

研究論文

RT ミドルウェアを用いたロボット・IoT システムの構築

木村宏樹^{*1}、平出貴大^{*1}、酒井昌夫^{*2}、島津達哉^{*1}

Construction of Robot/IoT System Using RT-Middleware

Hiroki KIMURA^{*1}, Takahiro HIRADE^{*1}, Masao SAKAI^{*2} and Tatsuya SIMADZU^{*1}Industrial Research Center^{*1*2}

システム構築を効率化する方法として、ミドルウェアを用いたソフトウェアのコンポーネント化がある。本報では、ロボットシステム用のソフトウェアプラットフォームである「RT ミドルウェア」を用いて、ロボット・IoT システムを構築した。アームロボット、エリアセンサや表示灯など各機器のコンポーネントを作成し、これらを組み合わせてシステムを制御した。

1. はじめに

製造現場における生産性向上を目的にロボットや IoT 活用の取り組みが進められる中、中小企業においてもこれらの技術の利用への関心が高まっている。ロボットシステムや IoT システムの構築は、システムインテグレータなどのベンダーが担うのが一般的だが、低価格で高性能なハードウェアの普及やソフトウェア開発手法の発展により、専門外の人でも小規模なシステムの構築も難しいことではなくなってきている。

ロボットに関してはオープンソースソフトウェアやミドルウェア等^{1)~8)}、IoT に関しては各種支援ツールやモジュール化されたセンサ等によるシステム構築の簡易化や効率化の流れがある。しかし、中小企業に対して、これらの技術にアクセスする機会や習得へのモチベーションを高める機会が十分に提供されているとは言えない。このようなことから当センターでは、これらの技術活用の具体例を示すことを目的に、安価で手軽に利用可能な既存技術を用いて具現化したロボット・IoT システムとして組立・検査の自動化システムを構築してきた⁹⁾。本システムでは、シングルボードコンピュータ「Raspberry Pi」と I/O 基板で PLC(Programmable Logic Controller)を構築し、ラダー・プログラムの開発環境であるオープンソースソフトウェア「OpenPLC」⁸⁾を用いてロボットや各機器を制御した。しかし、デジタル入出力による ON/OFF 制御が基本となるため測域センサなどのセンサ値を扱いにくいこと、既存のプログラムの流用などのソフトウェアの再利用がしにくいことに課題があった。

そこで本報では、これらの課題に対応するためロボットシステム用のソフトウェアプラットフォームである

「RT ミドルウェア」^{1)~5)}を使用した。RT ミドルウェアを用いて構築したシステムにより、センサ値の取り扱いやソフトウェアの再利用が容易にできることを検証した。

2. 実験方法

2.1 RT ミドルウェア「OpenRTM-aist」

RT ミドルウェア(RT-Middleware: RTM)は、アクチュエータやセンサなどのロボットの機能要素ごとにソフトウェアモジュール「RT コンポーネント(RTC: Robot Technology Component)」を作り、それを複数組み合わせることでロボットシステム(RT システム)を容易に開発するためのソフトウェアプラットフォームのことである。本研究では、(国研)産業技術総合研究所が開発し、無料で利用できる RT ミドルウェア「OpenRTM-aist(Open-source and open architecture Robot Technology Middleware implemented by AIST)-2.0.1-RELEASE」³⁾を使用した。

OpenRTM-aist における RTC と RT システムの関係を図 1 に示す。RTC には他の RTC とデータやコマンドなどをやり取りするための「ポート」と呼ばれるインタ

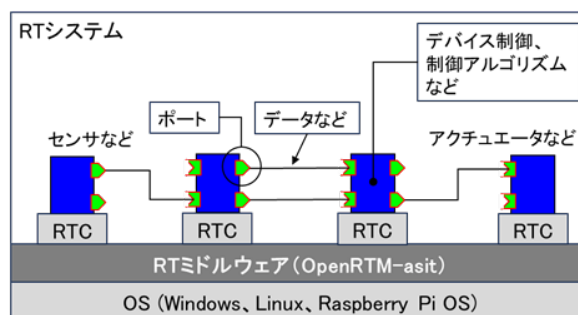


図 1 RTC と RT システムの関係

一フェースがあり、これらをソフトウェア上でつなぎ合わせることで RT システムを構築し、所望の機能を実現するものである。RTC の作成は、OpenRTM-aist のツール「RTCBuilder」を用い、ポート間の通信に関するコードなどの自動で生成される共通のソースコード(雛形コード)にデバイス制御や制御アルゴリズムなどの実現したい機能に関するコードを追記することで行った。

2.2 RTM によるロボット・IoT システムの構築

構築した組立・検査の自動化システムの外観を図 2 に示す。このシステムはボールペンの組立と検査をするもので主な機能は以下の 4 機能とした。

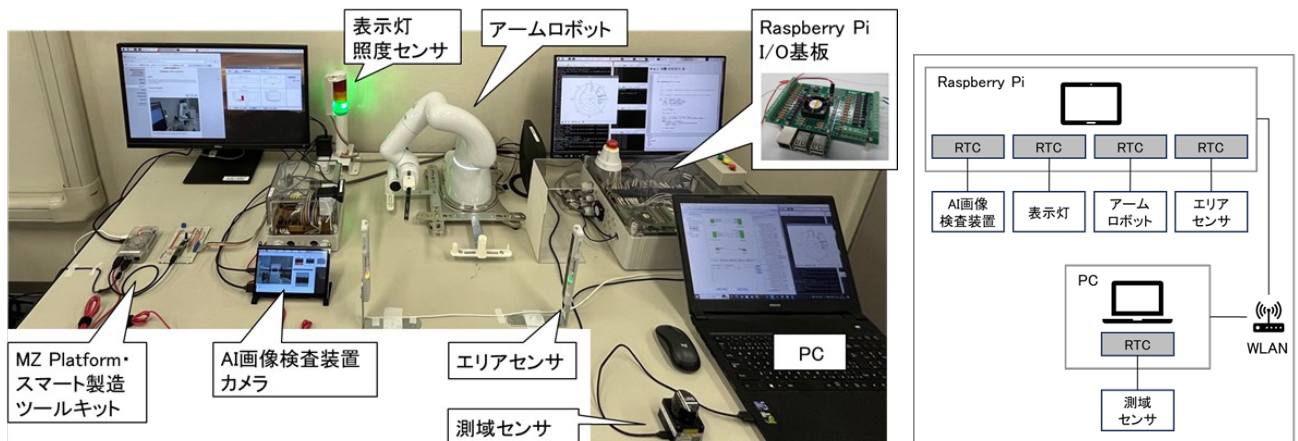
機能①:アームロボット((株)デンソーウェーブ: COBOTTA)がボールペンを組み立て、その組立の良否を AI 画像検査装置がカメラで撮影したボールペンの画像に対して AI による画像分類で判定する。

機能②:不良の場合に表示灯が橙点灯し、それを照度センサで検知して IoT ツール「MZ Platform・スマート製造ツールキット」¹⁰⁾により点灯回数を登録する。

機能③:人の侵入などでエリアセンサが遮光されるとアームロボットが停止し、表示灯が赤点灯する。
機能④:周囲を測域センサ(北陽電機(株): URG-04LX-UG01)が計測し、人の接近などにより定めた距離(閾値)以下となった場合に「アームロボットの動作速度が減速する。

本システムを制御するために構築した RT システムを

図 3 に示す。アームロボット、表示灯、エリアセンサ、AI 画像検査装置、測域センサ等の各 RTC のポートを接続することで、上記の機能を実現した。具体的には、組立不良判定時に表示灯を橙点灯させるために、AI 画像



(1) 構成機器

(2) 構成図

図 2 組立・検査の自動化システムの外観

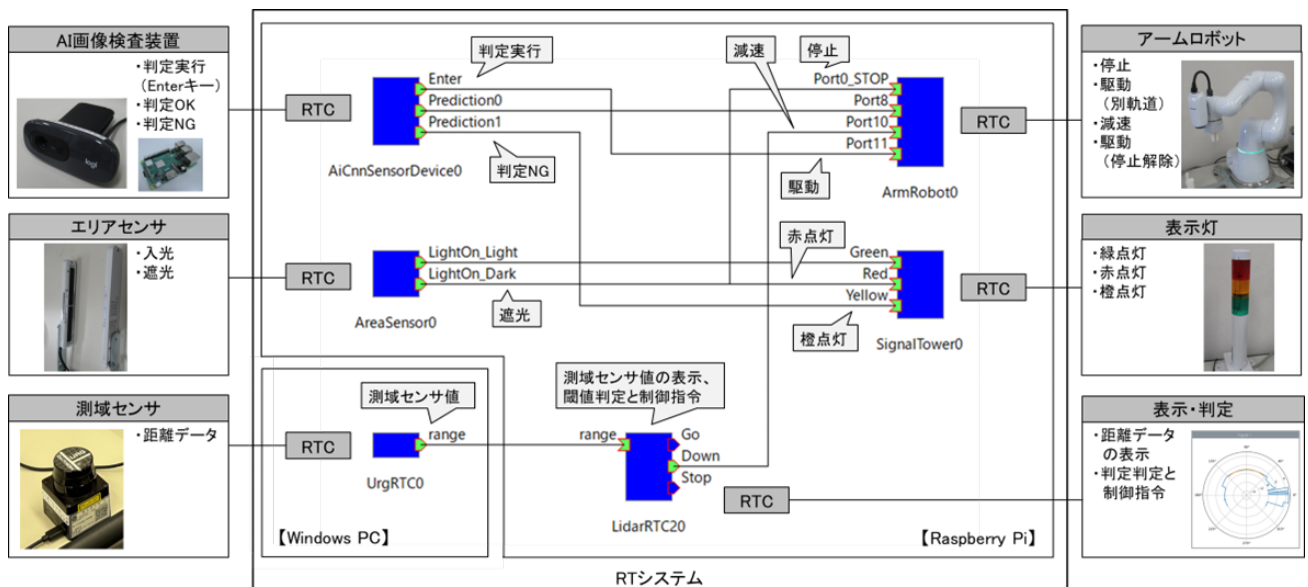


図 3 組立・検査の自動化システムを制御する RT システム

検査装置の RTC(AiCnnSensorDevice)の Prediction1 ポートと表示灯の RTC(SignalTower)の Yellow ポートを接続した。エリアセンサ遮光時にアームロボットを停止させるためにエリアセンサの RTC(AreaSensor)の LightOn_Dark ポートとアームロボットの RTC(Arm Robot)の Port0_STOP ポートを接続した。また、測域センサの計測値の表示と閾値による判定をするために、測域センサの RTC(UrgRTC)の range ポートとグラフ表示の RTC(LidarRTC2)の range ポートを接続し、閾値以下となった場合にアームロボットの動作速度を減速させるために LidarRTC2 の Down ポートと ArmRobot の Port10 ポートを接続した。

各 RTC の機能とポートの詳細を表 1 に示す。ポートのデータ型は TimedBoolean 型と RangeData 型の 2 種類を使用した。TimedBoolean 型は boolean 型のデータ (True/False) を扱い、ON/OFF の制御に使用した。RangeData 型は測域センサのスキャン範囲における距離データなどセンサ値を扱った。機能に記した内容の実

現のため、ポート値の判定や Raspberry Pi の GPIO の制御、センサ計測値のグラフ表示などを実行するコードを Python で十数行から数十行程度、雛形コードに追記して RTC を作成した。なお、測域センサの RTC (UrgRTC)は、URG-04LX-UG01 向けに C++で作成された既成の RTC¹¹⁾を利用して Windows PC で実行した。これにより、異なるプログラム言語で作成され、異なる OS 上で実行される RTC が混在する RT システムの動作について検証した。

2.3 RTC による RT システムの再構築の容易性の検証

2.2 に挙げたシステムの 4 機能のうち機能③を以下の機能⑤のように変更することで、RT システム再構築の容易性を検証した。

機能⑤:人の侵入などで測域センサが計測する距離が定めた距離(閾値)以下となった場合にアームロボットが停止し、表示灯が赤点灯する。

RTC のポートの接続先を変更して再構築した RT システムを図 4 に示す。LidarRTC2 の Stop ポートを

表 1 RTC の機能とデータポートの詳細

AI 画像検査装置の RTC [AiCnnSensorDevice]		アームロボットの RTC [ArmRobot]	
機能	Raspberry Pi の GPIO から AI 画像分類の判定状況と結果(良/否)を取得し、ポートに値を設定して送信	機能	ポート値を取得して駆動/減速/停止を決定し、Raspberry Pi の GPIO に出力
ポート	TimedBoolean 型 Enter: 判定実行(True/False) Prediction0: 判定・良(True/False) Prediction1: 判定・否(True/False)	ポート	TimedBoolean 型 Port0_STOP: 停止(True/False) Port8、Port11: 駆動 (True/False) Port10: 減速(True/False)
エリアセンサの RTC [AreaSensor]		表示灯の RTC [SignalTower]	
機能	Raspberry Pi の GPIO からエリアセンサ値を取得して遮光/入光を判定し、ポートに値を設定して送信	機能	ポート値を取得して表示灯の点灯/消灯を決定し、Raspberry Pi の GPIO に出力
ポート	TimedBoolean 型 LightOn_Light: 入光(True/False) LightOn_Dark: 遮光(True/False)	ポート	TimedBoolean 型 Green: 緑点灯(True/False) Red: 赤点灯(True/False) Yellow: 橙点灯(True/False)
測域センサの RTC [UrgRTC]		グラフ表示の RTC [LidarRTC2]	
機能	測域センサの計測値を取得し、ポートに値を設定して送信	機能	ポート値を取得して測域センサの計測値をグラフ表示(極座標)。距離を評価して速度を決定し、Raspberry Pi の GPIO に出力
ポート	RangeData 型 range: 測域センサの計測値	ポート	RangeData 型 range: 測域センサの計測値 TimedBoolean 型 Go、Stop: 駆動、停止(True/False) Down: 減速(True/False)

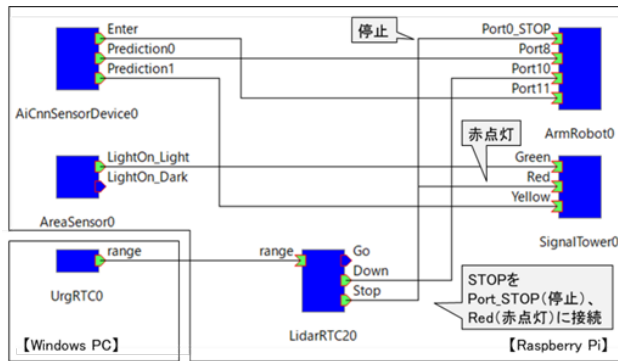


図4 再構築した RT システム

ArmRobotのPort0_STOPポートとSignalTowerのRedポートに接続を変更し、改修したRTシステムの動作について検証した。

3. 実験結果及び考察

図3に示したRTシステムの動作確認を行った。図5は、エリアセンサに手をかざして遮光した際、アームロボットが停止して表示灯が赤点灯した様子である。AreaSensorのLightOn_DarkポートとSignalTowerのRedポートの接続により所望の機能を実現した。図6に測域センサの計測値をグラフ表示した結果を示す。スキャン範囲45°から135°における距離データの最小値が定めた閾値(1m)以下となった際にアームロボットの動作速度が減速した。UrgRTCとLidarRTC2のrangeポートの接続、LidarRTC2における閾値判定及び



図5 エリアセンサによる表示灯の制御(赤点灯)

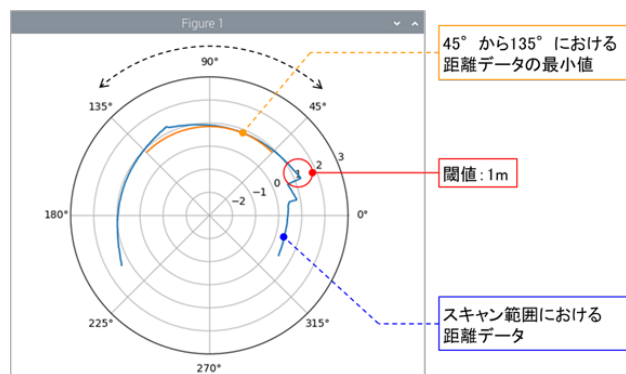


図6 測域センサの計測データと閾値判定

DownポートとArmRobotのPort10ポートの接続により所望の機能を実現した。これらの結果により、作成したRTCと既存RTC間や異なるOS間、異なるプログラム言語間も含めてRTC間のポート接続によりシステム構築が可能であることを確認した。

同様に、図4の再構築したRTシステムの動作確認を行った。測域センサの距離データの最小値が定めた閾値(0.5m)以下となった際にアームロボットが停止し、表示灯が赤点灯した。この結果により、RTCのポートの接続先を変えるだけでシステムを変更することができ、ソフトウェアの再利用が容易にできることを確認した。

4. 結び

本研究の結果は、以下のとおりである。

- (1) RT ミドルウェア(OpenRTM-aist)を用いて、ロボット・IoTシステムを構成する各機器のRTCを作成し、既存のRTCと組み合わせてRTシステムを構築した。
- (2) 課題であったソフトウェアの再利用に対し、既存のRTCの利用とポートの接続先の変更により、所望の機能の実現が容易にできることを確認した。
- (3) 課題であったセンサ値の扱いに対し、測域センサを例にグラフ表示や閾値判定などが実現できることを確認した。

文献

- 1) 大原賢一: 計測と制御, **57**(10), 679(2018)
- 2) 原田研介, 辻徳生: 計測と制御, **57**(10), 711(2018)
- 3) 国立研究開発法人 産業技術総合研究所: OpenRTM-aist, <https://www.openrtm.org/openrtm/ja>, (2024/04/10)
- 4) 安藤慶昭: 計測と制御, **57**(10), 682(2018)
- 5) 安藤慶昭: 日本ロボット学会誌, **34**(6), 366(2016)
- 6) Open Robotics: ROS Robot Operationg System, <https://www.ros.org/>, (2024/04/10)
- 7) 岡田慧: 計測と制御, **57**(10), 688(2018)
- 8) Open Source PLC Software: OpenPLC, <https://autonomylogic.com/>, (2024/04/10)
- 9) 木村宏樹, 酒井昌夫, 島津達哉, 平出貴大: あいち産業科学技術総合センター研究報告, **12**, 26(2023)
- 10) 国立研究開発法人 産業技術総合研究所 製造技術研究部門: MZ プラットフォームユーザー会, <https://ssl.monozukuri.org/mzplatform/>, (2024/04/10)
- 11) Yuki Suga: sugarsweetrobotics/UrgRTC, <https://github.com/sugarsweetrobotics/UrgRTC>, (2024/04/10)