

東京立派報
アソロロボラニ

43号



工学研究部 部報 第43号

もくじ

作者	題名	ページ
いまい	再履修のすゝめ	2
山岸 大騎	iPhone用ポータブルアンプの作成	3
皆川 太志	基盤加工機の使い方（備忘録）	4
Mark.IU	電気通信大学の大学生協は異常なのか？	8
皆川 太志	シフトレジスタの使い方	10
加藤 修一	実験計画法のすすめ	14
河村 大輝	LCDキャラクタディスプレイを使おう	16
横田 順	74HCによる 8bitCPU 制作	24
安井 駿斗&山岸 大騎	【試作品】ドミネーター制作記【PSYCHO-PASS】	30
住澤 拓秀	初めての Java Script	34
本多 寿矢	腹パン抱き枕制作計画	36
N.K.	戦艦大和がライン川を遡上できたか	39

再履修のすゝめ

作成者：いまい

新入生の皆さん、留年予備軍の皆さん、はたまた留年を決めてしまった皆さん、こんにちは。留年を決めたいまいます。今回は工研部報の場を借りて、再履修のやり方を書こうと思います。新入生の方が再履修を行えるようになるのは1年後なのでその時まで大切にとっておいてください。では、説明を始めましょう。

● 履修編

まず、手元に今年度の一年生用、二年生用、必要な方は三年生以上の時間割を印刷して用意してください。また、PCがある方は画面に今までの単位の結果の表を表示させてください。そして一番重要な学習要覧の冊子を用意してください。

用意ができたでしょうか？できた方は次の作業を行ってください。

1. 各時間割で今期とらないといけない授業をマーキングする（その時、再履修の科目はすべての曜日、時間のものをマーキングしてください）。
2. マーキングが終わったらその年に取らないと留年してしまうもの、必修のもの、必修ではないものの順番を学習要覧を見つつ頭の中で理解してください。
3. 再履修の科目はできるだけ楽にとれる先生の授業を優先順位のうえにおいてください。誰が楽に単位をとれるかは、先輩に聞けばおのずとわかってくるでしょう。
4. ここまで終わったら紙に空白の時間割表を作り、月曜一限から順番に埋めていってください。
5. 終わったらできた時間割にしたがって授業を受けるだけです。頑張ってください。また、履修登録を忘れないようにしましょう。俺は2年前期で痛い目を見ました。

● 授業編

さて、時間割の作成が終わりました。履修登録までしばらく時間があると思いますが、授業期間が始またらその時間割通りに行動しましょう。欠席は単位を落とす最大の原因です。再履修の授業には友達と一緒に出席することを勧めます。なぜなら、もし欠席しないといけなくなったり、何かの連絡事項があった時にお互いに確認しあえるからです。また、一年生の授業に混ざらないといけない場合、最初のうちはできるだけ静かに過ごしましょう。全員緊張でがちがちになっているので浮いてしまいます。

履修登録の際、数学の非常勤講師の授業をとる場合は教務課に申請しなければなりません。よっぽどのことがない限り申請が落とされることはないでちゃんと申請しましょう。あとから履修の取り消しをされても文句が言えません。

少ないようですが今言えることはこれぐらいです。うまく説明できている気がしないし、何か忘れていることもある気がするので、再履修がある人はぜひ相談してください。

iPhone 用ポータブルアンプの作成

はじめに

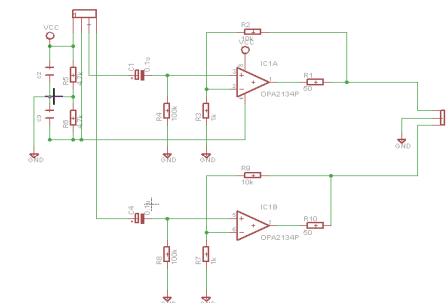
確か、この間 ER4S というちよいと高めなイヤホンを買ったんです。それを iPhone のイヤホンジャックにそのまま刺すのはかっこわるいよねーという感じの理由から、コンパクトなポータブルアンプを作ろうという事になったのだと思います。作ってから結構経つので、もうあまり動機を思い出せません・・・



回路

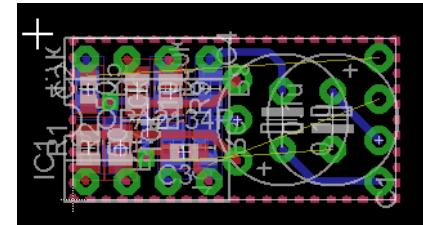
ポータブルという事で、部品点数がなるべく少ないのでポイントです。

音源は手持ちの iPhone4 をターゲットにしているので、音声入力は Dock のみ、電源も iPhone からの 3.3V で動作するようになります。メインの回路自体は、自作ポタアンで有名どころに CMoy ヘッドフォンアンプという名の非反転増幅回路があるのでも、それを採用しました。



部品選定・基板製作

電源電圧が 3.3V なので、実質 ±1.65V 駆動なわけです。CMoy に使われている OPA2134 は ±2.5V 動作なので、低電圧でも動作可能なオペアンプを選定します。色々試したので、もう思い出せませんが、確か Analog Devices 社の AD8616 を使用したと思います。ふんだんにチップ部品を使い小型化しました



実装

組み上がった後の図だけで申し訳ないのですが、箱の中身はこんな感じになっています。小さいスペースに頑張って入れるために色々とアクロバットな事になってますが、こんな配線も許されるのが自作の良いところです。Dock の端子とは、左上にある 1.27mm ピッチピンヘッダで繋ぎます。Dock 端子とピンソケットは、ケースの蓋に直接取り付けています。



評価・あとがき

これをつないで音楽を聴いてみると、明らかに中 / 高音域の明瞭感が増した気がします。が、なんとなく低音が物足りない感じです。これが本当なのか周波数特性を調べてみたい気もするのですが、まだ試験していません。すいません。

大きさ的には、とても小さくすると言う目標は概ね達成出来たので、取り敢えず良しとしておきましょう。

基板加工機の使い方（備忘録）

先進理工学科2年 皆川太志

1 はじめに

基板加工機とは回路基板を彫ってくれる機械です。我々が使えるものは東9(旧F棟)-309にあります。これさえあれば、ユニバーサル基板で配線スパゲッティなんてことが無くなります。ところが、私もそうでしたがあんなでっかい機械をいきなり使いこなすのは無理です。最初はとっても怖いです。そこで、なかなか使用に踏み出せなかつた私自身の経験を踏まえて、今後のために基板加工機の使い方を文章化しておこうと思います。

始めに断っておきますが、この説明は完璧ではありません。出来る限り詳しく書いていますがあくまで、備忘録です。これを読んだからといって、いきなり一人だけで使わないように。初めて使用するときは、必ず使い方を知っている人についてきてもらって手取り足取り教えてもらいましょう。

前置きが長くなりましたが、基本的な使い方を説明します。

2 使い方

これから基板加工機の使い方を順序を追いながら説明します。加工機を初めて触るときのサブマニュアルのように使ってください。

2.1 加工データの作成

2.1.1 EAGLE で回路図・プリント基板配置図を完成させる

まずはEAGLEという回路CADで彫りたい回路図を組み、プリント基板の配置図を作りましょう。EAGLEの使い方に関しては、紙面の都合もあり割愛します。ネットで調べたり、分かる人に聞いたりしながら頑張りましょう。

2.1.2 Eagle の CAMJobs で基板とドリルのデータの基板データを作成する

1. EAGLEのトップメニューのウインドウのCAMJobsの中からgerb 274x.camを選択・起動し、File→Open→Boadからプリント配置図(.brd)を開く。
2. SILK screen CMPをdimentionという名前に変更する。

3. 出力ファイルの拡張子を.plc から.dim に変更する。
4. 右側の枠内で dim 以外を選択解除する。
5. Process Job をクリックし、ドリル以外の基板データを作成する。
6. EAGLE のトップメニューのウインドウの CAMJobs の中から excellon.cam を起動し、プリント配置図 (.brd) を開く。
7. 出力ファイルの拡張子を.drd に変更する。
8. 右側の枠内で Drills と Holes 以外を選択解除する。
9. Process Job をクリックし、ドリルの加工データを作成する。

2.1.3 基板加工機の横のパソコンで加工データを作成する

ここまで来たら東9(旧F棟)-309に移動し作業です。

1. まずは、加工機ノートに名前や使用時間等を記録する。
2. パソコンを立ち上げる。ログイン名パスワードは加工機横のプリントに書いてある。
3. DesignPro を起動する。
4. 右上のアプリケーションを Converter にする。
5. ミリになってるのをインチにする。
6. Gin をおして作成した基板データ (.dim, .sol) を読み込む。また両面基板の時は CMP ファイルを読み込む。
7. Drillin をおしてドリルデータ (.drd) を読み込む。
8. インチをミリに直す。
9. 輪郭線を作成しますというボタンを押す。
10. sol をパターン裏対象に、dim を基板外形対象に設定する。
11. ツール直径は基本的に 0.3 ミリ、基板が細かいなあと思ったら 0.2 ミリにする。
12. 繰り返し回数は掘り残しがあるときには何回か掘る。2回でいい。
13. 重複率は通常 80 %。← 30 %にすると輪郭が太くてグッド。
14. 完了ボタンを押す。
15. 外形用の加工線を作成しますのボタンを押す。
16. 上の方は変えなくて良い。ツール直径は 1 ミリに見える。ミシン目はチェックつけて間隔は 2.0 ミリくらい。
17. ATC 管理設定のを押して、ドリルレイヤーを穴対象にして実行。
18. このあとで失敗してもいいように、ここまでを保存 (拡張子は.mit になる)。

2.2 基板加工機で加工する

2.2.1 加工のための設定

1. 加工機の電源を入れる。
2. アプリケーションの CAMT に切り替える。
3. 指定した加工を行いますをクリック。
4. 枚数の設定を行う。
5. 全部加工対象リストに移動する。ドリル加工（裏）、輪郭加工（裏）、外形加工（裏）をこの順で加工順位リストに移す。OK を押す。

2.2.2 加工機で基板を彫っていく

一応文章化していますが、一連の操作は複雑です。言葉だけで伝えるのには限界があります。必ず、出来る人が操作するところを見てやり方を覚えましょう。

1. 位置合わせでヘッドをどかし、基板をテープで固定する。
2. P1 のボタンの位置が作りたい基板の基準点（最右下）の位置 P1 点なので、P1 点に移動したら HOME ボタンをクリックする。
3. P2(最左上) に勝手に移動して、そこで良ければ HOME ボタンを押す。
4. ドリルを変える。レバーを右にするとダミードリルがとれる。ドリル（穴あけ用）を装着しレバーを左へ。
5. リングを反時計回りに回し、ドリルを引っ込ませる。
6. ドリルを作業範囲外へ退避し、スピンドルを回転させる。その後、リングだけに別の基板をもう一枚噛ますようにしてダウン。
7. 音が変わるものまで（二枚重ねたうち下の基板にぎりぎり触るくらいまで）リングを回す。
8. 続行をクリックすると彫り始める。
9. 掘り終わったらドリルをミリングカッター（表面彫り用）に換装。カッターをリングを回し引っ込む。
10. ドリルと違って間に板を挟まないで、基板にギリギリ触れるまでリングで調整する。
11. 動かすと溝がほれるので、ルーペで確認 太さが 0.3 になるようにする（これをツール直径としてさっき設定した）。
12. 続行を押して彫り始める。

13. 彫り終わったら、ミリングカッターをフォーミングカッター（切り出し用）に換装。
14. これは基板を貫通させるから設定方法はドリルと一緒に。彫り始め方も同様。

2.3 基板を彫り終わったら

1. テープで固定された基板を加工機から剥がす。
2. 小型のノコギリ等で彫り上がった基板を外す。
3. スチールウールで基板を磨き、彫りクズやバリを取り。これらが原因で回路が動かないことが多い。
4. 掃除機で加工機を掃除する。
5. ノートに異常の有無などを書く。

2.4 同じ基板をまた彫りたくなったら

保存した加工データ (.mit) を DesignPro で読み込むと手順 2.2 を省略することができます。デスクトップ上のショートカットアイコンにドラッグアンドドロップすることで簡単に読み込むことができます。

3 あとがき

どうでしょうか？この備忘録が少しでもあなたのお役に立てたのなら幸いです。逆に、迷惑被ったそこのあなた、本当に申し訳ない！最初に断ったとおり、この備忘録はかなり不完全です。わかりづらかったり間違っている箇所もあるかもしれません、そこは気付いたあなたが適宜修正していってください。よろしくお願ひします。

まあでも結局のところ、機械の使い方を覚えるのは人に手取り足取り教えてもらったほうが何倍も効率がいいです。これから基板加工機使いたいけど全然わけわかんないという人も、怖がらずにどんどん聞いてみてください。きっと工研の先輩方は優しく教えてくれるはずです。

電気通信大学の大学生協は異常なのか？

Mark. IU

1

日本の大学には学生生活協同組合、俗に言う“学生協”があります。多くの学生がよく利用し、大学で必要な物の殆ど一食べ物、飲み物、文房具、ノート、本、教科書、雑誌など一が手に入ります。これらはどの大学の学生協にも置いてある商品らしいです。しかし、我らが電気通信大学の学生協には大量のライトノベルと漫画が置いてあります。この事を他大学の友人に言ったところ、「普通生協にそんなものは置いてない、電通の生協はおかしい」と言わされました。本当に電通の学生協はおかしいのか他大学の学生協と比較して調べることにしました。

比較方法は単純です。それは他大学に行っている高校の有人に在学している大学の学生協について調べてもらう、だけです。調べてもらう点は、

- 1 ライトノベルは売っているのか?
 - 1.1 売っているならば売り場面積はどの位なのか?
- 2 漫画は売っているのか?
 - 2.1 売っているならば売り場面積はどの位なのか?

です。

2

今回自分が調べたのは早稲田大学早稲田キャンパス、東京理科大学野田キャンパス、東京工業大学大岡山キャンパス、首都大学東京南大沢キャンパス、そして電気通信大学の学生協です。次の表に結果をまとめます。

表1：調査結果

大学	ライトノベル	漫画	評価
電気通信大学	◎	◎	◎：大いにある
早稲田大学早稲田キャンパス	○	×	○：ある
東京理科大学野田キャンパス	△	×	△：少しある
東京工業大学大岡山キャンパス	◎	×	×：無し
首都大学東京南大沢キャンパス	△	×	

※ここで“大いにある”は専用コーナーがある程度。“ある”はちょっとしたコーナーがある程度、“少しある”は数種類置いてある程度です。

表を見てわかるように電気通信大学はライトノベル、漫画共に専用コーナーがある程で、早稲田大学はライトノベルがアニメ化した作品がある程度で漫画は無いそうです。東京理

科大学はライトノベルは専用コーナーができる程ありますが漫画は置いてないそうで、東京工業大学はライトノベルが一棚無い程度で漫画は無い、という事でした。

3

これらの結果からライトノベルは本の分類に入ることもあり、置いてある大学は少なからずあることがわかる。これは学生に人気なのも影響しているのかもしれません。しかし売り場面積は電通ほど広くなく、一番ひろかった東京理科大学野田キャンパスでも電通の四分の三ほどでした。

一方の漫画ですが、置いてある大学は殆どなく、専用のコーナーが作られる等はあり得ない、という調査結果になりました。オタクの割合が非常に多いと言われる東京工業大学でさえもライトノベルはありますが漫画は殆置いていない、ということでした。

これらの調査結果を見ると、電気通信大学の学生協は他大学の学生協とは少し違っているということがわかります。やはり、電気通信大学はどこかおかしいようです。

4

今回調べたのは数校だけなのでデータとしては信頼性にかけ、正しいとは言い難いものです。なので次回もし調査するならば調べる項目を増やし、調査対象大学を増やしたいと思います。

シフトレジスタの使い方

先進理工学科 2 年 皆川太志

1 はじめに

今回はシリアルデータをパラレル出力するための IC である、シフトレジスタの使い方を説明します。本当はある作品について書こうと思ったのですが、色々あって間に合いませんでしたすいません。新歓までには完成させるので許してください（震え声）。

2 シフトレジスタとは

シフトレジスタは図 1 のような形の IC です。冒頭でも触れましたが、シフトレジスタはシリアルデータをパラレル出力してくれます。え？日本語で話してくれだって？つまり、一本の線で送ったデータ（シリアルデータ）を、複数の端子へバラバラに出力（パラレル出力）できるんです。



図 1 シフトレジスタ (74HC595)

3 シフトレジスタの動作

ここからは実際にシフトレジスタ 74HC595 を例に、シフトレジスタの動作を解説します。とにもかくにも、まずはデータシートを熟読しましょう。それこそが電子工作がうまくなるコツです。読者の方もデータシートをお手元において読み進めていただくと理解の大きな助けになるかと思います。データシートはどこにあるかというと、Google という我らが偉大なる師に在処を聞いてみましょう。

3.1 まずはピン配置と、それぞれの役割を知ろう

さて、データシートを見ると次のような 74HC595 のピン配置があるはずです。

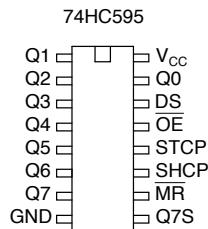


図 2 74HC595 のピン配置

もちろんデータシートにそれぞれのピンの役割ちゃんと書いてあるのですが、初心者の我々にはいささかわかりにくいですね。ここではそれぞれのピンの役割について解説していきます。

< V_{CC} > 電源です。データシートをみると +7V 以下の電源で動作するようですね。+7V 以下の電源を接続してください。

< GND > グラウンドです。これはさすがに説明不要でしょう。

< Q0~Q7 > これがパラレル出力です。8 つあるので、74HC595 一個あたり 8 ピンの出力を増やすことが可能であることがわかります。

< DS > こいつがシリアルデータ入力です。Q0~Q7 に出力したいデータをここから入力します。マイコンなどのシリアル出力ピンはここにつなぎましょう。

< \overline{OE} > パラレル出力許可ピンです。activeLOW とかいてあるので、0V が入力されている時にシフトレジスタのパラレル出力が有効になります。つまり、シリアル入力したデータをパラレル出力したいときに 0V を繋げばよいのです。とりあえず 0V をつないでおきましょう。

< SHCP > シリアル通信のクロックです。ここにクロックが入るたびに、シフトレジスタ内のデータがシフトしていきます。意味がわからない？ちゃんとあとで説明します。

< STCP > ラッチクロックです。ここにクロックが入ると、DS に入力したシリアルデータが Q0~Q7 にパラレル出力されます。そして次にラッチが来るまではそのデータを出力し続けます。つまり、送信したデータを出力して欲しいタイミングでラッチピンにクロックをいれればいいのです。

< $\overline{\text{MR}}$ > リセットピンです。activeLOW と書いてあるので、0V を入力するとリセット機能が有効になります。つまり、リセットピンに 0V を入力した状態で、SHCP にクロックを入力するとシフトレジスタはリセットされるんです。とりあえず電源電圧をそのまま入力しときましょう。

< Q7S > シリアル出力です。ここから下段のシフトレジスタにデータを移していくことが出来ます。下段？は？？とお思いの方、ちゃんとあとで説明するのでご安心を。

3.2 基本動作

シフトレジスタの全ての動作はデータシートにタイミングチャートがあるので、それと見比べて理解しましょう。ここでは基本的なシフトレジスタ動作を説明します。

1. シフトレジスタは、クロック (SHCP) の立ち上がりでシリアル入力 (DS) を 0 番目の内部レジスタ (一時メモリ、0 から 7 番目まである) に読み取ります。
2. クロック (SHCP) の立ち上がるたびに、シリアル入力 (DS) の内容を Q0(0 番目のレジスタ) に移動、Q0 の内容を Q1(1 番目のレジスタ) に移動、Q1 → Q2(2 番めのレジスタ)… と「シフト」していきます。(次々にデータがシフトするのでフルレジスタですね)
3. 手順 2. を 7 回繰り返すと、全部で 8 つのデータがそれぞれシフトレジスタのレジスタに格納されます。ここで注意して欲しいのは、まだレジスタのデータはパラレル出力ピンには出力されていないことです。ちなみに Q7 のデータだけはシリアル出力 (Q7S) に出力されています。このことは今は一見意味無いようでも重要なんです。
4. ここまで来たらシリアル送信したデータをパラレル出力しましょう。ラッチクロック (STCP) を叩くと、レジスタのデータはパラレル出力ピンに出力されます。レジスタに 0 が入っていれば 0V が、1 が入っていれば電源電圧が出力されます。

ここまで流れで 8 つのデータをシフトレジスタに送り、シフトレジスタの 8 つのパラレル出力ピンからそのデータを出力することができました。パラレル出力ピンにそれぞれ LED などを接続しておけばわかりやすいかと思います。

3.3 さらに出力ピンを増やす～シフトレジスタの直列つなぎ～

先ほどの手順 3 で、Q7S だけ出力しやがって！って思いました？しかし、このシリアル出力 (Q7S) をもう一つのシフトレジスタのシリアルデータ入力 (DS) へ接続するとどうでしょう？1-Q7(1つ目のシフトレジスタの 7 番目) → 2-Q1(2つ目のシフトレジスタの 1 番目) となるので、もう一つのシフトレジスタに前のシフトレジスタのデータを伝えることができます。つまり 2 つのシフトレジスタは直列つなぎすると、さらにピン数を増やすことが出来るのです！

接続の仕方は下の図 3 のようになります。同様に 3 つ 4 つとシフトレジスタをつなぐことができます。これを使えば思いのままに出力ピン数を増やすことができます。しかし、出力ピンの数だけクロックを打たないといけないので、全データの送信に時間がかかることがあります。

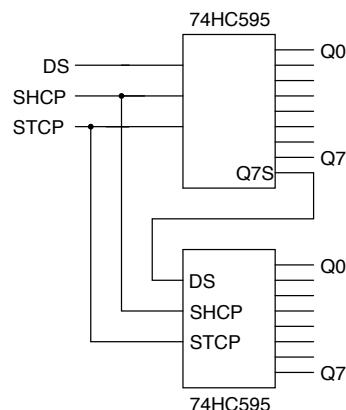


図 3 シフトレジスタ (74HC595) の直列つなぎ

あとがき

少し難しかったでしょうか？シフトレジスタに各種クロックやデータを送信する際には、おそらくマイコンを接続することになるかと思います。しかし、説明してきた動作をマイコンで実現するためのプログラムについては、ここでは紙面の都合もあり敢えて触れていません。プログラムについては工研 wiki や優秀な先輩に聞いてください。ネットで調べるのもいいかもしれませんね。

実験計画法のすすめ

知能機械工学専攻
修士1年 加藤 修一

1. 緒言

ものづくりをするにあたって、設計、計算というのは避けて通れない作業である。機械設計では、たとえばネジの大きさから、材質、寸法の決定といった設計計算を行い、電子回路設計でも、回路内の抵抗値やコンデンサーの値などの決定といった設計計算が必要となる。

しかし、実際に計算を行って行くと、一つの性能を満たすためには、もう一方が満たさなくなる背反的性質を持つものや、パラメータ同士の相関関係があるものなどがあり、解を見つけるのは非常に難しい。

そこで、考えられるのが実験計画法の導入である。最初の段階から細かく、総当たり的に計算を行うのは非常に不毛なことである。実験計画法を用いることで、少ない実験回数で、これらへんに解があるだろうといった「あたり」を見つけることができる。

本来は、実験などの授業でやる方が望ましいが、残念ながら本学の授業内で開講されていない（知能機械工学科旧カリキュラムの内容に基づく）ため、この場を借りて紹介する。

2. 実験計画法とは

実験計画法とは、その言葉の通り計画的に実験を行うことである。もともとは、農業のようになかなか回数を多く行えない実験を効率的に行うために、作られた。

たとえば、花を育てる際に、1日に水を何Lあげるか、肥料は何kgあげるか、これにより花はどのように成長するかという問題を考える。

この「水を何Lあげるか」、「肥料は何kgあげるか」という成長に関わる要因を「因子」という。

まず実験計画を立てるため、因子の水準数を決める。水準数は因子を考えられる範囲の中でどれだけ細かく考えるかということである。2水準では水は0.5L, 1.5Lの時を、肥料は1kg, 3kgのときを考えるという決定である。

今回は3水準で考え、簡単に水、肥料の量、「少」「中」「多」とする。

表1 植物の成長結果

		肥料の量		
		少	中	多
水の量	少	×	△	○
	中	△	△	○
	多	×	△	△

このとき、表1のように実験は3×3回で、9回行えば良いことになる。そして表1より肥料が「多」、水は「少」、「中」の箇所を細かく見れば適切な量を求めることができる。また、水準内では数値を用いるだけでなく、例えば、花の種類のように連続的でないものも取り扱うことができる。

水準を決めることで、本来無限に考えられた実験が有限回で効率的に行なうことが出来る。これが実験計画法の基本である。

3. 直交表を用いた実験計画法

先ほどの例では2因子3水準であったため、総当たりで行っても9回で済んだ。しかし、土の量や面積、日光の量など、因子をn個まで増やしていくと実験回数は3^n回となり、実験するのが難しくなる。

そこで、土の量と水の量の組み合わせによる成長への影響は無く、水の量と土の量はそれぞれ単独に成長へ影響していると考える。この組み合わせによる影響を「交互作用」というが、後の項で述べる。

交互作用がないとすれば、上手く条件を組み合わせることで、より少ない回数で実験結果を求めることができる。このとき用いられる組み合わせが直交表である。

表2 L9 直交表

		因子			
		1	2	3	4
No.	1	1	1	1	1
	2	1	2	2	2
	3	1	3	3	3
	4	2	1	2	3
	5	2	2	3	1
	6	2	3	1	2
	7	3	1	3	2
	8	3	2	1	3
	9	3	3	2	1

直交表は、その名の通り各列が直交している行列のことである。各数字が同じ頻度で、異なる組み合わせで登場している。各行は実験条件を表しており、各列に因子を当てはめて、条件を設定する。表内の数字が水準の内容を示し、先ほどの例だと1が「少」、2が「中」、3が「多」となる。

4因子3水準の例で考えると、総当たりで行うと81回実験を行う必要があるが、L9直交表に当てはめると、わずか9回の実験でそれぞれの因子の傾向を捉えることができる。

実際に用いる場合は、1列分を繰り返し実験での

誤差項として用いた方がよい。

少ない因子数では、それほど恩恵がないが、多い因子数では、大きく実験回数、計算回数を減らすことができる。

2 水準では L4,L8,L12,L16 直交表、3 水準では L9,L27,L81 直交表、4 水準、5 水準のものや、複合直交表などがある。

この田口らが確立した直交表を用いた実験計画法を、一般的に「実験計画法」と呼ぶことが多い。

4. 交互作用について

先ほどの考え方は、あくまで交互作用がない場合であり、実際は、各因子の組み合わせによって結果が変化しているかも知れない。その交互作用を考慮しつつ、直交表を用いる場合、線点図を利用することで求めることができる。

L9 直交表での線点図を図 1 に、L27 直交表での線点図を図 2 に示す。丸い印が因子であり、線が交互作用を表している。数字は、直交表の用いる列を表している。L9 直交表では 2 つの因子間の交互作用を、L27 直交表では 3 つの因子間の交互作用について求めることができる。

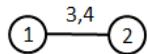


図 1, L9 直交表線点図

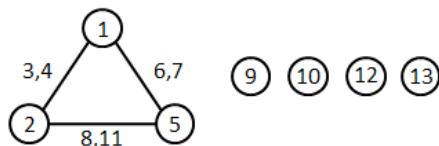


図 2, L27 直交表線点図

交互作用は、水準数から 1 引いた数の列を用いて求める。この線点図による列の割り振りは一定の法則があり、好き勝手に決めたり、入れ替えたりすることはできない。この線点図については、参考文献にある「直交表と線点図¹⁾」が詳しい。

5. 応答曲面法について

これまで、結果一つ一つをグラフで見る考え方を行ってきたが、複数の因子を捉えた一つの近似式で表すことで、交互作用も含めた検討が可能となる。

そこで応答曲面法(Response Surface Methodology : RSM)の導入があげられる。最小二乗法による近似式の延長と考えるとわかりやすいと思う。

一般的には、応答曲面法では得られた実験結果の近似式を二次の多項式で表すことが多い。次数をこれ以上上げると、不自然な傾きの変化が現れることがある。

二次式であるので、因子 2 個の間に働く交互作用をとることもできる。

詳しいことはインターネット上のフリーソフト²⁾の解説書などが詳しい。

6. 主成分分析・分散分析について

一応、応答曲面を求める際は、主成分分析³⁾や分散分析⁴⁾を用いて、各因子による寄与率を求めた方がよい。

3 項で上げた誤差項を一列分取っておくことで、誤差よりも寄与率が低いデータを見つけることも重要なとなる。誤差よりも寄与率が低いものは、考慮しなくてよい場合か、考慮していない交互作用が影響した場合、実験が上手くいっていない場合などが考えられる。

実験の妥当性を検証する意味でも、軽く利用できるようになっておいた方がよい。

7. 最後に

以上が、実験計画法の概要である。この手法を用いることで、効率的に設計計算を行うことが可能となる。実際に用いると、総当たりを行う必要がなくなることもあります、手放せなくなるほどである。

詳しく知りたい人は、図書館にも関係書籍⁵⁾があるので、ぜひとも勉強してほしい。

参考文献

- 1) 田口玄一:直交表と線点図、丸善株式会社(1962)
- 2) 轟 章のホームページ FREE SOFT
RSM for Excel 2013 年 3 月 16 日閲覧
<http://todoroki-lab.net/ssoft/soft.html>
- 3) 芳賀敏郎, 橋本茂司:回帰分析と主成分分析、日科技連出版社(1980) など
- 4) 広津千尋:分散分析、教育出版(1976)
- 5) 菅民郎:Excel で学ぶ実験計画法:シックス SIGMA と重回帰分析、オーム社(2004) など

LCD キャラクタディスプレイを使おう

知能機械工学科 1 年 河村大輝

今回は、秋月電商に 5 百円くらい（たしか、たぶん。）で売っている、SC1602BS *B という液晶ディスプレイの使い方を書いてみようと思います。

・概要

LCD キャラクタディスプレイというのは、下図のような液晶にドットで文字を表示してくれるモノです。

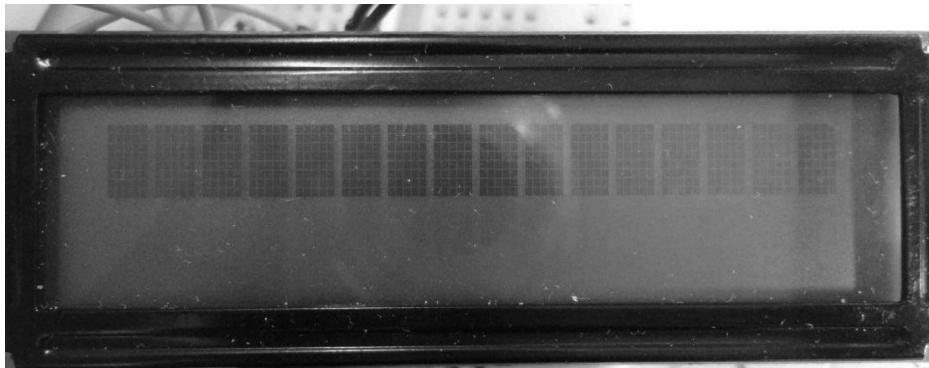


図 1

今回はこちらの LCD キャラクタディスプレイモジュールと、Atmel 社の AtTiny2313 というマイコンを使って文字を表示する方法を書いていこうと思います。

LCD ディスプレイは、文字を出すだけならば、マイコンのピン出力と delay_ms 関数だけで動かすことができるため、LED チカチカが出来るようになったマイコン初心者が挑戦するにはちょうどいい課題だと思います。また、タイマー割り込みを使えば、たとえば AD 変換を行いながら各レジスタの値を LCD ディスプレイに表示するなど、コンピュータープログラムでいう “printf” に近い動作をさせることができます。非常にデバッグがしやすくなるので、使えるといろいろ応用の効くアイテムだと思います。

・参考資料

下記のページでピンの名前や文字コードなどを参照しましょう。

液晶ディスプレイを使う

http://homepage3.nifty.com/mitt/pic/pic5_17.html

LCD モジュール (SC1602BS*S) 説明書

http://www.oaks-ele.com/oaks_board/oaks16/data/document/oaks16-lcd/man_sc1602.pdf

Attiny2313 データシート

<http://www.avr.jp/user/DS/PDF/tiny2313.pdf>

LCD モジュール付属の説明書

- ・文字表示の原理

基本的にパラレル通信という通信方式で LCD に文字や設定のデータを送信します。

パラレル通信とは、2 本以上の通信線を使って電圧の High Low でデータを送信する通信方式です。

今回は 8 本の信号線と、電圧の立ち下りで信号の読み込みをする E (イネーブル) の線、設定を変更するレジスタと文字を表示するレジスタを選択する RS (レジスタセレクト) の線を使い、合計 10 本の線で 8 bit のデータを送ります。

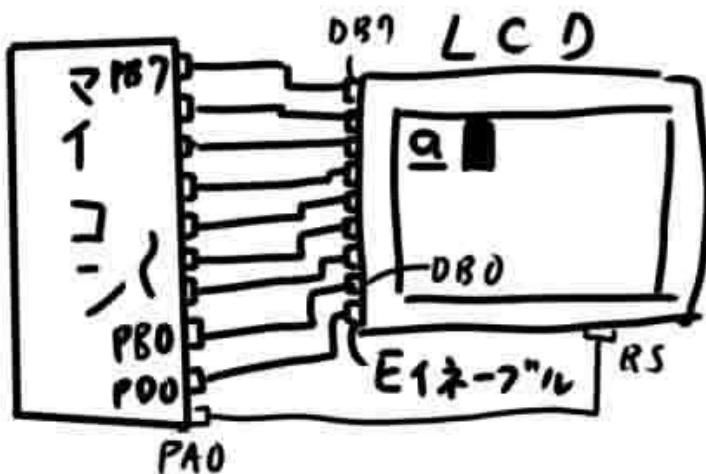


図 2 こんな感じにつなぐよ。

通信の手順は、
E (イネーブル) を Low(0v)に
↓
DB 0 ~ 7 と RS に送るデータを出力し
↓
E を High(5v)に
↓
220 n s 以上待ち
↓
E を Low に (この E の High から Low の立ち下りのタイミングで、LCD は DB ピンと RS ピンの信号を読み取ります)

という流れになります。

・回路

今回書くプログラムに使う回路です。尺の都合上ピンの名前等は LCD モジュール説明書やデータシートから確認してください

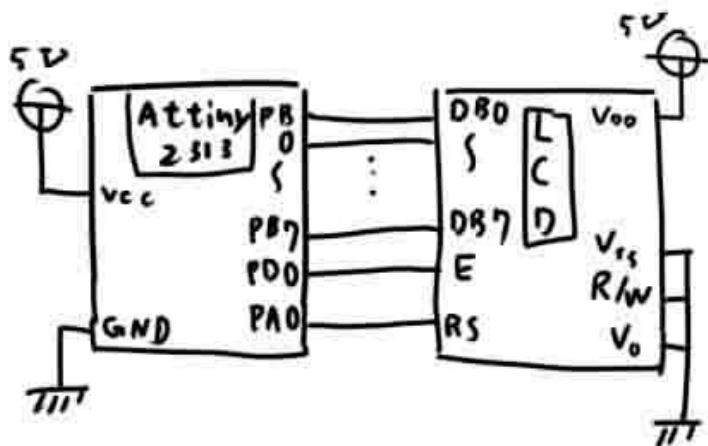


図3 回路図。僕が回路図と言ったら回路図です。

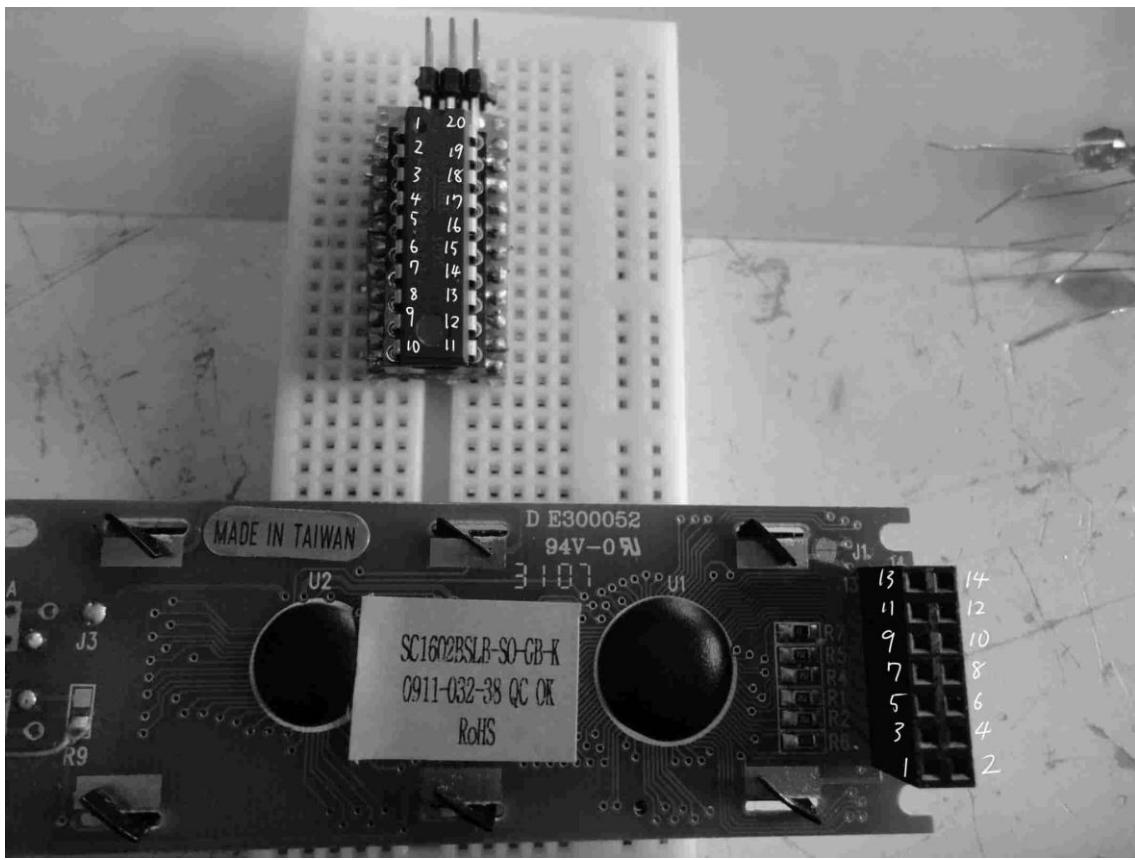


図4 ピンの番号

マイコンの PB0～7 は LCD の DB0～7 につなぎます。ここで 8 bit の文字情報などを送信します。

マイコンの PD0 は LCD の E (イネーブル) につなぎます。

マイコンの PA0 は LCD の RS (レジスタセレクト) につなぎます。

LCD の V0 は表示のコントラストを変えるのに使います。Vdd と同じ電位で最も薄く、Vss と同じ電位で最も濃くなります。今回はグラウンドにつないで最も濃く設定しています。

プログラムはうまく動いているのにコントラストが薄くて文字が表示されない、なんてことになっては泣くに泣けないので、しっかり確認しましょう。

Vdd に 5V、Vss に GND をつなげた状態で V0 が GND (グラウンド) につながっていれば、図1のように黒い四角が見えるはずです。

LCD の R/W は LCD からデータを読み出すか、データを書き込むかの選択をするピンです。今回は書き込みしか使わないので GND に設置して Low にしておきます。

- ・サンプルプログラム

```

//DB0~7→PB0~7
//RS→PA0
//E →PD0
#include <avr/io.h>
#include <util/delay.h>
//delay_ms 関数を宣言.
void delay_ms(uint16_t i)
{while(i--) _delay_ms(1);}

//まずは8ビットの信号を送信する関数を作る.
//初期化用のデータ送信関数.
void command_trans(unsigned char data)
{
    PORTD = 0b00000000;      //E を Low に。 DB0~7 のデータの読み取りを禁止.
    PORTA = 0b00000000;      //RS を Low。 制御レジスタ.
    PORTB = data;           //PORTB に引数を代入.
    PORTD = 0b00000001;      //E を High に。 DB0~7 のデータの読み取りを開始.
    delay_ms(1);            //1ms 待つ.
    PORTD = 0b00000000;      //E を Low に。 DB0~7 のデータの読み取りを禁止.
}

//文字表示用のデータ送信関数.
void data_trans(unsigned char data)
{
    PORTD = 0b00000000;      //E を Low に。 DB0~7 のデータの読み取りを禁止.
    PORTA = 0b00000001;      //RS を High。 データレジスタ.
    PORTB = data;           //PORTB に引数を代入.
    PORTD = 0b00000001;      //E を High に。 DB0~7 のデータの読み取りを開始.
    delay_ms(1);            //1ms 待つ.
    PORTD = 0b00000000;      //E を Low に。 DB0~7 のデータの読み取りを禁止.
}

int main(void)
{
    DDRB = 0b11111111;
}

```

```

DDRA = 0b00000001;
DDRD = 0b00000001;      //使用するピンを出力ピンに設定.

//LCD の初期化.
delay_ms(100);      //電源 ON 後、 1.5 ms 以上待つ.
command_trans(0b00110000);
delay_ms(10);        //4.1ms 以上待つ.
command_trans(0b00110000);
delay_ms(10);        //100us 以上待つ.
command_trans(0b00110000);
delay_ms(1);         //40us 以上待つ.
command_trans(0b00110000); //ビット長を 8bit に設定.
delay_ms(1);         //40us 以上待つ.

//以上の操作でデータの送信方式を 8bit モードに設定します.
command_trans(0b00111000); //2 行表示モードに設定.
delay_ms(1);         //40us 以上待つ.

command_trans(0b00001111); //ディスプレイ ON、カーソル ON、ブリンク ON に設定.
delay_ms(1);         //40us 以上待つ.

command_trans(0b00000001); //ディスプレイクリア.
delay_ms(10);        //1.64ms 以上待つ.

command_trans(0b00000110); //カーソル右移動、表示シフトなし.
delay_ms(1);         //40us 以上待つ.

//文字の送信.
command_trans(0b10000000); //カーソルのアドレスを先頭に表示.
delay_ms(1);         //40us.
data_trans('H');     //文字を LCD に表示.
delay_ms(1);         //40us.
data_trans('e');     //文字を LCD に表示.
delay_ms(1);         //40us.
data_trans('l');     //文字を LCD に表示.
delay_ms(1);         //40us.
data_trans('l');     //文字を LCD に表示.
delay_ms(1);         //40us.
data_trans('o');     //文字を LCD に表示.
delay_ms(1);         //40us.

```

```

data_trans('W');      //文字を LCD に表示.
delay_ms(1);         //40us.
data_trans('o');      //文字を LCD に表示.
delay_ms(1);         //40us.
data_trans('r');      //文字を LCD に表示.
delay_ms(1);         //40us.
data_trans('l');      //文字を LCD に表示.
delay_ms(1);         //40us.
data_trans('d');      //文字を LCD に表示.
delay_ms(1);         //40us.

while(1)
{
    //TODO:: Please write your application code
}
}

```

ここまで

- ・うまく動くと



こうなるハズです。

このプログラムはなるべく単純に作ったもので、まだまだ改良の余地はあります。入力ピンの読み込みができれば、ビジーフラグを読み取って通信速度を上げることもできますし、CGRAM（キャラクタジェネレーターRAM）をいじれば自分でドット絵を作ることもできるらしいです。参考資料を読み込んでいろいろチャレンジしてみましょう。

おわり

74HC による 8bitCPU 製作

情報・通信工学科 1年 横田 嶺

1. 目的

汎用論理 IC で CPU を作ること

2. 概要

・CPU とは

ノイマン型コンピュータの構成要素として演算装置、制御装置、記憶装置、入力装置、出力装置があります。CPU（中央演算処理装置）とはその名の通りコンピュータの中心的な処理装置としてはたらく電子回路であり人間で言う頭脳にあたります。一般に CPU はプログラムによって様々な数値計算や情報処理、機器制御を行なっていて、みなさんの使っているパソコン、テレビ、携帯電話などあらゆる電子機器に内蔵され欠かせないものとなっています。まさに現代社会を支えていると言っても過言ではありません。



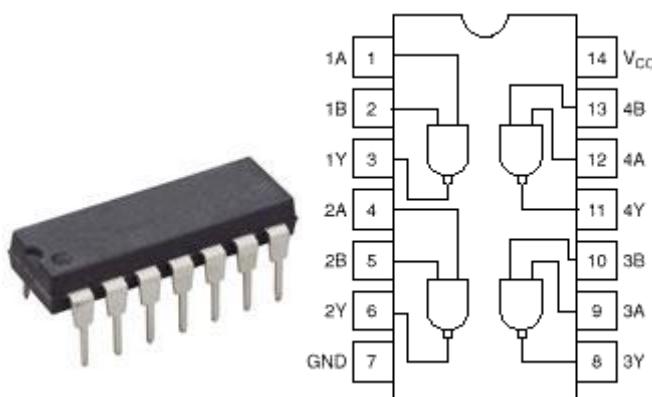
Intel Core i7



AMD Athlon 64

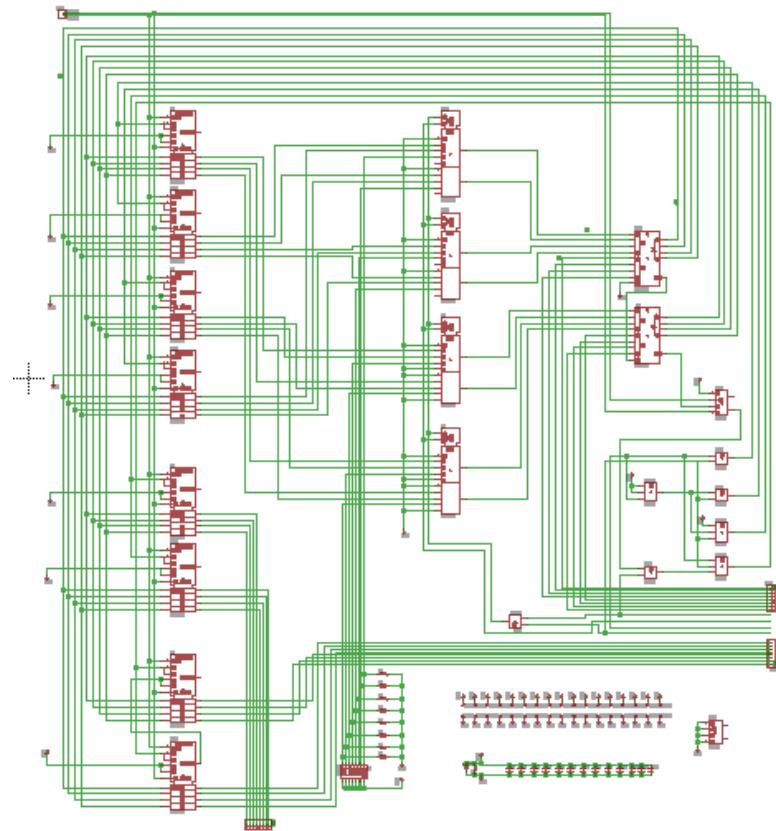
・74HC シリーズとは

デジタル回路では電圧の High (1) と Low (0) を用いて動作しています。すなわち二進数なのでブール代数の論理演算によって様々な動作をすることができます。74HC シリーズはテキサスインスツルメンツ社製汎用ロジック IC 74 シリーズの高速版 (High-speed CMOS) であり AND、OR、NOT などの基本的な演算を実装する論理ゲートやフリップフロップ、カウンタなどの論理機能ブロックをパッケージにしたものです。東芝や RENESAS など様々なメーカーから発売されています。



74HC00 NAND ゲート

3. 回路



CPU 部の回路図

・使用部品

・CPU 部分

- ・74HC74 (フリップフロップ : キャリーフラグ)
- ・74HC161 (フリップフロップ : レジスタ、カウンタ)
- ・74HC153 (データセレクタ)
- ・74HC32 (OR ゲート : 命令デコーダーの一部)
- ・74HC10 (3 入力 NAND ゲート : 命令デコーダーの一部)
- ・74HC283 (4bit 全加算器 : ALU)

・ROM 部分

- ・DIP スイッチ (4bit オペレーションコード + 8bit データ)
- ・74HC154 (スイッチ)
- ・74HC540 (インバータ : 論理反転)

・クロック・リセット回路部分

- ・74HC14 (シュミットトリガインバータ : チャタリング防止、
クロックジェネレータ)

4. 仕様

- ・スペック

比較用に現行最上位 CPU

	Intel Core i7 3970X	自作 8bitCPU
コア数	6	1
汎用レジスタ	64bit × 128	8bit × 2
浮動小数点レジスタ	82bit × 128	0
プログラムカウンタ	64bit	8bit
フラグレジスタ	64bit × 8	1bit
演算	浮動小数点演算可能	8bit 加算のみ
動作クロック	3.5GHz (最大 4.0GHz)	3MHz 程度

参照：[インテル® Itanium® アーキテクチャー・ソフトウェア・デベロッパーズ・マニュアル、第 1](#)

[巻: アプリケーション・アーキテクチャー](#)

- ・命令一覧

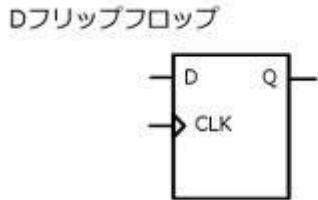
上位 4bit がオペレーションコード、下位 8bit がイミディエイトデータ

MOV A,Im	A レジスタに Im を転送	0011xxxxxxxx
MOV B,Im	B レジスタに Im を転送	0111xxxxxxxx
MOV A,B	A レジスタに B レジスタを転送	000100000000
MOV B,A	B レジスタに A レジスタを転送	010000000000
ADD A,Im	A レジスタに Im を加算	0000xxxxxxxx
ADD B,Im	B レジスタに Im を加算	0101xxxxxxxx
IN A	入力ポートから A レジスタへ転送	0010xxxxxxxx
IN B	入力ポートから B レジスタへ転送	0110xxxxxxxx
OUT Im	出力ポートへ Im を転送	1011xxxxxxxx
OUT B	出力ポートへ B レジスタを転送 ※	1001xxxxxxxx
JMP Im	Im 番地へジャンプ	1111xxxxxxxx
JNC Im	C フラグが 1 でないとき Im 番地へジャンプ	1110xxxxxxxx

※「OUT A」はありません

5. 仕組み

- ・レジスタと CPU



フリップフロップはクロックが立ち上がるのと同時に 1 か 0 か、すなわち 1bit を記録出来ます。つまりフリップフロップ 1 つで 1 bit のレジスタと言えます。D が入力、Q が出力なのですが、入力と出力と繋いでみるとどうなるでしょう。1 クロックで出力が入力に帰還しデータを保持することができます。ここで入力前にスイッチを付けます。すると「新しいデータのロード（書き込み）」か「自分自身をコピー（保持）」を選択することができます。

入力と出力を繋いだ状態を考えます。A レジスタから A レジスタへデータをコピーすることですから「MOV A,A」と表せます。大げさに言えば命令セットが「MOV A,A」だけの 1 クロックで 1 命令を実行する 1 bitCPU ですね。CPU の基本回路とはコレだったりします。例えば出力から入力へ戻る途中にインバータを挟んでみましょう。クロックが立ち上がるたびに A レジスタの中身が論理反転する回路になりました。アセンブラーで書けば「NOT A」です。このように転送の途中に演算回路があれば転送命令は演算命令に変わるので。

つまり CPU というのは転送命令を繰り返すだけのロジックというわけです。

- ・データセレクタ

データの流れを変えるにはスイッチが必要です。しかし CPU に物理的なスイッチを付けることはできないのでデジタルの信号、つまり H と L の電気的なスイッチを用います。論理演算により入力段の H,L で出力段の H,L を決めるようゲートを組んでやれば良いのです。自分でゲートを組まなくとも IC として様々なデータセレクタが存在するのでそれを使いましょう。使い方は真理値表を見れば一目瞭然です。

- ・ALU (演算機)

ALU は前述の通り転送の途中に挟んでやれば良かったので 74HC181 という NOT,OR,AND,EXOR など殆どの計算機能が網羅されている IC を使いたいところでですが現在入手が困難なため全加算器の 74HC283 を使用しています。加算するだけなので中身は論理和の組み合わせになっています。

・フラグ

条件分岐を設計する際に用います。今回は加算した結果がレジスタの扱える数(8bitなので11111111)を超えたときの繰り上がり(キャリー)を記憶するC(キャリー)フラグだけ使います。条件分岐命令はこの値の0か1かを参照して実行されます。

・プログラムカウンタ

現在の命令の位置(アドレス)を指示示す役割をするのがプログラムカウンタです。この値をROMに送ることでROMに格納された命令(+データ)を呼び出すことができます。命令が実行されるとカウントアップし次の番地へ移ります。このように命令は上から順に実行されます。

指定された番地へジャンプする「JMP Im」という命令がありました。この場合はプログラムカウンタ(PC)の値を変更すれば良いので「MOV PC,Im」と同じです。

・I/Oポート

純粋なプロセッサにI/Oポートは付いていませんがワンチップマイコンには付いていますね。仕組みは簡単。出力ポートはレジスタの出力をむき出しにすれば良いのです。その代わりレジスタとしては使えなくなりましたが。入力ポートはデータセレクタに直接H,Lを入力です。

・命令デコーダ

ここまで流れとしては、1:データセレクタがレジスタを選択しALUへ送る、2:ALUは演算を行い全レジスタに演算結果を送る、3:信号により各レジスタの保持またはロードの決定(転送先の決定)、4:次のクロックの立ち上がりと同時に実行(書き込み、つまり転送)

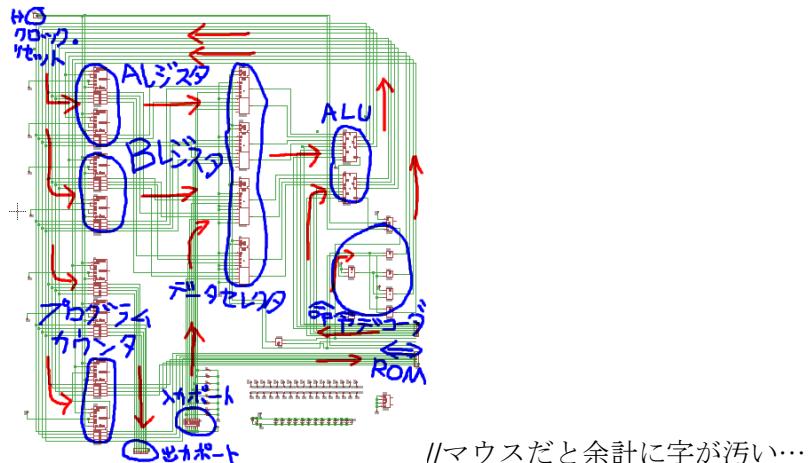
という感じでした。ここで指示しなくてはならないのはレジスタの選択信号と保持かロードの信号です。この指示を決めるのは命令の上位4bitと条件分岐命令で参照するCフラグになります。これら5bitを6つの信号に変換する回路が命令デコーダです。もちろん論理ゲートで組むのですがなかなか大変です。



*ワンチップマイコン

CPU・メモリ・I/Oポート・タイマを1つのチップにまとめたもので電子機器の制御用に使われている。メーカーによりPIC、AVR、H8などがある。

6. 動作



- 1 : クロック立ち上がる
 - 2 : プログラムカウンタからアドレスが出力される
 - 3 : ROM から指定されたアドレスのデータが送られる
 - 4 : 命令デコーダで命令を解釈する
 - 5 : データセレクタでレジスタを選ぶ
 - 6 : ALU で演算する
 - 7 : プログラムカウンタのカウントが進む
- ここまで 1 クロック

7. まとめ

スマセんまだ実際には作っていません。

とりあえず CPU の原理を勉強したかったのです。

汎用ロジック IC は 30 年近く前に活躍していたもので現代において大規模な集積回路の製作は FPGA などにその座を譲っているそうです。ですからこういう体験を今のうちにやっておかないと複雑な集積回路はますますブラックボックス化してしまいます。是非とも完成させたいものですね。続きは工研 wiki で。

今後の方針

- ・基板掘る (かなり大変そう…)
- ・ALU の強化
- ・ROM をスイッチではなく IC にする
- ・クロックは水晶を使う
- ・7 セグ LED を点灯させたい

8. 参考図書

「CPU の創りかた」渡波郁 2003 マイナビ

【試作品】ドミネーター製作記【PSYCH O-PASS】

学部4年(?) 安井駿斗 & 学部2年(?) 山岸大騎
執筆：安井駿斗

——^{システム}その銃口は、正義を支配する。

1 「携帯型心理診断・鎮圧執行システム”ドミネーター”起動しました」

皆さんは「サイ○パス」見ていますか？ 私は歴代の創作アニメの中でも最高級レベルで好きなんですが。思わずBDとか買っちゃいましたよ。割と痛い出費なのでコミケ¹の資金がなくなりそうで辛いですが、これは買わざるを得ない作品でした。

この部報が頒布される頃には放送が終了していると思いますが、今はクライマックスで次の放送が待ち遠しいです。朱ちゃんの覚醒っぷりがパネエ。

ここまでハマったアニメは久々でした。本当に何から何までカッコイイ。キャラはもちろん、演出・脚本、そして近未来という舞台設定を活かした数々のハイテクとその世界観。

そんなハイテクの中でもこの作品に深く関わり活躍する、主人公たちの武器。そう、それがドミネーターです。このドミネーターがとにかく一番カッコイイ。喋る銃くらいなら、な〇はStrikerSとかでも出てきましたが、敵に合わせて適切に変形して相手を攻撃するシステムを備えたドミネーターはマジでカッコイイです。

そんなドミネーターがめちゃくちゃ欲しい、とか思ってると、大日〇技研さんから早々にキットが販売されていました。（画像は著作権上載せられないので、各自「ドミネーター キット」とかでググってください）。

……いや、なんだよこれ。外装だけ？ こんな何もないハリボテが2万円？

ふざけるなど。俺らが欲しいのは

デコンポーザー（本気モード）に変形して！

「対象を完全排除します。ご注意下さい」とか喋って！！

ロボットとかを一撃で分子分解するような!!!!

そんなドミネーターだったはずだろ!!!!

……分子分解はまあちょっと無理かもしれないけど、それでも2つ目くらいまでなら実現できるでしょう！

Q. 実現できるならいつやるの？

A. 今でしょ！

そんなわけでたく、ドミネーターを作る計画が始動しました。

¹コミックマーケットの略。年2回開催されるオタクの祭典。電通生紳士の嗜み。

2 「慎重に仕様を定め、対象を製作してください」

製品の仕様を定める前に、皆さんに前置きとして伝えておかねばならないことがあります。それは、ドミネーターの変形はどれも**物理学上可能**であるという点です。ドミネーターには、通常・鎮圧モードの「パラライザー」、人身殺傷モードの「エリミネーター」、対象排除モードの「デコンポーザー」の3モードが存在します。図1, 2を見てわかる通り、デコンポーザーはパラライザーの原型を留めていません。しかしこれは、謎空間から謎のパーツが出入りしたり、パーツが空中に浮遊していたりするわけではありません。

すなわち、質量・体積は保存されており²、ニュートン力学に従っている、単純な「変形」なのです（このあたりの変形については、設定資料集に非常に詳しく説明されています）。そんな「実現可能性」を備えているのも、ドミネーター製作に踏み切った要因の一つですね。

それでは前置きが長くなりましたが、今回の仕様は以下のように定めました。

- 3モード変形可能（手動）
- 人に向けると犯罪係数を読み上げる ⇒ 人身感知・音声再生
- パラライザー時に光る部分はすべて光らせる
- 4月の新入生勧誘までには試作品を完成させる

ざっくりとはこんな感じです。いくら実現可能とはいえ自動変形は現実的に不可能と判断したため、今回は手動での変形を目指します。次二つについては、まあこれくらいはやりたいですね。既に件のキットに、LEDと音声回路を仕込んで喋らせている動画がYou○ubeに上がっているので、これ以上となれば人身を感知するしかないかと。

最後ですが、いわゆる納期です。こういうことはいつまでにやるのか、ということを目標に加えることで、製作を円滑に進められます。特にこの部報を読んでいるアナタに、工研の活動とはどのようなものなのか宣伝するのに最適と考えたため、この時期に、とりあえずの試作品を完成させることを目標にしました。放送終わってまだ旬だしね。



図1: パラライザー



図2: デコンポーザー

3 「対象の本格製作が開始されました」

3.1 外装部分の製作

まず当然、変形できるような構造にするため、設定資料集をよく見て考察します。その結果、ドミネーターには各部に高性能なマニュピレーターが存在し、マニュピレーターが各部品を支

²エネルギーは保存しているか分かりません。というかどう見ても保存してないです。

え、変形時にはそのマニュピレーターが伸び縮みしたり移動したりすることで、一見無理な変形を可能にしていることがわかりました。

今回はそのマニュピレーターを針金に置き換えて、変形を可能にさせることにします。見栄えは正直悪いですが、まあ試作なので贅沢は言わず。これでマニュピレーターの動きの研究をした上で、今後はしっかりとしたものに置き換えていくことにしようと思います。

外装パーツの作成は、どこの家庭にでもあるレーザー加工機³を用います。こいつはCADデータを元にレーザーを照射して、高精度でモノを焼き切るという素晴らしい機械です。これを使ってアクリル板を切り出し、それらを針金とパテで、理想とする変形が可能なようにつなぎ合わせていきます。

図3,4はそのようにしてできあがった外装部分です。パラライザーモードからデコンポーザーへの実際の変形の様子になります。とりあえず、こんなものでしょうかね。

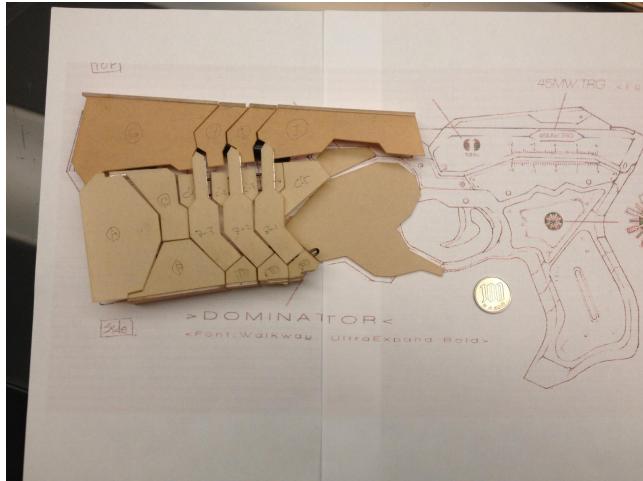


図3: パラライザー

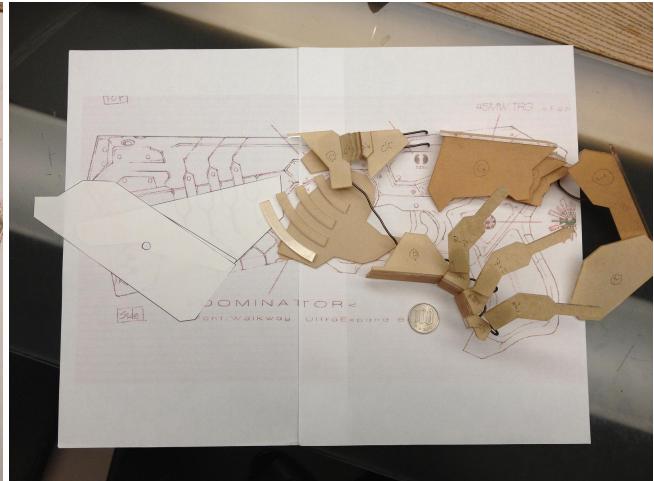


図4: デコンポーザー

3.2 内部回路の製作

次に、ハイテク武器ならではの電子回路の方です。回路設計を行い、これまたどこの家庭にもある基板加工機⁴で基板を作ろうと思います。

とりあえず、メインとなる制御ICには工研で非常によく使われるAVRマイコン、ATmega328のQFPパッケージを使い、基板の小型化を図ります。

音声再生には、MicroSDカードからデータを読み取り、mp3デコーダIC VS1053bでの再生とします。これで、PCでMicroSDカードに保存したmp3データを直接再生することができます。

人身感知には、焦電型赤外線センサを用います。これはよく、電灯の自動点灯・消灯に使われているセンサです。こいつは搭載された2つのセルから感知する赤外線情報を元に、そのセル間で得られた温度差を感知して出力するセンサです。通常これは2つのセルで周囲外気と人身の温度差を見ることで人身感知しますが、今回はこれを片方のセルの感受部を常に隠しておくことで、単純な温度センサのような使い方にします。このようにする理由は、至近距離に人身がある際に2つのセルが共に人身をとらえてしまい、温度差が得られなくなる可能性があるからです。

³普通ない。

⁴やっぱりない。

そして、ドミネーターの発光部分。単純に光らせるだけなら、LED に電池を繋ぐだけで大丈夫ですが、やっぱりドミネーターはスライド部分にある発光部が、ふわ～っと広がるあの光り方がカッコイイんですよね。やはりあのように光らせたいので、PWM制御してやります。マイコンのポートが足りなくなるのでシフトレジスタを使いましょう。ついでにグリップ部のLEDも同様。こちらは往復運動っぽいので、その違いもちゃんと再現します。また、不正ユーザー時に赤色になることを念頭に入れ、使う LED は赤緑 2 色の LED にしました。

あとついでに、グリップ握った時にユーザー認証ボイスが再生されるように、グリップ部にフォトリフレクタを仕込んでおくことにします。

まだ実装できていませんが、そんな機能をだいたいのつけた基板が図 7, 8 になります。小さくするために両面基板を使うことにして、寸法は 60 mm × 35 mm になり、グリップになんとか収まりそうです。



図 5: 発光部分

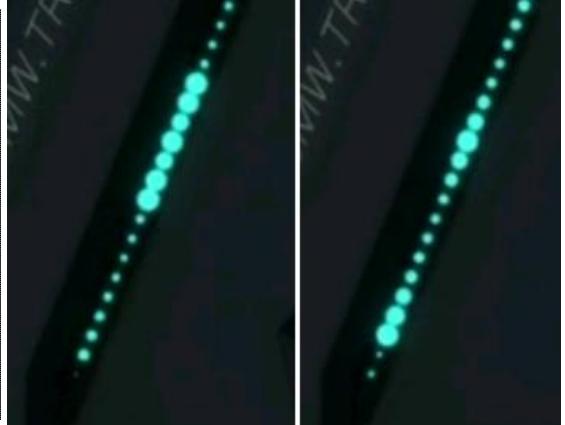


図 6: スライド部発光部分の広がる様子

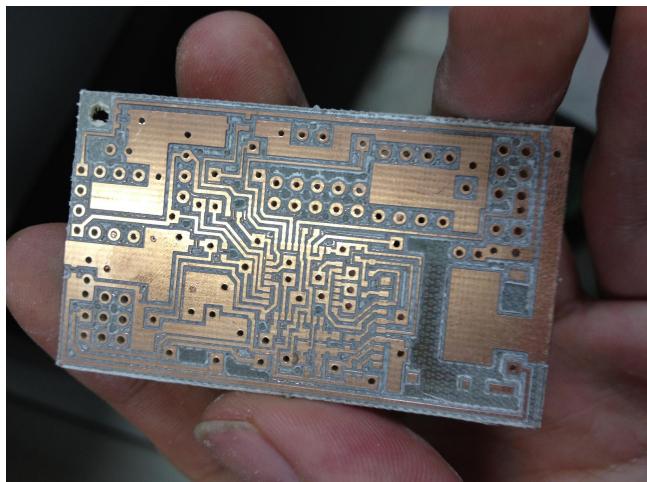


図 7: 基板表面

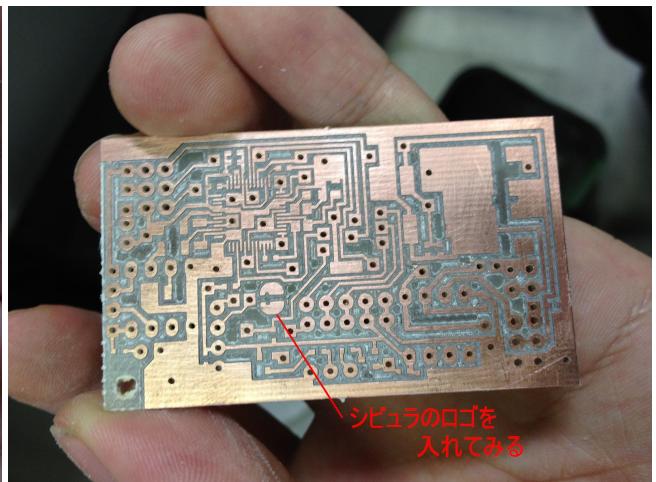


図 8: 基板裏面

4 「徹夜係数 オーバー”400” デスマーチ対象です」

このように部報に書いた以上、新勧の当日には恐らく試作品が展示されていると思います。恐らく見栄えとしてはかなり悪いものになっていると思いますが、あくまで試作品であり、今後検討を重ねた上でよりアニメに近いものに仕上げるつもりなので、今回は見逃してあげてください。

はじめての Java Script

情報・通信工学科 2年 住澤 桢秀

1. はじめに

はじめまして、情報・通信工学科新2年の住澤というものです。他の方々が難しいことを書いてらっしゃるので、私は簡単にまとめてみようかと思います。

新入生の皆さんにはプログラミング言語というものをご存知でしょうか？文字通り、パソコンでプログラムを書く時に必要な言語です。その種類は多数ありますが、この電通大では、1年後期に全員C言語というプログラミング言語を授業で学びます。必修です（泣）

今回私は、このC言語に似たJava Scriptという言語に挑戦してみようと思います。この言語は主に「動きのあるインターネットのサイト」に使われます。テキトーなサイトで右クリックして「ソースを表示」ってすると、この言語が使われてたりします。

この言語を始めるなら、htmlとか周辺の知識もいるみたいなんですが、とりあえず無くともなんとかなるかなあ～とか思ってます（笑）

まあ、初心者が興味本位で学び始めただけですから、ガチゼイの人はこの記事読み飛ばしちゃって下さい（笑）

一応、C言語をちょっとは知ってる人向けに書こうと思います。

2. 環境

今回はMBP (Mac Book Pro) でやってみたいと思います。

3. 準備

我が部が誇る超優秀なY先輩の助言のもと、まず、「mi」というテキストエディタをインストールしました。もちろん無料です。

次に、参考となるサイトを探しまして、「初心者のためのJavaScript講座」というサイトを見つけました。yahooで「JavaScript 入門」って検索すればすぐ出てきます。（googleだと出てくるかわからん笑）以降、このサイトにそって勉強していきます。

4. やってみた

ブラウザの初期設定とか、html全体の型とかはページの都合上飛ばします。サイトを見て下さい。

まず基本として、「/*」と「*/」で挟んだ間は「コメント」です。

alert("~~");と書くと警告が出ます。（~~の部分に警告文を書いて下さい）どこまでプログラムが正常に動いているかチェックする時とかに使います。

`document.write("～～");`と書くと、～～の部分に書いた文字を表示することができます。C 言語の `printf("～～");` と同じです。

次に変数について。C 言語では、数字なら `int` や `double`、文字なら `char` 等を使いましたが、JavaScript では、文字も数字も `var` を使います。(これ便利 笑) `var` 変数名; って感じで使います。たとえば、変数 `n` に 10 を変数 `moji` に「おはよう」と代入する時は、

`var n=10;`

`var moji="おはよう"` (文字を代入する時には " " でくくる)

みたいな感じで使います。

配列を宣言する時はちょっと特殊で

`var 配列名=new Array();`

と書いて、値を代入するには

`a[0]="1 月";`

`a[1]="2 月";`

みたいに書くか

`var a=new Array("1 月","2 月");`

みたいにカンマで区切る方法もあります。

あと、C 言語と同じように関数も存在します。構造も非常に似ていて、

`function 関数名() {`

関数の内部

`}`

って感じで書いてきます。関数名の後につく()には引数を書きます。

C 言語でもよく使う `if` 文、`switch` 文、`for` 文、`while` 文は JavaScript でも全く同じなので省略します。

5. 結論・感想

とりあえず、基本はこんな感じです。ここまで C 言語とほとんど同じなんで、C 言語やってる人はすぐできるでしょう。やってない人もやっとくと授業がラクになる・・・かも(先に C 言語の勉強するべきか 笑)

で、この後はページの都合上次回以降の部報で書けたら書きます。(というか私もそんなに進んでいないので書けません。)

では、このへんで失礼します。

6. 参考

初心者のための JavaScript 講座

<http://www.pori2.net/js/index.html>

腹パン抱き枕制作計画

知能機械工学科 1年 本多寿矢

はじめに

コミュ症なことで有名な私達電通生。大学に入学し親元を離れ、孤独な生活に枕を濡らすことも少なくないのでは、そこで自分の与えた刺激によって反応を返してくれる抱き枕があると楽しいのではないか、と考えたため今回の製作計画に至った。というのは建前である。

世の中には様々な嗜好がある。その中の一つに「腹パン」というものがある。お腹は人体における重要な器官・内臓を守っている。そんな急所を殴ることは危険な行為であり、特に対象が女性であればなおさら忌避される行動であろう。だがそれゆえにリョナ好事家には好まれる一つのジャンルとして確立されている。そんな行為である腹パンを二次絵などで見るだけではなく、実際にやりたいと思うことも無理はない。しかしそんなことをすれば、相手がよっぽど広い心を持った女性でない限り一発で警察のお世話になることだろう。そんな悩みを解決すべく、私はあるモノの製作計画を打ち立てた。それが今回の表題にもある「腹パン抱き枕」なのである。

作品コンセプト

昨年のエレコンで身につけた AVR マイコンの技術を用い、押すと音がなるような機構を内包した抱き枕を制作する。

留意事項

- 押す場所が分かりやすいイラストを用意する。
- 衝撃を感じるのには圧力センサは接触時間が短すぎるため、ポンベを用いた気圧センサによる検知をする。
- 耐久性を確保すべくポンベには行楽用浮き袋またはペットボトルを用いる。
- 気圧センサの検知した値を AVR の A/D 変換を用いてデジタルな値にして、鳴らす音を変える。
- マイコンの処理でランダムに鳴らしたり自分で好きな音を鳴らせるようにもする。

圧迫の検知について

圧迫を検知する機構としては次の2通りの方法を考えている。まず、気圧センサによって枕内部の圧力を検知する方法である。これはポンベを押す部分に仕込み押された圧力を圧力センサで検知するというものである。ポンベは押すためのサイズの大きいものとセンサなどの回路を入れるための小さなものを使い、2つのポンベをチューブで接続する（図1）。

次にスイッチを使い直接押されたことを検知するものであるが、これについても同様にポンベを2つ使う。しかし、回路などはポンベの外側に置き小さいポンベが膨らむことによってスイッチを押すようとする。（図2）

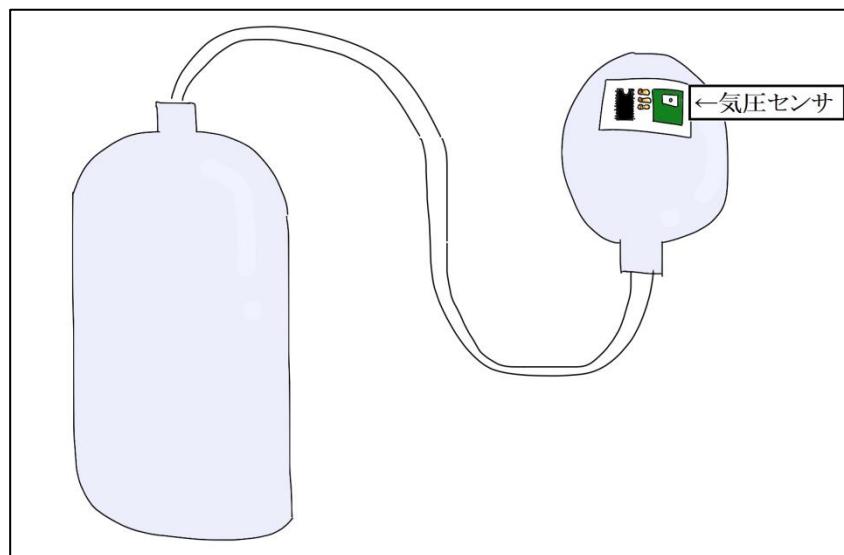


図1 気圧センサを使ったもの

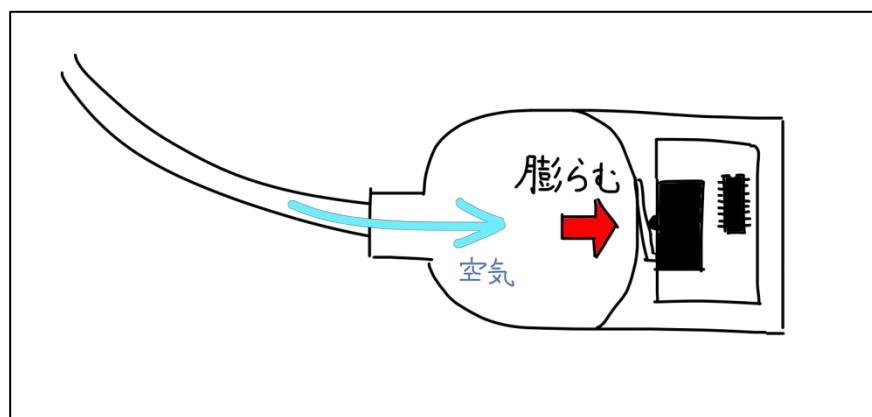


図2 スイッチを使ったもの

終わりに

まだ計画段階で全く見通しが立っていない感じがする。どうなることやら全く検討もつかない。製作を進めつつ AVR の使い方もしっかりと身につけていきたい。どのような機構を仕込んだとしても抱きまくらであることに変わりはないので、耐久性および見た目に重点を置けるようにしたい。

戦艦大和がライン川を遡上できたか

N.K.

1. 緒言

昨年、某魔法少女アニメ（以下、ストパン）の劇場版が公開された。ストパンは、第二次世界大戦の名パイロット達を魔法少女に置き換え、敵であるネウロイを倒していく物語である。その中では、第二次世界大戦中の兵器や、物品、建築物等が数多く描かれている。もちろん、私は5回見に行った。その中の描写で、戦艦大和が巨大な浮きをつけてライン川を遡上し、主人公たちを助けるというシーンがある。

私は、そのシーンを見て、ふと疑問に思った。いかに大きなライン川といっても、実際にそのようなことは、可能であるのだろうか、と。

そこで、今回その検証を行う。

2. 戦艦大和とは

戦艦大和は第二次世界大戦中に、大日本帝国海軍（以下、IJN）によって作られた大和型戦艦の一隻である。作られたきっかけや46cm砲を積んだ戦艦など、彼女のエピソードは事細かにあるが、作中と時代の流れが異なるうえ、今回はもっとも考えなくてはいけないであろう、その大きさを中心とした性能諸元についてここで述べる。

表1 大和の各大きさ、重量¹⁾

排水量	基準 64,000t 公試 69,000t 満載 72,809t
全長	263.0m
水線長	256.0m
全幅	38.9m
公試喫水	10.4m
燃料搭載量	6,500t
スクリュー直径	5m

たぶん、各数値は、最終時のものと新造当初のものが入り混じっていると思われるが、とりあえず、このまま考える。



図1、大和型戦艦²⁾

3. ライン川とは³⁾

スイスのトーマ湖から流れ、ボーデン湖に入り、ドイツ・フランスの国境を北に向かう川。流域としては、スイス、リヒテンシュタイン、オーストリア、ドイツ、フランス、オランダがあげられる。

第二次世界大戦中を基軸として、世界が描かれているため、ドイツ・オーストリアが敵の占領下、フランスは解放された形で描かれている。また、敵は水を苦手としているという設定から、ライン川が最前線となっている。

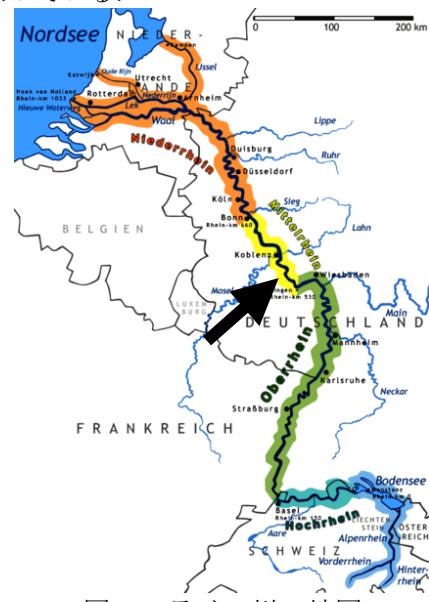


図2、ライン川の地図

劇中では、サン・ビットという町で主人公がピンチに陥り、戦闘が開始する。しかし、そのような地名はなく、近い名前でもビットブルクくらいである。ただ、ローレライ付近で戦闘が行われたように描写されている。劇中では、とりあえずローレライの場所を地図上の矢印で記載している。

（ちなみに、ローレライとライン川は結構遠い）

4. 大和を浮かべるために

地図上で確認したところ、全幅40m全長263mあったとしても、結構なところまで遡上できそうなので、ライン川の深さについて考える。ライン川を適当にgooglemapを用いて航空写真で見ていくと、全長100mぐらいの船は、上流まで遡上しているのが確認できる。この船を参考文献⁴⁾で確認すると全長110m、全幅11.4m、排水量3000t(水深3.5m)である。その文章内には、川幅の限界から11m程度が限界と書かれてはいるが、運行上の限界だと考え、ひとまずは無視する。

では、ここから得られた3.5mを基準に考えると、6.9mは持ち上げる必要が出てくる。船を38.9m×

$263m \times 10.4m$ の質量が $69,000t$ の密度が一定な立方体であると仮定すると、 $1m$ 持ち上げるのに必要な浮力は、 $6634t$ が必要になる。それでは、 $6.9m$ では、 $45,775t$ となる。

この質量を持ち上げるために、まず燃料を抜く。きっと川を遡上する段階で曲がりきれないことから、タグボートなどを用いて遡上していると考える。そして、川を動いていることから水も計算外として抜いて考える。

燃料と水の量は満載排水量から基準排水量を引いた値であるから、 $8,809t$ 。IJNでは、公試排水量を燃料・水が $2/3$ の時としていることから、 $5,873t$ がその質量となる。これにより、浮力は $39,902t$ になる。

武器、弾薬も引いてしまいたいが、劇中では主砲も用いての対空戦闘を行っていることから、弾薬は満載であると考えられる。

さて、ここからは浮きを用いて浮上していると考えられるので、浮きのサイズについて考える。映像を見ると、幅は大和の $1/3$ 程度、長さは全長の 6 割程度のように見える。

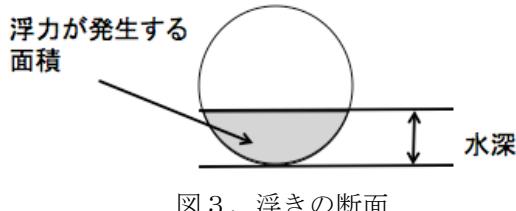


図3、浮きの断面

これをそのまま計算すると 1 個の浮きによる浮力は $21,595t$ 、これが二つあるので、 $43,190t$ となる。これで満足するように思えるが、大和自身は $3.5m$ になるが、浮きが $13.2m$ と意味のないことになっている。では、実際図3のように浮きが $3.5m$ 分沈んだ時の浮力は、面積から求めると $4,587t$ 、二つ合わせても $9,174t$ となり、足りなくなる。

よって、 $3.5m$ という前提条件が間違ないと考えられる。そもそも、スクリューが水上へはみ出すという恐怖があるので、これ以上でなければならない。

5. このままではまずいので

ライン川に限らず、川で用いられてきた軍艦から深さを考える。日本の川で用いられてきた砲艦で考えると最大 $4.2m$ のものがある。この時もスクリューよりも小さいし、この状態でも $41,131t$ を持ち上げなければならず、浮きによる浮力も二つでも $11,820t$ である。またドイツのモニター艦 Arminius では $4.55m$ で、 $38,809t$ に対して $13,193t$ である。

実際の船で考えても足りないため、今度は、川の断面積から水深を推定する。いつも通り、川の断面を長方形と置き、流量は、上流と下流で等しいものとする。すると断面積が等しくなければよいため、もっとも川幅が狭いローレライでは、川幅 $113m$ 、水深 $25m^5$ 、川幅は最大で $500m$ 位まで大きくなるため、流量が同じとしたときの水深は $5.65m$ ぐらいとなる。この水深では、スクリューもはみ出さない。

$5.65m$ では、 $31,512t$ 持ち上げる必要があり、そのときの浮きは $17,650t$ となる。これでも、不可能だったから、あの映像でのように行くのは、無理だと思われる。

6. 最後に

きっと、ライン川はもっと深いんだよ、運行を考えなければきっと余裕ですよ。余裕。資料が見つかからなかっただけで、流速さえわかれば、もっと水深が深いと推定できたはずですよ。

もっと、浮きも長くて、直径も大きいから、簡単に大和も浮かんだのですヨ、きっと。 $1\sim2m$ 持ち上げて何の意味があるか分からぬけど。大体川から $10km$ の範囲でしか戦闘できないのに、遡上させて何の意味があるかとか考えては行けないですヨ。あのアニメは女の子の可愛さを楽しむ物ですヨ。魔法少女だし。



図4、主人公可愛いよ、主人公

そして、フィクションは本当にはり得ないから面白いのですヨ。

参考文献

- 1) Wikipedia 大和(戦艦), 2013年3月18日閲覧 [http://ja.wikipedia.org/wiki/%E5%A4%A7%E5%92%8C_\(%E6%88%A6%E8%89%A6\)](http://ja.wikipedia.org/wiki/%E5%A4%A7%E5%92%8C_(%E6%88%A6%E8%89%A6)) など
- 2) 戦艦大和/壁紙/No.1, 2013年3月18日閲覧 <http://military-wallpaper.idol-net21.com/Yamato01-1280-768.htm>
- 3) Wikipedia ライン川, 2013年3月18日閲覧 <http://ja.wikipedia.org/wiki/%E3%83%A9%E3%82%A4%E3%83%B3%E5%B7%9D>
- 4) 小澤茂樹, ヨーロッパにおける河川輸送の現状と今後の展望—ライン川におけるコンテナ輸送を中心にして— 2013年3月18日閲覧 http://www.itej.or.jp/assets/www/html/archieve/jijyou/201007_00.pdf
- 5) Otash abroad ドイツ歴史街道 2013年3月20日閲覧 <http://jiyuseikatsu.web.fc2.com/germanrhein.htm>

国立大学法人 電気通信大学
工学研究部 部報 第43号

発行所 国立大学法人 電気通信大学工学研究部
〒182-8585 東京都調布市調布ヶ丘 1-5-1 サークル棟 2 階
URL <http://delegate.uec.ac.jp:8081/club/koken/>
E-mail koken@koken.club.uec.ac.jp

発行 皆川 太志
編集人 住澤 拓秀
表紙 河村 大輝
発行 2013年3月22日
執筆 工学研究部 部員