

2023 年度 (令和 5 年度)

創造工学セミナー II Final Report

手認識による  
集合住宅用ダストボックスの  
自動開閉機能の開発

研究メンバー

S520026 佐々木花野

S520032 仙台昌太

S520052 藤枝翔太

S520068 渡邊雅也

指導教員

金丸隆志 教授

# 目次

第1章 緒言 (藤枝担当)	1
1.1 研究背景	1
1.2 類似研究	3
1.3 研究目的	4
第2章 研究概要 (佐々木担当)	5
2.1 システム概要	5
2.2 認識手段	6
2.3 手認識に用いるハードウェア	7
2.3.1 認識用カメラ	7
2.3.2 システム動作用パソコン	8
2.4 ソフトウェア	9
2.4.1 開発環境及びライブラリ	9
2.4.2 MediaPipe	9
第3章 手認識 (佐々木担当)	10
3.1 手の骨格検出	10
3.2 手のパターン	11
3.2.1 認識パターン	11
3.2.2 パターン学習の手順	12
3.2.3 学習データの用意	12
3.2.4 学習結果の確認	12
3.3 認識精度評価実験	13
3.3.1 実験概要	13
3.3.2 実験手順	13
3.3.3 実験結果	13
3.3.4 実験考察	14
3.4 段階認識について	14
3.4.1 段階認識プログラム	15
第4章 ダストボックスのサイズとカメラの配置 (仙台担当)	18
4.1 ダストボックスのサイズ決定	19
4.1.1 ダストボックスのサイズ	19
4.1.2 最大サイズのダストボックス	20
4.2 手認識用カメラの位置及び角度の実験	20
4.2.1 実験概要	20
4.2.2 実験準備	21
4.2.3 実験方法	24

4.2.4 実験結果 .....	25
4.2.5 実験考察 .....	26
第5章 ダストボックスの模型製作（佐々木担当） .....	27
5.1 ダストボックス本体の製作 .....	27
5.1.1 本体の設計 .....	27
5.1.2 アクリル板のねじ穴開け加工 .....	30
5.1.3 ダストボックス本体の完成 .....	32
5.2 ダストボックスの蓋の製作 .....	32
5.2.1 蓋の設計 .....	32
5.2.2 ダストボックスの蓋の完成 .....	35
5.3 ダストボックスの模型の完成 .....	36
5.3.1 ダストボックスの組み立て .....	36
5.3.2 ダストボックスの模型の修正と完成 .....	37
第6章 模型の完成（渡辺担当） .....	38
6.1 模型の動作に必要なハードウェア .....	38
6.1.1 Arduino .....	38
6.1.2 サーボモーター .....	38
6.1.3 電源 .....	40
6.2 プログラム .....	41
6.2.1 Arduino シミュレーション .....	41
6.2.2 Arduino 側のプログラム .....	42
6.2.3 Python 側のプログラム .....	43
6.3 開閉機能の動作確認 .....	45
6.4 ダストボックスと開閉機構の接続 .....	47
6.5 手認識による集合住宅用ダストボックスの自動開閉機能の完成 .....	47
第7章 結論（藤枝担当） .....	49
謝辞 .....	53

# 第 1 章 緒言 (藤枝担当)

## 1.1 研究背景

近年、生活様式の変化により非接触の商品が注目されている。例えば、図 1 のように手をかざすことによってアルコールが噴出されるアルコールディスペンサー[1]や、足元にあるペダルを踏むことによって蓋が開閉するゴミ箱[2]などがあり、非接触の製品は多く存在している。特にアルコールディスペンサーは、公共の場に設置され不特定多数の人が利用することから非接触化する意義があると考えられる。



図 1 アルコールディスペンサー[1]

同じように公共の場で不特定多数の人が触れるものとして、マンションやアパートなどの集合住宅の屋外に設置されている共用のダストボックスに着目した。ゴミは生活していく上で必ず発生する物であり、多くの世帯のゴミが一つの大きな屋外のダストボックスに集められる(図 2)。数多くのゴミが投棄され、大勢の人が触れる屋外のダストボックスは衛生状態が悪く、不快感を覚える人もいるだろう。そのようなダストボックスを集合住宅の住民やゴミの回収業者など不特定多数の人が直接手で開閉しなければならない。また、カラスによるゴミの荒らし[3]を未然に防ぐため、ダイヤル式の南京錠などで施錠されているダストボックスも存在する。ダイヤル式の南京錠がかけられている場合、ダストボックスを開ける度にダイヤルを回して解錠し、その都度施錠する必要がある。

このように、衛生状態が悪いにも関わらず不特定多数の人が触れなければならないダストボックスを非接触化するニーズは高いと考えられる。



図 2 本研究メンバーの住宅に置かれたダストボックス

## 1.2 類似研究

本節では、日用品の非接触化に関連する類似研究を紹介する。

顔らは、「任意の位置での簡単なジェスチャによる家電操作システム」という、簡単なジェスチャで家電を操作するシステムを開発した[4]。

この研究では家電を操作するために、まず手を振ることにより、その位置に疑似相対座標系を定義する。次に、定義した座標系にコマンド空間を作成する(図 3)。コマンド空間に手をかざすことで家電操作が可能になる。しかし、このシステムは、家電が動作するまでに時間がかかってしまうという欠点がある。その理由は、複数のデバイス进行操作するために手順が多くなっているからである。また、操作完了後は定義した疑似相対座標系や、コマンド空間を消去する手順を忘れると、他の場所での家電操作が出来なくなる。

そこで本研究では、対象のデバイスを一つに絞ることで操作手順を簡略化し、短時間での操作可能なシステム開発を行っていく。

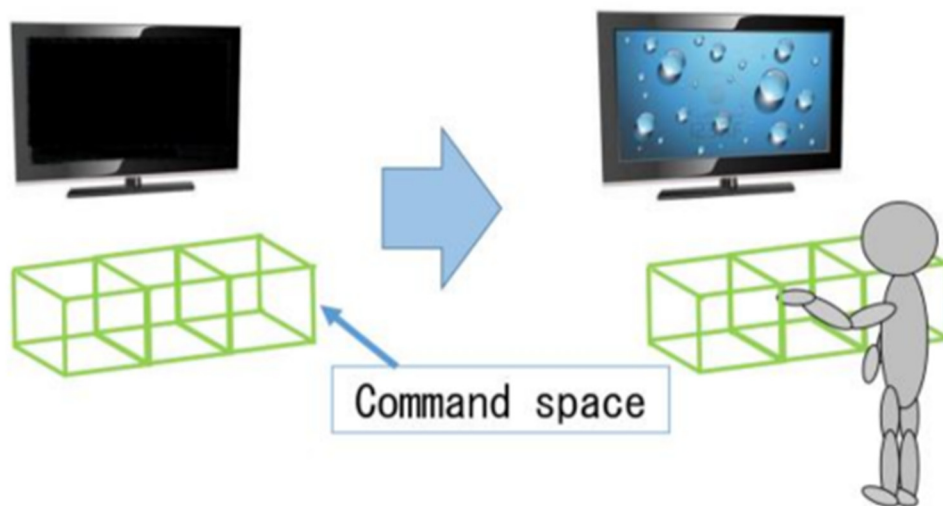


図 3 コマンド空間の概念図[4]

### 1.3 研究目的

以上から、本研究の目的は、非接触で解錠と開閉を行えるダストボックスを開発することに決定した。これによりゴミ袋を持っていて、片手が塞がっていた場合や、手が汚れていた場合でも、ダストボックスに触れずに開閉が可能になる。その結果、不特定多数の人が触れることがなく、衛生面に配慮した不快感の少ない製品になることが期待される。

さらに、非接触でのセキュリティの確立も目標とする。すなわち、屋外のダストボックスの製作におけるダイヤル式の南京錠の役割を機能に含める。その手段については 2.2 に示す。以上の目標が達成されることで、ゴミの荒らしなどの被害を防ぎつつ衛生面に配慮したダストボックスを実現できるだろう。

## 第2章 研究概要 (佐々木担当)

### 2.1 システム概要

1.3の研究目的を踏まえ、本研究では手認識によるダストボックスの自動開閉機能の開発を行う。非接触でダストボックスのロックの解除を行い、蓋を自動開閉するものである。システムのイメージ図を図4に示す。

このシステムは以下の流れで動作する。まず、ダストボックスに設置されたWebカメラにあらかじめ決められたハンドジェスチャを行う。次に、カメラで取得したハンドジェスチャをプログラムにより認識する。最後に、認識されたハンドジェスチャがあらかじめ決められたものと等しければダストボックスの蓋が自動的に開く。その後、一定時間経過後に自動的に閉まる。

このシステムを開発するために、本研究ではハンドジェスチャの認識システム、ダストボックス、モーターを用いた開閉機構の三点を製作することを目標とする。

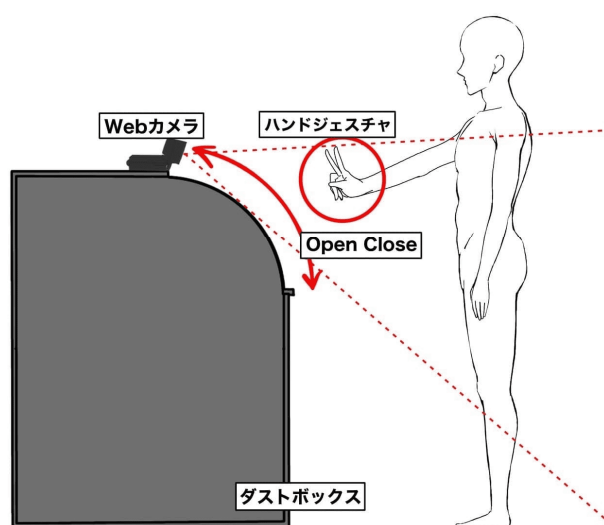


図4 自動開閉機能の完成予想図



## 2.2 認識手段

上で述べたように、ダストボックスの解錠と開閉には、ハンドジェスチャの認識、すなわち手認識が必要になる。非接触で解錠するにあたり、認識手段は手認識の他に顔認証や音声認識、赤外線センサーなどが考えられる。しかし、集合住宅用のダストボックスは屋外に設置されているため、直射日光があたると誤動作する赤外線センサーや、環境音に左右される音声認識は使用できないと判断した。また、顔認証は、マスクや眼鏡、帽子などで顔が隠れた場合使用できない。加えて、顔の情報を登録することはプライバシーの観点から不向きである。

これらを踏まえて、手認識を本システムの認識手段とした。

## 2.3 手認識に用いるハードウェア

本節では、手認識に用いるハードウェアの説明を行う。それ以外のハードウェアについては 6.1 を参照すること。

### 2.3.1 認識用カメラ

手認識用カメラとして、Logicoool 社製の C920n HD PRO ウェブカメラ[5](図 5)を用いる。詳しい仕様を表 1 に示す。



図 5 Logicoool C920n[5]

表 1 WEB カメラ仕様

型番	Logicoool C920n
寸法(高さ×幅×奥行)	43.3mm×94mm×71mm
ケーブル長	1.5m
重量	162g
最大解像度	1080 p/30 fps - 720p/ 30 fps
カメラ画素数	3MP
フォーカスタイプ	オートフォーカス
レンズタイプ	ガラス
対角視野 (dFoV)	78°
対応 OS	Windows8 以降 macOS 10.10 以降 ChromeOS
接続コネクタ	USB-A ポート

### 2.3.2 システム動作用パソコン

システム動作用パソコンとして Dell 社の Inspiron7391[6](図 6)を用いる。詳しい仕様を表 2 に示す。



図 6 Dell Inspiron 7391[6]

表 2 パソコンの仕様

型番	Dell Inspiron 7391
OS	Microsoft Windows 11 Home
CPU	Intel(R) Core(TM) i7-10510U
メモリ	8GB

## 2.4 ソフトウェア

本節では、研究で用いるソフトウェアの説明を行う。

### 2.4.1 開発環境及びライブラリ

システムの開発にあたり、Anaconda[7]というツールをインストールし、機械学習用の環境を構築した。開発環境は Spyder を利用し、プログラミング言語は Python 3.8.16 を用いた。使用したライブラリ等を表3に示す。

表3 ライブラリの仕様

名称	バージョン
MediaPipe	0.9.3.0
OpenCV	4.6.0
numpy	1.23.5

### 2.4.2 MediaPipe

本研究では手認識の際に手の骨格検出を行う。手の骨格検出には MediaPipe[8]を用いる。MediaPipe とは Google 社が提供するオープンソースの機械学習ライブラリである。MediaPipe は C++、Python などの言語に対応している。本研究では、Python により MediaPipe を用いる。MediaPipe では、画像や動画に対して顔や手、ポーズなどのリアルタイム検出を行うことができる[9](図7)。

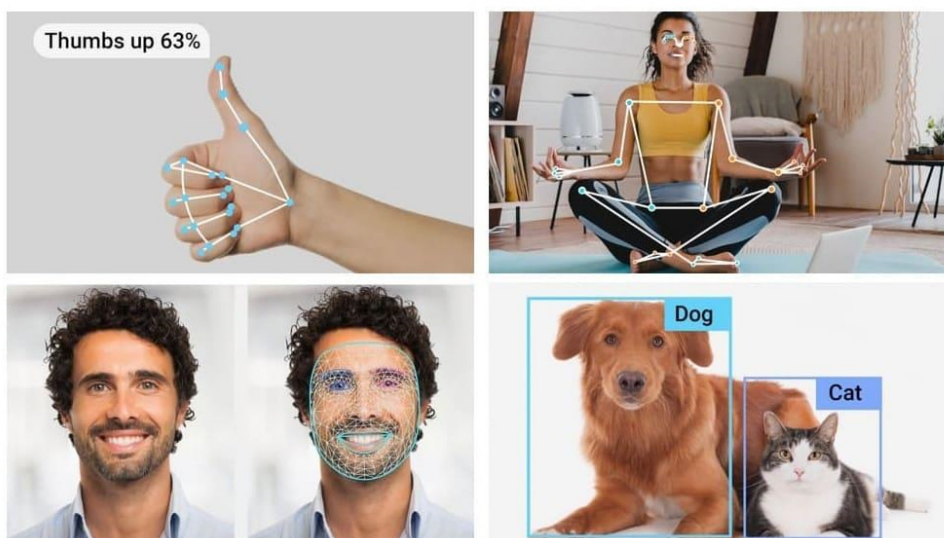


図7 MediaPipe のソリューション例[9]

## 第 3 章 手認識 (佐々木担当)

本研究では、MediaPipe を用いて手を認識する。ここで、本研究では手認識を「手の骨格検出とポーズ判定を組み合わせたもの」と定義する。本章では、手の骨格検出及びポーズ判定の詳細について説明を行う。

### 3.1 手の骨格検出

手の骨格検出には MediaPipe の提供するソリューションの MediaPipe Hands を用いる。MediaPipe については 2.4.2 を参照すること。MediaPipe Hands では、手に 21 個のキーポイントが設定されており、これらキーポイントの位置が推定される。21 個のキーポイントを図 8 に示す。

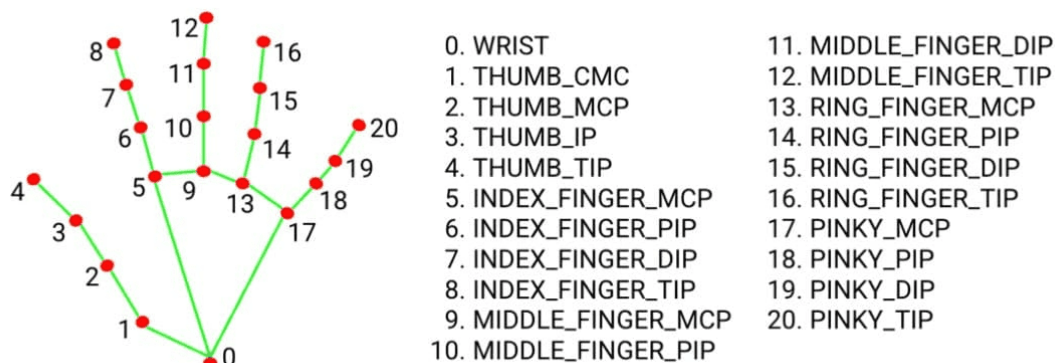


図 8 MediaPipe が定義するキーポイント

実際に MediaPipe Hands のサンプルプログラム[10]を実行した様子を以下の図 9 に示す。

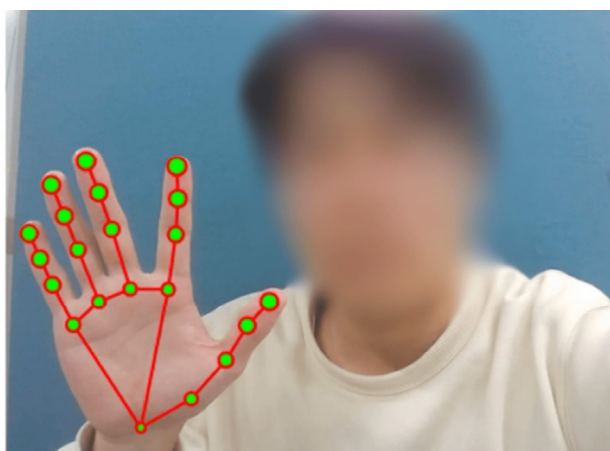


図 9 MediaPipe Hands サンプルプログラム実行の様子

## 3.2 手のパターン

本節では、手のパターンについて説明する。本研究での手のパターンとは、以下のようなハンドサインを指す。これらのパターンを用いて、ダストボックスの自動開閉を行う。

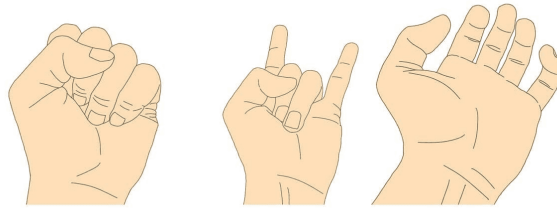


図 10 パターンの例

### 3.2.1 認識パターン

本項では、認識パターンについて説明する。本研究では、「チョキ」「グッド」「オッケー」の 3 パターンを、ダストボックスを開けるための認識パターンに決定した。加えて、これら 3 パターンと区別するために「その他」のパターンも用意する。また、本研究ではチョキをパターン 1、グッドをパターン 2、オッケーをパターン 3、その他をパターン 0 と設定した。決定した認識パターンは、パターン認識プログラムのニューラルネットワークの学習によって認識する。パターンの学習は、認識するパターンの手のキーポイントの座標を使用して行う。学習の手順は次項で説明する。

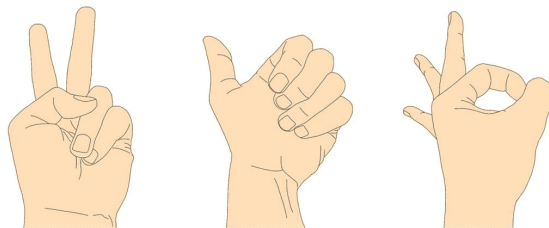


図 11 認識パターン

### 3.2.2 パターン学習の手順

本項では、パターン学習の手順を説明する。また、以下の手順は Kazuhito00 氏のプログラム[11]に本研究室で開発したパターン認識用プログラムを組み込んで利用する。

まず、mp\_hand\_record\_skeletons.py を用いて、特定のパターンのキーポイントの座標を記録する。このプログラムを Spyder 上で実行するとリアルタイムのカメラ映像が開く。このプログラムで開く映像上で「r」キーを押すと 5 秒間の待機状態ののち、21 個のキーポイントの座標が 2 秒ごとに 10 回記録される。このデータをパターンごとにフォルダに保存することで、学習データとしての用意が完了する。

次に、用意した学習データを用いて、mp\_hand\_learn\_skeletons.py によりパソコンに学習させる。このファイルを実行することで、学習結果を保存したファイル(拡張子 h5)を保存する。

最後に、mp\_pose\_recognize\_skeletons.py を実行することで、学習に基づいた認識結果が表示される。

### 3.2.3 学習データの用意

パターン 1、2、3 については 80 個の学習データを用意した。同じ手のパターンでも、カメラからの近さや、カメラに映す手の角度を変えながらデータを保存している。パターン 0 については 100 個のデータを用意した。手の形状としては、脱力している状態や、傘を持っている状態など日常生活で起こり得そうな状態としている。

### 3.2.4 学習結果の確認

3.2.3 のデータを学習後、mp\_pose\_recognize\_skeletons.py を実行した結果が図 12 である。

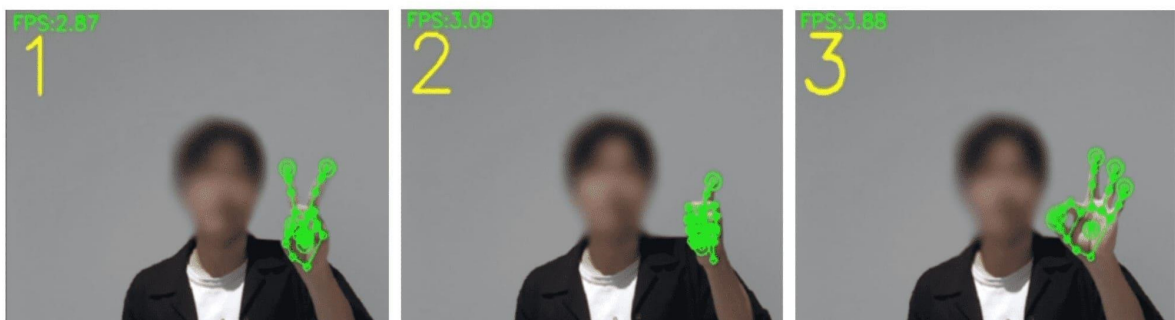


図 12 パターンを認識している様子

チョキをパターン 1、グッドをパターン 2、オクケーをパターン 3 と、設定したとおりに認識している。以上より、パターンの学習は成功したといえる。

### 3.3 認識精度評価実験

前節で行った、パターンの学習の認識精度を評価する実験を行う。

#### 3.3.1 実験概要

学習させた手のパターンを認識させたときに、パターンごとに設定した番号が画面上に正しく表示されていたかで、認識精度を評価する。

#### 3.3.2 実験手順

認識精度評価実験の手順は以下の通りである。

1. パターン 1、パターン 2、パターン 3 を 100 回ずつ行う
2. 認識精度を評価する

100 回の各回の間は手をカメラに映らないようにし、画面上から手の座標が消えたことを確認してから次のパターンを行う。また、パターンを行ってから 1 秒間正しい表示が出たら認識成功とカウントする。

#### 3.3.3 実験結果

実験結果は表 4 のようになった。

表 4 認識精度の評価

パターン	認識成功回数
チョキ	95
グッド	97
オッケー	90



### 3.3.4 実験考察

それぞれのポーズの認識成功回数は、チョキが95回、グッドが97回、オッケーが90回となった。いずれのパターンでも認識成功回数は90回を超え、実用可能だと考える。

オッケーの認識精度が特に低くなった原因としては、似ているパターンをパターン0に学習させたからだと推測する。オッケーの手の形状は、親指と人差し指以外パーの形状に類似している。そのため、パーの手の形状を学習させたパターン0と誤認識することが多かったと考える。

また、手認識の限界を調べるために、手を後ろ方向に倒した時に手自体が認識されなかったことも、認識精度を下げる原因となった。その原因を踏まえ、正しく認識できるようにカメラの正面に立ち、パターンを出したときには100%認識が成功した。

## 3.4 段階認識について

本節では段階認識について説明する。学習させたパターンは3つである。これら3つを連続で、特定の順番で認識したときにのみダストボックスが開くよう設定する。ここで、本研究では段階認識を「3パターンを連続で、特定の順番で段階的に認識するもの」と定義する。段階認識を用いる理由としては、1つのパターンでは意図しない手を認識し、誤って開くことを防ぐためである。本研究では、「チョキ」→「グッド」→「オッケー」の順で段階認識したときに開くよう設定する。

### 3.4.1 段階認識プログラム

本項では段階認識プログラムについて説明する。3.4の通り、「チョキ」→「グッド」→「オッケー」の順で認識したときにダストボックスを開くよう設定する。また、段階移行後に3秒間次のジェスチャの入力がなかった場合にはこれまでの入力のリセットされ、第一段階に戻る。これにより、意図しないパターンの入力を防ぐことができる。

動作手順は以下の通りである。下記のコードはカメラから映像を取得する while ループに含まれている。さらに、パターンの認識も同じループ内で行われ、変数 result に認識結果が含まれている。

第一段階の状態では result が 1(チョキ)の場合、次のループで第二段階に移行する。そして第二段階の状態では result が 2(グッド)の場合、次のループで第三段階に移行する。最後に第三段階の状態では result が 3(オッケー)の場合ダストボックスを開く、という仕組みである。なお、第一段階と第二段階の間、または第二段階と第三段階の間に 3 秒経過すると第一段階に戻る。

```
if state == dankai1:
    if result == 1:
        t1 = time.time()
        state = dankai2
        print("第二段階")

elif state == dankai2:
    t2 = time.time()
    if (t2 - t1) > 3:
        state = dankai1
        print("一定時間入力がないため第一段階に戻ります")
    elif result == 2:
        t3 = time.time()
        state = dankai3
        print("第三段階")

elif state == dankai3:
    t4 = time.time()
    if (t4 - t3) > 3:
        state = dankai1
        print("一定時間入力がないため第一段階に戻ります")
    elif result == 3:
        print("ゴミ箱を開きます")
```

図 13 がこのコードのフローチャートである。

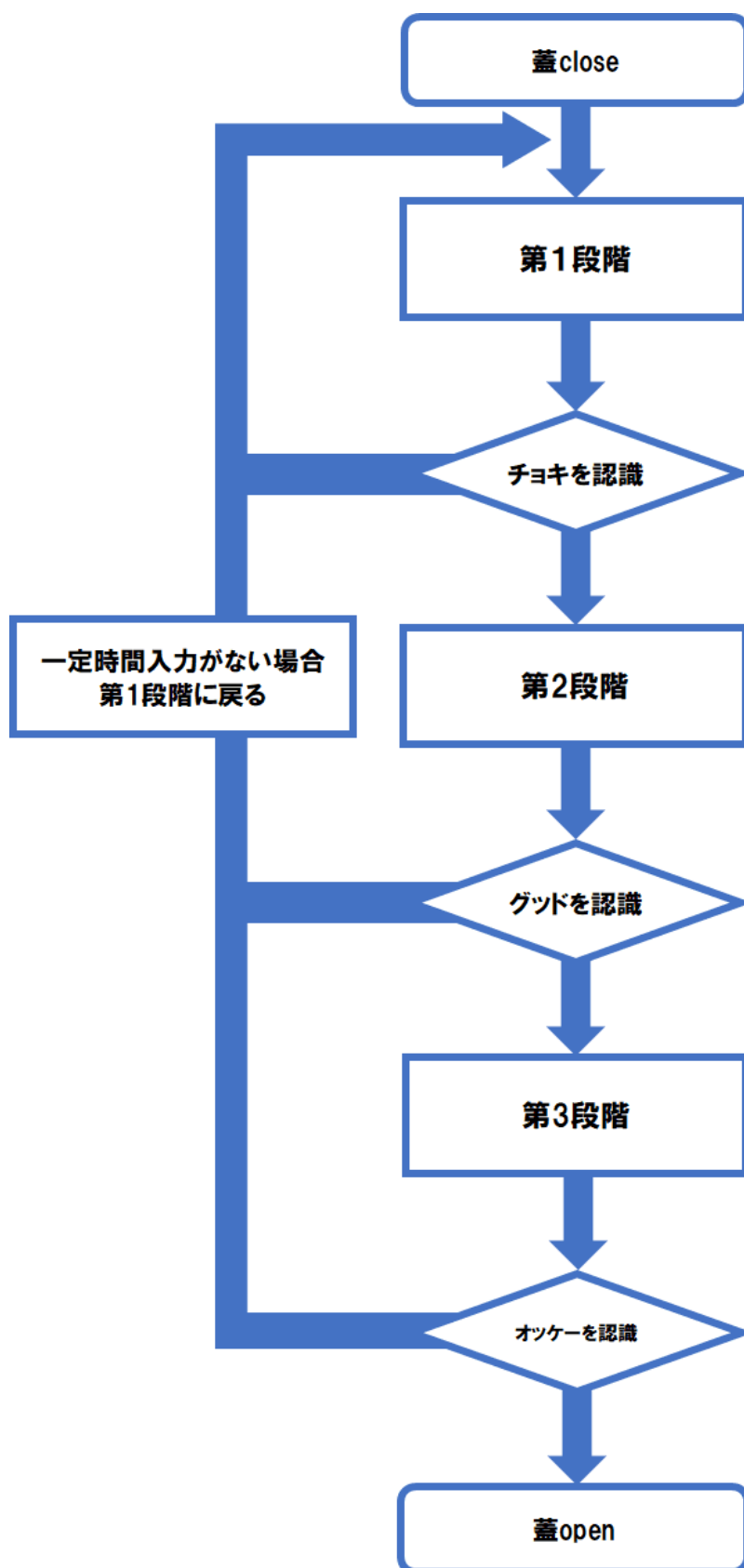


図 13 段階認識プログラムのフローチャート

## 第4章 ダストボックスのサイズとカメラの配置 (仙台担当)

本章では、2.1のシステムの一部であるダストボックスのサイズとカメラの配置について説明する。図14が想定しているダストボックスのイメージ図である。

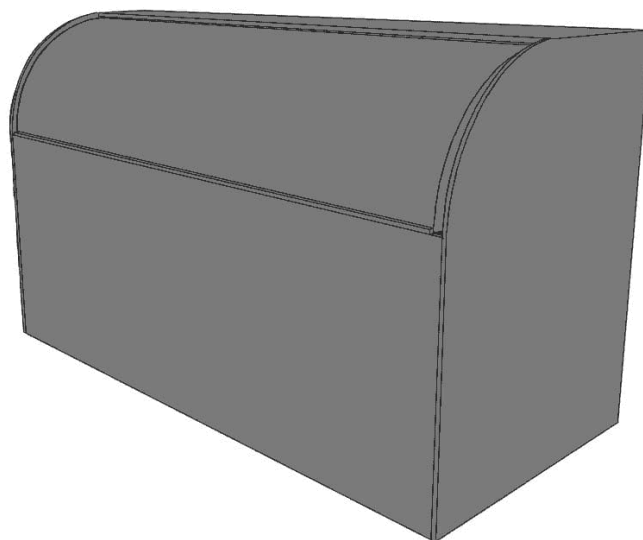


図14 使用を想定しているダストボックス

## 4.1 ダストボックスのサイズ決定

本節ではダストボックスのサイズについて説明する。

### 4.1.1 ダストボックスのサイズ

本項ではインターネット上[12]で取得したダストボックスのサイズについて説明する。本システムは集合住宅での使用を想定しているため、インターネット上にある10～20世帯用ダストボックスのサイズ表を参考に、合計13個のサイズデータを取得した。表5が取得したデータである。

表5 インターネット上で取得したサイズデータ表

	奥行 (mm)	高さ (mm)	横幅 (mm)	容量 (L)
1	900	1160	1950	1440
2	500	1160	1350	600
3	750	1160	1000	600
4	750	1160	1300	800
5	750	1160	1650	1000
6	750	1160	1950	1200
7	600	1160	1300	650
8	900	1160	1300	1000
9	600	1160	1650	850
10	900	1160	1650	1250
11	750	1060	600	315
12	750	1060	900	550
13	750	1060	1500	880
最大	900	1160	1950	1440
最小	500	1060	600	315

#### 4.1.2 最大サイズのダストボックス

本項では前節で取得したサイズデータをもとに決めたダストボックスの大きさについて説明する。取得したデータをもとに最大サイズを想定して図 15 のようなダストボックスに決定した。

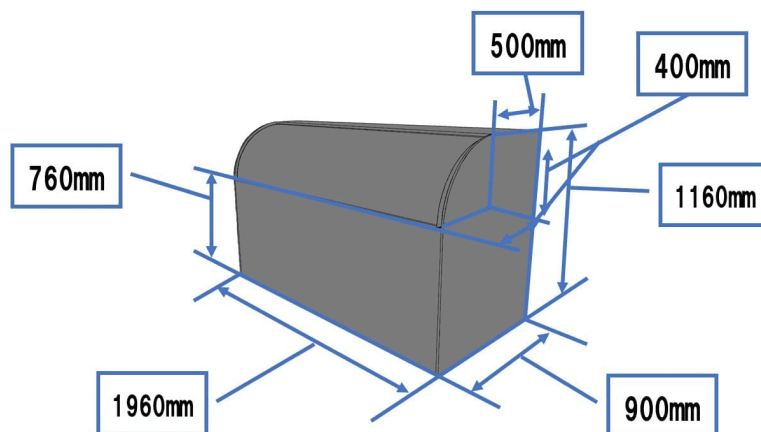


図 15 集めたデータの最大サイズから決定したダストボックスのサイズ

### 4.2 手認識用カメラの位置及び角度の実験

本節ではダストボックスに使用する手認識用カメラを設置する位置及び角度の実験について説明する。

#### 4.2.1 実験概要

手認識用カメラを設置する位置及び角度を決めるための実験を行う。使用者の身長、カメラへの映りやすさや手の出しやすさを考慮してカメラを設置しなければならない。

#### 4.2.2 実験準備

本項では手認識用カメラの位置及び角度の実験準備について説明する。手の映らない角度を除外するため、以下の流れで実験に使用する手の映りやすい角度を選定する。

1. 身長 176cm と身長 160cm の二人が両手を脱力した状態でカメラの前に立つ
2. カメラを手前側に設置した想定の高さ(76cm)と、奥側に設置した想定の高さ(116cm)で、 $0^{\circ}$  ~ $180^{\circ}$  の範囲内で  $10^{\circ}$  刻みにそれぞれ撮影する
3. 手前側に設置した場合の高さと奥側に設置した場合の高さで最も映りやすい3つの角度を選定する

実験において、本システムを使用する時にある程度の距離を取ることを想定して、使用者からダストボックス本体までの距離は 50cm とする。また、4.1.2 より、ダストボックスの手前側にカメラを設置した場合は高さ 76cm、ダストボックスの奥側にカメラを設置した場合は 116 cmとなる。

図 16～図 21 はカメラの設置位置によって、それぞれ手を認識しやすい選定した3つの角度での映り方の様子である。実験準備の結果、手前側にカメラを設置した場合は  $90^{\circ}$ 、 $100^{\circ}$ 、 $110^{\circ}$ 、奥側にカメラを設置した場合は  $70^{\circ}$ 、 $80^{\circ}$ 、 $90^{\circ}$  を実験に使用する。



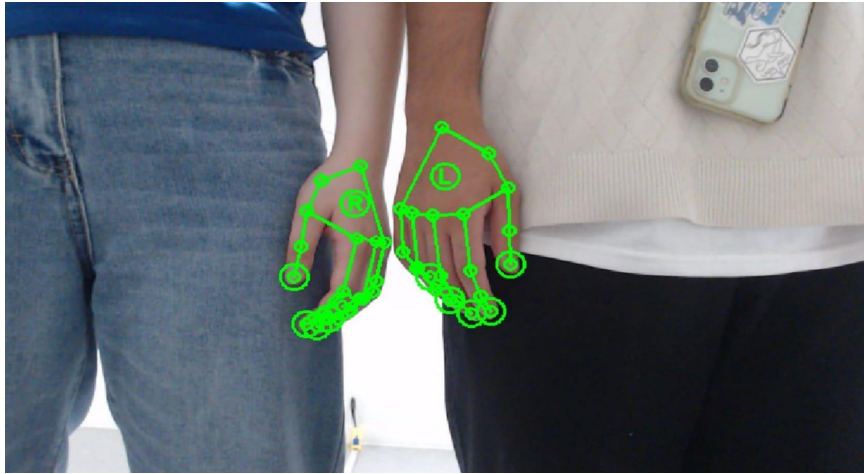


図 16 手前側に設置した想定の高さ 76cm 角度 90° の様子

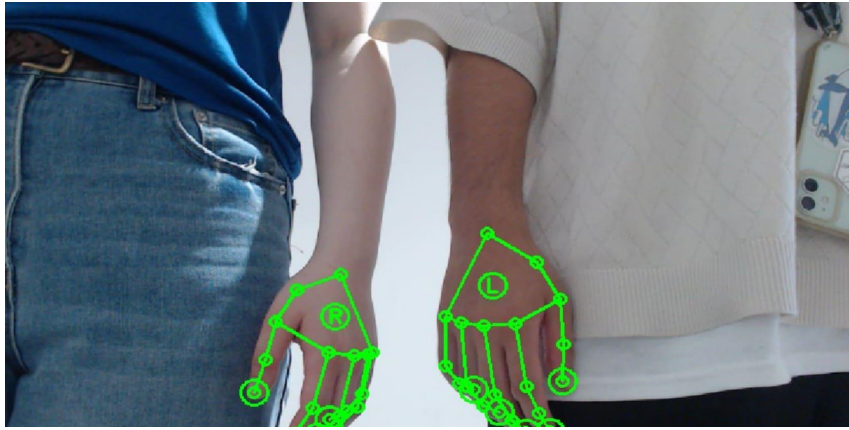


図 17 手前側に設置した想定の高さ 76cm 角度 100° の様子



図 18 手前側に設置した想定の高さ 76cm 角度 110° の様子

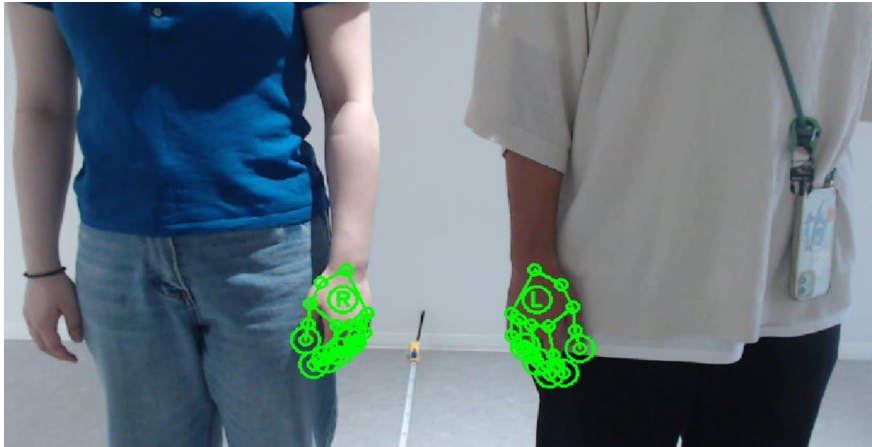


図 19 奥側に設置した想定の高さ 116cm 角度 70° の様子



図 20 奥側に設置した想定の高さ 116cm 角度 80° の様子

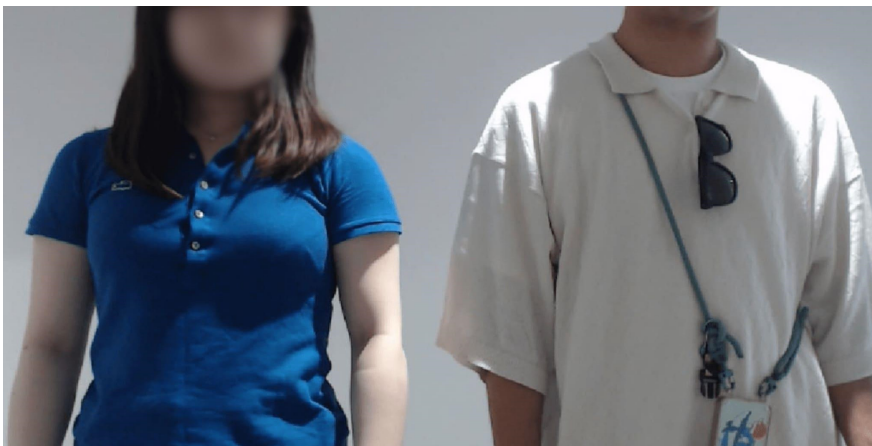


図 21 奥側に設置した想定の高さ 116cm 角度 90° の様子

### 4.2.3 実験方法

以下の流れで手認識用カメラの設置位置と角度を決める。

1. 4.2.2 と同様に、カメラを設置する高さは、ダストボックスの手前側に設置した場合は 76cm、奥側に設置した場合は 116cm とする。
2. 4.2.2 をもとに、設置角度はダストボックスの手前側に設置した想定では 90°、100°、110°、ダストボックスの奥側に設置した想定では 70°、80°、90° とする。
3. 段階認識を 100 回行い、最も認識精度の高いカメラの設置角度を決定する。

なお、100 回の各回の間は手をカメラに映らないようにし、画面上から手の座標が消えたことを確認してから次の段階認識を行う。認識成功の定義としては、第一段階に戻ることなく、段階認識が成功することとする。

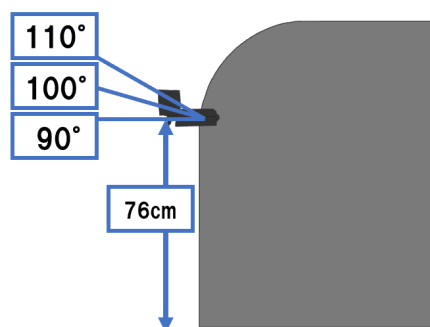


図 22 手前側に設置した場合

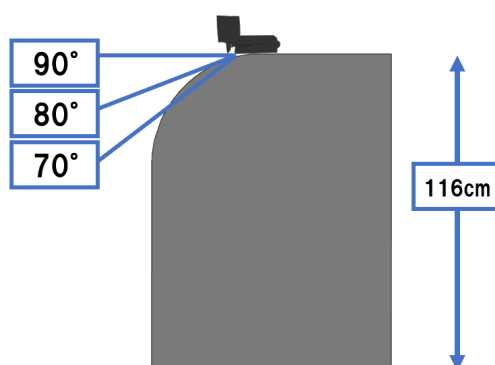


図 23 奥側に設置した場合

#### 4.2.4 実験結果

結果は図 24 及び図 25 のようになった。図 24 は手認識用カメラを手前側に設置した場合の認識成功回数、図 25 は奥側に設置した場合の認識成功回数である。最も認識成功回数が多かったのは手前側に 100° で設置した場合と、奥側に 80° で設置した場合の 99 回である。以上から手認識用カメラの設置位置は奥側の高さ 116cm、設置角度 80° を採用する。

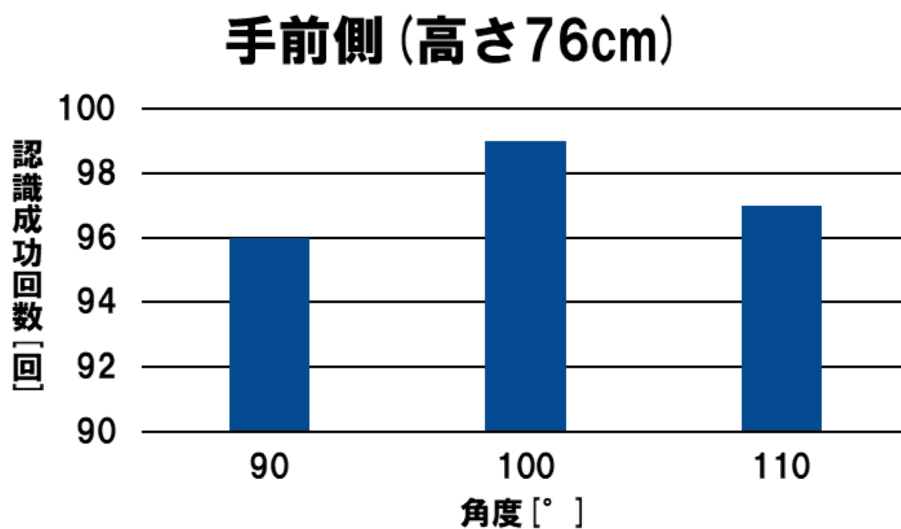


図 24 手前側に設置した場合

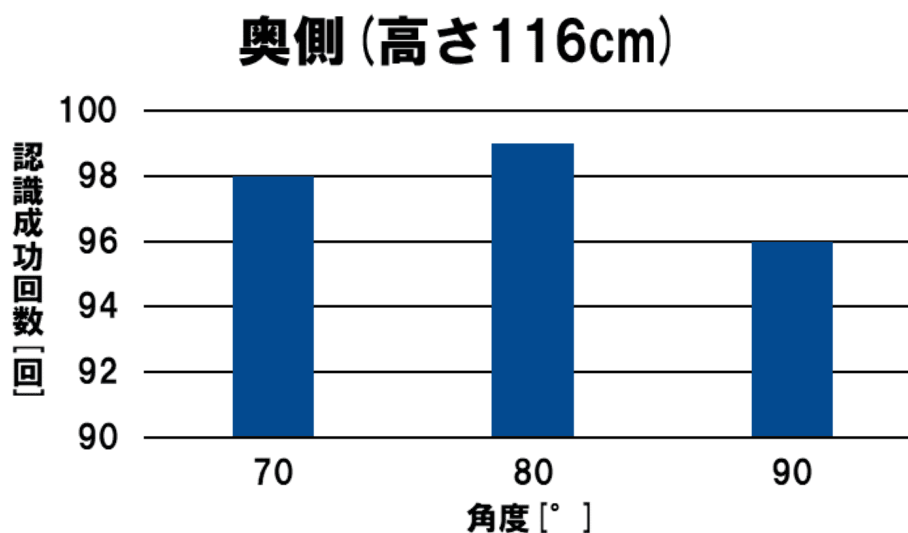


図 25 奥側に設置した場合

#### 4.2.5 実験考察

段階認識の成功回数が最も多かったのは、手前側の高さ 76cm 角度 100°、奥側の高さ 116cm 角度 80° に設置した場合である。どちらも成功回数は 100 回中 99 回と、実用可能な認識精度だといえる。以上から、手認識用カメラの設置位置として、奥側の高さ 116cm、角度 80° を採用した。

手前側ではなく奥側を採用した理由は、映像の範囲が広く、手が映りやすいからである。図 26 は手前側、図 27 は奥側に設置して撮影したものである。手前側に設置した時に比べ、奥側に設置した時の方が広範囲を映せていることが分かる。また、手前側に設置した場合、パターンを出すとカメラに対して手が垂直に映るため、カメラから隠れる手の部分が多く、パターンの認識までに時間がかかる。カメラに対して垂直に映らないようにパターンを出すにはしゃがむ必要がある。それに対して奥側に設置した場合は、必然的にカメラと手の距離をとることができるため、手が垂直に映ることはなく、しゃがむ必要もない。以上から、手認識用カメラの設置位置として、奥側の高さ 116cm、角度 80° を採用した。



図 26 カメラを手前側に設置した時の映像



図 27 カメラを奥側に設置した時の映像

## 第5章 ダストボックスの模型製作（佐々木担当）

ダストボックスを実寸大で製作することはコストや製作時間、置き場所などの点から現実的ではないことから 1/4 サイズの模型の製作を行うことにした。図 28 は 4.1.2 で決定したダストボックスの大きさを、1/4 にしたイメージ図である。

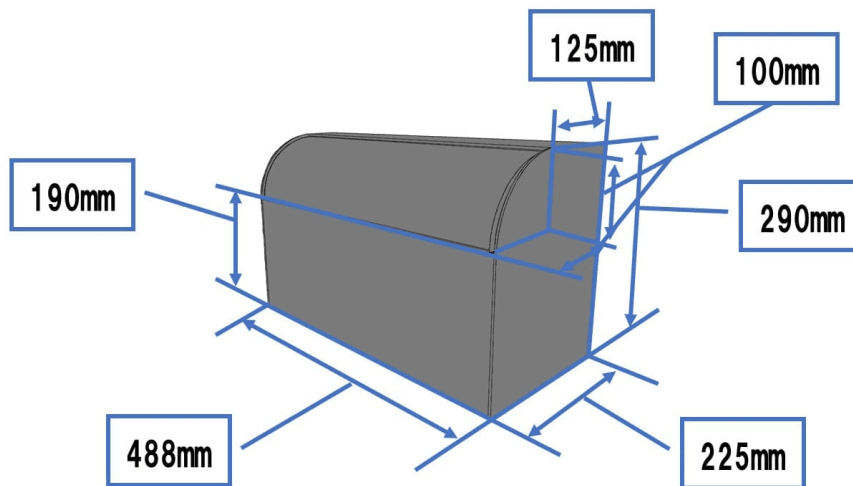


図 28 1/4 サイズのダストボックス

### 5.1 ダストボックス本体の製作

本節では、ダストボックス本体の製作についての説明を行う。

#### 5.1.1 本体の設計

設計において、本体を分解可能であることを目指した。その理由は、モーターやその他の機器をメンテナンスしやすくするためである。そのために、上下左右前後の 6 枚のアクリル板をねじによって固定することにした。図 29 が組み立て後の本体のイメージ、図 30 が分解した本体のイメージ、図 31 が本体の設計図である。

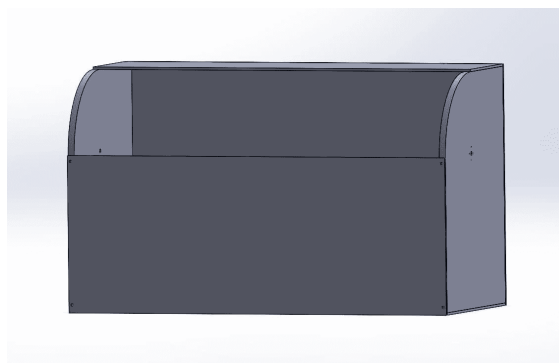


図 29 組み立て後の本体のイメージ

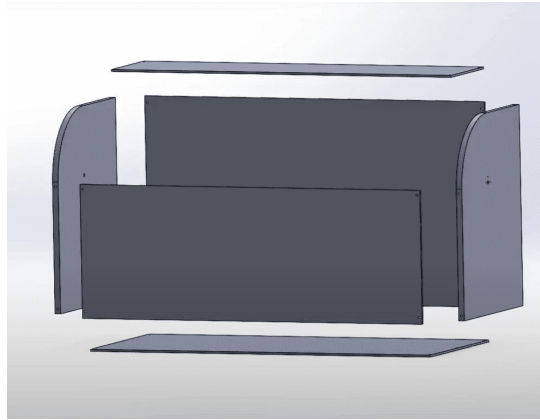


図 30 分解した本体のイメージ

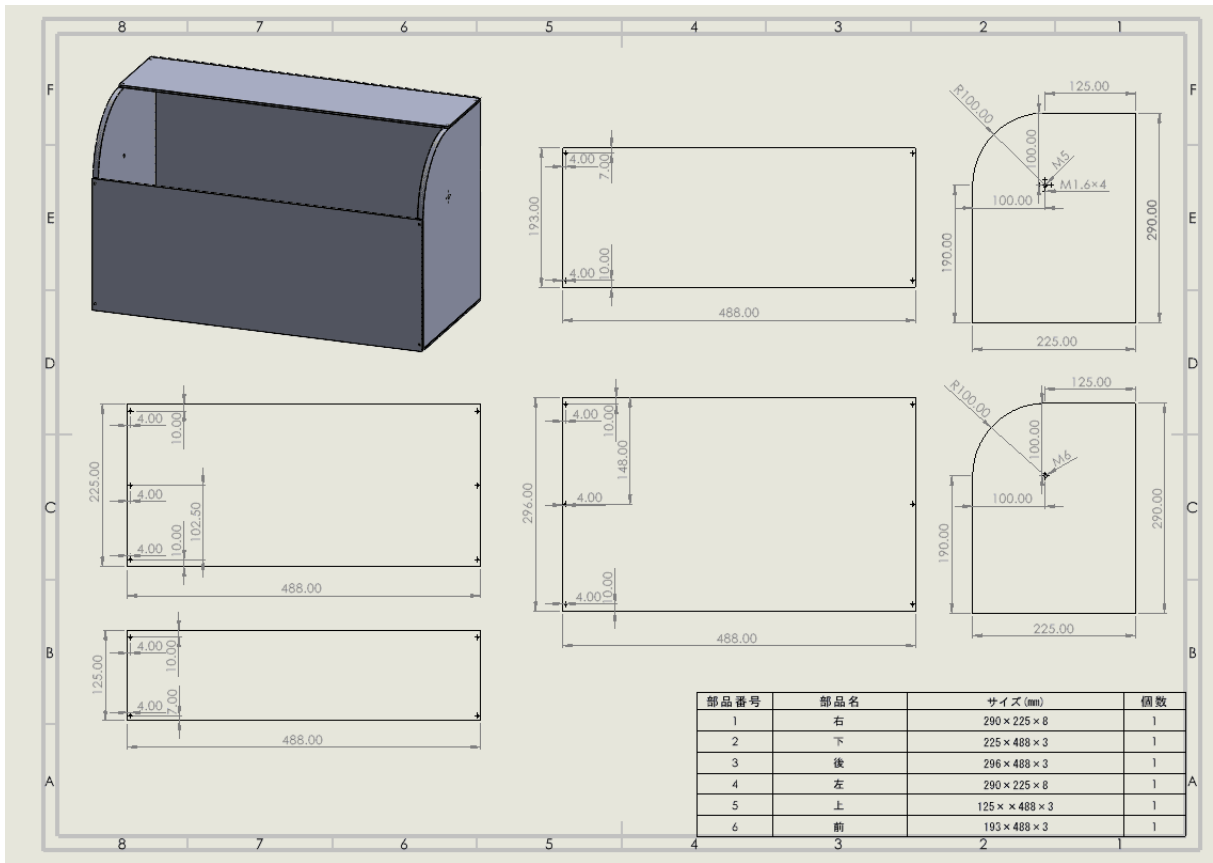


図 31 本体の設計図

アクリル板をねじで固定するためには、ねじ穴を切るための十分な厚みがなければならない。そのために、左右側面のアクリル板を厚くし、その板にねじ穴を切ることにした。図 32 がイメージ図である。

ねじ穴を切るためにアクリル板にどの程度の厚さが必要かを調べるため、厚さ 6mm、8mm、10mm の三種類のアクリル板を用意した。ねじ穴は M2、M3、M4 サイズを検討した。強度を保ちつつなるべく軽量である必要があるため、左右側面のアクリル板の厚さを 8mm とし、ねじは M3 を用いることにした。

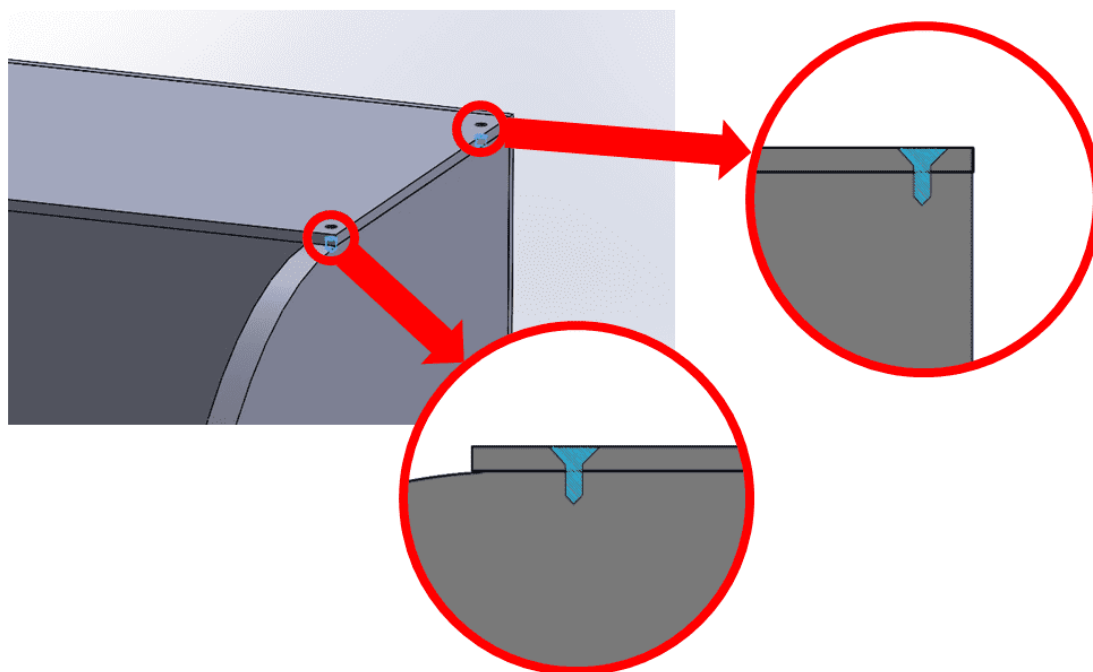


図 32 ねじ穴イメージ図



設計通りの大きさにカットされたアクリル板が図 33 である。アクリル板はアクリルオンライン[13]でカット及び穴あけ済みのものを購入した。購入時にサイズや厚みは1mm単位で指定している。また、ねじを通すための穴は穴あけとして 3mm の丸穴を、左右側面の円弧は角の丸めとして半径 100mm の丸めを指定している。



図 33 カット及び穴あけ済みのアクリル板

さらに届いた左右側面のアクリル板にねじ穴開けの加工を行う。この加工については次項で説明する。

### 5.1.2 アクリル板のねじ穴開け加工

左右側面のアクリル板へのねじ穴の加工は、工学院大学八王子キャンパス 19 号館ふらっと[14](図 34)で行った。加工にはボール盤(図 35)を使用した。5.1.1 の設計をもとに、ボール盤で M2.4 の下穴を開け、M3 のタップ切りを行い、ねじ穴開けが完了した。



図 34 工学院大学八王子キャンパス 19 号館



図 35 ボール盤

### 5.1.3 ダストボックス本体の完成

図36がダストボックス本体の完成したものである。これでダストボックス本体は完成とした。



図36 組み立てたダストボックス

## 5.2 ダストボックスの蓋の製作

本節では、ダストボックス蓋の製作についての説明を行う。

### 5.2.1 蓋の設計

蓋の設計にあたり、本体内部に広い空間を持ちゴミをなるべく多く収容できることを目指した。設計には3次元ソフトウェアであるSOLIDWORKS[15]を使用した。

蓋の開閉方式には、外開き、内開き、スライド式等がある。外開きの蓋は、自動で開いた際に使用者と当たる危険性がある。内開きの蓋は、中にゴミが多く入っていると開かなくなるため、収容量が少なくなってしまう。スライド式の蓋は、使用者に当たる危険性がなく、本体に沿って蓋が収納されるため内部に広く空間を持たせることができるというメリットもある。この特徴を踏まえて、蓋の開閉方式はスライド式を採用した。

蓋の形状として、モーターの回転運動を活かすため、当初は、四分円柱型を採用していた。しかし、四分円柱型の扇形の内部にモーターを取り付けると扇形の中心点とモーターの回転軸がずれてしまい、蓋を開けたときに本体と沿わなくなってしまう。そこで、図37のようにモーターの取り付け部分を増やし、円の中心点にモーターを取り付けられるよう改良した。

以上より図 38 の形状に決定した。図 39 がモーター取り付けのイメージ図、図 40 が蓋の設計図である。

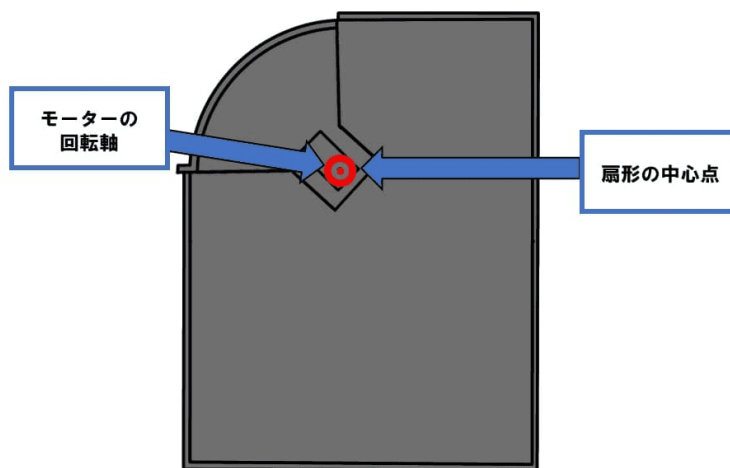


図 37 改良型モーター取り付け図

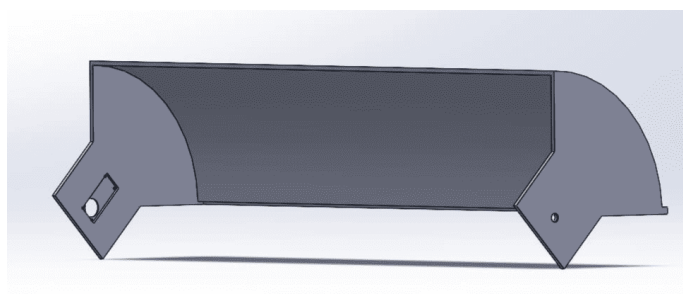


図 38 蓋のイメージ図

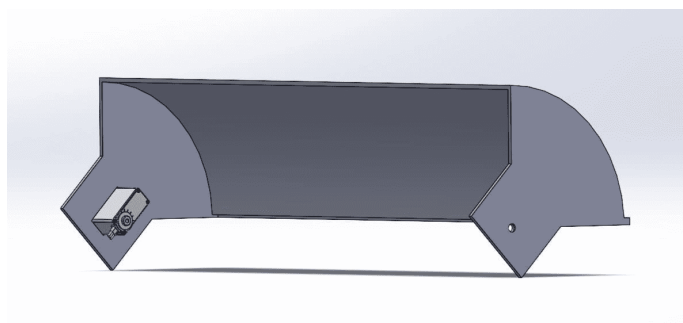


図 39 モーター取り付けのイメージ図

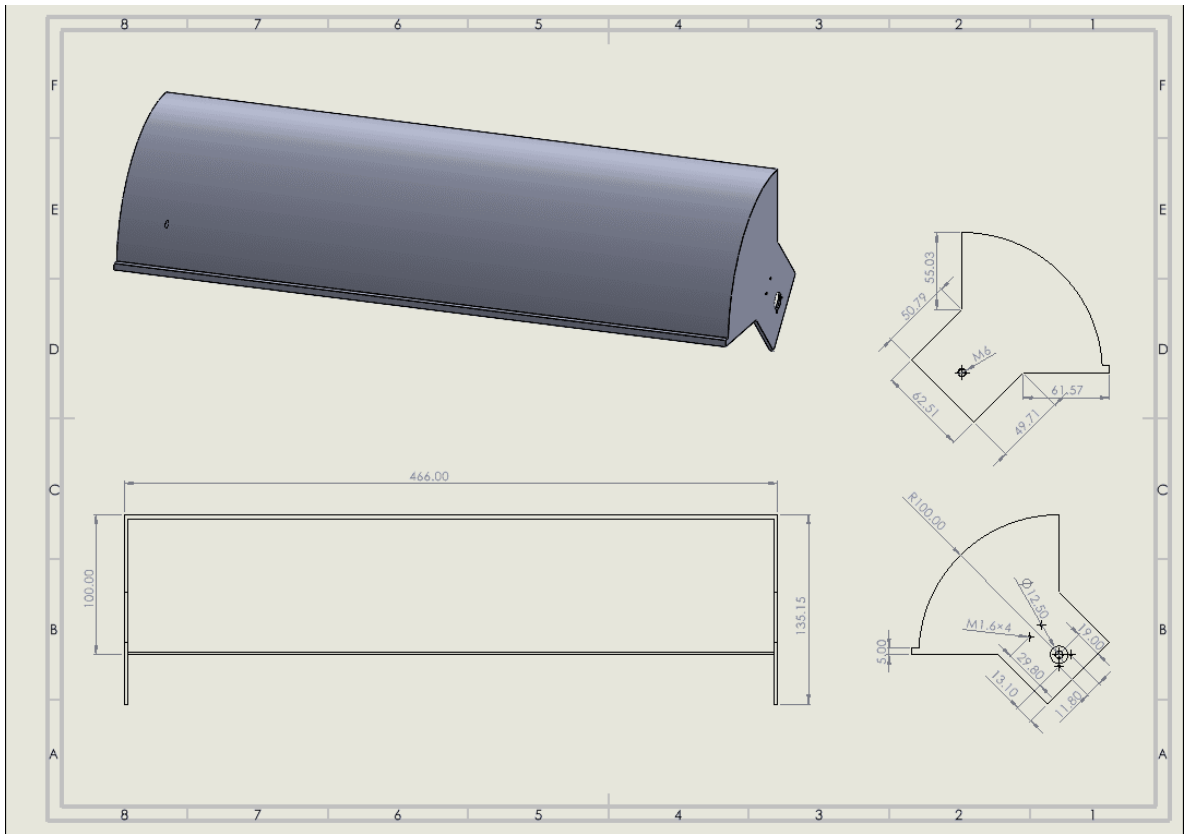


図 40 蓋の設計図

### 5.2.2 ダストボックスの蓋の完成

図 40 からわかるように、蓋には曲線部分やモーターの取り付け穴などがあり、複雑な形状をしている。そのため、3D プリンターでの製作を決定した。3D プリンターは DMM.make3D プリント[16]を利用し、5.2.1 で作成した CAD データを STEP 形式で出力して注文した。素材はエコノミープロレジン製、造形方法は SLA(光造形)を使用した。

図 41 が実際に 3D プリンターで製作した蓋である。これでダストボックスの蓋は完成とした。



図 41 3D プリンターで製作した蓋

## 5.3 ダストボックスの模型の完成

### 5.3.1 ダストボックスの組み立て

5.1、5.2 を経てダストボックスの模型の本体部分と蓋部分が完成した。図 42 が本体と蓋を組み立てたダストボックスである。



図 42 組み立て後のダストボックス

本体部分と蓋部分を組み立てたが、蓋を開けることができなかった。原因は二点挙げられる。一点目に、本体上面部分の亚克力板が下方向にたわんでしまったことである。上面の亚克力板は四隅がねじで止められているだけで、中心部分が浮いてしまい、自重に耐えられずたわんでしまった。二点目に、蓋部分が歪みにより膨らんでしまったことである。3D プリンターによる製作物には、薄くて長い形状は歪みやすいという特性がある。製作した蓋も厚さ 3mm と薄く、488mm と長い形状であるため、歪んでしまったと考える。以上が原因で、亚克力板と蓋が干渉し、蓋が開かなくなってしまった。

この問題を改善するために、ダストボックス本体を修正することにした。詳細については次項で説明する。

### 5.3.2 ダストボックスの模型の修正と完成

5.3.1 でダストボックスの模型が完成したものの、たわみと歪みの影響で蓋を開けることができなかった。そこで、本項では修正した部分についての説明を行う。

一点目に、本体の上面の設計を変更した。修正前では上面部分の亚克力板は四隅をねじで止められているだけで、中心部分が浮いてしまい、自重に耐えられずたわんでしまった。そこで、この亚克力板を背面方向に 3mm 大きいものに変更した。よって、上面部分が背面の亚克力板に乗る形になり、たわまなくなった。

二点目に、ダストボックスの側面の高さを 10mm 高くした。3D プリンターで用いた材質の特性上、蓋の歪みは直すことが難しい。そこで、ダストボックス側面の高さを上げることで蓋が歪んだままでも開くようにした。変更点としては、背面の亚克力板を上面方向に 7mm 大きいものに変更し、ダストボックスの側面に高さ 10mm、厚さ 8mm の亚克力板を追加した。よって、ダストボックス全体の高さを上げることに成功した。

完成したものが図 43 である。これでダストボックスの模型は完成とした。



図 43 完成したダストボックス



## 第 6 章 模型の完成 (渡辺担当)

### 6.1 模型の動作に必要なハードウェア

本節では、模型の動作に必要なハードウェアについての説明を行う。

#### 6.1.1 Arduino

モーター制御をするにあたり、Arduino UNO R3[17]というマイコンボードを使用した。Arduino とは、電子工作などでよく使用されるマイコンボードのひとつである。本研究では蓋の開閉を行うモーターの制御のために用いる。



図 44 Arduino UNO R3

#### 6.1.2 サーボモーター

本研究では、角度制御が容易なサーボモーターでダストボックスの蓋の開閉を行う。図 45 は近藤科学社製[18]のサーボモーター KRS-3204R2 ICS である。サーボモーターはマイコンにより角度を指定して制御できるのが特徴である。またサーボモーターの角度が一定に保たれているとき、保持力が働くため手動で動かすことが困難である。これはダストボックスの蓋を手動で開けることができないということを意味する。詳しい仕様を表 6 に示す。



図 45 サーボモーター KRS-3204R2 ICS

表 6 サーボモーター仕様

品番	KRS-3204R2 ICS
電源電圧	6~7.4V
最大トルク(6V)/(7.4V)	(7.5kgf・cm) / (9.3kgf・cm)
最高スピード(6V)/(7.4V)	(0.18s/60°) / (0.13s/60°)
寸法	33×15×26mm (突起部除く)
重量	27.2g
最大動作角度	270° (±135°)
通信規格	ICS3.6 (シリアル/PWM 選択式)
通信速度	115200/625000/1250000bps

### 6.1.3 電源

モーターを制御している Arduino からの電力供給は 5V である。しかし今回使用するサーボモーターは、6~7.4V での動作を推奨されているため、外部電源として図 46 のサンハヤト株式会社製の電源キット DK-801[19]を用いることにした。詳しい仕様を表 7 に示す。



図 46 電源キット DK-801[19]

表 7 外部電源の仕様

型番	DK-801
入力電圧	AC100V / max.50W
出力電圧	DC3V.5V.6V.9V / 最大 1.5A (連続 1.2A)
出力電圧	DC12V.15V / 最大 1.0A (連続 0.7A)
出力電圧交差	±0.5V (1A 時)

## 6.2 プログラム

図 47 は、手の段階認識後に、Arduino へと信号を送りモーターを制御するまでの流れである。この流れの実現方法を以下に記す。

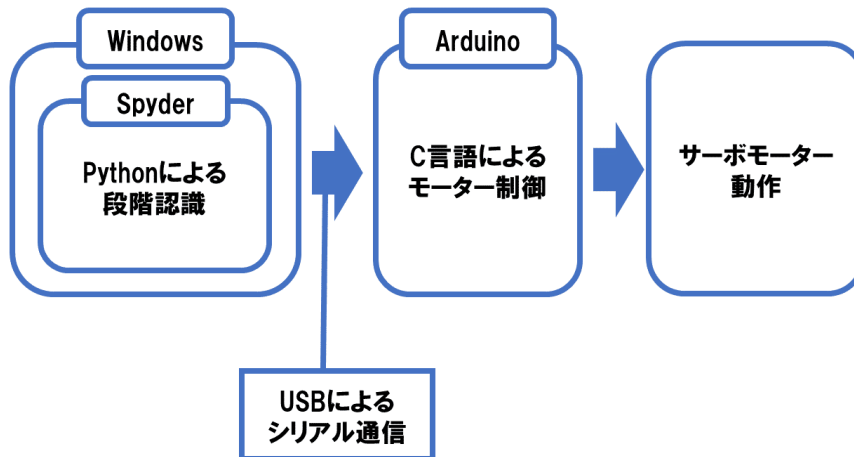


図 47 プログラムの流れ

### 6.2.1 Arduino シミュレーション

サーボモーターの角度を制御する方法の確認のため、Tinkercad[20]でシミュレーションを行った。Tinkercad とは、電子回路のシミュレーションを行うことができる Web アプリである。Tinkercad のシリアルモニタは、図 47 のシリアル通信による信号の送受信を模擬している。

そのシリアルモニタに指定の角度を入力すると、図 48 のようにサーボモーターが回転することを確認した。そのために Arduino 上で動作させたプログラムを 6.2.2 に掲載する。

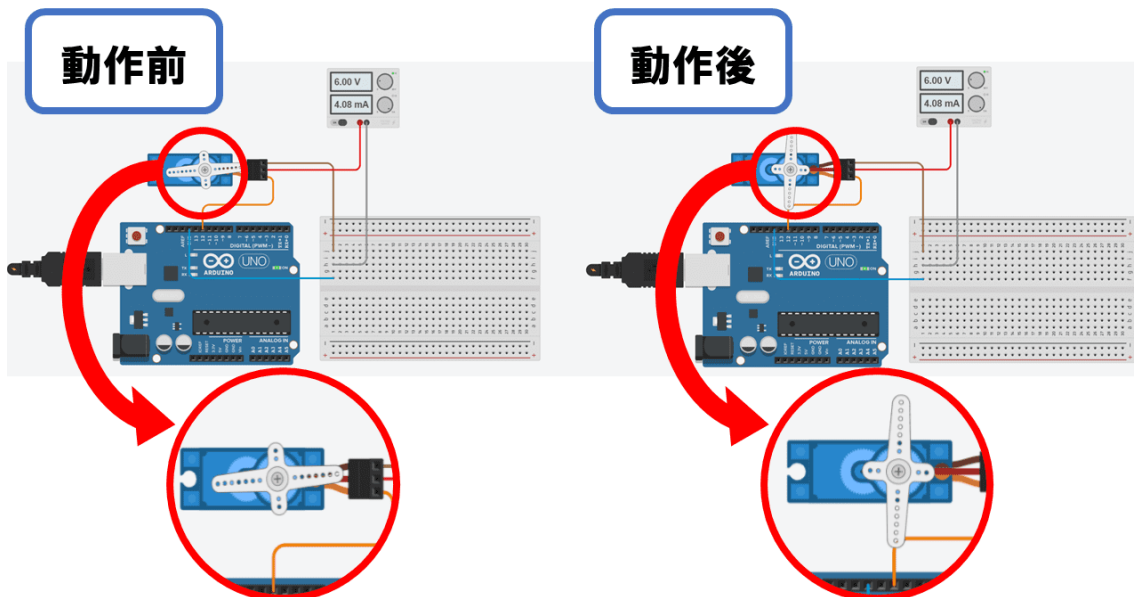


図 48 Tinkercad での回路シミュレーション

### 6.2.2 Arduino 側のプログラム

Arduino では、シリアル通信により送られてきた信号を受け取りサーボモーターの回転制御を行う。

以下がそのプログラムである。

```
#include <Servo.h>
Servo motor;

//Arduino のピンの番号を MOTOR で定義
#define MOTOR 12

//変数定義,最初の角度
int rot=90;
int x;

//初期化の処理
void setup()
{
  Serial.begin(9600);
  motor.attach(MOTOR);
  motor.write(rot);
}

//ループ処理
void loop(){
  if (Serial.available() > 0)
  {
    delay(20);
    x = Serial.parseInt();

//入力角度が 0 から 180 なら動く
    if(x >= 0 && x <= 180)
    {
      motor.write(x);
      deg(x);
    }
  }
}
```

```

}
//数値の表示
void deg(int x){
  Serial.print("Input: ");
  Serial.print(x);
  Serial.println(" deg");
}

```

### 6.2.3 Python 側のプログラム

Python では、手の段階認識が成功したとき、Arduino へ蓋が開閉するための信号の送信を行う。

以下がそのプログラムである。

```

//第一段階の時チヨキを認識すると第二段階へ
if state == dankai1:
  if result == 1:
    t1 = time.time()
    state = dankai2
    print("第二段階")

//第二段階に移行後、3秒以内にグッドが認識された場合第三段階へ
elif state == dankai2:
  t2 = time.time()
  if (t2 - t1) > 3:
    state = dankai1
    print("一定時間入力がないため第一段階に戻ります")

elif result == 2:
  t3 = time.time()
  state = dankai3
  print("第三段階")

//第三段階に移行後、3秒以内にOKが認識された場合Arduinoへ信号を送信
elif state == dankai3:
  t4 = time.time()
  if (t4 - t3) > 3:
    state = dankai1

```

```
print("一定時間入力がないため第一段階に戻ります")

elif result == 3:
    print("ゴミ箱を開きます")
    //開いた状態の 0° に指定する
    ser.write(b"0")
    //5 秒待つ
    time.sleep(5)
    //閉じた状態の 90° に指定する
    ser.write(b"90")
```

### 6.3 開閉機能の動作確認

6.2.2 と 6.2.3 のプログラムにより、手の段階認識後にサーボモーターが動作するかの確認を行った。Arduino とサーボモーターをパソコンに接続し、動作させた様子が図 49～図 54 である。3 段階の段階認識後に、サーボモーターが 90 度回転し、その後 5 秒経過してからサーボモーターが初期状態に戻ることを確認できた。

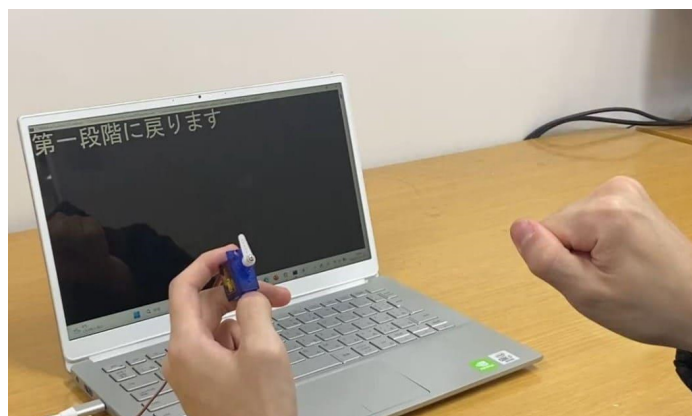


図 49 初期段階



図 50 チョキを認識



図 51 グッドを認識





図 52 オッケーを認識

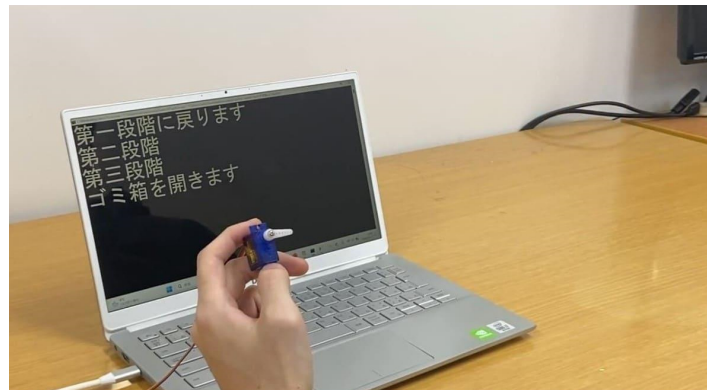


図 53 モーターが回転

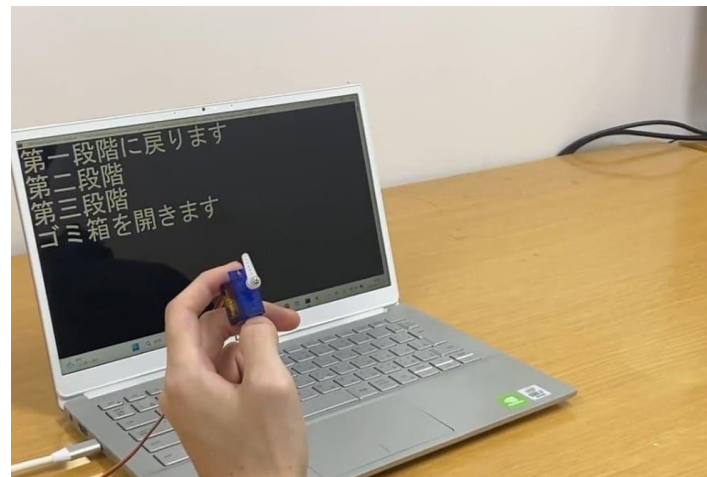


図 54 5 秒後に初期段階へ戻る

## 6.4 ダストボックスと開閉機構の接続

5.3.2 で組み立てたダストボックスにカメラとサーボモーターを取り付け、ダストボックスを完成させた様子が図 55 である。



図 55 全ての機器を接続し、完成したダストボックス

## 6.5 手認識による集合住宅用ダストボックスの自動開閉機能の完成

6.4でダストボックスと開閉機構を接続し、模型が完成した。動作確認として、実際に段階認識と自動開閉を行った様子が図 56～図 58 である。



図 56 チョキを認識



図 57 グッドを認識



図 58 オッケーを認識

段階認識後にパソコンから信号が出され、信号を受け取った Arduino がモーターを回転させ、ダストボックスの蓋が開き、一定時間経過後に蓋が閉まった。

これで手認識による集合住宅用ダストボックスの自動開閉機能は完成した。

## 第7章 結論 (藤枝担当)

本研究では、非接触で解錠と開閉を行えるダストボックスを開発した。本システムは、手認識のシステム、ダストボックス、モーターを用いた開閉機構の三点で構成される。

手認識のシステムについては、段階認識プログラムの開発により、手のパターンを判別し認識できた。また、三段階の認識を組み合わせることで非接触でのセキュリティを確立することができた。

ダストボックスについては、アクリル板で本体部分を、3D プリンターで蓋部分を製作した。これらを組み合わせ、軽量かつ容量が大きいダストボックスが完成した。

モーターを用いた開閉機構については、Arduino を用いてサーボモーターを制御することで実現した。また、製作したダストボックスの模型に組み込むことで、ダストボックスの蓋を自動開閉させることができた。よって、モーターを用いた開閉機構は完成した。

以上から、手認識による集合住宅用ダストボックスの自動開閉機能が完成した。

## 参考文献・URL

[1]手をかざしてシュッと消毒「tette」 | KINGJIM

<https://www.kingjim.co.jp/sp/tette/>

(最終閲覧日：2024/01/30)

[2]ゴミ箱 ペダル | ニトリネット【公式】 家具・インテリア通販

<https://www.nitori->

[net.jp/ec/keyword/%E3%82%B4%E3%83%9F%E7%AE%B1%E3%80%80%E3%83%9A%E3%83%80%E3%83%AB/](https://www.nitori-net.jp/ec/keyword/%E3%82%B4%E3%83%9F%E7%AE%B1%E3%80%80%E3%83%9A%E3%83%80%E3%83%AB/)

(最終閲覧日：2024/01/30)

[3]東京都環境局-プロジェクトチーム報告書

カラス対策プロジェクトチーム報告書～東京のカラス問題を解決するために～

[https://www.kankyo.metro.tokyo.lg.jp/nature/animals\\_plants/crow/project\\_report.html](https://www.kankyo.metro.tokyo.lg.jp/nature/animals_plants/crow/project_report.html)

(最終閲覧日：2024/01/31)

[4]顔世荀, 池勇勳, 梅田和昇

任意の位置での簡単なジェスチャによる家電操作システム

日本機械学会論文集, 87 巻 898 号 (2021)20-00310

[https://www.jstage.jst.go.jp/article/transjsme/87/898/87\\_20-00310/\\_pdf/-char/ja](https://www.jstage.jst.go.jp/article/transjsme/87/898/87_20-00310/_pdf/-char/ja)

(最終閲覧日：2024/02/01)

[5]株式会社ロジクール

C920n HD PRO ウェブカメラ

<https://www.logicool.co.jp/ja-jp/products/webcams/hd-pro-webcam-c920n.960-001261.html>

(最終閲覧日：2024/01/31)

[6]Dell-Technologies

[https://www.dell.com/support/manuals/ja-jp/inspiron-13-7391-laptop/inspiron-7391-](https://www.dell.com/support/manuals/ja-jp/inspiron-13-7391-laptop/inspiron-7391-setup-and-specifications/inspiron-7391%E3%81%AE%E5%9B%B3?guid=guid-d3cb1479-5af9-4e2f-9a8b-416fc88c775d&lang=ja-jp)

[setup-and-specifications/inspiron-7391%E3%81%AE%E5%9B%B3?guid=guid-d3cb1479-5af9-4e2f-9a8b-416fc88c775d&lang=ja-jp](https://www.dell.com/support/manuals/ja-jp/inspiron-13-7391-laptop/inspiron-7391-setup-and-specifications/inspiron-7391%E3%81%AE%E5%9B%B3?guid=guid-d3cb1479-5af9-4e2f-9a8b-416fc88c775d&lang=ja-jp)

(最終閲覧日：2024/01/31)

[7]Anaconda

<https://www.anaconda.com/>

(最終閲覧日：2024/01/31)

[8]MediaPipe

<https://developers.google.com/mediapipe>

(最終閲覧日：2024/01/31)

[9]MediaPipe-Solutions-Examples

<https://developers.google.com/mediapipe/solutions/examples>

(最終閲覧日：2024/01/31)

[10]MediaPipeHands

[https://developers.google.com/mediapipe/solutions/vision/hand\\_landmarker](https://developers.google.com/mediapipe/solutions/vision/hand_landmarker)

(最終閲覧日：2024/01/31)

[11]Kazuhito00-github

<https://github.com/kazuhito00/mediapipe-python-sample/>

(最終閲覧日：2024/02/06)

[12]株式会社ダイケン

<https://www.daiken.ne.jp/>

(最終閲覧日：2024/01/31)

[13]アクリルオンライン

<https://acrylic-online.com/>

(最終閲覧日：2024/01/31)

[14]工学院大学-ものづくり支援センター-ふらっと

[https://www.kogakuin.ac.jp/student/campuslife/manufacture\\_sc.html](https://www.kogakuin.ac.jp/student/campuslife/manufacture_sc.html)

(最終閲覧日：2024/01/31)

[15]SOLIDWORKS

<https://www.solidworks.com/ja>

(最終閲覧日：2024/02/06)

[16]DMM.make 3D PRINT

[https://lp.make.dmm.com/mass/a\\_ad/?utm\\_term=dmm+3d&utm\\_campaign=00-brand&utm\\_source=adwords&utm\\_medium=ppc&hsa\\_acc=7414407713&hsa\\_cam=16747407991&hsa\\_grp=138126501074&hsa\\_ad=659157970012&hsa\\_src=g&hsa\\_tgt=kwd-454008999539&hsa\\_kw=dmm+3d&hsa\\_mt=e&hsa\\_net=adwords&hsa\\_ver=3&gad\\_source=1&gclid=EAIaIQobChMI1YW-l8mJhAMVPM0WBR1W5QO1EAAAYASAAEgJPDPD\\_BwE](https://lp.make.dmm.com/mass/a_ad/?utm_term=dmm+3d&utm_campaign=00-brand&utm_source=adwords&utm_medium=ppc&hsa_acc=7414407713&hsa_cam=16747407991&hsa_grp=138126501074&hsa_ad=659157970012&hsa_src=g&hsa_tgt=kwd-454008999539&hsa_kw=dmm+3d&hsa_mt=e&hsa_net=adwords&hsa_ver=3&gad_source=1&gclid=EAIaIQobChMI1YW-l8mJhAMVPM0WBR1W5QO1EAAAYASAAEgJPDPD_BwE)

(最終閲覧日：2024/02/01)

[17]Arduino

<https://www.switch-science.com/products/789>

(最終閲覧日：2024/02/01)

[18]近藤科学

<https://kondo-robot.com/>

(最終閲覧日：2024/02/01)

[19]サンハヤト株式会社

<https://www.sunhayato.co.jp/>

(最終閲覧日：2024/01/31)

[20]Tinkercad

<https://www.tinkercad.com/>

(最終閲覧日：2024/01/31)

## 謝辞

本研究の遂行にあたり、2年間多大なご指導を賜った金丸隆志教授、工作機械の利用において多くの助言を頂いたふらっとの皆様方、協力していただいた研究室の先輩方、同級生に班一同、心より感謝いたします。