カム☆ロボ(ラズ☆ロボ)・カー紹介

(カムプログラムロボット工作セットに
汎用電動機制御基板でロボット・カー製作記事)

●目次

2017.10.8　　JH1CDV/開放電脳　松元博司

0.1 はじめに

‘楽しく、早く、安く’ 電子工作を提唱しています。
MFT2017で展示し好評を博した、ラズ☆ロボ・カーをカムプログラムロボット工作セットでバージョンアップしてみました、作り方を公開します。
まず、下のURLの映像を見てください。
https://www.facebook.com/466797943517881/videos/717124401818566/
https://youtu.be/_opalFqS50s
単体としては左右の距離センサと、左右のDCモータが、それぞれ独立に動いています。そのセンサ情報を、ラズパイ、NODE－RED、MQTTと経由して、相手のロボットに送ることで、この動きを実現しています、このような動きを実現でき、楽しんでいます。

こんなに楽しいロボットカーが、汎用電動機制御基板を使えば、簡単に作れます。
自分は、回路設計が専門なので、ソフト得意の人が、改良してくれて、情報をいただけると助かります。
このように、オープンハードウェアの考えに立ち、情報を公開しますので、みんなで、もっと楽しい電子工作ができると嬉しいです。

0.2 はじめに

単体としては左右の距離センサと、左右のDCモータが、それぞれ独立に動いています。そのセンサ情報を、ラズパイ、NODE－RED、MQTTと経由して、相手のロボットに送ることで、この動きを実現しています、このような動きを実現でき、楽しんでいます。

こんなに楽しいロボットカーが、汎用電動機制御基板を使えば、簡単に作れます。
自分は、回路設計が専門なので、ソフト得意の人が、改良してくれて、情報をいただけると助かります。
このように、オープンハードウェアの考えに立ち、情報を公開しますので、みんなで、もっと楽しい電子工作ができると嬉しいです。

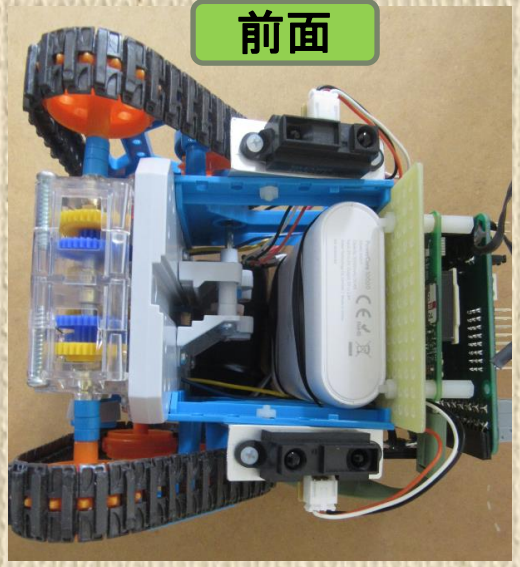
　ロボット・カーを動かすには、多くの分野の技術を知る必要があります、その技術、ノウハウを下記の順番で説明します。

１）　ハード：必要なハードの構成
２）　ソフト：使用しているデバイス用のpythonプログラム
３）　システム：ロボット・カーを動かすためのNODE－REDプログラム

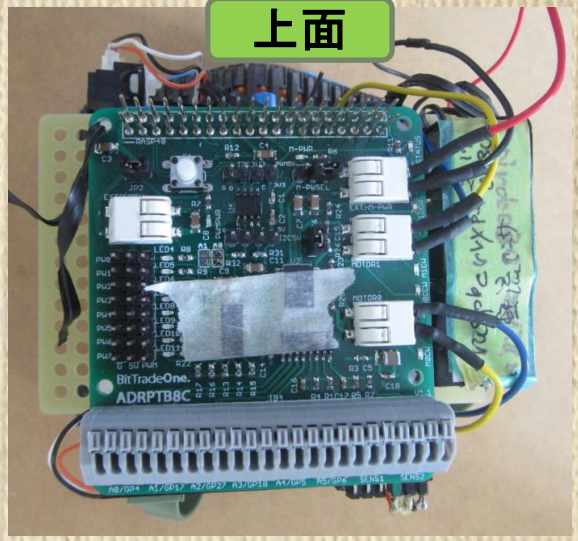
0.3 はじめに

完成した、カム☆ロボ(ラズ☆ロボ)・カーを見てください

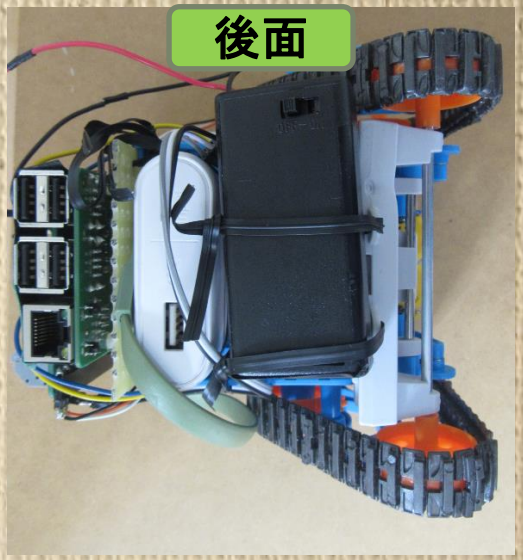
前面



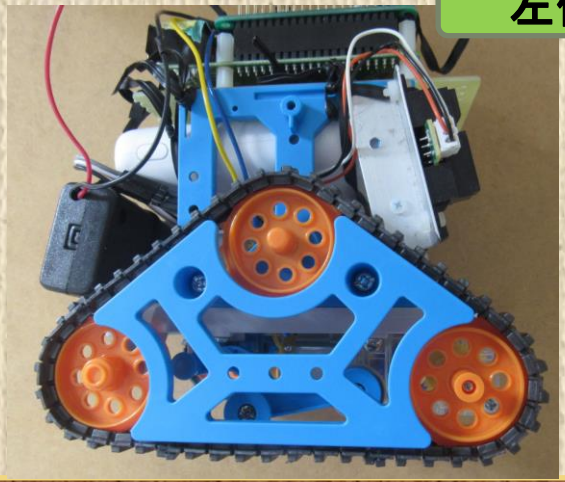
上面



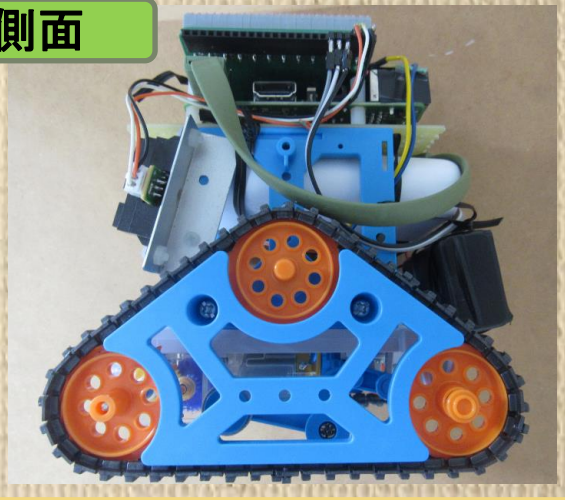
後面



左側面



右側面



1.1　ラズ☆ロボの設計図：ハード

ハード→ソフト→システムの順に説明します

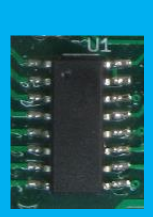
距離センサ
GP2Y0A21YK0F
10cm～80cm、5V

アナログ/GPIO入力
3.3V系：6入力
5V系：2入力

DCMotor出力
2出力

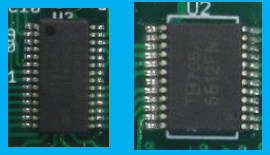
DCMotor

入力　センサ



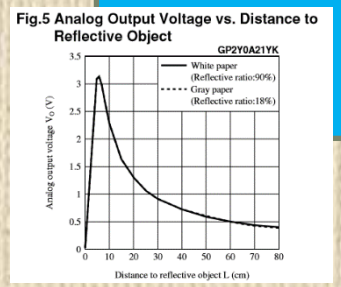
MCP3208　12bit
8ch　A／Dコンバータ　SPI

サンプリング速度：50ksps　入力範囲：0～3.3V



PCA9685
12bitPWM、16ch（内6ChをTB6612に使用）I2C

出力電流　1.2A　TB6612　Dual　DCモータ･ドライバ
正転/逆転/ショートブレーキ/ストップ制御機能付き
PWM出力：PCA9685により制御



出力　アクチュエータ

汎用電動機制御基板

1.2　汎用電動機制御基板で工夫した点

・バランスの取れた入出力対応

　アナログ入力と、DCモータ制御を一枚の基板で対応しているが見つからなかったので、この基板を作りました。

・去年のMFT2016で、ソフトの専門家から、はんだ付けしないでﾗｽﾞﾊﾟｲを使いたいという声を聴きました。

・信号の見える化

　以前の設計で、DCモータの回転の向きが分かりづらく苦労したので、モータの回転方向

　　時計回り：clock　wise（CW）

　　反時計回り：Counter　clock　wise（CCW）

　の信号に色を変えてLEDで表示し、実際にモータをつながなくてもデバッグできます。

・モータ電源を、外部だけでなく、内部の5V電源も切り替えられるようにしました。

・次は、結果的にそうなったという意見もありますが、ﾗｽﾞﾊﾟｲGPIO端子と、ADコンバータの信号線を同じ端子に接続しました、こうすることにより、ADコンバータの強力な入力保護回路が、ﾗｽﾞﾊﾟｲに対しても保護してくれます。

・arduino等、5V系の豊富なI2Cモジュールに対応できるようにI2C　5V系に対応

・PWMで3ピンのサーボモータ（例：SG90）を、直接接続できるようにしました。

1.3　必要な部品表

No.	品名	購入先	URL	参考単価	個数	参考価格
1	ﾗｽﾞﾊﾟｲ3	KSY	https://raspberry-pi.ksyic.com/	4,830	1	4,830
2	汎用電動機制御基板	ビット・トレード・ワン	http://btoshop.jp/2017/08/25/4562469771304/	3,980	1	3,980
3	カムプログラムロボット工作セット	tamiya	http://www.tamiya.com/japan/products/70227/index.html	3,456	1	3456
4	モバイルバッテリ	Anker	https://www.amazon.co.jp/Anker-PowerCore-%E3%83%A2%E3%83%90%E3%82%A4%E3%83%AB%E3%83%90%E3%83%83%E3%83%86%E3%83%AA%E3%83%BC-Android%E5%AF%BE%E5%BF%9C-A1263011/dp/B019GNUT0C/ref=zg_bs_2544551051_1?_encoding=UTF8&psc=1&refRID=EM9HW0DD4P29J6AYTNW3	2,399	1	2,399
5	マイクロUSBケーブル	100均		100	1	100
6	距離ｾﾝｻ シャープ測距モジュール　GP2Y0A21YK	秋月	http://akizukidenshi.com/catalog/g/gI-02551/	450	2	900
7	L字金具2枚（距離ｾﾝｻ取付用）	100均		100	1	100
8	ﾗｽﾞﾊﾟｲ実装ﾎﾞｰﾄﾞキット（予定） ﾎﾞｰﾄﾞ2枚、スペーサ、ビス、ナット　セット	ビット・トレード・ワン		500	1	500
					合計	16,265

1.4　組み立て・配線手順

下記の手順で組み立てます
１）　　カムプログラムロボット工作セットを組み立て
２）　　ラズパイ実装ボートを取り付け
３）　　ラズパイ実装ボードに左右の距離センサを取り付け
４）　　ラズパイ実装ボードにラズパイ３を取り付け
５）　　ラズパイ３に、汎用電動機制御基板を取り付け
６）　　ラズパイ実装ボードに、電池ボックス（単３*２）を取り付け
７）　　ラズパイ実装ボードに、モバイルバッテリを取り付け

次に配線をします
１）　　左右の距離センサと、汎用電動機制御基板をそれぞれ３本の信号線で接続
２）　　カムプログラムロボット工作セットについている左右のモータと、汎用電動機制御基板をそれぞれ２本の信号線で接続
３）　　電池ボックスと、汎用電動機制御基板を２本の信号線で接続
４）　　モバイルバッテリから、ラズパイ３に電源を接続（マイクロUSBケーブル使用）

ここまでで、ハードは完成です、部品がそろえば、数時間でできると思います。

2.1 ソフト:デバイス用のpythonプログラム

必要なソフトの構成を、下記のように考えてみます。
１）**入力処理**:python　センサからの情報を論理処理部に伝える（**監視**）
２）**論理処理**:NODE-RED　動かしたい内容を論理的にプログラム
３）**出力処理**:python　論理処理部からの情報で、アクチュエータ（DCモータ）を**制御**

従来のソフトの作り方は、Python言語で、入力、論理、出力処理を一つのプログラムで作るのが一般的であると思います、自分も今まではそういう作り方でした、ただこのような作り方だと、**ネットワーク処理、並列処理が難しく**なります。

今回、ネットワーク処理、並列処理をできるだけ容易に行うため、論理処理部をNODE－REDにし、入力、出力処理部を別々に、pythonプログラムで作ることにしました。

2.2 ソフト:デバイス用のpythonプログラム

入力処理部のソフト仕様
・ADコンバータから、値(ch0からch6)を読む
　左右の距離センサは、ch6,7に接続しているので、基準距離(例:30cm)を0とする
　基準距離より近ければ　　-0.5から0の値
　基準距離より遠ければ　　0から+0.5の値
・その値を、JSON形式で、複数の値を同時にMQTTプロトコルで、NODE－REDに送る
　JSON形式にしたのは、node-redで扱いやすく、複数の値を同時に送れ、値の種類を自由に定義できるからです。

　MQTTにしたのは、IoT用のプロトコルで、NODE－REDで標準でノードがあり、扱いやすいからです。

出力処理部のソフト仕様
・node－redからMQTTプロトコルで、複数の値を同時にJSON形式で受ける。
・DCモータを動かすための、値(ch6からch7)を読む
　読んだ値(-0.5~0~+0.5)により、DCモータを前進、後進させる

2.3 ラズ☆ロボの設計図：ソフト（デバイスドライバ相当）

オープンソースのPythonライブラリを少し改造しています

①センサ入力用Pythonソフト
センサの情報をNODE-REDに伝える
入力：12bitアナログ入力
出力：MQTT、ISON形式

②DCMotor出力用Pythonソフト
③サーボ出力用Pythonソフト
NODE-REDの情報でモータを制御する
入力：MQTT、ISON形式
出力：DCMotor　2ch、orサーボ8ch

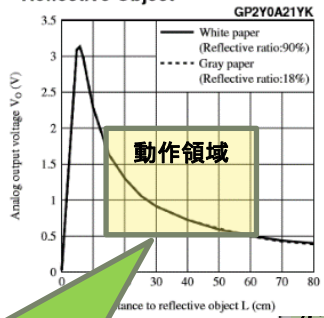
MCP3208-MQTTout-pub-...ut-pub-DCmotor.py (3.4.2)

File Edit Format Run Options Windows Help

```
# -*- coding: utf-8 -*-
# v1.0 2017/7/15 jh1cdv
# v1.1 2017/7/28 jh1cdv
# Input:MCP3208からSPI通信で12ビットのデジタル値を取得。0から7の8チャンネル使
# Output: -0.5 --- 0 --- +0.5
#       mqtt topic=/In-ch0,1
#


import RPi.GPIO as GPIO
from time import sleep

import context  # Ensures paho is in PYTHONPATH
import paho.mqtt.client as mqtt

import json


def on_connect(mqttc, obj, flags, rc):
    print("rc: " + str(rc))


def on_message(mqttc, obj, msg):
    print(msg.topic + " " + str(msg.qos) + " " + str(msg
```

Ln: 1 Col: (

Fig.5 Analog Output Voltage vs. Distance to Reflective Object



赤外線距離センサの特性です、あまり近いと予期せぬ動作をするかもしれません！！！

Thonny - /home/pi/raz-robo/pwm/pwm-pca9685-mqttin-sub.py @ 249 : 1

File Edit View Run Tools Help

pwm-pca9685-mqttin-sub.py

```
# Simple demo of of the PCA9685 PWM servo/LED controller library.
# This will move channel 0 from min to max position repeatedly.
# Author: Tony DiCola
# License: Public Domain
#
# name:pirobo
# Author:jh1cdv
# date:2017/07/02
# input sensor1,2
# process input to output -0.5 +0.5
# output: rotationserbo

import RPi.GPIO as GPIO
from time import sleep
```

M出力：PCA9685
により制御

下記3つのライブラリをインストールしています

- Adafruit_Python_PCA9685
- Adafruit-Motor-HAT-Python-Library
- paho.mqtt.python

2.4 ソフト:デバイス用のpythonプログラム

まず、ハードがソフトから見れるかどうか確認するため、sshターミナルソフト(例:Mobaxterm)で、下記コマンドを実行し、I2Cデバイスを確認しておきます。

```
pi@raspberrypi:~ $ sudo i2cdetect -y 1
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
60: 60 -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: 70 -- -- -- -- -- -- --
pi@raspberrypi:~ $
```

Pythonプログラムのファイルのリストを下記に示します

No.	名称	内容
1	MCP3208-MQTTout-pub-DCmotor.py	アナログ入力をMQTTでNODE-REDに渡す
2	pwm-pca9685-mqttin-sub-dcmotor.py	NODE－REDから、MQTTで情報を受け取り、DCモータを制御する

適切な、フォルダに、上記2ファイルをコピーしておく
自分は、WINSCPを使いコピーしています。

2.5 ソフト:デバイス用のpythonプログラム

Pythonプログラムには、下記ライブラリ等のインストールが必要です

No.	名称	内容	URL
1	Adafruit_Python_PCA9685	PWM出力のライブラリ	https://github.com/adafruit/Adafruit_Python_PCA9685
2	Adafruit-Motor-HAT-Python-Library	DCモータ出力のライブラリ	https://github.com/adafruit/Adafruit-Motor-HAT-Python-Library
3	paho.mqtt.python	MQTTのpythonライブラリ	https://github.com/eclipse/paho.mqtt.python
4	mosquitto	MQTTブローカ	http://mosquitto.org/2013/01/mosquitto-debian-repository/

ライブラリのインストールは、下記のようにしました。

1）　まずGITのサイトにブラウザで見てみます。

2）　サイトの説明にある下記コマンドをsshで実行します

2.6 ソフト:デバイス用のpythonプログラム

実行結果を張っておきます。
sudo pip install adafruit-pca9685
pi@raspberrypi:~ $ sudo pip install adafruit-pca9685
Installing collected packages: adafruit-pca9685
Successfully installed adafruit-pca9685-1.0.1
pi@raspberrypi:~ $

ほかのライブラリも同様にインストールします。
MQTTブローカは、下記のようにinstallします。
Sudo　apt-get install mosquitto

次に、pythonのプログラムを実行してみます。
pi@raspberrypi:~ $ cd raz-robo/spi-mcp3208-ad-conv/
pi@raspberrypi:~/raz-robo/spi-mcp3208-ad-conv $ sudo python3 MCP3208-MQTTout-pub-DCmotor.py
print文を入れてありますので、適切な値が表示されれば成功です。

2.7 ソフト:デバイス用のpythonプログラム

苦労した点

・出力用pythonプログラムで、jsonデータを読み込むとき、node－redで想定外の文字が余計に付加され、うまく読めませんでした、pythonの文字列置換機能で、余計な文字を削除して、何とか読めました。

工夫した点

・自立制御の一見複雑そうな動きを、左右全く独立で動かすことにより、簡単に動かすことができました、これは、以前　子供の科学という雑誌で、左右の太陽電池と左右のモータをクロスでつないで、光の方向に走る模型のカーというアイデアを見ました、今回そのアイデアを応用しています。

自分でも、こんなに容易に、左右・前後に人？（白い大きなもの？）についていくロボットカーが動いて、感激しています。

3.1 システム：NODE－REDプログラム

NODE－REDは、最初IBMが開発し、現在はオープンソースになっている、WEB用のJAVASCRIPT言語による、開発環境です。

ただ、今回の例では、なんとJAVASCRIPTのプログラムを１行も書かなくて、機能を実現できてしまいました。

ﾗｽﾞﾊﾟｲに標準でついています、少し頑張れば、Windows10でも動かせます、自分が、どこのハードの開発環境（NODE－RED）で、動かしているのが分からなくなり、それぞれのハードの得意、性能を考え、ハード分散の開発が容易にできる優れものです、是非動かしてもらえると楽しいですよ？

ラズ☆ロボの設計図：システム（システムソフト相当）

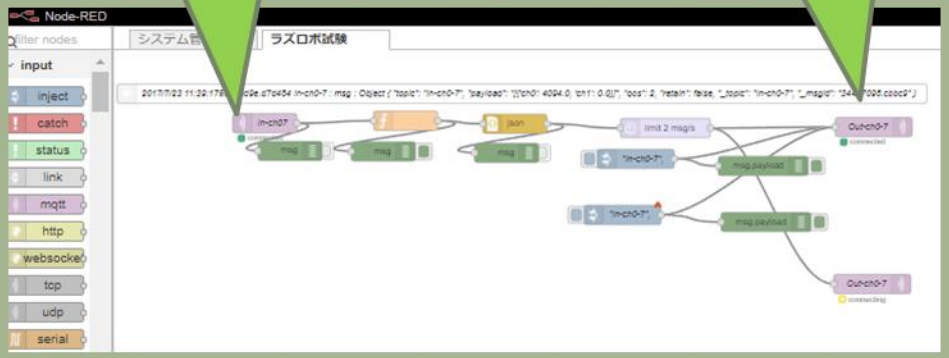
NODE－REDで、情報の行き先を指定しています。

①センサ
②Pythonソフト
③Node－Red
の順に情報が流れる

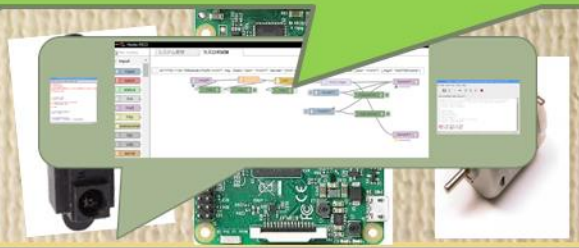
センサの情報を取得
センサ入力用Pythonソフト
からのMQTT入力
ISON形式

モータを制御
DCMotor出力用Pythonソ
フトへのMQTT出力
ISON形式

①Node－Red
②Pythonソフト
③センサ
の順に情報が流れる

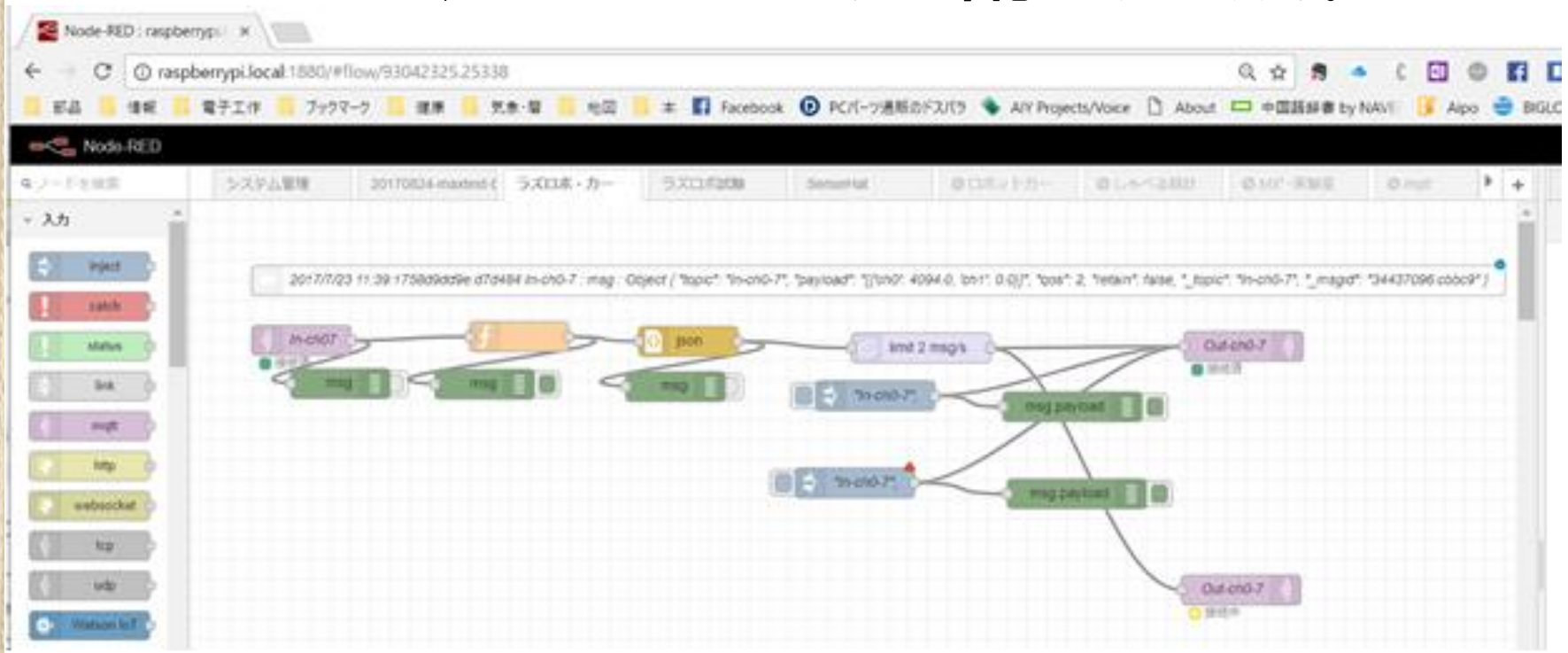


Node－Redからの情報はMQTTにより、
ローカルも、リモートも同等に扱われる
複数のラズロボが、対等に情報交換できるので、シンクロできる



3.3 システム:NODE－REDプログラム

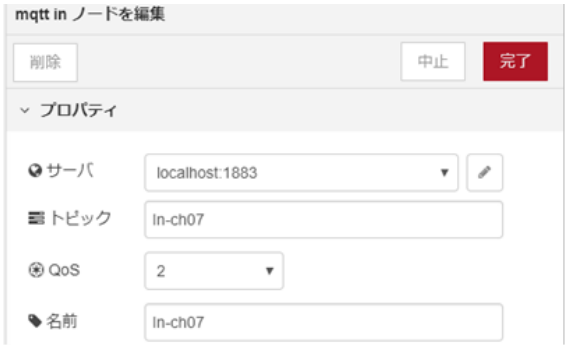
ﾗｽﾞﾊﾟｲのnode－redを起動します。
ブラウザから　http://raspberrypi.local:1880 （raspberrypi.localは、ﾗｽﾞﾊﾟｲの名前設定で設定した名前になります、デフォルトだと、raspberrypi.localです。）node－redに接続します。
node－redのソースを、node－redにコピーすると下記のようになります。



設定するのは、５種のノードです、ほかのノードはデバッグ用ですので、動かすだけなら必要ありません。

3.4 システム：NODE－REDプログラム

設定するのは、下記5種のノードです。



node－redの設定が出来たら、pythonのプログラムを起動します

sudo python3 MCP3208-MQTTout-pub-DCmotor.py &

sudo python3 pwm-pca9685-mqttin-sub-dcmotor.py

node－redのデバッグwindowに、センサの値が表示されれば成功です。

3.5 システム：NODE－REDプログラム

苦労した点

・**赤外線距離センサの動作環境に苦労しました。**

当初、自分が前を歩くと、ついてくるロボットカーを考えていたのですが、うまく扱えばついてくるようなときもありますが、安定して動かすため（赤外線の反射の影響と推定）には、実験の結果下記の環境でした。

・下（床、机上）を黒い布で覆う
・検出する反射物を、A3白程度の大きさにする

・**ラズパイの処理能力で少し苦労しました。**

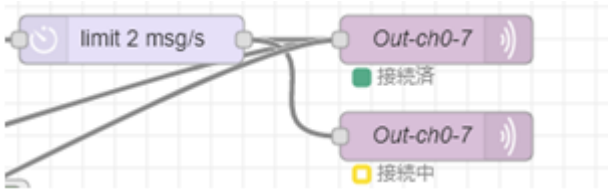
当初、データの見える化で、node-redのdashboardを使って、webブラウザでデータをみえるようにしていたのですが、毎秒１回以上更新すると、負荷が重くなるせいか、うまく動かなくなり、**いまはwebブラウザで見ないように設定**しています。

又、pythonプログラムでは、0.1秒ごとにデータを送り、delayノードで　**処理量を毎秒2メッセージに調整**しています。

3.6 システム：NODE－REDプログラム

ここまでが、単体動作の説明です。
デモ映像では、**2台のﾗｽﾞﾊﾟｲのシンクロ動作**ができています。
どうやってシンクロ動作ができるのか説明します。

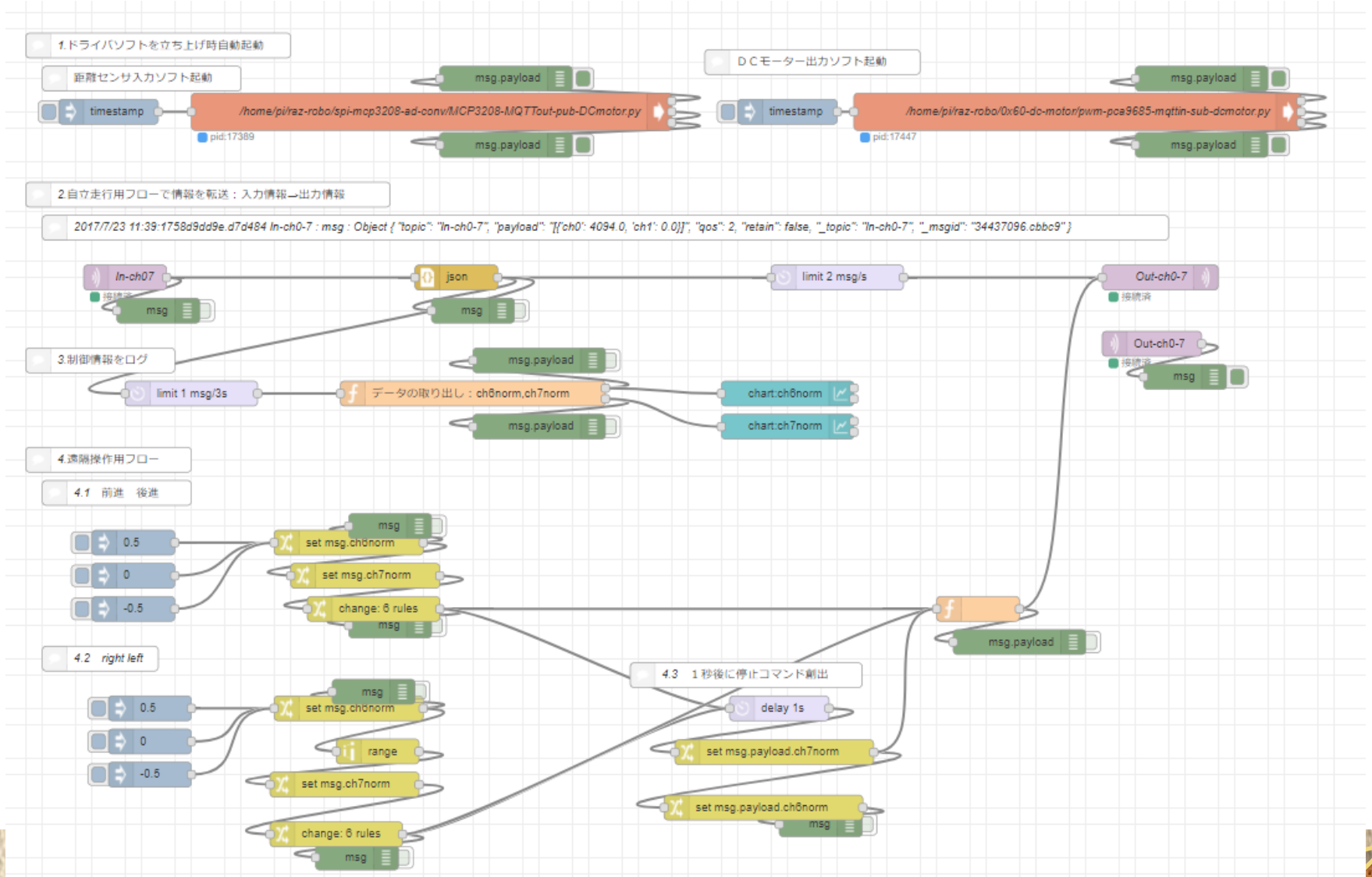
追加するのは、たった一つのノードだけです。



中身もサーバのアドレスが違うだけです。
新たにつなぐ**相手側のﾗｽﾞﾊﾟｲは、自分に接続されているセンサと同じように、別のﾗｽﾞﾊﾟｲの信号を読み込んで、動作する**ので、2台のﾗｽﾞﾊﾟｲが、あたかもシンクロしているように動くように見えるという考えです。
実際やってみると、拍子抜けするぐらいに簡単に動いてしまい、感激しました。

4.1　応用編：NODE－REDプログラム

MFT2017以降の新規機能（ログ、遠隔操作）について説明します。
新規機能含めた全体のフローを示します。



5.1　情報公開

pythonプログラムと、NODE-REDプログラムをダウンロードできるようにしますので
動かしてもらえると嬉しいです。
当面は、下記からダウンロード願います。
https://sites.google.com/site/kaihoudennou/home/20170805-mft2017-20170803.zip?attredirects=0&d=1

BTOさんから、更新情報含めて公開予定です。

これから作りたいもの
・DCモータとサーボモータを同時に制御して、リモコンカメラ付き　ラズ☆ロボ・カー
・センサを増やして、机から落ちない　ラズ☆ロボ・カー
・道路をトレースして走る　ラズ☆ロボ・カー
・自分の代わりに、色々なところに行ってくれる　ラズ☆ロボ・カー
・自分の代わりに、誰かのところに行って、会話できる　ラズ☆ロボ・カー
夢は広がります、みんなで楽しみましょう！！！

5.2 まとめ

・早く
　サーバ：NODE-RED、MQTT、JSON　素早く開発、ライブラリが豊富
・安く
　サーバ部品：ラズパイ　ZERO、２，３
・楽しく
　node－red+MQTTで動かしているので、電子ブロック感覚で、
　　機能を変更して遊べます。

電子工作の世界では、ハードも、ソフトも　多種多様になり、どんどん良い製品が出てきています、しかも、最近は、
　オープン・ソース・ソフトウェア、オープン・ソース・ハードウェア
と呼ばれ、無料で使わせてもらえる恵まれた環境になっていると思います、自分はハードが専門なのに、こんなソフトまで使わせてもらって、楽しんでいます、皆さん、特にソフトの専門家の方に、色々作ってもらい、情報を公開してもらえると嬉しいです。