

## RL78/G23

### ライントレースカー サンプルスケッチ (Arduino™ スケッチ)

#### 要旨

本アプリケーションノートでは、RL78/G23-64p Fast Prototyping Board (FPB) 用 Arduino ライブラリを用いて、ライントレースカーの制御方法を説明します。

#### 動作確認デバイス

評価ボード	: RL78/G23-64p Fast Prototyping Board
モータドライバ	: AE-DRV8835-S
フォトリフレクタ	: AE-NJL5901AR-8CH
モバイルバッテリー	: CHE-061-WH-IOT2
DC モータ(ライントレースカーキットの付属品)	: 2WD Mini Smart Robot Mobile Platform Kit for education

#### 商標・他社 TM

Arduino は Arduino SA の商標です。

## 目次

1. システム概要	3
1.1 使用モジュール	4
1.1.1 モータドライバ搭載モジュール	5
1.1.2 フォトリフレクタ搭載モジュール	5
1.2 動作説明	6
2. 動作確認環境	7
3. 開発環境構築	8
3.1 ボードの接続	8
3.2 使用端子一覧	9
3.3 Arduino™ IDE のセットアップ	10
4. ソフトウェア説明	13
4.1 サンプルコードの概要	13
4.1.1 ライントレースカーのアルゴリズム	14
4.1.2 フォトリフレクタの閾値設定方法	17
4.2 API 関数	19
4.3 サンプルスケッチの動作確認手順	20
4.3.1 サンプルスケッチの閾値設定とモータのスピード変更について	24
4.3.2 後退の制御	25
4.4 フローチャート	26
4.4.1 メイン処理フローチャート	26
4.4.2 呼び出し関数処理フローチャート	27
4.4.3 関数詳細	29
5. 注意事項	30
5.1 Windows デバイスマネージャーに COM ポートが表示されない	30
5.2 RL78/G23-64p Fast Prototyping Board に正しく書き込みができない	31
5.3 モバイルバッテリーで電源供給開始後、すぐに電源供給を停止してしまう	31
6. サンプルコード	32
7. 参考ドキュメント	32
改訂記録	33

## 1. システム概要

本システムは、RL78/G23-64p Fast Prototyping Board、モータドライバ、フォトリフレクタ、DC モータで構成されており、オーバルコース上の黒線に沿って車両を走行させることを目的としています。プログラムの作成と書き込みには Arduino™ IDE を使用します。

動作の概要として、FPB の電源投入後、フォトリフレクタがオーバルコース上の黒線と白地を判別し、黒線に沿って走行します。フォトリフレクタから得られる情報に基づき、プログラムは PWM 信号を生成し、モータドライバを介して 2 つの DC モータの方向と速度を制御します。これにより、車両は黒線に沿って走行します。

本システムで使用するサンプルコードのブロック構成、動作イメージを以下に示します。

図 1-1 ソフトウェアブロック図

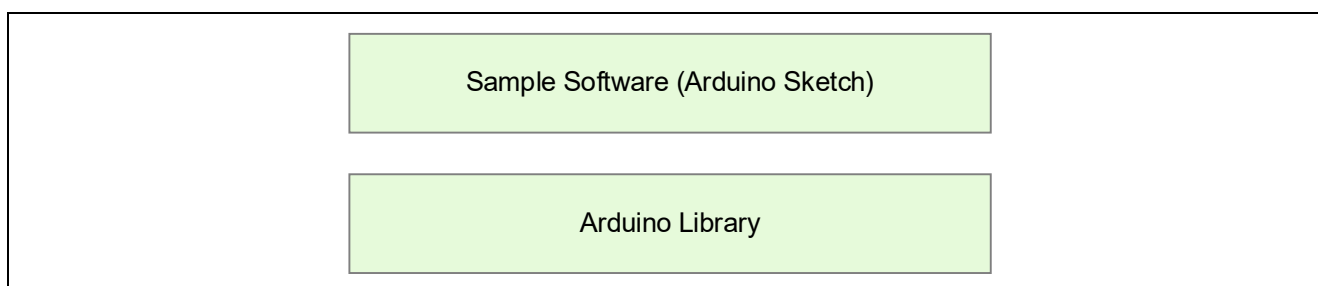
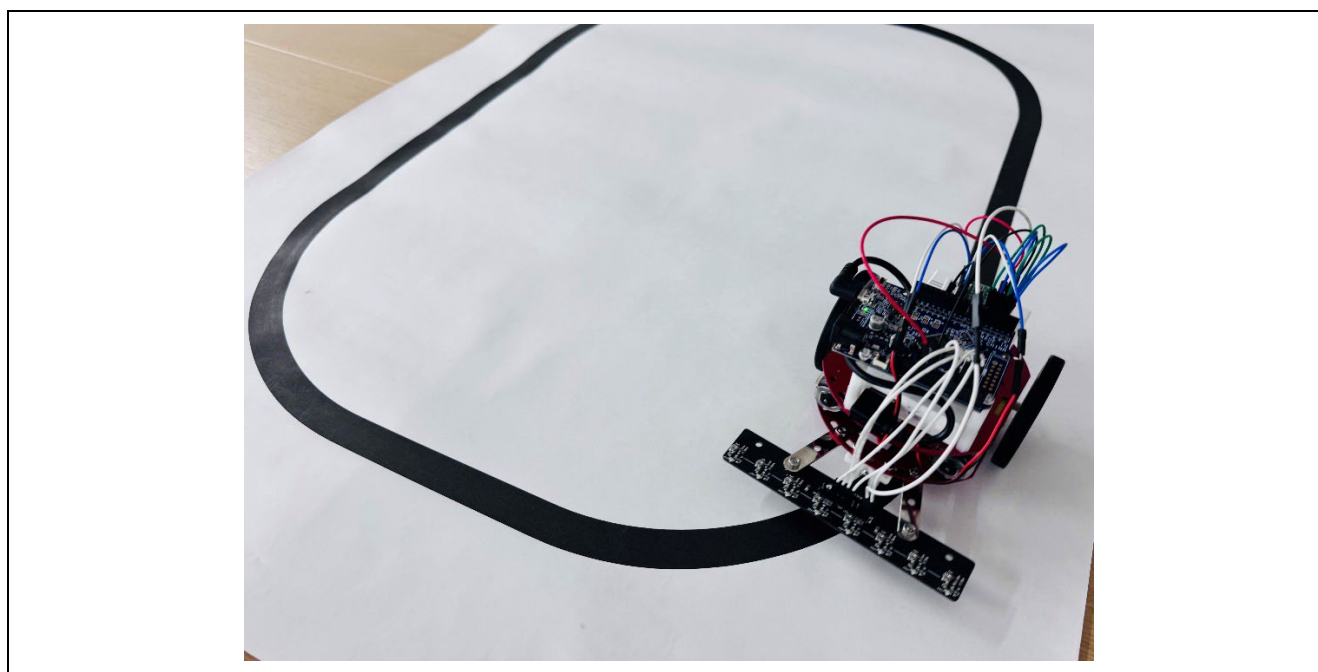


図 1-2 動作イメージ



## 1.1 使用モジュール

開発時のシステム構成では、電源をパソコンから供給し、フォトリフレクタの出力値や書き込んだプログラムの動作を確認します。

動作時のシステム構成では、電源をモバイルバッテリーに切り替え、ライントレースカーをコース上で走行させます。

本システムの開発時とライントレースカー動作時のシステム概略図を以下に示します。

図 1-3 開発時のシステム構成図

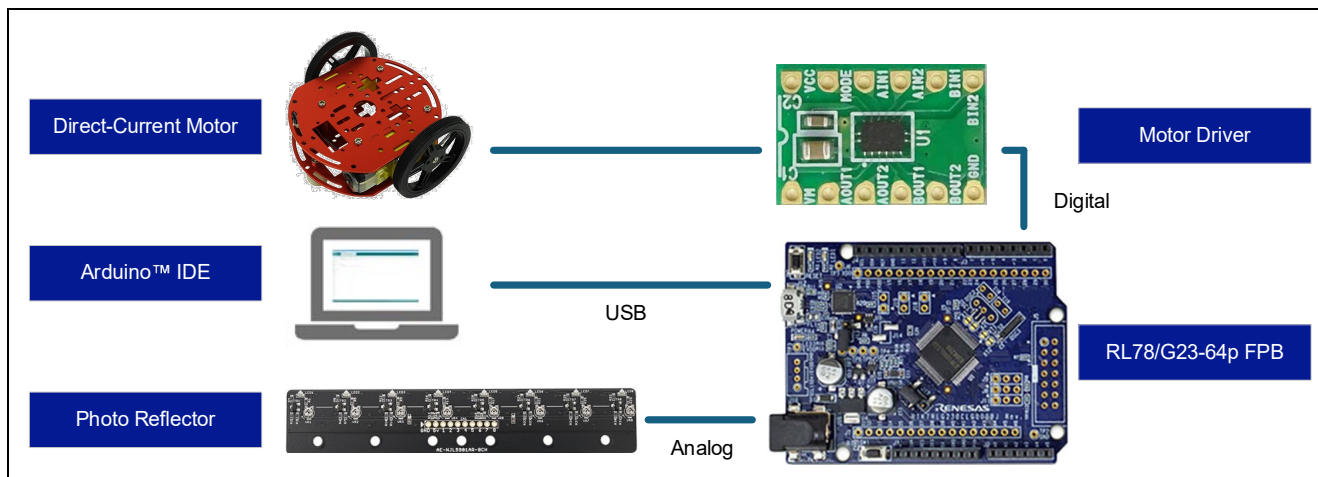
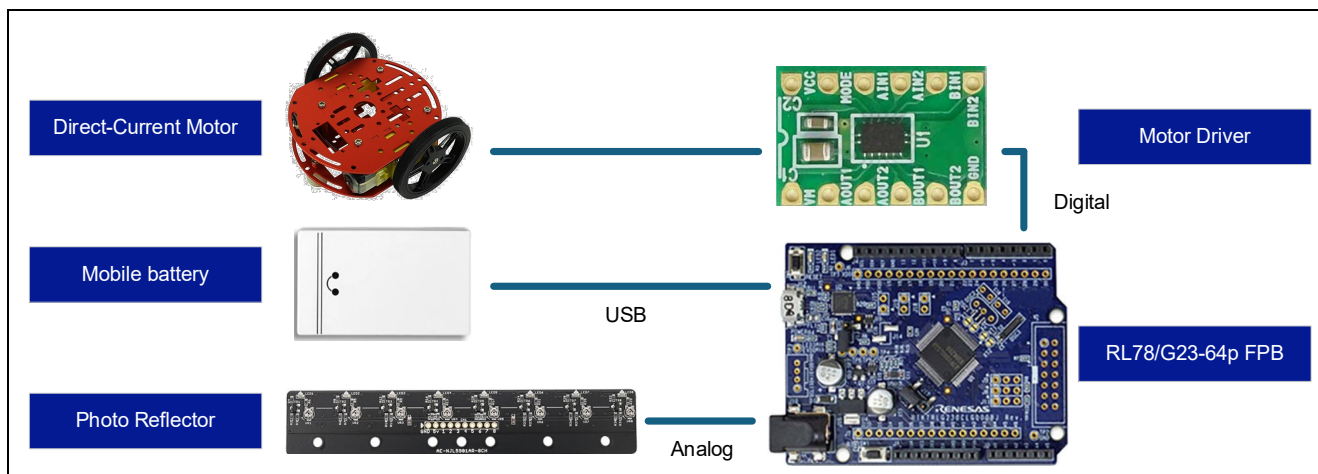


図 1-4 ライントレースカー動作時のシステム構成図



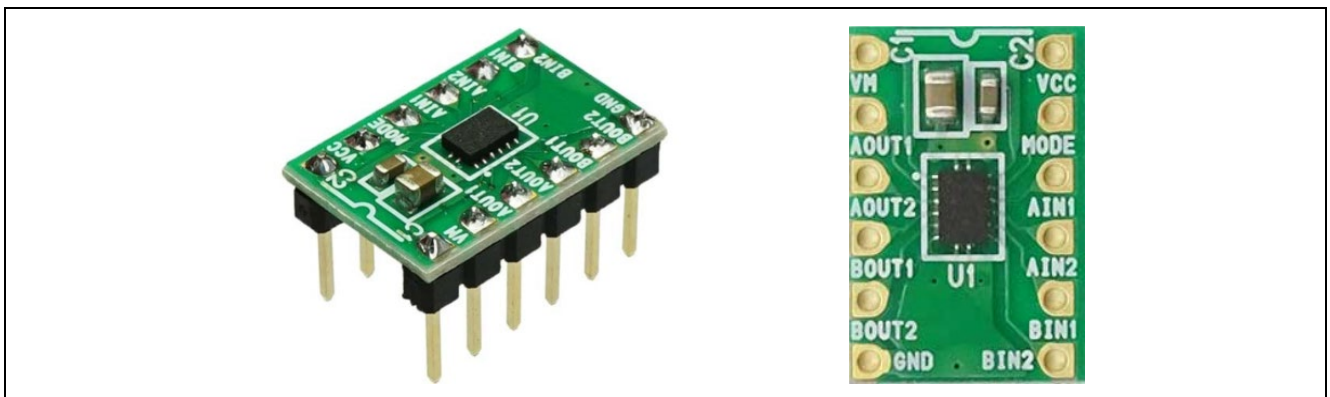
### 1.1.1 モータドライバ搭載モジュール

モータドライバは、モータの回転速度や回転方向を制御するための回路です。本システムでは DRV8835 搭載のモジュールを使用し、1 個のモータに対して 2 本の PWM 信号で回転速度と正転・逆転の方向を制御する IN/IN モードの回路を左右のモータ用にそれぞれ構成しています。

本アプリケーションノートでのライントレースカーは、モータドライバの逆転制御で前進し、正転制御で後退します。本サンプルプログラムでは、モータドライバの逆転制御で前進するように設定されています。DRV8835 の詳細な仕様については、データシートを参照してください。

図 1-5 に、本システムで使用するモータドライバ搭載モジュールを示します。

図 1-5 モータドライバ搭載モジュール

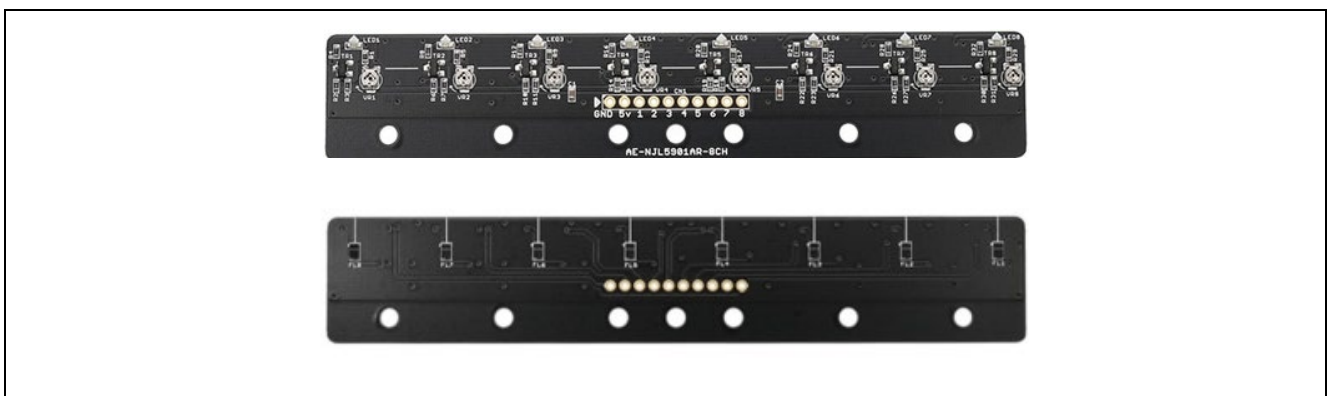


### 1.1.2 フォトリフレクタ搭載モジュール

フォトリフレクタは、物体からの反射光を検出し物体の色や距離を判定するセンサで、ライン検出や物体の位置把握に使用します。本システムでは、NJG5901AR フォトリフレクタを使用し、ライン上の黒と白を判定するために活用しています。フォトリフレクタは赤外線 LED を照射し、その反射光を受光素子で検出することで、物体の反射率を判別する仕組みから、黒を検出しているときに出力値が高く、白を検出しているときに出力値が低くなります。これは、黒い面が光を吸収しやすく反射が弱いため、内部のプルアップ回路によって出力が高くなる一方、白い面は光を強く反射し、受光素子によって電流が流れやすくなることで出力値が低くなるためです。この特性を利用して、サンプルスケッチでは、フォトリフレクタからのアナログ信号を RL78/G23 の ADC で読み取り、ライントレースカー走行時、オーバルコース上の黒線と白地を判定しています。フォトリフレクタの詳細な仕様については、データシートを参照してください。

図 1-6 に本システムで使用するフォトリフレクタ搭載モジュールを示します。

図 1-6 フォトリフレクタ搭載モジュール



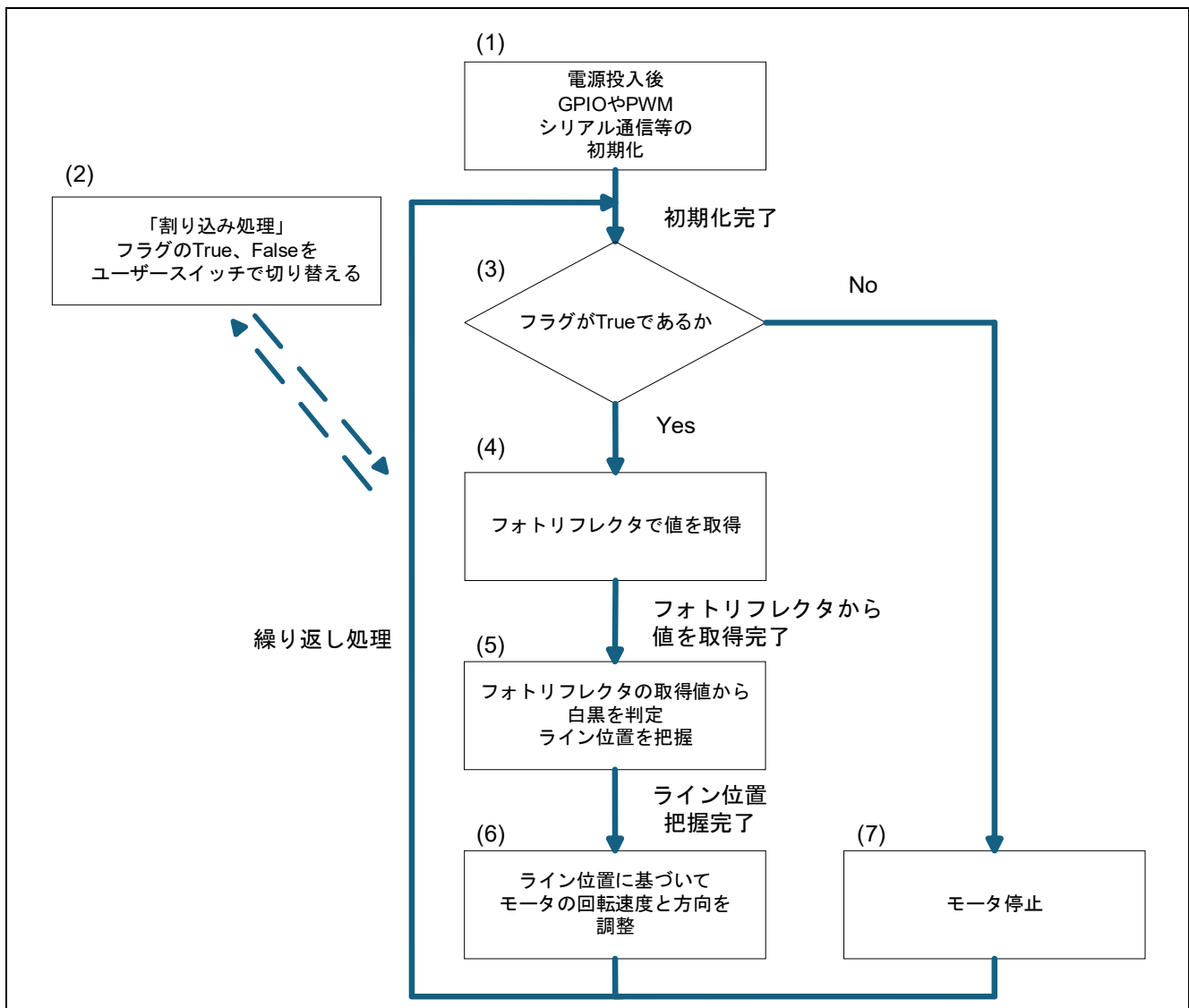
## 1.2 動作説明

図 1-7 に動作概要を示します。

- (1) 電源投入後、GPIO や PWM、シリアル通信等の初期設定を行う。
- (2) ユーザースイッチの押下によって、フラグの ON、OFF を切り替える割り込み処理を実行。
- (3) フラグが True であれば(4)を実行。False であれば(7)を実行。
- (4) フォトリフレクタが反射光を検出し、各センサの出力値を ADC で取得。
- (5) 各フォトリフレクタの取得値から白と黒を判定し、その結果を基にラインの位置を把握。
- (6) ライン位置に基づいてモータの回転速度と方向を調整し、モータを動作。
- (7) フラグが OFF の場合のモータ停止。

備考. ステップ(3)~(6)の処理は、ライントレースカーの電源がオンになっている間、繰り返します。  
 (4)のフォトリフレクタの取得した値は、開発時に Arduino™ IDE のシリアルプロッタ上で確認することが可能。センサの閾値設定や動作確認が可能です。

図 1-7 動作概要



## 2. 動作確認環境

本システムの動作確認環境は、以下のとおりです。

表 2-1 動作確認環境（ハードウェア）

項目	内容
評価ボード	RL78/G23-64p Fast Prototyping Board – RTK7RLG230CLG000BJ
モータドライバ	AE-DRV8835-S
フォトリフレクタ	AE-NJL5901AR-8CH
DC モータ(ライントレースカーキットの付属品)	2WD Mini Smart Robot Mobile Platform Kit for education
モバイルバッテリー	CHE-061-WH-IOT2 <sup>注1</sup>
動作電圧	5V

注 1. オートパワーオフ機能がついているモバイルバッテリーは、消費電力が小さいシステムで一定時間使用すると電源供給が止まります。そのためオートパワーオフ機能が無効のモバイルバッテリーを使用してください。

表 2-2 動作確認環境（ソフトウェア）

項目	内容	バージョン
OS	Windows 10 Pro	-
統合開発環境（IDE）	Arduino™ IDE	2.3.4
標準ライブラリ	RL78/G23-64p FPB ライブラリ	2.4.0

### 3. 開発環境構築

ボードの接続方法と Arduino™ IDE のセットアップを説明します。

本システムでは Arduino™ IDE 2.3.4 を使用しています。Arduino™ IDE 2.3.4 以降をインストールしていない場合は、インストールしてください。

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

#### 3.1 ボードの接続

図 3-1 に評価ボードとモータドライバ搭載モジュール、フォトフレクタ搭載モジュールの接続を示します。

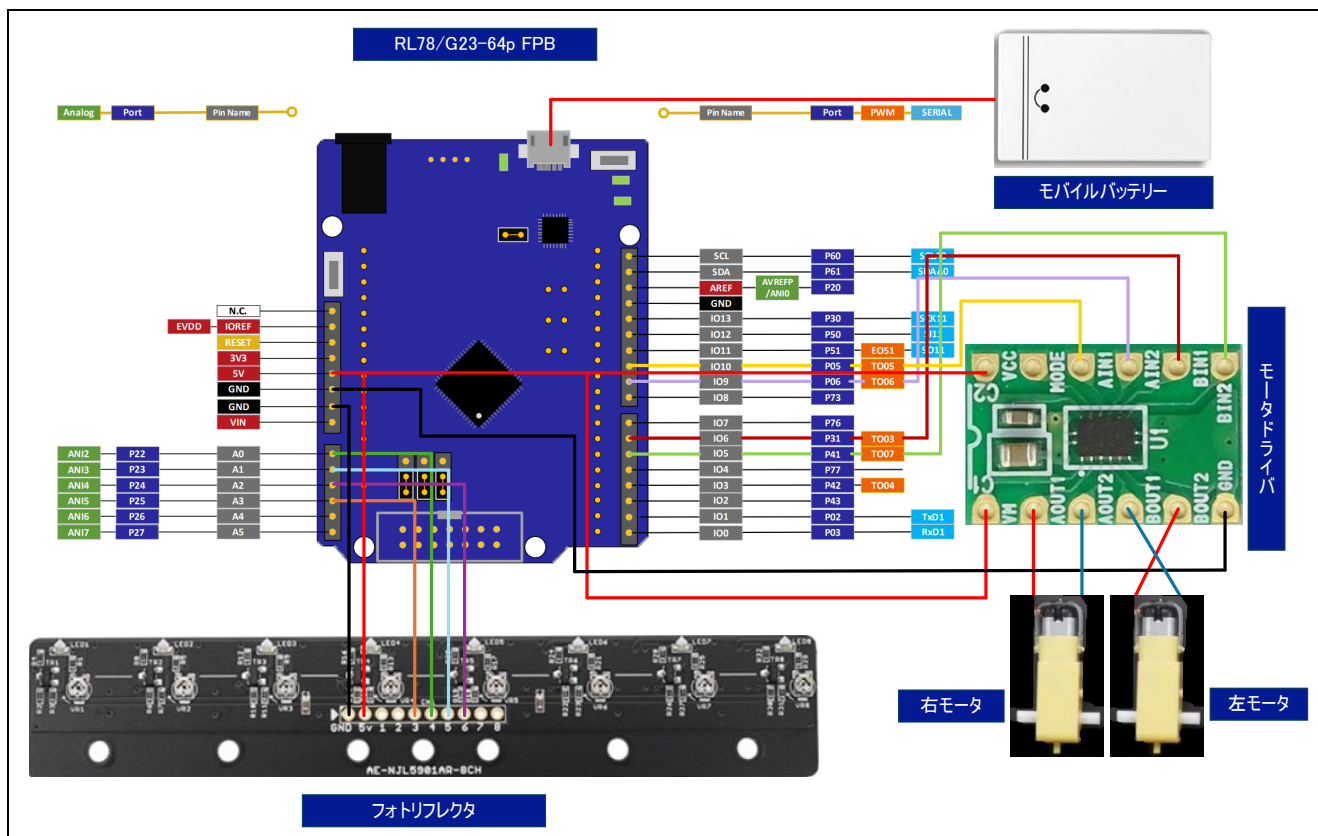
本システムでは、評価ボードへの電源供給は USB を使用します。評価ボードの回路を確認し、必要に応じてジャンパを設定してください。

本システムでは、評価ボードのジャンパを以下のように設定します。

表 3-1 評価ボードのジャンパ設定

ジャンパ	設定	機能
J8	1-2 ショート	COM port デバッグ使用
J9		
J11		
J13	オープン	
J17	1-2 ショート	マイコンへの 5V 電源供給

図 3-1 評価ボード、モータドライバ搭載モジュール、フォトフレクタ搭載モジュールの接続





### 3.2 使用端子一覧

本システムの使用端子を以下に示します。

表 3-2 本システムの使用端子一覧

Arduino™ 信号名	FPB の 端子番号	スケッチの 端子番号	端子の機能	接続先
IO~6	21	6	前進制御；PWM 出力(0%) 後退処理：PWM 出力で左モータ の速度制御	前進制御：モータドライ バの BIN1 後退制御：モータドライ バの BIN2
IO~5	4	5	前進制御；PWM 出力で左モータ の速度制御 後退処理：PWM 出力(0%)	前進制御：モータドライ バの BIN2 後退制御：モータドライ バの BIN1
IO~10	31	10	前進制御；PWM 出力(0%) 後退処理：PWM 出力で右モータ の速度制御	前進制御：モータドライ バの AIN1 後退制御：モータドライ バの AIN2
IO~9	30	9	前進制御；PWM 出力で右モータ の速度制御 後退処理：PWM 出力(0%)	前進制御：モータドライ バの AIN2 後退制御：モータドライ バの AIN1
A0	54	A0	アナログ入力：フォトリフレクタ から値を取得	フォトリフレクタ LED4
A1	53	A1	アナログ入力：フォトリフレクタ から値を取得	フォトリフレクタ LED5
A2	52	A2	アナログ入力：フォトリフレクタ から値を取得	フォトリフレクタ LED6
A3	51	A3	アナログ入力：フォトリフレクタ から値を取得	フォトリフレクタ LED3
5V	5V	-	5V 電力供給	・フォトリフレクタの 5V ・モータドライバの VSS ・モータドライバの VM
GND	GND	-	GND	・モータドライバの GND
GND	GND	-	GND	・フォトリフレクタ GND

各ボードの詳細な端子の説明は、以下のマニュアルを参照してください。

- RL78/G23-64p Fast Prototyping Board ユーザーズマニュアル (R20UT4814)
- [RL78G23 64pin Fast Prototyping Board · renesas/Arduino Wiki · GitHub](#)

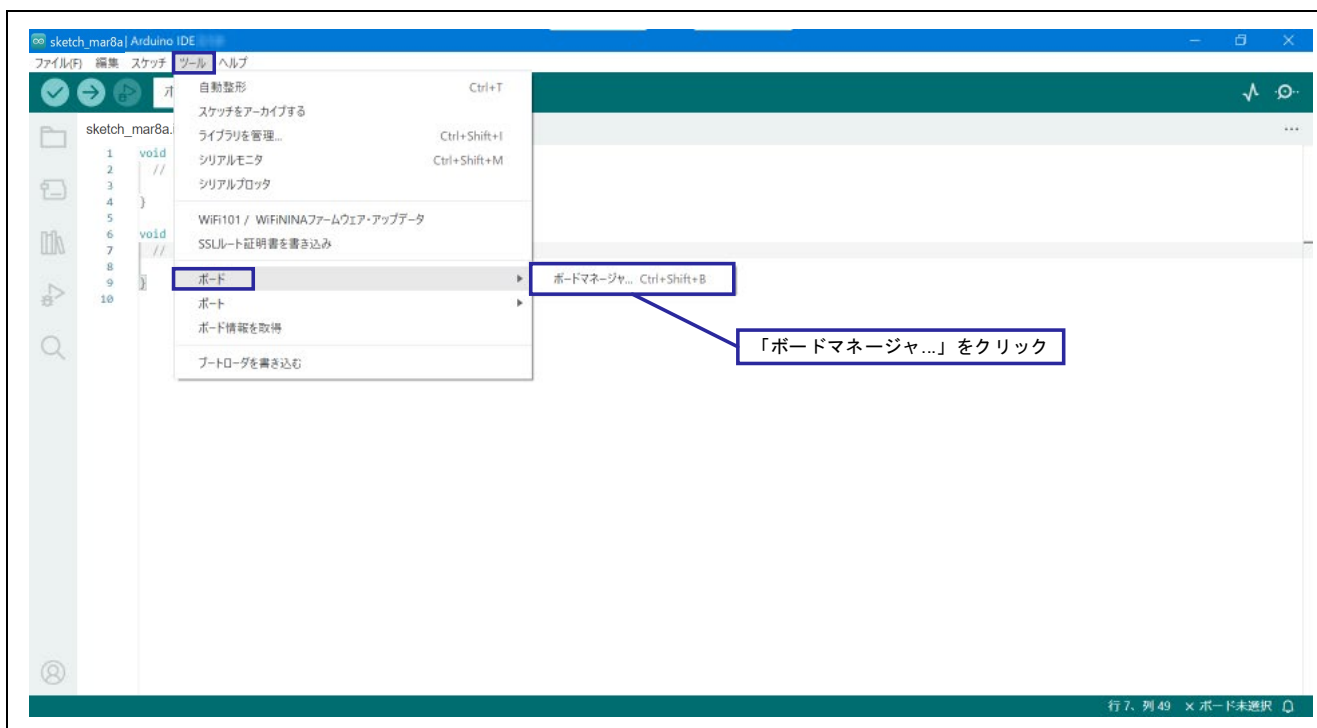
### 3.3 Arduino™ IDE のセットアップ

本章で Arduino™ のセットアップ手順を説明します。

備考. セットアップ手順は、[クイックスタートガイド · renesas/Arduino Wiki · GitHub](#)に記載されている手順と同様です。また上記サイトでは、LED を点滅させるサンプルスケッチが記載されています。必要に応じて、参照してください。

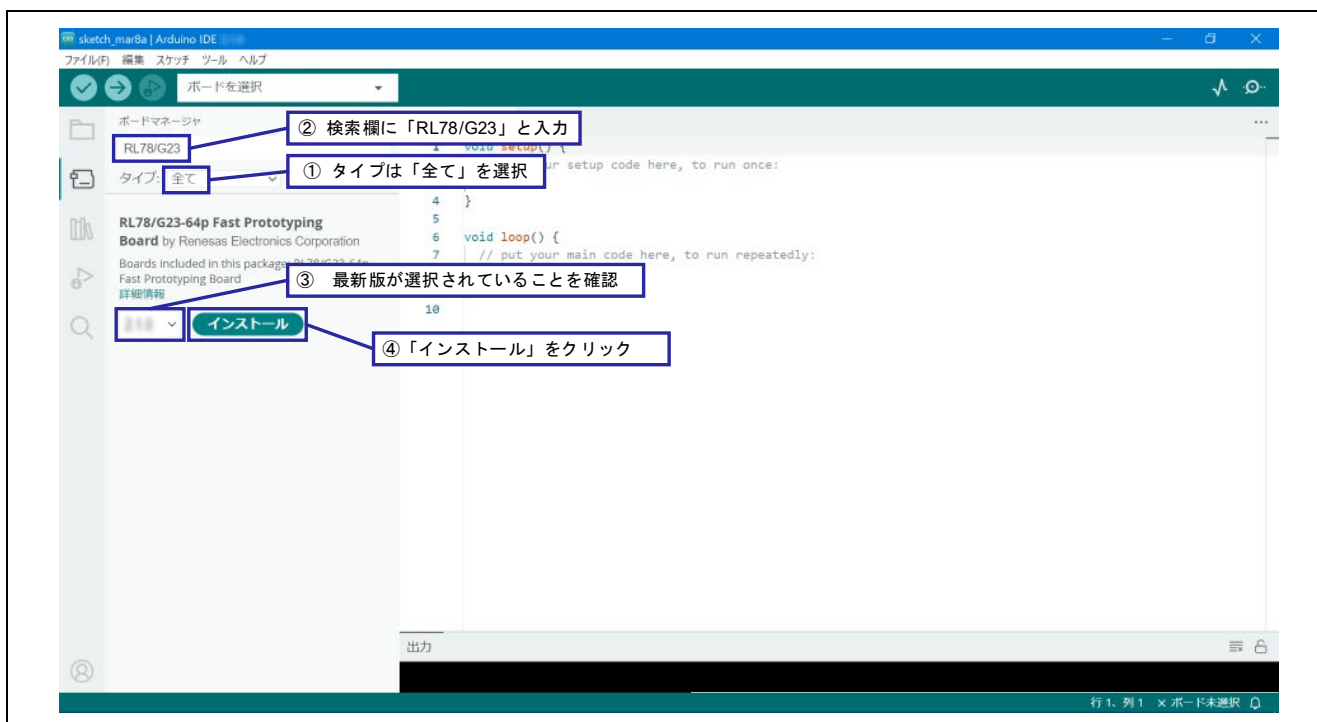
1. Arduino™ IDE を起動します。
2. [ツール] – [ボード:] – [ボードマネージャ] を選択します。

図 3-2 ボードマネージャの選択



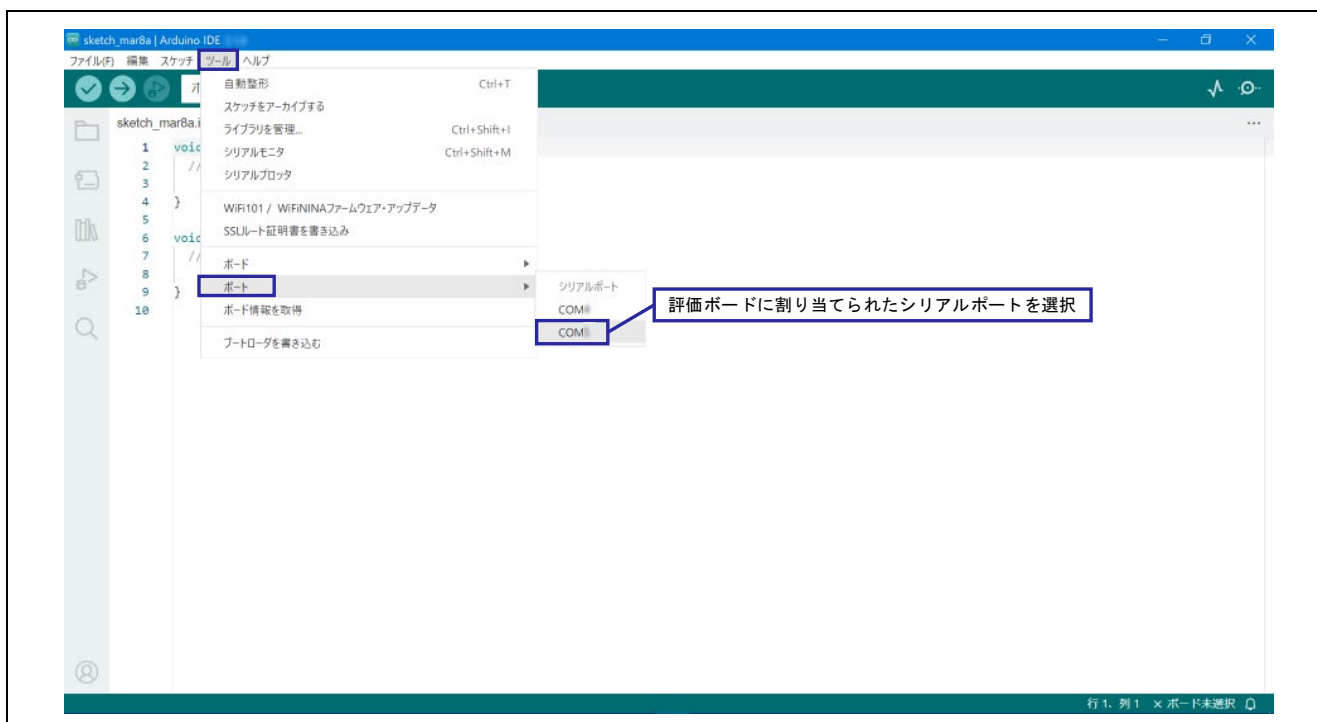
- [タイプ] は “全て” を選択し、検索欄に “RL78/G23” と入力し、表示された [RL78/G23-64p Fast Prototyping Board] の [インストール] をクリックします。

図 3-3 ボードマネージャのインストール



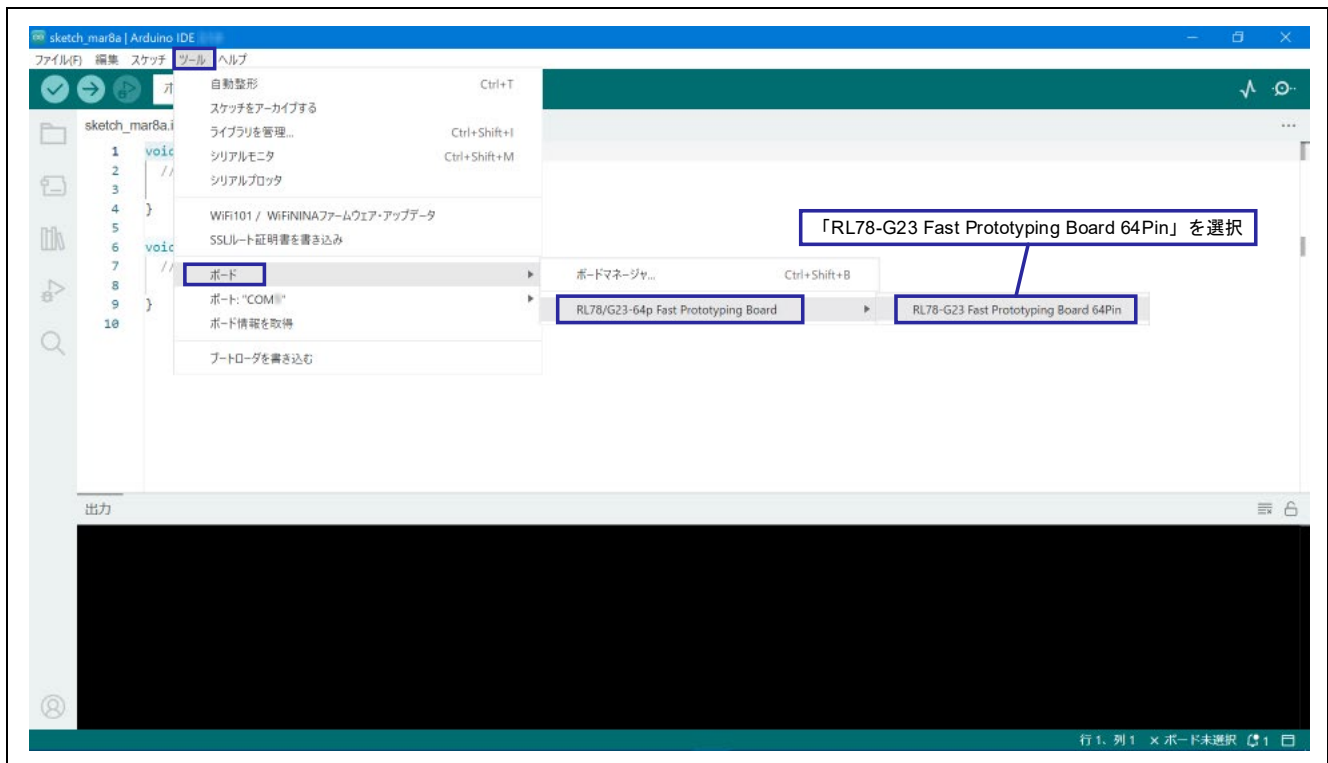
- [ツール] - [ポート] から評価ボードに割り当てられたシリアルポートを選択します。COM ポート番号は、Windows のデバイス マネージャーから確認できます。

図 3-4 シリアルポートの選択



5. [ツール] – [ボード] – [RL78/G23-64p Fast Prototyping Board] – [RL78-G23 Fast Prototyping Board 64Pin] を選択します。

図 3-5 ボードの選択



## 4. ソフトウェア説明

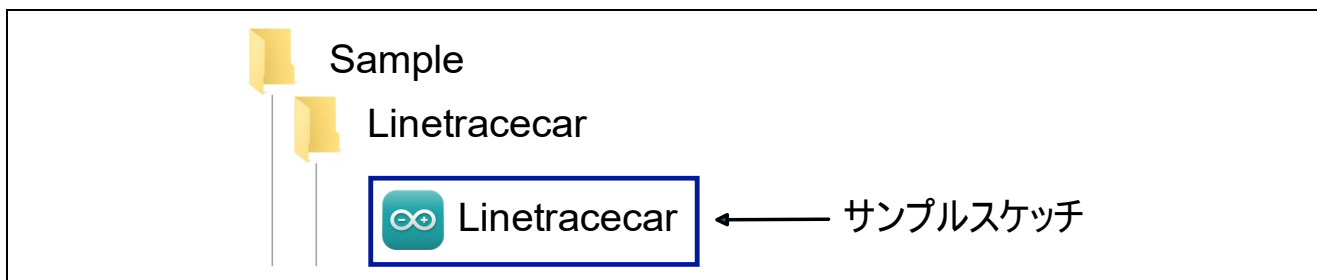
### 4.1 サンプルコードの概要

本サンプルコードは、Arduino™ IDE 上で実行するサンプルスケッチです。

ファイル構成を以下に示します。

使用する API 関数は「4.2 API 関数」を、サンプルスケッチの詳細は「4.3 サンプルスケッチの動作確認手順」を参照してください。

図 4-1 サンプルコードのファイル構成



#### 4.1.1 ライントレースカーのアルゴリズム

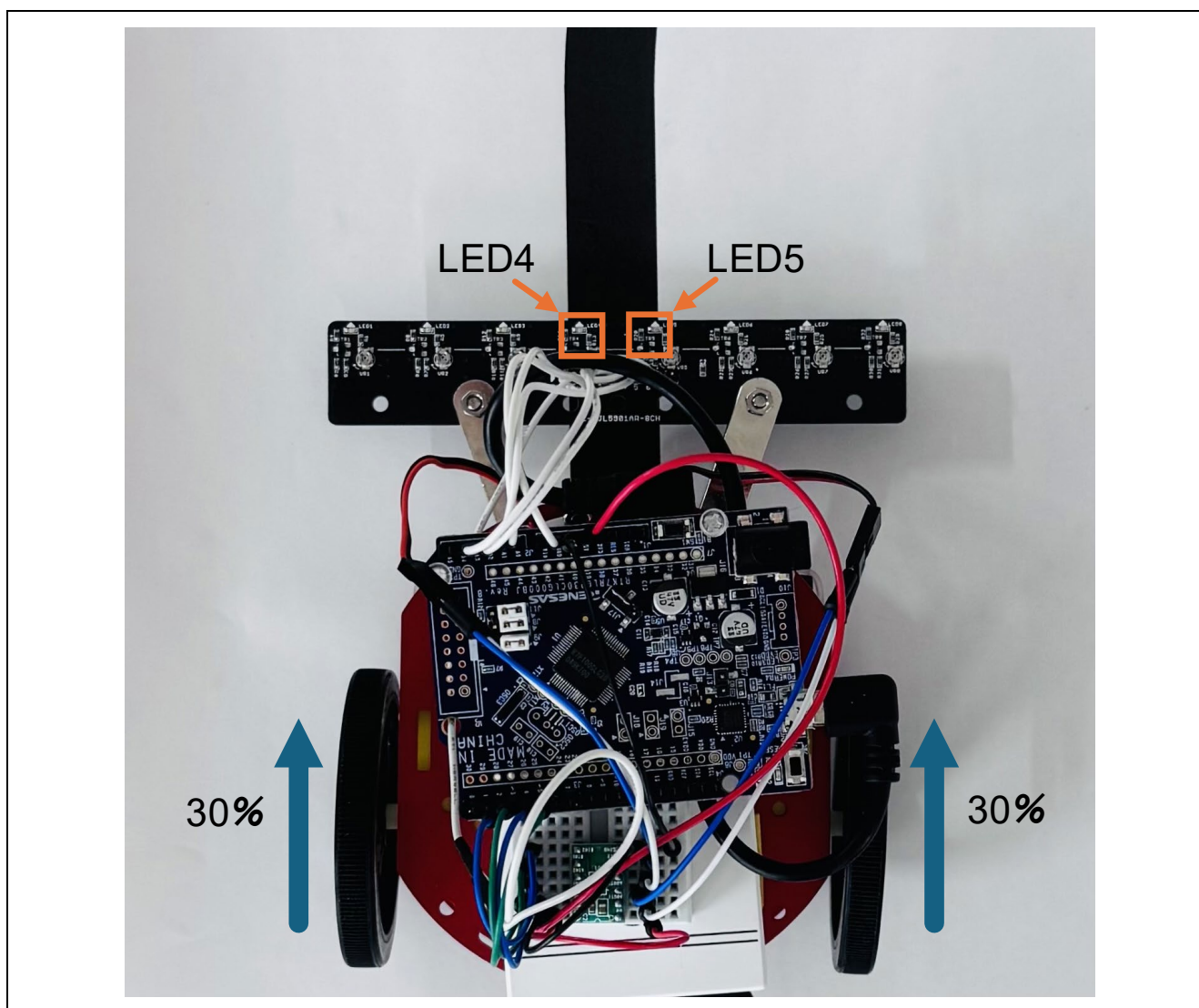
ラインレースカーでは、ラインの位置を把握するためにフォトリフレクタからのセンサデータを取得します。複数配置されたフォトリフレクタがライン上の黒と白を検出し、検出したデータを統合することで車両の位置を認識します。

フォトリフレクタから取得したデータを基に、**直線動作**、**直線補助動作**、**カーブ動作**の3つに分類され、各動作はセンサの判定結果に基づいてモータの速度と方向を調整することで制御します。

まず、**直線動作**は、フォトリフレクタの左右センサー（LED4、LED5）がライン上の黒を検出している場合に発生します。サンプルスケッチでは、両方のモータを**基本速度**（デューティ比 30%に設定）で駆動することで、車両を前方に進める制御を行います。ラインの中央を認識している状態で最も基本的な動作です。

図 4-2 に直線動作の関係を視覚的に示します。

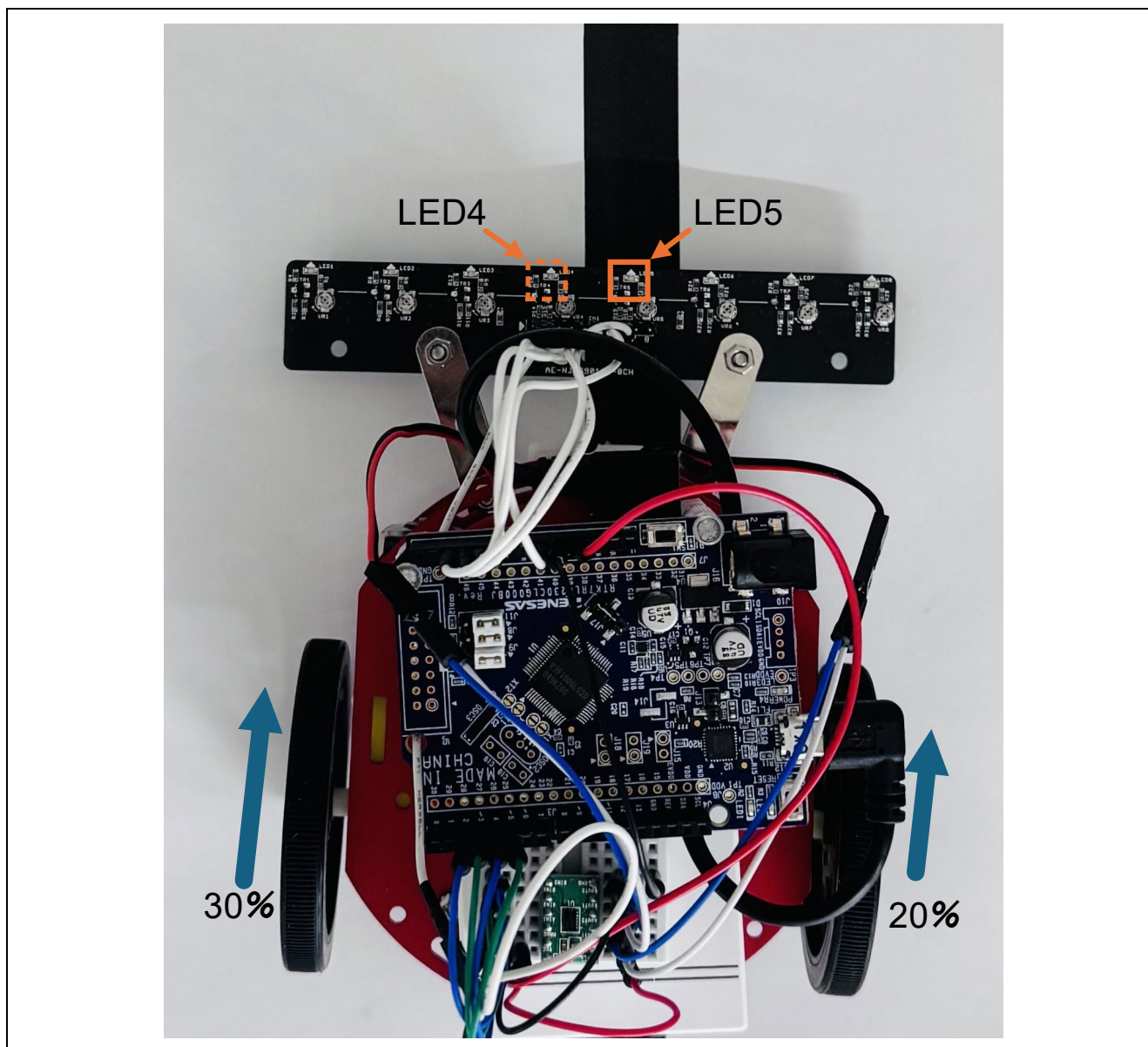
図 4-2 直線動作のイメージ図



次に、**直線補助動作**は、フォトリフレクタの片方のセンサが黒を、もう片方が白を検出した場合に発生します。この場合、ラインを維持するために、サンプルスケッチでは、ラインを維持するために**基本速度**から片方のモータのデューティ比を10%下げて、進行方向を補正します。この制御により、ラインの検出結果に応じた補助動作をします。

図 4-3 に直線補助動作の関係を視覚的に示します。なお、図では LED4 が白=0、LED5 が黒=1 の場合を例として示しています。

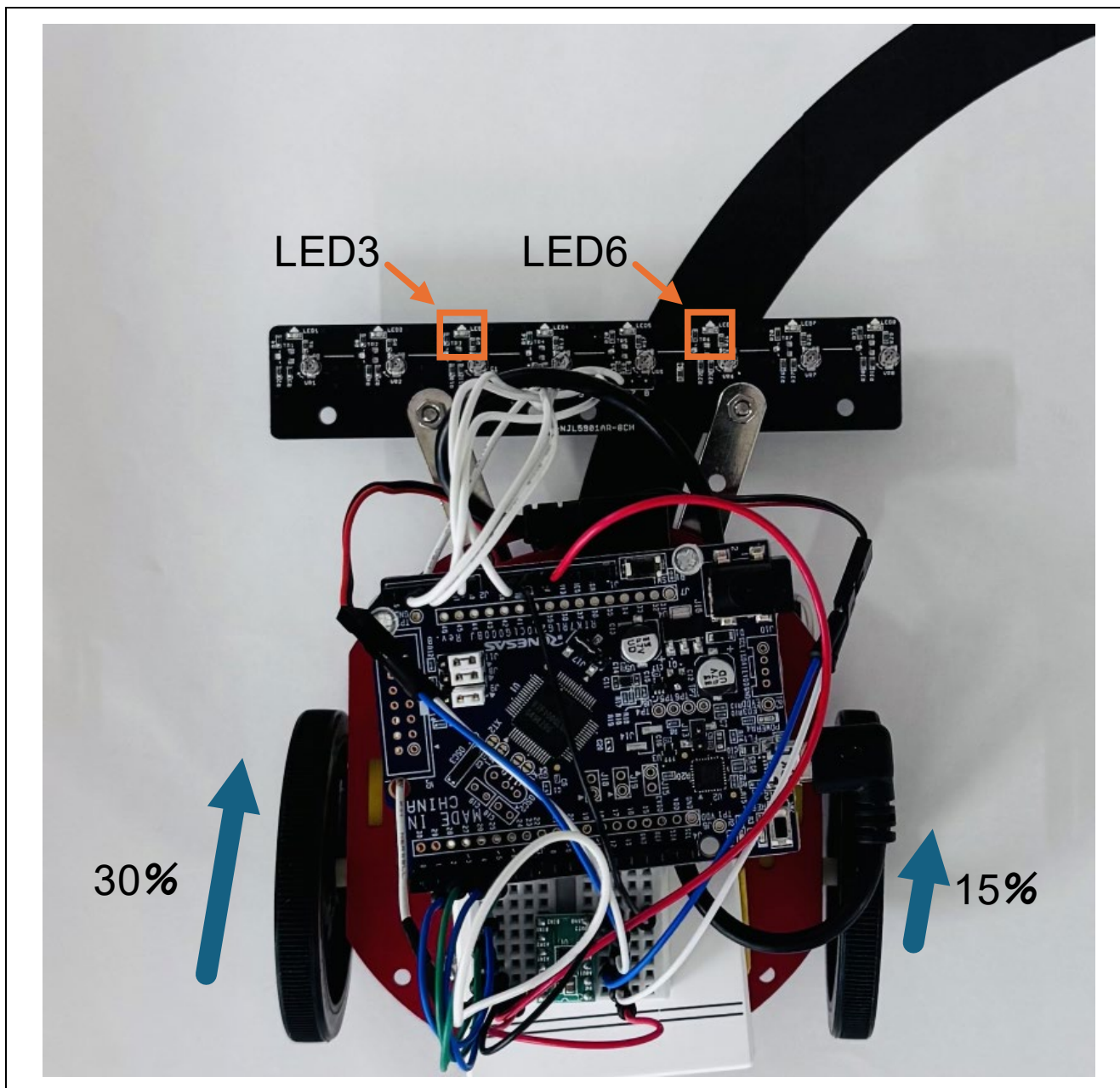
図 4-3 直線補助動作のイメージ図



カーブ動作は、補助センサー（LED3、LED6）が黒を検出した場合に発生します。サンプルスケッチでは、基本速度から片方のモータのデューティ比を 15% 下げることで、ラインの曲線に対応する動作を実現します。LED3 が黒を検出した場合は左カーブ、LED6 が黒を検出した場合は右カーブの動作を実行します。

図 4-4 では、LED6 が黒を検出した場合の右カーブ動作のイメージを示しています。

図 4-4 カーブ動作のイメージ図





### 4.1.2 フォトリフレクタの閾値設定方法

ここでは、フォトリフレクタによる黒線検知の閾値設定方法を説明します。閾値設定には Arduino™ IDE のシリアルプロッタ機能を用います。

#### 1. シリアルプロッタの表示

[ツール] – [シリアルプロッタ] を選択します。図 4-6 のようにシリアルプロッタを表示されます。

図 4-5 シリアルプロッタの表示方法

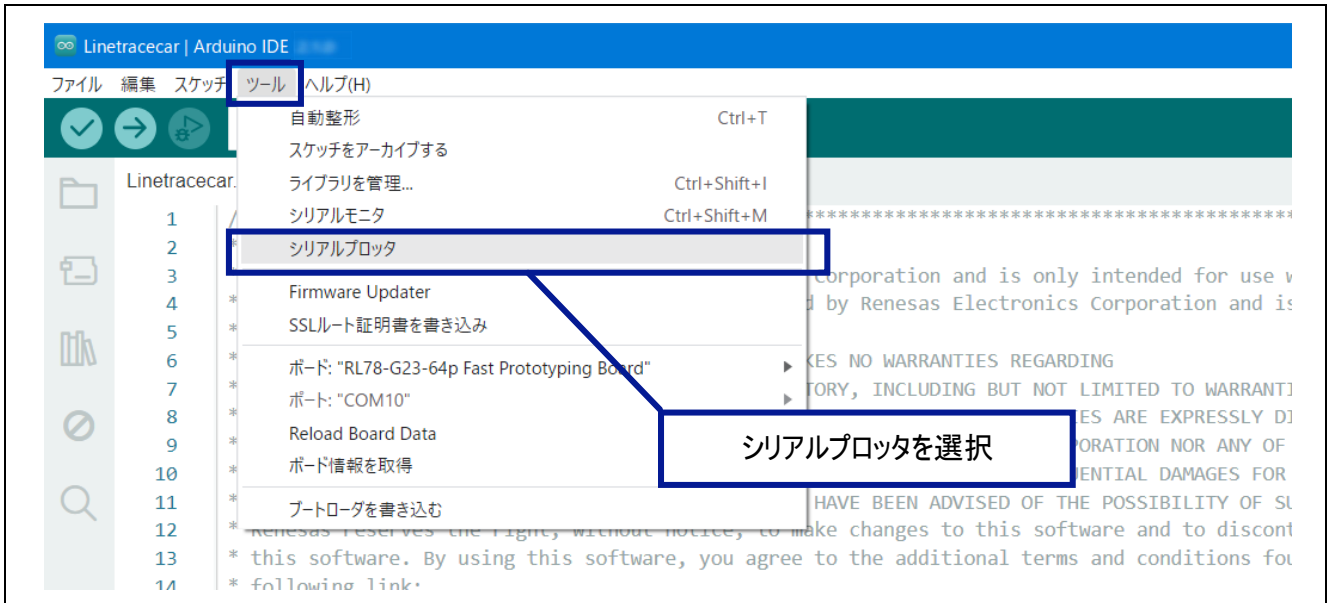
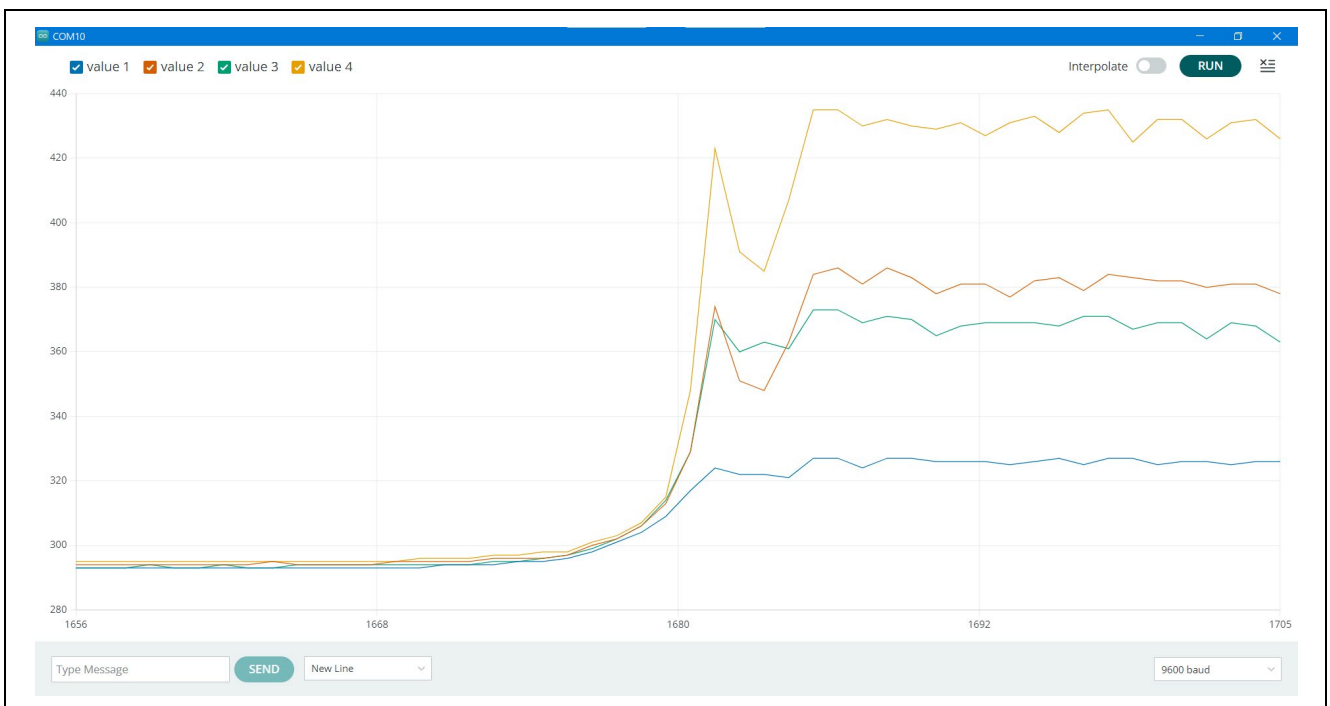


図 4-6 シリアルプロッタの表示例



## 2. フォトリフレクタの閾値設定

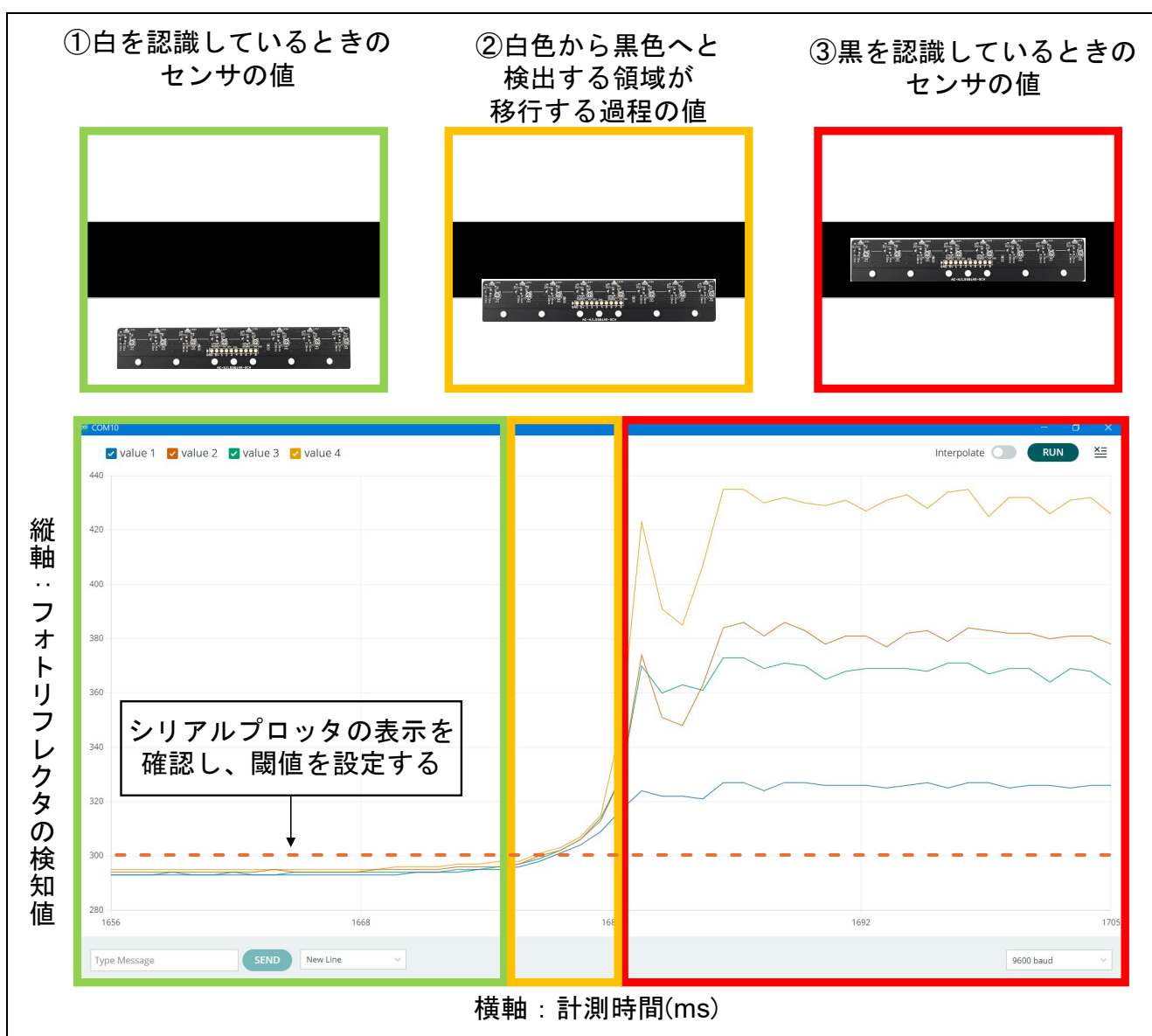
シリアルプロッタには、フォトリフレクタから取得されるデータが表示され、グラフの横軸は時間を示し、100ms ごとにセンサの検知値をリアルタイムに更新します。

シリアルプロッタの縦軸は、各センサの検知値を表しています。この値は、センサが検知した反射光の強さを基に、Arduino™ の analogRead() で取得されたものです。

analogRead() は、センサからのアナログ信号を 10 ビットのデジタル値 (0~1023) に変換します。この変換はリファレンス電圧 (通常 5V または 3.3V) を基準として行われます。黒線を検知した場合は値が大きくなり、白では値が小さくなります。シリアルプロッタ上では、センサが黒や白をどのように認識しているかが分かります。

サンプルスケッチでは、フォトリフレクタの検知値を基にしきい値を 300 に設定しています。この設定により、フォトリフレクタの検知値が 300 を超えた場合は「黒」、300 以下の場合には「白」と判定します。各動作は、この判定結果に基づいてモータの駆動を制御することで実現されています。図 4-7 に開発時、シリアルプロッタに表示される内容を示します。

図 4-7 シリアルプロッタに表示される内容



備考: 今回使用するフォトリフレクタの個体差を測定し、全てのセンサで黒線を判別できる値をしきい値としています。

## 4.2 API 関数

以下に本サンプルコードで使用する各ライブラリの API 関数を示します。

表 4-1 使用関数一覧

API 関数名	機能
<code>digitalWrite(pin,value)</code>	デジタル端子に HIGH/LOW を出力
<code>analogWrite(pin, value)</code>	デジタルピンに PWM 信号を出力
<code>analogRead(pin)</code>	アナログ端子から値を取得
<code>pinMode(pin, mode)</code>	ピンのモード(入力または出力)を設定
<code>Serial.Begin(baudRate)</code>	シリアル通信の初期化
<code>Serial.print(data) / Serial.println(data)</code>	フォトリフレクタのデータの取得速度をシリアルモニタに表示
<code>delay(ms)</code>	プログラムを指定した時間 (ms) 停止
<code>attachInterrupt(interrupt, ISR, mode)</code>	指定したピンに対して割り込みを設定 指定した条件(mode)を満たすと指定した関数(ISR)が呼び出される
<code>digitalPinToInterrupt(pin)</code>	ボードに基づき、指定されたピンに対応する割り込み番号を取得

各ライブラリの関数仕様は以下の Arduino™ ホームページを参照してください。

[API リスト](#) · [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](#)

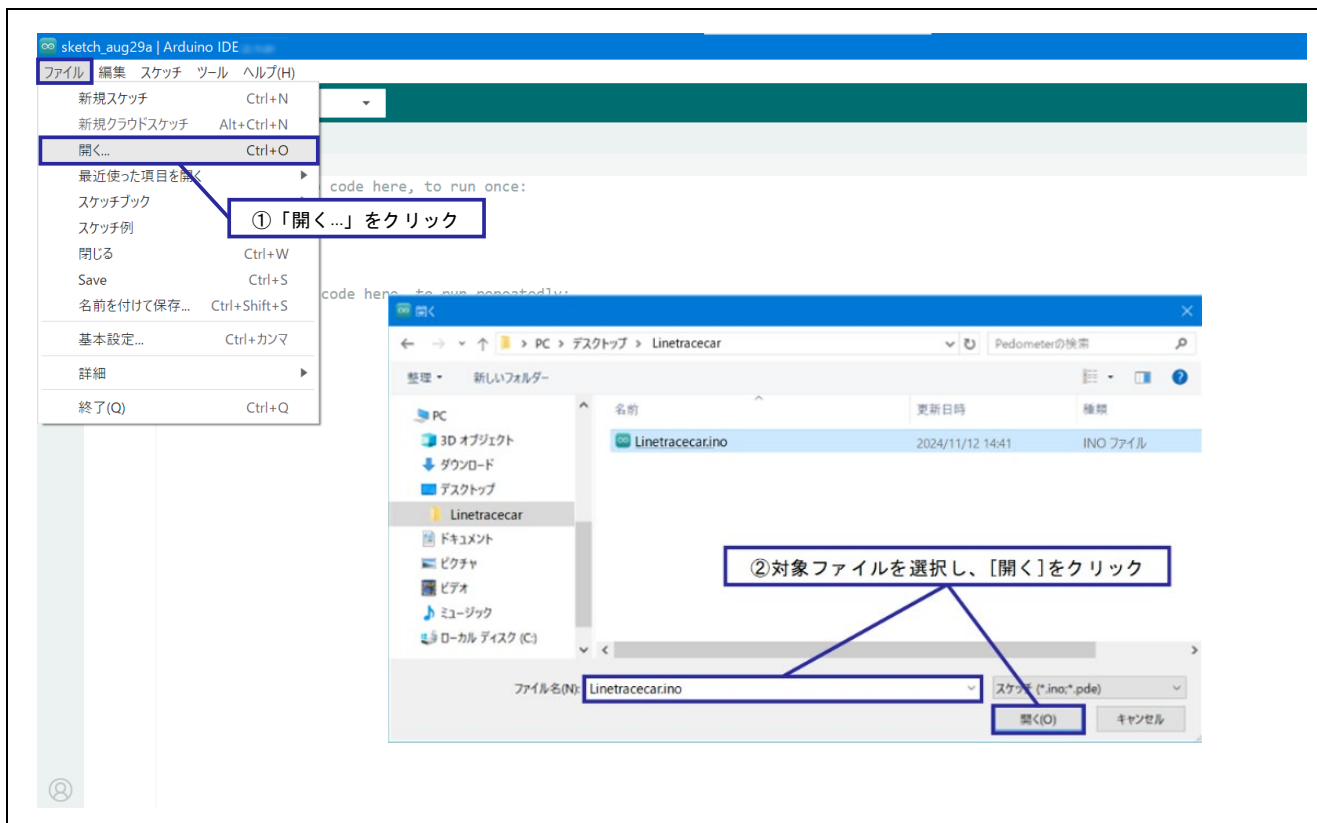
### 4.3 サンプルスケッチの動作確認手順

本サンプルスケッチの動作確認手順を以下に示します。

事前に「3.3 Arduino™ IDE のセットアップ」を行ってください。

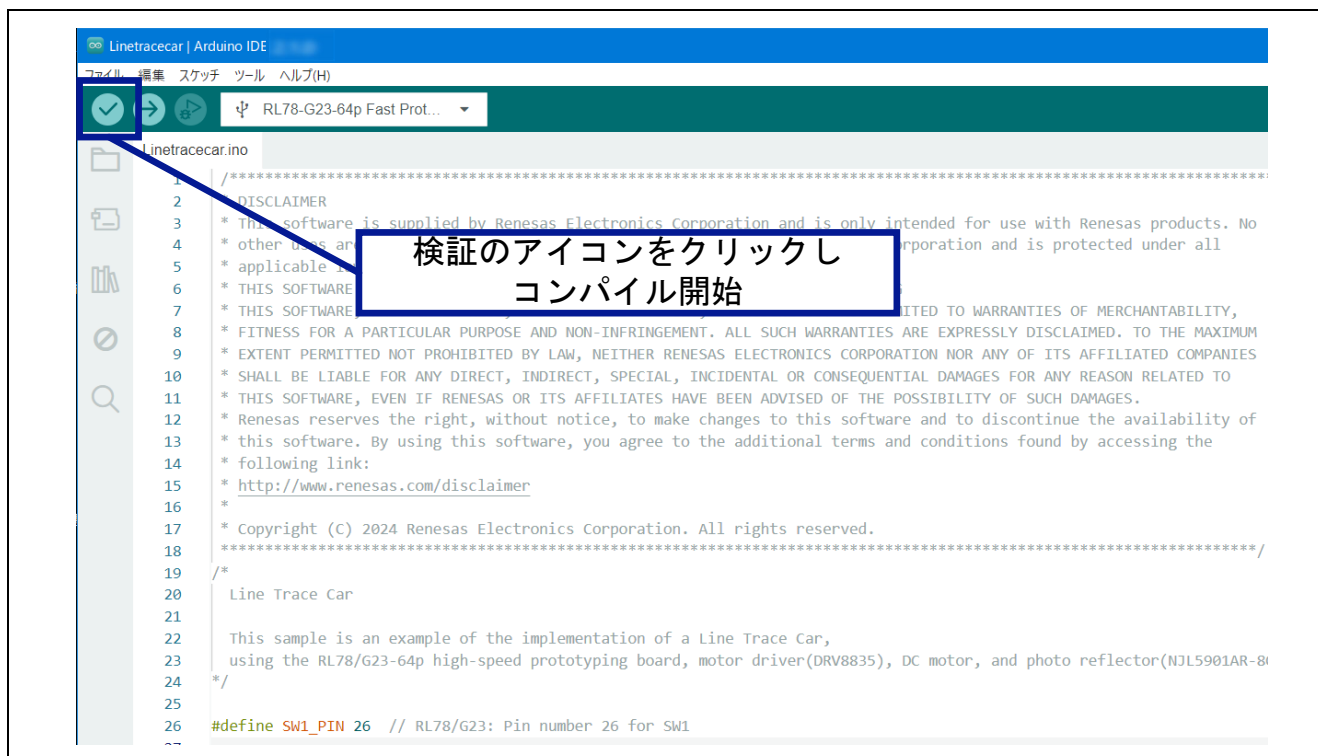
1. [ファイル]–[開く...]をクリックし、サンプルスケッチ Linetracecar.ino を開きます。

図 4-8 サンプルスケッチを起動



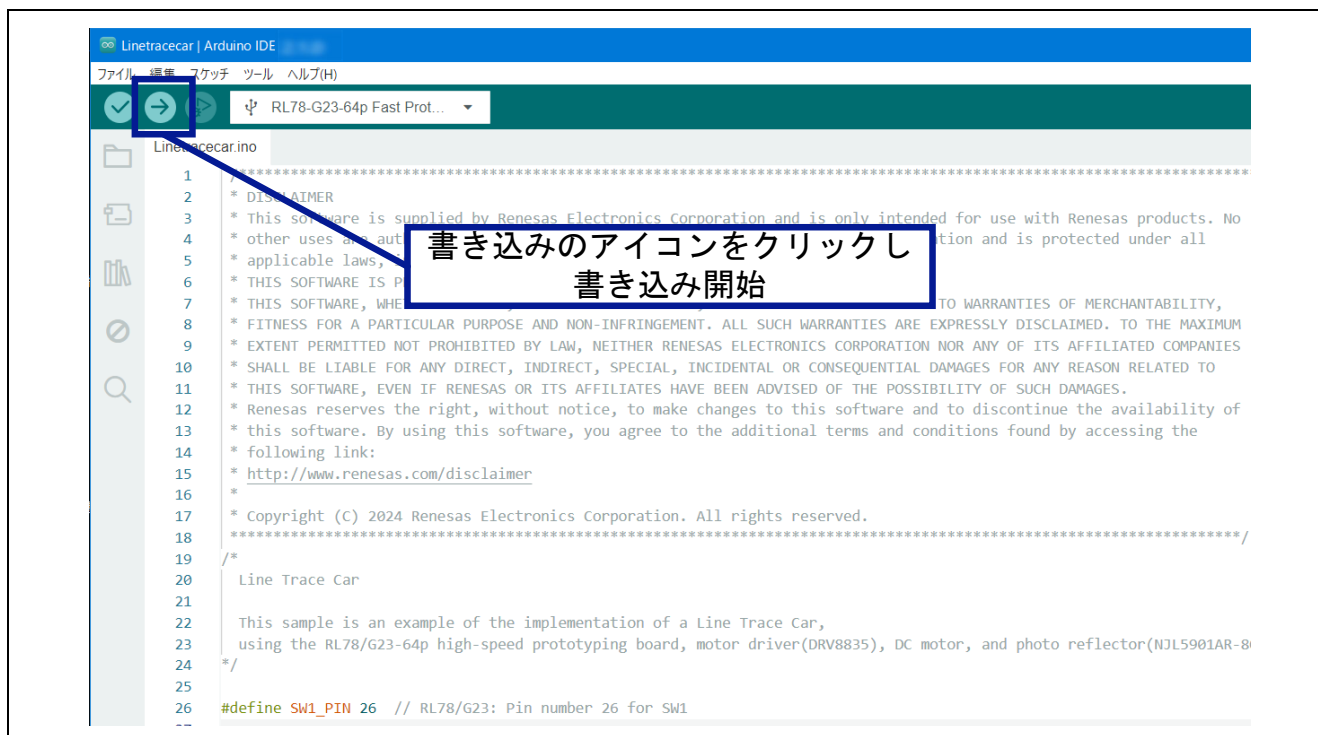
2. 検証のアイコンをクリックし、スケッチをコンパイルします。

図 4-9 スケッチのコンパイル



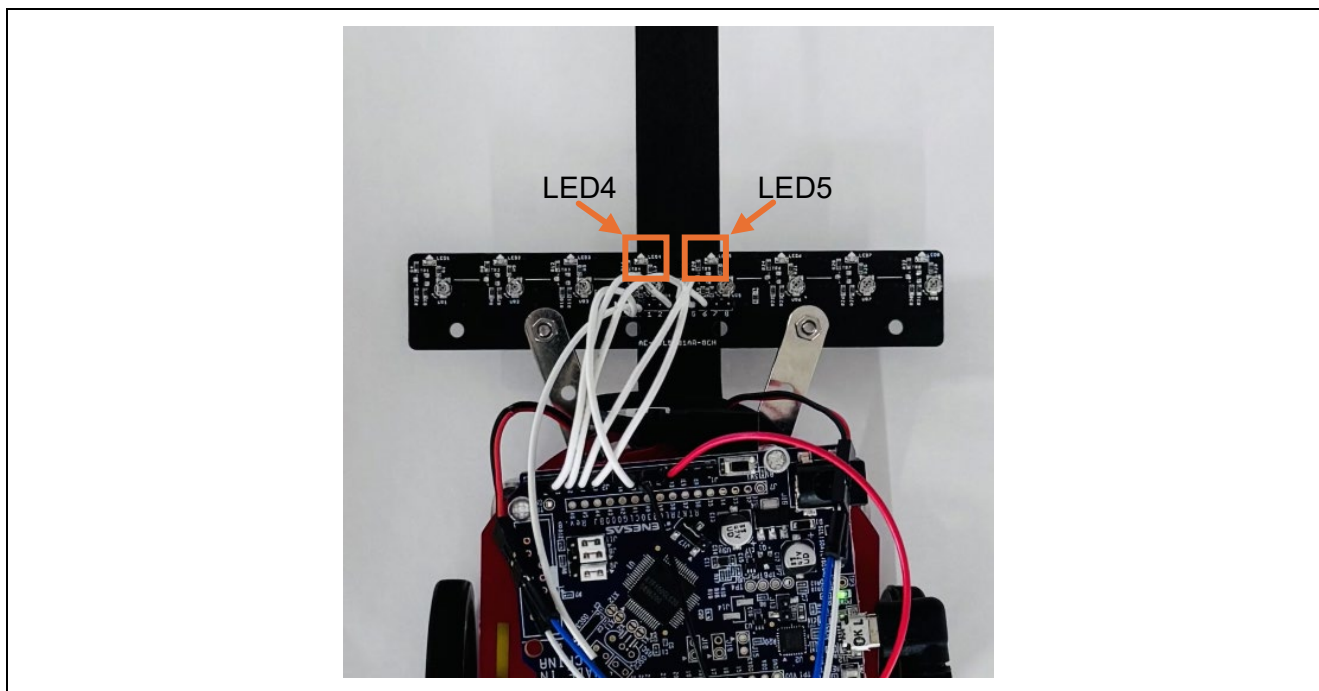
3. コンパイル完了後、書き込みのアイコンをクリックし、デバイスに書き込みます。

図 4-10 スケッチの書き込み



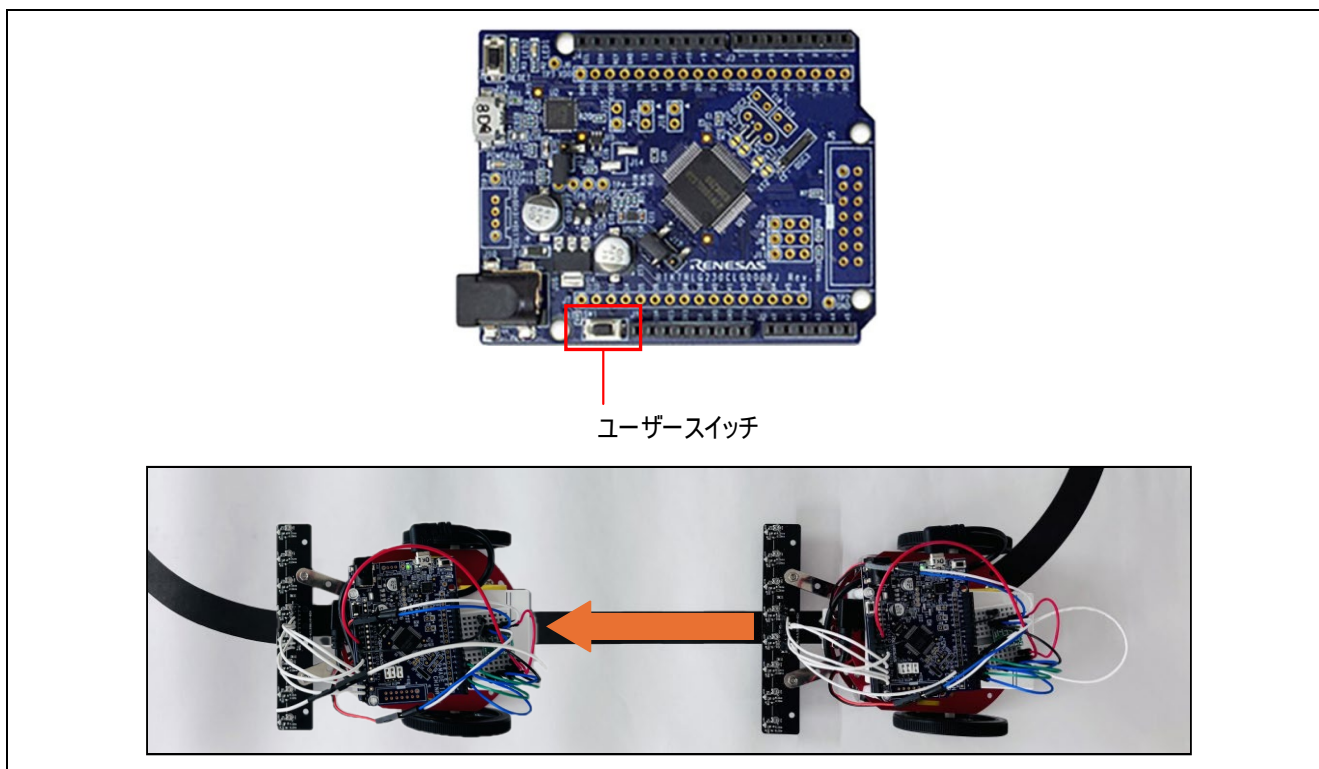
- 書き込み完了後、まず、ライントレースカーをコースの黒線の上に設置します。このとき、センサのLED4 または LED5 が黒線を認識できる位置に調整してください。

図 4-11 設置図の例



- 設置後、モバイルバッテリーや外部電源のスイッチをオンにしてください。この操作でマイコンが起動しますが、ライントレースカーは停止した状態で待機します。その後、ユーザースイッチを押すと、ライントレースカーが動作を開始します。

図 4-12 ライントレースカー動作開始方法と動作イメージ



6. 動作を終了したい場合は、再びユーザースイッチを押してください。この操作によりライントレースカーは停止します。その後、モバイルバッテリーの電源をオフにして、マイコンの動作を停止させます。

### 4.3.1 サンプルスケッチの閾値設定とモータのスピード変更について

このスケッチでは、ライントレースカーの動作を調整するためにいくつかの定数が設定されています。**threshold** はセンサが白と黒を判別する基準値です。この値はシリアルプロッタを使用して、センサが白と黒を検知する値を確認しながら設定してください。

**baseDutyCycle** は車両の基本速度を制御する値で、PWM 信号のデューティ比を決めます。このスケッチでは 0 が 0% (停止)、255 が 100% (最大速度) を意味します。値を上げると速度が速くなりますが、PWM のデューティ比を上げすぎるとモバイルバッテリーの電流不足で動作が停止する可能性があるため注意してください。

**adjustment** は直線補助動作時に片方のモータ速度を制御する値です。片方のセンサがラインを外れた場合、もう片方のモータ速度をこの値に設定して補正します。この値を高くすると素早く方向修正ができますが、直線動作が安定しなくなる場合があります。低くすると動きが穏やかになりますが、修正が遅れコースアウトする可能性があります。

**curveAdjustment** はカーブ検知時に片方のモータを減速させるための速度を設定する値です。カーブでは片方のモータが **baseDutyCycle** の速度で動作し、もう片方はほぼ停止状態で動作します。この「ほぼ停止状態」を実現する速度は **curveAdjustment** で設定しています。値を低くすると停止に近い状態ではなく微妙に動作し、値を高くすると完全に停止に近い動作になります。

よって、**adjustment** と **curveAdjustment** の値は **baseDutyCycle** よりも遅い値に設定する必要があります。これが守られない場合、モータ間の速度差が適切に生じず、黒線に沿って動作しません。これらの値を調整する際は、実際の動作を確認しながら設定するようにしてください。

図 4-13 フォトリフレクタの閾値の設定定数とモータのスピード設定定数

```
// 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 後退の制御

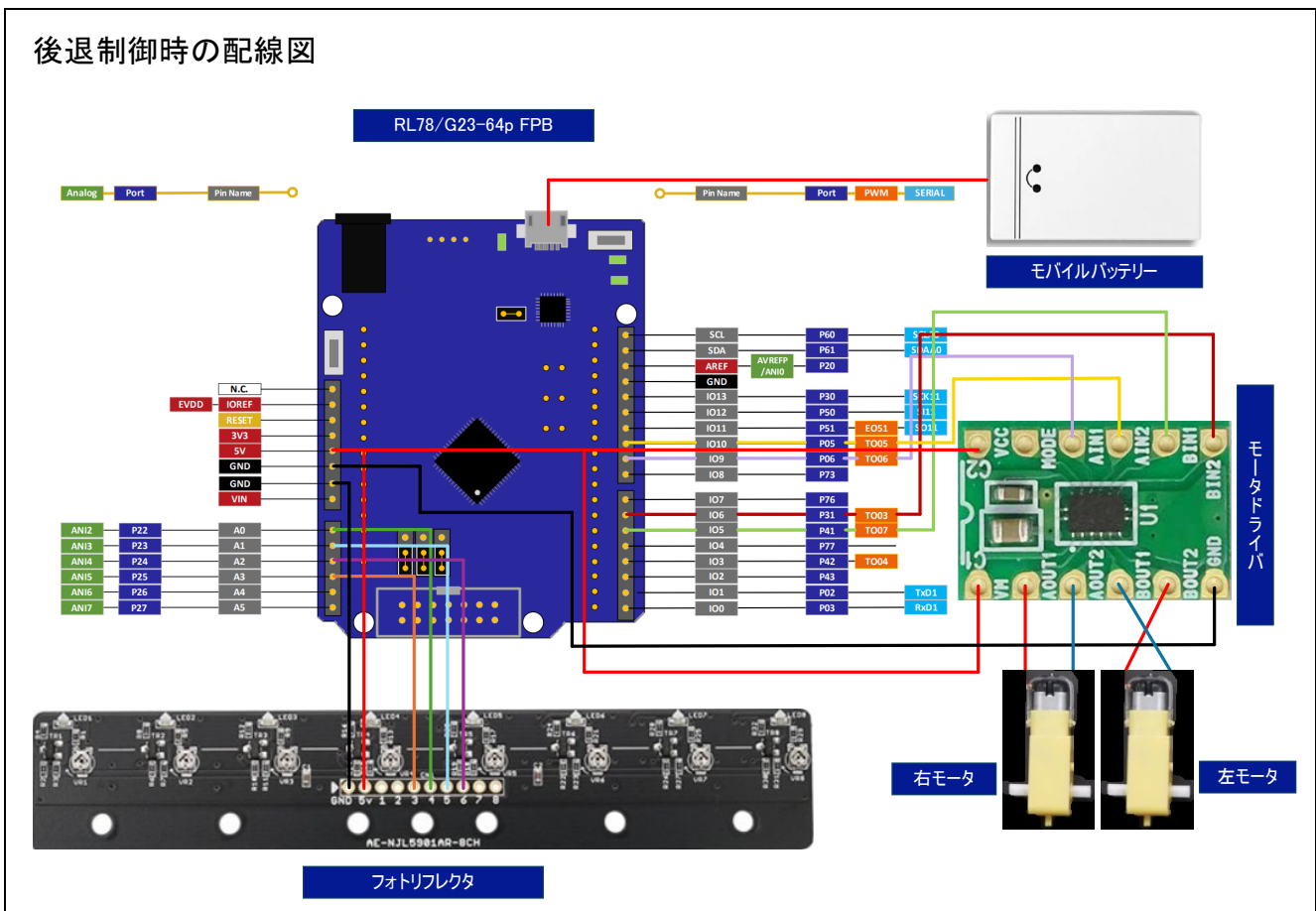
本サンプルスケッチでは、仕様としてライトレースカーの後退制御は記載しておりませんが、配線を修正することで実現できます。

後退制御を行うには、まずモータドライバの AIN1 と AIN2、BIN1 と BIN2 の配線を入れ替えます。この配線変更でモータの回転方向を切り替えることができます。

本サンプルスケッチで使用しているのは DRV8835 モータドライバです。その他のモータドライバでは、AIN2 や BIN2 の制御方法や PWM 信号の解釈が異なる場合があります。そのため、モータドライバの仕様書やデータシートを事前に確認し、挙動と一致しているかを確認してください。仕様が異なる場合は、プログラムを適宜修正する必要があります。

また、図 4-14 では、モータドライバにおける前進制御と後退制御の配線の違いを示しています。

図 4-14 前進制御と後退制御の仕様の違い

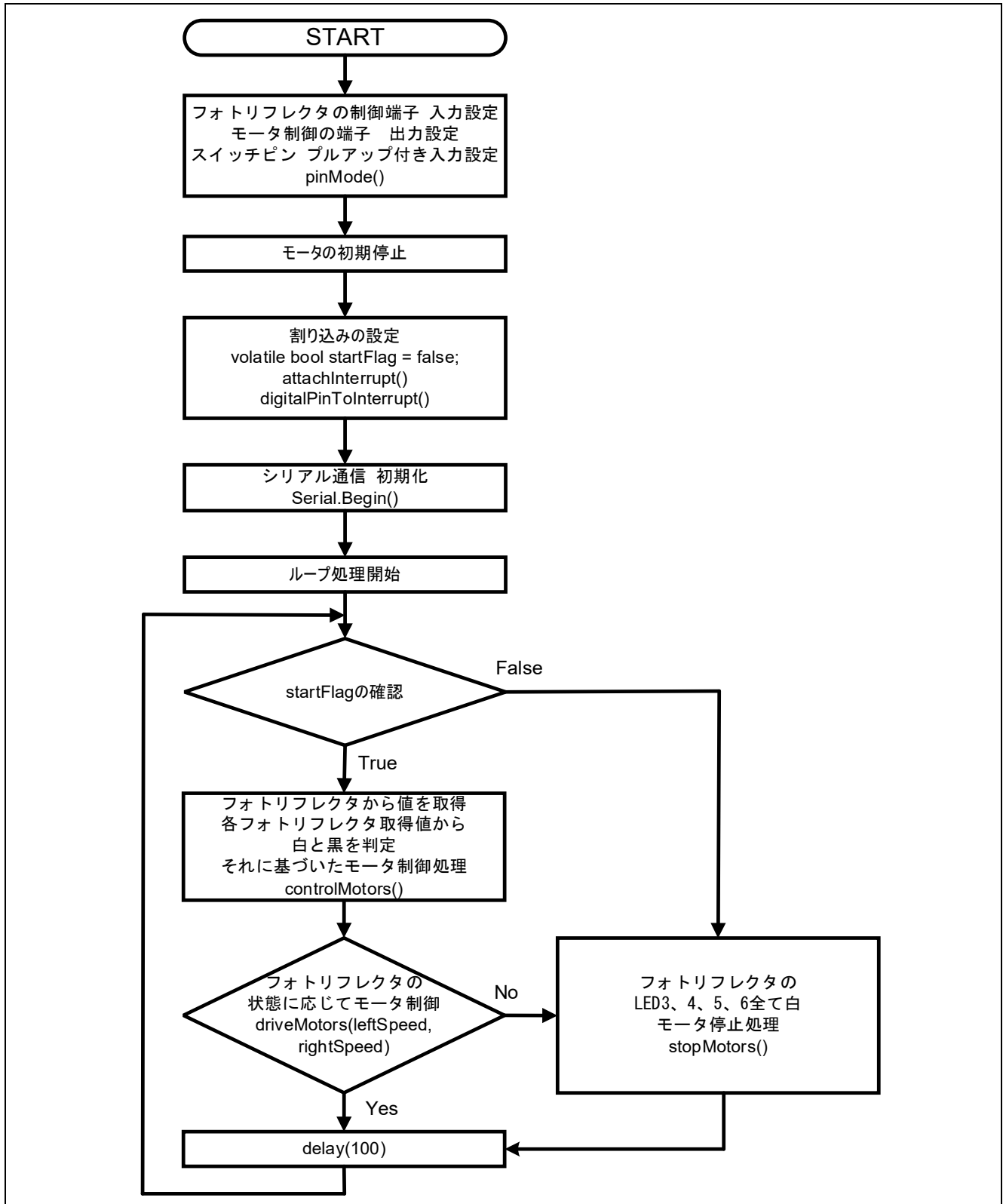


## 4.4 フローチャート

## 4.4.1 メイン処理フローチャート

以下にサンプルスケッチの処理フローを示します。

図 4-15 メイン処理フローチャート

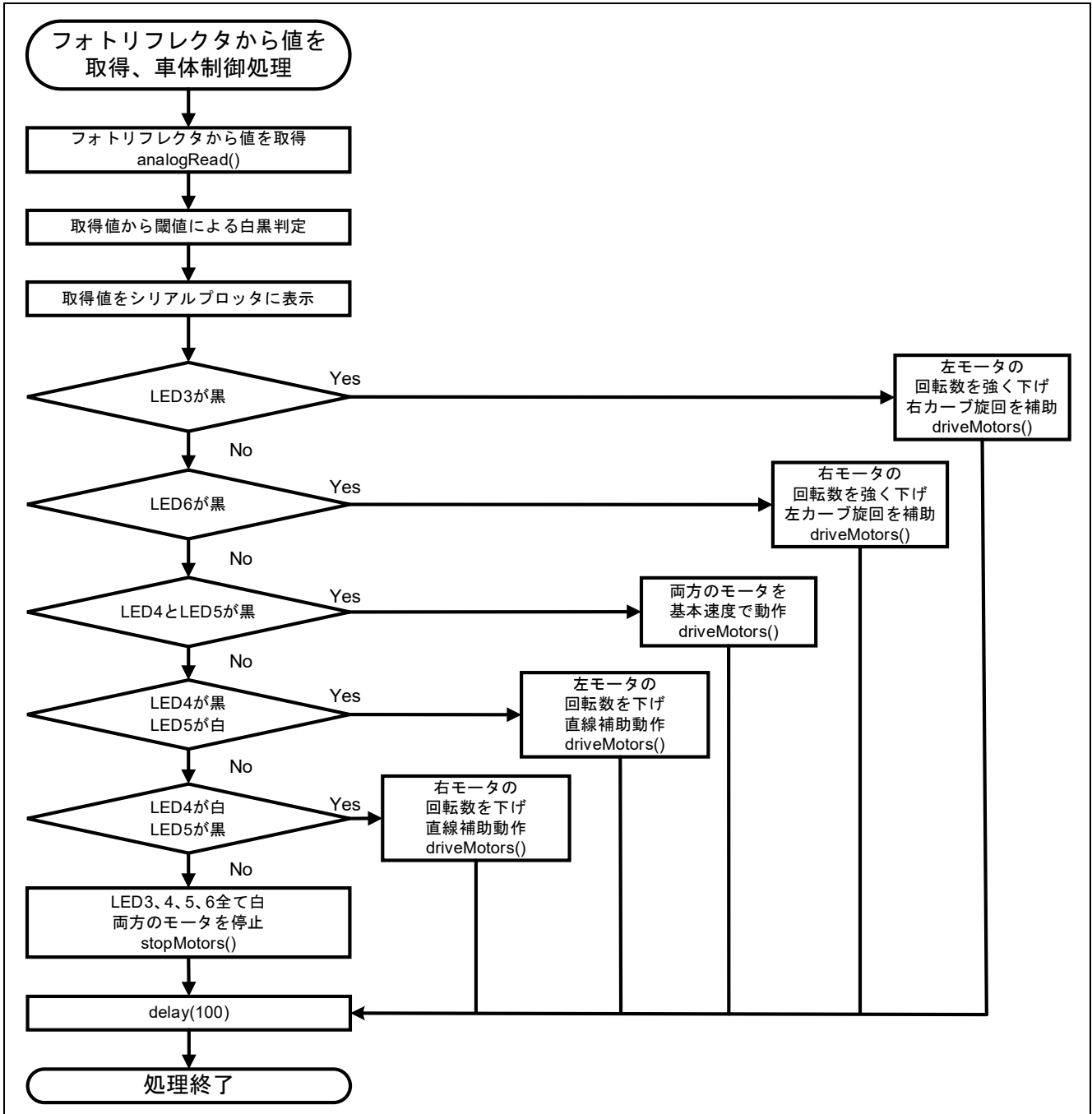


4.4.2 呼び出し関数処理フローチャート

以下に loop 関数から呼び出す関数の処理フローを示します。

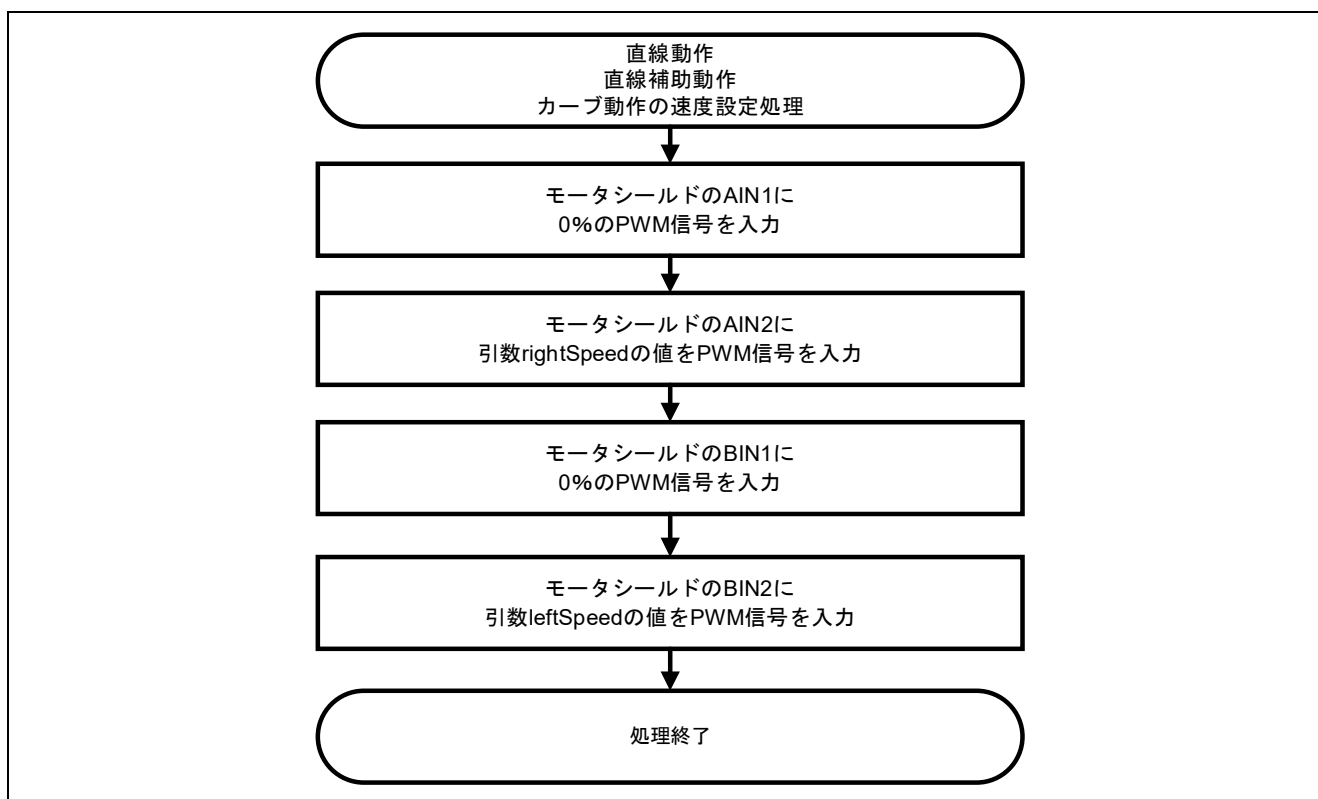
(1) フォトリフレクタから値の取得、車体制御処理 : controlMotors 関数 :

図 4-16 controlMotors 関数のフローチャート



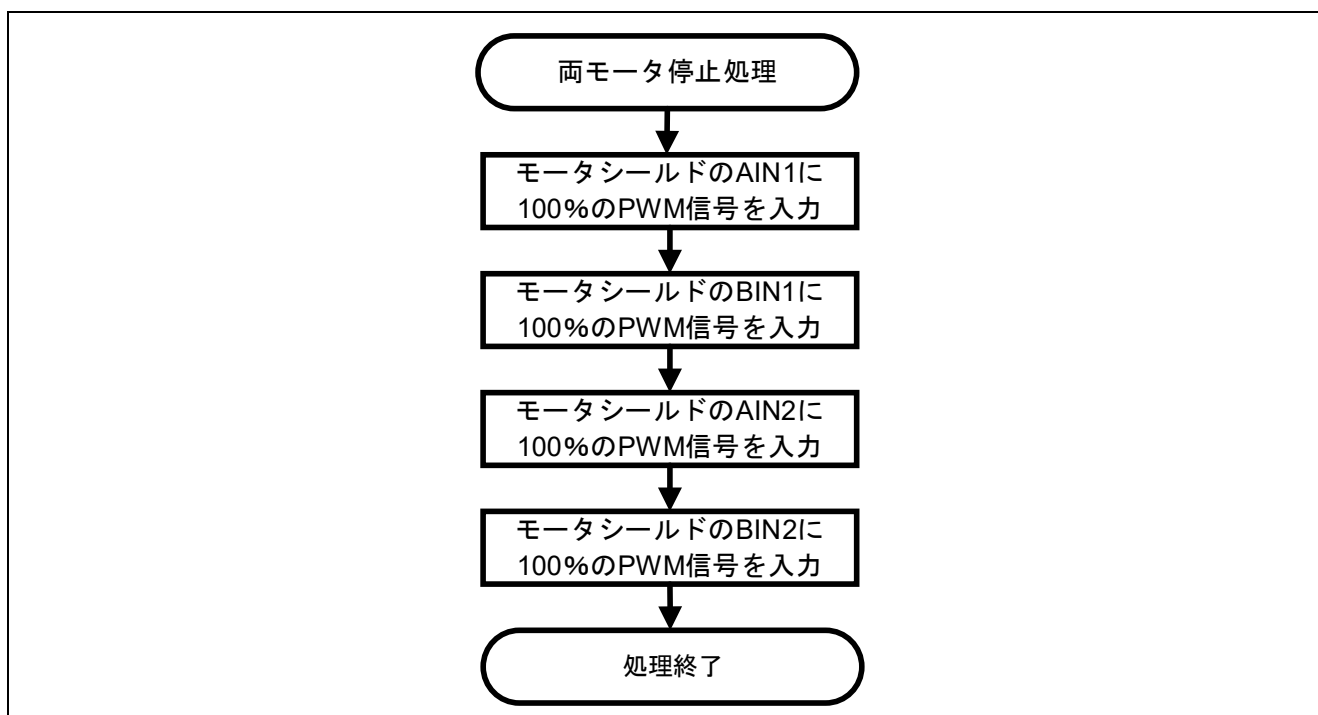
(2) 直線動作、直線補助動作、カーブ動作の速度設定処理 : driveMotors 関数 :

図 4-17 driveMotors 関数のフローチャート



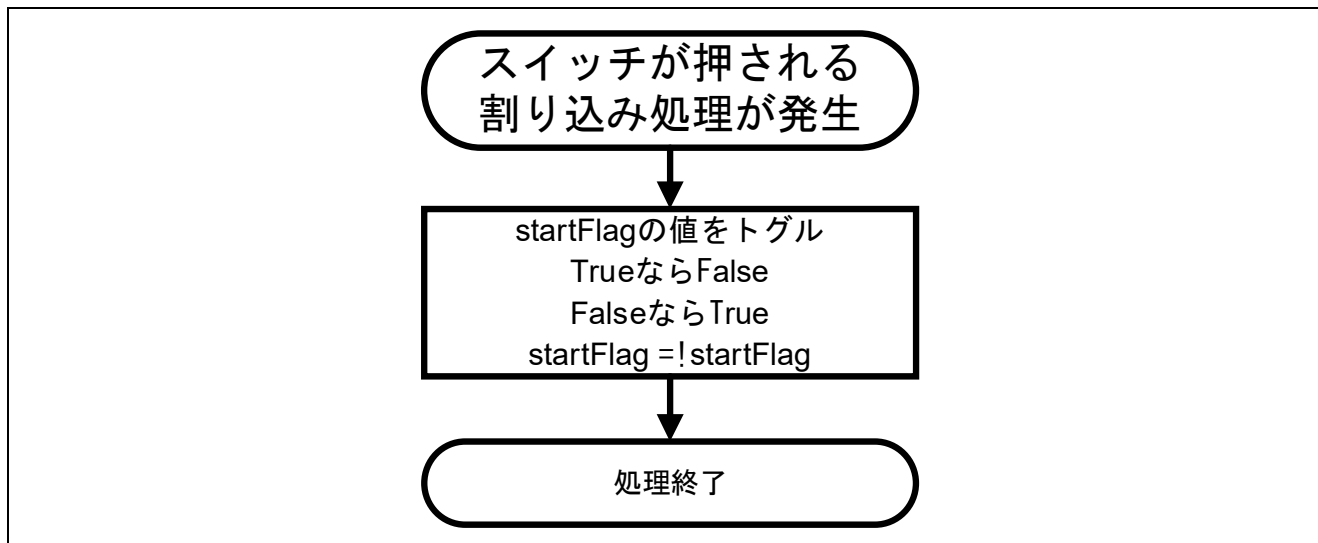
(3) 両モータ処理 : stopMotors 関数 :

図 4-18 stopMotors 関数のフローチャート



(4) 割り込み発生時に startFlag の値トグルする処理 : toggleStartFlag 関数 :

図 4-19 toggleStartFlag 関数のフローチャート



#### 4.4.3 関数詳細

呼び出し関数の詳細を以下に示します。

---

##### void controlMotors()

---

**概要** 各センサの値に基づいてモータの動作を制御する関数。  
センサが検知した状態に応じて、カーブや直進の動作をモータに指示する。

**引数** なし

**戻り値** なし

---



---

##### void driveMotors(int leftSpeed, int rightSpeed)

---

**概要** モータを指定された速度で動作させる関数。  
左モータと右モータの速度を個別に指定可能。

**引数** leftSpeed 左モータの速度 (PWM 値、0~255) 。

**戻り値** rightSpeed 右モータの速度 (PWM 値、0~255) 。

---



---

##### void stopMotors()

---

**概要** 両方のモータを停止させる関数。PWM 信号を停止状態 (255) に設定し、モータの回転を止める。

**引数** なし

**戻り値** なし

---



---

##### void toggleStartFlag()

---

**概要** 割り込み発生時に呼び出され、startFlag をトグル (True/False の切り替え) する関数。  
切り替え後の状態をシリアルモニタに出力する。

**引数** なし

**戻り値** なし

---

## 5. 注意事項

### 5.1 Windows デバイス マネージャーに COM ポートが表示されない

初めて PC と評価ボードを接続した時に、PC がポートを認識せず Windows デバイス マネージャーに COM ポートが表示されない場合があります。

表示されない場合は、評価ボードに実装されている FTDI 社製 USB-シリアル変換器 (FT232RQ) のドライバを以下の手順でインストールしてください。

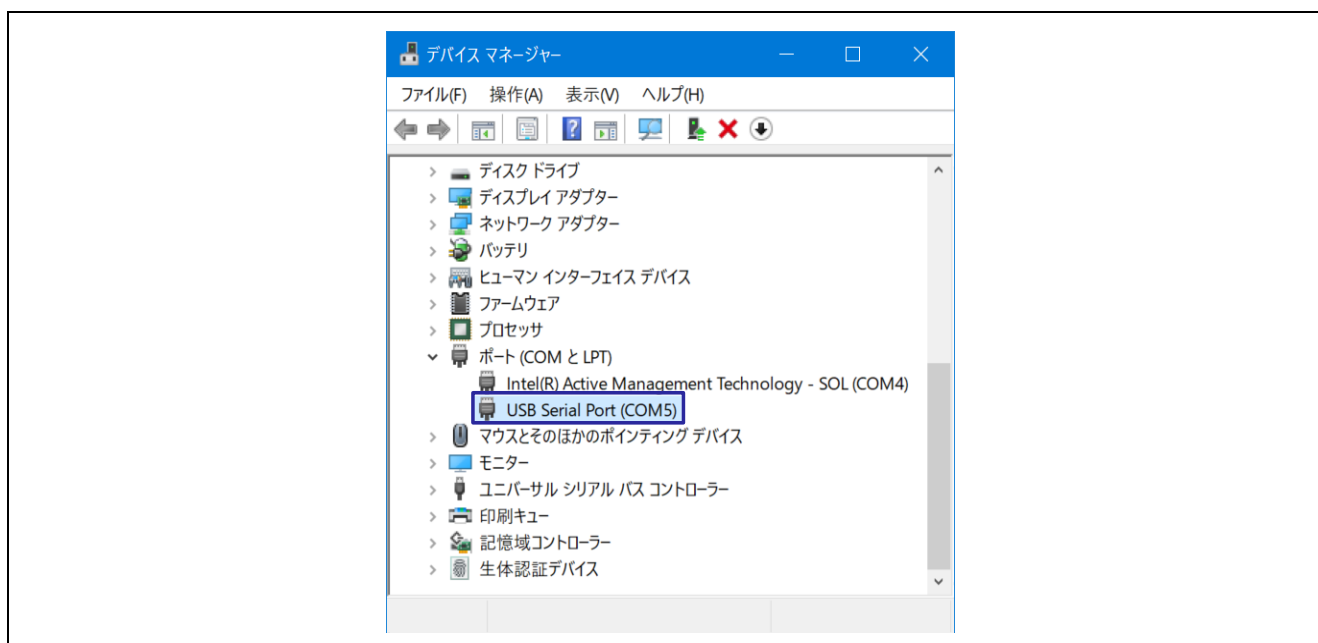
1. FTDI 社のホームページから対象 OS の最新版ドライバのインストーラをダウンロードし、インストールします。

<https://ftdichip.com/drivers/vcp-drivers/>

2. インストール完了後、Windows デバイス マネージャーの「ポート(COM と LPT)」に「USB Serial Port(COMx)」が表示されます。

以下の場合、対象 COM ポートが COM5 であることが確認できます。

図 5-1 ドライバインストール後の Windows デバイス マネージャー



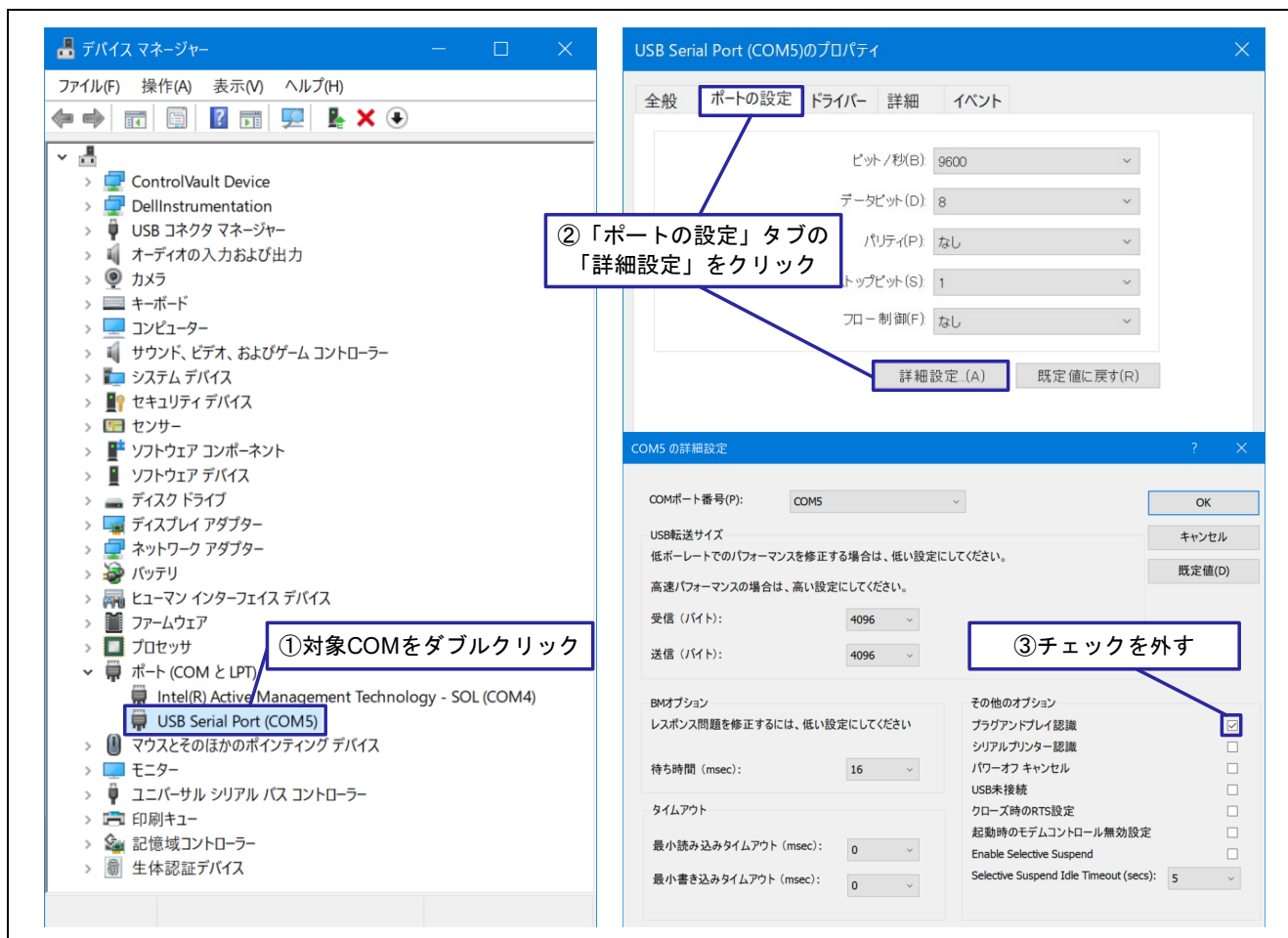
USB-シリアル変換器、COM ポートの詳細は、「RL78/G23-64p Fast Prototyping Board ユーザーズマニュアル」の「5.11 USB-シリアル変換器」、「5.12 USB-シリアル変換器リセットヘッダ」を参照してください。

## 5.2 RL78/G23-64p Fast Prototyping Board に正しく書き込みができない

マイコンボードに書き込む際、正しく接続できない場合があります。

接続できない場合は、Windows デバイス マネージャーから対象 COM のプラグアンドプレイ認識のチェックボックスのチェックを外してください。

図 5-2 対象 COM の設定例



## 5.3 モバイルバッテリーで電源供給開始後、すぐに電源供給を停止してしまう

オートパワーオフ機能がついたモバイルバッテリーを使用している可能性があります。

FPB のような微弱な電流で動作するボードを一般的なモバイルバッテリーで電源供給すると、保護機能 (オートパワーオフ機能) が働き供給を止めてしまうため、電源が自動で切れないモバイルバッテリーをご使用ください。

## 6. サンプルコード

本アプリケーションノートは、サンプルコードを用意しています。

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/G23 ユーザーズマニュアル ハードウェア編 (R01UH0896)

RL78/G23-64p Fast Prototyping Board ユーザーズマニュアル (R20UT4814)

(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)

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	Feb.25.25	-	初版

## 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しないでください。入力信号や入出力プルアップ電源からの電流注入により、誤動作を引き起こしたり、異常電流が流れ内部素子を劣化させたりする場合があります。資料中に「電源オフ時における入力信号」についての記載のある製品は、その内容を守ってください。

### 4. 未使用端子の処理

未使用端子は、「未使用端子の処理」に従って処理してください。CMOS 製品の入力端子のインピーダンスは、一般に、ハイインピーダンスとなっています。未使用端子を開放状態で動作させると、誘導現象により、LSI 周辺のノイズが印加され、LSI 内部で貫通電流が流れたり、入力信号と認識されて誤動作を起こす恐れがあります。

### 5. クロックについて

リセット時は、クロックが安定した後、リセットを解除してください。プログラム実行中のクロック切り替え時は、切り替え先クロックが安定した後に切り替えてください。リセット時、外部発振子（または外部発振回路）を用いたクロックで動作を開始するシステムでは、クロックが十分安定した後、リセットを解除してください。また、プログラムの途中で外部発振子（または外部発振回路）を用いたクロックに切り替える場合は、切り替え先のクロックが十分安定してから切り替えてください。

### 6. 入力端子の印加波形

入力ノイズや反射波による波形歪みは誤動作の原因になりますので注意してください。CMOS 製品の入力がノイズなどに起因して、 $V_{IL}$  (Max.) から  $V_{IH}$  (Min.) までの領域にとどまるような場合は、誤動作を引き起こす恐れがあります。入力レベルが固定の場合はもちろん、 $V_{IL}$  (Max.) から  $V_{IH}$  (Min.) までの領域を通過する遷移期間中にチャタリングノイズなどが入らないように使用してください。

### 7. リザーブアドレス（予約領域）のアクセス禁止

リザーブアドレス（予約領域）のアクセスを禁止します。アドレス領域には、将来の拡張機能用に割り付けられている リザーブアドレス（予約領域）があります。これらのアドレスをアクセスしたときの動作については、保証できませんので、アクセスしないようにしてください。

### 8. 製品間の相違について

型名の異なる製品に変更する場合は、製品型名ごとにシステム評価試験を実施してください。同じグループのマイコンでも型名が違えば、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、ノイズ輻射量などが異なる場合があります。型名が違う製品に変更する場合は、個々の製品ごとにシステム評価試験を実施してください。

## ご注意書き

1. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。回路、ソフトウェアおよびこれらに関連する情報を使用する場合、お客様の責任において、お客様の機器・システムを設計ください。これらの使用に起因して生じた損害（お客様または第三者いずれに生じた損害も含まれます。以下同じです。）に関し、当社は、一切その責任を負いません。
  2. 当社製品または本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、著作権その他の知的財産権に対する侵害またはこれらに関する紛争について、当社は、何らの保証を行うものではなく、また責任を負うものではありません。
  3. 当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
  4. 当社製品を組み込んだ製品の輸出入、製造、販売、利用、配布その他の行為を行うにあたり、第三者保有の技術の利用に関するライセンスが必要となる場合、当該ライセンス取得の判断および取得はお客様の責任において行ってください。
  5. 当社製品を、全部または一部を問わず、改造、変更、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、変更、複製、リバースエンジニアリング等により生じた損害に関し、当社は、一切その責任を負いません。
  6. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。  
標準水準： コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、産業用ロボット等  
高品質水準： 輸送機器（自動車、電車、船舶等）、交通管制（信号）、大規模通信機器、金融端末基幹システム、各種安全制御装置等  
当社製品は、データシート等により高信頼性、Harsh environment 向け製品と定義しているものを除き、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（宇宙機器と、海底中継器、原子力制御システム、航空機制御システム、プラント基幹システム、軍事機器等）に使用されることを意図しておらず、これらの用途に使用することは想定していません。たとえ、当社が想定していない用途に当社製品を使用したことにより損害が生じても、当社は一切その責任を負いません。
  7. あらゆる半導体製品は、外部攻撃からの安全性を 100%保証されているわけではありません。当社ハードウェア/ソフトウェア製品にはセキュリティ対策が組み込まれているものもありますが、これによって、当社は、セキュリティ脆弱性または侵害（当社製品または当社製品が使用されているシステムに対する不正アクセス・不正使用を含みますが、これに限られません。）から生じる責任を負うものではありません。当社は、当社製品または当社製品が使用されたあらゆるシステムが、不正な改変、攻撃、ウイルス、干渉、ハッキング、データの破壊または窃盗その他の不正な侵入行為（「脆弱性問題」といいます。）によって影響を受けないことを保証しません。当社は、脆弱性問題に起因したまたはこれに関連して生じた損害について、一切責任を負いません。また、法令において認められる限りにおいて、本資料および当社ハードウェア/ソフトウェア製品について、商品性および特定目的との合致に関する保証ならびに第三者の権利を侵害しないことの保証を含め、明示または黙示のいかなる保証も行いません。
  8. 当社製品をご使用の際は、最新の製品情報（データシート、ユーザーズマニュアル、アプリケーションノート、信頼性ハンドブックに記載の「半導体デバイスの使用上の一般的な注意事項」等）をご確認の上、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他指定条件の範囲内でご使用ください。指定条件の範囲を超えて当社製品をご使用された場合の故障、誤動作の不具合および事故につきましては、当社は、一切その責任を負いません。
  9. 当社は、当社製品の品質および信頼性の向上に努めていますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は、データシート等において高信頼性、Harsh environment 向け製品と定義しているものを除き、耐放射線設計を行っておりません。仮に当社製品の故障または誤動作が生じた場合であっても、人身事故、火災事故その他社会的損害等を生じさせないよう、お客様の責任において、冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、お客様の機器・システムとしての出荷保証を行ってください。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
  10. 当社製品の環境適合性等の詳細につきましては、製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。かかる法令を遵守しないことにより生じた損害に関して、当社は、一切その責任を負いません。
  11. 当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器・システムに使用することはできません。当社製品および技術を輸出、販売または移転等する場合は、「外国為替及び外国貿易法」その他日本国および適用される外国の輸出管理関連法規を遵守し、それらの定めるところに従い必要な手続きを行ってください。
  12. お客様が当社製品を第三者に転売等される場合には、事前に当該第三者に対して、本ご注意書き記載の諸条件を通知する責任を負うものいたします。
  13. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを禁じます。
  14. 本資料に記載されている内容または当社製品についてご不明な点がございましたら、当社の営業担当者までお問合せください。
- 注 1. 本資料において使用されている「当社」とは、ルネサス エレクトロニクス株式会社およびルネサス エレクトロニクス株式会社が直接的、間接的に支配する会社をいいます。
- 注 2. 本資料において使用されている「当社製品」とは、注 1 において定義された当社の開発、製造製品をいいます。

(Rev.5.0-1 2020.10)

## 本社所在地

〒135-0061 東京都江東区豊洲 3-2-24（豊洲フォレシア）

[www.renesas.com](http://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/](http://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。