

# 課題研究 LED 電光掲示板の製作 (交通会館連携事業)

## 【研究概要】

本事業では、LED 電光掲示板を東京交通会館に設置することを計画している。掲示板には、マイコンとセンサーによって測定した気温や湿度のデータ、様々なメッセージを表示させることができる。東京交通会館の担当者と打ち合わせを行い、事業内容を少しずつ決めながら研究を進めた。



図 1 貼り付けシールのデザイン

本研究では、交通会館の庭園の出入りに設置する電光掲示板を設置する計画である。

電光掲示板と一緒に交通会館のイメージキャラクターであるトッコちゃんを貼り付け、電光掲示板にはトッコちゃんが話しているように文字を表示するようにする。



図 2 完成予想

## 【開発環境・使用機器】

表 1 プログラム開発環境

品名	型番	用途
(1)マイコンボード	Arduino Nano Every	
(2)エディタ	Arduino IDE 1.8.19	
(3) PC	Windows 10 Enterprise LTSC	
(4)LED8x8 マトリクス	1088AS	表示器本体
(5)LED マトリクスドライバ IC	MAX7219CNG+0610	LED8×8 マトリクスの制御、電力供給
(6)リアルタイムクロックモジュール	DS3231	正確な時間を表示する
(7)温度センサーモジュール	DHT11 SIP	温度や湿度を測定する
(8)マイクロ SD カード	ELECOM 16GB	LEDに表示させる日本語文字のフォントを保存する

(4)LED マトリクス



(5)LED マトリクスドライバ IC



(6)リアルタイムクロックモジュール



(7)温度センサーモジュール



## 【制御基板・回路図】



図 3 制御基板

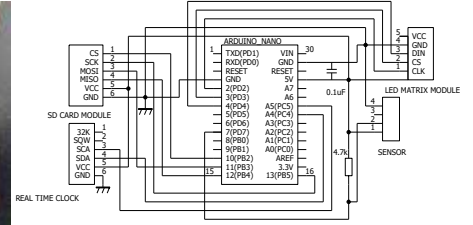


図 4 回路図

## 【これまでの取り組み】

### 1. LED 表示器の制作

昨年度の3年生が作成したプリント基板に、LED マトリクスドライバ IC と LED マトリクスを接続するためのピンソケットをはんだ付けした。その後、実際に基板へ IC と LED マトリクスを取り付けた。

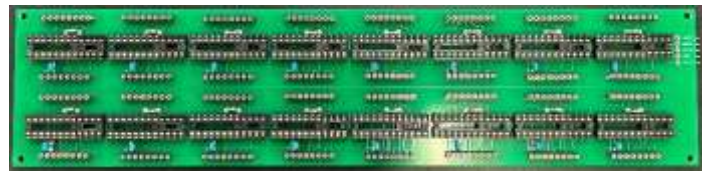


図 5 LED 表示器



図 6 実際の動作の様子

### 2. LED 表示器の制御

LED 表示器への文字出力は、Arduino によって行った。センサーで測定した気温と湿度や、リアルタイムクロックを使って日付や時間を表示することができた。また、交通会館に設置する際、表示するメッセージも検討し、LED 表示器への出力を確認した。

### 3. 表示する文字の考案

気温や湿度、英語表記など季節に応じた注意喚起などを表示するようにした。

(表示例)

- ・気温 26℃ 湿度 50% など
- ・本日のおすすめ商品は○○today`s recommended products are ○○

## 【感想】

今回、交通会館との連携事業で LED 表示器を作って基板製作はミスすることはあったが問題なく作成することができた。また、表示する文字を喋っているようにするなど季節に応じた文を表示するようにして工夫した。

## 【今後の見通し】

今後は交通会館へ行き完成予想図のように設置するのを見学しに行く。

## 8×8×8 LED キューブ

## 【研究概要】

- ・512個のLEDを用いた8×8×8のLEDキューブを製作する。
- ・マイコンを使用してLEDキューブを制御し、音に反応して点灯するプログラムを作成する。

## 【研究内容】

表1

使用部品	個数
青色LED5[mm]	512
抵抗10[kΩ]	8
抵抗330[Ω]	64
パワーMOSFET 2SK4019	8
シフトレジスタ 74hc595	8
シフトレジスタ 74hc238	1
Arduino nano	1
Sound センサモジュール	4
コンデンサ	1

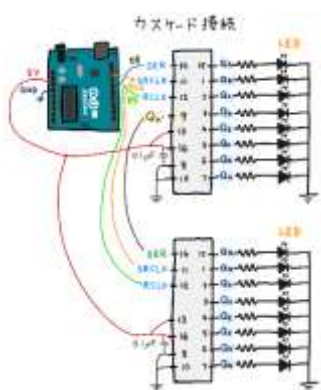
## ①LED キューブの作成



64個のLEDを自作のゲージに合わせてはんだ付けし、8層に重ねてLEDが8×8×8個のキューブを作成。

カソード(−)で隣と繋げる

## ②電子回路の設計



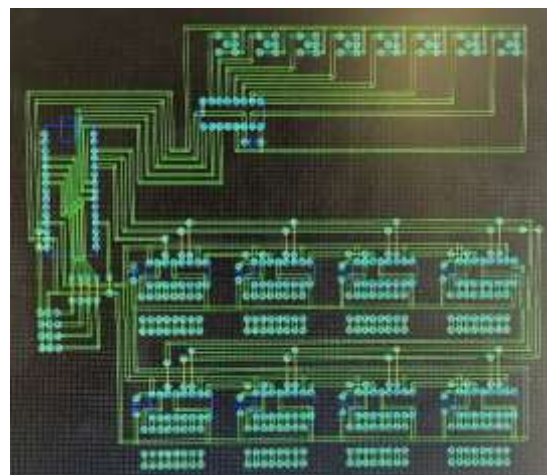
左の図を参考にしながらシフトレジスタ74hc595を8個カスケード接続、74hc238を1個でレイヤー(層)を制御する回路図をそれぞれ作成。

空いたArduinoのポートにSoundセンサモジュール(音センサ)を接続

←参考:<https://iot.keicode.com/arduino/arduino-shift-register-sn74hc595.php>

## ③基板の作成

②で作成した回路図を元に、PCBE(プリント基板パターン作成ソフト)で基盤を設計、基板加工機でプリント基板を制作した。



プリント基板作成

## ④配線

表1の使用部品(LED以外)を基板にはんだ付けしてLEDに繋げた。光らない所があった場合は基板の裏の導線やはんだをチェックし、導通を確認した。



←縦2列が光らない原因を探して修理中

## ⑤プログラミング

実習で使っているArduinoでC言語を用いるプログラミングにした。最初にレイヤーごとに光るプログラムで点灯確認し、ダイナミック点灯ができてから音に反応して点灯させるプログラムを作った。pinMode()やshiftOut()など初めて見るものだったので苦労した。

## 【反省】

杉工祭までにLEDをセンサに反応させて光らせる所までは行けたので良かったです。まだまだ書き込むプログラムがたくさん残っているので、もっと基盤やキューブ本体の作成を早く終わらせてプログラミングに時間を回せたら間に合ったのかなと思いました。

# マイコンによる LED 電光掲示板の制御

## 【はじめに】

電車の車体側面には行先表示板が付いている。よく見るとたくさんのLEDがドット状に付いていることに気が付き、どのようにして沢山のLEDを制御しているのか興味をもち、これを課題研究のテーマとした。

HUB75規格の電光掲示板(図1)を使用し、制御には専用のコントローラは使用せず、マイコンボードのESP32を使用し、制御用プログラムを自作することにした。

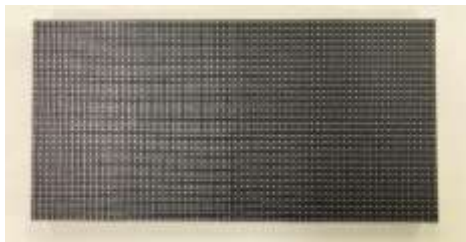


図1 LED電光掲示板

## 【開発環境】

表1 プログラム開発環境

名称	仕様
PC	メモリ 16GB, CPU 2.60GHz, 64bit
OS	Windows10 Home
エディタ	Arduino IDE 1.8.19

表2 使用機器一覧

名称	仕様
LEDマトリクスパネル 64x32	定格5[V] 4[A]
ESP32	DevKit:C
ACアダプター	出力5[V] 10[A]

## 【回路図】

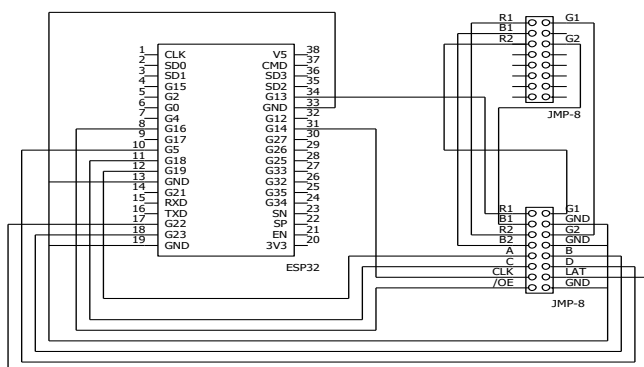


図2 専用基板の回路図

## 【専用基板の作成】

マイコンと電光掲示板の接続を容易にするため、専用の基板をpcbeで制作した。(図3) 基板の回路図は先述の図2のとおりである。



図3 専用基板

## 【プログラム】

行先表示板の再現のため、ESP32で電光掲示板を動かすためにPxMatrixというライブラリを使用することにした。

PxMatrixの画像表示プログラムを使用して製作するため、画像編集ソフトGimpを使用し、画像の作成をまず行った。マトリクスパネルのサイズに合わせて画像を作成したら、Image Convaterというサイトで画像を.cファイルに変換する。ファイルをダウンロードして、バイナリの数値だけをコピーしたらPxMatrixの画像表示を行う.inoファイルにバイナリの数値を張り付ければ準備完了。

注意点が1つあり、プログラム中の"display.setDriverchip(FM6126A)"が有効だと電光掲示板が光らない場合がある。

光らない原因は、使用している電光掲示板がドライバチップとしてFM6126Aを使用していない場合、このプログラムをコメントアウトする必要がある。

## 【LED点灯例】

① PxMatrixライブラリを用いて行先表示板の表示を一部再現(図4)



図4 64x32マトリクスパネルが動作している様子

② マトリクスパネルを2枚連結し、行先表示板の表示を再現(図5)



図5 128x32マトリクスパネルが動作している様子

## 【既知の不具合】

・マトリクスパネルを2枚に連結すると異常な表示になってしまう。

## 【感想】

3年間の実習で学んだマイコンの知識を元に調べる事から始めたが、わからないことがたくさんあった。ほとんど0からのスタートとなったが、数か月かけて製作を行い、行先表示板の再現表示が出来た時とても感動した。2枚連結時にパネルの表示が乱れる不具合は直すことが出来なくて残念だったが、素人でも専用のコントローラを使用せず再現表示が出来たことを知り、ESP32の万能さに驚きました。

## 四足歩行ロボット

### 【研究目的】

どのような原理・法則があり、自分の実力でどんな風に動かすことができるのかを試したかった。

### 【使用機器】

- ・ESP32-DevkitC ・サーボモータ(HSR-8494HB)
- ・操作端末(android) ・PC(HUAWEI)
- ・モバイルバッテリー(5V 4000mAh)
- ・ニッカドバッテリー(7.2V 2200mAh)
- ・3Dプリンタ

「NOVA3D Elfin3 mini」「XYZprinting da Vinci」

### 【開発環境】

- ・Arduino IDE ver. 1.8.16 ・Autodesk Fusion 360
- ・pcbe ver.0.63.13

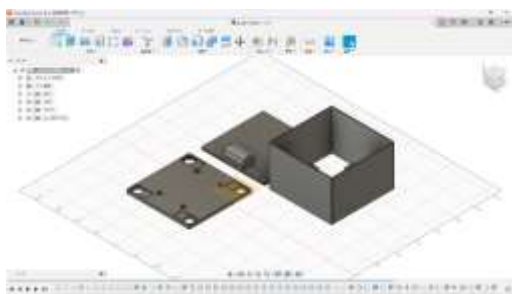
### 【ESP32 について】

ESP32 とは Wi-Fi と Bluetooth を内蔵した Espressif Systems 社のワンボードマイコンの一種である。Arduino と同様に開発環境を整えることにより Arduino IDE でプログラムを開発することができる。



### 【Fusion360 について】

Fusion360 とは、Autodesk 社が提供している 3DCAD ソフトであり、製図以外にも様々な機能が搭載されている。



### 【研究内容】

#### (1) 試作品について

まず最初に、簡単に形を考え設計し試作品を製作しました。四足での歩行方法を調べ実際に動か

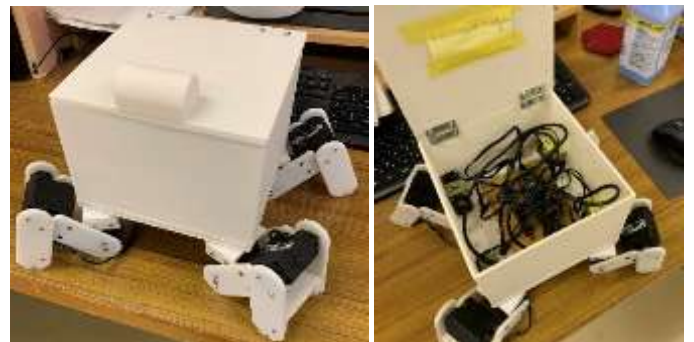
しながら実験を繰り返し、前進させるところまで出来ました。

#### (2) デザイン設計



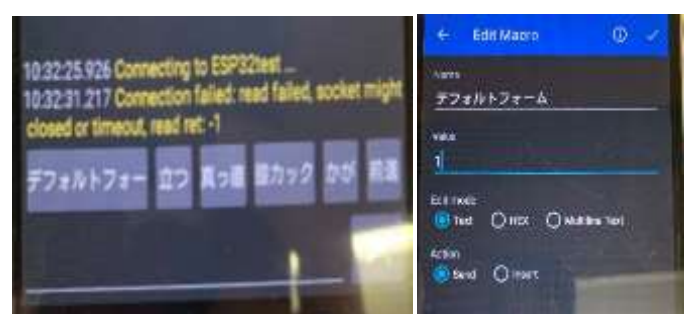
試作ではサーボモータの配線が丸見えだったので接続部分を全部隠すような箱形のデザインにしました。名前は豆腐に似ていることから「とーふーちゃん」です。

配線などに問題があった際などにすぐにメンテナンスできるように、上の蓋が開くようになってます。



#### (3) 操作方法

ロボットの操作は Android のスマホで「Serial Bluetooth Terminal」というアプリを使用しています。下の入力画面に特定の数字を打って送信するか、固定されているボタンを押すとシリアル信号が送られ動かすことができます。



# 画像処理マイコンカーの制御

## 【研究概要】

画像処理を利用したマイコンカーラリーの製作をして、大会に出場する。

### 1. 【目的】

画像処理マイコンカーを製作し、南関東大会出場を目標として、マイコン、およびプログラミングに対する理解を深める。

### 2. 【使用機器・開発環境】

#### (1) 使用機器（指定部品を使用）

- ・GR-PEACH                      ・電池ボックス
- ・モータドライブ基板        ・スタッド
- ・GR-MCR シールド基板    ・カメラ (jpeg)
- ・サーボモータ(HS-430)    ・タイヤ
- ・ギヤーボックス



プログラムを打ち込む作業は今も続いており、当面の目標はコースを一周させるということです。走らせるためにはカメラにコースの白い部分を読み取らせそこから振れずに走るよう数字を打ち込んで調整するのですが、一つ数字をずらすだけでかなり変わるので走らせて、打ち込んで、を繰り返す大変な作業です。

コースを一周することはできましたが、ここからスピードを上げていかなければいけません。

スピードをあげると角度の数値も変わってくるので、調整は更に難化すると思います。

### 3. 【内容】

#### (1) 本体の完成

中間報告の時終わっていなかった本体の作成を終わらせました。

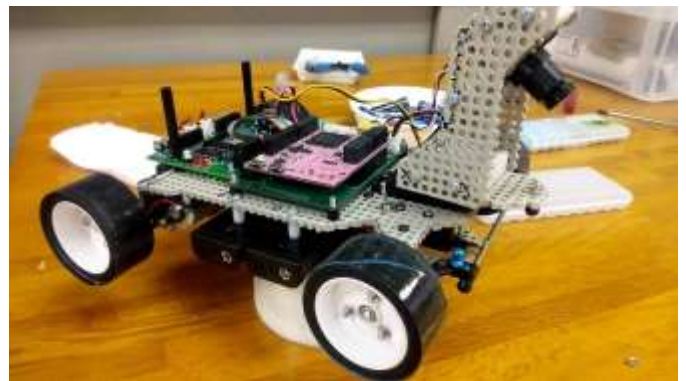
#### (2) プログラムの作成

- ・カメラで読み込む光の調整

これは白線が反射した光を正確に読みこませ走らせる作業です。

- ・角度の調整

直線でも曲がりでもそれぞれの数値が必要で、その設定をします。これはまだ途中です。



### 4. 【感想】

今回の文化祭発表までの作業を終えて感じたことは、作業の見積もりが甘かったということです。結果としてちゃんとコースを一周することができましたが、それまでの授業での作業時間内では終わっておらず、放課後もそれぞれの用事であまり残ることはできませんでした。

文化祭発表後は、本格的に大会に向けて作業していきます。そのため、もっとも懸念されるのは時間の活用だと感じました。時間を効率よく使いつつ、尚且つ取れる作業時間も増やしていけるようにしていきたいです。

目標である南関東大会出場と入賞を目指して頑張っていきたいと思います。

## 課題研究 3D シミュレーションゲームの制作

### 【1.目的】

- ・3D シミュレーションゲームの制作

### 【2.目標】

- ・広大なフィールドを走り回れたり、キャラと話したり 戦闘を楽しめるような RPG ゲームを作る。

### 【3.開発環境】

	品名	仕様・定格	メーカー
1	ハードウェア	AMD 3020e with Radeon Graphics 1.20GHs	Lenovo ideaPad3 15ADA05
2		Intel(R)Celeron(R)N4000 CPU@1.10GHz 1.10GHz	ARROWSTab Q5010/CE
3	OS	Windows 11 Home	
4	開発ソフト	2.4.5	Unity Hub
5		2020.3.16f1	Unity

### 【4.研究内容】

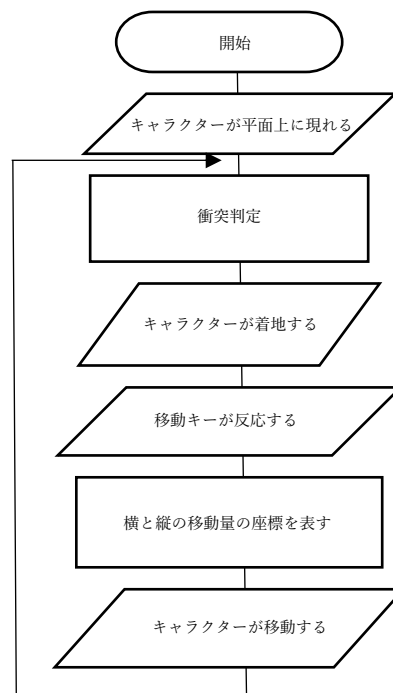


図1 フロチャート

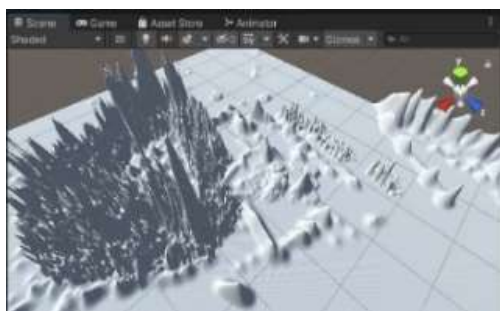


図2 マップの制作



図3 HP ゲージの実装



図4 会話の実装

障害物の衝突判定と物体を動かすプログラム (図1) を終了させ、現在は unity を使ってマップ制作 (図2) 及び NPC と会話するプログラム (図4) を制作している。

### 【5.検討考察】

- ・HP の減少するプログラムを作動させようとする (図3) プログラムがうまく作動しないため、プログラムの見直しをして修正していく予定である。
- ・移動キーを二つ同時に押しながら、ジャンプキーを押すとジャンプキーが反応しない不具合 (図1) を見つけたため、移動キーのプログラムを確認して対処していく。

### 【6.今後の展望】

- ・WASD で操作して視点移動はマウスで行う。
- ・ある程度自由なマップを作ってプレイヤーを飽きさせないようなゲームを作る。

# エレキギターのためのエフェクタ製作

## 1 「目的」

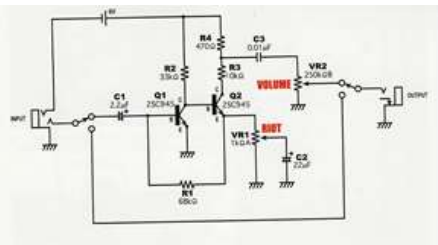
文化祭ライブで使用し、軽音部に寄付するため。

## 2 「使用部品」

ステレオジャック・モノラルジャック・LED・コンデンサ・9ピンスイッチ・DCジャック・電池スナップ・ダイオード・ICソケット・トランジスタソケット・ツマミ・オペアンプ・トランジスタ・可変抵抗器・抵抗器・コンデンサ・ダイオード  
参考「ド素人のためのオリジナルエフェクター製作」

## 3 「ファズ」

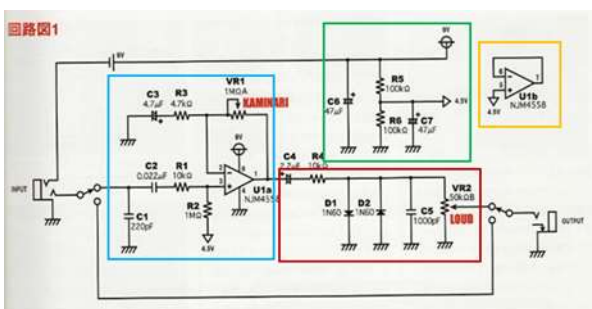
トランジスタの二段直結回路で構成されている。1段目から2段目へおくる電流の一部を入力側に帰還して増幅率を調節して歪みを出している。増幅量を増やすと歪みが深くなる。今回は可変抵抗で増幅率を調整し、音の歪み具合を出すようにした。



## 4 「ディストーション」

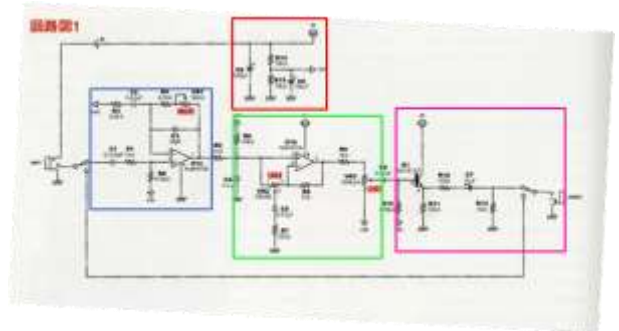
オーバードライブと同様の回路で構成されているが、オペアンプで増幅後にダイオードのクリップを用いて人為的に歪みを作り出している。

今回は非反転増幅回路という接続方法を用いており、増幅率は帰還部の可変抵抗の値で決まる。



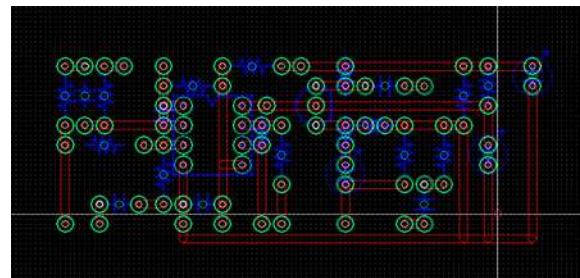
## 5 「オーバードライブ」

オーバードライブとは、電子楽器から過大な出力電圧が加えられるか、アンプ回路内部で過大增幅に陥った際に、回路性能の限界で飽和し出力音が歪んでしまう状態のこと。今回はオペアンプの帰還部にダイオードを挿入し音を歪ませている。



## 6 「作業」

学校にないパーツは秋葉原の「千石電子」と「秋月電子」で購入してきました。参考資料を元に「pcbe」を使用し、基板パターンを作成しました。



その後、基板を加工し、基板にパーツをはんだ付け、ケース加工と配線を行った。また、実際にギターに接続して波形をとるなどし、音やノイズの確認をした。



入力波形は正弦波



出力波形は歪んで方形波へ

## シャトル打ち出し装置

## 【研究目的】

バドミントンのノック出しを、誰でもでも出来るようにするためのシャトル打ち出し装置を、モータを使い作成する。

## 【使用機器】

名称	定格	メーカー
RS-380PH-4045	7.2V	マブチモータ
単1電池(4本)	1.2~1.5V	FUJITSU

## 【内容】

1. RS-380PH-4045のモータとスポンジを購入し、スポンジを丸く切り、それをモータにくっつけて、シャトルを飛ばす機構を作成しました。実際にモータを動かしてみるとかなりのパワーがあるのを確認できました。シャフトの長さが少し短かったので棒をくっつけて長くしました。スポンジを綺麗な丸に切るのは難しかったです。



成しました。実際にモータを動かしてみるとかなりのパワーがあるのを確認できました。シャフトの長さが少し短かったので棒をくっつけて長くしました。スポンジを綺麗な丸に切るのは難しかったです。

2. 木材を加工し装置の土台を作成し、実際にシャトルを発射できるようにしました。簡単な設計図を描いたことで作業をスムーズに進めることができました。モータ自体はスポンジに穴を空けそこに固定しているので、スポンジが入るぐらいの穴を空けその中に固定しました。スポンジはモータの振動を軽減する目的で使用しました。



3. 装置自体の移動を簡単にするために椅子の足を使うことにしました。壊れている椅子の足の部分を取り外し足の上に木の板を



くっつけそこに土台を乗っけるようにして固定しました。作業自体はそこまで難しいことはなくスムーズに進めることができました。実際にシャトルを飛ばしてみましたがシャトルを入れる角度で飛び方にムラがありました。シャトルを少し傾ける程度の角度が飛びやすいことがわかりました。

4. モータをスイッチで制御できるようにし電源は単1電池を4本使用しました。中が見えないように蓋を付けました。無事に動かすことができシャトルをある程度の距離まで飛ばすことができました。



## 【感想】

作業自体は5,6月頃から始めていたのですがそれらは全て失敗作になってしまい事実的に作業が始まったのは夏休みに入ってからになってしまいました。ですが作業が進み完成が近づいてくると少しずつ作業が楽しくなってきました。シャトルを飛ばせたときは感動も覚えました。細かいところを見ていくと不完全な部分もまだあるので今後の時間で直していきたいと思います。