踊る貯金箱

難波　升飛　服部　悠紀
信川　望夢　花房　樹

１．研究概要

本研究では、貯金を楽しく続けられる装置を作ることを目的として、３D プリンタと Arduino Nano を用いた踊る貯金箱を作製した。10 円、50 円、100 円、500 円硬貨を判別できる硬貨選別機構を備え、投入金額を正確に検出できる構造とした。

さらに、サーボモータによって頭や腕が動き、貯金額に応じて動作や音が変化する仕組みを取り入れることで、利用者の貯金意欲を高める工夫を行った。

２．開発環境。

- Arduino IDE

  ソースコードの作製には Arduino IDE を使用した。Arduino IDE はオフライン環境で利用でき、Arduino Nano のプログラム作成および書き込みを行うことができるため、本研究に適している。

- Creality ５．３

  ３D プリンタには Creality ５．３を使用し、貯金箱の各部位のパーツを作製した。硬貨選別部、頭部、腕などの部品を個別に印刷し、組み立てを行った。

- Autodesk Fusion 360

  ３D プリンタ用の印刷モデルの設計には Autodesk Fusion 360 を使用した。各部品の寸法や形状を調整しながら設計することで、実際の部品に適したモデルを作製した。

３．研究の具体的内容

（１）硬貨選別部分の作製（写真１）

構造の大型化を避けるために 10 円、50 円、100 円、500 円の硬貨の判定のみの設計とした。

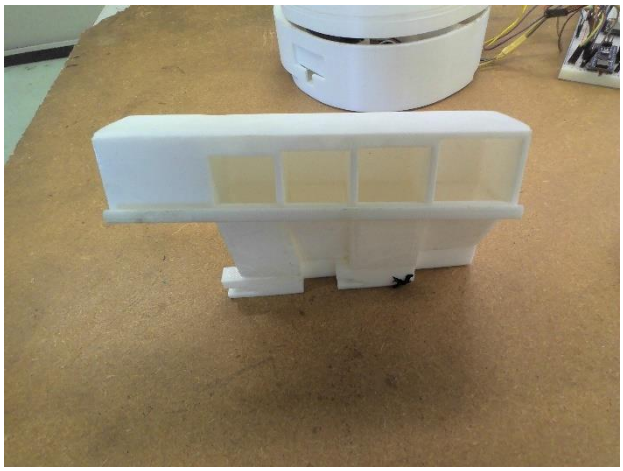

写真１　硬貨選別部

（２）頭部の作製（写真２）

３D プリンタを使用して土台を作製した。内部にサーボモータを搭載し、水平垂直方向に動く目や頭部の開閉を実現した。

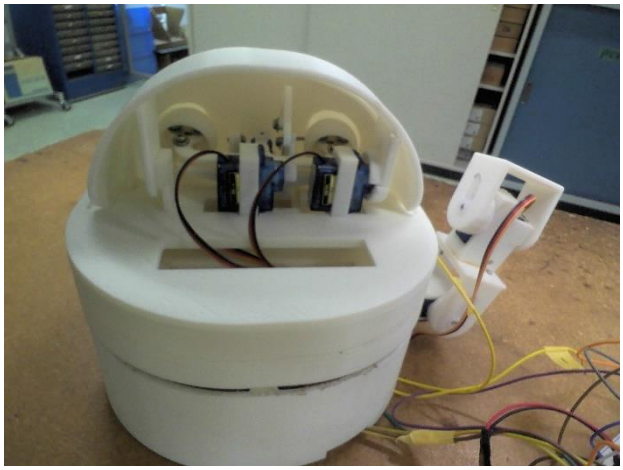

写真２　頭部

（３）腕の作製（写真３）

３D プリンタを使用して腕の外枠を作製した。外枠内にサーボモータを搭載し腕の振りによる踊り動作を実現した。

写真３　腕

（４）内部回路（写真４）

LCD ディスプレイに合計金額を表示する回路を作製した。

使用機器

- Arduino Nano
- PCA9685(複数のサーボモータ制御用)
- LCD ディスプレイ
- フォトインタラプタ(硬貨検知用)
- DFplayer mini(MP３再生用)
- スピーカ
- サーボモータ(大２、小７)

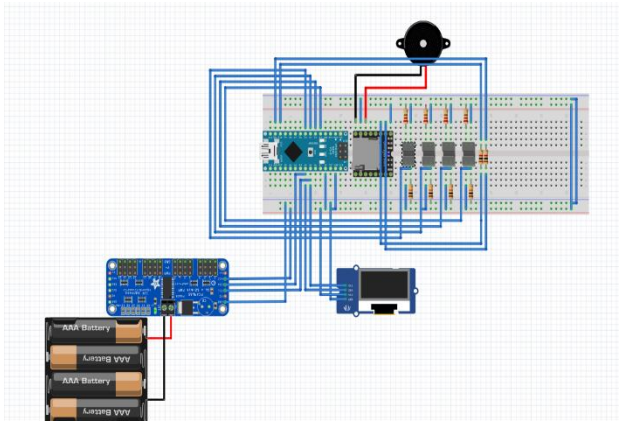

写真４　内部回路

（５）プログラムの作成

メイン制御ループ

```
void loop() {
  unsigned long currentTime = millis();
  checkAllSensors(currentTime);
  if (resetFlag) { resetDisplay();
resetFlag = false; }
  if (!actionReady && lastCoinTime > 0 &&
(currentTime - lastCoinTime > WAIT_TIME))
{
    actionReady = true;
    performAction();
  }
  delay(10);
}
```

- 10 ミリ秒周期でループを実行し、リアルタイム性を確保した
- ３つの主要処理を各周期で順次実行：
  １：全センサの状態監視
  ２：リセット条件のチェック
  ３：動作開始タイミングの判定
- 状態フラグ管理でシステムの動作モードを制御した

コイン検出システム

ポーリング制御の実装

```
void checkAllSensors(unsigned long
currentTime) {
  int currentState =
digitalRead(PHOTO_10);
  if (currentState != last10State) {
    if (currentState == LOW &&
(currentTime - last10Debounce >
DEBOUNCE_TIME)) {
      handleCoin(COIN_10, currentTime);
      last10Debounce = currentTime;
    }
    last10State = currentState;
  }
}
```

- 順次監視方式: 4 つのセンサを 10ms 間隔で巡回チェックした
- 変化検出アルゴリズム：
  1 ：現在値と前回値を比較
  2 ：LOW 状態とデバウンス時間経過を確認
  3 ：条件満足でコイン処理関数を呼び出し
- 効率的なリソース利用：単一マイコンで複数センサを管理

デバウンス処理

```
#define DEBOUNCE_TIME 100
if (currentState == LOW && (currentTime - lastDebounce > DEBOUNCE_TIME)) {
  handleCoin(COIN_VALUE, currentTime);
}
```

- 機械的ノイズ除去：スイッチ接触時の信号振動を無視した
- 100 ミリ秒ルール：状態変化後、100ms の安定期間を要求
- 誤検出防止：1 枚のコインに対して複数カウントを防止

個別サーボ制御

```
void moveServo(int servoNum, int angle) {
  int limitedAngle = constrain(angle, minAngle, maxAngle);
  int pulse = map(limitedAngle, 0, 180, SERVOMIN, SERVOMAX);
  pwm.setPWM(servoNum, 0, pulse);
}
```

- 安全制限：各サーボの可動範囲を constrain 関数で制限
  1. 瞼：90-140 度（自然な開閉範囲）
  2. 目：50-130 度（視野制限範囲）
  3. 腕・首：物理的限界内に設定
- 信号変換：角度（0-180 度）
- PWM 出力：PCA9685 にパルス信号を送信

複数サーボ同時制御

```
#define MAX_SIMULTANEOUS_SERVOS 4
void moveMultipleServos(int servos[], int angles[], int count) {
  if (count <= MAX_SIMULTANEOUS_SERVOS) {
    for (int i = 0; i < count; i++) {
      moveServo(servos[i], angles[i]);
    }
  }
}
```

- 同時動作制限：電源容量を考慮して最大４サーボに制限
- 配列ベース制御：サーボ番号と角度を配列でグループ化
- 同期動作：for ループで複数サーボを連続制御

階層的動作選択

```
void moveServosByAmount(int amount) {
  if (amount <= 500) {
    moveServoLevel1();//目の基本動作
  } else if (amount <= 1000) {
    moveServoLevel2();//目＋腕の動作
  } else if (amount <= 2000) {
    moveServoLevel3();//目＋腕＋首の動作
  } else {
    moveServoLevel4();//全身の豪華動作
  }
}
```

- 条件分岐構造：if-else if 文で金額帯を判定
- 比例原則：投入金額が多いほど複雑な動作
- 拡張性：新規動作レベルの追加が容易

基本動作例

```
void moveServoLevel1() {
  for (int pos = 90; pos <= 140; pos += 2) {
    moveServo(SERVO_SG90_1, pos);
//瞼を開く
    delay(60);
  }
  for (int angle = 0; angle < 360; angle += 10) {
    int x = 90 + 20 * cos(angle * PI / 180);
    int y = 90 + 20 * sin(angle * PI / 180);
    moveServo(SERVO_SG90_2, x);
//目の水平運動
    moveServo(SERVO_SG90_3, y);
//目の垂直運動
    delay(60);
  }
}
```

- 滑らかな動き：for ループで角度を漸次変化
- 円運動アルゴリズム：三角関数で目の円運動を実現
- 時間制御：delay 関数で動作速度を調整
- 段階的開閉動作：for ループにより瞼の角度を少しずつ変化させ、急激な動きを防いだ。
- 滑らかな動き：角度を漸次変化させることで、自然なモータ動作を実現した。
- 時間制御：delay 関数を用いて動作速度を一定に保ち、視認しやすい動作とした。
- 安全性配慮:角度範囲を制限することで、機構部への過度な負荷を防止した。
- 左右独立制御：水平用と垂直用のサーボモータを個別に制御し、複雑な視線移動を可能とした。

金額読み上げ機能

```
void playAmount(int amount) {
  int thousand = amount / 1000;
  int hundred = (amount % 1000) / 100;
  int ten = (amount % 100) / 10;
  int one = amount % 10;

  if (thousand > 0) {
    myDFPlayer.playFolder(1, thousand);
    delay(1000);
    myDFPlayer.playFolder(1, 12);//"千"
  }
  //百、十、一の位も同様
}
```

- 桁分解アルゴリズム：除算と剰余演算で各桁を分離
- 順次再生：上位桁から順に音声ファイルを再生
- 自然な読み上げ：数値＋単位の組み合わせで自然な発音

効果音選択システム

```
int getRandomSoundForAmount(int amount)
{
  int baseSound = 14;
  if (amount <= 500) {
    return baseSound + random(0, 3);
  } else if (amount <= 1000) {
    return baseSound + 3 + random(0, 3);
  }
  //他のレベルも同様
}
```

- ランダム選択：各レベルで３種類からランダムに選択
- ファイル管理：連番ファイルをレベルごとにグループ化
- 多様性確保：同じレベルでも異なる効果音を提供

LCD 表示制御

```
void updateDisplay() {
  oled.clear();
  oled.setCursor(0, 0);
  oled.print("Total: ");
  oled.print(totalAmount);
  oled.print(" Yen");

  if (totalAmount >= RESET_AMOUNT) {
    oled.setCursor(0, 2);
    oled.print("Thank you for");
    oled.setCursor(0, 3);
    oled.print("trying!");
  }
}
```

- 画面クリア：表示前に全画面を消去
- カーソル制御：setCursor で表示位置を指定
- 条件付き表示：2500 円達成で特別メッセージを表示

リセット条件判定

```
void handleCoin(int coinValue, unsigned long currentTime) {
  totalAmount += coinValue;
  if (totalAmount >= RESET_AMOUNT) {
    resetFlag = true;
  }
}
```

- 閾値監視：2500 円に達したらリセットフラグを設定
- 非即時実行：メインループでリセット処理を実行

完全リセット処理

```
void resetDisplay() {
  myDFPlayer.playFolder(1, 26);
//感謝メッセージ
  performThanksAction();//特別動作

  totalAmount = 0;
  lastCoinTime = 0;
  actionReady = false;

  updateDisplay();
}
```

- 多面的リセット：
  １：音声：感謝メッセージ再生
  ２：動作：特別なサーボ動作
  ３：データ：全変数を初期化
  ４：表示：画面を更新

拡張性

- モジュール構造：機能ごとの関数分割で保守性向上
- パラメータ調整：定数定義で動作特性を容易に変更

（６）３Ｄプリンタモデルの製作

AutodeskFusion360 を使って印刷モデルを作製した。硬貨選別部（写真５）、頭部と腕（写真６）、全体モデル（写真７）、製作した全体像（写真８）

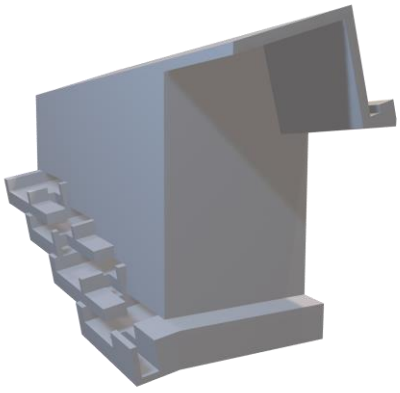

写真５　硬貨選別部

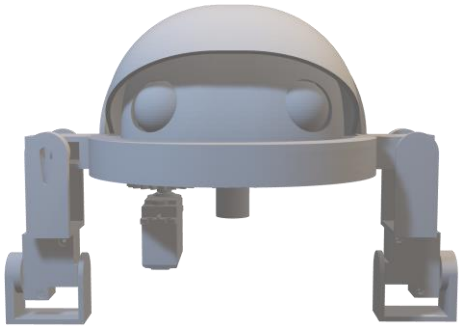

写真６　頭部と腕

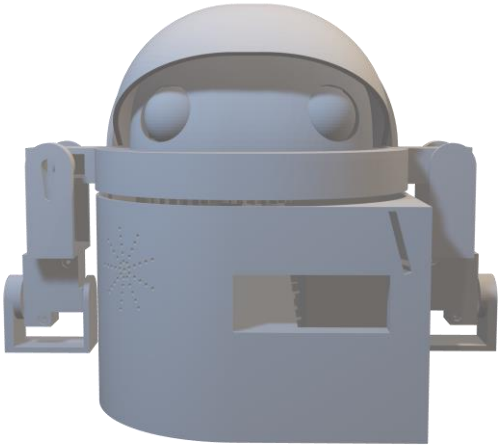

写真７　全体モデル

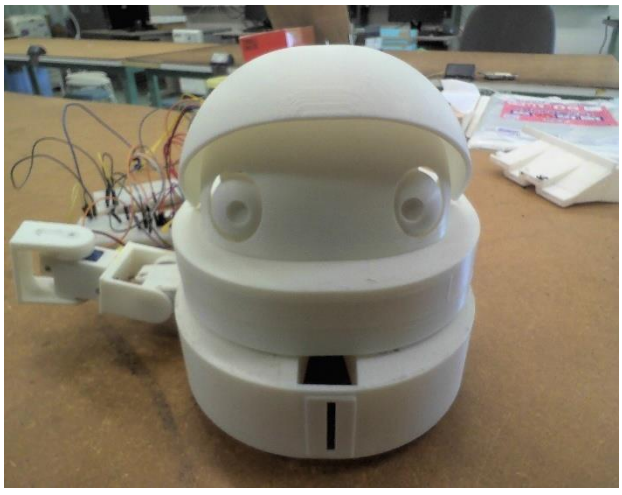

写真８　製作した全体像

４．研究のまとめ

本研究では、３Ｄプリンタと Arduino Nano を用いて、貯金行為を楽しいものにすることを目的とした踊る貯金箱を作製した。硬貨選別部では 10 円、50 円、100 円、500 円の判別を可能とし、投入金額を正確に検出できる構造を実現した。また、頭部や腕にサーボモータを搭載することで、目・頭・腕が連動して動き、貯金額に応じて動作が変化する仕組みを構築した。

プログラム面では、ポーリング制御やデバウンス処理によりセンサを安定して制御し、PWM 制御によってサーボモータを滑らかに動作させた。さらに、投入金額に応じて動作を段階的に変化させることで、利用者の貯金意欲を高める工夫を行った。一方で、硬貨選別精度の向上や動作パターンの多様化などの課題も残った。

本研究を通して、機構設計、電子回路、プログラミングを組み合わせたシステム開発の重要性と、ものづくりの難しさおよび面白さを学ぶことができた。

５．参考文献

サーボモータ勉強
https://curiouser.sakura.ne.jp/lutamesta/doku.php/gimmickkouza/electronic_basic/8/2_pca9685

目の参考
https://willcogley.notion.site/