麻布学園物理部無線班

2021年度回路図集

はじめに

昨年の回路図集では
「米インテルが「TigerLake」のコードネームで
開発してきたモバイルCPUを「第11世代
Coreプロセッサー」として正式に発表した」
とありましたが、最近では、
Microsoftが6月24日に「Windows10」の
後継となるOSである「Windows11」を発表、
またインテルの「第12世代Coreプロセッサー」
など進歩が凄まじいです。
我が部では、昨年の経験を生かしたロボットや、
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は保証
はございません。あらかじめ御了承下さい。

H2会計 海野博亮

目次

はじめに 2

アナログ系 5

ぶつみん ..... H1 加藤 p6

ロボット系 8

21 式制御器同期型機械義手「式神」 ..... H2 吉田・M3 伊藤・秋田 p9

ARCS ..... H1 畠山・M3 大谷 p13

walker ..... H2 中村・M3 広瀬 p16

あれ、僕またなんかやばい戦車つくっちゃいましたか？ .... H 1 谷中・M 3 榎本亘・鳥内 p18

AZAX LEGENDS ..... H 2 徳田・H 1 山田 p21

RCMB による海洋征服計画 ..... H 1 中澤 p24

GAGO ..... H 2 高角・M 3 加藤 p27

デジタル系 28

ポケモン図鑑 ..... H 2 海野・M 3 山田 p29

iWatch ..... H 1 礒野・M 3 和田 p33

ゲーム系 37

バンバンシューティング VR ..... H 2 池田・M 3 永谷 p38

1000 年後のテレスクリーン ..... H 1 高梨・M 3 榎本大智 p40

売り物 42

勧誘から逃げろ！ ..... M 2 天野 p45

定期試験 ..... M 2 石村 p47

弾よけゲーム ..... M 2 大和田 p50

シューティングゲーム ..... M 2 片山 p53

ココナッツ・シューティング ..... M 2 加藤 p55

スーパーマリオ外伝 ..... M 2 角野 p57

過激派ヴィーガンとの死闘 ..... M 2 小平 p61

リンゴをキャッチ.....M２小林 p63
石を避けて生き延びろ！.....M２坂東 p66
運をつかめ！.....M２三牧 p69
親方！空から女の子が！.....M２山田 p71
かき氷を食べたい.....M２鹿島 p73
イライラ棒.....M２石村・H2 海野 p75

ANALOG

DO YOU KNOW ANALOG?

ぶつみん

製作者　H1　加藤

・ストーリー

睡眠　快眠　公民　冬眠　人民　県民　臣民　安眠　遊民　仮眠　ビタミン　ムーミン　ピクミン　ジアミン　チアミン　ジャスミン　アルミン　……

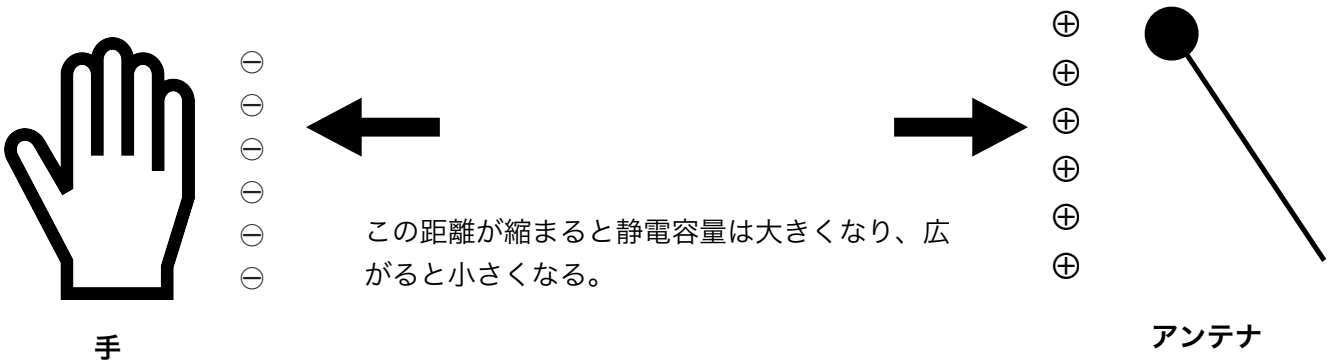
　そういえば進撃の巨人が完結しましたね。でも今回作ったのはアルミンではなくテルミンです。

・概要

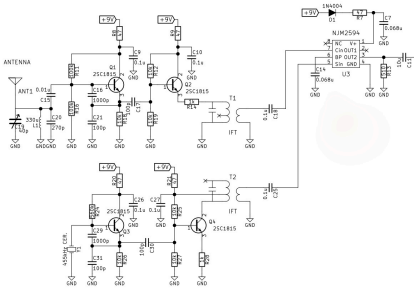
　テルミンとは、ロシアの発明家レフ・テルミンが1919年に発明した世界最古の電子楽器です。その特徴は、手で直接触れるのではなく空間中で手を動かすことで音程や音量を調整して演奏することです。

　テルミンには高周波の発振回路が2つ（1つは発振周波数が固定されていて、もう一つは発振周波数が変化する）あり、それらの周波数の差の変化を使って音程を変化させます。発振周波数が変化する高周波発振回路にはアンテナが付いており、手とアンテナの間の距離を変化させることで静電容量が変化するため発振周波数を変えることができます。ここで高周波の発振回路を使っているのは、手とアンテナによって変えられる周波数の変化はとても小さく、低周波の発振回路だと演奏するのに十分な音程の幅が得られないからです。

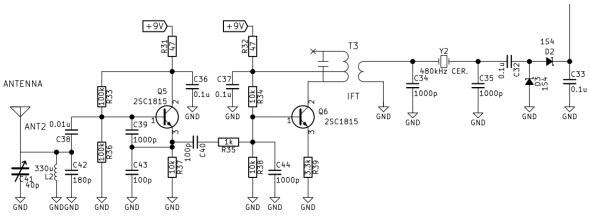
　また、音量も同じように手とアンテナによって周波数を変化させることで調節することができます。



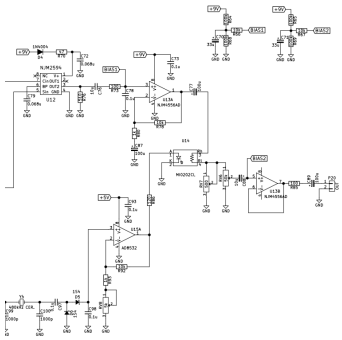
・回路図



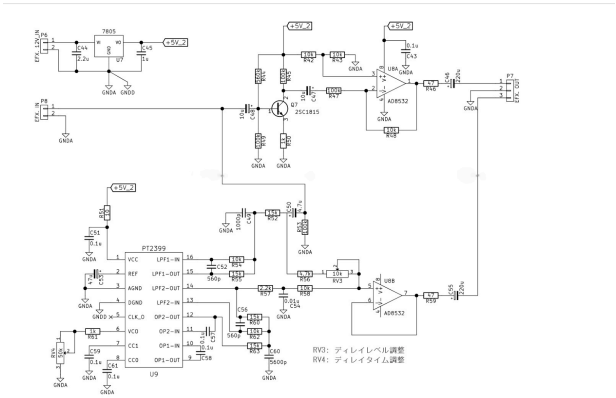
←音程を制御する回路です。概要に書いたように高周波発振回路が二つあり、一つはアンテナが付いているため発振周波数が可変で、もう一つは固定です。NJM2594というICを使って二つの発振周波数の差を取り出しています。



↑音量を制御する回路です。ここでは発振周波数の変化を直流電圧の変化に変換しています。



↑音程制御の回路で取り出した周波数を音声信号にしています。また、音量制御の回路からの電圧変化を抵抗値の変化に変えています。



←エフェクターの回路です。下半分がディレイの回路で、上半分がディストーションの回路です。

・感想

パソコンを使わない製作をしたいなと思いテルミンを作りました。でもアナログの知識が全然無かったので大変でした。来年はこれを活かした製作をしたいです。

ロボット

21式制御器同期機械義手「式神」

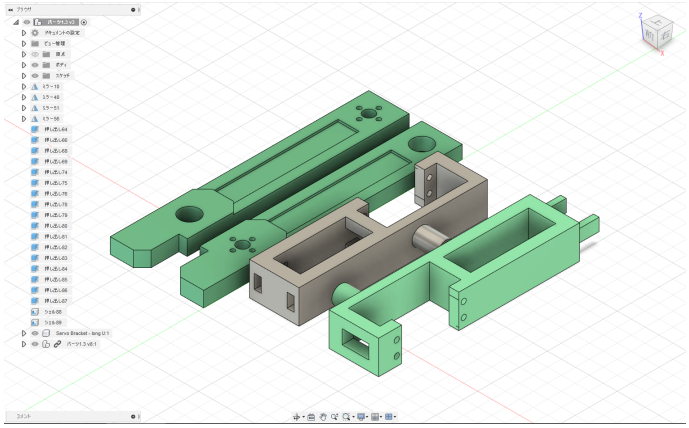
製作者　H2　吉田
M3　伊藤・秋田
協力　物無の皆様

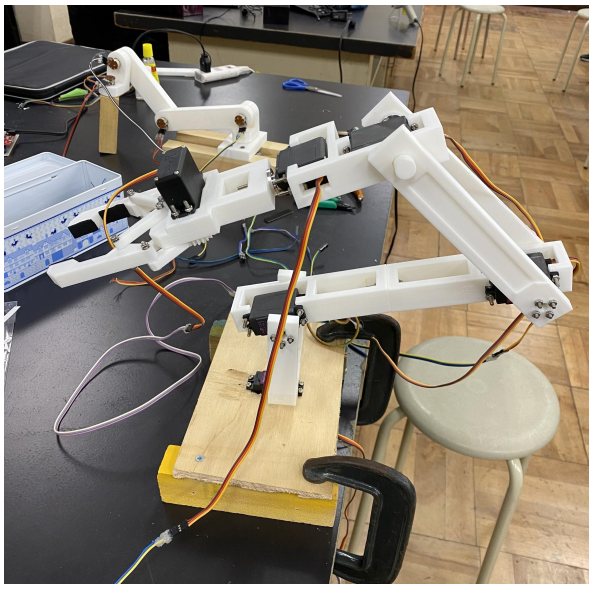
ストーリー

　ある夏、物無員のＩ藤は多汗症に悩まされていた。
「駄目だ！手汗のせいでペンも半田ごても滑ってマトモに使えない！」
「そうだ！自分の手の代わりに物を掴んでくれるロボットアームを作れば良いんだ！」
そしてＩ藤はY田さんとA田を仲間に入れ、多汗症治療の薬がある事を知らないままロボットアームを作り始める…

概要

コントローラーを動かすと本体が同じ動きをするロボットアームです。
本体とコントローラーは3Dプリンターで印刷しており、3Dモデルは3DCADと呼ばれるソフトで作成しています。





こんな見た目です。

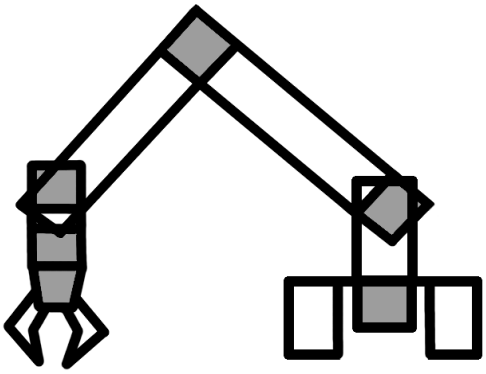
仕組み

Arduinoと呼ばれる制御基板にプログラムを書いて制御しています。
コントローラーに付いている可変抵抗の角度をArduinoが認識し、本体のサーボモーターを同じ角度にするというだけの単純なプログラムです。
サーボモーターの速度制御のために「VarSpeedServo.h」というライブラリを使用しています。
ライブラリは使用できるプログラムを増やすためのプログラム集みたいな認識でいいです。
もともとArduino IDEに内蔵されていないライブラリはWebでダウンロードして使えるようにする必要があります。

以下プログラム

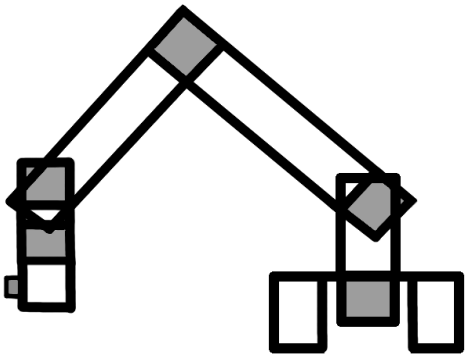
```
#include <VarSpeedServo.h> // ライブラリをインクルードすることで使用できるようになる

VarSpeedServo sabo1; // サーボの宣言
VarSpeedServo sabo2;
VarSpeedServo sabo3;
VarSpeedServo sabo4;
VarSpeedServo sabo5;
VarSpeedServo sabo6;

int An1 = A0; // アナログ入力ピンの設定
int An2 = A1;
int An3 = A2;
int An4 = A3;
int An5 = A4;
int angle1; // 角度の変数の設定
int angle2;
int angle3;
int angle4;
int angle5;
int buttonpin = 2;

void setup() {
  sabo1.attach(9); // サーボを動かす出力ピンの設定
  sabo2.attach(10);
  sabo3.attach(11);
  sabo4.attach(3);
  sabo5.attach(5);
  sabo6.attach(6);
pinMode( buttonpin, INPUT_PULLUP );
}

void loop() {
  angle1 = analogRead(An1); // 可変抵抗を読む(0~1023)
  angle2 = analogRead(An2);
  angle3 = analogRead(An3);
  angle4 = analogRead(An4);
  angle5 = analogRead(An5);
int button=digitalRead( buttonpin );
  angle1 = map(angle1, 0, 1023, 0, 180); // 読んだ抵抗値(0~1023)をサーボの角度(0~180)に変換
  angle2 = map(angle2, 0, 1023, 0, 180);
  angle3 = map(angle3, 0, 1023, 0, 180);
```

```
  angle4 = map(angle4, 0, 1023, 0, 180);
  angle5 = map(angle5, 0, 1023, 0, 180);
  sabo1.write(angle1,30, false); // 読んだ抵抗値によってサーボを動かす
  sabo2.write(angle2,30, false);
  sabo3.write(angle3,30, false);
  sabo4.write(angle4,30, false);
  sabo5.write(angle5,30, false);
if (button == HIGH)
{
sabo6.write(0, 30, false);
}
else  if (button == LOW)
{
sabo6.write, 80, 30, false);
}

  delay(5);
}
```



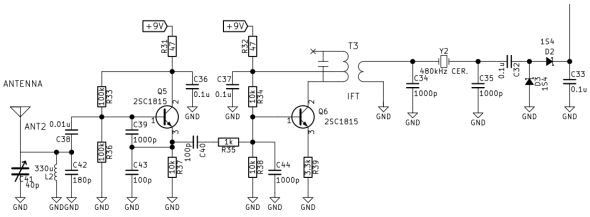
図（本体）

灰色の部分がサーボモーター(MG996R)で合計六か所付いており、それぞれのサーボモーターが制御基盤へ繋がっています。

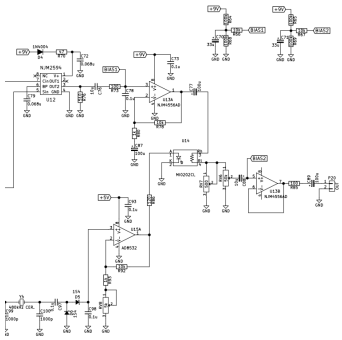


図（コントローラー）

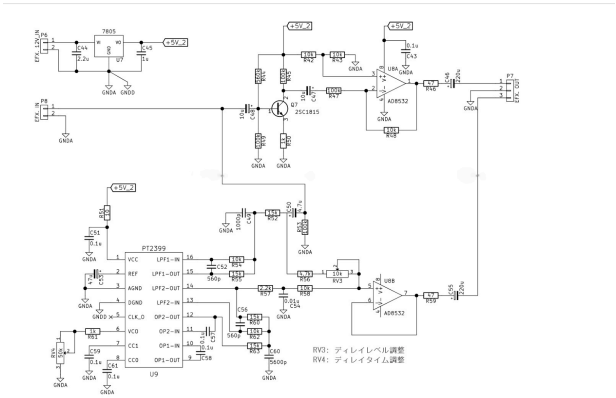
灰色の部分が可変抵抗で5か所に付いており、先端の灰色の部分のみ可変抵抗では無くスイッチが付いています。そしてサーボモーターと同様にそれぞれの可変抵抗とスイッチが制御基盤へと繋がっています。



↑音量を制御する回路です。ここでは発振周波数の変化を直流電圧の変化に変換しています。



↑音程制御の回路で取り出した周波数を音声信号にしています。また、音量制御の回路からの電圧変化を抵抗値の変化に変えています。



←エフェクターの回路です。下半分がディレイの回路で、上半分がディストーションの回路です。

・感想

パソコンを使わない製作をしたいなと思いテルミンを作りました。でもアナログの知識が全然無かったので大変でした。来年はこれを活かした製作をしたいです。

ロボット

21式制御器同期機械義手「式神」

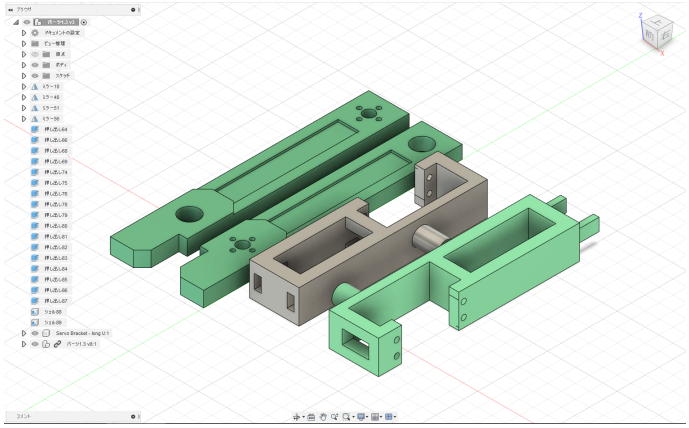
製作者　H2　吉田
M3　伊藤・秋田
協力　物無の皆様

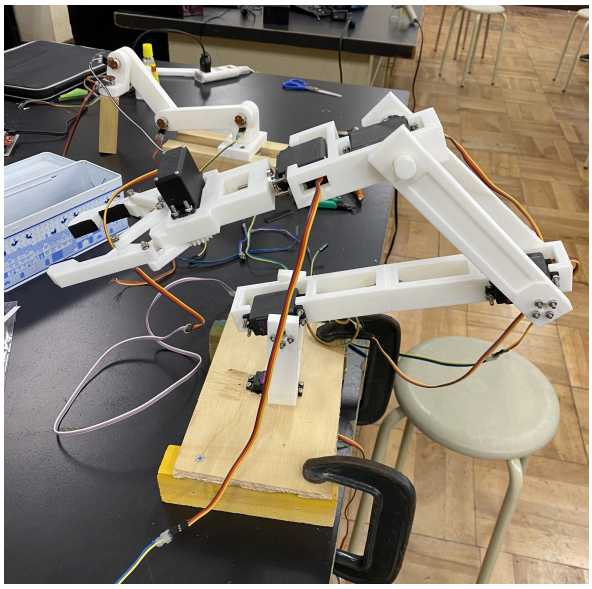
ストーリー

　ある夏、物無員のＩ藤は多汗症に悩まされていた。
「駄目だ！手汗のせいでペンも半田ごても滑ってマトモに使えない！」
「そうだ！自分の手の代わりに物を掴んでくれるロボットアームを作れば良いんだ！」
そしてＩ藤はY田さんとA田を仲間に入れ、多汗症治療の薬がある事を知らないままロボットアームを作り始める…

概要

コントローラーを動かすと本体が同じ動きをするロボットアームです。
本体とコントローラーは3Dプリンターで印刷しており、3Dモデルは3DCADと呼ばれるソフトで作成しています。





こんな見た目です。

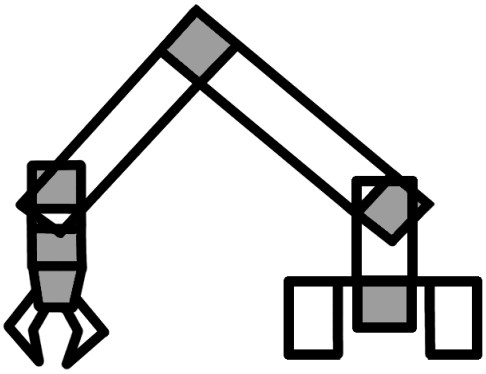
仕組み

Arduinoと呼ばれる制御基板にプログラムを書いて制御しています。
コントローラーに付いている可変抵抗の角度をArduinoが認識し、本体のサーボモーターを同じ角度にするというだけの単純なプログラムです。
サーボモーターの速度制御のために「VarSpeedServo.h」というライブラリを使用しています。
ライブラリは使用できるプログラムを増やすためのプログラム集みたいな認識でいいです。
もともとArduino IDEに内蔵されていないライブラリはWebでダウンロードして使えるようにする必要があります。

以下プログラム

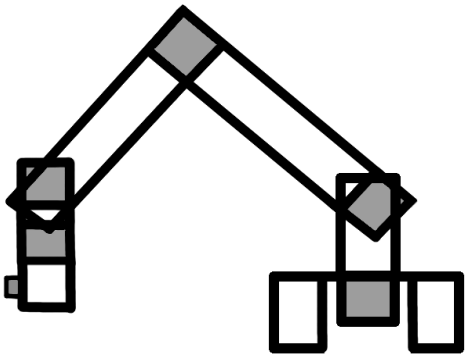
```
#include <VarSpeedServo.h> // ライブラリをインクルードすることで使用できるようになる

VarSpeedServo sabo1; // サーボの宣言
VarSpeedServo sabo2;
VarSpeedServo sabo3;
VarSpeedServo sabo4;
VarSpeedServo sabo5;
VarSpeedServo sabo6;

int An1 = A0; // アナログ入力ピンの設定
int An2 = A1;
int An3 = A2;
int An4 = A3;
int An5 = A4;
int angle1; // 角度の変数の設定
int angle2;
int angle3;
int angle4;
int angle5;
int buttonpin = 2;

void setup() {
  sabo1.attach(9); // サーボを動かす出力ピンの設定
  sabo2.attach(10);
  sabo3.attach(11);
  sabo4.attach(3);
  sabo5.attach(5);
  sabo6.attach(6);
pinMode( buttonpin, INPUT_PULLUP );
}

void loop() {
  angle1 = analogRead(An1); // 可変抵抗を読む(0~1023)
  angle2 = analogRead(An2);
  angle3 = analogRead(An3);
  angle4 = analogRead(An4);
  angle5 = analogRead(An5);
int button=digitalRead( buttonpin );
  angle1 = map(angle1, 0, 1023, 0, 180); // 読んだ抵抗値(0~1023)をサーボの角度(0~180)に変換
  angle2 = map(angle2, 0, 1023, 0, 180);
  angle3 = map(angle3, 0, 1023, 0, 180);
```

```
  angle4 = map(angle4, 0, 1023, 0, 180);
  angle5 = map(angle5, 0, 1023, 0, 180);
  sabo1.write(angle1,30, false); // 読んだ抵抗値によってサーボを動かす
  sabo2.write(angle2,30, false);
  sabo3.write(angle3,30, false);
  sabo4.write(angle4,30, false);
  sabo5.write(angle5,30, false);
if (button == HIGH)
{
sabo6.write(0, 30, false);
}
else  if (button == LOW)
{
sabo6.write, 80, 30, false);
}

  delay(5);
}
```



図（本体）

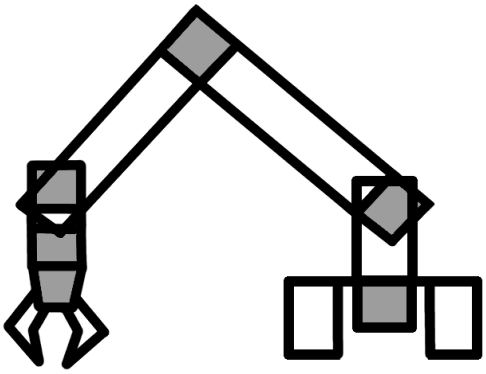
灰色の部分がサーボモーター(MG996R)で合計六か所付いており、それぞれのサーボモーターが制御基盤へ繋がっています。



図（コントローラー）

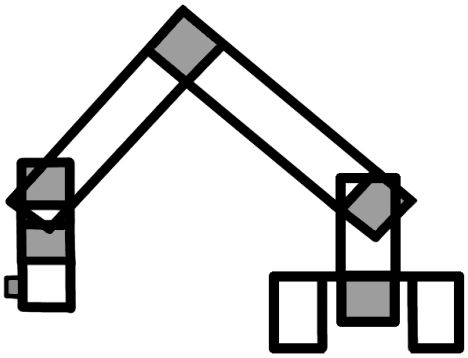
灰色の部分が可変抵抗で5か所に付いており、先端の灰色の部分のみ可変抵抗では無くスイッチが付いています。そしてサーボモーターと同様にそれぞれの可変抵抗とスイッチが制御基盤へと繋がっています。

```
  angle4 = map(angle4, 0, 1023, 0, 180);
  angle5 = map(angle5, 0, 1023, 0, 180);
  sabo1.write(angle1,30, false); // 読んだ抵抗値によってサーボを動かす
  sabo2.write(angle2,30, false);
  sabo3.write(angle3,30, false);
  sabo4.write(angle4,30, false);
  sabo5.write(angle5,30, false);
if (button == HIGH)
{
sabo6.write(0, 30, false);
}
else  if (button == LOW)
{
sabo6.write, 80, 30, false);
}

  delay(5);
}
```



図（本体）

灰色の部分がサーボモーター(MG996R)で合計六か所付いており、それぞれのサーボモーターが制御基盤へ繋がっています。



図（コントローラー）

灰色の部分が可変抵抗で5か所に付いており、先端の灰色の部分のみ可変抵抗では無くスイッチが付いています。そしてサーボモーターと同様にそれぞれの可変抵抗とスイッチが制御基盤へと繋がっています。

ARCS

Auto Rubik’s Cube Solver

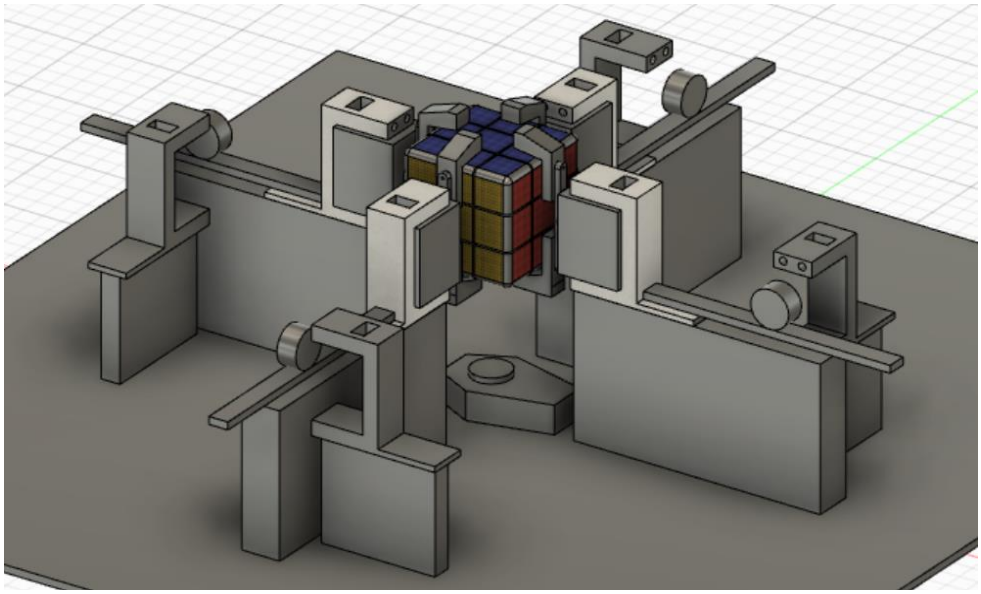
<Members>

H1 畠山由輝
M3 大谷 心雲

〈Story〉

突如物大に Rubik’s Cube が届いた。どうやら秘密結社 BTM からの機密暗号らしい。
この Rubik’s Cube を解けば機密情報が解読できるようだが…。我々物無員なら
Rubik’s Cube を解くなど造作もない…はず…。うーん…。我々は思いついた。
俺たち物無員なら頭どころか手すら使わずに Rubik’s Cube を解いてやるぜ！

〈Overview〉

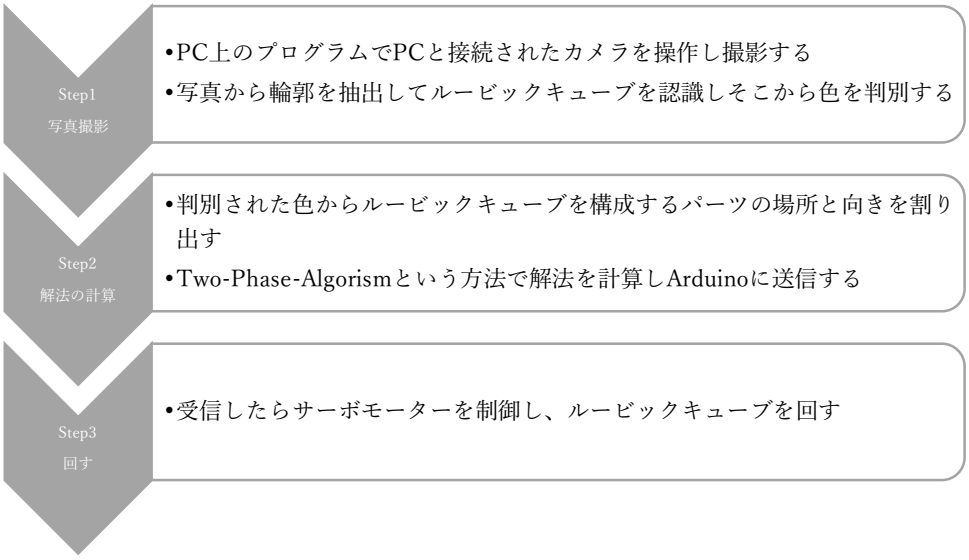


機会の中心にルービックキューブを置き、その下のカメラで下の面の 9 か所の色を

判定する。これを 6 回繰り返すことで揃っていないルービックキューブのすべての面の色を読み込む。プログラムでこのルービックキューブの解法を割り出し、全 8 か所のサーボモーターによってルービックキューブの面が回転し、揃えることができる。

<How Arcs Works>

ARCS ではサーボモーターの制御を Arduino というマイコンが行い、画像解析及び解法の計算を PC で行っている。また両者はシリアル通信という方法で通信している。
ルービックキューブを解くまでを流れ図で説明する。



Step1 と Step2 は PC 上の Python で書かれたプログラムで行っている
Step2 での解の計算をするプログラムはインターネットで見つけた物のほぼ丸写し

＜かんそう＞
畠山
僕はプログラム部分を担当しました。Python という言語を一から勉強しなんとかプログラム部分のほとんどは完成させることができました。また Unity というゲームエンジンを使いルービックキューブを解くアニメーションの製作も進めています。今回の製作では多くの種類のプログラミング言語に触れるとこができました。来年も今年の経験を生かして最後の製作を頑張ろうと思っています。

大谷
私は主にロボットの本体の製作を担当しました。3Dプリンターで部品を作る時があったのですが、そもそもモデリングが苦手で上手くいかなかったり、部の3Dプリンターが壊れて印刷できないなどの問題がありました。なんだかんだ私は(趣味で)3Dプリンターを買ったので解決しました。とにかく、モデリングの技術はすごく上がったかなと思います。総じて、学びは多かったです。実は少しだけArduinoも触れていたのですが来年以降はプログラミングを頑張りつつ、モデリングを極めたいと思います。

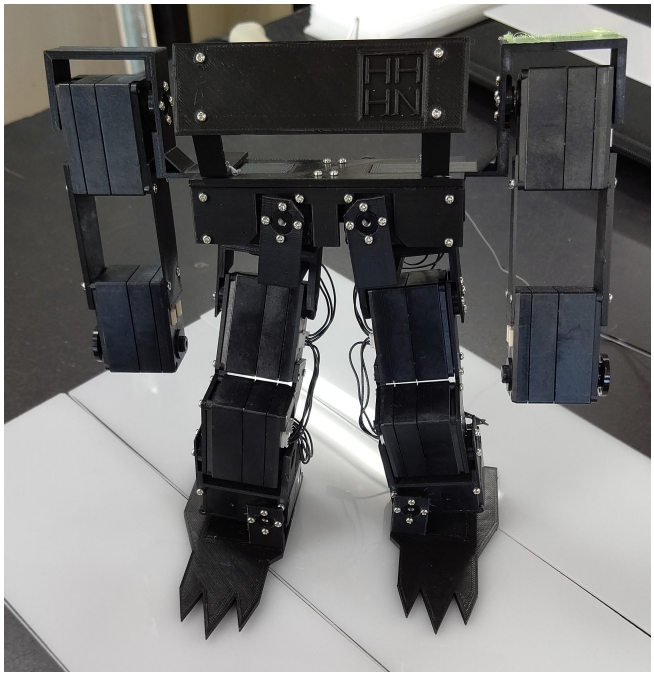
walker

製作:H2中村 M3広瀬
協力:物無の皆様

・あらすじ

トランス◯ォーマー → トランスフォーカー → トランスウォーカー → ウォーカー

外観



・概要

二足歩行ロボットです。あらすじに書いてある通り最初は車に変身する二足歩行ロボットを目指していましたが、最終的に単純な二足歩行ロボットに落ち着きました。
腕に4個、足に8個で計12個のサーボモーターを使用しています。

・サーボについて

近藤化学のKRS-2552RHVというサーボを使用しています。このサーボはPWM制御ではなくシリアル通信で制御しているため、配線の必要がほとんどありません。トルクも14.0kgf・cmなのでかなり安定して動かすことができます。

KRS-2552RHV→

・動きについて

前方向と左右方向に歩けます。前方向にはゆっくりと片足をあげ歩き、左右方向にはカニ歩きで移動します🦀

・通信について

TWILIGHTという簡単に赤外線通信ができるモジュールを使っています。ゲームのコントローラーの中に組み込んだため見た目もスッキリしています。

・感想

当初作ろうとしていた変身する二足歩行ロボットにはなりませんでしたが、無事に動いてよかったです。足のサーボモーターを増やせばもっと安定して動かすことができたと思います。(中村)

売り物に時間をかけすぎて一ヶ月程しか製作に参加できませんでしたが、無線通信やコントローラー、モーションの製作はできたので良かったです。aruduinoの使い方を学べたので来年に活かしたいと思います(広瀬)

あれ、僕またなんかやばい戦車つくっちゃいましたか？

（コイルガン戦車）

製作者：高１　谷中　中３鳥内、榎本（亘）

すとーりー

20XX 年のある時に、荒廃した地球にある車両を造った技術者集団が名を馳せた（去年の回路図集参照）。そのグループの古参であったＴ中研究員は、海へと進出していったＮ澤研究員やロボットアームにつかまれ、連れていかれた大賢者Ｙ田さん、指定暴力団『Ｔ緑会』の幹部となり有名大学を目指すＳ田研究員との別れの後、新たな技術者の育成をしていた。そこで新型車両を開発していたある日、期待の星、Ｔ内研究員があることを提案したのであった…

Ｔ内「あれ、これ無理やり百均 BB ガン積んで撃つよりコイルガン積んで撃ったほうがよくないですか？？」

Ｔ中「　!!?　そこに気づくとは…やはりきみは天才だッ!」

Ｔ内「…無線にしたらコイルガンの磁力邪魔になるかもしれませんけど…」

Ｔ中「たぶんヨシ！どうにかなるだろ！知らんけど！」

我らが総務さん「コイルガンが武器として扱われないのは作る人たちのモラルがあるからなんだよ」
（みなさんも安全に気を付けましょう）

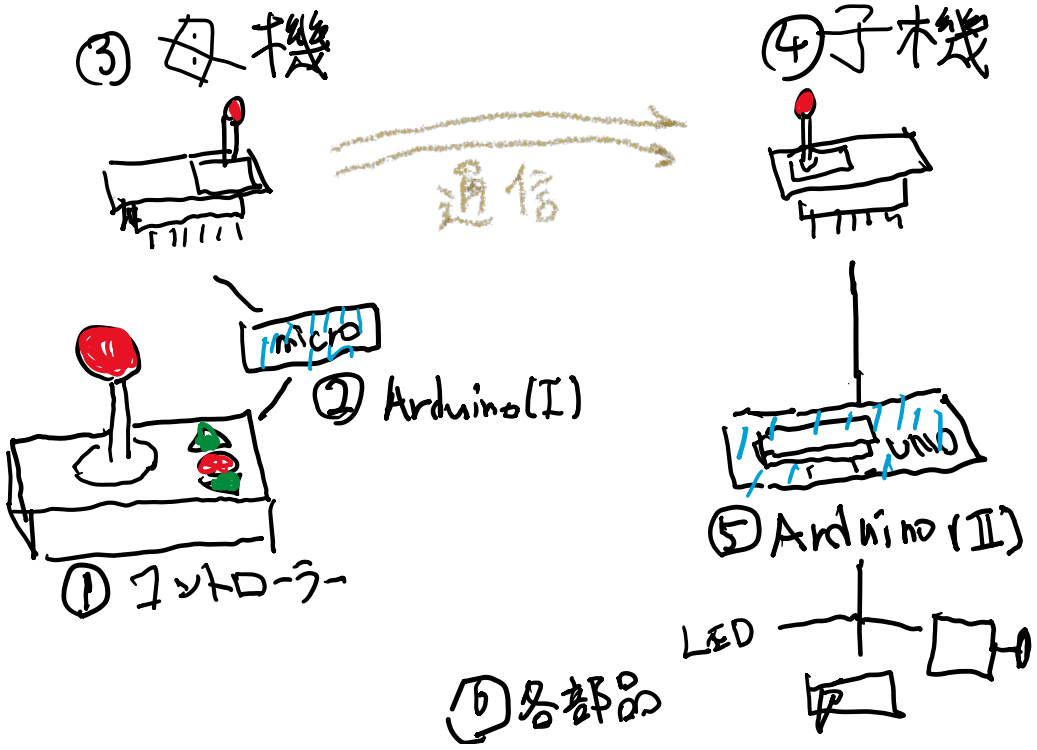
がいよう

去年あった戦闘車でしたが、車輪で走っていたので、今年はキャタピラを使って走らせます。ちなみにキャタピラは会社の名前で正式な名前は無限軌道だったりします。去年のモチーフは自衛隊の 16 式でしたが、今年はスウェーデン陸軍の駆逐戦車、Strv.109 です。特徴として砲塔がなく、主砲も完全に固定されており俯角・仰角がとれない代わりに油圧式サスペンションというもので車体丸ごとを砲塔のように上下左右に動かして、敵を狙うという、「走る砲塔」というコンセプトの下で開発された戦車です。それにコイルガンをのっけてそれを無線通信で動かそうという感じです。夢は物無の制作物を全部倒すことです。今年は陸海空すべてそろっているので展示が終わったらヘボコンバトルして全部破壊するつもりです。車体はすべて 3D プリンター製で、制御には Arduino、通信は Twe-Lite を使用します。

つかうぶひん

3D プリンター（樹脂積層型）、TAMIYA のキャタピラセット、Arduino UNO×2、スイッチ×たくさん、DC モーター×2、サーボモーター×4、自作コイル（銅）、コンデンサー、昇圧回路、Twe-Lite×2etc...

しくみ（本体）



① コントローラー：ここから H、L の信号を出します。ボタンが光ってるはずです。ちなみに、『ミサイルスイッチ』と『LED ボタン』のどっちで行うか迷って現在のものになりました。

② Arduino（Ⅰ）：ここでコントローラーから出された信号を処理して二進数に変換します。PIC などでも同様のことができますが、今回は時間的問題と技術の不足でこうなっています。

③ 母機（Twe-Lite）：これは入力された四桁までの二進数の数を発信する『母機』側です。1500 円で簡単に無線通信が行えて魅力的です（媚）。

④ 子機（Twe-Lite）：これは受信した二進数の数をそのまま出力します。ちなみに Twe-Lite についているモードピンの H、L を変えることで親機、子機の切り替えができます。

⑤ Arduino（Ⅱ）：この製作の心臓部です。Twi-Lite の出力の二進数に合わせて指示を出します。ちなみに使っているのは純正です、おっ金持ち～♪

⑥ 各部品：名前の通り各部品です。モーターや LED など様々です。手足みたいなもの。

しくみ（発射機構）

弾を撃つコイルガンです。コイルガンはコイルに大電流を一瞬だけ流すことで弾をコイル内に引き付け、その勢いで弾を飛ばすものです。昇圧回路を使って高い電圧を出し、それをコンデンサーに貯めてトライアックを用いて制御しています。

今回はコイルを二つ使っていて、一つ目のコイルを弾が通過したらそれをフォトトランジスタを用いて認識させ、二つ目のコイルに電流を流す仕組みになっています。

※危険なのでちゃんと仕組みを理解してから製作しましょう

感想

今回の製作では僕は車体を作り、鳥内君には発射の機構を作るという形にしました。僕が忙しいのもあって、あんまり指示はしませんでしたが、ちゃんとやってくれてると思います。ただ、ご時世のせいや、僕が忙しくて自分の固定の机を持てなかったのもありあんまり話をしたり、飯屋に行くことなどができなかったことが非常に悔やまれます。去年の僕の製作班の先輩方は忙しいのにいろいろな話をしてくださったので、より一層申し訳ない。この場を借りて謝ります。ごめんね鳥内。でも、まだあともう一年あるからこれからもよろしく。今後は君にもっと粘着していく気でいきます。

榎本（亘）君については、いろいろあって会えてなくて寂しいね。

そして重要なことがもう一つ。これを書いている現在、いまだ車体すら印刷できてません。状況が状況なので、どこまでいけるかも不透明ですが、絶対に完動させるぞー！えい・えい・むん！

（最近ジョジョを五部まで読んだため絵の中でスタンドを使っている）谷中

入り電と並行で製作しているのもあるんですが、進捗があまり出せていないのが申し訳ないです。ただ、初めての共同制作で結構やりたいことがやれて楽しかったです。　鳥内

エンド・オブ・ザ・ストーリー

こうして、ついに無限軌道を持つ車両が爆誕したのである。やったぜ。
完成した戦車を前に、物無員たちの口からは言葉が出ない。非常に驚かされているのだ。無理もない。

T内「また僕なんかやっちゃいましたか？」

彼の製作は続く…

T内's BIZARRE PRODUCTION

（鳥内の奇妙な製作）

第二部（完）

最後までネタ全開でゴメンね！

AZAX LEGENDS

H2 徳田　H1 山田

ストーリー

？？さぁ。敵がいたぞ。

　　ドローン視点に切り替える。

　　ハードウェアリブートを実行する！

　　攻撃中。

　　敵を撃破。敵部隊はこれで全滅だ。

概要

ストーリーはフィクションです。Ap＊xLegends ではありません。

　というわけで、ドローンのようなロボットを紹介したいと思います。このロボットは、四枚のプロペラをモーターで回すことにより空を飛び、センサーを使って機体の傾きを検知することで、空中でホバリングしたり回転したりします。遠隔操作は出来ないのでドローンとは呼べませんが(9 月 16 日現在)、空中での飛行を行うことができるロボットとなっています。

部品

Arduino nano… プログラムを書き込むための装置です。従来の arduino よりも小さく軽量であり、特に今回のような空を飛ぶロボットには重量が大きく関わってくるため、こちらを使用しました。

DC モーター…amazon で売っていたものです。右回転と左回転のモーターがあり、それぞれ 2 つずつ使っています。

モータードライバ DRV8835…arduino とモーターの間に繋いで PWM 制御を行うために使用します。

6 軸加速度センサーMPU6050…機体の角度を算出し、角度制御を行うために使用します。

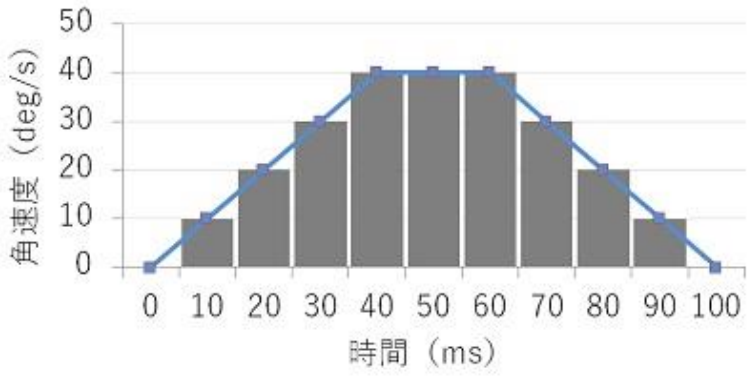
1.5v 電池…2 つ直列につなぎ、モーターの 3v 電源として使用します。

仕組み

　このロボットは本体についている 4 つの DC モーターを Arduino nano からの PWM 信号によって制御しています。また、角度の制御を MPU6050 という 6 軸のジャイロセンサーによって機体の傾きを算出することで行なっています。ここで、このロボットがどのようにして飛ぶのか説明したいと思います。

　まず、機体を空に飛ばすためには対角線上にあるモーターをそれぞれ時計、反時計回りに

回転させる(今回は、モーターの回る方向がもともと決まっています)ことが必要です。このようにして、機体がその場で回転してしまうことを防ぎます。この状態で全モーターの回転数を上げると機体は上昇し、下げると下降します。そして機体左側のモーターの回転数を上げ、右側のモーターの回転数を下げると機体は右へ傾きます。左の場合は逆です。
また、前後の移動は機体前後のモーターの回転数を上げ下げすることで行えます。
　次にこのロボットのプログラムについて説明したいと思います。簡単に説明すると、機体が水平方向に対してどれだけ傾いているかをジャイロセンサーで計測し、そのデータをモーターの回転数に反映させるプログラムとなっています。
また角度制御に関しては、機体が水平方向からどれだけ傾いているかをジャイロセンサーによって計測し、それを基にしてモーターの回転数を制御するプログラムとなっています。
これが、ジャイロセンサーから機体の角速度、加速度を計測するプログラムです。
ジャイロセンサーからは、この二つの情報しかえることが出来ません。ここからどのようにして機体の角度(傾き)や速度を求めるかというと、二つの値を積分します。これは、
位置→(微分)→速度→(微分)→加速度
位置←(積分)←速度←(積分)←加速度　の関係であることに基づいています。
しかし、ここで arduino では四則演算しか出来ず、積分の概念がないという問題点があります。そこを打開するために、データを検出する間隔を短くしそれぞれの値を足し続けることで、本来の積分に近いプログラムとしました。(下図)




```
  //+=について・・・  x +=  2;はx = x + 2  と同じ
 yaw    += gz * 131;
 //角度degreeを求めるために角速度を積分する。具体的には、角速度に時間をかけて足し続ける
  roll  = atan2(ay, az) * RAD_TO_DEG;
  pitch = atan(-ax / sqrt(ay * ay + az * az)) * RAD_TO_DEG;
 Serial.print(roll); //算出した角度をシリアルモニタに表示する rollはx軸、pitchはy軸、yawはz軸の角度
 Serial.print("   ");//スペースを表示
 Serial.print(pitch);
 Serial.print("   ");
 Serial.print(yaw);
 Serial.print("    ");
}
```

感想

（山田）

今回初めて一人でロボットのプログラムを作りました。センサーから算出した角度とモーターを動機させる所が難しかったです。

（徳田さん）

制作期間中あまり参加できず山田くんにほとんどを任せてしまい申し訳ないです。すみませんでした。

RCMBによる海洋征服計画

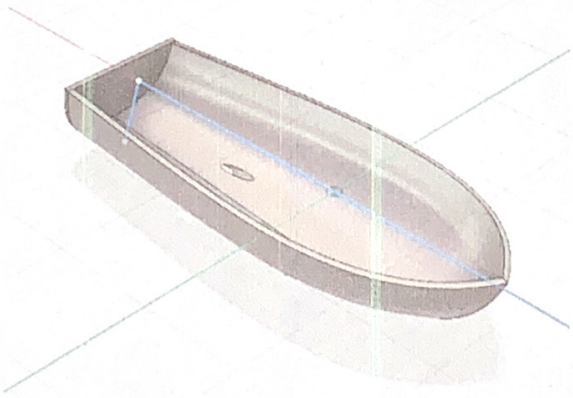
制作:H1　中澤凌太郎　協力:物無の皆様

ストーリー

私はN中将である。昨年私とT元少将とY元中将とS元大将は『㊙機動戦闘車開発計画』をもって、この広大なる物大連邦の大地の完全支配に成功した。更に私は計画の成功により、少将から中将へ昇格した。しかし、私はこれで満足しないのだ。陸を支配下に収めた次には大海原が私を待っている！海も征服しようではないか。新たな探査艇を開発し、制海権の確保を急がなくては。

概要

モーターボートです。無線コントローラーで前後左右自由に移動することが出来ます。車体は３Ｄプリンターで作りました。以下の画像は開発中の本体のデザインです。



使用部品

・Ａｒｄｕｉｎｏ　Ｎａｎｏ（コンピュータ）
・ＤＣモーター
・モータードライバ　ＴＡ７２９１Ｐ
・シャフト
・小型スクリュー
・ジョイスティック
・Ｂ基盤
・ＴＷＥＬＩＴＥ（無線通信マイコン）

仕組み

ジョイスティックにはＸ軸とＹ軸があり、倒れている方向によってそれぞれ０から１０２４までの値を取ります。倒していないときの値（初期状態）は（５１２，５１２）です。コントローラー側のＡｒｄｕｉｎｏで二つの軸の値の変化を読み取ります。例えば、前に倒すとＹの値が減少する変化が生じます。ＸとＹの値の組み合わせによって、Ａｒｄｕｉｎｏが二進数を出力します。

例：ジョイスティックが前に倒されて、Ｙの値が０になった
　　→ボートを前進させるための二進数０００をＡｒｄｕｉｎｏが出力。

出力された二進数は無線マイコンによって本体側のＡｒｄｕｉｎｏに送信されます。Ａｒｄｕｉｎｏは送られた二進数に応じてそれぞれ異なる命令を出します。コントローラーから送られた値が０００だったら前進、００１だったら後退、０１０だったら左旋回といった感じです。本体のＡｒｄｕｉｎｏから出された命令は、モータードライバを経由してＤＣモーターへと送られます。

このモーターボートはＤＣモーターを二個備え付けているので、片方のモーターだけを止めれば旋回も可能です。

ところで、この制作物は船である以上絶対に沈没してはなりません。そのため浮力の計算も必要になります。アルキメデスの原理より、浮力は物体が押しのけた液体の重さと等しいです。もしこの船の体積より、回路などを含めた重さが上回ったら船は沈没します。そんなことにならないように、船体はできるだけ大きくして、可能な限りの回路部分の軽量化を施してあります。

感想

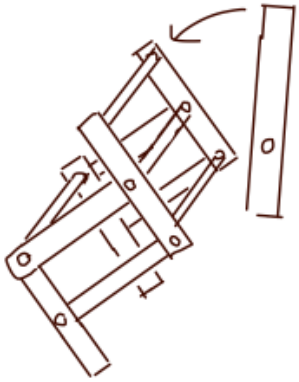
全てが試行錯誤を繰り返しでした。プログラムのでーたがすべて消えるなどのトラブル発生にも見舞われて、一からまた書き直すのは本当に骨が折れましたが、僕個人で問題を解決したことにより自主性が更に身についたと思います。

GAGO

製作者：H2　高角　M3 加藤　　協力：物無の皆様

ストーリー
20XX 年、地球ではとあるウィルスが蔓延していた。 このまま何もせずウィルスの餌食になるわけにはいかない。 そこで我々はロボットに乗り、地球から脱出する計画を打ち立て制作に取り掛かった……

概要
ジャンプして移動するロボットです。
棒が回転し、連結バーを押し込むことでバネを縮ませ、連結バーが棒から離れることでその力を解放してジャンプします。



モーターの回転の制御には Arduino を使っており、回転する力を増すためにギヤを使っています。

機構
機構については色々と実験をしました。
初めは引張バネをつけたユニットを紐で引っ張ることでバネが伸び、解放することで飛ぶ一足歩行型ロボットを目指していたのですが、モータのトルクが予想以上に足りなかったこと、バネを伸ばしても大した跳躍力を得られなかったことから、この形に落ち着きました。

感想
高角：飛べる機構を見つけるために色々と試行錯誤をするのが大変でしたが楽しかったです。加藤君、散々振り回してごめん(*_ _)
ちなみに、タイトルに特に意味はありません。
加藤：構造など大半を頼ってしまいました。ただ、プログラム、モデリング等のある程度の知識は教えてもらい、得ることが出来たのでとても良かったです。

0

01000010

01000010 01010100

01000010 01010100 01001101

01000010 0101 BTM 0100 01001101

01000010 01010100 01001101

01010100 01001101

01001101

0

0

01000100

01000100 01001001

01000100 01001001 01000111

01000100 01001001 01000111 01001001

01000100 01001001 01000111 01001001 01010011

01000100 01001001 01000111 01001001 01010011 01000001

01000100 01001001 01000111 01001001 01010011 01000001 01001100

01000100 01001001 01000111 0100 DIGITAL 1001 01010011 01000001 01001100

01000100 01001001 01000111 01001001 01010011 01000001 01001100

01001001 01000111 01001001 01010011 01000001 01001100

01000111 01001001 01010011 01000001 01001100

01001001 01010011 01000001 01001100

01010011 01000001 01001100

01000001 01001100

01001100

0

0

01010101

01010101 01010101

01010101 01010101 01010101

01010101 0101 UUU 0101 01010101

01010101 01010101 01010101

01010101 01010101

01010101

0

44 65 73 69 67 6e 65 64 5f 62 79 5f 54 61 69 79 6f 5f 49 73 6f 6e 6f

ポケモン図鑑

H2 海野さん

M3 山田耕大

ストーリー

俺、マサラタウンのサトシ！

概要

カメラでポケモンを認識して名前を表示します。ボタンを押すとタイプや説明、鳴き声を確認できます。

仕組み

１,全体

Python というプログラミング言語で画像検出のデータセットを作り、

swift という言語でそれをスマホアプリに

組み込みます。今回は iPhone アプリに組み込みましたが、python で作っているため別の使い方もあります。

２,分類器(cascade)

分類器とはその名の通り分類するためのものです。

画像を元に深層学習することで入力画像から目的のものを検出します。

詳しくは割愛しますが、下準備として

- 正解(今回はポケモン)の画像たくさん
- それとは関係ない、不正解の画像たくさん
- 正解画像のアドレスとサイズ
- 正解画像をさらに加工して作った vec ファイルというもの
- できた cascade を保存するフォルダ

などが入ったフォルダを用意します。

これを用意したら、AnacondaPrompt で

```
opencv_traincascade.exe -data ./(cascade を保存するフォルダ名)/
-vec ./vec/(vec のフォルダ名).vec -bg ./(不正解画像のファイル
名).txt -numPos（正解の枚数）-numNeg（不正解の枚数）
```

を実行すると検出器が完成します。

つぎにこの検出器をテストします。

```
import shutil
import os
import ｃｖ２
p_001_cascade = cv2.CascadeClassifier("cv/cascade/cascade.xml")
cap = ｃｖ2.VideoCapture(0)
```

```
p1 = 0
for i in range(100):
    ret, rgb = cap.read()
    p001 = p_001_cascade.detectMultiScale(rgb)
    for (x,y,w,h) in p001:
        cv2.rectangle(rgb,(x,y),(x+w,y+h),(255,0,0),2)
        p1 = p1 + 1
```

このコードを実行すると 100 フレームあたり何回検出したか教えてくれます。関係ない画像を見せた時に検出数がほぼ 0 になるように(主に不正解画像の枚数)修正します。

３,アプリ

iPhone のアプリを作りました。Mac の Xcode というものを使うと無料でアプリを作ることができます。Xcode は Swift という言語で動いていて、python でつくった cascade 分類器を用いるにはいくらかの手順が必要です。

アプリの機能としては、カメラの画像を取得しそれを画面表示。認識されたものを四角で囲いその名前をボタンに表示します。

ボタンを押すと画面遷移され、認識されたポケモンの説明欄に移ります。ここでは画面遷移以前の値を画面遷移後に渡してそれに基づき画像

と説明文の表示を行います。

ボタンを押すと鳴き声を聞くことができるようにしましたが、ここで、UI ボタンを設置しそれにアクションとして音を出すプログラムを組み込んだところ動かなかったため(画面遷移後であったことが理由か？)プログラミングからボタンを画面に作成し、それに音を再生する機能を持たせました。

感想

自分でも書いているコードの意味が分からなかいようなことが多かったので、勉強不足を痛感しました。初めてなりにもうちょっと自分から調べれたらよかったと思います。反省。

(山田)

この世に存在するすべてのものは不完全だと前回の製作で気が付いたことや、python で画像認識をやっていたことからもポケモン図鑑を作ろうということになりました。共同制作者を募ったら昼バレーの後に山田君が一緒にやってくれるということなのでそうなりました。色々とプログラムについて教えましたが、その製作の仕方の本質を理解し使えば(伝わっていれば)アナログだろうとデジタルだろうとロボットだろうと何でも活躍できるはずです。頑張ってください。　　(海野)

iWatch

製作: M3 和田真爾、H1 礒野大洋

感謝:磯村さん、物理部無線班の皆様

ストーリー

あなたの時間は限られている。
だから他人の人生を生きたりして無駄に過ごしてはいけない。
ドグマ（教義、常識、既存の理論）にとらわれるな。
それは他人の考えた結果で生きていることなのだから。
他人の意見が雑音のようにあなたの内面の声をかき消したりすることのないようにしなさい。
そして最も重要なのは、自分の心と直感を信じる勇気を持ちなさい。
それはどういうわけかあなたが本当になりたいものをすでによく知っているのだから。
それ以外のことは、全部二の次の意味しかない。

これは、Apple Watch を開発したスティーブ・ジョブズの言葉です。
しかし、彼は Apple Watch を発表する一年前に亡くなりました。
我々の時代の始まりです。

概要

複数の機能を搭載した時計型モジュールです。小型の液晶と、スイッチ、可変抵抗、WiFi、加速度センサー、RTC(リアルタイムクロック)モジュールを搭載しています。

操作は、スイッチ、可変抵抗、手首の角度で行います。幾つかのアプリケーションを搭載していて、時計、アラーム、サイコロ、ゲーム、ストップウォッチをメニュー画面で切り替えることができます。また、ゲームには、可変抵抗を使用して液晶上で遊ぶものと、iWatch 本体をコントローラーにしてブラウザ上で遊ぶものの２種類があります。

アプリ

時計

現在の時刻（日本時間）を表示します。常に NTP サーバー(要求を送るとほぼ正確な時間を返してくれるサーバー)に接続し、秒単位で時間を取得します。一応、アナログ時計モードとデジタル時計モードがあり、見た目を変えられるようにしています。

アラーム

分単位でアラームを設定し、その時間に液晶に「！」を表示します（音は鳴りません)。

ストップウォッチ

名前の通り、スイッチでスタート、ストップ、リセットを行い、RTC を用いて時間を計り

ます。

コントローラー

サーバーに接続し、ブラウザでゲームを遊ぶときなどに、コントローラーとして使うモードです。加速度センサーから手首の角度(X 要素、Y 要素共に 0 度から 180 度まで)を取得し、サーバーに送ります。

サイコロ

手首を振ると、液晶上にてサイコロの目がランダムに表示されたのち、ひとつに決まります。

ブロック崩しゲーム

液晶上に表示されます。C 言語で作ったブロック崩しゲームです。可変抵抗器のつまみを回してバーを操作し自機を跳ね返してブロックを壊すゲームです。自機(ボール)が下に落ちるとゲームオーバーです。

FPS もどき

html,javascript で作った FPS もどきです。腕時計の角度に応じて視点が動くシューティングゲームです。背景の色は明るい色の背景の上から暗い色の背景を描く、というのを繰り返して奥にいくほど暗くなるようにしています。

避けゲー

手首を傾けて自機を操作し、画面奥から迫ってくる赤い三角形(常に 10 個)をよけます。当たったら終了です。

部品・回路

部品

マイコンは ATmega328P が使われています。ボードは Arduino の互換ボードである Adafruit Pro Trinket 3V-12MHz(https://www.adafruit.com/product/2010)(以下 trinket)を使いました。このボードは、小型であるにもかかわらず、普通の Arduino とほぼ同等の機能を持っています。

WiFi モジュールは、ESP8266(=ESP-WROOM-02)です。Arduino として扱う事ができるので、サーバーへの要求(html や javascript のコード)をモジュール ESP 本体に書き込めます。非常に使いやすいです。

LCD は Sparkfun Micro OLED Breakout(Qwiic)を使っています。縦 48 横 64 の液晶で、青色に光ります。ライブラリが便利で使いやすかったです。

加速度センサーは KXR94-2050 を使用しています。X,Y,Z の３軸にかかる加速度をアナログ出力してくれます。「加速度」センサーという名前で角度を検出するのに使っているのはなぜかというと、地球上の物体には常に重力という一種の加速度がかかっていて、それが X,Y,Z のどの軸にどれくらいかかっているかを見ることによって、疑似的に角度がわかるからです。

RTCはRX-8900を使っています。これは、正確に時間を刻むことのできるモジュールであり、これ単体では正しい時間を出力することはできません。しかし、時間を刻み始める最初の時刻を設定できるので、もともとは一分ごとに NTP サーバーに接続してその時刻を再設定するつもりでした。

スイッチはタクトスイッチをプルアップして使い、可変抵抗は小さな固定抵抗を通してからtrinketに入力することにより、ショートを防いでいます。

回路(接続方法)

部品-接続方法-部品の形で並べていきます。

RTC(RX-8900)-i2c-trinket

LCD(Sparkfun Micro OLED Breakout(Qwiic))-i2c-trinket

ESP8266(=ESP-WROOM-02)-serial-trinket

加速度センサー(KXR94-2050):X軸-trinketのanalogピン(Y,Z軸も同様)

プログラム

※時計、アラーム、ストップウォッチ、サイコロのプログラムは書いておりません。

ブロック崩しゲーム

自機は現在の自機の座標に移動分の座標を足すことで動かしていて、壁等で反射させるときは移動分のx･y軸の正負を反転させています。ブロックの描画は二次元配列で縦5列、横8列の0か1の配列を作りfor文で配列の値が1の時にブロックが生成されるようになっていて、ゲームオーバー・クリア時には配列の中身をランダム関数で変えてブロックの配置がランダムになるようにしています。ブロックと自機の当たり判定関連はブロックの描画と同時に行っていてブロックの座標と自機の座標が重なった時にそのブロックの配列の値を0にすることでブロックを消し、自機を反射させて作っています。クリア判定はfor文で配列の値が全て0になっているかを調べてやっています。

FPSもどき

弾の移動はブロック崩しの自機の移動と同じ仕組みでやっていますが、移動量を割り算で少しずつ減らしています。弾が小さくなるのも同じようにやっています。弾の消失とリセットは弾の大きさで決めています。視点移動は背景の消失点の位置を動かして再現しています。これを書いている時点では未完成ですが敵の移動量と当たり判定は敵の大きさごとに分けて作るつもりです。

避けゲー

10 個の三角形の情報は、配列で管理しています。画面奥から迫ってくる三角形は実際は2Dで描画しているので、Z軸の座標を変えているのではなく、移動させつつ、大きくしています。また、遠くから飛んでくるイメージなので、三角形は、ウィンドウの大きさのn分の1の範囲（遠く）での三角形の座標を最初の地点とし、それをn倍したウィンドウの大きさの範囲（近く）での三角形の座標を目的地として移動するようにしています。また、10 個

一気に三角形が飛んでこないように、最初に出るタイミングを配列にしまっておき（[-10,-20,-30,...]というように)、これらを一律で+1 ずつしていって、0 になったら飛んでこさせるようにして、タイミングをずらしています。

サーバーとの通信(Websocket)

今回、web とハードがお互いにデータを送り合う必要があるということで、Websocket を使いました。Websocket とは、双方向通信に関して、HTTP プロトコル(コンピュータ同士が通信する際の約束、形式)では無駄が多かったということで作られたプロトコルのことです。しかし、リクエストやレスポンスの形式は HTTP に乗っ取ったものであり、双方向通信だけでなく、ハンドシェイク(通信の最初に行うもの)も簡単にできます。また、サーバーとクライアントのどちらかが「Ping」と送ったら、もう片方が自動的に「Pong」と送り返すようになっており、これで通信が途絶えていないか確かめることができます。

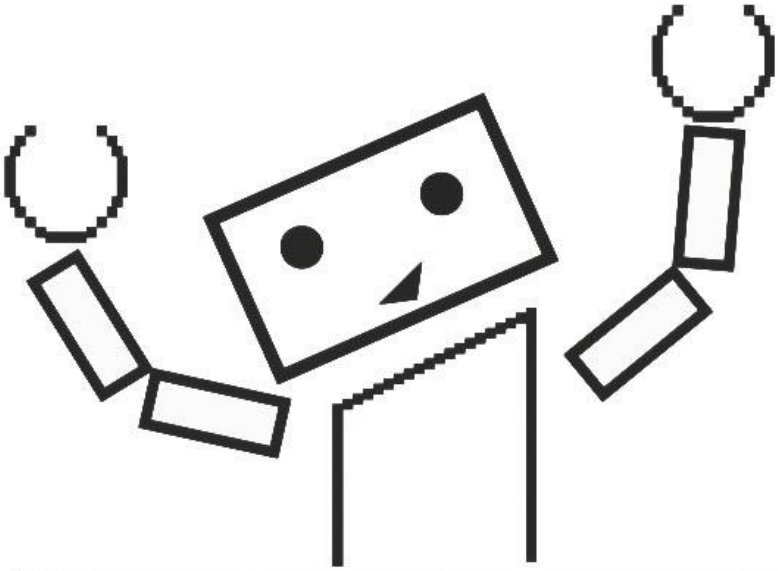
Web 上での図形の描画(Canvas API)

今回 web ゲームを作るにあたって、Canvas API を利用しました。これは、HTML 要素として canvas という図形の描画が可能な長方形を配置し、canvas 専用の関数(context.rectFill(x,y,x2,y2)など)でそこに任意の図形を描画できるというものです。

感想

和田: C 言語も html もほとんどやったことが無かったことや部活があまりできなかったこともあってか、半年に 2 つのゲームしか作れなかったことと勉強不足で磯野さんに迷惑かけたことが反省点だと思います。ブロック崩しは画面の大きさの関係などで自機の速度や角度を変えることができなかったため、FPS もどきでは細かい部分まで作れたことが嬉しかったです。

礒野:OB の方の製作を見て作り始めたもので、クオリティとしてはそれに勝るものはできませんでしたが、新しい技術を多く学べて、来年に確実に繋がる経験になりました。そして、積極的に製作してくれた和田くんにとても感謝しています。



＊部員が作ったゲームが
あらわれた

LV1　　HP ∞

BTM　GAME　AR　VR

バンバンシューティングVR

高２池田宜史　　中３永谷大成　　協力：物無の皆さん

ストーリー

ある日、愚かな中３のＮくんは黒板の掃除で手を抜いてしまう。ブチ切れた物理科はポケモンバトルをしかけてきた！！しかし Ｎ くんはポケモンを持っていない！！どうするＮ！？突然未来からやってきた Ｎ 谷に制作物のレールガンを授けられた永谷は自分自身で戦うことに！？！？！？！？！？！？！？！？！？！？！果たして勝負はどうなるのか！！！！！！！！！！！！！！

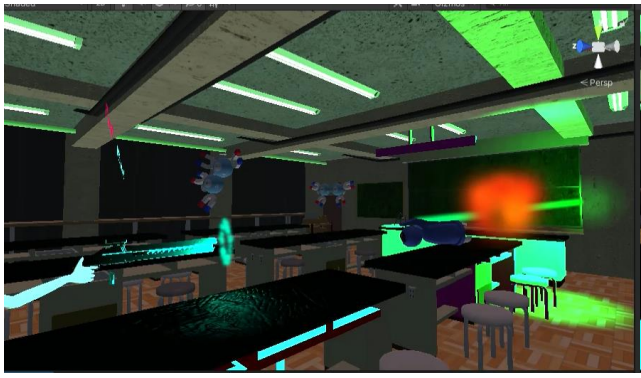
ルール

・４０秒以内にビーム銃を撃ってたくさん敵を倒そう！！
・バッテリーがなくなると撃てなくなるから気をつけよう！！
・たくさんポイントを稼ごう！！

概要

VR で物理大実験室を再現したステージに入り、センサー自分と連動させた腕を動かしてポケモンっぽい敵を倒します。
キャラは見た目が思いつかなかったので大好きなポケモンにしました。任天堂様ごめんなさい。

↓ゲーム画面



↓VR 視点

敵キャラ
（色違いは点数が高い）

使ったもの

・unity
・blender
・arduino
・９軸センサー（LSM9DS1）
・スマホ
・VR ゴーグル（elecom）

原理

　腕の動きの同期は、コントローラーの３つの９軸センサー(加速度センサー、ジャイロセンサー、磁気センサーを複合したセンサー)から二の腕、前腕、手の情報をそれぞれ arduino で受け取り、その情報を角度の形に変換した後、unity に送信、そしてそれをゲームに反映させることで実現しています。

　VR は、ゴークルにつけたスマホの角度を unity で受け取り、それをゲームに反映させ、そのゲームの映像をまたスマホに送り返すことで動かしています。

感想

池田　まず、共同制作者の永谷くんへの感謝でいっぱいです。彼の技術力は物無イチであることは確かです。感謝しています。彼は後輩ですが、頼りにできる後輩で、彼の優秀さに甘えてしまいました。この回路図集も彼の pc で書いています。反省の気持ちでいっぱいです。感謝の気持ちがとまりません。とても楽しい制作でした。すべての存在に感謝！！！

永谷　今年の制作では、ゲームの中身の方を作りました。unity とか色々勉強できて楽しかったです。池田さんには忙しいのに VR の仕組みとかを完全に預けてしまっていたのでそこは申し訳なかったです。それでも unity とかを色々教えていただいてほんとうにありがたかったです。あと誰にも言ってなかったのですが、文化祭２週間前くらいにゴーグルをなくしてしまって本当に焦っていました。一週間前には見つかってよかったです。物が無くなるのは物無っていう名前のせいなので早く改名したほうがいいと思います。

1000 年後のテレスクリーン

製作者 H1 高梨 M3 榎本

ストーリー　時は 2984 年。オセアニアでは死刑制度における執行者の精神的苦痛と人件費を無くすためにゴールドスタイン政権は全国のインターホンの中にランダムに死刑執行ボタンを設置するという政策を採決した。これに反感を抱いた市民ウィンストンは拡張現実化されたテレスクリーンを通してハッキングを仕掛ける事でゴールドスタインを暗殺しようとしていた…

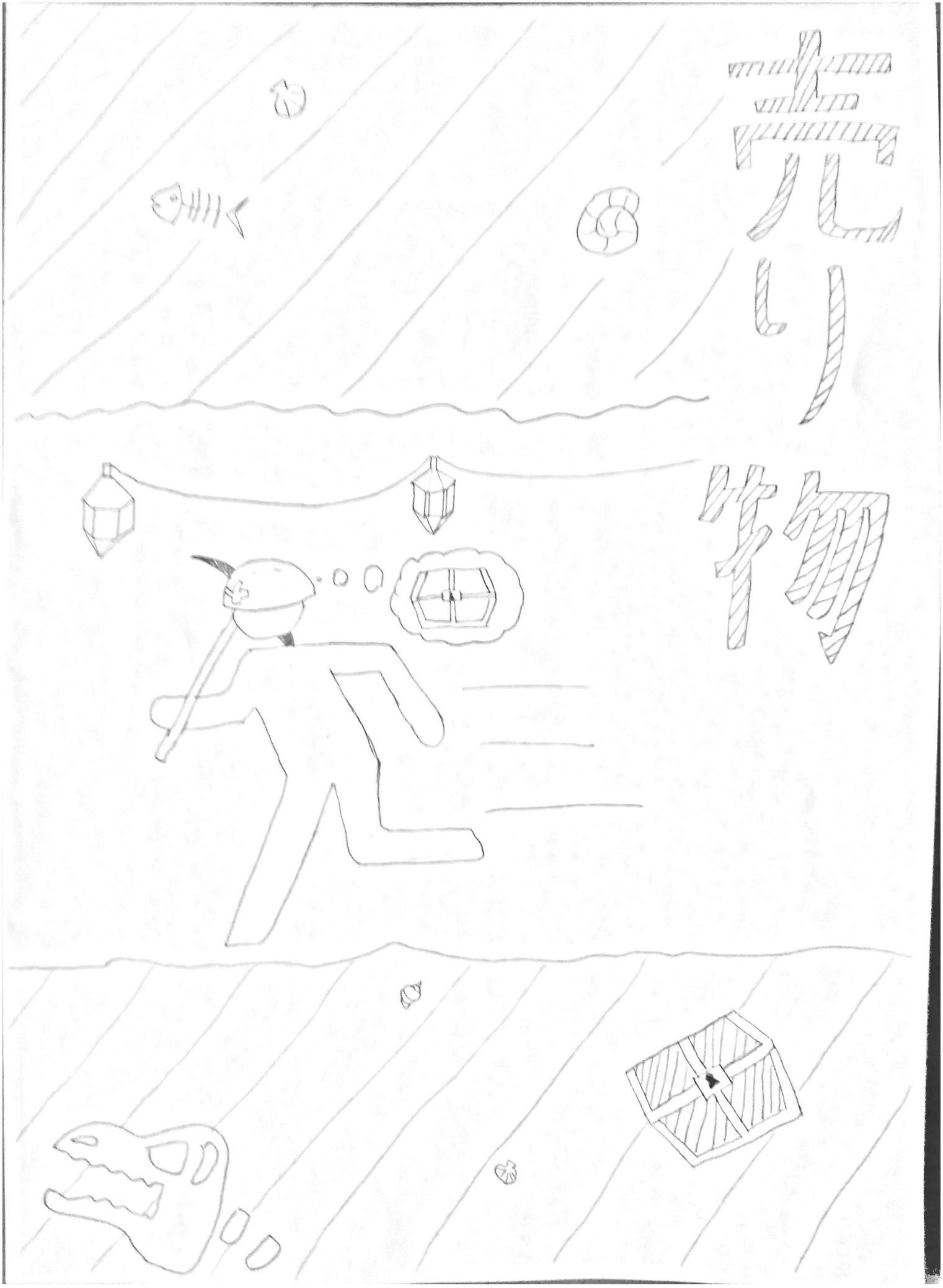
概要　ストーリーは某ディストピア小説に寄せただけで製作者の一切の政治的思想を表しているわけではありません。
ポケモンやドラゴンクエストなどの AR 技術を利用したスマホアプリが今年話題になって unity にも Vuforia や ARCore といった素人でも簡単に AR 技術を扱える API が現れたという事で何かそれを利用した製作をしたいなという事でとりあえず 3 つのゲームを作りました。

AR エアホッケー
Unity の Vuforia という API を利用しています。AR 技術にはあらかじめ登録したイラストなどを認識させてデジタル情報を表示させるマーカー型と、その場にあるものの位置や形を認識させてデジタル情報を表示させるマーカーレス型がありますがこの Vuforia はマーカー型に関する能力がとても優れています。Unity で作ったエアホッケーゲームに AR 用の座標を合わせて Vuforia のマーカートラッキングを使って現実世界にデジタル情報を表示させています。

顔シューティング（仮
AR 化のもととなるシューティングゲームの作成を最初に行った。
まずカメラ制御に必要な Script を作成した。これにより最大角度の設定やカメラ移動の速度などを制御する。これを MainCamera に紐付けして完成。
次に弾の発射部分の実装。Unity アセットストアから銃のアセットを入手し、銃口部分に弾を発射する Script を作成。これにより銃の発射機構の完成。
次に的の出現乱数を実装。Script で的の出現位置、出現間隔などの設定を行えるようにした。
最後に当たり判定の実装。Script を作成し、的のマテリアルに紐付け。
これで元となるシューティングゲームは完成。
そして的のマテリアルに Web カメラからリアルタイムで映像をテクスチャとして表示する Script を作成。インターホンに接続した Arduino と Unity を接続。
ここに Vuforia という AR プラグインを使って AR 化。

WebAR
まず通常の html での Web サイト作成と同様に html ファイルを作成。
コーディング
AR.js と A-Frame のリンクの紐づけをした後、A-Frame のタグを使用していく。
主なタグ
a-scene...シーン作成。AR.js との紐付けも行う。
a-assets...表示するオブジェクトファイルの指定。
a-marker...マーカー画像の指定。マーカーはジェネレータで作成可能。
a-entity...表示位置、サイズの指定。サイズはオブジェクトごとにかなり異なるので最初は指定しないことを推奨。
作成した html ファイル、3D モデルなどのオブジェクトファイル、マーカーのファイルなどを Github というオンラインホスティングサービスをしているサイトでレポジトリを作成してその中に保存。
最後に Netlify という Web サービスを利用して https で公開。

感想
榎本：AR の製作自体はネットの情報だけでは情報が古いところや自分の環境によるところがあったので上手くいかず、試行錯誤したところが一番時間がかかりました。
物無っぽい電子工作要素がインターホンしかないのと計画性のなさが今年の反省点です。
来年は計画的に動くことと、電子工作ももっとやっていきたいなと思います。
共同制作者の高梨さん、飽きたときのクソ絡みに付き合ってくれた方々、ありがとうございました。
高梨：unity についての理解が浅く、バグの修正に時間がかかりすぎたと思います。あと電子工作面を考えていなかったので製作がソフトウェアばかりになり後輩にあまり知識を渡せなかったことが一番の後悔です。1 年間付き合ってくれた榎本には感謝しかないです。



麻布のアソビ大全03

あざぶのあそびたいぜん

製作者　M3 広瀬晴喜

ストーリー

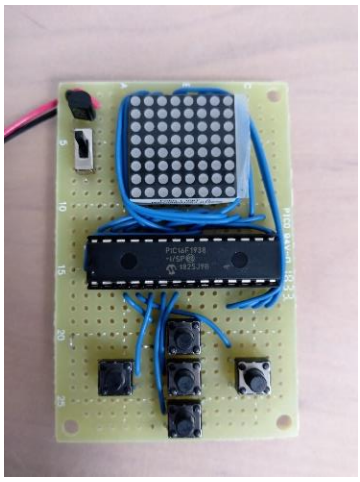
～アトO法律事務所～

質問来てた

Q,プログラムの原因不明のバグを放置して無理矢理プログラミングするのって犯罪ですか？

A,結論　犯罪になる場合とならない場合がある。

外観



部品説明

・スライドスイッチ…左上の細長い部品です。電源の ON/OFF を切り替えています。

・三端子レギュレータ…左上の黒い部品です。電池の電圧を下げる働きがあります。

・ドットマトリクス…上側の正方形の部品です。8*8 の LED で画面を表示します。

・PIC16F1938…中心の黒い部品です。ここでプログラムを動かしています。

・タクトスイッチ…下側の5つのボタンの部品です。ここから PIC に信号が入力されます。

概要

今年の売り物は3つのゲームが入っています。対応したボタンを押すとゲームが始まります。それぞれに難易度が 3 段階存在し、下のボタンを押すと難易度を示す顔が変わります。また、ゲーム終了後のエンディングは真ん中のボタンを長押ししてスキップできます

①弾避けゲーム…上のボタンを押すとスタート

上から降ってくる弾を上下左右に動いて避けます。一定時間避け続けるか、被弾するとゲームが終了します。

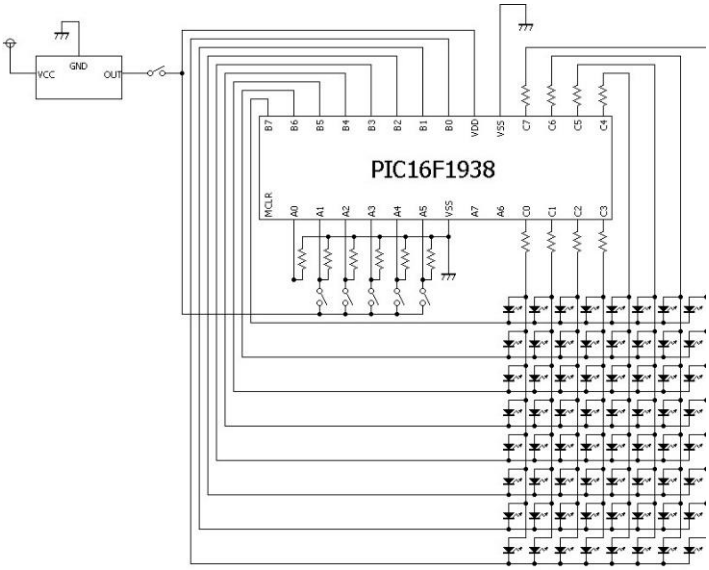
②宝探しゲーム…左のボタンを押すとスタート

隠された宝を画面左側のレーダーを頼りに探します。光っている場所が少ないほど宝に近いです。宝の真上に移動し真ん中のボタンを押すとクリアです。

③球拾いゲーム…右のボタンを押すとスタート

上から降ってくる弾を左右に動いて拾います。一定数拾うか落とすとゲームが終了します。

回路図



感想

この売り物の制作を通してはんだ付けの技術やプログラミングの知識をより一層深めることができました。来年度からの文化祭に向けての制作に生かしていきたいです。

勧誘から逃げろ！

制作者：M2　天野春翔

回路図設計者：H1　加藤さん

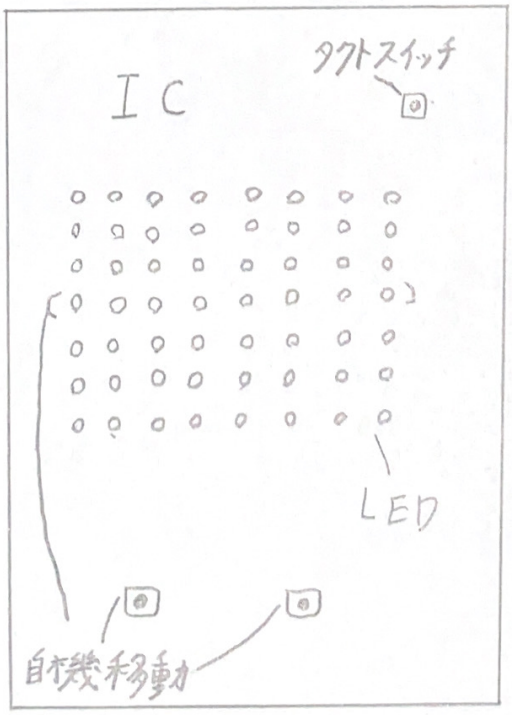
・ストーリー

友達が他の部活の勧誘にやってきた！
それを避けて早く制作を終わらせよう！

・ルール

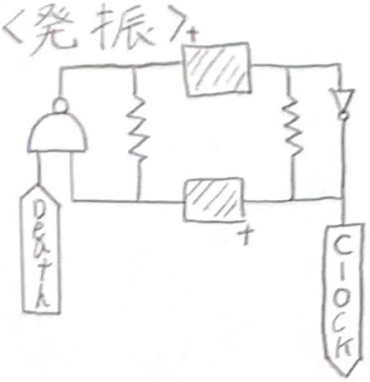
8×7のLEDの上下から赤色の球が流れてきます。
自機を左右に動かしてそれらを避けてください。

・外観

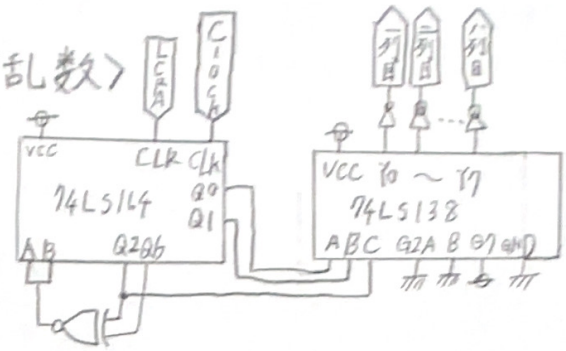


・回路図

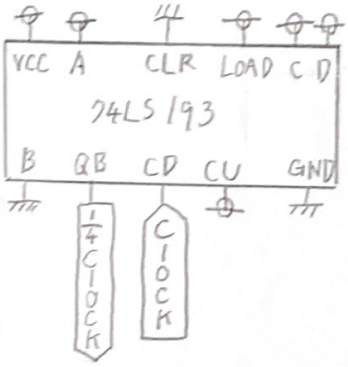
<発振>



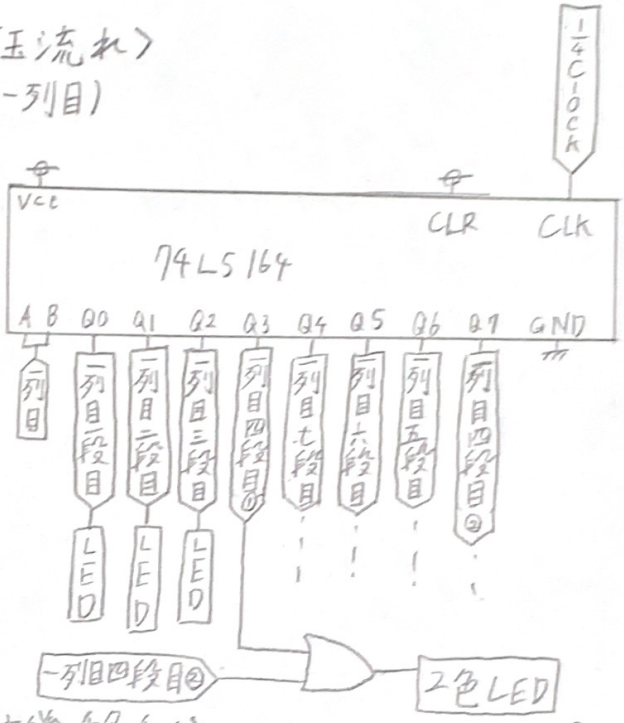
<乱数>



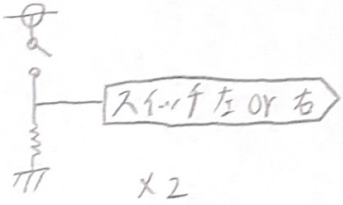
<遅い発振>



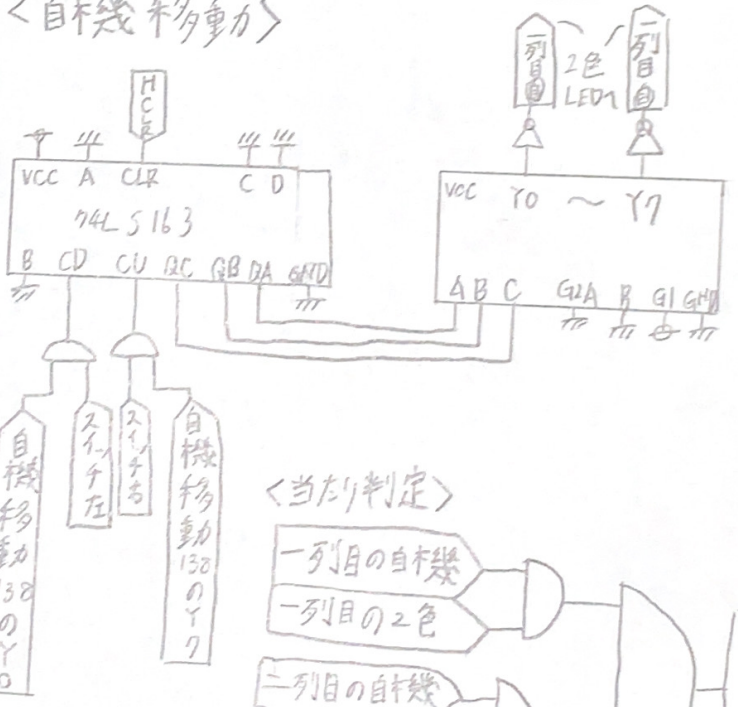
<玉流れ>
(一列目)



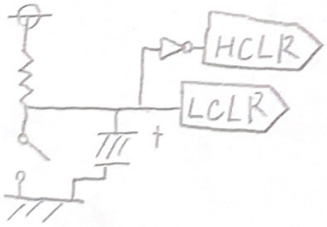
<スイッチ>
(自機)



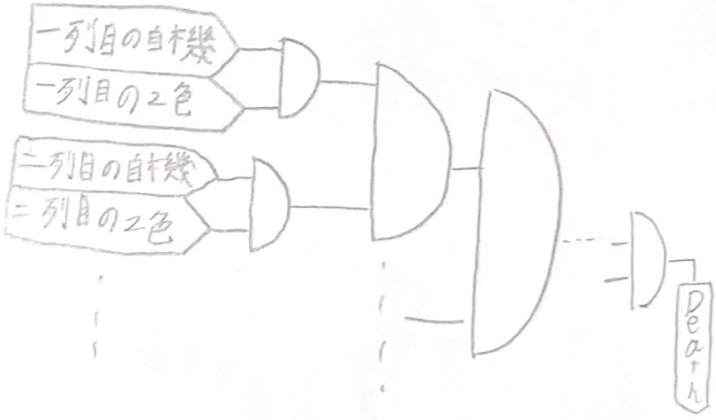
<自機移動>



<クリアスイッチ>



<当たり判定>



・感想

以前と比べくしてはんだ付けの技術が向上し、また電子回路への理解を深めることができて良かったです。

定期試験

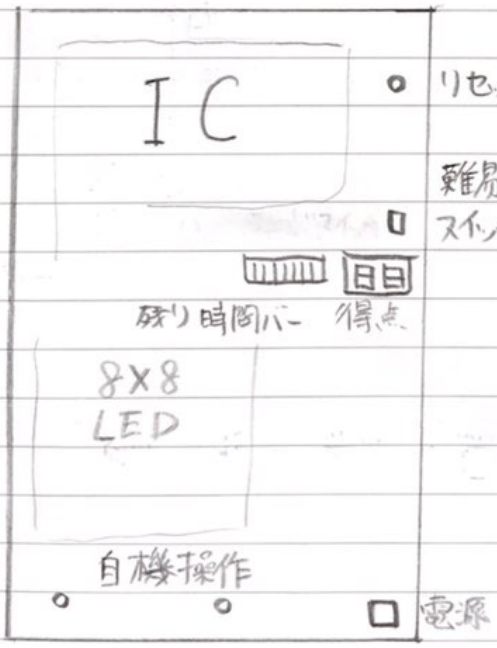
制作
M2 石村勇樹
回路図設計
H1 畠山さん

ストーリー

今学期もこの季節がやって来た。たくさん点を取ろう。
ただし赤点は取ってはいけない。

概要

8x8のLED表示盤の上から青と赤の玉がそれぞれ流れてくる。
最下段の緑の玉を左右のタクトスイッチで操作する。
青玉に当たると+1点、赤玉に当たると−1点。
中央右の残り時間バーが0になるまたは得点が0を下回るとゲームが終了し、その時点までの得点を競うスコアアタック。
右にあるスライドスイッチを上にすると、一度でも赤玉に当たると終了する鬼畜モードが楽しめる。

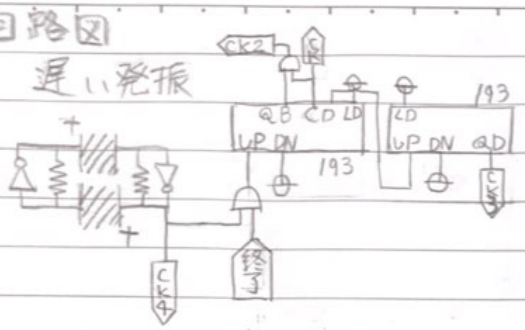


使用

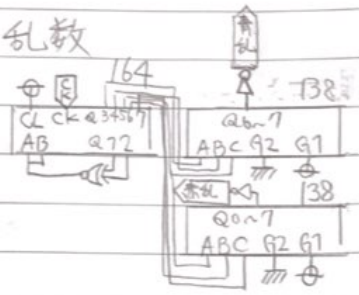
LS・HC IC/ICソケット	各約60個
タクトスイッチ	3個
7セグ2桁	1個
LEDアレイ	1個
2色LED	56個
3色LED	8個
抵抗	165個
コンデンサ	5個

回路図

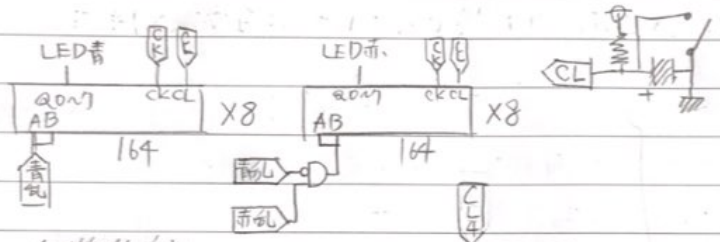
遅い発振



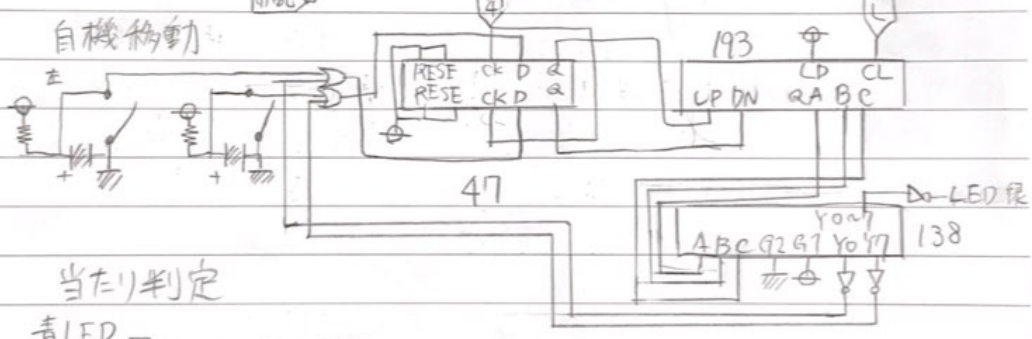
乱数



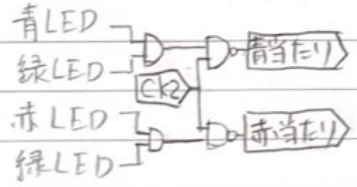
玉流れ(青)　玉流れ(赤)　クリア



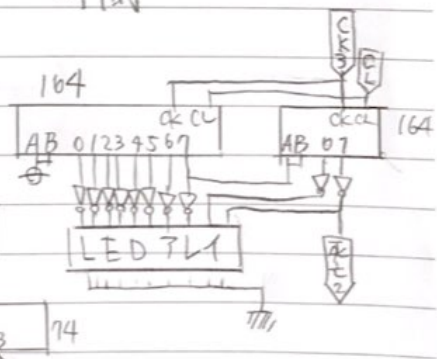
自機移動



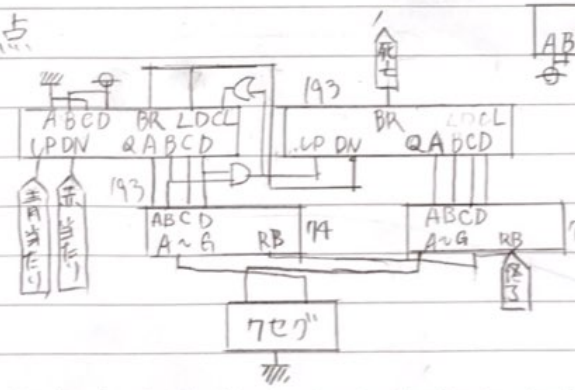
当たり判定



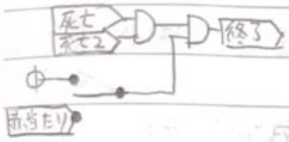
時間



得点



終了



感想

活動期間中、ほぼ滞りなく制作を進めることができ、無事に完動しました。
この制作を通して、論理回路やLSICについての知識が深まりました。

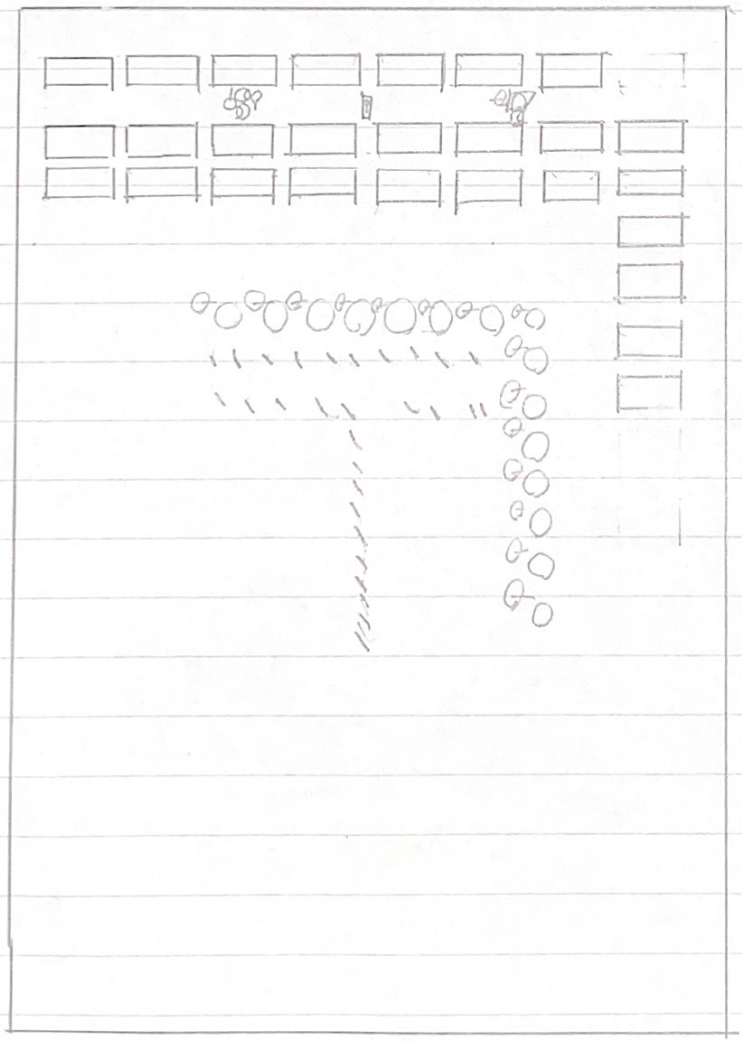
弾よけゲーム(中ーゲー)

中二 大和田
協力者さん 磯野さん

・ストーリー

時は令和、5Gの殺人光線が発び交う時代に平成のものですらない謎の機械が現れた。
果たして者はいつ終わるとも分からないデスゲームをクリアし、生き残ることができるか!(スコア制です)

・概要



・使用部品

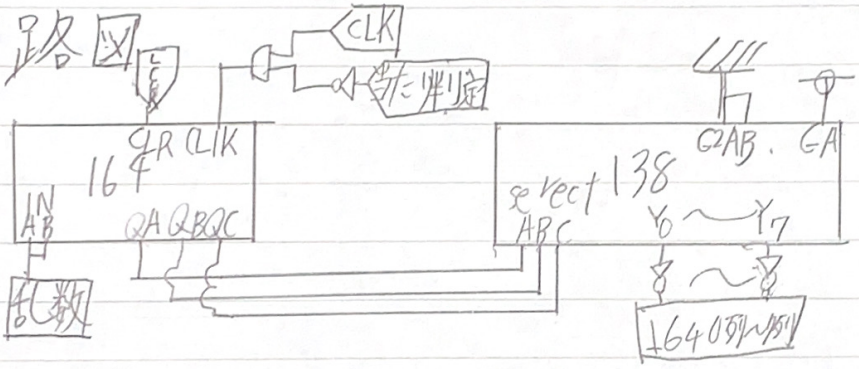
74LS IC ---概要の四角いやつで基本的な動きを作っている。

抵抗 ---これが電気の流れをそがいする。

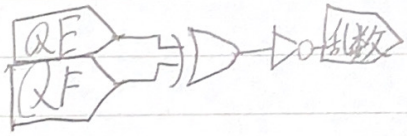
コンデンサー ---一時的に電気を流したりとめたりできる。

他LEDと配線材とスイッチなどもある。

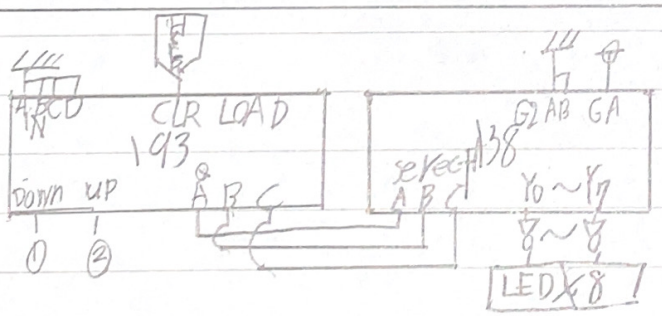
・回路図

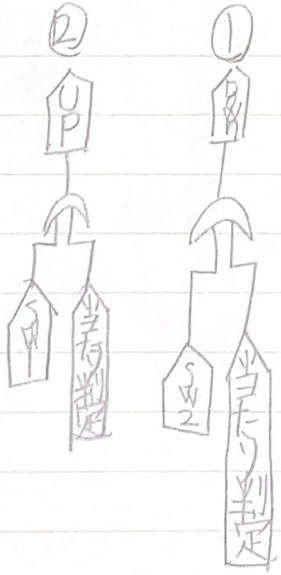


164の



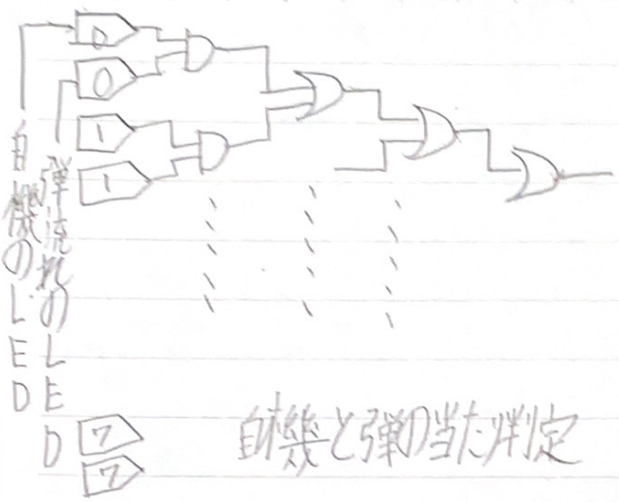
弾流れ

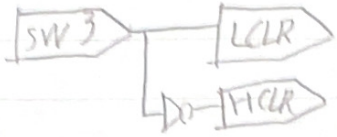




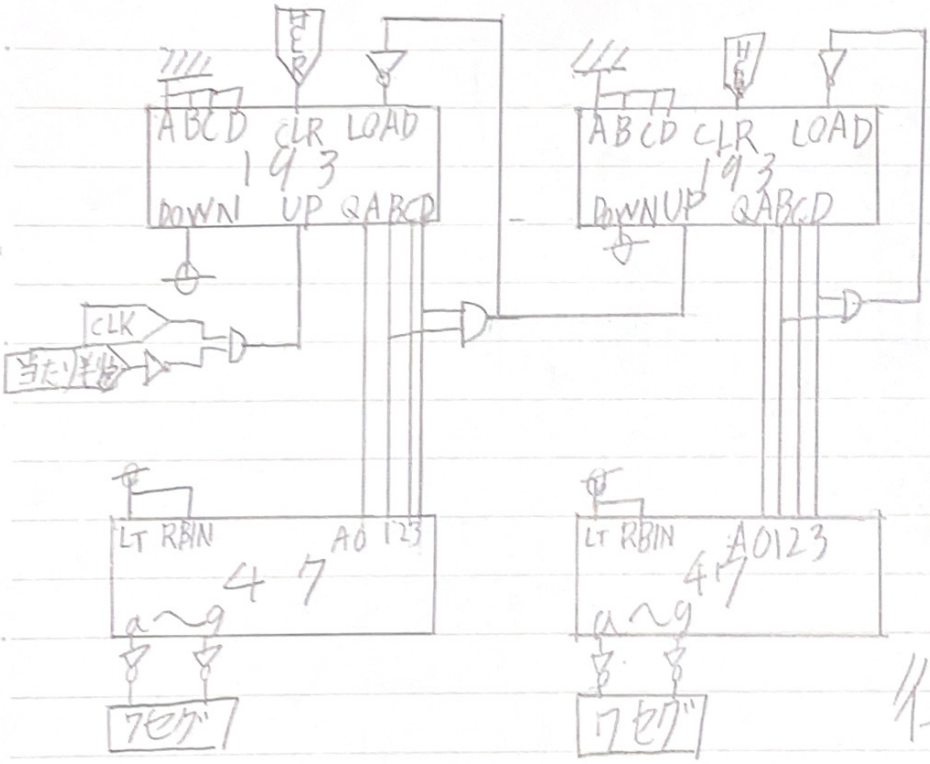
※SWは押すとロー

当たり判定は当たるとハイ





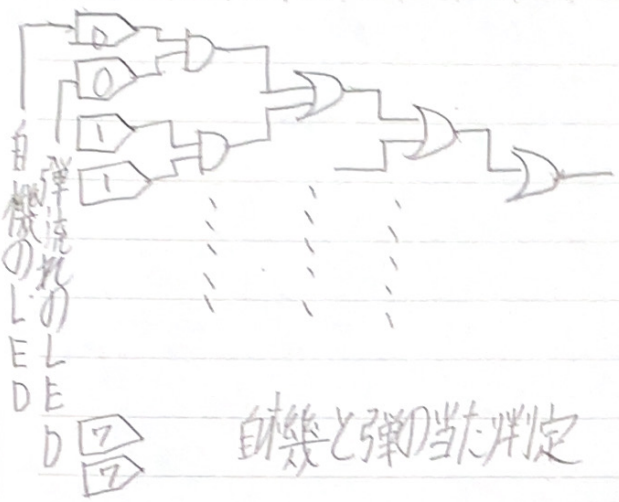
クリアスイッチ

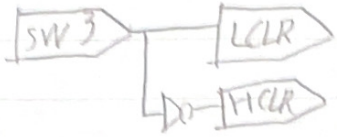


得点表示(秒数)

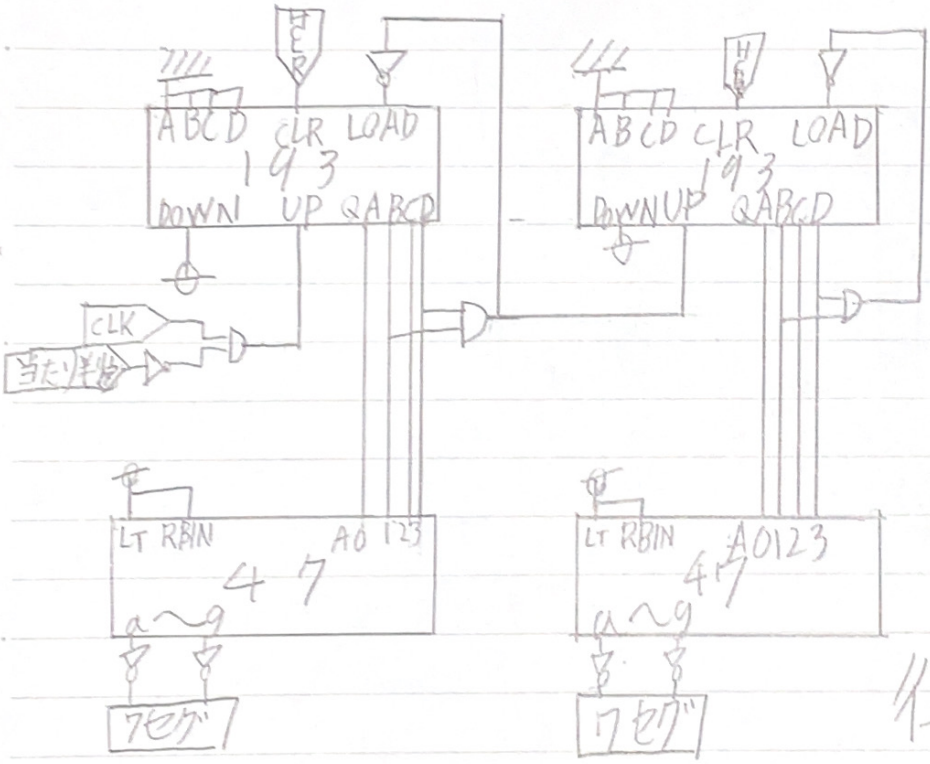
感想

あらていど製作にも慣れることができ、少しずつ原理が分かるようにもなってきました。LED1つ1つに配線をするのが大変だったです。今後もがんばって完動させます。





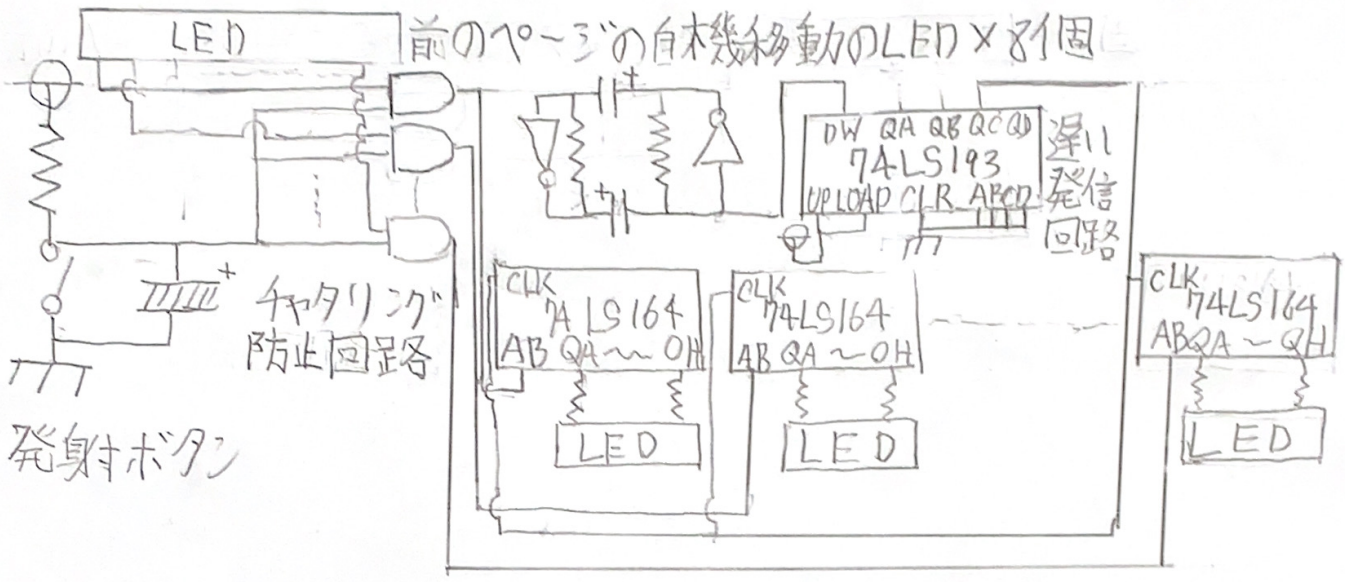
クリアスイッチ



得点表示(秒数)

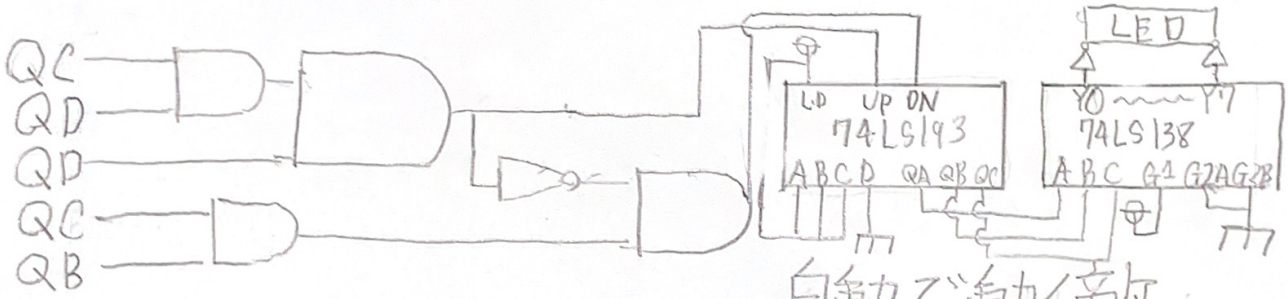
感想

あらていど製作にも慣れることができ、少しずつ原理が分かるようにもなってきました。LED1つ1つに配線をするのが大変だったです。今後もがんばって完動させます。



玉流れ8個

上の図の遅い発信回路の所からQA、QB、QC、QDからクロック取って来る

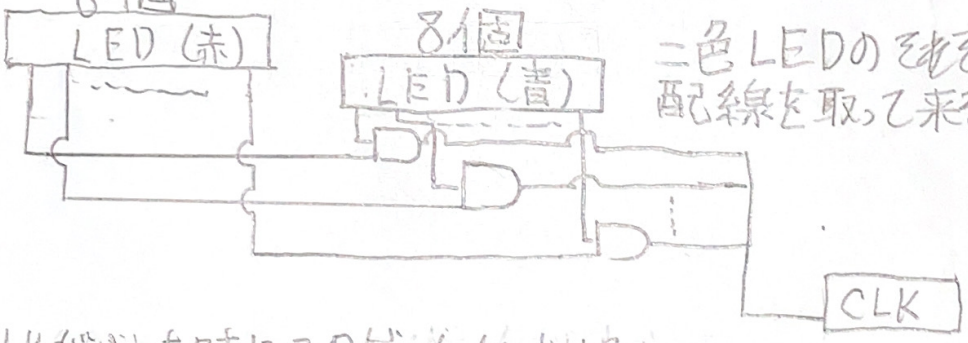


自動で動く高次

8個
LED (赤)

8個
LED (青)

二色LEDのそれぞれの色の配線を取って来る。



CLKが入った時にこのゲームが止まる。

感想

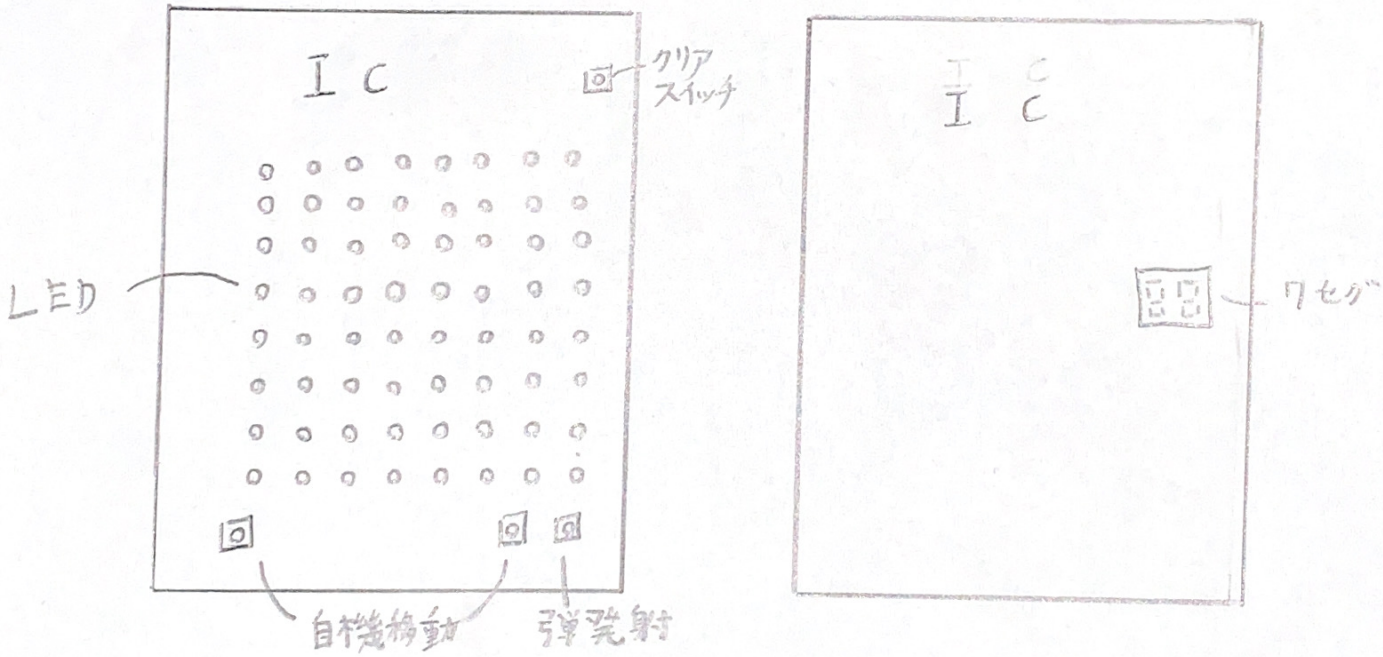
ハンダ付けの時、基盤の裏の銀色の円盤をはがしてしまってICソケットの位置を変えなければいけなくなりそのハンダ付けがとても面倒だった。ハンダ付けは一回で成功させる事がとても大切だということが分かった。また、これを書いている途中は、まだ完動していないので完動させられるか不安です。今は、前まで動いていた自機移動が動かなくなり何が原因かも分からないので本当に困っている。どうかこれを読む時は完動しますように!

ココナッツ・シューティング

制作:M2 加藤 理人
回路図設計:H1 磯野さん

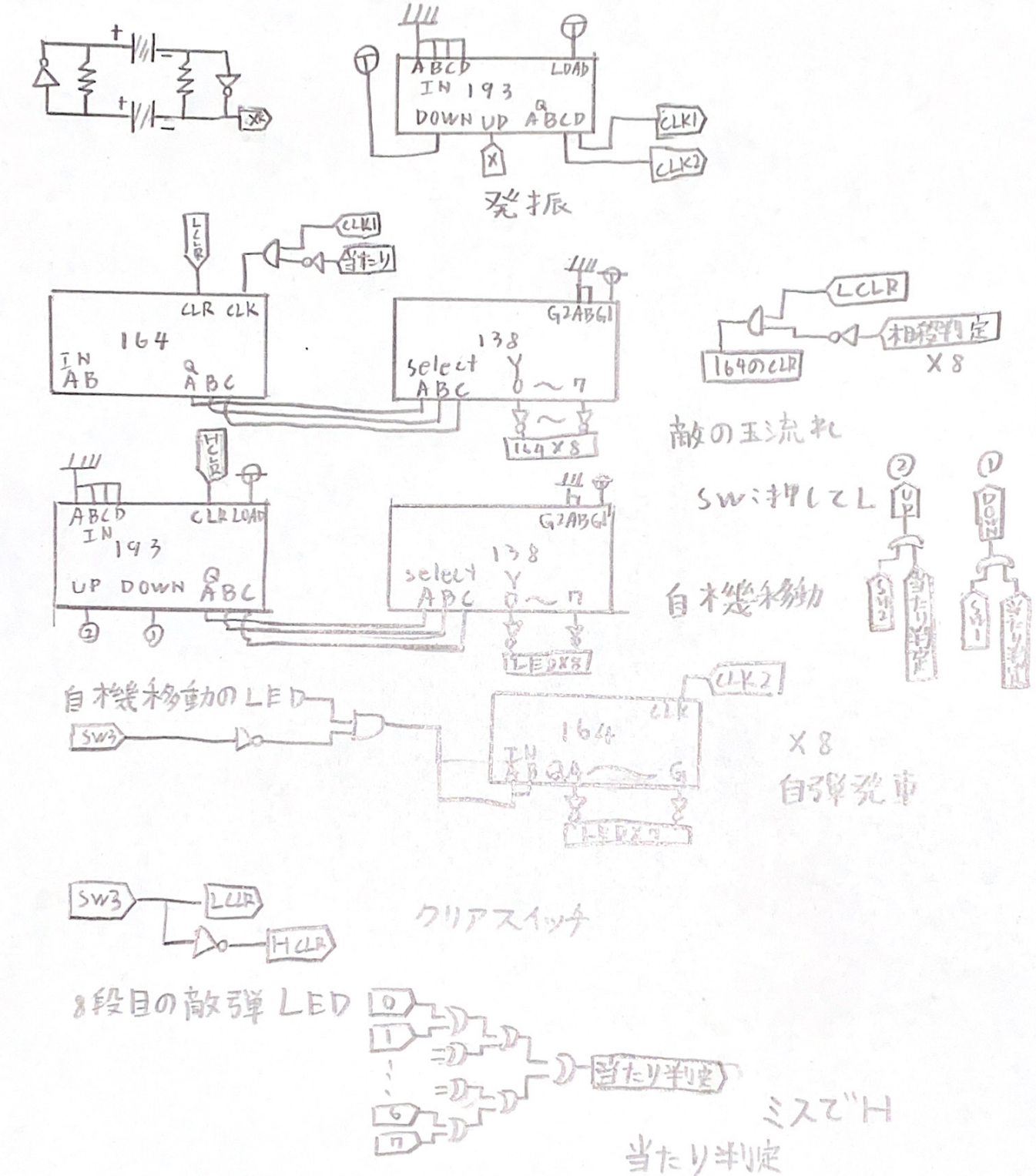
ストーリー
ある日、散歩をしていると、不幸にもココナッツが落ちてきてしまいました。ココナッツは人を年間150人も死亡させており、大変危険です。ココナッツを撃ち落とし、身の安全を確保しましょう。

概要



使用した部品
- 74LS IC
- 1KΩ抵抗
- コンデンサ
- 2色LED
- タクトスイッチ
- 7セグ

回路図


ABCD
LOAD
IN 193
DOWN UP
Q ABCD
CLK1
CLK2
X
発振
LCLR
CLK1
当たり
CLR CLK
164
IN AB
Q ABC
G2ABG1
138
select ABC
Y 0～7
164×8
LCLR
164のCLR
相殺判定
×8
敵の玉流れ
HCLR
ABCD
CLR LOAD
IN 193
UP DOWN
Q ABC
②
①
G2ABG1
138
select ABC
Y 0～7
LED×8
SW1が押してL
②
UP
①
DOWN
SW2
当たり判定
SW1
当たり判定
自機移動
自機移動のLED
SW3
CLK2
CLK
164
IN AB
QA～G
LED×8
×8
自弾発車
SW3
LCLR
HCLR
クリアスイッチ
8段目の敵弾LED
0
1
6
7
当たり判定
ミスでH
当たり判定

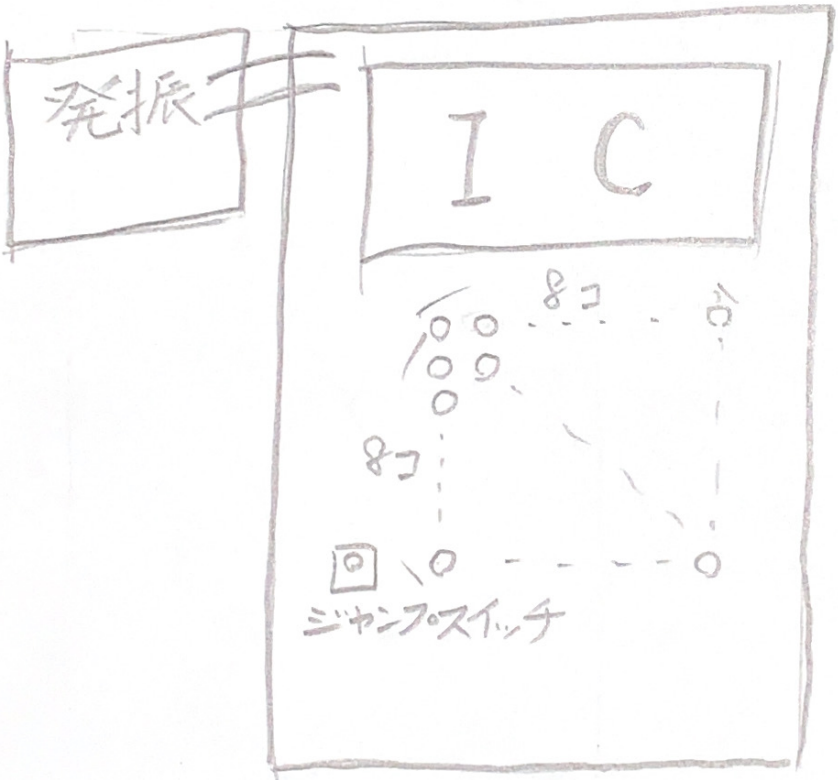
スーパーマリオタト伝

製作：M2 角野
回路図設計：H1 磯野さん
協力：物無の皆様

STORY

Nintendo Switchでマリオを遊んでいたら
突然バグってしまった！クリボーもキノコも
出てこないこんなゲームがってられるか！
いや…でも意外にシンプルで面白いかも…!?
よし，このバグったゲームを極めてやる！

ルール・概要



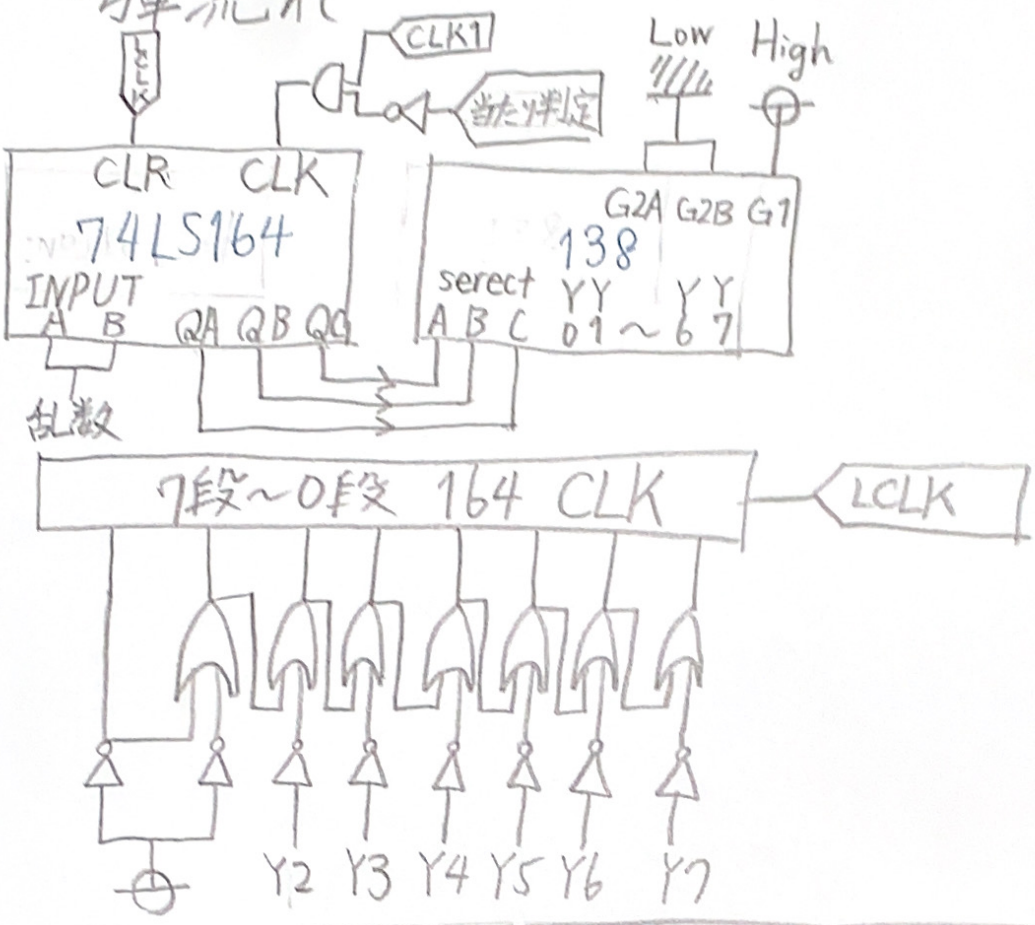
右側から壁が流れて来ます。ジャンプボタンを押すと自機がジャンプするので，ぶつからないよう上手くよけつづけて下さい。また，壁の上には乗れるようになっています。

使用した部品

- 74LSIC
- スイッチ
- LED（単色・2色）
- 抵抗
- コンデンサー
- 基盤

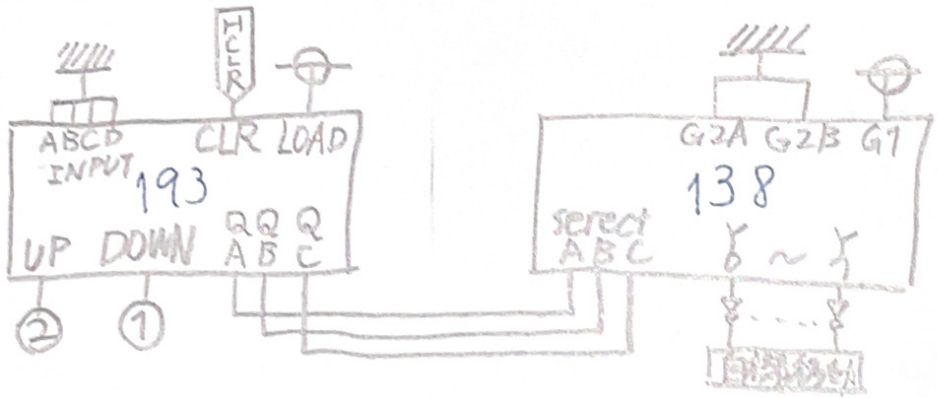
回路図

・弾流れ

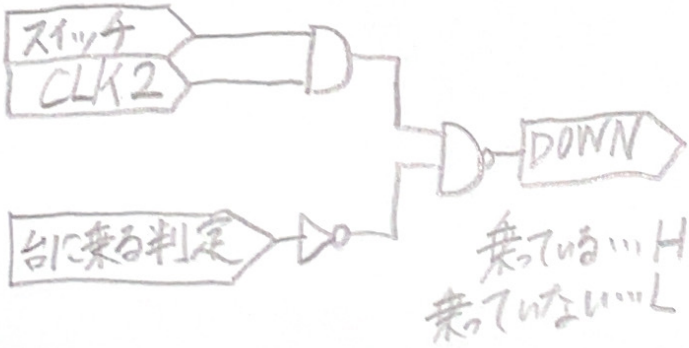


・自機移動

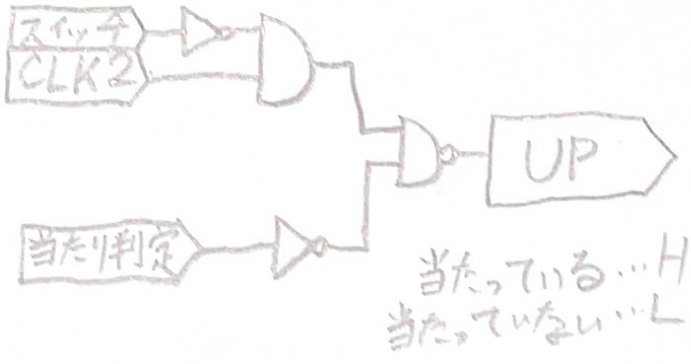
※スイッチは押したらLが出ます。



①

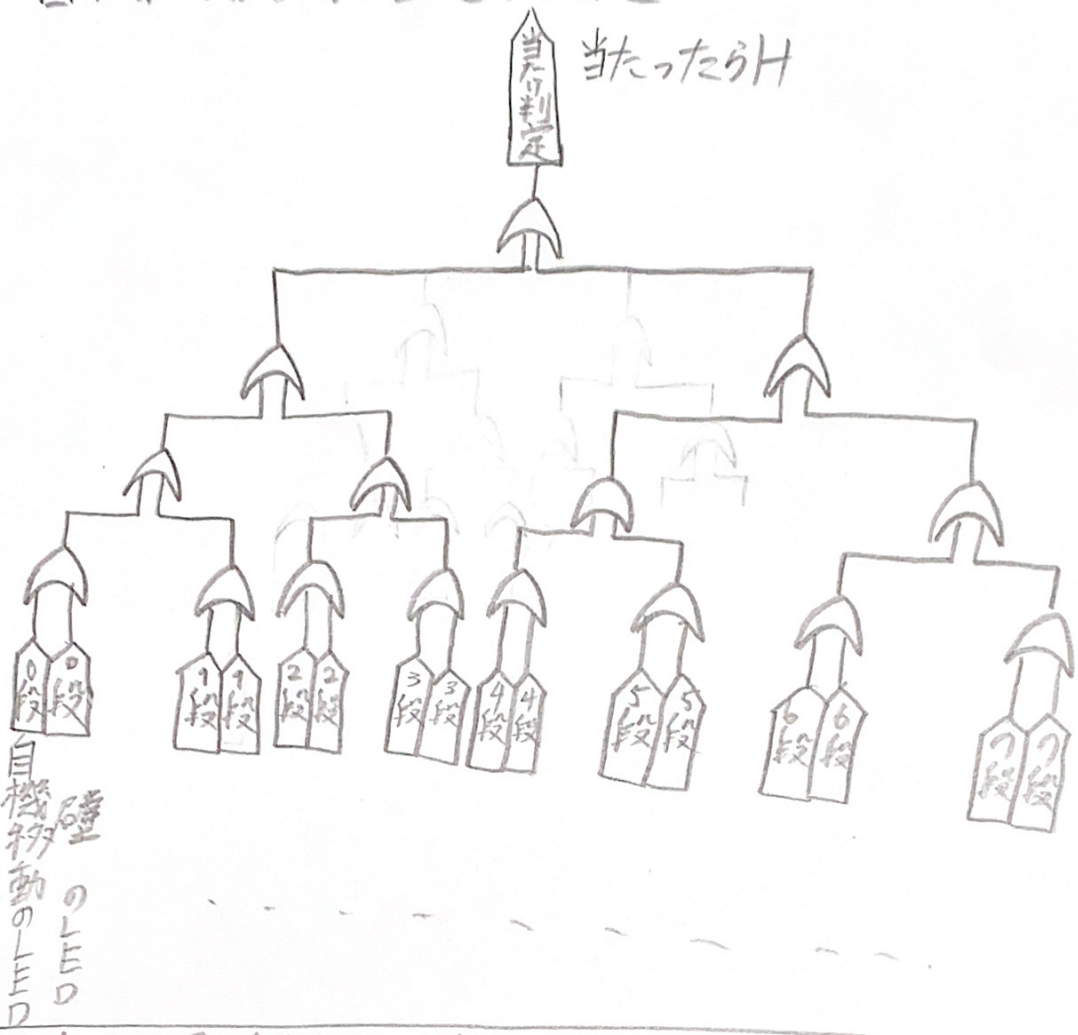


②



回路図（続き）

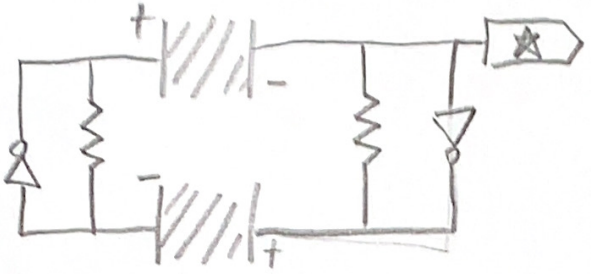
・自機と敵機の当たり判定

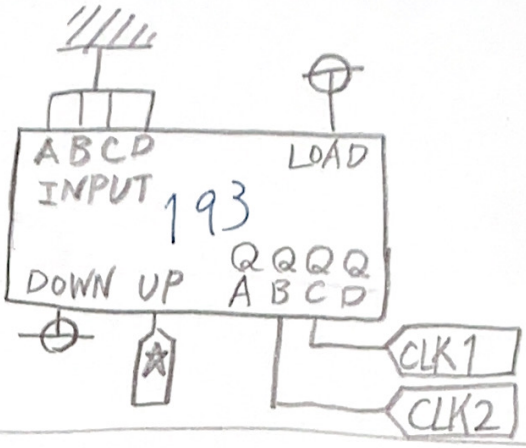


・台にのる，もしくは一番下に自機がいる判定

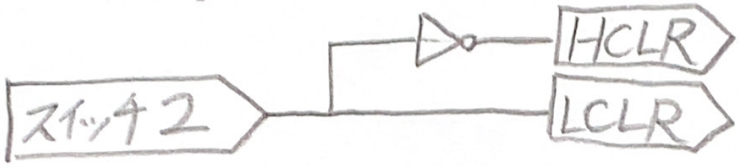
回路図(続き)

・発振

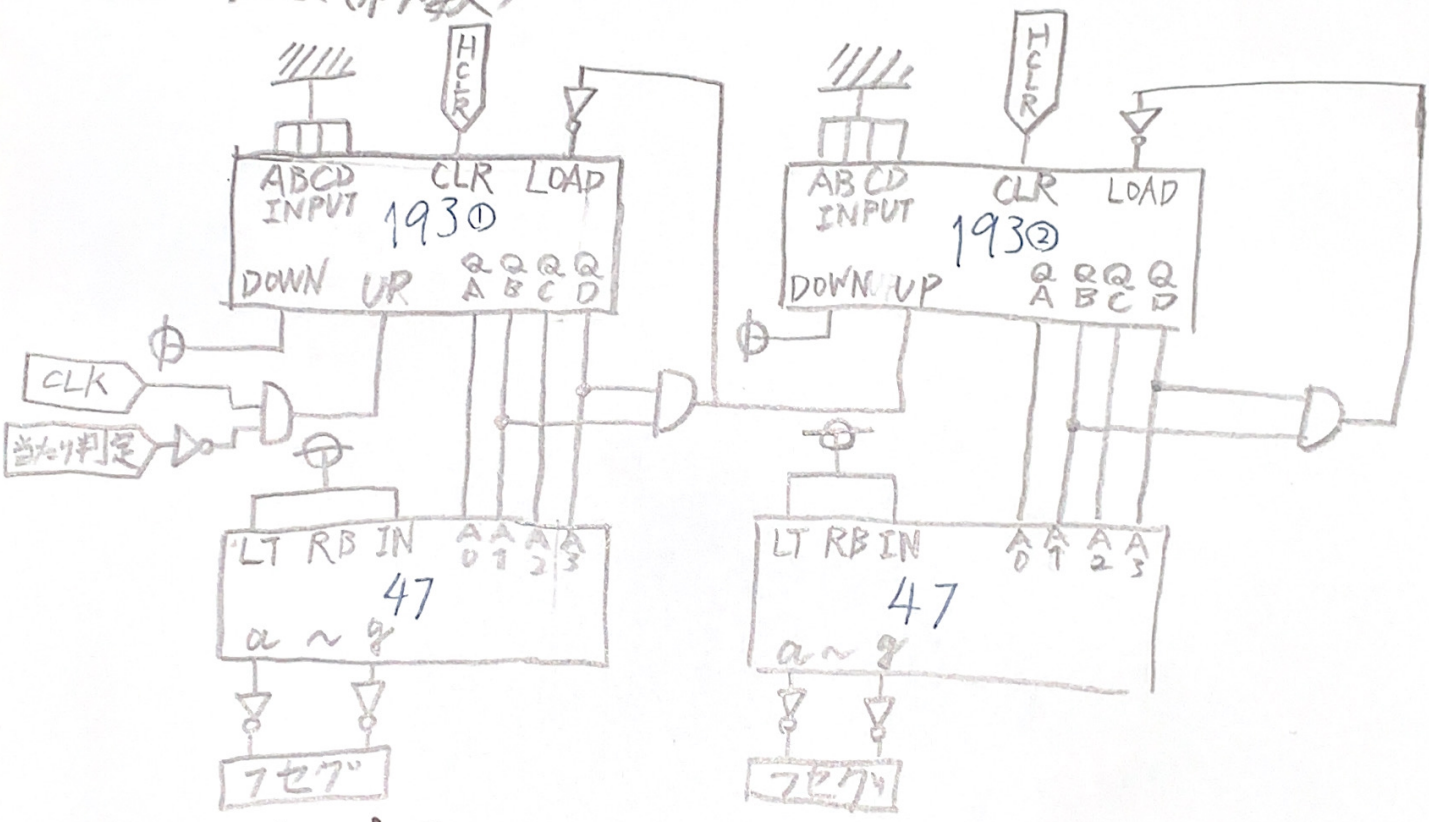




・クリアスイッチ



・得点表示(秒数)



感想

どこがどの役割をしているかが
分かったときが一番気持ち良かったです。
配線の技術が上がったと思います。

過激派ヴィーガンとの死闘

製作者:M2小平

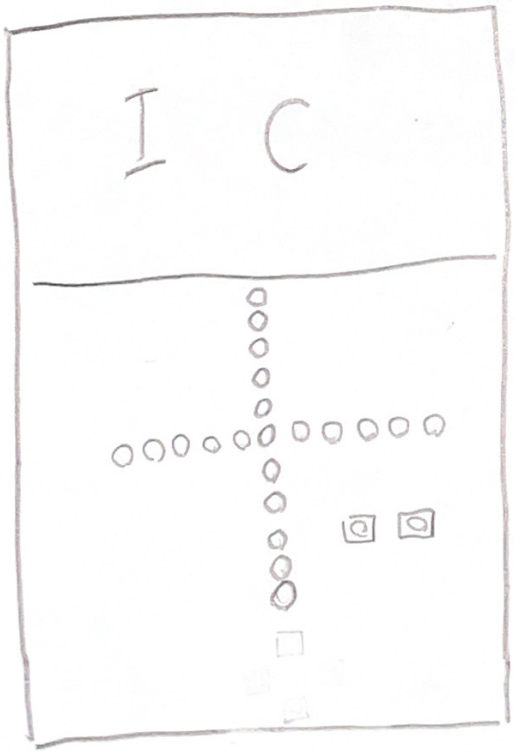
回路図設計者:H1高梨さん

ストーリー

やめて!過激派ヴィーガンのクソリプでトマトの写真を投げつけられたら、トマトが苦手な製作者の精神が燃え尽きちゃう!
お願い、死なないで製作者!あんたが今ここで倒れちゃ、他の物理部との交友関係、製作物の完成はどうなっちゃうの?ライフはまだ残ってる。ここを耐えれば、過激派ヴィーガンを論破できるんだから!
次回、「製作者死亡」。デュエルスタンバイ!

ルール:上下左右から弓単が飛んで来るので盾で受け取めましょう。

外観



使用した部品

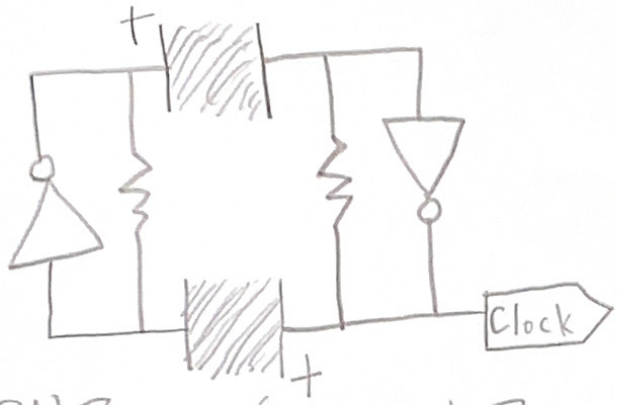
74LS IC
LED

感想

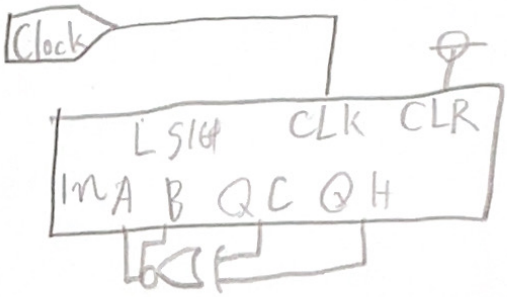
配線などの基本的な技術が向上した。

回路図

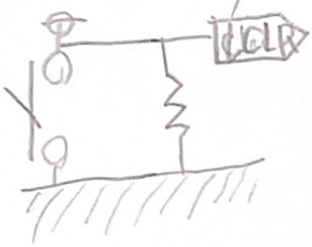
発振



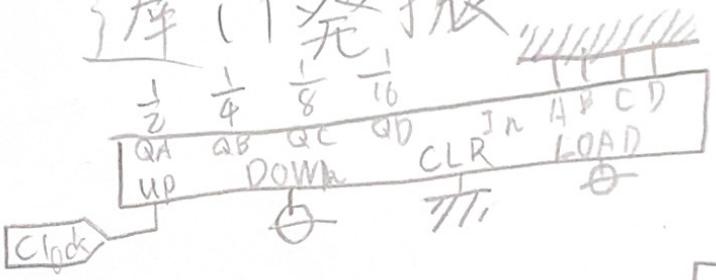
乱数



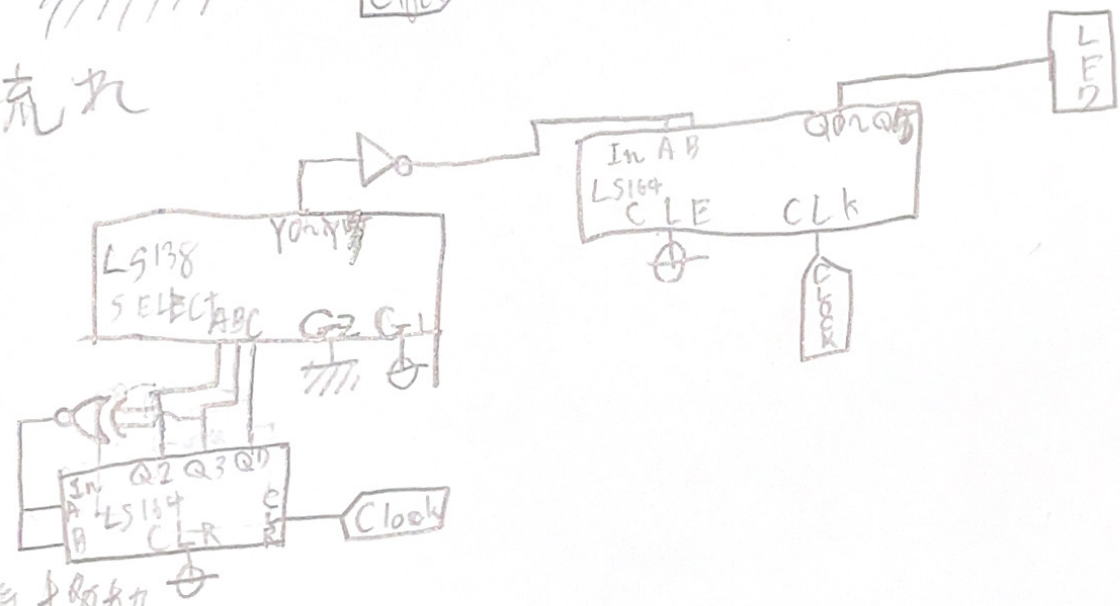
クリアスイッチ



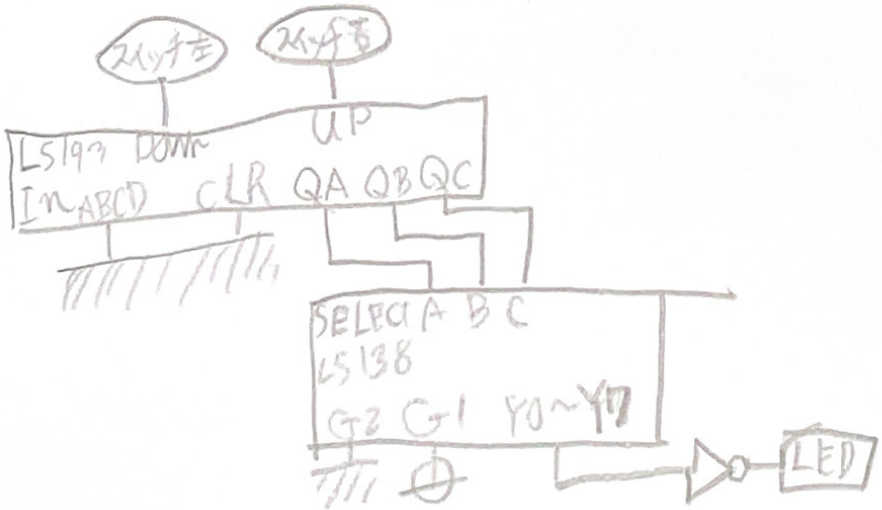
遅い発振



玉流れ



自機移動



リンゴをキャッチ

制作者 M2 小林弘明
回路図設計者 加藤さん

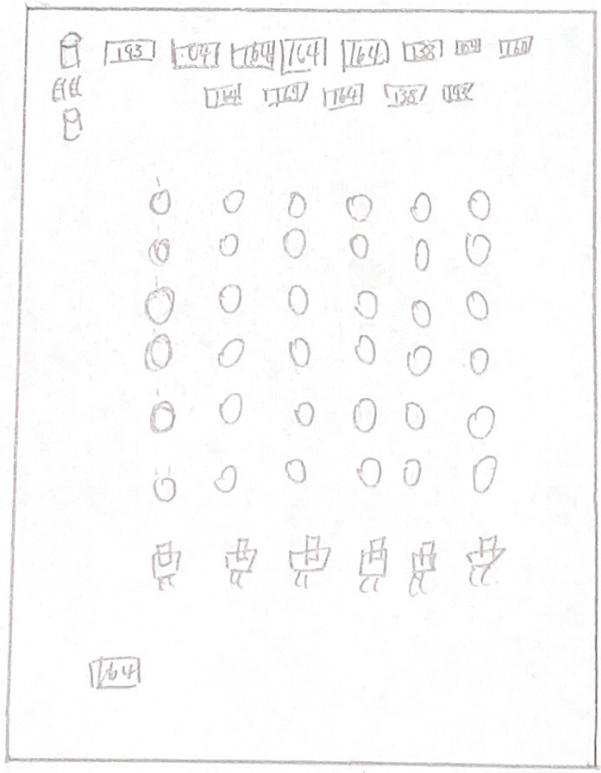
ストーリー

リンゴの木をゆさぶったら、リンゴがいっぱいおちてきた、一つのこらず、キャッチしよう、

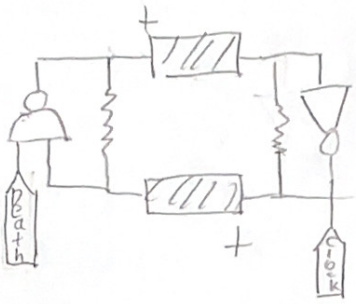
ルール

流れてくる玉をとるゲームです。

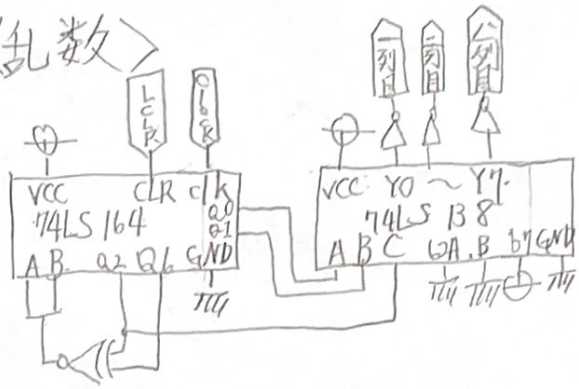
外見



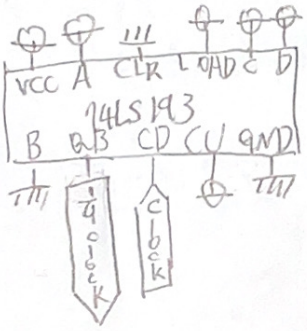
〈発振〉



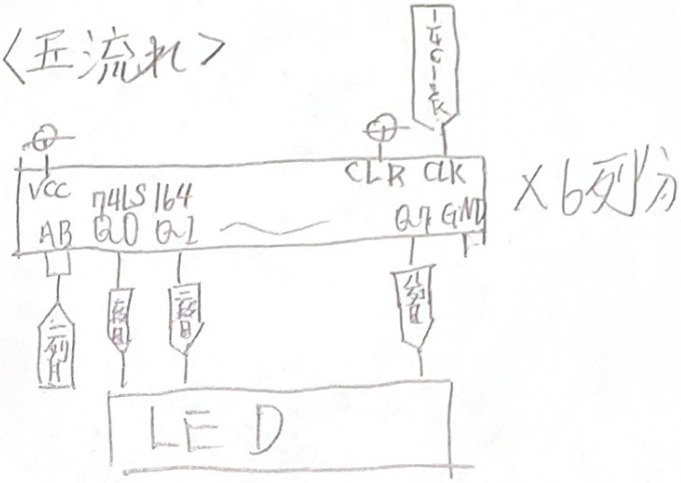
〈乱数〉



〈遅い発振〉

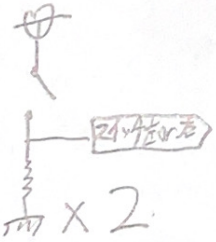


〈五流れ〉

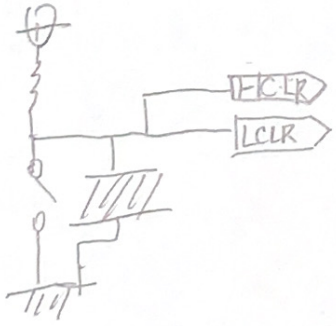


〈スイッチ〉

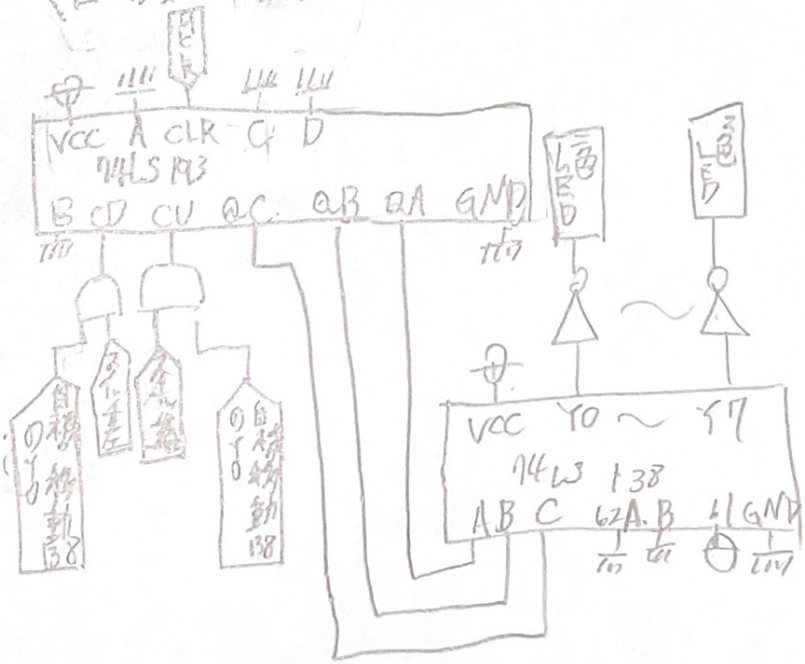
(自機)



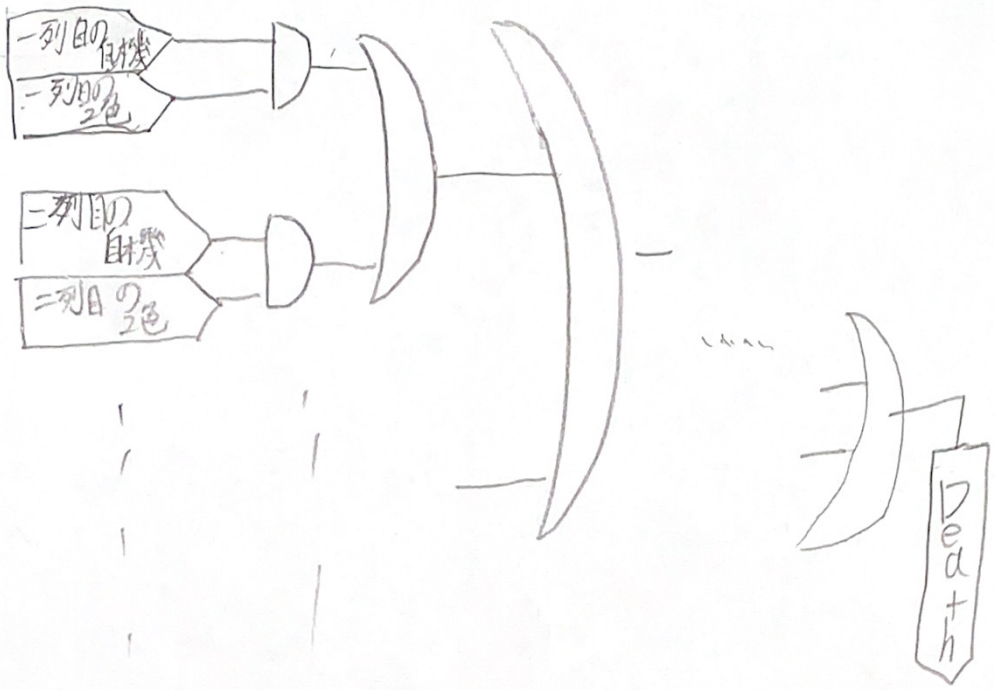
〈クリアスイッチ〉



〈自機移動〉



〈当たり判定〉



感想

先輩からアドバイスをもらいながら、
ずめ線をつかって見た目をよくしたり、いろんなことがありました。
LEDをつけるのをとてもくろうしました。

石を避けて生き延びろ!!

ストーリー

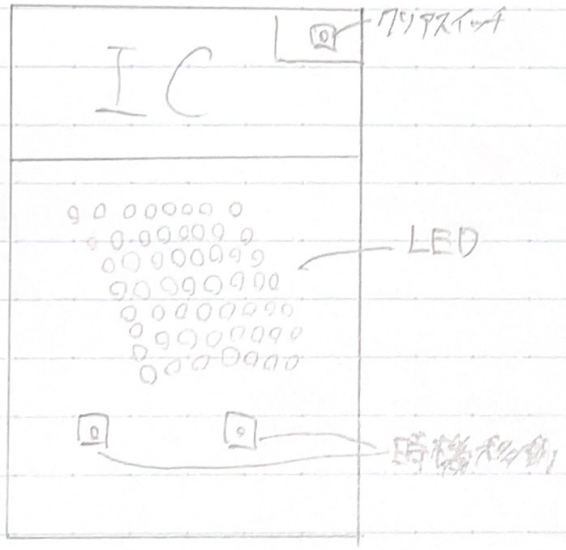
ある日 登山をしていたら山の頂上から
「日頃の恨みだ」と同輩から石を投げられた
石を避けて生きのびよう

制作

M2 坂東
回路図設計
H1 山田さん

使用した部品

- 74 LS IC
- LED (赤色, 2色)
- コンデンサ
- タクトスイッチ
- 1KΩ抵抗

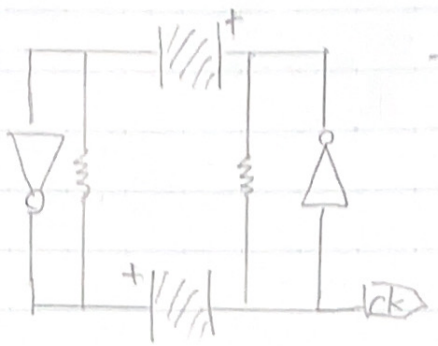


ルール

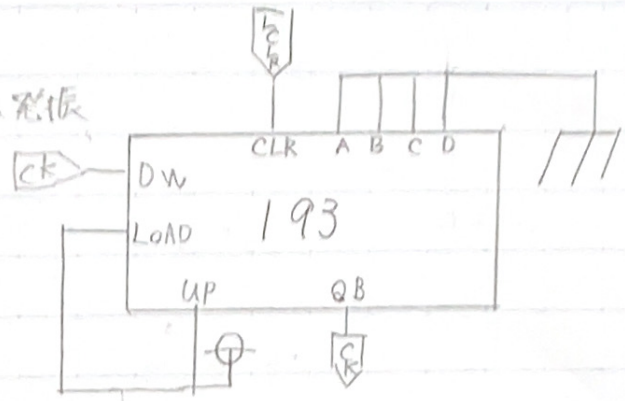
上からななめに降ってくるLEDの弾
を最下段1段を横移動して
避けていくゲームです。

回路図

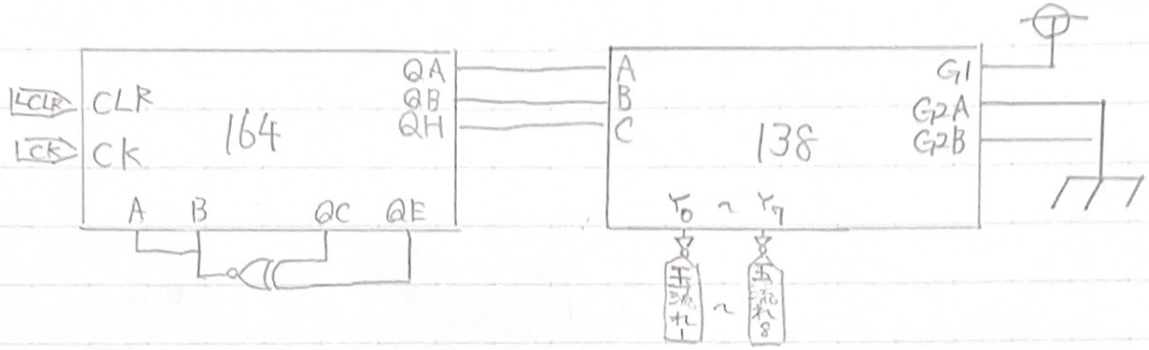
発振



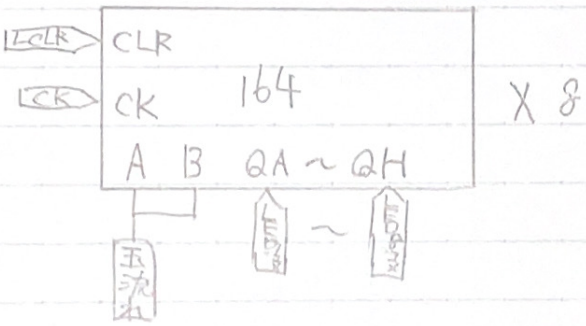
遅い発振



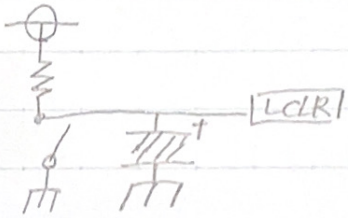
乱数



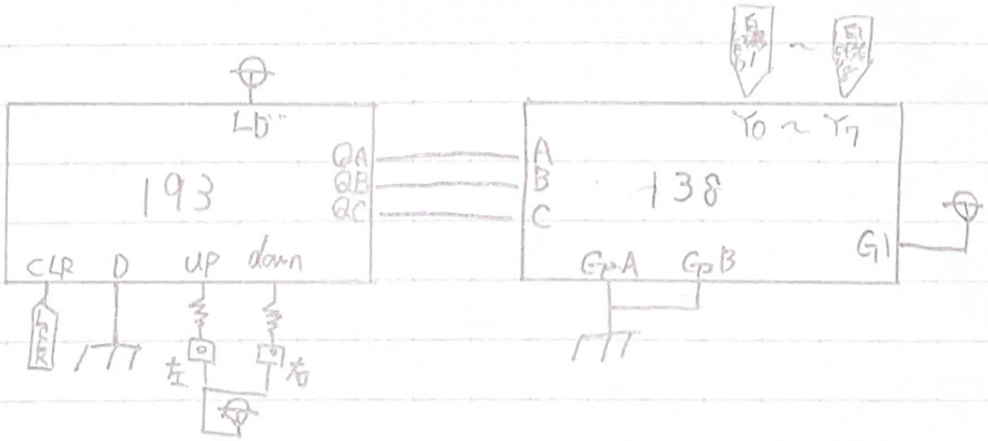
玉流れ



クリアスイッチ

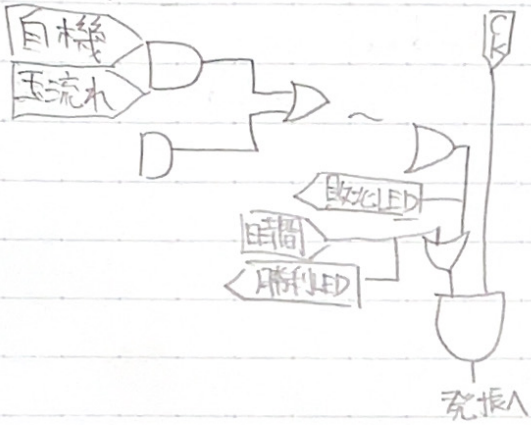


自機移動

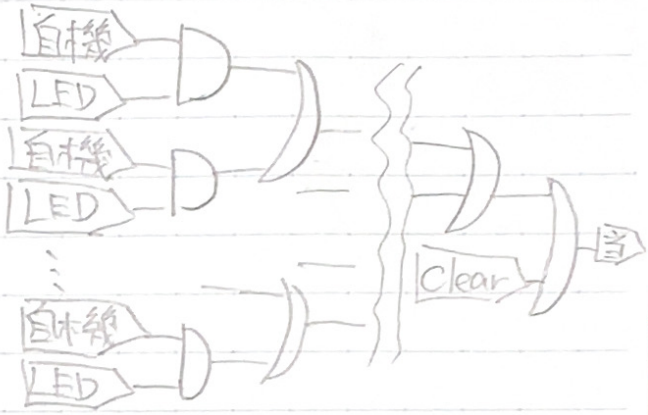


回路図

クリア判定



当たり判定



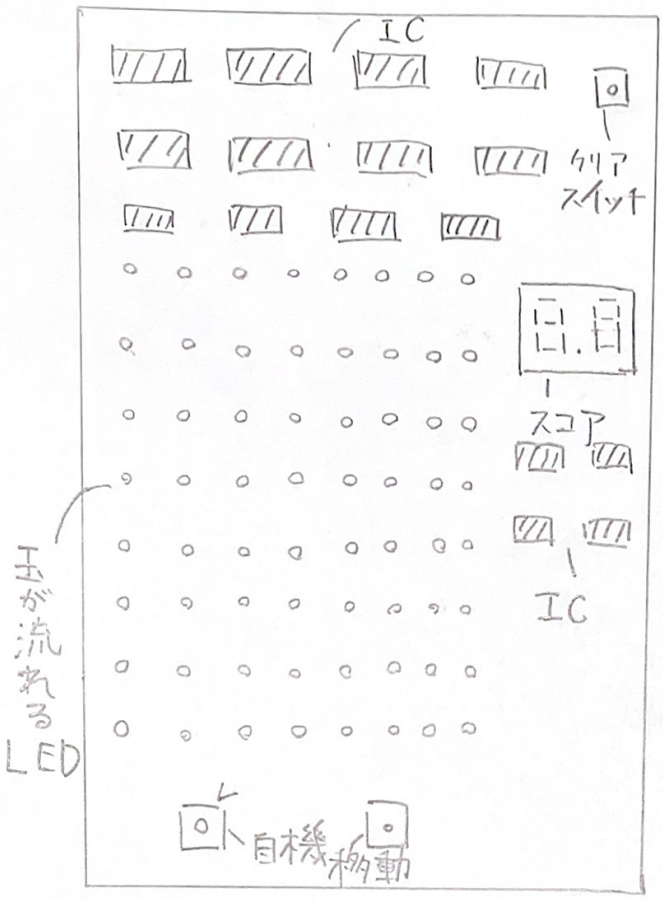
感想

はんだ付けが上部した時より上手にできるようになり、
回路について知ることができて良かったと思いました。

運をつかめ！

制作 M2 三牧
回路図設計 H1 高梨さん

①外観



②ストーリー

毎日のように不運にみまわれている
某物理部員M。
そんな彼の元に赤い幸運の玉が
降ってきた。運をつかみ
自らの体を幸運な体へと
変貌させろ！

③ルール

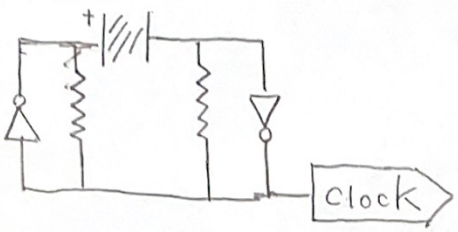
上から流れてくる赤い玉を
自操作の青LEDを使って、
キャッチしていくゲーム。

④使用した部品

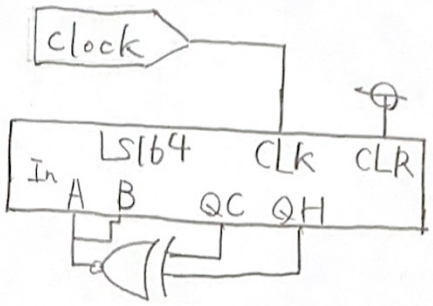
74LSIC　LED(赤、2色)
7セグ　タクトスイッチ
1kΩ抵抗　コンデンサ

⑤ 回路図

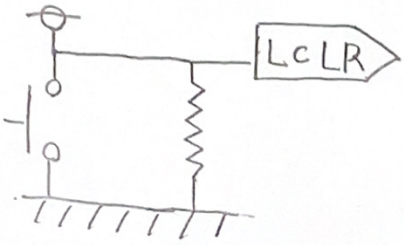
発振



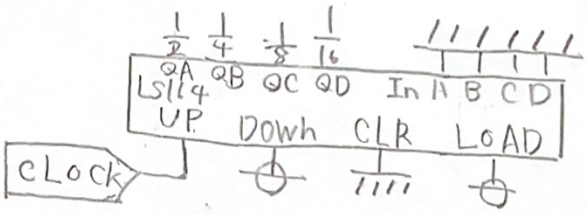
乱数



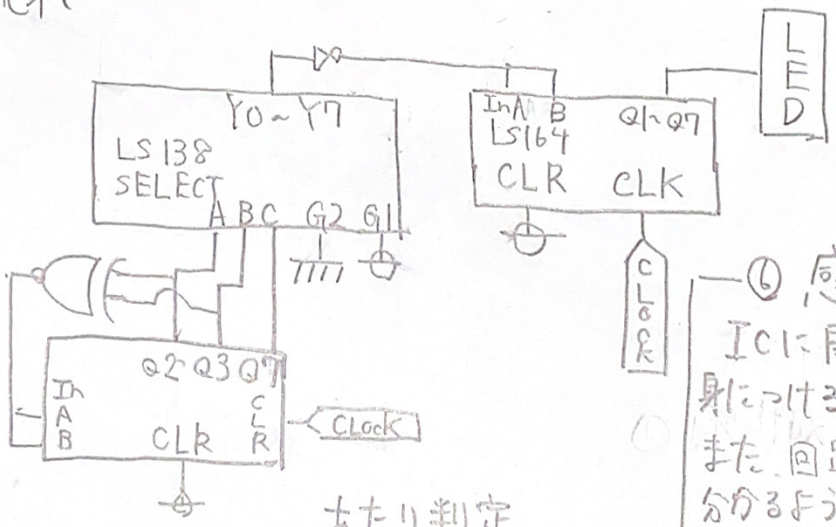
クリアスイッチ



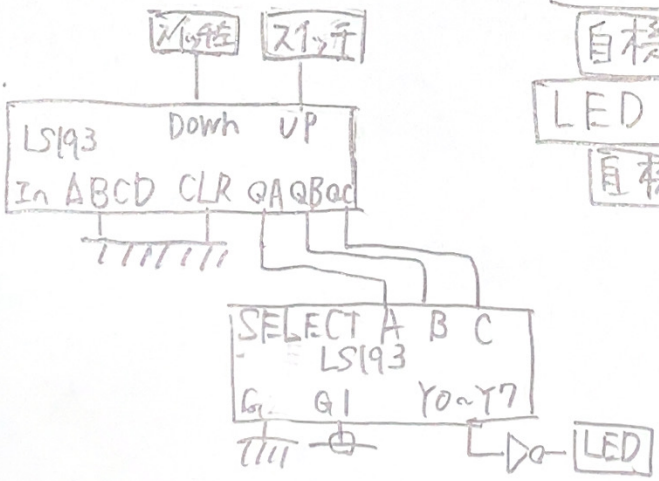
遅い発振



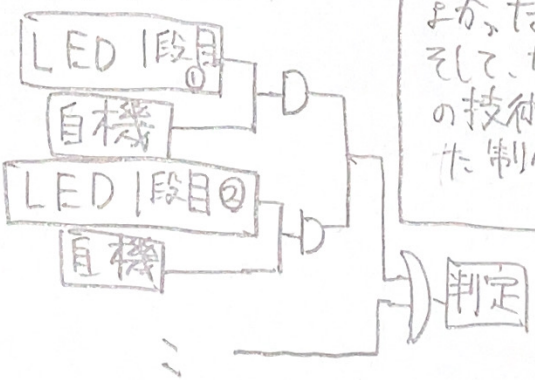
玉流れ



自機移動



あたり判定



⑥ 感想

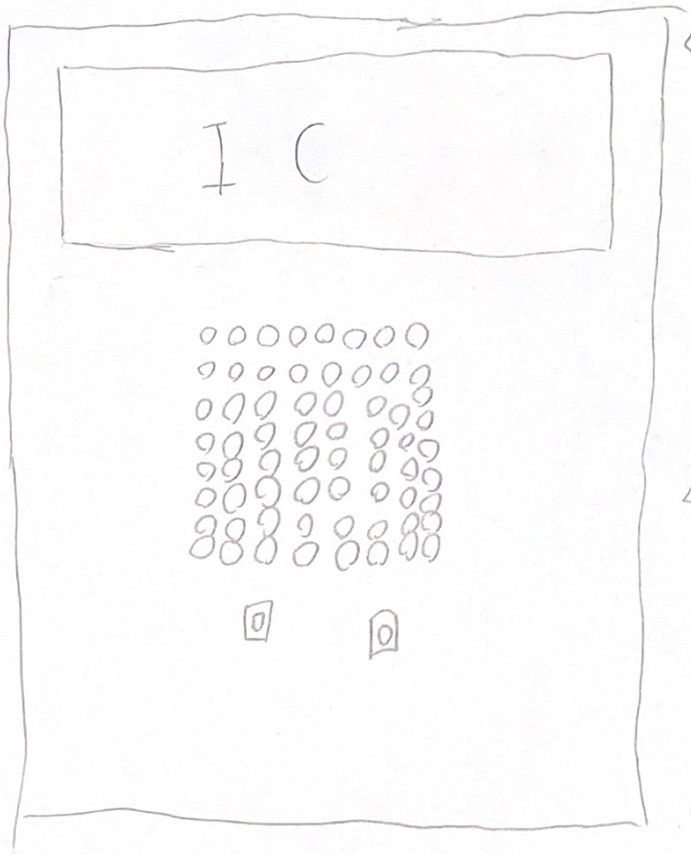
ICに関する知識を身につけることができた。また、回路図もある程度分かるようになってきたのでよかった。
そして、なによりはんだ付けの技術が上がったと感じた制作だった。

親方！空から女の子達が!!

製作：M2山田
回路図設計：H1山田さん
協力：物無の皆様

〈ストーリー〉空から緑色に光った女の子達が無数に降り注ぐ！
頑張って多くの命を助けよう！

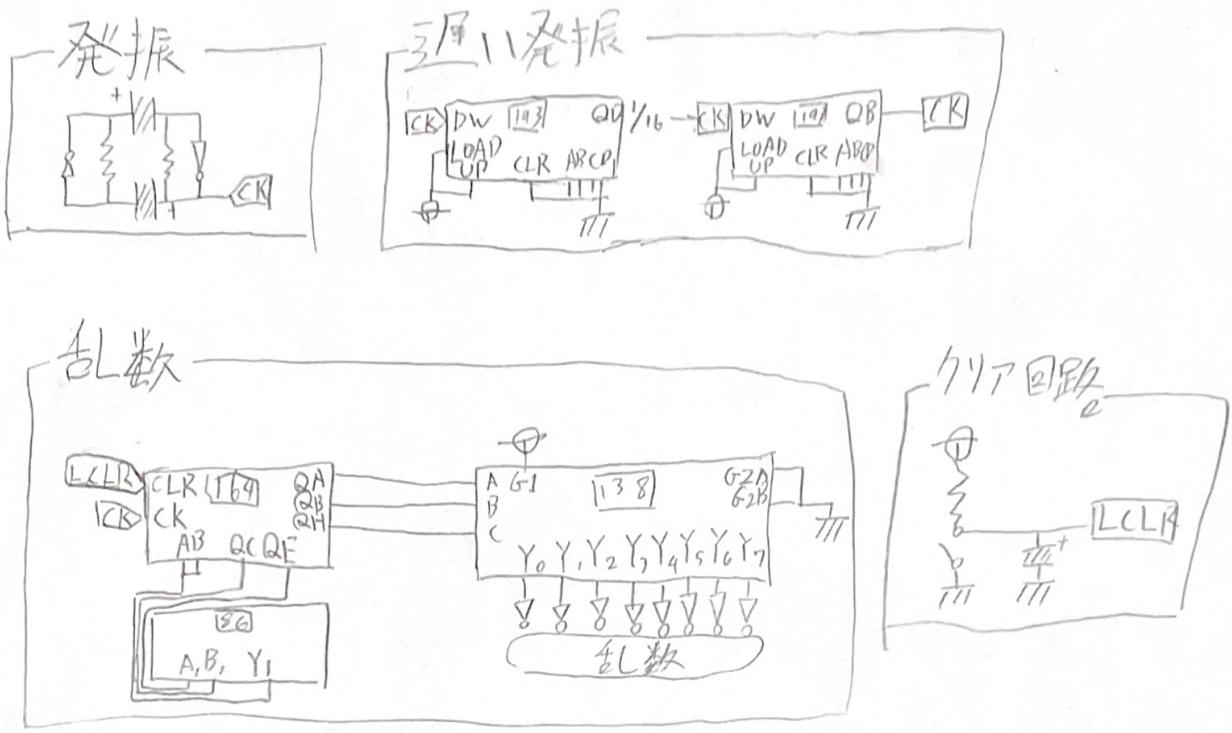
〈外観〉



〈内容〉
上から流れてくる
玉を自機で拾い、
スコアを加算していく
ゲームです。

〈使用した部品〉
74ls IC
タクトスイッチ
LED

〈回路図〉



〈感想〉

一つのミスから多くのバグが生まれ、解決までに時間をたくさん使ってしまった。この反省をいかし、今後は極力ミスをしないよう注意し、デバッグをすばやく終わらせられるようにしたい。

Date　　No.

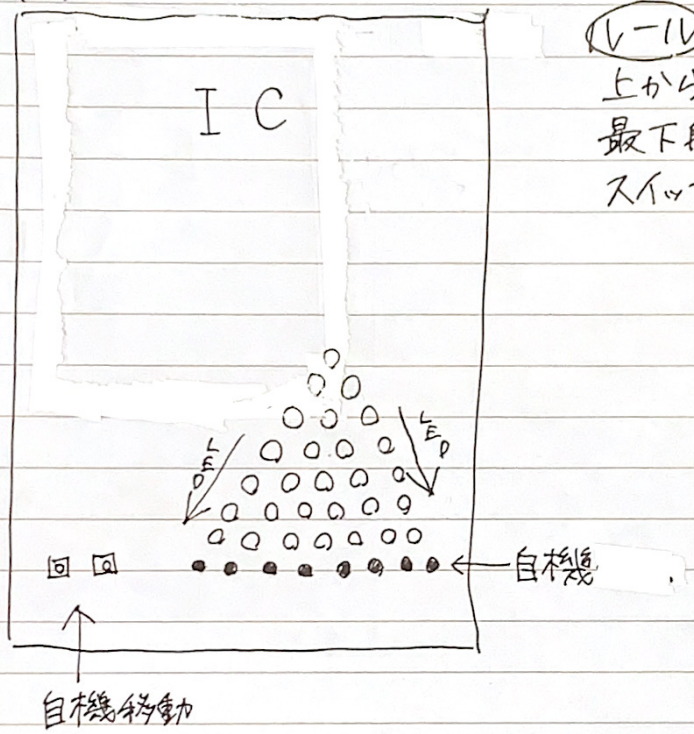
かき氷を食べたい

制作 M2 鹿島
回路図設計者 H1 谷中さん

1. ストーリー

富士山がおいしいかき氷に変わったらしい!! しかし実際に行ってみると他の物無量たちがかき氷を食べていて、さらに彼らはかき氷を独り占めするためにカケラを投げつけてきた! 転がってくるかき氷から逃げよう.

2. 概要



(ルール)

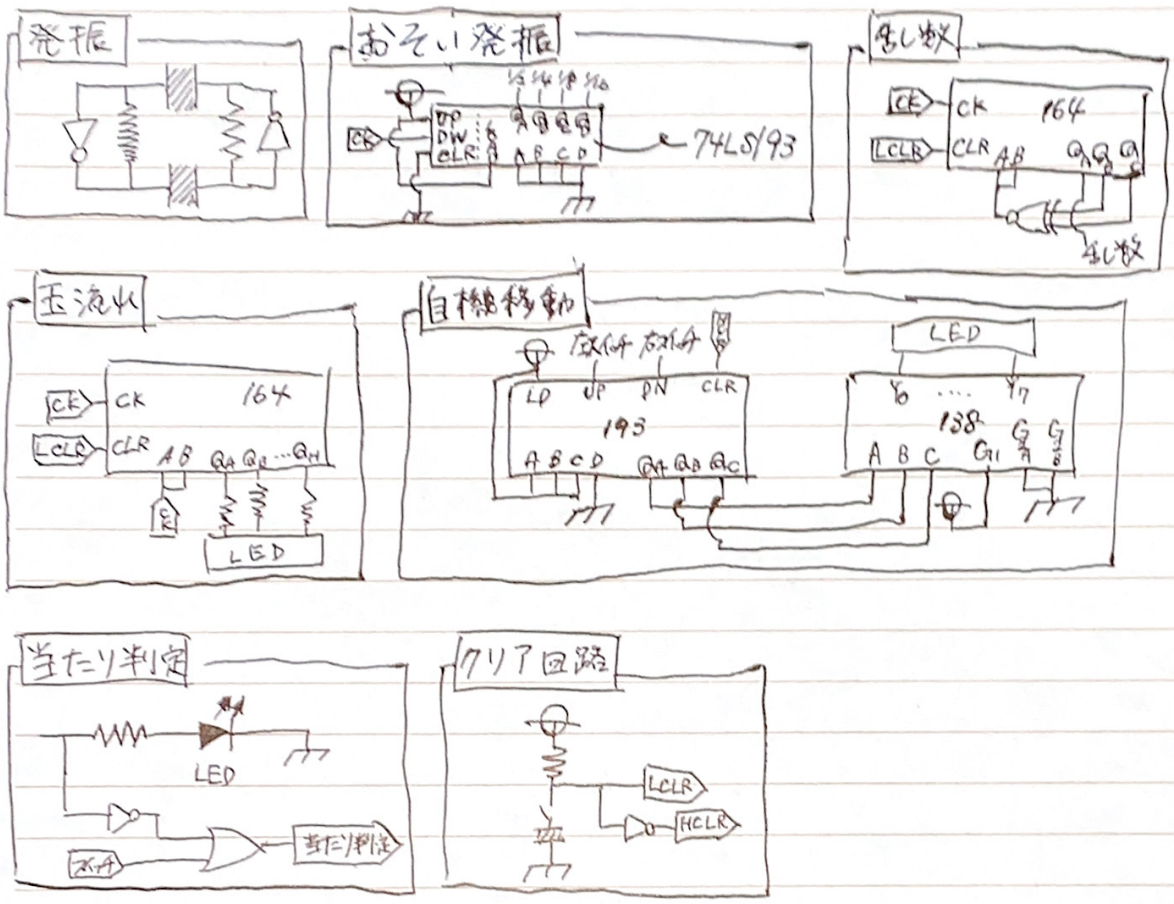
上から失印の2方向に降ってくるLEDを最下段でよけ続けるゲームです。2つのスイッチで左右に移動します。

3. 使用した部品

74LSIC
コンデンサ
1kΩ抵抗
LED(赤, 2色)

Date　　No.

4、回路図



5、感想

使用する配線材の種を少なくし、またていねいに素早く製作できるようになった。
はんだ付けも少し上手くなったため、嬉しかった。

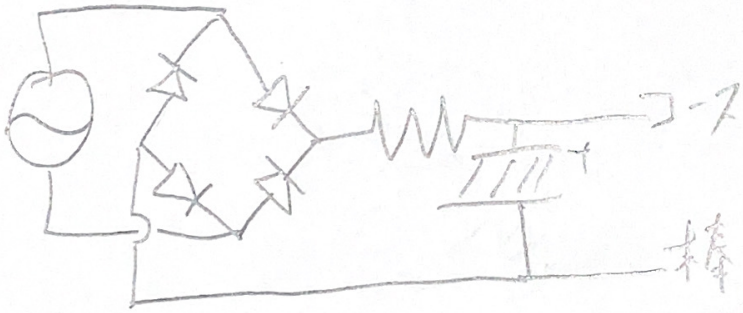
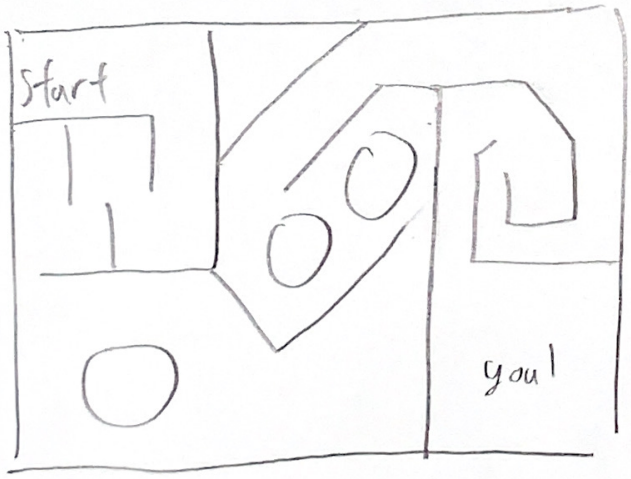
イライラ棒

M2石村
H2海野
協力 デバッグ待ちの中2
とひまな高2

ストーリー

いつまでも終らない回路図編集から逃れ、
良い夢を見るために、
レンゴクキョウジュウロウの無意識領域の
核を破壊しよう!!

回路図



コース