Application Note

RL78/G23

Line Trace Car Sample Sketch (Arduino™ sketch)

Introduction

This application note explains how to use the RL78/G23-64p Fast Prototyping Board (FPB) library for Arduino to control a line trace car.

Target Device

Evaluation Borad	: RL78/G23-64p Fast Prototyping Board
Motor Driver	: AE-DRV8835-S
Photo Reflector	: AE-NJL5901AR-8CH
Mobile Battery	: CHE-061-WH-IOT2
DC Motor (Line Trace Car Kit Included)	: 2WD Mini Smart Robot Mobile Platform Kit for education

Trademarks

Arduino is a trademark of Arduino SA.

Contents

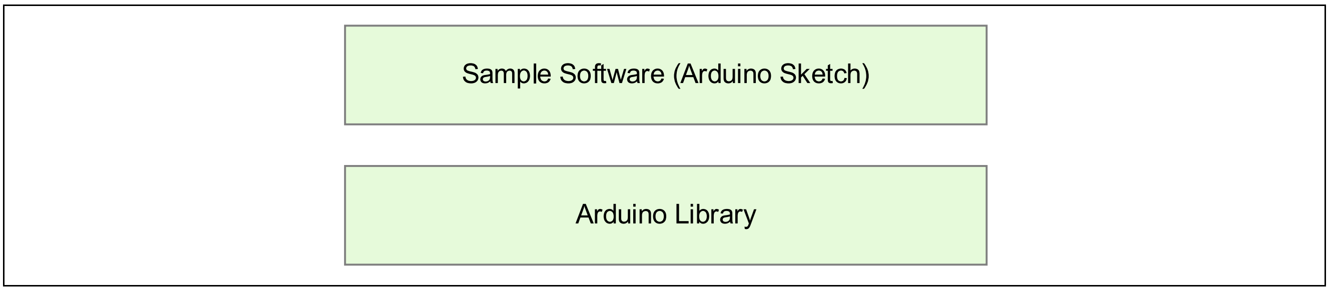
1. System Overview

This system consists of an RL78/G23-64p Fast Prototyping Board, a motor driver, a photo reflector, and a DC motor, and is intended to run the car along the black line on the oval course. The Arduino™ IDE is used to create and write the program.

As a summary of operation, after FPB is powered on, the photo reflector detects the black line and the white background on the oval course and travels along the black line. Based on the information obtained from the photo reflector, the program generates a PWM signal and controls the direction and speed of the two DC motors via the motor driver. This causes the car to run along the black line.

The following shows the block configuration and operation image used in this system.

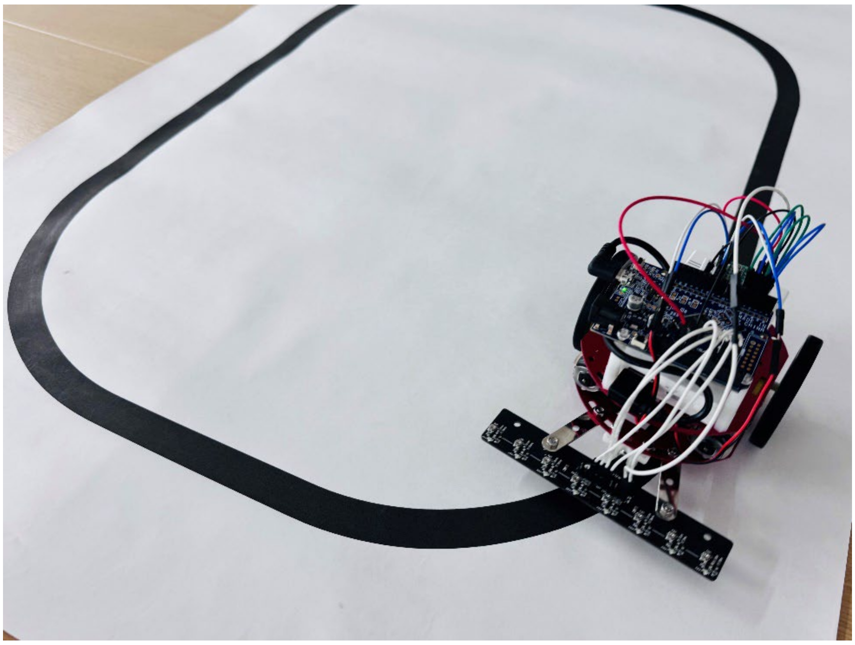
Figure 1-1 Software Block Configuration



Sample Software (Arduino Sketch)

Arduino Library

Figure 1-2 Operation Image



1.1 Modules Used

In the system configuration at the time of development, the power is supplied from a PC to check the output value of the photo-reflector and the operation of the written program.

In the operating system configuration, switch the power supply to the mobile battery and run the line trace car on the course.

The system configuration at the time of development and the time of line trace car operation is shown below.

Figure 1-3 System Configuration at Development

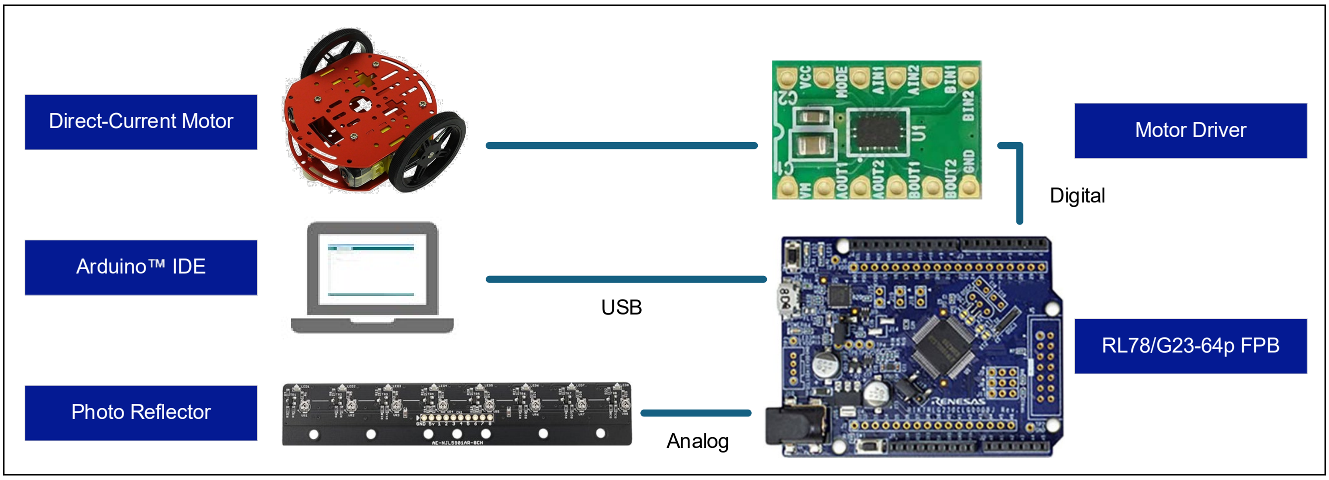
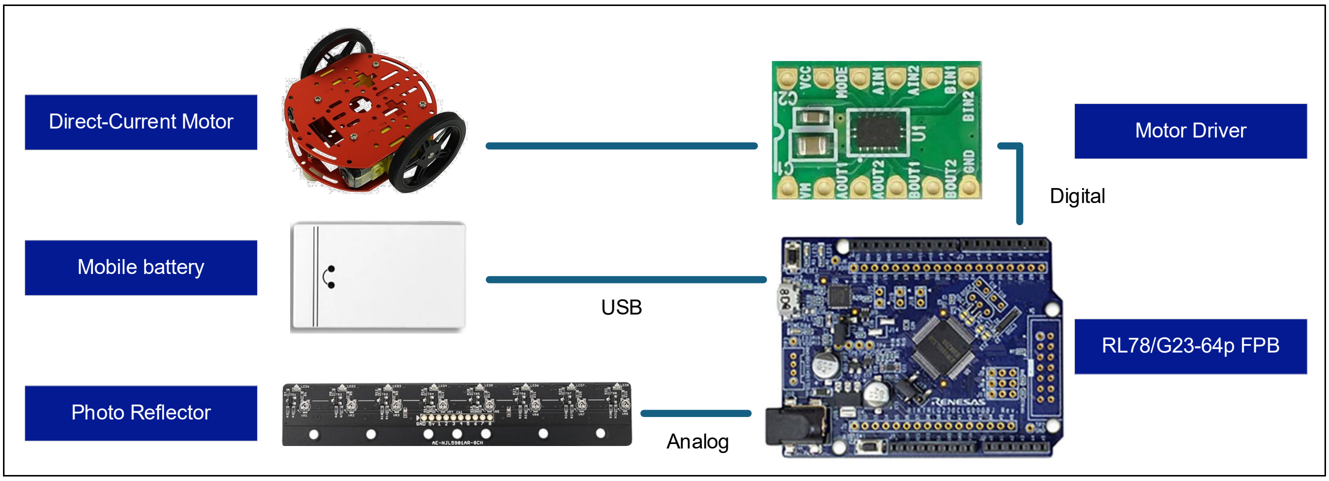


Figure 1-4 System Configuration for Line Trace Car Operation



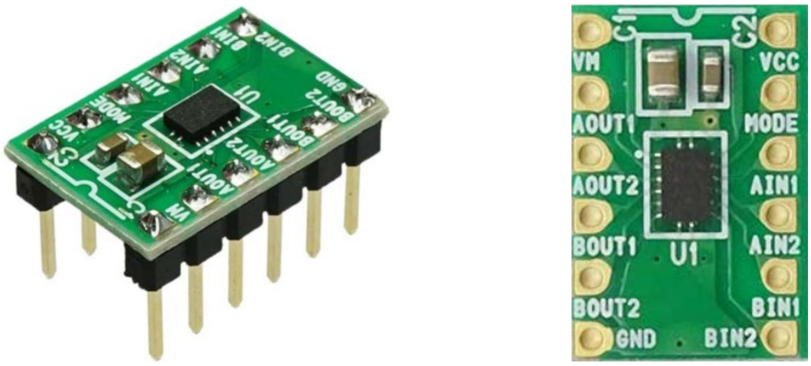
1.1.1 Module with Motor Driver

The motor driver is a circuit to control the rotation speed and direction of the motor. In this system, the module with DRV8835 is used and the IN/IN mode circuit, which controls the rotation speed and direction of forward/reverse rotation with two PWM signals for one motor, is configured for the left and right motors.

The line trace car in this application note moves forward by the reverse control of the motor driver and backward by the forward control. In this sample program, it is set to move forward by reverse control of the motor driver. Refer to the datasheet for detailed DRV8835 specifications.

Figure 1-5 shows the module with motor driver used in this system.

Figure 1-5 Module with Motor Driver

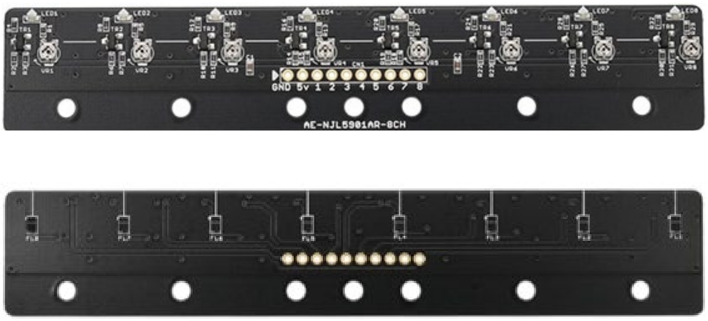


1.1.2 Module with Photo Reflector

The photo reflector is a sensor that detects reflected light from an object and determines the color and distance of the object and is used for line detection and position grasping of the object. This system uses an NJG5901AR photo reflector to determine black and white on the line. The photo reflector is a mechanism to determine the reflectance of an object by irradiating an infrared LED and detecting the reflected light with a light receiving element. Therefore, the output value is high when black is detected and low when white is detected. This is because the black surface easily absorbs light and reflects light weakly, so the output is high by the internal pull-up circuit, while the white surface strongly reflects light and the current easily flows by the light-receiving element, causing the output value to be low. Using this property, the sample sketch reads the analog signal from the photo reflector with ADC of RL78/G23 to determine the black line and white background on the oval course when the line trace car is running. Refer to the datasheet for detailed specifications of the photo reflector.

Figure 1-6 shows the module with photo reflector used in this system.

Figure 1-6 Module with Photo Reflector



1.2 Operations

Figure 1-7 shows the overview of the operation.

(1) After the power is turned on, perform the initialization of GPIO, PWM, and serial communication.

(2) Interrupt handling to switch the flag ON, OFF is executed by pressing the user switch.

(3) If the flag is True, execute (4). If False, execute (7).

(4) The photo reflector detects the reflected light, and the output value if each sensor is acquired by ADC.

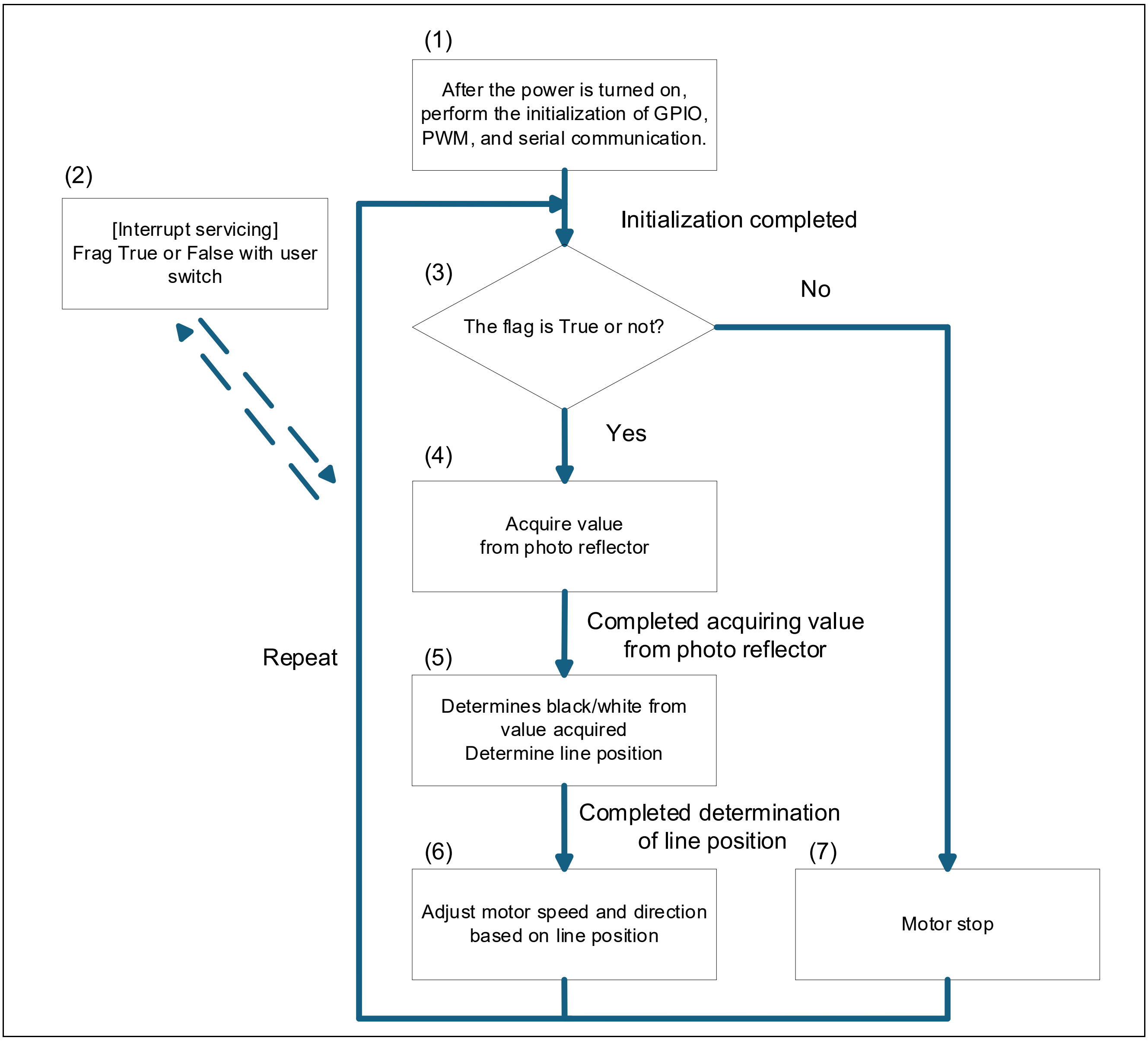
(5) White and black are judged from the acquired values of each photo reflector, and the position of the line is grasped based on the result.

(6) The motor operates by adjusting the motor rotation speed and direction based on the line position.

(7) Motor stopped when flag is OFF.

Remarks. Step (3) through (6) are repeated while the line trace car is powered in.
The value acquired by the photo reflector in step (4) can be checked on the serial plotter of the Arduino™ IDE during development. The sensor threshold settings and operation can be checked.

Figure 1-7 Overview of Operation


(1)
After the power is turned on, perform the initialization of GPIO, PWM, and serial communication.
(2)
[Interrupt servicing] Frag True or False with user switch
Initialization completed
(3)
The flag is True or not?
No
Yes
(4)
Acquire value from photo reflector
Completed acquiring value from photo reflector
Repeat
(5)
Determines black/white from value acquired Determine line position
Completed determination of line position
(6)
Adjust motor speed and direction based on line position
(7)
Motor stop

2. Operation Confirmation Environment

The following shows the operation confirmation environment of this system.

Table 2-1 Operation Confirmation Environments (Hardware)

Item	Contents
Evaluation board	RL78/G23-64p Fast Prototyping Board – RTK7RLG230CLG000BJ
Motor driver	AE-DRV8835-S
Photo reflector	AE-NJL5901AR-8CH
DC motor (Line trace car kit included)	2WD Mini Smart Robot Mobile Platform Kit for education
Mobile battery	CHE-061-WH-IOT2[Note1]
Operating voltage	5V

Note1. The mobile battery with an auto power off function will cut off the power supply if it's used with a low power consumption system for a certain period. Therefore, please use the mobile battery with the auto power off function disabled.

Table 2-2 Operating Confirmation Environment (Software)

Item	Contents	Version
OS	Windows 10 Pro	-
Integrated development environment (IDE)	Arduino™ IDE	2.3.4
Library	RL78/G23-64p FPB library	2.4.0

3. Build Development Environment

How to connect boards and how to set up the Arduino™ IDE are explained.

This system uses Arduino™ IDE 2.3.4. If you have not installed Arduino™ IDE 2.3.4 or later, please install it.

https://www.arduino.cc/en/software

3.1 Board Connection

Figure 3-1 shows the connection between the evaluation board, the module with motor driver, and the module with photo reflector.

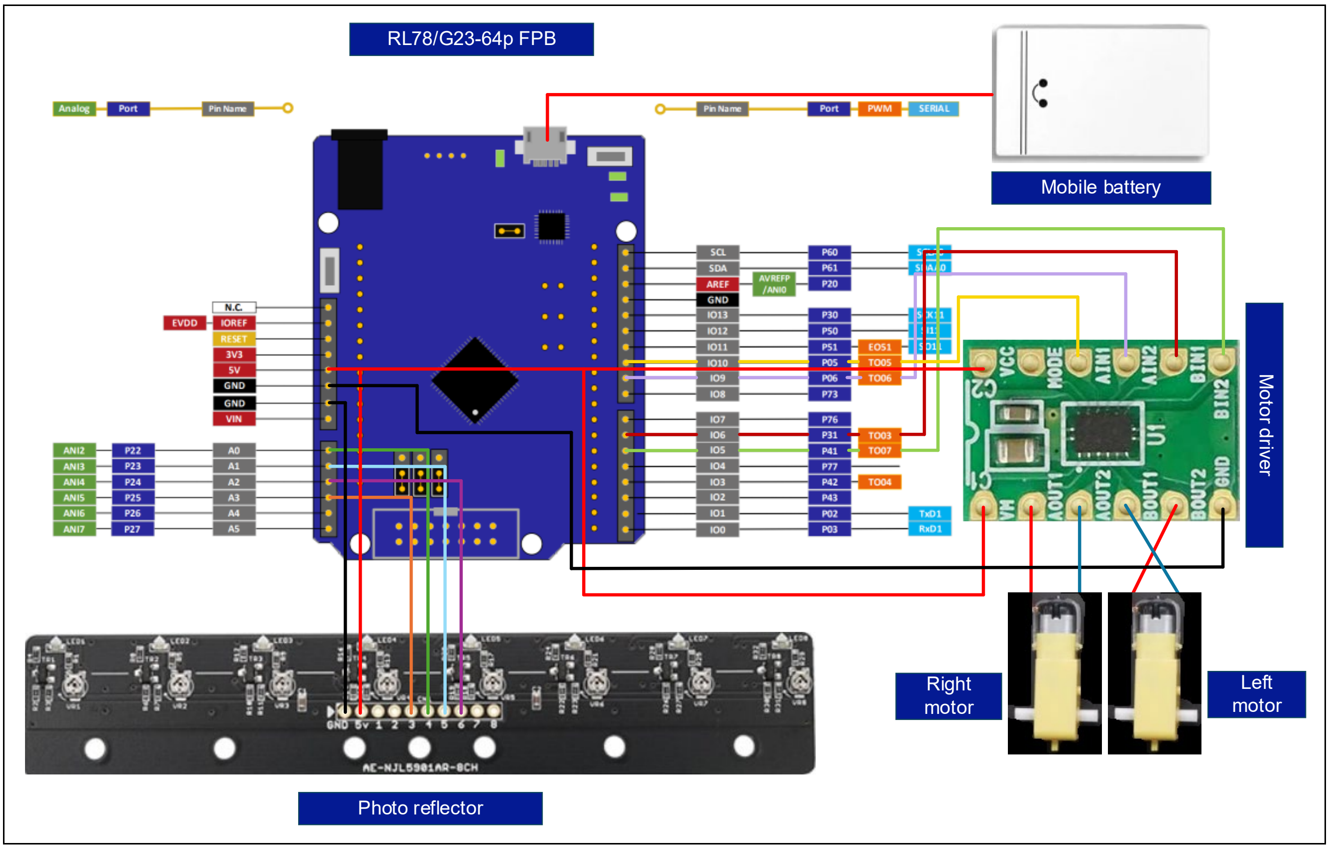
This system uses USB to supply power to the evaluation board. Check the circuit of the evaluation board and set jumpers as necessary.

In this system, the jumpers on the evaluation board are set as follows.

Table 3-1 Jumper Settings of Evaluation Board

Jumper	Setting	Function
J8	1-2 short	COM port debug
J9		
J11		
J13	Open	
J17	1-2 short	5V power supply to MCU

Figure 3-1 Connection of Evaluation Board, Module with Motor Driver and Module with Photo Reflector



3.2 List of Pins Used

The following shows the pins used in this system.

Table 3-2 List of Pins Used in This System

Arduino™ signal name	Pin number of FPB	Pin number of sketch	Pin function	Destination of Connection
IO~6	21	6	Forward control: PWM output (0%) Backward control: Speed control of the left motor with PWM output	Forward control: BIN1 of motor driver Backward control: BIN2 of motor driver
IO~5	4	5	Forward control: Speed control of the left motor with PWM output Backward control: PWM output (0%)	Forward control: BIN2 of motor driver Backward control: BIN1 of motor driver
IO~10	31	10	Forward control: PWM output (0%) Backward control: Speed control of the right motor with PWM output	Forward control: AIN1 of motor driver Backward control: AIN2 of motor driver
IO~9	30	9	Forward control: Speed control of the right motor with PWM output Backward control: PWM output (0%)	Forward control: AIN2 of motor driver Backward control: AIN1 of motor driver
A0	54	A0	Analog input: Acquire from photo reflector	Photo reflector LED4
A1	53	A1	Analog input: Acquire from photo reflector	Photo reflector LED5
A2	52	A2	Analog input: Acquire from photo reflector	Photo reflector LED6
A3	51	A3	Analog input: Acquire from photo reflector	Photo reflector LED3
5V	5V	-	5V power supply	・5V of photo reflector ・VSS of motor driver ・VM of motor driver
GND	GND	-	GND	・GND of motor driver
GND	GND	-	GND	・GND of photo reflector

For detailed pin descriptions of each board, please refer to the following manuals.

- RL78/G23-64p Fast Prototyping Board User's Manual (R20UT4814)
- RL78G23 64pin Fast Prototyping Board · renesas/Arduino Wiki · GitHub

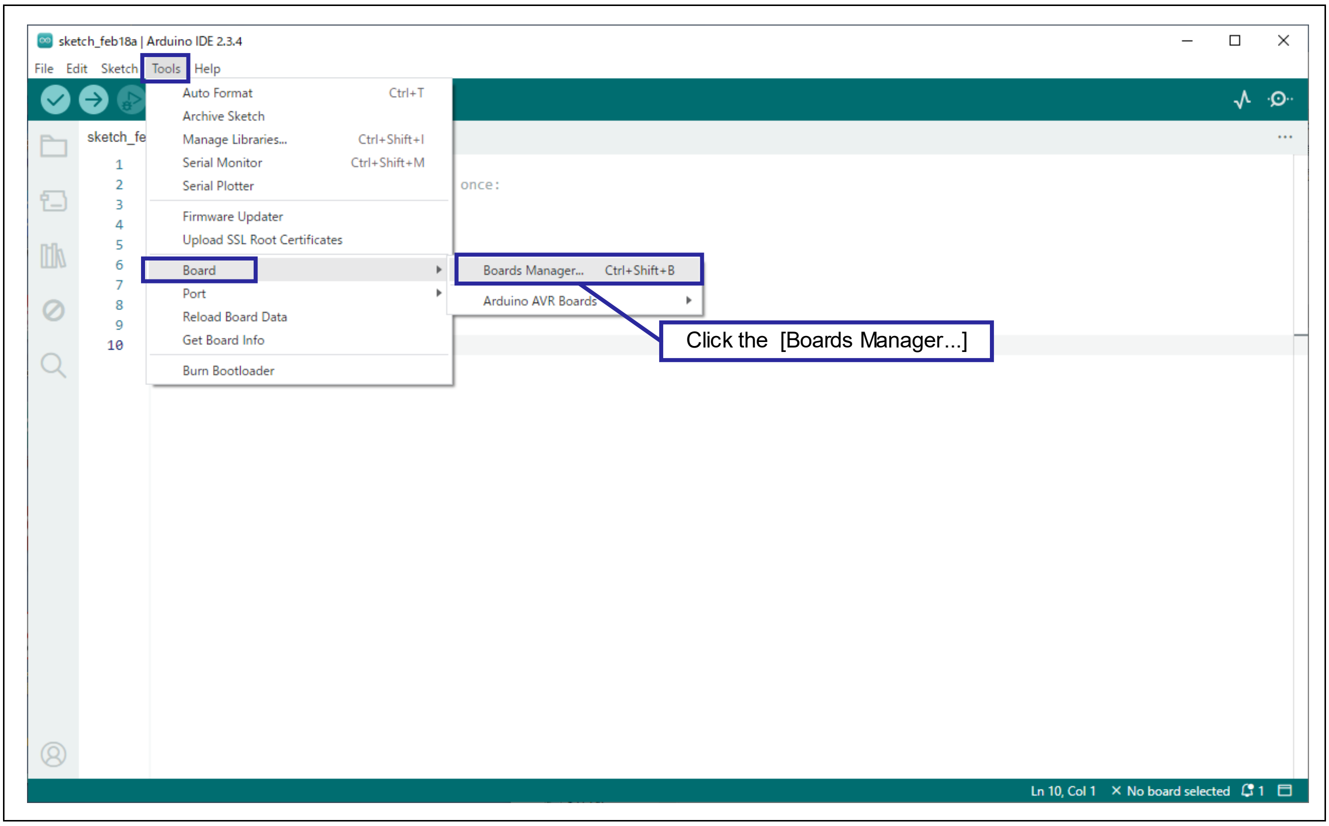
3.3 Setup of Arduino™ IDE

The setup procedure of Arduino™ IDE is explained.

Remark. The setup procedure is almost the same as the procedure explained on the Quick Start Guide : renesas/Arduino Wiki · GitHub. The sample sketch to flash LED is described on the above site. Refer to it if required.

1. Start the Arduino™ IDE.
2. Select the [Tools] – [Board] – [Board Manager…] menu.

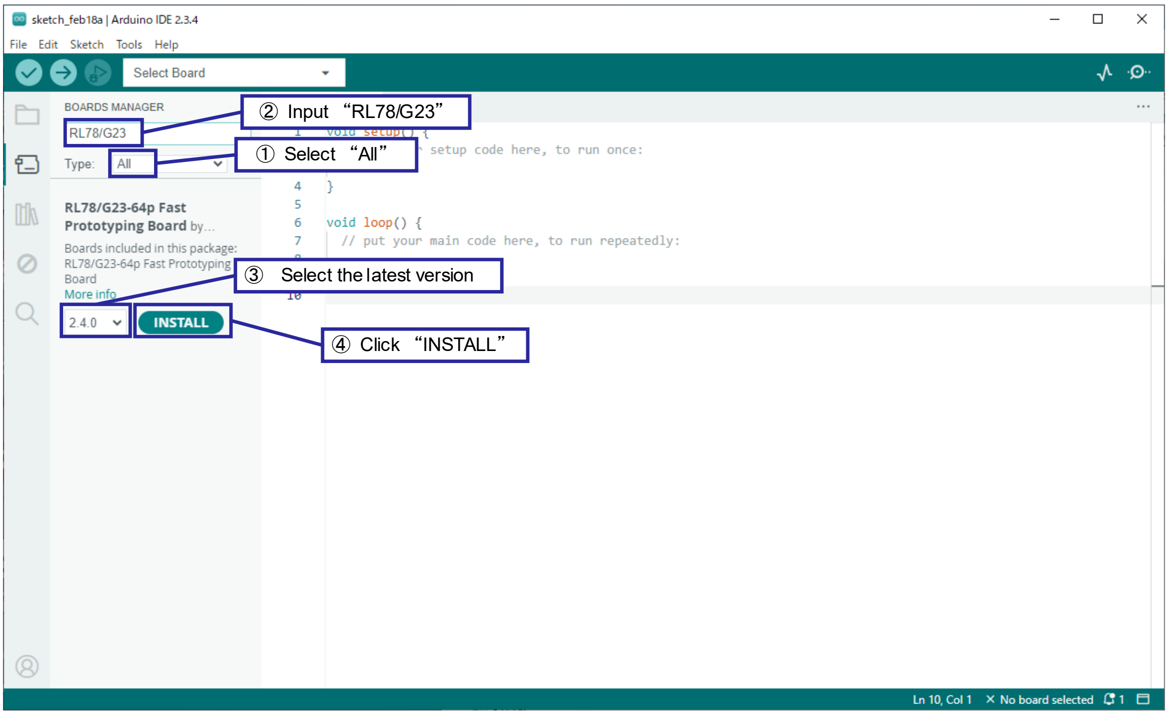
Figure 3-2 Selection of [Board Manager…]


sketch_feb18a | Arduino IDE 2.3.4
File Edit Sketch Tools Help
Auto Format Ctrl+T
Archive Sketch
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter
Firmware Updater
Upload SSL Root Certificates
Board
Port
Reload Board Data
Get Board Info
Burn Bootloader
Boards Manager... Ctrl+Shift+B
Arduino AVR Boards
sketch_fe
once:
Ln 10, Col 1 × No board selected 1

Click the [Boards Manager...]

3. Select “All” at the [Type] and input “RL78/G23” in the textbox. Then, “RL78/G23-64p Fast Prototyping Board” is displayed. Next, click the [INSTALL].

Figure 3-3 Installation of Board Manager



sketch_feb18a | Arduino IDE 2.3.4
File Edit Sketch Tools Help
Select Board
BOARDS MANAGER
INSTALL
Ln 10, Col 1 × No board selected 1

RL78/G23

Type: All

RL78/G23-64p Fast Prototyping Board by...

Boards included in this package: RL78/G23-64p Fast Prototyping Board

More info

2.4.0

```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:
```

② Input “RL78/G23”

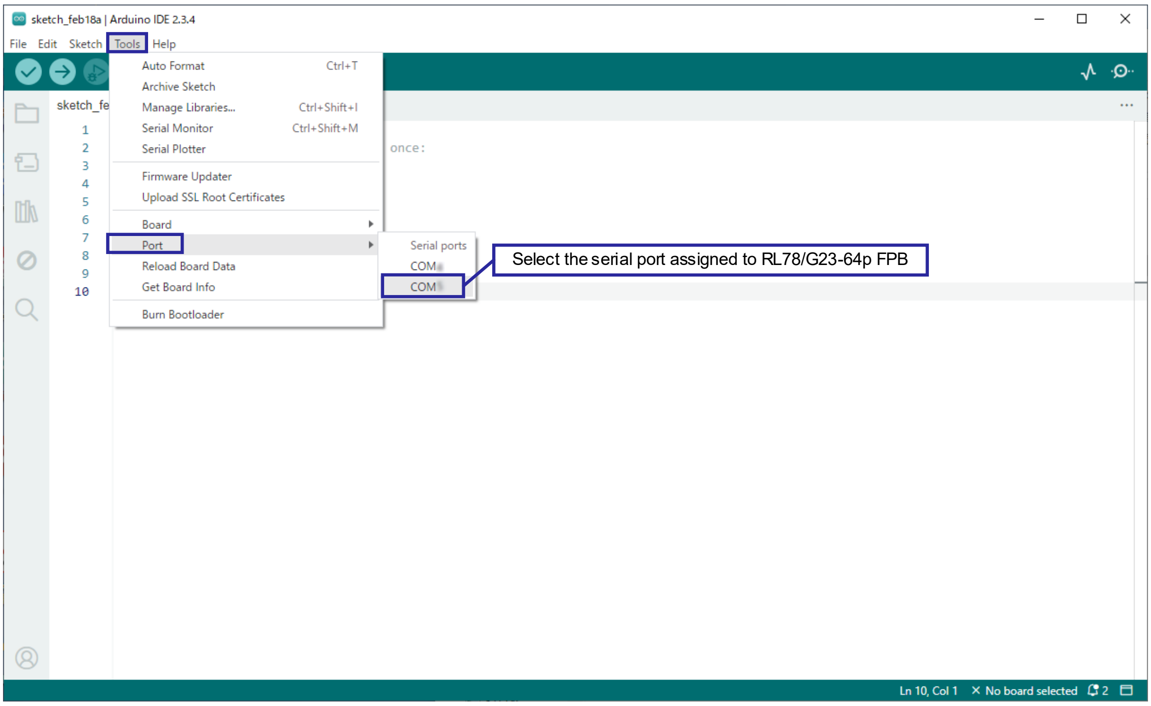
① Select “All”

③ Select the latest version

④ Click “INSTALL”

4. Select the serial port assigned to the RL78/G23-64p FPB from the [Tools] - [Port] menu.
COM port number can be checked at the Device Manager of Windows.

Figure 3-4 Selection of Serial Port

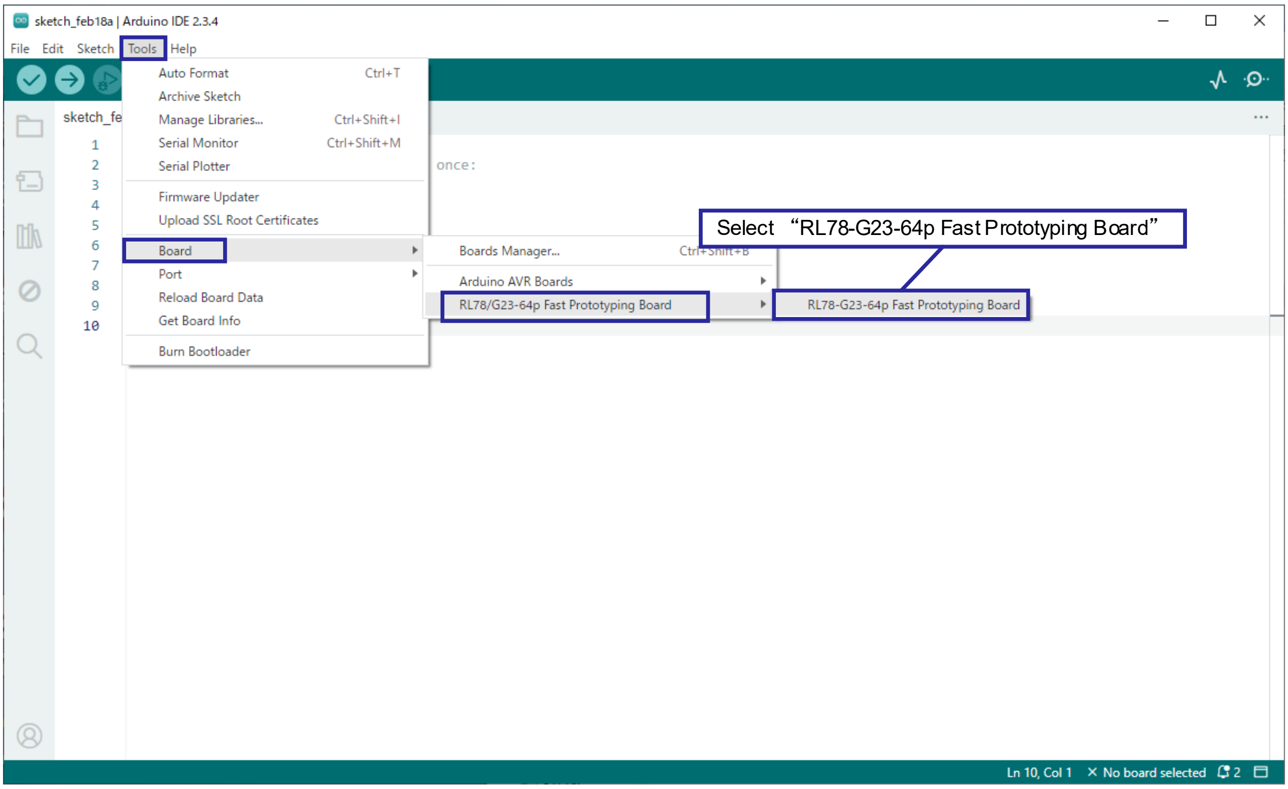


sketch_feb18a | Arduino IDE 2.3.4
File Edit Sketch Tools Help
Auto Format Ctrl+T
Archive Sketch
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter
Firmware Updater
Upload SSL Root Certificates
Board
Port
Reload Board Data
Get Board Info
Burn Bootloader
Serial ports
COM
COM
sketch_fe
once:
Ln 10, Col 1 × No board selected 2

Select the serial port assigned to RL78/G23-64p FPB

5. Select the [Tools] - [Board] - [RL78/G23-64p Fast Prototyping Board] - [RL78-G23 Fast Prototyping Board 64Pin] menu.

Figure 3-5 Selection of Board



sketch_feb18a | Arduino IDE 2.3.4
File
Edit
Sketch
Tools
Help
Auto Format Ctrl+T
Archive Sketch
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter
Firmware Updater
Upload SSL Root Certificates
Board
Port
Reload Board Data
Get Board Info
Burn Bootloader
Boards Manager... Ctrl+Shift+B
Arduino AVR Boards
RL78/G23-64p Fast Prototyping Board
RL78-G23-64p Fast Prototyping Board
Select "RL78-G23-64p Fast Prototyping Board"
sketch_fe
once:
Ln 10, Col 1
No board selected
2

4. Software

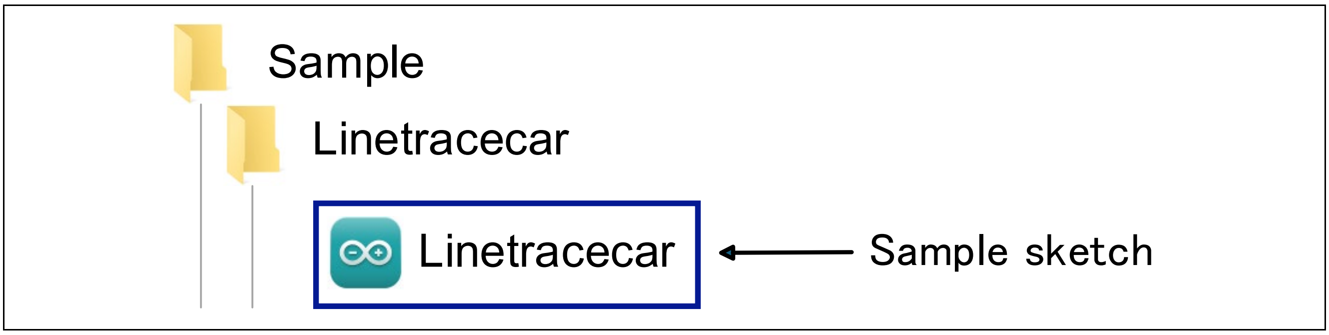
4.1 Overview of Sample Sketch

This sample code is a sample sketch executed on Arduino™ IDE.

The following shows the file structure.

Refer to “4.2 API Functions” for API functions to be used and “4.3 Operation Check Procedure of Sample Sketch” for details of sample sketch.

Figure 4-1 File Structure of Sample Code



4.1.1 Algorithm of Line Trace Car

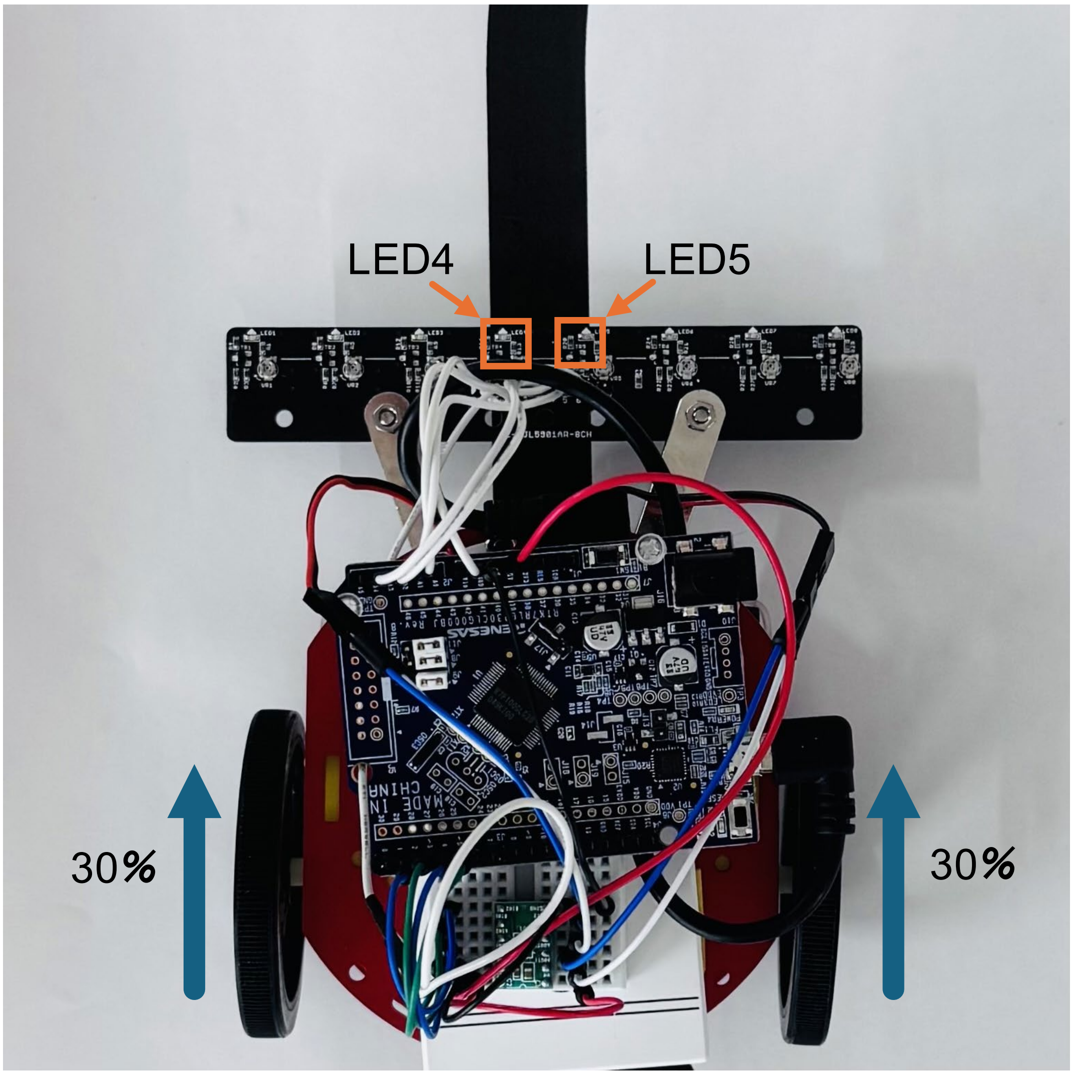
The line trace car acquires sensor data from the photo reflector to determine the position of the line. Multiple photo reflectors detect black and white on the line and recognize the position of the car by integrating the detected data.

Based on the data acquired from the photo reflector, it is classified into three types: **linear motion**, **linear auxiliary motion**, and **curve motion**, and each motion is controlled by adjusting the speed and direction of the motor based on the sensor's judgement results.

First, **linear motion** occurs when the photo reflector's left and right sensors (LED4, LED5) are detecting black on the line. In sample sketch, both motors are driven at a **basic speed** (set to a duty ratio of 30%), and control is performed to advance the car forward. This is the most basic operation when the center of the line is recognized.

Figure 4-2 shows the relationship between linear motion.

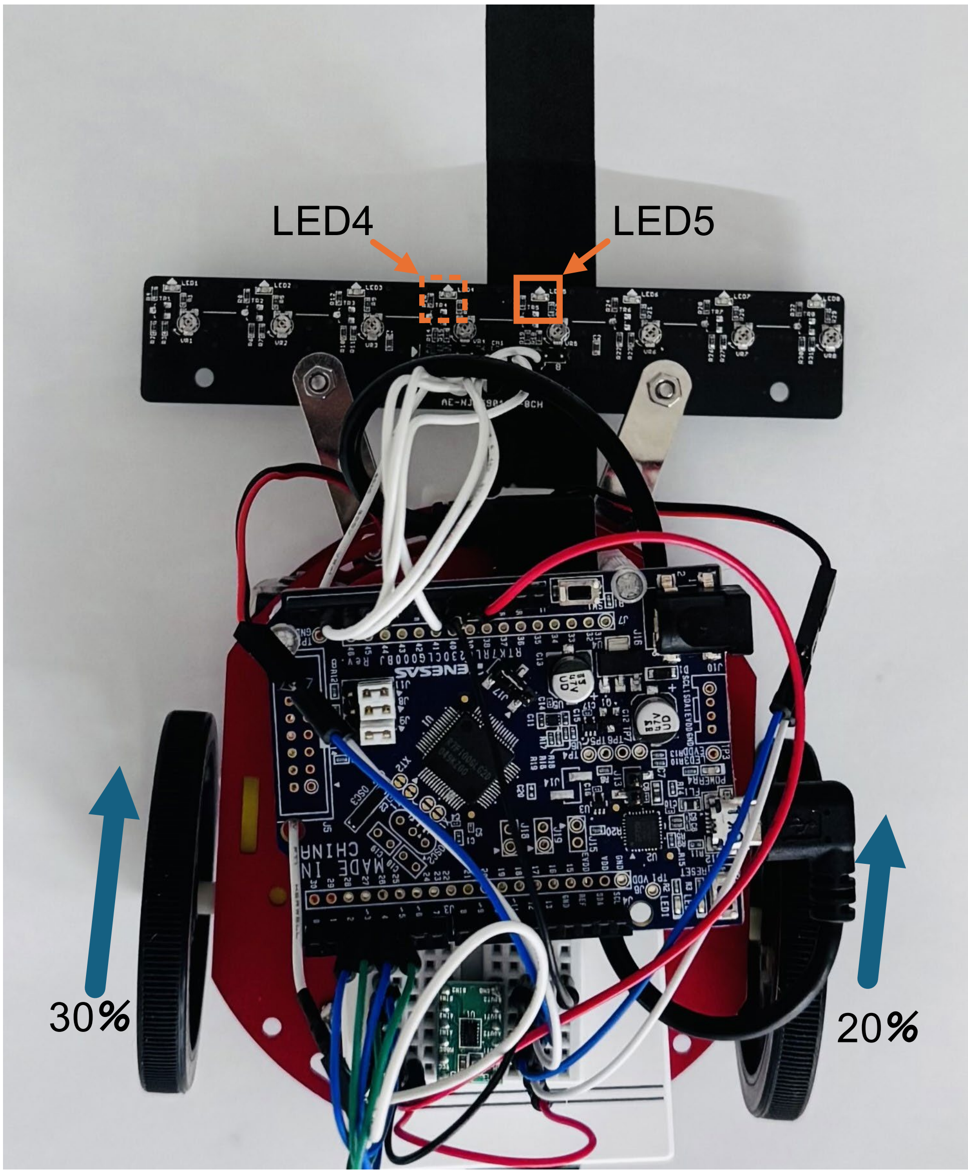
Figure 4-2 Image Figure of Linear Motion



Next, **linear auxiliary operation** occurs when one sensor of the photo reflector detects black and the other detects white. In this case, to maintain the line, the sample sketch corrects the direction by reducing the duty ratio of one motor by 10% from the **basic speed** to maintain the line. With this control, auxiliary operation is performed according to the detection result of the line.

Figure 4-3 shows the relationship between linear auxiliary operation. The figure shows the case where LED4 is white = 0 and LED5 is black = 1 as an example.

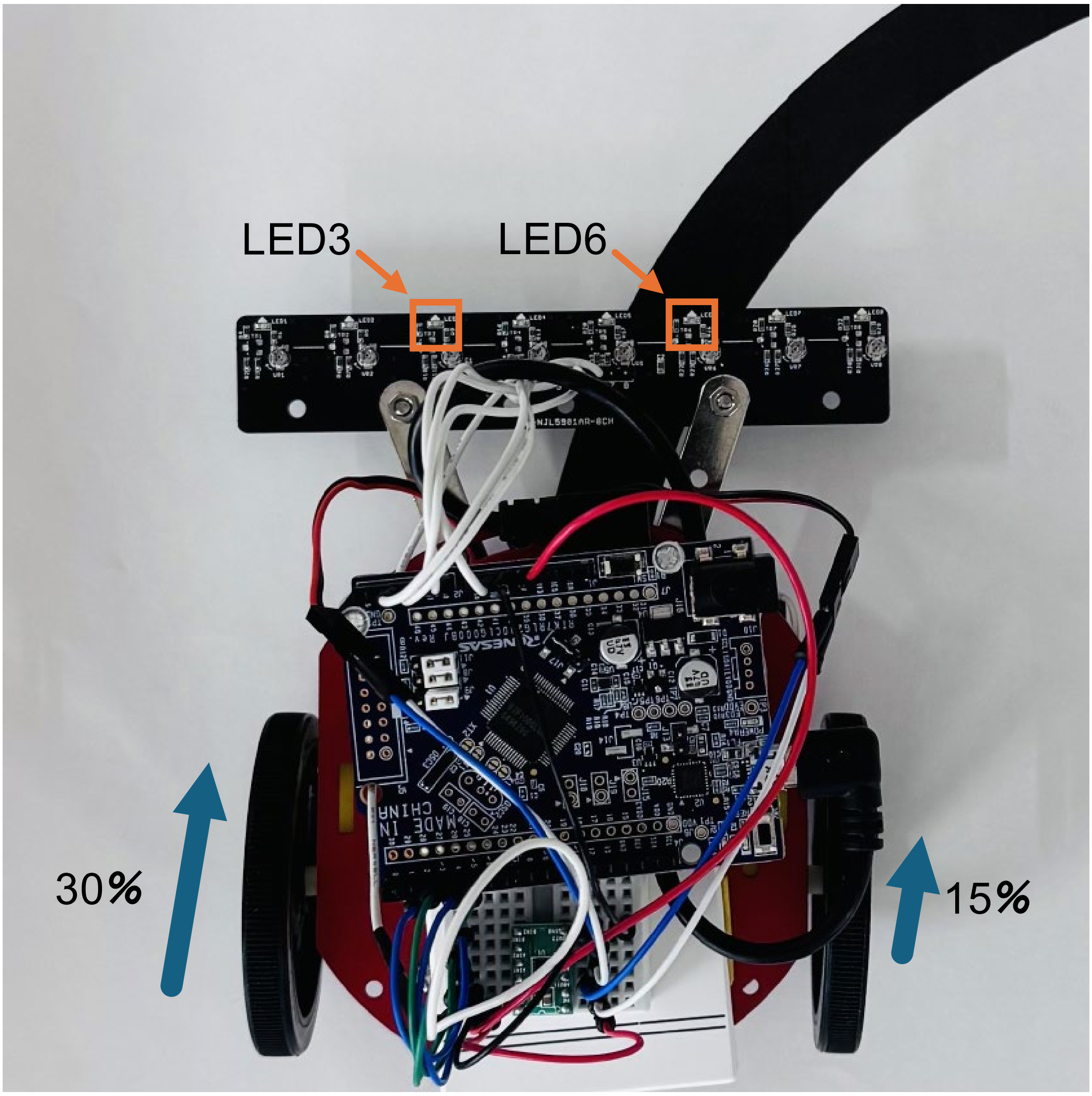
Figure 4-3 Image Figure Linear Auxiliary Operation



Curve operation occurs when the auxiliary sensor (LED3, LED6) detects black. In the sample sketch, by decreasing the duty ratio of one motor by 15% from the **basic speed** operation corresponding to the curve of the line is realized. If LED3 detects black, it performs the left curve and if LED6 detects black, it performs the right curve.

Figure 4-4 shows the image of the right curve operation when LED6 detects black.

Figure 4-4 Image Figure Curve Operation



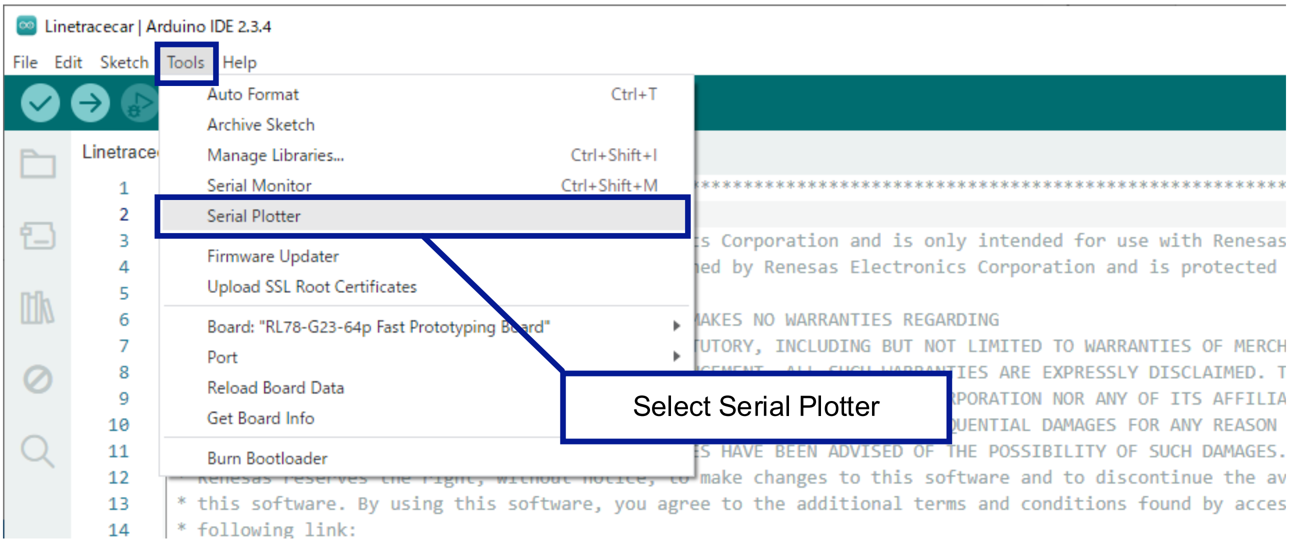
4.1.2 Photo Reflector Threshold Setting Method

This section describes how to set the threshold for black line detection by the photo reflector. The serial plotter function of the Arduino™ IDE is used to set the threshold value.

1. Display the serial plotter

Select [Tools] - [Serial plotter] menu. The serial plotter is displayed as shown in Figure 4-6.

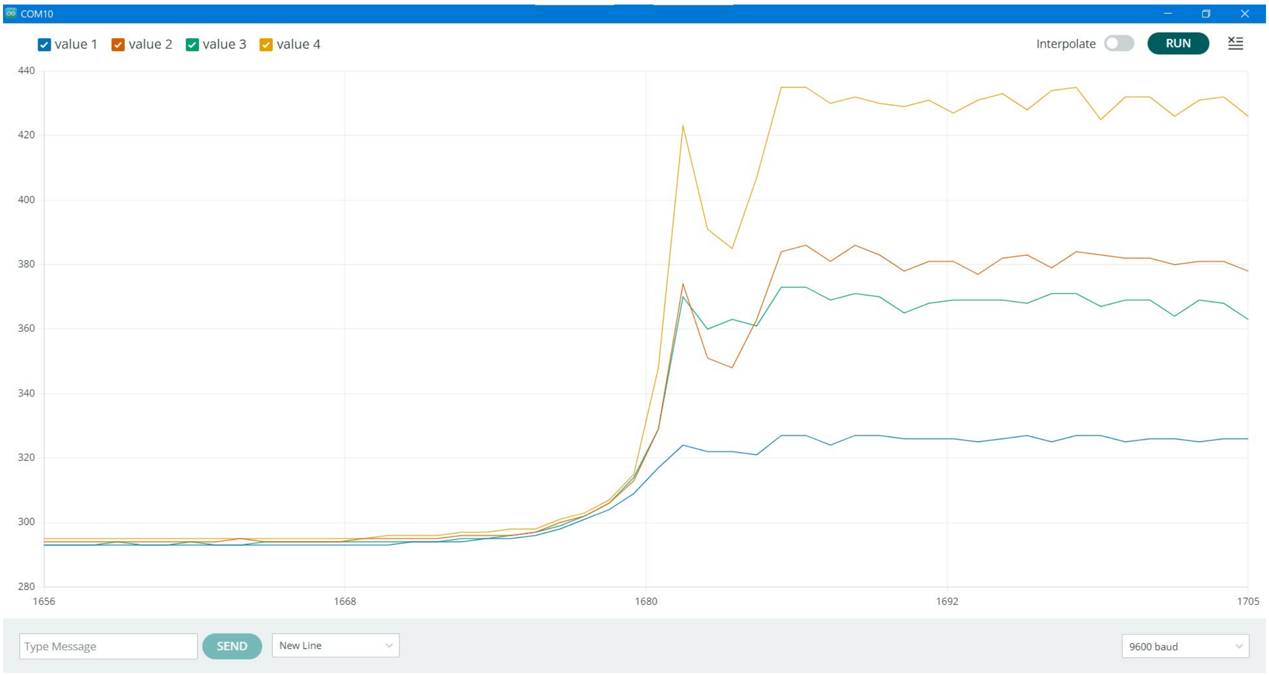
Figure 4-5 How to Display Serial Plotter



```
**************************************************************
s Corporation and is only intended for use with Renesas
ed by Renesas Electronics Corporation and is protected

AKES NO WARRANTIES REGARDING
TUTORY, INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCH
IES ARE EXPRESSLY DISCLAIMED. T
RPORATION NOR ANY OF ITS AFFILIA
UENTIAL DAMAGES FOR ANY REASON
S HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
make changes to this software and to discontinue the av
* this software. By using this software, you agree to the additional terms and conditions found by acces
* following link:
```

Figure 4-6 Example of Displaying Serial Plotter



2. Photo Reflector Threshold Setting

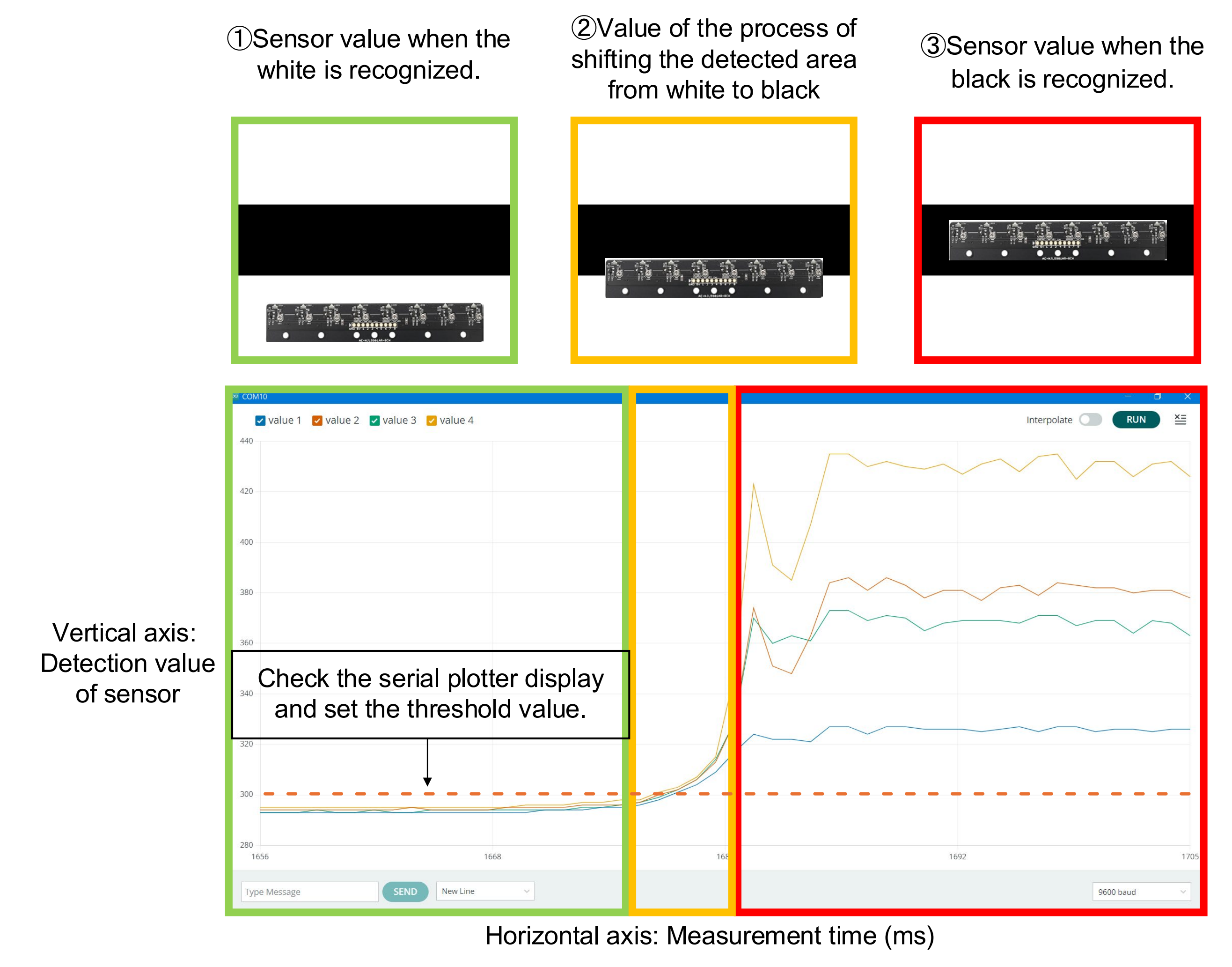
The serial plotter displays the data acquired from the photo reflector. The horizontal axis of the graph indicates the time. The sensor's detection value is updated in real time for each 100 ms.

The vertical axis of the serial plotter represents the detection value of each sensor. This value is acquired by Arduino™ analogRead() is based on the intensity of the reflected light sensed by the sensor.

analogRead() converts the analog signal from the sensor into a 10-bit digital value (0 to 1023). This transformation is based on the reference voltage (typically 5V or 3.3V). The value increases when the black line is detected and decreases when it is white. On the serial plotter, you can see how the sensor recognizes black and white.

In the sample sketch, the threshold is set to 300 based on the detection value of the photo reflector. With this setting, when the detected value of the photo reflector exceeds 300, it is judged as “black”, and when it is 300 or less, it is judged as “white”. Each operation is realized by controlling the drive of the motor based on this judgement result. Figure 4-7 shows the contents displayed on the serial plotter during development.

Figure 4-7 Contents Displayed on the Serial Plotter



①Sensor value when the white is recognized.

②Value of the process of shifting the detected area from white to black

③Sensor value when the black is recognized.

Vertical axis: Detection value of sensor

Check the serial plotter display and set the threshold value.

Horizontal axis: Measurement time (ms)

Remarks. Individual differences in the photo reflectors used in this study are measured, and the threshold is set to the value at which all sensors can distinguish the black line.

4.2 API Functions

The following shows the API functions of each library used in this sample code.

Table 4-1 List of Functions Used

API function name	Function
digitalWrite(pin,value)	Output HIGH/LOW to digital pins.
analogWrite(pin, value)	Output PWM signal to digital pins.
analogRead(pin)	Acquired value from analog pins.
pinMode(pin, mode)	Set pins mode (I/O)
Serial.Begin(baudRate)	Initialize serial communication
Serial.print(data) / Serial.println(data)	Display the data acquisition speed of photo reflector to the serial monitor
delay(ms)	Stop the programs for the specified time (ms)
attachInterrupt(interrupt, ISR, mode)	Interrupt is set for the specified pin Calls the specified function (ISR) when the specified condition (mode) is met
digitalPinToInterrupt(pin)	Acquire the interrupt number corresponding to the specified pin based on the board.

For function specifications of each library, refer to the website of Arduino™ and the other

API List · renesas/Arduino Wiki · GitHub

digitalWrite() | Arduino Documentation

analogRead() - Arduino Reference

analogWrite() - Arduino Reference

Serial.begin() - Arduino Reference

Serial.print() - Arduino Reference

Serial.println() - Arduino Reference

pinMode() - Arduino Reference

delay() - Arduino Reference

attachInterrupt() - Arduino Reference

digitalPinToInterrupt() - Arduino Reference

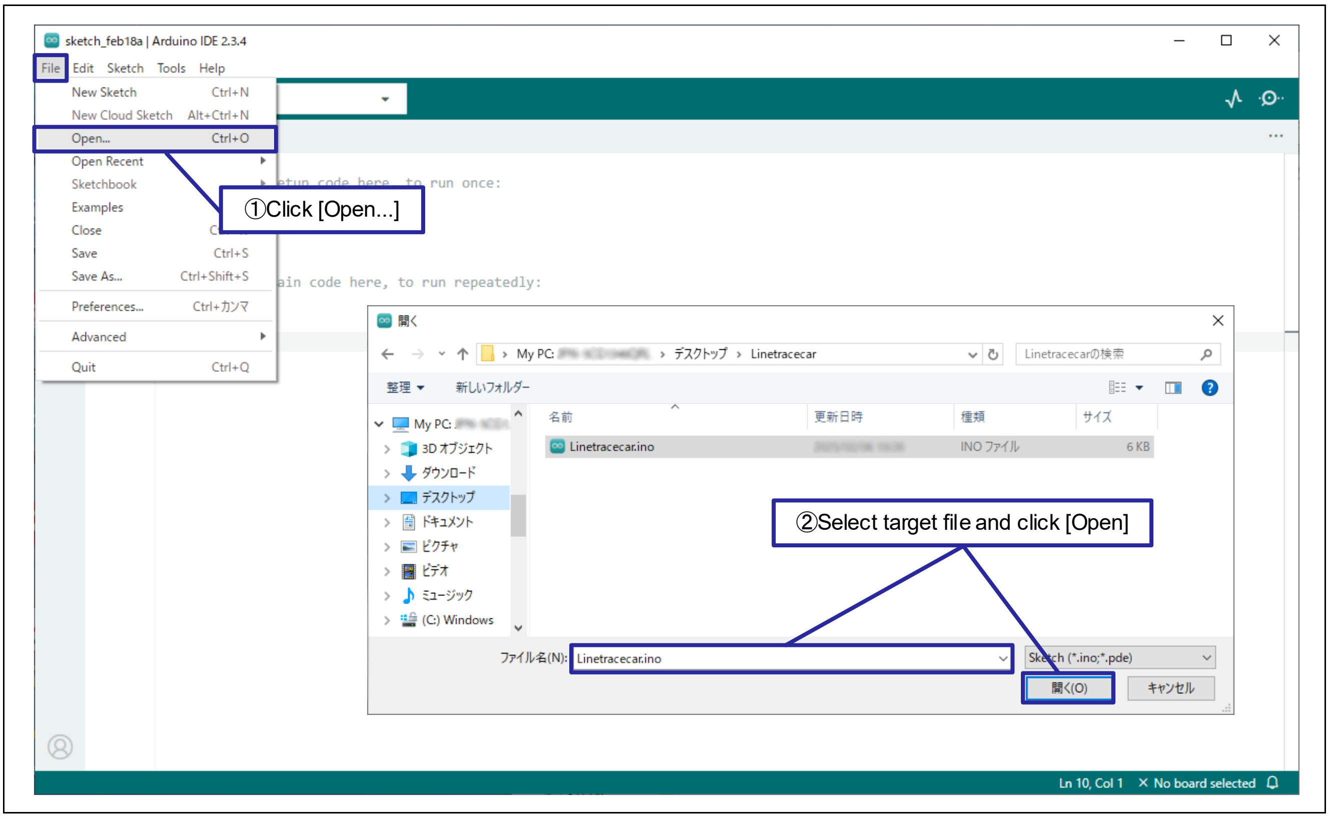
4.3 Operation Check Procedure of Sample Sketch

The procedure for checking the operation of this sample sketch is shown below.

Perform ”3.3 10Setup of Arduino™ IDE” in advance.

1. Click the [File] - [Open...] menu and open the sample sketch “Linetracecar.ino”.

Figure 4-8 Open Sample Sketch

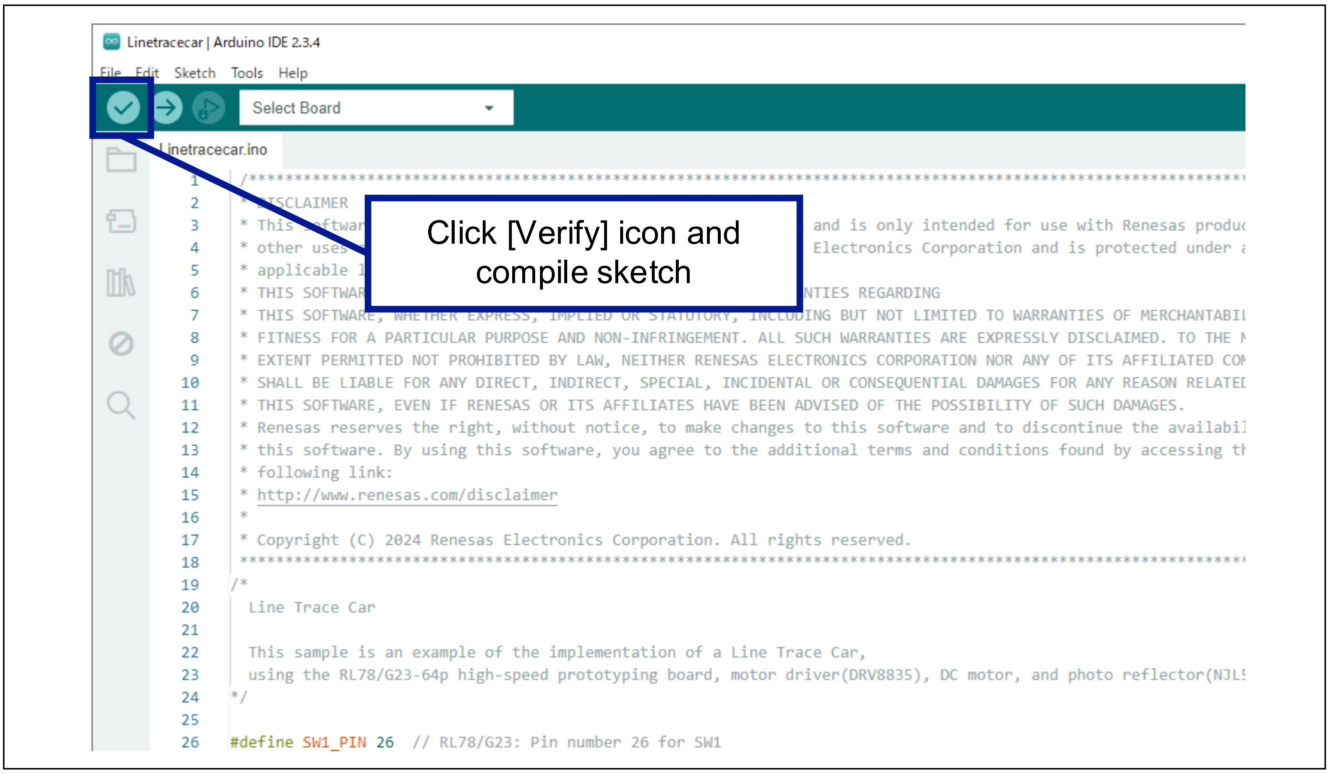


①Click [Open...]

②Select target file and click [Open]

2. Click the [Verify] icon and compile sketch.

Figure 4-9 Compile Sketch



Click [Verify] icon and compile sketch

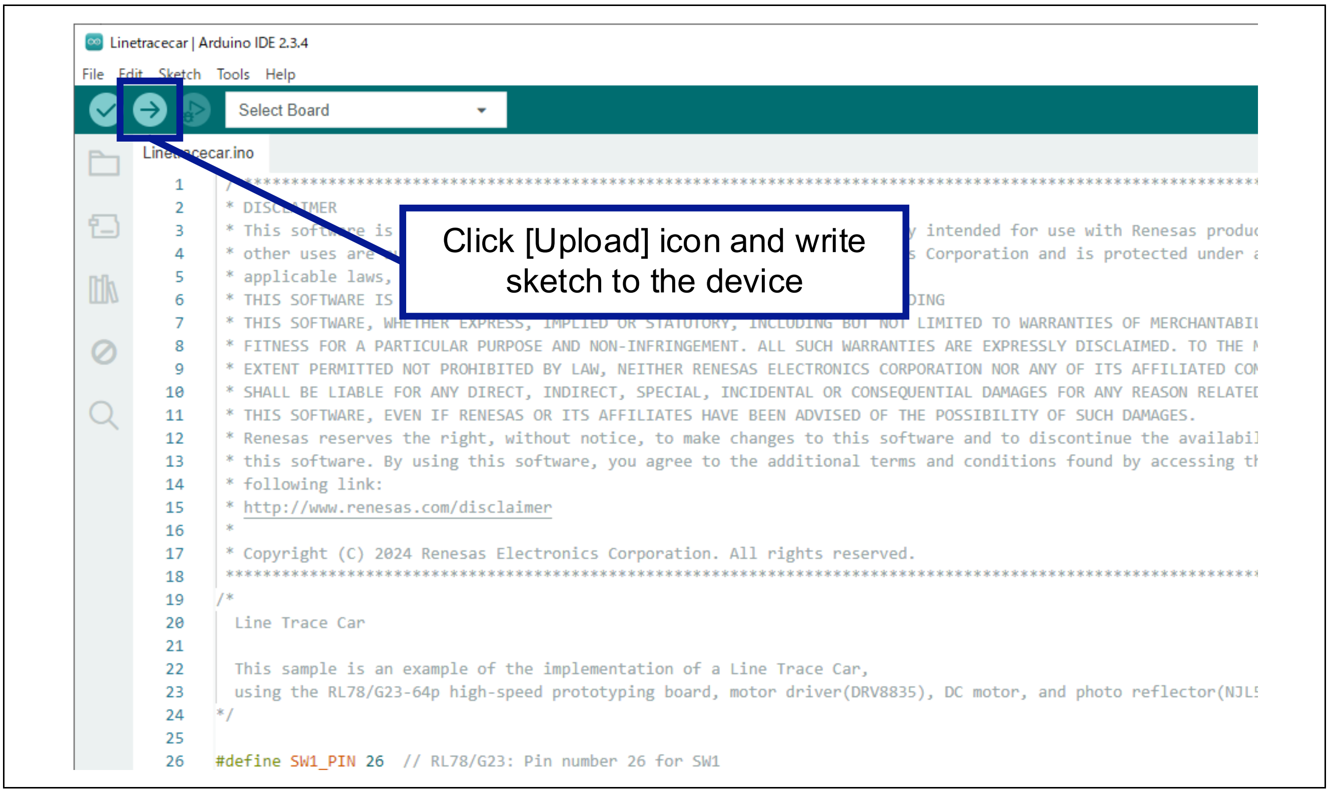
```
/**************************************************************************************************************
 * DISCLAIMER
 * This softwar                                              and is only intended for use with Renesas produc
 * other uses                                                Electronics Corporation and is protected under a
 * applicable l
 * THIS SOFTWAR                                              NTIES REGARDING
 * THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABIL
 * FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT. ALL SUCH WARRANTIES ARE EXPRESSLY DISCLAIMED. TO THE M
 * EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS ELECTRONICS CORPORATION NOR ANY OF ITS AFFILIATED COM
 * SHALL BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES FOR ANY REASON RELATED
 * THIS SOFTWARE, EVEN IF RENESAS OR ITS AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
 * Renesas reserves the right, without notice, to make changes to this software and to discontinue the availabil
 * this software. By using this software, you agree to the additional terms and conditions found by accessing th
 * following link:
 * http://www.renesas.com/disclaimer
 *
 * Copyright (C) 2024 Renesas Electronics Corporation. All rights reserved.
 **************************************************************************************************************
/*
  Line Trace Car

  This sample is an example of the implementation of a Line Trace Car,
  using the RL78/G23-64p high-speed prototyping board, motor driver(DRV8835), DC motor, and photo reflector(NJL5
*/

#define SW1_PIN 26  // RL78/G23: Pin number 26 for SW1
```

3. After compiling is finished, click the [Upload] icon and write sketch to the device.

Figure 4-10 Write Sketch to Device



Click [Upload] icon and write sketch to the device

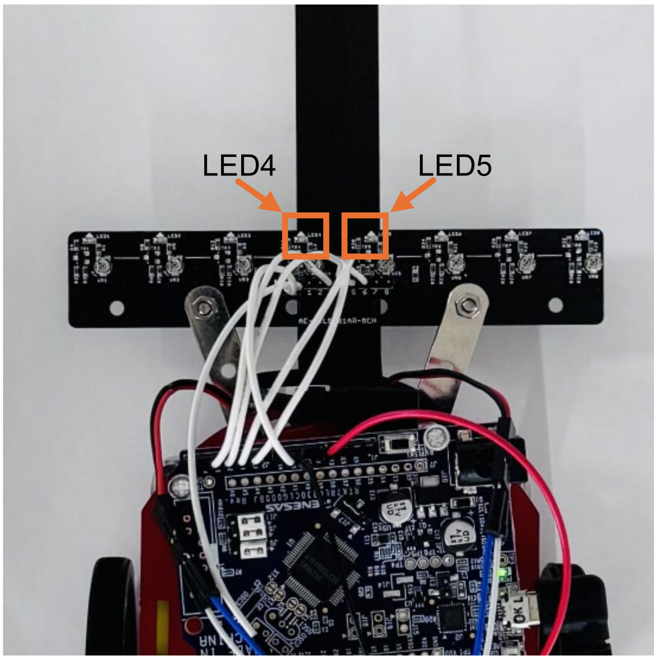
```
/**************************************************************************************************************
 * DISCLAIMER
 * This software is                                          y intended for use with Renesas produc
 * other uses are                                            s Corporation and is protected under a
 * applicable laws,
 * THIS SOFTWARE IS                                          DING
 * THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABIL
 * FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT. ALL SUCH WARRANTIES ARE EXPRESSLY DISCLAIMED. TO THE M
 * EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS ELECTRONICS CORPORATION NOR ANY OF ITS AFFILIATED COM
 * SHALL BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES FOR ANY REASON RELATED
 * THIS SOFTWARE, EVEN IF RENESAS OR ITS AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
 * Renesas reserves the right, without notice, to make changes to this software and to discontinue the availabil
 * this software. By using this software, you agree to the additional terms and conditions found by accessing th
 * following link:
 * http://www.renesas.com/disclaimer
 *
 * Copyright (C) 2024 Renesas Electronics Corporation. All rights reserved.
 **************************************************************************************************************
/*
  Line Trace Car

  This sample is an example of the implementation of a Line Trace Car,
  using the RL78/G23-64p high-speed prototyping board, motor driver(DRV8835), DC motor, and photo reflector(NJL5
*/

#define SW1_PIN 26  // RL78/G23: Pin number 26 for SW1
```

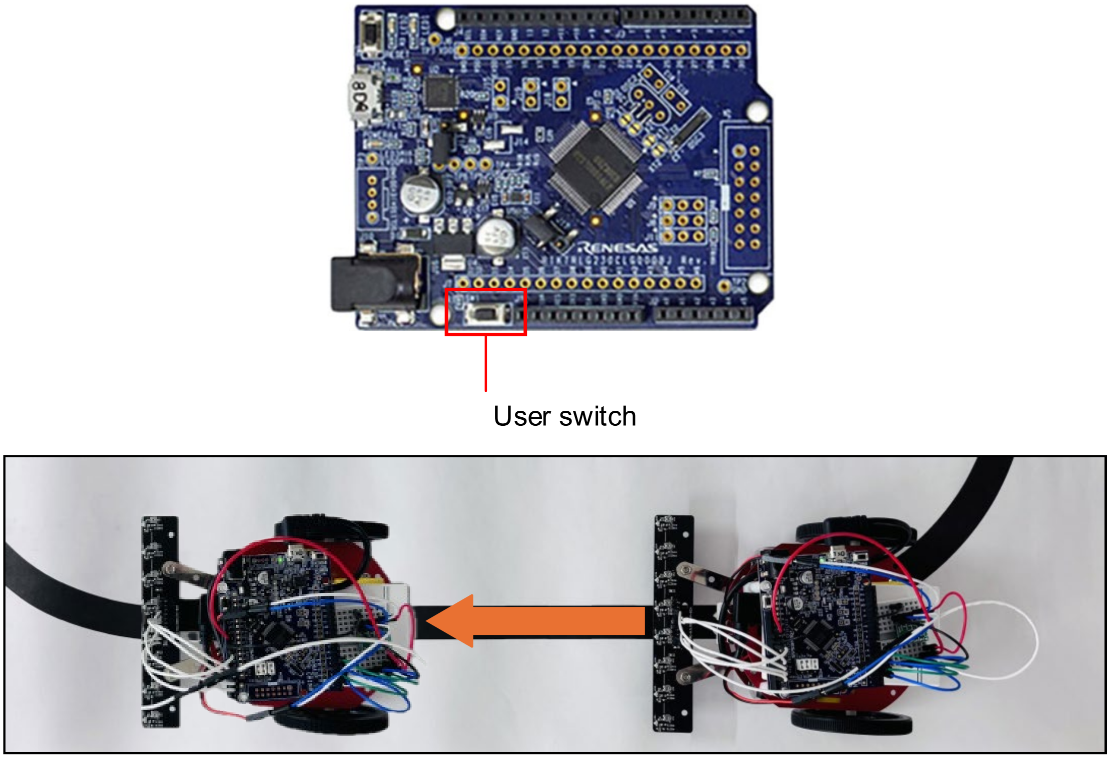
4. After writing is finished, place the line trace car over the black line of the course. At this time, adjust the position so that LED4 or LED5 of the sensor can recognize the black line.

Figure 4-11 Installation Diagram Examples



5. After placing, switch on the mobile battery or external power supply. This operation will start up the MCU, but the line trace car will stand by when it is stopped.
Then, when the user switch is pressed, the line trace car starts operation.

Figure 4-12 Line Trace Car Operation Start Method and Operation Image



6. If you want to finish the operation, press the user switch again. This operation stops the line trace car. Then, power off the mobile battery to stop the MCU operation.

4.3.1 Threshold Setting of Sample Sketch and Speed Change of Motor

In this sketch, several constants are set to adjust the behavior of the line trace car. The **threshold** is the reference value that the sensor distinguishes between white and black. Set this value using the serial plotter while checking the values that the sensor detects white and black.

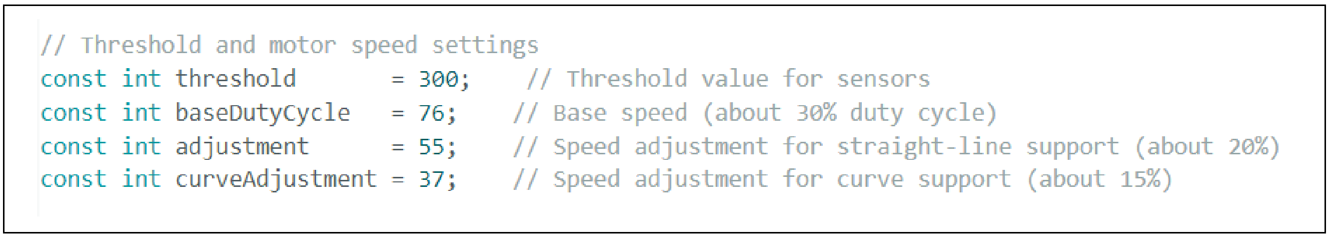
The **baseDutyCycle** controls the base velocity of the car and determines the duty cycle of PWM. In this sketch, 0 means 0% (stop) and 255 means 100% (maximum speed). Increasing the value increases the speed. However, note that increasing the duty ratio of PWM too much may cause the operation to be stopped due to insufficient current in the mobile battery.

The **adjustment** is used to control one of the motor speeds during linear auxiliary operation. If one sensor is out of the line, the other motor speed is set to this value to compensate. If this value is increased, the direction can be corrected quickly but linear operation may become unstable. Lower values will make the movement softer but may delay the correction and cause a course out.

The **curveAdjustment** is used to set the velocity for decelerating one of the motors when the curve is detected. In the curve, one motor runs at **baseDutyCycle** velocity and the other runs almost at standstill. The rate at which this “almost stopped condition” is realized is set by **curveAdjustment**. Lower values result in subtle movement rather than near stop, while higher values result in movement close to complete stop.

Therefore, **adjustment** and **curveAdjustment** must be set to a value that is slower than **baseDutyCycle**. If this is not observed, the speed difference between the motors does not occur properly, and operation does not follow the black line. When adjusting these values, set them while checking the actual operation.

Figure 4-13 Photo Reflector Threshold Setting Constant and Motor Speed Setting Constant



```
// Threshold and motor speed settings
const int threshold       = 300;     // Threshold value for sensors
const int baseDutyCycle   = 76;     // Base speed (about 30% duty cycle)
const int adjustment      = 55;     // Speed adjustment for straight-line support (about 20%)
const int curveAdjustment = 37;     // Speed adjustment for curve support (about 15%)
```

4.3.2 Control of Backward Movement

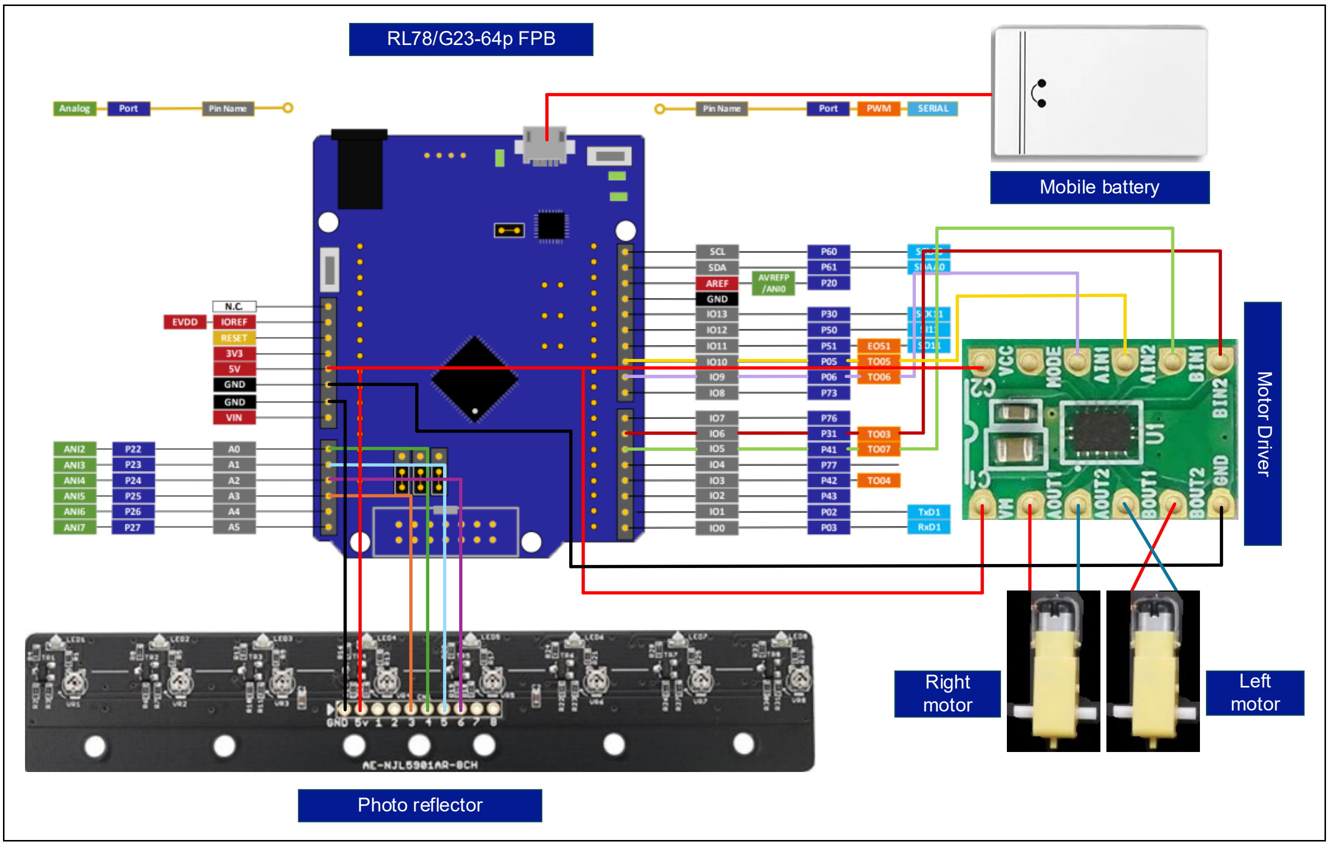
In this sample sketch, the backward control of the line trace car is not described as a specification, but it can be realized by correcting the wiring.

To perform backward control, first swap the motor driver AIN1 and AIN2, BIN1 and BIN2 wires. The motor rotation direction can be changed by changing wires.

DRV8835 motor driver is used in this sample sketch. Other motor drivers may differ in how they control ANI2 and BIN2 and in how they interpret PWM. Therefore, check in advance the specifications and data sheets of the motor driver to see if they match the behavior. If the specifications are different, the program must be modified accordingly.

Figure 4-14 shows the difference between the forward control and backward control wiring in the motor driver.

Figure 4-14 Differences between Forward Control and Backward Control Specifications

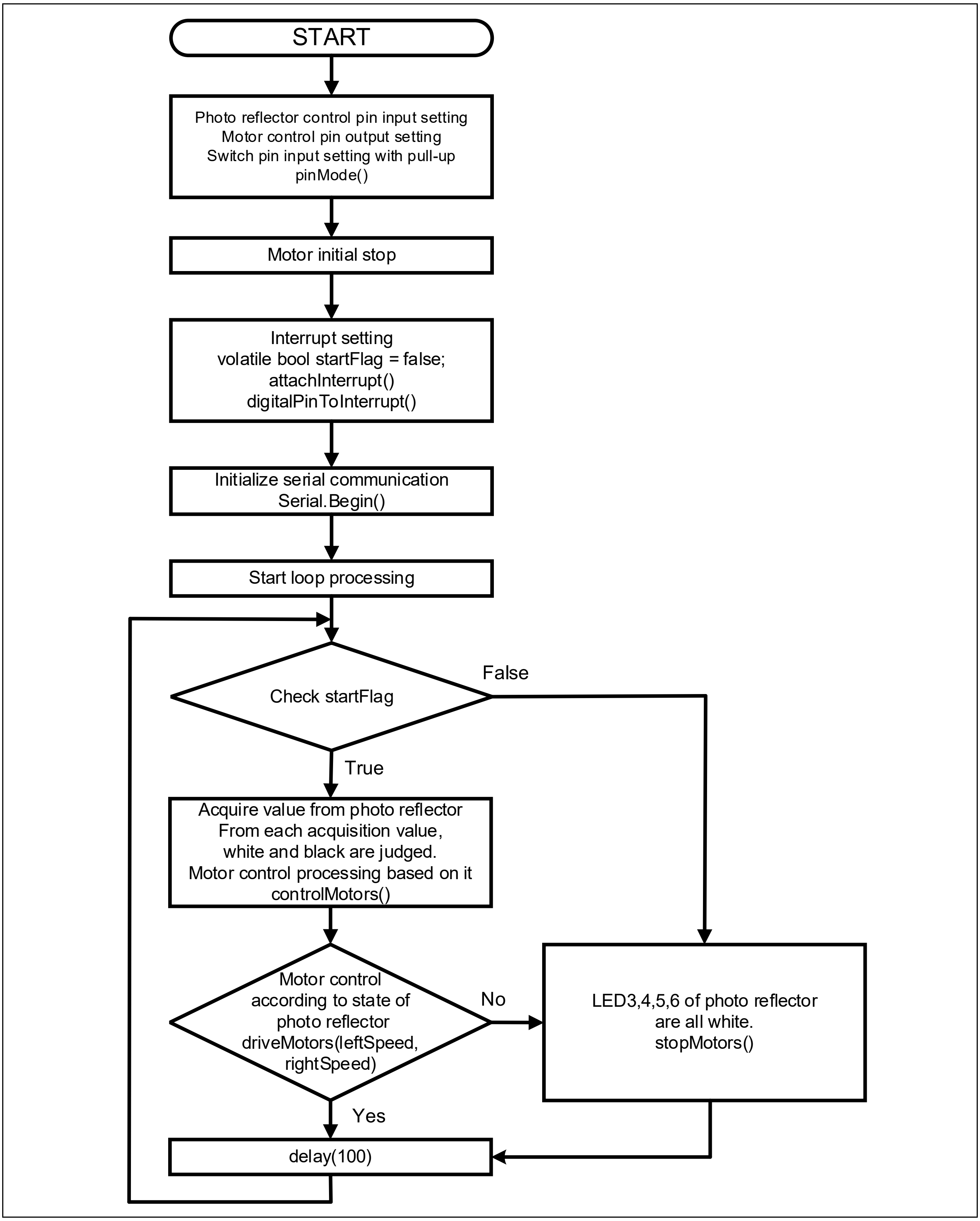


4.4 Flowchart

4.4.1 Main Processing

The following shows the flow of the sample sketch

Figure 4-15 Flowchart of Main Processing

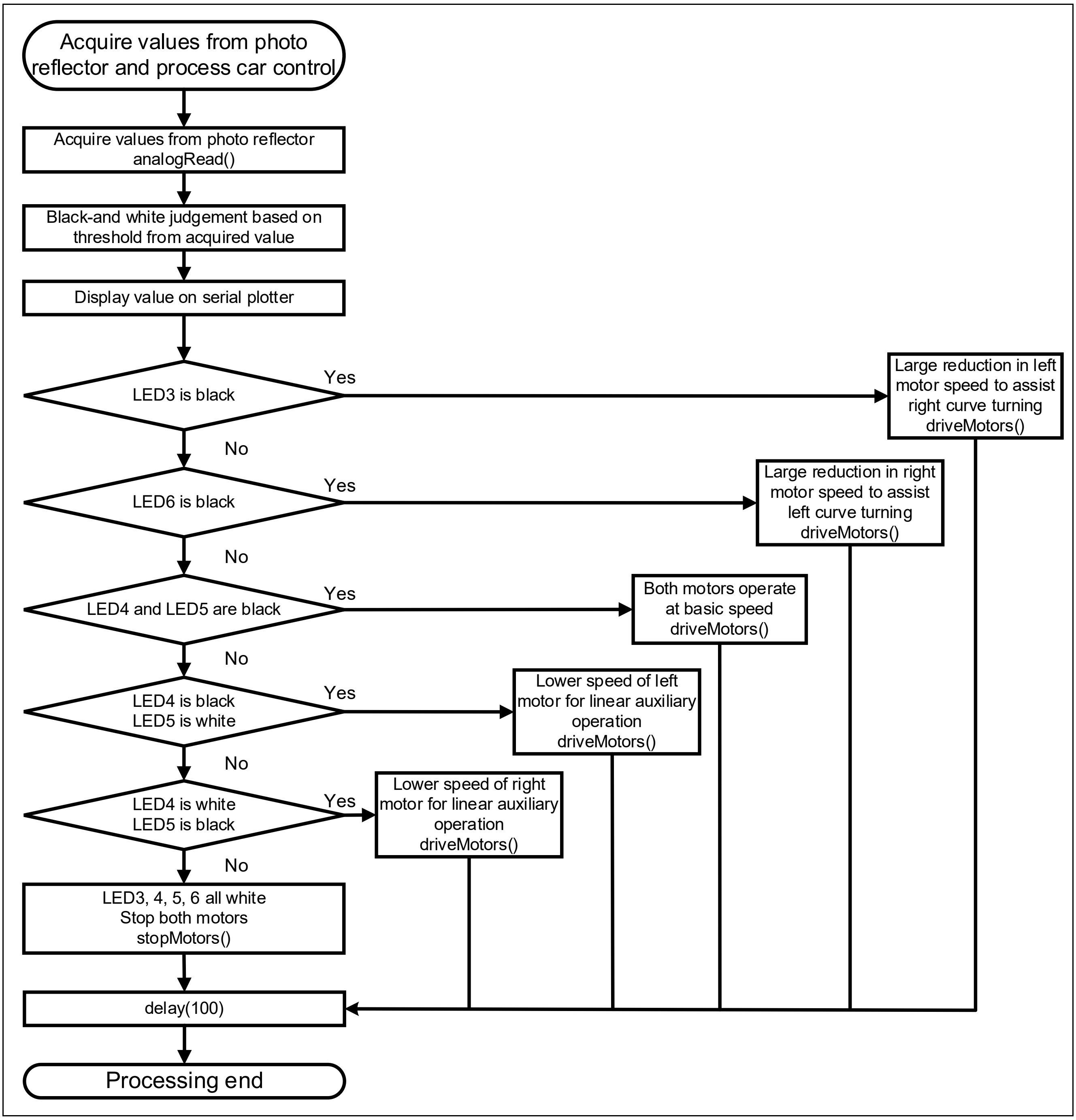

START
Photo reflector control pin input setting
Motor control pin output setting
Switch pin input setting with pull-up
pinMode()
Motor initial stop
Interrupt setting
volatile bool startFlag = false;
attachInterrupt()
digitalPinToInterrupt()
Initialize serial communication
Serial.Begin()
Start loop processing
Check startFlag
False
True
Acquire value from photo reflector
From each acquisition value,
white and black are judged.
Motor control processing based on it
controlMotors()
Motor control
according to state of
photo reflector
driveMotors(leftSpeed,
rightSpeed)
No
LED3,4,5,6 of photo reflector
are all white.
stopMotors()
Yes
delay(100)

4.4.2 Called Function Processing Flowchart

The following shows the flow of functions called from the loop function.

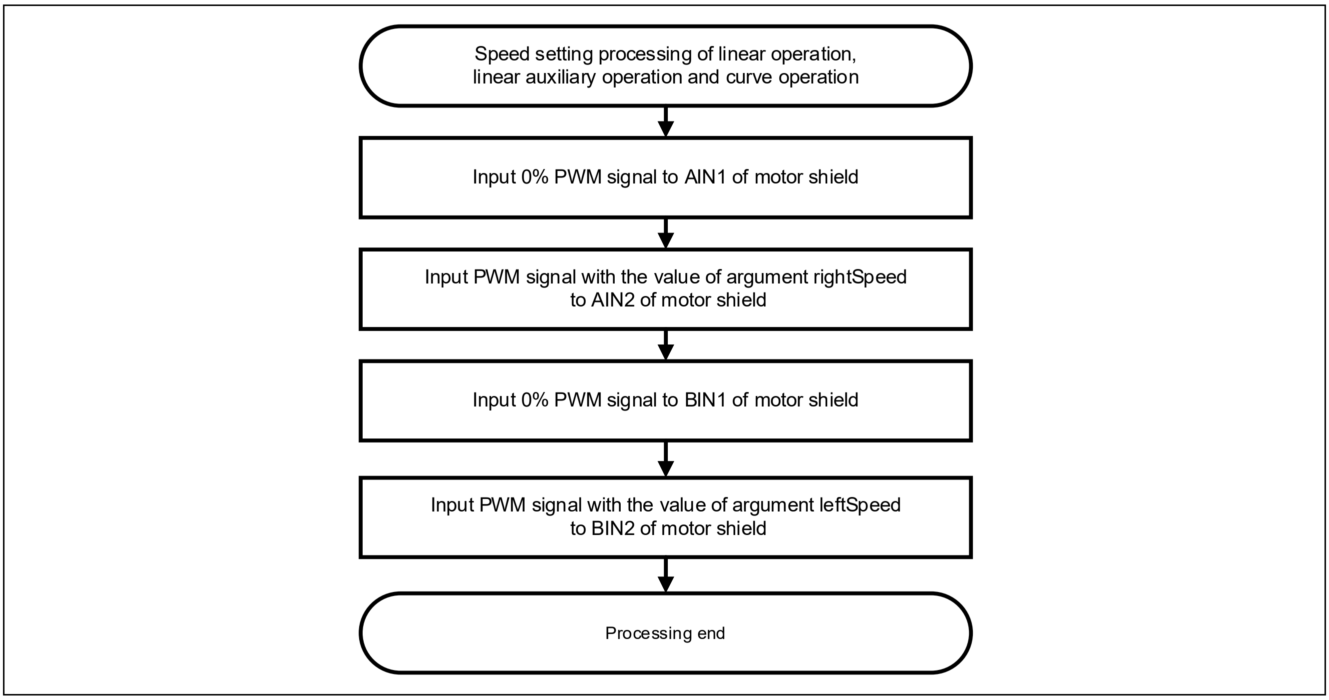
(1) Acquisition of values from the photo reflector and body control processing: controlMotors function

Figure 4-16 Flowchart of controlMotors Function


Acquire values from photo reflector and process car control
Acquire values from photo reflector
analogRead()
Black-and white judgement based on threshold from acquired value
Display value on serial plotter
LED3 is black
Yes
No
Large reduction in left motor speed to assist right curve turning
driveMotors()
LED6 is black
Yes
No
Large reduction in right motor speed to assist left curve turning
driveMotors()
LED4 and LED5 are black
Yes
No
Both motors operate at basic speed
driveMotors()
LED4 is black
LED5 is white
Yes
No
Lower speed of left motor for linear auxiliary operation
driveMotors()
LED4 is white
LED5 is black
Yes
No
Lower speed of right motor for linear auxiliary operation
driveMotors()
LED3, 4, 5, 6 all white
Stop both motors
stopMotors()
delay(100)
Processing end

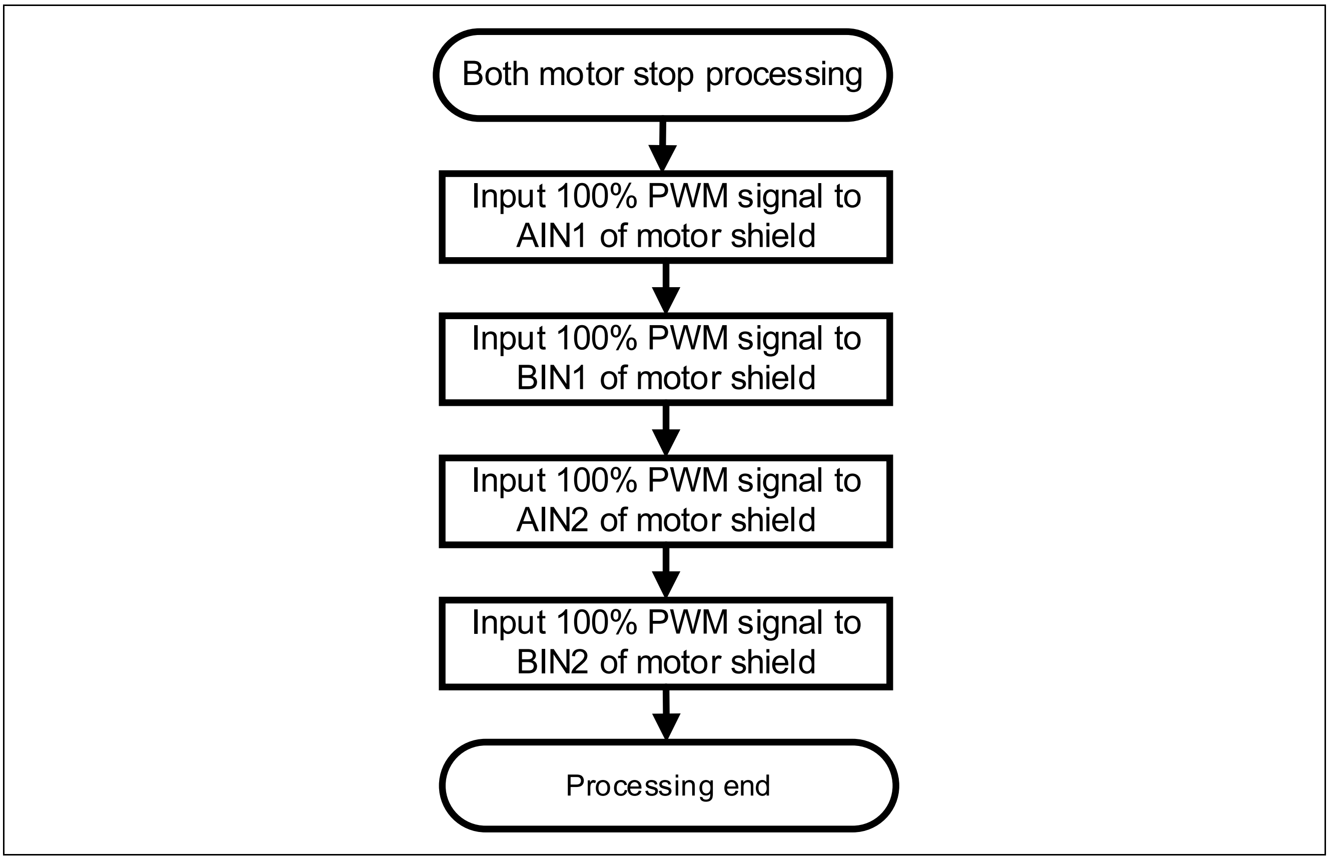
(2) Speed setting process for linear, linear auxiliary and curve operation: driveMotors function

Figure 4-17 Flowchart of driveMotors Function


Speed setting processing of linear operation, linear auxiliary operation and curve operation
Input 0% PWM signal to AIN1 of motor shield
Input PWM signal with the value of argument rightSpeed to AIN2 of motor shield
Input 0% PWM signal to BIN1 of motor shield
Input PWM signal with the value of argument leftSpeed to BIN2 of motor shield
Processing end

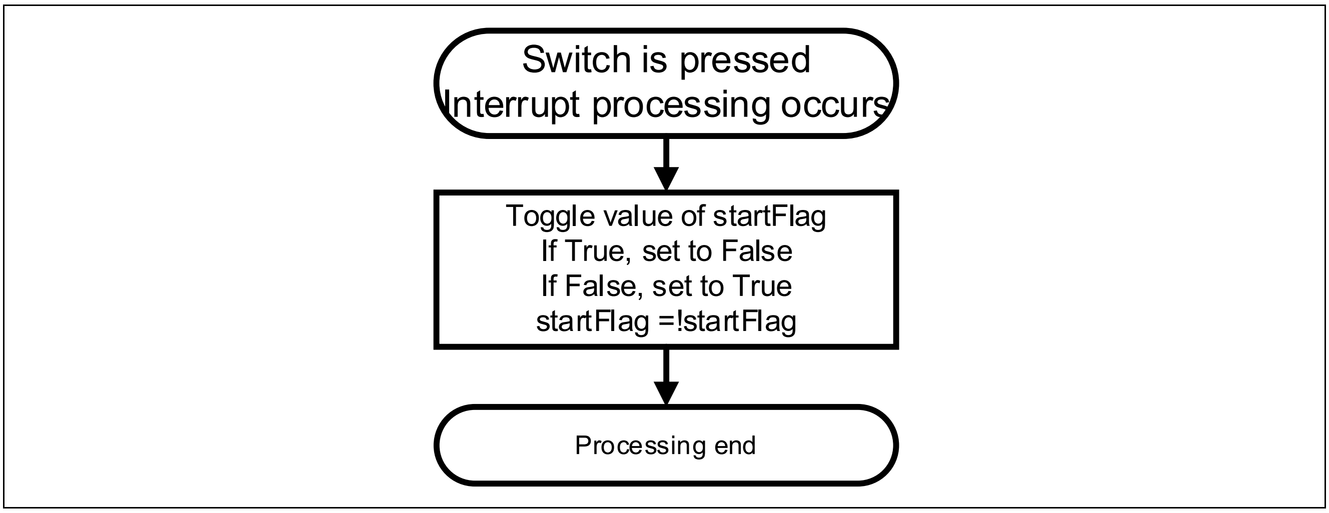
(3) Both motor processes: stopMotors function

Figure 4-18 Flowchart of stopMotors Function


Both motor stop processing
Input 100% PWM signal to AIN1 of motor shield
Input 100% PWM signal to BIN1 of motor shield
Input 100% PWM signal to AIN2 of motor shield
Input 100% PWM signal to BIN2 of motor shield
Processing end

(4) The process of toggling startFlag when an interrupt pccurs: toggleStartFlag function

Figure 4-19 Flowchart of toggleStartFlag Function



4.4.3 Specification of Functions

The details of the calling function are shown below.

void controlMotors()

Outline This function controls the motor operation based on the value acquired from each sensor. It instructs the motor to move on a curve or in a straight line according to the state detected by the sensor.

Argument None

Return value None

void driveMotors(int leftSpeed, int rightSpeed)

Outline This function operates the motor with specified speed.
It can specify the speed of the left motor and the right motor individually.

Argument leftSpeed Speed of the left motor (PWM value, 0～255)

Return value rightSpeed Speed of the right motor (PWM value, 0～255)

void stopMotors()

Outline This function stops both motors.
It sets the PWM signal to stop state (255) and stops the motor rotation.

Argument None

Return value None

void toggleStartFlag()

Outline This function is called when an interrupt occurs and toggles startFlag (switches True/False). It outputs the status after switching to the serial monitor.

Argument None

Return value None

5. Notes

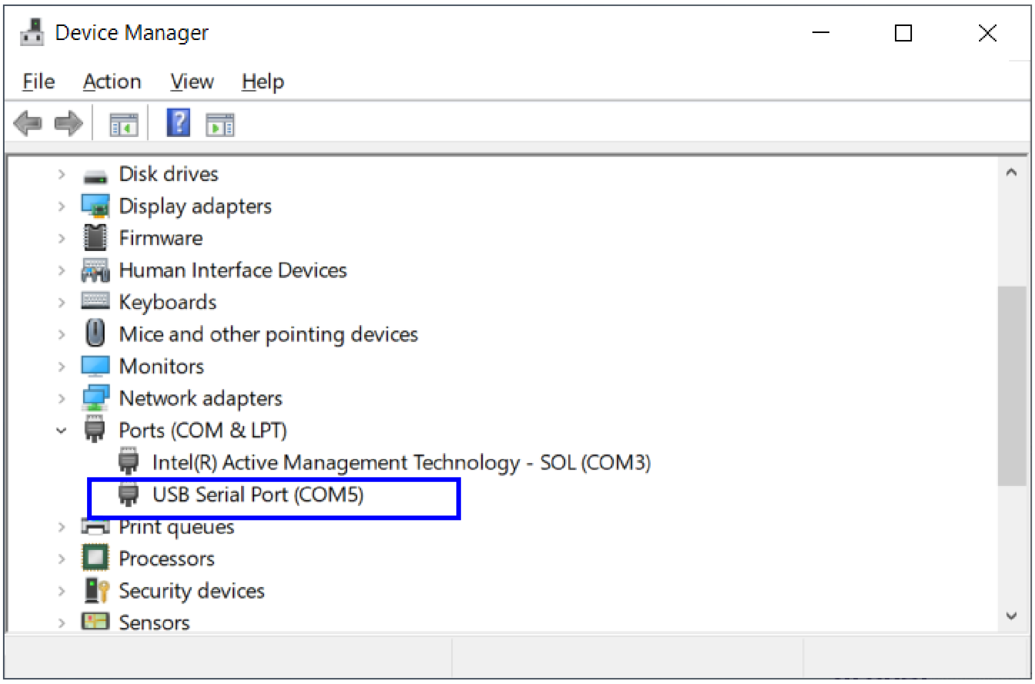
5.1 COM Port is not Displayed on the Windows Device Manager

When connecting the PC and the evaluation board (RL78/G23-64p FPB) for the first time, the PC may not recognize the port, and the COM port may not be displayed in Windows Device Manager.

If the COM port is not displayed, install the driver of the USB-to-serial converter (FT232RQ) from FTDI on the RL78/G23-64p FPB by the following procedure.

1. Download the latest driver installer for the target OS from FTDI's website and install it.
   https://ftdichip.com/drivers/vcp-drivers/
2. After installation, “USB Serial Port (COMx)” is displayed under the “Ports (COM & LPT)” on the Device Manager. In the following figure, COM5 is the target COM port.

Figure 5-1 Windows Device Manager after Installation of Device Driver



- Disk drives
- Display adapters
- Firmware
- Human Interface Devices
- Keyboards
- Mice and other pointing devices
- Monitors
- Network adapters
- Ports (COM & LPT)
  - Intel(R) Active Management Technology - SOL (COM3)
  - USB Serial Port (COM5)
- Print queues
- Processors
- Security devices
- Sensors

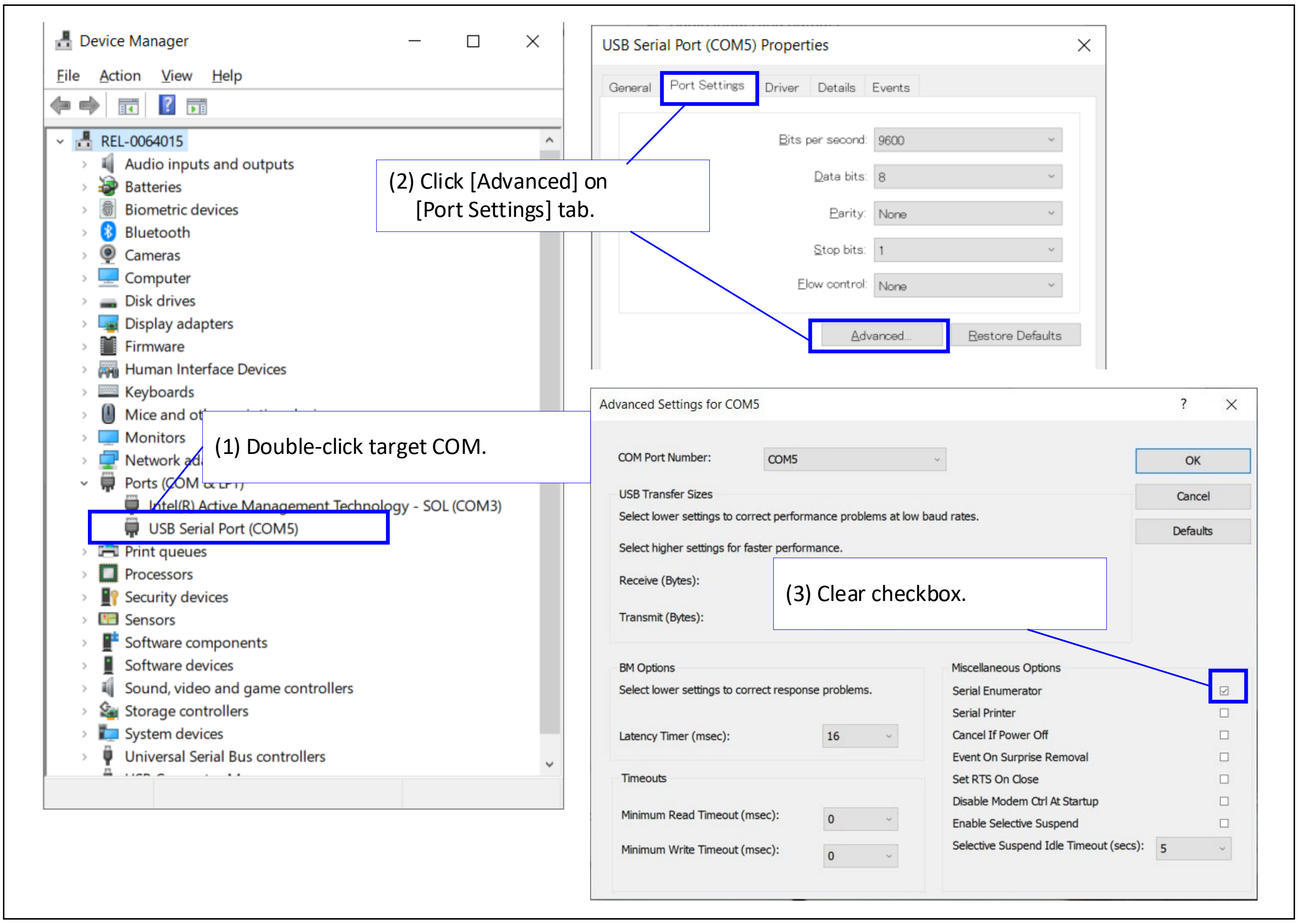
For details of USB-to-serial convector and COM port, refer to “5.11 USB-to-Serial Converter” and “5.12 USB-to-Serial Converter Reset Header” in RL78/G23-64p Fast Prototyping Board User’s manual.

5.2 Program is not Written Correctly to RL78/G23-64p Fast Prototyping Board

When writing to the MCU board, it may not be connected correctly.

If the program is not written correctly, double-click the target COM port on Windows Device Manager and clear the checkbox of [Serial Emulator].

Figure 5-2 Setting Example of Target COM



(1) Double-click target COM.

(2) Click [Advanced] on [Port Settings] tab.

(3) Clear checkbox.

Device Manager

REL-0064015
- Audio inputs and outputs
- Batteries
- Biometric devices
- Bluetooth
- Cameras
- Computer
- Disk drives
- Display adapters
- Firmware
- Human Interface Devices
- Keyboards
- Mice and ot
- Monitors
- Network a
- Ports (COM & LPT)
  - Intel(R) Active Management Technology - SOL (COM3)
  - USB Serial Port (COM5)
- Print queues
- Processors
- Security devices
- Sensors
- Software components
- Software devices
- Sound, video and game controllers
- Storage controllers
- System devices
- Universal Serial Bus controllers

USB Serial Port (COM5) Properties

General | Port Settings | Driver | Details | Events

Bits per second: 9600
Data bits: 8
Parity: None
Stop bits: 1
Flow control: None

Advanced... | Restore Defaults

Advanced Settings for COM5

COM Port Number: COM5

USB Transfer Sizes
Select lower settings to correct performance problems at low baud rates.
Select higher settings for faster performance.
Receive (Bytes):
Transmit (Bytes):

BM Options
Select lower settings to correct response problems.
Latency Timer (msec): 16

Timeouts
Minimum Read Timeout (msec): 0
Minimum Write Timeout (msec): 0

Miscellaneous Options
Serial Enumerator ☑
Serial Printer □
Cancel If Power Off □
Event On Surprise Removal □
Set RTS On Close □
Disable Modem Ctrl At Startup □
Enable Selective Suspend □
Selective Suspend Idle Timeout (secs): 5

OK | Cancel | Defaults

5.3 After the Power Supply by Mobile Battery is Started, It Immediately Stops

It is possible to use the mobile battery with the auto power-off function.

When powering a board that operates on a weak current like that of an FPB with a standard mobile battery, the protection function (auto power-off function) may activate, and stop the power supply. Therefore, please use a mobile battery that does not automatically turn off.

6. Sample Code

There is the sample code for this application note.

Sample code can be downloaded from the Renesas Electronics website.

7. Reference Documents

RL78/G23 User's Manual: Hardware (R01UH0896)

RL78/G23-64p Fast Prototyping Board User's Manual (R20UT4814)

The latest versions can be downloaded from the Renesas Electronics website.

Technical update

The latest versions can be downloaded from the Renesas Electronics website.

All trademarks and registered trademarks are the property of their respective owners.

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Feb.25.25	-	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 be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. 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.
6. 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.
7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION (“Vulnerability Issues”). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
8. 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.
9. 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.
10. 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.
11. 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.
12. 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.
13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
14. 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.5.0-1 October 2020)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
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/.