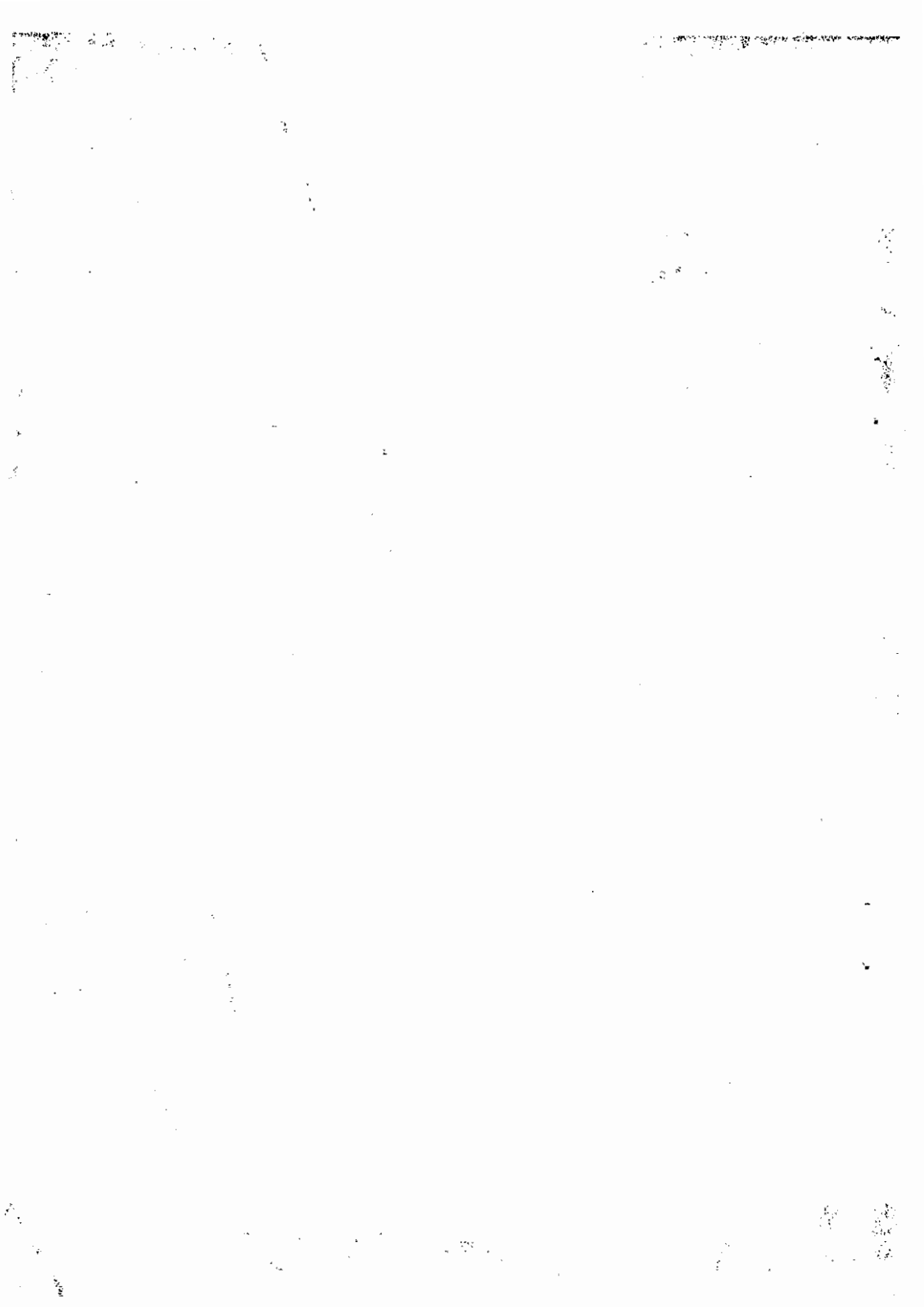


麻布の
愛した
物理

2014 回路図集



は い め に

昨年の回路図集では「三次元集積技術が活発に研究されている」とお伝えし、最近ではCPUとGPUをユーザーから見ると統合的に扱う事ができるHSAに向けたAMD社の製品がintel社の製品を最大で約1.4倍の処理能力で上回ったりと「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ります。

他方我が国でもKinectを始めとした今までに無い新たな媒体、素子を用いる事でさらには一步先のステップへ上がろうとしている。

さて、この回路図集は麻布学園物理無線班の部員が一年をかけて研究、製作してきた物の結晶が記されています。中には未完成のものもあるため本書の通りに作っても上手く動かない可能性がありますので予め御了承下さい。

H2 会計

四柳 右衛門

訂正

目次において間違いがありましたので訂正いたします。

× : セラフィムの小夜曲

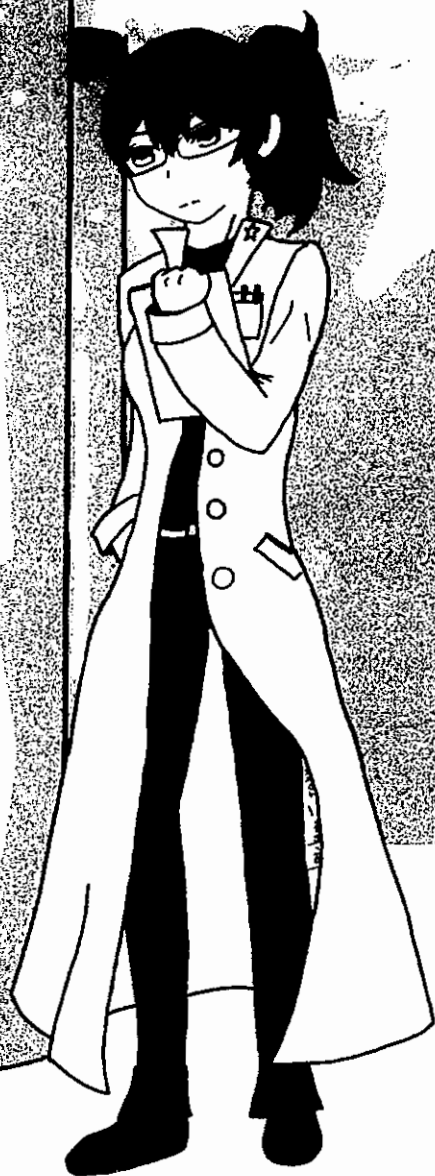
○ : Mii ~ミィ~

はじめに	P. 01	
目次	P. 02	
コンピューター系	P. 05	
Responsive Forest	P. 06	Team ファラビット
Let' s enjoy life with “server”	P. 07	H2 佐藤
Butsumu Sword	P. 08	H1 石倉 M3 佐藤
ロボット系	P. 11	
無線操作式電磁投射砲塔試作 Y 型 — code name γ —	P. 12	H2 四柳 H1 渡辺 M3 関根
BTMY	P. 22	H1 山本
くあこぶた～	P. 24	H2 佐藤
セラフィムの小夜曲	P. 28	H1 市川 M3 福島
ぐるぐるぐるりん	P. 32	H2 黒田
DEATH STER ～FINAL DISASTER	P. 34	H1 横尾 M3 小栗 M3 和田
三代合体	P. 38	H2 花園 H1 佐伯 M3 川本

ゲーム系	P. 47	
Chicken DIVE	P. 48	H2 佐藤
びーこん	P. 52	H1 宮嶋 M3 磯村 M3 中村
さよならゴミの日	P. 58	M2 飯島
The Block	P. 60	M2 庄田
TMFM	P. 64	M2 高橋
ドラゴン襲来	P. 68	M2 野末
イライラ棒	P. 72	M2 飯島 M2 高橋
販売品	P. 74	
Silent Gamer 1999	P. 75	M3 関根
チカチカ	P. 80	M3 川本



Computer



Responsive Forest

Team ファラビット

メンバー H-2 佐藤
H-1 石倉 H-1 佐伯
M-3 磯村 M-3 中村

IVRC って?

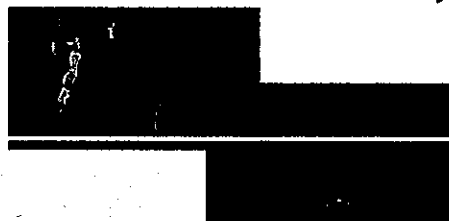
IVRC(国際学生対抗バーチャルリアリティコンテスト)は、学生が自らインタラクティブ作品の企画・制作・展示を行い、技術や芸術性を競うコンテストです。

全国のチームから予選本戦を行い、全国で最も有意義な VR(拡張現実)の作品を選出する大会です。本戦では外部から見学に来た人々にも投票を行っていただき、最優秀作品を決めます。

ResponsiveForest は、物理部無線班初の IVRC(高校生部門)参加にして全国3位を獲得しました。

これはなに?

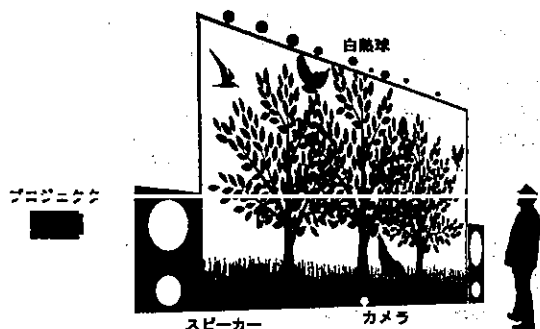
某国民的アニメのワンシーン、「人が踊って木が大木になる」というファンタジックなシーンがあります。キャラクターたちが歌いながら力を込めて踊っていると、その踊りに呼応するように、畑からは芽が出て、それがグングンと育ち、しまいにはとてつもない大木になってしまふ。誰でも思わずワクワクするあのシーンを実際に体験してみよう!というのがこの作品の趣旨になります。



どうやって動いているの?

この作品は大きく分けて、3種類の部位から構成されています。

1つ目は人の動きを読み取る Kinect 部分。2つ目はそのデータを処理し、木を生成するプログラム部分。さらにはそれを出力するプロジェクター類です。



Kinect は Microsoft の XBOX360 用デバイスで、マークなどをつけることなく人を認識し、その動きを追跡することができます。

PC 部分では Kinect のデータを読み取り、人の関節の動きを計算し、木を生成させます。木が成長すると動物がよってきたり、鳥がさえずりをはじめ、草が生えて森が成長していきます。これには Processing と OPEN NI(Kinect のミドルウェア)を用いています。

大会側からは Unity というゲームエンジンの利用権限提供打診もいただきましたが、今回の作品ではより自然な森の生成を行うため、

Processing にて描画部分から作成を行いました。

た。作品内で使われている動物のモーションデータはメンバーによる手書きです。

PC はそれに合わせてスピーカーを制御し、白熱球で日光を表現し、加湿器によって森の空気を表現します。これらによって、自らの動きに反応して育つ自然の森の成長を体験することができます。

チームメイトの感想つぶやき

佐藤: マングバンプの香りに誘われて

石倉: やすやすとはったぜ

佐伯: 帰りのポケセン楽しかったー! 閉まってたけど

磯村: わしわしはったねん

中村: まあぼくのおかげですよ(木漏れ日ピザ)

参考ページ

IVRC 2014 オフィシャル <http://ivrc.net/2014/>

Let's enjoy life with "server"

H-2 佐藤 惇

今年度、物理部無線班では部活専用のサーバを契約しました。この機会に Butumu.com というドメインも取得、チカチカのお客様ご案内にも使われています。ここでは部員向けの意味も込めて、とても簡単なサーバの基礎について書いてみたいと思います。

インターネットを使う上で、ユーザは PC を通じて世界中のページを見に行くことができます。このページをがいつでも見られる以上、だれかが年中無休でそのページを提供しているわけです。それが世界各地に散らばったコンピュータ、特にサーバと呼ばれるものなのです。

サーバも基本的には普通の PC と変わりません。その違いは外部からアクセスできるかということだけ。(レンタルサーバはまた少し異なりますが、そこを利用者が意識することはあまりありません)サーバにも家で使うコンピュータと同じく OS が入っていて、その中で様々なソフトが起動することでサーバーとして使うことができます。

ここで使うための OS としては Windows のサーバエディションである Windows Server もあるのですが、お値段もそれなりにするので、自宅で使う場合やレンタルサーバの場合は Cent OS という無料の Linux OS を使うことが多い

です。最近では OS ごとの差を気にしなくても使えるようなソフトが増えていますが、ある程度の操作をしようと思うとコマンドなどが必要になるため、しっかりと運用するには Linux の操作感に慣れておく必要があります。



OS が入っているだけでは外部からアクセスできず、まだただのコンピュータです。そこで Apache-Server というものをインストールします。これはサーバがサーバとして他のコンピュータと通信するためプロトコルなどを制御したり、外部から接続をするための窓口を開く(ポートを開くと言います)など様々なことを請け負ってくれます。これで、最低限の機能が準備できました。



XAMPP

ここに PHP や Java などをインストールすることでさらに機能を追加できます。レンタルサーバではこれらは標準インストールされていることが多いですが、個人でサーバーを作る際にはこれらのインストールから始める必要があります。

XAMPP(LAMPP)といってこれらのインストールや設定を一括して行ってくれるソフトもありますので、それらを利用してもいいでしょう。ここでの設定を間違えるとセキュリティ上大変まずい事になります。ユーザーが多いこともあり、情報も豊富なのでそれらを参考にして確実なインストールを行いましょ。

さて、これらの準備ができたあとはサーバの中身を作っていきます。一昔前のように HTML に載せた文字と写真だけのホームページでも問題はありませんが、最近の流行は動的なページを作成することです。動的なページとはそのページを訪問する相手によって内容を変えたり、ページの訪問者にコメントを残してもらえようページのことをいいます。おもに PHP を使って作成します。PHP は HTML などを出力するプログラミング言語です。例えば物無のチカチカのページではお客様の QR コードからそのカードの番号を調べ、プログラムの中でお客様の案内状態を検索し、その結果を HTML の中に埋め込むことでお客様ごとにご案内を行えるようになっているのです。このように双方向のコミュニケーションをとるためには PHP のような技術が必要不可欠です。



QR コードの例
(チカチカ用ダミー)

総括

サーバーは全世界に公開されるものですので、使い方を誤ると大変なことになります。しかし逆に正しく使いこなすことができれば、大変便利なものです。また、実際にサーバーを運用することがなくとも、その仕組みを知っているだけでも正しい危機管理ができるようになります。こういったことに興味をもって生活するのもこのインターネット社会において大変重要なことだと思います。

Butsumu Sword

製作者：H1 石倉 匠
M3 佐藤 慧斗
協力者：物無の皆様

1. ストーリー

「ありがとう。ミヤジマ。」
イイジマは少し盛り上がった土の前で合掌した。
ついに一人になってしまった。
もう何人の仲間が死んだらうか。もはや数えることすら諦めてしまっていた。
20xx年、世界は二十年前に生まれた魔王軍の手によって滅ぼされかけていた。
三年前、すでに真っ向から戦っても勝ち目がないことを悟っていた人類は
禁忌とされていた“STAP”を使用することを決定した。
こうして“STAP”によって生み出された禁断の人間の一人がイイジマなのだ。
STAPを実行した科学者ナカムラはせめて彼らを苦しめないために、
二十年前に悪魔を生み出したあの失敗を繰り返さないために、
彼らには“心”を与えなかった。
しかし、今、イイジマの体には“心”と呼ぶべきものが宿っていた。
仲間を失う代わりに得たものが“心”なのだとしたら…
もし、その“心”がこの世にあってはならないバグであるとしたら…
彼の戦いは続くのであった。

2. 概要

どうも、石倉です。ストーリーとは温度差がすごいです。このまま行きます。今回作ったのは、アクションゲームです。プレイヤーがアクションするやつです。まあ簡単に言うと、かの超良ゲーとも呼ばれた「ドラゴンクエストソード」的な Something をつくろうとの考えのもと作られたゲームなんです。画面の前でプレイヤーが剣を振るとあら不思議！ 画面の中の敵が切れちゃうんです。この前まで74LSシリーズを扱っていた僕らにはありえないほど最新の開発環境、Unityを使ったゲームなのです！
今回やっていることは以下に分けられます。

その一、ゲーム設計

判定を入れて切られた敵がダメージを受けたり、敵を倒したら次の敵が出てきたり、そういうことをやります。これは、すべて Unity で行いました。Unity は後ほど説明します。

その二、キネクト

Xbox のハードウェアにキネクト、というものがあります。Kinect です。今回はプレイヤーの動きを読み取るためにこれを使いました。キネクトについても後ほど説明いたします。

その三、キャラクター設計

3Dキャラクターをつくりました。Blender などを使って作りました。まあ説明することも特にないので。

…てかなんにも説明してねえじゃん

3. 回路図

ありません！ナイ！？エエ！？ナンデ！？

今回はPCとキネクトをつないで、あとはそれをプロジェクター等に映し出しているので配線はしてないわけです。

4. Unity

多分真面目な説明はここからになります。

さて、「Unityとはなんぞ？」ということから触れていきたいと思います。

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」するそうです。(Unityホームページより)

つまり、Unityという凄腕ツールによって僕らのようなただの物理部員にも比較的楽に重力とかを考えたゲームを作ることができるようになったのです。パソコンの中にバーチャルな3次元の世界を作り出しその中でゲームをつくれる、と考えれば良いかもしれません。

たとえば、矢印キーをおしてステージを傾け、上にあるボールを転がしてゴールまで運ぶゲームとか。または、弾丸を打って的確に当てるゲームとか。そんなゲームが出来てしまうのです！すごいでしょ！Unity！ありがとう！Unity!!!

*「びーこん！」のページも見てください。こっちよりぜったい詳しいです。

5. プログラミング

今回 Unity でやったことは以下の通りです。

- ・斬撃を創りだして敵がダメージを受けるようにする
- ・自分のHPとか敵の出現などのメイン的要素
- ・敵の攻撃パターンを決める

ざっくり言えばやったのはこれくらいです

1. 斬撃を創りだして敵がダメージを受けるようにする

4/24日時点では以下の様な方法をとっています。

プレイヤーの右手の位置を読み取る。

ゲーム内の、「現実世界でプレイヤーの右手がある位置」と同じ場所に

0.1秒ごとに玉を作り出す。

その玉に触れた敵がダメージを受ける。

どこまでも簡単にいえば、「現実世界でのプレイヤーの右手の位置にいるバーチャルの世界での敵にダメージを与える」てきな？です。



Cut!!

説明難しい……

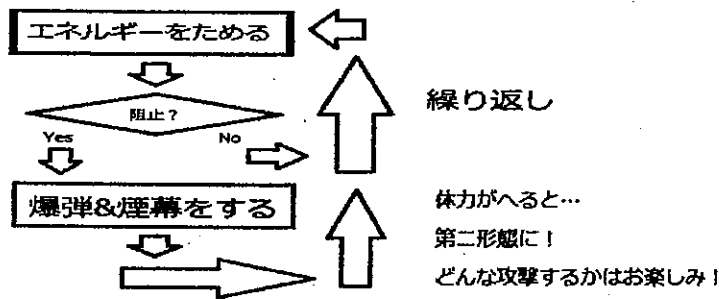
2. 自分のHPとか敵の出現などのメイン的要素

これは特に何も特別なこともしてないです。

敵が倒れれば次の敵が出てくるようにしました。

3. 敵の攻撃パターンを決める

いろいろありますが、ラスボスの攻撃パターンを見てみましょう。



6. キネクト

今回キネクト担当は佐藤くんだったのですが、彼いわく「特に書くことない」そうなので、とりあえずウィキペディアから引っ張って来ました。

Kinect(キネクト)は、(物理)コントローラを用いずに操作ができる体感型のゲームシステムで、ジェスチャーや音声認識によって直観的で自然なプレイが可能となる。

開発コード名はProject Natal。

動力学を意味する“Kinetics”と、繋がりを表す“Connection”を組み合わせた造語である。

キャッチコピーは「カラダまるごとコントローラー」。

マイクロソフトが提唱する「ナチュラルユーザーインターフェイス(NUI)」の一つ。

(wikipediaより)

あらかっこいい!

つまり、キネクトの前に立った人間の骨格を読み取り、その人がどんなポーズしているかを特定してくれるのです!

「キネクト」で検索すればどういものか出てくるでしょう。正直画像を載せていいか微妙なところなので、ごめんなさい。

かんそう

石倉 今回、後輩のゲーム作りを見てあげながらの製作であり、また、Unity という全く新しいものに取り組んだので、なかなか大変でした。

まだ終わってはいませんが、展示することができそうです。

佐藤 今年初めての自由製作でした。

感想としては、充実した一年を過ごすことができました。

3Dに関して全くの無知からのスタートだったので。

戸惑うことも多く石倉さんには迷惑をかけてしまいました。

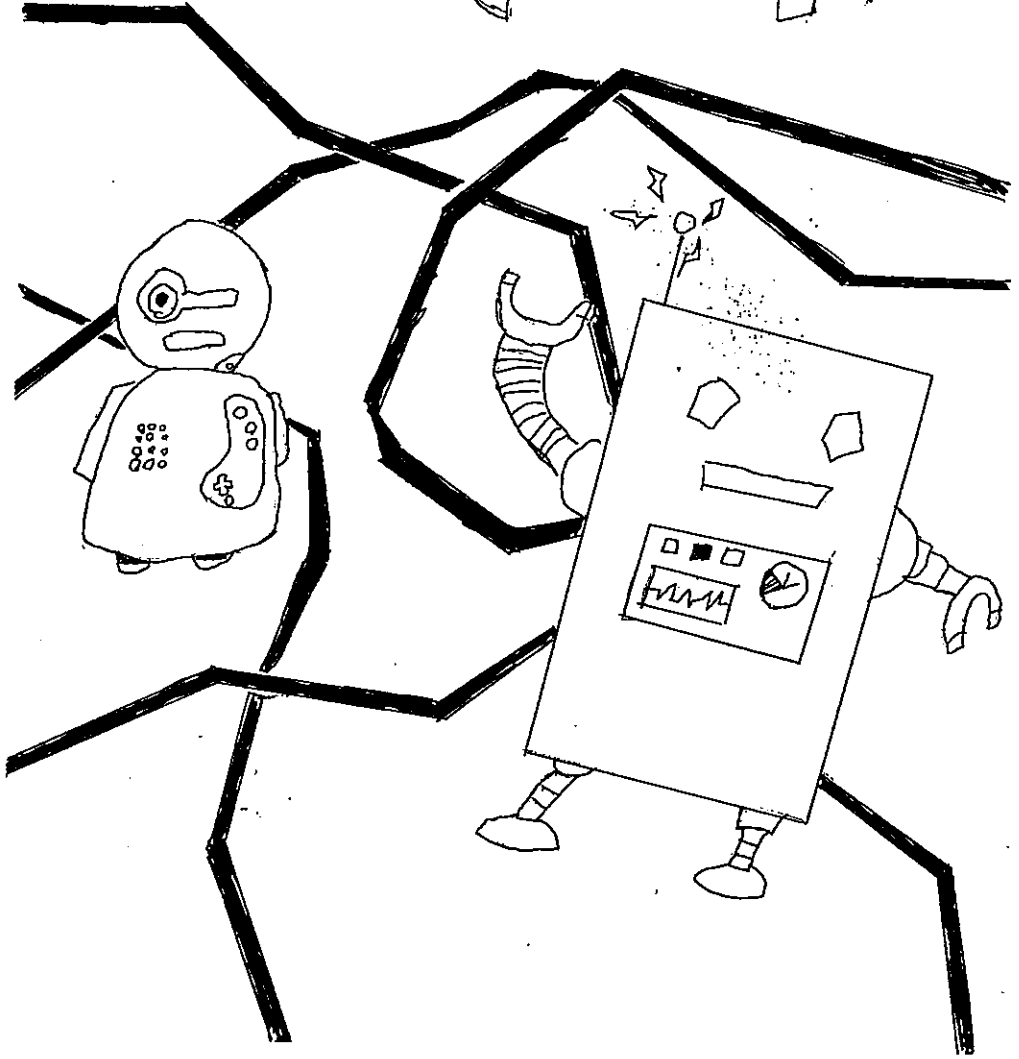
今年の製作の経験を通して次などに繋げていきたいと思えます。

参考

Unity ホームページ

Wikipedia(キネクト)

口味人



無線操作式二足走行 電磁投射砲塔試作 Y 型

— code name γ —

製作：高2 四柳
高1 渡辺
中3 関根
協力：物無員の皆様

Before Story ～開発前日奇譚～

その日男達は困っていた。というのも、彼らには製作をするにあたって各々問題があったのだ。

男のうち一人は関根、人呼んでアナログの申し子。その知識と技術の高さには定評があるものの裏の顔は殺傷兵器の開発にいそしんでいるマッドサイエンティストで今回もアブナイ物を作るつもりだがそのままでは当然許可が下りないためカムフラージュを必要としていた。

もう一人は渡辺、人呼んで不倒の調整師。繊細な調整によって二足歩行を作り上げる匠だが今回はさらに一步踏み込んだものを作るために新たなアイデンティティーを必要としている。

最後の一人は四柳、人呼んで革新の探究者。前人未到の地へ踏み込むことを求めて挑戦を続ける一方で、決して100%には届かない彼もまた背中を預けられる仲間を必要としていたのだ。

…時は満ちた。三人が出会ったとき彼らの頭に思い浮かぶはただ一つ

「(この二人となら…!)」

かくして三人の男達の扉は開かれた

———さあ、行こうか———

外観並びに概要

いきなりストーリーで詐欺ってしまいました。脚本担当は全く反省も後悔もしていません。大いに反省も後悔もしています。現実はずっと淡泊でした。ここからは真面目な解説になるのでご安心を。

・リモコン

ここは無線通信担当の四柳がお送りします。

今回のリモコンは何とコードレス、ワイヤレスなんです!!!

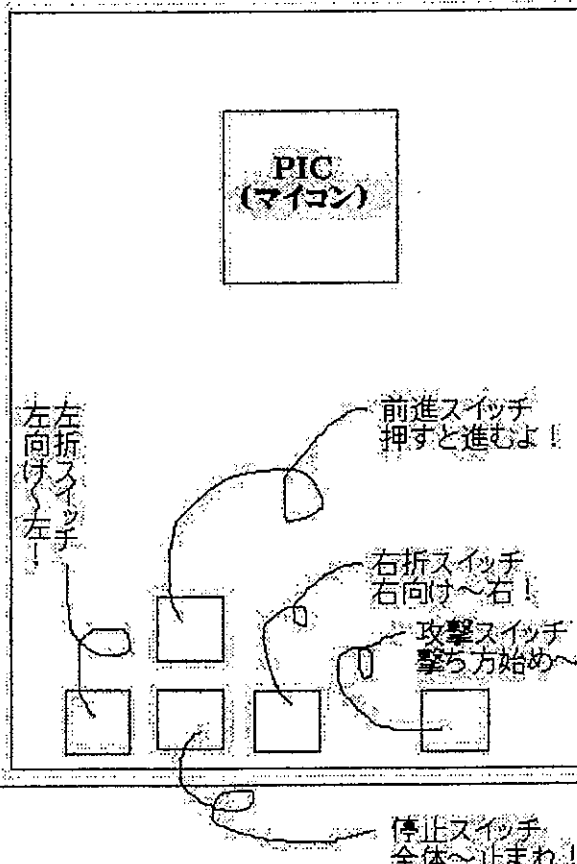
今回は無線で通信を行っているのでリモコンをどこまでも持ち歩くなってもできちゃいます。もっとも通信範囲がどれだけあるのか怪しいものですが…

通信自体の詳しい解説などはこの後ろの方に乗っているのに興味があればぜひご覧ください。以下にリモコンの模式図っぽいもの

のを載せておきます。

ペイントで書いたので見にく

いかもしれません。



なお全体と言っていますが↑一台しかありませんのであしからず

・本体の構造について

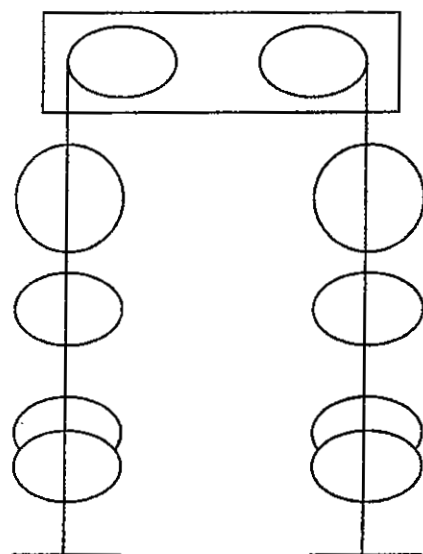
ここからは本体設計を行いました渡辺より。

僕は、去年も二足歩行ロボットを作りました。本体は去年と似ています。というより去年のものの改良です。

今年のロボットは去年の「膝なし」という特徴を捨てまして、サーボモーター10個で動かします。電池を積んでいるのでコンセントに繋ぐ必要もなし！無線駆動なのでコントローラも投げやすい！ロボットに絡まる心配もなし！という素晴らしいロボットです。以下に今現在のロボットの構造を載せておきます。まだ受信機も付けていませんし、コイルガンもあるので、文化祭時には変わっていると思いますが。



こいつが今年の二足ロボ



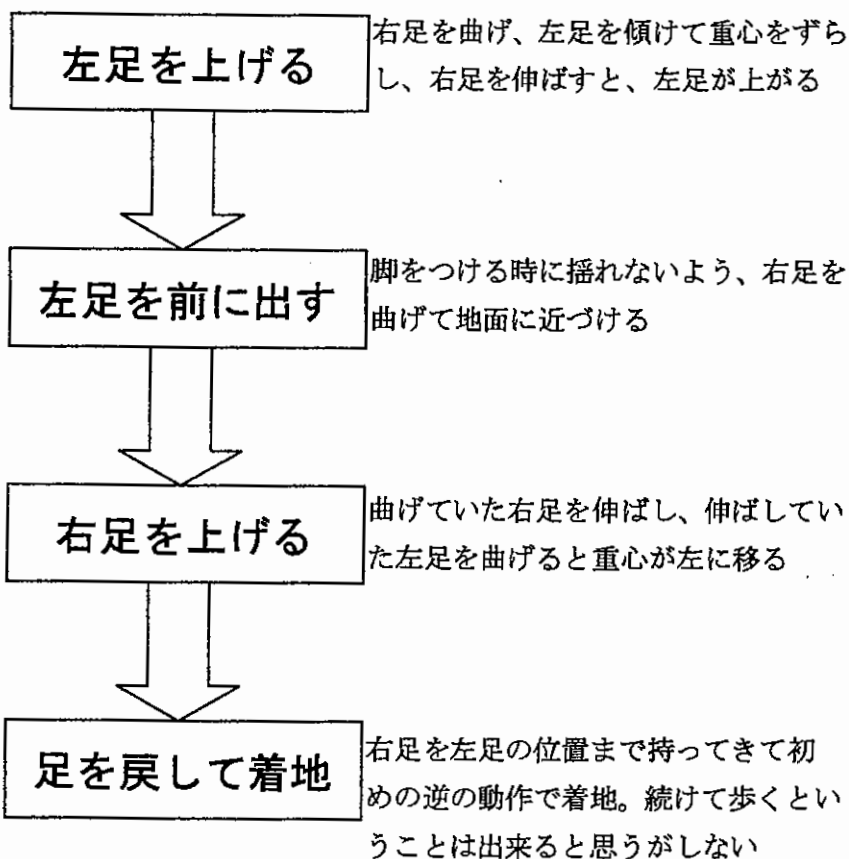
○がモーターの位置

細部説明

・二足プログラムについて

ロボットの心臓であるプログラムはPICというものを書いていますが、プログラム自体はそんな難しくないです。サーボモーターは角度を入れればその角度になってくれるものだと思ってもらって構いません。僕がしたのはそれぞれのサーボに角度を入れて歩かせただけなのですが、その調整が面倒なのです。特に二足歩行ロボッ

トっていうのは重心がぶれやすく倒れやすいのです。去年のも文化祭前は動いていたのですが当日はうまく動きませんでした。今年の二足は頭に電池っていう重いものを載せているせいで重心がぶれやすく、動きが気まぐれなので trial and error を繰り返して作っていくのが去年以上に大変で…。でもやはり計算より努力です。とはいってもなにも回路図集に載せないわけにはいかないので、下には前進のフローチャートを載せておきます。



・電池駆動について

電池は二種類積んでいます。どこにでも売っていきそうな9Vの四角い電池と Panasonic の eneloop×5です。充電電池です。これでロボットを動かす

ことができます。

1.なぜ、二種類あるのか？

なぜなら、モーターが電気を大量に使うからです。たとえ、電気が足りるとしても瞬間的にモーターに電気を持ってかれてロボットの心臓であるPICに流れる電気が足りなくなってしまうロボットが止まってしまうのです。コンデンサを使い必要な時に電気を補う方法もありますが、PICに大電流が流れると壊れてしまう恐れがあるのであまりしたくない方法です。

2.なぜ、5本積むのか？

理由は簡単でPICやサーボモーターは5Vで動くので5V以上の電圧がないといけません。enloopの電圧は1.2Vなので5本直列につないで6Vにしているのです。普通のアルカリ電池を使えば1.5Vなので4本で済むのですが。ちなみにPICやモーターに6Vとかをかけても壊れてしまうので「三端子レギュレータ」という部品を使って5Vに下げています。

3.なぜ、enloopなのか？

9V電池やアルカリ電池では出来ない理由というのは、「容量」が足りないからなんです。サーボモーターはとて多くの電気を食うので、大電流向きでない普通の電池を使ったらたぶんすぐに切れてしまうと思います。(試したことはないですが) それに対してenloopは一本で最高4000mAほどは出せるニッケル水素電池という種類の電池なのです！

ニッケル水素電池は充電できるのでコストが安い(それでも充電器代込みで1万円)という利点の他にこう言った大電流の放電に強くモーター10個を力強く回してくれているのです。

・ Weapon

コイルガン、英語ではcoil gunです。なんか物騒な制作物ですね。ここからはコイルガン担当の関根氏でお送りしていきます。

コイルガンを一言で表すならば、~~ハイ~~武器です。

1.コイルガンとは

コイルガンは簡単に言うとコイル(=電磁石)の作り出す磁力を利用して鉄釘を引っ張りその勢いで鉄釘を飛ばす装置です。それっぽく言うとコイルを瞬間的に電磁石にすることで鉄釘を加速させつつ減速はさせないことにより慣性の法則に従った鉄釘を高速で飛ばすという装置です。なんか難しそうですが一番難しいのは仕組みではなく昇圧器の部分です。

2.パーツと仕組み

構成は制御部と昇圧回路、コイル・砲身の三つに分けられます。制御部からの信号を受けて昇圧回路が動作しコイルに電流を流す仕組みになっています。昇圧回路とは電圧をロボット共通の5Vからコイルガンに必要な高電圧である300V台に昇圧する装置ですが、ここが一番難しいです。

・無線通信

ここからは再びリモコン等の通信担当四柳がお送りしていきます。

1.無線通信について

無線通信という言葉は初めて聞く人にとってはやや仰々しいものですが、わかりやすく言ってしまうと携帯電話やスマートフォン、テレビのリモコンなどが情報をやり取りするときに使っているのが無線通信そのもので、電波などを使うことによって銅線等を使わずに(=無線)通信しようというものの事です。

仕組みとしてはリモコンについているLEDをマイコン制御によって(詳しくは後述します)点滅させ、その点滅のパターンをロボット本体についている光に反応するセンサーで感知することで通信を成り立たせています。

2.可視光通信について

無線通信の中でも可視光線と言われるものを使って通信を行

うものです。この可視光線とは漢字で書くとその名の通りで目に見える光の事で、現在ある照明機器を少しいじるだけで通信設備として併用できるようになる等メリットが多いため研究がすすめられています。外乱光(=通信には関係ない光)によるノイズ対策が必要です。

この当初の予定ではこの可視光通信を使って無線通信しようとしていたのですが、技術的・時間的制約により現在は赤外線(テレビリモコンやガラケーで使われている)になっています。

3. 主な部品

PIC32MX440F512H・・・マイコンの一種。送信側のLEDの点滅や受信側でのパターンの読み取り、そのほかロボット本体内の他のマイコンの制御をしています。マイコンとはマイクロコントローラーの略で身近な所では家電製品の制御によく使われています。自身でプログラムを書き込むことができもともと備わっている機能と合わせて様々な動作を可能にする優れたものです。

LED・・・最近ではLED照明の普及もあって多くの人知っている発光素子。当初の予定では可視光帯のLEDの予定でしたが現在は赤外線帯です。

光センサー・・・光に反応する素子の事です。反応には抵抗値の変化や電流の変化などがありはやりの太陽光発電も電流の変化という意味で光センサーといえます。今回使用したのは光を検知すると電圧が変化するタイプのパッケージ品です。

4. 受信側動作

ロボット本体で信号を受信したPICはそのまま信号を伝えることでロボットの動作全般に対する制御も行っています。基本的には受信した信号をそのまま垂れ流しているだけなので特筆することもなかったのですがその気になれば自爆リセット信号流すことも出来ました。(今更ながらに何故ロボットにロマンの塊自爆スイッチを作らなかったのか後悔)

5. 無線通信詳細説明

ものすごく難しそうで難しくない少し難しい説明。

今回作ったリモコンはシステム設計上 16 個の命令を送信できるようになっていますが、ロボットの動作にはそれだけの動きはないのでスイッチはついていません。また、送信対象を識別するためのコードも 4 種類ありますがロボットは一台だけなのでプログラムで固定しています。

ここまででもしかすると気付いた方もいらっしゃるかもしれませんが、実は今回通信プロトコルとしてはテレビリモコンのものを参考にしています。なので符号誤りの確認・訂正も二連送照合と言って同じ通信を二度行い一致したら実行するという仕組みをそのまま使用しています。

ですが、通信に使っている周波数はおそらく異なっているのでは、テレビに向けても反応しない・・・・・・・・・・と思います。

最後に

感想その一：僕は売り物担当で全然本制作に割く時間がありませんでした!!とか言い訳ですね(笑) 正直教えてくれる先輩がいませんでした。

デジタルと違って調べづらく、ノイズとか予期せぬ問題が発生しやすいアナログは難しいですね。ただ、コイルガンに挑戦したおかげで製作力がついた気がします。アナログの知識も結構増えましたし。なんか共同制作者さんにかなり迷惑かけたので申し訳ないです。

ただ、技術を生むにはたくさんの犠牲が必要だと思いました。By 関根

感想その二：はい。二足歩行ロボットをつくったのは 2 年目となります。別に思い入れがあるって訳でもないのですが、去年は膝なし二足を作ってみたのですが、今年の目標は「コードレス」な二足でした。そのためにはコントローラの無線化と電池駆動。その二つのことを新たに加えねばならなかったのです。僕は歩行プログラム(一部は貰ったもの)、本体の設計(実際は去年の改良である)、電池駆動(初

めは関根君がやってくれると思っていた)の三つをやりました。現状(4/27)ではまだ動いてないのですが、文化祭までには満足いくものになっているかと思います。今年は一人ではなかったのが去年より楽しく製作出来ました。(あとは動いてくれれば…)共同制作の四柳さんと関根君にはもちろん、物無の全員に感謝しております。ありがとうございます。By 渡辺

感想その三：振り返れば今年も決していいとこまで行けた気はしないものの、今回は初めて後輩と共同制作ということでいろいろと楽しかったと思いますね。その分ほかの二人を振り回してしまった気もしますが本当に二人には感謝ですね。あと名前のYとかγとかは三人が円陣組んで手を重ねているイメージだったりします。まあ細かい話はまたあとでBy 四柳

・
・
・

四柳「コイルガンとは？」

関根「殺傷兵k

渡辺「この製作物はあくまで展示用の製作物であって、戦闘用の兵器では
ございません。ご了承ください」

BTMY

H1 山本 涼太郎



ストーリー

僕は F1 ドライバー。今モナコの街道を走っている。前を走るのは HANDA の車だ 1900 年代に黄金時代を築いたが今は F1 に最近復帰してきた新米チームだ。モナコ GP は後 3 コーナーしかない。僕が勝つには外角からドリフトして抜き去るしかない。ぶおおおん エンジン音がうなる。僕はブレーキを踏まずコーナーに突っ込んだ。

説明

はい F1 はドリフトしないですね。僕が作っているロボットは車です。でもただの車ではありません。なんとドリフトをするんです。みんなお馴染み頭文字 D でも登場人物はみんなドリフトを

していましたね。ドリフトとは後ろタイヤを滑らして曲がることです。たぶんみんな知っていますね。ドリフトはすごいスピードが出ないと成功しません。今までのぶつ無のギアボックスはギア比では大きすぎてそんなスピードが出ません。ですのでギア比の少ないのを使うことにしました。(ギア比とは上げるほどトルクが上がり速度が落ちます。)ドリフトは早くないとできないんです。

プログラム

特にこってはいないです。モーターは去年の食べ砲台と同じく回しっぱなしです。サーボモーターは角度を指定できます。サーボは旋回にしか使いません

感想

ドリフトはロマンです。毎年製作をしなかったことを後悔していますが、今年はしません。ドリフトは難しいですね、ドリフトラジコンはものすごいモーターを使っていますね。まだつくり途中ですが、一人製作は自分と向かい合えたと思います。

くあこぱた〜

制作 H-2-3 佐藤 惇
協力 物無の皆様
ネットの優秀なハッカー各位

・概要

実は去年「スマホを自分で作ってやろう!」みたいなことをやっていたのですが、そういうのは**お金が腐るほどたくさんある企業がやるものであって**、僕達のような貧乏人が気安く手を出すものじゃないというのがよくわかりました。血を吐くかと思いました。割と冗談ではなく。

で、じゃあどうしようかするのって話なんですけど、まあ当然

「楽しんで、すごいもん作りたいよね」

って話になるわけです。これぞ人類発展の原点です

で、そういう界隈で最近流行りの「ラピッドプロトタイピング」ってのがああるんです。これは

ラピッド == 早く
プロトタイピング == 試作

ってな感じで、「できるだけ簡単に、早く、なんか作っちゃおう」って話です。まあこれ自体は3Dプリンターなんかの技術についているのですが、これと似たような感じでソフトとか基板とか作る方法ってのがああるんです。

今回の制作はそういうのを駆使して賢くいろいろ作ってみる計画です。

(他の僕が作ってる制作物も基本的にはこのコンセプトです。)

・ハード

さて、まずはハード部分。

基本的にはヘリコプターですので、プロペラで上昇するわけですが、このヘリコプターにはそれが4つついています(クアッドコプターと呼びます)。

で、それぞれにモーターをつけて、それぞれの回転速度を制御することで前後上下左右に加えて回転運動を可能にしています。

ホントはここにハードの外観写真を載せたかったのですが、自宅で制作していた一号機の外装がふすまに突っ込んでしめやかに爆発四散してしまったので載せられないのです。暴走コワイ!

二号機を制作するかと思いきや、その他の制作がたくさん入ってきてしまったので、なかなか進みません。ラピッドが聴いて笑わせます。

さて、そんなわけなのでここでは内部構造の説明を。

① Arduino

ヘリの制御には「Arduino」ってのを使います。中身は物無員大好きPICと似たようなICなのですが、その周りに電源とかLEDなんかがくっついていて、色んな機能が簡単に使えるというものです。ロボット自作の世界標準と言っても過言ではありません。去年も書きましたがなんで物無員はPICがこんなに好きなのか謎

たとえば、LEDをチカチカさせるプログラム一つ書いてみても、

PIC	Arduino
<pre>#include <pic> #include <arch> #define _XTAL_FREQ 4000000 __CONFIG(UNPROTECT & MCLRDIS & BOREN & PWRTEN & WDTPDIS & INTIO); void main() { CMCON = 0x07; ANSEL = 0b00000000; TRISIO = 0b00000000; GPIO = 0b00000000; while(1) { GPIO2 = 1;</pre>	<pre>void setup() { pinMode(13, OUTPUT); } void loop() { digitalWrite(13, HIGH); delay(300); digitalWrite(13, LOW);</pre>

<pre> delay_ms(500); GPIO2 = 0; delay_ms(500); </pre>	<pre> delay(300); </pre>
---	--------------------------

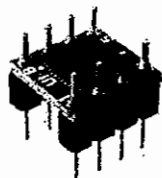
命令も充実しているので、PCでのプログラミングができる人なら違和感なく取り組めるのではないのでしょうか

これだけ差があります。これぐらいなら大差ないように見えますが、この調子でプログラミングを進めていくと、ロボットなどを作る際には大きな差になります。実際スゴイ

②加速度センサ

それから次に重要なのが加速度センサ。

これは物体の速度の変化を感知するもので、これをロボットに取り付けておくとその姿勢の変化や速度を読み取ることができます。で、重力も常に地面に向けて働く加速度なので、それを読み取ればロボットがどちらを向いているかも読み取ることができることとなりますね。すごく便利なものなんです。これローパスフィルタ(基板内蔵だけでは感度が高すぎる)やベクトル演算しだいで

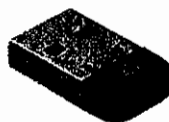


はそれなりの精度が出せます。

ついでに言うと、チップと基板(←みたいなもので、基板のサイズは一円玉以下)も軽いので、ヘリコプターに積んでも大きな負担にならないのもいいところです

ヘリコプターは機体を傾けて飛ぶものですので、これをいかに正確に読み取れるかは大変に重要な要素になります。

③ Bluetooth



それから通信部分です。今回はPCに手を認識させて、それと通信して制御することになるので(割と)高速な通信を無線で安定して行えることが重要です。

自ら無線通信をBluetoothを使っています。

スマートフォンの通信でも使っていますし、これで動いているキーボードやマウスも少なくありません。100mまでしか使えませんが、合法(※ここ重要)に無線通信を使うことができます。

で、使う部品も市販品。←みたいなPC用のBluetooth Dongleってのを使います。

これを交換基板なんかをかまして前述のArduinoにつなぐわけです。するとPC、スマホ、はたまたロボット同士など、なんでも好きなものと無線で通信できるようになります。

これと加速度センサーを使うことでWiiリモコンのように動くわけです。(ちなみにWiiリモコンもBluetoothで動いています。)

後輩たちにもBluetoothを使っている制作があるようですが、どうも必要以上に高いDongleを使うとかあって動かないことがあるようです。ご自分で制作される場合は家電量販店などで売っているDongleのうち最も安いもの(ただしSPPというプロファイルに対応してるもの)でも十分動くかと思えます。

これらを組み合わせてゴニョゴニョすると、本体が出来上がる、はずです。

④その他

フレームはタミヤの汎用プラ角材を使って組んでいました。これにアキバで見つけた名前も知らないモーターをTA7291Pで駆動し、ARdrone2.0というヘリコプターおもちゃのプロペラで推力を稼ぎます。

とにかくBluetoothのおかげで通信ノイズなどに気を使う必要が少ないのは助かります。

・ソフト

ではソフトの話。ヘリを制御する側です。

どちらかという僕はこっち専門なのでこっちのほうがメインですかね。こっこの技術は僕の他の制作物でも使っているのでそちらもお読みいただけると嬉しいです。



① Processing

こっちで使っている「ラピッドプロトタイピング」的秘術の名前はProcessing。

最近はやりのプログラミング言語で、電子アート用言語です。これはJavaというプログラミング言語をアート用に使いやすくしたものです。→がロゴ。

ちなみに、後述のロケットシュミレータや IVRC 作品もこの言語でプログラミングしています。無料なので、気軽に使うことができます。

この言語、とても簡単に使うこともできるのですが、Java ベースの言語なので、最近風の文法を用いて本格的なプログラミングにも活用できますし、豊富な Java の資産を活用できます。また、Windows 以外でも動くので Linuxer(かつ Java プログラマー)の僕にとっては大変ありがたいものです。

で、こんな「楽だ!便利だ!」って言ってもケーブルテレビなんかで一日中やってるテレビ通販程度の説得力しかないので、実例を挙げてみたいと思います。

Java	Processing
<pre>import javax.swing.JFrame; import java.awt.Color; import java.awt.Graphics; import java.awt.Graphics2D; import java.awt.geom.Ellipse2D; public class Sample extends JFrame { private static final long serialVersionUID = 1L; public Sample() { setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE); } public void paint(Graphics g){ super.paint(g); Graphics2D g2 = (Graphics2D)g; g2.setColor(Color.GREEN); Ellipse2D ellipse = new Ellipse2D.Double(50, 50, 100, 100); g2.draw(ellipse); } public static void main(String[] args) { JFrame f = new Sample(); f.setTitle("Sample"); f.setSize(200,200); f.setBackground(Color.WHITE); f.setVisible(true); } }</pre>	<pre>size(200, 200); stroke(0, 255, 0); background(255); ellipse(100,100,100,100);</pre>

なんと、これでどちらも白い四角に緑の丸を描くプログラムなんですね。左で書いてるのが馬鹿らしくなってきます。ほんとにもちろん Java でしかできないこともあります、そういうのは Java ベースってのを活かしてガリガリとオーバーライドしてやれば全く問題なし。Processing の特徴は、不必要なところは最大限簡潔に済ませられ、かつ、必要なところはどこまでも性能を追求できるというところにあります。つまり楽に効率的に小規模な制作をするためには理想の環境なのです。僕が作った他の制作物にも Processing を用いたのは、こういった特徴があるからなんですね。

で、ここでまで読んで気づいた方もいらっしゃるかと思いますが、こんな感じの表が前にも出てきましたね?

そう、Arduino。

実は Arduino は Processing の資産を活用して制作されていますので、どちらもある意味似た物同士なのです。

(Arduino は C、Processing は Java ですが)

そんなわけで Arduino と Processing の組み合わせは非常に都合がよかったです。

どちらも相互運用を考慮に入れているので、動かすのがとても簡単。2つの間の Bluetooth とデータ処理をゴニョゴニョと書きゃえ動くのです。

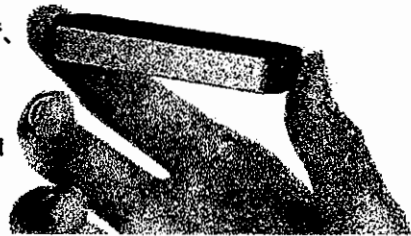
② Leap Motion

これはハードウェアの名前なのですが、ヘリコプター本体部分というよりも制御側なので、こちらに。PCなどに触れることなく直感的操作を行うためのデバイスで、赤外線投影と2台のカメラを内蔵することで、デバイスの上にある指と手の位置や向きをかなりの高精度(公称0.01mm)に認識してくれます。

これを使えばアイアンオンのアレも作れますね。というかドコモが似たようなの作ってました。夢があります
こういったデバイスの中では珍しく、本体が小さいのも特徴でUSBメモリぐらいの感覚で持ち運べるので外でUltraBookとともに使うことも可能です。

個人で使うにはその価格も魅力的です。(7000円強。)

最近はこれでプレイできるゲームなんかもあったりするので、持っているだけでも結構楽しいです。



実はこの機能で Web ページを見たりすることも可能なのですが、それについてはこれを使ったゲームの方で紹介したいと思います。

で、今回は LeapMotion の直感的な操作をヘリコプターの操縦に活かせるようにしようという話です。

LeapMotion は←のように手の傾きなども認識することができるので、これを使えばヘリにさせたい動きを人間が手で示すだけで、ヘリをそのとおりに飛ばすことも可能です。

ロボットアームなんかを制御するのも面白いかも

例えば今回の場合は

手を上下に	上昇下降	手を左右に傾ける	左右に回転(ロール)
手を前後に傾ける	前進後進(ピッチ)	手を右左に回転させる	回頭させる(ヨー)

と言った形で、直感的に操作できるようにしました。まるで飛行機ごっこをしているみたいですね

PC はこれらのデータを角度で受取り、ヘリの目標(理想の)姿勢を取得します。

そして、今の姿勢を加速度センサで読み取って、その差を修正するようにプロペラの回転を調節するってのを繰り返してやっていけば常に操縦者の思ったとおりにヘリコプターが飛んでくれることになります。

少し技術的な話をすると、

LeapMotion を Processing で使おうと思うと LeapMotionP5 というライブラリを使うのですが、これのデータの扱いが微妙に面倒くさい(手の形に合わせて構造化されていたりするので、しっかりと公式リファレンスを読んでおかないととても扱いづらいです。それから、LeapMotion では手の検出処理をコンピューターに丸投げしている(Kinect では内部に専用のプロセッサが入っている)ので、それなりの負荷になります。LeapMotion の処理とは別に重い処理がある場合はスレッド分けなどをしっかり行ったほうがいいのかも。結構カクつきます

感想

LeapMotion は IVRC の勉強会に参加した時に OculusVR とともに触る機会があり、その斬新さに惹かれて、思わず数日後には購入していました。実生活で使う機会があると言われるととてもそんなことはないのですが、このワクワク感だけでも個人的には買った意味はあったかなと思います。

プログラミングのあまりの楽しさにハードウェア制作が後回しになってしまったのは後悔しています。どう考えてもハードウェアが完成してからソフトを作るべきでした。

いろんなことにチャレンジできて、去年とはまた違ったベクトルで様々なことを経験出来たと思います。

Mii ～ミイ～

H1 市川 詠亮

M3 福島 弘也



1、ストーリー

その日、カブトムシ達は思い出した。奴らに支配されていた恐怖を…虫籠の中に囚われていた屈辱を…

100年前、カブト達は50cmを超える壁を3重に作り、巨大カブトのいない安全な領域を確保することに成功した。

しかし5年前、その壁を超える超巨大カブトが出現。壁は破壊され、カブト達は先端の壁を放棄し、現在のカブトの活動領域は狭まっている。

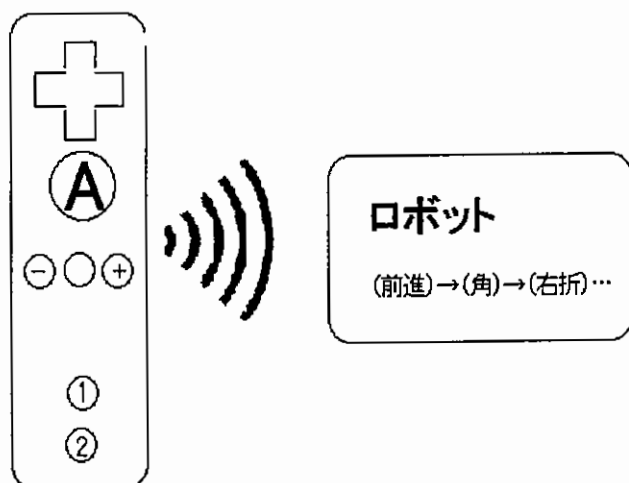
そして、また壁が破壊されてしまった。このままではカブトムシは滅んでしまう！

あなたはミカサカブトとなって、人類を、そして最愛のエレンカブトを守るために、立ちはだかる敵カブトを倒すのだ！

2、概要

1 ページ目の茶番はさておき、このロボットは対戦型ロボットです。Wiiリモコンで操作することができます。乾電池駆動なので、ラジコン型対戦ロボットと言っても良いかもしれません。

コードレスですし、充電式乾電池を使っているので、かなり省エネです。



Wiiリモコンで前後左右に移動でき、同じく角を上下に動かせるで、相手をステージから押し出すか、引っくり返せば勝利です。

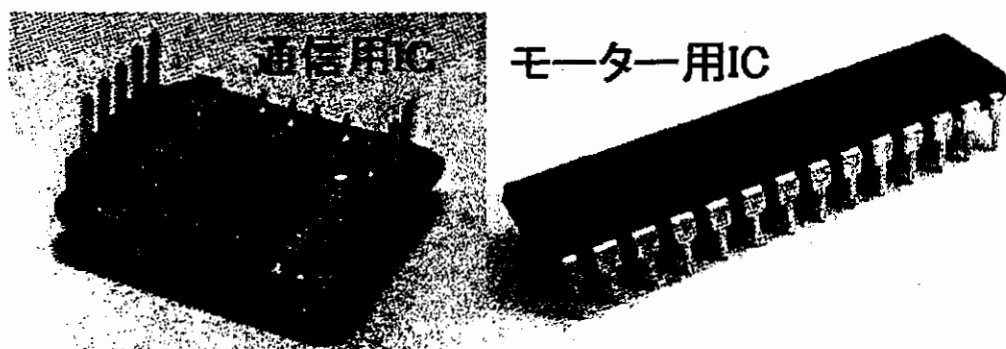
3、仕組み 担当:市川

wiiリモコンでは、Bluetoothと呼ばれる通信方式が採用されています。この方式にはいくつか種類がありますが、10m程の距離まで電波が届きます。

wiiリモコンからBluetoothで送られてくる信号を読み取り、ロボットは動いています。

ロボット本体には、通信をするICと、モーター等を動かすICが入っています。

やはりwiiリモコンとの通信は難しく、かなりの時間がかかりました。



4、本体、モーター 担当:福島

ロボットの本体には2種類のモーターを使用しています。角用のモーターと、前進用のモーターです。

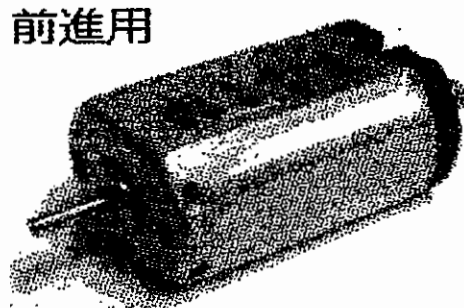
角用のモーターは、ラジコン等でも使われている、サーボモーターと呼ばれるもので、前進用のモーターは誰もが知っている、普通のモーターを使っています。

本体自体は、アクリルの板と金属板で構成されています。
初心者でも加工のしやすい材料をえらんでいます。

角用



前進用



4、感想

市川

楽しかったです(小並感)配信されているサンプルプログラムに頼ろうと思ったのが良くなかったですね。結局あれこれ調べて、プログラムを書き換えて、それでようやく通信部分が完成しました。

福島

楽しかったですがやっぱりまだボーっとしてる事が多かったです。それに平日よりも土曜日の方がぐだぐだしてしまいました。また春休みもあまり行けなかったのもありました。

あと、設計図書くのが遅かったりわからない時があったりしたのもそうですね。

来年も頑張ります。

ぐるぐるぐるりん

製作:H2-6 黒田健太

0.ストーリー

とある国の、陸軍開発局の一室で、その研究はひそかに行われていた。

W 田教授「遂に…遂に完成した…。」

I 島研究員「教授！一体何が出来たというのです!？」

W 田教授「これがあれば敵国に多くの損害を与えることが出来るやも知れん…。」

I 島研究員「教授！だから一体何が出来たというのです!？」

W 田教授「端から見ればただの筒…しかしこいつは無線で動かすことが可能…。これを使って敵国の拠点を叩けば…敵国は総崩れだろう…。」

I 島研究員「教授！拠点にこれが入ったとしても攻撃する手段が無いじゃないですか！それどころかほぼ何も出来ませんよ!」

W 田教授「ふふふ…ふははは…。」

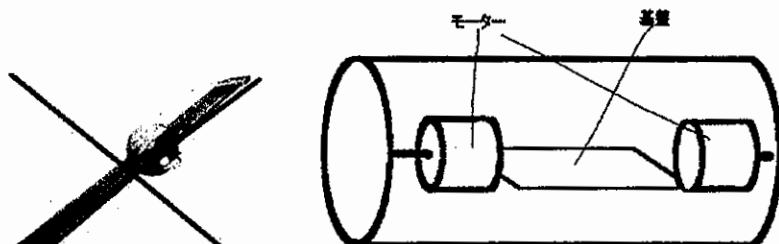
I 島研究員「教授！聞いてるんですか教授！実は聞こえてたりするんじゃないですか!？ちょっとくらい返事してくださいよ教授！返事しないとそれぶっ壊しますよ教授！ねえってば!…」

↑なんか…すみません。

1.概要

ぶっっちゃけたただのラジコン無線により操作できる筒状の回転マシン。回転するだけかって？回転することに意味があるんです(キッ だってかっかわいいじゃん!)

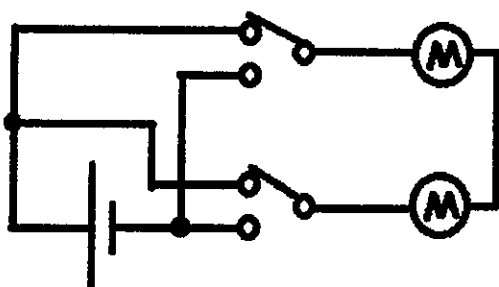
見た目はただ円柱状の物体ですが中身はこんな感じ↓です(調べてみたら案外くるくる(ryと形が違うことが判明)



図はだいぶ割愛してますが、基盤のところに無線モジュールとリレー(電気を流すとつながるスイッチ)があり、それによって電源から電気を供給してモーター回す…といったところ
です。

私はこの製作物において一切プログラミングをしていません。私がそんなにプログラミン
グに詳しくないのはありますが素人でプログラミングなどしなくとも既存のものを使って
簡単にラジコン的なものを作れることを知ってほしかったのです。手抜きじゃないです。
なので今回使用した部品とその解説を後にするとします。

3.回路図



ペイントで書けちゃいました。

電源とモーターの間のリレーのどちらが切り替わるかによってモーターの回転方向が変
わります。リレーに電流を流すための無線モジュールは紛らわしいので図から割愛。

4.使用した部品

- ・モーター…マブチモーターRC-260
- ・無線モジュール…受信側は RX315-2CH-AI、送信側は S-RC2260AR4-2CH106
両方とも秋葉原の aitendo で購入。
- ・遊星ギヤボックス…タミヤの遊星ギヤボックスセット。ギヤ比は 16:1 だったはず。
- ・本体…アクリルパイプ。東急ハンズで購入。

これぐらいのものです。全体で 1000 円かかってない気がします。このページをご覧のあな
たもお手軽にラジコンを作ってみてはいかがですか？

5.製作後記

正確にはまだ終わってないんですけどね^^; 今回の製作物は電子工作の手軽さをアピール
するためとはいえ、ちょっと手を抜きすぎたかも知れません。言い訳じゃありませんよ
最後の製作物でしたが個人的に後悔はありません。少ししょぼい感じもしますがそれもま
たいい、ということで。後々のページで遺言として長々と語るかも知れないのでこの辺で。
理科系四部活展の音ゲー管体のほうもよろしくね☆

DEATH STER

～FINAL DISASTER

製作：H1横尾、M3小栗、M3和田

協力：物無の皆様方

ストーリー

時は20XX年。とある寂れた町に倒産間近かのロボット工場があった。

そこでどうやってこの危機を乗り越えるか会議が行われた

社長「もうこの会社も終わりだ」

社員「あきらめるのはまだ早いですよ」

社長「いったいどうすればいいというんだ」

社員「今まで誰も作ったことのないようなものを作りましょう」

社長「そんなものが俺たちに作れるわけがない」

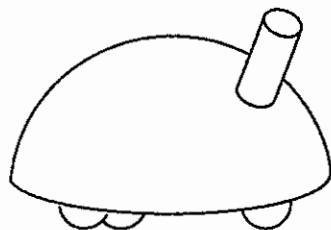
社員「私たちがならきっと作れます」

そうして何でもできるロボット death ster が作られたのであった。

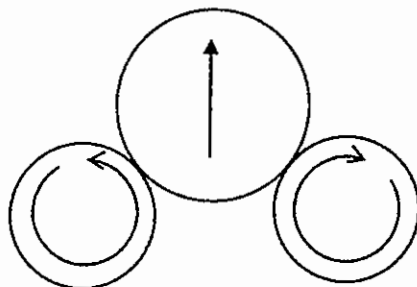
そしてもうこれ以上何も起こらないように最後の災害と名付けた

概要

まず最初に、この製作物を簡単に説明するといろいろな部品を付け替えることでほとんど何でもやることのできる万能ロボットです。とは言っても実際にできることはほんの少しですが・・・

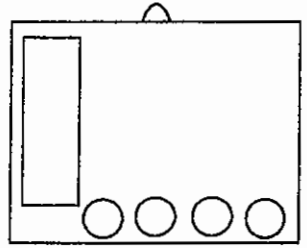


右上にある絵は death ster に球を飛ばすことができるもの（筒みたいなもの）をつけたものです。この球を飛ばすものは取り外しすることができ、つけることで death ster は球を飛ばすことができるようになります。どうやって球を飛ばしているのかというと、二つのモーターを逆回転することで球をはじいて飛ばしています。（右図は、左右二つがモーターでまんなが球です）筒の部分は固定されているので狙いをつけたいときは本体を動かして狙いをつけます。



次に、death ster のコントローラについて説明します。

右図はコントローラの図で、コントローラの右下にある4つのスイッチを押すことでdeath sterを前後左右に動かすことができます。左から順番に前、後、左、右に移動するときにおすすイッチになっている。コントローラの上についているLEDは、赤外線を出することができるLEDで、スイッチを押すことでLEDから赤外線がでてdeath sterの上についている赤外線センサーが反応してdeath sterが動くという仕組みになっています。



これでdeath sterの説明はほとんど終わりましたが、はじめに書いたようにdeath sterの可能性は無限大なので将来的にはもっとすごいものができるかもしれません。

プログラム

death sterにおいてプログラムは主にコントローラと本体の通信に使われています。どのようなプログラムなのかというと、プログラムはコントローラ用(送信)と本体用(受信)に分かれていて、赤外線LEDをどのぐらいの時間光らせるかでどのボタンが押されたのかを区別しています。実際にどのようなプログラムを書いたのかを知りたい方もいると思いますのでプログラムを載せておきます

送信

```
#define XTAL_FREQ 8000
#define spi_h

void ledon1(void);
void ledon2(void);
void ledon3(void);
void ledon4(void);

void main(void){
  OSCCAL=8000;
  int i,s,a1,a2;
  while(1){
    a=0;a1=0;a2=0;
    if(GP4==0)
      __delay_us(400);
    else
      break;
    if(GP4==0)
      __delay_us(800);
    else
      break;
    if(GP4==1)
      __delay_us(800);
    else
      break;
    if(GP4==0)
      __delay_us(800);
    else
      break;
    if(GP4==0){
      a=2;
      __delay_us(800);
    }
    if(GP4==0){
      a1=1;
      __delay_us(800);
    }
    if(GP4==0)
      __delay_us(800);
    else
      break;
  }

  if(GP4==0)
    __delay_us(800);
  else
    break;
}

if(GP4==0)
  __delay_us(800);
else
  break;
if(GP4==1)
  __delay_us(800);
else
  break;
if(GP4==0)
  __delay_us(800);
else
  break;
if(GP4==0)
  __delay_us(800);
else
  break;
if(GP4==0)
  __delay_us(800);
else
  break;

if(GP4==0)
  a=a2+a1;
else
  break;
switch(a){
  case 1:
    for(i=0;i<(a1+i){
      ledon1();
      __delay_us(80);
    }
    break;
  case 2:
    for(i=0;i<(a1+i){
      ledon2();
      __delay_us(80);
    }
    break;
  case 3:
    for(i=0;i<(a1+i){
      ledon3();
      __delay_us(80);
    }
    break;
  case 4:
    for(i=0;i<(a1+i){
      ledon4();
      __delay_us(80);
    }
    default:
      __delay_us(80);
      break;
  }
}
}

void ledon1()
GP0=1;
GP1=0;
GP2=0;
GP8=0;
}
void ledon2()
GP0=0;
GP1=1;
GP8=0;
GP8=0;
}
void ledon3()
GP0=0;
GP1=0;
GP2=1;
GP8=0;
}
void ledon4()
GP0=0;
GP1=0;
GP2=0;
GP8=1;
}
}
```

受信

```
#include <stdio.h>
#include "pin_20776.h"
#define _STAL_PRESQ 0x000
//Name INTRG_IO, FWT, NOPROTECT
//base delay/clock=8000000

__CONFIG(BORDDIS & UNPROTECT & PWRTEN & WDTDIS & MCLRBN & INTIO);

void out(void){//外部線 Low 出力
void outb(void){//外部線 HI 出力
void startb(void){//スタートビット
void stopb(void){//ストップビット
//GP0 スイッチ 1
//GP1 スイッチ 2
//GP2 スイッチ 3
//CP8 外部線 GPS
static int kalamu = 0; //LED 点滅用

void main(void){
while(1){
GPIO=1;
__delay_us(1000);
GPIO=0;
__delay_us(1000);
}
}

main(){
int a=0; //送信データ
int b=1; //受信監視用
int c=0; //繰り返し送受信回数
TRISIO = 0x07;
GP8=0;
while(1){
if(GP0==0){
__delay_us(10);
if(GP0==0){
a = 10;
b = 0;
}
}
if(GP1==0){
__delay_us(10);
if(GP1==0){
a = 20;
b = 0;
}
}
if(GP2==0){
__delay_us(10);
if(GP2==0){
a = 30;
b = 0;
}
}
}
if(a > b){
switch(a){
case 10:
//スタートビット
startb();
//データ 4 ビット
outb(0);outb(0);outb(0);outb(0);
//ストップビット
stopb();
__delay_us(10);
break;
case 20:
//スタートビット
startb();
//データ 4 ビット
outb(0);outb(0);outb(0);outb(0);
//ストップビット
stopb();
__delay_us(10);
break;
default:
break;
}
b++;
}
}
}

void stopb(){
//ストップビット HI-Low-Hi-Low-Hi
outb(0);outb(0);outb(0);outb(0);
}

void outb(){//外部線 LED 点滅
GP8=1;
__delay_us(800);
}

void outb(){//外部線 LED 点滅
kalamu=20;
while(1){
GP8=1;
__delay_us(10);
GP8=0;
__delay_us(10);
kalamu--;
if(kalamu==0)
break;
}
}

if(GP8==0){上
}
if(GP4==0){下
}
if(GP1==0){右
}
if(GP3==0){左
}

void main() {
ANSEL = 0; // Configure AN pins as digital
GMCON = 7; // Turn off the comparators
TRISIO = 0; // configure pins of GPIO as output
do {
GPIO_P0 = 1; // beginning of a repeat loop
Delay_us(1000); // turn ON diodes on GPIO
GPIO_P0 = 0; // wait 1 second
Delay_us(1000); // turn OFF diodes on GPIO
} while(1); // wait 1 second
} // endless loop (as this condition is
always satisfied)
```

いつか作りたい部品

文化祭に出せるかはわからないがポテトサラダを作れるものを作りたいと思っています。ポテトサラダを作るといってもじゃがいもを潰すところをやってもらおうというものです。いまのところ考えているところは筒の中にじゃがいもをいれて網に押し付けることでじゃがいもをポテトサラダみたいにしようという計画です。じゃがいもは結構固いので実際に潰すことができるのかが問題だと思っていますがそれ以外は何とかできそうななので実際に作れるかは、時間によります。」

感想

横尾 まずは、あまり製作に参加できず、申し訳ございません。二人には迷惑をおかけしました。

デススターの完成しないという呪いを打破したかったのですが、現段階（五月）であともう少しです。頑張ります。

基本やったことのあるDCモーターを使いました。自分なりにはすいすいいったと思います。二人には感謝です。

小栗 ぼくも、横尾さんと日程が合わず迷惑をおかけしました。すみませんでした。今まで完成しなかったデススターですが何とか形にはなりそうです。無計画なままやったのであとになって問題が出てきたりしましたが、楽しかったです。来年の事は未定ですがpc系に挑戦したいと思っています。

和田 去年やっていた電子工作とは一線を画すロボットを作るということで自分なんかにできるのかと不安に思っていたが、自分に判らないところなどは先輩に聞いたりした結果問題なく作ることができたので、自分にとってはいい経験になったと思った。

三代合体 ガンナー

Crafter:H2 花園

H1 佐伯

M3 川本

～STORY～

2 x x x年、人類は危機に瀕していた！

世界征服を企む闇の科学者ガラム博士が、兵器による進攻を始めたのだ！
その状況下で、三人の若者が立ち上がった。彼らは、己の持つ科学力で、三台のマシンからなる合体ロボを作り上げたのだ！

しかし、ロボの起動を目前とした彼らに、ガラム博士の魔の手が迫る！

「まだこんな反乱分子が残っていたとはな。だがその希望の芽もすぐに断たれる。」そんな危機の中、彼らは、

口論をしていた。

「やっぱ1号は年長者だろ！」「大抵1号は若者なんですよ！」

「とりあえず乗らないと話が進まないじゃないですか…」

そう、彼らは歳が一つずつ離れているため、部活的先輩後輩の意識ができてしまっていたのである。その結果、最も年上と最も年下の間で、どちらがよく目立つ1号に乗るか口論になってしまったのだ。

「…このガラム博士が直々に潰しに来てやったのだ。さあ絶望を見「黙れ」「黙って下さい」「今ちょっと無理です」「(´・ω・`) ショボーン」

「…もうジャンケンとかでいいじゃん」「…ハッ」「…天才ですか!？」

「…さっきからそれ言ってたじゃないですか…」三人の心が一つになり、最終兵器が動き始める！

説明しよう！

これは、物無員の科学力、技術力、そして浪漫が結集した、大型ロボ兵器
なのである！

そのボディには無数の兵器が搭載され、圧倒的戦闘力を誇るのだ！

真面目に説明しよう！

物無で初？の複数変形&合体ロボです。三体の車型メカが自動で集合、合
体して人型二足歩行ロボットになります。1，2号の合体、集合には磁力、
2，3号の合体にはモーターを使ったジョイントを使用しています。



←理想



↑
約50cm

←現実
↓

1号

担当:M3 川本惇之助

合体していない時は普通に走る車です。

一つのモーターで前輪を左右に曲げ、もう一つのモーターで後輪を前後に回転させて走らせています。

結局はただのラジコンカーです。

合体すると、後ろの部分が開いて顔が出てきます。

*合体方法

ホールセンサーという、ホール効果を利用して磁界を検出し、一方向のみの磁力を検出できるセンサーを使って、

2号機の間所を探します。

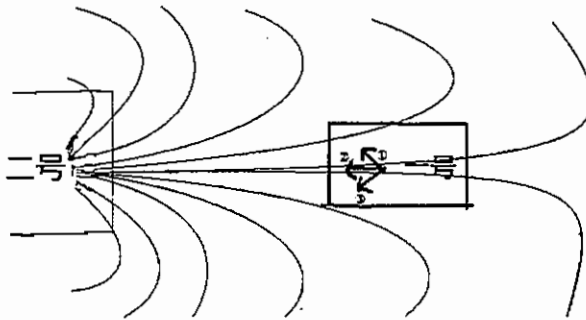
ホール効果とは、電流の流れているものに対し、電流に垂直に磁場をかけると、

電流と磁場の両方に直交する方向に起電力が現れる現象のことです。

左右のホールセンサーが磁力を検出しているときは、中心のホールセンサーと向きを合わせるように曲がり、

中心のホールセンサーが磁力を検出しているときは直進して、2号機から出ている磁力線をたどります。

そして、ある程度近づき磁力の強さが一定以上になったら、2号と合体します



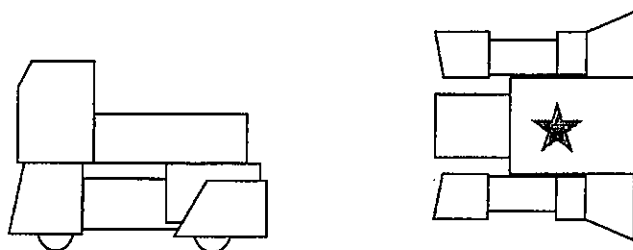
川本の感想！

今年はチカチカ隊長をやっていたので、あまりロボットの方は進んでいません。佐伯さんと花園さんには迷惑をかけ申し訳なく思っています。まだプログラムを書いていないので(5月24日現在)早く書かないとですね。本体はできているので文化祭までには終わると思いますけど。

*****ここから花園に移ります*****

2号

合体時に胴体となる2号は、三体の中では中核的なポジションを担っている機体です。



通常時

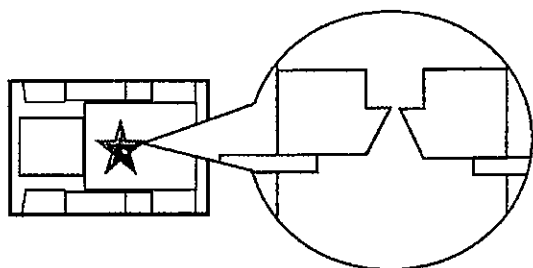
合体準備時

(上から)

*通常時

2号は、普段はトラックのような形状をしたラジコンです。しかし、機動性は1号に劣ります。もともと2、3号は単体での走行を目的としていません。後述の無線規格で操作しています。一応前輪がDC(一般的なモーター)で駆動します。

*合体準備時



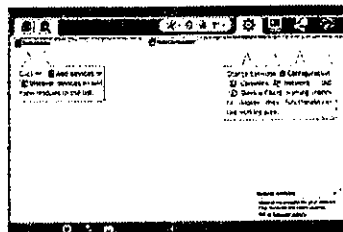
トラックの荷台部分(左図

星印) ↑合体準備の図に内蔵されているサーボモーター(関節的役割)によってトラックの車輪部分が横に開き、腕になります。1、3号への命令(後述)や、腕を振ってバランスをとるなどの役割があります。

*Zig Bee (Xbee)(ジグビー)



↑ Xbee



↑ X-CTU

この制作物に用いられている Digi international 製の無線通信モジュールです。ラジコンなどと違って、相互に通信することや、単体から複数への送信できます。このロボットでは、合体時に互いの状態確認などが必要なので使用することにしました。

一度 X-CTU というソフトで設定を書き込んだ後、PIC とのシリアル通信で動作します。周波数について勉強しなくてもできるので、楽といえば楽です。本来は Xbee Wi-fi という機能もあるそうですが…

~~因みに同じことをするなら bluetoothの方が楽という話もあります。~~

*感想

現状(5/6) : Xbee 難しいです。Xbee が実用化される物無初の制作物となって欲しいですね。まあ使えるので(震え声)。

さて、最後まで迷走し切りました。この制作において、企画、1, 2号の設計、製作をメインに担当しました。他人に比べプログラムの才能がない私は、その分長所(と言えなくもない)金属加工、設計で役目を持つとしたのですが、そのおかげでまた佐伯君が苦勞人ポジションになってしまいましたorz。まあ、

楽しかったんで良しとします。 ←制作はこれに尽きます

未来に向けて。加工設備は買い換えていいかと。あといい加減PICの初期設定、サーボの駆動プログラムとかはみんな大体同じなだから文章、データ化して使い回されるべきだと。みんなそこで悩んでいますが、本来悩むべきは各々の発展の部類なのです。

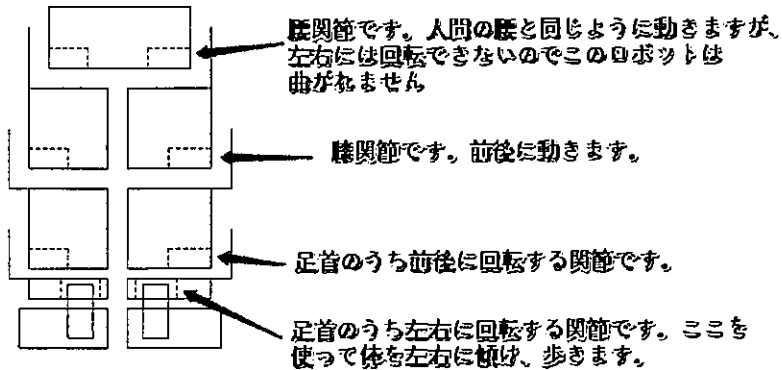
ここまでの、出番を終えとしましょう。次は、苦勞人佐伯君担当の3号機の解説です。

*****ここから佐伯に移ります*****

3号

3号は合体時にロボット全体の足になる部分です。合体する前から足なのはしょうがない

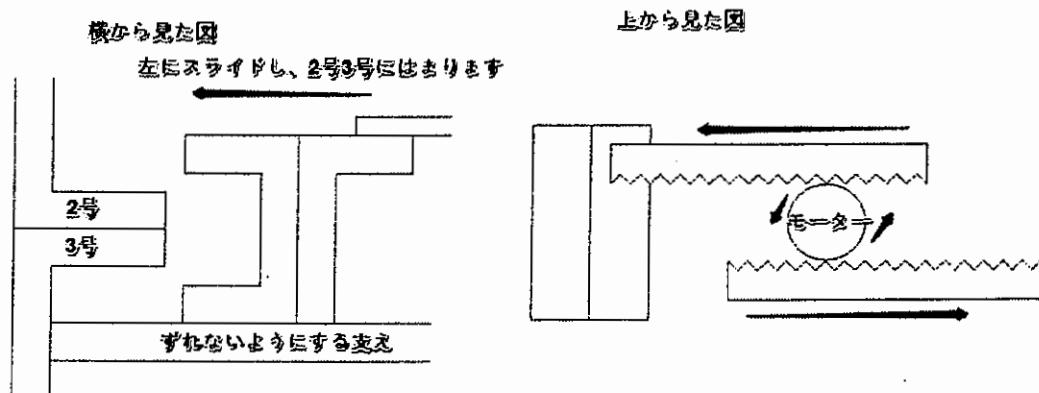
*全体像



当初は均一な太さのバスになる予定だったのですが、スペース、形状などの都合で車化するのは断念しました。

*合体方法

二号機と三号機は合体にモーターを使います。図にあるようにモーターを回転させるとモーターに取り付けられた棒が回転し、コの字材を移動させて二号と三号をではさみこみます。本体外部のL字材の支えもあるので見た目よりがっちりとしていて歩行時に揺れても問題ありません。



*3号の仕組み

Microchip の PIC16F1938 というマイコンを使用しています。9v 電池(四角い電池)から電池を取り出し動いています。

三号には上の図にもあるように両足に4つずつ、計8つのサーボモーターが搭載されており、それらを動かして歩くことができます。

サーボモーターは角度を調整できるモーターです。詳しくは下に書いておきました。

*歩き方

サーボモーターは両足共に、腰と太ももの間に一つ、膝に一つ、足首に二つついています。これらのモーターを使って体を左右に揺らしながら歩きます。歩くときには二号機の腕を振ることで少しバランスをとっていたりもします。

*サーボモーターとは

サーボモーターとはモーターのシャフト(軸)の回転角度を測定できるエンコーダーを備えたモーターのことです。エンコーダーとはさまざまな機器や装置の移動方向や移動量、角度を検出する電子部品の総称です。このロボットに使用しているサーボモーターはエンコーダーに可変抵抗を使用

しているようです。可変抵抗とは付いているつまみを回すと抵抗値が変化し、その結果角度を検出することができるものです。これを使いモーターの角度を測定して任意の角度で止められるようになっています。

通常のモーターは電流を流している間はずっと回り続けますが、サーボモーターは電流の他に信号を流すことで角度を0度から180度の間で自在に変えることができます。

このサーボモーターを動かし歩きます。

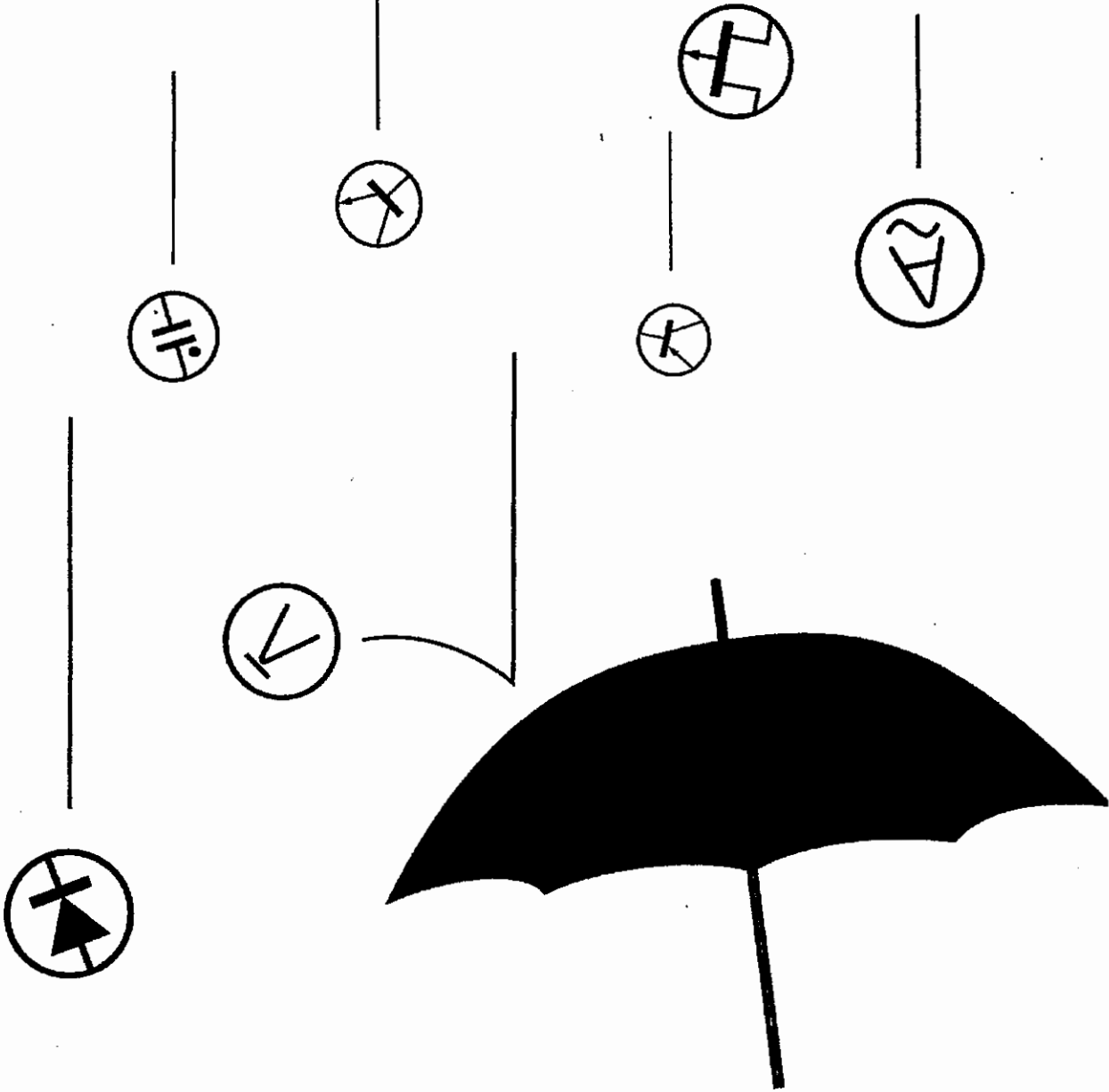
*感想

今年はサーボモーターの制御、A/D変換(この制作物には載っていませんが、可変抵抗でサーボモーターの角度を調節し、その角度を7segに表示するものを作りました。ロボットの調整に使います)、Xbee、金属加工、本体設計などなどいろんなことをやりました。ほとんど一から作り上げたものではないのですが。

ギリギリまで本体の調整しているので5/10でもなかなか歩かせることができません(歩かせる段階にはいっていません)が文化祭までに頑張ってロボットを歩かせます。

今年は二足歩行で歩いてコイルガン撃ってくる兵器があるようで、それとかぶってしまった感がありますね。文化祭では、アピールポイントの”合体”を全面に押し出していきたいと思います。

ゲーム系



Chicken DIVE

H2 佐藤 博

ヘリコプターを作っている過程で加速度センサが完成し、僕がヘリコプターづくりにも飽きていたのもそれが想像以上に面白かったのでひとつぐらいゲームを作ってみようという話になりました。なし崩しでできた制作物なので何から何まで雑ですね。題名の元ネタがわかる人はいるのでしょうか。僕はとんかつにはソース派です

ヘリコプターの方の副産物なので、ヘリコプターの方を先に読まれることをおすすめします。Processingを使って実装しています。(Processingについてはヘリコプターの項をお読みください) 後述しますが、HTML版も制作しています。

ゲームの中身としては→のようになります。地面に設置されている敵の飛ばす砲弾やレーザー、ミサイルをよけながら地面に降下していきます。平坦な場所に着地でクリア、次のレベルに進みます。敵に撃ち落とされるか、地面に墜落するとステージをやり直しになります。

①マップの生成

マップは自然な山型を作るため、一次元パーリンノイズを使用します。

```
PGraphics mapImage = createGraphics(width,height); //PGraphicsをインスタンス化

float noiseSeed((int)random(1000)); //パーリンノイズのSeed値を実行ごとに变化させる
float count = 0;
for (int i = 0; i < width; ++i) {
    count += 0.003;
    map[i] = noise(count)*height*0.8+50; //マップのデータを配列に。
}
mapImage.beginDraw(); //PGraphicsの描画開始
mapImage.background(0);
mapImage.stroke(0,255,0);
mapImage.noFill();
for (int i = 1; i < width; ++i) {
    mapImage.line(i-1,map[i-1],i,map[i]); //配列をもとに描画
}
mapImage.endDraw();
```

ステージ開始前にPGraphicsに描画することでフレームごとの処理を減らせます。実際には合わせて二次元ノイズを扱うことで地形のマッピングを行います。

②敵の配置

レベルに合わせた数の敵を配置します。

後々の管理を考えてEnemyクラスを継承した各種の敵のコンストラクタを使って、Enemy型のArrayListに敵をaddしていきます。オブジェクト指向の強みですね。

```
for (int i = 0; i < 3*level; ++i)
{
    int x = int(random(width));
    enemyList.add(new ShellEnemy(x,map[x]));
}
```

例ではただランダムな場所に配置するようになっています。

こうすることで表示を

```
for (Enemy e:enemyList)
{
    e.update();
}
```

とシンプルにできます。計算速度から言うと望ましくはないですが、大した数があるわけではないので許容範囲ではないでしょうか。これで下準備は完了です。

③プレイヤー操作の演算について

ローカル動作版では自作物理エンジン「GPhy」を用いています。このエンジンは惑星シュミレーションに特化し、物体の引力、斥力その他の基本要素のほか、推力を持ったロケットブースター、均一に分布しない気体の空気抵抗や、内部質量分布の偏った物体などの計算もサポートしています。一部機能の演算には CUDA を使い、GPU 搭載 PC では高速に処理できます。

→は地球に接近する隕石群のシュミレーターで、他にも H-2A ロケット打ち上げのシュミレーションも実装しました。

着陸船をエンジンとして扱うことで、高精度の演算を可能にしました。

レーザー砲台の照射による空気の膨張もシュミレートしています。(HTML 版では JavaScript の実行速度の関係上簡易的なものに交換しています)



④着地判定について

このゲームのキモとなるのが着地判定。実際の着陸船と同じく、減速した上で平らなところに着陸しないと墜落してしまいます。速度については運動を PVector で与えておけば `speed.mag()` で取得できます。地面の傾きに関しては、`map[]` 配列からプレイヤーの位置の右と左の高さを求め、その傾きを `atan2()` で計算し、許容範囲以内なら着陸成功としています。

⑥レーザー砲台について

レーザー砲台では線と円の当たり判定が必要になります。

ここではその解説を行いたいと思います。右図の直線 AC' をレーザーの射線、B を着陸船の場所とします。C は B から AC' におろした垂線。このとき、AC の長さは $AB\cos\theta$ になります。

$AB\cos\theta$ はベクトル AB とベクトル AC の内積を AB ベクトルの大きさで割ればいい事になります。

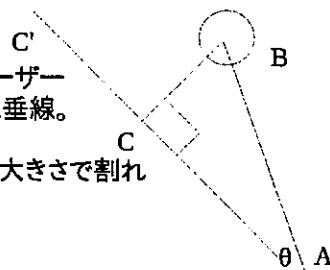
これは Processing では `AC.dot(AB)/AB.mag()`

で求められます (AB は AB の PVector)

あとは C'A をこの長さに合わせれば C の位置が求められます。

ベクトル AC-ベクトル AB をしてそのベクトルの長さを求めることで、BC の距離がわかります。

これがレーザーの半径+着陸船の当たり判定半径より小さければレーザーは宇宙船に命中しているという事になります。



⑦HTML 版について

近いうちに HTML 版を物理部無線版のサーバにアップロードする予定です。

Processing には Processing.js というラップがあり、HTML 内にコードを埋め込むことでインターネット上でも Processing を動かすことができます。速度は遅いながらもローカルと遜色ない機能を提供しており、PGraphics なども活用できます。また、JavaScript は COM ポートの操作にも対応しているので、(少し複雑なコードになってしまいますが) Processing.js から COM ポート経由でコントローラーを使うこともできます。

また、LeapMotion も JavaScript 経由でも利用できる所以 HTML 版で実装しています。LeapMotion を利用している方が(相当珍しいと思いますが)いらっしゃいましたら、是非体感してみてください。加速度センサによる操作以上に直感的な操作を行うことができます。

⑧感想

今年は IVRC のおかげかコンピュータを使った(ある意味物無らしくない)制作がだいぶ多いように感じます。それがいいか悪いかは別としても、物無に最近流行りの新たなプログラミング技術が導入されている事自体は素晴らしいことだと思います。

今までに使われてきた伝統的な技術を使うことも重要な意味を持ちますが、新技術を使うことによって得られるものもそれ以上にあることを今年の制作を通じて実感しました。

物無員には今まで以上に積極的に様々な技術に挑戦して欲しいと思います。

お詫び

ページ配分を誤って
しまい空白ページを
作ってしまいました。
ここに深くお詫び申
しあげます。

次のページからは再び目次通りに進みます。

~~ジョータのつもかけないですみません。~~

びーこん!

製作：高3 宮嶋優大

中3 磯村隆正

中村航

協力：物無の皆様

ストーリー

時はフロンティア時代、人類の開拓する地域は地球外にまで広がっていた。未知なる資源や土地を求めて、名のある大企業はこぞってトレジャーハンター達に投資をした。

開拓の技術は日に日に進歩し、世界中のトレジャーハンター達はロボットの遠隔操作や、高度な操作技術、管制センターからの支援情報を駆使して開拓を進めている。

そこで、諸君らトレジャーハンター候補生には最終試験として対戦形式のゲームを行ってもらう。

健闘を祈る。



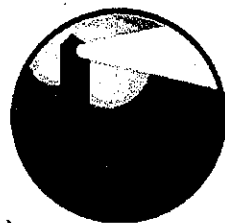
概要

豚肉の薫製ではありません。beacon（標識灯、無線標識）です。

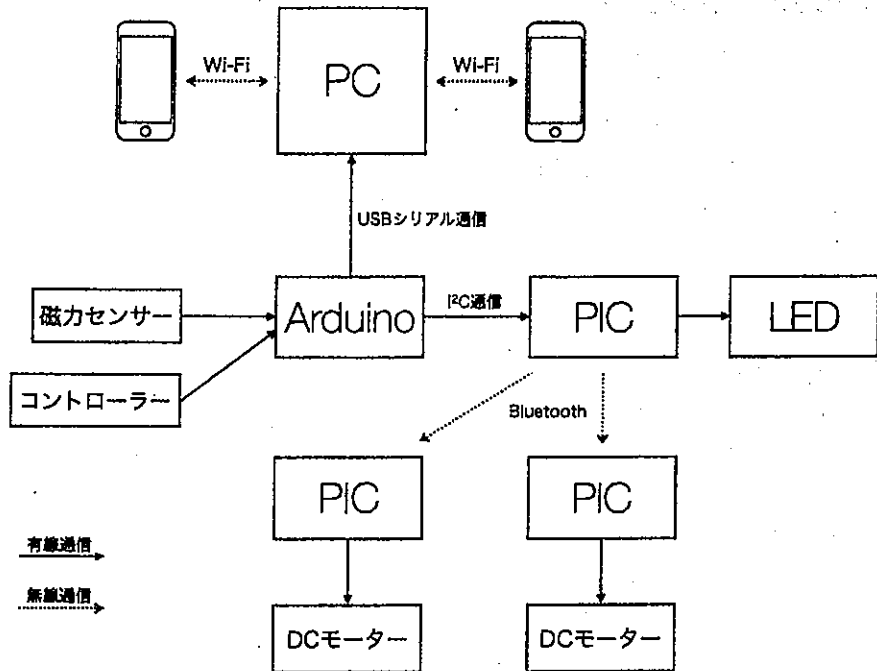
ロボットを動かしながら、ポイントを奪い合うゲームです。

このゲームのすごいところは、このゲームがスマートフォンでコントロールされること。そして、スマートフォンの画面の中と、現実世界の情報を使い分けながらゲームを進めていくというところにあるのです。

ロボットの走るフィールドには、色とりどりのLEDが光り輝き、さぞ美しい光景になるでしょう。旧名「Ride on the Light」。



ブロック図



回路、プログラム

回路

Bluetooth

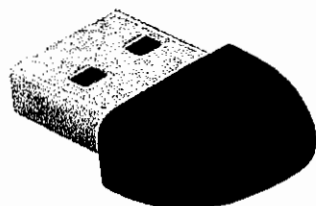
これはよく使われる近距離無線通信の規格です。iPhoneやandroidなどのスマートフォンにはもちろん、一部のPC、プリンター、ワイヤレスヘッドフォンなどに使われています。同じ近距離無線通信の規格にIrDAという、ガラケーなどによく使われる赤外線通信の規格があります。この二つの大きな違いは赤外線通信の通信可能距離は最大1m程度なのに対して、Bluetoothは機器間の距離が100m以内(Class 1の場合)であれば、壁などの障害物があっても通信が可能な点です。またBluetoothは、音楽などの大容量データもやり取りすることができ、消費電力が低いのも特徴です。

Bluetoothにはバージョンがあり、現在はVer4.1が最新です。このシステムでは一世代前のVer3.0+EDRを使っています。また、Bluetoothには電波強度を規定したクラスという概念があります。Class 1、Class 2、Class 3の三種類があり、Class 1は最大通信距離が100m、Class 2は10m、Class 3は1mとなっていて、このシステムで使うのは Class 1です。なので

理論上では本体から100m以内にいればどこにいても操作が可能です。しかし、障害物や他の電波の干渉によって実際の通信距離は7-8割、もしくはそれ以下になってしまいます。

Bluetooth Dongle

PLANEX社のBT-Micro3E1Xというものを使っています。これはパソコンなどにBluetoothインターフェースを追加するUSB接続のアダプタです。USBコネクタのついた親指ほどの大きさの機器で、Bluetoothに対応していない機器でもBluetooth機器と無線で通信できるようになります。PICそのものにはBluetoothを送受信できる機能を持っていないため、これを使用しました。



磁気センサー

A1324LUA-TというIC内蔵のアナログリニア出力ホール効果センサを使っています。0ガウス時の出力電圧2.5Vを基準として、1ガウスあたり5mVでリニアに磁力(磁束密度)を測定することができます。簡単に言うと、これは磁力を感知するセンサーでS極を近づけると出力が5Vに近づいていき、N極を近づけると0Vに近づきます。

これは自機の位置を特定するために使っています。磁気センサーを各マスのLEDの基盤に一つずつ付け、2つの自機の底面にそれぞれ下がN極、S極の磁石を付けました。自機を走らせると、自機の位置に対応した磁気センサーが反応するという仕組みです。

コンパレータ

これはオペアンプの一種で二つのアナログ入力の電圧の大小を比較し、デジタルで出力します。非反転入力の電圧が反転入力よりも高ければ、出力はHiになり、非反転入力が反転入力よりも低くなれば、出力はLowになります。このシステムでは、64ピンの磁気センサーのデータを圧縮するために使っています。コンパレータは片方の入力には磁気センサーに繋がれていて、もう片方は半固定抵抗に繋がっています。これによって半固定抵抗で指定した電圧を上回れば(もしくは下回れば)Hiになります。そして、すべての出力をエンコーダーに繋ぎ、6bit×2(正と負を調べるため)の12bitにします。64ピンをAD変換する訳にはいきませんからね。

LED回路

このシステムでは宝や障害物の表示のために使っています。全てで64マスあり、2色なので128ピンのデータを扱うため直接PICに繋ぐことは出来ません。そこで、中1ゲ-でよく使われる74LSICの74LS154を使いました。4bitのデータをデコードするものです。これをいくつか使うことで12bitにまで圧縮することが出来ました。しかし、これだと一度に1つのマスしか光らせることが出来ません。そのため、ダイナミック点灯をすることにしました。ダイナミック点灯とは複数のLEDを一瞬だけ順番に一つずつ点灯することで、人間の目の錯

覚によって同時に複数のLEDが点灯してみせる方式です。これにより、同時にLEDを点灯させることが出来ました。

プログラム

PIC

このシステムでは2種類のPICを使っています。

一つ目はPIC18F4550です。PIC18Fは8ビットファミリの最上位シリーズです。

PIC18F4550はUSBモジュールを内蔵しています。そのため、PCとのシリアル通信が可能となります。PCへの出力はArduinoで行いますので、PCからの入力もPIC18Fで行います。PCからシリアル通信で送られてきた、自機の移動情報とLEDの点灯させる場所の情報を受け取ります。そして、自機の移動情報を2つのPIC24FにI2C通信で送信し、LEDの情報はダイナミック点灯の信号に変換し、LEDの基盤に流します。

そして、二つ目はPIC24FJ64GB002です。PIC24FシリーズのPICマイコンは、マイクロチップ社の16ビットファミリの下位シリーズに相当します。実行速度がPIC16の5MIPSから、16MIPSとなって3倍になり、データ幅が8ビットから16ビットと倍になります。さらに16ビットの乗算器がハードウェアで内蔵されていますので、圧倒的な演算速度を誇ります。性能の割に価格が低く、価格性能比がもっとも良いマイコンです。特にPIC24Fの最新の28ピン/44ピンのものは、内蔵モジュールが100ピンクラスと同じ内容となっており、ピン配置に困ることが多いので、内蔵モジュールの入出力ピンを自由に配置できる「ピン割付機能」が内蔵されました。これで、内蔵モジュールの組み合わせを自由に使うことができるようになりました。さらに内蔵クロックだけで最高性能が出せるようになりましたので、これもピンの有効活用に寄与しています。このPICにはUSBホストコントローラが内蔵されているため、Bluetooth Dongleを介してBluetooth通信が可能になります。このシステムでは4つのPIC24J64GB002を使用しています。そのうち2つはPIC18Fから送られて来た自機の移動情報をBluetoothを用いて送信します。もう2つはBluetoothで送られてきた自機の移動情報によってモーターを制御します。

iPhoneアプリ

Unityというゲームエンジンで主にC#で開発されています。このUnityについて軽く解説します。Unityというのは主に3Dゲームを開発するためのツールで、これによって複雑な物理演算や他OSなどへの移植など、面倒で地道な作業を自動で行ってくれるのです。

Unityは通常版は無料で使えて、結構流行ったので知っている方もいるかもしれませんね。凝った作りのスマホゲームでは、これらのゲームエンジンが使われていることが結構あるので起動画面に注目していると中々面白いかもしれません。（彼の有名なInfinity BladeはUnreal Engineで作られている。）

このように便利なUnityですが、外部との連携などの融通がなかなか効かないので、面倒なことも多いです。Arduinoからの通信などはキーボード入力やマウス入力として処理しています。正直電子工作との連携などの用途には向いていないと思うので、iPhoneアプリをきちんと開発する気概がある人は使わない方が普通に作れると思います。VRなどで使っている人はいます。

Unityの悪口はこのくらいにして、プログラムの解説をしたいと思います。

UnityからArduinoへの情報の送信が実現できなかった都合上、Unity上の入力はUnityの中で完結しています。

Unityの機能に甘えまくって作ったので適当に飛ばします 無料のゲームエンジンは色々あるが、デバイス間のWi-Fi通信やNetworkViewという機能を使ってやり、ジャイロスコープの制御もUnityのライブラリで実装しました。UI関係はモバイル環境のUI開発には(画面の描画回数的な制限の所為で)必須と言われる、NGUIというUnityアセット(\$80くらいする)の少し古いバージョンが運の良いことに無料になっていたのので、それを利用しました。



UNREAL
ENGINE



Arduino

ArduinoはC++風のArduino言語というもので開発します。

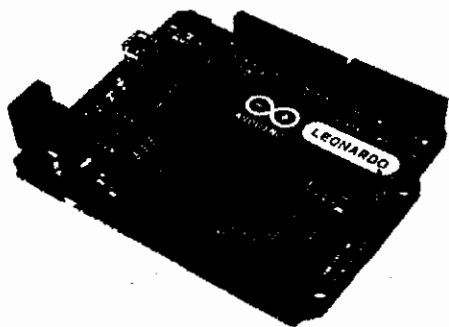
ArduinoはPCとPICやその他のセンサーなどを仲介するような位置づけで使われています。

PICとチップの性能はほぼ同等ですが、よりPCサイドの情報を処理しています。

具体的にはArduino Leonardo R3という製品を使用していて、これはATmega32u4というチップが内蔵されています。

他の人がもう既に書いているかも知れませんが、Arduinoというのは「Electronics

Prototyping Platform」と銘打たれているように、マイコン特有の難しくて長ったらしいコードを記述しなくても、簡単に電子工作のアイデアを形にできるように設計されています。特に今回使用したArduino LeonardoはそのままHID(Human Interface Device)として動作し、数行のコードでマウス情報やキーボード情報を簡単に送信することができます。Unityでは基本的に外部のデバイスとの連携を想定していないので、全て情報はHIDからの入力として処理しています。



感想

宮嶋

僕は去年はチカチカ隊長をやっていたため、あまり製作に参加できませんでした。なので、好きな物を全力で作れたのは今年が初めてでした。僕の担当はPICのプログラミングやPICの基盤の作成などでした。Bluetoothのプログラミングはかなりハードでしたので、現在はまだ完全には動いていませんが、当日までには完成させたいです。初心に戻って、中1基盤と大量の74LSICを使って基盤を作ったのも楽しかったです。あと、この製作物はロボット、コンピューター、ゲーム、アナログの全ての要素を含んでいるのは地味にすごいと思います。

磯村

最初の企画段階から紆余曲折あって、かなり血迷っていた感じがあったのでまだ手をつけていない深刻な問題が多数存在しています(2014年4月時点)。

社(部)畜パワーでここから一気に畳み掛けていきたいですね。

自分自身、決して多くの仕事をこなしたというわけではありませんが、iPhoneアプリ制作から最終的にはArduinoのプログラムの実装まで自分で行うことができたので、かなりいい経験ができたと思います。これを来年以降の製作に活かしていけるかなと思います。

中村

はい…今ゴールデンウィークなのに僕の感想がまだで回路図集が終わらない為に感想書きます…

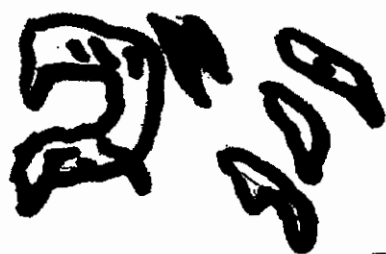
今回の制作は初めての何を作るか自由に決められる制作だったのでロボットでゲームしたいなぁと思いこのロボット?ゲーム?を作ったわけですよ。で、僕は本体を担当しておりました。ロボット本体とステージ(現時点では未完成)を作りました。ステージが終わってるかどうかは今心配なのですが未来の僕が頑張ると信じましょう。終わってるよね?終わってれば満足です。以上!

サヨナラ

製作者: M2 飯島

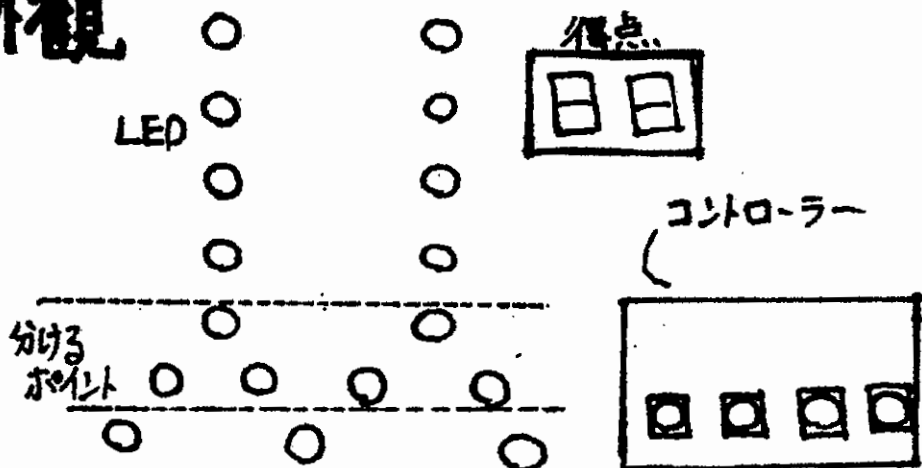
回路図: H. 山本 涼太郎氏

協力者: 物標の雫



の 日

外觀



ルール

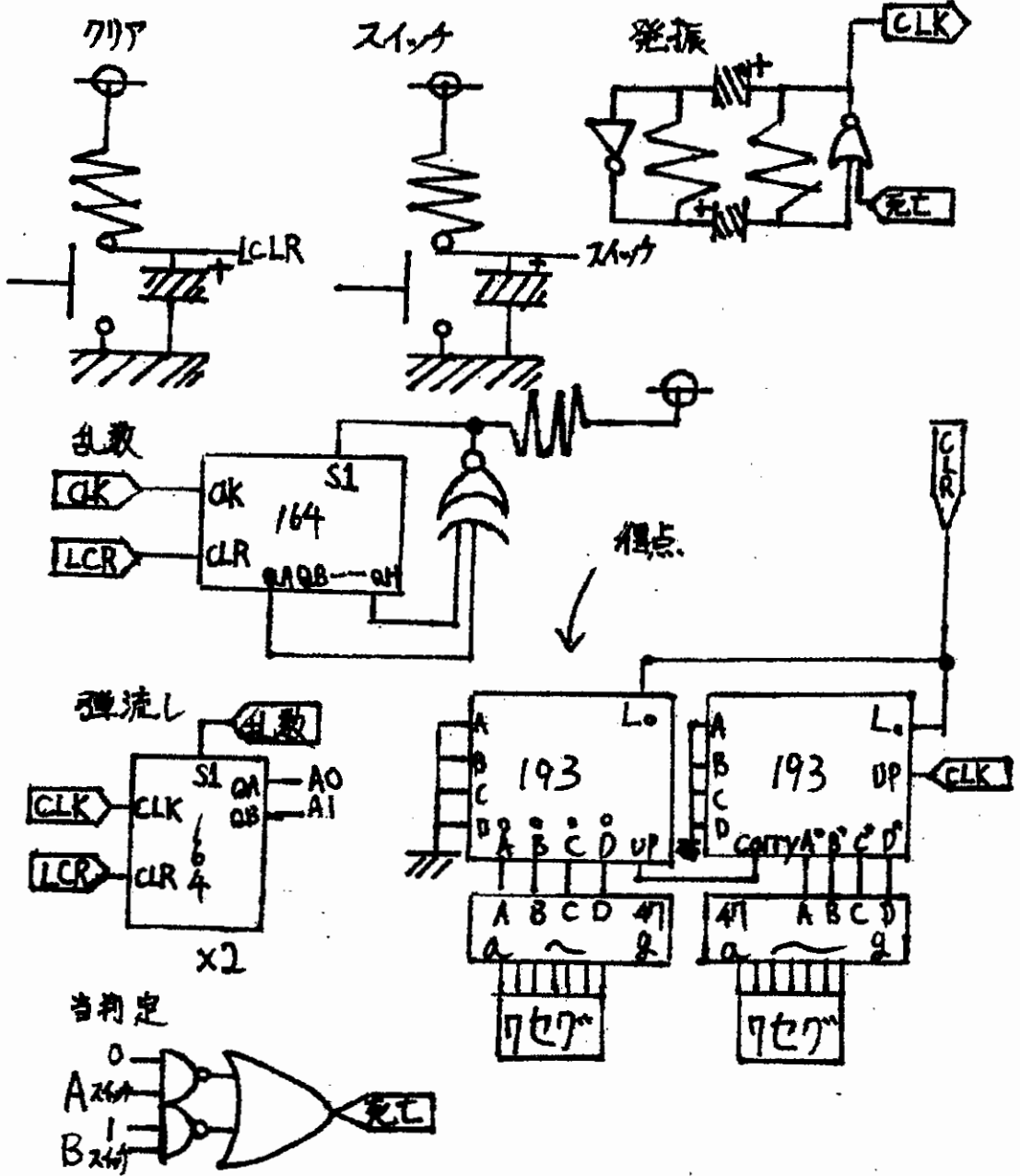
上から流れてくる赤と緑の球に、分けるボタンを押して、インコンを押し、人を通り、通過する。上から流れてくる赤と緑の球に、分けるボタンを押して、インコンを押し、人を通り、通過する。上から流れてくる赤と緑の球に、分けるボタンを押して、インコンを押し、人を通り、通過する。

ストーリー

生活に疲れた、毎日流れてくる赤と緑の球に、分けるボタンを押して、インコンを押し、人を通り、通過する。生活に疲れた、毎日流れてくる赤と緑の球に、分けるボタンを押して、インコンを押し、人を通り、通過する。

回路

※下のようにつくりも完全に動作しない恐れがあります。



感想

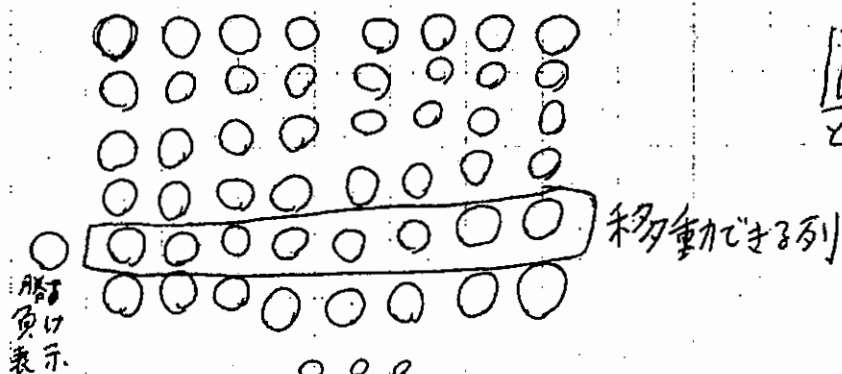
僕自身が電子工作を今までやったことがなかったので、最初は配線材をむくのにも時間をかけてしまい、あせ、た時期もありました。か、物無の監設計者さんの熱い指導もあって、ようやくできました。自分なりに、力を入れて、色かすけようとしたのが一番のポイントです。

The

制作: M2 注田
回路: H1 石倉 Jk

Block

外觀



ルール

1. 上から落ちてくるブロックをよける。
2. ブロックは下にたまっていくので消す。
3. 二十個消せば勝る。自分が落ちてきたブロックにあたりと負け。
4. 勝つと左の勝負け表示のLEDが緑、負けると赤に光る。

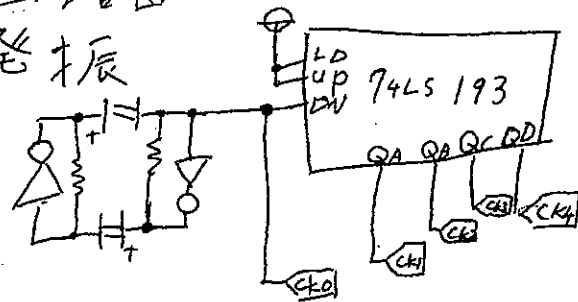
ストーリー

朝起きるといきなり四角い光となっていた。

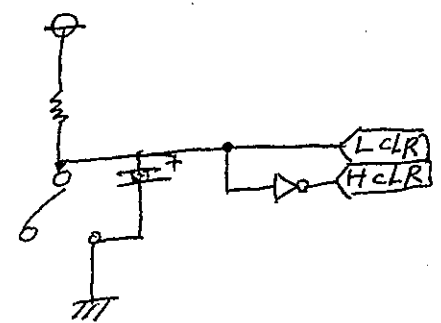
あたりを見まわれ 上から無数のブロックが！
この光にな。たつもりで一緒につるぞう！

回路図

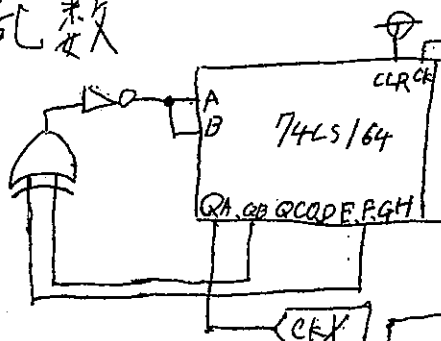
発振



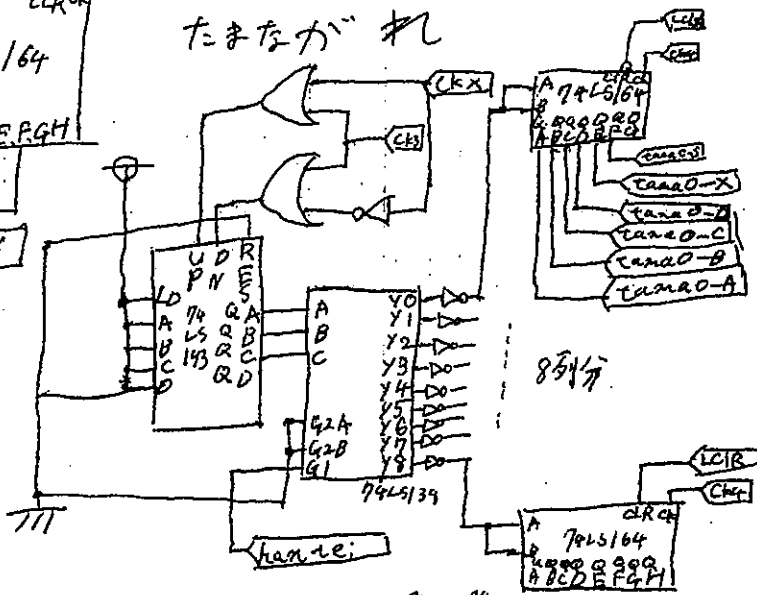
リセツ



乱れ

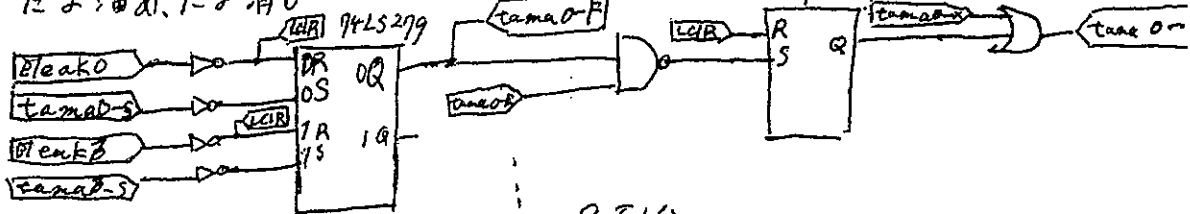


たまなか

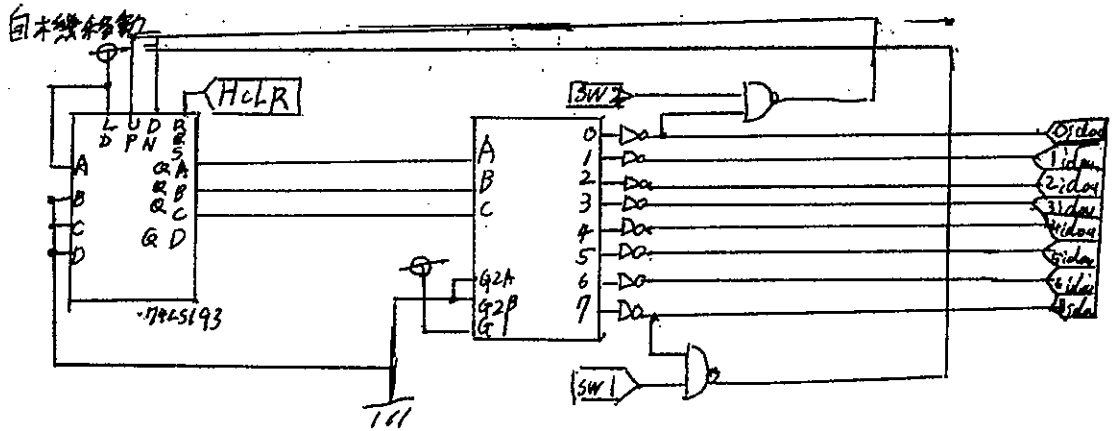


8分

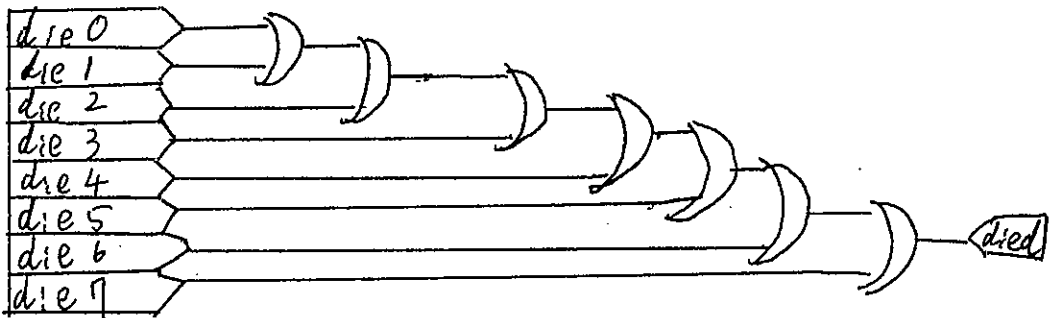
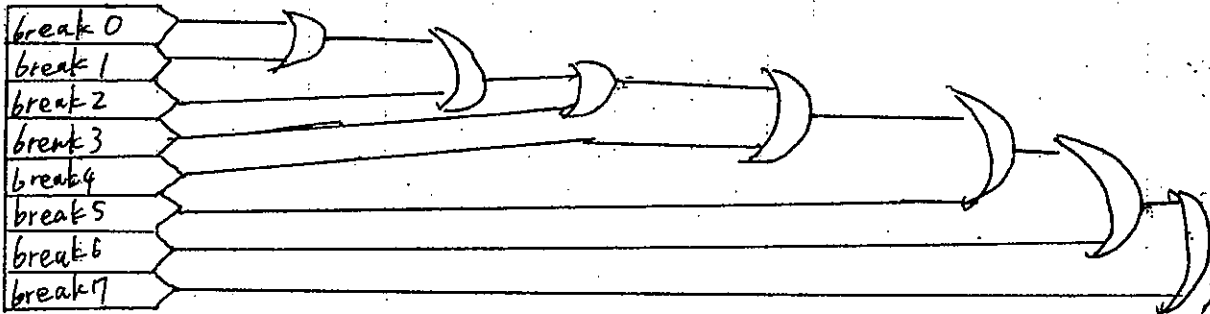
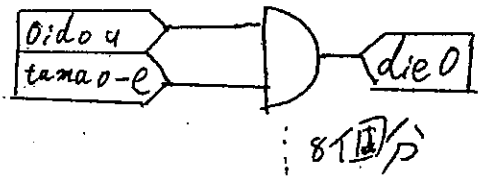
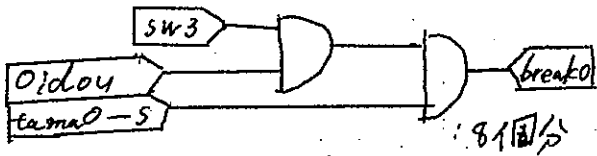
たま溜め、たま消し



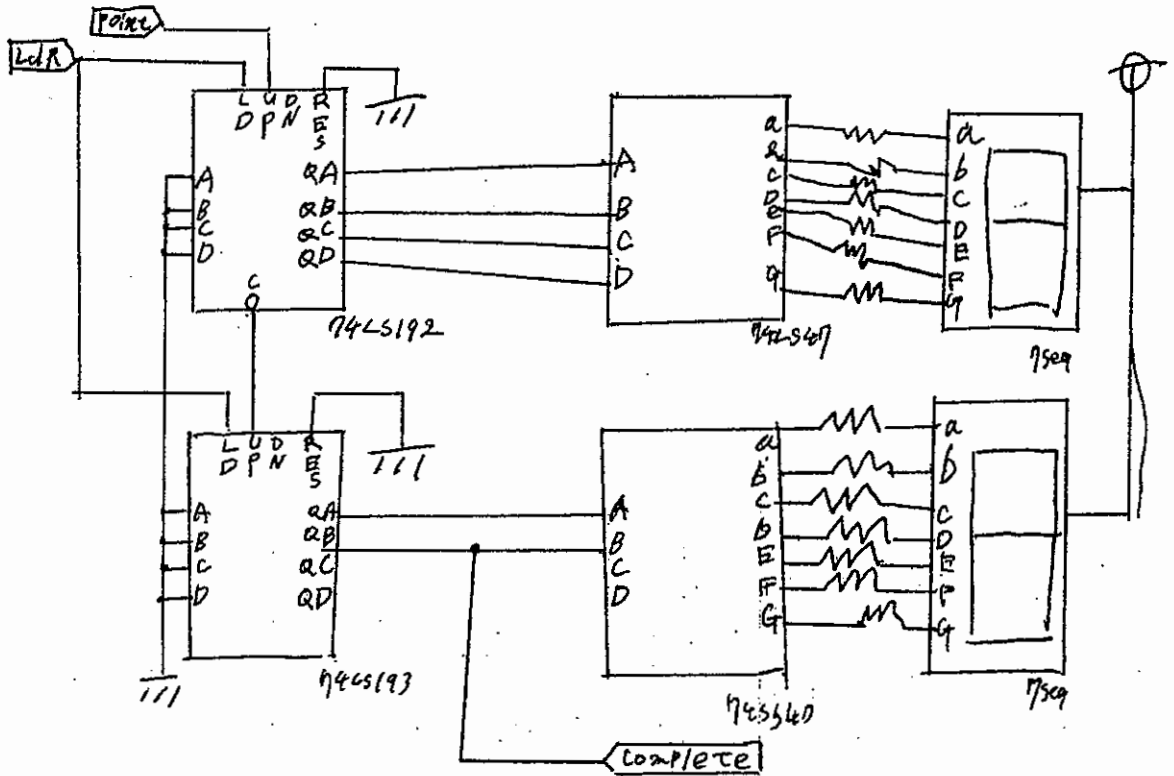
8分



当在0判定



カウント



感想

僕の初めての制作は、かなり難関か、とでも。来年はより難しい物を作っていくことになっていきますが、完璧力が25%か、んばりたいです。

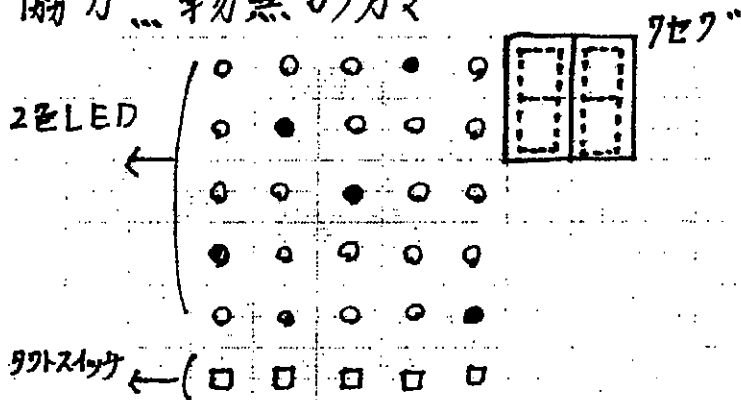
T M F M

～首都防衛作戦～

製作者... 中二高橋

回路図設計者... 高一渡辺さん

協力... 物無の方



〈ルール〉

天から降りてくる緑の球の列のスイッチをおすと加算。緑の球が地に落ちるとゲームオーバー。空中で夫々輝く赤の球をおすとゲームオーバー。

〈物語〉

あるとき日本で「TMFM」という映画が作られ、世界中でヒットしました。これに目をつけた中国の人々は、よく映画とは共通点があるしかならないニセモノを作りました。これがそのゲームです。後に、勝手に公式ゲームといわれたとして、つかまてしまいましたとさ。

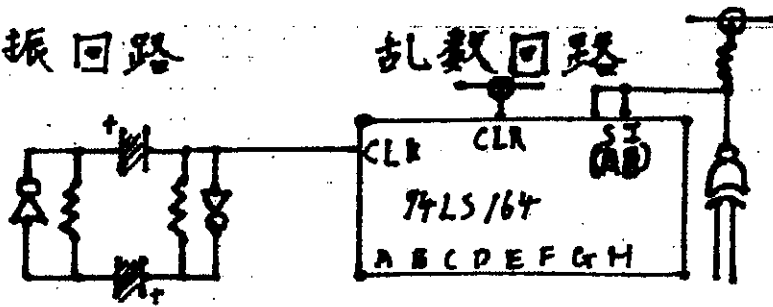
〈感想〉

はじめの製作でしたか楽しかったです。ゲームの名前をまじめに考えれば良かったと心の底から後悔しています。

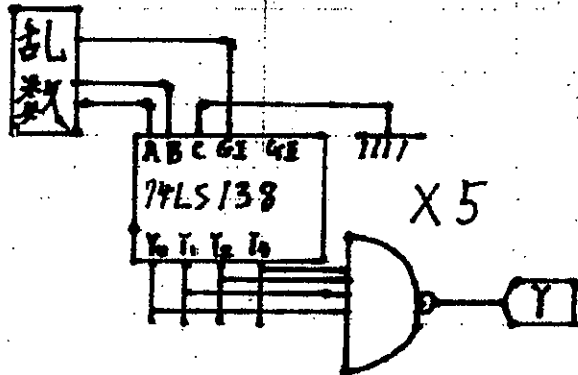
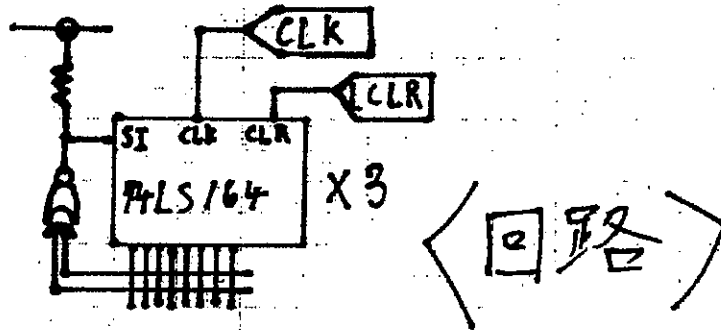
2

発振回路

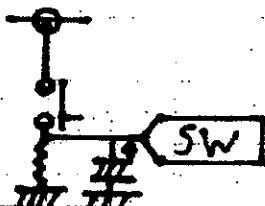
乱数回路



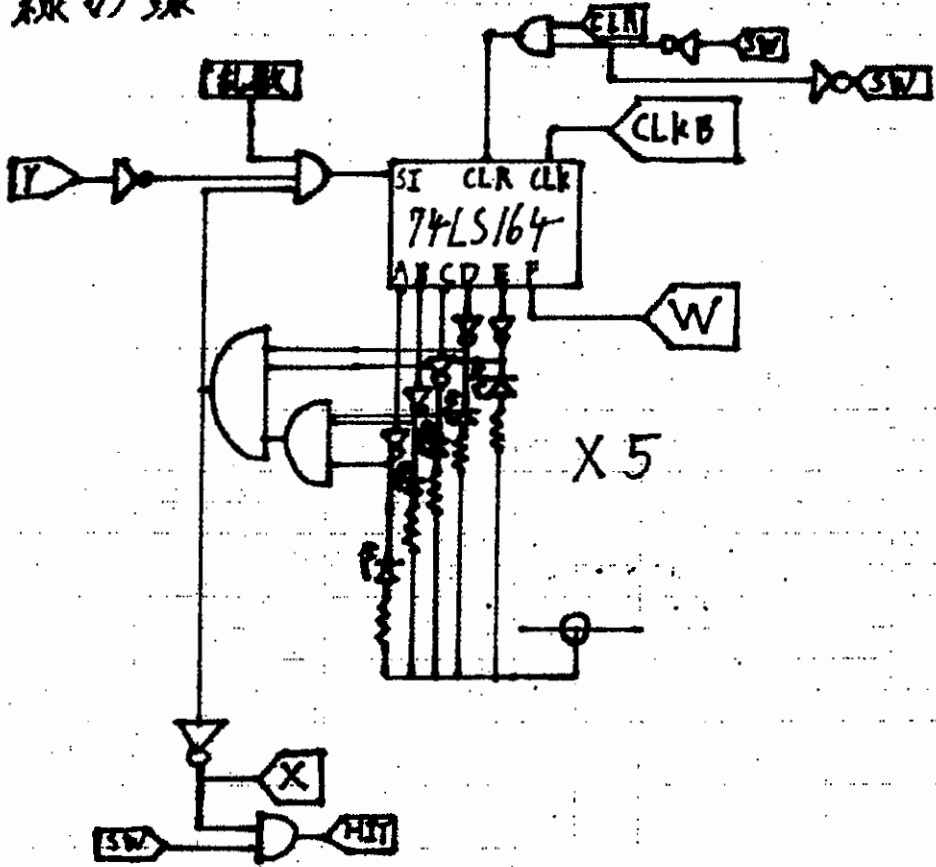
乱数回路 + 赤い球



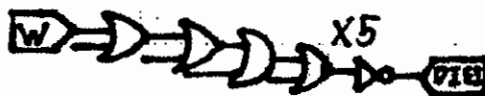
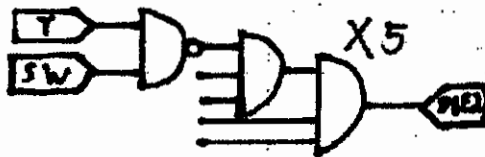
スイッチ



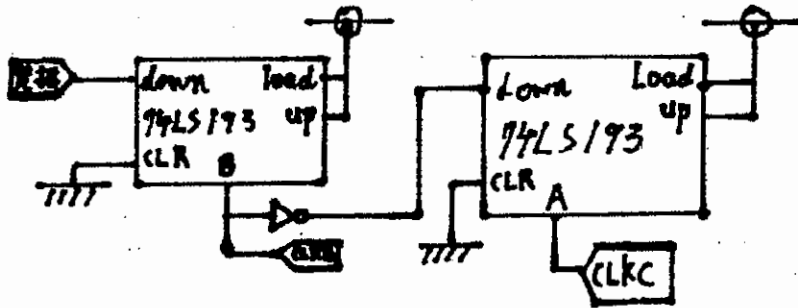
緑の球



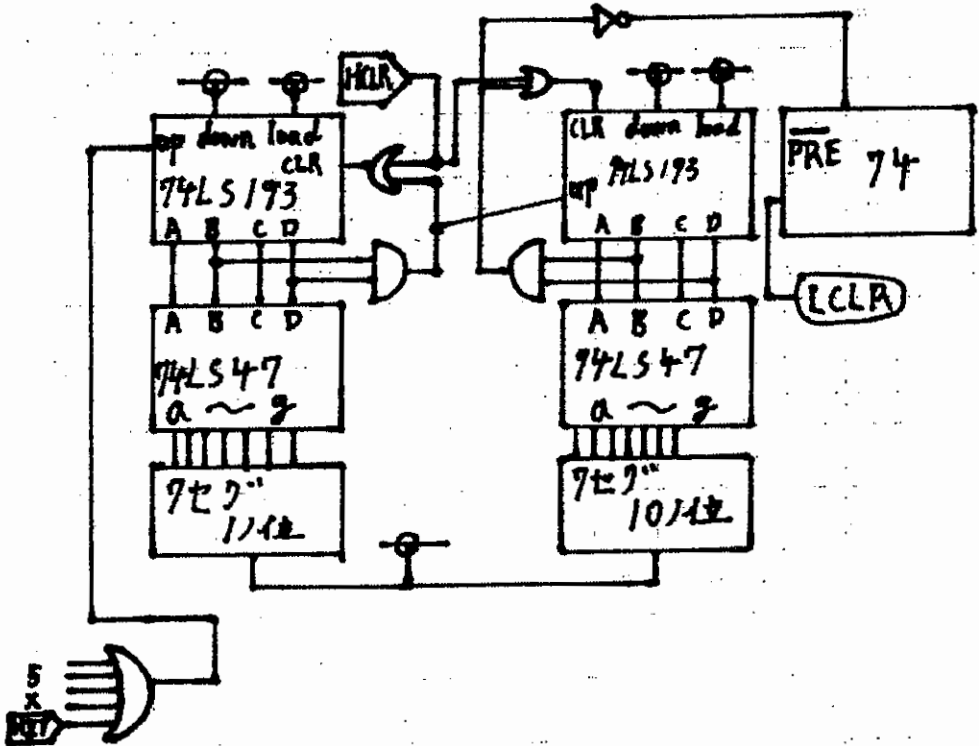
死亡判定

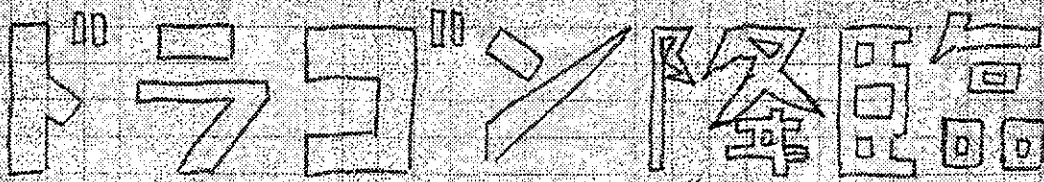


4



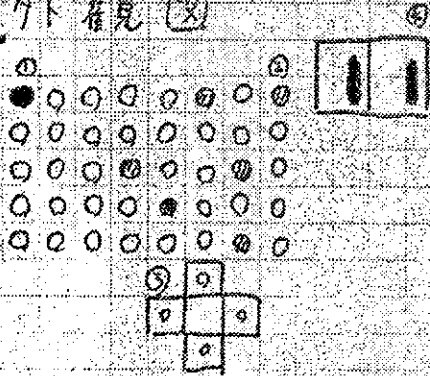
得点回路





*吹ぬてく子の火炎弾です。

1. 外観図



- ①...自機 (青王)
- ②...敵 (赤王)
- ③...上下左右移動スイッチ
- ④...制限時間

2. ルール

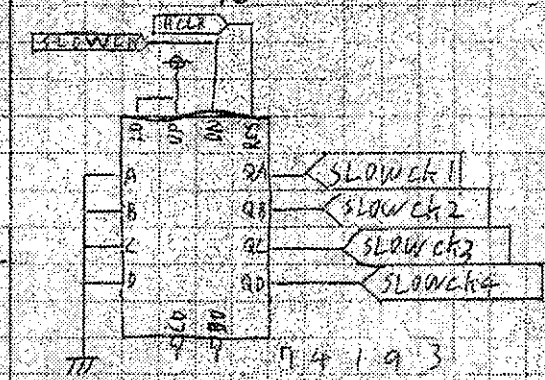
60秒以内に一番右の列に到達すればクリアですが、その過程で一回でも敵にあつたか、たゞアウトです。

3. 回路図

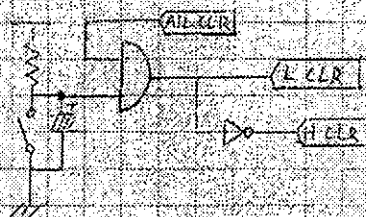
発振回路



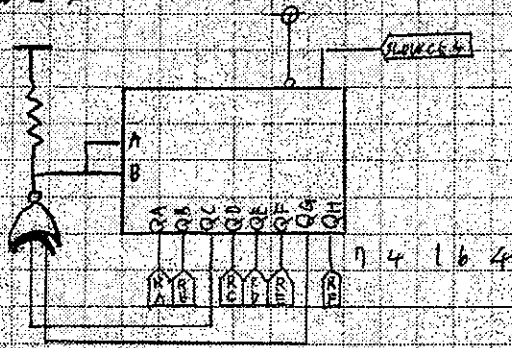
遅い発振回路



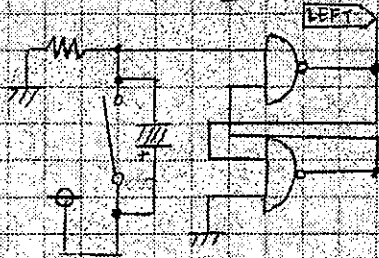
クリアスイッチ



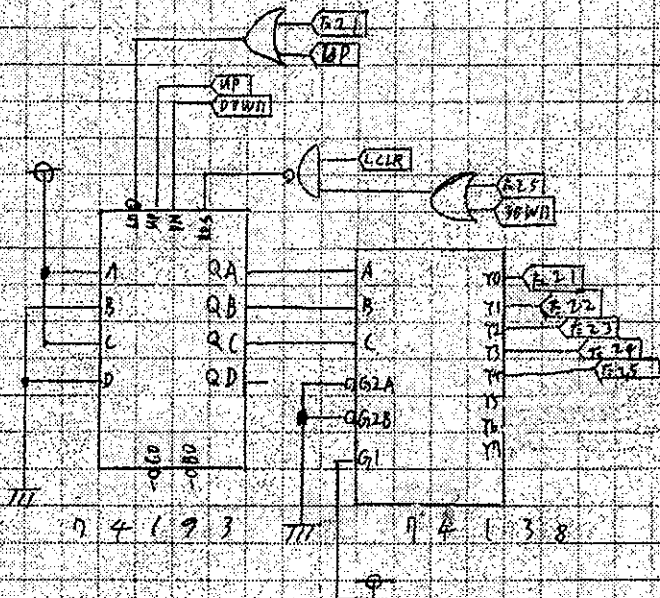
乱数回路



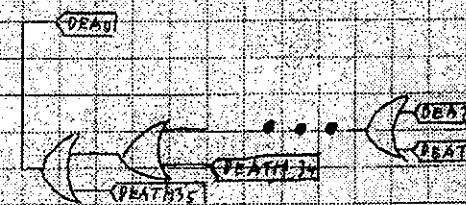
上下左右スイッチ



自機移動回路



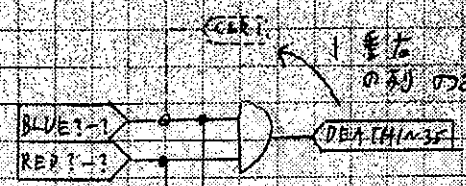
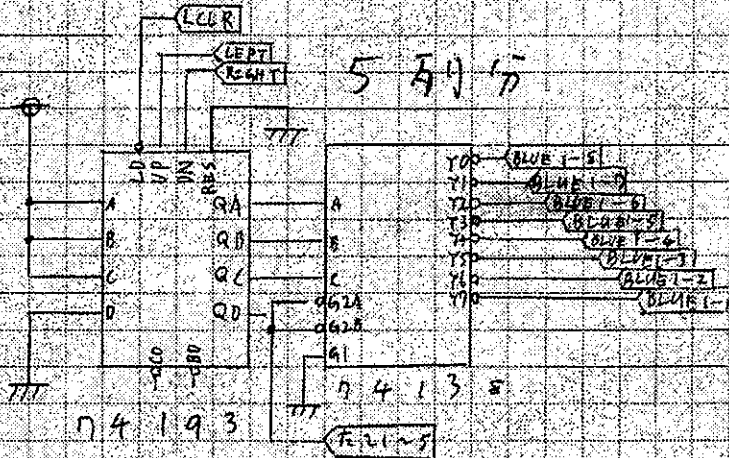
死亡判定



本当の死亡判定

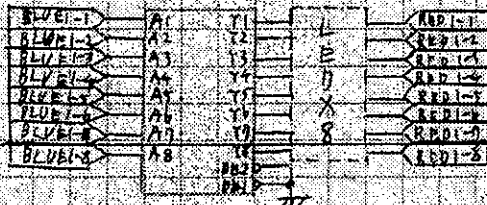


5列分



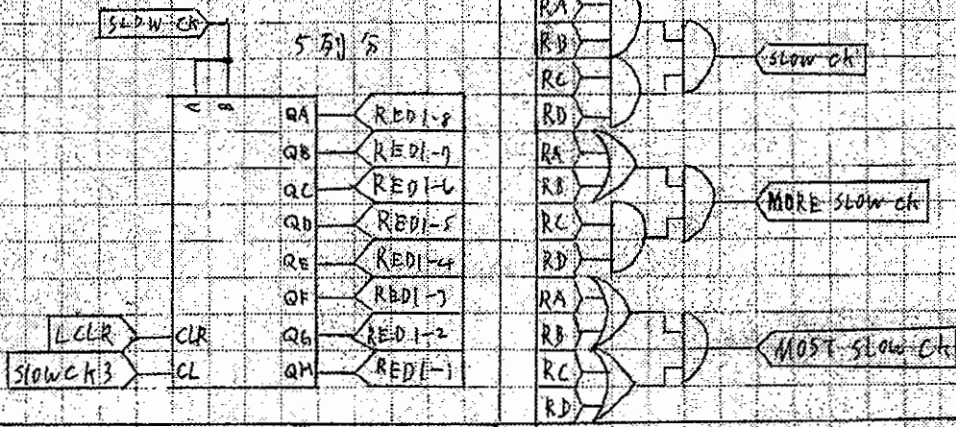
LED

5行分



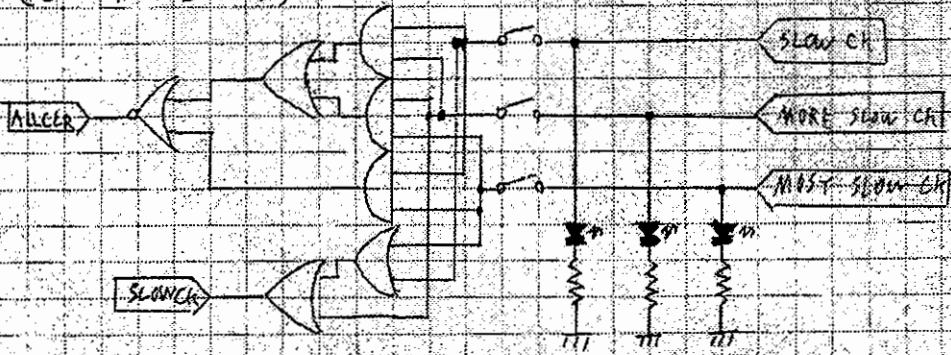
玉流し回路

乱数調整

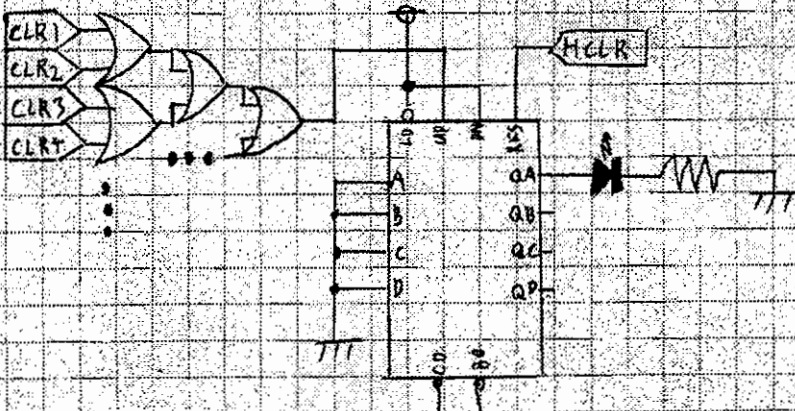


2ホック同時入力

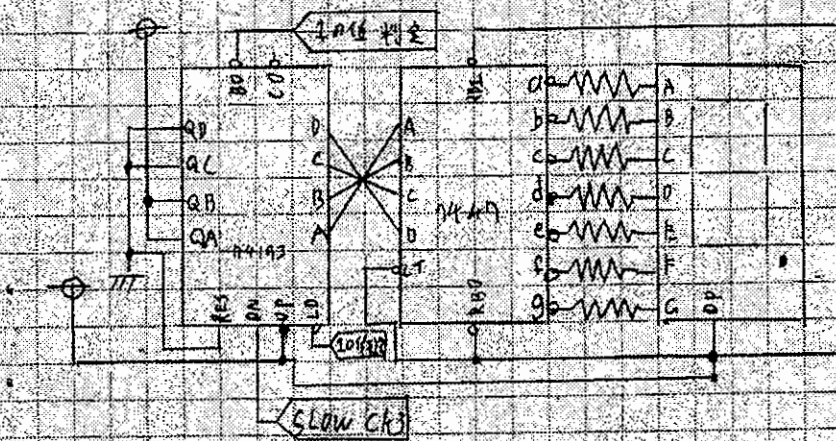
(E-H4=2) 防止回路



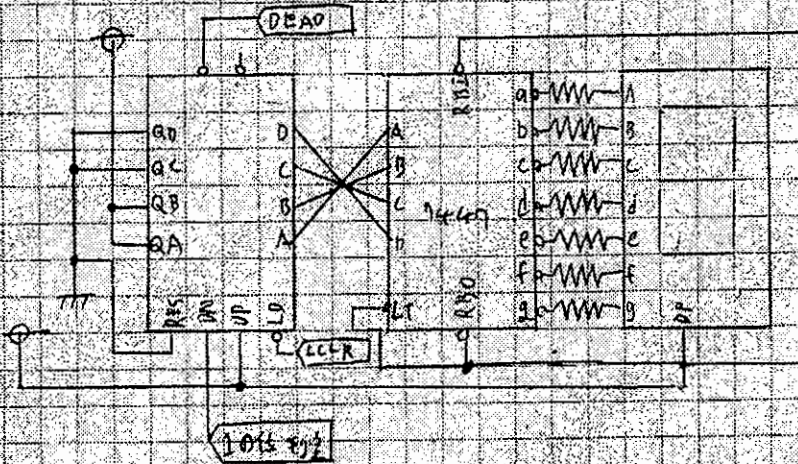
クイック時エフェクト



7セグメント：1の位



7セグメント：10の位



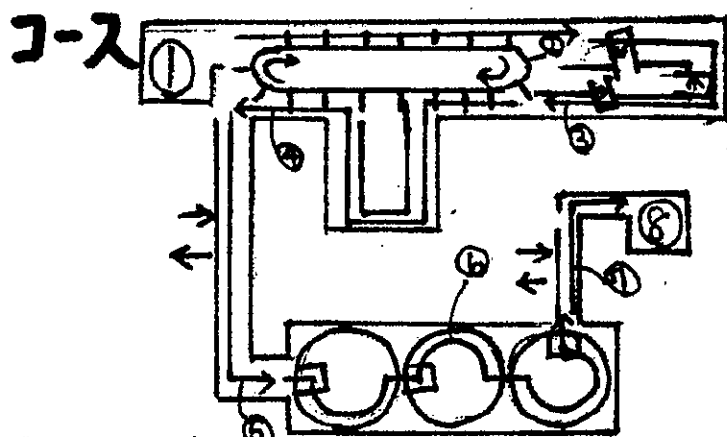
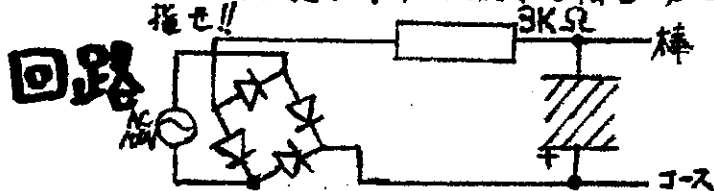
感想

企画書を書いた時点では単純な作業を考えたつもりでしたが、実際に作り始めるととても難しく、何度もケツスえをしごしました。次の制作の方も、と丁寧な作りにしたいと思う。

人 木 泰

～回路：M2高橋，本体：M2 飯島～

ルール 金属棒を、金属でできたコースに当たらないよう遊んでいきましょ。ボールを指すゲームです。もしもコースに当たってしまったら、ボールを指すうと、あなたの想像を絶する大爆発が!! (火花が散りましょ。あなたに襲いかかる様な障害物をよけながら進んでいって!!)



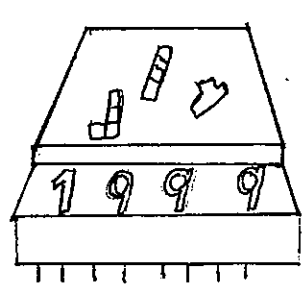
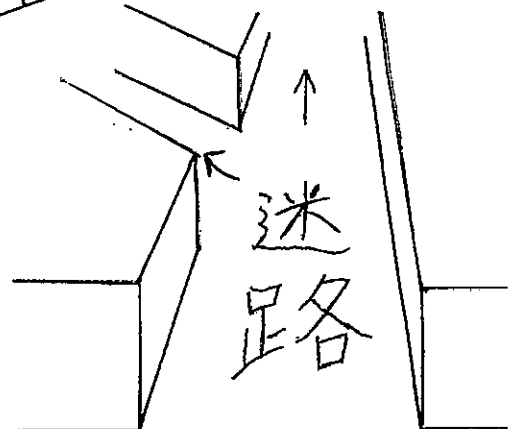
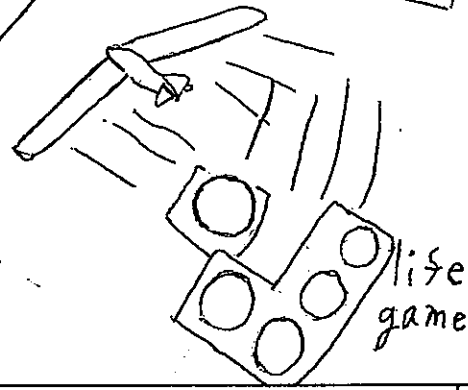
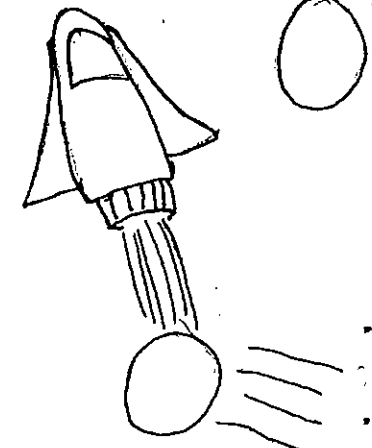
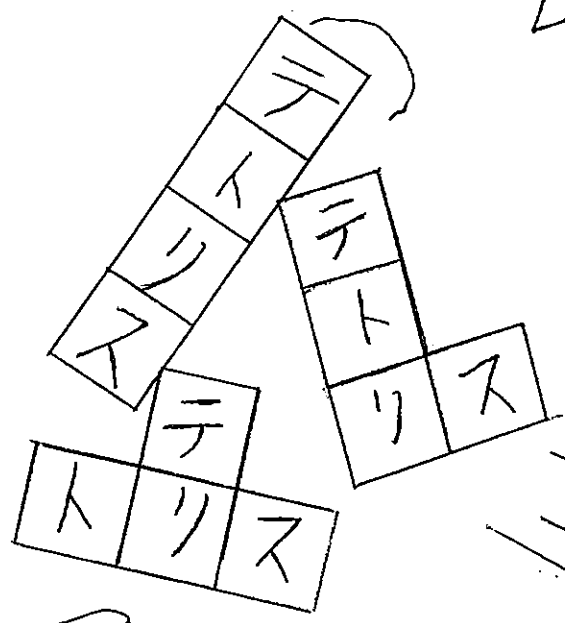
①. スタート ここから伝説は始まる --! 最初は木で守られているので安心な

②. キャタピラ キャタピラの動きに合わせて動きましょう。手の震えは命取り!

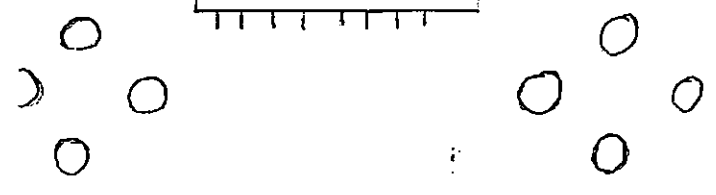
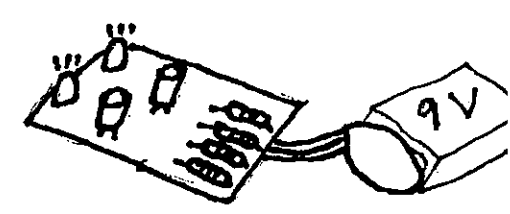
③. 逆風車 風車が進行方向と逆に回っている...! うまくタイミングを掴みましょ。

④. 再来のキャタピラ もう一回キャタピラを通ります。道が狭いので加減をせまらましょ。

士
売
り
物



チ力チ力



Silent Gamer 1999

製作：M3 関根史人

協力：物無の皆様

概要

"Silent Gamer 1999"は去年の12月頃に設計され、量産が開始された最新モデルのゲーム機です。

8x8のドットマトリクスと7セグメントLEDを利用し、豊富なゲーム表現が可能です。

更に、メインコンピューターの"PIC16F1938-I/SP"の豊富なデータ領域を利用し、メインコンピューターに5個以上のゲームとアプリを搭載、非常に楽しめるボリューム感あふれる売り物となっております。

入力部に関しては8個のタクトスイッチを無理ない形で押せるようにされているので、スマートフォンとかいうガラケーつぶしの無線機と違って指が疲れず、無理なく安心してご利用いただけます。

ここまでボリュームが高いと消費電力が気になる方も多いでしょう。この文章を読んでもる方の中にも充電切れに悩む方も多いことでしょう。

しかし、ご安心ください！

Silent Gamer 1999は"省エネ"にも対応しておりますので電池を交換どころか1日中抜く必要がほとんどなく、充電電池と組み合わせればかなり"コスパ"がよくなります。

電池に関しましては単三電池二本で動かすモノとなっております。

南蛮から押し売りされている製品で動かない電池も"Silent Gamer 1999"なら動きます。

そう、"Silent Gamer 1999"ならね

詳細な材料については下記の通りでございます。

・B基板(去年より一回り大きいです。一昨年と同じですが、エッチングと

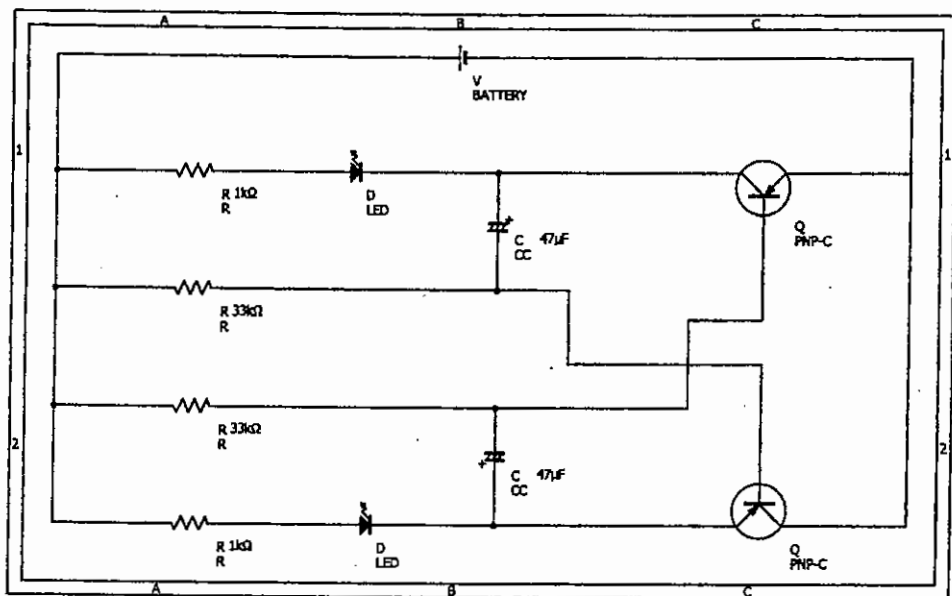
チカチカ

M3 川本惇之助

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、二つのLEDという物が交互に輝く、その名の通りの物です。その原理は、電池から流れた電気がコンデンサーに溜まり、その電気がLEDに流れ、それが二つ交互に行われる、というものです。容量の違うコンデンサーを使えば、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無い方でも可能です。



↑
回路図です。

麻布学園物理部無線班

回路図集 2014

発行日：2014年6月8日

編集責任者：黒田 健太、四柳 雄太

販売責任者：四柳 雄太

デザイン：花園 佳月

石倉 匠

市川 詠亮

佐伯 大地

磯村 隆正

関根 史人

飯島 勇人

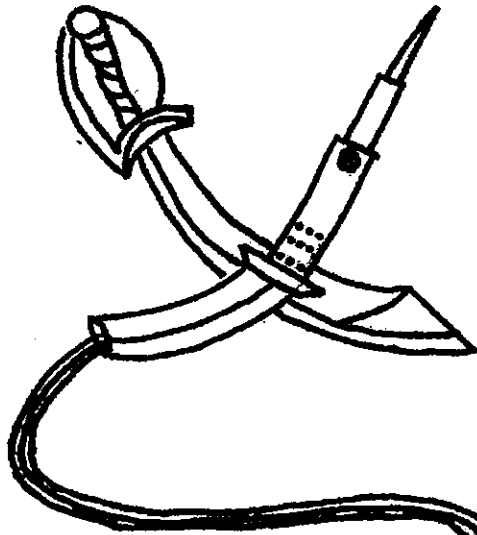
庄田 拓海

乱丁、落丁はお取替えいたします。

転載、複製は自由です。

麻布'67

回路図集



2014年

無線班
物理部

