
SunFounder PiCar-V

www.sunfounder.com

2024 年 02 月 04 日

目次

第 1 章	PiCar-V について	1
1.1	部品一覧	2
1.2	前書き	10
1.3	車の組み立て	11
1.4	回路構築	24
1.5	Raspberry Pi を始めよう	28
1.6	サーボ構成	64
1.7	組み立てを続ける	66
1.8	旅に出よう！	72
1.9	付録	89
1.10	ありがとうございました	101
第 2 章	著作権表示	103

第 1 章

PiCar-V について

PiCar-V には、広角 USB ウェブカメラ、配線が少なくシンプルな 3 つのまったく新しい回路基板、調整可能な構造が内蔵している一部のアクリルプレートや車を制御するほとんどすべてのプラットフォームに適した新しいコードが装備されている。

本書では、ハードウェアとソフトウェアの両方を考え、説明、物理部品のイラストを使用して PiCar-V を構築する方法について説明する。これらのすべてがどのように機能するかを学ぶことを楽しむだろう。次のリンクをクリックすると、PDF ユーザーマニュアルを表示したり、コードのクローンを作成したりできます：https://github.com/sunfounder/SunFounder_PiCar-V/tree/V3.0。

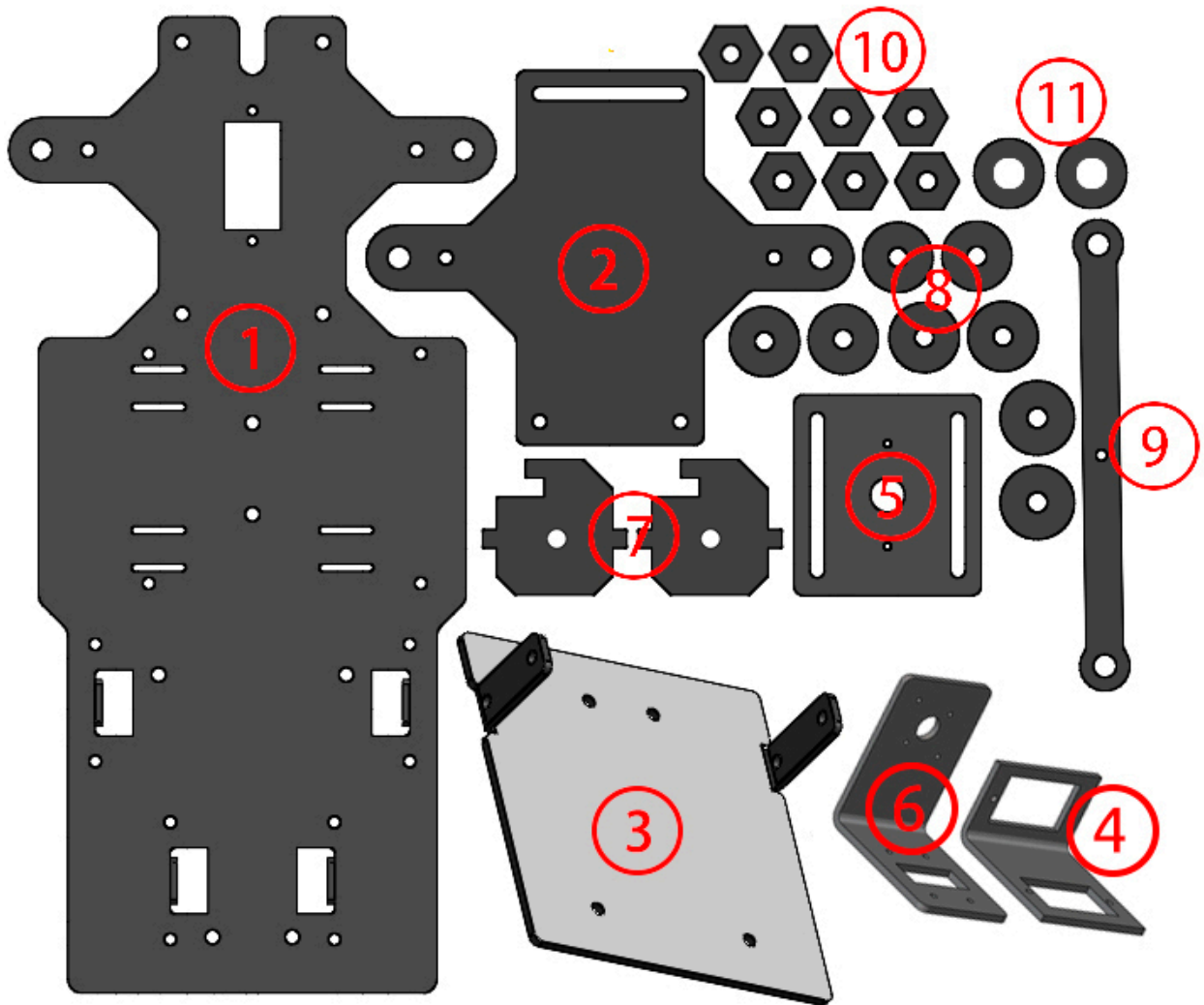
Github の上で弊社のページでお気軽に問い合わせください。

製品の関するご意見、またはマニュアルにない新しいプロジェクトを学習したい場合は、遠慮なくこちらまでご連絡ください。できるだけ早めにマニュアルに更新します。

サポートメールアドレスはこちらまで：cs@sunfounder.com

1.1 部品一覧

1.1.1 プレート



1. 上部プレート x 1
2. 前部ハーフプレート x 1
3. 後部ハーフプレート x 1
4. パン/チルトプレート x 1
5. パンとチルトのベースプレート x 1
6. カメラマウントプレート x 1
7. サーボコネクタプレート x 2

- 8. ベアリングシールド x 8
- 9. サーボリンクージプレート x 1
- 10. 六角前輪固定プレート x 8
- 11. ガスケットプレート x 2

注釈: ベアリングシールドはガスケットプレートに似ているが、ガスケットプレートの口径はベアリングシールドよりも大きくなっている。

1.1.2 サーボ x 3

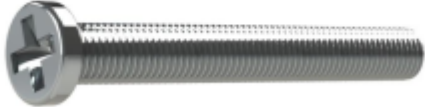


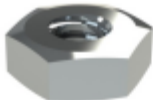
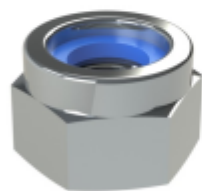
- 1. サーボ
- 2. 1 アームロッカーアーム
- 3. アームロッカーアーム
- 4. 4 アームロッカーアーム

5. ロッカーアーム固定ネジ

6. ロッカーアームねじ

1.1.3 メカニカルファスナー

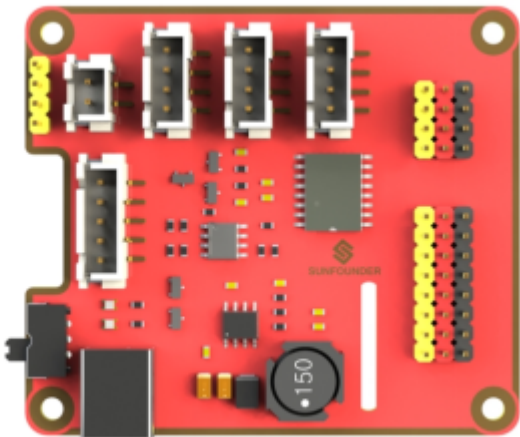
Name	Component	Qty.
M1.5x4 Self-tapping Screw		10
M2x8 Screw		6
M2.5x6 Screw		4
M2.5x12 Screw		8
M3x8 Screw		8
M3x8 Countersunk Screw		2
M3x10 Screw		7
M3x25 Screw		4
M4x25 Screw		2

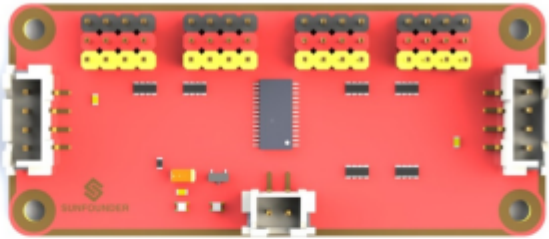
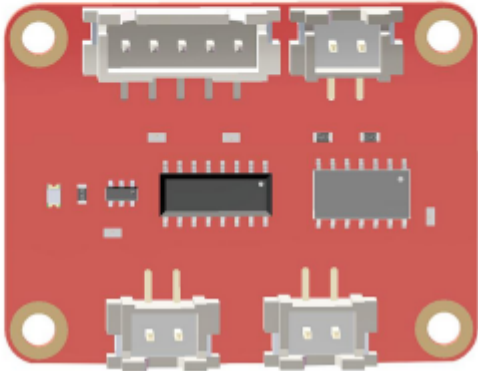
M2 Nut		6
M2.5 Nut		12
M3 Nut		21
M4 Self-locking Nut		2
M2.5x8 Copper Standoff		8
M3x25 Copper Standoff		8

1.1.4 配線

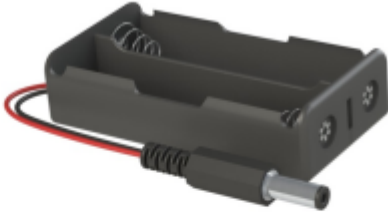
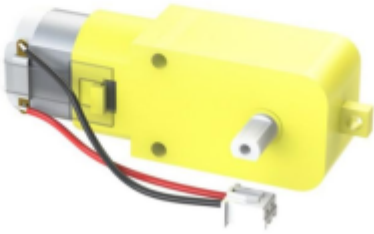
100mm HX2.54 5-Pin Jumper Wire		1
50mm HX-2.54 4-Pin Jumper Wire		1
50mm HX-2.54 2-Pin Jumper Wire		1
100mm HX-2.54 2-Pin Jumper Wire		1

1.1.5 PCB

Robot HATS		1
------------	--	---

PCA9685 PWM Driver		1
Motor Driver Module		1

1.1.6 その他の部品

2x18650 Battery Holder		1
DC Gear Motor		2

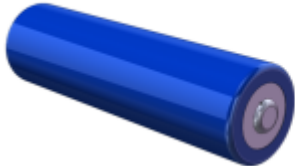
120° Wide-angle USB Camera		1
Rear Wheel		2
Front Wheel		2
Ribbon (30cm)		1

1.1.7 工具

Cross Screwdriver		1
Cross Socket Wrench		1
M2.5/M4 Small Wrench		1
M2/M3 Small Wrench		1

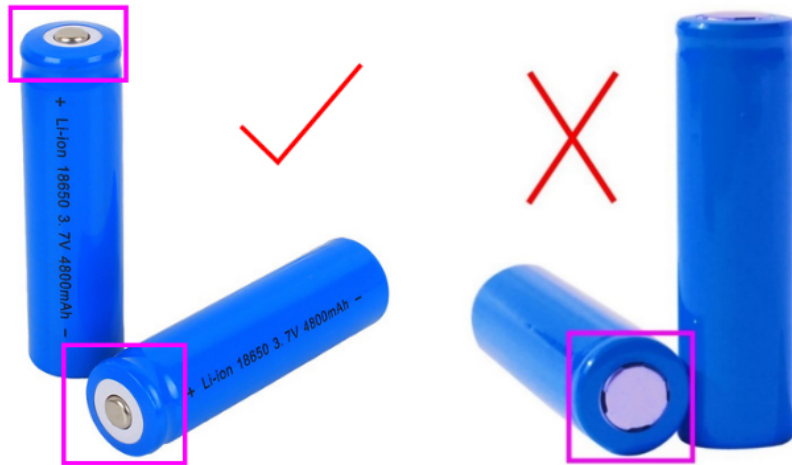
1.1.8 別途用意する必要な部品

このキットには以下の成分が含まれていません。

Component	Name	Qty.
	18650 3.7V Rechargeable Li-ion Battery	2

注釈:

1. 保護ボードなしの 18650 バッテリーを使用することを推奨します。他のバッテリーを使用すると、保護ボードの過電流保護により、製品の電源が切れて動作を停止する可能性があります。
2. 保護回路なしのバッテリーの場合は、バッテリーホルダーとの接続を確実にするために、アノードが膨らんでいる (正極が尖った) バッテリーを購入してください (以下を参照)。



3. 長時間使用するために、可能な限り最大電力のバッテリーを使用してください。
-

1.2 前書き

PiCar-V は以前のスマートビデオカーキットに基づいて開発された。新しい PCA9685 PWM ドライバー、小型で高性能の DC モータードライバーモジュール、電源とインターフェイスが統合されたロボット HAT、より鮮明な視界を備えた広角カメラ、いくつかの優れたホイールとタイヤなどが搭載されている。詳しくは後で説明する。

この新しいバージョンの基本的な機能は、PiCar-V をリモートで制御したり、キャプチャしたビデオデータをストリーミングしたりするなど、前のとほぼ同じである。それでは新規機能は何か？謎を解き明かそう！



1.3 車の組み立て

箱を開けてたくさんの部品を見るとワクワクしますか？あなたの忍耐を保ち、それを楽にしてください。次の手順の一部の詳細は注意深く観察する必要があることに注意してください。各ステップを終えた後、マニュアルの図に基づいて作業を再確認する必要があります。はじめましょう！

1.3.1 前輪

以下に示すように、M4x25 ネジ を一つの ステアリングコネクタプレート、3つの ベアリングシールド、3つの 六角前輪固定プレート、と一つの 前輪 を通して M4 セルフロックナット に挿入する：

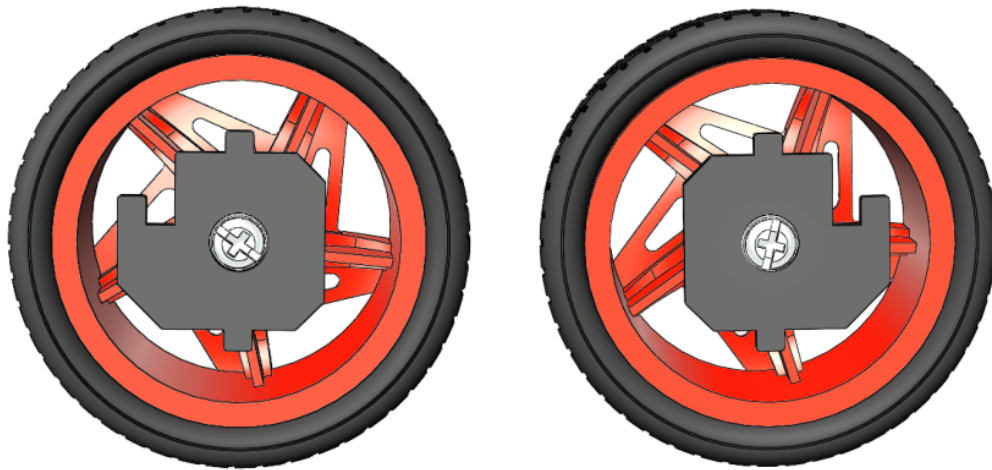


クロスソケットレンチを使用して M4 セルフロックナット を固定してからドライバーを使用して M4x25 ネジ を締める。



注釈：セルフロックナットはしっかりと締まっていることを確認してください。車輪とステアリングコネクタが動かなくなるまでネジを締め、次にネジを少し緩めて、ステアリングプレートだけが動くようにする。したがって、接続が緩すぎない場合、ホイールは柔軟に回転できる。

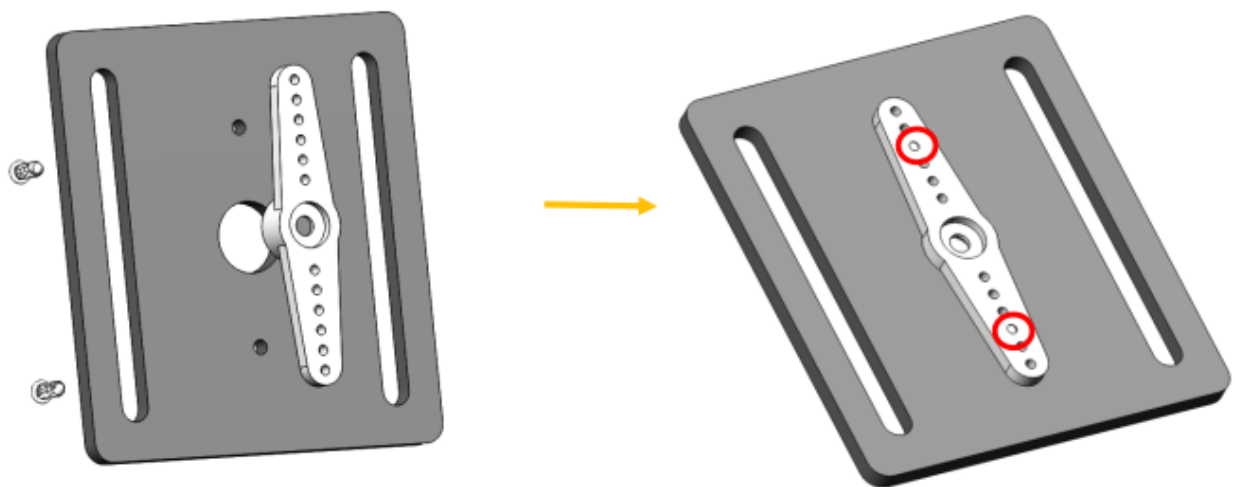
同じ方法で他の前輪を組み立てるが、車輪のステアリングコネクタプレートは前のものと対称であることを覚えておいてください。



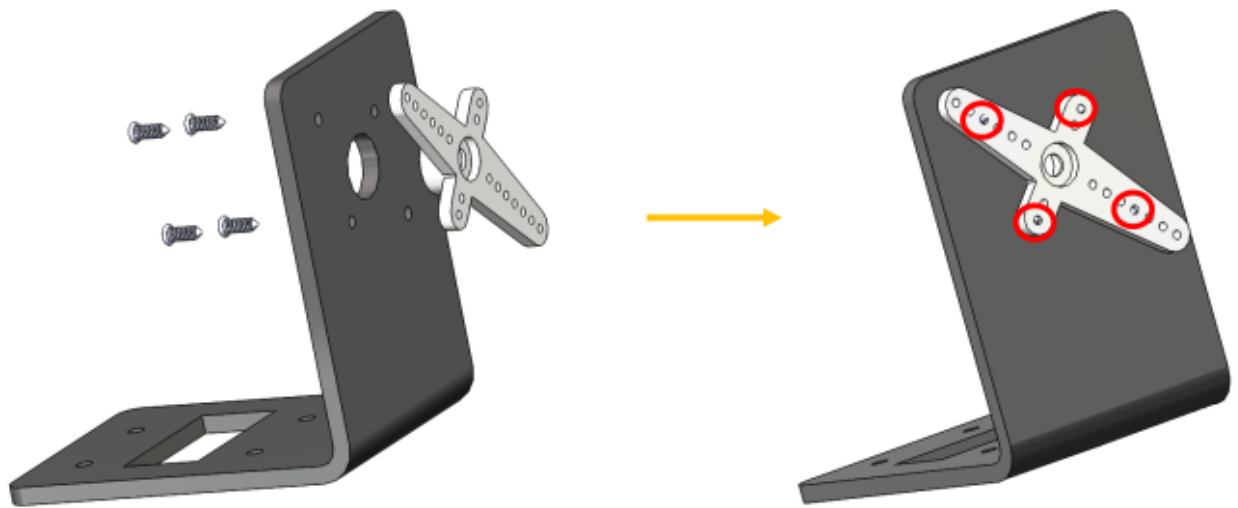
これで2つの前輪の組み立てが完了した。

1.3.2 パンとチルト

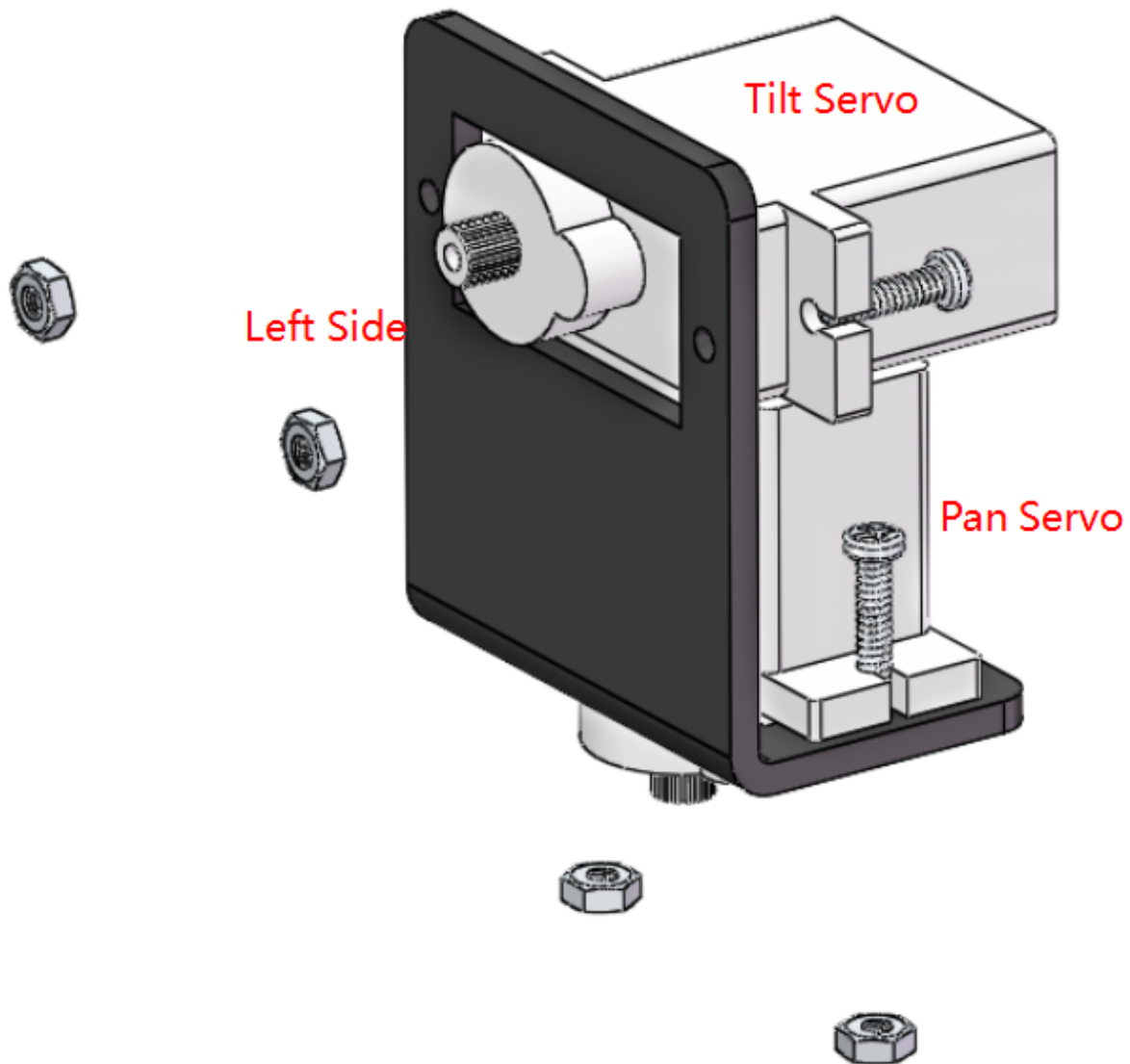
2 x アームロッカーアームを取出し、2M1.5 x 4 ネジを使って、パン-チルトベースプレートに取り付けます。(方向は次のように)。



カメラマウントプレートにも同様の操作を行ってください。



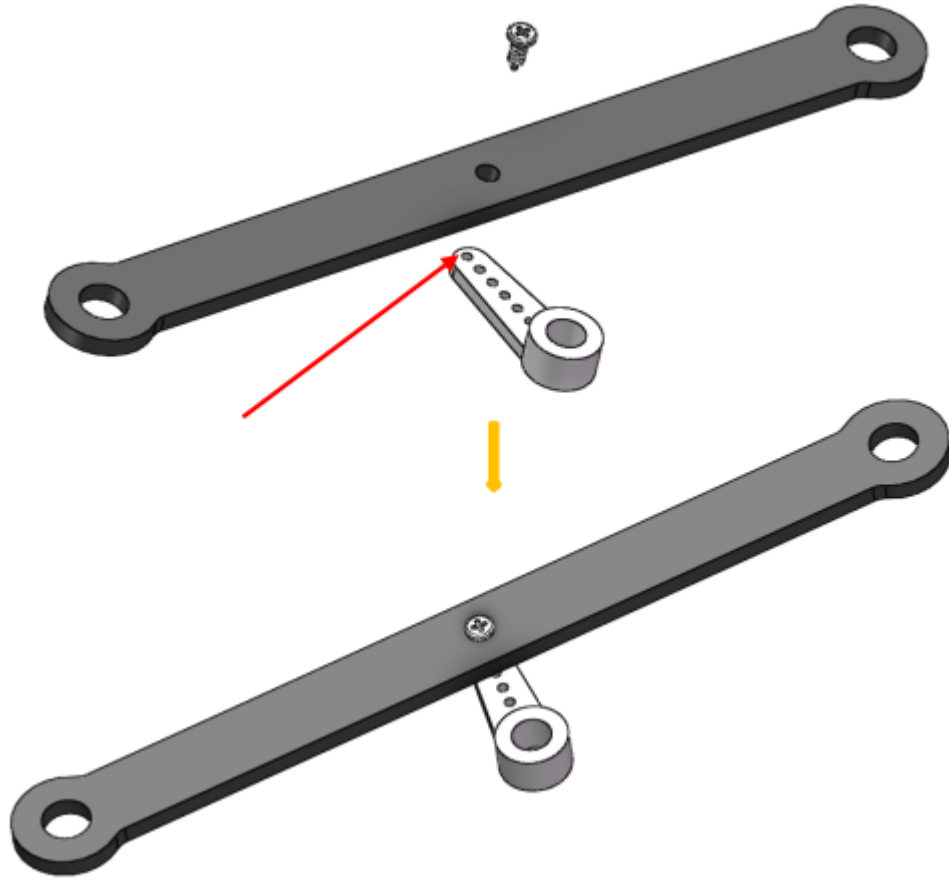
4つの M2x8 ネジ と M2 ナット を使用して、2つのサーボを **パン&チルトプレート** に取り付ける（2つのサーボシャフトは左側にある）。



1.3.3 ステアリングパーツ

ステアリングリンクageと1アームロッカーアームをM1.5x4セルフタッピングネジで接続します。

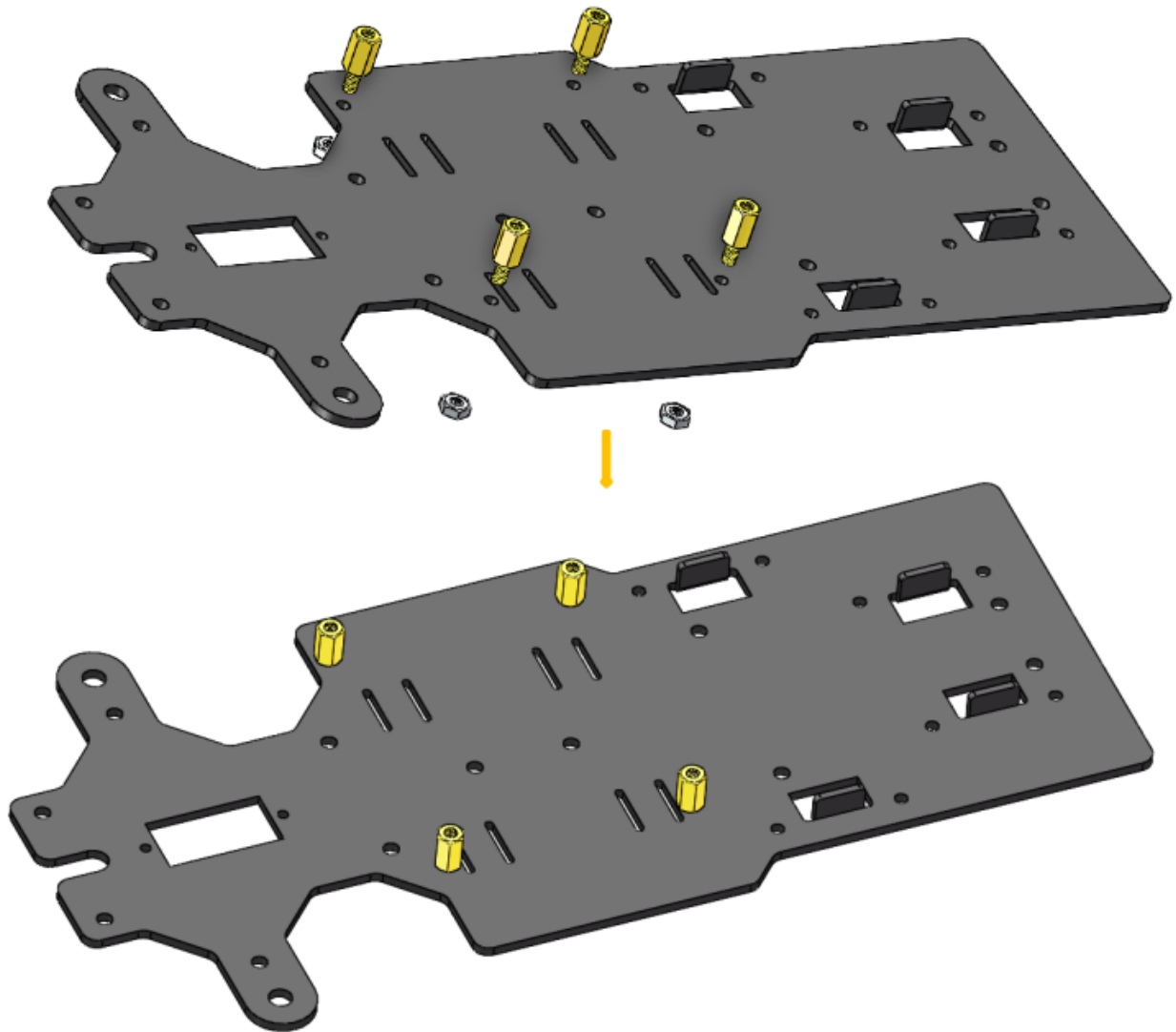
注釈: ギアから最も遠いアームの **最初** の穴（下の **矢印** で示されている）に挿入します。



また、それらをしっかりと締めてから、ネジを少し緩め、ステアリングリンクageが柔軟に動くようにする。

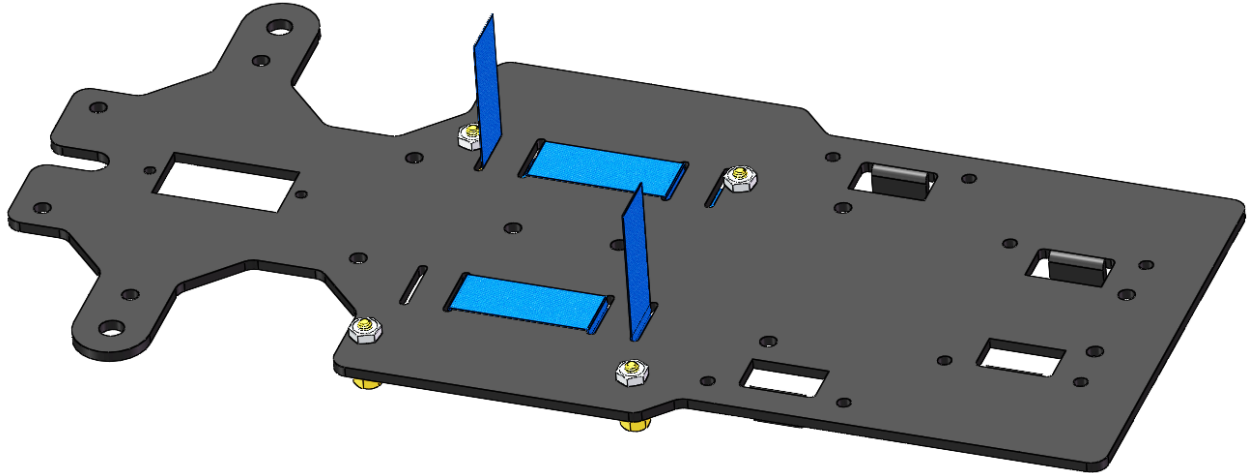
1.3.4 上部プレート

最初に **M2.5x8 銅製スタンドオフ** と **M2.5 ナット** を **上部プレート** に取り付ける。突き出た支柱が上に向くように注意してください。

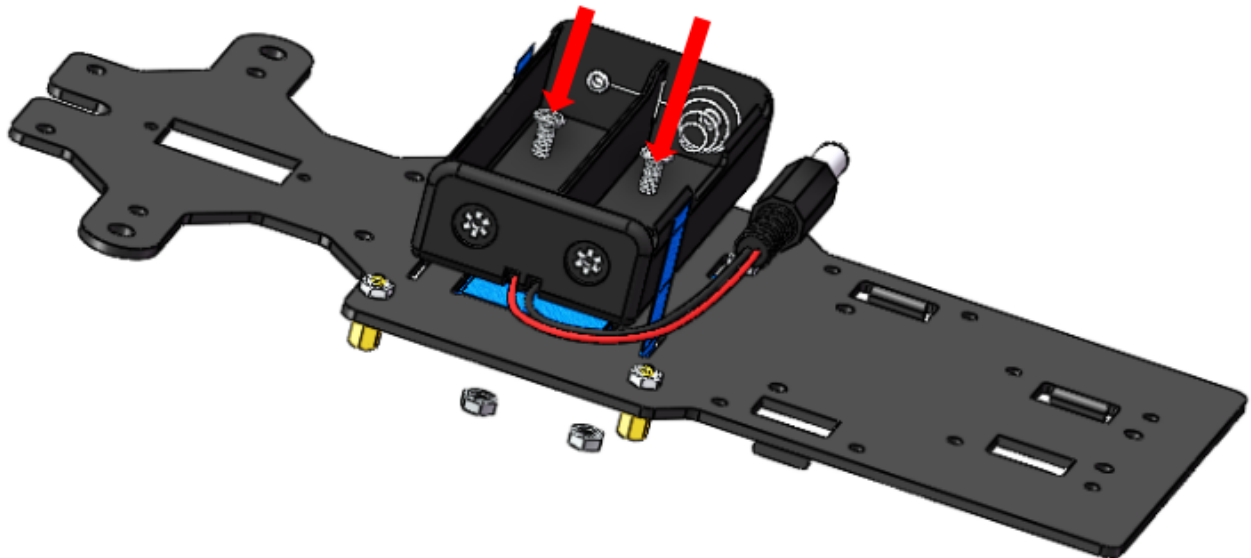


1.3.5 バッテリーホルダー

上部プレートを裏返しにする。**リボン**を半分にカットする。プレートの穴に通す。方向に注意して、バッテリーを後で簡単に取り外すことができるように、プレート的一方の端を長くしてください。

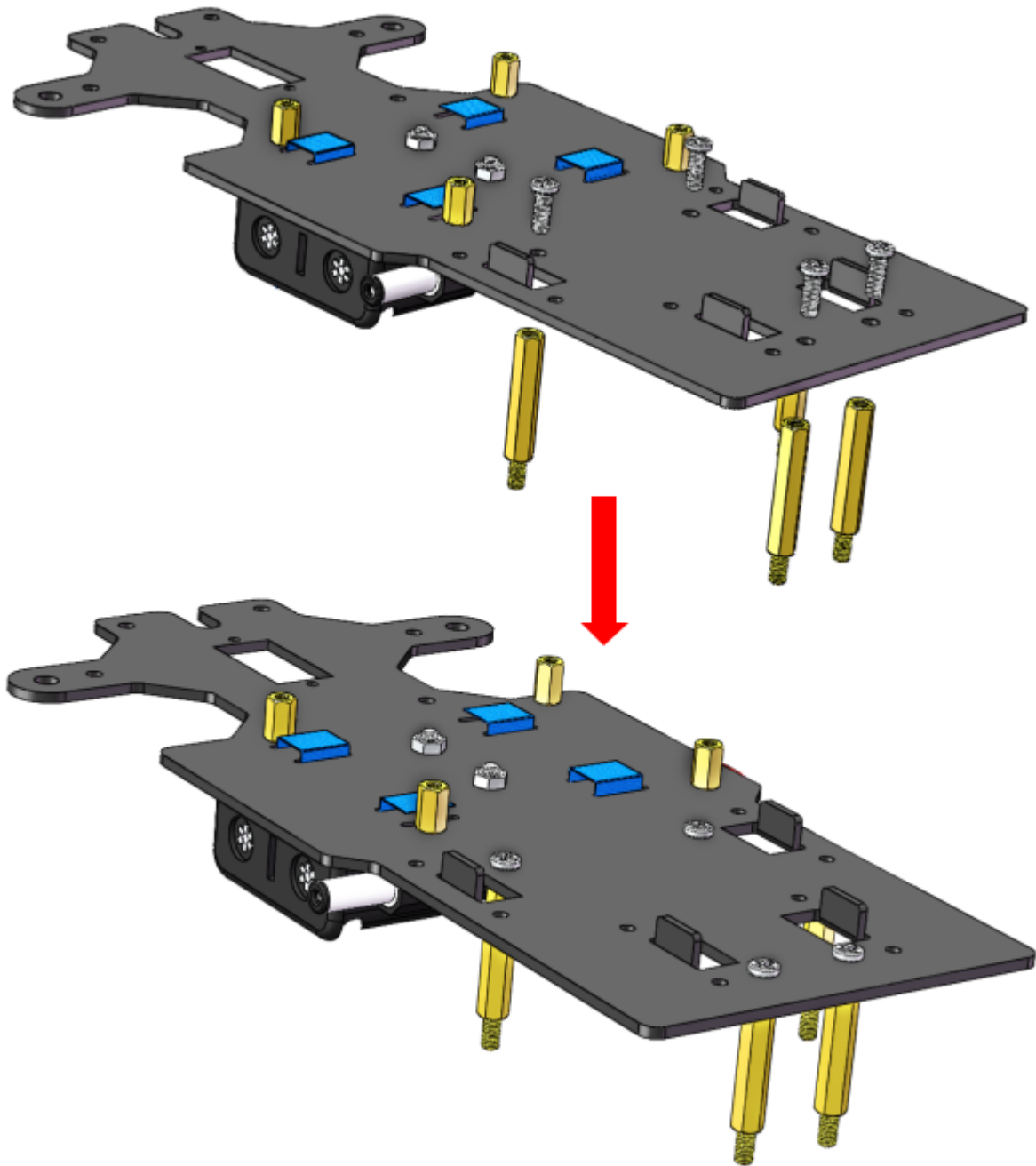


2本の **M3x8 皿ネジ** と **M3 ナット** でバッテリーホルダーを固定する。バッテリーホルダーの配線の方向に注意を払ってください。



1.3.6 後輪（ネジ）

4本の **M3x25 銅製スタンドオフ** を備えた4本の **M3x8 ネジ** を差し込む。

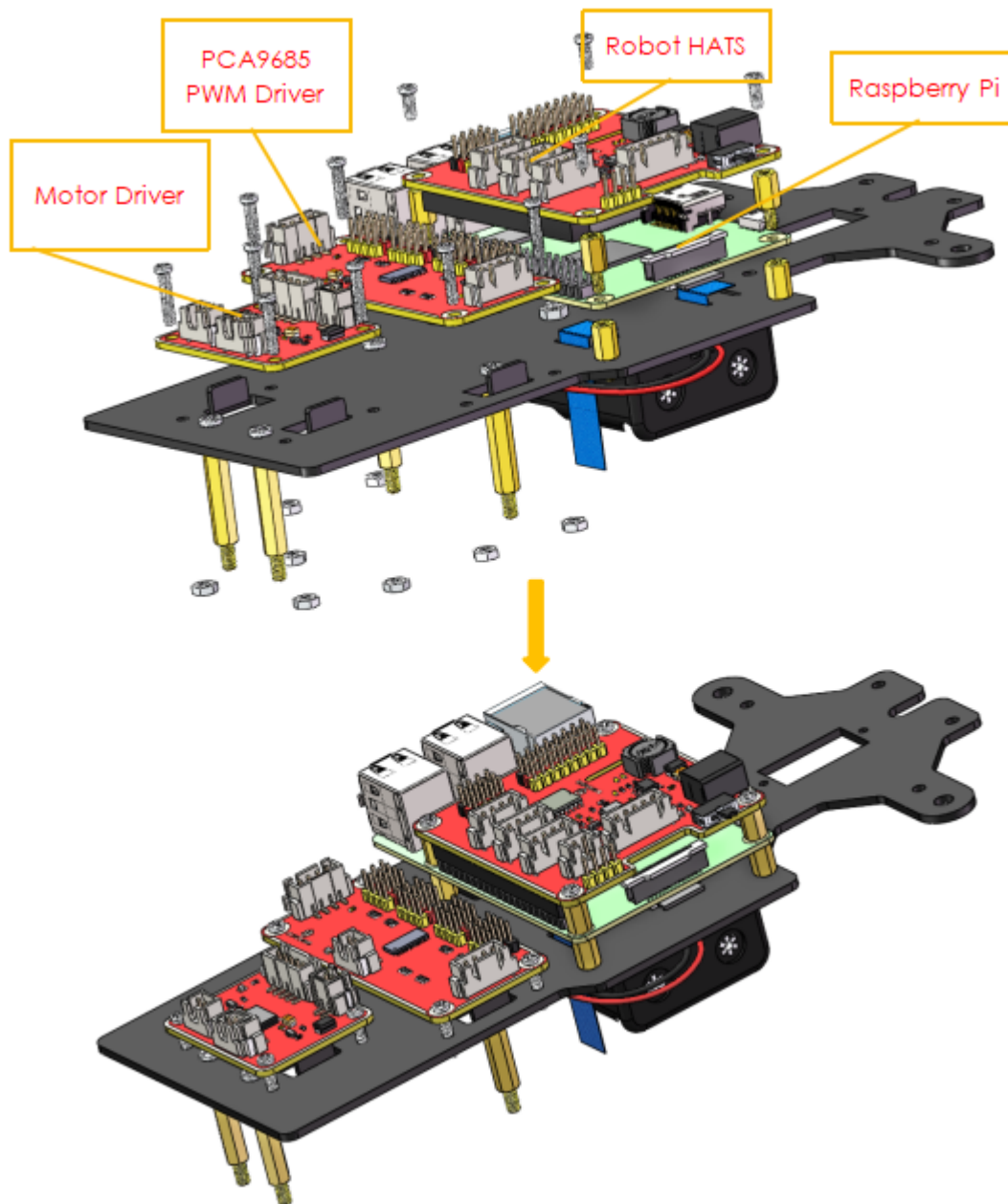


1.3.7 PCB 組み立て

- 1) Raspberry Pi (TF カード差込済み) を 8 つの M2.5x8 シングルパス銅製スタンドオフ で組み立てから ロボット HATS を差し込む。
- 2) 4 つの M2.5x6 ネジ で ロボット HATS を固定する。

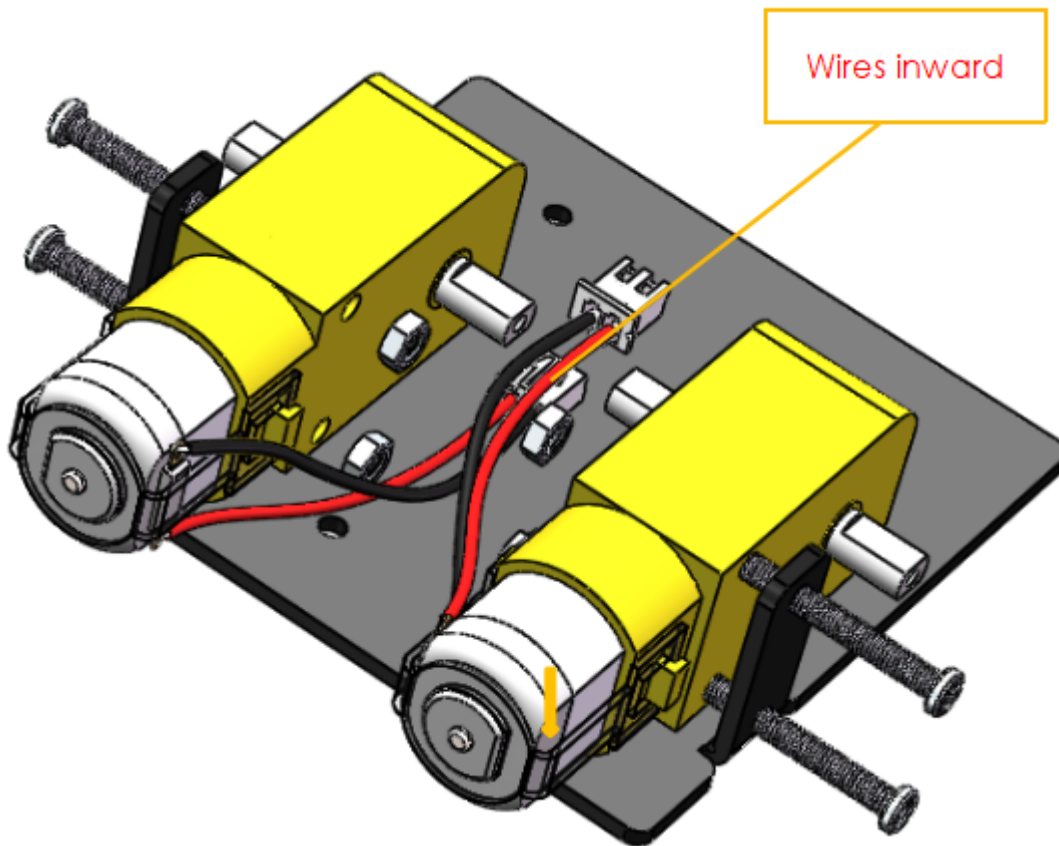
PCA9685 PWM ドライバー と モータードライバー を 8 つの M2.5x12 ネジと M2.5 ナットで 下部プレート に固

定する：

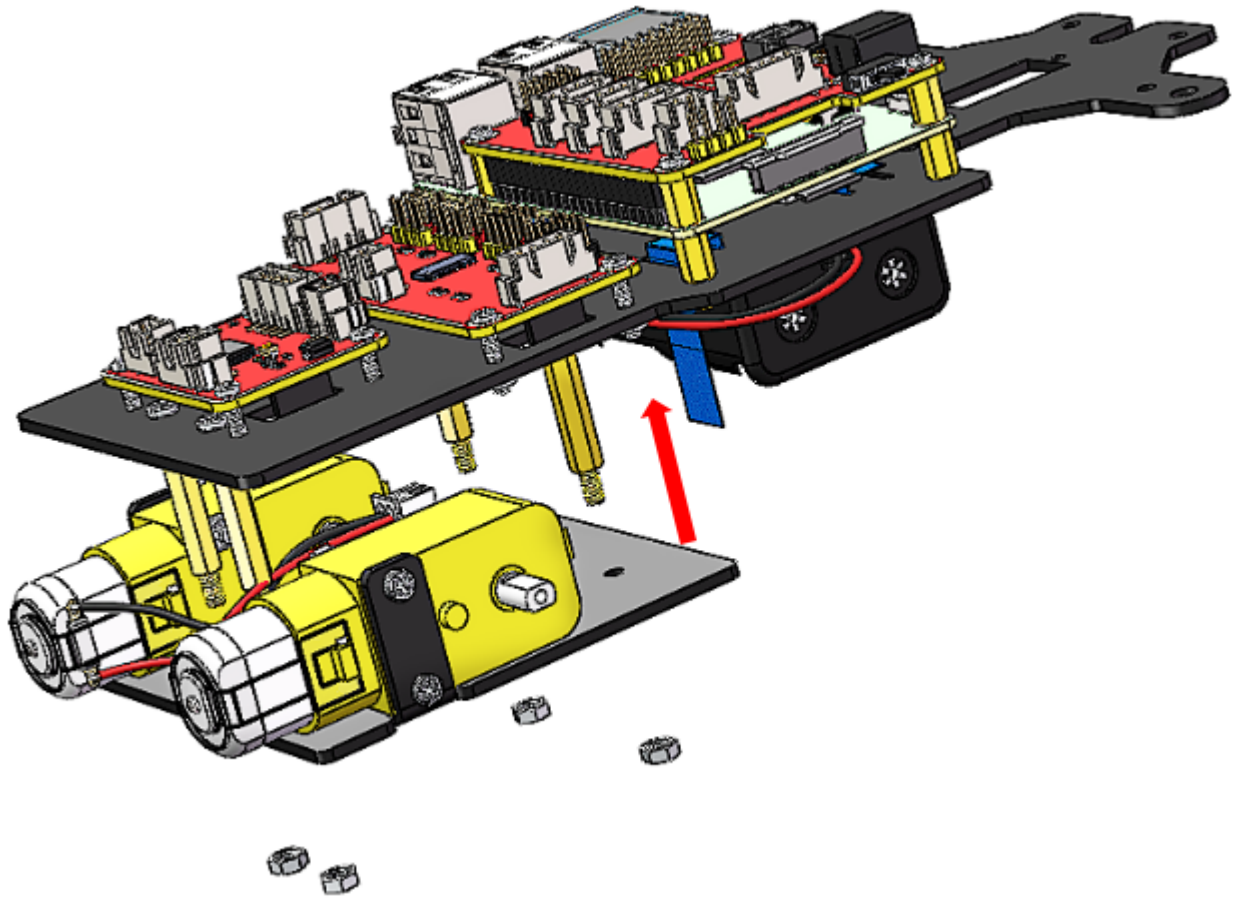


1.3.8 後輪の固定

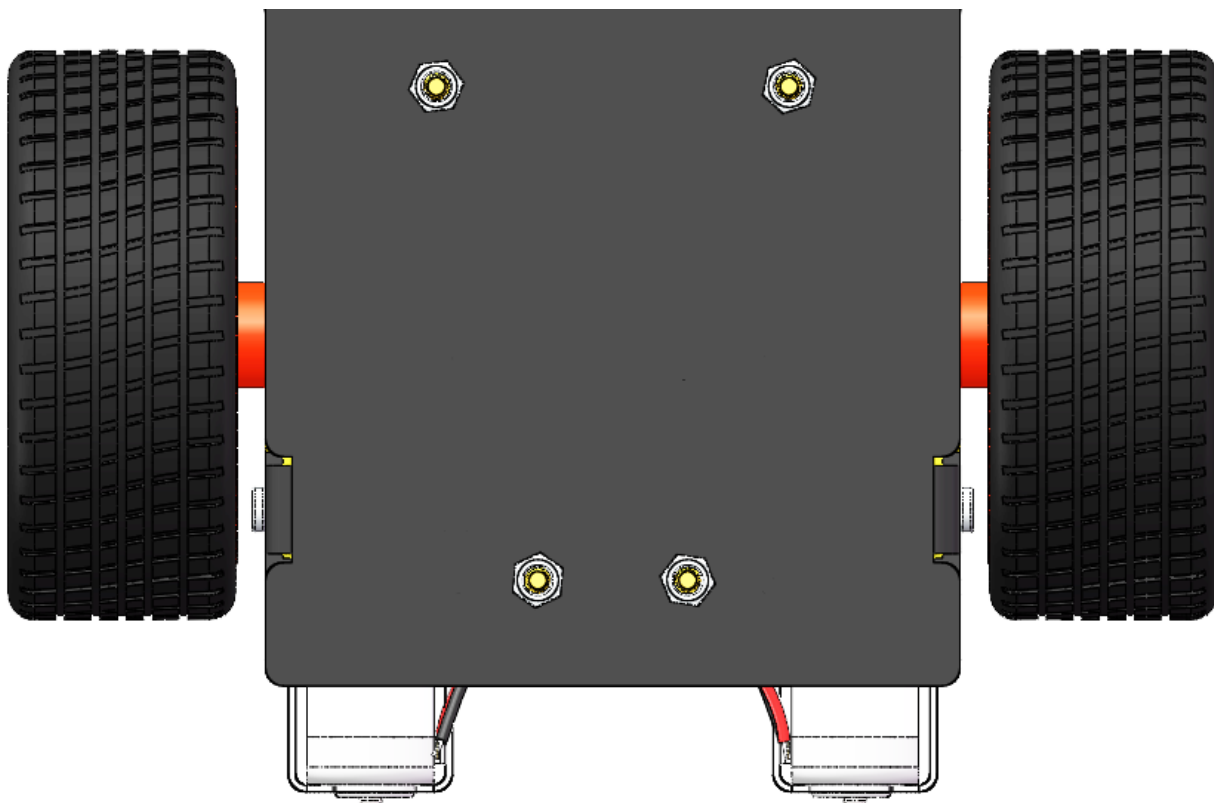
4つの M3x25 ネジと M3 ナットで2つのモーターを組み立てる。モーターを配線で内側に配置するように注意してください。こうすると、回路を接続することを便利にする。



4つの M3 ナットで後輪を組み立てる。

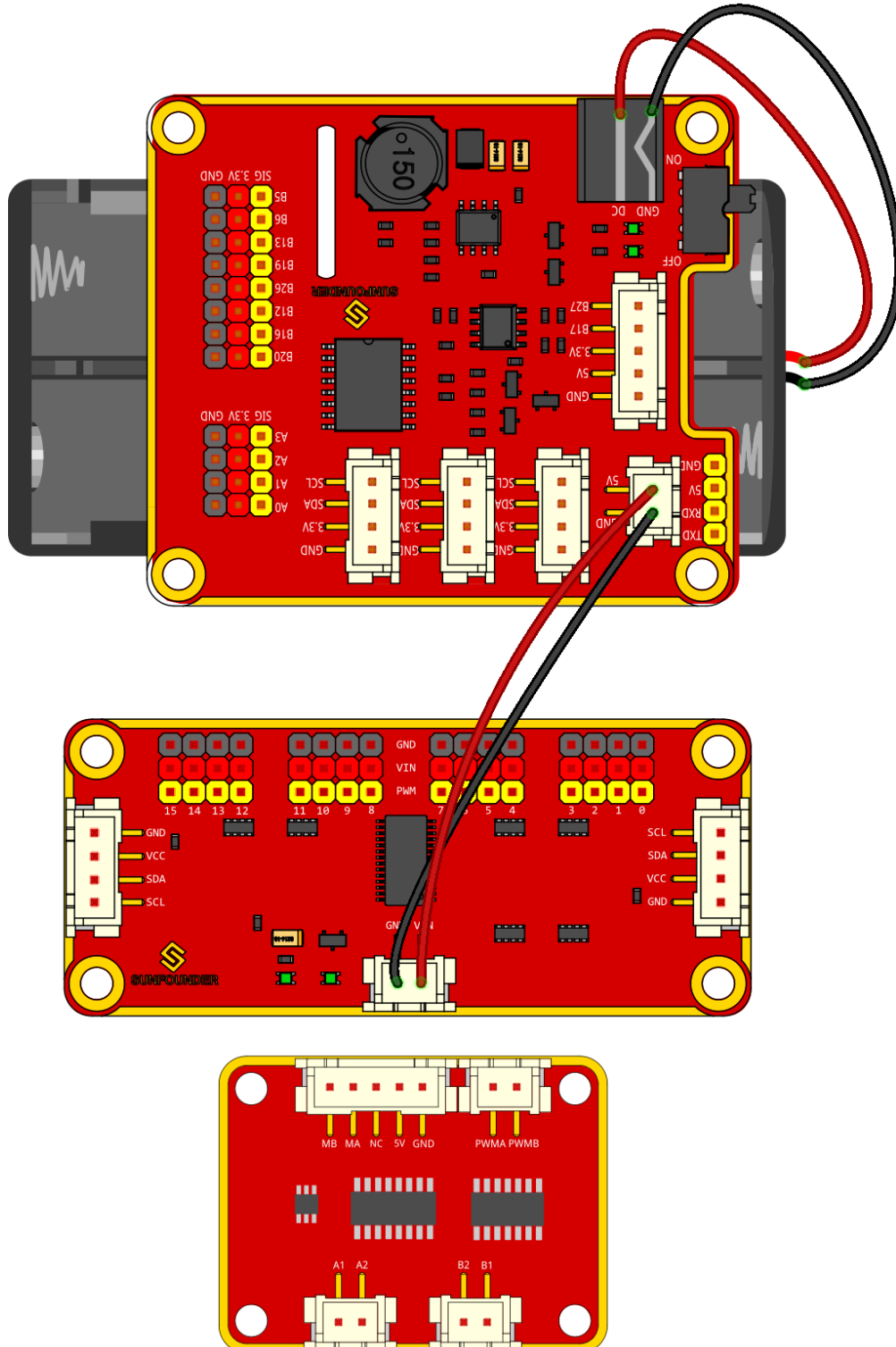


後輪 をモーターシャフトに合わせ、回転させてやさしく差し込む。

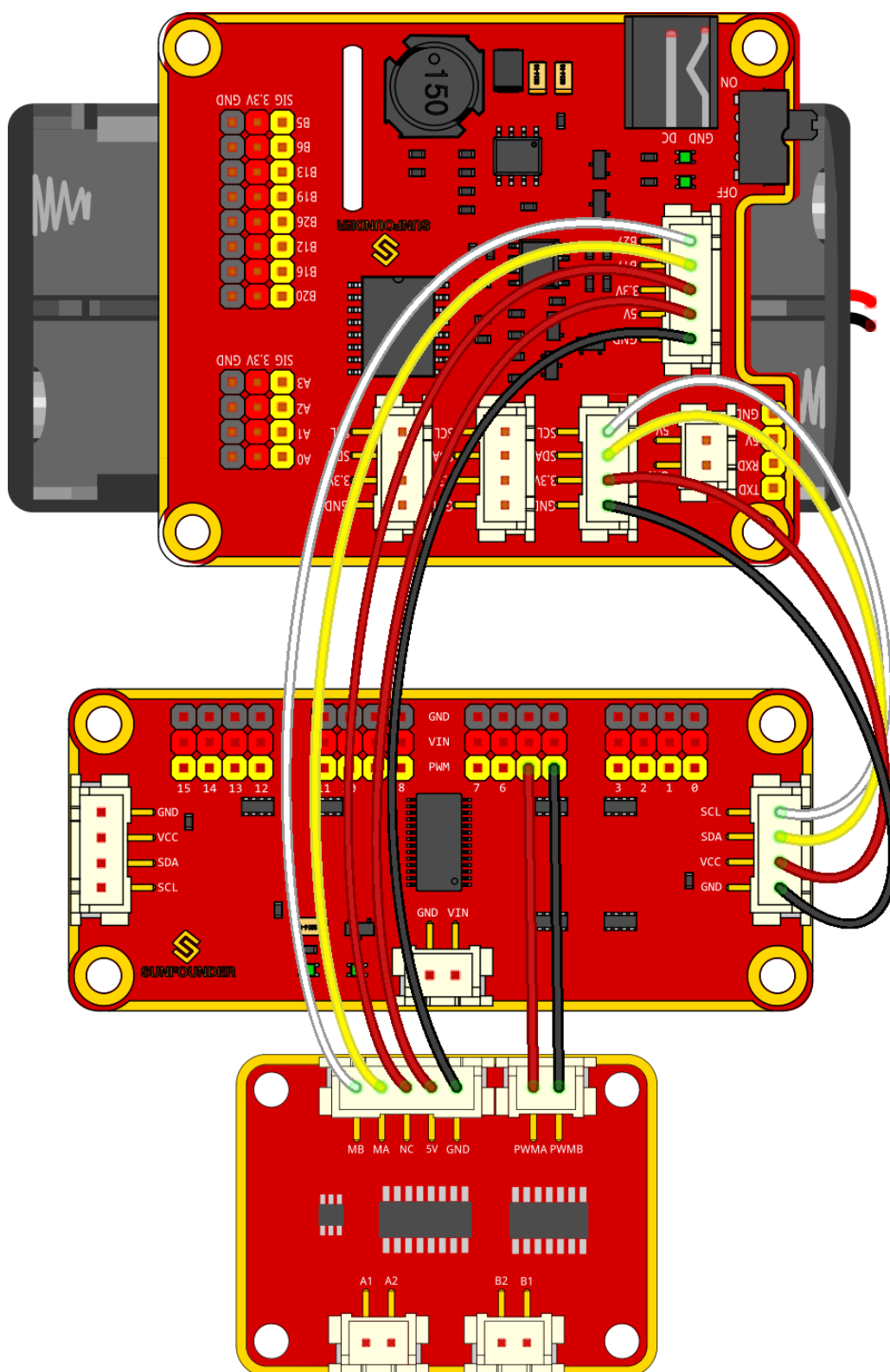


1.4 回路構築

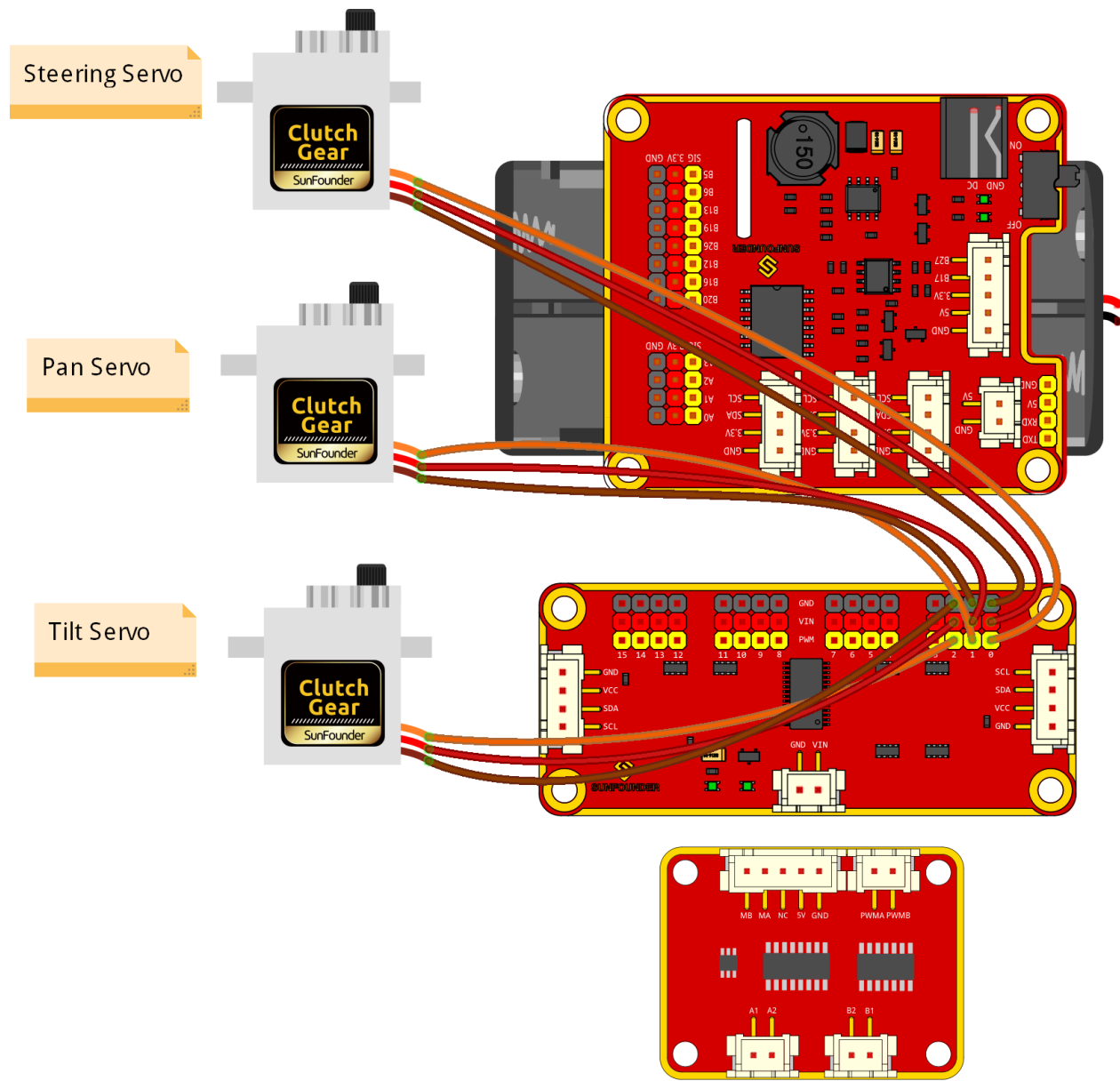
1.4.1 電源を接続する



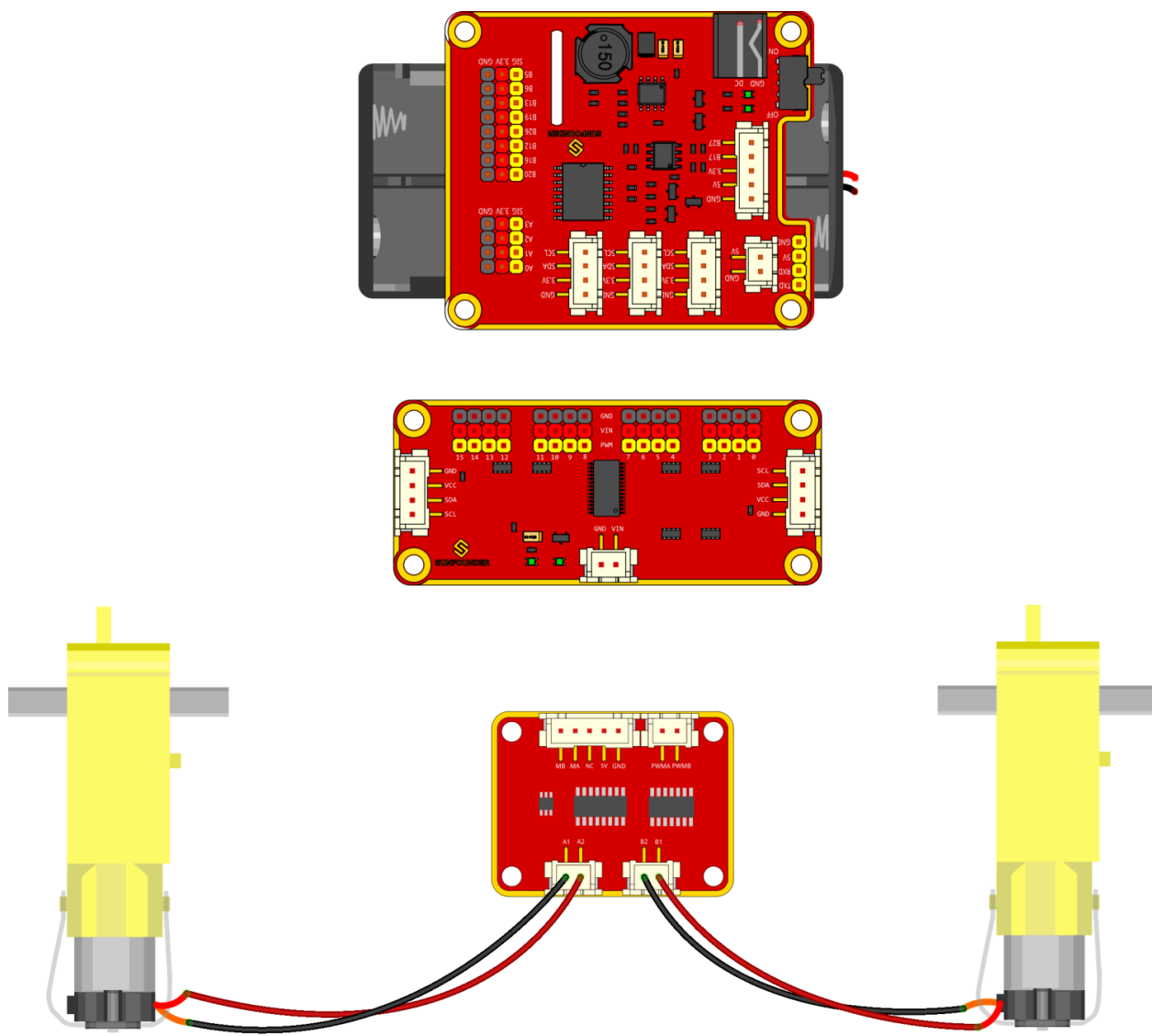
1.4.2 モジュールを接続する



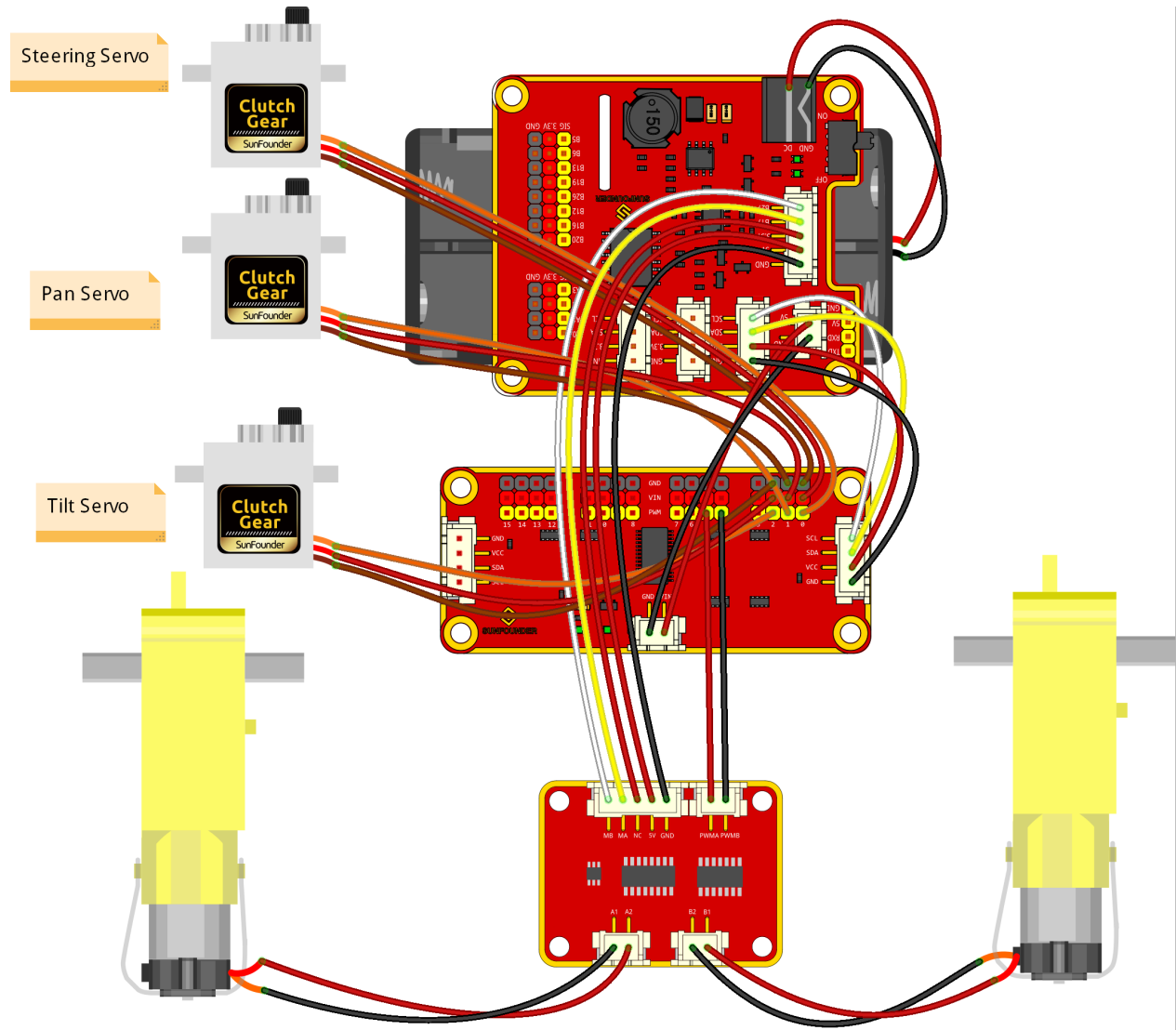
1.4.3 サーボを接続する



1.4.4 モーターを接続する



完全な接続は次のように表示される。



1.5 Raspberry Pi を始めよう

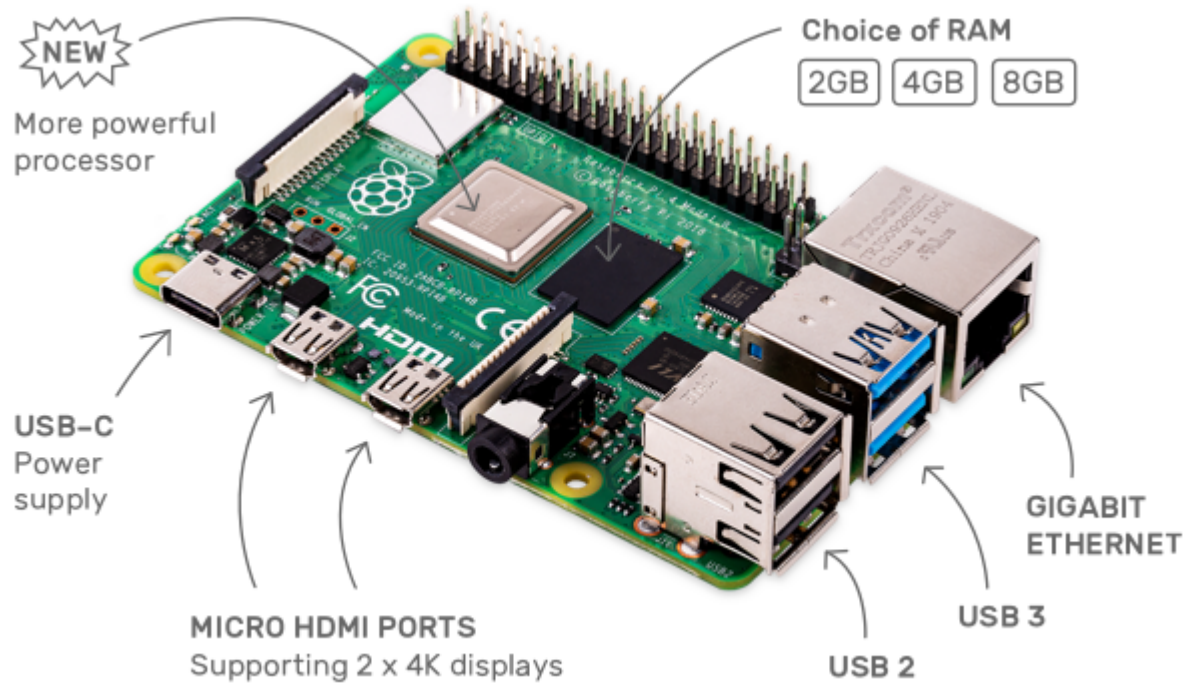
このセクションでは、Raspberry Pi OS のインストール方法、Raspberry Pi への wifi の設定、Raspberry Pi へのリモートアクセスして対応するコードを実行する方法を教えます。

1.5.1 何が必要ですか？

必要なコンポーネント

Raspberry Pi

Raspberry Pi は、低コストでクレジットカードサイズのコンピュータで、コンピュータモニターや TV に接続し、標準のキーボードとマウスを使用します。これは、すべての年齢の人々がコンピューティングを探求し、Scratch や Python のような言語でプログラミングを学ぶことを可能にする優れた小さなデバイスです。



電源アダプター

電源ソケットに接続するために、Raspberry Pi にはマイクロ USB ポート（多くの携帯電話に見られるものと同じ）があります。少なくとも 2.5 アンペアを提供する電源が必要です。

マイクロ SD カード

Raspberry Pi は、すべてのファイルと Raspberry Pi OS を保存するためにマイクロ SD カードが必要です。少なくとも 8GB の容量を持つマイクロ SD カードが必要になります

オプションのコンポーネント

スクリーン

Raspberry Pi のデスクトップ環境を表示するには、TV スクリーンまたはコンピューターモニターとして使用できるスクリーンが必要です。スクリーンに内蔵スピーカーがある場合、Pi はそれらを介して音を再生します。

マウス & キーボード

スクリーンを使用する場合は、USB キーボードと USB マウスも必要です。

HDMI

Raspberry Pi には、ほとんどの現代の TV とコンピューターモニターの HDMI ポートと互換性のある HDMI 出力ポートがあります。スクリーンに DVI ポートまたは VGA ポートのみがある場合は、適切な変換ラインを使用する必要があります。

ケース

Raspberry Pi をケースに入れることができます。これにより、デバイスを保護できます。

サウンドまたはイヤホン

Raspberry Pi には約 3.5mm のオーディオポートが装備されており、スクリーンに内蔵スピーカーがない場合やスクリーン操作がない場合に使用できます。

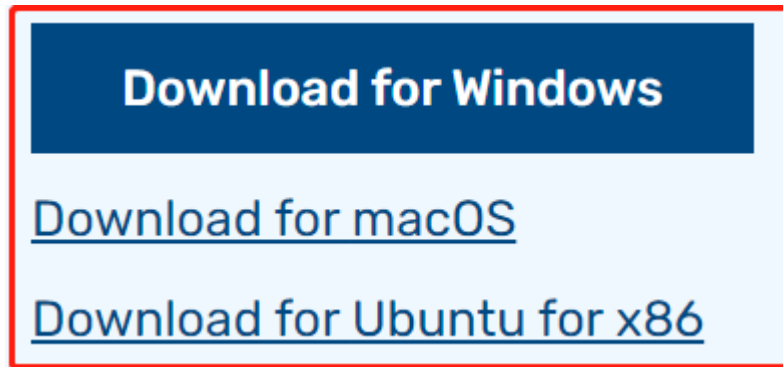
1.5.2 OS のインストール

必要なコンポーネント

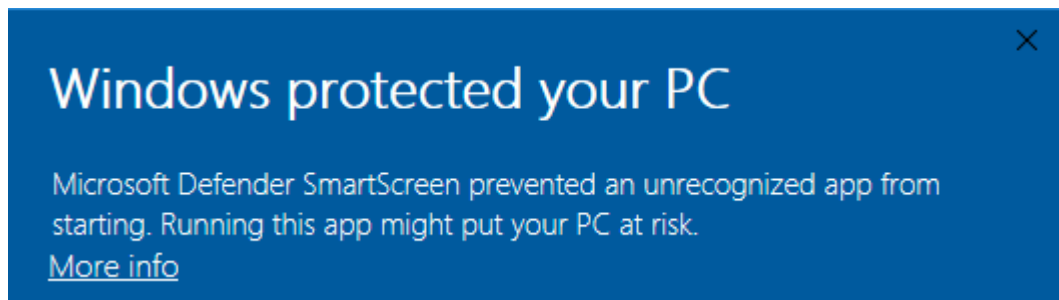
- Raspberry Pi 4B/Zero 2 w/3B 3B+/2B/Zero W
- 1 x パーソナルコンピュータ
- 1 x マイクロ SD カード

手順

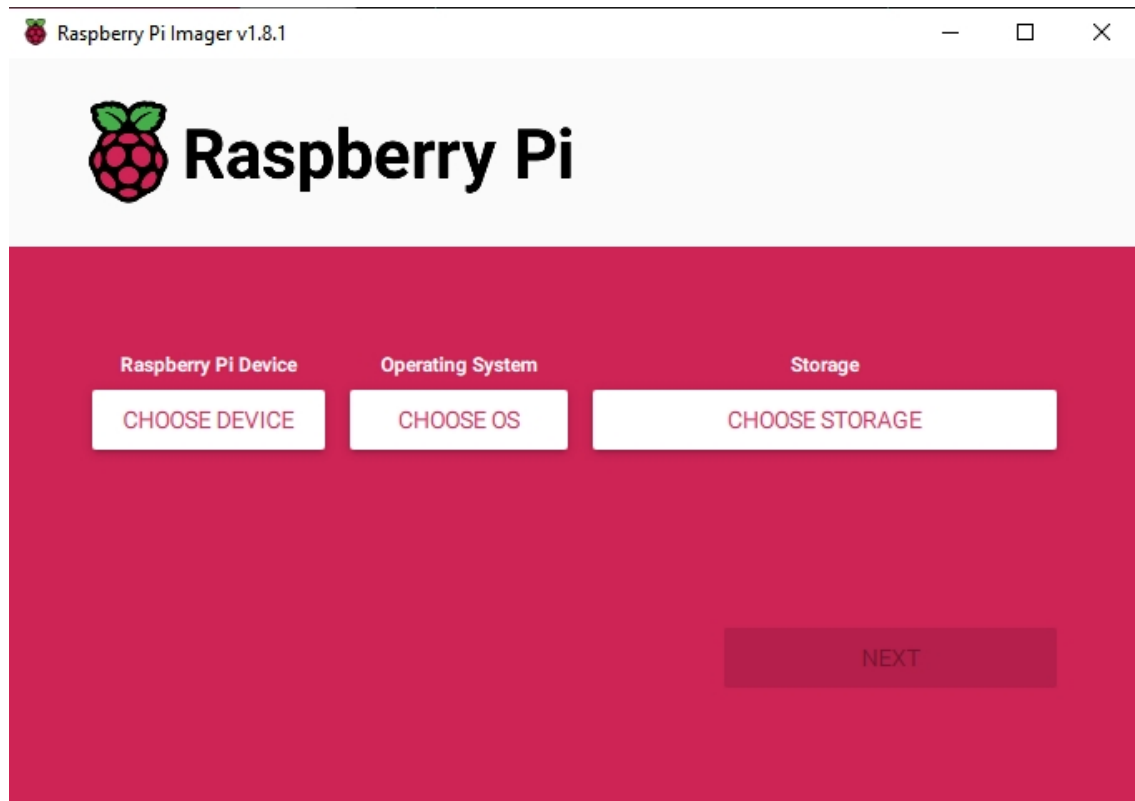
1. Raspberry Pi のソフトウェアダウンロードページにアクセスしてください： [Raspberry Pi Imager](#)。お使いのオペレーティングシステム用の Imager バージョンを選択します。ダウンロード後、ファイルを開いてインストールを開始してください。



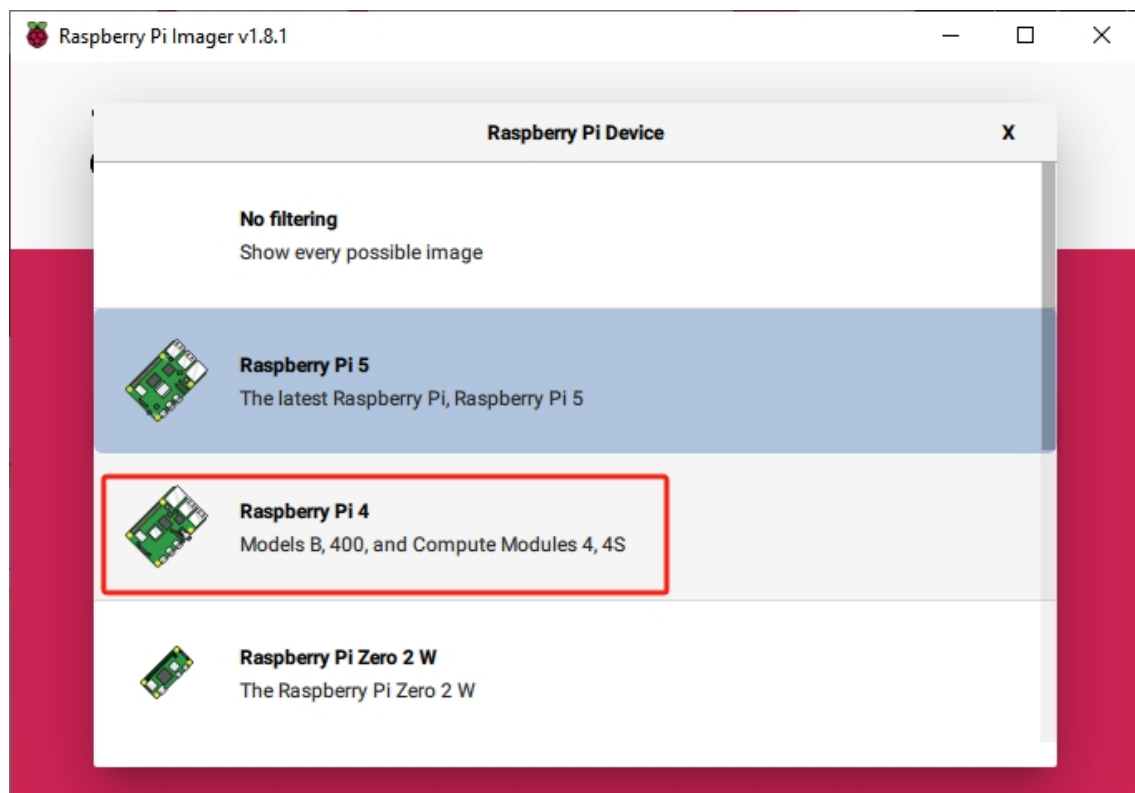
2. インストーラーを起動すると、OS によってはセキュリティ警告が表示される場合があります。例えば、Windows では注意メッセージが表示されることがあります。このような場合は、**詳細情報** を選択してから **とにかく実行** を選択します。画面の指示に従って、Raspberry Pi Imager をインストールしてください。



3. SD カードをコンピューターまたはラップトップの SD カードスロットに挿入してください。
4. アイコンをクリックするか、ターミナルで `rpi-imager` を実行することで、Raspberry Pi Imager アプリケーションを開いてください。



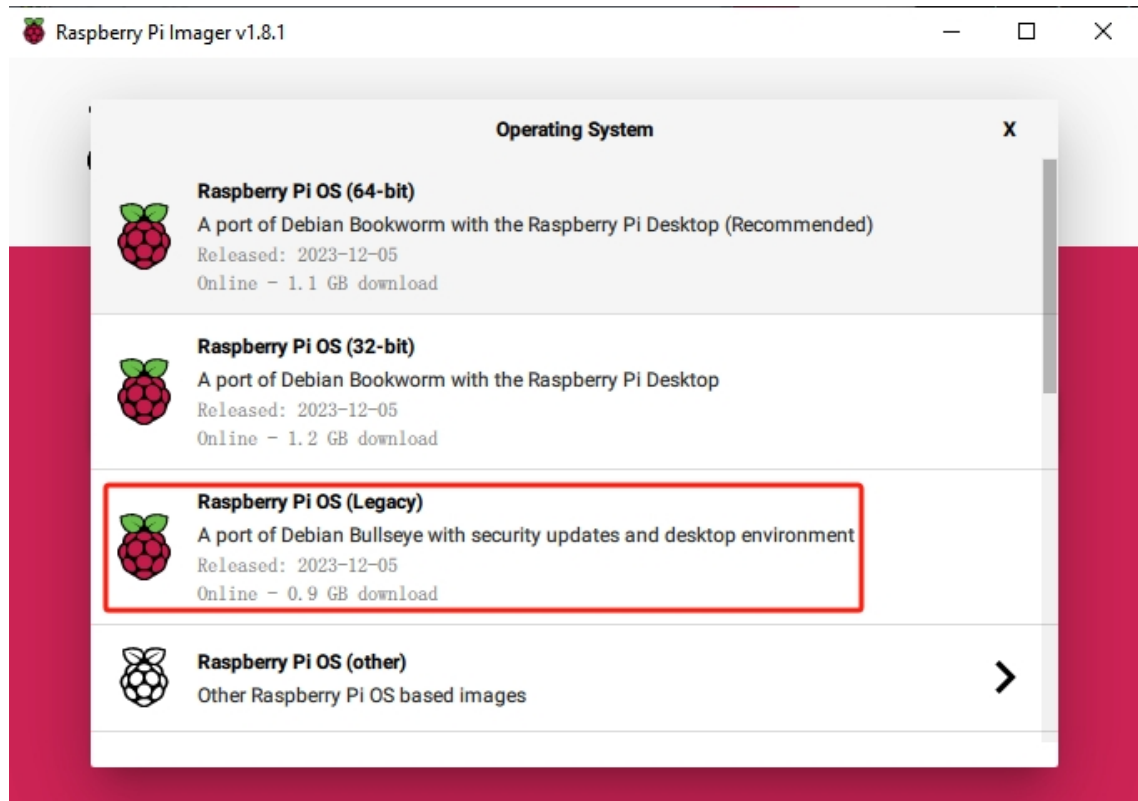
5. **デバイスを選択** をクリックし、リストから特定の Raspberry Pi モデルを選択してください（注：Raspberry Pi 5 は該当しません）。



6. **OS を選択** を選び、その後で **Raspberry Pi OS (Legacy)** を選択してください。

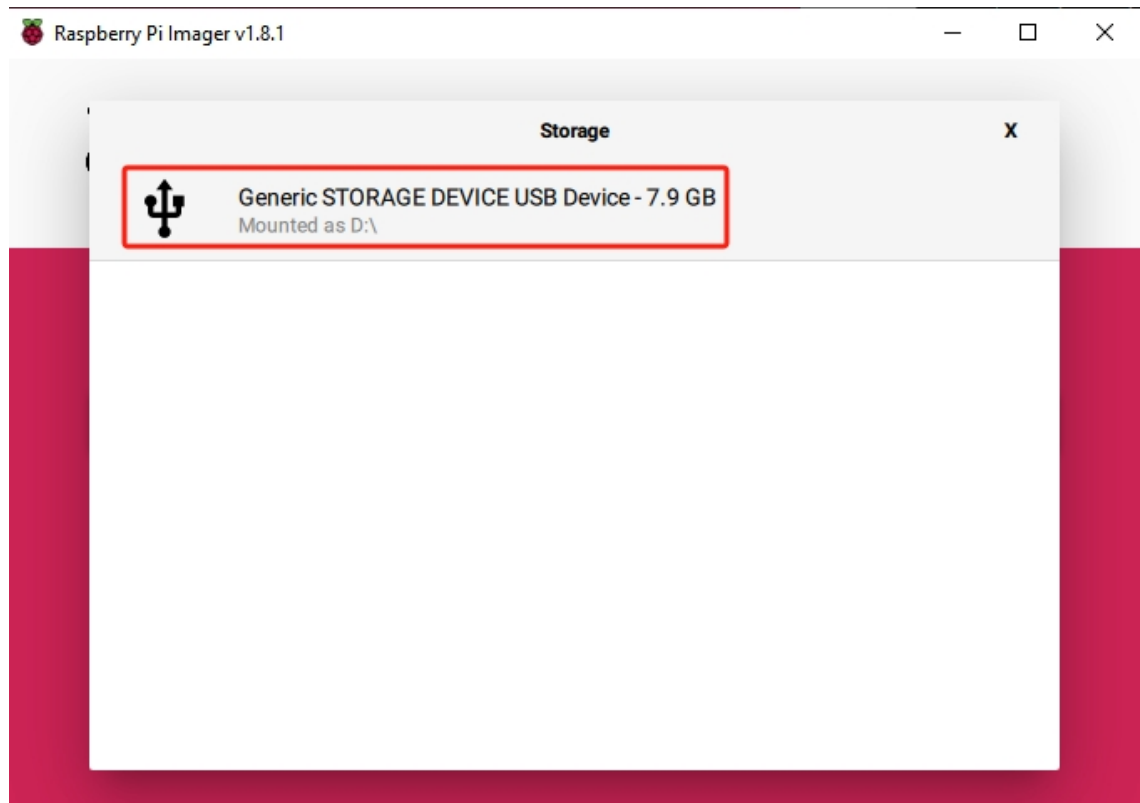
警告:

- **Bookworm** バージョンはスピーカーが動作しないため、インストールしないでください。
- **Debian Bullseye の Raspberry Pi OS (Legacy)** バージョンをインストールする必要があります。

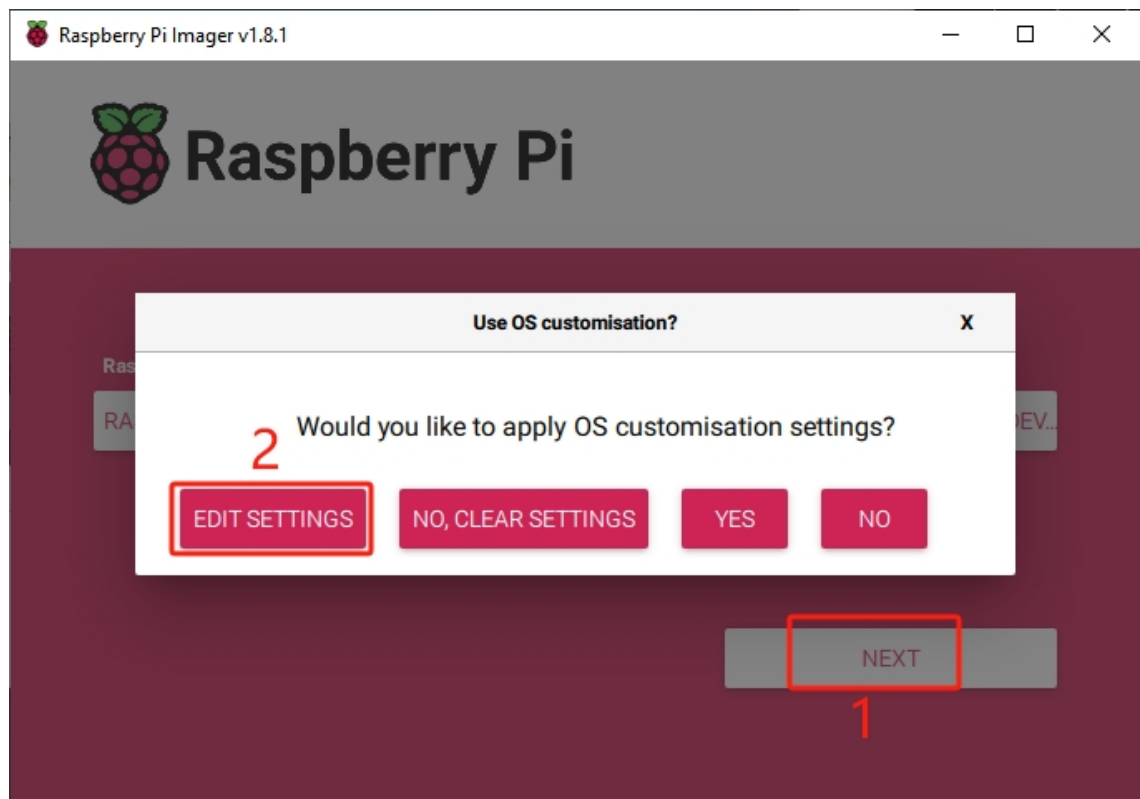


7. **ストレージを選択** をクリックし、インストール用の正しいストレージデバイスを選んでください。

注釈: 複数のストレージデバイスが接続されている場合は特に、正しいデバイスを選択してください。
不明な場合は、他のデバイスを切断してください。

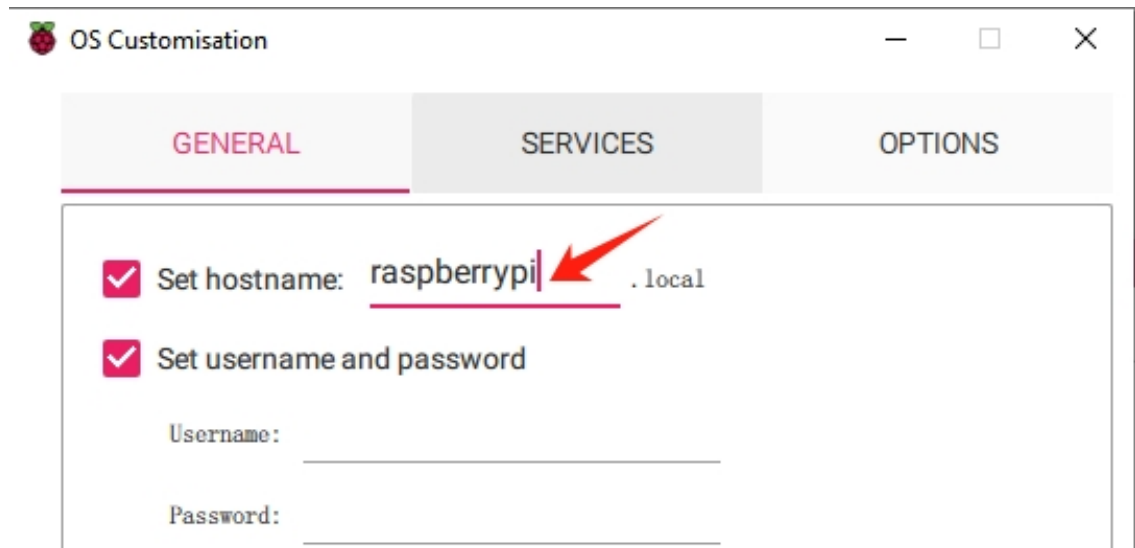


8. 次へ を押して、**設定を編集** を選択し、OS の設定をカスタマイズしてください。



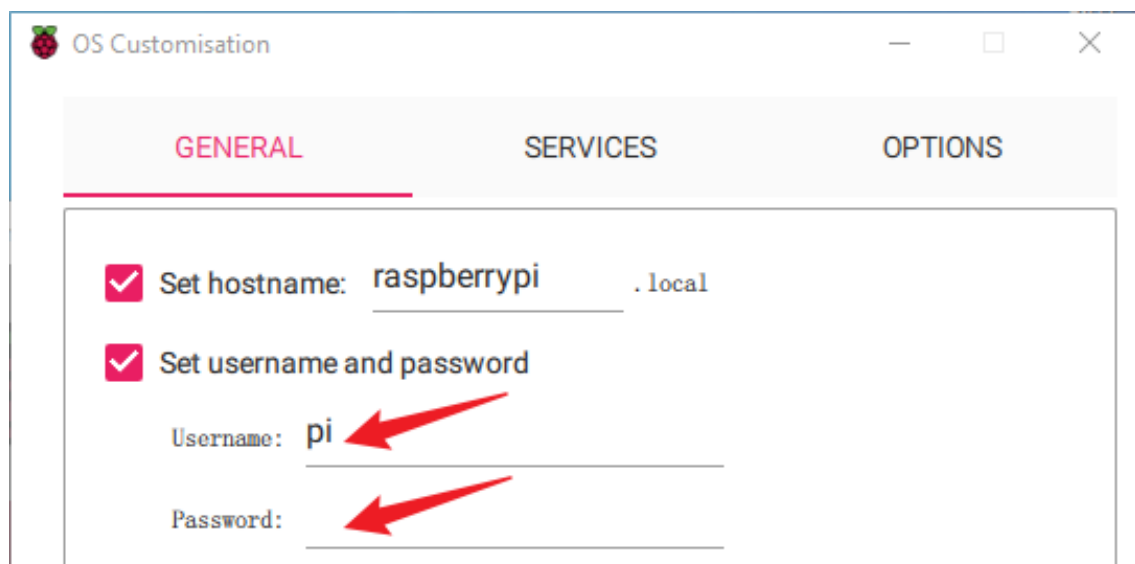
9. Raspberry Pi の **ホスト名** を設定してください。

注釈: ホスト名は、ネットワーク上で Raspberry Pi が自身を識別するために使用する名前です。
<hostname>.local や <hostname>.lan を使用して Pi に接続できます。



10. Raspberry Pi の管理者アカウント用に **ユーザー名** と **パスワード** を作成してください。

注釈: Raspberry Pi にはデフォルトのパスワードがないため、セキュリティのために独自のユーザー名とパスワードを設定することが重要です。



11. ネットワークの **SSID** と **パスワード** を入力して、無線 LAN を設定してください。

注釈: Wireless LAN country should be set the two-letter ISO/IEC alpha2 code for the country in which you are using your Raspberry Pi.

☒ **Configure wireless LAN**

SSID: _____

Password: _____

☐ Show password ☐ Hidden SSID

Wireless LAN country: GB ▼

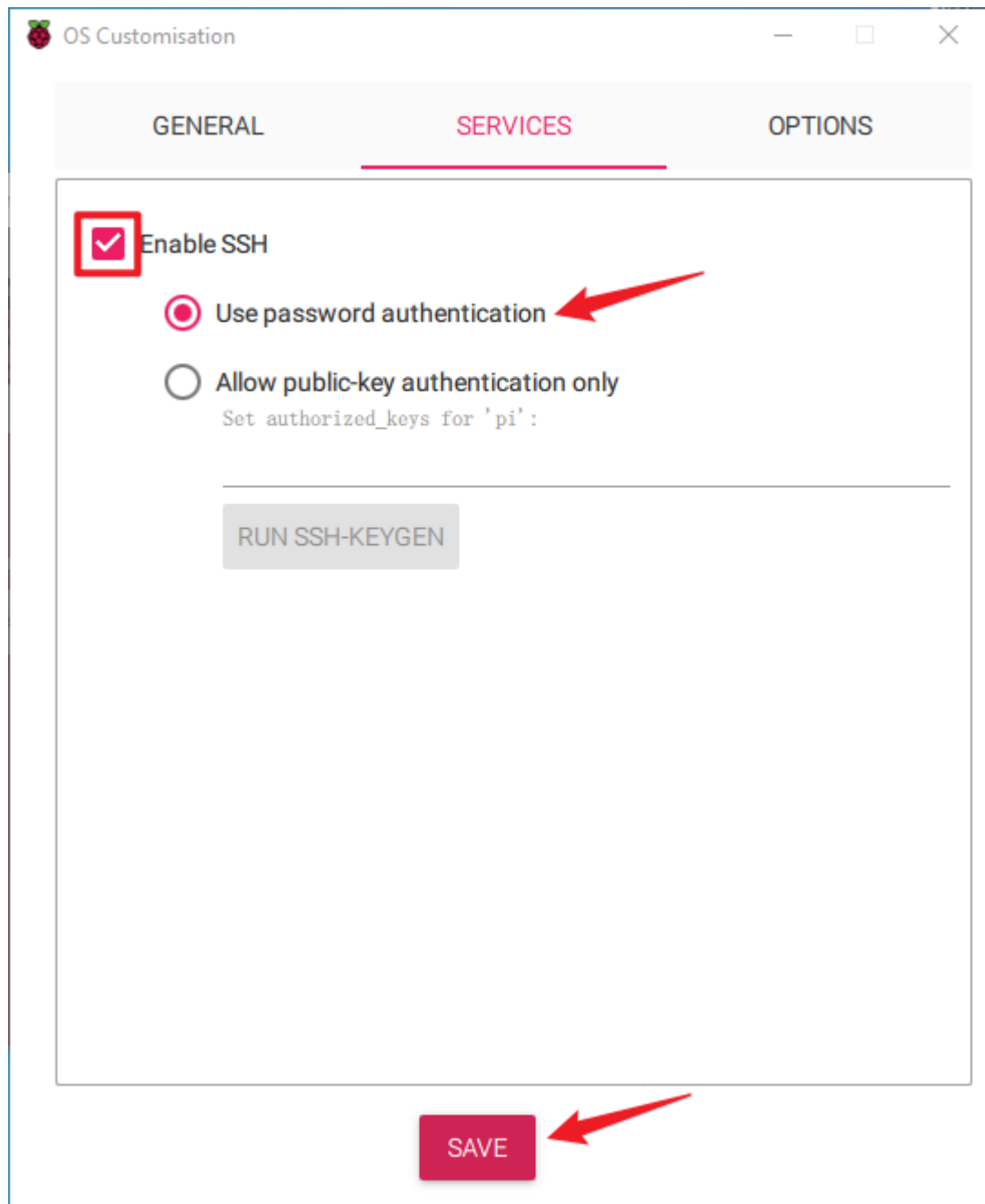
☐ **Set locale settings**

Time zone: Asia/Shanghai ▼

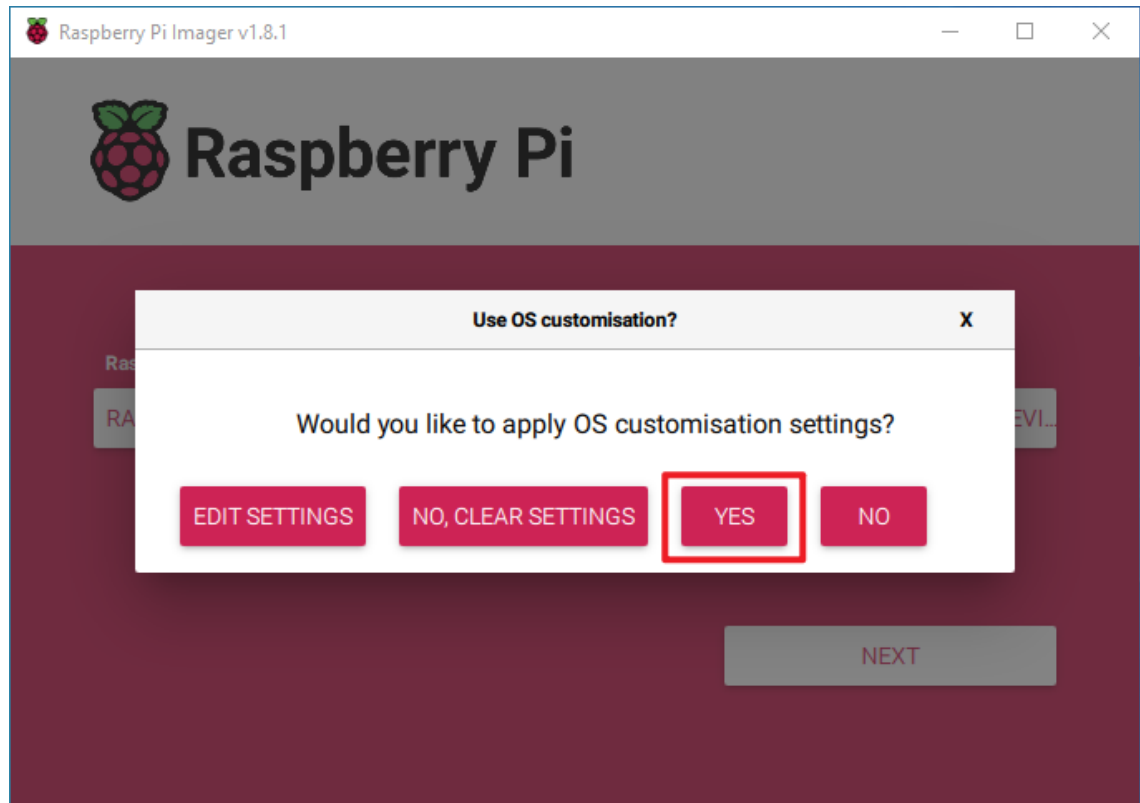
Keyboard layout: US ▼

SAVE

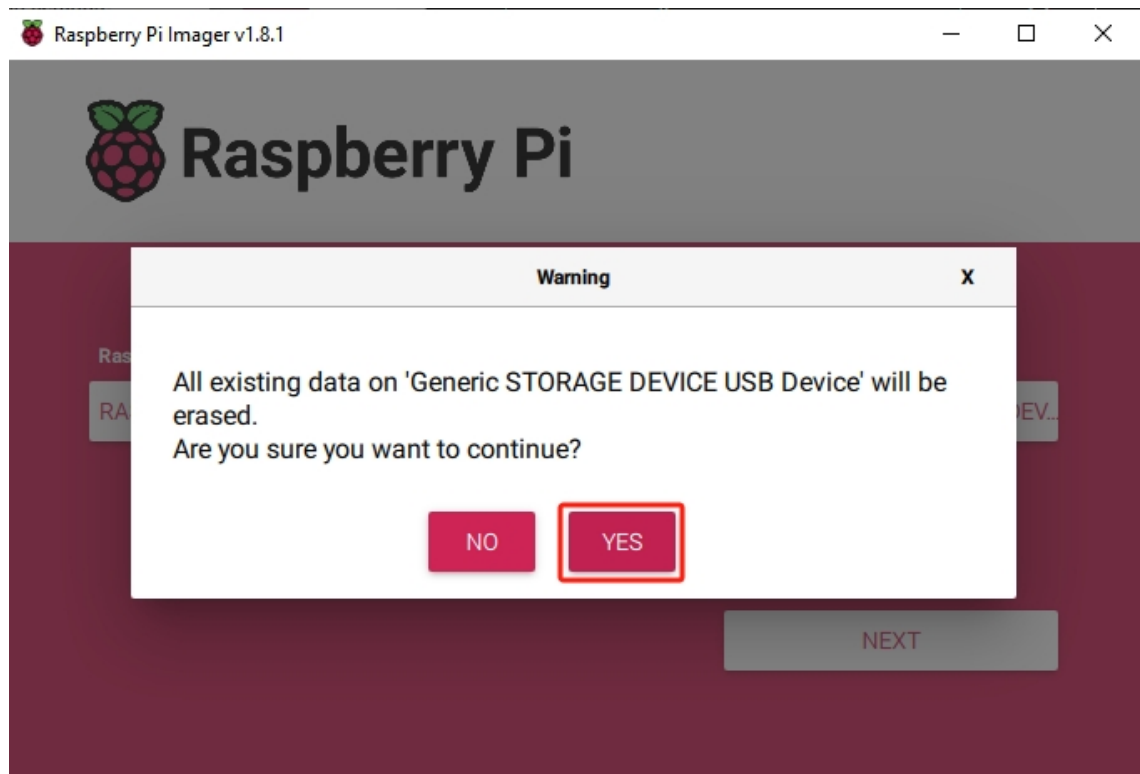
12. Click **SERVICES** and enable **SSH** for password-based remote access. Remember to click **Save**.



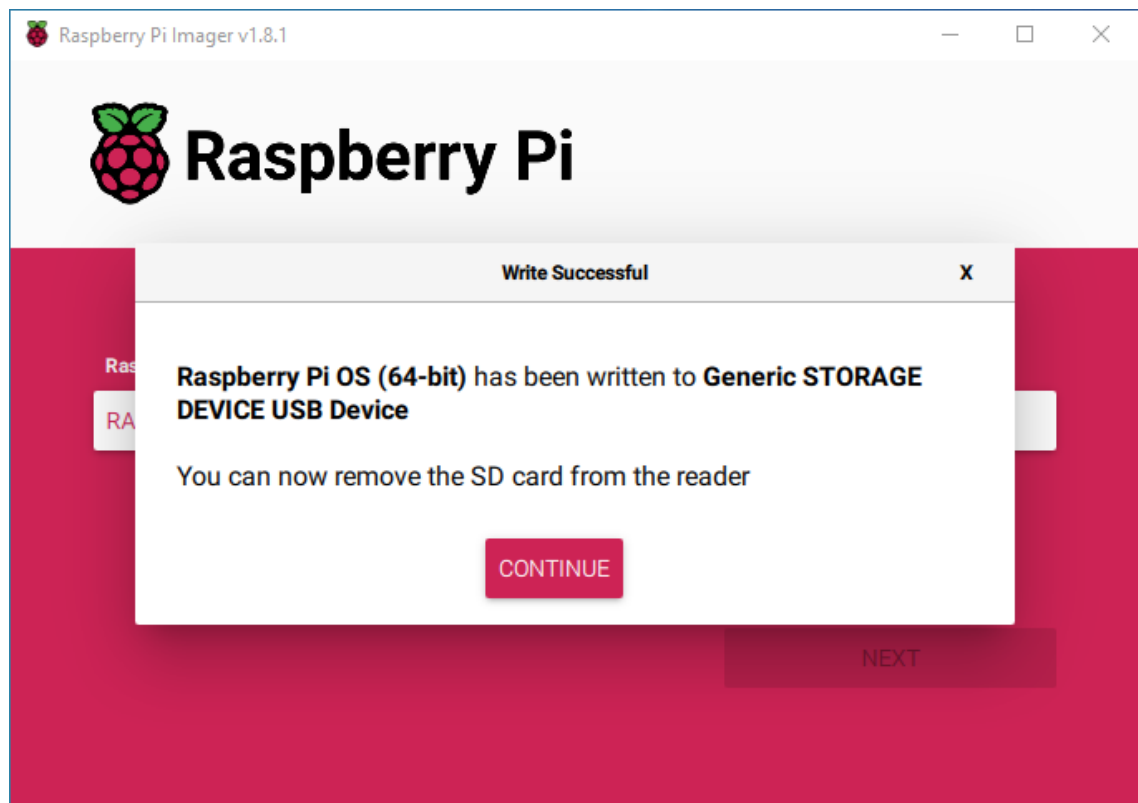
13. 選択を確認するために **はい** をクリックしてください。



14. SD カードに既存のファイルがある場合は、データ損失を避けるためにバックアップを取ってください。バックアップが不要な場合は、**はい** をクリックして進んでください。



15. OS が SD カードに書き込まれるのを待ちます。完了すると、確認ウィンドウが表示されます。

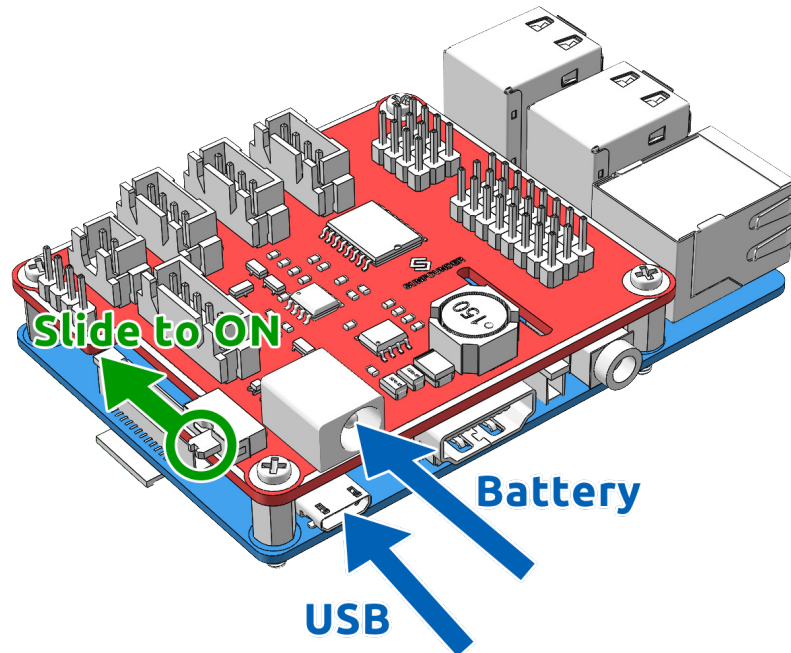


1.5.3 Raspberry Pi のセットアップ

Raspberry Pi の電源供給（重要）

1. Raspberry Pi OS がセットアップされた SD カードを、Raspberry Pi の裏側にある microSD カードスロットに挿入します。

組み立て説明書に従って、バッテリーケーブルを挿入し、電源スイッチをオンにします。次に、USB ケーブルを Raspberry Pi に挿入します。



画面がある場合

注釈: ロボットに搭載された Raspberry Pi ZERO は画面に接続しにくいので、画面を使わない方法で設定してください。

画面がある場合、Raspberry Pi を操作するのが簡単になります。

必要なコンポーネント

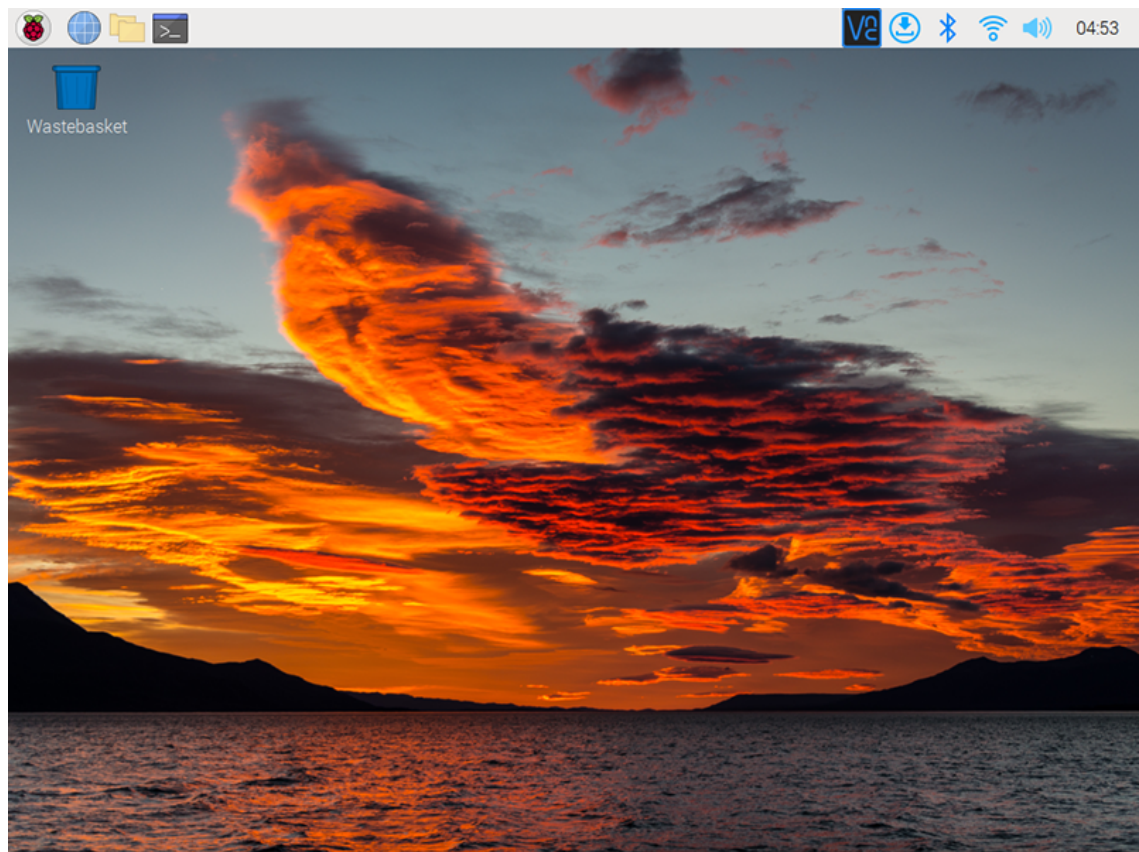
- Raspberry Pi 4B/3B 3B+/2B
- 1 * 電源アダプター
- 1 * マイクロ SD カード
- 1 * 画面用電源アダプター
- 1 * HDMI ケーブル
- 1 * 画面
- 1 * マウス
- 1 * キーボード

1. マウスとキーボードを接続してください。

2. 画面を Raspberry Pi の HDMI ポートに接続し、画面が壁のコンセントに差し込まれ、オンになっていることを確認してください。

注釈: Raspberry Pi 4 を使用する場合は、画面を HDMI0（電源ポートに最も近い）に接続する必要があります。

3. 電源アダプターを使用して Raspberry Pi に電力を供給します。数秒後に Raspberry Pi OS のデスクトップが表示されます。



画面がない場合

モニターがない場合でも、リモートで Raspberry Pi にログインできます。

必要なコンポーネント

- – Raspberry Pi 4B/Zero 2 w/3B 3B+/2B/Zero W
- 1 * 電源アダプター
- 1 * マイクロ SD カード

SSH コマンドを適用して Raspberry Pi の Bash シェルを開くことができます。Bash は Linux の標準デフォルトシェルです。シェル自体は、ユーザーが Unix/Linux を使用する際のコマンド（命令）です。必要な操作のほとんどはシェルを通じて行うことができます。

コマンドウィンドウを使用して Raspberry Pi にアクセスすることに満足していない場合、リモートデスクトップ機能を使用して GUI を介して簡単に Raspberry Pi 上のファイルを管理することもできます。

各システムの詳細なチュートリアルについては以下を参照してください。

Mac OS X ユーザー

Mac ユーザーにとって、コマンドラインからよりも VNC を通じて直接 Raspberry Pi のデスクトップにアクセスする方が便利です。Raspberry Pi 側で VNC を有効にした後、Finder から設定されたアカウントのパスワードを入力してアクセスできます。

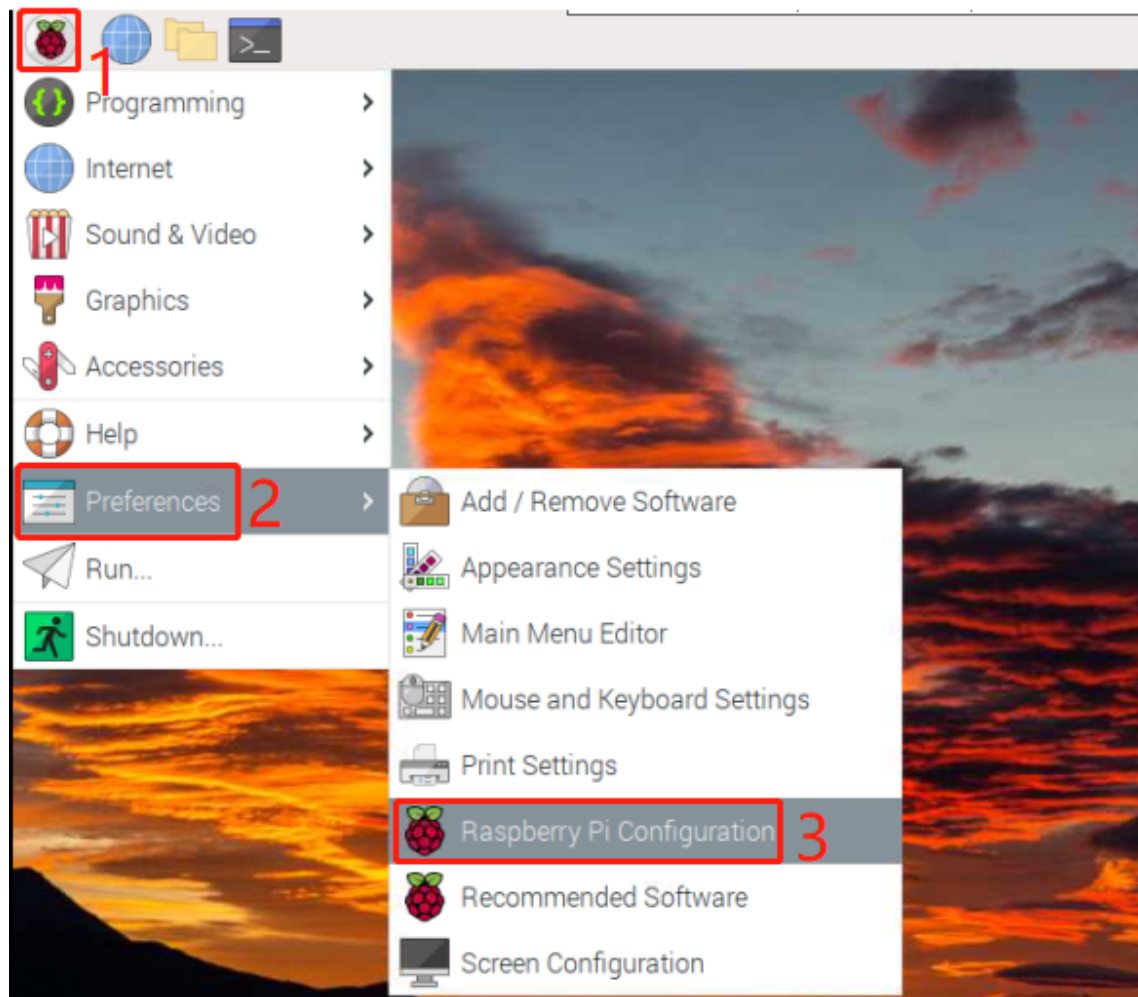
この方法では、Mac と Raspberry Pi 間の通信が暗号化されないことに注意してください。通信はあなたの家庭やビジネスネットワーク内で行われるため、保護されていなくても問題にはなりません。しかし、心配な場合は [VNC® Viewer](#) などの VNC アプリケーションをインストールできます。

また、一時的なモニター（TV）、マウス、キーボードを使用して直接 Raspberry Pi のデスクトップを開き、VNC を設定できると便利です。できない場合でも問題ありません。SSH コマンドを使用して Raspberry Pi の Bash シェルを開き、そのコマンドを使用して VNC を設定することもできます。

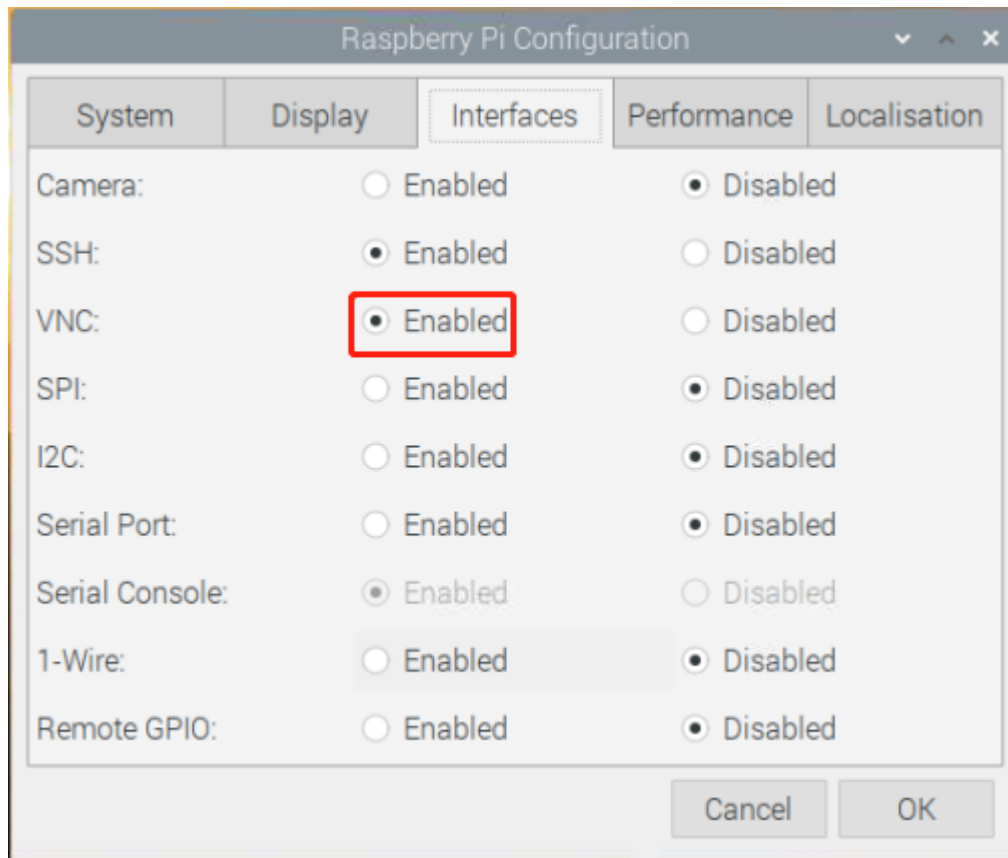
- **一時的なモニター（または TV）がありますか？**
- **一時的なモニター（または TV）がありませんか？**

一時的なモニター（または TV）がありますか？

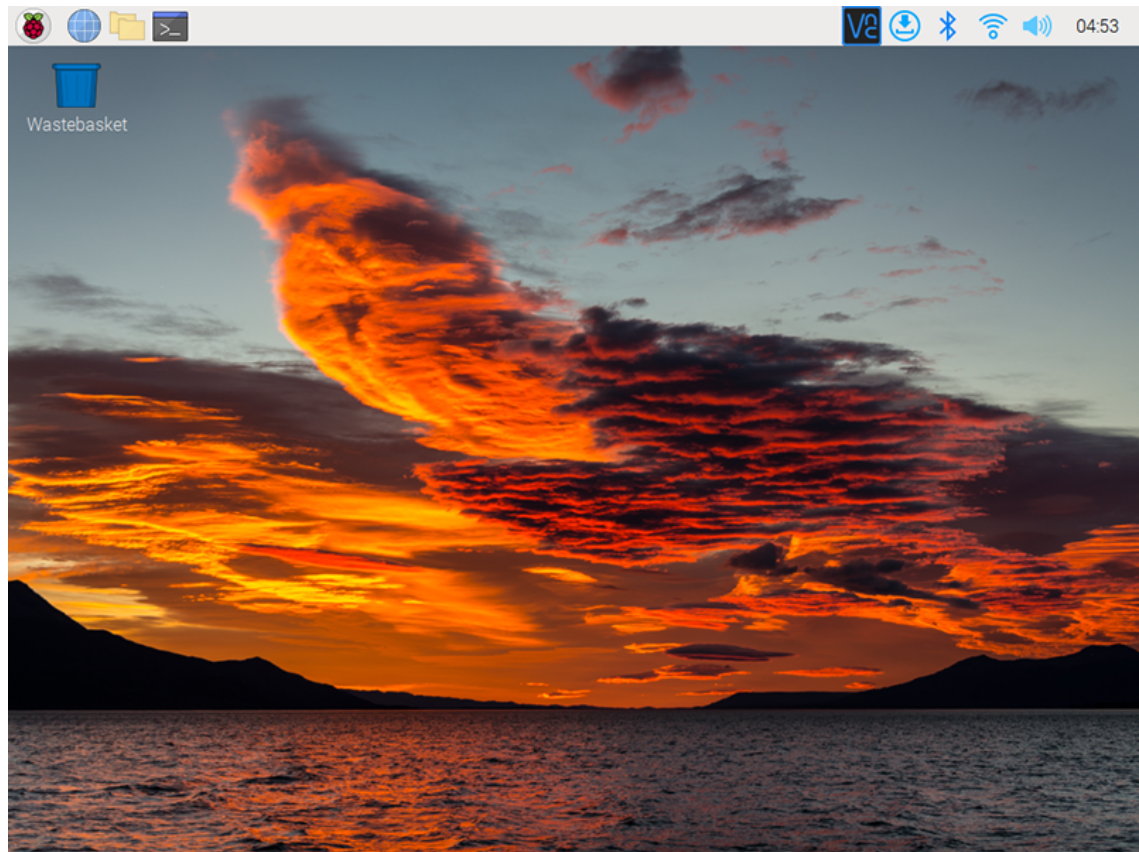
1. モニター（または TV）、マウス、キーボードを Raspberry Pi に接続し、電源を入れます。図の番号に従ってメニューを選択します。



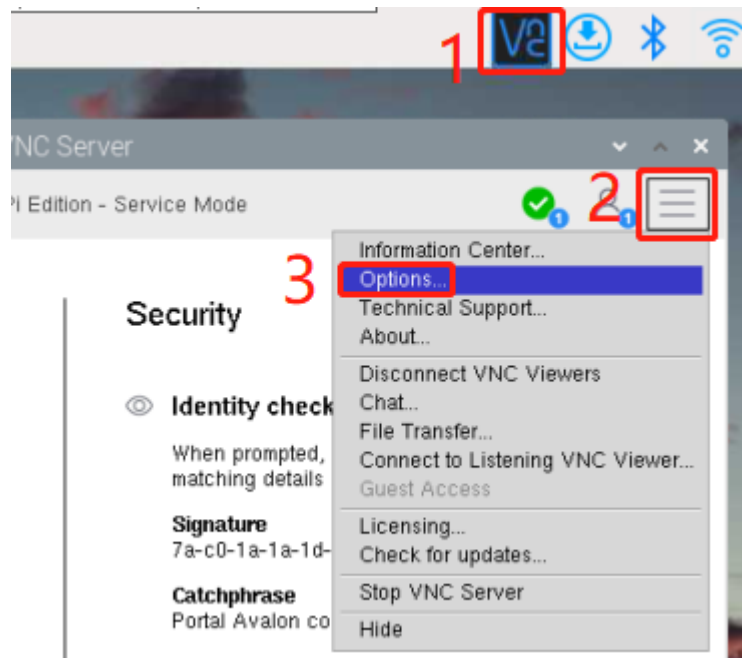
2. 次の画面が表示されます。 **インターフェース** タブで **VNC** を **有効** に設定し、 **OK** をクリックします。



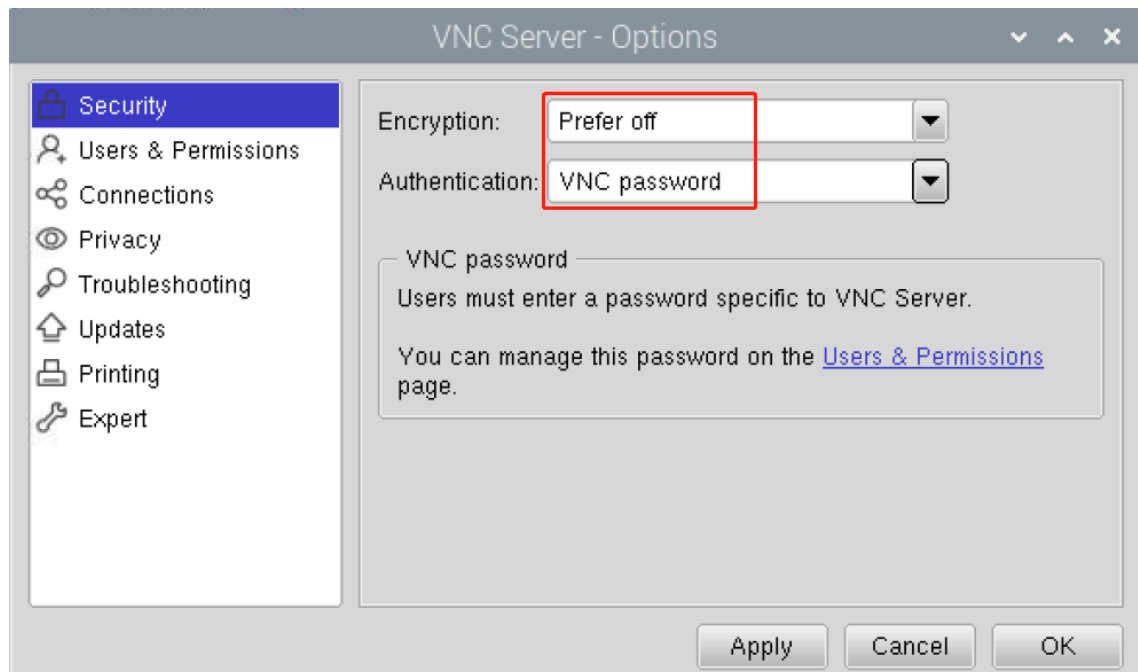
3. 画面の右上に VNC アイコンが表示され、VNC サーバーが起動します。



4. VNC アイコンをクリックして VNC サーバーウィンドウを開き、右上の **メニュー** ボタンをクリックして **オプション** を選択します。

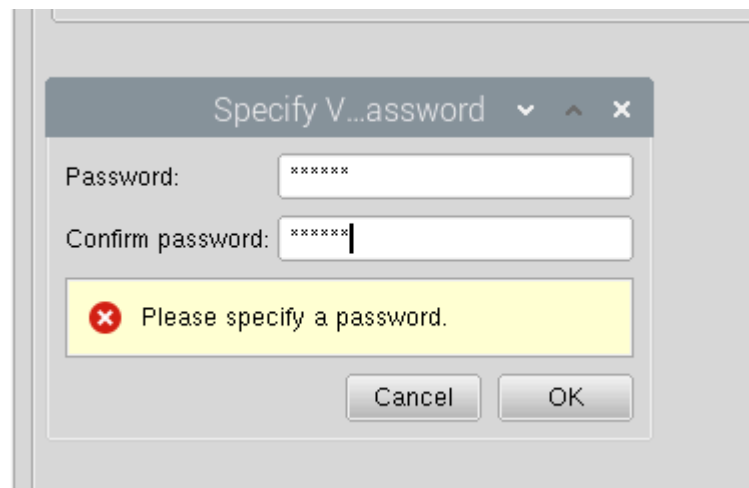


5. 次の画面が表示され、オプションを変更できます。



暗号化を **オフ推奨** に設定し、認証を **VNC パスワード** に設定します。

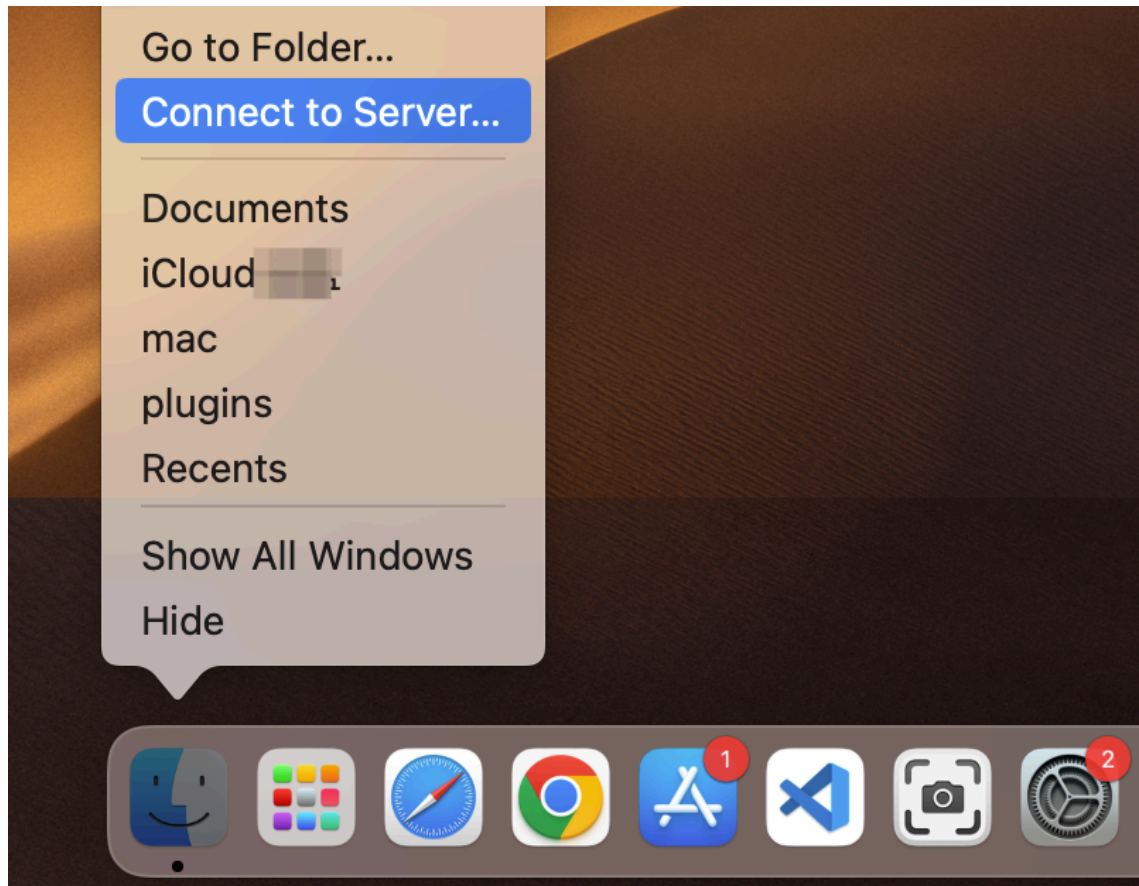
6. **OK** ボタンをクリックすると、パスワード入力画面が表示されます。Raspberry pi のパスワードと同じパスワードを使用することも、異なるパスワードを使用することもできますので、入力して **OK** をクリックしてください。



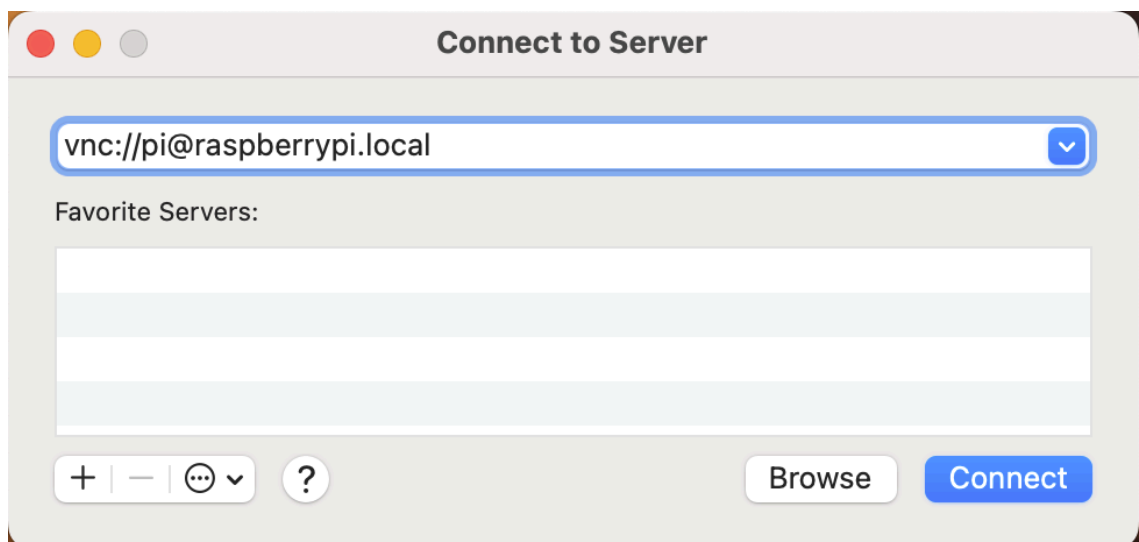
これで、Mac から接続する準備が整いました。モニターを切断しても大丈夫です。

ここからは、Mac 側での操作になります。

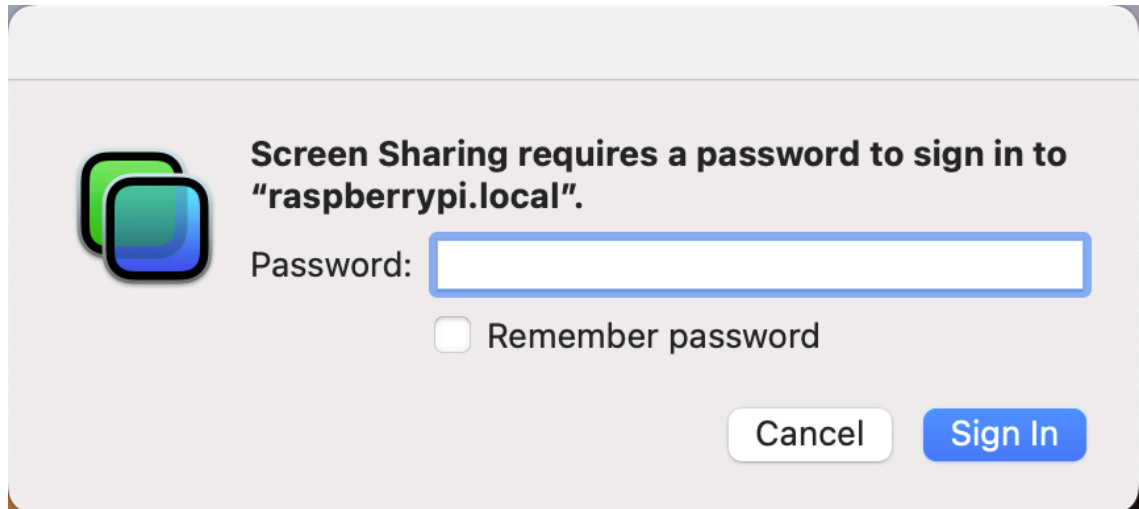
1. 次に、右クリックで開ける Finder のメニューから **サーバーへ接続** を選択します。



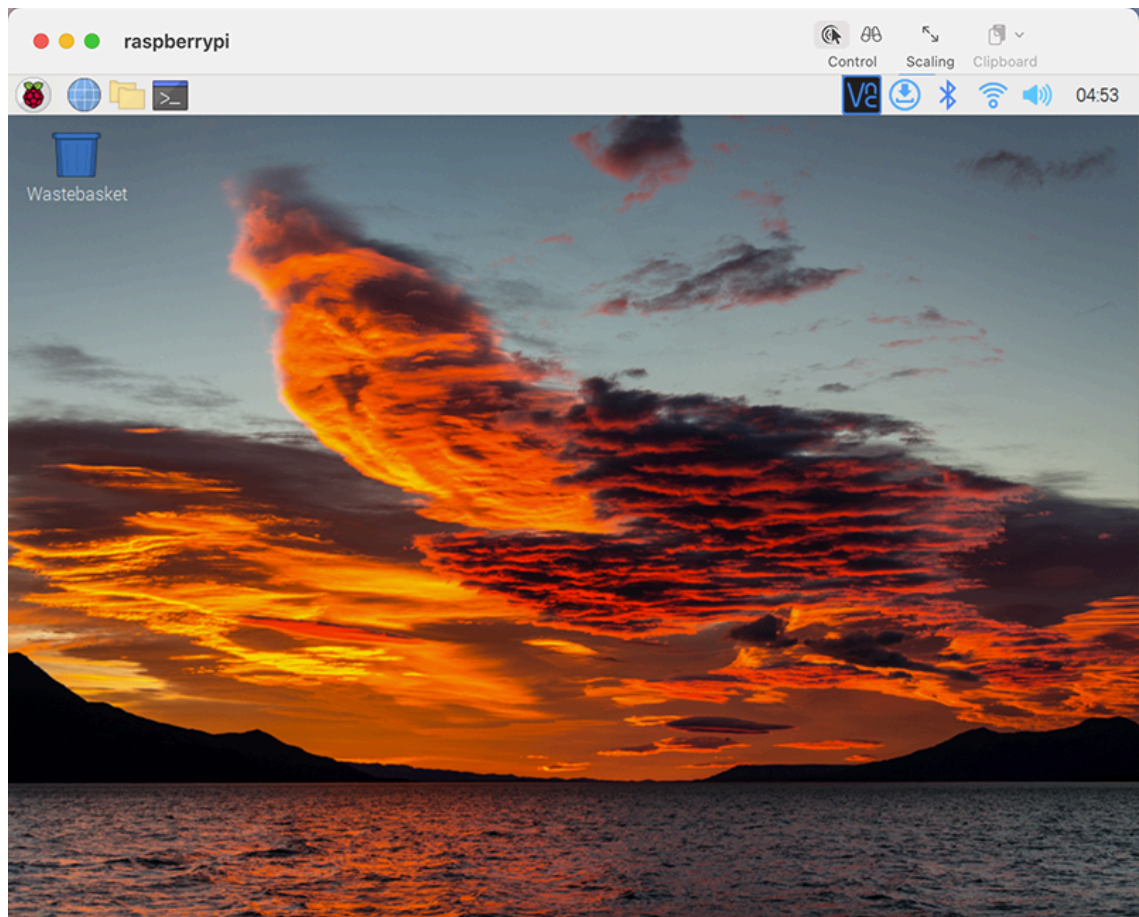
2. `vnc://<username>@<hostname>.local`（または `vnc://<username>@<IP address>`）と入力します。
入力後、**接続** をクリックします。



3. パスワードを求められますので、入力してください。



4. Raspberry Pi のデスクトップが表示され、そのまま Mac から操作できるようになります。

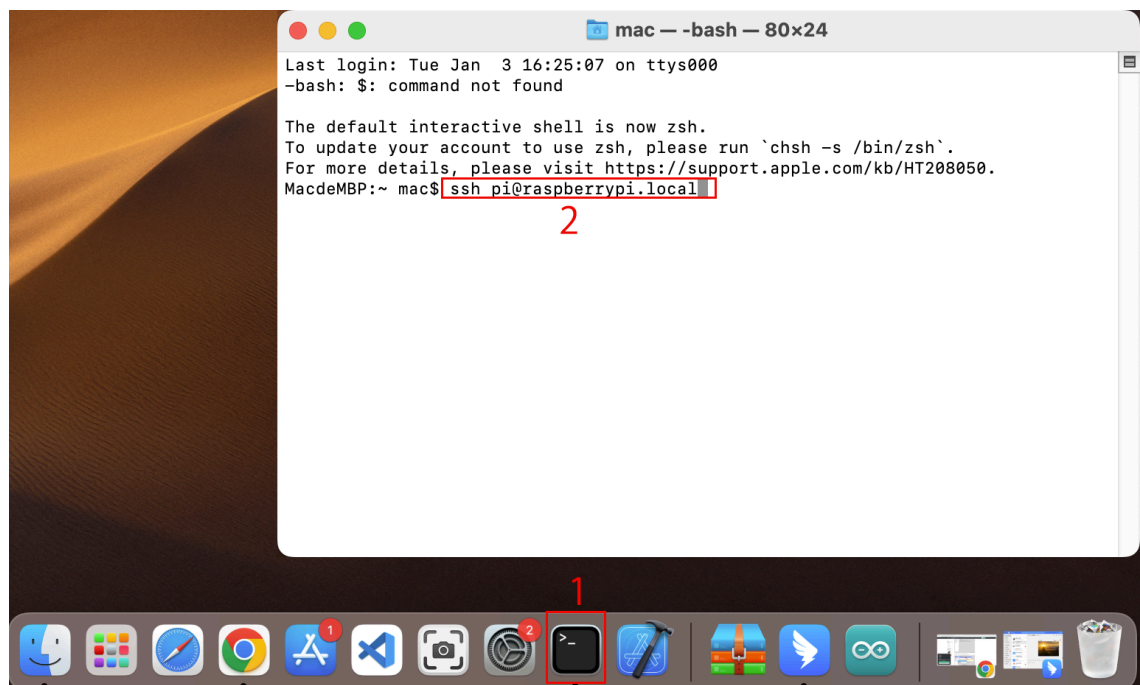


一時的なモニター（または TV）がありませんか？

- SSH コマンドを使用して Raspberry Pi の Bash シェルを開くことができます。
- Bash は Linux の標準デフォルトシェルです。
- シェル自体は、ユーザーが Unix/Linux を使用するときのコマンド（指示）です。
- 必要なことのほとんどはシェルを通じて行うことができます。
- Raspberry Pi 側の設定を行った後、Mac の **Finder** を使用して Raspberry Pi のデスクトップにアクセスできます。

1. `ssh <username>@<hostname>.local` と入力して Raspberry Pi に接続します。

```
ssh pi@raspberrypi.local
```



2. 初めてログインする時にのみ表示されるメッセージですので、**yes** と入力してください。

```
The authenticity of host 'raspberrypi.local
(2400:2410:2101:5800:635b:f0b6:2662:8cba)' can't be established.
ED25519 key fingerprint is SHA256:oo7x3ZSgAo032wD1tE8eW0fFM/kmewIvRwkBys6XRwg.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])?
```

3. Raspberry Pi のパスワードを入力してください。入力されたパスワードは表示されませんので、間違いがないように注意してください。

```
pi@raspberrypi.local's password:
Linux raspberrypi 5.15.61-v8+ #1579 SMP PREEMPT Fri Aug 26 11:16:44 BST 2022;
root@raspberrypi ~ # aarch64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Sep 22 12:18:22 2022
pi@raspberrypi:~ $
```

4. Mac から VNC 経由で Raspberry Pi にログインできるように設定してください。まず、以下のコマンドを実行してオペレーティングシステムを更新します。

```
sudo apt update
sudo apt upgrade
```

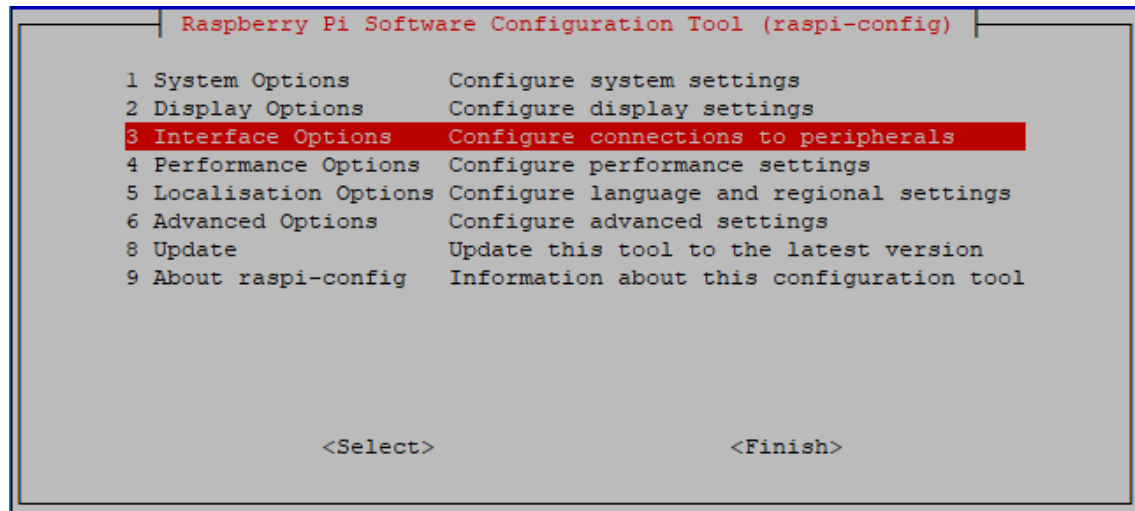
プロンプトが表示されたら、「Y」と入力して続行してください。

アップデートが完了するまでに時間がかかる場合があります。（その時のアップデートの量によります。）

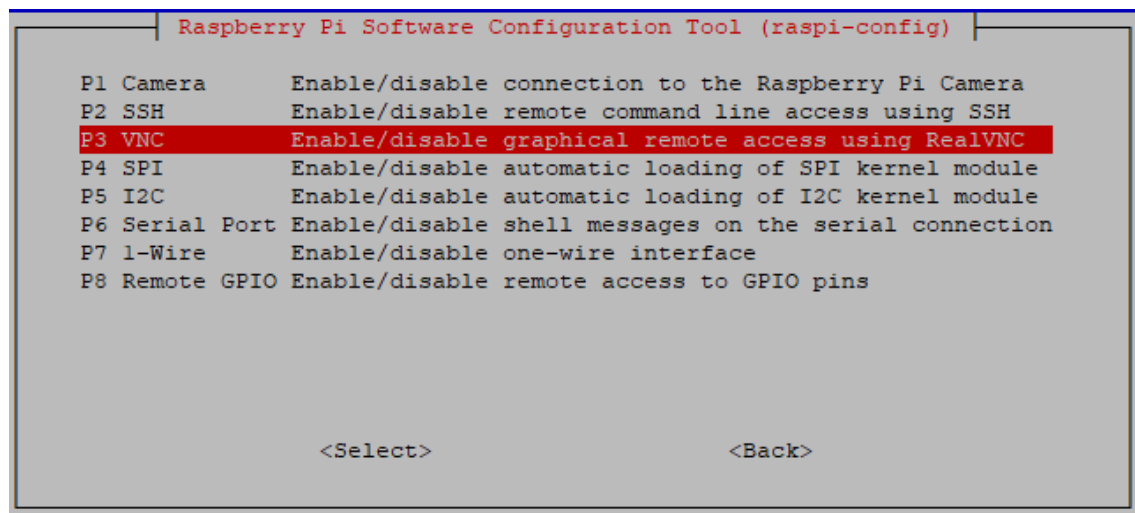
5. **VNC サーバー** を有効にするために、以下のコマンドを入力してください。

```
sudo raspi-config
```

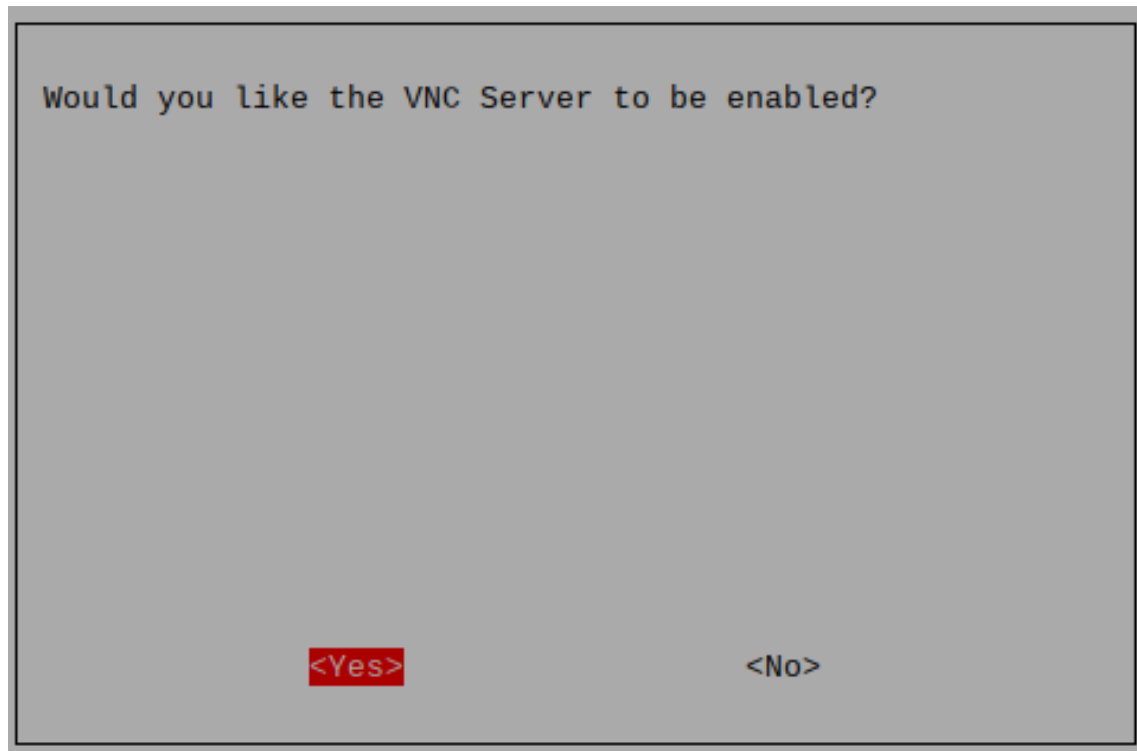
6. 次の画面が表示されます。キーボードの矢印キーで **インターフェースオプション** を選択し、**Enter** キーを押してください。



7. 次に VNC を選択してください。



8. キーボードの矢印キーを使用して <Yes> -> <OK> -> <Finish> の順に選択し、設定を完了させてください。

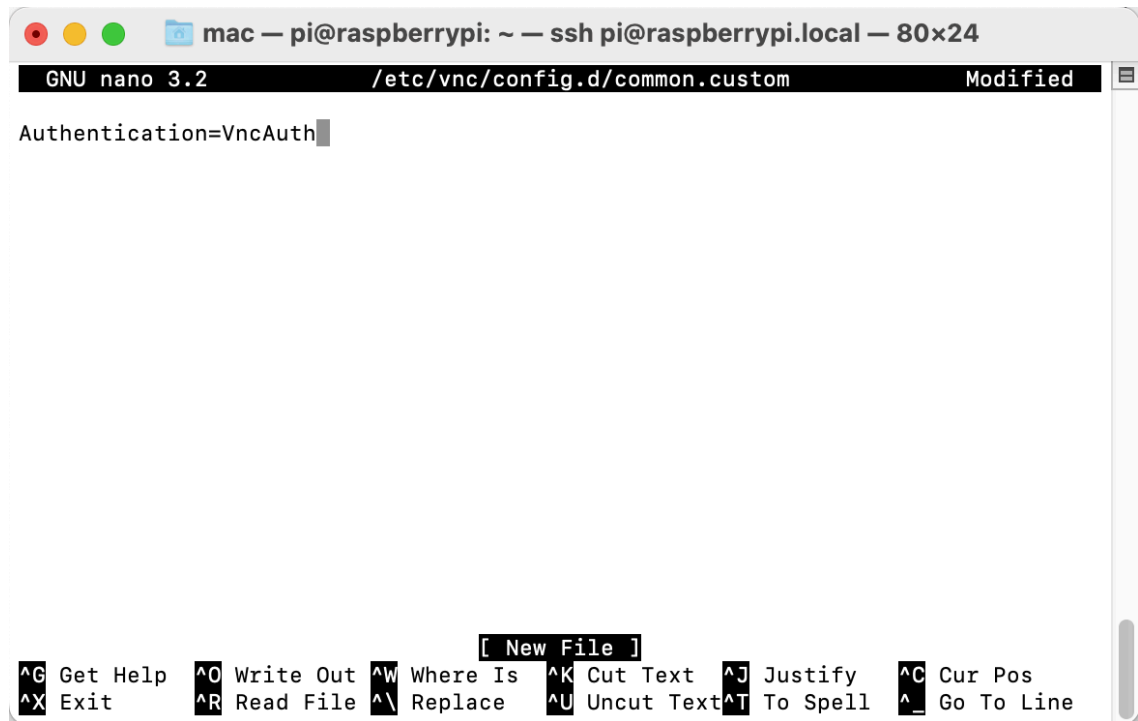


9. VNC サーバーが起動したので、Mac から接続するための設定を変更しましょう。

コンピューター上のすべてのユーザーアカウントのすべてのプログラムに対してパラメータを指定するために、`/etc/vnc/config.d/common.custom` を作成してください。

```
sudo nano /etc/vnc/config.d/common.custom
```

`Authentication=VncAuthenter` と入力した後、`Ctrl+X -> Y -> Enter` を押して保存して終了してください。



```
mac — pi@raspberrypi: ~ — ssh pi@raspberrypi.local — 80x24
GNU nano 3.2 /etc/vnc/config.d/common.custom Modified
Authentication=VncAuth

[ New File ]
^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Uncut Text ^T To Spell  ^_ Go To Line
```

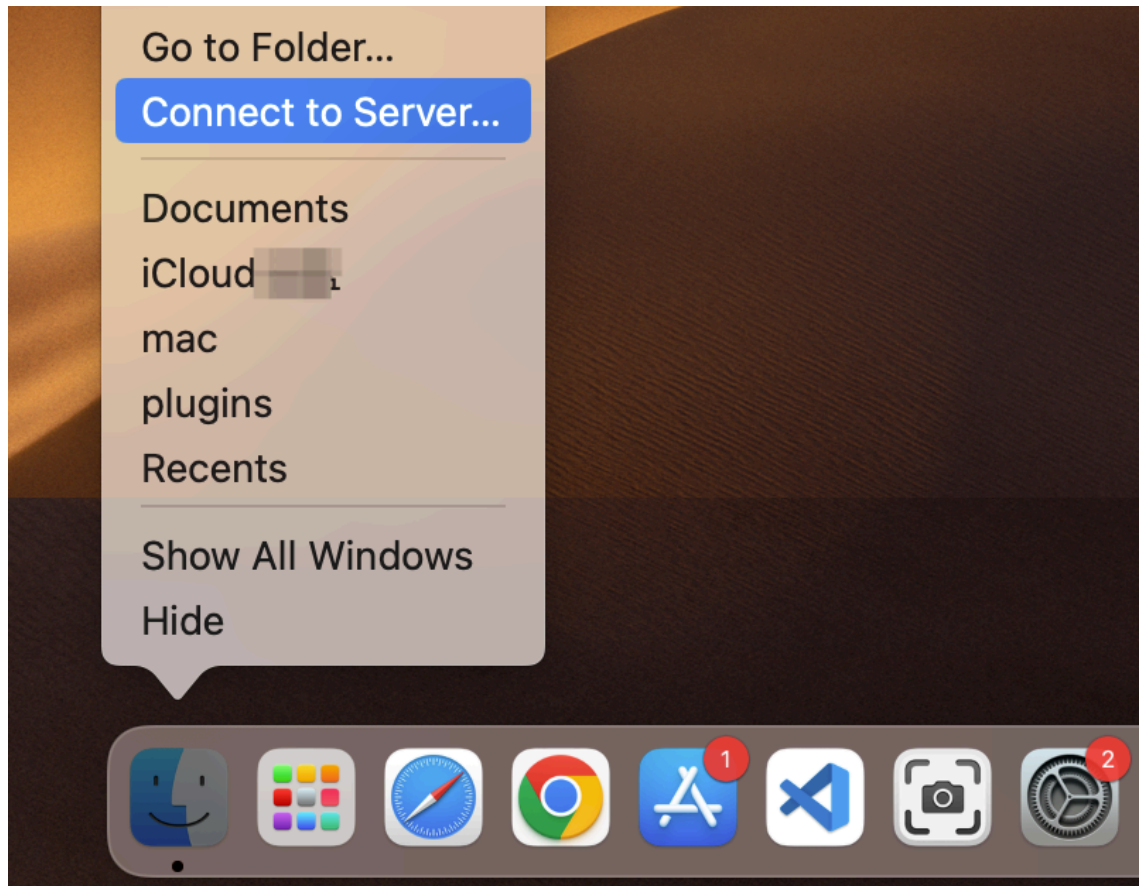
10. さらに、Mac から VNC 経由でログインするためのパスワードを設定してください。Raspberry Pi のパスワードと同じものを使用することも、異なるパスワードを使用することもできます。

```
sudo vncpasswd -service
```

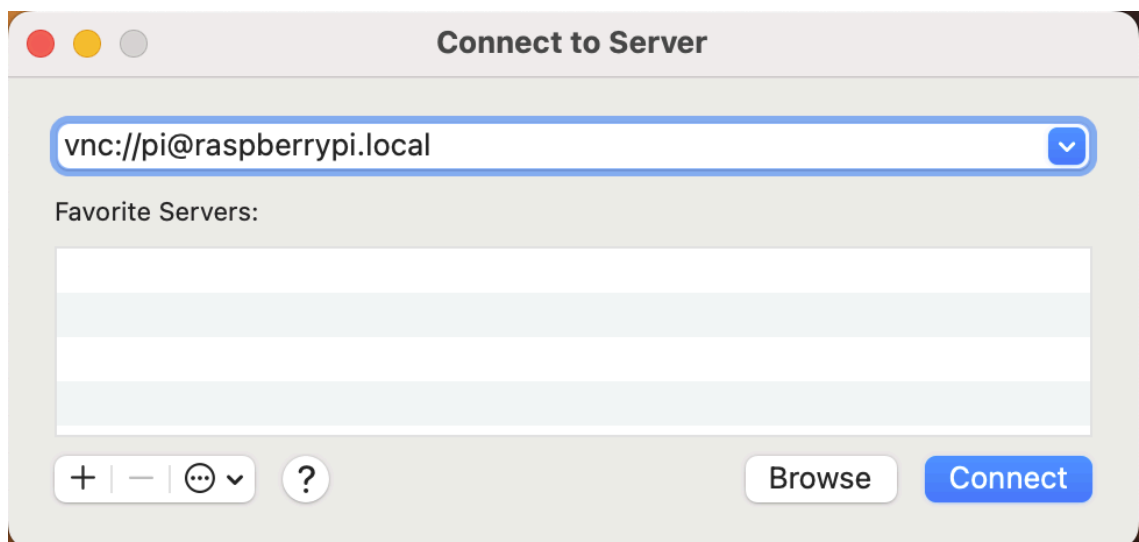
11. 設定が完了したら、変更を適用するために Raspberry Pi を再起動してください。

```
sudo sudo reboot
```

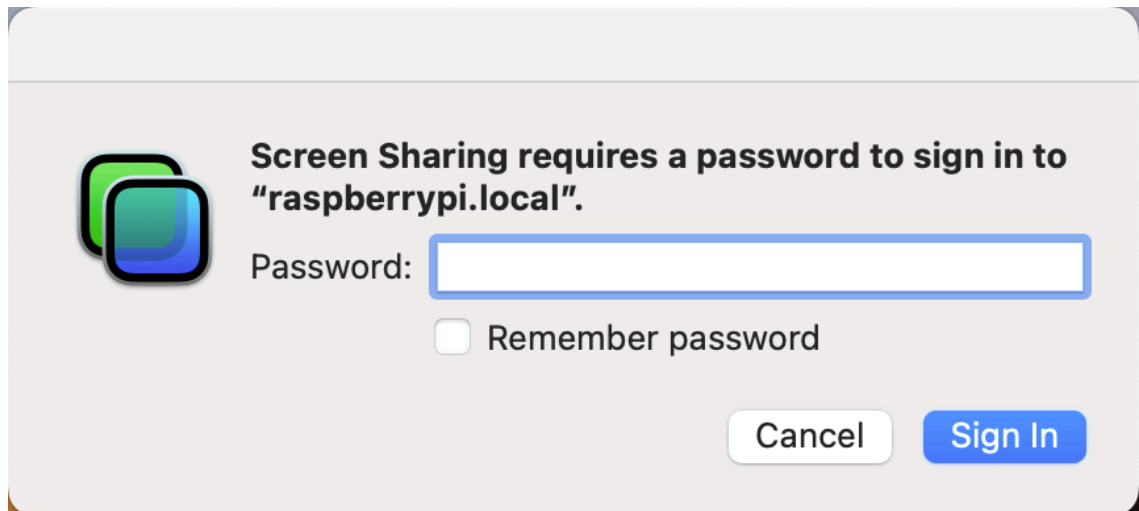
12. 次に、右クリックで開くことができる **Finder** のメニューから **サーバーに接続** を選択してください。



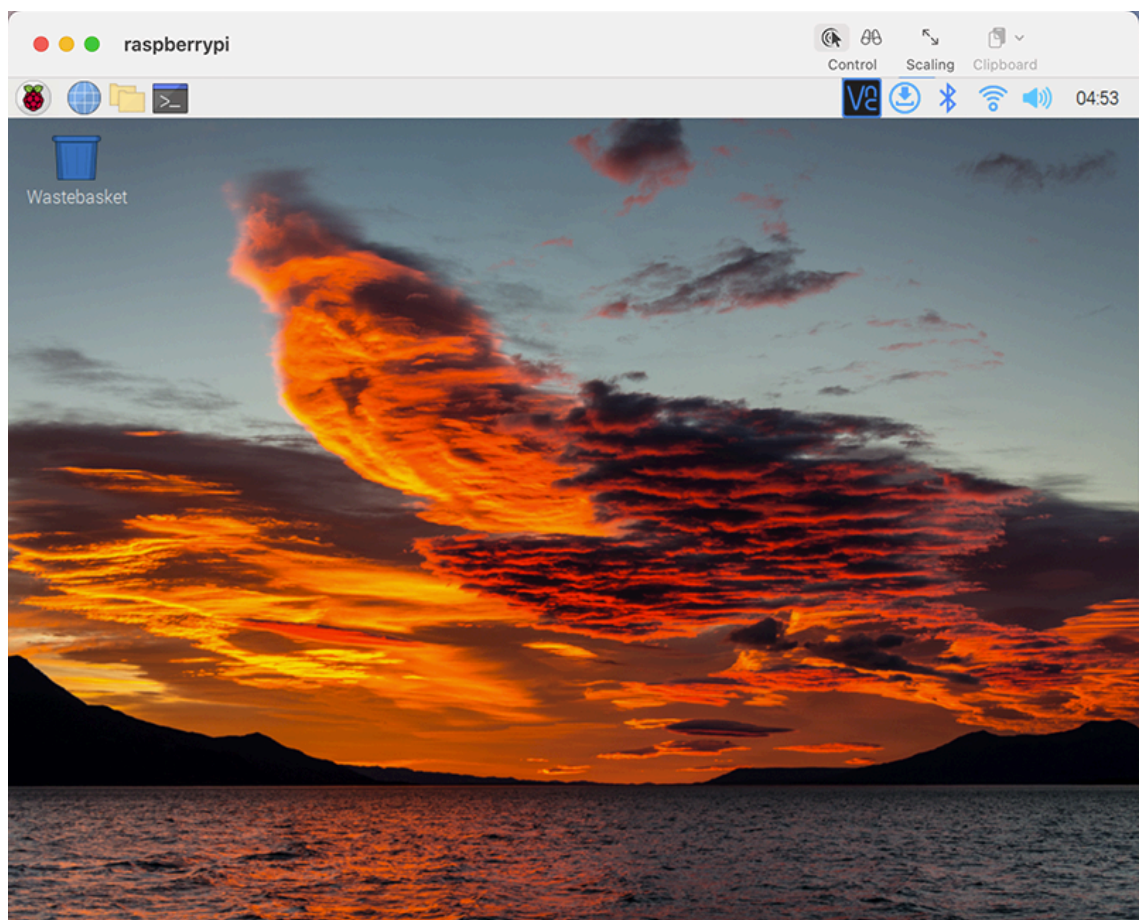
13. `vnc://<username>@<hostname>.local`（または `vnc://<username>@<IP address>`）と入力します。
入力後、**接続** をクリックします。



14. パスワードを求められますので、入力してください。



15. Raspberry Pi のデスクトップが表示され、そのまま Mac から操作できるようになります。

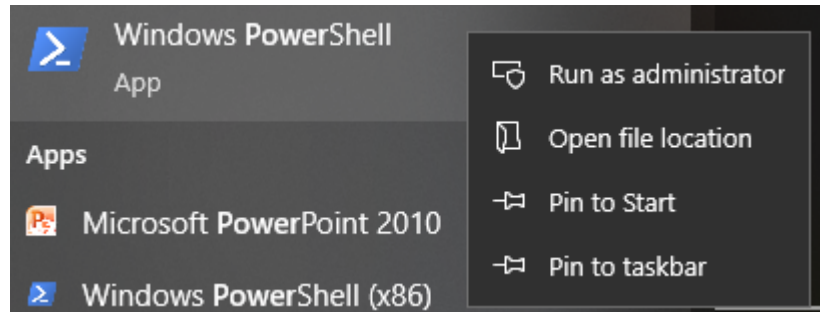


Windows ユーザー

リモートで Raspberry Pi にログイン

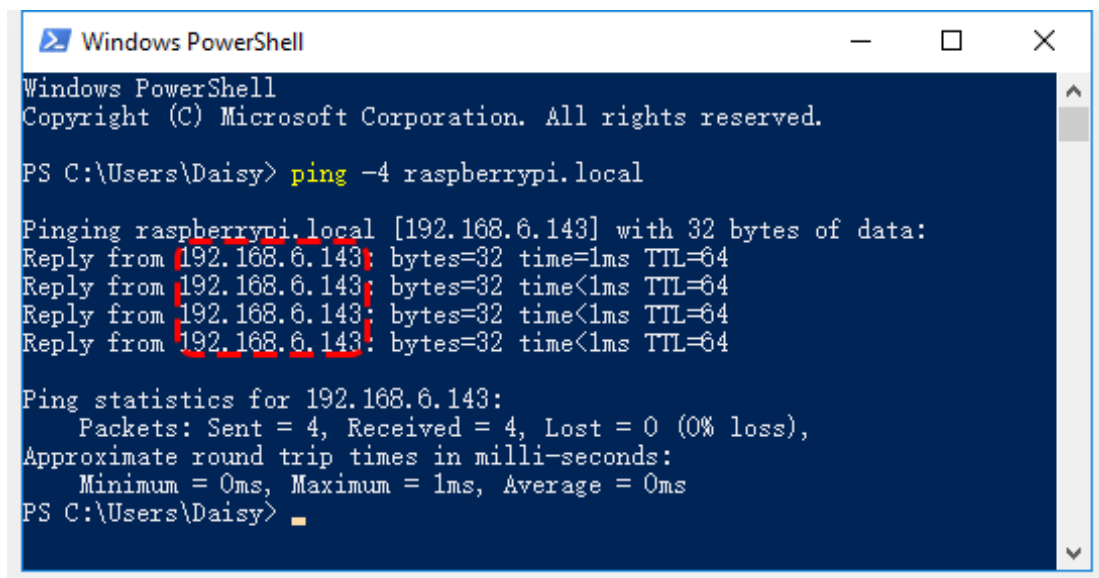
Windows 10 を使用している場合、以下の方法でリモートから Raspberry Pi にログインできます。

1. Windows デスクトップの検索ボックスに powershell と入力し、表示されたメニューから Windows PowerShell を右クリックして **管理者として実行** を選択します。



2. 次に、`ping -4 <hostname>.local` と入力して Raspberry Pi の IP アドレスを確認します。

```
ping -4 raspberrypi.local
```



上記のように、ネットワークに接続された後に Raspberry Pi の IP アドレスを確認できます。

- ターミナルが `Ping request could not find host pi.local. Please check the name and try again.` と表示された場合は、入力したホスト名が正しいか確認してください。
- まだ IP を取得できない場合は、Raspberry Pi のネットワークまたは WiFi の設定を確認してください。

3. この時点で、ssh <username>@<hostname>.local（または ssh <username>@<IP address>）を使用して Raspberry Pi にログインできるようになります。

```
ssh pi@raspberrypi.local
```

警告: ``The term 'ssh' is not recognized as the name of a cmdlet...``というプロンプトが表示された場合、

システムが古いか ssh ツールがプリインストールされていないことを意味します。手動で:ref:`openssh\_powershell`を行う必要があります。

または、:ref:`login\_windows`のようなサードパーティのツールを使用してください。

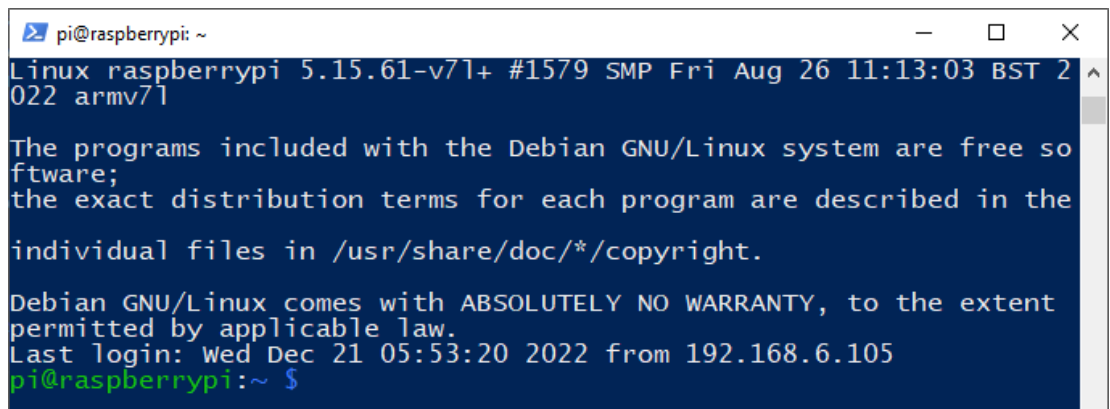
4. 初めてログインする際にのみ表示されるメッセージですので、``yes``と入力してください。

```
The authenticity of host 'raspberrypi.local
(2400:2410:2101:5800:635b:f0b6:2662:8cba)' can't be established.
ED25519 key fingerprint is SHA256:oo7x3ZSgAo032wD1tE8eW0fFM/kmewIvRwkBys6XRwg.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])?
```

5. 以前に設定したパスワードを入力してください。（私の場合は ``raspberrypi`` です。）

注釈: パスワードを入力しても文字は画面に表示されませんが、これは正常です。正しいパスワードを入力する必要があります。

6. これで Raspberry Pi が接続され、次のステップに進む準備ができました。



```
pi@raspberrypi: ~
Linux raspberrypi 5.15.61-v7l+ #1579 SMP Fri Aug 26 11:13:03 BST 2022 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Dec 21 05:53:20 2022 from 192.168.6.105
pi@raspberrypi:~ $
```

リモートデスクトップ

コマンドウィンドウを使用して Raspberry Pi にアクセスするのに満足できない場合は、リモートデスクトップ機能を使用して GUI を介して簡単に Raspberry Pi 上のファイルを管理することもできます。

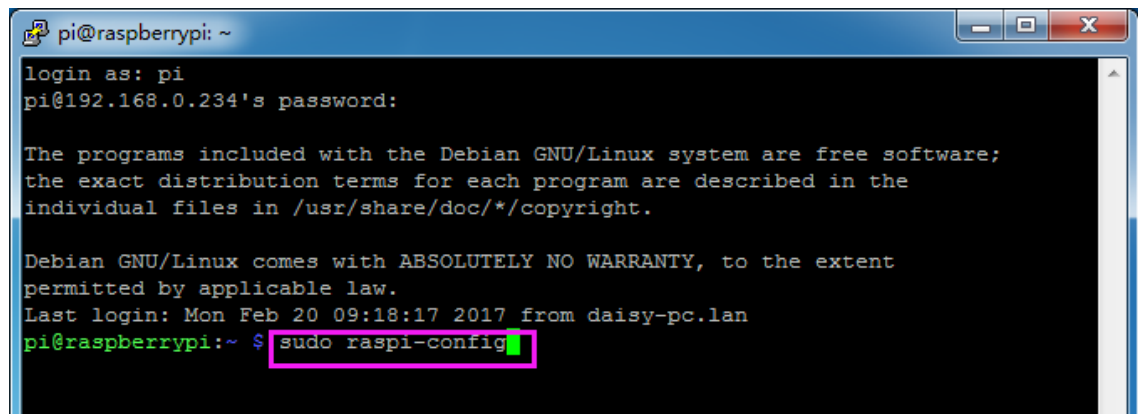
ここでは `VNC® Viewer` <<https://www.realvnc.com/en/connect/download/viewer/>>`\_` を使用します。

VNC サービスを有効にする

VNC サービスはシステムにインストールされています。デフォルトでは VNC は無効になっています。設定で有効にする必要があります。

1. 以下のコマンドを入力してください：

```
sudo raspi-config
```

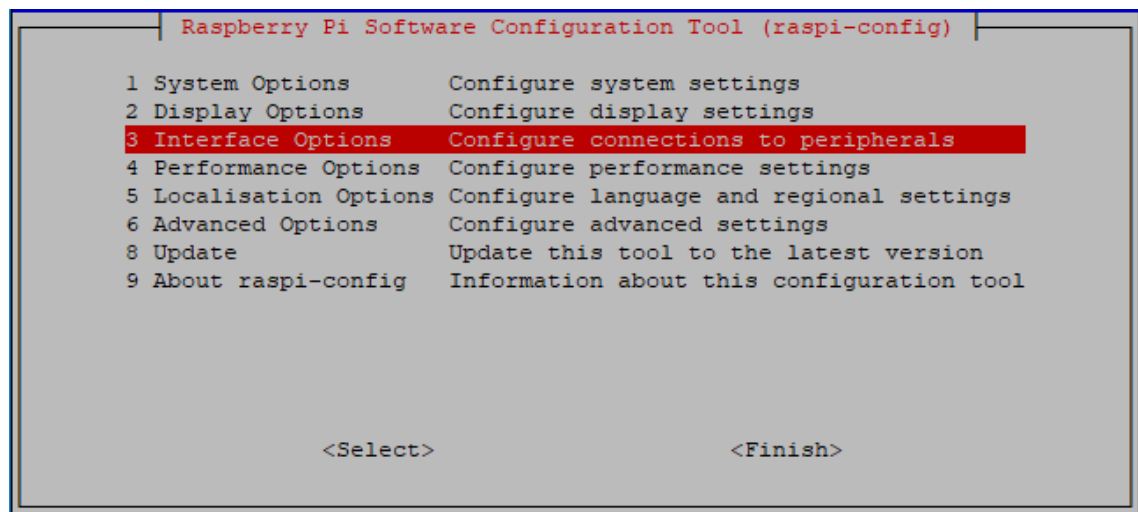


```
pi@raspberrypi: ~
login as: pi
pi@192.168.0.234's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Feb 20 09:18:17 2017 from daisy-pc.lan
pi@raspberrypi:~ $ sudo raspi-config
```

2. キーボードの下矢印キーを押して****3**** ****インターフェースオプション****を選択し、****Enter****キーを押します。

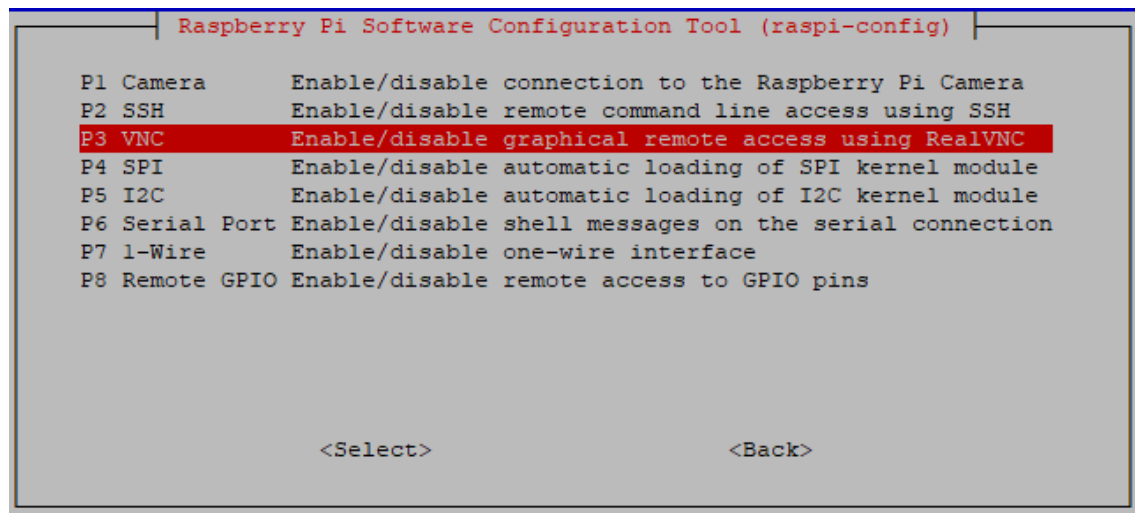


```
Raspberry Pi Software Configuration Tool (raspi-config)

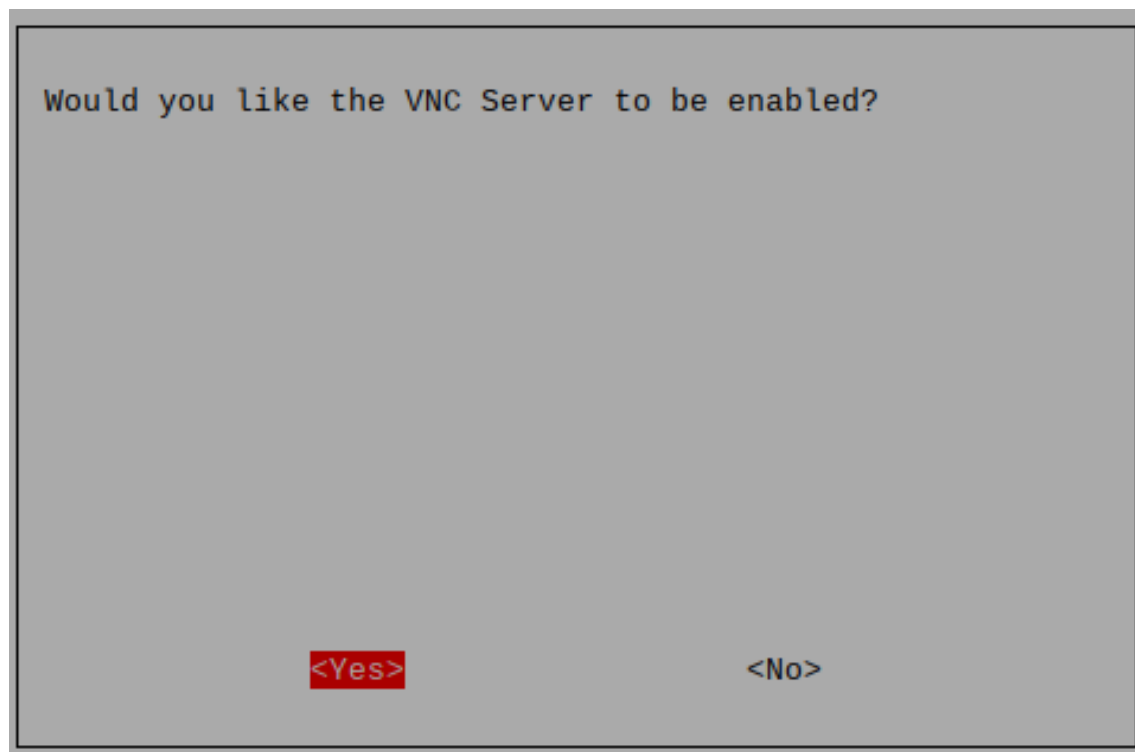
1 System Options          Configure system settings
2 Display Options         Configure display settings
3 Interface Options       Configure connections to peripherals
4 Performance Options     Configure performance settings
5 Localisation Options    Configure language and regional settings
6 Advanced Options        Configure advanced settings
8 Update                  Update this tool to the latest version
9 About raspi-config      Information about this configuration tool

<Select>                  <Finish>
```

3. その後、****VNC****を選択します。

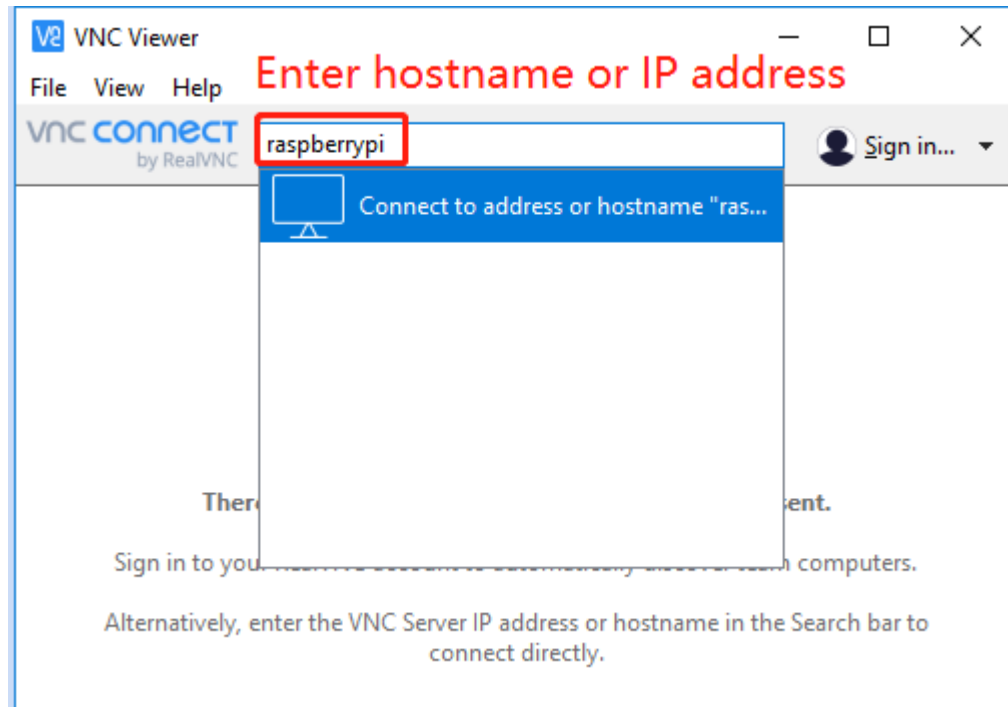


4. キーボードの矢印キーを使って**<Yes>** -> <OK> -> <Finish> を選択し、設定を完了させます。

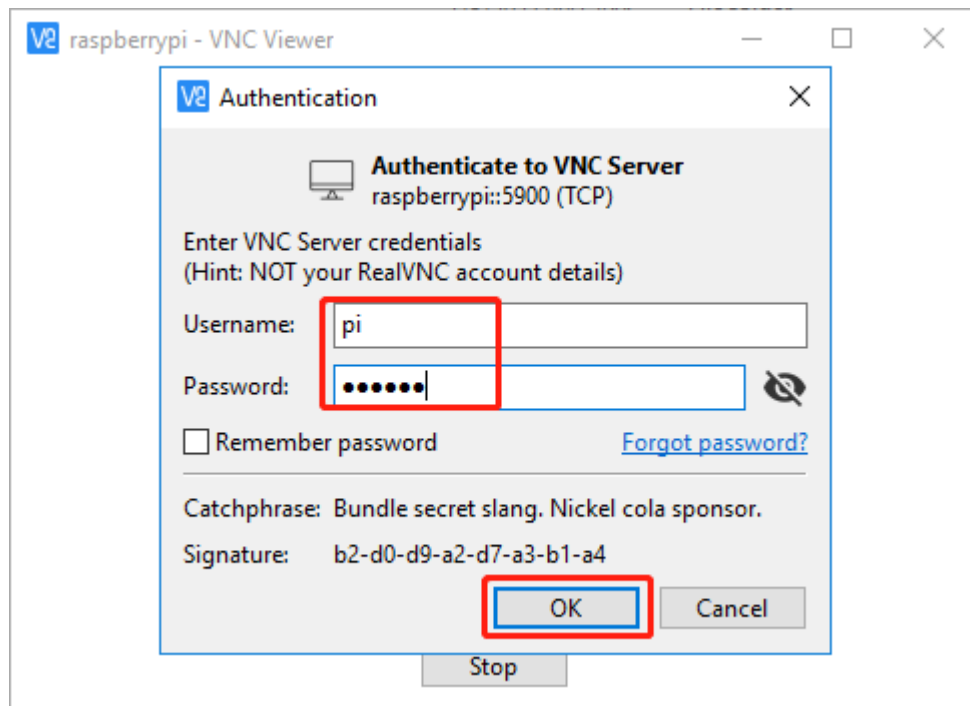


VNC にログイン

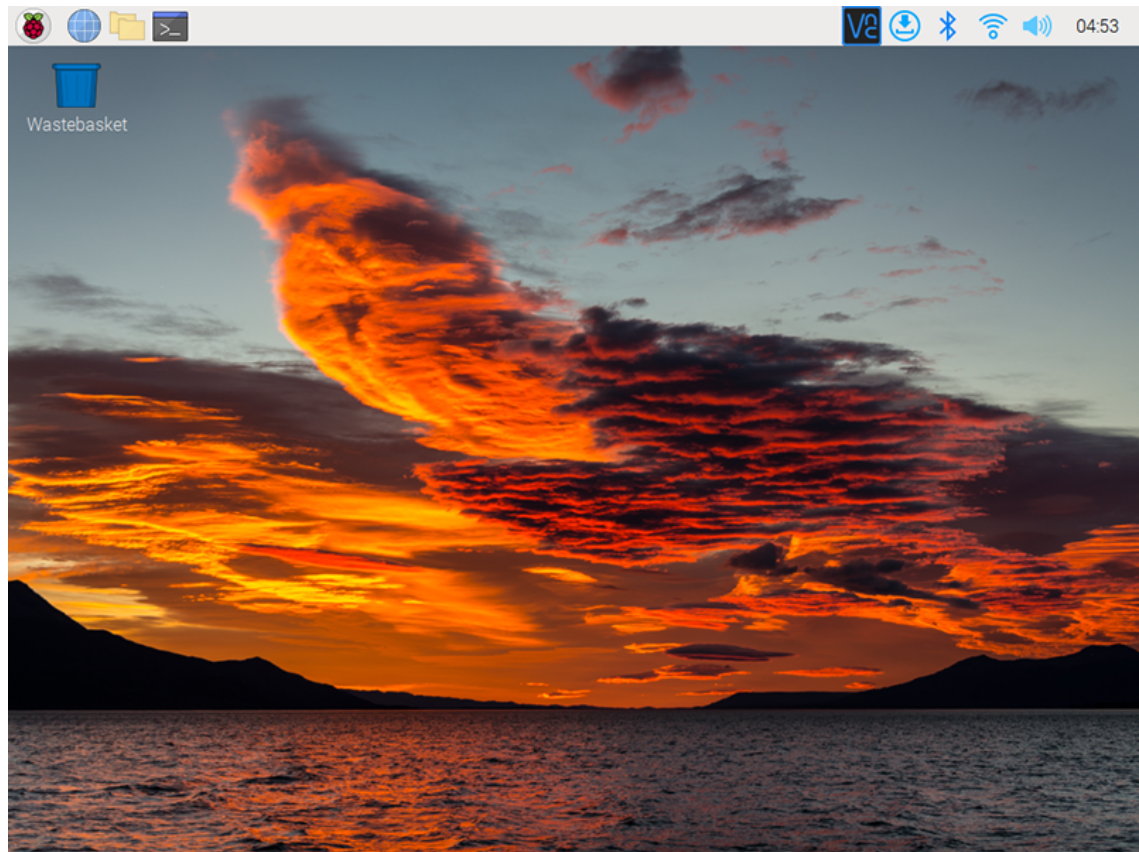
1. 個人のコンピュータに `VNC Viewer` <<https://www.realvnc.com/en/connect/download/viewer/>>`\_` をダウンロードしてインストールする必要があります。
2. インストールが完了したら開き、ホスト名または IP アドレスを入力して Enter キーを押します。



3. Raspberry Pi の名前とパスワードを入力した後、**OK**をクリックします。

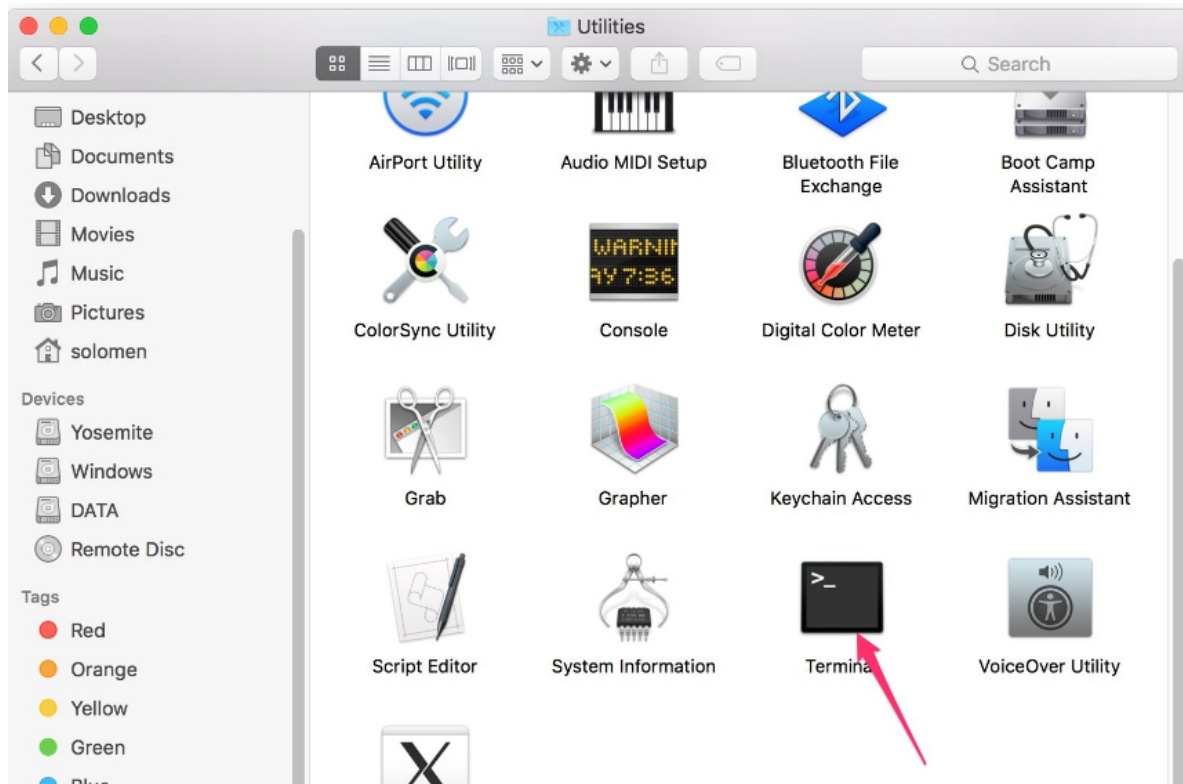


4. これで Raspberry Pi のデスクトップを見ることができます。



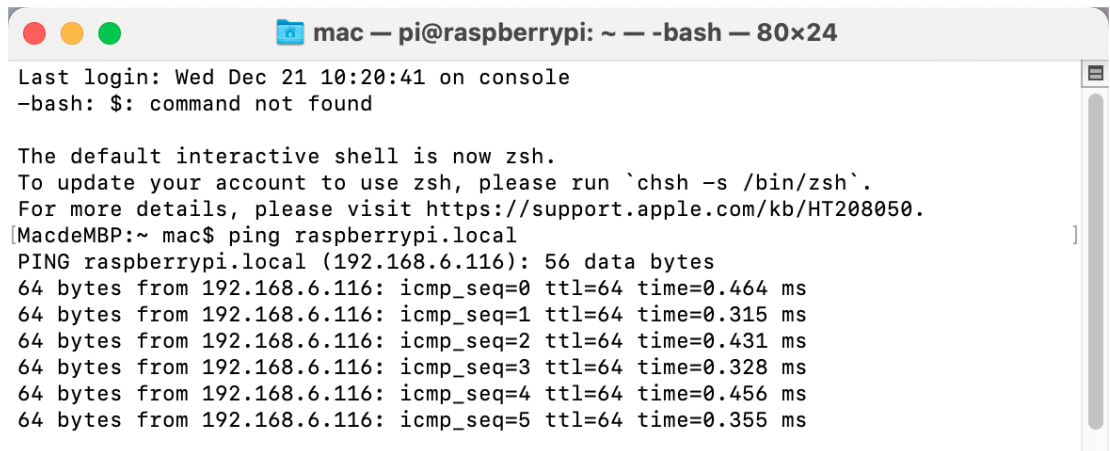
Linux /Unix ユーザー

#. Go to **Applications->Utilities**, find the **Terminal**, and open it.



1. `ping <hostname>.local` を入力して、Raspberry Pi が同じネットワーク上にあるか確認してください。

```
ping raspberrypi.local
```



上記のように、ネットワークに接続された後、Raspberry Pi の IP アドレスを確認できます。

- ターミナルが `Ping request could not find host pi.local. Please check the name and try again.` と表示された場合は、入力したホスト名が正しいか確認してください。
- まだ IP を取得できない場合は、Raspberry Pi のネットワークまたは WiFi の設定を確認してください。

2. `ssh <username>@<hostname>.local`（または `ssh <username>@<IP address>`）と入力します。

```
ssh pi@raspberrypi.local
```

注釈: プロンプトが ``The term 'ssh' is not recognized as the name of a cmdlet...`` と表示された場合、

システムが古すぎて ssh ツールがプリインストールされていないことを意味します。手動で [Powershell 経由で OpenSSH をインストールする](#) を行う必要があります。

または、[PuTTY](#) のようなサードパーティのツールを使用してください。

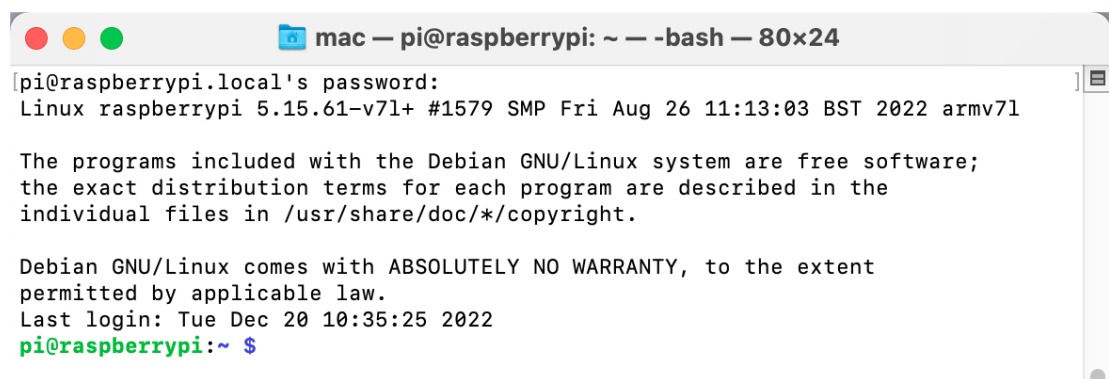
3. 次のメッセージは初めてログインするときのみ表示されるので、`yes` と入力してください。

```
The authenticity of host 'raspberrypi.local
(2400:2410:2101:5800:635b:f0b6:2662:8cba)' can't be established.
ED25519 key fingerprint is SHA256:oo7x3ZSgAo032wD1tE8eW0fFM/kmewIvRwkBys6XRwg.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])?
```

4. 以前に設定したパスワードを入力してください。（私の場合は `raspberry` です。）

注釈: パスワードを入力するとき、文字はウィンドウに表示されませんが、これは正常です。正しいパスワードを入力する必要があります。

5. これで Raspberry Pi が接続され、次のステップに進む準備ができました。



```
mac - pi@raspberrypi: ~ - -bash - 80x24
[pi@raspberrypi.local's password:
Linux raspberrypi 5.15.61-v7l+ #1579 SMP Fri Aug 26 11:13:03 BST 2022 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Dec 20 10:35:25 2022
pi@raspberrypi:~ $
```


1.6 サーボ構成

また、このキットで使用されるサーボはソフトウェアによって調整され、他のサーボほど物理的な固定点がないため、ここではソフトウェアを介してサーボを構成する。最初に、構成の前にソフトウェアの実装を完了してください。

注釈: 本章では、バッテリーを取り付けて電源スイッチを ON にスライドすることを忘れないでください。

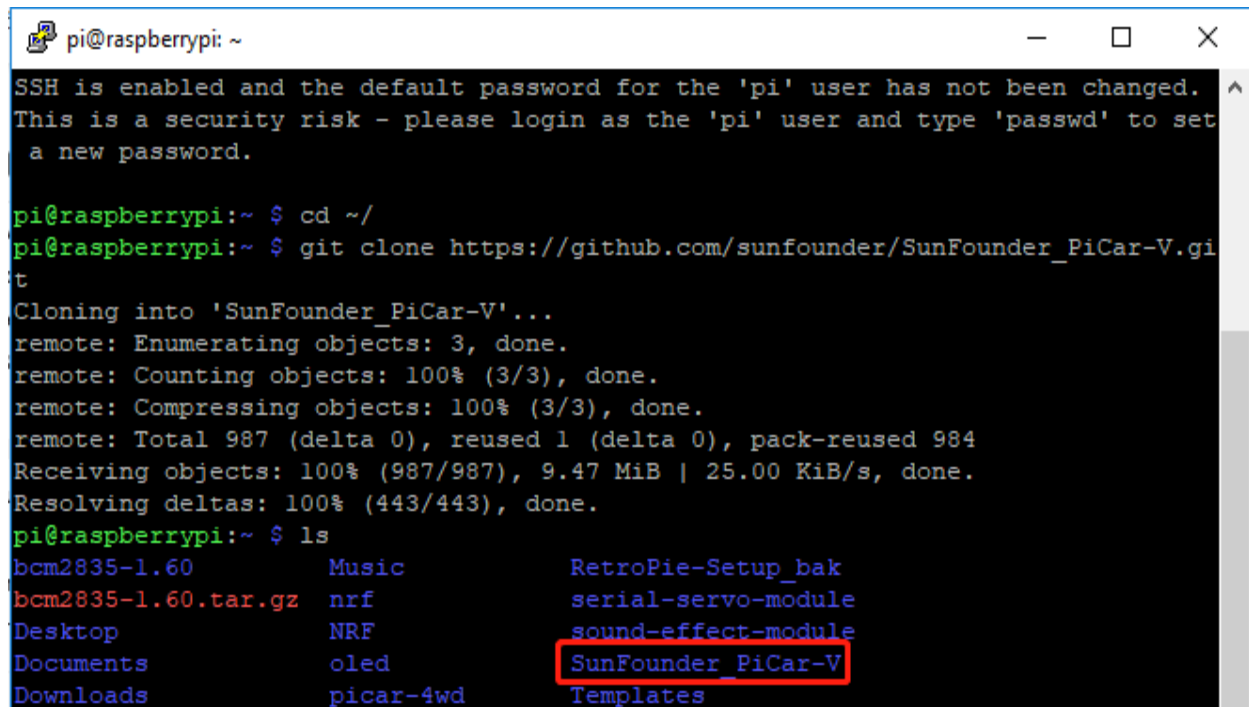
1.6.1 ソースコードを取得する

ソースコードは Github リポジトリにある。 `git clone` でソースコードをダウンロードする：

```
cd ~/
git clone https://github.com/sunfounder/SunFounder_PiCar-V -b V3.0
```

注釈: 入力時に十分に注意してください。ユーザー名とパスワードの入力を求められた場合は、入力を間違えている可能性がある。それは発生した場合、Ctrl+C を押して終了し、再試行してください。

ls コマンドで確認すると、コードディレクトリ `SunFounder_PiCar-V` が表示される：



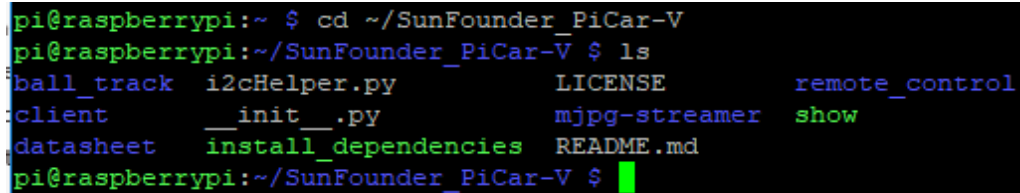
```
pi@raspberrypi: ~
SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set
a new password.

pi@raspberrypi:~ $ cd ~/
pi@raspberrypi:~ $ git clone https://github.com/sunfounder/SunFounder_PiCar-V.git
Cloning into 'SunFounder_PiCar-V'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 987 (delta 0), reused 1 (delta 0), pack-reused 984
Receiving objects: 100% (987/987), 9.47 MiB | 25.00 KiB/s, done.
Resolving deltas: 100% (443/443), done.
pi@raspberrypi:~ $ ls
bcm2835-1.60      Music          RetroPie-Setup_bak
bcm2835-1.60.tar.gz nrf            serial-servo-module
Desktop          NRF            sound-effect-module
Documents        oled          SunFounder_PiCar-V
Downloads        picar-4wd     Templates
```


1.6.2 コードディレクトリに入る

```
cd ~/SunFounder_PiCar-V
```

コードディレクトリに入ると、インストールスクリプトが表示される：



```
pi@raspberrypi:~ $ cd ~/SunFounder_PiCar-V
pi@raspberrypi:~/SunFounder_PiCar-V $ ls
ball_track  i2cHelper.py      LICENSE          remote_control
client      __init__.py       mjpg-streamer   show
datasheet  install_dependencies  README.md
```

1.6.3 スクリプトを介して環境をインストールする

install_dependencies スクリプトを使用して、必要なすべてのソフトウェアと構成を実行することはできる。代わりに一步一步に実行する場合は、「**付録 1：サーバーインストールスクリプトの機能**」の手順に従ってください。

```
sudo ./install_dependencies
```

注釈:

1. インストールスクリプトは、必要な部品をインストールし、動作環境用に構成する。インストール中に Raspberry がインターネットに接続されていることを確認してください。接続しない場合は、失敗する恐れがある。
2. インストールを完了した後、Raspberry Pi は再起動すると示す。再起動するには yes と入力してください。

1.6.4 サーボを 90 度に設定する

再起動後、picar ツールを実行する：

```
cd ~/SunFounder_PiCar-V
picar servo-install
```

注釈：「`OSError: [Errno 121] Remote I/O error`」エラーメッセージが表示された場合は、raspi-config を開く：

```
sudo raspi-config
```

次に、**3 Interfacing Options** → **<P5 I2C>** → **<YES>** → **<OK>**を選択して I2C サービスを有効にします。キーボードの上、下、左、右のキーを使って選択できます、最後は<Enter>キーを押して確認します。

コードの実行後、ロッカーアームをサーボに差し込む。ロッカーアームが時計回りと反時計回りに回転し、特定の位置で停止する。サーボが良い状態にあると示す。以下の条件のいずれかがサーボに発生した場合、サーボは不良である：

- 1) 雑音あり、熱い。
- 2) サーボラインを抜いてロッカーアームを回すと「カ」「カ」「カ」のように聞こえるか、またはギアの駆動音が鳴らさない。
- 3) ゆっくりで継続的に回してください。

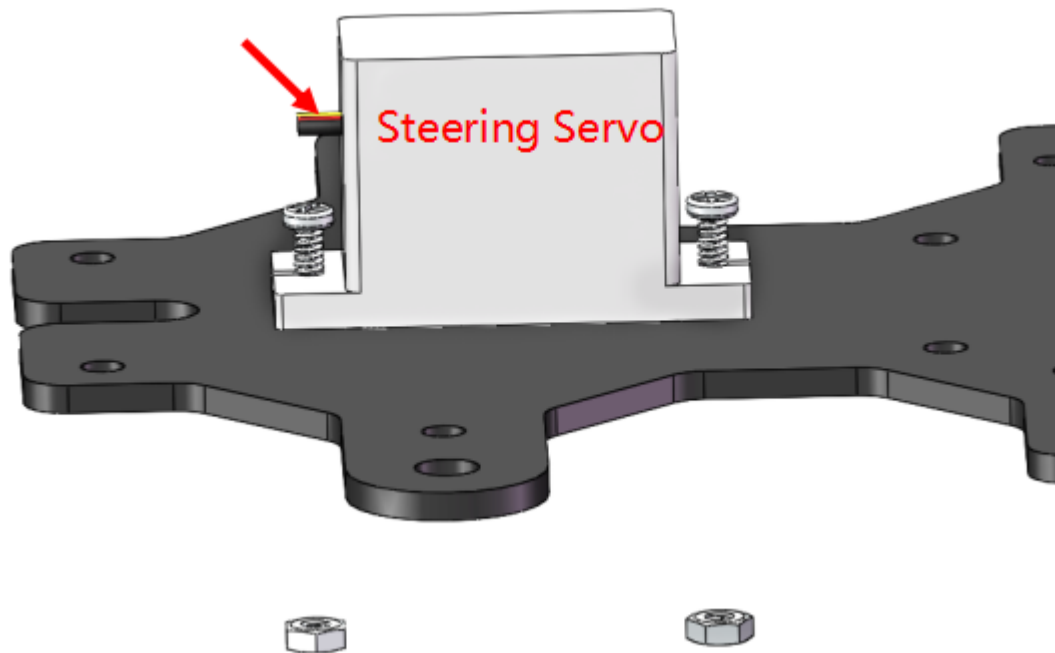
上記の状況のいずれかが発生した場合は、service@sunfounder.com <mailto:support@sunfounder.com>`\_\_に送信してください。新品に交換してあげる。使用または組み立ての過程で破損した場合は、公式ウェブサイト`www.sunfounder.com にて購入してください。

1.7 組み立てを続ける

警告： バッテリーを取り付けて電源スイッチを ON にスライドさせてから、組み立てのプロセス全体でサーボインストールを実行し続けてください。

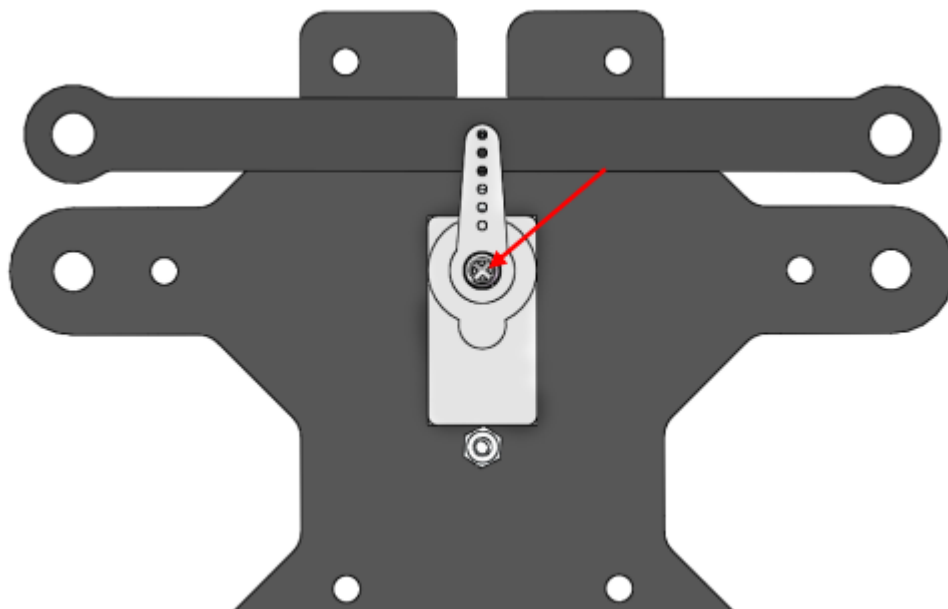
1.7.1 ステアリングサーボの組み立て

2つの **M2x8 ネジ** と **M2 ナット** を使ってステアリングサーボを上部プレートに取り付ける（サーボワイヤーの方向に注意してください）。

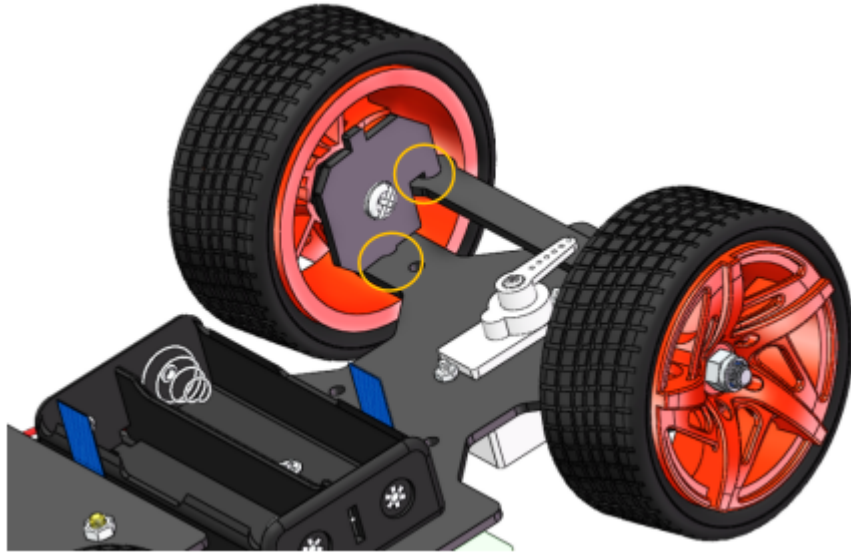


1.7.2 フロントハーフシャシ

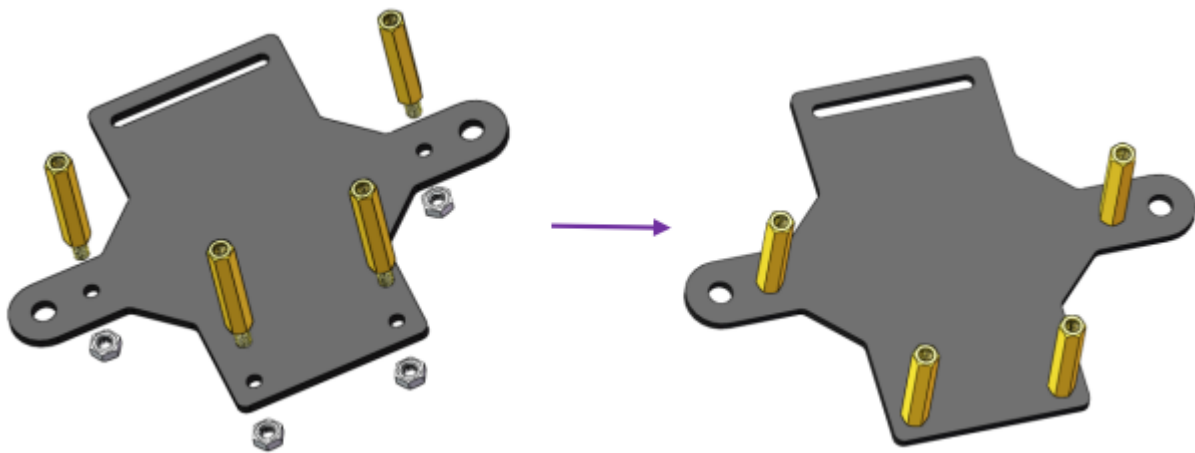
ロッカーアーム固定ネジ**（最短）で**ステアリングリンケージをロッカーアームに接続する。



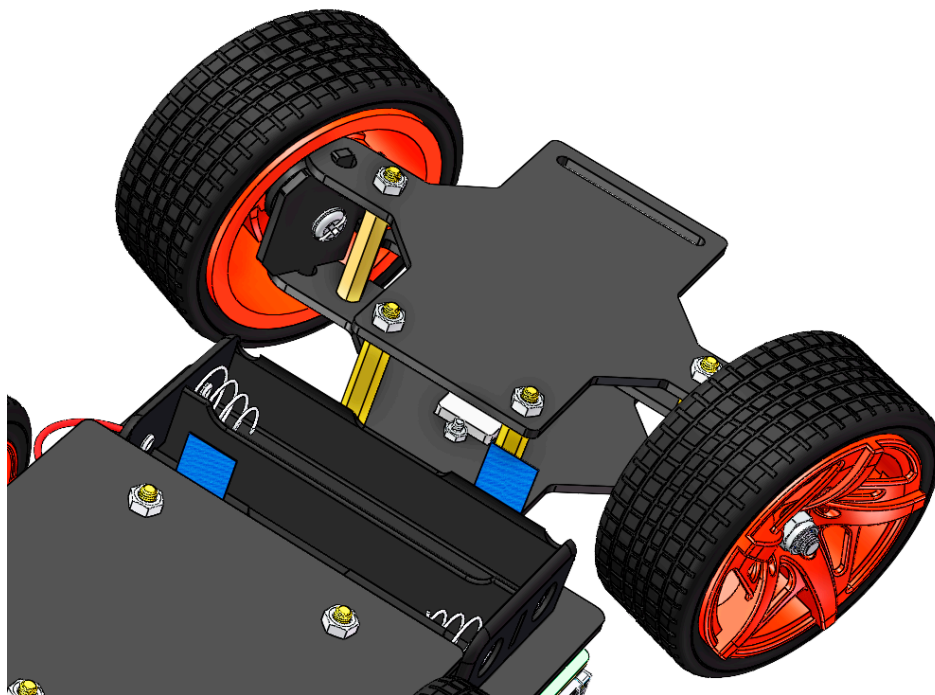
ホイールを上部プレートに入念に取り付ける。



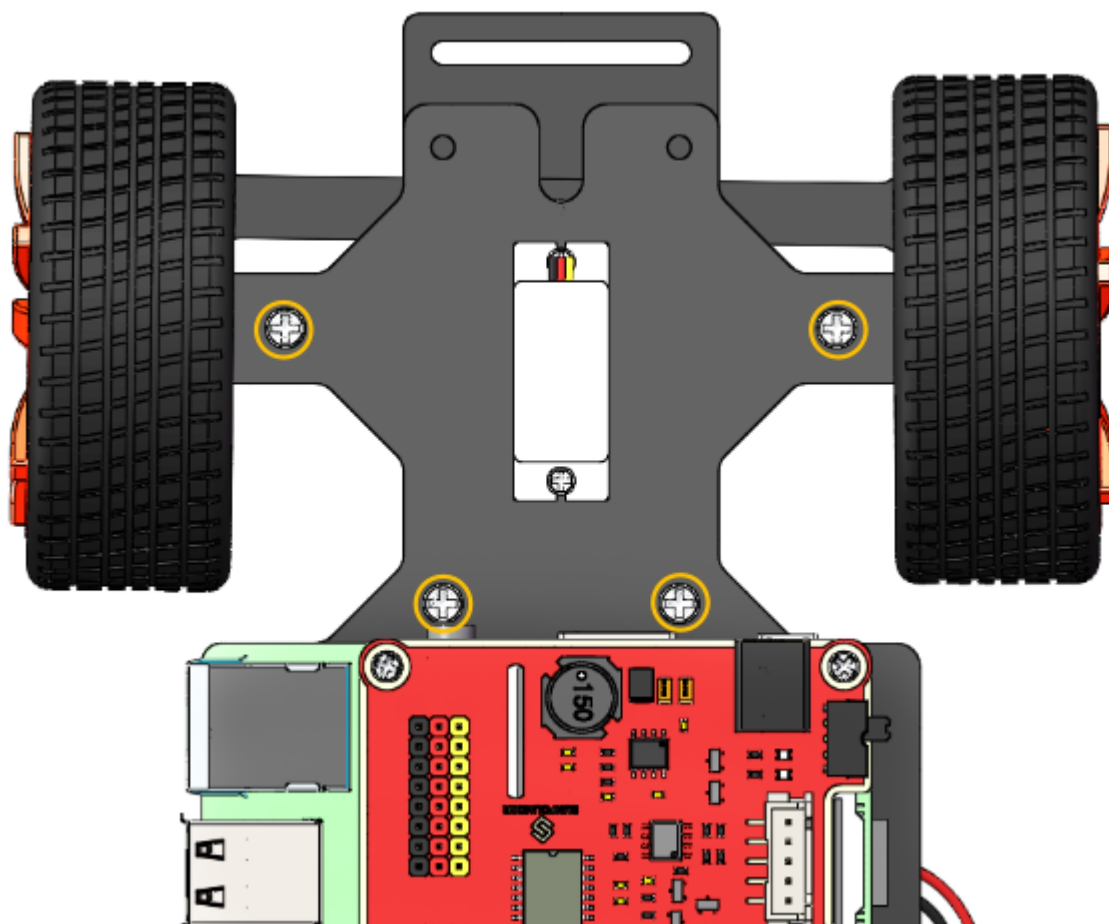
4つの M3x25 銅製スタンドオフと M3 ナットでフロントハーフシャーシを組み立てる。



次に、組み立てられたフロントハーフシャーシを、スタンドオフを穴に合わせている上部プレートに置く。

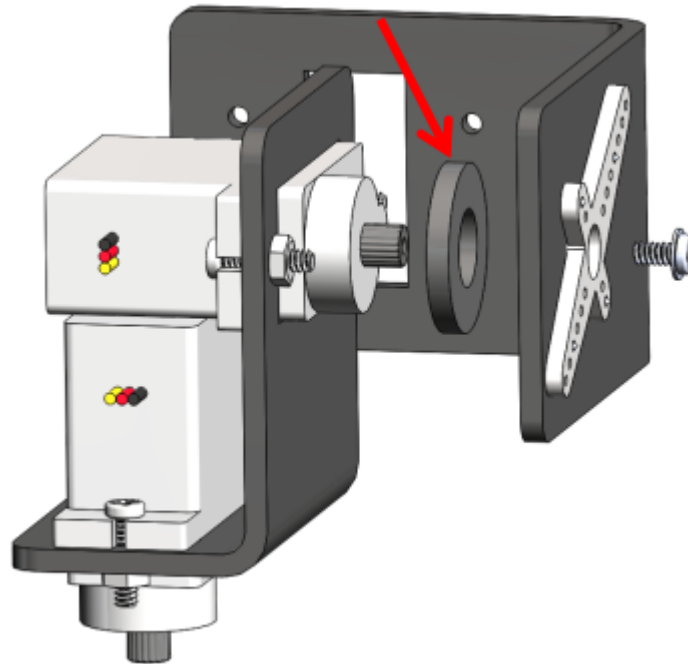


入念に持ち、上下を逆にして、4本の **M3x8 ネジ** でスタンドオフと上部プレートを固定する。

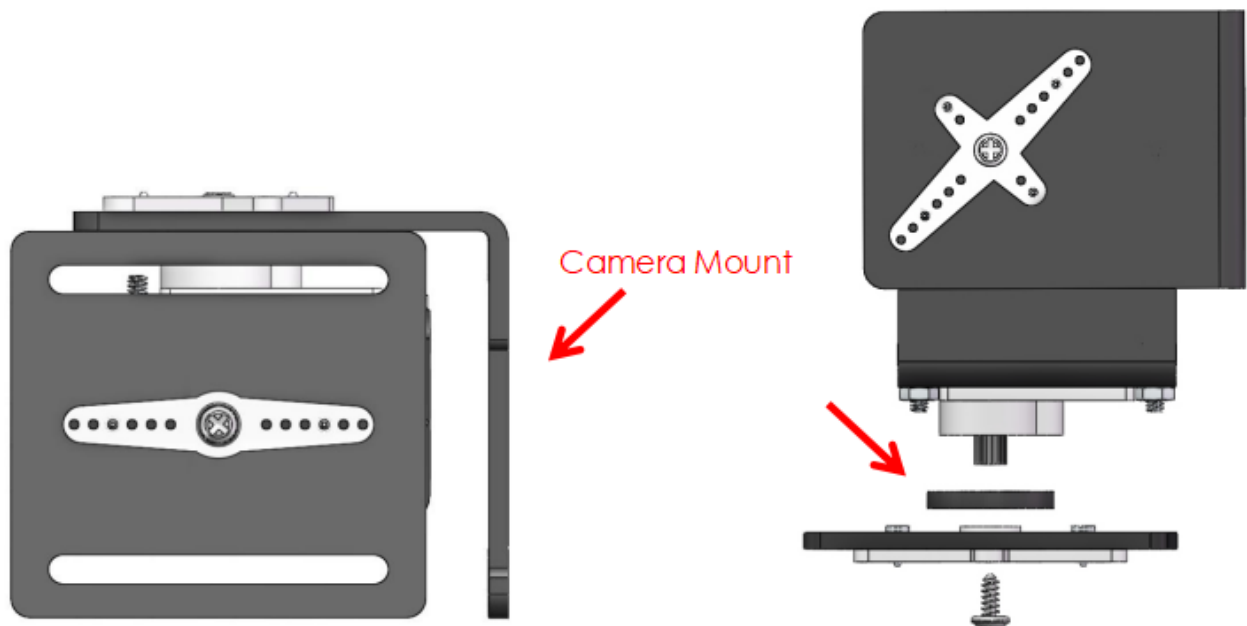


以下に示すように、組み立てられた **パン&チルトプレート**、**ガスケットプレート** と **カメラマウントプレート** を **ロッカーアーム固定ネジ** で固定角度で組み立てる:

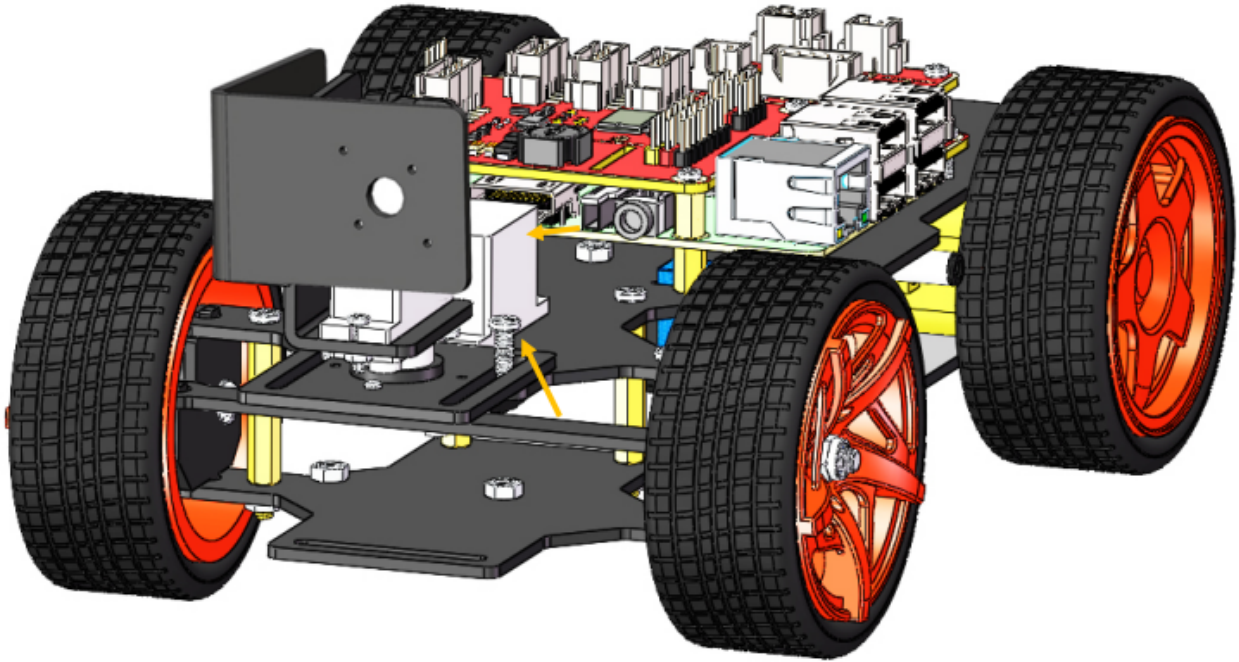
注釈: サーボが故障した場合、手でサーボを回さないでください。



同様に、サーボシャフトが 90 度に回転していることを確認する。次に、以下に示す角度で、**パン&チルトベースプレート** を **ロッカーアーム固定ネジ** と **ガスケットプレート** で組み立てる。



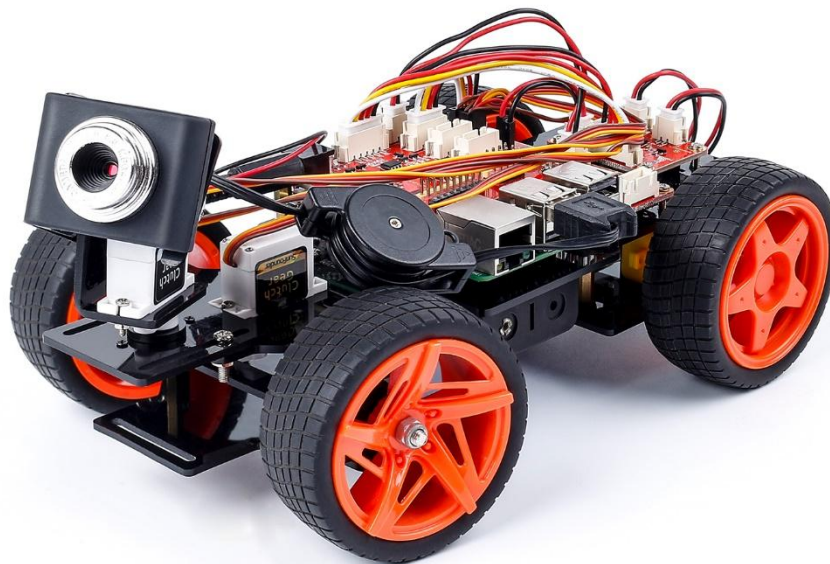
2つの M3x10 ネジと M3 ナットを使用して、パン&チルトプレート を車に取り付ける。



1.7.3 カメラを組み立てる

カメラを取り出し、カメラマウントに嵌め込む。USB ケーブルを Raspberry Pi の USB ポートに接続する。

これで、アセンブリ全体が完了した。おめでとう！今すぐ車の電源を切ることができる！バッテリーを充電することを忘れないでください。



1.8 旅に出よう！

すべての手順が完了していて、機械的なアセンブリとソフトウェアのインストールに問題がないことを確認してください。

1.8.1 遠隔操作

これはこれから行うことである：

Raspberry Pi を **server** として使う。車を制御し、カメラでキャプチャした画像を送信するための API を使用して Web サーバーを実行する。

次に、PC、携帯電話、タブレットを **client** として、カメラから画像を取得し、Web サーバーの API を呼び出して車を制御します。

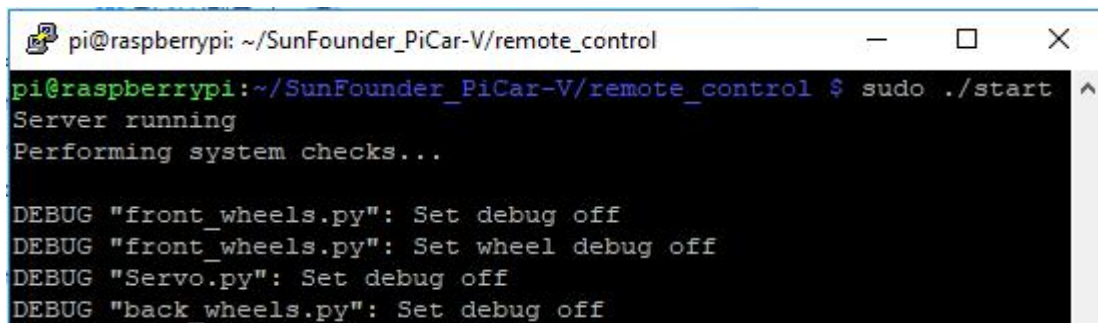
その後、再び車の電源を入れてください。最初のテストに時間がかかるため、Raspberry Pi の電源アダプターを使用して車に電源を供給することも勧める。

サーバーを実行する（Raspberry Pi での操作）

Raspberry Pi にリモートでログインする。**remote_control** ディレクトリの下で起動スクリプト **start** を実行して、Web サービスを開始する。

```
cd ~/SunFounder_PiCar-V/remote_control
python3 manage.py migrate
sudo ./start
```

スクリプトがサービスを有効にし、対応するデータが表示される。同時にハードウェアが初期化されるため、前輪に接続されているサーボとパン&チルトが回転し、ハードウェアの初期化が完了したことを示す。



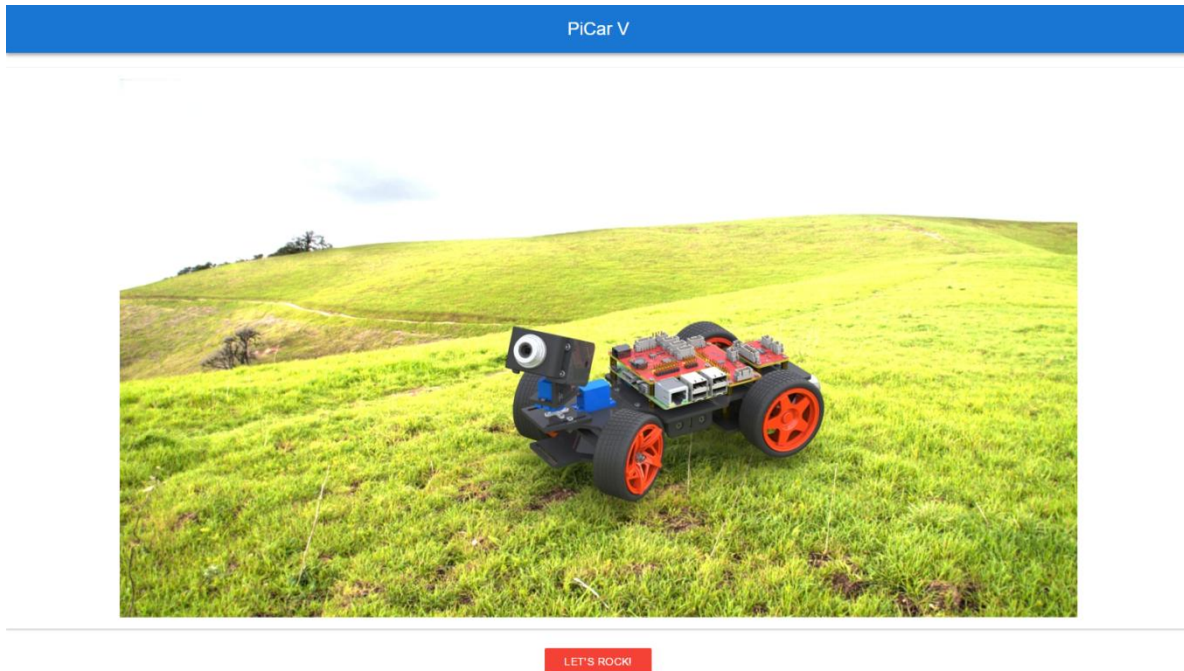
```
pi@raspberrypi: ~/SunFounder_PiCar-V/remote_control
pi@raspberrypi:~/SunFounder_PiCar-V/remote_control $ sudo ./start
Server running
Performing system checks...
DEBUG "front_wheels.py": Set debug off
DEBUG "front_wheels.py": Set wheel debug off
DEBUG "Servo.py": Set debug off
DEBUG "back_wheels.py": Set debug off
```

上記のような結果が得られれば、サーバーは準備できた。次に、クライアントを起動する。

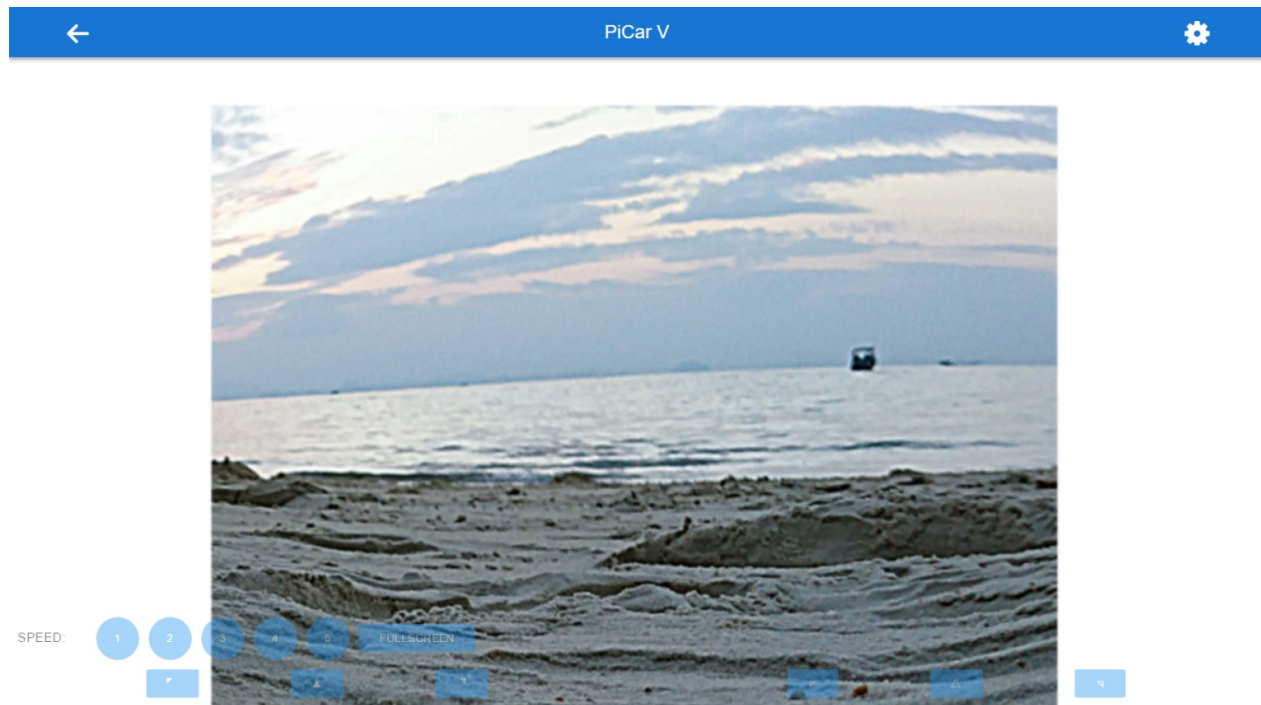
警告: クライアントの実行を停止するまで、サーバーをいつも動作させる。

クライアントの実行 (PC での操作)

http://<RPi_IP_address>:8000/ で車のサーバに訪問する。ウェルカムページが表示される：



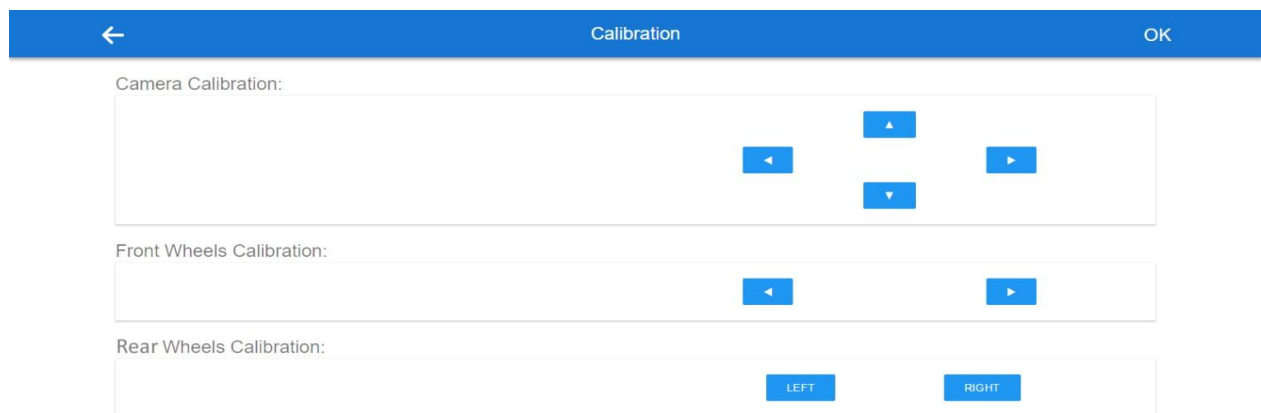
LET'S ROCK をクリックして、操作インターフェイスに入る。



このページでは、キーボードの **W**、**A**、**S**、**D** キーを押して、車を****前方****、**後方**、**左折**、**右折** に制御したり、矢印キーを押してカメラの移動を制御したりすることができ、数値 **1~5** は速度レベルを変更する。

補正

もう一度 **FULLSCREEN** をクリックして、タイトルバーを表示する。次に、ページの右上隅にある設定ボタンをタップして、補正ページに進む。



補正には、**カメラ補正**、**前輪補正**、**後輪補正** の3つの部分が含まれている。

このページに入ると、車が前進する。そうでない場合は、**後輪補正** の **左****と ****右** をクリックして、ホイールの角度を調整する。

注釈: ボタンを押すたびに角度が少しずつ変化する。時間あたりの変化を大きくするには、ボタンを長押ししてく

ださい。

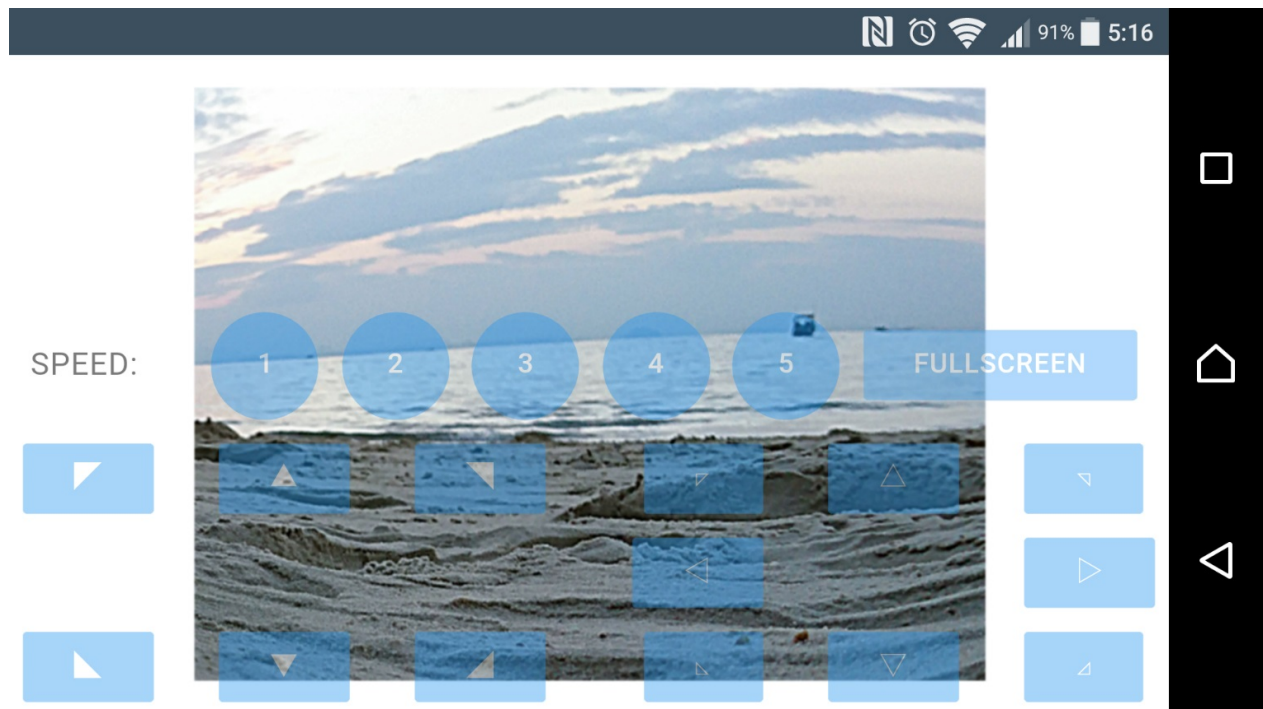
すべての補正が完了したら、**OK** をクリックして結果を保存する。

携帯電話用

また、携帯電話の場合は、**FULLSCREEN** ボタンをタップすると、表示とパフォーマンスが向上する。次に、ページ上の 5 つの速度レベルのボタンをタップして速度を制御し、矢印ボタンを押して車の方向とパン&チルトを制御する。ただし、**一度** にタブできるのは 1 つのタッチポイントのみである。



Android 操作システムのスマートフォンから次のスクリーンショットを取得する。



見た目は PC ほど良くないかもしれないが、以降の更新は **Github** で不定期にリリースされる可能性がある。リポジトリを使い、変更を加えたプルリクエストを送信してください。テスト後に問題がなければ、リクエストを知ることを楽しむである。

1.8.2 ボール_追跡装置

キーボードを使用して車の状態を制御するだけでなく、ボールトラッキングコードも作成した。

Install OpenCV

Before installing OpenCV, you need to install the dependencies first. Different operating systems require different dependent libraries. (If you don't know your operating system version, you can try to install both.)

- If your operating system is **RASPBERRY PI BUSTER**, you need to install the following dependencies.

```
sudo apt-get install libatlas-base-dev libjasper-dev libqt4-test libwebp6 libtiff5
↪libopenexr23 libgstreamer1.0-0 libavcodec-dev libavformat-dev libswscale-dev libqtgui4
↪-y
```

- If your operating system is **RASPBERRY PI BULLSEYE**, you need to install the following dependencies.

```
sudo apt-get install libatlas-base-dev libjasper-dev libwebp6 libtiff5 libopenexr25
↪libgstreamer1.0-0 libavcodec-dev libavformat-dev libswscale-dev -y
```

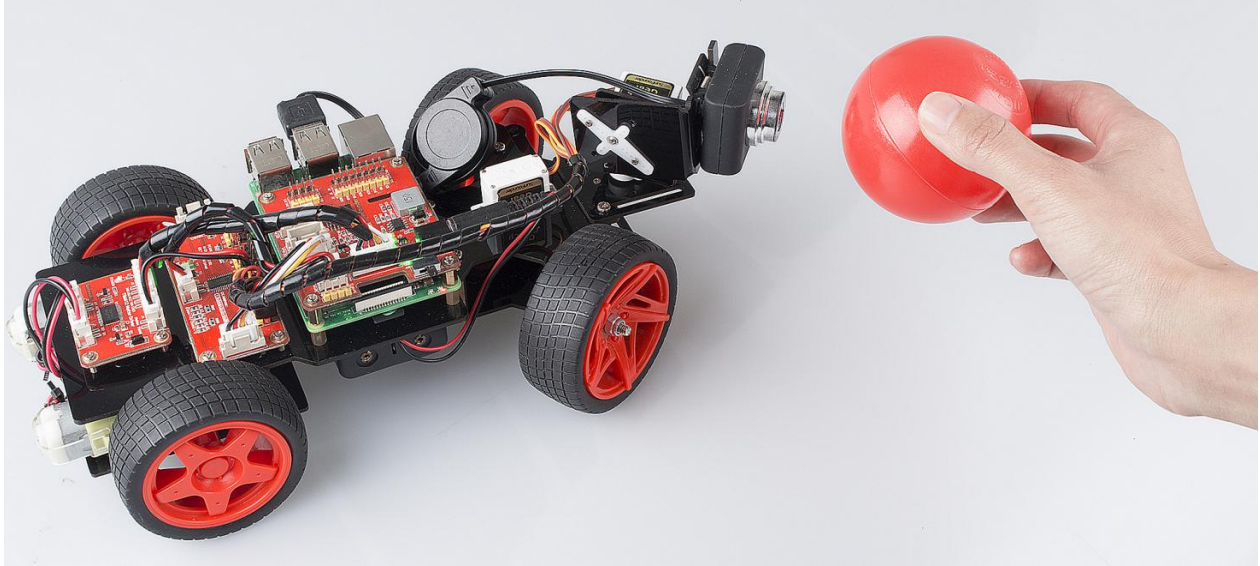
Next is to install OpenCV.


```
sudo pip3 install opencv-python==3.4.6.27
```

Run the Code

```
cd ~/SunFounder_PiCar-V/ball_track python3 ball_tracker.py
```

コードが実行されたら、赤いボール（少なくともカメラの直径よりも大きい）を見つけ、カメラの前の 10~50cm の範囲内に配置して、車があなたのボールを追跡するようにする。

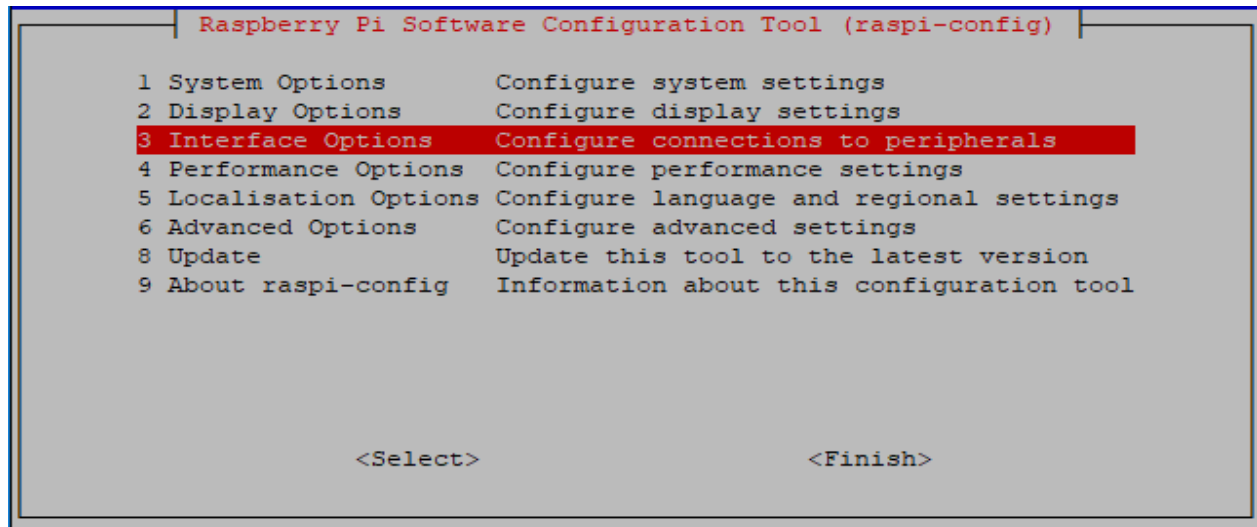


車があなたのボールを捕まえたかどうかを知りたい場合は、以下の手順を実行してください：

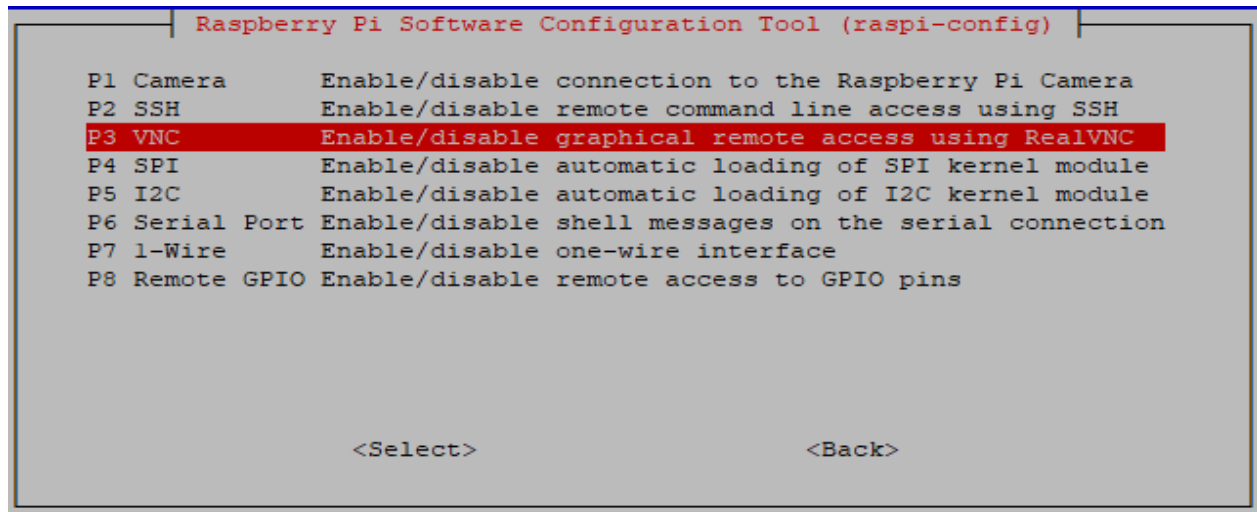
ステップ 1： Raspberry Pi で VNC を有効にする。

```
sudo raspi-config
```

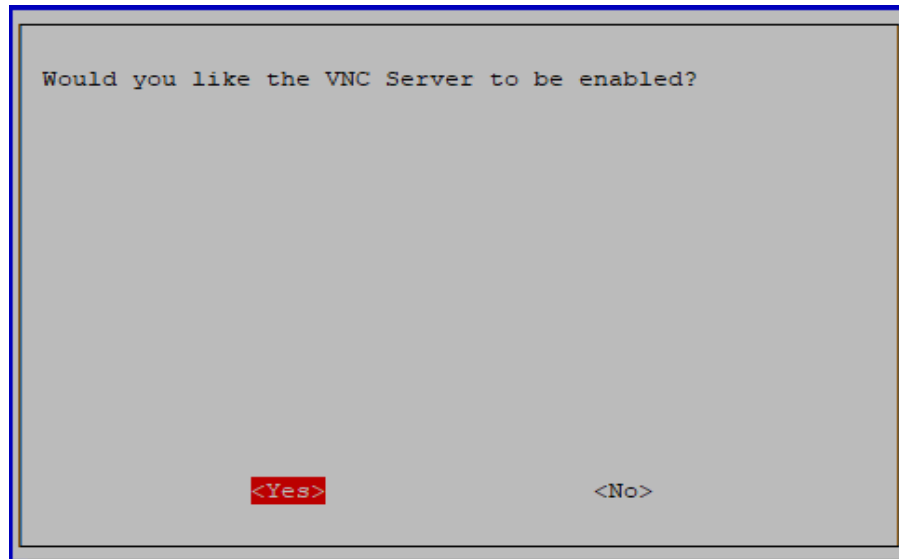
3 Interfacing Options を選択する。**Enter** キーを押してください。



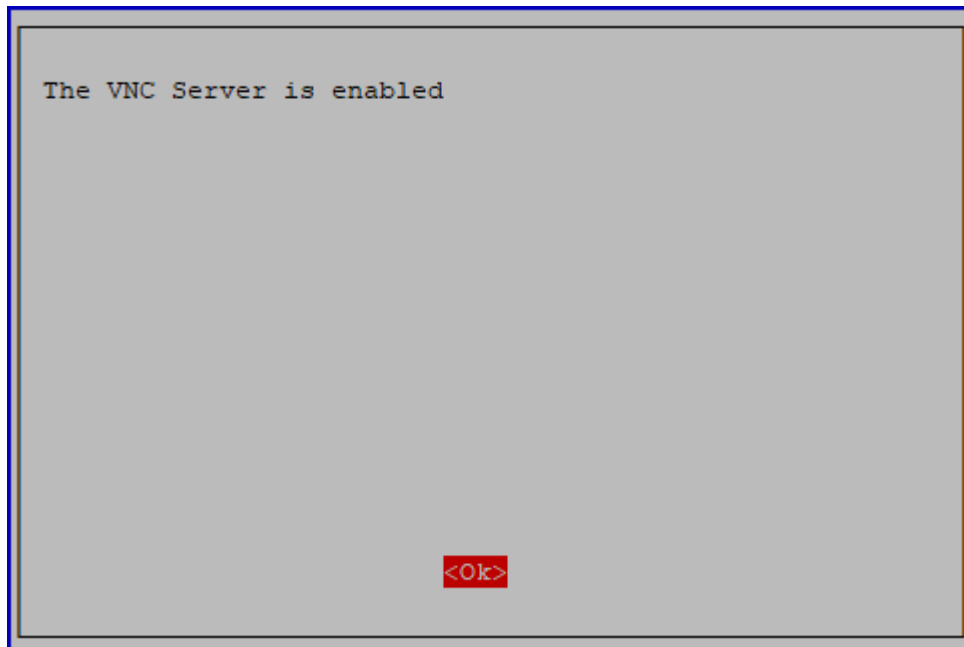
P3 VNC



キーボードの左矢印キーを押して、**はい**を選択します。



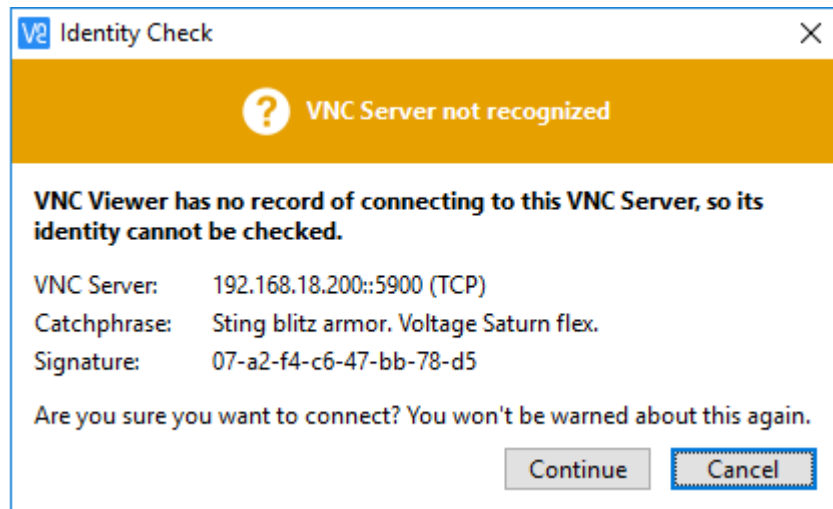
最後に **OK -> Finish** を選択して、構成を終了する。



ステップ 2: お使いのコンピューター用の VNC ビューアーをダウンロードしてインストールする。

<https://www.realvnc.com/en/connect/download/viewer/>

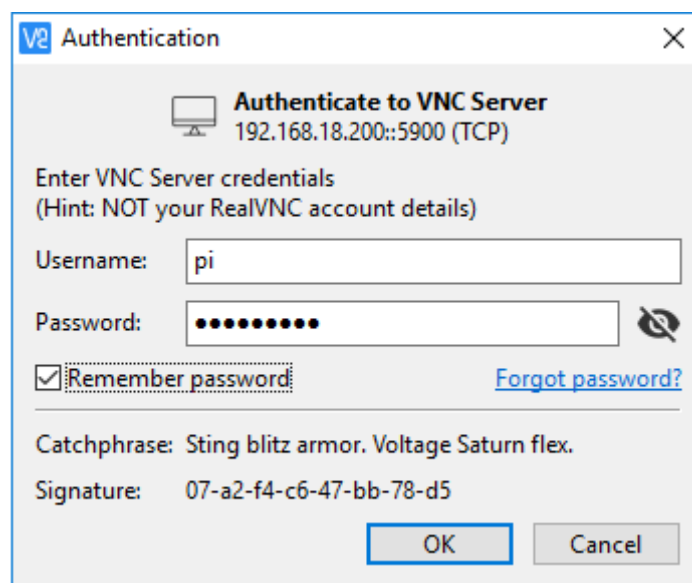
注釈: Windows への実装中に次のプロンプトが表示された場合は、**Continue** をクリックしてインストールを続行してください。



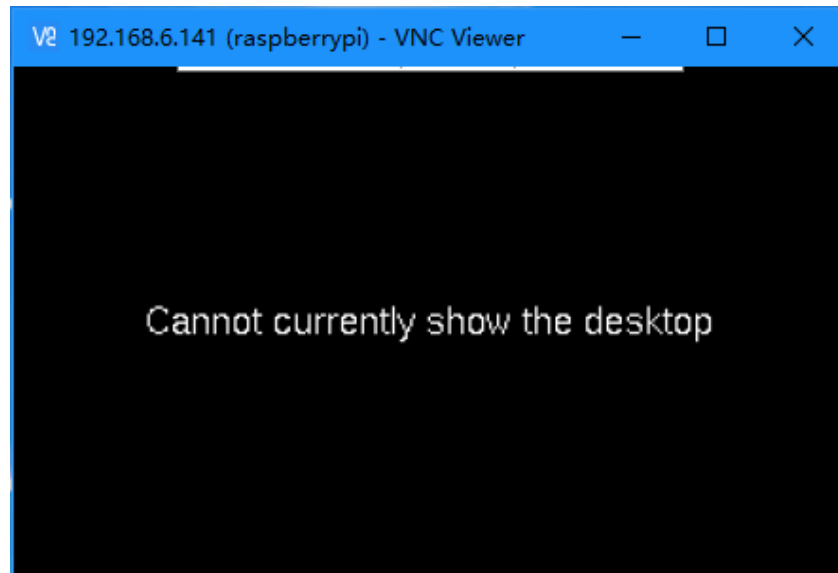
ステップ 3: インストールした VNC ビューアーを開き、Raspberry Pi の IP アドレスを入力して、Enter ボタンを押す。



ステップ 4: このページで Raspberry Pi のユーザー名とパスワードを入力し、デフォルトでは、別々は pi と raspberry である。次に OK をクリックする。



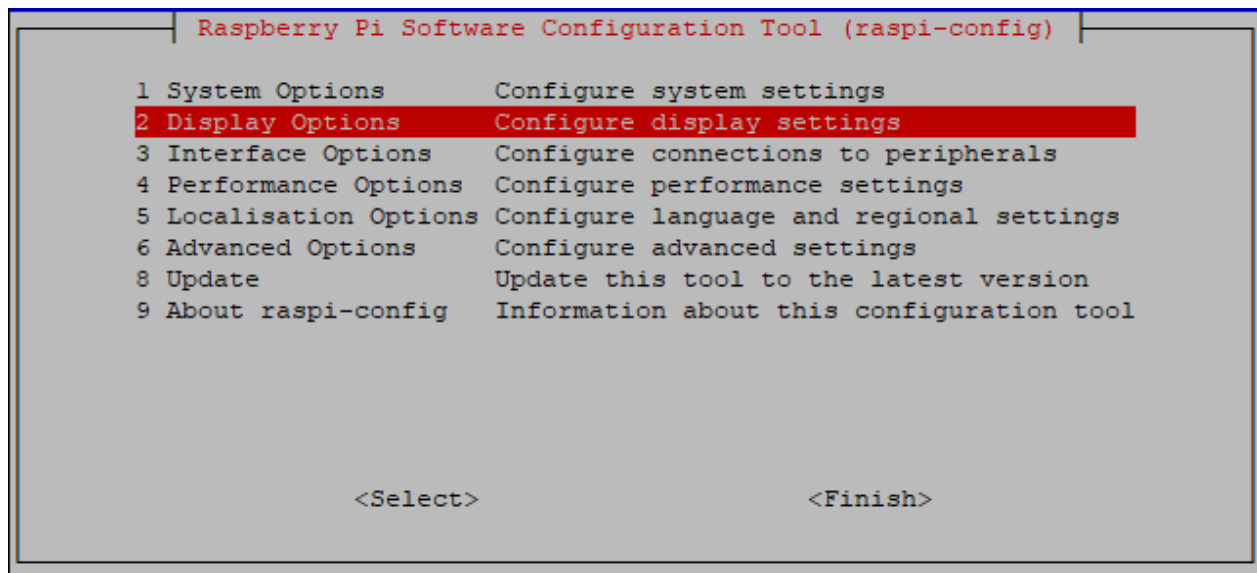
VNC Viewer でログインすると、画面が真っ暗になる場合があります。画面が正常に表示されている場合は、ステップ 5 に進みます。



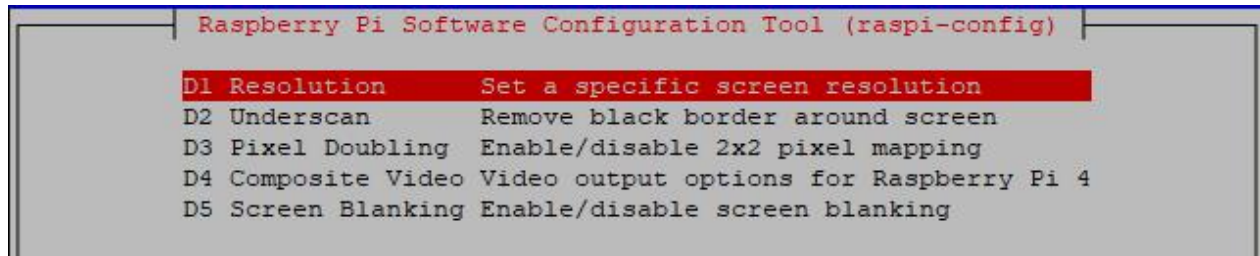
RaspberryPi の解像度を最高に変更します。

```
sudo raspi-config
```

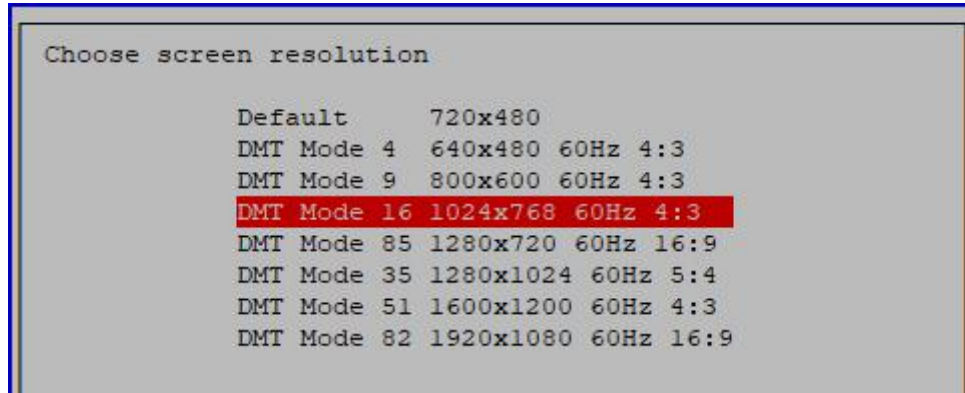
選択 **2 Display Options**。



選択 **D1 Resolution**。



デフォルトをモード 16 のような高解像度に変更します。次に **OK** をクリックする。



Raspberry Pi を再起動し、再度ログインすると、機能します。

```
sudo reboot
```

ステップ 5: 次に、Raspberry Pi デスクトップに入り、ターミナルをクリックし、次のコマンドを入力して、コード `ball_tracker.py` を開く。

```
cd ~/SunFounder_PiCar-V/ball_track  
sudo nano ball_tracker.py
```

次に、コードを次のように修正する：

```
Show_image_enable = True  
Draw_circle_enble = True
```

Ctrl+X と Y を押して変更を保存する。

```

192.168.18.200 (raspberrypi) - VNC Viewer
pi@raspberrypi: ~/SunFounder_PiCar-V/ball_track
File Edit Tabs Help
GNU nano 3.2 ball_tracker.py Modified
from picar import front_wheels, back_wheels
from picar.SunFounder_PCA9685 import Servo
import picar
from time import sleep
import cv2
import numpy as np
import picar
import os

picar.setup()
# Show image captured by camera, True to turn on, youwill need #DISPLAY and it$
show_image_enable = True
draw_circle_enable = True
scan_enable = False

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell

```

注釈: この手順を完了したら、VNC 経由で Raspberry Pi にログインするか、またはモニターを使用してください。そうでない場合、次の警告が出される：

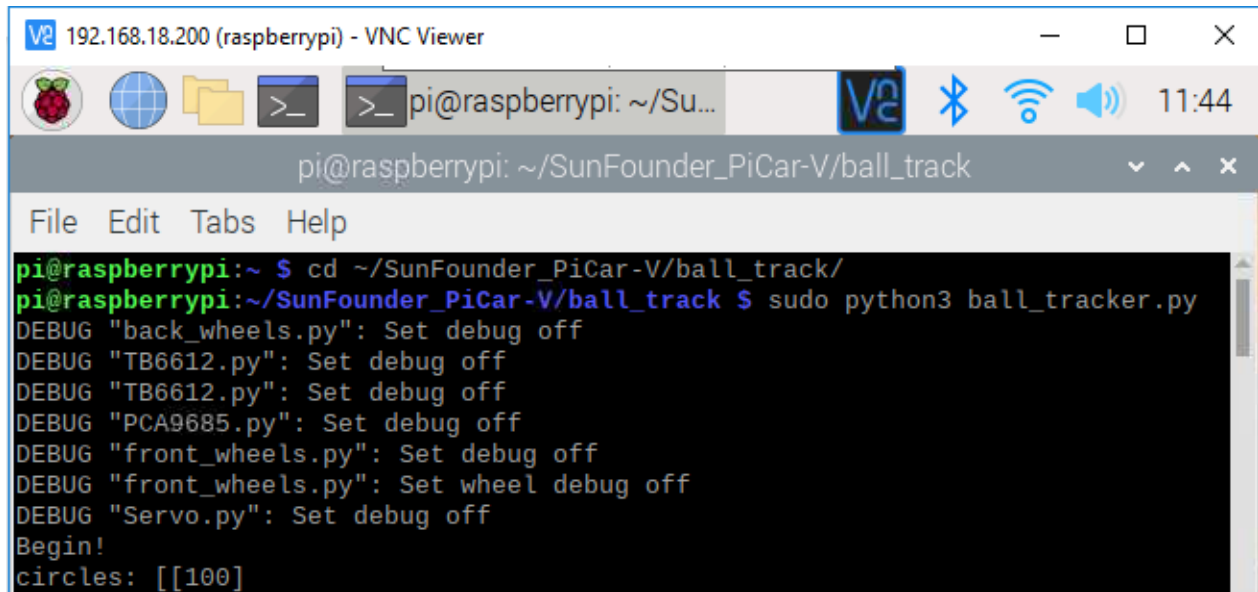
```

pi@raspberrypi:~/SunFounder_PiCar-V/ball_track $ python3 ball_tracker.py
Warning: Display not found, turn off "show_image_enable" and "draw_circle_enable"
DEBUG "back_wheels.py": Set debug off

```

ステップ 6: ball_tracker.py を実行する。

```
python3 ball_tracker.py
```

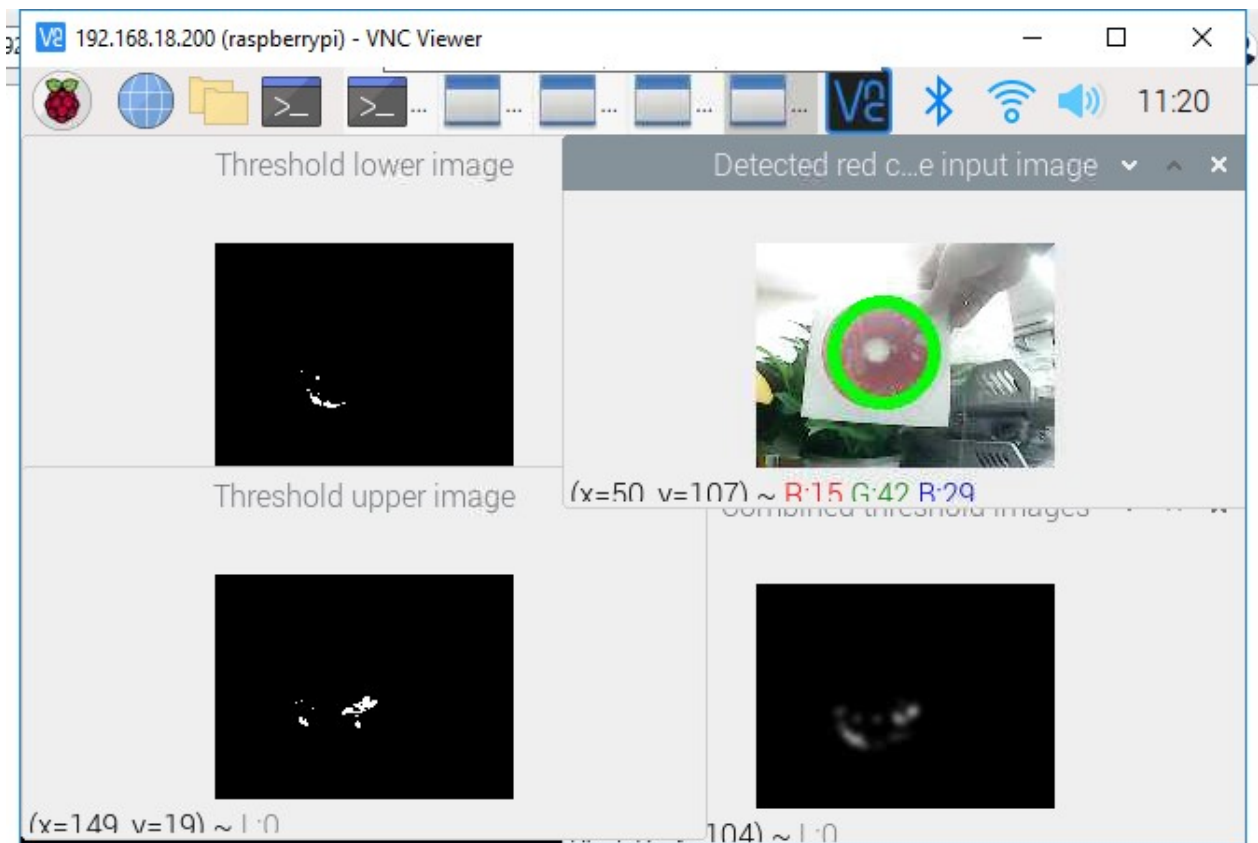


```

V2 192.168.18.200 (raspberrypi) - VNC Viewer
pi@raspberrypi: ~/SunFounder_PiCar-V/ball_track
File Edit Tabs Help
pi@raspberrypi:~ $ cd ~/SunFounder_PiCar-V/ball_track/
pi@raspberrypi:~/SunFounder_PiCar-V/ball_track $ sudo python3 ball_tracker.py
DEBUG "back_wheels.py": Set debug off
DEBUG "TB6612.py": Set debug off
DEBUG "TB6612.py": Set debug off
DEBUG "PCA9685.py": Set debug off
DEBUG "front_wheels.py": Set debug off
DEBUG "front_wheels.py": Set wheel debug off
DEBUG "Servo.py": Set debug off
Begin!
circles: [[100]]

```

ステップ 7: コード `ball_tracker.py` を実行すると、これらの 4 つの画像が表示される。カメラの前に赤いボールを置くと、車が赤いボールを捕まえたことを示す緑の円が表示される。

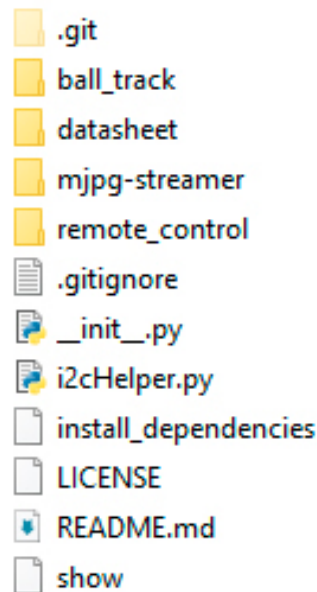


1.8.3 ファイル分析

/home/pi に 2 つのフォルダーがある：モーターとステアリングを制御するための **SunFounder_PiCar** と、パン/チルトとワイヤレス制御のための **SunFounder_PiCar-V** ここで PiCar-V のコードを調べる。

コードに含まれる内容と指示が多すぎるため、すべての詳細を説明しない。プログラムの Python 2、Python 3、Django、HTML、CSS の詳細については、関連する Web サイトにアクセスするか、または本を購入して自分で学習してください。次のパートでは、全体的な構造とプロセスについて簡単に説明する。

まず、コードフォルダー内のファイルを確認する：



- **.git** は Git リポジトリのすべての情報を確認するための隠しディレクトリで、リポジトリを作成すると、自動的に生成される。
- **Ball_track** は車に赤いボールを追跡させるために使用される。
- **データシート** はハードウェアモジュールで 사용되는一部のチップのマニュアルを保存する。
- **mjpg-streamer** はカメラのデータが転送されるオープンソースのストリーミングメディアライブラリである。
- **Remote_control** は Web サーバーのコードを保存するためのもので、API リクエストに基づいて車のモーターとサーボを制御する。
- **.gitignore** は Github リポジトリが同期されるときに無視されるファイルタイプのリクエストを記録する。
- Python パッケージの標準の必要なドキュメントであるプロジェクトを作成すると、**__init__.py** は自動的に生成されるため、そのままにしておいてください。
- **i2cHelper.py** は I2C 接続を構成したり検出したりするために Python 2 によって作成された Python スクリプト。

リプトである。

- `install_dependencies`, 簡単なインストールと環境設定のための実行可能な `bash` スクリプト。
- その名のとおり、`LICENSE` は GNU V2 ライセンスのテキストファイルである。
- `README.md` と `show` はステートメントとプロンプトに関する通常の情報を記録する。

サーバーコード

サーバーコードは Django 1.10 に基づいている（必要に応じて最新のリリースに適応する）。これに興味がある場合は、Django の Web サイト <https://www.djangoproject.com/> にアクセスするか。ここでは、Web サーバーがどのように機能するかを学習する。

注釈: バグを修正し、一部の関数の更新をリリースするために、Github でコードが不定期に更新される場合がある。したがって、このコードは初期バージョンである。以下の場合、Github リポジトリの更新を表示できる：

https://github.com/sunfounder/SunFounder_PiCar-V/tree/V3.0

コードディレクトリを開き、`ls` でファイルを確認します。

```
pi@raspberrypi:~ $ cd SunFounder_PiCar-V/
pi@raspberrypi:~/SunFounder_PiCar-V $ ls
datasheet  install_dependencies  mjpg-streamer  remote_control
client      i2cHelper.py          LICENSE         README.md show
```

`remote_control` は、Web サーバーのメインコードディレクトリである。`cd remote_control` でコードディレクトリを開き、`ls` によってファイルを確認する：

```
pi@raspberrypi:~/SunFounder_PiCar-V $ cd remote_control
pi@raspberrypi:~/SunFounder_PiCar-V/remote_control $ ls
db.sqlite3  manage.py  remote_control  start  static
```

- Django プロジェクトを作成すると、`db.sqlite3` が生成される。
- `manage.py` は Django プロジェクトのメインプログラムであり、プロジェクトの作成時に生成される。通常は変更する必要はない。
- `remote_control` には、メインコードファイルが含まれている。
- `start` は `sudo python manage.py runserver 0.0.0.0:8000` を実行するために書かれた小さなスクリプトであり、便宜上、属性のあるインストールを使用したサーボ **インストール** である。
- `static` はいくつかの静的な画像を Web に保存するものである。

Django ウェブサーバーは通常、`sudo python manage.py runserver ``` の実行を通じて起動する。アドレス `0.0.0.0:8000` は、リスニングアドレスが LAN 上のすべてのアドレスをカバーし、ポート番号が 8000 であることを意味する。ここでは、```remote_control` フォルダのコードに焦点を当てる。`cd remote_control` を使用してディレクトリに入る：

```
pi@raspberrypi:~/SunFounder_PiCar-V/remote_control $ cd remote_control
pi@raspberrypi:~/SunFounder_PiCar-V/remote_control/remote_control $ ls
driver  __init__.py  settings.py  templates  urls.py  views.py  wsgi.py
```

- `driver` は車のドライバーを保存する。
- `__init__.py` は Python パッケージの標準的な必要なドキュメントである Django プロジェクトを作成すると自動的に生成される。そのままにしておいてください。
- `settings.py` は自動的に生成され、関連する設定を保存する。
- `templates` HTML 形式でウェブを格納するためのジャンゴアプリです。
- `urls.py` は自動的に生成され、コードに関連付ける URL を構成する。
- `views.py` は URL によって関連付けられるページコントロールのコードである。テンプレートを呼び出してページを表示し、ドライバーが車を制御する。
- `wsgi.py` は自動的に生成され、変更する必要はない。詳しくは Django の公式サイトをご覧ください。

これがコードの動作原理である。`urls.py` に自動的に関連付けられるメインプログラム `manage.py` を実行して、URL に応答する。Chrome のような Web ブラウザを実行して `http://<rpi_ip_address>:<port> ``` または、**クライアントを介して構成済みの API にアクセスすると、`urls.py` の関連付けにより、`manage.py` が `views.py` に変わる。**次に、`views.py` はこれを処理し、テンプレートをブラウザに返す。さらに、ブラウザで設定されたパラメーターに基づいて `driver` パッケージを呼び出し、車を制御する。

ここで、フォルダー `driver` を開き、以下を確認する：

```
pi@raspberrypi:~/SunFounder_PiCar-V/remote_control/remote_control $ cd driver/
pi@raspberrypi:~/SunFounder_PiCar-V/remote_control/remote_control/driver $ ls
camera.py  config  __init__.py  stream.py
```

`driver` フォルダーには、主にパン&チルトを制御するためのドライバーモジュールとカメラストリーマーが含まれている。

パン&チルトを制御するための `camera.py`。

`config` は補正データを保存するために使用される。

`__init__.py` はパッケージの必須ファイルであり、そのままにしておいて下さい。

`stream.py` は MJPG-streamer に基づくビデオストリーミングサービスである。

表示するフォルダ `templates` を終了して開くと、以下が表示される：

```
pi@raspberrypi:~/SunFounder_PiCar-V/remote_control/remote_control/driver $ cd ../
pi@raspberrypi:~/SunFounder_PiCar-V/remote_control/remote_control $ cd templates/
admin.py  \__init__.py  models.py  tests.py
apps.py   migrations  templates  views.py
```

このフォルダーは、テンプレートを簡単に呼び出すために、`manage.py startapp` によって作成される。したがって、`templates` 以外のファイルは変更されていない。`templates` をもう一度開く：

```
pi@raspberrypi:~/SunFounder_PiCar-V/remote_control/remote_control/templates $ cd templates
pi@raspberrypi:~/SunFounder_PiCar-V/remote_control/remote_control/templates/templates $ ls
base.html  cali.html  run.html
```

レイヤーで構成される 3 つの HTML ファイルがある。低レベルの `base.html` には、デフォルトでは、HTML の `<head>`、**レイアウト全体**、と**ホームページのコンテンツが含まれている**。表層：補正用の `cali.html` と、車を制御するための `run.html`。

1.8.4 高級段階

今、車は走行している！それを制御するか、または Windows、Linux、または Mac OS X でキャプチャされた画像を表示してみてください。

それでは次に何をすべきか？

はい、センサー！Robot HATS には 8 つのデジタルチャンネルと 4 つのアナログチャンネルと 2 つの I2C ポートがある。さらに、車のヘッドの亚克力板にある 3mm のスロットにより、さまざまなセンサーの接続が可能になり、車がよりスマートになり、興味をそそられるようになる！

そして、コーディング！プログラミングに関する知識を身につけて、コードをコンパイルできる場合は、キット用に提供されているコードを変更して、車をよりすばらしいものにすることができる。たとえば、前輪と後輪を制御するコードは、`SunFounder_PiCar` フォルダで見つけて `install-dependences` によってプログラムをインストールした場合は、`/home/pi` ディレクトリにダウンロードされる。

ファイルの内容を確認してください：

```
cd ~/SunFounder_PiCar
ls
```

コードを変更したら、車の Github リポジトリのページに入り、フォークして改善に役立つだろう。また、問題を投稿して、Github ページでプルをリクエストすることもできる：

PiCar-V: https://github.com/sunfounder/SunFounder_PiCar-V/tree/V3.0

PiCar ドライバー：https://github.com/sunfounder/SunFounder_PiCar

もちろん、質問やアイデアがある場合は、service@sunfounder.com までお気軽に送信してください。

1.9 付録

1.9.1 付録 1：サーバーインストールスクリプトの機能

質問があるかもしれない：Raspberry Pi にサーバーをインストールすると、インストールスクリプトはどのように機能するか？ここで、詳細な手順を説明しよう。

1. pip をインストールする。

```
sudo apt-get install python-pip
```

2. pip を使って django をインストールする。

```
sudo pip install django
```

3. i2c-tools と python-smbus をインストールする。

```
sudo pip3 install smbus2
```

4. PiCar ドライバーをインストールする。

```
cd ~/
git clone --recursive https://github.com/sunfounder/SunFounder_PiCar.git
cd SunFounder_PiCar
sudo python setup.py install
```

5. ソースコードをダウンロードしてください。

```
cd ~/
git clone https://github.com/sunfounder/SunFounder_PiCar-V -b V3.0
```

6. MJPG-Streamer ファイルをシステムディレクトリにコピーする。

```
cd ~/SunFounder_PiCar-V
sudo cp mjpg-streamer/mjpg_streamer /usr/local/bin
sudo cp mjpg-streamer/output_http.so /usr/local/lib/
sudo cp mjpg-streamer/input_file.so /usr/local/lib/
```

(次のページに続く)

(前のページからの続き)

```
sudo cp mjpg-streamer/input_uvc.so /usr/local/lib/  
sudo cp -R mjpg-streamer/www /usr/local/www
```

7. パスをエクスポートする。

```
export LD_LIBRARY_PATH=/usr/local/lib/ >> ~/.bashrc  
export LD_LIBRARY_PATH=/usr/local/lib/ >> ~/.profile  
source ~/.bashrc
```

8. I2C1 を有効にする。

/boot/config.txt ファイルを編集する：

```
sudo nano /boot/config.txt
```

最後に行を追加する：

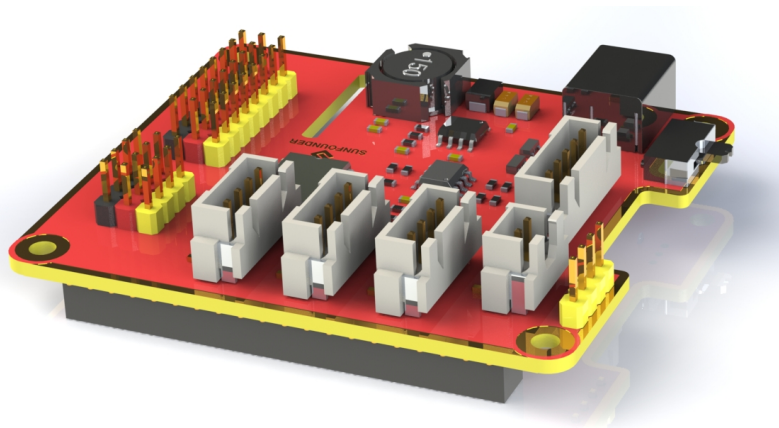
```
dtoverlay=i2c-arms
```

9. 再起動する。

```
sudo reboot
```

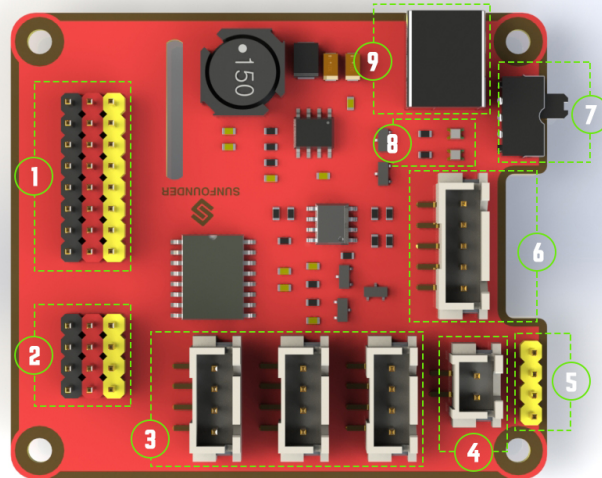
1.9.2 付録 2：部品

ロボット HATS



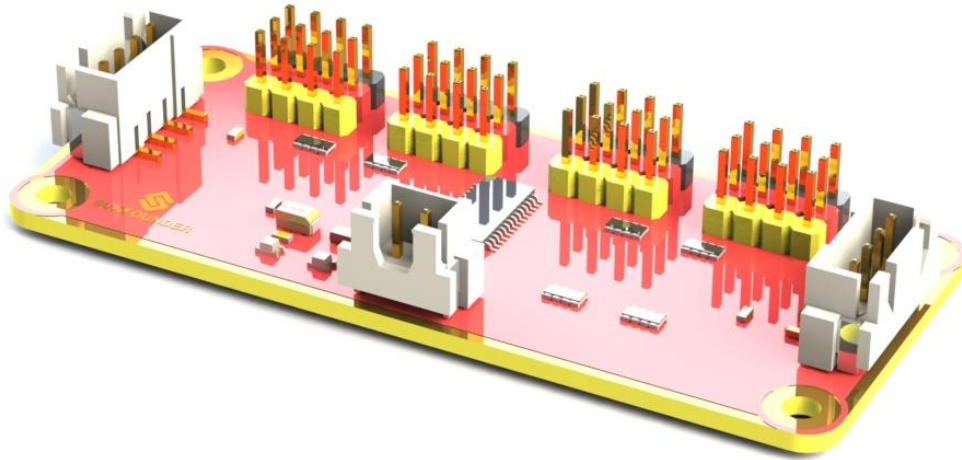
ロボット HATS は 40 ピン Raspberry Pi に特別に設計された HAT であり、Raspberry Pi モデル B +、世代 2 モデル B、世代 3 モデル B、世代 3 モデル B +、世代 4 モデル B で動作し、GPIO ポートから Raspberry Pi に電力を供

給する。HATS のルールに基づいた理想的なダイオードの設計により、USB ケーブルと DC ポートの両方を介して Raspberry Pi に電源を供給できるため、バッテリーの電力不足によって TF カードが損傷することを防ぐことができる。PCF8591 は I2C 通信とアドレス 0x48 を備えた ADC チップとして使用される。

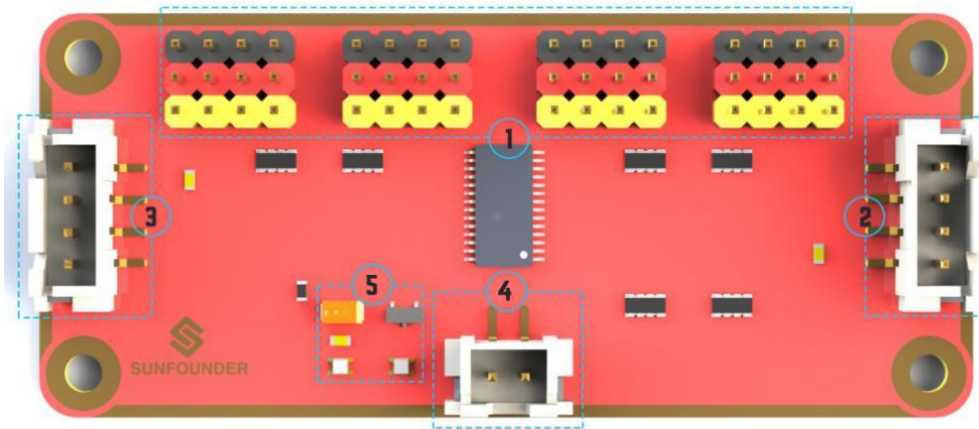


1. **デジタルポート**：3 線式デジタルセンサーポート、信号電圧：3.3V、VCC 電圧：3.3V。
2. **アナログポート**：3 線 4 チャンネル 8 ビット ADC センサーポート、基準電圧：3.3V、VCC 電圧：3.3V。
3. **I2C ポート**：3.3V I2C バスポート
4. **5V 電源出力**：PWM ドライバーへの 5V 電源出力。
5. **UART ポート**：4 線 UART ポートと 5V VCC は USB への SunFounder FTDI シリアルと完全に連携する。
6. **モーター制御ポート**：モーター用 5V、モーター MA と MB の方向制御、フローティングピン NC、SunFounder モータードライバモジュールとの連携。
7. **スイッチ**：電源スイッチ
8. **電源インジケータ**：電圧を指示する-2 つのインジケータが点灯：> 7.9V。1 つのインジケータ：7.9V～7.4V。インジケータ点灯なし：<7.4V。バッテリーを保護するために、インジケータが点灯していない場合は、充電する際にこれを取り出してください。電源インジケータは単純なコンパレータ回路によって測定された電圧に依存する。負荷によっては検出電圧が通常より低下する場合があるので参考値としてご利用ください。
9. **電源ポート**：5.5/2.1mm 標準 DC ポート、入力電圧：8.4～7.4V（制限された動作電圧：12V～6V）。

PCA9865



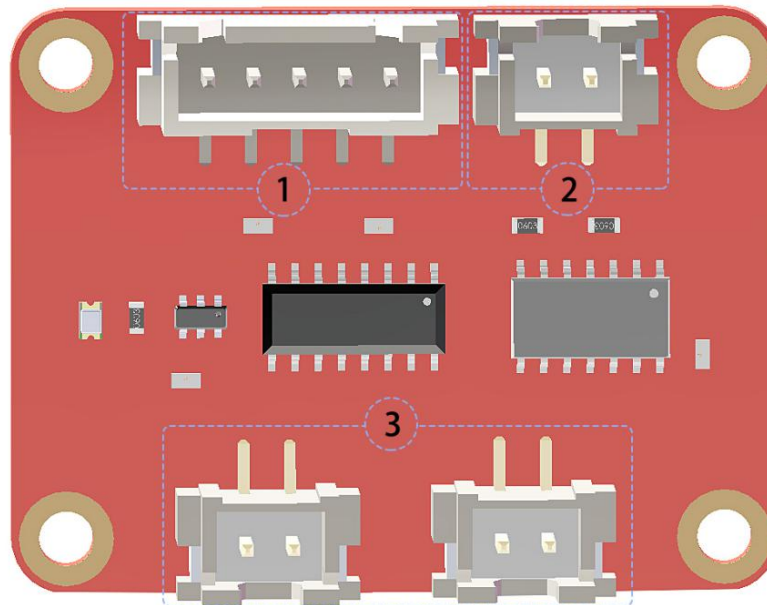
PCA9865 16 チャンネル 12 ビット I2C バス PWM ドライバー。独立した PWM 出力電力をサポートし、並列接続用の 4 線式 I2C ポート、PWM 出力用の区別された 3 色ポートをより簡単に利用できる。



1. **PWM 出力ポート**：3 色ポート、独立したパワー PWM 出力ポート、サーボへの直接接続。
- 2 & 3. **I2C ポート**：4 線式 I2C ポートは並列で使用できる。3.3V/5.5V に対応。
4. **PWM 電源入力**：最大 12V。
5. **LED**：チップと PWM 電源入力用電源インジケータ。

モータードライバーモジュール

モータードライバーモジュールは低発熱の小型パッケージモータードライブである。



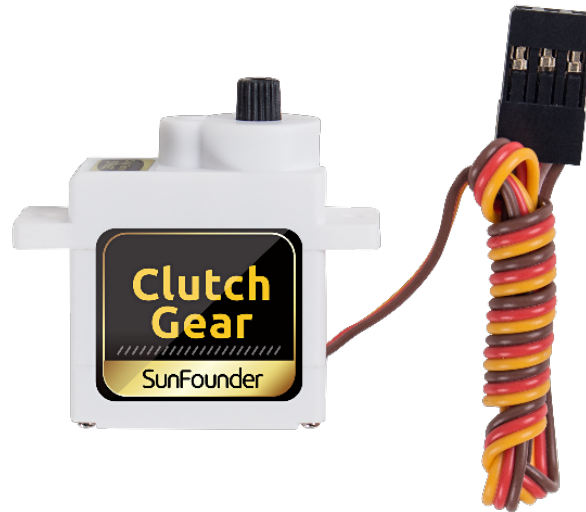
1. **電源とモーター制御ポート**：チップとモーターに電力を供給し、モーターの方向を制御するためのピンが含まれている。
2. **モーターの PWM 入力**：2つのモーターの速度を調整するための PWM 信号入力。
3. **モーター出力ポート**：2つのモーターの出力ポート。

USB ウェブカメラ



このカメラは 120° の広角をサポートするため、広く鮮明な視界が得られて、PiCar-V で使用するときのエクスペリエンスが向上する。

SunFounder SF006C サーボ

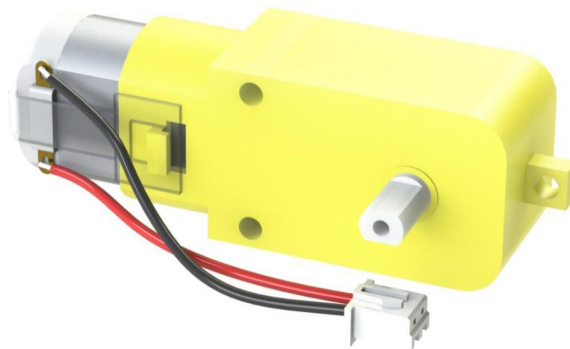


クラッチギアデジタルサーボの内部に DC コアモーターが搭載されており、一定の負荷をかけた後、ステアリングギアの減速装置は自動的にクラッチを切り、損傷や通常の負荷から製品を保護する。

パフォーマンスの機能：

アイテム	V = 4.8V	V = 6.0V
消費電流*（無負荷）	≤ 50mA	≤ 60mA
ストール電流	≤ 550mA	≤ 650mA
定格トルク	0.6 kgf·cm	0.7 kgf·cm
最大トルク	1.4 kgf·cm	1.6 kgf·cm
負荷速度なし	≤ 0.14sec/60 °	≤ 0.12sec/60 °

DC ギアモーター



これは減速機付きの DC モーターである。以下のパラメーターを参照してください：

Motor	Model	F130SA-11200-38V
	Rated Voltage	4.5V-6V
	No-load Current	≤80mA
	No-load Speed	10000±10%
Gear Reducer	Gear Ratio	1:48
	Speed (no-load)	≈200rpm (≈180rpm in test)
	Current	≤120mA

1.9.3 Filezilla ソフトウェア



ファイル転送プロトコル（FTP）は、コンピュータネットワーク上でサーバーからクライアントへコンピュータファイルを転送するために使用される標準的な通信プロトコルです。

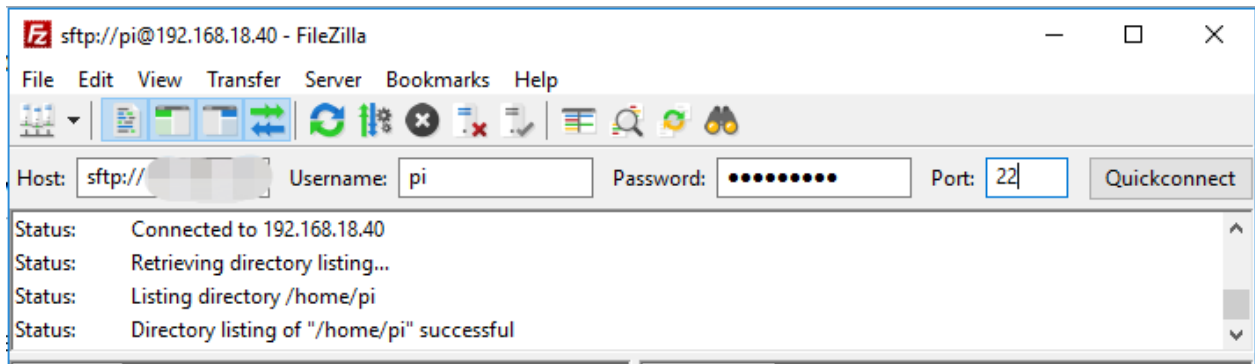
Filezilla は FTP だけでなく、FTP over TLS（FTPS）および SFTP もサポートするオープンソースソフトウェアです。Filezilla を使用して、ローカルファイル（写真やオーディオなど）を Raspberry Pi にアップロードしたり、Raspberry Pi からローカルにファイルをダウンロードすることができます。

ステップ 1: Filezilla をダウンロードする。

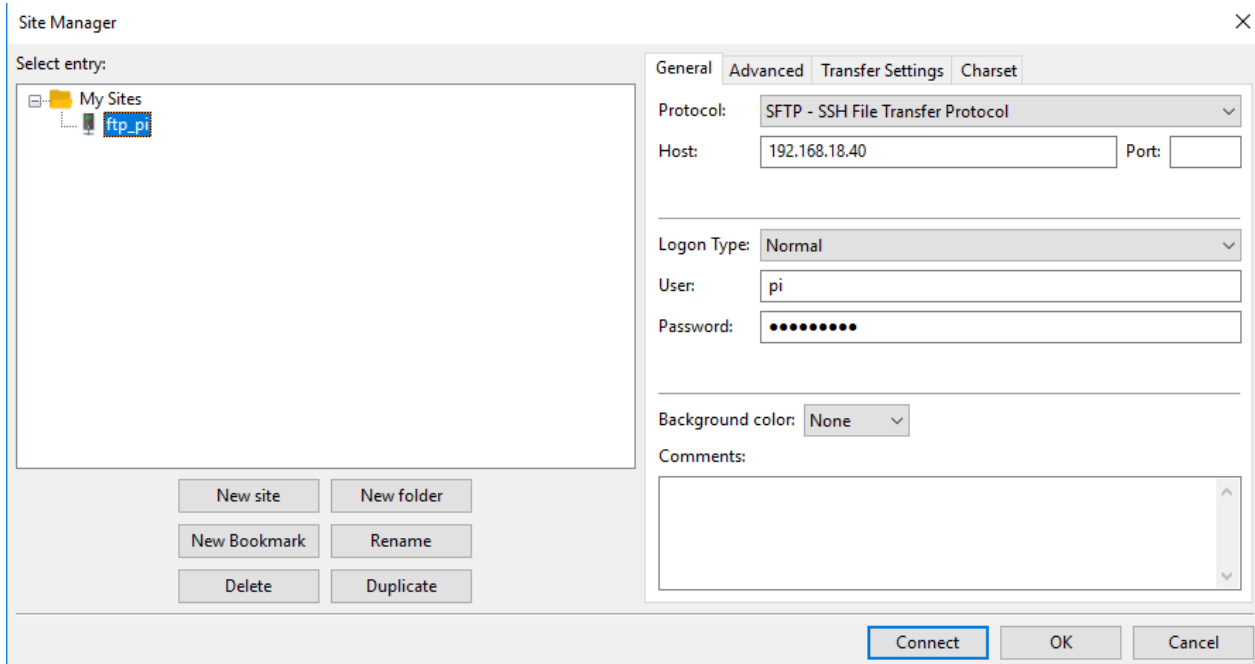
クライアントを [Filezilla の公式ウェブサイト](#) からダウンロードしてください。Filezilla には非常に良いチュートリアルがありますので、[Documentation - Filezilla](#) を参照してください。

ステップ 2: Raspberry Pi に接続する

素早くインストールした後、[FTP サーバーに接続する](#) ためにそれを開いてください。接続方法は 3 つありますが、ここでは **クイック接続** バーを使用します。**ホスト名/IP**、**ユーザー名**、**パスワード**、**ポート (22)** を入力し、**クイック接続** をクリックするか、**Enter** を押してサーバーに接続します。

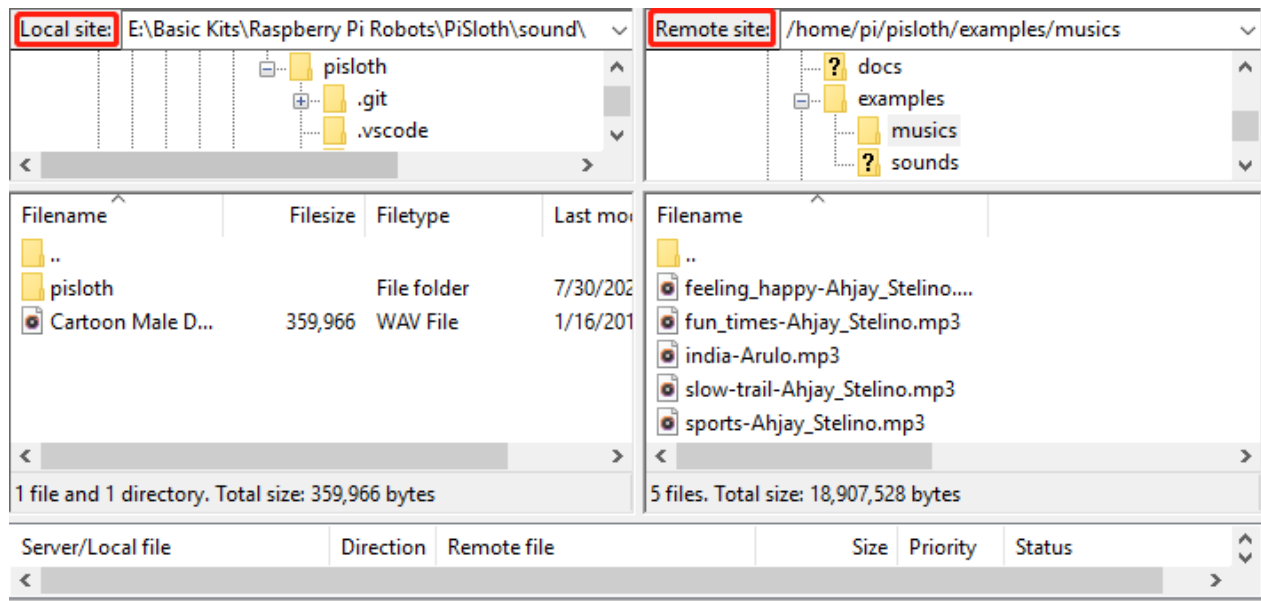


注釈: クイック接続は、ログイン情報をテストするのに良い方法です。永続的なエントリーを作成したい場合は、クイック接続に成功した後に **ファイル -> 現在の接続をサイトマネージャーにコピー** を選択し、名前を入力して **OK** をクリックします。次回からは、**ファイル -> サイトマネージャー** 内で以前に保存したサイトを選択することで接続できます。



ステップ 3: ファイルをアップロード/ダウンロードする。

ローカルファイルをドラッグアンドドロップすることで Raspberry Pi にアップロードすることができますし、Raspberry Pi 内のファイルをローカルにダウンロードすることもできます。



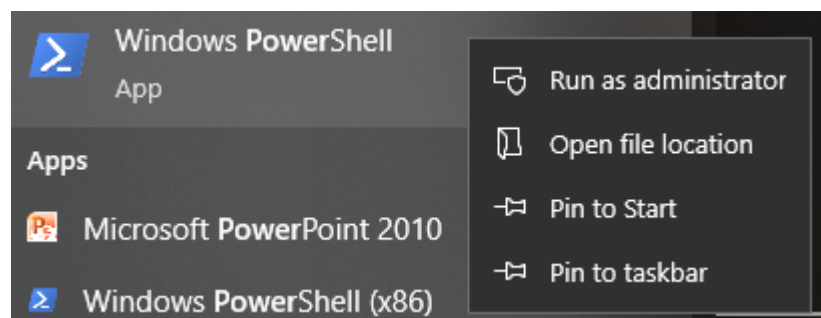
1.9.4 Powershell 経由で OpenSSH をインストールする

When you use `ssh <username>@<hostname>.local` (or `ssh <username>@<IP address>`) to connect to your Raspberry Pi, but the following error message appears.

```
ssh: The term 'ssh' is not recognized as the name of a cmdlet, function, script
file, or operable program. Check the
spelling of the name, or if a path was included, verify that the path is correct
and try again.
```

これは、お使いのコンピュータシステムが古すぎて OpenSSH がプリインストールされていないことを意味します。以下のチュートリアルに従って、手でインストールする必要があります。

1. Windows デスクトップの検索ボックスに `powershell` と入力し、Windows PowerShell を右クリックして、表示されるメニューから **管理者として実行** を選択します。



2. 次のコマンドを使用して OpenSSH.Client をインストールします。

```
Add-WindowsCapability -Online -Name OpenSSH.Client~~~~0.0.1.0
```

3. インストール後、次の出力が返されます。

```
Path      :  
Online    : True  
RestartNeeded : False
```

4. 次のコマンドを使用してインストールを確認します。

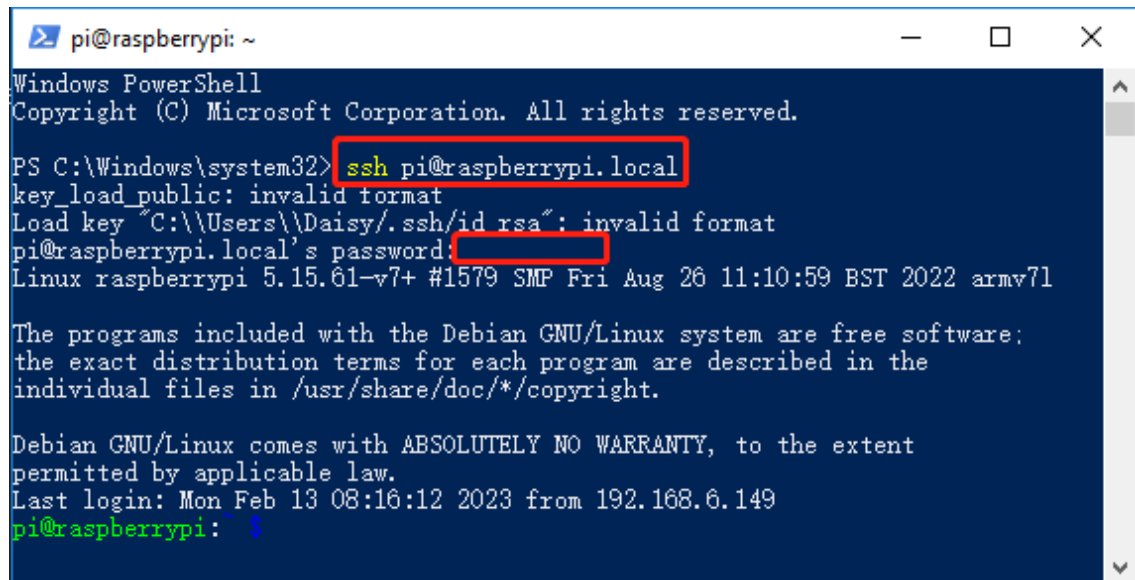
```
Get-WindowsCapability -Online | Where-Object Name -like 'OpenSSH*'
```

5. これで OpenSSH.Client のインストールが成功したことがわかります。

```
Name : OpenSSH.Client~~~~0.0.1.0  
State : Installed  
  
Name : OpenSSH.Server~~~~0.0.1.0  
State : NotPresent
```

警告: 上記のプロンプトが表示されない場合、Windows システムがまだ古すぎることを意味し、:ref:`login\_windows` のようなサードパーティの SSH ツールをインストールすることをお勧めします。

6. これで PowerShell を再起動し、引き続き管理者として実行します。ここで、ssh コマンドを使用して Raspberry Pi にログインすることができ、以前に設定したパスワードの入力を求められます。



```
pi@raspberrypi: ~
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Windows\system32> ssh pi@raspberrypi.local
key_load_public: invalid format
Load key "C:\\Users\\Daisy\\.ssh\\id_rsa": invalid format
pi@raspberrypi.local's password:
Linux raspberrypi 5.15.61-v7+ #1579 SMP Fri Aug 26 11:10:59 BST 2022 armv7l

The programs included with the Debian GNU/Linux system are free software:
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Feb 13 08:16:12 2023 from 192.168.6.149
pi@raspberrypi: ~$
```

1.9.5 PuTTY

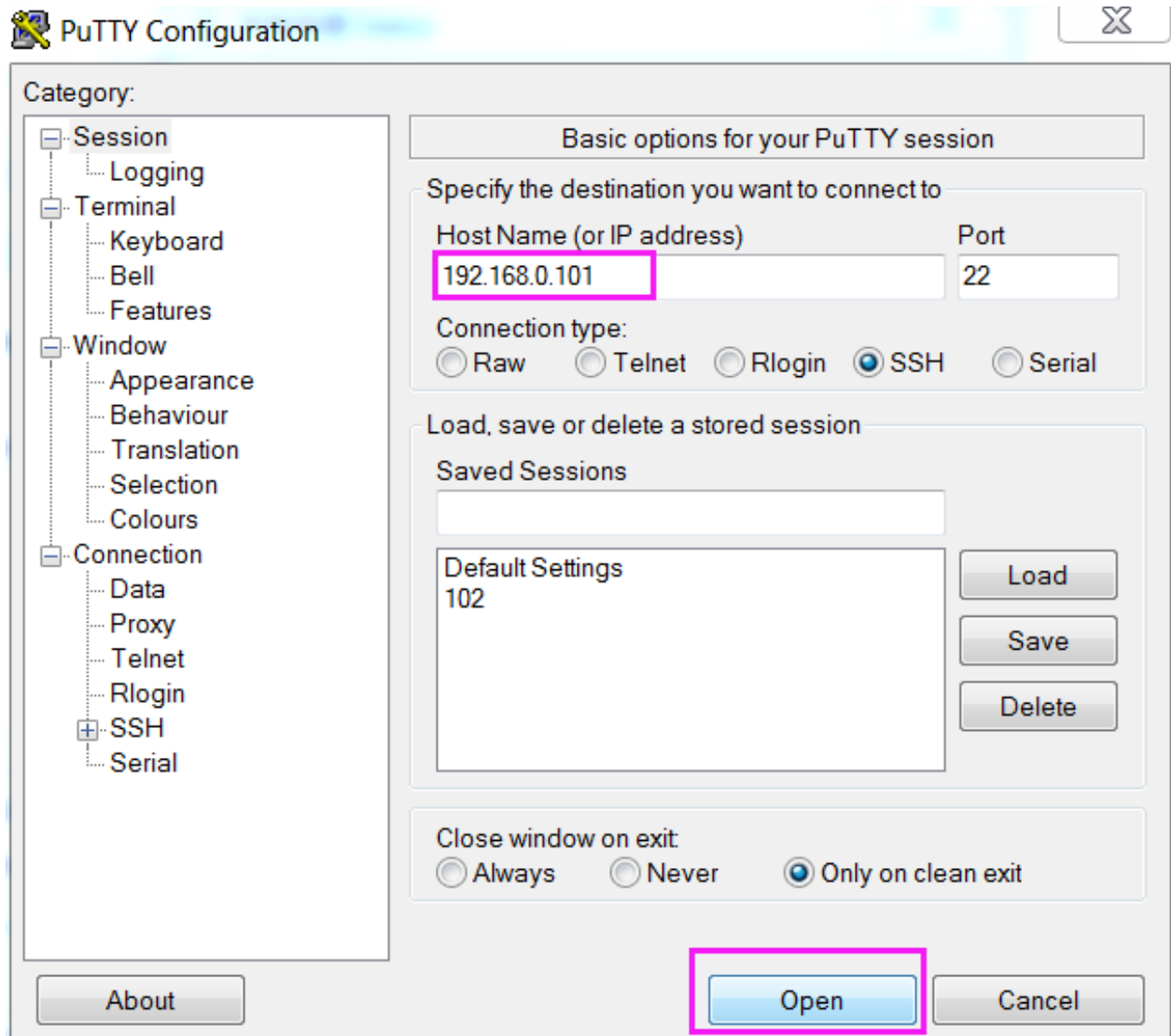
Windows ユーザーの場合、SSH を利用するためのアプリケーションがいくつかあります。ここでは PuTTY を推奨します。

ステップ 1

PuTTY をダウンロードする。

ステップ 2

PuTTY を開き、左側のツリー構造の **セッション** をクリックします。 **ホスト名（または IP アドレス）** の下のテキストボックスに RPi の IP アドレスを入力し、 **ポート** にはデフォルトで 22 と入力します。



ステップ 3

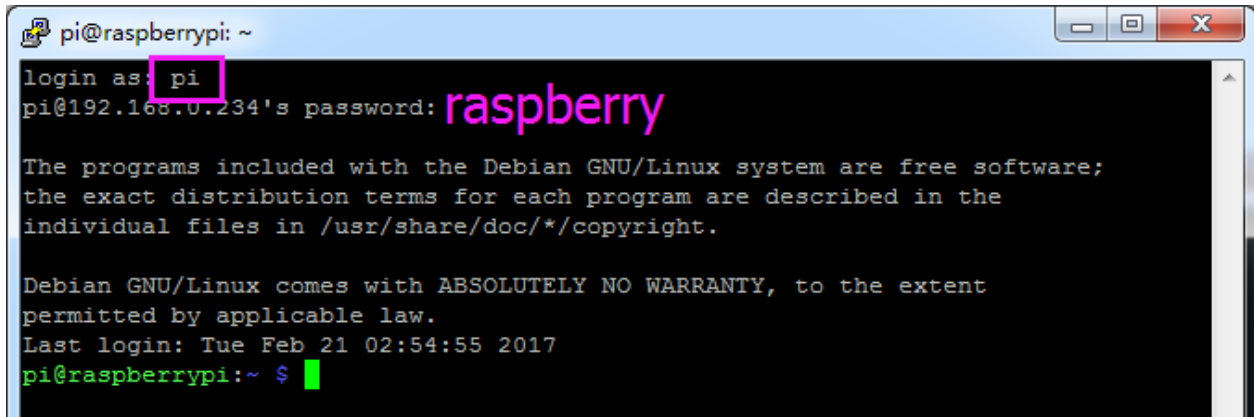
Click **Open**. Note that when you first log in to the Raspberry Pi with the IP address, there prompts a security reminder. Just click **Yes**.

ステップ 4

PuTTY ウィンドウに「**ログイン名:**」と表示されたら、「**pi**」(RPI のユーザー名)と入力し、**パスワード** には「**raspberry**」(変更していない場合のデフォルト)を入力します。

注釈: パスワードを入力するとき、文字はウィンドウに表示されませんが、これは正常です。正しいパスワードを入力する必要があります。

PuTTY の横に「非アクティブ」が表示された場合、接続が切断されて再接続が必要であることを意味します。



ステップ 5

これで Raspberry Pi が接続され、次のステップを実行する時間です。

1.10 ありがとうございました

私たちの製品を評価した評価者、チュートリアルのための提案を提供したベテラン、そして私たちをフォローしてサポートしてきたユーザーのおかげで。私たちにあなたの貴重な提案は、より良い製品を提供する私たちのモチベーションです!

特定の感謝

- レン・ヴィセソン
- カレン・ダニエル
- フアン・デラコスタ

さて、このアンケートに記入するのに少し時間を割いてもらえますか?

注釈: アンケートを送信した後、トップに戻って結果を確認してください。

第 2 章

著作権表示

このマニュアルのテキスト、画像とコードを含むがこれらに限定されないすべての内容は、SunFounder Company が所有している。関連する規制と著作権法に基づき、著者と関連する権利所有者の法的権利を侵害することなく、個人的な研究、調査、享楽、またはその他の非営利目的でのみ使用してください。許可なく営利目的でこれを使用する個人または組織については、会社は法的措置を取る権利を留保する。