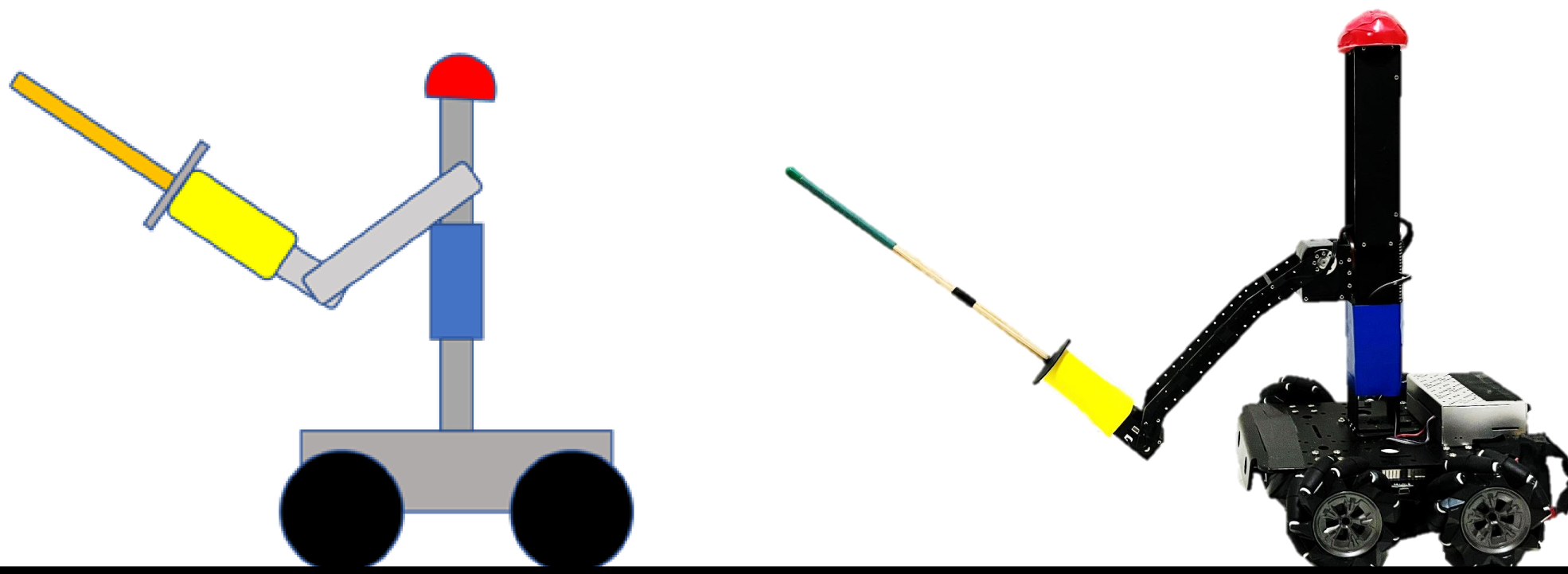


競技規則に載っていきそうなシンプルなロボット

デ ィ ム シ フ テ ィ

「Dimsifty」の作り方



●導入

1. 発表者紹介
2. 自律型ロボット製作のきっかけ

●展開

3. ハードウェア①【ロボットの機械設計】
4. ハードウェア②【ロボットの電子回路】
5. ソフトウェア①【コンピュータの環境構築】
6. ソフトウェア②【打突アルゴリズムの構築】
7. 実際の試合【第1回新ROBO-剣】

●まとめ

8. 新しいことに挑戦するには
9. 質疑応答



公開資料の更新や当日の追加資料があればこちらで公開する予定です。
また、資料につきましてご質問がありましたらご記入ください。

<https://forms.gle/j4Ba73vbVnNtrmGGA>

1. 発表者紹介

2023.07.15
第21回 ROBO-ONE Conference(2)
＜新ROBO-剣＞ Dimsifty / セキ

【高校生のとき】

神戸市立科学技術高等学校 機械工作部ROBO

→ここで二足歩行ロボット製作に目覚める



【大学生のとき】

大阪電気通信大学 自由工房HRP

→ここでROBO-剣のロボット製作に目覚める



【社会人になってから】

いろいろあって関東で就職

→現在でも二足歩行ロボット協会の展開する大会に参加

1. 発表者紹介

2023.07.15
第21回 ROBO-ONE Conference(2)
＜新ROBO-剣＞ Dimsifty / セキ

ロボット活動の内容は
99%がROBO-ONEだった。



今回の資料は

ROBO-ONEをやってる人で
新ROBO-剣に出ることを
考えている人向けの資料
になります。

ローテクで実現する画像処理入門のためのロボット

Information



SIN ROBO-KEN Ranking

100 pt **3 位**

Robot ID	1159
Robot name	Dimsifty
Robot kana	ディムシフティ
Robot eng	Dimsifty
Team name	セキ
Team kana	セキ
Team eng	Seki
Weight	1.97 kg
Height	26.7 cm
Country	Japan

Comment

ローテクで実現する画像処理入門のためのロボット

●導入

1. 発表者紹介

2. 自律型ロボット製作のきっかけ

●展開

3. ハードウェア①【ロボットの構成要素】

4. ハードウェア②【ロボットの基礎制御】

5. ソフトウェア①【コンピュータの環境構築】

6. ソフトウェア②【打突アルゴリズムの構築】

7. 実際の試合【第1回新ROBO-剣】

●まとめ

8. 新しいことに挑戦するには

9. 質疑応答

2. 自律型ロボット製作のきっかけ

2023.07.15
第21回 ROBO-ONE Conference(2)
<新ROBO-剣> Dimsifty / セキ

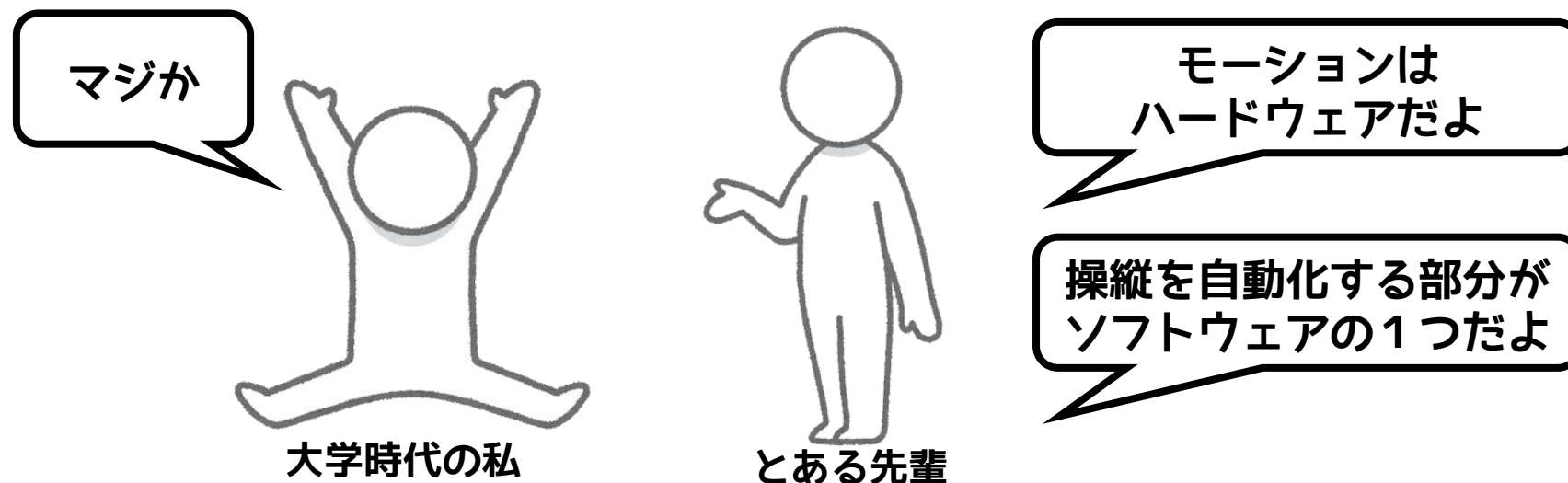
① 「ロボットを作ってます」って自信を持って言いたい

自分のROBO-ONE用ロボットはハードウェアを楽しむ割合が高い。

ソフトウェアは手動操縦やメーカの完成された製品でカバーしている。

→ソフトやらずに「ロボット作ってます」はなんか違う気がする。

→少しでもいいからソフトウェアの世界を経験したい。



2. 自律型ロボット製作のきっかけ

②そもそも自分は**ロボット操縦の適性が高くない**
高校在学中から現在に至るまで何回かあったこと

**自分の製作したロボットを
他人が操縦した時に結果が出る。**

→ロボットはそれなりの質で作れているのに、自分の操縦が悪かった
ことでロボットの性能を引き出せていないことが多かった。

→独り身になった今、**勝手に動いてくれるロボットで勝ちたい**と思った。

●導入

1. 発表者紹介
2. 自律型ロボット製作のきっかけ

●展開

3. ハードウェア①【ロボットの構成要素】
4. ハードウェア②【ロボットの基礎制御】
5. ソフトウェア①【コンピュータの環境構築】
6. ソフトウェア②【打突アルゴリズムの構築】
7. 実際の試合【第1回新ROBO-剣】

●まとめ

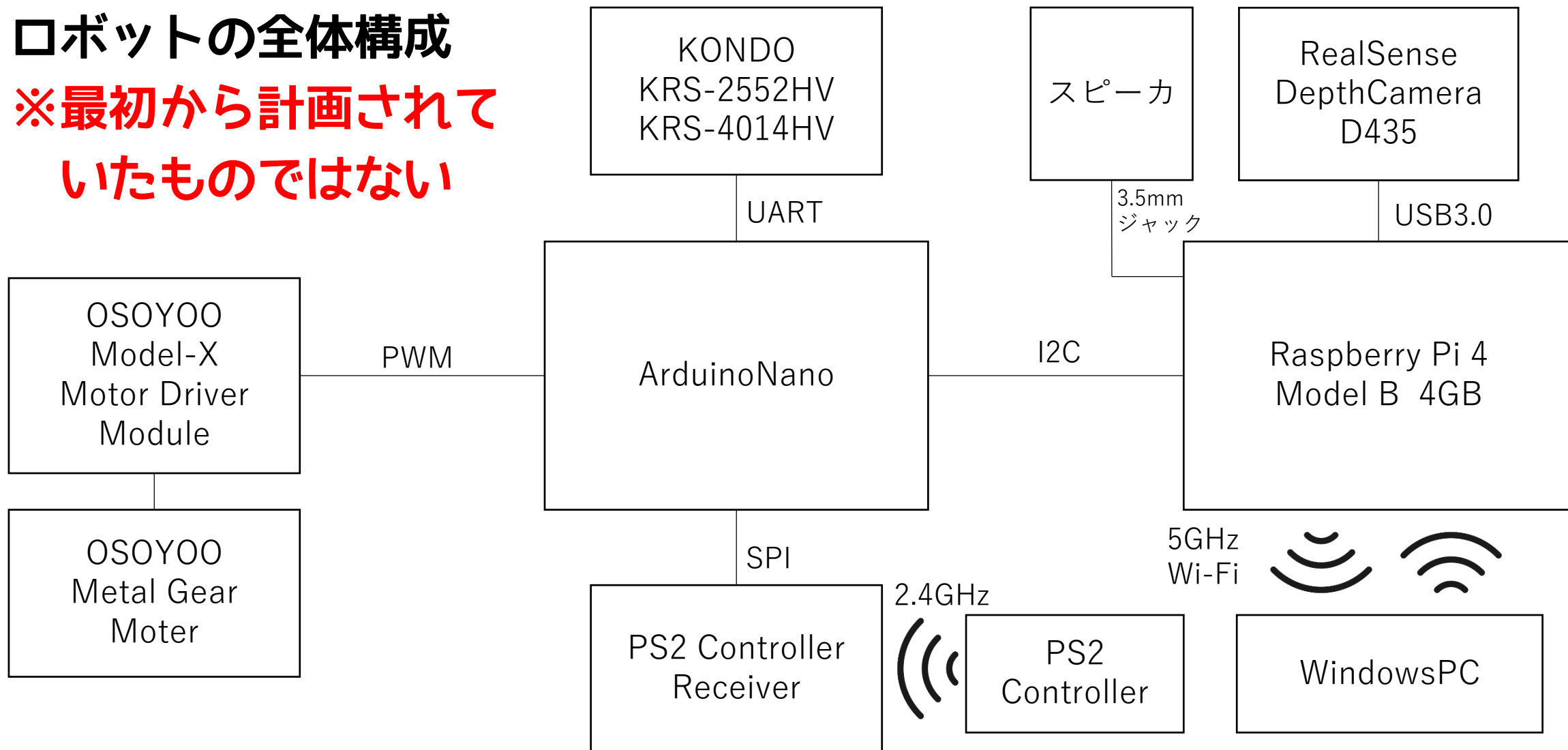
8. 新しいことに挑戦するには
9. 質疑応答

3. ハードウェア① 【ロボットの構成要素】

2023.07.15
第21回 ROBO-ONE Conference(2)
<新ROBO-剣> Dimsifty / セキ

ロボットの全体構成

※最初から計画されて
いたものではない



ロボット製作における現実

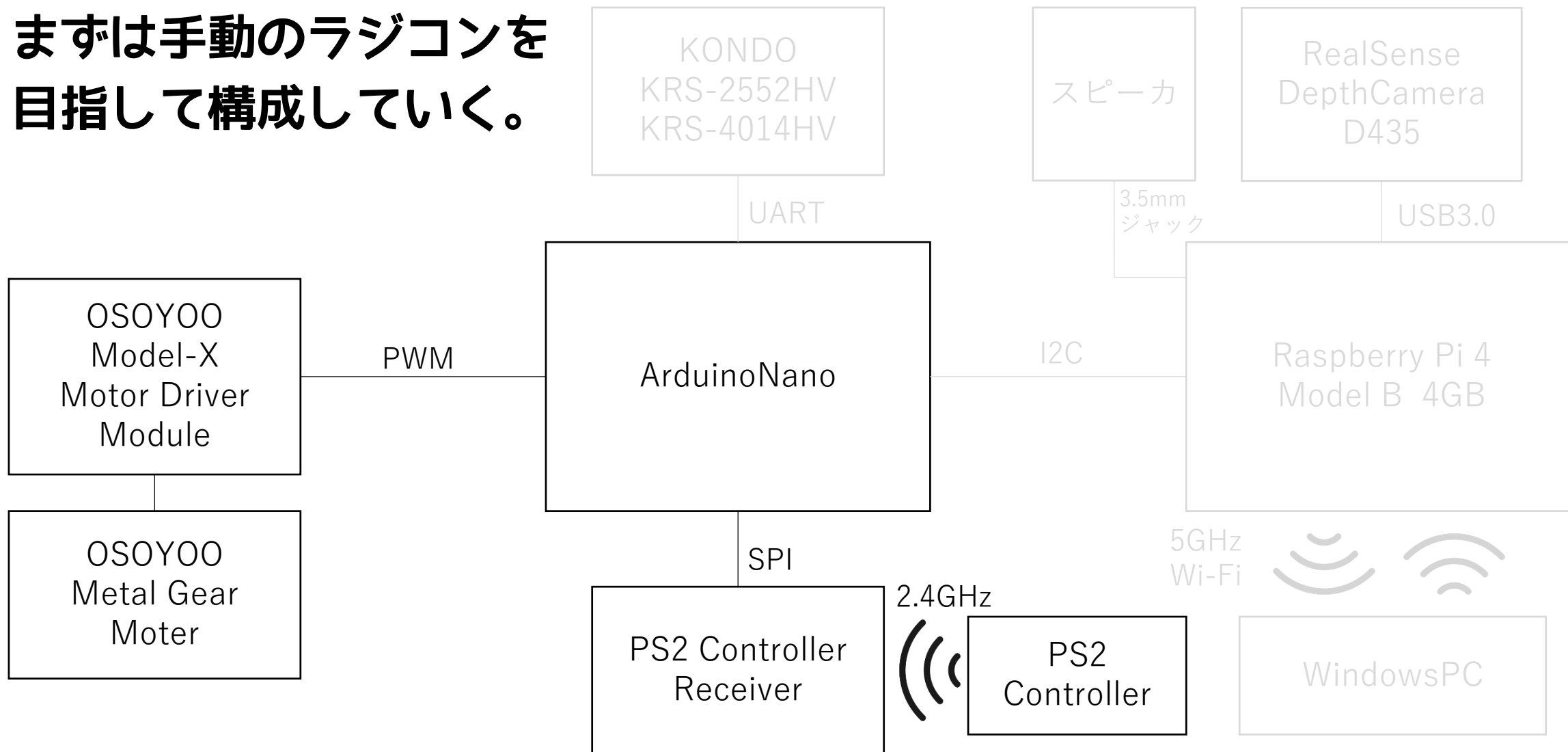
**Dimsiftyは最初から全体の構成が
決まっているロボットではなかった。**

- 技術的に自分が作りたいものをどこまで実現できるかわからなかった。
- 競技規則を見ながら、自分が作れそうな部分から作っていき
どんどん機能を足していくことで現状の構成に。

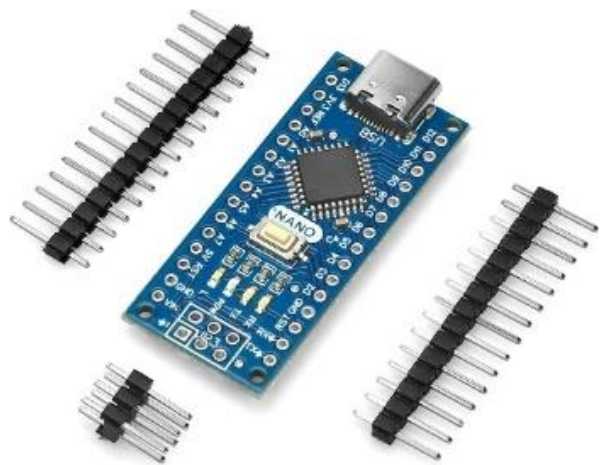
3. ハードウェア① 【ロボットの構成要素】

2023.07.15
第21回 ROBO-ONE Conference(2)
<新ROBO-剣> Dimsifty / セキ

まずは手動のラジコンを
目指して構成していく。



使用したマイコンボード ArduinoNano(ATmega328P)



選定理由

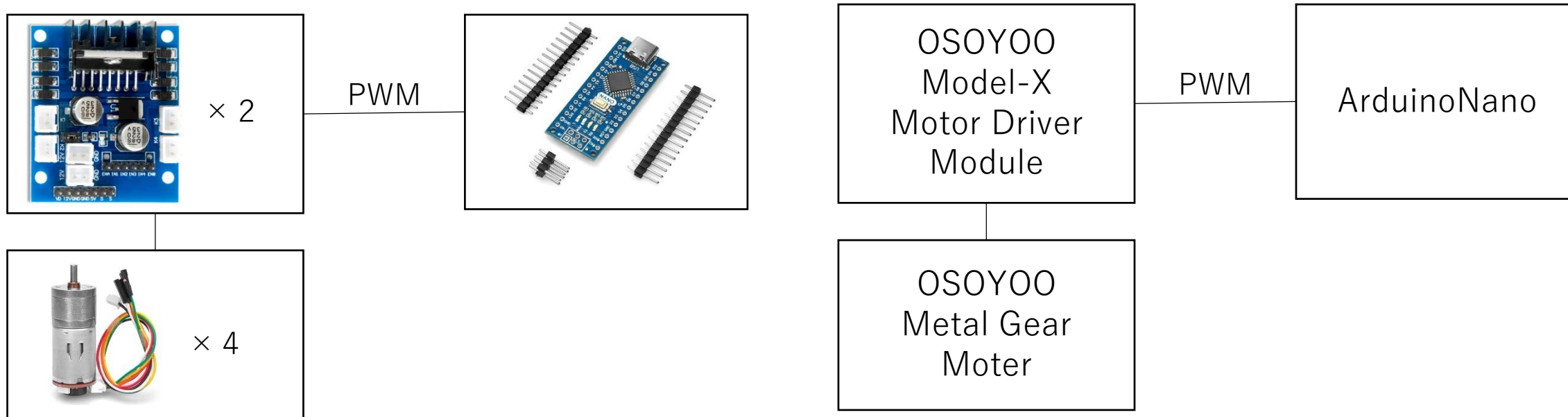
- ・ インターネット上に使用事例が大量に載っている
- ・ OSOY00のロボットカーがArduinoで制御されている

3. ハードウェア① 【ロボットの構成要素】

2023.07.15
第21回 ROBO-ONE Conference(2)
<新ROBO-剣> Dimsifty / セキ

ArduinoNanoにモータドライバを接続。

OSOY00より公開されているサンプルスケッチを用いると簡単に動作させることができる。モータについてるセンサは使用していない。



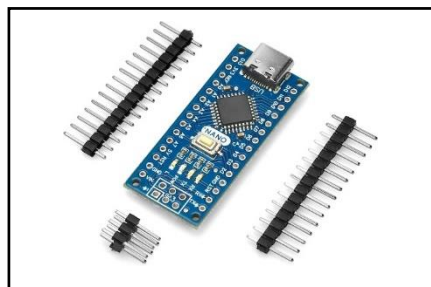
3. ハードウェア① 【ロボットの構成要素】

2023.07.15
第21回 ROBO-ONE Conference(2)
<新ROBO-剣> Dimsifty / セキ

ArduinoNanoにPS2無線コントローラを接続。

難しく見えるのはSPIだが、実際は優秀なライブラリがあるため

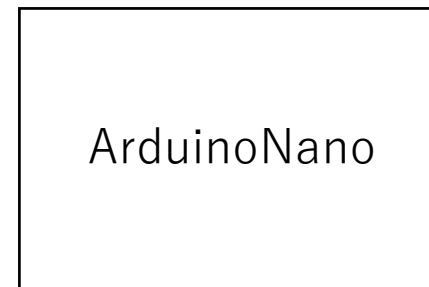
調べて出てきた記事通りに配線してサンプルスケッチを書き込めばOK。



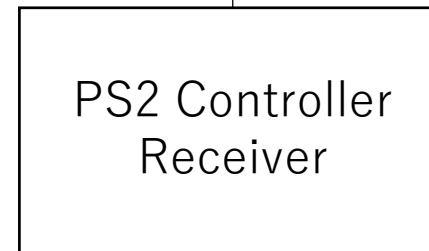
SPI



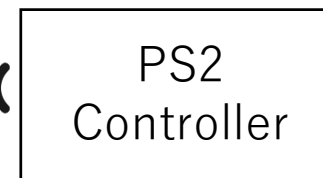
2.4GHz



SPI



2.4GHz



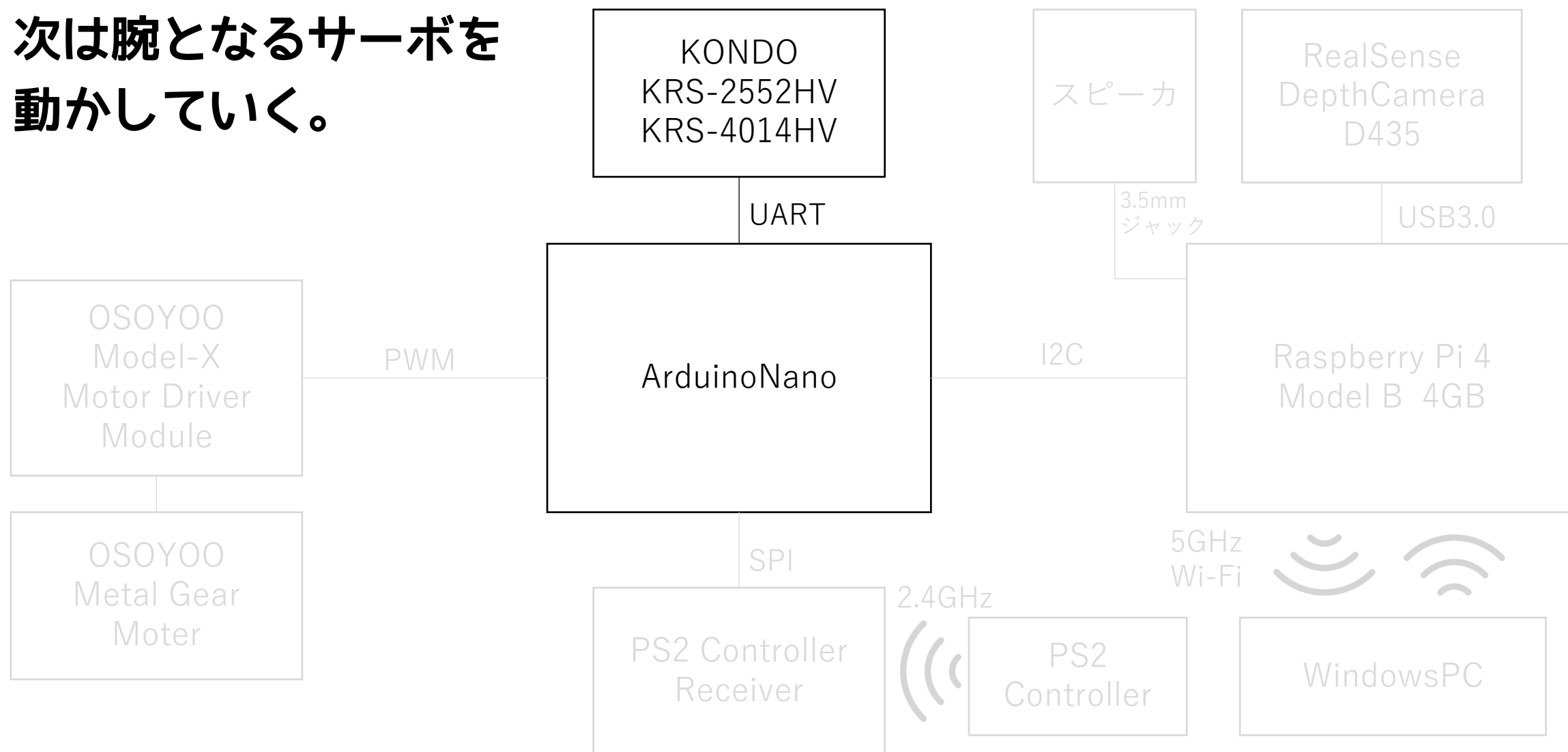
3. ハードウェア①【ロボットの構成要素】

2023.07.15

第21回 ROBO-ONE Conference(2)

<新ROBO-剣> Dimsifty / セキ

次は腕となるサーボを動かしていく。



3. ハードウェア① 【ロボットの構成要素】

2023.07.15
第21回 ROBO-ONE Conference(2)
<新ROBO-剣> Dimsifty / セキ

使用したサーボモータ

KRS-2552RHV / KRS-4014HV



選定理由

- ・ 使い慣れているから。

3. ハードウェア① 【ロボットの構成要素】

2023.07.15
第21回 ROBO-ONE Conference(2)
<新ROBO-剣> Dimsifty / セキ

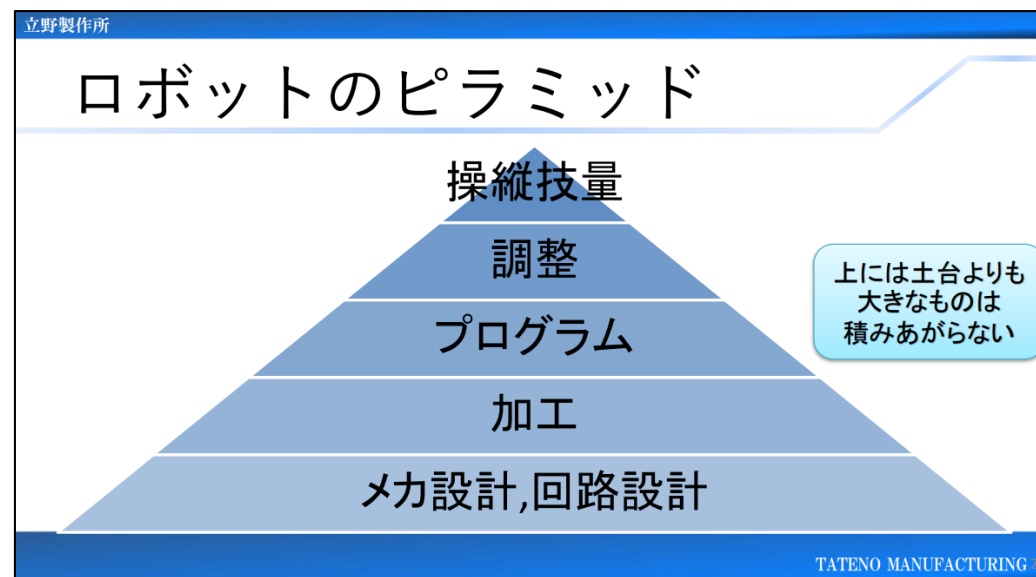
腕は過去のROBO-ONE用ロボットの製作経験を取り入れつつ製作。

→ 基礎は過去のカンファレンスのスライドや自作ガイドも参照

コンセプト

- ▶ 剛性がしっかりとっていて
試合中に絶対に壊れない機体
- ▶ 現存するすべての機体に対して
有効な攻撃を持つ機体

引用元：プニの製作秘話 / ぷに（中井 裕斗）
<https://biped-robot.or.jp/upload/dbmaterials/57221ecda385118c7a5d6eb45e808448cd3original.pdf>



引用元：それなりに動く二足の作り方 / 立野伸英
<https://biped-robot.or.jp/upload/dbmaterials/89285296140b4c4878034e451f8a57f9485original.pdf>

3. ハードウェア① 【ロボットの構成要素】

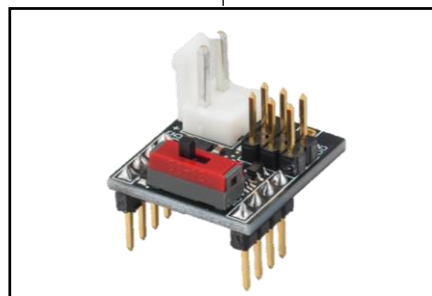
2023.07.15
第21回 ROBO-ONE Conference(2)
<新ROBO-剣> Dimsifty / セキ

確実かつ簡単に動かすならICS変換基板を使用する。

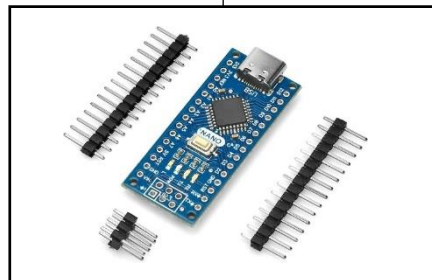
公式から配線図・ライブラリ・サンプルスケッチが公開されている。



KONDO
KRS-2552HV
KRS-4014HV



KONDO
ICS変換基板



ArduinoNano

ICS変換基板の使用方法 (1) 準備編

Posted on 2017.08.02 in

ICS変換基板は、マイコンボードなどに搭載されているUART端子、またはI/O端子のソフトウェアシリアルを使用してサーボモータなどICS機器と通信するための変換基板です。

従来は、ICS機器とマイコンボードを中継する回路を作成する必要がありましたが、この基板を使用することで簡単に制御することができるようになります。

接続参考図

下図を参考にお使いのマイコンボードと接続します。
必要に応じて、付属の4ピンヘッダを半田付けしてご使用ください。

マイコンボード



3. ハードウェア①【ロボットの構成要素】

2023.07.15

第21回 ROBO-ONE Conference(2)

<新ROBO-剣> Dimsifty / セキ

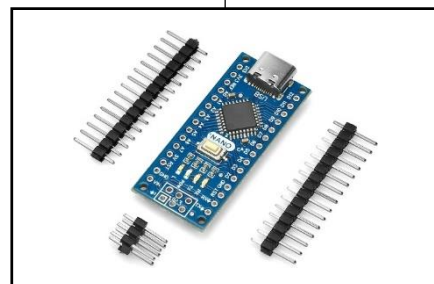
今回はArduinoNanoにサーボを
直で接続。サーボを動かすだけなら
単純な回路でも動く。



UART

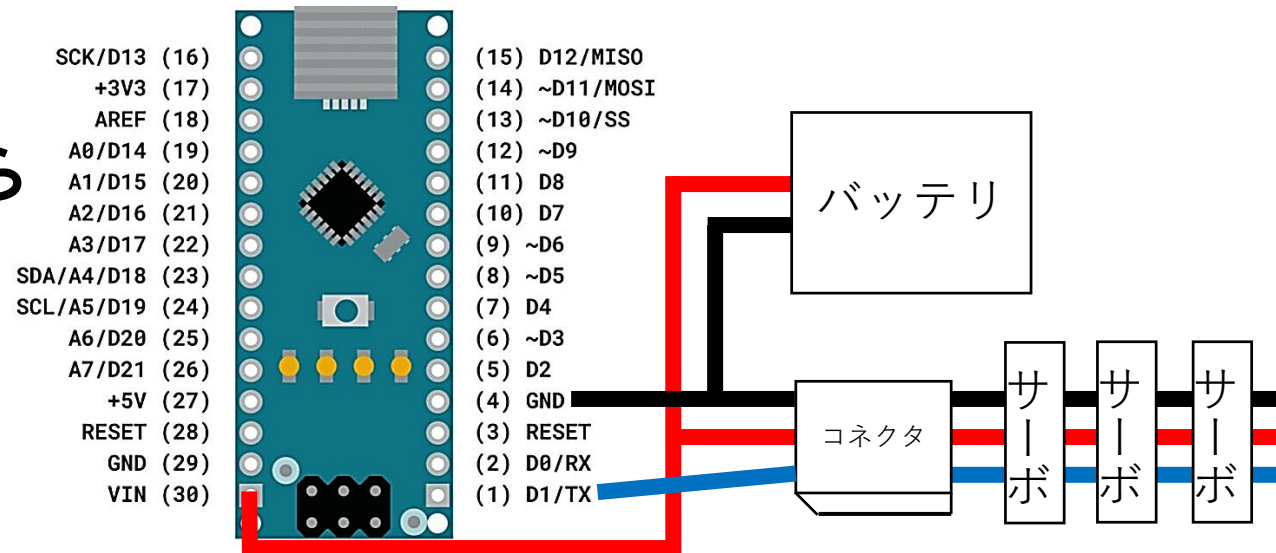
KONDO
KRS-2552HV
KRS-4014HV

UART

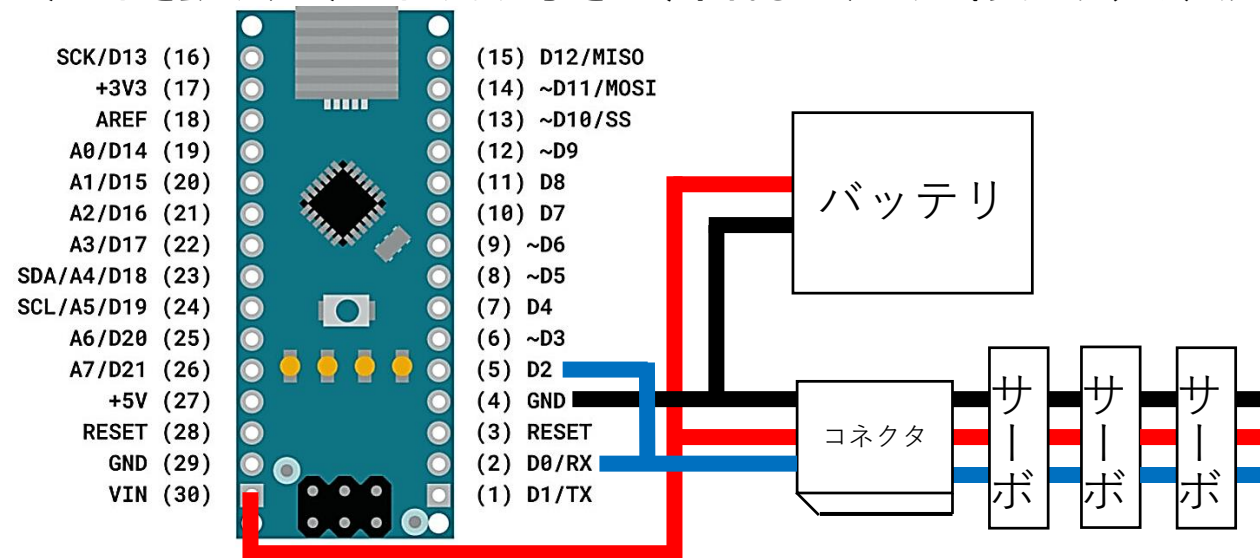


ArduinoNano

サーボを動かすだけの一番単純なパターン



サーボを動かす/サーボから返事を受け取るパターン（多少のリスクあり）



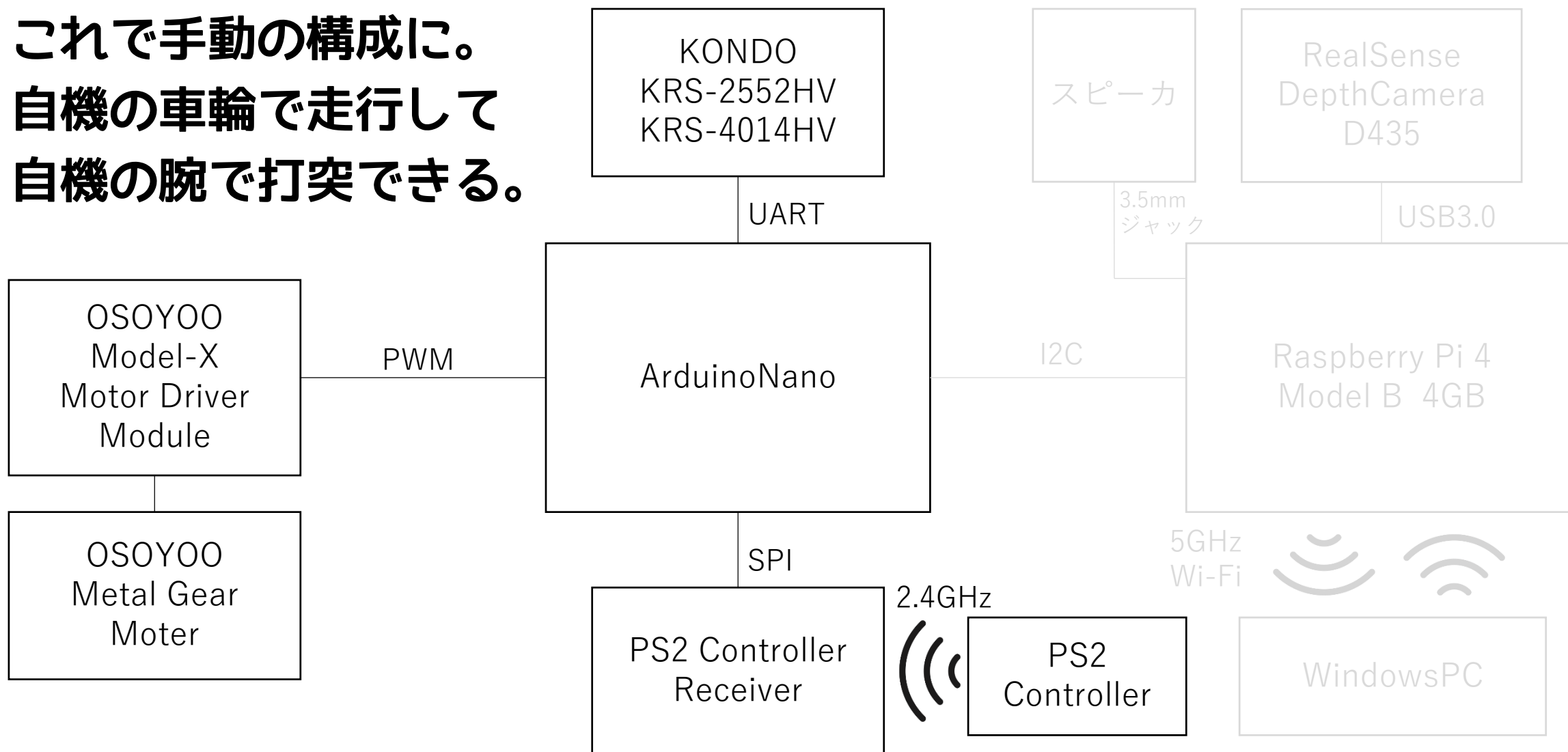
3. ハードウェア①【ロボットの構成要素】

2023.07.15

第21回 ROBO-ONE Conference(2)

<新ROBO-剣> Dimsifty / セキ

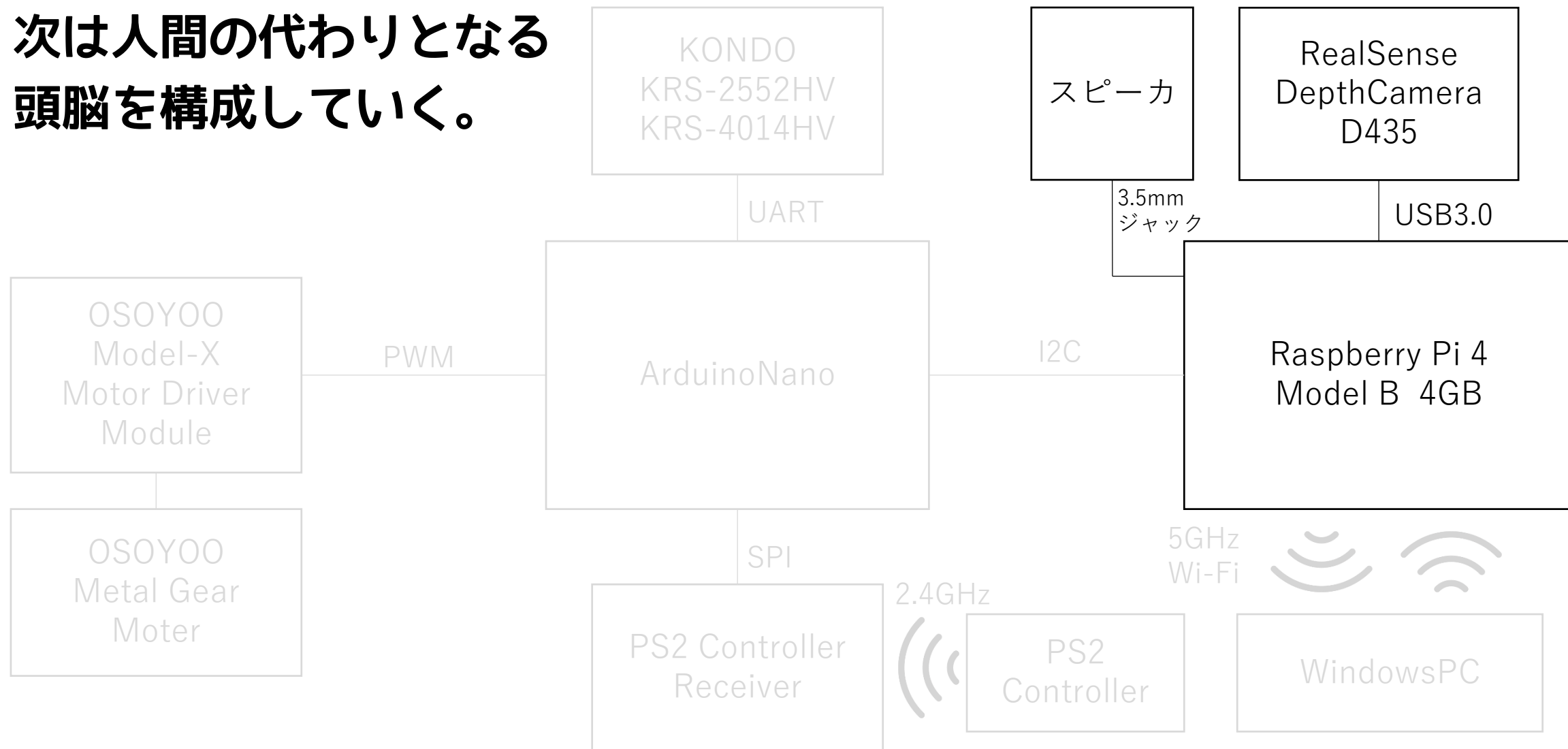
これで手動の構成に。
自機の車輪で走行して
自機の腕で打突できる。



3. ハードウェア① 【ロボットの構成要素】

2023.07.15
第21回 ROBO-ONE Conference(2)
<新ROBO-剣> Dimsifty / セキ

次は人間の代わりとなる
頭脳を構成していく。



3. ハードウェア① 【ロボットの構成要素】

2023.07.15
第21回 ROBO-ONE Conference(2)
<新ROBO-剣> Dimsifty / セキ

使用したシングルボードコンピュータ
RaspberryPi4 Model B(4GB)



選定理由

- ・ インターネット上に使用事例が大量に載っている

使用したカメラ

intel RealSense™ Depth Camera D435



選定理由

- ・ インターネット上に使用事例が大量に載っている
- ・ 深度カメラとRGBカメラを組み合わせた処理が簡単

3. ハードウェア① 【ロボットの構成要素】

2023.07.15
第21回 ROBO-ONE Conference(2)
<新ROBO-剣> Dimsifty / セキ

RaspberryPi4にD435をUSB3.0で接続。

打突時の発声のためにスピーカをRaspberryPi4のイヤホンジャックに接続。

→これらが人間の**脳・目・口**と同じ役割を果たす。



3.5mm
ジャック



USB3.0



スピーカ

RealSense
DepthCamera
D435

3.5mm
ジャック

USB3.0

Raspberry Pi 4
Model B 4GB

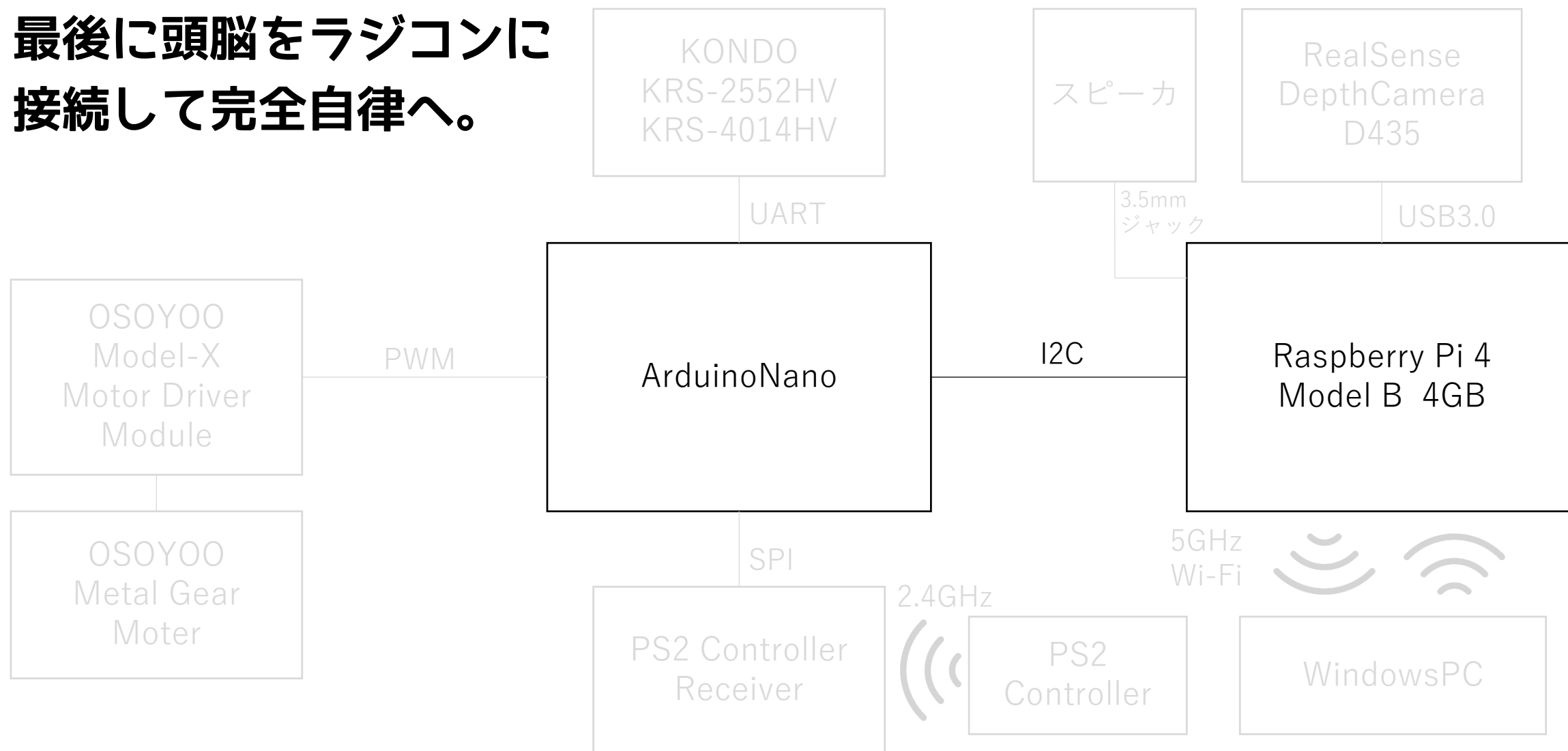
3. ハードウェア①【ロボットの構成要素】

2023.07.15

第21回 ROBO-ONE Conference(2)

<新ROBO-剣> Dimsifty / セキ

最後に頭脳をラジコンに
接続して完全自律へ。



3. ハードウェア① 【ロボットの構成要素】

2023.07.15

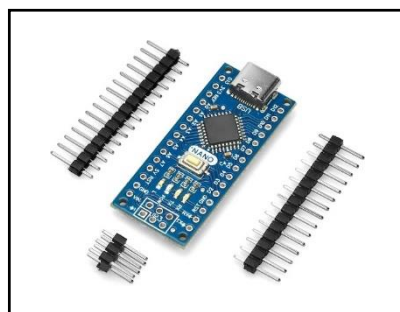
第21回 ROBO-ONE Conference(2)

<新ROBO-剣> Dimsifty / セキ

ArduinoNanoとRaspberryPi4をI2Cで接続。

- ・ RaspberryPi4の仕様上、5v $\leftrightarrow$ 3.3v のレベル変換が必要ない
- ・ Arduinoの仕様上、I2Cは割り込みで処理されるため通信がラク
- ・ ArduinoのUARTとSPIは使用済みで使えない

…といった条件から自然とI2Cに決定。



5v系

I2C



3.3v系

ArduinoNano

I2C

Raspberry Pi 4
Model B 4GB

●導入

1. 発表者紹介
2. 自律型ロボット製作のきっかけ

●展開

3. ハードウェア①【ロボットの構成要素】
4. **ハードウェア②【ロボットの基礎制御】**
5. ソフトウェア①【コンピュータの環境構築】
6. ソフトウェア②【打突アルゴリズムの構築】
7. 実際の試合【第1回新ROBO-剣】

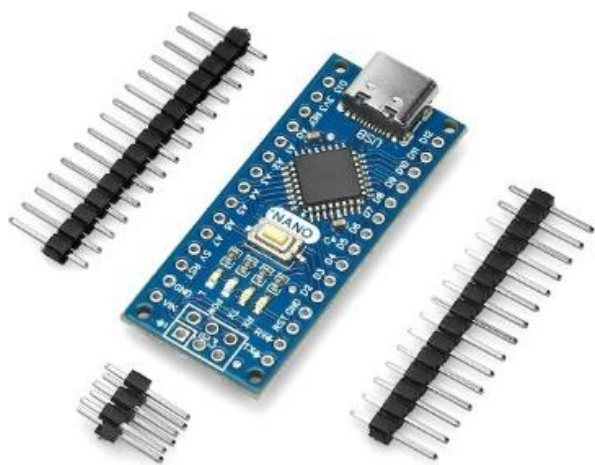
●まとめ

8. 新しいことに挑戦するには
9. 質疑応答

ロボットの基礎制御

→ Arduinoの中身。

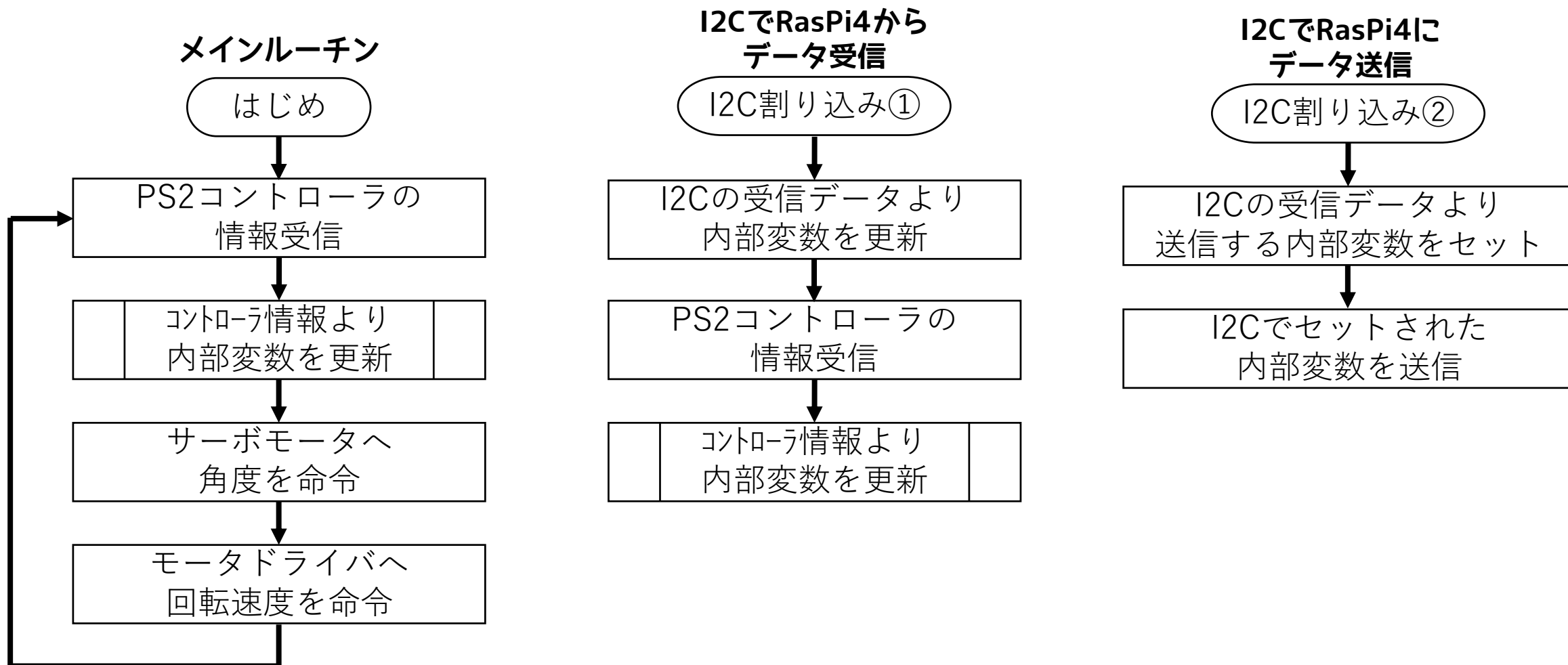
ハードウェアコントローラとしての機能を持たせる。



- ・ モータドライバへの命令
- ・ サーボへの命令
- ・ PS2コントローラのボタンデータの受信
- ・ 上3つの情報をRaspberryPi4と送受信

これらが不自由なく連立できる状態を作りたい。

Arduinoの内部処理



●導入

1. 発表者紹介
2. 自律型ロボット製作のきっかけ

●展開

3. ハードウェア①【ロボットの構成要素】
4. ハードウェア②【ロボットの基礎制御】
5. ソフトウェア①【コンピュータの環境構築】
6. ソフトウェア②【打突アルゴリズムの構築】
7. 実際の試合【第1回新ROBO-剣】

●まとめ

8. 新しいことに挑戦するには
9. 質疑応答

コンピュータの環境構築

→ラズパイに何の機能を乗せるか



- ・ RealSenseSDK (カメラを使う)
- ・ OpenCV (画像処理)
- ・ NumPy (数値計算)
- ・ SMBus (I2C通信)
- ・ simpleaudio (音声出力)

インストールするための記事は沢山あるが . . .

環境構築でよくあること

**前提知識が足りてないと
ネット記事通りにならなかった時に詰む**

- バージョン違いとかインストール順序の違いとか些細なことでうまくいかないことがある。
- 前提知識が無いので、その「些細なこと」を解決するのに膨大な時間がかかる

よくある詰みポイント①

「訳も分からずフリーズする」

→総当たりで原因探るしかない

→最悪諦める（OS選択しなおしに至ることもあった）

よくある詰みポイント②

「よくわからないエラーが出る」

→エラーが出るだけマシと考える

→エラーメッセージをとりあえず検索エンジンへ

→エラーによっては違う記事がヒットしまくる

環境構築を乗り越えるには . . .

①粘り強く時間をかける（当たり前）

→ 解決するために必要な一番の要素は “経験を積む” こと。

②英語の記事を見る

→ 調べたい内容を英語で検索する。ヒットする数が圧倒的に違う。

→ 記事を見るときは機械翻訳でもいい。

③SNSでエラーを検索する

→ 確率は低いが誰かの投稿が稀にヒットする。

●導入

1. 発表者紹介
2. 自律型ロボット製作のきっかけ

●展開

3. ハードウェア①【ロボットの構成要素】
4. ハードウェア②【ロボットの基礎制御】
5. ソフトウェア①【コンピュータの環境構築】
6. ソフトウェア②【打突アルゴリズムの構築】
7. 実際の試合【第1回新ROBO-剣】

●まとめ

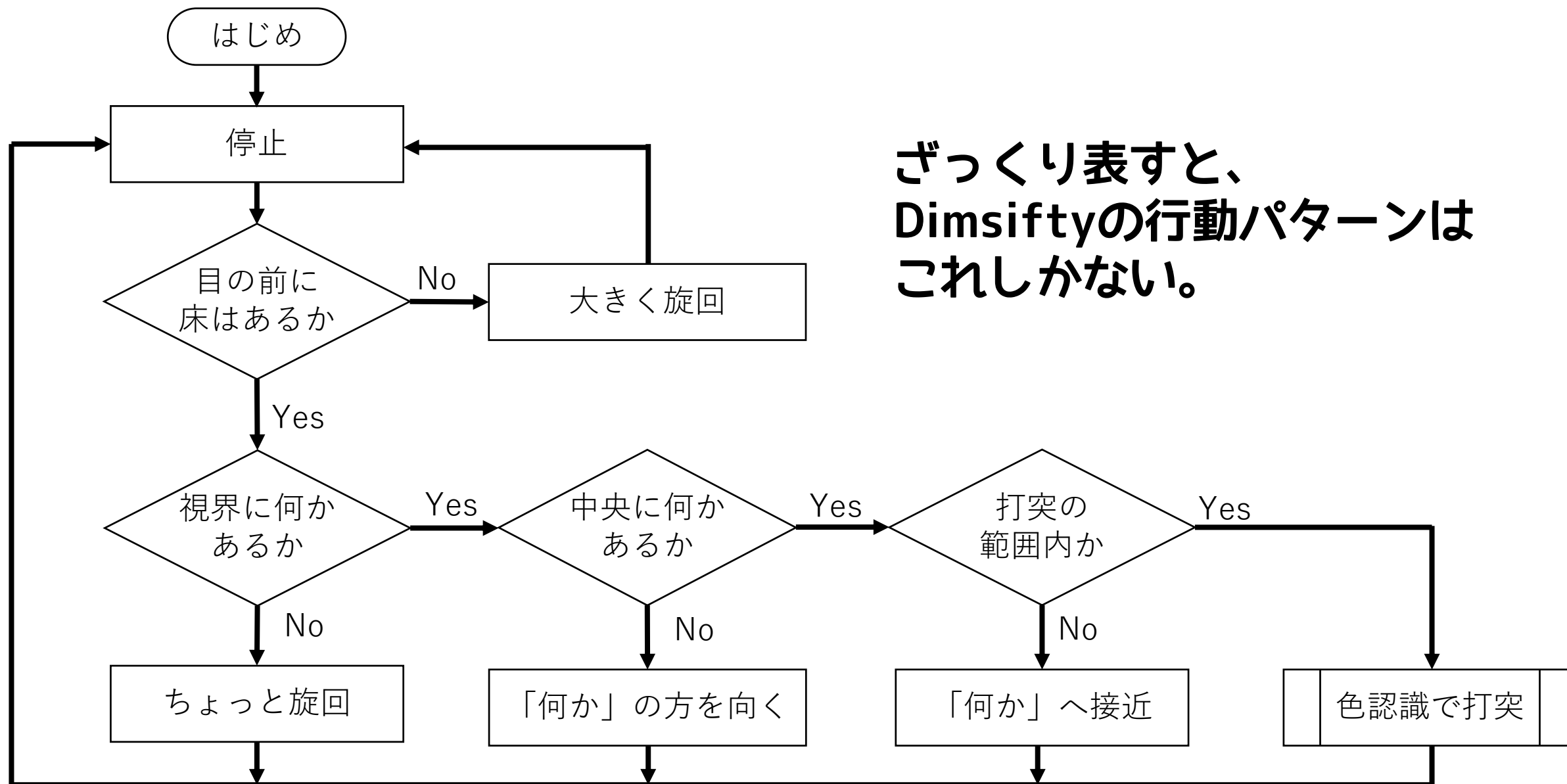
8. 新しいことに挑戦するには
9. 質疑応答

6. ソフトウェア② 【打突アルゴリズムの構築】

2023.07.15

第21回 ROBO-ONE Conference(2)

<新ROBO-剣> Dimsifty / セキ



処理
「停止」

停止

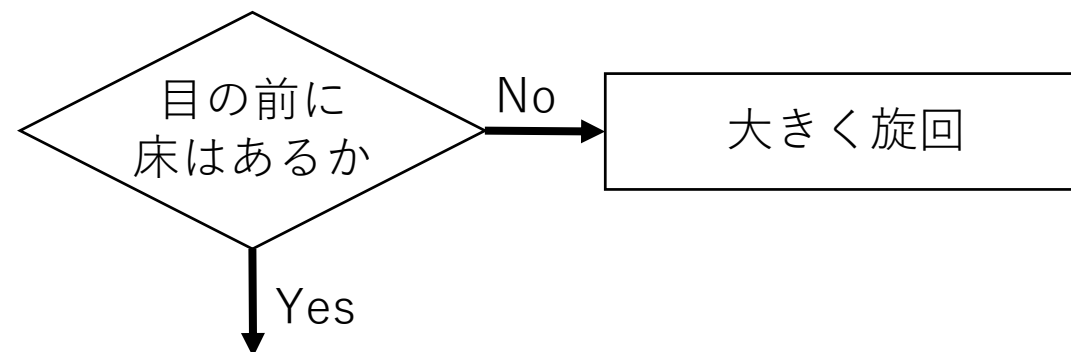
カメラの信頼性を上げるために、いちいち停止する。

→処理を高速に行う事は想定していない。

※D435のカメラは、動きながらも新ROBO-剣に最低限必要な
情報を得ることができるはずだが、止まらないと精度が出せなかった。

条件分岐

「目の前に床はあるか」



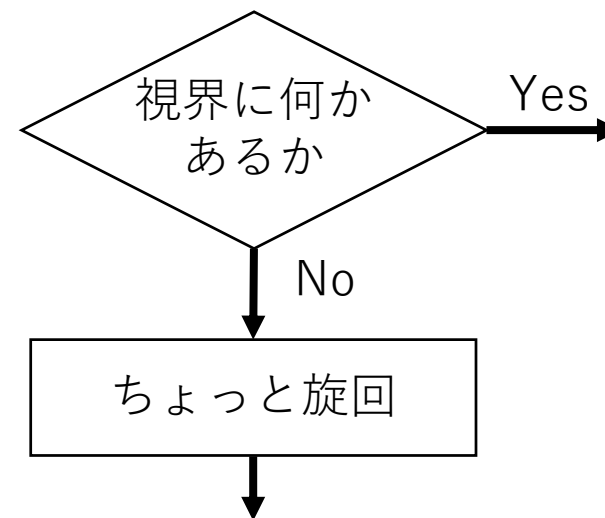
深度カメラの下方の深度を取得する。

床と深度カメラの角度と距離は一定なので、

想定値以上の距離が出たら「床がない」と判断される。

条件分岐

「視界に何かあるか」



深度カメラの中央横一直線の深度を取得する。

●設定した距離内に少しでも何かがあれば…

→その「何か」に対してアプローチを開始する。

●なにもなければ…

→視野を変えるためにちょっと旋回。

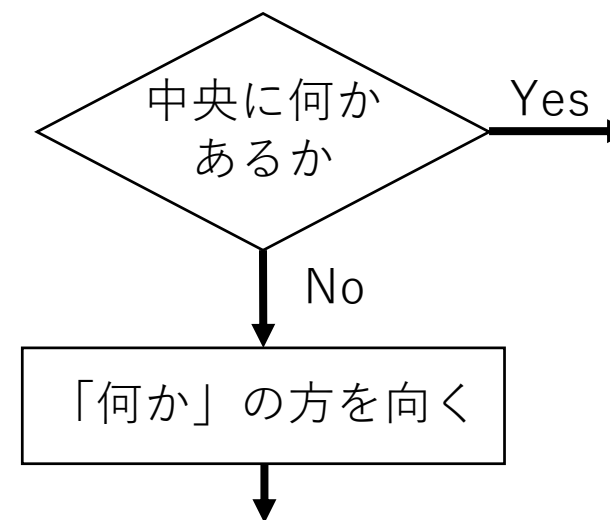
※永遠に旋回する状態になることを避けるため、
数回旋回したら前左右にランダムに動く。

条件分岐

「中央に何かあるか」

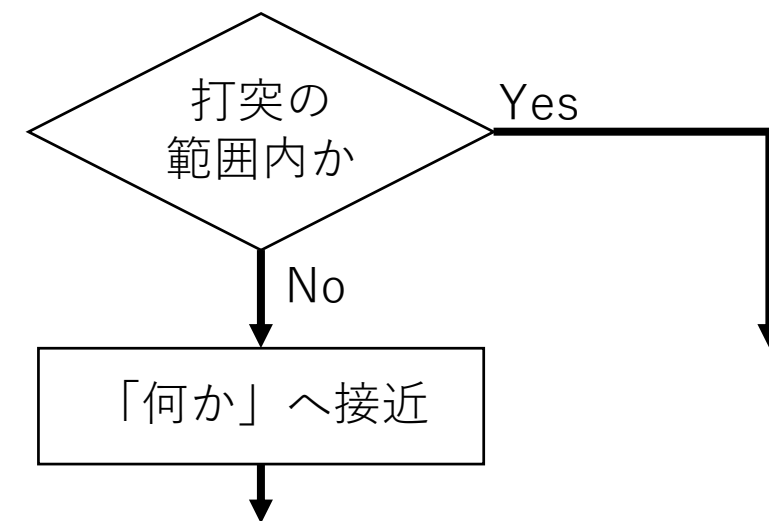
「何か」が中央横一直線の
どこで計測されたかに応じて分岐。

- 中央あたりで「何か」が検知されていたら…
→進行方向上に「何か」があるので次の段階へ
- 画面端に「何か」が検知されたら…
→「何か」が画面中央にくるように旋回



条件分岐

「打突の範囲内か」



中央あたりの「何か」が自機から
どれくらい離れているかに応じて分岐。

●竹刀が届かないくらい離れているなら...

→とりあえずちょっと前進して接近

●竹刀が届きそうな範囲なら...

→色認識を使って打突する。ここで初めて色認識が登場。



「何か」の正体とは…？

**現時点でロボットは「何か」が本当に
相手のロボットであるかどうかはわかっていない**

「中央横一直線に映り込む想定距離以内にある物体」を検知する条件
だと以下のようなモノすべてが「何か」の対象になってしまう。

- ・ 相手ロボット
- ・ 相手のオペレータ
- ・ 会場の設置物（机・椅子）
- ・ 自分自身
- ・ 審判
- ・ カメラと三脚

「何か」がロボットか否か判定するには？

競技規則上、リング上には自機と相手しかいないはず。

→「何か」が相手だとすれば、**どれだけ接近しても床は存在している**

→「何か」が相手じゃないとすれば、それはリング外の物体を検知しているため、**接近していけばいつか床が無くなる**

新ROBO-剣という競技以外では使い物にならない手法ではあるが…

リングから落ちないロボットになっていれば

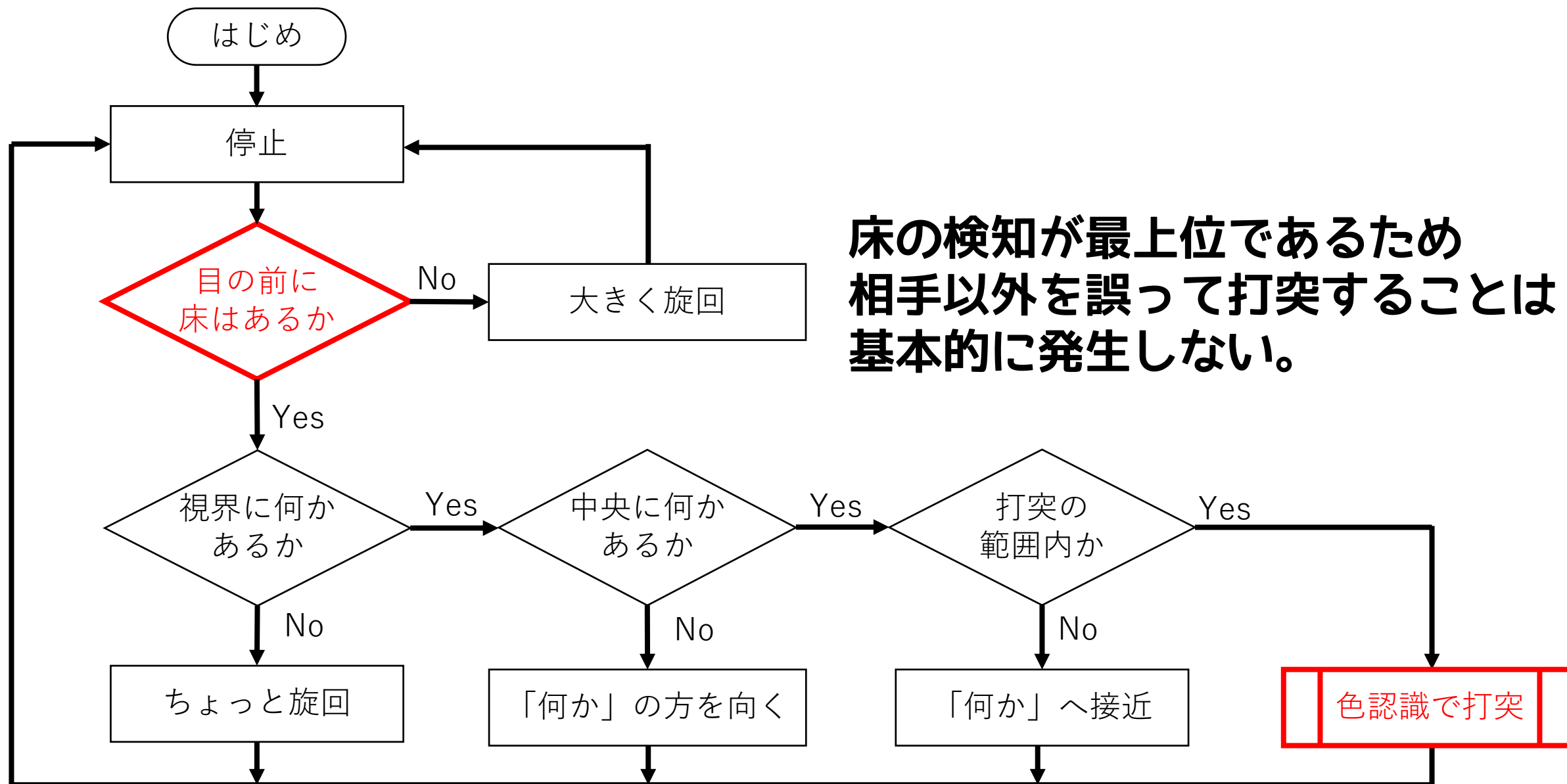
「何か」の正体がわからなくても**誤って打突することはない。**

6. ソフトウェア② 【打突アルゴリズムの構築】

2023.07.15

第21回 ROBO-ONE Conference(2)

<新ROBO-剣> Dimsifty / セキ



「何か」がロボットか否か判定するには？

→ 「リング上の物体であるかどうか」を床の存在で判別する。

●この手法のメリット
処理を作るのが簡単

→ 画像処理とか一切いらない。PSDセンサー数個でもできる。

●この手法のデメリット

接近しないと自機と相手の間に床があるかどうかの判定ができない

→ 近づくまでは誤ってリング外にあるものへ突き進んでしまう。

…でも検知の条件がキツすぎて同じ場所でじっとしている

よりかはマシだと思って割り切っている。

6. ソフトウェア② 【打突アルゴリズムの構築】

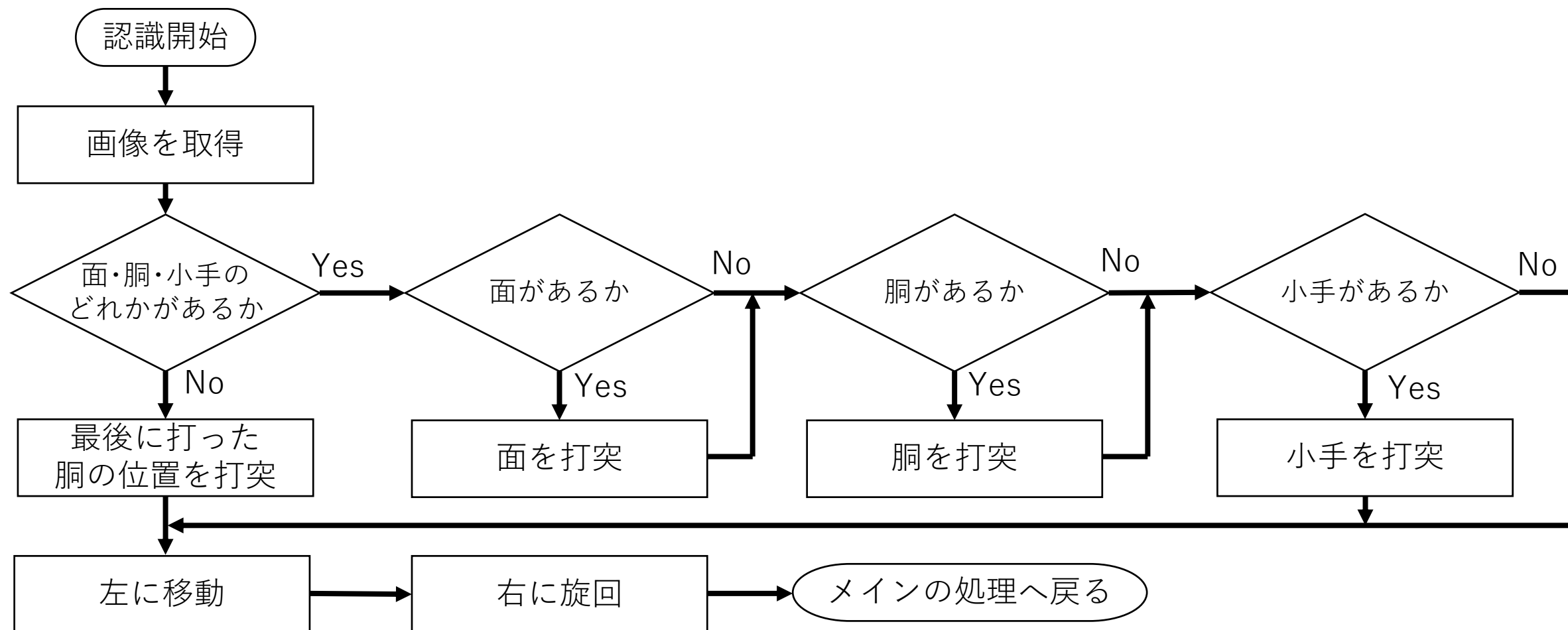
2023.07.15

第21回 ROBO-ONE Conference(2)

<新ROBO-剣> Dimsifty / セキ

定義済み処理

色認識で打突



処理

「画像の取得」

画像を取得

RGBカメラから画像を取得し、必要な下処理をする。

●深度カメラの情報と組み合わせ、必要ない部分を黒で塗りつぶす

→背景が消えるため**ロボットだけが映った画像が得られる。**

●ガンマ値・コントラスト・彩度を適度に上げる

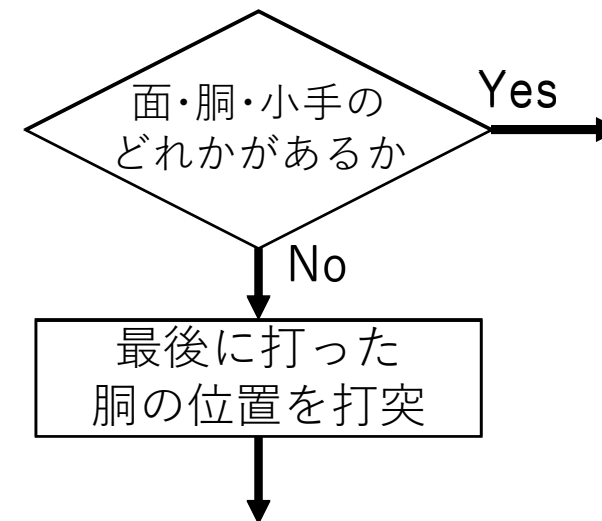
→画像としては悪くなるが、色認識するだけなら問題ない。

→**照明の変化に強い画像が得られる（と思った）。**

これらは OpenCV と RealSenseSDK が全部やってくれる。

条件分岐

「面・胴・小手のどれかがあるか」



画像に赤・青・黄の塊があるかどうかで分岐。

●少しでも塊があれば・・・

→あれば精密に打突するための処理に進む。

●塊がなければ・・・

→相手のロボットはいるはずなのに打突部位が見つからない状態。

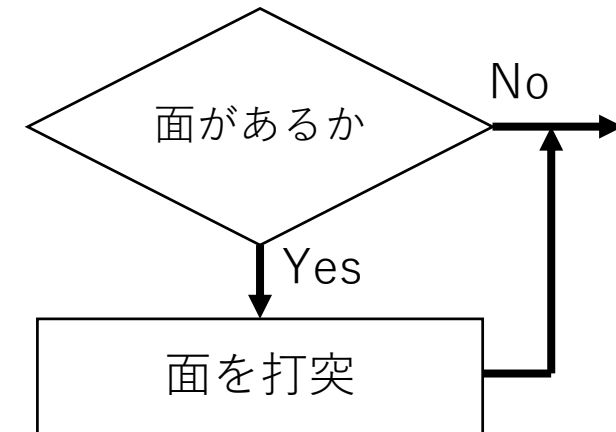
とりあえず当たることに賭けて打突してみる。

※現状の競技規則に打突ミスのペナルティが無いからできる事。

ロボットとしての正しい処理は「打突しない」だと思う。

条件分岐

「面(胴)(小手)があるか」



画像に赤(青)(黄)の塊があるかどうかで分岐。

●塊があれば・・・

→ **目標となる座標を計算**して打突する

●塊がなければ・・・

→ 次の色の判別に進む

※ **最低1色は必ず打突、最大で3色連続で打突する。**

～目標となる座標の決定方法～

- ①画像上における塊の重心座標をOpenCVで取得
→画像上でのX座標とY座標が確定
- ②深度カメラの情報を使い、重心座標での奥行きを取得
→実空間上でのZ座標が確定
- ③奥行きとD435の画角から実空間上の重心座標を算出
→実空間上でのX座標とY座標が確定

～打突ポーズの決定方法(逆運動学)～

①根本ロール軸の角度を決定

X座標とY座標の逆正接により算出できる。

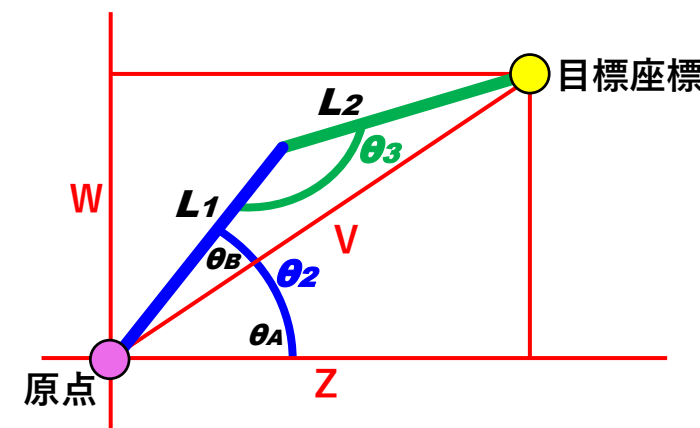
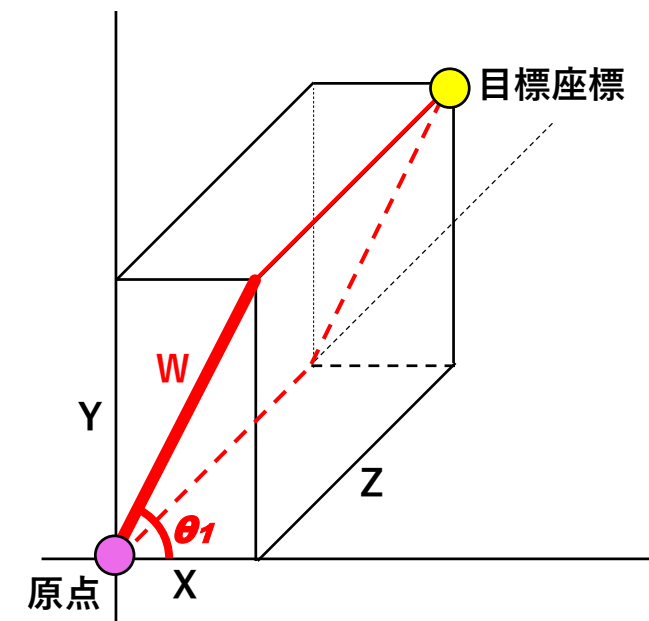
$$\theta_1 = \tan^{-1} \frac{Y}{X}$$

②根本ピッチ軸と手首ピッチ軸の角度を決定

余弦定理により算出できる。

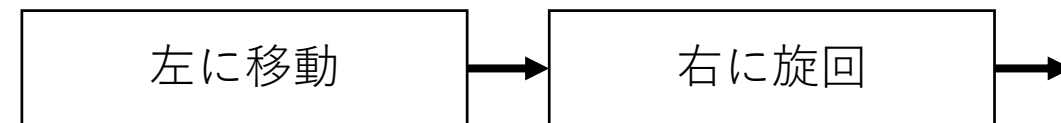
$$\theta_2 = \theta_A + \theta_B = \left(\tan^{-1} \frac{W}{Z} \right) + \left(\cos^{-1} \frac{L_1^2 + V^2 - L_2^2}{2 \times L_1 \times V} \right)$$

$$\theta_3 = \cos^{-1} \frac{L_1^2 + L_2^2 + V^2}{2 \times L_1 \times L_2}$$

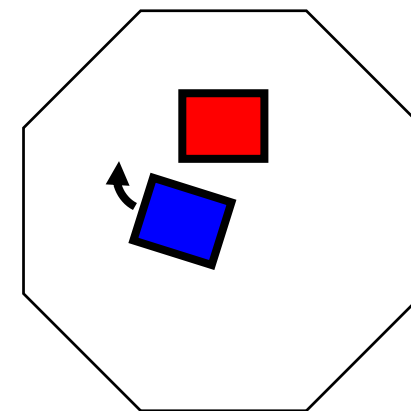


処理

「左に移動」 「右に旋回」



同じ場所で打突し続けないように
打突位置を少しずつ移動させていく。



お互いに打突範囲内になると竹刀の振り合いになるが、
こちら移動→打突→移動→打突・・・を繰り返すので
相手はこちらの移動に追従してから打突する必要がある

→こちらが先手をとれる確率が上がる（と思った）

●導入

1. 発表者紹介
2. 自律型ロボット製作のきっかけ

●展開

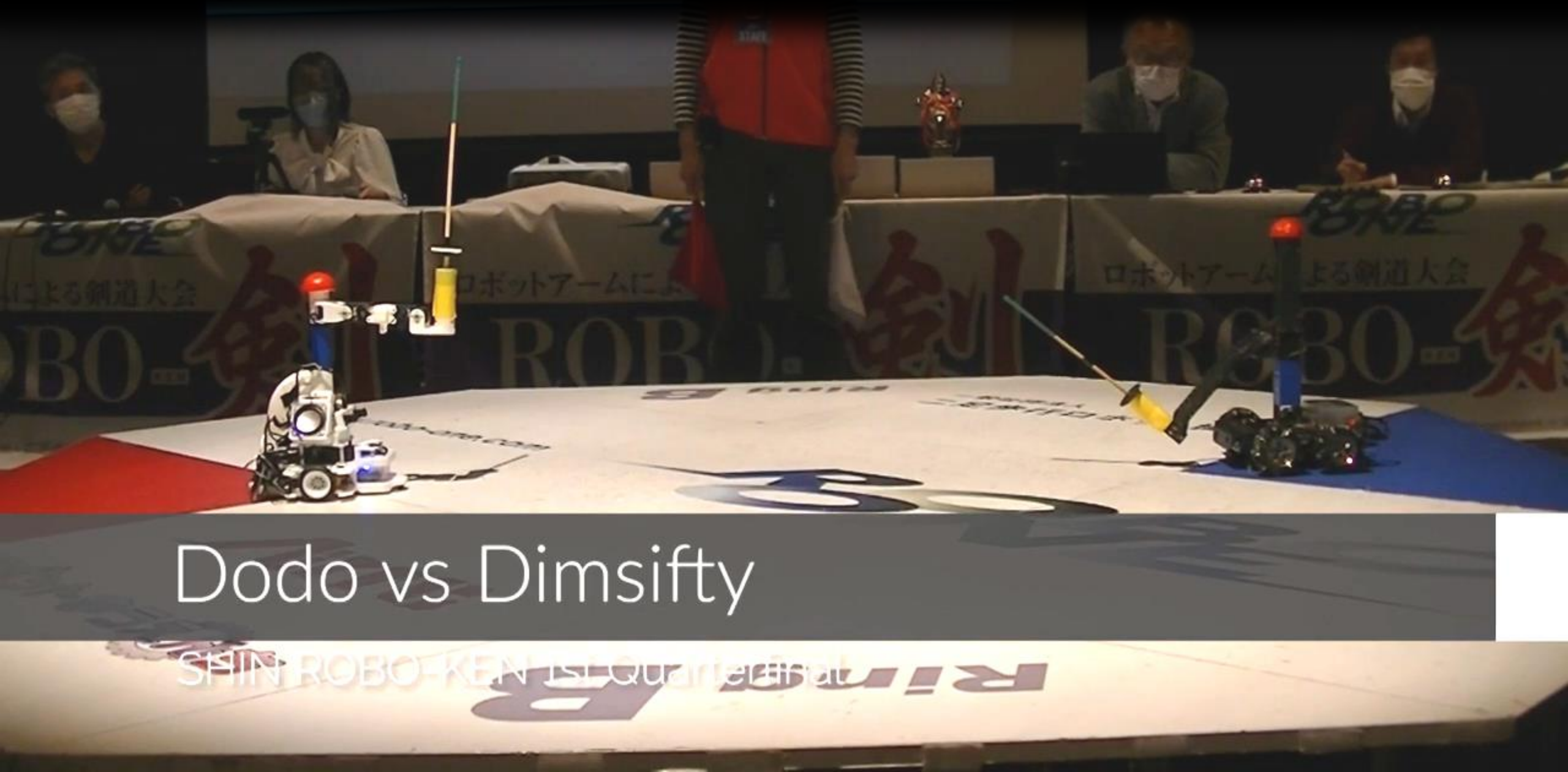
3. ハードウェア①【ロボットの構成要素】
4. ハードウェア②【ロボットの基礎制御】
5. ソフトウェア①【コンピュータの環境構築】
6. ソフトウェア②【打突アルゴリズムの構築】
7. 実際の試合【第1回新ROBO-剣】

●まとめ

8. 新しいことに挑戦するには
9. 質疑応答

Dodo vs Dimsifty | SHIN ROBO-KEN 1st Quarterfinal

<https://www.youtube.com/watch?v=hIHpYRWi0Q>



Dodo vs Dimsifty

SHIN ROBO-KEN 1st Quarterfinal

会場で起きた大問題

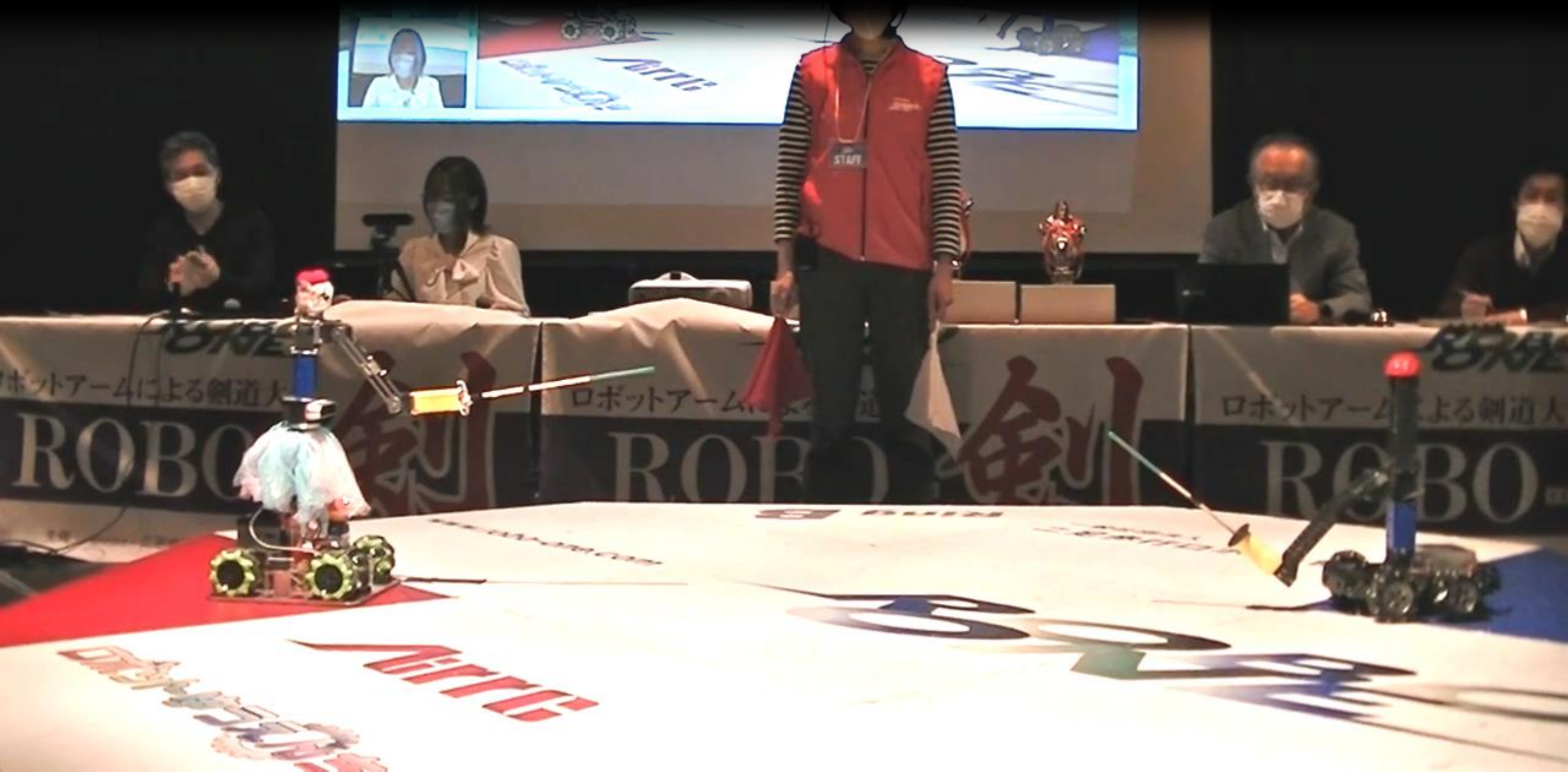
「Wi-Fiが停止する」

2.4GHzのWi-FiはUSB3.0と干渉するため、
パソコンとRasPi4の通信は5GHzのWi-Fiを使用していた。
→使っていたポケットWi-Fiをアクセスポイントとしていたが
会場では何故かDFS機能が多発して接続が何度も途切れた。

※DFS機能とは、野外で使われる5GHzのWi-Fiが
気象レーダーなどに影響を与えないようにする機能のこと。

KOBIS vs Dimsifty | SHIN ROBO-KEN 1st Semifinal

<https://www.youtube.com/watch?v=nv6lII2i1UQ>



これからの新ROBO-剣に求められそうなこと

動いているロボットにどうやって打突するか

打突されないように
動き続ける
(打突されない)



確実に打突できる
ように停止する
(打突しやすい)

理想は「**自機が動きながら、動いている相手を打突できる**」ロボット
→ 実験大会ではそれに近い動作ができそうなロボットがいた。

●導入

1. 発表者紹介
2. 自律型ロボット製作のきっかけ

●展開

3. ハードウェア①【ロボットの構成要素】
4. ハードウェア②【ロボットの基礎制御】
5. ソフトウェア①【コンピュータの環境構築】
6. ソフトウェア②【打突アルゴリズムの構築】
7. 実際の試合【第1回新ROBO-剣】

●まとめ

8. 新しいことに挑戦するには
9. 質疑応答

8. 新しいことに挑戦するには

- ・ **つくる部分を分割しよう（目標の細分化）**
達成感を感じられるポイントを増やす。
- ・ **未完成でもいいから作品を表に出そう（期日の設定）**
目標を設定することが自分を追い込む要素になる。
- ・ **過去の自分と比べて評価しよう（自己分析）**
自分の適性に合ったロボット作りの進め方を理解する。
- ・ **遅すぎることもなんて無い（踏み出す勇気）**
やりたいと思った時からきっともう始まってるんだと思う。

●導入

1. 発表者紹介
2. 自律型ロボット製作のきっかけ

●展開

3. ハードウェア①【ロボットの構成要素】
4. ハードウェア②【ロボットの基礎制御】
5. ソフトウェア①【コンピュータの環境構築】
6. ソフトウェア②【打突アルゴリズムの構築】
7. 実際の試合【第1回新ROBO-剣】

●まとめ

8. 新しいことに挑戦するには
9. 質疑応答



公開資料の更新や当日の追加資料があればこちらで公開する予定です。

また、資料につきましてご質問がありましたらご記入ください。

<https://forms.gle/j4Ba73vbVnNtrmGGA>

ご清聴ありがとうございました。