

はじめに

昨年の回路図集では「コンピュータの処理能力向上は集積に留まらない新たな段階に進みつつある」とありましたが、最近ではコンピュータの能力向上と同時にインターネットに繋ぐことのできる製品が増え、IoTの時代が近づきつつあります。

我が部でも近年のネットワーク社会に遅れを取らぬよう、iPhoneのアプリを使ったり、物無をインターネットを通じて発信していかうとしたりしています。

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・製作したもののについての詳細を記したものです。ただし、未完成のものも含まれており、必ずしも本書の通りに作っても上手く動作する保証はありませんのであらかじめ御了承下さい。

会計：H2 渡辺悠太

Index

- 1 はじめに
- 2 目次

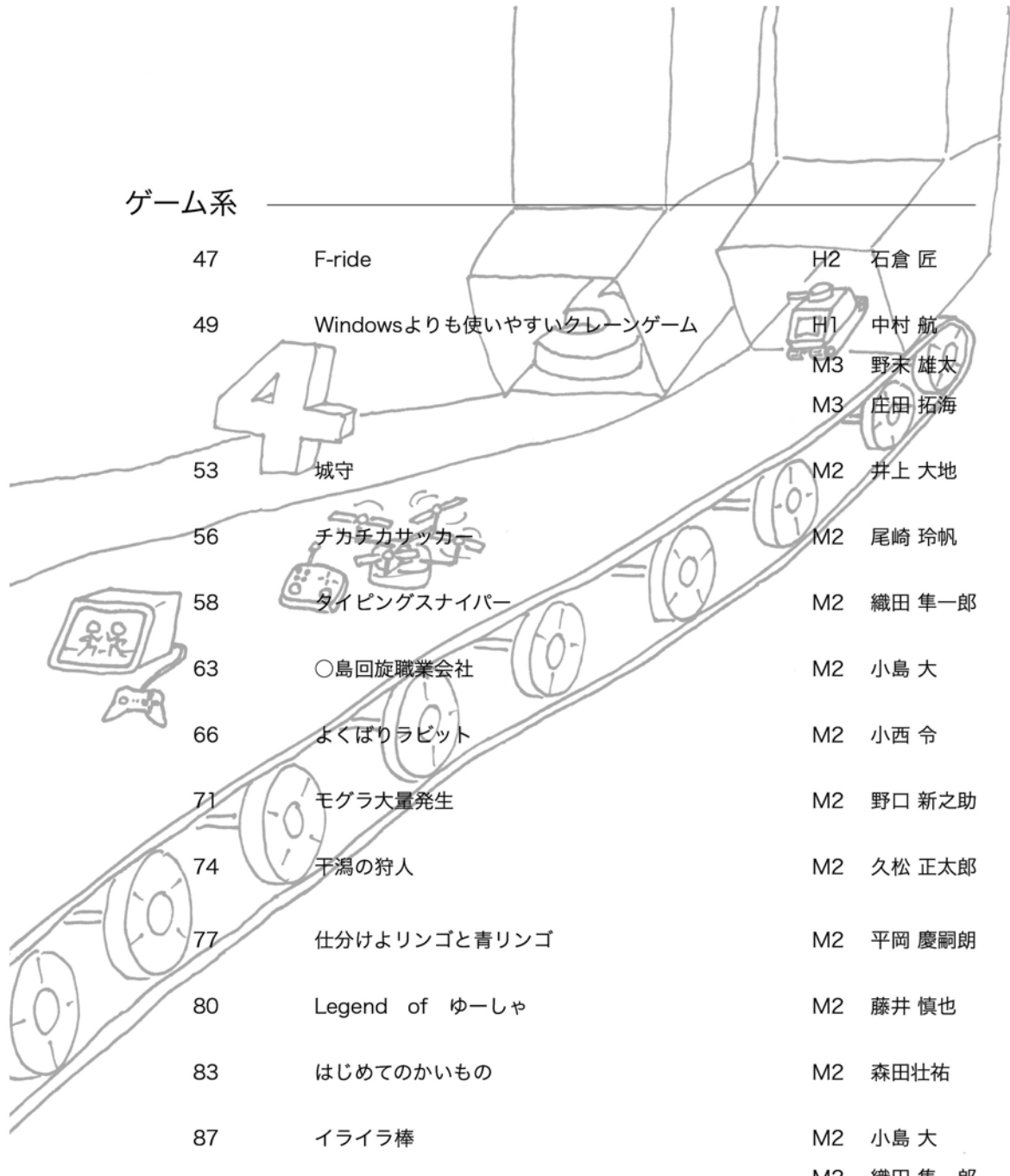
コンピューター系

- 5 Melting Internet H1 磯村 隆正

ロボット系

- 10 Quick Programming 〜クイプロ〜 H2 宮嶋 優大
M3 高橋 秀明
- 15 産ミガメ H2 渡辺 悠太
H1 和田 直輝
M3 飯島 勇人
- 22 small maroon号 H2 山本 涼太郎
H1 佐藤 慧斗
- 26 Wake-Up-Snake! H2 横尾 幸丸
- 29 階段登ルンです H2 佐伯 大地
- 32 ホバーが奏でる恋のキャンパス H2 市川 詠亮
H1 小栗 秀介
- 34 Drone Tank MK.1 H1 関根 史人
- 41 the just lighting only of ball H1 福島 弘也
- 43 ROBOT M2 野口 新之助
M2 藤井 慎也

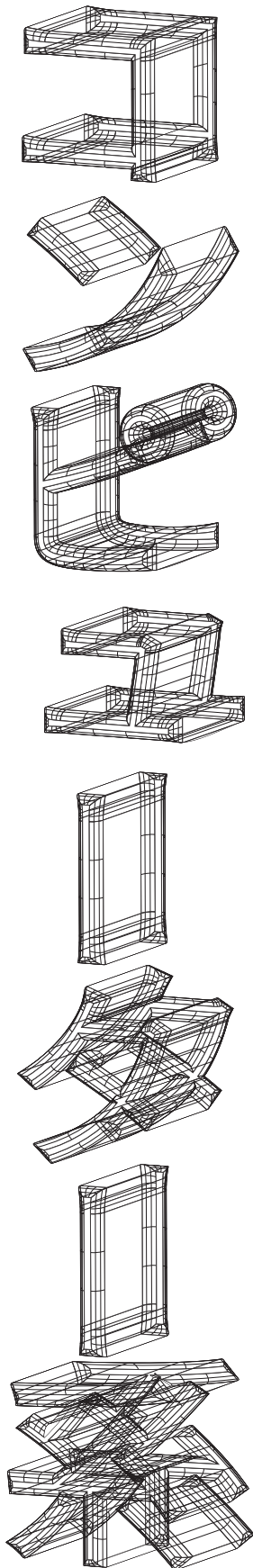
ゲーム系



47	F-ride	H2	石倉 匠
49	Windowsよりも使いやすいクレーンゲーム	H1	中村 航
		M3	野末 雄太
		M3	庄田 拓海
53	城守	M2	井上 大地
56	チカチカサッカー	M2	尾崎 玲帆
58	タイピングスナイパー	M2	織田 隼一郎
63	○島回旋職業会社	M2	小島 大
66	よくばりラビット	M2	小西 令
71	モグラ大量発生	M2	野口 新之助
74	干潟の狩人	M2	久松 正太郎
77	仕分けよリングと青リング	M2	平岡 慶嗣朗
80	Legend of ゆーしゃ	M2	藤井 慎也
83	はじめてのかいもの	M2	森田 壮祐
87	イライラ棒	M2	小島 大
		M2	織田 隼一郎

売り物

90	Sodom and Gomorrah	M3	高橋 秀明
91	チカチカ	M3	飯島 勇人



the latest
technology of
micro computer
and
related computings

Melting Internet

製作： H1 磯村隆正

はじめに

巷では「Internet of Things（モノのインターネット）」というものが少し前から流行り、実際に製品としてそのようなコンセプトで作られたものは多く存在しています。例えば、スマホで外から操作できるエアコンなどがわかりやすい例でしょう。

この Internet of Things（以下 IoT と表記）とは、文字通り「モノのインターネット」であり、「ヒトのインターネット」ではないということが肝要です。簡単に言うと、これまでのメールやホームページ、SNS のように、人から人に情報を伝えるのではなく、人から物、物から人、あるいは物から物へインターネットを介して情報を伝える技術、仕組みであると言えるでしょう。具体的には、製品に様々なセンサーを取り付け、製品自身が人の手を介さずインターネットで情報を送信したり、インターネットで受け取った情報に基づき、製品が自動で作業を実行するというシステムが主な内容でしょう。このシステムをあらゆるものに搭載することで、現実世界の物理的な情報をもっと便利に利用することができるのです。

しかし、実際に製品が登場しているといっても、現在この IoT を実現している製品システムはほとんどの家庭には存在していないというのが現状です。さらに、今ある IoT 的製品群は完璧に便利なものとは言えません。先ほど挙げたエアコンの例で言えば、まだそのような機能の付いている製品がそこまで多くない上に、付いていたとしても「どこでもリモコン」の域を出ていないものがほとんどです。

では、どうすればリモコン以上だと言えるのでしょうか。その答えの一つとして、端的に言えば、いかにしてコンピューターのやることを増やし、人間のやることを減らすかということを考える必要があると思います。もともと人間のやっていたことを機械が代わりにやるためには、ほとんどの場合、当然難しい処理が必要になります。そのためには、もっと多くの精密な情報をセンサーで収集し、もっと優れたアルゴリズムでコンピューターに判断させることが必要でしょう。

しかし、人間は未だに機械でもできる簡単なことを行っていることがあります。特にインターネットはまだ登場して間もない技術なので、デジタル化された多くの情報がありながら、その多くは人間が読んで理解しなければ使うことができません。

例えば、クックパッド等でネットからレシピを探してきて料理を作るという過程において、材料が何グラム必要かということを人間が理解して、秤などで計測する必要があります。しかし、材料の量は誰が見ても、誰が計っても同じで、そこに複雑な判断は必要ありません。当然コンピューターが行うことも容易でしょう。逆に新しいレシピを研究するときに、計測した量を記録して、自動でレシピを作ってくれるシステムもあれば素敵です。これらは電子秤をインターネットに接続したり、軽量カップなどが中身を自動で計測、送信してくれるようにすることで達成できるでしょう。

さらに、センサーは容易に計測することができるけれど、人間が知覚することがとても難しい情報もあります。地図アプリで使われるような GPS などがそれです。GPS を利用すれば、その人が世界のどこにいて、どの方向に向いているのかということを簡単にデジタルな情報として取得することができます。Google Maps などの地図アプリは、この技術を使って得た位置情報とインターネットで取得した地図情報をあわせて画面に表示します。こうすることで、人間が近くが目立つ建物を探したり、道の向かっている方向をコンパスで見たりする必要はなくなります。また、ルートや特定の場所を検索することも容易です。

その一方で、センサーで直接計測することは困難でも、人間の行動を観測することで簡単にわかる情報もあります。自動車や電車などのワイパーを操作したときの強さと位置情報を収集し、分析すれば、どこでどれだけ雨が降っているのかということがかなり正確にわかります。速度を計測すれば、渋滞の場所や運転の快適な道を分析することも可能でしょう。他にも、カメラ撮影の際の位置情報、撮影時刻などを集めれば、どこがどの時間帯で人気のあるスポットであるのか分析できます。さらに、色々な面でハードルは上がりますが、その被写体を解析すれば、そこが自然の美しい場所なのか、観光客で賑わう場所なのか、はたまたデートスポットなのかという情報化をすることができるでしょう。

このように、インターネットは「モノ」も利用することでよりデータを有効に使えると言えるでしょう。また今現在、そのように簡単に自動化できる状況というのは多々存在しているのです。このことを踏まえると、実際にどのようなものが真の IoT であるのか、ということが見えてくると思います。

概要

今回、IoT をテーマにして、インターネットを使ったものを作ろうと思います。文化祭の展示としてやるという前提なので、多くの人が参加しているという実感をできるだけ大切にしようと考えました。作るものは大きく分けて3つあります。

- ①製作物稼働状況の分析による展示混雑度測定システム
- ②位置情報を使った潜在展示来訪者測定システム
- ③持ち物の重さコレクション

まずは①の概要から説明していきます。読んで字のごとく、物理部無線班の文化祭展示がどれくらい混んでいるのかを示すものです。物無のウェブサイト上で誰でも見ることができます。

展示されているロボットやゲームのプレイのサイクル（ゲームの開始やクリアをしたタイミング）のデータを集めて、混雑度を予測して表示します。測定が簡単で、数が多く、プレイ率が高い（見てだけの人が少ない）であろう中一ゲー（大きい基盤に LED やスイッチなどがたくさんついているゲーム）を中心にデータを集めようと考えています。

②では、展示に関係のあるページにアクセスした人の位置情報をもとに、展示に来よう

と思っている人がどこにいるのか調べます。

まず、物無の文化祭ページやチカチカ順番待ち確認ページ、①のページへアクセスしたときに、どこからアクセスされたのかを調べます。今回、アクセスした端末の GPS や IP アドレス、Wi-Fi の接続などから現在位置を取得する Geolocation API というものを利用します。取得した位置情報は個人情報とは全く結びつけられませんし、無許可で取得することはできません。この情報は誰からでもアクセスできるようにするようなことは絶対にありませんし、文化祭が終われば全て破棄されるので安心してください。

③では、圧力センサーを利用して、お客さんの持ち物の重さを測定します。同じような重さのものや、重さのわかりやすいものを取得して、〜と同じ重さや、〜何個分のような形でその重さを表示したりします。さらに、それが実際何なのかを入力して、データの一部にすることができます。

解説

詳しい回路図やソースコードは物無ホームページや GitHub などに載せたいと思います (回路図集とはなんだったのか…)。

ですから、ここでは製作物で使用している技術を簡単に解説したいと思います。

Node.js

Node.js (以下 node と表記) とは、サーバーサイドの JavaScript 環境です。一口で説明するとなんだかよくわかりませんね。まず、JavaScript というのは今ではほとんどの Web ページで使われているもので、スクリプト言語 (プログラミング言語のようなもの) です。ほとんどの場合、ブラウザで実行されるもので、node のようにサーバーサイドで動作するものは多くありません。

node の一番の利点は、PHP などのサーバーサイド環境と違い、ノンブロッキング I/O であることでしょう。ノンブロッキング I/O とは、時間のかかる要求を行った際に、結果を待たずに次の処理に移るという仕組みです。普通は結果が帰ってくるまで処理を停止しますが、それを行わないので、重い入出力処理を効率よく非同期で行うことができます。これを使うことで、多くのサーバーリクエストを処理し、リアルタイムで様々な入出力が可能になります。

また、多くの便利なパッケージが提供されていて、それらを npm というパッケージマネージャーで簡単に扱うことができるのも魅力です。

MongoDB

MongoDB とは、オープンソースのドキュメント指向データベースで、C++ で開発されています。複雑な検索が可能ながら、優れたパフォーマンスが実現できることや、様々な環境で高いパフォーマンスが実現できるドライバが開発元から提供されています。そのため、多くのアクセスをリアルタイムで処理する場合に有効です。

Socket.IO

Socket.IO とは、Node.js 用のサーバー側と、ブラウザ側がセットになった JavaScript ライブラリです。全ての環境でリアルタイム通信ができるように、様々なリアルタイム Web の技術を一様に使えるようにするために作られています。リアルタイムで Web 通信を行う方法は様々で、普段ホームページにアクセスするときに使う HTTP 通信を利用する古典的なものから、Adobe Flash などのプラグインを必要とするもの、WebSocket などの新しい規格など様々で、一つ一つ実装するのはとても大変です。Socket.IO は自動でその中で最適な方法を選択してくれます。

Arduino

もう既にこの単語を知っているという方も多いのではないのでしょうか。この回路図集の中でも見たかもしれません（僕も去年少し触れていますね）。

物無の中では、ハードウェア新勢力として PIC 派の憎悪の対象（？）のようになっているようです。しかし、Arduino を理解するためにも、そもそも PIC と相対するものではないということを認識しておく必要があるでしょう。

そもそもの PIC と Arduino の違いとして、PIC は Microchip Technology Inc. の製造するマイクロコントローラー（以下マイコンと表記）のシリーズのことであるのに対し、Arduino はオープンソースのプラットフォーム、またはその設計に基づいて開発されたマイコンのことです。Arduino はハードウェアの設計や開発環境は全て、オープンソースで提供されています。オープンソースなので、もちろん誰でも Arduino を作ることができるし、無料で設計情報や IDE（統合開発環境）を手に入れることができます。

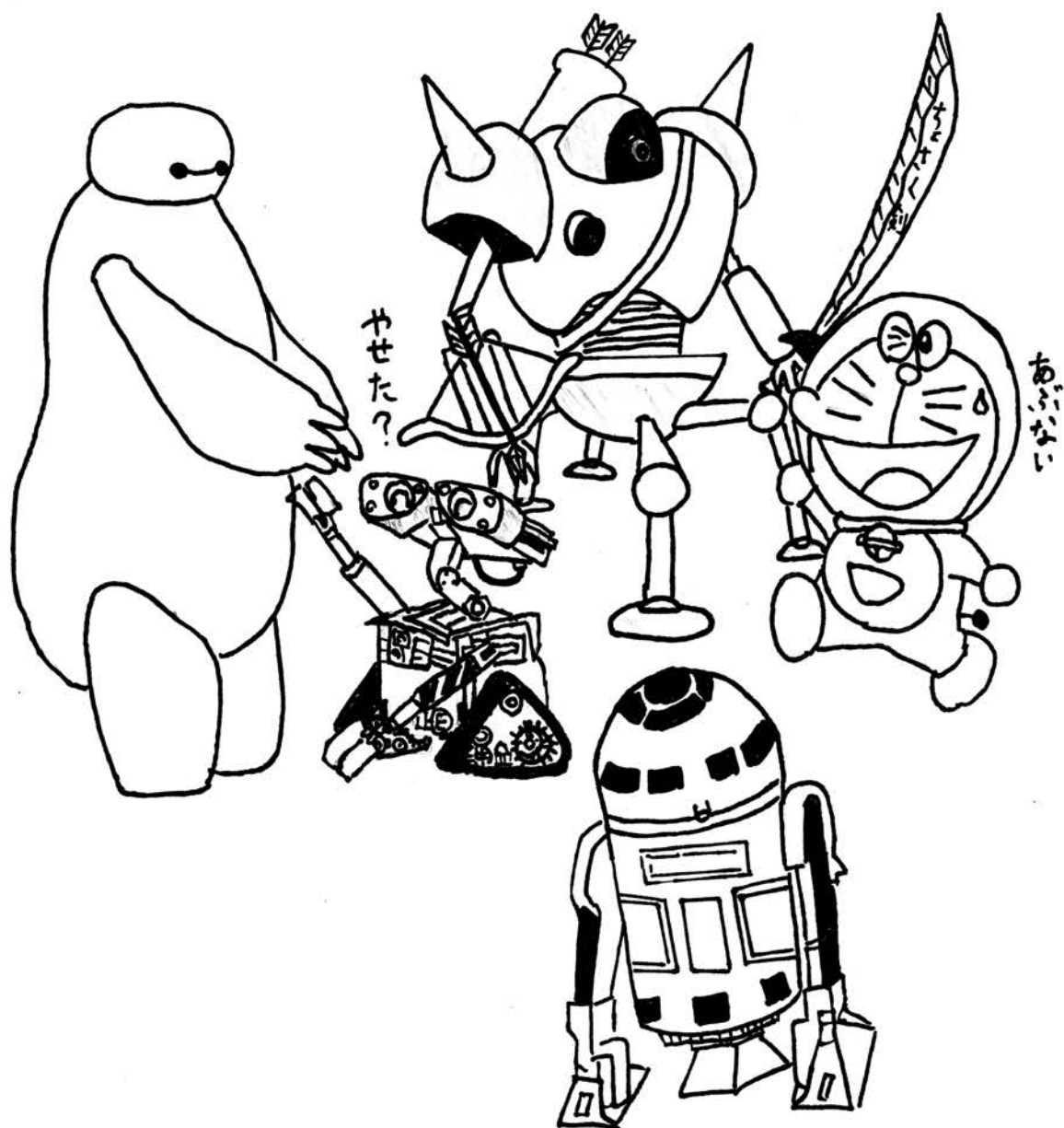
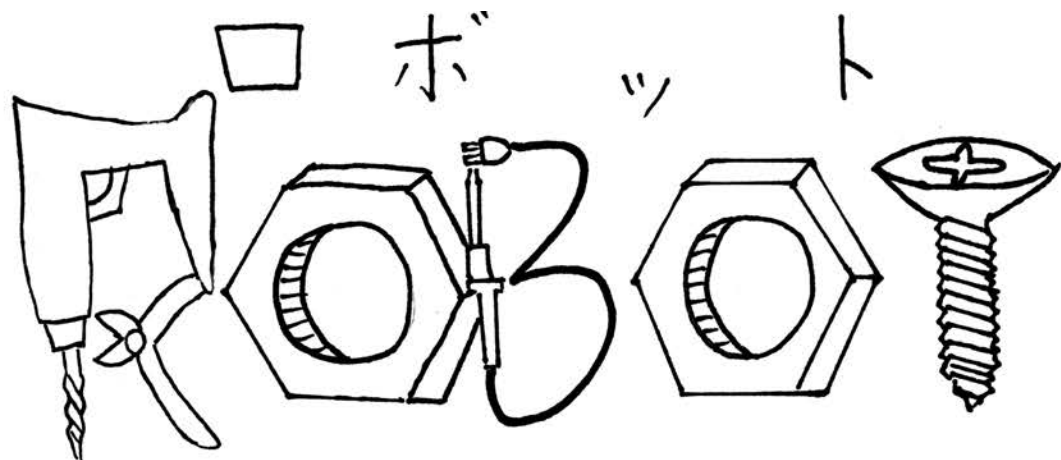
このように、「Arduino」はそもそも一定のマイコンシリーズを指す言葉ではないのです。だから、PIC と比べるなんていう行為は端から間違っているのです。むしろ最近の物無内の流行りとしては mbed（ARM のワンボードマイコン）の対抗馬でしょう（といっても ARM チップ搭載の Arduino もあるわけですが）。

最近では Arduino として Intel が Galileo や Edisonなどを投入しています。これらでは Intel の Atom プロセッサなどの高い性能を享受したり、関数一つで Arduino のコードの中から Linux のコードを実行することなどが可能です。

感想

部内ではほとんど挑戦している人のいない分野への挑戦で、何をやっているのかよくわからないと飽きるほど言われました。自分自身も殆ど Web の知識がなく、手探りの状態だったのですが、漸くなんとかわかるようになってきたという感じです。そういう道のプロの方がこの文章を読んでいるなら是非色々教えて頂きたいものです。

ここで書いた物無ホームページには butumu.com からアクセスできます。今後も継続的に更新されていくことが予想されますのでぜひチェックしてみてください。



Quick Programming

～クイプロ～

共同製作者 -Partner

H'2 宮嶋 優大

M'3 高橋 秀明



ストーリー -Story

“EMERGENCY!! EMERGENCY!!”

突然所内に警報音が鳴り響いた。

“危険物発見 直ちに避難せよ”

どうやらすぐに逃げないと死んでしまうらしい。

もうだめだ...

と諦めかけた時、iPhoneの通知が鳴った。

“何だろう？”

そこに表示されていたのはカメラだった。

そこでいつも持ち歩いている極秘資料の暗号を写した。

すると突然後方からすさまじい音が聞こえ、僕は気を失った。

概要 -Summary

このシステムではiPhoneを用いてQRコードで書かれたプログラムを読み取り、赤外線
で通信しロボットを動かす。おおまかなシステム概要は次のようになっている。

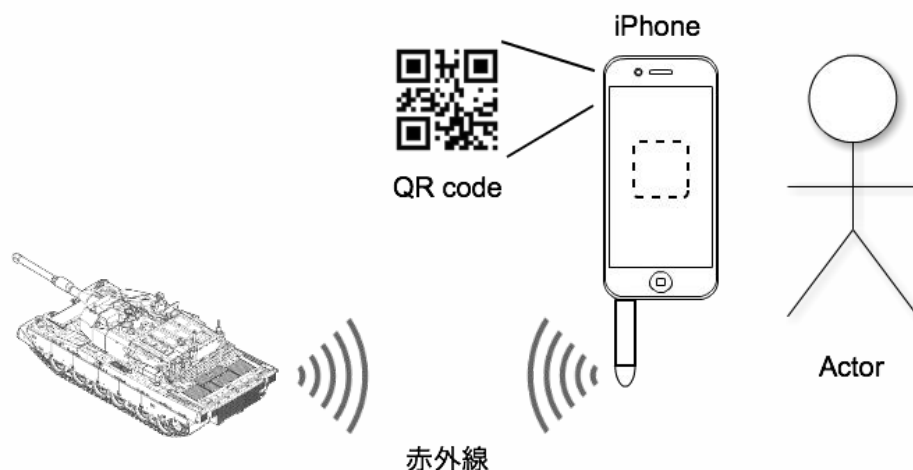


図1 モデル図

具体的にはQRコード、iPhone、iPhoneAPP、赤外線送信装置、赤外線受信及びデバイス
制御装置、デバイス本体から成り立っている。

では順を追って見ていこう。

QR-Code

QRコードとはQuick Response Codeの略で、小さな正方形の点を縦横同じ数だけ並べたマトリックス型二次元コードである。高速読み取りができるように開発された。3隅の四角い切り出しシンボルが特徴的だ。近年では広告や遊園地の入場券、航空券などに使われていて、誰しも一度は必ず見たことがあるだろう。バーコードは横方向にしか情報を持たないのに対し、QRコードは縦横に情報を持つ。



図2 QRコード

QRコードは意外と高性能である。格納できる情報量が多く、数字だけでなく英字や漢字など多言語のデータも格納できる。また、誤り訂正機能が付いているため、コードの一部に汚れや破損があってもデータの復元が可能である。最大30%まで訂正可能なのだ。

このシステムではこのQRコードを使い、ロボットのアルゴリズムの組み立てを行う。

簡単に言うと、ロボットをどう動かすか決めることができるのである。このQRコードを変えることによって、ロボットの動作が変わる。QRコードには動作コード、動作の継続秒数（または動作回数）を表す 1 Byte のデータの集合が書かれている。

今回、アルゴリズムの組み立て方法にQRコードを選んだ理由はアルゴリズムの組み立てを物質化し、実際に手にとって変えられるようにするためである。従来では、LEGO社のMindstormsのようにパソコンを使ってアルゴリズムの組み立てをパソコンで行いロボットを動かすというものがあったが、パソコンの知識が必要だったり、操作が難しかったりして対象年齢が高かった。（普通にプログラミングするよりかはよっぽど簡単だが）そこで、アルゴリズムの組み立てを物質化することにより、より直感的に、より容易にアルゴリズムを組み立てることができるのだ。言ってしまうえば、子供がパズルを組み立てるのと同じ要領だ。ただ、パズルの組み立て方は無限大である。100回組み立てれば、100通りの動作をする。これは幼い子供達の発想力を膨らませること間違い無しである。

iPhone

これは持っている人も多いでしょう。Apple社のiPhoneである。展示ではiPhone5を使っているがiOS8.0以上の端末であればどんなiPhoneでも動く。

これを使おうと思った理由は身近にあるものを使ってロボットを操作したいと思ったからだ。

iPhone App

iPhoneアプリとはiPhoneで動くアプリケーションのことである。AppStoreでは100万点を超えるAppが販売/配布されている。iPhoneアプリの開発にはいくつか必要なものがある。それは「Mac」、「Xcode」、「Apple Developer登録」、「開発言語の知識」、この四つである。

「Mac」、これはiPhoneアプリの唯一の開発ソフト、Xcodeに付随する。XcodeがMac専用のソフトのため必然的にMacがなければ開発ができないのだ。これがiPhoneアプリ開発における一番の壁であるだろう。Macはご存知の通り高性能が故に値段が高い。Macがないから開発を諦めたという人も少なくはないと思う。

「Xcode」、これは先ほど述べたが唯一のiPhoneアプリの開発ソフトである。MacStoreで無償で配布されているため、問題はないだろう。Xcodeはプログラムのコードを書くためのエディタや部品をドラッグ&ドロップするだけでアプリの見た目を作ることができる

インターフェースビルダー、自分が書いたプログラムをコンピュータでもわかる下級言語に変換するコンパイラ、プログラムの動作に問題がないかプログラムの実行中に動作を止めて中の状態を確認できるデバッガが入っている。また、実機がなくても試せるiOS Simulatorや二つのファイルを比較できるfilemergeなどのソフトが標準で付属している。Apple社ならではのグラフィックでとても見やすく、使いやすくなっている。iPhoneアプリやMacアプリの開発に特化しているが、多様なプログラミング言語のソースコードをコンパイルできるため普通にプログラミングエディタとしても十二分に使える。



図3 Xcode

「Apple Developer登録」、これは自分が作ったiPhoneアプリを実機で動かすためにはApple Developer登録が必要なのである。そして登録するためには年間参加費\$99、日本円にして12744円（2015年4月現在、ちなみに2015年1月までは8424円だった）を支払わなければならない。iPhoneアプリ開発における二番目の壁であるだろう。これをすると同時に自分が作ったアプリをAppStoreで配布または販売することができる。また、アプリの販売で出た利益や広告収入の一部はAppleから受け取ることができる。

「開発言語の知識」、当然ながらiPhoneアプリの開発にはプログラミングの知識が必要だ。iPhoneアプリの開発言語はObjective-Cとその後継位置づけられるSwiftの二種類がある。どちらもオブジェクト指向であるため、C言語に比べ記述する内容が少し複雑である。Swiftの詳しい記述の仕方はネットや本を見て調べてほしい。

では、このシステムで用いているiPhoneアプリの詳細について説明する。このiPhoneアプリでは「QRコードの読み取り」、「アラート表示」、「コード変換」、「赤外線でのコードの送信」、この四つのことを行っている。

「QRコードの読み取り」はカメラを使いQRコードを検索し、認識したら10進数のコードに変換する。どの角度でも読み取れるようにアルゴリズムが組まれている。

「アラート表示」はQRコードを認識すると、選択肢がついたアラートを表示する。送信するデータが正しかったら送信ボタンを押し、次のステップに移る。キャンセルを押すと、もう一度QRコードの読み取りを行う。

「コード変換」は読み取ったコードを赤外線で送れるように変換し、それを音声ファイルにする。作成する音声ファイルはwavファイルである。詳しいことは次節、赤外線送信装置にて説明する。

「赤外線でのコードの送信」は「コード変換」で作った音声ファイルの再生を行う。赤外線送信装置をイヤホンジャックにさしているため、これによって赤外線を発しているこのアプリはAppStoreにて配布する予定である。

赤外線送信装置

赤外線通信はテレビのリモコンや携帯電話の通信等に使われている。赤外線受光素子は600 μ sごとに38kHzの信号が入っていればL、そうでなければHが出力される。

送信するデータは開始コマンド、データ容量、データ、終止コマンドによって構成されている。このデータを、Hは38kHzの正弦波600 μ s、Lは無信号に変換し、これを送信する。

iPhoneには赤外線装置が付いていないため、イヤフォンジャックに接続する赤外線送信装置を作った。赤外線LEDを二つ使い、カソードをどちらもGNDに、アノードを片方をR、もう片方をLにつないだ。iPhoneの性質上、赤外線通信に必要な周波数の電気信号を出せなかったため、二つのLEDを交互に光らせることにした。

赤外線受信及びデバイス制御装置

赤外線を受信して、コードを変換し、実行する。マイコンはPIC24Fを使っている。

デバイス本体

戦車のような外見をしていて、前進/後退（2段階の速度調節可能）、左/右回転（90°/180°/270°/360°）、停止、玉発射ができる。

感想 - Impressions

宮嶋 iPhoneアプリの開発は全てネットを参考にし、手探りに行った。Swiftはc言語と違い、オブジェクト指向なので理解するのに時間がかかった。Swiftの記述自体は簡単だったが、標準で装備されているクラスの使い方に苦戦した。

もともとはQRコードをパズルのようにして、自由に組み合わせられるようにしたかったが、QRコードが高機能すぎたため逆に扱いずらく、現状ではできていない。

高橋 今回は売り物の両立とがたいへんでした。しかしプログラムが学べたので良い経験になったと思います。来年はサーボモーターなどを使った難しい製作にチャレンジし後輩に教えていきたいです。

産ミガメ

製作者 H 2 渡辺悠太
H 1 和田直輝
M 3 飯島勇人

ストーリー

むか〜しむかし、あるところにコニシといううさぎと、何の変哲もない亀がいました。

あるとき、亀はチカチカの基板をつくりながらコニシとうさぎの魅力について語り合いました。

亀「にわか **wwwwww**、うさぎ **wwwwwwwwwwww**」

(飛び交うつまようじ!!!!)

コニシ「だからなんで投げるんだよ (●`ε'●) 拾うの大変なんだよ!!!!!!」

「もう怒った。じゃあ走って勝負を決めよう、この物大から広尾駅までの坂をどっちが早く着けるか」

話の流れでコニシは亀に競走を持ちかけてしまいました。

しかしながら、勝負は亀の勝ちに決まっていました。小西はうさぎではありましたが重力に逆らうことが精一杯ででんぐり返しもうさぎ跳びもできなかったからです。

競走の前日、亀はどうせならばかつこよく勝とうと自らの身体を機械に作り変えました。

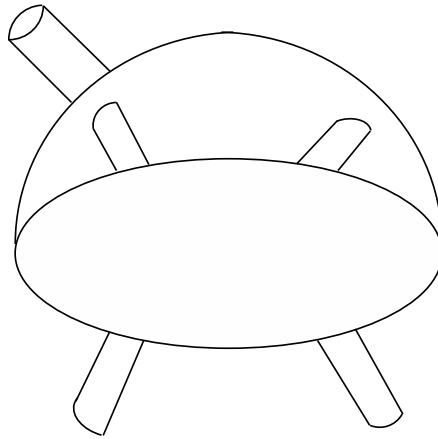
鋼鉄の足、8つのサーボモーター、それらを統べる超音波通信!!

また、対コニシ用の大砲と、より軽量化の実現した発泡スチロール製の甲羅!!!!

これがあれば無敵だと思った亀はその場で寝てしまいました。

途中ですがいったんストーリーは終わりです。続きは最後の感想まで。

外観



わかりづらい絵を失礼します。ストーリーにもありましたがこの亀は無線で動いています。上部に超音波通信のアンテナがありリモコンと送受信しています。足は甲羅の内部と地面と接する部分の付け根に関節がありサーボモーターで動いています。甲羅にあるサーボが足を持ち上げ、付け根にある方が 180°回転させてもとに戻すことで進んでいます。大砲はピンポン球が発射される予定です。

概要

簡単にこれがどんなものか知りたいならこれを読むべし

まず最初にこれを読んでいるあなたからするとカメとはどのようなものでしょう？歩くのが遅い、長寿、甲羅が固い、かわいいなどなどいろいろあると思いますが、なんと!!このカメは!!歩くのは遅い（本物よりは早いかな?）、手荒に扱ったらすぐに壊れる（短命）、甲羅もやわらかい（発泡スチロール）、不恰好とまあ本物のカメと比べるとひどいことこの上ないです。

まあ自虐ネタはこのへんにしといて真面目にこれを紹介すると本体は甲羅が発泡スチロールでできていて、中身に基盤やらモーターなどいろいろが入っています。動き方には二種類あり、普通にカメらしく歩くのと足をしまつて甲羅が回転するというものがあります（マリオのノコノコみたいに）。電源は電池とACアダプタを使っています。当初はコンセント要らずの電池などだけで動かしたいと思っていましたがそれだと大量の電池を使うことになってしまうのでそれはあきらめるかもしれません（これを書いているときはまだ製作の途中なので何とも言えません）。本体を動かすための通信は超音波通信を使っています。それとこれはまだ製作途中なので実際に作るかわかりませんが余裕があれば本

体の上に砲台もつくかも知れません。

ここでは大まかにしか説明をしていないのでもしここを読んで気になったら後の部分をめんどくさがらずに読むことをお勧めします。

担当者は本体が飯島・和田、プログラムが和田、通信が渡辺さんです。

ここからは各部位の細かいところを紹介していきます。

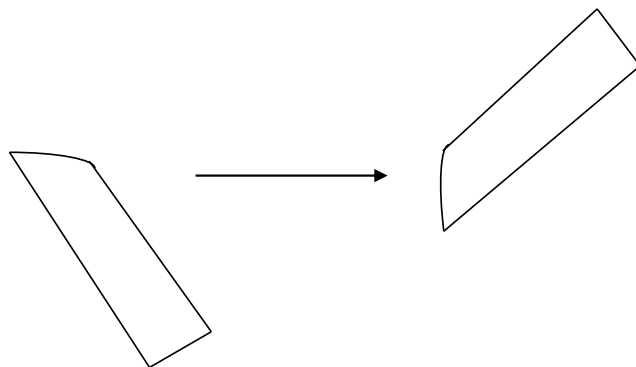
ウォーキングタートルの生態

歩き方…このカメの歩き方は普通のロボットや人間などにみられるような

歩き方とは違う歩き方とその場で回転するという歩き方がありますが、回転する方はただ足をしまいDCモーターを回転させているという単純極まりないのでこれ以上の説明はなしとさせていただきます。

それではもう一つの歩き方の方を説明します。ロボットや人間は足を地面からあげて歩くやり方やキャタピラを使い戦車のよう

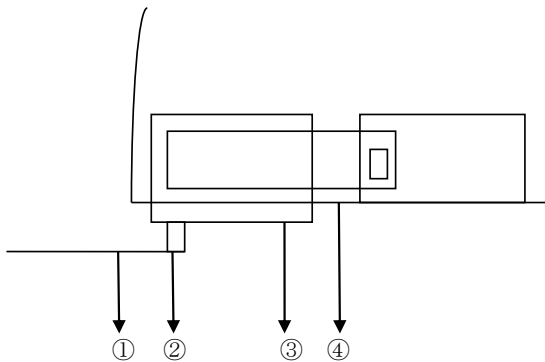
に歩くやり方が一般的だと思いますが、このカメは人間でいうすり足みたいな歩き方をしています。



上の図を説明すると、先のとんがっている部分をほとんど動かさずに反時計回りに動かすと反対側が上に上がるという原理を利用して歩いています。(図がわかりにくくてすいません)

足………金属でできていて、固い。形は陸ガメ風じゃなくウミガメ風の形をしています。裏には滑り止めがついています。サーボモーターという細かい角度の調整ができるモーターを一つの足につき二個ずつ使っていて、それが四足あります。

言葉だけの説明は難しいので以下↓カメの足部分



- ①…下から見える足の部分。地面に接している部分には滑り止めもついていて滑りにくくなっている。
- ②…この小さい長方形のものは右上のものと一緒で、サーボモーターの動く歯車の部分です。
- ③…すぐ右側にあるものと同じものでサーボモーターというものです。上の図だと少しわかりづらいですが左は縦、右は横に置いてあります。プログラムは主にここで使われていて、コントローラーで前に進むというボタンが押されたら前に進めるように複数のサーボモーターを同時に制御するというようなプログラムが書かれています。
- ④…この部分は足とは直接は関係ないのですが甲羅と腹の鉄板部分です。

甲羅………発泡スチロールでできていて衝撃とかには多少は耐えられてもなにかに刺されたりするのには非常に弱い。半球ではなく中が空洞になっていてなかの基盤などが落ちてしまったり中身が見えてしまうことを防いでいる。この部分に関していうとただ中に空洞がある発泡スチロールを被せただけです。

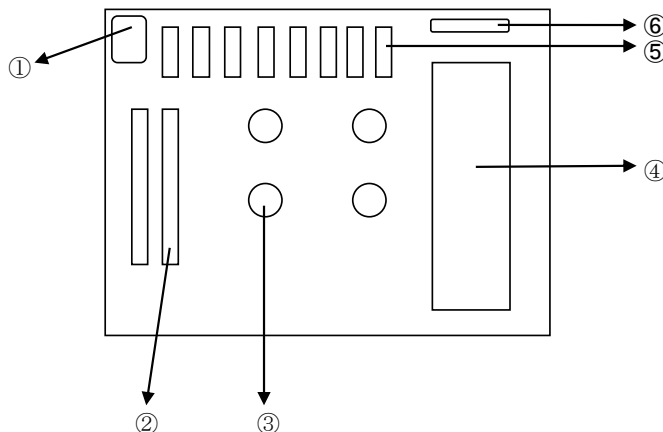
なぜ発泡スチロールなんていう耐久性が微妙なものを使っているのかというと発泡スチロールはほかの材質に比べるとものすごく軽いんです。後でこのカメの歩き方も書くと思いますが本体が重すぎると歩くのに支障が出かねないのもものすごく軽い発泡スチロールでできています。それなら薄い金属やプラスチックで作ればいいんじゃないかと思う人もいると思いますが、きれいな半球を作るのが自分たちの技術力だと難しいのとお手頃な値段でも売っていなかったからです。

腹部……裏から見ればわかると思いますがこのカメの腹部は甲羅とは明らかに違うものでできています。それはなんと金属!!甲羅よりも耐久力に勝っています。これも甲羅と同じく金属板を切ってつけたという簡単なものです。なぜ金属を使ったのかというと中にいろいろつけるときに加工が簡単だからという理由です。

武装……カメの甲羅の上には砲台がついており、どのように球を飛ばすのかというと二つのモーターをそれぞれ逆に回転させそこに球を落とすことによってとばすというものです。残念ながら今のところ球は一発ずつしか発射することができません。いつか時間があったら改良して連射などをできるようにしたいです。

基盤

この基盤はカメの甲羅の中に搭載されています。下の絵を見てもよくわかんないと思うので各部品がどのようなものか軽く説明します。



- ①…「DCジャック」電源を供給しているものです。5V電源を使っています。
- ②…「モータードライバ」これは直接PICの出力ピンをモーターにつなぐと電流が足りないためにうまく動かすことができないので、モータードライバを経由して電流を増幅してモーターにつないでいます。
- ③…「テスト用スイッチ」コントローラーのボタンを使ってもこのカメは動くんですがテスト用として基盤にもカメを歩かせるスイッチがあります。
- ④…「サーボモーターピン」名前の通りサーボモーターに使うピンですサーボモーターはここに接続することで動いています。
- ⑤…「pic」これは **pic** というものでプログラムはここに書きます。今回は **pic16f1938** という **pic** を使っています。
- ⑥…「書き込みピン」PICキットというものを使ってパソコンとPICをこの書き込みピンを通してつなぎ、プログラムの書き込みや動作テストなどを行います。

超音波

ここからは H2 の渡辺がお送りします。なぜ超音波なんてつけたか、超音波使いたかったんです。いいでしょう。つけたって。以降は超音波の簡単な説明と回路のみです。

超音波とは

超音波とは人間には聞こえない高い音の総称であり、一般には 20000Hz 以上僕が使っているのは 40000Hz の音です。まるで危ないという偏見持っている人もいるかもしれませんが、この程度の音の大きさ高さじゃ普通の音聞いているのと変わりませんよ～。30kHz 位は聞こえる可能性があります、40kHz は人間には聞こえません。聞こえる人は思い込みか人間やめてますっ！

特徴は扱いやすい、向きを合わせないといけない、障害物に弱い etc.

こんなのなぜ使ったか？使いたかったからだよ！

超音波回路

基本的には市販品の送信機(40kHz)を受信機(10～50kHz)で受けてその可聴部分をハイパスで切るというものです。もともと FM をやろうとした名残が受信機にあるのですが、もともとしっかりとしたバンドパスフィルタを作っていたのですが失敗し今の形です。とても回路はシンプルです。

感想

M3 飯島 今回はチカチカ隊長の役職が長引いてしまい、あまり関われませんでした。最初はサーボモーターの加工の仕方も分からず、大変でしたが、先輩方の指導のおかげもあってある程度扱い方がわかるようになりました。やはり金属加工は楽しいなあと思いました。

来年はプログラミングにも挑戦してみたいです。

ストーリーの続き

寝坊などしない亀は翌朝ちゃんと起き、コニシと同じタイミングで物大に立ちました。

コニシ「さーんぽしてるだけ〜」

龜「www」

物無「よ～い、どん！！！！」

刹那、亀の砲台から 744 本のつまようじが発射される！！！！

コニシ「だからなんで投げるんだよ (●`ε´●) !!!!!!!」

亀「このすきに！！！」

そう、亀はピンポン球を発射する大砲をこの目のためにつまようじ仕様に換えてきたのです。

勝負は決したかに思われました。

コニシ「否！！！」

「くらえ!!!前回り受け身iiiiiiiiiiiiiiiiiiiiiii」

そう、コニシには伝家の宝刀前回り受け身が合ったのです。

亀「グバああ、ごほ」

亀はその場で分解されてしまいました。

H1 和田 今回は三人で一緒にカメを作っていましたが飯島とは飯島がチカチカをやり始めてしまったせいもありあまり一緒に製作できなく、渡辺さんとも本体と通信担当という違いからそこまで協力して製作をすることができなかったのので来年また誰かと一緒に製作することがあればもっと協力してやってみたいと思いました。来年はもう少し難しいことにも挑戦したいと思いました。

H2 渡辺 ああ、超音波で一年が終わってしまった。もっとさっと終わらせて別のことにしようと思ったのに…。和田君ごめんね。一人で本体プログラム任せて。飯島も頑張ってるね。今からでも亀協力…。まだ超音波終わってないや…。はい。楽しかったなあ。個人的には楽しめたと思います。以上

Name

small maroon 号

Producer H2 Ryotaro.Y

H1 Keito.S

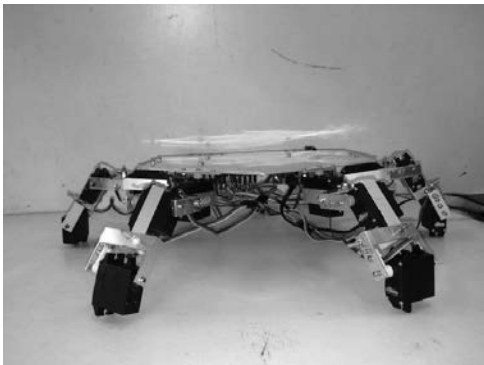
Follower butumu

Story

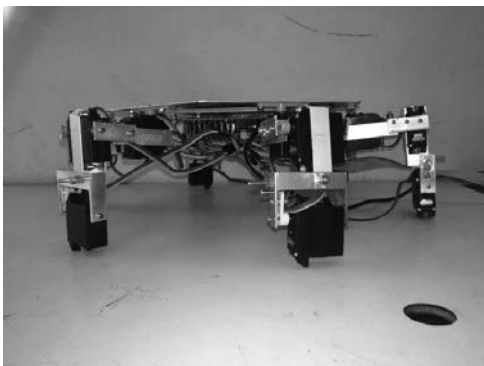
この世界はブラック久松の手によって闇に染められようとしていた。
そこで立ち上がったのはホワイト久松自分の分身の横暴を止めるため立ち上がった。
しかしホワイト久松は力が驚くほどに弱い。
そこで彼はこの small maroon 号を作り上げた。
この世界を救うことができるのか。ホワイト久松の隠された秘密とは。
こう御期待。

Image

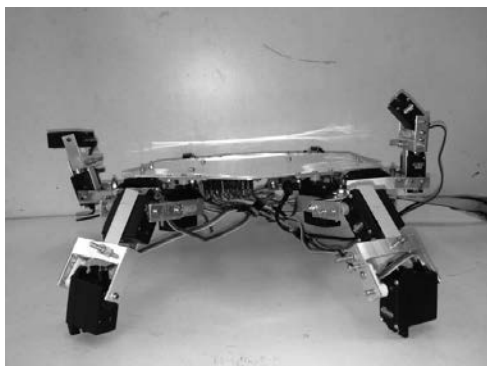
中姿勢



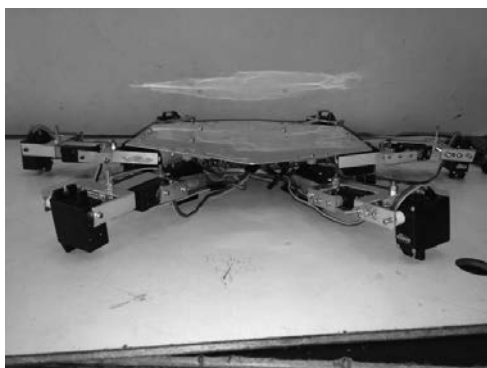
高姿勢



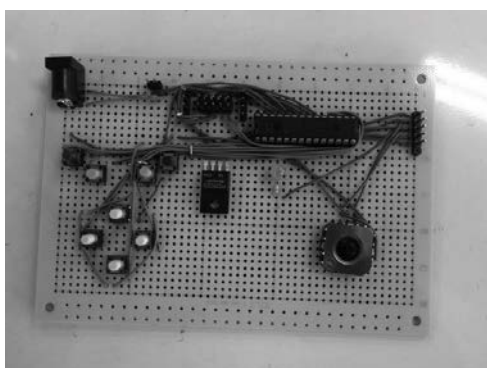
四足



低姿勢



コントローラー

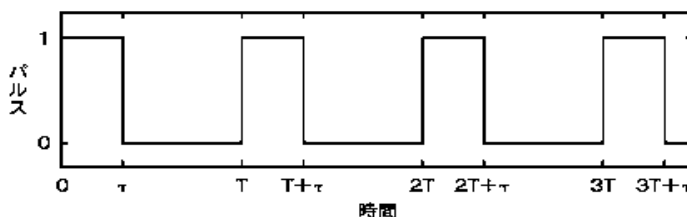


About

六足で動くロボットです。低姿勢、中姿勢、高姿勢、四足の四つの体勢で動きます。動き方として低、中、高姿勢は六歩行へ進み、右左に回ります。中姿勢は大きく回るのと小さく回る、の二種類に分かれています。で四足は前後に回り前後に進みます。一つの足につき3つのサーボにつき、計18個ついています。サーボが一個1000円チョイなので20k円ですね。高いなと思います。



サーボモーターとはパルス波を送って制御します。パルス波はデジタルのHとLの信号を繰り返し送り、アナログのように見せかけるものです。するとそのHとLの比に合わせてサーボの角度が決まります。サーボは一周 360° 回らずに半周 180° しか回りません。 180° の角度にそれぞれに対応したTとtの比



率に対応しています。サーボモーターのメリットは四角いので加工した金属と組み合わせやすいです。後、角度が絶対的に決まっているので初期位置を気にする必要がないです。デメリットは 180° しか回らない、価格が高いことです。サーボはTが20msでtが1~2msですがそれぞれのサーボでHになっている時間をかぶせないようにしても時間が足りません。ですから多少サーボそれぞれのHのタイミングをかぶらせています。

足の関節はそれぞれ横に動かすのに1つ縦に動かすのに2つです。中姿勢は縦の関節が 45° ずつ、高姿勢は本体側の縦の関節が 90° 曲がり、低姿勢は本体からもっとも遠い関節が 90° 曲がっています。四足は前足と後足を上げています。四足は上げている足がバランスを崩してくるので

コントローラーは有線です。8つのボタンとアナログスティックがついています。4つのボタンが姿勢を変えるもの、ほかの4つが旋回するものです。アナログスティックは進む方向に傾けると進みます。アナログスティックはAD変換というものをしています。AD変換とは0~5Vを1024等分するものです。アナログスティックはスティックの位置によってそれぞれのピンの抵抗が変わり電圧も変わります。その電圧によってスティックの位置をPICで判定して縦3等分横3等分の9等分で位置を分けています。AD変換は難しいものではなく

歩き方では足の角度を変えた時に足の接地点を変えないように少しずつ関節の角度を変えながら歩かせていきました。ですので物理シュミレーターなどは使っていません。1つ1つ変えていって歩かせています。

Impressions

H1 Ryotaro.Y

今までのどの年よりまじめにやりました。足が多いのでバランスを取るのは楽でしたが制御が多くて大変でした。プログラムが1動作につき18個の角度を指定しているので気持ち悪いことになってしまいました。とりあえず一通り終わったので周りの手伝いなどをできたらなあと思います。

M3 Keito.S

初めてのロボット系で本体などの加工をしましたが、なかなかうまくできました。去年とは違う系統の制作物だったのでいい経験になったと思います。たくさんの中1に囲まれ楽しかったです。

Wake-Up-Snake!

製作者 H 2 横尾

協力者 物無のみんな

1 ストーリー

♪～ジャーパーネット ジャーパーネット 夢のジャーパーネット幸子～♪

さて、今回紹介する商品は～～なんと！！目覚まし蛇です！（おおおっつ）

この蛇はなんと、目覚ましがついてるんです。本来この蛇は南米の地域に住むガラガラ蛇で、

ホルマリン漬けにされ標本になる予定だったのですが、物理学者 sidetail 氏による動物愛護の心によって救われたというレア物です！

この蛇なんとたったの4万8千円！（やっすっつ）

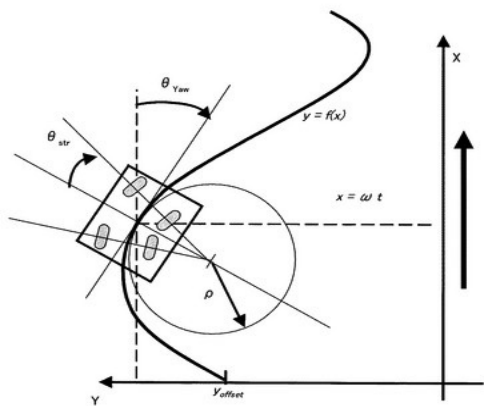
しかし売りません。僕が持って帰ります。（えええっつ）

♪～ジャーパーネット ジャーパーネット 夢のジャーパーネット幸子～♪

2 本体

はい。本題にはいりましょう。今回のロボットはずばり蛇です。正直、当初はくねくねするだけで、進まないだろうと思ってましたが、なぜか進んでしまったので結果オーライでした。単純に言ってしまうとこの蛇ロボットは、節が五つの部分に分かれていて（もうちょっとふやしたかった）、その間の四つにサーボモーターという角度が調節できるモーターをはさんでいます。そしてある角度の動きを作ってそれを後ろに時間をずらして行って伝えていき、蛇っぽい動きにしています。

そのある動きというのは一般にサーペノイド曲線とよばれているもので、このような動きを蛇がすることによって、前に進むのです。



サーペノイド曲線の図

次にではどのようにして曲がればいいのかと考えた結果、まがりたい側のカーブを小さくすることによってだんだん曲がっていくという仕様にしました。実際一回に曲がれる角度というのがせいぜい 30 度くらいで果たしてそれは曲がるということなのかと指摘されたのですが、押し切っちゃいました。

・目覚まし機能つき

実はさっきまでの前に進むだとか曲がるだとかいう話は蛇の走行モードという機能でして今回さらに目覚まし機能というものを付けました。

といっても別にたいしたものではなく、アナログの目覚まし時計のアラームがなると同時に、スピーカーにはいつてきた電気信号を受け取り、がしやがしや動きます。そして、スイッチをおすとその動作がとまるというものです。がしやがしやすることで、硬い金属のロボットが動くわけですから、お宅の起きない子供たちを簡単に起こすことができます。これでもう安心ですね。

ここでは時計は時計で動いてくれるため、べつに問題はなく、一番の問題は電気信号がはいったときにそのまま蛇が動き続けるという動作をすることでした。プログラムの問題上、割り込みをすることができず、どうしようか考えた結果、ICを使うことにしました。正直懐かしかったです。これを使うことにより、その電気信号を維持し続けることができ、動作を続けられるようになったのです。

3 プログラム

今回、説明をあまり難しく書きたくなかったので、詳しいことは書いてないのですが、ふろふえっしょなるな方は、こちらのC言語を読んで理解していただけるとありがたいです。乗せるプログラムは本当のことをいってしまうと目覚ましとつなげようとしている段階ですので、ご了承をおねがいします。

```
include <htc.h>
#include <math.h>
#include <stdio.h>
#include "pic16f1938.h"

__CONFIG (FOSC_INTOSC & WDTE_ON
& PWRTE_ON & MCLRE_OFF & CP_OFF & CPD_OFF & BOREN_OFF & FCMEN_ON);
__CONFIG (WRT_OFF & PLLEN_ON & STVREN_ON);
#ifdef _XTAL_FREQ
#define _XTAL_FREQ 32000000
#endif

int just1=455;//30 110
int just2=755;
int just3=1055;
int just4=1365;
signed int a1=0;
signed int a2=0;
一部省略
signed int a40=0;
long serbo=0;

void mezama(void);

void delay_us(long time){while(time>0){__delay_us(1);time--;}}
void delay_maef(long time,signed int a1){
    while(time>0){
        __delay_ms(1);
        time--;
    }
    just1=a1+455;
    just2=a6+755;
    just3=a11+1055;
    just4=a16+1365;
    a40=a39;
    a39=a38;
    一部省略
    a2=a1;
}
```

```

void interrupt warikomi (void){
    if(TMR2IF){
        TMR2=0;
        TMR2IF=0;
        serbo++;
        if(400<serbo&&serbo<=just1){RA0=1;} else {RA0=0;}
        if(700<serbo&&serbo<=just2){RA1=1;} else {RA1=0;}
        if(1000<serbo&&serbo<=just3){RA2=1;} else {RA2=0;}
        if(1300<serbo&&serbo<=just4){RA3=1;} else {RA3=0;}
        if(serbo>=1600){serbo=0;}
    }
    if(INTF){
        INTF=0;//RB0/INT 外部の割り込み禁止
        while(RB1==0){
            mezama();
        }
        RB2=1;
        _delay_ms(10);
        RB2=0;
    }
}

void progress(void){

    delay_mae(2,0);
    delay_mae(2,2);
    一部省略
}
void right(void){

    delay_mae(2,0);
    delay_mae(2,1);
    一部省略
}

void left (void){

    delay_mae(2,0);
    delay_mae(2,2);
    一部省略
}

void mezama (void){

    delay_mae(10,17);
    delay_mae(10,-14);
    一部省略
}

void main (void){
    OSCCON=0b01110000;
    ANSELA=0;
    ANSELB=0;
    TRISA=0;
    TRISB=0b00000011;
    TRISC=0;
    T2CON=0b00001100;
    TMR2IF=0;TMR2IE=1;TMR2=0;//(ry
    PR2=0x4a;
    OPTION_REG=0b10000000;WPUB0=1;INTCON=0b11000000;
    INTEDG=0;//割り込み条件を下リエッジにする
    INTF=0;//RB0/INT 外部割り込みフラグをクリア
    INTE=1;//RB0/INT 外部割り込み許可
    GIE=1;//すべての割り込み許可
    while(1){
        while(1){
            progress();
            while(RB5){right();}
            while(RB4){left();}
        }
    }
}

```

4 感想

最後の製作物ですね。しみじみでございますよ。最後の最後でひとり製作ってのも風情でした。正直、Y氏がキャラに似合わず手伝ってくれてうれしかったです。あと完成しそうなのでさらにめっちゃくちゃうれしかったです。とりあえず、文化祭当日、蛇がいちばん輝いていることを願います。そしてそれが展示大賞につながることを願って。

幸子の名言:毎日、なにかが幸せだったらそれでOK !

階段登ルンです

制作:H2 佐伯大地

協力:物無の皆様

その１ 「階段登ルンです」とは

階段登ルンです(かいだんのぼルンです)は、物理部無線班が2014年夏から開発している直動アーム式ロボットの登録商標(第226xxxx号ほか)で、同ジャンルのパイオニア的製品である。日本国外ではStepClimberの商品名(登録商標第227xxxx号)で販売されている。(wikipediaより(大嘘))

ど～しても階段を登りたい人の、ど～しても階段を登りたい人のための、ど～しても階段を登りたい人による階段登りロボット。それがこの「階段登ルンです」です。

その２ 概要説明

簡単にいうと、階段を登るロボットです。段差を検知すると、アームが下に伸び、本体が上に押し上げられます。その後アームを後ろに下げることによって本体が前に進みます。そしてアームを上、前と順に元の位置に戻すことで、全体が段差の上に登ります。本体サイズは麻布学園の階段の高さに合わせました。

その３ 部品・本体・機構説明

DC モータ

最も一般的なモータです。電気を回すと回るだけのモータです。

このロボットでは、タイヤ部分に2つ、上下するアーム用に1つ、前後するネジ用に1つの計4つ使用しています。また、次項のマイクロスイッチというスイッチを、モータが回転しすぎないようにストッパーとして使用しています。

マイクロスイッチ

スイッチにもいろいろと種類がありますが、今回は検出用スイッチとしてよく採用されるマイクロスイッチを使ってみました。簡単に言うと、押すとON/OFFが切り替わるごく普通のスイッチなのですが、わずかな動きでON/OFFするようにテコの機構がついています。前後、上下のストッパー用に4つ付けています。

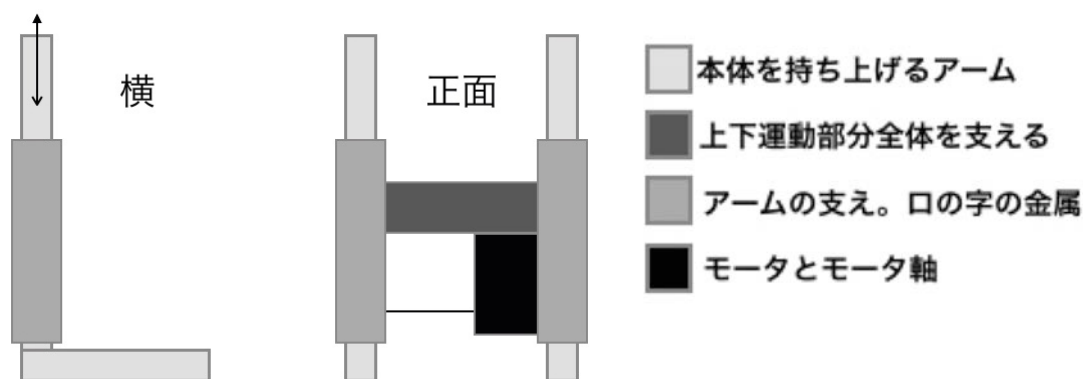
測距モジュール

測距モジュールとは、その名の通り距離を測るものです。その中でもいろいろと種類がありますが、今回は精度があまり必要なかったので赤外線式の測距モジュールにしました。他には超音波式などがあります。

赤外線 LED と赤外線受信機から構成され、ものにあたって反射して戻ってきた赤外線を受信し、受信角度からものまでの距離を測定します。

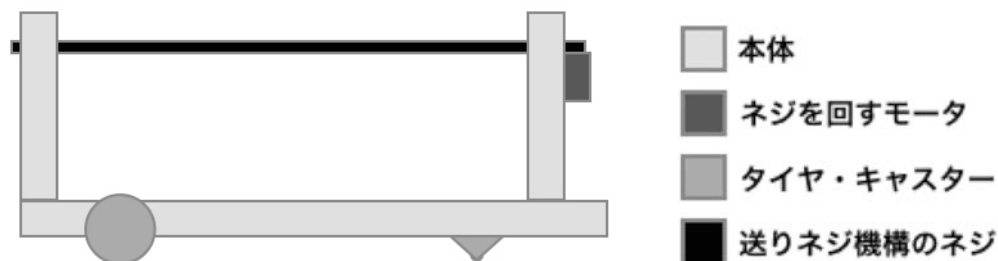
このロボットでは全面に取り付けてあり、段差に到達したかだけでなく、上昇時に視界が開けたか(本体が段差より高くなったか)も検出します。

上下運動部分



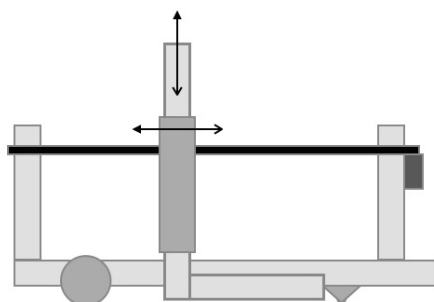
アームが上下運動できる限界まで移動すると当たる部分にスイッチをつけ、スイッチが押されるとモータが止まり、次の動作へ移行するようになっています。

前後運動部分(横から見た図)



こちらも上下運動部分同様スイッチをつけています。ロボットの一番前と一番後ろ(この図だと一番左と一番右)まで上下移動部分が移動すると、それ以上は行かないようになっています。

全体図



上下運動部分と前後運動部分を組み合わせると、全体となります。前後運動部分のネジと上下運動部分の「上下運動部分全体を支える」ところに埋め込まれたナットで送りネジ機構をつくりだしています。前後運動部分の上で上下運動部分が前後に(この図だと左右に)動くイメージです。

送りネジ機構

送りネジ機構とは、回転運動を直線運動に変える機構の一つです。

ナット(あの六角形のやつ)が回転しないよう固定し、ネジを回すと、ナットが前後に動き、ものを固定できます。そのことを固定ではなく、移動につかったのが送りネジと言われる機構なのです。

このロボットではアームを前後に動かすために、本体中央にネジが通されています。前後運動のために後述のラックギアではなく送りネジを採用したのは、送りネジは動くのがナットなので本体から機構部分が飛び出ないからです。

ラックギア機構

ラックギアは歯車の一種で、棒状の金属の一辺が歯車のようにギザギザになっているものを指します。これと歯車を組み合わせることで、歯車の回転運動を直線運動に変えます。ピニオンと呼ばれる小径歯車と組み合わせるので、ラック・アンド・ピニオンとも呼ばれます。

このロボットではアームを上下運動させるために使っています。

その4 感想

展示してある本体は実は2号機です。初号機は本体の長さを短くしすぎて、台の上に昇ったあとに重心が台の外にあり、アームを引っ込める途中に落ちこちてしまったので、本体を長くしてみました。

このロボットを作る途中でいろんな機構を調べたのですが、それがとても楽しかったです。でも行き当たりばったりで最後の制作物としてはクオリティが低いと感じております。

文化祭ギリギリまで調整や要素の追加をしたいです。

ホバーが奏でる恋のキャンパス

H2 市川 詠亮 H1 小栗 秀介

1,ストーリー ～ロボットが出来るまで～

両親のせいで借金の取り立てにあっていて麻布学園に通う小栗ひめ君は、すごい美少年だった。そんなある日、とうとう借金取りが学校にまでやってきてしまった。その危機を借金を肩代わりして救ってくれたのは横尾幸子先輩だった。

借金を肩代わりしてくれる条件として要求されたのは、学校に女装して登校することだった…

これではいけないと感じたひめ君は女装をしないで登校するためにホバー対戦バトルにあけくれるのだった。

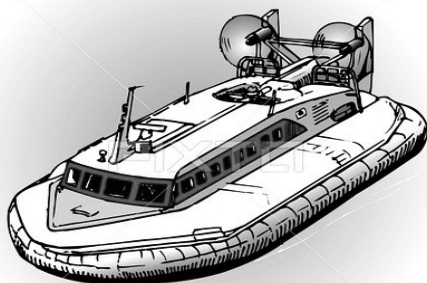
(題名は乙女が奏でる恋のキャンパスとかいうアドベンチャーゲームからパクリました。)

2,概要

このロボットは対戦ゲームになっています。2つのホバークラフトを操作し、相手をステージから追い出した側の勝利です。

ホバークラフトは前後左右に動く事ができて、高機動モードと低機動モード(シナンジュかよ)の二つがあり、二つのモードを使い分けて相手を押し出します。

コントローラーはwiiリモコンで、無線で繋がっています。



pixta.jp - 7617876



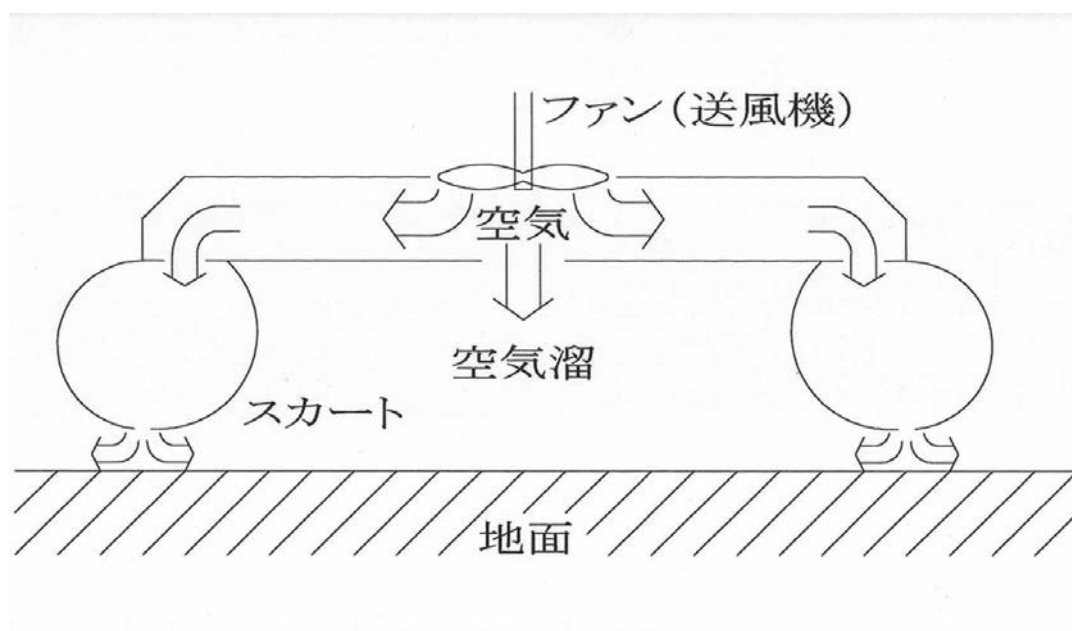
※画像はイメージです。

3,仕組み

ホバークラフトは空気を送ると空気圧が高まり空気が押し出されるので本体が浮きます。3つのモーターを使っています。乾電池駆動です。

wiiリモコンからの信号を識別して、それに対応した行動をするプログラムを書いています。wiiリモコンは改造していません。

wiiリモコンはBluetoothと呼ばれる通信方式を使っています。メジャーな通信方式でiPhoneやゲーム機、PCなどで使われています。wiiリモコンはコマンド情報と、位置情報、リモコンの傾きやどのように動いたかを、Bluetoothで送信しています。



4,感想

~~2週間でつくれます。超余裕。~~

~~—なんでかって？プログラムが去年のを完全に流用してるからさ！~~

市川

楽しく制作できました。小栗ありがとう。

小栗

楽しく制作できました。市川さんありがとうございました。

Drone Tank MK.1

H1 関根 史人

・ストーリー

“199X 年、世界は核の炎に包まれた”

まさに突然の出来事だった。
某国が衛星を核爆発させ、EMP により大陸全土が停電。
それと同時に侵攻、上陸作戦を展開していった。
そして、敗北を余儀なくされた。
しかし、あいつだけは違う。
まだ諦めるには早いというのである。
今ここにレジスタンス達は立ち上がったのであった。

・はじめに

ストーリー雑すぎて申し訳ないです。
発想力と睡眠時間が足りませんでした。
今回は極力、電気に詳しくない方でも読みやすいように
考慮したつもりですが、読みにくい点多々ございます。
どうかご了承ください。
なにか分からない点や、詳しく聞きたい方は以下のアドレスまでどうぞ
お気軽に送ってください。

メールアドレス
hama911p@docomo.ne.jp

・ 概要

簡単に言えばおもちゃの戦車です。

しかし、おもちゃを作るために半年掛けたと考えると悲しくなるので、アピールポイントを書き残して置きたいと思います。

・ 初めて本体を独立させ、一切触れずに動かせる！

「は？毎年無線で動くロボットあるじゃないかおい！」

どうせそういわれると思いました。今回は電池駆動で動くので、電源装置いらずで、独立して動くので初めてです。(多分。

・ なんとキャタピラ走行！

「どう考えても手抜きじゃ(ry」

気にしてはいけません。

でもキャタピラのほうがちゃんと動くので… (言い訳

使ったのは田宮のキャタピラセットで、アルミ板に少し穴を開けるだけで使えるので、超おすすめアイテムです。

一見、タイヤのほうが簡単そうですが、曲がるためのギミックを考えると、キャタピラのほうが楽です。

ちなみにタイヤだと、ギアボックスで左右逆に回しても曲がりません。

・ 強力なコイルガン搭載！

「コイルガンはパクリ」

どうせそういわれると(ry。

コイルガンはコイルに電池をつなぐと釘とか砂鉄とか色々引き寄せられるのを応用したもので、コンデンサという要するに電気のバケツに電気を蓄えてコイルに一気に大電流を流し、釘を飛ばすものです。

某動画投稿サイトにも結構関連作品が投稿されています。

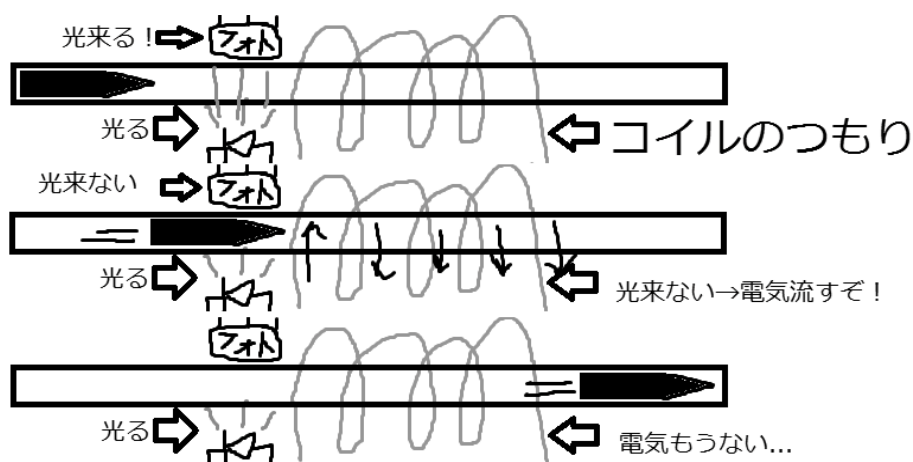
威力は？ってよく訊かれますけど、

真面目な話、威力追求すると危ないのでやめましょう

今回は大威力ではなく、「効率を追求した小型化」がメインなので、我慢してください

今年はなんとコイルが3つも付いていて、弾の通過具合をフォトトランジスタで認識、トラックで大電流をコイルに流し込みます。

以下三段式コイルガンの二段目と三段目の仕組みです。



要は「コイルの前に弾が来る→電氣流す」それだけの話です。

コイルを通過した頃にはコンデンサの中身は空っぽになっているので、引き戻される心配はないです、多分。

実際に作ると空中配線だらけになるので、超面倒です。

割と一番重要なのが昇圧回路だったりしますが、今回は去年と同じ

「Aitendo」とかいう怪しいお店の昇圧器使っています。

便利ですけど、怪しいものがちらほら見えるので多用は禁物。

・超スムーズ！砲塔回転！

「実は砲台は誰かがもう作ってたり…」

左右はステッピングモーター、上下はサーボモーターで動かしています。

正直、ステッピングはお勧めできません。

サーボモーターと違って無限回転できるのは魅力ですが、中身はコイルが巻かれているだけの金属の塊で、本体が歪むほど重い上、コイルの逆起電力により、マイコンを破壊しにくる凶悪な部品です。

なので、たいていの場合はサーボモーターを使います。

サーボモーターは内部にマイコンとモーターを備えていて、ステッピングモーターよりも遥かに高い精度で角度を指定できます。

ステッピングよりも結構小型で、ほとんどプラスチックで出来ているので、非常に軽く、「サーボでサーボを支える」ことも容易に実現できます。

ただ、サーボはステッピングと違い、180度程度しか回転できないので注意

今回はその特性を生かし、本体に直結で、負担の少ない左右回転部分をステッピング、無限回転しなくてもよい砲身部分をサーボで動かしました。

・ AD 変換！アナログスティック付コントローラー！

「ボタンスイッチだけのコントローラーは雑だと思います。」

アナログスティックというのは 3DS とかゲームキューブとかのクルクル回せるあれのことです。



使ったのは秋葉原の「千石電商」のアナログスティックで、X 方向と Y 方向、そしてタクトスイッチが付いていて、スティック自体がボタンスイッチになっています。(まあカバーつけないと押しにくくて使えたものではないですが...さて、本題の制御方法ですが、PIC の **AD 変換** を利用しています

AD 変換とは「Analog-Digital 変換」の略称で、アナログ信号をデジタルに変換する技術のことです。

基本的にパソコンやスマホなどの演算機は 0 と 1 だけで計算していることはご存知の方も多いと思いますが、この AD 変換では入力端子に流れてくる電圧の大きさ（正確には電流）を 0 と 1 の信号、つまり 2 進数に変換します。

たとえば、入力電圧が 2.43V だったとすると、AD 変換結果は

入力電圧/電源の電圧(VCC の電圧、PIC の VDD 端子の電圧)

なので、 $2.43V/5.00V=0.486$ となります。

これに 2 の 12 乗、つまり 2 を 12 回掛けた 1024 で 0.486 に掛けます。すると

$$0.486 \times 1024 = 497.664$$

になり、これが AD 変換結果になる...といたいところですが、

今回使った PIC は現在の超高性能スマホやパソコンのように小数点以下の計算ができません。(やろうと思えばできますが、ふつうはやりません。)

なので、小数点以下切り捨てで、

$$497 = 256 + 128 + 64 + 32 + 16 + 1$$

となります。

2進数に直すと

0b0011111001

ですね(最初の 0b は二進数を表す。全体で 12bit なので 12 桁)

では、本題に戻ります。

アナログスティックは **縦方向(y)と横方向(x)の角度**で軸の位置が決まります。

そこで、分圧という仕組みを利用して角度を電圧で表します。

それを AD 変換で読み取る、というわけです。

・ XBee ! 双方向高速通信 !

XBee というのは秋月電子などで販売されている無線モジュールで、Wifi などより圧倒的省電力、ローリスクで無線通信が行えます。

価格は 3000 から 7000 程度で、割と簡単に通信できます。(xbee は 2 機必要)

そもそも Xbee とは”Wifi”、”Bluetooth”のように”Zigbee”という独自の規格で動作しており、Zigbee を使ったその他の無線機と通信するための無線機で、短距離通信が基本で、だいたい数メートルから数十メートルが限界ですが上位機種になると 1km 以上は飛ぶそうです。

想定としてはおそらく、ZigBee という名前の由来が「ジグザグに飛ぶハチ」をイメージしているので、ハチの集団のようにロボットやマイコン同士のネットワークを組むことだと思います。

なので、**一個当たりの単価が安く、消費電力も低い**のです。

ただし、今回は数千円もする Xbee をバカみたいは何個も使える余裕はなく、その必要もないので、2 個だけで通信します。

Xbee の無線通信処理はすべて Xbee 上のマイコンがやってくれるので、問題ないのですが、Xbee とロボット側のマイコンは自分でプログラミングしなければなりません。Xbee にはその方法が二通りありますが、(詳細は割愛します。)

今回はシリアル通信でやることにしました。

シリアル通信とはデータの通信を 1、2 本の配線で行う技術(電源用で 2 つ要る)で、対義語の並列通信と違い少ない配線で通信できるものです。

USB ケーブルなどがその代表例です。

ただ、その分通信速度に制限があったり、プログラムが面倒になったりしますが、結局のところ大抵の電子部品がシリアル通信で動くので、覚えていて損はないでしょう。

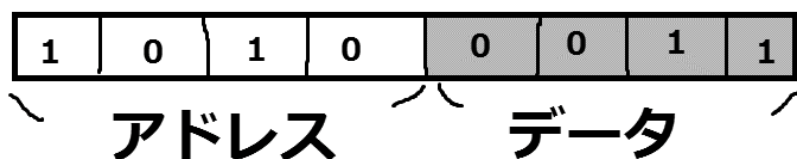
で、今回のシリアル通信は 8bit での通信ですが、送る情報は

AD 変換結果 x4(4bit のデータ)

その他スイッチ

であり、だいたい 20bit 程度になります。

とてもではないですが一度に 8bit につめて送れません。
そこで今回、それを可能にするために 8bit を分割して送信することに成功しました。
まず、8bit を上位 4bit と下位 4bit に分割します。
上位 4bit をアドレス、下位 4bit をデータと呼びます



で、アドレスで送るデータを指定して、データにその情報を書き込みます。
今回は

	アドレス	データ
AD 変換 右の縦方向	0000	0110
AD 変換 右の横方向	0001	1110
AD 変換 左の縦方向	0010	0010
AD 変換 左の横方向	0011	0101

このような感じにしてデータを送っています。
なので、いままで以上に無線でより多くの情報を本体とコントローラー間で通信でき、より細かく、複雑な制御ができるようになりました。
あと、こういう制御は bit 演算子(&とか|とか)が結構役立つので、やるなら当然覚えたほうがいいと思います。

・二次電池！目指せ！三次電池！

三次電池ほしいです。
そんなことはさておき、今回はニッケル水素電池を使ってロボットを動かすことにしました。

Eneloop だとか Evolta だとか色々新しい電池出ていますよね。
今回、負の歴史を増やさぬために軽く書き残して置きたいと思います。
最初、採用しようとしていたのが鉛蓄電池とかいう電池で、
最大出力が他の二次電池よりかなりあり、常に充電しながら使っても問題ないので、かなり昔から車の補助電池として重宝されています。
今回も「ニッケル水素より安くてモーターを動かす大出力が出せて、文化祭中は充電しながら使えていいね！」とか安易に考えて手を出したのですが、
実態は**希硫酸につけた鉛**で、壊れたらかなりやばいです。冗談抜きで。
それに**本体が歪むレベルで重く**、ギアの設定変えても遅かったのも、

モーターの電圧上げたら普通に融けて死にました（笑）
なので結局、鉛蓄電池はあんまりメリットないです。
積むなら**ニッケル水素**一択ですね。
最近のスマホは全部リチウム電池ですが、
リチウム電池はデリケートなので**多分大電流は取り出せません**。
なので、モーターが動かせず、詰むだけなのでやめたほうがいいと思います。
まとめると、
・鉛蓄電池…重い・危ない
・リチウム電池…危ない・出力が弱い
・ニッケル水素…軽い・まあ壊れにくい・大出力
ということですね。
二次電池はニッケル水素一択です。大事なことなので二回言いました

・簡素な感想

今回は「最低限絶対に動くロボット」をコンセプトに制作させていただいたわけですが、個人的にはうまくいったと思っております。
この制作で身についた技術も多いですね。
なので、結構満足しています。
ここでそんなに長ったらしく書きたくないのでこれで終わりにしようと思います。
ではでは。

A rolled ball

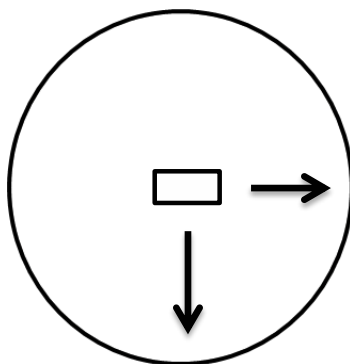
製作 H1 福島 弘也
ご協力 物無の皆様

1、ストーリー

そう、僕はボールだった。おもちゃのボールだった。犬に追いかけるために光り続ける。人に投げられ、転がりながらそれを超えるだけだった。

2、概要

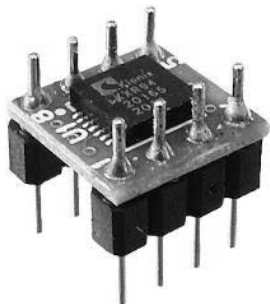
ストーリーは気にしないでください。目の前に犬がいたのです。これは、直径15cmくらいのボールです。それを人に転がしてもらい、上半分のLEDを赤色に光らせて下半分のLEDを青色に光らせ続けるといったものです。中にセンサーが入っていて、それにより転がっている方向や重力の向きを読み取っています。



真ん中あたりに一つのセンサーを配置してあります
センサーで数値を読み取っています。

3、仕組み

このロボットには一つのセンサーがついていて加速度センサーというものです。



これは加速度を判定することで、傾きや動き、振動や衝撃といったさまざまな情報が得られます。主にこちらで重力の方向を感知します。

このセンサーの結果でLEDを光らせています。

4、感想

最初は玉乗りロボットを作ろうと思ったのですが、オムニホイールというものを使ったり制御が難しそうだったりということで結論に達し挫折しました。そのため、これをつくることにしたのですが、下調べが足りなかったせいや全く別のものを作ろうとしたせいがあったのでなかなか進んでいません。来年はどうにかしたいです。

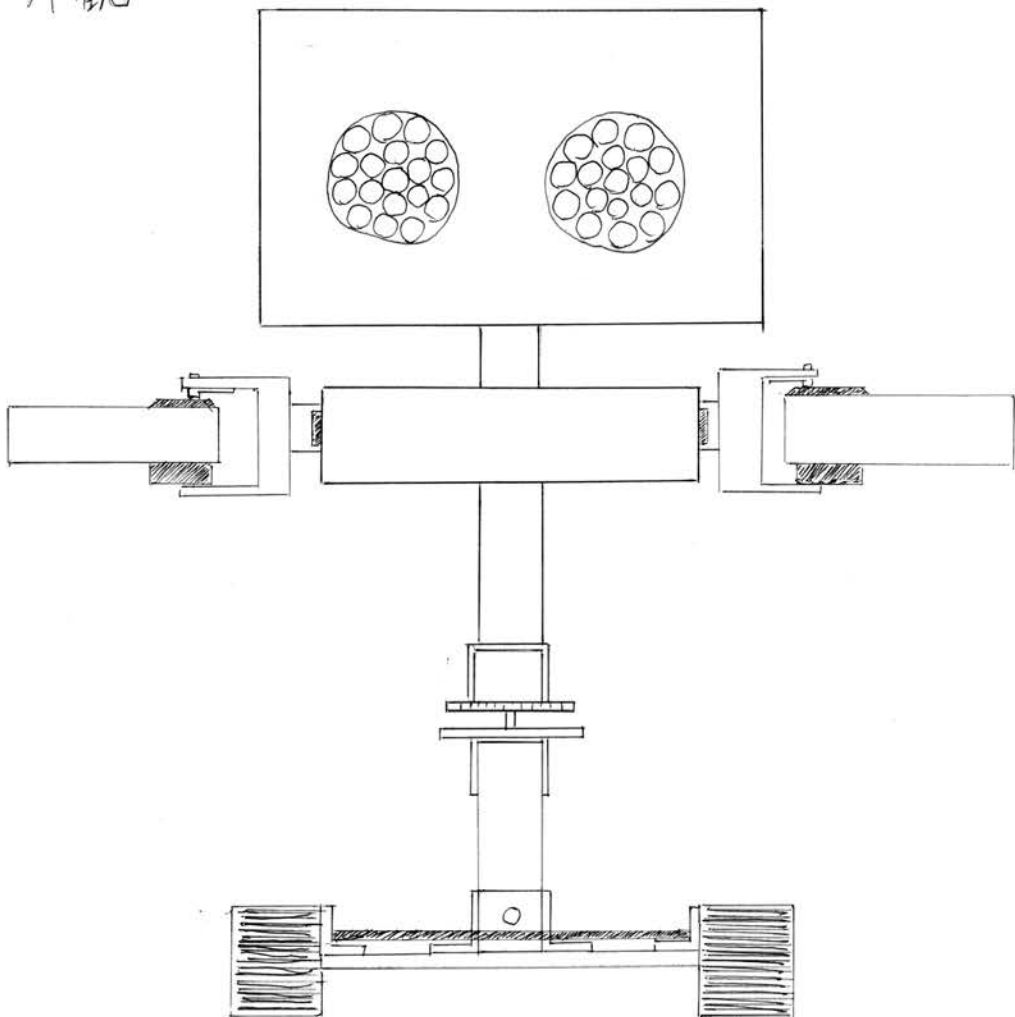
ROBOT

製作者: H1 佐藤 弘

M2 野口

M2 藤井

外觀



説明

基本的に外観は任○堂の某ゲームに登場するロボットが元となっています。頭と胴にモーターが搭載されていて頭と胴が回転し、腕は胴と共に回転する仕様となっています。腕にはサーボモーターを使用し、肩と肘が曲がる様になりました。

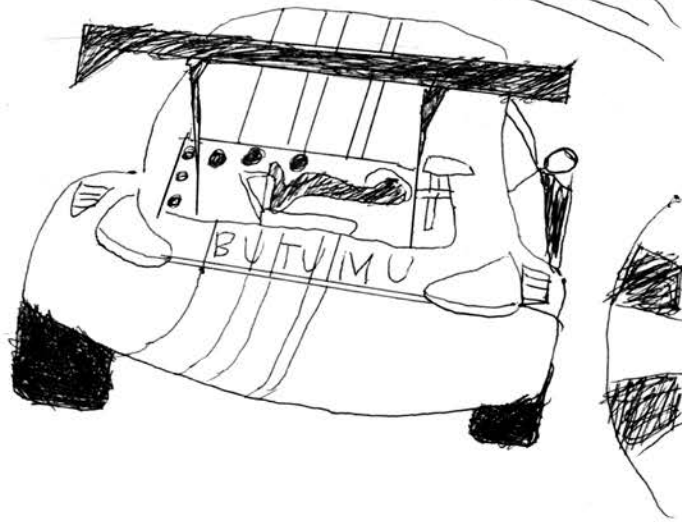
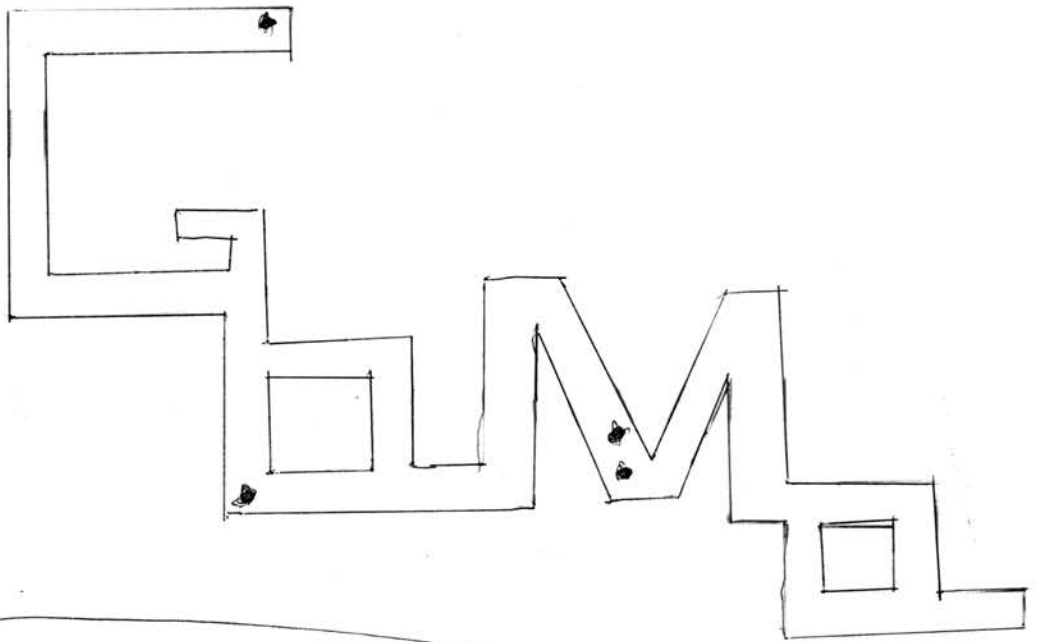
足はキャタピラで支えるようになっていますが実際の所四輪走行です。まだモーターの取り付けなどは完全に終わってはいませんが、外観のみならばバランスをたもって静止していることは現在(2/25)でも可能です。余裕があれば手にギミックをつけたいと考えております。

感想

く藤井>

プログラムよりは金属加工の方がまだできるのではないか、と思っで本体を作り始めると、恐ろしい位センスとスペックと計画性が無いことに気がつきました。残りの時間で早急に完成させたいと思います。

く野ロ>
僕はプログラムを初めてやったため、と
ても大変でした。本体をつくるのもとて
も大変で、ネジが何度も外れて大変でし
たが、楽しかったです。来年の製作もが
んばりたいです。



F-ride

製作者：H2 石倉 匠

協力者：物無の皆様

1. ストーリー

花「『魂』を！賭けよう！」

テ「グッド！」

エ「エフッ！ライドォ！」ドキュウウヘン

・

・

・

テ「いや、こんな凡ミスで君が負けるとは思わなかったよ」

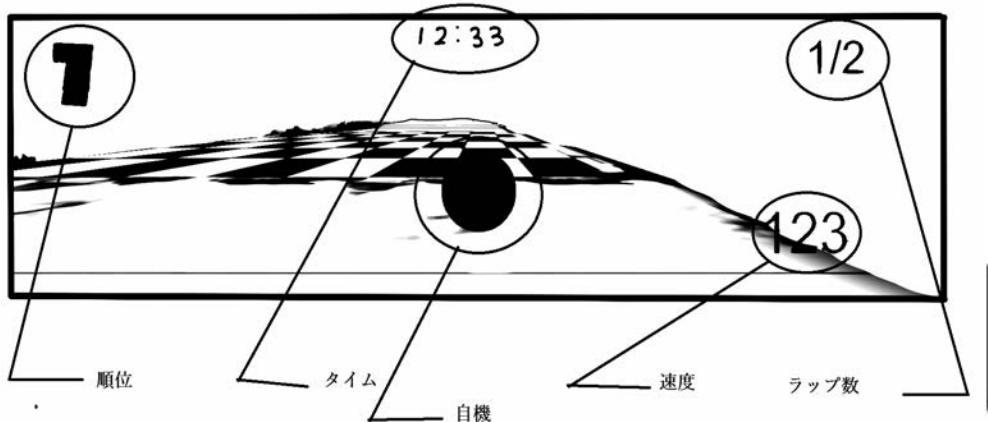
テ「さてと！レースをぞっk」

花「ラグいラグい！今絶対ラグだって今の！おかしいって絶対！クソゲーかよ！」

2. 概要

今回このゲームを制作しました、石倉と申します。あのシーン、「スタート後 2000 メートルノ直線コースが有り…」みたいにゲーム本体自体が解説してて結構シュールな感じです。

では、ゲームの概要を説明します。今回製作したのは『自分の体をコントローラーにしたレースゲーム』です。画面は下図のとおりです。プレイヤーがしゃがむと加速し、真っ直ぐ立つと減速します。カーブは体を傾けることでできます。コースを指定された回数周回するタイムを競います。まだ実装できるかわかりませんが、今までで早かった人のデータを記録し、ゴーストとして表示することも考えています。



図は製作途中のものです。

3. Unity

今回の制作物は普段物理部が使わないような開発環境を使用しました。Unity です。まずはそれについての説明をします。

Unityとは、「統合開発エンジンで、ゲームとインタラクティブな3Dコンテンツ制作のための常識にとらわれない強力な機能を提供」するものです。(Unityホームページより)
Unityは主にPCや携帯端末での3Dゲームや2Dゲームの制作のプログラミング作業を比較的簡単にしてくれる、物理演算を組み込んだ開発環境です。

4. プログラミング

今回のプログラムはどちらかと言えばゲームシステムのプログラムがメインでした。

- ・自機が走るようにする
- ・ステージを作る
- ・ステージギミックを作る

1. 自機が走るようにする

今回は対象の物体を物理的に「押す」Add forceという命令をほぼ毎フレームごとに行うことにしました。つまり、連続的に前に押す力をかけることで自機を加速させたのです。

2. ステージを作る

Unityの機能の一つ、Terrainを使用しました。Terrainは地形のエンジンで、絵を描く要領で地形を組み上げることができます。色も塗ることができるのです。

3. ステージギミックを作る

ステージギミックは動く床、加速装置、跳ね返る壁などを作りましたが、どれもそこまで特筆するプログラムではないので、省かせていただきます。

5. キネクト

今回はゲームのコントローラーとして「キネクト」という製品を使いました。そのキネクトについての説明をさせていただきます。

キネクトはモーションセンサーです。キネクトの前に立つとセンサーがその人の骨格を読み取り、その位置が座標のデータとしてPCに送られます。僕たちはそのデータを利用できるのです。

つまり、『人の体をコントローラーにしてゲームをプレイ』できるのです。例えば右手を振るとゲーム内でパンチを繰り出す、やジャンプをしたらゲーム内でもジャンプする、などのことができます。

感想

石倉 一年ほど前から作ってみたいと思っていたゲームなので、今回物理部でいちばん最後の制作でやることができ満足でした。

参考

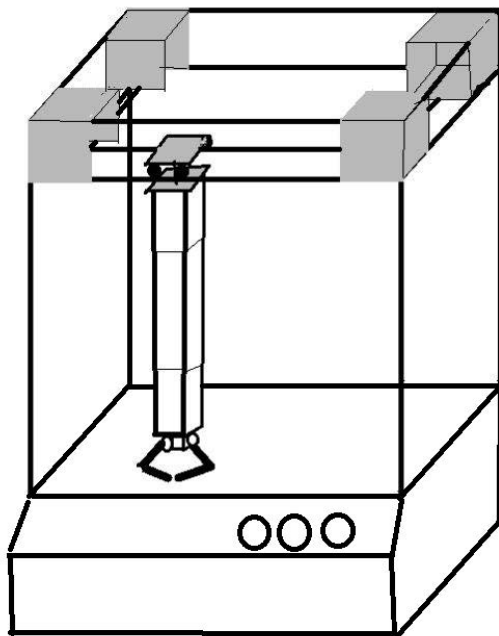
Unity ホームページ

Windows よりも使いやすいクレーンゲーム

製作者 高1 中村航 中三野末雄太 中三庄田拓海

協力 物無の皆様

外観図



・ すとりー

ある日、クレーンゲーム製作研究所に異変が発生した。もともと3人しかいなかった研究員が二人失踪したのである。ある一人のS田という研究員は決して知ってはいけないという伝説の存在“レッドポイント”の在り処をスマッシュな兄弟やポケットに潜んだ怪獣とともに8個も知ってしまい政府に追われ姿を消した。またあるN末という研究員はカードゲームにのめりこみあまり闇のデュエルにまきこまれ“次回N末死す”というメッセージを残して姿を消した。では、残った研究員のN村はどうなのであろう？彼はもともと他の研究員をあてにし働かなくてもいいと聞いてこの研究所に入ったためやる気がなかった。さて、この研究所はくれーんげーむを本当に完成させられるのだろうか？

・ 概要

はい、題名は気にしないでください。見かけた高校生の台詞です。

はい、すとーりーは気にしないでください。ほぼ事実です

さて、このロボットの概要はクレーンゲームを自作すればただでプレイできね？文化祭でもゲームセンターが好きな学生とか JK とか来て客層増えるんじゃないかね？とかいうテキトーな発想から生まれたクレーンゲームです。ボタンは概観図にあるように3つありそれぞれ右から横移動、縦移動、つかむとなっています。横移動、縦移動押している間だけアームが一方方向に動きボタンから手を離したらつかむボタンを押すまで動かなくなるもの、つかむは押すとアームが開き下まで伸び閉じ上までもどり初期位置にもどったあと再度アームを開き閉じるものと予定しております（2/22 現在）。まあ正直ただのクレーンゲームですね、工夫とかは消えました。今年の製作物の中ではイライラの次に大きいです。

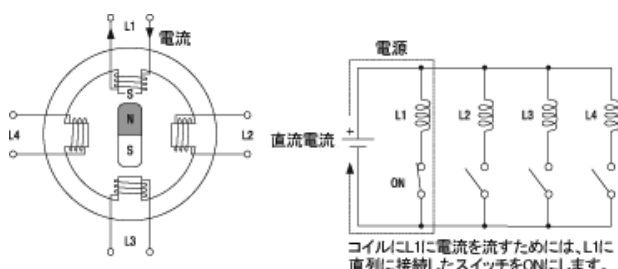
・ 構造

まずはアームの先の構造について説明します。アームの先はL字材を歯車につけてその歯車をステッピングモーターというモーターによって回し開閉するというシンプルな作りになっております。このステッピングモーターとは高速移動は出来ませんが固有の回転数（今回この製作物で使っているのは 1.8 度）ずつ回転することができ細かい調整の出きるモーターです。基本的な構造と動作原理は図があったほうが分かりやすいと思ったのでオリエンタルモーター株式会社さんのホームページからお借りました。

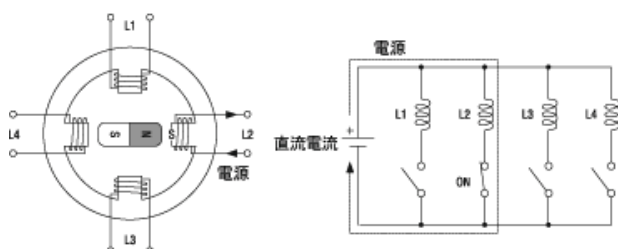
下図はステッピングモーターの簡易モデルです。

コイル L1 に電流を流すと、L1 は電磁石になります。

中央のマグネットは、L1 の磁力に引き寄せられて回転し、その前で停止します。



コイル L2 に電流を流して電磁石にすれば、マグネットは L2 の方向へ回転し、停止します。

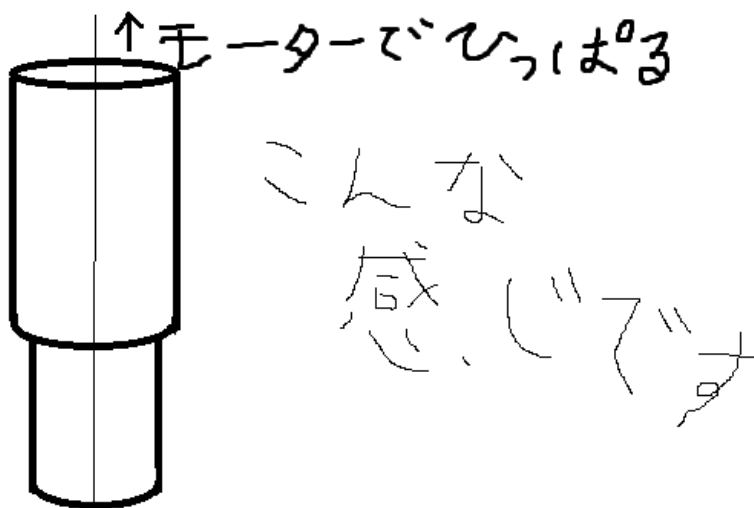


この簡易モデルでは、ステップ角度 90°となりますが、ローター（マグネット）とステーター（コイル）の磁極数を増やせば、ステップ角度は小さくなります。

ということで今回は秋月電子通商さんのステッピングモーター17PM-K044-AKZ を使わせて頂きました。つかむ力は正直弱いですが（実際のクレーンゲームを意識したとかしてないとか）。気になる人は秋月へGO！

続いて上下移動についての説明です。直径の違う塩ビパイプを重ねたものをモーターにつけたひもによって上下に動かしています。そう…お気づきかもしれませんが上下移動といっても実際は上に動くだけで下には自由落下です。なんで実際のクレーンゲームみたいに下方向に押し出せませんね。だがしかしこのアームの重量を量ったところなんと 1.6kg もありました。つまり下方向にも押し出せる…かも？しれませんね。1.6kg もあるせいでモーターは二つ使っております。この仕組みの他にマジックハンドとかナットとネジでとかいろいろ試案があったという・・・

↓は簡単な図です。マウスで字書いたのはノリです。反省しています



最後に縦横移動についての説明をします。縦横の移動はアームがクレーンゲームの機体の上についているレールの上をキャスターで横に移動しさらにそのレールが同じように機体の上についているレールの上をキャスターで縦に移動することで縦横移動します。移動するときは動きたい方向に対応するモーターが糸で動きたい方向に引っ張ります。図を載せようと思ったのですがうまく書けなかったので割愛します。

・ 感想

中村

はい。今回の製作はこれまで他人に製作の方向性とか作り方とか任せてばかりだったんでじゃあ後輩とだけ組めば他人任せではなく製作できるんじゃないかね？って感じで進めてったで

すがまさか、その結果自分の仕事量がこんなに多くなるとは思いませんでした（中3二人は反省してね）。これまで任せていたことを深く反省します。それと今回は本当に先輩方に色々手伝ってもらいました。ありがとうございました。そのおかげでプログラムの知識が増えました。やったー！

野末

今回製作したクレーンゲーム（の僕が作った部分）において最大の悩みは重量でした。ステッピングモーターがかなり重いため、ギア比の大きいギアボックスを使わなければならないのです。しかし、そうするとアームのまきあげに時間がかかるので結局ギアボックスを二つとりつけましたがその分のスペースもかなりギリギリでした。さらに、アームと筒を速く巻き上げることができる力と速さがあっても、タコ糸がその重量に耐えきれず、切れてしまうこともありました。これらのことから、単純な設計だけでなく重さや材質も考えておけば、製作がよりスムーズに進んだと思います。また、今回の製作では、共同製作者が長期間部活を休んだり、僕自身も出席率が下がり気味になりしばしば迷惑をかけたので、来年からは共同製作者との連携を大事にしていきたいと思います。

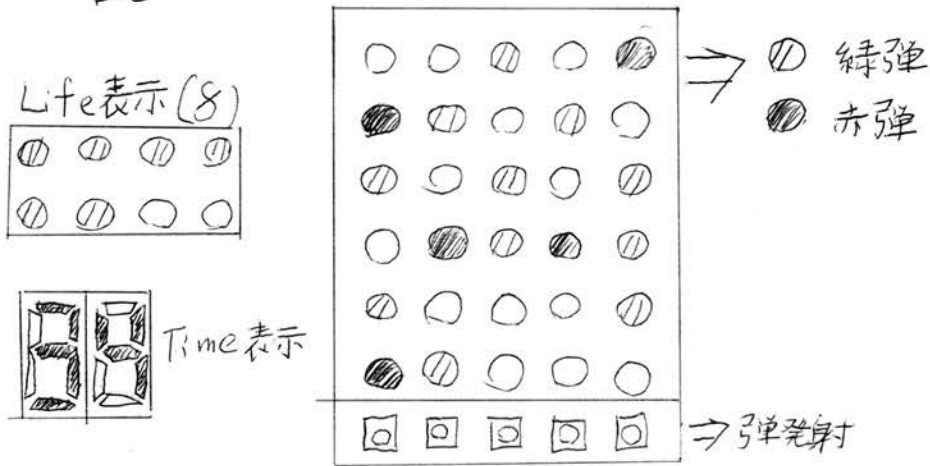
庄田

まず第一に出席率が著しく低かったことを深くお詫びします。
初めての自由制作としてクレーンゲームを作っているわけですが、
中1とは比べ物にならないくらい難しいですね。来年はこの経験を活かし
もう少し良い製作がしたいです

城守

制作: M2 井上
 回路図: H1 福島さん
 協力: 物無の皆さん

1, 外観



2, ルール

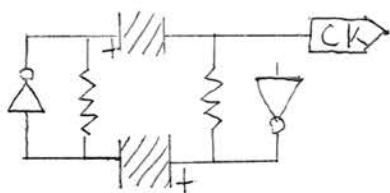
上から赤と緑の弾が流れてきます。
 緑だけ打ち落としてください。
 緑をうち損いたらLifeが1へります。
 赤をうち落としてもしつじが1へります。
 Lifeが0になったらゲーム終了です。

3, ストーリー

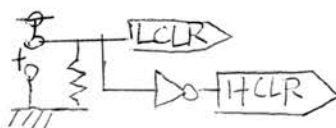
R城にG国がせめこきた。G国の兵をうち落とすための撃手達は配置につく、しかし、乱戦の中味方の兵がまぎれこしまう。
 無事にG国の兵だけをたおせるのか、い、

4, 回路図 (省略あり)

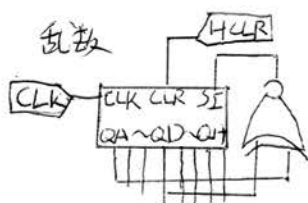
発信



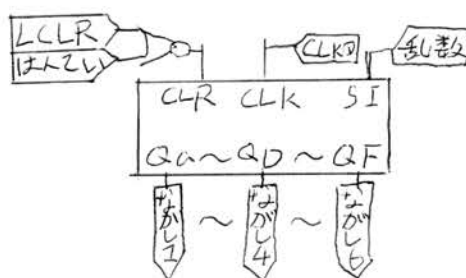
クリアスイッチ



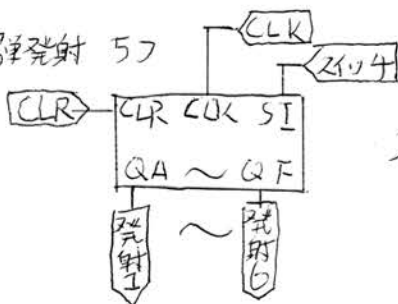
乱数



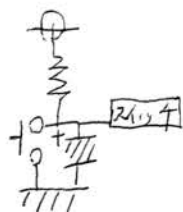
弾発射 赤、緑5個ずつ



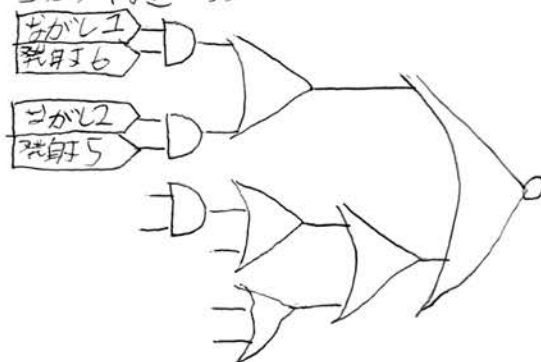
弾発射 5つ



スイッチ 5つ



当たり判定 5つ

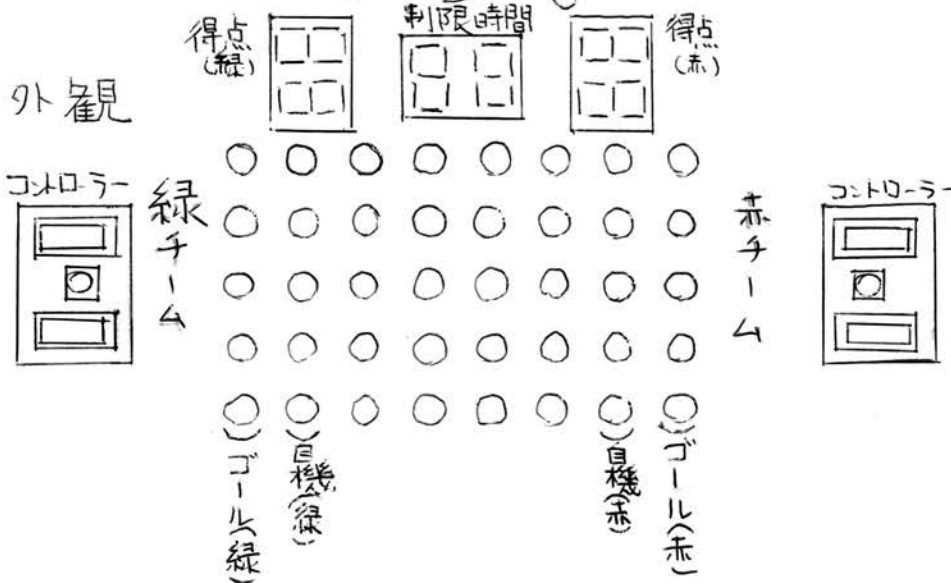


おまけ
601
502
403
304
205
106
↑
発射

チカチカ

製作者 M2 尾崎 玲帆
回路図 H1 佐藤 隼斗

サッカー



ルール

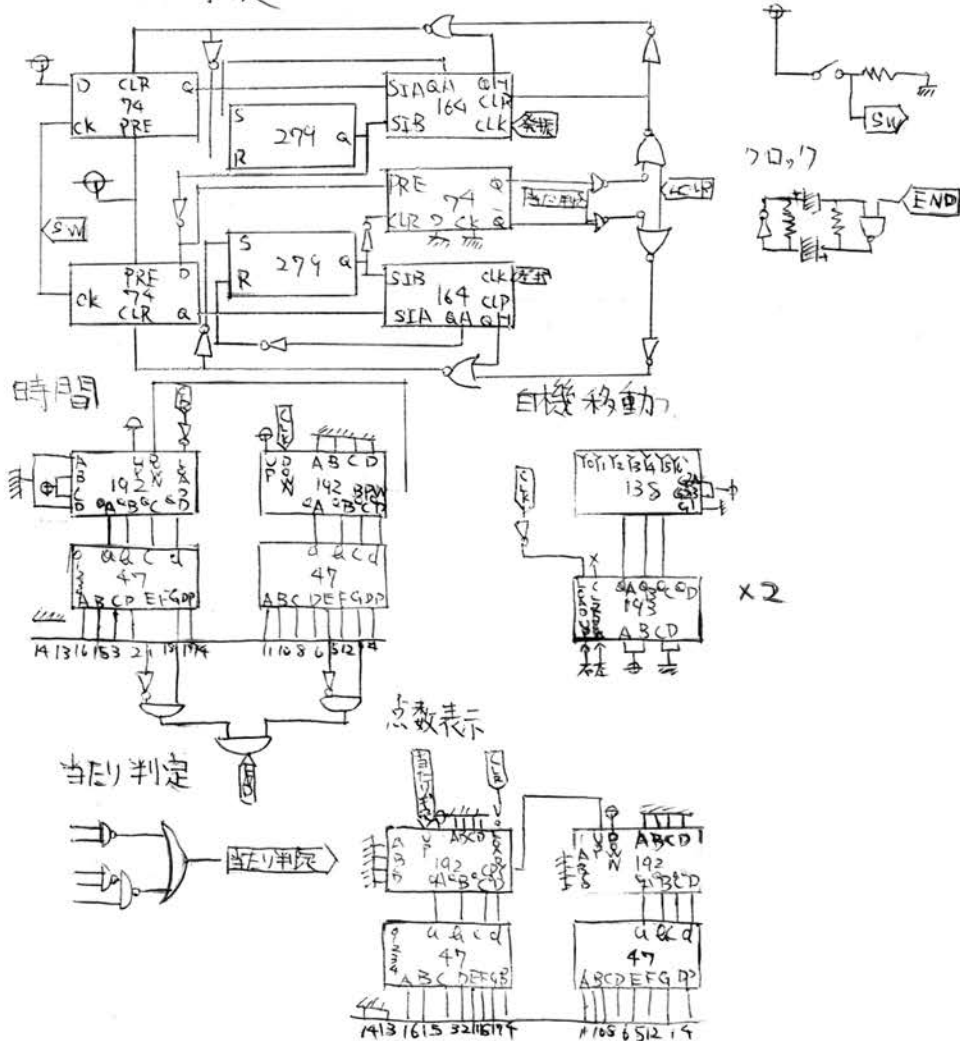
端から二列目で各チームが縦方向に自機を移動敵ゴールに向かい横方向に玉を発射し、敵のゴールに玉が届けば7点です。制限時間(1分)内に多く得点したチームが勝ちです。敵の玉を防ぐためにはコントローラーで上下に移動し玉に自分の玉をあてるか自機で玉にあたる必要があります。(自分の玉は各列に2個ずつしかありません)コントローラーの操作 真ん中のボタンで玉を発射(緑チームの玉は緑色 赤チームは赤色で光ります) 敵陣に向けて左側のボタンで自機を左に右のボタンで右に移動させます(赤チームの自機は赤、緑チームの自機は緑色で光ります)

ストーリー

ある日、目覚めるとLEDライトになっていた!!しかも前にいるLEDライトのせいで横移動ができず、攻撃したいけど手段が見あたらない!!たのにお腹に力を入れるとLEDの玉をうつるようになっていた。そこで相手の後ろにあるゴールらしきものに玉をうつてみたら相手が防ごうとした。どうやら、ゴールに玉を入れればよかった。さあ、縦移動を可能にするのはどうした?

圖路回

彈數制限



感想

製作ははじめのことばかりだったのではじめははんだごこの扱いにも
なれず何度も痛い目にあいましたが、優しい先輩方のアドバイスもあり徐々に
道具の扱いに慣れていきあまり苦勞せずスムーズに作業をこなせるよ
うにもなりました。ぜひこのゲームを愛し人をほしいです。

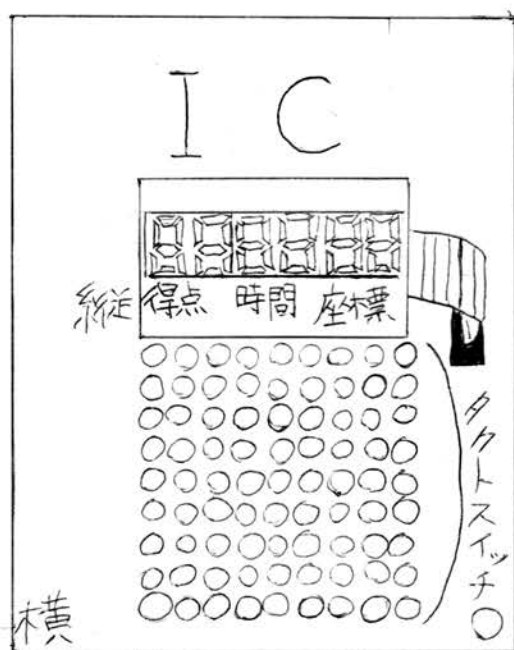
タイピング スナイパー

制作者 M2 織田 隼一郎
回路図設計者

H1 和田さん

1 外観

協力 物無の皆さん



2. 内容

81個のスイッチの中から、上に表示される座標の場所にあるスイッチを押していくゲームです。3つのワセグのうち、左が得点、真ん中が時間、右が座標を表示します。座標の10の位が縦、1の位が横を表示します。

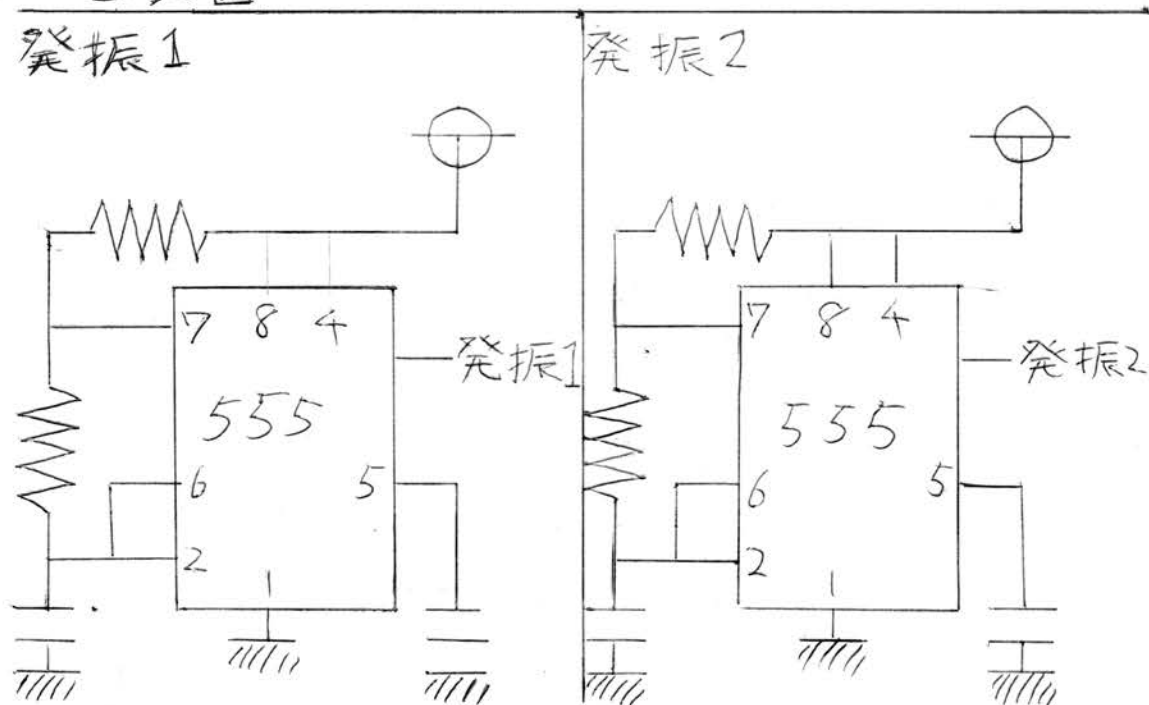
3. ストーリー

とある国のスゴ腕スナイパーは、あるビルの一室に潜入し本部からの指示を待った。その反対側のビルに集まった犯罪組織のメンバーを皆殺しにするため…本部から送られた座標の窓を撃ち、犯罪組織をせん滅せよ！

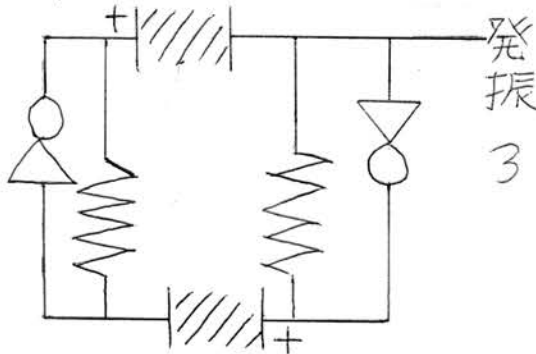
4. 感想

1番完動できなかったのは非常に悔しいですが、無事早く完動した上イライラ棒に就けて光栄です。これから楽しんで製作したいです。

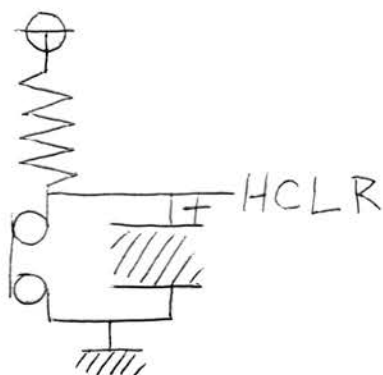
回路図



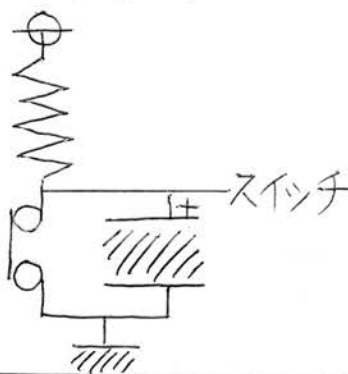
発振3



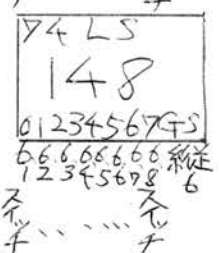
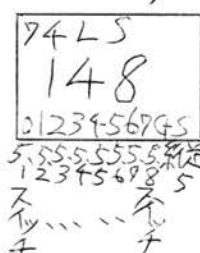
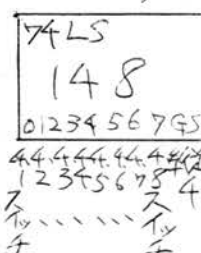
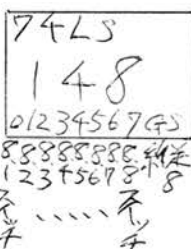
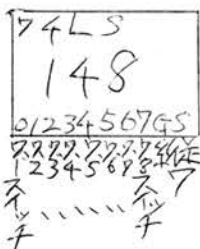
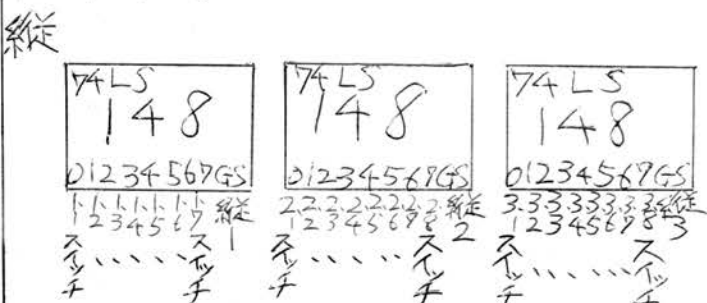
HCLR



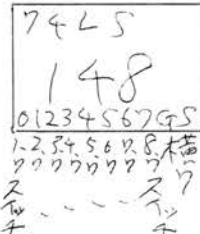
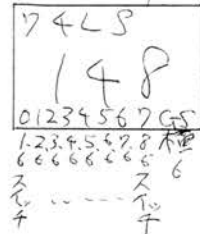
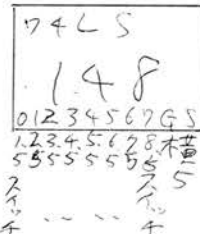
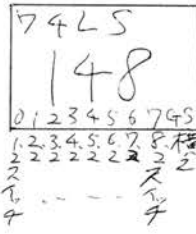
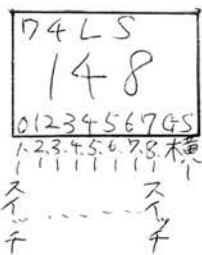
スイッチ

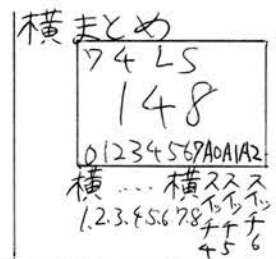
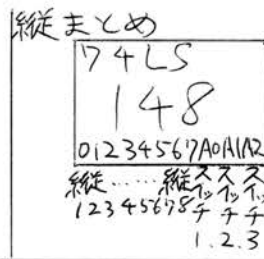
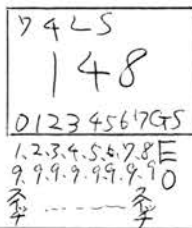
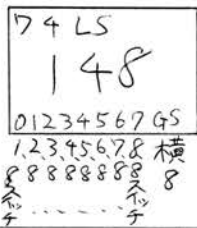


スイッチ

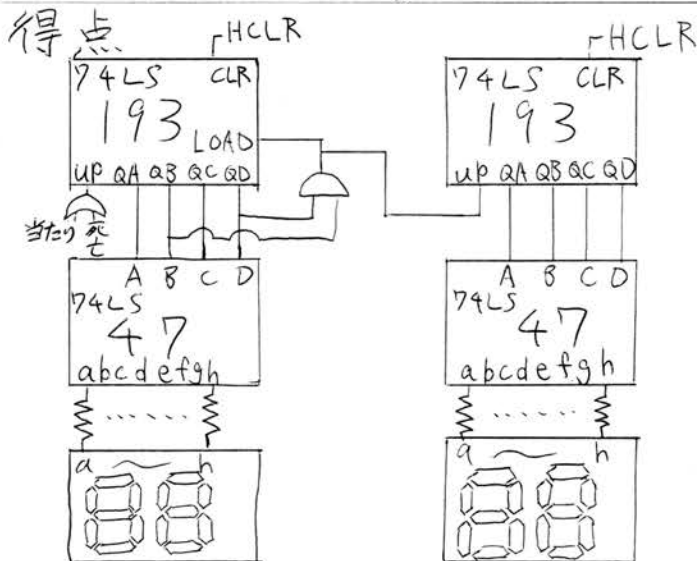
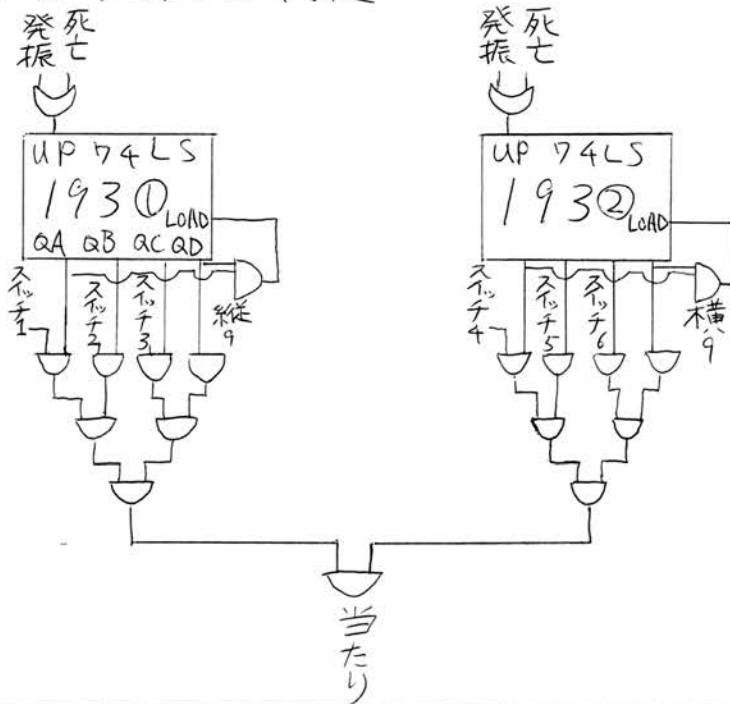


スイッチ

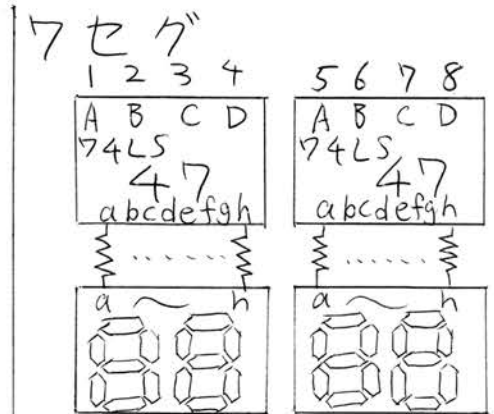
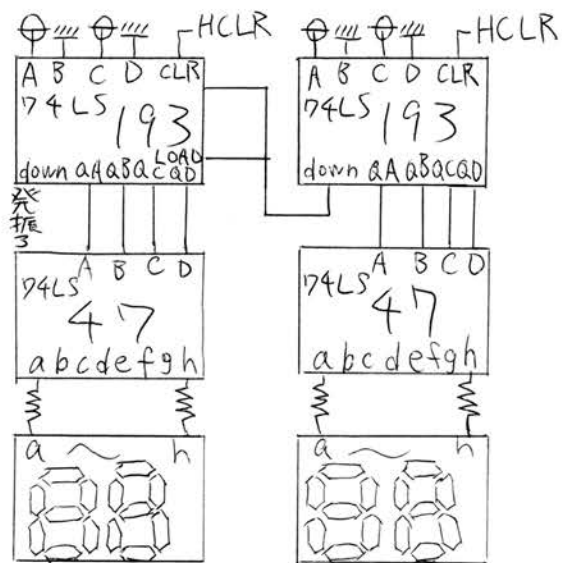


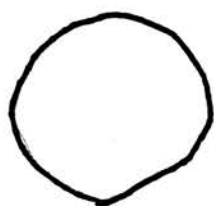


当たり、死亡判定



時間



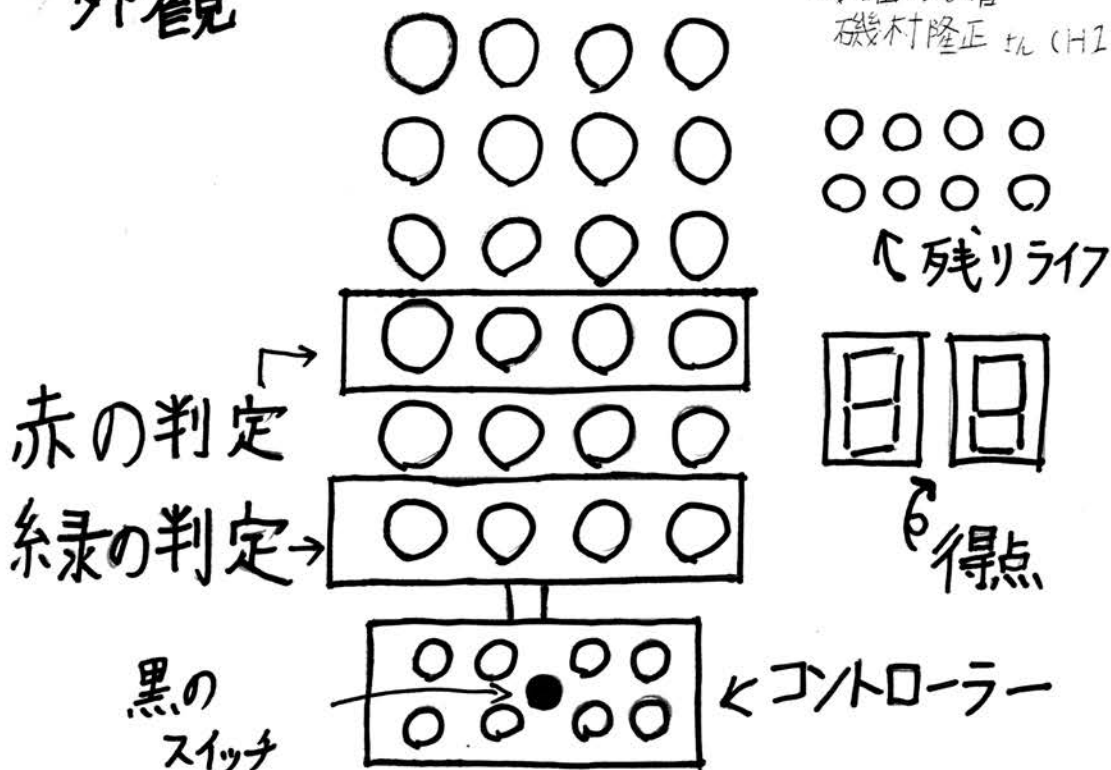


島回旋

職業会社(株)

制作者 小島 大 (M2)
回路図設計者
磯村隆正 氏 (H1)

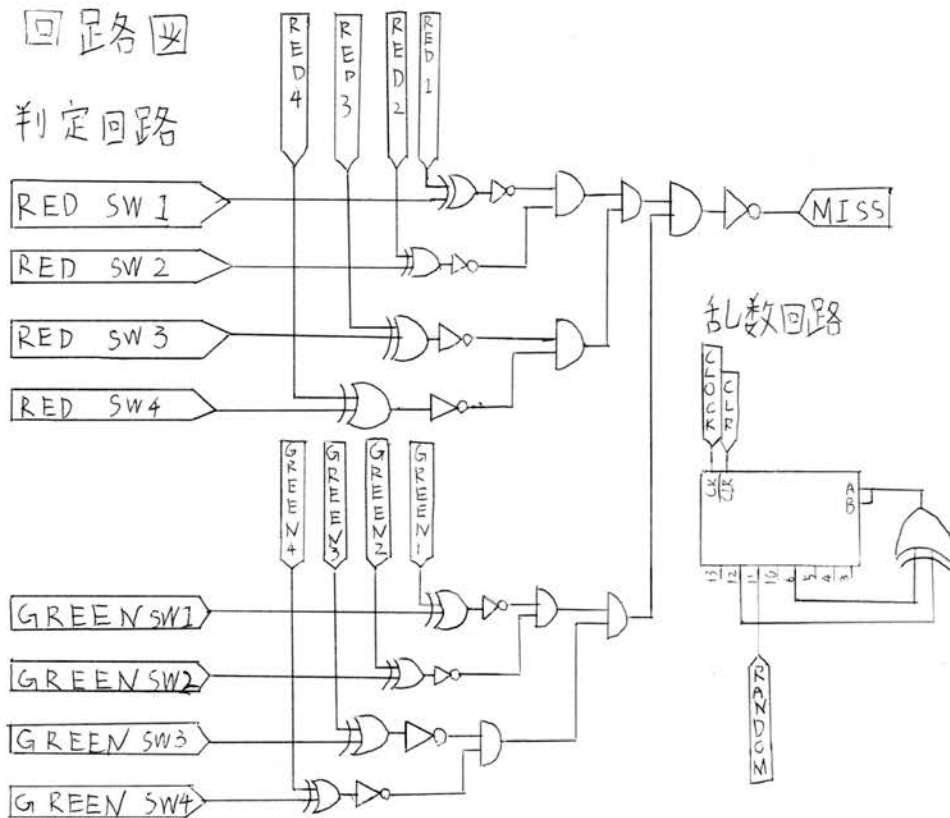
外観



ルール 上から赤と糸緑の玉が落ちてくるので、光っている色と場所の対したスイッチを押してつづければ消えます。つづいているのは手をはなせず押しつづけます。8回ミスで死亡です。一回だけ黒のスイッチを押すことで光が消えぬ。

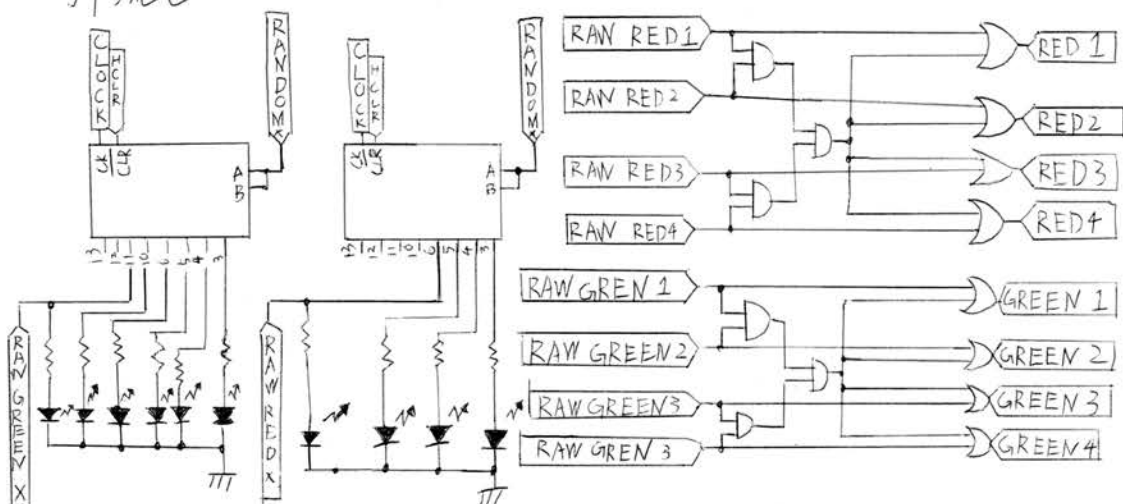
回路四

判定回路

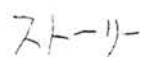


弾流シ

弾数調整回路



經過時間表示



働いていた会社で倒産し、働くあがなくなるとK島はゲームに溺れる生活を送っていた。しかし、ある日のことI島と名乗る男がやってきた。その男はいくらなものも分ける仕事をしているらしく、K島にはその才能があるといわれた。そこからは順調に進んでいき、金持ちになったとさ。

感想 今回の製作では配線線材のミスなども多くしたので
来年の作品はミスを減らしていきたいです。
今回の製作はたのしかったです。

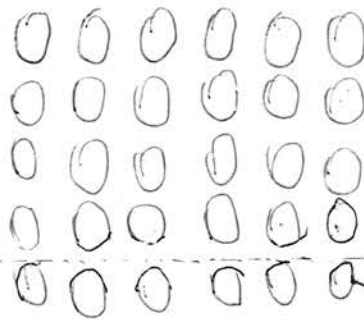
よくばりラビット

制作者 M2 小西

回路図設計者 H16 磯村さん

協力者 物無の皆様

1. 概観図



ラビット



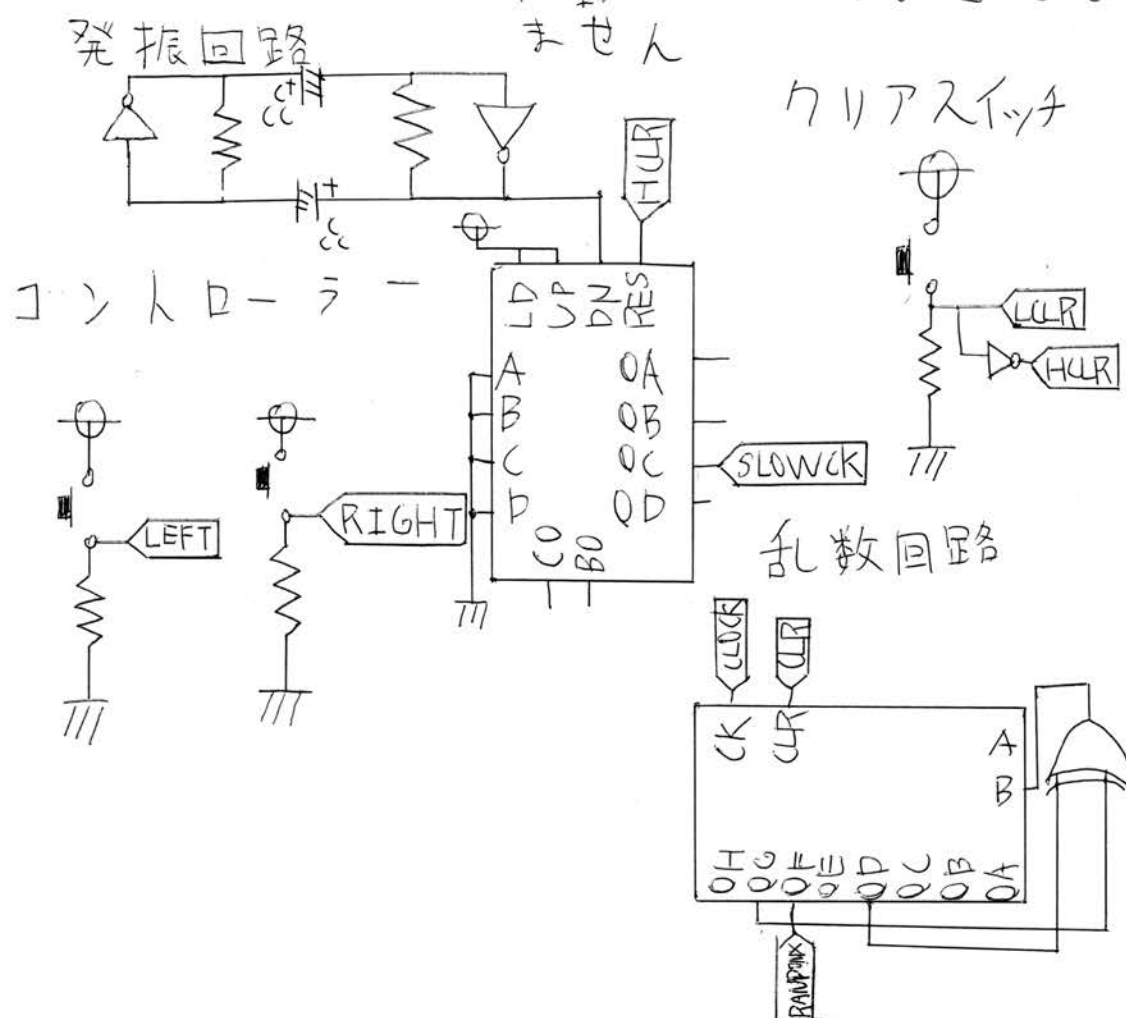
2. ストーリールール

一匹のよくばりなラビットがいました。ある日彼をねらって猟師がきました。猟師はいんじんをえさにラビットをうとうととしています。ラビットはうたれてしまうのかえ

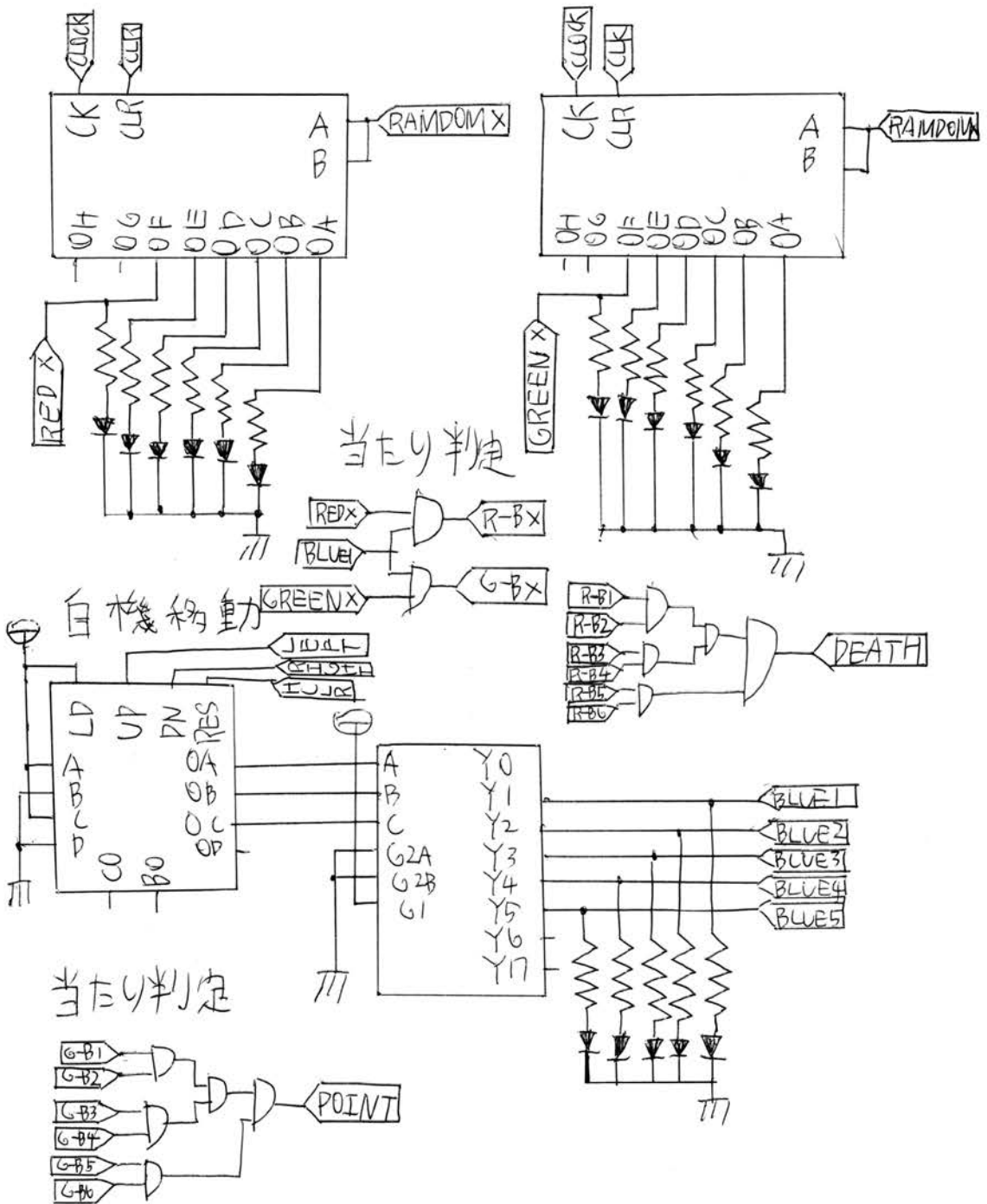
れともたくさんのにんじんをえるのか果
たして結末は！

ルールは上から赤色の弾と緑のにんじん
がおちてくるので青色のラビッツを動か
して、赤をよけつつ時間内に緑をとる
というものです。

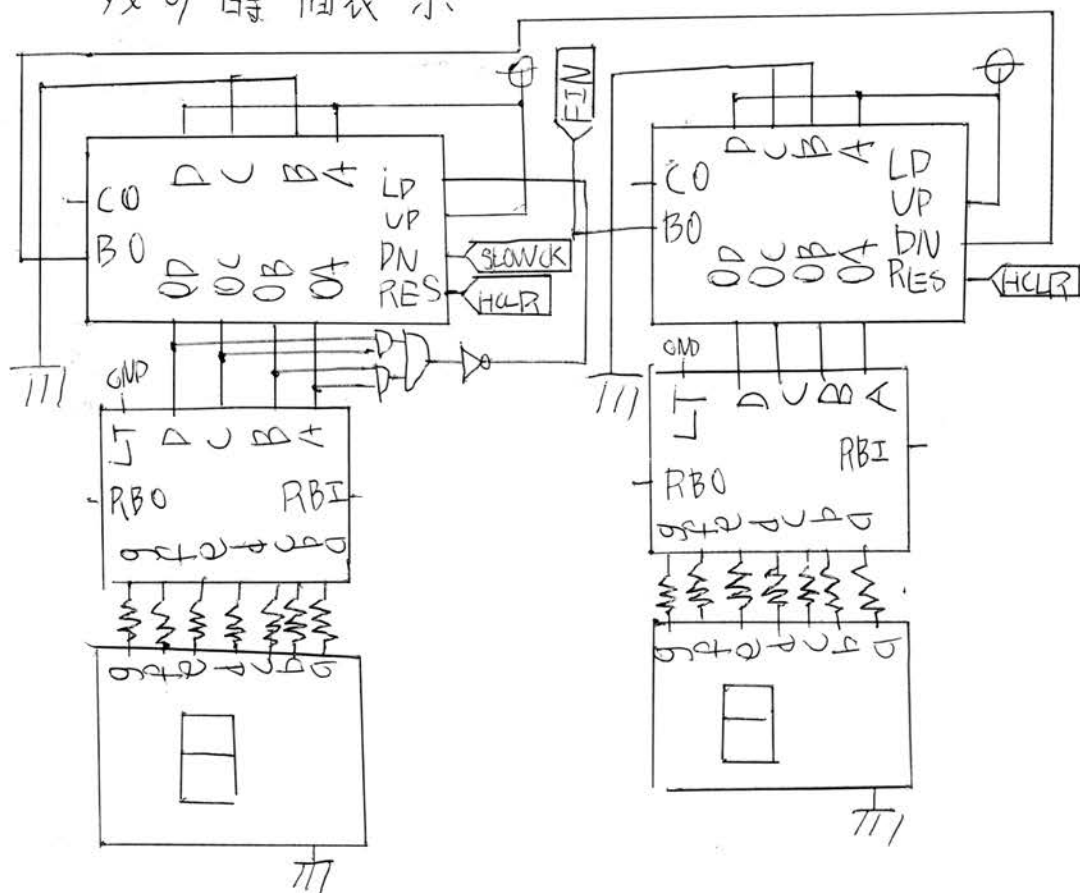
ろ、回路図※この通りにやっても必
ず動くわけではござい
ません



弾流し



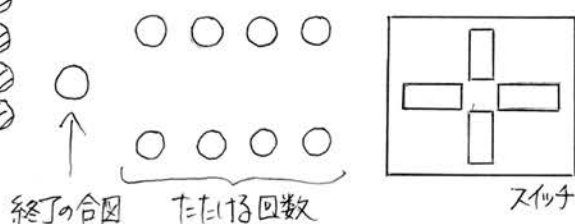
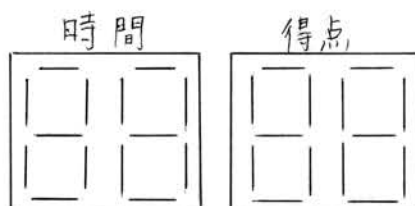
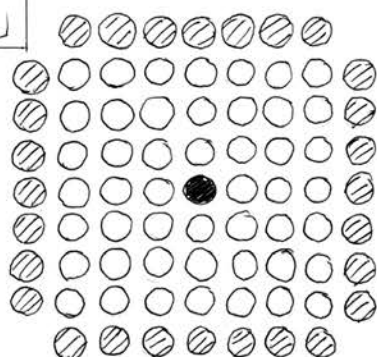
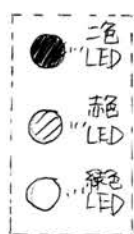
残り時間表示



モグラ大量発生

製作者：M2 野口
回路設計者：H1 小栗さん
協力：物無の皆さ

外観



ルール

赤LEDがランダムに光るのでプレイヤーを動かかし、赤LEDが光っている方向にスイッチを押します。すると一点入ります。緑LEDがプレイヤーの動ける場所です。しかし、たたける回数は8回までで、途中で真ん中を通ることで補充することができます。制限時間は60秒です。

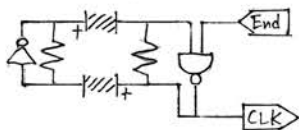
ストーリー

ある日、ある人が、道を歩いているとモグラに会った。モグラに会ったその人はモグラに近寄ってしまい、モグラになってしまった。モグラの人類征服はとん

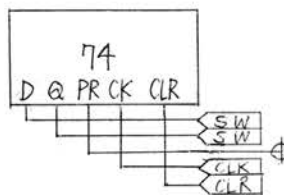
どんどん進んでしまい、人間は地球上で、その人の息子だけにな、
てしまった。息子はT=ま=まハンマーを手にとっていたため、モグラを
全て倒すことを決意した。

回路図

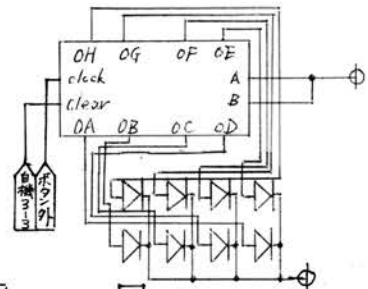
<クロック>



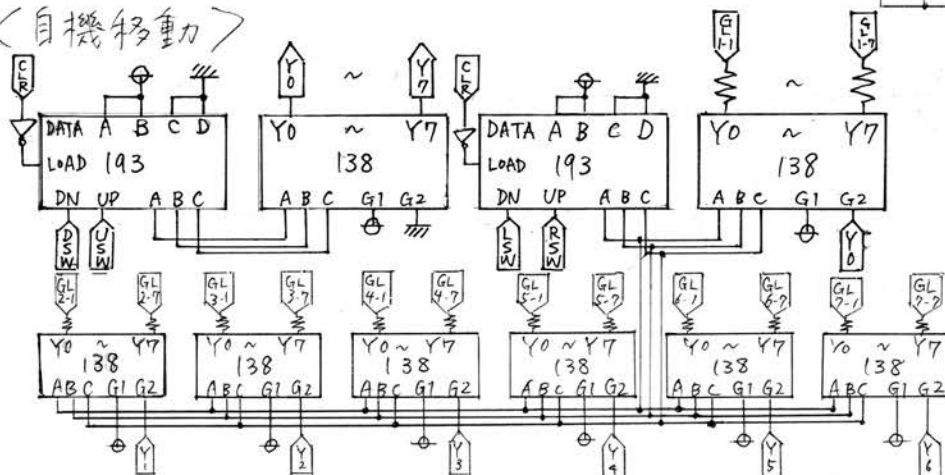
<チャタリング消し>



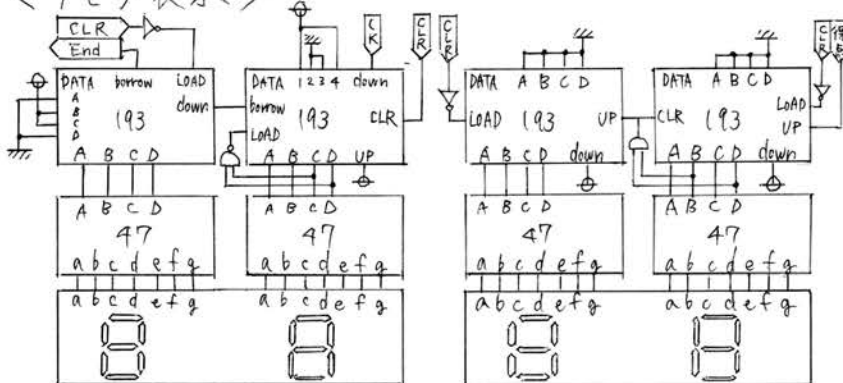
<たたき制限>



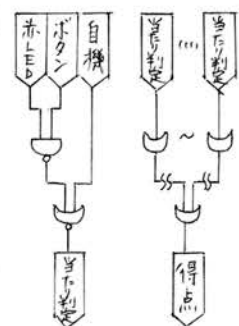
<自機移動>



<7セグ表示>



<当たり判定>



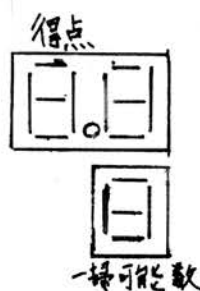
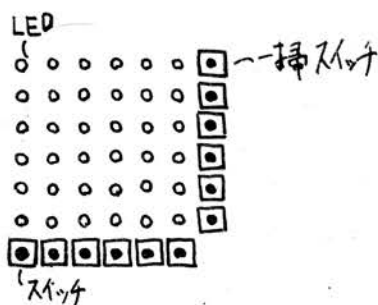
今回の製作で、物無の方々にはとてもお世話になりました。本当にありがとうございます。初めての製作で、こんなに大変なのか、と思いました。一番始めに考えていたゲームとほぼ同じように出来上がったので良かったです。来年の製作もがんばります。

干潟の狩人

制作: 久松正太郎(M2)

回路: 佐藤誓斗さん(H1)

Appearance



Rules

- ① スイッチを押してその列の赤い光を消すと1点
- ② 緑色の光に当たると減点
- ③ 一括スイッチを押すと横一列消せる。緑には当たらない
- ④ 一括スイッチは3回まで

Story

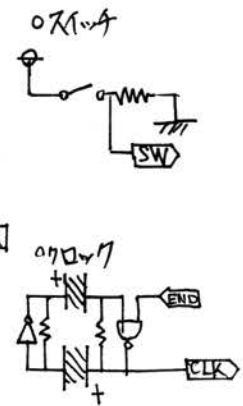
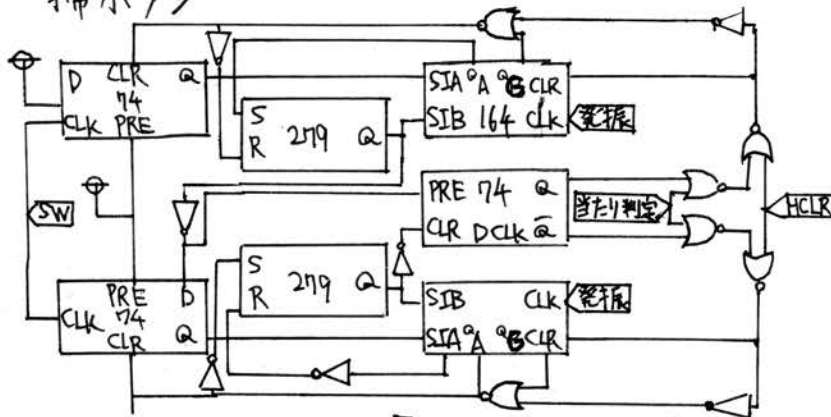
足に何かに触れる。オタマジャクシの死体だ。
 ここ東京湾ではそう珍しいものではない。
 そろそろ暑くなってきたこの時期、満月の今日は奴らは特に騒ぐ。
 人類は数年前海からの襲撃を受けた。

それは、アサリだ。私の仕事は人類に潮を吹きかける、

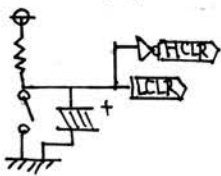
この忌々しい二枚貝を狩ること、則ち潮干狩りだ。
 海面から細く小さな水柱が上がる。奴らからの戦いの合図だ。
 私は熊手を構えた。

Circuit

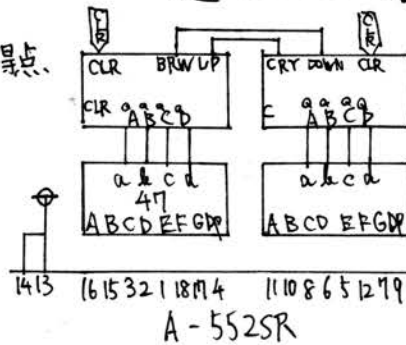
0-掃ボタン



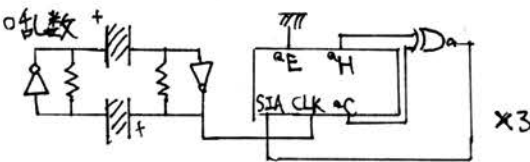
07047スイッチ



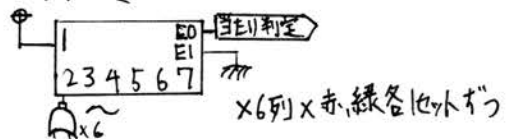
07047スイッチ



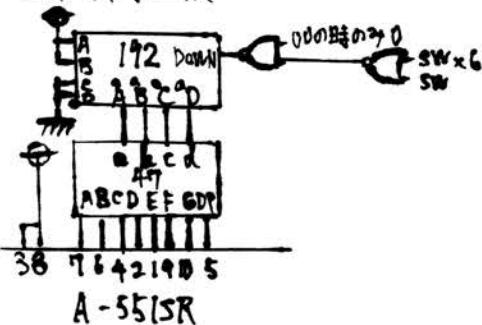
07047スイッチ



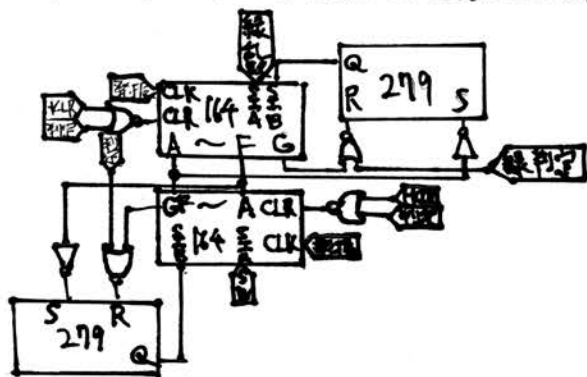
07047スイッチ



07047スイッチ



・降ってくる玉の回路(緑の玉と敵の回路)



+ 未定の部分があります。

回路図集の
提出

Impressions

今紙を見たら裏うつりして泣きました。しかし期限を大幅に遅れたので、書き直す訳にもいきません...というのが1ヶ月の反省です。そして、自分の回路に対する理解が全然にも浅かったです。そのせいで大きなミスをし、進行が通れてしまいました。

でも、やる時はかなり集中してやっていたので、その点は評価しても良いと思います。

僕の製作物は今年の中(ゲー)で最も複雑です。っていうのを完動しなかった時に言い訳にしたいのではないですね。

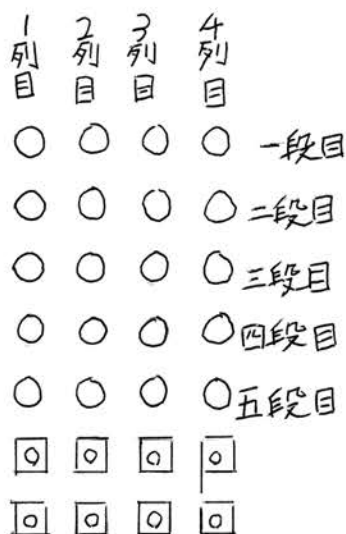
とにかく、完動を目指して頑張ります。

仕掛け リングと 青リング

制作・M-2 平岡
回路図 H-1 和田さん
協力 物無の皆様

・外觀

IC 等



・ ストーリー

ここは麻布の一角、そこでは、強制労働が行われていた。
ある日、労働者の一人がぬけ出そうとして寝返った。

(長官)なにをやっているんだ！

(労働)さんぼしてるだけ〜。

労働者にむけてつまようじが飛んだ！と思われたが、それはリングと青リングだった。

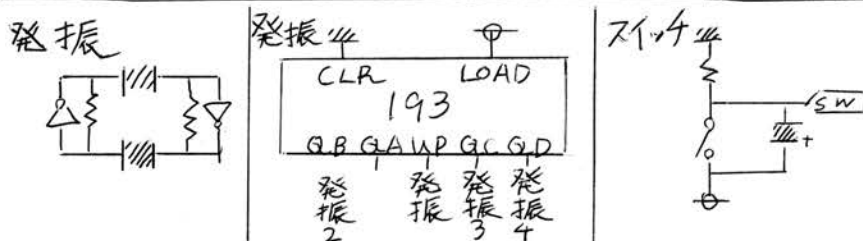
労働者は、大のうつき好きなのに、リングはきらいで、青リングが大好きという偏食を持っていた。

労働者は、青リングを食べ、リングを受けとめることができるのか？
(これは実話を基に作られたフィクションです。)

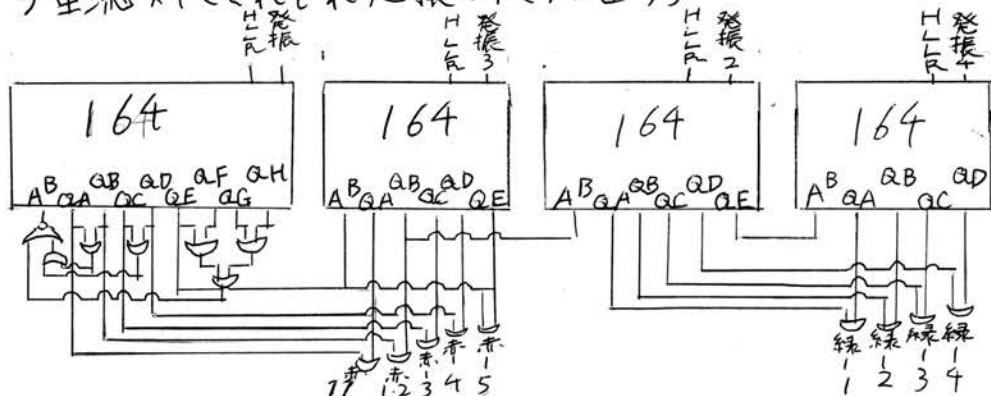
・ ルール

このゲームはタイムアタックで、30秒の間に獲得した得点を競います。

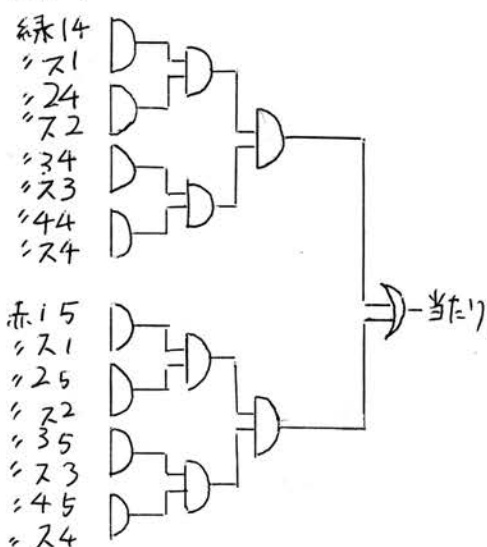
緑の玉は四段目に、赤の玉は五段目にきた時に、その列の対応する色のボタンを押すと、得点になります。



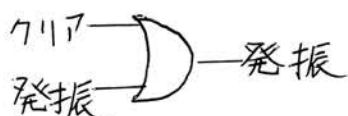
3 単流 x 4 (それぞれ発振の早さが違う)



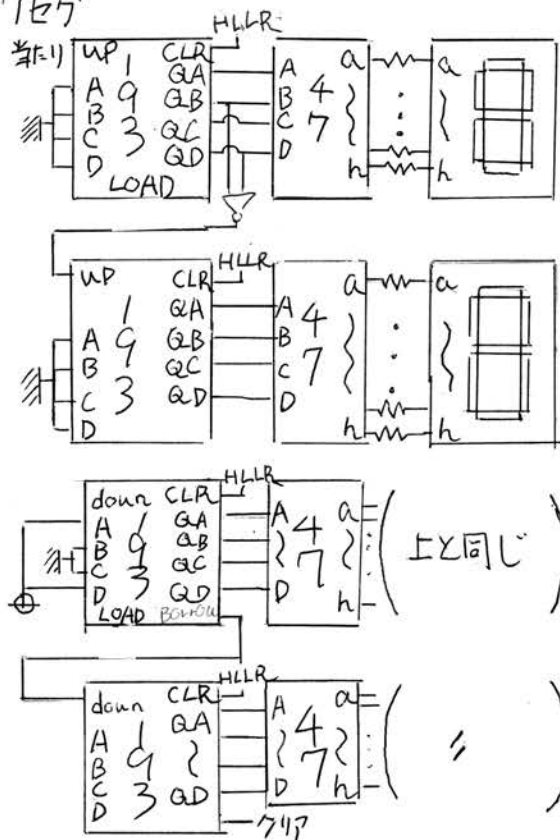
当たり



クリア (全体が止まる)



7セグ

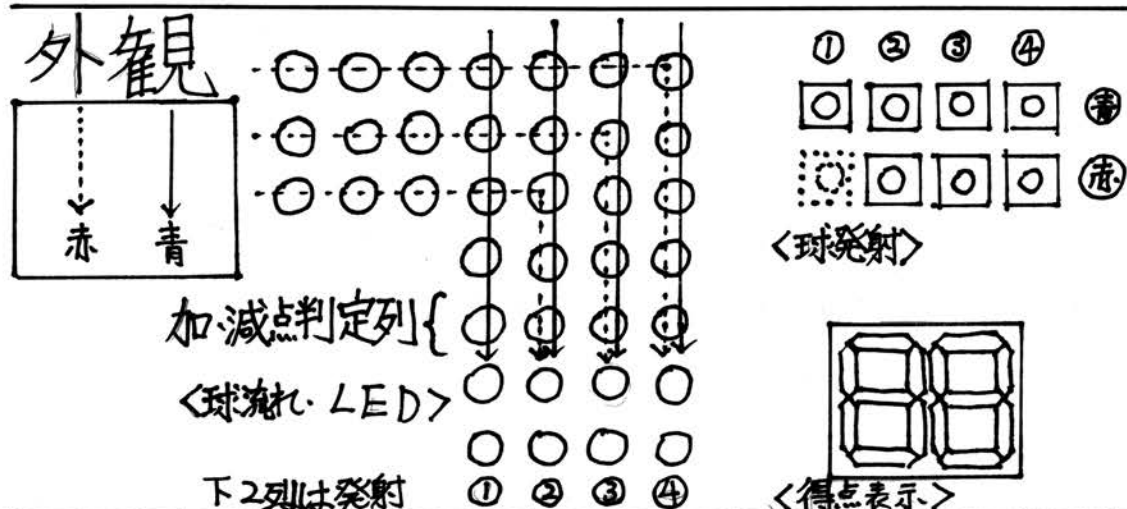


感想

- 製作の際、多くのミスやバグを起こしてしまい、時間がかかってしまったので、これからは、ミスを減らしたい。
- 製作に積極的にとり組めなかったのて、これからは積極的にとり組みたい。

レゴ ゆーし

製作者: M2 藤井
回路図: H1 中村さん
設計者: H2 渡辺さん
協力: 物無の皆さま



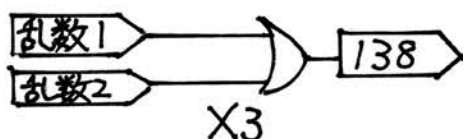
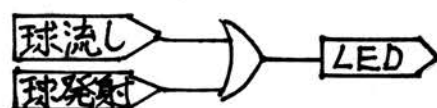
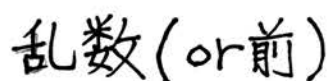
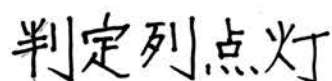
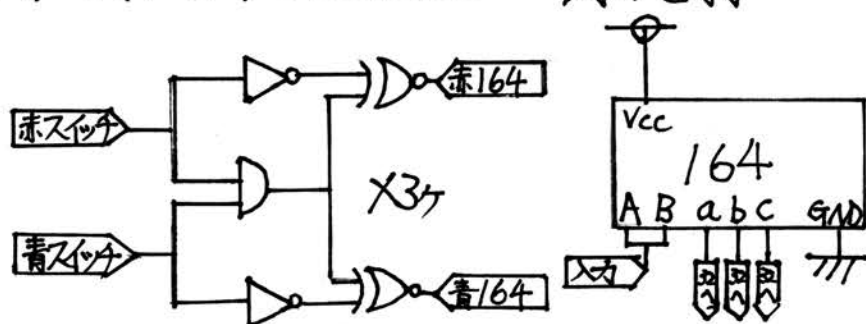
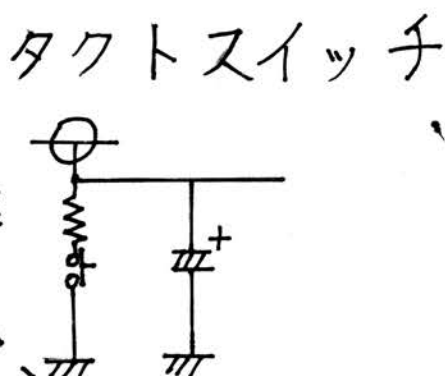
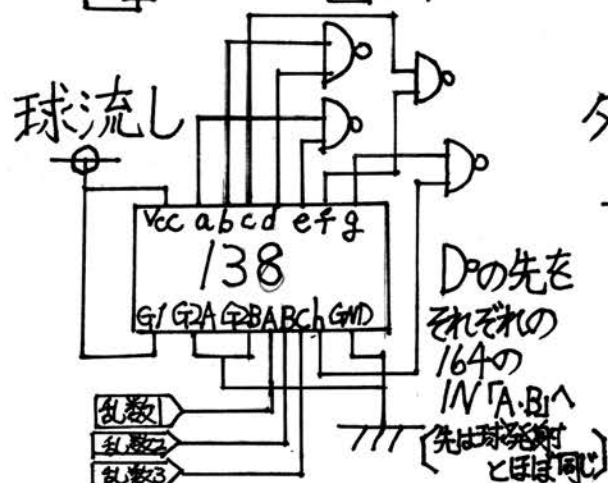
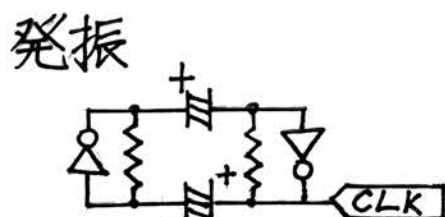
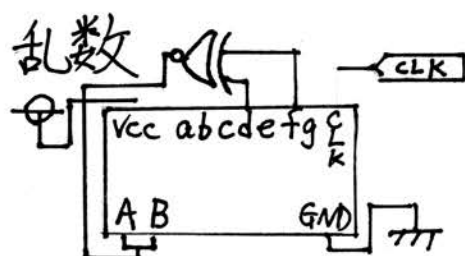
ルール説明

上4列から流れてくる青球と左上3段から流れてくる赤球を、下にある対応する列と色のスイッチを押して発射し、下から3段目で一致させることで加点します。赤と青、青と赤を一致させることで減点してしまいます。20点取ることですクリアです。GAMEOVERは存在しません。

ストーリー

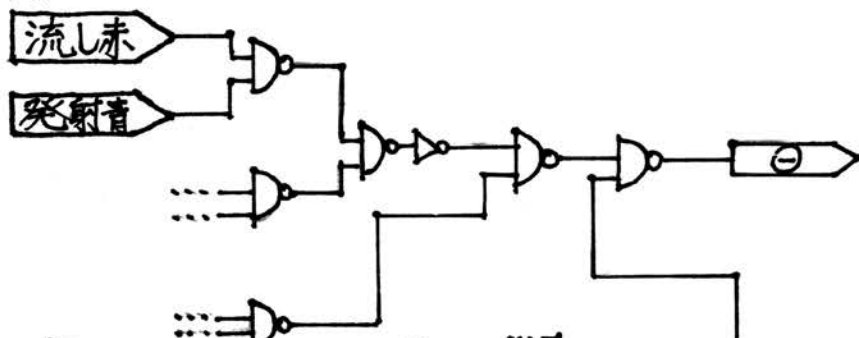
僕は伝説の勇者とその仲間ある町に入ろうとした瞬間、仲間の勇者が突然倒されてしまった。町の関には大量のモンスターが！
「くっ！お討ちだ〜！」「待て！俺の盾がな〜」そう忘れていたこのパーティー、僕以外1人の盾要員だったんだ。「グヒャッ」「あー！」
果たして僕だけでこのモンスター全員の攻撃を防ぎ町に入れるのだろうか？

回路

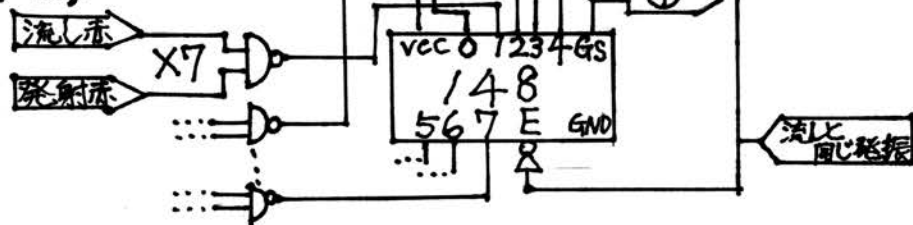


当たり判定

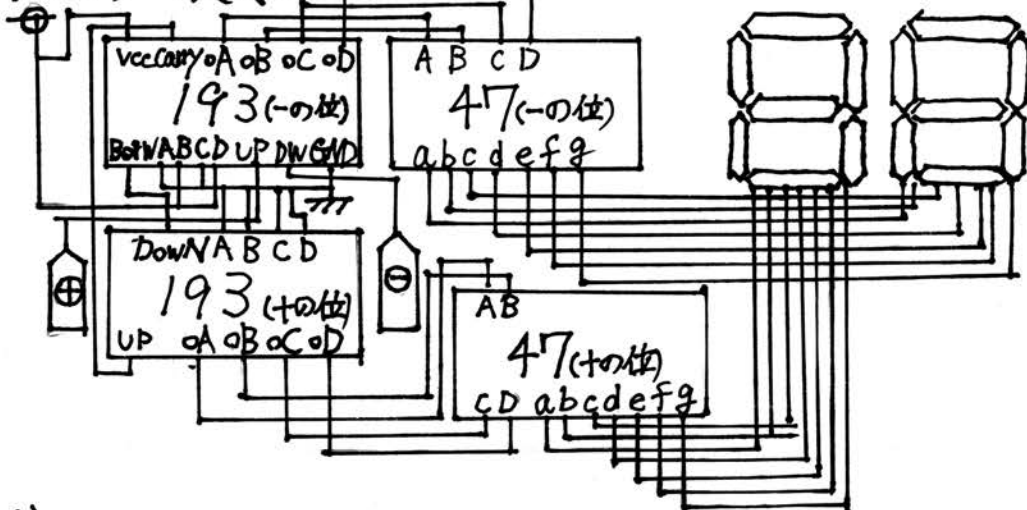
<減点>



<加點>



7セグ表示



感想

最初の方に速さ重視で汚く配線したこと
で回路図設計者さんに迷惑をかけてしま
いました。もと技術を高めたいと思います。

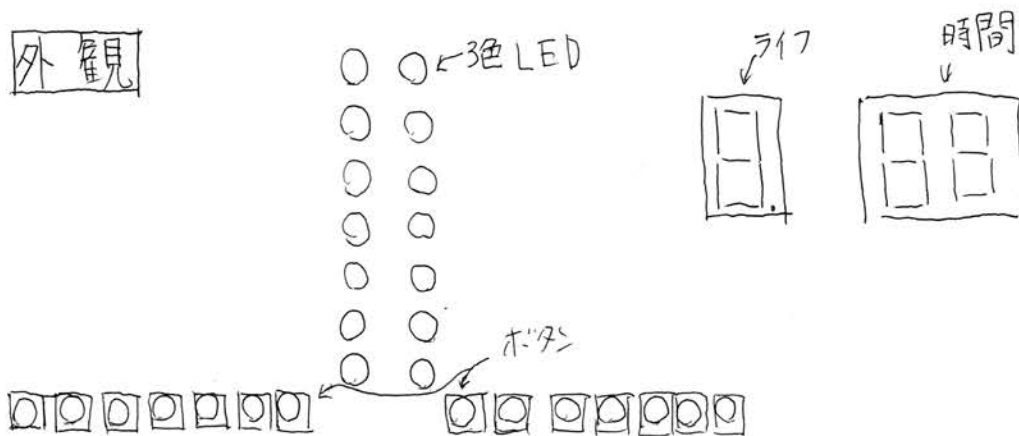
はじめてのかいもの

製作：M2 森田

回路図設計者：H1 関根さん

協力：物無の皆さま

外観



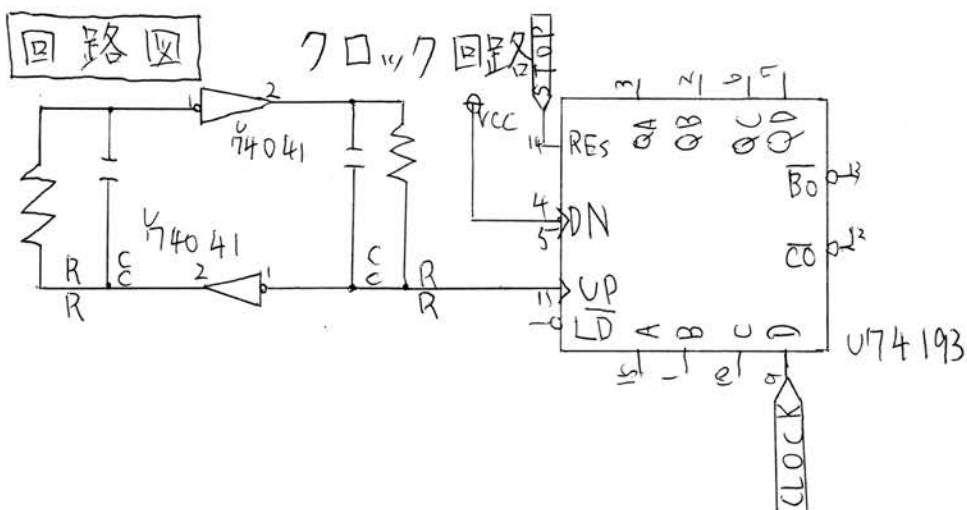
ルール

上から赤青緑黄水紫白の光が流れてきます。それに対処したボタンを押し、失敗するとライフが減ります。また、光っていない時にボタンを押すとライフが減ります。ライフが0になるまでの時間を競います。

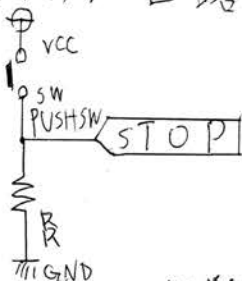
ストーリー

毎夕飯の材料のトマトとサバとレタスと
カボチャとお水とブドウとヨーグルトを
100円で買ってきても

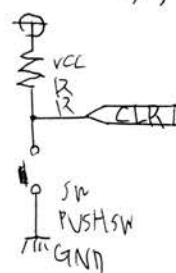
息子 1? 1? 1?
1~33
娘 1? 1? 1?
1~24



ストップ回路

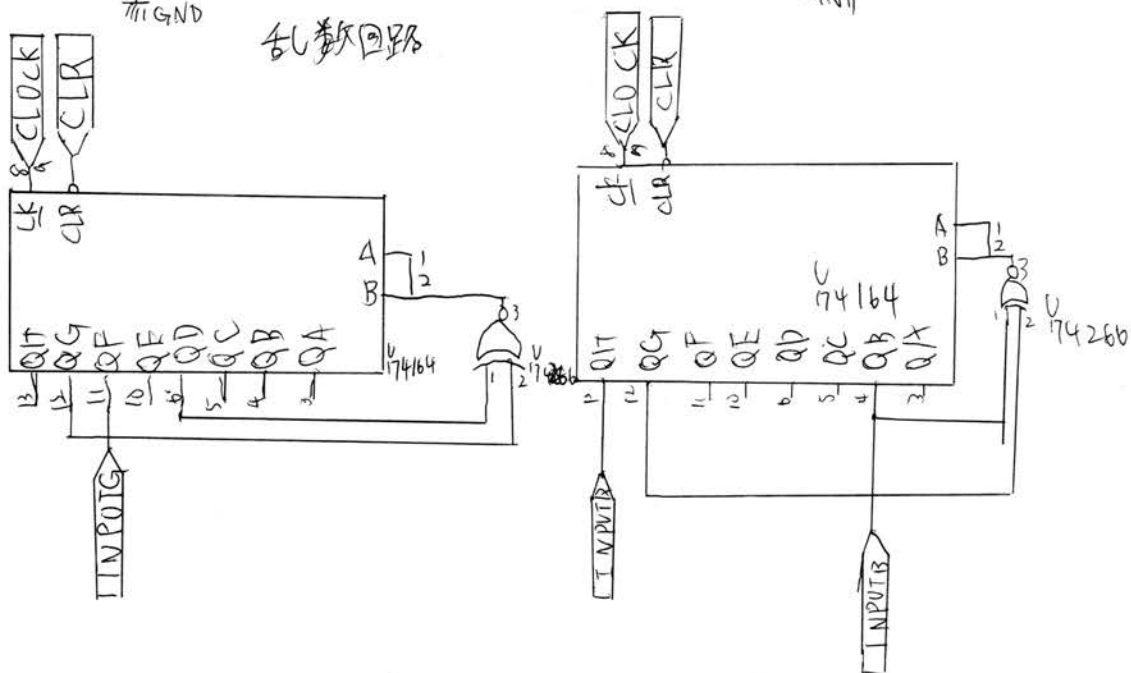


クロック止める回路

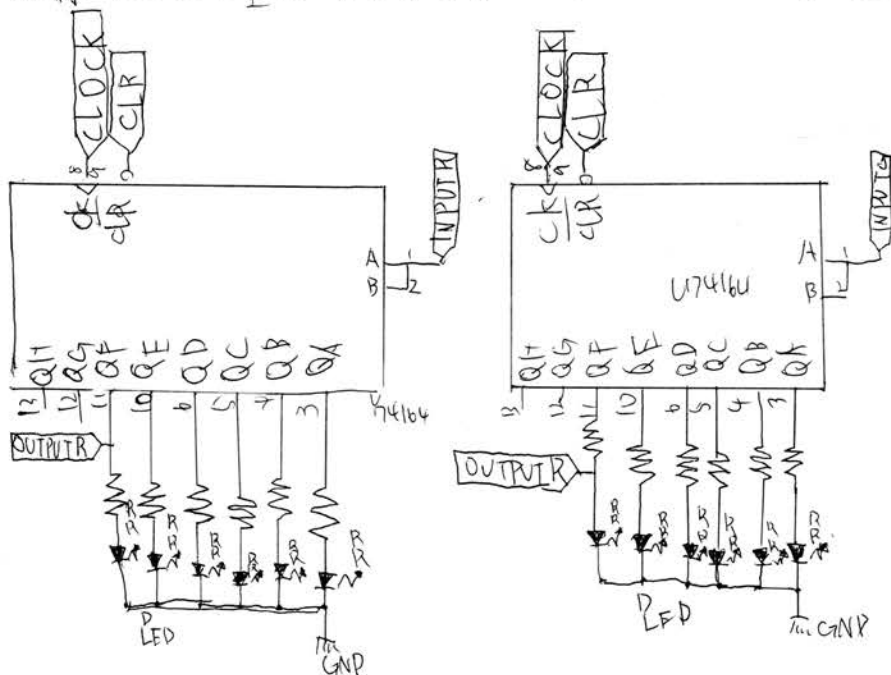


74193回路

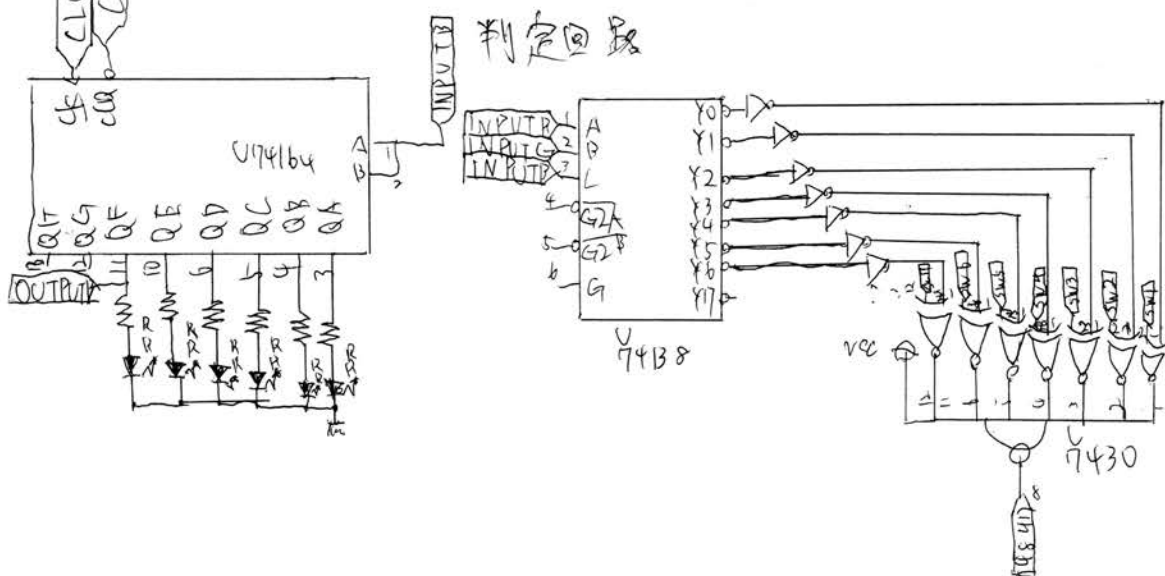
乱数回路



LEDの表示部分、左から赤、緑、青になっています。
 実際は3色LEDですが、分かりにくいので独立させて



2列あるので“2倍ですが”、省略させていただきま



IPRA BOU

製作者

M2 小島 大

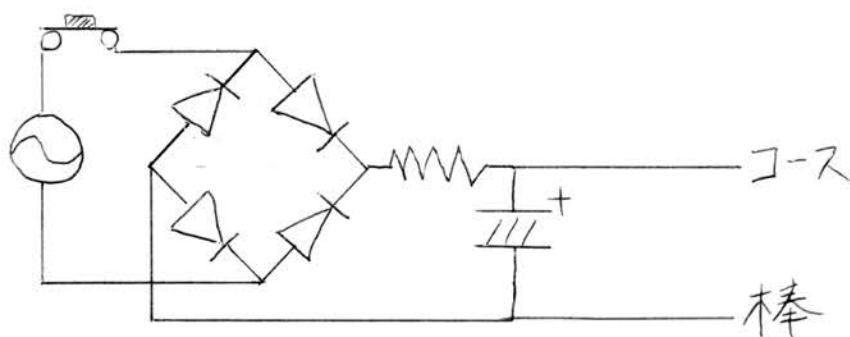
織田 隼一郎

協力物無の皆様

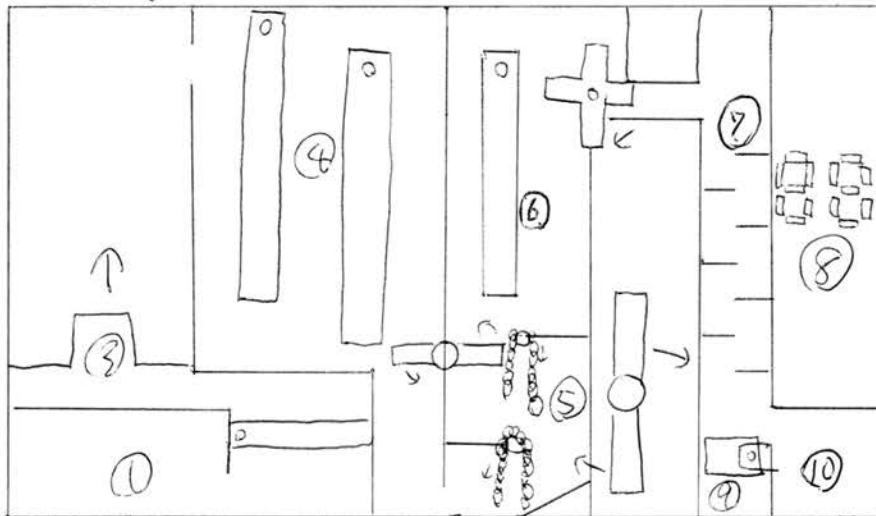
ストーリー

宇宙旅行をしていたらいきなり襲われ、閉じこめられた。様々な罠をくぐりぬけて脱出せよ！

回路図 (フラッシュ回路)



コース



- ①スタート…全ての始まり。囚われていた場所。
- ②③とびら…注意して持ち上げる！
- ④細い道…ただの壁ではないぞ…
- ⑤シャッター…タイミングよく通り抜ける！
- ⑥細い道…さらに細い道。
- ⑦ネジ山…注意して通すだけ。
- ⑧タイマー…カウントが60になると…
- ⑨閉さ…はやくしないとゴールがとじる。
- ⑩出口…や、たね、やと出口！

感想

小島 初めてのプログラミングで莫大に楽しかったのですが、楽しかったです。進行が速くなりましたが、クオリティの高いものを作れてよかったです。

織田 モチベーションが下がったこともありましたが、金属加工などは楽しかったです。始めてでしたが本気で良かったです。手伝って下さった先輩方には感謝の言葉しかありません。

物売

500円 チョップ



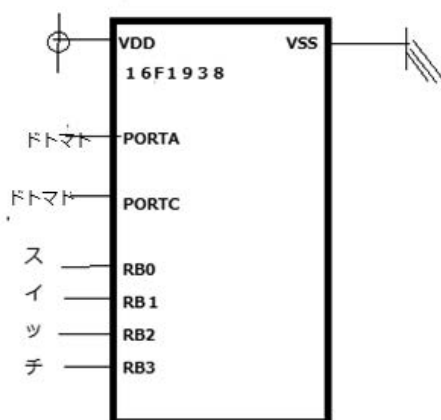
Sodom and Gomorrah

中3 高橋

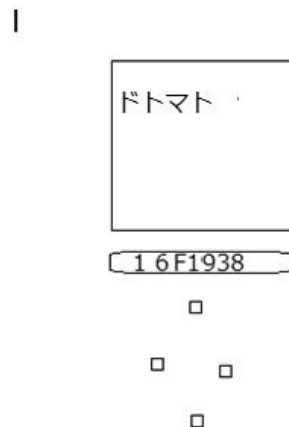
今年の売り物は落ちてくる玉を避けながらゴールを目指すものです。

ステージが現時点では7つあり、ステージがあがるにつれて落ちてくる玉の量と速さが上がって行きます。ALLCLEAR と表示されるか、GAMEOVER と表示されるまで、ゲームは止まりません。また、GAMECLEAR と表示されたあと、次のステージにすすむためには、下のスイッチを押す必要が在ります。

回路図



外観



感想 関根さんと渡辺さんに随分助けられました。感謝したいです。
機会を与えてくれた庄田にも少しは感謝しています。

チカチカ

M3 飯島勇人

・チカチカとは

チカチカ(本名=自走マルチバイブレーター)とは、2つのLEDが交互に輝く、その名の通りの物です。その原理は、電池から供給された電気がコンデンサという物に溜まり、その電気がLEDに流れ、それが2つ交互に行われる、というものです。容量の違うコンデンサを使えば、点滅の速度も変えられるわけです。仕組みはなんだか難しそうに見えますが、制作は半田ごてを持ったことのない方でも可能です。

↓回路図です。

