RL78/G16

Modbus ASCII/RTU

はじめに

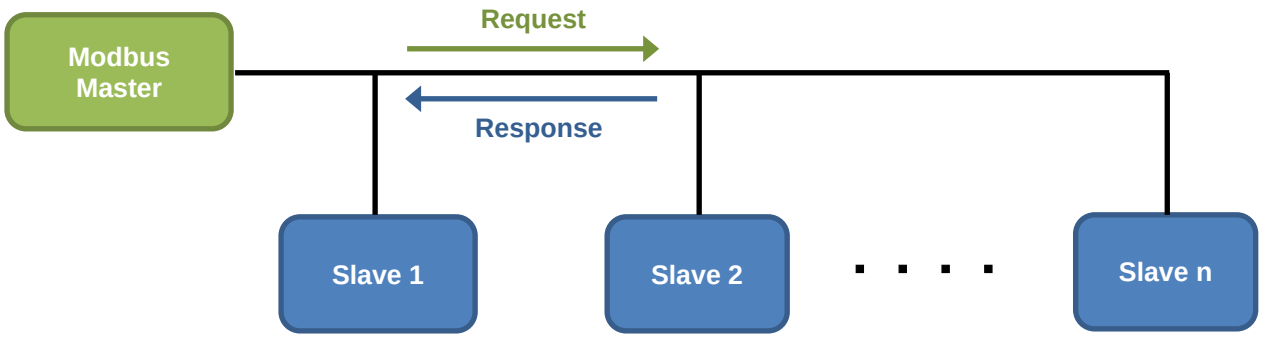
Modbus は、Modicon Inc.(AEG Schneider Automation International S.A.S)が PLC 用に開発した通信プロトコルで、PLC のみならず電子機器間においてもデータの受け渡しの目的で使用されています。仕様が一般公開されており利用が無料であること、実装が比較的容易であること等により、ファクトリーオートメーションやプラントオートメーションの分野で広く普及しています。そのため、産業用機器を接続するのに最も一般的なシリアル通信プロトコルとなっています。

例えば工場内では、各種有線通信または無線通信のゲートウェイを介して、機器の接続に Modbus が使用されています。Modbus が採用されているマスタ機器は、ゲートウェイ, HMI, SCADA(Supervisor Control And Data Acquisition), PLC 等になります。また、スレーブ機器は、I/O コントロール, 温度や湿度の計測機, 計量器, Frequency Transformer, モータユニット等に使用されています。

シリアル通信をベースとした Modbus の伝送モードには Modbus ASCII（American Standard Code for Information Interchange）と Modbus RTU（Remote Terminal Unit）の 2 種類が存在します。

Modbus ASCII は伝送データが ASCII 文字列となり、1 バイトデータが 2 文字の ASCII コードに変換されます。データの区切りには終端文字が付加されるため、Modbus RTU に比べてデータ量が多く伝送時間が長くなりますが、伝送データの解析が容易になります。一方、Modbus RTU は伝送データの変換を行わず、バイナリで伝送されます。データの区切りは 3.5 文字分以上の無通信時間で判定する必要があるため、Modbus ASCII に比べてデータ量が少なく伝送時間が短くなりますが、伝送データの解析にはタイマを使用した処理が必要になります。

Modbus の通信方式はシングルマスタ/マルチスレーブ方式で、マスタからの要求にスレーブが応答します。Modbus のネットワークには 1 台のマスタと 1～247 台のスレーブが存在し、スレーブはネットワーク内でユニークなアドレス（1～247）を持っている必要があります。また、同一ネットワークでも異なるデータフォーマット同士では通信を行うことができないため、1 つのネットワーク内で Modbus ASCII、Modbus RTU のどちらかに統一する必要があります。



Modbus ASCII/RTU の物理層には RS-485 規格が使用されることが多く、RL78 マイコンとルネサス製 RS-485 トランシーバを UART で接続することにより、Modbus ASCII/RTU で通信するスレーブ機器を容易に実現できます。特に低消費電力動作、 I/O コントロール、温度や湿度の計測等のシンプルな機能には、RL78 マイコンが最適であり、本サンプルプログラムを利用して早期に開発がおこなえます。

要旨

本アプリケーションノートでは、RL78 マイコンとルネサス製 RS-485 トランシーバを組み合わせ、Modbus ASCII/RTU でマスタ/スレーブ機能を実現するサンプルプログラムについて説明します。

動作確認デバイス

RL78/G16, RAA788152 (RS-485 トランシーバ), PmodUSBUART

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、十分に評価してください。

目次

1. 仕様

本アプリケーションノートでは、UART 通信を用いた Modbus 通信の使用例を示しています。GPIO を用いて RAA788152 の送受信許可を設定し UART0 端子から Modbus フレームの送受信を行います。送受信の文字間インターバル、フレーム間インターバル計測は TAU0 を使用します。

表 1-1 使用する周辺機能と用途

周辺機能	用途	
PORT	P13	RAA788152 の送受信許可を制御する
SAU0	UART0	Modbus フレームの送受信を行う
SAU1	UART2	Modbus フレームの送受信ログ、エラーログを出力する
TAU0	Channel 0	Modbus フレームの文字間インターバル、フレーム間インターバルの計測に使用する
	Channel 1	マスタモード時、スレーブからの応答待ちタイムアウトに使用する

2. 動作確認条件

本アプリケーションノートのサンプルコードは、下記の条件で動作を確認しています。

表 2-1 動作条件

項目	内容
使用マイコン	RL78/G16(R5F121BCAFP)
使用ボード	● RL78/G16 Fast Prototyping Board (RTK5RLG160C00000BJ) ● RS-485 Transceivers Evaluation Board (RTKA788152DE0000BU)
動作周波数	● 高速オンチップ・オシレータ・クロック：16MHz ● CPU／周辺ハードウェアクロック：16MHz
動作電圧	5V
統合開発環境 (CS+)	ルネサス エレクトロニクス製 CS+ for CC V8.10.00
C コンパイラ (CS+)	ルネサス エレクトロニクス製 CC-RL V1.12.01
統合開発環境 (e2 studio)	ルネサス エレクトロニクス製 e2 studio V23.7.0
C コンパイラ (e2 studio)	ルネサス エレクトロニクス製 CC-RL V1.12.01
周辺デバイス構成例	4. ハードウェア説明を参照してください。

注意 RL78/G16 Fast Prototyping Board で、COM port デバッグ使用する場合、統合開発環境での Debug hardware: COM port(RL78)のターゲット・ボードとの接続は、プルダウンから RL78/G16 Fast Prototyping Board に割り当てられた COM ポート番号を選択してください。

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。

RL78/G16 シリアル・アレイ・ユニット (UART 通信) (R01AN6903) アプリケーションノート

RL78/G16 タイマ・アレイ・ユニット (インターバル・タイマ) (R01AN7023) アプリケーションノート

4. ハードウェア説明

4.1 ハードウェア構成例

RL78/G16 Fast Prototyping Board（RTK7RLG160C00000BJ）と、RS-485 トランシーバとして 5V Half-Duplex RS-485 Transceivers Evaluation Board（RTKA788152DE0000BU）を使用する前提でのハードウェア構成例を記載します。別途 PmodUSBUART を使用することでログ出力を行うことができます。ログの仕様については 5.13 ログ仕様を参照してください。

4.1.1 RL78 – PC(GUI)環境

図 4-1 に RL78 – PC(GUI)環境の構成例を記します。

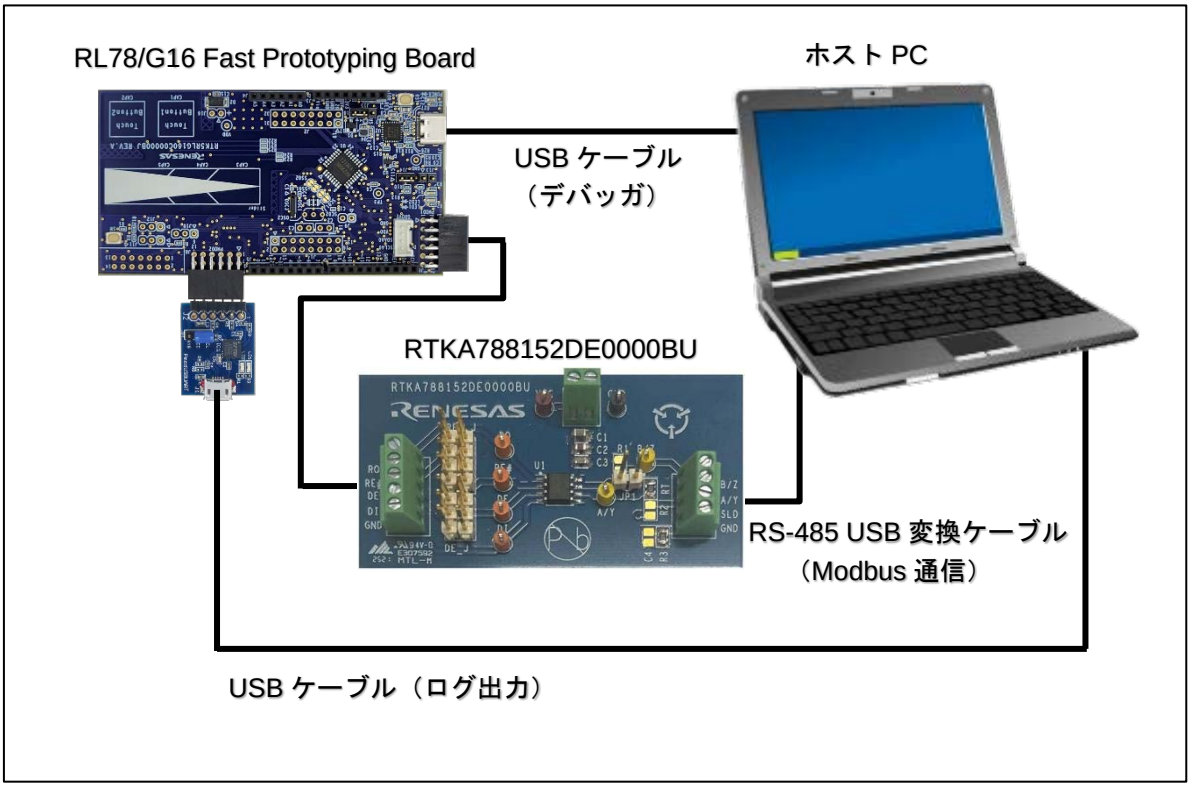


図 4-1 RL78 – PC(GUI)環境の構成例

4.1.2 RL78 – RL78 環境

図 4-2 に RL78 – RL78 環境の構成例を記します。

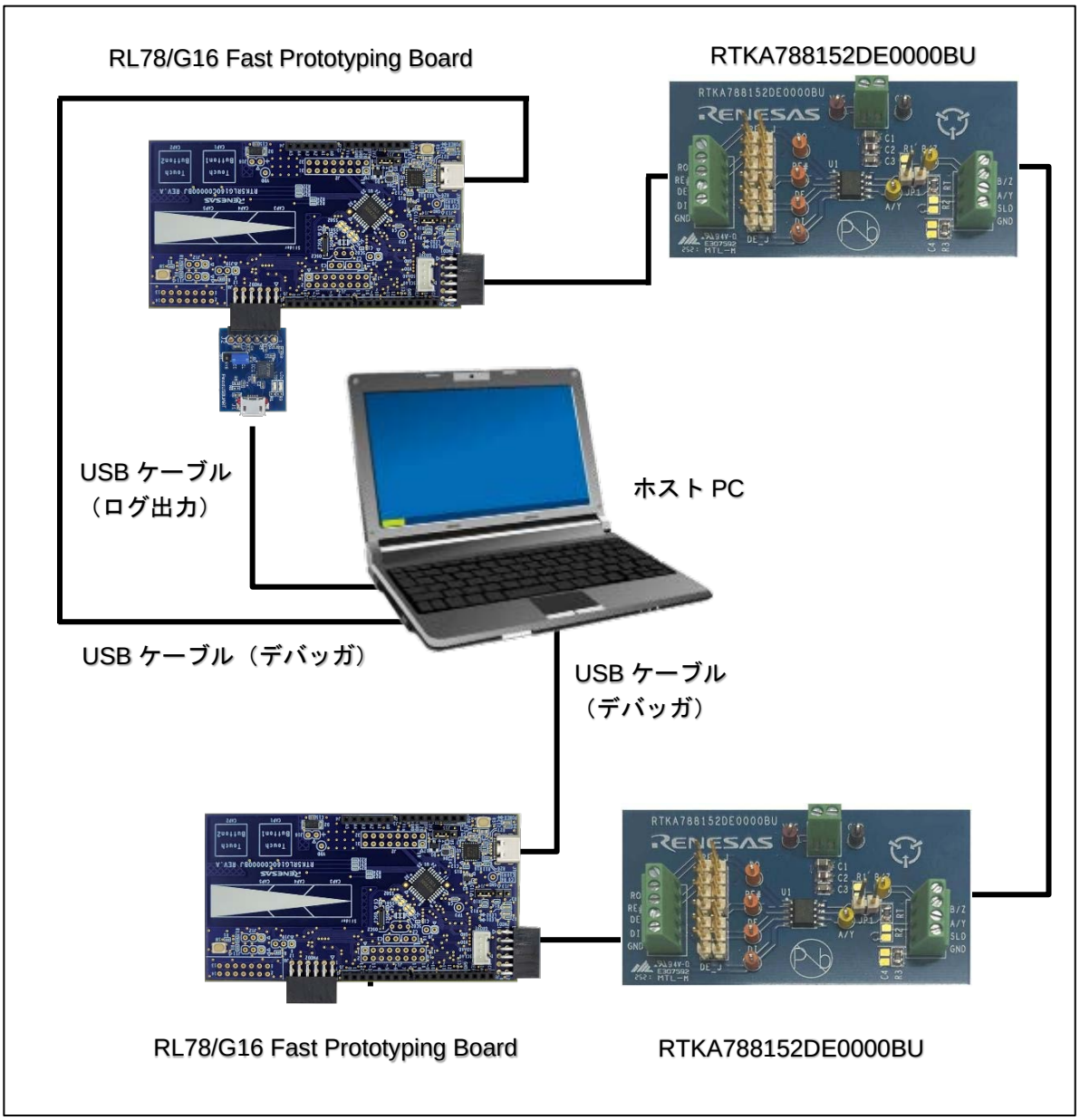


図 4-2 RL78 – RL78 環境の構成例

4.1.3 RL78 – ユーザ Modbus 機器環境

図 4-3 に RL78 – ユーザ Modbus 機器環境の構成例を記します。

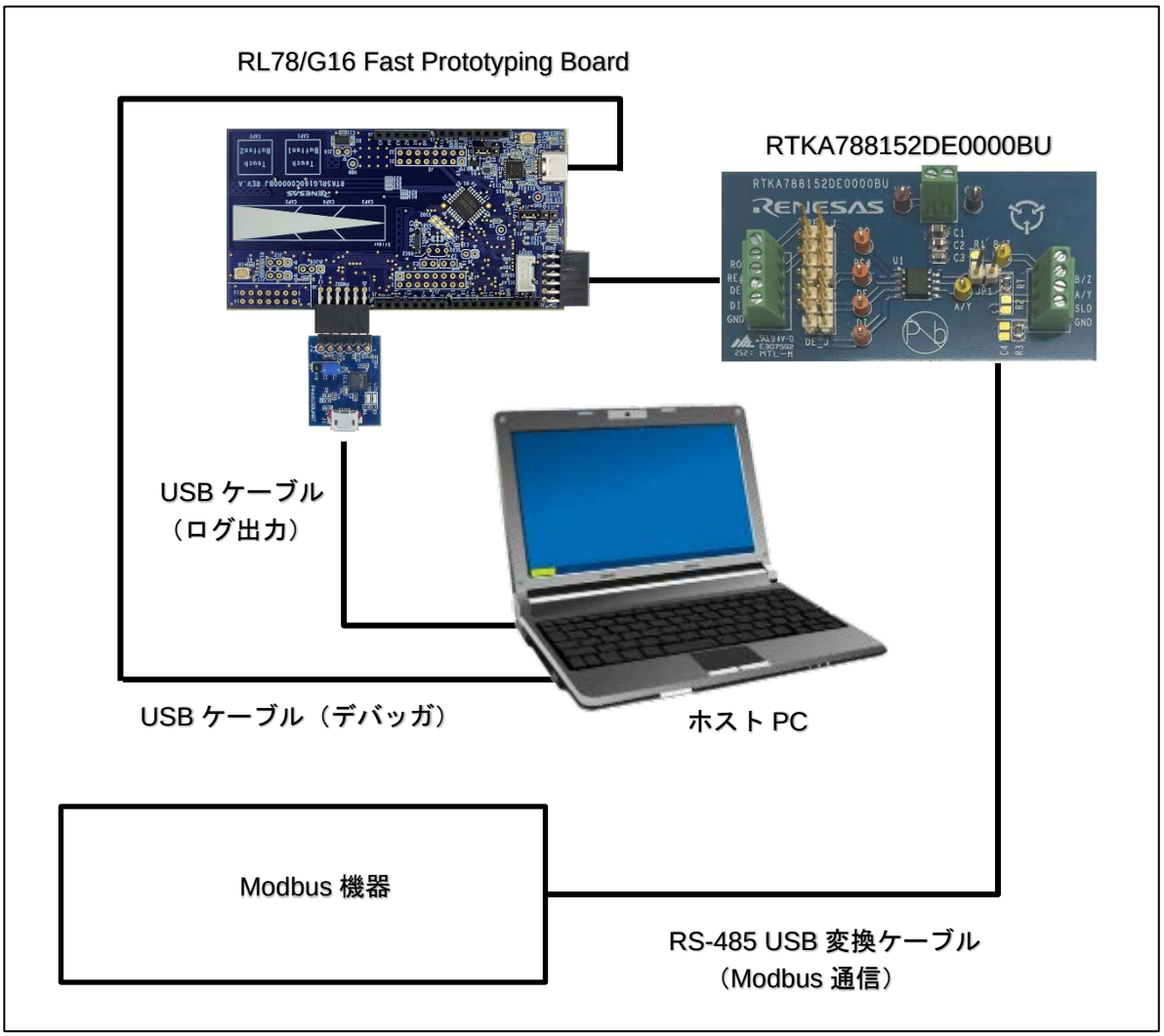


図 4-3 RL78 – ユーザ Modbus 機器環境の構成例

4.2 RL78/G16 Fast Prototyping Board - RTKA788152DE0000BU 接続端子表

表 4-1 に RL78/G16 Fast Prototyping Board - RTKA788152DE0000BU 接続端子表を示します。

表 4-1 RL78/G16 Fast Prototyping Board - RTKA788152DE0000BU 接続端子表

RL78/G16 Fast Prototyping Board			RTKA788152DE0000BU		
コネクタ名	ピン番号	端子名	コネクタ名	ピン番号	端子名
Pmod™1	1	P13	IN	2, 3	RE#, DE
	2	P03/TxD0		4	DI
	3	P04/RxD0		1	RO
	5	GND	POWER	2	GND
	6	TARGET_VCC		1	VCC

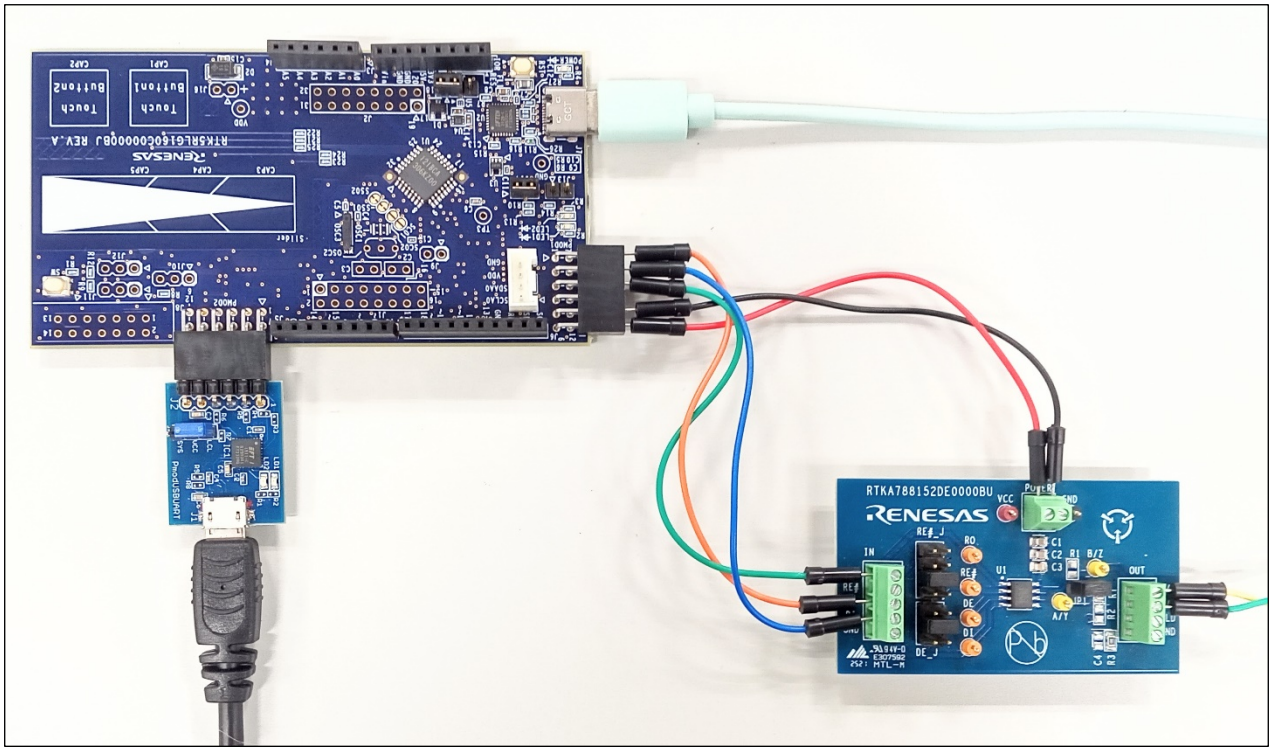


図 4-4 RL78/G16 Fast Prototyping Board - RTKA788152DE0000BU 接続写真

4.3 RTKA788152DE0000BU のジャンパピン設定

RTKA788152DE0000BU の受信許可(RE#)／送信許可(DE)を 1 つの GPIO で行うため、RTKA788152DE0000BU の RE#_J コネクタの 5pin と 6pin をショートに設定してください。また、DE_J コネクタは 3pin と 4pin をショートに設定してください。この設定により、RE#と DE はショートされます。また、RS-485 終端要件を満たすために、最初または最終端では終端抵抗のため JP1 をショートに設定してください。

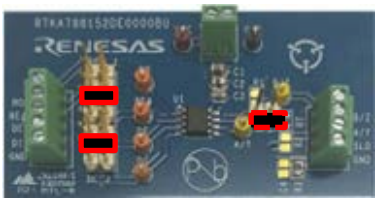


図 4-5 RTKA788152DE0000BU ジャンパピン接続写真

4.4 使用端子一覧

表 4-2 に使用端子と機能を示します。

表 4-2 使用端子と機能

端子名	入出力	用途
P13	出力	RAA788152 の送受信許可を制御
P03/TxD0	出力	Modbus フレームの送信
P04/RxD0	入力	Modbus フレームの受信
P21/TxD2	出力	Modbus フレームの送受信ログ、エラーログを出力

5. ソフトウェア説明

5.1 動作概要

本サンプルコードでは、UART の送受信機能を使用し、接続された機器と Modbus プロトコルを用いたシリアル通信を行います。マスタ/スレーブの機能を持ち、通信方式は ASCII および RTU に対応しています。なお、マスタ/スレーブ、ASCII/RTU は、定数定義にて変更可能です。

アクセス可能なレジスタは RAM に保持しており、ファンクションコードに応じた操作を行います。

5.1.1 Modbus 通信用 UART 通信設定

表 5-1 に UART 通信設定（Modbus 通信用）を示します。Modbus 通信用 UART の通信設定は r_modbus_serial.c および r_modbus_time.c を修正することで変更可能です。

表 5-1 UART 通信設定（Modbus 通信用）

ボーレート	データ長	パリティ	ストップビット	フロー制御
19200bps	ASCII(7bit), RTU(8bit)	Even	1bit	なし

5.1.2 Modbus ログ出力用 UART 通信設定

表 5-2 に UART 通信設定（ログ出力用）を示します。Modbus ログ出力用 UART の通信設定はコード生成から変更可能です。

表 5-2 UART 通信設定（ログ出力用）

ボーレート	データ長	パリティ	ストップビット	フロー制御
19200bps	8bit	Even	1bit	なし

5.1.3 対応ファンクションコード

表 5-3 に対応ファンクションコードを示します。

表 5-3 対応ファンクションコード

コード	ファンクション名	機能
01(0x01)	READ COILS	Discrete Output の ON/OFF 状態を読み出します。
02(0x02)	READ DISCRETE INPUTS	Discrete Input の ON/OFF 状態を読み出します。
03(0x03)	READ HOLDING REGISTERS	Holding Register の内容を読み出します。
04(0x04)	READ INPUT REGISTER	Input Register の内容を読み出します。
05(0x05)	WRITE SINGLE COIL	Discrete Output へ ON/OFF 状態を書き込みます。
06(0x06)	WRITE SINGLE REGISTER	保持レジスタへ内容を書き込みます。
15(0x0F)	WRITE MULTIPLE COILS	連続した複数の Discrete Output へ ON/OFF 状態を書き込みます。
16(0x10)	WRITE MULTIPLE REGISTERS	連続した複数の保持レジスタへ内容を書き込みます。
23(0x17)	READ/WRITE MULTIPLE REGISTERS	連続した複数の保持レジスタへ内容を書き込み、連続した複数の保持レジスタの内容を読み出します。

5.1.4 Modbus レジスタ割り当て

スレーブモードにおける Modbus レジスタ割り当てを記載します。サンプルコードでは各レジスタに機能を割り当てていません。以下に記載する各レジスタに対して、Modbus プロトコルの本サンプルコードでは、下記の値が Read/Write できるように実装されています。

5-4 に Modbus レジスタ割り当て（1/2）、表 5-5 に Modbus レジスタ割り当て（2/2）を示します。

5-4 Modbus レジスタ割り当て（1/2）

レジスタ名	アクセス単位	レジスタ数	アドレス範囲	初期値
Discrete Output	1bit	16	0x0000	1
			0x0001	1
			0x0002	1
			0x0003	1
			0x0004	1
			0x0005	1
			0x0006	1
			0x0007	1
			0x0008	1
			0x0009	1
			0x000A	1
			0x000B	1
			0x000C	1
			0x000D	1
			0x000E	1
			0x000F	1
Discrete Input	1bit	16	0x0000	1
			0x0001	1
			0x0002	0
			0x0003	0
			0x0004	1
			0x0005	0
			0x0006	1
			0x0007	0
			0x0008	0
			0x0009	0
			0x000A	1
			0x000B	1
			0x000C	0
			0x000D	1
			0x000E	0
			0x000F	1

※Discrete Input は g_discrete_input[] = {0xCA, 0x35};の場合、下記の動作になります。

- 4bit 読み出しで 0x0A
- 8bit 読み出しで 0xCA
- 12bit 読み出しで 0xCA05
- 16bit 読み出しで 0xCA35

表 5-5　Modbus レジスタ割り当て（2/2）

レジスタ名	アクセス単位	レジスタ数	アドレス	初期値
Input Register	2byte	8	0x0000	0x01FF
			0x0001	0x03FF
			0x0002	0x07FF
			0x0003	0x0FFF
			0x0004	0x1FFF
			0x0005	0x3FFF
			0x0006	0x7FFF
			0x0007	0xFFFF
Holding Register	2byte	8	0x0000	0x0000
			0x0001	0x0000
			0x0002	0x0000
			0x0003	0x0000
			0x0004	0x0000
			0x0005	0x0000
			0x0006	0x0000
			0x0007	0x0000

5.2　オプション・バイトの設定一覧

表 5-6　オプション・バイト設定にオプション・バイト設定を示します。

表 5-6　オプション・バイト設定

アドレス	設定値	内容
000C0H	11111110B	ウォッチドッグ・タイマ動作許可 （リセット解除後、カウント開始） HALT/STOP モード時、カウンタ動作停止
000C1H	11110111B	SPOR 検出電圧 立ち上がり時 TYP. 2.90 V (2.76 V ～ 3.02 V) 立ち下がり時 TYP. 2.84 V (2.70 V ～ 2.96 V) P125 端子機能 ：RESET 入力
000C2H	11111001B	高速オンチップ・オシレータ・クロック ：16 MHz
000C3H	10000101B	オンチップ・デバッグ許可

5.3 ファイル構成

図 5-1 にファイル構成を示します。

```
r01an6961-rl78g16-modbus /src/
 │   modbus.c
 ├──Modbus
 │       r_modbus.c
 │       r_modbus.h
 │       r_modbus_serial.c
 │       r_modbus_serial.h
 │       r_modbus_timer.c
 │       r_modbus_timer.h
 │
 └──smc_gen
     ├──Config_PORT
     │       Config_PORT.c
     │       Config_PORT.h
     │       Config_PORT_user.c
     │
     ├──Config_UART2
     │       Config_UART2.c
     │       Config_UART2.h
     │       Config_UART2_user.c
     │
     ├──Config_WDT
     │       Config_WDT.c
     │       Config_WDT.h
     │       Config_WDT_user.c
     │
     ├──general
     │       r_cg_macrodriver.h
     │       r_cg_port.h
     │       r_cg_sau.h
     │       r_cg_sau_common.c
     │       r_cg_sau_common.h
     │       r_cg_systeminit.c
     │       r_cg_userdefine.h
     │       r_cg_wdt.h
     │       r_smc_entry.h
     │
     ├──r_bsp
     │       ...
     │
     ├──r_config
     │       r_bsp_config.h
     │       r_bsp_config.inc
     │
     └──r_pincfg
             Pin.h
```

図 5-1 ファイル構成

5.4 スレーブモード（ASCII）

本サンプルコードのスレーブモード（ASCII）動作では、メイン処理にて周辺機能の初期化を行いModbus 受信フラグがセットされるまで待機します。Modbus 受信フラグは UART0 の割り込み処理からModbus ASCII フレームを受信した場合にセットされます。Modbus の受信フラグがセットされた後はスレーブアドレス、ファンクションコード、チェックサムの値をチェックし、全て適合した場合にコールバック関数をコールします。TAU00 にて Modbus フレームの文字間インターバルのタイムアウトをチェックしており、UART0 受信割り込みから Modbus パケットが完成するまでに TAU00 の割り込みが入った場合は、文字間インターバルエラーとみなし受信カウントおよび受信バッファをクリアします。

ASCII における文字間インターバルエラーのタイムアウト時間は、規格上では規定されておらず、「1 秒を超える間隔はエラーが発生したことを意味するが、一部のアプリケーションではもっと長いタイムアウト時間が必要になる場合がある」となっています。本サンプルコードでは 1 秒としています。

また、Modbus 通信フレームのログ、エラー発生時のログを UART2 へ出力します。

5.4.1 メイン処理

図 5-2 にメイン処理（スレーブモード（ASCII））のフローチャートを示します。

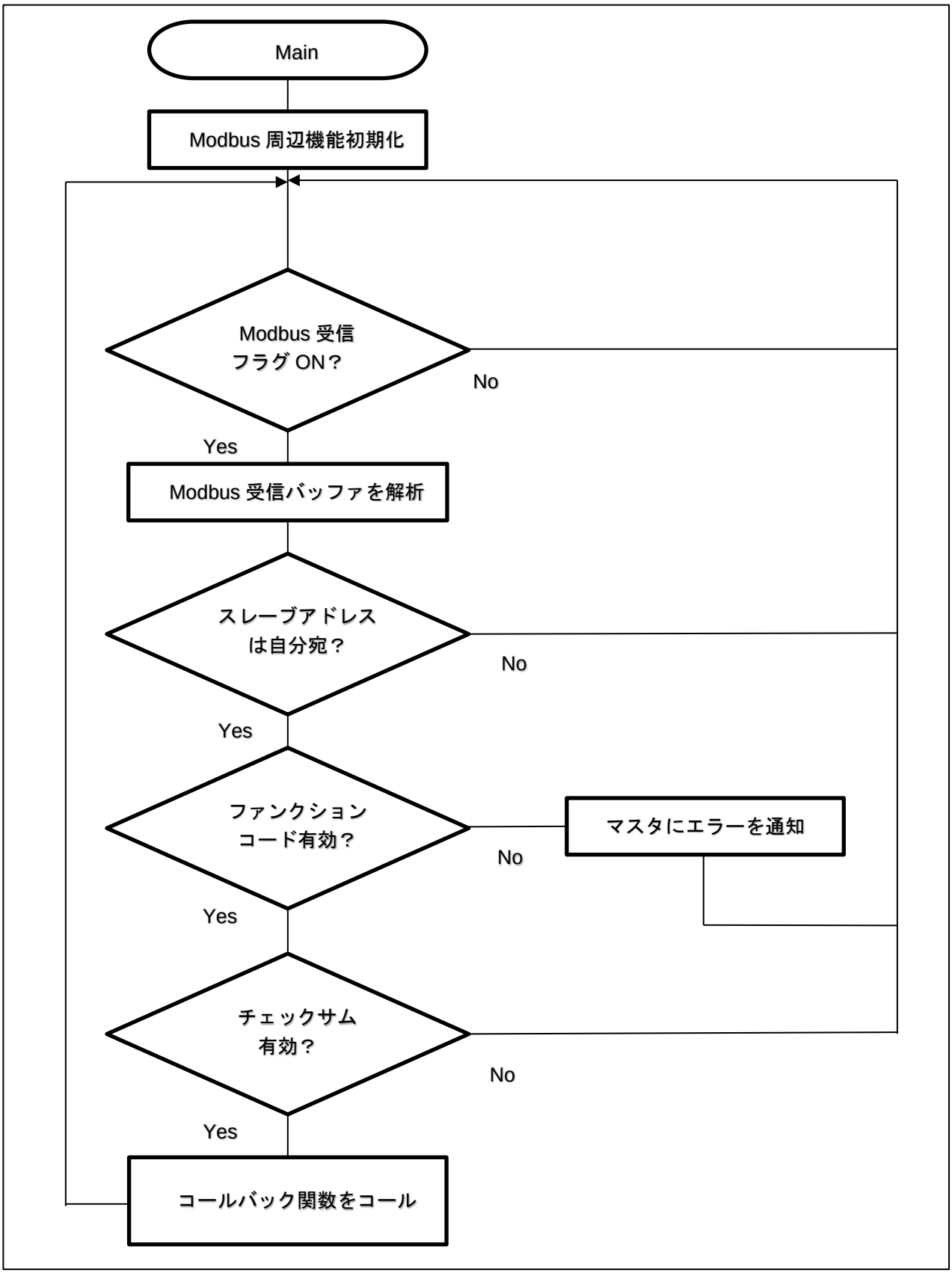


図 5-2 メイン処理（スレーブモード（ASCII））

5.4.2 シリアル受信割り込み処理

図 5-3 にシリアル受信割り込み処理（スレーブモード（ASCII））のフローチャートを示します。

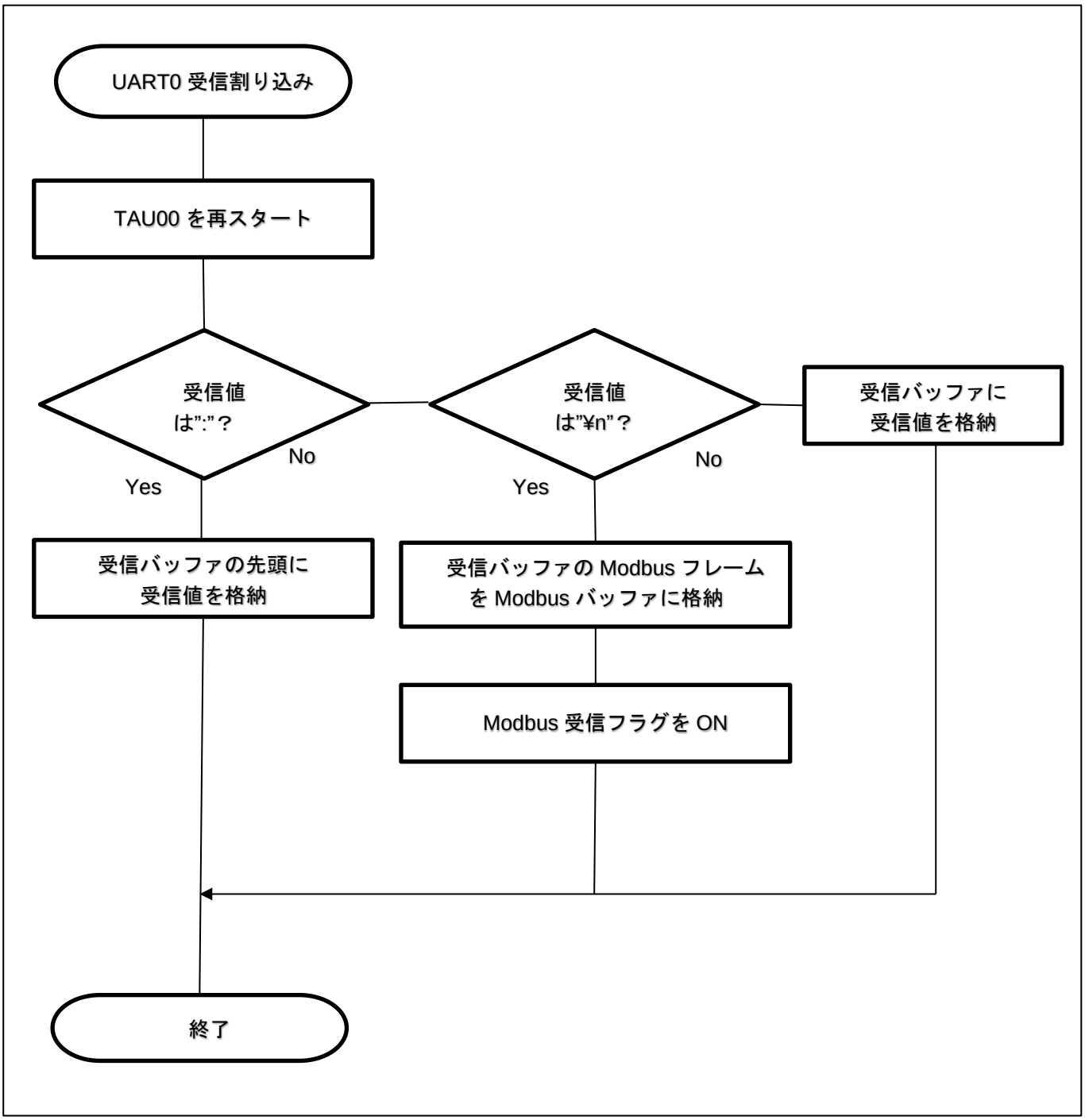


図 5-3 シリアル受信割り込み処理（スレーブモード（ASCII））

5.4.3 文字間インターバルエラー割り込み

図 5-4 に文字間インターバルエラー割り込み処理（スレーブモード（ASCII））のフローチャートを示します。

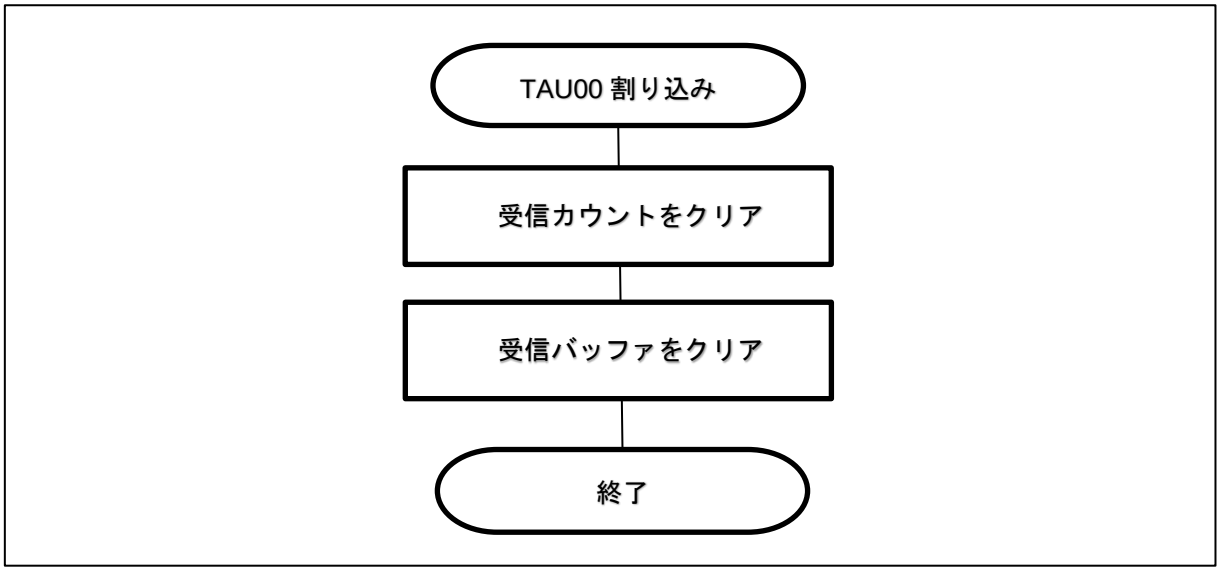


図 5-4 文字間インターバルエラー割り込み処理（スレーブモード（ASCII））

5.5 スレーブモード（RTU）

本サンプルコードのスレーブモード（RTU）動作では、メイン処理にて周辺機能の初期化を行い Modbus 受信フラグがセットされるまで待機します。TAU00 にて Modbus フレームの文字間インターバルとフレーム間インターバルのタイムアウトをチェックしており、UART0 割り込みから Modbus パケットが完成するまでに TAU00 の割り込みが入った場合は、文字間インターバルエラーフラグをセットし、TAU00 をフレーム間インターバルの時間に変更してスタートします。文字間インターバルエラーフラグがセットされている状態で再び UART0 受信割り込み処理が発生した場合は文字間インターバルエラーとみなし、受信カウント数および受信バッファをクリアします。このフラグがセットされている状態で UART0 受信割り込みが発生せずに TAU00 の割り込み処理が入った場合は正常終了とみなし、文字間インターバルエラーフラグをクリアして Modbus 受信フラグをセットします。Modbus の受信フラグがセットされた後はスレーブアドレス、ファンクションコード、チェックサムの値をチェックし、全て適合した場合にコールバック関数をコールします。

RTU における文字間インターバルエラーのタイムアウト時間は 1.5 文字分、フレーム間インターバルは 3.5 文字分となっており、本サンプルコードの通信設定により、それぞれの時間は以下のように設定しています。

各インターバル ＝ 文字数 * 1 文字辺りのビット数 * 1 ビット辺りの通信時間 * HOCO の精度分（1%）

文字間インターバル：1.5 * 11 * 1/19200 * 1.01 ≒ 868[us]

フレーム間インターバル：3.5 * 11 * 1/19200 * 1.01 ≒ 2026[us]

本サンプルコードでは、1 つのタイマ（TAU00）で文字間インターバル、フレーム間インターバルの 2 つの判定を行っているため、文字間インターバル・タイマの割り込みが発生した後、TAU00 を 1158[us]（2026 – 868）に変更して動作させ、フレーム間インターバルの判定を行っています。

また、Modbus 通信フレームのログ、エラー発生時のログを UART2 へ出力します。

5.5.1 メイン処理

図 5-5 にメイン処理（スレーブモード（RTU））を示します。

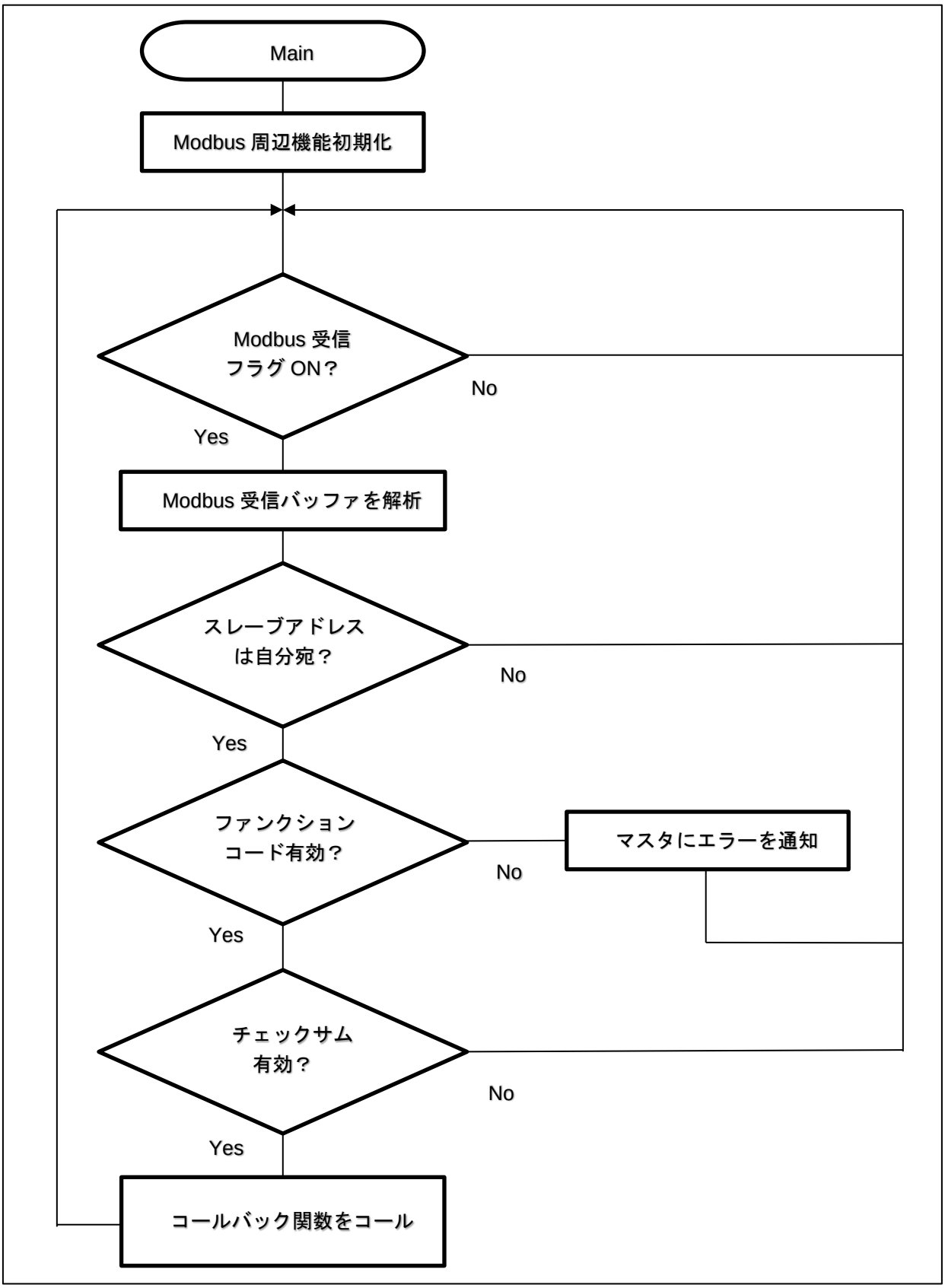


図 5-5 メイン処理（スレーブモード（RTU））

5.5.2 シリアル受信割り込み処理

図 5-6 にシリアル受信割り込み処理（スレーブモード（RTU））のフローチャートを示します。

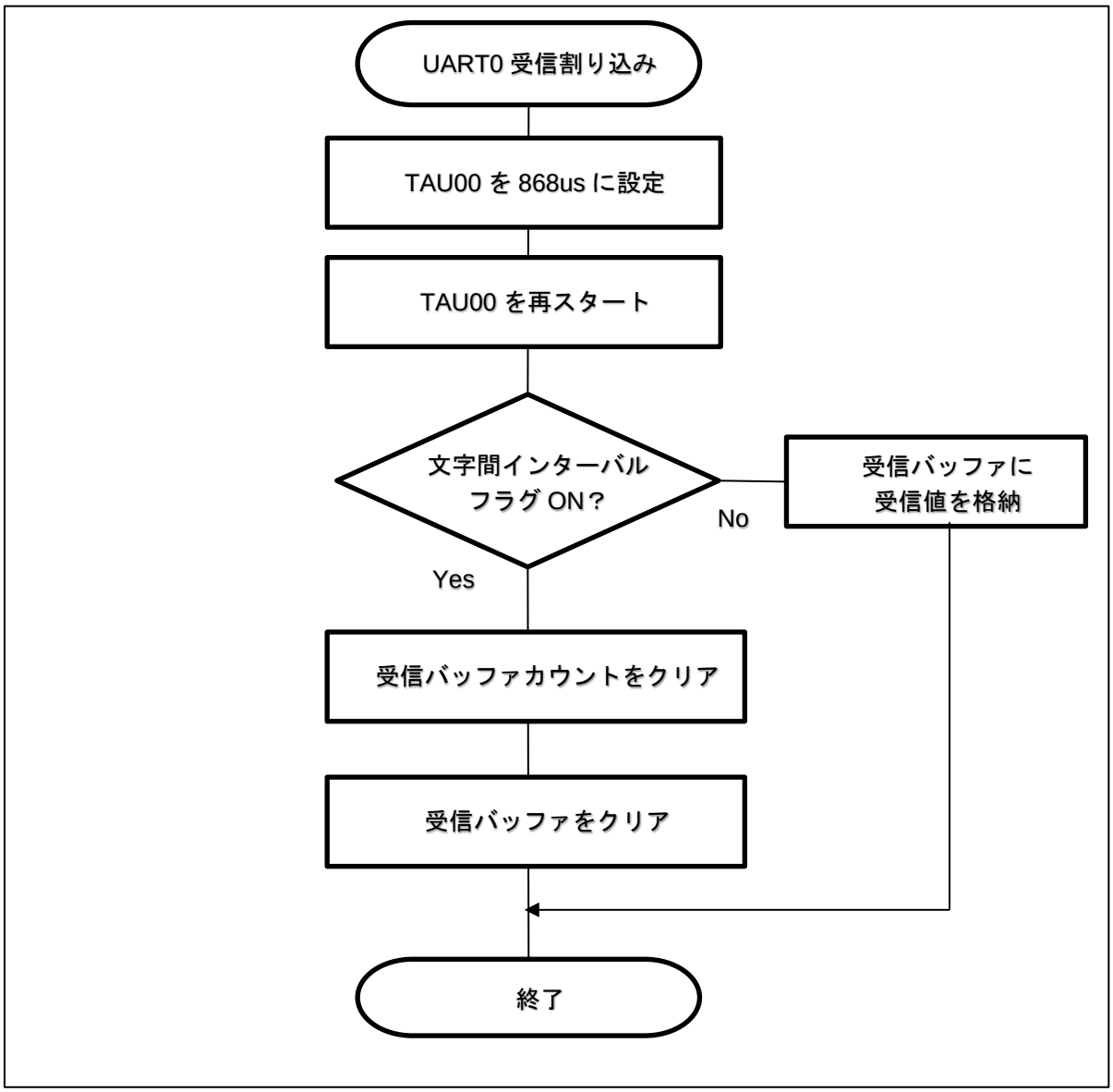


図 5-6 シリアル受信割り込み処理（スレーブモード（RTU））

5.5.3 文字間インターバル割り込み処理

図 5-7 に文字間インターバル割り込み処理（スレーブモード（RTU））のフローチャートを示します。

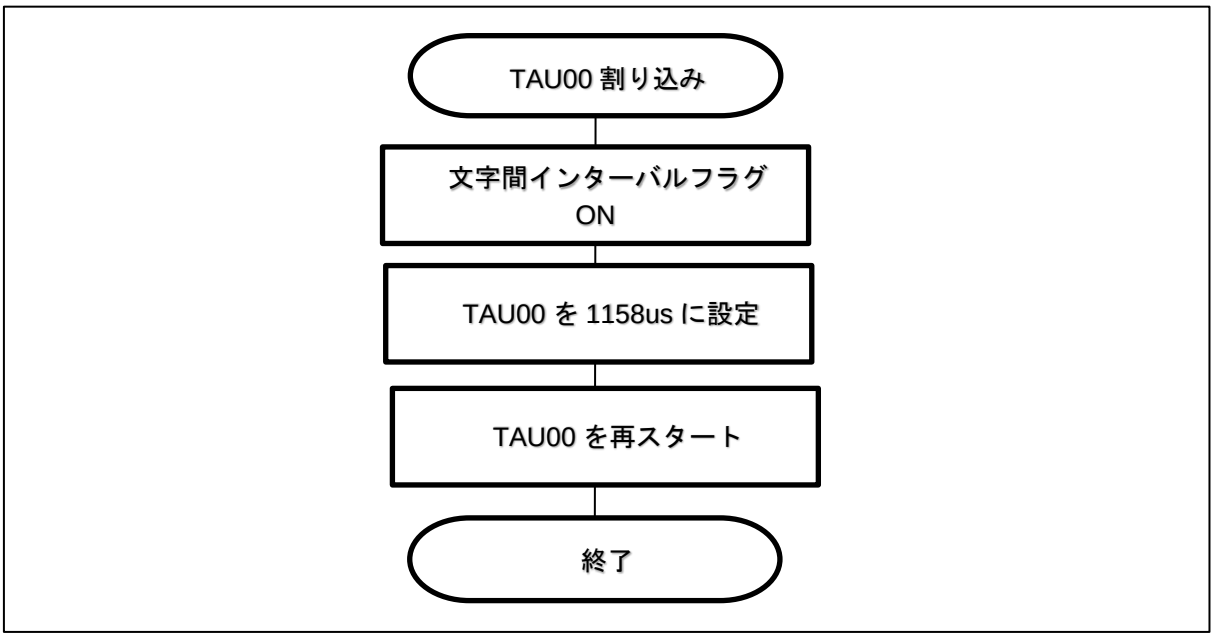


図 5-7 文字間インターバル割り込み処理（スレーブモード（RTU））

5.5.4 Modbus 受信完了割り込み処理

図 5-8 に Modbus 受信完了割り込み処理（スレーブモード（RTU））のフローチャートを示します。

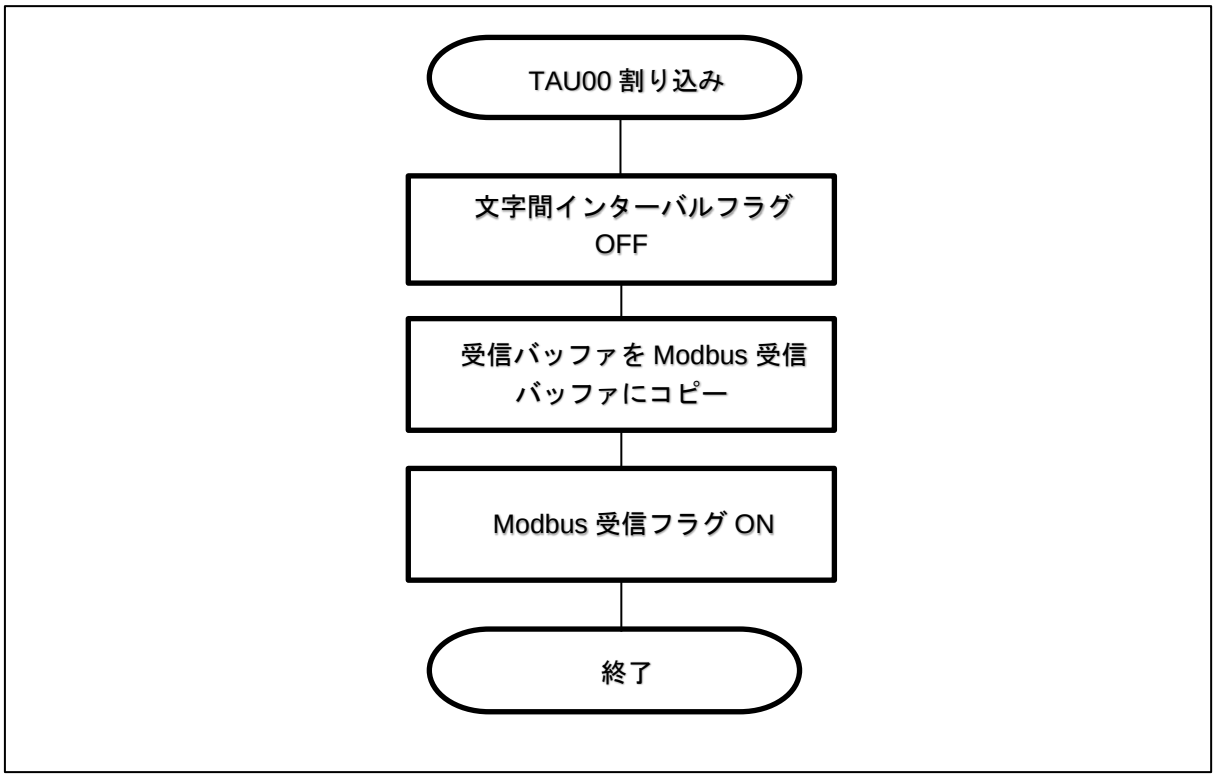


図 5-8 Modbus 受信完了割り込み処理（スレーブモード（RTU））

5.6 マスタモード（ASCII）

本サンプルコードのマスタモード（ASCII）では、1 秒おきに SLAVE ID=0x01 へ Read Coils を送信します。スレーブから応答を受信した場合は、コールバック関数をコールします。TAU00 にて Modbus フレームの文字間インターバルのタイムアウトをチェックしており、受信割り込みから Modbus パケットが完成するまでに TAU00 の割り込みが発生した場合は文字間インターバルエラーとみなし、受信カウントおよび受信バッファをクリアします。

ASCII における文字間インターバルエラーのタイムアウト時間は、規格上では規定されておらず、「1 秒を超える間隔はエラーが発生したことを意味するが、一部のアプリケーションではもっと長いタイムアウト時間が必要になる場合がある」となっています。本サンプルコードでは 1 秒としています。Read Coils の送信間隔（1 秒）は、TAU01 を使用します。

また、Modbus 通信フレームのログ、エラー発生時のログを UART2 へ出力します。

5.6.1 メイン処理

図 5-9 にメイン処理（マスタモード（ASCII））のフローチャートを示します。

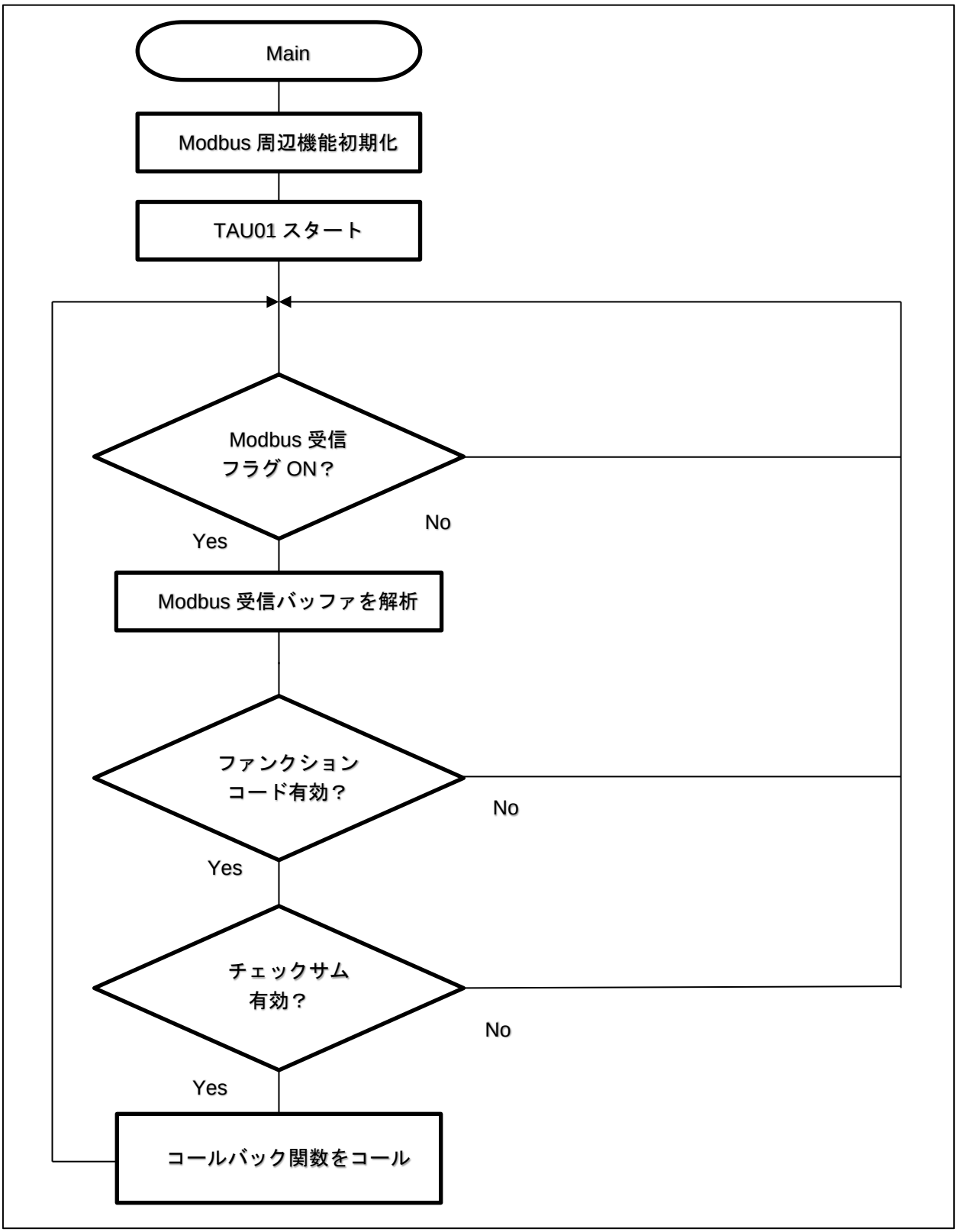


図 5-9 メイン処理（マスタモード（ASCII））

5.6.2 Read Coil 送信割り込み処理

図 5-10 に Read Coil 送信割り込み処理（マスタモード（ASCII））のフローチャートを示します。

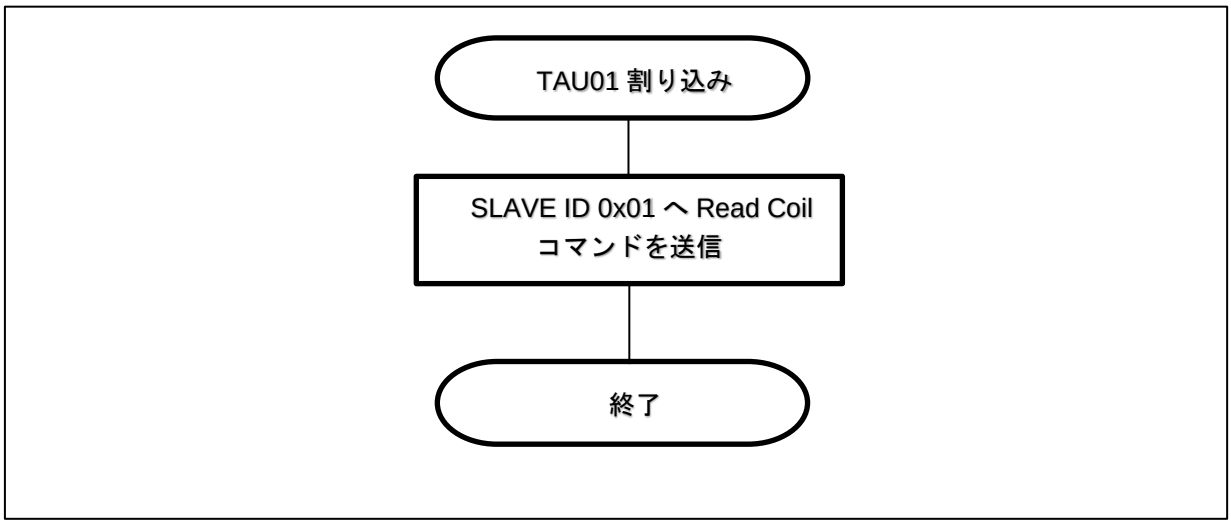


図 5-10 Read Coil 送信割り込み処理（マスタモード（ASCII））

5.6.3 シリアル受信割り込み処理

図 5-11 にシリアル受信割り込み処理（マスタモード（ASCII））のフローチャートを示します。

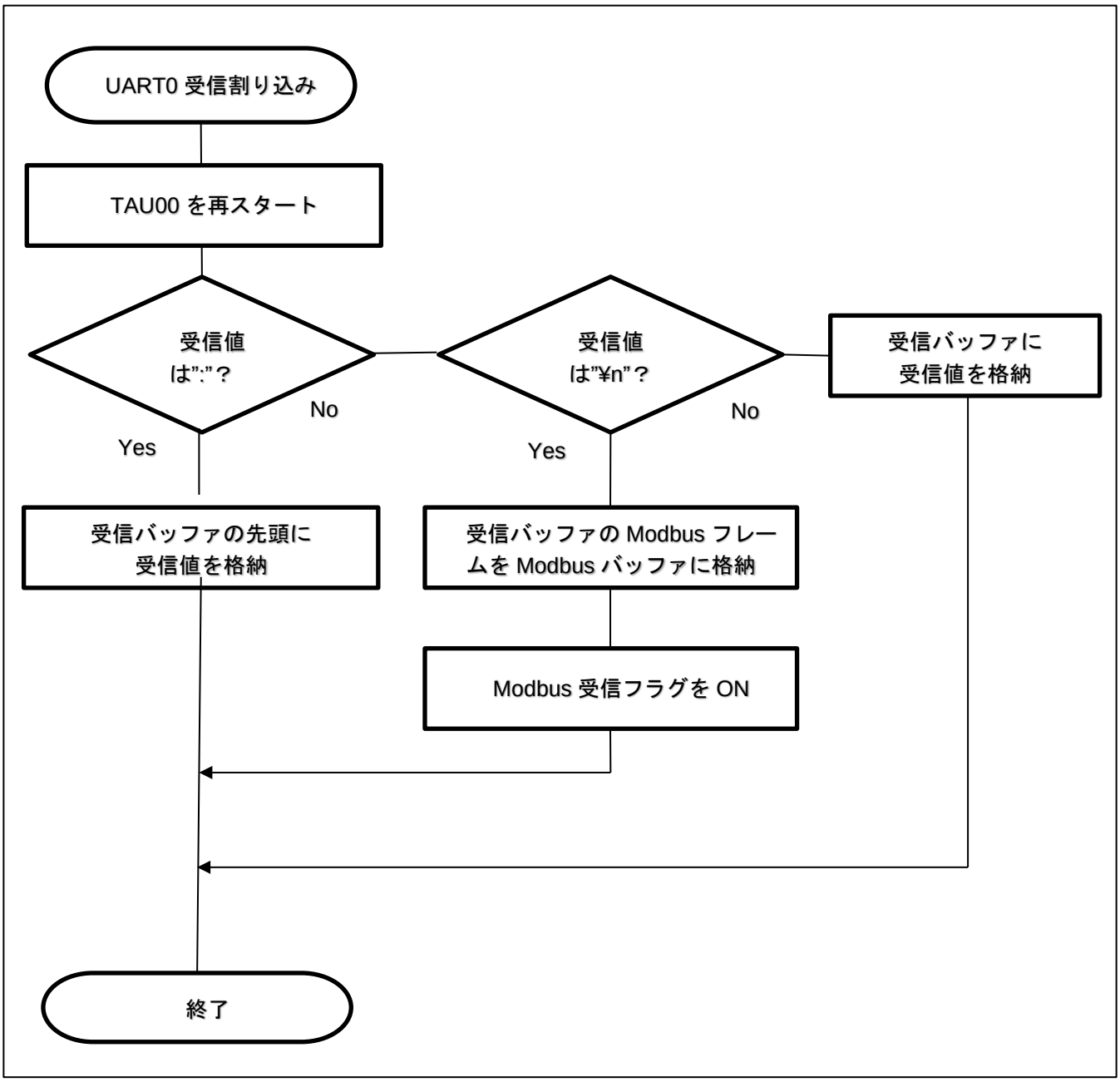


図 5-11 シリアル受信割り込み処理（マスタモード（ASCII））

5.6.4 文字間インターバルエラー割り込み処理

図 5-12 に文字間インターバルエラー処理（マスタモード（ASCII））のフローチャートを示します。

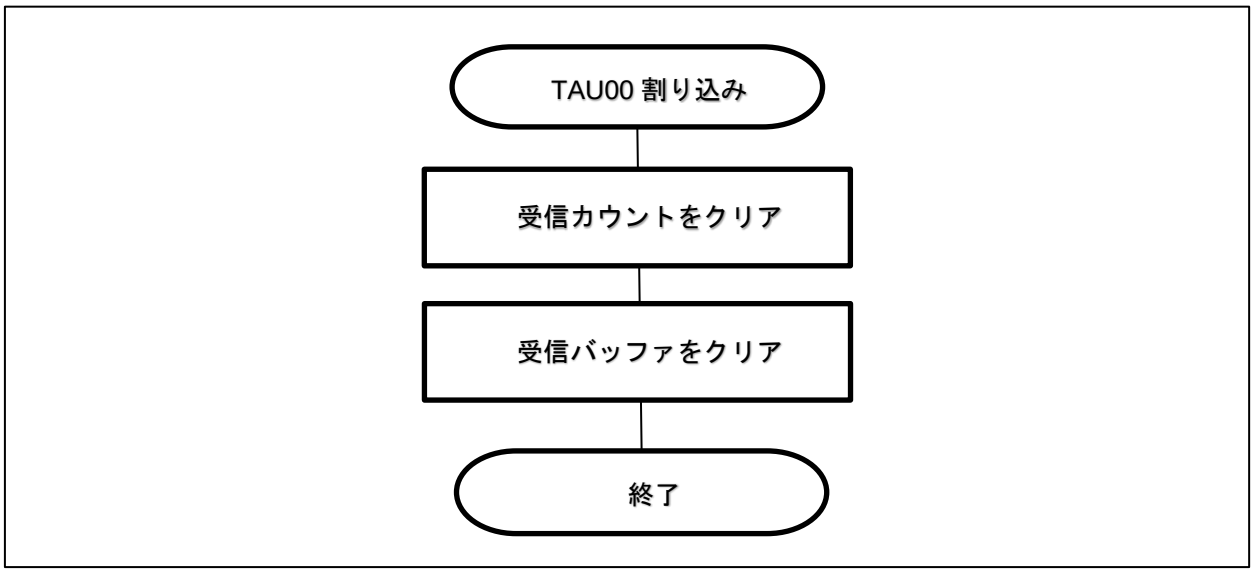


図 5-12 文字間インターバルエラー処理（マスタモード（ASCII））

5.7 マスタモード（RTU）

本サンプルコードのマスタモード（RTU）では、1 秒おきに SLAVE ID=0x01 へ Read Coils を送信します。スレーブからコマンドを受信した場合はコールバック関数をコールします。TAU00 にて Modbus フレームの文字間インターバルとフレーム間インターバルのタイムアウトをチェックしており、UART0 受信割り込みから Modbus パケットが完成するまでに TAU00 の割り込みが入った場合は、文字間インターバルエラーフラグをセットし、TAU00 をフレーム間インターバルの時間に変更してスタートします。文字間インターバルエラーフラグがセットされている状態で再び UART0 受信割り込みが発生した場合は文字間インターバルエラーとみなし、受信カウント数および受信バッファをクリアします。このフラグがセットされている状態で UART0 受信割り込み処理が発生せずに TAU00 の割り込みが発生した場合は正常終了とみなし、文字間インターバルエラーフラグをクリアして Modbus 受信フラグをセットします。Modbus の受信フラグがセットされた後はスレーブアドレス、ファンクションコード、チェックサムの値をチェックし、全て適合した場合にコールバック関数をコールします。

RTU における文字間インターバルエラーのタイムアウト時間は 1.5 文字分、フレーム間インターバルは 3.5 文字分となっており、本サンプルコードの通信設定により、それぞれの時間は以下のように設定しています。

各インターバル ＝ 文字数 * 1 文字辺りのビット数 * 1 ビット辺りの通信時間 * HOCO の精度分（1%）

文字間インターバル：1.5 * 11 * 1/19200 * 1.01 ≒ 868[us]

フレーム間インターバル：3.5 * 11 * 1/19200 * 1.01 ≒ 2026[us]

本サンプルコードでは、1 つのタイマ（TAU00）で文字間インターバル、フレーム間インターバルの 2 つの判定を行っているため、文字間インターバル・タイマの割り込みが発生した後、TAU00 を 1158[us]（2026 – 868）に変更して動作させ、フレーム間インターバルの判定を行っています。Read Coils の送信間隔（1 秒）は、TAU01 を使用します。

また、Modbus 通信フレームのログ、エラー発生時のログを UART2 へ出力します。

5.7.1 メイン処理

図 5-13 にメイン処理（マスタモード（RTU））のフローチャートを示します。

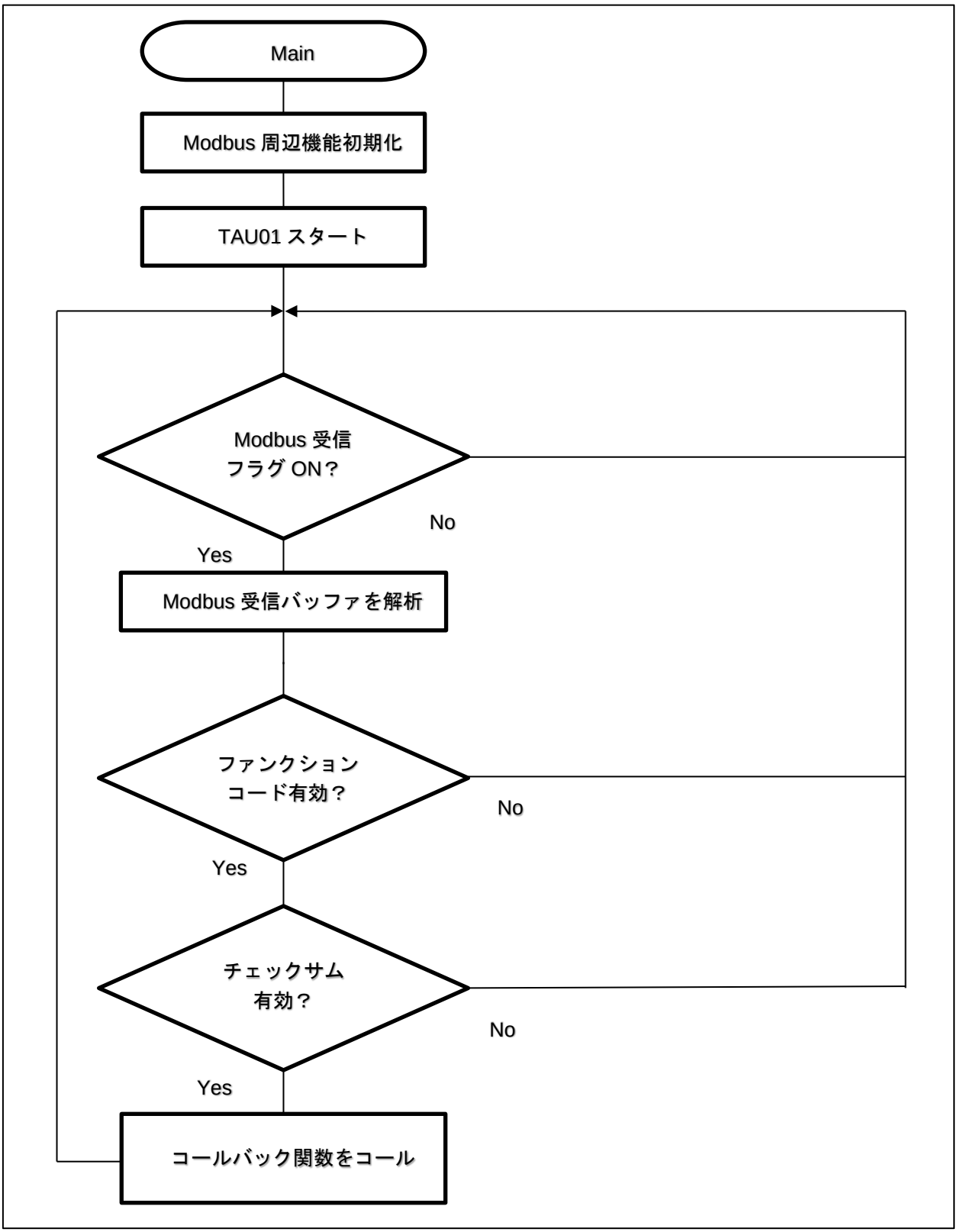


図 5-13 メイン処理（マスタモード（RTU））

5.7.2 Read Coil 送信割り込み処理

図 5-14 に Read Coil 送信割り込み処理（マスタモード（RTU））のフローチャートを示します。

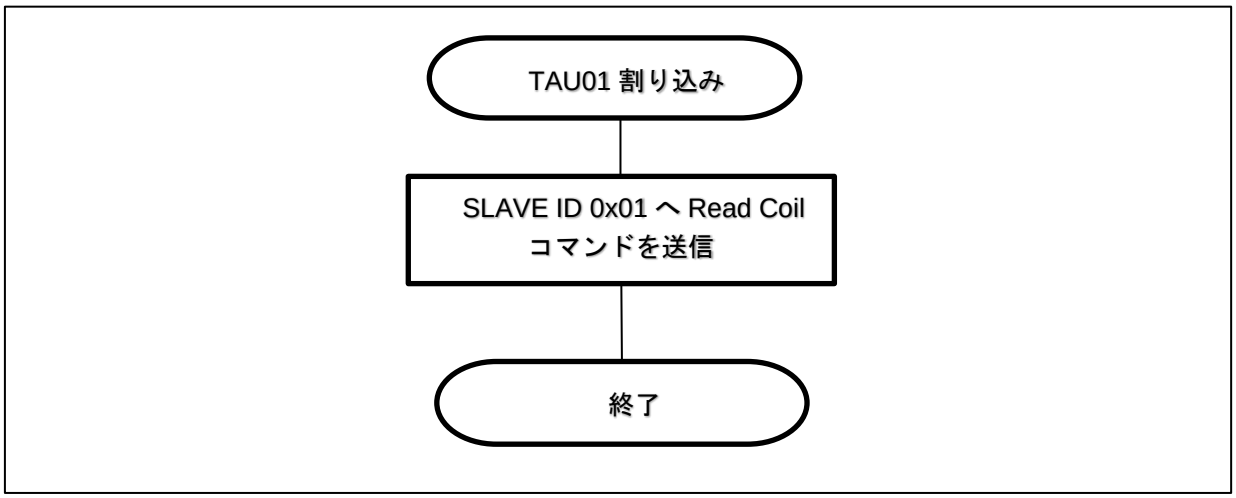


図 5-14 Read Coil 送信割り込み処理（マスタモード（RTU））

5.7.3 シリアル受信割り込み処理

図 5-15 にシリアル受信割り込み処理（マスタモード（RTU））のフローチャートを示します。

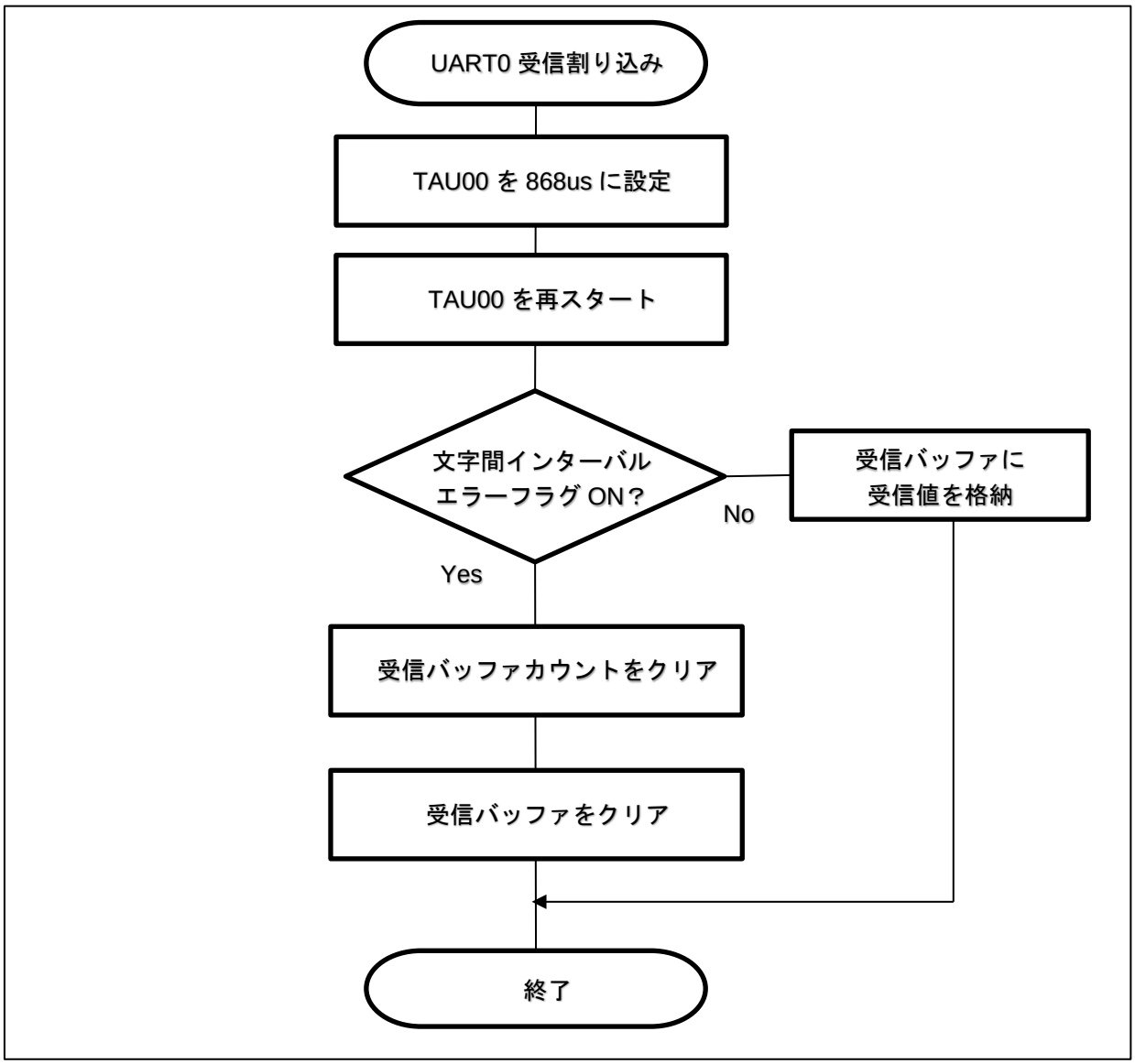


図 5-15 シリアル受信割り込み処理（マスタモード（RTU））

5.7.4 文字間インターバル割り込み処理

図 5-16 に文字間インターバル割り込み処理（マスタモード（RTU））のフローチャートを示します。

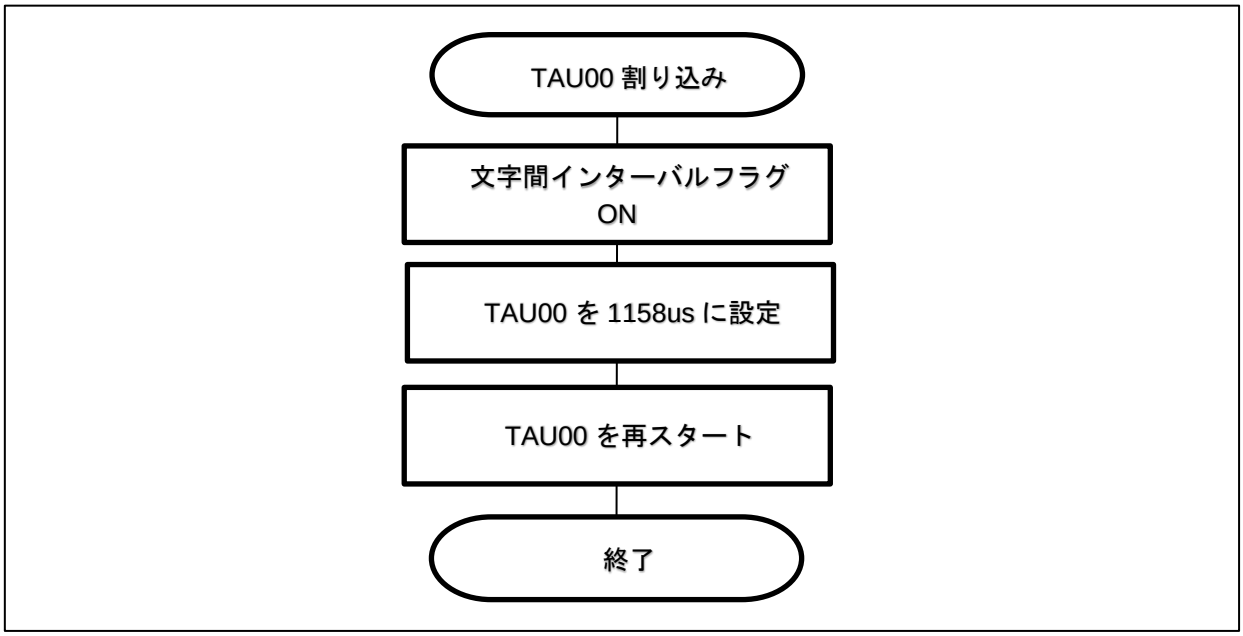


図 5-16 文字間インターバル割り込み処理（マスタモード（RTU））

5.7.5 Modbus 受信完了割り込み処理

図 5-17 に Modbus 受信完了割り込み処理（マスタモード（RTU））のフローチャートを示します。

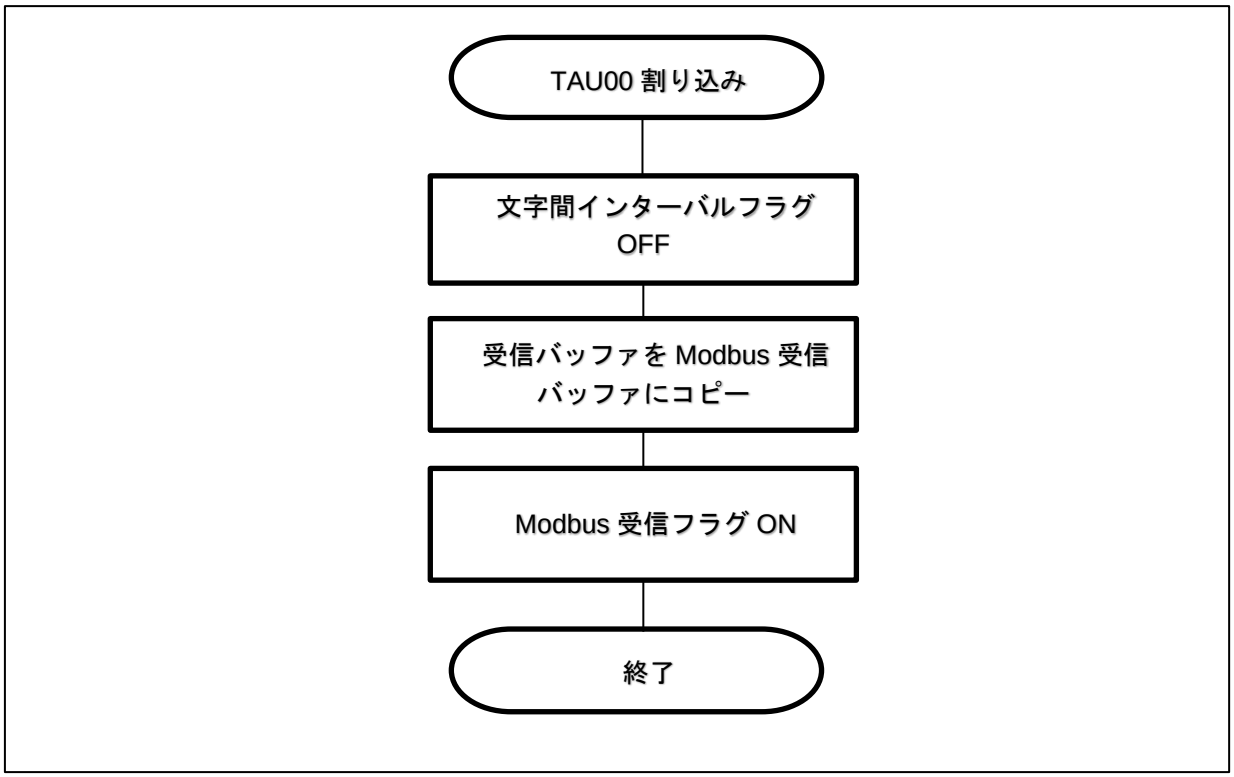


図 5-17 Modbus 受信完了割り込み処理（マスタモード（RTU））

5.8 定数一覧

5.8.1 Modbus 動作設定定数

表 5-7 に Modbus 動作設定定数一覧を示します。

表 5-7 Modbus 動作設定定数一覧

定数名	値	用途
MODBUS_MODE	0x01～0x04	サンプルコードの動作モードを指定します。 0x01 : マスタモード（RTU） 0x02 : スレーブモード（RTU） 0x03 : マスタモード（ASCII） 0x04 : スレーブモード（ASCII）
MODBUS_RTU_MASTER_MODE	0x01	マスタモード（RTU）を指定する際に使用します。
MODBUS_RTU_SLAVE_MODE	0x02	スレーブモード（RTU）を指定する際に使用します。
MODBUS_ASCII_MASTER_MODE	0x03	マスタモード（ASCII）を指定する際に使用します。
MODBUS_ASCII_SLAVE_MODE	0x04	スレーブモード（ASCII）を指定する際に使用します。
MODBUS_SEND_BUFFER_SIZE	1～256	Modbus の送信バッファの最大サイズを指定します。
MODBUS_RECV_BUFFER_SIZE	1～253	Modbus の受信バッファの最大サイズを指定します。

5.8.2 Modbus ステータス定数

表 5-8 に Modbus ステータス定数一覧を示します。

表 5-8 Modbus ステータス定数一覧

定数名	値	用途
MODBUS_STATUS_NONE	0x00	Modbus 動作開始前状態
MODBUS_STATUS_WAIT_RECEIVE	0x01	Modbus フレーム受信待ち状態
MODBUS_STATUS_RECEIVED	0x02	Modbus フレーム受信完了状態
MODBUS_STATUS_WAIT_FRAME_INTERVAL	0x03	フレーム間インターバル待ち状態 （RTU モードの場合のみ使用）
MODBUS_STATUS_ERROR_CHARACTER_INTERVAL	0x04	Modbus フレーム受信中に文字間インターバルエラーが発生した状態

5.8.3 Modbus 受信結果定数

表 5-9 に Modbus 受信結果定数一覧を示します。

表 5-9 Modbus 受信結果定数一覧

定数名	値	用途
MODBUS_RECEIVE_NONE	0x00	Modbus フレーム未受信
MODBUS_ERROR_NONE	0x01	Modbus フレーム受信完了後エラーがない
MODBUS_ERROR_NULL_POINTER	0x02	Modbus フレームの解析結果格納先構造体アドレスに NULL が指定された
MODBUS_ERROR_NOT_MYSELF	0x03	Modbus フレームを解析した結果、自身宛てではなかった
MODBUS_ERROR_ILLEGAL_FUNCTION	0x04	Modbus フレームを解析した結果、結果ファンクションコードが異常であった
MODBUS_ERROR_MISMATCH_CHECKSUM	0x05	Modbus フレームを解析した結果。結果チェックサムが異常であった
MODBUS_ERROR_CHARACTER_INTERVAL	0x06	Modbus フレーム受信中に文字間インターバルエラーが発生していた
MODBUS_ERROR_ILLEGAL_FRAME_LENGTH	0x07	受信した Modbus フレームが想定されるフレーム長よりも短い
MODBUS_ERROR_RECEIVE_ERROR_RESPONSE	0x08	マスタモード時、スレーブからエラー応答を受信した

5.9 変数一覧

表 5-10 にグローバル変数一覧を示します。

表 5-10 グローバル変数一覧

型	変数名	用途
unsigned char[]	g_modbus_rx_buffer	Modbus フレーム受信バッファ
unsigned char	g_modbus_status	Modbus 通信の状態
unsigned char	g_modbus_frame_size	Modbus 受信フレームサイズ
unsigned short[]	g_holding_register	保持レジスタ
unsigned short[]	g_input_register	入力レジスタ
unsigned char[]	g_discrete_output	DO
unsigned char[]	g_discrete_input	DI
unsigned char[]	g_rx_buffer	UART 受信バッファ

5.10 構造体一覧

表 5-11 に構造体一覧を示します。

表 5-11　構造体一覧

型名	フィールド	用途
st_modbus_receive_frame_t	uint8_t slave_address uint8_t function_code uint16_t read_address uint16_t read_data uint16_t write_address uint16_t write_data uint8_t array_data_len uint16_t array_data[]	Modbus フレームを解析した結果を格納します

5.11 関数一覧

5.11.1 API 関数

表 5-12 にサンプルコードの API 関数一覧を示します。

表 5-12　サンプルコードの API 関数一覧

関数名	概要
R_MODBUS_Main	Modbus 通信のメイン処理
R_MODBUS_Init	Modbus 通信で使用する周辺機能の初期化
R_MODBUS_Close	Modbus 通信で使用する周辺機能の終了
R_MODBUS_Send	Modbus フレーム送信
R_MODBUS_Receive	Modbus フレーム受信
R_MODBUS_Parse	Modbus フレーム解析
R_MODBUS_Send_Error	Modbus エラー送信

5.11.2 コールバック関数

表 5-13 にサンプルコードのコールバック関数一覧を示します。

表 5-13　サンプルコードのコールバック関数一覧

関数名	概要
r_modbus_callback_slave_receiveend	スレーブモード時 Modbus フレームの受信完了時に呼び出されます
r_modbus_callback_master_receiveend	マスタモード時 Modbus フレームの受信完了時に呼び出されます
r_modbus_callback_error	Modbus 通信／フレーム解析処理でエラーが発生した場合に呼び出されます

5.12 関数仕様

サンプルコードの関数仕様を示します。

[関数名] R_MODBUS_Main

概要	Modbus 通信のメイン処理
ヘッダ	r_modbus.h
宣言	void R_MODBUS_Main (void)
説明	Modbus 通信の受信を検知し、コールバック関数をコールします。マスタモード時は受信要求を検知しスレーブへ Modbus フレームを送信します。
引数	なし
リターン値	なし
備考	なし

[関数名] R_MODBUS_Init

概要	Modbus 通信で使用する周辺機能の初期化
ヘッダ	なし
宣言	void R_MODBUS_Init (void)
説明	Modbus 通信で使用する周辺機能の初期化を行います。
引数	なし
リターン値	なし
備考	Modbus 通信を開始する前に必ず実行してください。

[関数名] R_MODBUS_Close

概要	Modbus 通信で使用する周辺機能の終了
ヘッダ	なし
宣言	void R_MODBUS_Close (void)
説明	Modbus 通信で使用する周辺機能を終了します。
引数	なし
リターン値	なし
備考	Modbus 通信を終了する際に必ず実行してください。

[関数名] R_MODBUS_Send

概要	Modbus フレーム送信
ヘッダ	なし
宣言	void R_MODBUS_Send(uint8_t slave_address, uint8_t function_code, uint8_t *send_data, uint8_t send_data_len)
説明	send_data を send_data_len の数分 Modbus フレームとして送信します。
引数	スレーブアドレス、ファンクションコード、送信データアドレス、送信データ長
リターン値	なし
備考	ASCII モードの場合は関数内でデータ変換を行います。第一引数には RTU データを渡してください。チェックサムは内部で自動計算を行います。

[関数名] R_MODBUS_Receive

概要	Modbus フレーム受信
ヘッダ	なし
宣言	uint8_t R_MODBUS_Recieve(st_modbus_receive_frame_t * modbus_receive_frame)
説明	UART0 の状態を確認し Modbus フレームを受信していた場合、引数 modbus_receive_frame に Modbus フレームをパースして格納します。正常受信した場合、受信中にエラーが発生していた場合はコールバック関数をコールします。本関数はノンブロッキングです。
引数	Modbus フレーム構造体アドレス
リターン値	Modbus フレームの受信結果
備考	リターン値に指定される値は 5.8.3 Modbus 受信結果定数を参照してください。

[関数名] R_MODBUS_Parse

概要	Modbus フレーム解析
ヘッダ	なし
宣言	uint8_t R_MODBUS_Parse(uint8_t * modbus_rx_buffer, uint8_t modbus_frame_size, st_modbus_receive_frame_t * modbus_receive_frame)
説明	modbus_rx_buffer を解析し引数 modbus_receive_frame に格納します。関数内でスレーブアドレスが自分宛てか、チェックサムは正しいか、対応するファンクションコードは存在するか判定し Modbus フレームの解析結果としてリターン値に返します。
引数	Modbus フレームアドレス、Modbus フレームサイズ、 Modbus フレーム構造体アドレス
リターン値	Modbus フレームの解析結果
備考	リターン値に指定される値は 5.8.3 Modbus 受信結果定数を参照してください。

[関数名] R_MODBUS_Send_Error

概要	エラー応答を送信します
ヘッダ	なし
宣言	void R_MODBUS_Send_Error (uint8_t slave_address, uint8_t function, uint8_t error_code)
説明	function で指定されたファンクションコードのエラーとして error_code を送信します。slave_address が 0x00（ブロードキャスト）の場合は送信を行いません。
引数	受信フレームの宛先スレーブアドレス、 エラー発生ファンクションコード、エラーコード
リターン値	なし
備考	なし

[関数名] r_modbus_callback_slave_receiveend

概要	スレーブモード時 Modbus フレームの受信完了時に呼び出されます
ヘッダ	なし
宣言	void r_modbus_callback_slave_receiveend (st_modbus_receive_frame_t modbus_receive_frame)
説明	サンプルコードではファンクションコードに応じた処理を実装しています。
引数	Modbus 受信フレーム
リターン値	なし
備考	なし

[関数名] r_modbus_callback_slave_receiveend

概要	マスタモード時 Modbus フレームの受信完了時に呼び出されます
ヘッダ	なし
宣言	void r_modbus_callback_master_receiveend (st_modbus_receive_frame_t modbus_receive_frame)
説明	サンプルコードではファンクションコードに応じた処理を実装しています。
引数	Modbus 受信フレーム
リターン値	なし
備考	なし

[関数名] r_modbus_callback_error

概要	Modbus 通信でエラーが発生した場合呼び出されます
ヘッダ	なし
宣言	void r_modbus_callback_error (uint8_t error_type, st_modbus_receive_frame_t modbus_receive_frame)
説明	サンプルコードではエラー種別に応じた処理を実装しています。
引数	エラー種別、Modbus 受信フレーム
リターン値	なし
備考	なし

5.13 ログ仕様

表 5-14 にサンプルコードのログ仕様について記します。

表 5-14　ログ仕様

ログ出力タイミング	ログ内容（出力される文字列）
Modbus フレームの受信	[RX]+受信した Modbus フレーム
Modbus フレームの送信	[TX]+送信した Modbus フレーム
Modbus フレーム解析後の異常発生	Modbus 受信結果定数名 5.8.3 Modbus 受信結果定数を参照してください。
受信コールバック関数コール	CALL_BACK_FUNCTION+ファンクション名

5.14 ROM/RAM サイズ

表 5-15 に本サンプルコードで使用する ROM/RAM サイズを記載します。下記サイズは Modbus 送信バッファサイズ 64Byte、Modbus 受信バッファサイズ 64Byte 構成での ROM/RAM サイズとなります。(最適化レベル=デバッグ優先でのサイズです。最適化レベルで変動します。)

表 5-15 ROM/RAM サイズ

動作モード	ROM サイズ	RAM サイズ
スレーブモード(ASCII)	12,926Byte	895Byte
スレーブモード(RTU)	10,793Byte	895Byte
マスタモード(ASCII)	11,230Byte	909Byte
マスタモード(RTU)	10,145Byte	909Byte

6. 動作確認手順

6.1 RL78 - PC(GUI)環境

6.1.1 接続例

4.1.1 RL78 – PC(GUI)環境を参照してください。

6.1.2 ファームウェア定数設定

modbus.h の定数 MODBUS_MODE を MODBUS_RTU_SLAVE_MODE に設定し、プロジェクトをビルド後デバッグツールへダウンロードします。

ファイル構成については 5.3 ファイル構成を参照してください。

※例では RL78 はスレーブモード(RTU)での通信を行いますが MODBUS_MODE の設定値を変更することで通信方式の変更が可能です。設定値については 5.8.1 Modbus 動作設定定数を参照してください。

6.1.3 GUI パラメータ設定

図 6-1 に GUI パラメータ設定を記します。

尚、Serial setting の COM ポートはご使用の環境に合わせて設定してください。

※例では GUI はマスタモード(RTU)での通信を行いますが Connection と Serial setting の設定値を変更することで通信方式の変更が可能です。

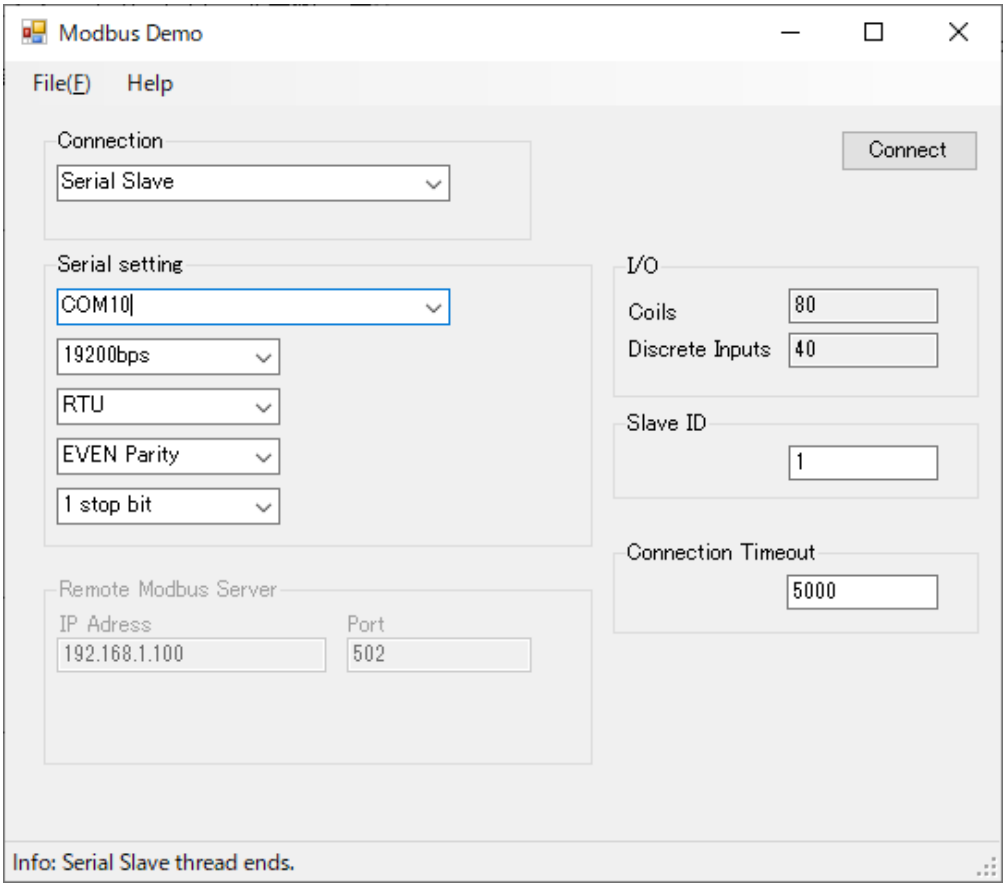


図 6-1 GUI パラメータ設定

6.1.4 実行手順

RL78 のプログラムを実行後、GUI の Connect ボタンを押下することでデモ動作を開始します。

動作例：

スレーブ側ログ出力

```
[RX]01020000000879CC
CALL_BACK_FUNCTION_READ_DESCRETE_INPUTS
[TX]010201CA21DF
[RX]010F0000000801013F55
CALL_BACK_FUNCTION_WRITE_MULTIPLE_COILS
[TX]010F00000008540D
[RX]01020000000879CC010F0000000801027F54
CALL_BACK_FUNCTION_READ_DESCRETE_INPUTS
[TX]010201CA21DF
```

6.2 RL78 - RL78 環境

6.2.1 接続例

4.1.2 RL78 – RL78 環境を参照してください。

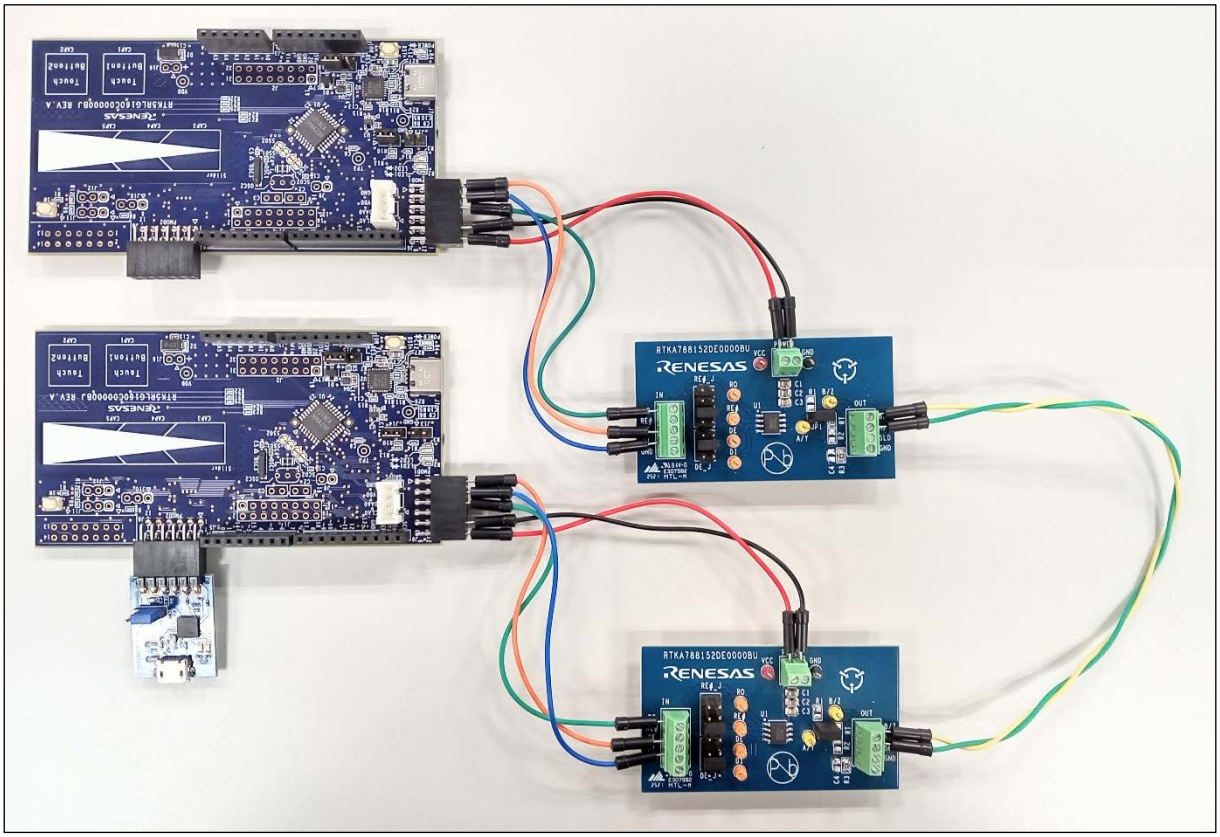


図 6-2 ボード接続写真

6.2.2 ファームウェア定数設定

modbus.h に定義されている定数 MODBUS_MODE を MODBUS_RTU_SLAVE_MODE に設定してプロジェクトをビルド後、スレーブ側の RL78 Fast Prototyping Board のデバッグツールへダウンロードします。

modbus.h に定義されている定数 MODBUS_MODE を MODBUS_RTU_MASTER_MODE に設定してプロジェクトをビルド後、マスタ側の RL78 Fast Prototyping Board のデバッグツールへダウンロードします。

※例では RL78 はスレーブモード(RTU)、マスタモード(RTU)での通信を行いますが MODBUS_MODE の設定値を変更することで通信方式の変更が可能です。設定値については 5.8.1 Modbus 動作設定定数を参照してください。

6.2.3 実行手順

スレーブ側の RL78 のプログラムを実行後、マスタ側の RL78 のプログラムを実行することでデモ動作を開始します。

マスタ側のログ出力:

```
[TX]010100000001FDCA
[RX]010101019048
CALL_BACK_FUNCTION_READ_COILS
[TX]010100000001FDCA
[RX]010101019048
CALL_BACK_FUNCTION_READ_COILS
```

スレーブ側のログ出力:

```
[RX]010100000001FDCA
CALL_BACK_FUNCTION_READ_COILS
[TX]010101019048
[RX]010100000001FDCA
CALL_BACK_FUNCTION_READ_COILS
[TX]010101019048
```

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/

お問合せ先

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

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2023.09.29	—	第一版
1.10	2023.10.20	—	RTU のフレーム間インターバルの値を変更（サンプルコード）

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. 静電気対策
   CMOS 製品の取り扱いの際は静電気防止を心がけてください。CMOS 製品は強い静電気によってゲート絶縁破壊を生じることがあります。運搬や保存の際には、当社が出荷梱包に使用している導電性のトレーやマガジンケース、導電性の緩衝材、金属ケースなどを利用し、組み立て工程にはアースを施してください。プラスチック板上に放置したり、端子を触ったりしないでください。また、CMOS 製品を実装したボードについても同様の扱いをしてください。
2. 電源投入時の処置
   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. 電源オフ時における入力信号
   当該製品の電源がオフ状態のときに、入力信号や入出力プルアップ電源を入れないでください。入力信号や入出力プルアップ電源からの電流注入により、誤動作を引き起こしたり、異常電流が流れ内部素子を劣化させたりする場合があります。資料中に「電源オフ時における入力信号」についての記載のある製品は、その内容を守ってください。
4. 未使用端子の処理
   未使用端子は、「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。
5. クロックについて
   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。
6. 入力端子の印加波形
   入力ノイズや反射波による波形歪みは誤動作の原因になりますので注意してください。CMOS 製品の入力がノイズなどに起因して、V_{IL}（Max.）から V_{IH}（Min.）までの領域にとどまるような場合は、誤動作を引き起こす恐れがあります。入力レベルが固定の場合はもちろん、V_{IL}（Max.）から V_{IH}（Min.）までの領域を通過する遷移期間中にチャタリングノイズなどが入らないように使用してください。
7. リザーブアドレス（予約領域）のアクセス禁止
   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、将来の拡張機能用に割り付けられている リザーブアドレス（予約領域）があります。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。
8. 製品間の相違について
   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うと、フラッシュメモリ、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、ノイズ幅射量などが異なる場合があります。型名が違う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. 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。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、お客様の責任において、お客様の機器・システムを設計ください。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. 当社製品または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. 当社製品を組み込んだ製品の輸出入、製造、販売、利用、配布その他の行為を行うにあたり、第三者保有の技術の利用に関するライセンスが必要となる場合、当該ライセンス取得の判断および取得はお客様の責任において行ってください。
5. 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。
6. 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。
7. あらゆる半導体製品は、外部攻撃からの安全性を 100%保証されているわけではありません。当社ハードウェア／ソフトウェア製品にはセキュリティ対策が組み込まれているものもありますが、これによって、当社は、セキュリティ脆弱性または侵害（当社製品または当社製品が使用されているシステムに対する不正アクセス・不正使用を含みますが、これに限りません。）から生じる責任を負うものではありません。当社は、当社製品または当社製品が使用されたあらゆるシステムが、不正な改変、攻撃、ウイルス、干渉、ハッキング、データの破壊または窃盗その他の不正な侵入行為（「脆弱性問題」といいます。）によって影響を受けないことを保証しません。当社は、脆弱性問題に起因しまたはこれに関連して生じた損害について、一切責任を負いません。また、法令において認められる限りにおいて、本資料および当社ハードウェア／ソフトウェア製品について、商品性および特定目的との合致に関する保証ならびに第三者の権利を侵害しないことの保証を含め、明示または黙示のいかなる保証も行いません。
8. 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。
9. 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。
10. 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。
11. 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。
12. 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。
13. 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。
14. 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.5.0-1　2020.10)

本社所在地

〒135-0061 東京都江東区豊洲 3-2-24（豊洲フォレシア）

www.renesas.com

お問合せ窓口

弊社の製品や技術、ドキュメントの最新情報、最寄の営業お問合せ窓口に関する情報などは、弊社ウェブサイトをご覧ください。

www.renesas.com/contact/

商標について

ルネサスおよびルネサスロゴはルネサス エレクトロニクス株式会社の商標です。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。