

ROSロボットで学ぶ 次世代のIoTアーキテクチャ

工学博士 田胡 和哉 著

コロナ社

【本書のご利用に当たって】

本書で解説している内容を実行・利用したことによる直接あるいは間接的な損害に対して、著作者およびコロナ社は一切の責任を負いかねます。利用はすべて利用者個人の責任において行ってください。

本書に掲載されているソフトウェアおよび Web ページ URL に関する情報は、本書執筆時点のもので、将来にわたって保証されるものではありません。特に、ROS パッケージは仕様や実行条件の変更が頻繁にあります。これによっては、本書で解説しているサンプルなどが正常に動作しなくなることもあるので、あらかじめご了承ください。

本書の発行に当たって、読者の皆様に問題なく実践していただけるよう、できる限りの検証を行っていますが、以下の環境以外では構築・動作を確認していないので、あらかじめご了承ください。

PC 本体：Windows10 Education 64 bit (CPU：Intel Core i7-9700, メモリ：32 GB)

開発環境：WSL2 Ubuntu 22.04.1 LTS, Docker Desktop V4.18.0, VSCode 1.77.3,
その他環境は、docker 内で随時インストール

また、上記環境を整えたいかなる状況においても動作が保証されるものではありません。ネットワークやメモリの使用状況および同一 PC 上にあるほかのソフトウェアの動作状況によって、本書のプログラムが動作できなくなることがあります。併せてご了承ください。

本書の購入者に対する限定サービスとして、本書に掲載している参考文献リンク、Python のソースコードと必要なデータファイルは、以下の Web ページからダウンロードできます。ぜひご利用ください。

<https://momoi.org/books/iot>

まえがき

本書の執筆中に、米国の電気自動車メーカーのテスラが、作業に用いることができる人型ロボットの開発を表明しました。ロボットの新しい時代の幕開けとも捉えられます。

これまで、工業用ロボットやFA（Factory Automation）の分野で、わが国は主導的な立場を保ってきました。また、本田技研工業のASIMOは二足歩行技術に関して驚くべき先進性を示しました。世界と歴史に誇るべき大きな成果だと考えられます。しかしながら世界では、さらに新しい方向性に基づく技術が着実に育ってきています。そこでは、クラウドや深層学習などの新しい技術が採用されているばかりでなく、技術開発の手法自体も変化しつつあります。

これを例えれば、スマートフォン（以降スマホと表記）の出現に比肩しうる、非常に大きな変革と新たな可能性が近付いているともいえます。スマホとクラウドによって、技術だけでなくIT産業の構造自体が変わったといっても過言ではありません。同等の規模の変化が近付いているのであれば、ぜひチャレンジしてみようではありませんか。うまく波に乗れるのと取り残されるのでは、雲泥の差があります。

そのためには、まず全体として何が起きつつあるのか把握することがたいへん重要です。例えば、IoT（Internet of Things）という言葉における“Things”とは、最初はセンサを指していましたが、最近では、ロボットのようなセンサ情報に基づいて自分で環境に働きかけることができる能力を持った機構も含まれるようになってきました。そうすると、自動運転もIoT技術の一つとなります。また、“Internet”とは、単なるデータ通信機構ではなく、情報や価値の共有の意味合いが含まれており、そのような視点が必要になります。このように、“IoT”は非常に広い領域をまとめて指す言葉となっており、基本的な骨格

をまず明確に把握する必要があります。個々の技術を深めることは、その後でも遅くはありません。

そこで本書では、自律移動ロボットに焦点を絞り、これに関連するすべての技術を山に登って周囲の地形を確認するように、全体的に俯瞰できるようにすることを試みます。自律移動ロボットとは、車輪や無限軌道を用いて自分の知能以移動する機能を持ったロボットで、倉庫などの屋内で物を移動するために利用が広がっています。ここではさらに、作業するためのアームを持つ場合も含めます。

この分野では、最近、ROS (Robot Operating System) が利用されることが増えてきました。しかしながら ROS は、この分野全体ではありません。本書では、ROS を道具の一つとして捉え、全体を従来よりさらに具体的かつ手軽に見通せるようにすることを試みました。

また、各章末には「チャレンジしよう」と題し、その章で理解した知識をより深く自分のものにするための課題を準備しました。解答は準備しておりませんが、ヒントや手順を掲載しています。ぜひ取り組んでください。

さらに本書の最後では、エンジニアや研究者の勉強とキャリアについても触れます。受験勉強のように、いろいろな技術についてひたすら詳しくなることを目指すのではなく、何のためのどのような知識をどうやって獲得するのかをよく考える必要があるのは、分野を問いません。特に、エンジニアとして成功するためには、ビジネスの方向性と技術の全体像の双方を併せて考えられるようになっていくことが重要です。自律移動ロボットを例にとり、これについて考えてみることを通じて、全体のまとめとします。

なお、本書に関連したプログラムや訂正など、追加の内容を

<https://momoi.org/books/iot>

に掲示しますので、併せてご利用ください。

2023年9月

田胡 和哉

目 次

1. 準備作業

1.1 本書の構成	1
1.2 試行に必要な準備	4
1.3 実際の準備作業	6
補 足 資 料	10

2. 自律移動ロボットの事例

2.1 自律移動ロボットとは何か	12
2.1.1 自律移動機能	12
2.1.2 外界に働きかける機構	15
2.1.3 タスクプランニング	16
2.1.4 ロボットの機構	17
2.2 自律移動ロボットの利用	18
2.3 自律移動ロボットのハードウェア例	19
2.4 アームの例	20
2.4.1 軽量なロボットアーム	20
2.4.2 試作用自走型ロボットアーム	21
補 足 資 料	22
チャレンジしよう	22

3. ロボットハードウェア

3.1 アクチュエータとその制御	24
------------------	----

3.1.1	アクチュエータを用いる IoT デバイス	24
3.1.2	IoT デバイスに使用されるモータ	25
3.1.3	モータの制御とサーボ機構	28
3.1.4	モータの形状	29
3.1.5	I2C	30
3.2	センサ	31
3.3	伝統的なエッジ構成技術	34
3.3.1	Arduino	34
3.3.2	メイン CPU	35
3.3.3	エッジ基板の実装	36
3.3.4	処理性能	38
3.4	新しいエッジ構成技術	40
	補足資料	45
	チャレンジしよう	49

4. ROS と自律移動ロボットのソフトウェア

4.1	自律移動機能の実装	51
4.1.1	想定するハードウェア	51
4.1.2	実現すべき機能	52
4.2	ROS のプログラミング	55
4.2.1	ドキュメント	56
4.2.2	インストール	57
4.2.3	ROS のプログラム実行	58
4.2.4	ROS の通信	58
4.2.5	プログラムの実行	60
4.2.6	ROS の開発環境	62
4.2.7	Python による通信プログラム	62
4.3	ROS による自律移動の実現	64
4.3.1	ROS におけるモータの制御	66
4.3.2	モータ制御のシミュレーション	67
4.3.3	MCU との連携方法	68

4.3.4	odometry	68
4.3.5	tf2	70
4.3.6	Python による tf2 の利用	72
4.3.7	障害物検出とコストマップ	75
4.3.8	Navigation2	78
4.3.9	自律移動のシミュレーション	81
4.3.10	ROS の使いこなし	89
4.4	ROS ナビゲーション機構利用のまとめ	90
4.5	自律移動機構の原理入門	92
4.5.1	自己位置推定	92
4.5.2	SLAM	95
	補 足 資 料	97
	チャレンジしよう	102

5. ロボットアームの制御ソフトウェア

5.1	制 御 の 目 標	103
5.1.1	ロボットが行う作業	103
5.1.2	マニピュレータのモデル化	104
5.1.3	パスプランニング	105
5.1.4	トラジェクトリプランニング	106
5.1.5	アームのモーシオンプランニング	107
5.1.6	把持プランニング	107
5.2	ROS を用いたアーム制御	108
5.2.1	全 体 の 構 造	108
5.2.2	形状のモデリング	109
5.2.3	試 行	110
5.3	ROS による把持動作の実現	113
5.4	ロボットのモーシオンプランニングの原理入門	114
5.4.1	C-Space	114
5.4.2	自律移動のモーシオンプランニング	115
5.4.3	トラジェクトリプランニングの枠組み	117

5.4.4 経路計画	119
補足資料	120
チャレンジしよう	121

6. アプリケーション

6.1 ピッキングロボットによる在庫管理の思考実験	122
6.1.1 全体の構成	122
6.1.2 在庫管理システム	123
6.1.3 入出庫システム	124
6.1.4 実装方法	126
6.2 Behavior Tree によるタスク実行	127
6.3 ロボットの自律行動	129
6.3.1 ティーチング	129
6.3.2 行動の自律化	129
6.3.3 自律行動の必要性	130
6.4 全体システム	131
補足資料	132
チャレンジしよう	133

7. AI とコンピュータビジョン

7.1 本章で触れる事柄	135
7.2 コンピュータビジョンとロボット	137
7.2.1 OpenCV	137
7.2.2 ROS からの OpenCV 利用	139
7.3 visual SLAM	143
7.3.1 開発現状の把握	145
7.3.2 対象画像の特徴点抽出による方法	145
7.3.3 画素の直接比較による方法	147
7.3.4 ロボットへの応用	149

7.4 機械学習と深層学習	149
7.4.1 機械学習	150
7.4.2 深層学習	150
7.4.3 深層学習のフレームワーク	155
7.5 物体検出, セグメンテーションと Spatial AI	156
7.5.1 物体検出	156
7.5.2 セグメンテーション	157
7.5.3 visual SLAM への深層学習の導入	157
7.5.4 Spatial AI	158
7.6 深層強化学習とその活用	159
7.6.1 強化学習	159
7.6.2 深層強化学習	161
7.6.3 深層強化学習の改良	161
7.6.4 深層強化学習の実装	162
7.7 現在できることとできないこと——AI利用のまとめ——	164
補足資料	166
チャレンジしよう	170

8. システム基盤

8.1 クラウド技術	171
8.1.1 仮想化	171
8.1.2 VM	173
8.1.3 コンテナ	173
8.1.4 コンテナオーケストレーション	175
8.2 エッジのソフトウェア	178
8.2.1 RTOS	178
8.2.2 開発環境	179
8.2.3 専用ハードウェアへの対応	181
8.2.4 深層学習を用いた現実的なエッジシステムの実現	182
8.3 QoS管理とフォグコンピューティング	182
8.3.1 ネットワークとその仮想化	183

8.3.2	DDS	184
8.3.3	フォグコンピューティング	185
8.4	分散処理モデル	187
8.5	次世代のIoT基盤と技術課題	189
8.5.1	解くべき課題	189
8.5.2	システム像	190
	補足資料	193
	チャレンジしよう	195

9. エンジニアによる価値創造の具体像

9.1	エンジニアとしてすべきこと	197
9.1.1	エンジニアの職種	197
9.1.2	製品・サービスの提供におけるエンジニアの役割	199
9.1.3	エンジニアとして何をすべきか	201
9.1.4	試作の勤め	202
9.2	自律移動ロボットによる社会の変革	203
9.2.1	対象となる産業分野	203
9.2.2	ロボットによる物流業の変革	204
9.3	新しい開発エコシステムの出現	208
9.3.1	ROSの生かし方	209
9.3.2	コンテナの持つ意味	210
9.4	Technology Outlook	210
9.4.1	ロボットによる物流革命の可能性	210
9.4.2	技術開発スタイルの変化	211
9.4.3	戦略	211
	補足資料	213
	チャレンジしよう	213
	あ と が き	215
	索 引	216

1

準備作業

本章では、学習の準備を行います。本書の全体構成を理解してください。各章は相互に密接に関連した内容なので、部分だけを取り出して理解することは困難ですが、必ずしも最初の章から順番に読む必要はありません。また、実際のプログラムによる試行を行うので、環境の整備が必要です。標準的には、Windows10, 11によるPCを用いることを想定しています。

1.1 本書の構成

本書は、アーム付きの自律移動ロボットを例にとり、それを構成する全技術を見ることにより、IoTシステムの具体的な全体像を描き出すことを主目的としています。これに対応して、要素ハードウェア、ナビゲーション機構、アーム制御機構、機械学習、システムアーキテクチャを取り上げ、なるべく原理に遡った理解を、できうる限り現実的な対象を用いて理解できるように試みます。

2章では、試作目的を中心にロボットの具体像を紹介します。

3章では、要素ハードウェアとして、センサ、アクチュエータのほかに、ロボット実現に用いるエッジ用GPU (Graphics Processing Unit)、FPGA (Field Programmable Gate Array)、RISC-Vについても触れます。

4章では、自律移動を実現するナビゲーション機構を見ます。**自律移動機能**とは、人間の助けを借りずに自らの判断で目的地まで移動できる機能を指します。このための機構として、ROS2を例にとり、SLAM (Simultaneous Localization And Mapping) について触れます。また、自律移動機能の一般的な原理について、数式によらない入門的な解説を試みます。

5章では、アーム機構の制御に関してROS2を例にとり、アームを動かす計画を作成するモーションプランニング技術について触れます。また、その一般的な原理について、数式によらない入門的な解説を試みます。

6章では、アームを持った自律移動ロボットの具体的なアプリケーションの例として、倉庫への適用を思考実験として検討します。これを通じ、ナビゲーション機構とアーム機構を組み合わせ、意味のある仕事を実行するための方法について、タスクプランニングを主体になるべく具体的に解説します。

7章では、機械学習に関して、深層学習を用いた画像処理と強化学習の実現について触れます。

8章では、ロボットをクラウドから制御するためのアーキテクチャ関連技術として、コンテナとそのオーケストレーション、**フォグコンピューティング** (fog computing) などに触れます。本書でテーマとなるIoTの観点から、ロボット機能を単体ではなく、クラウドも含めたシステム全体で実現するものと考えて検討を進めます。

9章では、特に物流を例にとり、技術とビジネスの関係、および動向予測を試みます。また、これに対応してエンジニアのキャリアについても考えます。学習の動機となるので、ここから読み始められても結構です。

このように、対象を限定することによって、初めて詳細な全体像を具体的に描き出すことが可能になります。その過程で、**ROS** (Robot Operating System) やフォグコンピューティングなどのプラットフォームとなりうるものについても触れます。プラットフォームとは、技術的な側面を超え、その上で開発や利用のためのエコシステムを生み出して、活動させるためのものです。プラットフォームの方向性を見据えて行動することは、自分の技術を生かすためにきわめて重要です。

ROSは、ロボットのソフトウェアを実現するためのプラットフォームです。これまでのROS1に続いてROS2が出現しています。ROS1は、要素技術の研究開発プラットフォームとしての利用法が主でしたが、その成功を受け、ROS2では商業利用にも適用できるようにすることが目的となっており、本格

的な利用が始まると予想されます。以下では、ROS1でもROS2でも成り立ち、特に区別する必要のない事柄ではROSと表記します。

ROSは、ロボットを実現するために必要な機能のすべてを提供するスイートではなく、新たな機能を後からプラグインの形式で追加できるロボット実装のための技術プラットフォームとしての役割を持ちます。オープンソースに基づいて、多くの有効な技術開発の成果がROS上で使えるようになったのが、ROSが成功した理由です。今後、このような性格のプラットフォームをビジネスの場でどのように生かすべきか、技術の全体像を把握することによって継続的に検討してゆくことがきわめて重要です。

同様に、クラウドベンダもクラウドというプラットフォームを提供するプラットフォームとして、新しいハードウェアや深層学習に関する技術を取り込んで、どのようにIoTに関連するビジネスに対応してゆくか、熾烈な競争^{しれつ}を行っています。

本書では、このような現状認識に基づいて、ビジネスへの適用も視野に入れながら説明を行ってゆきます。各章の最後では、「補足資料」として、文献を掲げたり、補足説明を行ったりします。文献などの数を増やすのではなく、テーマごとにまとめてなるべく解説を付すようにしています。

各章の最後には、さらに「チャレンジしてみよう」として、章ごとの学習内容の振返りと課題出題を行います。演習などに用いることも考え、課題の正解は掲載しません。また、正解が限定されない課題もあり、よく考えていただくことを目標としています。その代わりに、調査方法など、解答のための道筋をなるべく解説するようにします。自分なりに試行し、考えた成果を書き下すことは学習においてきわめて重要ですので、積極的に実施するようにしてください。

課題のなかには、多くの手数がかかるものもあります。通常のレポート課題のように、すべてを完全に実施することが目的ではなく、自分で考えて挑戦してみることが趣旨であり、簡単に解けるようには設定されていません。あきらめることなく、粘り強い対応をお願いします。

本書に関連したプログラムや訂正など、追加の内容を

<https://momoi.org/books/iot>

に掲示しますので、併せてご利用ください。

1.2 試行に必要な準備

本書では、以下の各章で、ここで述べた各技術要素を見てゆきます。各要素に関しては、なるべく具体的な試行が行えるようにします。そのような試行では、特に ROS の利用が有効です。

ROS を用いた試行においては、Linux を利用するためのリテラシーがある程度必要です。あらかじめこれを身に付けるようにしておいてください [補足][†]。

試行の目的の一つは、アルゴリズムを具体的に理解することです。さらに、関係する要素を網羅することが可能となり、ロボット技術に関する全体像が見えてきます。説明を読むより、自分でプログラムを動かしてみたほうが、はるかによく理解できます。この目的には、ROS を動かせる PC があれば、そのシミュレータを用いて実際にプログラムを動かしてみることが可能です。個人が勉強する目的では、この環境でも十分役に立ちます。

これを可能にする標準的な環境として、Windows10 以上の OS が動き、WSL2 が有効化されている PC が挙げられます。さらに、Windows 側に VSCode をインストールし、WSL2 上で利用するためのプラグインもインストールしてください。以下では、WSL2 を単に WSL と表記します。

WSL を利用する場合、X ウィンドウが必須になります。X ウィンドウは、画面を表示する X サーバと、表示を要求する X クライアントから成り立っており、ROS には最初から X クライアント機能が組み込まれています。例えば、ROS のロボットシミュレーション結果は、多くが X ウィンドウ上にグラフィカルに表示されます。

したがって、今回は X サーバを用意する必要があります。Windows11 の新

[†] 章末の補足資料を参照して下さい。

しいバージョンでは、別途Xサーバをインストールする必要はありませんが、そうでない場合は、Windows用のXサーバであるVcXsrvの利用が便利です。

深層学習のプログラムを試してみる場合には、できるだけ高性能なGPUが付けられたPCを利用することが望ましいです。

さらに、docker環境をインストールすると効果的な場合もあります。Windows環境では、Docker Desktopが利用可能で簡単にインストールできます。ROSのプログラムがインストール済みのdockerイメージが配布されているので、非常に簡単にROSプログラムを試してみることが可能になります。また、バージョンなど、条件の異なる複数のインストールを試してみることも容易に行えるようになります。

最初に、WSLの有効化、VSCode、Xウィンドウ、Docker Desktopのインストールを行っておいてください。ROSの動かし方については、必要になった時点で、4章で説明します。

ロボットを実際に使用するにはシミュレータによる準備だけでは不足であり、センサの性能や制御プログラムの能力を、実際にロボットを運用して確かめてみる必要があります。これらをすべて本番用のロボットで行うことは効率的でない場合があります、そのときには試作用のロボットを用います。ソフトウェアの環境は、シミュレータを用いる場合と同一です。

試作用には、Turtlebot3のようなROS対応のロボットを購入することになります。最近では、このような目的に用いることが可能なロボットが複数販売されています。このとき、ROS対応であることを必ず確かめる必要があります。特に、車輪を駆動するモータの回転を検出するセンサであるロータリエンコーダは必須です。シミュレータ上で試したプログラムは、ほぼそのまま実ロボットでも試すことが可能です。

これとは別に、企業での試作では、システムの実現可能性を検証する目的でハードウェアの試作を行う場合があります。最近のIoT用CPUやGPUは、メモリなどの中核部品と併せてモジュール形状で売られている場合が多く、そのモジュールを利用したロボットの制御基板などを自作することは比較的容易に

あ と が き

本書では、ロボットを例にとったIoT技術について、ハードウェアからAIに至るまで総括的に見てきました。また、これに基づいた、Technology Outlookを得ることを試みました。山の上から地域全体を俯瞰するような視点が得られたでしょうか。これは、技術に基づいて新たな価値を創成しようとする際に、不可欠の作業です。皆さんも、ぜひここでの情報に基づいて戦略立案を試みてください。わが国のIT産業に最も欠けており、今後の充実が望まれる作業領域がここです。著者自身も、本書の執筆の過程で、ロボット技術とその利用に関して、オープン化など、これまでのアプローチとは大きく異なる技術、方法がとられるようになり、新しい事業の核が世界規模で構築されつつあることを強く感じました。

一方、本書での記述は、技術提供を指向する研究、開発活動には不十分で、特定の技術領域に関するより突っ込んだ調査が必要になります。しかし、どの技術領域を指向するにしても、事前に、ここで提供したような、全体像を描いた上で取り組むのは、不可欠なことであると考えられます。本書を、このような目的にもご活用いただければ望外の幸いです。

索引

【あ】		【く】		衝突検出	106
アクション	60	グラフベースの SLAM	96	自律移動機能	1, 51
アクチュエータ	12	クロスコンパイラ	180	深層学習	150
アーム	15	グローバルプランナ	55	深度推定	148
				【す】	
【い】		【け】		ステッピングモータ	27
インフレーション	77	ゲート機構	154	スループット	38
				【せ】	
【う】		【こ】		セグメンテーション	157
運動学	117	誤差逆伝播法	152	セマンティック SLAM	158
		コストマップ	76		
【え】		コンテナ	173	【そ】	
エコシステム	212	コンピュータビジョン	16, 135	倉庫管理システム	123
エージェント	160			ソケット	187
エッジ	17	コンボリューション層	153	測 距	32
エッジサーバ	186			疎な方法	145
エピソード幾何	146	【さ】		ソフトウェアの部品化	190
エンドエフェクタ	15	再帰型ニューラルネット		ソフトコア	42
		ワーク	154	ソフトリアルタイム	38
【お】		サービス	60	損失関数	152
オーケストレーションツール	175	サブスクリाइブ	59		
		サーボ機構	28	【た】	
		サーボモータ	15, 29	タスク	16
【か】				タスクプランナ	16
開始姿勢	16	【し】		タスクプランニング	16
仮説検証	199	四元数	70	多様体	165
活性化関数	151	自己位置推定	78, 92		
		姿 勢	15	【ち】	
【き】		自由度	105	超音波センサ	32
キーフレーム	147	終了姿勢	16	直流ブラシモータ	25
逆運動学	117	出 荷	122		
強化学習	150	順運動学	117	【て】	
教師あり学習	150	順伝播型ニューラル		ティーチング	129
教師なし学習	150	ネットワーク	154		
協働ロボット	17	ジョイント	13		

ディープラーニング		ハードリアルタイム	38		
フレームワーク	155	パブリッシュ	59	【め】	
転移学習	162	バンドル調整	147	メイン CPU	25
点 群	34			メッセージ	59
		【ひ】		メッセージパッシング	59
【と】		ピッキング	19	【も】	
トピック	59	ピッキングロボット	123	モーションプランナ	13
トラジェクトリプランニング		ピック アンド プレイス	19	モーションプランニング	13
	106			物のインターネット	206
トランザクション	123	【ふ】		【ゆ】	
		フィンテック	205	ユニオンマウント	175
【な】		フォグコンピューティング			
内発的な報酬	162		186	【ら】	
ナビゲーション	14	フォトインタラプタ	31	ラストワンマイル	204
		複雑系	165	ランドマーク	94
【に】		物体検出	156	【り】	
入 荷	122	不 変	175	粒子フィルタ	93
ニューラルネットワーク		ブラシレス直流モータ	28	リンク	13
	136	プラットフォーマ	206	リンク構造	13
【ね】		プーリング層	153	【る】	
ネットワークの仮想化	183	【へ】		ループクロージング	97
ネットワークファイル		ベイズ統計	92	【れ】	
システム	180	【ほ】		レジストリ	175
ネットワークブート	180	ポテンシオメータ	28	レスポンス	38
【は】		【ま】		【ろ】	
配位空間	114	マイクロカーネル	178	ローカルプランナ	55
パケットロス	185	マニピュレータ	103	ローコード	132
把 持	16, 104	マルコフ性	160	ロータリエンコーダ	28
把持点	107	【み】			
把持プランニング	16, 107	密な方法	147		
パスプランニング	105				
ハードコア SoC	42				

【A】

A-D コンバータ	28
AGV	18
AI	16
AMCL	92
AMR	18

Arduino

34

【B】

Behavior Tree	127
BT	127

【C】

CAN	31
CNN	153
C-Space	114

	[D]	Linux	4		[S]		
		LSTM	154		SE	197	
DDS	60, 184			[M]	server-client	60	
dense な方法	147	MCL	92	SIer	197		
docker	173	MCU	25	SIMD	43		
dockerfile	175	MIPI	33	SLAM	14, 95		
docker イメージ	175	MoveIt2	108	グラフベースの——	96		
DQN	161			SLAM Toolbox	87		
DWA 法	116			Smach ライブラリ	127		
DX	198			SoC	40		
	[E]	Navigation2	64	sparse な方法	145		
EC	205			Spatial AI	158		
	[F]	odometry	69	SPI	30		
		OMPL	119	SRDF	110		
FNN	154	OpenAI gym	163			[T]	
FPGA	1, 41	OpenCV	135	Task Constructor	113		
	[G]			Technology Outlook	210		
GAN	154	PID 制御	53	tf2	70		
Gazebo	82	publish-subscribe	59	ToF	32		
GPGPU	43	PWM	25	Turtlebot3	19		
GPIO	25					[U]	
GPU	1, 43			URDF	82, 109		
	[H]	QoS	182			[V]	
HDL	181	Q 学習	160	visual SLAM	35, 143		
Humble	56			VLIW	44		
	[I]	RGB-D カメラ	34	VM	171		
I2C	30	RISC-V	41	VPN	35		
IMU	33	RMI	188			[Y]	
IoT	1, 24	RNN	154	YOLO	156		
	[K]	ROS	2				
Kubernetes	176	ROS ノード	55				
	[L]	ROS パッケージ	55				
LiDAR	12, 32	roscore	60				
		RPC	188				
		RTOS	178				
		Rviz2	82				

— 著者略歴 —

1983年 筑波大学大学院工学研究科博士前期課程修了 (情報工学専攻)
1986年 筑波大学大学院工学研究科博士後期課程修了 (情報工学専攻)
工学博士
1986年 筑波大学助手
1988年 東京大学助手
1990年 日本 IBM 株式会社東京基礎研究所研究員
2002年 東京工科大学助教授
2010年 東京工科大学教授
2021年 東京工科大学名誉教授

ROS ロボットで学ぶ次世代の IoT アーキテクチャ

Learning Next Generation IoT System Design with ROS Robots

© Kazuya Tago 2023

2023年11月10日 初版第1刷発行

検印省略

著者 田 胡 和 哉
発行者 株式会社 コロナ社
代表者 牛来真也
印刷所 壮光舎印刷株式会社
製本所 株式会社 グリーン

112-0011 東京都文京区千石 4-46-10

発行所 株式会社 コロナ社

CORONA PUBLISHING CO., LTD.

Tokyo Japan

振替00140-8-14844・電話(03)3941-3131(代)

ホームページ <https://www.coronasha.co.jp>

ISBN 978-4-339-02938-3 C3055 Printed in Japan

(松岡)



JCOPY < 出版者著作権管理機構 委託出版物 >

本書の無断複製は著作権法上での例外を除き禁じられています。複製される場合は、そのつど事前に、出版者著作権管理機構（電話 03-5244-5088, FAX 03-5244-5089, e-mail: info@jcopy.or.jp）の許諾を得てください。

本書のコピー、スキャン、デジタル化等の無断複製・転載は著作権法上での例外を除き禁じられています。購入者以外の第三者による本書の電子データ化及び電子書籍化は、いかなる場合も認めていません。落丁・乱丁はお取替えいたします。