TOPPERS 活用アイデア・アプリケーション開発コンテスト

部門	：	アプリケーション開発部門
作品のタイトル	：	TOPPERS/ASP と Wio Terminal を用いた HackSPi 向け無線コントローラ
作成者	：	田中康介（四国職業能力開発大学校）
共同作業者	：	
作品の対象者	：	ET ロボコン参加者
使用する開発成果物	：	TOPPERS/ASP、TOPPERS/ASP Arduino ライブラリ、SPIKE-RT

目的・狙い

ET ロボコンでは、ライントレース走行を行う際に、カラーセンサから取得する値（以後、RGB 値）を使用してラインの判別を行う必要がある。本作品は、コース上の手の届かない場所の RGB 値の取得を遠隔操作で行うこと、試走時の限られた時間の中で効率的に調整作業を行うことを目的として製作した。

本作品を用いることで、走行体(HackSPi)を RGB 値の取得位置まで遠隔操作で移動させ、取得した RGB 値を即座に競技フィールド外のチームメンバに送信し、速やかにライン判別の閾値調整ができる。

アイデア/アプリケーションの概要

本作品は，ET ロボコンの走行体(HackSPi)の RGB 値の取得を目的とした、遠隔操作を行うための専用コントローラである。Wio Terminal と HackSPi に搭載された Raspberry Pi 間を XBee による無線通信で接続している。無線コントローラのスティック操作で走行体の前後左右の移動を行い、取得地点まで移動できる。移動後、RGB 値の取得ボタンを押すことで RGB 値を 10 回取得し、平均値の算出を行う。取得した RGB 値と平均値は Wio Terminal の液晶画面と HackSPi に SSH 接続した PC の画面上に表示される。

１．開発の背景・概要

１．１ 開発の背景

１．１．１ 開発のきっかけ

開発のきっかけは、ET ロボコン 2023 に参加したことである。走行体はカラーセンサから取得した RGB 値によって、ライントレース走行を行う。しかし、走行環境の違い（照明やコースの土台など）によって、RGB 値が変動する可能性がある。

ET ロボコンでは、大会の本番走行の前にコースの「試走」を行うことができる。そこで、RGB 値の取得を行い、RGB 値の閾値調整を行うことで走行環境の違いに対処する。

図 1 に RGB 値の取得位置の例を示す。L コースは START/GOAL 地点における、白・黒・青ラインとラインの分岐地点における、白・黒・青ライン、さらに各カラーサークルの計 15 カ所である。左右反転した R コースも同様の位置を取得する必要があり、合計 30 カ所の取得位置となる。RGB 値を取得するにあたり、コースの周囲からでは手の届かない位置が存在する。このような問題に対処する為に、遠隔操作で効率良くコース上の RGB 値を取得がする仕組みが求められる。

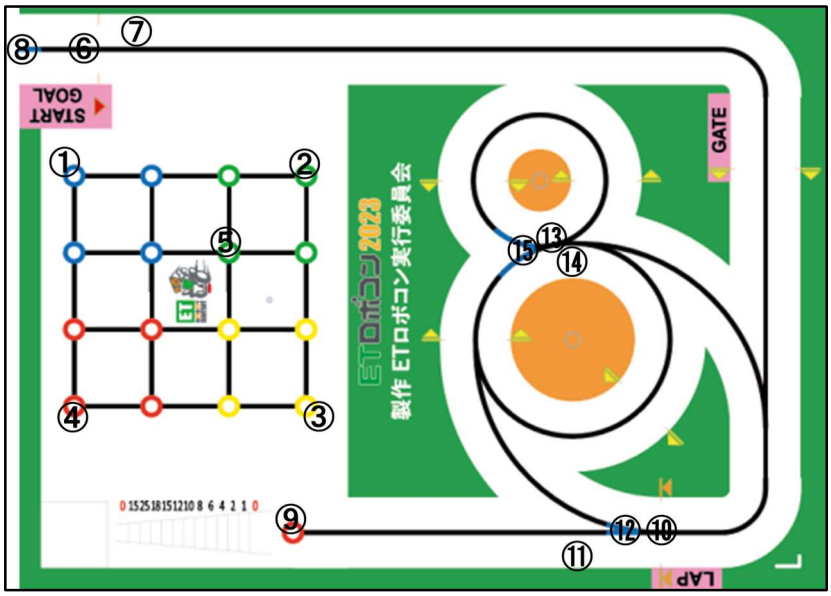

図 1　RGB 取得位置（L コース）

１．１．２ 試走について

試走について説明する。ET ロボコン 2023 競技規約の関連箇所を以下の通り要約した。

・競技フィールドに設置されたコースを使用して、走行体を調整することができる。

・調整治具を持ち込むことができる

・競技フィールド内へ立ち入ることができる人数は最大２名まで

尚、ET ロボコン 2023 での試走時間は地区大会では 25 分、チャンピオンシップ大会では前日 60 分、当日 15 分の時間が設けられた。

１．１．３ 従来の調整方法と見えた課題

図 2 に ET ロボコン 2023 までの調整方法を示す。表 1 に各役割と責務を示す。

ET ロボコン 2023 までは、走行体に HackEV を用いて出場した。PC と HackEV 間は、HackEV に搭載されている Bluetooth SPP の機能を用いて PC から ASCII 文字の操作コマンドを送信することにより遠隔操作を実現した。

図 2 左にもあるように、操作役が走行体から離れた場所におり、指示役と声による連携が必要となる為、取得位置の細かい指定が難しい。また、操作役が RGB 値を取得した後に調整作業を行う為、タイムロスに繋がっていることも課題として挙げられる。

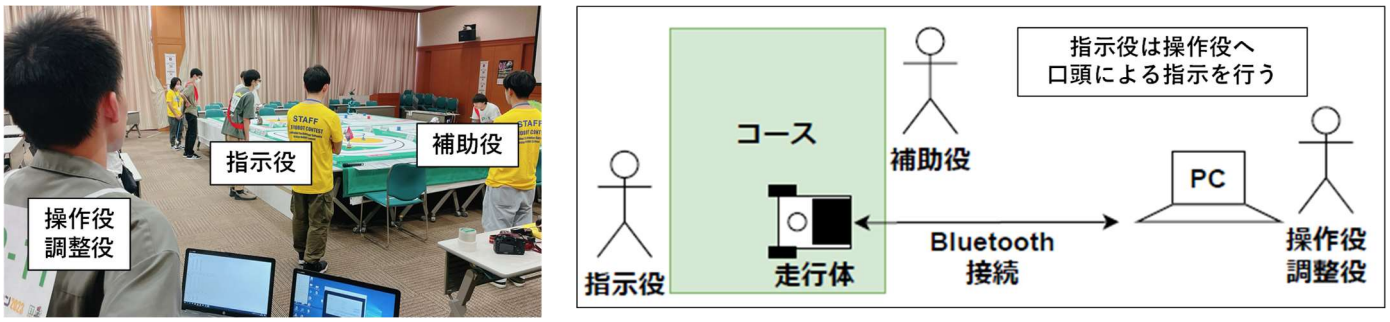

図 2　ET ロボコン 2023 までの調整方法

表 1　各役割と責務

役割名	責務
指示役	操作役に口頭で走行体の移動指示を行う
操作役	PCからラジコン操作を行う
調整役	プログラムの閾値の調整を行う
補助役	コースから走行体が落ちないよう補助を行う

１．１．４ 無線コントローラの開発

前項の課題を解決するために、「無線コントローラ」の開発を行った。

無線コントローラを用いることで図 3 のように、操作役と指示役の統合、操作役と調整役の分離を行うことができた。無線コントローラは、ET ロボコン 2024 大会を見据えて走行体「HackSPi」を前提とした設計を行っている。

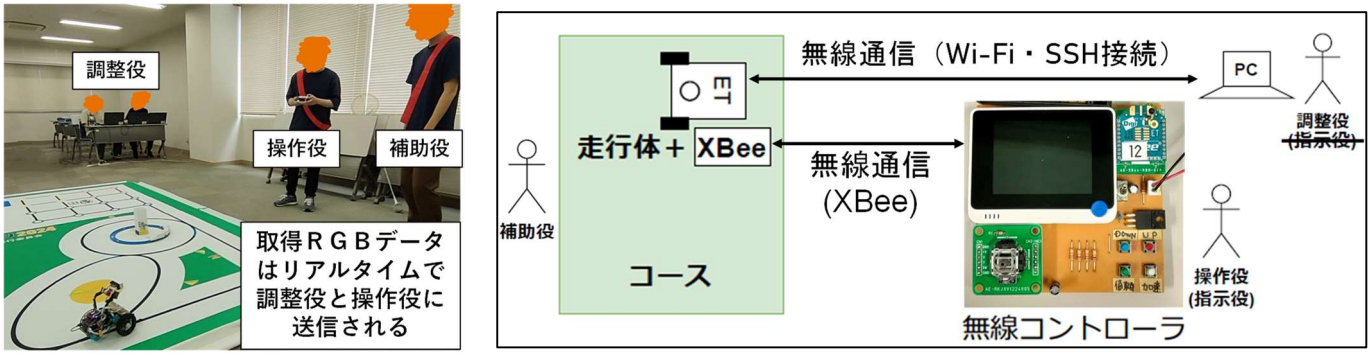

図 3　無線コントローラを用いた際の調整方法

１．２ 無線コントローラの機能

開発した無線コントローラを図 4 に示す。

・HackSPi をジョイスティックで遠隔操作する機能

・加速ボタンを押しながら、ジョイスティックを操作することで、高速走行に切り替わる。

・HackSPi 停止時に、RGB 値を無線コントローラの LCD に表示する機能

・RGB 値の取得ボタンの押下時に、RGB 値を取得し表示する機能（図 5）

・50msec 周期で 10 回取得し、平均を算出

・取得データと平均値を、無線コントローラの LCD と HackSPi に SSH 接続した PC の画面に表示する

・RGB 値の取得ボタンの押下時に、図 6 のように場所に応じた番号（以下、「場所番号」と呼ぶ。）を PC の画面に表示することで、どの場所のデータなのか見返すことができる。

※場所番号は、事前に決めておく必要があり、操作役は RGB 値の取得ボタンを押す前に、UP ボタンと DOWN ボタンで設定する。

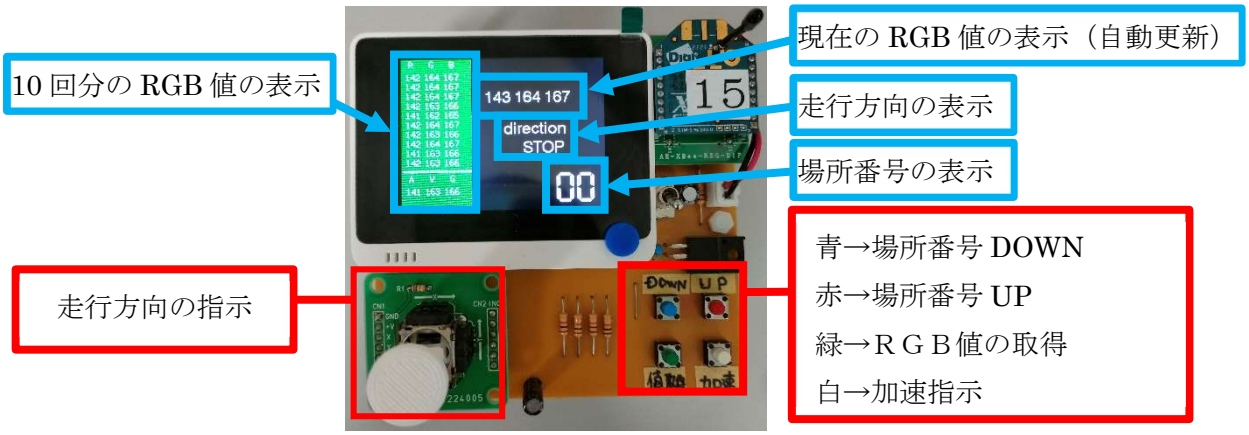

図 4　開発したコントローラ

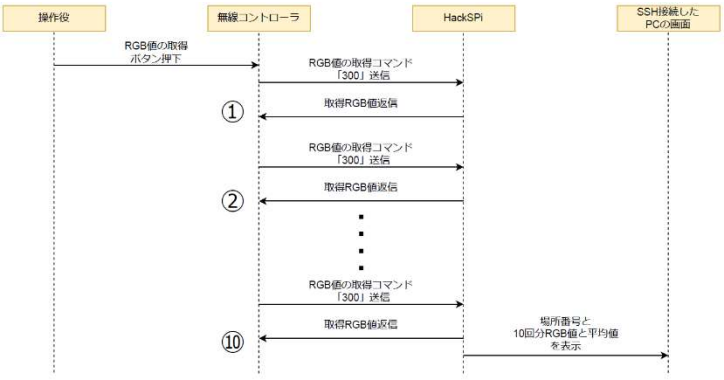

図 5　RGB 値の取得ボタン押下時のシーケンス

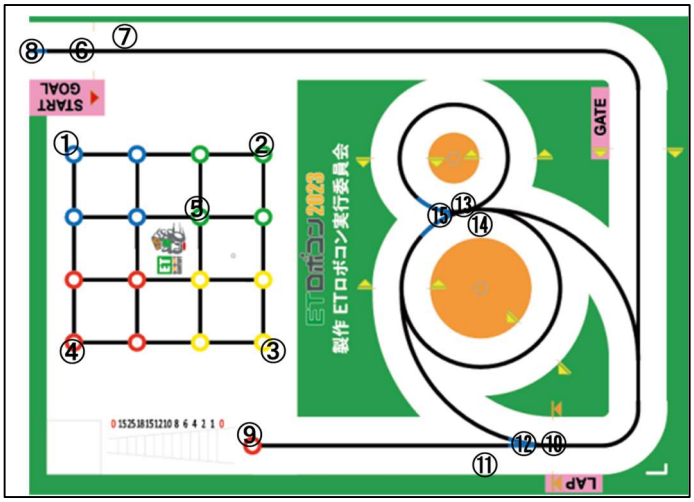

図 6　場所番号

２．システム構成

２．１ システム全体の構成

システム全体の構成図を図 7 に示す。ET ロボコン実行委員会より提示されている HackSPi 環境に XBee（無線モジュール）を追加し、無線コントローラと無線通信を行っている。XBee は、図 8 のように USB ケーブルで RaspberryPi と接続している。

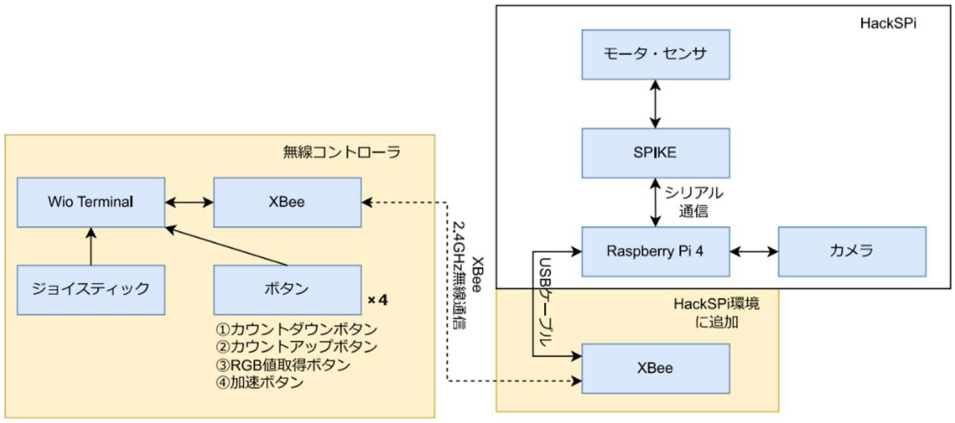

図 7　システム全体の構成図

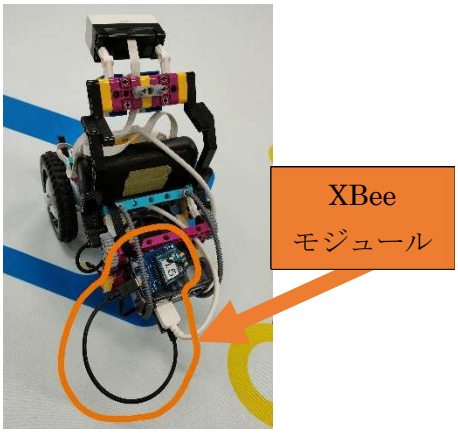

図 8　HackSPi と XBee

２．２ 無線コントローラの回路構成

Wio Terminal の本体裏に搭載された 40 ピンの GPIO より制御を行っている。（図 9、図 10）

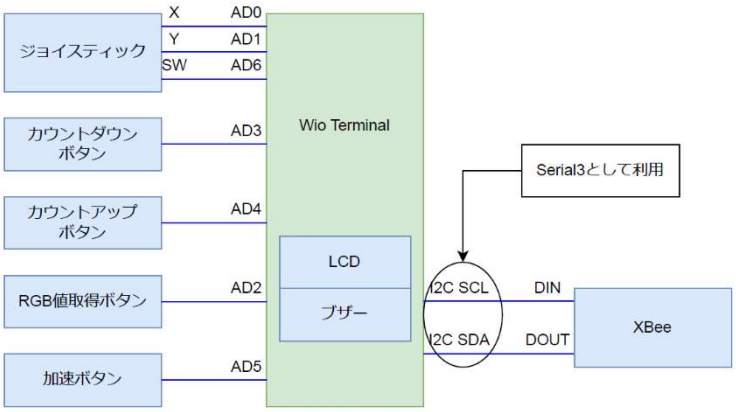

図 9　無線コントローラの回路構成

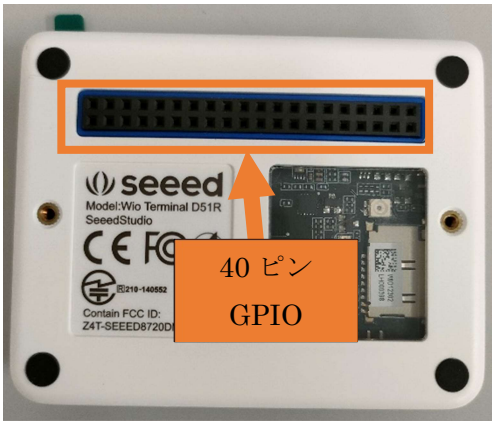

図 10　Wio Terminal 本体裏の GPIO

図 11 に無線コントローラの回路図を示す。無線コントローラの電源には、アルカリ単 3 電池 4 本（6V）を使用し、電源レギュレータ回路で 5V に変換し、WioTerminal に供給している。

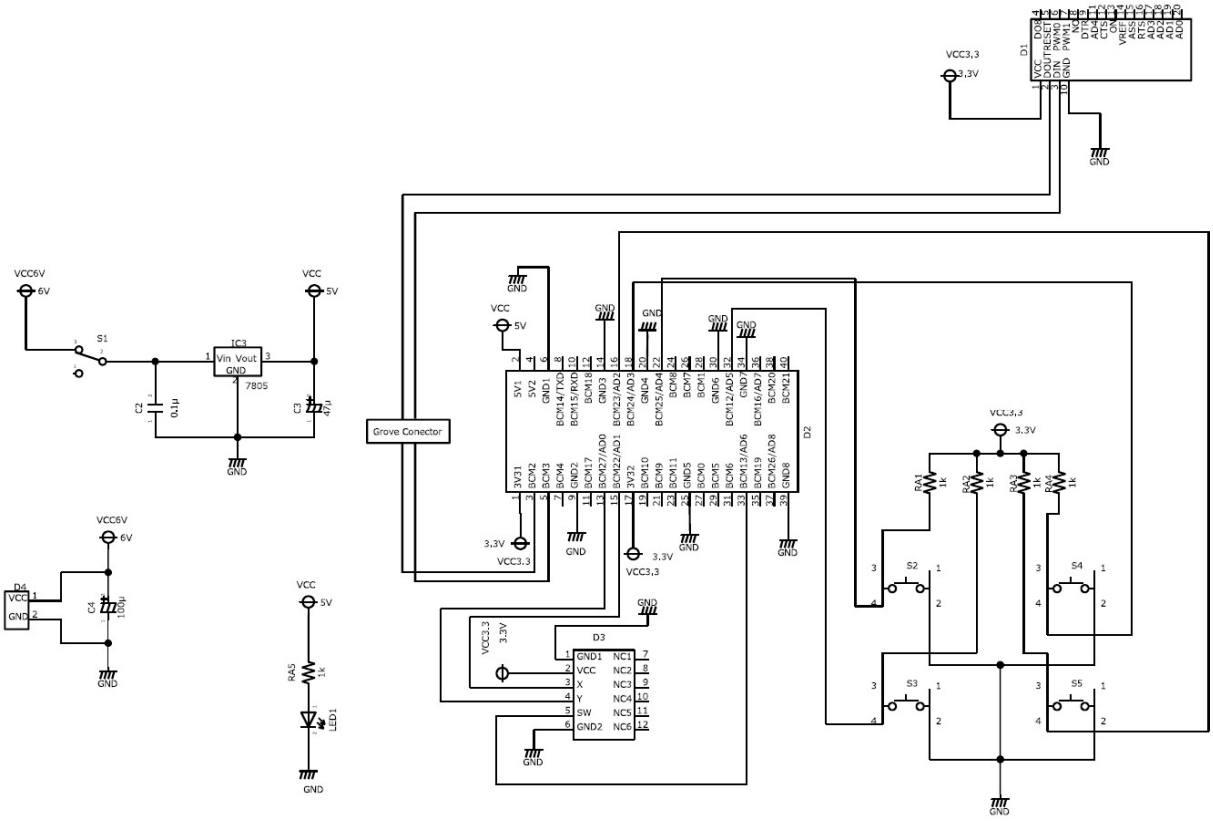

図 11　無線コントローラの回路図

表 2 に無線コントローラ作成に必要な部品の一覧を示す。

表 2　部品表

部品名	定数	数量	型番	用途
Wio Terminal		1	102991299	
ピンヘッダー 2×20 (40P)		1	PH-2x20SG/RH	Wio Terminal GPIOピン接続用
LED		1	OSG8HA3Z74A	電源確認用
トグルSW		1	2MS1-T1-B4-VS2-Q-E	電源操作用
三端子レギュレーター		1	TA4805S	電源電圧変換用
電池ボックス 単3×4本		1	SBH-341-1AS	電源用
XHコネクター ベース付ポスト トップ型 2P		1	B2B-XH-A(LF)(SN)	電池ボックスコネクタ用
XHコネクター ハウジング 2P		1	XHP-2	電池ボックスコネクタ用
XHコネクター ハウジング用コンタクト		1	SXH-001T-P0.6	電池ボックスコネクタ用
電解コンデンサ	100μF	1	35ZLH100MEFC6.3X11	電源回路用
電解コンデンサ	47μF	1	35PX47MEFC5X11	電源回路用
セラミックコンデンサ	0.1μF	1	RD15F104Z1HL2L	電源回路用
抵抗	1kΩ	5	CF25J1KB	
XBee ZB S2Cモジュール		2	XB24CZ7WIT-004	無線送受信用
XBee変換基板		1	AE-XBee-REG-DIP	XBee2.54mmピッチ変換用
Xbee USBアダプタ		1	SSCI-034951	Raspberry Pi接続側
タクトSW		4	DTS-63-N-V-BLK	
ジョイスティック		1	RKJXV1224005	
ジョイスティック変換基板		1	AE-RKJXV1224005	ジョイスティック2.54mmピッチ変換用
丸ピンIC用ソケット1×40		2	6604S-40	基板実装用(適当なピン数折って使用)

２．３ HackSPi と無線コントローラの通信仕様

表 3 に HackSPi と無線コントローラ間の通信仕様を示す。

無線コントローラから HackSPi へ ASCII コードで 3 文字のコマンドを送信し、HackSPi はコマンドに応じた処理を行った後、10 文字の文字を返す。

④RGB 値の取得コマンドはボタン押下時に 10 回送信され、RGB 値を 10 回受信後、平均値の算出を行う。

表 3　通信仕様

No.	コマンド	無線コントローラからHackSPi	HackSPiから無線コントローラ	備考
①	通信確認	100	1 000 000 000	―
②	RGB値の取得 （1回）	200	2 255 255 255 （255はRGB値）	無線コントローラに、 常時RGB値を表示させる用途
③	移動	0①② ①：加速=1　通常=0 ②：前=1　後=2　左=3　右=4	1 000 000 000	―
④	RGB値の取得 （10回＋平均）	3〇〇 （〇〇は場所番号）	3 255 255 255 （255はRGB値）	HackSPiにSSH接続した PCへ表示する用途

２．４ 無線コントローラのタスク構成

図 12 にタスクの呼び出し関係を示す。表 4 にタスクの一覧表を示す。タスクは各通信コマンドごとに用意している。

main_task は各ボタンとジョイスティックの状態を監視し、状態によってイベントフラグを立ち上げ、move_task、get_rgb_task、realtime_rgb_task のトリガーとなる。また、セマフォを用いてシリアル通信と LCD の排他処理を行っている。

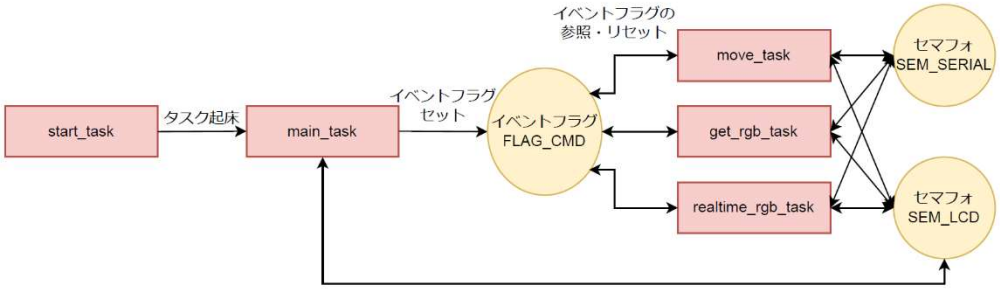

図 12　タスクの呼び出し関係図

表 4　タスクの一覧表

タスク名	優先順位	役割	排他制御（セマフォ）	実行周期[msec]	備考
start_task	10	通信確立待ち	―	100	①通信確認コマンド送信後、通信確立後にmain_taskを起こし終了
main_task	10	各タスクの起動	SEM_LCD	50	ボタン、ジョイスティックの監視。場所番号のカウントUP/DOWNもこのタスクで行っている
move_task	5	ラジコン操作用	SEM_LCD SEM_SERIAL	50	③移動コマンドの送信を行うタスク
get_rgb_task	5	ボタン押下時RGB値取得用	SEM_LCD SEM_SERIAL	100	④RGB値（10回）の取得コマンドの送信を行うタスク
realtime_rgb_task	15	停止時RGB値取得用	SEM_LCD SEM_SERIAL	100	②RGB値（1回）の取得コマンドの送信を行うタスク

３．導入方法

３．１ RasPike 開発環境の構築

RasPike 開発環境は、ET ロボコン実行委員会より公開されているドキュメントを参照

https://github.com/ETrobocon/RasPike/wiki/RasPike_install

（RasPike の開発環境セットアップ手順 URL）

３．２ WiringPi 追加インストール

Raspike 環境上で動作する遠隔操作プログラムには、XBee とのシリアル通信に WringPi を用いている。その為、ターミナル上でコマンド入力による、WiringPi のインストールを行う。

①ホームディレクトリに移動

```
$ cd ~
```

②GitHub からインストールコードをダウンロード

```
$ git clone https://github.com/WiringPi/WiringPi.git
```

Wiring Pi のソースコードがWringPi ディレクトリとしてクローンされる。

③ディレクトリ内に移動してビルドを行う。

```
$ cd WiringPi
$ ./build
```

図 13 の通りの結果が得られればインストールが完了

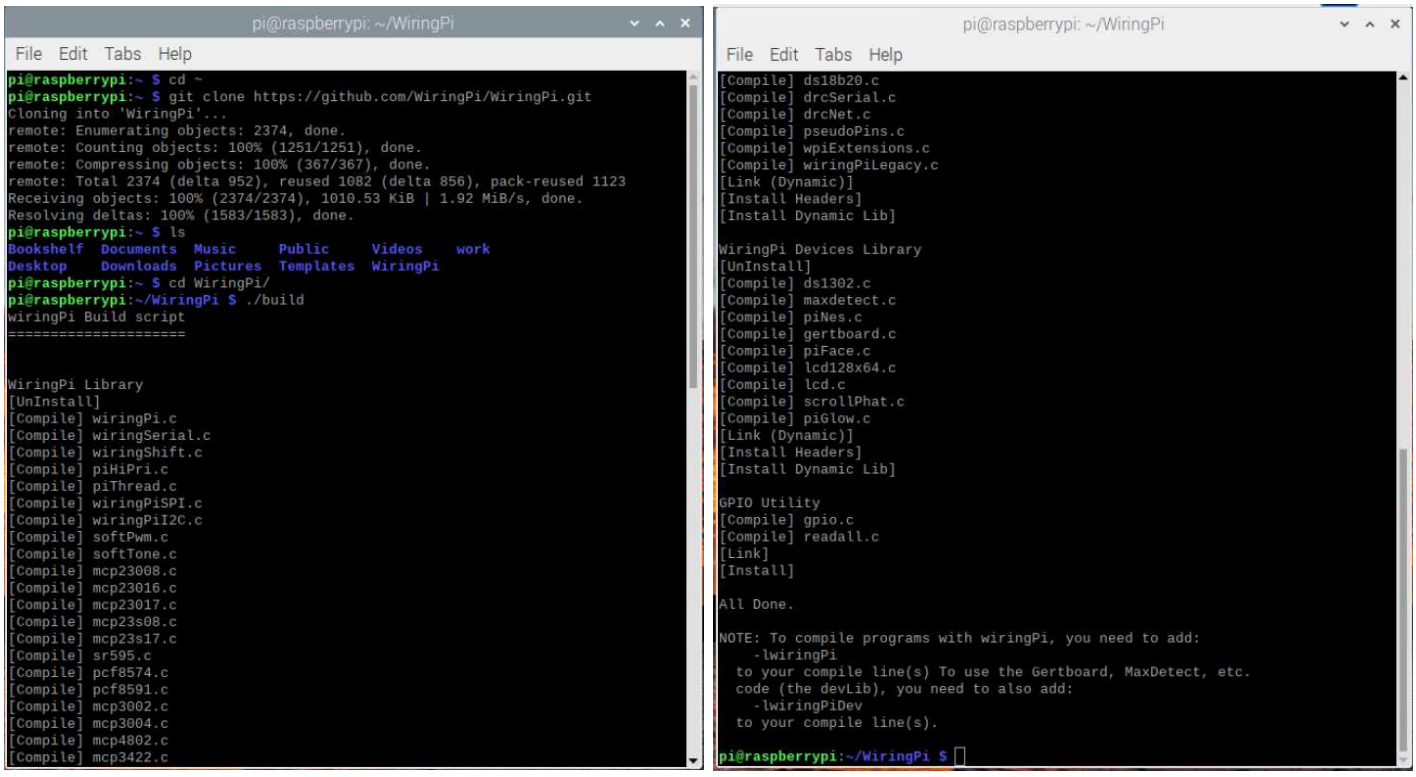

図 13　WringPi のインストール画面

３．３ Wio Terminal 開発環境の構築

①Arduino IDE インストール

バージョンは Arduino IDE 2.2.1 を用いて開発を行っています。

https://www.arduino.cc/en/software （公式サイト）

②ボードマネージャーを追加（図 14）

https://files.seeedstudio.com/arduino/package_seeeduino_boards_index.json

上記追加ボードマネージャーの URL を基本設定のウィンドウに張り付ける。

③ボードライブラリをインストール（図 15）

ボードマネージャで「Wio Terminal」と検索し、「Seeed SAMD Boards」をインストール

④既定のボードを Wio Terminal にする（図 16）

ツール→ボードから Wio Terminal を選択し、規定のボードを Wio Terminal に設定

⑤液晶表示用ライブラリを追加インストール

https://github.com/Seeed-Studio/Seeed_Arduino_LCD

【seeed-Arduino-LCD ライブラリ】を ZIP 形式でダウンロード

「スケッチ」→「ライブラリのインクルード」→「.ZIP 形式のライブラリをインストール」

→ダウンロードした ZIP ファイルを選択

⑥Arduino_TOPPERS_ASP を追加インストール

https://github.com/exshonda/Arduino_TOPPERS_ASP/releases/

【Arduino_TOPPERS_ASP】を ZIP 形式でダウンロード

「スケッチ」→「ライブラリのインクルード」→「.ZIP 形式のライブラリをインストール」

→ダウンロードした ZIP ファイルを選択

図 14　ボードマネージャーを追加

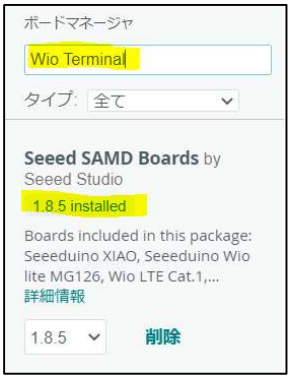

図 15　ボードライブラリをインストール

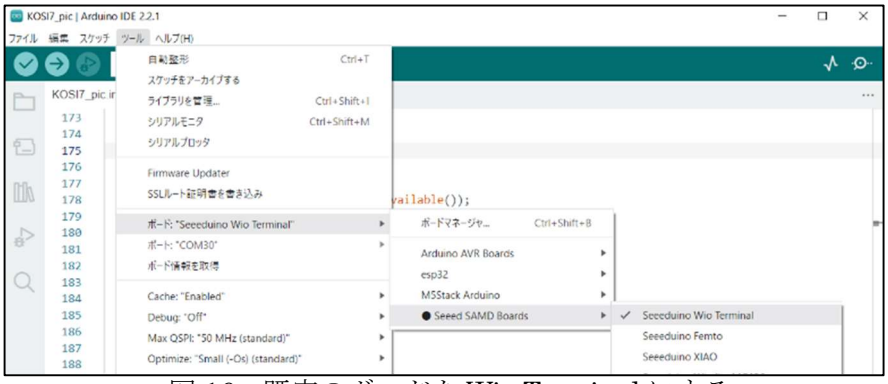

図 16　既定のボードを Wio Terminal にする

３．４ プログラムのダウンロード

HackSPi 側プログラムと無線コントローラ側プログラムは下記サイトに公開している。

・HackSPi 側プログラム名：con_pi

移動速度は適宜変更のこと（define 定義済み）

・無線コントローラ側プログラム名：con_Wio

ジョイスティックの移動方向の判定の閾値は適宜変更のこと

https://sites.google.com/view/toppers-hackspi-con/

（TOPPERS/ASP と Wio Terminal を用いた HackSPi 向け無線コントローラサイト URL）

３．５ プログラムの書き込み・実行

HackSPi 側プログラム書き込み方法

書き込む前に「con_pi」フォルダを「work/RasPike/sdk/workspace」内にコピーしておくこと（ZIP ファイルであれば解凍しておく）

以下ターミナルに入力

①ホームディレクトリに移動

```
$ cd ~
```

②Workspace に移動

```
$ cd work/RasPike/sdk/workspace/
```

③プログラムをコンパイル（書き込み）

```
$ make img=con_pi
```

④プログラムを実行

```
$ make start
```

無線コントローラ側プログラム書き込み方法

① Wio Terminal の Type-C 端子と PC を USB ケーブルで接続

② ダウンロードした「con_Wio」内の ino ファイルを Arduino IDE で開く

③ 図 17 のように Wio Terminal の接続されている COM ポートを選択し、

④ 「→」マークの書き込みボタンを押す

⑤ Wio Terminal は書き込み後に即時実行されます。

図 18 のような Raspberry Pi との通信確認待ち画面に遷移すれば書き込みと実行完了。

図 17　無線コントローラ書き込み

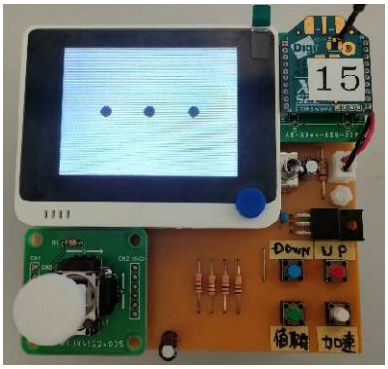

図 18　Raspberry Pi との通信確認待ち画面

4．公開しているデータについて

データ配布用サイト

https://sites.google.com/view/toppers-hackspi-con/

（TOPPERS/ASP と Wio Terminal を用いた HackSPi 向け無線コントローラサイト URL）

・ラズパイのプログラム　（con_pi）

・無線コントローラのプログラム（con_Wio）

・無線コントローラの回路仕様（システム構成図、回路ブロック図、回路図）

5．実行の様子

実際に無線コントローラを用いて走行デモを行う動画です。

https://youtu.be/QAntg3IIea0

6．今後の展望

ET ロボコン 2024 のプライマリークラスでは、新たに画像処理を用いる課題が追加された為、遠隔操作でカメラ画像の取得を行う機能の追加実装などが考えられる。

また、無線コントローラのプログラムは、静的な構造面でまだまだ改善点がある。主に main_task 周辺のコードの改良に取り組みたい。

7．謝辞

本作品の作成にあたり、多くの方々にご助力いただきました。

ET ロボコン 2024 実行委員会の方々には、規約の確認など丁寧に対応いただきました。チーム PCSEIT2 のメンバーには、本作品を実際に使用いただき、多くのフィードバックをいただきました。四国職業能力開発大学校の指導教員である及川達裕先生には、終始適切なご指導を賜りました。以上のご指導、ご協力をいただいた皆様に感謝の意を表します。

8．参考文献

［1］ET ロボコン 2023 競技規約

https://docs.etrobo.jp/rules/2023/ETRC2023_rules_primary_advanced_1.2.0.pdf