

TR-IMU1647X

ADI 6-Axis IMU Pose Estimation Board

TR-IMU1647X アプリケーションマニュアル

初版：2019年09月24日



TECHNO ROAD Inc.

<http://techno-road.com/>

ご使用になる前に

この度は TR-IMU1647Xをお買い上げいただき、誠にありがとうございます。

本製品をお役立て頂くために、このマニュアルを十分お読み頂き、正しくお使い下さい。

今後とも、弊社製品をご愛顧賜りますようよろしくお願いいたします。

＊本製品の内容及び仕様は予告なしに変更されることがありますのでご了承ください。

<取り扱い上の注意>

- 本製品は評価用です。研究開発の用途以外では使用しないでください。
- 本製品には、民生用の一般電子部品が使用されております。宇宙、航空、医療、原子力、運輸、交通、各種安全装置など人命、事故に関わる特別な品質、信頼性が要求される用途でのご使用はご遠慮ください。
- 極端な高温下や低温下、または振動の激しい環境での使用はご遠慮ください。
- 水中、高湿度、油の多い環境でのご使用はご遠慮ください。
- 腐食性ガス、可燃性ガス等の環境中でのご使用はご遠慮ください。
- 基板の表面を水で濡らしたり、金属に接触した状態で電源を投入しないでください。
- 定格を超える電源を加えないでください。
- ノイズの多い環境での動作は保証しかねますのでご了承ください。
- 発煙や発火、異常な発熱があった場合には、すぐに電源を切ってください。
- 本書に記載される製品及び技術のうち「外国為替及び外国貿易法」に定める規制貨物等（技術）に該当するものを輸出または国外に持ち出す場合には、同法に基づく輸出許可が必要です。

<保証について>

- 本製品の保証は、初期不良交換のみとなります。
万一初期不良品であった場合、商品到着後**30** 日以内に弊社までご連絡ください。不良品と引き換えに交換品をお届けいたします。
- お客様の都合による返品の場合、返送料はお客様負担でお願いいたします。（未使用品に限ります）
- 万が一、本製品を使用して事故または損失が発生した場合、弊社では一切その責任を負いません。
- 本製品の仕様範囲を超える条件において使用された場合については、動作は保証されません。
- 製品を改造した場合、保証は一切適用されません。
- 他社製品との接続互換性および相性問題は保証いたしません。

～目次～

第 1 章	本書について	1
第 2 章	製品概要	2
1.	製品の特徴.....	3
第 3 章	ハードウェアについて	4
1.	部品配置図(全体)	4
第 4 章	製品の使い方	5
1.	機器のセットアップ	5
2.	アプリのインストールから起動まで	6
3.	SPI モードの使い方 — ADI_DRIVER の動作方法.....	7
	必要な PC 環境	7
	インストール手順	7
	実行方法	8
4.	SPI モードの使い方 — ホスト側からの操作.....	10
5.	姿勢角出力モードの使い方 — IMU_Platform Tool で動作.....	11
6.	姿勢角出力モードの使い方 — 他のソフトで動作する場合	14
7.	姿勢角出力モードの使い方 — 各モードの出力形式.....	16
	姿勢角出力.....	16
	ジャイロ出力	16
	レジスタ値出力	16
	バイナリ出力	17
	姿勢角・加速度出力	17
	ジャイロ・加速度・温度出力	17

第 1 章 本書について

本書では TR-IMU1647X の使い方、Linux 環境での ADI_DRIVER の動作方法、各モードの通信書式などを記述します。

その他の情報は以下の資料を御覧ください。

ハードウェアマニュアル

- IMU1647X_ハードウェアマニュアル.pdf

ソフトウェアの開発マニュアル

- IMU1647X_ソフトウェア開発マニュアル.pdf

第 2 章 製品概要

TR-IMU1647X (以下「モジュール」と記載)は、アナログデバイセズ社の多様な IMU を取り付け可能なモジュールであり、非線形相補フィルタと組み合わせることで高精度な姿勢角の出力が可能です。

現在動作確認済みの IMU は ADIS16470、ADIS16475-2 のみです。

また、ソースコードは GitHub で公開しているので搭載するシステムに合わせてアレンジすることも可能です。

URL : <https://github.com/technoroad/TR-IMU1647X>

(IMU センサは別売りのため、お客様で別途ご用意する必要があります)

- ・ ARM Cortex-M0(STM32F070CBT6 48MHz)コアを使用
- ・ 基板が小さいので小型ロボットにも搭載可能
- ・ 磁気センサを使用しないため、地磁気を使用できない屋内などでも使用可能
- ・ 100Hz で 6 軸(加速度 3 軸、ジャイロ 3 軸)情報の取得
- ・ 100Hz で非線形相補フィルタ(舩屋賢 先生提供)による姿勢角の演算
- ・ USB(仮想 COM)によるデータ出力に対応
- ・ SWD 端子(MIPI-10)を通してマイコンのデバッグ・書き込みが可能
- ・ ST マイクロ社のマイコンのため無料で容量無制限の開発(TrueStudio)が可能

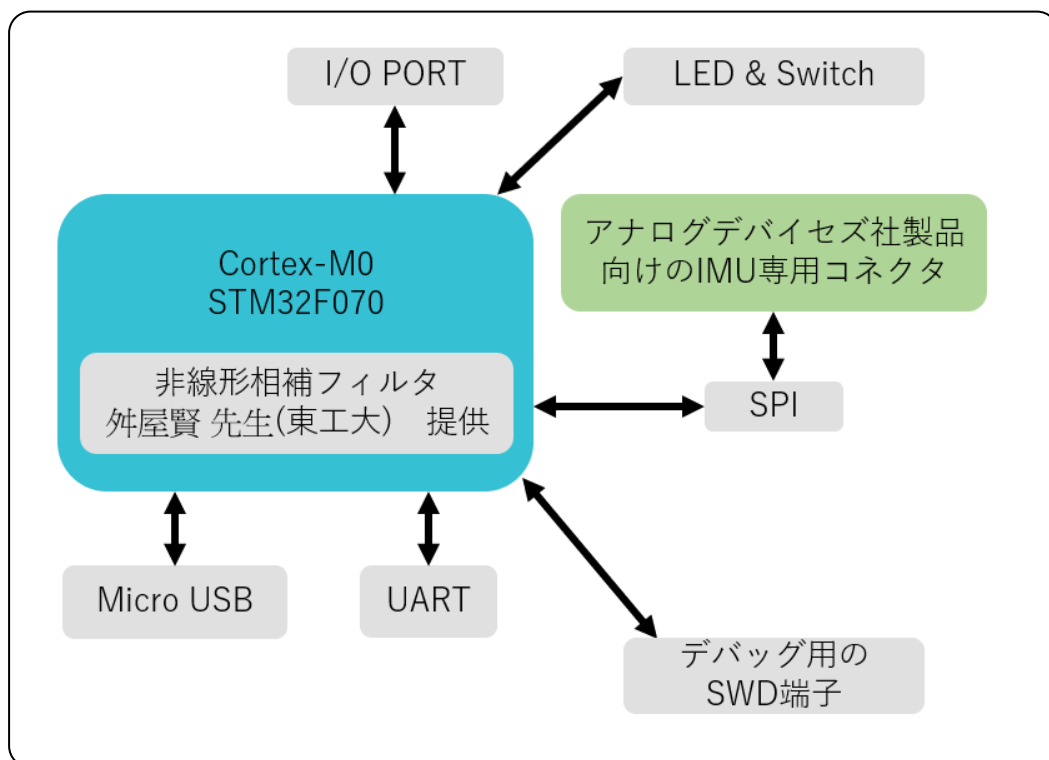
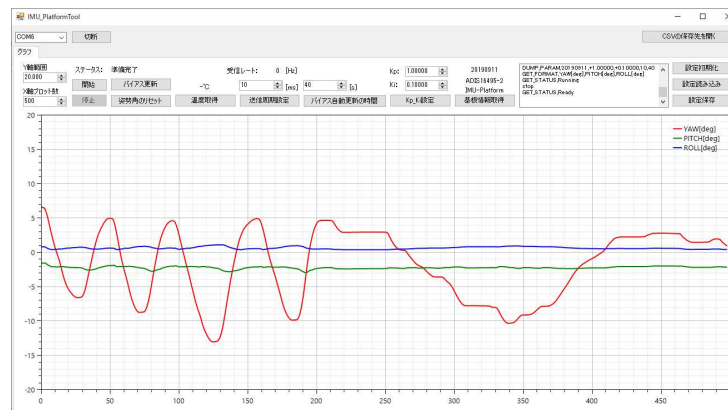


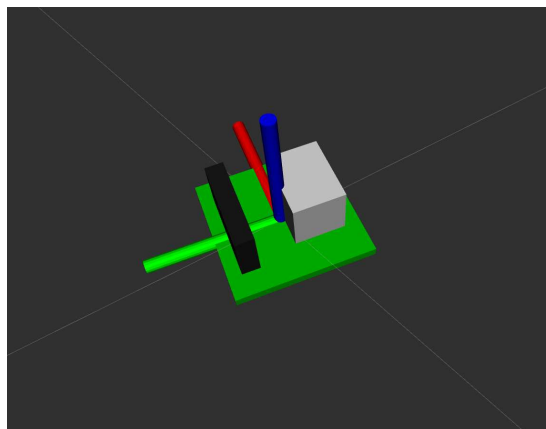
図 1 . TR-IMU1647X システム概略図

1. 製品の特徴

- 姿勢角出力モード
 - ロール・ピッチ・ヨー角の推定値を出力するモード
 - PC 側で姿勢角の推定値を求めるアルゴリズムが不要
 - 舩屋賢 先生(東工大)より提供の非線形相補フィルタを搭載
 - 最大で 100Hz で出力が可能
 - CSV 出力であるため解析やプログラミングが簡潔にできる
 - 各種コマンドによって PC 側からの操作が簡単にできる
 - 姿勢角+ α の出力もスライドスイッチで選択可能
 - ROS へ接続することも可能



- SPI モード
 - SPI 通信を模擬し、ROS の adi_driver に接続できるモード
 - ROS への組み込みが簡単
 - PC から直接センサを操作できる



第3章 ハードウェアについて

1. 部品配置図(全体)

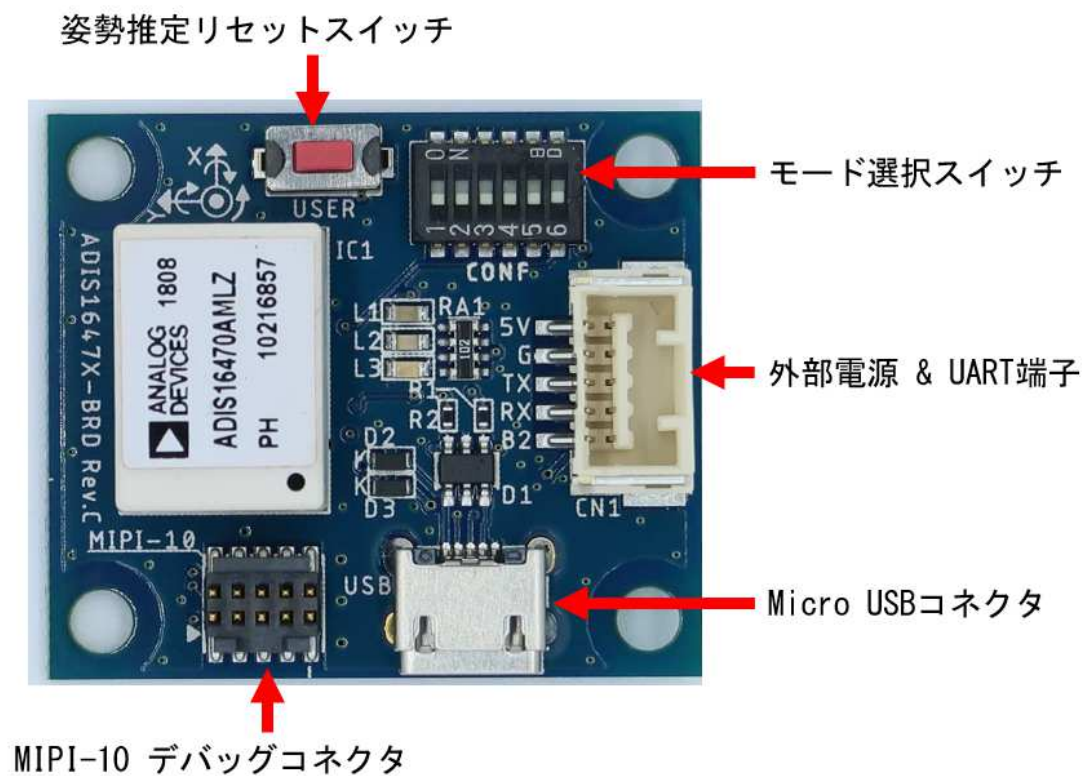


図 TR-IMU1647X 部品配置図(全体)

第4章 製品の使い方

1. 機器のセットアップ

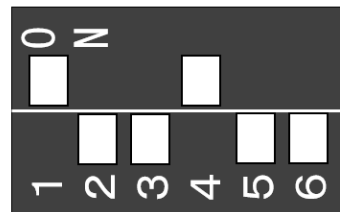
電源を入れる前にモード選択スイッチで動作モードの選択を行ってください。

動作モードの出荷時の設定は姿勢角出力、起動時のキャリブレーションなしを選択しています。

モード選択スイッチ

動作モード	1	2	3	4
SPI モード	OFF	OFF	OFF	OFF
姿勢角[deg]出力	ON	OFF	OFF	OFF
ジャイロ[deg/s]出力	OFF	ON	OFF	OFF
レジスタ値出力	ON	ON	OFF	OFF
バイナリ出力	OFF	OFF	ON	OFF
姿勢角[deg]・ 加速度[g]出力	ON	OFF	ON	OFF
ジャイロ[deg/s]・ 加速度[g]・ 温度[deg]出力	OFF	ON	ON	OFF

※表にない組み合わせは未実装です。



(図は出荷時設定です)

ON

OFF

DFU選択スイッチ

USBとUARTの切り替え
OFF : USB
ON : UART

起動時のキャリブレーションの有無
OFF: あり(120秒)
ON: なし

動作モードの選択を行ったら USB かもしくは、外部電源コネクタから電源を入れてください。
その後、赤 LED が光った後に消灯したら起動完了です。

(IMU センサは別売りのため、お客様で別途ご用意する必要があります)

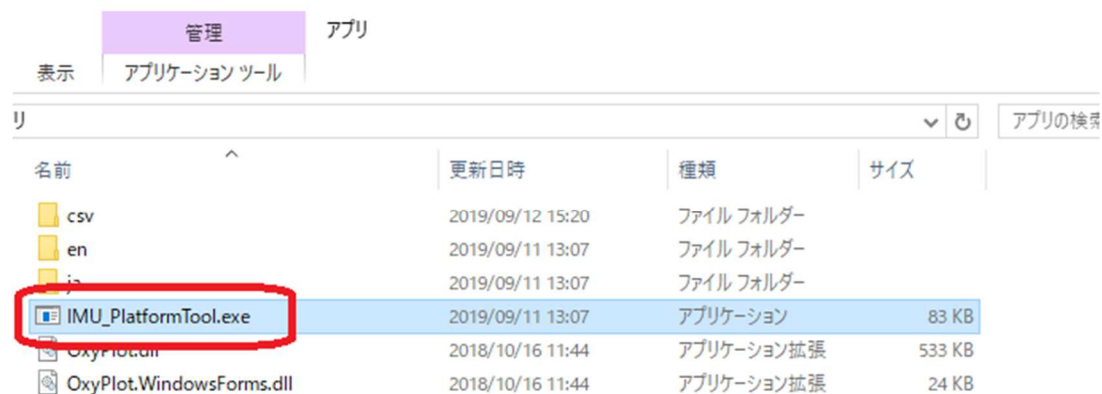


2. アプリのインストールから起動まで

以下ページより最新版のアプリのダウンロードを行ってください。

https://github.com/technoroad/IMU_PlatformTool/releases

ダウンロードが完了したら zip を任意の場所へ解凍した後、フォルダの中の
“IMU_PlatformTool.exe” をダブルクリックで起動してください。



詳しい動作の説明は以下の項目で説明します。

[5. 姿勢角出力モードの使い方 — IMU_Platform Tool で動作]

3. SPI モードの使い方 — ADI_DRIVER の動作方法

SPI モードは ADI_DRIVER 対応の機器である USB-IIS を模擬したモードであり、ADI_DRIVER のチュートリアルと同じ方法で動作可能です。

ADI_DRIVER を使用するには ROS (Robot Operating System) をインストールした環境が必要となります。

必要な PC 環境

- OS Ubuntu18 に ROS Melodic をインストールしたもの
もしくは
- OS Ubuntu16 に ROS Kinetic をインストールしたものを
をご用意下さい。

また以下のコマンドで USB ポートへのアクセス権限をログインユーザへ付与しておいて下さい。

```
$ sudo adduser [your_user_name] dialout
```

※コマンド実行後にログインユーザのログアウト、再ログインが必要です。

インストール手順

- ワークスペースの作成
ターミナルより以下のコマンドを入力して下さい。

```
$ mkdir -p ~/catkin_ws/src  
$ cd ~/catkin_ws/src  
$ catkin_init_workspace  
$ cd ..  
$ catkin_make
```

最後のコマンドがエラーなく終了すればワークスペースの作成は完了です。

- ソースのインストールとビルド
ターミナルより以下のコマンドを入力して下さい。

```
$ cd ~/catkin_ws/src  
$ git clone https://github.com/tork-a/adi_driver  
$ cd ..  
$ rosdep update  
$ rosdep install --from-paths src --ignore-src --rosdistro=${ROS_DISTRO} -y  
$ catkin_make
```

最後のコマンドがエラーなく終了すればソースのインストールとビルドは完了です。

- パスを通す
ターミナルより以下のコマンドを入力して下さい。

```
$ source ~/catkin_ws/devel/setup.bash
```

- ※これはターミナルを開く毎に必要です。
もしくはコマンドを ~/.bashrc に記載して下さい。

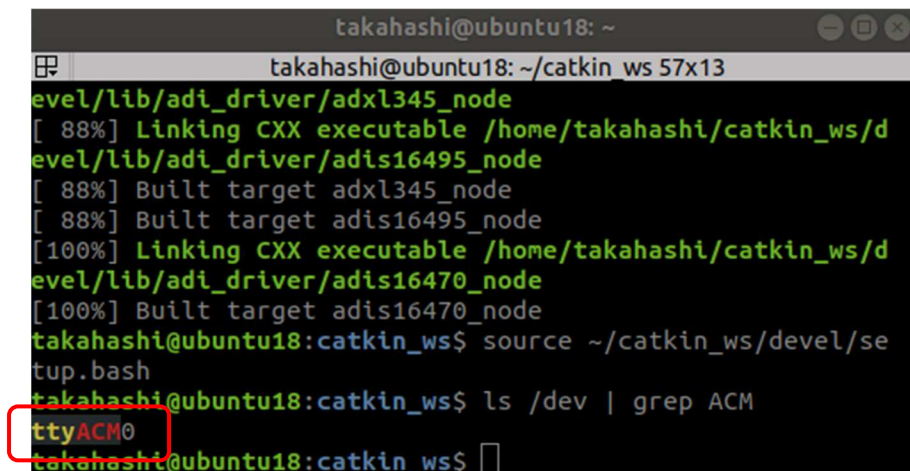
ここまででインストールは完了となります。

実行方法

- 機器の接続
PC とモジュールを USB ケーブルで接続して下さい。
- ポート名の確認
以下のコマンドでポート名を確認して下さい。

```
$ ls /dev | grep ACM
```

次の画像はコマンド実行時の一例です。（ここでは ttyACM0 というポート名）

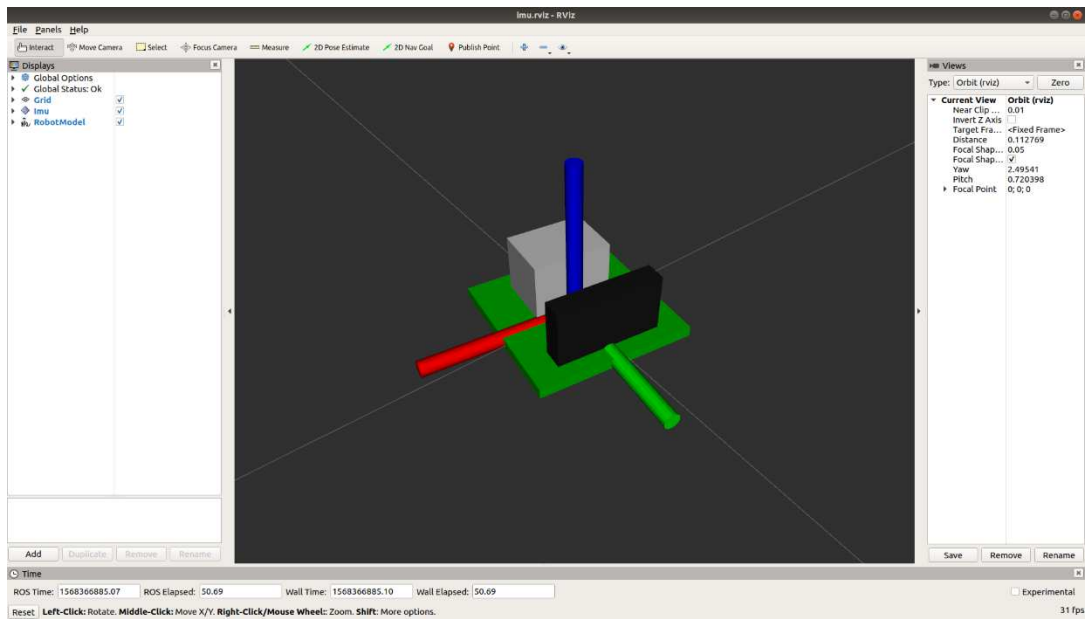


```
takahashi@ubuntu18: ~  
takahashi@ubuntu18: ~/catkin_ws 57x13  
evel/lib/adi_driver/adxl345_node  
[ 88%] Linking CXX executable /home/takahashi/catkin_ws/d  
evel/lib/adi_driver/adis16495_node  
[ 88%] Built target adxl345_node  
[ 88%] Built target adis16495_node  
[100%] Linking CXX executable /home/takahashi/catkin_ws/d  
evel/lib/adi_driver/adis16470_node  
[100%] Built target adis16470_node  
takahashi@ubuntu18:catkin_ws$ source ~/catkin_ws/devel/se  
tup.bash  
takahashi@ubuntu18:catkin_ws$ ls /dev | grep ACM  
ttyACM0  
takahashi@ubuntu18:catkin_ws$
```

- 起動
以下のコマンドで launch ファイルを起動してください。
※ここでは adis16495 に接続しています。
※引数の device に先ほど確認したポート名を記載して下さい。

```
$ roslaunch adi_driver adis16495.launch device:=/dev/ttyACM0 with_rviz:=true
```

次の画像のなウィンドウが立ち上がれば起動完了です。
センサの姿勢を変えるとウィンドウ内のセンサの姿勢が変化します。



また、ジャイロ等の値を確認したい場合は以下のコマンドを別ターミナルより実行して下さい。

```
$ rostopic echo /imu/data_raw
```

詳細なオプションは launch ファイル、もしくは以下の URL をご参照下さい。

https://github.com/tork-a/adi_driver

➤ 終了

各コマンドを実行しているターミナルをアクティブ化（クリックする）し、
Ctrl + c キーを入力して下さい。

ADI_DRIVER の動作方法は以上です。

4. SPI モードの使い方 — ホスト側からの操作

USB(もしくは UART)で以下の書式通りに送ることによってセンサの任意のレジスタへの読み込み・書き込みが可能です。

(1) レジスタの読み込み

コマンド

MODE	ADDR1	DATA1	ADDRn	DATAn
0x61	0x00~0x7F	0x00	0x00~0x7F	0x00

レスポンス(コマンドと同じ文字数)

HEADER	DATA1_HIGH	DATA1_LOW	DATAn_HIGH	DATAn_LOW
0xFF	0xNN	0xNN	0xNN	0xNN

任意のレジスタの値を読み込みます。

ADDR の値が 0x00~0x7F ならば読み込み、0x80~0xFF なら書き込みになります。

コマンドの長さによって1回の通信で複数のレジスタを読み込めます。

(2) レジスタの書き込み

コマンド

MODE	ADDR1	DAT1	ADDRn	DATAn
0x61	0x80~0xFF	0x00	0x80~0xFF	0x00

レスポンス(コマンドと同じ文字数)

HEADER	DATA1_HIGH	DATA1_LOW	DATn_HIGH	DATAn_LOW
0xFF	0x00	0x00	0x00	0x00

任意のレジスタに値を書き込みます。

ADDR の値が 0x00~0x7F ならば読み込み、0x80~0xFF なら書き込みになります。

コマンドの長さによって1回の通信で複数のレジスタを書き込めます。

読み込みの時、値は一回の通信分遅れて帰ってくるのでレジスタの読み込みを1回多く行う必要があります。

例1：プロダクト ID のみを取得する場合(ADIS16495 の時)

コマンド：0x61 0x7E 0x00 0x00 0x00

レスポンス：0xFF 0x00 0x00 0x40 0x71

例2：角速度3軸をまとめて取得する場合(ADIS16495 の時)

コマンド：0x61 0x10 0x00 0x12 0x00 0x14 0x00 0x16 0x00 0x18 0x00 0x1A 0x00 0x00 0x00

レスポンス：0xFF 0x00 0x00 0x57 0xED 0x00 0x05 0x33 0x3E 0xFF 0xFB 0xDE 0x49 0xFF 0xFE

5. 姿勢角出力モードの使い方 — IMU_Platform Tool で動作

弊社の GitHub で公開している IMU_Platform Tool(Windows 専用)では姿勢角のグラフ描画や設定の変更、取得データを CSV で保存する機能があります。

URL : https://github.com/technoroad/TR_IMU_Platform

アプリのインストールから起動までの説明は「2 アプリのインストールから起動まで」を参照してください。

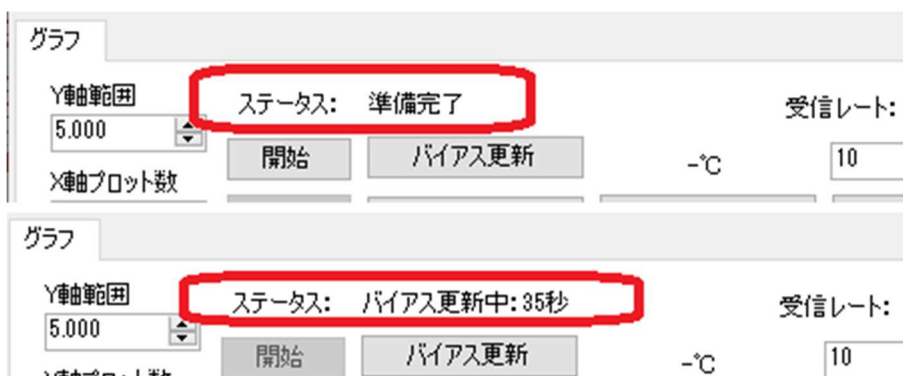
アプリを使用する PC とモジュールを USB で接続したらアプリを起動します。起動したら画面左上のリストからモジュールの COM ポートを選択した上で隣の接続ボタンを押します。



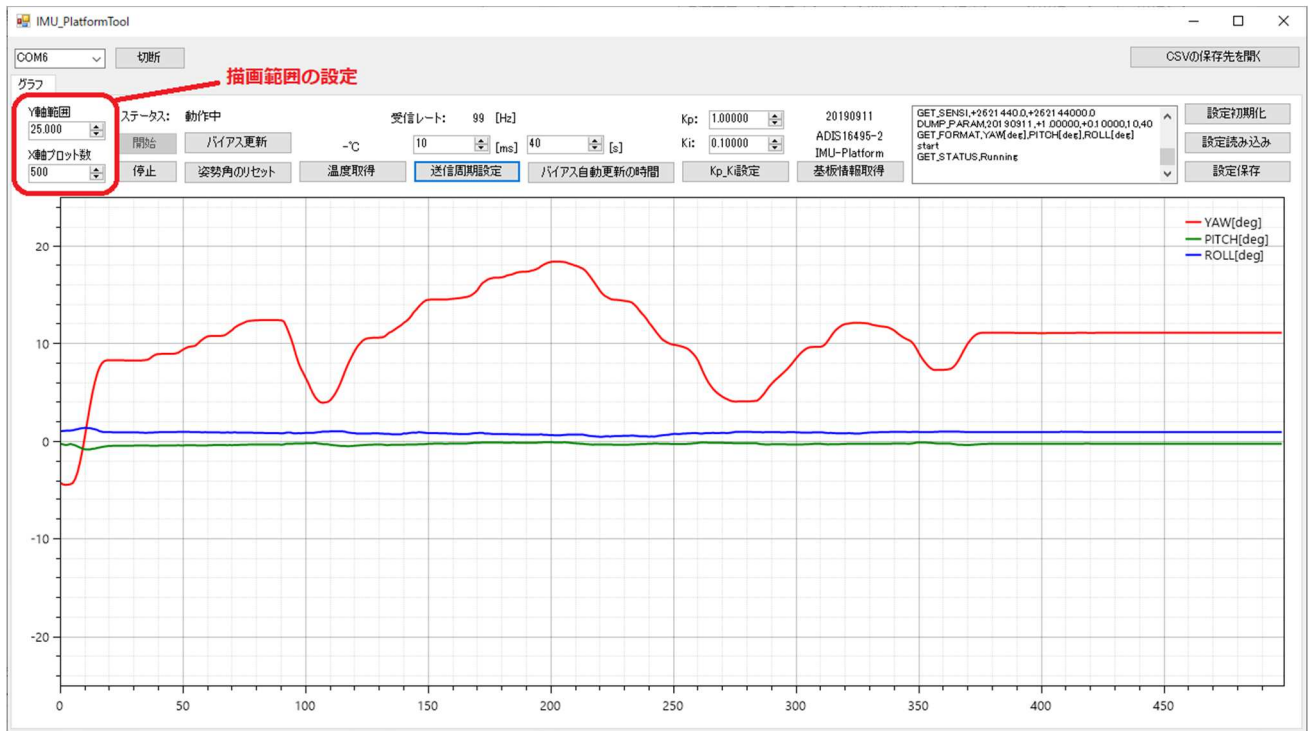
接続が完了すると「準備完了」と表示され、「開始ボタン」が有効になります。ただし、モード選択スイッチで起動時のキャリブレーションを有効にしていると終了するまで待機する必要があるため、「開始ボタン」は無効のままで終了するまで待機する必要があります。

キャリブレーションが終了すると「準備完了」と表示され、「開始ボタン」が有効になります。

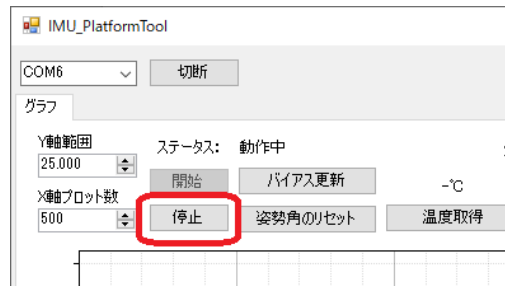
また、キャリブレーション中はモジュールを静止させてください。



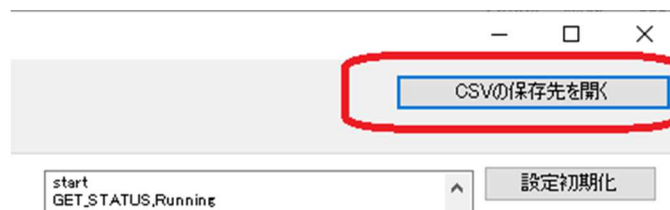
「準備完了」と表示されてから「開始ボタン」を押すとモード選択スイッチで選択した形式でデータ出力が開始します。グラフの描画範囲を変更したい場合は画面左側の「Y 軸範囲」と「X 軸プロット数」を変更すると適応されます。



データの出力を停止する場合は「停止ボタン」を押してください。押すとステータスは準備完了となり、再び「開始ボタン」が押せるようになります。



取得したデータは CSV データとして保存されていて、閲覧する場合は画面右上の「CSV の保存先を開くボタン」を押してください。押すとエクスプローラーでファイルが開かれます。



UI の説明を行います。これらの設定や取得はデータ出力中でも適応されます。

- 温度取得ボタン：センサの温度を取得します。
- 送信周期設定ボタン：姿勢角などの送信周期を設定します。
 - この変更は後述の「設定保存ボタン」を押さなければ保存されません。
 - TR-IMU-Platform は最大で 1[ms]で、1[ms]刻みで設定可能。
 - TR-ADIS1647X は最大で 10[ms]で、10[ms]刻みで設定可能。
- 受信レート：PC が受信したデータ数を表示します。
- バイアス自動更新の時間ボタン：起動時のバイアスの更新時間を設定します。
 - この変更は後述の「設定保存ボタン」を押さなければ保存されません。
- Kp_Ki 設定ボタン：非線形相補フィルタの比例ゲインと積分ゲインを設定。
 - この変更は後述の「設定保存ボタン」を押さなければ保存されません。
- 基板情報取得ボタン：以下の情報を取得します。
 - ファームウェアのバージョン情報(ビルドした年月日)
 - ファームウェアがターゲットとしているセンサの型番
 - 基板の型番(TR-IMU-Platform と TR-ADIS1647X のどちらか)

受信レート: 0 [Hz]		Kp: 1.00000	20190911	
-℃	10 [ms]	40 [s]	Ki: 0.10000	ADIS16495-2 IMU-Platform
温度取得	送信周期設定	バイアス自動更新の時間	Kp_Ki設定	基板情報取得

- パラメータ：送信周期設定、バイアス自動更新の時間、**Kp_Ki** 設定が対象
- 設定初期化ボタン：マイコンのパラメータが初期値に戻されます。
- 設定読み込みボタン：マイコンのパラメータを読み出します。
- 設定保存ボタン：パラメータをマイコンの **FLASH** 領域に保存します。

通信のログ

CSVの保存先を開く

20190911 15:16:49.2 U-Platform 反情報取得

DUMP_PARAM,20190911,+1.00000,+0.10000,1.0,40
GET_FORMAT,YAW[deg],PITCH[deg],ROLL[deg]
GET_STATUS,Running
stop
GET_STATUS,Ready

設定初期化

設定読み込み

設定保存

— YAW[deg]
— PITCH[deg]
— ROLL[deg]

6. 姿勢角出力モードの使い方 — 他のソフトで動作する場合

全てのデータとコマンドが CSV 形式であるため操作が簡単に行えます。
ここでは Teraterm を例にして説明します。

Teraterm の設定は以下のようにしてください。

受信の改行コード : AUTO

送信の改行コード : CR+LF もしくは、LF のみ

以下、文字を入力後に改行 (<CR><LF>もしくは、<LF>のみ) が必須です。

新しい接続でモジュールの COM を開いたら以下のコマンドを実行するとデータの出力が開始します。

```
$ start<CR><LF>
```

姿勢角出力の場合は以下のように取得できます。

```
$ -0.009,-0.056,+0.067<CR><LF>
```

```
$ -0.010,-0.056,+0.067<CR><LF>
```

```
$ -0.010,-0.054,+0.068<CR><LF>
```

停止する場合は以下のコマンドを実行してください。

```
$ stop<CR><LF>
```

モジュールの状態を取得したい時は以下のコマンドを実行してください。

```
$ GET_STATUS <CR><LF>
```

返り値は以下の中から 1 行のみ取得できます。

```
$ GET_STATUS,Ready<CR><LF>
```

```
$ GET_STATUS,Running<CR><LF>
```

```
$ GET_STATUS,AutoBiasUpdating<CR><LF>
```

” Ready“ならば出力の停止中、”Running“ならば出力中、
”AutoBiasUpdating“ならば起動時のバイアス更新中と区別できます。

データの送信周期を任意に変更したい場合は以下のコマンドを実行してください。
以下の場合は 100[ms]に設定されます。

```
$ SET_SEND_CYCLE,100<CR><LF>
```

設定後、モジュールの電源を落とすと設定が消えてしまいます。

保存する場合は以下のコマンドを実行してください。

```
$ SAVE_PARAM<CR><LF>
```

コマンド一覧を以下に示します。

コマンド	引数	機能	返り値
start	なし	データの出力を開始します。	start
stop	なし	データの出力を停止します。	stop
START_BIAS_CORRECTION	なし	センサのバイアス値を更新します。	START_BIAS_CORRECTION
WRITE_REG	,ページ番号,アドレス,値	センサの指定したページ番号のアドレスに値を書き込みます。 1647Xセンサの場合はページ番号は無視されます。	WRITE_REG ,ページ番号,アドレス,値
READ_REG	,ページ番号,アドレス	指定したページ番号のアドレスの値を読み込みます。	READ_REG ,ページ番号,アドレス,値
READ_TEMP	なし	マイコンからセンサの温度を読み込みます。	READ_TEMP;温度[deg]
SET_KP_KI	,比例ゲイン,積分ゲイン	姿勢角を求めるフィルターの比例ゲインと積分ゲインを設定します。	SET_KP_KI ,比例ゲイン,積分ゲイン
RESET_FILTER	なし	姿勢角の値を0[deg]へリセットします。	RESET_FILTER
LOAD_INIT	なし	マイコンのパラメータを初期値に戻します。	LOAD_INIT
SAVE_PARAM	なし	マイコンのパラメータを保存します。	SAVE_PARAM
DUMP_PARAM	なし	マイコンの全てのパラメータを返します。	DUMP_PARAM ,バージョン ,比例ゲイン,積分ゲイン ,送信周期[ms] ,起動時の待機時間[s]
SET_SEND_CYCLE	,送信周期[ms]	姿勢角など送信周期を設定します。	SET_SEND_CYCLE ,送信周期[ms](反映値)
SET_STARTUP_TIME	,起動時の待機時間[s]	起動時の待機時間を設定します。	SET_STARTUP_TIME ,起動時の待機時間[s]
GET_VERSION	なし	ファームウェアのバージョンを取得します。	GET_VERSION ,バージョン
GET_PROD_ID	なし	ファームウェアで対応しているセンサ型番を取得します。	GET_PROD_ID ,センサの型番
GET_FORMAT	なし	姿勢角などの出力データのフォーマットを取得します。	GET_FORMAT ,データのフォーマット
GET_SENSI	なし	ジャイロと加速度センサの感度を取得します。	GET_SENSI ,ジャイロ感度,加速度感度
GET_BOARD_NAME	なし	ボードの名称を取得します。	GET_BOARD_NAME ,ボードの名称
GET_STATUS	なし	現在の状態を取得します。 待機中: Ready、実行中: Running、 バイアス自動更新中: AutoBiasUpdating, "残り時間"	GET_STATUS ,現在の状態
コマンド表に無い文字列	なし	コマンドではない文字列を受信したときに返します。	ERROR_NONE_CMD ,受信した文字列
各コマンドのエラー	なし	コマンドの引数にエラーがある場合、 コマンドの先頭に"ERROR_"を付与して返します。	ERROR_WRITE_REG etc...

7. 姿勢角出力モードの使い方 — 各モードの出力形式

モード選択スイッチの選択で変更できる各モードの出力形式を示します。
以下のモードは前述のコマンドが使用できます。

姿勢角出力

フォーマット：

yaw(Z)[deg],pitch(Y) [deg],roll(X) [deg]

例：

-9.446,+6.729,-27.438

-10.819,+5.496,-27.048

-12.263,+4.515,-27.607

ジャイロ出力

フォーマット：

X 軸ジャイロ[deg/s],Y 軸ジャイロ[deg/s],Z 軸ジャイロ[deg/s]

例：

+36.330,-13.031,-1.613

+54.027,-30.788,-2.177

+57.086,-32.556,-2.660

レジスタ値出力

フォーマット：

X_GYRO,Y_GYRO,Z_GYRO,X_ACCL,Y_ACCL,Z_ACCL,CSUM

例：

00007BC8,FFFD54FA,00040DB8,000B5E92,FF9153AF,0F9ECC03,5F

000217F3,FFFF2011,0000B22C,000BE7B6,FF929DCF,0F9D1F8F,18

0005CD59,000040A9,FFFE6EBB,000B7F78,FF940C30,0F9B2C53,34

CSUM は各データを 8 ビットごとに分解して、全て足した値の下位 8 ビットです。
UART 通信の場合は CSUM の比較を推奨します。

バイナリ出力

フォーマット：

バイト番号	データ名
0	HEADER1(0xAC 固定)
1	HEADER2(0xCA 固定)
2	LENGTH(24 固定)
3~6(4byte)	X_GYRO(MSB)
7~10(4byte)	Y_GYRO(MSB)
11~14(4byte)	Z_GYRO(MSB)
15~18(4byte)	X_ACC(MSB)
19~22(4byte)	Y_ACC(MSB)
23~26(4byte)	Z_ACC(MSB)
27	CSUM

CSUM はバイト番号 3~26 を全て足した値の下位 8 ビットです。

姿勢角・加速度出力

フォーマット：

yaw(Z) [deg],pitch(Y) [deg],roll(X) [deg],X 軸加速度[g],Y 軸加速度[g],Z 軸加速度[g]

例：

+0.164,+0.168,-0.001,+0.043,-0.019,+0.998
 +0.163,+0.228,+0.003,+0.050,-0.017,+0.998
 +0.167,+0.278,+0.0,+0.055,-0.014,+0.998

ジャイロ・加速度・温度出力

フォーマット：

X 軸ジャイロ[deg/s],Y 軸ジャイロ[deg/s],Z 軸ジャイロ[deg/s],
 X 軸加速度[g],Y 軸加速度[g],Z 軸加速度[g],温度[deg]

例：

+0.158,+0.332,-0.005,+0.053,-0.005,+0.998,+31.500
 +0.150,+0.343,-0.002,+0.047,-0.002,+0.998,+31.500
 +0.153,+0.342,-0.001,+0.040,-0.002,+0.999,+31.400

＜本文書について＞

本ドキュメントに含まれるソースプログラムおよび、本ドキュメントに関する権利や知的所有権は弊社が所有しています。

弊社の許可なく、本ドキュメントの全て、または一部に関わらず、複製, 改変, 転用等を行うことはできません。

＜商標について＞

本ドキュメントに掲載されている会社名, 製品名は、それぞれ各社の商標または登録商標です。

TECHNO ROAD Inc.

株式会社テクノロード

<http://techno-road.com/>

E-mail:post@techno-road.com