

4つのアームを持つロボット競技会用ロボットの製作

加治木琥南^{*}, 相馬翔^{*}, 渡邊善飛^{*}, 加來涼雅^{**}, 江口智弘^{*}(指導教員)

^{*}電子情報技術科, ^{**}情報システム技術科

毎年開催されている九州ポリテックビジョンロボット競技会では、ロボットがピンポン玉を取得し指定の的に早く投入する競技を行っている。今年度はアームを4つ使用することで移動せずに連続でピンポン玉を取得し発射すること、ピンポン玉を取得した場所から複数の的に狙えるようにすることを目標にロボットを製作した。正しい位置に停止できれば上下どちらからもピンポン玉を取得でき、射出精度も80%以上を実現できた。さらに、X Beeを使用してロボットとノートPC間を無線通信することで、対戦相手の獲得した的に応じて、リアルタイムで狙う的を変えられるようにできた。

1. はじめに

毎年参加している九州ポリテックビジョンロボット競技会において、本研究室で製作した従来のロボットは取得から射出までに要する時間が長いという問題があった¹⁾。そこで、設置台前に移動したら、その場から移動することなく4つのピンポン玉を順番に取得して射出する動作を繰り返すことによって、競技時間を短縮できると考えた。

具体的には以下の2点について実施することを本研究の目的とした。

- ① 各ピンポン玉の前まで移動せず、4つのピンポン玉を取得できるようにする。
- ② ピンポン玉を取得した場所から移動せず、複数の的に狙う。

2. 競技会規定²⁾

競技フィールドおよび投入台を図1, 2に示す。左右各4か所の高さの異なる設置台に置かれたピンポン玉を取得し、9か所の的に入れる早さと正確さを競う。予選では9か所の的の内、予め指定された5か所への投入を行い、要した時間と得点を競う。本戦は対戦形式で行われ、縦横斜めのいずれか1列の的に早く投入させたチームの勝利となる。相手が投入した的についてはこちらが後から投入すると、取得状況を上書きすることができる。出場するロボットは、本体にマイクロコンピュータを必ず搭載し、競技開始前の状態で外周長1200[mm]以内、重量10[kg]以内とする。競技中はタブレットPCもしくはノートPCを用いて、ピンポン玉の取得及び射出の指示に関してロボットと通信をしてよい。

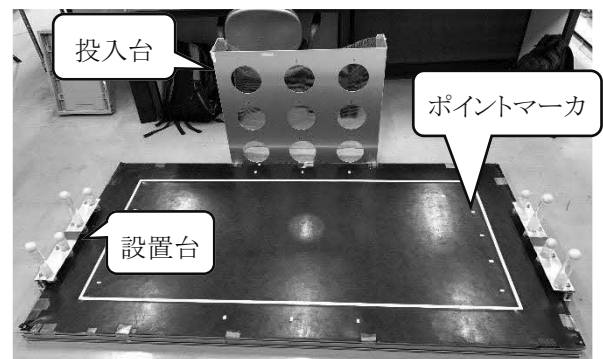


図1 競技フィールド

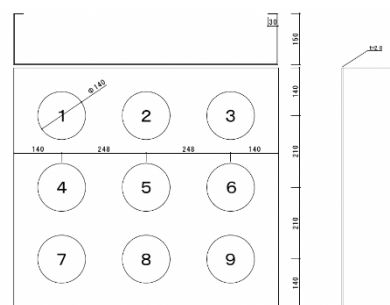


図2 投入台

3. 製作するロボットのコンセプト

規定上、ロボットはピンポン玉を同時に1個しか所持することができない。また、投入台は設置台から離れているため、投入台正面まで移動してピンポン玉を投入すると大変時間がかかる。できる限り短時間で競技を完了するために、設置台前に移動したら、その場から移動することなく4つのピンポン玉を取得して射出する動作を繰り返すロボットにすることを考えた。そのために、

140[mm]おきに設置してあるピンポン玉に位置が合うように取得装置を4つのアームで実現した。ロボットの構成および外観を図3, 4に示す。

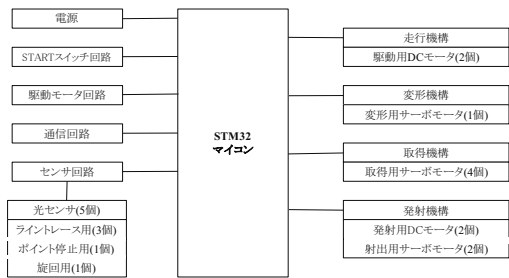


図3 ロボットの構成

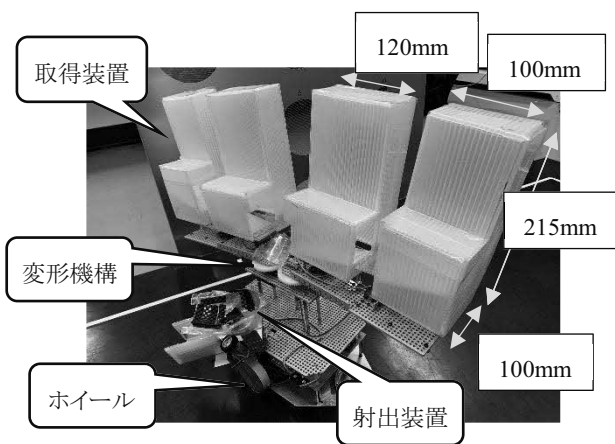


図4 ロボット外観

4. 駆動装置

4.1 ホイール

駆動装置は、主に DC モータ(AO-8008, タミヤギヤードモーター, 3633K75), ホイール(φ65mm), ハブ, ボールキャスター(フリーベアコーポレーション, C-5L-5)から構成されている。ホイールとボールキャスターを2個ずつ使用し駆動させた。モータドライブ回路(モータドライバ:TB6612)を図5に示す。STM32 マイコンからのPWM信号によって速度を, IN1 及び IN2 の組合せによってモータの回転方向を制御した。また, マイコンとモータドライバの間にLEDを挿入することで, 信号の状態を確認できるようにした。

4.2 センサ

光センサ(反射型フォトマイクロセンサ, オムロン, EE-SF5-B)によって白ラインやポイントマーカを検知する。図6に示すように, 光センサ5つをラインレース用センサ(左センサ, 中央センサ, 右センサ), ポイント停止用センサ, 旋回用センサとして配置している。ラインレー

ス用センサは, ロボットが移動するラインを検知する。ポイント停止用センサは, ピンポン玉設置台前のポイントマーカを検知する。旋回用センサは 90 度に旋回する場所を検知する。

4.3 ライントレース

フィールドの走行は, 前方に配置したラインレース用センサ3つを使用して, 白いラインを追いながらラインレースを行う。直進時の速度は約 210[mm/s]であった。ホイール横に配置している旋回用センサが進行方向のラインを検知すると, 即座に左に旋回する。その後, 中央センサがラインを検知したタイミングでラインレースを再開する。

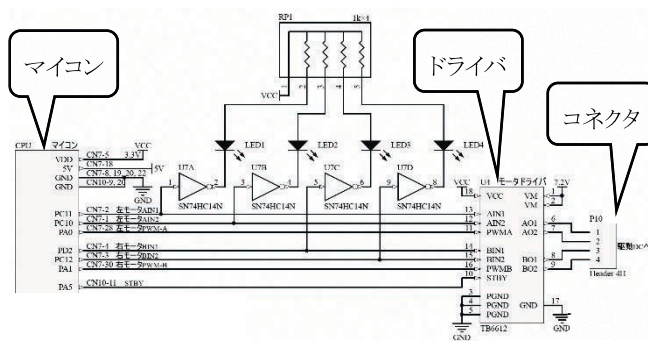


図5 モータドライブ回路

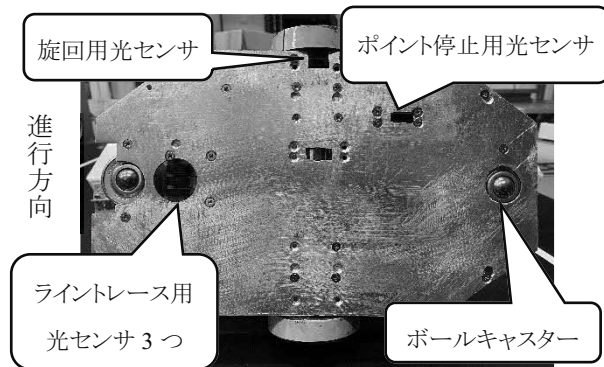


図6 センサ配置

5. 取得および射出装置

5.1 取得装置の構成と性能

本年度は, 箱型の取得装置を製作し, それを上から被せることで高低どちらの設置台からもピンポン玉を取得可能な構造にした。取得箱の角度は取得用サーボモータ(Tower Pro, MG92B)によって制御した。

取得装置および内部の斜面を図7に示す。取得したピンポン玉は箱内部に設けた斜面によって, 4つの取得箱の中央の射出装置まで送る。

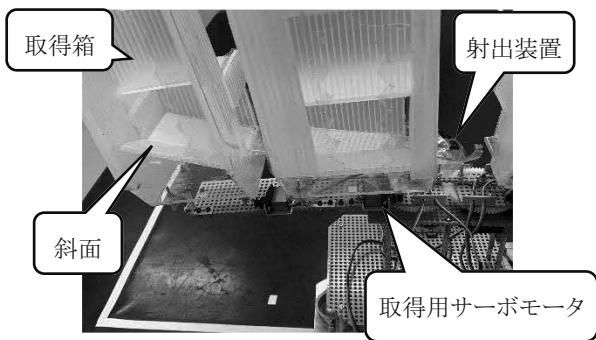


図7 取得装置

5.2 変形機構の構成

本競技会では競技開始前のロボットの外周長が1200 [mm]と規定されており、取得装置のアームを開いた状態だと規定内に収まらない。また、開いた状態で走行すると設置台と接触しピンポン玉を落とす危険があった。そのためアーム台を中央で折りたたむことができる変形機構を取り入れた。

変形機構を図8に示す。変形機構にはギアを使うことで、変形用サーボモータ(Tower Pro, SG-5010)を1つ使用するだけで変形することを可能にした。これにより競技開始前はアームを閉じておき、設置台前方まで移動したらアームを開き、移動することなく4つのピンポン玉を取得できた。

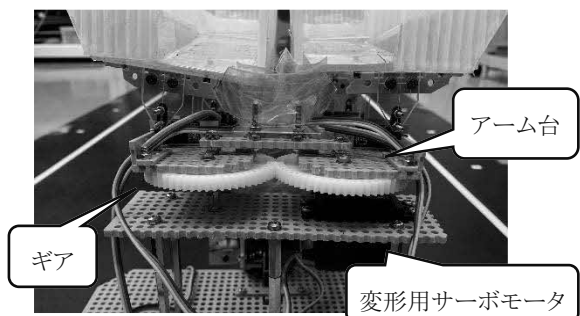


図8 変形機構

5.3 射出装置の構成と性能

射出装置を図9に示す。本ロボットは、各ピンポン玉の取得位置からの的を狙うため、射出部分を上下左右に動かす必要がある。射出のための方向調整は射出用サーボモータ(Tower Pro, MG92B)2つによって制御した。取得直前に射出装置を的に合わせ、取得後に射出用DCモータ(TAMIYA, AO-1001 FA-130タイプノーマルモータ)2つを回転させるようにしている。傾斜によってピンポン玉を射出装置まで誘導する。その後、ピンポン玉を二つの射出用DCモータに取り付けたタイヤで挟み込むようにして飛ばし、投入を行う。

斜めからの的を狙うと、正面から狙うより投入可能面積が狭くなるため、高い射出精度が求められる。そのため射出用DCモータの回転数を増し、バレルの形状をコの字型にしてピンポン玉の接地面積を減らすことにより射出速度を上げた。また、タイヤの角度を斜め下に傾け、バレル下部に跳ねさせて直線性のあるピンポン玉の射出を可能にした。予め各設置台から狙う的を決め、事前に位置を合わせて必要となる射出用サーボモータの角度を確定した。

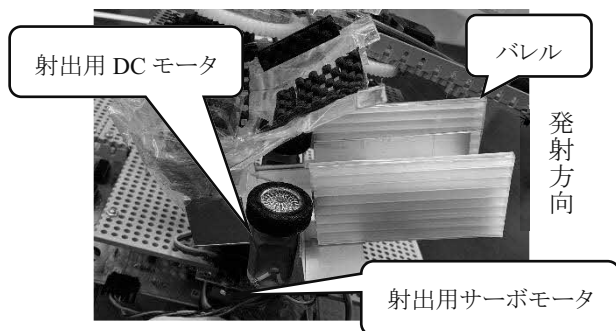


図9 射出装置

今回使用したDCモータの印加電圧を約2.5[V]とした。これは、電流を小さくしつつピンポン玉の軌道がなるべく直進になるように実験した結果である。この時のモータ2つの電流は約0.5[A]であった。

実際の競技会での射出投入成功率をまとめ、その結果を表1に示す。各結果から射出位置からの的までの距離が長い程成功率が低下する傾向にあり、振動による射出装置のブレや、ロボット本体の停止位置のずれが大きく影響すると考えられる。

表1 競技会でのピンポン玉の射出投入成功率

	的	射出数	成功数	成功率
投入台 向かって 左側からの 射出	1	10	10	100.0%
	4	12	10	83.3%
	7	8	6	75.0%
	5	3	1	33.3%
	9	1	0	0.0%
投入台 向かって 右側からの 射出	3	4	4	100.0%
	6	1	1	100.0%
	9	4	4	100.0%
	5	2	2	100.0%

6. 通信機能³⁾

本戦は陣取りのビンゴ形式で行われるためリアルタイムでの的の変更が必要となってくる。そのため、XBeeによる無線通信機能を導入し、的の番号や指示の送信と、ロボットの状態の受信を行えるようにした。XBeeとは、ZigBee規格に対応した無線モジュールのひとつである。通信方法は、半二重通信で行った。半二重通信は双方向通信において、一度に片方しか通信できない通信方法である。

STM32のシリアル通信機能を使い、操作者がノートPCで予選では指定した番号の的に入ったかの成否(M/N)、本戦では狙う的の番号(1~9)、射出終了(F)を入力し、その文字或いは番号をロボットへXBeeを介して送信する。ロボットが受信すると、取得動作、射出動作へと移行する。ロボット側からはライトレース中の各センサの反応をPCへ送信する。

7. 動作フロー

ロボットの動作は以下の通りである。

- ① スタート地点にロボットを配置後、ノートPCから開始の信号を送信する。
- ② 旋回センサが検知したら左に90度回転後、ライトレースをする。
- ③ ポイントセンサが3回または9回検知したら、ロボットを一時停止後、アーム台を開いて待機する。
- ④ 的の番号を指定し、バレルの向きを的に合わせて、ピンポン玉を取得し発射する。

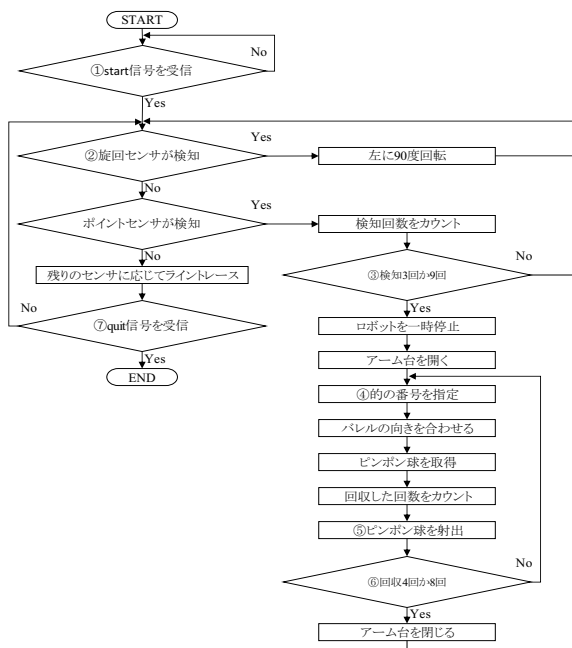


図10 動作フローチャート

- ⑤ ピンポン玉が的に入ったかを目視で確認し、成功失敗に応じて次の動作の指示をする。
- ⑥ 4回または8回収動作を行ったら、アーム台を閉じて再度ロボットを走らせる。
- ⑦ 緊急停止信号を受信したらロボットを停止させる。

8. 競技会報告

競技会には合計10チームが参加した。予選では想定通りに走行でき、指定された5つの的への投入に要した時間が52秒で1位という結果だった。本戦では2本先取で試合が行われ決勝まで進出することができた。決勝はセットカウント1-1まで持ち込み、作戦通り速攻で挑んだものの、相手からの取得状況の上書きを見落として指示を間違えてしまい、最終的に1-2で敗北し準優勝となった。

他チームのロボットには、オムニホイールを用いてショートカットを行うものや、垂直ベルトコンベアを用いてピンポン玉の移動を行うものがあった。優勝はバレルを伸ばしたタイプのロボットであった。

取得後即座に打つロボットは他に無く、本ロボットの取得から射出のインターバルは参加チームの中でも最速だった。本研究の目的であった「最小限の移動で4つのピンポン玉を取得し、複数の的を狙う」は達成できたといえる。

9. まとめ

今年度のロボットでは、取得装置の数を増やしてその場から4つのピンポン玉を取得できるようにした。そのままだとロボットの外周長が規定を超えてしまうため、変形機構を組み込むことで規定内に収め、取得から射出のインターバルを早くすることができた。また、ピンポン玉が直線性のある軌道を描くように射出装置を調整し、移動せず複数の的を狙えるようにした。

今後の課題は、外乱光によりライトレースの動作が不安定になること、まれに取得不良になること、本研究室の設置台が歪んでおり実際の競技会のものとは多少異なることなどの改善である。

参考文献

- 1) 榊原美月, 島川桜菜, 船本拓実:「ピンポン玉投入競技用ロボットの製作と競技会報告」, 令和5年度卒業研究概要集, 熊本県立技術短期大学校(2023), pp.61-64.
- 2) 「九州ポリテックビジョンロボット競技会～競技会規則～」, 九州職業能力開発大学校(2025)
- 3) 永田武:「RaspberryPiによるIoTシステム開発実習」, 森北出版株式会社, 2021-9-21, pp.79-105.

離床検知装置の通知方法の改善と機械学習手法の検討

*鶴田 健人, **奈須 光希, *廣瀬 龍輝, *江口 智弘(指導教員)

*電子情報技術科, **情報システム技術科

高齢者施設における被介護者(以下, 高齢者という)が夜間の睡眠中にベッドから離床することやベッドから転落することがある。これを未然に防ぐために, 離床を検知したら職員へ通知する離床検知装置の開発に取り組んできた。今年度は, 教師なし学習を用いて検知結果の正否を判断するようにした。高齢者の状態を表した ORB 記述子特徴量を用いて Isolation Forest アルゴリズムによって学習をさせた。テストデータによる正解率は 90.03%であったが, 装置に組み込んだ状態では異常状態を正常状態と判断した。LINE から Raspberry Pi への通知は, LINE Messaging API や Cloud Run 関数などを用いて実現できた。

1.はじめに

高齢者施設において高齢者が, 夜間の睡眠中に単独でベッドから離床することでベッドから転落し受傷に繋がることもあるため, 施設職員(以下, 職員という)は夜間の見回りをおこなっている。夜勤職員の人数は限られており, 見回りが及ぼす職員への負担は小さくない。職員の負担軽減のため, 危険に至る前的高齢者の状態を瞬時に伝える離床検知装置を開発してきた。

これまでに実施した高齢者施設での評価実験の結果から全体的な評価は高かった。¹⁾しかし, 以下の2点について課題が残っていた。

- (1)部屋の明るさによって誤検知が起きる。
- (2)カメラの位置を変えることができず, 装置の配置に自由度がない。

(1)に対して昨年度は教師あり学習を用いて, 良否の判断をすることで対応したが, 不具合が見つかるたびに学習させる必要があった。また, (2)に対しては, 携帯端末から Raspberry Pi へ設置されたカメラの位置を通知できれば, 画像処理で対応が可能と考えた。そこで, 以下の2つを本研究の目的とする。

- (1)教師なし学習(Isolation Forest)を用いて離床を検知する。
- (2)LINE から Raspberry Pi への通知を可能にする。

2.機械学習とは

機械学習とはコンピュータプログラムがデータから学習し, その結果を基に解決に導く手法のことである。機械学習には教師あり学習と教師なし学習がある。

教師あり学習とはラベル付きデータを使ってモデルを学習させる方法である。ラベル付きデータが与えられ

るため, モデルは明確な答えを学習でき, 高い予測精度が期待できる。しかし, 学習データ量が多い場合ラベル付けには時間とコストがかかる, 訓練データを過学習すると新しいデータに対する汎化能力が低くなり, 予測精度が悪化するというデメリットがある。

教師なし学習ではデータにラベルが不要のため, ラベル付けの手間やコストを削減できるというメリットがある。しかし, ラベルがないため, 目標の評価基準に向けて学習することが難しいといったデメリットがある。

Python で実装されている機械学習のためのライブラリ Scikit-learn を使用した。

3.離床検知装置とは

離床検知装置とは, 高齢者施設において高齢者の就寝時間中に使用して, 高齢者の起き上がりや動きを検出し, LINE を通して携帯端末に通知する装置である。

従来, オプティカルフローで得られた速度ベクトルの大きさが 10 以上かつ 5 個以上を「起き上がり」または「足のはみ出し」の検知条件として判定した。¹⁾しかし, 明度の違いで発生した光と影の変化を動きとして検知してしまうことがあった。そのため昨年度は, 従来の検出方法で得られた結果が正しいかどうかを判断するために教師あり学習を取り入れた。²⁾しかし, 教師あり学習では, 様々な条件に対応した学習データを準備する必要があり, すべての条件を満足できるモデルを作成することが困難であった。

そこで, 今年度は教師なし学習を取り入れ, 離床につながる起き上がりや足のはみ出しなどの異常の状態を検知できることを目指した。つまり, 正常状態を学習させることになるが, 本装置において臥位を正常状態とし

ている。動きがない状態では従来使用していたオプティカルフローでは検出できないため、別の特徴点抽出手法も併せて検討した。

4. 特徴点抽出

本研究では動きのない状態でも、その状態を少ないデータ量で表せる特徴量記述子を用いた。OpenCV では「detectAndCompute」メソッドを用いて、キーポイントを検出した後に記述子を計算する。

キーポイントとは、エッジやコーナーなど画像の中で特徴的な点(以下、特徴点という)を指す。キーポイントは、位置、スケール、方向、応答値などの情報を持つ。記述子とは、キーポイントの周囲の情報を数値ベクトルとして表現したものである。ただし、記述子には座標情報を含まない。記述子を学習用データとするため、1 秒ごとに求めた。

特徴点抽出の代表的手法として、SIFT, ORB, FAST などがある。代表的なアルゴリズムの特徴を表1にまとめた。本研究では、表1のように次元数が少なく、高速低計算量の ORB (Oriented FAST and Rotated BRIEF)記述子を用いた。以下の図 1 は、SIFT, ORB, FAST で特徴点抽出を行った画像である。

表 1 各特徴点抽出の特徴表³⁾

特徴	SIFT	ORB	FAST
特徴点検出方法	ドッグを用いたスケール空間	FAST 検出器と回転不変の BRIEF 記述子の組み合わせ	FAST アルゴリズムによるコーナー検出
計算速度	遅い(高精度のため)	高速(計算負荷が軽い)	超高速(計算効率が高いため)
次元数	128 次元	32 次元	-
精度	高精度	高精度	やや低い
利点	詳細な特徴点の抽出が可能	高速, 低計算量	超高速, 簡単なアルゴリズム
欠点	計算量が大きく遅い	精度が若干劣る	精度が低い

5. 学習用データ作成

5.1 背景差分と ORB 記述子

学習データを作成するために、被験者4人(20代, 身長 165cm~175cm, 男性)に明るい状態で仰向け、寝返り、膝立ての 3 パターンで寝てもらいその様子を撮影した。撮影した動画を左右反転させたもの、夜間用に暗くさせたものをそれぞれ作成し、全部で 48 個の動画を準備した。

備した。

本研究では学習用データを作成するにあたって、人物の動きの特徴点を抽出するために背景差分を行った。「cv2. createBackgroundSubtractorMOG2」メソッドを利用して背景差分を行い、1 秒ごとに処理を実施し、異常検知に適した特徴点を抽出するための前処理として適用している。この手法では、各ピクセルの輝度値を複数のガウス分布でモデル化し、背景に属すると考えられるピクセルを抑制し、前景のみを抽出する。以下の図 2 が本研究で背景差分したベッドの画像の例である。

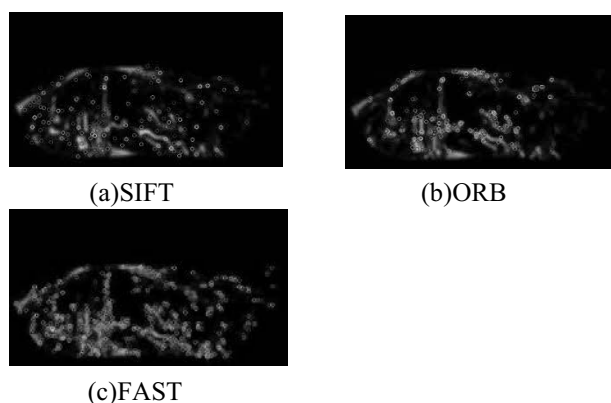


図 1 アルゴリズムによる特徴点抽出の違い

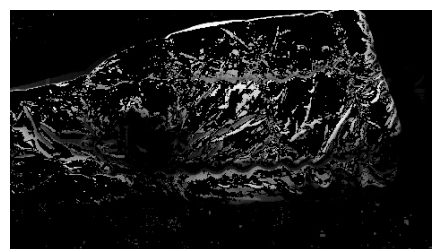


図 2 背景差分後のベッドの画像

5.2 標準化

標準化とは、データの各特徴量を平均 0, 標準偏差 1 に変換する処理のことである。標準化を行うことで異なる特徴量を持つスケールの違いを解消し、モデルの学習を安定させることが出来る。Scikit-Learn の「StandardScaler」というインスタンスを用いることで標準化を行うことができる。標準化では次のような式で各特徴量を変換している。

$$z = \frac{(x - \mu)}{\sigma} \quad (1)$$

ここで x は元の特徴量, μ はその特徴量の平均, σ はその特徴量の標準偏差を表す。

5.3 次元数削減

表 1 のように SIFT 記述子の次元数は 128 次元, ORB 記述子は 32 次元である。次元数を削減することができ

れば、計算量が減少し、処理の高速化、効率化ができる。動画から求めた記述子を CSV に保存し、その CSV データを主成分分析 (PCA) にかけることで次元数を削減し、96 個の学習データを作成した。

主成分分析とは、多数の要因を少数の総合指標に集約したいときに使う。何番目の主成分まで使えば元のデータの特徴を十分に把握できるかを判断する基準となる寄与率という指標がある。その合計である累積寄与率は、数値が高いほど、元のデータの特徴を把握できているため、精度の高い分析を行うことができる。

本研究では累積寄与率が 90% を超える最小の次元数を取得し、その次元数をもとに「fittransform」メソッドで次元削減を行った。図 3 に累積寄与率のグラフ例を示す。この例では ORB 記述子の 32 次元を PCA にかけて、90% が 23 次元に相当している。

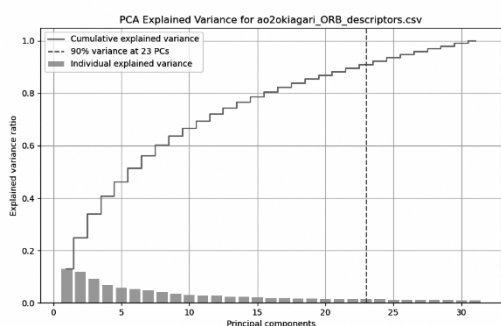


図 3 次元数削減に用いた累積寄与率の例

6. モデル学習

Isolation Forest は、バイナリツリーを使用したデータ異常検知の手法である。簡単に言うと、正常なデータと異常なデータを切り分けることに特化した手法である。

まずランダムにデータを分割する。異常なデータは少ない分割で孤立するため、孤立するまでのステップ数をカウントすることで「正常 or 異常」を判定できる。

今回は「仰向け」、「寝返り」、「膝立て」の 3 つの姿勢を正常データとして学習させ、カメラで得た新規データの「正常 or 異常」を判定している。正常データだけを学習して、そのデータをモデル化する。新しいデータがその正常データの範囲に収まるか外れるかを評価し、収まるデータは「正常」、外れるデータは「異常」として分類する。

本研究ではデータの分割に「train_test_split」メソッドを使用して、作成した学習データを「訓練用データ」と「テストデータ」の 2 つのデータに分割した。

「train_test_split(x, test_size, random_state)」の x は分割したいデータを表しており、PCA で次元削減した

記述子にあたる。test_size で分割の割合が決められ、float 型で指定するとその割合がテストデータになる。例えば、0.2 と値を指定することで、訓練用データを 8 割、テストデータを 2 割に分割できる。random_state は、固定されたランダムな訓練用データを返すようにできる。

Isolation Forest を使用して、作成した訓練データを「model.fit(X_train)」を用いて学習する。「model.fit(X_train)」メソッドは、Isolation Forest による機械学習モデル (model) を訓練データ (X_train) を使って学習させるためのメソッドである。2 割のテストデータに対する予測の正解率は 90.03% だった。

7. 離床検知装置への適用

7.1 適用方法

Isolation Forest を用いた教師なし機械学習を離床検知装置へ適用した。本装置では、動画を読み込み、「cv2.createBackgroundSubtractorMOG2」メソッドによる背景差分を 1 秒ごとに実行し、「cv2.bitwise_and」メソッドを用いて前景部分のみを抽出した後、「cv2.ORB_create」メソッドを使用して特徴点を検出し、記述子を計算して CSV に保存する。次に、累積寄与率 90% を基準に PCA で次元削減を行い、保存済みの学習モデルを読み込んで予測を実施し、最終的に結果を保存する。

7.2 結果・考察

本研究は離床検知装置の改善を目的として行ったが、結果として想定していた結果が得られなかった。Isolation Forest の出力がしきい値を超えたかどうかで、異常を検出した場合は「-1」、正常の場合は「1」と出力したが、すべての動きにおいて正常を意味する「1」を出力するという結果となった。すべての動きにおいて正常と認識する結果となった原因として次の 3 つを考えた。

撮影条件の違いにより、学習時のデータと実験時のデータに差異が生じた。特にカメラの画角の違いが、特徴点の抽出や異常検知の精度に影響を与えた可能性がある。

次に、PCA の条件がファイルごとに最適化されていたことが問題となったと考えられる。PCA による特徴抽出がデータごとに異なる基準で行われたため、一貫した特徴量の取得が難しくなり、異常検知モデルの判断基準が不安定になった可能性がある。

さらに、過学習の影響と交差検証の不足も精度低下の一因と考えられる。モデルが特定のデータに過度に適応したことで、新しいデータへの汎化性能が低下し、

異常検知の精度が不安定になった。適切な交差検証を行わなかったため、モデルの性能を過信し、実運用時の誤検知率が上昇した可能性がある。

8.LINE メッセージを Raspberry Pi に転送するシステム構成について

8.1 システム構成

従来の装置では LINE Notify を用いたため、Raspberry Pi から LINE へ一方方向で通知を送ることしかできなかった。これでは双方向でやり取りができず、ユーザから任意のタイミングで Raspberry Pi に情報を送ることができないため、システムに自由度がなかった。そのため、LINE から Raspberry Pi にメッセージを送り、その情報をもとに、Raspberry Pi が動作するプログラムの作成に取り組んだ。また、Raspberry Pi が写真を撮った際には、LINE Notify を用いて撮ったものを送るようにした。このシステムを作成するのに図 4 に示す 6 つのモジュールを用いた。

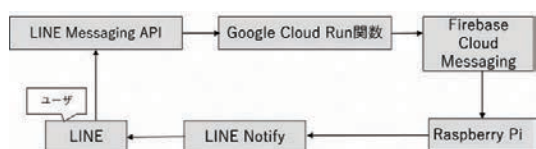


図 4 システム全体図

8.2 各モジュールについて

「LINE」はスマートフォンからデータの送受信が可能なモバイルメッセンジャーアプリケーションである。LINE を使った双方向通信を実現するため「LINE Messaging API」を利用した。Raspberry Pi は外部から直接アクセスしにくい。そのため、Raspberry Pi と LINE を繋ぐためにサーバーを用意しておく必要がある。サーバーの運用、保持をなくすため、サーバーレス関数である「Google Cloud Run 関数」を利用した。「Firebase Cloud Messaging」はプッシュ通知するために利用した。「Raspberry Pi」は Firebase Cloud Messaging からメッセージを受信し、受け取った情報をもとに処理を行う。

8.3 システム作成の結果

本研究を通じて LINE から Raspberry Pi へメッセージを送り、受信したメッセージによって Raspberry Pi での処理を選択するということを確認する実験を行った。

LINE で「a」というメッセージを送り、Raspberry Pi で「a」を受信したら「1+5」という演算をしてその結果を表示する。LINE では図 5 のように、「a」をコールバックした。図 6 のように Raspberry Pi 側では Received data 内

に「a」を受信出来たら実行結果「6」を示している。このことよりユーザからのメッセージによって Raspberry Pi の処理を変更することが可能になった。将来的に本装置の設定をユーザが直接変更できるようにしたい。

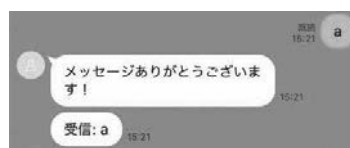


図 5 LINE「a」送信

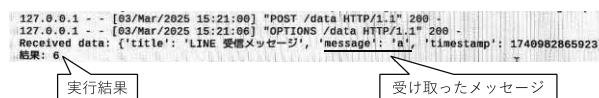


図 6 「a」送信後の実行結果

9.おわりに

本研究の目的は、以下の 2 点であった。

- (1) 教師なし学習を用いて離床を検知する。
- (2) LINE から Raspberry Pi への通知を可能にする。

ベッド上の状態を、背景除去した動画から ORB 記述子を用いて特徴を表現し、そのデータを主成分分析にかけ Isolation Forest アルゴリズムによる教師なし学習を行った。テストデータによる正解率は、90.03%であったが、装置に組み込んだ状態では異常状態を正常状態と判断した。一連の流れは構築できたが、学習データの作成条件、過学習の可能性を見直したい。

LINE から Raspberry Pi への通知は、LINE Messaging API や Cloud Run 関数などを用いて実現できた。これを使い、カメラの位置設定ができるよう機能を追加したい。

参考文献

- 1) 岩本誠斗, 吉松楓華, 江口智弘: 高齢者施設で使用する離床検知装置の改善と実用的評価, 令和4年度(2022年度)卒業研究概要集, (2022), pp61-64.
- 2) 上田真史, 南匠, 江口智弘: 離床検知装置への機械学習の適用, 令和5年度(2023年度)卒業研究概要集, (2023), pp65-68.
- 3) 株式会社 PROMPT:【0から学ぶ AI】第 280 回:特徴量抽出(SIFT, SURF, ORB) ~画像から特徴点を抽出する手法を解説(2025/02/08)
<https://service.ai-prompt.jp/article/ai365-280/>
- 4) 藤吉弘亘: 画像局所特徴量 SIFT と最近のアプローチ(2025/02/28)
https://www.jstage.jst.go.jp/article/jjsai/25/6/25_753/_pdf

MATLAB/Simulink × ラズパイを用いた ライントレース PID 制御安定化検討

*北村昌太郎, *渡邊天, **米田琉世, 塚本晃史(指導教員)
*電子情報技術科 ・ **情報システム技術科

本研究ではモビリティ分野で広く使用される MATLAB/Simulink を用いて、ライントレースロボットの PID 制御による安定性向上に関する研究を行った。赤外線センサでラインを検出し、機器をラインに追従するように走行させた。そして、比例(P)、積分(I)、微分(D)の各ゲインを組み合わせた制御を適用することで、より精度の高い軌道追従に取り組んだ。また、最適な目標値の設定、制御プログラムの改良、PID ゲインの変更などにより制御の安定性が向上することを確認した。そして、更なる PID 制御の精度および安定性の向上を目指した。

1. はじめに

近年、自動車には数多くのマイコンが搭載されており、エンジンの制御を始めとする様々な機能を持つマイコンを用いた組み込み制御を行っている。

これらのシステムは早期開発、検証が可能なモデルベース開発と呼ばれる手法を用いて開発されている。中でも MATLAB という開発環境は、ISO 26262¹⁾ や ISO 21434²⁾ などの自動車業界の国際規格に準拠しており、多くの自動車メーカーやサプライヤーが導入している。

本研究では、モデルベース開発として、センサでラインを検出し、そのラインに沿って走行するライントレースロボットの制御を、検証をしながら設計開発した。

一般的なライントレースは ON/OFF 制御で行われることが多いが、今回は比例制御(P)、積分制御(I)、微分制御(D)を組み合わせ、それぞれが持つ特性を活用して目標値への収束を図ることができる PID 制御を採用することで、より高精度かつ安定した制御を目指した。

また、SCSK ニアショアシステムズ株式会社との共同研究として本研究に取り組み、研究に使用したソフトウェアの技術研修を受講すると共に、適切な運用方法や解析手法についてアドバイスをいただいた。

2. システム概要

本研究では MATLAB, Simulink というソフトを使用した。MATLAB, Simulink で作成したプログラムの出力をロボットのラズパイに送信することで、ロボットの動作を制御する制御命令とすることができる。

ライントレースを実現するには、黒のラインとライン外の白の面を往復前進するようなプログラムを作成すればよい。その制御プログラムを、主に Simulink 上でブ

ック線図として作成した。

2.1 使用ソフト

1) MATLAB(matrix laboratory)

MATLAB は MathWork 社が開発している科学技術計算のための数値計算ソフトウェアである。数学計算、プログラミング、データの可視化が得意という特徴があり、簡単な操作で計算問題を解決することに特化している。

2) Simulink

Simulink は MATLAB 上で動作し、計算や出力等の特徴を持ったブロックによりブロック線図を構築できる。このブロック線図は、コーディングによるプログラミングと比べ、処理やデータの流れが視覚的に分かりやすいという特徴を持つ。この特徴によりプログラムの効率と柔軟性が向上するため、組み込みシステムの中でも特にロボット業界やモビリティ業界で使用されるツールとなっている。以下の図 1 にブロック線図の例を示す。

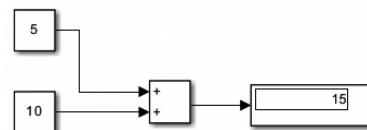


図 1 計算ブロック線図の例

2.2 使用ハード

使用したロボット³⁾には、付属していたモータや赤外線センサに加え、Raspberry Pi4 モデル B を取り付けました。このロボットの写真を図 2 に示す。また、図 3 にハードウェア間の構成を示すブロック図を示す。

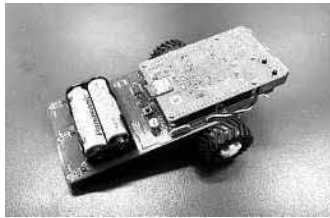


図2 ライントレースロボット

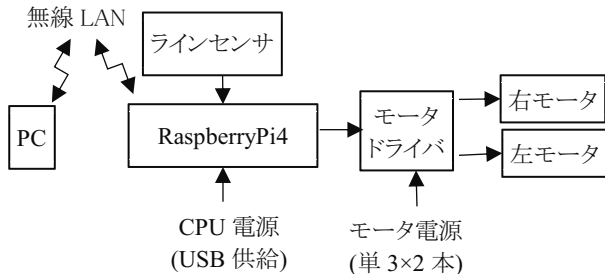


図3 ハードウェア構成のブロック図

PID 制御では、制御量を収束させるための基準として目標値を定める必要がある。今回使用したセンサの出力値は、黒と白を読み取り時、それぞれ黒は 253、白は 147 である。センサの出力より、黒と白の出力値のちょうど中間の 200 に目標値を設定した。これは、左右それぞれの方向のカーブで制御量がかわらないようにするためである。

2.3 参照したプログラム

ロボットの販売元である ZEP エンジニアリング株式会社が公開している資料を基に、PID 制御プログラムを作成した。そのブロック図を図 4 に示す。

例として、図 5 に、作成した PID プログラム(P=3, I=2, D=1)を示している。このように、PID の各ゲインを組み合わせた出力値でロボットの制御を行った。

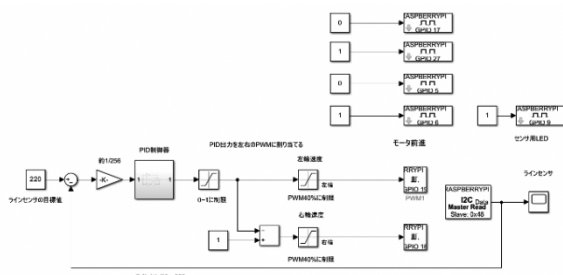


図4 参照したプログラムのブロック線図

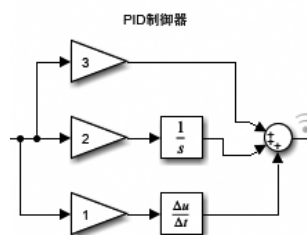


図5 PID プログラムのブロック線図

3. 改善点とその検討

3.1 制御プログラムの不安定性

参照したプログラムには以下の欠点が存在した。

- ① プログラム終了後にロボットが動き続けてしまう
- ② ロボットの動作による振動やブレが激しい
- ③ 積分値が過剰に溜まり、異常な数値を示してしまう
- ④ ロボットの動作がカクカクしている
- ⑤ 偏差が 0 のときのモータ動作の改善

実際にロボットをコースで走らせ、様々な検証を行い、多観点から制御の安定化を目指す必要がある。

3.2 改善策の検討

3.2.1 制御プログラムの改善策

3.1 で述べたプログラムの欠点に対し、それに対応するブロックを追加し、プログラムの改善に取り組んだ。プログラムの改善に関して、以下のとおり実施した。

- ① 停止プログラムの実装
 - ② 不完全微分ゲインブロックの実装
 - ③ 積分飽和対策プログラムの実装
 - ④ 一次遅れゲインの実装
 - ⑤ 目標値と一致しているときのロボットの動きの改善
- 上記の改善の結果については4.1で述べる。

3.2.2 安定化基準の検討策

制御の安定化には、各 PID ゲインの効果を比較し、理解することが大切である。そのため、直線や曲線を含めた図 6 のコースを設定し、2通りの検証をした。

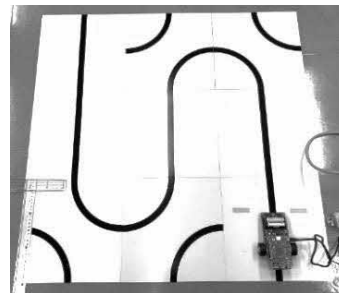


図6 設定したコース

① 目標値に対する二乗偏差の測定

プログラムに図 7 のようなブロック群を追加した。このブロック群により、目標値との偏差を U として、それを MathFunction(数学関数)ブロックで 2 乗し、Integrator(積分)ブロックで積分することで、二乗偏差の累計を確認する。各 PID ゲインを変化させ、15 秒間ラインレースを行い、黒のラインからのズレ具合を確認することで、各 PID ゲインに対する偏差の程度を表にまとめ、評価した。

② 走行距離測定

各 PID ゲインを変化させ、15 秒間にロボットが走行した距離を測定した後、基準点からの距離を定規で計測し、それをその PID ゲインの走行距離とする。走行距離と各 PID ゲインの関係性を表にまとめ評価した。

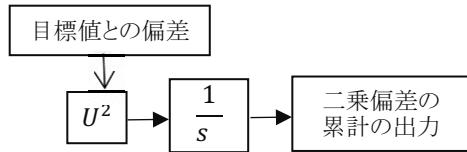


図 7 標準偏差測定のプロック図

4. 改善結果

4.1 制御プログラムの改良結果

① 停止プログラムの実装

モータの回転と GPIO ピンの出力の関係性より、プログラム終了直前にモータの回転を停止するプログラムを実装した。

② 不完全微分ゲインの実装

微分ゲインブロック ($\Delta u/\Delta t$) を不完全微分ゲインブロック ($s/(0.1s+1)$) に変更した。これにより、高周波成分による過剰な制御入力を抑制できる。また、制御信号に関係する低周波成分を加えることが出来るため、振動やノイズを抑制しつつ、必要な微分効果を維持することができる。しかし、後述するように微分ゲインはゲインを 0 にしたため、結果的に不採用となった。

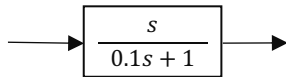


図 8 不完全微分ブロック

③ 積分飽和対策

ワインドアップ現象(上下制限による積分量過剰の状態)により、積分値が過剰に溜まってしまふのを防ぐため、図 9 のように制御量が飽和した際、積分要素を停止する対策プログラムを追加した。しかし、以降の測定実験では積分飽和が起きなくなったため、後述の実験の結果に寄与したとは言えない。

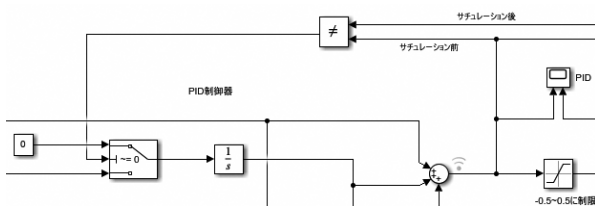


図 9 積分飽和対策のプロック群

④ 一次遅れゲインの実装

図 10 の一次遅れゲインブロックを目標値とセンサの出力値との偏差に対して追加した。急激な変化を抑え、より緩やかな動きを実現するためである。しかし、カーブなどの大きい動作に支障が生じ、精度が低下する問題が発生したため、採用には至らなかった。

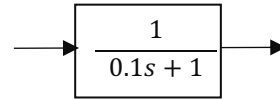


図 10 一次遅れブロック

⑤ 目標値と一致しているときのロボットの動きの改善

図 11 の既存のプログラムでは目標値とセンサの出力が一致しているときに右輪が正転し、左輪が停止してしまっていた。これに対し、図 12 のようにプログラムを変更し、目標値とセンサの出力が一致しているとき、両輪が同じ出力で正転するようにした。

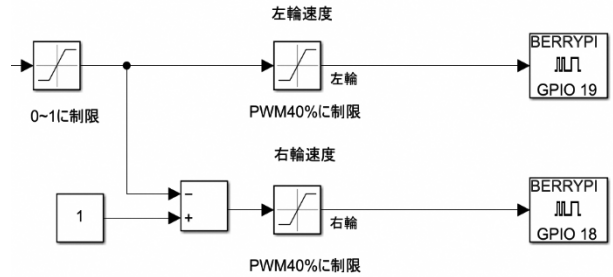


図 11 変更前のプログラム

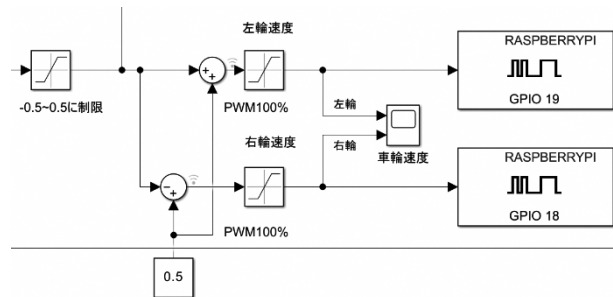


図 12 変更後のプログラム

これらの改良を通じて、より PID 制御の安定性を向上させたプログラムを作成した。

4.2 安定化基準の測定結果

3.2.2 で検討した安定化基準について測定実験を実施し、PID の各ゲインについて偏差と走行距離という二点から安定性の評価を行った。なお、D ゲインは振動を引き起こし、偏差が大きくなるため、今回の実験では D ゲインを 0 に設定した。

4.2.1 偏差測定

PID の各ゲインを 0 から徐々に増加させてロボットのラインレースを観察したところ、 $P=0.4$ 、 $I=0.8$ 付近のゲインでロボットは比較的安定的かつ効率的なラインレースを行うという結果が得られた。

各 PID ゲインを $P=0.3\sim 0.5$ 、 $I=0.7\sim 0.9$ の範囲で設定し、コースを 5 回ずつ走らせ、その平均の偏差測定を行った。その実験結果を表 1 に示す。そして、図 13 に縦軸を P ゲインごとの偏差の合計、横軸をその P ゲインという棒グラフとしてまとめ、P ゲインによる偏差量の比較を行うと、P ゲインが小さいほど偏差が少なく、安定性が向上することが分かった。

表 1 P, I ゲインでの偏差結果(桁は 10^4 単位は無し)

I (積分ゲイン)	P(比例ゲイン)				
	0.3	0.35	0.4	0.45	0.5
0.7	3.265	3.405	3.739	3.539	3.541
0.75	2.665	3.112	3.322	3.623	3.737
0.8	3.067	3.287	3.489	3.767	3.575
0.85	2.998	3.010	3.080	3.316	3.529
0.9	3.121	3.229	3.572	3.510	3.521

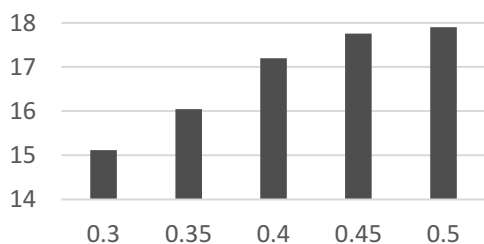


図 13 各 P ゲインによる二乗偏差の合計の比較

4.2.2 走行距離測定

結果を表 2 にまとめた。そして、図 14 の縦軸を走行距離、横軸を P ゲインという折れ線グラフにして確認した。積分制御(I)の値が大きいほど走行距離が長くなる傾向が確認された。

表 2 P, I ゲインでの走行距離結果(単位は[cm])

I	P				
	0.3	0.35	0.4	0.45	0.5
0.7	24.84	25.22	26.76	25.44	26.96
0.75	23.18	28.20	25.90	27.26	29.30
0.8	30.80	29.58	32.82	30.40	30.98
0.85	23.96	28.80	29.42	31.26	29.98
0.9	30.40	29.92	32.94	32.32	32.10

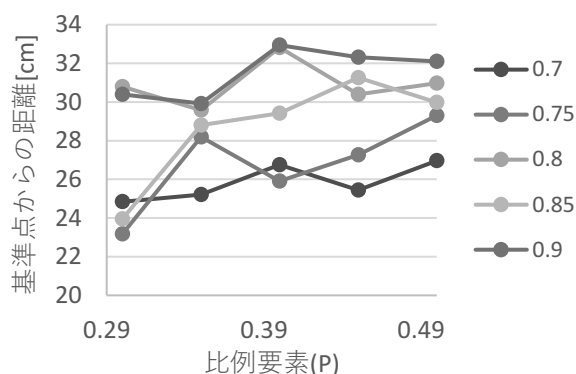


図 14 各 P ゲインの走行距離比較

5. おわりに

本研究では、MATLAB/Simulink を用いたラインレースロボットの制御プログラムの改良を行い、PI 制御の安定化を実現した。偏差測定の結果、比例要素を増加させると制御応答が向上する一方で、目標値に対するオーバーシュートや振動が増加し、安定性が低下することが確認された。また、積分ゲインの適切な調整により、カーブ走行時の目標値への復帰速度や走行距離が向上することが明らかになった。

今後の課題として、より高度な制御手法として I-PD 制御の導入や、環境変化に適応可能な適応型制御の検討が挙げられる。そして、限界感度法⁴⁾などの PID ゲイン調整法を用い、制御プログラムのさらなる最適化を行い、ロボットの走行性能向上を目指す。

最後に、今回の研究活動においてご支援およびご協力いただいた SCSK ニアショアシステムズ株式会社様、及び尚綱大学短期大学部の牧岡准教授に心より感謝を申し上げます。

参考文献

- 1) MathWorks, "MATLAB および Simulink における ISO 26262 のサポート - 自動車業界標準 - MATLAB & Simulink", <https://jp.mathworks.com/solutions/automotive/standards/iso-26262.html>, (参照 2024 年 5 月 8 日).
- 2) MathWorks, "MATLAB、Simulink、および Polyspace における ISO 21434 のサポート - 自動車業界標準 - MATLAB & Simulink", <https://jp.mathworks.com/solutions/automotive/standards/iso-21434.html>, (参照 2024 年 5 月 8 日).
- 3) ZEP エンジニアリング, "[VOD/KIT] MATLAB/Simulink×ラズパイで学ぶロボット制御入門", <https://www.zep.co.jp/kzenyouji/movie/mz-matrobe-on2/>, (参照 2024 年 9 月 6 日).
- 4) こんとろラボ, "限界感度法の利点・欠点・使い所。実はアテにしちゃダメ!?", <https://controlabo.com/ultimate-sensitivity-method/>, (参照日:令和 7 年 3 月 6 日).

マイコンを用いた外部デバイスとのシリアル通信ソフトの改良

*後藤颯斗, *中村清太郎, **宮内理玖斗, 塚本晃史(指導教員)

*電子情報技術科**情報システム技術科

本研究では昨年度に作成した、「温度湿度センサデバイス(XY-MD02)」と通信を行い、温度湿度情報を取得するプログラムの改良を行った。使用するマイコンは昨年度と同様、STM32, H8, PIC の 3 つのマイコンを用いる。昨年度のプログラムを改善したところ、昨年度は未達成だった項目をほぼすべて達成することができた。

1. はじめに

各種製造装置などのシステムはコンピュータやセンサなど数多くの機器で構成されている。機器同士は様々な方式で接続されており、相互に情報のやり取りを行っている。その接続方式の中でもシリアル通信の割合は高い。しかしシリアル通信はハード・ソフト共に種類が多く、新規に製品を開発する際に適切なハード・ソフトを選定しなければコストアップや品質低下などを招いてしまう。

昨年度の研究では、STM32, H8, PIC の 3 つのマイコンを用いて、「温度湿度センサデバイス(XY-MD02)」とシリアル通信を行い、温度情報を取得するプログラムを作成した。そして、プログラムコードの処理時間や設計データから考察し、最適なプログラムの研究を行った。昨年度の達成状況を表 1 に示す。

表 1 昨年度の各マイコンの達成状況

	STM32	H8	PIC
温度情報取得	×	○	○
湿度情報取得	×	×	×
処理の高速化	○	○	○
処理の効率化 (割り込みの活用)	○	×	×
情報の画面表示	×	○	×

本年度の研究の目標は、表 1 にある未達成の項目を達成することである。

2. 測定装置と構成

2.1 構成部品

①マイクロコントローラ

本研究では温度湿度センサからのデータをマイクロコントローラで処理し、PC または LCD を利用して表示させる。今回使用するマイクロコントローラの仕様を表 2 に

示す。

表 2 マイクロコントローラの仕様 1), 2), 3)

マイコン 名称	STM32 Nucleo Board STM32F411RET6	AKI-H8 /3052F	PIC16F 1937
動作周波数	100MHz	25MHz	8MHz
bit 数	32bit	16bit	8bit
ROM	512KB	512KB	14.25KB
RAM	128KB	8KB	512B
動作電圧	3.3V	5V	5V
FPU 搭載の有無	あり	なし	なし

②温度湿度センサ

温度湿度センサとして SHT20(温度及び湿度センサ)が内蔵されている XY-MD02(SAH Electronics)を使用した。その外形を図 1 に、仕様を表 3 に示す。



図 1 XY-MD02 の外形

表 3 XY-MD02 の仕様 4)

通信プロトコル	Modbus-RTU
通信速度	9600bps
通信可能距離	1000m(RS485)
動作電圧	DC 5V~30V
温度範囲(保証値)	-40~60°C
消費電力	0.2W 以下

このセンサは Modbus_RTU プロトコルに準拠した温度湿度情報要求コマンドを受信することによって、デー

タを返信する。取得した温度湿度情報は文字列として表示する。例えば 23.1[°C]であれば 231 とくるので、100 の位の値を取得し文字に変換する。この処理を 10 の位、1 の位と行い、10 の位と 1 の位の間に小数点を追加した文字列をディスプレイに表示する。

2.2 システム構成

STM32, H8 のシステム構成図を図 2 に、PIC のシステム構成図を図 3 に示す。

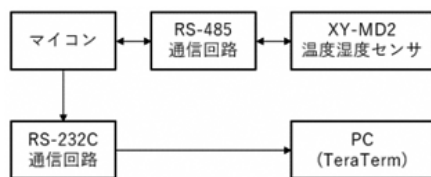


図 2 STM32, H8 のシステム構成図

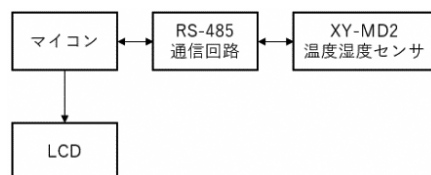


図 3 PIC のシステム構成図

3. プログラムの改善

3.1 参考にする昨年度のプログラム

昨年度のプログラムは温度情報の取得、表示が出来るマイコンがあり、湿度も同様に同じ処理を行い、情報の取得、表示を行う。

処理の効率化は STM32 だけが成功していた。H8 と PIC は STM32 の処理を参考にして、割り込み処理を構成することとした。

3.2 改善結果

3.2.1 温度、湿度情報の改善結果

STM32, H8, PIC それぞれのマイコンで同じ処理を追加した。温度、湿度情報取得の改善前のプログラムを図 4 に、改善後のプログラムを図 5 に示す。

```
uint8_t msg[] = {0x01, 0x04, 0x00, 0x01, 0x00, 0x01, 0x60, 0x0A, };
```

図 4 改善前の温度湿度情報取得プログラム

```
uint8_t msg[] = {0x01, 0x04, 0x00, 0x01, 0x00, 0x02, 0x20, 0x0B};
```

図 5 改善後の温度湿度情報取得プログラム

図 4, 5 の 8 ビットの整数を宣言する `uint8_t` に `msg` という配列を宣言する。この配列に 8 バイト分のデータを格納する。

図 4 の改善前のプログラムは温度情報だけを要求していたのに対し、図 5 の改善後のプログラムは温度、湿度情報を要求する 8 バイト分のデータに書き換えた。これにより、温度、湿度情報を取得することが可能になった。

湿度情報を追加したので、湿度情報を作成するプログラムも追加した。温度を処理するプログラムを図 6 に、温度湿度を処理するプログラムを図 7 に示す。

```
tempe_buf =
(UART_RcvData[3]<<8 | UART_RcvData[4]);
```

図 6 温度を処理するプログラム

```
tempe_buf =
(UART_RcvData[3]<<8 | UART_RcvData[4]);
humi_buf =
(UART_RcvData[5]<<8 | UART_RcvData[6]);
```

図 7 温度湿度を処理するプログラム

`RcvData` は温度湿度センサから通信で受信したデータが保存されている配列変数で、`RcvData` の 3 バイト目と 4 バイト目を取り出して温度情報の作成、`RcvData` の 5 バイト目と 6 バイト目を取り出して湿度情報の作成を行っている。

取り出した温度、湿度情報を TeraTerm で出力した画面を図 8 に示す。

```
Temperature = 21.2
Humidity    = 54.6
Temperature = 21.2
Humidity    = 54.6
Temperature = 21.2
Humidity    = 54.8
```

図 8 TeraTerm の画面

3.2.2 処理の効率化の改善結果

H8 と PIC の改善前のプログラムは温度、湿度情報を要求するための 8 バイト分のデータの送信と受信を実行するプログラムが `main` 関数内で処理が行われていた。このプログラムでは、データの送信や受信の処理が終了しない場合、次の処理に移行できなくなる。これにより、`main` 関数内の処理が停止してしまうため効率化が

できていない状態であった。

改善後のプログラムはタイマ割り込み処理と受信割り込み処理を活用し、効率化を図ったプログラムとなっている。改善前のフローチャートを図9に、改善後の main 関数とタイマ割り込み、受信割り込みのフローチャートを図10、図11、図12に示す。

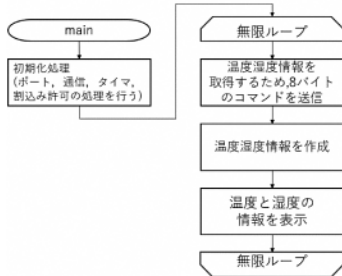


図9 改善前のフローチャート

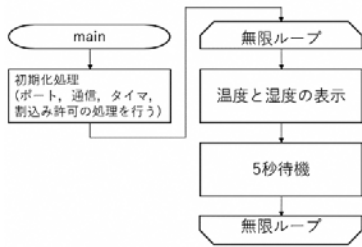


図10 main関数のプログラムのフローチャート

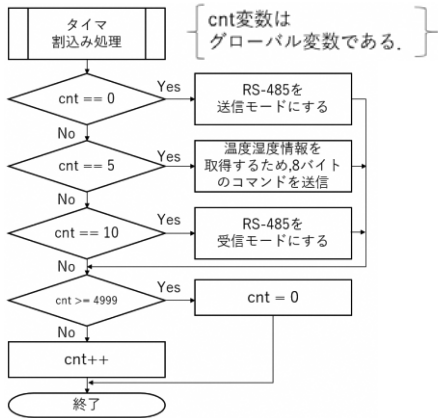


図11 タイマ割り込み処理のフローチャート

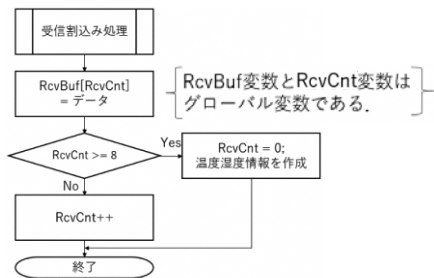


図12 受信割り込み処理のフローチャート

main 関数では温度と湿度情報を表示して 5 秒待機する動作を繰り返す無限ループとなっている。このループが繰り返される中でタイマ割り込みと受信割り込み処理が発生し、それぞれの処理を行っている。

これにより、main 関数内で他の処理を実行することが可能となった。

3. 2. 3 PIC の画面表示の改善

PIC マイコンは通信ポートが 1 ポートしかないため、センサと通信すると温度と湿度の情報を表示することができない。そのため、LCD を用いて画面表示を行う。図13のフローチャートにLCDの処理を追加する。LCDを用いた画面表示のフローチャートを図13に示す。

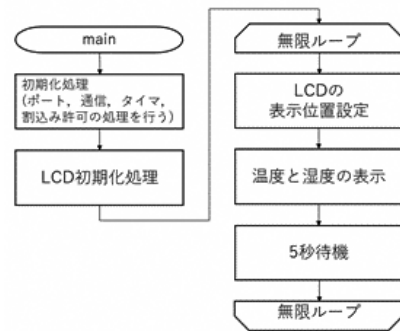


図13 LCDを用いた画面表示のフローチャート

5 秒周期で LCD の画面の更新を行っており、5 秒毎に温度湿度情報を取得して情報の表示を行う処理の繰り返しとなっている。RS485 の送受信モード切り替え信号、送信信号、受信信号をオシロスコープで観測した波形を図14に示す。

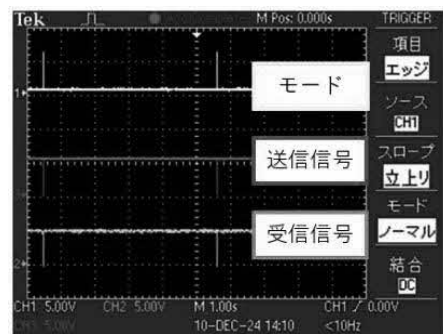


図14 オシロスコープ波形(時間軸 1[s/div])

'H'になっている間隔が 5 マスであり、1 マス 1 秒と設定しているため、温度、湿度情報を取得する処理の周期が 5 秒ということがわかる。取得した温度、湿度情報

の表示結果を図 15 に示す。



図 15 LED の画面

3.2.4 STM32 の画面表示の改善

昨年度は TeraTerm に出力するプログラムを作成しておらず、本年度からプログラムを追加した。

追加したプログラムについては図 8 にある通り、簡易的であるため省略する。

3.2.5 ハードウェアの修正

データの受信を正しく行うためにプルアップ抵抗を受信するピンに追加した。プルアップ抵抗を追加する前と後のオシロスコープの波形を図 16 に示す。

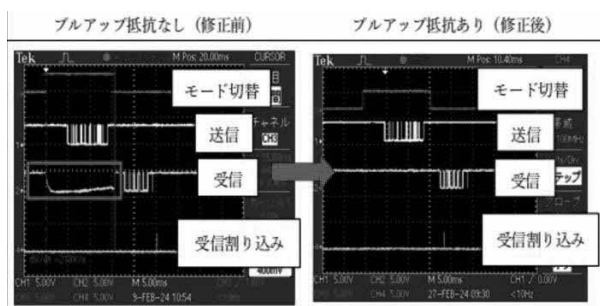


図 16 プルアップ抵抗によるオシロスコープの波形

修正前は受信ラインがハイインピーダンス状態となり、波形が不安定で誤動作が発生していた。修正後は受信データが正しく取得され、割り込み処理も正常に動作した。これにより、モード切り替え、送信、受信、といった一連の通信動作が確実にできるようになった。

4. おわりに

本研究では、STM32、H8、PIC の 3 つのマイコンを用いて、昨年度に作成された「温度湿度センサデバイス (XY-MD02)」とシリアル通信を行うプログラムの改良を行った。本研究で達成した項目を表 4 に示す。

表 4 本研究で達成した項目

	STM32	H8	PIC
温度情報取得	◎	○	○
湿度情報取得	◎	◎	◎
処理の高速化	○	○	○
処理の効率化 (割り込みの活用)	○	×	◎
情報の画面表示	◎	○	◎

表 4 は各マイコンの課題の達成状況を示している。○は昨年度達成した項目、◎は今年度達成した項目である。

本研究では掲げている目標のほぼすべてを達成することができた。温度、湿度情報の取得はセンサに要求する 8 バイト分のデータを書き換えることで実現した。STM32 の情報の画面表示として、プログラムの追加を行い、PIC では、LCD を追加することで画面表示が可能となった。処理の効率化は STM32 の処理を参考に H8 と PIC に組み込んだ。

本研究の課題としては、H8 は処理の効率化の改善ができなかったことが挙げられる。また、STM32 の内部プルアップ抵抗が利用できるか確認を行う。そして、今回のセンサとの通信は単体で行われていたので、次は複数のセンサとの通信を試みる⁵⁾。

参考文献

- 1) “NUCLEO-F411RE, 製品概要”, STMicroelectronics <https://www.st.com/ja/evaluation-tools/nucleo-f411re.html>, (参照 2025 年 1 月 28 日)。
- 2) “AKI-H8/3052 マイコンボード”, マニュアル, <https://akizukidenshi.com/goodsaffix/AKI-H8-3052F-fi-pro.pdf>, (参照 2025 年 1 月 28 日)。
- 3) “PIC マイコン PIC16F1937-I/P”, データシート, <https://akizukidenshi.com/goodsaffix/pic16f193X.pdf>, (参照 2025 年 1 月 28 日)。
- 4) “M5Stack Modbus 温度, 湿度測定器 XY-MD02 の使い方”, ロジカラブログ, <https://logikara.blog/modbus-xy-md02/>, (参照 2025 年 1 月 28 日)。
- 5) “RS-485 配線ガイド”, supremainc.com, https://kb.supremainc.com/knowledge/doku.php?id=ja:tc_technology_rs-485_wiring_guidelines (参照 2025 年 3 月 6 日)。

準同型暗号を用いたデータ解析の応用

松本海星^{*}, 柳澤宏樹^{*}, 毛利陸大^{**}, 里中孝美(指導教員)

^{*}電子情報技術科, ^{**}情報システム技術科

本研究では, IoT デバイスのセンサデータをクラウドサーバ上で安全に活用するために, 準同型暗号方式を用いたデータ処理システムを検討した. TenSEAL, pyFHE の Python ライブラリを用いて BFV 方式 と CKKS 方式の暗号処理のプログラムを実装し, パラメータ設定, 鍵生成, 符号化, 暗号化, 準同型演算などの処理時間を比較し, ボトルネックとなる処理の分析を行なった. また, 本データ処理システムでは AES 暗号と CKKS 暗号を併用する方式を採用し, CKKS 暗号のみの従来法に比べて計算時間は増大せず, IoT 端末で暗号計算の負担を軽減できることを確認した.

1. はじめに

近年, IoT デバイスが普及して, データの収集から活用までをクラウド上で完結させることが可能となったが, データ漏洩のリスクがあり, 個人情報保護という観点で問題となっている. これらのデータを安全に利活用するため, 準同型暗号方式が注目されている.

本研究では, pyFHE¹⁾ と TenSEAL を用いて暗号処理を実装し計算時間を分析し, 応用として, クライアント・サーバによるセンサデータの統計処理システムを構築する. 準同型暗号方式には, CKKS(Cheon, Kim, Kim, and Son)²⁾方式と BFV(Brakerski-Fan-Vercauteren)³⁾方式があるが, 計算能力の低い IoT デバイスへの実装は実用面で現実的でないと考えられる. 先ず, 準同型暗号処理の高速化に向けて, pyFHE の Python ライブラリを用いて, CKKS 方式と BFV 方式のパラメータ, 鍵生成, 符号化, 暗号化, 準同型演算などの主要なステップの処理時間を計算して, 両方式の比較とボトルネックとなる処理の分析を行なった.

次に, クライアント端末での暗号処理の計算負荷を軽減するため, Lauter⁴⁾らが提案した共通鍵暗号 AES⁵⁾(Advanced Encryption Standard)と準同型暗号 CKKS を用いる方式を採用する. 本システムではクライアント端末は Arduino で測定した脈拍センサのデータを収集し, AES 暗号化したセンサデータをクライアントとサーバ間で送受信し, サーバは AES 暗号を解読して, 再度 CKKS 暗号でセンサデータの解析を行う. CKKS のみを用いた方式とハイブリッド方式の処理時間を比較してハイブリッド方式の有効性を確認する.

2. 準同型暗号による秘匿計算技術

2.1 準同型暗号

準同型暗号とは, データを暗号化した状態で演算を行い, 復号時に正しい演算結果が得られるように構成された暗号方式である. 平文 m と n があつたとき, 式(1), (2)の関係が成り立つような暗号方式である. $+$, \times は平文に対する加算, 乗算演算で, \oplus , \otimes は暗号文に対する加算, 乗算演算である.

$$\text{Decrypt}(\text{Encrypt}(m) \oplus \text{Encrypt}(n)) = m + n \quad (1)$$

$$\text{Decrypt}(\text{Encrypt}(m) \otimes \text{Encrypt}(n)) = m \times n \quad (2)$$

$\text{Encrypt}(m)$ は, 平文 m を暗号化することにより得られる暗号文で, $\text{Decrypt}(\text{Encrypt}(m))$ は, 暗号文 $\text{Encrypt}(m)$ を復号して得られる平文を指す.

2.2 CKKS, BFV 準同型暗号方式

CKKS, BFV 暗号方式では, 平文は円分多項式で表現される. メッセージ空間 M と平文空間 P を仮定し, CKKS 方式の暗号方式について説明する. 平文空間 P で用いる円分体の剰余環を $\mathbb{Z}[X]/(X^N + 1)$ とする. 平文は $\mathbb{Z}[X]$ を $X^N + 1$ で割った円分多項式の剰余環で表現される.

$$X^N + 1 = \prod_{j=0}^{N-1} (X - \zeta^{2j+1}) = 0 \quad (3)$$

ただし, $\zeta = \exp\left(\frac{2\pi\sqrt{-1}}{N}\right)$ である. 複素線形空間 $C^{N/2}$ のメッセージ $m = (\alpha_0, \alpha_1, \dots, \alpha_{N-1})$ を平文多項式の係数にパッキングすると, 式(4)の円分多項式 $p(X)$ が得ら

れる. $X=\zeta, \zeta^3, \dots, \zeta^{2N-1}$ を式(4)に代入すると式(5)が得られる. 多項式(4)の係数 α_k が決まれば, z_j は式(6)より求めることができる.

$$p(X) = \alpha_0 + \alpha_1 X + \dots + \alpha_{N-1} X^{N-1} \quad (4)$$

$$z=(p(\zeta), p(\zeta^3), \dots, p(\zeta^{2N-1}))^T \quad (5)$$

$$z_j = \sum_{k=0}^{N-1} \alpha_k (\zeta^{2j+1})^k = \sum_{k=0}^{N-1} \alpha_k s_{jk} \quad (6)$$

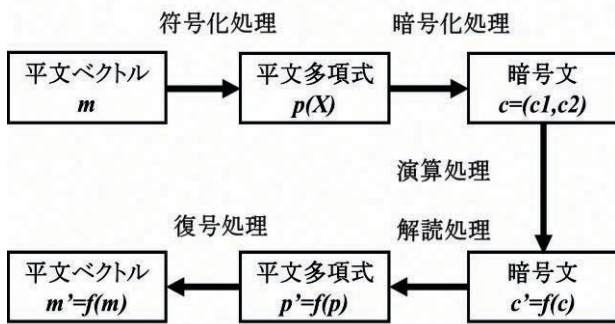


図1 準同型暗号のフレームワーク

図1は準同型暗号のフレームワークを示している. それは, 符号化処理, 暗号処理, 演算処理, 解読処理, 復号処理から構成される.

- (1)符号化処理では, 平文の内容を表す複素数ベクトル m を整数係数多項式 $p(x)$ に変換する.
- (2)暗号処理では, 整数係数多項式を暗号化して整数係数多項式の暗号文 $c=(c1, c2)$ を生成する.
- (3) 演算処理では暗号準同型演算 $c' = f(c)$ を行う.
- (4) 解読処理では計算結果を復号して整数係数多項式 $p'(x)$ を求める.
- (5) 復号処理では 整数係数多項式を複素数ベクトルに m' 変換して最終的に求めたかった結果を得る.

2.3 鍵生成, 暗号, 復号

2.3.1 鍵生成

秘密鍵 sk と公開鍵の一部 a をランダムに作成する. ノイズ $e0$ をガウス分布からサンプルする. 2つの多項式からなる公開鍵 $pk=(pk1, pk2)$ は

$$pk=(-a*sk + e0, a) \quad (7)$$

で与えられる.

2.3.2 暗号と解読

まず, v , ノイズ $e1, e2$ をランダムに作成し, 平文 m から暗号文を生成する. 暗号文 $Encrypt(m, pk1, pk2)=c$ は以下ようになる.

$$c=(c1, c2)=(pk1*v+e1+m, pk2*v+e2) \quad (8)$$

2.3.3 解読

秘密鍵 sk を使った解読について説明する.

$$Decrypt(c1, c2)=c1+c2*sk \quad (9)$$

$$=pk1*v+e1+m+(pk2*v+e2)*sk \quad (10)$$

$$=m+(e0*v+e1+e2*sk) \approx m \quad (11)$$

係数 $v, e0, e1, e2, s$ と計算式 $(e0*v+e1+e2*sk)$ は小さい係数の多項式である.

3. pyFHE による CKKS 方式の実装評価

3.1 CKKS 暗号方式の実装例

pyFHE の Python ライブラリを用いて, CKKS 方式の演算の例を示す. 図2は CKKS 暗号方式の初期化の例で, 秘密鍵, 公開鍵, 再線形化鍵 (relin_key) を設定する. 図3は CKKS 暗号の加算の例である. 平文の message1, message2 が与えられたとき, 平文多項式 plain1, plain2 を生成し, それを暗号化して暗号多項式 ciph1, ciph2 を得る. evaluator.sum では ciph1 と ciph2 との準同型加算の結果 ciph_sum を計算し, decrypted_sum により解読した計算結果を求める.

```
params
=CKKSParameters(poly_degree,ciph_modulus,
big_modulus, scaling_factor)
key_generator = CKKSKeyGenerator(params)
publickey = key_generator.public_key
secret_key = key_generator.secret_key
relin_key = key_generator.relin_key
encoder = CKKSEncoder(params)
encryptor=CKKSEncryptor(params,public_key,secret_key)
decryptor=CKKSDecryptor(params, secret_key)
evaluator = CKKSEvaluator(params)
```

図2 CKKS 暗号方式の初期化

```
message1 = [0.5, 0.3 + 0.2j, 0.78, 0.88j]
message2 = [0.2, 0.11, 0.4 + 0.67j, 0.9 + 0.99j]
plain1 = encoder.encode(message1, scaling_factor)
plain2 = encoder.encode(message2, scaling_factor)
ciph1 = encryptor.encrypt(plain1)
ciph2 = encryptor.encrypt(plain2)
ciph_sum= evaluator.sum (ciph1, ciph2,)
decrypted_sum = decryptor.decrypt(ciph_sum)
```

図3 CKKS 暗号の加算の例

```

明文多項式 (plain1, plain2)
plain1 = -68383068x7 - 391013813x6 + 24216163x5 +
289910292x4 + 115490928x3 - 18981252x2
- 268899682x + 424128020
plain2 = -345901268x7 + 17083128x6 + 152003947x5 +
445602857x4 - 280929570x3 - 138563147x2 +
- 282795311x + 432181084
準同型加算の解読結果 (sum)
sum = [(0.700000005166 + 8.6739440074e-09j),
(0.409999997618 + 0.200000005854j),
(1.180000003902 + 0.670000002561j),
(0.899999994653 + 1.86999999617j)]

```

図4 CKKS 暗号方式の演算結果

図4は CKKS 暗号方式の演算結果である。明文多項式 plain1, plain2 は N=8 のときに式(4)に対応する。図4の sum の値は、message1 と message2 の加算値として得られていることがわかる。

3.2 暗号方式の計算時間の評価

表1は CKKS, BFV 方式の処理時間である。処理時間の比(T1/T2)によれば、CKKS の時間は、BFV に比べて処理時間を要している。パラメータの設定は 445.5 倍で、暗号処理は 56.1 倍で、準同型乗算は 29.4 倍で、復号処理は 16.8 倍となった。同様に、CKKS の準同型加算と準同型乗算の処理時間比は 2.9 倍と 29.4 倍で、BFV より演算時間を要している。この理由は、整数方式の BFV 方式に比べて、CKKS 方式が固定小数点で小数の演算を行うからである。また、CKKS, BFV 方式では、暗号処理時間は 70.16ms, 1.25ms となり符号処理時間の 0.90ms, 0.59ms に比べて大きくなった。準同型加算と乗算の処理時間は、CKKS 方式で 2.63ms, 109.5ms となり、BFV 方式で 0.92ms, 3.72ms となり、乗算が非常に大きくなった。また、明文の加算と乗算は、CKKS 方式で 0.80ms, 0.59ms となり、BFV 方式で 0.76ms, 0.61ms では顕著な差はみられなかった。

図5は暗号乗算の処理時間と次数の関係を示している。表2に、次数が 6~10 で スロット数を 64 から 1024 に増加したとき、BFV の処理時間は、0.012s から 1.250s に増加した。CKKS の処理時間は 0.731s から 12.375s に増加した。スロット数が 1024 のとき、CKKS の処理時間は、BFV の時間は 9.9 倍である。スロット数を増加させたときに、CKKS の処理時間は急激に増大する傾向があった。

表1 CKKS, BFV 方式の処理時間

項目	CKKS T1	BFV T2	T1/T2
パラメータ設定	534.64ms	1.20ms	445.5
鍵生成等準備	3.20ms	2.84ms	1.1
符号処理	0.90ms	0.59ms	1.5
暗号処理	70.16ms	1.25ms	56.1
準同型加算	2.63ms	0.92ms	2.9
準同型乗算	109.47ms	3.72ms	29.4
復号処理	17.81ms	1.06ms	16.8
明文の加算	0.80ms	0.59ms	1.4
明文の乗算	0.76ms	0.61ms	1.2

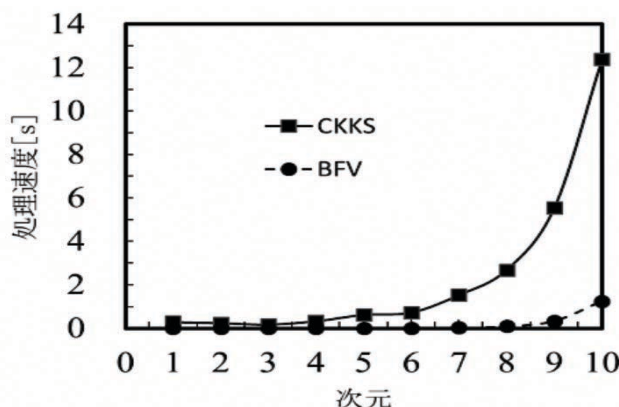


図5 暗号乗算の処理時間と次数の関係

表2 暗号乗算の処理時間と次数の関係

次数	6	7	8	9	10
スロット数	64	128	256	512	1024
BFV[s]	0.012	0.031	0.093	0.340	1.250
CKKS[s]	0.731	1.538	2.695	5.553	12.375

4. CKKS 方式による IoT データの統計処理システム

4.1 脈拍センサの測定と送信

図6は Arduino MEGA 2560 による脈拍センサの計測システムを示している。センサを指先にテープで固定して接触させる。光源の LED により生体内に向けて光を入射し、反射光量の変化として心臓の拍動に伴う動脈血流量の変化を測定する。図7は、脈拍センサの出力波形である。表3の脈拍センサの出力波形の特性において、振幅の最大値は 995mv で、最小値は 26mv で、心拍数は 89 回となった。Arduino は脈拍センサのデータを取得して、クライアント PC(Xeon E-212 1G CPU 3.4GHz)で暗号化して、サーバにデータを転送する。

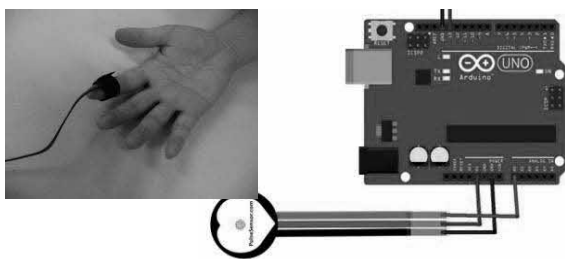


図6 Arduino による脈拍センサの計測システム

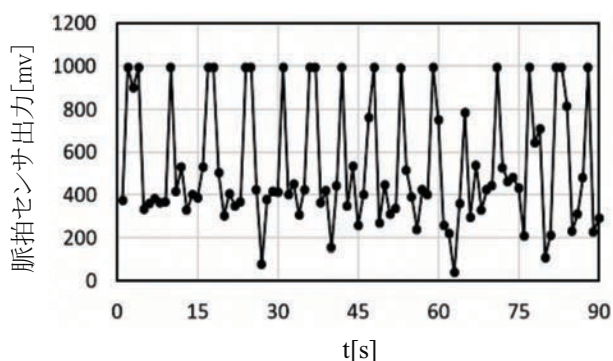


図7 脈拍センサの出力波形

表3 脈拍センサの出力波形の特性

最大値	最小値	P-P 振幅	心拍数
995mV	26mV	969mV	89

4.2 センサデータの通信方式

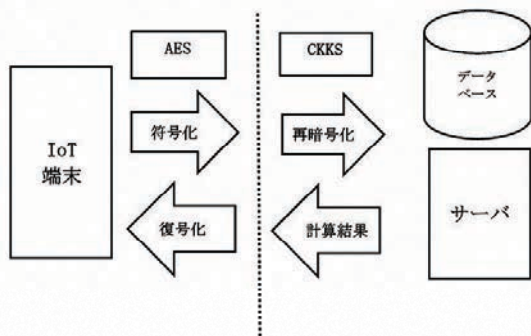


図8 AES と CKKS 方式の IoT データ処理システム

図8のように IoT 端末で暗号計算の負担軽減のため、AES と CKKS 方式を組み合わせた IoT データ処理システムを検討した。クライアント側は AES 暗号方式でデータを暗号化して、受信側で読後、CKKS の暗号演算で心拍数の平均値を計算し、計算結果を AES で暗号化し、クライアントに送信する。表4は、ハイブリッド暗号方式(A)とクライアントとサーバ側でCKKSを用いる方式(B)の場合でデータ件数と暗号時間、計算時間の平

均を示している。1件のデータ数は毎分428個で、件数を128から1024に増加させても、方式AとBで通信による時間の差異はなく、件数が1024であるときに、暗号時間は方式Aで9.4ms、方式Bで10.9msとなり、計算時間は方式Aで56.2ms、方式Bで50.0msとなり、それぞれ同程度であった。

表4 暗号方式による計算時間の比較

処理件数	暗号化時間(ms)		計算時間(ms)	
	方式A	方式B	方式A	方式B
128	10.9	4.7	37.5	37.5
256	9.4	11.0	42.2	31.2
512	12.5	9.4	46.9	45.3
1024	9.4	10.9	56.2	50.0

5. まとめ

本研究では、準同型暗号方式を用いて脈拍センサのデータの統計処理システムを実装した。準同型暗号処理の高速化のために、pyFHE ライブラリを使用してBFV方式とCKKS方式の暗号処理の計算時間を分析し、暗号処理のボトルネックを明らかにした。スロット数(データ数)が1024のとき、CKKS方式の処理時間は、BFV方式に比べて9.9倍であり、BFV方式がCKKS方式よりも高速であった。本システムではLauterが提案したAESとCKKS暗号のハイブリッド方式を採用したが、従来法に比べて計算時間は増大せず、IoT端末で暗号計算の負担を軽減できることを確認した。

参考文献

- 1) S. Erabelli: pyFHE - a Python library for fully homomorphic encryption, Master's Thesis, Massachusetts Institute of Technology, (2020), <https://hdl.handle.net/1721.1/129204>.
- 2) J. Cheon, A. Kim, M. Kim, and Y. Song: Homomorphic Encryption for Arithmetic of Approximate Number, Advances in Cryptology- ASIACRYPT 2017, Lecture Notes in Computer Science, vol=10624, (2017), pp.409-437.
- 3) Z. Brakerski, C. Gentry, and V. Vaikuntanathan: Leveled Fully Homomorphic Encryption without Bootstrapping, in Proc. of the 3rd Innovations in Theoretical Computer Science Conference, (2012), pp. 309-325.
- 4) K. Lauter, M. Naehrig, and V. Vaikuntanathan: Can homomorphic encryption be practical?, Proc. of the 3rd ACM workshop on Cloud computing security workshop, (2011), pp. 113-124.
- 5) M.J. Dworkin: Advanced Encryption Standard (AES), NIST federal information processing standards publication, (2023), <https://doi.org/10.6028/NIST.FIPS.197-upd1>, (Accessed March 9, 2025)

CLIP モデルを利用したテキスト生成とモーショング

大塚愛理*,佐野勇太*,藤野敦大**, 里中孝美(指導教員)

*電子情報技術科, **情報システム技術科

本研究では, 体操の身体動作を記述するデータベースを構築し, CLIP モデルを用いてモーショング画像入力によるテキスト生成(Image to Text)とテキスト入力によるモーショング生成(Text to Motion)の手法について記述する. 13 種類の体操の基本動作を代表する特徴画像の入力に対して, 運動のカテゴリを記述する説明文を生成する方法を提案した. 人間の骨格特徴点の時系列データから特徴画像を生成し, ViTトランスフォーマと CLIP モデルを用いた対照学習で画像のテキスト生成を行った. テキスト生成において 13 種類の体操クラスの正解率は 94.11%から 94.23%となった. テキスト入力によるモーショング生成では, MDM モデルを用いて HumanAct12 の 12 種類の基本動作のシーケンスを学習させる実験を行い, テキスト入力の指定した基本動作のモーショングを生成できることを確認した.

1. はじめに

2020年にTransformerを利用したVision Transformer (ViT)モデルが google から発表され, 自然言語処理と画像認識との融合した生成 AI 技術¹⁾が進展してきた. 従来の画像認識 AI は自動運転, 医療診断, セキュリティなどの分野で応用され, 大量の学習データを用いて性能を向上させてきたが, 画像のラベル付けする作業に, 多大な時間とコストを要する課題があった. その画像のラベル付けの問題を解決した手法が OpenAI から 2021 年に提案された CLIP (Contrastive Language-Image Pre-training)モデルである. CLIP モデルは, インターネット上にある4億枚の多様な画像を用いて学習しており, テキストによる画像の検索, 類似画像検索, 画像とテキストを組み合わせた分類などの応用が期待される.

本研究では, CLIP モデルを用いて体操の基本動作のモーショング画像に対してモーショングの説明文を生成する手法 (Image to Text)を扱う. また, MDM モデルを用いてモーショングのテキスト入力に対してモーショングを生成する手法 (Text to Motion)に取り組んだ.

第1に, Image to Text では OpenPose²⁾を利用して体操の動画から骨格の特徴点の位置, 回転角の時系列データを取得し, モーショングの特徴画像を生成する. 体操の特徴画像を ViT に入力し, 体操クラスの説明文を生成する. 事前学習した CLIP モデルを用いてモーショング画像と説明文との関係性を追加学習させる実験を行い, その識別性能を評価する.

第2に, Text to Motion では HumanML3D の 12 種

類の学習データセット HumanAct12 を用いて基本動作のテキスト入力と基本動作のモーショングとの対応を学習させる. MDM (Motion Diffusion Model) モデルに HumanAct12 の身体動作のテキストを入力し骨格の特徴点の位置, 回転角のシーケンスを生成させる.

2. 言語処理と Transformer モデル

2.1 CLIP モデル

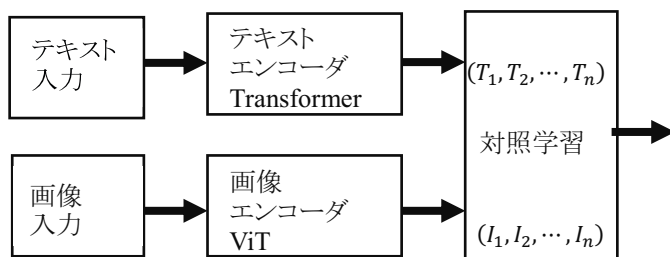


図1 CLIP モデルの構成

CLIP モデルは, テキスト, 画像, 音声などのデータを自然言語処理の埋め込み表現により数値ベクトルとして表現し, それらを統合的に扱う. 図1は, CLIP モデルの構成を示している. テキストエンコーダは, Transformerを用いて, テキストの特徴ベクトルを出力する. 画像エンコーダは ViT を使用し, 画像の特徴ベクトルを出力する. 画像特徴ベクトル (I_1, I_2, \dots, I_n) とテキスト特徴ベクトル (T_1, T_2, \dots, T_n) との関連付けを行うことで, 画像入力によるテキスト生成 (Image to Text) が可能となる. (ただし, I_j , T_j はj番目クラスの画像, テキストの特徴ベクトルである.)

2.2 対照学習

図 2 は、対照学習の説明図である。画像ベクトル (I_1, I_2, \dots, I_n) とテキストベクトル (T_1, T_2, \dots, T_n) の間のペアワイズコサイン類似度を式(1)で定義する。

$$\cos \alpha = \frac{(T_i I_j)}{|T_i| |I_j|} \quad (1)$$

対角線上にある正しいペアに関してはコサイン類似度が大きくなるように、間違っただペアに関してはコサイン類似度が小さくなるように学習を行う。これにより、 $N \times N$ マトリックスにおいて、意味が近い画像とテキスト同士が近いベクトルになるように構成する。

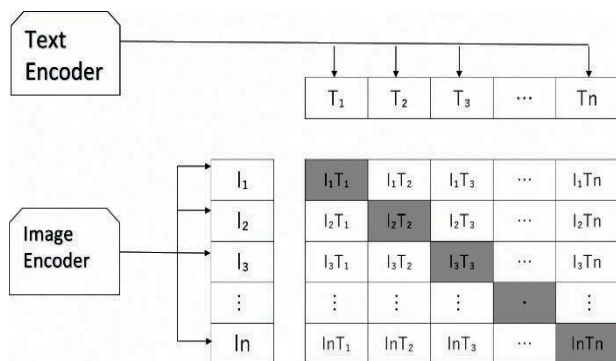


図 2 対照学習の説明図

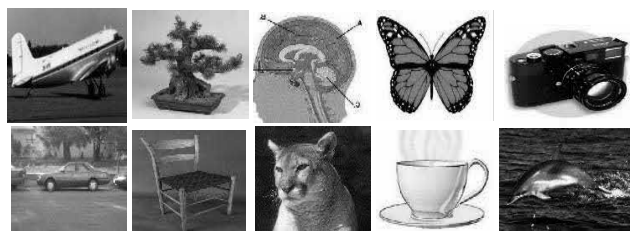


図 3 Caltech-101 データセットの例

表 1 画像とテキストの平均コサイン類似度

	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
I1	0.24	0.19	0.17	0.20	0.19	0.20	0.18	0.20	0.18	0.19
I2	0.19	0.28	0.21	0.21	0.19	0.20	0.19	0.19	0.21	0.19
I3	0.19	0.21	0.28	0.20	0.19	0.21	0.20	0.21	0.20	0.19
I4	0.22	0.20	0.21	0.28	0.21	0.20	0.21	0.20	0.21	0.19
I5	0.20	0.20	0.19	0.20	0.28	0.21	0.19	0.22	0.22	0.21
I6	0.21	0.22	0.21	0.21	0.23	0.27	0.21	0.22	0.21	0.22
I7	0.20	0.21	0.19	0.20	0.18	0.18	0.28	0.19	0.20	0.18
I8	0.18	0.22	0.20	0.21	0.20	0.21	0.20	0.31	0.22	0.21
I9	0.20	0.20	0.21	0.21	0.20	0.20	0.21	0.22	0.26	0.19
I10	0.23	0.22	0.21	0.22	0.20	0.22	0.22	0.23	0.22	0.29

Caltech-101 データセットから図 3 に示す 10 クラスの画像を取得してゼロショット学習の識別試験を行い、コサイン類似度を評価した。表 1 は、画像とテキストの 10 回の平均コサイン類似度を示している。横軸は i 番目の

画像 I_i の特徴ベクトルで、縦軸は、図のテキストのラベル T_i は airplanes, bonsai, brain, butterfly, camera, car, chair, cougar, cup, dolphin である。対角線上のコサイン類似度が最大の出力となり、事前学習ですべての識別を正しく識別した。他の事前学習のデータセットのカテゴリにない画像については正しく識別できなかったため、新規のデータセットでは追加学習を行う必要があった。

2.3 画像エンコーダ

図 4 は、モーション画像を用いた ViT のアーキテクチャを示している。ViT の入力画像を小さなパッチに分割し、それを連続したデータとして扱うことで各画像パッチに対して埋込みベクトルを生成する。入力層の機能は (1)パッチ分割と線形化, (2)埋め込み(ベクトル化), (3)位置情報, クラスの識別情報の追加である。パッチ分割では、入力画像を固定のサイズにパッチ分割し、2次元の画像から 1次元データに変換する。埋め込みでは、線形射影により低次元のベクトルに変換し、埋込みベクトルを生成する。次に、パターン識別に必要な入力画像のクラスのトークンと元の画像の各パッチ位置情報を埋め込みベクトルに追加する。

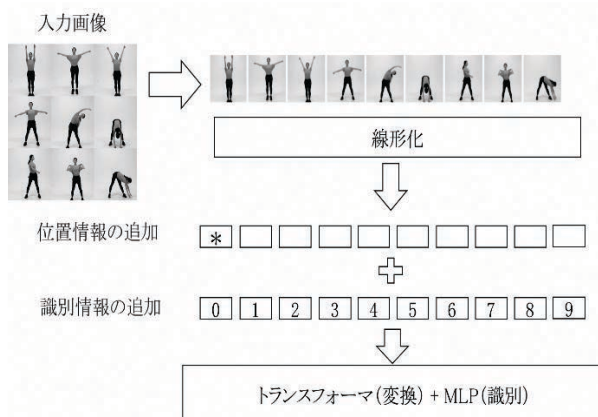


図 4 Vision Transform の説明図

3. モーション画像のテキスト生成

図 5 は、姿勢情報を用いた画像のテキスト生成のフローを示している。テレビ体操の動画を入力し、OpenPose により、18 個の特徴点の時系列データを取得する。テレビ体操は 13 種類の運動から構成され、総時間は約 3 分 15 秒で、時間間隔約 15 秒で動画を分割する。以下の (T1) ~ (T13) のテキストを生成する CLIP の追加学習ネットワークを構成した。(T1)伸びの運動, (T2)腕を振って足を曲げ伸ばす運動, (T3)腕を回す運動, (T4)胸を反らす運動, (T5)体を横に曲げる運動, (T6)体を前後に曲げる運動, (T7)体をねじる運

動, (T8)腕を上下に伸ばす運動, (T9)体を斜め下に曲げ, 胸を反らす運動, (T10)体を回す運動, (T11)両足でとぶ運動, (T12)腕を振って足を曲げ伸ばす運動, (T13)深呼吸である. 図 6 は体操の画像の例である.

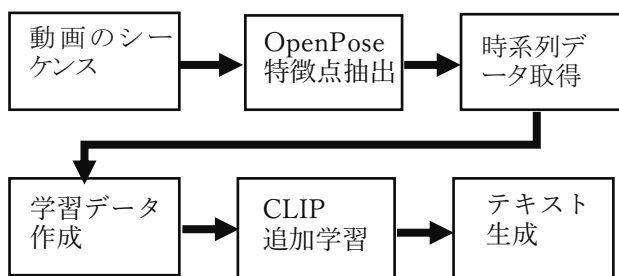


図 5 姿勢情報を用いた画像のテキスト生成のフロー

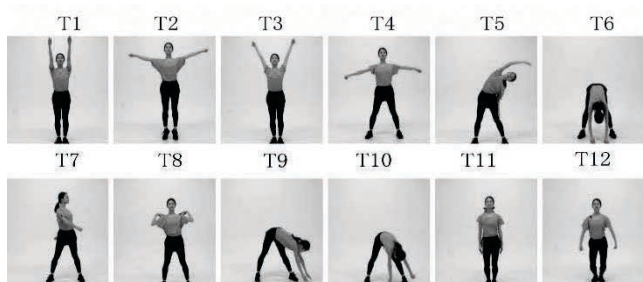


図 6 体操の画像の例

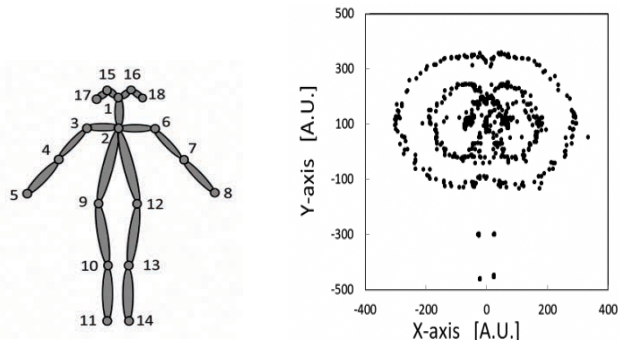


図 7 関節の特徴点

図 8 関節特徴点の散布図

学習用データセットは 9 人の 1000 枚の特徴画像から構成される. テレビ体操の動画から特徴点抽出を行い, 120 フレーム長の時系列データから 224×224 ピクセルの特徴画像の学習用データセットを生成した.

図 7 は, OpenPose を用いて抽出した関節の特徴点の説明図である. 各特徴点は(1)鼻, (2)首, (3)右肩, (4)右肘, (5)右手首, (6)左肩, (7)左肘, (8)左手首, (9)右尻, (10)右膝, (11)右足首, (12)左尻, (13)左膝, (14)左足首に対応する. 図 8 は, T2 の運動の関節特徴点の散布図で, 体操の動画から抽出した特徴点座標 (x_j, y_j) ($j=1 \sim 18$) の時系列データを 2 次元プロットした. CLIP の追加学習では, 2 種類の特徴画像のデー

タセット M1 と M2 を用いた. 図 9 はデータセット M1 の体操 (T1~T12) の関節特徴点の散布図の例である. 図 10 は, データセット M2 のスケルトン画像で, 16 フレー

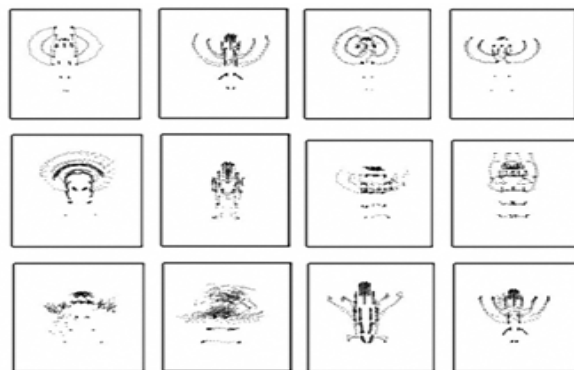
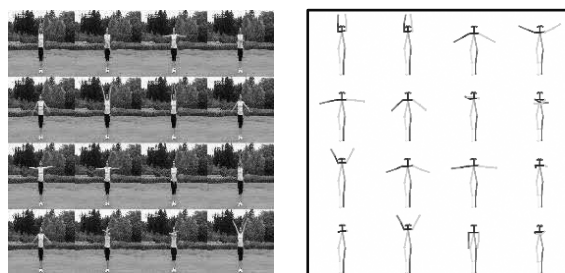


図 9 体操 (T1~T12) の関節特徴点の散布図 (M1)

ムの骨格の動きの画像を用いた. 図 10(a) は 15 秒の運動の動画をサンプリングして得た原画像で, 図 10(b) はそのスケルトン画像である. 図 4 の説明図のように, スケルトン画像をパッチ分割して, ViT で学習を行う. 図 9 と図 10(b) のデータを用いてテキスト出力の正解率を調べた. 学習用データセット M1 の場合は, 1400 回で 94.11% となり, M2 の場合は 2400 回で正解率 94.23% となった.



(a) 原画像

(b) スケルトン画像

図 10 体操 (T1~T12) のスケルトン画像 (M2)

4. モーション生成モデル

図 11 は CLIP を利用したモーション生成モデルである. 本研究では 24 個の骨格の回転角度及び位置のシーケンスで表現したモーションのデータセット HumanML3D を用いて CLIP (ViT-B/32) モデルのテキストで指定したモーションを生成する. モーションの符号化で, 骨格の回転角度シーケンスをモーショントークン x_1, x_2, \dots, x_n に変換する. テキストからモーションへの変換では, 全フレームのモーションの関節の位置 x_1, x_2, \dots, x_n , ステップ数 T , 入力テキストを CLIP によって変換したベクトル c を入力する. MDM モデルは拡散モデルで, ノイズの除去と付与を反復しながら入力

テキストで与えられたモーションを生成する. MDM においてモーション生成のモデルを学習させるために HumanML3D のサブセット HumanAct12 を用いた. それはラベル化された 9 から 403 フレーム長の 12 個の動作(準備運動する, 歩く, 走る, ジャンプする, 飲む, ダンベルの持ち上げる, 座る, 食べる, ハンドルを回す, 電話する, ボクシングする, 投げる)からなる.

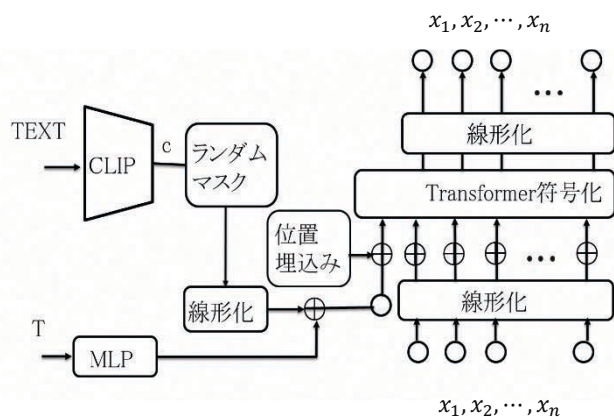


図 11 CLIP を利用したモーション生成モデル

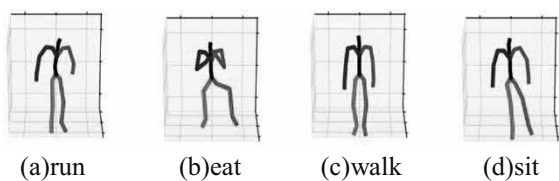


図 12 HumanAct12 のポーズの例

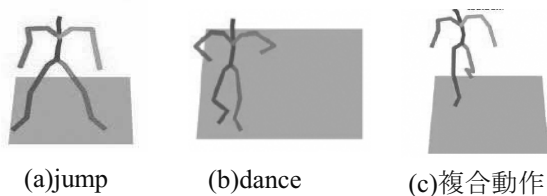


図 13 MDM モデルのモーション生成の結果

図 12 は HumanML3D のサブセット HumanAct12 の run, eat, walk, sit のポーズの例である. 図 13 は, MDM モデルのモーション生成の結果である. (a)と(b)は, テキスト入力により HumanML3D の jump と dance の動作を指定して同一のカテゴリの基本動作を生成することはできたが, 乱数の初期値に依存して異なる jump と dance の結果が得られた. (c)は, 複合動作で「人は歩いて途中でツールボックスをピックアップする」を入力した場合である. テキストエンコーダの学習データのサンプル数が少なく, 3 つのモーション「歩く(walk)」, 「しゃがむ(sit down)」, 「拾う(pick up)」の複数の動作を組み込んだ合成が難しいことがわかった.

5. まとめ

本研究では, CLIP モデルを用いて体操の基本動作のシーケンスの画像に対して運動のカテゴリを記述する説明文を生成する手法 (Image to Text) を提案した. また, テキスト入力によるモーション生成 (Text to Motion) に取り組んだ.

Image to Text 手法 では, OpenPose を用いてテレビ体操の動画から特徴点抽出を行い, 特徴点の時系列データから骨格の特徴点の散布図, スケルトン画像を用いて, 動画のキャプションを自動生成した. CLIP モデルの追加学習により, テレビ体操の 13 種類の予測を行い, 特徴点の散布図, スケルトン画像の正解率はそれぞれ 94.11%, 94.23%となったが, 追加学習の時間が増大した.

Text to Motion 手法では, HumanML3D のサブセット HumanAct12 の時系列データを用いて MDM モデルによりテキスト入力によるモーションの生成に取り組んだ. テキストから基本動作のモーションを生成することはできたが, 複数の基本動作を組み合わせたテキストを入力すると, テキスト通りにモーションが生成されることが多く発生した.

今後の課題は, 自作のデータセットを学習させ, 複数の動作を組み合わせたモーションを生成することである.

参考文献

- 1) A. Radford, J. W. Kim, C. Hallacy, A. Ramesh et al.: Learning Transferable Visual Models From Natural Language Supervision, ICML(2021), pp. 8748–8763.
- 2) Z. Cao, T. Simon, S. Wei and Y. Sheikh: Realtime Multi-Person 2D Pose Estimation using Part Affinity Field, CVPR, (2017) No.121, pp.1302-1310.

電気探査用電極切替器の製作

内田 紗会, 大西 涼太, 松田 賢晟, 池上 知顯(指導教員)
電子情報技術科

地下水の流れを二次元的に測定する方法として、古くから電気探査の一つである比抵抗法が用いられている。地盤の比抵抗を広い範囲にわたり精度よく測定するためには、数十本の棒電極を地表に打設し、適切な電極配置となるように、電流を流す電極と電位を測定する電極を切替えながら測定する必要がある。一般に電極の選択・切替えは各電極と切替器間を単線ケーブルで接続する方法が用いられている。この方法では長さの異なる多数の導線が必要で、これらの導線は大きなループを形成するため、電位信号は外部からのノイズの影響を受けやすくなる。本研究は各電極のそばに小型の切替器を設置し、これらをデージーチェーン接続することで省配線化と耐ノイズ性を有する電気探査用電極切替器を製作した。

1. はじめに

電気探査比抵抗法^{1),2)}は地面に打設した複数の電極に電流を流すことにより大地の比抵抗(抵抗率)を測定し、これから地下の構造や地下水脈などを2次元的に推測する物理探査法である。この測定法を用いることにより地下水資源だけでなく、ため池や河川の堤防からの漏水を検出し、堤防などの決壊を未然に防ぐことが可能になる。

高密度比抵抗法による地下水調査では、測線上に多数の電極を数メートルの等間隔に配置し、各電極から1台の電極切替器まで多数の単線電線で接続するため、多数の長い電線が必要となり、配線作業に手間や時間がかかる。また、地下深部の比抵抗測定では電位電極間に生じる電圧は数 mV 以下と小さくなるため、外部からのノイズによって測定精度の低下が懸念される。

本研究では電極切替えを一箇所集中して行うのではなく、各電極のそばに切替器を取り付ける。これらを一定長の LAN ケーブルで数珠つなぎし、信号線としてツイストペア電線を用いることで、配線作業の効率化や外乱ノイズによる測定信号の劣化低減が期待できる。

2. 比抵抗法の原理

電気探査による地下水脈調査では、一般に4極法による比抵抗法が用いられる。地表に設置した一対の電流電極(C1, C2)から電流(I)を流し、別の一対の電位電極(P1, P2)間の電位差(V)を測定し、これらの電極の

位置や間隔における見掛け比抵抗 ρ_a ($\rho_a = G \frac{V}{I}$, G: 電極位置係数)を求め、測定値の解析から地盤の比抵抗構造を推定する方法である。図1に水平方向の分解能が高い Dipole-Dipole 電極配置の比抵抗法を示す。この方法では多数の棒電極を直線上に等間隔で打設する。電極間隔 C1-C2 と電極間隔 P1-P2 を同じ長さaとする。電極間隔 C2-P2 をna ($n = 1, 2, 3 \dots$)として、電極切替器によりaおよびnを変えながら、電極位置係数Gを $G = n(n+1)(n+2)\pi a$ として見掛け比抵抗 ρ_a を求める。

3. 電極切替システム

3.1 切替器

図2に示すように各電極にはそれぞれ切替器が接続され、これらは電流電極(C1, C2), 電位電極(P1, P2)に接続する信号線(C1_, C2_, P1_, P2_)で数珠つなぎされている。切替器内部のスイッチ(S_{C1}, S_{C2}, S_{P1}, S_{P2})に

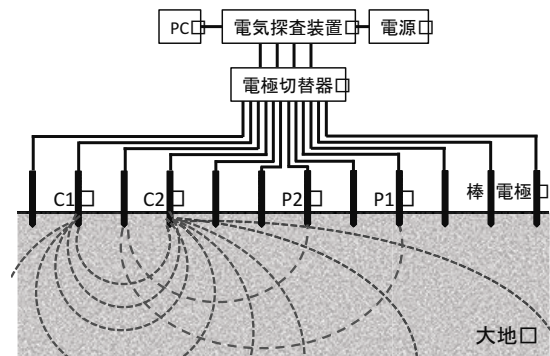


図1 比抵抗法(Dipole-Dipole 電極配置)

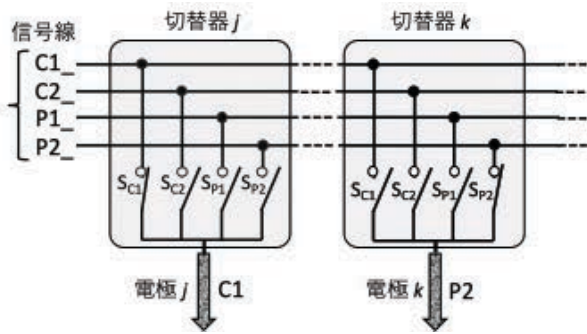


図2 信号線, 切替器, 電極間の接続

よって選択された信号線が電極に接続される。この図では電極 *j* を C1 電極, 電極 *k* を P2 電極に設定した状態である。図3に本研究で作製した電極切替器の回路図を示す。切替スイッチ素子として2種類の PhotoMOS リレーを用い, 電流電極には CPC1978(800V, 0.7A), 電位電極には AQW214 (400V, 1A)を用いた。PhotoMOS リレーの制御には安価な 8bit マイコン (PIC16F18346)を用い, マスターコントローラから RS485 インターフェースを通して 115.2kbps で送られてくる信号を割込みによって受信し, 電極制御コマンドを解析し, PhotoMOS リレーを On-Off する。PIC のプログラム作成には MPLAB X IDE v6.20(XC8, MCC)を用い, PIC へ

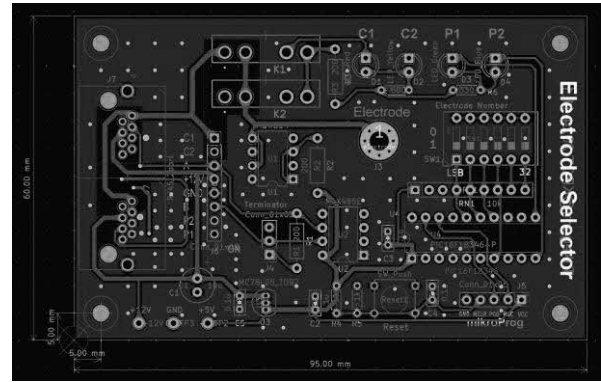


図4 電極切替器 PCBレイアウト(KiCAD)

の書き込みには mikroProg for PIC (MIKROE)を用いた。各切替器には4本の信号線(C1_, C2_, P1_, P2_)の他に RS485 信号線(A, B), 電源(+12V, GND)の計8本の導線が必要となる。今回, 安価でケーブルの取回しや着脱が容易な LAN ケーブル・コネクタと2連のモジュージャックを用いた。切替器間を接続する LAN ケーブルは単線のツイストペアで, 信号線同士の電磁波干渉を防ぎ, 電気特性を安定させるための十字介在が入った 100m の LAN ケーブルを用いた。これを長さ約 3m の 33 本のケーブルに切断し, 各ケーブルの両端に RJ45 プラグを T568B 結線で作成した。

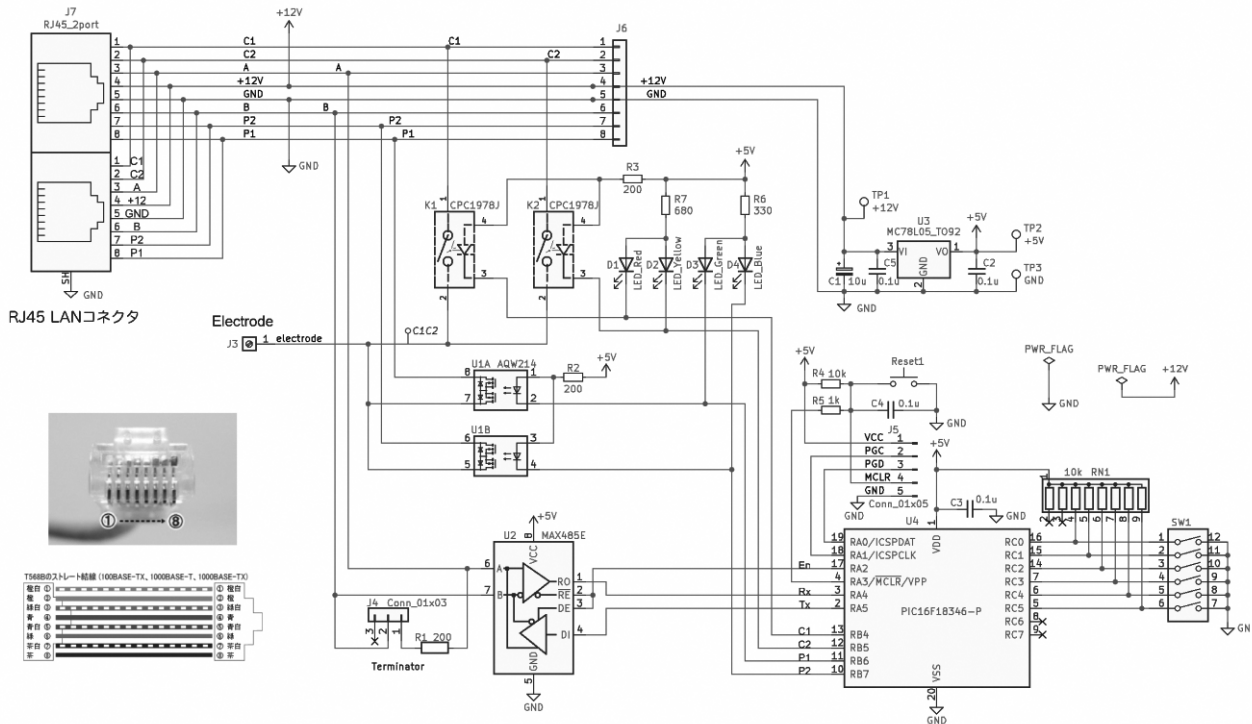


図3 電極切替器回路

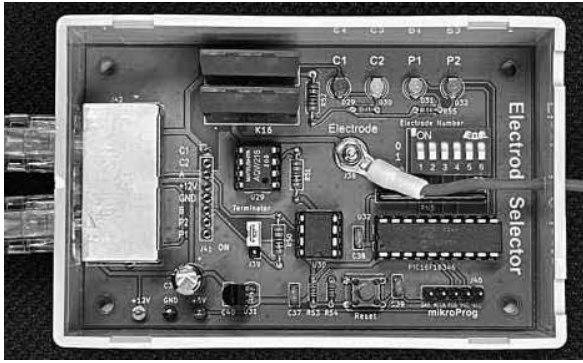


図 5 電極切替器

比抵抗法では一般に電極間間隔を数 m とし、数十 m から数百 m の範囲で測定され、効率よく測定するためには 30 本以上の電極が必要で、同数の切替器が必要となる。ユニバーサル基板を用いて手配線で多数の切替器を製作することは、時間や信頼性の点から合理的ではない。そこでオープンソース EDA ツール (KiCAD 8.0) を用いて PCB 基板を設計し (図 3, 図 4), ブレッドボードおよびユニバーサル基板を用いて切替器を試作し動作を確認した後、PCB 製造メカ (Seed Studio) に発注した。図 5 に示すように部品を実装した PCB 基板を予備も含めて 35 個製作し、それぞれを ABS 樹脂ケースに収納した。切替器が接続される電極番号 (0-63) は基板上の 6 極 DIP スイッチで設定し、その値を PIC 起動時に読取るようにした。PCB 中央付近の端子に長さ約 40cm のシリコン電線 (18AWG) をビス

表 1 ESP32-S3-DevKitC-1-N8 の仕様

CPU	32bit LX7 dual core
クロック周波数	240MHz
動作電圧	3.3V
ROM	8MB
SRAM	520kB
対応無線機規格	Wi-Fi IEEE 802.11 b/g/n Bluetooth (v5.0・BLE)
USB-UART IF	CP2102N
電源電圧	3.3V/ 5V
サイズ	62.74×25.4mm

止めし、その他端に棒電極に接続するためにワニ口クリップを付けた。

3.2 電極切替マスターコントローラ

図 6 に示すマスターコントローラを用いて LAN ケーブルでダイジーチェーン接続された 32 個の電極切替器を制御した。制御に用いたマイコン ESP32-S3-DevKitC の仕様を表 1 に示す。Wi-Fi 通信や Web サーバ・クライアントプログラムは VSCode と PlatformIO を用いて作成した。マスターコントローラから RS485 インターフェースを通して、表 2 に示す制御コマンドを全ての切替器 (スレーブ) へ送信し、信号線と棒電極の接続を制御する。

ESP32 を Wi-Fi アクセスポイント (SSID: Electrode),

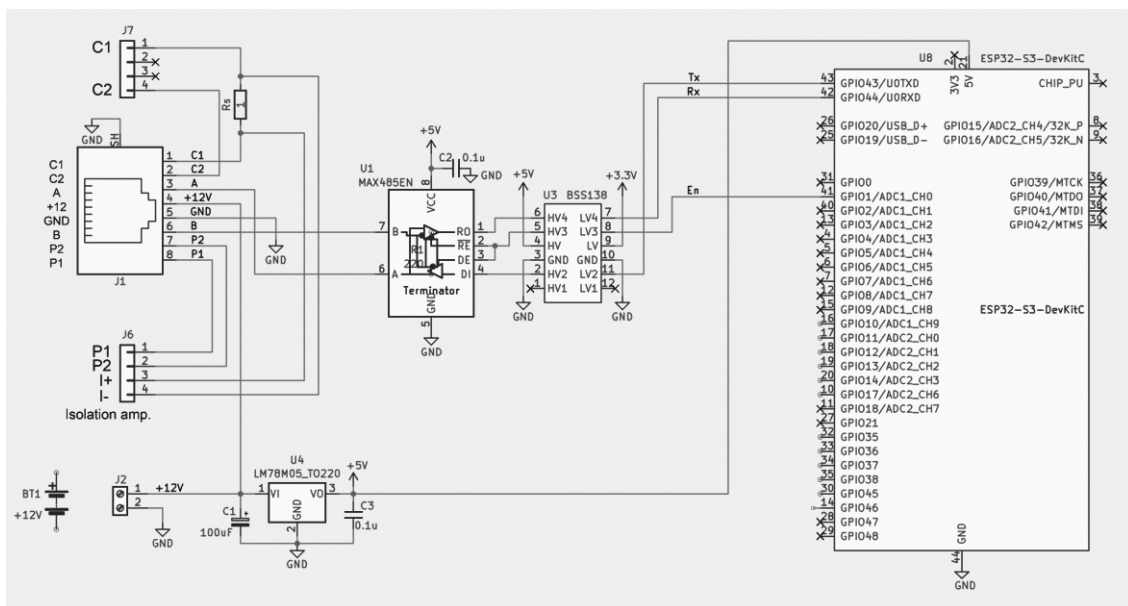


図 6 電極切替マスターコントローラ回路

表 2 電極制御コマンド

機能	コマンド	使用例
C1,C2,P1,P2 に割当て電極 $jklm$ を設定(Set)	$Sjklm$	S00041410
電極 j を信号線 $\#$ に割当(Make)	$EjM\#$	E04MC1
電極 j を信号線 $\#$ から解除(Break)	$EjB\#$	E12BP1
全電極を信号線から開放(Break)	BA	BA
信号線 $\#$ に接続中の電極の問合せ	$? \#$?P2
電極 j に接続中の信号線の問合せ	$?j$?23

j, k, l, m : 電極番号 [00, 01, ..., 30, 31]
 $\#$: 信号線 [C1, C2, P1, P2]

および Web サーバ(IP:192.168.5.5)として立上げ、PC やモバイルデバイスの Web ブラウザから接続する。この時 ESP32 の内部フラッシュメモリに保存された HTML, CSS JavaScript ファイルが Web クライアントへ送信され、ブラウザに図 7 に示す制御パネルが表示される。制御パネル上のボタンなどのクリックによりイベントが発生すると、WebSocket によって JSON 形式のデータがクライアントからマスターコントローラへ送信され、電極切替えが行われる。

3.3 動作確認試験

図 8 に示すように作製した電極切替器 32 個を LAN ケーブルでデジチェーン接続し、マスターコントローラにより、電極切替器の動作試験を行った。切替器 PCB 上の PhotoMOS リレーの動作モニタ用 LED の点灯状態から、作製した切替器が正常に動作することを確認した。

4. おわりに

本研究では電気探査比抵抗測定用に 32 電極測定に対応できる電極切替器と制御用のマスターコントローラを製作し、その動作を確認した。今後、比抵抗測定器として必要な方形波パルス電源や AD コンバータによる電流・電圧測定回路の作製、および測定データ可視化プログラムの作成を行い、実際の地下水の流動調査を行う予定である。



図 7 Web ブラウザによる電極制御画面

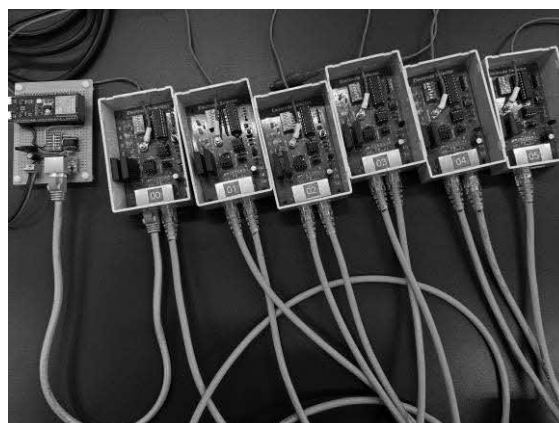


図 8 マスターコントローラ(左端)と電極切替器

切替器収納ケースの LAN モジュラージャック用の穴あけにおいて、精密機械技術科の中野教授から CNC ミリングによる加工をご指導いただいた。ここに謝意を表します。

参考文献

- 1) 電気探査(比抵抗法): <https://www.kinki-geo.co.jp/wp-content/uploads/2019/07/denkitansa.pdf/> (参照日 2025 年 2 月 25 日).
- 2) 物理探査: http://www.chikatansa.co.jp/tansa-tansa_02.html. (参照日 2025 年 2 月 25 日).

RTK-GNSS と IMU による自律走行ロボットカーの作製

永石 陸翔*, 松岡 拓実*, 栗屋 幸奈**, 池上 知顯*(指導教員)

*電子情報技術科, **情報システム技術科

衛星測位システム GNSS (Global Navigation Satellite System)において, RTK (Real Time Kinematic)-GNSS 測位法により高い精度で測位可能であることから, ドローンやロボットカーの自律運転に利用されている. 本研究では昨年度に技大実習棟 A 屋上に設置した RTK-GNSS 基準局からの位置補正情報を使い, RTK-GNSS モジュールとアンテナをロボットカーに搭載し, 本校構内において RTK-GNSS 法により自律運転できるかを検証した. さらに, IMU (Inertial Measurement Unit)による位置推定について検討した.

1. はじめに

GPS に代表される GNSS(衛星測位システム)は, 自動運転やドローンなど, 様々な分野で使用されている. 中でも RTK-GNSS は図1に示すように移動局と基準局の2箇所で GNSS 衛星からの信号を受信し, 基準局からの位置補正情報を用いることで, 実時間で数 cm の精度の位置情報を得ることが可能である.^{1),2)}

IMU(慣性計測装置)は, 物体の動きを感知するセンサ群の総称であり, 加速度センサとジャイロセンサが主な構成要素となる. 加速度センサは物体の加速度を測定し, ジャイロセンサは物体の角速度を測定する.

GNSSとIMUの両方またはいずれかを用いて, 自律走行ロボットカーを精確に安定して動作させることを目的として研究を行った.

2. 使用機器の概要

2.1 基準局(Base:RTK-GNSS)

本研究では昨年度, 技大実習棟 A の屋上に設置した

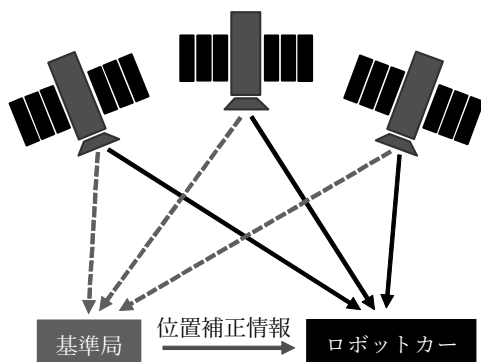
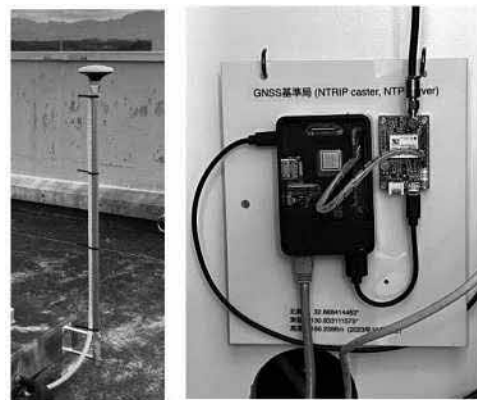


図1 RTK-GNSS 測位原理

RTK-GNSS 基準局(図2)を使用した. 基準局の位置情報等を表1に示す. 基準局のマイコン(Raspberry Pi 3B+)で得られた補正情報や観測情報をインターネットで送信するために Ntrip (Network Transport of RTCM via Internet Protocol) Server を立て, これらの情報の配信サーバである学外の Ntrip Caster(geortk.jp)に送信した. また, 自分自身, 学内配信用の Ntrip Caster として動作させている.



(a)アンテナ (b)受信モジュールと Raspberry Pi (Ntrip Server / Caster, NTP Server)

図2 RTK-GNSS 基準局

表1 技大 RTK-GNSS 基準局情報

緯度	32. 88841448°
経度	130. 83311158°
高度(標高)	186. 239m (153. 499m)
NTRIP Caster /ポート番号	geortk.jp /2101 (学外) 192.168.100.210 /2101 (学内)
マウントポイント	KPCT (ID, パスワード無し)

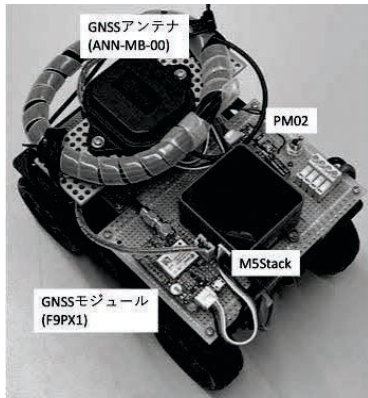


図3 ロボットカー(GNSS 移動局)

2.2 ロボットカー(移動局:Rover)

図3に作製したロボットカーの外観を、図4にその回路構成を示す。車体の各車輪にギヤボックスを介して取り付けられた4個のDCモータは左側と右側それぞれ前後2個を並列接続して駆動するため、クローラと似た動作となる。ロボットカーの制御にはドローンやロボットの自律飛行・移動を制御するフライトコントローラ(Pixhawk 6C Mini)に自動操縦ファームウェア ArduPilot をインストールして用い、GNSS モジュール(F9PX1)とアンテナを移動局として用いた。ロボットカーの電源として、LiPo 電池(7.4V, 4000mAh)を用い、Pixhawk からのモータへ駆動用 PWM 信号を ESC(Electronic Speed Controller)に入力して、モータの回転速度や方向を制

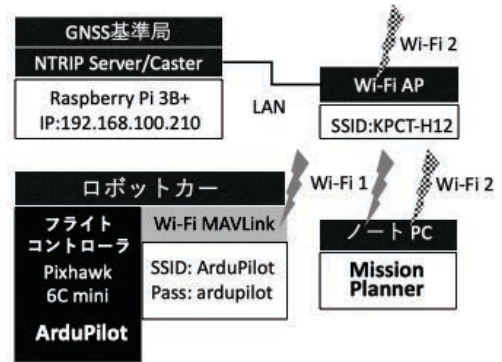


図5 機器間の通信接続

御した。また、IMU のみの自律走行実験のため、マイコンモジュール(M5StackCoreS3)を搭載した。

図5のようにノート PC に ArduPilot の地上管制プログラム Mission Planner をインストールし、ロボットカーに搭載したマイコン(ESP-WROOM02)による MAVLink Wi-Fi Bridge を用いて ArduPilot と無線通信した。また、ノート PC は GNSS 基準局と学内 Wi-Fi AP (SSID: KPCT-H12)経由で接続し、基準局の Ntrip Caster からの位置補正情報を受信し、Pixhawk にシリアル接続された GNSS モジュールへ送る。ArduPilot は GNSS モジュール内で RTK 演算により求められた位置情報と Pixhawk に内蔵された IMU や電子コンパスなどのセンサーデータを用いて自動操縦を行う。

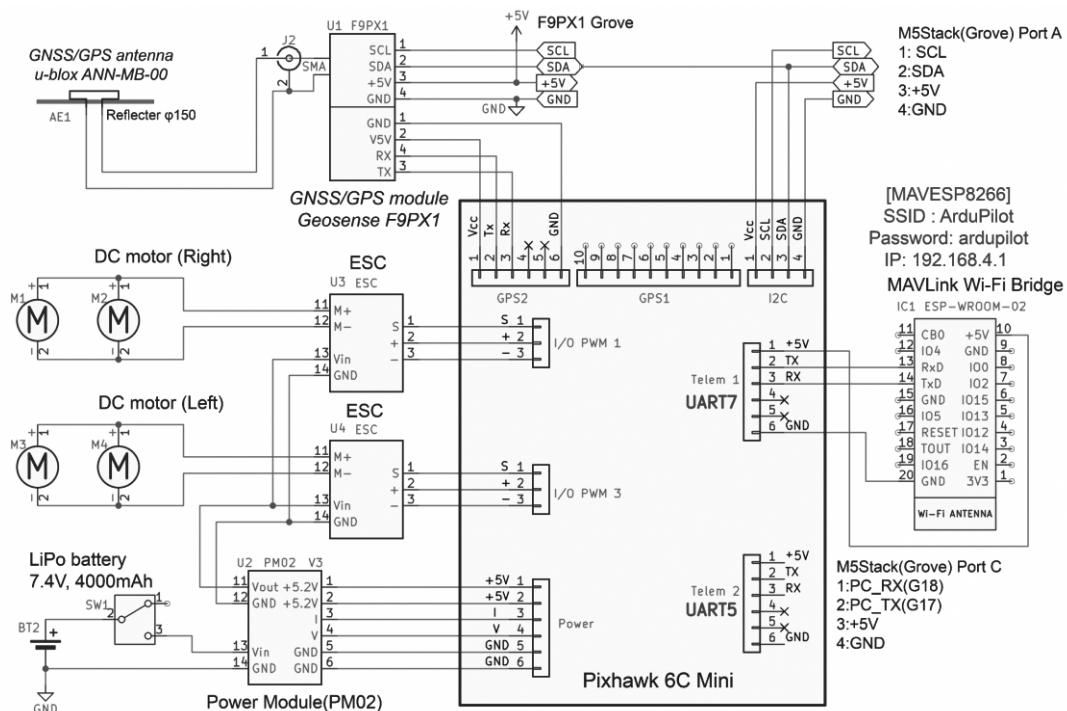


図4 ロボットカーの回路構成



図 6 Mission Planner の制御画面

3. RTK-GNSS による自律走行実験

3. 1 実験手順

Mission Planner から ArduPilot の制御パラメータを設定し、作製したロボットカーが地図上に設定した走行ルートに沿って自律走行できるか検証した。

実験を行うために以下の6つの手順を行う。³⁾

- ① ロボットカーの電源を入れ、Pixhawk の ArduPilot を起動させ、Mission Planner を起動させた PC と Wi-Fi で接続する。
- ② Pixhawk に搭載されている IMU のキャリブレーションを行う。「初期設定」→「Accel Calibration」を表示し、Calibrate Accel を実行すると Pixhawk を横に向けたり、裏返したりという指示が画面に表示されるので、指示に従ってキャリブレーションを完了させる。
- ③ Mission Planner の「フライト・プラン」上で走行ルートを作成する。
- ④ 作成した走行ルートを Pixhawk に送信し、書き込みを行う。
- ⑤ 「フライト・データ」上の「アクション」の項目内で、モードを「AUTO」にセットし、WP(ウェイポイント)を 1 にセットする。
- ⑥ 図 6 の制御画面のように GPS 3D Fix が表示されていることを確認した後、「Arm/Disarm」をクリックし、自律走行を開始させる。

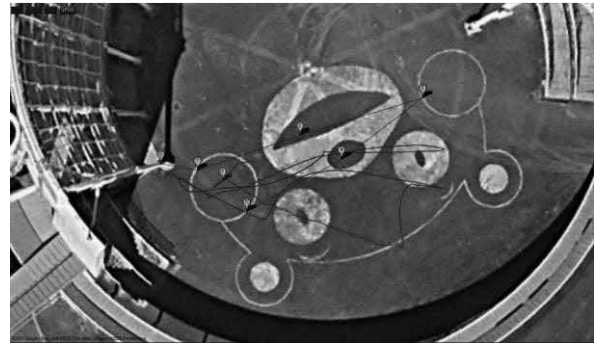


図 7 Mission Planner による走行ルートの設定と自律走行軌跡

3. 2 実験結果

図 7 の細線はは実際に走行したときの軌跡で、設定したルート(丸ピンを結んだ直線)と比較すると、大きく外れた走行となった。原因としては ArduPilot の設定パラメータの不備や、Fail Safe エラーのため Pixhawk 内蔵の電子コンパスからの方位データを用いなかったことが考えられる。

4. IMU を使った位置推定

4. 1 概要

位置情報を取得するには GNSS 信号が必要だが、トンネルや森では使用できない場合もある。そのため、安価な IMU(6 軸の加速度・ジャイロ MEMS センサ)のみで位置推定が行えるのか検証した。IMU にはロボットカーに搭載した M5StackCoreS3 に内蔵された BMI270 を使用し、加速度データを取得した。データ取り込みとグラフ化のプログラムには Arduino 言語を使用した。Arduino 言語は主に電子工作や組み込みシステムのプロジェクトに使用されるプログラミング言語で、C/C++ をベースにしている。プログラムはデバイスなどの初期化を行う関数 `setup()` と、その後繰り返し実行される `loop()` で構成されている。Arduino 言語は標準ライブラリや、電子工作で使われる多くのデバイスの制御用ライブラリが提供されており、プログラミングが容易である。

また、JavaScript 言語と HTML 言語を用いて、M5Stack の Web サーバ機能を使った Web プログラムを作成した。図 8 のように RTK-GNSS による測位結果を表示し、ロボットカーの手動操作や IMU 自律運転において目標座標の設定を行うことができる。スライダを

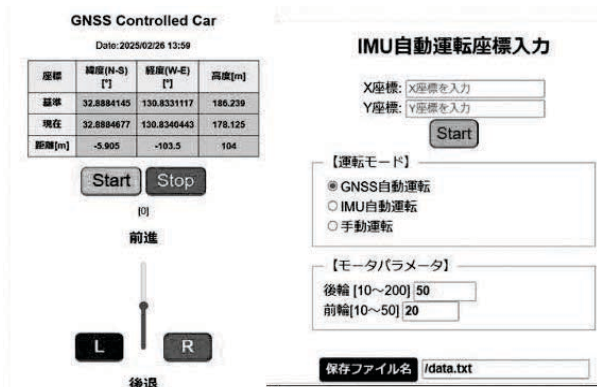


図 8 Web ブラウザでの設定・制御

動かしたり、ボタンを押したりすることで、JSON 形式のデータを M5Stack (Web サーバ)へ送信し、それらのイベントに応じてモータが動作するようにした。

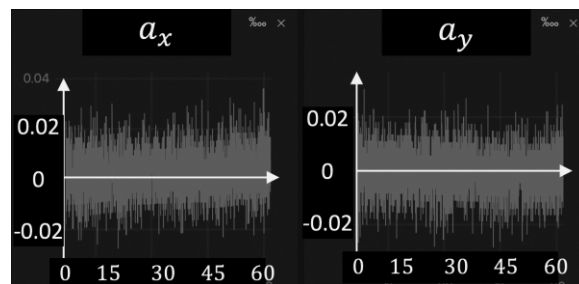
4. 2 IMU の加速度データによる移動量の推定

IMU から得られた加速度を積分して速度を求め、さらに積分することで変位を求めるプログラムを作成した。図 9 に静止状態の IMU からの 100S/s のレートで水平面 (x 軸, y 軸)方向の加速度データ(a_x, a_y)を 60 秒間取り込み、これらから計算した速度(v_x, v_y), 変位(d_x, d_y)の計算結果を示す。予め静止時の 500 個の加速度データの平均値からバイアスを求め、測定データから差し引くことで取り除いたが、バイアスのドリフトやノイズにより、静止状態であるにもかかわらず速度が変化し、正しい移動量は得られなかった。IMU による移動量の推定にはカルマンフィルタやローパスフィルタなどの信号処理を用いる必要があると思われる。

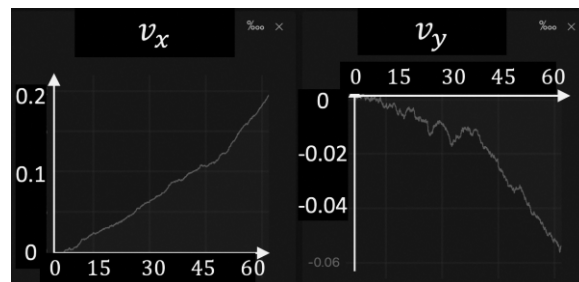
5. おわりに

本研究では地上管制プログラム Mission Planner を用いて、RTK-GNSS 測位のロボットカーの自律走行を行った。その結果、設定ルートから大きく逸脱し、ウェイポイントを大回りするなど数 m の誤差が生じ、設定した通りの自律走行はできなかった。ArduPilot の設定パラメータが適切でなかったことや、電子コンパスからの方位データを用いなかったことが考えられる。さらに、屋外でのロボットカーと Wi-Fi AP 間の通信状況によっては、基準局からの位置補正情報が得られず、安定して RTK-GNSS の Fix 解が得られなかったことなどが考えられる。

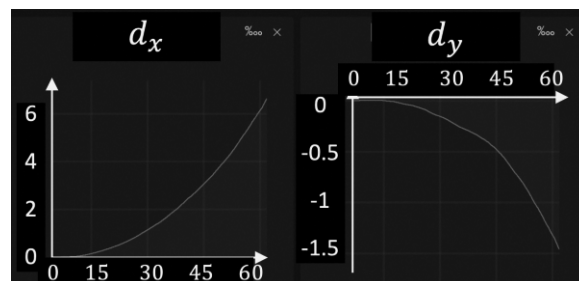
今後は、ArduPilot のパラメータの最適化や外部磁気



(a) IMU から取得した加速度 [m/s^2]



(b) 加速度を積分した速度 [m/s]



(c) 速度を積分した変位 [m]

図 9 IMU による x 軸, y 軸の加速度, 速度, 変位

コンパスの利用, カルマンフィルタを用いた IMU によるロボットカーの自律運転などを試みる予定である。

参考文献

- 1) トランジスタ技術, 2016 年 2 月号, pp.66.
- 2) トラ技 Jr., 2024 年 夏号 第 58 号, pp.19-21.
- 3) ArduPilot, <https://ardupilot.org/planner/docs/mission-planner-overview.html> (参照日 2025 年 1 月 20 日).
- 4) Complete Parameter List - Copter documentation, <https://ardupilot.org/copter/docs/parameters.html> (参照日 2025 年 1 月 20 日).
- 5) ARDUPILOT Versatile, Trusted, Open, <https://ardupilot.org/rover/docs/parameters.html#gps-auto-config> (参照日 2024 年 11 月 19 日).
- 6) はてなブログ, <https://dronejo.hatenablog.com/entry/2018/01/14/190920> (参照日 2025 年 2 月 21 日).

加速度データ無線送信システムの開発

*高崎 柊也, **鬼塚 夏鈴, *福田 真(指導教員)
*電子情報技術科, **情報システム技術科

建築物や橋梁の経年劣化を評価するにはいくつかの手法があるが本研究では加速度データを用いることとした。加速度データを用いることにより位置の変化が分かる。そこで本研究では加速度データを無線送受信するシステムの開発を行った。本研究において加速度データの送信側には Raspberry Pi Zero WH を、受信側には Raspberry Pi 4 の無線 LAN 機能を用い、データ処理には Python を用いた。昨年度の課題である、通信距離の延伸と高速度かつ高精度データ送信に取り組み、通信距離を2倍以上延伸することに成功した。

1. 緒言

建築物や橋梁の経年劣化を評価するにはいくつかの手法があるが本研究では加速度データを用いることとした。加速度データを用いることにより位置の変化が分かる。そこで本研究では加速度データを無線送受信するシステムの開発を行った。

昨年度の研究¹⁾では1秒間に1000個のデータを集計し、Raspberry Pi Pico WH を用いた無線送信システムを開発した。これに対し、共同研究をしている平田様からは建築物や橋梁の経年劣化を評価すること以外にも他の用途でも使えるよう24bitのサンプリングレート1kHz以上のデータを集計したいと要望があった。また昨年度のシステムでは無線通信の距離が70m程度であったため長い橋などでは通信距離が不足する可能性が指摘された。そこで今年度は高速かつ高精度のデータ転送システムの作成と通信距離の延伸を目的として研究を行った。

本研究において加速度データの送信側には Raspberry Pi Zero WH を、受信側には Raspberry Pi 4 の無線 LAN 機能を用い、データ処理には Python を用いた。複数台の加速度センサのデータを集計後に送信し、外部の PC で受信したデータを取り出すシステムの開発を行った。なお本研究は、アールエステクノロジー株式会社と連携した共同研究となっている。

2. 実験装置

2.1 システム外観

本研究では複数台での無線通信を行うために、Wi-Fi ルータと、加速度センサデータを無線送信するための Raspberry Pi Zero WH を用いた。また加速度センサはアナログ値を測定するセンサを使用し、AD コンバー

タを用いてデジタルデータに変換する。AD コンバータを使用するシステムの開発を行うことで高速かつ高精度データの送信が可能になると考えた。

システムの概観を図1に示す。

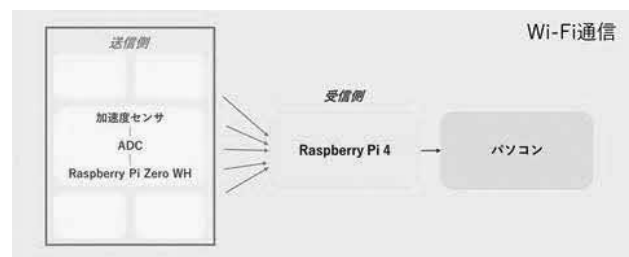


図1 システム外観

昨年度のシステムでは無線通信に使用する Wi-Fi を M5Stack マイコンによって提供していた。しかし M5Stack のスペックでは高速かつ高精度データの送受信が難しいため、今年度は Wi-Fi ルータを用いることとした。

2.2 使用機材

今年度使用した機材を記述する。まずデータ送信用マイコンとして Raspberry Pi Zero WH (以下ラズパイ Zero と呼称) を使用した。ラズパイ Zero のスペックを表1に示す。

表1 ラズパイ Zero

CPU	ARM1176JZ-F
メモリ	512MB
動作温度	0~50°C
消費電流	400mA(MAX)
電源電圧規格	DC5V
Wi-Fi 規格	IEEE 802.11 b/g/n 2.4GHz

昨年度は Raspberry Pi Pico WH を使用したが本年度はラズパイ Zero を使用した。変更理由は Raspberry Pi Pico WH のスペックでは高速通信が難しく、またプログラムの変更、追加が容易ではない点である。ラズパイ Zero は Raspberry Pi Pico WH と比較して高価であるが、2,000 円程度の違いであり、その分スペックが高く、昨年度の課題をクリアするには十分なものであると判断した。比較のため Raspberry Pi Pico WH のスペックを表 2 に示す。

表 2 Raspberry Pi Pico WH

CPU	ARM Cortex M0+
メモリ	256KB
動作温度	-20°C~85°C
消費電流	300mA(MAX)
電源電圧規格	DC5V
Wi-Fi 規格	IEEE 802.11 b/g/n 2.4GHz

ラズパイ Zero との比較ではメモリの違いがあげられる。Raspberry Pi Pico WH は 256KB に対しラズパイ Zero では 512MB と大きく異なっており、高速かつ高精度データ送信ではラズパイ Zero が適切であると考えられる。図 2 にラズパイ Zero の概観を示す。

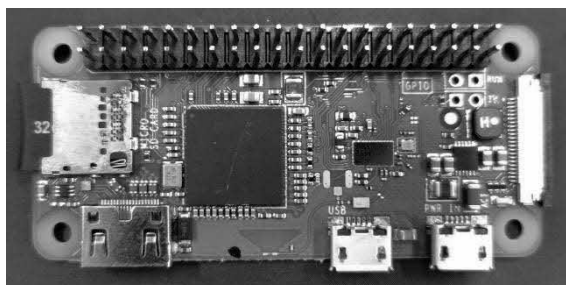


図 2 ラズパイ Zero 概観

サイズは 65mm×30mm となっておりコンパクトなシステムを構成することができる。

次にデータ集計用マイコンの Raspberry Pi 4 のスペックを表 3 に示す。

表 3 Raspberry Pi 4

CPU	quAD-core Cortex-A72(ARM v8)
メモリ	8GB
動作温度	0~50°C
消費電流	400mA(MAX) 1.7A(MAX)
電源電圧規格	DC5V
Wi-Fi 規格	IEEE 802.11 b/g/n 2.4GHz

高速通信されるデータを各センサから受信し集計するためラズパイ Zero と比較してもスペックの高いものが必要となるため、昨年に引き続き今年度の研究でも Raspberry Pi 4 を使用した。

次に AD コンバータの説明を行う。今年度使用した AD コンバータは ADS1256 である。ADS1256 のスペックを表 4 に示す。

表 4 ADS1256

Bit 数	24bit
サンプリングレート	30ksps
入力チャンネル数	8
動作温度	-40~85°C
消費電力 (代表値)	36mV

ADS1256 は高精度 AD コンバータであり、扱える bit 数は 24 ビット、サンプリングレートも 30ksps となっており、今年度に必要なスペックを十分に満たしていると判断した。今年度は ADS1256 チップを搭載し、ラズパイ Zero に取り付け可能な WaveShare 社の拡張ボードを使用した。拡張ボードの概観を図 3 に示す。

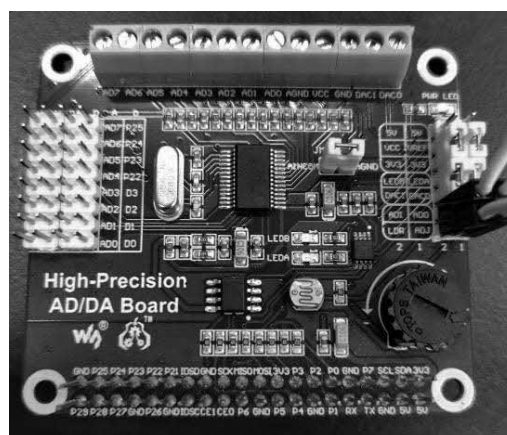


図 3 ADS1256 拡張ボード

ADS1256 拡張ボードのサイズは 65mm×50mm となっており、ラズパイ Zero に取り付け可能であり、ラズパイ Zero からの電力供給で稼働するため今年度の研究で使用した。

今年度の一つの目標として、高速かつ大容量のデータ転送を可能にすることがあるが、もう一つ、橋の面積をカバーできるほどの広い通信環境を確保することがある。前年度のシステムでは、Wi-Fi を提供でき AD コンバータとしての機能がある M5Stack を加速度センサに取り付け、外部 PC でデータを取り出していたが、広

い通信環境が望めなかった。今年度は、橋の面積をカバーできる広い Wi-Fi 環境を確保するために、メッシュ Wi-Fi に対応しているバッファロー社の WSR-5400AX6P-BK を用いて通信環境を作った。WSR-5400AX6P-BK の基本スペックを表 5 に示す。

表 5 WSR-5400AX6P-BK

伝送方式	CCK, DSSS, OFDM, OFDMA, MIMO
周波数範囲	2.4GHz:2412~2472Hz
消費電力	14.6W
動作保証環境	温度:0~40℃ 湿度:10~85%

消費電力はやや大きいが通信距離の延伸に必要なメッシュ Wi-Fi システムを利用できるため選定した。ここでメッシュ (Mesh) Wi-Fi とは、網目 (メッシュ) のようにネットワーク機器がつながり合う通信形態のことである。これを用いると、間隔をあけて複数の Wi-Fi ネットワーク機器を配置するだけで容易に Wi-Fi 範囲を拡大できる。図 4, 図 5 でメッシュ Wi-Fi の説明を行う。

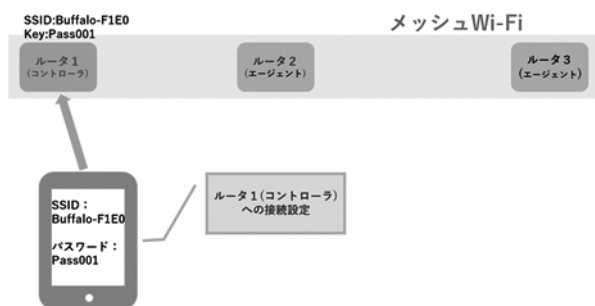


図 4 ルータ 1 の接続

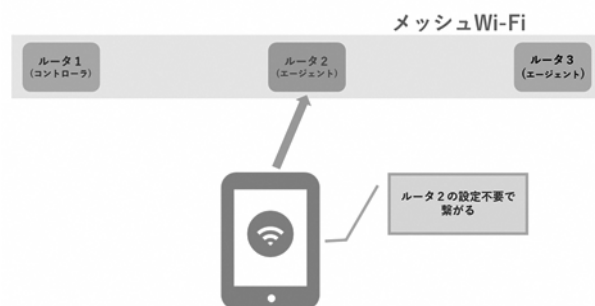


図 5 ルータ 2 の接続

図 4 の状態ではユーザーが使用するデバイスはルータ 1 に接続している。ユーザーが移動しルータ 1 の範囲を超えてルータ 2 の接続可能範囲に入ったものが図 5 になる。通常の Wi-Fi システムではルータが変わった際に改めて新しいルータの SSID とパスワードを用いて接続をやり直す必要がある。しかしメッシュ Wi-Fi システムでは自動で切り替わりが行われ全体として一つの Wi-Fi システムを利用することが可能となる。本研究ではセンサデータを送信するマイコンは同一の Wi-Fi 中にあることでデータ集計が可能なシステムを構築しているためメッシュ Wi-Fi による通信距離の延伸が必要となる。データ送信用マイコンがグローバルネットワークに接続されている場合はそれぞれのマイコンがグローバル IP アドレスを持つためメッシュ Wi-Fi は必要ないが、本研究ではクローズな無線通信環境を構築することを目的としているためローカルな IP アドレスしか持つことができない。

今年度は WSR-5400AX6P-BK を 3 台購入したので、1 台がネットワークを管理する「コントローラ」、ほかの 2 台はコントローラに従う「エージェント」となる。コントローラは、エージェントを通じて、新しい Wi-Fi ネットワーク機器のメッシュ Wi-Fi への参加を調整したり、スループットや信号強度、利用可能な帯域幅などの情報を把握し、通信経路が適切に維持されるよう最適化を行う。図 6 に WSR-5400AX6P-BK の概観と寸法を載せる。



高さ:177mm
幅:60mm
奥行:175mm
重量:650g

図 6 WSR-5400AX6P-BK 概観

WSR-5400AX6P-BK は使用状況にも依存するが 30 台程度の接続まで可能なため本研究において多数のセンサを配置しても十分対応可能なものである。

3. 実験

3 台の WSR-5400AX6P-BK をコントローラ 1 台、エー

ジェント 2 台のとして接続実験を行った。昨年度までのシステムでは加速度センサのデータを送信する Raspberry Pi Pico WH からデータを受信するノートパソコンまでは障害物のない見通しのよい環境で 70m 程度が限界であった。今年度は技大の敷地内において図 7 の配置で接続実験を行った。



図 7 機材配置場所

図 7 中の△点にコントローラを設置し、○点にノートパソコン、◇点にラズパイ Zero を配置した。それぞれの距離は○と△が 97m、△と◇が 95m であった。実験手順はルータ 3 台にコントローラから順に電源を入れ、次に加速度センサに接続されているデータ送信用のラズパイ Zero とデータ受信用ノートパソコンを Wi-Fi に接続する。Wi-Fi 接続が確認されたら加速度センサのデータを取得し、Wi-Fi 送信する Python プログラムをラズパイ Zero 内で実行し、ノートパソコンでは受信データを確認する。今回の実験では送信されたデータをノートパソコンで受信できたことが確認され、昨年度のシステムより無線通信の距離を延伸できたことが確認された。

同時に ADS1256 を接続したデータの送信も可能であることが確認された。

4. まとめ

今年度の卒業研究では昨年度のシステムの課題であった通信の距離の延伸についてメッシュ Wi-Fi システムを利用することで実現することができた。今年度のシステムであれば理論上 30 台程度のセンサを同時集計することができ、さらに Wi-Fi システム全体の通信の距離は 200m 程度まで伸ばすことが可能である。また今年度選定した AD コンバータ ADS1256 を用いれば高速かつ

高精度なデータ送信が可能であることが確認された。

今後の課題としては 2 点あげられる。1 点目はシステムの持ち運びである。今年度の実験では WSR-5400AX6 P-BK の電源は技大のコンセントより電源ケーブルを用いて確保した。しかし橋などで使用する場合電源の供給が問題となる。モバイルバッテリーの使用も考えられるが長時間の稼働が難しくなるためあまり得策ではないと思われる。システム全体の可搬性を担保するための検討が必要である。2 点目はさらなる高速度かつ高精度データ通信に向けての取り組みである。今年度は ADS1256 モジュールを使用したはそのプログラムを Python プログラムで利用したため十分な性能を発揮させることができなかった。インタプリタ言語である Python より高速な処理が可能な C 言語等による AD 変換システムが必要であると考えられる。近年 Raspberry Pi シリーズのプログラミング言語としては Python が主流となっているため C 言語によるシステム開発を独自に行う必要がある。

今年度の研究では生成 AI を利用して開発を行ったが、あまり効果的ではなかった。対話処理によるプロンプトエンジニアリングを行ったがこちらが欲しいプログラムが出力されることはなく、問題が多いプログラムが出力され、修正をしても動かないプログラムが多かった。今後生成 AI の利用は増えていくと思われるが自分のプログラミングスキルとプロンプトエンジニアリングのスキルの両方がないと開発には利用できないと思われる。

参考文献

- 1) 草津 尚輝, 栗原 拓人, 西村 昂樹, 令和 5 年度 (2023 年度) 卒業研究概要集, 加速度データ無線送信システムの開発, pp89-92.

移動体速度検出システムの開発

藤本 脩平, 松本 昂大, 福田 真(指導教員)
電子情報技術科

高速で移動する物体を検知して、その速度を算出し、検知情報と速度情報を集計するシステムの開発を行う。近年、競技や教育用途でマイコンカーなどの高速移動体を使う場面が増えており、その速度を正確に測定することは、高速移動体の性能向上や競技会などでの戦略を建てるために必要と考えられる。そこで本卒業研究ではセンサデータの送受信を通して IoT 分野の技術を学ぶとともに、センサデータの送受信、データ集計システムと集計データの表示システムを完成させた。

1. 緒言

本研究では令和 5 年度卒業研究¹⁾をもとにセンサ、無線通信を用いて、移動体の速度を検出する装置を開発する。昨年度の研究ではセンサを 3 台までしか使用できず、また速度の正確な算出が困難であった。そこで今年度は昨年度のシステムを改良し、より多くのセンサデータを集計し、正確な速度計測システムの開発を目的とした。

2. 実験

2.1 実験装置

今回使用した機器は、データ集計とホームページサーバとして使用する Raspberry Pi 3 Model B+(以下ラズパイと呼称)、システムのセンサ値監視とセンサデータの無線送信を行う ESP32-WROOM-32E(以下 ESP32 と呼称)、物体の通過を検知する透過型光電センサと昇降圧 DC-DC コンバータモジュールである。使用したラズパイのスペックは表1となる。昨年度のシステムではデータ集計とホームページサーバも ESP32 を使用したが、今年度はデータ集計とホームページサーバにはラズパイを使用した。その選定理由は、ストレージ容量やソフトウェアの柔軟性、Wi-Fi やイーサネットポートを標準で搭載しており、インターネットやローカルネットワークへの接続が容易という点である。

表1 ラズパイスペック

メモリー	1GB, DDR2, 450MHz, 定電圧 SDRAM
Bluetooth	4.2 対応
電源	5V 2.5A
最大消費電力	約 12.5W

次に ESP32 マイコンのスペックは表 2 となる。ESP32 はセンサ監視とセンサデータを送信する子機に使用した。ESP32 マイコンを選定した理由は、比較的 low コストで Wi-Fi 利用の環境構築が容易であるためである。また IoT 開発において人気の高いデバイスであるため参考資料も豊富に存在するため開発が容易になると考えた。また、2.4GHz 帯の Wi-Fi を内蔵しており本研究での無線データ送信に非常に有効である。

表2 ESP32 スペック

電圧電源	5V
出力電圧	3.3V
フラッシュメモリ	4MB
クロック周波数	240MHz
価格	1個 1,600 円
動作温度	-40°C~125°C

次に物体の通過を検知するセンサとして、今年度は透過型光電センサ PTB-802-5L(投光器)、PTB-802-5D 1(受光器)を使用した。スペックは表 3 となっている。

表3 透過型光電センサスペック

電源電圧	DC5V
使用電圧範囲	DC4.5V~5.5V
消費電流(投光器)	DC50mA 以下
消費電流(受光器)	DC10mA 以下
出力	DC30V DC100mA 以下
価格(投光器)	1,364 円
価格(受光器)	1,826 円

今回採用した透過型光電センサは数メートルまで物体

の通過を検知可能で、投光器と受光器を移動体が通過する場所を挟むように配置することで物体を検知する。昨年度使用した距離センサでは物体までの距離によって通過検知を行ったため移動体の形状により通過時間の正確な検知が困難であったが、今年度のセンサは透過型光電センサのため正確な検知が容易となった。また動作速度も 100ms 以下で反応するため昨年度より詳細な通過時間を検知することが可能となる。

次に昇降圧 DC-DC コンバータモジュールのスペックは表 4 である。

表 4 DC-DC コンバータモジュールスペック

入力電圧	1.2V~15V
出力電圧	3.3V または 5.0V
無負荷消費電流	40 μ A VIN=1.5V VOOUT=3.3V
最大負荷電流(昇圧時)	50mA
最大負荷電流(降圧時)	100~200mA
価格	990 円

昇降圧 DC-DC コンバータモジュールはセンサの電源に使用した。移動体検出システム全体の可搬性を担保するためセンサの電源には乾電池を使用することとしたが、乾電池の電圧は使用により減衰するため安定的な電圧供給システムが必要となる。そこで昇降圧 DC-DC コンバータを使うことで、安定した電力供給、長寿命化、高効率化、柔軟性、コンパクト化を実現することができた。

2.2 システム外観

システムの構成は図 1 のとおりである。

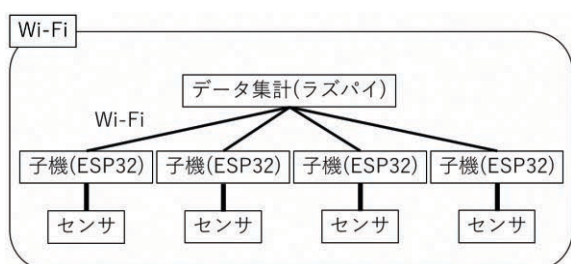


図 1 システム外観

センサデータを取り入れた ESP32 からラズパイに無線でデータの送信を行う。無線通信の方法としてはポータブル Wi-Fi で Wi-Fi 環境を提供し、それぞれのセンサデータがどのセンサのものか判別するために、子機となる ESP32 に固定 IP アドレスを与えて、データを判別す

る。子機から得られた移動体の通過情報をラズパイに集計し、距離と時間の情報から速度を算出する。またそれぞれの子機と投光器には電池モジュールと DC-DC コンバータモジュールを接続し、可搬性のある電力供給システムを構築した。図 2 にシステムの一部である投光器に電池モジュールと DC-DC コンバータモジュールを接続したものを示す。

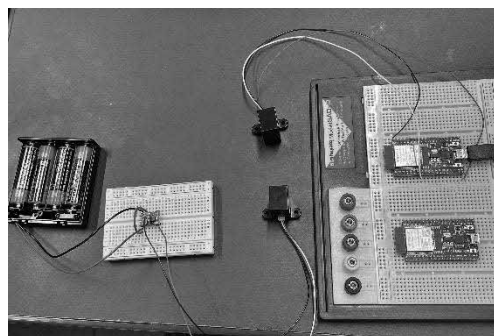


図 2 投光器とその周辺

2.3 物体通過情報の検出

移動体が通過したという情報を検出するために、通過する移動体を挟む形でセンサを取り付ける。投光器から受光器へ常に光を発射しているが、移動体が通過することで、その光が遮断される。受光器に光が届かない瞬間を通過情報として検出するシステムを開発した。図 3 にセンサ値取得プログラムを示す。

```

int valSW;

void setup() {
  Serial.begin(115200);
  pinMode(33, INPUT_PULLUP);
}

void loop() {
  valSW = digitalRead(33);
  Serial.print("Sensor Value:");
  Serial.print(valSW);
  if (valSW == 0){
    Serial.println("==Object detction");
  }
  else
  Serial.println("==No object");

  delay(500);
}
  
```

図 3 センサ値取得プログラム

このプログラムでは透過型光電センサが移動体を検知するまでメッセージ「No object」を返し、通過を検知したら「Object detction」を返すプログラムとなっている。また実際に集計されるデータは未通過時に1、通過時に0を無線送信するようになっている。図 4 に実際にセンサ値を取得した場合のシリアルモニタを示す。

```
14:08:47.493 -> Sensor Value:1==No object
14:08:48.021 -> Sensor Value:1==No object
14:08:48.501 -> Sensor Value:0==Object detction
14:08:49.028 -> Sensor Value:0==Object detction
```

図 4 シリアルモニタ

センサが反応した際に「No object」から「Object detction」に変わっていることが確認される。また「Object detction」になった場合にデータが送信され、センサが反応していない状態では通信を行わないシステムとなっている。

昨年度のシステムはポーリングによって物体の通過検知を行っていたが、本年度は割込み処理によって通過検知を行った。そのため昨年度のシステムより物体の通過を正確に検知できるようになった。

2.4 ホームページに速度・タイムの表示

ESP32 からラズパイに C 言語で書かれたプログラムを用いてデータを Wi-Fi で送信する。ラズパイで受信したデータは Python で処理され html プログラムに反映される。今回は様々な Python の拡張モジュールを利用し、html にデータを反映することができた。表 5 に使用したモジュールの一部を示す。

表 5 仕様したモジュール

モジュール	役割
threading	並列実行
HTTPServer	Web サーバの実装
BaseHTTPRequestHandler	イベントハンドラ
time	時間測定
socket	ソケットを利用した通信

表 5 の threading はデータの受信、ホームページの更新などの複数のスレッドを使用して Python のプログラムを並列に実行するためのモジュールである。HTTPServer は Python で簡単にホームページサーバを起動するためのモジュールである。BaseHTTPRequestHandler はホームページ上でイベントハンドラを扱うためのモジュールである。次に time は時間を計測するためのモジュールである。最後に socket モジュールはソケットを利用した通信を行うために使用した。

次にデータの集計と速度表示を行うプログラムの説明を行う。Wi-Fi によって ESP32 より送信されたデータはラズパイに集計される。集計する際に移動体を検出する場所にそれぞれセンサを取り付けた ESP32 を準備す

るが、それぞれの ESP32 を識別するため通信に使用するポート番号を別々に設定した。図 5 にポート番号の設定プログラムを示す。

```
host = '0.0.0.0'
port1 = 50000
port2 = 50001
port3 = 50002

server_socket1 = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server_socket1.bind((host, port1))
server_socket1.listen(1)
print(f"wifi_server is listening on {port1}")

server_socket2 = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server_socket2.bind((host, port2))
server_socket2.listen(1)
print(f"wifi_server is listening on {port2}")

server_socket3 = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server_socket3.bind((host, port3))
server_socket3.listen(1)
print(f"wifi_server is listening on {port3}\n")

data1 = None
data2 = None
data3 = None
```

図 5 データ集計

ここでは 3 台の ESP32 からのデータをそれぞれポート番号 50000, 50001, 50002 で集計している。またデータが来たかどうかを判定する変数として data1, data2, data3 を準備した。移動体の検出は決められたコースを通過することを想定したため、data1, data2, data3 を通過する順番は決まっている。そのためこれらの変数が None でなければ通過したと判断し if 文の入れ子構造により通過時間の情報を管理した。それぞれの子機を通過した時間を用いて通過タイム、ラップタイム、一周の速度を求めるプログラムを作成した。

最後に得られた通過時間、速度の情報を web ブラウザで閲覧できるプログラムを作成した。図 6 に集計したデータを html に渡すプログラムを示す。

```
class SimpleHTTPRequestHandler(BaseHTTPRequestHandler):
    def do_GET(self):
        self.send_response(200)
        self.send_header('Content-type', 'text/html')
        self.end_headers()

        if not result_queue1.empty():
            data1 = result_queue1.get()
            data2 = result_queue2.get()
            data3 = result_queue3.get()
        else:
            data1 = "NO data available,"
            data2 = "NO data available,"
            data3 = "NO data available,"
```

図 6 集計データを html に渡すプログラム

パソコンやスマートフォンなどのクライアントからの接続要求を BaseHTTPRequestHandler モジュールを用いて通信が行われたことをイベントハンドラとして利用した。渡されたデータを用いて html の更新を行うが Python で

は HTTPServer モジュールを用いることで実装することができる。図 7 に実装したプログラムを示す。

```
def start_http_server(host="192.168.179.4",port=5000):
    server_address = (host, port)
    httpd = HTTPServer(server_address, SimpleHTTPRequestHandler)
    print(f"starting HTTP server on {host}:{port}...\n")
    httpd.serve_forever()

start_http_server()
```

図 7 ホームページサーバの実装

このプログラムでは host と port を指定するだけで簡単に Python を用いてホームページサービスを提供できる。実行結果のウェブブラウザ画面は図 8 となる。

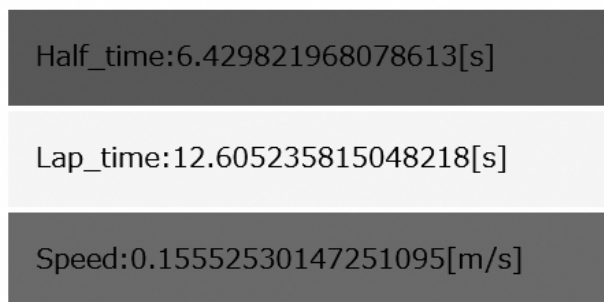


図 8 ウェブブラウザ画面

ここでは通過タイム、ラップタイムそして周回速度が表示されている。なお図 8 は表示テストのため有効数字は考慮していない。今回表示されている内容は実験のため簡単なものとしているがその html ソースの一部は図 9 となっている。

```
<html>
  <head><title>DEMO Data</title></head>
  <body>
    <div style="text-align:center;font-size:40px;">Demo Data</div>
    <p>half_time:[data1][s]</p>
    <p>lap_time:[data2][s]</p>
    <p>speed:[data3][m/s]</p>
    <div style="background:#008b8b;border:#ffffff solid 2px;font-size:12px;">
    <div style="background:#ffff00;border:#ffffff solid 2px;font-size:12px;">
    <div style="background:#ff4500;border:#ffffff solid 2px;font-size:12px;">
  </body>
</html>
```

図 9 html ソース

この html ソースは単純なため装飾などは簡単に変更することが可能となっている。

3. まとめ

今回の研究では昨年度のシステムを発展させて移動体の速度検出システムとその表示を行うシステムの開発を行った。使用したセンサを距離センサから透過型光電センサに変更したことで物体の通過を検知することが容易になった。特に距離センサでは移動体の通過判断を距離で判定するため、移動体の形状で閾値の設

定を変更する必要があったが、今年度のセンサはその必要はない。また昨年度のシステムでは物体の通過をポーリングで検知していたためリアルタイムに通過を検知することが難しかったが、今年度のシステムは割り込み処理を用いたため通過時刻の検知はより精密になった。さらに集計に用いるマイコンをシステム全体に Wi-Fi を提供するマイコンと分けることで接続可能なセンサの台数を増やすことに成功した。同時にラズパイをホームページサーバとしたため、ホームページとしてはより複雑なページを作成することも可能となった。

今後の課題としてはシステム全体をよりコンパクトにするためにパッケージ化と DC-DC コンバータモジュールの作成があげられる。今回の研究ではウェブブラウザでの表示を最優先としたため DC-DC コンバータモジュールは市販品を使用した。技大電子情報技術科では DC-DC コンバータに関する授業もあるため、DC-DC コンバータモジュールの自作に挑戦してみたい。

最後に卒業研究の雑感を述べる。今回の卒業研究では chatGPT に代表される生成 AI を利用した。生成 AI は用語の検索では非常に効力を発揮したが、プログラムソースについては少し信頼性に乏しかった。そのままでは動かないプログラムを修正することに多くの時間を費やしたこともあった。今後の生成 AI の進化は未知数であるが現状、プログラムの構成やアイデアを出すことには優れているが、信頼性に少し欠けるという印象を持った。

参考文献

- 1) 笹原由太, 前田大空, 佐々木謙多, 令和5年度(2023年度)卒業研究概要集, 高速移動体検出システムの開発, p 85-88.

RFID を用いた在庫管理システムの開発

荒木 来伽, 村上 陽菜愛, 小川 斗夢*, 福田 真(指導教員)
電子情報技術科, *情報システム技術科

LAMP 構成(「Linux」「Apache」「MySQL」,「PHP」)を使い, 在庫管理システムを開発する. 現在, 学校内の在庫管理は紙ベースで教員がする仕事である. また, 校内の図書館では本の管理がシステム化されていない. そのため, 多くの備品や本を手軽に管理するシステムが必要とされている. 本卒業研究では RFID タグをリーダーで読み取ったあと, ソフトウェア処理を行い, ウェブブラウザでユーザーが快適に管理を行えるようなシステム開発をしつつ, 処理内容を学ぶことを目的としている. 今回は RFID を用いてデータを取り込み, データベース化してウェブブラウザに表示させる図書管理システムの完成を目指した.

1. 緒言

本研究では RFID リーダー, タグを用いて在庫管理システムを作成する. RFID を利用する理由としては, データ書き換えや同時読み込みが可能のためである.

RFID とは Radio Frequency Identifier の略である. 無線通信などで用いられる高周波数帯のラジオ波を用いて, ID 情報が書き込まれた専用タグとデータのやり取りを行う通信技術のことを言う. 例えば, Suica の非接触型マネー決済, ユニクロのレジにおけるタグの読み込みなどがある. RFID を読み取るまでの一連の流れは以下のとおりである.

- ① RFID リーダライタから IC タグ・RF タグに向けて情報を照射
- ② IC タグ・RF タグのアンテナが情報を受信し, 内部の IC チップが起動
- ③ 起動した IC チップ内の情報を IC タグ・RF タグのアンテナから送信
- ④ 送信された情報を RFID リーダライタが受信
- ⑤ RFID リーダライタが受信した内容を処理システムに送信

RFID リーダーからタグ内のデータを SPI 通信で取り込み, そのデータをウェブブラウザに表示されるようにするシステムを開発した.

2. 実験

2.1 システムの全体

システムの構成は図 1 のとおりである. RFID タグの情報をリーダーが読み取り, そのデータを Arduino マイコンで読み取り, データを在庫管理用データベースサーバの MySQL のテーブルに入れる¹⁾. そして PHP で MySQL のデータをウェブサイト上で見られるようにする.

その後ホームページサーバとして起動させる Apache でウェブブラウザからのリクエストに応じて在庫状況を配信する. 今回は本の貸出を管理する図書管理システムを開発した.

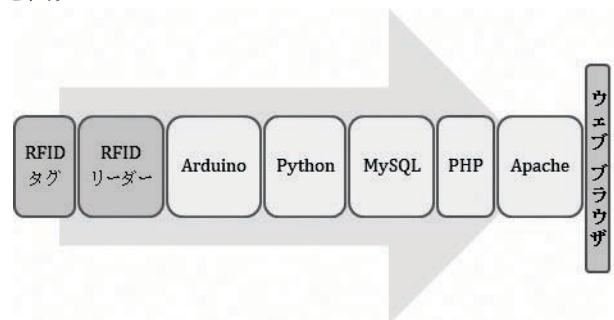


図 1 システムの構成

2.2 システムの外観

今回, RFIDリーダーはSunFounder RFID RC522 モジュールを使用する. また RFID の情報を取得するマイコンは Arduino UNO R4 WiFi BOARD を用いる²⁾. 図 2 に RFID リーダーとマイコンを接続した概観を示す. また RFID タグは図 3 の NFC カードを使用する.

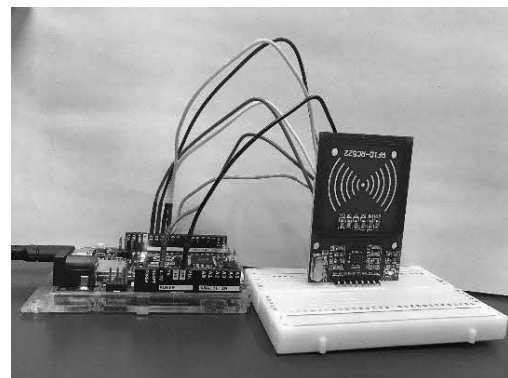


図 2 RFID リーダーとマイコンの外観

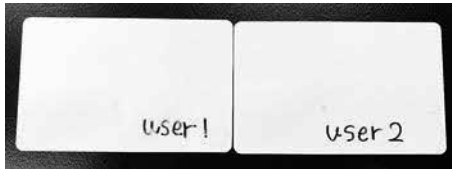


図 3 RFID タグ

RFID タグは今回使用するカードタイプのものやコインタイプのものなど様々な形状がある。カードタイプのものは学生証やバスカードに使用され、コインタイプのものは備品にリングで取り付けなどして使用される。今回は図書管理システムとして利用者の貸出カードと本を管理するタグとしてカードタイプを使用した。

3. 使用機器

今回使用する Arduino のスペックは表 1 である。

表 1 Arduino UNO R4 WiFi BOARD

動作電圧	5V
入力電圧(VIN)	6~24V
DC 電流 (各 I/O ピン毎, 最大値)	8mA
動作周波数	48 MHz
フラッシュメモリ	256KB
SRAM	32KB
価格(Amazon:2月25日調べ)	5,500円

Arduino はルネサスの RA4M1 を搭載しているので RA4M1 のデータシートやマニュアルを参照すれば、SPI インターフェースの動作原理が理解できる³⁾。Renesas RA4M1 マイコンの SPI インターフェースは最大 24MHz で、Arduino のライブラリでは 20MHz が上限である。今回使用する RFID リーダーインターフェースである SunFounder RFID RC522 モジュールの SPI インターフェースは最大 10M bit/s であるので Arduino のスペックで十分と判断できたためこのマイコンを選定した。また価格も 5,000 円台で購入することができ、システム全体を低価格で設計することができる。図 4 は使用したマイコンの外観である。

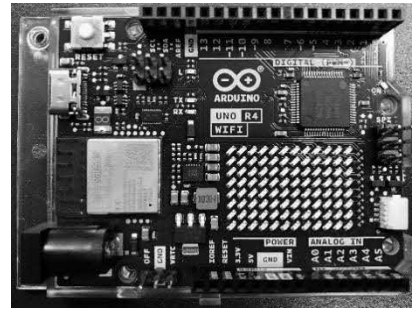


図 4 Arduino UNO R4 WiFi BOARD

使用する SunFounder RFID RC522 モジュールのスペックは表 2 のとおりである。動作周波数は 13.56MHz で、動作電流は 13-26mA / DC 3.3V、読み取り範囲は 0~35mm となっている。データ転送スピードは最大 10M bit/s で、Arduino, RaspberryPi に対応している。図 5 は使用した RFID RC522 モジュールの外観である。

表 2 SunFounder RFID RC522 モジュール

動作周波数	13.56MHz
動作電流	13-26mA / DC 3.3V
読み取り範囲	0~35mm (mifare1 カード)
SPI インターフェイス (データ転送スピード)	最大 10M bit/s
価格(モジュール, キーリング IC タグ, NFC カード×2 セット)	749円



図 5 SunFounder RFID RC522 モジュール

なお今回使用した NFC カードは 1 枚 100 円程度で購入可能である。

4. システムの内観

4.1 マイコンへのデータ読み込み

マイコンと PC をタイプ C-USB コードで接続する。次に RFID リーダーの処理をマイコンで実行するためのコードを図 6 に示す。このコードは Arduino IDE でコンパイルし、Arduino にダウンロードする。プログラム中の Setup の部分は最初の 1 回だけ実行され、RFID リーダーを初期化し、パソコンとシリアル通信 (SPI 通信) を開始して「RC522 RFID Reader Ready」と表示する。次の Loop の部分でタグがあるかないかの確認が繰り返される。その後タグの情報が取り出されて、RFID リーダーがリセットされるという流れになる。

```
1 #include <SPI.h>
2 #include <MFRC522.h>
3
4 #define SS_PIN 10
5 #define RST_PIN 9
6
7 MFRC522 rfid(SS_PIN, RST_PIN);
8
9 void setup() {
10   Serial.begin(9600); // シリアル通信開始
11   SPI.begin(); // SPI通信開始
12   rfid.PCD_Init(); // RC522初期化
13   Serial.println("RC522 RFID Reader Ready");
14 }
15
16 void loop() {
17   if (!rfid.PICC_IsNewCardPresent() || !rfid.PICC_ReadCardSerial()) {
18     delay(50);
19     return;
20   }
21
22   //Serial.print("UID: ");
23   for (byte i = 0; i < rfid.uid.size; i++) {
24     Serial.print(rfid.uid.uidByte[i] < 0x10 ? "0" : "");
25     Serial.print(rfid.uid.uidByte[i], HEX);
26   }
27   Serial.println();
28
29   rfid.PICC_HaltA(); // RFIDリーダーをリセット
30   delay(1000);
31 }
32
```

図 6 プログラム

4.2 取得したデータと MySQL の仲介

本研究では RFID リーダーで読み取ったデータをシリアル通信で受信し、MySQL データベースを用いて書籍、ユーザー情報の管理や貸出、返却処理を行うシステムを Python で実装している⁴⁾。

図 7 は返却処理を行うプログラムの一部である。入力された id の値を持つデータが book テーブルにあるかを調べ、存在しない場合はエラーメッセージを表示し終了する。存在する場合は rental テーブルに入力された id の値を持つ貸出中のデータがあるかを調べ、存在する場合は貸出状況の値を編集してメッセージを表示し終了させる。存在しない場合はメッセージを表示し終了

する。

```
def process_rfid_data(rfid_data):
    """RFIDデータを処理する関数"""
    # bookテーブルでbtagidの存在確認
    cursor.execute("""
        SELECT COUNT(*) FROM book WHERE btagid = %s
        """, (rfid_data,))
    book_exists = cursor.fetchone()[0]

    if book_exists == 0:
        print(f'エラー: btagid: {rfid_data} の記録が見つかりませんでした。')
        return

    # rentalテーブルでbtagidのレコードを検索 (returned = 0 の条件追加)
    cursor.execute("""
        SELECT returned FROM rental
        WHERE btagid = %s AND returned = 0
        """, (rfid_data,))
    result = cursor.fetchone()

    if result:
        # レコードが存在し、returnedが0の場合、1に更新
        cursor.execute("""
            UPDATE rental
            SET returned = 1
            WHERE btagid = %s AND returned = 0
            """, (rfid_data,))
        db.commit()
        print(f'btagid: {rfid_data} を返却しました。')
    else:
        print(f'btagid: {rfid_data} はすでに返却済みです。')
```

図 7 プログラム

4.3 データベース化

Python で処理したデータをデータベース化するために、MySQL を用いる。図 8 は本の情報を管理するテーブルであり、図 9 ユーザー情報を管理するテーブル、図 10 は貸出状況を管理するテーブルである。図 10 中の utagid と btagid はそれぞれユーザーと本を識別するためのタグの ID である。また rentaldate は貸し出された日付となっている。さらに、returned は現在貸し出されているかどうかを表すものである。

Field	Type	Null	Key	Default	Extra
title	varchar(255)	NO		NULL	
writer	varchar(255)	NO		NULL	
publish	varchar(255)	NO		NULL	
price	int	NO		NULL	
publishdate	date	NO		NULL	
categoryid	varchar(3)	YES		NULL	
btagid	varchar(255)	NO	PRI	NULL	

図 8 book テーブル

Field	Type	Null	Key	Default	Extra
name	varchar(255)	NO		NULL	
number	int	YES		NULL	
utagid	varchar(255)	NO	PRI	NULL	

図 9 user テーブル

Field	Type	Null	Key	Default	Extra
utagid	varchar(255)	NO		NULL	
btagid	varchar(255)	NO		NULL	
rentaldate	date	NO		NULL	
returned	int	NO		0	

図 10 rental テーブル

4.4 ウェブブラウザ表示

貸出状況を表示するための PHP プログラムを作成した。MySQL にある rental テーブルと user テーブルと book テーブルからデータを取得し、ウェブページ上に一覧表示する。また、検索条件をフォームに入力し、検索する機能を持ったプログラムとした。

ここまでで PHP プログラムをウェブブラウザとして映し出すことはできるが、ウェブサーバソフトである Apache を用いることで、データの受け渡しややり取りを引き継ぐことができるようになる。今回、コマンドラインで Python プログラムを実行し、そのデータをウェブブラウザに反映させることはできた。完成したウェブブラウザを図 11 に示す。



図 11 ウェブブラウザでの表示

5. 図書管理の自動化

ユーザーが貸出処理の Python プログラム、返却処理の Python プログラム等、それぞれのファイルを起動しなければ処理ができない。そのような手間を省くため、シェルスクリプトを使って、端末上で処理内容をユーザーが簡単に決められるようにした。図 12 は、生徒がユーザーの時の貸出処理である。貸出処理の他に、返却処理、蔵書検索・予約、貸出履歴、予約履歴・キャンセルを行える。また、教員も生徒と同様の機能を利用できる。このシステムの管理者は書籍情報の登録、書籍情報の編集、ユーザー情報の登録、編集、貸出履歴の確認、予約履歴・キャンセルを実現することができる。

```
rfid@sv1:~$ ./test2.sh
ユーザータグを読み込ませてください
読み取った: s
以下の番号を選択してください:
1: 貸出処理
2: 返却処理
3: 蔵書検索・予約
4: 貸出履歴
5: 予約履歴・キャンセル
0: 終了
番号を入力してください: 1
ユーザー認証成功: 04B6027A071294
予約していた書籍があります。
貸出成功: utagid=04B6027A071294, btagid=04D5687A071290
```

図 12 実際に動作させた画面

6. まとめ

今回製作した RFID による在庫管理システムでは、RFID リーダーから得たデータを Arduino で処理し、適切な処理を経たあとデータベースサーバへ書き込むことができた。さらにデータベースサーバに書き込まれたデータは PHP でウェブブラウザに映し出すことができた。同時に 2 台の RFID リーダーを利用しても問題なくデータの更新がなされたため、実際の図書管理システムとして稼働することも可能であると思われる。また同一の RFID リーダーで図書情報もユーザー情報も読み取ることが可能であるためシステム全体のコストダウンを実現することもできた。

今後の課題としては、端末上で Python を起動しなくてもウェブブラウザ上だけで処理を進めることができるようにすることである。また、ユーザー側が見やすく、操作しやすいウェブサイトにすることが課題としてあげられる。その実現のためには、Python, MySQL, PHP, Apache を適宜変えていく必要があるだろう。

今後は図書管理システムの実装のみならず、学校内の備品管理のためのシステムの開発を目指したい。また図書管理システムとして返却期限が迫った図書についてメールを送信するなどのシステムの実装にも挑戦してみたい。

参考文献

- 1) progate“MySQL でデータベースを作成しよう”
<https://prog-8.com/docs/MySQL-database-setup>(参照 2025 年 1 月 30 日)。
- 2) Indoor Corgi Let us be your smart tech buddy “Arduino IDE のインストールと設定 (Windows, Mac, Linux 対応)”
<https://www.indoorcorgielec.com/resources/Arduinoide/Arduino-ide>(参照 2025 年 1 月 30 日)
- 3) KARAKURI MUSHA “【Arduino】UNO R4 WiFi のはじめかた(ことはじめ)”
<https://karakuri-musha.com/inside-technology/Arduino-unor4wifi-tips-helloworld01/>
(参照 2025 年 2 月 4 日)。
- 4) Qiita“Ubuntu に MySQL をインストールして、Apache の PHP からアクセスするまで”
<https://qiita.com/yuu1111main/items/98baacaf381a2b778ccc>
(参照 2025 年 1 月 30 日)。

LEGO SPIKE を用いたプログラミング教材の開発

明石滂斗, 三浦功大, 上門汰輝, 西川奨真, 坂田聡(指導教員)

情報システム技術科

近年, 日本において情報教育の重要性が高まり, 大学入学共通テストに「情報」科目が導入されるなど, プログラミング教育の必要性が強調されている. しかし, 中学生がプログラミングに触れる機会は限られており, 論理的思考力の育成や情報分野への興味を促進するための効果的な教育方法が必要とされている. 本研究の着想の背景として, 私たちは1年次に LEGO® MINDSTORMS® EV3 を用いたライトレースの設計を行う授業を経験した. この経験を活かし, EV3 の後継モデルである LEGO® Education SPIKE™ プライムセットを活用し, 中学生向けのプログラミング教材の開発を試みた. 本研究では, 中学生が論理的思考力を養いながら, プログラミングに興味を持てるような教材を設計することを目的とする.

1. はじめに

今日, デジタル技術の急速な発展に伴い, プログラミング教育の重要性が高まっている. 日本においても, 2022 年度より高等学校で「情報I」が必修科目となり, アルゴリズムやプログラミングの基礎を理解することが求められるようになった.

このため, 日本の中学校では 2021 年度からプログラミング教育が導入されているが, 課題点として指導内容の多さに比べて, 授業時間は少ないという実態がある¹⁾.

本研究では, 中学校でのプログラミング教育の促進に貢献するため, 中学生を対象とした LEGO® Education SPIKE™ プライムセット(以下 SPIKE とする)を活用したプログラミング教材の研究を行い, 論理的思考や制御構造の概念を学べる教材を作成することを目的とする.

2. プログラミング教育に適した教材開発

2.1 教材開発の方向性

中学校段階でのプログラミング教育は, 「生活や社会の中から情報の技術に関わる問題を見出して課題を設定する力」, 「課題の解決策を, 条件をふまえて構想し, 全体構成やアルゴリズムをアクティビティ図等に表す力」, 「試行・試作等を通じて解決策を具体化する力」等の育成が中心的な目的とされている²⁾.

これらの目的をふまえ, 本研究では SPIKE を活用し,

中学生がプログラミングの基礎を学びながら, 論理的思考力を養うことができる教材の開発を目指す. 特に, 視覚的に理解しやすいビジュアルプログラミングを用いることで, 初心者でも直感的にプログラムの流れを把握できるようにする. さらに, ロボットを実際に動作させることで, 試行錯誤を繰り返しながら学習できる環境を整えることにする.

また, プログラミング教育の実践事例では, チームで協力するグループワークが多く行われているため, 本研究でもそれにならい, グループワークを中心としたチーム学習に取り組んでもらう.

2.2 SPIKE とは

本研究で使用する SPIKE を図 1 に示す. SPIKE は, カラフルなレゴブロックパーツと使いやすいハードウェア, ドラッグ&ドロップで直観的に使える Scratch ベースのプログラミング言語を組み合わせた小学校高学年から中学, 高校生向けの STEAM 教材である³⁾.

中学生向けプログラミング教材の開発において, SPIKE は, 学習者がブロックを組み立てるという手法を通じて, プログラミングの基本的な概念を視覚的に理解できる. 中学生にとって, 実際に手を動かし, 目の前で動くロボットを操作できる体験は, 抽象的なプログラミング概念を身近に感じさせ, 学習への興味を引き出すことが可能である.



図 1 LEGO® Education SPIKE™ プライムセット

2.3 プログラミング方法

SPIKE で使用できるプログラミング環境として、ワードブロック、アイコンブロック、python の 3 種類が準備されているが、本研究ではワードブロックを採用した。

図 2 にワードブロックのプログラム作成画面を示す。

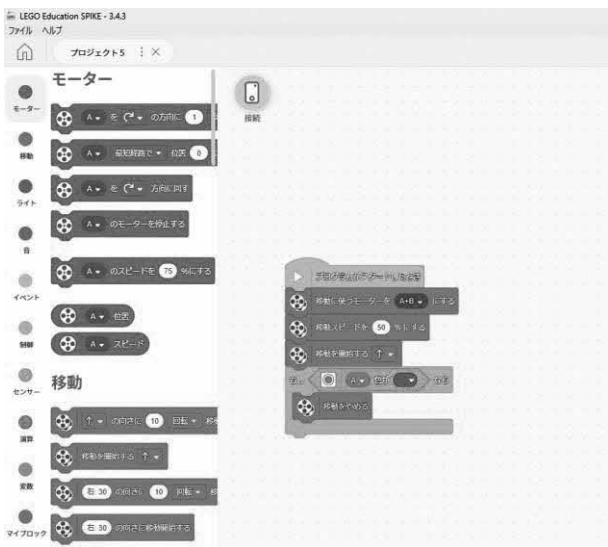


図 2 プログラム作成画面

ワードブロックは、モーター、移動、制御等の操作を行えるブロックを組み合わせることでプログラミングが可能である。外観は、MIT メディアラボの Scratch と非常によく似ている。Scratch は世界中の教育現場で取り入れられていることから、本研究でも Scratch とよく似たワードブロックは、中学生のプログラミング教材として最適であると判断した。

2.4 コースの作成

SPIKE を使用したプログラミング教室を行うにあたり、

SPIKE の実習用コースを作成した。コースの大きさは、中学校の教室で使用することを前提として設計し、それでいて十分な広さといえる 120cm×150cm とした。

中学生にプログラムを設計・実践してもらう上で、本研究で重視したのは条件分岐、ループ構造といった制御構造の体験である。そのため、単純な条件分岐が行える色の判定を基にしたコース設計とした。図 3 に作成したコース図を示す。

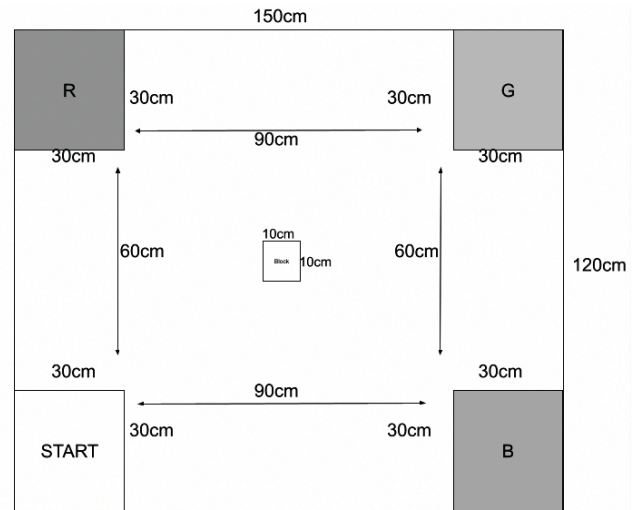


図 3 コース図

このコースでは、主に目的地までの移動、物体の保持、カラーセンサーによる色の判別、色の判別後の条件分岐の 4 つの機能をプログラムできることを意識した。

このコースは、図3の真ん中にあるブロックエリアに置かれるランダムな色のブロックを、その色のエリアに運ぶことが目的である。それぞれの機能の詳細については、次章で後述する。

2.5 説明用資料の作成

本教材の学校での実施を想定し、授業の流れに沿った説明用資料を作成した。

資料の内容としては、まず、プログラミングの基礎概念や SPIKE の基本操作を解説し、初心者でもスムーズに学習を進められるようにした。特に、ビジュアルプログラミングのブロック構造や、ロボットの動作制御に関する基本的な考え方を図やスクリーンショットを用いて視覚的に説明することで、直感的に理解しやすい構成を意識した。

3. ブロックキャッチャーについて

3.1 機能

本教材で作成するロボット「ブロックキャッチャー」について説明する。ブロックキャッチャーとは、掴んだブロックの色を認識し、目的の場所に運ぶことを目的としたロボットである。このロボットは、カラーセンサーを使用してブロックの色を判別し、ブロックエリアにランダムに配置された色ごとのブロックを、正しい場所へと運ぶことができる。ブロックキャッチャーの機能は目的地までの移動、物体の保持、カラーセンサーにおける色の判別、色の判別後の条件分岐である。作成した機体を図4に示す。

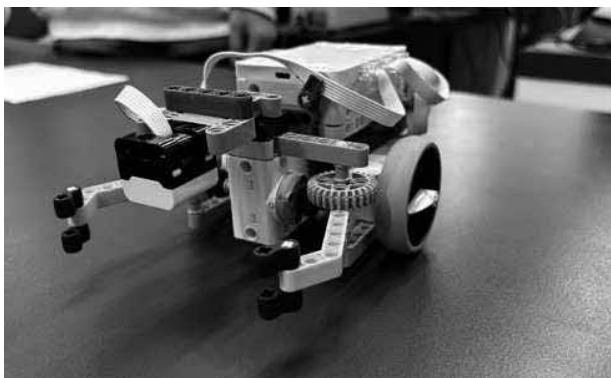


図4 作成した機体

本教材ではこの機体を学生に一度作成してもらい、その後、自分たちがプログラムを作成する際に、自由に改良してもらう。

以下に、各機能の詳細を述べる。

3.1.1 目的地までの移動

本機体はモーターを用いて指定した目的地まで移動させる。移動の制御には、モーターの回転数や距離センサーを活用し、正確な位置へ移動することができるが、初期設定ではモーターの回転数のみを設定し直進や旋回を行う。

3.1.2 物をつかむ

目標地点まで物体を保持する機能であるアーム部分にはモーターを搭載し、開閉動作を行うことで物体を保持できる。物体を掴む動作は、あらかじめ設定した角度制御によって行う設計となっている。

3.1.3 色の判別

カラーセンサーを用いることで、保持した物体の色を識別する機能を実装している。色を識別し、その色に応じた移動動作を決定する役割を担っている。使用したカラーセンサーを図5に示す。



図5 カラーセンサー

3.2 フローチャート

本機体の動作を明確にするため、制御の流れをフローチャートで示す。フローチャートでは、機体がどのように情報を取得し、判断し、動作するかを視覚的に整理することで、プログラムの構造を理解しやすくしている。本機体が図3のコース図で動作する際のフローチャートを図6に示す。

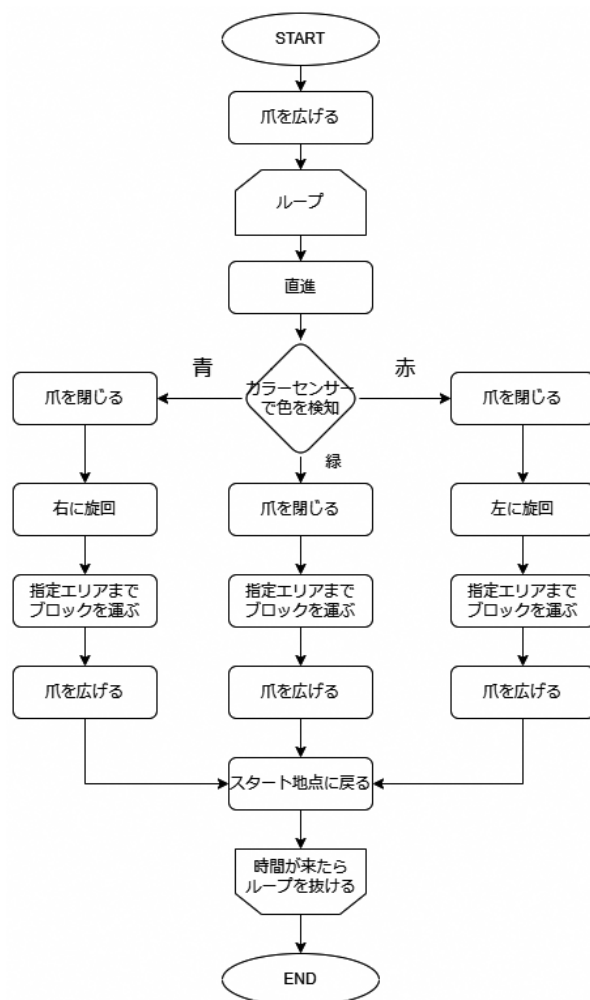


図6 フローチャート

このフローチャートは、中学生に提示する資料の一部であるため、プログラミングに初めて触れる中学生の

ことも考え、直感的に理解しやすいようにシンプルな処理の流れを意識して作成した。

動作としては、カラーセンサーで認識した色に応じて、機体の動作を変更する条件分岐を設けた。例えば、赤色の場合は赤のエリアに移動し、青色の場合は青エリアに移動させるといった制御を行う。この流れをループさせることによって、色に応じた分類作業を自動化させている。

4. シミュレーション

プログラミング経験のある大学生数人に協力してもらい、授業のシミュレーションを行った。これは、教材や説明用資料の内容を事前に検証し、実際の授業で生じる可能性のある課題点や改善点を明確化することを目的としている。

シミュレーションは、実際の授業を想定した形式で行い、まず SPIKE の基本機能や基本動作を説明した後、参加者が SPIKE を用いてロボットとプログラムの作成・実行をする形で行った。その後、カラーセンサーを活用した条件分岐や、物体の仕分けといった応用的な演習を行い、最後に教材や説明資料に関するフィードバックを収集した。

シミュレーションの結果、説明資料の組み立てガイドの写真の角度が悪く、組み立て手順が分かり辛い箇所があることと、組み立てガイド自体が、必要なパーツと手順のみの簡素なものとなっているため、すべてのグループが正しく組み立てを行うには、補助者のサポートが必要ではないかといった意見があった。また、プログラムのエラー処理に関する説明が不足している。よくあるミスと対処法を事前にまとめておくとよい、といった指摘もあったが、概ね問題なく分かりやすい資料であったと肯定的な評価を受けた。

これらの結果をふまえ、説明資料の改良や授業の進め方の調整を行うことで、よりスムーズな学習体験を提供できるように改善を図れると考える。

5. おわりに

本研究では中学生向けのプログラミング教材の研究・開発を行い、最終的な教材のモデルと授業資料を完成させ、初心者でも直感的に理解できる内容を意識した教材を作成することができた。当初は中学生を対

象とした実践的な授業の実施を予定していたが、諸事情により授業の実施には至らなかった。教材の研究に関しては、中学生のプログラミング教育の現状や今回使用した SPIKE について理解することから始め、教材開発と授業準備を行った。完成した教材を用いた大学生によるシミュレーションを行うことで、本教材の有効性や改善点が指摘された。

特に、組み立てガイドの写真角度の調整やエラー発生時の対応方法の明確化が、授業の質の向上につながった。今後は、これらのフィードバックを反映し、実際の教育現場で活用しやすい資料へ改良を進める必要がある。

今後の課題としては、中学生を対象とした実践的な授業を行い、実際の学習効果や理解度を評価することが挙げられる。また、プログラミング初心者の学習過程をより詳細に分析し、より直感的で分かりやすい教材の開発を進めることも重要であると考えられる。

参考文献

- 1) 一般社団法人日本産業技術教育学会, 中学校プログラミング教育の実態調査の報告—令和元年度 技術・家庭科技術分野「D 情報の技術」の現状—, https://www.jste.jp/main/teigen/210127_jr_c_hosa.html (参照日 2025 年 1 月 21 日)。
- 2) 文部科学省, 教育の情報化の手引き-追補版-, https://www.mext.go.jp/content/20200608-mxt_jogai01-000003284_004.pdf(参照日 2025 年 1 月 23 日)。
- 3) LEGO® Education, LEGO® Education レゴ® エデュケーション SPIKE™ プライムセット, <https://education.lego.com/ja-jp/products/lego-education-spike-prime-set/45678/>(参照日 2025 年 1 月 23 日)。

toio と micro:bit を用いたプログラミング教材の作成

岩下大樹, 松村亮汰, 宮本凌太郎, 坂田聡(指導教員)
情報システム技術科

本研究では, 小学校教育におけるプログラミング教材として, キューブ型ロボット toio と教育向けのマイコンボードである micro:bit を用いて, 宝探しゲームを作成した. この教材は, Scratch を用いて作成し, 児童は, toio の動作を制御し, プログラミング思考を育むことができる. 宝探しのゲーム性を通じて, 児童が論理的思考を学び, 最適な動作を考えることを促進する. また, 本教材内にはレベル設定などの機能を取り入れている.

1. 研究背景・研究目的

文部科学省は学校教育分野における情報化の推進のため, 2020 年 4 月から小学校でプログラミング教育が必修化されている.¹⁾²⁾ その目的は, 次の通りである.

- プログラミング的思考の育成
- プログラムや情報技術の社会における役割について気付き, それらを上手に活用してよりよい社会を築いていこうとする態度を育む
- 各教科等の中で実施する場合については, 「教科等での学びをより確実なものにする」

本学の所在地である菊池郡菊陽町では, プログラミング教材として, キューブ型ロボット toio が小学校に導入されている³⁾⁴⁾ ため, 本研究ではプログラミング学習用教材として toio を用いている. 今年度は教育向けマイコンボードである micro:bit を併用することで, toio の機能を拡張した教材の開発に取り組んだ. ビジュアルプログラミングが可能のため, 小学生向けの教材作成に適していると考え採用した.

2. プログラミング教材に用いる機器の概要

2.1 toio とは

本研究で用いるプログラミング教材である toio を図 1 に示す. toio は, 幅 31.8mm, 奥行き 31.8mm, 高さ 20mm の小型キューブ型ロボットであり, ブロックや工作物をキューブに乗せて bluetooth 接続されたコントローラーでの操作や, 自立走行が可能な体感型 Toy-プラットフォームである.

このキューブ型ロボットに用いられている技術は, リアルタイム絶対位置検出や, 6 軸姿勢検出・イベント検出



図 1 キューブ型ロボット toio

がある. リアルタイム絶対位置検出とは専用のマットなどに印字された目に見えない特殊パターンをキューブ底部の読み取りセンサー(光学センサー)で認識し, 位置情報に変換するものである. また, 6 軸姿勢検出・イベント検出とは 6 軸姿勢検出システム(6 軸加速・3 軸ジャイロ)により, キューブの三次元的な姿勢や動きだけでなく, 外からの刺激による様々なイベントを検出することができる.

2.2 micro:bit とは

micro:bit は, 英国放送協会(BBC)が主体となって開発した教育向けのマイコンボードである. ユーザーが動作をプログラミングできる 25 個の LED と 2 個のボタンスイッチのほか, 加速度センサと磁力センサ, 無線通信機能(BLE)を搭載している. USB ケーブルで PC と接続し, プログラムをドラッグアンドドロップで書き込むことが可能である.⁵⁾ micro:bit 本体を図 2 に示す.

2.3 toio キューブと micro:bit の関係

toio と micro:bit は Bluetooth 接続を介して連携が可

能である。また、micro:bit の電源は USB 接続または単四電池 2 本で供給される。

この2つの機材の共通点として、どちらもローコードプログラミングが可能である。

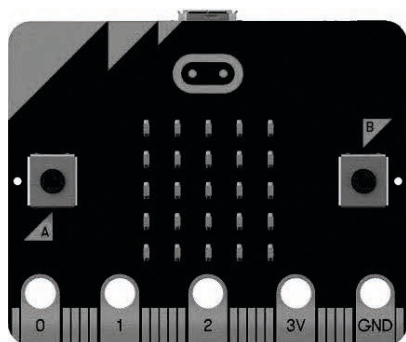


図2 マイコンボード micro:bit

3. プログラミング教材の開発

3.1 ライントレース

本研究では、小学生向けのプログラミング教材として、toio キューブと micro:bit を組み合わせ、ロボットプログラミングの一般的な題材であるラインレースロボットの作成を試みた。

ラインレースは、toio キューブが黒いラインを自動で追従するシステムである。micro:bit に GROVE 社のラインファインダ v1.1 を接続し、ラインの有無を検出することで toio の左右のモーター回転数を調整し、ラインを外れないように進行する。これにより、児童は「センサーを用いた自立移動」という制御方法を体験できる。図3にラインレースのフローチャートを示す。

実際にラインレースロボットを作成し、教材としての特徴を分析した結果、以下の点が明らかとなった。

1. プログラミングの実践的な体験：
ラインレースのプログラムを組むことが容易であり、センサーによるライン検出やモーター制御も併せて体験可能
 2. プログラムの自由度が低い：
シンプルな構造のため、児童が独自の工夫を加えにくい
 3. 組み立てる段階で個体差が生じる：
センサーの取り付け位置などで正常に動作しない恐れがある
- これらの特徴を踏まえ、組み立て作業は事前準備で完了させ、児童にはプログラムの作成に専念させる形とした。

また、プログラミングを作成以前に、まずプログラミン

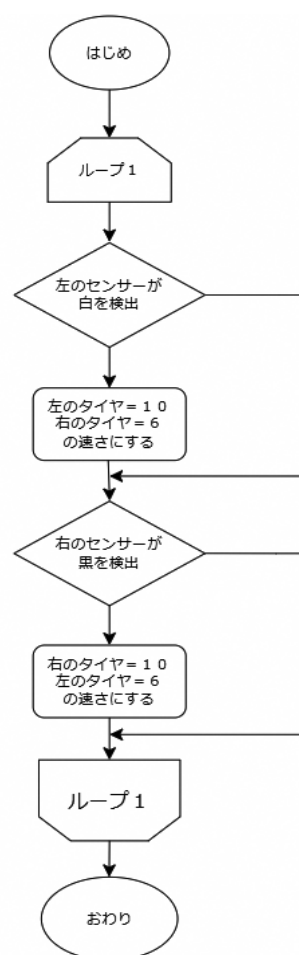


図3 ライントレースのフローチャート

グ自体に興味を持ってもらうことも大事だと考え、micro:bit のジャイロ機能を活用しゲーム形式で論理的思考を学習可能な教材も併せて開発する。

3.2 宝探し

小学校教育におけるプログラミング教材として、toio キューブと micro:bit を組み合わせた「宝探しゲーム」を開発した。この教材は toio の動作を micro:bit で制御することで、児童が論理的思考や順次処理を学習することを目的としている。この「宝探しゲーム」は、micro:bit のジャイロセンサーを使用して、micro:bit を傾けることで toio キューブを前後左右に移動させ、プレイマット上に隠された宝物を探し出すものである。

本ゲームでは、児童の理解度に応じて選択可能な 3 段階のレベル(レベルの概要は後述3. 2. 4)を設定している。児童は実物の toio キューブを動かしながら宝探しを行うが、プレイマット上での toio キューブの位置情報はプログラムを介してゲーム画面と同期しており、プレイマット上もゲーム画面上も表示されない宝を、経路を網羅し探し当てる設計となっている。

この「宝探しゲーム」を通じて、児童は論理的思考を楽しく体験しながら学ぶことができると考えられる。図 4 に宝探しのフローチャートを示す。

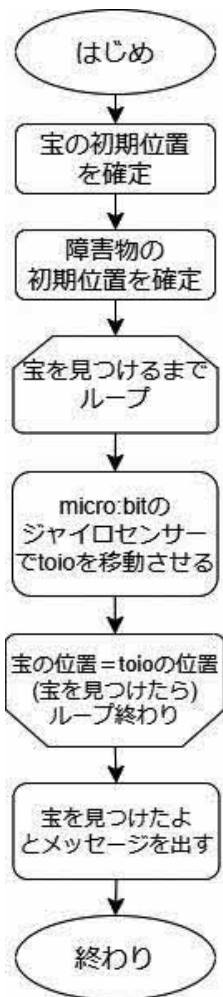


図4 宝探しのフローチャート

3.2.1 開発環境

toio キューブの動作制御プログラミングには、プログラミング言語の Scratch⁶⁾を使用し、開発環境として toio Do⁷⁾を使用した。Scratch とは MIT メディアラボで開発されたビジュアルプログラミング言語⁸⁾で、ブロックを組み合わせるだけでプログラミングの作成が可能である。そのため、世界中の教育現場で取り入れられており、本研究の目的である小学生向けプログラミング教材の開発環境として最適であると判断した。

今回作成した toio キューブの動作に関するプログラミング開発環境画面を図 5 に示す。

3.2.2 アプリの構成

本研究で作成した学習教材である宝探しの、ハードウェア及びソフトウェア構成は以下のとおりである。

ハードウェア構成は、toio キューブと micro:bit、Scratch が使用できるパソコンまたはタブレット、そしてト

イオ・コレクション付属のプレイマットである。

ソフトウェア構成は、Scratch を使用して toio キューブの動作を制御するプログラムである。このプログラムには、宝探しのセットアップとアクションの 2 つのフェーズが含まれている。セットアップではレベル設定や初期配置のランダム化などが行われ、アクションではプレイヤーの命令を toio キューブに伝える処理が行われる。セットアップ画面とアクション画面を図 6, 7 に示す。



図 5 開発環境画面

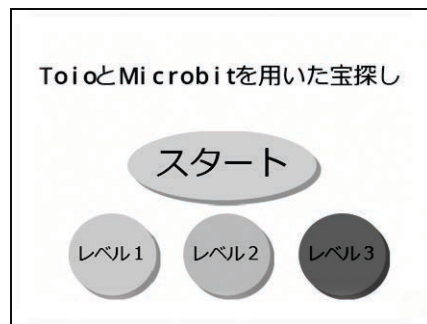


図 6 セットアップ画面

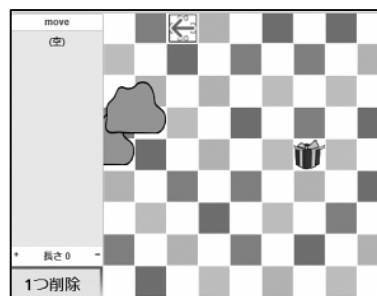


図 7 アクション画面

3.2.3 レベル設定

はじめにプレイヤーはレベル(難易度)の選択を行う。レベルは、ゲームの動作を学習するためのレベル 1 から、後述する侵入不可エリアが出現する難易度の高いレベル 3 までの 3 段階を設定した。レベル 1 は、基本的な移動操作に慣れることを目的としており、レベル 2 では宝の数を3つに増やし、児童が複数の目標を探すことで探索力と計画性を養うことを目的としている。レベル 3 ではプレイマット上に仮想的な障害物を配置し、進

行を妨げるエリアが発生する。これにより、児童は障害物を避ける工夫を求められる。また、出現する宝の数は変更可能である。

3.2.4 初期配置のランダム化

宝探しに使用する toio キューブとゲーム内に登場する宝は、ランダムに配置されるようにプログラムした。

toio キューブや宝の位置座標は、プレイマットの中心を原点として横方向(x 軸)と縦方向(y 軸)にそれぞれ±4 の整数値で定義される。この位置座標を使用して、レベル設定に応じた障害物の数と宝、および toio キューブが重複しないように初期配置する。

3.2.5 侵入不可エリア

侵入不可エリアとは仮想的な障害物のことであり、図7 に示したアクション画面のように石オブジェクトが配置される。侵入不可エリアの初期配置は、宝や toio キューブの初期配置アルゴリズムと同様の物を使用し、すべてが重複しないように設定される。侵入不可エリアに toio キューブ侵入すると、画面上に警告文が表示され強制的に1マス前の位置に移動される。なお、侵入不可エリアは最高レベルの3 を選択した場合のみ出現するように設定している。

4. 模擬授業による評価

本研究で作成した教材を使用して、模擬授業を実施した。授業の生徒役として、プログラミング教育を受けている大学生6名を対象に、実際の授業と同じ流れで教材を説明し、操作を行ってもらった。授業終了後に聞き取り調査を行い、本研究で作成した教材の評価を確認した。なお今回は「宝探し」のみでシミュレーションをしている。

評価結果として共通している項目は以下のとおりである。

① 操作の直感性

micro:bit を傾けることで toio を操作する方法は、多くの協力者にとって直感的で分かりやすいものであった。しかし、微妙な傾け具合による移動量の調整が難しいとの意見もあり、調整のための補助機能の必要性が示唆された。

② ヒントの活用

画面上に宝の位置が表示されているが、小さくて見えない設計のため、障害物を避けながら全経路を探索する必要がある。この点については「ゲーム感覚で楽しく取り組める」との意見が多かった一方、「ヒントがなく、どこを探せばよいか分からない」とい

う意見も見られた。ヒントの段階的な提示や、宝に近づくほどヒントの頻度を増やすなどの工夫が求められる。

③ 障害物の影響

レベル3では障害物が配置されており、toio の進行を妨げることでより戦略的な思考が求められる設計となっている。シミュレーションでは、「障害物を回避するルートを考える必要があるので工夫のし甲斐がある」という肯定的な意見があった。

シミュレーション授業を通じて、教材の基本的な仕組みは問題なく機能することが確認できたが、操作の調整やヒントの工夫など、いくつかの改善点が明らかになった。今後は、これらの課題を解決するための改良を加えたうえで、改めて小学校での活用を目指す。

5. まとめ

今回の研究では、toio キューブと micro:bit を用いたプログラミング教材の作成を行った。micro:bit のジャイロセンサーと toio キューブの座標処理動作に関するプログラミングを行うことで、ロボットプログラミングの基礎を学ぶためのできる教育コンテンツとして設計した。模擬授業で指摘された改善点について検討を行い、小学校におけるプログラミング教材としての実用性を向上させていきたい。

参考文献

- 1) 文部科学省 プログラミング教育に関する資料, https://www.mext.go.jp/a_menu/shotou/zyouhou/detail/1416328.htm(参照日 2025年1月23日)。
- 2) STAR プログラミングスクール 小中高のプログラミングが必修化～, <https://www.star-programming-school.com/article/programming-compulsory/>(参照日 2025年1月23日)。
- 3) 菊陽西小学校の取り組み, https://www.town.kikuyo.lg.jp/kiji0032869/3_2869_up_gp3heke6.pdf(参照日 2025年1月23日)。
- 4) toio のテクノロジー, <https://toio.io/platform/technology.html>(参照日 2025年1月23日)。
- 5) 新しくなった BBC micro:bit, <https://microbit.org/ja/new-microbit/>(参照日 2025年1月23日)。
- 6) Scratch について, <https://scratch.mit.edu/>(参照日 2025年1月27日)。
- 7) toio Do とは, <https://toio.io/do/>(参照日 2025年1月27日)。
- 8) ビジュアルプログラミングとは, <https://toio.io/blog/visual-programming/>(参照日 2025年1月27日)。

ドローンを用いたプログラミング的思考を育むための教材開発

相澤由依子, 岩木大河, 岩谷聖也, 松山惟生, 小田信彦(指導教員)
情報システム技術科

小学校では、2020 年度からプログラミング的思考の育成に重点を置いたプログラミング教育が必修化され、児童の論理的思考力や問題解決能力の育成が進められている。本研究では、小学校のプログラミング教育への貢献を目的とし、本学で学んだ情報技術を基盤とした教材を開発した。特に、ドローンを活用した学習プログラムを開発し、実際に小学校でのプログラミング教室を実施した。本年度は菊陽北小学校において授業を行い、開発した教材の有用性を検証するとともに、改善点に関する知見を得た。

1. はじめに

近年、人工知能の飛躍的な進化によって、人の働きを必要としない職業が増加するなど、社会の在り方が大きく変化すると予測されている。このような急激な技術革新に伴う社会の変化の中で、次世代を担う子供たちには、情報技術を効果的に活用しながら、論理的かつ創造的に思考し、課題を発見・解決していく力が求められている。そのため、「プログラミング的思考」の育成が重要視されている。「プログラミング的思考」とは、自分が意図する一連の活動を実現するために、どのような動きの組み合わせが必要かを考え、それぞれの動きに対応する記号をどのように組み合わせればよいのか、さらにその記号の組み合わせをどのように改善すればより意図した活動に近づくのか、といったことを論理的に考える力であると定義されている¹⁾。

このような背景のもと、日本では 2020 年度から小学校において「プログラミング的思考」の育成に重点を置いたプログラミング教育が実施されている。しかし、教育現場では指導者の技術的スキルの不足や教材の充実度といった課題があり²⁾、効果的な教育方法の模索が続いている。

本研究では、小学校のプログラミング教育への貢献を目的とし、本学で学んだ情報技術を活かした教材を開発した。その有用性を検証するため、近隣の小学校においてプログラミング教室を実施することとした。

2. プログラミング教育に適した教材開発

2.1 教材開発の方向性

本研究では、楽しさ、学びの定着、創造性を重視しながら教材開発を進めた。

小学校のプログラミング教育では、専門的なプログラ

ミングスキルの習得ではなく、「プログラミング的思考」の育成に重点が置かれている。そのため、本教材では、児童がプログラミング行為自体を難しく感じないように、適切なプログラミング言語とプログラミングデバイスを選定した。

プログラミング結果を表現する手段として、ディスプレイ上に表示されるバーチャルなものではなく、実際に目の前で動作し、児童が楽しさを実感できるものを採用した。

また、プログラミング教室を効果的に進めるためのテキストを作成した。このテキストは、プログラミング初心者でも無理なく学べるよう、基本から応用までを段階的に解説し、達成感を得ながらステップアップできる構成とした。さらに、ゲーム形式の課題では、児童が自由に課題を選択できる仕組みを取り入れることで、創造性を高める工夫を施した。

2.2 ドローンについて



図1 Tello EDU 本体

本教材では、プログラミング結果を表現する手段として、子供たちにとって未来的なテクノロジーの象徴であるドローンを採用した。児童は、プログラミングの結果をドローンの三次元的な動きとして自らの目で視覚的に確認できるため、学習成果が分かりやすく、また主体的に学習意欲を維持しやすい。

具体的には、本教材で使用するプログラム言語と連携可能な Tello EDU³⁾ を選定した。Tello EDU はプロペラ回転時の接触事故を防ぐためのプロペラカバーを備えており、安全性が高い。

2.3 Scratch について

本教材では、プログラミング言語として Scratch を採用した。Scratch は、アメリカのマサチューセッツ工科大学 (MIT) メディアラボが開発したプログラミング言語であり⁴⁾、ブロックを組み合わせることでプログラムを作成できる。直感的な操作が可能であることから、世界 200 以上の国と地域で利用されており、最も普及しているプログラミング学習ソフトの一つである。

また、プログラミングデバイスには、小学校の授業で使い慣れているタブレット型 PC を採用した。



図 2 Scratch の画面

2.4 ドローンコースの作成

本教材では、ドローンを飛行させる手順を考える題材として、飛行コースを作成した。本コースは、学習の段階に応じて柔軟に変更ができる設計としている。

図 3 は、離陸 → 150 cm 前進 → コーンを 1 周回る → ゴールまで前進するという基礎的な飛行を行うコースである。

図 4 では、プログラムの繰り返し文(ループ)を活用し、ドローンが一定の飛行を繰り返しながら進むコースを設計した。具体的には、ドローンが前進しながら、コーンの手前で下降 → 通過 → 上昇という一連の飛行を繰り返すプログラムを組むことを想定している。

図 5 では、画像認識と条件分岐を活用し、ドローンがコース上のマーカーの色を認識しながら適切な判断をして進む仕組みを導入した。たとえば、緑色のマーカーを認識した場合は右へ旋回し、赤色のマーカーを認識した場合は上昇するといった条件分岐のプログラムを組むためのコースである。なお、色付きマーカーは機械システム技術科の協力を得て、ランダムに表示されるものを使用した。

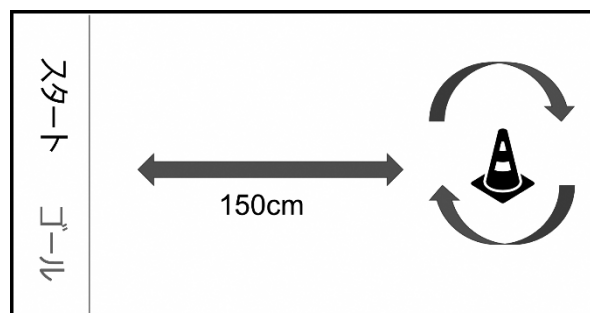


図 3 基礎的なコース

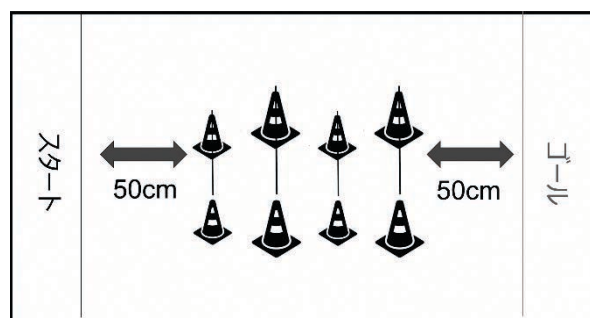


図 4 繰り返し文を用いたコース

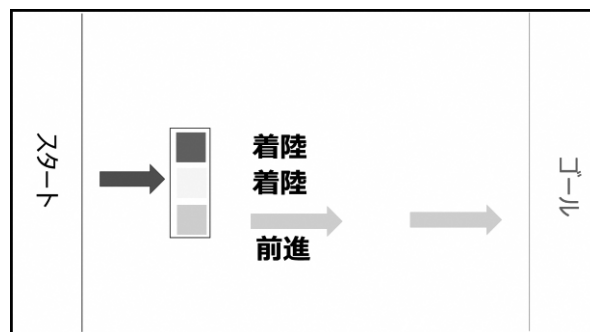


図 5 画像認識を用いたコース

図 6 は、これまでのすべての障害物を取り入れたゲーム形式のコースである。ゲームのルールは加算方式とし、コース内の障害物を自由に通過できる。同じ障害物は何回でも飛行可能とするが、3 回目以降は 1 回ごとに 3 点減点する。ゴール時には 10 点を加算し、スタートからゴールにかかった時間が基準タイムに近いほど高

いボーナス点を与える。さらに、条件分岐文や繰り返し文を活用したプログラムには追加点を付与する。

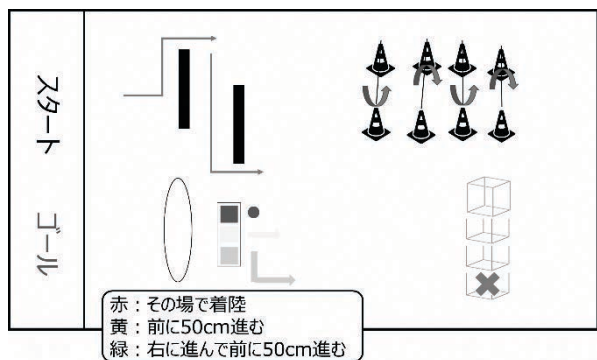


図 6 ゲーム形式のコース

2.5 プログラミング教室のテキスト

本教材のテキストは、プログラミング初心者でも無理なく学べるよう、基本から応用までを段階的に解説し、達成感を得ながらステップアップできる構成とした。これにより、学習内容の定着を図ることを目的としている。また、ゲーム形式の課題では、児童が自由に課題を選択できる仕組みを取り入れることで、創造性を高める工夫を施した。

テキストの内容としては、プログラミングの基本概念、注意事項、ドローンの動かし方、条件分岐文の説明、繰り返し文の説明、ゲームのルールを記載した。図 7 に作成したテキストを示す。

図 7 プログラミング教室のテキスト

3. プログラミング教室の出前授業

本年度は、菊陽町立菊陽北小学校の4年生から6年生の22名を対象にプログラミング教室を実施した。児童を上級生と下級生の2つのグループに分け、授業を進めた。まず、図8のようにプログラミングの基本概念や使用するツールについて説明を行った。その際、作成したテキストを活用し、児童が理解しやすいように工夫した。



図 8 テキストを使って教えている様子

また、単なる講義形式ではなく、児童自身が手を動かしながら学べるように配慮した。具体的には、コースを見ながらグループメンバーと協力しつつフローチャートを作成する活動を取り入れた。この過程を通じて、児童は論理的な思考の組み立て方を学ぶことができた。

続いて、作成したフローチャートに基づいて Scratch でプログラムを組み、図 9 のようにドローンを飛行させる作業を行った。初めは思うように飛ばないこともあったが、その都度問題点を見つけ出し、修正を重ねながらプログラムの完成を目指した。児童が主体的に問題を解決できるよう、必要に応じてヒントを出すなどのサポートを行った。

今回のプログラミング教室に対する評価は概ね良好であり、本教材はプログラミング学習に有用であると判断した。担当の小学校教員や児童からは、「児童は試行錯誤しながら学ぶ楽しさを実感している様子だった。」「難しいと思っていたフローチャート作成も、教師役の大学生の丁寧な説明で理解できるようになった。」などの意見が寄せられた。

以下に、今回のプログラミング教室で明らかになった問題点と、それらに対する改善案を示す。

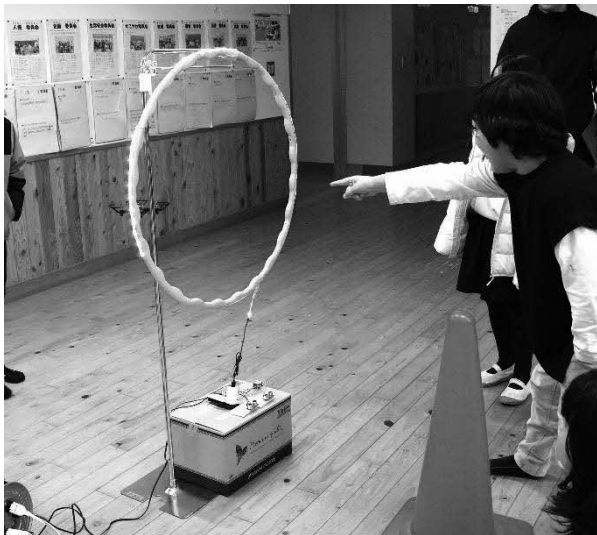


図9 ドローンを飛行させている様子

一つ目は、繰り返し文が児童にとって難解な構文であることを想定できていなかったことである。また、大学生の指導スキルも不足していたため、繰り返し文の説明に時間がかかりすぎてしまった。今回の経験から、児童は繰り返し範囲の設定を容易にはできないことが分かった。そのため、繰り返し文の基礎を学ぶための練習問題を追加するなどの工夫が必要と考える。

二つ目は、時間管理ができていなかったことである。授業の進行において、説明・実践・準備などの時間配分を十分に考慮していなかった。その結果、予期せぬ事態が発生した際に調整できず、多くの未実施内容が残ってしまった。この問題に対しては、事前にシミュレーションを十分に行い、実施内容ごとに制限時間を設定することで改善を図る。また授業中はタイムキーパー役を配置し、時間管理を徹底することが望ましい。

三つ目は、ドローンが電磁波の影響を受け、安定した飛行ができなかったことである。オープンキャンパスの経験を踏まえ、大学生が持つスマートフォンの通信機能を停止させたが、それでも電磁波の影響が見られた。原因として、プログラミング教室を実施した会場のWi-Fi設備の影響や、狭い空間でタブレットPC同士の距離が近かったことが考えられる。この問題に対しては、Wi-Fiの周波数帯の影響を受けにくいドローンの使用を検討する必要がある。

4. おわりに

本研究では、小学校でのプログラミング教育への貢献を目的とし、Scratch, タブレットPC および Tello EDU を活用したプログラミング学習用教材を開発し、菊陽北

小学校にてプログラミング教室を実施した。

今回のプログラミング教室を通じて、教材の有用性が確認できた一方で、準備不足による時間管理の甘さや、環境の影響によるドローン動作の不安定さなどの問題点が明らかになった。

今後は、これらの問題点を解決したうえで、さらに有用性の高い教材の開発を進めるとともに、教師役の大学生の指導スキル向上にも取り組んでいきたい。これにより、小学校のプログラミング教育の質の向上に貢献できると考える。

参考文献

- 1) 文部科学省, 「小学校段階におけるプログラミング教育の在り方について(議論の取りまとめ)」, https://www.mext.go.jp/b_menu/shingi/chukyo/chukyo3/053/siryo/_icsFiles/afiedfile/2016/07/08/1373901_12.pdf, (参照 2025-02-06).
- 2) 「小学校のプログラミング教育の現状とは? 今後の課題についても解説!」, <https://miraii.jp/programming-71>, (参照 2025-02-06).
- 3) 「Tello 公式ウェブサイト」, <https://www.ryzerobotics.com/jp/tello-edu>, (参照 2025-02-06).
- 4) 「Scratch(スクラッチ)」, https://www.mext.go.jp/miraino_manabi/content/285.html, (参照 2025-02-06).

Raspberry Pi を用いた節電対応デジタルサイネージの制作

柿本匠, 高田湧眞, 渡辺晴斗, 坂田聡(指導教員)
情報システム技術科

本研究ではセンサーと Raspberry Pi を利用して、節電対応のデジタルサイネージを作成することを目的としている。システムは、汎用の液晶モニターと Raspberry Pi で構成されており、Raspberry Pi に取り付けられたセンサーが人を感知した場合だけ、モニターに所定の情報を表示することで、動作時間を減らすものである。今年度は 3 種類のセンサーを用いて人の感知を行い、デジタルサイネージとしての動作と昨年度に制作したシステムに比べて節電効果を検証した。

1. はじめに

デジタルサイネージとは、ディスプレイなどの電子的な表示機器を使用して情報を発信するメディア形式である。デジタルサイネージは、紙媒体と比較して情報の更新が容易であり、学内でも PR 動画の掲示に活用されている。しかし、従来のシステムでは閲覧者が不在でも映像が流れ続け、手動で電源を管理する必要があるため、効率的な運用が求められていた。

本研究では、人を感知して必要な場合にのみ映像を表示し、それ以外はディスプレイをスタンバイ状態に切り替えることで、消費電力の削減を実現するシステムを開発した。

2. システムの概要

2.1 開発環境

表 1 に開発環境を示す。プログラムの作成は Raspberry Pi 上で行い、Python を用いた。各センサーと Raspberry Pi 本体は GPIO で接続し、外部ライブラリである RPi.GPIO で制御した。外部コマンドとして選出した画像を表示させる画像ビューア `sxiv` と、映像出力関係の制御をするための `xrandr` を追加インストールして使用した。

2.2 ハードウェアの構成

(1) Raspberry Pi 4 Model B

Raspberry Pi とは、2012 年に Raspberry Pi 財団から発売された ARM プロセッサを搭載した高性能なシングルボードコンピュータである。小型で比較的安価ながら、高いパフォーマンスを提供する製品である。また、Raspberry Pi は効率的なエネルギー利用と省電力機能により低消費電力で動作することができる。本研究では、Raspberry Pi 4 Model B を使用した。このモデルはメモリが 4GB、USB2.0 ポート 2 つ、USB3.0 ポート 2 つ、

表 1 開発環境

OS	Raspberry Pi OS (kernel 6. 1. 21)
使用言語	Python
外部コマンド	<code>sxiv</code> (画像ビューア), <code>xrandr</code>
外部ライブラリ	RPi.GPIO

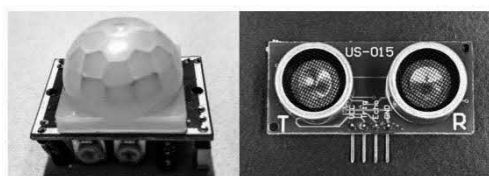


図1 HC-SR501

図2 US-015

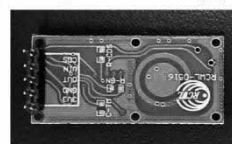


図3 RCWL-0516

Micro-HDMI ポート 2 つを搭載している。

(2) 各種センサーについて

本システムに用いるセンサーとして、移動する人物を検知可能、かつ、安価で入手しやすいものを 3 種類選出した。センサーの外観を図 1 から図 3 に示し、表 2 に各センサーの仕様を示す。

● 赤外線人感センサー(HC-SR501)

人体から発せられる赤外線を感知することでスイッチが入る仕組みである。

● 超音波距離センサー(US-015)

左側の発信部から超音波を発信し、対象物から反射してくる超音波を右側の受信部で受信し、距離を測定する。

● マイクロ波レーダーセンサー(RCWL-0516)

ドップラーレーダー技術を使用して、移動物体と人体を検出できる。

表 2 各センサーの仕様

	赤外線	超音波	マイクロ波
動作電圧	5~20V	5V	4~28V
検出距離	7m 以内	4m 以内	7m 以内
検知角度	120 度以内	15 度以内	120 度未満
測定方法	赤外線	超音波	波長の測定
測定項目	熱源感知	距離	物体移動
通信方法	デジタル		
価格	330 円	350 円	110 円

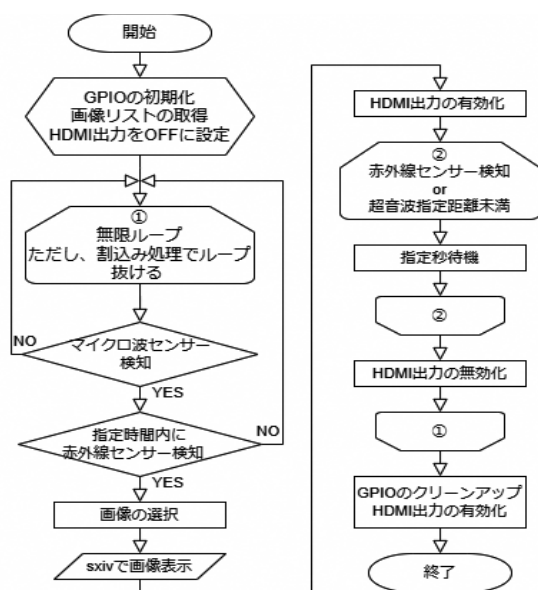


図 4 フローチャート

これらのセンサーを選定した理由は、一般的に流通しており入手しやすく、比較的安価であることに加え、異なる原理に基づいて人の存在を検知できるためである。複数の種類を使用することで、検知精度を向上させることを意図している。

(3) ディスプレイ

PC 用の汎用ディスプレイとして、「iiyama ProLite XB2481HSU」(以下「モニター」)を使用した。

2.3 プログラム構成

図 4 にプログラムのフローチャートを示す。このプログラムには待機時間 2 ヶ所と距離 1 ヶ所が設定可能である。今回の研究では、これらの箇所の最適値を検証し、実装することでシステムとしての完成を目指す。

2.4 HDMI の入出力制御

このシステムでは、モニターをスタンバイの状態にするために、xrandr コマンドを使用した。通常、Raspberry Pi では tvservice コマンドを用いるが、今回の環境では使用できなかったため、代わりに xrandr コマンドを使用して HDMI 出力の制御を行い、モニターをスタンバイ状態に切り替えた。

2.5 画像取得と表示プログラム

本システムでは、Python を使用してフォルダー内の画像を取得し、ランダムに選択して表示するプログラムを実装している。指定したフォルダー内の画像ファイルをすべて検索し、リストに格納する。取得した画像リストからランダムで 1 つの画像を選択し、sxiv を用いて全画面表示する。この仕組みによって、フォルダーに画像を追加するだけで表示コンテンツを簡単に更新でき、柔軟な運用が可能となる。

3. 動作検証

3.1 画像表示開始トリガーと表示タイミング

本システムでは、モニターへの HDMI 出力を無効にすることで、待機中はモニターをスタンバイ状態にしている。人を感知した際にモニターに画像を表示するためには、適切なトリガー条件の設定が必要である。検知プロセスは、まずマイクロ波レーダーセンサー(以下「マイクロ波センサー」)で物体の動きを検知し、次に赤外線人感センサー(以下「赤外線センサー」)で人間であるかを確認する二段階方式を採用した。そのため、2 つのセンサーが同時に検知した瞬間をトリガーにすると誤検知が発生する可能性がある。このため、マイクロ波センサーで動きを検知した後、任意の時間内に赤外線センサーが人を検知した場合に画像を表示するようにプログラムを設計した。トリガーの時間設定によってシステムの挙動が変化するため、複数の設定値で検証を行った。

3.1.1 検証環境・方法

図 5 のように、センサーの前に 3m 四方の検証エリアを設け、各方向から人を移動させた際の検知精度を確認した。検証では、表 3 の基準に従って評価を行った。

表 4 に稼働率と適正率の算出方法を示す。各データ検証回数は 10 回とし、適正率が最も高い値を設定値にする。

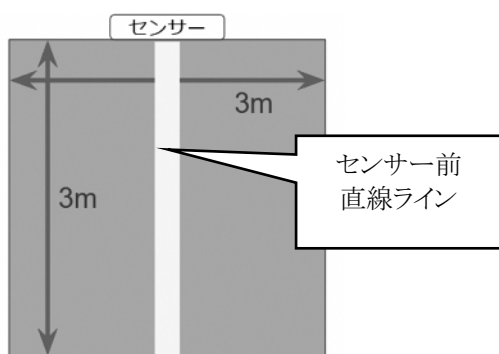


図 5 検証環境

表 3 評価方法

評価分類	評価説明
適正	人がセンサー前ラインを越える前に画像が表示された
遅い	人がセンサー前ラインを越えた後に画像が表示された
×	人が検証エリア (3m四方) から出ていくまでに画像が表示されなかった

表 4 算出方法

稼働率(%)	人を検知できた回数/検証回数×100
適正率(%)	評価が“適正”の数/検証回数×100

3.1.2 検証結果

表 5 に、検証値の評価結果を示す。この評価結果から、人の検知稼働率がどの条件においても高く保てたことは、このシステムの運用上最も重要な要件を満たしていることが示された。反応のタイミングに関しては、評価結果から 6 秒が最適だと判断した。6 秒より短く設定した場合、赤外線センサーが検知できず画像が出力されるプログラムが動作しないことがあった。一方で、待機時間を長く設定した場合、マイクロ波センサーが待機中に複数回反応してしまい、人がいない状態でも画像を表示してしまうようになった。この検証結果から、検証値を長くすると誤作動が増えるため、適正だと判断した 6 秒をこのプログラムの設定値にする。

3.2 画像表示継続処理検証

人が見ている間は情報を継続表示するため、赤外線センサーまたは超音波距離センサー(以下「超音波センサー」)で人の存在を 10 秒間隔で継続的に確認する処理を実装した。具体的には、赤外線センサーが検知

表 5 検証値評価

検証値[秒]	稼働率[%]	適正率[%]
3	100	60
4	100	40
5	100	50
6	100	70
7	100	50
10	100	60

表 6 超音波センサーから離れる距離を変えて検証

検証距離[m]	継続可判定[回/10 回]
1	10
1.5	5
2	2
3	0

または距離が 3m 以内である間は画像を表示し、範囲外になった場合は HDMI 出力を停止し、画像の表示を終了するよう設計している。これにより、閲覧者が画面を見ている間、画像が意図せず消えることを防ぐ。

3.2.1 検証環境・方法

設定値の有効性を確認するための検証を行う。超音波センサーから離れる距離を 1m から 3m の間で検証を行い、継続率を検証した。ここでいう継続率とは、センサーが人を検知して表示を開始した後、再度人を検知して継続処理を行った回数を検証回数で割ったものである。また、表示を開始した画像の表示時間は、継続処理中の人の存在を判定する間隔とほとんど同じである。しかし、センサーが反応してから画像が表示されるまでに機器の都合によるわずかな遅延があるため、人の存在を判定する間隔を短くすると画像の表示時間が短くなり閲覧者が画像から情報を読み取れない可能性がある。反対に、人の存在を判定する間隔を長く設定すると画像の表示時間は長くなるが、消費電力が増加し節電効果が低下する。今回は 10 秒に設定し、この値が適切であるかを実測によって確認した。検証回数はそれぞれ 10 回とし、検証距離を 1m から 3m の間で変えて検証を行った。

3.2.2 検証結果

表 6 はセンサーから離れた検証距離と継続した回数を示している。センサーからの距離が離れるにつれて画像の継続表示が難しくなることが確認された。これは、

表 7 検証結果

人存在判定間隔[s]	実測表示時間[秒/10 回]
5	3.4
10	8.2
15	13.2
20	18.5

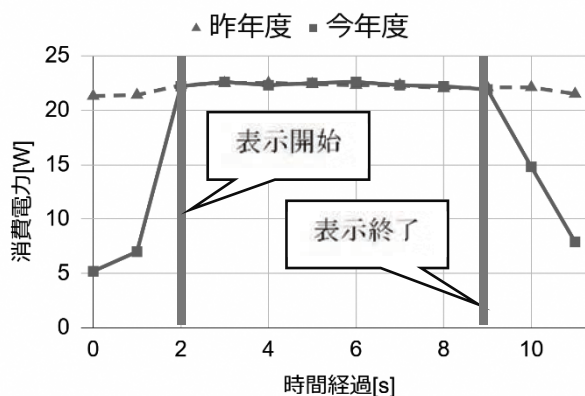


図 6 消費電力の推移

人が立ち止まる位置がセンサーから遠くなるほど、継続的な検知が困難になるためである。しかし、本システムは 23.8 インチのモニターを使用し、近距離での閲覧を想定しているため、この結果は実用上許容できる範囲であると判断した。

表 7 には、プログラム上で設定した人の存在を判定する間隔と、実際に画像が表示されている時間を計測してまとめたものである。この結果より、判定の間隔を 10 秒に設定することで実用性と節電効果のバランスを図れると判断した。

4. 消費電力の計測

昨年度のシステムでは、ディスプレイに黒い画面を常時表示し、人を感じると動画に切り替えて再生する方式を採用していた。しかし、この方法ではディスプレイが常に稼働し続けるため、スタンバイ状態に移行せず、消費電力が高いままとなっていた。今年度のシステムでは、人を感じた場合のみディスプレイに画像を表示し、感知しない場合はディスプレイをスタンバイ状態に移行させるプログラムを実装することで、より高い節電効果を目指した。

本システムと昨年のシステムにおいて、人を感じて画面表示を行う際の消費電力の変化を図 6 に示す。▲ポイントは昨年制作したシステム、■ポイントは本システ

表 8 平均消費電力

	平均消費電力[W]
昨年度画像表示中	22.2
昨年度画像非表示中	21.6
本システム画像表示中	21.6
本システムスタンバイ時	5.81

ムを指している。縦線は、グラフ上における画像表示の開始と終了を示している。グラフや表 8 のデータから分かるように、昨年のシステムは画像を表示している前から消費電力がほとんど変わっていない。しかし、本システムでは、画像表示前は約 5.8W と低い消費電力で待機しており、画像表示中は昨年とほぼ同じ約 22W で動作している。この結果から昨年のシステムと比べて大幅な消費電力削減を実現することができた。

5. まとめ

この研究を通じて、昨年よりも節電効率の高いデジタルサイネージの制作をすることができた。複数のセンサーを組み合わせることで、より人の感知精度が高いシステムを制作することができた。フォルダーに写真を追加するだけで画像を表示する仕組みにより、誰でも簡単に扱える点もこのシステムの長所である。結果として、校内で実用できるデジタルサイネージの制作ができた。

参考文献

- 1) そぞら, 【電子工作入門】ラズベリーパイで超音波距離計を作る, <https://qr.paps.jp/issCn>.(参照日 2024 年 6 月 26 日).
- 2) メタエレ実験室, 【Raspberry Pi】マイクロ波レーダー型人感センサーを使ってみた, <https://qr.paps.jp/Yw87U>.(参照日 2024 年 7 月 3 日).
- 3) Hatena Blog, Raspberry pi で HDMI 出力の ON OFF, <https://qr.paps.jp/TiVQu>.(参照日 2024 年 7 月 3 日).

発行者：熊本県
所属：技術短期大学校
発行年度：令和7年度(2025年度)