Roverの自動制御

構成

第1章.機体開発

1.1 roverを自動制御するのに使用した材料

今回roverを自動制御するのに使用した材料は以下である。

|  |
| --- |
| 機体：TAMIYA製rover（ブラシモーター付き） |
| モーター: ブラシモーター（機体に付属されているもの）サーボモーター×1（TAMIYA） |
| フライトコントローラ: Pixhawk1 |
| ESC: HobbyWing QuicRun WP 1060 Brashed ESC |
| バッテリー：Ni-MHバッテリー |
| 受信機：FUTABA R3008SB 2.4GHz T-FHSS AIR 空用 S.BUSテレメトリー対応受信機 |
| ブザー：Pixhawk1に付属されているもの |
| GPS: u-blox社GPS module with Compass(Pixhawk1に付属されているもの) |
| 安全スイッチ：Pixhawk1に付属されているもの |
| 配線：MXケーブル（Pixhawkと各機器をつなげるための配線）、ギボシ（モーター配線） |
| 無線モジュール: Telemetry Radio(Pixhawk1に付属されているもの) |
| その他：PC、ペンチ、六角レンチ、ドライバー 、養生テープ、SD-cardリーダ、  USB-Cable(Type-B) |

1.1.1 機体

　今回購入した機体は、図1.1.1.1のTAMIYA製のroverキットを購入した。価格はおよそ2万である。

機体を購入するにあたり、機体を走行させるための駆動モーターが含まれているか確認をする。万が一駆動モーターが含まれていない場合、別途で購入する。

1.1.2 モーター

　今回使用したモーターはTAMIYAのブラシモーター（図1.1.2.1）である。今回のブラシモーターはTAMIYA製のroverキットを購入した際に含まれていた物を使用した。

モーターのスペックは、

1.1.3 フライトコントローラ

　今回使用したフライトコントローラはPixhawk1である（図1.1.3.1）。アメリカの3D Robotics社が販売するフライトコントローラであり、Pixhawk1内にMPU6000という加速度センサ・ジャイロセンサ・地磁気センサが含まれている。

　注意として,Pixhawk1を購入する際に、MXケーブルやGPS,安全スイッチ、無線モジュール、ブザーが付属することを確認する。上記付属品がない場合、roverを手動、自動制御することができない。万が一付属品が含まれていない場合、別途で購入する。

　今回フライトコントローラを使用する理由は、roverの自動制御を行うためである。

自動制御を行うにあたり、roverに脳みそを与える必要があるため、脳みそ役であるフライトコントローラ（Pixhawk）を使用した。



1.1.4 ESC（Electronic Speed Controller）

　今回使用したESCは、HobbyWing QuicRun WP 1060 Brashed ESCである。（図1.1.4.1）

　購入した理由は、ブラシモーター専用の部分である。ESCを購入するにあたり、使用する駆動モーターの種類とスペックに気を付けるべきである。

駆動モーターの種類に関して、基本的に2種類しかなく、ブラシモーターかブラシレスモーターである。最近では、ほとんどの駆動モーターはブラシレスモーターが使用されている。しかし、今回購入したroverキットにブラシモーターが付属されていたため、ブラシモーター用のESCを購入した。

駆動モーターのスペックに関して、モーターに流れる電流と電圧範囲を知る必要がある。

1.1.5 バッテリー

　今回使用したバッテリーの種類は、Ni-MHである。（図1.1.5.1）

使用した理由に関して、使用したESCがLi-PoかNi-MHのバッテリー対応であったからである。

バッテリーの選択方法として、使用するESCやモーターを考慮することである。使用するESCがどのようなバッテリーを好むのか、使用するモーターにどのくらい電流や電圧を送る必要があるのかなどをそれぞれのマニュアルを見ながら考慮する。

バッテリーを購入するにあたり、コネクタの形状を確認する必要がある。

1.1.6 受信機

　今回使用した受信機は、FUTABA R3008SB 2.4GHz T-FHSS AIRである。

　使用した理由に関して、S.BUSテレメトリー対応であったからである。S.BUSテレメトリーを使用することで、Pixhawk1を送信機で操作することが可能になる。

1.1.7 ブザー

　今回使用したブザーは,Pixhawk1に付属されていた物である。（図1.1.7.1）

　ブザーを使用する理由に関して、Pixhawk1のセンサキャリブレーションなどの可否確認するために使用している。ブザーがあることにより、Pixhawk1の処理状況を音で把握することができるメリットがあるため、今回使用した。

1.1.8 GPS

　今回使用したGPSはu-blox社のGPS module withCompassを使用した。（図1.1.8.1）

　使用した理由は2つあります。

　1つ目は、Pixhawk1に付属されていており、Compassセンサも含まれていたからです。

　2つ目は、Pixhawk1でGPS moduleとしてサポートされているからです。Pixhawk1にサポートされていることにより、今後出てくるMission Plannerソフトを使用する際に、自動でGPSとして認識してくれます。

1.1.9 安全スイッチ

　今回使用した安全スイッチは、Pixhawkに付属されている安全スイッチを使用した。

　使用した理由に関して、Pixhawkに接続されたモーターを武装解除するためです。安全スイッチがない場合、モーターの武装解除ができず、モーターが動きません。また、安全スイッチ以外にも、Mission Plannerと送信機を用いることで,モーターの武装解除ができるらしい。

1.1.10 配線

　今回使用した配線は、MXケーブルである。MXケーブルはPixhawk1に付属されているものを使用した。

　使用した理由に関して、Pixhawk1が対応しているケーブルがMXケーブルであるからである。Pixhawkと各部品（モーターや受信機,センサ等）を接続するのに,MXケーブルを用いて接続する。

1.1.11 無線モジュール

　今回使用した無線モジュールは,Pixhawk1に対応したTelemetry Radioである。（図1.1.11.1）

　使用した理由に関して、Pixhawk1に付属されていた、Pixhawk1でサポートされていたからである。

1.2 機体の組み立て

1.2.1 フレームの組み立て（駆動モーターを取り付ける込み）

　機体：TAMIYA製のroverの組み立て作業（駆動モーターを取り付ける込み）を行う。

組み立て方法に関して、購入したroverに付属されている説明書をご覧していただきたい。Roverを組み立てるにあたり、苦労した点を以下である。

①：ベアリングの選択方法

②：デフギヤの組み立て

初めに①に関して、ベアリングを選択するにあたり、大きさがほとんど同じであり、組み立て時にどれを選択するのが難しい。

①の解決方法として、組み立てに使用するベアリングを説明書に記載されているベアリングと比較していただきたい。説明書に記載されているベアリングの大きさは実寸大と同じであるため、比較することでどのベアリングを選択すれば良いかある程度検討がつく。また、ベアリングの表面内の線の数を数え、説明書に記載されているベアリングの線と比較するというのもおすすめである。それでも、ベアリング選択が難しい場合の最終解決方法として、roverの組み立てに反映させ、ベアリングのサイズがあっているか確認する方法である。合っていなかった場合、別のベアリングで試し、合うまで同じ作業を繰り返すことである。

　②のデフギヤの組み立てに関して、デフギヤを両端のシャフトで固定する方法が難しい。難しい理由は、説明書がわかりにくい、実際に説明書通りに行っても固定できないことがあるからである。デフギヤを両端のシャフトで固定することができない場合、roverのモーターは回っても、roverの走行ができない問題が発生する。

　デフギヤを両端のシャフトで固定するための解決方法として、YouTubeなどの動画を参照する。”デフギヤの組み立て方”と調べると、たくさん出てくるため、調べてみることをお勧めする。特に、デフギヤを両端のシャフトでどのように固定しているのか注意しながら確認する。

　動画を見てもうまく固定がうまくいかない場合、デフギヤを両端のシャフトで押し付けながら、両端のシャフトを回転させる。デフギヤが固定されるまで、両端のシャフトを回転させる。ある程度、回転させたら、両端のシャフトを引っ張る。引っ張った際にデフギヤが落ちてしまう場合は、もう一度両端のシャフトを回転させる。両端のシャフトを引っ張り、デフギヤが固定された場合、デフギヤの組み立てが成功したことを意味する。ここで言うデフギヤが固定される定義として、両方のシャフトを引っ張った際に、デフギヤが外れないことである。

1.2.2 ESCを取り付ける

　ESCの取り付けにあたり、取り付け場所を把握する必要がある。基本的には、駆動モーター付近にESCを取り付けることをお勧めする。今回は図1.2.2.1のようにESCを設置した。

　次にESCとモーターの接続方法について説明する。接続にあたり図1.2.2.1を見ていただきたい。接続作業にあたり、ギボシとペンチを用意する。

1.2.3 サーボモーターを取り付ける

　サーボモーターの取り付けにあたり、ESCの時と同様、取り付け場所を把握する必要がある。

取り付け場所に関して、今回の場合、roverの説明書にサーボモーターの取り付け場所が書かれていたため、説明書通りに設置した。（図1.2.3.1）

1.2.4 フライトコントローラ(Pixhawk1)を取り付ける。

　機体の組み立てとモーターの取り付け後、フライトコントローラ（Pixhawk1）を取り付ける。

　今回取り付け場所に関して、場所がなかったため、3Dプリンタを用いて台を作り、下記図1.2.4.1のように台の上にroverの中央になるように養生テープで設置した。

取り付け時の注意点として、roverの前後方向を確認することである。Roverの前方向をPixhawk1の矢印の先端方向と合わせるようにする。

1.2.5 Pixhawkの配線をする。

　Pixhawkの配線方法に関して、図1.2.5.1に記す。今回Pixhawkと接続するものは下記となる。

①：ESCと駆動モーター

②：サーボモーター

③：バッテリー

④：無線モジュール

⑤：GPS module

➅：ブザー、安全スイッチ

⑦：受信機

初めに①のESCと駆動モーターの配線方法について説明する。

②のサーボモーターの配線について

③のバッテリーの配線について

④の無線モジュールの配線について

⑤のGPS moduleの配線について

➅のブザー、安全スイッチの配線について

⑦の受信機の配線について

1.2.6 機体組み立てにおいてのトラブルシューティング

　今回私が機体開発をしたうえでの問題点と原因、解決方法について下記に記す。

　問題点①：ベアリングの選択が難しい

　原因①：ベアリングの大きさがほとんど同じだから

　解決方法①：ベアリングの大きさを説明書と比較する。

2章システム開発（自動制御プログラム作成）

2.1 Mission Plannerの導入とPixhawkの初期設定

2.1.1 Mission Plannerの概要

Mission Plannerの概要について説明していく。

Mission Plannerはフライトコントローラを自動制御するためのソフトであり、今回フライトコントローラ（Pixhawk）を用いて、roverを自動制御するのに使用する。その他の使用目的として、フライトコントローラ内のセンサのキャリブレーションやセンサ情報や走行軌跡などをグラフとして可視化することができる。

2.1.2 Mission Plannerの導入

　Mission Plannerの導入に関して、下記URLからMisssion Plannerをインストールする。

<http://firmware.ardupilot.org/Tools/MissionPlanner/>

【MissionPlanner-1.3.70.msi】をクリックしてダウンロード⇒インストールする。万が一、上記のヴァージョンがない場合、違うヴァージョンをインストールする。（図2.1.2.1）

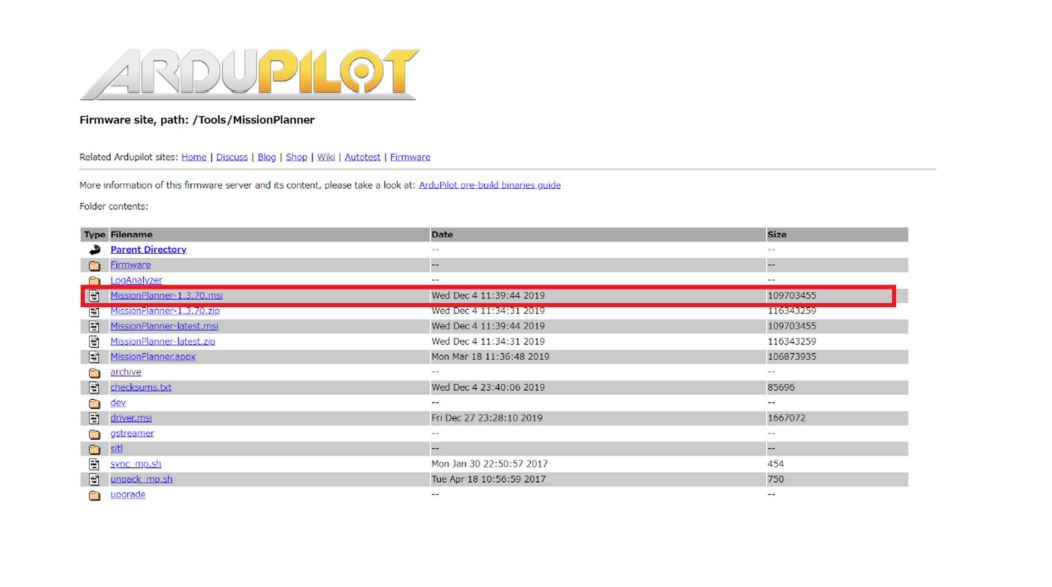


図2.1.2.1

【MissionPlanner-1.3.70.msi】をクリックしてダウンロードすると、下記図2.1.2.2のような画面が出てくる。Nextボタンを押し、Mission Plannerをインストールする。

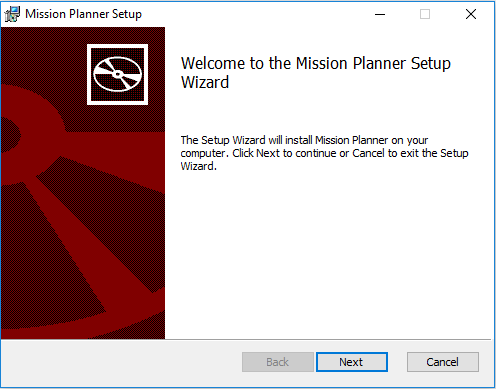


図2.1.2.2

　インストール完了後、PCでMission Plannerソフトを起動し、下記図2.1.2.3の画面が出れば、インストール完了である。

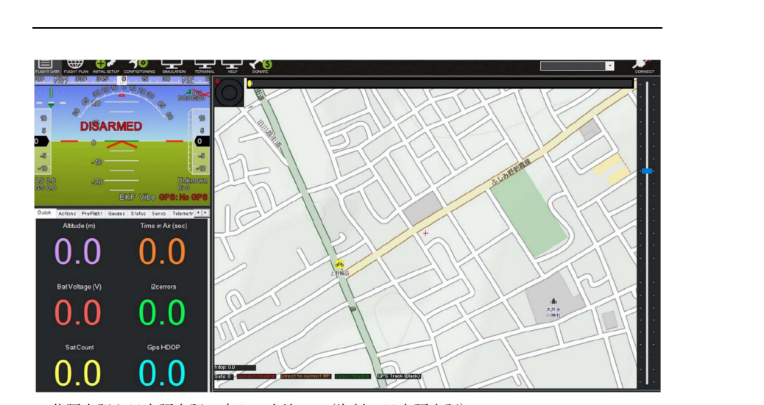


図2.1.2.3

2.1.3 Pixhawkの初期設定について

　Pixhawk1の初期設定について、Mission Plannerソフトを用いて初期設定を行っていく。

　Pixhawk1の初期設定を行うにあたり、Pixhawk1をUSB Cable TypeBを用いてPCと接続する。（図2.1.3.1）



　図2.1.3.1

次にMission PlannerとPixhawkの接続をする。図2.1.3.2のようにMission Planner画面の右斜め上にPixhawk1のPort番号と通信レートを入力するもしくは選択を行う。Port番号と通信レートの確認方法は、デバイスマネージャーのポート項目からPixhawk1のPort番号と通信レートを確認する。入力もしくは選択後、Connectボタンをクリックする。



図2.1.3.2

無時コネクトが成功すると図2.1.3.3のようなコネクトボタンの表記になる。



図2.1.3.3

次にファームウェアのインストールを行う。（図2.1.3.4）

Mission Plannerの一番上のメニューバーから初期設定項目を選択し、ファームウェアインストールを選択する。選択後、図2.1.3.4のように様々な機体写真が表示される。今回はroverの自動制御を行うため、図2.1.3.4の赤線のArduRoverをクリックし、ファームウェアをインストールする。ここで言うファームウェアは脳みそ役であり、今回rover用の脳みそをPixhawk1に入れているイメージで良いかと思われる。

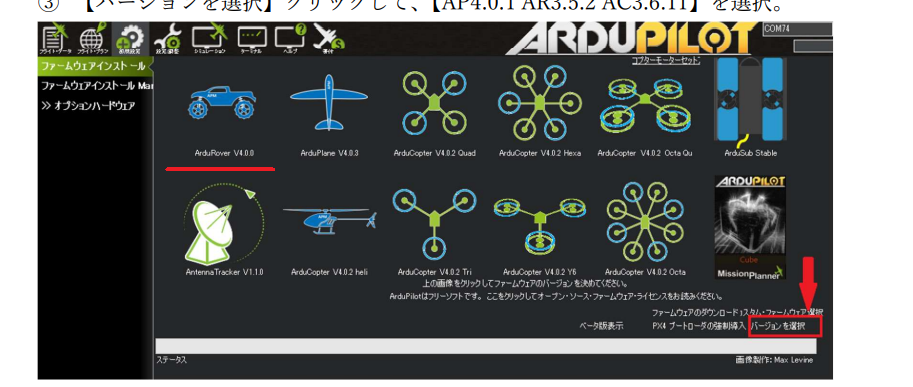


図2.1.3.4

ファームウェアインストール時、図2.1.3.5のような表記があった場合、図2.1.3.5のような対応をする。



図2.1.3.5

下記図2.1.3.6のような表記が出てきた場合、OKボタンをクリックすることで、ファームウェアのインストールが完了となる。以上でPixhawk1の初期設定が終了となる。



図2.1.3.6

2.2 Roverのパラメータ設定と書き込み

Roverのパラメータ設定方法について説明する。パラメータ設定する項目は以下である。

　　①：SERVO\_xFUNCTION(各モーターの役割・Ch設定)

②：SERIAL\_PROTOCOL(GPSや無線モジュールなどの役割設定)

③：SERIAL\_BAUD(GPSや無線モジュールなどの通信レート設定)

　初めにパラメータ設定場所から、Mission Planner画面の一番上のメニューバー「設定・調整」を選択。選択後、画面左側のフルパラメータリストを選択する。選択が完了すると図2.2.1のような表記となる。

　①のパラメータ設定から説明する。①のパラメータ設定に関して、Pixhawk1と接続しているモーターのChを入力する。

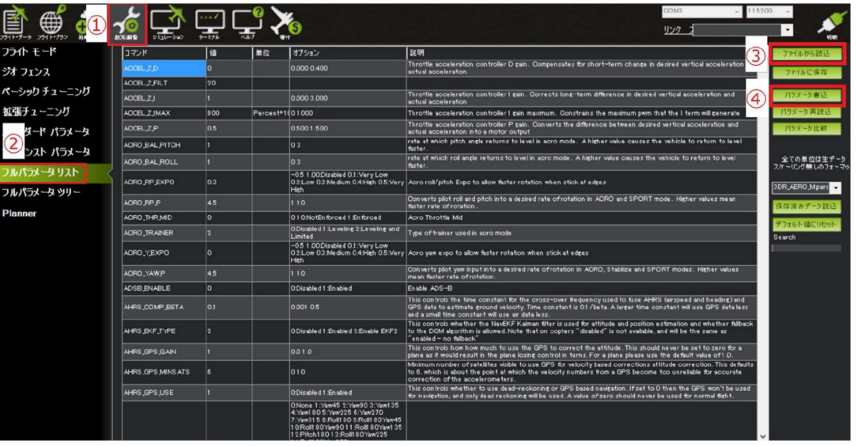


図2.2.1

2.3 Pixhawkのキャリブレーション

Roverの自動制御をPixhawk1で行うにあたり、Pixhawk1のキャリブレーションを行う必要がある。キャリブレーション項目に関しては、下記の通りである。それぞれのキャリブレーションを行う理由に関して、Pixhawk1が制御する機器をMission Plannerが把握、調整するためである。次の段落からそれぞれの項目のキャリブレーション方法について説明していく。

2.3.1加速度センサのキャリブレーション

初めに加速度センサのキャリブレーションから行っていく。下記図2.3.1.1を見ていただきたい。図2.3.1.1の画面にたどり着くためには、下記順番で画面を移動する必要がある。

①：Mission Plannerの起動時画面で、一番上のメニューバーから初期設定ボタンを選択。

②：画面左側の必須ハードウェアを選択する。

③：「必須ハードウェア」項目中にある「Accel Calibration」を選択する。

図2.3.1.1の画面から加速度センサのキャリブレーションを行っていく。キャリブレーションを行う前に、Pixhawk1表面の矢印向きが車体の前方と向きが合っているか確認する。

キャリブレーション手順に関して、、Calibrate Accelボタンをクリックする。クリック後下記図2.3.1.2のようなメッセージが表示される。メッセージの意味として、加速度キャリブレーションを行うにあたり、水平方向からキャリブレーションを行うということである。車体を水平方向に置き（置き方：図2.3.1.3）、数秒静止させ、緑色の次へボタンを押すという動作を6回行う必要がある。6回行う方向に関しては、図2.3.1.3を拝見していただきたい。方向は、水平、左、右、下、上、逆さである。特に方向に関して、Pixhawk1表面の矢印アイコンの向きを意識していただけると良いかと思われる。

　6方向の加速度センサキャリブレーションを行い、「Calibration successful」と表示されたら終了する。（図2.3.1.4）

　万が一キャリブレーション途中または終了時に、「Calibration FAILED」と表示された場合（図2.3.1.5）、最初からキャリブレーションをやり直す。文章だとどうしても伝わりにくい部分もあるため、YouTubeなどの動画ツールで、“Mission Planner Calibration”と調べていただけると様々な参考動画がでてくるため、そちらも確認していただけると良いかと思われる。

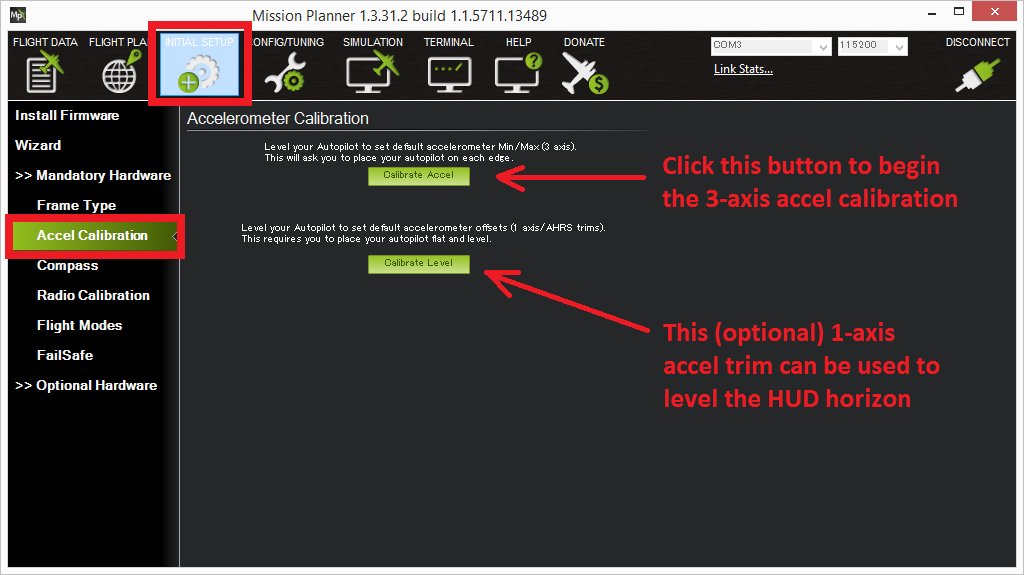


図2.3.1.1

  
図2.3.1.2





図2.3.1.3

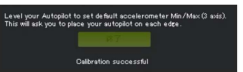


図2.3.1.4



図2.3.1.5

2.3.2コンパスセンサのキャリブレーション

　コンパスのキャリブレーションに関して、下記図2.3.2.1を見ていただきたい。図2.3.2.1の画面にたどり着くためには、下記順番で画面を移動する必要がある。

①：Mission Plannerの起動時画面で、一番上のメニューバーから初期設定ボタンを選択。

②：画面左側の必須ハードウェアを選択する。

③：「必須ハードウェア」項目中にある「Compass」を選択する。



図2.3.2.1

　キャリブレーション方法に関して、図2.3.2.1の④のFitness項目の「Default」を「Relaxed」に変更する。\* 「Default」だとキャリブレーションが完了しない可能性大

⑤のStartボタンをクリックし、Compassキャリブレーションを始める。車体を360°回転させるのだが、方向に関して、下記図2.3.2.2を見ていただきたい。それぞれ2.3.1の加速度センサキャリブレーションと同様に、水平方向から反時計方向に360°回転させる。回転させていくと、Mag1の緑色ゲージが動き出す。下記図2.3.2.3のように緑色ゲージが横幅いっぱいとなり、ゲージ横隣りに「SUCCESS」メッセージが表示された場合、作業終了である。車体の電源を抜きMission Plannerの再起動を行い、コンパスのキャリブレーションを終了する。

　万が一、キャリブレーション途中または終了時に、「Calibration FAILED」とゲージ横隣りに表示された場合、再度同じ作業を行う。

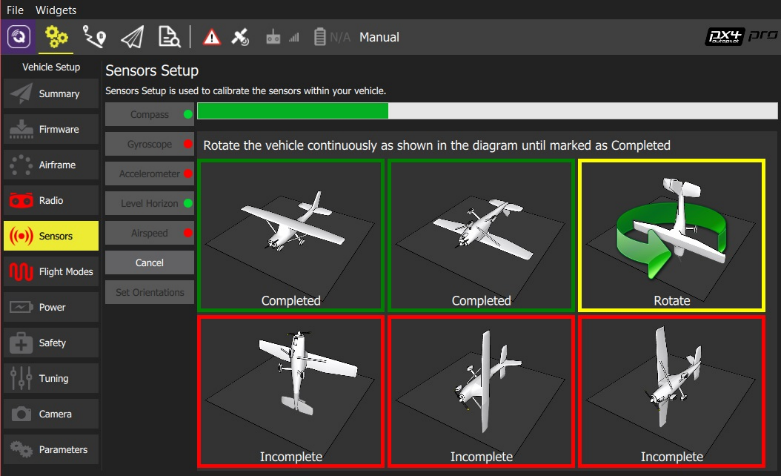


図2.3.2.2



図2.3.2.3

2.3.3送信機のキャリブレーション

送信機のキャリブレーションに関して、下記図2.3.3.1を見ていただきたい。図2.3.3.1の画面にたどり着くためには、下記順番で画面を移動する必要がある。

①：Mission Plannerの起動時画面で、一番上のメニューバーから初期設定ボタンを選択。

②：画面左側の必須ハードウェアを選択する。

③：「必須ハードウェア」項目中にある「Radio Calibration」を選択する。

送信機のキャリブレーション手順は以下である。

①：送信機の電源を入れる。

②：バッテリーをESCと接続し、車体の電源を入れる。

③：受信機のLEDが緑色になっていることを確認する。（\*赤色だった場合、送信機と受信機のバインド作業を行う。）

④：図2.3.3.1中のように緑色バーが表示されていることを確認し、③のラジオキャリブレーションボタンをクリックする。

⑤：送信機の各スイッチを動かし、緑色バーが変化することを確認する。

➅：図2.3.3.1中のの③ラジオキャリブレーションボタンをクリックする。

⑦：送信機の各レバーを上下左右360°動かし、スイッチ類を上下前後させる。

⑧：図2.3.3.2のようにそれぞれの緑色ゲージに赤線が出ていることを確認する。

⑨：「完了したらクリック」ボタンを押し、ラジオキャリブレーションを終了する。

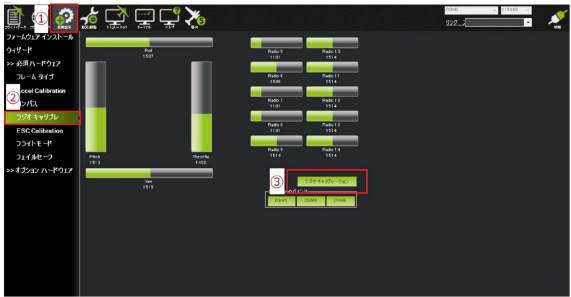


図2.3.3.1



図2.3.3.2

2.4 Servo Outputの設定

　Servo Outputの設定に関して、図2.4.1を見ていただきたい。画面#の列に番号が振られているが、これはPixhawk1のMain PWM chの番号を表す（図2.4.2）。Main PWM chは全て7個あり、Pixhawk1と各モーターを接続したchが図2.4.1のようにMission Plannerで緑色ゲージに反映されている(例：Pixhawk1のMain PWM 1,2chにサーボモーターと駆動モーターを接続している場合、図2.4.1の緑色ゲージは1,2に反映される)。今回の設定では（図2.4.1）、Pixhawk1のMain PWM ch1にサーボモーターが、Main PWM ch3に駆動モーターが接続されている。うまく接続されている場合、図2.4.1で緑色ゲージがch1,3に反映される。

　反映確認後、図2.4.1の赤線を見ていただきたい。赤線の部分に関して、functionを表しており、Pixhawk1に接続されているそれぞれのモーターの役割（機能）を設定することができる。今回では、2.6 roverの手動制御が含まれているため、function部分をRC PassTrowに変更し、送信機の電源を入れ、モーターが動作するか確認する。

　モーターの動作確認後、送信機のスティック役割を変更したい場合、Pixhawk1のMain chで接続しているモーターを変更する必要がある。それぞれのMain chにスティック役割が決まっており、以下である。

Main ch1 => Roll

Main ch2 => Pitch

Main ch3 => Throttle

Main ch4 => Yaw

Main ch5 => Switch

例として、送信機のPitchスティックで駆動モーターを動かしたい場合、Pixhawk1のMain ch2に駆動モーターを接続する必要がある。接続後、Servo Outputの設定で緑色ゲージが2に反映されていることを確認し、FunctionをRC passTrowに変更する。変更後、送信機のPitchスティックを動かし、駆動モーターの動作確認を行う。



図2.4.1



図2.4.2

2.5 roverとMission Plannerの無線接続

　roverとMission Plannerの無線接続にあたり、初めに準備物を説明する。準備物は、Pixhawk1で購入した、付属品図2.5.1の無線モジュール（Telemetry Radio）とMXケーブルである。

次にroverとMission Plannerの無線接続手順について以下の順番で行う。

①：MXケーブルを無線モジュールに取り付ける。（USB端子がなしの方に取り付け）

②：①：をPixhawk1のTELEM1 portに挿入する。

③：USB端子がついている無線モジュールをPCに接続する。

④：PCのデバイスマネージャーを用いて無線モジュールのPortと通信レートを確認。

⑤：roverのバッテリーの電源を入れる。

➅：PCでMission Plannerを起動する。

⑦：図2.5.2中に④：で確認したPortと通信レートを選択もしくは記入する。

⑧：図2.5.3のように接続完了マークが出た場合、完了となる。

注意点として、roverとMission Plannerの無線接続は、2.2 Roverのパラメータ設定と書き込みを行った後に必ず行っていただきたい。

　理由に関して、roverとMission Plannerの無線接続をするにあたり、使用する無線モジュールのパラメータをMission Plannerで設定する必要があるからだ。

注意点確認後、無事無線通信ができた場合、roverとMission Plannerの無線接続の説明を終了とする。



図2.5.1



図2.5.2



図2.5.3

2.6 roverの手動制御

2.6.1 Roverの手動制御セットアップ

Roverの手動制御の準備作業として、送信機の電源をONにし、roverの電源をONにする。その際に、roverに接続されている受信機のLEDが緑色になっているか確認をする。万が一受信機のLEDが赤色だった場合、バインド作業をもう一度行う必要がある。また、Mission Plannerを起動し、図2.5.2のように無線モジュールの接続場所と通信レートを入力または選択をし、コネクトボタンを押す。図2.5.3のようにDISCONNECTボタンに切り替われば、接続が完了である。

それぞれの電源をONにし、Mission Planner接続後、Roverの手動制御のセットアップ場所を把握する必要がある。セットアップ場所はMission PlannerのServo Outputである。セットアップ場所(図2.6.1.1)は、下記の手順で把握できる。

①：Mission Plannerの起動時画面で、一番上のメニューバーから初期設定ボタンを選択。

②：画面左側の必須ハードウェアを選択する。

③：「必須ハードウェア」項目中にある「Servo Output」を選択する。

セットアップ場所が把握できた場合、roverの手動制御セットアップを行っていく。セットアップ方法は図2.6.1.1のServo Output項目の赤線部分であるfunctionを変更することである。図2.6.1.1では、Pixhawk1のMain ch1,3にサーボモーターと駆動モーターを接続している。接続がうまくいっている場合、図2.6.1のように#1.3に緑色ゲージが反映されている。# 1,3のfunction部分をRC PASS THROWに変更し、送信機のスティックを動かす。送信機のスティックを動かし、roverが動作した場合、セットアップが終了となる。万が一動作しない場合、Pixhawk1のMain chの配線が合っているかどうか、また、Mission Plannerの再起動と再度セットアップ作業を行うことで解決する。



図2.6.1.1

2.6.2 なぜ手動制御項目をいれたのか？

　理由に関して、二つある。

　一つ目は、自動制御中のroverの不意な暴走を防ぐためである。万が一の自動制御中に、自分の設定した方向に進まない、変な方向に進む場合には、一旦手動制御に変更し、原因を突き止めるようにする必要がある。

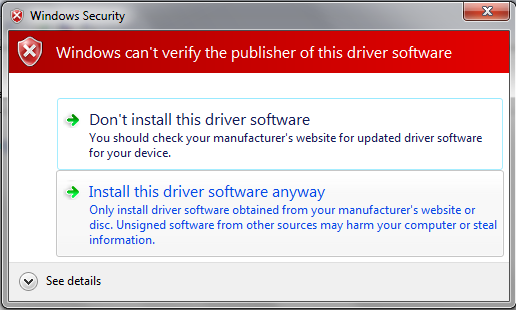
　二つ目は、roverの自動制御のハードルを下げるためである。いきなり自動制御を始めてしまう場合、どのようにroverを自動制御したら良いかわからない問題が発生する。この問題を解決するにあたり、手動制御を間に挟むことで、人間が送信機を用いてroverを制御することで、自動制御するための要素をピックアップしやすくすることが可能となる。

　したがって上記の理由から手動制御を自動制御manualの中に入れている。

2. システム開発でのトラブルシューティング

システム開発を行う上での、トラブルシューティングのまとめを下記に示す。

問題点①：Mission Plannerインストール時、下記画像が出てインストールできない。



原因①：PCのセキュリティー問題

解決方法①：Install this driver software anywayボタンを押すことでインストールできる。

問題点②：PCがMacでMission Plannerがインストールできない。

原因②：Mission Plannerはwindowsでしか用意されていないため。

解決方法②：代わりにQqround Control(QGC)ソフトを使用する。

3章roverの自動制御

3.1作業環境構築

　Roverを自動制御する上での環境構築について説明していく。はじめに自動制御する場所に関して、なるべく障害物がない広場や公園を選択する。

3.2 自動制御手順

　自動制御手順に関して以下である。

①：送信機とroverの電源をON、Mission Plannerを起動する

②：Servo Outputの設定

③：モーターの回転方向テスト

④：Waytpointの設定

⑤：ARMED確認

➅：安全スイッチをONにする。

①に関して、roverの自動制御をするにあたり、送信機の電源をONにする。次にroverのバッテリーをESCに接続し、電源をONにする。接続後、Pixhawkに接続されている受信機の色が緑色（バインド完了）になっていることを確認する。確認後、Mission Plannerを起動する。無線モジュールをPCに接続し、Misssion Plannerの接続ボタンを押す（詳細部分：2.5）。接続ボタンが緑色になったことを確認できたら、①の作業を終了する。

②に関して、自動制御用のServo output設定を行う。設定場所に関しては、2.4 Servo Outputの設定を見ていただきたい。自動制御用のServo output設定にするためには、Function部分を変更する必要がある。図3.2.1では、Pixhawk1のMain Output1にステアリングサーボモーター、Main Output3に駆動モーターが接続されている。うまくモーター接続出来ている場合、図3.2.1のように#1,3に緑色ゲージが反映されている。#1ステアリングサーボモーターのfuntion項目をGroud Stearingにし、#3の駆動モーターのfunction部分をThrottleに変更する。変更後、エラーが出なければ②作業を終了する。



3.2.1 自動制御用のServo output設定

　③に関して、モーターの回転方向テストを行う。モーターの回転方向テスト画面場所に関して、以下である。

①：Mission Plannerの起動時画面で、一番上のメニューバーから初期設定ボタンを選択。

②：画面左側の必須ハードウェアを選択する。

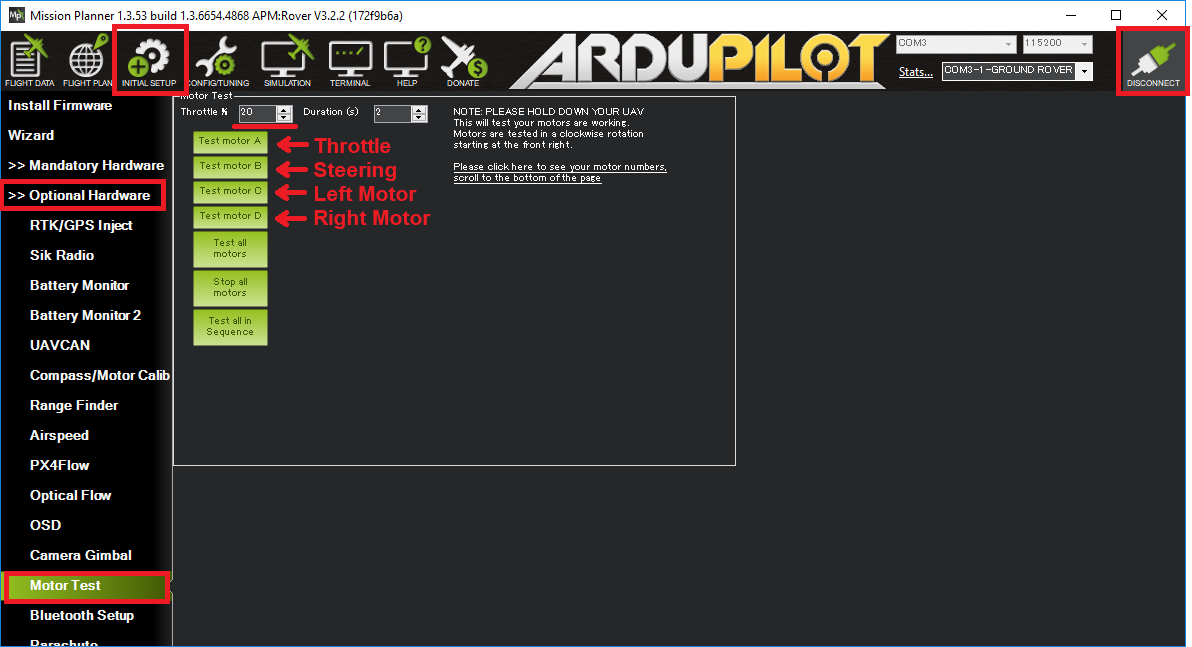
③：「Optional Hardware」項目中にある「Motor Test」を選択する。

モーターの回転方向テストの項目設定に関して、Throttle%項目は20程度にする。Throttle%は送信機のThrottleスティックを20%動かしている状態ととらえる。Throttle%を上げることで、モーター回転数が上がることに値する。Duration項目は、回転する期間を表し、デファクト状態2のままにする。次にTest motor AがThrottle(駆動モーター)、Test motor Bが(Steering ステアリングサーボモーター)の回転を意味する。

モーターの回転方向テストに関して、Test motor Aボタンを押し、駆動モーターの回転方向を確認する。回転方向が逆の場合、図3.2.1自動制御用のServo output設定画面で、function項目Throttleの左側のReverseボタンにチェック入れる。

　Reverseボタンにチェック後、もう一度モーターの回転方向テストを行い、回転方向が変更されたのを確認する。

　次にTest motor Bボタンを押し、ステアリングサーボモーターの回転方向を確認する。確認後、回転方向を変更したい場合は、上記の同様作業を行い、Stering項目のReverse項目にチェックを入れる。その後、もう一度モーターの回転方向テストを確認する。

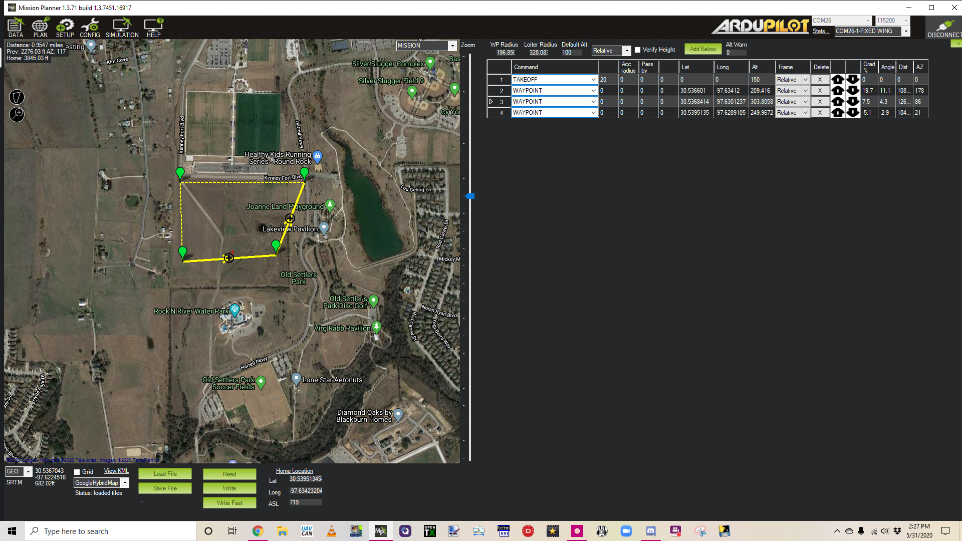


3.2.2　モーターの回転方向テスト

④：Waytpointの設定に関して、設定場所は、Mission Planner起動時画面から、メニューバーからPlanボタンをクリックする。クリック後図3.2.3のようにフライトプラン画面が出現する。

次にwaytpointを指定する。waytpointとは、roverを自動制御する上でroverが通る道（point）である。指定方法に関して、図3.2.4のように地図上にポインタークリックすると、watypointが出現する。Waytpointをマウスで長押しし、roverを自動制御したい道まで操作する。Waytpointを指定すると、右側にwaytpointが反映されているか確認する。Roverの自動制御経路ができたら、図3.2.4の書き込み緑色ボタンを押し、waytpointの設定を終了する。

今回例として、roverの自動制御するにあたり、waytpointを5個使用し、roverを1周自動制御するようにした。図3.2.4ではwaytpointが4つしかないように見える。しかし、最後の一つに関して、Return\_To\_Launchを使用している。意味としては、roverが1周後、home pointに戻ることである。Return\_To\_Launchに関しては、地図上には表示されない部分なので無視していただいてよい。



3.2.3 フライトプラン画面



3.2.4 waypoint指定方法

⑤：ARMED確認に関して、自動制御するにあたり、動力をONにする儀式である。（ブザーを付けるとピーという音が鳴ります。）Mission Plannerのフライトデータ画面中のアクションタブをクリックして、以下の作業を行う。

[Arm/Disarm]と書かれている緑色ボタン（図3.2.5）を押し、画面がDISARMEDからARMEDに変化したか確認をします。エラーが出た場合、Armできない原因を探す。原因詳細はトラブルシューティングを確認する。また、Arm/Disarmボタンをもう一度押すとDISARMEDになり、その後駆動モーターが回らなくなります。



3.2.5 Arm/Disarm確認

➅：安全スイッチをONにしましょう。安全スイッチは駆動モーターの武装解除するためのものです。安全スイッチがないとモーターは駆動しません。安全スイッチを数秒クリックし、赤色の点滅から赤色点灯になった場合、武装解除成功です。

3.3 roverの自動制御実装中の確認

　Roverの自動制御中で以下の点の確認をする。

* Roverの前身方向や左右方向が正しいかどうか
* Waytpoint1に向かっているかどうか
* Waytpoint1を超えたら次にWaytpoint2に向かっているかどうか
* 目的地付近でroverが停止するかどうか

　上記の確認が取れない場合はRover自動制御でのトラブルシューティングを拝見し、原因と解決策を把握していただきたい。

3.4 roverの自動制御後

　Roverの自動制御終了後、roverの電源を切る。電源を切った際に、Mission Plannerのヘッダー部分である接続ボタンが赤色になったのを確認する。（赤色の場合roverの接続が切れたことを意味する。）

次にroverの部品などの破損がないか確認をする。万が一破損等があった場合、その場で修復できるかどうか判断する。修復可能な場合、その場で修理する。もう一度自動制御を行う場合は、3.2の作業をもう一度行う。

3.5 目的地到着時(WP)でのアクション実行

　3.5の目的地到着時(WP)でのアクション実行に関して、Rover自動制御時に目的地に到着した場合、Pixhawkに接続されているサーボモーターを動作させることを行った。

　方法に関して、図3.5.1を見ていただきたい。Mission PlannerのFlIGHT PLAN画面で、Waytpointを設定時のCommand選択からアクションを実行するこができる。

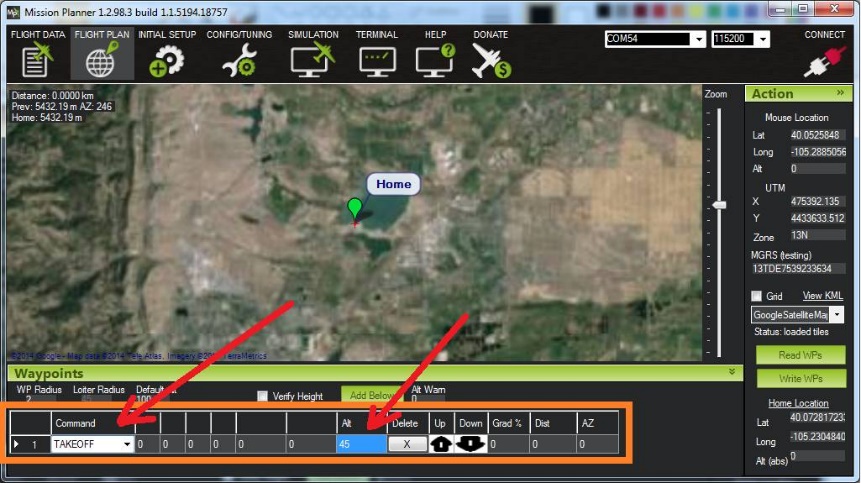


図3.5.1　目的地到着時のアクション設定場所

　今回のアクション設定内容に関して、サーボモーターを動作させることなので、図3.5.1から下記手順でアクション設定内容を把握し、設定する。

1. CommandをDO\_SET\_SERVOにする

2. Set Noをサーボモーターが接続されている出力チャンネルに設定

3. サーボに出力するPWM値を設定する。

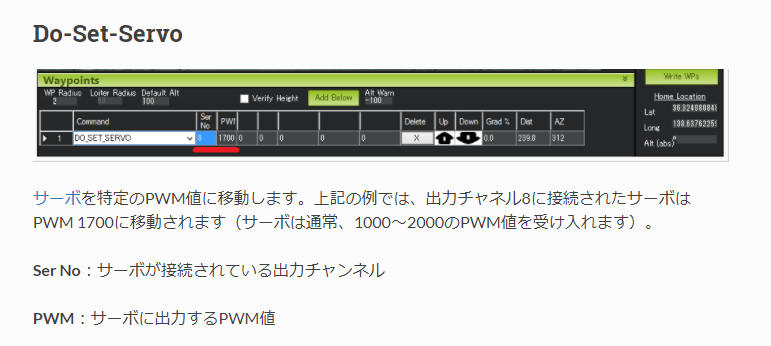


図3.5.1 アクション設定内容

3.6 Rover自動制御でのトラブルシューティング

　この章では、rover自動制御でのトラブルシューティングを問題、原因、解決方法という形でまとめた。

問題点1:

Auto modeでroverを自動制御しようとすると、少し走り出して、止まってしまう問題

原因1：

GPSの情報が信頼できない問題・車スピードが遅すぎる問題

解決方法1:

STEER2SRV\_MINSPDを2にする。

<https://ardupilot.org/rover/docs/tuning-steering-and-navigation-for-a-rover.html>

問題点2:

Auto modeでroverを自動制御しようとすると、目的地とは全然違う方向に行ってしまう問題。

原因２：

コンパスのキャリブレーションがうまくいっていない。

　ステアリングサーボが逆になっている。

　人工衛星数が足りていない。（Mission Plannerのゲージで確認）

　ウェイトポイントが近すぎる。

解決方法2:

コンパスのキャリブレーションを行う。

　ステアリングサーボをモーターテストでリバースに☑マークする。

問題点3:

自動制御でのローバーの軌跡がwaytpointとかなりずれてしまう問題

原因調査3:

ArduPilotのroverドキュメントから情報を探る

　rover ardupilot auto modeで検索し、<https://ardupilot.org/rover/docs/auto-mode.html>

から情報収集する。ナビゲーション調整からヒントが得られる可能性あり。

　また、<https://ardupilot.org/rover/docs/tuning-steering-and-navigation-for-a-rover.html>

から情報収集する。

原因調査結果３：

<https://ardupilot.org/rover/docs/auto-mode.html>　でわかったこと。

[NAVL1\_PERIOD](https://ardupilot.org/rover/docs/parameters.html#navl1-period)パラメータを0.5ずつ上げることでwaypointからずれにくい。

[NAVL1\_DAMPING](https://ardupilot.org/rover/docs/parameters.html#navl1-damping)パラメータを0.05ずつ刻むことで短いコースを走れる。

[WP\_SPEED](https://ardupilot.org/rover/docs/parameters.html#wp-speed)は自動制御の速度を示す。

[WP\_RADIUS](https://ardupilot.org/rover/docs/parameters.html#wp-radius)は車両が次のウェイポイントに向かって曲がり始める可能性のあるウェイポイントの前の距離を指定くれるもの。このパラメータを低く設定しすぎると、ウェイポイントに非常に近づくため、車両が役に立たない方向転換を試みる可能性があります。

[WP\_OVERSHOOT](https://ardupilot.org/rover/docs/parameters.html#wp-overshoot) 設定が低すぎると、ウェイポイント間のラインからその位置がずれてしまう

[ATC\_STR\_ANG\_P](https://ardupilot.org/rover/docs/parameters.html#atc-str-ang-p)　値を高くすることで方位誤差を修正してくれる。

<https://ardupilot.org/rover/docs/tuning-steering-and-navigation-for-a-rover.html>

　SPEED\_TURN\_GAIN回転中にローバーがどれだけ減速するかを制御します。100から少しずつ下げていく。

* STEER2SRV\_P = 2
* TURN\_MAX\_G = 1
* NAVL1\_PERIOD = 6
* SPEED\_TURN\_GAIN = 100
* CRUISE\_SPEED = 2
* CRUISE\_THROTTLE = 20　で設定してみる。

結論：STEER2SRV\_MINSPDが低すぎる可能性大！

有効原因：

STEER2SRV\_MINSPDが小さすぎる問題。低速に自動制御していることが問題。

解決方法：

https://ardupilot.org/rover/docs/tuning-steering-and-navigation-for-a-rover.html

　GPSの情報が信頼できない場合、STEER2SRV\_MINSPDを2にあげる必要があり。

　万が一、自動制御の速度を上げてもダメな場合、L1ナビゲーションチューニングまたはステアリングコントローラーチューニングを調整する必要があり。

原因3: ATC\_STR\_RAT\_FFが高すぎるから

解決方法3: ATC\_STR\_RAT\_FFのパラメータ調節（回転速度調節）

問題点4: roverの自動モード後、roverの動きが予期せぬ円運動状態になっている時

原因4:不明

解決方法4:Radio　キャリブレーションのroll barをリバースにする。

　　　　　MIS\_RESTARTは1で

初めてローバーを走らせる時のチェックリスト

<https://ardupilot.org/rover/docs/rover-first-drive.html>

問題点5:自動モード後、safety switchを押しなさいというエラー後、safety switchを押し、armedすると、roverが動かない問題。

原因5:自動手順を間違えたことが原因。

解決方法5:一旦Mission Plannerを再起動、または、PCを再起動させる。

問題点6:

　Roverの自動制御に関して、最終WPを通った後、出発点に戻らない問題点。

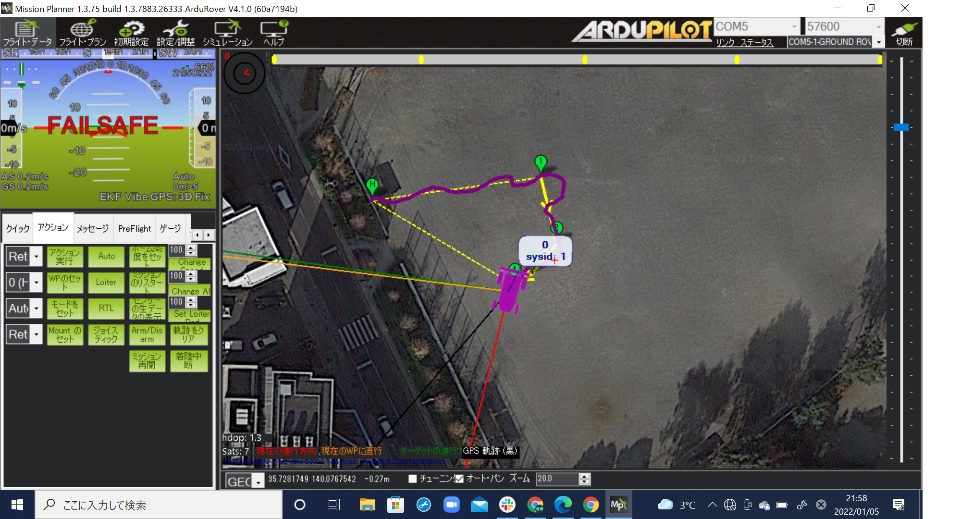


図6.1 出発点に戻らない問題点

原因予測6

　WP作成時にReturn\_to\_Launchの設定に問題あり？

解決方法6

　Return\_to\_Launchの設定を他の方法で行ってみる。

Roverの自動制御

構成

　1. 機体開発

1.1 roverを自動制御するのに使用した材料

1.1.1 機体

1.1.2 モーター

1.1.3 フライトコントローラ

1.1.4 ESC（Electronic Speed Controller）

1.1.5 バッテリー

1.1.6 受信機

1.1.7 ブザー

1.1.8 GPS

1.1.9 安全スイッチ

1.1.10 配線

1.1.11 無線モジュール

1.2 機体の組み立て

　1.2.1 フレームの組み立て（駆動モーターを取り付ける込み）

1.2.2 ESCを取り付ける

1.2.3 サーボモーターを取り付ける

1.2.4 フライトコントローラ(Pixhawk1)を取り付ける。

1.2.5 Pixhawkの配線をする。

1.2.6 機体組み立てにおいてのトラブルシューティング

2章システム開発（自動制御プログラム作成）

2.1 Mission Plannerの導入とPixhawkの初期設定

　2.1.1 Mission Plannerの概要

　2.1.2 Mission Plannerの導入

　2.1.3 Pixhawkの初期設定について

2.2 Roverのパラメータ設定と書き込み

2.3 Pixhawkのキャリブレーション

2.3.1加速度センサのキャリブレーション

2.3.2コンパスセンサのキャリブレーション

2.3.3送信機のキャリブレーション

2.4 Servo Outputの設定とモーターの回転方向テスト

2.4.1 Servo Outputの設定

2.4.2 モーターの回転方向テスト

2.5 roverの無線接続

2.6 roverの手動制御

　Roverの手動制御セットアップ

　なぜ手動制御項目をいれたのか？

3章roverの自動制御

作業環境構築

自動制御の手順

自動制御後

Rover自動制御でのトラブルシューティング

運転免許があれば、仕事ができる。

価値観の違い

物理量の変化率を考える。

ベクトル解析

発散（入り口と出口の現象から中身の現象を予測する。）

回転