

情報科学実験B 課題4レポート

氏名 山久保孝亮
所属 大阪大学基礎工学部情報科学科ソフトウェア科学コース
メールアドレス u327468b@ecs.osaka-u.ac.jp
学籍番号 09B22084
提出日 2024 年 7 月 24 日
担当教員 松本真佑, 小南大智

1 システムの仕様

今回私が作成したのはブラウザ観賞用ジョイスティックマウスである。このマウスは、二つのジョイスティックを用いて実際のマウスの機能及びネットサーフィンをするうえで便利な機能を追加したものである。以下にその機能を記述する。

1.1 左手に持つジョイスティックの仕様

1. ジョイスティックを傾けることでマウスカーソルを移動させる。
2. ジョイスティックを押し込むことで実際のマウスの右クリックと同じ結果になる。

1.2 右手に持つジョイスティックの仕様

1. ジョイスティックを上下に傾けることでホイール操作ができる。
2. ジョイスティックを左に傾けることでブラウザ上での”戻る”ボタンを, 右に傾けると”進む”ボタンを押したときと同じ結果となるようにした。
3. ジョイスティックを押し込むことで実際のマウスの右クリックと同じ結果になる。

2 ハードウェア回路

今回私が作成したハードウェア回路は以下の図 1 のようになる。[1][6]

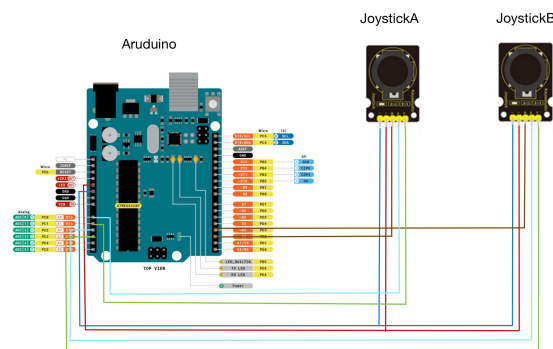


図 1: ハードウェア回路

ジョイスティックのピンは左から順番にグラウンド, 5V, スイッチのピン, ジョイスティックの横方向の傾け度合い, ジョイスティックの縦方向の傾け度合いを表すピンをそれぞれ接続する。今回は左手に持つジョイスティック A のスイッチのピンは D2 に, 右手に持つジョイスティック B のスイッチのピンは D3 に設定した。また, ジョイスティック A の傾け度合いは横方向と縦方向でそれぞれアナログピン A0 と A1 に, ジョイスティック B に関してはアナログピン A2 と A3 に設定した。

ここで, ジョイスティックの仕組みについて記述する。ジョイスティックは以下の図 2 のように, X 軸, Y 軸に対応した可変抵抗が存在する。[7]

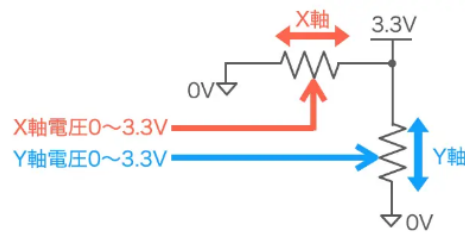


図 2: ジョイスティックの構造

これによりそれぞれの成分について 0V から 3.3V までの値が出力され、スティックの操作位置を読み取ることができるようになる。

3 制御プログラム

今回のジョイスティックマウスを実現するに当たって、Arduino Uno では Mouse ライブラリを使用できなかったため一旦シリアル通信でノート PC にジョイスティックから取得したデータを送信し、ノート PC 内で Python のプログラムを実行してマウスに関する処理を実行した.[2] したがって、今回のレポートの制御プログラムの説明ではジョイスティックから情報を取得しノート PC に送信するプログラム (Arduino のプログラム) と、送られてきたデータからマウスを操作するプログラム (Python のプログラム) に分けることができる。

3.1 Arduino のプログラム

ここでは Arduino のプログラムについて記述する。プログラムのフローチャートは以下になる.[2] このフローチャートに基づいてプログラムを説明する。まず、今回使用したピンの情報は以下になる。

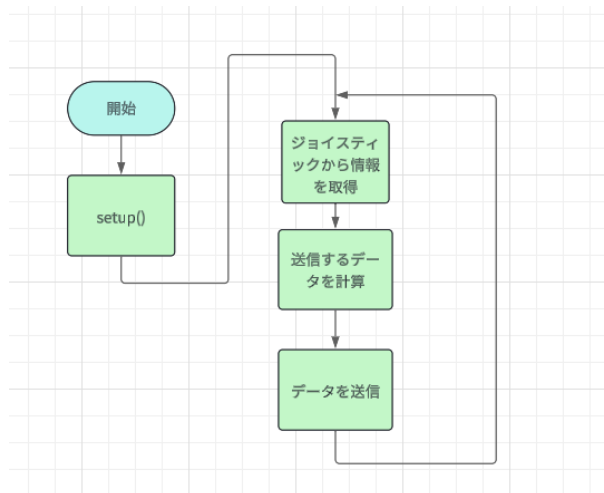


図 3: Arduino のプログラムのフローチャート

```

1 const int xPinA = A0;
2 const int yPinA = A1;
3 const int xPinB = A2;

```

```
4 const int yPinB = A3;
5 const int buttonPinA = 2;
6 const int buttonPinB = 3;
```

A0 と A1 はそれぞれ左手側のジョイスティックの x 軸,y 軸のピン,A2 と A3 は右手側のジョイスティックの x 軸,y 軸のピンを表す. buttonPinA は左手側のジョイスティックを押し込んだ時のピンを,buttonPinB は右手側のジョイスティックを押し込んだ時のピンを表す.

3.1.1 setup() の処理

setup() ではシリアル通信を開始し,それぞれのピンモードを指定する.

3.1.2 ジョイスティックから情報を取得

これ以降の記述は loop() 内の処理である. ここではまずジョイスティックから情報を取得する. そのためのコードは以下のようになる.

```
1 int xValueA = analogRead(xPinA);
```

analogRead() は引数にピン番号を入れると返回值として 0 から 1023 までの整数値を返す. これを xPinB,yPinA ,yPinB にも使用してジョイスティックから情報を取得し,xValueB,yValueA,yValueB に格納する. また,digitalRead() を使用してボタンの状況も情報を取得する.

3.1.3 送信する情報を計算

次に,送信する前の段階である程度データを計算しておく. ここで計算するデータは以下の通りである.

```
1 int A_X_Vector = 0, A_Y_Vector = 0, B_X_Vector = 0, B_Y_Vector = 0;
```

A を含む変数はマウスカーソル,B を含む変数はホイールや進む,戻るボタン等に対応している. ここで 0 に初期化をしているのは,ジョイスティックのわずかな変化によってマウスカーソル及びホイールが変化しないようにするためである. 詳細はのちに記述する. 次に,以下のような条件分岐を上で初期化した 4 つの変数に対して行う.

```
1 if (xValueA >= 550) {
2     A_X_Vector = map(xValueA, 550, 1023, 0, sensitivity);
3 } else if (xValueA <= 450) {
4     A_X_Vector = map(xValueA, 450, 0, 0, -sensitivity);
5 }
```

これは,xValue の値が 450 から 550 までの間はどちらの条件分岐にも入らないようになっている.450 と 550 を境界値にした理由としては,analogRead() による結果が 0 から 1024 であり,中央値が約 500 であることからそこからそれぞれ 50 ずつをジョイスティックの変化として考えないようにしたためである. このときは A_X_Vector の値が初期値の 0 のままとなる. これによってジョイスティックのわずかな変化には反応しないようになっている. xValue の値が 550 以上の場合は map() を使用して A_X_Vector の値を変更している.map() は第二引数から第三引数までしかとらない第一引数の値を,第四引数から第五引数の間の値に変更する関数である.[4]

また,sensitivity は変更後の最大値であるから,この値を変更することでマウスカーソル等の感度を変更することができる. デフォルトでは 10 とした.

3.1.4 データの送信

今回のプログラムでは以下のようにデータを”,”で区切って送信した.

```
1 Serial.print(A_X_Vector);  
2 Serial.print(",");  
3 Serial.print(A_Y_Vector);  
4 Serial.print(",");  
5 Serial.print(B_X_Vector);  
6 Serial.print(",");  
7 Serial.print(B_Y_Vector);  
8 Serial.print(",");  
9 Serial.print(buttonStateA == LOW ? 1 : 0);  
10 Serial.print(",");  
11 Serial.println(buttonStateB == LOW ? 1 : 0);
```

9 行目と 11 行目の buttonState に関しては, それぞれボタンが押されていれば 0 を, ボタンが押されていないならば 1 を送信する.

3.2 Python のプログラム

ここでは Python のプログラムについて記述する. プログラムのフローチャートは以下ようになる. こ

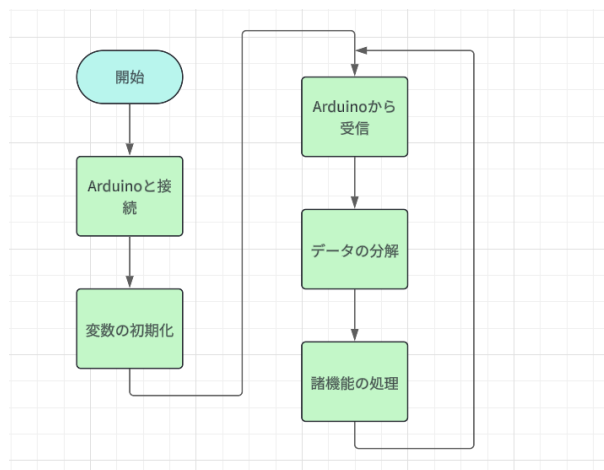


図 4: Python のプログラムのフローチャート

のフローチャートに基づいてプログラムを説明する.

3.2.1 Arduino と接続

以下のようなプログラムで Arduino と PC を接続した.

```
1 ArduinoSerial = serial.Serial('COM3', 9600)  
2 time.sleep(1)
```

2 行目は Arduino との接続を安定させるためである.

3.2.2 変数の初期化

次に変数を初期化する.

```
1 back_interval = 1
2 last_back_time = 0
3 forward_interval = 1
4 last_forward_time = 0
5 previous_right_button = 1
6 previous_left_button = 1
```

1 から 3 行目は”戻る”機能を,4 から 6 行目は”進む”機能を実現するための変数であり,フローチャートの繰り返しが始まる前に初期化している.

3.2.3 Arduino から受信

フローチャートのように, 以下で記述する内容は無限ループの中の処理である. 接続された Arduino から送られてきたデータを取り出す操作は以下のプログラムで実現される.

```
1 data = ArduinoSerial.readline().decode('utf-8').rstrip()
2 values = data.split(",")
3 if len(values) != 6:
4     print(f"Unexpected data format: {data}")
5     continue
```

まず data に送られてきた UTF-8 の文字列から改行文字を取り除いたものを格納する. その data は Arduino のプログラムで記述したように,”,”で区切られているので 2 行目の split() を使って分け, リストとなり values に格納される. リスト values の長さが想定している 6 でなければ, 条件分岐させエラーメッセージとともに繰り返しの最初まで戻す. この処理の目的は, 期待された受信データとは異なる形式として受け取ったデータを使用してマウスの処理をしてしまうことを避けるためである. 期待されたデータの形式であれば,continue 以下のプログラムが処理される.

3.2.4 データの分解

データの分解は以下のプログラムによって行われる.

```
1 left_x_vector, left_y_vector ,right_x_vector ,right_y_vector ,left_button ,right_button
  = map(int , data.split(","))
```

map() を使って各要素を int 型に変更しそれぞれの変数に格納する. それぞれの変数については 3.2.5 の諸機能の処理にて詳細を記述する.

3.2.5 諸機能の処理

ここでは今回実現したマウス操作に関する様々な機能の実現方法を記述する.
まずはマウスカーソルの操作について記述する.

```
1 (mouse_x,mouse_y) = mouse.get_position()
2 mouse.move(mouse_x+left_x_vector,mouse_y-left_y_vector)
```

上のプログラムの一行目では現在のマウスカーソルの x 座標と y 座標を `get_position()` を使用して取得している。そしてそれらに 3.2.4 で取得した左手に持つジョイスティックから得られる、どれだけマウスカーソルを移動させるかを表す `left_x_vector` と `right_y_vector` を合わせたものを引数として `move()` を使って移動させる。ここで縦軸に関して加算ではなく引き算をした理由は、縦軸の正の方向が画面の下側であるためである。[5]

次に記述するのはそれぞれのジョイスティックを押し込んだ時の機能である。

```
1 if right_button == 0 and previous_right_button == 1:
2     mouse.click('left')
3 if left_button == 0 and previous_left_button == 1:
4     mouse.click('right')
5 -----
6 previous_right_button = right_button
7 previous_left_button = left_button
```

1 行目と 3 行目の条件文は、それぞれのジョイスティックが押し込まれたタイミングかどうかを判定している。それぞれが押し込まれた瞬間に `mouse.click()` を使って左クリックと右クリックの機能を実現している。右手のジョイスティックを押したときに左クリック、左手のジョイスティックを押したときに右クリックのように、実際のマウスと左右を反転させた理由としてはジョイスティックを操作しながら押し込むという操作が難しいと感じたためである。また、点線以下の 6,7 行目は繰り返しの最後に現在のボタンの状態を記憶するものであり、条件分岐の際に使用される。

次にホイール操作について記述する。

```
1 if right_y_vector != 0:
2     wheel_speed = np.interp(right_y_vector, (-10,10), (-1,1))
3     mouse.wheel(wheel_speed)
```

右手側のジョイスティックの縦方向の傾け度合いを格納している `right_y_vector` が 0 でなければ、`np.interp()` を使って -1 から 1 の値になるように値を変更しホイール操作の速度を計算する。そして `mouse.wheel()` を使用してホイール操作を行う。`mouse.wheel()` の引数には -1 から 1 の値までしか取れないために速度を計算した。

最後に”戻る”機能と”進む”機能の処理について記述する。

```
1 if right_x_vector < -5:
2     current_back_time = time.time()
3     if current_back_time - last_back_time >= back_interval:
4         pyautogui.hotkey('alt', 'left')
5         last_back_time = current_back_time
6 if right_x_vector > 5:
7     current_forward_time = time.time()
8     if current_forward_time - last_forward_time >= forward_interval:
9         pyautogui.hotkey('alt', 'right')
10        last_forward_time = current_forward_time
```

右手側のジョイスティックの横方向の方向け度合いを格納している `right_x_vector` の値は Arduino のプログラムで示したように、`-sensitivity` から `sensitivity` の値まで取ることができる。今回は `sensitivity` が 10 であるときの処理である。戻る機能や進む機能は少しの変化で反応してしまうと間違えてその操作をしてしまう可能性がある。したがって、`sensitivity` の半分の値になるまでジョイスティックを傾けないと処理されないように設定した。

4 考察や工夫点

今回の課題を通して私が考察したのはマウスやトラックボール等ではなくジョイスティックを使用することの是非についてである。

今回のジョイスティックマウスが先ほど挙げたマウスなどと異なる点は机などの平面に接地していなくてもよいという点である。トラックボールマウス等はどの場所に置いても操作することはできるが安定した場所でないとは本来の操作感は得られない。ジョイスティックなら手を机からおろしたような場所であっても操作することが可能である。今回作成したものは Arduino に接続されているためジョイスティックを動かす範囲が制限されてしまっているが、Bluetooth 等を用いてジョイスティックの情報を離れたところから伝達できるようになれば自由に持ち運べるようにできると考えられる。このジョイスティックマウスの効果的な使用方法としてはパソコンの画面をテレビ等に映して動画や映画を鑑賞する際に使用できる点だと私は考えた。これは、リモコンよりもマウスの操作性に近くなるためである。

しかし、マウスカーソルの操作感などについては改良の余地があると感じたので sensitivity やジョイスティックのどのあたりまで反応しないようにするか等の調整などが課題としてあげられる。

5 感想

今回の課題を通して私が感じたことは自分で考えることの楽しさである。これまでの実験やプログラミングの授業では指定された仕様を満たすような課題を作成してきたが、実験 B の課題 4 に関しては何を作るのが生徒側に委ねられていた。最初は何を作るべきが迷ってしまい時間を持て余していたが一度方向性が固まるとどんな機能を追加すればよいかなどを考えることに楽しみを見出すことができた。4 回生や大学院に進学した後に行う研究も私が何をするかを決めることになるので、今回の課題での経験は非常に役立つものになると感じた。

6 謝辞

4 月から約 4 か月間授業中の質問対応やレポート採点などをしてくださった教授、TA の皆様、本当に有難うございました。この授業で学んだことを活かして、今後の学習及び研究に取り組んでいきたいと思います。

参考文献

- [1] <https://docs.keyestudio.com/projects/KS0349/en/latest/KS0349.html#project-36-joystick> 7/11 アクセス
- [2] <https://elchika.com/article/137e8fc6-cc28-420e-b41d-d75da29f6568/> 7/11 アクセス
- [3] <http://www.musashinodenpa.com/arduino/ref/index.php?f=0&pos=2113> 7/11 アクセス
- [4] <http://www.musashinodenpa.com/arduino/ref/index.php?f=0&pos=2743> 7/11 アクセス
- [5] <https://laboratory.kazuuu.net/using-mouse-to-control-a-mouse-in-python/> 7/16 アクセス
- [6] 情報科学実験 B 指導書
- [7] <https://logikara.blog/joystick/> 7/20 アクセス