

2019 年度

麻布学園物理部無線班

回路図集



はじめに

昨年の回路図集では、

「米グーグル社や中国の研究チームが従来のコンピューターをはるかに上回る量子コンピューターの開発に力を入れており、世界を驚嘆させたスーパーコンピューターも音のものとなりつつあります。」

とありましたが、最近では

米国の大手通信事業者ベライゾン・ワイヤレスが家庭向け5Gサービスを開始したり、日本においても「高速かつ大容量の通信が可能」という5Gの特性を活かし、2020年の東京オリンピック・パラリンピックにおける活躍が期待されていたりと新技術の商用化がどんどん進んでおります。

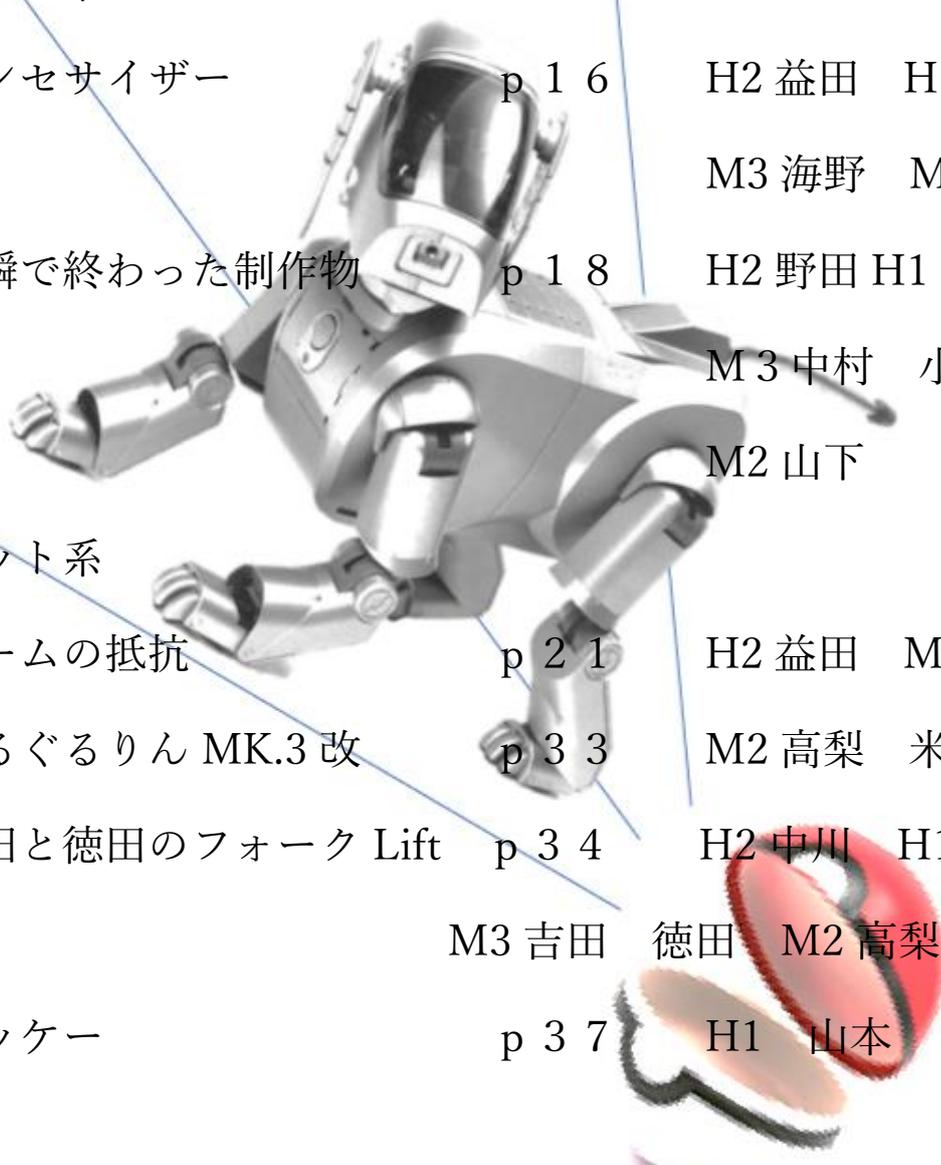
我が部では、昨年の経験を活かしたロボットや、これまでにない形を求めた製作物など多種多様な研究に挑戦しています。

この回路図集は、麻布学園物理部無線班の部員が一年をかけて研究、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。あらかじめ御了承下さい。

H2 会計 益田 隆太郎

目次

はじめに	p 3	
基礎講	p 4	
Arduino 講	p 1 2	
アナログ系		
シンセサイザー	p 1 6	H2 益田 H1 齊木 M3 海野 M2 山田
一瞬で終わった制作物	p 1 8	H2 野田 H1 小野 M3 中村 小沢 M2 山下
ロボット系		
オームの抵抗	p 2 1	H2 益田 M3 池田
ぐるぐるりん MK.3 改	p 3 3	M2 高梨 米田
吉田と徳田のフォーク Lift	p 3 4	H2 中川 H1 宮田 M3 吉田 徳田 M2 高梨 米田
ホッケー	p 3 7	H1 山本



四足ダーツ p 3 9 H1 杉田、田才

M2 有島

動きを止めるな！ p 4 2 H1 田中 円藤

M2 小清水

ダンゴムシの逆襲 p 4 5 H2 深沢 M3 高角

ゲーム系

Watch's adventures p 4 8 H 2 深沢 M3 高角
in wonderland

麻布へ GO! p 4 9 H1 田中 円藤

イライラ棒 p 5 1 M2 山田

ありふれたゲームで世界最強 p 5 3 M2 磯野

山男の壁をよける！！ p 5 6 M2 加藤

富山ラーメンの逆襲 p 5 9 M2 谷中

かませ！！対空ミサイル！ P 6 4 M2 中嶋

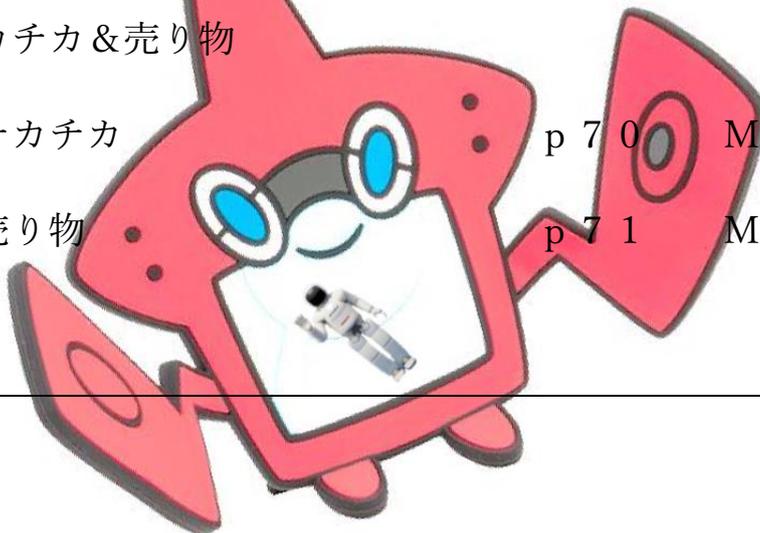
ローリングウッドの逆襲 p 6 5 M2 中澤

トレジャーハンター p 6 7 M2 畠山

チカチカ&売り物

チカチカ p 7 0 M3 池田

売り物 p 7 1 M3 中村



基礎講

講師 M3 山本大悟

1. 電気の用語

①電気

金属は電気を通すと言いますが、それは金属原子から飛び出した自由電子が動くということです。マイナスの性質を持つ電子はマイナスの電荷（負電荷）を持つと言います。それに対し、陽子はプラスの電荷（正電荷）を持つと言います。電気とはこのような電荷の移動などによって発生する物理現象のことを言います。

②電流

電流とは電荷の移動のことを言います。正電荷の流れる向きが電流の向きなので、電子の流れる向きと電流の向きが逆になります。また、ある場所を流れる電荷の量のこととも言います。単位は A（アンペア）で表します。

③電圧

電氣的な高さを電位と言います。回路上の 2 点間の電位の差のことを電位差と言います。実際にはある場所を 0V と決め、その場所との電位差を電圧と呼びます。単位は V（ボルト）で表します。

④抵抗

電流を流れにくくする力のことです。一定の抵抗を持つものを抵抗器と言います。記号は R、単位は Ω （オーム）で表します。

2. 抵抗器

(1) 抵抗器とは

抵抗器とは一定の抵抗をもつもののことを言。回路図での略号は R、記号は  や  と表す。

(2) 固定抵抗器

固定抵抗器とは抵抗値が固定された抵抗器のこと。抵抗値を変える回数が少ないときに使う。

◎種類

①カーボン皮膜抵抗器

安価で電子工作でよく使われる。形は



②金属皮膜抵抗器

カーボン皮膜抵抗器と形は似ているが、高精度である。

③酸化金属皮膜抵抗器

カーボン皮膜抵抗器と形は似ているが、熱に強く、小型でも大電流を流せる。

④チップ抵抗器

表面実装（電子部品の足を基盤に固定しない）用の小型「抵抗で、多くの製品にも使われている。

◎カラーコード

上の①～③の抵抗器にはカラーコードが付いている。カラーコードは基本、4本描かれており抵抗の大きさを示す抵抗値が分かる。

読み方

		黒	茶	赤	橙	黄	緑	青	紫	灰	白	金	銀	無
十の位	1桁目	0	1	2	3	4	5	6	7	8	9			
一の位	2桁目	0	1	2	3	4	5	6	7	8	9			
〇倍	3桁目	10 ⁰	10 ¹	10 ²	10 ³	10 ⁴	10 ⁵	10 ⁶	10 ⁷	10 ⁸	10 ⁹	10 ⁻¹	10 ⁻²	
誤差	4桁目		±1%	±2%								±5%	±10%	±20%

(3)可変抵抗器

固定抵抗器に対して、抵抗値を変えることができる。

(4)半固定抵抗器

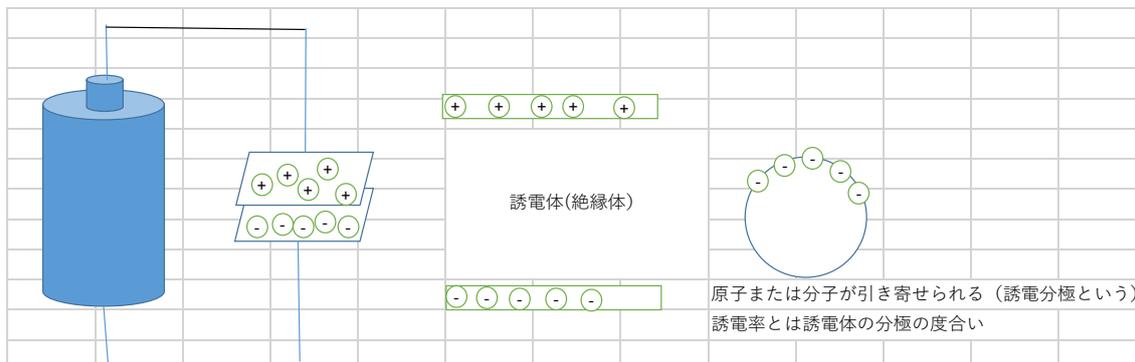
抵抗値を変える回数が少ないときに使う。1回調整するとあまり変えない。ドライバーでの調整が一般的。

3. コンデンサ

(1) コンデンサとは

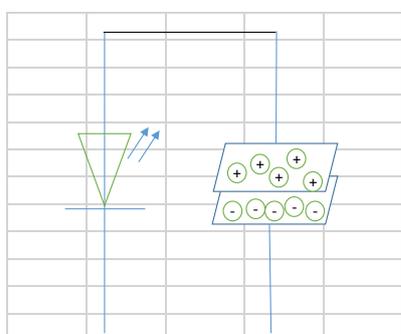
コンデンサとは電荷を蓄えるもの。単位電圧あたりの蓄えられた電荷のことを静電容量といい、記号は「C」、単位は「F」（ファラッド）。

(2) コンデンサーの仕組み



このときコンデンサーの容量は

容量 (C) = 誘電率 ϵ (エプシロン) \times 導体の面積 \div 導体の間隔
が成り立つ。



電池を取り外し、LED をつなげると、LED が光る。

(3) 種類

①アルミ電解コンデンサー

一番よく使う。極性があり、マイナスの部分に白線で描かれている。

②セラミックコンデンサー

比較的容量が小さい(\sim pF)が、高電圧 (\sim 3kV) に耐える。

③積層セラミックコンデンサー

セラミックコンデンサーより容量が大きい。小型で温度特性が優れている。

他にも、タンタルコンデンサーなどの種類がある。

(4) 値の読み方

①電解コンデンサー

\sim V は耐圧

\sim F は容量

②セラミックコンデンサー

容量を 3 桁の数字、許容差 (誤差) をアルファベットで表す。

(例) 103K

・数値 10×10^3 (pF) = 10000 (pF) = 10 (nF)

・許容差 C : ±0.25 (pF) D : 0.50 E : ±2.00 F : ±1.00

J : ±5% K : ±10% M : ±20% P : +100%、-0% Z : +80%、-20%

よって、上の例の場合だと±10%

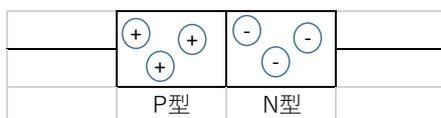
4. ダイオード

(1) ダイオードとは

ダイオードとは電流を一定の方向にしか流さないもののことを言う。

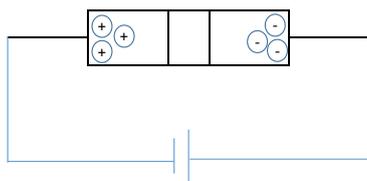
電圧を加えると光るものを発光ダイオードという。

(2) P-N 接合



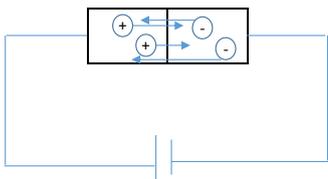
P-N 接合によって、次のように一定の方向のみに電流を流す。

①P 型側にマイナス端子をつなぐ (抵抗はさむ)



半導体の正孔はマイナス極に、電子はプラス極に引きつけられるため、電流は流れない。また、P 型半導体と N 型半導体の間に空乏層ができる。

②P 型側にプラス端子をつなぐ (抵抗はさむ)



正孔はマイナス極に、電子はプラス極に引きつけられるため、正孔や電子が移動する。

また、このとき電流は P 型から N 型の方向に流れている。

(3) LED (発光ダイオード)

P 型と N 型の電位差によって色が変わる。青色は電位差が大きい。

大きい電位差の半導体を作るのが困難であったため、青色は開発が遅れた。

5. トランジスタ

トランジスタとは増幅、スイッチング作用をする部品。半導体「P」、「N」を3つつないだもの。「PNP」、「NPN」の接合構造のものをハイポラトランジスタと言い、主にこれである。

・ピンの名前

3つの端子（ピン）があり、それぞれB（ベース）、C（コレクタ）、E（エミッタ）。

・型番

(例) 2SC○○○○○



A・・・PNP/高周波用

○○○○・・・登録番号

B・・・PNP/低周波用

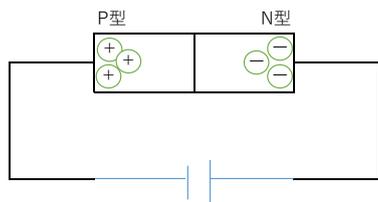
C・・・PNP/高周波用

D・・・PNP/低周波用

(1) 仕組み

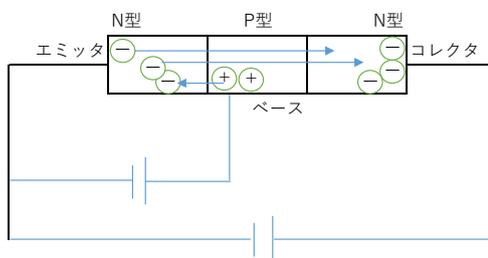
ここではNPN型を例に考える。

①まず、ダイオードで見てきたPN接合を考える。



これだと、正孔や電子は端子に引きつけられるだけで、電流は流れない。

②次に①のPN接合にN型の半導体をつける。



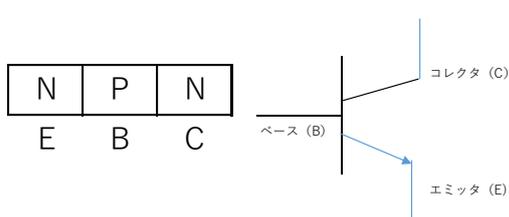
左の図で、エミッタ側の電子はコレクタ側へ移動しようとする。その電子のうち、一部はP型中の正孔と再結合するが、そのほかの電子はコレクタまで移動する。このとき、P型内の正孔が多いとエミッタ側の電子と再結合しやすくなり、エミッタ・コレクタ間

の電流が流れにくくなるので（電子が流れにくくなる）、P型は薄くする（P型内の正孔を少なくする）ことが多い。このように、ベース端子のあるP型半導体によって、通り抜ける電子の量を制御できる。

※再結合・・・再結合とは正孔と電子が一緒になって消滅する過程のこと

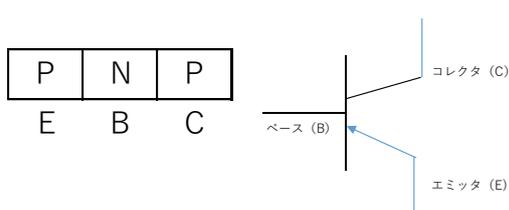
(2) 型の種類

①NPN型



B から E に電流を流すと、C から E に増幅された電流が流れる。

②PNP型



E から B に電流を流すと、E から C に増幅された電流が流れる。

(3) 増幅作用

コレクタ電流 (I_C) がベース電流 (I_B) の何倍かを示す値を直流電流増幅率と言い、 h_{FE} で表す。 h_{FE} はたいてい数十～数百を示す。

$$h_{FE} = I_C / I_B$$

(4) スイッチング作用

B→E に流す電流を C→E 間の電流を変化させるのに十分な量にしておき、B の電流を ON-OFF させることで、C→E 間も ON-OFF させること。

Arduino 講

昨年度講師 H1 田中基樹 円藤颯太

1 Arduino とは？

Arduino とは PIC と同じようにマイコンで、パソコンからプログラムを書き込むことができます。Arduino と PIC との違いは PIC だととても長いプログラムを大幅に省略することができ、簡単に様々なプログラムを書くことができるという点です。ですが、PIC は 100 円と安価なのですが、正規の Arduino だと 3000 円近くするというデメリットもあります。そのため、ぶつむでは 1000 円ほどで買える **びんぼー**でいいの というものを使います。プログラムの書き方は、Arduino と何も違いはないので、こっちを使ったほうがお得ですね。

2 Arduino の基本的な説明

Arduino は PIC とは違い、アナログ専用のピンとデジタル専用のピンが分かれています。また、Arduino のファイルスケッチの例からプログラムの例を見ることができるので、ぜひ使ってみてください。

3 Lチカのプログラミング

また、Arduino は C++言語という言語を使って書きますが、基本と黄なことは C 言語と全く変わらないので、Arduino を使う場合は C++言語を勉強しなければいけないということはありません。それでは実際にプログラムを見てみましょう。

```
void setup(){//プログラムを実行したときに、一回だけ実行されます。
pinMode(13,OUTPUT);
}
void loop(){//プログラムを実行すると、setup()の後は、loop()が実行され続きます
digitalWrite(13,HIGH);
delay(1000);
digitalWrite(13,LOW);
delay(1000);
}
pinMode()関数で 13 番ピンをアウトプット設定にします。
digitalWrite()関数でピンの high or low を設定します。
```

delay()関数で 1000ms(1s)待つと設定します。
これで 1 秒おきに LED がチカチカするプログラムの完成です。

4 Servo モーターのプログラム

```
#include<Servo.h>    //ヘッダーファイルの読み込み
Int pos=0;
Servo myServo;      //Servo に名前を付けます。詳しくは C++言語で勉強します。
void setup(){
myServo.attach(9);  //9 番ピンを使う。
}
void loop(){
for(pos=0;pos<=180;pos++){
myServo.write(pos);
delay(10);
}
for(pos=180;pos>=0;pos--){
myServo.write(pos);
delay(10);
}
}
```

Servo の write()関数で名サーボモーターを何度の各出に移動させるかを命令します。
また、この関数を使う時は、delay()関数で 10ms 程間を開けておきましょう。そうしない
とサーボモーターの処理が追いつきません。

5 Arduino 間の通信

Arduino 同士の通信は PIC に比べるととても簡単に行うことができます。通信の方法は大
まかに 3 つの方法があります。SPI 通信 I2C 通信シリアル通信です。SPI 通信と I2C 通信
は CLK と同期してデータを送る通信方法です。ここではシリアル通信について詳しく見て
いきたいと思います

シリアル通信

シリアル通信はこれらの中で一番汎用性があります。

なぜシリアル通信が汎用的かというと、シリアル通信によって、Arduino 間の通信だけでな
くパソコンとの通信もすることができるからです。それではまず始めに、パソコンとの通信
を見てみましょう。

```
void setup(){
Serial.begin(9600);    //シリアル通信の速度を指定します。9600 や 115200 が一般的
```

```

}
void loop(){
Serial.println("a");
delay(1000);
}

```

このプログラムを実行すると Arduino のツール — シリアルモニターを開くと、a と 1 秒ごとに表示されます。このシリアル通信を使って、データをパソコンの画面に表示することができます。

それでは Arduino 間の通信についてみていきましょう

送信側のプログラム

```

#include<SoftwareSerial.h>
SoftwareSerial mySerial(2,3); //RX:2,TX:3
void setup(){
mySerial.begin(9600);
}
void loop(){
mySerial.write("hello");
}

```

受信側のプログラム

```

#include<SoftwareSerial.h>
SoftwareSerial mySerial2(2,3); //RX:2,TX:3
void setup(){
mySerial.begin(9600);
}
void loop(){
if(mySerial2.available()){
Serial.println(mySerial2.read());
}
}

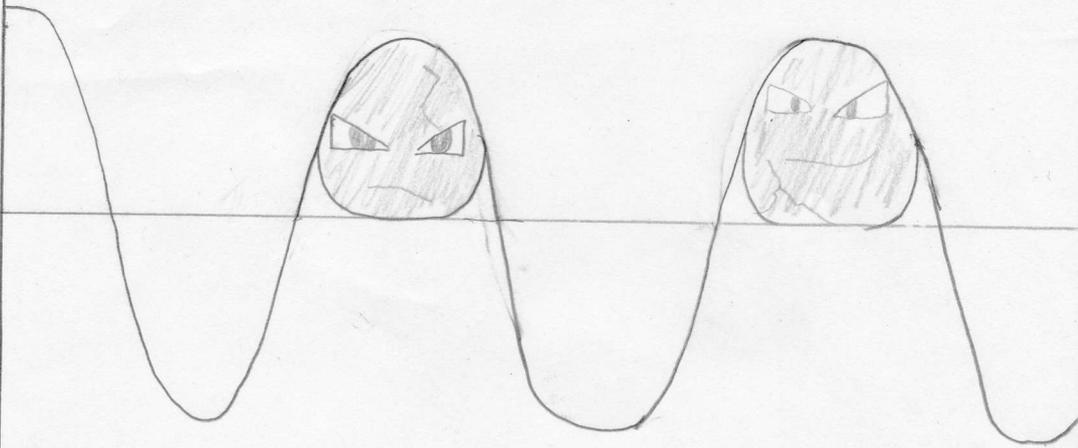
```

まずは#include でファイルをインクルードします。

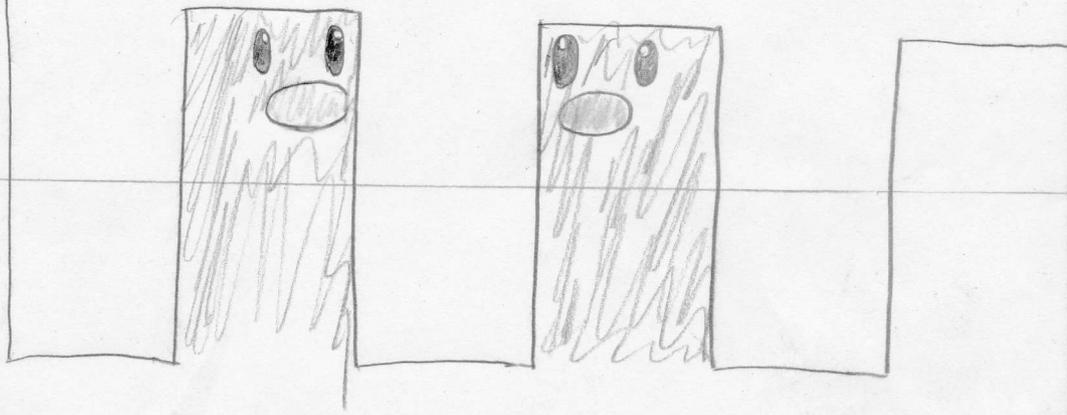
次に、SoftwareSerial mySerial2(2,3);ここで使用する SoftwareSerial に名前を付け、RX と TX を指定します。RX はデータを受け取る側 (receive) TX はデータを送信する側 (transmission) です。write()関数でデータを送信します。available()関数でシリアル通信が行うことができるかを調べ、シリアル通信が行われているときは、read()関数でデータを読み込みそれをシリアルモニターに表示するというプログラムです。

この方法を使うことによって、processing ともシリアル通信をすることができます。

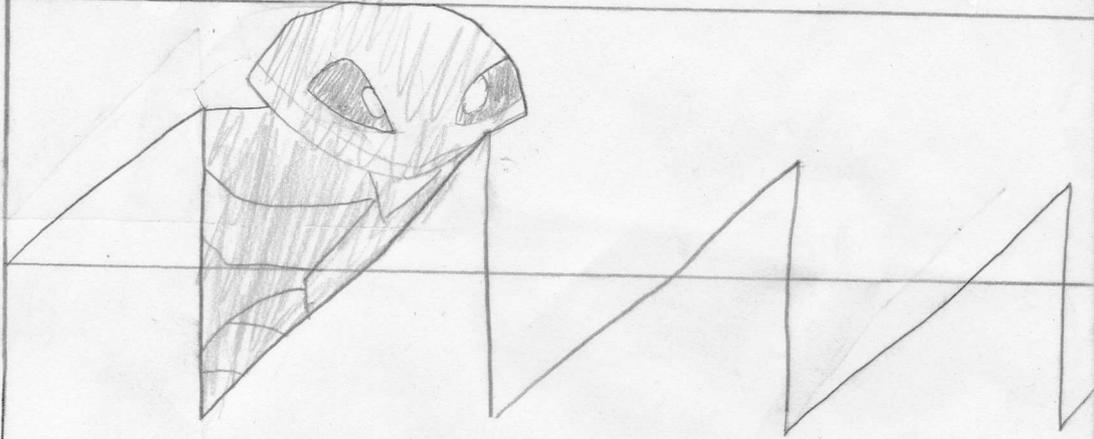
正弦波



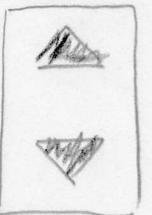
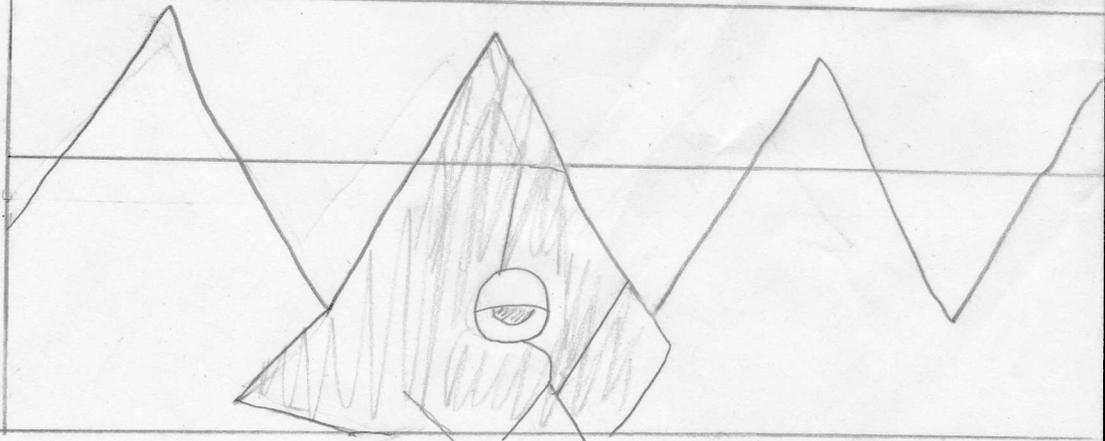
矩形波



のこぎり波



三角波



神からもらったチート（アナログシンセサイザー）で惑星を守護するらしい

製作者・H2益田隆太郎さん
 ・H1齊木有悟さん
 ・M3海野博亮
 ・M2山田恭平

STORY

ブラック部活に日夜通うごく普通の高校生のM田さんは、ある日後輩のU野くんの製作物の爆発からS木さんをかばい五臓六腑がはじけ飛んで死んでしまう。

しかし、それを見ていた神のY田にチート（アナログシンセサイザー）をもらって多くの脅威にさらされる惑星に転生する。

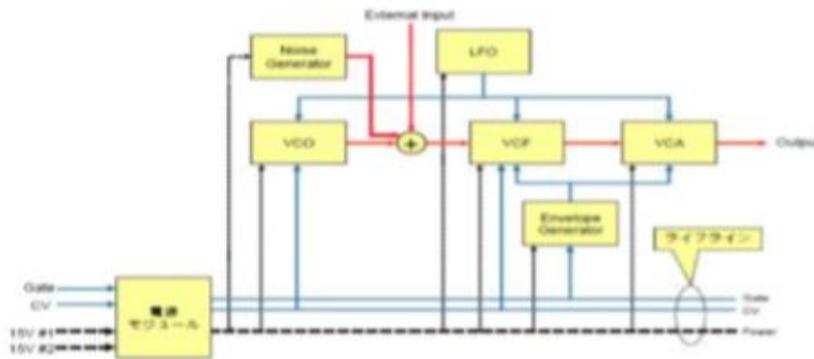
惑星をめぐる宇宙規模の争いとは?
 惑星を裏で牛耳る組織モンキーパラダイスの陰謀は?
 すべてを操る黒幕が今、明かされる!?

次回 「黒幕サルース」

概要

ストーリーは「全く」関係ないです。

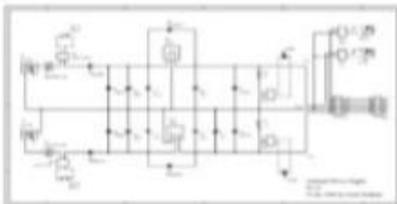
アナログシンセサイザーは下の図のようにいくつかのアナログ回路を使って音を出すものです。



電源モジュール

回路図

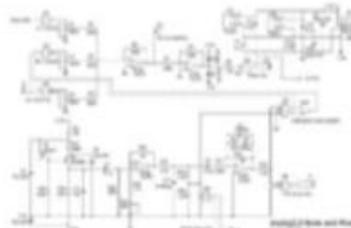
電源モジュール



内容は名前のままで電源の回路です。

ノイズジェネレーターとミキサー

回路図



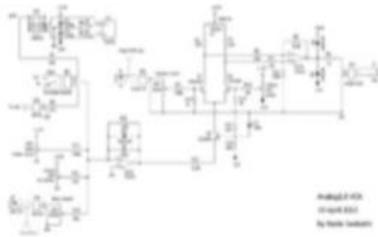
ノイズジェネレーター...ホワイトノイズとピンクノイズというノイズを出す。

ミキサー...簡単に言うと音を出す。

VCA

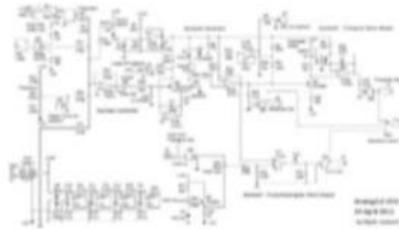
VCO

回路図



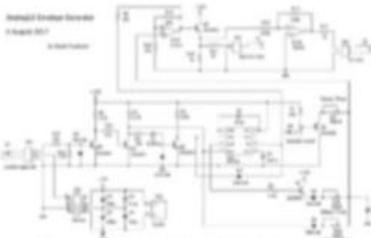
音量を調節する回路です。
エンベロープジェネレーター

回路図



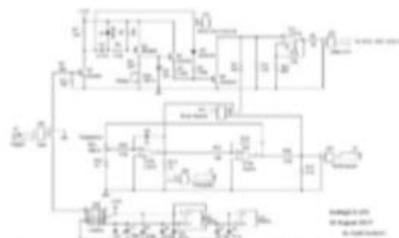
基本の波形を出す。
LFO

回路図



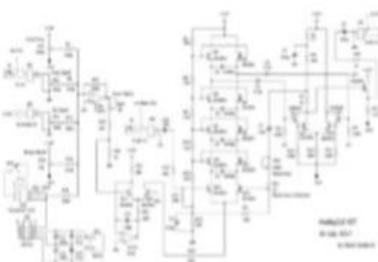
VCAとVCFの時間を制御する回路。VCAとVCF、VCOの周期を変化させる。
VCF

回路図



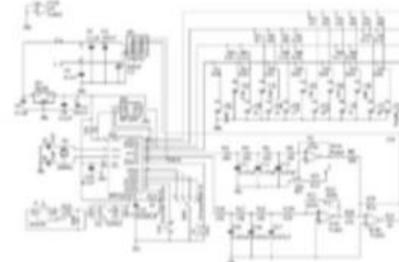
鍵盤

回路図



波形を変化させる

回路図



音程を決める。

感想

益田：楽しかったです。皆優秀でよかったです。

斉木：いろんなことが学べたと思う。楽しかったです。

海野：いくつかの回路を担当しましたが、役目はまっとうできたのではないかと思います。最後に一言、また僕何かやっちゃいました？

山田：初めての製作でしたが、何とか自分の担当ができてよかったです。

一瞬で終わった製作物

製作: H2 野田 協力: 物無負
 H1 小野 大平
 M3 中村
 小沢
 M2 山下

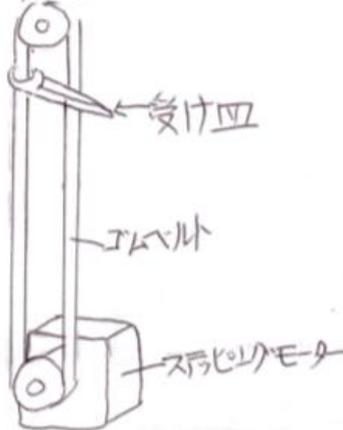
・ストーリー

9月上旬、我が物理部 無線班まで製作期間が始まった。部員達が製作を始め殺気ちている中、大人の男がう叫んだ。「完動しました—————!!」

・概要

- ・アルミ球を鉛動で車云かす装置です。
- ・仕掛けはほとんどアルミで制御しています。
- ・スピニングモーターには298Nを巻いてつないでいます。

・リフト



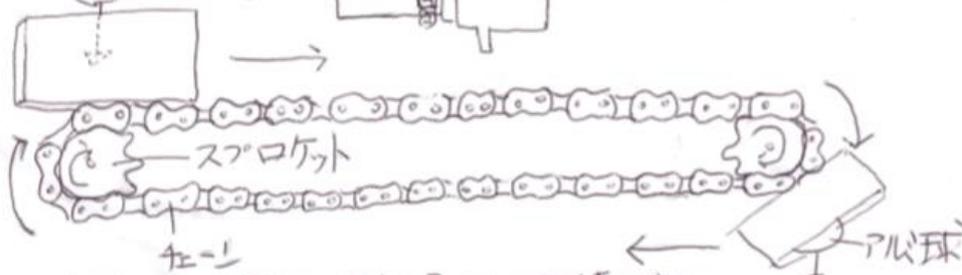
1. アルミ球が受け皿に来たらスピニングモーターを回して受け皿を上にも上げます。

受け皿

2. ①→②の順に回り、アルミ球をレベルの上に落とし時。

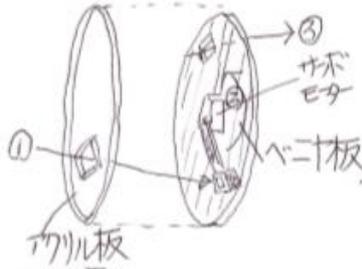
ベルトコンベアー

アルミ球



アルミ球がゴンドラに入ってきたら、スプロケットをつけたスピニングモーターが回り、ゴンドラが移動し時。

観覧車

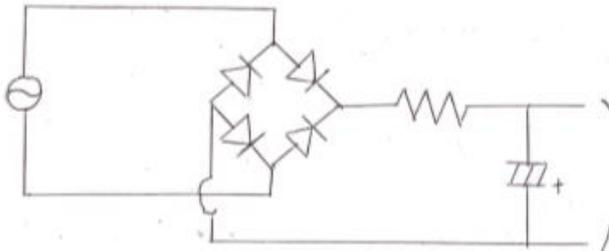


1. アルミ球が入る

2. サーボモーターが回す

3. アルミ球が出る

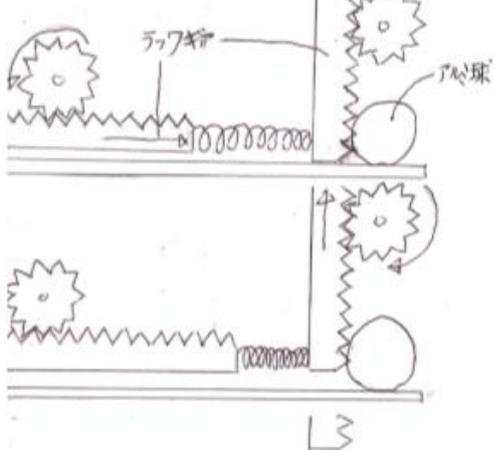
放電



この間アルミ球が通ると放電します

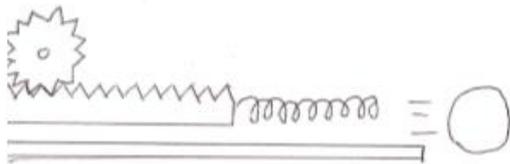
大石包

ラックギア×2とギアモーター×2とバネを使います。



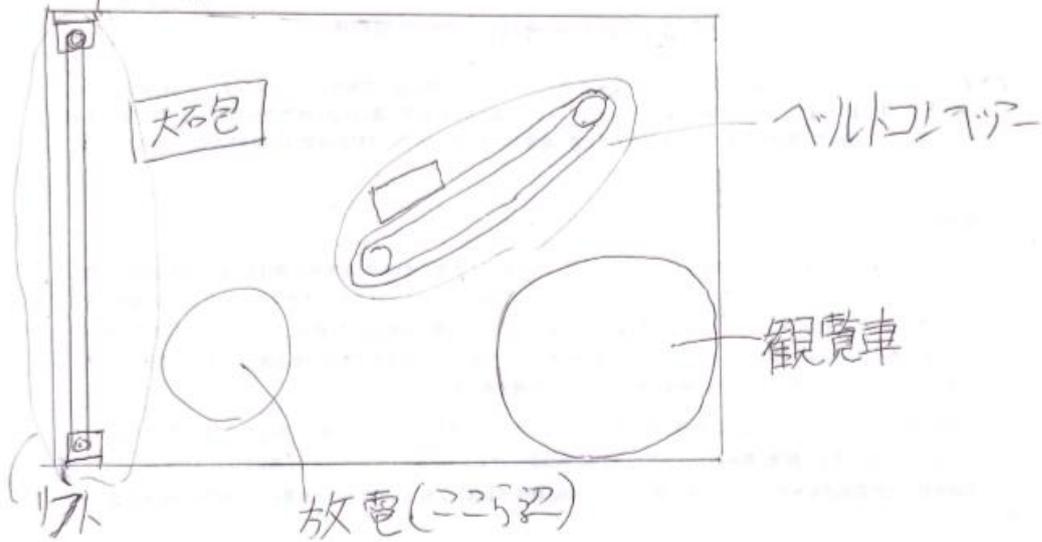
1. ギアモーターを回し、ラックギアでバネを押し縮めます

2. バネがある程度縮まったらバネを押し込んでいるラックギアを持ち上げます



3. 押し込みがなくなったバネが元に戻るのを利用してアルミ球をはじきだします

全体図



感想

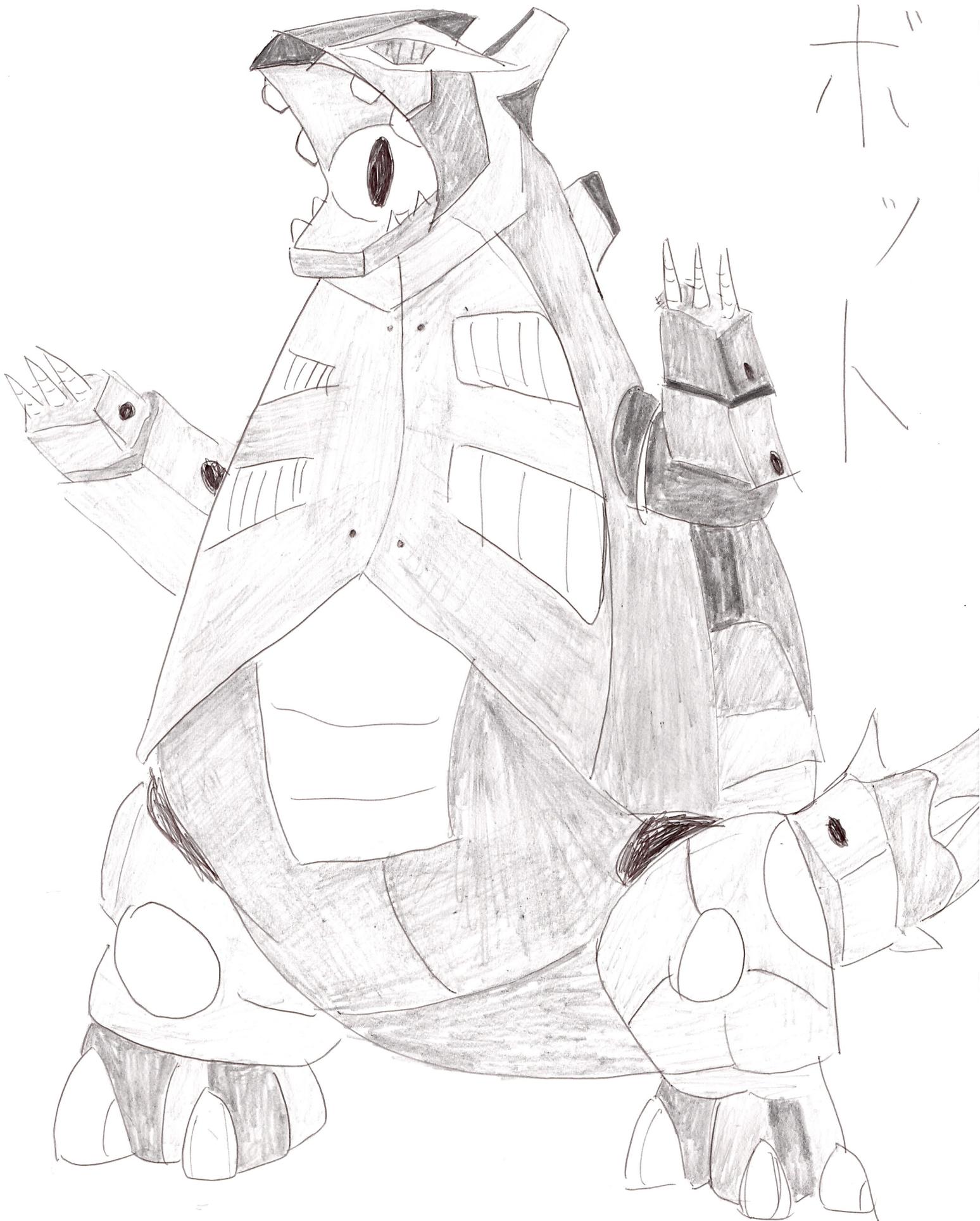
野田 大石と本体が担当しました。本体はだいたい終わりましたが大石が全然終わらなかったのは反省したいです。11かせる来年は来ないですか

小野 楽しかったです

水戸 いろいろと反省もしたが、先輩に迷惑をかけてしまいました。中のときも製作を楽しむことができたので良かったです

中村 かなり積極的に製作を進めたので来年は頑張りたいです

山下 出席数が多い多くなかったので来年はもっと部活に行く日を増やしたいです



ボ

ツ

ト

オームの抵抗

H2 益田 隆太郎 M3 池田 宜史

ストーリー

「ドドドドドドドド」地響きが鳴り渡った。オームはI田を怒らせたのである。踏みつぶそうとするI田にオームがとった最後の抵抗とは…

概要 このロボットはジブリの「風の谷のナウシカ」に出てきた巨大な虫である王蟲を再現しようとしたロボットです。「足を上げる→つま先を出す→足を下げる→地面をつま先でける」という動きを繰り返して、前進、後退、旋回をします。



これが王蟲です。

機構

簡単な構造です。図1が中身で、この機構の外に厚紙で作られた殻と図2の機構がついています。

図1

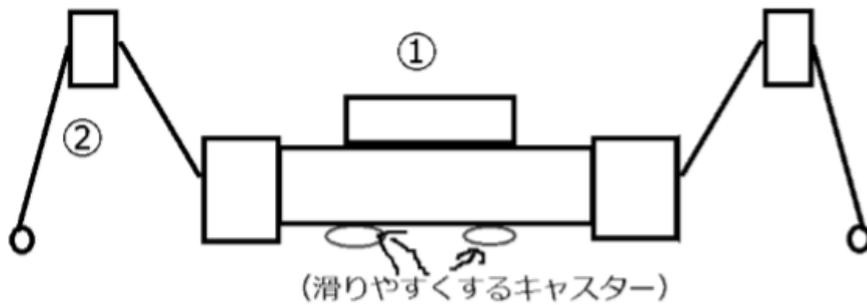
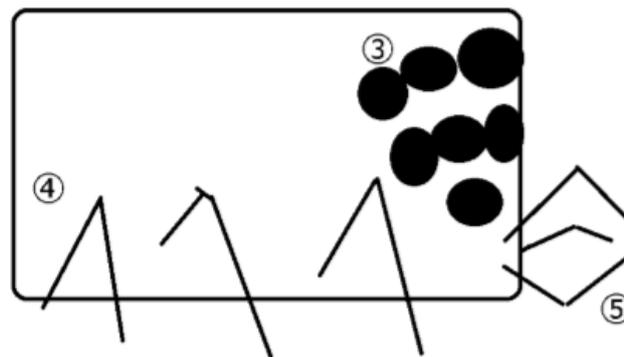


図 2



①基盤…ここには2つの基盤があります。一つ目は、足のモーターの動きとコントローラーを連動させたり、目のLEDの光り方を制御したりするためのプログラムを書きこむarduinoと呼ばれる基盤です。二つ目は、モーターとLEDを動かすために電源と繋げる基盤です。二つ目の基盤はただ繋げているだけです。最後にプログラムと説明を載せます。

②足…足は二本の金属と二個のモーターで出来ていて①の基盤で制御されています。肩

や肘にあたる部分が動き、概要で書かれている動きをします。つま先にはゴムの滑り止めがあります。飾り足とは違います。

③目…原作の王蟲は怒っているときに目が赤く光り、普通の時は目が青に光ります。なのでそれを再現し、動いているときには赤く光り、動いていないときに緑に光らせています。

④…飾り足

⑤…口

プログラム

```
#include <Servo.h>//モーターを動かすために必要です。

Servo frkt;//右の前の肩

Servo frhz;//右の前の肘

Servo flkt;//左の前の肩

Servo flhz;//左の前の肘

Servo hrkt;//右の後ろの肩

Servo hrhz;//右の後ろの肘

Servo hlkt;//左の後ろの肩

Servo hlhz;//左の後ろの肘

int aa = 0;//モーターの動かす角度を示すために使います

int bb = 0;//上と同じです

void setup() {

  pinMode(2, INPUT);//コントローラーからの前移動の入力を受け取ります

  pinMode(4, INPUT);//後ろ移動の入力を受け取ります
```

```

pinMode(7, INPUT); //右旋回の入力を受け取ります
pinMode(8, INPUT); //左旋回の入力を受け取ります
pinMode(11, OUTPUT); //目を赤く光らせます
pinMode(12, OUTPUT); //目を緑に光らせます
frkt.attach(5); //ここから
frhz.attach(6); //
flkt.attach(9); //
flhz.attach(10); //
hrkt.attach(A0); //
hrhz.attach(A1); //
hlkt.attach(A2); //
hlhz.attach(A3); //ここまではそれぞれのモーターへの出力です
}

void loop() {
  if (digitalRead(2) == HIGH) { //もし前移動が来たらこれを実行するという事
    for (aa = 45; aa >= 0; aa--) {
      bb = 0;
      bb++;
      frkt.write(aa); //何度動くかの指定 ここから
      flkt.write(aa);
      hrkt.write(bb);
      hlkt.write(bb); //ここまで
      digitalWrite(11, HIGH); //目を赤く光らせます
    }
  }
}

```

```

digitalWrite(12, LOW); //目を緑に光らせません
delay(5); //1000分の5秒待つ
} //肩を上げる/下げる
for (aa = 0; aa <= 120; aa++) {
    hrhz.write(aa);
    hlhz.write(aa);
    frhz.write(aa);
    flhz.write(aa);
    digitalWrite(11, HIGH);
    digitalWrite(12, LOW);
    delay(5);
} //肘を後ろに下げる/前に出す
for (aa = 0; aa <= 45; aa++) {
    bb = 45;
    bb--;
    frkt.write(aa);
    flkt.write(aa);
    hrkt.write(bb);
    hlkt.write(bb);
    digitalWrite(11, HIGH);
    digitalWrite(12, LOW);
    delay(5);
} //肩を下げる/上げる

```

```

for (aa = 120; aa >= 0; aa--) {
    frhz.write(aa);
    flhz.write(aa);
    hrhz.write(aa);
    hlhz.write(aa);
    digitalWrite(11, HIGH);
    digitalWrite(12, LOW);
    delay(5);
    //肘を出す/下げる
} //前終わり
} else { //もし前の入力 came なかったら次に進む
    if (digitalRead(4) == HIGH) { //もし後ろの入力がきたら実行
        for (aa = 0; aa <= 45; aa++) {
            bb = 45;
            bb--;
            hrkt.write(bb);
            hlkt.write(bb);
            frkt.write(aa);
            flkt.write(aa);
            digitalWrite(11, HIGH);
            digitalWrite(12, LOW);
            delay(5);
        } //前と逆の動きをして後ろに進みます
    }
}

```

```
for (aa = 0; aa <= 120; aa++) {  
    hrhz.write(aa);  
    hlhz.write(aa);  
    frhz.write(aa);  
    flhz.write(aa);  
    digitalWrite(11, HIGH);  
    digitalWrite(12, LOW);  
    delay(5);  
}  
//  
for (aa = 45; aa >= 0; aa--) {  
    bb = 0;  
    bb++;  
    frkt.write(aa);  
    flkt.write(aa);  
    hrkt.write(bb);  
    hlkt.write(bb);  
    digitalWrite(11, HIGH);  
    digitalWrite(12, LOW);  
    delay(5);  
}  
//  
for (aa = 120; aa >= 0; aa--) {  
    frhz.write(aa);  
    flhz.write(aa);
```

```

    hrhz.write(aa);
    hlhz.write(aa);
    digitalWrite(11, HIGH);
    digitalWrite(12, LOW);
    delay(5);
} //うしろ終わり
} else { //もし前移動と後ろ移動の入力がなかったら次へ
    if (digitalRead(7) == HIGH) { //もし右入力が来たら実行
        for (aa = 45; aa >= 0; aa--) {
            bb = 0;
            bb++;
            flkt.write(aa);
            hlkt.write(bb);
            digitalWrite(11, HIGH);
            digitalWrite(12, LOW);
            delay(5);
        } //肩を上げる/下げる
        for (aa = 0; aa <= 120; aa++) {
            flhz.write(aa);
            hlhz.write(aa);
            digitalWrite(11, HIGH);
            digitalWrite(12, LOW);
            delay(5);

```

```

} //肘を後ろに下げる/前に出す
for (aa = 0; aa <= 45; aa++) {
    bb = 45;
    bb--;
    hlkt.write(bb);
    flkt.write(aa);
    digitalWrite(11, HIGH);
    digitalWrite(12, LOW);
    delay(5);
} //肩を下げる/上げる
for (aa = 120; aa >= 0; aa--) {
    hlhz.write(aa);
    flhz.write(aa);
    digitalWrite(11, HIGH);
    digitalWrite(12, LOW);
    delay(5);
} //みぎ終わり
} else { //もし前もうしろも右も入力がなかったら次へ
    if (digitalRead(8) == HIGH) { //左入力が来たら実行
        for (aa = 45; aa >= 0; aa--) {
            bb = 0;
            bb++;
            hrkt.write(aa);

```

```

frkt.write(aa);
digitalWrite(11, HIGH);
digitalWrite(12, LOW);
delay(5);
} //肩を上げる/下げる
for (aa = 0; aa <= 120; aa++) {
frhz.write(aa);
hrhz.write(aa);
digitalWrite(11, HIGH);
digitalWrite(12, LOW);
delay(5);
} //肘を後ろに下げる/前に出す
for (aa = 0; aa <= 45; aa++) {
bb = 45;
bb--;
hrkt.write(bb);
frkt.write(aa);
digitalWrite(11, HIGH);
digitalWrite(12, LOW);
delay(5);
} //肩を下げる/上げる
for (aa = 120; aa >= 0; aa--) {
frhz.write(aa);

```


ぐるぐるぐるりん MK.3 改

製作者 M2 高梨勇之介

M2 米田賢史

協力者 物無の皆様

ストーリー

物無員 T 梨は今年の展示テーマに反抗し好きなゲームのロボットを作ろうとしていた…

T 梨：「虹6 楽しいンゴwww」

Y 田：「おっ、そうだな（適当）」

T 梨：「せや、虹6 のドローン二次製作にしたろwww」

Y 田：「去年もそんなのなかったっけ？」

T 梨：「そんじゃ作るか～（聞こえないふり）」

クソ茶番失礼しました

概要

PC でカメラの映像を見ながらリモコンで動かせる某 FPS に出てきたようなロボットです。（ジャンプは出来ない）

去年ぐるぐるぐるりん改という製作物があったのですが（3月23日現在知った）大きく異なる点はラズベリーパイではなく Arduino を使っていること、モーターをサーボモーター2個 DC モーター2個の計4個使っていることです。また電源は3V を使っているので事故などの危険性が少ないです。

感想

高梨 中1 が2人だけしか集まらず作業が進まなかったです。

また助言をしてくれた宮田さんや益田さんには感謝してます。

「某 FPS」を知っている人がいたら話しかけてくれると嬉しいです。

米田 中1 のみの製作ということもあって色々と先輩の助言なども頂いて製作することができました来年はもっと自分で考え自主的に行動できるようにしたいです。

吉田と徳田のフォーク Lift

【参加者】 H2 中川さん H1 宮田 M3 吉田 徳田 M2 高梨 米田 【協力】 物無の皆さま
ストーリー

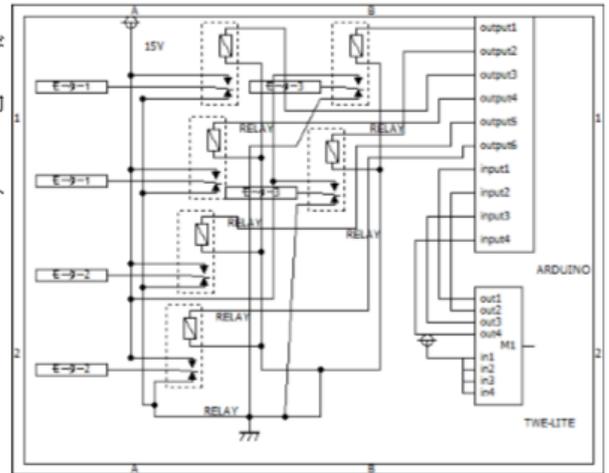
6人くらいで作ったフォークリフトです☆
本体設計様、プログラム担当様、基盤作成者様、物理部無線班の皆様ありがとー☆
物無員が7万円でリフトを作ってあげる制作物です☆

概要

これは、モーター六つで力強く駆動し、二つの爪（フォーク）で物をつかんで持ち上げるフォークリフトです。後述するコントローラーで無線操作できます。

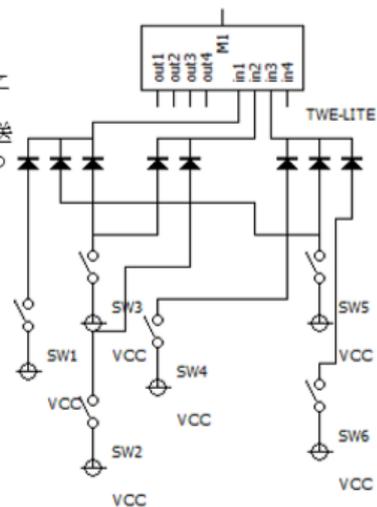
制御基板

Arduino というマイコンを使って制御しています。ギヤードモーターは、Arduino の出力電流では回らないので、右の図のようにリレーを使って15V電源を制御する回路を組みました。しかし、リレー1つだとモーターを回すか回さないかしか制御できないため、モーター1つにつき二つリレーを使用することで、両方向への回転をできるようにしました。



コントローラー

無線機は twe-lite というものを使用しています。とても簡単に無線ができるのですが、入力ピンが四つしかありません。そこで、右図のような回路で2進数で今押しているボタンの番号を送信して、それを Arduino で変換することで、四つのピンで六つのボタンの情報を送ることにしました。



プログラム

シンプルです。2進数を10進数に変換するだけでした。具体的に言うと、

1の位×1 + 2の位×2 + 4の位×4

をします。

この式の答えを参照して前進・後退・右旋回・左旋回を切り替えています。

本体

概要で書いた通りまさにフォークリフトです。ただ何と言ってもその大きさは600mm×800mm、フォーク部分（上下するところ）の最大の高さ500mmという巨大なもので総重量は10kg弱にもなります。このフォークリフト（という名の巨大ラジコン）を駆動部分と素材の部分に分けて解説したいと思います。

・駆動部分

まずはフォークリフトたるもの動かなくてはならない！！というわけでタイヤの解説からして行きます。しかし前述した通り本体は重く並みのモーターでは動きません。というわけでこの製作で用いたのは「AO-8033 タミヤギヤードモーター 540K75」と呼ばれるものです。

このモーターは一個につき12kg/cmという ↓AO-8033

高いトルクを持っており、ここではそれを4つ用いることで安定した動きを実現させています。

そしてフォークリフトの肝であるフォーク部分では、さらに強力な

「AO-8037 タミヤギヤードモーター 540K300」を使用しています。

これは一つにつき48kg/cmという物無史上最高クラスの高トルクのモーターで、これで様々なものを確実に持ち上げることができます。こうして（モーターだけで3万円以上かかったことはさておき）フォークリフトは動くことができます。

・素材の部分

過去最重量級の製作物とだけあって3Dプリンターの合成樹脂や木材では到底強度が足りません。というわけで今年新たに導入したのが「アルミフレーム」と言われる加工されたアルミです。

これは右図のような特徴的な断面をしており、刻まれた ↓アルミフレーム

溝に専用のナットをはめることで切断や接着を一切することなくそれら以上の強度をもたらすことが可能です。これらをベースに枠を作り厚さが5mmにもなるアクリルを固定し本体を構成しています。

フォークの爪の金属部分はアルミでは強度不足なので鉄板を用いて、それを補っています。また、上下の機構は金属製のチェーンを用いて絶対に切れないようにし不慮の事故を防いでいます。



まとめると、（お金の暴力で）新しいものを多数取り入れているのでその力強さを感じながらみてください。

感想

米田

物無の初めての製作でわからないことも多かったのですが先輩方に教えていただきなんとか完動することができました。来年はもっと自力でできることを増やしたいと思います

高梨

かなり出席率が低かった事もあり、かなり先輩方に迷惑をかけてしまいましたが、結果的に完成して安心しています。

徳田

割と早い時期に完動できてよかったです。中1との絡みが少なかったのが心残りです。もう少し関われたかなと思います。

吉田

回路やコントローラーは何度も試行錯誤して作り直すなどしていましたが、ardino や twe-lite について学ぶことができましたし、早く完動できてよかったです。

宮田 簡単なプログラムを書くのに長い時間をかけてしまい、すいませんでした。はんだ付けが下手なのも直したいです。後輩ともそこそこ絡めてよかったです。

中川さん

とりあえずでっかいものを作りたい欲求は前からあったので、最後の年に叶ってうれしいですね。しっかり動いているし新たな素材も多数使えて満足です。共同製作者には迷惑かけ続けてた気もしますが、最後まで付いてきてくれてありがとう！！

ホッケー

製作者：H1 山本
製作協力者：物無の皆様

1. 外観・仕様

横：約 83cm
縦：約 100cm
高さ：約 130cm

基本的にはゲームセンターなどにある一般的なホッケー台と形は変わりませんが、大きさはそれらの半分以下です。もともと縦は 2m にする予定でしたが、安定性やカメラ（後述）の都合により、この大きさになりました。

(図 1)



(図 1) 全体像



(図 2)

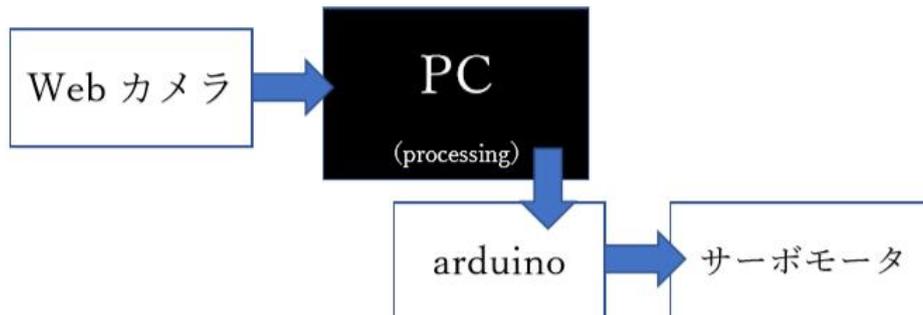
また、一般的なホッケー台は、エアホッケーになっており空気が下から出ますが、これは空気が出ず、球を転がす形になっています。ただ、ホッケー台を作るだけでは図工の時間になってしまうので、上部（高さ約 120cm）にカメラを取り付けて、球を認識し、コンピュータ側の腕が自動で動くようになっています。カメラは Logicool の 1000 円ぐらいの web カメラを使っています。



(図 3)

2. 制御方法

制御方法は web カメラで認識した位置をもとに、パソコンで腕の角度を計算し、arduino で腕を回すサーボモータを制御しています。



このような図のようにデータを送り、制御しています。

①Web カメラ→PC (processing)

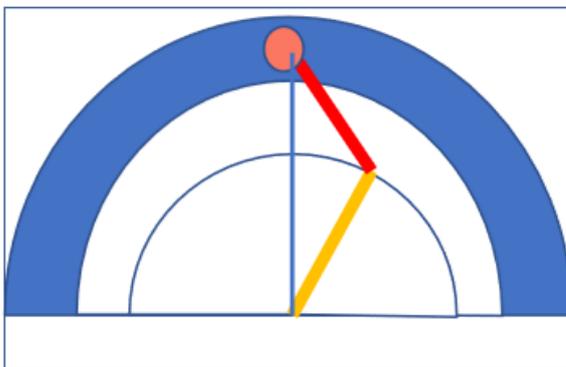
Web カメラから送られてきた画像データをもとに、processing というソフトでパソコンの画面に表示し、球の座標や腕の角度を計算します。

◎球の座標の計算方法

プログラムは割愛しますが、基本的には Web カメラから送られる画像をパソコンの画面に表示し、画面上の球の色のピクセルの平均を出します (赤い球なら、赤く表示されているピクセルの平均)。出したピクセルの平均値を球の位置とします。

◎腕の角度の計算方法

目標となる位置へ、2本の腕を動かします。サーボモータは基本的に 0~180 度までしか動



かないので、ラケットを動かせる範囲は限られます。腕の本数を増やせば、動かせる範囲は大きくなります。

左図の青い部分がラケットの動かせる範囲です。2本の腕ともう1辺で三角形を作ることによって、サーボモータの角度を求めることができます。

②PC→arduino

パソコンで計算した腕の角度のデータを arduino にシリアル通信で送信し、サーボモータを動かせるようにします。

③arduino→サーボモータ

Arduino からサーボモータに回路をつなぎ、サーボモータを動かします。

3. 感想

今回の製作では、1 から全てを自分で考え、作りました。初めてのことが多く、時間がかかってしまった部分もありましたが、完成に近づいているので良かったです。画像認識やサーボモータの角度計算は意外と楽しかったです。最後に製作に協力して頂いた方々ありがとうございました。

四足ダーツ

共同製作者 H1 杉田太郎 M2有島 H1 田才圭
協力 物理部メンバー

ストーリー

「どうしたんだいの○太君」
の○太は、友達皆ラジコンを持っているのに、自分だけ持っていないで悔しがっていた。
「そんな君にはこの四足ダーツをあげよう」
「何？このロボットみたいなの」
「この四足ダーツは、四足歩行の操作ができるうえに、ダーツができるんだよ」
その後、の○太はこの四足ダーツを使って友達と一緒にラジコンで遊びつくした。

概要

ストーリーは創作が苦手なので某作品を使ってしまいました。先輩の提案で四足ダーツを作ることになり、最初は的を作る予定だったのですが中断し、ロボットに専念することになりました。

説明

自分的にこんな感じかなと理解しているだけなので、不備が多いと思います。

1 回路等

自分にもよくわからない部分が多いですが、TA7291 や arduino など方向など、操作をしています。

2 仕組み

オムニホイールというタイヤで四方向に動くのを実現しています、たまにタイヤが勝手に動きますが、自重により動かないと信じています。

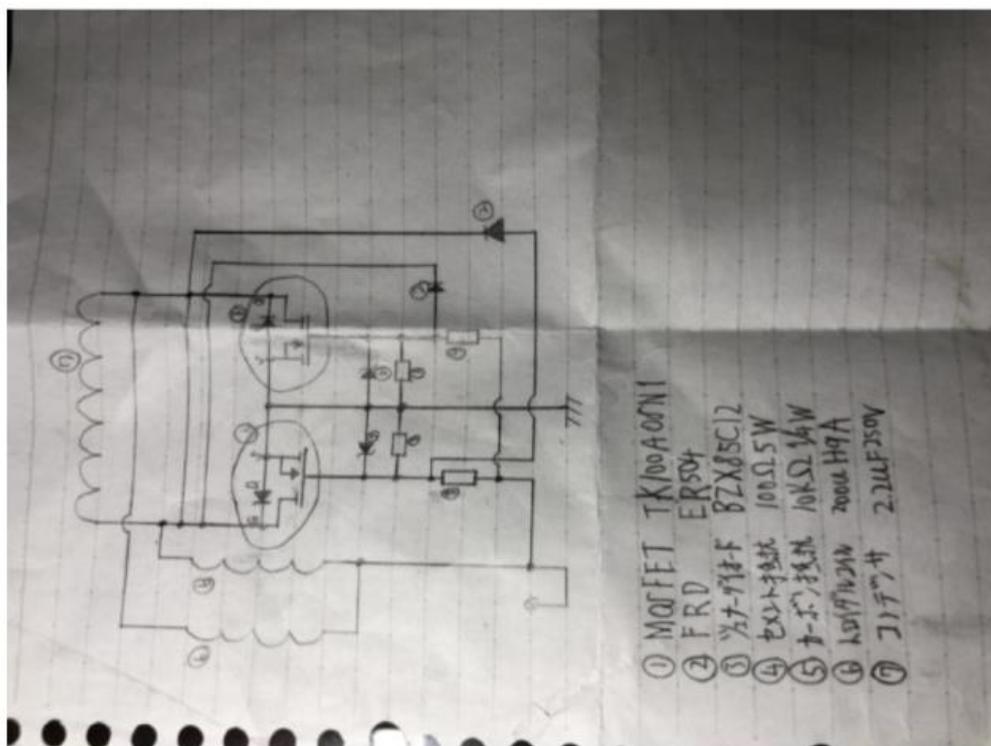


また、ダーツをするうえにあたってコイルガンを使用しますが、コイルガンの重さでつぶれないかという不安もあります。

オムニホイールは、車輪に回転する部分は何個もあり、これで横への回転もしています。



念の為、コイルガンの構造です。



感想

有島 先輩の杉田さん達と作りました。自分はこれが初めての製作で、何回も自分が

arduino 等に無知なのがわかりました。この製作で分かった知識を次の製作に繋げていけたらと思います。

動きを止めるな！

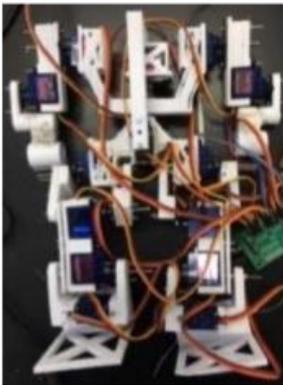
製作者 高1 田中 円藤
中2 小清水
協力 ぶつむの皆様

1 ストーリー

ある剣道の授業からの帰り道、その日は今学期最後の授業だったので、僕は剣道の竹刀と防具を運んでいた。まだ眠かった僕はうっかり竹刀を階段から落としてしまった。面倒だなと思いつつ取りに行くと、そこには髪を逆立て一層殺気立った港区1の剛腕、S木君の姿が！すべてを察した僕は謝ろうかとも思ったが、あることをひらめいた。そうだ二足歩行ロボットに防具を付ければ、僕が行かなくてもばれないじゃないか！！

(S木君は柔道部ですが、すごくいいやつです。)

2 全体像



3 概要

よくある二足歩行ロボットに見えますが、3Dプリンターによって作られているのでとても軽いです。そのため、正確に動かさなければ転倒してしまいます。そこで、習いたての三角比を使ってみました。動かしたい場所と目的地の座標を指定すると自動でサーボモーターのお角度を自動で求めてくれるというものです。



原理は上の図に書いてある通りです。

これらのサーボモーターは Arduino とサーボモーターを I2C 通信によって操作できるドライバーを使いました。

プログラムでは、 \cos の値から角度を求めてくれる $\text{acos}()$ 関数や \tan の値から角度を求めてくれる $\text{atan}()$ 関数、平方根を求めてくれる $\text{sqrt}()$ 関数などのス額に関する関数を多用しました。また、全てのサーボモーターを同時に動かすため、すべてのサーボモーターを一度ずつ動かしていくという一般的なプログラムを書きました。

以下はサーボを動かすプログラムの一部抜粋です。

```
#include <Wire.h> //Arduino 既存のライブラリー
#include <Adafruit_PWMServoDriver.h> //ドライバーのライブラリー
Adafruit_PWMServoDriver pwm = Adafruit_PWMServoDriver();

#define SERVOMIN 150 // this is the 'minimum' pulse length count (out of 4096)
#define SERVOMAX 600 // this is the 'maximum' pulse length count (out of 4096)
void setup() {
  pwm.begin();
  pwm.setPWMPFreq(60); // 周波数の設定
  delay(10);
}
int p0d;
void loop(){
  p0d=servo(p0d,90,0);//(初期値、目的地、ピン番号)
}
```

```

int plus(int psd,int de) { //現在のサーボの位置と目的地の位置の大小で処理を分けています。
    if (psd < de) {
        return 1;
    } else if (psd > de) {
        return -1;
    } else {
        return 0;
    }
}

int servo(int p00d, int p00de, int pin1) {
    p00d += plus(p00d, p00de);
    servo_write(pin1, p00d);
    return p00d;
}

void servo_write(int pin3, int angle) { //実際にサーボを動かすプログラムです
    angle2 = map(angle, 0, 180, SERVOMIN, SERVOMAX);
    pwm.setPWM("pin3", 0, angle2);
    delay(3);
}

```

感想

初めての二足歩行ロボットの制作でしたが、3Dプリンターが故障することがよくあったので、思ったように制作を進めることができず予想以上に時間がかかってしまいました。それでも、毎回3Dプリンターの故障という予期せぬ事態に対応できたので、常に先輩の下にいた去年より成長できたと思います。

ダンゴムシの逆襲

製作 H2 深澤 孝光
M3 高角 怜志
協力 物無の皆さま

ストーリー

研究员中村「ふっふっふ、ついに究極のダンゴムシを作ることに成功したぞ！」

研究员海野「わーい」

ダンゴムシ「…」

と、その時、突如ガラスの割れる音が聞こえた。

海野「中村、ダンゴムシがないぞ！！！」

中村「それはまずい、今すぐ探し出せ！！！」

ダンゴムシ「ニンゲン、ユルサナイ」

こうしてダンゴムシの逆襲が始まった。

外観

※画像はイメージです



本体

1.動かし方

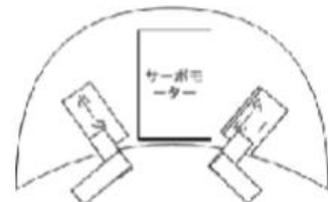
左下のジョイスティックを上下で移動、左右で向き変更、右側のスイッチで丸まったり元に戻ったりします。

四角い奴は通信用→



2.機構

一個一個の節は右図のようになっていて、中央のサーボモーターという角度を調節できるモーターでダンゴムシが丸まったり開いたりする機構を、左右の DC モーターが移動する機構を担っています。



プログラム

今回は pic を使いました。DC モーターの動かし方は簡単なので割愛させていただきます。

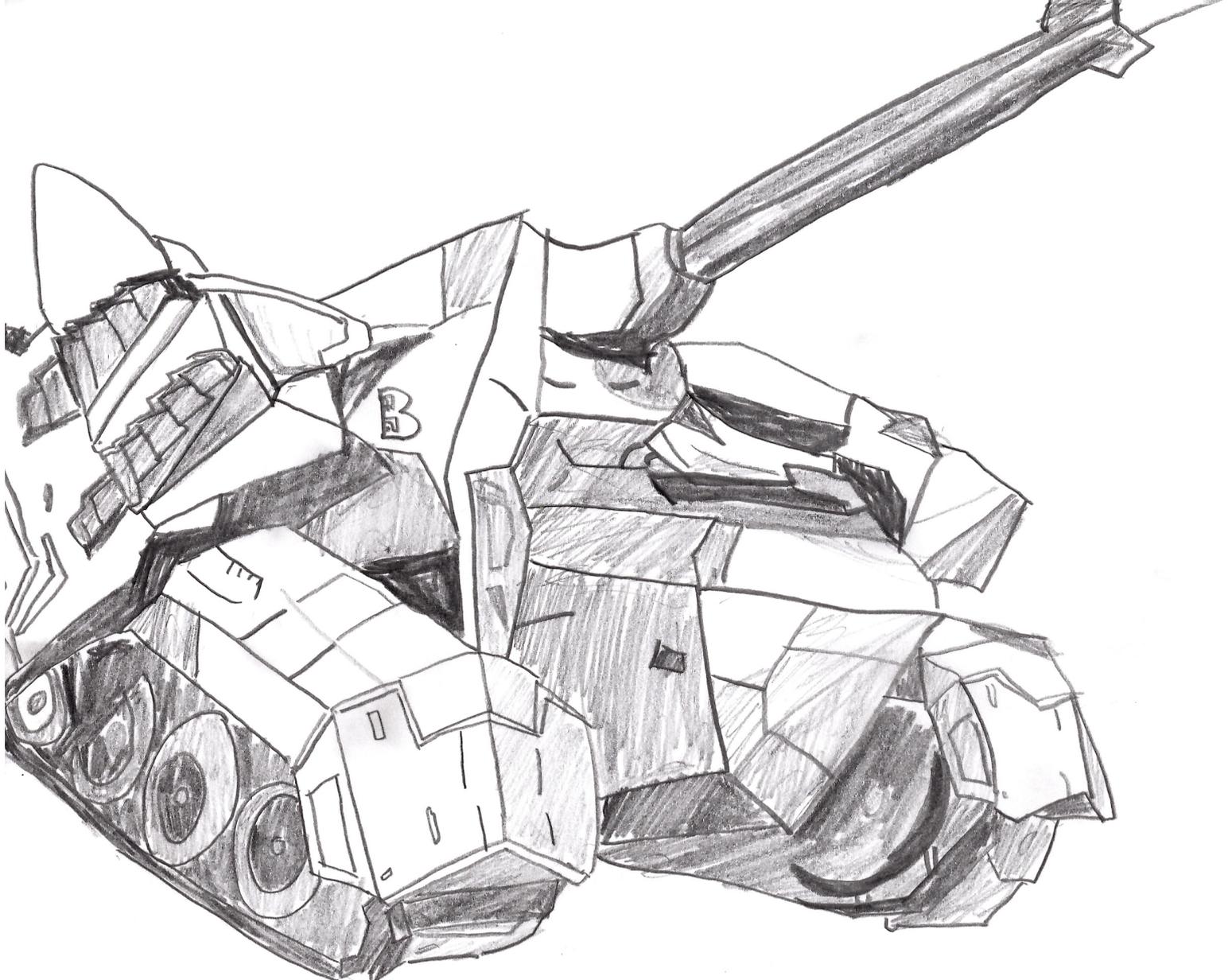
サーボモーターはパルス波で動くモーターですが、pic は直接パルス波を作れません。そこでタイマーという仕組みを使って動かします。一定時間 L を出力した後、変数で設定した時間 H を出すことでパルス波ができます。

無線の通信にはトワイライトを使いました。これが出す PWM は pic では読み取れないので、コンデンサーで GND とつなぐことでアナログに変換し、それを pic で読み取ります。

感想

深澤 サーボの操作と、本体の 3D モデル作りが大変でした。普通に歩くのではなく、丸まるというのは思いのほか難しい形になると知りました。

高角 初めての共同製作で、慣れないことや迷惑をかけてしまうこともありましたが、いろいろなことを学べて良かったと思います。



BTM game

Watch's adventures in wonderland

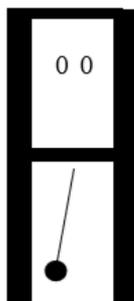
製作 H2 深澤 孝光
M3 高角 怜志
協力 物無の皆さま

ストーリー

「これは困ったぞ。遅刻だぞ。」
するとその益サギが、目の前でジャケットのポケットから振り子時計を出してながめ、急いで飛んでいきました。野リスはあまりの不思議に、益サギを追ってかけだしました。原っぱの向こうの端まで行くと益サギがちょうど大きな中サギ穴に飛び込むところでした。野リスも追いかけて飛び込むと……

(ウサギ穴だけは思いつかなかったので適当)

外観



時間は時間、分、秒が表記されます。
上にボタンが付いていて、それで時間の調節とアラームのセットができます。

概要

振り子の位置を、距離センサーを使って把握し、時計を進めます。
振り子を動かし続けるために、モーターの回転を摩擦を使って伝えています。
プログラムはただの60進法での時間カウントなので、割愛します。

感想

深澤 過去にあった24時間製作くらいの気持ちで作り始めましたが、配線に思いのほか時間がかかりました。

高角 これはネタではじめたのですがすぐ終わる制作物だと思ってたんですが、思いのほか時間がかかりました。

麻布へ GO !

製作者 高1 円藤 颯太
田中 基輝
協力 現高3の方を含む
物無の皆様

・ストーリー

E 藤 「俺よくよく考えたら自転車にもう3年近くおいてないんだよね。」

M 輝 「え、お前自転車に乗れないの、ケノ雑魚やんけwww。俺、毎日通学に自転車に乗ってるから、余裕だよwww。」

E 藤 (言い返したいが、もう俺には自転車はないし、本当にもう乗れないかもしれない。いったいどうすればいいんだ。)

??? 「この自転車を使わないかい？」

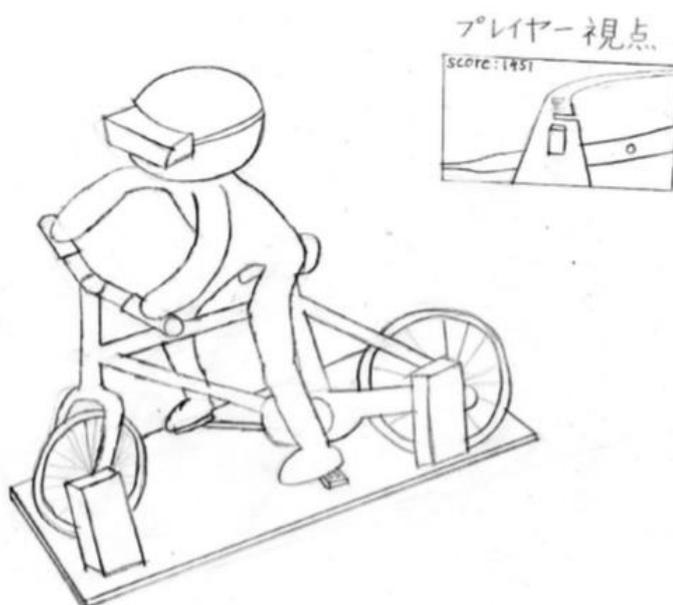
E 藤 「え、さっきまでこんなところに自転車なかったに、、、。いったいどなたか、でもこれで自転車に乗れるぞ。でも乗れなかったらM輝どころかY本にまで煽られる。どうすれば、、、。そうだなゲームで練習しよう！」

しかし、そこに待ち受ける脅威。

お願い、取らないでサドルを。ここを耐えれば、文化祭に出せるんだから！

次回「サドル 死す」

・外観



・概要

自転車をコントローラーとする、VRゲームです。自転車のペダルをこくと、ゲーム内でも自転車が動き出し、ハンドルを曲げるとそれに合わせてゲームでも曲がるようになっています。

その自転車を使って、ゲーム内に設置された障害物をよけながら、ゴールを目指すゲームです。自転車のハンドルのところに arduino Pro Micro と可変抵抗を取り付けてあり、その arduino Pro Micro はパソコンとつなぐとキーボードとして認識させられるので、それを使ってゲームを操作しています。そしてゲームのほうは、unity で制作しており、VR は、パソコンとスマホをつなぎ、スマホの角度を unity 側に伝え、そのゲーム画面を unity remote5 で映すということをしています。

< 概要の補足 >

ちょっとした arduino Pro Micro のところの説明をしたいと思います。正確に書くと arduino Pro Micro はパソコンとつなぐと、32u4マイコンによって USB 周辺装置として認識させられるのでプログラムの書き方によってはオリジナルのキーボードを作れます。そこで、今回は unity であらかじめパソコンのキーボードを使って遊べるゲームをつくり、パソコンのキーボードは使わず、arduino Pro Micro を使って操作しています。また、可変抵抗のつまみを回すと電圧が変えられるので、それを arduino で読み取り、その値に応じて信号を変えています。



< arduino Pro Micro の図 >

・感想

本来ならばもっと人がいたと思いますが、...でも無事に終わりそうなので少し安心しています。(3月17日現在)基揮君がいなかったらやばかったので、本当に感謝しています。今年こそは後輩と一緒にやりたいです。

イライラ ボウ

製作者

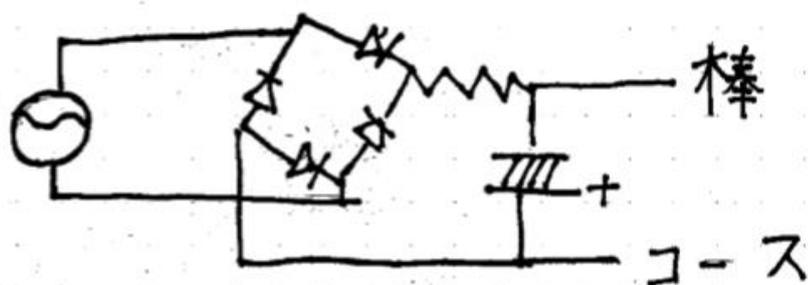
M2 山田恭平

協力 物無の皆様

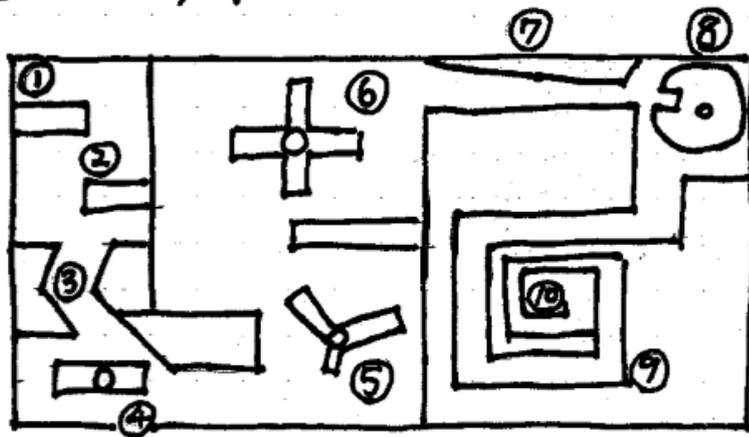
ストーリー

Y君はとある部活の帰り、道に迷って
しまった。様々な仕掛けを通過して家に
帰ろう。

回路図



コース



- ① スタート
- ② ジグザグ…うまくさけよう
- ③ くノ字の道…うまく曲がろう
- ④ 回転棒…ぶつからないようにしよう
- ⑤ Y字…うまくタイミングをはかろう
- ⑥ 十字…Y字よりおもしろいので気をつけよう
- ⑦ せぼまる道…ぶつからないようにしよう
- ⑧ 回転円盤…欠けている所に入ろう
- ⑨ グルグル…目がまわらないようにしよう
- ⑩ ゴール

感想 初めての一人製作でしたが先輩方に助けて
いただいたおかげで完動させることができました。
感謝の気持ちを忘れないようにしたいです。

ありふれたゲームで

世界最強

製作 M2 磯野大洋

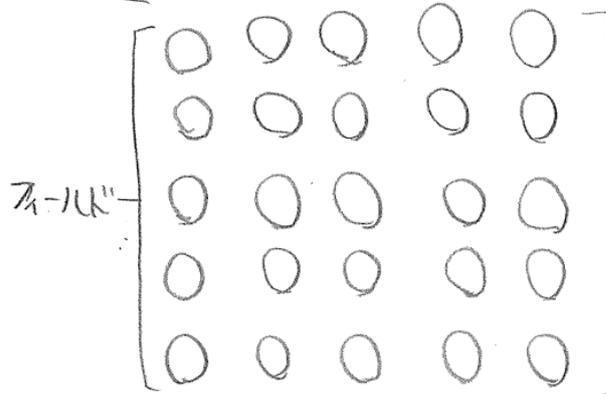
回路図設計 H1 基暉さん

協力 物無の皆さん

自機
操作



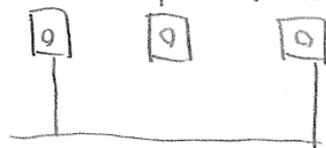
自機 [○ ○ ○ ○ ○]



自機 [○ ○ ○ ○ ○]

赤色発射

自機
操作



内容

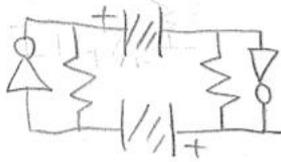
自機から弾を
発射して、相手に
あてましょう。
弾は相殺できます。

ストーリー

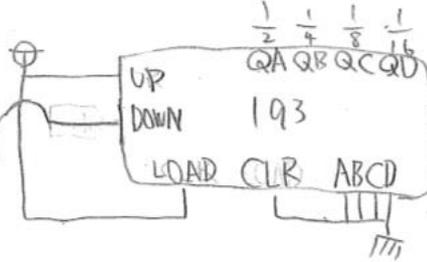
あ！
野生の麻布生が
勝負をしかけてきた！
生徒の投げた赤本を
よけながら、
あなたも3DSを
投げよう！

回路四

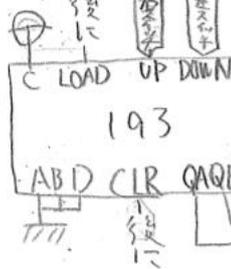
発振回路



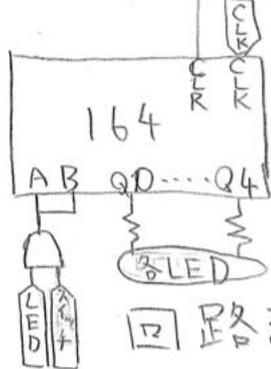
*発振を遅くする回路



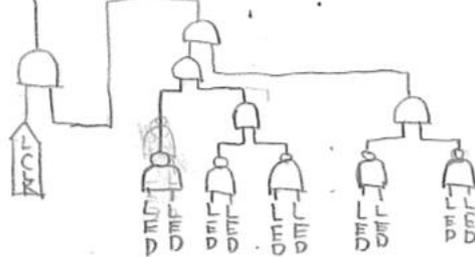
自機移動



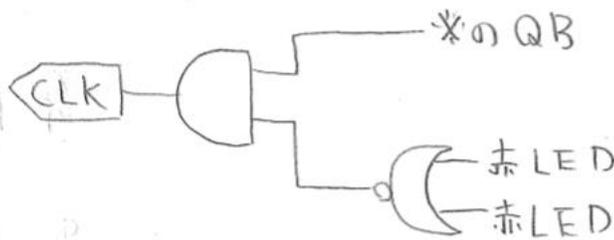
弾発射 X 5



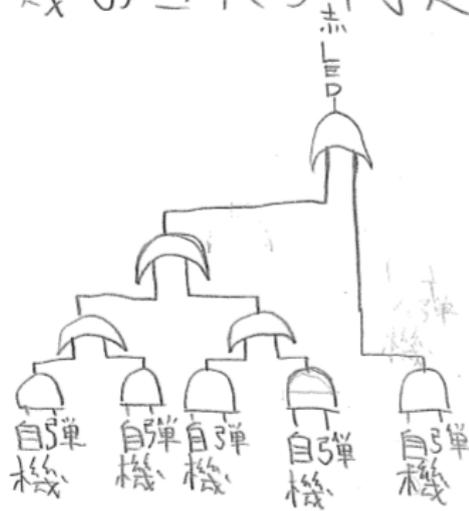
弾同士(の当たり)判定



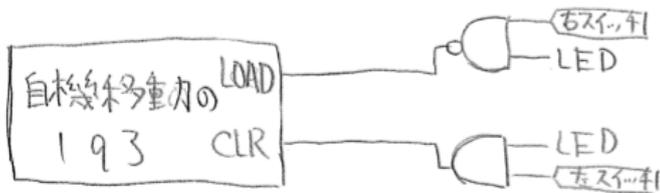
回路を止める回路



弾と自機の当たり判定



右端にある自機を右スイッチを押して左端にする



左端にある自機を左スイッチを押して右端にする。

感想

この製作をしたことで、入部した頃の頃よ!)
はんだ付けが上手くなったと思います。
また、ICのこともよく知れたと思います。

富山ラーメン屋の逆襲

製作 M12 谷中
 目録 四段村
 H1 音木さん
 二橋カ
 物産の管さま

ストーリー

時はさかのぼり、1940年、シベリア。今は七き者MとT中が
 何やら話をしているようだ。

T中「おなかすいてござ文死にそうナリ」

M「こういうときは富山ラーメンが食べたい」

その時、目の前にカツ丼屋と富山ラーメンの屋台が現れた

M「これは富山ラーメン行くしか無いよ…」

T中「カツ丼！カツ丼行こう」

目の前でスルーされて富山ラーメン屋の屋台主は激怒し、
 近くにあるその屋台が、逆襲を開始した。

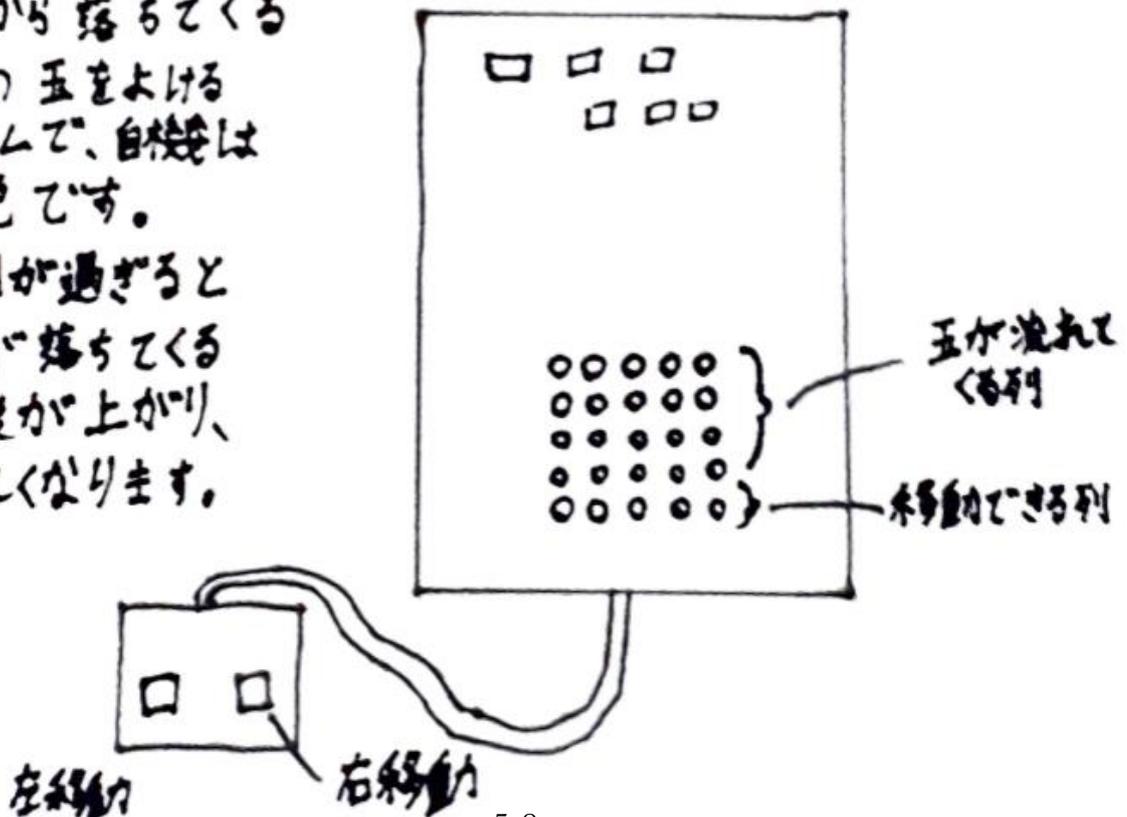
ルール

上から落ちてくる

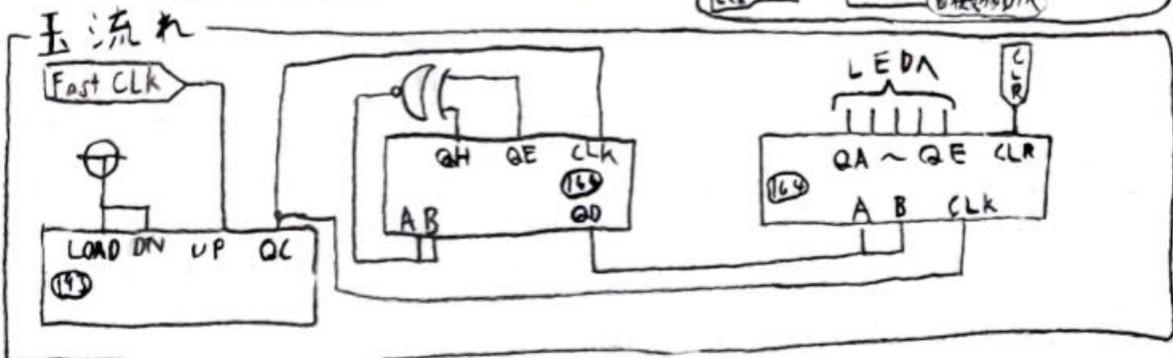
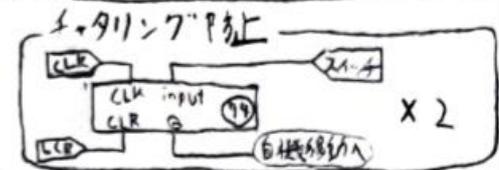
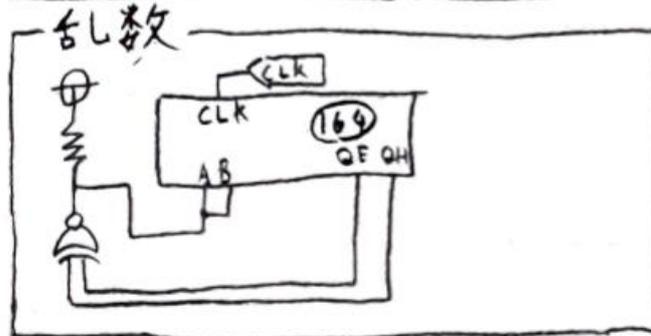
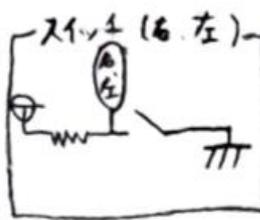
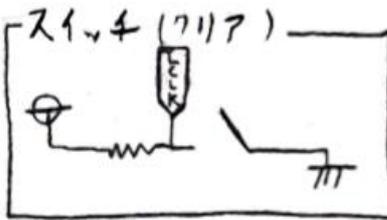
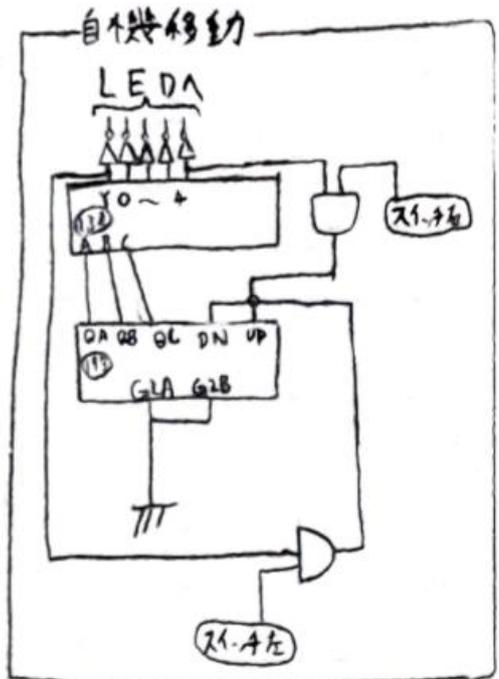
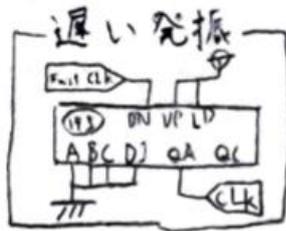
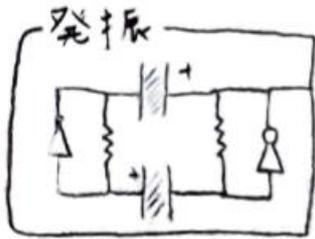
糸緑の玉をよける
 ゲームで、自機は
 赤色です。

時間が過ぎると
 玉が落ちてくる
 速度が上がり、
 難しくなります。

外観



回路図



感想

自分思っていたよりもバグや謎配線がたくさんありましたが、ゲートの種類や働きなどを覚えることができました。はんだ付けも上達したり、学ぶ所は多かったです。

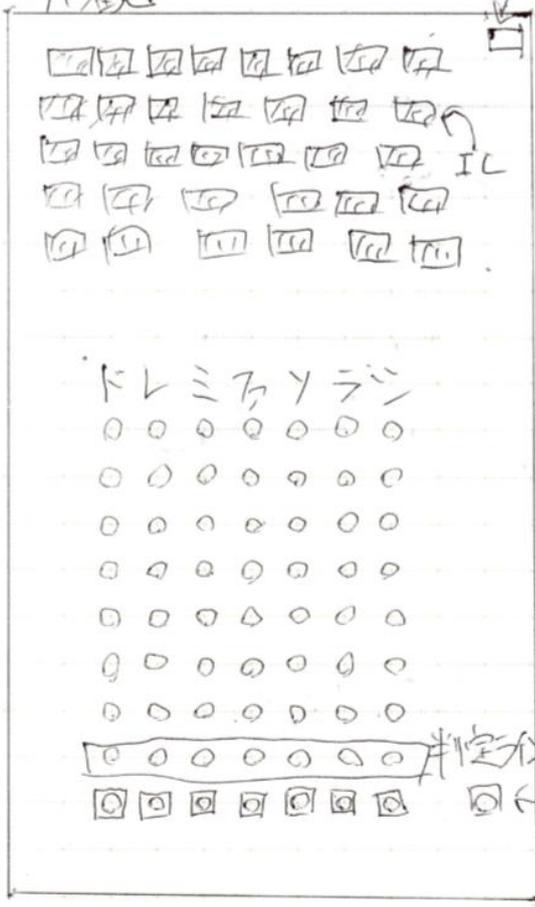
かませ!! 対空ミサイル!

M-2 中嶋
回路設計 田堂 坂 H-1

ストーリー

ある日戦争が起こり、物々が標的に決まってしまう。
 空中から降る爆弾を対空ミサイルでマイクよくかくれ。
 美しい爆音を奏しよう。

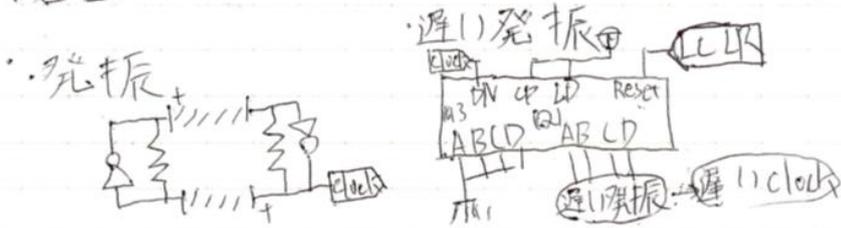
外観



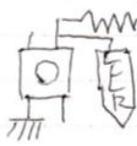
電源

このゲームは、光りながら降る
 LEDが判定ラインに落ちると
 下のボタンを押し、音を鳴らす
 というゲームです。

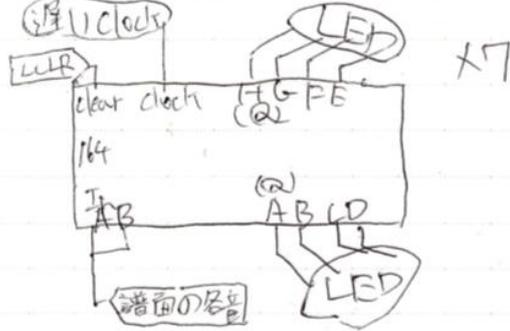
回路図



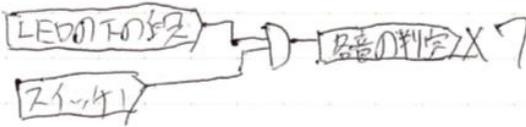
・ 74175 (LOW)



・ 玉流れ



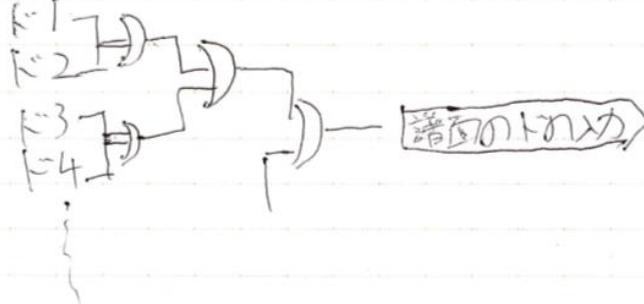
・ 当たり判定

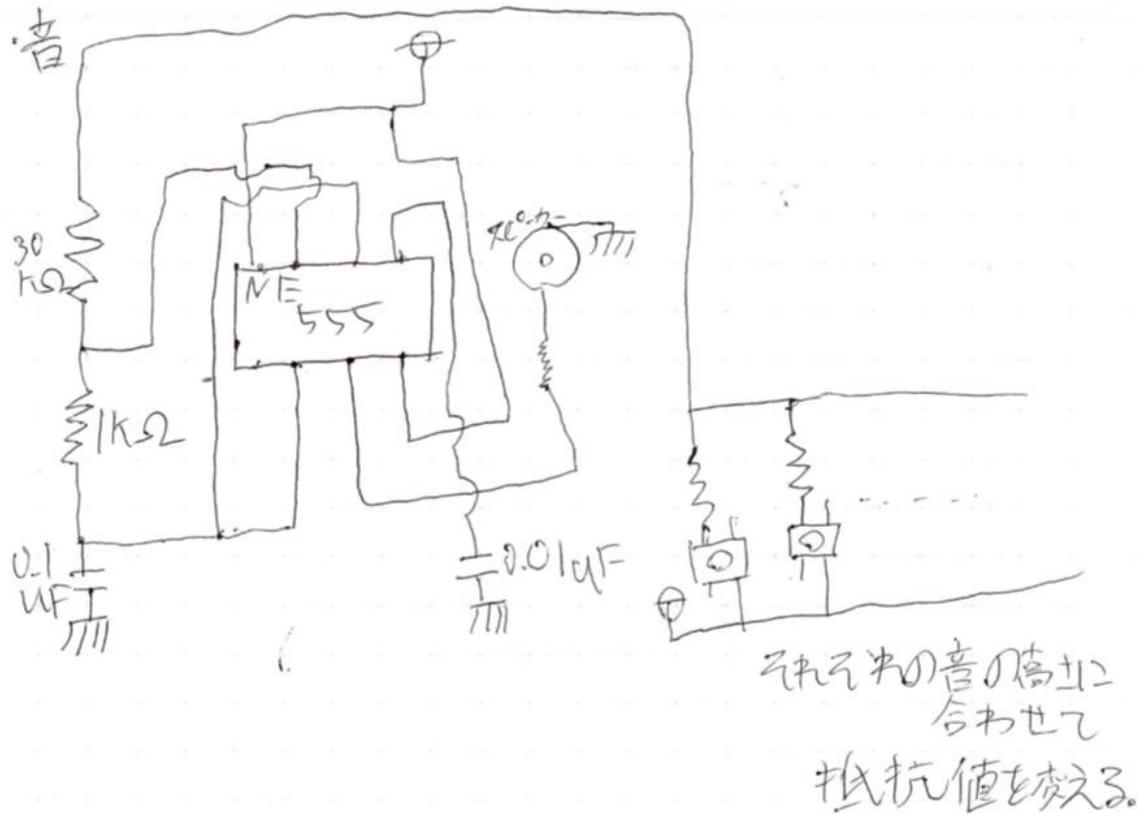


・ 譜面



・ 譜面入力





感想

この製作をしたことで、はんだ付けや配線などといった物作りの難しさを感じました。

ローリングウッドの 逆龍衣

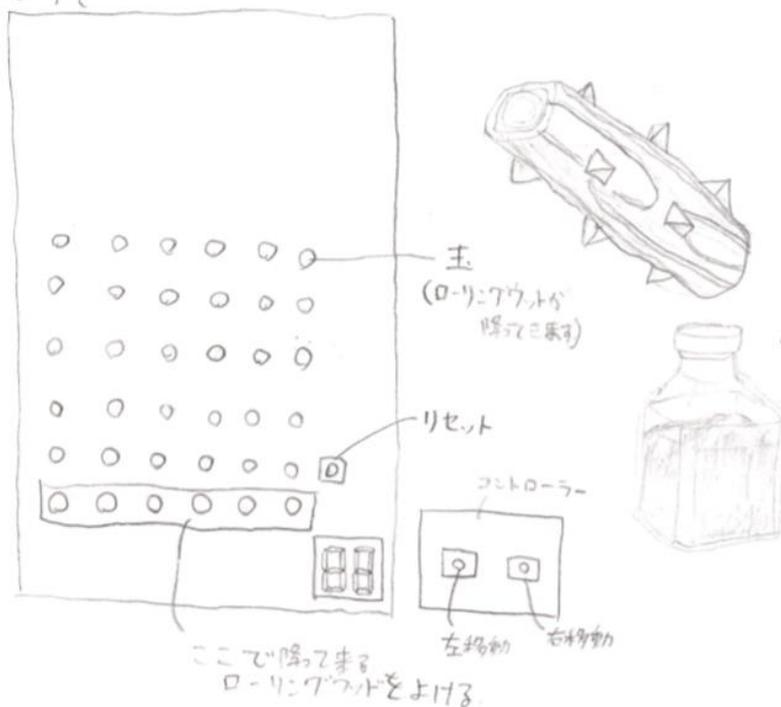
製作 M2 中澤凌太郎
設計者 H1 田才さん

1. ストーリー

あなたはランバージャック。いつものように木を切りに出かけて行った。しかし、何らかの物体につまづいてしまい、愛用の「レイジ」という呪文をこぼしてしまっ！なんとあなたはローリングウッドとなり、いっせいに襲いかかて来た！とにかく逃げよう。

2. 概要

上から降ってくる赤 LED (ローリングウッド) をとにかくよけましょう。制限時間が切れるまで逃げ切るとあなたの勝利です！



トレジャーハンター

制作者 M2 畠山
回路図設計者 H1 小野

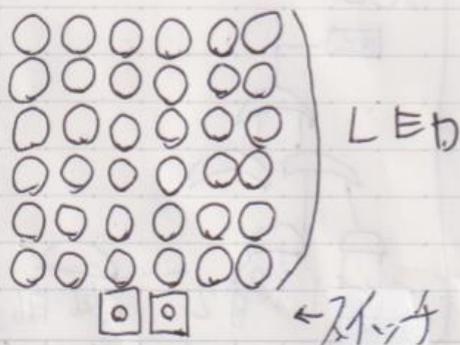
ルール

6×6のLEDがあり、上から青い光と赤い光が降りてくる。赤の光にあたらないように、そして青の光にあたるように左右のスイッチで緑の光の自機を操作する。

ストーリー

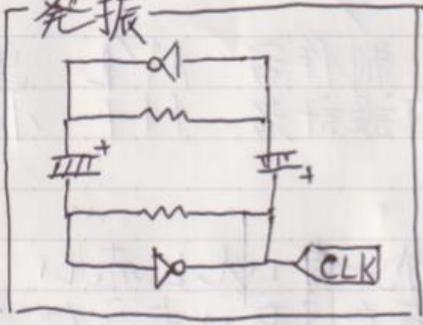
君は宝物を集めるトレジャーハンターだ。君は宝物がねむっているところをさがしているとき、ある石場にやってきた。と、その時ゴゴゴという音とともにたくさんの岩と宝物が落ちてきた。どうやら上の方の岩場で崖くずれがおこったらしい。そしてそこにあった宝も一緒に落ちてきたのだ。君は危険をかえりみず、宝を集めることにした。

外観



回路図

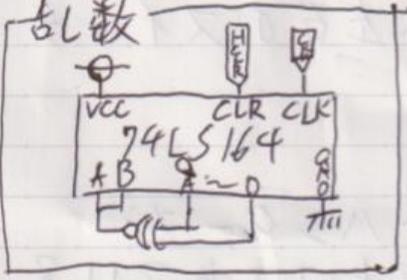
発振



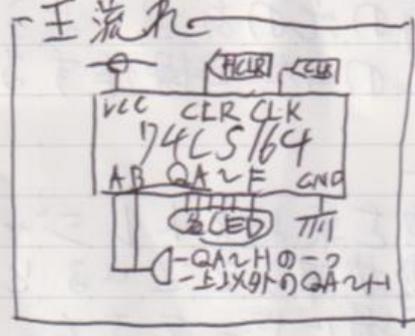
発振遅延停止



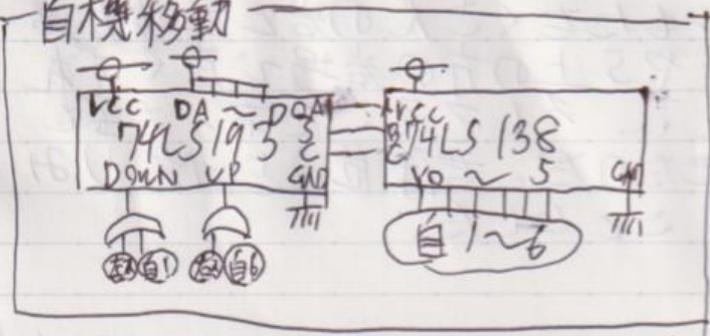
乱数



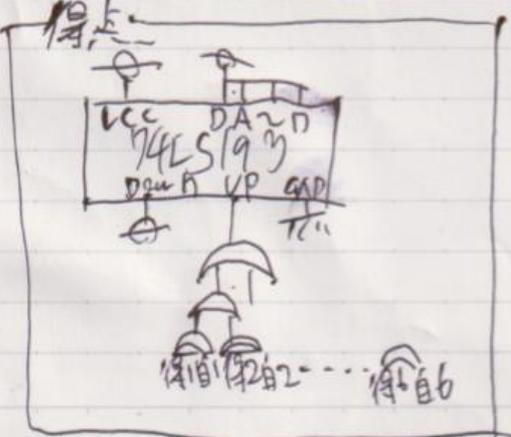
王流丸



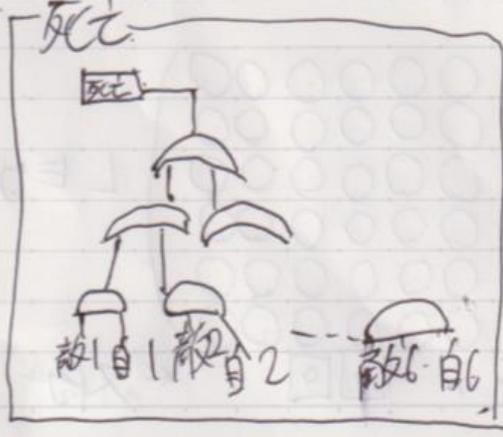
自機移動



得点



反転

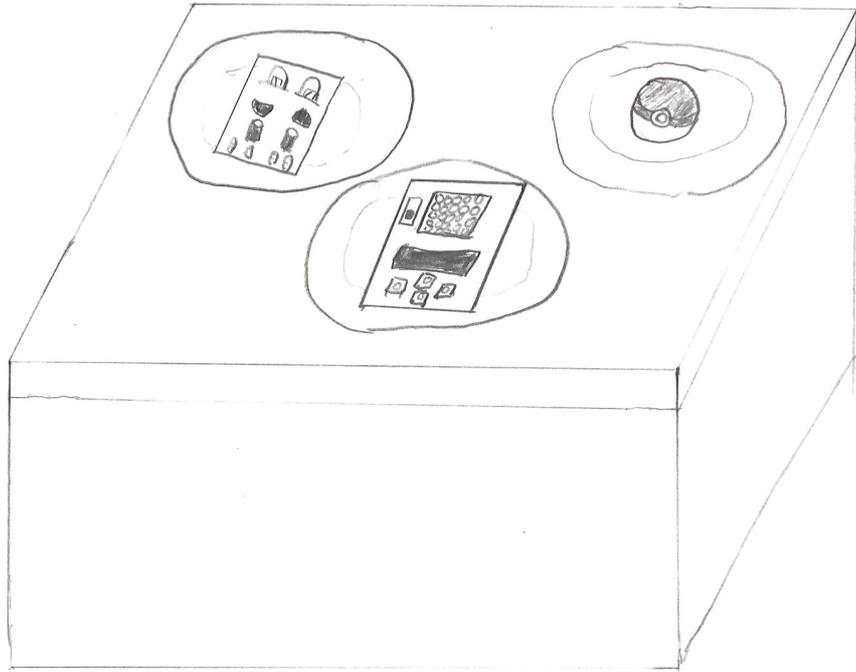


チカチカ

&

売り物

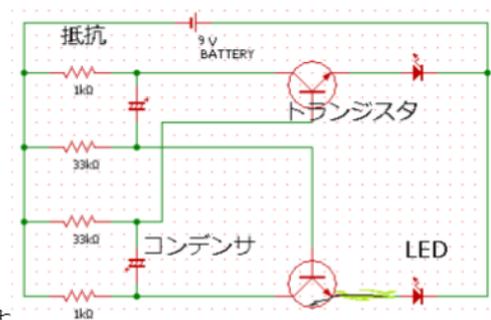
「3つのうち1つえらべ!!」



チカチカ

M3 池田 宜史

チカチカはチカチカ教室で実際につくれる回路で仕組みは簡単です。正式名称は「自走マルチバイブレータ」です。



これが回路です

なぜチカチカするか説明します

- 1 電気は抵抗の少ないほうにながれる性質があるので $1\text{ k}\Omega$ (一番上と一番下) の方の抵抗にながれます。
- 2 電気をため、たまりきると一気に流す性質があるコンデンサーに電気がたまります。
- 3 電気で押すスイッチようなせいしつのあるトランジスタに、コンデンサーでたまり切った電気が流れます。
- 4 LED に電気が流れ光がつきます。
- 5 コンデンサーが空になり 1 に戻ります。

ポケトGAMES

製作者 中村陽 協力 物無の皆様

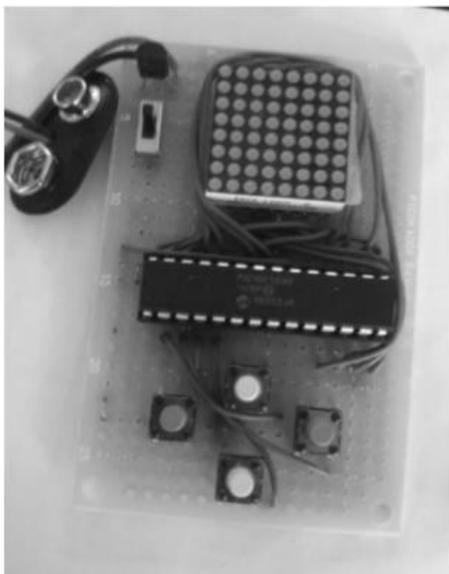
ストーリー

やめて！英語のテストで、赤点を取ったら、闇の契約で売り物を作る個数を1ケタ増やすことになってる中村の精神まで燃え尽きちゃう！

お願い、赤点を取らないで中村！あんたが今ここで補修になったら、英語の塾の先生との約束はどうなっちゃうの？時間はまだ残ってる。ここを耐えれば、英語のテストで赤点回避をできるんだから！

次回、「中村死す」。デュエルスタンバイ！（このあと本当に赤点を取りました）

外観



部品の説明

ドットマトリックス・・・上についてるやつです。8×8のLEDが付いています。

三端子レギュレータ・・・右上のやつです。電圧を下げる働きがあります。

PIC16F1938・・・真ん中についてるやつです。このゲームのプログラミングがされています。

スライドスイッチ・・・三端子レギュの下のやつです。ゲームのON/OFFを切り替えます。

タクトスイッチ・・・下の4つです。ゲームの操作ができます。

