RL78/G10

Wi-Fi モジュール制御（ESP-WROOM-02 TCP 送受信）

要旨

本アプリケーションノートでは、RL78/G10 を使用し Wi-Fi モジュール ESP-WROOM-02 を制御するためのソフトウェアとその応用サンプルプログラムの使用方法を説明します。

ESP-WROOM-02 は AT コマンドをサポートしています。RL78/G10 から UART 通信で AT コマンドを送信することにより、ESP-WROOM-02 を制御します。

本制御用ソフトウェアを使うことにより、クライアントとして TCP/IP データ送受信ができます。

動作確認デバイス

Espressif Systems 製 ESP-WROOM-02 (ESP8266EX)

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

Wi-Fi モジュール ESP-WROOM-02 を RL78/G10 から UART 通信で制御します。

同梱するサンプルプログラムではクライアントとしてサーバに接続し、TCP/IP データ送受信を行います。

表 1.1 にシステム機能概略、図 1.1 にシステム構成例を示します。

表 1.1 システム機能概略

無線通信 I/F	無線 LAN 規格 • IEEE 802.11 b/g/n (2.4GHz Wi-Fi)
	対応プロトコル • IPv4 TCP • セキュリティ WPA/WPA2 • 暗号化 AES
マイコンとの接続 I/F	マスタ： ルネサスエレクトロニクス製 RL78/G10 スレーブ：Espressif Systems 製 ESP-WROOM-02 (ESP8266EX) 通信方式：UART、115200 bps モジュール制御 • AT コマンド (UART 通信) による制御 • Station モードとしての使用 • アクセスポイントとの接続、切断 • クライアントとしてネットワークへのデータ送信、受信

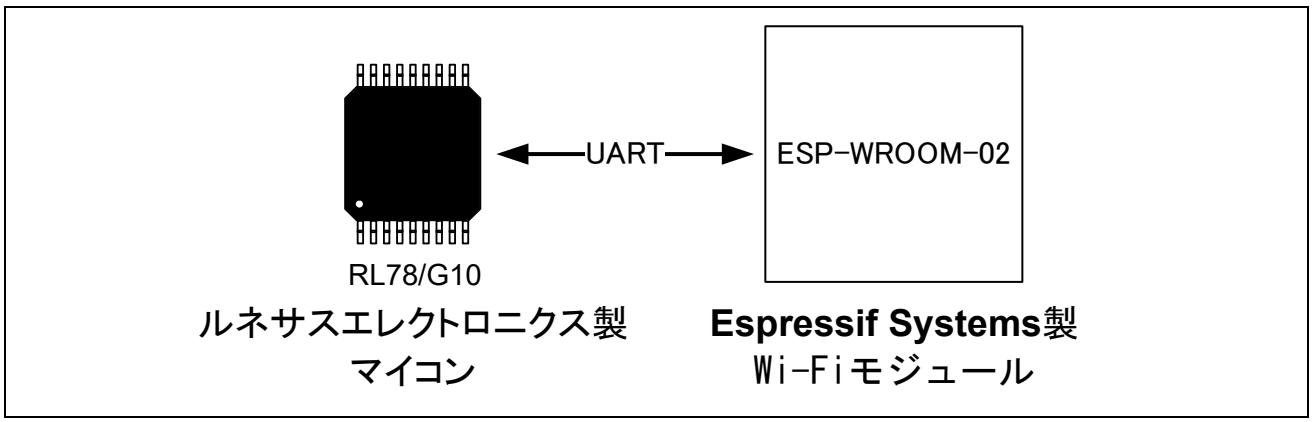


図 1.1 システム構成例

1.1 Wi-Fi モジュール ESP-WROOM-02 (ESP8266EX)の動作概要

1.1.1 ブートモード

ESP-WROOM-02 にはブートモードとして､Flash メモリに書き込まれたプログラムを実行する Flash Boot Mode と、プログラムを Flash メモリに書き込むための UART Download Mode があります。

本アプリケーションノートでは Flash Boot Mode のみを使用し、予め書き込まれた公式ファームウェアを実行します。ブートモードは、Wi-Fi モジュールリセット時の端子状態で選択されます。本アプリケーションノートでは、表 1.2 の通りに設定します。

表 1.2 ブートモード設定方法

ブートモード	GPIO15	GPIO2	GPIO0
Flash Boot Mode	L	H	H

1.1.2 制御方法

UART 通信を使用し、AT コマンドで ESP-WROOM-02 を制御します。本制御用ソフトウェアで提供する API 関数を実行することで、Wi-Fi モジュールへの AT コマンドの送信、リザルトコードの受信が可能です。

ESP8266EX における AT コマンド（モジュールに送る命令）と、リザルトコード（モジュールからの返答）のフォーマットは、以下の様になっています。

AT コマンド：

A	T	コマンド	パラメータ	CR	LF

リザルトコード：

CR	LF	リザルトコード	CR	LF

注意. ESP-WROOM-02 の AT コマンド詳細については、ESP8266EX の AT 命令セットを参照してください。

1.1.3 動作モード

ESP-WROOM-02 には動作モードとして、Station モード、SoftAP モード、SoftAP+Station モードが用意されています。

Station モードでは、Wi-Fi モジュールを無線クライアント（子機）として使用します。通常、無線クライアント（子機）はアクセスポイントと接続して使用されます。SoftAP モードでは、Wi-Fi モジュールをアクセスポイント（基地局）として使用します。

本アプリケーションノートでは、Wi-Fi モジュールをアクセスポイントに接続することを想定して Station モードのみに対応します。

2. 動作確認条件

本アプリケーションノートの制御ソフトウェアとその応用サンプルコードは、下記の条件で動作を確認しております。

表 2.1 動作確認条件

項目	内容
使用マイコン	RL78/G10 (R5F10Y47) ROM: 4 KB, RAM: 512 B
動作周波数	高速オンチップ・オシレータ・クロック (fIH): 20 MHz CPU/周辺ハードウエア・クロック: 20 MHz
動作電圧	3.3V
統合開発環境 (CS+)	ルネサス エレクトロニクス製 CS+ for CC V8.07.00
C コンパイラ(CS+)	ルネサス エレクトロニクス製 CC-RL V1.11.00
統合開発環境 (e^2 studio)	ルネサス エレクトロニクス製 e^2 studio V2022-01 (22.1.0)
C コンパイラ (e^2 studio)	ルネサス エレクトロニクス製 CC-RL V1.11.00
統合開発環境 (IAR)	IAR Systems 製 IAR Embedded Workbench for Renesas RL78 V4.21.3
C コンパイラ (IAR)	IAR Systems 製 IAR C/C++ Compiler for Renesas RL78 V4.21.3.2447
Wi-Fi モジュール	Espressif Systems 製 ESP-WROOM-02 (ESP8266EX) AT version:1.7.3.0 SDK version:3.0.3

3. ハードウエア説明

3.1 ハードウエア構成例

図 3.1 に本アプリケーションノートで使用するハードウエア構成例を示します。

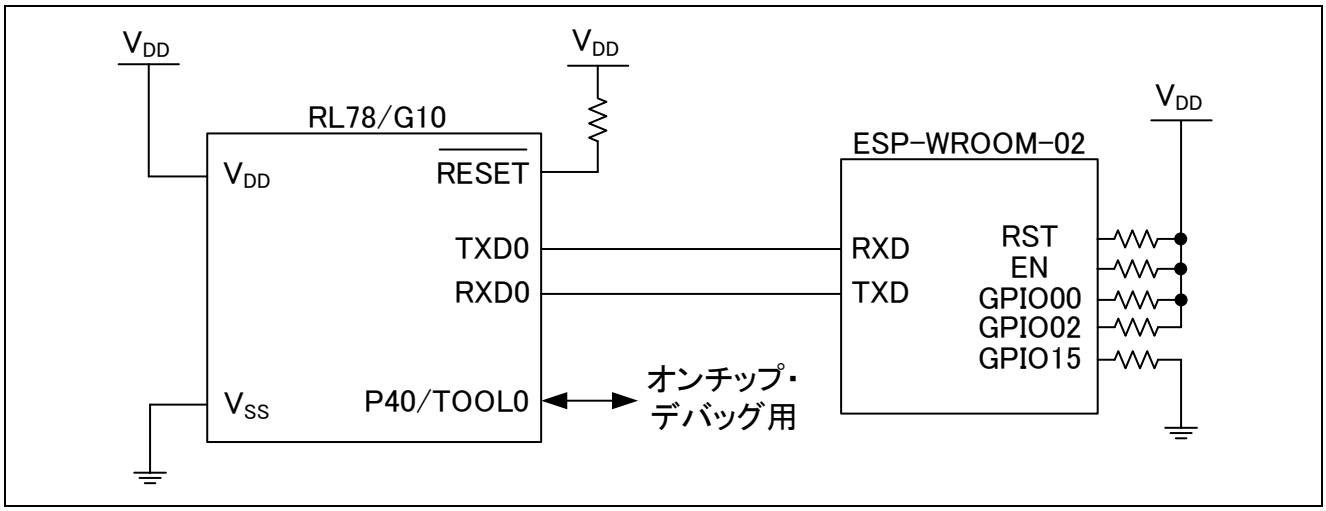


図 3.1 ハードウエア構成

注意 1. この回路イメージは接続の概要を示す為に簡略化しています。実際に回路を作成される場合は、端子処理などを適切に行い、電気的特性を満たすように設計してください。(入力専用ポートは個別に抵抗を介して VDD 又は VSS に接続して下さい) 。

3.2 使用端子一覧

表 3.1 にマイコンの使用端子と機能を示します。

表 3.1 使用端子と機能

端子名	入出力	内容
TXD0	出力	シリアル・データ送信 (UART)
RXD0	入力	シリアル・データ受信 (UART)

4. ソフトウェア説明

4.1 ソフトウェア構成

図 4.1 に本制御用ソフトウェアを使用したソフトウェア構成例を示します。

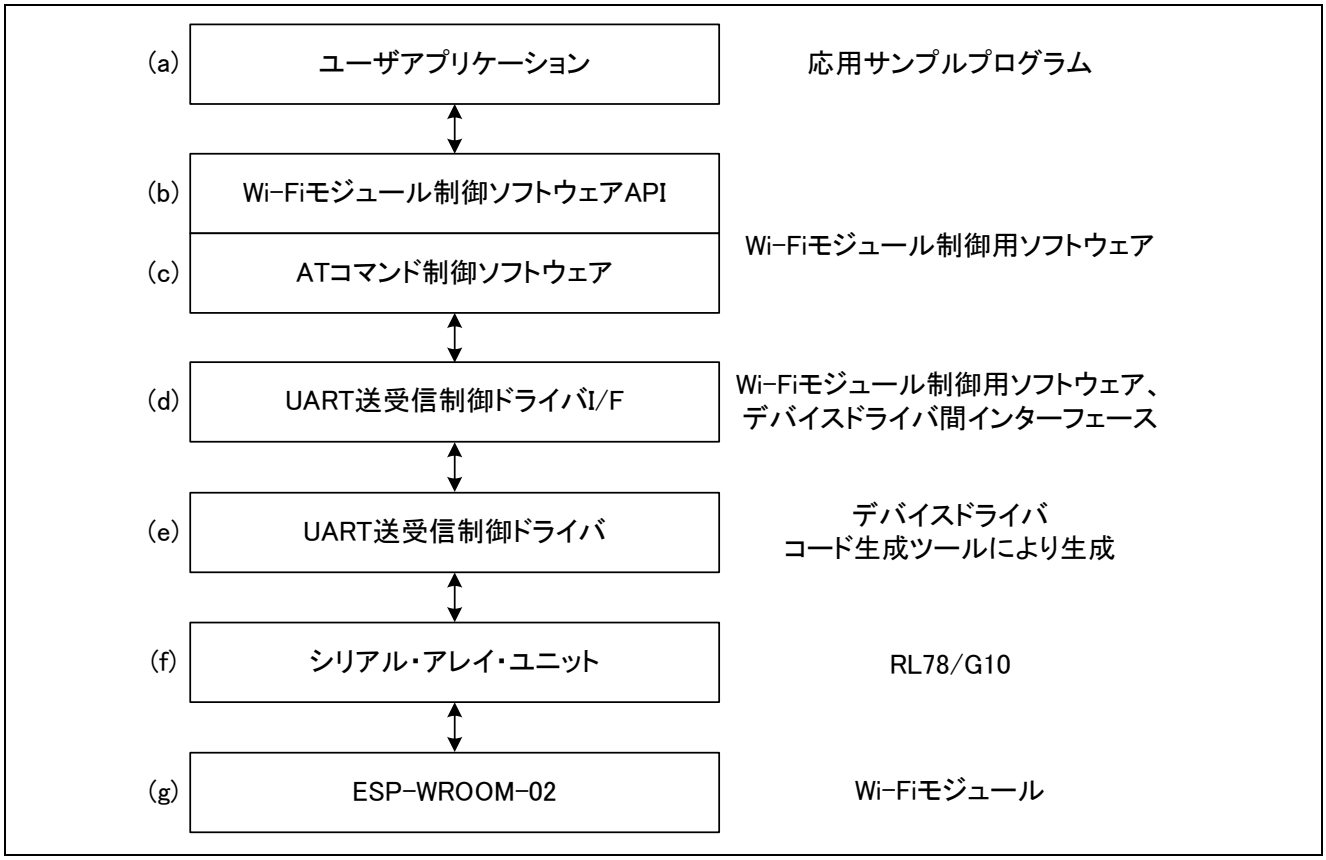


図 4.1 ソフトウェア構成例

(a) ユーザアプリケーション
本制御用ソフトウェアの API をコールするアプリケーションです。例として、応用サンプルプログラムを同梱しています。

(b) Wi-Fi モジュール制御ソフトウェア API
本制御用ソフトウェアの API です。

(c) AT コマンド制御ソフトウェア
Wi-Fi モジュールとの UART 送受信を制御するためのソフトウェアです。

(d) UART 送受信制御ドライバ I/F
制御ソフトウェア、ドライバ間のインタフェースです。使用するデバイス、UART 構成に応じて設定し直してください。

(e) UART 送受信制御ドライバ
UART 通信制御用のソフトウェアです。ルネサス統合開発環境で用意されているコード生成ツールを使用して RL78/G10 用ソフトウェアを生成しています。
使用するデバイスに応じてコード生成を行ってください。

(f) シリアル・アレイ・ユニット
シリアル・アレイ・ユニット(SAU)を UART 送受信モードで使用します。

(g) ESP-WROOM-02
Wi-Fi モジュールです。

4.2 必要メモリサイズ (CC-RL V1.11.00)

使用メモリ	サイズ [Byte]
ROM	3144
RAM	140
スタックサイズ	184

注意. 必要メモリサイズは C コンパイラのバージョンやコンパイルオプションにより異なります。
使用メモリは応用サンプルプログラムのメモリサイズを含みます。

4.3 オプション・バイトの設定一覧

表 4.1 にオプション・バイトの設定一覧を示します。

表 4.1 オプション・バイト設定一覧

アドレス	設定値	内容
000C0H	11101111B	ウォッチドッグ・タイマ動作禁止 (リセット解除後、カウント停止)
000C1H	11110111B	SPOR検出電圧: 立ち上がり時: TYP. 2.90V（2.76～3.02V） 立ち下がり時: TYP. 2.84V（2.70～2.96V）
000C2H	11111001B	高速オンチップ・オシレータ・クロックの 動作周波数: 20 MHz
000C3H	10000101B	オンチップ・デバッグ動作許可

4.4 フォルダ構成

本アプリケーションノートのフォルダ構成を表 4.2 に示します。

表 4.2 ディレクトリ構成

ディレクトリ構造	説明	図 4.1 との対応
¥an-r01an4791xx0110-rl78g10-communication <DIR>	本アプリケーションのルートフォルダ	-
r01an4791jj0110-rl78g10.pdf	本アプリケーションノート	-
¥workspace¥{IDE}¥ESP_CS+¥src <DIR>	プログラム格納用フォルダ	-
¥sample <DIR>	応用サンプルプログラム格納用フォルダ	-
testmain.c	応用サンプルプログラム	(a)
¥r_esp_wroom_02 <DIR>	Wi-Fi モジュール制御用プログラム格納用フォルダ	-
r_esp.c	Wi-Fi モジュール制御用プログラム	(b), (c)
r_esp.h	Wi-Fi モジュール制御用プログラム ヘッダファイル	(b), (c)
¥cg_src <DIR>	ソースファイル格納用フォルダ	-
r_cg_cgc.c 注意 1	CGC モジュール	(a)
r_cg_cgc.h 注意 1	CGC モジュール ヘッダファイル	(a)
r_cg_cgc_user.c 注意 1	CGC モジュール	(a)
r_cg_macrodriver.h 注意 1	ヘッダファイル	(a)
r_cg_main.c 注意 1	メイン処理モジュール	(a)
r_cg_sau.c 注意 1	SAU モジュール	(e)
r_cg_sau.h 注意 1	SAU モジュール ヘッダファイル	(e)
r_cg_sau_user.c 注意 1	SAU モジュール	(e)
r_cg_systeminit.c 注意 1	システム初期化モジュール	(a)
r_cg_userdefine.h 注意 1	ユーザ定義ファイル	(a)
r_esp_hw.c 注意 1	ドライバ I/F	(d)
r_esp_hw.h 注意 1	ドライバ I/F ヘッダファイル	(d)

注意 1. 使用するデバイスに応じて、コード生成ツールで生成してください。既存プロジェクトに統合する場合は、既存ソフトウェアと競合しないか確認してください。

4.5 戻り値

表 4.3 に制御用 API で使用する戻り値を示します。

表 4.3 API 関数戻り値

定数名	内容
ESP_OK	Wi-Fiモジュールからの応答: “OK”
ESP_ERROR	Wi-Fiモジュールからの応答: “ERROR”
ESP_ALREADY_CONNECT	Wi-Fiモジュールからの応答: “ALREADY CONNECT”
ESP_TIMEOUT	タイムアウト
ESP_OVERFLOW	バッファのオーバーフロー発生
ESP_OTHERS	その他のエラー

4.6 定数一覧

表 4.4 に制御用 API で使用する定数を示します。

表 4.4 定数一覧

定数名	設定値	内容
FCLK_MHZ	20	fCLK 周波数 MHz
ESP_RECV_BUFSIZE	128	UART受信バッファサイズ
ESP_LOOPNUM_1MS	FCLK_MHZ * 143	1 ms ウエイト用ループ回数

4.7 構造体一覧

表 4.5 に制御用 API で使用する構造体を示します。

表 4.5 構造体一覧

[構造体名] esp_params_t		
概 要	Wi-Fi モジュール制御パラメータ格納用構造体	
変数	uint8_t * const ssid	アクセスポイントの SSID
	uint8_t * const pwd	アクセスポイントのパスワード
	uint16_t timeout_ms	Wi-Fi モジュールからの応答待ちタイムアウト時間 [ms]
	uint8_t ip_address[15]	IP アドレス格納用バッファ
	uint8_t mac_address[17]	MAC アドレス格納用バッファ
	uint8_t * const ip_address_target	IP address of TCP server
	uint8_t * const port_target	Port number of TCP server

4.8 API 関数一覧

表 4.6 に制御用 API の関数一覧を示します。

表 4.6 API 関数一覧

関数名	概要
R_ESP_Init	Wi-Fiモジュール初期化
R_ESP_SendCommandToESP	Wi-FiモジュールへのATコマンド送信
R_ESP_Receive	Wi-Fiモジュールからのデータ受信開始
R_ESP_WaitResultFromESP	Wi-Fiモジュールからのリザルトコード受信待ち
R_ESP_ConnectToAP	アクセスポイントへの接続
R_ESP_DisconnectFromAP	アクセスポイントからの切断
R_ESP_TCP_Open	TCPコネクション確立
R_ESP_TCP_Close	TCPコネクション終了
R_ESP_GetIPAddress	Wi-FiモジュールのIPアドレス取得
R_ESP_GetMACAddress	Wi-FiモジュールのMACアドレス取得
R_ESP_SendDataToNetwork	ネットワークへのデータ送信
R_ESP_WaitDataFromNetwork	ネットワークからのデータ受信待ち
R_ESP_GetDataFromNetwork	ネットワークからのデータ取得
R_WaitMilliSeconds	ミリ秒ウエイト
R_Wait1MilliSecond	1ミリ秒ウエイト
R_strstr_WithTail	文字列中の任意のパターンを検索

4.9 API 関数仕様

Wi-Fi モジュール制御 API 関数の詳細を以下に示します。

4.9.1 Wi-Fi モジュール初期化

[関数名] R_ESP_Init		
概　要	Wi-Fi モジュール初期化	
ヘッダ	r_esp.h	
宣　言	esp_err_t R_ESP_Init(esp_params_t* params);	
説　明	• Wi-Fi モジュールの初期設定を行います。 AT コマンドエコー: オフ Wi-Fi モード: Station モード • システム起動時に一度だけ呼び出してください。	
引　数	esp_params_t* param	Wi-Fi モジュール制御パラメータ
リターン値	• 初期化結果を返します。	
	ESP_OK	Wi-Fi モジュールからの応答: “OK”
	ESP_ERROR	Wi-Fi モジュールからの応答: “ERROR”
	ESP_TIMEOUT	タイムアウト
備考	なし	

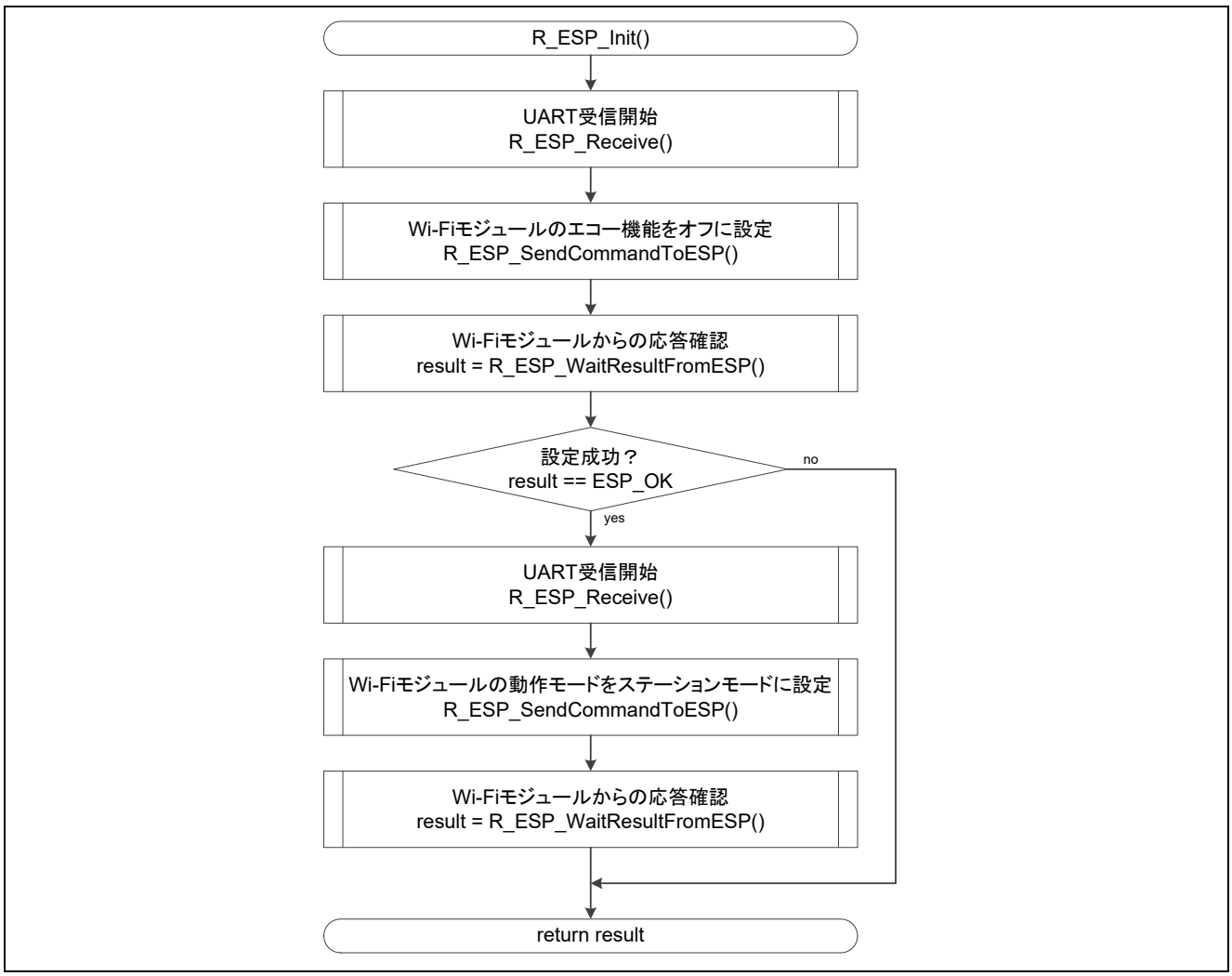


図 4.2 Wi-Fi モジュール初期化

4.9.2 Wi-Fi モジュールへの AT コマンド送信

[関数名] R_ESP_SendCommandToESP	
概　要	Wi-Fi モジュールへの AT コマンド送信
ヘッダ	r_esp.h
宣　言	esp_err_t R_ESP_SendCommandToESP(uint8_t * const buf_command);
説　明	• 引数で受け取った AT コマンドを Wi-Fi モジュールに送信します。 • コマンドの終端には CR+LF、ヌル文字('¥0')を付加してください。
引　数	uint8_t * const command　送信コマンド格納領域の先頭アドレス
リターン値	ESP_OK
備考	• Wi-Fi モジュールへ AT コマンドを送信し、リザルトコードを受信するためには、以下の関数を順番にコールしてください。 1. R_ESP_Receive() 2. R_ESP_SendCommandToESP() 3. R_ESP_WaitResultFromESP()

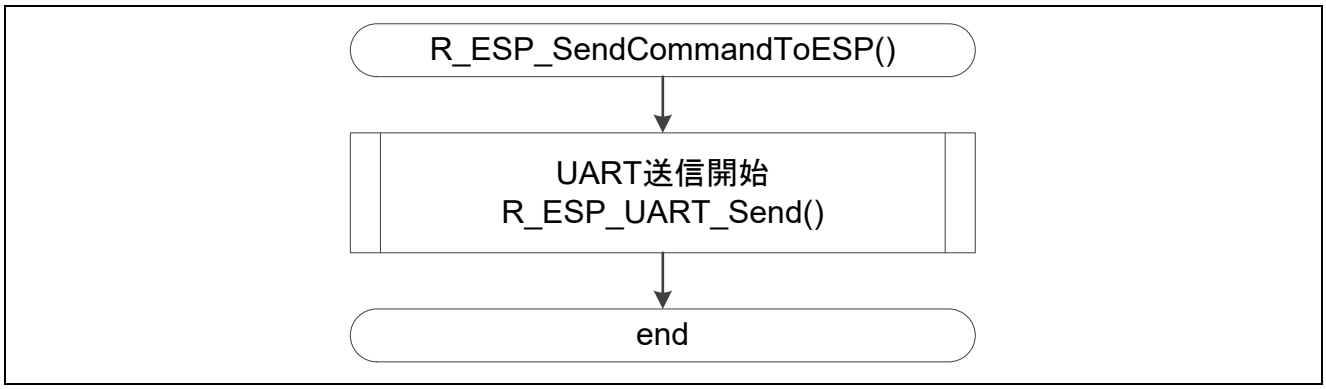


図 4.3 Wi-Fi モジュールへの AT コマンド送信

4.9.3 Wi-Fi モジュールからのデータ受信開始

[関数名] R_ESP_Receive

概　要	Wi-Fi モジュールからのデータ受信開始
ヘッダ	r_esp.h
宣　言	esp_err_t R_ESP_Receive(esp_params_t* params);
説　明	• 受信用バッファをクリアした後、UART ドライバをコールし UART 受信動作を開始します。
引　数	esp_params_t* param　　Wi-Fi モジュール制御パラメータ
リターン値	ESP_OK
備考	なし

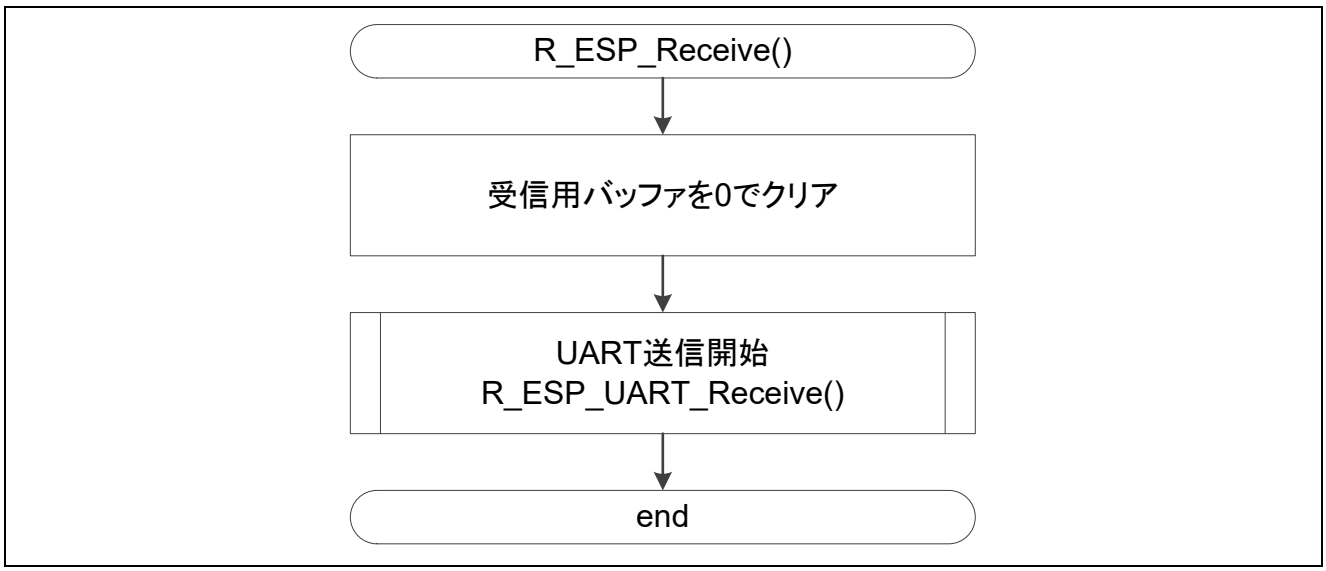


図 4.4 Wi-Fi モジュールからのデータ受信開始

4.9.4 Wi-Fi モジュールからのリザルトコード受信待ち

[関数名]	R_ESP_WaitResultFromESP	
概　要	Wi-Fi モジュールからのリザルトコード受信待ち	
ヘッダ	r_esp.h	
宣　言	esp_err_t R_ESP_WaitResultFromESP(esp_params_t* params);	
説　明	• UART 受信バッファを監視し、受信したデータに”OK”, “ERROR”, “ALREADY_CONNECT” のパターンが現れるまで、もしくはタイムアウト時間までウエイトします。	
引　数	esp_params_t* param	Wi-Fi モジュール制御パラメータ
リターン値	• リザルトコードの受信結果を返します。	
	ESP_OK	Wi-Fi モジュールからの応答: “OK”
	ESP_ERROR	Wi-Fi モジュールからの応答: “ERROR”
	ESP_ALREADY_CONNECT	Wi-Fi モジュールからの応答: “ALREADY_CONNECT”
	ESP_TIMEOUT	タイムアウト
備考	• Wi-Fi モジュールへ AT コマンドを送信し、リザルトコードを受信するためには、以下の関数を順番にコールしてください。 1. R_ESP_Receive() 2. R_ESP_SendCommandToESP() 3. R_ESP_WaitResultFromESP()	

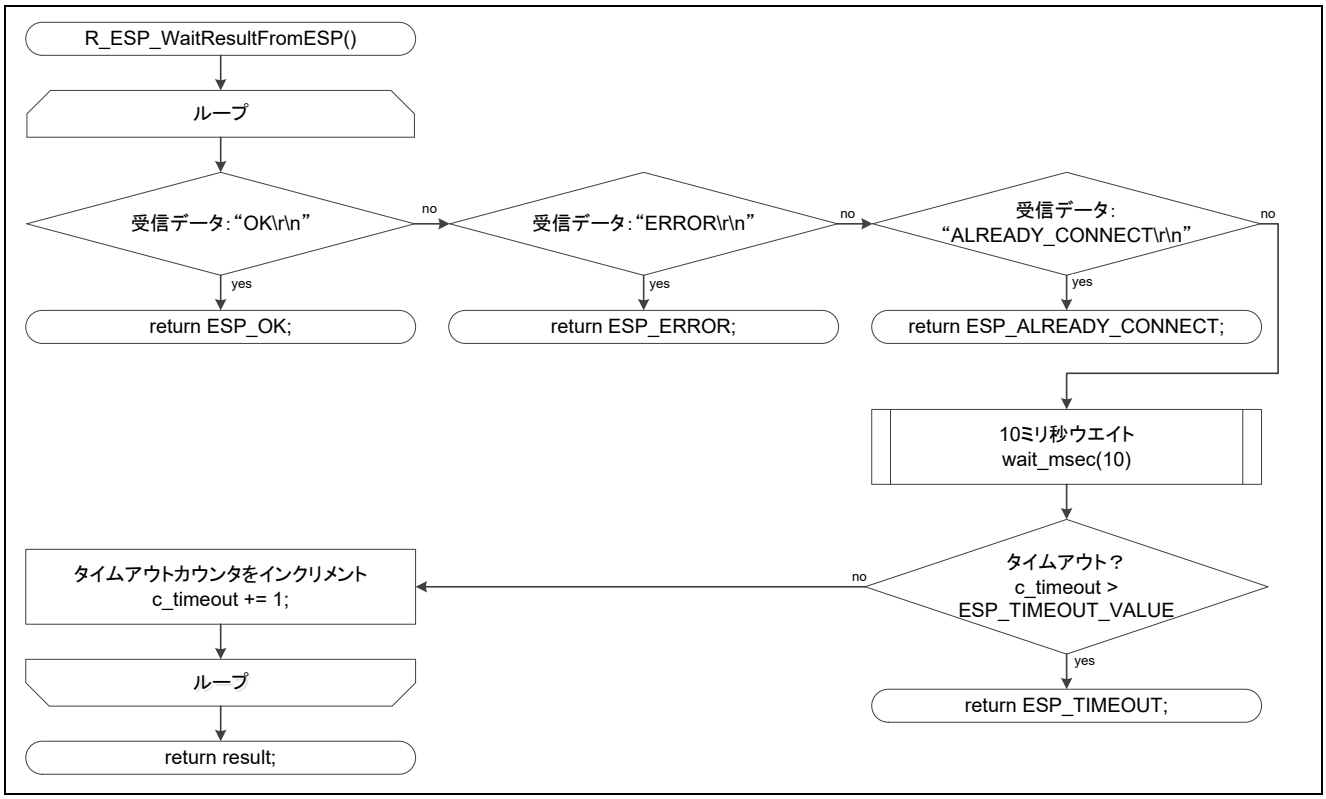


図 4.5 Wi-Fi モジュールからのリザルトコード受信待ち

4.9.5 アクセスポイントへの接続

[関数名]	R_ESP_ConnectToAP	
概　要	アクセスポイントへの接続	
ヘッダ	r_esp.h	
宣　言	esp_err_t R_ESP_ConnectToAP(esp_params_t* params);	
説　明	• Wi-Fi モジュールをアクセスポイントに接続します。	
引　数	esp_params_t* param	Wi-Fi モジュール制御パラメータ
リターン値	• アクセスポイントへの接続結果を返します。	
	ESP_OK	Wi-Fi モジュールからの応答: “OK”
	ESP_ERROR	Wi-Fi モジュールからの応答: “ERROR”
	ESP_TIMEOUT	タイムアウト
備考	なし	

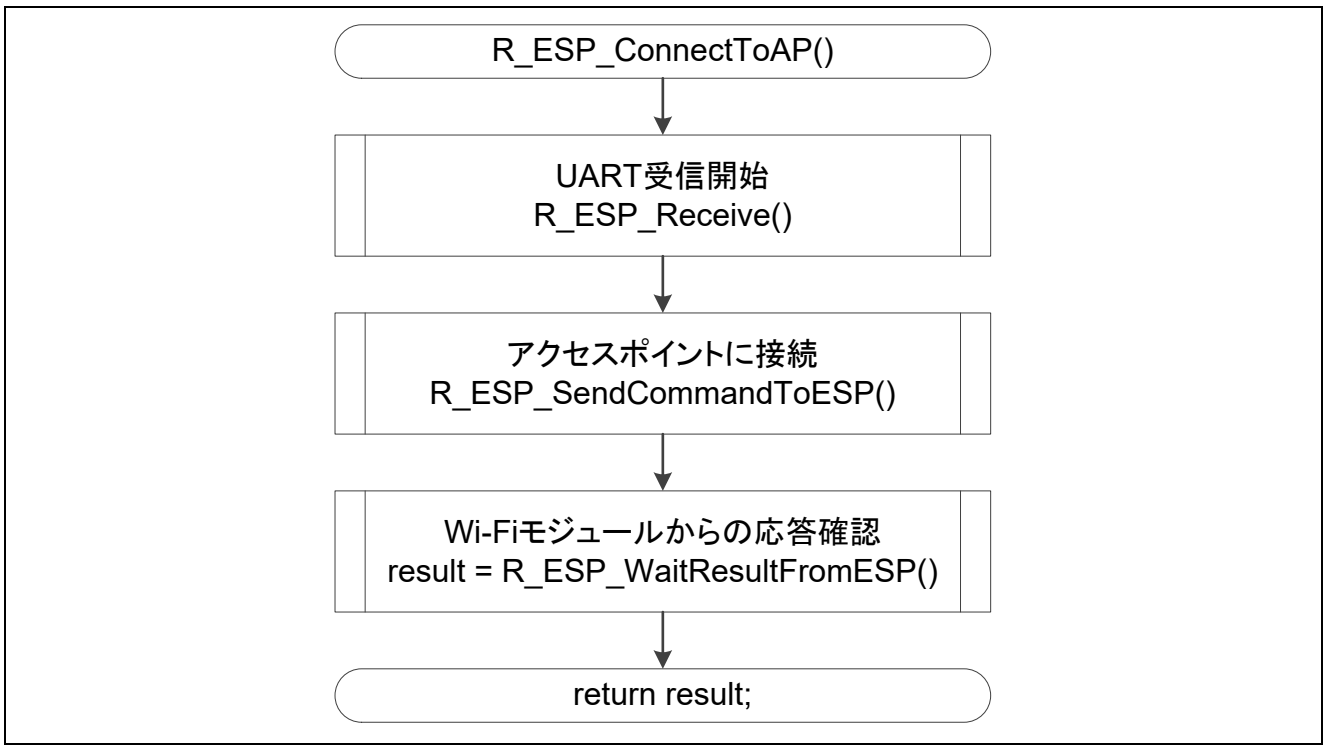


図 4.6 アクセスポイントへの接続

4.9.6 アクセスポイントからの切断

[関数名]	R_ESP_DisconnectFromAP	
概　要	アクセスポイントからの切断	
ヘッダ	r_esp.h	
宣　言	esp_err_t R_ESP_DisconnectFromAP(esp_params_t* params);	
説　明	• 接続中のアクセスポイントから切断します。	
引　数	esp_params_t* param	Wi-Fi モジュール制御パラメータ
リターン値	• アクセスポイントからの切断結果を返します。	
	ESP_OK	Wi-Fi モジュールからの応答: “OK”
	ESP_TIMEOUT	タイムアウト
備考	な	

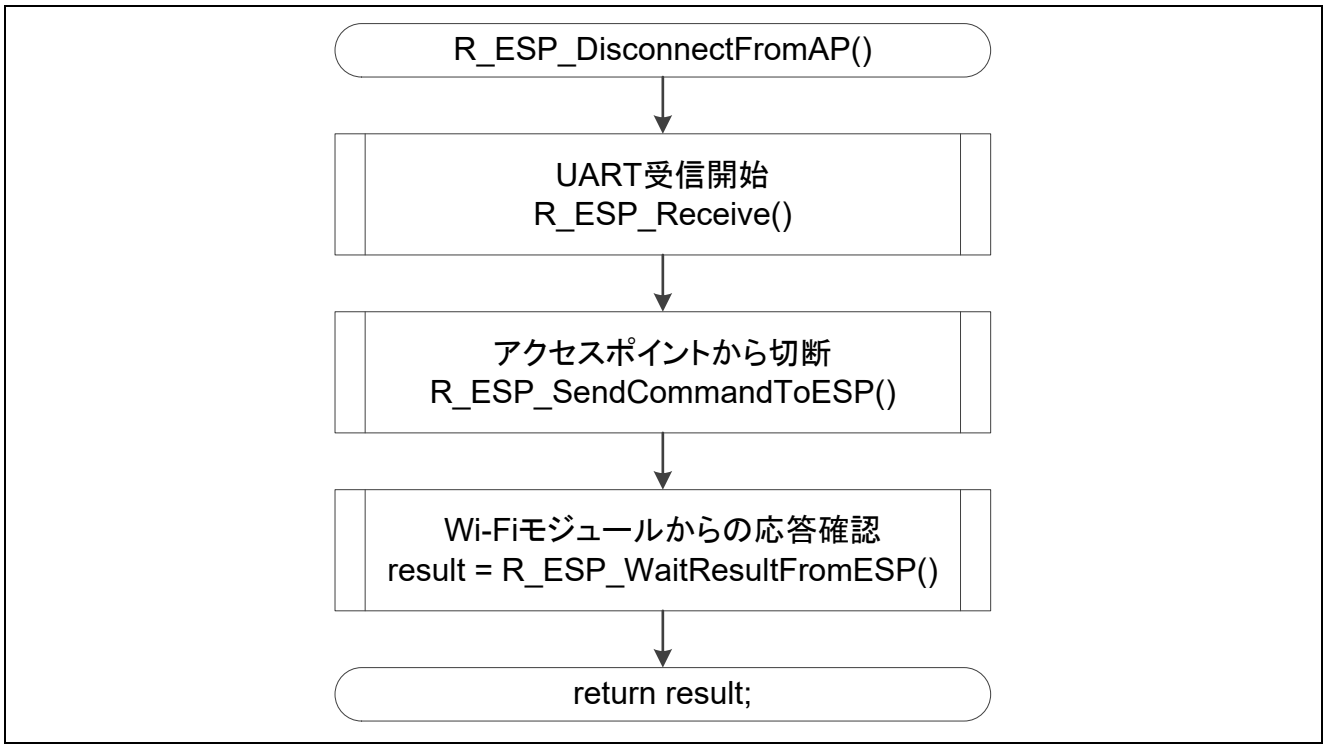


図 4.7 アクセスポイントからの切断

4.9.7 TCP コネクション確立

[関数名]	R_ESP_TCP_Open	
概　要	TCP コネクション確立	
ヘッダ	r_esp.h	
宣　言	esp_err_t R_ESP_TCP_Open(esp_params_t* params);	
説　明	• TCP クライアントとしてサーバへ接続し、コネクションを確立します。	
引　数	esp_params_t* param	Wi-Fi モジュール制御パラメータ
リターン値	TCP コネクション確立結果を返します。	
	ESP_OK	Wi-Fi モジュールからの応答: “OK”
	ESP_ERROR	Wi-Fi モジュールからの応答: “ERROR”
	ESP_ALREADY_CONNECT	Wi-Fi モジュールからの応答: “ALREADY CONNECT”
	ESP_TIMEOUT	タイムアウト
備考	なし	

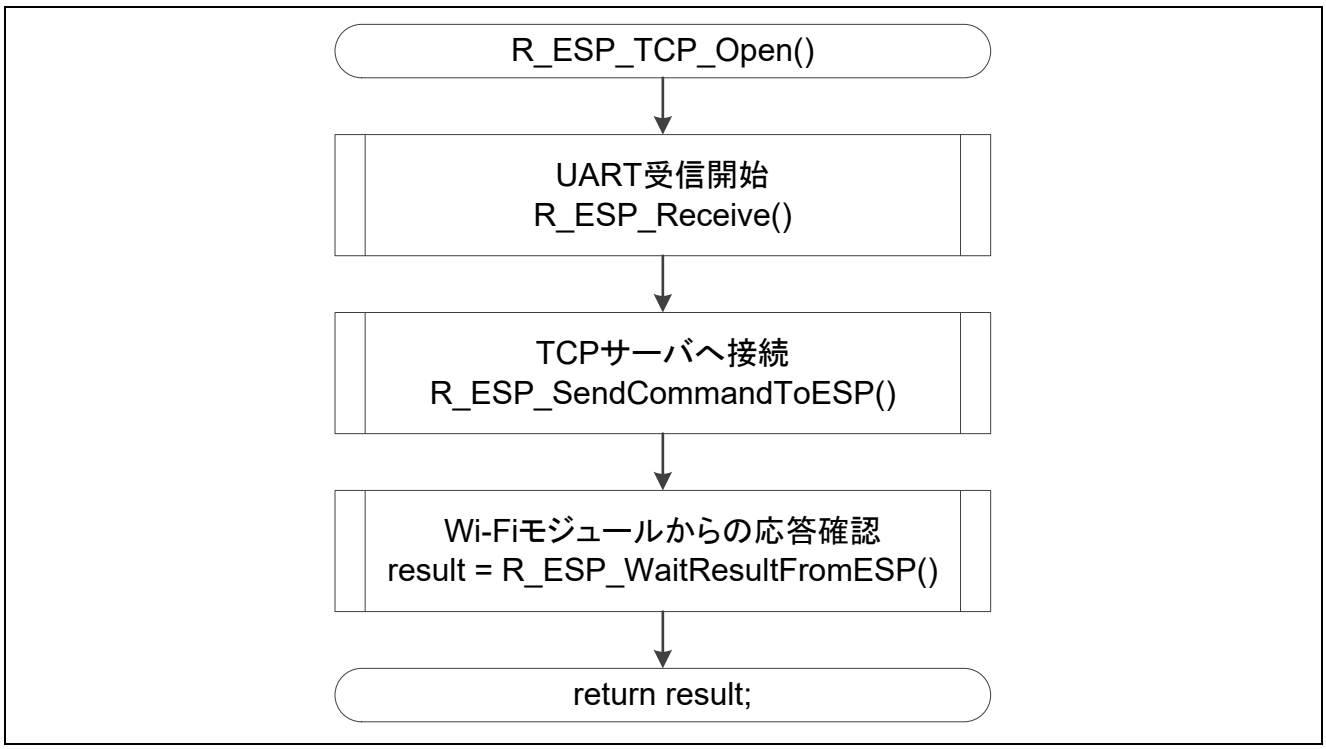


図 4.8 TCP コネクション確立

4.9.8 TCP コネクション終了

[関数名]	R_ESP_TCP_Close	
概　要	TCP コネクション切断	
ヘッダ	r_esp.h	
宣　言	esp_err_t R_ESP_TCP_Close(esp_params_t* params);	
説　明	• TCP コネクションを終了します。	
引　数	esp_params_t* param	Wi-Fi モジュール制御パラメータ
リターン値	• TCP コネクションの終了結果を返します。	
	ESP_OK	Wi-Fi モジュールからの応答: “OK”
	ESP_TIMEOUT	タイムアウト
備考	なし	

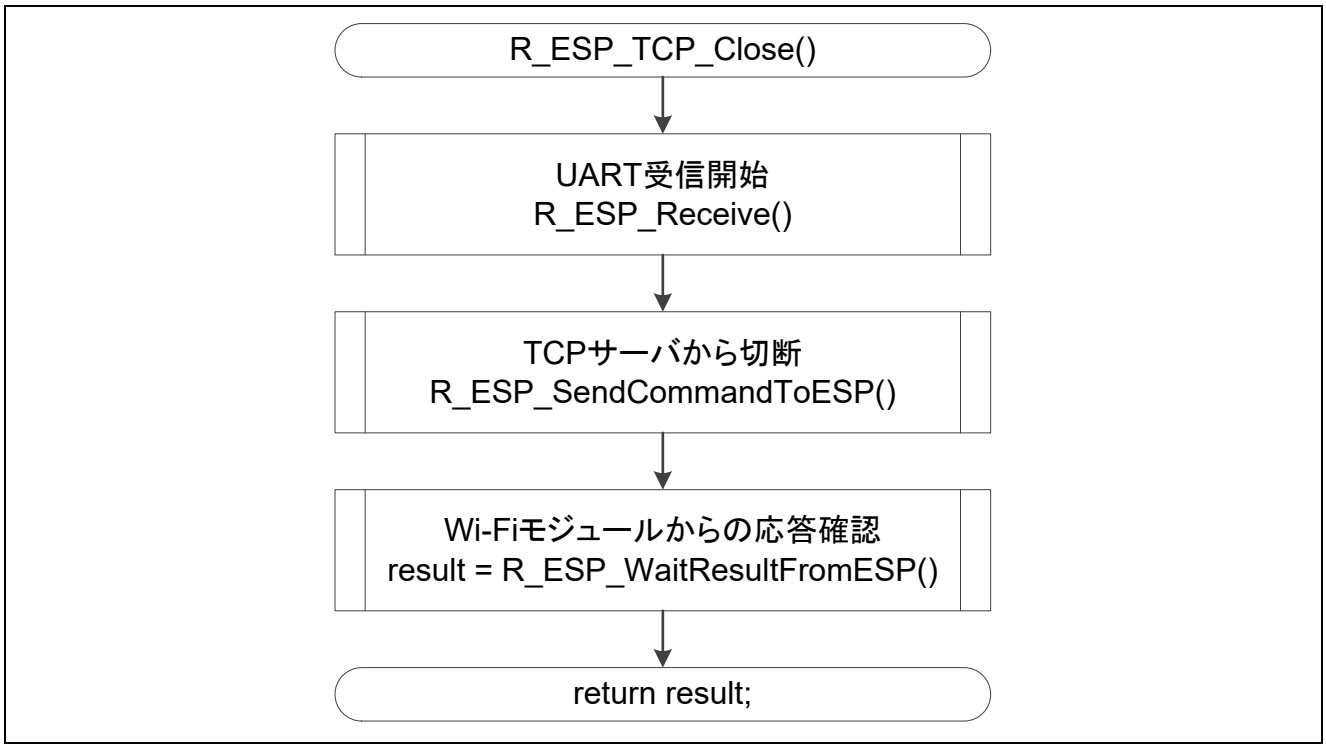


図 4.9 TCP コネクション終了

4.9.9 Wi-Fi モジュールの IP アドレス取得

[関数名]	R_ESP_GetIPAddress	
概　要	Wi-Fi モジュールの IP アドレス取得	
ヘッダ	r_esp.h	
宣　言	esp_err_t R_ESP_GetIPAddress(esp_params_t* params);	
説　明	• Wi-Fi モジュールに割り振られた IP アドレスを取得します。 • IP アドレスのフォーマットは文字列、カンマ区切りで最大 15 文字です。	
引　数	esp_params_t* param	Wi-Fi モジュール制御パラメータ
リターン値	IP アドレスの取得結果を返します。	
	ESP_OK	Wi-Fi モジュールからの応答: “OK”
	ESP_ERROR	Wi-Fi モジュールからの応答: “ERROR”
	ESP_TIMEOUT	タイムアウト
備考	なし	

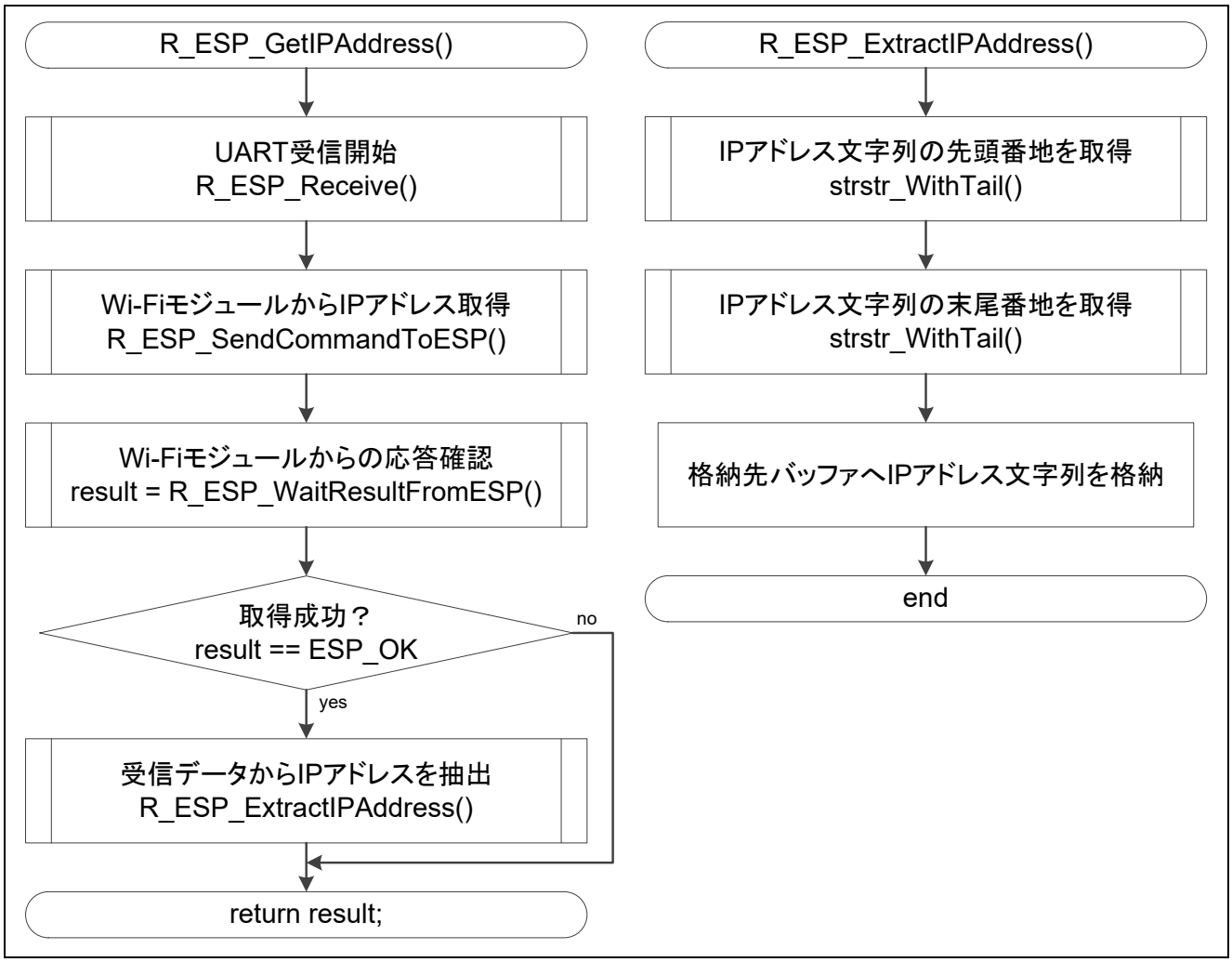


図 4.10　Wi-Fi モジュールの IP アドレス取得

4.9.10 Wi-Fi モジュールの MAC アドレス取得

[関数名]	R_ESP_GetMACAddress	
概　要	Wi-Fi モジュールの MAC アドレス取得	
ヘッダ	r_esp.h	
宣　言	esp_err_t R_ESP_GetMACAddress(esp_params_t* params);	
説　明	• Wi-Fi モジュールの MAC アドレスを取得します。 • MAC アドレスのフォーマットは文字列、セミコロン区切りで最大 17 文字です。	
引　数	esp_params_t* param	Wi-Fi モジュール制御パラメータ
リターン値	MAC アドレスの取得結果を返します。	
	ESP_OK	Wi-Fi モジュールからの応答: “OK”
	ESP_ERROR	Wi-Fi モジュールからの応答: “ERROR”
	ESP_TIMEOUT	タイムアウト
備考	なし	

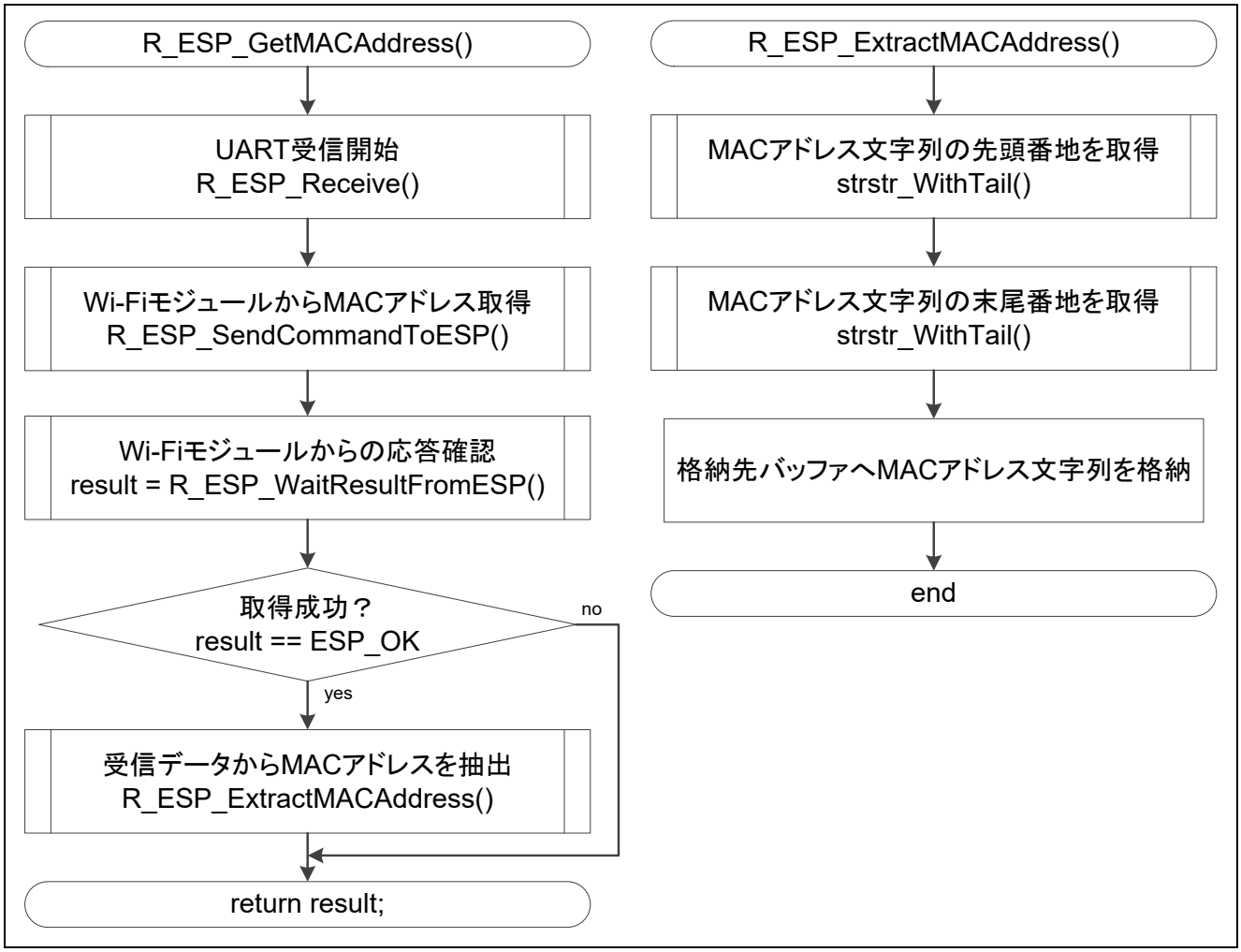


図 4.11　Wi-Fi モジュールの MAC アドレス取得

4.9.11 ネットワークへのデータ送信

[関数名]	R_ESP_SendDataToNetwork	
概　要	ネットワークへのデータ送信	
ヘッダ	r_esp.h	
宣　言	esp_err_t R_ESP_NotifyDataLength(esp_params_t* params, uint8_t * const buf_senddata);	
説　明	•　接続先の IP アドレスへデータを送信します。	
引　数	esp_params_t* param	Wi-Fi モジュール制御パラメータ
	uint8_t * const buf_senddata	送信データ格納領域の先頭アドレス
リターン値	データの送信結果を返します。	
	ESP_OK	Wi-Fi モジュールからの応答: “OK”
	ESP_ERROR	Wi-Fi モジュールからの応答: “ERROR”
	ESP_TIMEOUT	タイムアウト
備考	送信データの最大長: 2048 bytes	

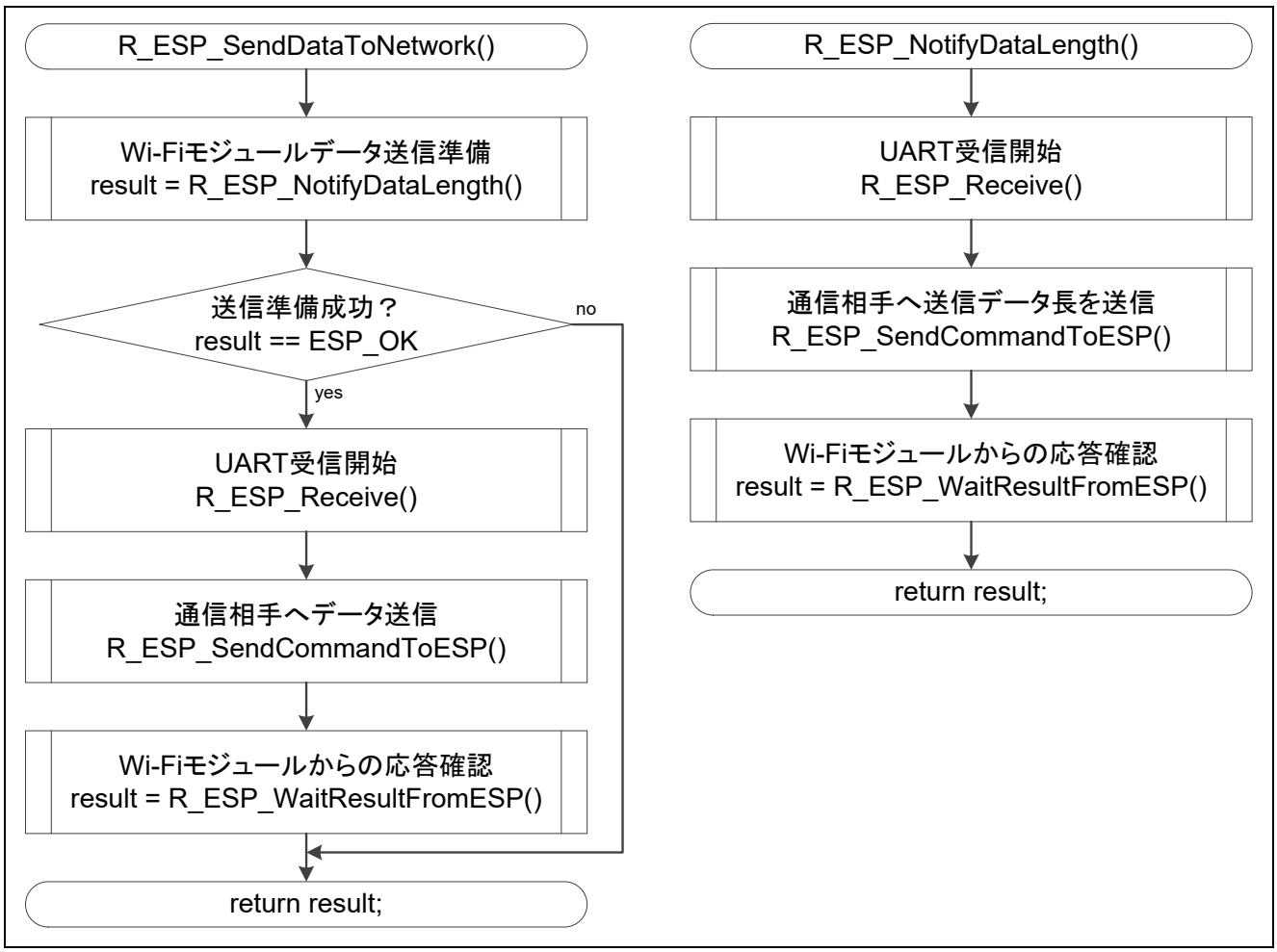


図 4.12　ネットワークへのデータ送信

4.9.12 ネットワークからのデータ受信待ち

[関数名]	R_ESP_WaitDataFromNetwork	
概　要	ネットワークからのデータ受信待ち	
ヘッダ	r_esp.h	
宣　言	esp_err_t R_ESP_SendDataToNetwork(esp_params_t* params, uint8_t *buf_senddata);	
説　明	• UART 受信バッファを監視し、受信したデータに”+IPD:”のパターンが現れるまで、もしくはタイムアウト時間までウエイトします。	
引　数	esp_params_t* param	Wi-Fi モジュール制御パラメータ
リターン値	データ受信待ち結果を返します。	
	ESP_OK	Wi-Fi モジュールからの応答: “OK”
	ESP_TIMEOUT	タイムアウト
備考	なし	

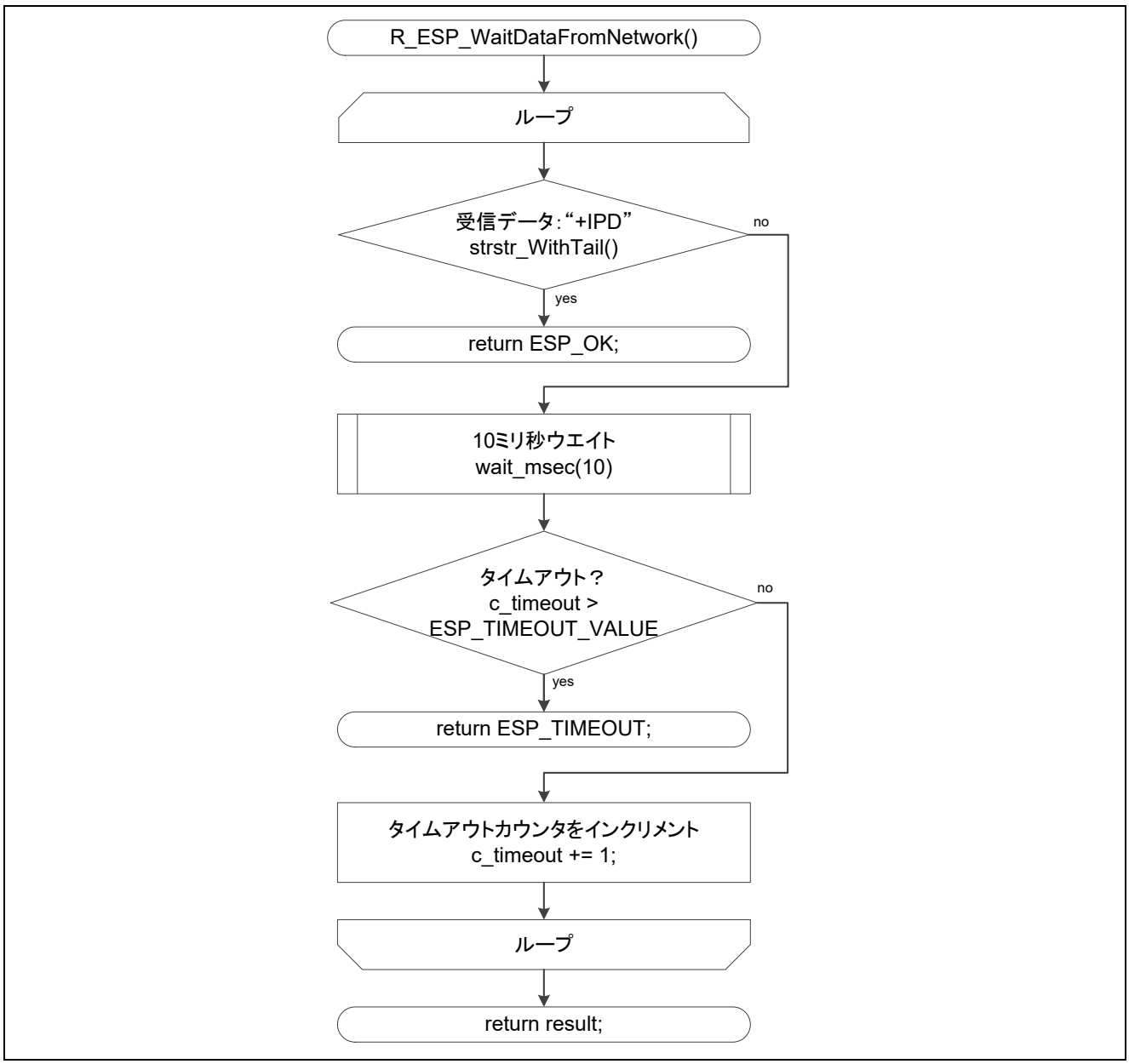


図 4.13　ネットワークからのデータ受信待ち

4.9.13 ネットワークからのデータ取得

[関数名]	R_ESP_GetDataFromNetwork	
概　要	ネットワークからのデータ取得	
ヘッダ	r_esp.h	
宣　言	esp_err_t R_ESP_WaitDataFromNetwork(esp_params_t* params);	
説　明	• 受信用バッファに格納されたリザルトコードの中から、ネットワークデータを抽出します。	
引　数	uint8_t *buf_dest	抽出データ格納先バッファ
	uint8_t buf_length	格納先バッファのサイズ
リターン値	データ取得結果を返します。	
	ESP_OK	Wi-Fi モジュールからの応答: “OK”
	ESP_TIMEOUT	タイムアウト
備考	なし	

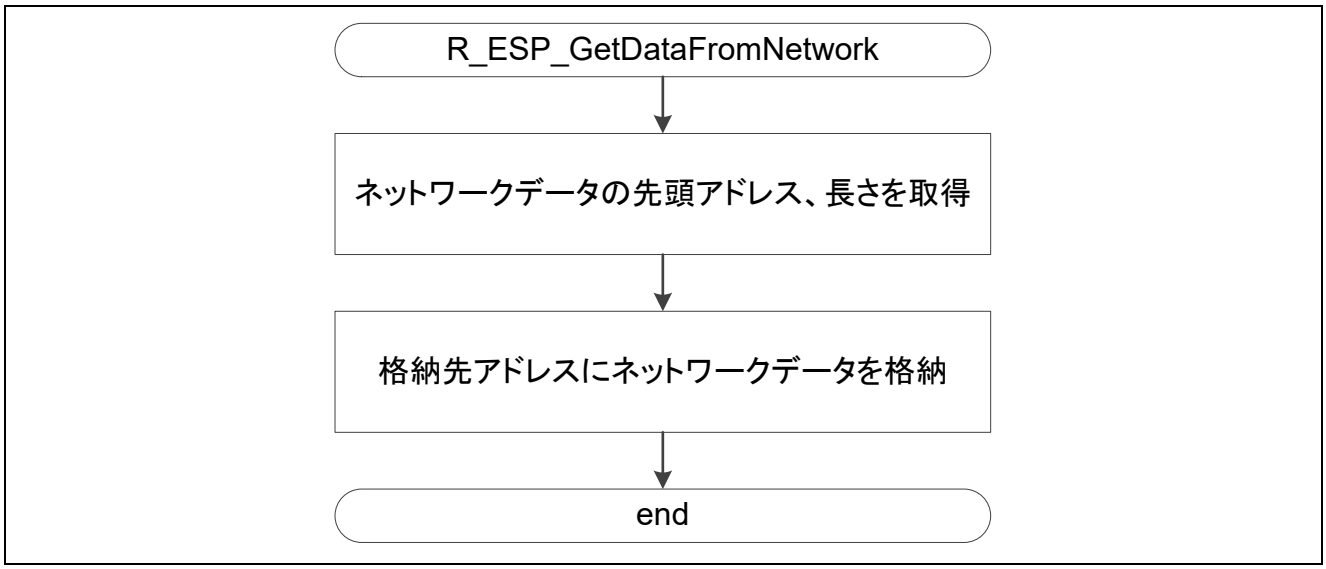


図 4.14　ネットワークからのデータ取得

4.9.14 ミリ秒ウエイト

[関数名]	R_WaitMilliSeconds	
概　要	ミリ秒ウエイト	
ヘッダ	r_esp.h	
宣　言	void R_WaitMilliSeconds(uint16_t msec);	
説　明	• 引数で指定された秒数分、R_Wait1MilliSecond 関数を呼び出します。	
引　数	uint16_t msec	ウエイトミリ秒数
リターン値	なし	
備考	なし	

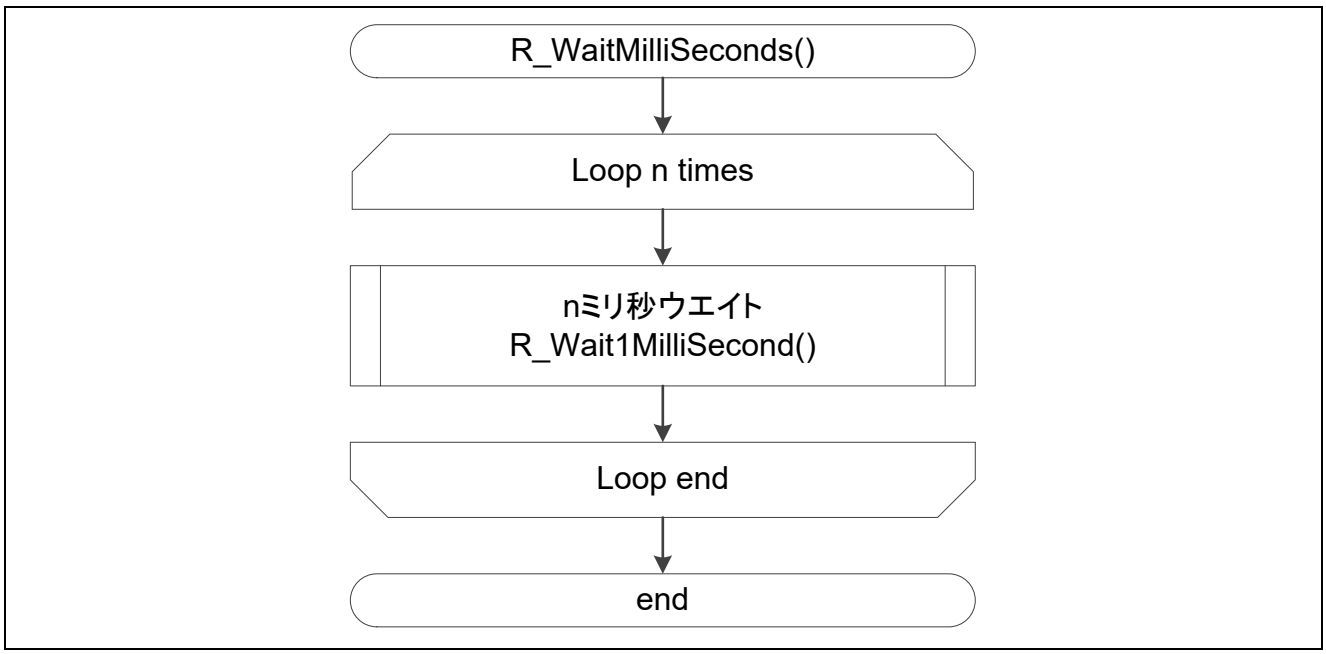


図 4.15 ミリ秒ウエイト

4.9.15 1 ミリ秒ウエイト

[関数名]	R_Wait1MilliSecond
概　要	1 ミリ秒ウエイト
ヘッダ	r_esp.h
宣　言	void R_Wait1MilliSecond(void);
説　明	• CPU によって 1 ミリ秒ウエイトします。
引　数	なし
リターン値	なし
備考	なし

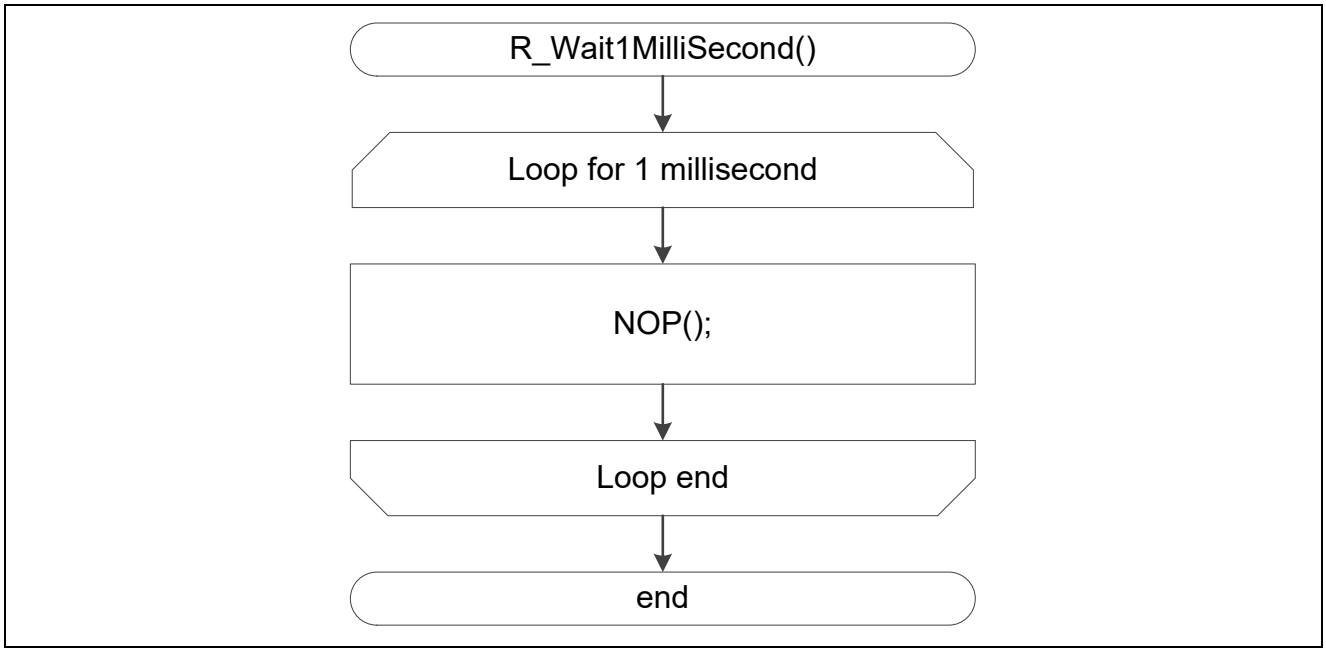


図 4.16　1 ミリ秒ウエイト

4.9.16 文字列中の任意のパターンを検索

[関数名]	R_strstr_WithTail	
概　要	文字列中の任意のパターンを検索	
ヘッダ	r_esp.h	
宣　言	uint8_t* R_strstr_WithTail(uint8_t *str_head, uint8_t *str_tail, uint8_t *key);	
説　明	• 検索対象文字列の中から検索パターンを検索し、検索パターンを最初に発見したアドレスを返します。	
引　数	uint8_t *str_head	検索対象文字列の先頭アドレス
	uint8_t *str_tail	検索対象文字列の末尾アドレス
	uint8_t *key	検索パターン
リターン値	パターン検索結果を返します。	
	¥0	パターンが発見されなかった場合
	上記以外	パターンが最初に現れたアドレス
備考	なし	

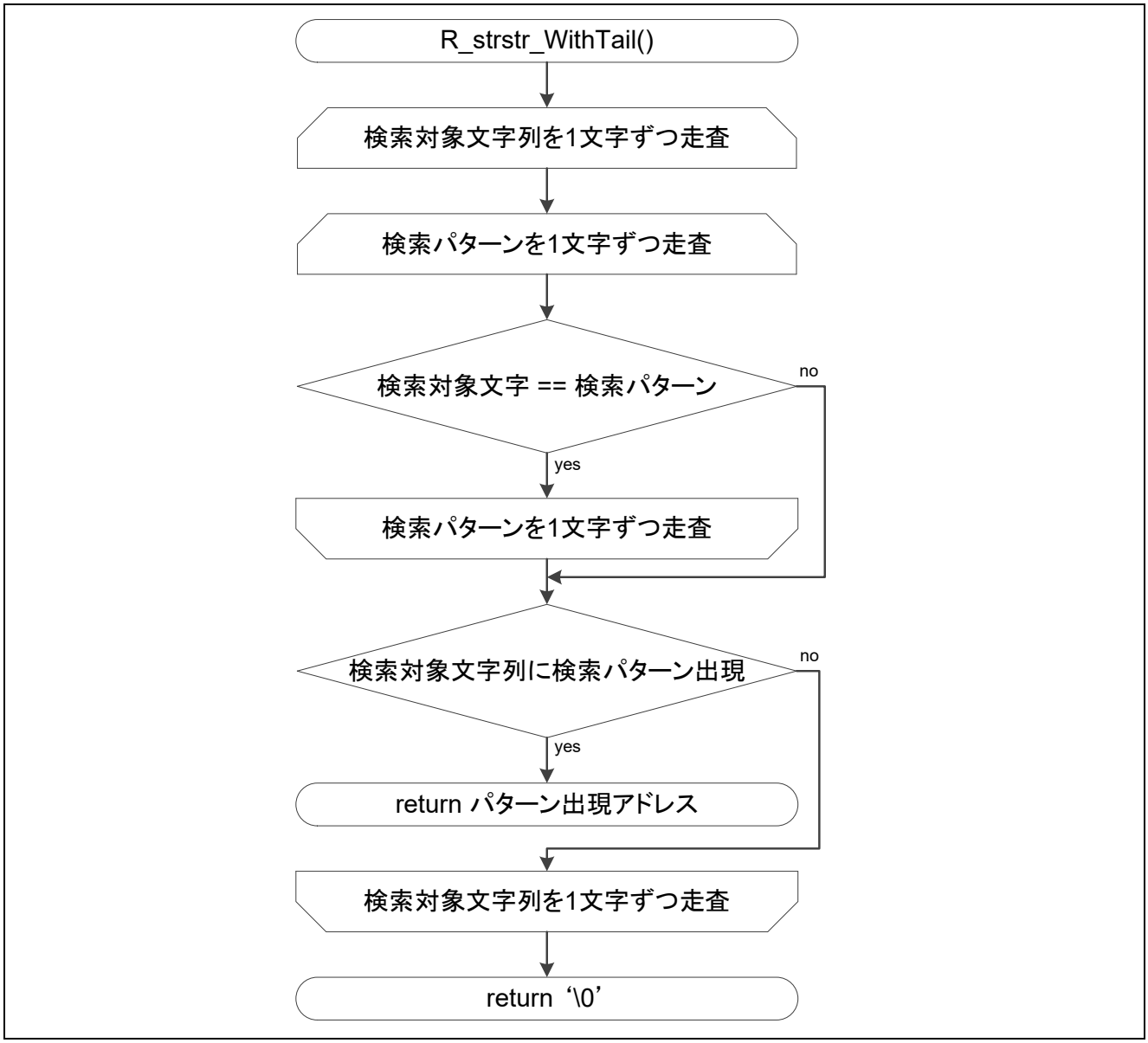


図 4.17　文字列中の任意のパターンを検索

4.10 ドライバ I/F 関数一覧

表 4.7 にドライバ I/F 関数一覧を示します。

表 4.7 ドライバ I/F 関数一覧

関数名	概要
R_ESP_UART_Send	UART送信開始
R_ESP_UART_Receive	UART受信開始
R_ESP_UART_GetRxCount	UART受信完了データ長取得
R_ESP_UART_Start	UART動作開始
R_ESP_UART_Stop	UART動作停止

4.11 ドライバ I/F 関数仕様

Wi-Fi モジュール制御用ソフトウェアのドライバ I/F 関数仕様を示します。UART0 のドライバを呼び出すように設定しています。SAU の構成に合わせて修正する必要があります。

4.11.1 UART 送信処理

[関数名] R_ESP_UART_Send		
概　要	UART 送信開始	
ヘッダ	r_esp_hw.h	
宣　言	void R_ESP_UART_Send(uint8_t * const tx_buf, uint16_t tx_num);	
説　明	• UART 送信処理を開始します。本サンプルプログラムの場合、R_UART0_Send()をコールします。	
引　数	uint8_t * const tx_buf	送信データ格納領域の先頭アドレス
	uint16_t tx_num	送信データ長
リターン値	なし	
備考	なし	

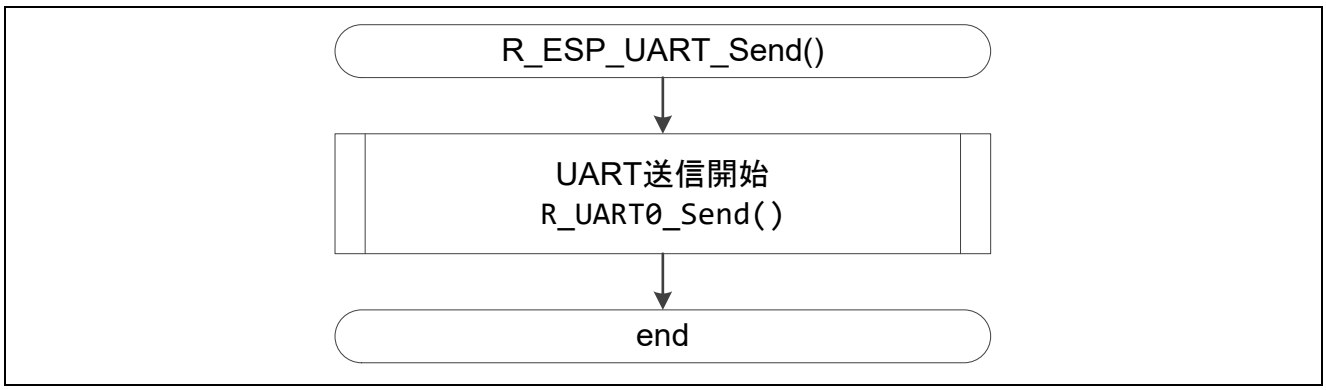


図 4.18 UART 送信開始

4.11.2 UART 受信開始

[関数名]	R_ESP_UART_Receive	
概　要	UART 受信開始	
ヘッダ	r_esp_hw.h	
宣　言	void R_ESP_UART_Receive(uint8_t * const rx_buf, uint16_t rx_num);	
説　明	• UART 受信処理を開始します。本サンプルプログラムの場合、R_UART0_Receive()をコールします。	
引　数	uint8_t * const rx_buf	受信データ格納領域の先頭アドレス
	uint16_t rx_num	受信データ長
リターン値	なし	
備考	なし	

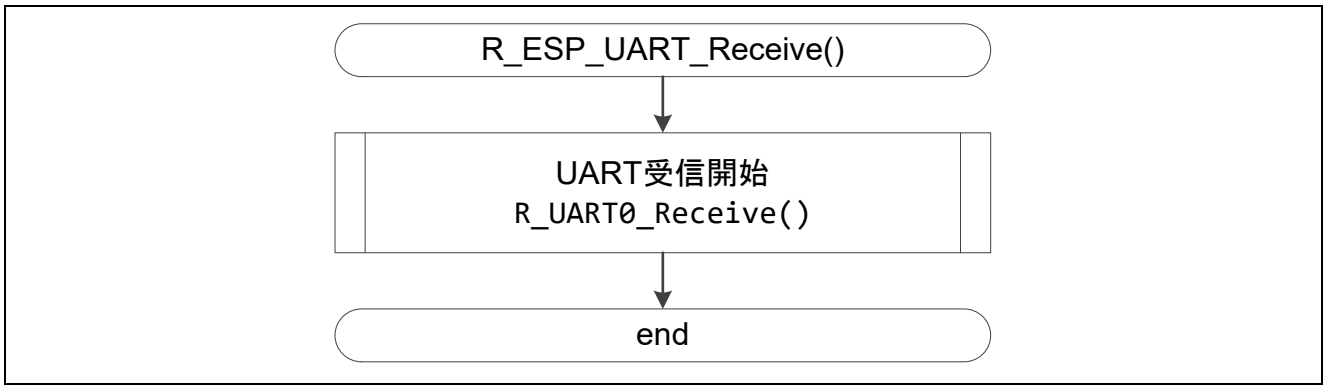


図 4.19 UART 受信開始

4.11.3 UART 受信完了データ長取得

[関数名]	R_ESP_UART_GetRxCount
概　要	UART 受信完了データ長取得
ヘッダ	r_esp_hw.h
宣　言	uint16_t R_ESP_UART_GetRxCount(void);
説　明	• UART 受信データ数カウンタ値を取得します。本サンプルプログラムの場合、g_uart0_rx_count を取得します。
引　数	なし
リターン値	UART 受信データ数カウンタ値
備考	なし

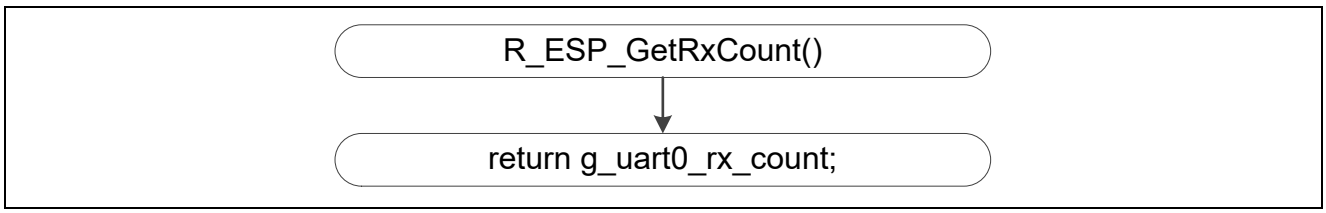


図 4.20 UART 受信完了データ長取得

4.11.4 UART 動作開始関数

[関数名]	R_ESP_UART_Start
概　要	UART 動作開始
ヘッダ	r_esp_hw.h
宣　言	uint16_t R_ESP_UART_Start(void);
説　明	• UART の動作を開始します。
引　数	なし
リターン値	なし
備考	なし

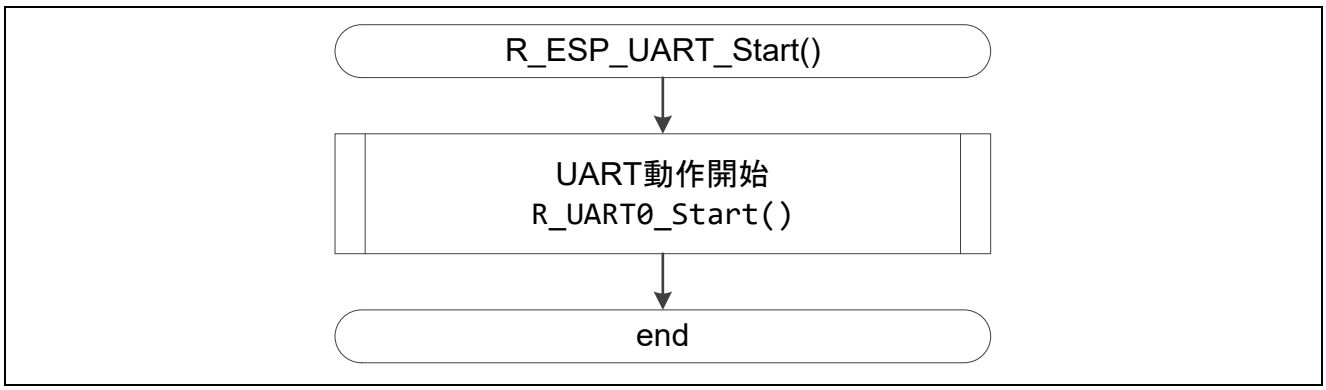


図 4.21 UART 動作開始関数

4.11.5 UART 動作停止関数

[関数名]	R_ESP_UART_Stop
概　要	UART 動作停止
ヘッダ	r_esp_hw.h
宣　言	uint16_t R_ESP_UART_Stop(void);
説　明	• UART の動作を停止します。
引　数	なし
リターン値	なし
備考	なし

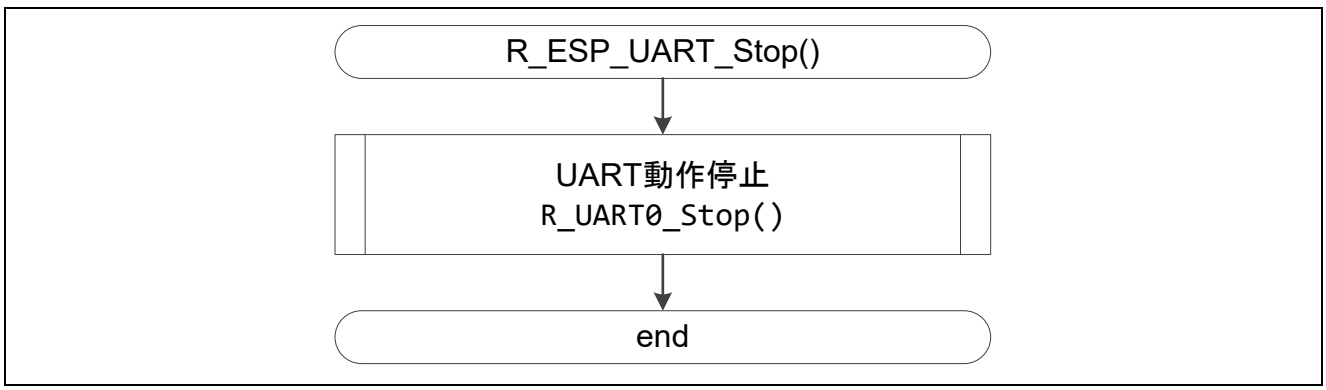


図 4.22 UART 動作停止関数

4.12 デバイスドライバ

4.12.1 コード生成ドライバ

本アプリケーションノートでは、コード生成ツール(CG)を使用してデバイスドライバを生成しています。CG の設定については、図 4.23、図 4.24、図 4.25 を参照してください。

図 4.23 SAU CG 設定(1/3)

図 4.24 SAU CG 設定(2/3)

図 4.25 SAU CG 設定(3/3)

5. 応用例

応用サンプルプログラムとして、testmain.c を sample フォルダに格納しています。（表 4.2 参照）

本応用サンプルプログラムは、本制御ソフトウェアの使用例として、アクセスポイントへのアクセスと、TCP サーバとのデータ送受信を行います。図 5.1 に応用サンプルプログラムのシステム構成例を示します。

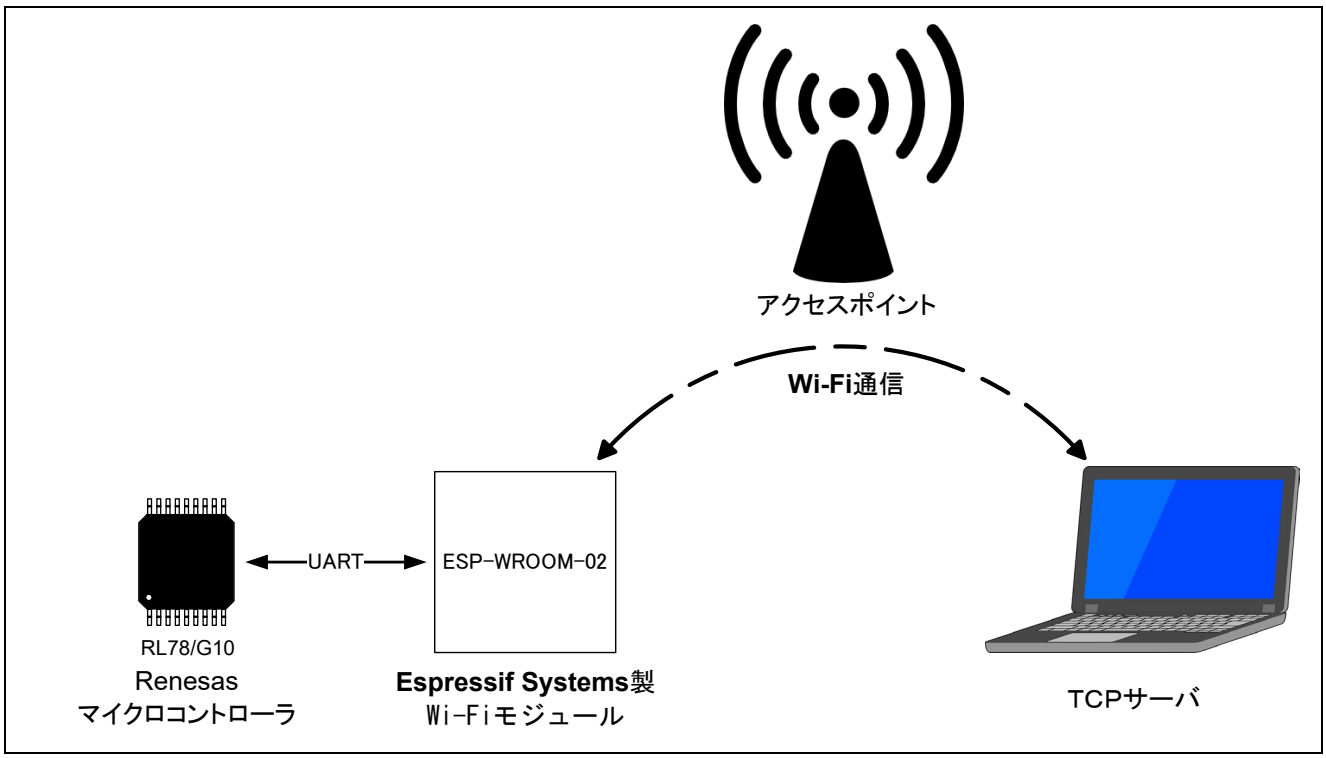


図 5.1 応用サンプルプログラム システム構成例

5.1 応用サンプルプログラムの概要

応用サンプルプログラムの概要を示します。

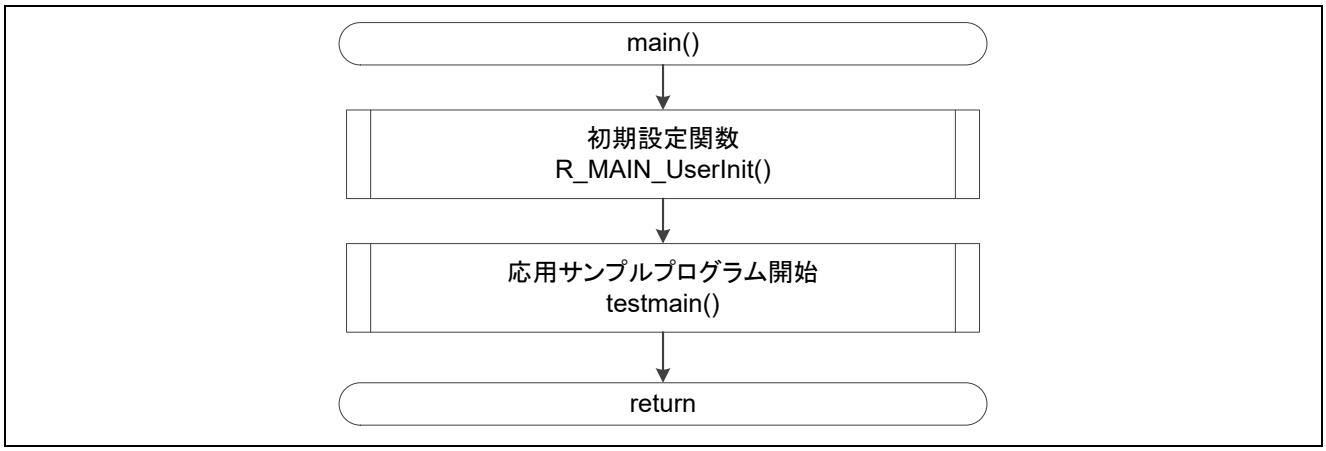
① RL78/G10 は、UART 経由で Wi-Fi モジュールの初期設定を行います。
② RL78/G10 は、Wi-Fi モジュールをアクセスポイントに接続させます。
③ RL78/G10 は、Wi-Fi モジュールから IP アドレスを取得します。
④ RL78/G10 は、Wi-Fi モジュールから MAC アドレスを取得します。
⑤ RL78/G10 は、Wi-Fi モジュールを TCP サーバに接続させます。
⑥ RL78/G10 は、データを TCP サーバへ送信します。
⑦ RL78/G10 は、TCP サーバからの返答を待ちます。
⑧ RL78/G10 は、受信データをバッファ（内蔵 RAM）に保存します。
⑨ RL78/G10 は、1 秒間ソフトウェアでウエイトします。
⑩ ⑥から⑨を繰り返します。

5.2 応用サンプルプログラムフローチャート

アプリケーションノートに同梱する応用サンプルプログラムのフローチャートを示します。

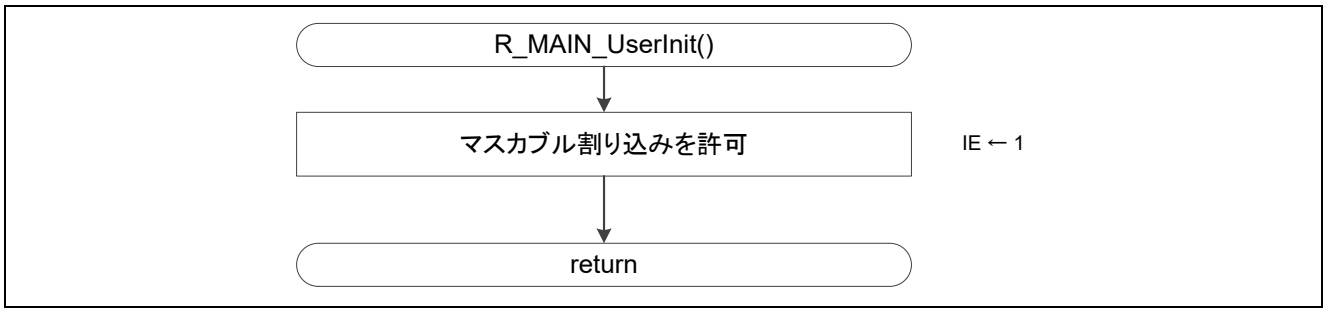
5.2.1 main 関数

[関数名]	main
概　要	main 関数
ヘッダ	なし
宣　言	void main(void);
説　明	• ユーザ初期設定を行った後、応用サンプルプログラム関数を呼びます。
引　数	なし
リターン値	なし
備考	なし



5.2.2 ユーザ初期設定関数

[関数名]	R_MAIN_UserInit
概　要	ユーザ初期設定関数
ヘッダ	なし
宣　言	static void R_MAIN_UserInit(void);
説　明	• マスカブル割り込みを許可します。
引　数	なし
リターン値	なし
備考	なし



5.2.3 応用サンプルプログラム関数

[関数名]	testmain
概　要	応用サンプルプログラム関数
ヘッダ	なし
宣　言	void testmain(void);
説　明	• Wi-Fi モジュールの初期設定から、TCP 送受信までを行うまでのサンプルプログラムです。
引　数	なし
リターン値	なし
備考	なし

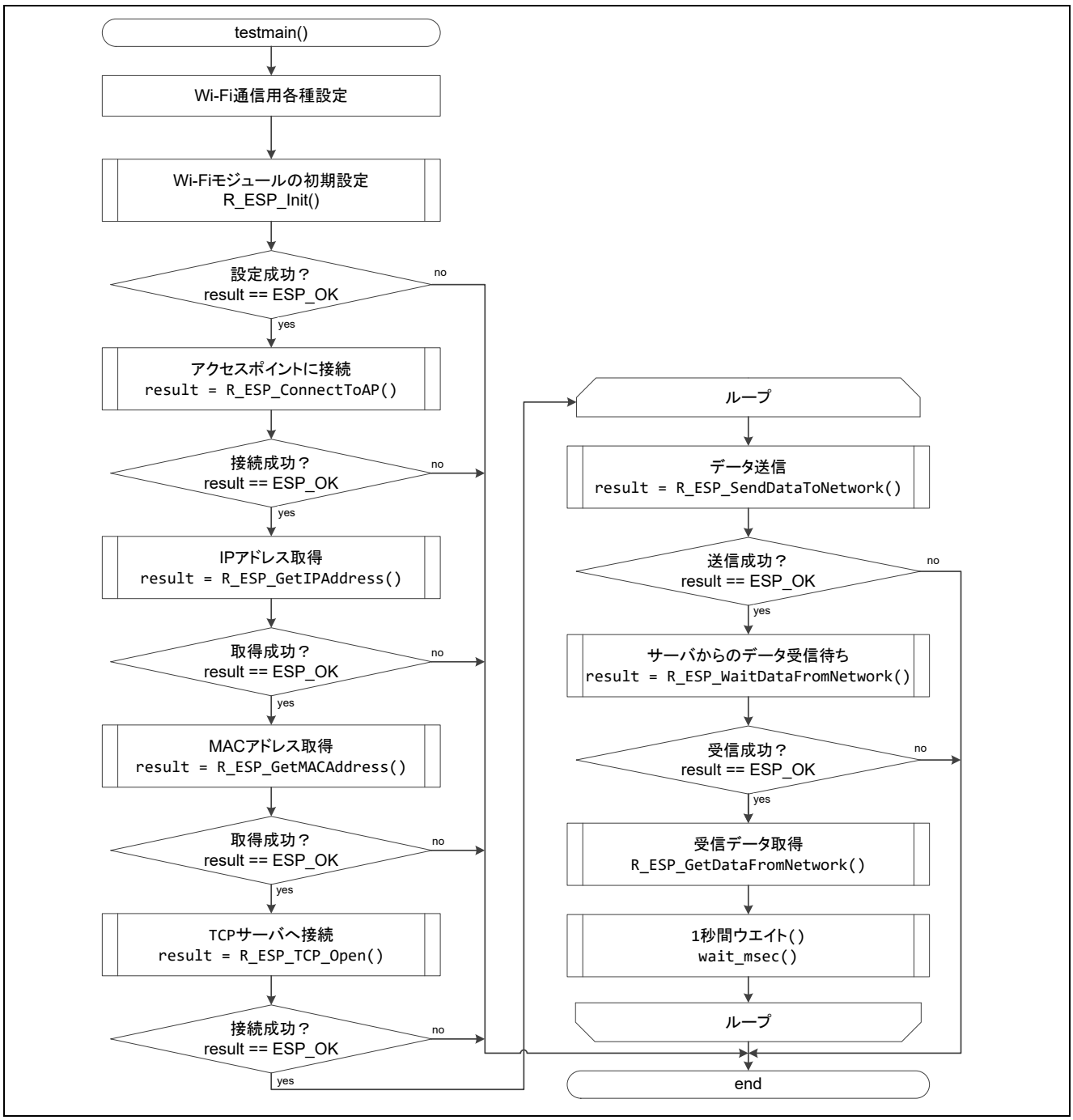


図 5.2　応用サンプルプログラムフローチャート

5.3 サンプルコードの修正

コード生成をやり直す場合は、以下のようにファイル及びプロジェクトの修正が必要になる場合があります。

対象環境 ：CS+版、e2studio 版

ファイル名：cstart.asm

修正箇所 ：stack area の記載場所

cstart.asm に生成される.DS 0x40 を任意のスタックサイズに変更する。

```
;-------------------------------------------------------------------------------
;   stack area
;-------------------------------------------------------------------------------
; !!! [CAUTION] !!!
; Set up stack size suitable for a project.
.SECTION .stack_bss, BSS
_stackend:
    .DS     0x40
_stacktop:
$ENDIF
```

↓

```
;-------------------------------------------------------------------------------
;   stack area
;-------------------------------------------------------------------------------
; !!! [CAUTION] !!!
; Set up stack size suitable for a project.
.SECTION .stack_bss, BSS
_stackend:
    .DS     0xC8
_stacktop:
$ENDIF
```

対象環境 ：IAR 版

修正箇所 ：スタックサイズの記載場所

プロジェクトのオプションから「一般オプション」→「スタック/ヒープ」→「デフォルトのオーバーライド」→「スタックサイズ(バイト)」に任意のスタックサイズを入力する。

6. サンプルコード

サンプルコードは、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。

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2019.07.04	—	初版発行
1.10	2022.06.24	6	動作確認条件を更新
		9	必要メモリサイズを更新
		10	ディレクトリ構成を更新
		37	サンプルコードの修正を追加

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. 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。