

# RaspberryPiで RCB-4をコントロール

近藤科学株式会社  
技術部 信原 卓弥

# 1.はじめに

- 会社紹介
- 自己紹介
- RaspberryPiとRCB-4のデモ機の紹介

# 近藤科学会社紹介

ラジコン部門

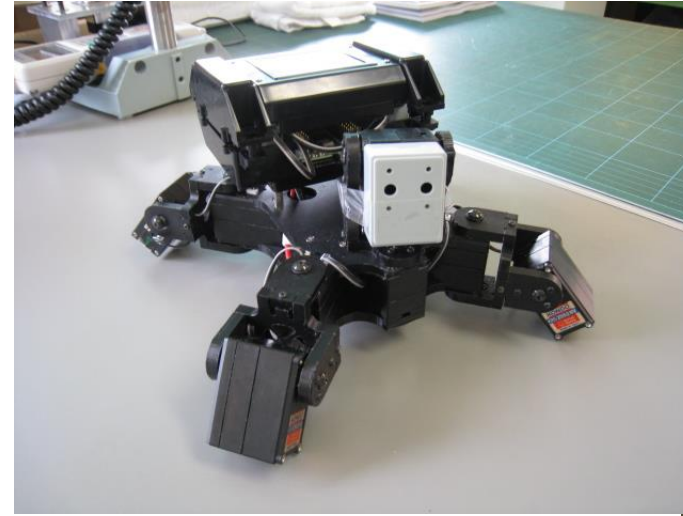


ロボット部門



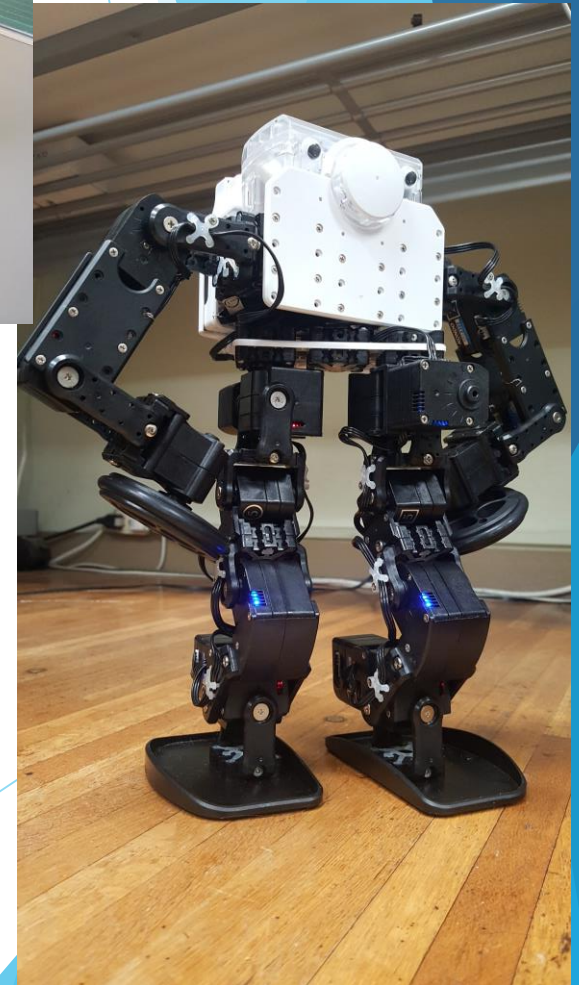
# 自己紹介

名前: 信原 卓弥

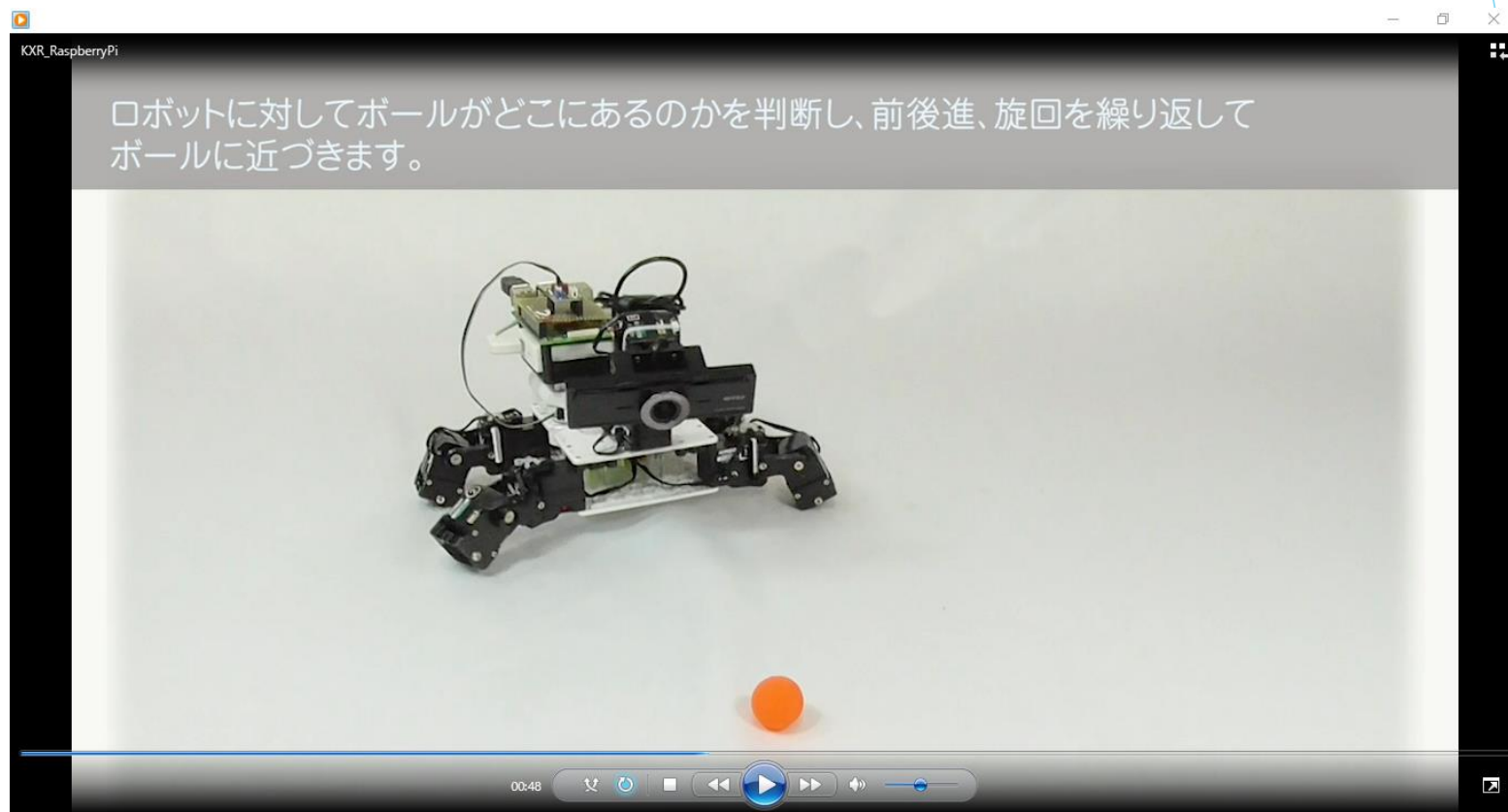


個人でKinectで動く  
ロボットを作りました

RCB-4 Library for Python  
ICS Library for Arduino の開発者本人



# 「RaspberryPiでRCB-4をコントロール」でできること



[https://www.youtube.com/watch?v=\\_0QYJdzrOmE](https://www.youtube.com/watch?v=_0QYJdzrOmE)

詳しくは『RaspberryPiとRCB-4との連携について』をご覧ください。

# 「RaspberryPiでRCB-4をコントロール」

1. はじめに
2. **RCB-4およびHeartToHeart4について**
3. Pythonライブラリについて
4. RCB-4を外部から動かす方法
5. KXR-L4 カメ型ロボットに搭載した事例紹介
6. 今後の展望

# RCB-4

KRSサーボモータを組み合わせることでロボットの動作(モーション)を動作させるコントロールボード



## RCB-4HV

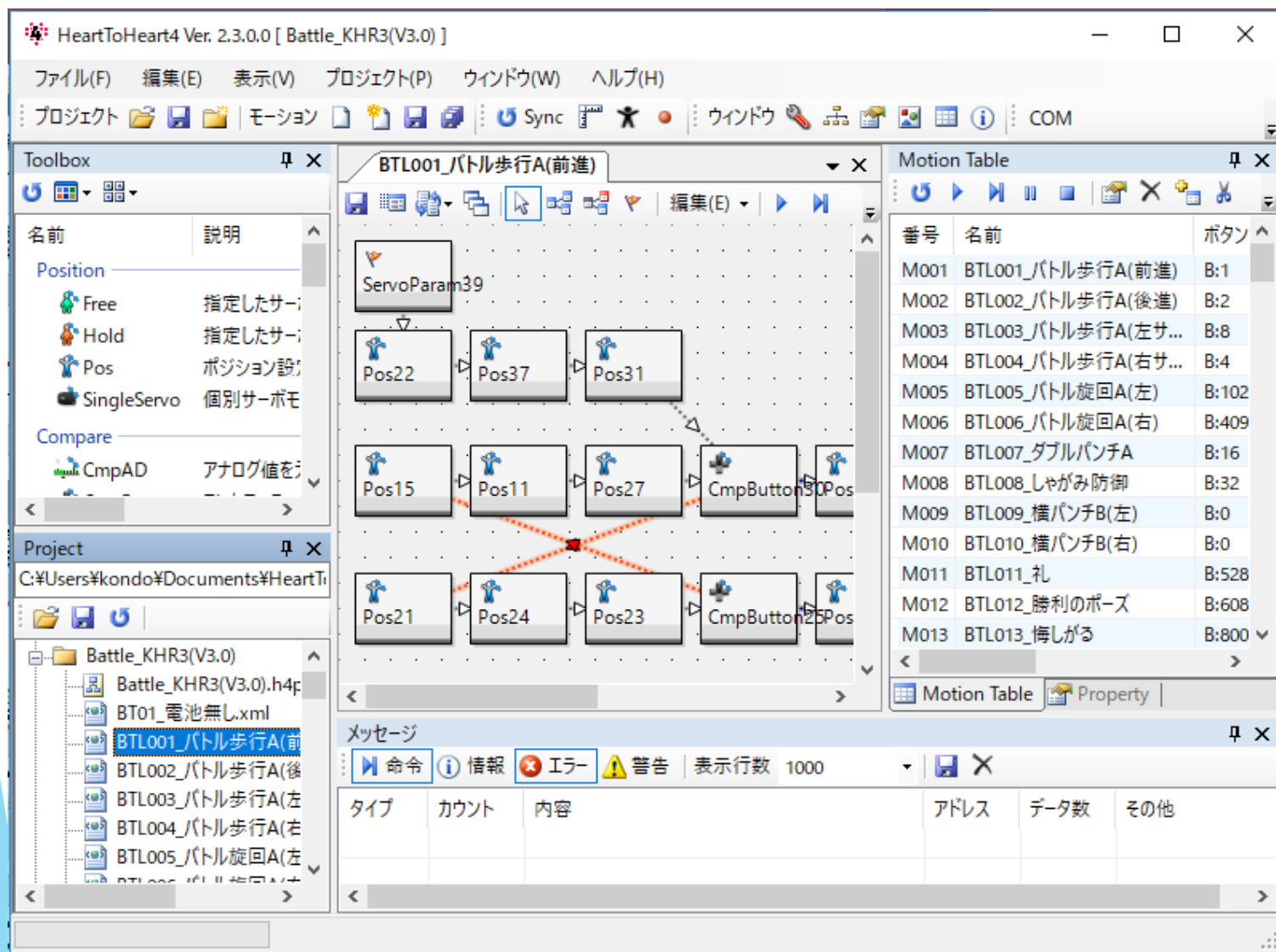
- ・KHR-3HVに搭載されている
- ・すべてのKRSサーボに対応
- ・AD端子が10本
- ・PIO端子も10本実装されている



## RCB-4mini

- ・KXRシリーズに搭載されている
- ・KRS-3300など小型サーボ向け
- ・AD端子が5本でPIO端子はなし
- ・低電圧でも誤動作しないように電源の強化

# HeartToHeart4



・ロボットのモーションを作成するソフトウェア

・GUIで簡単に作成できる

・送信機のボタンやセンサによる分岐、繰り返し、モーションジャンプなど多彩なコントロール類

・トリム調整やセンサのミキシングなど機能が豊富



# RCB-4のできること

## RCB-4構成のロボット

RCB-4



AD入力センサ

KRSサーボモータ



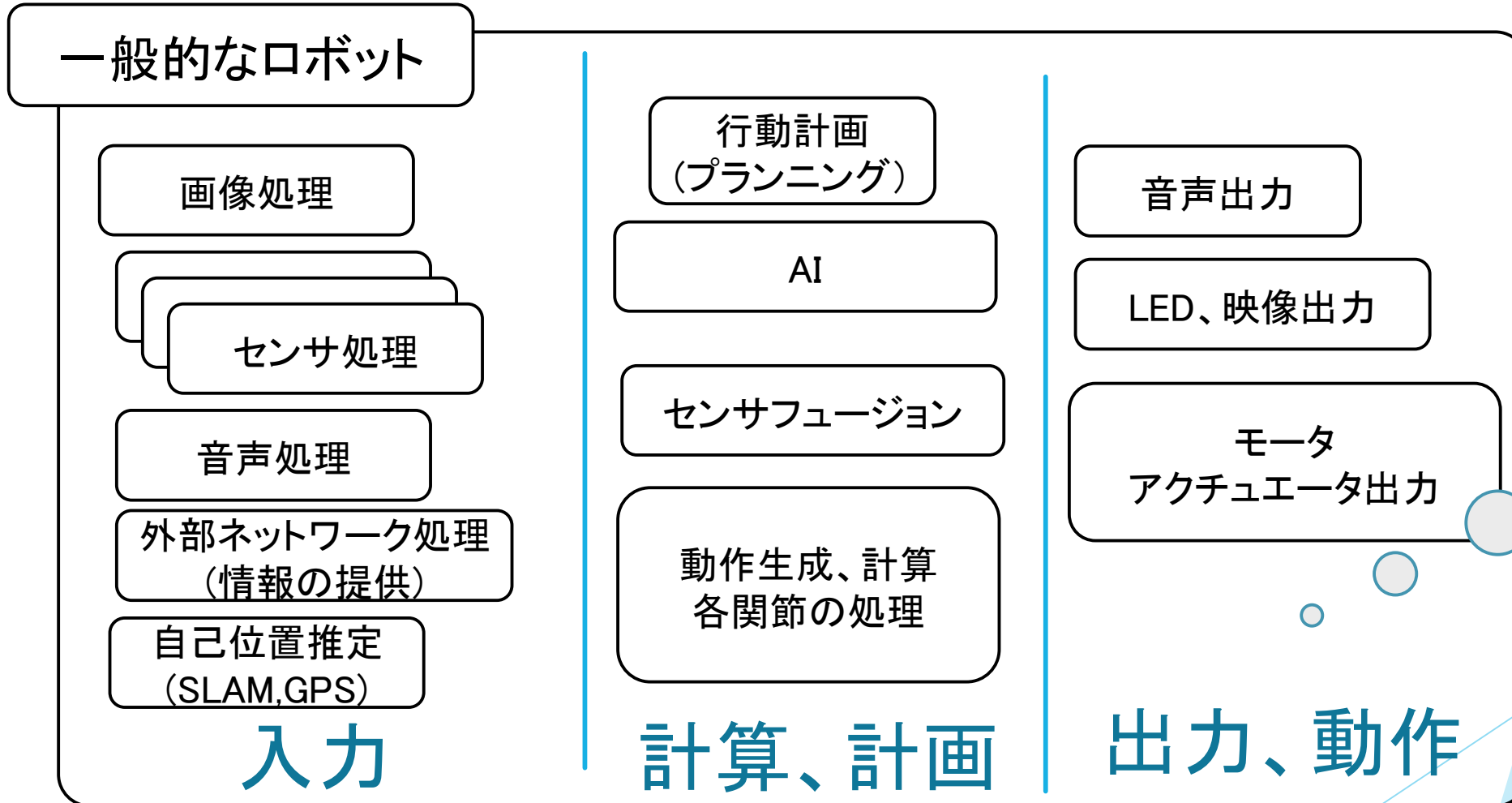
## ● 利点

- モーションを容易に作成できる
- 調整があまり難しくない
- 見た目ですぐ動くかわかる
- モータの指示があまり気にならない

## ● 欠点

- モーション再生に特化しているので決められた動作しかできない
- センサの入力が少ない

# 一般的なロボット



ソフトウェア  
以外に  
機構や回路  
とかもある

これだけあるとやれることに限界がある

# RCB-4と組み合わせる

## RCB-4を組み込んだロボット

画像処理

センサ処理

音声処理

外部ネットワーク処理  
(情報の提供)

自己位置推定  
(SLAM, GPS)

行動計画  
(プランニング)

AI

センサフュージョン

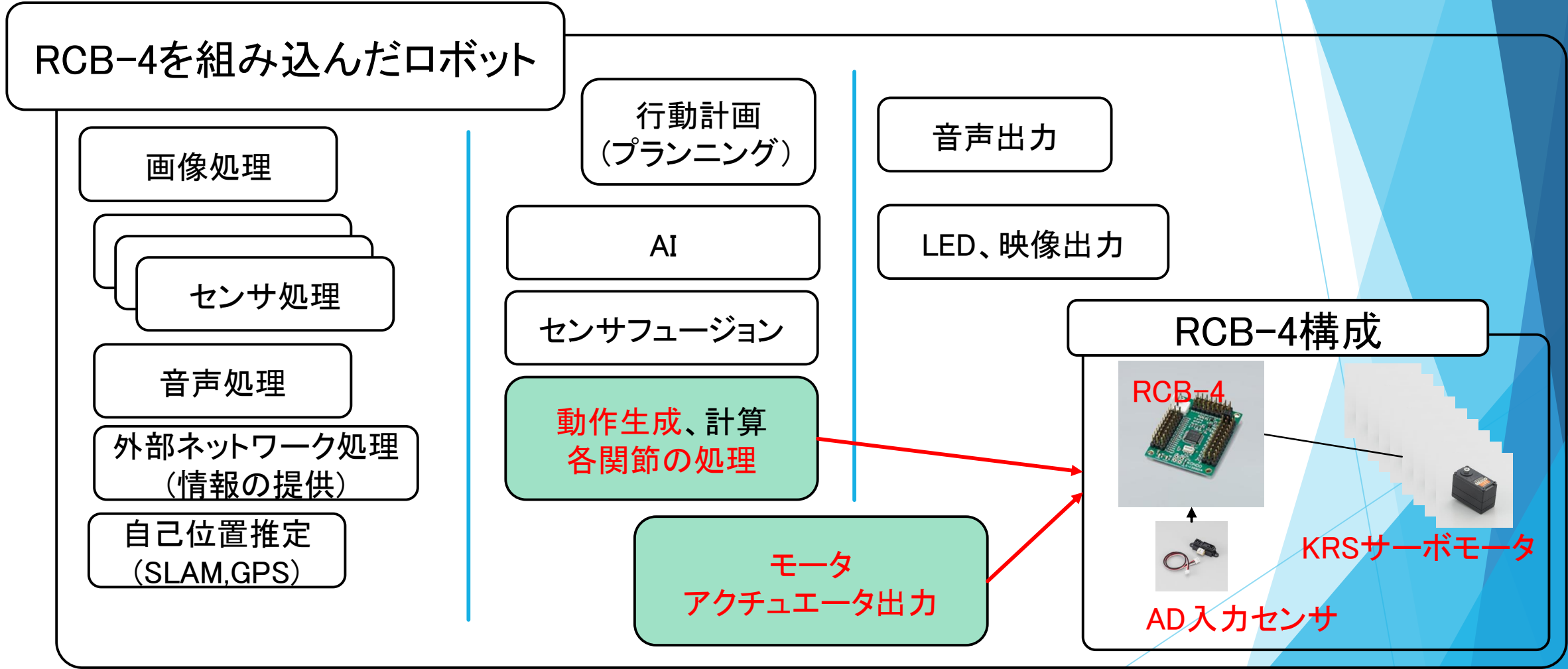
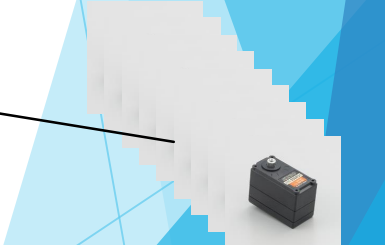
動作生成、計算  
各関節の処理

モータ  
アクチュエータ出力

音声出力

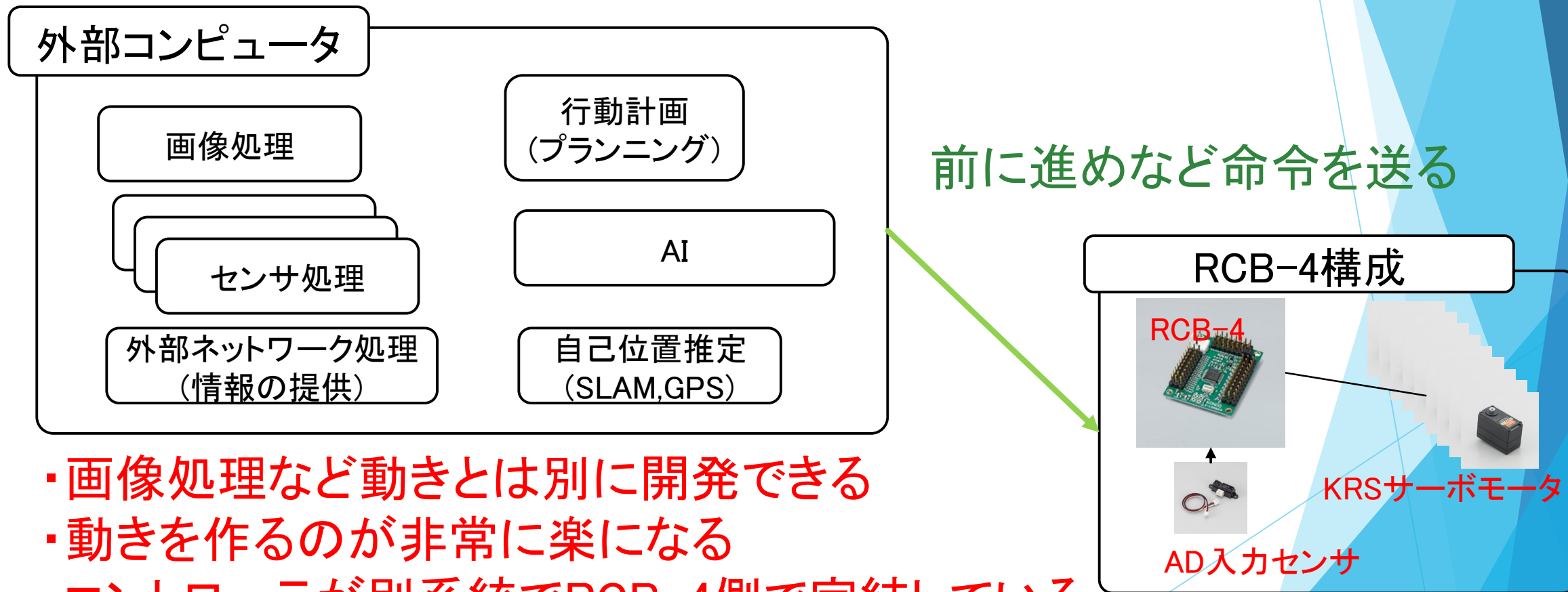
LED、映像出力

## RCB-4構成



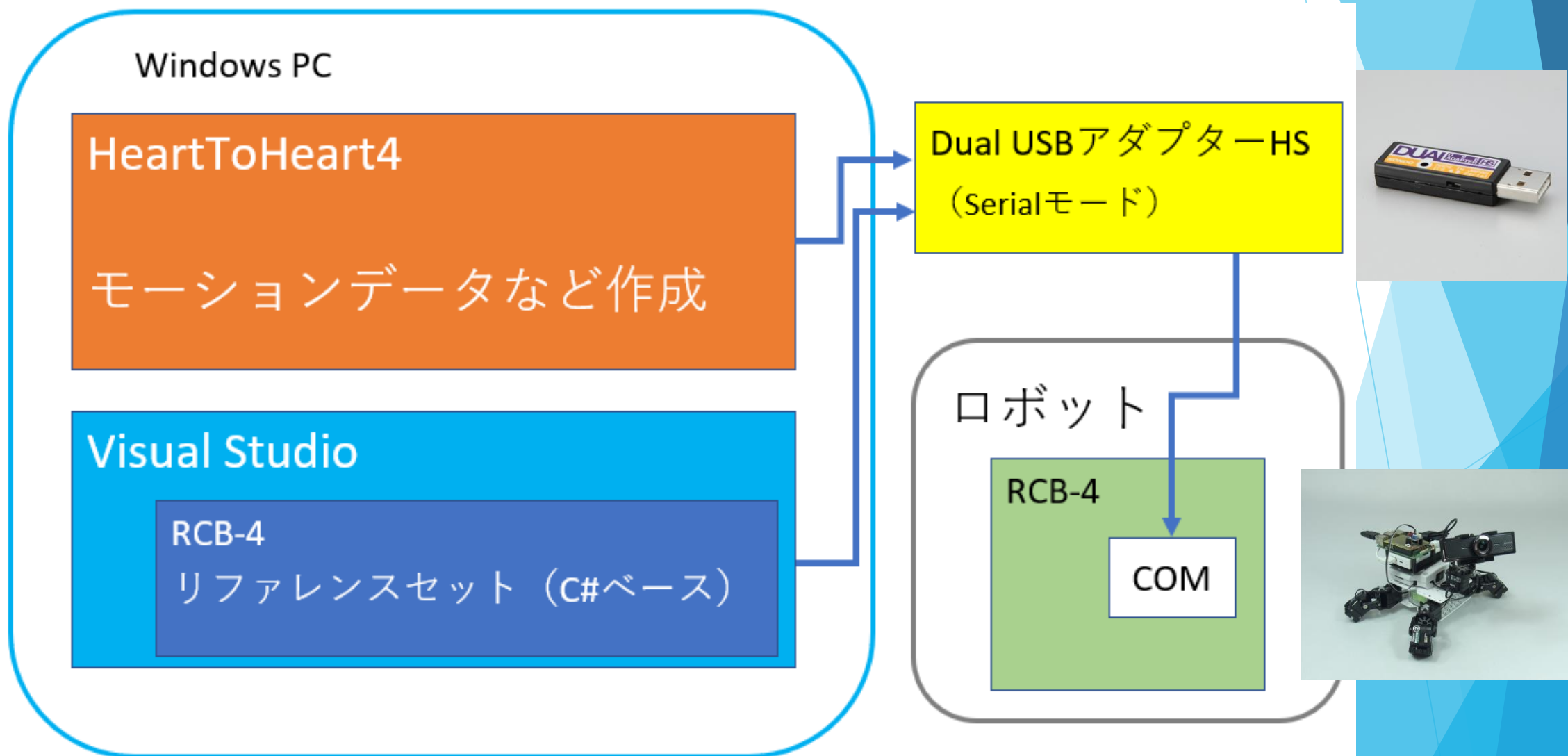
# RCB-4と組み合わせる

RCB-4の構成を組み込むと「**役割を切り分ける**」ことができる



- ・画像処理など動きとは別に開発できる
- ・動きを作るのが非常に楽になる
- ・コントローラが別系統でRCB-4側で完結している
- ・サーボモータの通信等は外部コンピュータは負担しない

# RCB-4に外部からアクセスする方法 (WinPC)



# RCB-4に外部からアクセスする方法 (WinPC)

詳しくは 弊社HP => テクニカルガイド  
=> RCB開発資料

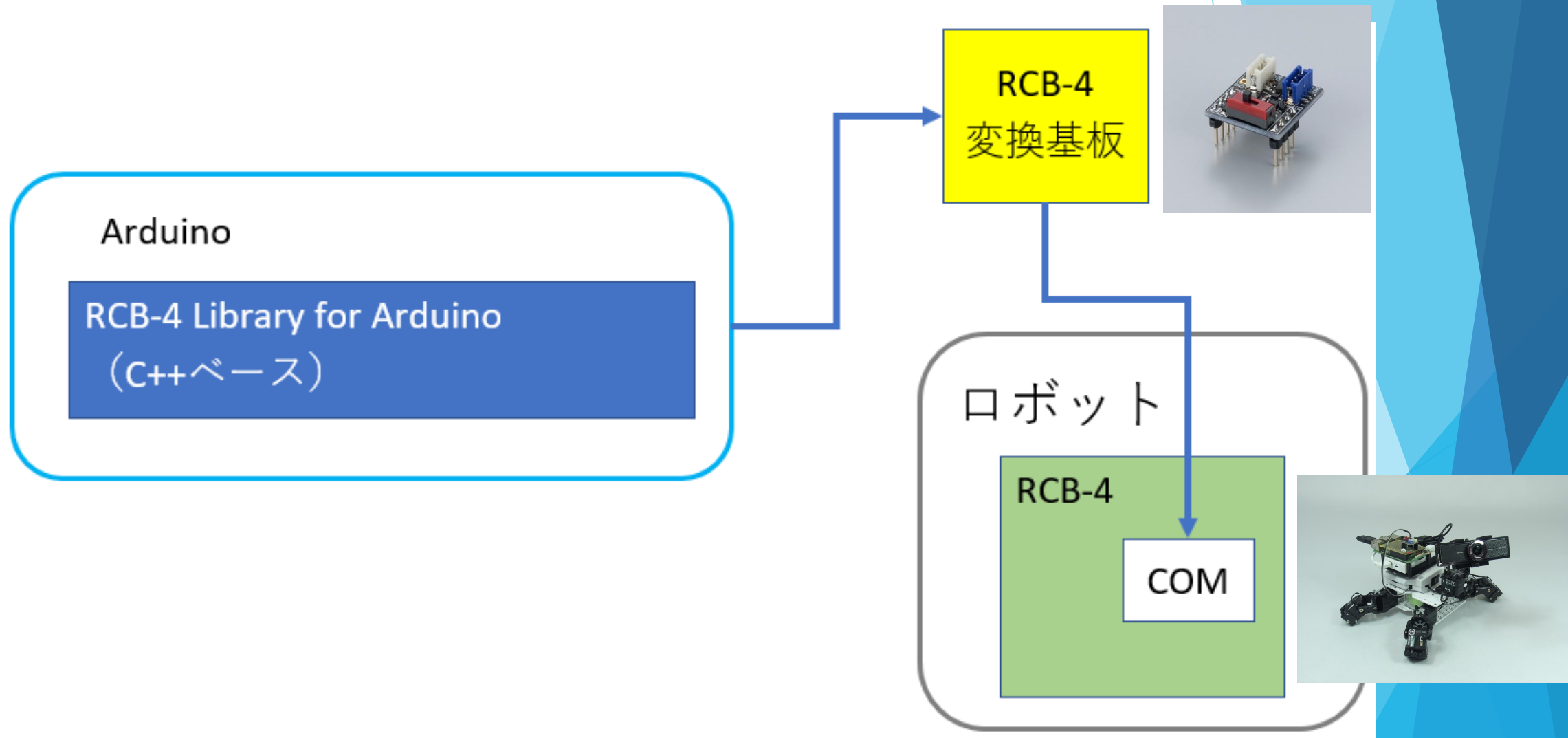
[https://kondo-robot.com/archives/faq\\_category/rcb-dep](https://kondo-robot.com/archives/faq_category/rcb-dep)

『[RCB-4のリファレンスセット](#)』

(HTH4のバージョンによって保存先のアドレスが違うので注意)

『[KMR-M6をリモートブレインで動かす](#)』

# RCB-4に外部からアクセスする方法(マイコン等)



# RCB-4に外部からアクセスする方法(マイコン等)

詳しくは 弊社HP => テクニカルガイド  
=> RCB開発資料

[https://kondo-robot.com/archives/faq\\_category/rcb-dep](https://kondo-robot.com/archives/faq_category/rcb-dep)

『[RCB-4 Library for Arduino ver.1.1](#)』

(※最新版はV1.1です。(2019/6現在))

『[RCB-4変換基板の使用方法](#)』



# RCB-4に外部からアクセスする方法 (Python)

Pythonが動くPC  
LinuxPCやRaspberryPiなど  
シングルボードコンピュータ

RCB-4 Library for Python

Dual USBアダプターHS  
(Serialモード)  
またはRCB-4変換基板

ロボット

RCB-4

COM



# 「RaspberryPiでRCB-4をコントロール」

1. はじめに
2. RCB-4およびHeartToHeart4について
3. Pythonライブラリについて
4. RCB-4を外部から動かす方法
5. KXR-L4 カメ型ロボットに搭載した事例紹介
6. 今後の展望

# RCB-4 Library for Python

- 概要

RCB-4をPython経由で動かすためのライブラリ

- 弊社HPからダウンロード

詳しくは 弊社HP => テクニカルガイド => RCB開発資料

[https://kondo-robot.com/archives/faq\\_category/rcb-dep](https://kondo-robot.com/archives/faq_category/rcb-dep)

## 『[RCB-4 Library for Python ver.1.0.0B](#)』

- サンプルプログラムを全て載せていないのでベータ版
- バグがない限り変更の予定なし

▪ MITライセンスなので、自己責任でフリーで使っていただいて構いません

# ライブラリファイルの中身

Rcb4Lib\_for\_Python\_V100B/

└ Rcb4Lib (PythonベースのRCB-4ライブラリ)

└ sample (サンプルプログラム)

| └ Rcb4AckTest.py

| | .....

| └ (sample ...etc)

└ FunctionList V100B(See to index\_html) (htmlベースの関数一覧)

└ readmeTop.md (readme ファイル)

# ライブラリファイルの中身

- サンプルプログラム

- ACK
- Rcb4MotionPlay
- Rcb4SetKrrButton

※ベータ版のため一部整備が不十分な場合があります。

- FunctionList

- doxygenでソースコードから関数一覧をHTML形式で列挙しています
- index.htmlをご覧ください

# ライブラリ動作条件

- **【重要】**Python 3.4以降  
enumを使用のため
- pySerial  
Python経由でシリアル通信を行うパッケージ  
USBのシリアルでもUARTでもここからアクセスできる
- RCB-4とPythonの動くコンピュータが接続されていること
  - USB経由 => ドライバのインストール
  - UART経由 => UARTとRCB-4は逆論理なので、  
RCB-4変換基板を間に入れる
- pySerial経由でデータをやり取りできるようにポートを確認しておくこと

# サンプルプログラムの動かし方

- Zipファイルで供給しているので、そのまま解凍してください
- サンプルプログラムは相対パスでライブラリのフォルダをリンクしているので、**解凍するときは同じフォルダに解凍してください。**
- sampleフォルダ内のPythonプログラムを実行してください。  
接続が確立、設定できていれば動きます  
※COMの設定部分は機体に合わせて書き換えてください  
RCB-4にはサンプルのプロジェクト、モーションを入れておいてください

# 「RaspberryPiでRCB-4をコントロール」

1. はじめに
2. RCB-4およびHeartToHeart4について
3. Pythonライブラリについて
4. **RCB-4を外部から動かす方法**
5. KXR-L4 カメ型ロボットに搭載した事例紹介
6. 今後の展望



# RCB-4を外部から動かす方法

- モーションの再生方法

- RCB-4でモーションとは別に個別にサーボを動かす方法

# 外部からのモーション再生について

## HTH4モーションテーブル

### ① MotionPlay

モーション番号を指定して直接モーションを起動する方法

(Rcb4MotionPlay)

### ② SetButton

KRR等のボタンデータをトリガにしてモーションを起動する方法

(Rcb4SetKrrButton)

番号	名前	ボタン番号	比較
M001	BTL001_バトル歩行A(前進)	B:1	=
M002	BTL002_バトル歩行A(後進)	B:2	=
M003	BTL003_バトル歩行A(左サ...	B:8	=
M004	BTL004_バトル歩行A(右サ...	B:4	=
M005	BTL005_バトル旋回A(左)	B:1024	=
M006	BTL006_バトル旋回A(右)	B:4096	=

モーション番号

ボタン番号

# 直接モーションを起動する方法(MotionPlay)

- ・直接モーションを起動方法させる方法

①再生しているモーションを中断(サスペンド)する



②プログラムカウンタのリセット



③モーションのアドレスをセット



④モーションをリスタートする

- ・ジャンプ命令だけで行くと危険なので現在再生している モーションを中断してからジャンプします

# モーションジャンプをほかのモーション再生中に行った場合



詳しくは弊社ウェブサイトの『KMR-M6をリモートブレインで動かす(6回目)』をご覧ください。

- ・モーション終了前に違うモーションを再生すると意図しない動きになる
- ・RCB-4のPythonライブラリに現在の再生しているモーション番号を取得できる関数を用意(getMotionPlayNum())

# ボタンデータをトリガでモーションを起動する (SetButton)

- ・KRCから受信するデータ部分に直接データを上書きします

RAMアドレス	機能
...	...
\$0088h	ICS34割り当てアドレス
\$008Ah	ICS35割り当てアドレス KRR(SIO5-8,ID31)のデータが保存されるアドレス
...	...

このデータを直接  
変えます

- ・HTH4のサンプルモーションでボタン長押しのもーションが  
組まれているので、連続動作をさせることができる

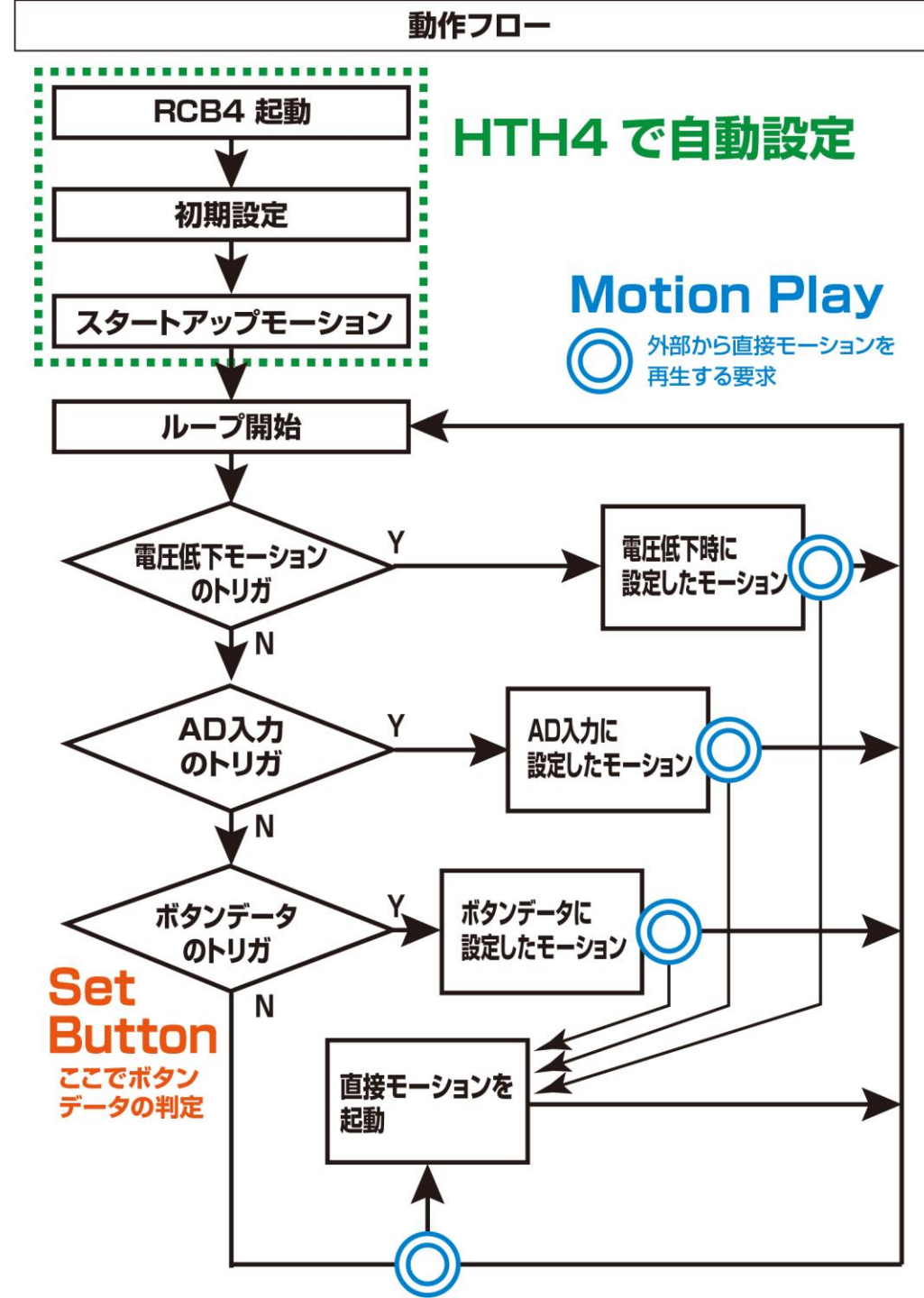
# 直接起動とボタン起動 の実行順序

## ① MotionPlay

モーション再生中に強引に中断して再生させる

## ② SetButton

ボタンデータのトリガ判定時に判定しモーションを再生させる



# 直接起動とボタン起動の使い分け

直接モーションの起動 (MotionPlay)	ボタンデータをトリガにした起動 (SetButton)
モーションの再生がどこからでもできる	他のモーションが再生している間は再生されない
モーションの再生の切り替えはうまく切り替えないと倒れたりバランスを崩したりする	モーションはホームポジションまで戻ってから再生の判定をされるため、バランスを崩すことがない
KRRが接続されていても動作させることができる	KRRが接続されていたらトリガが上書きされるので再生ができない
最後までモーションを再生してから同じモーションを再生させることになるので、連続動作にはならない	ボタン判定の連続動作がサンプルモーションとして用意されているので、連続動作の息継ぎがなく動かすことができる

得手不得手があるのでうまく使い分けてください

# RCB-4を外部から動かす方法

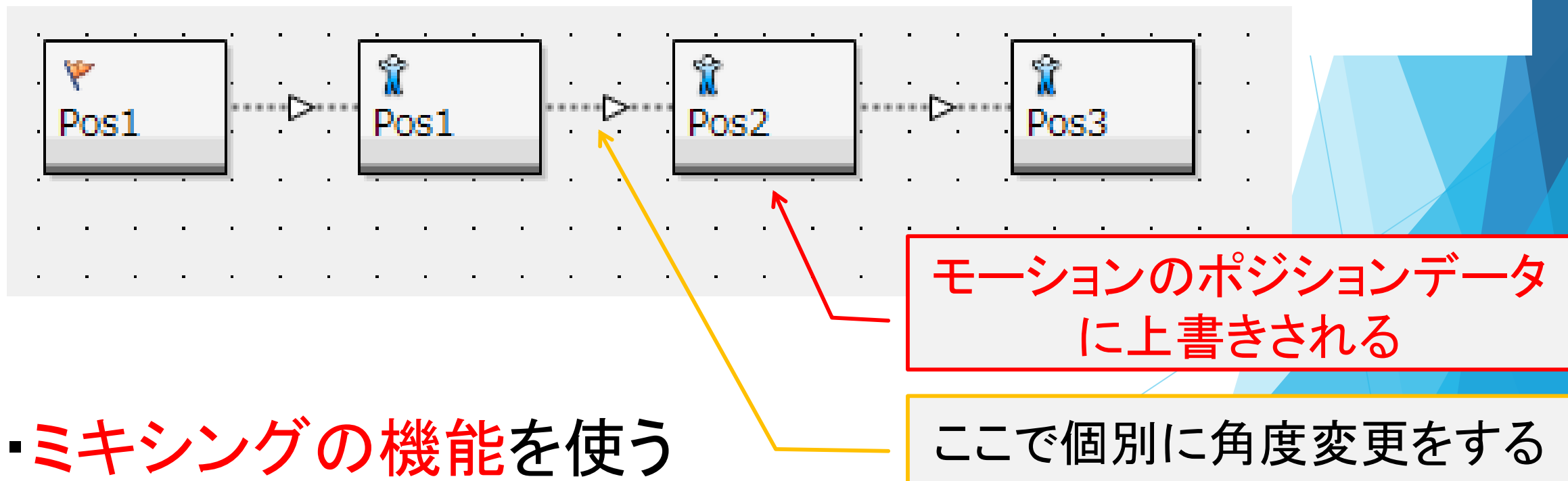
- モーションの再生方法

- RCB-4でモーションとは別に個別にサーボを動かす方法



# RCB-4でモーションとは別に個別にサーボを動かす方法

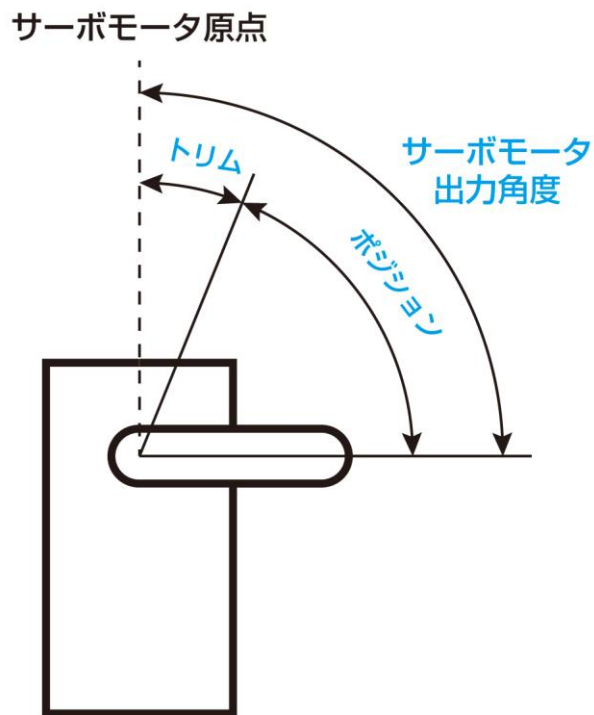
- ・モーション再生中はモーションのデータが再生されてしまうので個別にサーボモータを動かそうとすると動かした瞬間元に戻ってしまう



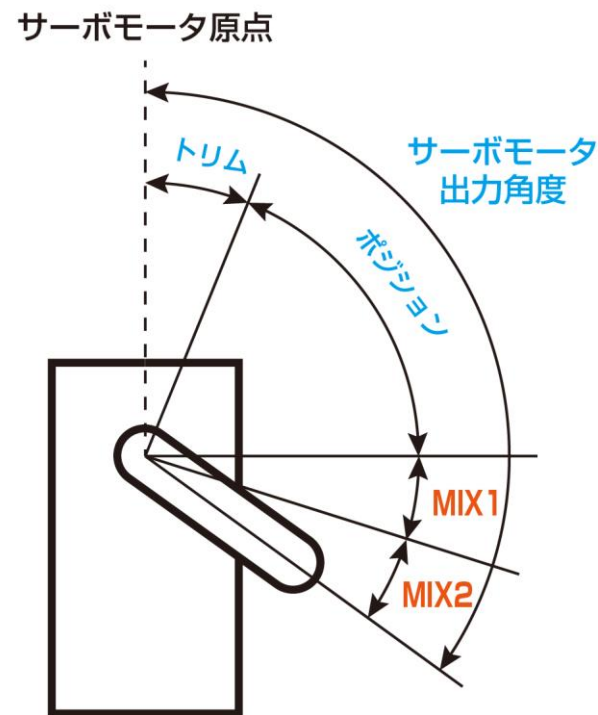
- ・**ミキシングの機能**を使う

# ミキシングについて

## ミキシング OFF



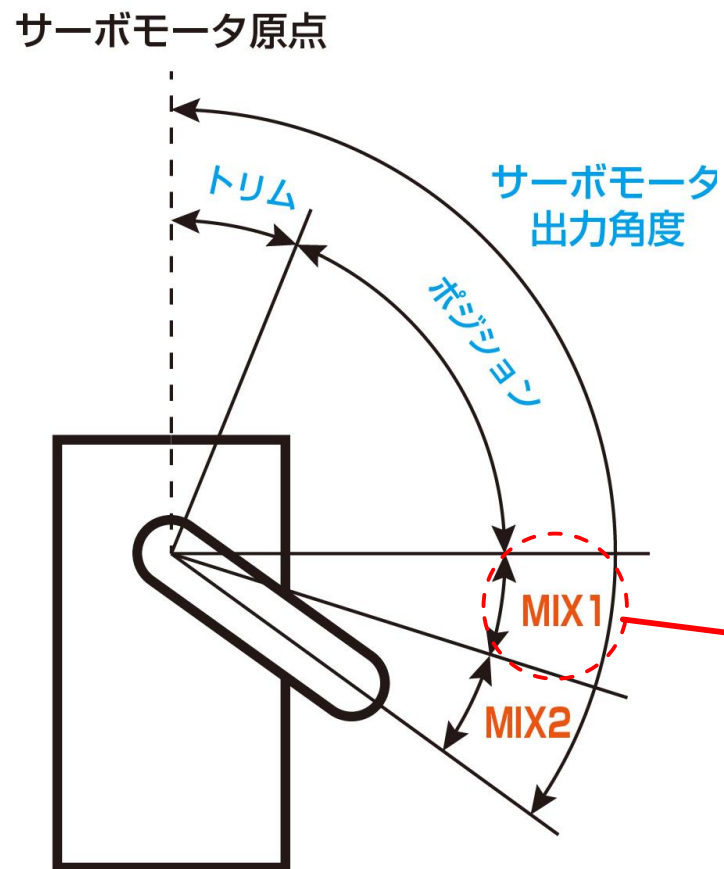
## ミキシング ON



ミキシング = MIXソースの値(AD値や変数の値) × MIXの倍率

動作角 = ポジション + ミキシング(MIXソースの値 × MIXの倍率)

# ジャイロセンサのミキシングについて

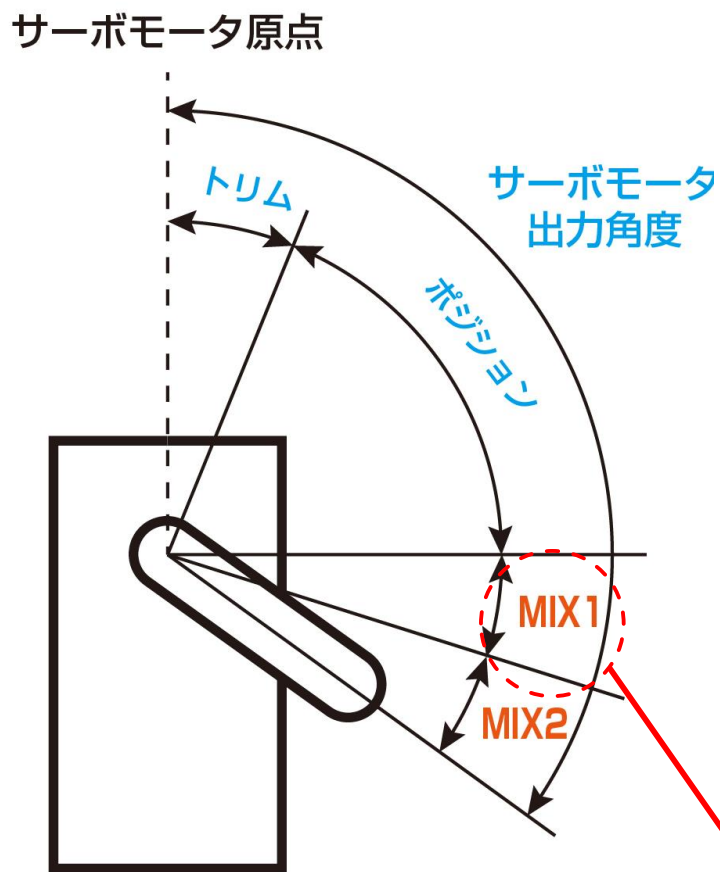


ジャイロセンサのミキシングは  
ジャイロセンサの値を倍率をかけて足す

・ADミキシング(ジャイロ等) => ユーザ変数のミキシング

# ユーザ変数のミキシング

- ユーザ変数:  
RAM上に20個自由に使える変数を用意してある



RAMアドレス	機能
...	...
0x0462	ユーザ変数1
...	...
0x0488	ユーザ変数20
...	...

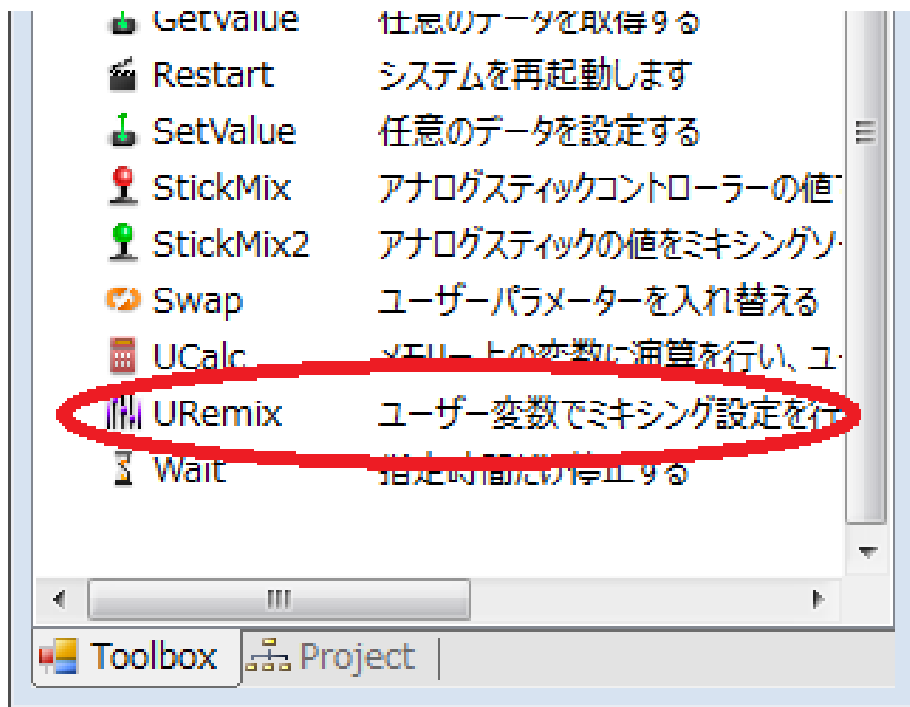
外部から書き換える

ユーザ変数を割り当て

# ユーザ変数のミキシングの設定と動かし方

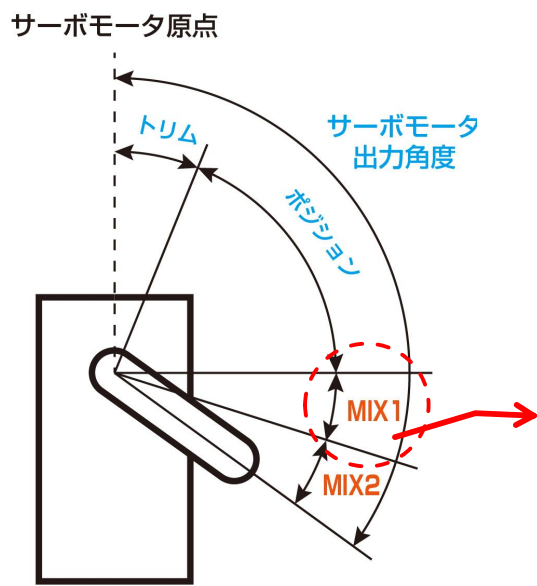
## ・URemixでユーザミキシング設定

プロジェクト設定では設定できないのでスタートアップモーションに入れる



# ユーザ変数のミキシングの設定と動かし方

- ・ユーザ変数を書き換えることでサーボに設定したポジション値が足される
  - ・モーション内で書き換える => SetValueダイアログ
  - ・外部から書き換える => `setUserParmeter()` <=ライブラリの関数



外部から書き換える

ユーザ変数を割り当て

- ・ユーザ変数値を書き換ええない限り保持されます

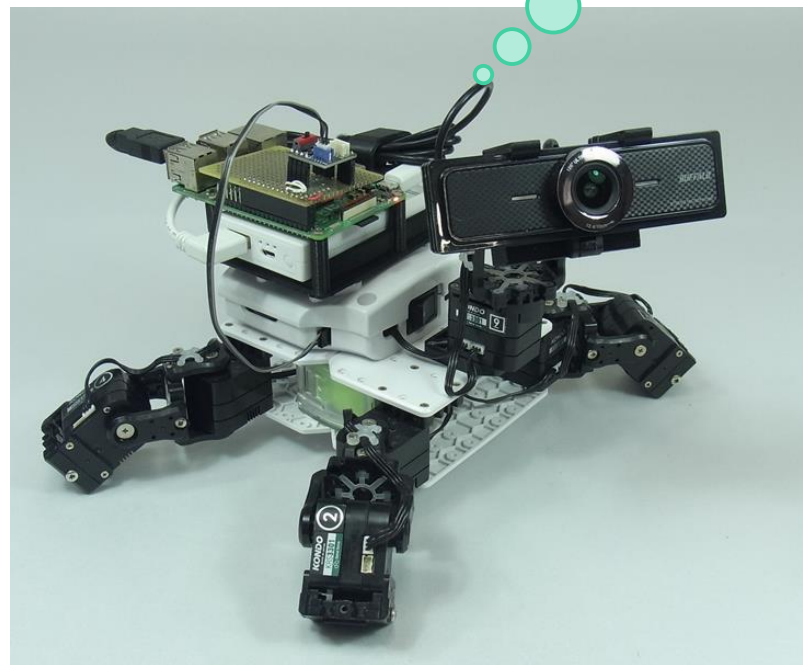
# 「RaspberryPiでRCB-4をコントロール」

1. はじめに
2. RCB-4およびHeartToHeart4について
3. Pythonライブラリについて
4. RCB-4を外部から動かす方法
5. **KXR-L4 カメ型ロボットに搭載した事例紹介**
6. 今後の展望

# KXR-L4 カメ型ロボットに搭載した事例紹介

- ハードウェア構成
- 行動のパターン
- 画像処理
- 動作方法

冒頭で紹介した  
ロボットです

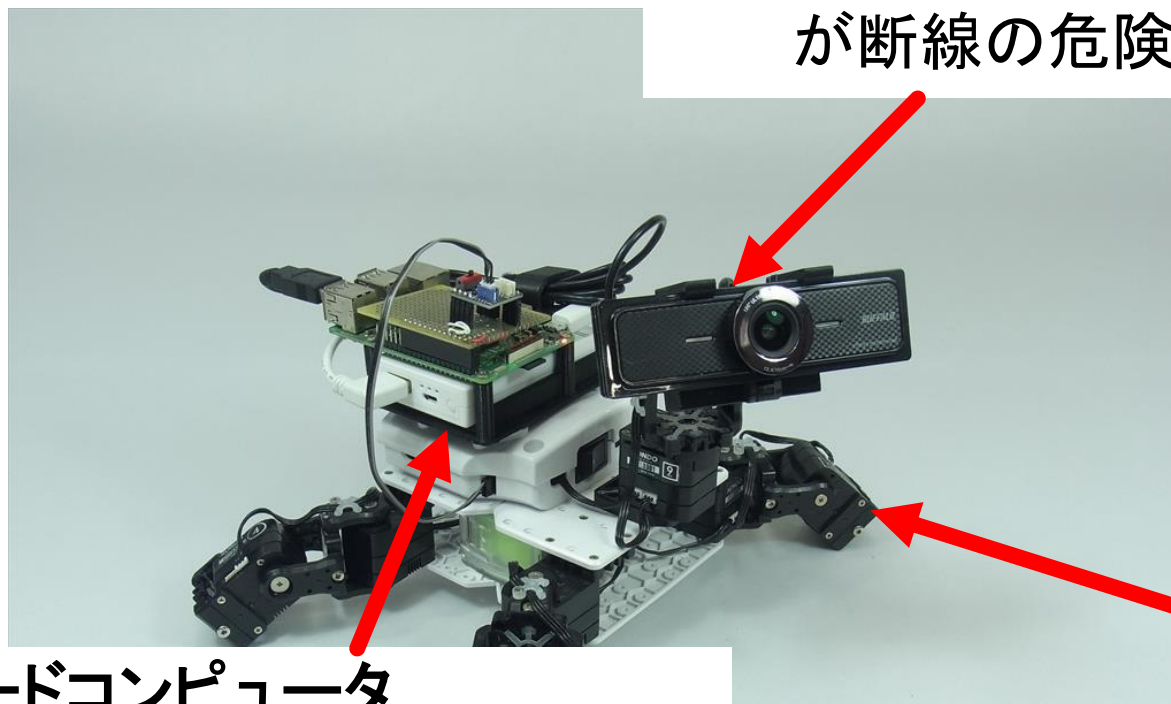




# ハードウェア構成

## ・カメラ

- ・USBカメラで広角のもの
- ・普通のラズパイカメラだとケーブルが断線の危険性がある



## ・シングルボードコンピュータ

- ・Raspberry Pi 3 Model B
- ・RCB-4変換基板
- ・モバイルバッテリー
  - ・意外とRaspberryPiの電源がシビア

## KXR-L4T

- ・カメラ用にピッチ軸を追加
- ・カメラのマウントやRaspberryPiのマウントは3Dプリンタを使用

# 全体構成



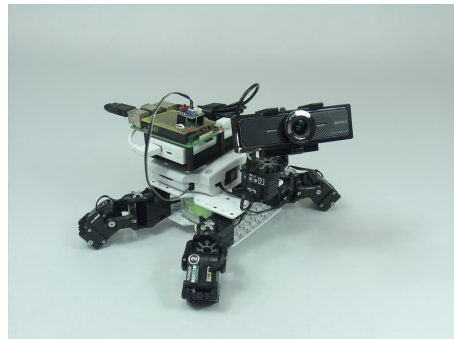
UART  
RaspberryPi 3 B+

RCB-4  
変換基板

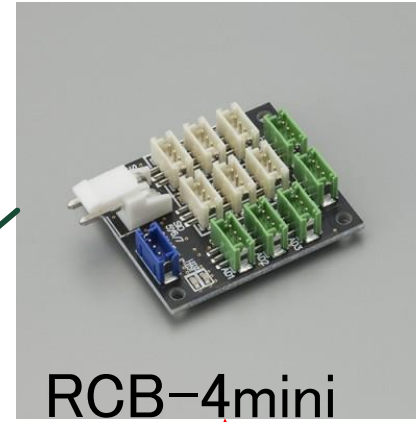
- ・モーション再生  
(SetButton経由)
- ・カメラの角度更新

モバイルバッテリー

画像処理  
行動パターン決定



KXR-L4T

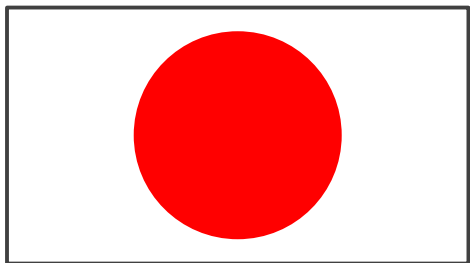


モーション: 前進、後退、左右旋回、  
バタバタ  
カメラ駆動用にユーザミキシングx2

# 行動パターンについて

赤いものを識別させ、それに応じてモーションを再生させます

①近い場合はバタバタするモーションを再生

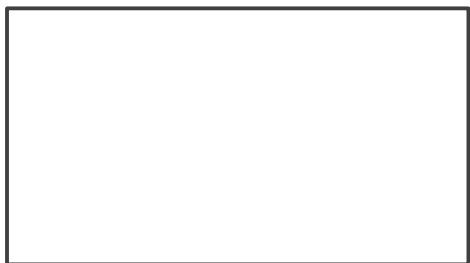


カメラ画像

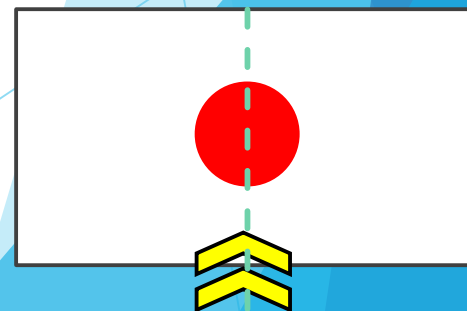
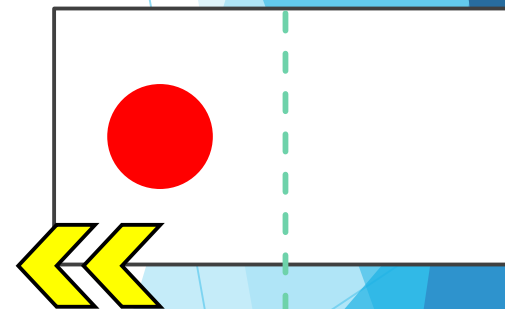
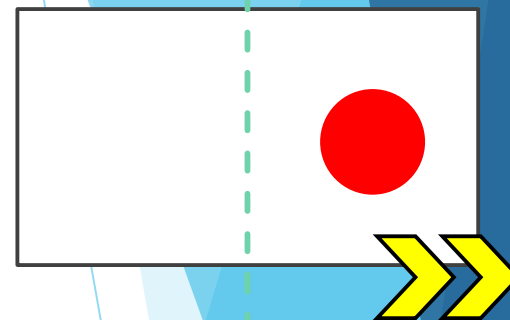
②遠い場合、

- ・右に物がある場合は右に旋回をする
- ・左に物がある場合は左に旋回をする
- ・正面に物がある場合は前進をする

③何も無い場合は移動しない(待機)



カメラ画像



# 画像処理について

## ● openCV(3.2)を利用

(特に深層学習とか使用せず、基本手法のみで実装)

## ● 画像の認識等の方法

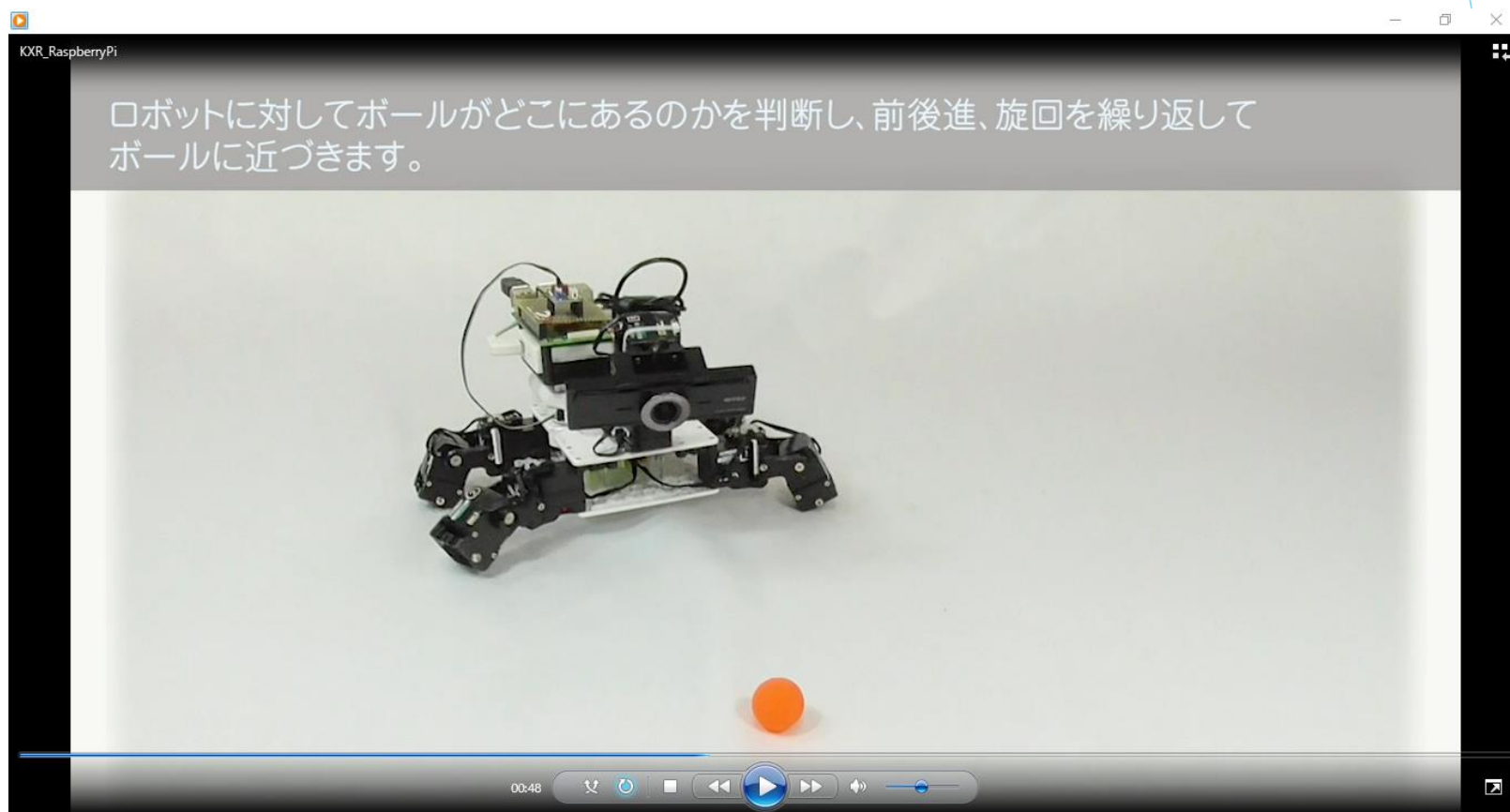
- ・画像を取得
- ・赤色の抽出
  - RGBからHSV色空間に変換する
- ・2値化
- ・ラベリング
  - 赤いものをうまくつなげて1つの物体にして番号をつける
- ・面積抽出、重心の抽出
  - 面積 ⇒ 物が近いかどうか判断する
  - 重心 ⇒ 首を振る角度を決める。行動をパターンを決定  
前との差分で追跡しているものの判断

画像の大きさと  
処理スピードのトレードオフ

# 動作方法

- **モーションを動作させる**
  - KRR等のボタンデータをトリガにしてモーションを再生する方法
- **首を振って対象に向ける**
  - ミキシング機能を使って角度を制御
    - ※実は絶対角しか制御できない
  - カメラ画像のデータはどれだけ移動するかしかわからない
    - ⇒サーボの現在角度を読み取って計算した角度を足す
- **コマンドを送る部分を別スレッド処理にする**
  - RCB-4ライブラリの関数では、コマンド送信後データ受信まで待っている
  - 前回と値が変わった時のみデータを送信(更新)する

# 動作検証



# 「RaspberryPiでRCB-4をコントロール」

1. はじめに
2. RCB-4およびHeartToHeart4について
3. Pythonライブラリについて
4. RCB-4を外部から動かす方法
5. KXR-L4 カメ型ロボットに搭載した事例紹介
6. **今後の展望**

# 外部のコンピュータと接続しやすくなった

- ・カメラ画像からの情報を使用することでロボットの行動に幅が広がった
- ・外部ネットワークから情報も受け取れる
- ・ハードウェア側をRCB-4に任せることで  
ソフトウェア、ハードウェアを別々に作ることができる
- ・RCB-4のコマンドからわかりやすい関数にしたことで使いやすくなった  
(でも、プログラムを組めない人にはまだ敷居が高い)



ぜひライブラリをご利用ください！

感想など随時募集中！  
メールでいただけるとありがたいです。

ご清聴ありがとうございました