

ADI Trinamicのモータ・コントローラ用のROS 1ドライバ、ロボットの開発が容易に

著者：Krizelle Paulene Apostol、ソフトウェア・システム・エンジニア
Jamila Macagba、シニア・ソフトウェア・システム・エンジニア
Maggie Maralit、ソフトウェア・システム設計エンジニアリング・マネージャ

概要

アナログ・デバイセズは、ROS (Robot Operating System) のエコシステムにおいて当社の製品を簡単に使用できるようにするための取り組みを行っています。一例としては、それらの製品に対応するROSドライバの開発が挙げられます。本稿では、それらのドライバを、アプリケーション、製品、システム（自律型ナビゲーション・システム、セーフティ・バブル・マップ、データ収集用ロボットなど）に組み込んで使用する方法を紹介します。その方法を活用すれば、新たな技術を直ちに評価し、サードパーティの製品との相互運用性の問題を回避することが可能になります。本稿の内容は、様々な製品を対象としています。そのなかで特に注目していただきたいのは、**ADI Trinamic™のモータ・コントローラ**向けに開発されたROSドライバです。それらのドライバは、組み込みモーション・コントロールに最適なボード・レベルの完全なモジュールとして実装されています。各モータ・コントローラには、モーション・コントロールに関するADI Trinamicの専門技術や、当社独自のアナログIC向けのプロセス技術、電源設計のスキルなどが適用されています¹。

ROSとは何か？

本稿のテーマである**ROS**は、ロボティクス向けのミドルウェアです。ドライバから最先端のアルゴリズムに至るまで、一連のソフトウェア・ライブラリや強力な開発ツールが含まれています。これを活用すれば、ロボットのシステムやアプリケーションを開発するのが容易になります。対象とする分野は、民生、産業、自動車など多岐にわたります。また、Linux、Windows、Mac OSなどのプラットフォームや、一部の組み込みプラットフォームなどをサポートしています。加えて、すべてのソフトウェアがオープンソース化されており、商用利用することも可能です。ROSはグ

ローバルなコミュニティのサポートを得て開発されており、豊富なリソースが提供されています。そのため、ユーザは設計／開発をより容易に進めることができます。

ROSの2つのバージョン

初期のROSは2007年に開発されました。それ以降、ROSは、自動運転車、産業用ロボット、航空機などの分野で使用されるロボットの開発に広く利用されるプロトタイピング用のプラットフォームと位置づけられました。ROSは継続的に進化を遂げており、現在はROS 1とROS 2の2つのバージョンが存在します。

ROS 1をベースとするシステムとROS 2をベースとするシステムは独立した形で存在することになります。ただ、ROSに対応するブリッジを利用すれば、両システムの間で通信を行ったりデータ交換を実施したりすることができます。詳細については、[ros2/ros1_bridge](#)のページをご覧ください。

表 1. ROS 1 と ROS 2 の主な相違点²

項目	ROS 1	ROS 2
通信プロトコル	XML-RPC + TCPROS	DDS
アーキテクチャ	ROSマスタ + 分散型	完全な分散型
ビルド・システム	Catkin (CMakeベース)	colcon/ament (CMakeベース)
ビルド出力	ros_ws/devel	ros_ws/install
パラメータ	グローバル・パラメータ・サーバ ダイナミック・リコンフィギュア	ノードごとのパラメータ
ローンチ	XML	Python (+XML, YAMLの代替形式)
コマンド	roslaunch、roswarn、rostopicなど	ros2 launch、ros2 run、ros2 topicなど
プラットフォーム	主にUbuntu	Linux、Mac OS、Windows



ROSをサポートするプラットフォーム

ROS 1の最終版はROS Noeticと呼ばれています。そのサポートは2025年5月に終了する予定です。一方のROS 2については、2020年6月の提供開始以来、開発ディストリビューションが継続的にリリースされている状態にあります。

ROS 1をサポートするプラットフォーム、ROS 2をサポートするプラットフォームの一覧については、それぞれのリンク先を確認してください。

ROSで用いられる基本的な概念

続いて、ROSで用いられる基本的な概念について説明します。図1に示すように、ROSではパッケージ、ノード、トピック、サービス、メッセージといった概念が使われます。なお、以下で説明するROSの基本的な概念は、ROS 1とROS 2でほぼ同等の意味を持ちます。

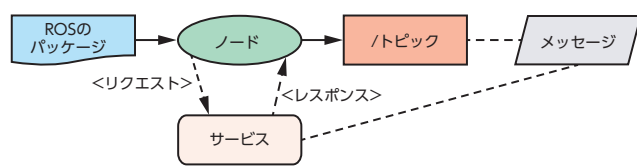


図1. ROSにおける基本的なデータの流れ

パッケージ

パッケージは、ROSのプログラムやノードを対象とする主要な組織構造です。また、ROSにおける最も小さなビルド/リリースの単位でもあります。パッケージを作成する際には、ROSに専用のワークスペースを設定することが重要です。このワークスペースをcatkinワークスペースと呼びます。Catkinは、ROSの正式なビルド・システムとして使用されています。

ノード

ROSのノードとは、特定のタスクを実行するプロセスのことです。ROSの環境内で、実行可能なプログラムとして作成されます。ノードは、Pythonのクライアント・ライブラリ (rospy) やC++のクライアント・ライブラリ (roscpp) といったROSのクライアント・ライブラリを使用することによって互いに通信することができます。また、トピックのサブスクリバまたはパブリッシャあるいはその両方を兼ねることが可能です。ノードは、サービスを提供することも利用することもできます³。

トピック

トピックは、ROSノードによって生成 (ROSの用語ではパブリッシュ) されるデータ・チャンネルです。

ROSでは、パブリッシャのノードがトピックのブロードキャストとなります。一方、サブスクリバのノードはトピックのリスナーになります。

図2において、generic_motor_controlのノードは、ブロードキャストとして機能します。/cmd_velは、velocity_publisherがパブリッシュするトピックです。つまり、velocity_publisherは、モータの制御に必要な速度の情報 (command velocity) を提供します。

一方、velocity_subscriberはros_applicationのノードです。このノードはリスナーとして機能し、トピックである/cmd_velをサブスクライブします。つまり、velocity_subscriberは、velocity_publisherが提供する速度の情報を参照/使用します。

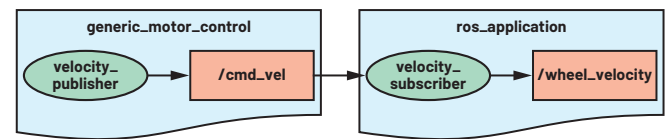


図2. パブリッシャとサブスクリバの関係

メッセージ

上述したとおり、トピックはデータ・チャンネルです。それに対し、メッセージは各センサーに対応するROSのメッセージ形式でデータを表したものです。以下、各種のセンサーとそれに対応するROSのメッセージ形式の例を列挙します。

- ▶ ToF (Time of Flight) カメラ: sensor_msgs/Image, sensor_msgs/PointCloud
- ▶ 慣性計測ユニット: sensor_msgs/Imu
- ▶ モータ制御用センサー: geometry_msgs/Twist
- ▶ ホイール・エンコーダ: geometry_msgs/TwistStamped, geometry_msgs/TwistWithCovarianceStamped

ROSのトピックは、メッセージの送信 (トピック・パブリッシャ) またはメッセージの受信 (トピック・サブスクリバ) によって伝達されます。両者の間では、データ・タイプが一致している必要があります。例えば、図2では、velocity_publisherのノードからの速度の情報 (command velocity) を、velocity_subscriberのノードが参照/使用します。トピックのパブリッシャであるvelocity_publisherがデータ・タイプとしてgeometry_msgs/Twistを使用している場合には、トピックのサブスクリバであるvelocity_subscriberも同じデータ・タイプを使用しなければなりません。

サービス

パブリッシャとサブスクリバによる通信モデルは、オープンエンドの形式を採用しています。分散型のシステムで求められることが多いやり取りは必要ありません⁴。

ROSのサービスは、リクエストの送信とレスポンスの受信から成るノート間の通信を可能にするものです。パブリッシャとサブスクリバの通信モデルには、.srvファイルを使用します。このファイルには、サービスに関する説明を記述します。また、リクエストとレスポンスのメッセージ・タイプなどの指定も行います。

サービスは、クライアントとサーバが存在する双方向の同期通信モデルです。サーバのノードはサービスを提供する側です。一方、クライアントのノードは、リクエストを送信してサーバのノードからのレスポンスを待ちます。

ここで図3をご覧ください。この例において、server_nodeはvel (command velocity) の値を変更するサービスであるSetVelocity.srvを提供します。そして、サービスはfloat32型の速度の値を受け取ります。その上で、string型のステータス情報として、リクエストされた速度が設定された場合には「Success」、設定されなかった場合には「FAIL」を返します。

client_nodeは、command velocityに2.5mbpsを設定するというリクエストを送信します。このリクエストを受け取ったserver_nodeは、「Success」のレスポンスを送信します。

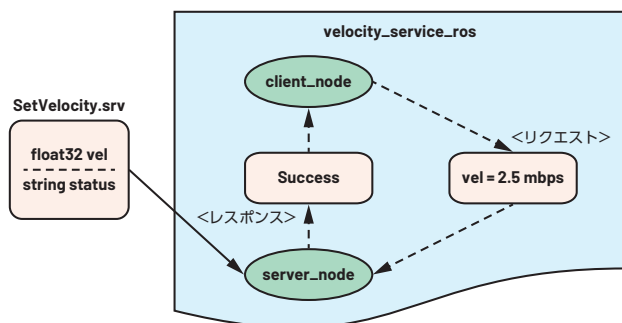


図3. サービスの使用例

アナログ・デバイセズのソリューションをROSのエコシステムに組み込む

ROS-Industrialコンソーシアムは、ROS向けソフトウェアの高度な機能を、産業関連のハードウェア／アプリケーションに拡張するためのオープンソース・プロジェクトを運営しています⁵。アナログ・デバイセズは同コンソーシアムの正式メンバーとして、まずは産業分野向けの専用モジュールを対象とした開発を進めています。

当社は、様々な専用モジュール用のROSドライバを開発しています。また、開発済みのドライバの有用性を実証した上でROSの機能を活用するために、社内使用する自律型モバイル向けプラットフォームとしてAnalog Devices Autonomous Mobot (ADAM)を開発しました（図4）。



図4. ADAMの外観

ADAMの概要

ADAMは、ROSと他のROS対応デバイスと連携する形で動作します。このプラットフォームは、モバイル・ロボット・アプリケーション、具体的には自律型のナビゲーション・アプリケーションにアナログ・デバイセズのROSドライバを組み込むことを証明しています。

図5は、ADAMのハイ・レベルのハードウェア構成を示したものです。ご覧のようにADAMは様々なモジュールによって構成されています。主なデバイスとしては以下のようなものが挙げられます。

- ▶ **ADIS16470**：高精度のジャイロ・スコープ、加速度センサー、磁気センサー、圧力センサーを組み合わせた多軸対応の慣性計測ユニット（IMU：Inertial Measurement Unit）です。主に、位置／方向の推定精度を高めるためのセンシング・フィードバック・デバイスとして使用されます。
- ▶ **ADBMS6948**：マルチセルに対応するバッテリー・モニタです。直列に接続された最大16個のバッテリー・セルを対象とし、全温度範囲において非常に高い精度で測定を行うことができます。
- ▶ **EVAL-ADTF3175D-NXZ**：市場で最も高い解像度を備えるCMOSベースのToFカメラです。その機能は、深度の計算／処理機能、レーザ・ドライバ、パワー・マネージメント機能、リファレンスのファームウェア／ソフトウェアを備える開発ツールによって補完されます。

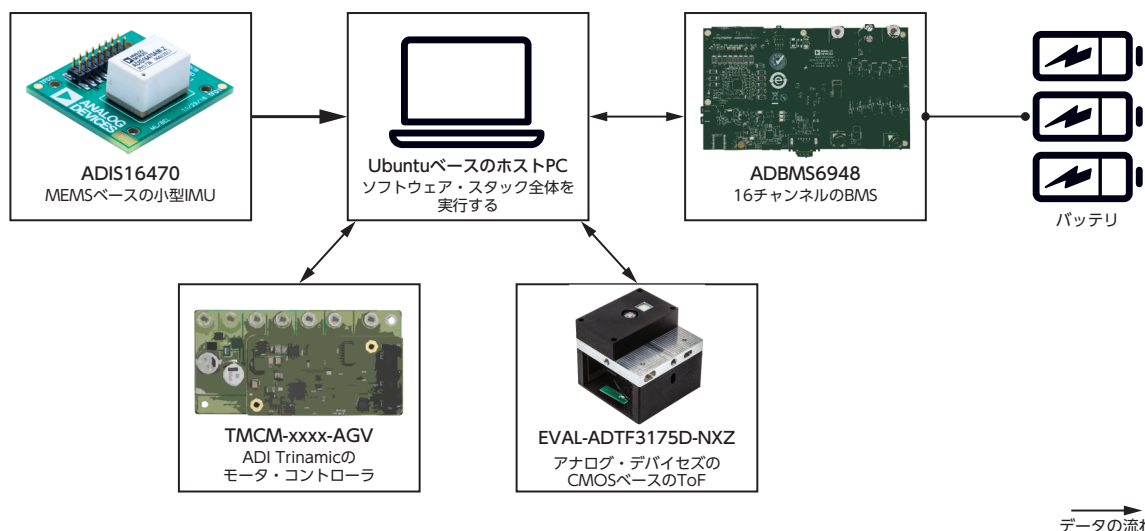


図5. ADAMのハイ・レベルのハードウェア構成

- ▶ ADI Trinamic のモータ・コントローラ：組み込みモーション・コントロールの用途に向けたボード・レベルの完全なソリューションです。ADI Trinamic が提供するモーション・コントロールの専門技術に加え、アナログ・デバイセズのアナログ IC 向けプロセス技術、電源設計のスキルが適用されています¹。

図6は、ADAMにおいてROSに関連する部分のハイ・レベルのアーキテクチャを示したものです。このアーキテクチャでは、ROSドライバと、自律型ナビゲーションに必要な複数のアプリケーション/アルゴリズムのノードを使用します。IMUのデータ (/imu/data_raw) とADI Trinamicのモータ・コントローラのフィードバック (/tmc_info) を、姿勢の推定のための入力として使用する形でロボットのオドメトリ (/odom) が実装されています。LiDARのデータ (/scan) は、SLAM (Simultaneous Localization and Mapping) のアルゴリズムでマップを生成する際に主要な入力として使用します。ToFのデータ (/image_raw) も、他のSLAMアルゴリズムの入力として使用できます。move_baseのノードは、ユーザからの目標姿勢の指示 (/goal) を保持しており、速度のコマンド (/cmd_vel) をADI Trinamicのモータ・コントローラに送信します。

ADI Trinamicのモータ・コントローラ用ROSドライバ

ADI Trinamicのモータ・コントローラ (TMC : Trinamic Motor Controller) は、組み込みモーション・コントロールの用途に向けたボード・レベルの完全なソリューションです。ADI Trinamic が提供するモーション・コントロールの専門技術に加えて、アナ

ログ・デバイセズのアナログ IC 向けプロセス技術や電源設計のスキルが適用されています¹。TMCは、単軸/多軸のステッパ・モータやブラシレスDC (BLDC) モータなど、様々な種類のモータをサポートします。また、CAN (Controller Area Network)、EtherCAT[®]、RS-232、RS-485、USBなどのインターフェースに対応しています。更に、Trinamic Motion Control Language (TMCL[™])、CANopen[®] over EtherCAT (CoE)、CANopen、Modbusといった様々なプロトコルをサポートします¹。

ユーザによるアプリケーション開発の支援に向けては、[TMCL-IDE](#)という統合開発環境 (IDE) が提供されています。同IDEを利用することにより、各種モジュールを簡単に再プログラムすることができます。TMCL-IDEでは、スタンドアロンの動作に向けたTMCLと標準のCANopen[®] プロトコルのうちいずれかを使用します。このIDEを使用することで、パラメータの設定、リアルタイムでのデータの視覚化、スタンドアロンのアプリケーションの開発/デバッグを容易に実施できます。

TMCを採用すれば、新たなレベルのインテリジェントなアクチュエータを実現できます。ROSがロボティクスの分野でますます広く活用されるようになり、各モジュール用のROSドライバなど、製造分野や産業用オートメーション分野などを対象としたユース・ケースを更に拡大するための開発が行われるようになりました。具体的には、各ROSドライバによって次のような処理が行われるようになることが期待されています。

- ▶ モータの速度、位置、トルクの制御
- ▶ モータ・コントローラとモータの情報のモニタリング

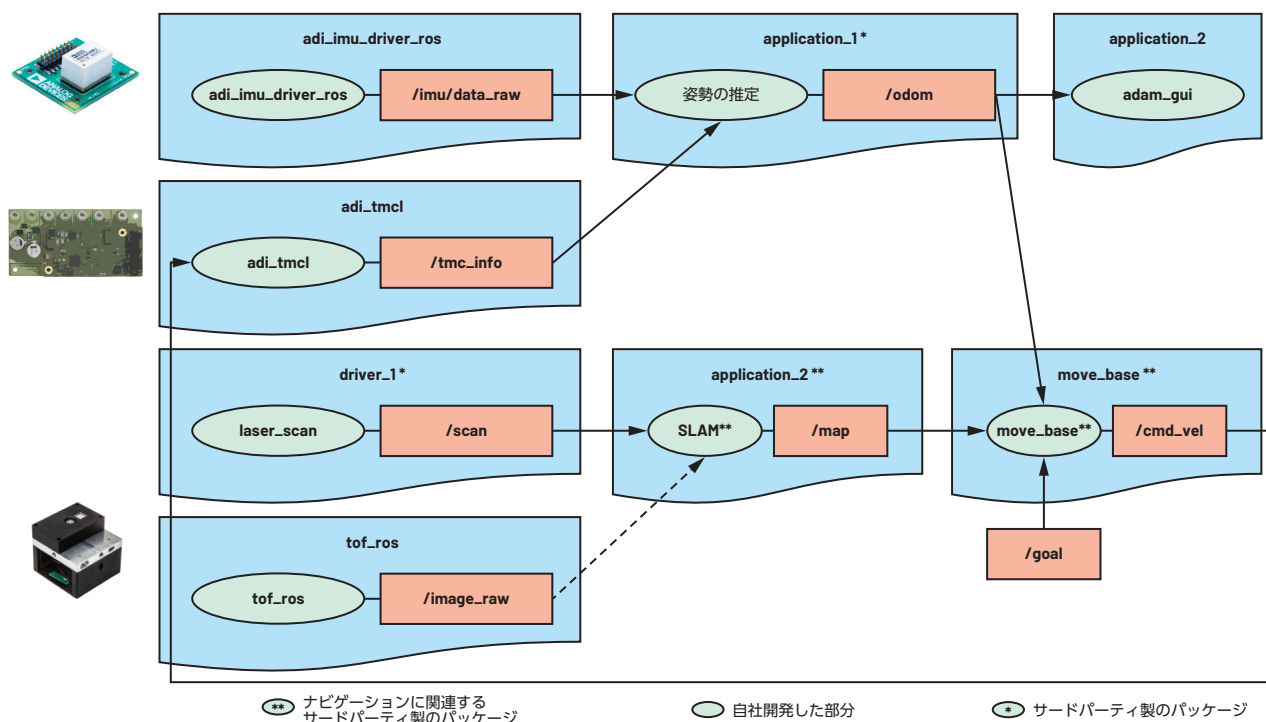


図6. ADAMのナビゲーション・スタック。
ROSに関連する部分のハイ・レベルのアーキテクチャを示しています。

TMC用のROSドライバが提供する機能は、TMCL-IDEが提供するものに似ています。ただ、ROSドライバを使用すれば、システムのノードが、追加のドライバをインストールすることなく、それらのTMCを直ちに使用できるようになるという点が異なります。本稿の執筆時点では、アナログ・デバイセズのROSドライバはCANインターフェース（具体的にはSocketCAN）のみをサポートしています。他のインターフェースも開発中であり、まもなくサポートが追加される予定です。

現在サポートされているADI Trinamicのモータ・コントローラ・モジュール（TMCM）の一覧は、[こちら](#)からご覧になれます。

ソフトウェア・アーキテクチャ

図7に示したのは、adi_tmclのハイ・レベルのソフトウェア・アーキテクチャです。この図からわかるように、adi_tmclにはドライバを追加する必要はありません。Linuxベースの大半のシステムはSocketCANドライバをデフォルトでサポートしています。adi_tmclはこのドライバを採用しているので追加の必要がないということです。また、adi_tmclは独自のTMCLプロトコル・インタプリタを備えています。そのため、ユーザがTMCLに準拠する送信/受信コマンドを要求すれば、それを解釈することができます。最下位層のTMCL ROSノードは、ROSシステム上のディレクト・インターフェースをパブリッシャ、サブスクライバ、サービスの形で提供します。それぞれは、一連のパラメータによって設定可能な特定の機能を提供します。これについては、後ほど説明します。

機能

adi_tmclは、以下のような様々な機能を提供します。

- ▶ 様々な TMC ボードのサポート
- ▶ TMCL-IDE による TMC モジュールのワンタイムでの構成
- ▶ モータの運転/停止
- ▶ TMC とモータの情報の取得
- ▶ カスタムの TMC コマンドの実行
- ▶ 各軸のパラメータの値の取得
- ▶ 各グローバル・パラメータの値の取得
- ▶ 複数の TMC ボードの設定のサポート
- ▶ ROS システム/アプリケーションへの簡単な統合

これらの機能については、「Mastering Trinamic Motor Controllers with the ROS1 Driver」(ROS 1ドライバを搭載するTrinamicモータ・コントローラを使いこなす)という来月号の記事でサンプルを交えながら活用方法を示します。ぜひそちらもご覧ください。

まとめ

ADI Trinamicのモータ・コントローラを使用すれば、新しいレベルのインテリジェントなアクチュエータを実現できます。ROSが特にロボティクス分野でますます広く活用されるようになり、各モジュール用のROSドライバなど、製造分野や産業用オートメーション分野などを対象としたユース・ケースを更に拡大するための開発が行われるようになりました。

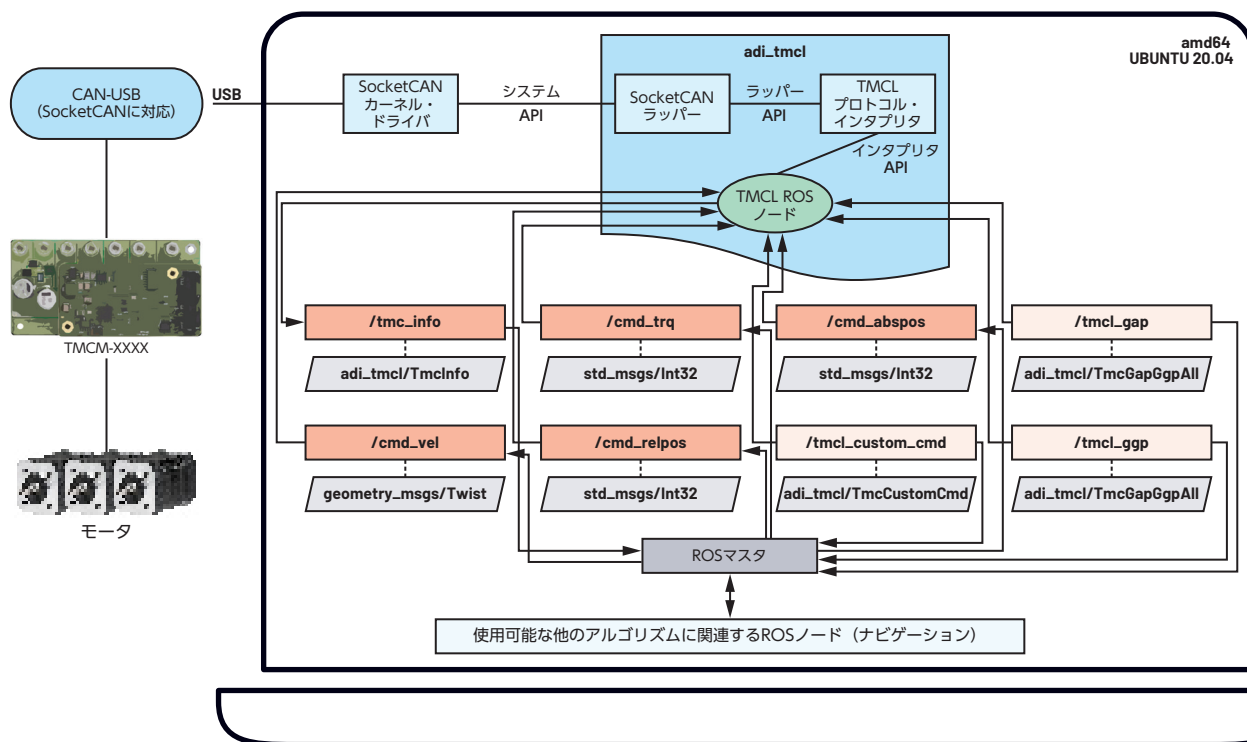


図7. adi_tmclのハイ・レベルのソフトウェア・アーキテクチャ

本稿では、ROSによって以下のような効果が得られることを示しました。

- ▶ 産業用のアプリケーション向けに拡張できるといった付加価値が追加される
- ▶ ROS の通信フレームワークにより、サードパーティの製品との相互運用性をより容易に実現できる
- ▶ お客様が、自社製のシステムにおいて、アナログ・デバイスズの製品を活用するための選択肢が広がる
- ▶ 新たな技術を素早く評価し、直ちに使い始めることができる

詳細については、アナログ・デバイスズの[産業用ロボット](#)のページをご覧ください。

参考資料

¹ [「ADI Trinamic ハードウェア：モータ・コントロールおよびモーション・コントロール向け」](#) Analog Devices

² [「ros2/ros2_documentation」](#) GitHub, Inc.

³ [「Understanding ROS Nodes. \(ROS ノードについて理解する\)」](#) ROS.org.

⁴ [「Services \(サービス\)」](#) ROS.org.

⁵ [「ROS-Industrial」](#) ROS.org.

[「モーション・コントロール・アプリケーション向けの工業用の通信プロトコルとインターフェース」](#) Analog Devices

次のステップ

- ▶ ADI Trinamic のモータ・コントローラ用 ROS 1 ドライバについての解説記事を読む
- ▶ ADI Trinamic のモータ・コントローラ用 ROS 2 に関する記事（今後公開）を読む
- ▶ ADI Trinamic のモータ・コントローラ用 [ROS 1/ROS 2](#) ドライバのダウンロード
- ▶ ADI Trinamic の [モータ／モータ・コントローラ](#) の評価用ボードの購入



著者について

Krizelle Paulene Apostol は、アナログ・デバイスズのソフトウェア・システム・エンジニアです。2019年12月にフィリピン支社（カヴィテ）に入社しました。インテリジェント・モーション／ロボティクス・グループに所属し、フィリピン開発センターの一員として業務に従事。ROS（Robot Operating System）、Gazeboによるシミュレーション、ファームウェア、通信プロトコル、アルゴリズムの開発などを対象とした様々なプロジェクトに携わってきました。FAITH大学でコンピュータ工学の学士号を取得しています。



著者について

Jamila "Jam" Aria Macagba は、アナログ・デバイスズのシニア・ソフトウェア・システム・エンジニアです。2018年7月にフィリピン支社（カヴィテ）に入社しました。インテリジェント・モーション／ロボティクス・グループに所属し、フィリピン開発センターの一員として業務に従事。主に、ROS（Robot Operating System）に対応するドライバの開発／統合に注力しています。フィリピン大学ロスバニオス校で電気工学の学士号を取得しました。



著者について

Maggie Maralit は、アナログ・デバイスズのソフトウェア・システム設計エンジニアリング・マネージャです。2019年4月にフィリピン支社（カヴィテ）に入社しました。インダストリアル・モーション／ロボティクス・グループに所属。フィリピン開発センターの一員として業務に従事しています。現在は、フィリピンの拠点で産業用ロボティクスのプロジェクトを担当するエンジニアのグループを統括。2009～2010年にはHPのアプリケーション・スペシャリスト、2010～2013年にはCanon Information Technologies Philippinesのシニア・ソフトウェア・エンジニア、2013～2015年にはIonics EMSのファームウェア開発エンジニア、2015～2019年にはContinental Automotive Singaporeのシニア組み込みソフトウェア・エンジニアを務めていました。フィリピン大学ロスバニオス校（ラグナ州）でコンピュータ科学に関する学士号を取得しています。

