

UFO キャッチャーの製作

中山 堪太, 桑原 滝太
小原 璃央, 入谷 健仁

1. 研究概要

今回 UFO キャッチャーを製作するにあたり、私たちは Arduino を使用した。

Arduino とはマイクロコンピュータの一種であり、モータを動かす際の信号を送るなどのことができるもので、制御の際には作ったプログラムのデータを書き込んで、その内容の通りに信号を送ってくれる。

使用した理由は、班員のほとんどが使用したことが無く、挑戦してみたいと考えたためである。

また、プログラムは、スケッチという C 言語に類似している言語を用いて、主にモータの制御を行った。今回は、モータ等の制御をどのように行うのかを学ぶとともに、Arduino への理解を深めることを目的とし、作業を行った。

2. 研究の具体的内容

私たちは今回、ゲームセンターの、ボタン 2 つで操作する UFO キャッチャーをイメージし、それになるべく近づけるよう製作した。

具体的には以下のものを製作しようと考えた。

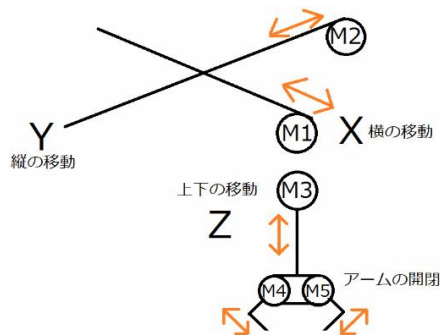


図 1 動作イメージ

・ X,Y,Z 軸(横, 縦, 上下の移動)

ボタンを押すことによってモータを動かす

図のように横, 縦, 上下とクレーンの移動を行う。

・ アーム (景品をつかむ)

アームの下降後、開いているアームが閉じて景品をつかむ。そのまま自動で景品口へと戻りアームを開いて景品を落とす。その後最初から動かすことができる。



図 2 完成品

(1) 今回使用した部品について

(a) マイコンとモータの制御について

Arduino の信号では直接モータを動かすことはできないので、モータドライバとして以下の IC を使用している。

・ IC(TA7291P)について

Arduino から 2 ビットの信号を受けとり、モータの正転、逆転、停止といった制御をする IC である。

モータの制御方法は、HIGH 信号または LOW 信号を入力して動作している。信号の入力による動作を以下の表にまとめた。

入力 1	入力 2	動作
1	0	モータ正転
0	1	モータ逆転
1	1	モータ停止
0	0	モータ停止

↑ 1 が HIGH, 0 が LOW

表 1 信号による動作

(b)サーボモータについて

サーボモータは、普通のモータとは違い、0度から180度まで角度を指定して動かすことができるモータである。今回はそれをアームの開閉に用いている。

サーボモータの角度制御は、PWM(パルス幅変調)方式により行う。PWM方式とは、パルスの周期を一定にしてパルス幅を変えることで目的に合った制御を行うものである。サーボモータはパルスの幅が回転角に対応するようになっており、パルス幅が狭ければ0度中位なら90度、広ければ180度というような動きをする。







	狭い	中位	広い
パルス幅			
サーボモータの回転角			

図3 パルス幅による回転角度



図4 サーボモータ

(c)リミットスイッチについて

図4は動作の停止に使われるスイッチである。今回はクレーンが端まで移動したときのモータの動作停止に用いている。

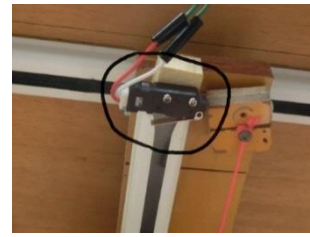


図5 リミットスイッチ

(2)機構について

(a)X軸,Y軸(横と縦の移動)

図6,7は水平面上のクレーン移動(スライド移動)をおこなう部分である。移動をスムーズに行うためカーテンレールを使い、クレーンを取り付ける台座にカーテンランナーを取り付けた。また、カーテンレールの端にリミットスイッチを取り付けている。

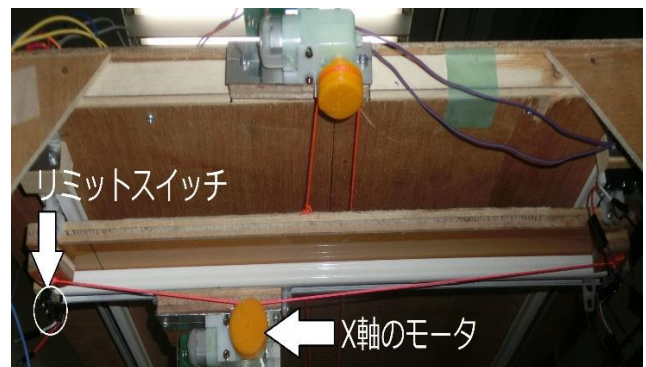


図6-1 X軸の機構1

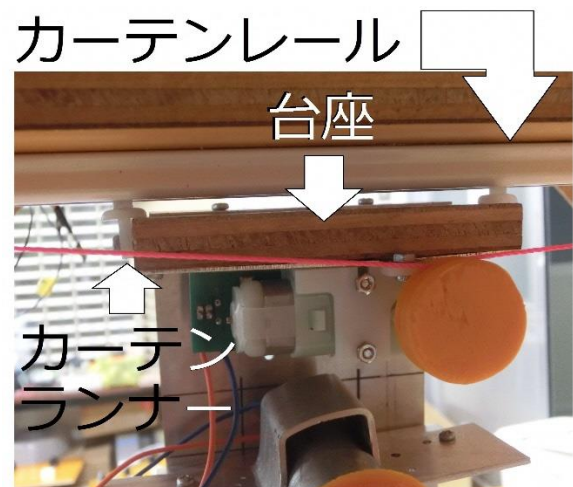


図6-2 X軸の機構2

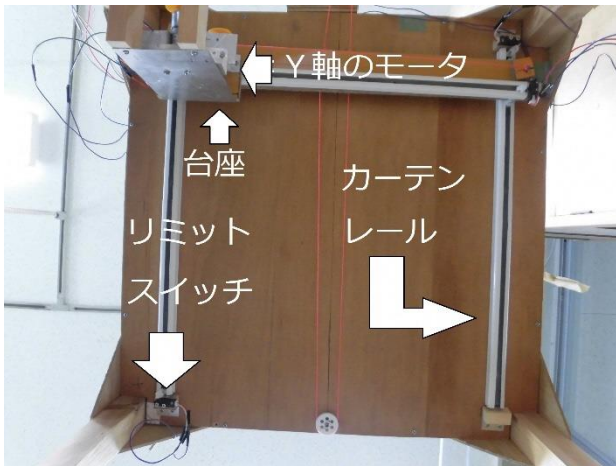


図7 Y軸の機構

(b)Z 軸(上下の移動)

パイプのスライドにより，それに固定しているアーム部分が上下する。パイプを1本ではなく2本にした理由は動作する際にクレーンを安定させるためである。

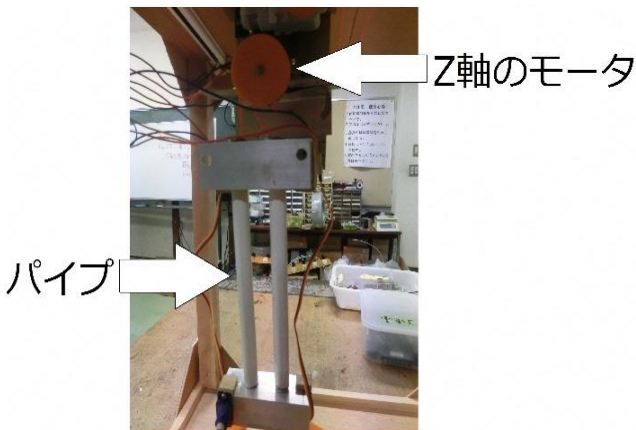


図8 Z軸の機構

(c)アーム(景品を掴む)

アームは3Dプリンタで製作した。サーボモータの回転する部分に取り付けており，アームが開閉できるようになっている。また，アームの開閉の動作がかなり急なため緩衝材としてツメの先にスポンジを取り付けている。

サーボモータ

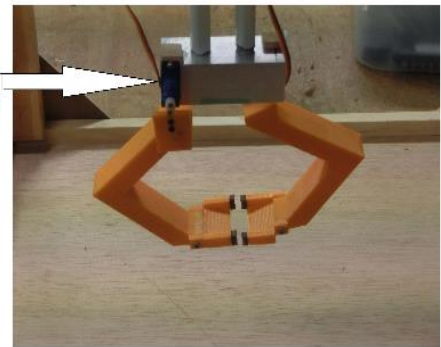


図9 アームの機構

(3)制御プログラムについて

(a)サーボモータの制御

```
void setup(){
  myServo.attach(A2); //myServoをA2番ピンに設定
  myServo.write(110); //myServoを110度に
  myServo2.attach(A3); //myServoをA3番ピンに設定
  myServo2.write(110); //myServoを110度に
```

図10 サーボモータの制御プログラム

図10はサーボモータ制御の一部を抜き出したものである。信号を出すピンや，アームの開閉角度を設定している。

(b)タイマの制御

```
MsTimer2::set(100,flash); //msタイマを0.1秒単位で数えるよう設定
...
if (digitalRead(7) == LOW && f == 3){ // fが3の状態、ボタン2が押されているとき
  MsTimer2::start(); // msタイマ開始
  digitalWrite(11,HIGH); // Yのモータが正転
  digitalWrite(12,LOW); //
  ms = 0; //msタイマをリセット
  f = 5; // fに5を代入
}
if (f == 5){ // fが5のとき
  if (ms >= 10){ //1秒カウントしてから
    digitalWrite(3,HIGH); // Zのモータが正転
    digitalWrite(4,LOW); //
    f = 6; // fに6を代入
  }
}
}
```

図11 タイマの制御プログラム

図11はタイマの制御，フラグによる動作の順序管理の一部を抜き出したものである。

今回本物の UFO キャッチャーに近づけるため、タイマを使用して動作の間にインターバルを置いた。タイマは 0.1 秒単位でカウントするように設定した。インターバルを置くことで次の動作へ急に移行することがなく、一定の余裕をもって気持ちよく遊ぶことができると考え、搭載した。

また、動作ごとに順番をつけるため、軸の移動やアームの開閉、ホームに戻ったときのリセット処理といった動作が前の過程に戻ることが無いよう、フラグ管理という方法を用い、動作を部分ごとに分けた。

3. 研究のまとめ

この研究で私たちは、班員のほとんどが使用したことのない Arduino を使ってモーターやリミットスイッチなどの制御を学び、0 からものを作る大変さを知ることができた。研究を始めたばかりの頃は、筐体の詳細な内容を決められず、なかなか製作できなかった。そのため、プログラムや配線といった工程にあてる時間が少なくなり、岡工祭の展示に間に合わせるができなかった。中でも特に、アームの開閉の際に用いるサーボモーターの角度の調整に苦労した。その理由は、アームをサーボモーターに取り付ける際、手作業だったことや Arduino の知識が不足していたことにある。当初アームの開閉具合がうまく合わないのは、手作業でアームを取り付けることでズレが生じるからだと考えていたが、本当は UFO キャッチャーを動かすための電気信号の全てが Arduino を介していたことが原因だった。つまり、Arduino 内の電力を、全ての処理に回していたため、サーボモーターへ供給される電力が足りず、サーボモーター自体の動きが不安定だったということだ。その対策として、外部電源をもう一つ用意し、Arduino を介さず直接サーボモーターへと引っ張ってくることで供給される電

力を補った。しかし、それらの反省面はあるものの、今回の製作でスケジューリングの大切さ、大変さを深く知ることができた。また、今回製作した UFO キャッチャーを使用した際、ゲームセンターに置いてある筐体と比べると、あらゆる面で物足りない部分は感じたが、ほとんど 0 の知識から完成したことに対して、班員は、なんとか間に合ったという安堵と達成感で溢れており、この課題に挑戦してよかったと思うとともに、進学した後、今回学んだ技術や知識を生かしていきたいと思った。

参考文献

- ・これ 1 冊でできる！Arduino ではじめる 電子工作 超入門 改訂第 3 版
- ・実践 Arduino! 電子工作でアイデアを形にしよう
- ・ゼロからよくわかる! Arduino で電子工作 入門ガイド