BUTUMU

2022 / 9 / 30 ~ 2022 / 10 / 2

AZB 75th

物理部無線班
回路図集

Circuit

はじめに

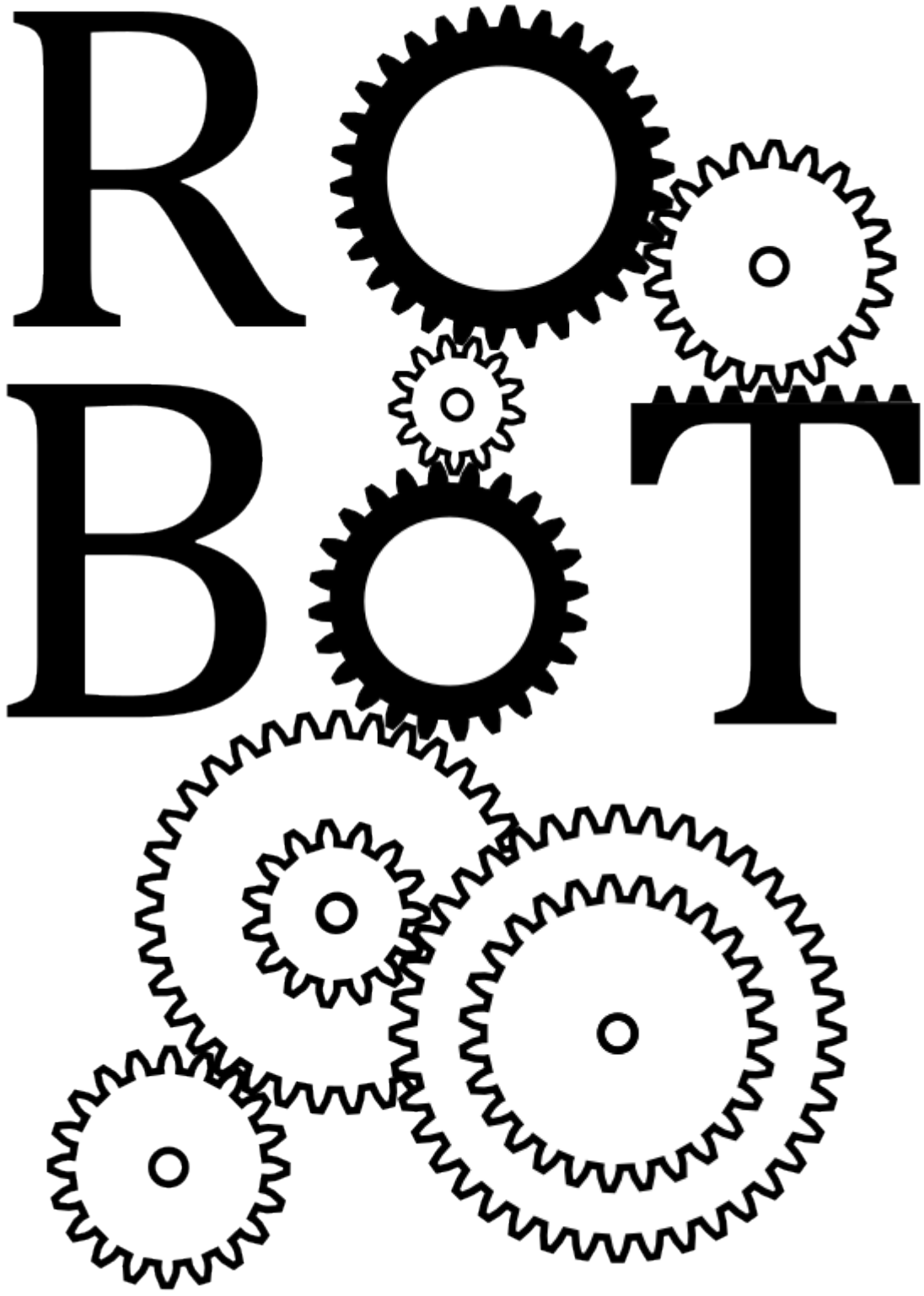
昨年の回路図集では「Microsoftが6月24日に『Windows10』の後継となるOSである『Windows11』を発表」とありましたが，最近では3月4日にスマートグラスの「Nreal Air」が発売されるなど，コロナ禍による行動制限もありAR，VR製品の市場が拡大してきています。

我が部では，昨年の経験を活かしたロボットや，これまでにない形を求めた製作物など，多種多様な研究に挑戦しています。

この回路図集は麻布学園物理部無線班の部員が一年かけて研究，製作したものについての記録です。未完成のものも含まれており，本書の通りに作っても動作する保証はございません。予め御了承下さい。

H2会計 加藤稜大

-目次-


ROBOT

22式弦楽器自動演奏装置「空震」

製作者　H1 伊藤
M3 鹿島
協力　物無の皆様

ストーリー

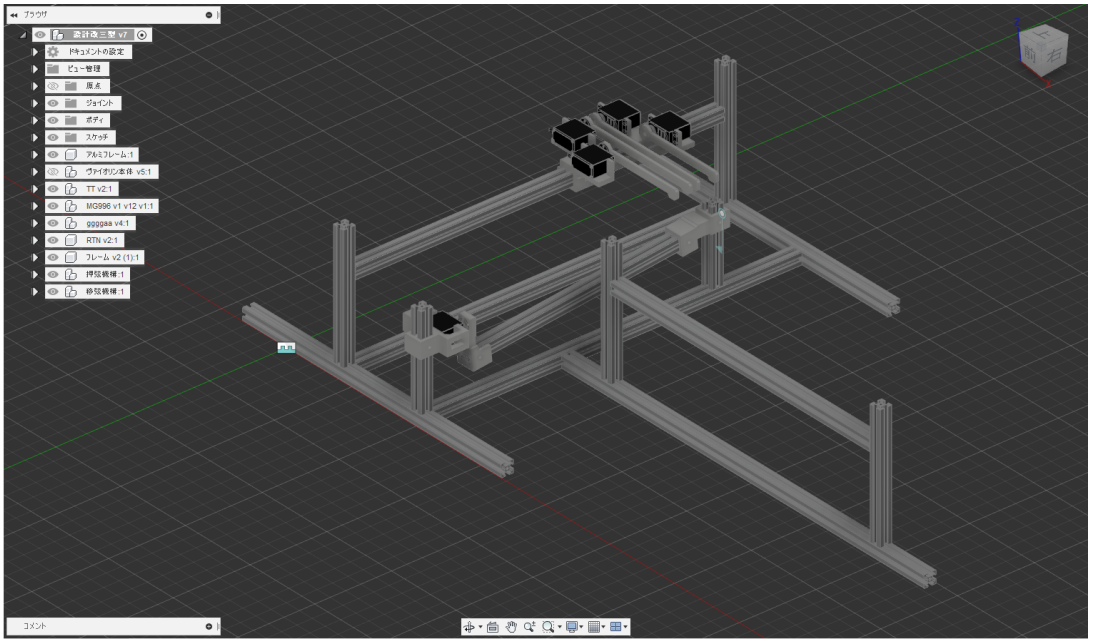
20XX年、物無連邦と管弦共和国はワタリロウカ半島を巡り、激しい戦いを繰り広げていた。劣勢となった物無連邦は管弦共和国兵の模倣兵器の開発を命じる。そして一年後、管弦共和国兵を完全に模倣した特殊兵器が完成した……

※超適当に作りました

概要

自動でヴァイオリンを演奏するロボットです。パソコン側で作曲した曲を送信すると、全自動で演奏してくれます。
本体のフレームはSUS社さんのアルミフレームとブラケットを使用し、高い耐久性と安定性を兼ね備えています。その他駆動パーツやサーボモーターブラケット等は、CADにて設計し印刷しています

CADでの設計↓



使用した部品

・アルミフレーム×沢山
・サーボモーターMG996r×5
・ステッピングモーター×2
・Arduino×2
・モータードライバーシールド×2
・やる気×適量

仕組み

arduinoと呼ばれるマイコンにプログラムを書いて制御しています。
パソコンから送る曲のデータはMIDIと呼ばれるデータで、専用ソフトで簡単に作曲することが可能です。そして、パソコンから送信されたMIDIデータは音程が数字として送信されているのでマイコン側で特定の関数と配列を使用し、指の配置を決定します。
マイコンから指示を受けたサーボモーターとステッピングモーターは、素早く所定の位置へ移動します。

以下プログラム

```
#include <A4988.h>  //ライブラリのインクルード
#include <Servo.h>
#include <MIDI.h>

int x=0;
int servo1[8] = {10, 0, 0, 10, 10, 10, 10};  //配列や変数の定義
int servo2[8] = {6, 6, 6, 0, 0, 6, 6};
int servo3[8] = {4, 4, 4, 4, 4, 15, 15};
int servo5[5] = {138, 120, 105, 90};
int set[8] = {0, -20, 0, -20, 0, 0, 20, 20};
int LR = 1;
int y=0;
int nt = 0;

A4988 violin(200, 8, 12);   //モーターの定義
A4988 bow(200, 4, 7);
Servo ser1,ser2,ser3,ser4,ser5;
MIDI_CREATE_INSTANCE(HardwareSerial, Serial, MIDI);

void setup() {
  violin.begin(110, 30);  //モーターの設定
  bow.begin(80, 30);
   ser1.attach(3, 500, 2400);
   ser2.attach(5, 500, 2400);
   ser3.attach(6, 500, 2400);
   ser4.attach(9, 500, 2400);
   ser5.attach(10, 500, 2400) ;
   MIDI.begin(MIDI_CHANNEL_OMNI);
   Serial.begin(38400);
   ser1.write(servo1[0]);
   ser2.write(servo2[0]);
   ser3.write(servo3[0]);
   ser4.write(0);
   ser5.write(servo5[0]);
}
void loop() {
            if (MIDI.read())  //MIDIを受け取った時の処理
            {
            switch(MIDI.getType())
            {
            case midi::NoteOn:
            if (MIDI.getData1() < 83 & MIDI.getData1() > 54) {
            yamato(MIDI.getData1());
            }
            LR = LR * -1;
            nt = 1;
```

```
            break;
            case midi::NoteOff:
            nt = 0;
            break;
            }
            }
            switch(y){
            case 600:
            LR = -1;
            break;
            case 0:
            LR = 1;
            break;
            }
  if (nt = 1) {
            bow.rotate(LR);
            y = y + LR;
  }
}

void yamato(int hori){     //MIDIがONになった時のモーターの動きの決定
  int tinu = ((hori - 48) / 7)-1;
  int soki = (hori - 48) % 7;
  ser5.write(servo5[tinu]);
  ser3.write(servo3[soki]);
  ser2.write(servo2[soki]);
  ser1.write(servo1[soki]);
  violin.rotate(set[soki] - x - 2);
  x = set[soki];
}
  ser3.write(servo3[soki]);
  x = set[soki];
}
```

感想

伊藤
優秀な後輩と気力によって、最速で完動（完全動作の意味）させることが出来ました。来年はAIを使ったロボットを作りたいと思っています。

鹿島
pythonなどプログラミング言語の学習が不十分であったこともあり、あまり役に立つことができませんでした。すみません。共同製作者の伊藤さんには一から丁寧に教えていただき本当に助かりました。ありがとうございました。

RECT

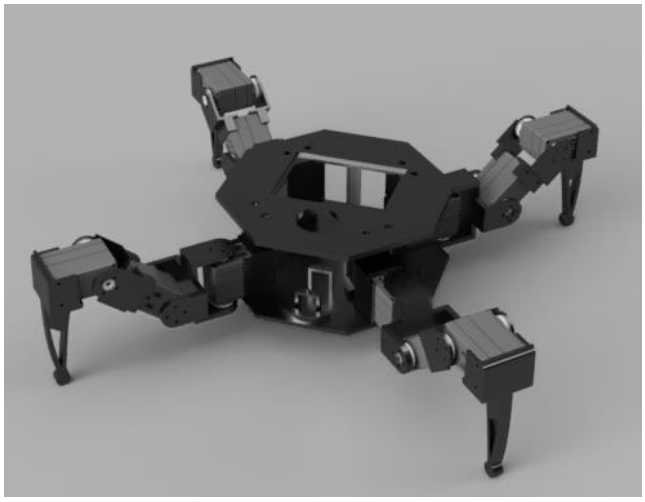
製作:　H1　広瀬 晴喜
山田 耕大
M3　三牧 周平

RECT

ストーリー

N 村さん「このサーボを お前に預ける」
3 人「！」
N 村さん「おれの大切なサーボだ いつかきっと返しに来い 立派な物無員になってな」
――そして少年達の製作は一年後のこの場所から始まる――

概要



四足歩行ロボットです。昆虫のように体の側面から水平に足を出して歩きます。Wii コントローラーを使った遠隔操作が可能で、それぞれのボタンに前後左右移動や回転移動が割り当てられています。また、距離センサーで周囲の物体との距離を測り、PC の画面に映すこともできます。

使用部品

・Arduino Uno Rev 3
　ロボットの頭脳ともいえる部分で、プログラムを実行して各部品に信号を送ります。
・USB ホストシールド&USB ドングル
　Wii コントローラーを Arduino で使うための部品です。
・Wii コントローラー　任天堂純正品の MotionPlusInside のものを使っています。
・サーボモーター(sg-90)&距離センサー
　周辺の物体との距離を測るために使っている部品です。

・XBee モジュール＆XBee インターフェースボード＆Xbee ピッチ変換基板

Arduino と PC との無線通信を担っています。インターフェースボードは PC と XBee を接続するための部品で、ピッチ変換基板はピン幅が狭い XBee をユニバーサル基板で使うための部品です。

・サーボモーター（KRS-2552RHV）

ロボットの関節となる部品です。シリアルサーボに分類される、サーボモーターのなかでも高価なもので、トルクも十分にあります。

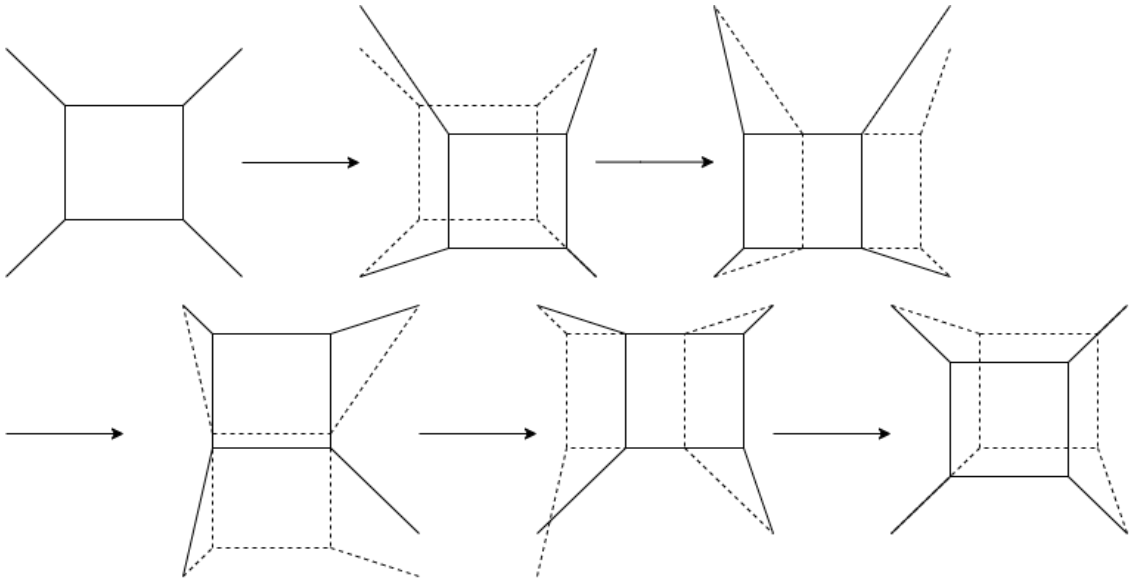
・ICS 変換基板

Arduino と上のサーボモーターをつなぐ部品です。

仕組み

・歩行モーションについて

足の先の位置を固定して体の重心を動かしてから足を前に出すことで滑らかで安定した歩行ができます。



・部品の接続について

・Arduino（サーボ用）---ICS 変換基板---サーボ

近藤科学が配布しているライブラリを使って Arduino と ICS 変換基板が UART 通信し、その信号をサーボモーター用の信号に変換して伝えます。今回足に使っているものはシリアルサーボであるため、信号の線は一本のみであり、これを全てのサーボで共有して使います（デイジーチェーン接続と言います）。このため、配線を比較的簡潔にすることが出来ます。

・Arduino（サーボ用）---Arduino（通信・計測用）

Wii コントローラーからの入力を受け取るために、デジタル I/O 同士を何本かつない

で、パラレル通信しています。はじめはシリアル通信で済ませるつもりでしたが、他の部品とのシリアル通信が原因でこの Arduino にかかる負担が重く、通信に失敗しやすかったため、パラレル通信を使いました。

・USB ドングル---USB ホストシールド---Arduino(通信・計測用)
USB ホストシールドを通じてドングルと Arduino を通信させています。これは海外の方が配布しているライブラリを使っています。シールド基盤なので配線はしていません。ドングルは Wii コントローラーとの相性が厳しいらしいので Arduino と通信させたい場合は BT-micro4 を使うことをお勧めします。

・USB ドングル---Wii コントローラー
Bluetooth 接続です。最初はコントローラーの登録が必要です。

・Arduino(通信・計測用)---XBee(Arduino 側)
XBee と Arduino で UART 通信をしています。この通信は計測したデータを PC に送信することが主な用途です。XBee 単体だとピッチ幅がユニバーサル基板と異なり、配線が難しくなるので秋月のピッチ変換基板を使ってユニバーサル基板で配線しています。ピッチ変換基板には電源用のレギュレーターがついているので電源電圧は 5V でも大丈夫ですが、ピンへの入力はその限りではないため電圧を 3.3v まで下げる必要があります。また XBee からの出力を受け取るときは 3.3V で出力されますが Arduino は H と認識してくれるのでそのままつないでも問題ありません。

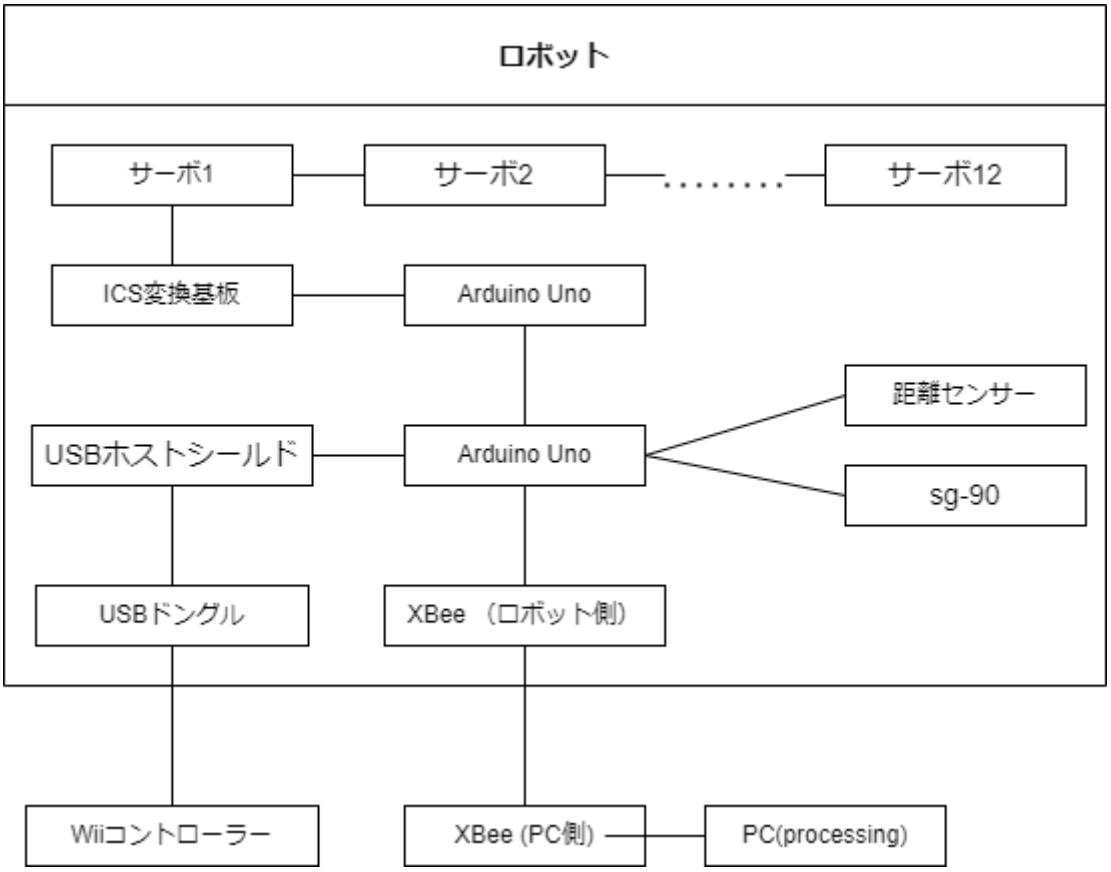
・XBee(Arduino 側)---XBee(PC 側)
Arduino から受け取ったデータを PC 側へ無線で送信しています。使用前に XCTU という専用の設定ソフトを使って役割を設定する必要があります。

・XBee(PC 側)---PC(Processing)
XBee からのデータを USB ポート経由で Processing というソフトへ送ります。送られたデータを Processing 内で処理し、その結果を画面に映します。

・Arduino(通信・計測用)---sg-90
Arduino からセンサーを動かすためのサーボへ信号を送ります。ArduinoIDE に元から備わっている Servo.h というライブラリで簡単に制御できます。

・Arduino(通信・計測用)---距離センサー
HC-SR04 という安価な超音波距離センサーモジュールを使っています。

（接続図）



感想

広瀬： 自分が最高学年である初めての製作であり、指示の出し方など至らない点もあったと思いますが、それでもついてきてくれた二人にはとても感謝しています。途中でボトムアップ気味になっていってしまったので、来年は作る前に計画を綿密に練ってから作業したいと思います。

山田：後半はあまり製作にかかわれませんでしたが、fusion は初めて触ったので楽しかったです。

三牧：今回の制作で僕はモデリングやプログラミングなど様々なことをしました。かねてからの夢であったロボットを作るということを経験することができ、とても楽しかったです。広瀬さんに頼りきりだったというのも感じたので来年はこの製作の経験と反省を生かしさらに素晴らしいロボットを作りたいと思います。製作途中に 2 回も PC が壊れたのはすみませんでした。

リニアモーターカー

製作者　M3 片山泰賀
H2 中澤凌太郎さん
協力　　物無の皆様

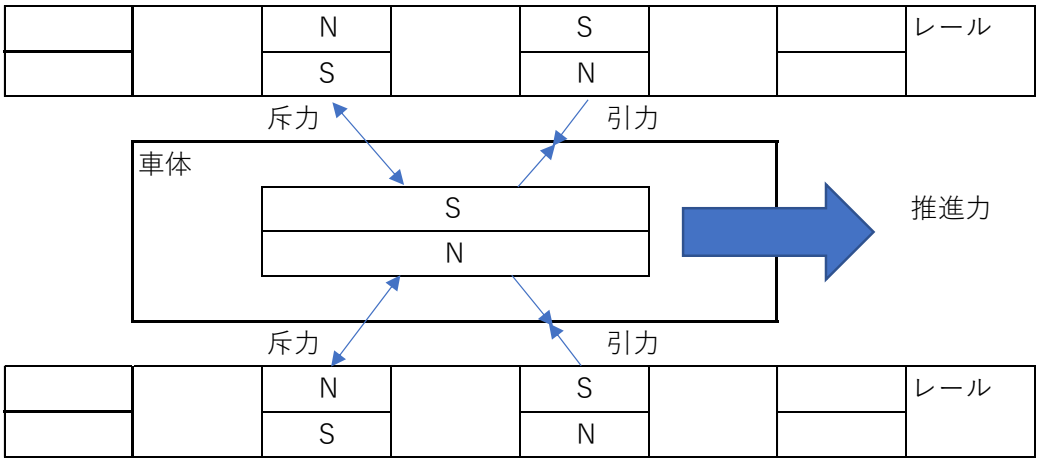
① ストーリー…

自分（片山）は小学校 3 年生のときに図鑑で読んだリニアモーターカーに心を惹かれそれを作ってみたいと思い、挑戦するも失敗に終わってしまいました。自分はもう一度リニアモーターカーの制作をしたいと思い、今回の製作物に選ばせて頂きました。
実はこのリニアモーターカー物無で作られるのは初めてではなくて過去に 2008 年度の文化祭でも作られています。当時のものとは作りは違うものの、参考にさせて頂いたところがあります。
気になる人は下記のリンクのものを見てみてください。
http://butumu.com/schematic/pdf/2008.pdf

② 概要…

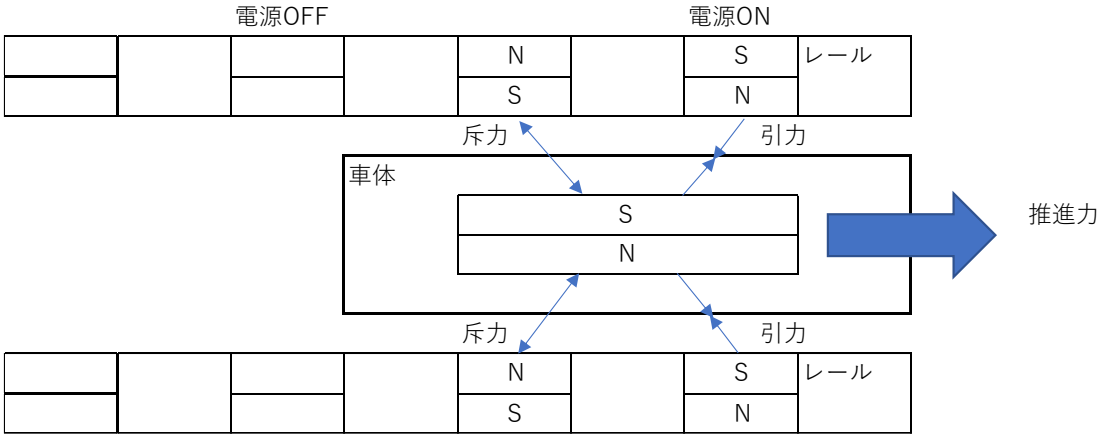
(1) リニアモーターカーとは

皆さんはリニアモーターカー（英名： maglev train）の名前の由来を知っているでしょうか。リニアモーターカーとは和製英語でありリニアとは「直線的な」という意味の言葉です。本来回転方向に力を発生させるモーターを直線方向に力を発生させるものがリニアモーターカーです。
実際に走っているリニアモーターカーの仕組みについて簡単に説明します。
リニアモーターカーは車体とレールに電磁石を配置して図(1)のように引力と斥力を発生させることによって前方向への推進力を得ています。前へ進んだリニアモーターカーはレール側の電磁石と車体側の電磁石が引き合って止まってしまうと考えられます。　　図(1)



しかしここがリニアモーターカーの面白いところでもあるのですが、実際に走っているリニアモーターカーでは車体側の電磁石とレール側の電磁石のお互い引きあっている磁石同士が同じ位置に来た時にレール側の電磁石の極を入れ替えることで再び前方向の推進力を生みだしています。これを繰り返すことでリニアモーターカーは前に進みます。リニアモーターカーは車体が通る瞬間に横のレールのところの電磁石だけを図(2)のように稼働させています。

図(2)（図(1)から進んで電磁石の極が入れ替わり後ろの電磁石の電源が OFF になり前の電磁石の電源が ON になった状態）

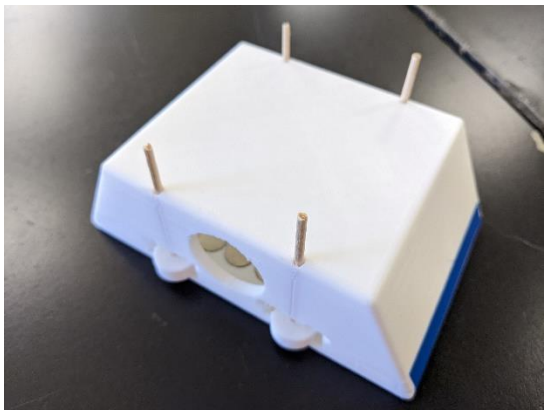


このようにすることで、リニアモーターカーは動いています。そもそも電車などの車輪がついた乗り物には地面との摩擦が発生するためにある一定の速度以上を出すと車輪が空回りしてしまい出せる速度にも限界があります。そこでリニアモーターカーは車体を浮かせることでその限界をなくしています。

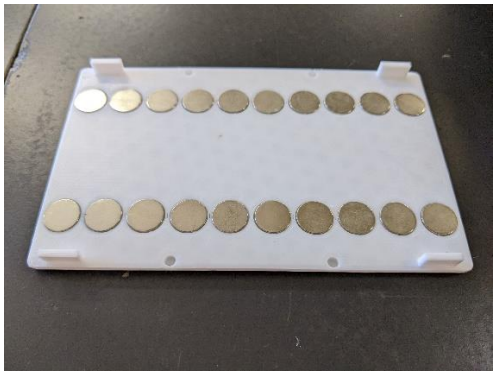
(2) 本製作について

リニアモーターカーの仕組みがある程度わかってもらったと思うのでここで本製作についての説明をしたいと思います。本製作で初めはレール側に電磁石を取り付けて車体側に磁石を取り付けてレール側の磁石の極を入れ替えることで車体を動かそうとか考えていましたがそれをするためには大量の電磁石を制御する必要があるので断念し車体側に電磁石をレール側に磁石を取り付けて車体側の電磁石をArduino(マイコンの一種)で制御することで車体を前に進めることにしました。車体の浮上は車体の底とレールの底に磁石を配置してその斥力で浮かせることにして車体は左右にずれないようにするために車輪を車体の横につけています。また

制御のための Arduino を乗せるのですが車体をなるべく軽くするために本製作では Arduino Nano を使用しています。

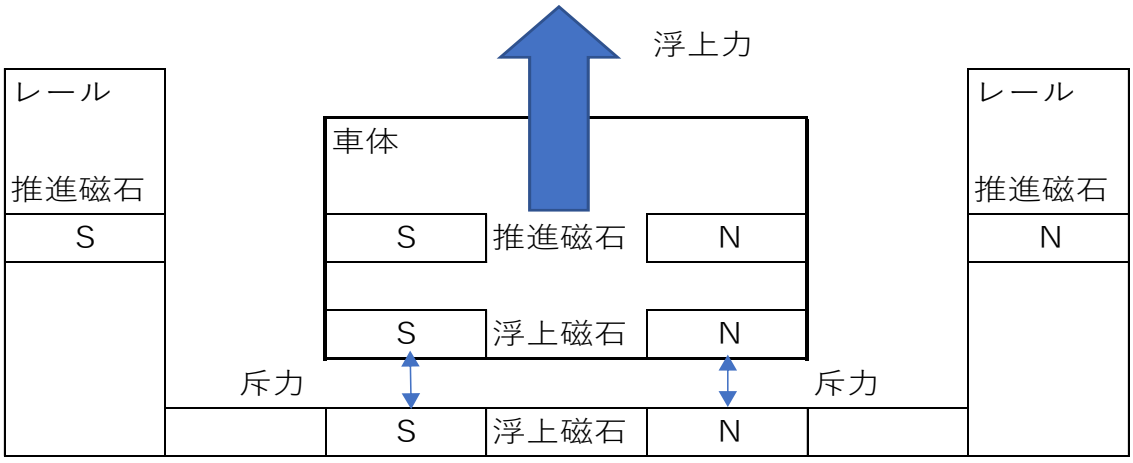


写真(1)



写真(2)

リニアモーターカーの外観は写真(1)のようになっています。この中に電磁石、乾電池、モバイルバッテリー、Arduino、基盤が入ります。横の穴に電磁石を左右取り付けます。穴から小さく見えるのが浮上用の磁石です。車体の上のパーツを取り外して車体の底面が見えるようにした写真は写真(2)の容易になっています。浮上用の磁石は二列に分けて配置しておりそれぞれ S 極、N 極に分かれています。レールの底も同じように二列の S 極と N 極の磁石が並べられておりその磁石同士の斥力で浮いています。

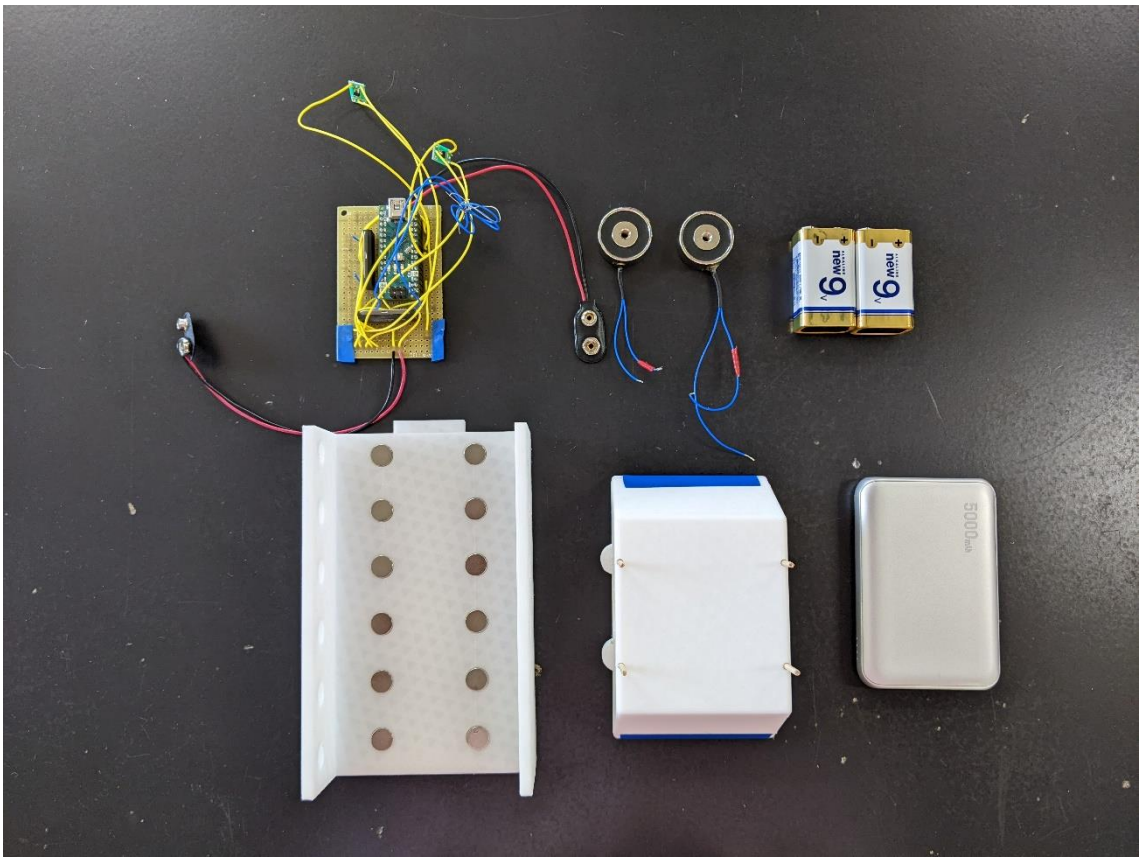


図(3)

まとめると上の図(3)のように車体とレールには二種類の磁石が使われていて車体とレールの底面に使われている浮上用磁石と車体の側面の電磁石とレールの側面に使われている推進用磁石です。これらの働きによって本製作のリニアモーターカーは浮いて走行することを可能にしています。

③ 使用した部品

使用した部品の説明です。使用した部品は写真(3)より左上から Arduino Nano、磁気センサー、基盤、回路、電磁石×２個、９V 乾電池×2 個、レール（写真では１個ですが直線型×4 個。カーブ型×8 個)、車体、モバイルバッテリー、磁石×約 300 個（写真では車体とレールについている）です。

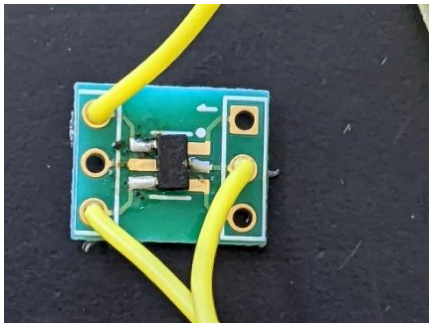


写真(3)

これらの部品の内 Arduino Nano、磁気センサー、基盤、回路、電磁石×２個、９V 乾電池×2 個、モバイルバッテリー、磁石×20 個を車体に積んで走行します。車体のほうに電磁石をつけるということはその電磁石だけを操作すれば良い分プログラミングも簡単ですが電磁石を制御するための Arduino Nano やその電源となるモバイルバッテリー、電磁石の電源となる乾電池などのものを車体に乗せなければいけなくその分車体が重くなってしまい進みにくくなるという欠点もあります。

④ 原理

ここまでリニアモーターカーが電磁石の極を切り替えて車体が走行すると説明していましたがその原理を説明していませんでした。ここではそれを説明したいと思います。そこで登場するのが写真(4)の磁気センサーです。磁気センサーといってもその中にも種類があり片極検知、両極検知、交番検知の 3 種類がありこれはそのなかの交番検知とい



写真(4)

う種類のものです。交番検知は磁石の極の向きが入れ替わったことを検知するものです。これを進行方向に対して車体の電磁石の前の側面に取り付けることでレール側の側面についている推進用磁石は S 極と N 極が交互に配置されているため車体が進んで次のレールの側面の推進用磁石のところに来たら磁気センサーがそれを感知して信号を送ります。Arduino がその信号を受け取ると電磁石に流れる電流の向きを反転させます。そうすることによって車体の電磁石の極の向きを替えてそれを繰り返すことで車体は前に進んでいます。

以下に Arduino のプログラミングを載せます。

```
const int aPin = 10;
const int bPin = 11;
const int cPin = 12;
const int dPin = 13;
const int IN1 = 6;
const int IN2 = 7;

 void setup() {

  pinMode(aPin,OUTPUT);
  pinMode(bPin,OUTPUT);
  pinMode(cPin,OUTPUT);
  pinMode(dPin,OUTPUT);
  pinMode(IN1,INPUT);
  pinMode(IN2,INPUT);

}

 void loop() {

  if(IN1 == HIGH && IN2 == LOW){
  digitalWrite(aPin,HIGH);
  digitalWrite(bPin,LOW);//電磁石 1 を n 極にする
```

```
  digitalWrite(cPin,HIGH);
  digitalWrite(dPin,LOW);//電磁石 2 を s 極にする
}
   else if(IN1 == LOW && IN2 == HIGH){
  digitalWrite(aPin,LOW);
  digitalWrite(bPin,HIGH);//電磁石 1 を s 極にする
  digitalWrite(cPin,LOW);
  digitalWrite(dPin,HIGH);//電磁石 2 を n 極にする
  }

 }
```

⑤ **感想**

自分は鉄道が好きなこともあり、片山君の制作に興味を持ったので一緒に制作することにした。制作を始めた頃は『実物のようにレール側の極を上手いタイミングで交互に切り換えれば走る』と考えていたが、もちろん現実はそんなに甘くはなかった。最初は実物と同じような仕組みで、レール側の電磁石を切り替えることにより前進させようとしたが、一度に何十個もの電磁石の極を切り替えるのは難易度が高すぎるので断念。結局車体側に電磁石を取り付けてそれを交互に切り替える方式にした。振り返ると本当に試行錯誤の連続だったと思う。最も苦労した点は、実際にリニアモーターカーを電子工作で制作している例があまりにも少なく、情報収集が困難だった点だ。でも諦めるという選択肢は僕らにはなかったので、二人で「どうすれば前に進むか」を必死に考えた。「この案だとこうなってしまい走らないのでは？』と思う箇所があれば積極的に言い合い、案を何度も改良した。片山君の『車両の側面に車輪をつけることにより、カーブを曲がりやすくする」というアイデアはとても良いと思った。僕にとっては最後の制作となったが、この経験が非常に有意義な物であったことは間違いない。

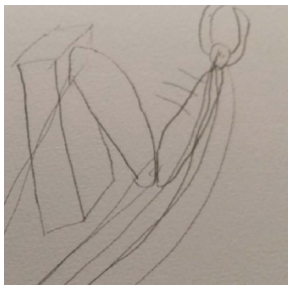
（中澤凌太郎さん）

今年の製作を通して自分はリニアモーターカー、3D プリンター、への理解を深めることができました。Arduino などのプログラミングについてももう少し詳しく知りたかったのですがそれが出来なかったことは残念です。実際のリニアモーターカーでは車体の左右と上下の誘導はレールに取り付けた 8 の字コイルを使って行っていますしそれを使うには実際のリニアモーターカーでも時速 150 キロまで速度を上げてからなので今回の製作では断念しました。今後はこの 8 の字コイルを製作で使えないか考えたいです。最後にこの一年間の製作に置きまして協力してくださった物無員の皆様に感謝申し上げます。（片山泰賀）

バイオニックハンド製作譚

製作者
H1 榎本大智
M3 井上翔介
M3 小林宏明
協力　物理部無線班の皆様

Story
「疼くッッッッッ、疼くぞこの左腕がッッッッッッッッッ！」
不治の病、厨二病を患っていたE本は連日物大に現れては左手を抱え、こう叫んでいた。
この状況を見かねたマッドサイエンティストI上とその相棒のK林はE本を黙らせる為にロボットアームの製作をすることにした。
そう、それこそが今は物無の伝説となったバイオニックハンドそのものだったのである。



・購入した部品
Myoware筋電位センサー　5000円ぐらい
２個目を買いに行こうとしたら円安で消滅していました。
筋電位を取得できる低価格帯のセンサーは日本にはもう存在していない気がします。転売しません。

Myoware 筋電位センサー対応電極パッド　10枚　2000円ぐらい
公式のサプライの電極パッドで最初に買いました。誤作動はなかったのですが小さくて貼る位置の調整が難しいのとジェルの関係ですぐ使えなくなってしまうので扱いに困りました。

OMRON 電極パッド　２枚　2000円ぐらい
公式サプライの反省を経て２個目として買いました。
洗って繰り返し使えるのも良いですし、大きさもちょうど良いので買った３つ

の中ではベストでした。

Panasonic 電極パッド　４枚　1500円ぐらい
筋電位センサーの調子が悪い時に電極パッドのせいにして買いました。
安くても普通に動いてくれているので公式サプライだけがイマイチだったのだなあと。

MCP3208 AD変換器　500円ぐらい
RaspberryPiで使うには割とポピュラーなAD変換機らしいです。
ネットにコードがいっぱい転がっているのですがなかなか動作してくれず、苦労しました。

・使用した部品など
RaspberryPi 3 model B
製作初期に借りたものをパクって最後まで使いました。
正直３ではディープラーニング等に使うにはスペック不足かなと思っていたので最後まで使えたことに驚きました。

３Dプリンター製ロボットアーム
井上がすべて設計してプリントしてくれました。おかげで僕に知識がありません。

仕組み・概要

まずこの筋電義手はアーム本体とRaspberryPi、そしてRaspberryPiのアタッチメントとしてMyoware筋電位センサーの３つを主軸として構成されています。
使用したプログラミング言語はPythonです。
筋電位センサーからの信号を元にRaspberry　Pi上でリアルタイムに動きを判別、処理して同じ動きをアームに行わせるのが今回の目的です。

アーム本体はFusion360というソフトを利用して設計し、3Dプリンターで印刷しました。そこにサーボモーターを取り付け、２軸で動くようにしました。

アームの製作終了後は、筋電位センサーとRaspberryPiの接続をしました。ここで筋電位センサーはアナログ、Raspberry　PIはデジタルなのでAD変換機を用いて変換しなければなりません。今回はMCP3208を購入し、PythonでAD変換用のコードを書きました。

Raspberry　Pi上で筋電位センサーの信号が読み取れるようになったので次に判別、処理をディープラーニングという技術を用いて行うことにしました。
ディープラーニングとは人工知能技術の一つで機械に大量の情報の重みとバイアスを記録させたニューラルネットワークを作り、新しいデータをニューラルネットワーク上で処理することで判別、分類などを行う技術です。。
この技術を用いることで本来判別の難しい筋電位の情報を高精度でロボットアームに反映させることができます。

今回は「握る」「反る」「曲げる」「前腕の動き」の４つをニューラルネットワークに学習させ、リアルタイムの筋電位情報を処理していきます。
最後に４つの動きに対応して動くサーボモータのプログラミングと判別した動きを結びつけて完成です。

特に時間がかかったのはAD変換のコードとディープラーニングの部分でした。

以上はかなり噛み砕いて原型のとどめていない概要なのですが、ディープラーニングについて僕らもあまり理解できていないのでどなたか教えていただきたいです。

感想・製作者紹介

H1 榎本大智

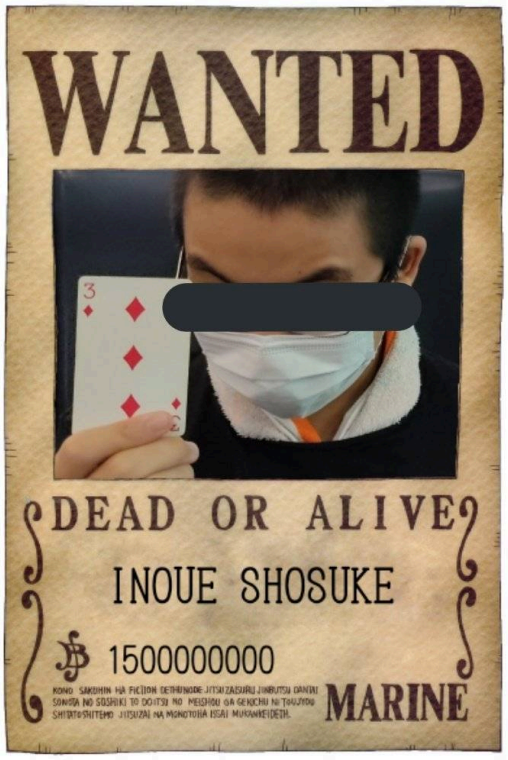
常に動き回っており止まることを知らないその男のことを人々は親しみを込めて「多動症患者」と呼んでいる。
写真は先日ウクライナに救援に行った時のものである。

製作の感想
僕としては満足です。
やれることは結構したんじゃないかな。
道草を楽しめ　大いにな

M3 井上翔介

懸賞金15億ドルの男。
昨今の円安により相対的に懸賞金が下がっている。
卓越した能力はもちろんのこと一番の武器はそのポーカーフェイスである。

製作の感想
ロボットアームのモデリング、プログラム作りの一部を手伝ったM3の井上翔介です。人生初めての共同制作でしたが、いつ休みいつ部活に来るのかなどを連絡できず、結果制作が進まなかったというような日もたくさんあり、意思疎通の難しさを知りました。また、どのように制作を進めていくのかを、リーダーの榎本さんに丸投げしてしまっていたところがあり、来年からは自分が何のために何をしているのかを完全に理解して制作に臨みたいと思っています。物理部無線班員の皆様、1年間ご協力いただきありがとうございました。



世界最高の頭脳を持つ男。
秘密結社によって物無にくることを妨害されているため出席率が低い。
来年度は物無を征服するらしい。

製作の感想
サーボモーターを担当した小林です。
Pythonという新しい言語を勉強するのはとても大変でしたが理解が深まり、
新しい世界が見えたということはとても感慨深いです
先輩や同輩のおかげでここまで来れました。

二足ロボット

制作者　H1 加藤 直広
M3 加藤 理人
協力　物無の皆様

・ストーリー

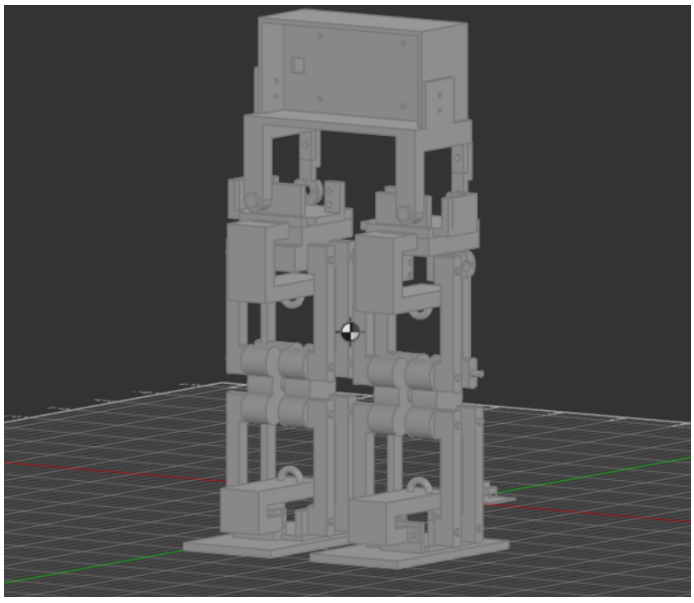
時は2122年、近年のAIとロボットの成長は凄まじく、世界中でロボットが溢れかえっていた。彼らは人類を遥かに凌駕し、第一次産業から第三次産業までのあらゆる産業を担った。人類の職の大半は彼らに奪われた。まともな職を持つ人類なんて既に全体の1%にも満たない。人類は日々、空き缶集めに勤しみ、街を歩くロボットに物乞いをする。そんな人類を気にも留めないロボットたちは今後開催されるロボットオリンピックに向けて、競技用ジャンプロボットの制作に取り掛かるのであった。

・概要

ジャンプと歩行の両方を二足でこなすことのできるロボットです。腰と足にそれぞれ二つついているサーボモーターを用いて歩行とジャンプを行い、腰にあるもう二つのサーボモーターを用いて機体のバランスを取り歩行をサポートします。

・外観

腰、股関節、膝、足を持つ人間の足を再現しています。



・機構

・本体の機構機構

機体の枠組みはすべて3dprinterで印刷されたpla素材のプラスチックで作られています。

腰と足にそれぞれ二つ付いているサーボモーターと平行リンクによってこのロボットは安定して直立し続けることができます。平行リンクとは、四角形における2組の対辺が共に等しければ

四角形は平行四辺形であるという平行四辺形の条件を用いたものです。このロボットにおける腰と膝の間と膝と足の間では骨を用いて平行四辺形が作られているため、足と膝と腰は常に平行になっています。

ジャンプは両足の膝部分にあるばねを収縮、解放させることで行います。ばねの収縮は腰と足にあるサーボモーターを回して行い、ばねの解放は腰と足にあるサーボモーターを反対向きに回して行います。

歩行は主に腰と足にあるサーボモーターを用いるとともに、歩行中のバランス制御のために腰にもう二つあるサーボモーターを用いて行います。バランス制御のためのサーボモーターはロボット全体の重心を上にするために、腰部分にあります。

・コントローラーの機構

コントローラは上パーツ、基板、基板固定パーツ、下パーツの順で重ねられて構成されています。

上パーツ

タクトスイッチが押せるように穴が空いています。 下パーツ 基板からの配線が出せるように上パーツとの接着面にすきまがあります。

基板固定パーツ

基板とこのパーツをねじで留め、移動しないようにしています。 基板 動作を指令する4つのタクトスイッチがついており、それぞれからRaspberry Piにつなぐ配線が 出ています。

・プログラミングの機構

今回はPythonを用いてプログラミングを行いました。

プログラムの工程

1今回使うモジュールのインポート

今回はラズベリーパイのピンを設定するためのGPIO用モジュール、 時間を制御するtimeモジュール、並列処理をさせるためのconcurrest.futuresモジュール をインポートします。

2角度からデューティ比へ書き換え

角度から数式により、サーボモータを制御するためのデューティ比という数値への 書き換えを行います。

3ピンの設定

サーボモータ、タクトスイッチのどの部品をどのピンに繋ぐか、出力モードか入力モードかを決定します。

4サーボモータの位置ごとに関数を定義

各々のサーボモータに対応した、使用したい位置を定義した関数を定義します。 これによって、並列処理のコードを簡略化できます。

5スイッチ制御

スイッチが押されたことを認識し、それぞれのスイッチに対応した行動プログラムへ 移行させます。

6行動プログラム

ジャンプ　　足を曲げて屈んだ状態から一気に膝を伸ばし、バネの力を受けてジャンプします。 足を曲げる、伸ばすそれぞれの時並列処理によって複数のモータを同時に動かします。

歩行　　左右のバランスを2つのモータでとりつつ、4つのモータで足を交互に出すことによって、 歩行します。足を出す時に左右のバランスを取るために並列処理を使います。

・感想

加藤理人　Pythonについて僕はこの一年間を通して、プログラミング言語としてPythonを学びました。Pythonについての本を読む、簡単なゲームのコードを書いてみる、という順序で学習を進め、ロボットの製作に移行しました。Pythonはこの先使う機会が多くなると思うので、技術を上げたいと思います。モデリングについて3Dプリンターで印刷するためのモデリングについては、実際に動かしてみた時のことを考えて、どれくらいのサイズ、スペースが必要か、よりトラブルを少なくするためにはどのような形のパーツを作れば良いかなどを検討しながら作らなければならないことを学習しました。

加藤直広　僕はこの制作で機構の考案と本体のモデリングを担当しました。担当箇所を明確に分けてしまったことで共同制作者の加藤君に暇な時間を与えてしまった事は大きな反省点です。すみませんでした。制作期間中はとても楽しく有意義な時間だったと思います。ありがとうございました。

妖怪車人間

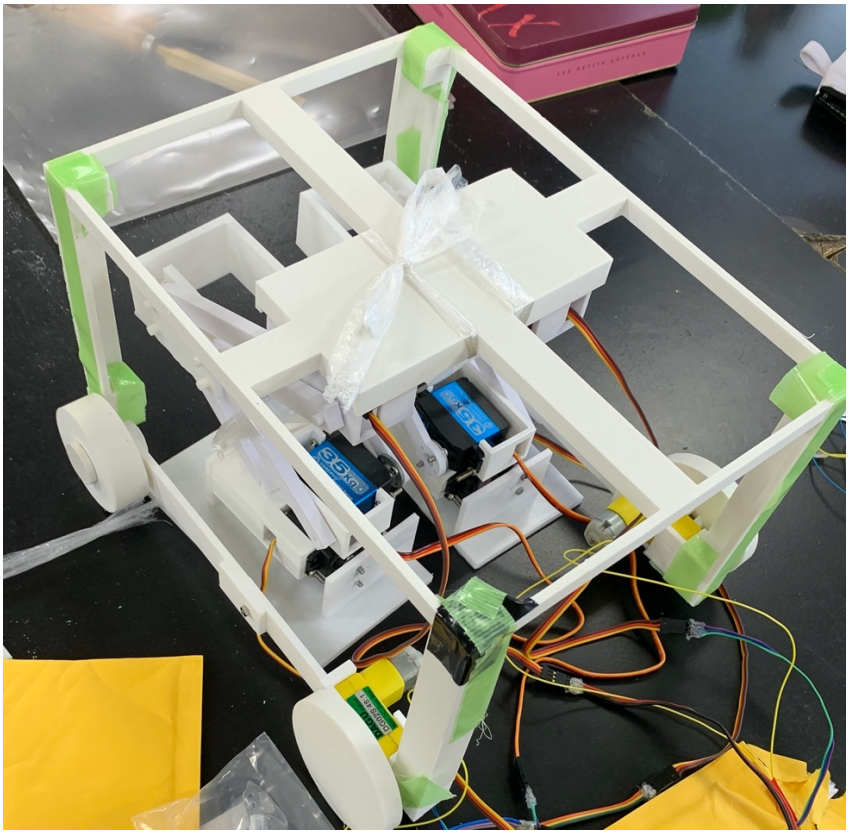
製作者：H1秋田M3小林　　協力:物無の皆様

・ストーリー

ごくごく普通の学校「麻布なんとか学園」に通う、ごくごく普通の中学４年生「秋〇泰〇」は、ある夏の日に奇妙な車に遭遇した。それから秋〇が無免許、無車券、無保険で操縦するようになった車は、なんと妖怪だったのだ。

・概要

タイヤで走る車の形態と二足で歩行する人型の形態の二つの形態があるロボットです。

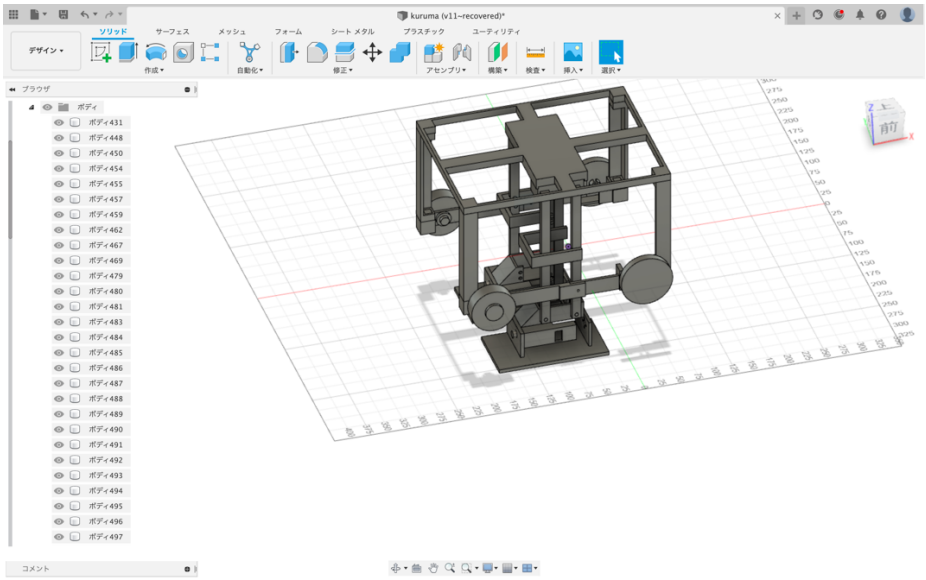


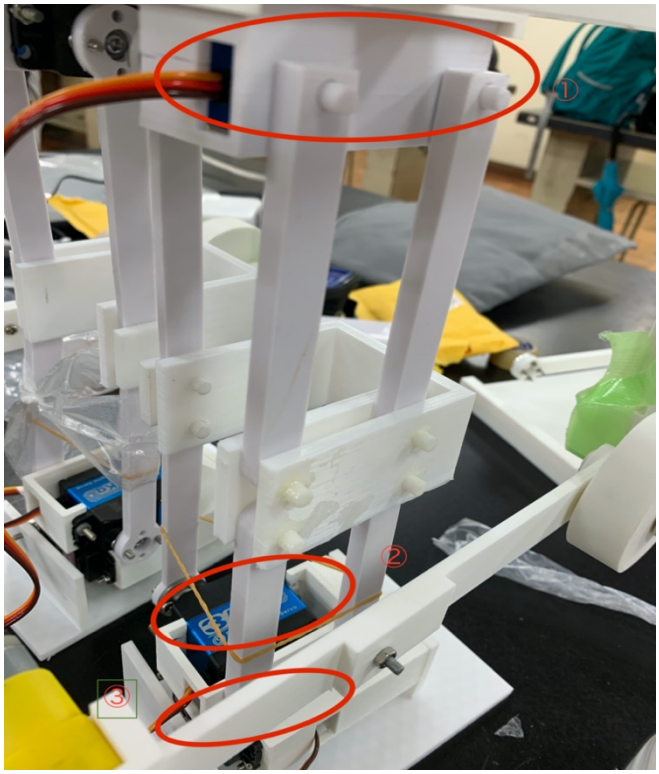
・主に使用した部品

サーボモーター、arduino、DCモーター、スイッチなど

・原理

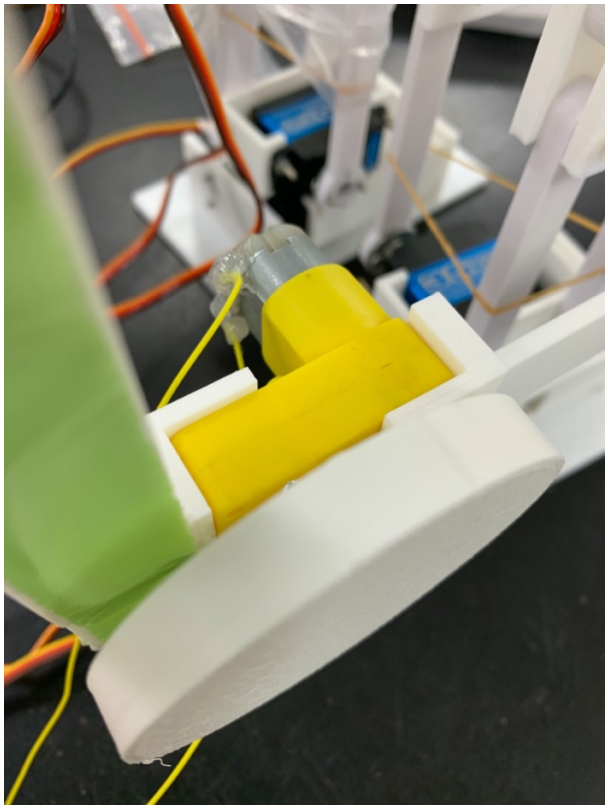
車体などは基本的に３Dプリンターで印刷しました。また、３Dモデルは Fusion360という3DCADソフトをつかいました。





形態変化の時は足を折り畳んだり伸ばしたりして変化します。歩行はあらかじめ組んでおいたプログラムでサーボモーターを動かして歩かせます。また、サーボモーターの制御arduinoという制御基盤にプログラムを書いて制御します。
足を曲げたり伸ばしたりするのは基本的に①②のサーボモーターをうごかして関節を曲げます。足を上げた時バランスを崩さないように③のサーボモーターを動かし機体全体を傾けて歩きます。

車状態はDCモーターを回して移動します。



・感想

車体の重量が重く思ったようにサーボモーターが動かなかったりなど、トラブルが何度か起こりかなり苦労した。二足で歩行させようとなると片足を上げたときのバランスなど懸念点はいくつもあったので、このロボットが初めて歩いた時は少し嬉しかった。（秋田）

サーボモーターを使って初めての製作でしたが回路を作っても動かなかった時は悲しかったですが先輩のお陰で動くことができて嬉しかったです(小林)

物無　回路図集原稿　変形戦車

製作者　H-1 和田
M-3 小平

おはなし

ある日の朝、ヘッドくんはいつも通り風にふかれて転がっておりました。
すると向こうから、チェストくんが同じように転がってきました。そこで二人はお互いにA学園の文化祭に行きたいと思っていたことを知り、合体して二人でA学園まで這って行くことにしました。
しかし、這っていたのではA学園の文化祭にはとても間に合いません。そこで二人は自分達の“足”として戦車を作ることを決心したのでした…
ちなみに彼らに“腕”がないのはここだけの話です…

概要

ボタンを押したらモーターが動いて変形する戦車です。本当にこれだけなんですが、何か書かないと怒られそうなので適当に埋めておきます。当初は全身プラ板で作るつもりだったのですが、W氏は某指定暴力団に恐喝され、K氏は某劇団に攫われ、完成のメドが立たなくなって３Dプリンターでの製作に切り替えました。ちなみにこの原稿を書いている時点では2割程しか完成していないのですが、はたして完成するんでしょうか…？

使用した部品

Arduino Micro	マイコン(マイクロコントローラ)がくっついた基盤です。それ以上は僕も知りません。プログラミングや制御はすべてコレで行っています。
SF3218MG	amazonで購入した謎のサーボモーターです。商品説明いわく「ストールトルク 20kg/cm」だそうで、主に脚の関節に使っています。
SG92R	よく見かけるTower Pro社製のサーボモーターです。これもamazonで購入しました。主に上半身の関節に使っています。
ノーマルモーター	これまたよく見かけるタミヤ製のモーターです。一番普通のやつですね。
連結式クローラー	これまたタミヤ製のキャタピラです。これが無ければ走れません。
ギヤボックス	これこれまたまたタミヤ製のギヤボックスです。
プラ板	タミヤ製のプラ板です。一部ウェーブ製のものも混じっているかも知れません。脚はプラ板で作っています。
スイッチ	返事がない。ただのスイッチのようだ。
(３Dプリンター)	脚以外のパーツを作るのに使いました。

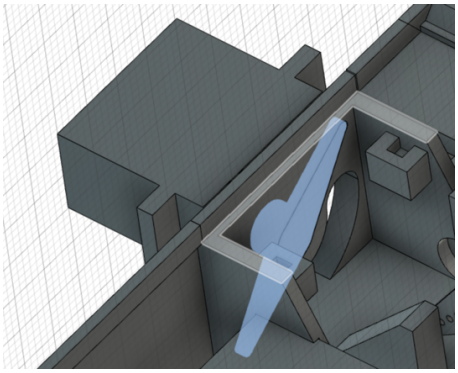
構造

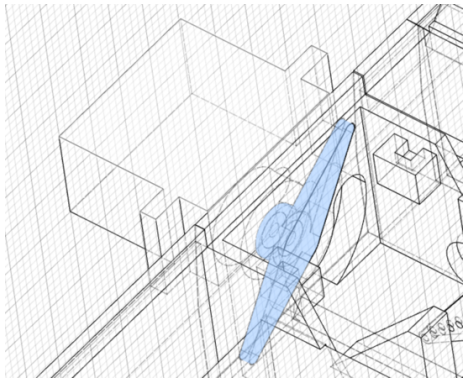
スイッチを押したらモーターが回転して戦車が変形します。それだけです。おわり。

と言いたいところですが、これだけだと流石に杜撰にも程があるので少しだけ補足しておきます。

① サーボモーターの固定

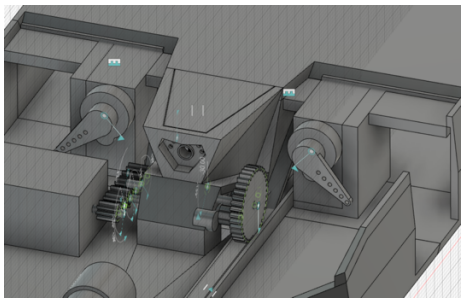
サーボホーン(とモーター本体)を装甲に埋め込む形で固定しています(下図参照)。サーボモーターとサーボホーンを接続するネジを外すだけで分解ができます。ラクでいいですね。

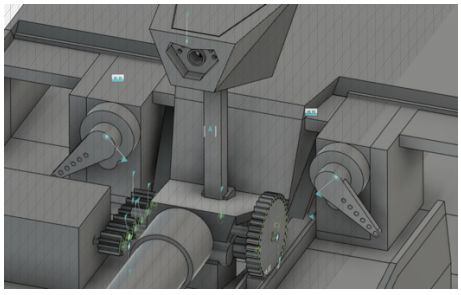




② 頭部・砲身の変形

サーボモーターではなくDCモーターで変形させています。頭部を動かすギヤのシャフトに砲身を動かすギヤも刺さっていて、ギヤ比を変えることで移動距離の差を誤魔化しています(下図参照)。モーターをたくさん使わずに済みます。安上がりでいいですね。





③ さらなる補足

装甲の固定には磁石を多用しており、ネジは殆ど使用していません。組立・分解がラクでいいですね。

感想

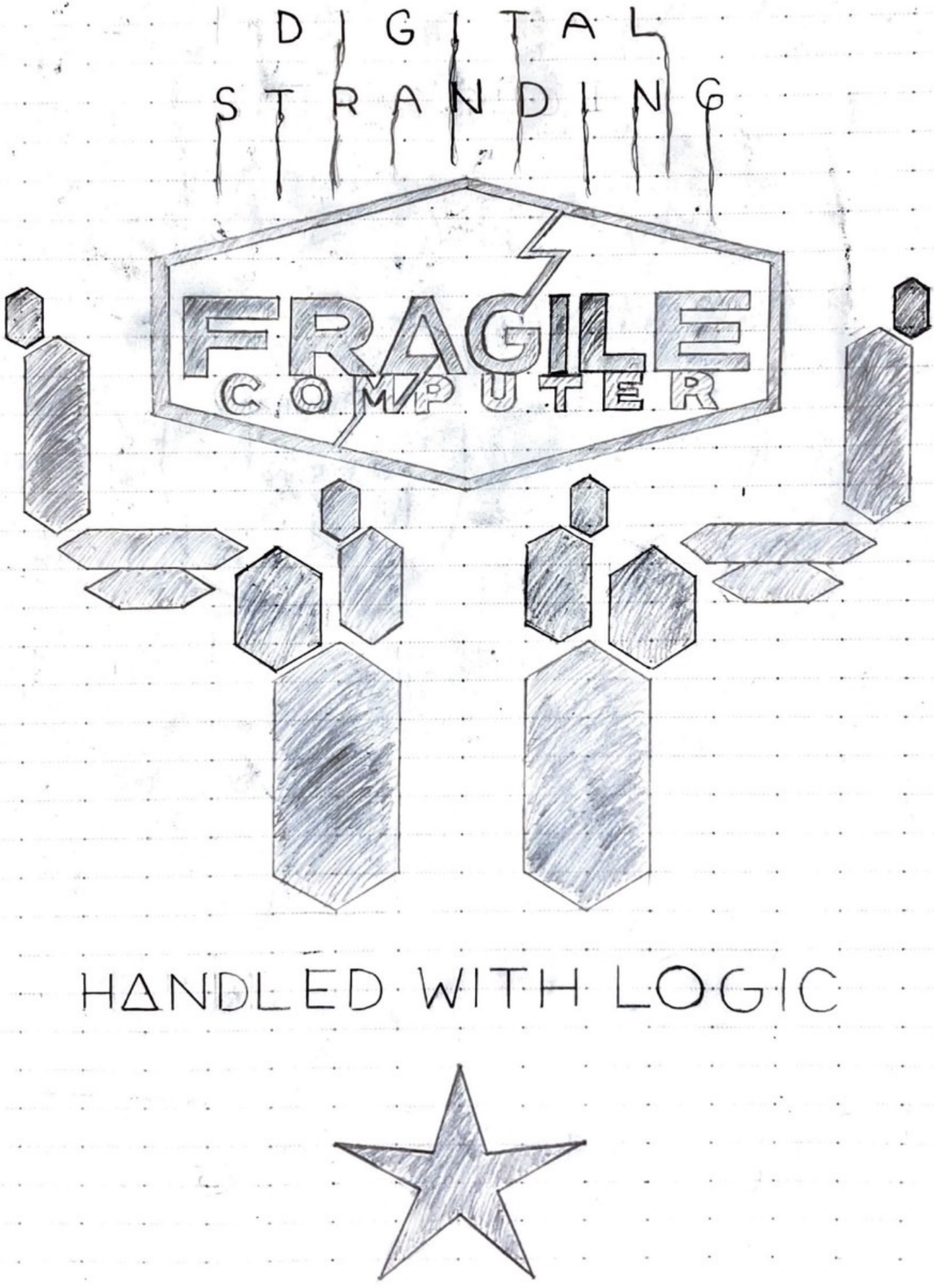
和田

「概要」の欄にも書いた通り、まだ殆どできていないため「完成しないのでは」と思うと震えが止まりません。ただ、このような状況に陥ったのも僕が「Fusion 360(3DCADソフト)の勉強するの面倒くさい！」と言ってプラ板での製作を決定したり、某塾の宿題を口実に部活に来なかったりしたせいです。小平君への仕事の振り方についても、ただ作業を与えるだけで物無の製作において最も楽しい(と僕は思っている)、「自分で機構を練って形にする」という活動を現状させてあげられていない点については深く反省するべきだと感じました。来年度の製作では、今年度以上に宿題がきつくなってくることまで考慮して、共同製作者が自分の独創性を存分に発揮できるような役割分担を考えられたらいいなと思います。

小平

プラ板の切り出し作業などを担当しました。修正などが楽だったので何でもかんでも3Dプリンターに頼らないのも良いなと感じました。

諸事情があって夏休み前は週一ペースでしか来れず、夏休み中はそれ以上に来れなかったことがかなりの痛手でした。次の製作ではこのようなことがない様に気をつけます。


DIGITAL
STRANDING
FRAGILE
COMPUTER
HANDLED WITH LOGIC

DENTAKU∞の創り方

製作　M3 坂東　H2 礒野
協力 物無の皆様、特に畠山、宮田さん

ストーリー（最後だけ嘘）

数学苦手野「うわーん！四則演算ができないよー！」
文系梨「俺文系だから関係ねえわ」
アーティスト中「その数式はなんか違くない？」
東進模試山「今日は部活行かない。」
鉄緑田「宿題終わんないよ～！」
高身長澤「え？この問題が解けるかだって？」
徹夜藤「ちょっと寝る～」
坂東「機械にやらせればいいんじゃないですか。」
はんだ付け野「じゃあ電卓を創ろう！」

概要

この製作物は名前の通り「電卓」です。
数字の入力、四則演算、結果の表示、すべてできるので間違いありません。
え？容量が 8bit だから 255 までの計算しかできないしマイナスも表示できないだって？
そんな電卓あるわけねえだろ！
と言いたいところですが、これも間違いありません。
仕組みの説明に移ります。

仕組み

0.電卓って何なの？

電卓とはすなわち CPU(Central Processing Unit)のことです。そして CPU とは「プログラムの通りに動く計算機」のことです。この文章では CPU の仕組みについて触れつつ、DENTAKU∞の仕組みを説明していきたいと思います。

1.まずは 1+1 から

1+1=10

安心してください。誤字ではありません。

これは「2 進法」で書いた 1+1=2 なのです。

普段使っている 10 進法は 9(1×9)まで一桁で表せて、

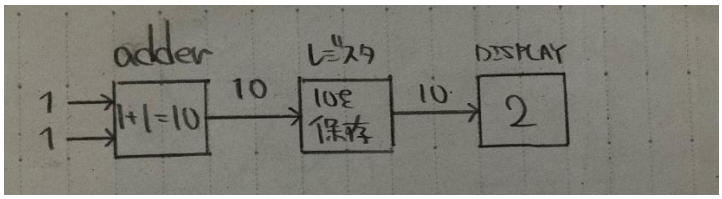
10(10×1+1×0)になると繰り上がって 2 桁となります。

それが 2 進法では 2 で繰り上がって、10(2×1+1×0)と表されます。

電気の世界には H(高い電圧)と L(低い電圧)の二つしかないので、2 進法を使って考えます。

2.回路っぽく

次はこれを回路っぽくしてみましょう。



ここでは 3 つだけ登場させます。

①レジスタ/一つの値を保存し、出力する。

②adder/二つの値を入力するとその和を出力する。

③DISPLAY/入力された値を表示する。ここでは 2 進法を 10 進法に変換。

見づらいので 2 進法で表記するのは一旦やめます。

存在は覚えておいてください。

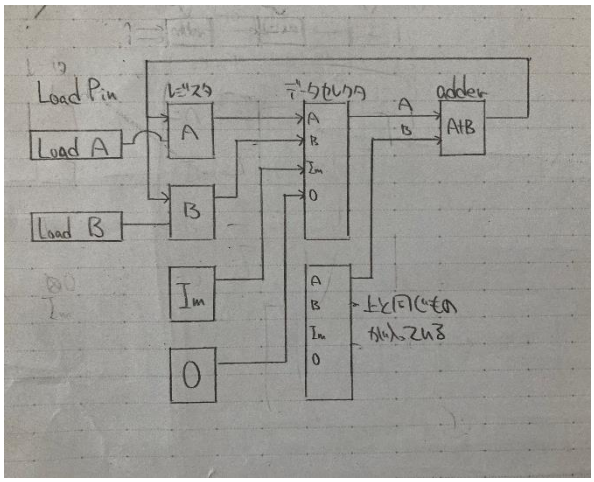
3.レジスタを活用しよう

実は上の図ではレジスタはいらないんですよね。

なぜなら全ての値が固定されているので、わざわざ保持する必要がないのです。

では adder の入力を自由に変えられるようにしましょう。

電卓とはそういうものですからね。



④データセレクタ/入力のうちから一つのデータを選んで
出力する。

⑤Im(immediate) data/プログラム上に記載する任意の値

⑥LoadPin/レジスタは入力された値を保存するかどうかを
この LoadPin の入力で決めることができる。
値を保存したいレジスタを選択できる。

つまりここではデータセレクタのおかげで adder に入力する値を選べるので、

A+A,A＋B,A+Im,A+0

B+B,B+Im,B+0

Im+0,0+0

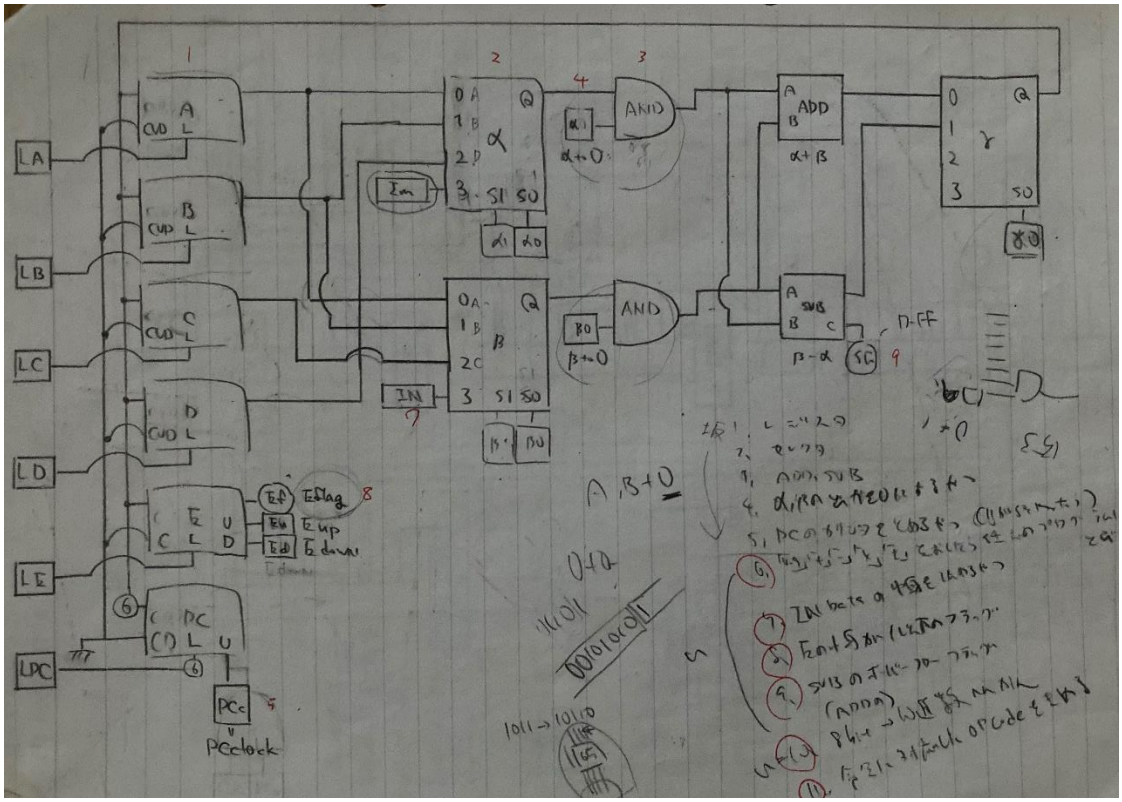
が可能なわけです。

これは、A+B ができるようになっただけでなく、
A+0 をすることによって、A を B にコピーしたり、

Im+0 をして A,B の値を任意の値にしたり、
0+0 で 0 にしたりと、いろんなことができるようになります。
この「値の転送」が CPU の動作の根幹になるのです。

4.さっさと電卓の図を見せろ

CPU の根幹を説明したので、電卓の説明に入ります。

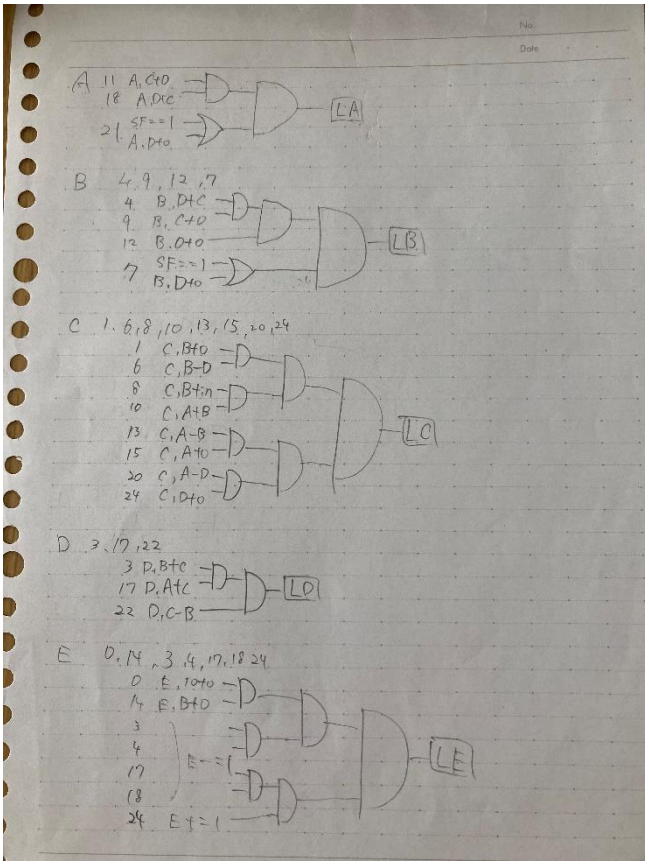


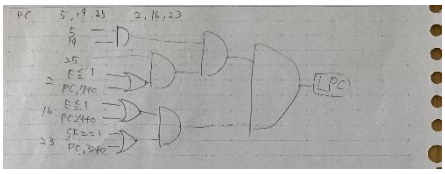
これが電卓の図です。
右から、
　LoadPin,レジスタ,データセレクタ,DATAto0,加算器と減算機,データセレクタ
です。
　⑦DATAto0/データセレクタに入れたい値が多すぎたため作成した、データを選んだあとで 0 に上書きするもの。
　⑧PC（プログラムカウンタ）/この後出てくるプログラムを書き込むところ場所から任意の番号の命令を抜き出すためのもの。その番号を出力する。
　⑨IN（Indata）/テンキーから入力される値。０～９。
特に新しいものはありません。
減算機が増えたので、またもやデータセレクタで
足し算の結果と引き算の結果どちらを Load させるか
選んでいます。動きの説明の前に、レジスタの LoadPin について説明させてください。
レジスタは LoadPin が L の時、ずっと値を読み込んでしまいます。

なので、A+B を A に入れると、すぐに(A+B)+B になって、
(A+B+B)+B になって、、、と増えていってしまいます。
なので DENTAKU∞では
　A+B を C に保存
　C+0 を A に保存
　結果、A+B が A に保存される
という方法をとっています。

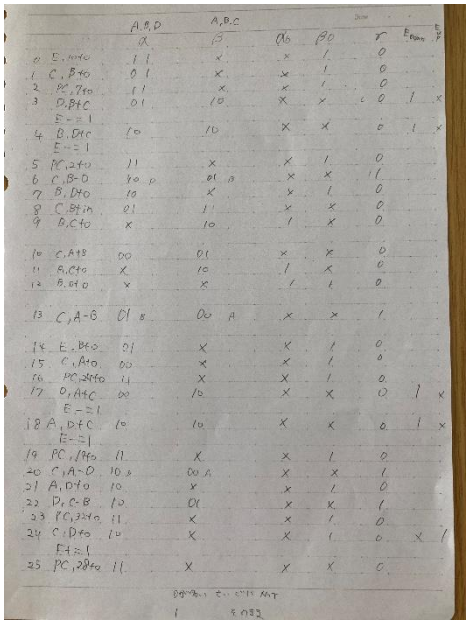
5.プログラム

A+B のやり方は前述した通りで、今度はそれをプログラムにする必要があります。
「A+B を C に入れる」という動作に OP(operation)CODE という命令番号を振ります。
プログラムは固定されたものなので、番号を二進法で表し、それをスイッチで代用します。これは読み込まれるだけの物なので、ROM(Read Only Memory)と呼ばれます。
また本体は LoadPin やデータセレクタで何番目を選ぶかの入力によって決まるので、ただの命令番号を本体用に分かりやすくする工程が必要です。それがデコーダです。
順番が逆ですが、デコーダ用の紙を先に見せて、デコーダの説明を省きます。必要な命令を pin ごとに抜き出してあげるということです。



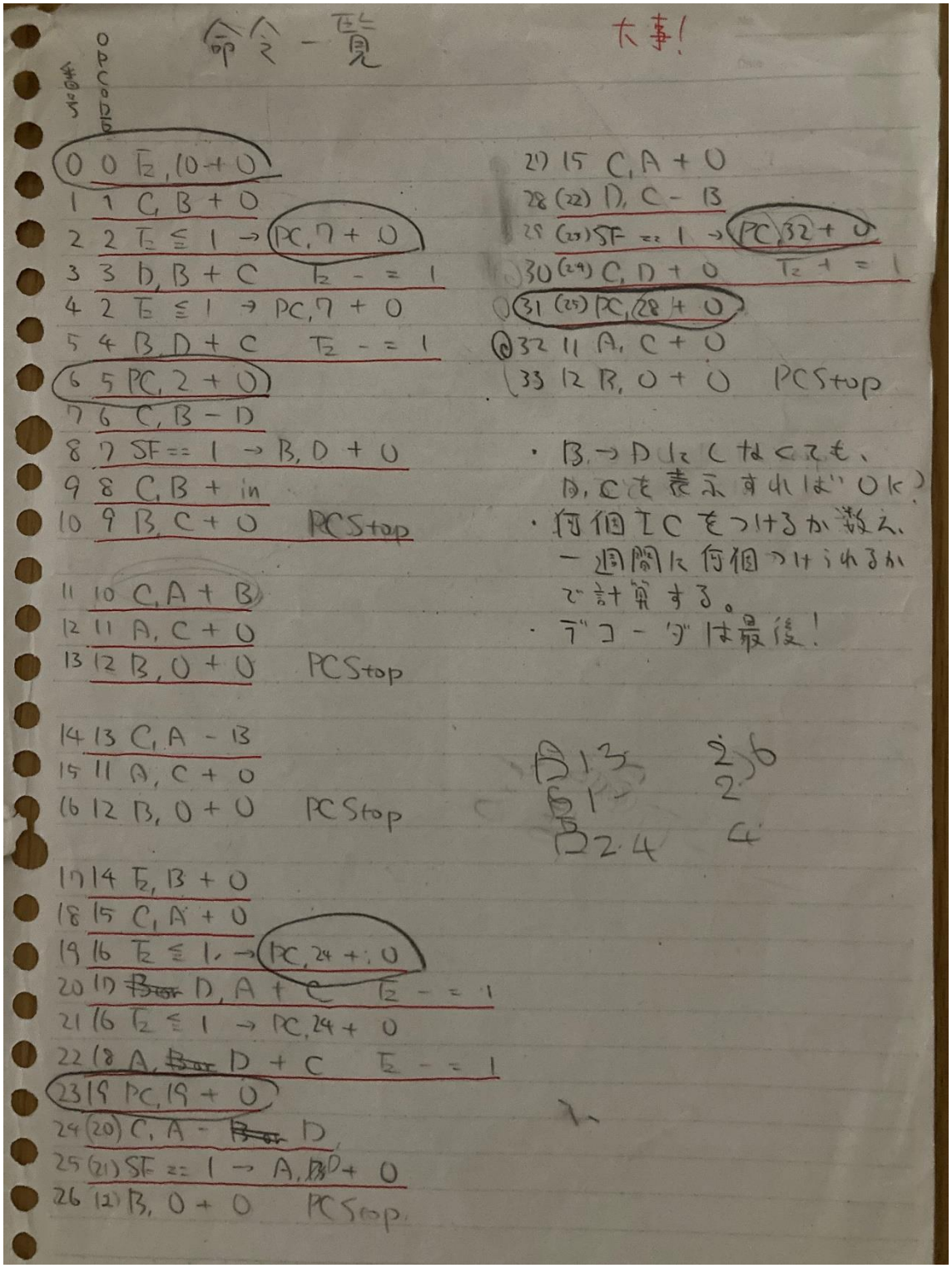


←↑LoadPin のデコーダ回路図



↑その他の Pin の命令ごとの処理

次は、プログラムを載せます。



番号０～１０がＢにテンキーで値を入力するプログラム、１１～１３が足し算、１４～１６が引き算、１７～２６が掛け算、２７～３３が割り算です。

足し算引き算は見てわかる通りで、

掛け算はまずA×BのBを別のレジスタEに入れ、AをCに入れます。A+CをDに、D+CをAに、というのを繰り返してそのたびにEを減らし、残り一になったら止めています。プログラムに書いてある、E<=1というのはE<=1だったら、DとAのどちらが大きいかを引き算で生じるSF==1（引き算の繰り下がり）を判定し、最終的なAの値を決めています。

割り算はまずA÷BのAをCに入れます。C−BをDに、D+0をCに、というのを繰り返して、C−Bが0以下、つまりSF==1になったら繰り返しをやめてCをAに入れています。繰り返している間Eを増やし続け、商をかぞえて、Aには余りが入っているという結果になります。

7.ディスプレイ

ここは簡単で、表示したいレジスタの値をディスプレイ用のレジスタに入れ、その中身を一減らすごとにまた別のレジスタの値を一増やし、2進法から10進法への変換を行っています。これをしないと7セグで表示できないので、あまり難しくない回路にしました。

8.部品

今回使った部品も書いていきます。ICとは、いろんな機能の詰まったすごい箱のことです。知っていても知らなくても関係ないので紹介しませんでした。正しくはロジックICと言います。

IC たち

74HC193	レジスタ。Loadがめんどい元凶。実は一瞬Loadができて解決した。
74HC153	データセレクタ。4つから選べるが2桁しか入力できない。
74HC283	ADDER。こいつがあれば減算機も作れる。
74HC04,08,32	NOT,AND,OR。いつでも使う。特にAND。
74HC138	デコーダ。と言っても3bitを8つに分けるだけ。ROMで使う。
74HC74	1bitしか保存できないがいろんな機能付き。一瞬Loadの立役者。

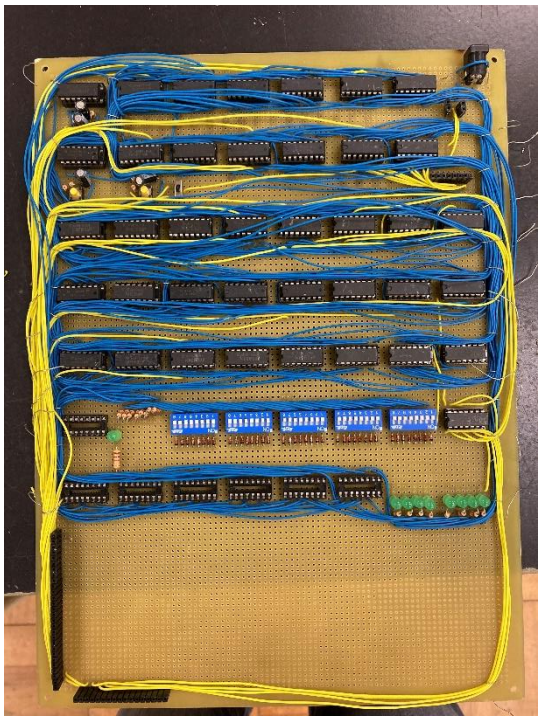
IC 以外たち

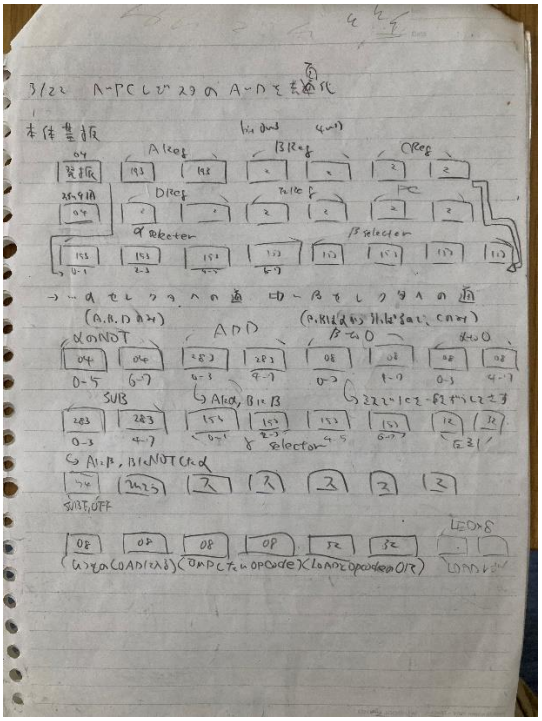
コンデンサ	スイッチと発振でしか使わない。実はICの隣につけたほうがいい。
抵抗	ROMのところに1Kを付けていたら１０Kだった。ミス。
配線材	青色→黄色→青色とサンドイッチになっている。
基板	昔のやつから切り取ることを覚えた。無駄な買い物はやめよう。
リレー	ROMはっきりより一瞬Loadに役立った。

9.終わりに

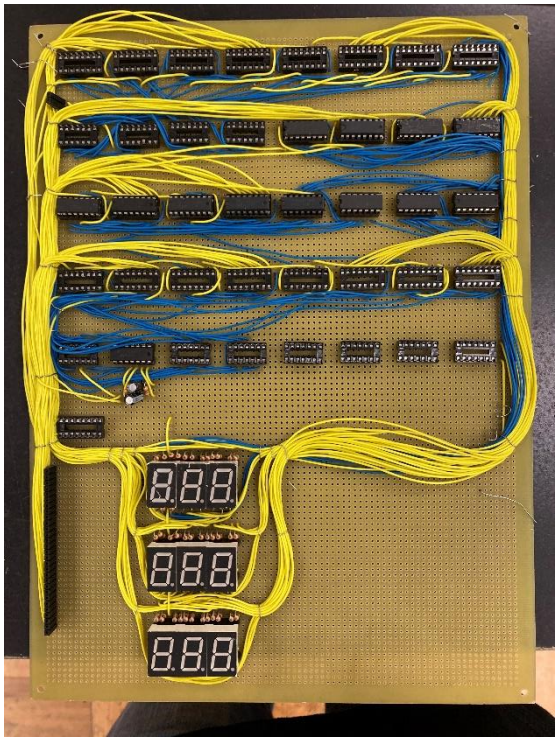
終盤は結構テキトーに説明してしまいましたが、詳しくCPUの機構を知りたい方は「CPUの創り方」という本をご購入下さい。僕もそれを参考にしました。CPUの根本をよく理解した後ならDENTAKU∞の仕組みもわかるはずです。

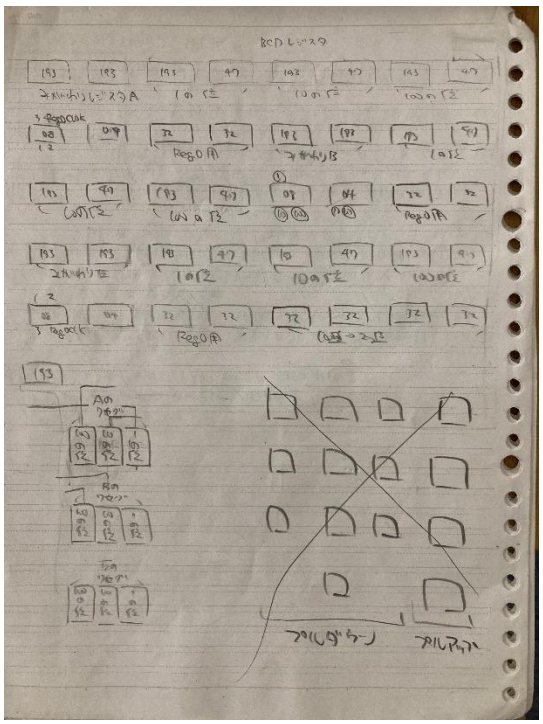
最後に外観を載せて終わりにしたいと思います。



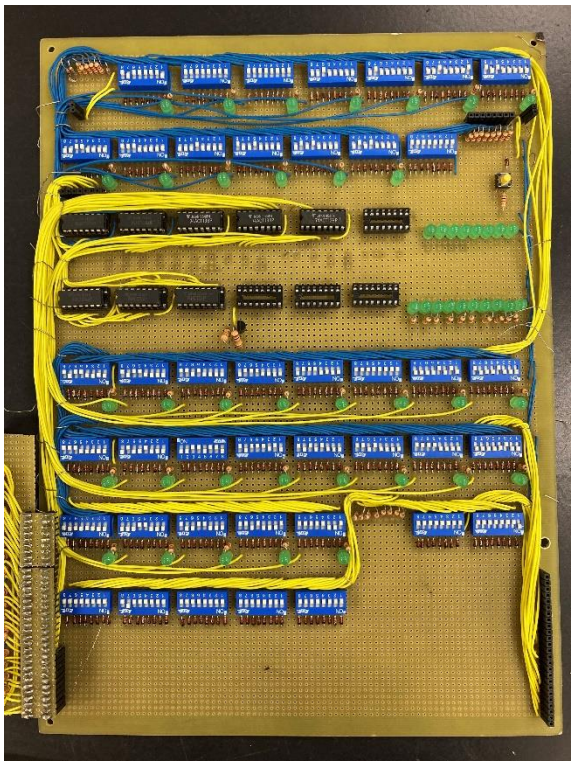


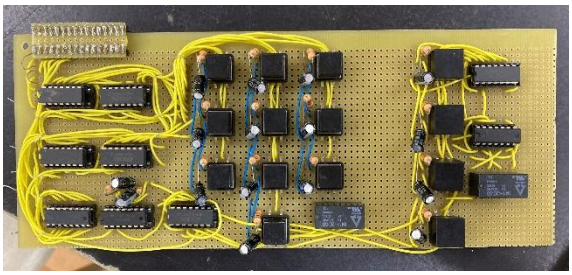
↑本体基板





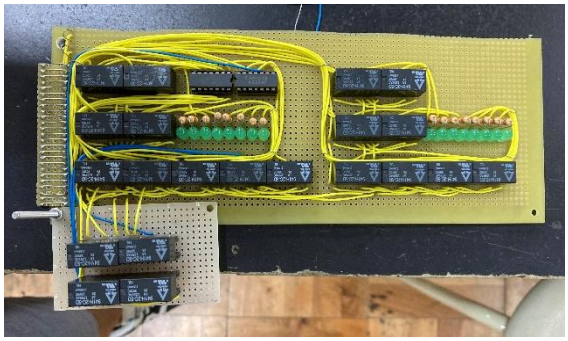
↑ディスプレイボード

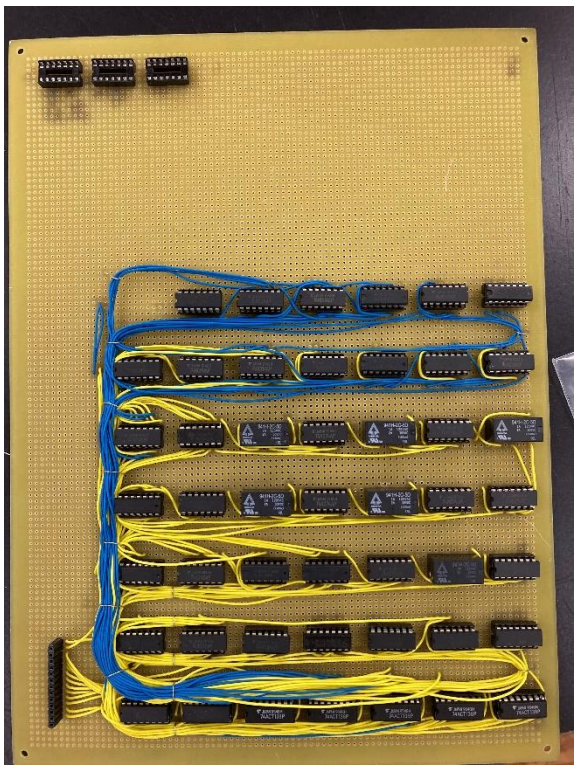


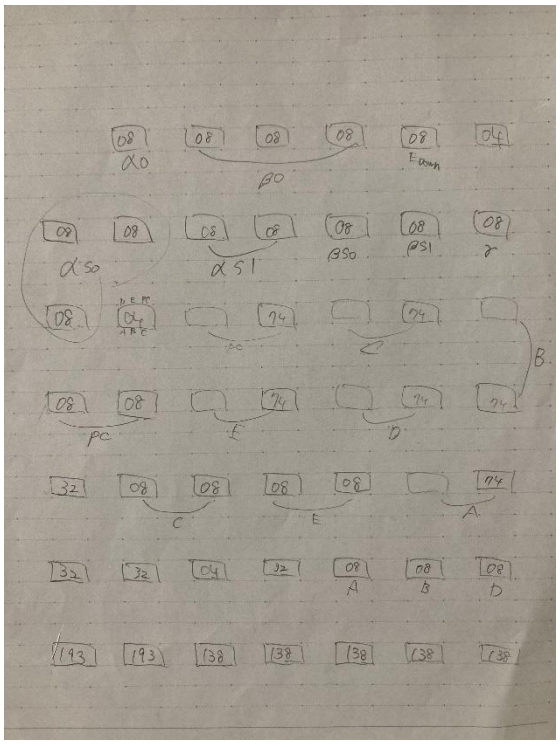


↑テンキー

←ROM ↓ROM はっきりリレー





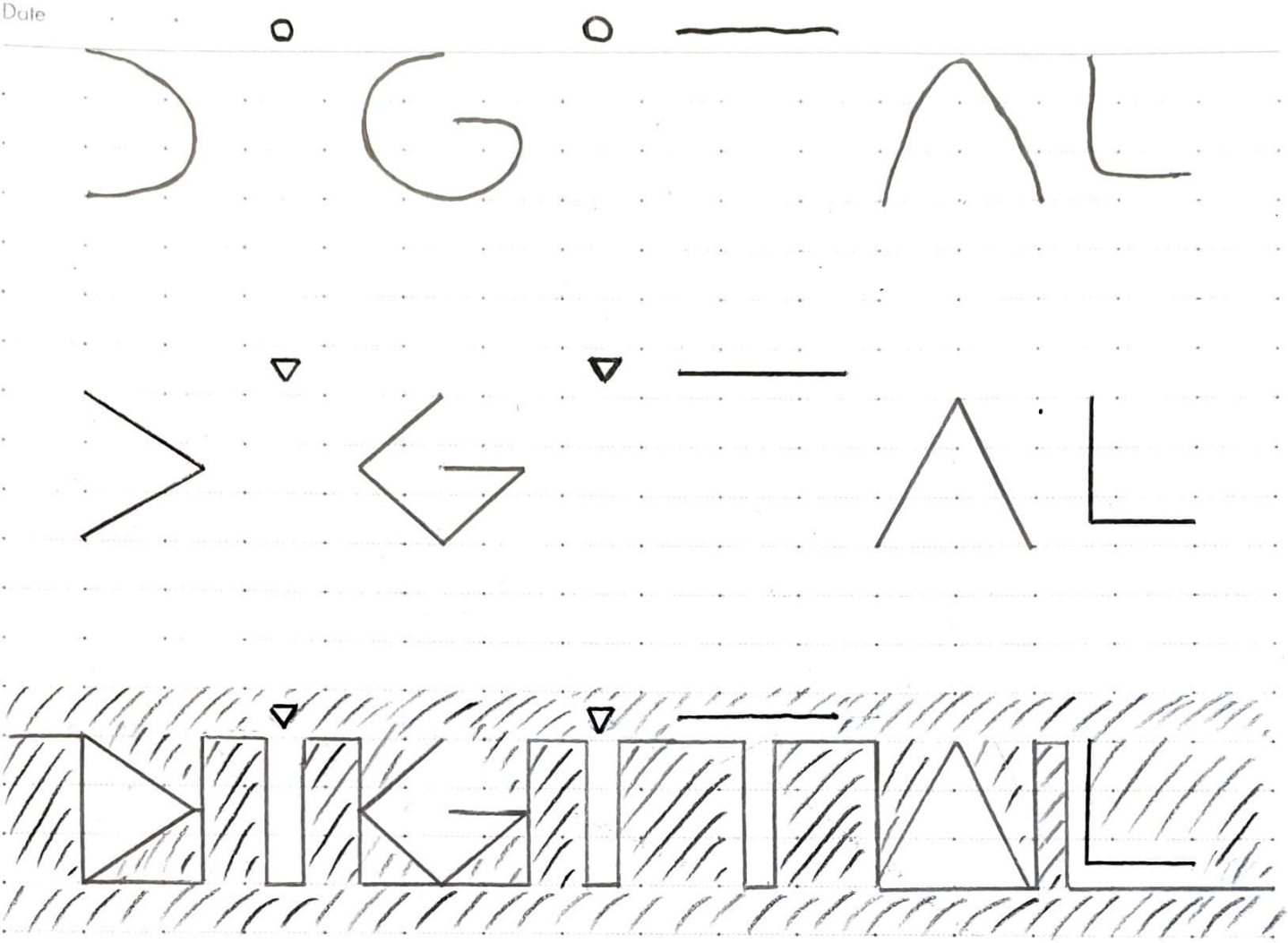


↑デコーダ

感想

坂東　仕組みを理解するのに時間がかかり苦労しました。売り物の制作もあり大部分を礒野さんに任せることになってしまったのは申し訳なかったです。

礒野　坂東くんのおかげで機構も本体製作もうまくできていると思います。最後の製作にふさわしいものが出来上がりそうでとてもうれしいです。



Welcome to the high and low beyond.

DOKI DOKI Physics Club!

製作者　H2　加藤
M3　山田
協力　物無の皆様

ストーリー

「さかな〜」
「ちんあなご〜」

「なのか〜」

執筆日は文化祭6日前です。提出がこんなに遅れて本当にすみませんでした。

概要

プロジェクションマッピング（嘘）です。
参加者が操作する物体をOpenCVを用いて検出し、プロジェクターから様々な演出を壁に写します。チ〇ムラボみたいなものです。
演出にはGLSLを、物体検出にはPythonを使用し、その他の処理や一部の演出にはTouchdesignerを使用しています。

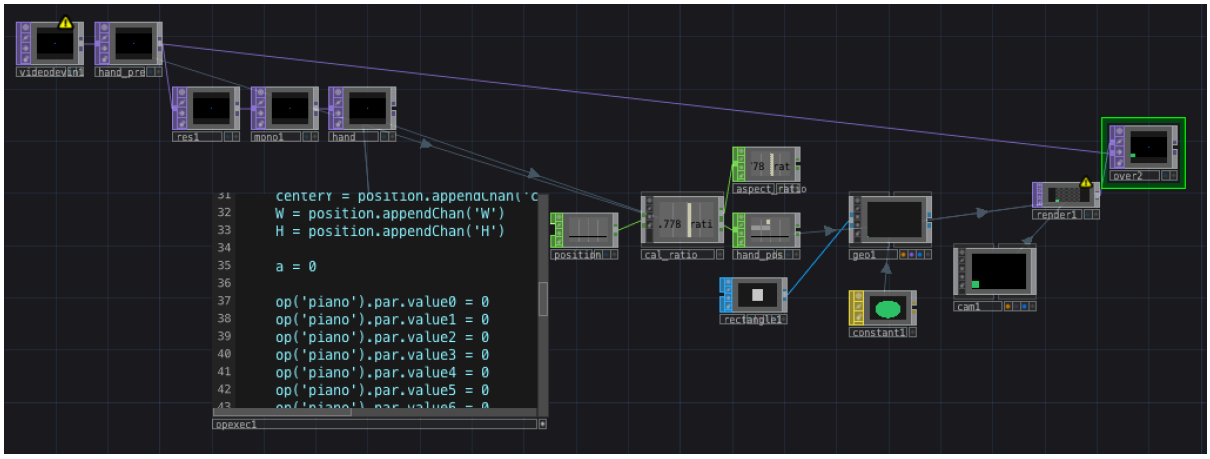
仕組み

1.物体検出

まずはTouchdesignerから。

Touchdesignerとは、主にノードの組み合わせで計算を行うジェネレティブ・プログラミングです。

ビジュアル系に特化しており、さまざまな映像作品に用いられています。



今回は上記の画像のようにプログラムしました。

左側では入力された画像を軽量化した上でPythonへ出力しています。

右側はPythonから出力された物体の座標を元にオブジェクトを設置しています。

Pythonの説明に入ります。

次ページのコードを使用しています。

入力された映像の情報を配列として都合の良い形に変換し、それを用いて輪郭検出を行い物体の座標・高さ・横幅をpositionという名前のCHOPに入力しています。

輪郭検出とは、入力された画像に対し、一定の明るさを持つ部分を白、それ以外の箇所は黒とすることで画像を二値化する処理のことです。

今回はある程度の大きさを持った白い部分の中心座標を取得しています。

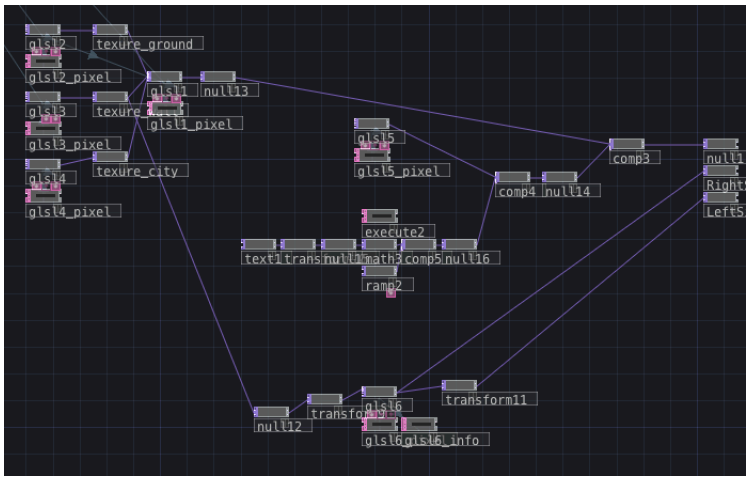
```
import cv2
import numpy as np

def onPostCook(changeOp):

	image = changeOp.numpyArray(delayed=True)
	image = np.array(image)
	image = np.uint8(image * 255.0)
	image = np.flipud(image)
	image = cv2.bitwise_not(image)
	image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
	ret, binary = cv2.threshold(image, 70, 255,cv2.THRESH_BINARY_INV)
	contours, hierarchy = cv2.findContours(binary, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_NONE)

	area_num = 1000
	hands = list(filter(lambda x: cv2.contourArea(x) > area_num, contours))

	resX = op(hand).width
	resY = op(hand).height

	position = op(position)
	position.clear()
	position.numSamples = len(hands)
	centerX = position.appendChan(centerX)
	centerY = position.appendChan(centerY)
	W = position.appendChan(W)
	H = position.appendChan(H)
	a = 0
	for i in hands:
		position.numSamples = len(hands)
		hand = cv2.boundingRect(i)
		W[a] = hand[2]
		H[a] = hand[3]

		centerX[a] = hand[0] + hand[2]/2
		centerY[a] = -(hand[1] + hand[3]/2) + resY

		a = a + 1
		if a == len(hands):
			break
	pass
	return
```

2.演出

演出は複数作りましたが、その中の一つであるトンネルの映像について説明します。

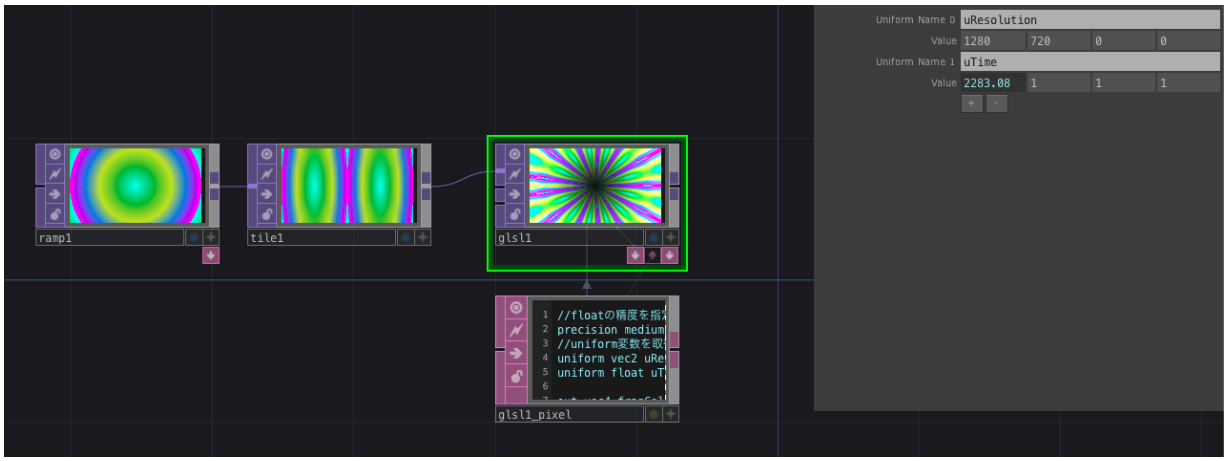
この映像はTouchDesignerのGLSL TOPを使って作っています。GLSLとはシェーダー言語の一つで、OpenGL Shading Languageの略称です。その特徴をざっくり説明すると、GPUを操作できるということです。したがって大量の計算を同時に行うことができるので、特に3Dや2Dのグラフィックを描くのに適しています。ここら辺で、詳しい説明はネットの記事に任せて本題のトンネルに入っていきましょう。

TouchDesignerのオペレータは下図のようになっています。ごちゃごちゃしていて分かりづらいですね。ごめんなさい。



⇦左上がメインの部分、真ん中はおまけ的な感じでつけた部分で、一番右が出力（正面、左右）となっています。

ここでは画面の中心へ向かって進んでいくような映像の作り方を載せようと思います。テクスチャはどんな画像でもいいですが、今回はRamp TOPとTile TOPを組み合わせて適当な画像にしています。



GLSL TOPの設定として、GLSLタブのInput Extend Mode UVをRepeatに、Vectorsタブの Uniform NameにuResolution、uTimeを追加してください。今回の場合、uResolutionの値は上図の通りに、uTimeの値は　absTime.frame * 0.01　にしています。

次のページにGLSLのコードを載せます。多少の三角関数などの知識があれば、コメントとどっちがメインなのか分からないくらい短いコードで書けてしまいます。製作では、これをもっといじったり経過時間によってテクスチャの画像を変化させたりしてました。

```
//floatの精度を指定。lowp, mediump, highpの3種類がある。
precision mediump float;
//uniform変数を取得。
uniform vec2 uResolution;//1280 × 720ピクセル
uniform float uTime;//ここではTouchDesignerを起動してからのフレーム単位の時間

out vec4 fragColor;
void main(){
	//座標を正規化。図1を参照。
	vec2 p = (gl_FragCoord.xy * 2.0 - uResolution.xy) / uResolution.y;
	//出力される色。
	vec3 destColor = vec3(0.0, 0.0, 0.0);

	//アークタンジェントを計算。図2を参照。
	float a = atan(p.y, p.x);
	//各点の原点からの距離を求め,そこに時間を引いている。
	  進んでいるように見せているのはココ。
	float r = length(p) - 5. * uTime;

	//各点を(a,r), つまり極座標で表した面にテクスチャを貼り付けている。
	destColor = texture(sTD2DInputs[0], vec2(a, r)).rgb;

	//原点からの距離を掛けることで、奥(中心)に行くほど黒っぽくなるように見せている。
	float shadow = length(p) + 0.1;
	destColor *= shadow;

	vec4 color = vec4(destColor, 1.0);
	fragColor = TDOutputSwizzle(color);
}
```

図1

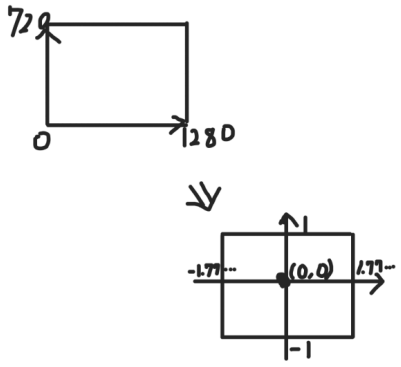
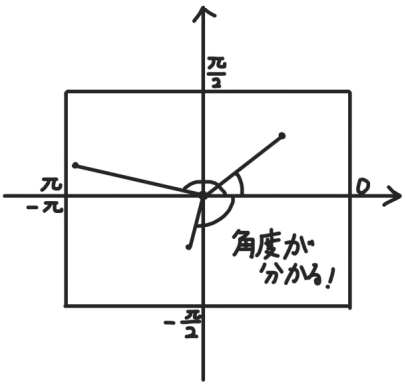


図2



感想

山田：仕組みの理解や行動を起こす事にかなりの時間がかかってしまいました。来年の製作はやるべき事をしっかり捉え、勉強や試行錯誤を増やしていきたいと思います。

加藤：楽しかったです。しかし一年間に計画が二転三転してしまい、山田くんにとても迷惑をかけてしまいました。それでもついてきてくれた彼には本当に感謝しています。

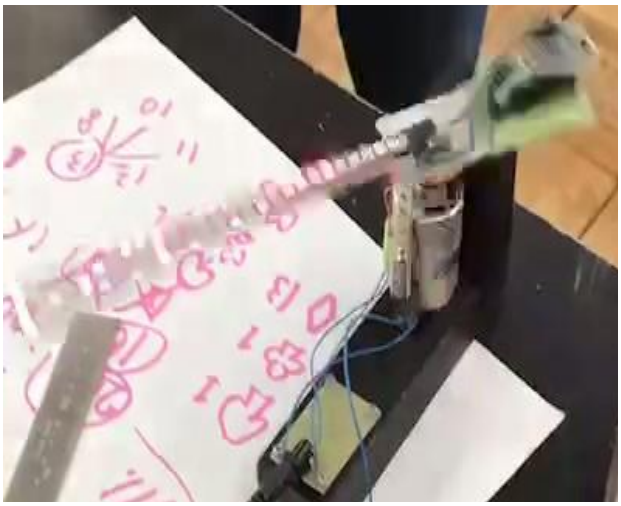
バーサライタ

制作:H1 鳥内　M3 石村

ストーリー

T内「夏暑すぎ～！こんな暑い中部活なんて行けるか！家で冷房効かせてFPSするぞ！」
I村（暑さで頭おかしくなったか？~~元からか~~。こうなったら部活に来させるためになんか涼めるものを…そうだ！モーターに板をつけて…)「T内さん！扇風機作ったんでこれ使って部活来てくださいよ！」
T内「いや、扇風機くらいじゃ行く気にはならないな～w」
I村「ただの扇風機な訳ないじゃないですか。時代はゲーミング、1680万色に光りますよ」
T内「行〝ぎたいっ!!!!…!!!」

外観



概要

LEDがついた棒をモーターで回転させ、LEDの残像を利用して空間に画像や文字を表示するものです。

使った部品

・マブチモーターのモーター　RS-540SH
・Wavesのワイヤレス給電モジュール 12V→5V 1A arduinoとLEDの給電に使用
・木材　30mm×40mm
・BTF-LIGHTNINGのLEDテープ　WS2812B
・arduino nano
・L字金具　モーター固定用
・コの字金具　ワイヤレス給電モジュール固定用

原理

LED28個を回転部分についているarduinoで制御し、回転に合わせて表示するLEDを変えることで様々なものを表示しています。arduinoとLEDテープにはワイヤレス給電モジュールを使って給電していて、LEDの制御にはAdafruit NeoPixel Libraryを使用しています。

感想

鳥内
去年の入り電の制作でLEDの制御に興味を持ち、それがきっかけで始めた制作でしたが、想像以上の出来になりいい経験になりました。来年もこの経験を制作に活かしていきたいです。
石村
一年間でバーサライタの
ハードとプログラムの一部を
制作しましたが、制作に消極的で、結果として共同製作者さんの鳥内さんにご迷惑をお掛けしてしまいました。これが反省点です。
良かった点は、製作物が無事動作したこと、arduinoをそれなりに使えるようになったこと、そして楽しく制作できたことです。

スマートグラス

製作：高２畠山　高２高梨　中3角野

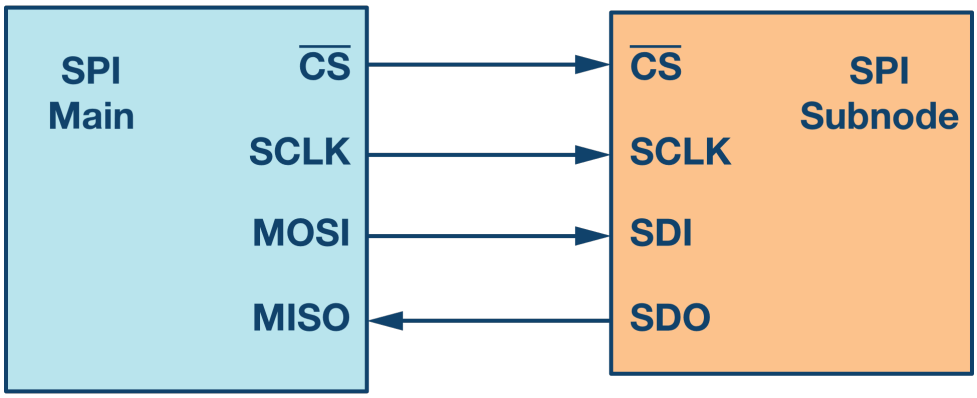
ストーリー：時は令和 世はIT戦国時代 物無国内では横暴な総務が圧政を敷いていた。民が虐げられる様子に耐えられなくなった流浪の侍は自ら製作した”うぇあらぶるでばいす”を身に着けて今、立ち上がる…

お頼み申すお頼み申す
卒爾ながら冥府は何処
外道が笑えば民が泣く
諸行に埒なき沙汰あれば
魔道の始末を請け負いいたす
己は尾籠の三一侍
士道死花無用の所存

概要：
ストーリーは最後のやつ書きたかっただけです。
眼鏡のレンズ部分が液晶になっており、それを使って現在時刻などの情報を確認することが出来ます。またカメラやボタンを使って数式を読み取って答えを確認したり音楽を再生したりすることが可能です。AndroidとHTTP通信で接続して時刻を合わせるといった機能もあります。

仕組み：
眼鏡のレンズ部分に透明なOLEDモジュール(以下液晶)をつけて、それをフレーム部分にある小型のRaspberry Pi(以下ラズパイ)で制御しています。
OLEDとは有機ELディスプレイのことで一般的な液晶より薄型で軽量だという特徴があるのでこの製作のような用途に適しています。
液晶とラズパイ間ではSPI通信を使っています。一般的なSPI通信ではメインノード(送信側)とサブノード(受信側)の二つのデバイス分かれていて、またCS(チップセレクト),SCLK(クロック),MOSI(メインインサブアウト),MISO(略)の4つのピンを使用します。この製作の場合はラズパイがメインノードで液晶がサブノードとなります。仕組みはSCLKピンで相互のクロックを同期させて立ち下がりごとにMOSI/MISOから1bitずつデータを送受信して通信するという感じです。
簡単にまとめると電圧の上がり下がりに合わせて少しずつデータをやり取り

しているといったところでしょうか。何かの参考になれば幸いです。



SPI通信の概要

カメラで読み取った数式を解いたりするなどの機能はAPI（自身のプログラムで外部のサービスを使えるインターフェース）を利用しました。
Androidとラズパイを接続するアプリはAndroid Studioというソフトを使って製作しています。
感想：
畠山　Pythonのコードを担当した畠山です。予期せぬ不具合に見舞われたこともありましたが班員の助けもあって、これが製本されているころにはこの製作は完成しているはずです。一年間ありがとうございました。
高梨　液晶とラズパイをSPI通信で繋ぐこととAndroid関係を担当しました。言い出しっぺにもかかわらず出席率が低かった僕に代わってしっかり班をまとめて主導してくれた畠山に感謝しかありません。また短期間でPythonを学習して実践できるようになった角野も物無員として素晴らしい人材だと思います。1年間ありがとうございました。
角野　初めて先輩方との共同制作をして、一からPythonのプログラミングを教えてもらい、最後には自分一人でもある程度コードを書けるようになれたのでよかったです。

ゲーム

げーむ

げ ー む

ゲーム

笑えなくなった戦車ゲー

設計者　谷中

~Story~

T中は老いた。T中の中にあったのは空虚な”DAIGAKU・ZYUKEN”という名のものだけであり、去年までのような大掛かりな兵器（ロボット展示）をもう長く作っていなかった。しかし、物理部員の心の拠り所である”SEISAKU”だけは欠かさなかった。だが、大きな事件がT中を襲うこととなる...

T中「ことしはどんなロボットにしようかなーーー！！！」

不敬罪済発言を繰り返すタイプの後輩・T内
「T中さんまだそんなもの作ってたんですかwww」

問題発言も辞さないタイプの後輩・Y田T士郎
「時代はUnityですよ、T中さんパソコンも使えないんすかwww
しょうがないっすよね、先の時代の”敗北者”なんですからwww」

T中「ハァハァ、”敗北者”...?　取り消せよ、今の言葉...!!!やってやろうじゃねえか！」

「老害の意地みせたるわぁ！！！」

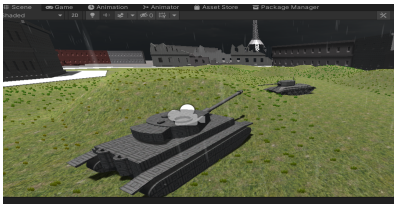
こうしてロボットしか作ってこなかったT中は初のUnity製ゲームを作ることとなった...!!!

仕組み

Unityでゲームを作ります。WotとWarThunderの中間くらいのを目指してます。
これを書いている現時点（9/1) では完成度は4割程度です。
戦車の動きが完成し、マップが4割程度完成しています。
さーてどうやってつくっているのかなー？行ってみよう！（やけくそ深夜テンション）

使用ソフト：みんなのアイドルUnity、VisualStudio

本体（自機）
モデリング：Fusion360



物無員の中でよく使われるモデリングソフト。本当は精密なものを作るのがウリであり、3Dゲームのキャラを作るならBlenderを使っている人のほうが圧倒的に多いが、僕はこれしか使えないのでやむなく使用している。ぴえん。

足回り：WheelCollider
優秀な当たり判定。もとは車に使うものだが、戦車のサスペンションとして採用。めちゃくちゃいい動きをしてくれる有能な奴。でも扱うのは簡単ではないツンデレ、使いこなせれば環境トップ。みんなにぜひ使ってもらいたい。ちなみにかなりざっくり目で、アセットストア産に比べたらお粗末かもしれない。ちなみに発進するときに少し沈み込むところが大好き（急な癖語り）

砲塔回転：RotateTowards
指定したオブジェクトの方向に指定した速度で回っていく。これをカメラの向いている方向Vectorに向けることで”戦車乗ってますよ”感を出している。マウスのX、Yの動きでカメラが回るのでキーマウ勢みたいで強そう。

結構詳しく書いたつもりですが、疑問に思いましたことでしょう。無限軌道がないやないかと。許してください。僕には物理エンジンで履帯を回すことは叶いませんでした。その代わりに、アニメーションやエフェクトなどでごまかしてます。皆さん、完成度が高い自機をご所望であれば、「PhisicsTank」と調べてください。すごい完成度が高い戦車が見れますよ。

MAP
Unityで作りました。イメージはWotのパリです。洋風な建築をイメージして作ってます。
オブジェクト類は無料のアセットやFusion360で作成しています。お気に入りは噴水です。

敵

強化学習AI

左図：コマンドプロンプト

ここからは強化学習のお時間です。観察・行動・報酬の3条件を与えて学習することで推論モデル・Brainを作成して動かします。めっちゃ複雑ですが、完成したAIの挙動を見るのもまた一興なのです。現時点では完成していませんが、相手が動いてるだけで感動もの。ML-Agentsで調べてみてください。解説しているサイトや動画もありますので、ぜひやってみてください。

（余談：Pytorchというものを使用しますが、僕は読みを間違えていました。）

~Final Story~

こうしてゲームの説明を終えたT中。しかし最期の瞬間は近づいていた。

T中「おれの最期も迫ってる...今から言うことを物無員のみんなに伝えてくれ...」

「俺の物無人生に悔いはない！次期総務会計！あんまり知らない顧問！そして物理大実験室！」

「愛してくれて.....ありがとう!!!」

感想

最期までご覧いただきありがとうございます。初のUnity製作ということもあり、慣れないことの連続でしたが、同輩や後輩たちの助けで無事にここにいます。

タイトルの「笑えなくなった」ですが、これを作ろうと製作を始めた矢先にどっかのロツア（伏字）が戦争をおっ始めやがったからです。全然笑えません。理由があろうと人を殺すことを大義にするのはよくないです。このゲームは人を殺す道具としての戦車を作りたくて始めたわけではありません。あんまりここで書くことでもないのですが、早く平穏な日々に戻ることを願っています。

最期までワンピースネタで通しましたが、まだFilm Red見てないので早く見に行きたいです。

BUTSUMU KART

製作者：高一　永谷大成　中三　ウレマン麻人
協力：物無のみなさん

＜ストーリー＞

遂に物理部は念願の物無テーマパークを完動させたぞ！！！！しかし、あまりにも某Ｎ天堂のゲームに似すぎていたため、目をつけられてしまった！！アクセル全開にして著作権から逃げ切ろう！！

＜概要＞

Arduino と Unity のシリアル通信を使ったアーケードのレースゲームです。筐体は木材でできており、ボタンとハンドルとアクセルペダルで操作します。ブレーキはありません。僕らには必要ないので……

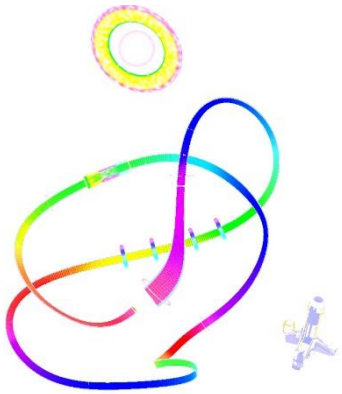
＜使った部品、ソフト＞

・Arduino UNO
・可変抵抗
・タクトスイッチ
・木材
・ディスプレイ
・３D プリンター
・Unity
・blender
・Arduino IDE
・血と汗と涙

＜原理＞

可変抵抗やタクトスイッチの信号を Arduino からのシリアル通信で Unity に送ってゲームを動かしています。私は真に驚くべきゲームのプログラムを書いたのですがこの余白はそれを描くには狭すぎるので省略します。代わりに面白そうなゲームのスクショを張っておきます。コースは四種類ありますが、最初に作ったレインボーロードが一番難易度が高くまた思い入れも深いコースになっています。デバッグをしているうちに自分がうまくなりすぎてしまい、ちょうどいい難易度のコースを作るのが難しかったです。

ミニマップ　→



＜感想＞

永谷

Arduino はめちゃくちゃ便利なので触るか迷っている人は今すぐやりましょう。最初は一人の予定だったのもありゲーム部分を大体作ってしまったので、ほとんどウレマンにプログラム部分をやらせてあげられなかったのが唯一の心残りです。一方モデリングはたくさんやってくれたのでそこはうれしかったです。

ウレマン

最初のうちは Unity をよく利用していましたが、夏休みごろからずっと blender と格闘し、最終的に木材を切る仕事をしていたので何を作っているのか自分でも分からなくなりそうでした。

マックス

「楽しかったです。」

物無員「誰！？」

To be continued…

BUTUNITHM

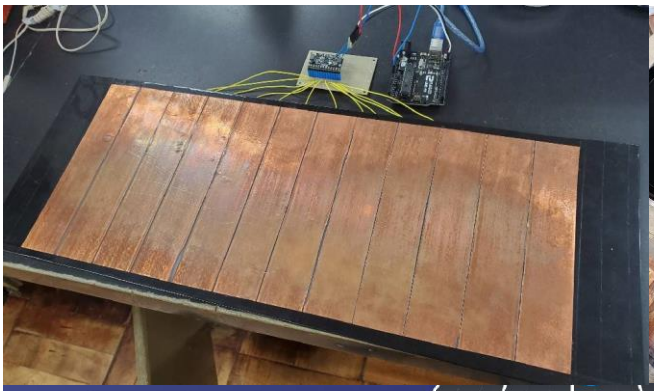
H1 大谷 心雲、榎本 亘

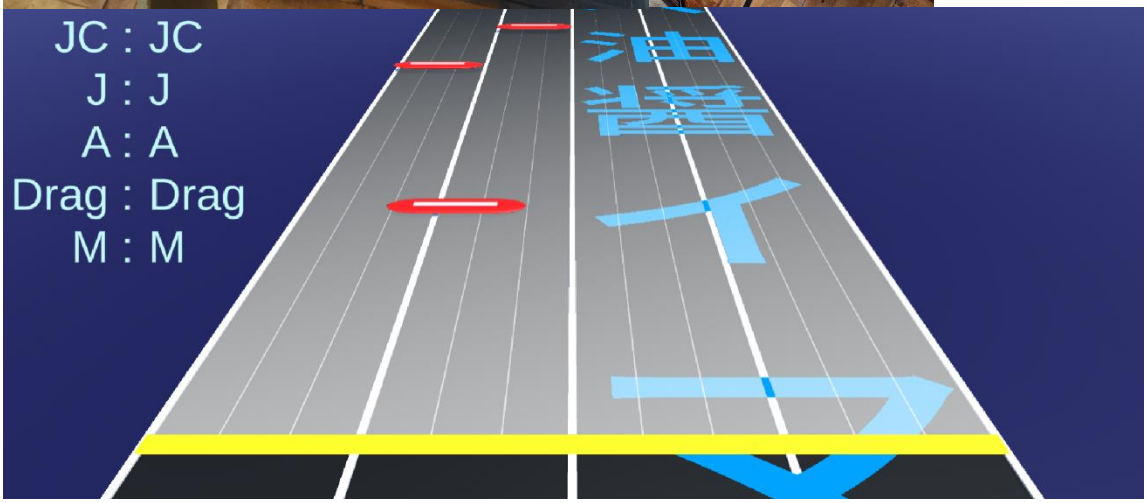
STORY

あたいの名前は『ヒナタ』！ゲーム制作を担当する物無員の一人で、麻布生に最高の楽しみを与えるために、毎日頑張ってんのよ！ありがたく思ってよね、麻布生！
――あたいは考えた。『最高の娯楽を生み出すゲームとは何か？』そして1つの回答を導き出したのよ。
「麻布生にとって最高の娯楽を生み出すゲーム……それは『リズムゲーム』でしょ！
――あたいの自信作 BUTUNITHM！でもなかなか人気が出ないの！調べてみたら原因はすぐにわかったわ、なんと麻布のライバルサークル、パ〇研が別のリズムゲームを展示してるみたい！
でもあたいは諦めない。パ〇研になんか負けないんだからっ！

――「行くわよっ！FREEDOM DRIVE！」

OVERVIEW





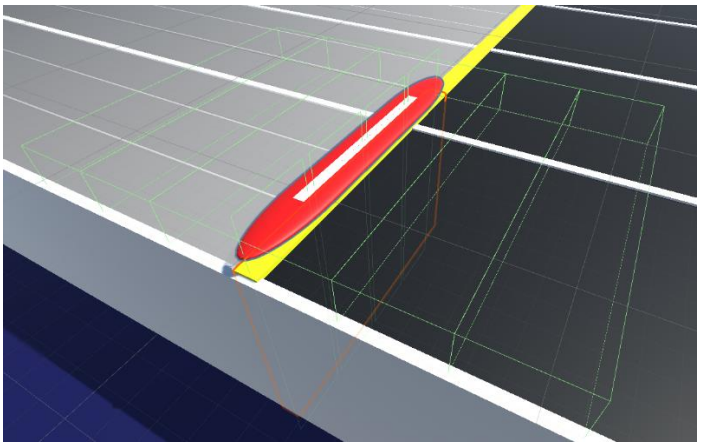
PARTS

・MPR121（静電容量センサ）
・Arduino
・銅箔テープ

SOFTWARES

・Unity（物理エンジン）
・C#（プログラミング言語）

PRINCIPLES



左の画像のように、ノーツに細分化した当たり判定を付け、判定線と呼ばれる線に重なっていることを検知。

また、MPR121 センサで触れたかどうかを検知し、Arduino からシリアル通信で Unity に送る。

当たり判定が検知されているときに Arduino から信号を受けとると判定し、ノーツを削除するというシステム。そのときに重なっていた判定の種類によって判定を5つに細分化している。

ノーツの出現タイミングは Unity のタイムラインという機能を使って、曲に合わせてノーツが降ってくるように調整している。

IMPRESSIONS

大谷

Unity でのゲーム制作についてのノウハウが全くない状態からこの音ゲー制作にとりかかったため、ものすごく苦労しましたね。そもそもプログラミングをちゃんと勉強したのは今年が初めてですし。なので製作期間の最初3,4か月はほぼ Unity の使い方について調べて終わりました。まぁ今では「Unity 使える」といえる程度には知識つきました。その後も今日まで Google で音ゲーの作り方などたくさん調べて、やっとこの形に収まりました。もともとはゲームセンターに置いてある某チュウ〇ズムというゲームを再現したいという思いで始めた制作ですが、いい感じになったんじゃないのかな。でも個人的には 80％くらいの満足度です。まぁやはり妥協した点が多いんですよね。例えばエアーという、腕を振り上げるシステムは導入できていませんし。しかし残念には思いません。何故って？来年の文化祭でもこのゲームを展示する予定だからです。もっともっと改良してさらに良いゲームとして仕上げて来年の文化祭で展示して見せます。乞うご期待！

榎本

自分でもシステムを作ろうとしたんですけど上手くいかなくて、結局大谷にシステム任せちゃうことになってしまったので、大谷には感謝。 モチベが湧かなかったり生活習慣が終わったりで部活に来なかったりするタイミングはあったけどなんだかんだ部活は楽しくやれたと個人的には思ってます。 音ゲーをするのは好きだったのでこの製作が出来て良かったし、Unity についてもある程度知ることが出来たのでそこは良かったかなと。 ただ、やっぱり迷惑かけちゃったと思ってるので来年の製作では誰にも負けぬくらいのもの作り上げてこうと思ってます。乞うご期待。

格闘ゲーム

M３　大和田
H２　山田

ストーリー

皆さんこんにちは
「ゆっくり霊夢と」
「ゆっくり魔理沙だぜ」
「今回は物無員がつくったとある格闘ゲームについて解説していくよ」
「それではいくよおおー」

概要

このゲームは、二人のプレイヤーがそれぞれキャラクターを選択し、戦う格闘ゲームです。制限時間内に敵を倒す、もしくはゲーム終了時により多くの体力が残っていることで勝利となります。攻撃判定がキャラの端にしかないため、市販されているゲームに例えるとすればスマブラよりもストーリートファイターに近い仕様です。

使用したもの

Unity（2019年度版）　ゲームの主要な処理を行うゲームエンジンと呼ばれるソフトです。ゲームのステージ、UIの製作に使用しました。
Microsoft　Visual　Studio　プログラムを記述するために使用しました。言語はＣ＃です。Unityと連携させてゲームを制御します。
Arduino　コントローラーとunityを接続するために使用しました。
Joyスティック　レバーです。上下左右の入力、複合では地方港への入力ができます。
スイッチ　ボタンです。筐体に使用しました。

ゲームの構成

このゲームの構成は、コントローラーとゲームプログラムの2要素によって構成されています。これから、それぞれについてより具体的な説明をしていきます。
・コントローラー
①　ゲームコントローラー
手で持つタイプのコントローラーです。これは元からunityに対応しているので割愛します。
②　筐体のコントローラー
筐体に埋め込んであるコントローラーです。これはボタンの入力をarduinoで認識し、そのデータをシリアル通信でunityに送って動かしています。
③　シリアル通信
シリアル通信とは、伝送路上を1度に1ビットずつデータを送るという通信方法で、この制

作の場合arduinoがシリアル通信を開始し、データを書き込み、それをunityが読み取るという形で通信しています。
以下arduinoの該当プログラムです。

```
void setup{
 Serial.begin.(9600)
 }
 void loop{
   Serial.print(“A”)
   Serial.println(“B”)
}
```

Serial.beginにより、シリアル通信を開始しています。その後の9600は、シリアル通信で1秒間に送れるデータのビット数を表しています。この場合、1秒間に9600ビットまでの通信ができます。
Serial.printは、シリアル通信で、データを書き込むメゾッドで、この場合Aと入力されます。また、Serial.printlnは前者とあまり変わらず、送るデータを区切るか区切らないかという違いがあります。
この場合、このプログラムではAB,AB,AB,ABというように送信されます。

以下unityの対象のc＃スクリプトです。

```
SerialPort serialPort;
String data;
void Awake()
{
    serialPort = new SerialPort(”COM3”,9600);
      serialPort.Open;
}
void update()
{
    data=serialPort.ReadLine;
}
```

SerialPortとは、unityでシリアル通信をする際に使用するメゾッドです。
new　SerialPort(“COM3”,9600)で、今後使うシリアル通信を宣言しています。この場合、COM3という名前で9600ビットまでの通信規格です。
serialPort.Open、serialPort.ReadLineはそれぞれシリアル通信を開く、シリアル通信を読み取る、という意味になります。
④

・ゲームプログラム

①　キャラクター選択画面
キャラクターを選択し、選択したキャラクターをゲーム上に反映させるプログラムです。
キャラクターそれぞれに番号を割り当てており、ボタンを押すとその値が保存されます。
以下該当プログラムです。

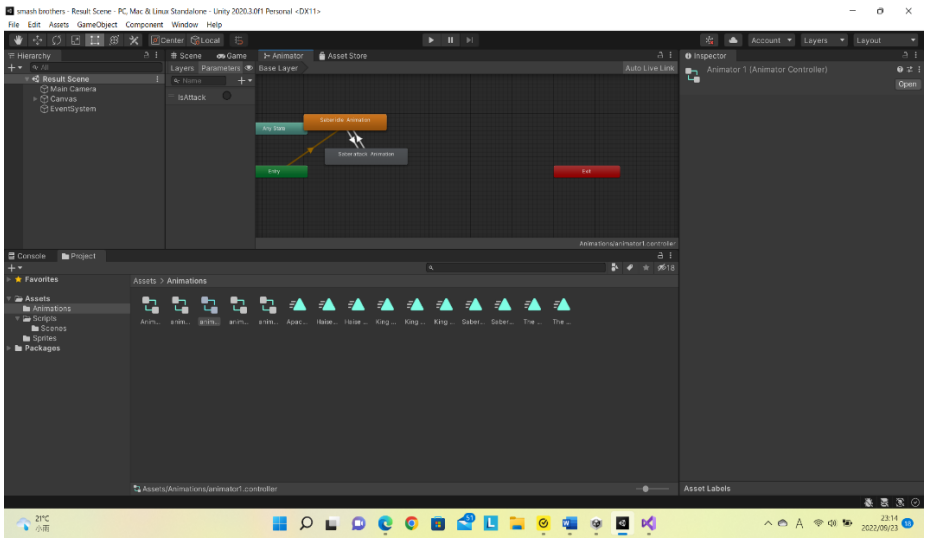
```
 public void OnClickA()
  {

     Debug.Log("キングクリムゾン");
     PlayerPrefs.SetInt("PlayerA", 0);
```

}
このPlayerprefs.SetIntというメソッドを使用することにより、値をキーと共に保存することが出来ます。そして、保存した値を次のシーンのスクリプトに渡します。

② ゲーム画面
ゲーム画面では、選択したキャラクターをそれぞれゲーム上に出現させるスクリプトを記述しています。ゲームスクリプトのPlayerprefs.getIntによって先ほど保存した値を取得することが出来ます。If文によって、取得した値ごとに異なるキャラクターのスクリプトに変移できるようにしています（以下一部を抜粋）。SetActiveによってUnityのinspectorを制御し、選択したキャラクターを画面上に出現させます。

```
if (PlayerA == 0)
    {

        Debug.Log("Aのキャラはキングクリムゾン");
        Kingcrimson.SetActive(true);
        int hp0 = CharacterSelect.Ahp;
        if (hp0 == 0)
        {
            SceneManager.LoadScene("ResultScene");
        }
    }
    else if (PlayerA == 1)
    {
        Debug.Log("Aのキャラはセイバー");
        Saber.SetActive(true);
        int hp1 = CharacterSelect.Ahp;
        if (hp1 == 0)
        {
            SceneManager.LoadScene("ResultScene");
        }
    }
```

③ キャラクターについて
本ゲームのキャラクターは2dの画像を繋げることにより動かしています。つまり、アニメと全く同じです。これから、それらの設定およびUnity上での動かし方について説明していきたいと思います。
まず、お絵描きアプリでキャラクターを描きます。そしてUnity上でSpriteという画像ファイルとしてそれらの絵をインポートします。
次に、アニメーションの設定です。Unityにはアニメーションを制御するアニメーターコントローラーと呼ばれるものがついています。これにはアニメーションの再生速度やアニメーション同士の遷移を設定できる機能がついています。（以下画像）



アニメーション同士の遷移は画像の矢印の方向に行われ、遷移にはそれぞれ条件（パラメーター）があります。

1つ目はfloat型で、特定の値よりも大きかったり小さかったりしたら、別のアニメーションに変移します。2つ目はint型で、これはfloat型とほとんど同じですが、指定できる値の範囲が違います。3つ目はTrigger型で、キーが押されといった特定の入力によって別のアニメーションに変移します。4つ目はbool型で、これは一回しかできないTriggerのようなものです。本ゲームではTrigger型とfloat型を使用しました。

また、キャラクターはパソコン上のキーボードで動かすこともできます。製作途中でキャラクターの挙動を確認するときに使用しました。

```
float x = Input.GetAxisRaw("Horizontal");
    float x2 = Input.GetAxisRaw("Vertical");

    if (A0==0) {
        if (x > 0)
        {
            transform.localScale = new Vector2(0.6f, 0.5f);
            if (Input.GetKeyDown(KeyCode.RightShift))
            {
                transform.localPosition += new Vector3(3, 0, 0);
                attack0 = true;
                audioSource.Play();

            }

        }
        if (x < 0)
        {
            transform.localScale = new Vector2(-0.6f, 0.5f);
            if (Input.GetKeyDown(KeyCode.RightShift))
            {
                transform.localPosition += new Vector3(-3, 0, 0);
                attack0 = true;
                audioSource.Play();
            }
            }
```

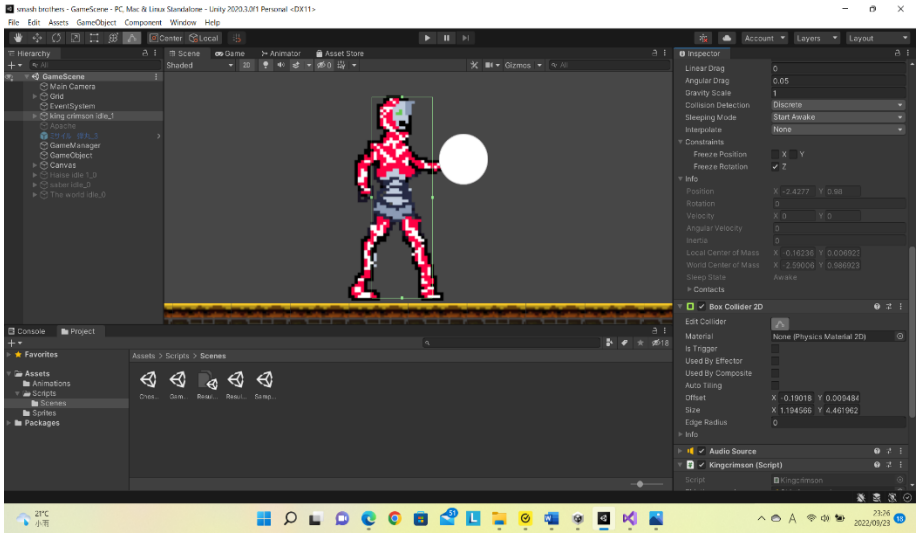
Input.GetaxisRawでキーボードから入力することをUnity側に伝え、その後（“Horizontal”）

において、具体的にどのキーボードからの入力なのかを指定します。Horizontalはキーボードの左右矢印キーを指定しています。その後Transfom.localScaleでキャラクターをどの方向へ動かすかXYZの3座標で指定します。

④　当たり判定について
どんなゲームのキャラクターにも当たり判定と呼ばれるものがついています。そもそも、当たり判定とは、ゲーム上の物体同士の接触や衝突を判定するために存在します。これがないと、キャラクターが壁や地面をすり抜けたりする現象が発生します。
バトルロワイアルゲーム、ApexLegendsのキャラクターヒットボックス（当たり判定）（Game Withより）

（~~マジでレヴナントとレイスは強化しなきゃヤバいよ、頼むからヒットボックスくらいは改善してくれ~~）

本ゲームのキャラクターヒットボックス（キャラクターを囲んでいる細い線です）



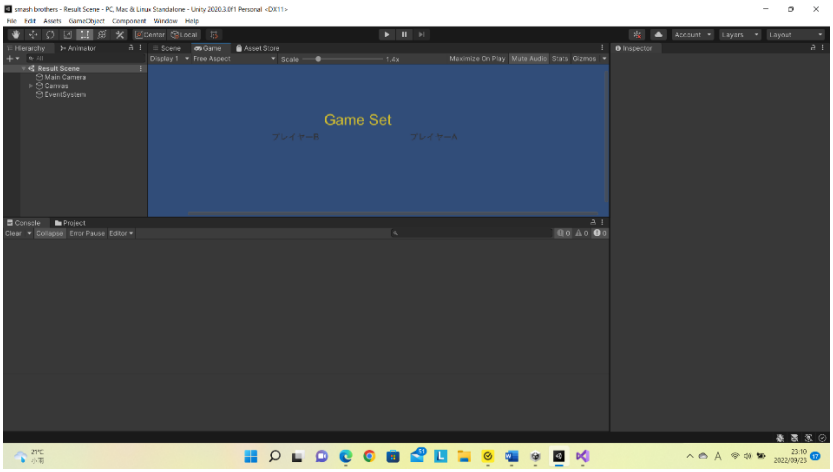
勿論、キャラクターのデザインによって当たり判定の大きさは若干違います。そのため、特定のキャラクターに対しては機能していても他のキャラクターには機能していないといったことが発生するので、手作業で確認しました。

⑤ ゲーム終了判定について

続いて、ゲーム終了判定について説明していきたいと思います。

まず勝利条件には、制限時間内により体力の多いプレイヤーが勝利する条件と、敵を倒したプレイヤーが勝利する条件があります。

また、ゲームが終了すると、以下のようなリザルト（結果）画面に遷移し、ゲームが終了します。



制限時間になるとシーンを遷移させるスクリプト

```
void Update()
 {
    if (isTimeup == false) {
       countTime -= Time.deltaTime;   //ゲームが開始してからの秒数をフレームで記録
       GetComponent<Text>().text = countTime.ToString("F2");

    }
    if (countTime < 0)
    {
       isTimeup = true;
       SceneManager.LoadScene("ResultScene");
     }
```

Unity上で、ゲームが開始してからの時間をフレームで計測し、その時にbool型の変数をfalseで設定します。0秒以下になったらtrueとなり、シーンが遷移します。

どれかのキャラクターの体力が0になるとシーンを遷移させるスクリプト

```
if (PlayerA == 0)
    {

       Debug.Log("Aのキャラはキングクリムゾン");
       Kingcrimson.SetActive(true);
       int hp0 = CharacterSelect.Ahp;
       if (hp0 == 0)
       {
          SceneManager.LoadScene("ResultScene");
       }
```

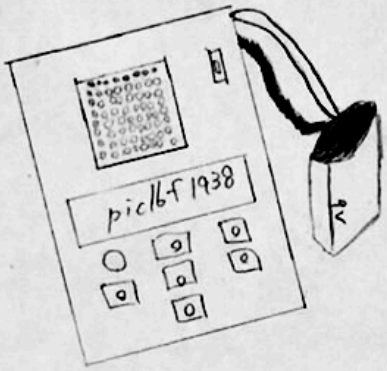
```
    }
    else if (PlayerA == 1)
    {
        Debug.Log("Aのキャラはセイバー");
        Saber.SetActive(true);
        int hp1 = CharacterSelect.Ahp;
        if (hp1 == 0)
        {
            SceneManager.LoadScene("ResultScene");
        }
    }
```

使っているメソッドは先ほどとほぼ同じなので、説明を省略します

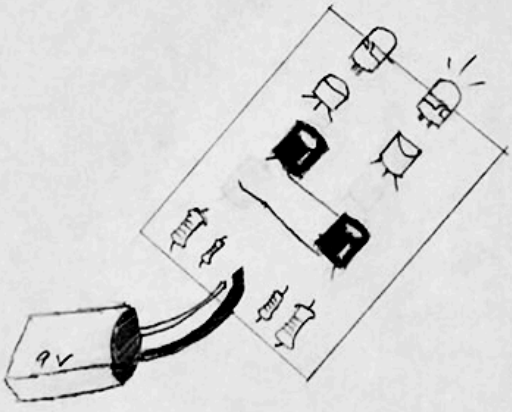
製作の感想

山田　初めてのUnity制作で、わからないことや知らないことがたくさんあった。非効率的なスクリプトを書いていた箇所もあったので反省したい。また、2Dという特性上、どうしてもキャラクターやステージのクオリティを上げるのはなかなか難しく、苦労した。
たくさんの人に助けてもらったので感謝したい。

大和田　はじめての制作、初めてのパソコンで、分からないことはたくさんあったが、山田さんやその他先輩に教えてもらうことで何とか制作を形にすることができた。
また、最終調整やキャラ選択などの自分でわからないものを山田さんに押し付けるような形になってしまっていたので反省したい。
また、制作自体はとても楽しく、思い出となった。



チカチカ & 売り物

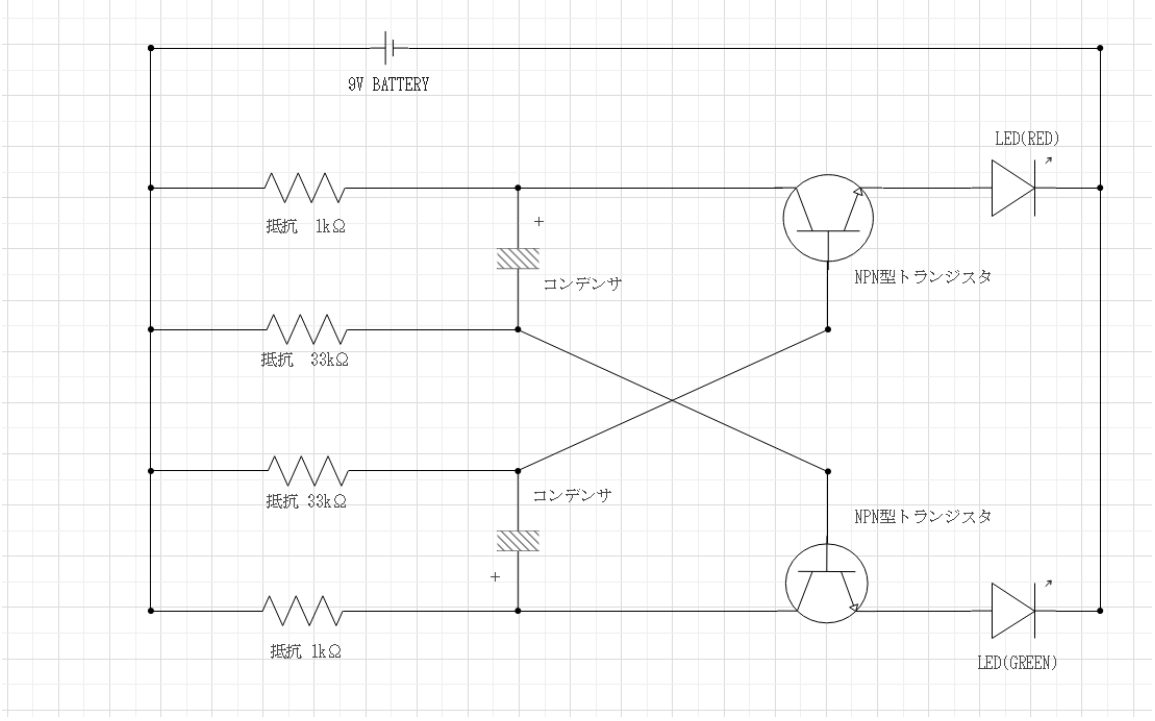


チカチカ

M3　鹿島礼次

チカチカとは、文化祭のチカチカ体験で実際に作れる回路です。正式名称は「非安定マルチバイブレータ」です。

こちらが回路図です。



赤と緑の LED が交互に光ります。仕組みを見ていきましょう。

①＋極から流れる電流は、より抵抗の少ない 1kΩ 抵抗のあるほうに流れます。

②電流は抵抗を通過したあとコンデンサ（電気を貯め、たまりきると一気に放出する）に流れ、貯まります。

③貯まりきった電気が一気にトランジスタ（電気を増幅する、電気で押すスイッチのような役割）に流れます（このとき、2 つのトランジスタのわずかな非対称性から片方のトランジスタのみ ON になります）。

④LED に電気が流れ光が付きます。

⑤コンデンサが空になり、①に戻ります。

売り物

制作　M3 坂東　協力　物無の皆様

ストーリー

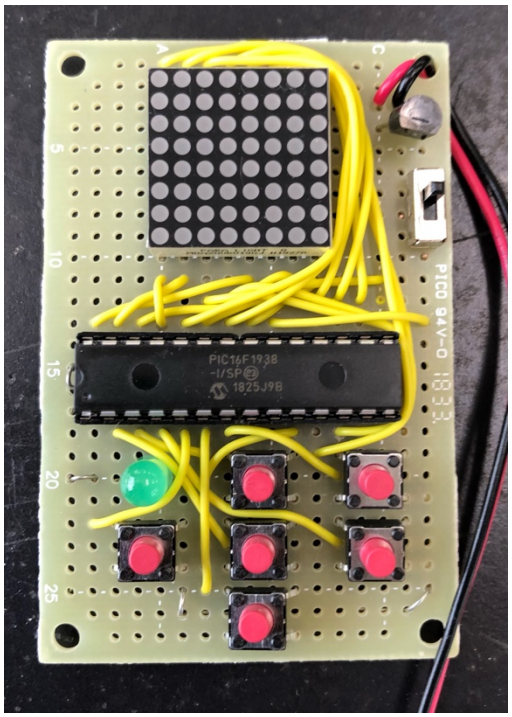
B「来月からポケモンのランクマで幻が使えるんだけどマーシャドー持ってないから貸して」

M「いいけどタダであげるわけにはいかないなぁ」

B「何をしたらいい？」

M「とりあえず売り物 100 個作ってもらおう」

外観



部品説明

・三端子レギュレータ･･･右上にある黒い半円のやつです。電圧を下げる働きがあります。

・スライドスイッチ･･･三端子レギュレータの下にあるやつです。電源の ON/OFF を切り替えます。

・ドットマトリックス･･･上にある正方形のやつです。8×8 の LED で画面を表示させます。

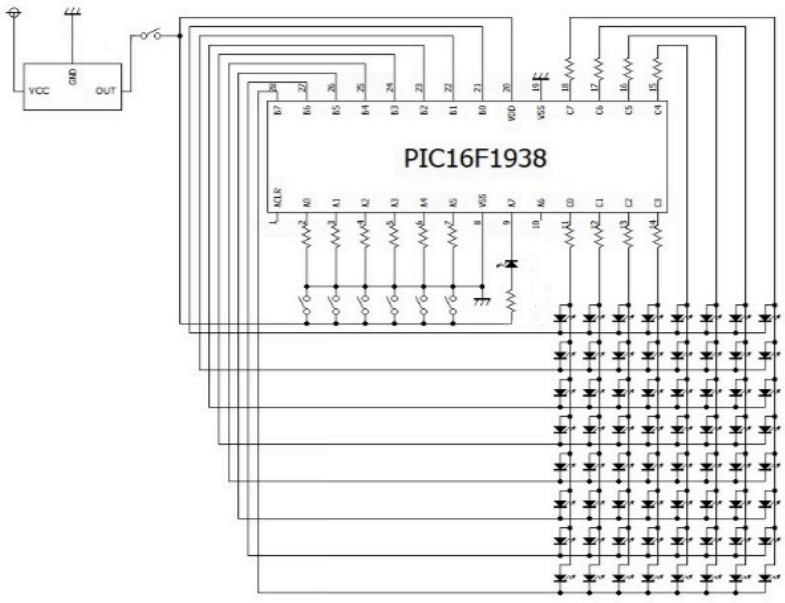
・PIC16F1938･･･真ん中にある長方形のやつです。このゲームのプログラミングがされています。

・タクトスイッチ･･･下にある合計 6 個のです。真ん中が決定、右上がリセット、その他が十字キーです
・LED･･･タクトスイッチの左上にあるやつです。光ります。

概要

4 つのゲームが遊べます。それぞれ対応したスイッチを押すと始まります
1 つ目のゲーム･･･上のスイッチを押すと始まります。3 と 5 と 6 と 9 が表示された時、スイッチを押すゲームです。最初にプレイ人数を選べます。1 人用を選んだ時は決定を 2～4 人用の時は十字キーをそれぞれの人が押してプレイできます。
1 人用はミスするまで続きます。2～4 人用は誰かが 5 回成功したらその人の勝ちです。
2 つ目のゲーム･･･右のスイッチを押すと始まります。上から玉が降ってくるので避けるゲームです。
LED が光ってる時真ん中のスイッチを押すと自機がいる列の玉を消せます。
3 つ目のゲーム･･･左のスイッチを押すと始まります。順番に数字が 10 個出てくるのでそれを覚えて出てきた順に入力していくゲームです。
4 つ目のゲーム･･･下のスイッチを押すと始まります。迷路です。矢印の書いてある方向に進めます。

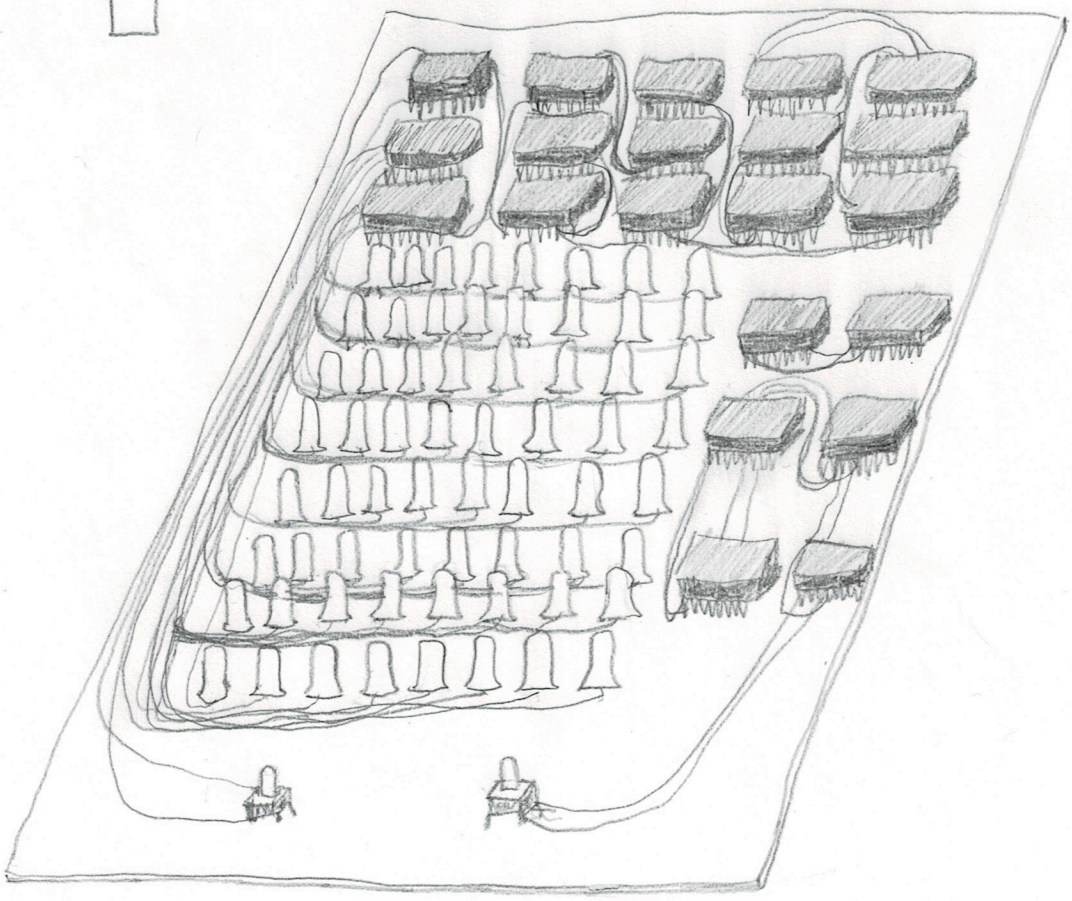
回路図



感想

プログラミングの知識を身に付けられ、はんだ付けも上達したと思うのでこれから先に活かせれば良い活かしたいなと思います。

中1ゲー



賭博転生録カイジ

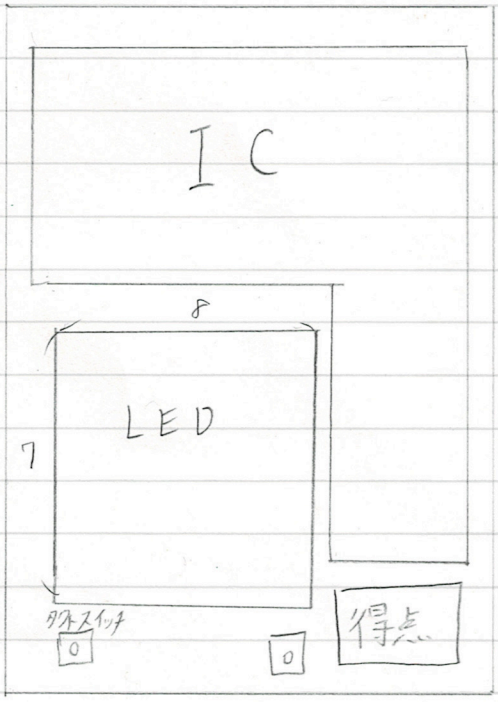
制作：M2 伊藤至
回路図設計：H1 山田さん

・ストーリー

ある日、なぜかパチカスに転生してしまった!?手をつけてはいけないお金をパチンコで増やして借金を返そう！お金をへらすことは絶対に許されない。
???「勝つことが全てだ!!勝たなきゃゴミだ！」

・根元要

8×7のLEDの四行目にある自機をスイッチで左右に動かす。上から降ってくる玉にあたると−1，下から上がってくる玉にあたると+1となる。一定時間が経過するか、有り金を全て擦ってしまったらゲーム終了。スコア制のゲーム。

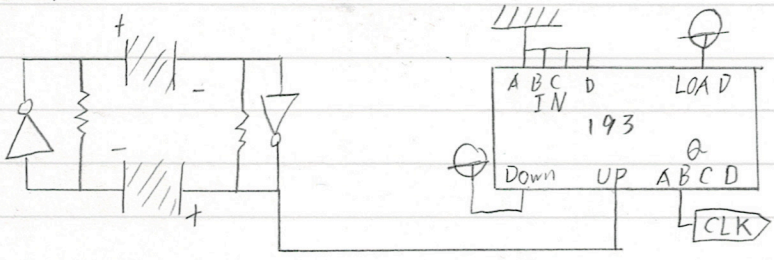


・使用部品

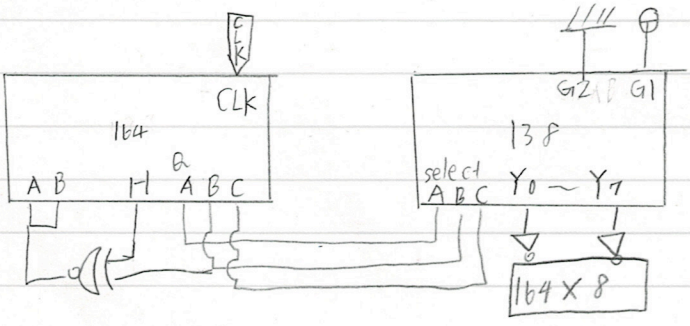
74LS IC，ICソケット
単色LED(赤、緑)，三色LED
抵抗，コンデンサ，7セグ
タクトスイッチ，配線材

・回路図

発振

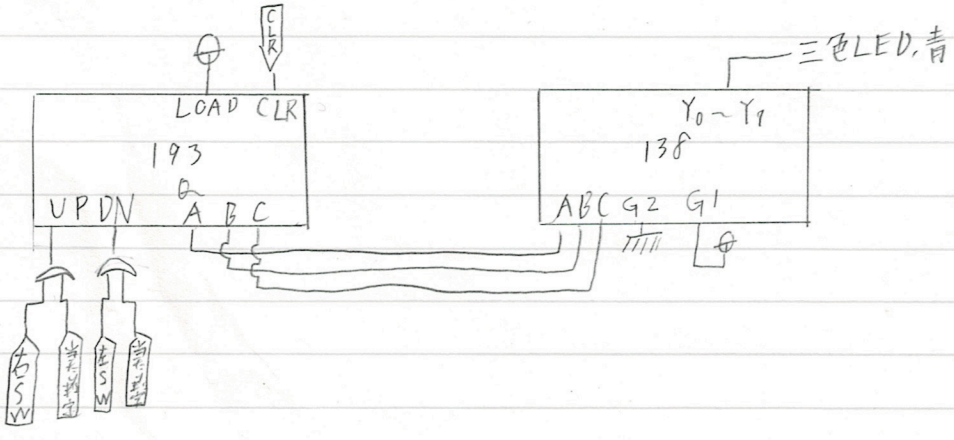


玉流れ



X 2

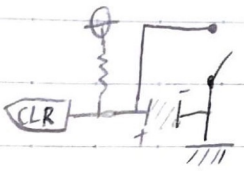
自機移動力



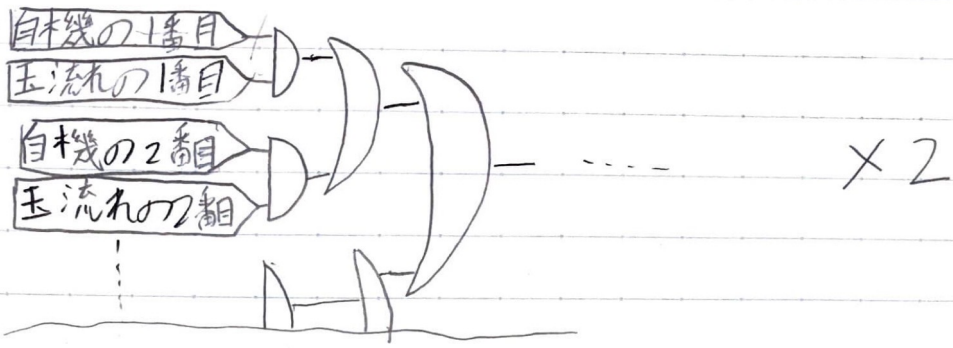
No.

Date

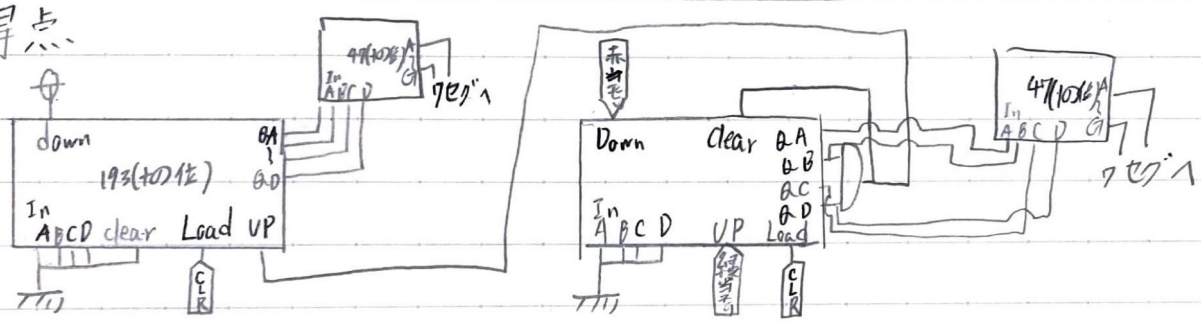
クリア



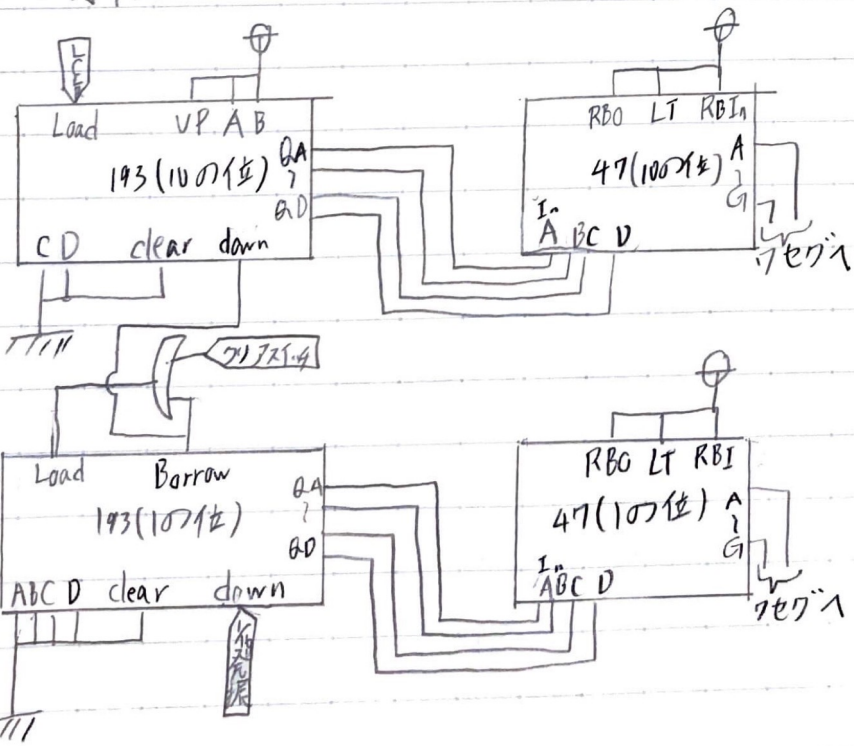
当たり判定



得点



時間制限



感想

1年間の制作を通して電子工作の基礎を学ぶことができました。来年の製作もこの経験を生かしてがんばります。

透明ボールで遊ぼう

製作
M2 西澤
回路図設計 ご協力
H11 釜田さん 物無の皆さま

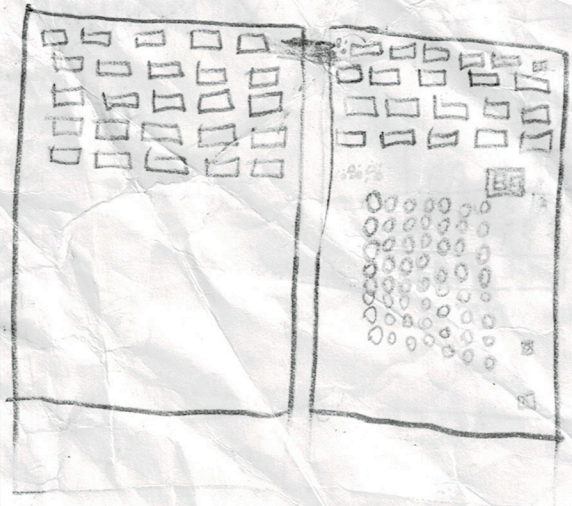
次週予告

「さーて、来週のブリムさんは？」

N澤です。いや、T真の方がいいか…T直です。引き続き友達とドッジボールをしている僕ですが、同学年の物無員1人がボールを全投入してきやがったんです！アヤツ！とまぁそれはさておき残り時間を見てみるといっ○!?どういうことだよ！しょうがない、このまま耐えるしかないか。
さて来週は、「運動神経皆無」「瞬間移動会得」「ザ・ワールド」の3本です！

根元要

①外観



②ルール

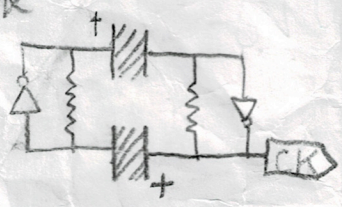
左右から弾がとんでくるので、中央の列でどちらもよけてください。よけた利少数女がそのまま得点になります。

③使用した部品

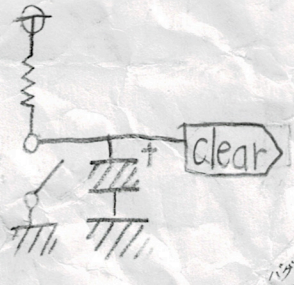
- ☑はんだ
- ☑配線材
- ☑スズメ物泉
- ☑74LS(HC)IC
- ☑3色LED
- ☑1kΩ抵抗
- ☑コンデンサー
- ☑クリアスイッチ
- ☑2色LED
- ☑7セグ

回路図

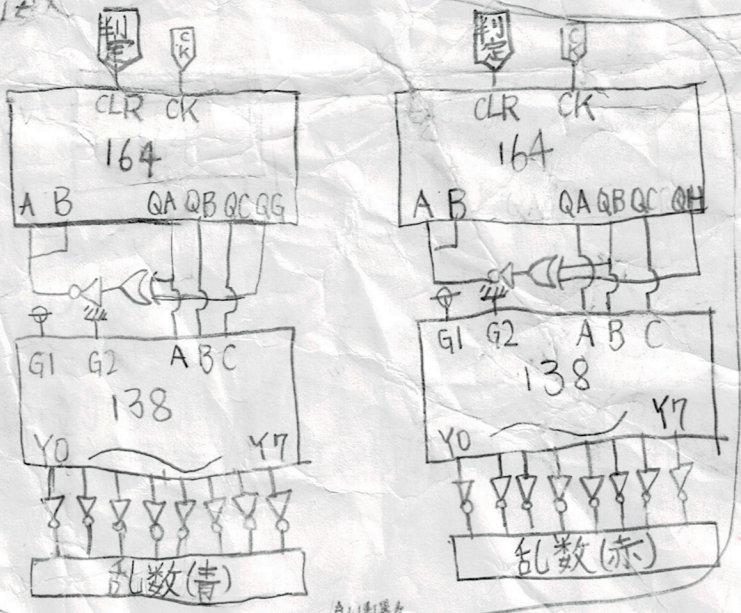
発振



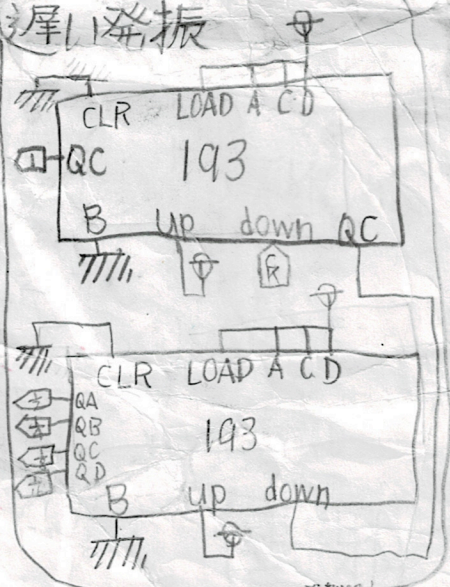
クリアスイッチ



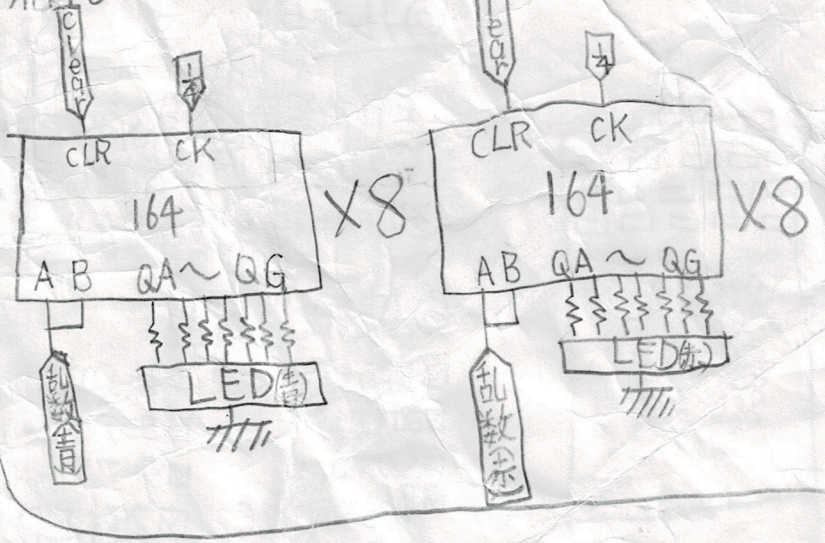
乱数



遅い発振



玉流れ

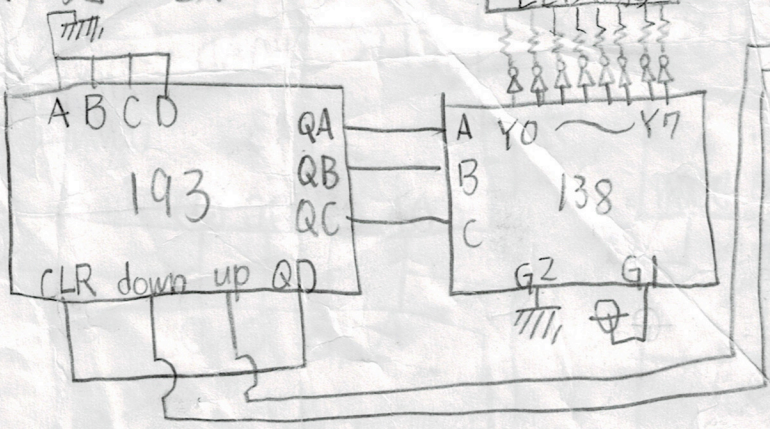


※乱数(青,赤)はそれぞれ
Y0～Y7から1つずつとる。

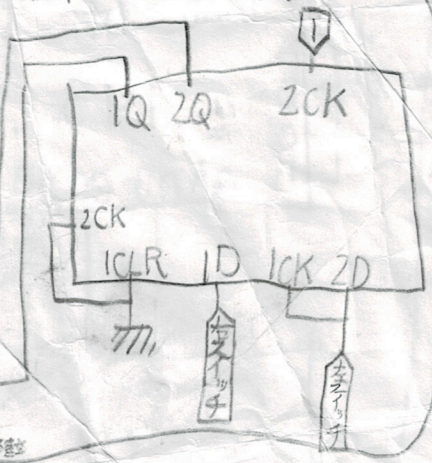
透明ボールで遊ぼう

(続き)

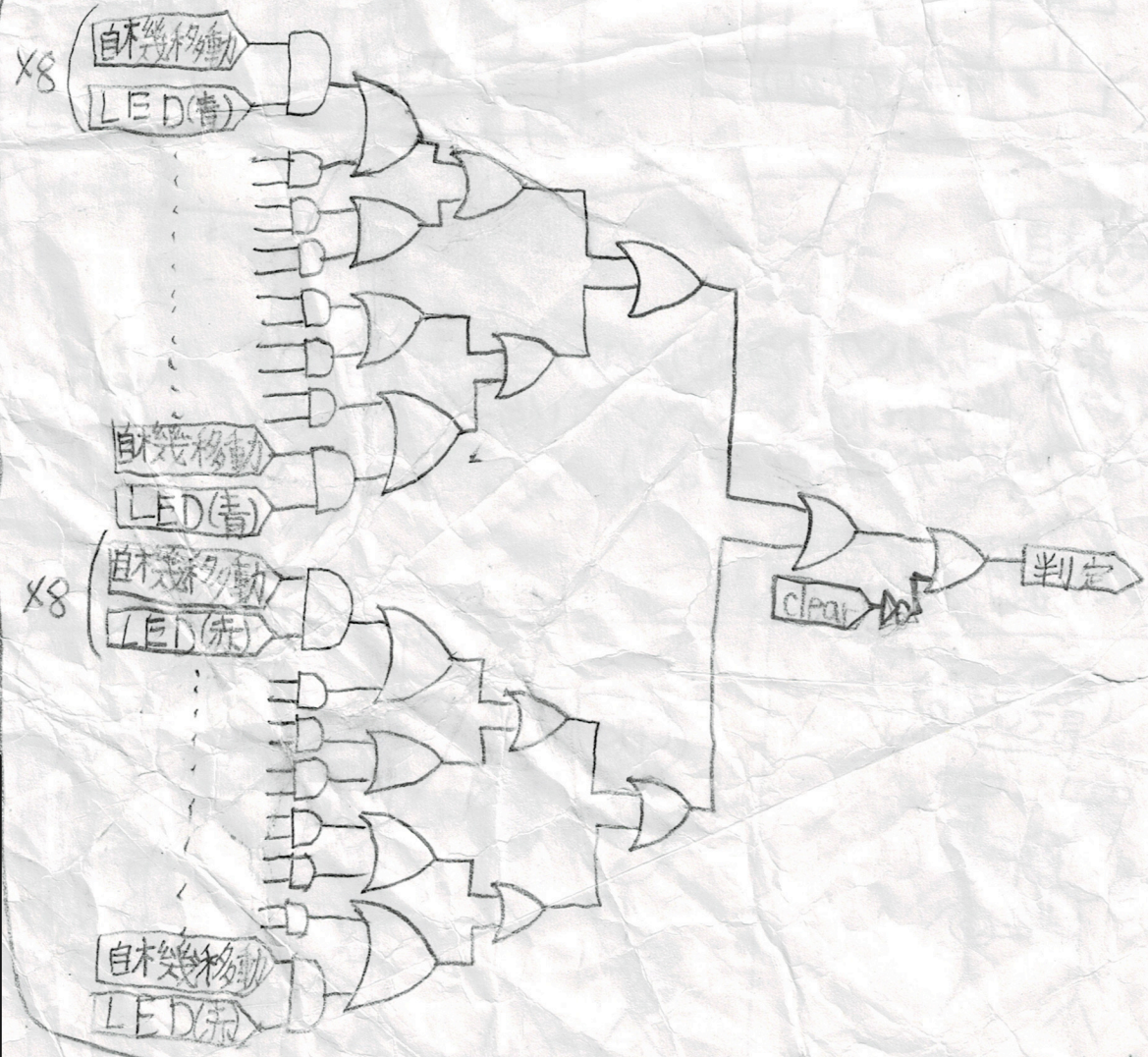
自機移動



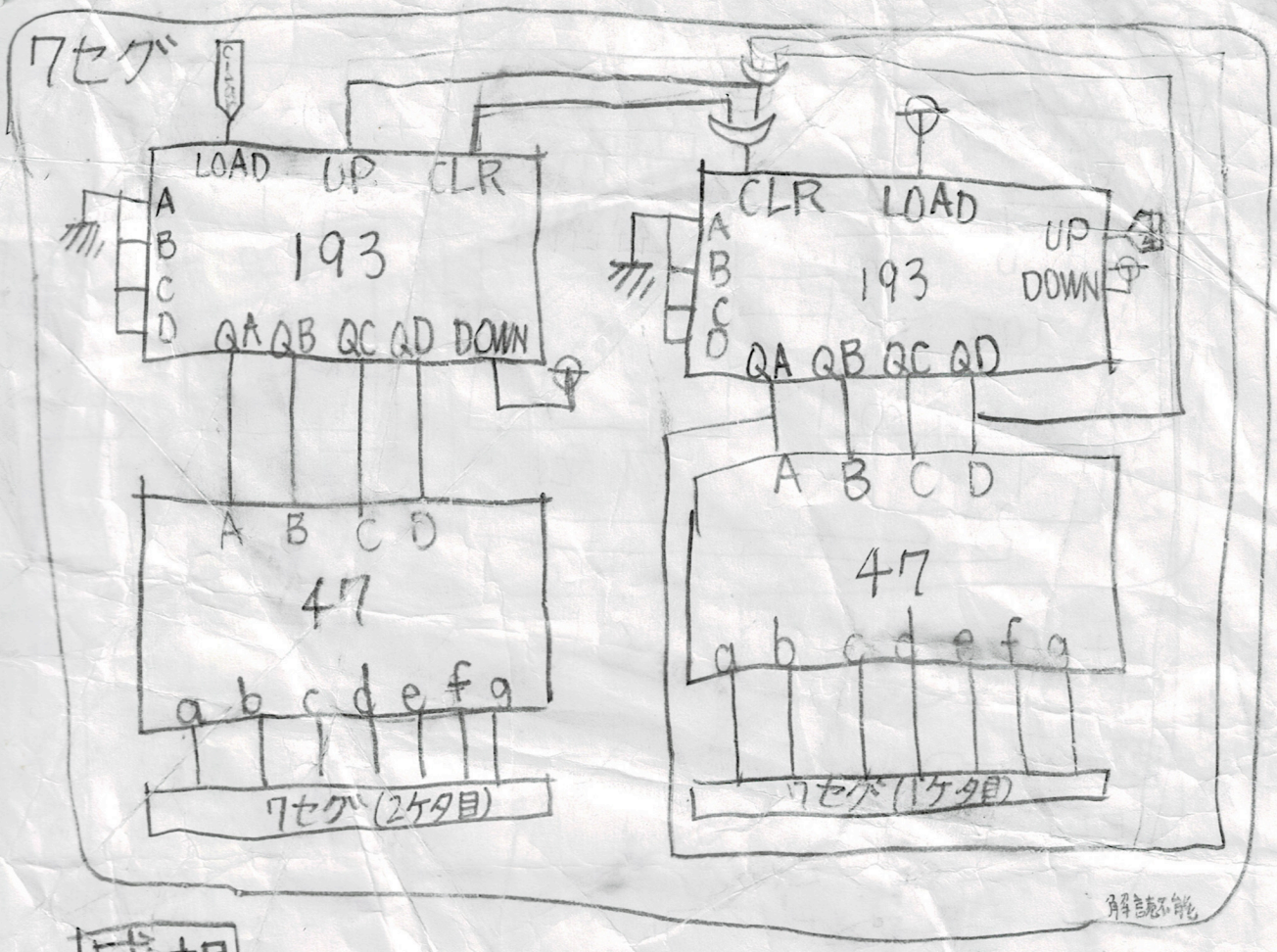
チャタリング防止



当たり判定



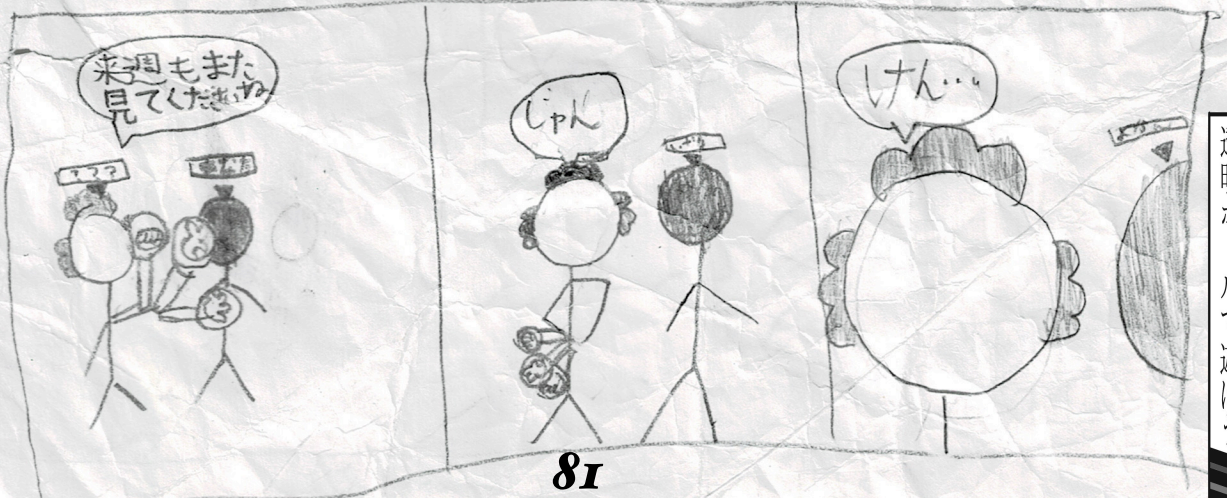
透明ボールで遊ぼう



感想

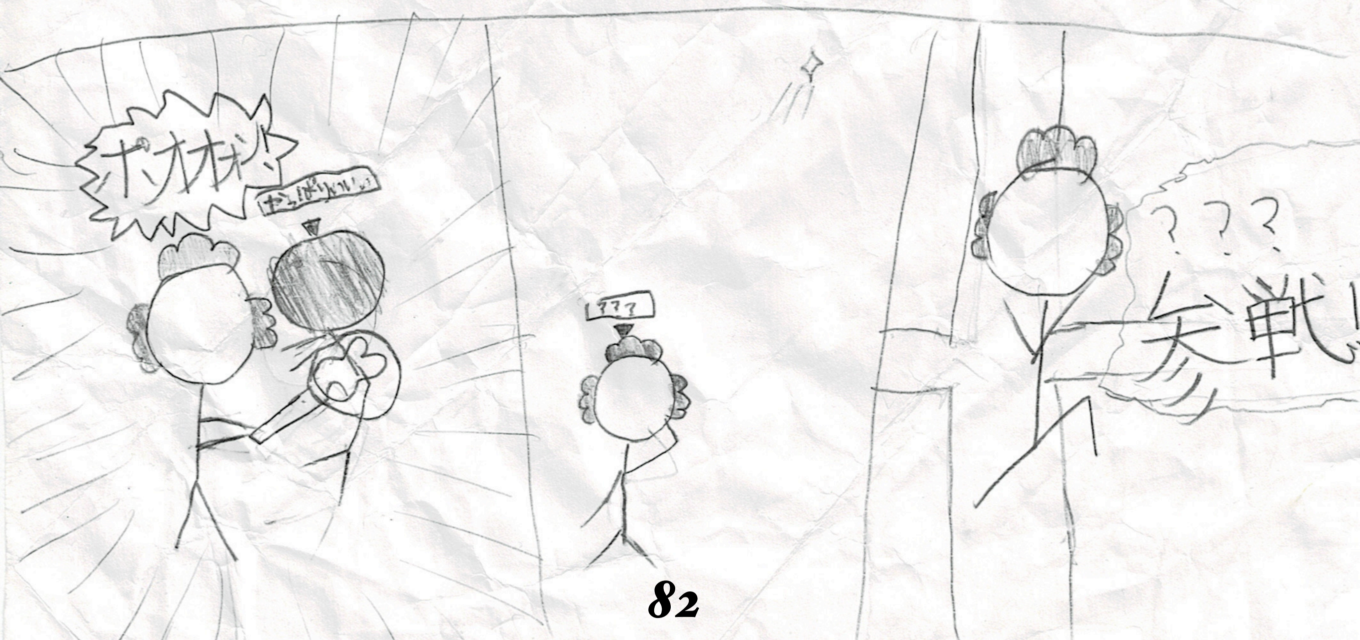
中1ゲーはバグの宝庫だという言葉は本当でした。作ってはバグり、修正してはバグり、メンドい修正をしてもバグりました。とにかくここから完動させます！ ボンバルバ

※おまけ



←※スペシャルサンクス：妹

ページのバランス
考えろ
for 妹



友達を救え！

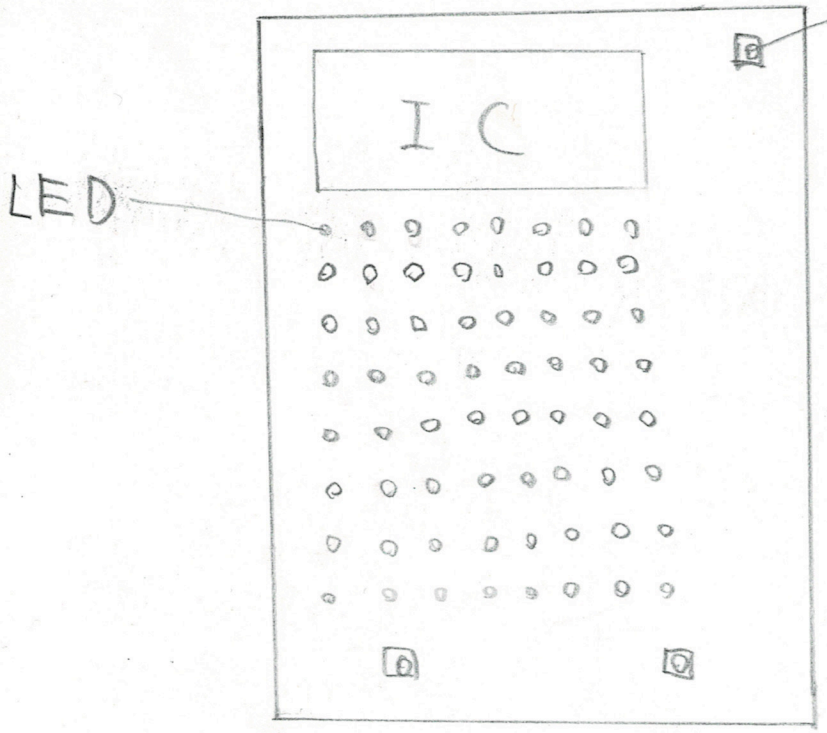
製作者
M2 水谷
回路図設計者
H1 永谷さん

・ストーリー

APEX中毒になり洗脳されてしまった
友達が突然おそってきた！
銃撃戦に勝って目を覚まさせてあげよう！

・概要

敵が撃ってくるたまをよけながら、自分も
たまを発射し、敵に当てよう。

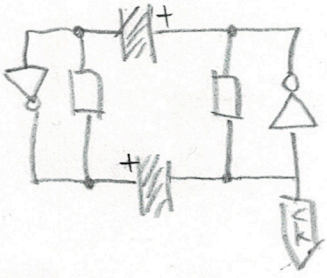


・使用した部品

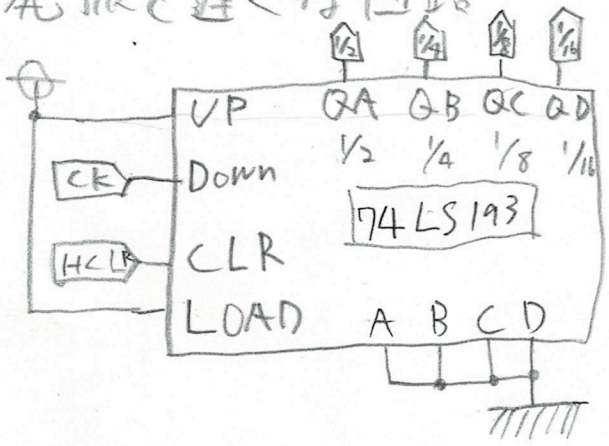
- LED
- タクトスイッチ
- 74LS IC
- 抵抗
- コンデンサー
- 配線材

回路図

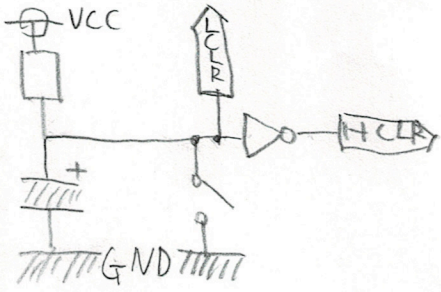
発振



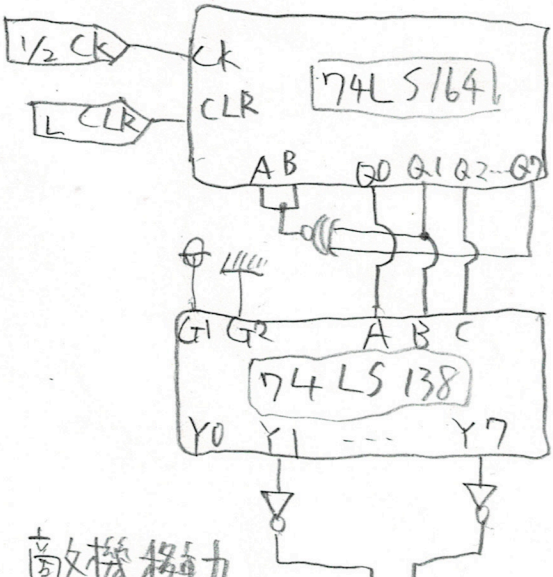
発振を遅くする回路



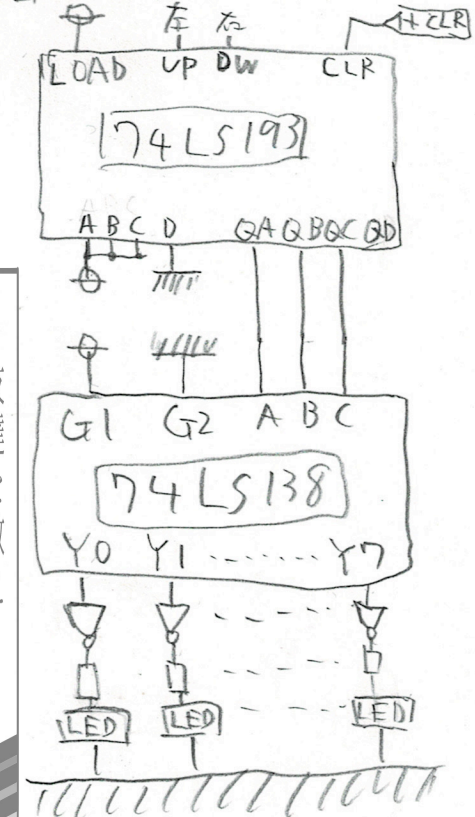
クリアスイッチ



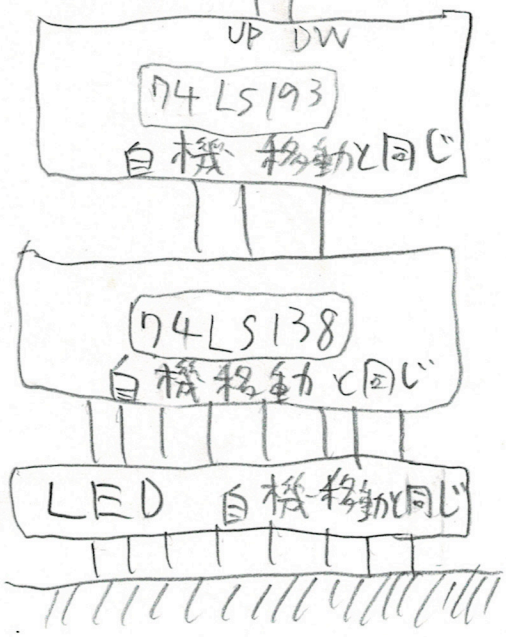
乱数回路



自機移動

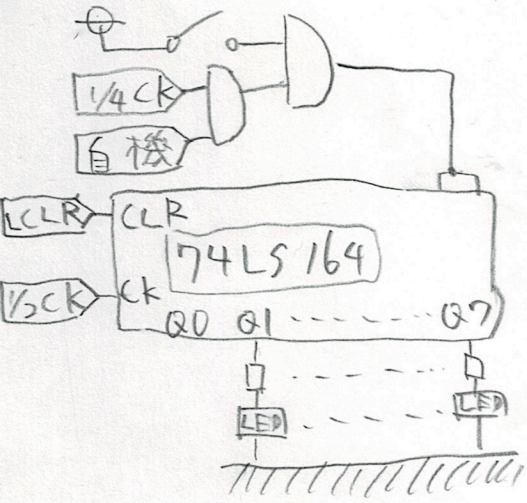


敵機移動

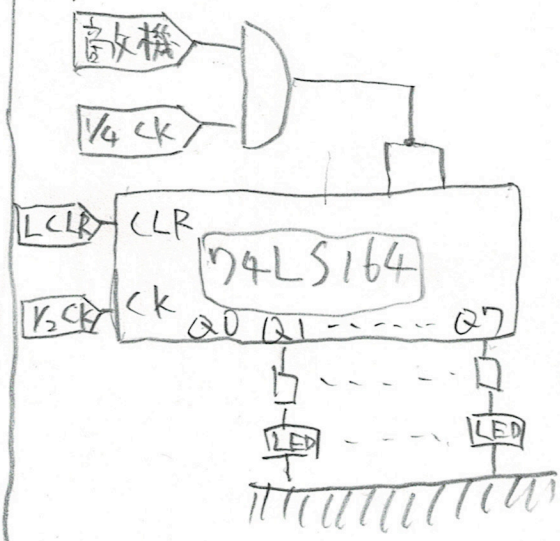


友達を救え！

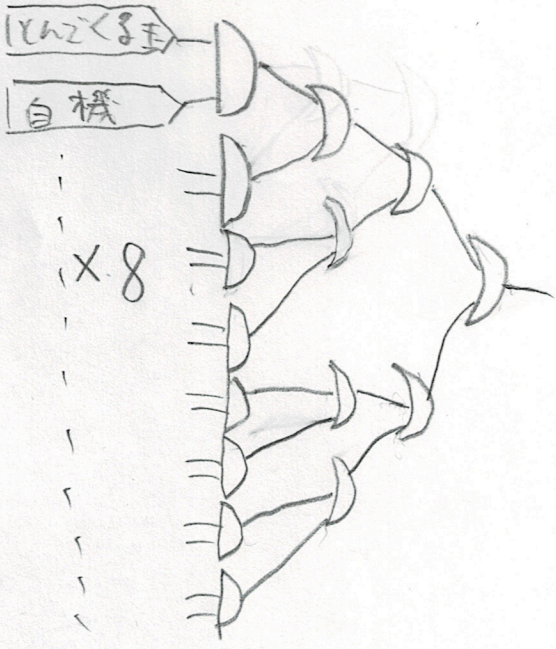
玉流れ（自機）



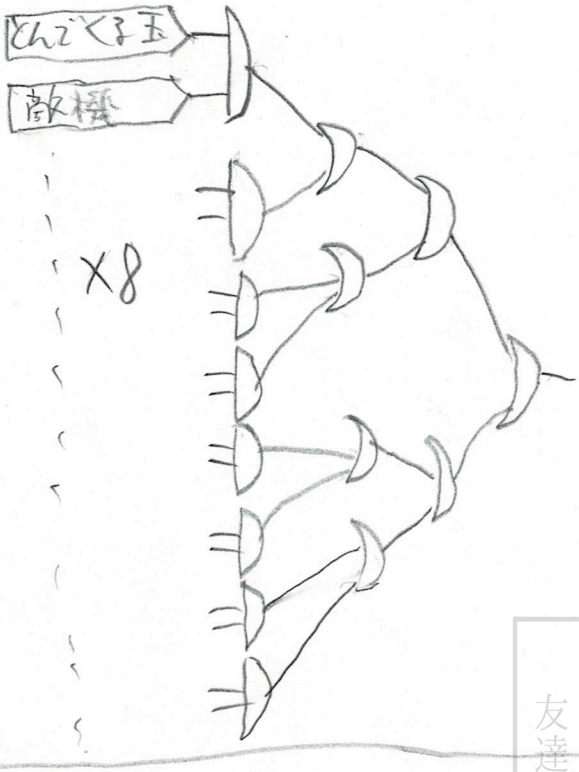
玉流れ（敵機）



当たり判定（自機側）



当たり判定（敵機側）



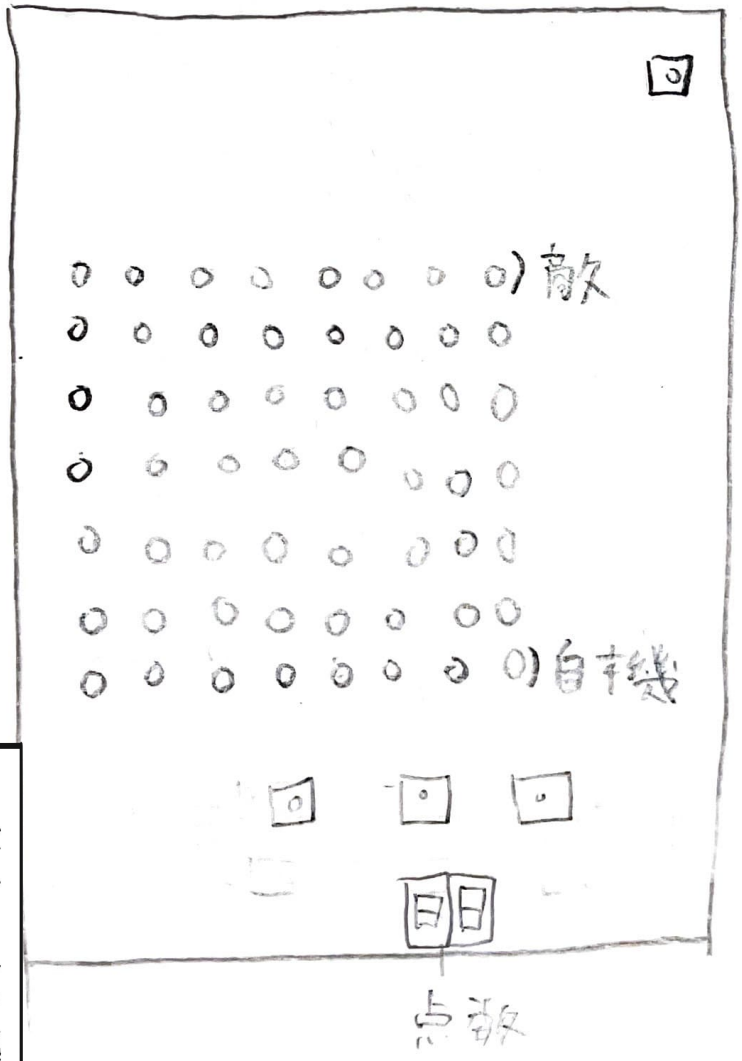
<感想>
はんだ付けや、配線などの作業が一年前と比べてすごく上手になった。先ぱいに優しく教えてもらいながらできたので楽しかった。

宇宙人の襲来

制作
M2 涌田
回路回設作
H1 楊さん
ご協力
物無の皆さま

ストーリー
UFOが大量に攻めてきた!
レーザーを打ってうち落とそう!

外観

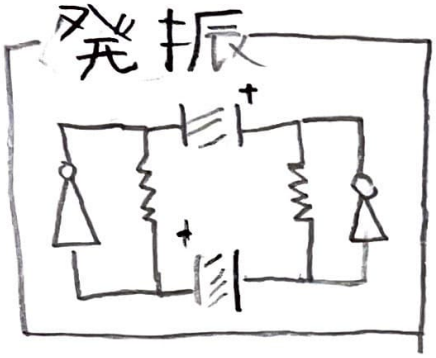


ルール
上に出てくる敵をひたすらうつゲームです。

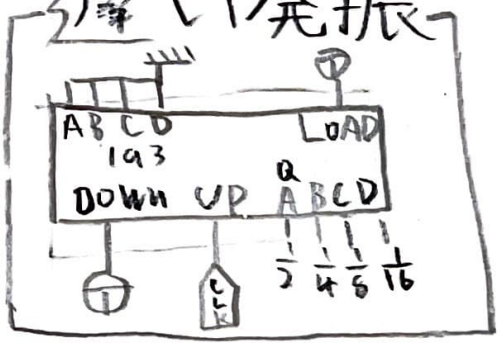
感想
この一年でしんだ付けかかなり上手くなったと思います。接　不良などもあったけど、楽しかったです。

回路図

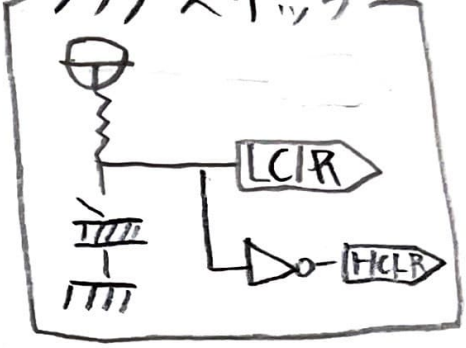
発振



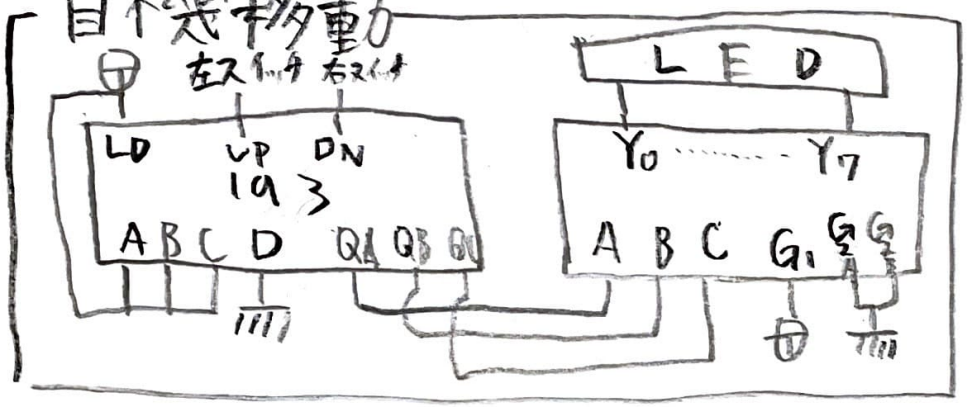
遅い発振



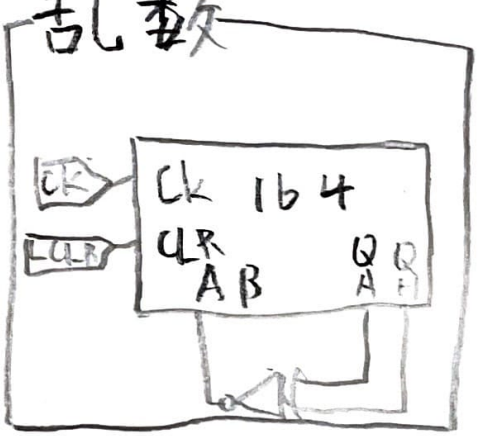
クリアスイッチ



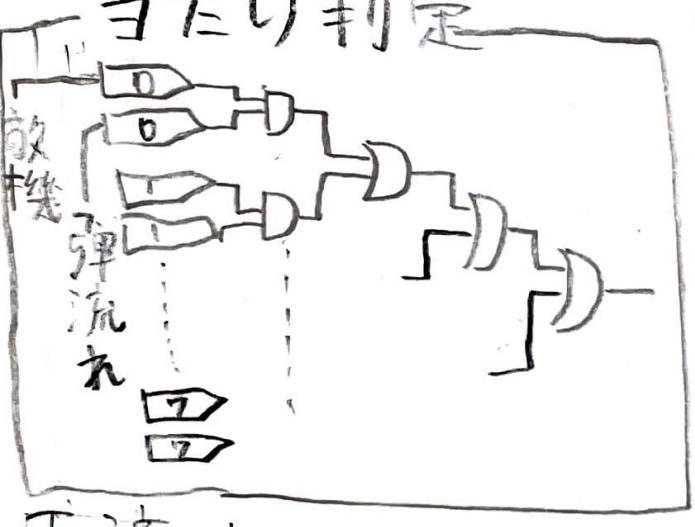
自機移動



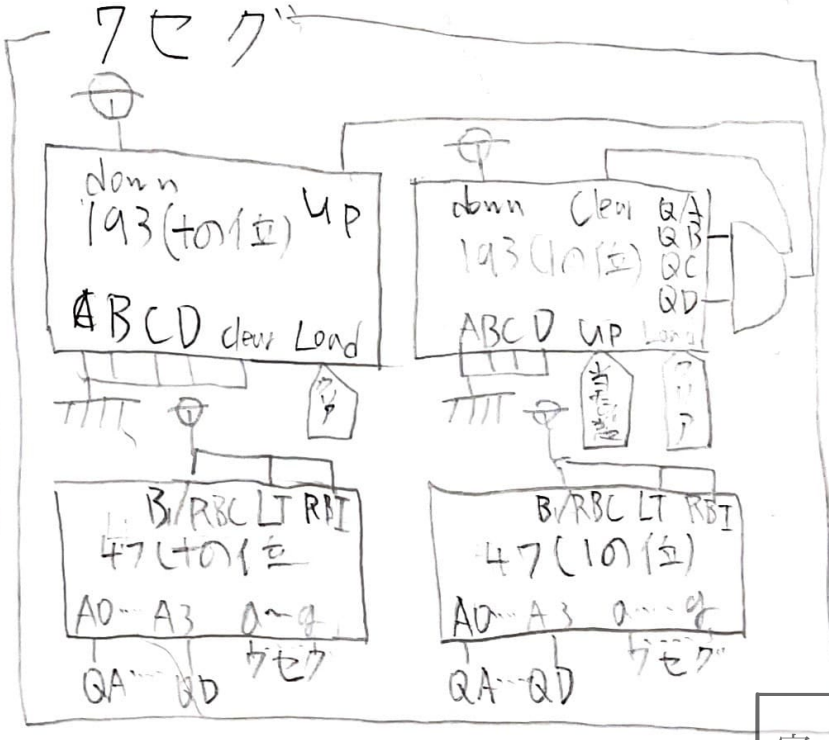
乱数



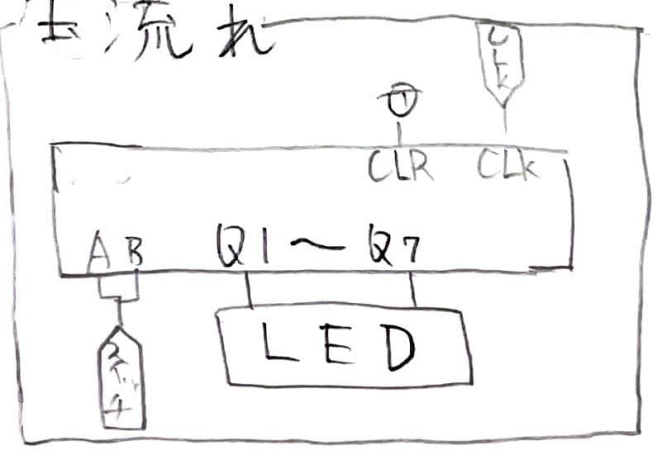
当たり判定



7セグ



玉流れ



イライラ棒

制作

中一一同

M3　石村

協力

物無の皆様

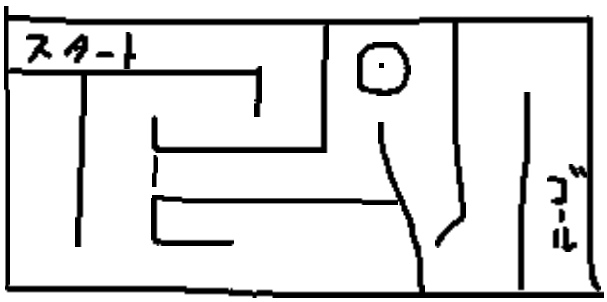
ストーリー

文化祭直前のある日、とある部員が物理大実験室にやって来ると、そこには某競合展示団体により爆弾が設置してあった。

これでは数ヶ月かけた制作物が台無しになってしまう!!!

そこで、彼は一触即発の中、慎重にニッパーの刃を進め始めた。

概要



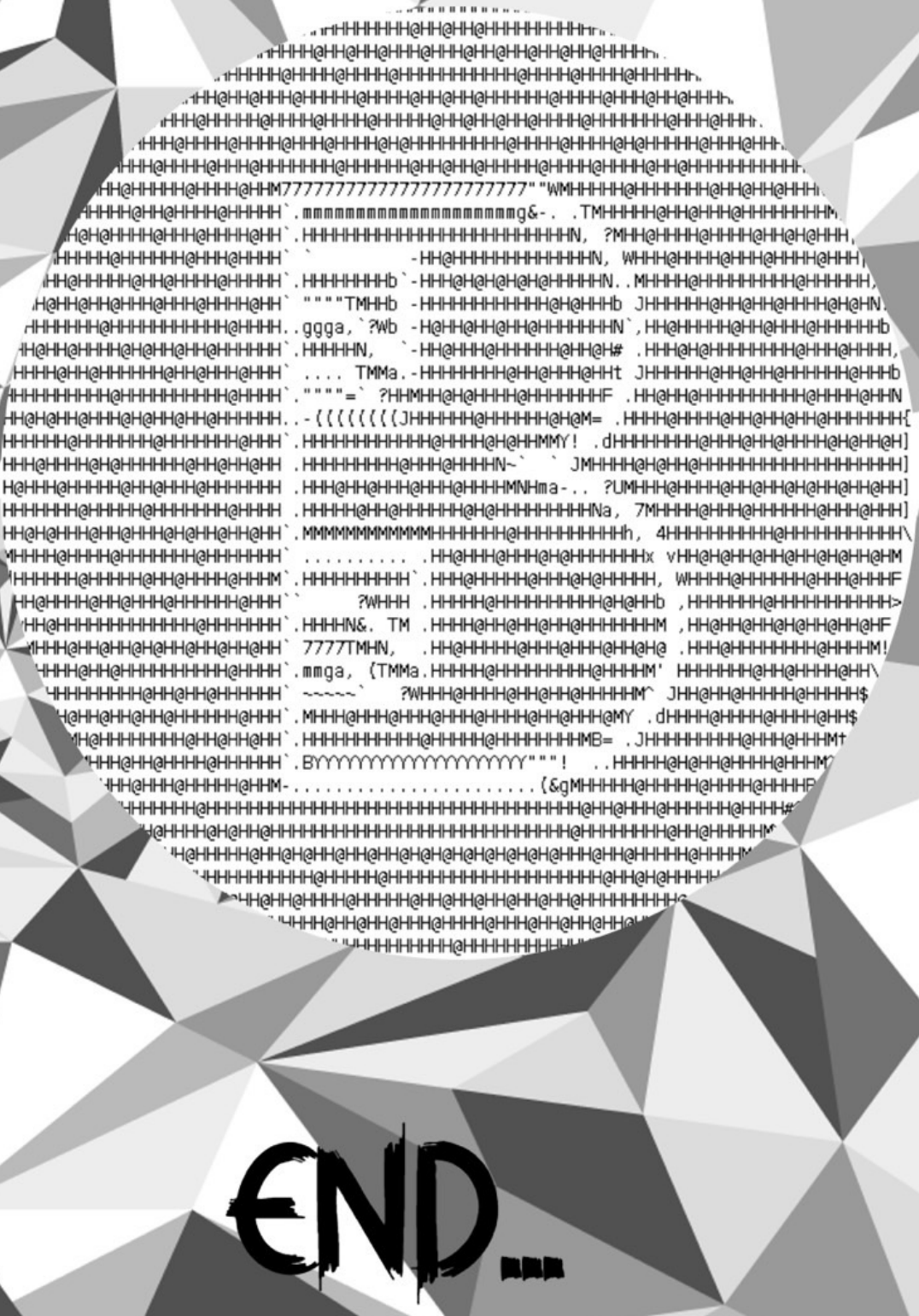
麻布学園物理部無線班
回路図集 2022

編集責任者　　加藤稜大　礒野大洋

デザイン　　大谷心雲　伊藤拓真
大和田春樹　坂東直哉
伊藤至　　水谷哲

物理部無線班公式 HP
URL　http://butumu.com/

2022



END...