

# ナノローバー取扱説明書

～ ユーザープログラム編 ～

(2019.07.25)

このたびは、ナノローバーをお買い求めいただき、誠にありがとうございます。本ドキュメントをよくお読みいただき、操作を行ってください。

## 目次

1	はじめに / 注意事項.....	2
2	Arduino IDE による開発.....	3
2.1	Arduino IDE による開発環境の構築 .....	3
2.2	スケッチの書込み方法.....	5
3	スケッチで使用可能な関数について .....	8
3.1	エンコーダについて .....	10

## 1 はじめに / 注意事項

本ドキュメントは、ナノローバーをお客様独自にプログラミングして使用いただく際の取扱説明書です。以下の注意事項を守り、安全に十分配慮してご使用ください。

本製品の使用にあたっては下記注意事項に従い、正しくご使用ください。

- 「ナノローバー取扱説明書」をご確認いただいた上で、本ドキュメントの記載内容を実施してください。
- 本製品を使用または本製品で使用するアプリケーション設計はお客様が自らの責任において行うものであり、結果アプリケーションが人または物、本製品に重大な損害を発生させた場合、製造元ならびに販売元は何らの責任を負わないことをご了承の上で本製品をご利用ください。
- 本製品を動作させる際は、目を離さず、その場を離れたりしないでください。
- 本体、制御基板などに強い衝撃を与えないでください。
- 本体、制御基板などから煙が発生した場合、すぐに電源をお切りください。
- 本製品を幼児の近くで使用したり、幼児の手の届くところに保管したりしないでください
- 基板上の端子（金属部分）に触れると静電気により故障する恐れがあります。
- 基板上の端子同士が金属などでショートすると、過電流により故障する可能性があります。
- 壁などにぶつかり、進めない状態でモータを回転させる（モータがロック状態になる）とモータに過電流が流れ、モータ、制御基板の破損につながる可能性があります。
- 本製品の使用においては、弊社製ではない様々なソフトウェアやライブラリを使用することがあります。それら弊社製ではないソフトウェアやライブラリの動作や入手性については、弊社では一切保証いたしかねます。

## 2 Arduino IDE による開発

本製品は、Wi-Fi / Bluetooth の 2 種の無線通信に対応していますが、複数の無線通信規格を同時に使用することは仕様上できません。使用する無線通信規格の設定や、Wi-Fi の SSID/パスワードの設定には、Arduino IDE を用いたスケッチの書き換えが必要となります。

本章では、Arduino IDE による開発環境の構築と、スケッチの書き込み方法について解説します。

### 2.1 Arduino IDE による開発環境の構築

本製品の Arduino IDE による開発は、Arduino IDE のボードマネージャが使用可能かつ、USB シリアルによるスケッチの書き込みが可能なデバイスであれば、OS 等を問わずに行うことが可能です。ROS を使用する場合は、ROS を動作させる Linux 系デバイスにも、開発環境を構築してください。ただし、Raspberry Pi など ARM 系の CPU を持つデバイスではコンパイラが対応しておらず、開発することはできません。

開発環境の構築手順は以下の通りです。

#### ① Arduino IDE のインストール

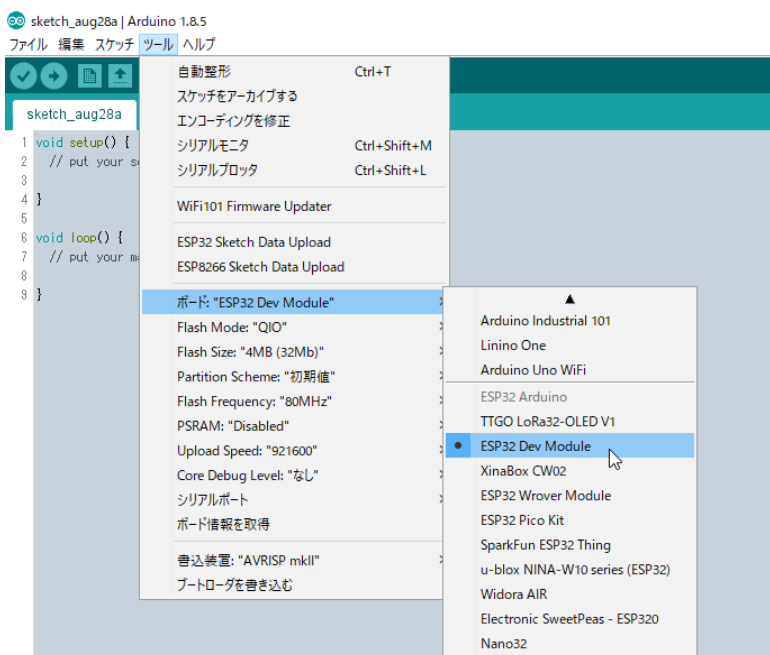
[Arduino.cc](https://www.arduino.cc) を参照し、Arduino IDE をインストールしてください。ナノローバーはバージョン 1.8.9 にて動作確認を行っております。

#### ② arduino-esp32 のセットアップ

ESP32-WROOM-32 を Arduino IDE から使用できるようにするため、ボードマネージャを使用してハードウェアライブラリのインストールを行います。[arduino-esp32 のプロジェクトページ](#)を参照し、インストールを行ってください。

ナノローバーはバージョン 1.0.2 で動作確認を行っております。1.0.2 以外がインストールされた場合は、バージョン選択から 1.0.2 を選択してください。

ハードウェアライブラリが正常にインストールされた場合、下図のように ESP32 系のボードを選択することができるようになります。



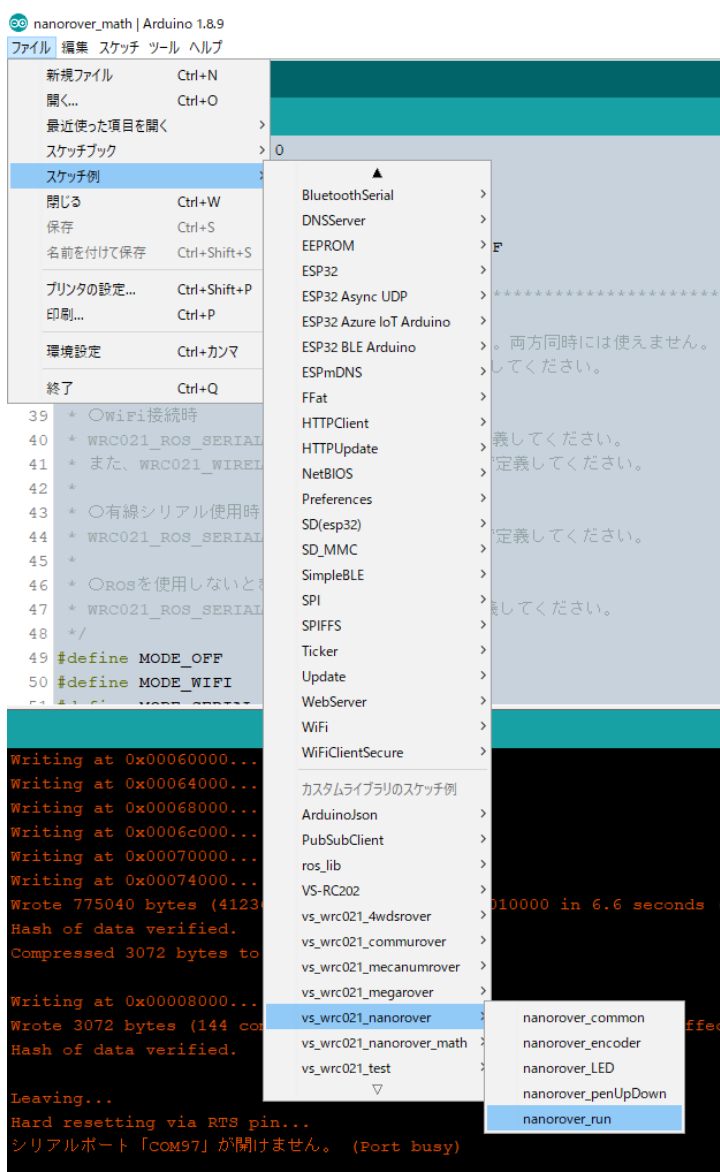
### ③ ナノローバー用ライブラリのインストール

ナノローバードキュメントフォルダ内の Arduino 用ライブラリフォルダ内にある `ros_lib` フォルダおよび `vs_wrc021_nanorover` フォルダを、Arduino の `libraries` フォルダにコピーします。

Arduino の `libraries` フォルダのパスは、Windows のデフォルト設定では以下の位置になります。

`C:\Users\自分の PC のユーザー名\Documents\Arduino\libraries`

Arduino IDE を再起動後下図のように、メニュー>ファイル>スケッチ例に、`ros_lib` および `vs_wrc021_nanorover` が表示され、いくつかのサンプルスケッチが選択可能であればインストールは成功しています。



※表示内容はその他のライブラリのインストール状況などにより変化します。

## 2.2 スケッチの書込み方法

スケッチの書込み方法を以下に示します。

### ① ナノローバーと PC の接続

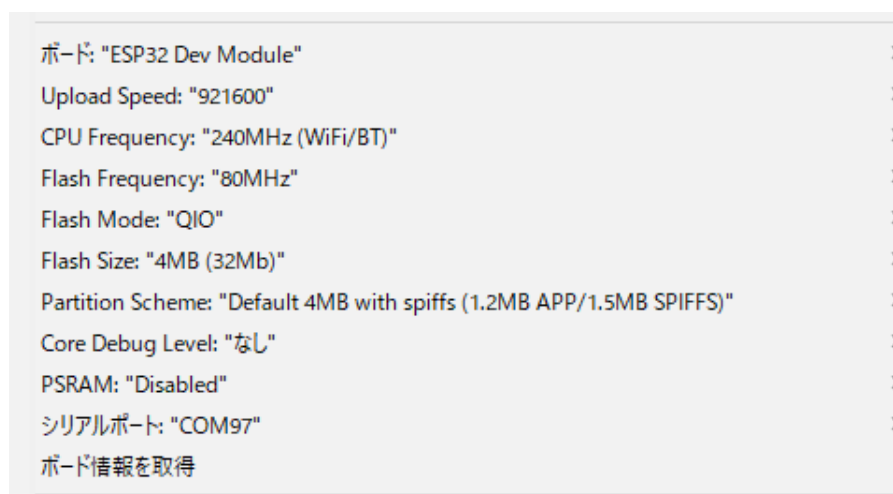
ナノローバーと PC とを、USB ケーブルで接続してください。接続後、ナノローバーの LED が点灯することを確認してください。LED が点灯しない場合は、正しく接続できていませんので、PC と USB ケーブルが正しく接続されていることをご確認のうえ、再度接続をお試しください。

### ② スケッチを開く

ここでは例として、Arduino IDE のメニューから、ファイル>スケッチ例>vs\_wrc021\_nanorover>nanorover\_run を開いてください。nanorover\_run の機能については、スケッチ内のコメントを参照してください。

### ③ 書込み設定

Arduino IDE のメニューから、ツール>ボード:"\*\*\*\*"を開き、ESP32 Dev Module を選択してください。その後、下図と同様に書込み設定を行ってください。なお、シリアルポートは自動的に選択される場合があります。



※よく書込みに失敗する場合は、Upload Speed を 115200 に設定してください。

### ④ 書込みの実行

Arduino IDE のメニューから、ツール>シリアルポートを開き、適切なポートを指定し、スケッチの書込みボタンを押して書込みを実行してください。初回のコンパイルや Arduino IDE を再起動した直後のコンパイルでは、処理に長時間かかることがありますが、Arduino IDE の仕様です。

書込み成功時に Arduino IDE に表示されるメッセージを次に記載します。

```
Writing at 0x000a8000... (97 %)
Writing at 0x000ac000... (100 %)
Wrote 1173184 bytes (654167 compressed) at 0x00010000 in 10.9 seconds (effective 857.8
kbit/s)...
Hash of data verified.
Compressed 3072 bytes to 144...

Writing at 0x00008000... (100 %)
Wrote 3072 bytes (144 compressed) at 0x00008000 in 0.0 seconds (effective 1445.7 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
```

上記のように、書込みが 100%まで行われてリセットが行われていれば、正常です。

書き込み時になんらかのエラーが発生した場合、エラー内容がメッセージとして表示されます。以下に、エラーメッセージ例と対処法を示します。

```
java.io.IOException: jssc.SerialPortException: Port name - COM15; Method name -
setEventsMask(); Exception type - Can't set mask.
    at processing.app.Serial.dispose(Serial.java:166)
    at processing.app.SerialMonitor.close(SerialMonitor.java:116)
    at processing.app.AbstractMonitor.suspend(AbstractMonitor.java:90)
    at processing.app.Editor$DefaultExportHandler.run(Editor.java:2160)
    at java.lang.Thread.run(Thread.java:748)
Caused by: jssc.SerialPortException: Port name - COM15; Method name - setEventsMask();
Exception type - Can't set mask.
    at jssc.SerialPort.setEventsMask(SerialPort.java:279)
    at jssc.SerialPort.removeEventListener(SerialPort.java:1064)
    at jssc.SerialPort.closePort(SerialPort.java:1090)
    at processing.app.Serial.dispose(Serial.java:163)
    ... 4 more
```

シリアルモニタの初期化に失敗した場合に表示されます。シリアルモニタの初期化失敗は Arduino IDE のバグが原因です。もう一度書込みを実施することで正常に書き込めます。

Archiving built core (caching) in:

C:\Users\\*\*\*\AppData\Local\Temp\arduino\_cache\_512565\core\core\_espressif\_esp32\_esp32\_PartitionScheme\_default,FlashMode\_qio,FlashFreq\_80,FlashSize\_4M,UploadSpeed\_921600,DebugLevel\_none\_a799b7492e9b9d0e443eff87836edf1f.a

c:/users/\*\*\*/documents/arduino/hardware/espressif/esp32/tools/xtensa-esp32-elf/bin/./lib/gcc/xtensa-esp32-elf/5.2.0/././././xtensa-esp32-elf/bin/ld.exe: cannot open output file

C:\Users\\*\*\*\AppData\Local\Temp\arduino\_build\_777804\nanorover\_common.ino.elf:

Permission denied

collect2.exe: error: ld returned 1 exit status

**Permission denied** は、権限不足示すエラーですが、発生原因は分かっていません。大抵の場合、もう一度書込みを実行することで正常に書き込むことができます。



### 3 スケッチで使用可能な関数について

ナノローバーでは、タイヤの制御などが簡単にできるよう、いくつかの関数を用意しています。これらの関数を使用することで、ナノローバーを簡単に走行させることができます。以下に関数の一覧を示します。

関数	wheelRun(long spdL, long spdR)
概要	タイヤを指定速度で回転させます
引数と戻値	[引数] 左右輪それぞれへの速度指令値[mm/s] [戻値] なし
使用例	<pre>wheelRun(-20,20);    //20mm/s で前進 delay(2000);          //2 秒間待機  wheelRun(0,0);        //両輪停止 delay(1000);          //1 秒待機</pre>

関数	readEnc (long *encL, long *encR)
概要	現在のエンコーダ値を取得します
引数と戻値	[引数] 左右輪のエンコーダ値を格納する変数のポインタ [戻値] なし
使用例	<pre>while(1){     readEnc(&amp;encL, &amp;encR); //エンコーダ値の取得      //ENC_PER_MM は標準状態での 1mm あたりのエンコーダ値     if(encR/ENC_PER_MM &gt; 50.0){ //右車輪が 50mm 以上進んだら         break;                //while ループから出る     }     delay(10); }</pre>

関数	clearEnc ()
概要	エンコーダ値をクリアして 0 にします。
引数と戻値	[引数] なし [戻値] なし
使用例	<pre>readEnc(&amp;encL, &amp;encR); //エンコーダ値の取得  if(encL &gt; 1000    encR &gt; 1000){ //どちらかのエンコーダ値が 1000 以上なら     clearEnc();                //エンコーダ値をリセットする }</pre>

関数	penUpDown(int isUp)
概要	ペンを持ち上げる or 下げる
引数と戻値	[引数] 0 → ペンダウン / 1 → ペンアップ [戻値] なし
使用例	<pre>penUoDown(0);    //ペンダウン  wheelRun(-20,20); //20mm/s で前進 delay(2000);      //2 秒間待機 wheelRun(0,0);    //両輪停止  penUoDown(1);    //ペンアップ</pre>

関数	ledSet(uint8_t ch, uint8_t r, uint8_t g, uint8_t b)
概要	ペンを持ち上げる or 下げる
引数と戻値	[引数] 0 → ペンダウン / 1 → ペンアップ [戻値] なし
使用例	<pre>penUoDown(0);    //ペンダウン  wheelRun(-20,20); //20mm/s で前進 delay(2000);      //2 秒間待機 wheelRun(0,0);    //両輪停止  penUoDown(1);    //ペンアップ</pre>

各関数を使用したサンプルスケッチが、vs\_wrc021\_nanorover ライブラリのスケッチ例に含まれています。Arduino IDE のメニューからファイル＞スケッチ例＞vs\_wrc021\_nanorover を開き、各スケッチをご利用ください。サンプルスケッチ一覧を以下に示します。

名称	機能
nanorover_common	ナノローバーを無線通信または ROS で操作する際に使用する
nanorover_run	ナノローバーを走行させる
nanorover_penUpDown	ペンアップダウン機能を使用する
nanorover_encoder	エンコーダを使用して走行させる
nanorover_LED	LED を点灯させる

### 3.1 エンコーダについて

ナノローバーのタイヤに使用されているモータにはエンコーダが取り付けられており、`readEnc()`関数を使用することで、簡単に値を取得することができます。

エンコーダの分解能は、3ppr（モータ 1 回転に 3 パルス）です。

ギアの減速比は 199.008:1 のため、タイヤ 1 周では 594.012 パルスとなります。

タイヤの直径は約 41.0mm、タイヤ 1 周は約 128.8mm となります。

エンコーダ出力値 `Mrenc` は 4 通倍出力のため、距離は以下の数式により計算できます。

$$(\text{車輪移動距離[mm]}) = \frac{Mrenc}{3 \times 199.008 \times 4} \times 41.0 \times \pi$$

#### 商品に関するお問い合わせ

TEL: 06-4808-8701

FAX: 06-4808-8702

E-mail: infodesk@vstone.co.jp

受付時間 : 9:00~18:00 (土日祝日は除く)

**ヴイストーン株式会社**

**www.vstone.co.jp**

〒555-0012 大阪市西淀川区御幣島 2-15-28