

## RL78/G14

### Onboard LED Flashing Control (Arduino API)

---

#### Introduction

This application note describes a method for controlling flashing of LEDs mounted on an RL78/G14 Fast Prototyping Board (FPB) using a program written in an Arduino language using the FPB.

#### Target Device

RL78/G14 Fast Prototyping Board

When applying the sample program covered in this application note to another microcomputer, modify the program according to the specifications for the target microcomputer and conduct an extensive evaluation of the modified program.

## Contents

1. Specifications .....	3
1.1 Program Execution Environment.....	4
1.2 Program (Sketch) Configuration.....	5
1.3 Preparations for Project Startup .....	5
1.4 Definitions in the Program (sketch) .....	6
1.5 Initial Setting Processing .....	7
1.6 Main Processing Part .....	7
2. Operating Conditions .....	8
3. Related Application Notes.....	8
4. Hardware .....	9
4.1 Example of Hardware Configuration .....	9
4.2 List of Pins Used.....	9
5. Software .....	10
5.1 Summary of Operation .....	10
(1) Pins to be used are set in the setup function. ....	10
(2) The main processing is performed in the loop function. ....	10
(3) In the poll_sw function, the state of the SW_USR switch is confirmed every 10 milliseconds, and the number of times the switch is pressed is counted. ....	10
5.2 List of Constants .....	11
5.3 List of Variables .....	11
5.4 List of Functions .....	12
5.5 Specification of Functions .....	13
5.6 Flowcharts .....	15
5.6.1 Initial setting function.....	15
5.6.2 Main Processing Function .....	16
5.6.3 SW_USR check and wait time function.....	17
6. Sample Code.....	18
7. Reference Documents .....	18
Revision History .....	19

## 1. Specifications

In this application note, the flashing of LEDs (LED0 and LED1) mounted on an FPB is controlled by a program written in an Arduino language using the FPB. The time (wait time) until the change to the next operation is modified using a switch (SW\_USR).

After turning on power, LED0 and LED1 are lit. Initially, this program extinguishes LED0 and LED1. Next, each time the wait time elapses, LED0 is lit, LED1 is lit, and LED0 is extinguished. Thereafter, operations Nos. 1 to 4 shown below are repeated.

No.	Operation	LED1	LED0
0	Both lit	ON	ON
1	Both extinguished	OFF	OFF
2	LED0: lit	OFF	ON
3	LED1: lit	ON	ON
4	LED0: lit	ON	OFF

The wait time is initially set to 1 second. Each time the switch (SW\_USR) is pressed the setting is modified, to 0.75 seconds, 0.5 seconds, 0.25 seconds, and then to 1 second.

Table 1.1 shows peripheral functions used in this program and their uses.

Table 1.1 Peripheral functions used and their uses

Peripheral Function	Use
Digital input	Reading the state of the switch (SW_USR)
Digital output	Control of the LEDs (LED0, LED1)
Timer Array Unit (TAU)	Measurement of the wait time

## 1.1 Program Execution Environment

In this application note, a program in an Arduino language is executed in a development environment specific to the RL78 family. A conceptual diagram of the program execution environment is shown in Figure 1.1.

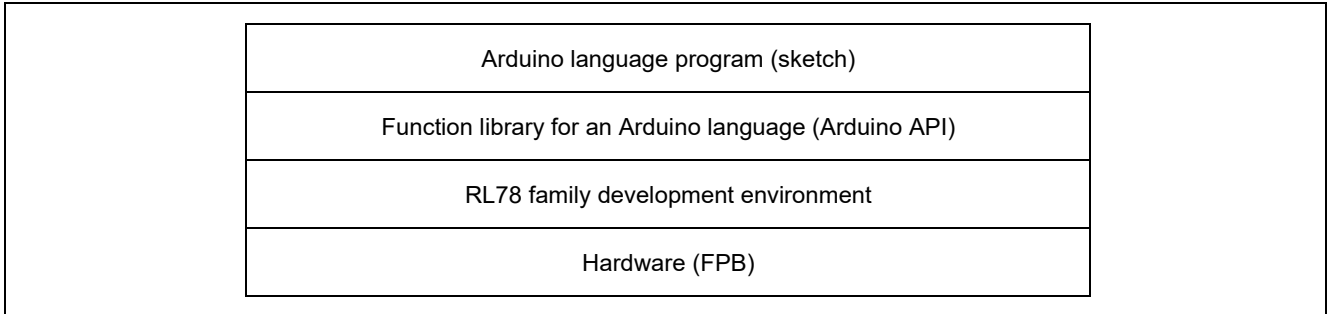


Figure 1.1 Program execution environment

Library functions that can be used in this application note are shown in Table 1.2.

Table 1.2 Library functions

Item	Library Function	Function
Digital I/O	pinMode(pin, mode)	Specifies the operation mode (input mode/output mode/input mode with internal pull-up resistor enabled) for the pin specified by pin.
	digitalWrite (pin, value)	Sets the pin specified by pin to the state specified by value (high level/low level).
	digitalRead(pin)	Reads out the state of the pin specified by pin
Time control	millis()	Returns, in millisecond units, the time from the start of program execution to the present time.
	micros()	Returns, in microsecond units, the time from the start of program execution to the present time.
	delay (ms)	Stops the program for the specified time in millisecond units.
	delayMicroseconds (us)	Stops the program for the specified time in microsecond units.

## 1.2 Program (Sketch) Configuration

Subfolders are prepared for each integrated development environment below the folder (workspace) in which the project is stored. In the folders for each of the integrated development environments the files are stored that are used in the RL78 family development environment.

In each sketch subfolder, AR\_SKETCH.c is stored which is the Arduino language program (sketch). When viewing or modifying sketch, the "AR\_SKETCH.c" file in the sketch subfolder is used.

## 1.3 Preparations for Project Startup

Preparations for project startup are different depending on the integrated development environment used. For details, refer to the following application note.

RL78 Family Arduino API Introduction Guide (R01AN5413)

## 1.4 Definitions in the Program (sketch)

Definitions in the program (sketch) are indicated in Figure 1.2.

```

/*-----↓
  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 ↓

```

Figure 1.2 Program definition details

First, digital input and output pins to be used are defined. The ledPin1 pin, which controls LED0, is assigned the value 16 and allocated to D16. The ledPin2 pin, which controls LED1, is assigned 17 and allocated to D17. The swPin pin, which controls the switch (SW\_USR), is assigned 18 and allocated to D18. These functions are installed on the FPB, and so cannot be changed.

Next, four times (in 10 millisecond units) are defined in the array TIME\_TABLE, which specifies the intervals in LED control.

Next, the variables sw\_flag and f\_edge\_count used to manage the state of the switch (SW\_USR) are defined. sw\_flag stores the state of the switch (SW\_USR) at intervals of approximately 10 milliseconds. f\_edge\_count counts (using values from 0 to 3) the number of times the switch (SW\_USR) has been pressed. The value of f\_edge\_count determines the value to reference in the array TIME\_TABLE, to determine the interval of LED control.

## 1.5 Initial Setting Processing

The initial settings section of the program (sketch) is shown in Figure 1.3.

In the setup function, input or output mode is specified for each pin.

```

/*-----↓
  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↓
}↓
↓

```

Figure 1.3 Initial setting processing section

## 1.6 Main Processing Part

The leading section of the main processing, which is executed repeatedly, is shown in Figure 1.4. When preparations for project startup have been set correctly, the startup screen is as in Figure 1.4.

```

/*-----↓
  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)
  ↓
  ↓
}↓
↓

```

Figure 1.4 Leading section of main processing

## 2. Operating Conditions

The operation of the sample code provided with this application note has been tested under the following conditions.

Table 2.1 Operating conditions

Item	Description
Microcontroller used	RL78/G14 (R5F104MLAFB)
Operating frequency	<ul style="list-style-type: none"> <li>● High-speed on-chip oscillator clock (<math>f_{IH}</math>): 32 MHz</li> <li>● CPU/peripheral hardware clock: 32 MHz</li> </ul>
Operating voltage	3.3V (can be operated at 2.75 V to 5.5 V) LVD operation: Reset mode LVD detection voltage ( $V_{LVD}$ ) At rising edge: 2.81 V typ. (2.76 V to 2.87 V) At falling edge: 2.75 V typ. (2.70 V to 2.81 V)
Integrated development environment	Renesas Electronics CS+ for CC V8.03.00 Renesas Electronics e <sup>2</sup> studio V7.7.0 IAR Systems IAR Embedded Workbench for RL78
C compiler	Renesas Electronics CC-RL V1.09.00 IAR Systems IAR C/C++ Compiler v4.20.1 for RL78

## 3. Related Application Notes

The application notes related to this application note are shown below.

Refer to these together with this application note.

RL78 Family Arduino API Introduction Guide (R01AN5413)

RL78/G14 7-Segment LED Lighting Control (Arduino API) (R01AN5387)



## 4. Hardware

### 4.1 Example of Hardware Configuration

Figure 4.1 shows the hardware (FPB) that is used in this application note.

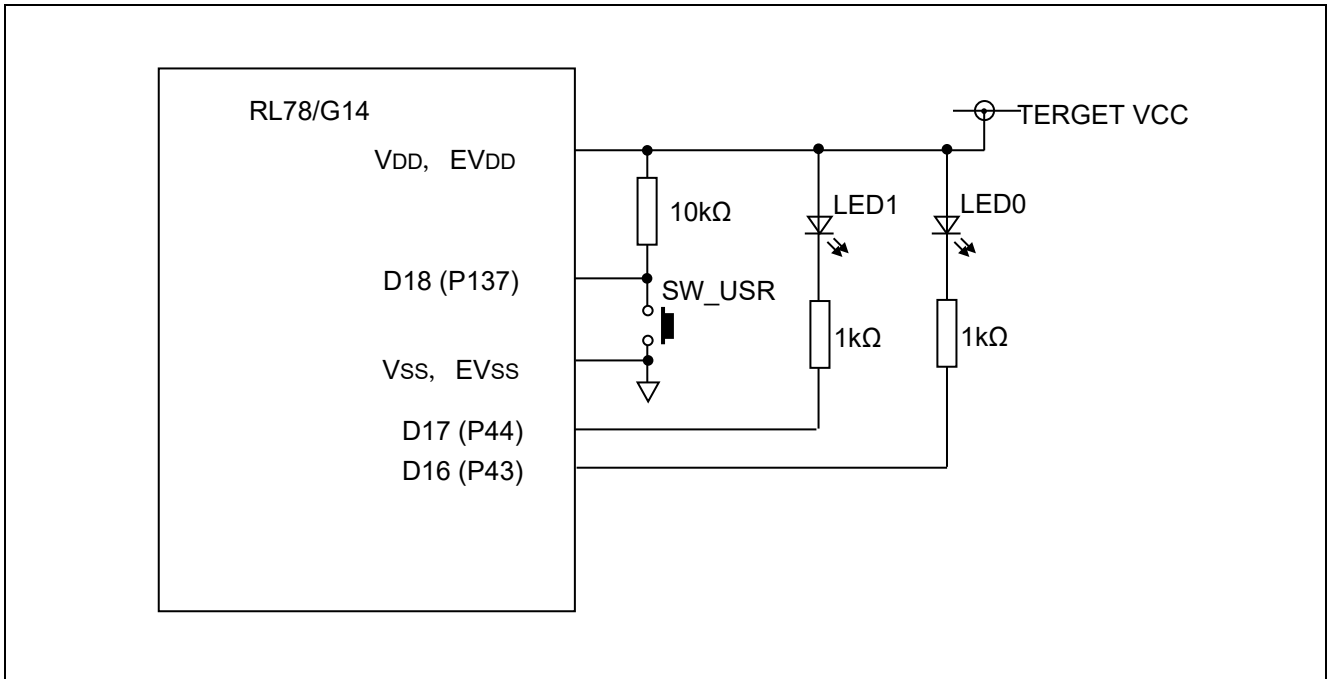


Figure 4.1 Hardware configuration example

Note: This conceptual diagram is simplified in order to summarize the connections.  
As the power supply voltage, 3.3 V is supplied via USB.

### 4.2 List of Pins Used

Table 4.1 shows the pins used and their functions.

Table 4.1 Pins used and their functions

Pin	Port Name	I/O	Function
D16	<b>P43</b>	Output	Control of LED0
D17	<b>P44</b>	Output	Control of LED1
D18	<b>P137</b>	Input	Switch (SW_USR) input

## 5. Software

### 5.1 Summary of Operation

In this application note, when the initial settings (pin settings) are completed and the main processing (loop) is started, the lit states of the LED0 and LED1 mounted on the FPB change every 1 second. Upon pressing the SW\_USR switch, the timing with which the lit states change is modified to 0.75 second, and then to 0.5 second, 0.25 second, and then 1 second.

Details are explained in (1) to (3) below.

(1) Pins to be used are set in the setup function.

- The drive pin for LED0 (ledPin1) is set for digital output.
- The drive pin for LED1 (ledPin2) is set for digital output.
- The read pin for the SW\_USR switch (swPin) is set for digital input.

(2) The main processing is performed in the loop function.

- ledPin1 is set to HIGH to extinguish LED0.
- ledPin2 is set to HIGH to extinguish LED1.
- The poll\_sw function confirms the state of the SW\_USR switch, and waits for the wait time to elapse.
- ledPin1 is set to LOW to light LED0.
- The poll\_sw function confirms the state of the SW\_USR switch, and waits for the wait time to elapse.
- ledPin2 is set to LOW to light LED1
- The poll\_sw function confirms the state of the SW\_USR switch, and waits for the wait time to elapse.
- ledPin1 is set to HIGH to extinguish LED0.
- The poll\_sw function confirms the state of the SW\_USR switch, and waits for the wait time to elapse.

(3) In the poll\_sw function, the state of the SW\_USR switch is confirmed every 10 milliseconds, and the number of times the switch is pressed is counted.

- The state of the SW\_USR switch is read in.
- Processing waits for 10 milliseconds. (Chattering countermeasure)
- A check as to whether the SW\_USR switch has been pressed is performed.
- If the SW\_USR switch has been pressed, a flag (e\_flag) is set.
- Processing from the beginning is repeated the number of times specified in the TIME\_TABLE array.
- When processing for the specified number of times has been completed, the flag (e\_flag) is checked.
- If the flag (e\_flag) is set, the number of presses is counted.

## 5.2 List of Constants

Table 5.1 shows constants that are used in the sample code.

Table 5.1 Constants used in sample code

Constant Name	Setting Value	Description
ledPin1	16	Number of the pin that controls LED0
ledPin2	17	Number of the pin that controls LED1
swPin	18	Number of the pin that reads SW_USR
TIME_TABLE	100, 75, 50, 25	Wait time (10 millisecond units)

## 5.3 List of Variables

Table 5.2 lists global variables.

Table 5.2 Global variables

Type	Variable Name	Description	Function used
Int	sw_flag	Stores state changes of SW_USR.	poll_sw()
Int	f_edge_count	Counts the number of times SW_USR has been pressed.	poll_sw()

## 5.4 List of Functions

Table 5.3 shows a list of functions.

Table 5.3 List of functions

Function Name	Overview
loop	Main processing (sketch)
setup	Initialization function (sketch)
poll_sw	SW_USR state confirmation and wait processing (10 millisecond units)
pinMode	Specifies the operation mode of a pin (input mode/output mode/input mode with internal pull-up resistor enabled)
digitalWrite	Outputs data to a pin
digitalRead	Read the state of a pin
micros	Returns the time, in microsecond units, from the start of program execution until the present time
millis	Returns the time, in millisecond units, from the start of program execution until the present time
delay	Stops the program for the specified time, in millisecond units
delayMicroseconds	Stops the program for the specified time, in microsecond units

## 5.5 Specification of Functions

The function specifications of the sample code are shown below.

[Function name] loop	
<b>Overview</b>	Main function
<b>Header</b>	AR_LIB_PORT.h, AR_LIB_TIME.h, r_cg_macrodriver.h, AR_SKETCH.h, r_cg_userdefine.h
<b>Declaration</b>	void loop(void)
<b>Description</b>	Upon starting, with the specified time interval, LED0 and LED1 are both extinguished →LED0 is lit→LED1 is lit→LED1 is extinguished
<b>Argument</b>	None
<b>Return value</b>	None
[Function name] setup	
<b>Overview</b>	Initialization function
<b>Header</b>	AR_LIB_PORT.h, r_cg_macrodriver.h r_cg_userdefine.h
<b>Declaration</b>	void setup(void);
<b>Description</b>	Pins used by the program (sketch) are set
<b>Argument</b>	None
<b>Return value</b>	None
[Function name] poll_sw	
<b>Overview</b>	SW_USR check and time wait function
<b>Header</b>	AR_LIB_PORT.h, AR_LIB_TIME.h, r_cg_macrodriver.h, r_cg_userdefine.h
<b>Declaration</b>	void poll_sw(void);
<b>Description</b>	Every 10 milliseconds the state of SW_USR is checked. This operation is repeated a specified number of times and the number of presses of SW_USR is counted.
<b>Argument</b>	None
<b>Return value</b>	None
[Function name] pinMode	
<b>Overview</b>	Function to set the pin function
<b>Header</b>	AR_LIB_PORT.h, r_cg_macrodriver.h, r_cg_userdefine.h
<b>Declaration</b>	void pinMode(uint8_t pin, uint8_t mode)
<b>Description</b>	The pin indicated by the first argument is set to the mode indicated by the second argument
<b>Argument</b>	uint8_t pin : Number of the pin to be specified uint8_t mode : Specifies the pin mode with OUTPUT/INPUT/INPUT_PULLUP
<b>Return value</b>	None

---

**[Function name] digitalWrite**

---

<b>Overview</b>	Function for digital data output to a pin	
<b>Header</b>	AR_LIB_PORT.h, r_cg_macrodriver.h, r_cg_userdefine.h	
<b>Declaration</b>	void digitalWrite(uint8_t pin, uint8_t value);	
<b>Description</b>	The data indicated by the second argument is output to the pin indicated by the first argument	
<b>Argument</b>	uint8_t pin	: Number of the pin for data output
	uint8_t value	: Data to output (HIGH/LOW)
<b>Return value</b>	None	

---

**[Function name] digitalRead**

---

<b>Overview</b>	Function for digital data read out from a pin	
<b>Header</b>	AR_LIB_PORT.h, r_cg_macrodriver.h, r_cg_userdefine.h	
<b>Declaration</b>	uint8_t digitalRead(uint8_t pin);	
<b>Description</b>	The state of the pin specified by the argument is read out	
<b>Argument</b>	uint8_t pin	: Number of the pin to be read out
<b>Return value</b>	uint8_t	: Data that was read out (HIGH/LOW)

---

**[Function name] delay**

---

<b>Overview</b>	Wait function in millisecond units	
<b>Header</b>	AR_LIB_TIME.h, r_cg_macrodriver.h, r_cg_userdefine.h	
<b>Declaration</b>	void delay(uint32_t time);	
<b>Description</b>	Processing waits for the time specified by the argument, in millisecond units	
<b>Argument</b>	uint32_t time	: Wait time (in millisecond units)
<b>Return value</b>	None	

## 5.6 Flowcharts

### 5.6.1 Initial setting function

Figure 5.1 shows a flowchart of the initial setting.

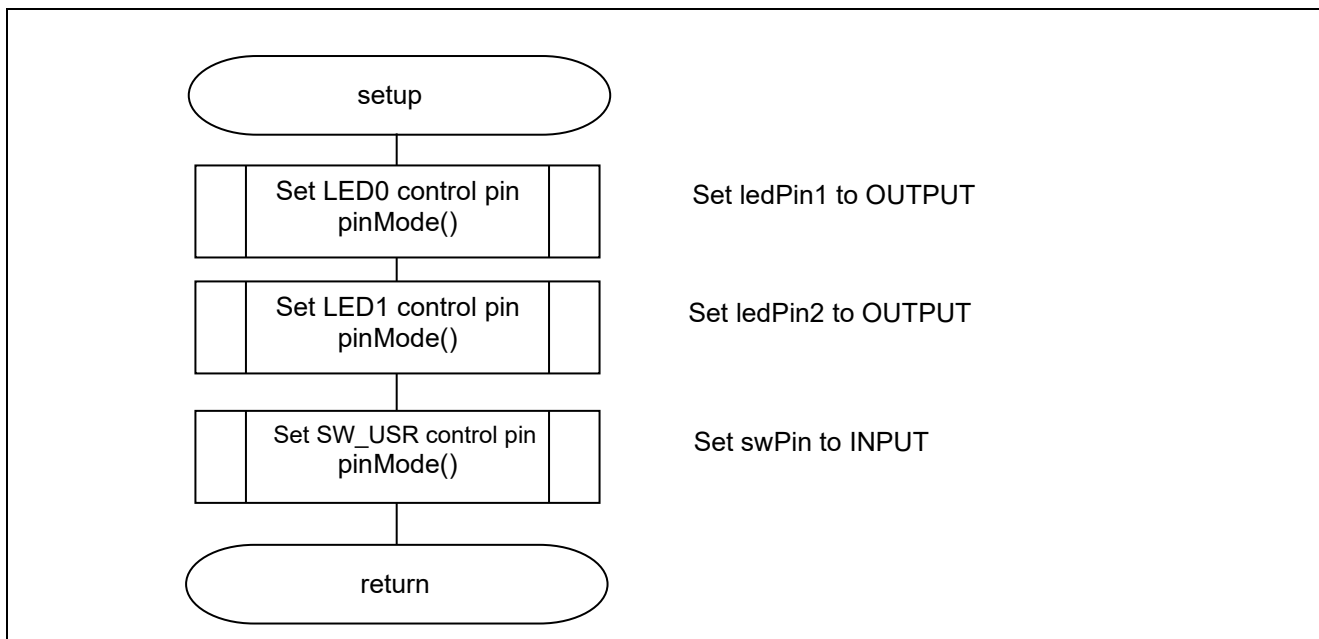


Figure 5.1 Initial setting function

**5.6.2 Main Processing Function**

Figure 5.2 shows a flowchart of the main processing function.

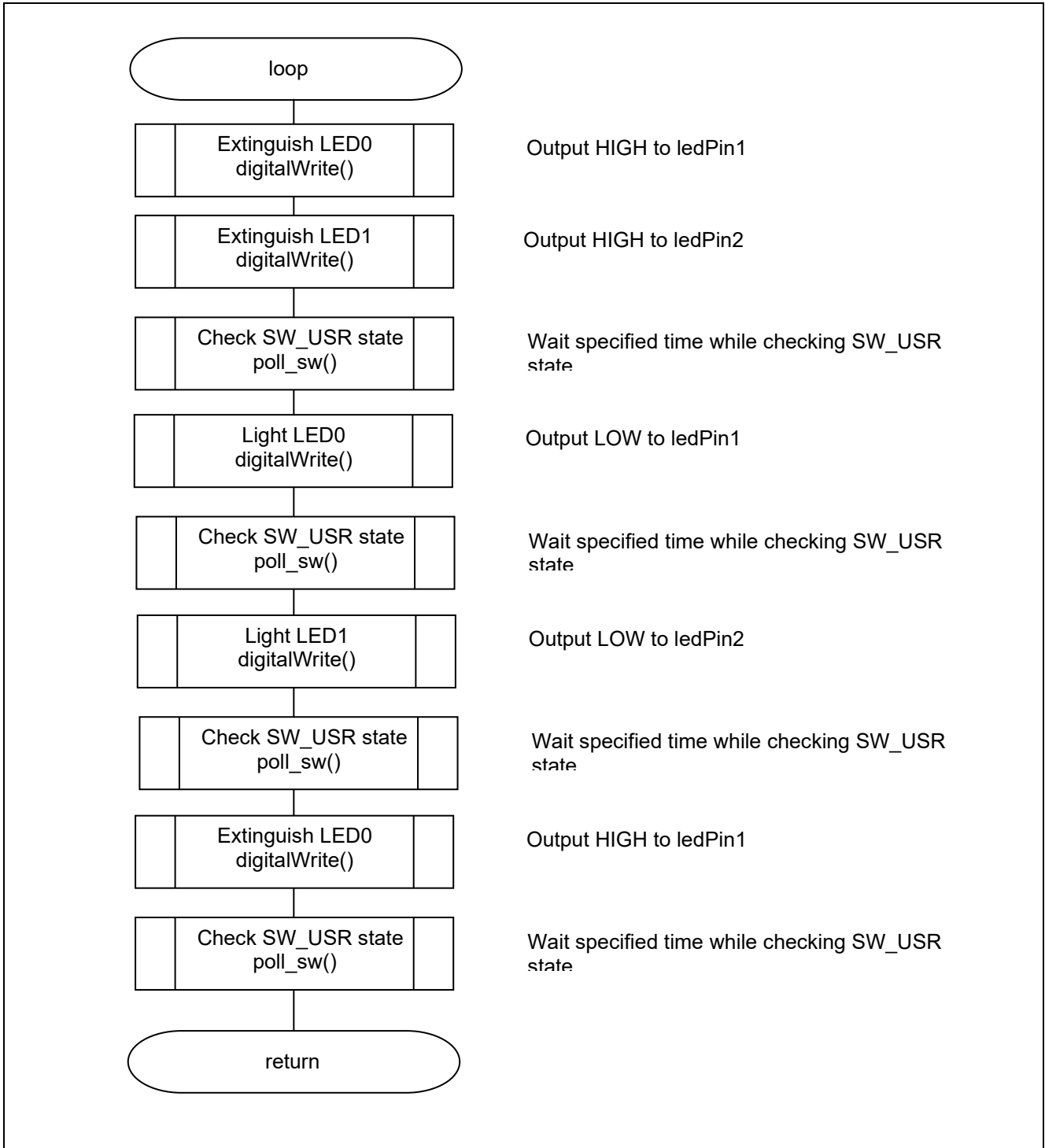


Figure 5.2 Main processing function



**5.6.3 SW\_USR check and wait time function**

Figure 5.3 shows flowcharts of the SW\_USR check and wait time functions.

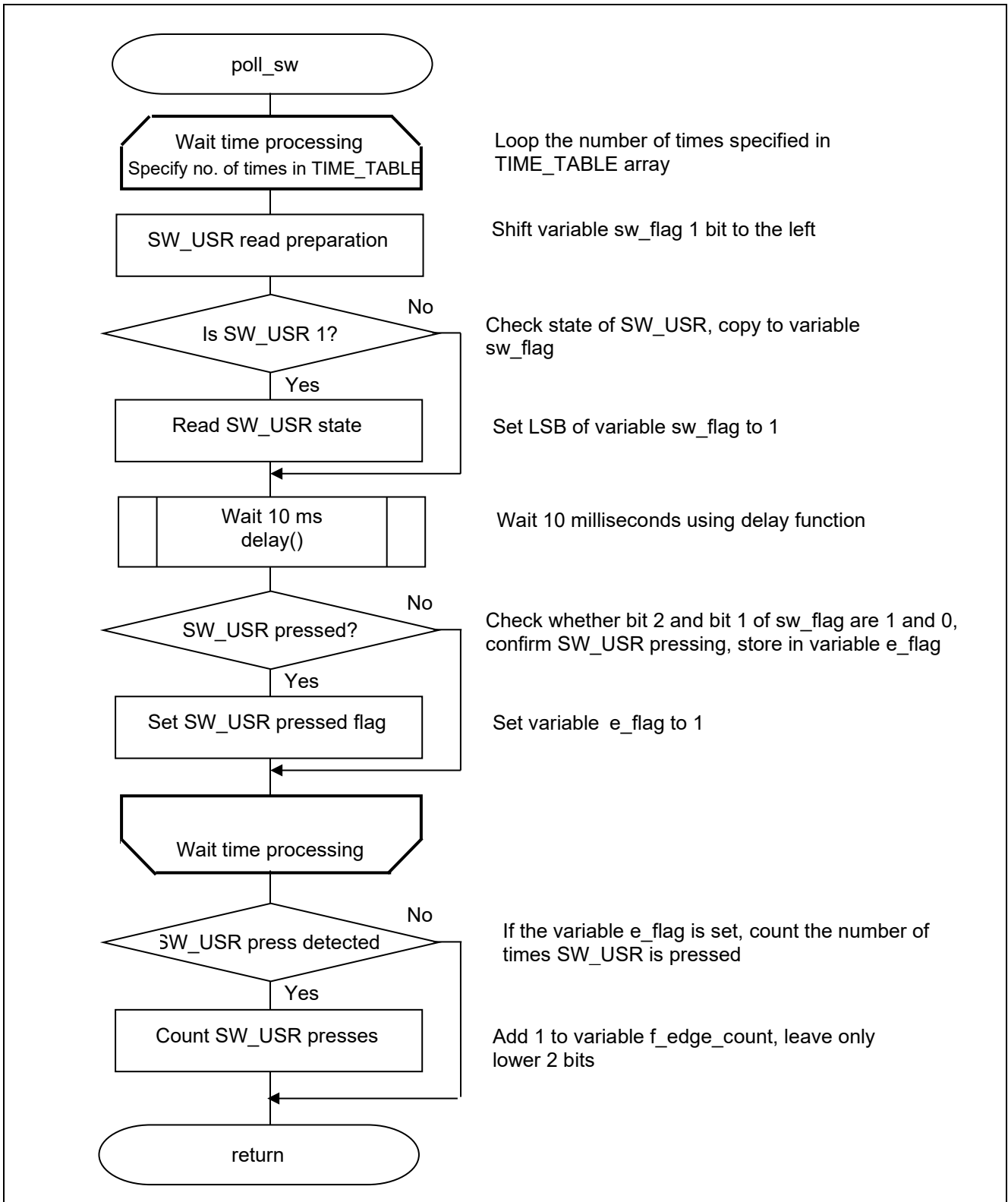


Figure 5.3 SW\_USR check and wait time function

## 6. Sample Code

Sample code can be downloaded from the Renesas Electronics website.

## 7. Reference Documents

RL78/G14 User's Manual: Hardware (R01UH0186)

RL78 family User's Manual: Software (R01US0015)

RL78/G14 Fast Prototyping Board User's Manual (R20UT4573)

RL78 Family Arduino API Introduction Guide (R01AN5413)

(The latest version can be downloaded from the Renesas Electronics website.)

Technical Update/Technical News

(The latest version can be downloaded from the Renesas Electronics website.)

## Website and Support

Renesas Electronics Website

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

Inquiries

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

**Revision History**

Rev.	Date	Description	
		Page	Summary
1.00	June.30.20	—	First Edition

## General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products

The following usage notes are applicable to all Microprocessing unit and Microcontroller unit products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

### 1. Precaution against Electrostatic Discharge (ESD)

A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity.

Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.

### 2. Processing at power-on

The state of the product is undefined at the time when power is supplied. The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the time when power is supplied. In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the time when power is supplied until the reset process is completed. In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the time when power is supplied until the power reaches the level at which resetting is specified.

### 3. Input of signal during power-off state

Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.

### 4. Handling of unused pins

Handle unused pins in accordance with the directions given under handling of unused pins in the manual. The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of the LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible.

### 5. Clock signals

After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is stabilized. When the clock signal is generated with an external resonator or from an external oscillator during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Additionally, when switching to a clock signal produced with an external resonator or by an external oscillator while program execution is in progress, wait until the target clock signal is stable.

### 6. Voltage application waveform at input pin

Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between  $V_{IL}$  (Max.) and  $V_{IH}$  (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between  $V_{IL}$  (Max.) and  $V_{IH}$  (Min.).

### 7. Prohibition of access to reserved addresses

Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.

### 8. Differences between products

Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

## Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
  2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
  3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
  4. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
  5. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The intended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below.
    - "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.
    - "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.
- Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user's manual or other Renesas Electronics document.
6. When using Renesas Electronics products, refer to the latest product information (data sheets, user's manuals, application notes, "General Notes for Handling and Using Semiconductor Devices" in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
  7. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
  8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
  9. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
  10. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
  11. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
  12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

(Rev.4.0-1 November 2017)

## Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,  
Koto-ku, Tokyo 135-0061, Japan  
[www.renesas.com](http://www.renesas.com)

## Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

## Contact information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit:  
[www.renesas.com/contact/](http://www.renesas.com/contact/)