

9 軸 IMU/AHRS haya_imu マニュアル v3.2

新規作成日：2023 年 4 月 26 日

内容

1. 使用環境	3
2. 使用手順	3
2.1 ROS/ROS2 パッケージのインストール	3
2.2 USB での接続	4
2.3 USB シリアルポート番号の固定	4
2.4 パラメータの確認	5
2.5 ROS ローンチ	5
2.6 Topic、tf の確認	7
2.7 初期バイアスの測定(キャリブレーションモード)	8
2.8 (オプション)ROS2 における DDS の切り替え	10
3. LED 表示	10
4. 追加説明	10
5. 参考情報	11
5.1 主な規格	11
5.2 ROS/ROS2 パッケージ	11
6. ファームウェアの書き換え	13
6.1 書き換えモードへの切り替え	12
6.2 書き換え手順	12
7. リビジョン履歴	13

0. はじめに

haya_imu v3.xはMicrochip製Cortex-M4(クロック周波数120MHz)、TDK製6軸ICM-42688、Memsic製高精度3軸AMR方式地磁気センサMMC5983MAの実装により、通常出力モード、デモンストレーションモード、キャリブレーションモード(初期バイアス測定モード)、6軸フュージョン回転ベクトルクォータニオン、9軸フュージョン回転ベクトルクォータニオン、3軸オイラー角は最大1000Hz(Best Effort)まで同時出力可能な9軸IMU/AHRSです。本マニュアルは、haya_imuの扱い方を示す資料とします。



1. 使用環境

- UBUNTU 20.04、18.04、16.04、22.0(ROS2)推奨
- ROS noetic、melodic、kinetic、ROS2 humble 推奨

2. 使用手順

2.1 ROS/ROS2 パッケージのインストール

※ROS

```
$cd ~/catkin_ws/src
```

```
$git clone https://github.com/soarbear/haya\_imu\_ros2.git
```

```
$cd ~/catkin_ws
```

```
$catkin_make
```

※ROS2

```
$cd ~/ros2_ws/src
```

```
$git clone https://github.com/soarbear/haya\_imu\_msgs.git
```

```
$git clone https://github.com/soarbear/haya\_imu\_ros2.git
```

```
$cd ~/ros2_ws
```

```
$colcon build --packages-select haya_imu_msgs haya_imu_ros2
```

2.2 USB での接続

haya_imu を対向装置へ USB で接続できた場合、赤 LED が常時点灯することを確認できます。

2.3 USB シリアルポート番号の固定

※ROS

- 以下ルールの作成によって、ttyACM_haya に固定します。

```
$chmod +x ~/catkin_ws/src/haya_imu_ros/script/create_rules.sh
```

```
$~/catkin_ws/src/haya_imu_ros/script/create_rules.sh
```

```
$sudo udevadm control --reload-rules && sudo udevadm trigger
```

- また、以下ルールの削除によって、固定した USB シリアルポート番号を解除します。

```
$chmod +x ~/catkin_ws/src/haya_imu_ros/script/delete_rules.sh
```

```
$~/catkin_ws/src/haya_imu_ros/script/delete_rules.sh
```

```
$sudo udevadm control --reload-rules && sudo udevadm trigger
```

※ROS2

- 以下ルールの作成によって、ttyACM_haya に固定します。

```
$chmod +x ~/ros2_ws/src/haya_imu_ros2/script/create_rules.sh
```

```
$~/ros2_ws/src/haya_imu_ros2/script/create_rules.sh
```

```
$sudo udevadm control --reload-rules && udevadm trigger
```

- また、以下ルールの削除によって、固定した USB シリアルポート番号を解除します。

```
$chmod +x ~/ros2_ws/src/haya_imu_ros2/script/delete_rules.sh
```

```
$~/ros2_ws/src/haya_imu_ros2/script/delete_rules.sh
```

```
$sudo udevadm control --reload-rules && udevadm trigger
```

2.4 パラメータの確認

params.yaml に載ってあるパラメータの値を確認して、目的、必要に応じて変更します。

2.5 ROS ローンチ

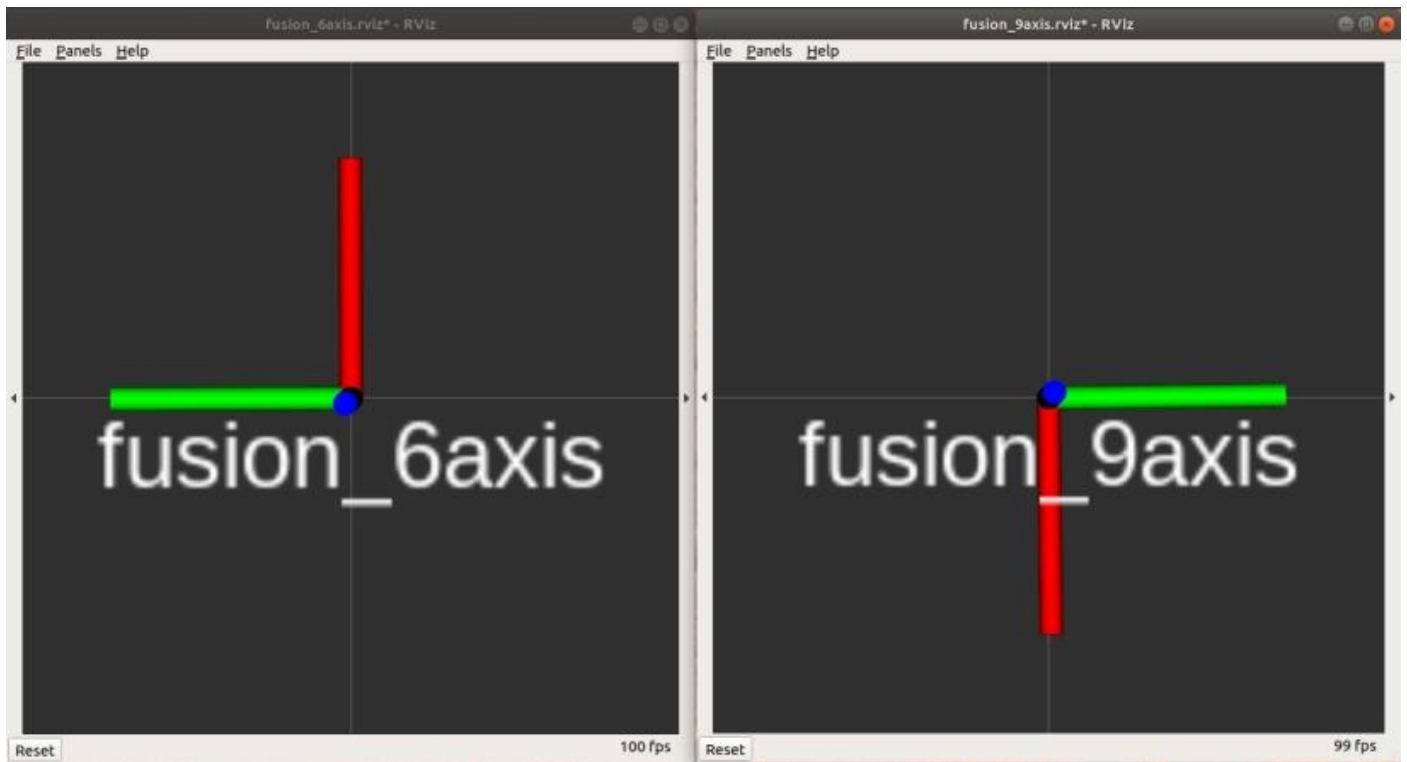
※ROS

- 通常出力モード、キャリブレーションモードに設定された場合

```
$roslaunch haya_imu_ros haya_imu.launch
```

- デモンストレーションモードに設定された場合

```
$roslaunch haya_imu_ros haya_imu_demo.launch
```



※ROS2

- 通常出力モード、キャリブレーションモードの場合

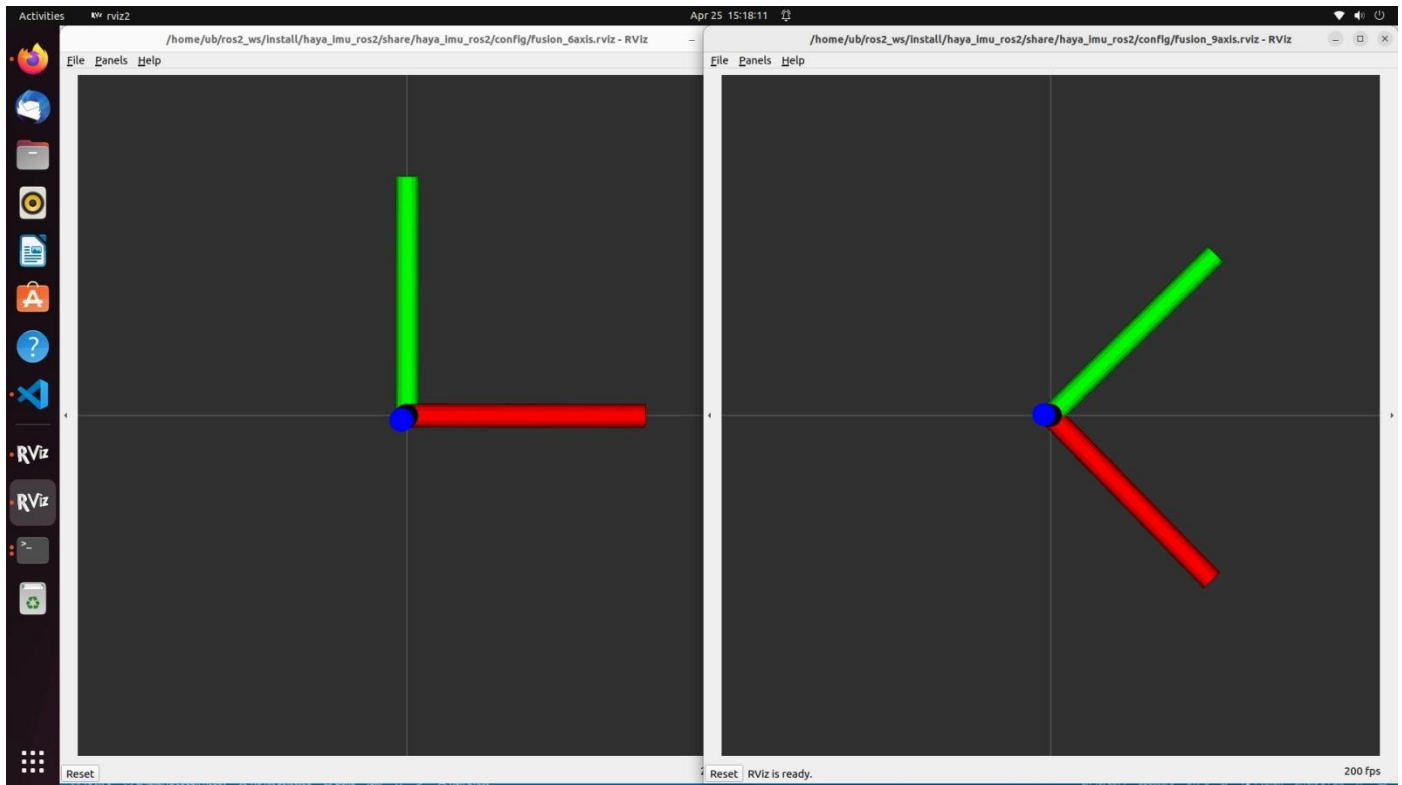
```
$source ~/ros2_ws/install/setup.bash
```

```
$ros2 launch haya_imu_ros2 haya_imu_launch.py
```

- デモンストレーションモードの場合

```
$source ~/ros2_ws/install/setup.bash
```

```
ros2 launch haya_imu_ros2 haya_imu_demo_launch.py
```



2.6 Topic、tfの確認

※ROS

- メッセージのタイプ

imu_data(Message: haya_imu_ros/ImuData)、通常出力モード、キャリブレーションモード用

tf(Message: geometry_msgs/TransformStamped)、デモンストレーションモード用

- Topicのデータを確認する例

```
$rostopic echo imu_data
```

- Topicの出力レートを確認する例

```
$rostopic hz -w 100 imu_data
```

※ROS2

- メッセージのタイプ

imu_data(Message: haya_imu_msgs/msg/ImuData), 通常出力モード、キャリブレーションモード用

tf(Message: geometry_msgs/msg/TransformStamped), デモンストレーションモード用

- Topic のデータを確認する例

```
$source ~/ros2_ws/install/setup.bash
```

```
$ros2 launch haya_imu_ros2 haya_topic_echo_launch.py
```

もしくは、\$ros2 topic echo /imu_data

- Topic の出力レートを確認する例

```
$source ~/ros2_ws/install/setup.bash
```

```
$ros2 launch haya_imu_ros2 haya_topic_hz_launch.py
```

もしくは、\$ros2 topic hz -w 1000 /imu_data

2.7 初期バイアスの測定(キャリブレーションモード)

src/config/params.yaml の imu_mode: 10(キャリブレーションモード)に設定して、ジャイロスコープ→

加速度センサ→地磁気センサの順に、以下要領のとおり初期バイアスの測定を行います。

※ ROS

```
$roslaunch haya_imu_ros haya_imu.launch
```

```
$rostopic echo /imu_data
```

※ ROS2

```
$source ~/ros2_ws/install/setup.bash
```

```
$ros2 launch haya_imu_ros2 haya_imu_launch.py
```

※ ステップ 1: ジャイロスコープ 初期バイアスの測定の要領

上記 roslaunch をリリースして、IMU を動かさず、平らに 30 秒ほど置いて、**calibration_index** は「48」

に変わると、ジャイロスコープ 初期バイアスの測定は完了とします。

※ ステップ 2: 加速度センサ 初期バイアスの測定の要領

XYZ 軸のプラス方向にそれぞれ-1.0G をかけて測定を行ってください。

- ・ X 軸のプラス側を上に向けて

加速度出力の `linear_acceleration` の第 1 引数がほぼ 1.0 になると、10 秒ほど静止状態キープ

- ・ Y 軸のプラス側を上に向けて

加速度出力の `linear_acceleration` の第 2 引数がほぼ 1.0 になると 10 秒ほど静止状態キープ

- ・ Z 軸のプラス側を上に向けて

加速度出力の `linear_acceleration` の第 3 引数がほぼ 1.0 になると 10 秒ほど静止状態キープ

3 軸プラス方向を上に向けることを 1 組として、`calibration_index` は「60」に変わると、加速度センサの初期バイアスの測定は完了とします。また、1 組で 3 に変わらないと、もう 1 組を追加する場合があります。

如何に 1.0 または-1.0 に近けることは、精度の良さに繋がります。

※ ステップ 3: 地磁気センサ 初期バイアスの測定の要領

「8 字回し」： XY 平面、YZ 平面、XZ 平面すなわち 3 次元空間の各平面（順番関係なし）において、各平面ずつ、8 字の形で haya_imu を 10 回程回します。8 字の幅は 70-100cm 程とします。`Calibration_index` は「63」に変わると、地磁気センサの初期バイアスの測定は完了とします。測定完成時の画面は下図に示します。

```

アクティビティ 端末 ▾ 2023-04-26_17:21:14 en  🔊 🔍 🔌
ub@ubuntu18: ~
---
header:
  seq: 63007
  stamp:
    secs: 1682497274
    nsecs: 735146404
  frame_id: "imu_link"
angular_velocity:
  x: -0.0412902832031
  y: -0.0370025634766
  z: -0.0153961181641
angular_velocity_covariance: [0.000784000000000001, 0.0, 0.0, 0.0, 0.000784000000000001, 0.0, 0.0, 0.0, 0.000784000000000001]
linear_acceleration:
  x: -0.00274658203125
  y: 0.00811767578125
  z: 1.00457763672
linear_acceleration_covariance: [4.0576900000000015e-05, 0.0, 0.0, 0.0, 4.0576900000000015e-05, 0.0, 0.0, 0.0, 4.7059600000000006e-05]
magnetic_field:
  x: -5.70635986328
  y: -23.7878417969
  z: 12.0328979492
magnetic_field_covariance: [0.0144, 0.0, 0.0, 0.0, 0.0144, 0.0, 0.0, 0.0, 0.0144]
orientation:
  x: 0.00389323662966
  y: 0.00107160676271
  z: 0.0446005314589
  w: 0.998996734619
  x: -0.00162210129201
  y: 0.00369776878506
  z: 0.982103049755
  w: -0.188301086426
euler_ypr:
  x: 0.00787434395384
  y: 0.00179378344995
  z: 0.0892384536019
temperature imu: 20.0
calibration_index: 63
  
```

初期バイアスは測定完了後に自動的に MCU フラッシュに保存して、IMU 再起動の際に MCU フラッシュから自動的に読み込みます。

2.8 (オプション)ROS2 における DDS の切り替え

以下の手順より、Fast DDS(デフォルト)を Cyclone DDS(例)あるいは他の DDS へ切り替えます。

- Fast DDS(デフォルト) → Cyclone DDS

```
$export RMW_IMPLEMENTATION=rmw_cyclonedds_cpp
```

```
$printenv RMW_IMPLEMENTATION
```

```
$source ~/ros2_ws/install/setup.bash
```

```
$ros2 launch haya_imu_ros2 haya_imu_launch.py
```

- Cyclone DDS → Fast DDS(デフォルト)

```
$export -n RMW_IMPLEMENTATION
```

```
$printenv RMW_IMPLEMENTATION
```

```
$source ~/ros2_ws/install/setup.bash
```

```
$ros2 launch haya_imu_ros2 haya_imu_launch.py
```

3. LED 表示

- Red = 電源供給 USB+5V
- Yellow = 6 軸または、9 軸フュージョン四元数出力

4. 追加説明

- 6 軸フュージョンの基準 0 度方位=IMU の初期方位、9 軸フュージョンの基準 0 度方位=磁東方向。

最大出力レートは haya_imu の実力値、実効値は対向装置のリソースパフォーマンスに関わります。

- `ros2 topic echo` と `ros2 topic hz` の出力値は「過小評価」と見られる際に、かわりに

```
$ros2 launch haya_imu_ros2 haya_topic_echo_launch.py と、
```

\$ros2 launch haya_imu_ros2 haya_topic_hz_launch.py が代用できます。

- 掲載規格はセンサの検証または特性評価から得られたデータであり生産での保証ではありません。
- 令和5年9月までは ICM-42688-V 実装、令和5年9月以降は ICM-42688-V、ICM-42688-P

区別せず出荷します。

5. 参考情報

5.1 主な規格

- 型番 haya_imu v3.x
- 内蔵 IC Microchip Cortex-M4(120MHz)、ICM-42688V/ICM-42688P/MMC5982MA
- 外部接続 USB2.0+ Type-C、USB+5V 給電、消費電力 150mW(環境温度 21℃ 実測値)
- 最大出力レート(Best Effort)
 - 6軸フュージョン / 9軸フュージョン回転ベクトル四元数 1000Hz
 - 3軸オイラー角(Z軸回り→Y軸回り→X軸回り順) 1000Hz
 - 加速度センサ 1000Hz
 - ジャイロスコープ 1000Hz
 - IMU 内部温度データ 1000Hz
 - 3軸地磁気センサ 500Hz(セトリセット機能使用)
- 測定レンジ
 - 加速度センサ ±8g
 - ジャイロスコープ ±2000dps
 - 地磁気センサ ±800μT
- バイアス測定補正 初期バイアス測定、動作時即時測定、内蔵補正機能あり
- 寸法 38.0mm × 39.0mm × 4.8mm(突起物含む)
- 取付穴 M3x4、隣り合う穴の中心間距離 32.0mm

5.2 ROS/ROS2 パッケージ

- ROS 専用パッケージの置き場所: https://github.com/soarbear/haya_imu_ros.git
- ROS2 専用パッケージの置き場所: https://github.com/soarbear/haya_imu_ros2.git
- ROS2 専用メッセージの置き場所: https://github.com/soarbear/haya_imu_msgs.git

6. ファームウェアの書き換え

6.1 書き換えモードへの切り替え方

RESET ボタンをダブルタップ（1回目と2回目タップの時間間隔 0.5s 以内）して、オレンジ LED は 5 回点滅して（点灯 1s、消灯 1s、5 回繰り返し）、通常モード→ファーム書き換えモードに切り替わります。

6.2 書き換え手順

- OS 環境、UBUNTU 18.04 以降.
- Clone

```
$mkdir ~/upload_haya
```

```
$cd ~/upload_haya
```

```
$git clone https://github.com/soarbear/bossac\_samg55.git
```

- Make & Upload

```
$sudo apt-get update
```

```
$sudo apt-get install build-essential
```

```
$sudo apt-get install libwxbase3.0-dev
```

```
$sudo apt-get install libwxgtk3.0-gtk3-dev
```

```
$sudo apt-get install libwxgtk3.0-dev
```

```
$sudo apt-get install libreadline-dev
```

```
$cd ~/upload_haya/bossac_samg55
```

```
$make clean
```

```
$make bin/bossac -j4
```

```
Upload→ $bin/bossac -e -w -v -b -i --offset=0x2000 [file_name].bin
```

- Reconnect

Upload して約 10 秒後、ターミナルウィンドから Reset 文字が見えたら、USB コネクタを抜き挿します。

7. リビジョン履歴

リビジョン	日付	内容	担当者
V3.2	2023年5月25日	新規作成	(株) 翔雲・Yanata