Pythonで RCB-4をコントロール

近藤科学株式会社

はじめに

● RaspberryPiとRCB-4のデモ機の紹介

「RaspberryPiでRCB-4をコントロール」でできること



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。

ROBO-ONE参加機体を活かしてautoに参加できます!



RCB-4

KRSサーボモータを組み合わせてロボットの 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







これだけあるとできることに限界がある



RCB-4と組み合わせる

<u>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</u>



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 $(\text{Serial} \pm - \not\models)$ またはRCB-4変換基板 ロボット RCB-4 COM

RaspberryPiからモーションを再生





手順2-1 RaspberryPiの準備

【シリアル端子(UART)から通信する場合】

RaspberryPi3場合は、config.txtを書き換えてUARTを有効にし、Bluetooth を無効にする必要があります。

sudo nano /boot/config.txt

UART enable enable_uart=1

BT disable dtoverlay=pi3-miniuart-bt 詳しくは、『RCB-4とRaspberryPi の接続方法』をご参照ください。 https://kondorobot.com/faq/raspberrypi_wire_r cb4

手順2-2 RaspberryPiの準備

【USBアダプターから通信する場合】

FTDIのカーネルとDual USBアダプターHSを関連付ける必要があります。

\$ sudo su modprobe ftdi-sio echo 165C 0008 > /sys/bus/usb-serial/drivers/ftdi_sio/new_id exit

※書き込むにはroot権限(スーパーユーザー権限)で変更する必要があります。

詳しくは、『弊社USBアダプタをLinuxで使うには(2019版)』をご参照ください。 https://kondo-robot.com/faq/usb_adapter_for_linux_2019

手順3 RCB-4 Library for Pythonをダウンロード



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

『<u>RCB-4 Library for Python ver.1.0.0B</u>』

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

手順4 プログラムを実行する

・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。

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

外部からのモーション再生について

<u>MotionPlay</u>
 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)

Motion Table								
i o 🕨 🕅 🗉 🔤 🎦 🗙 🔚								
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	=					
ション者	6 号 7	ドタン番・	号					

HTH4モーションテーブル

ライブラリファイルの中身

Rcb4Lib_for_Python_V100B/ ト Rcb4Lib (PythonベースのRCB-4ライブラリ) ト sample (サンプルプログラム) ト Rcb4AckTest.py ト (sample ...etc) ト FunctionList V100B(See to index_html) (htmlベースの関数一覧) ト 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

RCB-4を外部から動かす方法

直接モーションを起動する方法(MotionPlay)

・ジャンプ命令だけで行うと危険なので現在再生している
 <u>モーションを中断してからジャンプ</u>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モーションが 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



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

RAM上に20個自由に使える変数を用意してある



ユーザ変数:

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



UserRemixDialog						
特殊マセミンが設定 (a	בד₩/D≾th	た1百日を再設定)				
SIO1 2 3 4	197011					
2前		MIX1 V-7	MIX1 倍率	MIX2-V-Z		
	0				1	
	1	OFF	1	OFF	1	
	2	OFF	1	OFF	1	_
□ 左肩 (3-)	3	OFF	1	OFF	1	
	4	OFF	1	OFF	1	
□ 左腿 (ヨー)	5	OFF	1	OFF	1	
□ 左腿 (ロール)	6	OFF	-5	OFF	1	
■ 左腿(ピッチ)	7	OFF	1	OFF	1	
□ □ 左膝	8	OFF	1	OFF	1	Ŧ
3105 6 7 8						
名前	ID	MIX1 ソース	MIX1 倍率	MIX2 ソース	MIX2 倍率	*
■ 腰 (∃−)	0	OFF	1	OFF	1	
□ 右肩 (ピッチ)	1	OFF	1	OFF	1	Ξ
🔲 右肩(ロール)	2	OFF	1	OFF	1	
■ 右肩 (3-)	3	OFF	1	OFF	1	
■ 右肘	4	OFF	1	OFF	1	
□ 右腿 (3-)	5	OFF	1	OFF	1	
	-				1.	



「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

<u>・カメラ</u> ・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

<u>・シングルボードコンピュータ</u>

Raspberry Pi 3 Model B
 <u>•RCB-4変換基板</u>
 <u>•モバイルバッテリ</u>
 ・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





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)