

自作クレーンゲームの制作

遠藤 翔世 河端 亮

1. 研究概要

1年生のとき、マイコン研究同好会の岡工祭の展示作成で一度クレーンゲームを作成した。技術力や経験値が不足していたため、いろいろな問題があった。技術力や経験値を得た3年生で完璧なクレーンゲームを作成することを決めた。

2. 研究の具体的内容

誰が見ても「クレーンゲームじゃん」って思えるものを作れるようにゲームセンターのUFO キャッチャーをイメージしたり、実際にゲームセンターに通ったりして、できるだけ本物に近づけるように作成した。

(1) 材料について

- Arduino Mega 2560 x 1
- ユニバーサル基盤 x 3
- アーム (amazon 購入)
- DC モーター x 3
- モータードライバ x 2
- 電線ケーブル x 沢山
- 押しボタンスイッチ x 3
- フォトインタラプタ x 4
- セラミックコンデンサ x 9
($0.1\mu\text{F}$ x 9)
- 抵抗 x 8
(330Ω x 4, $10\text{K}\Omega$ x 4)
- アクリル板 x 沢山
- スイッチング電源 x 1
(5V/2A 12V/10A 24V/1.5A)
- アルミフレーム
(600mm x 14本, 45mm x 1本)
- V溝ベアリング x 10

- DIN レール
(600mm x 2)
- タイミングベルト
(1.2m 以上 x 2)
- はんだ 適量
- メッキ線 適量
- 熱縮チューブ 適量
- 電子基板ピンヘッダ 適量
- ダイオード x 1
- トランジスタ x 1
- リレー x 1
- ねじ x 沢山
- ナット x 沢山
- ワッシャ x 沢山
- スプリングワッシャ x 沢山
- スペーサー x 沢山
- アルミ板 x 沢山
- 結束バンド x 沢山

(2) 使用ソフトについて

今回、使い方や言語学習の時間を短縮するために、すべて情報技術科の実習で勉強したソフトウェアを使うことにした。

- Arduino IDE (2年生)
Arduino IDE はC言語風の「Arduino 言語」によってプログラムを制作・コンパイル・デバッグ等し、それを Arduino ボードに転送するための「統合開発環境」と呼ばれる、PC 上で作動させる一種のソフトウェアである。
- Fusion 360 (3年生)
Fusion 360 は、オートデスク社が開発しているクラウドベースの3D CAD、CAM、CAE ソフトウェアである。3D モデル、プリント基板、2D・3D 図面の設計が可能で

ある。Windows と macOS で利用可能で、Android と iOS 向けの簡易アプリケーションも配布されている。ソフトウェアのサブスクリプションライセンスは有償だが、学生・教員・教育機関は無償で利用可能である。また、非商用の個人用途であれば機能限定版ライセンスが無料で使用可能である。

- CorelDRAW (3 年生)
CorelDRAW は、コーレル株式会社が開発、販売するベクトル画像編集ソフトウェアである。
- 123d Design (3 年生)
AUTODESK 社から提供されている無償の 3DCAD である。
- xyzware pro (3 年生)
XYZware は 3D のオブジェクトをカスタマイズ、造型するために XYZprinting が開発した専用のアプリケーションです。stl 形式のファイルをインポートし、オブジェクトの完成図を自動でシミュレーションすることができるものである。

(3) 年間計画について

2 年生

3 月 材料準備・全体を考える

3 年生

4 月 枠組みを作る

5 月 枠組みを作る

6 月 アームの動く部分を作る

7 月 アームの巻き上げる部分を作る

8 月 電子回路をする

9 月 電子回路をする

10 月 プログラムをする

11 月 遊んで岡工祭展示

12 月 最終調整・報告書作成

1 月 発表会

(4) 枠組み作成について

枠組みはアルミフレームを使って作った。大きさは 600mm x 640mm x 600mm である。アルミフレームを使うのは初めてでアルミフレームを接続するにはブラケットというものを使うのが基本的な接続方法である。だが調査不足により買うのを忘れていた。仕方がないのでアルミ板を使用して自作した。まずアルミフレームのデータシートの採寸を見ながら自作ブラケットの設計をした。



図 1



図 2

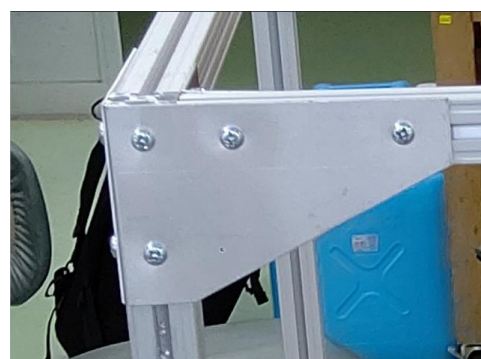


図 3

次にこの三種類(図1、図2、図3)のブラケットを作成した。作成時にちゃんと寸法を見ながら手作業で印をつけていたので、ズレが発生した。まず枠組みの箱の部分を作った。



図4

次に箱の中身を作った。



図5

この時点でずれていたのが、ねじがうまく入らなかったり、アルミフレーム同士できれいはまらなかったりした。

最後に箱(図4)と中身(図5)を接続した。



図6

ズレが大きすぎてこの二つが全然はまらなかった。ここで最新の注意を払って作業していれば効率よく進めることができていた。

(5) アーム動作部分作成について

アームの動きの部分はDINレールを使って作成した。DINレールにレーザー加工機で加工した亚克力板の部品を取り付けてそこにモーターを取り付けた。モーターにはタイミングプーリー(図7)を付けた。



図7

タイミングベルトを使用してDINレールの上の車(図8)を動かす。この車もレーザー加工機を使用して作成した。

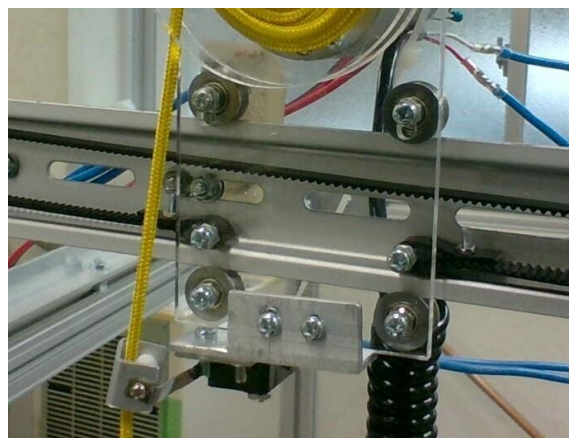


図8

図9はDINレールの全体である。これを二つ作り、車の一つはDINレールを接続して、X軸移動とY軸移動ができるようになった。



図9

まだ何もくっつけていない車にはアームを取り付けた。ここにはアームが衝突したか判断するための構造(図 10)を取り付けた。この構造はアームが重いのでその性質を利用してアームが空中にある場合は常にリミットスイッチが押されている状態になっている。一番下に当たるか景品に当たるかをするとアームを釣っている紐が緩んでリミットスイッチが押されない状態になる。これをプログラムで検出してアームの降下上限をしているのである。



図 10

(6) 電子回路について

最初に、アームの開閉のための回路に取り掛かった。アームの開閉には 24V を使用している。Arduino では 24V という大きな電圧をかけてしまうと壊れてしまうので、リレーを使用して制御した。

リレーを Arduino で制御しようとするとう電流が少なすぎてリレーが全く動かないので、増幅回路を使用して電流を大きくして制御することにした。ここで頭の片隅に置いてあった電子回路の授業を思い出しながら電流の計算であったり、抵抗値がどのくらい必要なのか考えたりした。回路図は、draw.io というサイトを使用して作成した。実際にできたものは、(図 11) である。

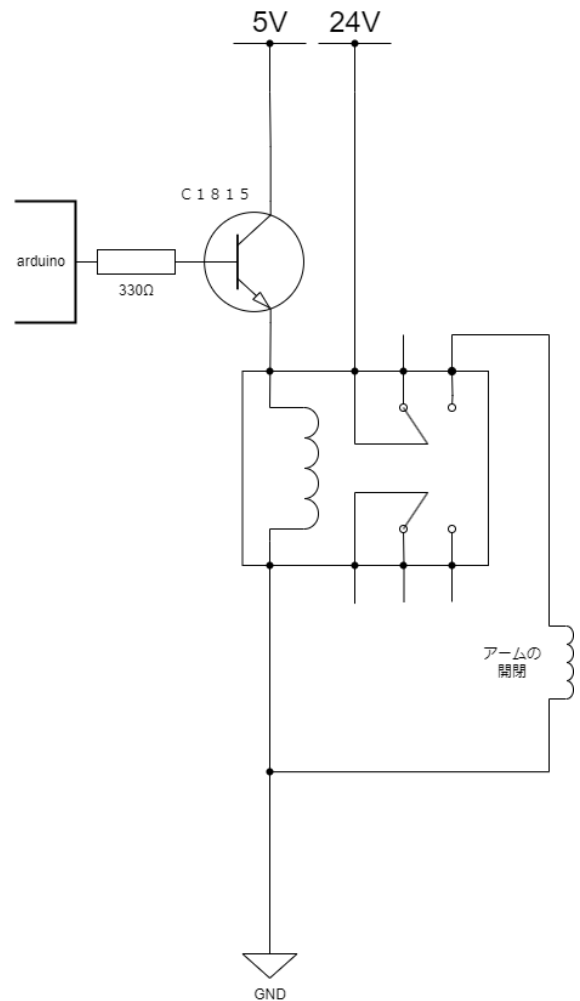


図 11

実際に取り掛かると色がよく見えないので、抵抗のカラーコードがよく見えていなくて、330Ωだと思っていたものが 33Ωだったので気づくまで「なんで動かないのだろう」と考えていた。ここで電子回路をするときは、きちんとテスターを使って抵抗値を測ったほうがいいと思った。これを見ている人も電子回路をするときはほんとに気を付けてほしい！実際に回路を組んだものが(図 12) である。

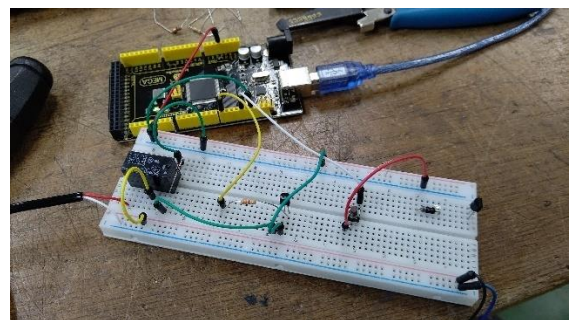


図 12

次にアームの移動制限に必要なセンサーの回路に取り掛かった。今回はフォトインタラプタというセンサーをスイッチとして利用して使用した。これも drawio というサイトを使用して回路図を設計した。実際の回路図は (図 13) である。

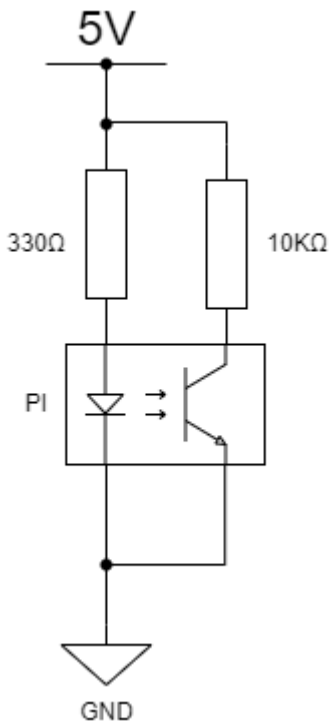


図 13

実際に回路を組んだものが図 14 である。

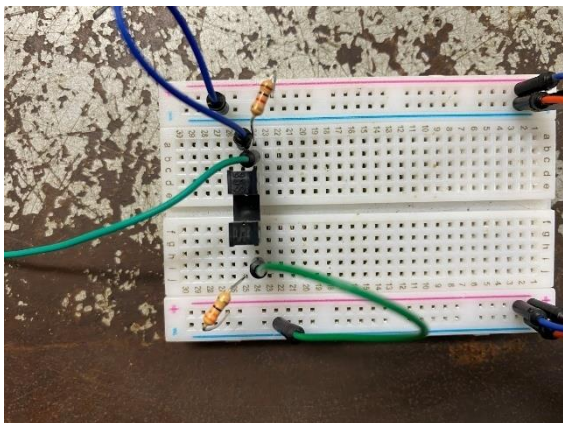


図 14

このセンサーを使ってどうやってアームの移動制限として使用するかというと、DIN レールの上を動く車のねじをあえて長くすることによってフォトインタラプタの間をねじが

通り、スイッチとして使用できると考えた。実際に取り付けた時の写真が (図 15) である。上手にフォトインタラプタの間を通るように調整するのが大変だったが、これのおかげで最終手段としてリミットスイッチも考えていたが、使わずに済んだ。



図 15

次に Arduino ではモーターが直接制御できないのでモータードライバを使用しないといけないのである。工作室に転がっていた「TA8050」を使って回路を考えた。

まず「TA8050」について

端子説明

端子番号 記号：端子の説明

- 1 DI1 : 出力の状態を制御する端子。
- 2 DI2 PNP タイプの電圧コンパレータを内蔵している
- 3 M(+): DC モーターがつながる端子で Sink、Source とも 1.5A の電流容量を持ちます。また、モーターの逆起電圧吸収用のダイオードを VCC 側と GND 側に内蔵している
- 4 GND: 接地端子
- 5 M(-): 3 Pin との間にモーターがつながる端子で、3 Pin と同等の機能を持ち、1、2 Pin により制御される
- 6 NC: 非接続端子
- 7 Vcc: 電源端子

真理値表

入力		出力	
DI1	DI2	M(+)	M(-)
H	H	L	L
L	H	L	H
H	L	H	L
L	L	OFF	

図 16

実際に設計した回路図が(図 17)である。

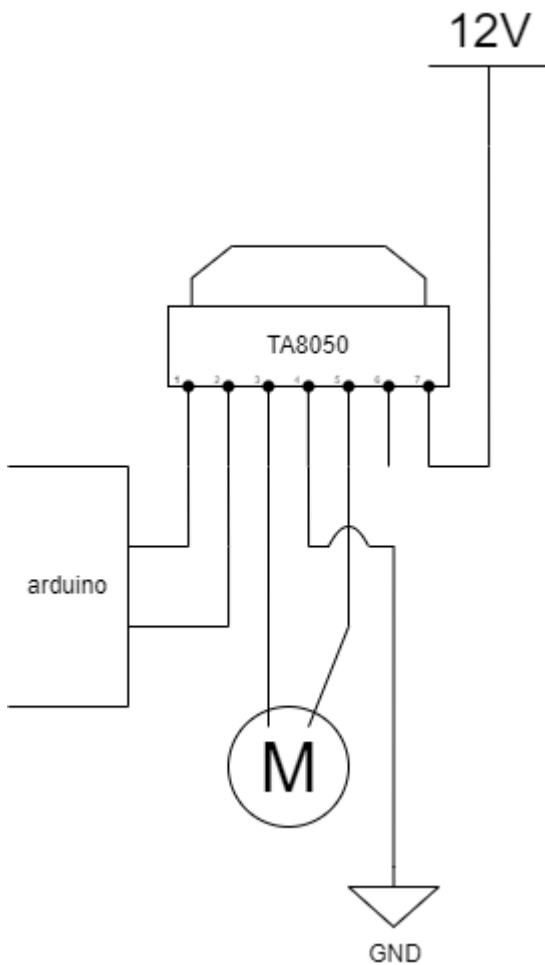


図 17

回路を作ったものが(図 18)である。

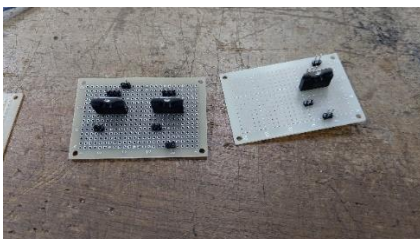


図 18

モータードライバが作成できたので実際に制御してみたところうまく制御ができなかった。原因を探ってみたところ、今回使用している DC モーターの起動電圧がモータードライバの最大定格を超えてしまっていた。(最初からちゃんと調べていればよかった。)仕方がないので「L298N」を使用した(図 19)。これは中華製で IC がよく熱を持つので使いたくはなかったがクレーンゲームはあまり複雑な動作がないので使うことにした。



図 19

これを使うとうまく制御することができた。案の定少し IC が熱を持ってしまっていたが問題がなかったので「L298N」を使うことにした。

最後にクレーンゲームのすべての配線を行った。クレーンゲームの大きさが大きいので大量のビニール線が必要になった。新しくたくさん線を出すとすごくもったいない気がした。1年生の実習で行う、リレーの実習で使うビニール線をゴミ箱からたくさん取ってきて、できるだけ無駄な線をなくそうと考えた。線と線同士を、熱収縮チューブを使いながら長い線をたくさん作成した。電子回路をするには、少し太かったが頑張って配線した。配線している時の写真が図 20 である。

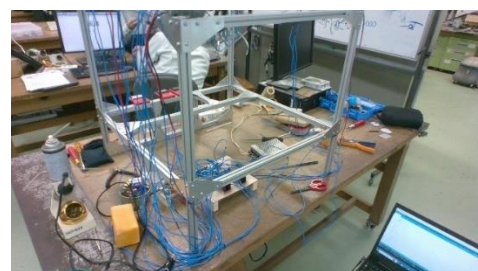


図 20

次が電源周りの配線の写真が(図 21)である。

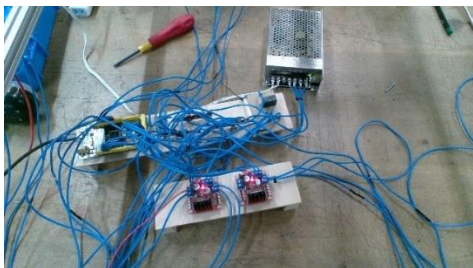


図 21

この時とても気を付けながら配線をしてたつもりだが、フォトインタラプタの配線をしているさい本来 GND に配線しないといけないところを 5V に配線してしまった。それに気づかず電源を入れてしまった。この時 Arduino を 2 枚壊して制御のために使っていた PC を危うく壊してしまうところだった。

配線ミスをしてしまった原因は(図 21)のような汚い配線をしてしまったことと回路の導通チェックをしていなかったことである。配線を終わったらテスターを使って導通チェックを行い、きちんと自分が思っているように配線ができているか確認をしたほうがいいのときれいに配線をしてどの線がどこに配線されているかわかりやすくしたほうがいいと思った。

(7) Arduino Mega のピン接続について

プログラムをする際に自分なりの名称を決めてプログラムをすることにした。

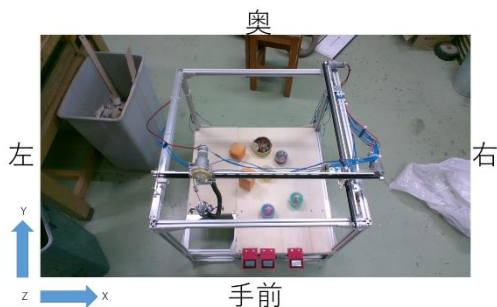
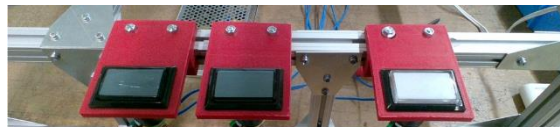


図 22

(図 23)のように左や右、奥、手前という風

に名称を付けた。以後のピンの入出力の説明では、右、奥と表現する。次にボタンの名称はこのような感じである。



ボタン① ボタン② ボタン③

図 23

次に Arduino mega のピン接続は(図 24)のような感じである。

4	アーム奥側センサ
5	アーム手前センサ
6	アーム右側センサ
7	アーム左側センサ
8	ボタン①
9	ボタン②
10	ボタン③
11	アーム降下上限
12	アーム開閉
42	モーター右移動
43	モーター左移動
44	モーター奥移動
45	モーター手前移動
46	アーム下降
47	アーム上昇

図 24

(8) プログラムについて

プログラムでは、いきなりすべての動きをプログラムで書くのは難しいと思い、動作ごとに制御することができるようにプログラムを書くことにした。

最初に、ボタンを押すとモーターが回転するとかボタンを押すとアームの開閉が行われるような簡単な動きのプログラムを作成した。動作ごとに制御できるようになってから、本格的にクレーンゲームの動きに沿ってプログラムを書くことにした。switch 文をうまく活用して、動作ごとに分岐しながらプログラムを書いた。クレーンゲームのフローチャートは(図 25)である。

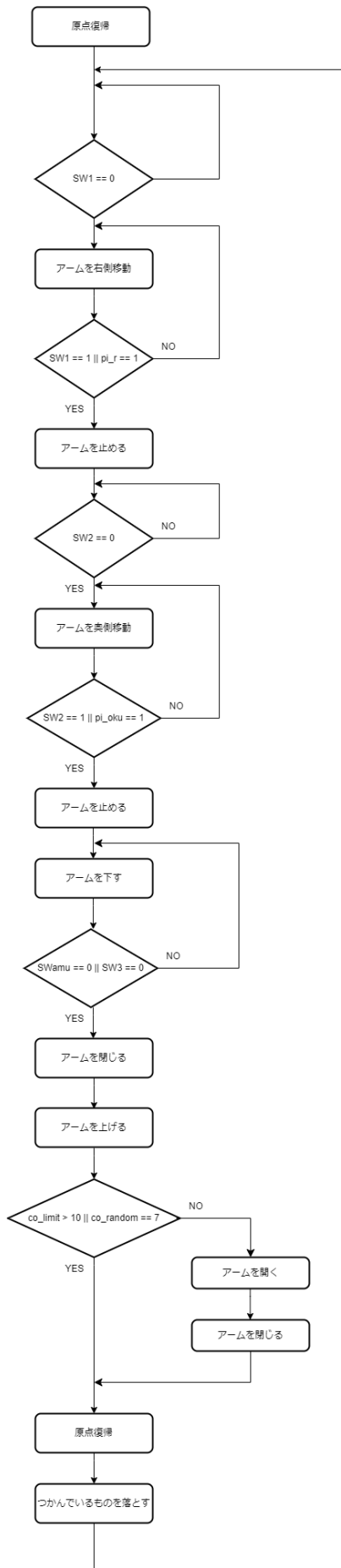


図 25

フローチャートのことを考えながら、プログラムを書いていくことにした。

<各動作のプログラムの説明 (抜粋) >

```

110 case 1: //右側に移動する
111 if ( sw1 == 0 )
112 {
113   if ( pi_r == 1 ) nextf = 1;
114   swlf = 1;
115   migi();
116 }
117 if ( sw1 == 1 && swlf == 1 )
118     nextf = 1;
119 if ( nextf == 1 )
120 {
121   mode = 2;
122   nextf = 0;
123   swlf = 0;
124   tomeru();
125 }
  
```

このプログラムは、右移動のプログラムである。111 行目でボタン①が押されるとアームが右側に移動する。117 行目で押されているボタン①が離されるか、113 行目で移動上限(フォトインタラプタの右)が反応したらアームを止める。右と同じように奥側移動もプログラムを書いた。

```

167 case 3: //アーム降下
168 if ( swamu == 1 || sw3 == 0 )
169     swamuf = 1;
170 if ( swamuf == 0 )
171 {
172   co_ue++;
173   orosu();
174 }
175 else
176 {
177   tomeru();
178   delay(500);
179 }
  
```



```

178  toziru();
179  mode = 4;
180  swamuf = 0;
181  delay(500);
182 }
183 break;
184 case 4: //アーム上昇
185  ageru();
186  if ( co_ue * 1.1 < co )
187  {
188    tomeru();
189    mode = 5;
190    co = 0;
191    co_ue = 0;
192  }
193 co++;
194 break;

```

このプログラムは、アームの降下と上昇のプログラムである。169 行目でアームの降下上限としているリミットスイッチが反応するかボタン③が押されるまでアームを下におろしている。このときアームをおろしている時間をカウントしている。168 行目でリミットスイッチが反応するか。ボタン③が押されたらアームを止めるようにしている。アームを止めたらアームを閉じている。0.5 秒待つてから、アームを持ち上げている。186 行目でアームを下すときにカウントした時間を使ってアームを上昇している。if 文の中でカウントした時間と 1.1 倍しているのは、アームが重いのでモーターが動き出すまでの時間を考慮している。

```

3 #define UPLIMIT 10 //上限
4 #define RANDOM 10 //確率

197 case 5: //確率判定
198 //確率を検出
199 co_random = random(RANDOM);

```

```

200 //確率 or 上限が来たら次へ
201 if ( co_random == 7 || co_limit ==
        UPLIMIT )
202 {
203   co_limit = 0;
204   mode = 6;
205 }
206 //確率 or 上限ではない場合は景品を落
        として次へ
207 else
208 {
209   delay(200);
210   hiraku();
211   delay(200);
212   toziru();
213   mode = 6;
214   co_limit++;
215 }
216 break;

```

このプログラムは、確立機としてクレーンゲームを動かすためのプログラムである。確立を検出するために Random 関数を使用した。このランダム関数とは、指定した数値の間で乱数を習得できる。199 行目で乱数を受け取ります。201 行目で乱数を受け取った値が 7 か回数上限の 10 回目があるとアームはなんにもせずに次の動作に行きます。乱数が 7 以外の値か回数上限ではない場合は、一度アームを開いて景品を落とすようにしました。アームの力を弱くすることができたらしていました。

```

220 case 6: //原点に戻す (手前)
221 if ( pi_temaef == 1 ) pi_temaef = 1;
222 if ( pi_temaef == 0 )
223 {
224   temaef();
225 }
226 else

```

```
227 {  
228   tomeru();  
229   oku();  
230   delay(600);  
231   tomeru();  
232   pi_temaef = 0;  
233   mode = 7;  
234 }  
235 break;
```

このプログラムは、原点の手前側に戻すためのプログラムである。224 行目でアームを手前側に移動している。221 行目で手前側にあるフォトインタラプタが反応すると、アームを止める。しかし、モータはすぐには止めることができないので止めたい位置で止まらなかった。229, 230 行目で少し奥側に戻している。原点の左側に戻すためのプログラムもこれと同じように書いている。

3. 研究のまとめ

今回課題研究で、クレーンゲームを作成してモノづくりに対しての興味を持った。最初に年間計画を立てていたが、全然見通しを持てなかったのが原因で年間通りに進まなかった。3 年生の夏休みに就職組は履歴書を書いたり、若年者モノづくりコンテストに参加したり、ロボットコンテストに参加したりと夏休みに課題研究をたくさんする予定が崩れた。最初に作り始める段階で自分たちの行事や予定を確認し、課題研究の年間計画を立てるべきだと思った。

クレーンゲームを作成するにあたって、必要なものは考えて買ったけれど、あまり使い方を調べていなかったのが原因で枠組み作成時にブラケットが必要だとか、モータードライバを作成時に起動電圧を調べていないのでうまくモータを制御することができなかった。最初の準備はとても大切だなと思った。

電子回路をするときは、どこにどのくらい

の電圧をかけるかを明確にしていたので、電子回路を考えるのは、すぐにできた。配線時にきちんと配線しないと予想もしてないところに電流が流れて、Arduino が二枚壊れたり、PC を破壊しかけたりと大変な目に遭った。これからは電子回路を作成するときは、配線を間違えないように気を付けたいと思った。

プログラムは、3 年間の学習と経験があったので特に調べたり、すごく迷ったりすることがなかったのでスムーズに進めることができた。3 年間プログラムの勉強をしていてよかったと思った。

計画通りに進めることができなくて、文化祭展示をすることができなかった。周りの人に感想を聞くことができなかったのが心残りである。課題研究を通してたくさん学ぶことができた。将来もこの課題研究を生かしたものづくりをしていきたいと思った。

クレーンゲームのプログラム

<https://github.com/Lettyscarlet/program.git>