

目次

目次・まえがき	1
部員自己紹介	2～4
モールス信号キーボードの自作	背の高い草 5～6
素人と学ぶミサイル	眠い人 7～12
(前略)機械学習プログラム(笑)	よつばのくろおばあ 13
ブラウン管の仕組み	O.T. 14～15
(前略)携帯電話はどのように通信しているのか	こすみ 16～21
概念の哲学——剣持刀也氏に学ぶ(略)	Qibr(略) 22～25
オンラインでポーカーをしよう	K.T. 26～28
ペーパークラフトを設計しよう	こめんつあんと 29～32
ロボレーブ参戦記	部員一同 33～44
(前略)俺は CPU のママになるんだ!!!	おいしいドーナツ 45～53
あとがき・その他	54

まえがき

本日は 2025 年度第 79 回文化祭にお越しいただきありがとうございます。
います。

アマチュア無線研究部ではアマチュア無線のみに留まらず、ロボットプログラミング、電子工作、3D プリンターによる造形、3DCG など多岐にわたる活動を幅広く行っております。文化祭の展示ではこの一年間の活動の成果を見られるものとなっているので、楽しんでいただけると幸いです。また、作品展示だけでなく電子工作教室やロボットプログラミング講習なども行っているため、ぜひご参加ください！
また、作品展示で分からないことや気になったことなどあればぜひお聞きください。これからは灘校文化祭をお楽しみください！

(※電子工作教室とロボットプログラミング講習は事前の申し込みが必要です)

部員自己紹介

部員自己紹介

※()内の学年については文化祭時点での学年を表記しています。

83 回生(中一)・新高 80 回生(高一)

つい先月入学したばかりの一年生たちです! 残念ながら自己紹介を載せることはできませんが、初めての文化祭ということで楽しみながら参加してもらえると嬉しいです。

79 回生(高二)

・ T.R(部長)

79 回生の T.R です。部長です。

最近ポーカールにハマっていて夜しか寝れません。ツーペアやスリーカードで待機してるときにフルハウスが来たときの感動といったらありはしないですよね～。去年の加賀ロボレーブでは決勝進出して、結果は同率 5 位でした。あと 3 アマ持ってます。

・ おいしいドーナツ(会計)

会計のおいしいドーナツです。部誌の編纂をした偉人です。CPU 自作をここ半年くらいの功績としています。ローグライクが好きですが、最近ドハマりするゲームを見つけれていません。
あと最近 YouTube の動画すべて見尽くした気がします。

・ こめんつあんと

79 回生のこめんつあんとです。少林寺拳法同好会、生物研究部と兼部しています。最近は語学と言語学に浮気中。
国際生物学オリンピック日本代表のなり損ない。かなしい。

・ Kohenyan

79 回生の Kohenyan です。ほぼ幽霊部員。
パソコン部の部長やってるかもしれません。

・ ヌーガー

79 回生のヌーガーです。部誌の表紙を担当しました。
高 2 にも関わらず入部歴半年の半端者です。

- ・キノコ

79 回生のキノコです。こんなところを見てくれてありがとうございます。

- ・Qibr_1. 1. 2. 3. 10. 7

私のことはどうでもいいので、
VRChat のネメシスちゃんをよろしくお願いします

- ・K. T.

ふざけた文章を書けるようになりたい

- ・pitapa79@青シャツ

79 の pitapa79 やで 3 ハム持ちです
去年は登校時のシャツが 100%青色を達成しました
多分文化祭当日も水色のシャツ来てます
あとデジタル委員会のお偉いさんやってます
デジタル委員会でえらいので部誌は書けません
ごめんねポセイドン

- ・N. S.

79 回生のそすいーです。競技プログラミングしてます。

80 回生(高一)

- ・O. T. (副部長)

副部長です

- ・眠い人

最近友達で行ったカラオケで歌いすぎて喉が死んでる 80 回生の眠い人です。
3D プリンターを使ってみたいけど何も分からないまま気付いたら入部から
2 年がたってました。

- ・よつばのくろおばあ

アマ研の半幽霊部員。
現在、本格的な幽霊部員への降格の危機に瀕している。

81 回生(中三)

- ・mm

身長が伸びました

部員自己紹介

- N. A

なんかしたい

- S. S.

部誌も作品も~~もちろん~~できていません。(3/31 時点)

- こすみ

81 回生のこすみです。最近はやodobashicameraにハマってます。

店舗も Web も。買い物はほとんどヨドバシです。一時期、自分でアフィリエイト(成果報酬型広告)をやっていたため、広告の闇をある程度理解しており、アフィリエイトアンチです。アマ研の中で唯一

アマチュア無線をやってます。僕のコールサインは JQ3KHO です。

アマチュア無線はもちろん、移動体通信(モバイル通信)やネットワークが好きです。あと決済も。メインカードは住信 SBI ネット銀行のプラチナデビットです。年会費はもちろんキャンペーンで還元です。

将来は JGC 修行をしたいです。赤組です。使ってるキャリアは docomo と KDDI で、プランは eximo と povo です。eximo はもちろん U18 割でほぼタダ運用、povo も使ってません(docomo 障害対応&旅行用です)。

家のネットは光コラボでドコモ光です。マンションの設備が VDSL なので、100Mbps までしか速度がでません。フレッツ網内、障害発生時のために Starlink を契約したいのですが、本体代が高いです。二次元の女の子には興味がありません。マジで。推しとかマジでいないんで。今一番取りたい資格は、総合旅行業務取扱管理者で、一回落ちました。ここまで、結構お金を持ってる雰囲気醸し出していますが、かなりケチで、お金はあんま持ってません。最後にですが、X(旧 Twitter)にもハマっていて、ユーザー名は@7834896534_です。フォローお願いします。

82 回生(中二)

- 背の高い草

バスケットとアマ研のハイブリッドです(特定可能)

応用情報技術者試験にチャレンジ中です。何個展示を出そうかな～

- 午前ティー

82 回生の午前ティーです。加賀ロボレーブ出ました。npca に入ってます。

- Y, Y

アマ研とパソ研に入ってます。機械&パソコン大好き

私たちはモールス信号で打ち込めるキーボードを製作しました。モールス信号というのは

楽曲「はいよろこんで」に出てくる「・・・－－
－・・・」のように・(短)と－(長)で文字を表す方法の一種で、無線などで使われていました。

なので、モールス信号を打ちなれている人やモールス信号を少し触ってみたいという人のために作ったのがこのキーボードです。

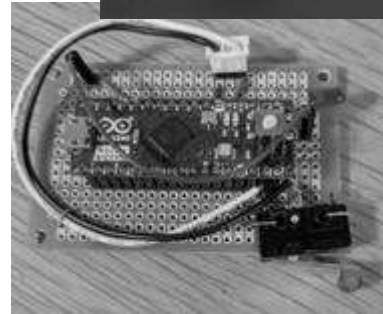
内部の構造は、主に Arduino micro (以下、Arduino)、スイッチ、LED で構成されていて、Arduino の USB 端子にパソコンを接続して、スイッチでモールス信号を打ち込むとパソコンに有線キーボードをつないだ時のような挙動をします。

Arduino micro を使用した理由はキーボード化の簡単さにあります。Arduino micro はその他の Arduino ボードに比べてシリアル通信の IC が単純なのでファームウェアを更新することなくライブラリの使用だけでパソコンに設定なくしてキーボードとして認識させることができます。このことで使い勝手のよさが上がります。

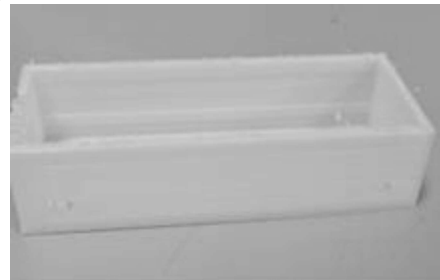
プログラムは次ページの QR コードから GitHub 経由でダウンロードしてください。コードは自分が書いたものをたたき台にしてデバッグを ChatGPT にしてもらったものです。

苦勞したこと

- ・ Arduino micro はエラーがなくても赤字でメッセージを表示する
 - ・ USB microB ケーブルが抜ける
 - ・ 箱の印刷ミス
- (3D プリンターで制作したため)



↑ 中の基盤



↑ 印刷失敗。これで半日無駄になる。

モールス信号キーボードの自作

以下コードの大まかな説明 by ChatGPT

必要なハードウェア:

1. ボタン (1つ)
2. 抵抗 (10k Ω) (プルダウン抵抗として使用)
3. Arduino ボード (UNO など)

接続:

- 1 ボタンの片方を pin 2 に、もう片方を GND に接続。
- 2 抵抗を pin 2 と 5V の間に配置 (プルアップ設定の場合は不要)。
- 3 状態確認用の LED を pin 13 に接続 (オプション)。

プログラムの特徴

▮ **isPressing フラグ**

ボタンの押下/離しを確実に区別するためのフラグを追加しました。

▮ **lastInputTime の正確な更新**

文字入力の区切りを正しく検出するために修正しました。

▮ **デバウンス不要**

長押し・短押しの判定により、物理ボタンのバウンスを無視できるように設計しました。

▮ **シリアル出力の見やすさ改善**

各文字間にスペースを追加しました。

まあこんな感じで AI がコーディング (少なくともデバッグは) してくれる世の中になったので、皆さんもぜひ活用して Arduino なんかで遊んでみてはと思います。

Thank you for reading!



皆様おはこんばんにちは、眠い人です。

今回調べたのはミサイルの誘導方法についてです。

とはいうものの、私の知っていることなんてレーダーが電波の反射を測って距離を求めるものということぐらいなのでレーダーホーミングミサイルなんて分かりません。

そんなわけで赤外線誘導ミサイルの、それもシーカーに焦点を当ててみました。

それでは、素人と学ぶミサイル、始めていきましょう。

あ、それはそうとして素人なので調べることや理解することを放棄した部分や誤解している部分も多いと思いますが無視してください。

1, 比例航法

ご存じの通り、ミサイルは逃げても追っかけてくるものです。

では、彼らはどうやって目標を追いかけているのでしょうか。

方法はいろいろあるみたいですが、指令誘導とかGPS 誘導とかは無視してホーミング誘導の話をしします。

ここでいきなりですが皆さん、ゲームや映画で「fox1」とか「fox2」とか言っているのを見たことがありますか？

あれは空対空ミサイルに使われることがほとんどのようですが、ホーミング誘導の分類はまさにあれの通りで、主に3種類に分けられ、fox1 と fox2、そして fox3 があります。

これらに共通しているのは、目標からの電波や赤外線といった電磁波を元に追尾する仕組みであるということです。

では何が違うのかというと、追尾している電磁波が何かという点です。(明らかに説明力の低い文)

Fox2 はおもに **IRH**(赤外線ホーミング) などの目標本体が発しているものを追尾するものに使われます。

それに対して **Fox1** と **fox3** は主要なものにそれぞれ **SARH**(セミアクティブレーダーホーミング)と **ARH**(アクティブレーダーホーミング)というものがあります。

これらはどちらもレーダー波が目標に反射したものを追尾するもので、違いは目標から反射してきたレーダー波が、**SARH** では発射母機の発したものであるのに対して **ARH** はミサイル自身が発したものだという点です。

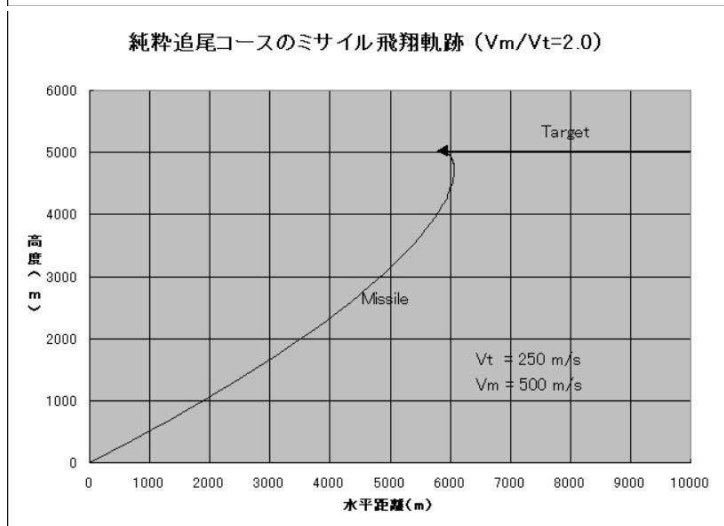
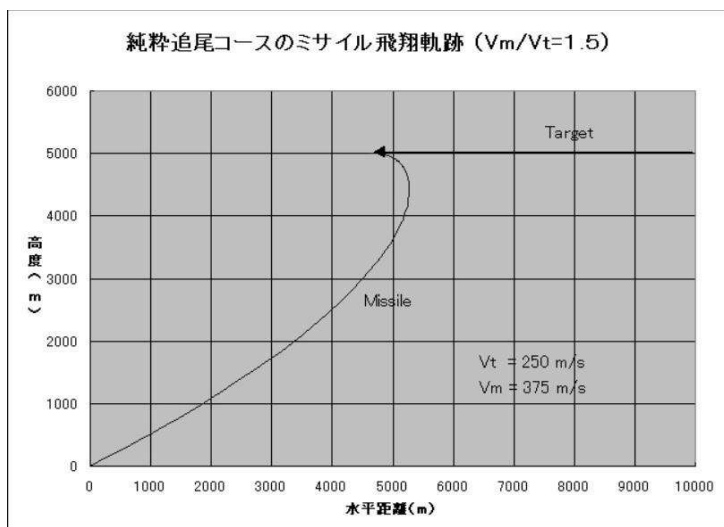
ホーミング誘導は動いている目標に使われますが、動いているということは目標までのルートが変化するという点です。

では、ホーミング誘導は目標を見つけてどのように追尾しているのでしょうか。

素人と学ぶミサイル

一番単純な方法が純粋追尾航法と言われる方法で、目標がその時々にいる方向に向かって進み続ける方法です。しかし、この方法には大きな問題点があります。

というわけで、はい。



このように純粋追尾航法では命中前に大きく旋回する必要があるのです。

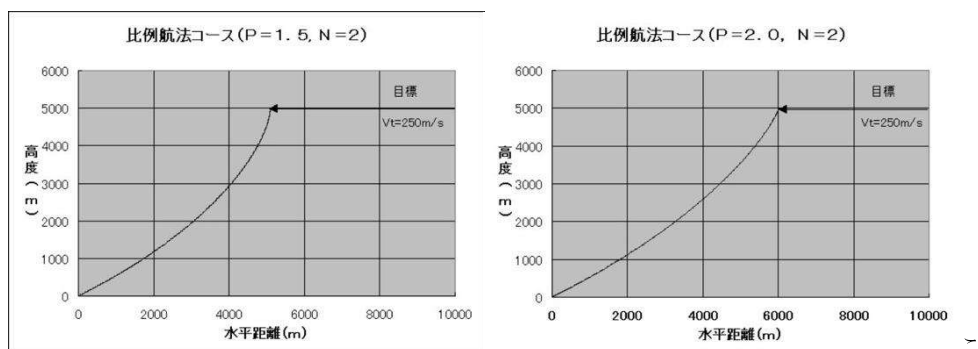
そこで比例航法が使われます

で、比例航法とは何かという話なのですが、多分ですが、ミサイルから見た目標の角速度に比例する角速度でミサイルを旋回させるものだと思います。

なぜ多分なのかというと、私が比例航法をいまいち理解できていないからです。

なので間違っている可能性も高いです。

そんな比例航法ですが、先ほど例に挙げたミサイルが目標の 1.5 倍速と 2 倍速の場合において航法定数といわれる比例定数を 2 にした場合は次のような軌道を描きます。



うすることで直前に急激な旋回も必要なくなり、そのほかにもいろいろな問題が解決するそうです。

2, シーカー

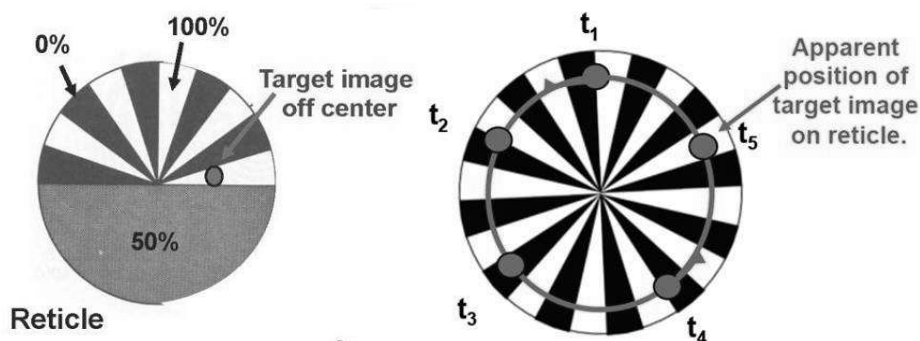
というわけで比例航法をするには角度の情報は大事だと分かったところでシーカーの話しましょう。とはいうものの赤外線画像誘導だとかはまったくもって分からないのでレティクル走査の話になります。ではまずレティクルとは、右図のように赤外線透過部分と透過しない部分が描かれたものの事です。

レティクル走査方式のシーカーはこのレティクルを回すことで目標の方向を調べます。

といわれてもなぜこれで目標の方向が分かるの理解できませんよね。

なのでここからは図を軽く説明する形で解説していきます。

ちなみに、これらレティクルは下図のように位相変調型や周波数変調型などといった種類があります。



まず、レティクル走査にはスピンスキャンとコニカルスキャンの 2 種類があります。まずはスピンスキャンの解説からです。

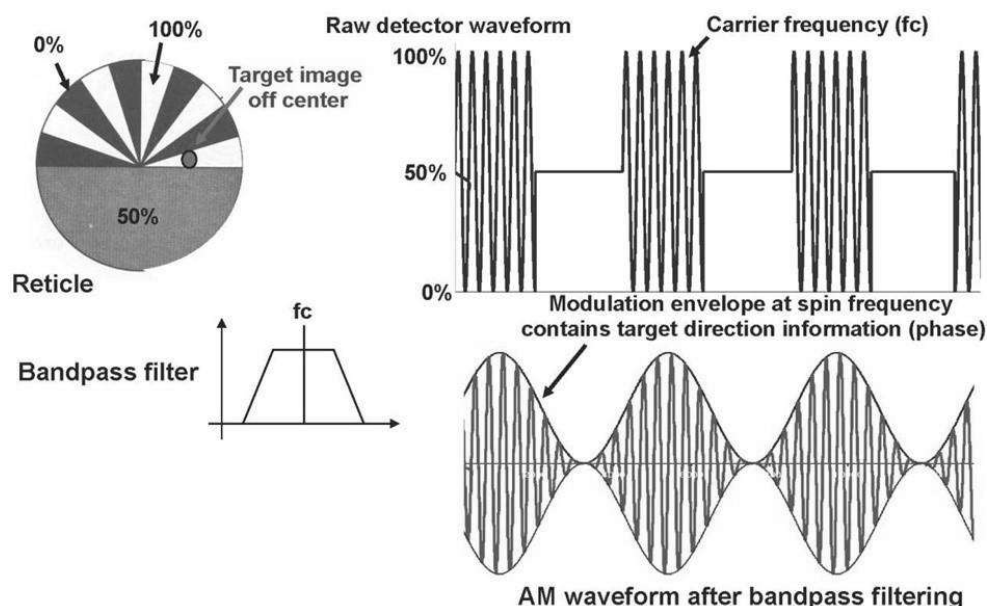
素人と学ぶミサイル

スピンスキャンで使われるレティクルは位相変調型と呼ばれます。

このレティクルでは赤外線は透過する部分と透過しない部分を交互に通過した後ある程度透過する部分を通ることで右上の図のような波形となって検出されます。

そしてその波形を AM で変調すると右下の図のような波形となって得られます。

その結果、この波形の頂点(?)のところが一致するので目標のある方向が分かるという仕組みです。



そしてコニカルスキャンについてです。

こちらは次のページの図の上段のような構造になっているシーカーの小さいほうの鏡が動きます。

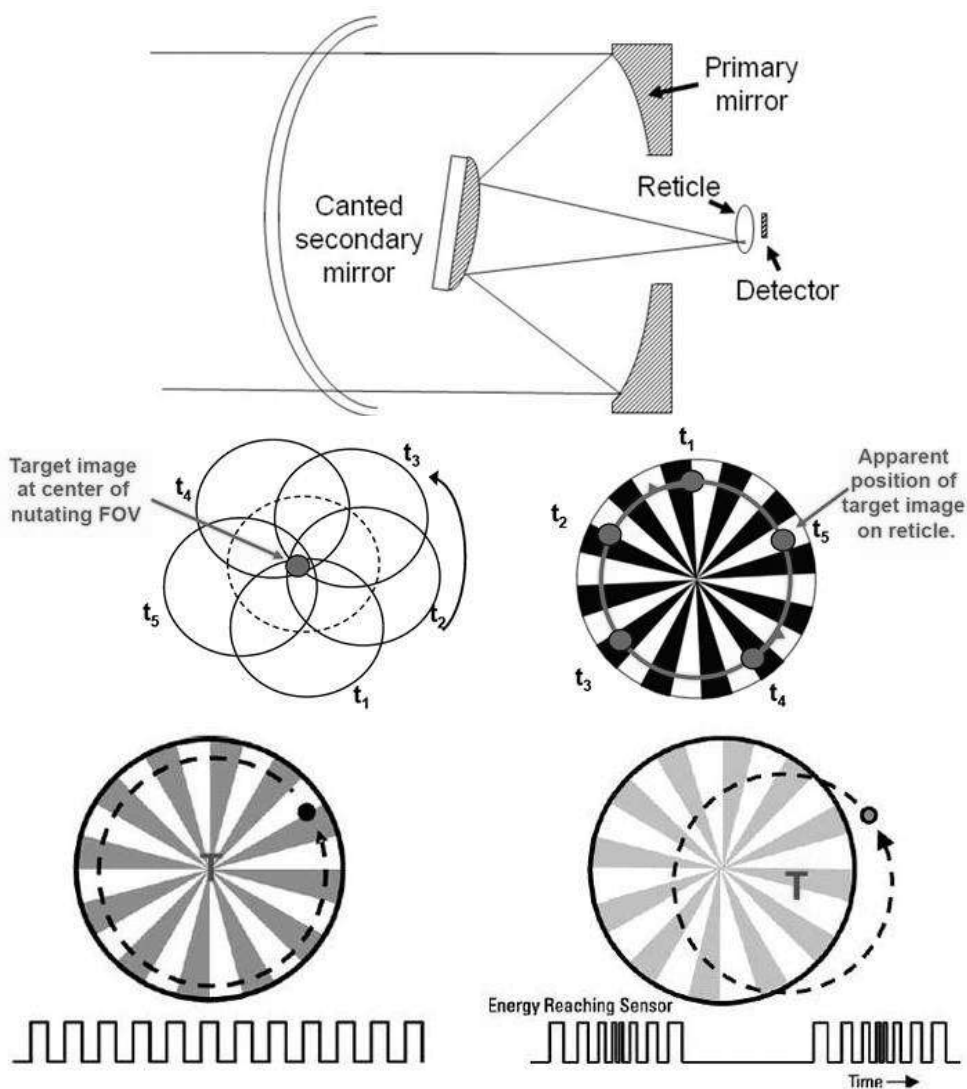
その結果シーカー正面からシーカーに向かってくる赤外線などを次ページ中段の左図のようにセンサーにむけてぐるりと回しながらあてるという構造になっています。

この構造では、目標が中央に近ければレティクルによる赤外線量の増減の間隔が近くなり、目標が離れていれば間隔のばらつきが大きくなります。

これを処理することで目標がどの方向にいるかが分かるという仕組みになっています。

これだけを聞くと処理も構造も複雑になっているのに何の意味も感じられないことでしょう。

しかし、後者が前者にとってかわったほどには後者は優れているのです。

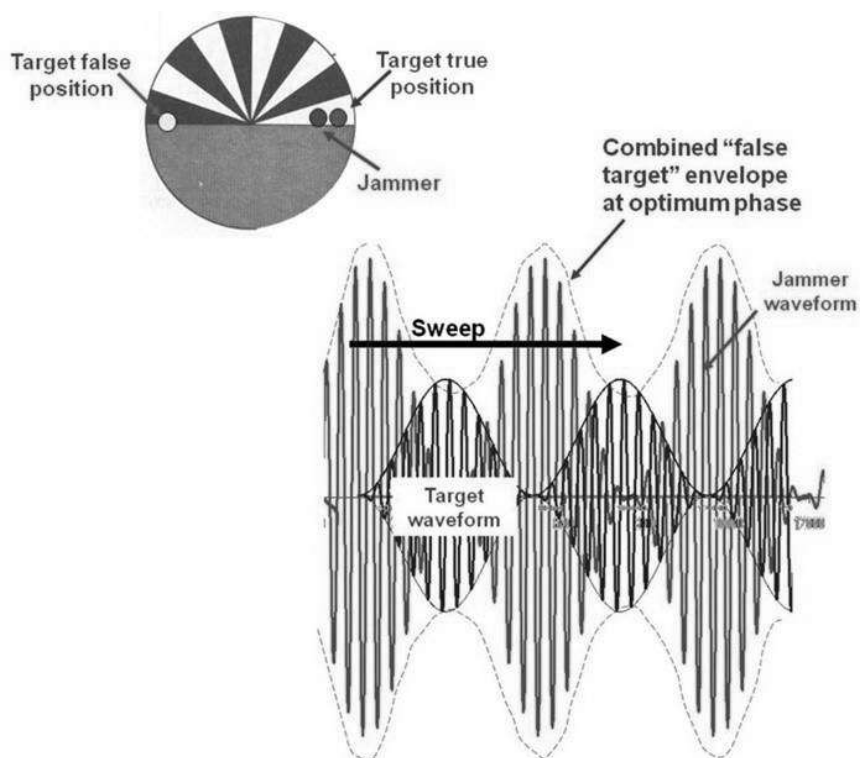


スピンスキャンにはジャマーという赤外線誘導ミサイルを妨害するために大きく赤外線を発する装置が弱点です。

この装置にタイミングをずらして赤外線を送られると次のページの図のように本来の目標が発する赤外線がノイズと同じ扱いになり、目標があると勘違いしてミサイルが見当違いの方向に進んで目標を見失ってしまいます。

それに対してコンカルスキャンでは、確かに中心付近にいるときには妨害されて少し狙いをずらされたりするものの、スピンスキャンと違い大きく外れたときにはジャマー自体が効果がなく狙いを外しきって見当違いの方向へ行かせることは難しくなっているのです。

素人と学ぶミサイル



という感じであまり閉まらない終わり方になってしまいました。

期限自体過ぎてしまっているのもこれ以上は無理ですのでここで切り上げさせていただきます。

読んでいただきありがとうございました。

【参考文献】

『比例航法の基本原理』

<http://kubota01.my.coocan.jp/navigation.html>

『History of the Electro-Optical Guided Missile』

<https://ia801900.us.archive.org/26/items/history-of-the-electro-optical-guided-missiles/Histoy%20of%20the%20Electro-Optical%20Guided%20Missiles.pdf>

『[VOICEROID 解説] 赤外線追尾ミサイルのキホン』

<https://www.nicovideo.jp/watch/sm38908066>

『レティクルスキャン式のシーカーをジャマーで妨害してみました』

https://x.com/sa_tsuklog/status/1513073433684643841

こんにちは、よつばのくろおばあです。アマ研で半幽霊部員やってます。

書くネタがないので部誌書くのやめようと思ったんですが、書かなかったら完全無欠の幽霊部員になりそうなので書きます。という経緯なので、この記事は面白くないです。先に言っておきます。

さて、記事の中身に入っていくんですが、このあいだふと思ったんですよ。

『機械学習、自分でやってみたくね』

ということでこの記事のテーマは「機械学習プログラム自作してみた」です。機械学習っぽいムーブをするプログラムを **Scratch** で組んでみたいと思います。なぜ **Scratch** を使うかというとプログラムを作るのが簡単だからです。決してプログラミング言語をまともに勉強したことがなくて **Scratch** 以外の選択肢がないからではありません。

じゃあ早速やっていきましょう！

今回の機械学習プログラムですが、何を学習させるかで難易度が 9 割決まると思うんですよ。できるだけ簡単なプログラムで終わるのがいいんですが…ということでネタに走ります！テーマは「適切なわんこそばの提供数の決定」にします。理想とするわんこそばの数を決めておいて、その数に合うようにわんこそばを提供することが目標です。理想とするわんこそばの数を「理想数」、提供するわんこそばの数を「提供数」として、15 分でプログラムを組んでみると添付画像のようになりました。一応やっていることとしては、「理想数」と「提供数」の差を 100 から引いた値を「満足度」とし、満足度が低かったらその試行を切り捨てるということを繰り返しています。提供数は 100 から「満足度」を引いた値「絞り」の幅の範囲内で変えるようにしています。ぶっちゃけ機械学習とはかけ離れている感が否めませんが、機械学習は「試行」に対して「評価」（このプログラムでは「満足度」）を繰り返すことで正解に近づくプロセスなので、まあ大きくは外れてないと信じています。ようし、執筆終わり。



ブラウン管について調べてみました。間違っていたらすみません。

1.ブラウン管とは？

今多い液晶よりも前にテレビやパソコンの画面に映像を映し出すのにはブラウン管が使われていました。

真空管を使った装置で昔の家庭や職場などで広く利用されていました。

液晶が使われている製品が板のような形状なのに対し、ブラウン管が使われている製品は箱のような形状です、上に物を置いていることなどもありました。

2.ブラウン管の仕組み

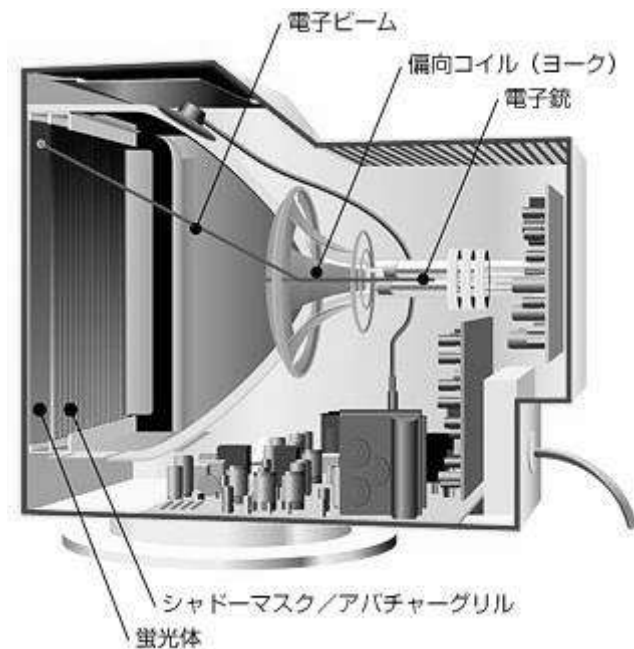
奥に取り付けられている真空管内で電子銃から電子ビームを飛ばし、それを偏向コイルという電磁コイルで曲げて画面を照らします。

画面には赤、緑、青の小さな異なる色蛍光体があり、これら1つずつの集まり1ドットにそれぞれの蛍光体を照らすための電子銃が一本ずつ合計3本ありそれぞれを電子ビームで照らして色、また電子ビームの強さを換え輝度を変えて表現していて、それを左から右に一行走査線と呼ばれる線を描き下の段にいき一行ずつ描くことを高速で繰り返し行うことで映像をみせています。

そのため少しギラギラしているようにみえます。

この時、電子ビームが蛍光体を照らす前にはシャドーマスクというものがあり他のドットを誤って照らさないようにしています。

ブラウン管の仕組み



液晶テレビは映像の加工などで処理を行っているためブラウン管の方が処理が速いです。

これで終わります。読んでいただきありがとうございました。

【参考文献】

・ Wikipedia ブラウン管

<https://ja.wikipedia.org/wiki/%E3%83%96%E3%83%A9%E3%82%A6%E3%83%B3%E7%AE%A1>

・ TDK Corporation 第 1 回 ブラウン管の仕組みとテレビの進化の歴史

<https://www.tdk.com/ja/tech-mag/device/001>

普段使っている携帯電話はどのように通信しているのか

普段使っている携帯電話はどのように通信しているのか

こすみ

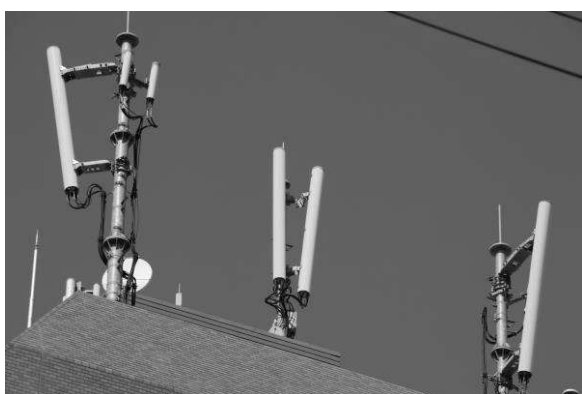
皆さんが普段使っている携帯電話と通信をしている基地局についてご紹介したいと思います。

執筆期限を大幅に過ぎていたため、適当な記事になっていることをお詫びします。

ひとまず 5G(第 5 世代移動通信システム)などの規格について説明したいところですが、検索すればいくらでも出てきますので、ここでは取り扱いません。総務省の情報通信白書とか各携帯キャリアの HP にいくらでも載っています。この記事は実質、基地局(アンテナ)の写真集です。

①セクタアンテナ

・ビル屋上タイプ



上を見上げて歩いてみればすぐ、このようなアンテナが見つかると思います。

一般的な 4G や 5G を提供している基地局のものです。

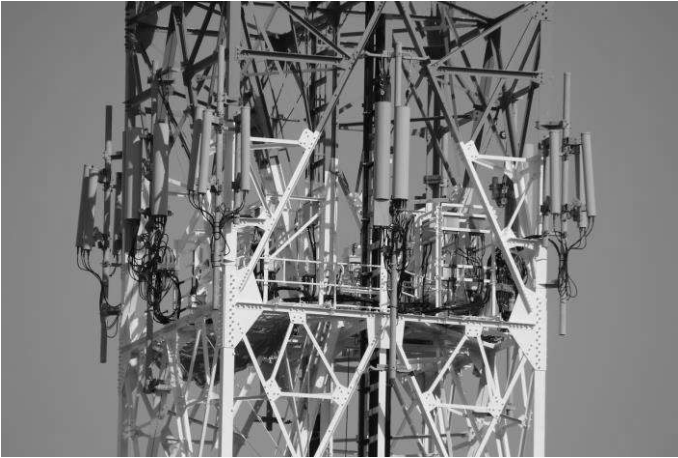


一般的にはこのように、エリアを三分割して、それぞれのエリアに対して、電波を出しています。

都心部では、通信に余裕を持たせるために、六分割されている場合もあるそうです。

普段使っている携帯電話はどのように通信しているのか

・鉄塔タイプ



各鉄塔により、事業者や目的は違いますが、上記写真はNTTコミュニケーションズ浪速ビルについていました。~~日本橋のオタク向けに電波を出しているのでしょう。~~

また、一部の鉄塔には、災害時に広範囲で通信できるようにするためのものもあります。

NTT の局舎って、災害にめっちゃ強いんですね。非常用電源とかもちろんあります。

②無線機一体型基地局

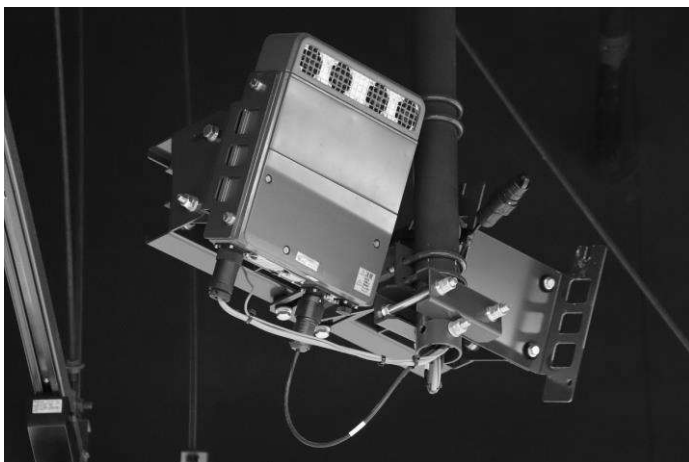


これはドコモのものです。28GHz 帯の電波で通信します。なお iPhone の現行機種で 28GHz 帯に対応しているものはありません。~~さっさと対応しろ。~~

普段使っている携帯電話はどのように通信しているのか



これはソフトバンクのもですね。右にちょこっとついているのは、GPS 用のアンテナです。
なぜ GPS 用のアンテナがついているのかと言うと、時刻を正確に取得するためです。
簡単に言うと、時刻を正確に取得することで他の基地局と正しく連携できるようになり、移動したときに通信が途切れないようにする為などに使われています。これまた詳しい規格は難しいので触れません。
~~僕も実はよくわかってない部分がある。~~



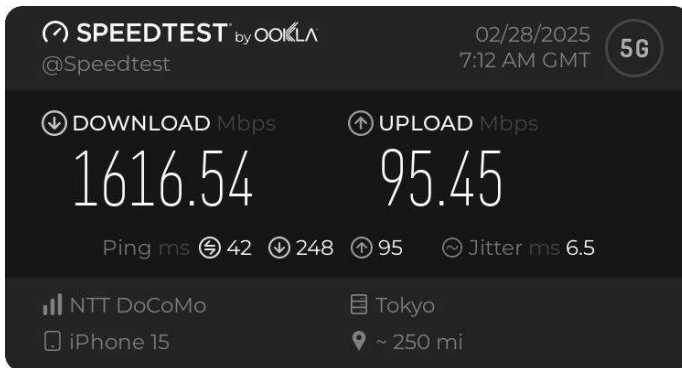
これもドコモのもの(28GHz)です。家電量販店の携帯売り場を狙っています。

ー本章の解説

人が多く集まる商業施設やスタジアムによく設置されています。主に Sub6 やミリ波と呼ばれる高周波を利用しています。この下でスピードテストをすると結構速度が出るんですね。

下の画像はヨドバシ梅田の地下一階にある docomo の基地局付近で計測した結果です。

普段使っている携帯電話はどのように通信しているのか



すごい速いですね。ダウンロードが1616Mbpsというのは1.6Gbpsで一般的に提供されている1Gbpsの回線よりも速いですね。(一部地域で提供されている10Gbps,20Gbps回線は除く)

実はヨドバシ梅田の地下一階にあるdocomo基地局は一般的な5G基地局がインターネットに接続する方法と少し違う方式でインターネットに接続しているため、結構速いです。詳しい仕様は5GSAで検索すると出てきます。5GSAは別途契約が必要です。また対応機種も限られます。

速すぎて、スピードテストを一回すると2GBぐらいデータ量を消費します。月の契約データ量が少ない方はご注意ください。~~ちなみに僕は、月の契約データ量の半分以上をスピードテストで消費しています。~~

③番外編

基地局はもちろん、全然関係ないものをとりあげます。

•ビックカメラの電子棚札用アクセスポイント



ビックカメラの値札は、電子ペーパーを用いた電子棚札です。このアクセスポイントはその値札の制御用です。店舗の天井で見ることができます。

普段使っている携帯電話はどのように通信しているのか

- ・アドオンセルと呼ばれる小型基地局



繁華街などで見られる、小規模なエリアをカバーするアンテナ(基地局)です。

- ・楽天の基地局



商店街の天井についでるやつです。

こういうタイプの基地局があるおかげで、人ごみの中でも通信ができます。

普段使っている携帯電話はどのように通信しているのか



なんばにある楽天モバイルのデカイ看板の上にあります。

④最後に

ここでは基地局部分のみを取り上げましたが、移動体通信は、移動しても通信が途切れないようにするための技術(ハンドオーバー)など、様々な技術が他にもあります。面白いので、ぜひ調べてみてください。

また、離島などに行くと、スターリンクを用いた基地局など、面白いものを見ることができるので、ぜひ行きたいです。

X(旧 Twitter)をやっています。@7834896534_です。フォローしてください。



ある日、私はある存在と青天の霹靂とでも言うべき出会いを果たした。それが、バーチャルユーチューバー「剣持刀也」氏だ。



剣持刀也氏。画質が悪いのは半分ほど意図的。

剣持氏の特徴的な点として、まず配信内においてはあらゆることに対しとても早い反応速度を持ちながら、しかし反応の内容の面白さを損なうことなく、エンターテイナーとしてとても秀逸な振る舞いを見せていたり、しかしゲームが下手であったり機械類に疎かったりなどの意外性をも持っていたり、遅刻魔と言われていたり、ロリコンであると言われていたりすることが挙げられる。

では、私は剣持氏をどのように知ったのか、それは彼の「ロリコン・エピソード」である。これはいくつもの実例があり、ここでは紹介しきれないため、本題である「概念のロリ化」について詳しく見ていく。

剣持氏は「概念のロリ化」によって、精神の平穏を保っていると語る。要は、数多の苦痛を伴う概念を、彼のもっとも好むロリに見立てることで安寧を乞うのだ。なるほど、概念を既知の範囲に窄めることで対処を易しくするのは言われてみれば合理的であるし、普通の人間には一見おかしく見える「ロリ化」だが、一般化すれば同じようなことを実践している方も多いのではないだろうか。

しかし、「概念のロリ化」について、剣持氏が提唱する「ロリは弱く、弱い存在に対し強い存在は余裕ゆえに許しが生まれるため、ロリに対しては許しが生まれる」という論。理論的には間違っていないが、同時に注意すべきは（剣持氏が配信内でも名言していたが）「見下す」可能性が生まれるところにある。そもそも、「許し」であったり「かわいさ」であったりは余裕がなければ生まれない感情であり、ともすればそれは見下しているがゆえの同情になることもある。

概念の哲学——剣持刀也氏に学ぶ「概念のロリ化」

我々はロリに対して、ある種崇拜に似た感情を持つことがある。しかしその上で、我々はロリに対し「かわいい」という感情をも抱いている。であれば、我々は崇拜にも似た感情を持ちつつ、同時にロリを見下しているということになる。この矛盾は見逃せないものになる。では、我々は一体どのような感情をロリに抱いていると言うべきだろうか？

これについて、私は「人がかわいいという感情を抱くときの原因とはなにか」について考えることにした。無論、人が持つ精神的な余裕が関与していることは否定できないどころか筋の通った理論ではあるが、これだけでは説明しきれない部分も存在するのだ。

#ロリ

554,377 作品

ロリとはロータの省略形、主として少女ないし幼女の意味で使われる。ピクシブ百科事典で見る

某サイトにて「ロリ」で検索したときの合致した件数。こんなに愛されているのに見下しているなんて到底思えない。これは主観。

ここで、初めに「かわいさとは何か」について考える。

まず、剣持氏もやっていたように、かわいいものとそうでないものを比較してみる。ここで注意したいのは、かけ離れたものを比較しても、共通する部分が少なすぎるため意味をなさないということだ。剣持氏は「鋼鉄と綿あめ、どちらが『かわいい』か、多くの人は綿あめだと答えるでしょう」と述べていた。私もそれに同意するが、しかし鋼鉄と綿あめでは比較がしにくいのだ。

では、実際に比較をしてみる。例えば「人」というものに対して焦点を当ててみると、同じ人でも年齢によって人が感じる「かわいさ」は変わってくる。題材でもある話だが、幼い子供と成人した大人、老人を比較したとき、幼い子供の方がかわいいと答える人が多くなるだろう。

では人ではない動物ならばどうだろうか？ 動物園や水族館のような施設があることからわかるように、——無論、大型の肉食動物などについては、前述のように精神的余裕が保証されている前提になるが——様々な動物に対し、違わぬかわいさを多くの人が感じていることがわかる。ところが、「猿」という言葉は時に罵倒に使われるではないか。この違いは何ゆえなのだろうか。

ここで、私は「かわいさの基準」に、「無垢であること」や「無邪気であること」が含まれているように思う。猿の例で書くならば、ほかの動物と比べ、猿は比較的に近い知能を持つということが周知されている環境下では、人は「この猿はもしかすると、ほかの動物と違い私に悪意を持っているかもしれない」などと考えるだろう。すると、人は猿の無垢さ、無邪気さを信じられなくなる。そして、感じていた「かわいさ」が薄れ、次第に嫌悪感に代わり、挙句罵倒に使われるまでになったのではないか。

ここで、人の例に戻り、無垢さや無邪気さの観点で各年齢層を見つめてみる。すると、

概念の哲学——剣持刀也氏に学ぶ「概念のロリ化」

なるほど幼い子供の方が純粹無垢に近いとわかる。人は大人になるにつれ、次第に自他の違いを知り、他人を疑うことや自分を繕うことを学ぶからだ。その点、幼い子供は得てして繕うことを知らないものだし、話す内容はほとんどが本意であったり裏のない言葉だ。

私はここに、ロリというもののかわいらしさや、ロリという概念の美しさがあると思う。ショタであってはならないのかなどのは冗長になるため割愛させていただくが、しかしロリというものの美しさはやはりこの「無垢さ」「無邪気さ」にあると思う。

では、「概念のロリ化」とは何か？ ロリという概念についての大雑把な解釈は以上によりある程度まとまったが、「概念」を「ロリ」にするとは？ そもそも、概念を変えるとは何か、概念とは何か？

概念について、各所では以下のように定義されている。

Wikipedia では、「人が認知した事象に対して、抽象化・ 普遍化し、思考の基礎となる基本的な形態となるように、思考作用によって意味づけられたもの」と定義されている。

コトバンクでは、「物事の概括的な意味内容」や「事物の本質をとらえる思考の形式。個々に共通な特徴が抽象によって抽出され、それ以外の性質は捨象されて構成される」と定義されている。

学校から購入した国語辞典、三省堂が発行する「現代新国語辞典」第六版では、「同類のものの中から共通の性質を抜き出すことによってとらえられる、一般的な意味内容」（一部改）や「複雑なものごとについてのおおまかな内容」と定義されている。

偶然家にあった国語辞典、岩波書店が発行する「広辞苑」第三版では、「事物の本質をとらえる思考の形式」と定義されている。

以上から、概念とは「人が事象をとらえる際に普遍化すること、あるいはその思考」だと考えることができる。

このことから、「概念のロリ化」という概念は、「人が事象をとらえるときに、その事象をロリに置き換えながら普遍化すること」と言えるだろう。剣持氏はこれにより、精神の安定を図っているのだ。

以上のことを踏まえると、「概念のロリ化」とは、次のようなステップを踏むことで、あらゆる事象に対し許しを生むことが可能になっていると推察できる。

- ・ 事象を認識する
- ・ 事象をとらえる際、苦痛を伴うと判断した場合、それが「ロリである」と強く暗示する
- ・ とらえた事象がロリであると思い込める
- ・ 苦痛であれど、それがロリの無邪気さゆえの行動だと思うことで、寛容になれる

では実際に、様々な事象について、ロリに置き換えて概念をとらえてみる。

例えば私がこの部誌を執筆している最中話題に上がっているのは、花粉症。SNS では例年よりもひどい気がするなどのような投稿が散見されているが、これが意識を持たない単なる杉や檜や榛の木ではなく、意識を持つものの悪意のない無垢な少女が行っていると仮定してみる。

概念の哲学——剣持刀也氏に学ぶ「概念のロリ化」

するとどうだろう、意識を持たない相手だったならば、悪意があると断定してしまい、「あいつ（杉など）ら、明確に悪意を持って私を攻撃しているだろう！ ああ花粉が鼻に入り込んでくる！ アルミホイル製マスクをつけなければ！！」となると、相手が幼女だと思うと、「他の子（杉など）に喜んでほしくて花粉を飛ばしているだけで、人間が勝手に苦しんでいるだけなら、ロリのほうを優先すべきだな」と冷静な判断を下すことができるのだ。他にも、例えば謎の腰痛がして寝れなかったなどであれば、「これはロリをおぶったから腰を痛めただけであり、特に苦しいわけではない。記憶がないのは幸せすぎて覚えていないだけだ」と思い込むことで苦痛がないどころか幸福に包まれることになる。それだけ自己暗示というものが強力であり、概念というものが人にとって強烈な刺激であるとわかる。

このように、「概念のロリ化」をすることであらゆる事象に対し許しを生みつつ、ロリへの崇拝の感情を保持することができるとわかった。また、一般的な概念のすり替えと違う点として、ほぼ非実在の存在ゆえにすり替えが容易かったり、インターネットにおいては容易な成分の摂取が可能な点などがあることから、比較的優れたストレスへの対処法であると考えられる。

今回は剣持氏の論からでは読み取れないところまで推察したもの、おおむね間違いのない内容にできたと思う。以前までの私は、常日ごろ概念のロリ化を行っているわけではなく、むしろ苦痛を受け止めることで耐性を得ようとしていたが、そのようではむしろ逃げ場を失くし心が荒み、最終的には病んでしまうだけになる。だからこそ、逃げ場としてロリに縋ってみたり、あるいは己の手でロリ以外の概念のすり替え先を模索してみたりすることで精神の平穏を保とうとするのもよいのではないだろうか。

オンラインでポーカーをしよう

オンラインでポーカーをしよう

K. T.

スプレッドシートでポーカーを遊んでみました。

①きっかけ

よく部室でポーカーをして遊んでいたのですが、春休みに入りリアルで集まれなくなったためオンラインでポーカーをしたいと思います。しかし、既存のサービスは自由度が低かったり、ルールが違ったりしたため、自分で作ってみることにしました。スプレッドシートなのは共有が便利そうだったからです。

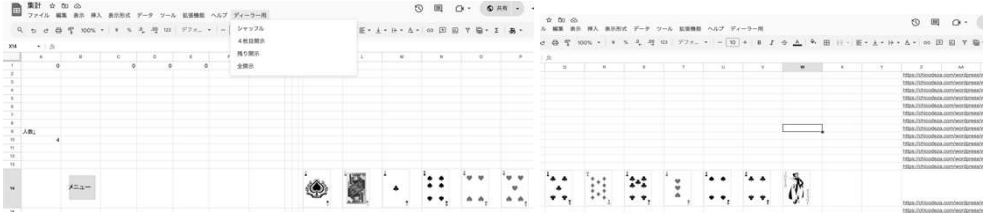
②ポーカーのルール

まず、うちの部活でのポーカーのハウスルールについて説明します。アマ研では、ポーカーの詳しいルールを知っている人が誰もいないので正式なルールとはかけ離れていると思います。がご了承ください。まず参加者に2枚ずつトランプを配り、場に5枚出します。この時、場の5枚のうち3枚は開示しておき、残り2枚を伏せておきます。その状態で参加者は参加費 20チップを支払い、レイズかチェックかを選びます。レイズした場合は他の参加者はコールかフォールドかレイズかを選びます。そしてその後、場にあるカードの4枚目を開け、同じ手順を繰り返します。その後5枚目を開示し、場の5枚と手札の2枚から5枚選んで強い役を作れた人の勝利です。

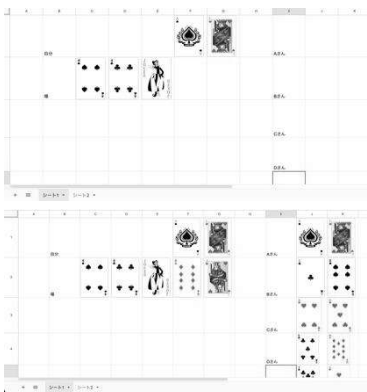
③仕組み

今回はスプレッドシートの関数だけでは実装が困難だったため、GAS(Google Apps Script)というスプレッドシートの処理を自動化できるツールを使用しました。僕は表計算ソフトやGASを初めて使ったので回りくどいやり方をしているかもしれませんが、温かい目で見てください。さて、今回使った関数はSUM、IMAGE、IMPORTRANGEの3つです。名前からわかるかと思いますが一応解説します。SUMは指定したセルの数字を足す関数で、チップの管理をするために使いました。IMAGEは指定したリンクの画像を表示する関数で、トランプの画像を表示するために使いました。IMPORTRANGEは別のスプレッドシート上のセルにある数字やIMAGE関数で表示された画像を表示する関数で、参加者のトランプやチップを共有するために使いました。トランプを表示するのにIMAGE関数を使ったのは、これで表示できるようにするためでもあります。今回は最大5人まで参加できるようにしました。用意したのはディーラー用のスプレッドシートとA~Eさんまでの5人分の個人用スプレッドシートです。次ページの画像はディーラー用のスプレッドシートになります。

オンラインでポーカーをしよう



まず、メニューと書いてあるボタンを押すことで、上にディーラー用というメニューが生成されます。このメニューの中の「シャッフル」を押すことで上のような画面になります。1～54までの数をシャッフルし、その中から 15 個の数字を選び、その数字を元にして、2 列にトランプの画像リンクを生成しています。そしてそのリンクを用いて IMAGE 関数でトランプを表示する、という仕組みです。左から 10 枚は 5 人の手札で、最後の 3 枚は場に開示してあるカ



ードです。左の画像はこの時の A さんの個人用スプレッドシートです。上の画像に表示しているトランプと対応していることがわかるかと思います。このトランプは、先ほど解説したように IMPORTRANGE 関数を用いてディーラー用のスプレッドシートから取得しています。次に、「4 枚目開示」を押すと、場の 4 枚目のカードが、「残り開示」を押すと、5 枚目と全員分の手札がそれぞれ開示されます。全てのボタンを押した時の A さんの画面が左画像です。全員分の手札が開示され、勝敗を決めることができます。

また、チップの計算はシート2で行うことができます。写真の下の方にあるように、トランプはシート1で、チップはシート2で別々に管理しています。チップの計算の仕組みはあまり面白くないので省略します。

④苦勞したこと

今回は初めて表計算ソフトや GAS を使ったこともあって苦戦しました。こういうふうに作りたいというイメージがあっても、それを実現するための関数が存在するのか分からないなどという問題があったため、ChatGPT を使ってみました。使用した感想としては、補助としては優秀という感じです。こういうのがしたい、と聞くとさまざまな案を生成してくれましたが、Excel の関数が入っていたり、今のバージョンでは動かなかったりしたため、そのままでは役に立ちませんでした。しかし、その情報を元に検索すると、有用な情報を手に入れることができました。また、コードを書いた後のデバッグ作業も楽になったため、これから何か作ろうと思っている方は、生成 AI の助けを借りる(丸投げするのではなく!) のも良いかもしれません。最後に GAS のコードを貼って終わります。ありがとうございました。

オンラインでポーカーをしよう

```
1 function menu() {
2   const ui = SpreadsheetApp.getUi();
3   ui.createMenu('ディーラー用')
4     .addItem('シャッフル', 'myFunction')
5     .addItem('4枚目開示', 'reveal4')
6     .addItem('残り開示', 'reveal5')
7     .addItem('全開示', 'revealall')
8     .addToUi();
9 }
10
11 function myFunction() {
12   var spreadsheet = SpreadsheetApp.getActiveSpreadsheet();
13   // 現在開いているシートを取得
14   var sheet = spreadsheet.getActiveSheet();
15   for (let i = 0; i < 15; i++) {
16     sheet.getRange(14, i + 1).setFormula("");
17     sheet.getRange(16, i + 1).setFormula("");
18   }
19   // 54枚のカードを作成
20   const cards = [...Array(54)].map((_, i) => i + 1);
21   // シャッフル処理
22   for (let k = cards.length - 1; k > 0; k--) {
23     const j = Math.floor(Math.random() * (k + 1));
24     [cards[k], cards[j]] = [cards[j], cards[k]];
25   }
26   // 最初の20枚を表示
27   for (let i = 0; i < 15; i++) {
28     const getVal = cards[i]; // シャッフル済みデッキから順番に取得
29     if(i<13){
30       sheet.getRange(14, i + 1).setFormula('=IMAGE("https://chicocodeza.com/wordpress/wp-content/uploads/torannpu-illust$(getVal).png")');
31       sheet.getRange(i+1,26).setFormula("https://chicocodeza.com/wordpress/wp-content/uploads/torannpu-illust$(getVal).png");
32     }
33     else if(i>=13){
34       sheet.getRange(i+1,26).setFormula("https://chicocodeza.com/wordpress/wp-content/uploads/torannpu-illust$(getVal).png");
35     }
36   }
37 }
38
39 function reveal4() {
40   var spreadsheet = SpreadsheetApp.getActiveSpreadsheet();
41   var sheet = spreadsheet.getActiveSheet();
42   sheet.getRange(14, 24).setFormula('=IMAGE(Z14)');
43 }
44
45 function reveal5() {
46   var spreadsheet = SpreadsheetApp.getActiveSpreadsheet();
47   var sheet = spreadsheet.getActiveSheet();
48   sheet.getRange(14, 25).setFormula('=IMAGE(Z15)');
49   for (let i = 1; i < 14; i++) {
50     sheet.getRange(16,i+10).setFormula('=IMAGE(Z$(i))');
51   }
52 }
53
54 }
```

1. 備えよう。

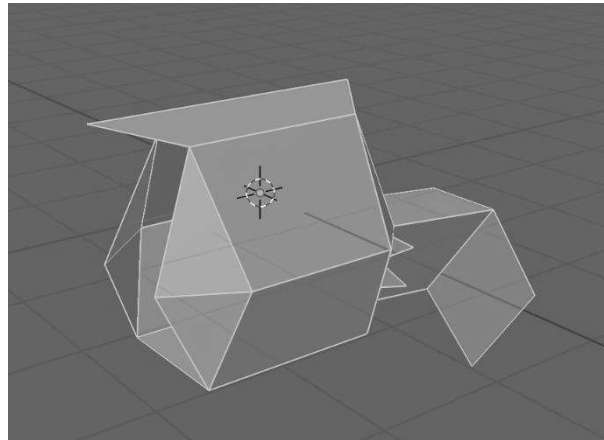
Saluton、こめんつあんとです。皆さん、ペーパークラフトはお好きですか？型紙を切って折って貼るだけで複雑な形が組みあがる、折り紙とはまた違った魅力を持った存在です。しかしもしかすると、与えられた型紙を組み立てるだけでは満足できなくなる時がやってくるかもしれません。その時のために、自力でペーパークラフトを設計する方法を書いていきます。

2. 題材選び編

まずはペーパークラフトにする題材を選びます。筆者自身経験が豊富ではないので、「お手本がたくさんあるとやりやすいので定番のテーマを選ぶべき」説と、「既製品が周りにたくさんあるとやる気に悪影響が出るのでマニアックなテーマを選ぶべき」説が頭の中で対立しています。どっちなんだろうね。好きな題材を選んでください。今回は「1928 年版アノマロカリス旧復元〜トゥソイアの背甲を添えて〜」を作ります。マニアックですね。

3. 3D モデリング編

次に制作したい形を 3D データで作ります。今回は無料 3DCG ソフト「blender」を使用しますが、他のソフト (MAYA や Metasequoia、あるいは Fusion などの CAD) でも構いません。そして作った物がこちらになります。モデリングの解説は多分文章より映像のほうが分かりやすいと思うので、YouTube のチュートリアル動画でも見て勉強すると思います。私もそうだった。

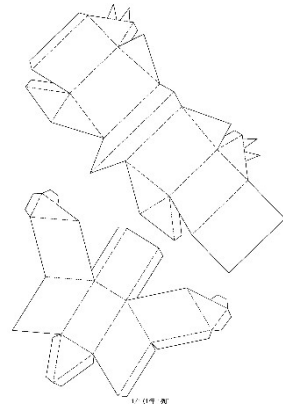


一つ気を付ける点として、あまり形を複雑にしすぎないように気を付けてください。いずれ組み立てるときに地獄を見る可能性があります。自分で組む分にはいいかもしれませんが、今回は生物研究部の部誌で配布する予定なので、サクッと組める程度に留めておきます。

ペーパークラフトを設計しよう

4. 展開編

いよいよ展開図にします。今回はペーパークラフト設計ソフト「ペパクラデザイナー」の無料版を使っていきます。有料版と比べて不自由な無料版ですが、これぐらいの単純なモデルなら扱うことができます。ガチでやりたい人は有料版買ってね。展開したものがこちら。



5. お気持ち編

ソフトの使い方の解説？するつもりはありません。そもそもこの記事はペパクラデザイナーや blender のつかいかた解説ではなく、ペーパークラフトの作り方です。というよりも、「ペーパークラフトって自作できるんだぞ」というのが言いたいただけかもしれません。インターネットって意外とすごくて、これってできるのかな？という問いさえ立ててしまえば検索するだけでやり方が見つかったりします。皆も気になることがあったらすぐに google 先生に頼ろうね！（極論）

Ex1. 生物研究部の部誌

今回題材にした「1928 年版アノマロカリス旧復元〜トゥゾイアの背甲を添えて〜」は皆さんご存じでないと思いますが、せっかくなので生物研究部の部誌として執筆した部分をまるまる持ってきました。アマ研部誌の薄い本化も防げて我ながら妙案ですね。

アノマロカリス旧復元のペーパークラフト

1. はじめに

皆さん、古生物学は好きですか？古生物学は現在生きている生物ではなく、化石などの証拠を基に過去の生物を研究する学問です。化石というとつい博物館にある恐竜の全身骨格のような完全なものを想像してしまいがちですが、実は生物の全身が化石化することはとても珍しく、基本的には生物の一部分、あるいは生物が海底を這った痕など、限られた証拠から生物を復元することになります。そのため最初から生物の正しい姿を復元できるとは限らず、古生物学の歴史には数えきれないほどの誤った復元が存在します。今回はそんな誤った復元の一つのペーパークラフトを配布します。色を塗るなどしてから組み立てて遊んでみて下さい。

2. ペーパークラフトの組み立てについて

山折りと谷折り、のりしろ等は察してください。小学生の皆さんは展開図を頭の中で組み立てるのが得意だと聞いていますが、そうでない人は頑張ってください。のりはスティックのりなどがいいと思います。複数個作りたい場合はスキャンしてデータにしてしまうのも手ですが、素材にしたい紙の上にこの型紙を重ねて頂点の位置をコンパスの針などで刺し、あとで頂点をつなぎ合わせることで簡単に形を写すことができます。

3. なんだこの生物は

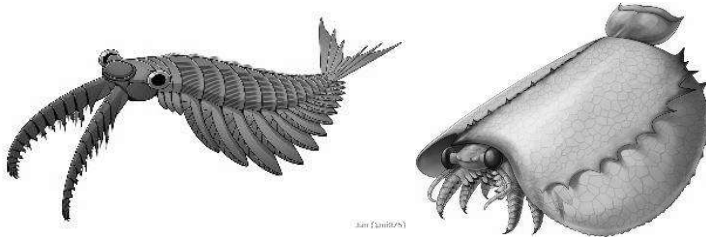
みなさんは一体どれだけ古生物についてご存じでしょうか？三葉虫や今回扱うアノマロカリスについてご存じの方もいれば、恐竜ぐらいしか知らない人もたくさんいると思います。アノマロカリスは今から五億年以上前の海中に生息していた原始的な節足動物の仲間です。地球上に食物連鎖が成立した最初期の頂点捕食者として、少なからずファンが存在する(比較的)人気の古生物です。

ペーパークラフトを設計しよう

4. なんだこの復元は

そんなアノマロカリスですが、最初から現在のような復元がなされていたわけではありません。発見された当初は体がバラバラになった化石しか見つかっていなかったで、それぞれの部品が別の生物として考えられていました。今回のペーパークラフトはそのような状況で提唱された復元の一つです。実際にはアノマロカリスの頭部についている付属肢がトゥゾイアという別の生物の胴体として考えられています。最新の復元ではなくわざわざこんな旧復元を選んだ理由は、単に最新復元は複雑な形をしていてペーパークラフトの製作が間に合わないというのがありますが、「正しくない」とされてしまった学説のことも忘れないでいてほしいと思うからでもあります。確かに学術的にはもはや意味をなさないかもしれませんが、それでも当時の研究者たちが必死で復元した結果であり、正しくないと切り捨てるのは少しかわいそうな気がします。生物の研究だけでなく、生物研究そのものの歴史もなかなか面白いということをぜひ皆さんに感じていただきたいです。

下図は左からアノマロカリスとトゥゾイアの復元画。



5. 画像の出典

図 1

ウィキメディア・コモンズより 最終閲覧日:2025/03/29

https://commons.wikimedia.org/wiki/File:20191203_Anomalocaris_canadensis.png

図 2

同上

https://commons.wikimedia.org/wiki/File:Tuzoia_life_restoration.jpg

ロボレーブ参戦記

こんにちは。アマチュア無線研究部です。

アマ研は名前の通りアマチュア無線をやっていますが、それ以外にも活動内容として電子工作・3Dプリンター・3Dモデリング・プログラミング・ロボット作成などがあります。

アマ研では年に二回ロボットの大会である

R o b o R A V E に(大阪大会と加賀大会)

出場しているのですが今回はその大会について参戦記を書いていこうと思います。ぜひ最後まで読んでいってください。

まずR o b o R A V E とは何ぞやと思う方も多いと思うので、その概要やルールについてお話しします。R o b o R A V E (図1)とは”Robots Are Very Educational”の頭文字で、2001年にアメリカで生まれた教育プログラムです。

コンピューターを使ったロボット動作のプログラミング学習や、操作体験を通して、子どもの科学とものづくりへの興味・関心を高め、創造力や柔軟な思考力を育むことを目的としています。ロボレーブ (RoboRAVE) とは - ロボレーブ世界大会 2024 開催!

この活動は日本にも広まり加賀でも国際大会が開かれています。(大阪大会のロボレーブ(図2)は株式会社ダイセン電子工業さんが開催しており加賀大会とは主催者が違いますがR o b o R A V E の理念は同じです。)

R o b o R A V E 大会にはいくつか競技がありますがアマ研は毎年ラインフォロイングという競技に出ています。

{ラインフォロイングのルール}

- ・黒線をたどってタワー(ゴール)までボールを運び、スタート地点まで戻ることのできるロボットを設計し、組み立て、プログラミングします。

- ・黒線の太さは中学生の部で1.25cm高校の部で0.75cm(図3)

- ・タワーのサイズ タワーは全ての部門で共通(高さ20cm、幅10cm、長さ35cm)で上部に10cm×10cmの開口部を設け、入れられたボールを排出するために後ろ側が空いています。タワーはテープでトラックに固定され、ボール排出部分に仕切りがあります。(図4)

ロボレーブ参戦記

・2つのミッション（ミッション 1：基本ミッション、ミッション 2：追加ミッション）の合計が得点となります。

・ミッション 1（基本ミッション）

ホームを出発しラインをたどり、少なくとも1つのボールをタワーに入れ、再びラインをたどってホームまで戻ってくる。

ミッション1でタワーに入れられたボールは、ロボットがタワーを出発すると取り除かれ、カウントされません。

・ミッション 2（追加ミッション）

ホームを出発しラインをたどり、指定された数のボールをタワーに入れる。ロボットはホームに帰る必要はありません。

【ボールの指定数】

小学生の部：120

中学生の部：200

高校生の部：250

（図5）

制限時間 3分のタイムオーバー、あるいはミッション終了を宣言した時点で競技終了となります。

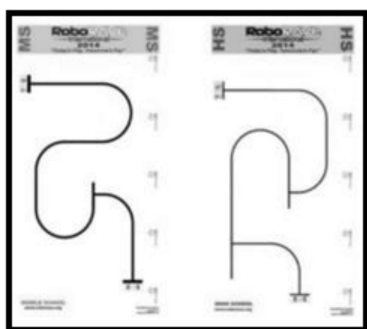
（図1）



（図2）



（図3）



（図4）



(図 5)

	ホーム を 出発	1つ目 の “T”を 曲がる	2つ目 の “T”を 曲がる	タワー で 止まる	ボール を 入れる	ホーム へ 戻り 始める	1つ目 の “T”を 曲がる	2つ目 の “T”を 曲がる	ホーム に 到着	合計
小学生 の部	50	—	—	100	100	50	—	—	100	400
中学生 の部	25	25	—	100	100	25	25	—	100	400
高校生 の部	25	25	25	50	100	25	25	25	100	400
ボールポイント：ミッション2で運んだボールの数が、得点に加算されます。 もし、指定数を超えた数のボールを運んだ場合、超えた数を指定数から引いた数が得点に加算されます。										

先ほどのルール説明でラインをたどる、だとかラインをたどらないだとかのお話があったんですが、ラインをたどるという定義が曖昧、というかしっかり定義づけされておらず、それっぽく進みながらタワーにボールを入れさえすればいいので、最悪センサーなしのモーター2つで何秒すすんで何秒回転して…ってやってもいいんですが、が、が、が~~~~~そんなことする方が面倒くさいので真面目にライントレースする方法についてお教えします。

まず、ラインセンサーとは何かについて。簡単に言えば、その下が黒いか白いかを判定するアート 引越し センサーです。これを機体の右側と左側に設置して右側に線があったら右側に、左側に線があったら左側に曲がることで、線の上を走ることができます。

普通のコースならセンサー2つで進めるんですが、T字路をいい感じのスピードで走りつつ安定性を確保するセンサーが3個要ります。2つでもセンサーの間の距離を

小さくすれば確かにT字路として検知するのは出来るにはできるんですがカーブを曲がるときに直進と回転を繰り返さないといけなくなり、スピードが落ちるのでセンサーを3つ横一列に配置してどれか2つ以上反応すればT字路とみなす、とすれば真ん中2つのセンサーの間にいい感じの距離が取れるのでいい感じのスピードを確保しつつT字路を検知することができます。うん、これは令和最新版。

要するにセンサーが白か黒かで適切に条件分岐すればいいということです。

さて、車輪についてですが、4輪オムニ機体みたいな変わった機体にしない限りは2つでいいです。

その場合機体の後ろ側にはコロコロ（正式名称なんなんでしょうね(ボールキャスターっていうらしいですよ)）をつけて走らせます。

今までの経験から言うと変に凝った機体を作るよりもタイヤ2つで後ろにコロコロをつけたスタンダードな機体の方がうまくいきます。

ただ男の子である以上、ロマン(4輪オムニ機体など)は心惹かれるものなので後ほど語りたいと思います。

さて細かい話は一旦置いておき、おおまかなロボット作成と大会の流れを説明していきます。

ロボレーブ参戦記

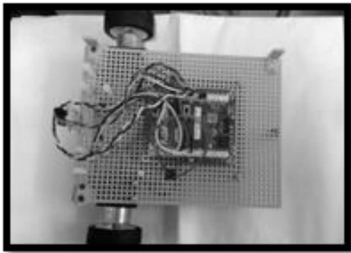
<ロボット作成>

- ①土台にモーター・ラインセンサー・タッチセンサー・※マイコンなどの部品を付ける。
- ②ピンポン玉を入れるための箱を機体に取り付ける。
- ③プログラミングをする。

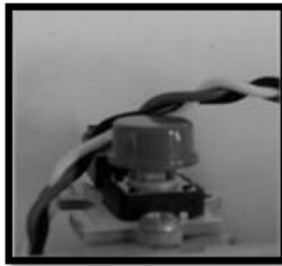
優先順位は

ハードウェア→ソフトウェアとなります。

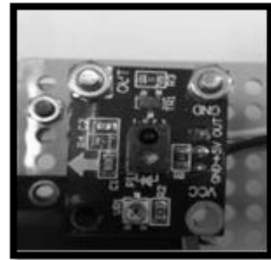
ソフトからやると後から調整しないといけないので大変です。(n敗)



(土台)



(タッチセンサー)



(ラインセンサー)

※マイコン・・・プログラムを入れる小さなコンピューターのこと。

ラインセンサーには閾値(色が黒か白かを識別する境目の値)があり、ラインセンサーについているネジを精密ドライバーで回すことで閾値を変えることができます。ですが、その値を調整する地道な作業があるので忍耐力が必要です。

線の色を間違って読み取ってしまうとプログラムが合っても正しく動きません。

世知辛いですね。この面から考えても作成の優先順位はハードウェア→ソフトウェアだと言えますね。それとモーターのギア比が違っていると上手く移動できないので、(左右で速さが違うので)揃えておかないといけません。

(ロボット走行の様子)

プログラムに関してはタイヤ2つのスタンダードな機体なら特筆することはありません。前述した通りです。

ピンポン玉を入れるカゴにはスライド式の蓋を付けたり、ピンポン玉がタワーに入りやすくなるように箱を傾けたりします。

もう一つ大切なことは、チーム内で明確に役割を分担することです。そこをはっきりさせておかないと作業効率が落ちます。チーム内での役割が被るのも良くないですね。大きなプロジェクトなどならともかくラインフォロイングロボットを作る規模なら一つのことを一人で担当した方がよいです。



〈大会の流れ〉

- ①チーム受付・開会式
- ②練習・ロボットの体積チェック
- ③予選(スコアリング)・交流会
(交流会は加賀大会のみ)
- ④決勝
- ⑤閉会式

ロボレーブの開会式では

「1, 2, 3 Wow!」という掛け声が毎年恒例となっています。

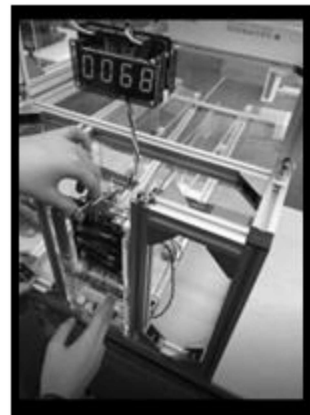
開会式後は練習、予選、決勝という順に進んでいきます。

予選が終わればピンポン玉の獲得個数をカウントして順位を決め、上位から数チームが決勝に進みます。決勝ではピンポン玉の個数はもちろんのことスタートからゴールに到達するまでの時間も考慮されます。

ピンポン玉をカウントするときは右のような機械を使います。交流会は大会側が用意した軽食やジュースが参加者にふるまわれるパーティーのようなものです。外国からの参加者が大半なので異文化交流にもなります。

ビンゴ大会なども開催されたことがありました。競技以外のことも楽しめるのがロボレーブのいいところですね。

ロボレーブ大会についての概要について、一通り説明が終わったところで実際に今年の大阪大会と加賀大会の参戦記を書いていこうと思います。



ロボレーブ参戦記

【2024年大阪大会】

大阪大会についてですがいかんせん写真を部誌用に撮っていなかったので文章だけの紹介になります。

結果としては入賞することができませんでした。中学生チームのうち一つはかなりいいところまで行ったのですが惜しくも・・・という感じでした。

敗因は明確で準備時間不足ですね。理屈上では間違っているところがなくとも不備というものはごまんとあります。

その不備を当日までに洗い出すことができていませんでした。

この記事(【2024年加賀大会】より前)

を書いている時点では加賀大会にはまだ参加していなのでこの反省を活かそうと思います。

【2024年加賀大会】

今年は11月16日から17日にあり、20人で参加しました。(僕は15日の夜中から加賀に前日入りしました。)

大会会場の最寄り駅は加賀温泉駅で住吉駅から団体乗車券を購入してサンダーバードと新幹線を使い継ぎ、一度敦賀駅で乗り換えてから行きました。去年までは大阪からサンダーバード一本で行けたのですが今年から新幹線が開通したためこのようなルートになっています。

敦賀駅には福井県の駅弁が売っていて皆思い思いのものを買っていました。

(蟹弁当・焼き鯖寿司・シュウマイ弁当 etc.)



(敦賀駅で買った崎陽軒のシュウマイ弁当)

加賀温泉駅についた後はタクシーに乗って宿に向かいました。この時点では21時をまわっていたので気が付きませんでした。この宿は山の中にあるため自然豊かな景観が素晴らしかったです。



(早朝に宿の中から撮った写真)



(一階・徹夜部屋)

宿は一棟丸ごと貸切ったので広々と使うことができ、快適でした。

2階建てになっていて、1階では例のごとく徹夜で()作業をする帳尻力の権化たちが群がり、2階は寝室として使っていました。寝室は2段ベッド×7+個室といった様式で物珍しかったです。大会前は内心どこかで、トランプやUNOなどを大人数でワイワイやってやろうと楽しみにしていたのですが、その見通しはかすりもしなかったです。おかしいですね。会場はみやびの宿という宿泊施設でした。みやびと名前に付いているだけあって華々しい会場でした。



(「1, 2, 3 Wow!」の掛け声)



(頂いたNASAのステッカー)

加賀大会には1日目と2日目があり、前述した通り1日目に練習と予選、2日目に予選と本番がありました。

1日目 概要

開会式→練習→予選→交流会

交流会では例年通りのビンゴ大会、太鼓や獅子舞のパフォーマンスがなされました。

今年は来賓にNASAの方がいらっしゃってお話になりました。

加えてNASAのステッカーも頂きました。

ロボレーブ参戦記

2日目 概要

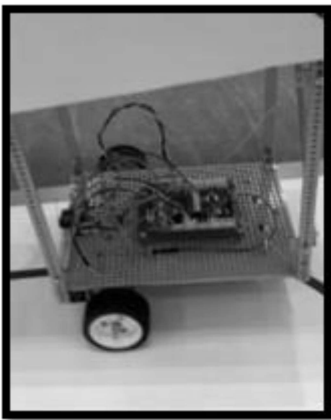
予選→決勝→閉会式

さて加賀大会について大まかな内容は話し終えたので、ここからは出場した各チームにバトンタッチしたいと思います。

チーム名「NAREC J a m」

K.H 君(執筆者) 中一

M.Y 君 S.S 君 Y.Y 君+協力してもらったT.T先輩、ありがとうございました。



機体の概要

機体の大まかな形は前輪2個と後ろにバランス用のキャスターボールを取り付けたかなりノーマルな形です。この形だとラインレースの制御が簡単なのでよく使われます。

ラインセンサーは前部に4個取り付けてあり、内側の2つのセンサーは通常のラインレース用に、外側の二つはT字路の検知用に使っています。

制御の概要

制御方法にはPD制御というものを使っています(興味のある方はYoutubeでPD制御と検索してみてください)。これにより動きがとても滑らかに…なるはずだったのですがゲインが小さすぎたのか普通の制御方法とあまり変わりませんでした。徹夜してプログラムしたのに…結果

NARECの中でトップクラスに無難で最も早く完走できたNAREC J a mの機体でしたがなんと当日の朝になってマイコンとシリアル通信用コードが壊れてしまったのです。

その時はさすがにパニックになってしまいましたね、みんなごめん。

何とか予備のマイコンに切り替えましたが挙動が変わり、結果は惜しくも

1267/3000 点

にとどまりました。

細かい話

・NAREC JamではT字路の検出を「外側左のセンサーが黒を検出したら右に90度回る」といった感じで制御していました。そのため、ロボレースのスポンサーロゴが黒字で印刷されていたのをT字路と認識して曲がってしまったりもして大変でした。

・機体の上部に取り付けた段ボールが殺風景だったので青色で「NAREC」の文字を塗装しておきました。公開するかと思ったけど結構よさげでした。

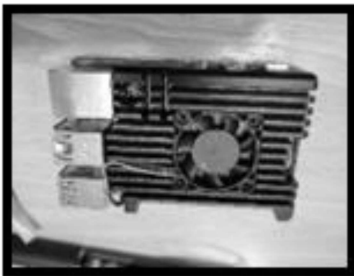
・NAREC Jamの名前の由来は僕が電子工作を始めるきっかけになった子供パソコン Ichigo Jam に由来しています。

プログラミングの初めの一步に最適です。

チーム名「NAREC Chickens」

D.K 君 半魚人君 高一

どうもこんにちはみんなに名前を間違えられるのが最近気になる D.K と蜜柑が食べたい半魚人です。ドンキコング NAREC Chickens という班でそれぞれハード担当とソフト担当をしていました。うちの機体自分で設計しておいてなんですが基本的に殆どの場所がキチワイなので、ご覚悟を。ちなみにキチワイは D.K にとっては褒め言葉です。



マイコン（基盤のこと）は Raspberry Pi 5B 4GB Model（左写真、以下ラズパイ）を使用しています。2万円弱でピンを通じて電子基板を接続可能で、Wi-Fi を飛ばせ、GPU も搭載されると電子工作用の基盤としてなかなか優秀なハガキ程度の大きさのコンピューターです。本来はこれにモニターとキーボード、マウスをつないで内部データの編集などをしますが、加賀にまでそんな大きいもの持って行けないので、ラズパイ本体をモバイルホットスポットとして Wi-Fi を飛ばし、それを通じて VNC（要するにモバイルデスクトップ）で操作しています。

ロボレーブ参戦記

移動方法は DAISEN 製の 6ch モータードライバ（モータを制御するもの）と四輪オムニホイールです。オムニホイールとは、左の写真のようにタイヤ自身に横向きに小さいタイヤがついていてタイヤの本来の進行方向とは垂直な方向に動いた際にも摩擦で止まらず小さいタイヤのおかげで動きを受け流しスムーズに動けるといって代物です。これを前後左右四方向に取り付けることにより、前に進む際は左右のホイールを回転させ前後の小さいホイールが回る、横に進む際は前後のホイールを回転させ左右の小さいホイールが回る、斜めに進めたい場合も前後左右のホイールを回転させながら小さいタイヤが回転を受け流すといったように機体自身を回転させることなく好きな方向に移動できるのが四輪オムニホイールです。最もライントレースでは、回転させないことによって走行が安定するメリットよりも速度が出せる二輪式のほうが強いのですが……まあ、こちらのほうがカッコイイので、ヨシ！

そしてセンサー、この機体の一番気色悪いポイントです。ほとんどの場合というか普通の人が設計すれば固定されているラインセンサーが、なんと、ぐるぐる回ります！ 理論としては、センサーを二つしか取り付けていなくても、ヘリコプターの羽みたいにくるぐる回せば円状のデータがとれていいねって感じです。余談ですが、これを回さずに大量のセンサーを円状に取り付けることにより強引に円状のデータを得ているものをエンジェルリングと言い、これはロボコンなどのハードな大会でも使われています。その円状のデータ処理に関しては後程。オムニホイールは 360° 好きな方向に移動できる為ベクトル制御と相性が良く、進行方向を単に右・左・直進ではなく角度でデータをくれるこのセンサーのつけ方をしました。

最後に、プログラムですがこの機体はモータードライバーが壊れていることに気づかなかったせいで機体が動かず結局実戦で走行できていません。なので、製作者の意図通りに動いたという机上論でお話しします。このプログラムの流れとしては、センサー二個がそれぞれ半回転した際、センサーが毎秒 200 回集計したデータを円上に配置し、センサーが反応した 2 点をつなげ、それに沿って進むというものです。この中で一番難しいのが 2 点を繋げるところで、40 ほどの白か黒かを判別したデータから、黒の連続部位をまとめ、n 度に存在する点にするという処理が、如何せん高速回転するセンサーという仕様上誤差が大きく、飛び地で黒を検出するセンサーの処理に苦戦しました。さらに T 字路処理の際は三点がそれぞれ独立した点であることも検知しなければいけないので、結構複雑なプログラムになりました。実機でこれをテストできたらなあ。

こんな感じです。個人的にはラズパイに適当につけたヒートシンク（冷却装置兼カバー）がかっこよくて好きです。ぶっちゃけた話、一回見ないとどんなに気持ち悪いかわからないと思うので、一回アマチュア無線研究部に来てみてください。まあその頃には分解されて再利用されている気もしますが。

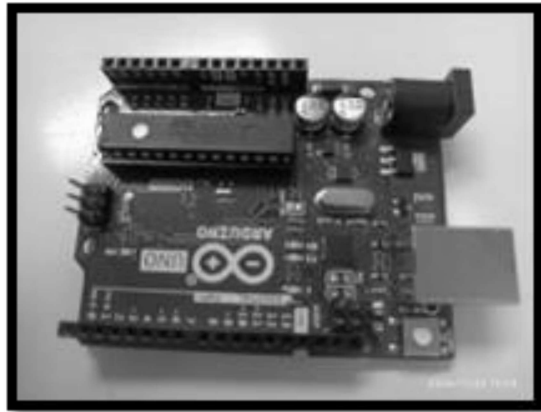
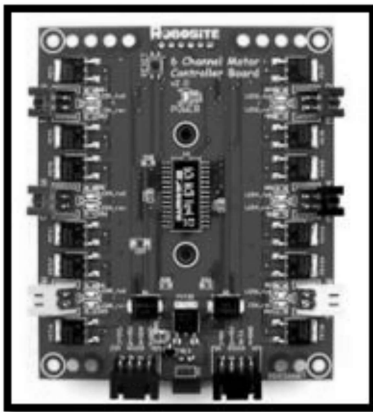
チーム名「NAREC7」

TR君 KY君 TK君 NS君 高一

僕たちの機体にはA r u d i n oというマイコンを使いました。マイコンからタイヤに直接電流を流すとマイコンがそれに耐えられずに壊れてしまうのでモータードライバーという部品を間に挟みました。

(6CHモータードライバー)

(A r u d i n o)

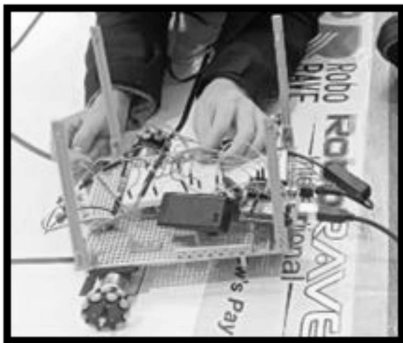


A r u d i n oと6CHモータードライバーを接続するためにI 2 C通信というものを使うのですが、それに苦戦しており、プログラム担当の人たちが夜通し調整してくれました。

ハードウェア

初めは4輪オムニで制作を進めていたのですが、途中から安定性をとって対向2輪に路線変更しました。それに伴いスタンダードな挙動をするようプログラムしました。

(左図 変更後の機体)



ロボレーブ参戦記

[illegible][illegible]

※ I 2 C のプログラム

```

%Nucleus1=1:eol(1000);

Nucleus1_C_1=1;
Nucleus1_C_2=1;
Nucleus1_C_3=1;
Nucleus1_C_4=1;
Nucleus1_C_5=1;
Nucleus1=1:eol(1000);

Nucleus2_C_1=1;
Nucleus2_C_2=1;
Nucleus2_C_3=1;
Nucleus2_C_4=1;
Nucleus2_C_5=1;
Nucleus2=1:eol(1000);

Nucleus3_C_1=1;
Nucleus3_C_2=1;
Nucleus3_C_3=1;
Nucleus3_C_4=1;
Nucleus3_C_5=1;
Nucleus3=1:eol(1000);

Nucleus4_C_1=1;
Nucleus4_C_2=1;
Nucleus4_C_3=1;
Nucleus4_C_4=1;
Nucleus4_C_5=1;
Nucleus4=1:eol(1000);

Nucleus5_C_1=1;
Nucleus5_C_2=1;
Nucleus5_C_3=1;
Nucleus5_C_4=1;
Nucleus5_C_5=1;
Nucleus5=1:eol(1000);

```

```

PlotterView[0]
PlotterView[1]
PlotterView[2]
PlotterView[3]
PlotterView[4]
PlotterView[5]
PlotterView[6]
PlotterView[7]
PlotterView[8]
PlotterView[9]
PlotterView[10]
PlotterView[11]
PlotterView[12]
PlotterView[13]
PlotterView[14]
PlotterView[15]
PlotterView[16]
PlotterView[17]
PlotterView[18]
PlotterView[19]
PlotterView[20]
PlotterView[21]
PlotterView[22]
PlotterView[23]
PlotterView[24]
PlotterView[25]
PlotterView[26]
PlotterView[27]
PlotterView[28]
PlotterView[29]
PlotterView[30]
PlotterView[31]
PlotterView[32]
PlotterView[33]
PlotterView[34]
PlotterView[35]
PlotterView[36]
PlotterView[37]
PlotterView[38]
PlotterView[39]
PlotterView[40]
PlotterView[41]
PlotterView[42]
PlotterView[43]
PlotterView[44]
PlotterView[45]
PlotterView[46]
PlotterView[47]
PlotterView[48]
PlotterView[49]
PlotterView[50]
PlotterView[51]
PlotterView[52]
PlotterView[53]
PlotterView[54]
PlotterView[55]
PlotterView[56]
PlotterView[57]
PlotterView[58]
PlotterView[59]
PlotterView[60]
PlotterView[61]
PlotterView[62]
PlotterView[63]
PlotterView[64]
PlotterView[65]
PlotterView[66]
PlotterView[67]
PlotterView[68]
PlotterView[69]
PlotterView[70]
PlotterView[71]
PlotterView[72]
PlotterView[73]
PlotterView[74]
PlotterView[75]
PlotterView[76]
PlotterView[77]
PlotterView[78]
PlotterView[79]
PlotterView[80]
PlotterView[81]
PlotterView[82]
PlotterView[83]
PlotterView[84]
PlotterView[85]
PlotterView[86]
PlotterView[87]
PlotterView[88]
PlotterView[89]
PlotterView[90]
PlotterView[91]
PlotterView[92]
PlotterView[93]
PlotterView[94]
PlotterView[95]
PlotterView[96]
PlotterView[97]
PlotterView[98]
PlotterView[99]

```

結果は5位でした。

まとめ

加賀大会では大阪大会での失敗を活かして調整できていたチームもいくつかあって良かったと思います。

男は CPU も産めるぞ。俺は CPU のママになるんだ!!!

男は CPU も産めるぞ。俺は CPU のママになるんだ!!! おいしいドーナツ

これを見るだけで、CPU 出産の疑問は解決。必勝!出産マニュアル
突然ですが、ママになりたいと思ったことはありますか?ありますよね?そう、あるんですよ。
てなわけで今回は CPU のママになっていきましょう。

1 そもそも CPU ってなんだ?

皆さんは CPU とかいうマイナーで専門的な(大嘘)部品は知らないと思うので簡単に CPU とは何かを簡単にお教えます。CPU とはコンピューターの頭脳的な something です。興味がある人は調べてみてね!

2 CPU の自作方針

今から CPU の自作方針を話すと約束したな。あれは嘘だ。ウァ~~~~!!
さすがに適当すぎるので CPU についてもうちょい話します。優しい。さて、CPU とは center processing unit (中央演算装置)のことで、コンピューターの重要な演算をしたり他の部品に指令を出したりする部品です。右のやつはそこらへんで見つけた旧世代 Xeon。近年は intel core とか intel Xeon とか AMD Ryzen とか AMD EPYC とか snapdragon とかとか色んなシリーズの CPU があります。これらの CPU の弱点として、使える回路の長さが短いというのが挙げられます。例えば intel core i9-14900KF は最大周波数が 6GHz(intel 公式より、オーバークロックががんばったら 9GHz でるらしい。やばいね。)となっており、要は最速で 1s/6G に一回クロックを作動させることが可能ということです。光の速さで情報を伝達できたとしても、一クロックあたりに使える回路の長さはわずか 5cm(300,000km/6G)です。＼ブッパー／コレハ、イマセソネ。マイクラだとクロック回路を作るのに 3 マスは必要だというのに...そこで私は考えました。CPU のクロック周波数を下げればいいのではないかと。マイクラのクロック回路にも人権はあるんです。てな訳で、今回作っていく CPU のスペックはこちらになります。



汎用レジスタ 4bit×2

アドレス空間 4bit(16B)

プログラムカウンタ 4bit

フラグレジスタ 1bit

動作クロック 3MHz 程度

トランジスタ数 約 1500(正確な個数は不明)

あゝ あゝ コゝ ミゝ カゝ スゝ ウゝ ウゝ ウゝ ! ○ゝ ねゝ えゝ えゝ えゝ えゝ ! (音割れ)

男は CPU も産めるぞ。俺は CPU のママになるんだ!!!

一応さっき話した intel core i9- 14900KF のスペック(ここら辺は正直気にする必要がないので公式サイトに載っていませんでした。多分こんなもんやろっていう推測に過ぎないので、必ずしも正しいとは限りませんが、比較にならないということは理解いただけると思います。)上げときます。はいドカン！

汎用レジスタ 64bit×16
アドレス空間 64bit(最大 16EB(16×2^{60}))
プログラムカウンタ 64bit
フラグレジスタ 16bit?
動作クロック 6GHz
トランジスタ数 2G~3G

やばい。話にならない。でも回路長くとれる(10m)し発熱少ないいい点はこっちにもあるから...(震え声)汎用レジスタが何だとかどうだとかいう話は後々します。ちなみに 4bit と 64bit って 16 倍な気がする(実際回路は 4bit を 16 個つなげるだけだし 4bit の数字(文字)を 16 個集めれば(例:1145141919810931) 64bit になる)けど表現できるものの数は 2^{60} 倍ですからね、約 1000^6 ですからね。すごい全然違うホラ～！

2 CPU の自作方針(本物)

先ほど CPU の仕様についてお話したので、自作方針についてお話ししたいと思います。まず、CPU(仮想空間上のものはずるい)を自作する場合、選択肢としてロジック IC、VLSI、FPGA の三つがあります。ロジック IC は自作 CPU といえたいな作成方法で、後述する汎用 IC を用いて CPU を作ります。今回はこれ。VLSI は、Very Large Scale Integration(超大規模集積回路)の略語で、量産型 CPU の作成方法として最もメジャーなものです。小さな IC をいっぱい並べて CPU を作ります。一応は OpenMPW という Google のお金で LSI を作らせてくれるプログラムがあるらしいのですが、こんなノリで使える程手軽なものではないので今回は断念。FPGA はプログラムで回路を制御してなんやかんやして CPU を作るらしい？あまり詳しくないので分かりませんが「CPU 自作」とかで適当に gg したらそこそこ出てくるので手軽ではあるらしいです。結局電子工作が一番作ってて楽しいし何をしてるか分かりやすいし何よりも完成品が分かりやすく手元に残るので電子工作の知識は多少必要でもロジック IC で作った方がいいです。知らんけど。さて、ロジック IC で作るのは決まったのですが、正直はんだ付けとかしたくない。今回、当然 ROM(計算式とかを入力するところ)も自作しなければならないのですが、先ほど仕様のところで話した通り 16B、つまり 128bit 要るんですよ。つまり ON-OFF スイッチ 128 個！うんち！ここだけでも最低 128×2 の 256 接点要るんですよ。256!? ROM だけで!?さすがにうせやろ!?は一つつかえ案件なのでブレッドボード(画像 1)を使います。ブレッドボードとは部品を差すだけではんだ付けなしで回路が作れるっていうスバラシな代物です。このブレッドボードにロジック IC とか DIP スイッチとかそのほかの部品をぶすぶす差していきましょう。イッパ!!デッデッデッデッデッ!

男は CPU も産めるぞ。俺は CPU のママになるんだ!!!

3 イカれたメンバーを紹介するぜ！

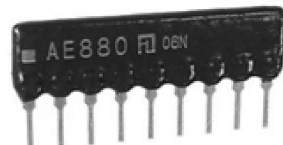
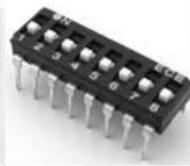
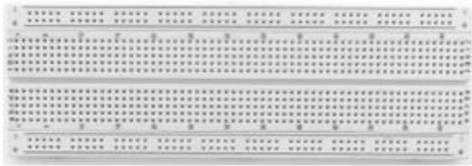
このイカれた時代で今回使うイカれた部品についてイカれた説明をします。

・汎用 IC(74HC シリーズ) そもそも汎用 IC とは何ぞやというお話ですが、簡単に言うと論理回路の集合体です。入力された電圧の high/low を判断し、それぞれの IC 毎に決まったパターンで電圧を出力する、というものです。IC 毎にどういうパターンで出力されるのか、というのはそれぞれのデータシートを見て判断します。データシートはちゃんと見ましょう(N敗)。今回使う 74HC シリーズは基本的な汎用 IC で、high/low の閾値が固定なので調整が楽です。



・DIP スイッチ カチカチ切り替えるタイプのスイッチがいっぱい集まったやつです。ROM を作るのに使います。

・ダイオード(アレイ) 電流の流れる ROM のマトリックス構造(電流の逆流によって意図しない電気の流れ方を防ぐ構造)に使います。今回はダイオードアレイというダイオードを複数個集めたものを使います。これがブレッドボードとか DIP スイッチとかと相性良すぎてめっちゃくちや気持ちよかった...これも前後左右間違えるととんでもないことになるのでデータシートを読みましょう(N+1 敗)。



↑ブレッドボード

↑DIP スイッチ

↑ダイオードアレイ

・その他部品

抵抗とか LED とかコンデンサとか。ここらは基本的な電子部品なので画像とか部品の性質とかは割愛して、いちおう何に使うかだけ説明します。抵抗...LED に電圧をそのまま流してぶっ壊さないように挟んだりブリッジ回路にならないようにしたりするために突っ込みます。LED...数値の出力のために使います。点灯と消灯で 1 と 0 を表現します。コンデンサ...セラミックコンデンサと電解コンデンサの二つがあり、セラミックコンデンサと一部の電解コンデンサ(両極性タイプ)をバイパスコンデンサとして使います。バイパスコンデンサは電源を安定供給しつつ回路の応答速度を上げるものです。電源のプラスからマイナスまで電流を送るのに多少のラグが発生するので、それを小さくするために使います。

男は CPU も産めるぞ。俺は CPU のママになるんだ!!!

4 部品を買おう

部品の説明が終わったので、実際に部品を買いに日本橋に行きましょう。普段私は部品を買う時、共立電子によく行くのですが、こまごまとしているものも多くあるので千石電商にも行きます。という訳で Let's 日本橋散策やありましょう。こちらが今回使う部品一覧です。抵抗や LED は部室に大量にあるので買わない・コンデンサ等は電子部品屋に行ったら無限にあるのでどうでもいい、となってくると問題は 74HC シリーズとダイオードアレイ、8bit の DIP スイッチです。74HC シリーズは論理 IC としてメジャーではあるので確かに IC コーナーに並んではいるのですが、一部売り切れていたの店ハシゴしました。悲しいね。夏真っ只中に行ったので本当に疲れました。ちなみに日本橋の電子部品屋の店員さんは優しいので聞いたら色々教えてくれます。また、74HC154 が廃版になっていたの、74HC138×2 で代替します。一応何をどう置き換えるかの説明を。74HC154 は 4bit の入力を 1~16 番ピンに出力する、というスグレモノです。なんでこんな良いものが廃版になるんだよ。これを 74HC138 という 3bit の

入力を 1~8 番ピンに出力する部品で置き換えます。これだけで置き換え方が分かる人にはわかと思うので置き換え方については割愛します。ほかにも日本橋には様子がおかしい PC パーツ屋や面白い店やプロキシがいそうなビデオ屋もいっぱいあるので興味がある人は是非日本橋に行ってみてください。

5 CPU の挙動

作り方については話さないのかって？面倒くさい。えっと、電子工作ってのは回路図と部品さえあればできちゃうものなので、自分が回路を作っていない限り話せることってほとんどないんですよ。てなわけで割愛します。作り方が知りたい人は「CPU の創り方」という本を読んでください(丸投げ)。実際の制作過程とかもお話したいところではあるんですが、残念ながらもう作ってしまったので、「過程」として残っているものはありません。まあええわ。ここでは CPU の(想定している)挙動について説明します。

```
74HC74 × 1
74HC161 × 4
74HC153 × 2
74HC32 × 1
74HC283 × 1
74HC10 × 1
74HC154 × 1
74HC540 × 1
74HC14 × 1
DIPSW(8pin) × 1x
DIPSW(16pin) × 16
ダイオード1S1588 × 128
抵抗1kΩ × 11
抵抗10kΩ × 9
抵抗100kΩ × 4
抵抗3.3kΩ × 1
抵抗33kΩ × 1
抵抗100Ω × 2
LED × 12
パスコン(0.1uF) × 13
コンデンサ100uF 16V × 1
コンデンサ10uF 16V × 3
Push SW × 2
トグルスイッチ × 2
```

男は CPU も産めるぞ。俺は CPU のママになるんだ!!!

(1) ROM

さきほど ROM について「計算式を入力するところ」と雑に紹介してしまったのもうちよい kwsks。ROM とは Read Only Memory の略で、一般的な PC の ROM には BIOS が格納されています。基本的に簡単に変わってはいけないデータなどを入れる場所です。今回は機械語、つまり CPU が直接実行できるプログラムをここに書き込みます。プログラムを書き込むのに Read Only とはこれいかに...ROM には DIP スイッチを使い、スイッチのオフで 0、オンで 1 を表します。ROM に書き込む機械語ですが、オペレーションコード 4bit+イミディエイトデータ 4bit の 8bit を 16 個の 16B となっています。オペレーションコードとはそのコードで何の処理を行うのかを表すものです。(表参照) イミディエイトデータは表の N を表すものです。表の入力とはなんぞやって思う人もいますが、この入力ってのはスイッチが正常な挙動で動いているかどうかを確かめるためにあるらしい？私もよく理解していないのですが、多分そんな深く考えなくていいと思います。私は。あと、機械語ですが、0~F の 16 進数で表していきます。また、N の値が関係ないものは N を 0 にしています。

0(0000)	AにNを足す
1(0001)	AをBにする
2(0010)	Aを入力にする
3(0011)	AをNにする
4(0100)	BをAにする
5(0101)	BにNを足す
6(0110)	Bを入力にする
7(0111)	BをNにする
8(1000)	該当なし
9(1001)	Bを出力する
A(1010)	該当なし
B(1011)	Nを出力する
C(1100)	該当なし
D(1101)	該当なし
E(1110)	C=0の時N行目にジャンプ
F(1111)	N行目にジャンプする

(2) プログラムカウンタ

何行目のコードを実行するかを表す物。基本 1 ずつ加算するだけ。これで話はおしまい？ ならば、もう こないからねー

(3) レジスタ

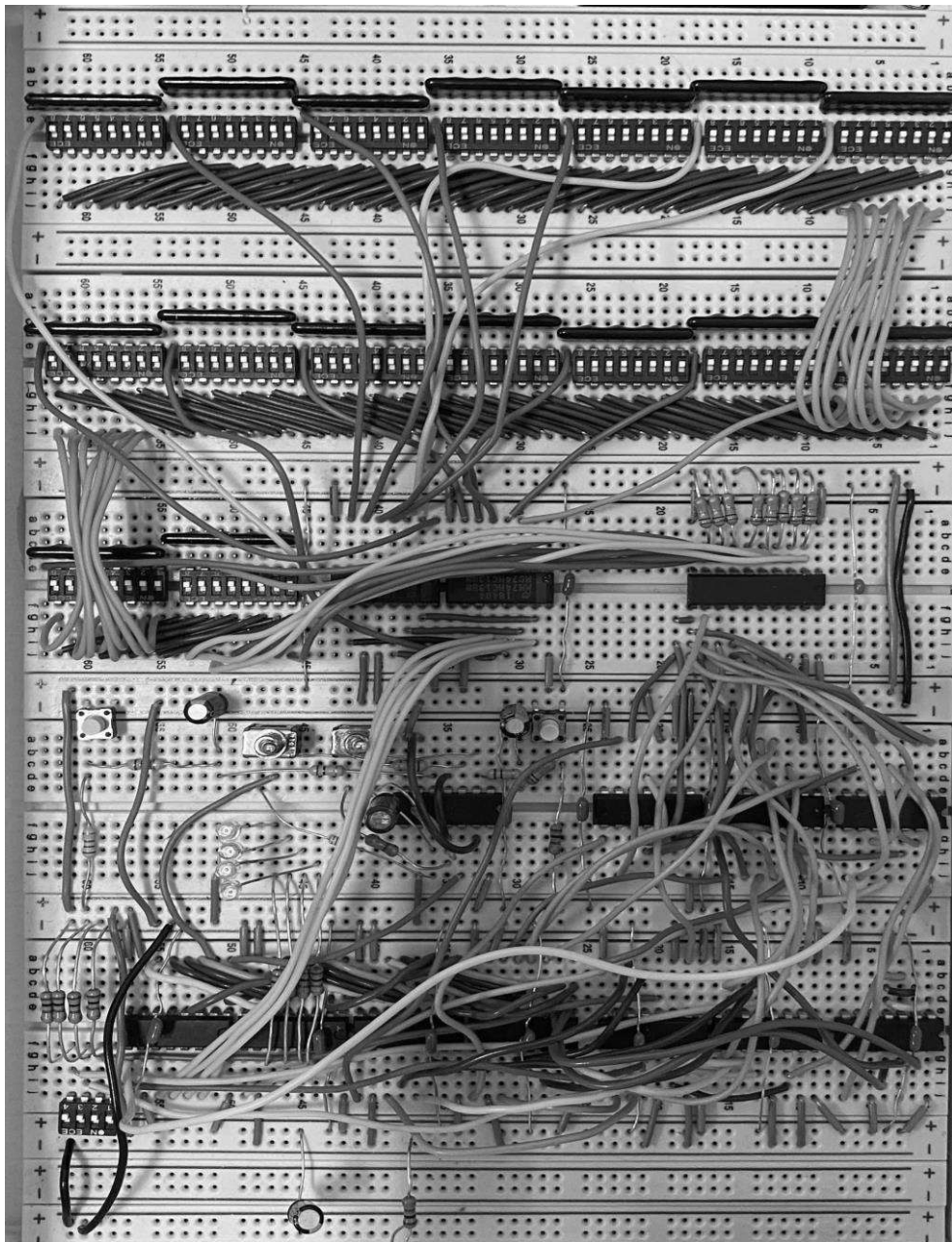
AとかBとかの値を保存しておくもの。まあ普通の変数みたいなものだと考えてもらえれば。ちなみにこっちは揮発性なので電源を切ればレジスタが 0 になります。あと、オペレーションコードのところで急に C とかいうわけわかんないレジスタが出てきたと思うんですが、これはフラグレジスタといって A とか B とかの桁があふれていないか(数値が 16 以上かどうか)を記録しておくものです。9+7=0 とか言われたらたまったもんじゃないですからね。フラグレジスタがあることでちょっとだけ複雑なこともできたりします。

6 デバッグ(迫真)

先ほど話した通り、制作過程で残っているものがないので、「実際に作ってみた」ができないんですよね。仕方がないのでこの方法で。

男は CPU も産めるぞ。俺は CPU のママになるんだ!!!

「完成したものがこちらです。」



男は CPU も産めるぞ。俺は CPU のママになるんだ!!!

と言いたいところなのですが、残念ながらまだ完成していません。なぜならちゃんと動かないからです。という訳で地獄のデバッグ作業 Let's やりましょう！

(1) 大まかな回路を確認する

間違ったピンに部品が刺さっていないかとか LED・ダイオード・コンデンサの向きが間違っていないかとかをざっと確認します。ちなみにコンデンサを逆向きに指すとピンからすっぽ抜けて飛んでいくらしいです。ヒェー!!!

残念ながらそんなことはなかったので次へ。

(2) 正常に動作しない条件を確認する

実は電源すらつかない時って考えられる原因が分かりやすいのでうれしくて、中途半端に動いたり動かなかったりされる方が面倒くさいですよ。地道に一つ一つコードを試してみているとおかしくなるかを調べたかったんですが、会誌の締切りが来てしまったので、無理でした。ツクツェ...ナツクツェ...コトモチホトウ...モソクケインデスワ!!!!(号泣会見)文化祭当日までにはさすがに終わっていると思うので、良ければ見てってください。

(3) それでも分からない時は？

これで原因が完全に分からない場合は、考えられうる原因を全て調べなければなりません。うへえ。テスター過労死不可避。おかしいところが分かったら、そこを治すだけ。ね、簡単でしょ？今回はブレッドボードで作ってるのでまだマシですけど、はんだ付けでやってたら考えられる原因多すぎて剥げますよ。はんだがついてないとかどっか干渉してるとか、下手したらはんだ付けの熱で IC がぶっ壊れるとか(あんまないけど)。

これでよ〜〜〜〜うやく完成です。YATTA！

7 使えるコードの例

・L チカ B1 B3 B6 B12 B8 B8 B12 B6 B3 B1 F0

LED を意味ありげにちかちかさせるコードです。A の後の数値を変えれば光り方も変わります。意味ありげにチカチカはしますが特に意味はないです。

・ラーメンタイマー B7 01 E1 01 E3 B6 01 E6 01 E8 B0 B4 01 EA B8 FF

最初 LED を 3 つ光らせ「A が 16 以上になるまで A を+1 する」を二回実行した後(65 クロック)、LED を 2 つ光らせ、またしても「A が 16 以上になるまで A を+1 する」を二回実行し(130 クロック)、消灯、点灯、A が 16 以上になるまで A を+1 し(194 クロック)、最後 LED を一つ光らせる(195 クロック)、というコードです。自動クロックは 1Hz or 10Hz なので、1Hz にしてあげると 195 秒とちょうどお湯を注ぐ時間+3 分くらいの時間が測れます。あと一分近くになると LED が点滅するので、分かりやすくいいですね。

男は CPU も産めるぞ。俺は CPU のママになるんだ!!!

8 おわりに

こんな他の本のパクリのくせに読みにくくてめちやくちな文章を読んでいただき、ありがとうございます。**BIG KANSYA** します。ところで Word の自動編集機能ってクソ過ぎませんかね。文法：「い」抜きばっか言いやがって。乱れた日本語を使うことの！どこが！問題なんですか！頭に来ますよ。

・おまけ 私とログライク

皆さん、好きなゲームジャンルは何ですか？私はログライクです。ログライクの何が良かったって、やっぱ成長が目に見えて分かることですよ。頭がおかしいケタが出たりしたらもう脳汁ドバドバでやめられないですよ。てなわけで私のおすすめログライクゲームについてちょっと話そうと思います。

1 出会い

中三の時、私は原神というゲームにハマってました。その原神を作ってる会社(ほぼ or みほよ)から崩壊スターレイルというゲームが出るという話を聞いてですね、さすがにやるしかないと思った訳です。それでスターレイルを始め、楽しいな一フンフンくらいに思いながらプレイしていたところ、模擬宇宙というコンテンツがありました。これがま面白いわけで、自分が育成したキャラがバフを大量に確保して超火力を叩き込むという、ね。我が子の成長を見届けている感じがしてすごい楽しかったわけです(ここにきてタイトル回収)。この模擬宇宙が初めて触ったログライクなんですが、そこから新たにゲームを始めることもなく一年弱過ごしていました。すると、出てしまったのです。去年 GOTY(Game Of The Year)三冠を達成した超新星、**balatro** が。私は模擬宇宙は好きでも他にログライクゲームをやっていた訳でもありませんし、当然情報収集などもししていなかったのも、のぼろろの **balatro** 実況動画を見るまで存在すら知りませんでした。ちなみに **balatro** とはどのようなゲームか、というとポーカーとソリティアが融合(公式)ログライクが合わさったゲームで、トランプが全部 A のデッキでポーカーをしたらどうなるかを身をもって体験できるゲームです。ともかくのぼろろの **balatro** 実況動画を見た時、私はこのゲームを遊ぶために生まれてきたのかもしれないと悟り、始めました。結果中三の冬休みは飯風呂睡眠以外は **balatro**、という何とも終わっている生活習慣になっていました。いやね、言い訳させて欲しい。あの中毒性の塊みたいな BGM と SE と演出が悪い。なんだあれは。遊ぶ麻薬ですよ。麻薬ならしょうがないだって？だよな～よく分かってるじゃないですか。ともかく、この終わっている生活習慣の結果がプレイ時間 400 時間超、という訳です。なんだよこれ...こうして私はすっかりログライクプレイヤーになってしまった訳です。

男は CPU も産めるぞ。俺は CPU のママになるんだ!!!

2 育み

どっぷりログライクにハマってしまった私。一般的なログライクプレイヤーであれば **Slay the Spire** くらいやったことないとだめだよなーと **Slay the Spire** を始めました。やはり王道デッキ構築ログライクゲームということもあってドハマリ。こっちのインフレと敵のインフレのバランスが絶妙で面白いんですよ。カードのシナジーなども大量にあるのでいろいろなビルドが楽しめますし、今の難易度に飽きたら「アセンション」で難易度変更もできる。また、ボスギミックがめちゃくちゃ良い。これまで使ってきたデッキだけではどうしようもないようなギミックを持っているので、そのために対策カードを詰んだりポーションを節約したりすると道中がしんどくなる... みたいな。とにかく難易度調節が適切で素晴らしいです。そのくせ強いデッキが作れた時は爽快感満点っていう... なんすか、ログライクの頂点ですか? そうだった。

3 新たな一面

ここまでバランスの取れた()ゲームばかりやっていたのですが、ここでおかしな子と出会います、overdungeon 君です。彼は半年前くらいにやらかして問題になったポケットベアによる作品で、開発者がクラロワと **Slay the Spire** を合わせたゲームと明言しています。このゲームは他のゲームであれば存在してはいけないような組み合わせが大量にあってもうとんでもないです。あと vampire survivors。これは普通にやれば普通の生存系ログライクなんですが、金の卵(通称金玉)を集めればもうおかしいことになります。本当に気持ちいいのでぜひやってみてください。あと個人的に spell disk。このゲームは単純にバランスがめちゃくちゃで、「○○を○○とする」系のどう考えても強いアイテムがあり、上手くいったときはすべてがよい感じに噛み合ってめちゃくちゃでぐちゃぐちゃになって敵が溶けていくのでめちゃくちゃ楽しいです。是非。

4 最後に

以上、私とログライクについてでした。ここでは紹介しきれませんでしたが、Hades というゲームもバランスが良くて育成要素が楽しいのでお勧めです。あと崩壊スターレイルの模擬宇宙はリリース当初でも楽しかったのにアップデートでどんどん新要素が追加されて別物みたいになっていますが相変わらず楽しいので興味がある人は是非やってみてください。あと私はやっていないのですが、学園アイドルマスターというゲームが **Slay the Spire** の育成要素と似ているらしく、キャラクターも可愛い人が多いので個人的に興味があります。以上、会計のおいしいドーナツでした。

あとがき

ここまで読んでいただき、ありがとうございました。

アマ研の会計、おいしいドーナツです。今回部誌の編集を担当させていただいたのですが、二回目とはいえ記事の進捗を部員に確認したりそろっていないレイアウトをそろえたりだとか度重なる Word のクラッシュに悩まされたりとやはり大変でした。編集が終わったのが 3/31 なのですが、まだ見ぬ 83 回生と新高 80 回生に思いを馳せつつ、文化祭が一日一日と近づいていることに少々恐怖を感じております。部誌の内容を見ていただいたら分かる通り、アマチュア無線研究部は様々な活動をしております。部員のやりたいことを応援するという部活の空気と仕組みがあるからだと思います。もし部誌の内容に興味を持ったこれから灘校に入学する方・在校生はぜひアマ研に入部してほしいですね(笑)。
これからも灘校文化祭をお楽しみください!進化して歓迎いたします

Web 版記事と X(旧 Twitter) アカウトのご案内

<https://drive.google.com/drive/folders/1yqlH0H5sw2rTMd0g-ewtFQhoRo1QWCY8>

X(旧 Twitter) @JR3YYJ <https://twitter.com/JR3YYJ>

左側の QR コードから Web 版記事を閲覧・DL 出来ます。

また、右側の QR コードからアマ研公式 X(旧 Twitter) アカウトを開くことが出来ます。



発行年月日 / 2025 年 5 月 2 日

発行者 / 灘校アマチュア無線研究部

灘校アマチュア無線研究部 2025

いかなる場合にも第三者によるこの冊子の複製・頒布、
及びスキャンデータの送信・公開は禁止します