RL78/G14

オンボード LED 点滅制御 (Arduino API)

要旨

本アプリケーションノートでは、RL78/G14 Fast Prototyping Board (FPB) を用いて Arduino 言語のようなプログラム記述で FPB に搭載されている LED を点滅制御する方法を説明します。

対象デバイス

RL78/G14 Fast Prototyping Board

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
1.1 プログラム実行環境 4
1.2 プログラム (スケッチ) の構成 5
1.3 プロジェクトの起動準備 5
1.4 プログラム (スケッチ) の定義 6
1.5 初期設定処理 7
1.6 メイン処理部 7

2. 動作確認条件 8

3. 関連アプリケーションノート 8

4. ハードウェア説明 9
4.1 ハードウェア構成例 9
4.2 使用端子一覧 9

5. ソフトウェア説明 10
5.1 動作概要 10
5.2 定数一覧 11
5.3 変数一覧 11
5.4 関数一覧 12
5.5 関数仕様 13
5.6 フローチャート 15
5.6.1 初期設定関数 15
5.6.2 メイン処理関数 16
5.6.3 SW_USR チェック及び時間待ち関数 17

6. サンプルコード 18

7. 参考ドキュメント 18

1. 仕様

本アプリケーションノートでは、FPB を用いて Arduino 言語のようなプログラム記述で FPB に搭載された LED (LED0 と LED1) を点滅制御します。スイッチ (SW_USR) によって次の動作に変わるまでの時間 (ウエイト時間) を変更します。

電源投入後、LED0 と LED1 は点灯しています。最初に、本プログラムで LED0 と LED1 を消灯させます。次に、ウエイト時間が経過する毎に、LED0 を点灯、LED1 を点灯、LED0 を消灯させます。以降、つぎの No.1～ No.4 を繰り返します。

No.	動作	LED1	LED0
0	全点灯	ON	ON
1	全消灯	OFF	OFF
2	LED0: 点灯	OFF	ON
3	LED1: 点灯	ON	ON
4	LED0: 消灯	ON	OFF

ウエイト時間は、最初は 1 秒に設定されています。スイッチ (SW_USR) を押下する毎に、0.75 秒、0.5 秒、0.25 秒、1 秒に設定が変更されます。

表 1.1 に本プログラムで使用する周辺機能と用途を示します。

表 1.1 使用する周辺機能と用途

周辺機能	用途
デジタル入力	スイッチ (SW_USR) 状態の読み込み
デジタル出力	LED (LED0，LED1) の制御
タイマ・アレイ・ユニット(TAU)	ウエイト時間の計測

1.1 プログラム実行環境

本アプリケーションノートでは、RL78 ファミリ固有の開発環境上で、Arduino 言語のようなプログラムを実行させています。プログラム実行環境の概念図を図 1.1 に示します。

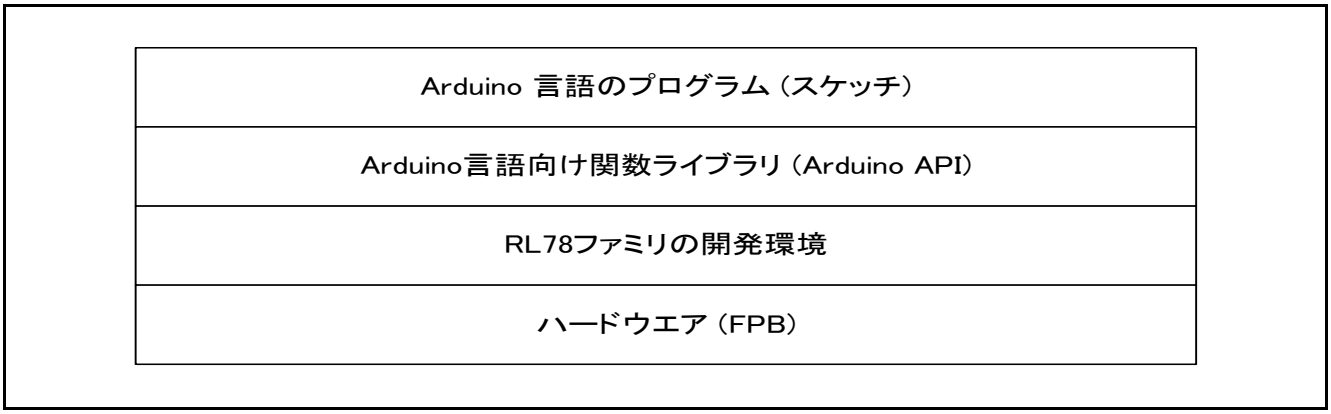

図 1.1 プログラム実行環境

本アプリケーションノートで使用可能なライブラリ関数を表 1.2 に示します。

表 1.2 ライブラリの機能

項目	ライブラリ関数	機能
デジタル入出力	pinMode(pin, mode)	pin で指定した端子の動作モード(入力モード / 出力モード / 内蔵プルアップ抵抗を有効にした入力モード) 指定
	digitalWrite (pin, value)	pin で指定した端子を value で指定した状態 (ハイレベル / ロウレベル) にする。
	digitalRead(pin)	pin で指定した端子状態を読み出す。
時間管理	millis()	プログラムの実行を開始した時から現在までの時間をミリ秒単位で返します。
	micros()	プログラムの実行を開始した時から現在までの時間をマイクロ秒単位で返します。
	delay (ms)	ミリ秒単位でプログラムを指定した時間だけ止めます。
	delayMicroseconds (us)	マイクロ秒単位でプログラムを指定した時間だけ止めます。

1.2 プログラム (スケッチ) の構成

プロジェクトが格納されているフォルダ (workspace) より下の階層には、統合開発環境ごとにサブフォルダが準備されています。それぞれの統合開発環境のフォルダには RL78 ファミリの開発環境関で使用するファイルが格納されています。

サブフォルダ sketch には、Arduino 言語のプログラム (スケッチ) である AR_SKETCH.c が格納されています。スケッチを参照もしくは変更する場合は、sketch の中の”AR_SKETCH.c”ファイルを使用します。

1.3 プロジェクトの起動準備

使用する統合開発環境ごとにプロジェクトの起動準備が異なります。詳細は、つぎのアプリケーションノートを参照してください。

RL78 ファミリ　Arduino API 導入ガイド　(R01AN5413)

1.4 プログラム (スケッチ) の定義

プログラム (スケッチ) の定義内容を**図 1.2** に示します。

```
/*------------------------------------------------------------------↓
     Definition area. Write pin definition here.↓
------------------------------------------------------------------*/↓
↓
1) int ledPin1 = 16;                // assign D16 pin to ledPin1 for LED0.↓
   int ledPin2 = 17;                // assign D17 pin to ledPin2 for LED1.↓
   int swPin = 18;                  // assign D18 pin to swPin for SW_USER.↓
↓
2) const int TIME_TABLE[]=↓         // interval definition↓
   {
       100, 75, 50, 25↓
   };↓
↓
3) int sw_flag = 0xFF;              // SW status flag↓
   int f_edge_count = 0;            // falling edge count↓
↓
   void wait(void);                 // delay with SW check↓
↓
   void setup(void);                // initialisation procedure↓
   void loop(void);                 // main function of sketch↓
```

図 1.2 プログラムの定義内容

最初に、使用するデジタル入出力端子を定義します。LED0 を制御する ledPin1 端子に 16 を指定して D16 に割り当てます。LED1 を制御する ledPin2 端子に 17 を指定して D17 に割り当てます。スイッチ (SW_USR) を制御する swPin 端子に 18 を指定して D18 に割り当てます。これらは、FPB に実装されている機能なので、変更はできません。

次に LED 制御の間隔を指定する配列 TIME_TABLE に 4 つの時間 (10 ミリ秒単位) を定義しています。

次にスイッチ (SW_USR) の状態を管理するための変数 sw_flag と f_edge_count を定義します。sw_flag は、約 10 ミリ秒間隔でスイッチ (SW_USR) の状態を格納します。f_edge_count は、スイッチ (SW_USR) が押された回数をカウント (値 : 0～3) します。f_edge_count の値で配列 TIME_TABLE の参照先を決定し、LED 制御の間隔を決定します。

1.5 初期設定処理

プログラム (スケッチ) の初期設定部分を図 1.3 に示します。

ここでは、setup 関数として、各端子の入出力を指定します。

```
/*----------------------------------------------------------------------↓
     Arduino setup area. Write setup program here.↓
----------------------------------------------------------------------*/↓
↓
void setup(void){↓
  // put your setup code here, to run once:↓
↓
    pinMode(ledPin1, OUTPUT);   // set D16pin to output mode↓
    pinMode(ledPin2, OUTPUT);   // set D17pin to output mode↓
    pinMode(swPin, INPUT);      // set D18pin to input mode↓
}↓
```

図 1.3 初期設定処理部分

1.6 メイン処理部

繰り返し実行されるメイン処理の先頭部分を図 1.4 に示します。「プロジェクトの起動準備」が正しく設定されていると、起動画面は図 1.4 となります。

```
/*----------------------------------------------------------------------↓
     Arduino loop area. Write the program in loop function here.↓
----------------------------------------------------------------------*/↓
↓
void loop(void){↓
  // put your main code here, to run repeatedly:↓
↓
↓
    digitalWrite(ledPin1, HIGH);    // output HIGH to ledPin1 (turn off LED0)
    digitalWrite(ledPin2, HIGH);    // output HIGH to ledPin2 (turn off LED1)
```

図 1.4 メイン処理の先頭部分

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 動作確認条件

項目	内容
使用マイコン	RL78/G14 (R5F104MLAFB)
動作周波数	● 高速オンチップ・オシレータ・クロック (f_{IH}) : 32MHz ● CPU／周辺ハードウェア・クロック : 32MHz
動作電圧	3.3V (2.75V～5.5V で動作可能) LVD 動作 : リセット・モード LVD 検出電圧 (V_{LVD}) 立ち上がり時 TYP. 2.81V (2.76V～2.87V) 立ち下がり時 TYP. 2.75V (2.70V～2.81V)
統合開発環境	ルネサス エレクトロニクス CS+ for CC V8.03.00 ルネサス エレクトロニクス e² studio V7.7.0 IAR Systems IAR Embedded Workbench for RL78
C コンパイラ	ルネサス エレクトロニクス CC-RL V1.09.00 IAR Systems RL78 用 IAR C/C++ コンパイラ v4.20.1

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 ファミリ Arduino API 導入ガイド (R01AN5413)

RL78/G14 7 セグメント LED 点灯制御 (Arduino API) (R01AN5387)

4. ハードウェア説明

4.1 ハードウェア構成例

本アプリケーションノートで使用するハードウェア (FPB) を図 4.1 に示します。

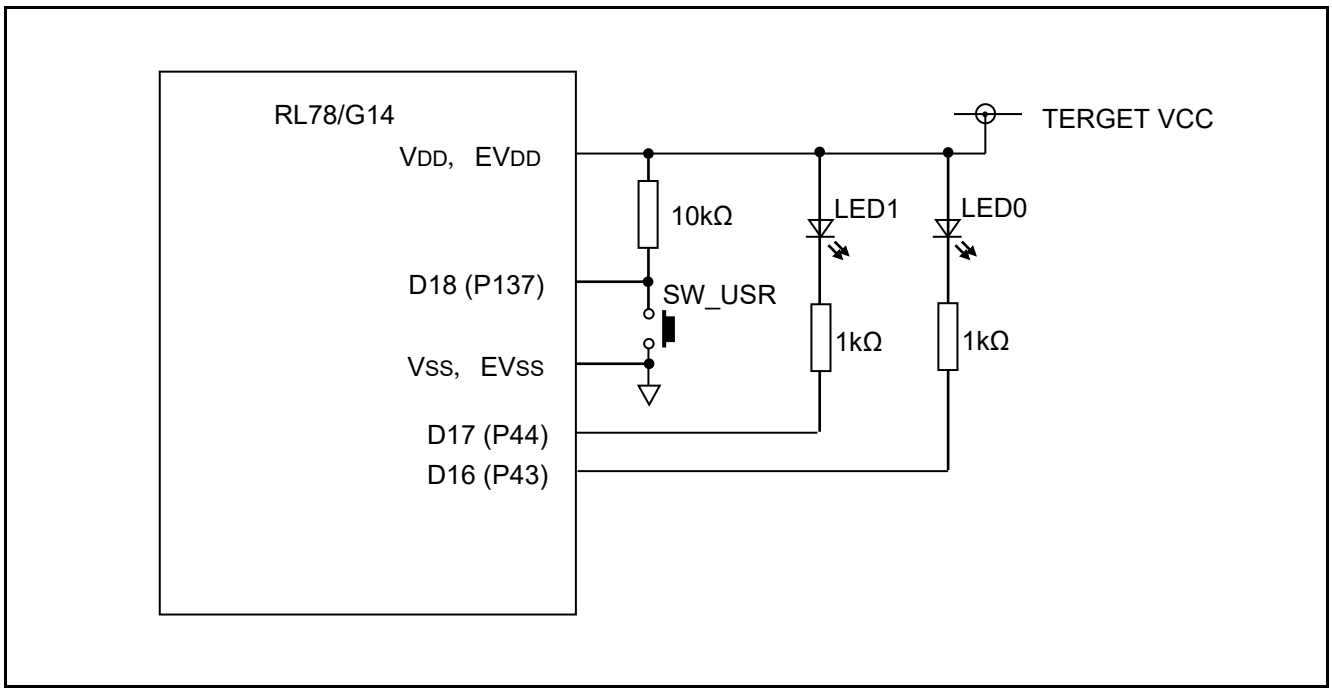

図 4.1 ハードウェア構成例

注意 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。
電源電圧は USB から 3.3V を供給しています。

4.2 使用端子一覧

使用端子と機能を表 4.1 に示します。

表 4.1 使用端子と機能

端子	ポート名	入出力	機能
D16	P43	出力	LED0 を制御
D17	P44	出力	LED1 を制御
D18	P137	入力	スイッチ (SW_USR) 入力

5. ソフトウェア説明

5.1 動作概要

本アプリケーションノートでは、初期設定 (端子の設定) が完了して、メイン処理 (loop) が起動すると、FPB に実装されている LED0 と LED1 の点灯状態が 1 秒毎に変化します。SW_USR スイッチを押下すると、点灯状態の変化タイミング (ウエイト時間) が 0.75 秒、0.5 秒、0.25 秒、1 秒と変化していきます。

下記①～③に詳細を記載します。

① setup 関数で使用する端子の設定を行います。

- LED0 のドライブ用端子 (ledPin1) をデジタル出力に設定します。
- LED1 のドライブ用端子 (ledPin2) をデジタル出力に設定します。
- SW_USR スイッチの読み取り用端子 (swPin) をデジタル入力に設定します。

② loop 関数でメイン処理を行います。

- ledPin1 に HIGH を設定して、LED0 を消灯します。
- ledPin2 に HIGH を設定して、LED1 を消灯します。
- poll_sw 関数で、SW_USR スイッチの状態を確認し、ウエイト時間の経過を待ちます。
- ledPin1 に LOW を設定して、LED0 を点灯します。
- poll_sw 関数で、SW_USR スイッチの状態を確認し、ウエイト時間の経過を待ちます。
- ledPin2 に LOW を設定して、LED1 を点灯します。
- poll_sw 関数で、SW_USR スイッチの状態を確認し、ウエイト時間の経過を待ちます。
- ledPin1 に HIGH を設定して、LED0 を消灯します。
- poll_sw 関数で、SW_USR スイッチの状態を確認し、ウエイト時間の経過を待ちます。

③ poll_sw 関数で、SW_USR スイッチの状態を 10m ミリ秒ごとに確認し、押された回数をカウントします。

- SW_USR スイッチの状態を読み込みます。
- 10 ミリ秒待ちます。 (チャタリング対策)
- SW_USR スイッチが押されたかをチェックします。
- SW_USR スイッチが押されていたら、フラグ (e_flag) をセットします。
- 配列 TIME_TABLE で指定された回数、先頭から処理を繰り返します。
- 指定された回数の処理が完了したら、フラグ (e_flag) をチェックします。
- フラグ (e_flag) がセットされていたら、押下回数をカウントします。

5.2 定数一覧

表 5.1 にサンプルコードで使用する定数を示します。

表 5.1 サンプルコードで使用する定数

定数名	設定値	内容
ledPin1	16	LED0 を制御する端子の番号
ledPin2	17	LED1 を制御する端子の番号
swPin	18	SW_USR を読む端子の番号
TIME_TABLE	100,75,50,25	ウエイト時間 (10 ミリ秒単位)

5.3 変数一覧

表 5.2 にグローバル変数を示します。

表 5.2 グローバル変数

型	変数名	内容	使用関数
Int	sw_flag	SW_USR の状態変化を格納する。	poll_sw()
Int	f_edge_count	SW_USR が押された回数をカウントする。	poll_sw()

5.4 関数一覧

表 5.3 に関数一覧を示します。

表 5.3 関数一覧

関数名	概要
loop	メイン処理 (スケッチ)
setup	初期化関数 (スケッチ)
poll_sw	SW_USR の状態確認とウエイト処理 (10 ミリ秒単位)
pinMode	端子の動作モード (入力モード / 出力モード / 内蔵プルアップ抵抗を有効にした入力モード) 指定
digitalWrite	端子へのデータ出力
digitalRead	端子状態を読み出す。
micros	プログラムの実行を開始した時から現在までの時間をマイクロ秒単位で返します。
millis	プログラムの実行を開始した時から現在までの時間をミリ秒単位で返します。
delay	ミリ秒単位でプログラムを指定した時間だけ止めます。
delayMicroseconds	マイクロ秒単位でプログラムを指定した時間だけ止めます。

5.5 関数仕様

サンプルコードの関数仕様を示します。

[関数名] loop	
概　要	メイン関数
ヘッダ	AR_LIB_PORT.h、AR_LIB_TIME.h、r_cg_macrodriver.h、AR_SKETCH.h、r_cg_userdefine.h
宣　言	void loop(void)
説　明	スタートすると、指定された時間間隔で、LED0 と LED1 を全消灯→LED0 点灯→LED1 点灯→LED1 消灯を行う。
引　数	なし
リターン値	なし

[関数名] setup	
概　要	初期化関数
ヘッダ	AR_LIB_PORT.h、r_cg_macrodriver.h、r_cg_userdefine.h
宣　言	void setup(void);
説　明	プログラム (スケッチ) で使用する端子の設定を行う。
引　数	なし
リターン値	なし

[関数名] poll_sw	
概　要	SW_USR チェック及び時間待ち関数
ヘッダ	AR_LIB_PORT.h、AR_LIB_TIME.h、r_cg_macrodriver.h、r_cg_userdefine.h
宣　言	void poll_sw(void);
説　明	10 ミリ秒毎に SW_USR の状態を確認する。この動作を指定回数分繰り返して SW_USR の押下回数をカウントする。
引　数	なし
リターン値	なし

[関数名] pinMode		
概　要	端子機能設定関数	
ヘッダ	AR_LIB_PORT.h、r_cg_macrodriver.h、r_cg_userdefine.h	
宣　言	void pinMode(uint8_t pin,uint8_t mode)	
説　明	第 1 引数で示された端子を第 2 引数で示されたモードに設定する。	
引　数	uint8_t pin	: 指定する端子の番号
	uint8_t mode	: OUTPUT/INPUT/INPUT_PULLUP で端子のモードを指定
リターン値	なし	

[関数名]	digitalWrite	
概　要	端子へのデジタルデータ出力関数	
ヘッダ	AR_LIB_PORT.h、r_cg_macrodriver.h、r_cg_userdefine.h	
宣　言	void digitalWrite(uint8_t pin, uint8_t value);	
説　明	第 1 引数で示す端子に、第 2 引数で示すデータを出力する	
引　数	uint8_t pin	:出力する端子の番号
	uint8_t value	:出力するデータ (HIGH/LOW)
リターン値	なし	

[関数名]	digitalRead	
概　要	端子からのデジタル データの読出し関数	
ヘッダ	AR_LIB_PORT.h、r_cg_macrodriver.h、r_cg_userdefine.h	
宣　言	uint8_t digitalRead(uint8_t pin);	
説　明	引数で指定された端子の状態を読み出す。	
引　数	uint8_t pin	:読み出す端子の番号
リターン値	uint8_t	:読み出したデータ (HIGH/LOW)

[関数名]	delay	
概　要	ミリ秒単位でのウエイト関数	
ヘッダ	AR_LIB_TIME.h、r_cg_macrodriver.h、r_cg_userdefine.h	
宣　言	void delay(uint32_t time);	
説　明	引数で指定したミリ秒単位の時間を待つ。	
引　数	uint32_t time	:ウエイト時間 (ミリ秒単位)
リターン値	なし	

5.6 フローチャート

5.6.1 初期設定関数

図 5.1 に初期設定のフローを示します。

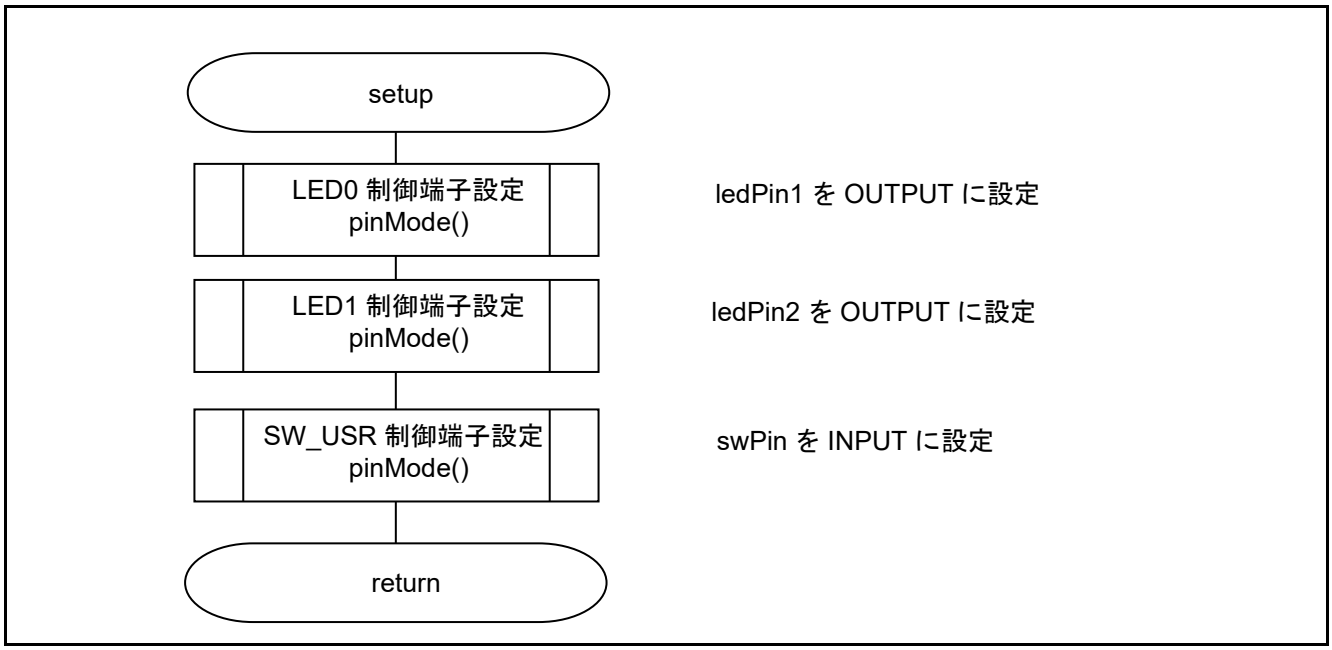

図 5.1 初期設定関数

5.6.2 メイン処理関数

図 5.2 にメイン処理関数のフローチャートを示します。

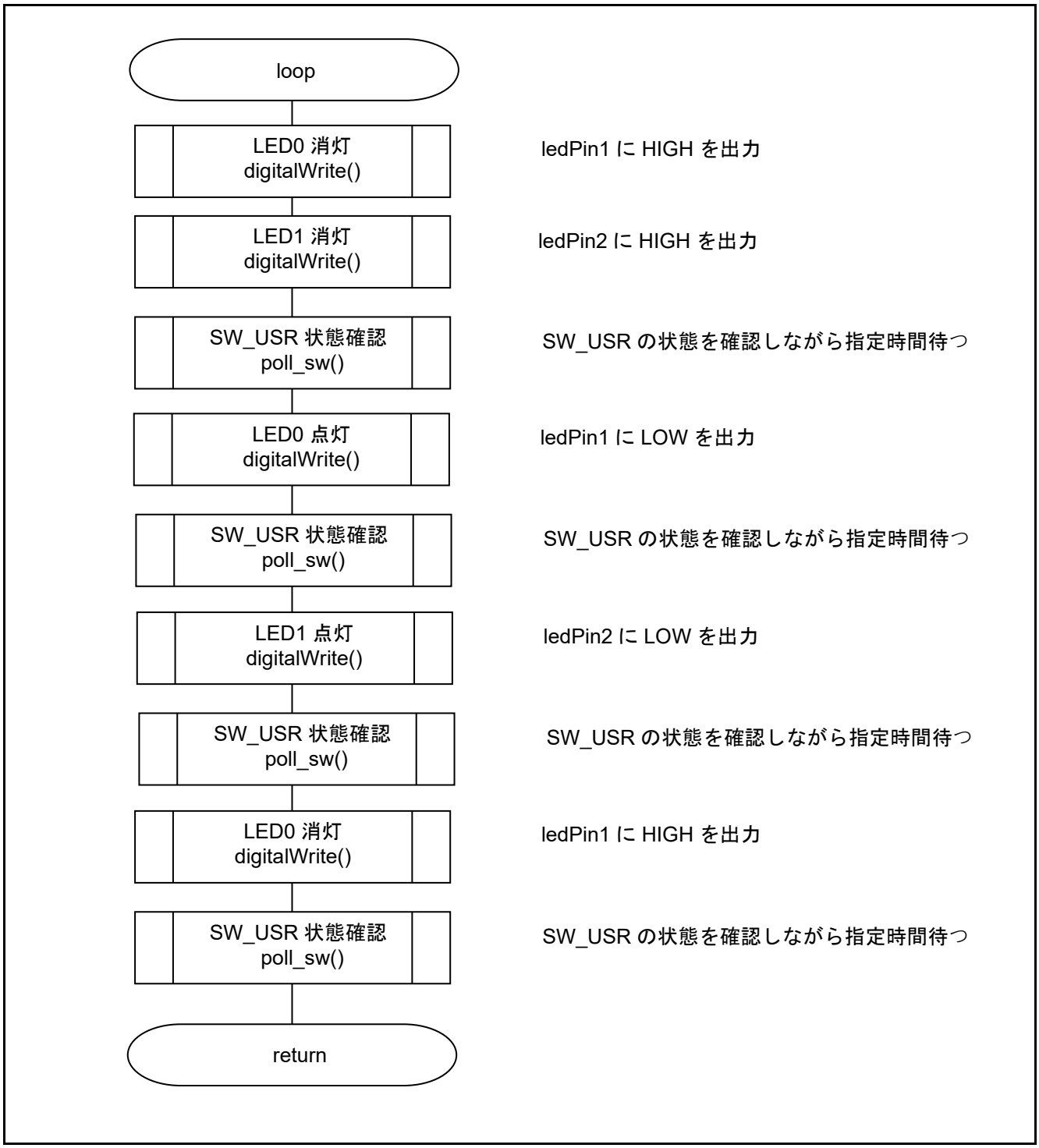

図 5.2 メイン関数

5.6.3 SW_USR チェック及び時間待ち関数

図 5.3 に SW_USR チェック及び時間待ち関数のフローチャートを示します。

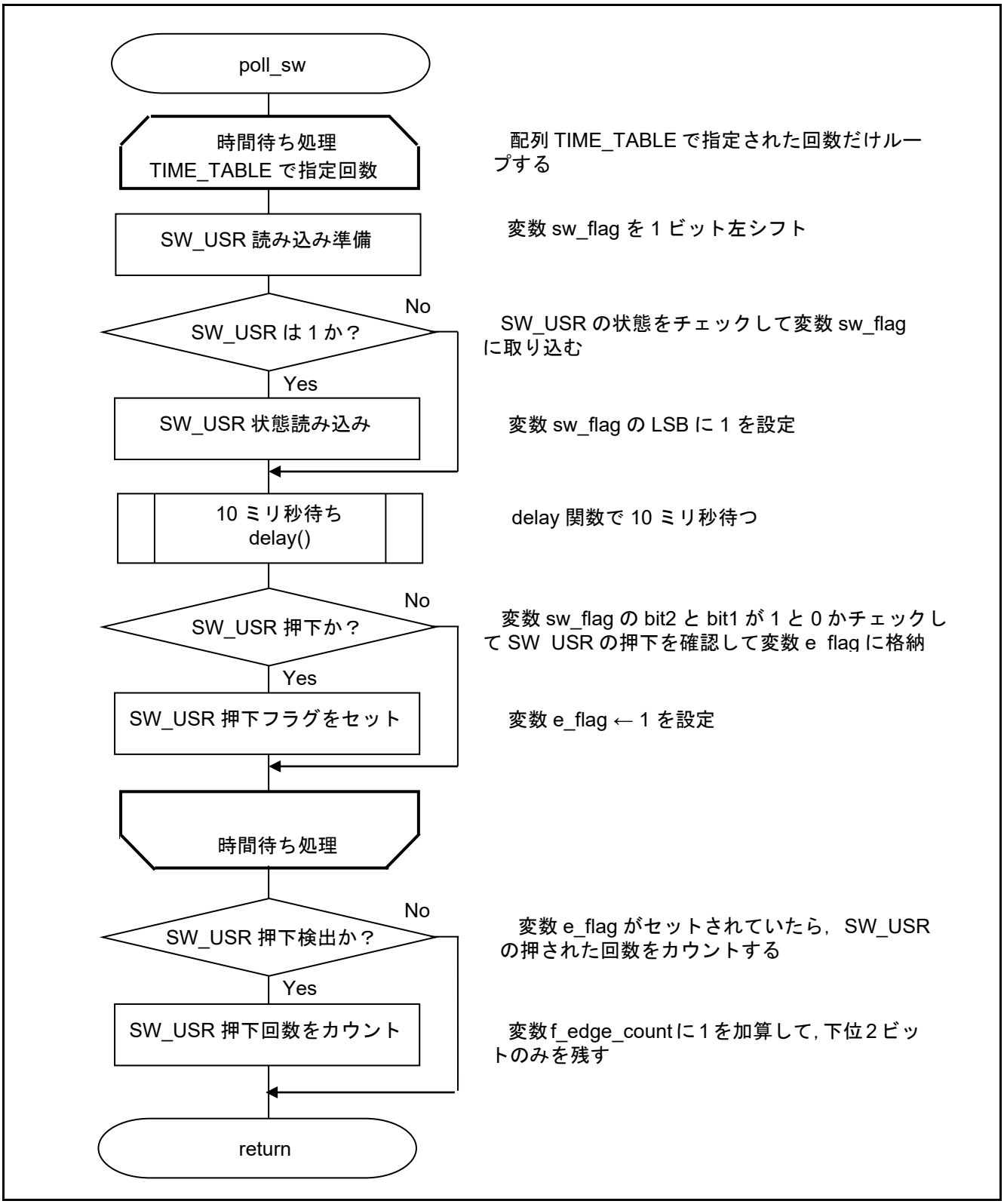

図 5.3 SW_USR チェック及び時間待ち関数

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。

7. 参考ドキュメント

RL78/G14 ユーザーズマニュアル ハードウェア編 (R01UH0186)
RL78 ファミリ ユーザーズマニュアル ソフトウェア編 (R01US0015)
RL78/G14 Fast Prototyping Board ユーザーズマニュアル (R20UT4573)
RL78 ファミリ Arduino API 導入ガイド (R01AN5413)
(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。)

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/

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。

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2020.6.30	—	初版発行

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。

© 2020 Renesas Electronics Corporation. All rights reserved.

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)

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。

© 2020 Renesas Electronics Corporation. All rights reserved.