

# 知的ロボットインターフェース 2班

工藤 大山 土田 白坂 石井

## ロボットの名称

忠犬ハチ公

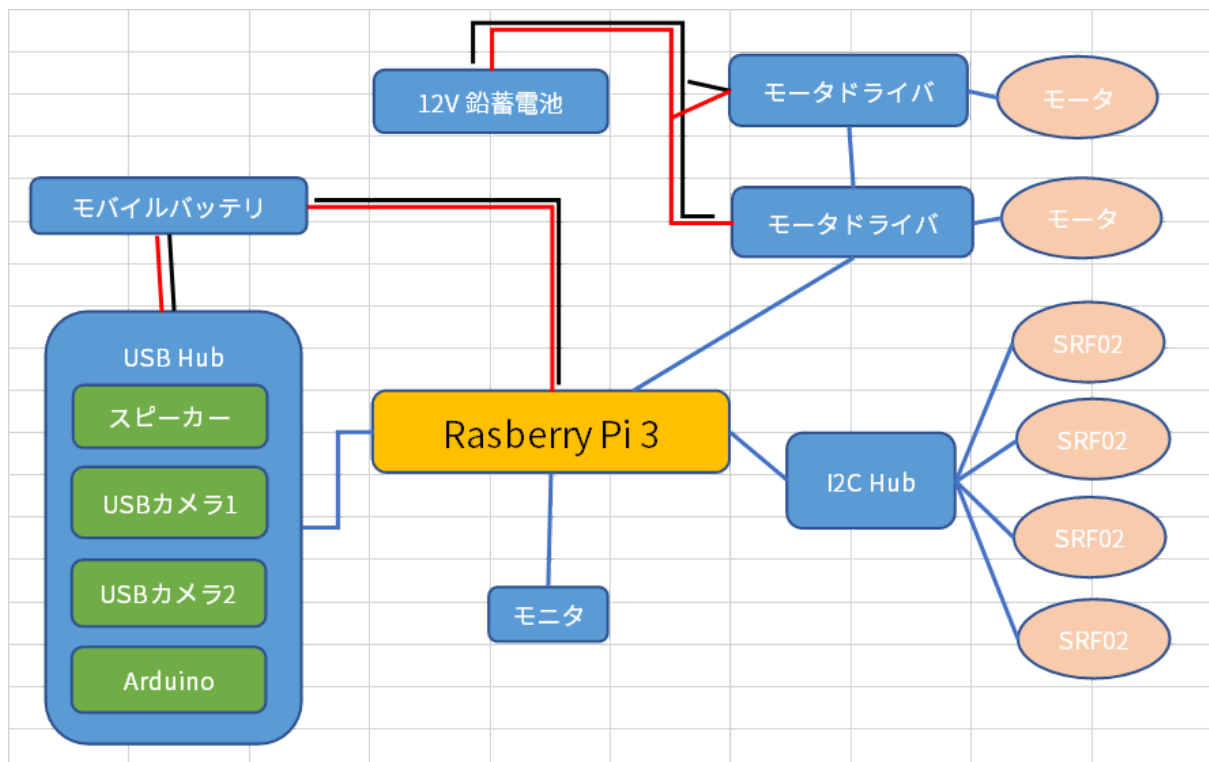
写真、動画: <https://photos.app.goo.gl/hJGTFqUHYKH6EjwG7>

## ロボットのコンセプト

まるで自分の犬のようにユーザーと遊ぶことができる犬ロボット。人の顔を見つけると近づいてくるとても人懐っこい性格である。ボール遊びをしたり、はたまたジャンケンをすることができるというとてもかしこい犬で、主人を楽しませることも忘れない。親しい人の顔には特別に笑顔でワンと吠えることもある。その賢さと愛らしさから、人は彼を忠犬ハチ公と呼ぶようになった。

## ロボットのハードウェア構成

基本的にはシンプルな構成となっている。



12V 鉛蓄電池からステッピングモータ 2つ分の給電をし、それらはラズパイからSPI通信で接続したモータドライバによって制御している。

また、測距センサ(SRF02)はラズパイからI2C通信し、接続している。  
モニタとはラズパイのHDMI端子からHDMIケーブルを用いて接続している。  
ラズパイのUSB端子にUSB Hub を接続し、Hubからはスピーカー、USBカメラ 2台、  
Arduinoに接続している。  
ラズパイそのものの給電はモバイルバッテリーから行っており、USB Hubの給電もモバイル  
バッテリーより行っている。

#### 特徴

- ・タイヤの回転中心と機体の底面中心がほぼ一致しており、回転による位置のずれがない
- ・ロボット全体が290\*300\*340 (高さ、横幅、奥行) と比較的コンパクトにまとまっている

## ロボットのソフトウェア構成

ディレクトリ構成は以下ようになった。尚、ここに示したのは主要なもので、拡張子がないものはディレクトリである。

```
/home/ri/Intel_robot
├── app
│   ├── 103_emmergencystop.py
│   ├── 201_main.py
│   └── steppingmotor
│       ├── motor.c
│       └── srf02.c
├── data
│   ├── haarcascades
│   └── wav_file
```

#### <ステッピングモーター>

基本的にはwiringPi.hを用いてステッピングモーターに指令を出している。以下の表1のようなコマンドを持っている。

表1 ステッピングモーターのコマンド集

命令	説明
p 回転数	<回転数>で指定された回転数を両輪の回転数とする
r 回転数	<回転数>で指定された回転数で回転する。負の数を与えると逆回転になる
s	一時停止する
e	停止し、プロセスを終了する

s命令やp命令で回転数を変える時、速度を徐々に変えるように設定されている。具体的には回転数を100ずつ徐々に変えるようになっている。また、故障を防ぐため最大回転数を指定しており、5000回転より上の回転数にはならないようになっている。

ポート番号50001でサーバーを立てており、char型の配列bufを受け付けている。このbufに入ったコマンドで先ほどのモーターが動く仕組みになっている。今回のロボットでは、pythonで書かれたカメラと測距センサーの値を加味したプログラムからコマンドを受け付けている。

---

<201\_main.py>

このプログラムによってカメラ画像の取得とロボットの行動を決定し、ステッピングモーターに指令を送っている。カメラ画像から人物検出に関しては、opencvのcascadeを使用している。(使用したcascade:

<https://github.com/opencv/opencv/tree/master/data/haarcascades> の

haarcascade\_frontalface\_default.xml) 取得したbounding boxにおける左上のx、y座標とheight、widthから中心のx座標とbounding boxの大きさを計算し、それを元に行動を決定する。

初期状態である人が検知されていない状態ではロボットは停止する。人が検出できた場合、以下のようなフローチャートで行動が決定される。まず、bounding boxの大きさが閾値以上であれば顔が近くにあると判断、つまり人に近づけたと判断し停止する。それ以外の場合、画面の横幅の4分の1から4分の3までに顔の中心があれば前進するが、左端や右端に中心があった場合、方向に応じて左折や右折を行う。

また、当初は測距センサーの値を用いて障害物へ対応しようとしたが、測距センサーの精度が悪かった為使用せず、代わりに手で急停止を行うスクリプト:

103\_emmergencystop.pyを用意した。

---

## <測距センサー>

まず、I2Cデータバスを4つのセンサに対してオープンする。なお、本事件ではSRF02をセンサとして使用した。各センサはコマンドレジスタ0にReal Ranging Mode - Result in centimetersを意味する0x51の信号を送ることで測距を開始する。このセンサは音波を利用して距離を測定しているので、音波の往復のために最低でも66000μsec待つ必要がある。そしてコマンドレジスタ2に測距離データの上位バイト、コマンドレジスタ3に測距離データの下位バイトをそれぞれリクエストし、それぞれのデータを1つにまとめることで距離データを獲得する。

しかし、このセンサはセンサ自体を動かした瞬間や、センサの前を物体が横切った瞬間などに上下限を超える不適切な値を提示してしまうので、このバグを取り除くための仕組みを施した。

まず、それぞれのセンサに対し連続して送られてくる距離データを格納しておくための配列を用意した。この配列のサイズをNとし、N個の距離データを格納するたびにそれらN個のデータの中から最も信頼できる距離データとして中央値となるデータを採用し、配列を初期化した。中央値を採用することによって、不適切な値を取り除くことが可能となった。Nを大きくしすぎると、距離データの獲得に時間がかかりすぎるといった問題が発生するため、今回はN=4で実装を行った。

---

### <プロセス間通信>

モーターを制御するプログラム（motor.c）と人を認識し、追いかけるプログラム（201\_main.py）の連携は、ソケットによる通信を用いて行った。まず、201\_main.pyにおいて人の顔を認識したら、ソケットを作成し、モーターへの指令を50001番ポートへと送る。motor.cにおいては、50001番ポートで201\_main.pyからの指令を待ち受け、指令が送られてきたらそれに合わせてモーターのスピードや回転の向きを変化させる。なお、トランスポート層のプロトコルはTCPを用いた。

## ロボットの動作の特徴

### <ロボットの特徴>

- ・自動で人を追いかけることができる
- ・どのような行動をするか音声で確認することができる。
- ・滑らかにスタート、ストップができる。
- ・緊急停止ボタンを押さなくてもプログラムで停止可能。
- ・行動を画像で表現しているため、将来的には行動ベースでロボットの感情の表現が可能。
- ・人を顔で検出しているため、追いかけるには顔を近づけることが必要。
- ・測距センサーを使用していないため、障害物には弱い。

## 各人のアプリケーション

### 工藤 ボール遊び

テニスボールを追いかけるアプリを作った。以下のようなアルゴリズムで動いている。

- ①1つのカメラを用いてRGB値からテニスボールを探し、座標・輪郭を割り出す(図1)
- ②その座標が半分より右側であれば右側に回転し、左側であれば逆回転をし、調整をする。(図2)
- ③座標が真ん中になったら直進する
- ④輪郭が大きくなったところで止まる。

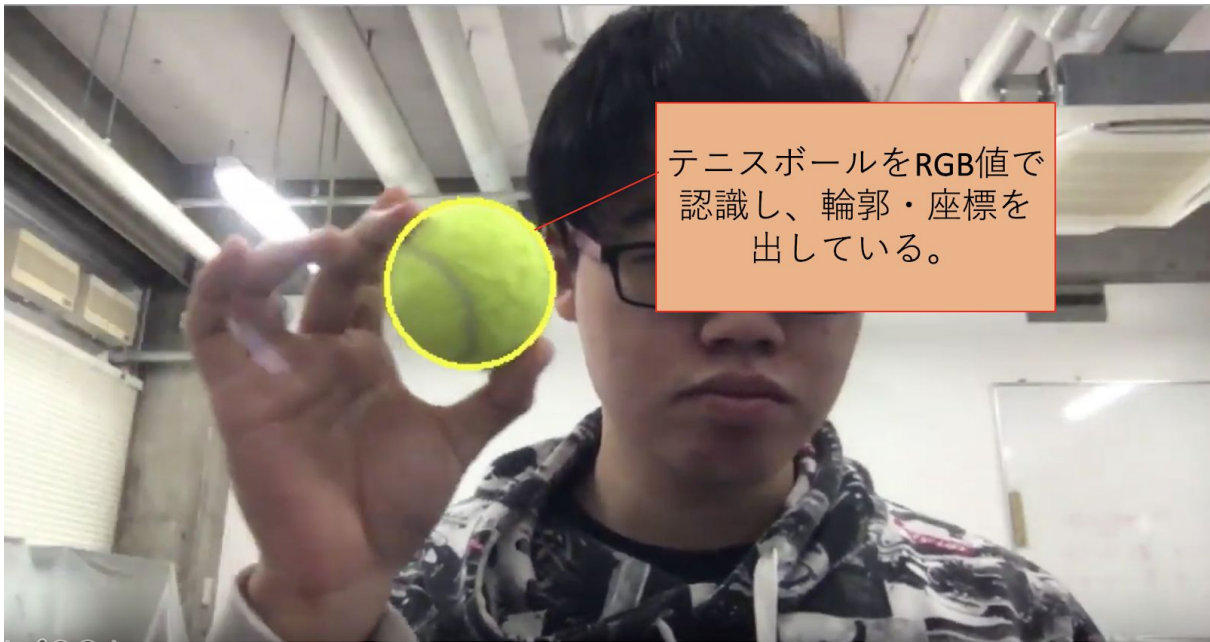


図1 RGB値から輪郭・座標を割り出している様子

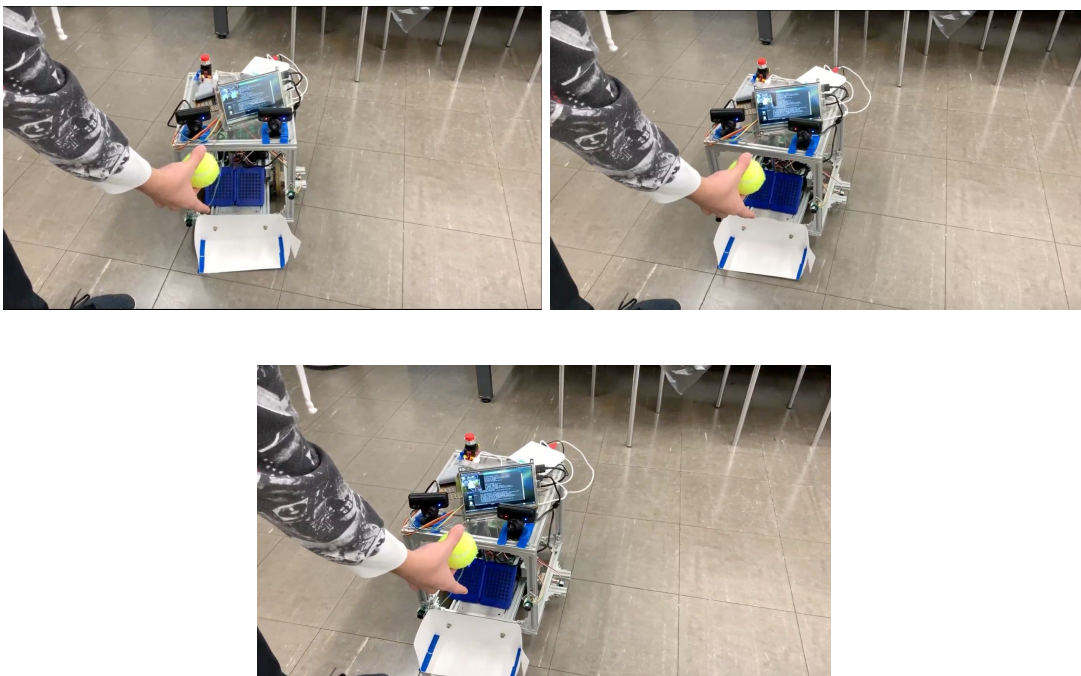


図2 テニスボールの位置によって回転し、直進している様子

## 工夫した点

実際にはテニスボールの認識以外の緑色のノイズにも反応してしまうため、複数の検出によってのみ反応させるという閾値をもうけた。また、モーターとのプロセス間通信で遅延が発生しており、物体認識精度が落ちるという現象があったため、最低限のプロセス間通信の

みをするように命令数を減らすようなアルゴリズムにした。他にも音声認識を用いたプログラムを作ろうと試みたが、音声辞書を作るところで時間がかかってしまい、作成には至らなかった。