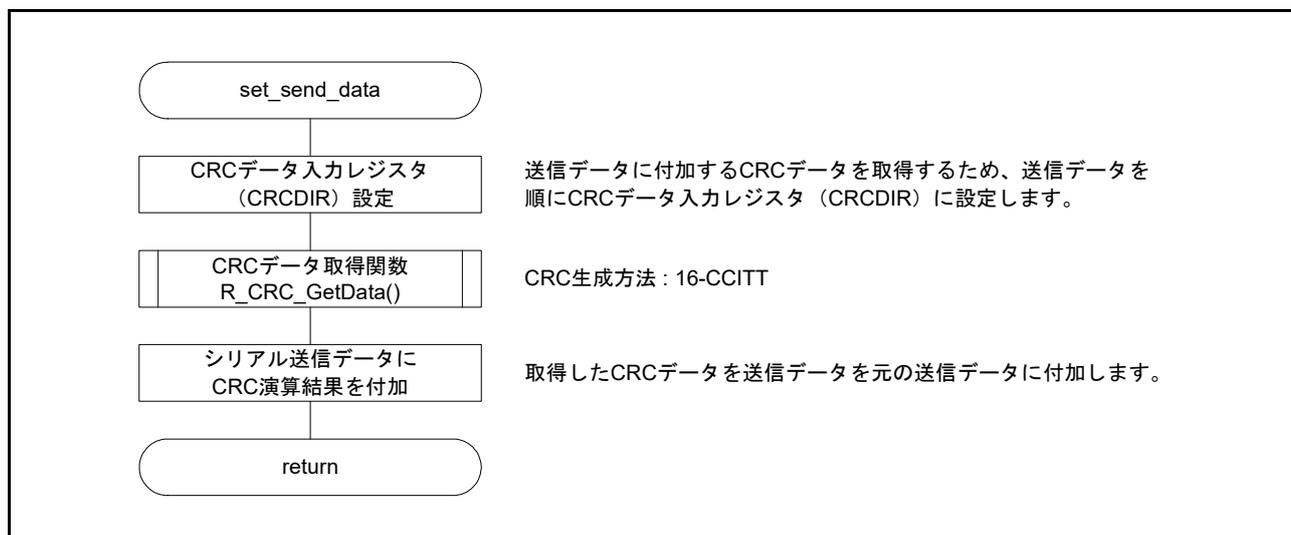


図 6.5 シリアル送信データ設定処理

### 6.9.4 シリアル受信データ確認処理

図 6.6 にシリアル受信データ確認処理のフローチャートを示します。

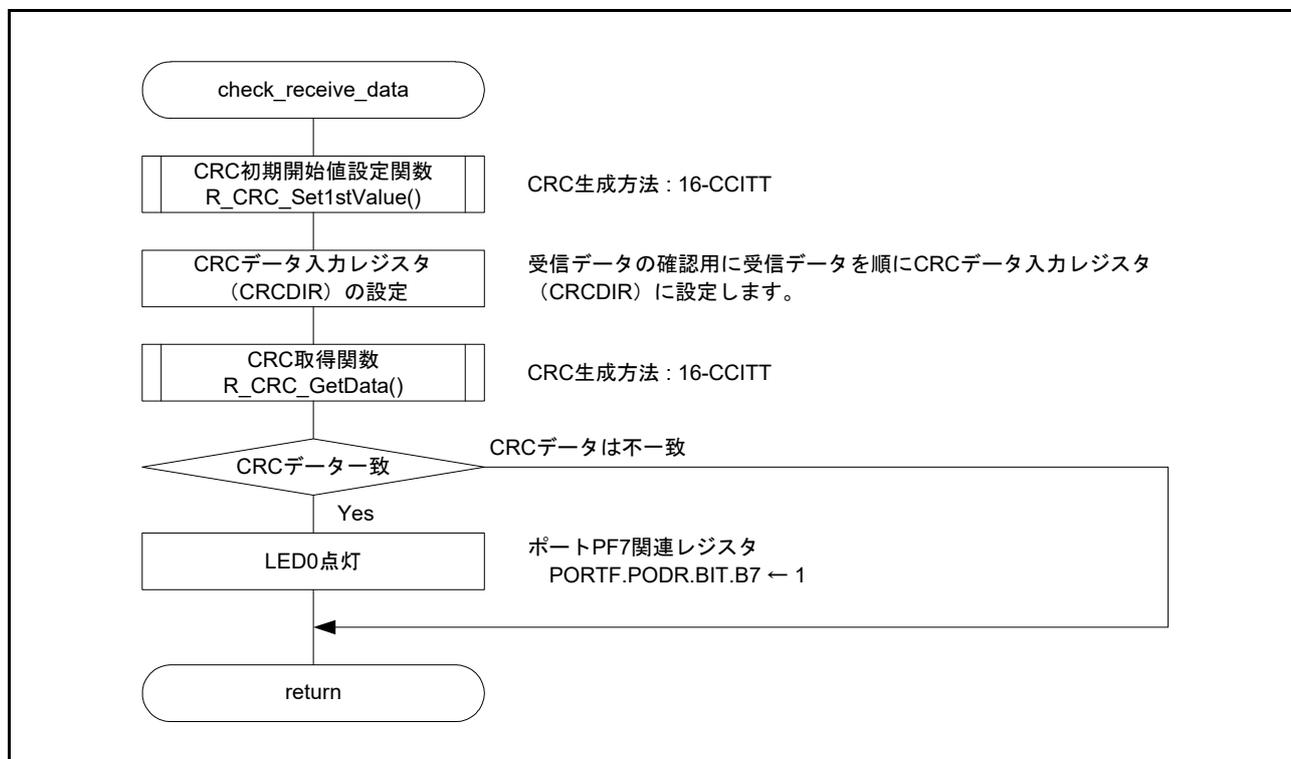


図 6.6 シリアル受信データ確認処理

### 6.9.5 CRC オープン関数

図 6.7 に CRC オープン関数のフローチャートを示します。

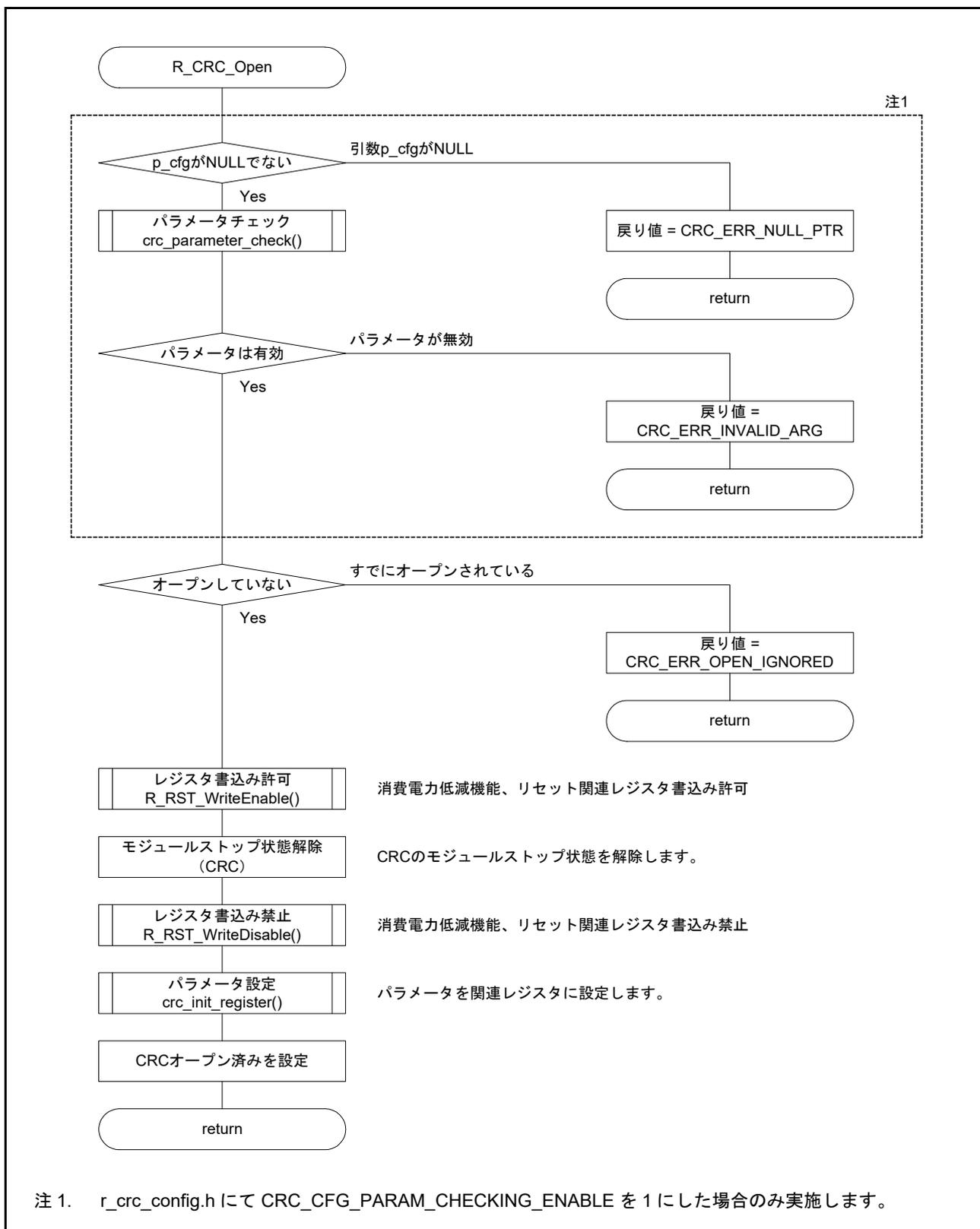


図 6.7 CRC オープン関数

### 6.9.6 CRC データ出力レジスタ初期開始値設定関数

図 6.8 に CRC データ出力レジスタ初期開始値設定関数のフローチャートを示します。

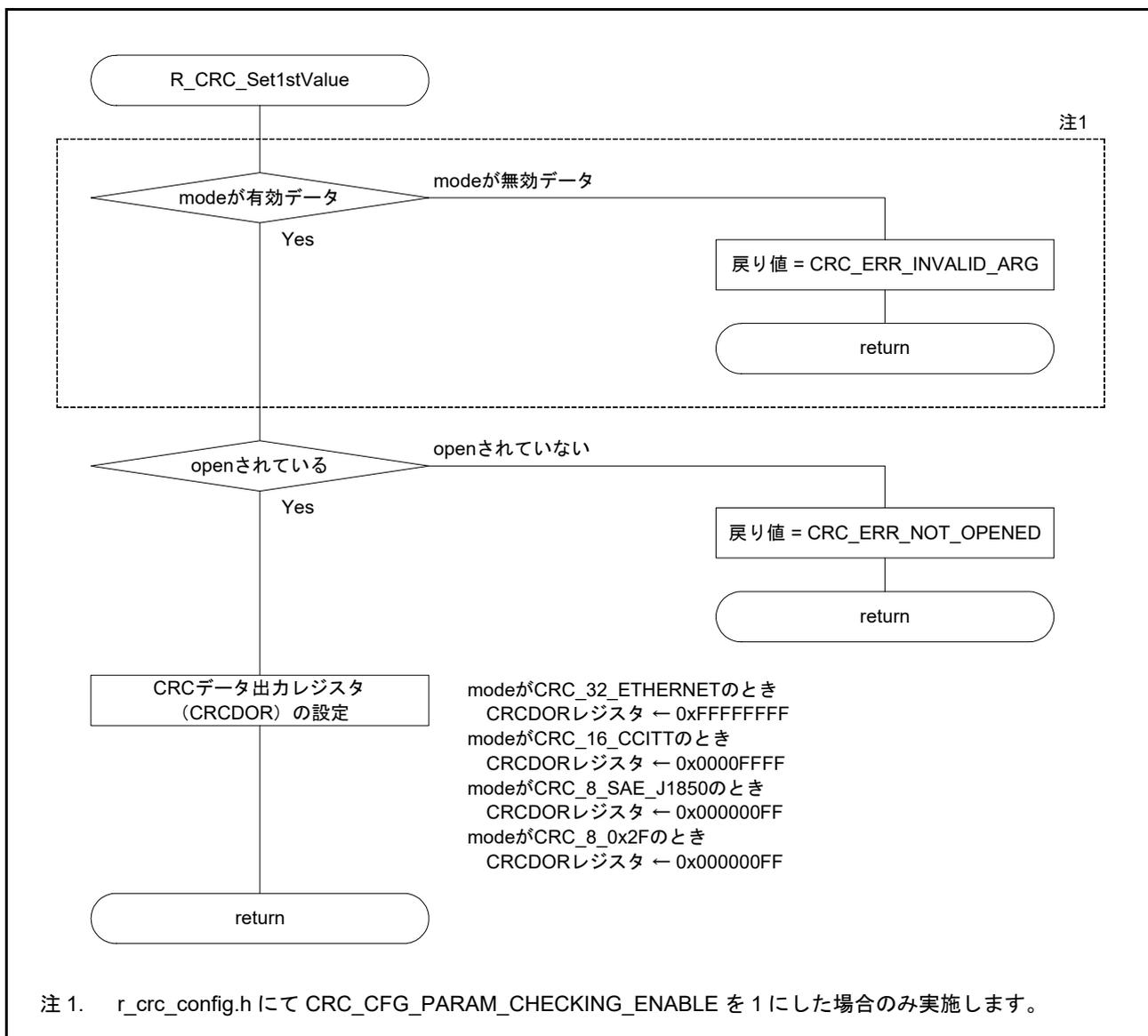


図 6.8 CRC データ出力レジスタ初期開始値設定関数

### 6.9.7 CRC データ取得関数

図 6.9 に CRC データ取得関数のフローチャートを示します。

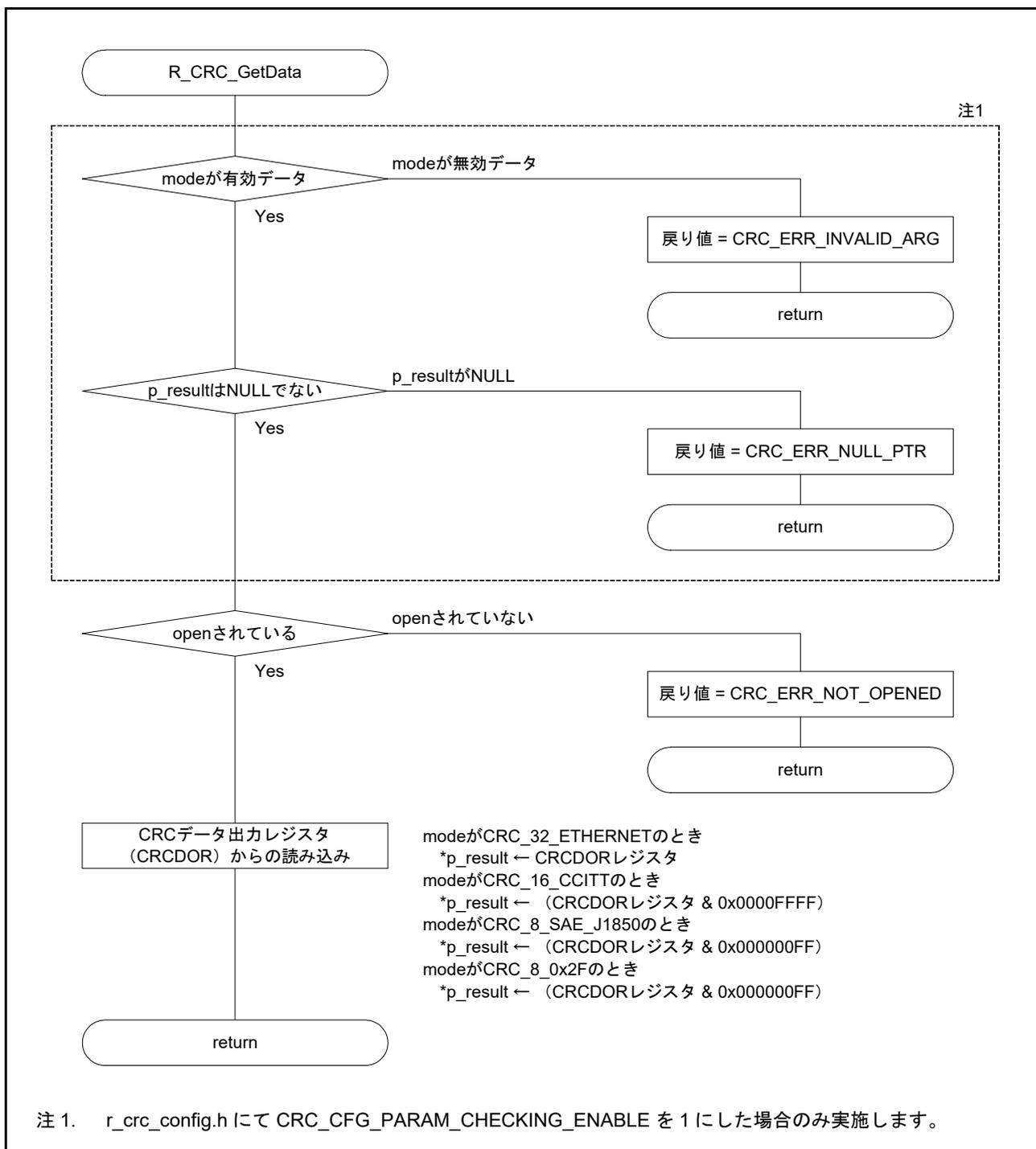


図 6.9 CRC データ取得関数

## 7. サンプルプログラム

サンプルプログラムは、ルネサス エレクトロニクスホームページから入手してください。

## 8. 参考ドキュメント

- ユーザーズマニュアル：ハードウェア

RZ/T1 グループ ユーザーズマニュアルハードウェア編

(最新版をルネサス エレクトロニクスホームページから入手してください。)

RZ/T1 Evaluation Board RTK7910022C00000BR ユーザーズマニュアル

(最新版をルネサス エレクトロニクス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 エレクトロニクスホームページから入手してください。)

- ユーザーズマニュアル：開発環境

IAR 統合開発環境 (IAR Embedded Workbench® for Arm) に関しては、IAR ホームページから入手してください。

(最新版を IAR ホームページから入手してください。)

## ホームページとサポート窓口

ルネサス エレクトロニクスホームページ

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

お問い合わせ先

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

## 改訂記録

## CRC演算器（CRC）アプリケーションノート

Rev.	発行日	改訂内容	
		ページ	ポイント
0.10	2015.02.18	—	初版発行
0.20	2015.03.10	10	6.3 固定幅整数一覧 表6.3 サンプルプログラムで使用する固定幅整数 修正
1.00	2015.04.10	—	Web掲載に際しRevのみ変更
1.10	2015.07.06	2. 動作環境	
		4	表2.1 動作環境 統合開発環境 表記一部修正、追加
		6. ソフトウェア説明	
		9	6.2.4 説明文 参照を追加
		9	表6.2 タイトルを一部追加
1.20	2015.12.03	2. 動作環境	
		4	表2.1 動作環境 統合開発環境 一部修正
		6. ソフトウェア説明	
		9	6.2.4 必要メモリサイズ 削除
1.30	2017.04.05	2. 動作環境	
		4	表2.1 動作環境 統合開発環境の内容変更
		6. ソフトウェア説明	
		—	6.2.4 必要メモリサイズ 削除
		2. 動作環境	
1.40	2018.06.07	4	表2.1 動作環境 統合開発環境の内容変更
		5. ハードウェア説明	
		7	図5.1 ハードウェア構成例 モジュール名変更
		8. 参考ドキュメント	
		22	IAR 統合開発環境名変更

すべての商標および登録商標は、それぞれの所有者に帰属します。

## 製品ご使用上の注意事項

ここでは、マイコン製品全体に適用する「使用上の注意事項」について説明します。個別の使用上の注意事項については、本ドキュメントおよびテクニカルアップデートを参照してください。

### 1. 未使用端子の処理

【注意】未使用端子は、本文の「未使用端子の処理」に従って処理してください。

CMOS 製品の入力端子のインピーダンスは、一般に、ハイインピーダンスとなっています。未使用端子を開放状態で動作させると、誘導現象により、LSI 周辺のノイズが印加され、LSI 内部で貫通電流が流れたり、入力信号と認識されて誤動作を起こす恐れがあります。未使用端子は、本文「未使用端子の処理」で説明する指示に従い処理してください。

### 2. 電源投入時の処置

【注意】電源投入時は、製品の状態は不定です。

電源投入時には、LSI の内部回路の状態は不確定であり、レジスタの設定や各端子の状態は不定です。

外部リセット端子でリセットする製品の場合、電源投入からリセットが有効になるまでの期間、端子の状態は保証できません。

同様に、内蔵パワーオンリセット機能を使用してリセットする製品の場合、電源投入からリセットのかかる一定電圧に達するまでの期間、端子の状態は保証できません。

### 3. リザーブアドレス（予約領域）のアクセス禁止

【注意】リザーブアドレス（予約領域）のアクセスを禁止します。

アドレス領域には、将来の機能拡張用に割り付けられているリザーブアドレス（予約領域）がありません。これらのアドレスをアクセスしたときの動作については、保証できませんので、アクセスしないようにしてください。

### 4. クロックについて

【注意】リセット時は、クロックが安定した後、リセットを解除してください。

プログラム実行中のクロック切り替え時は、切り替え先クロックが安定した後に切り替えてください。

リセット時、外部発振子（または外部発振回路）を用いたクロックで動作を開始するシステムでは、クロックが十分安定した後、リセットを解除してください。また、プログラムの途中で外部発振子（または外部発振回路）を用いたクロックに切り替える場合は、切り替え先のクロックが十分安定してから切り替えてください。

### 5. 製品間の相違について

【注意】型名の異なる製品に変更する場合は、製品型名ごとにシステム評価試験を実施してください。

同じグループのマイコンでも型名が違っていると、内部 ROM、レイアウトパターンの相違などにより、電気的特性の範囲で、特性値、動作マージン、ノイズ耐量、ノイズ輻射量などが異なる場合があります。型名が異なる製品に変更する場合は、個々の製品ごとにシステム評価試験を実施してください。

## ご注意書き

1. 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・システムの設計において、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て生じた損害（お客様または第三者いずれに生じた損害も含みます。以下同じです。）に関し、当社は、一切その責任を負いません。
2. 当社製品、本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、著作権その他の知的財産権に対する侵害またはこれらに関する紛争について、当社は、何らの保証を行うものではなく、また責任を負うものではありません。
3. 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。
4. 当社製品を、全部または一部を問わず、改造、改変、複製、リバースエンジニアリング、その他、不適切に使用しないでください。かかる改造、改変、複製、リバースエンジニアリング等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。
5. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。

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、産業用ロボット等

高品質水準： 輸送機器（自動車、電車、船舶等）、交通制御（信号）、大規模通信機器、  
金融端末基幹システム、各種安全制御装置等

当社製品は、データシート等により高信頼性、Harsh environment向け製品と定義しているものを除き、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（宇宙機器と、海底中継器、原子力制御システム、航空機制御システム、プラント基幹システム、軍事機器等）に使用されることを意図しておらず、これらの用途に使用することは想定していません。たとえ、当社が想定していない用途に当社製品を使用したことにより損害が生じても、当社は一切その責任を負いません。

6. 当社製品をご使用の際は、最新の製品情報（データシート、ユーザーズマニュアル、アプリケーションノート、信頼性ハンドブックに記載の「半導体デバイスの使用上の一般的な注意事項」等）をご確認の上、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他指定条件の範囲内でご使用ください。指定条件の範囲を超えて当社製品をご使用された場合の故障、誤動作の不具合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。
7. 当社は、当社製品の品質および信頼性の向上に努めていますが、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、データシート等において高信頼性、Harsh environment向け製品と定義しているものを除き、耐放射線設計を行っておりません。仮に当社製品の故障または誤動作が生じた場合であっても、人身事故、火災事故その他社会的損害等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、お客様の機器・システムとしての出荷保証を行ってください。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、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
8. 当社製品の環境適合性等の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。かかる法令を遵守しないことにより生じた損害に関して、当社は、一切その責任を負いません。
9. 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・システムに使用することはできません。当社製品および技術を輸出、販売または移転等する場合は、「外国為替及び外国貿易法」その他日本国および適用される外国の輸出管理関連法規を遵守し、それらの定めるところに従い必要な手続きを行ってください。
10. お客様が当社製品を第三者に転売等される場合には、事前に当該第三者に対して、本ご注意書き記載の諸条件を通知する責任を負うものといたします。
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禁じます。
12. 本資料に記載されている内容または当社製品についてご不明な点がございましたら、当社の営業担当者までお問合せください。

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が直接的、間接的に支配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。

(Rev.4.0-1 2017.11)



ルネサスエレクトロニクス株式会社

■営業お問合せ窓口

<http://www.renesas.com>

※営業お問合せ窓口の住所は変更になることがあります。最新情報につきましては、弊社ホームページをご覧ください。

ルネサス エレクトロニクス株式会社 〒135-0061 東京都江東区豊洲3-2-24（豊洲フォレシア）

■技術的なお問合せおよび資料のご請求は下記へどうぞ。  
総合お問合せ窓口：<https://www.renesas.com/contact/>