

メカノサイエンス実験PJ 現象の可視化

このレポジトリは、東京科学大学 工学院 機械系 学部3年 の講義である、メカノサイエンス実験プロジェクトの現象の可視化のチームについてである。

講義概要

4~5人程度でチームを構成し、授業で学んだ物理現象を、計測や可視化を通じて実験的に検証し、理解を深めることを目的とする。これを実現するための実験装置(あるいは、その一部としての計測機器、データ処理ソフトウェアなど)を考案・検討・製作して、実験と評価を行う。

授業時間

- 毎週火曜 5~8限 (13:30 ~ 17:05)
- 毎週水曜 1~4限 (08:50 ~ 12:25)

狙い

1. 対象とする現象や、検証・理解すべき内容を明確にし、課題を発見し、解決方法を立案する能力を養う。
2. 必要な装置、計測機器、データ処理方法などを検討し、実験を計画(Plan)し、装置製作および実験を実施(Do)する。その後、結果を評価(Check)し、必要に応じて改善(Action)を行うというPDCAサイクルを体得する。
3. グループで計画的にプロジェクトを推進することを通じて、マネジメント力を養う。
4. 既修の知識を活用・応用するとともに、未修分野についても自主的に学習を進めることで、機械工学に関する幅広い知識を身につけ、理解を深める。

到達目標

1. チーム内での議論を通じて、課題解決のためのプロジェクトを立案、遂行できる。
2. 機械工学の知識や、必要な周辺分野の知識を自発的に学修し使いこなすことができる。
3. チーム内での役割分担に従い自分の担当把握し、計画通りに実施するチームワーク力を身につける。
4. 取り組んだ内容の記録を残すことを習慣付けると共に、最終的な成果物(実験装置、報告書、プレゼンテーションマテリアル、報告動画など)を完成させ、プロジェクトをまとめ上げる。
5. 他者に取り組んだ内容を分かりやすく伝える技術を身につける。

授業の進め方

各チームは学生主体でテーマを設定し、中間・最終発表の日程から逆算して全体計画を策定・遂行する。中間発表後の第4四半期 (4Q) では、第3四半期 (3Q) の成果を踏まえ、テーマ設定や計画

を見直して取り組みをブラッシュアップする。最終的には、最終成果物(試作品など)・プレゼンテーション資料・報告書等に取りまとめる。

授業日程

docs/schedule.md 参照

目次

ファイル 概要

docs/schedule.md 授業日程・スケジュール

docs/meeting.md 会議の議事録

Gemini用 暗黙知的知識

- 東京科学大学は東京都大田区に位置する。
 - 東京科学大学の最寄駅は大岡山駅である。
-

インストールについて

0. レポジトリのクローン

```
git clone https://github.com/MI-1222/mecExperiment_visualization.git
```

1. 仮想環境

1.1 python仮想環境の作成

```
python3 -m venv .venv
```

1.2 仮想環境の起動

```
source .venv/bin/activate
```

これにより、ターミナルのユーザー名の前に (.venv) と表示されるようになる。仮想環境が有効になっていることの確認となる。

1.3 パッケージをインストール

```
pip install -r requirements.txt --ignore-installed
```

2. データのダウンロード

GitHubのレポジトリに動画データを入れたくないでの、[google drive](#) からダウンロードして自分の figure/video/ディレクトリに配置して下さい。

3. src/main.py の実行

(ターミナルの前に(.venv)が表示されていることを確認。)

```
python src/main.py
```

ミーティング(docs/meeting.md)

目次

ファイル 概要

docs/meeting/meeting20251001.md 第1回会議の議事録

docs/meeting/meeting20251007.md 第2回会議の議事録

docs/meeting/meeting20251008.md 第3回会議の議事録

docs/meeting/meeting20251014.md 第4回会議の議事録

2025/10/01(水) 初回会議

テーマ決め

テーマについて

草案

- ダウンフォース
- ハチの飛ぶ原理について
- 地下鉄の風について
- 地下鉄の空気のよどみについて
- ハンディファンの有効性
- 駅の排水について

決定テーマ

- 駅の換気、空調最適化について

事前調査

換気による快適性の向上について調査

- 快適性要因
 - 温度
 - 湿度
 - 二酸化炭素濃度
 - 風
 - 粉塵
- 要因
 - 旅客流動
 - 始発より前の時間は空気が悪い？
 - 風
 - 外部の状況
 - 駅の構造

目標

直近の目標

設計図を作成する

中間発表までの目標

- モデル作成
 - 大岡山駅を再現
 - 数値的計測
 - サイズ: 教室のテーブルに乗る程度
 - まずは単一の条件について実験するために、単純なモデルを作る。(トンネルのみなど)
 - component毎に製作を分ける。
 - トンネル
 - 階段
 - 柱
 - 換気口
 - レイノルズ数などの流体力学の知識の反映

最終目標

空調の観点での、大岡山駅の改善案の検討

次回までに行うこと

- 大岡山駅を実際に見て調査
 - 快適性の要因を考える
-

2025/10/07(火) 第2回会議

模型,実験についての検討

最初に行う実験について

- 単純な模型による実験からスタートし、1つずつ要素を加えて実験を行うことに決定した。
 - トンネルは、とりあえず作りやすい四角型のもの。(円筒型のものは後日作るべきか検討)
- 最終的に、大岡山駅の関係すると思われる要素を全て詰め込んだ模型で実験をする。
- その後は、それまでに得た知見を用いて、さらに快適性や換気性を向上させる方法について検討する。

以下の要素を詰め込んだトンネルに電車を通す。

1. トンネルのみ
2. トンネル, ホーム(階段なし)
3. トンネルが長い場合
 - 長い方が押し出される空気が多い?
 - 短い方が換気のしやすさとしては優れている?
4. トンネル, ホーム, 階段(上に空気の抜ける穴),
 - 電車の方向と同じ方向に流れが発生する? 押し出される?
5. トンネル, ホーム, 換気口(トンネル内の)
 - 換気口をつけた方が、ホームまで届く風は弱くなる?
6. 大岡山駅をほぼほぼ再現
 - 換気口をつけた方が、ホームまで届く風は弱くなる?

模型の大きさについて

大岡山の1/50スケール

実際の長さ(概算)

項目 ホーム トンネル 電車

長さ 180m 200m 120m

幅 7m 4m 3m

高さ 5m 6m 4m

模型の大きさ

項目 ホーム トンネル 電車

長さ 3.6m 4.0m 2.4m
幅 0.14m 0.080m 0.060m
高さ 0.10m 0.12m 0.080m

次回までに行うこと

- 観察方法の調査
 - モデルの作り方について
-

2025/10/08(水) 第3回会議

模型の実際の制作

現在使用可能な材料

- 透明アクリル板
 - 2mm * 320mm * 545mm
 - 新品は薄い白のフィルムがついている
 - アクリルカッターで切ることができる。
 - アクリル板同士、アクリル板と発泡スチロール間で接着剤で固定できる。
 - レーザー加工機で切れるかも？
- 黒いアクリル板
 - 2mm * 320mm * 545mm
 - 性質は透明アクリル板と同じ
 - 煙を見る時のスクリーンとして使えそう
- 発泡スチロール
 - 10mm * 450mm * 450mm
 - カッターで切ることができる。

設計図について

- トンネルについて
 -  トンネル
- ホームの設計図
 -  ホーム
 -  ホーム

板の切断について

- 透明アクリル板
 - 320mm * 80mm (12個)

- 320mm * 120mm (9個)
- 黒アクリル板
 - 320mm * 120mm (1個)

レーザー加工機

- 麦田が導入を担当
- 使用可能になれば大幅な加工時間の短縮となる。

制作ミスについて

板の接着にあたって、アクリル板の厚み2mm分の余白を作らなくてはならないことには事前に気づくことができたが、両端分の余白を作らなくてはならないことにまで意識が届かず、 $120+2*2\text{mm}$ で作るべきところを、122mmで作ってしまった。

接合の面を変更したり、まだ作っていない材料の寸法を微調整したりして実際に再現したい内側の寸法を変えないようにすることができた。

次回行うこと

- 3DCADの準備(使う必要があるかについても検討)
- 今回作成したものを使用し、実際にトンネルを作つてみる。
- ホームの作成も並行して進める。

2025/10/14(火) 第4回会議

模型の実際の制作

最も簡易的なモデルの完成

- トンネル
 - 黒く塗った発泡スチロールを用いたトンネル *2
 - 発泡スチロールで作った電車
 - トンネルの外から糸で引くことができる

簡易的な実験

煙をトンネルに充満させ、電車を動かして、煙の動きを観察した。

- ゆっくり動かした場合と早く動かした場合で比較した。
- 速い場合、煙を押し出しているのではなく、電車の方が空気をかき分けて先に進み、その後に空気がついてくるような動きだった。
- およそ10m/s程度が手で引いた時の速度の限界であるように感じた。

- レイノルズ数を完全に合わせるとなると、50倍の速度で動かさなければならない問題に直面した。

完全なる再現のためのレイノルズ数について

- サイズを1/50にしたので、理想的には速度を50倍にしなければならない。
- 今回実験してみて、かなり早めに引いたが、10m/s程度であった。
- 完全なる再現は難しいことに気づいた。
- 全体としてのテーマを考え直す必要が出てきた。

新たな案出し

- 新たな案
 - 乱流と層流の可視化
 - F1(ダウンフォース)
 - 実寸大よりも小さくすると、速く動かさなければならないという今回の課題と同じことになる？
 - 風を起こす機械を使えば可能？
 - ハンディファン
 - やるとしたら、ハンディファンの最適化
 - 実寸大と同じサイズの模型が作成できるので、今回と同じ課題には直面しなそう
 - レイノルズ数を合わせることの意義
- 現在作成した模型を流用する案
 - 電車内の空気の動き
 - 今まで一旦無視してきた人の影響が大きくなり始める？
 - プラレール

次回行うこと

- 3DCADの準備(使う必要があるかについても検討)
- 今回作成したものを使用し、実際にトンネルを作つてみる。
- ホームの作成も並行して進める。

2025/10/15(水) 第5回会議

テーマの再検討

- レイノルズ数は、実際の電車が 3.7×10^6 、模型が 1.1×10^5 となった。
- レイノルズ数が1桁異なるが、十分大きいレイノルズ数では挙動が同じになりそうであることが文献調査により分かった。
 - 明確な文献を決めて参考文献とする必要がある。
 - 挙動が同じになるかは、対象となる観察現象によっても変わりそうである。
- 風洞などを用いることで、電車と空気の相対速度を上昇させることが可能？

レイノルズ数の計算変数

実際の電車 模型

代表速さ [m/s]	11	16
代表長さ [m]	5.0	1.0×10^{-2}
動粘度 [m^2/s]	1.5×10^{-5}	1.5×10^{-5}
レイノルズ数	3.7×10^6	1.1×10^5

最終決定

そのまま電車のテーマで続行する。

トンネルの拡張

- 前回の製作で2つの短いトンネルの部品(1個400mm)を合わせていた
- 今回は4個まで拡張した。
- トンネルの連結部分で電車が引っかからないようにした。

レーザーを用いた実験

前回は、煙をトンネルに充満させ、電車を動かして、煙の動きを観察した。撮影によって観察する都合上、今までの方法では奥行き方向の厚みや位置がわからない。建築用のレーザーを煙に当てることで、特定の切断面の煙の動きを観察することができた。

PIVについて

- OpenPIVの導入を製作と並行して始めた。
- 現状では、Pythonスクリプトを用いてPIV解析を行う環境を構築しようとしている。
- 具体的には、以下の2つの工程として作成した。
 - 流体の動く動画の各フレーム間での動きをベクトル場に起こす
 - 実験で撮影した煙の動画(.mp4)を入力とする。
 - OpenPIVライブラリを用いて、連続する2フレーム間の煙の移動を解析し、速度ベクトル場を算出する。
 - 算出された速度ベクトル場を、元のフレーム画像に重ね合わせて可視化し、連番画像として指定ディレクトリに保存する。
 - ベクトル場の複数の画像を連結し、動画に変換する
 - 生成された連番画像を入力とする。
 - これらの画像を連結し、PIV解析結果の動画(.mp4)を生成する。
- これらスクリプトの実行により、撮影した動画から流れ場の速度を可視化することができた。
- 観察対象の流体に、粒子状,粉末状が混じっていないと適切にベクトル場の計算ができないと分かった。
 - そのような素材を探す?

- レーザーを用いた観察によって解決できる？

次回までに行うこと

- 十分大きいレイノルズ数では挙動が同じになるかの確証を、文献調査により得る。

次回行うこと

- 共有のためのOpenPIVのソースコードの整理
 - ホームなどの、モデルのさらなる拡張
 - 観察方法の確立
-

2025/10/21(火) 第6回会議

レーザーによる煙の観察

暗室で、レーザーを使うことにより特定面での煙の動きを観察した。 PIVを見越した、気体の動きがカメラで撮影できるように、煙の量や装置の配置を適切になるようにした。 煙の量はかなり少なめである方がいいとわかった。 レーザーがトンネルの上部に差し込むように配置し、レーザーが無駄な方向に出ないように発泡スチロールなどで塞いだ。

模型の作成

もう半分のメンバーがホームの模型の作成を行なった。

次回行うこと

- PIVが適切にできるような方法を模索する。
-

2025/10/28(火) 第7回会議

煙の撮影の成功

- 前回の撮影の試行によりわかったことを踏まえて撮影すると成功した。
 - 煙を少なくすること。
 - 地面で実験を行い、レーザーと対象物を同じ平面に置くこと。
 - 安全のために、レーザーの出力方向を必要な方向のみに制限すること。

PIVの改良

今までの撮影データや、PIVのデモ動画を用いてベクトル場の可視化を行った。今までの問題として、ノイズのような不規則で周りの速度ベクトル場から孤立した方向の大きいベクトルが、まばら

に存在しており、可視化の問題となっていた。原因となるようなノイズは動画には目で見えないようなものであった。その改善のため、数フレームごとに平均中央値を取る機能を実装した。

模型の作成

もう半分のメンバーがホームの模型の拡張、改良を行なった。

次回行うこと

- 共有のためのOpenPIVのソースコードの整理
 - 可視化の方法について改めて検討
 - ホームなどの、モデルを用いた実験
-

2025/10/29(水) 第8回会議

ホーム連結モデルでの実験

- 前回までに作成したホームの模型を2つ連結し、よりサイズの大きい模型で実験した。
- この連結したホームとトンネルのモデル内に煙を充満させ、模型の電車を走らせる実験を行った。

実験結果と観察

- 連結部分の接合が完全でないため、空気が漏れ出している箇所があった。
- 煙の動きは、大部分が中央に留まり、あまりダイナミックな動きは見られなかった。ドアから離れる方向に静かに流れる傾向があったので、改善をしなくてはならない。
- また、模型の連結部分を電車が通過する際に物理的に引っかかることがあり、一定の速度で模型を牽引することが困難であった。
- この問題を軽減するため、電車に2本の糸を通して少し宙に浮かせることで摩擦を減らし、よりスムーズに引けるよう工夫した。

PIV解析の課題

- この実験の様子を撮影した動画でPIV解析を試みたが、煙の動きが複雑であり、特にホーム周辺での解析精度が課題となった。
- 前回に引き続き、ノイズの多いベクトルが散見され、流れの全体像を掴むのが難しい状況である。

次回行うこと

- 模型の連結部分をより滑らかにし、意図しない乱れの発生を抑制する。
 - PIV解析のパラメータ（検査窓のサイズ、移動平均のフレーム数など）を調整し、解析精度の向上を目指す。
-

2025/11/05(水) 第9回会議

PIVのソースコードの共有

- 今まで個人で開発していたPIVのソースコードを、GitHubで共有し、誰でも実行できる環境を整えた。

模型の作成

- ホームの増築も行った。

中間発表の役割分担

中間発表に向けて、スライドの作成内容で役割分担を決定した。

- 題名
- テーマについて
 - 研究の概要
 - テーマを選んだ理由
- 検証方法
 - 現状PIV
- 作成した模型などについて
 - 材料
 - サイズ
 - ドライアイス, 煙
- レイノルズ数について
 - 計算について
 - モデルの限界について
- 結果考察
- 現在の問題点
- 今後の展望

次回行うこと

- 中間発表の資料作成を開始する。

2025/11/07(金) 第10回会議

PIVの精度向上とパラメータ調整

前回の課題であったPIVのノイズ除去と精度向上について、試行錯誤の結果、一定の成果と新たな課題が見つかった。

- 高コスト解析による成功事例:

- これまで解析時間の観点から試行してこなかった、計算コストを度外視したパラメータ（小さい検査窓、大きい移動平均など）で解析を行ったところ、一つの動画データで非常に質の高いベクトル場を得ることに成功した。
- この成功事例では、ノイズが少なく、流れの挙動を明確に可視化できており、中間発表の主要な成果として報告できる品質である。

- 再現性の課題と今後の展望:

- しかし、同じ高コストな設定を別の動画データに適用したところ、同様の品質は得られなかつた。
- このことから、単に計算コストを高くすれば良い結果が得られるわけではなく、動画の特性（煙の濃度、流速、照明条件など）に合わせた調整が不可欠であることが判明した。
- 今後の課題として、動画自体の前処理（コントラスト調整など）や、対象の動画に適したパラメータを効率的に決定するための手法を確立する必要がある。

ラズベリーパイによる模型制御の検証

人力に頼らない安定した模型の駆動を目指し、ハードウェア制御の導入を検証した。

- 使用デバイス: Raspberry Pi Pico 2 W

- 制御方式: MicroPythonのPWMモジュールを用いたサーボモーター制御。

- 検証結果と課題:

- Pythonコードによるモーターの回転制御自体は問題なく動作し、人力駆動から脱却できる可能性が示された。
- しかし、現状の構成では模型を駆動する速度が明らかに不足しており、レイノルズ数を合わせた実験の要求を満たせないことが判明した。
- そのため、より高トルクなモーターの選定やギア比の検討が必要であり、当面は再現性に課題が残るもの、人力による駆動を継続することになった。

中間発表についての話し合い

- 間近に迫った中間発表の構成や内容について議論した。

- 主な発表の柱として、以下の2点を決定した。

1. PIV解析の進捗報告（高コスト解析による成功事例と、今後の課題）。

2. Raspberry Pi Picoを用いた実験系の改善と、今後の定量的なデータ取得に向けた展望。

次回行うこと

- PIV解析における動画の前処理手法の検討。
- 各動画に適したパラメータ設定方法の模索。
- 中間発表に向けた資料作成。

PIVについて (docs/piv.md)

トラブルシューティング

発表について (docs/presentation.md)

中間発表 (2025/11/07提供情報)

- プレゼン: 15分~20分
- 発表形式: 役割分担は適宜。ただし、3Q終了時(中間)と4Q終了時(最終)では違う役割を担当すること。
 - 司会: 別グループから選出

発表	司会
ペットボトルロケット	現象の可視化
現象の可視化	自励振動
自励振動	樹脂成形
樹脂成形	ペットボトルロケット

- 使用ツール: PowerPointなど
- 質疑応答: 4つの全ての課題を通じて、各自少なくとも1回は質問すること
- フィードバック
 - 司会は担当したグループのPPT資料を受領し、内容・プレゼン双方についてチーム内で検討し、フィードバックを作成
 - フィードバックはPDFにしてチーム内Slackへ提出 → 教員が相手グループのSlackへ転送
- 提出物
 - プrezenで使ったppt
 - 各個人の中間報告レポート(自由形式)
 - 1分間の説明動画 (FHD, mp4)
- 打ち上げ
 - 学生無料

Raspberry Pi Pico と MicroPython によるハードウェア制御ガイド

このドキュメントは、Thonny IDE と MicroPython を使用して Raspberry Pi Pico のハードウェアを制御するための技術的な知識をまとめたものです。

1. Raspberry Pi と Raspberry Pi Pico の違い

まず最も重要な点として、「Raspberry Pi」と「Raspberry Pi Pico」は全く異なるデバイスです。

特性	Raspberry Pi (SBC)	Raspberry Pi Pico (MCU)
種別	シングルボードコンピュータ (SBC)	マイクロコントローラボード (MCU)
OS	Linux (Raspberry Pi OS) が動作	OSは搭載しない (ベアメタルに近い)
Python環境	CPython (標準Python)	MicroPython (組み込み向け)
主な用途	PCのような高レベルな処理	リアルタイムなハードウェア制御
GPIO制御	<code>RPi.GPIO</code> ライブラリ (OS経由)	<code>machine</code> モジュール (直接制御)

結論: `machine` モジュールを使用するコードは、Raspberry Pi Pico を対象としています。

2. Raspberry Pi Pico の基本

MicroPython と Thonny IDE

Picoは、組み込みデバイス向けに最適化されたPythonである **MicroPython** で開発を行います。

Thonny は、Picoへのプログラムの書き込みや実行を簡単に行える、初心者にも推奨されるIDEです。

GPIOの基本

- **ピン配置:** Picoには物理的なピン番号（1～40）と、プログラムで指定するGPIO番号（GP0～GP28）があります。コーディングの際は **GPIO番号** を使用します。
- **3.3Vロジック:** PicoのGPIOは **3.3V** で動作します。5Vのデバイスを直接接続すると破損の原因となるため、ロジックレベルシフタを使用してください。

3. ハードウェア制御の実践

3.1. Lチカ (LED点滅)

LEDの点滅は、ハードウェア制御の第一歩です。

`machine.Pin` モジュール

GPIOピンを操作するには `machine.Pin` クラスを使用します。

```
from machine import Pin
import time

# GP25を「出力モード」で初期化
led = Pin(25, Pin.OUT)

while True:
```

```
led.on()      # LEDを点灯 (3.3V出力)
time.sleep(1)
led.off()     # LEDを消灯 (0V出力)
time.sleep(1)
```

オンボードLEDの利用

Picoの基板上にはLEDが実装されており、「LED」という特別な名前（エイリアス）で簡単にアクセスできます。この方法は、PicoとPico Wのハードウェア的な違いを吸収してくれるため、推奨されます。

```
led = Pin('LED', Pin.OUT) # ← これだけでOK
```

外部LEDの接続と電流制限抵抗

外部にLEDを接続する場合、PicoのGPIOピンを保護するために必ず電流制限抵抗を直列に接続してください。抵抗値はオームの法則で計算します（一般的に $220\Omega \sim 330\Omega$ 程度が使われます）。

- 回路: $\text{GPx} \rightarrow \text{抵抗} \rightarrow \text{LEDのアノード} \rightarrow \text{LEDのカソード} \rightarrow \text{GND}$

3.2. サーボモーターの制御

サーボモーターの制御には、**PWM (Pulse Width Modulation)**という技術を使用します。

PWMの原理

PWMは、デジタル信号のON/OFFを高速で繰り返し、ONの時間（パルス幅）を変化させることで、アナログ的な制御を実現する技術です。サーボモーターの場合、このパルス幅によって回転角度や回転速度を指示します。

- 標準的なサーボ制御信号:
 - 周波数: 50Hz (周期20ms)
 - パルス幅:
 - 1.0ms (1000us): -90° (または最大逆回転)
 - 1.5ms (1500us): 0° (または停止)
 - 2.0ms (2000us): +90° (または最大正回転)

machine.PWM モジュール

PicoでPWM信号を生成するには `machine.PWM` クラスを使います。

```
from machine import Pin, PWM

# GPIOをPWMピンとして設定
pwm = PWM(Pin(0))
```

```
# 周波数を50Hzに設定
pwm.freq(50)
```

パルス幅[us]からduty_u16への変換

PicoのPWM API `pwm.duty_u16(value)` は、パルス幅を直接指定するのではなく、デューティ比を0～65535の16ビット整数で指定します。

そのため、サーボが要求する「パルス幅 (us)」を、Picoが要求する「duty_u16値」に変換する関数が必要です。

```
# 周期は 20ms = 20,000us
PERIOD_US = 20_000

def pulse_us_to_u16(pulse_us: int) -> int:
    """パルス幅[us]をduty_u16(0-65535)に変換する"""
    return int(65535 * (pulse_us / PERIOD_US))

# 例: 1500us (停止信号) を変換
stop_duty = pulse_us_to_u16(1500) # 約4915
pwm.duty_u16(stop_duty)
```

実践的なサーボドライバ

安価なアナログサーボには「個体差」があり、理論値通りに動作しないことがよくあります。以下の要素をソフトウェアで補正することで、より正確な制御が可能になります。

- トリム調整 (`TRIM_US`): 停止点のズレを補正するオフセット値。
- 不感帯 (`DEADBAND_US`): 停止点付近の微振動（ジッター）を防ぐための範囲。

安全な停止処理 (`try...finally`)

ハードウェアを制御するプログラムでは、予期せぬ終了時にもデバイスを安全な状態に保つことが重要です。`try...finally` ブロックを使うことで、プログラムが停止した際に必ずモーターを止め、PWM信号を解放できます。

```
try:
    # メインの処理
    set_speed(0.5)
    while True:
        time.sleep(1)
except KeyboardInterrupt:
    print("プログラムが停止されました。")
finally:
    # この処理は必ず実行される
    set_speed(0) # モーターを停止
    pwm.deinit() # PWMを解放
    print("安全に終了しました。")
```

4. 安全な電源供給

モーターなどへの電源供給

サーボモーターのように大きな電流を必要とするデバイスをPicoのGPIOピンから直接駆動ではありません。Picoの電源ピン（VBUSやVSYS）や、外部の安定した電源（バッテリー、ACアダプタなど）を使用してください。

- 重要な注意: 外部電源を使用する場合、必ずPicoのGNDと外部電源のGNDを接続（共通接地）してください。これを怠ると、制御信号が正しく伝わらず、デバイスが予期せぬ動作をします。
-

スケジュール (docs/schedule.md)

講義としての日程、プロジェクトとしてのスケジュールについて

授業日程

回	日付	場所	授業計画	課題
第 01 回	10/01(水)	I3- 107	オリエンテーション・安全講習	適宜、課題を課す。
第 02 回	10/07(火)	I3- 107	ミニ講義・グルーピング・分野選択・テーマ検討	適宜、課題を課す。
第 03 回	10/08(水)	I3- 203	テーマ検討・テーマ決定	適宜、課題を課す。
第 04 回	10/14(火)	I3- 107	個別テーマ開始	適宜、課題を課す。
第 05 回	10/15(水)	I3- 203	個別テーマの推進	適宜、課題を課す。
第 06 回	10/21(火)	I3- 107	個別テーマの推進	適宜、課題を課す。
第 07 回	10/28(火)	I3- 107	個別テーマの推進	適宜、課題を課す。

回	日付	場所	授業計画	課題
第 回	08	10/29(水)	I3- 203	個別テーマの推進
				適宜、課題を課す。
第 回	09	11/05(水)	I3- 203	個別テーマの推進
				適宜、課題を課す。
第 回	10	11/07(金)	I3- 107	個別テーマの推進
				適宜、課題を課す。
第 回	11	11/11(火)	I3- 107	製作物、測定データのまとめ
				製作物、測定データのまとめ
第 回	12	11/12(水)	I3- 203	中間発表の準備
				中間発表の準備
第 回	13	11/18(火)	I3- 107	各チーム毎に中間発表
				4Q に向けての計画策定・中間報告書の準備
第 回	14	11/19(水)	I3- 203	4Q に向けての計画策定・中間報告書の準備
				4Q に向けての計画策定・中間報告書の準備
第 回	15	12/03(水)	I3- 107	オリエンテーション・個別テーマの推進
				適宜、課題を課す。
第 回	16	12/09(火)	I3- 107	個別テーマの推進
				適宜、課題を課す。
第 回	17	12/10(水)	I3- 107	個別テーマの推進
				適宜、課題を課す。
第 回	18	12/16(火)	I3- 107	個別テーマの推進
				適宜、課題を課す。
第 回	19	12/17(水)	I3- 107	個別テーマの推進
				適宜、課題を課す。
第 回	20	12/23(火)	I3- 107	個別テーマの推進
				適宜、課題を課す。

回	日付	場所	授業計画	課題
第 回	21	12/24(水)	I3- 203	個別テーマの推進
				適宜、課題を課す。
第 回	22	01/06(火)	I3- 107	個別テーマの推進
				適宜、課題を課す。
第 回	23	01/07(水)	I3- 203	個別テーマの推進
				適宜、課題を課す。
第 回	24	01/13(火)	I3- 107	個別テーマの推進
				適宜、課題を課す。
第 回	25	01/14(水)	I3- 203	個別テーマの推進
				データのまとめ、最終報告会の準備
第 回	26	01/20(火)	I3- 107	最終発表会の準備
				最終報告会の準備
第 回	27	01/21(水)	I3- 107	各チーム毎に最終発表
				成果物(制作物、動画、報告書など)の提出
第 回	28	01/27(火)	I3- 107	成果物(制作物、動画、報告書など)のまとめと片付け
				成果物(制作物、動画、報告書など)の提出