

4WDS ローバーX40A

取扱説明書

(2025.03.12)

このたびは、4WDS ローバーX40A 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	各部詳細・機能 .....	3
2.1	電源投入方法 .....	4
3	寸法について .....	5
4	システム構成 .....	6
4.1	システム構成図 .....	6
4.2	バンパーセンサの番号について .....	7
5	ハードウェアについて .....	8
5.1	バッテリーの充電 .....	8
5.2	制御基板 VS-WRC058 について .....	10
5.3	ハブ基板 VS-CN028 について .....	12
5.4	エンコーダによる車輪移動距離の測定について .....	13
5.5	座標系の定義 .....	13
6	ステアリング軸の挙動について .....	14
7	Arduino IDE による開発 .....	15
6.1	Arduino IDE による開発環境の構築 .....	15
6.2	スケッチの書き込み方法 .....	17
8	無線コントローラーでの操作方法 .....	21
9	Wi-Fi の接続方法 .....	23
8.1	Wi-Fi 接続設定と接続の確認 .....	23
10	コマンドによるメモリマップの読み書き .....	24
11	HTTP リクエストによるメモリマップの読み書き .....	26
12	その他、制御・開発方法について .....	27
11.1	ROS からの制御 .....	27
11.2	Windows からの制御サンプル .....	27
11.3	Arduino ライブラリ vs_wrc058_fwdsrover .....	27

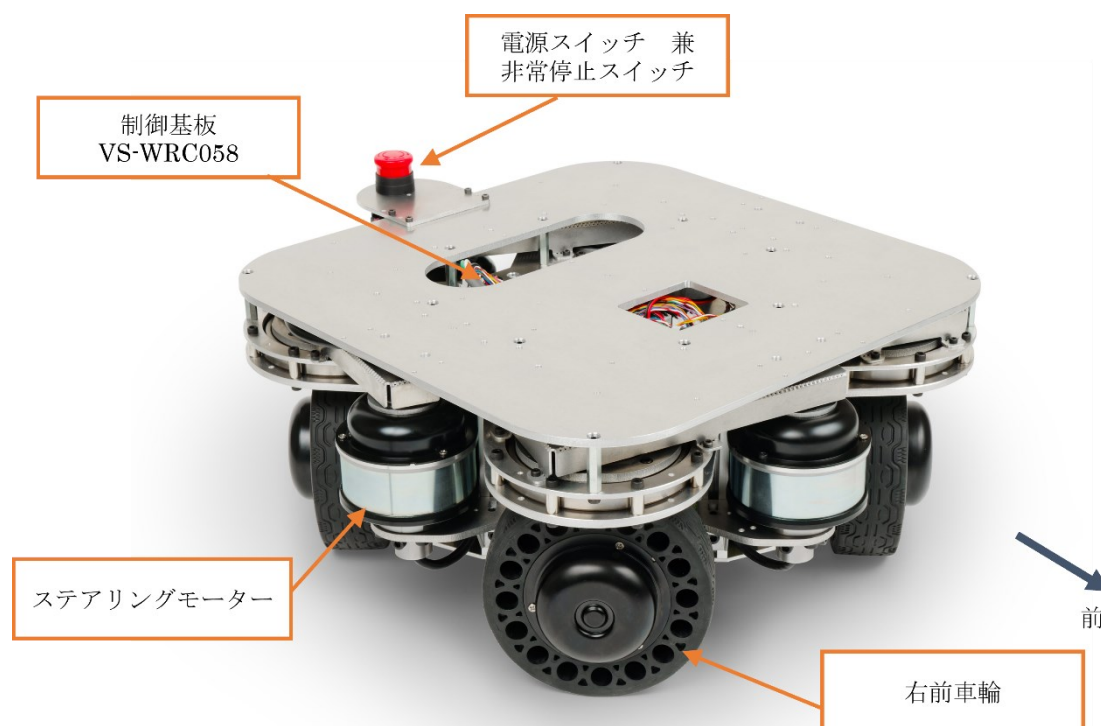
## 1 はじめに / 注意事項

本書は、4WDS ローバーX40A 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。

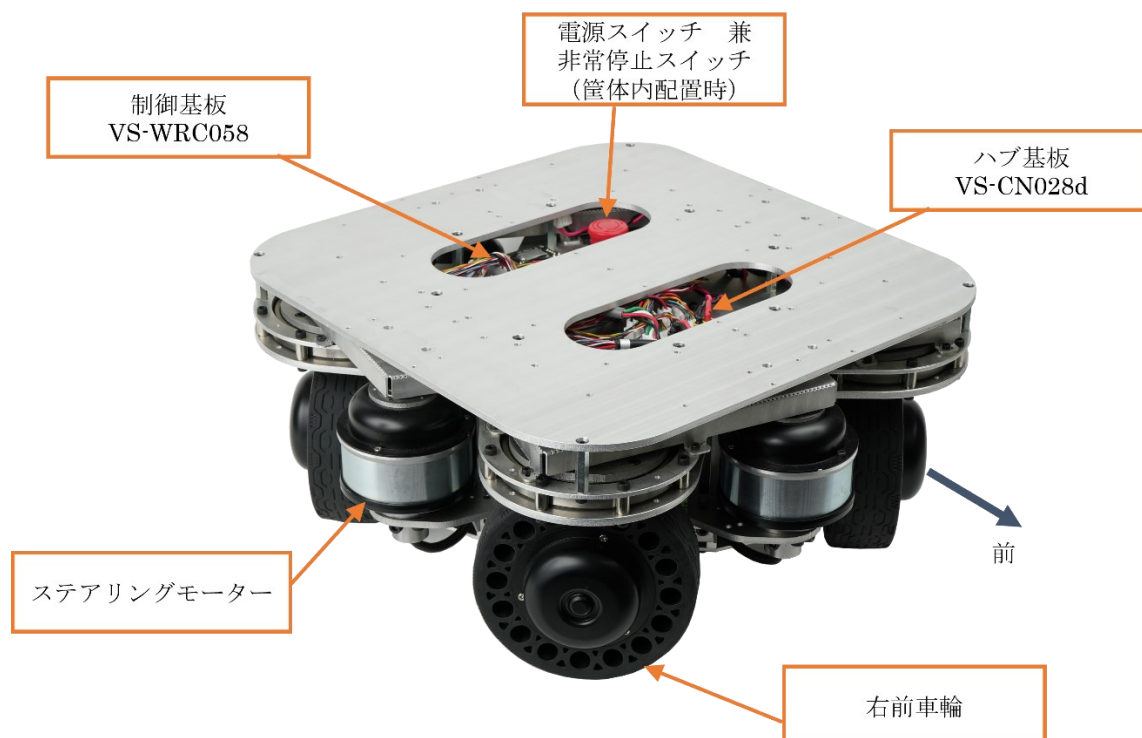
本製品の使用にあたっては下記注意事項に従い、正しくご使用ください。

- 本製品は、人の生命にかかわる用途として設計、製造されたものではありません。また、国内外の法規ならびに規制に違反する用途に使用することは固く禁じます。お客様がこのような用途に本製品をご使用になって何らかの損害が発生しても、製造元ならびに販売元は何らの責任を負いません。
- 本製品は、ロボット等アプリケーション設計・開発に十分な知識と経験を有するプロフェッショナル向けの製品です。本製品を用いたアプリケーション設計・開発ならびに動作作成などは十分習熟したプロフェッショナルが行ってください。
- 本製品を仕様に記載されている範囲外で使用された場合に何らかの損害が発生しても、製造元ならびに販売元は何らの責任を負いません。
- 水中や多湿、極端な高温・低温状態、粉じんなど本製品に悪影響を及ぼすおそれがある環境では使用しないでください。使用環境に起因する誤作動や生じた損害について、製造元ならびに販売元は何らの責任を負いません。
- 本製品を組み込んだロボット等の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。
- 本製品を分解・改造された結果生じるいかなる損害にも、製造元ならびに販売元は何らの責任を負いません。
- モータが起動した際の初期動作や動作作成などでお客様や周囲の人および物に重大な損害を発生させるおそれがあります。ロボット等のアプリケーションの組立・調整、モーション作成は周囲の安全を十分に確保した上で行ってください。
- 本製品は研究開発向け製品です。産業用に設計されたものではありません。
- 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。
- 動作中、基板上の素子が高温になることがありますので、絶対に触れないでください。
- 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。
- 制御基板のコネクタに大きな力が加わると、コネクタが破損する可能性があります。USB ケーブルを、ローバー本体に固定されていない PC 等に接続して使用される際などは特にご注意くださいませ。

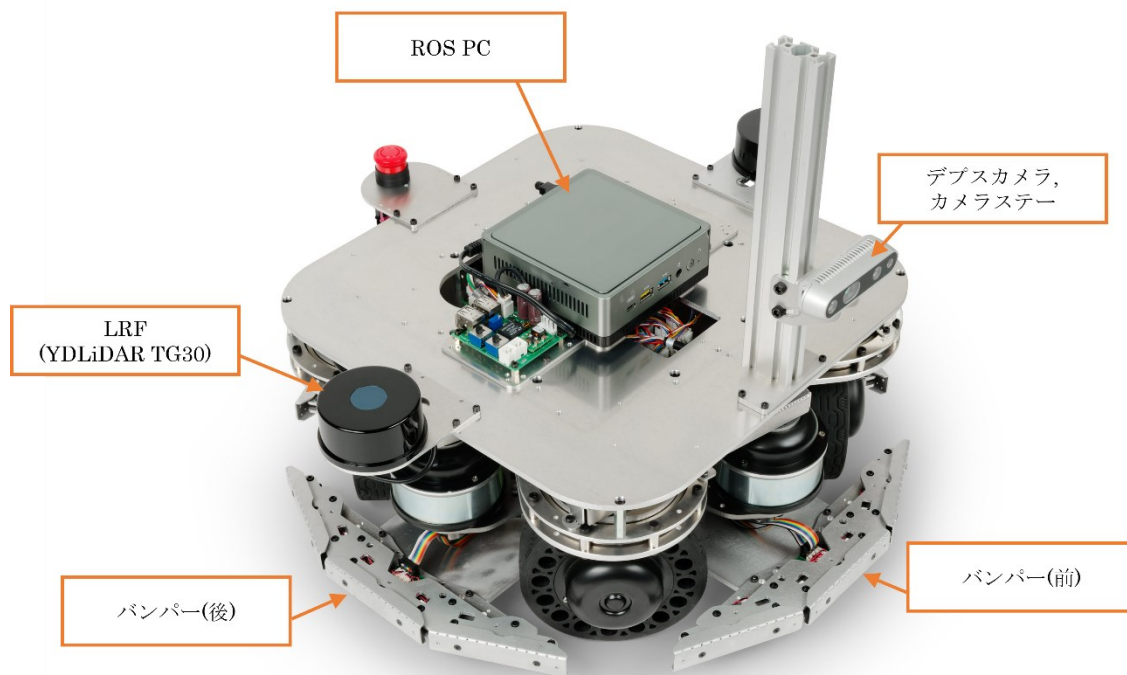
## 2 各部詳細・機能



2024 年 2 月までに出荷の機体



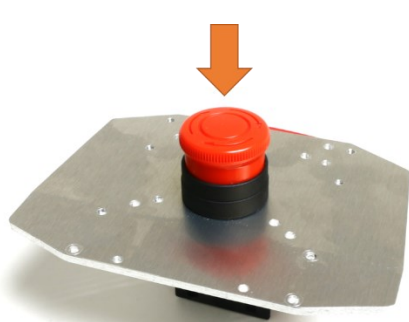
2024 年 3 月以降に出荷の機体



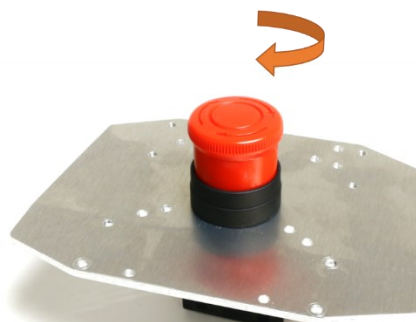
※オプション搭載時

## 2.1 電源投入方法

電源スイッチ 兼 非常停止スイッチを解除方向に回していただくことで、電源が ON になります。電源スイッチ 兼 非常停止スイッチを押し込んでいただくことで、電源が OFF になります。



電源 OFF (遮断) 状態

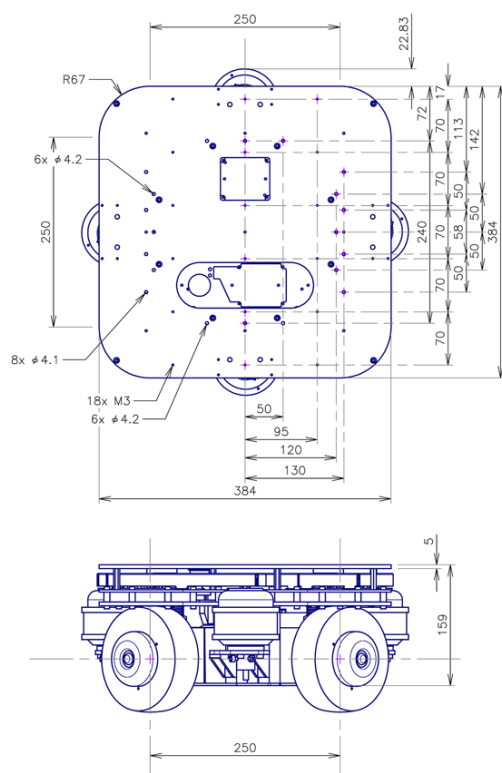


電源 ON (通電) 状態

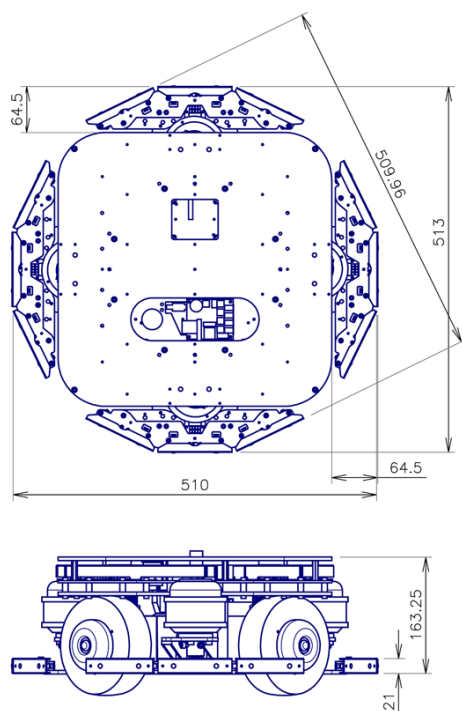
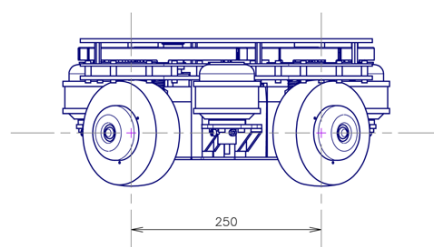
### 【注意】

電源 ON の状態で長時間放置すると、バッテリーが過放電状態となり、動作や充電が行えなくなります。使用しないときは電源を OFF にしてください。

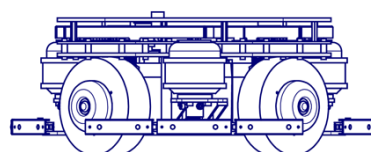
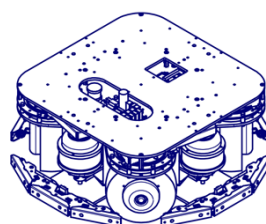
### 3 寸法について



CADデータは提供しておりません

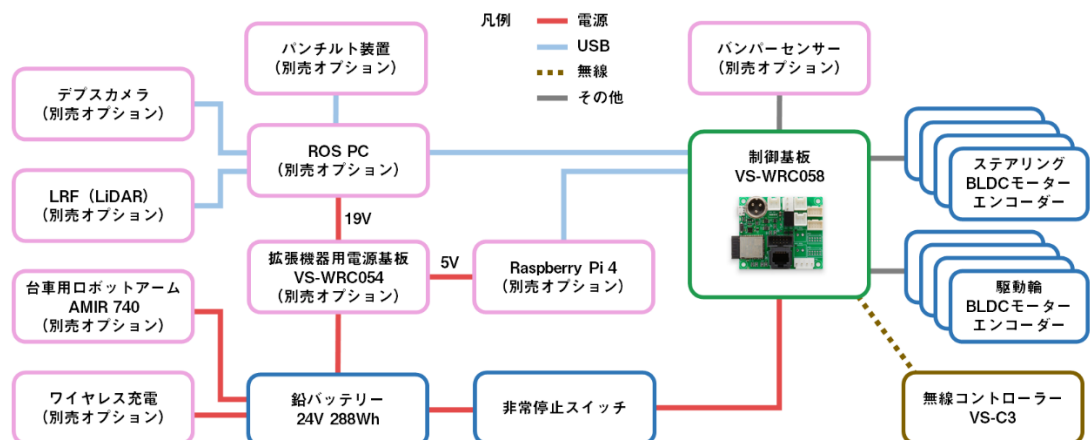


CADデータは提供しておりません

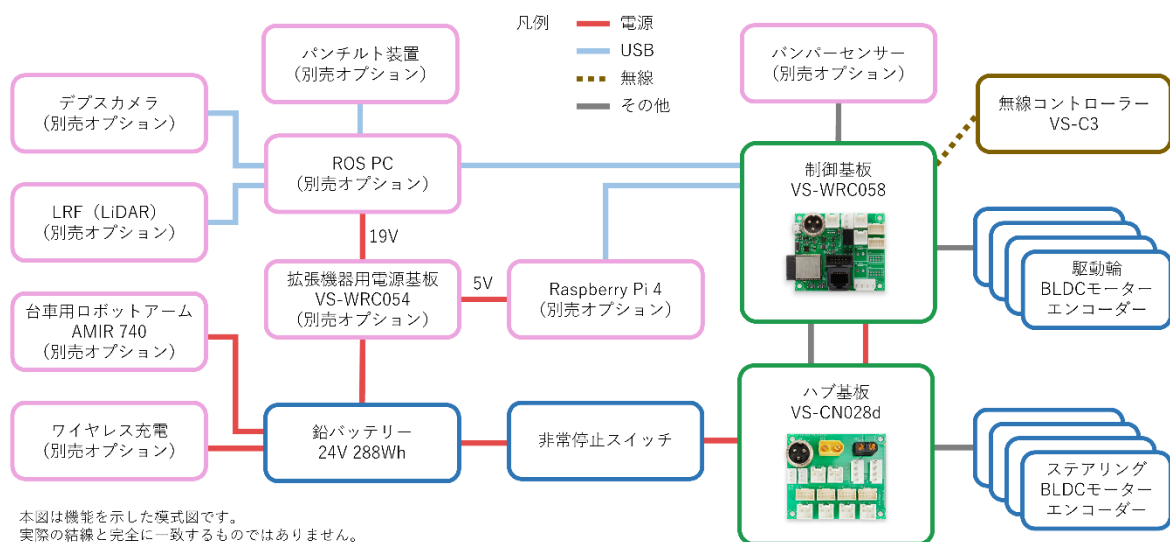


## 4 システム構成

### 4.1 システム構成図

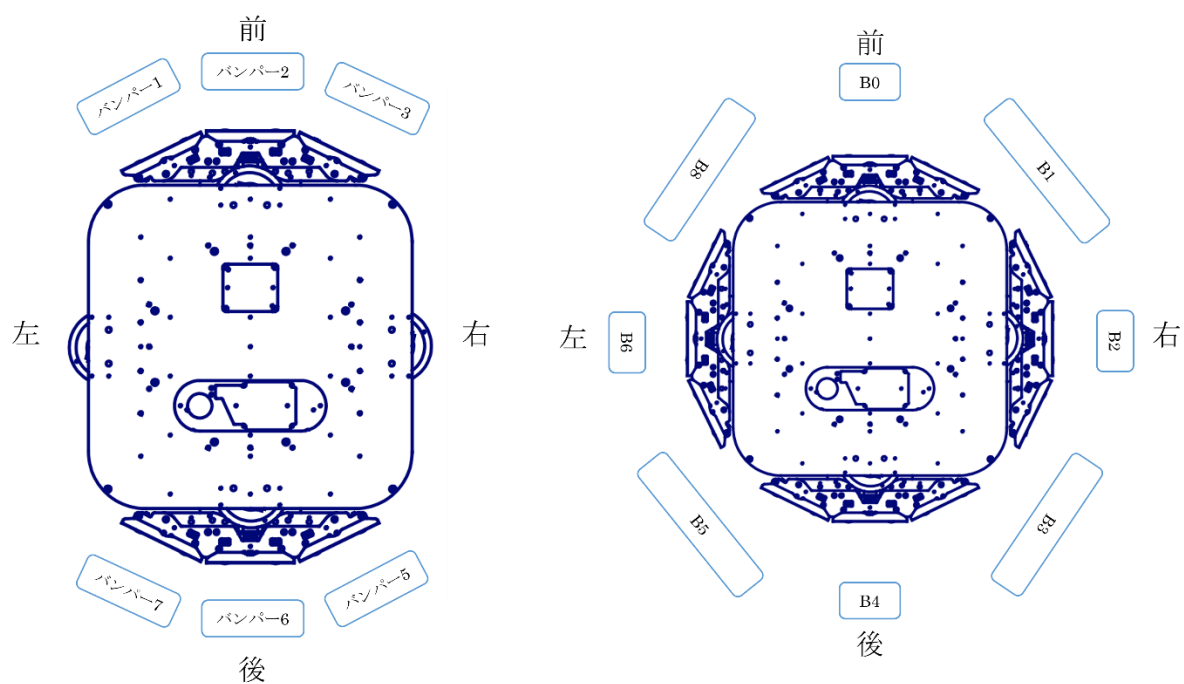


2024 年 2 月までに出荷の機体



2024 年 3 月以降に出荷の機体

#### 4.2 バンパーセンサの番号について



※バンパー4は欠番です。



## 5 ハードウェアについて

### 5.1 バッテリーの充電

バッテリーの充電は、製品に付属の充電器を用い、以下の手順で行ってください。充電器の機種がご購入時期によって異なります。お手元の充電器に合った手順で充電を行ってください。

#### 〔PCX-3000（青い充電器）の場合〕

- ① 充電器の裏蓋を開け、電源ケーブルと充電ケーブルを取り出してください。
- ② 充電器の電源スイッチが **OFF** になっていることを確認し、充電ケーブルをハブ基板 **VS-CN028**（24年2月以前に出荷の機体では制御基板 **VS-WRC058**）に接続してください。接続するコネクタの位置については 5.3 節（**VS-WRC058** に接続する場合は 5.2 節）を確認してください。また、向きを間違えないようにご注意ください。
- ③ 充電器の電源ケーブルをコンセントに接続してください。
- ④ 充電器の電源スイッチを **ON** にし、表示部に「**CHE**」と表示されることを確認してください。数秒後に充電が自動でスタートします。
- ⑤ 充電がスタートされたら、表示が以下のようにになっていることを確認してください。



- ⑥ 充電を終了するときは、電源スイッチを **OFF** にし、充電コネクタから充電ケーブルを取り外してください。

※充電中にコンセントや充電ケーブルを外すと大変危険です。必ず **OFF** スイッチで充電を停止してから取り外ししてください。

※充電中は目を離さず、その場を離れたりしないでください。

※充電の際は周りに火気や可燃物が無いことをよく確認してください。

#### 【注意】

充電を行っていない状態で充電器が制御基板に接続されていると、バッテリーから充電器に放電されてしまい、過放電となる危険があります。

充電完了後は、充電器を制御基板から速やかに外してください。

〔S120 (黒い充電器) の場合〕



S120 Battery Charger

- ① 同梱の AC ケーブルを充電器本体に接続し、AC プラグをコンセントに接続してください。
- ② 充電器のランプが緑の連続点灯または、緑と赤の交互の点滅状態になることを確認してください。
- ③ 充電ケーブルをハブ基板 VS-CN028 (24 年 2 月以前に出荷の機体では制御基板 VS-WRC058) に接続してください。接続するコネクタの位置については 5.3 節 (VS-WRC058 に接続する場合は 5.2 節) を確認してください。また、向きを間違えないようにご注意ください。
- ④ 接続されると自動的に充電が始まります。充電器のランプが赤く連続点灯していることを確認してください。
- ⑤ 充電が完了するとランプが緑色に変化するか、消灯します。ケーブルを充電コネクタから取り外してください。充電完了直前に、充電中と充電終了の状態が交互に繰り返されることがありますが、正常です。

※充電中は目を離さず、その場を離れたりしないでください。

※充電の際は周りに火気や可燃物が無いことをよく確認してください。

**【注意】**

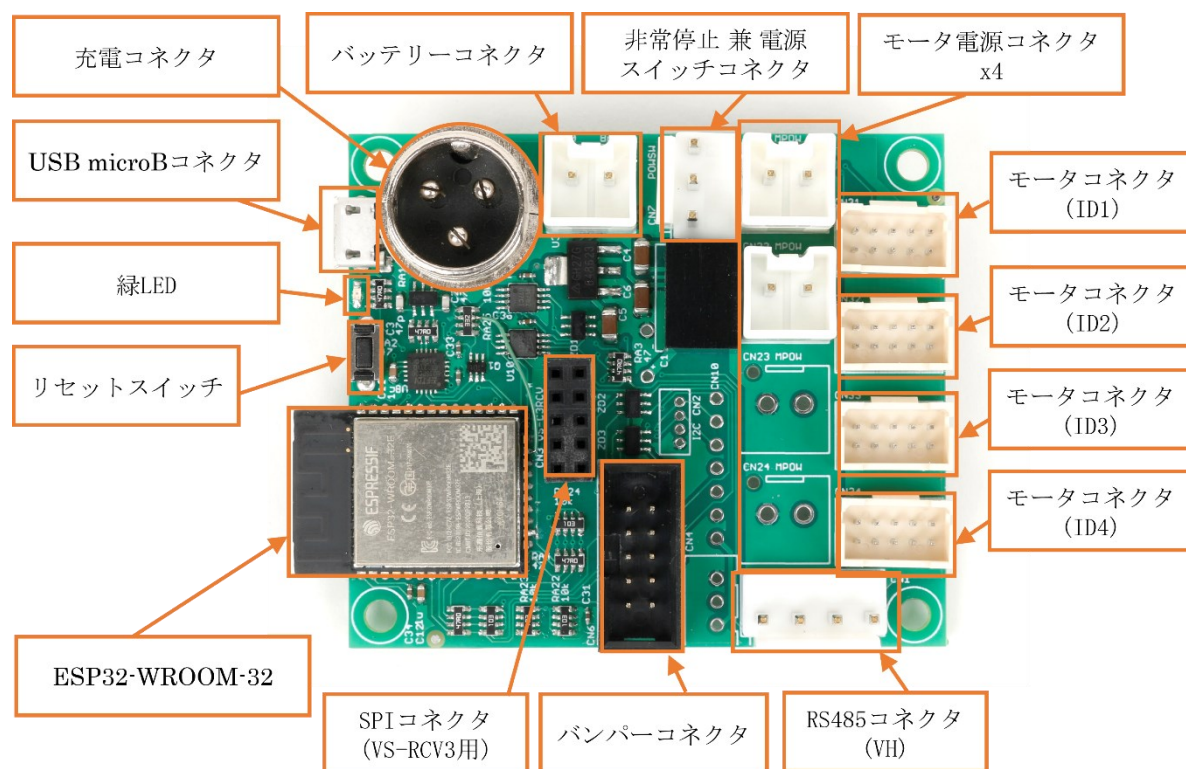
充電器がコンセントに接続されていない状態で制御基板に接続されていると、バッテリーから充電器に放電されてしまい、過放電となる危険があります。

充電器をコンセントに接続していない状態で、基板に接続しないでください。

## 5.2 制御基板 VS-WRC058 について

VS-WRC058 は ESP32-WROOM-32 を搭載した Arduino 互換の制御ボードです。Arduino IDE を使用してプログラムの作成、書込みが可能です。Arduino IDE を用いた開発方法については、6 章を参照してください。

コネクタの配置等を下図に示します。



VS-WRC058c1 コネクター一覧

- 非常停止 兼 電源スイッチコネクタ  
非常停止 兼 電源スイッチからのケーブルを接続するコネクタです。**24 年 2 月以降出荷の機体では使用できません。**
- リセットスイッチ  
押下することで基板をリセットします。
- 緑 LED  
電源 ON 時に点灯しますが、外部からの通信などの入力があった際は消灯します。
- バッテリーコネクタ  
バッテリー接続用のコネクタです。**24 年 2 月以降出荷の機体では使用できません。**
- 充電コネクタ  
充電ケーブル接続用のコネクタです。**24 年 2 月以降出荷の機体では使用できません。**
- USB microB コネクタ  
Arduino スケッチの書込みや、USB シリアル通信、ROS で制御する際に使用します。

USB ホストの機能はありません。

○ モータ電源コネクタ

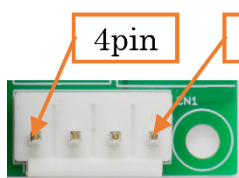
モータの電源ケーブルを接続し、電源を供給するコネクタです。バッテリーコネクタと形状が同一ですが、接続を間違えると故障する恐れがございますのでご注意ください。

○ モータコネクタ (ID1-4)

モータの通信ケーブルを接続するコネクタです。どのコネクタに接続するかでモータの ID が決まります。4WDS ローバーX40A では、ID1 に右前モータを、ID2 に左前モータを、ID3 に左後ろモータを、ID4 に右後ろモータを接続してください。

○ RS485 コネクタ(VH)

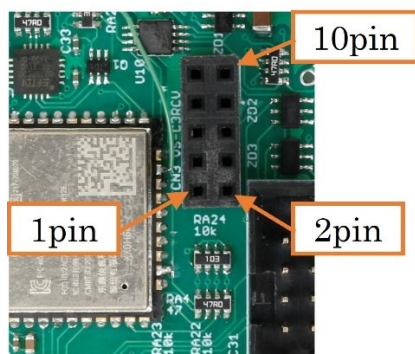
接続先のデバイスの電源を 4WDS ローバーのバッテリーから直接供給可能な RS485 コネクタです。バッテリー電圧が直接出力されているので取扱いに注意してください。RS485\_A, RS485\_B は 3.3V です。ピン配置は以下の通りです。



- 1pin : RS485\_B
- 2pin : RS485\_A
- 3pin : VBATT(バッテリー電圧)
- 4pin : GND

○ SPI(VS-C3 受信機)コネクタ

標準状態では、VS-C3 の受信機である VS-RCV3 が接続されています。SPI 通信のコネクタとしても使用可能です。ピン配置は以下の通りです。



- 1pin : NC
- 2pin : NC
- 3pin : MISO
- 4pin : MOSI
- 5pin : SS
- 6pin : SCK
- 7pin : NC
- 8pin : +5V
- 9pin : +3.3V
- 10pin : GND

○ バンパー接続コネクタ

バンパーオプションのご注文で付属するバンパーを接続するコネクタです。

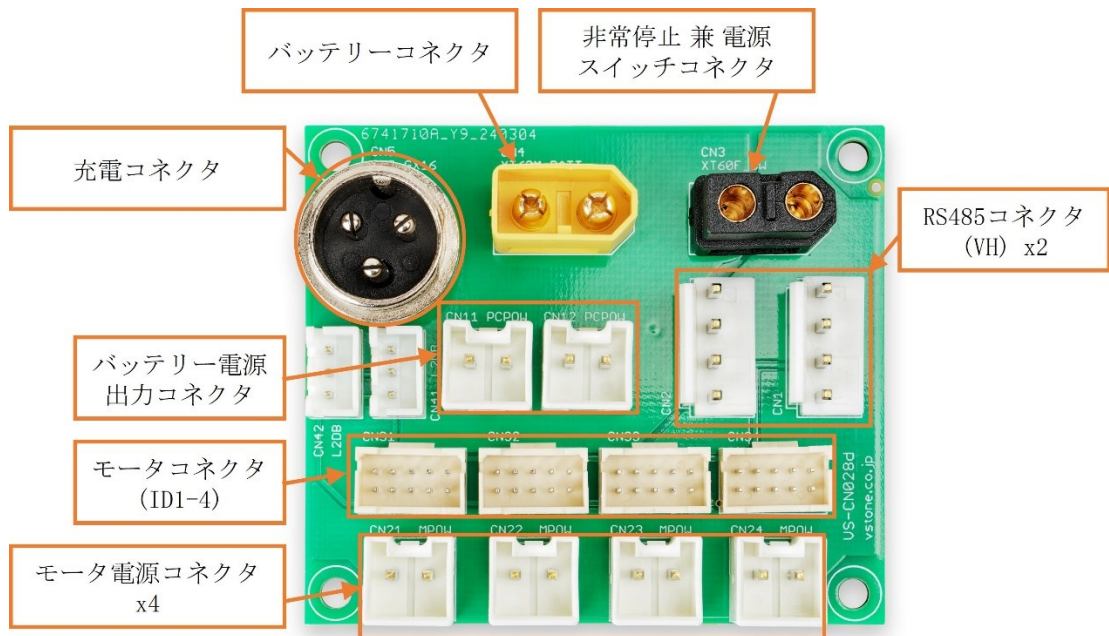
**【注意】**

ローバー本体外に配置した PC と制御基板とを USB ケーブルで接続した状態で走行させ、ケーブルに引っ張られた USB コネクタが破損する事例が発生しております。USB コネクタに大きな力がかからないよう、十分ご注意の上運用してください。



### 5.3 ハブ基板 VS-CN028 について

VS-CN028 は、制御基板やオプション等への電源の分配、制御基板から制御するモータ個数を増やすためのハブの役割を持つ基板です。コネクタの配置等を下図に示します。



VS-CN028 コネクター一覧

#### ○ 非常停止 兼 電源スイッチコネクタ

非常停止 兼 電源スイッチからのケーブルを接続するコネクタです。24 年 2 月以降出荷の機体では VS-CN028 上のコネクタを使用してください。

#### ○ バッテリーコネクタ

バッテリー接続用のコネクタです。24 年 2 月以降出荷の機体では VS-CN028 上のコネクタを使用してください。

#### ○ 充電コネクタ

充電ケーブル接続用のコネクタです。24 年 2 月以降出荷の機体では VS-CN028 上のコネクタを使用してください。

#### ○ モータ電源コネクタ

VS-WRC058 のモータ電源コネクタと同仕様です。

#### ○ モータコネクタ (ID1-4)

VS-WRC058 のモータコネクタと同仕様です。

#### ○ RS285 コネクタ (VH)

VS-WRC058 の RS485 コネクタと同仕様です。

#### ○ バッテリー電源出力コネクタ

バッテリーの電力がそのまま供給されているコネクタです。非常停止 兼 電源スイッチの状態に関わらず電力が供給されますのでご注意ください。オプションの拡張機器用電源基板の接続が可能です。

#### 5.4 エンコーダによる車輪移動距離の測定について

エンコーダの分解能は、4096ppr（モータ 1 回転に 4096 パルス）です。

ダイレクトドライブモータのため、タイヤ 1 周でも 4096 パルスとなります。

タイヤの直径は約 140mm、タイヤ 1 周は約 440mm となります。

よって距離は以下の数式により計算できます。

$$(\text{車輪移動距離}[\text{mm}]) = \frac{Menc}{4096} \times 140 \times \pi$$

#### 5.5 座標系の定義

4WDS ローバーのロボット座標系は x 軸の正方向を前方向、y 軸の正方向を左方向とする右手系で定義されています。

## 6 ステアリング軸の挙動について

### 6.1 起動時のキャリブレーション

4WDS ローバーは電源投入後、ステアリングのキャリブレーションを行います。キャリブレーション中はステアリング軸がゆっくりと回転します。また、標準のファームウェアでは外部からの操作は受け付けず、キャリブレーションをキャンセルすることはできません。後述の Arduino ライブラリを改造することでこれらの動作をキャンセルすることができますが、ステアリングが正常に動作せずロボットが故障する可能性がございますので十分にご注意ください。

キャリブレーション動作では、ステアリングをストッパーに当たるまで正回転/負回転させ、ステアリングの可動範囲および適切なセンター位置の割出しを行っています。キャリブレーション終了後は、ステアリングは自動でセンター位置に移動します。

### 6.2 タイヤ、ステアリング軸保護のための動作制限

ある程度の速度で走行中にステアリング操作を行うとタイヤやステアリング軸に大きな負荷がかかり、破損する可能性があります。

破損を防止するため、ステアリング軸および車輪の動作に制限があります。

- ・ 一定速度以上で走行している際には一定角度以上のステアリング軸の回転を行わない
- ・ 一定角度以上のステアリング軸の回転を行う際には、移動速度指令値を 0m/s とする

上記の制限のため、一定速度以上で走行中にステアリング軸を大きく変化させる必要がある際には、一度 4WDS ローバーは減速し、それからステアリング軸を回転させ、再度加速します。そのためカクカクとした動作に見えることがありますが、安全のための仕様となっております。

## 7 Arduino IDE による開発

制御基板 VS-WRC058 は Arduino IDE を用いて内部のスケッチ（ファームウェア）を書き換えることが可能です。Wi-Fi の設定や、モータの制御ゲインの変更、ROS の使用設定には、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。

### 6.1 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。

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

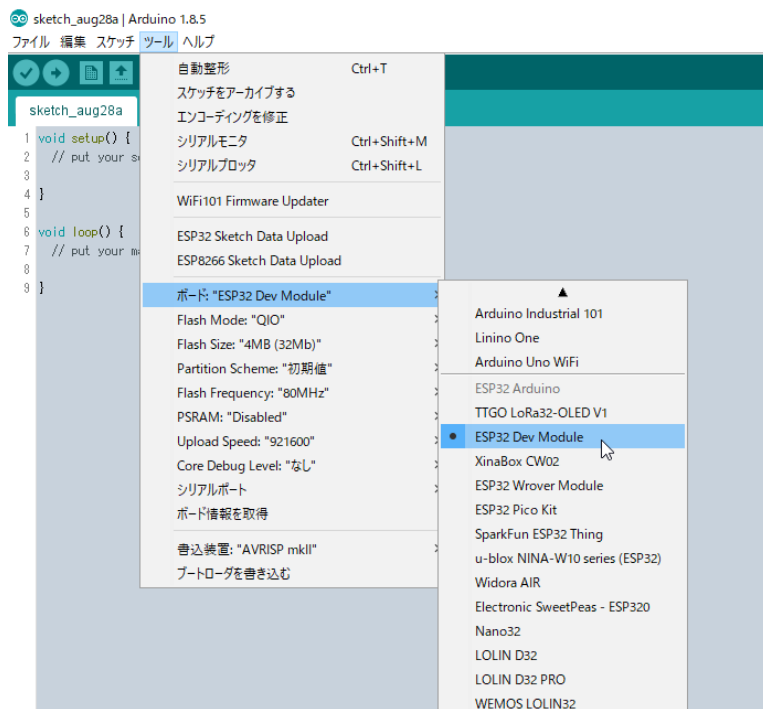
[Arduino.cc](https://arduino.cc) を参照し、Arduino IDE 1.8 系または 2.0 系をインストールしてください。以下の解説では 1.8 系の画像を使用していますが、2.0 系でも手順は同様です。

#### 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 のインストール方法解説ページ](#)を参照し、インストールを行ってください。

**4WDS ローバーはバージョン 2.0.13 で動作確認を行っています。2.0.13 以外がインストールされた場合は、バージョン選択から 2.0.13 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。

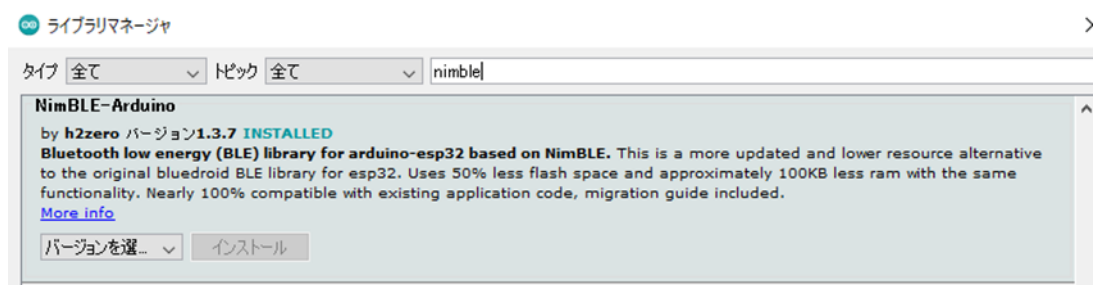




### ③ NimBLE-Arduino ライブラリの導入

Bluetooth の機能を使用するため、NimBLE-Arduino ライブラリを導入します。

メニューから、スケッチ>ライブラリのインクルード>ライブラリの管理を選択してください。ライブラリマネージャが開いたら、検索窓に「nimble」と入力し、NimBLE-Arduino が出てくることを確認してください。



**NimBLE-Arduino のバージョン 1.4.1 をインストールしてください。1.4.1 以外では動作しない場合がございます。上図のバージョン表示の箇所が、1.4.1 INSTALLED となれば問題ありません。**

### ④ VS-WRC058 用ライブラリの導入

4WDS ローバーSDK フォルダ内の Arduino ライブラリフォルダ内にある `ros_lib` フォルダおよび `vs_wrc058_fwdsrover` 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\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_wrc058_fwdsrover` が表示され、サンプルスケッチ `fwdsrover_x40a_common` 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。

## 6.2 スケッチの書き込み方法

スケッチの書き込み方法を以下に示します。

### ① 4WDS ローバーと PC の接続

VS-WRC058 と PC とを、USB ケーブルで接続してください。接続後、4WDS ローバーの電源を入れ、VS-WRC058 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、バッテリーコネクタが接続されていないか、バッテリーの電圧が低下しています。

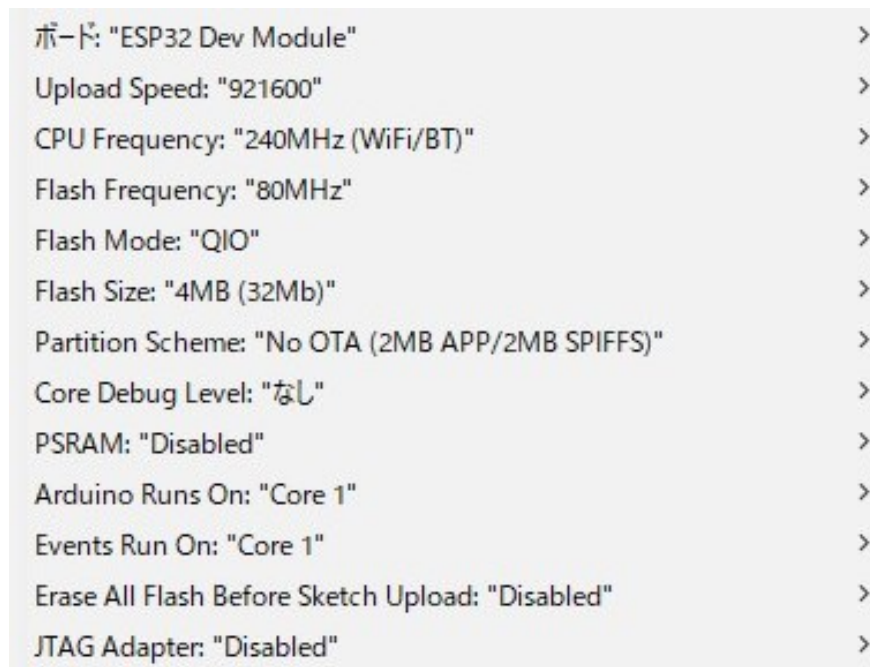
### ② スケッチを開く

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\_wrc058\_fwdsrover＞fwdsrover\_x40a\_common を開いてください。

fwdsrover\_x40a\_common 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 書き込み設定

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、下図と同様に書き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る他、ポート名称は環境によって変化いたします。



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

### ④ 書き込みの実行

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、スケッチの書き込みボタンを押して書き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、Arduino IDE の仕様です。

書き込み成功時に 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\fwdsrover\_common.ino.elf:

Permission denied

collect2.exe: error: ld returned 1 exit status

**Permission denied** は、権限不足を示すエラーですが、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。

## 8 無線コントローラーでの操作方法

4WDS ローバーには、無線コントローラー「VS-C3」付属しています。コントローラーを使用することで、PC 等のデバイスを介さなくても 4WDS ローバーを操縦することができます。4WDS ローバーを手動操作したい場合や、購入後の動作確認にお使いいただけます。

購入直後の 4WDS ローバーまたはサンプルスケッチの `fwdsrover_x40a_common` では VS-C3 コントローラーの入力が最優先で作用するようになっています。

以下に操作方法を示します。なお Arduino ライブラリ `vs_wrc058_fwdsrover` 内の `vs_wrc058_fwdsrover.cpp` を書き換えることにより、操作方法を変更することが可能です。



### ○ 縦移動モード、横移動モード、超信地旋回モード

アナログスティックを使った走行には、縦移動モードと横移動モード、超信地旋回モードの 3 つのモードがあります。R1 を押した際には横移動モードに、L1 を押した際には縦移動モードに、L2 を押した際には超信地旋回モードになり、それぞれ移動可能な方向が制限されます。

縦移動モードで移動可能な方向は、前後（x 軸方向）および x 軸から左右約 90 度の範囲です。真横には移動することはできず、超信地旋回も行えません。

横移動モードで移動可能な方向は、左右（y 軸方向）および y 軸から左右約 90 度の範囲です。直進方向には移動することができず、超信地旋回も行えません。

超信地旋回は超信地旋回モードでのみ行うことができます。

- △×○□ボタン

△ボタンを押すと現在の速度に+0.1m/s、×ボタンを押すと現在の速度に-0.1m/s されます。  
○,□ボタンを押すと 0.5rad/s で旋回します。旋回速度と並進移動速度は合成されます。

- 十字ボタン

各方向へ、0.3m/s の速度で走行します。複数のボタンを同時押しすると速度が合成されます。

- L1,L2,R1,R2 ボタン

アナログスティックの入力に対するセーフティー機能および最大並進移動速度と最大旋回速度の選択ボタン、移動モードの選択ボタンとして機能します。いずれかのボタンを押している間のみ、アナログスティックの入力が有効になります。L1 または L2、R1 と R2 を同時押しすることで、高速移動が可能です。

- 左右アナログスティック

左スティックの上下方向が前後への並進移動速度、左スティックの左右方向が左右への並進移動速度になります。

縦移動モードでは、右スティックの左右方向が旋回量の指令値となります。

横移動モードでは、右スティックの上下方向が旋回量の指令値となります。

超信地旋回モードでは、右スティックの左右方向が旋回量の指令値となります。

大きく倒すほど速く走行します。

## 9 Wi-Fi の接続方法

制御基板 VS-WRC058 は Wi-Fi に接続することができます。Wi-Fi に接続することで、離れた PC 等から 4WDS ローバーを制御することが可能です。なお、6 章の Arduino IDE による開発環境の構築をまだ行っていない場合は、そちらを先に終えてください。

### 8.1 Wi-Fi 接続設定と接続の確認

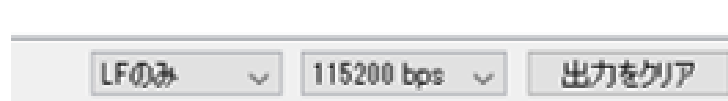
4WDS ローバーを Wi-Fi アクセスポイントに接続し、同じネットワーク内のデバイスからアクセスできるように設定します。以下の手順に従って操作を行ってください。

#### ① Wi-Fi 接続設定

サンプルスケッチ `fwdsrover_x40a_common` を開き、スケッチ内の指示に従って Wi-Fi の接続設定を行い、スケッチを書き込んでください。

#### ② IP アドレスの確認

Arduino IDE のシリアルモニタを開き、シリアルモニタの右下の選択リストから、改行コードおよび通信速度を以下の図の通り設定してください。



4WDS ローバーのリセットボタンを押してください。本体リセット後、指定した SSID のアクセスポイント接続を試みます。接続に成功すると、以下のようなメッセージがシリアルモニタに表示されます。

```
Connecting to ****
.....
WiFi connected.
IP address:
192.168.1.32
```

SSID が\*\*\*\*のアクセスポイントに接続したことが分かります。このとき 5 行目に表示されている IP アドレスが、4WDS ローバーに割り当てられた IP アドレスとなりますので控えておいてください。この場合は 192.168.1.32 です。

しばらく待ってもアクセスポイントに接続できない場合、4WDS ローバーを再度リセットしていただくか、各設定値に誤りがないかをご確認ください。また電波強度が弱いところでは安定して接続ができません。混信にもご注意ください。



## 10 コマンドによるメモリマップの読み書き

4WDS ローバーには 256byte のメモリマップが存在しています。メモリマップには各種制御に関する情報が書き込まれており、この値を読み取ったり、指令値を書き込んだりすることで 4WDS ローバーを制御することができます。

メモリマップはリトルエンディアンになっており、0x00～0xff までのアドレスを指定して各 byte にアクセスすることができます。メモリマップの詳細は、メモリマップ.pdf を参照してください。

Arduino ライブラリ `vs_wrc058_fwdsrover` では、USB からコマンドでメモリマップにアクセスできるように通信関数が実装されています。また、HTTP リクエストによる読み書きも可能で、10 章で解説します。

ここではコマンドの形式について解説します。コマンドはテキスト形式で送受信され、解説では 1 文字を[]で囲って表します。すなわち”abc”は[a] [b] [c]と記載します。また、[SP]はスペース、[LF]は改行コードです。

基板には ID が振られており、[ID]で表します。ID の一覧は下表の通りです。

ID	基板
0x10	VS-WRC058（制御基板）ホイールモータ
0x1F	VS-CN028（ハブ基板）ステアリングモータ

### 〔読み込み〕

読み込みコマンドは、メモリマップから値を読み込むコマンドです。コマンド形式は以下の通りです。なお、一度のコマンドで読み込めるバイト数は 63byte までです。

#### 読み込みコマンド形式

[r][ID] [SP] [アドレス H] [アドレス L] [SP] [バイト数 H] [バイト数 L] [LF]

アドレスには読み込み開始地点のメモリマップのアドレスを、バイト数には読み込むバイト数を記述します。例えば、VS-WRC058 のメモリマップのアドレス 0x10 から 0x13 までの 4byte を読み込む場合のコマンドは以下の通りです。

読み込みコマンド例：“r10 10 04\n”

## 〔書き込み〕

書き込みコマンドは、メモリマップに値を書き込むコマンドです。コマンド形式は以下の通りです。

### 書き込みコマンド形式

**[w][ID] [SP] [アドレス H] [アドレス L] [SP] [データ 1H] [データ 1L] [データ 2H] [データ 2L] [LF]**

アドレスには、書き込み開始地点のメモリマップのアドレスを、データにはメモリマップに書き込む値を記述します。メモリマップはリトルエンディアンになっています。

例えば、制御基板（VS-WRC058）のメモリマップアドレス 0x90 の MS16\_S\_XS は 2byte の整数型ですが、ここに 0x1234 を書き込む場合のコマンドは以下の通りです。

書き込みコマンド例：“w10 90 3412\n”

また、連続するアドレスであれば以下のように記述することでまとめて書き込むこともできます。例えば、メモリマップアドレス 0x20 の MU16\_FB\_VP0 に 0x1234 を、0x22 の MU16\_FB\_VP1 に 0x5678 を書き込む場合、以下のように記述することができます。

書き込みコマンド例；“w10 20 34127856\n”

## 11 HTTP リクエストによるメモリマップの読み書き

Arduino サンプル `vs_wrc058_fwdsrover` では、4WDS ローバーに対して GET リクエストを投げることで、メモリマップの読み書きを行うことができます。

### [読み込みリクエスト]

以下の形式で GET リクエストを投げることで、メモリマップから値を読み込むことができます。

読み込みリクエスト形式：

`http://4WDS ローバーIP アドレス/read?id=デバイス ID&addr=アドレス&length=バイト数`

デバイス ID に読み込む基板の ID (10 章参照) を、アドレスにメモリマップの読み込み開始アドレスを、バイト数に読み込むバイト数を記述します。いずれも 16 進数表記です。一度に読み込むことができるバイト数は最大 63byte です。

### [書き込みリクエスト]

以下の形式で GET リクエストを投げることで、メモリマップに値を書き込むことができます。

書き込みリクエスト形式：

`http://4WDS ローバーIP アドレス/write?id=デバイス ID&addr=アドレス&data=書き込みデータ`

デバイス ID に読み込む基板の ID (10 章参照) を、アドレスにメモリマップの書き込み開始アドレスを、書き込みデータに書き込むデータを 9 章の書き込みコマンド同様の形式で記述します。

例えば、制御基板 (VS-WRC058) のアドレス 0x20 (MU16\_FB\_VP0) に 0x1234 を、0x22 (MS16\_FB\_VP1) に 0x5678 を書き込む場合のリクエストは以下の通りです。

書き込みリクエスト例：

`http://4WDS ローバーIP アドレス/write?id=0x10&addr=20&data=34127856`

## 12 その他、制御・開発方法について

### 11.1 ROS からの制御

4WDS ローバーX40A は USB シリアルまたは Wi-Fi によって ROS と接続し、制御することが可能です。詳しくは、4WDS ローバーX40A\_ROS 取扱説明書をご確認ください。

### 11.2 Windows からの制御サンプル

Windows から USB シリアル接続で 4WDS ローバーを走行させるサンプルプログラムを添付しています。詳しくは Windows サンプルフォルダを確認してください。

### 11.3 Arduino ライブラリ vs\_wrc058\_fwdsrover

ライブラリの各関数についての説明は、フォルダ内のソースファイル、ヘッダファイルに記載されています。

#### 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