

# 18 Arduino を使用してテーブルのバランスをとり 自走する装置の作成

佐々木 健太

指導教員

菅野 研一

## 1. はじめに

昨年の卒業研究の「Arduino を使用してテーブルのバランスをとる装置の作成」を見て、室内専用の飲み物運び器が作れるのではないかと思い取り組んだ。

また、私が組み込みシステムに興味があり、卒業研究で取り組んでみたいテーマだったため取り組んだ。

## 2. 開発環境について

開発環境は、以下の通りである。

・OS	Windows10
・ハードウェア	Arduino UNO
・使用言語	ArduinoIDE-1.8.4
・アプリケーション	Auduino Bluetooth Controller

## 3. 昨年の装置について

昨年の卒業研究「Arduino を使用してテーブルのバランスをとる装置の作成」で製作された装置を使い、新たに部品を追加し、プログラムの記述を増やすことで本研究の装置の製作を行った。

## 4. 製作物

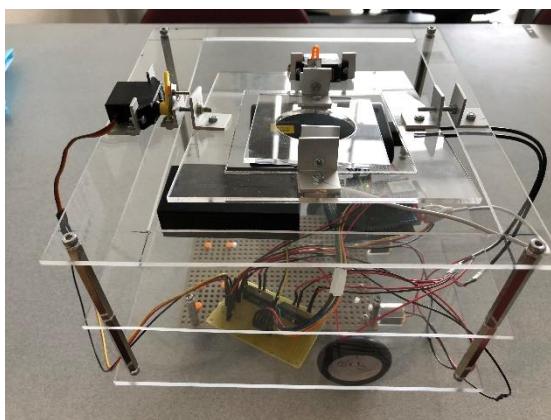


図 1 完成した装置

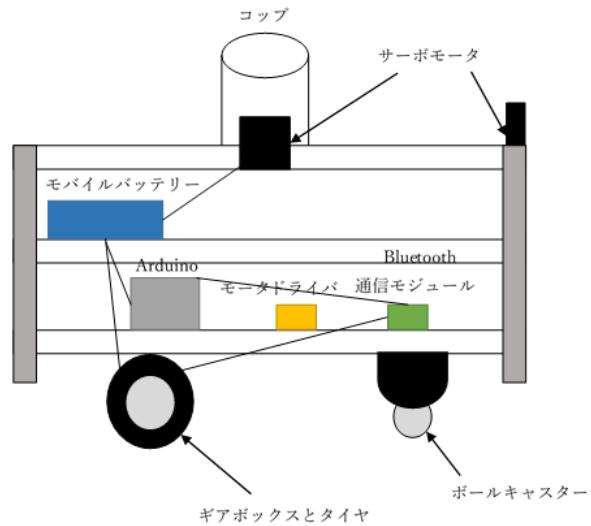


図 2 装置の構成

## 5. 装置の概要

昨年の装置は静止状態でテーブルのバランスを取り水平を保つことで、装置が傾いてもコップに入った飲み物がこぼれないというものであった。その装置にギアボックスとタイヤ、ボールキャスター、モータ ドライバ、Bluetooth 通信モジュールを取り付けて、Google play Store で配信されている、スマートフォンアプリ「Arduino Bluetooth Controller」からの遠隔操作により走行が可能である。電源はモバイルバッテリーから供給する。

主に室内専用の飲み物運び器としての運用を想定し製作した。

## 6. 動作の流れ

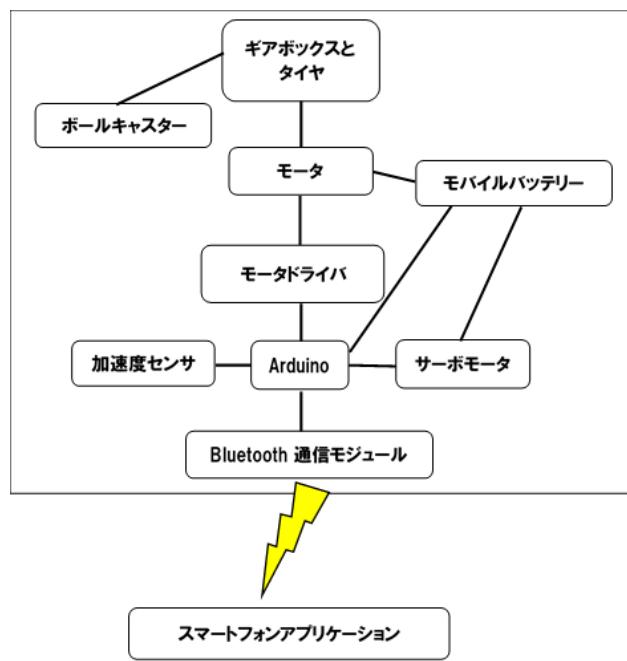


図 3 システム構成図

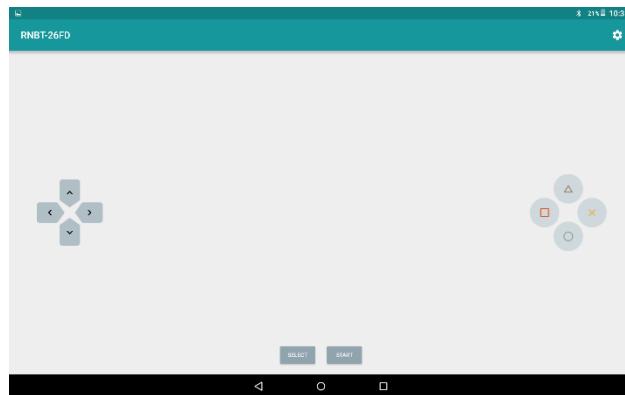


図 4 アプリケーション画面

### 【アプリケーション】

Google play Store で配信されているスマートフォンアプリ「Arduino Bluetooth Controller」を使用する。

アプリケーション画面で進みたい方向の十字キーを押すことで任意の方向へと進む。

○ボタンを押すことでタイヤの停止を行う。

### 【動作の手順】

- ① スマートフォンアプリ「Arduino Bluetooth Controller」のボタンを押す。
- ② RN-42 使用 Bluetooth 通信モジュール評価キットで受信し、Arduino にデータを送る。
- ③ Arduino で受信したデータに対応したデータを TA7291P に送る。
- ④ TA7291P で受信したデータに対応したモータの動作、回転率をギアボックスとタイヤに送る。
- ⑤ モータが指定された動作、回転率で動作する。
- ⑥ 加速度センサの値を読み取り、Arduino に送る。
- ⑦ Arduino で受信したデータに対応した角度をサーボモータに送る。
- ⑧ サーボモータが指定された角度になるよう動作する。

## 7. 追加部品

部品名	制御内容
ギアボックス とタイヤ	装置の走行及び方向転換。 装置の後輪部分。
ボール キャスター	ボール部分が転がるようにな っており、装置を支えている。 装置の前輪部分。
モータドライバ TA7291P	タイヤのモータの回転率や動 作を制御。
RN-42 使用 Bluetooth 通信 モジュール評価 キット	Bluetooth 通信を行う。 スマートフォンアプリで押さ れたボタン情報を受信。
モバイル バッテリー	Arduino、ギアボックスとタイ ヤ、サーボモータ、通信モジ ュールの電源供給に 2 個使用。

表 1 部品表

- ・ギアボックスとタイヤを取り付け走行可能に.
- ・前輪はボールキャスターを設置し、方向転換が可能にした.
- ・モータドライバを取り付け、タイヤの速度調節と方向転換が可能に.
- ・Bluetooth 通信モジュールを取り付け、Bluetooth 通信が可能に.
- ・Bluetooth 通信を用いてスマートフォンアプリ Arduino Bluetooth Controller でタイヤを遠隔操作可能に.
- ・電源をコンセントと PC から供給していたのをモバイルバッテリー×2に変更.

## 8. モータドライバ



図 5 TA7291P

Arduino の Digital 出力とモータを直接つなげるとなれる電流が多すぎて、最悪 Arduino が壊れてしまう危険性がある。そのため今回はモータドライバ TA7291P を仕様が理解しやすく、モータの回転率の調整も楽にできるため採用した。

表 2 端子解説

端子番号	端子記号	説明
1	GND	GND
2	OUT1	出力端子(モータ端子と接続)
3	NC	非接続
4	Vref	制御電源端子(回転率の調整)
5	IN1	入力端子(HIGH or LOW)
6	IN2	入力端子(HIGH or LOW)
7	VCC	ロジック側電源端子
8	VS	出力側電源端子(モータの電源)
9	NC	非接続
10	OUT2	出力端子(モータ端子と接続)

表 3 入力端子解説

IN1	IN2	動作
LOW(0)	LOW(0)	ストップ
HIGH(1)	LOW(0)	回転
LOW(0)	HIGH(1)	逆回転
HIGH(1)	HIGH(1)	ブレーキ

## 9. Bluetooth 通信モジュール



図 6 RN-42 使用 Bluetooth 通信モジュール評価キット

スマートフォンアプリ「Arduino Bluetooth Controller」と通信するために使用。

内臓プロファイルには仮想シリアル接続するための SPP(Serial Port Profile)が組み込まれているため一対一の安定した通信が可能である。

通信距離は~30m 程度。

入力電圧は 5V、3.3V の両方対応。

消費電流は 8~50mA (アイドル時 2mA)

プログラムを書き込む際にはデジタル出力ピンの 0 と 1 番を抜いてから書き込まないとエラーが出る。

接続時の初期設定は(表 4)の通りである。

表 4

設定	値
速度	115200 bps
長さ	8 bit
パリティ	なし
ストップビット	1

## 10. プログラム

テーブルのバランスを取り、水平に保つプログラムは昨年の卒業研究で完成していたため、そのプログラムに記述を加えて機能を実装する。

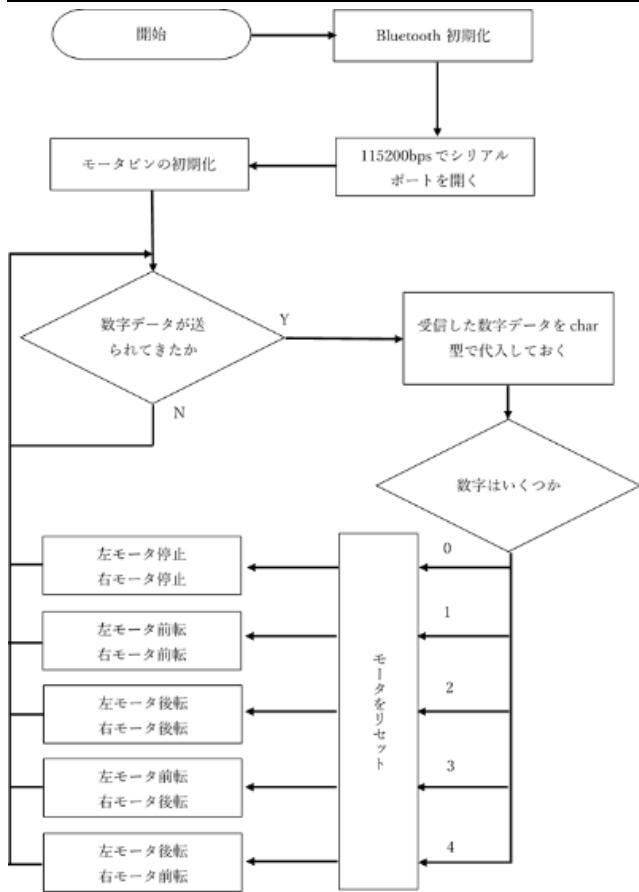


図 7 フローチャート

## 11. 配線図

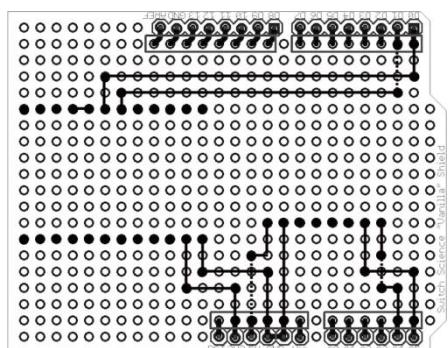


図 8 Arduino バニラシールド

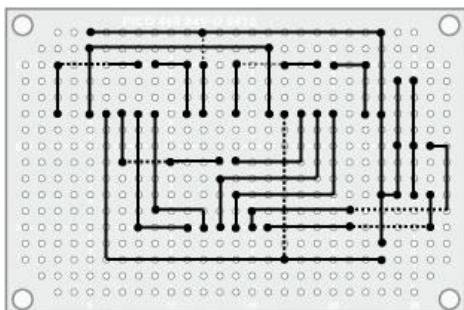


図 9 ユニバーサルシールド

## 12. おわりに

本研究では「テーブルのバランスをとるシステム」と「遠隔操作走行」の両立が可能になった。

昨年の装置のテーブル部分が破損していたため本来行わない作業であったアクリルボードの加工を行い新たに作成した。そのため予定より時間がかかってしまったが、普段行わない作業ができたためいい経験になった。

シールドに配線する際のはんだ付けに、初めて行った作業であったため、予想よりも時間がかかってしまった。

走行時に水平な場所でもテーブルが傾いてしまうことがあった。それは走行時の振動に加速度センサが反応してしまい、誤動作していることが原因だと考えられる。今後の課題としたい。

## 参考文献

[1]Arduino を使用してテーブルのバランスをとる装置の作成

(産業技術短期大学校矢巾校卒業研究報告書)

2016 濱田凌太)

[2]モバイルバッテリーで Arduino とモータを動かす

<https://poli-studio.com/2016/04/12/3098/>

[3]Arduino でラジコン戦車 その 2 無線化

[https://nucoline.wordpress.com/2016/11/04/arduino% E3%81%A7%E3%83%A9%E3%82%B8%E3%82%B3%E3%83%83%BC%8A%88%A6%E8%BB%8A%E3%80%80E3%81%9D%E3%81%AE%EF%BC%92%E3%80%80E7%84%A1%E7%9A%E5%8C%96/](https://nucoline.wordpress.com/2016/11/04/arduino%E3%81%A7%E3%83%A9%E3%82%B8%E3%82%B3%E3%83%83%BC%8A%88%A6%E8%BB%8A%E3%80%80E3%81%9D%E3%81%AE%EF%BC%92%E3%80%80E7%84%A1%E7%9A%E5%8C%96/)

[4]モータードライバー(TA7291P)の使い方 [Arduino]

[https://www.petitmonte.com/robot/motor\\_driver\\_ta7291p.html](https://www.petitmonte.com/robot/motor_driver_ta7291p.html)

[5]Bluetooth 通信する (Bluetooth モジュール RN-42 使用)

[http://www.hiramine.com/physicalcomputing/arduino/bluetooth\\_rn42.htm](http://www.hiramine.com/physicalcomputing/arduino/bluetooth_rn42.htm)