

RoboCup Logistics League用通信プログラムを搭載した 組込機器の作成と評価

The Development of Embedded Devices Communicating with RefBox of RoboCup Logistics League

山北 善輝¹ 辻 和輝¹ 植村 渉^{1*}

¹ 龍谷大学

¹ Ryukoku University

Abstract: 日本国内で FESTO 社の全方位移動型ロボットである Robotino を用いる競技として、技能五輪全国大会（移動式ロボット職種）と、その予選に位置する若年者ものづくり競技大会（ロボットソフト組込み職種）がある。その競技大会の方向性が、従来のロボット制御のプログラミングの技術だけでなく、ロボットそのものを作成する能力も問うようになり、2016 年から共通のロボットではなく自作ロボットを使う方針に変わった。現在は Robotino から自作ロボットへの移行期間となっており両方のロボットが使えるが、経過処置 3 年目の 2018 年においても、Robotino で出場するチームは多い。それらのチームから、Robotino で参加できる大会を開催してほしいとの要望があった。現在、RoboCup では、Logistics League (RCLL) において、Robotino を用いた競技を行っている。ただし、RCLL では競技に必須の加工マシンが高価で、ドイツやヨーロッパ圏以外での運営が難しい。そこで、本研究では RCLL に必須の加工マシンを用いず、人の手で加工を行うことで、同等の競技の実施を目指す。その際、ゲームの流れを制御する審判用プログラム (RefBox) は、従来の RCLL のプログラムをそのまま用いるため、加工マシンとの通信部の置き換えが必要である。世界中で安価に手入れることができる小型マイコンである Raspberry Pi を使い、加工マシンの通信部を作成し、日本などヨーロッパ圏以外にて RCLL を開催することを目指す。

1 はじめに

若年者ものづくり競技大会では、ロボット制御の技術を競う競技としてロボットソフト組込み職種がある。この競技では、FESTO 社の全方位移動型ロボット Robotino を用いる。2015 年に世界大会の方針が変わり、ロボットそのものを作成する能力も問うようになった。現在は Robotino から自作ロボットへの移行期間となっており両方のロボットが使えるが、経過処置 3 年目の 2018 年においても、Robotino で出場するチームは多い。図 1 に過去 3 年間の参加チーム数と使用ロボットの割合を示す。Robotino を使うチームの割合は減ってきているが、チーム数そのものは減っておらず、2018 年は前年よりも増えている。

現在は移行期間のため、Robotino を用いても出場できるが、いずれ Robotino は使用できなくなる。そこで出場チームから Robotino で参加できる大会を開催してほしいとの要望があった。Robotino を用いる他の競技として RoboCup Logistics League (RCLL)[1] が

あるが、競技に必須の加工マシンは高価で運営が難しい。そこで、本研究では RCLL に必須の加工マシンを用いず、人の手で加工を行うことで、同等の競技の実施を目指す。その際、ゲームの流れを制御する審判用プログラム (RefBox) は、従来の RCLL のプログラムをそのまま用いるため、加工マシンとの通信部の置き換えが必要である。世界中で安価に手入れることができる小型マイコンである Raspberry Pi を使い、加工マシンの通信部を作成し、日本などヨーロッパ圏以外にて RCLL を開催することを目指す。以下、2 章では RoboCup Logistics League について紹介し、3 章にて使用した通信プロトコルである Modbus/TCP について述べる。4 章にて作製した加工マシンについて述べ、5 章でアンケート調査を行う。最後に 6 章でまとめる。

2 RoboCup Logistics League

RCLL ではシアン、マゼンダに分かれた 2 チームが同時に作業環境の探索、加工した製品の搬送、製品の納品の工程を移動式ロボット Robotino で行う競技である [2]。RCLL において運営が用意する機器として

*連絡先：龍谷大学理工学部電子情報学科
〒520-2194 滋賀県大津市瀬田大江町横谷 1-5
E-mail: wataru@rins.ryukoku.ac.jp

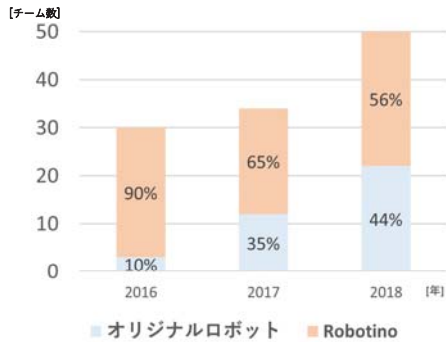


図 1: 2016年から2018年の3年間の若年者ものづくり競技大会における参加チーム数と使用ロボットの割合

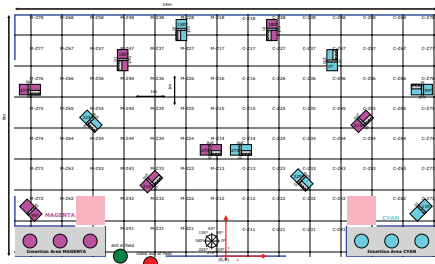


図 2: RCLL のゲームフィールド [2]

Modular Production System(MPS) (図 2) と RefBox がある。MPS は実際の工場での加工マシンを模したものであり、RefBox は審判の役割を果たすプログラムである。RefBox は試合中に、2種類の通信を行っている。Robotino とは、製品の搬送情報を Protocol Buffers を用いて通信している。MPS とは、加工の情報を Modbus/TCP を用いて通信している。ゲームフィールドを図 3 に示す。この図において、シアンとマゼンタの長方形が MPS を示しており、 8×14 [m] のフィールド内に各チーム 7 台、計 14 台の MPS が配置される。MPS は 5 種類あり、1 チームあたり Base Station(BS) $\times 1$ 、Cap Station(CS) $\times 2$ 、Ring Station(RS) $\times 2$ 、Delivery Station(DS) $\times 1$ 、Storage Station(SS) $\times 1$ で構成されている。図 2 における MPS の位置、角度は一例を示したものであり、実際の競技では両チームの配置が同じになるようにフィールドの y 軸に対称になるように、ランダムに配置される。

3 Modbus/TCP

Modbus/TCP[3] とは Modicon 社が 1979 年、同社のプログラマブルロジックコントローラ向けに開発したシリアル通信プロトコル Modbus を、イーサネットでも使えるように拡張したものである。通信には、Function code(図 4 の 7 バイト目)を用いたフォーマットを使う。

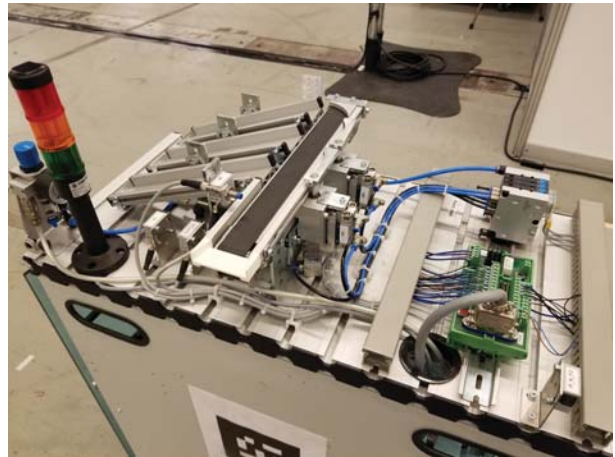


図 3: RCLL で用いる Modular Production System[2]

Modbus/TCPのフォーマット	Transaction identifier	Protocol identifier	Length	Unit identifier (識別子)	Function (コード)	Data (データ)
	byte 0,1	byte 2,3	byte 4,5	byte 6	byte 7	byte 8

図 4: Modbus/TCP 共通フォーマット [4]

MPS と RefBox の間では、 $0x04$ と $0x10$ のコードを用いる。これは、Read Input Register と Write Multiple Registers である。0 から 6 バイト目までは、機械的に値が決まるため、プログラム側で扱うのは 7 バイト目以降である。

$0x04$ のコードは、サーバ内のレジスタの値を読み取るコードである。Request として、7 バイト目に $0x04$ のコードを格納し、読み出す値の先頭番地 2 バイトを Starting Address で指定する。最後に Quantity of Input Registers にて、読み取るデータ数を 2 バイトで指定する。Response のフォーマットは、届いたリクエストと同じコードを 7 バイト目に格納し、返すデータのバイト数を Byte count に与える。そして Input Registers に実データを格納し返答する。

$0x10$ のコードは、サーバ内のレジスタの値を書き換えるコードである。Request として、7 バイト目に $0x10$ のコードを格納し、書き込むデータの先頭番地 2 バイトを Starting Address で指定する。次の Quantity of Registers にて、書き込むデータ数を 2 バイトで指定する。そして、その後ろに実際のデータが続く。Response は Request で受け取った Quantity of Registers までを、そのまま送り返せばよい。いずれもチェックサムは不要である。

Request		値
Function code (コード)	1 Byte	0x04
Starting Address (開始アドレス)	2 Byte	0x0000 から 0xFFFF
Quantity of Input Registers (入力レジスタの数)	2 Byte	0x0001 から 0x007D

Response		値
Function code (コード)	1 Byte	0x04
Byte Count (返信バイト数)	1 Byte	2N
Input Registers (入力レジスタの値)	2N Byte	入力レジスタの値

N: Request の入力レジスタの数

図 5: code0x04 Read Input Register[5]

Request		値
Function code (コード)	1 Byte	0x10
Starting Address (開始アドレス)	2 Byte	0x0000 から 0xFFFF
Quantity of Input Registers (入力レジスタの数)	2 Byte	0x0001 から 0x007B
Byte Count (返信バイト数)	1 Byte	2N
Registers Value (レジスタの値)	2N Byte	設定するレジスタの値

N: 入力レジスタの数 (Quantity of Input Registers)

Response		値
Function code (コード)	1 Byte	0x10
Starting Address (開始アドレス)	2 Byte	0x0000 から 0xFFFF
Quantity of Input Registers (入力レジスタの数)	2 Byte	1 to 123(0x7B)

図 6: code0x10 Write Multiple Registers[5]

4 DummyMPS の作製

RCLL の競技を行うには、審判のプログラムである RefBox が試合の流れを制御している。そのため、MPS との通信部を何らかの形で実装する必要がある。ここでは、組み込みマイコンである Raspberry Pi を用いて、この通信部を実装する。また、従来の RCLL では、MPS が製品の加工を自動で行っていたが、DummyMPS では、審判(人)が手作業で行う。置き換えた MPS を DummyMPS と呼び、図 7 に示す。

MPS では通信プロトコルとして Modbus/TCP(4) を用いている。Modbus/TPC のクライアント側は、パソコンなどが対応するため、多数の Linux のディストリビューション用のパッケージが存在する。一方、サーバー側は、産業用電子機器となるため、一般ユーザが触る必要性がなく、情報が少ない。そこで、RefBox と



図 7: 作製した DummyMPS

MPS 間の通信パケットを解析することで、サーバー側の通信を確認して、同様の通信を行うプログラムを作成する。

RefBox と MPS との間では、クライアント・サーバ方式で通信が行われており、RefBox がクライアント、MPS がサーバとして機能している。0x04 の Function code を用いて、RefBox は MPS の状態を確認する。本来であれば、MPS の状態が変化したときのみ通信を行えば良いが、問い合わせを開始できるのがクライアント側のため、RefBox は常時 MPS に状態確認を行う。0x10 の Function code を用いて、RefBox が MPS のシグナルを制御し、MPS の状態を視覚的にわかるようにしている。

RefBox と通信機能を持った DummyMPS を用いることで、シグナルの変化から試合状況を視覚的に把握できるようになる。

5 アンケート調査

2018 年 12 月に愛知県岡崎工業高等学校で説明会を開催した。参加した愛知県、三重県のチームの先生 5 名に対して DummyMPS を用いた RCLL の説明を行った。その際、説明を聞いてどの程度面白さを感じたかを評価するアンケートを行った。結果を図 8 に示す。

全員が面白さを感じた(5 以上)と回答した。今回提案する DummyMPS は、Robotino を使った競技大会を簡単に実施するための物であり、説明会参加者の移行先の大会として期待できる。アンケート結果と図 1 の Robotino の競技者数より、参加者から移行先として面白さを感じてもらえたと考えられる。

6 まとめ

若年者ものづくり競技大会に出場しているチームから Robotino を用いた大会を開催して欲しいという要望を受けた。本研究ではそのような競技である RCLL

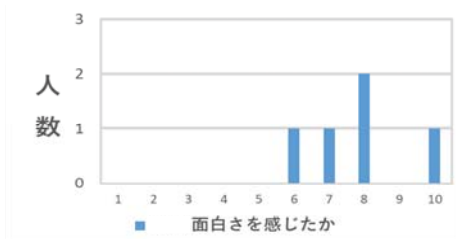


図 8: 提案する League の面白さの評価 (10 段階)

を実施するために, DummyMPS を作製した. これにより, 高価な機器を使わずとも, RCLL を実施できるようになった. そして, その競技大会に対する興味も高いことを確認できた.

参考文献

- [1] 中央職業能力開発協会 (JAVADA), “若年者ものづくり競技大会”, <http://www.javada.or.jp/jyakunen20/> (閲覧日, 2019 年 01 月 04 日)
- [2] Vincent Coelen, et al., “RoboCup Logistics League Rules and Regulations 2018”, <http://www.robocup-logistics.org/rules> (閲覧日, 2019 年 01 月 04 日)
- [3] 株式会社エム・システム技研, “計装豆知識: Modbus (モdbus)”, MS-TODAY, 1999 年 11 月号, p.14. http://www.m-system.co.jp/mstoday/plan/mame/b_network/9911/index.html (閲覧日, 2018 年 08 月 30 日)
- [4] 株式会社エム・システム技研, “Modbus プロトコル概説書”, リモート I/O 概説書、コンフィギュレータ取扱説明書, NM-5650 (改 10), pp.35 - 36, <http://www.m-system.co.jp/mssjapanese/kaisetsu/nmmodbus.pdf> (閲覧日, 2018 年 08 月 30 日)
- [5] Modbus-IDA, “MODBUS Application Protocol Specification”, Ver 1.1b, pp. 16 - 31, Dec, 2006, http://www.modbus.org/docs/Modbus_Application_Protocol_V1.1b.pdf (閲覧日, 2018 年 08 月 30 日)