

自動運転ミニカー AD-RC Car TT02

マニュアル

Last update: 2024/3/21

合同会社 MODECO

1 目次

1	目次.....	1
2	使用に際しての注意事項.....	1
3	ご準備いただくもの.....	3
4	本体・モバイルバッテリーの充電について.....	4
4.1	タミヤ RC サーボ バッテリーの充電方法.....	4
4.2	モバイルバッテリーの充電方法.....	4
5	本体の接続方法.....	5
5.1	接続用ケーブルの準備.....	5
5.2	ケーブルの接続.....	5
6	各コンポーネントの動作確認について.....	7
6.1	テスト用プログラムの場所.....	7
6.2	LIDAR のテスト.....	7
6.3	カメラのテスト.....	8
6.4	エンコーダのテスト.....	8
6.5	PWM 駆動のテストとキャリブレーション.....	9
7	納品時設定情報.....	11
8	GPIO ピン接続.....	11
9	その他注意事項.....	12
10	改定履歴.....	12

2 使用に際しての注意事項

注意

- ・自動運転ミニカー AD-RC Car TT02 キット(以下本製品)を使用するに伴った、あらゆる損害、賠償の責任を負いかねます。電源やバッテリーの取り扱い、ならびに配線等をするにあたって、各種技術情報を理解した上で、自己責任でのご利用をお願いいたします。
- ・本製品は実際に走行を伴います。周囲に人のいない、安全な場所で走行するようお願いいたします。

- ・本製品はタイヤ部が回転し、また本体が走行に伴い移動し他の障害物等に衝突する危険があります。各所の配線の巻き込み、損傷がないよう、十分に注意してご使用ください。特にラズパイ等搭載機器に有線キーボードやマウス、モバイルバッテリーの充電ケーブル、HDMI ケーブルなど、外部配線を繋ぎながらの走行テストは、配線を引っ張りコネクタ部の損傷につながる可能性がありますので避けてください。
- ・電源を繋ぐ前に、各種配線が正しく接続されていることをご確認ください。
- ・防水、防塵等は施されておりません。本製品は屋内での利用を想定しています。水濡れ、砂埃を被ることなどがないようにご注意ください。
- ・バッテリーを充電される際は、製品に付属のバッテリーの使用上の注意をよく読み、適切にご利用ください。

- ・マニュアルの不具合等が修正されている場合があります。最新の情報は下記 URL または右 QR コードの商品ページからご確認ください。

<https://modeco-ad.jp/jpn/modecar/modecar-tt02/>



3 ご準備いただくもの

本製品をご利用頂くにあたって、ご準備いただくものは下記です。

[準備が必要なもの]

- USB PD (Power delivery, Type C コネクタのもの)対応充電器 (18W 以上)
- USB 接続式有線キーボード (初期設定に必要. 設定後は無線のものだけでも可)
- USB 接続式有線マウス (初期設定に必要. 設定後は無線のものだけでも可)
- HDMI ポートで接続, 表示が可能なモニタ

[ご準備いただくと便利なもの]

- Bluetooth 接続型または USB レシーバ接続タイプの無線キーボード・マウス
(走行時に外す必要が無いうえ, そのまま遠隔でジョグ運転等に利用可能.)
- 予備用走行バッテリー([タミヤニカドバッテリー 7.2V カスタムパック](#)等)
(走行時間が長くなる場合. 付属のバッテリーは”540 モーター搭載のマシンで約7分の連続走行が可能”との記載あり)
- ワイヤレス HDMI 送受信機
(走行時に HDMI ケーブルを外す必要がない. ただし画面が遅延しがちで操作性が悪いものもあるので注意する)

4 本体・モバイルバッテリーの充電について

注意

- ・TAMIYA の本体付属のバッテリー，ならびにモバイルバッテリーを使用，充電する際には，必ず付属の製品の取扱説明書に沿って正しくご利用ください。使用中，充電中に関わらず，あらゆる障害，損害，賠償に関する責任を負いかねます。これらに関しては製品をそのまま利用しておりますので，製品に問題がありましたらそれぞれの販売店，製造元にご連絡ください。

4.1 タミヤ RC サーボ バッテリーの充電方法

RC カー本体のバッテリー（付属のものは青色）は本体左側です。バッテリーのカバープレート（右写真赤枠）の，向かって左側のみのボディピンを外し，回転してバッテリーの上をあけ，バッテリーを取り出します。右側のボディピンはエンコーダモジュールとの干渉がありますので，できるだけ外さないでください。

発熱する場合がありますので，できるだけ本体外の熱等に影響を受けない場所で充電してください。取り外した本体に，付属の充電器を接続し，充電器・アダプタ部をコンセントに接続します。



4.2 モバイルバッテリーの充電方法

付属のモバイルバッテリーは本体後方にマグネットで取り付けています。ラズパイのOSを終了状態にし，ラズパイと接続している USB TypeC ケーブルをモバイルバッテリー側で取り外してください。

ケーブルを取り外しましたら，お手持ちの USB PD 対応充電器を付属のケーブル（ラズパイと接続していたものとは別のケーブル）で接続し，充電してください。



注意

- ・PD 対応充電器は付属しておりません。お客様でご用意ください。その際，20W 以上対応のものをお勧め致します。
- ・取り外した瞬間ラズパイは電源が切れますので，必ずシャットダウン状態にした後に取り外してください。
- ・ラズパイのコネクタ部は比較的弱い構造ですので，できるだけラズパイ側は抜き差しをせず，モバイルバッテリー側で抜き差しするようにしてください。



5 本体の接続方法

本製品を利用して自動走行の準備をするには、付属の下記の各製品を接続します。

- ・ Raspberry Pi 4 Model B (以下, ラズパイ)
- ・ モバイルバッテリー
- ・ RP Lidar A1M8 (以下, LIDAR)
- ・ Raspberry Pi Camera 2 (以下, カメラ) (接続済み)
- ・ HDMI モニタ (ご用意いただいたもの)
- ・ USB 有線キーボード & マウス (ご用意いただいたもの)

5.1 接続用ケーブルの準備

接続用のケーブルとして、右の写真の袋に下記のケーブルが付属しています。

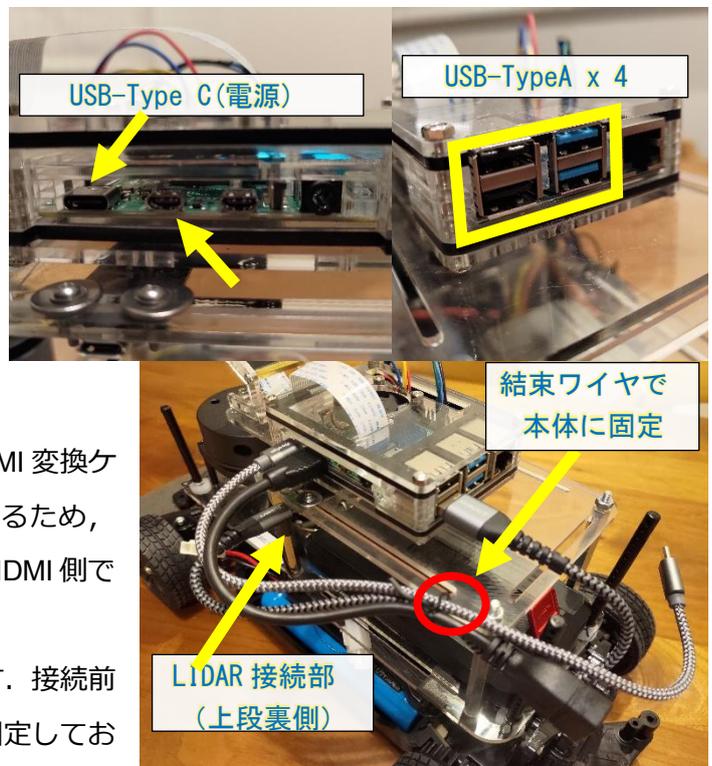
- ・ USB Type-C ケーブル (ラズパイ-モバイルバッテリー間)
- ・ USB Micro ケーブル (ラズパイ-LIDAR 間)
- ・ HDMI-Micro 変換アダプタ (ラズパイ-HDMI ケーブル間)
- ・ HDMI ケーブル (モニタ接続用)



5.2 ケーブルの接続

※ラズパイの電源となる USB-Type C ケーブルのモバイルバッテリーへの接続は最後です。

1. USB Micro ケーブルを用いて、ラズパイと LIDAR を接続します。ラズパイの TypeA 端子のうちいずれか (どれでも良い) と、ラズパイ下にある LIDAR の小基板の USB Micro コネクタを接続します。
2. ラズパイの Micro-HDMI に、Micro-HDMI-HDMI 変換ケーブルを接続します。ラズパイの端子を保護するため、HDMI の抜き差しはこの HDMI 変換ケーブルの HDMI 側で行ってください。
3. USB-Type C ケーブルをラズパイに接続します。接続前に、1~3 のケーブルを結束ワイヤ等で本体に固定しておきます。また、モバイルバッテリーのスイッチを長押しし、



モバイルバッテリーの緑の LED が点灯した状態にしてください。TypeC ケーブルをモバイルバッテリーに繋ぐとラズパイがすぐに起動しますので、モバイルバッテリー側は起動準備が整うまで接続しないようにしてください。

注意

- ・ラズパイの各種コネクタは強度的に十分ではない場合があります，走行中の引っ掛かり等で破損する可能性があります。走行中，ケーブルの引っかかりやタイヤへの絡みを予防するため本体に結束バンド等で固定してください。
- ・付属のモバイルバッテリーには，小電流時のオート OFF 機能がついており，ラズパイがスリープすると OFF になる場合があります。電源ボタンを長押しし，モバイルバッテリーの LED インジケータが緑点灯した状態でお使いください。

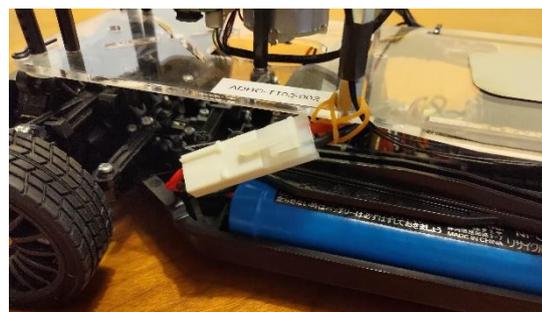
4. お手持ちの有線式のキーボード，マウスを，4つある USB Type A 端子のいずれかに接続ください。

5. お手持ちのモニタの HDMI 入力端子と，上記 2 の HDMI コネクタを接続してください。

6. ラズパイと接続した Type-C ケーブルとモバイルバッテリーを接続すると，ラズパイが起動します。

(電源スイッチはありません。終了時は，まず OS をシャットダウン状態にし，その後モバイルバッテリー側の Type-C ケーブルを引き抜き，電源を切断します。)

7. 走行の準備ができれば，本体左側にある RC サーボの本体バッテリー (青色) のコネクタを接続し，本体右側のモータ部横の本体のサーボ電源スイッチを ON にします。このとき，ラズパイから PWM 信号による指令値が入力されていないとアラーム (ピー，ピー音) が鳴りますが問題ありません。プログラムによりサーボ用 PWM が出力されれば動作します。



※プログラムの開発中など走行を予定しない準備中は，外部の USB PD 対応充電器と接続してラズパイをご利用いただけます。この場合も接続とともに起動，引き抜きで電源強制 OFF となりますので，引き抜き前に OS のシャットダウンを忘れないようご注意ください。

※しばらく走行しない場合には，過放電を予防するため，必ず RC サーボの電源スイッチ (モータ部横) を OFF にし，サーボ用バッテリーのコネクタも取り外しするようにしてください。

※モバイルバッテリーの Type-A 端子からラズパイの Type-C 端子に電源供給をすると，モバイルバッテリーを Type-C で充電しつつ使用することもできますが，ラズパイへの供給電力が不足する場合がありますので (Type-A 経由では 5V2.4A まで)，ラズパイの接続機器数や計算負荷が大きい場合は Type-C 端子から給電してください。

6 各コンポーネントの動作確認について

6.1 テスト用プログラムの場所

GitHub <https://github.com/Hiroyuki-Okuda/OpenMiniCarWorks> を参照してください。

下記の説明では、home ディレクトリから下記コマンドでリポジトリを clone することを想定します。

```
git clone https://github.com/Hiroyuki-Okuda/OpenMiniCarWorks.git
```

`/usr/rccar/home/OpenMinicarWorks/scripts/device_test/rccar_tests/` フォルダ内に各種のテスト用の python プログラムが含まれています。まずはターミナルを開き、ここまで移動してください。

```
cd /home/adrccar/OpenMinicarWorks/scripts/device_test/rccar_tests/
```

テスト用のプログラムフォルダではここを起点にします。各テストが終わったら `cd ../` 等のコマンドでこのフォルダまで戻ってください。

6.2 LIDAR のテスト

RPLidar A1M8 が動作するかテストします。

```
cd /test_lidar
python3 RPLidarA1-M8_test.py
```

[テストが成功する場合]

```
...
7: Got 138 measurements
8: Got 144 measurements
...
```

等と表示されます。

[テストが失敗する場合]

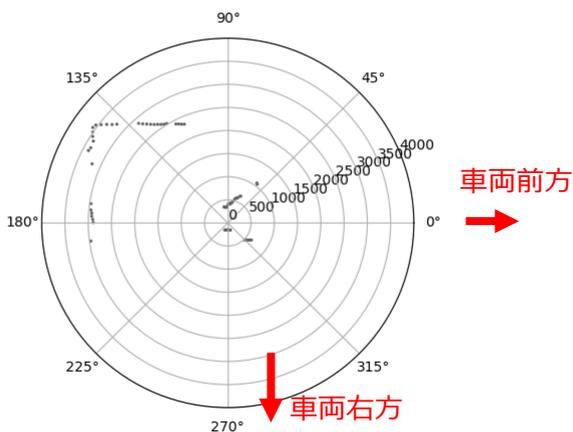
```
Traceback (most recent call last):
...
FileNotFoundError: ...      '/dev/ttyUSB0'
...
```

等といったエラーが表示されます。LIDAR の USB デバイス `'/dev/ttyUSB0'` が開けないエラーです。USB ケーブルの接続を確認してください。

Roboticia の RPLidar の example から視覚的に動作を確認することもできます。home/adrccar から、

```
cd /RPLidar/example
python3 animation.py
```

からサンプルを実行できます。ただし、使用するラズパイではフレーム処理に処理落ちが発生しエラーで停止することがありますので、安定しては動作しませんのでご注意ください。



RPLidar/example/animation.py の動作画像. 図右の 0 度が車両前方方向. ただし Rplidar は時計回りに動作, 図は反時計周りでプロットされていることから左右逆になるので注意. (図の上が車両左側, 下が車両右側)

6.3 カメラのテスト

搭載の Raspberry Pi CAMERA 2 が動作するかテストします.

```
libcamera-hello
```

カメラが正しく認識されていれば, 画面がプレビューウィンドウに表示されます.

※実行時, INFO に交じて WARN が表示されていますが, いまのところ動作に問題は見つかっておりません

```
[camera_test.py]
```

プレビューを表示し, test.jpg に画像を保存します.

```
[camera_test_OpenCV.py]
```

OpenCV で画像キャプチャを試します.

(AD-RC Car TT02 Ver. 現在, エラーは出ませんが OpenCV の VideoCapture.Read()関数に失敗します.

libcamera2 利用版をご利用ください)

```
[camera_test_picamera.py]
```

libcamera を用いて画像キャプチャを試します. 成功したら test.jpg に画像を保存します.

```
[camera_test_picamCV.py]
```

libcamera を用いて画像をキャプチャし, OpenCV2 の imshow で画面表示します.

libcamera でカメラ画像を取得し OpenCV で処理・表示できるかどうかのテストです.

6.4 エンコーダのテスト

搭載の光学エンコーダをカウンタで読み取れるかテストをします.

現状では speed_observer2.py を利用してください.

```
cd test_encoder
```

```
python3 speed_observer2.py
```

エンコーダのカウント値と速度の概算値が表示されます。(現在ソフトウェアカウントのため不正確です)

タイヤを回して値が変化していれば成功です。前進方向に動かすと+、後進方向に動かすと-の速度が表示されます。(エンコーダパルスの発生で速度を更新する仕様ですので停止中は速度が更新されない場合があります。走行中の速度計測には問題ありません)

6.5 PWM 駆動のテストとキャリブレーション

PWM を用いてステアとスピードの指令をテストします。

精度が良い反面、12番ピン(18番ピンも共用)または13番ピン(19番ピンも共用)ピンの2チャンネルしか使えないpigpioによるハードウェアPWM版と、どのGPIOピンも使えるRPi.GPIO版がありますが、本製品ではpigpio版を強く推奨します。ここではpigpio版の説明をします。

pidpio版のフォルダに入ります。2行目はpigpioデーモン(サービス)を起動しています。pigpioの機能を使うには事前にこの起動が必要です。事前にラズパイ起動時に自動起動するように設定した場合は不要です。(その場合でも、再度実行してもかまいません)

```
cd test_PWM/pigpio
sudo pigpiod
```

[PWMの指令値のキャリブレーション]

1. まず本製品の本体をリフトアップし、タイヤが接地しないようにします。
2. RCサーボの電源を入れる前に、キャリブレーション用のプログラムを実行します。

```
python3 key_calib.py
```

※タミヤRCサーボでは、PWM信号が入力されていない場合にピーピーと警告音が鳴ります。スクリプトを動かし、ニュートラルポジションに相当するPWMを出力した後にサーボの電源をONにしてください。

3. RCサーボの本体バッテリーを接続し、RCサーボの電源スイッチをONにします。

注意

- ・車両やバッテリーにより、ニュートラル相当のPWMのDuty比が異なります、RCサーボの電源をONにした瞬間にタイヤが高速で回転する場合がありますので、ONにする前に必ず、タイヤが接地しないか(ステアを切っても接触が無い)、ケーブル巻き込みが無い、確認してください。
- ・高速回転しているタイヤに触れるとケガをする恐れがあります。回転部に触れないようにしてください。

4. キーボードを用いたジョグ運転テストを実行します。Wで加速、Sで減速(後退)、Aでステアを左に、Dでステアを右に、それぞれ一回押すごとに少しずつ変化させます。Nキーでステア/スピードともにニュートラルポジションに戻ります。

※後退側を入力する場合、タミヤのRCサーボシステムでは、前進動作後、後退に相当する指令値を入れるのみでは後退せず、ブレーキ動作となります。停止→後退→停止→後退、と2回、後退指令を入れて初めて後退を開始します

5. 画面に各 Ch の Duty 比が表示されますので、下記の時の Duty 比をメモしてください。

- ・ステアリングの中心
- ・ステアリングの右いっぱい
- ・ステアリングの左いっぱい
- ・およそ停止する範囲
- ・前進を開始する値（ニュートラルから小さくなる方向が前進方向です）
- ・後進を開始する値（一度後進に入れ、ニュートラル範囲に戻してから、値を徐々に増加させ値を探ります）
- ・前進方向で安全に走行できそうな速度（タイヤは空転しているので早く回転しますが全力走行のおよそ 10% 程度を目安にしてください）

これらの値から、ステアリングならびにスピードの指令値のニュートラルポジションを計算してください。

（例）ステアリング：見目でステアリングが中心となる位置

（例）スピード：前進開始時と後進開始時の平均値（あるいは少し後進側に近い値をお勧めします）

6. このように計算したニュートラルポジションの値を、key_calib.py プログラムの該当部分に書き込みます。

```
PWM_Hz = 70
PWM_NeutralStr = 10.00 #@70Hz
PWM_NeutralSpd = 10.48 #@70Hz
PWM_CruiseSpd = 10.48 - 1.4 #@70Hz # Cruising speed
```

※#マーク以下はコメントです。

※PWM_CruiseSpd はプログラム中では使っていませんが、安全に走行できるスピードに相当する PWM 値をメモしておくと便利です。

7. プログラムのパラメータを書き換えたあと、再度キャリブレーションを実行し、想定通りに動作するか確認します。もし想定通りであれば、キャリブレーション終了です。

※このとき、もし無線のキーボードをお持ちであれば、key_calib.py を実行後、有線の接続ケーブルをすべて外し、無線キーボードで操作すると走行しながらキャリブレーションできます。ただし、必ず一度リフトアップ状態でキャリブレーションして N キーで停止可能であることを確認してからにしてください。

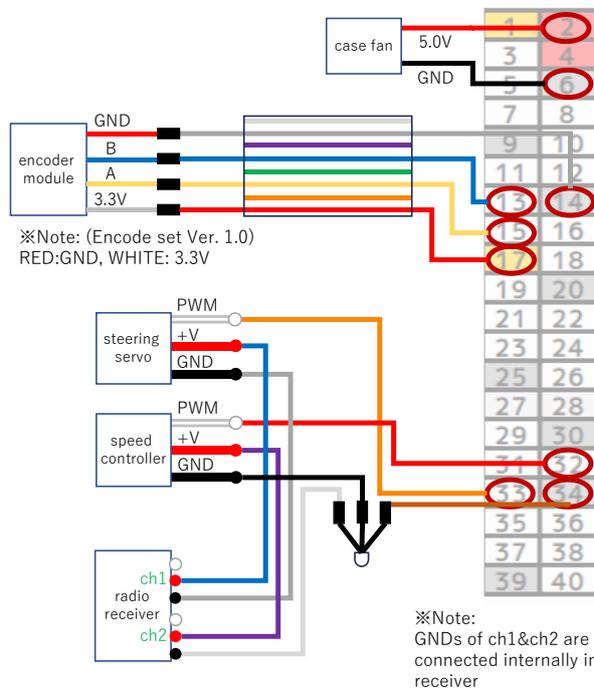
※バッテリーの減りや充電、バッテリー交換等によってニュートラルポジションが変化する場合があります。その場合は再度キャリブレーションを実施してください。

※動作がよさそうであれば、key_jog.py プログラムにより走行をテストできます。key_jog.py の上記 PWM パラメータを同様に書き替え、実行します。これは単に key_calib.py に比べジョグのステップが大きいだけのプログラムですが、無線キーボードを使って簡易的に遠隔走行できます。

7 納品時設定情報

納品型番	AD-RC Car TT02 Version 1.03 (ADRC-TT0203)	
シリアル番号	S#TT02-24** (車両右前のフレームに表示)	
Raspberry OS ログイン I D	adrcar	
パス	adpi2023 (出荷時自動ログイン設定)	
PWM 設定	ステアサーボ GPIO CH	Ch 13
	スピコン GPIO CH	Ch 12
	PWM 周波数	70Hz
	ステアサーボ Duty 比	10.00 % ± 1.0 (目安)
	スピコン Duty 比	10.48 % ± 2.0 (目安)
エンコーダカウンタ 設定	装着輪	左後方
	エンコーダ GPIO CH	A: Ch 22 B: Ch 27
	エンコーダ 歯数	36 / rot
	(参考) タイヤ直径	66.0mm

8 GPIO ピン接続



配線図 (ADRC-TT0225)

3.3V		1	2	5V	
SDA I2C	GPIO2	3	4	5V	
SCL	GPIO3	5	6	GND	
GPIO4		7	8	GPIO14	UART TXD
GND		9	10	GPIO15	UART RXD
GPIO17		11	12	GPIO18	
GPIO27		13	14	GND	
GPIO22		15	16	GPIO23	
3.3V		17	18	GPIO24	
MOSI SPI	GPIO10	19	20	GND	
MISO	SPI	GPIO9	21	22	GPIO25
SCKL	SPI	GPIO11	23	24	GPIO8
GND		25	26	GPIO7	
HAT	ID_SD	27	28	ID_SC	HAT
GPIO5		29	30	GND	
GPIO6		31	32	GPIO12	
GPIO13		33	34	GND	
GPIO19		35	36	GPIO16	
GPIO26		37	38	GPIO20	
GND		39	40	GPIO21	

Raspberry PI 4 ピン-GPIO 割り当て

9 その他注意事項

- ・Github : <https://github.com/Hiroyuki-Okuda/OpenMiniCarWorks> の wiki にも情報がありますので合わせてご参照ください。
- ・消費電力の観点から、CPU ケースファンは 3.3V ピンにつないでおりますが、ケースファンの動作が悪い場合があります。その場合は 5V ピンにて動作させてください。(ケースファンは 3.3V/5V 両対応)
- ・64bit 版の Raspberry PI OS Bullseye(V11)ではカメラ用の libcamera が動作しない場合が確認できています。問題がある場合は 32bit 版をお使いください。
- ・本体のバッテリー電圧によってサーボのニュートラルに対応する Duty 比が変化します、適宜キャリブレーションにより PWM の変化範囲をご調整ください。

10 改定履歴

- 2023/10/16 Version 1.00 初版作成
- 2023/11/12 8. Version 1.01 8.GPIO ピンの誤りを訂正
- 2024/1/15 8. Version 1.03 TT0203 シリーズ(V.1.03 機)用のピン接続を反映
6.2 RPlidar/animation.py の紹介を追加
9. その他注意事項を追記
- 2024/03/21 5.2 ケーブルの接続 にて電源のオート OFF に関する注意を追記
- 2025/01/19 Modeco WEB への QR コードを追加.
8. Version 2.05 TT0225 シリーズ用の配線図に更新

合同会社 MODECO
お問い合わせ : contact@modeco-ad.jp
<http://modeco-ad.jp/jpn/>