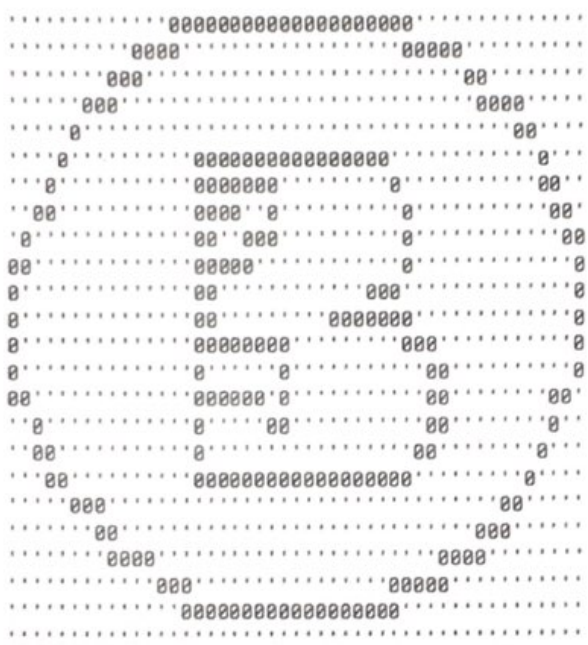
Robotfactory

カイロズシュウ2020

ブツリブムセンハン

```
>>welcome to robotfactory
¥¥ system wake
>>booting....
|------------|100%
>>Done.
¥¥load bookinfo
>>ｶｲﾛｽﾞｼｭｳ2020
  author:ﾌﾞﾂﾘﾌﾞﾑｾﾝﾊﾝ
¥¥open
>>loading...
```



```
>>Done.
¥¥start reading
>>Loading chapters....
   Robot...     loaded.
   Game...      loaded.
   Computer...  loaded.
   Analog...    loaded.
>>Finished.
>>Processing...Done.
>>Are you ready?
```

はじめに

昨年の回路図集では
「米国の大手通信業者ベライゾン・ワイヤレスが家庭向け5Gサービスを開始」
とありましたが最近では、
米インテルが2020年9月2日に「Tiger Lake」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ており、スレッド性能やグラフィック性能がさらに改善され、さらなる躍進をとげています。
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、99種99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1年かけて研究、製作したものについての記録です。未完成のものも含まれており、本書の通りに作っても動作する保証はございません。あらかじめ御了承下さい。

H2会計 田中基暉

目次

はじめに p3

ロボット系 p6

カマキリ H1 海野 H1 高角 H1 徳田 p7

六足 M3 山田 H1 中村 H2 斎木 p10

Bluetooth H1 中村 p12

戦車 M3 中澤 M3 谷中 H1 吉田 H2 杉田 p13

蛇 M3 礒野 M3 山下 H2 田中 p15

アナログ系 p20

テスラコイル H2 小野 p21

コンピューター系 p26

Azabac71 M2 和田 M3 畠山 H2 宮田 p2 7

ゲーム系 p35

OTAGE M3 中嶋 p36

Bututendou Labo M3 加藤 M3 高梨 H1 池田 H2 円藤 p38

チカチカ売り物 p41

売り物 M3 礒野 p42

角材避け in 物無 M2 廣瀬 p46

IC 拾い M2 山田 p48

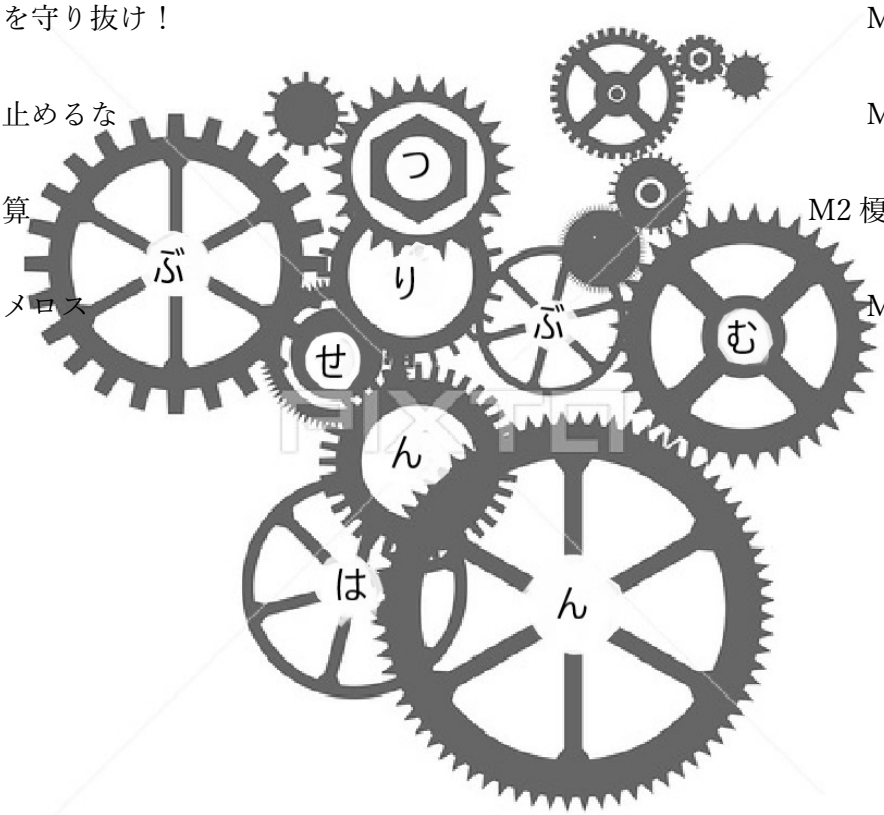
Escape from the eyes M2 永谷 p52

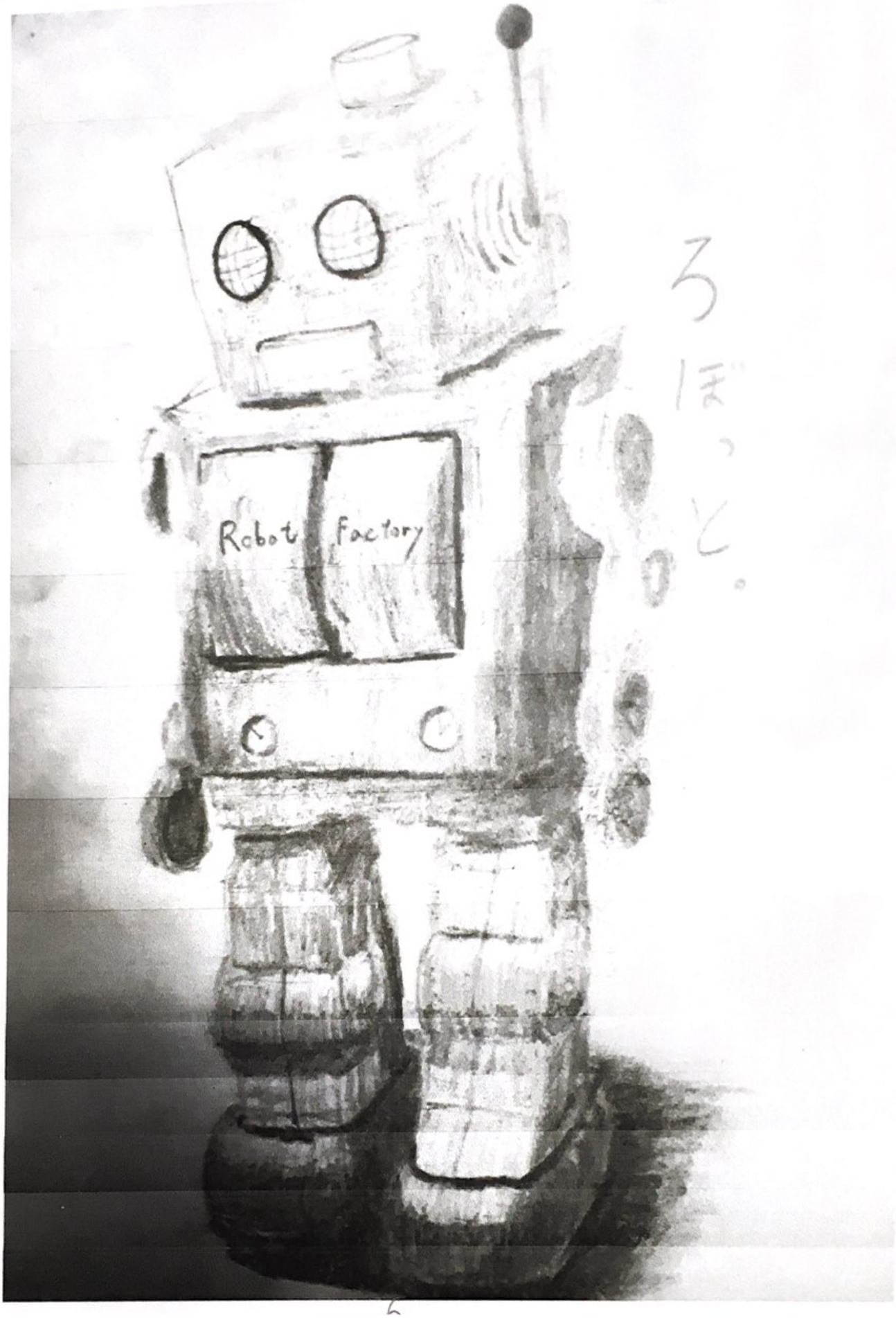
制作物を守り抜け！ M2 大谷 p54

動きを止めるな M2 鳥内 p56

マルス算 M2 榎本大智 p58

避けろメロス M2 秋田 p60





3ぼっと。

鎌切

[製作]H1　高角　海野　徳田　[協力]物無の皆様

・ストーリー

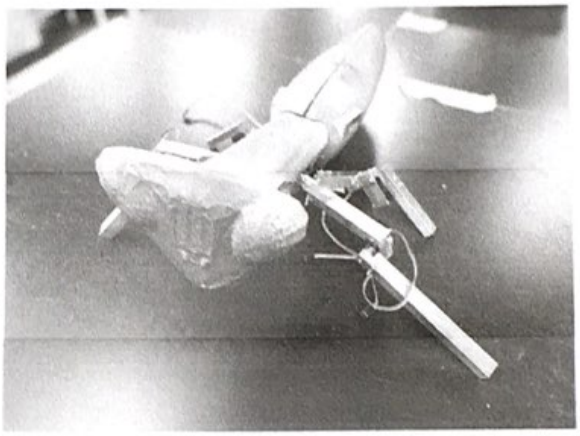
2019年4月6日麻布は、謎の火事に見舞われた。幸いにもこの事件での負傷者は0人だったがしかしその原因は未だ解明されず多くの憶測が飛び交っている。だが、その憶測の全てにおいて共通しているものがあった。それは、謎の鎌を持った生物が出てきていることだ。一番有力な説では、この生物はカマキリなのではないかとされている。又、この勇敢な1匹のカマキリがこの事件での負傷者0人に大きく貢献していたらしい。我々は、このカマキリに大きな感銘を受け、これを象徴するロボットを作ることにしたのだった...

・概要

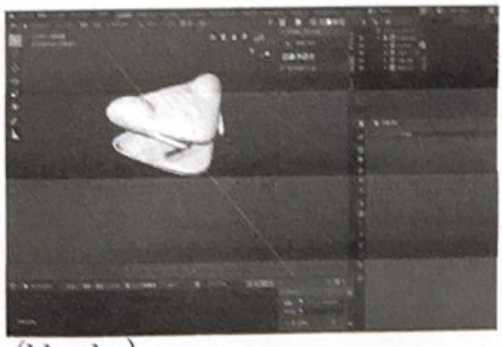
カマキリのロボットです。無線コントローラーで移動させたり、鎌をうごかしたりできます。かっこいいです。

・外観

見た目はこんな感じです。首がないですね。
足が細いので、バランスが取れるようにするようにするために腹に足をつけてしまいました。昆虫博士の人がいたらごめんなさい。



今回は、3dプリンターで印刷する用のモデル製作を二つのソフトを使って行いました。体などの、サーボモーターを入れるなど正確性が求められる部分はfusion360というソフトを、顔などの作り込みをするのにはblenderというソフトを使っています。


(blender)

(fusion360)

・機能

鎌以外の四本の足で歩いたり、羽を広げたりすることができます。その他に超音波距離センサによって正面の物体との距離を測り、物体が近くに来ると鎌で捕まえようとしたり、モードを変えると踊ったりします。この距離はコントローラーに付いているLEDで表されます。

・仕組み

本体

seeeduinoというarduinoの互換機によって操作されています。足や羽に用いられているサーボモータはpca9685というサーボドライバによってseeeduinon単体で操作可能であるサーボモータの数よりも多くを少ないピン数で操作しています。これは外部電源を必要とするため、モバイルバッテリーを本体に積んでいます。

また、超音波センサもseeeduinoによって操作されコントローラにそれから得られた情報を送っています。

コントローラとの通信はTWILITEによって行われています。

コントローラ

TWILITEによって本体を操作しています。本体から受信した情報をもとにPICによってカマキリとその正面いある物体との距離をLEDに表示させています。

・感想

海野

コントローラと本体の中の基板とプログラミングを担当しました。本体がなかなかできなくて暇でした。UNNNO って打つと海野って出てきます。

中村

この政策にはあまり関われませんでしたが、楽しっかたです。明日休みなのもとても嬉しいです。文化祭絶対勝つぞー。おー。

高角

本体を作りました。モデリングの技術を上達できてよかったです。

顔を凝れてよかったです。

徳田

初めてのモデリングで、本体の印刷が遅くなり迷惑をかけてしまいました。すみません。

六足歩行ロボット

Ｈ２齋木　Ｈ１中村　Ｍ３山田

協力　物無の皆様

・ストーリー

麻布学園には、人数、活動内容さまざまな部活が存在する。その中でも、全部活最大級の人数を擁する生物部と、高い技術力を持つ物理部無線班は、同じ理科系ということもあり、激しく対立していた。生物部からすれば、物無の技術は文化祭で大きな脅威になる。そこで生物部部長は、物無員を殲滅するべく、最強のスパイ・Ｓを送り込んだ……

これは、ある日の物無での出来事である。

Ｓ「（どうやってこの部活を潰すか…。こんなパソコンより重い物を持ったことがないような部員ども、俺の戦闘力をもってすれば全員瞬殺できるんだが、それでは俺だとバレてしまう…。お、あれは後輩のＮじゃないか）Ｎくん，英語の補修は終わったのかい？」

Ｎ「Ｓさ～ん、僕、そんなことより六足歩行のロボットを作りたいんですよ～」

Ｓ「六足かぁ…（そうだ、生物部の俺の知識で、ロボットと偽って最強の昆虫を作らせて、そいつを暴れさせよう！）それ、楽しそうだね！一緒にやろうよ！（あとは、もうひとり優秀な部員が欲しいな。そうだ、Ｔ緑会に入っている部員Ｙなら優秀だ！しかも、まだ小学生だから簡単に騙せる。彼を入れよう！）」

Ｓ「Ｙくん、一緒にロボットを作らないか？」

Ｙ「Ｓさん！僕、やってみたいです！」

こうして、二人の部員は、何も知らずに恐ろしい生物を作り始めるのであった……

・概要

六足歩行のロボットです。六本の脚と、それを動かすモーターで、自由に動き回ることが出来ます。また、今回はスマートフォンから操作できるようにしました。

・仕組み

そのものはシンプルです。脚一本に三つモーターを使用し、それを六本組み合わせ、スマートフォンからおくった信号からマイコンで制御し、歩行します。

今回の六足では、特殊なサーボモーターを使用しています。まずサーボモーターとは、動きが遅い代わりに、動く角度を指定することができ、力がとても大きいのが特徴です。そのため、ロボットの関節によく使われています。しかしながら六足では、通常の市販のサーボモーターでは、パワーが圧倒的に足りません。そこで、強力なサーボモーターを使用した、完成品として売っている二足歩行ロボットが安売りされていたので、それを購入し、解体してサーボモーターを入手するという方法を採りました。ＩＣＳサーボという、性能がとても高い代わりに、制御方法が特殊なモーターなので、制御に手間がかかります。制御は、Arduino というマイコンを使用しました。

本体は、３Ｄプリンターでフレームを作成しています。ＰＣ上の３Ｄソフトでデータを作成し、プリンターで印刷しますが、サーボモーターに正確に装着させるため、精度の高い物が要求されます。

・通信は、Androidスマートフォン向けのアプリを開発し、TWILITEというモジュールで行います。（この部分はまた別の章で詳しく解説しています）

感想

完成品を解体して入手したモーターだったので、制御方法を見つけるのに非常に時間がかかってしまい、遅延の大きな原因になってしまいました。しかし、部内で初めてのことなので、自分で全て調べて、0から構築していくのは大きな達成感がありました。まだまだ道半ばですが、やり切ります。（齋木）

携帯する六足の運命

製作　H1　中村陽
協力　H2　斉木さん
　　M3　山田
　　物無の皆様

ストーリー

Nは知っていた…生物部から最強のスパイ・Sが送り込まれていたことを。

N「Sさ～ん、今日は何をしますか？(今作っているこの最強の生物…どうにかしないと暴走してやがてこの部活が滅ぼされてしまう…。)」

S「…。」返事がない。無線のヘッドホンを着けていて周りの音が聞こえないようだ。

N「(そうだ！この生物の頭を無線で支配して僕のスマホから思い通りに操作できるようにしてやろう！)」

かくして孤独の男Nと最強スパイSにより部活の存亡を賭けた静かな戦いが始まったのであった。

概要

androidスマートフォン用アプリです。
スマホからBluetoothモジュール(RN-42)をつけたarduinoに文字列を送り、無線でロボットを操作できるコントローラーです。

←(RN-42)
arduinoとシリアル通信をすることにより文字列を送ることを可能にしました。
Bluetoothを使っているので教室の範囲程度ならどこからでも安定して通信することができます。

・スマホ側の説明
アプリ開発にはAndroid StudioというGoogleが提供しているAndroid端末アプリ用のものを使いました。スマホに出来ることなら大体のことができるので自由度が高いです。アプリには近くのBluetoothデバイス一覧を表示・接続させる機能、画面を遷移させコントローラー画面にし、前後左右の指示をarduinoに送る機能を実装しています。

感想

何か新しいことをしてみたいと思い、スマホのアプリを作りました。過去にスマホを使った製作はありましたがBluetoothを使うのは初めてだと思います。割といい感じにつくれたと思います。部員の皆さんにもアプリを作ることをオススメします。あと、この製作物名は僕の好きな漫画に出てくる能力名からとってます。

㊙機動戦闘車開発計画

リーダー　H2　杉田さん　　無線担当　H1　吉田さん
プログラム担当　M3　中沢　　CAD 担当　M3　谷中

（1）ストーリー

時は令和、世界は第三次世界大戦によって荒廃し、一面の荒れ地には生物兵器「BucHiKu」達が蔓延っていた。
地下シェルターに避難していた平凡な中学生、Ｔ中とＮ沢は同シェルター内のＹ田の死に際にある場所への地図を渡され、
地図の示す場所に赴くと、古びたラボがあった。そこできれいな形を保った USB と、「修行に出ます　Ｓ田博士」
と書かれたメモを見つける。そして USB 内には謎のプログラムと超かっこいい戦闘車のモデリングがあった。
彼らはＳ田博士のデータを基に対生物用機動戦闘車を作り始める。
これはのちに人類を救った英雄と称えられる２人の物語である...

（2）概要

主砲には１８３㎜砲を搭載、時速６０km/ｈでの走行が可能。完全無人走行を可能とし、強力な主砲は一撃で生物兵器を粉砕する。単三電池で動いている超かっこいい機動戦闘車。排気ガス 0 で荒廃した地球にも優しい

（ここから制作裏話）

簡単に言えば車の上に砲塔をのっけて、大砲を打てるようにしたものです。車体も砲塔もほぼすべて３Ｄプリンタ製の超かっこいい戦闘車君で、モデルは日本の１６式機動戦闘車、そしてイタリアのチェンタウロです。もっと砲塔をちっさくスリムにしたかったですが銃の大きさのため砲塔がクソでかになりました。口径といい某紅茶の国製かな？
（正面からさえ見なければ）超かっこいいのができてる気がします。SD プリンタ万歳！

（3）使用した部品

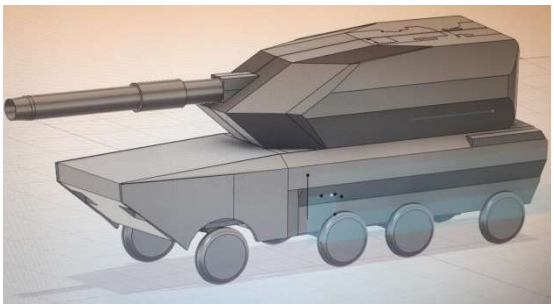
・PIC16F1938（マイコン）	制御部品の一つです。コントローラーから贈られた信号に対して、プログラム通りの動きをします。
・Arduino（あるでぃーの）	PIC と同じく、プログラムで動作する部品で、本製作における心臓部です。後述のモーターたちを動作させる信号を送ります。
・モータードライバ	Arduino からの信号に対してモーターの回転を制御させる部品です。
・ステッピングモーター	回転量、速度を自由に制御できるモーターです。Arduino を使わなくても動くことができます。本製作では Arduino によって動かしています。
・サーボモーター	0°～180°の範囲を自由に回転可能なモーターです。
・Twe-lite（トワイライト）	二つセットで使われており、一方はコントローラーに、もう一方は車体についています。主に信号を無線で送信・受信し制御部品に送る部品です。
・スイッチ	押したら回路がつながって信号が流れる。その単純さゆえに出番が多い部品です。
・ジョイスティック	すごそうな見た目をしているのに実はスイッチ 4 つでできている部品です。本製作では制御を司っています。

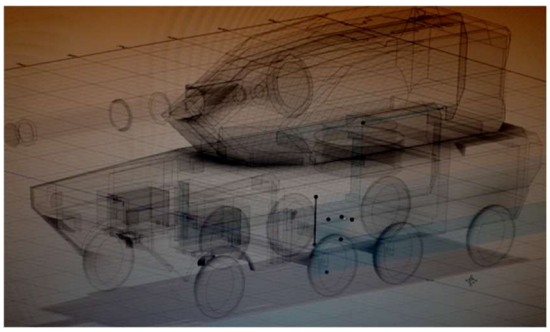
（4）原理

～車体～

前輪がサーボモーターによって左右に曲がり、後輪はギアボックスにより前進・後退します。真ん中の二対のタイヤは姿勢安定用の予備輪です。

前面のくぼみにはライトをつける予定です（時間が余れば）





～砲塔回転部～

砲塔はステッピングモーターでギアを回し、

砲塔にあるギアと噛み合わせて回転することができます。

～砲塔内部～

内部にはもともとコイルガンなりなんなりを載せる予定でしたが、

電力と技術の関係上断念し、100 均 BB 弾銃を使用しています。

引き金をサーボで引きます。後部には電池ボックス君が乗ります。

～制御～

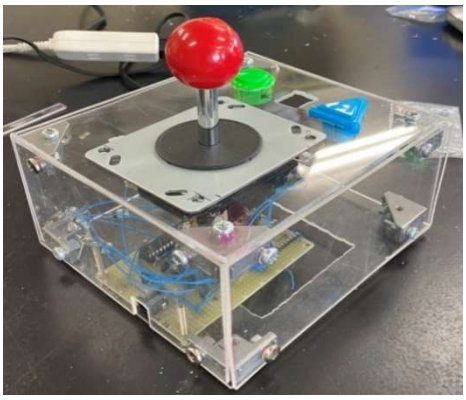
twe-lite を使った無線での制御をしています。

twe-lite のデジタル入力受付可能なピンが 4 つしかないので

ジョイスティックとボタン合わせて 11 種類の入力ができないところを、

PIC をプログラミングして回路に組み込むことで 4 桁の 2 進数に

変換し、無線通信による制御を可能にしています。



（5）感想

最後までいることができなく大変申し訳なく思っていますが、とても個性的なメンバーと製作することができて楽しかったです。（杉田さん）

自分の技術力の足りなさで共同製作者には迷惑をかけました。これを書いている時にはまだ完全動していませんが、ここまで製作でき、良い後輩がついてきてくれたことには感謝しかないです。（吉田さん）

初めての共同製作でしたが、先輩の方々のサポートもあり、共に製作できたことをうれしく思います。マイコンに関する知識も増え、僕にとって有意義な製作でした。共同製作者の皆様本当にありがとうございました。（中沢）

僕が皆さんを誘ったことで始まったこの製作ですが、理想と現実の違いをとても実感させられました。

僕は 3D CAD によるモデリングをしましたが、プログラミングを中沢君に丸投げした感じがあってとても申し訳なく思っています。現在まだ完動はしていませんが、文化祭に向け悔いなく完動させたいです。（谷中）

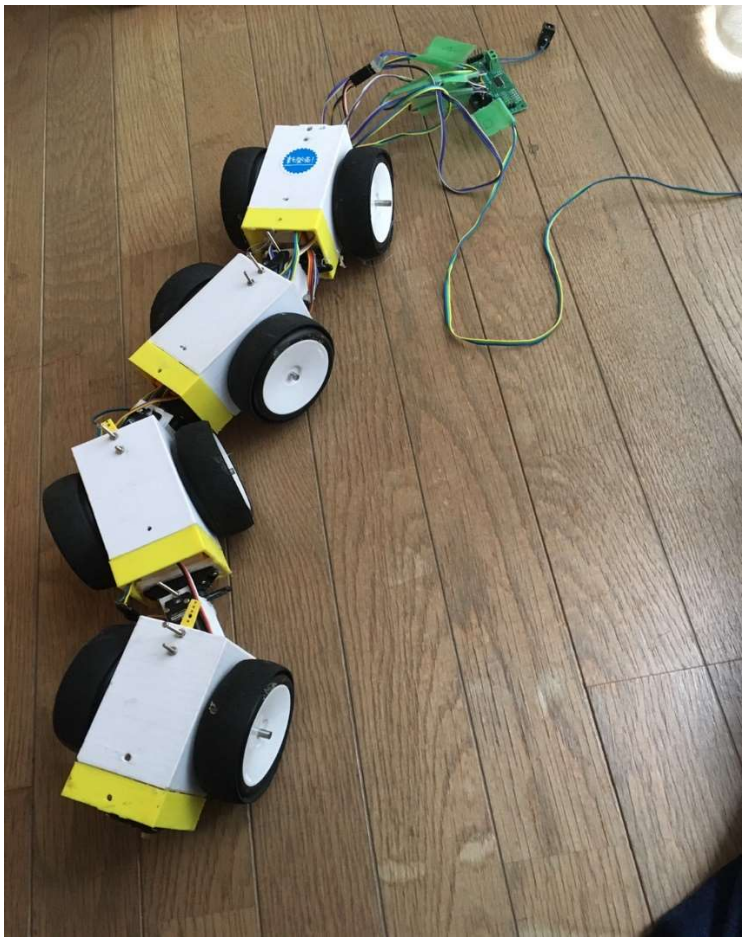
Ultimate Curve

制作者　中３　磯野、山下　高２　田中
協力　物無の皆様

ストーリ

人間は二足歩行、シマウマは四足歩行、クモは八足歩行。
一輪車は一輪、自転車は二輪、自動車は四輪。
では、蛇はどうか。蛇には手も足もない。
ではどうやって前に進んでいるのか。
その理由を探るべく我々はアマゾンの奥地に向かった。
そこで我々が目にしたものとは、、、
多くのタイヤとサーボモーターと、ひとつのラズベリーだった。

概要



スマホで操作できる蛇型ロボットです。

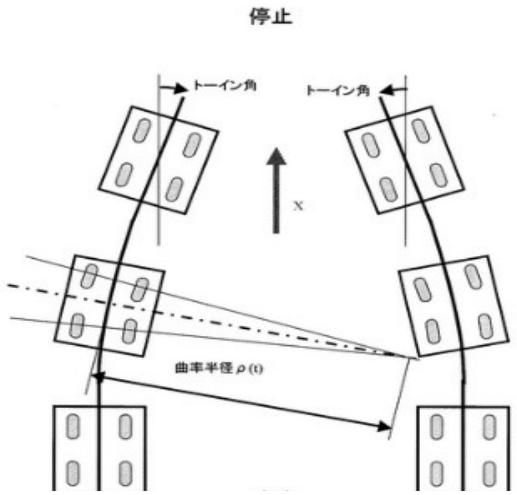
部品

raspberry pi 3b+　　PCA9685・・・サーボモーター用のドライバーです
micro servo 2bbmg

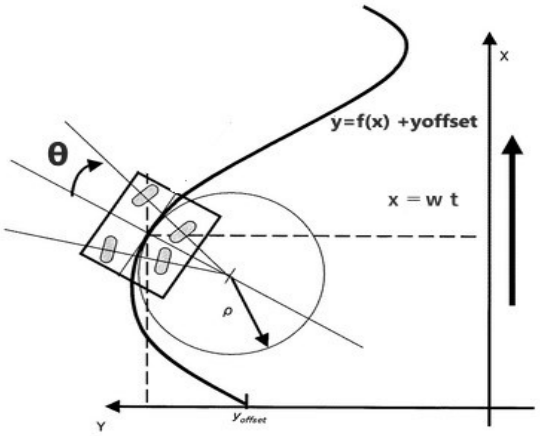
原理

１　グライド推進

ここではロボットの制御の話の前に、まず蛇の推進方法（グライド推進）について簡単に説明します。グライド推進とは簡潔に言うと、進行方向と進行方向に直行する方向の摩擦係数の差によって、推進力を得るというものです。また蛇は、進行方向の摩擦は少なく、進行方向と直行する方向の摩擦の大きいタイヤのような素材でできています。（下の図がタイヤと板で作った模式図）その場合、蛇は進行方向と直行する方向にくねるのですが、摩擦が大きいため、くねるのに大きなパワーが必要になります。そのため、パワーが摩擦の少ない進行方向に逃げ、前に進んでいきます。（簡単に言うと、左右には摩擦が大きくて動きづらいから、摩擦の少ない前に行くか～という感じです）これがグライド推進で、これを実現するために必要なのは、くねる角度の設定でしょう。プログラムの話に入る前にくねる角度の設定を説明します。（ちょっと高度な数学なので、読み飛ばしてもらっても結構です）

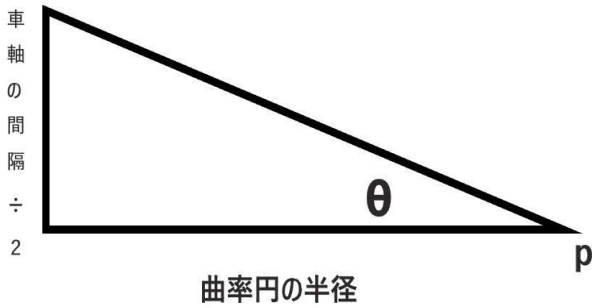


２くねり角の設定



上で用いた蛇の模式図で考えてみましょう。
まず、推進を時間 t を用いて表します。
x = wt（w は比例定数）
そして
y = f(x)とします。
蛇の動きは正弦波(y=sin(x))によって近似できることが知られているので、f(x)=sin(x)とします。

この時、くねりの角度は左の図における θ
にあたります。この θ は f(x)の曲率半径[1]（f(x)に接する円の半径）と車軸の直線のなす角にあたります。
θ を求めるために曲率円の中心 p,車軸の中心、車軸と車軸の中点の三点でなる三角形を取り出して考えてみます。



ここで車軸の間隔÷２＝a(定数)とします。
また曲率円の半径 r は曲率円の半径の公式より、
$r = (1+f'(x)^2)^{3/2}/|f''(x)|$
f(x)の一階微分と二階微分を用いてこのように表せます。そして今回は、蛇の動きを正弦波によって近似するため、
$r = (1+(\cos(x))^2)^{3/2}/|\sin(x)|$となります。よって、$\theta$ の値は

$$\theta = \arctan(a/\{(1+\cos(x)^2)^{3/2}/|\sin(x)|\})$$

これで求められます。因みに先頭の arctan はアークタンジェントと言って tan の逆関数で、tan の値から角 θ を求めるものです。また、x はパラメター t を用いて wt とあらわされていたことにより、 t の関数として θ を表すことが出来るのです。プログラムでは最適な w の値を試行錯誤で求めて実装しています。次からやっとプログラムの話になります。

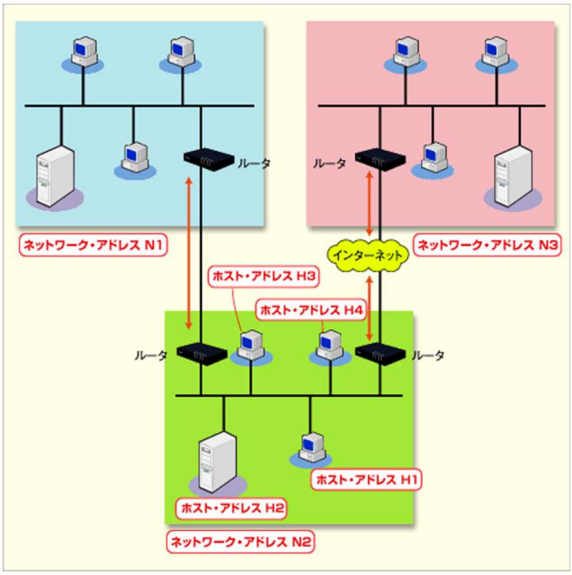
3 raspberry pi

raspberry pi とは蛇型ロボットに搭載されているパソコンのようなものです。ここでブラウザと通信をしています。ブラウザ上のホームページのプログラムは後々詳しく説明するとして、ここではブラウザと raspberry pi の通信について説明します。この通信では IP アドレスというものが重要になってきますが、IP アドレスとは何かご存知でしょうか？名前を聞いたことは在るけどよく分からないということがほとんどだと思います。IP アドレスとはパソコンなどのコンピューターに割り振られた番号のことで、32bit(2 進数で 32 桁)であらわされていますが、人間がわかりやすいように 8bit（2 進数で 8 桁）ずつ分けて 10 進数あらわしたものです。

[1] 曲率円の半径　完璧に余談ですが、曲率円の半径は意外と身近にあります。道路の標識で R=200m のようなものを見たことは在りませんか？これはカーブのきつさを表す標識なのですが、カーブに接する円の半径、つまり曲率円の半径を表しています。半径 R が小さということはカーブが急ということを意味しているんですね。

ex)11000000.10101000.00000001.00000000→192.168.1.0

　このうち上 24 桁をネットワークアドレス、下八桁をホストアドレスと言います。
ネットワークアドレスはネットワークを識別するためのもので、左の図のようにグループに割り当てられるアドレスである一方、ホストアドレスはそのグループの中で個々を識別するために割り当てられるアドレスとなっています。
　raspberry pi とブラウザの通信では、同一の WIFI に raspberry pi とデバイス（僕のスマホ）をつなげることによって同一のネットワーク下に置き、raspberry pi の IP アドレスにデバイス（僕のスマホ）がアクセスすることによって通信を行っています。

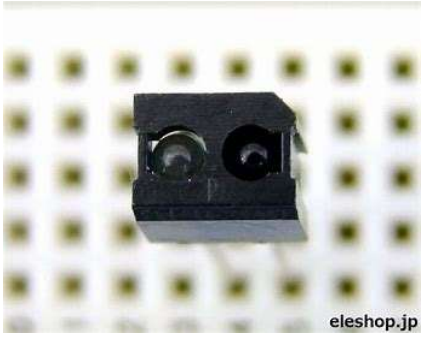


4 サーボモーターのプログラム

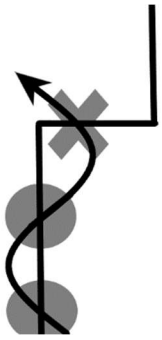
すべてのサーボモーターを同時に動かすために、それぞれのサーボモーターを目的の角度まで 1 度ずつ動かしていくという一般的なプログラムを書きました。動かすのはプラス 1 度なのかマイナス 1 度なのかということは、その時のサーボモーターの角度と目的の角度を比較して決めています。

5 黒線を沿うプログラム

黒線を沿うには黒線と黒線を引いてない場所の色（この場合は白）を判別する必要があります。
それに僕たちが使ったのが「反射型フォトセンサ」です。



←反射型フォトセンサ



←図 1

反射型フォトセンサとは、上図の左の丸い部分から光を出し、その光が何かにあたって反射されると右の丸い部分がそれを感知し、反射された光の量によって数値を返すというものです。白はたくさん光を反射し、黒は光をよく吸収するので、数値に大きな差が出ます。これによってしっかり黒線と認識することができます。実際のロボットはくねりながら前進しているので、ある一定の間隔（図１の黒丸）で黒色が認識されていれば黒線は直線である、それより早くまたは遅く（図１のバツで）認識された場合、黒線は曲がっていると判断しその方向に曲がります。

感想

礒野　初め、僕の担当はプログラムでした。そしてサーボやフォトセンサのプログラミングをもときさんに教えていただきながら組みました。ですが、僕の知識不足や怠慢でほぼすべてのプログラミングをもときさんにやっていただいてしまいました。それに加え、僕は本体のことにも全然関わっていません。もときさんのおかげで素晴らしい製作物ができたと思いますが、僕の姿勢については反省しています。これから２０日ほど製作できるので、その間僕にできる事を全力でやろうと思います。

もとき　物無で最後の制作でしたが、蛇の動きを研究し、数学的に動きを解析することが出来たので、当初僕が目標として抱いていた理論的な物体の制御という面において成功したと思います。また、後輩と切磋琢磨することが出来とても楽しかったです。

不明

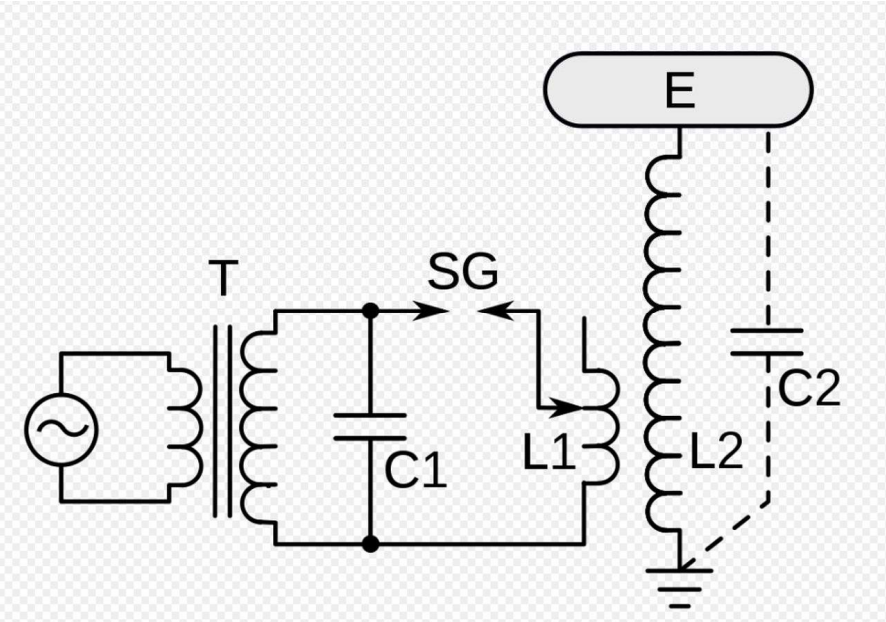
by ONO

テスラコイル　　製作者：H2　小野敦

ストーリー
ある天才博士がいた。彼は数年にわたる研究の末、劣等機種たるスマートフォンを破壊する装置の開発に成功したのだった。
「遂にテスラコイルが完成した、これでスマホをこの世から駆逐できる。この瞬間を記念して写真を撮ろう」
（ガラホを取り出す）
「何、なぜだ、なぜ動かん。まさかドコモと LINE が結託しｔ・・・

悲報：ドコモのガラホにおける LINE サービス全面終了！

・概要
知っている人も多いと思いますが放電を発生させるための装置です。原理としてまず一次側の入力信号は変圧器の巻数比により増幅され二次側の回路に伝えられます。また一次、二次の回路はそれぞれ直列共振回路になっています。直列共振回路においてはＬとＣの合成インピーダンスはある周波数で 0 になります。そうすると結果として回路には大変大きな電流が流れＬとＣの両端には大きな電圧が現れることになります。一次、二次の回路の共振周波数をそろえ、その周波数の電圧を一次側にかけることで、結果として二次側から大きな電圧を取り出すことができます。
・回路図



この回路図では、まず C1 が充電され一次側の回路の電圧が上がり SG で放電が起こります。放電で電圧を裁断した瞬間にはいろんな周波数の成分の電圧が含まれており、その中の

丁度いい周波数の成分が共振現象を起こします。

T（トランス）：

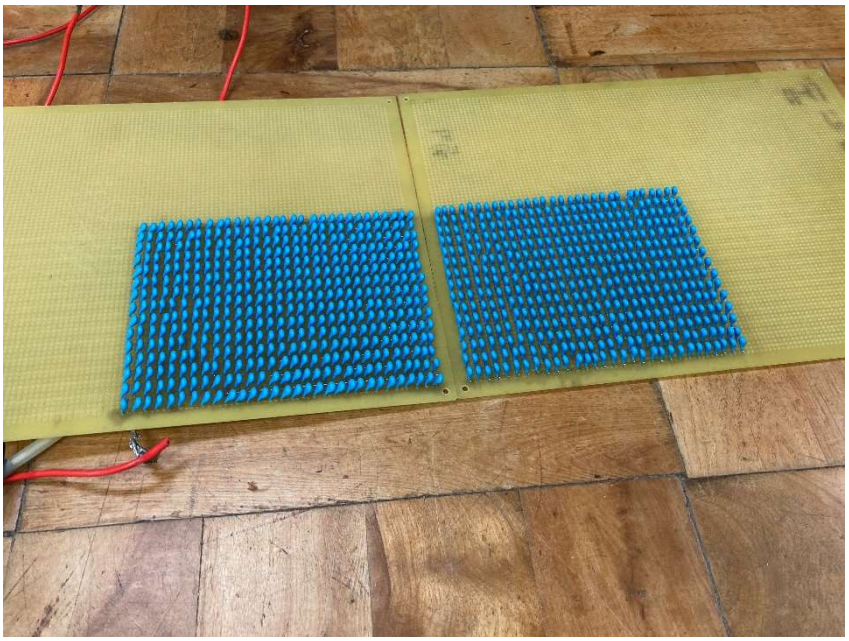
今回は家庭用電源を 15kv まで上げるネオントランスを使用しました。結構危険なんで注意して下さい。

L1（一次コイル）：

一次コイルは太めの配線材などを 10～20 回程度巻いて作ります。今回は巻き数を変えやす

いようになまし銅管を使用しました。なまし銅管という割には硬かったためいびつな形になってしまいました。作ろうとする場合は巻く前にもう一回なました方が良いと思います。

C1（一次コンデンサー）：



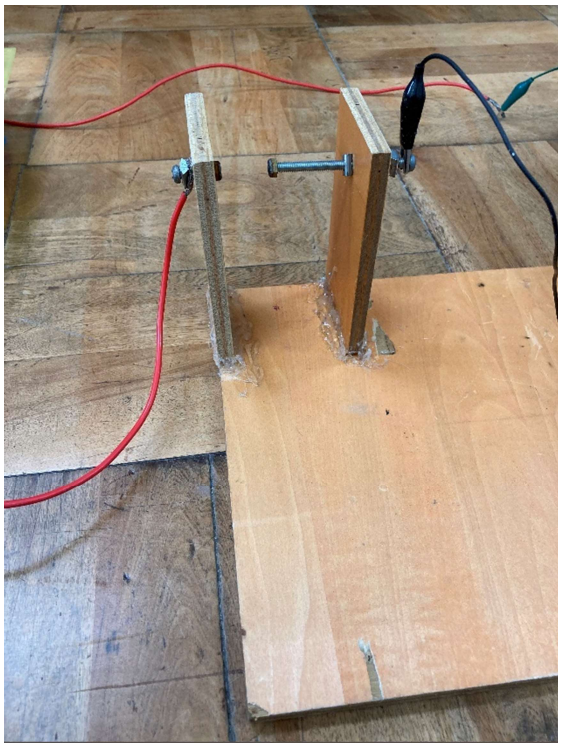
今回は 2kv1000pf のものを大量に直並列に繋ぎ 30kv4nf にして使いました。

L2　（二次コイル）：

今回は直径 10 ㎝程の塩ビパイプに太さ 0.5 ㎜のエナメル線を 1000 回程度巻きました。

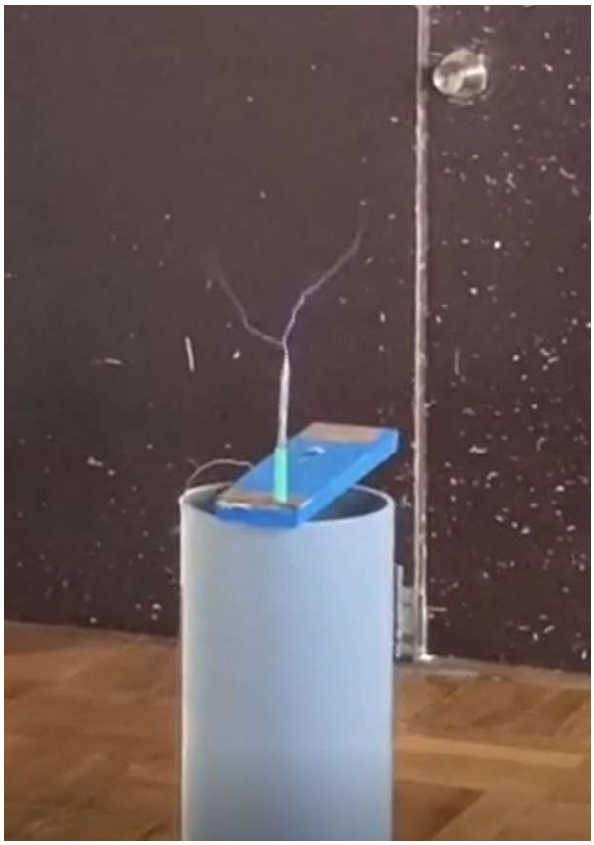
E（放電部）：どういうのがいいかよく分からなかったので近くにあったネジを使用しました。

SG(スパークギャップ)：



間隔を調整しやすいようにネジを向かい合わせて使いました。

・放電の様子（白黒だとよくわからないかも）↓



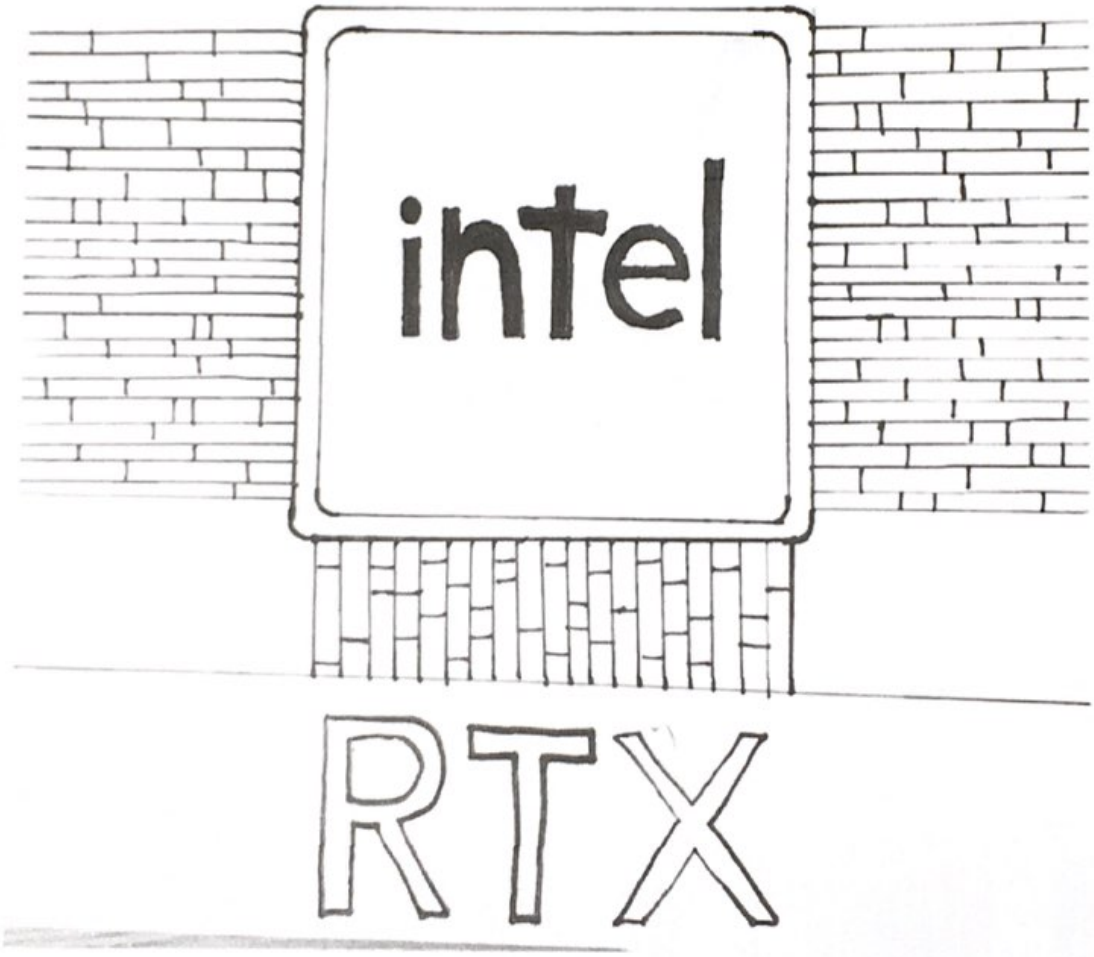
・感想

放電は 10～15 ㎝ぐらいのものが出せましたが、思ったより動作が安定しませんでした。多分この回路図だと安定しにくいと思うのでこれを真似して作るのはあまりお勧めしません。製作自体は楽しかったです。

P.S. 製作がうるさくて申し訳ございませんでした。この場で謝罪させていただきます。

Computer

コンピューター



AZABAC-71

制作：中二　和田　中三　畠山　高二　宮田
協力：物理部無線班の皆様

はじめに～*the story of "AZABAC-71"*～

AZABAC（アザバック）シリーズは、通産省から譲り受けたコンピューターを皮切りに始まり、物理部無線班の長い歴史の中で受け継がれてきたコンピューターを自作する製作です。
最新の作品である２００７年の AZABAC-67 以降はこのシリーズを受け継ぐ製作はなされず、近年ではコンピューター系の製作物すら少なくなってきているのが現状です。

人々は不可能だと思った・・・
人々は地味だと嫌った・・・
しかし、三人の男が立ち上がる！
彼らの手には、一対の基盤があった！
「AZABAC-71」、始動！

概要

シリーズの最新作である「AZABAC-71」は、単純なロジック IC のみを使った、単純な 8bitCPU の自作を目標として、基板二枚を使った回路となりました。回路は多くの部分を「CPU の作り方」という本を参考に作らせていただきましたが、絶版になっている IC を取り替えたり、機能的な強化を図るにつれてオリジナリティの高いものとなりました。

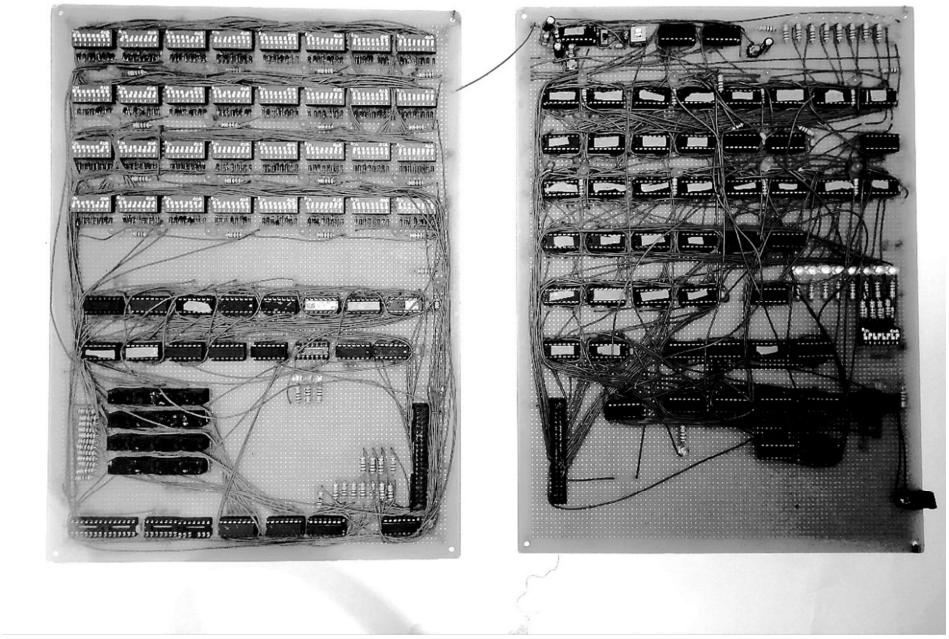
8bitCPU とは？

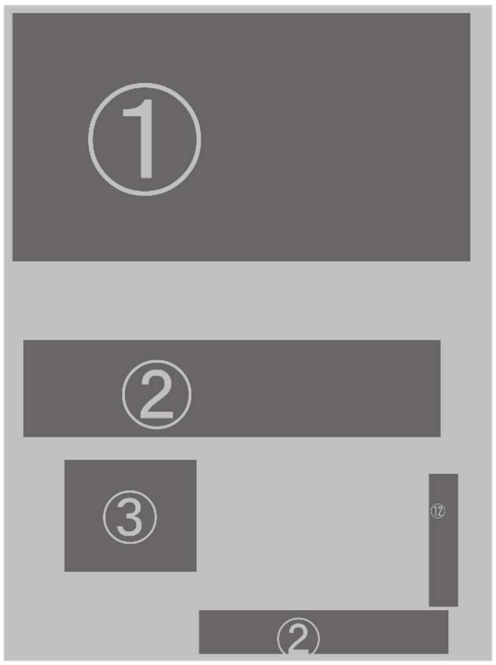
さて、読者の皆さんは CPU(Central Processing Unit)とは何かご存じでしょうか？

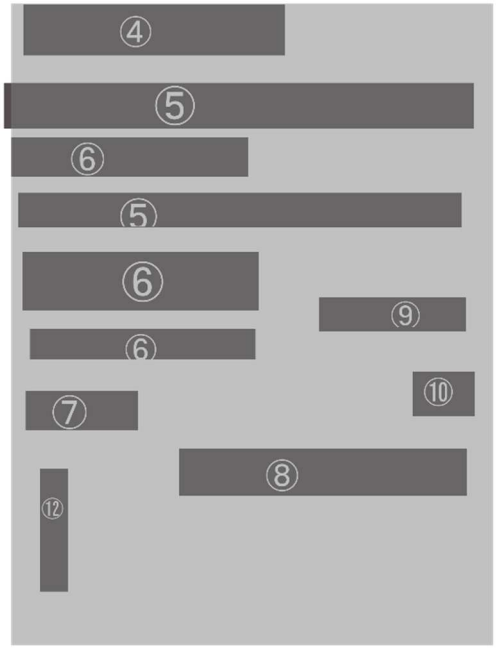
ズバリ、CPU とは、「計算機」です。

とはいえただの電卓ではありません。言い直せば、「プログラムの通りに計算ができる計算機」となります。皆さんの使っているスマホやパソコンも、CPU がプログラムに従って計算をしているから動いているんですね。今回の CPU は、最大８bit（０と１が八つ）まで計算できるので、8bitCPU と分類されます。ちなみに僕が今この文書を描いている PC は６４bit の CPU を使っているそうです。途方もないですね・・・。

外観







使った部品

特大基盤	２枚
LSIC	７０個超
IC ソケット	７０個超
コネクタ	２個
各種 LED	約４０個
各種抵抗	約６０個
各種コンデンサ	５個
リレー	１６個

部分ごとの説明（前頁と合わせてご覧ください）

① ROM

ROM（Read Only Memory）とは、プログラムの途中に書き換わることのない数値を入れておく入れ物の事です。と書くと分かりづらいですが、写真をちょっとよく見ればわかる通り、ただのスイッチの配列です*。参考書籍に習い、逆流による混線防止のためにダイオードを置きますが、向きには気を付けましょう。

プログラムと聞くと、アルファベットやら記号やらで書かれているのを想像しがちですが、すべて大本をたどれば０と１になります。０と１で書くプログラムのことを機械語と言ったりしますが、この CPU では８つのスイッチを二組、ON（１）にしたり OFF（０）にしたりすることでプログラムを書くんですね。今回は、8bit を命令、8bit を Im(Immediate)データという命令に付加する情報を表すために使用します。つまり、AZABAC-71 には 32 個のスイッチがついているので、命令を 16 行書けるってことですね。今回はどこを読み取っているかわかりやすいように、LED を付けて、読み取り中は光るようにしました。*ホントはヒューズを使うらしいです。

② プログラムデコーダ

何番目のプログラムを読むかを受け取った際、ROM からプログラムを読み取り、右側の基盤が読めるように変換（デコード）する回路です。たとえば、「００００００００１」という命令を、「A レジスタに Im データ（今回は１）を足したものを A レジスタに代入」といったように。この場合は、右側の基盤の「A レジスタに入れることを表す配線」と「A レジスタに代入することを表す配線」に電流を流し、「１」という数字はこの命令における付加情報なので、Im データとして「１」を送ることで、右側にプログラム通りの変化がおこるわけですね。

①　リレー群

製作途中、RAM から命令を読み取る際に困ったことが起きました。今回、１３８という IC の出力をそのままスイッチにつなげ、スイッチの内容を読み取っていたのですが、LED をつけてからというもの、スイッチを ON にする数が増えれば増えるほど、スイッチから流れる電圧が下がり、ついにはプログラムデコーダで読み取れなくなってしまったのです。このままではプログラムを書くことはできないので、リレー（電気式のスイッチのようなもの）を使って、電源をそのまま流し、０と１を明確にしました。すべての命令（行）について行ったので、命令が切り替わるたびにリレーのカチカチ音がしていい感じです。

②　クロック回路

この回路全体の標準クロックであるマスタークロックを生み出します。発信回路とも言います。このクロックを加工したもので回路の動くタイミングを計るんですね。
今回は、手動で動かすマニュアルモード、自動で動くオートモードを実装し、オートモードでは二種類の速度を切り替えられるようにしました。

③　レジスタ群

プログラムの通りに計算された値をしまっておくための箱がレジスタです。レジスタから取り出した値に操作をして（適当な値を加えて）、レジスタに戻す（代入する）のが CPU の動作の基本になります。
それぞれのレジスタには LSIC の７４１９３を使用していますが、この IC には LOAD というピンが存在します。このピンに信号を入れることで値を読みこむタイミングを制御できます。UP ピンに信号が入るたびに中の値が増える「カウンタ」の機能もついており、プログラムカウンタにするのが楽でした。
今回の CPU にはレジスタが３種類あります。

- 汎用レジスタ
  八つあるレジスタ（A～H の名前がついています）のうち、六つを使っています。それぞれに値を入れたり取り出したりすることができます。（高度なプログラミング言語における「変数」のような使い方ができます。）
  六つあると書いたんですが、命令の関係上使えるのは四つだけになります。プログラムデコーダを改造すれば六つフル活用できるようになるはずです。
- プログラムカウンタ
  G レジスタを使っています。今どの行のプログラムを読んでいるのかを判断するため

- のレジスタですね。このレジスタの中身をそのまま左側の基盤に渡すと、左側の基盤に搭載されたプログラムデコーダから右の基盤への指示が返ってきます。
- I/O ポート（後で書きます）

① データセレクタ

たくさんのレジスタの中から「どのレジスタから値を読むか」を選択する機構です。今回は参考書籍に則り７４１５３を使用しました。一つの IC で２桁分制御でき、選択したものからさらに絞る必要のある場合もあるということで、４つのレジスタで IC４つ、８つだと１２個必要になります。今回は、二進数３桁の入力で８つのレジスタを選択できます。

② ALU

ALU（Arithmetic and Logic Unit）は、この CPU を CPU たらしめるものの一つで、計算する役割を担います。参考書籍に則り、７４２８３を使用し、８bit の加算ができ、結果がこの値以上になったときに知らせてくれるようになっています。（キャリーフラグを立てることで知らせています。）Logic とか言ってますが論理演算はできません。

③ フリップ・フロップ群

レジスタに使用している７４１９３ですが、一つ問題点があります。この IC は **LOAD ピンに H が入っている間はずっと値を読み込み続けてしまう**のです。このままだとクロックによる制御がしづらいので、何とかしてクロックの立ち上がり（電流が流れた瞬間）を感知し、その瞬間**だけ**値が書き換わるような仕組みを作る必要がありました。
そこで、この製作物では ALU の出力とレジスタの入力の間にフリップフロップを挟み、レジスタの LOAD ピンとずらして動かすことで対応しました。
ずれたクロックの生成には７４１６４という IC を使っています。

④ 出力

I/O ポートというレジスタの出力はほかのそれとは違い、セレクタではなく LED につながっています。そのため、８つの LED は常にこのレジスタ（H レジスタ）の中身を映し続け、H レジスタを操作することで LED の光り方を変えることができるのです。

⑤ 入力

では、データセレクタ側の H レジスタに相当するピンはどうなっているのかと言いますと、スイッチにつながっています。この状態で「H レジスタから値を読み込む」ことはこのスイッチの状態を読み込むことと同義になるというわけですね。

少し話が前後するのですが、前に書いた「プログラムカウンタ」も、出力がセレクタにつながっていません。その代わりに左側の基盤につながっているのですが、では空いたセレクタの入力はどうなっているのかというと、すべて電流が流れていない状態（GND）に固定されています。こうすることで、G レジスタを「ゼロレジスタ」として使うことができます。

①　ゼロレジスタ

このレジスタを参照することで、ゼロを得ることができます。何に使うかというと、「何も足さない」作業をするのに使います。実は、この CPU の ALU は、「あるレジスタに一つの数値を足す」ことしかできないんですね。そこで、「B レジスタに１２を代入」のような命令の時には、「１２にゼロレジスタを足して B レジスタに代入」と解釈します。

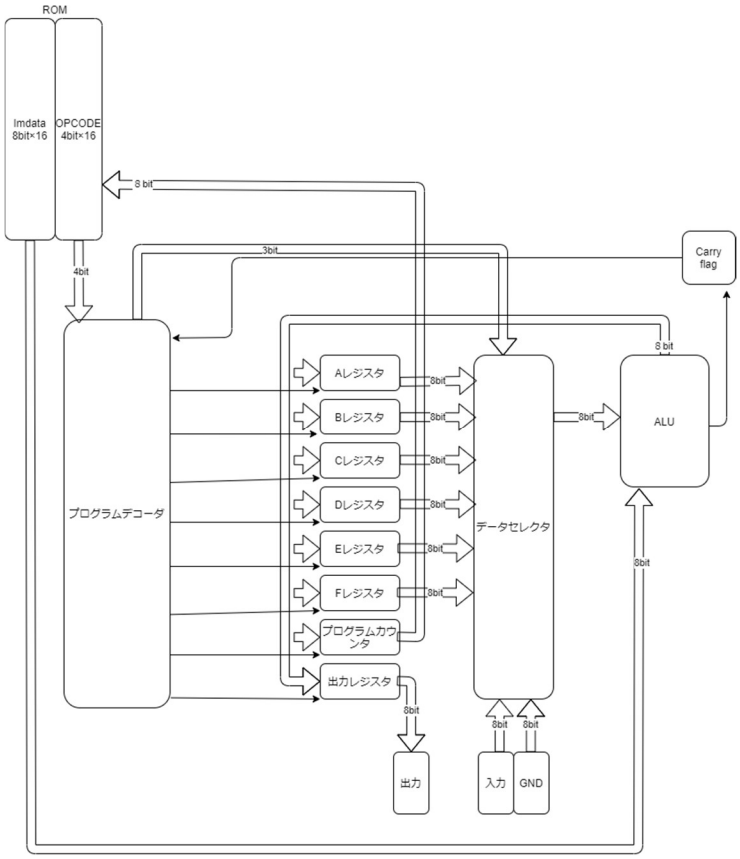
②　コネクタ

この製作物は基板二枚組ですが、間をじかに配線するのではなく、３６ピンのコネクタとケーブルを使って橋渡ししています。二枚の基盤の片方には ALU とレジスタ関連のものを、もう片方には ROM 関連のものを乗せることで改造しやすくなりました。実際に、ROM 側の基盤を取り外し、SRAM と新しいプログラムデコーダを乗せた基盤と接続することで、より長いプログラムが書けるようになります。

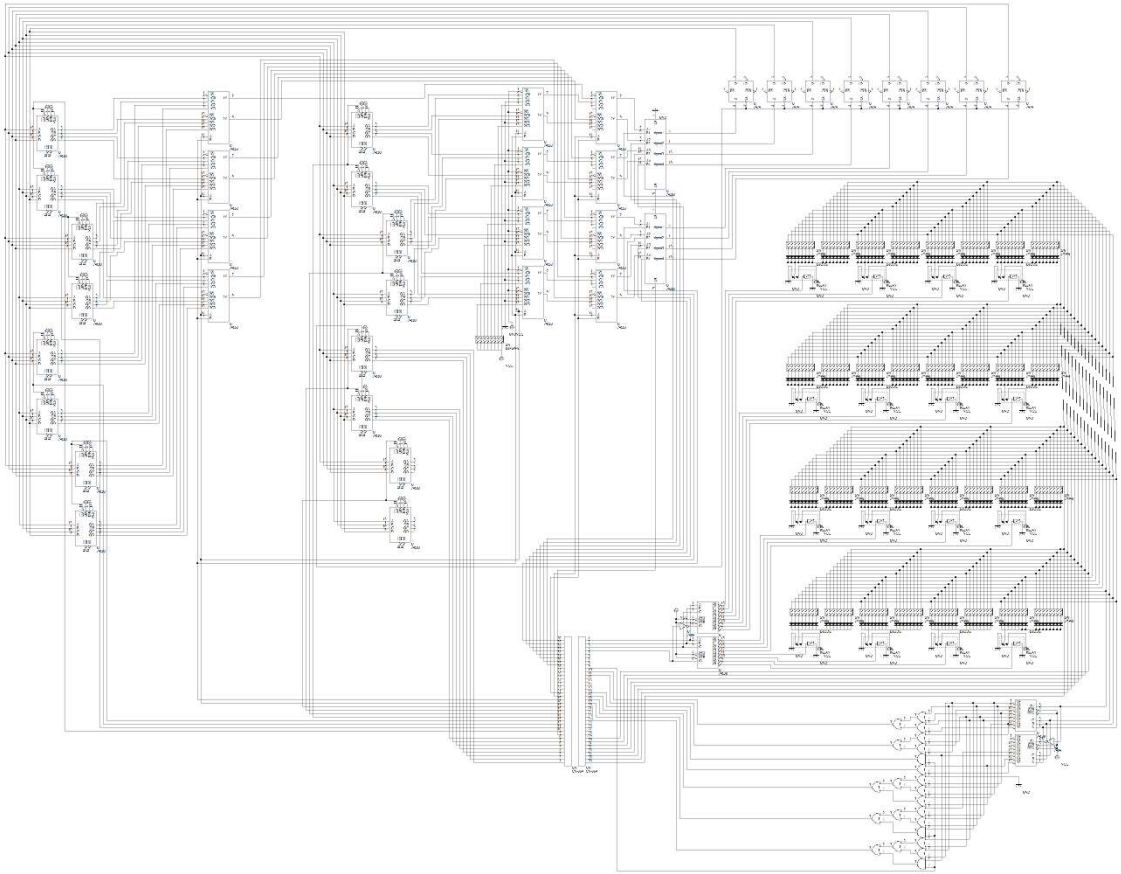
仕様書

アーキテクチャ	RISC、ハーバード型
I/O アドレッシング方式	メモリマップド I/O
汎用レジスタ	６バイト（４８ビット）
入出力	８ビットずつ
最高動作速度	不明
命令形式	１２ビット固定長

ブロック図


ROM
Imdata
8bit×16
OPCODE
4bit×16
8 bit
4bit
3bit
Carry
flag
8 bit
プログラムデコーダ
Aレジスタ
Bレジスタ
Cレジスタ
Dレジスタ
Eレジスタ
Fレジスタ
プログラムカウンタ
出力レジスタ
8bit
データセレクタ
ALU
8bit
出力
入力
GND

回路図



感想

宮田：コンピューター系の製作物は初めてでした。しかし、最後の年としては今まで物無で培った技術や知識の結晶としていい製作物になったのではないでしょうか。付き合ってくれた後輩たちには感謝しかないです。ほんとにありがとね。

畠山：初めは CPU の仕組みを理解するのに苦戦したのですが何とか理解することができました。ですが、仕組みを理解していても原因不明のバグを直すのがとても難しかったです。また、この制作でロジック IC などの知識が増えたので中一ゲーの回路図設計に生かそうと思いました。

和田：僕は先輩方に言われた通りに配線とかをしてただけで CPU の仕組みを殆ど理解していなくて、工作は楽しくても先輩方が何をしているのか分からないことが多かったので、次の制作の時は自分の制作物の仕組みを理解したいと思う。

ゲーム

OTOAGE

～おとあげ～

製作　M3中嶋
協力　M3中嶋
スポンサー　物理部無線班

宇宙の起源とその崩壊

N嶋「音ゲー楽し～！！」
ちか隊「分かる～神ゲーだよなｗｗｗ」
えぬ嶋「俺たちの制作これにしようぜｗｗｗ」
N嶋「「「それなｗｗｗｗ」」」
油そば替え玉嶋「自転車残ってるしそれで音ゲー作らん？」
全ての自然数嶋「「「は？？」」」
～こうして自転車は解体され、音ゲーが完成することはなかった…我々は両手を挙げて、迫りくる期限を待つしかなかったのである。～

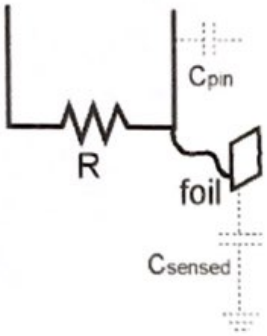
説明

概要

~~このゲームは自転車をコントローラーとし、ブレーキやハンドルの方向を感知して操作する音ゲーである。~~
静電容量式センサと超音波距離センサをarduinoを介してコントローラーとして、Unity内のゲームを遊ぶというものである。
ゲーム内では音楽のリズムに合わせて奥から音符たちが流れて来るので、曲を聞きながら指定された場所を押すという一般的な音楽ゲームである。
しかしこのゲームではリズムに合ってるかどうか判定する箇所が平面であり、下の部分は静電容量式のパネルで、上は距離センサを用いて空間に手をかざし、判定するものとなっている。

回路

延々と僕のスパゲッティーコードを貼り付けるのは見づらいので原理のみ解説します。



パネルの方のセンサは、左図の回路にパルスを流し、右下のセンサに触れると人間の静電容量によってパルスの検出が若干遅れるのでそれを感知する仕組みです。

距離センサは、何個も設置することで二次元的な腕だいたいの座標を求めて空中のコントローラーとしてます。

Unity

音ゲーの仕組みとしては１秒間にどれだけの速さで譜面が動くか設定し、そこから逆算してノーツを設置しています。いちいちソースコードに書くと長くなって鬱陶しいのでtxtから入力できるようにしました。
ノーツの種類はタップ、払うフリック、押さえ続けるホールド、指示通りに手を動かすスライドの4種類あります。空中に設置されているものは空中で同じことをします。判定は判定を行うエリアと物理判定をしています。~~あたまわるい~~
スライドの描画はベジェ曲線を利用して、様々なな傾斜のスライドができるようになっています。

感想

計画力のなさを実感しました。迷走続きで今制作中の形を思いついたのが今年の１月くらいなので振り返って見ると計画性皆無でした。共同制作者がいないこともありましたが、もっと制作のビジョンを最初からはっきりさせれば誰かしら来てくれたのかな、と思っています。
来年はプログラミングをもっときっちり学んでクオリティの高いゲームを作りたいです。

Bututendou Labo

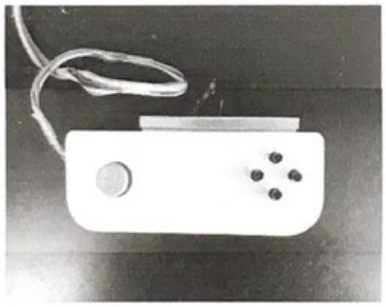
製作者　高２　円藤　颯太
　　高１　池田　宜史
　　中３　加藤　稜大
　　高梨　勇之介
協力　物無の皆様

・ストーリー

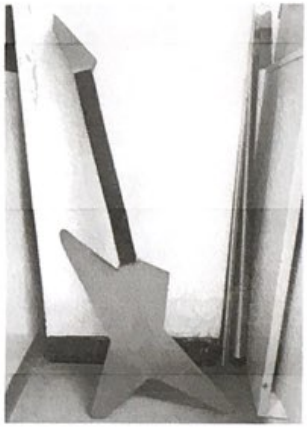
昔々あるところに二人の神がいました。二人仲よく日々過ごしていたのですが、ある日サマボケのメインヒロインの意見の食い違いから大喧嘩してしまいました。そのうち一人の神は怒りのあまり台パンで彼の仕事で使うノートパソコンを粉々にしてしまいました。そしてその粉々になった破片は大地に降り注ぎ、「モノリス」となってまだサルだった人たちに知性を与えました。そのうちの一つがこの板型ゲーム機なのです。
※この物語はフィクションです
※サマボケのメインヒロインは鳴瀬しろはです

・全体の概要

Unity という 3D ゲーム開発ソフトで作った四つのゲームを、Arduino というもので一つのコントローラーでプレイできるようにした制作物です。一部のゲームには専用のアタッチメントがついており、よりゲームへの没入感を高めます。



＜コントローラー＞



＜アタッチメント（ギター）＞

・ゲームの説明

＜ゲーム１＞

弓を射るゲームです。弓のアタッチメントがついていて、それを動かすことで視点操作をし、射ることでゲームの中の弓を操作します。そのため、まるで本当に撃っているかの

ような感覚になれます。遊べるステージは、

「スナイプ」：矢を当てるたびに離れていく的を正確に撃ち抜いていきます。

（物無員最高記録は１２０m）

「動的射撃」：動く的を正確に撃ち抜いていきます。(物無員最高記録は 50 点)

「早撃ち」：どんどん出てくる的を、的が消える前に撃ち抜いていきます。

（物無員最高記録は９８秒）

の三つです。

＜ゲーム２＞

ギター型のアタッチメントがあり、それを操作して遊ぶ音ゲーです。赤、青、緑、黄色、紫のレーンがありそれぞれに対応したボタンを押しながら、ノーツに合わせてタイミングよくジョイスティクをはじくと点数が入ります。曲数は 4 曲、ノーツの流れる速さを変更できます。

＜ゲーム３＞

一定時間敵の攻撃を避け続ける弾幕ゲームです。難易度は 3 種類から選べて経過時間によって敵の攻撃が変わります。3D なので割と難易度高めです。弾幕の撃ち方が綺麗に見えるように工夫しました。

＜ゲーム４＞

制限時間内にゾンビを倒して建物から脱出するゲームです。ゾンビは大小２種類いて、大きいゾンビを倒すと建物の出口が開きます。難易度は easy、normal、hardcore の３つあります。easy ではライトマシンガンを使い、normal からはハンドガンを使います。hardcore は通路にレーザーがあり、踏むとダメージを受けます。 プレイヤーの HP は５で、小さいゾンビとレーザーから受けるダメージは 1、大きいゾンビから受けるダメージは２です。

・仕組み

ゲームは unity で作っています。プログラムは４人分を載っけるのは多すぎるので割愛。コントローラは Aruduino Pro micro により制御しており、Aruduino Pro micro 内の 32u4 マイコンによって USB 周辺装置として認識されるので、プログラミングによってオリジナルのキーボードが作れます。そして unity でキーボードによって遊べるゲームを製作し、パソコンのキーボードで操作するのではなく、Aruduino Pro micro で制作したコントローラで操作しています。また、本体は３D プリンターによって設計しており、それにより、コントローラを付属のアタッチメントに付けられるようにしています。

・感想（詩？）

・円藤

今回が初後輩との製作でした。全員がきちんとやることがあって楽しい製作となった気がします。なぜか一句　ひさかたの光のどけき春の日に予定狂わすコロナ許さん

・池田

コロナなどありましたが、初めて四人という多人数での製作をしたので、純粋に楽しかったです。ここで一句 秋来ぬと目にはさやかに見えねども 風の音にぞおどろかれぬる

・加藤

忙しない日常の中で 制作を続けた その向こう側に 待ち受けるのは完動かスキーかスキー旅行の前にゲームが一応作り終わったので良かったです

・高梨

真っ青な秋晴れ、刺してくる木漏れ日はあたかも「あなたの行く方向はそちらではないですよ」とささやいているようだった。だから部活は休もうかな。

チカ チカ

TIKA TIKA

&

URIMONO

売り物

BUTURINTENDO SWITCH

製作者　M3　礒野大洋　協力　物無の皆様

ストーリー

礒野「ストーリー書けって言われても　この通り完成まで程遠い」

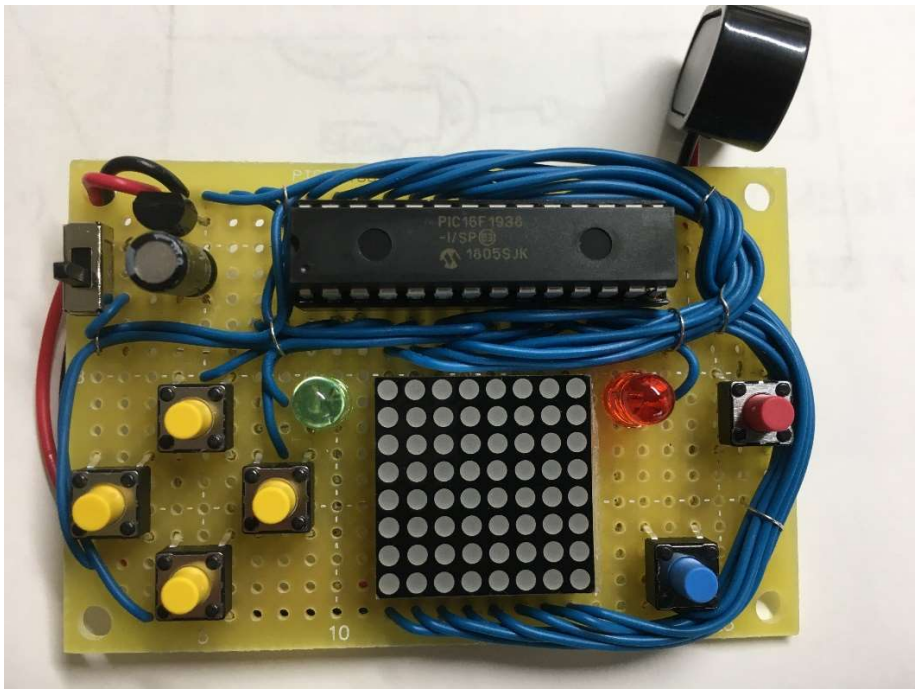
「量産終わったんでしょってお嬢さん　いや×３そりゃご冗談を」

「回路図見るか？って思ってる　役者は揃ってんのに家にこもってる　または踊ってる心躍ってる？」

「みんなこぞってゲーム実況撮ってる　僕も売り物で撮りますか」

「それで天下取りますか」

外観（がいかん）



部品の説明

LED・・・左右対称に１個ずつある丸いやつです。光るか消えるかしかしません。

ドットマトリックス・・・中心にある正方形やつです。８×８の LED でゲーム画面を表します。

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

タクトスイッチ・・・左に４つ、右に２つあるやつです。左は十字キー、右は AB ボタンの役割をします。

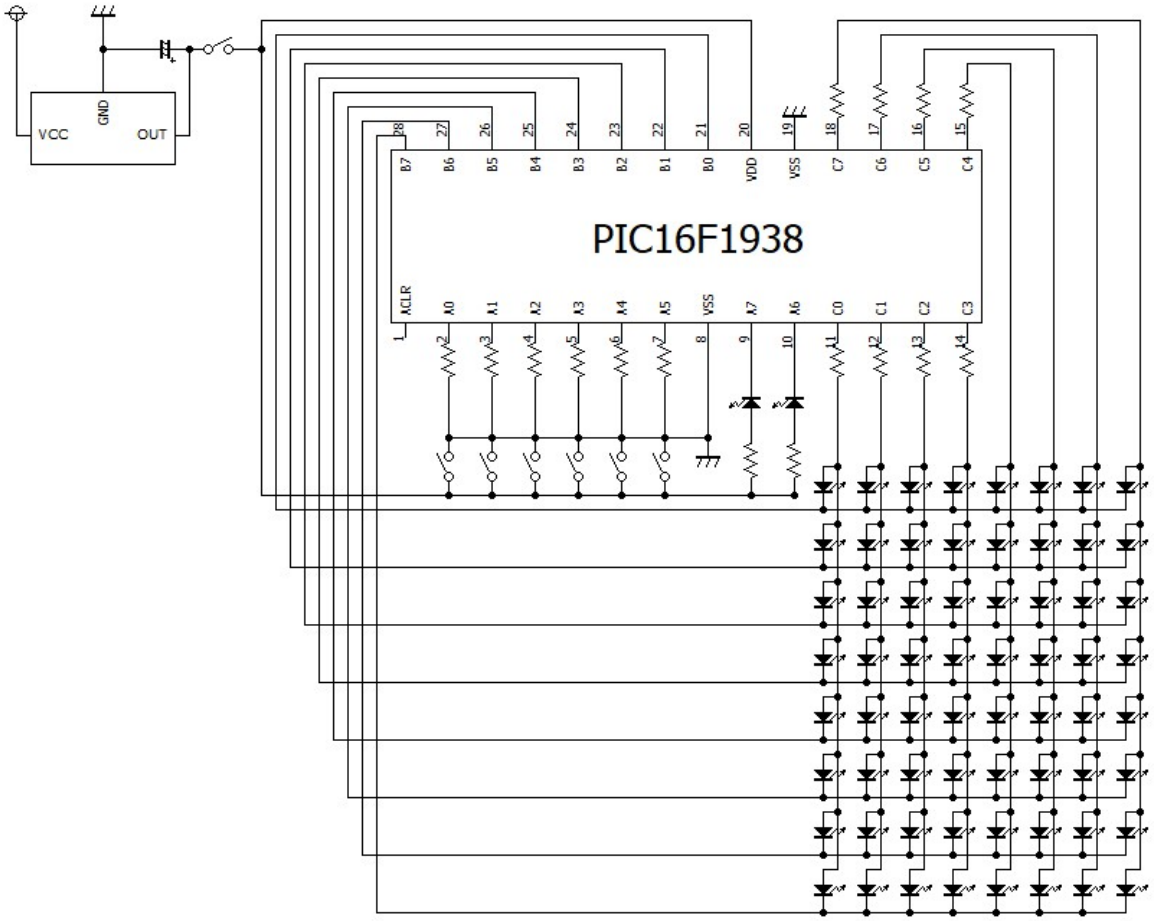
三端子レギュレータ・・・一番左上の半円に近いやつです。電圧を下げる働きがあります。
電解コンデンサ・・・レギュレータの下にあるやつです。電圧を安定させる働きがあります。
スライドスイッチ・・・コンデンサの左にあるやつです。レバーの上下で電源の ON/OFF を切り替えます。

概要（がいよう）

今年の売り物は４つゲームが入っています。電源を入れたときに表示されるメニュー画面でゲームを選択できます。

ゲームの説明

１つ目のゲーム・・・上から流れてくる弾をよけたり、ビームで消したりしてゴールを目指すゲームです。
２つ目のゲーム・・・表示されたコマンドを覚えて、間違えずに入力するゲームです。
３つ目のゲーム・・・常に進む蛇の向きだけを変えて、餌の場所まで向かうゲームです。
４つ目のゲーム・・・ほかの人が指定した場所を、センサーの反応をたよりに探し当てるゲームです。



回路図（かいろず）

感想

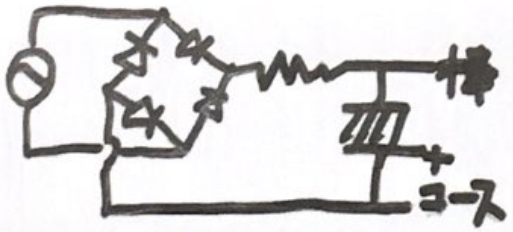
ゲームの構想や部品の配置、配線のルートなどをいろんな人の力を借りながら、１人で考えたのは、
自分にとってとてもいい経験だったと思います。この経験を来年の製作に活かしていきたいです。

イライラ棒

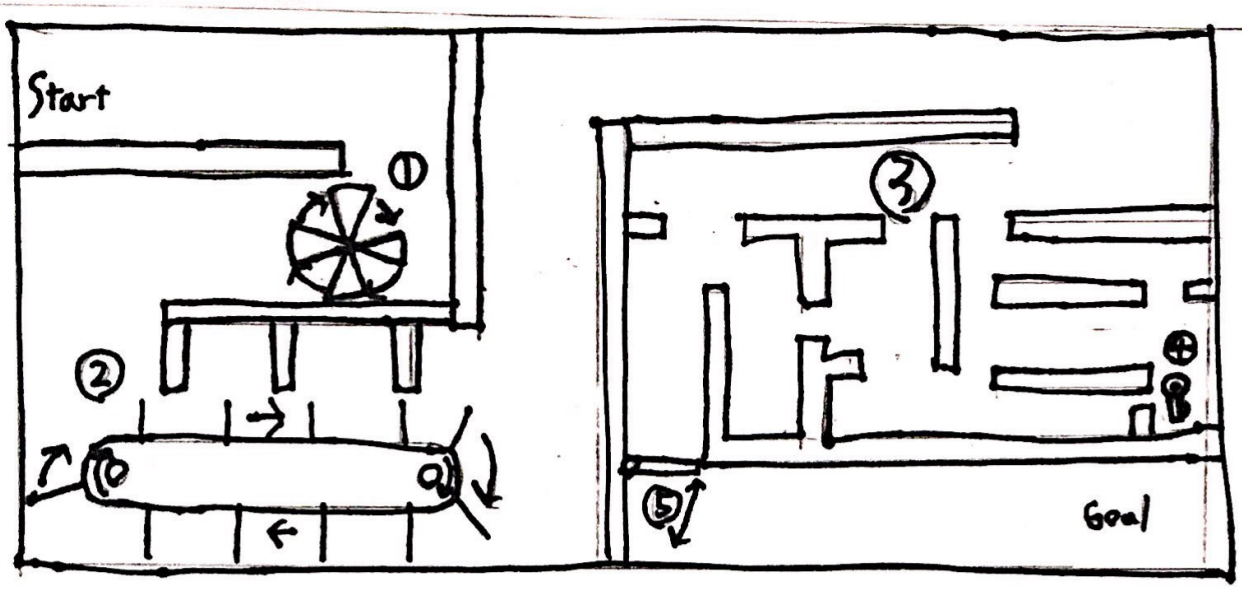
制作
M2 永谷
M2 島内
協力
物無の皆様

ストーリー
20XX年、とあるロボット工場で
幸か不幸か、あるロボットに自我が目覚めてしまった。
さまざまな機械を手負いくぐって工場から
脱出しよう!!

回路



CamScannerでスキャン



①十字の板が回転している。上手に避けよう。

②コンベアの上を仕切りが移動している。
安全地帯を活用しつつ進もう。

③迷路。安全に通れそうな道を見極めよう。

④鍵。とるとゴール前の扉が開く。

⑤扉。鍵をとって開けよう

感想

PICの使い方を理解できたので、とてもいい経験になりました。
教えてくださった先輩方、手伝ってくださったみなさん ありがとうございました。
永谷

木工についての知識を深められ、良い勉強になった。島内

CamScannerでスキャン

角材避け in 物無

製作：M2 広瀬 晴喜

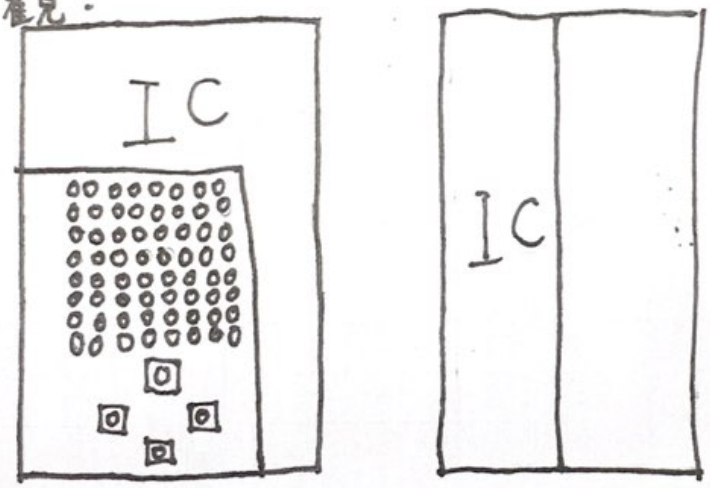
回路図設計：H1 中村さん

協力：物無の皆さん

ルール：上下からLEDの壁が流れてくるのでボタンで上下左右に動かして避けてください。

ストーリー：ある日、いつものように製作をしていると連日のストレスで暴徒化したイライラ製作員が角材を投げてきた！足に当たったら間違いなく怪我をするだろう。製作を続けるために角材を避けよう！

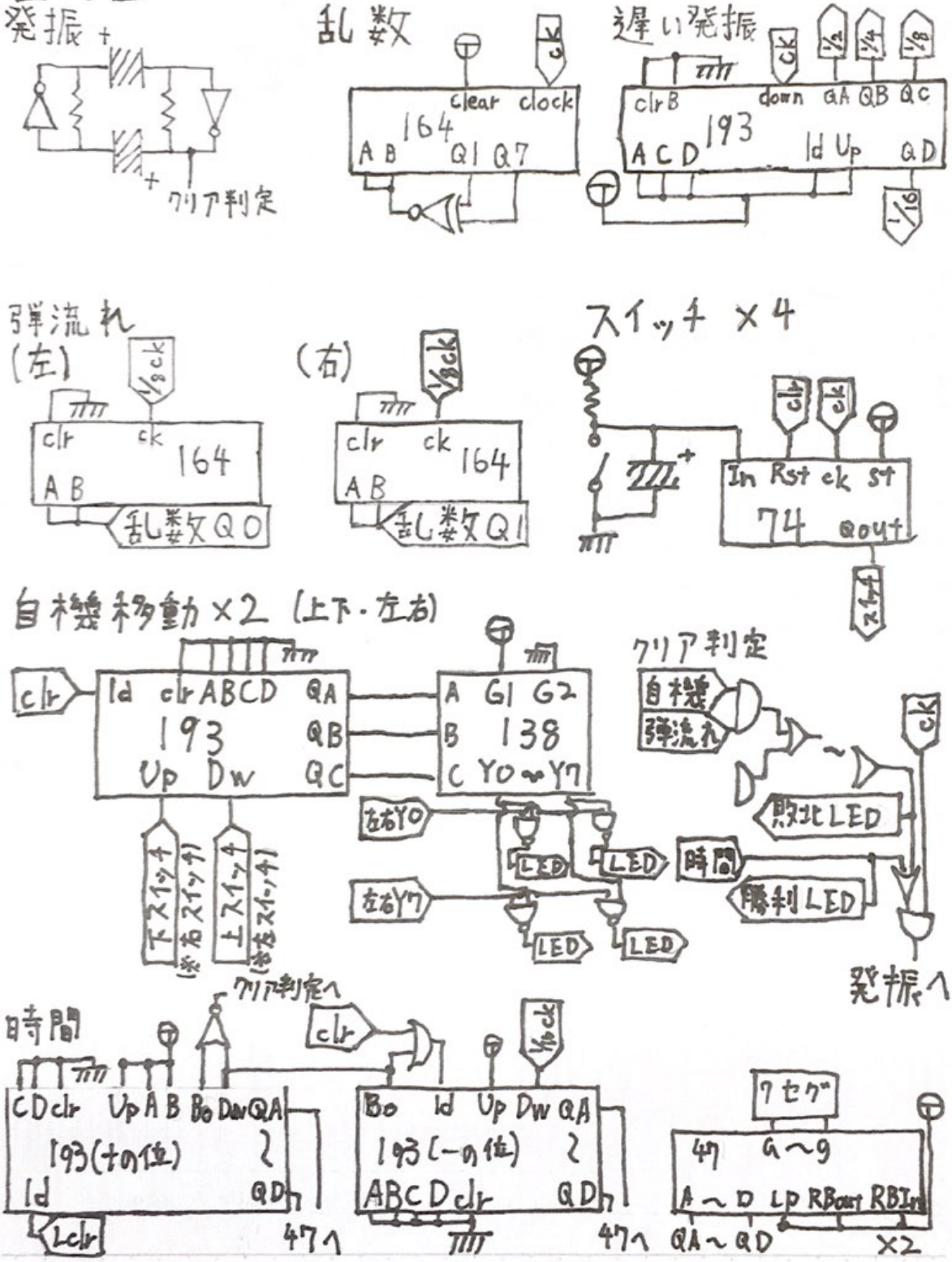
外観：



感想

中一の頃よりもはんだ付けが上手くなったと思います。また、ICの知識もつけることができたので良かったです。

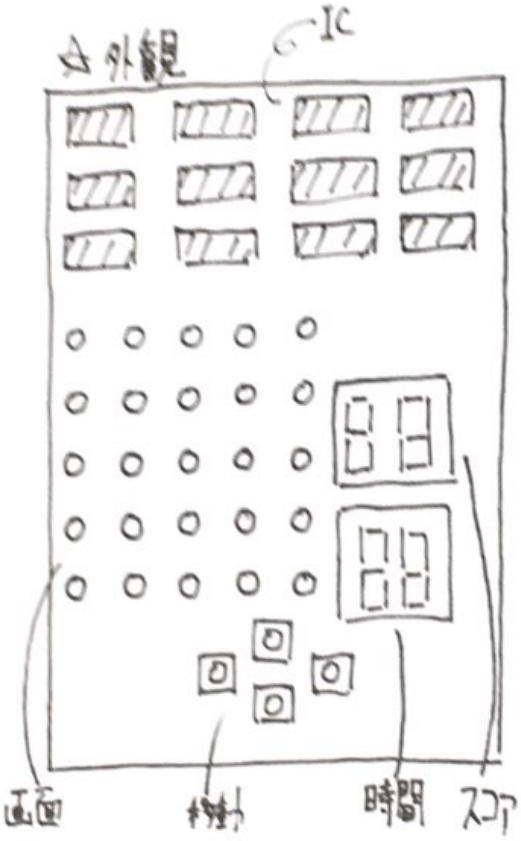
回路図


発振
クリア判定
乱数
clear clock
164
A B Q1 Q7
ck
遅い発振
clrB down QA QB QC
193
A C D ld Up QD
1/2 1/4 1/8 1/16
弾流れ
(左)
clr ck
164
A B
乱数Q0
1/8ck
(右)
clr ck
164
A B
乱数Q1
スイッチ ×4
In Rst ck St
74 Qout
スイッチ
自機移動×2 (上下・左右)
clr
ld clr ABCD QA
193 QB
Up Dw QC
A G1 G2
B 138
C Y0~Y7
下スイッチ (兼右スイッチ)
上スイッチ (兼左スイッチ)
左右Y0
左右Y7
LED
クリア判定
自機
弾流れ
敗北LED
時間
勝利LED
ck
発振へ
時間
クリア判定へ
clr
CD clr Up A B Bo Dw QA
193(十の位)
ld QD
Lclr
47へ
Bo ld Up Dw QA
193(一の位)
ABCD clr QD
47へ
7セグ
47 a~g
A~D LP RBout RBIn
QA~QD
×2

IC拾い

制　作：M2 山田
回路図設計：H1 中村さん

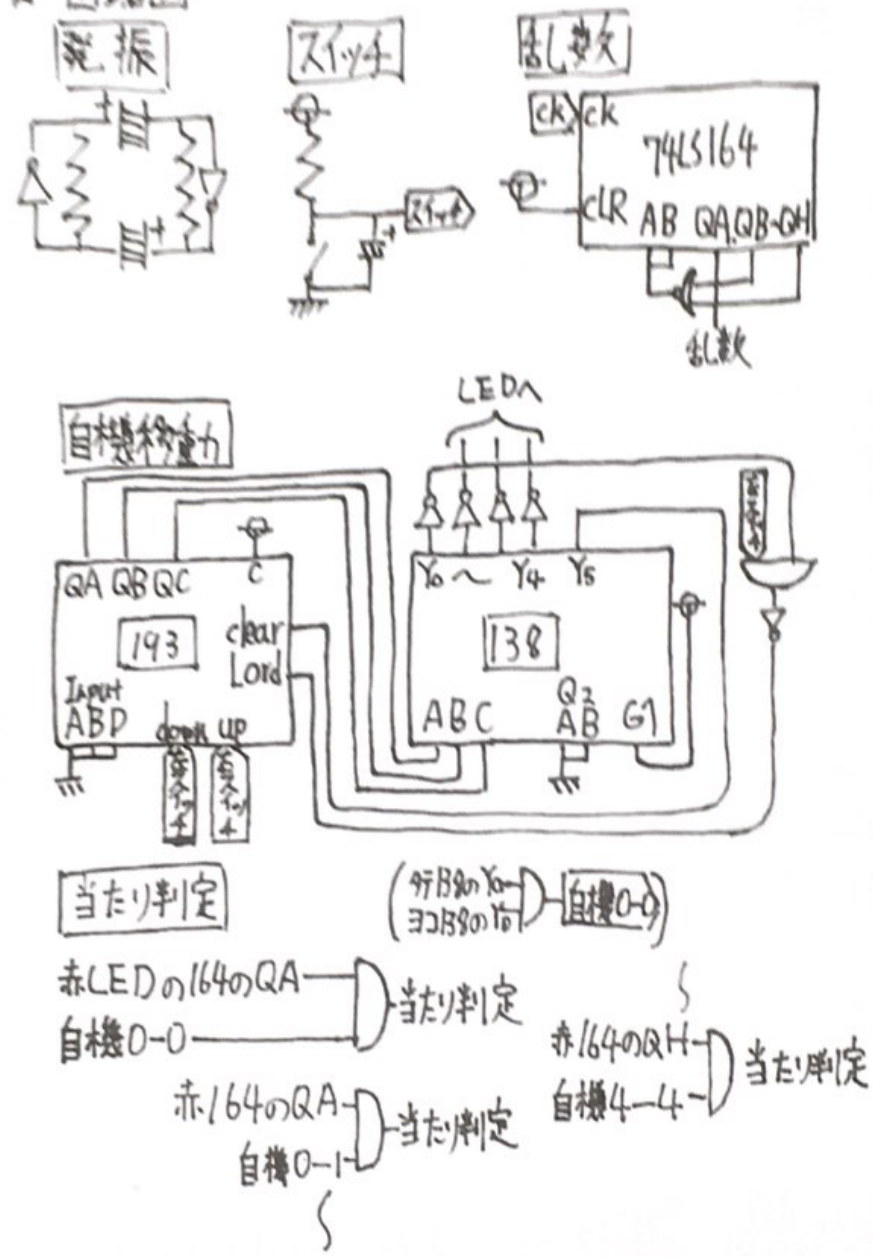
☆外観



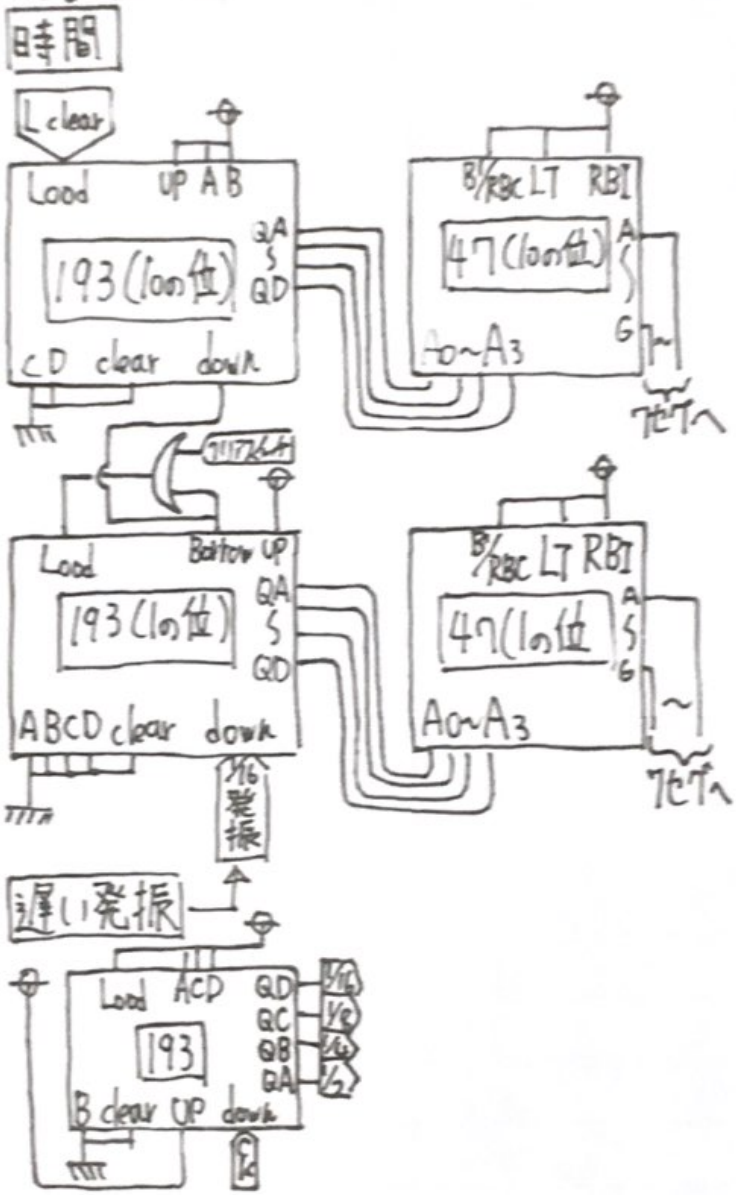
☆ストーリー
物無員Hは困惑していた。大切な電子部品の箱をひっくり返し、中身がちらばってしまったからだ！
一つ残らず探し出し、この事実を隠ぺいしよう。

☆内容
次々と出てくる目的地の赤LEDを目指して、自分の緑LEDを重力かしましょう。ゲームオーバーやクリアはなく、30秒以内のスコアが出ます。

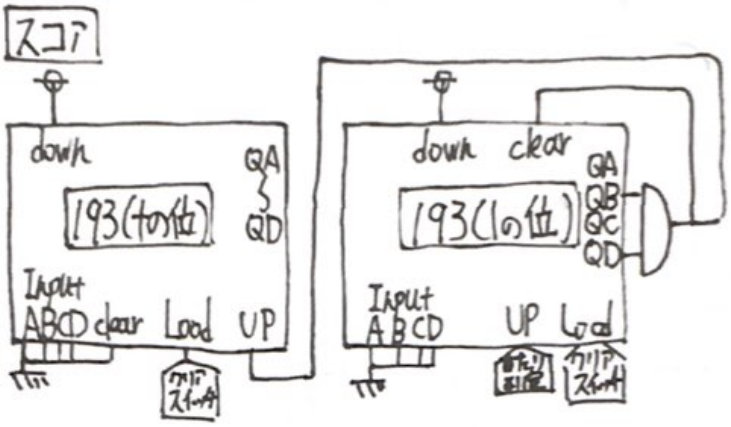
☆ 回路図


発振
スイッチ
スイッチ
乱数
ck
ck
74LS164
CLR
AB QA QB-QH
乱数
LEDへ
自機移動力
QA QB QC
C
193
clear
Load
Input
ABD
down
up
スイッチ
スイッチ
Y0 ～ Y4
Y5
138
ABC
Q2
AB
G1
当たり判定
分子B8のY0
ヨコB8のY0
自機0-0
赤LEDの164のQA
自機0-0
当たり判定
赤164のQA
自機0-1
当たり判定
赤164のQH
自機4-4
当たり判定

☆回路図(つづき)


時間
L clear
Load
UP A B
193(100位)
QA
S
QD
CD clear down
B/RBC LT RBI
47(100位)
A
S
G
A0~A3
7セグへ
Load
Bottom UP
193(1の位)
QA
S
QD
ABCD clear down
B/RBC LT RBI
47(1の位)
A
S
G
A0~A3
7セグへ
1/16
発振
遅い発振
Load ACD
193
QD
QC
QB
QA
B clear UP down

☆ 回路図(つづき)



☆感想

製作開始時に比べればはんだ付けがだいぶ、きれいになったと思います。嬉しいです。

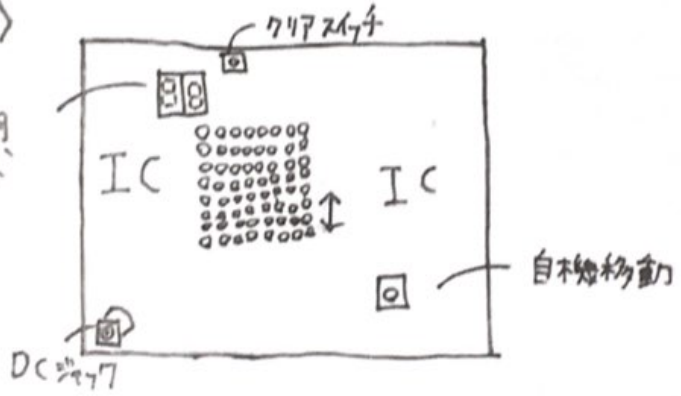
Escape From The Eyes

製作者：M2 永谷大成
[illegible]：H1 高角さん
十勇かが、物語の皆様

ルール：8×8のLEDの横から赤い線が流れてくるので、押すと上昇するボタン1つでよけます。生きのびた時間が得点になります。

ストーリー：鳥が人々の監視から逃れようとします。
しかし、このゲームには、終わりがありません。
このことは、いくら元気だって逃れる事のできない鳥の運命を表しています。

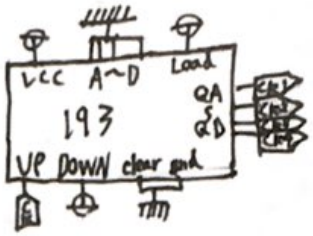
〈外観〉



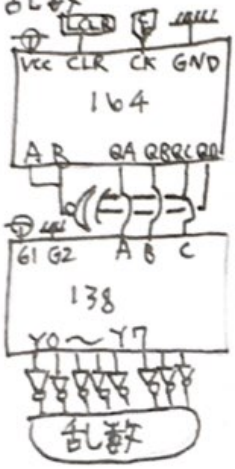
使用した部品
- 74LS IC
- 2色LED
- コンデンサー
- タクトスイッチ
- 抵抗
- DCジャック

回路図

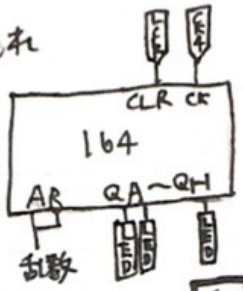
遅い発振



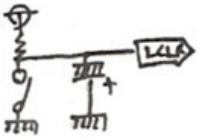
乱数



玉流れ

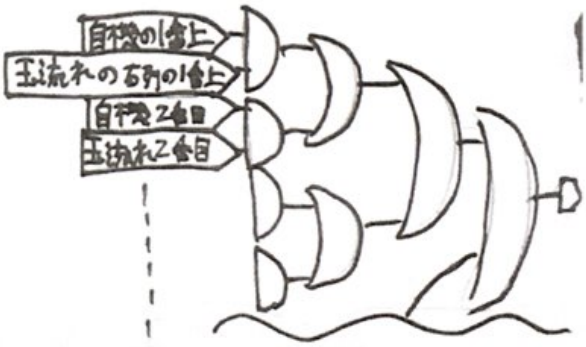


クリアスイッチ

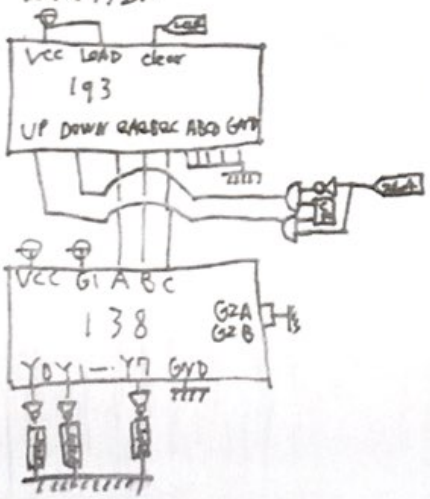


<感想>
この製作を通して、工作や、電子回路に対する理解を深めることができ、本当によかったです。
回路図設計者さん、協力してくださった仲間のみなさん ありがとうございました

当たり判定



自機移動



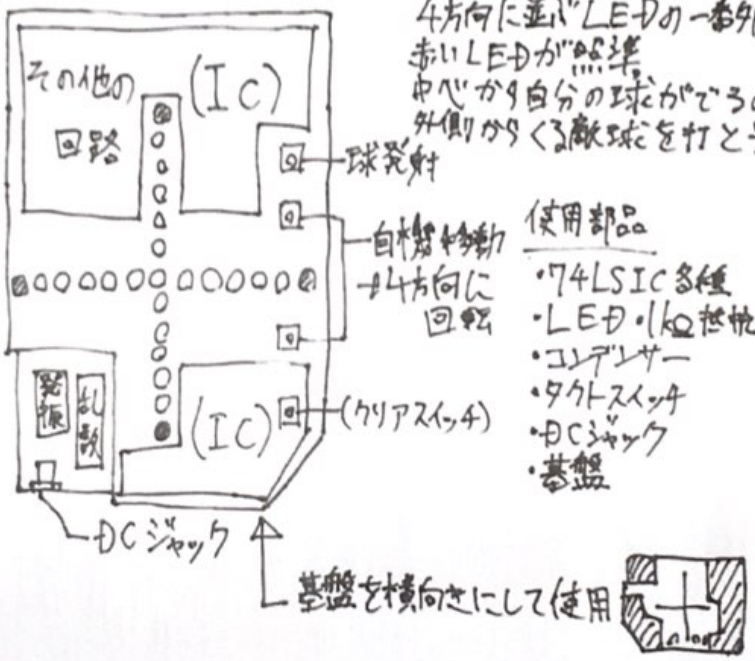
製作物を守り抜け!!

製作 M2 大谷
回路図設計 H1 高角さん

ストーリー

文化祭直前の物大。
某K学部員は物理部の展示大賞獲得を
阻止するため、物部員の製作物を破壊しに
やってきた！文化祭の展示を邪魔させるわけには
いかない！我々は文明の利器(銃)を使って
難しいくるK学部員を打ち倒し、製作物を守り抜けよう。

概要



ルール

4方向に並ぶLEDの一番外の
赤いLEDが照準。
中央から自分の球がでるので、
外側からくる敵球を打とう。

使用部品

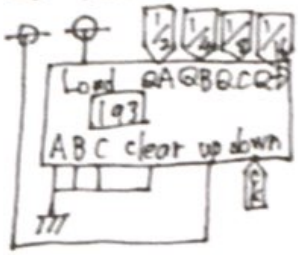
- 74LSIC 各種
- LED・1kΩ抵抗
- コンデンサー
- タクトスイッチ
- DCジャック
- 基盤

回路図

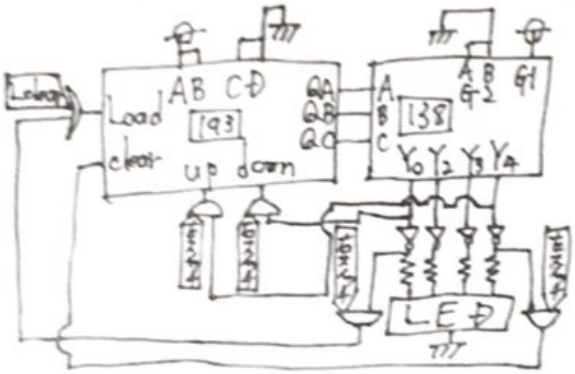
・発振

・乱数

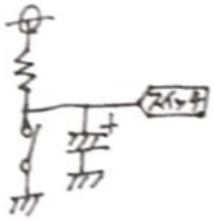
・遅い発振



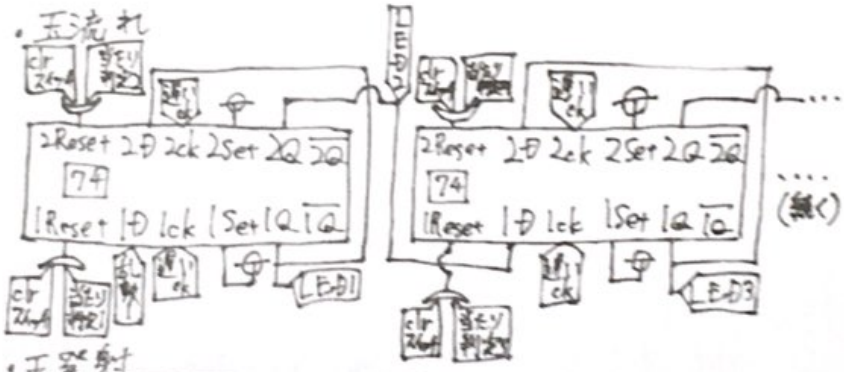
・四本機移動



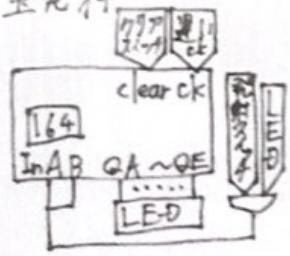
・スイッチ



・玉流れ



・玉発射



感想

途中何度も接触不良や誤のバグにぶつかってしまった。最終的にはだいぶ回路についても知れたし、はんだづけもうまくなったと思う。

動きを止めるな

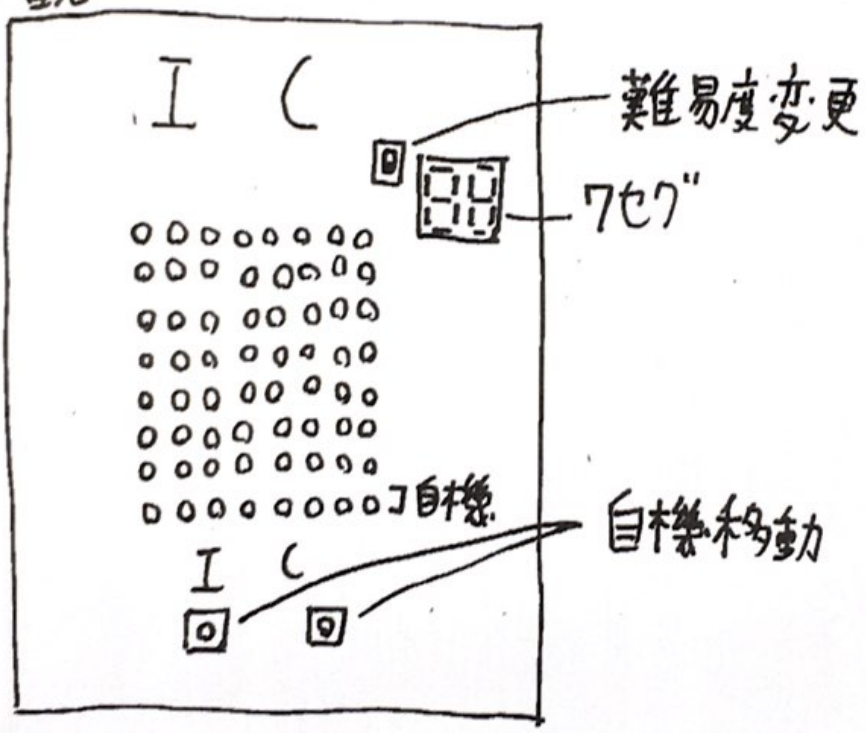
制作
M2.鳥内
回路図設計
H1 池田さん

ストーリー
同輩がTRPGに誘ってきた!
無視して逃げよう!

使用した部品

74LSIC	7セグ
コンデンサ	LED(赤,2色)
タクトスイッチ	スライドスイッチ
1kΩ抵抗	

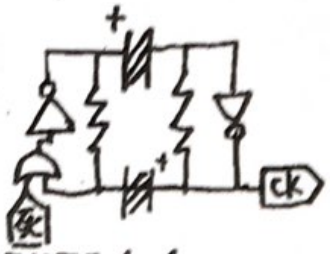
外観



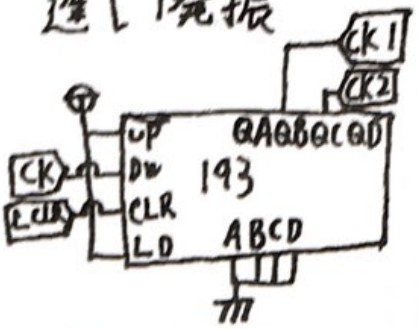
内容
上から降ってくる敵を避けるゲームです。
難易度を変更できます。

回路図

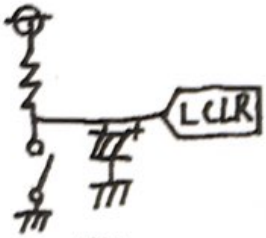
発振



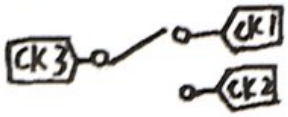
遅い発振



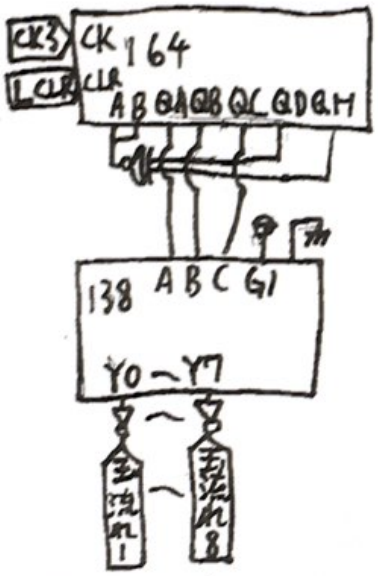
クリアスイッチ



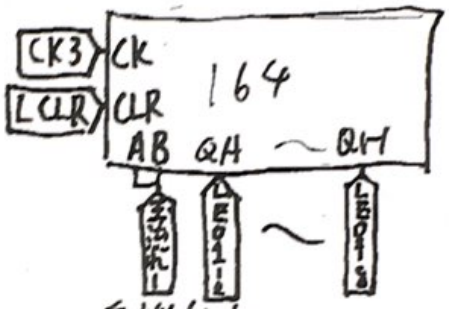
難易度変更スイッチ



乱数

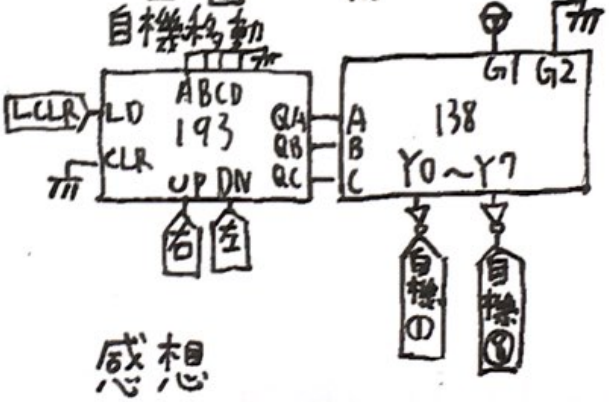


玉流れ



X8

自機移動



当たり判定

感想

中1ゲーの制作を通して汎用ロジックICの仕組みや様々な部品の仕組みをある程度理解でき、はんだ付けも入部した頃より大分上達しました。

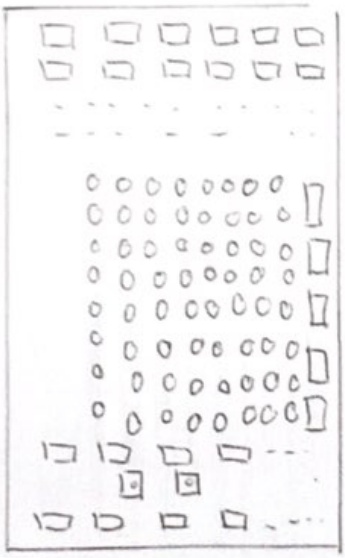
マルス算

制作者：M2 榎本 大智
回路図設計者：H1 池田さん
協力：物無の皆様

1. ストーリー

生きていれば無数の弾幕から身を守らなければいけない時もありますよね。
身を守る手段として最も優れているのは避けることなんですよ。
偉いんも当たらなければどうということはないって言ってました。
???「1人で避け続ければいけるか？」

2. ルール・外観



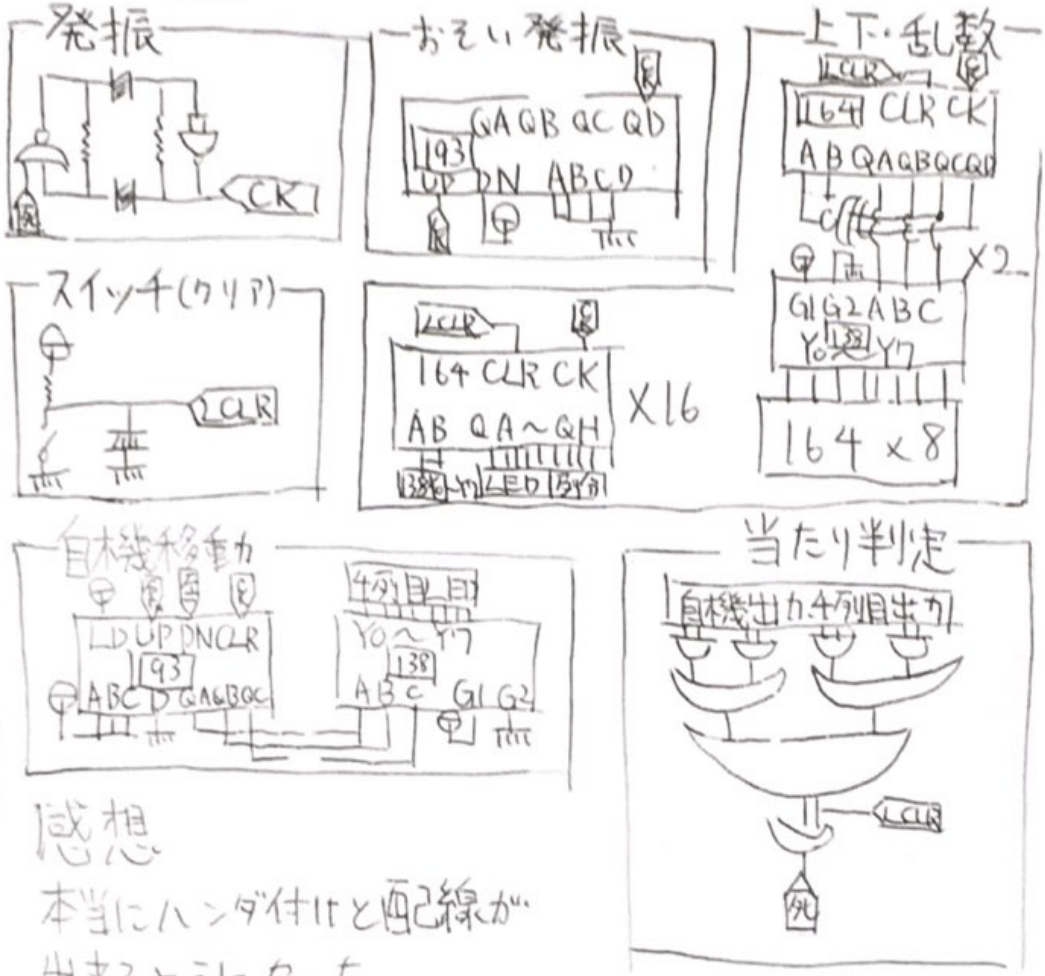
このゲームは上下から来るLED1つの弾を4列目の動かせるLEDを動かして避けるゲームです。

当たらなければ、生き続けられます。

3. 使用した部品

- 74LS IC
- LED(3色.2色)
- タクトスイッチ
- 抵抗
- コンデンサー

4.回路図



感想

本当にハンダ付けと配線が出来るようになった。

最初は配線の先もむけなかったので中1ゲーをやって良かったと思う。

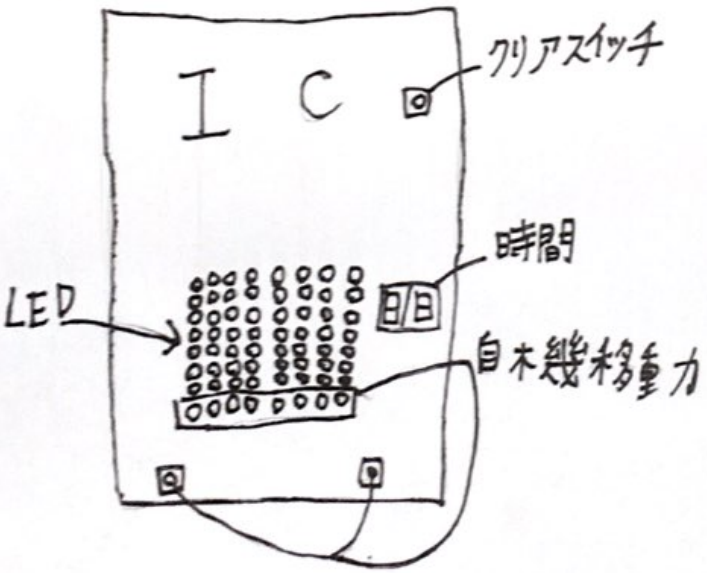
避けろメロス！

制作者
M2 秋田
回路図設計者
H1 海野さん
協力
物無の皆様

ルール
8×8のLEDの上から緑色の玉が流れてきます。
赤い玉を動かしそれを避けてください。
右のスイッチをおすと右に、左のスイッチをおすと左に動き、はじから反対のはじにも移動できます。
もし玉に当たるとその場で止まります。右上のスイッチを押すとリセットされ初めからできます。

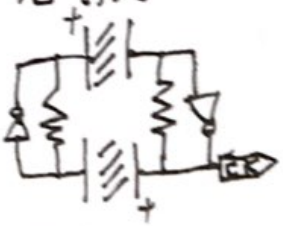
ストーリー
王の暗殺に失敗したメロスは、妹の結婚式に行くため、兵士をよけて逃げだそうとしました。

外観



回路図

発振

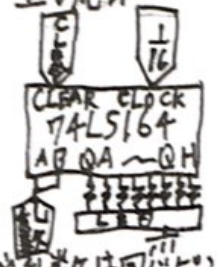


乱数

遅い発振

クリアスイッチ

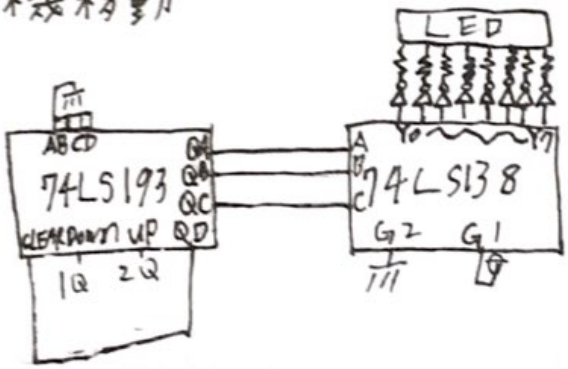
玉流れ



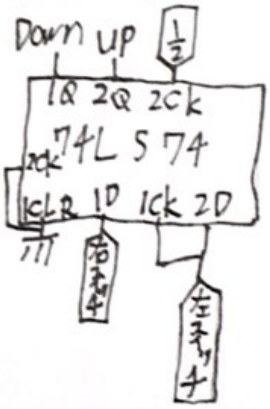
× 8

※乱数は同じピンからとらない。

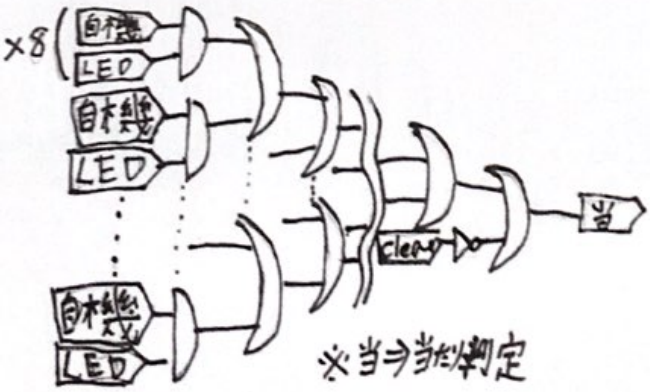
自機移動



チャタリング防止



当たり判定



※当⇒当たり判定

LEDキューブ

【製作】H1中村　H2もときさん　【協力】物無の皆様

・ストーリー

冬のある日
N「もときさ～ん、僕、LEDキューブが作りたいんですよ～」
M「いいね！共同製作にしよう。」
N「ありがとうございます！」

M「じゃあ僕はプログラミングするから本体作っておいてよ。」
N「わかりました！」

一ヵ月後
N「本体完成しました！」
M「自分の製作が忙しくて、まだプログラミングしてないからやっておいてよ。」
N「わかりました！」

一週間後
N「プログラミングできました！」
M「もう完成じゃん」

なんやかんやで数か月後
M「しばらく動かしてないからバグってないから確認しておいて。」
N「わかりました。」

ある日の例会にて
M「皆さん回路図集を書いてください」

その後
M「LEDキューブの回路図は僕がかいておくよ」
N「ありがとうございます。」

しばらくして
M「やっぱ回路図かいてよ」
N「えぇ」
M「特別に期限延ばしてあげるから。」
N「わかりました」

最近
M「まだ書いてないの？」
N「すみません。」
M「今年は製作物の数が少ないからその分沢山書いておいてよ。」
N「少しは手伝ってくださいよ」
M「やだ。」

今日(深夜1時)
N「(一人で全部書き終わった…)」

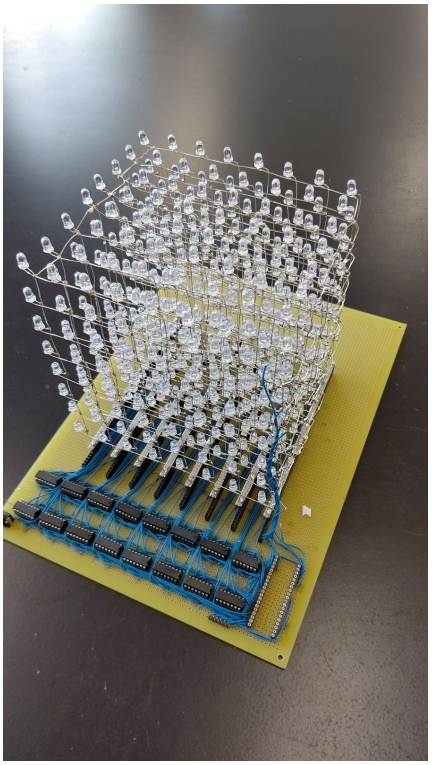
これを読んだあなた。どうかMさんの暴挙の真相を暴いてください。
それだけが私の望みです。

中村陽

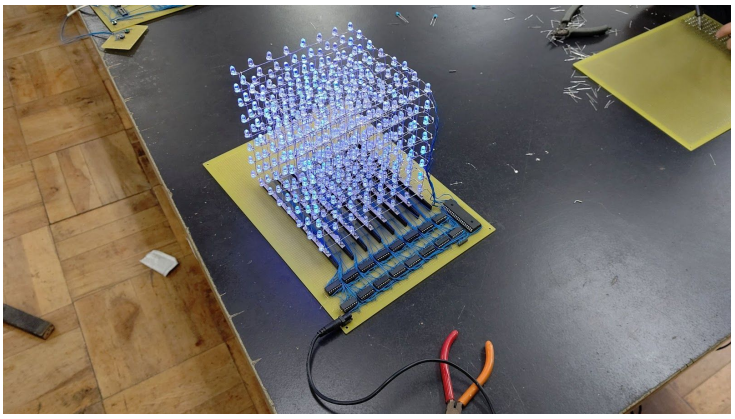
ストーリー長くてごめんなさい。ページ数稼ぎです。

・概要

LEDを8×8×8で立方体の形に並べてPICというICを使い様々なパターンで光らせることができます。



←外観↓

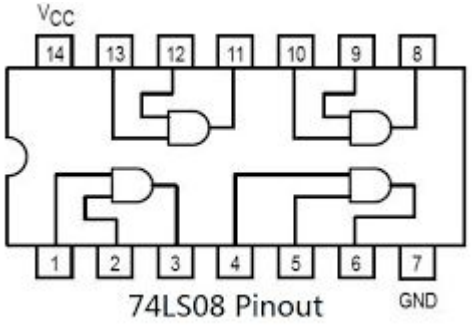


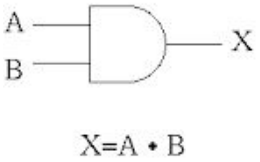
・機構

部品の説明

PIC16F1939…プログラミングをすることができるICです。

74LS08…ANDという働きをしてくれるICです。





動作表

A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

LED…光ります。青色です。

仕組みの説明

PIC1つだけでは512個ものLEDを制御することは出来ないため、74LS08というICを使い、段階的にLEDを制御出来るようにしてます。

プログラミングの説明

少しだけ紹介させていただきます。

```
int cube[8][8][8]={{{0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0}},

{{0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0}},

{{0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0}},

{{0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0}},

{{0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0}},

{{0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0}},

{{0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0}},

{{0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0}};
```

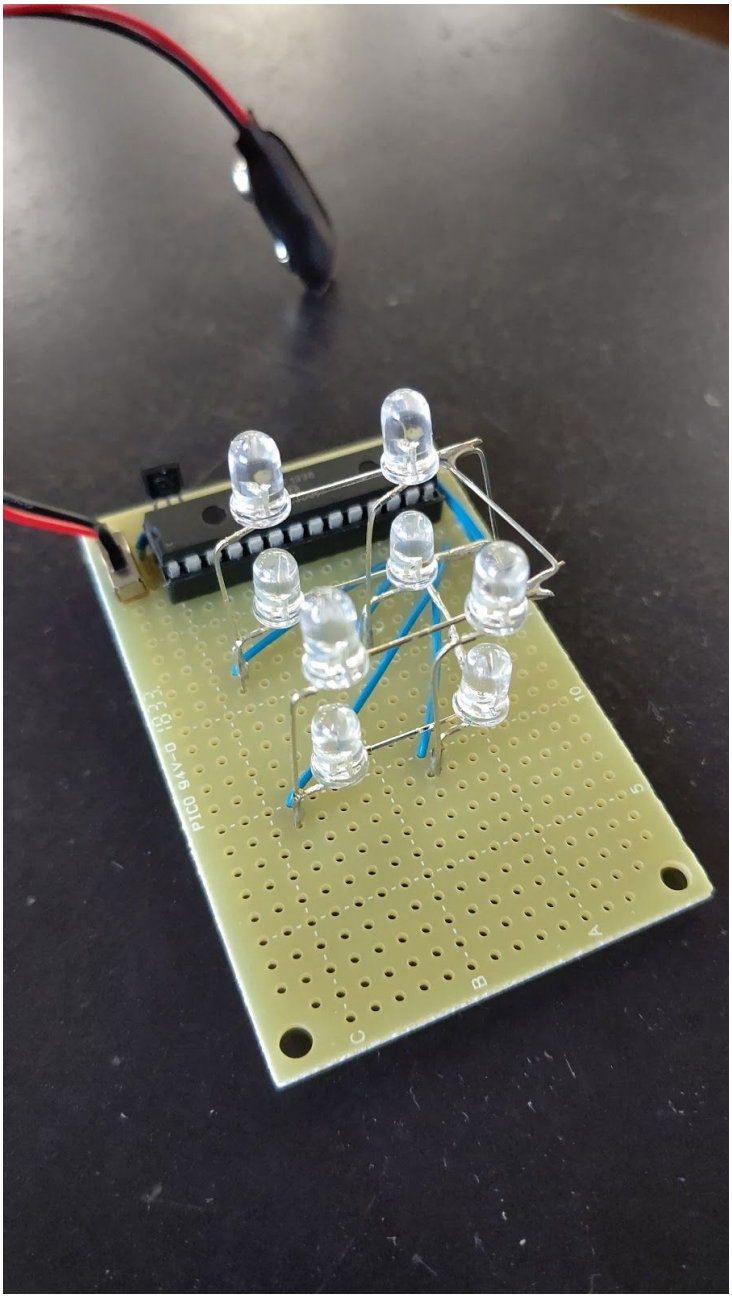
←だいぶ見づらくなってしまっていますがこれは三次元配列と呼ばれるもので、任意の0と書かれている部分を1に置き換えることでLEDを光らせることができます。

```
void display(){
        for(int z=0;z<8;z++){
        for(int x=0;x<8;x++){
        for(int y=0;y<8;y++){
        if(cube[z][y][x]==1){
           PORTB=0b00000001<<x;
           PORTD=0b00000001<<y;
           PORTC=255^1<<z;
           __delay_us(1);
        }else{
           PORTB=0b00000000;
           PORTD=0b00000000;
           PORTC=0b11111111;
        }
        }
        }
}
}
```

↑　これはfor文の三重ループと呼ばれるもので左の三次元配列の1になっている部分を探すことでそれに対応しているLEDを実際に光らせることができます。

・おまけ

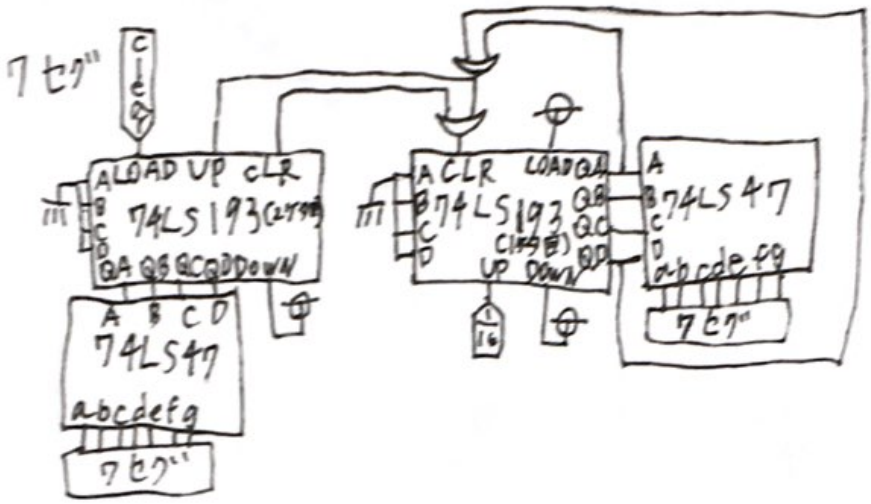
小さいLEDキューブを作りました(^.^)名前はLEDキューブminiです。



画像は全然miniじゃないですね。

・感想

~~共同製作とは何かを考えさせられた製作でした。~~
512個ものLEDを空中ではんだ付けしたのでかなり疲れました。接触不良もなく無事に動いてくれるようになって良かったです。(中村)



感想

この製作をして、はんだ付けが元々下手だったのがすこし上手くなった。
また、ICについて理解がふかまりよかったと思った。