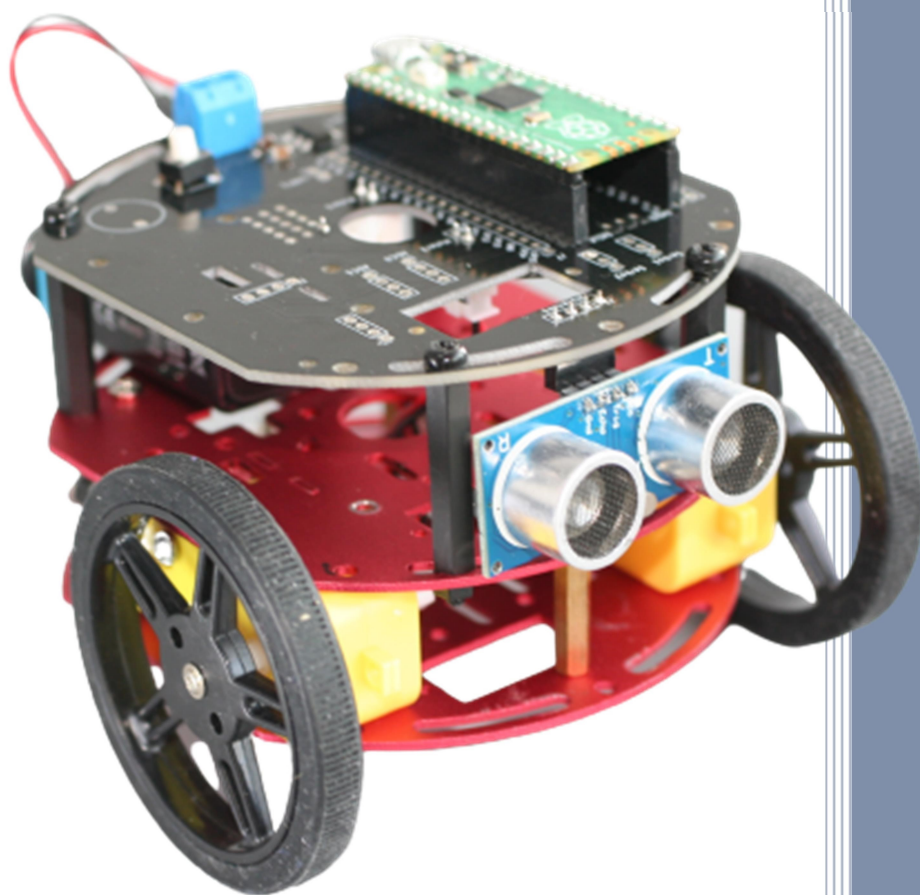


ARE SERIES

ARE-CAR-2W 組立・使用マニュアル

Raspberry Pi Pico 版



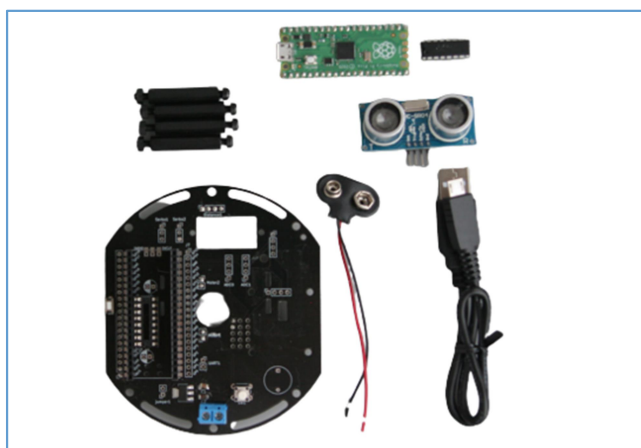
Ability

はじめに

当マニュアルは、Ability の電子工作&プログラミング講座でも使用している、ロボットカー（ARE-CAR-2W）の Raspberry Pi Pico 版の組立てと、使い方について記述しています。

1.セット内容

1.基板&電子部品一式



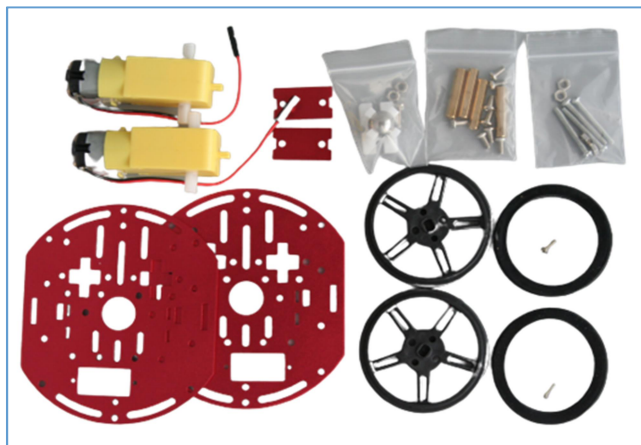
基板用支柱×4
基板（ARE-CAR（Pico））×1
Raspberry pi pico×1
L293D モータードライバ×1
超音波距離センサー×1
バッテリースナップ×1
USB ケーブル×1

ピンヘッダー、ターミナルブロック、タクトスイッチ、コネクタ等の電子部品は、はんだ付けしてお届けします。

※基板には DC モーター、赤外線ライントレースセンサー用のコネクタと、UART 用のコネクタ付き

※ご用意いただくものは 9V 乾電池×1、ドライバー、両面テープ等（乾電池が固定出来れば何でも可）

2.FT-DC-002（FEETECH 社製）

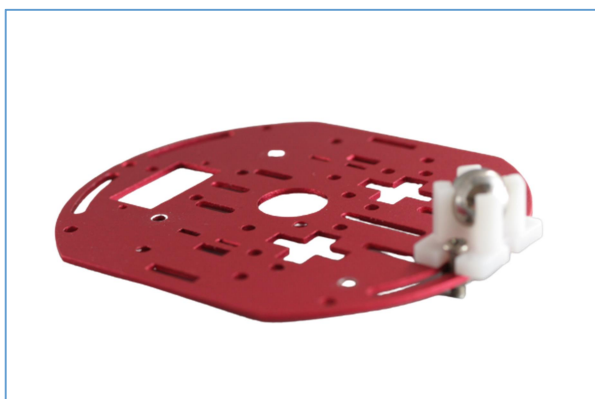


詳細については以下サイトでご確認ください。

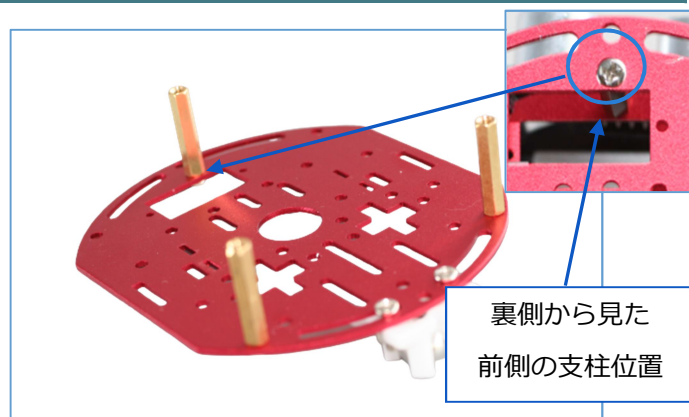
<https://www.feetechrc.com/products.html?keyword=FT-DC-002>

2.組み立て手順

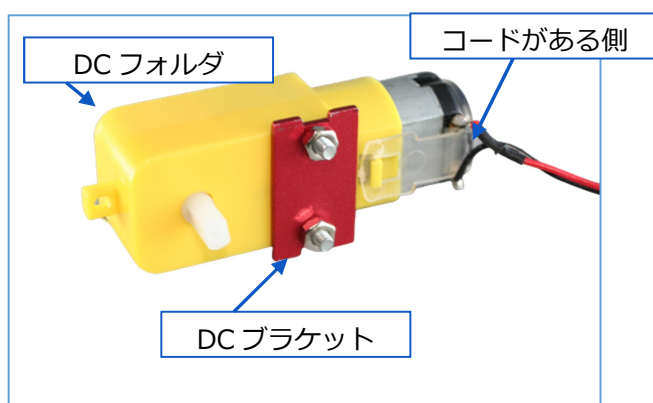
1.FT-DC-002 の組み立て



車体に後輪を取り付けます。



車体に金色の支柱を3本、ネジで固定します。



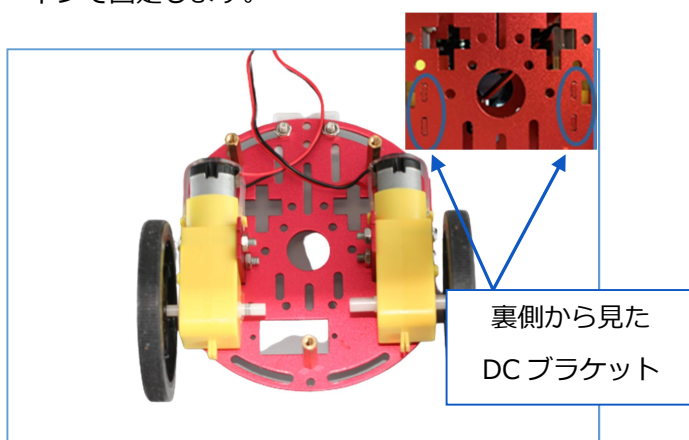
DC フォルダに DC ブラケットを、ネジとナットでコードのある側に固定します。



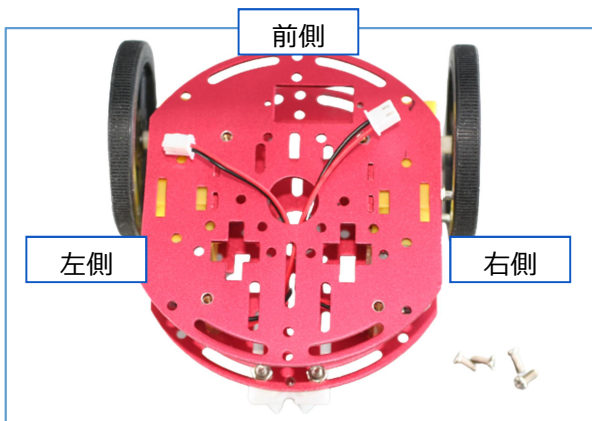
DC ブラケットと反対側のシャフトに、車輪をネジで固定します。



同様に2輪作ります。



DC ブラケットの端を車体に差し込むようにしてDC モーターを配置します。

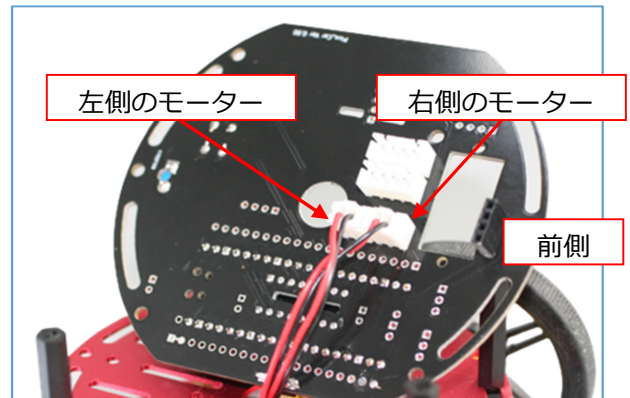


もう 1 枚の車体と支柱をネジで固定します。
モーターのケーブルは真ん中のフォールを通して
上に出してください。

2.基板と部品の取り付け



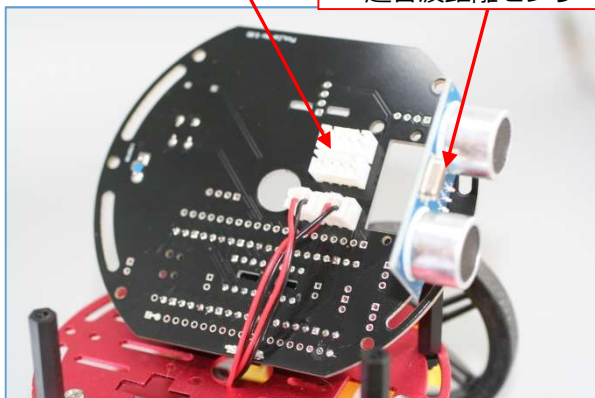
車体に黒色の支柱をナットで固定します。



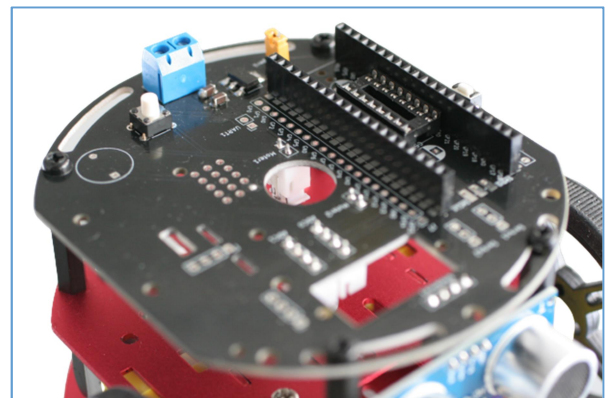
基板のコネクタにモーターのケーブルを挿します。

赤外線ライトレースセンサーのコネクタ

超音波距離センサー

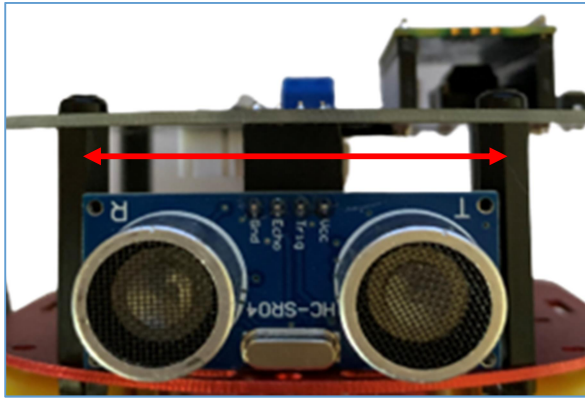


基板のコネクタに超音波距離センサーを挿します。

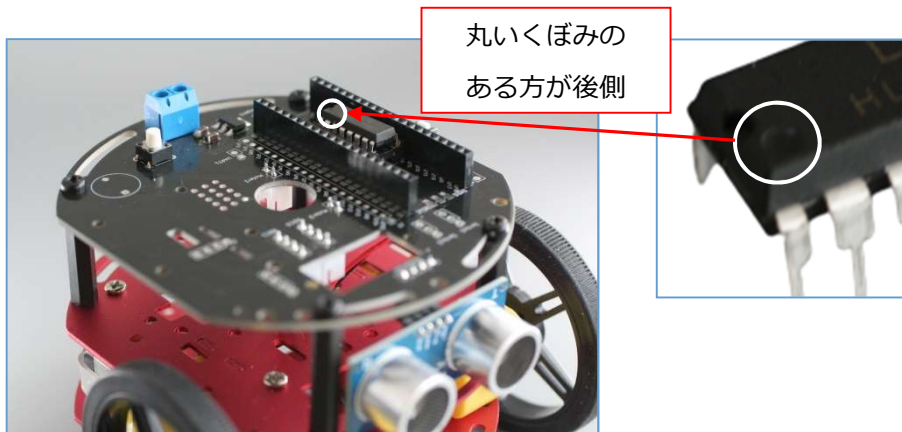


支柱に基板をネジで固定します。

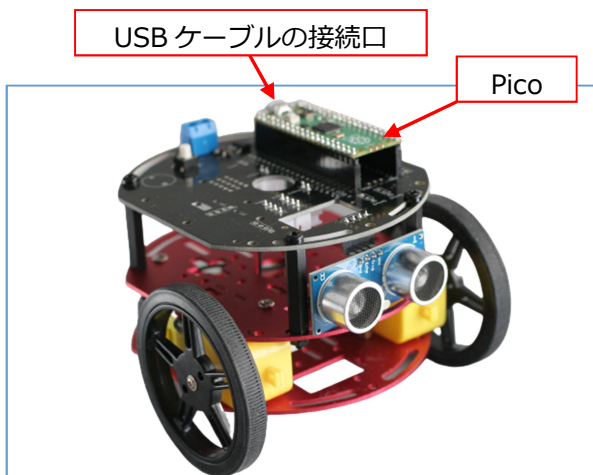
※赤外線ライトレースセンサーはセットに含まれません。



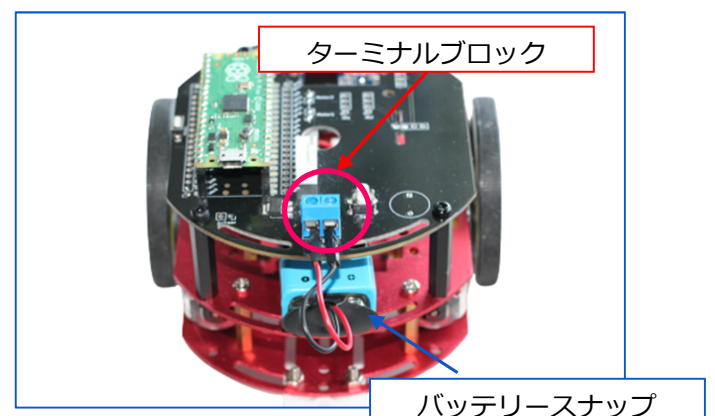
距離センサーが支柱に沿うように、幅を調整して取り付けてください。



モータードライバを挿します。



Pico の 1 ピンを、基板の GP0 ピンの位置に挿します。Pico の USB ケーブルの接続口側が車体の後側になります



ターミナルブロックに、バッテリースナップを取り付けます
赤いコードが「+電極」で、ターミナルブロックの左側に付けます。黒いコードが「GND」で、ターミナルブロックの右側に付けます

3. モーターの個体差の調整

モーターやギアの個体差により、真っ直ぐに走行しない場合があります。

その場合は、Pico に搭載しているサンプルプログラムの調整値を修正します。

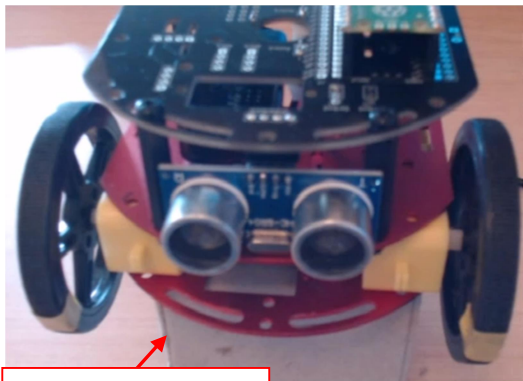
開発環境の構築や Thonny でのプログラミングについては、以下サイトの講習 1 を参考にしてください。

https://robot.ability-evolves.com/online-course/2wbeginner/#beginnercourse_2w_class1

1. 調整手順

1. 組み立てたロボットカーの車体を台に乗せて、車輪が空転するようにします。

両輪にテープ等で目印を付けると回転が見やすいです。



台に乗せる

2. PC と Pico を USB ケーブルで接続し、Thonny で Pico 搭載の `adjust_rotation.py` を開きます。

前進時の速度から調整しますので、22、23 行目をリマークします。

```
12 def main():
13     """
14     メイン関数
15     ロボットカーの両輪の回転速度を調整する
16     :param: なし
17     :return: なし
18     """
19     while True:
20         motor_functions.forward(30) # 前進関数をCall
21         time.sleep(2)
22         motor_functions.backward(30) # 後進関数をCall
23         time.sleep(2)
24
```

22, 23 行目をリマーク

`motor_functions.forward`、`motor_functions.backward` のパラメータは、回転速度の割合を指定します。

FT-DC-002 のモーターの使用電圧は 3V～6V ですが、ロボットカーの電源電圧は 9V ですので、

モーターに 6V 以上の電圧がかからないように、**最大でも 60%以下**で設定してください。

モーターの個体差にもよりますが、20%～30%くらいがお勧めです。

3. プログラムを実行して両輪の回転速度差を確認します。速度差がなければ前進の調整は終了です。

4.速度差があった場合、Pico 搭載の motor_functions.py のソースを開きます。

motor_functions.py の 27 行目の調整値を変更します。

```

25
26
27 FORWARD_ADJ_VALUE = 1.0 # 前進の調整
28 BACKWARD_ADJ_VALUE = 1.0 # 後進の調整
29 TURN_R_ADJ_VALUE = 1.0 # 右回りの調整
30 TURN_L_ADJ_VALUE = 1.0 # 左回りの調整
31

```

前進の調整の値を変更

これらの調整値は左側のモーターに使用します。

左側の回転速度が遅い場合は値を大きくし、速い場合は小さく設定します。

設定したらファイルを保存して、adjust_rotation.py を実行します。

左右の車輪が同じように回転するまで、値の変更と実行を繰り返して調整します。

5.後進も調整します。

```

12 def main():
13     """
14     メイン関数
15     ロボットカーの両輪の回転速度を調整する
16     :param: なし
17     :return: なし
18     """
19     while True:
20         motor_functions.forward(30) # 前進関数をCall
21         time.sleep(2)
22         motor_functions.backward(30) # 後進関数をCall
23         time.sleep(2)
24

```

20, 21 行目をリマーク

22, 23 行目のリマークを外す

前進と同様に、左右の車輪が同じように回転するまで、値の変更と実行を繰り返して調整します。

motor_functions.py の 28 行目の調整値を変更します。

```

25
26
27 FORWARD_ADJ_VALUE = 1.0 # 前進の調整
28 BACKWARD_ADJ_VALUE = 1.0 # 後進の調整
29 TURN_R_ADJ_VALUE = 1.0 # 右回りの調整
30 TURN_L_ADJ_VALUE = 1.0 # 左回りの調整
31

```

後進の調整の値を変更

6.右回り、左回りの調整に、前進の調整値を設定します。

motor_functions.py の 29、30 行目の調整値を変更します。

```

25
26
27 FORWARD_ADJ_VALUE = 1.0 # 前進の調整
28 BACKWARD_ADJ_VALUE = 1.0 # 後進の調整
29 TURN_R_ADJ_VALUE = 1.0 # 右回りの調整
30 TURN_L_ADJ_VALUE = 1.0 # 左回りの調整
31

```

前進の調整値を設定

4. ロボットカーの走らせ方

1. 検知した物体に付いていくプログラムについて

サンプルソース (code.py) として、検知した物体に付いていくプログラムを搭載しています。

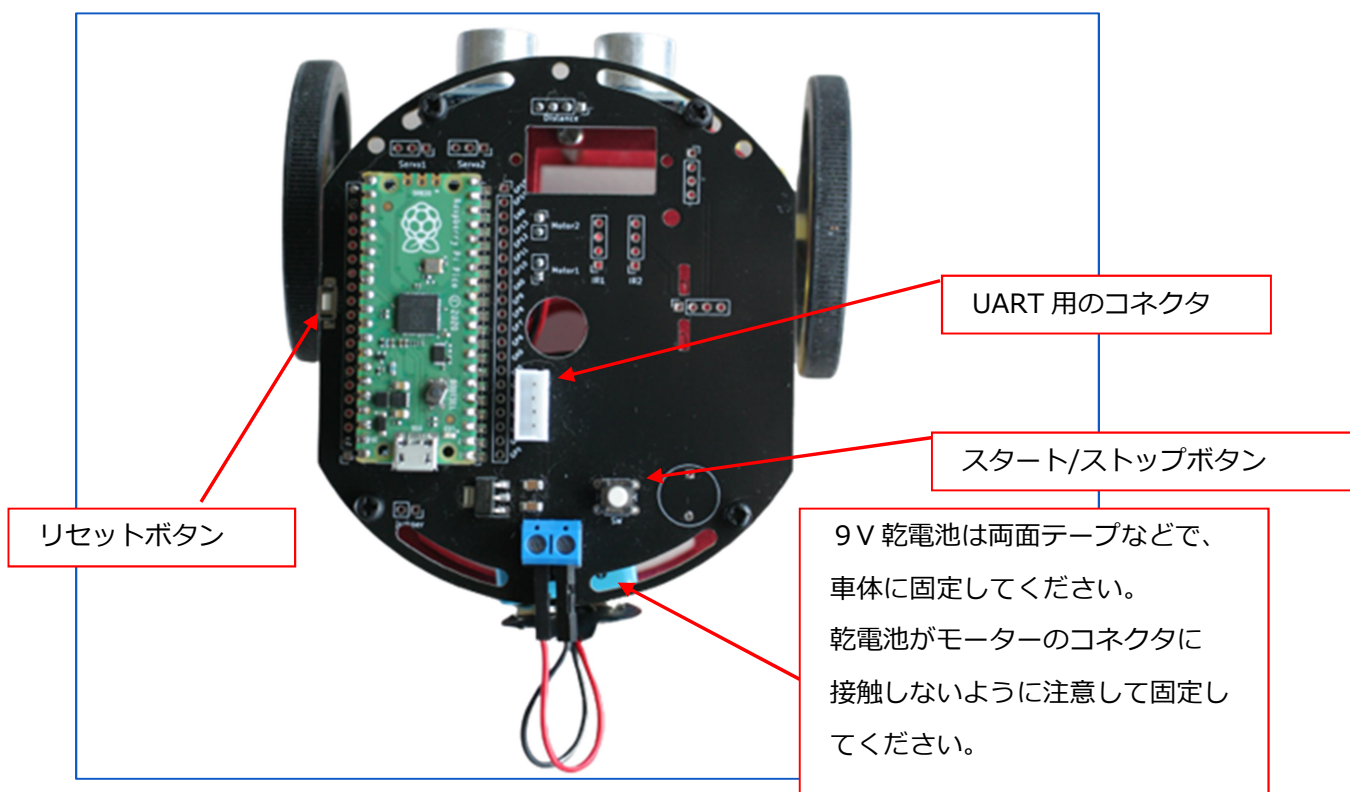
電源から電力が供給されると LED を点灯しながら、スタート/ストップボタンが押されるまで待機します。

スタート/ストップボタンが押されると、超音波距離センサーで、障害物を検知するまで走行し、障害物との距離が 10cm 以上、15cm 未満の間は停止します。

障害物との距離が 10cm 未満になると後進します。

モーターの回転速度はデフォルト値で 15% になっています。モーターの個体差で、15% では回転しないときや速度を変更したいときは、「3. モーターの個体差の調整」と同様に、code.py の `motor_functions.forward`、`motor_functions.backward` のパラメータを変更し、Pico に保存してください。

※速度を早く設定しすぎると、超音波距離センサーの計測が追いつかず、ぶつかってロボットカーが破損する恐れがあります。速度を上げる場合は、ご注意の上、自己責任でお願いいたします。



ロボットカーには電源スイッチがありませんので、使用しないときは、バッテリースナップから乾電池を取り外してください。

また、破損の恐れがありますので、必ず通電していない状態 (※) で、電子部品の取り付けや取り外しをおこなってください。

※通電していない状態：PC と Pico を接続していない、バッテリースナップから乾電池を外す

2. 走らせ方

- ① バッテリースナップに乾電池を付けて車体に固定したら、床などのフラットな場所に置きます。
基本のロボットカーには、前側に距離センサーが 1 個ですので、前方向しか計測できません。
また、段差等があると、落下して破損する恐れがありますのでご注意ください。
- ② Pico の LED の点滅を確認します。(待機状態)
LED が点滅しない場合、リセットボタンを押してください。
- ③ スタート/ストップボタンを押すと距離を計測して走行を開始します。
- ④ 再度スタート/ストップボタンを押すとプログラムを終了します。
- ⑤ 走行を再開するには、リセットボタンを押してから、スタート/ストップボタンを押してください。

5. 特記事項

基板には、Ability のオンライン講座用に赤外線ライントレースセンサー用のコネクタと、UART 用のコネクタが付いています。2022 年 6 月現在は、それらの部品を使用する講座を準備中ですので、使用方法などの説明は割愛させていただきます。

また、基板には Servo 等の印字もありますが、当マニュアルで説明した配線以外は、開発中のため保証対象外とさせていただきます。

当マニュアルで説明する操作をおこなうことによって、万が一事故など起きても、Ability では一切の責任を負いません。

当マニュアルの内容について著作権法の定める範囲を超えて、Ability に無断で複写、複製、転載することはご遠慮ください。

当マニュアルに記載しているホームページアドレス、製品の仕様等は予告なく変更されることがあります。
当マニュアルの内容について運用した結果の影響については、Ability では責任を負いかねます。
予めご了承ください。

Ability ホームページ：

<https://robot.ability-evolves.com>

お問い合わせ：

<https://robot.ability-evolves.com/inquiry>

E-Mail：

information@ability-evolves.com