

What about

麻布学園物理部無線班

2回
80集路

Electronic circuit?



はじめに

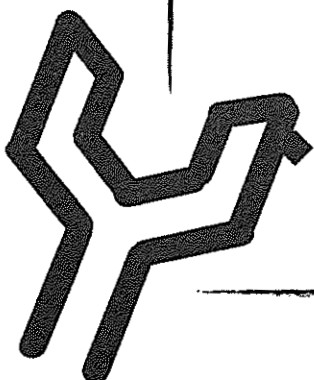
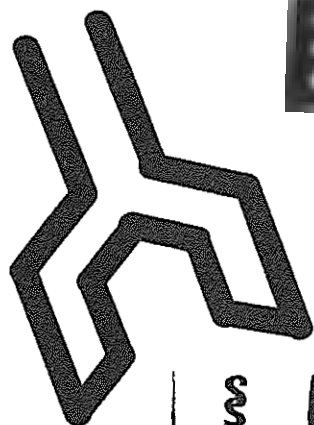
『Gという単位を使っている時点で、我々とはあまり関係ない話』と去年の回路図集にありましたが、CPUクロックは限界に達し、マルチコア、77トコアの時代となり始め、一方マイコンの多機能化も止まることなく、チップで様々なことが出来るようになりました。そんな中、我が部では16bit、32bitのCPUにも手を出し始め、少し追いつき始めています。

この回路図は、麻布学園物理部無縁班の部員が7年間研究製作したものを記したものです。中には未完成のものもある為、この本の通りに作ってもうまくいかない場合がありますので、あらかじめ御了承ください。

H2 会計 川崎敦史

目次

§	はじめに				P1
§	目次				P2
§	アナログ系				P4
◆	タッチセンサ	H1	渡邊	翔	P5
◆	STEP ~逃走中~	M3	棚辺	駿介	P8
§	コンピュータ系				P11
◆	AZABAC61	H2	川崎	敦史	P12
		M3	池上	涼平	
§	ロボット系				P24
◆	超電磁式鈍足列車	H2	中嶋	剛大	P25
		H1	西村	陽樹	
		M3	竹下	諒	
	Climber-3	H2	竹島	啓純	P37



◆ Lorelai2008 H1 菅原 寛生 P42
M3 中岡 勇太

◆ Teacher-4 H2 中嶋 剛大 P52
H2 竹島 啓純

§ ゲーム系 P56

◆ アウトバーンレーシング M3 飯田 亮 P57

◆ 将軍を倒せ M2 伊藤 卓 P61

◆ 跳べ!水跳拳 M2 岩下 義明 P64

◆ 空手戦隊バカナンジャー M2 梅本 颯 P67

◆ X城危機一髪!! M2 白水 陽久 P70

◆ たこ殴り M2 鈴木 瞬也 P73

◆ 的あてゲーム M2 藤森 正悟 P75

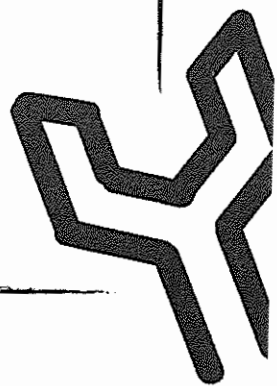
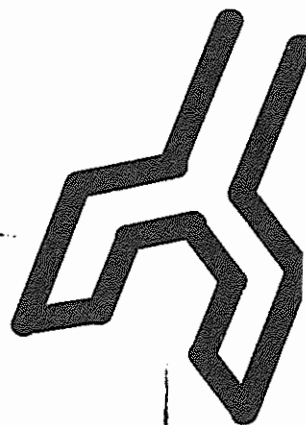
◆ いろいろ棒 M2 岩下 義明 P78
M2 伊藤 卓

§ 売り物 P82

◆ 千力千力 M3 棚辺 駿介 P83

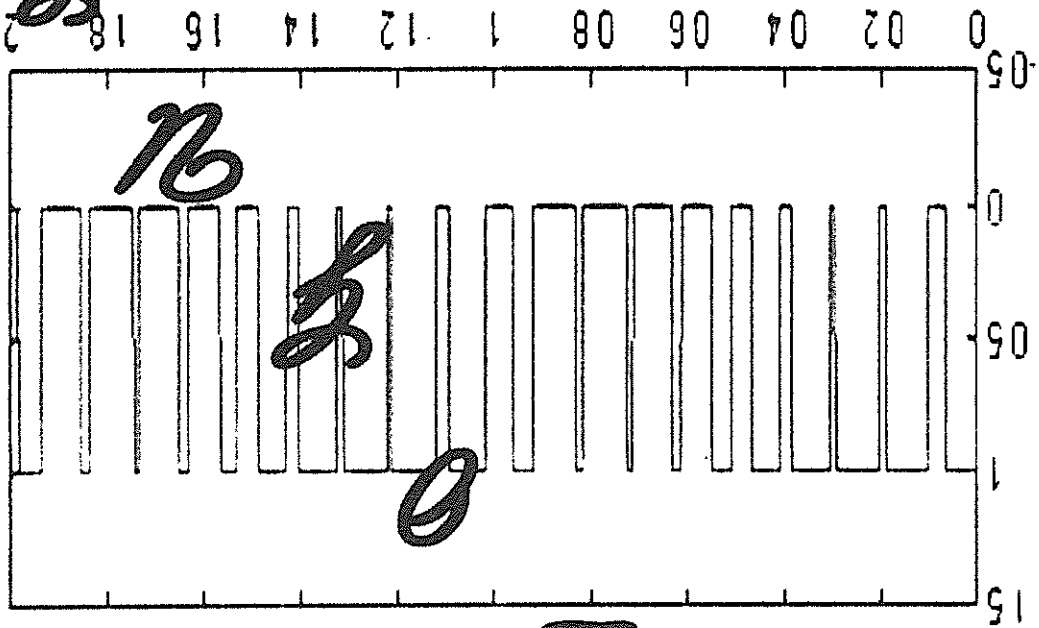
◆ Renda Game M3 中岡 勇太 P84

§ あとがき P87

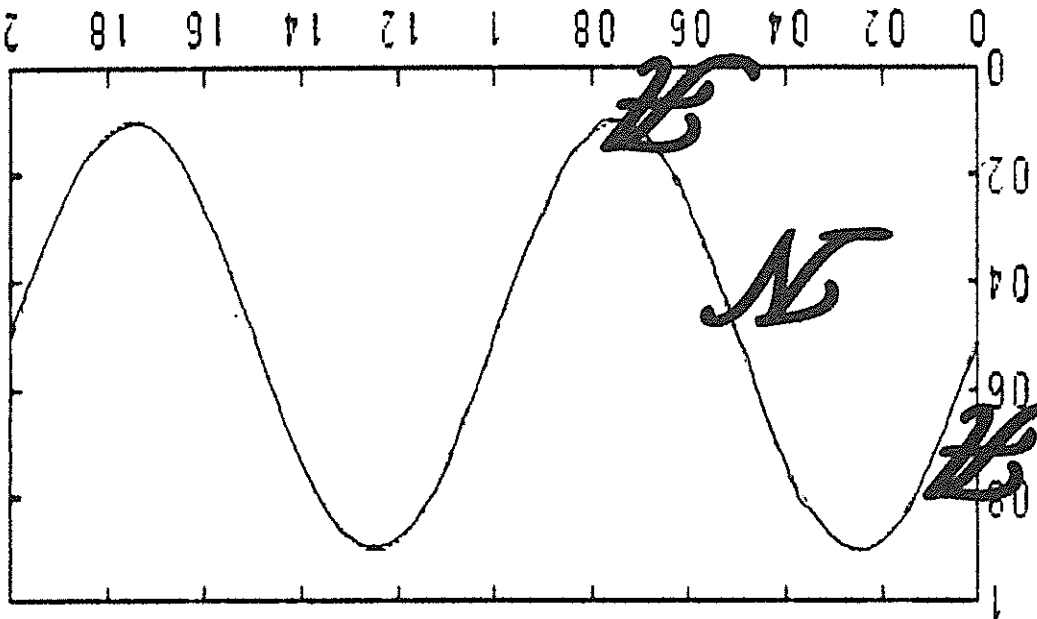


A 1Hz sine wave modulated with a 10Hz square wave

Time (s)



Pulse-Width Modulated Sine Wave



Source Signal Sine Wave of 1 Hz

タチバナ

第1話

～触ると感電!?
地味に危険な文化祭～

プロデューサー
物無の皆様

製作
H1-2 渡邊 翔

① サイトルについて

完全に後付け設定。このページがなければこんなことを考えることもなかたでしょう。ちなみ、第1回原稿にはここに和也くんがいらします。

② 製作者について

ハッキリいってかなりの変人。これ売ってる時はどんな格好で何をしてるか分かりません。去年の文化祭(2007年度)において、顔を白く(正確には Kiss のように)ぬって踊ってた人といえば、分かる人は分かるかも。

③ 製作物について

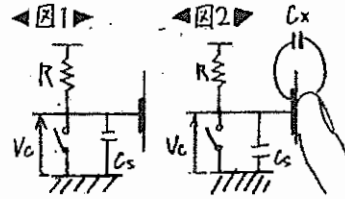
スバリ、今年、僕は一年間「タチバナ」を手がけてきました。あの、ゲームセンターにあるクイズゲームの液晶や、パソコンに使われているやつです。最近では携帯電話にもついているみたいですよ。Docomo & Sharp 乙。

① 詳しい仕組みについて

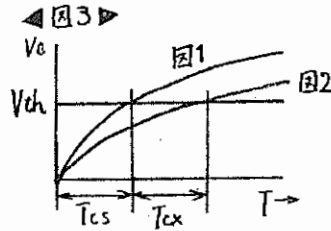
今回74LS151を作る上で使用したのは、静電式と呼ばれるもので、
5MHzに1トパソコンのも同じ原理で作られています。

静電式では、図1・2・3のように、指など導電体と触れることにより生
じる「静電容量の変化」を検出します。

まず、この回路構成では人体がアースされてい
ないと C_x の回路が形成されないはずですが、
実際には問題ありません。



人体のようにある程度表面積のある導電体はそれ単体で静電容量を持っています。
実際の値はESDの試験の人体モデルから100pF程度と見られ、これは C_x に比べて十分大きいと言えます。したがって、人体がアースされていなくても接地体と見なすことができます。



② ハードウェアについて

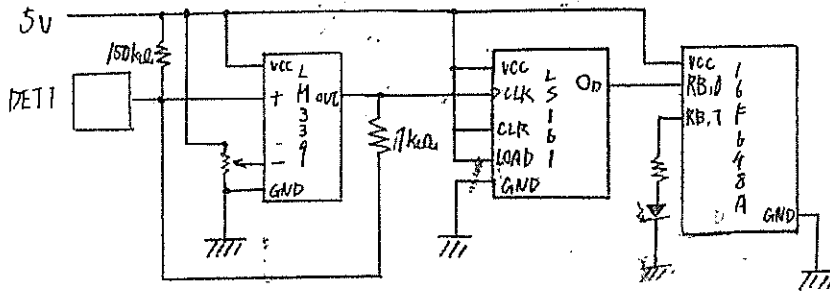
検出電極は、銅板とアルミ板の上下をボルトで固定し、さらにこれと回路のある基板へと固定した単純なものです。マイコンはPIC16F648Eを使い、74LS151の抵抗は今のところ150kΩです。銅板の量が増えた時は74LS151で制御します。

③ 判定方法について

次頁の回路図を参照して下さい。銅板からは、コンパレータの抵抗による発振信号がカウンタ(74LS161)に入ります。元の時の信号を迅速にPICで読みとるのが不可能なので(およそ5MHzくらい)、カウンタの2進出力の4ヶ月、74LS151にこれらの信号をPICでカウントします。その回数によって

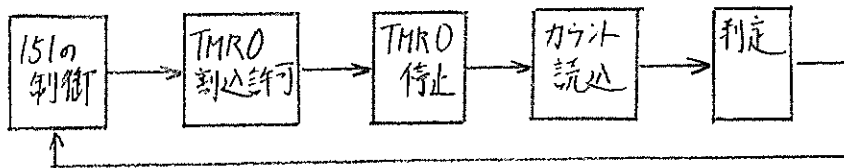
触れているかを判定する回路、これは超簡単なものです。

⑦ 回路図について



⑧ プログラムについて

ま、「判定方法について」で書きましたけど、あのままです。でも、項目は作れたのでフローチャートに付載せよう。一本道です。

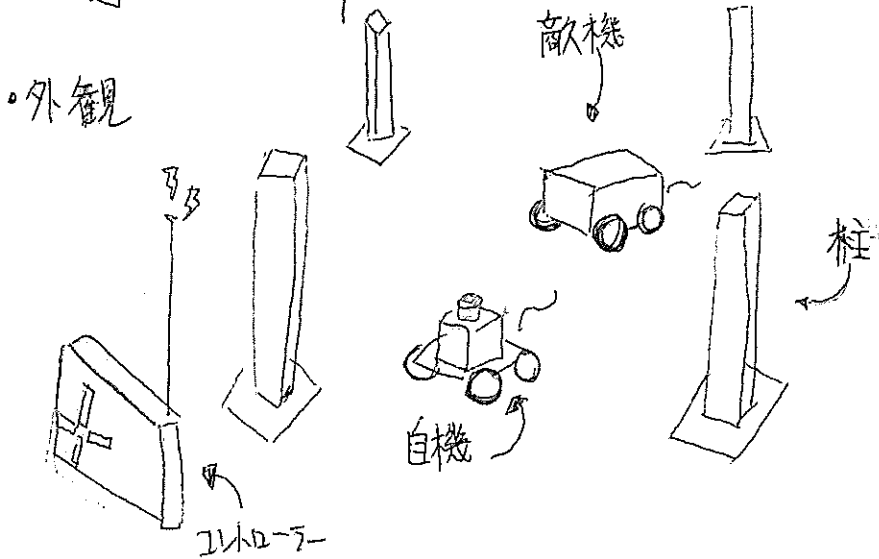


⑨ 感想について

うん、完全初級近いです。いんいん！実は今回かいたこのシステムの前に数回進路変更があったので、数ヶ月はそれに消滅した。なにか手書きの量がある。しかしそれにしては、何と美しい文字、むしろ字の人間は美しい心を持つ。それはそれで、文化祭ではキボトヤとされているように。

ガクが所、竹島氏とあるHPの方が大変お世話になりました。お二方の提案によって作られているよりのものです。どうもありがとうございました。

STEP 逃走中



内容

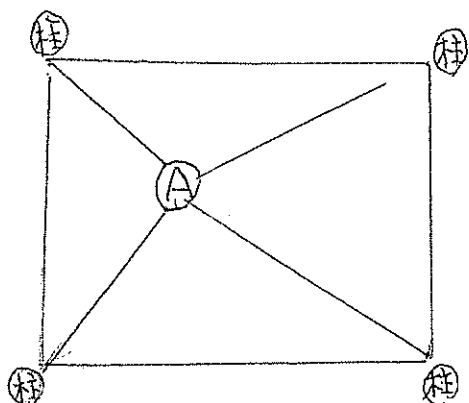
簡単に説明すると追いかけてくる敵から逃げるゲームです。
4つの周りの柱には超音波受信機がついており、
自機から発射された超音波を受信し、4柱の時差で場所を
特定し、そこへ敵が自動的に進み、自機に触れると終わりです。
自機の制御は無線により、コントローラで行います。

超音波とは、人間の可聴域を超えた波長であり、
形としては音と同じです。

無線といえどもトランシーバーなどのイメージもありますが
ラジコやラジオにも作られる電波の1つである。やはりこれも
波長で出来ている。波長を変えるとモーターに違う信号が
入り左右に曲ったりすることが出来る。ラジコと同じです。

。位置特定

敵が自機の位置を特定するためは何をしているかと言うと、三角法を使っています。三角法とは、三角形を利用して物体との距離を計ります。



図を用いると、Aが柱へ向かって超音波を送ります。すると、4つの柱は超音波が到着するに差がつかず。その差を計算すると三角法により、位置特定が出来ます。

その情報を敵機へ送ると、自動的にその位置へ移動します。

これを繰り返すことにより、自機を追いかけることが可能です。

。三角法

知っている人は知っていると思いますが、

あるA点からB点までの距離を知りたい時に、

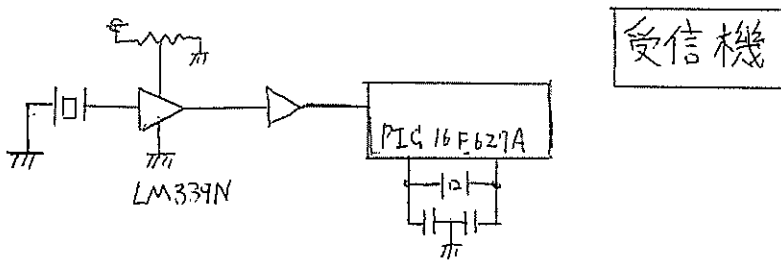
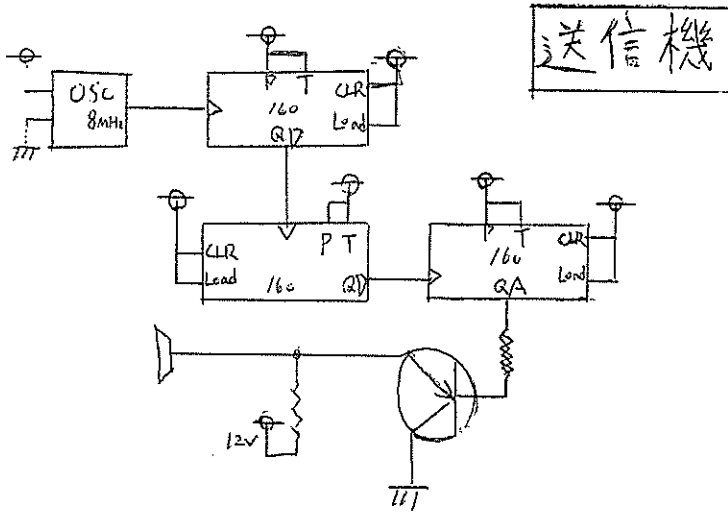
三角形ABCを作り、それと相似三角形DEFを作る。

すると、DEFの辺の長さが分かれば、ABCの長さも分かる

というものです。

現在は未完動中。目指せ完動中!!

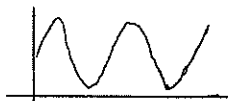
○ 回路 (超音波のみ)



○ 説明

・送信機

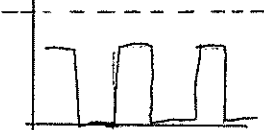
8MHzの発振を74LS160において $\frac{1}{200}$ にして、40kHzまで落とします。それでトランジスタと12Vを大きくし、送信機に入れると超音波が出ます。



(図1)

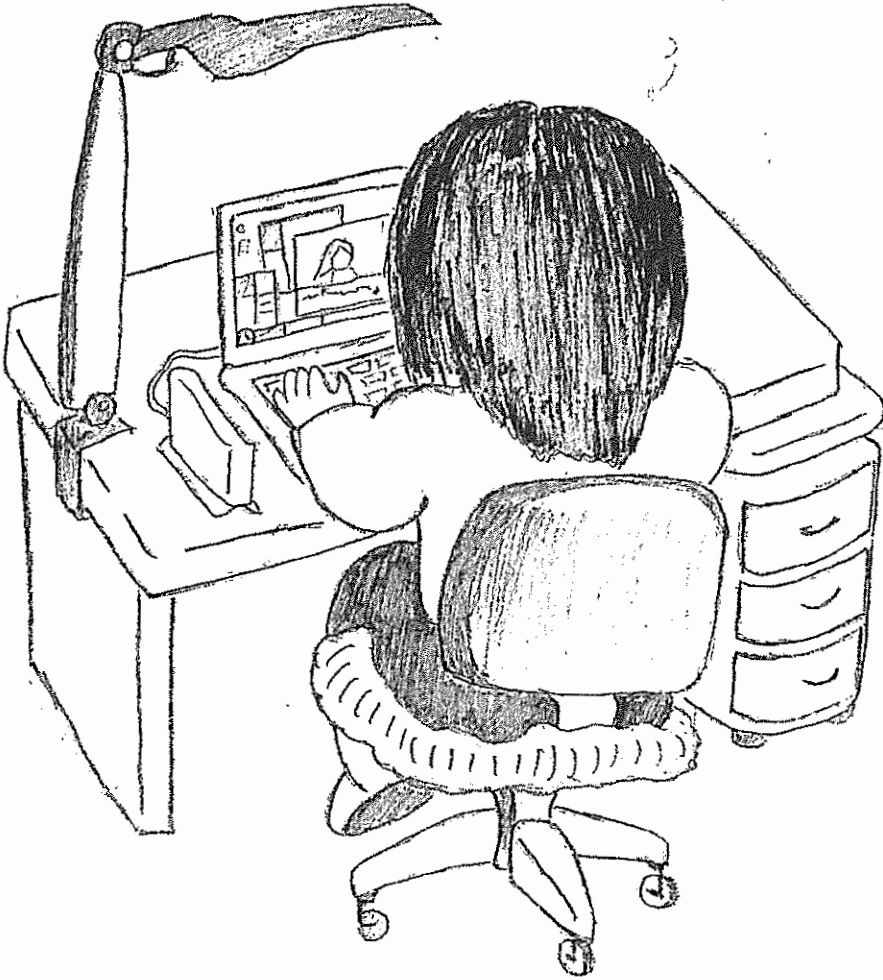
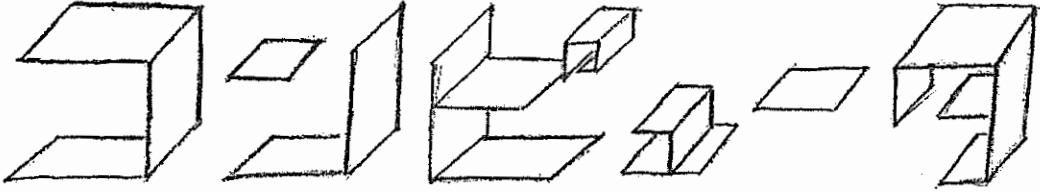
・受信機

送信機から出る超音波は(図1)のように入ります。それをLM339Nというコンパレータに入れると(図2)のように制御されます。すると、マイコンPICが読みとってくれます。



(図2)

お付き合い有難うございました。



AZABAC-61^{Beta}

製作

M3 池上

H2 川崎

協力

物理部無線班の方々

・概略

まず始めに、このプロジェクトが何をやろうとしているのかを説明したいと思います。

例えば、電子回路から液晶へと文字や絵を表示させたいと思っ
たとします。確かに、秋葉原などには液晶表示器というものが売っ
ています。でも、それらの液晶には文字や絵を表示する仕組みは付
いていないのです。ですから、それを簡単にやってしまえる仕組み
を自分たちで作ろう、というのがこの企画なのです。

・実際の製作

このプロジェクトは大きく2つに大別され、1つは液晶を裸のま
まより制御しやすくするための回路、2つ目は文字や絵を記録して
表示するための回路となります。

ここからは、様々な仕組みなどをその2つに分けて紹介したい
と思います。

```

switch(level){
case 0:
break;
case 1:
if(score < score_level_1){
game_over();
}else{
level++;
}
break;
case 2:
if(score < score_level_2){
game_over();
}else{
level++;
}
break;
case 3:
if(score < score_level_3){
game_over();
}else{
level++;
}
break;
}

```

H8/3664を

使用した

液晶の

表示

川口 池上

ここでは、H8/3664というマイコンを使用して液晶への表示を実現させるという技術について説明したいと思います。

使用するもの

- ・マイコン (マイクロコンピュータ)

マイコンというのは、回路に直接組み込めるコンピュータのことですが、ここでは基本的な機能のみ付いた小さなコンピュータとして使います。ほかの製作などでも多く使われているPICマイコンは使わず、ここではより性能の高いH8を使うことにしています。なお、実際に使用したのは、秋月電子通商の「AKI-H8/3664BPタイニーマイコンボード」という製品です。

- ・キーボード (PS/2)

最近はUSBに取って代われようとしています、昔のPCでは主流だったキーボードです。規格などが単純なので、液晶に文字を入力するためのデバイスとして使います。

まず、何をさせるのか

液晶とH8は、8ピンでつながっており、すべてH8側からの出力となっています。ピンの内訳は、同期クロック、クリア、赤色データビット0,1、緑色データビット0,1、青色データビット0,1

となっています。詳しいことは液晶の説明の方を見てみると分かるかと思います。

描画は、図2のように左上から右方向へと1ドットずつ行っています。

—ピンの内容—

- ・同期クロック

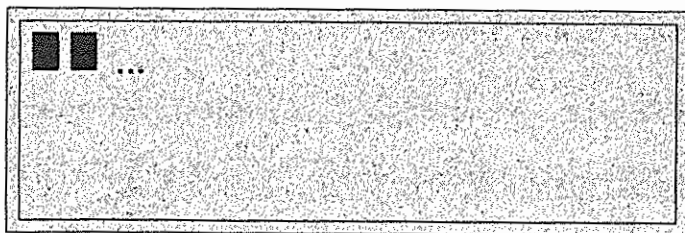
このクロックを入れると、液晶側がデータビットを参照して1ドットを描画します。

- ・クリア

これを入れると、液晶画面の表示内容がクリアされます。

- ・データビット

ドットの赤・緑・青のデータです。



↑描画の順番

H8/OSというもの

H8/OSというものが、SourceForgeにて配布されています。これは複数の種類のH8で共通に使えて、様々なデバイスの使用を簡単にすることの出来るC言語のライブラリです。

この製作でもこれを使ってプログラムを書いております、キーボードの使用などにも使用しています。

また、ソースコードのコンパイルにはGCCという様々なPC環境で使えるC、C++、Objective-Cコンパイラを使用しています。H8は、Windows上だけでなく、Mac OS XやLinuxの上でも開発が可能なすごい代物でもあるのです。

感想

前年は、簡単なゲーム系のもので作ったので、今年の製作の最初のうちはコンピューターの右も左も分からず大変でした。また**H8**の活用技術は手探り状態で、ネット上の情報が役に立ちました。いろいろと助けてくださった先輩の方々に、ネットに情報を公開されている方々に深く感謝致します。

また、今回の製作はコンピュータ系だったのでプログラム中心でしたが、今度はロボット系のもので作っても良いかな、とも思います。ずっとパソコンの前にも飽きるので。

カラー液晶制御技術

H2 川崎

さて、ここでは文字データや画像などを表示させる液晶の表示・制御方法を書いていきたいと思います。

まず、軽くこの液晶技術の概要を述べときたいと思います。今回の目標としていているものは、“SRAM”と呼ばれているデータの記憶領域に、1ドットごとの色データをH8から入れていき、それを他のマイコンからそのデータを読み出して液晶に出力するというものです。

1. 液晶表示に必要な物

①液晶本体

今回使った液晶は「LTA042B010F」というもので、縦96ドット・横400ドット、色は1色につき6ビットで26万色まで再現することが可能です。この液晶は秋葉原の秋月という店で、300円で売っています。何でそんなに安いのかというと、メーカーに問い合わせても仕様書などをもらうことが出来ないらしく、自分で調べなければいけないかららしいです。この液晶の使用については後のページで説明をしていきます。

②バックライト

まずバックライトとは何かというと、その名のとおり、液晶の後ろ側から画面を明るく照らすものです。バックライトが無ければ、初代ゲームボーイのようになってしまい、親の目から逃れるために布団の中で遊ぶことが出来なくなってしまいます。

この液晶には制御ピンとは別にバックライトを収容するスペースがあり、買った時からキチンと入っていました。このバックライトを光らせるには何百Vと必要になってきます。そのために秋月で「極細冷陰極管+小型インバータセット」を買ってそれについているインバータを使います。インバータとは何かというと、直流電力から交流電力を電氣的に生成する回路のことをさします。先ほども言ったとおり、何百Vも出ているため、小さいくせに触ると感電して激痛が走るので生憎な代物です(笑)

③マイコン

マイクロコンピュータの略ですね。今回はどのようなマイコンを使うのかというと、「PIC16F877A」というものです。このPICはレジスタ、ピン数ともに多いので回路やプログラム設計をするとき便利です。ちなみに発振子は20MHzのものを使用しています。ではこのマイコンでどのような作業をするのかというと、液晶への入力信号です。この液晶には複雑で高速な信号を送らなければいけないのでプログラムを書くのは非常に面倒でした。

④SRAM

SRAMとは何かというと、データの読み出しや書き込みが簡単に出来るデータの記憶領域のことを指します。ではこれで何をするのかというと、簡単に言うとパソコンのビデオメモリの役割を果たしています。つまり、ここに表示させたいデータを入れて液晶に出力するのです。そこでどのようにデータを入れていくのかというと、液晶の左上のドットから順に18ビットの色データを外部のマイコンで入れていき、その保存したデータを、こちら側のマイコン (PIC16F877A) で順番に出力していきます。その作業を無限ループでやっていきます。次にどれくらいのデータが必要になってくるかを計算すると・・・

96 (縦) × 400 (横) = 38400ドット になります。
今回は18ビットという非常に大きい数値を使いたくなかったため、1色につき2ビットの計6ビット (64色) を使うことにしました。よって容量が38400byte以上あるRAMを使えば大丈夫です。今回も毎度お馴染みの秋葉原の秋月という店で「SRAM2B2565SLMX」という256kbitの容量があるRAMを使うことにしました。

II. 液晶の仕様

ここでは液晶をどのようにしたら動かすことが出来るといったような、液晶本体の説明をしていきたいと思えます。

まずは、液晶のピン配置とピンの説明から。

Pin No	記号	機能	備考
1	TEST	テストパターン表示	Lで入力有効
2	NCLK	サンプリングクロック	制御タイミングのベース
3	GND	GND	0V
4	HSYNC	水平同期信号	横ドット制御
5	VSYNC	垂直同期信号	縦ドット制御
6	不明	不明	一応L固定
7	GND	GND	0V
8	B5(MSB)	青表示データ(MSB)	MSB LSB 111111 明 111110 ↑ ↓ 000001 ↓ 000000 暗
9	B4	青表示データ	
10	B3	青表示データ	
11	B2	青表示データ	
12	B1	青表示データ	
13	B0(LSB)	青表示データ(LSB)	
14	GND	GND	0V
15	G5(MSB)	緑表示データ(MSB)	MSB LSB 111111 明 111110 ↑ ↓ 000001 ↓ 000000 暗
16	G4	緑表示データ	
17	G3	緑表示データ	
18	G2	緑表示データ	
19	G1	緑表示データ	
20	G0(LSB)	緑表示データ(LSB)	
21	DVDD	ロジック用電源	5V
22	R5(MSB)	赤表示データ(MSB)	MSB LSB 111111 明 111110 ↑ ↓ 000001 ↓ 000000 暗
23	R4	赤表示データ	
24	R3	赤表示データ	
25	R2	赤表示データ	
26	R1	赤表示データ	
27	R0(LSB)	赤表示データ(LSB)	
28	AVDD	アナログ用電源	5V
29	Vo	コントラスト微調整	50KΩ程度の可変抵抗
30	GND	GND	0V
31	GVDD	LCDゲートドライバロジック用電源	+5V
32	GND	GND	0V
33	VSS	LCDゲート負電源	-9V
34	GND	GND	0V

次にピンの説明をしていききたいと思います。

①B0～B5・G0～G5・R0～R5

図にあるとおり、1色につき6bitずつ用意されていて、数値が高ければ高いほど、明るい色を出すようになっています。例えば、赤色を表示させたい場合は、「000000000000111111」となります。また白色を表示させたい場合は「111111111111111111」黒色を表示させたい場合は、「000000000000000000」と入力すれば大丈夫です。

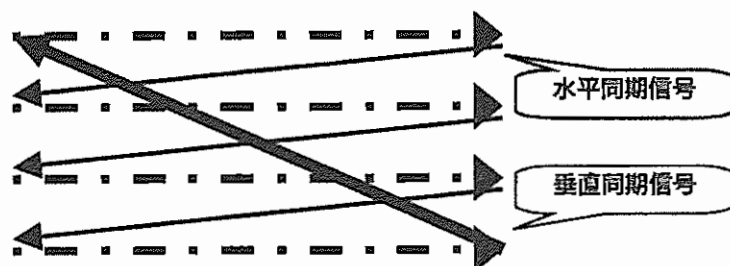
②NCLK・HSYNC・VSYNC

これら3つの入力が液晶制御の要となります。まず3つのピンについて説明を。

- ・XCLK・・・この端子電圧の立下りエッジで水平同期信号・垂直同期信号・色データが採取されます。それから水平方向のアドレスがインクリメント(+1)されます。
- ・HSYNC・・・323クロック以上入れた後、このピンをHにすると水平方向のアドレスがリセットされ、水平方向のアドレスがインクリメントされます。
- ・VSYNC・・・この端子をHにして153クロック以上入れると、垂直方向のアドレスと水平方向のアドレスがリセットされます。

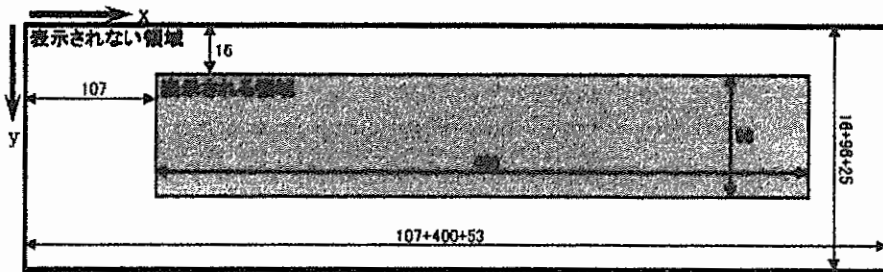
～画像入力の基礎～

向かって右方向をX、下方向をYとします。左上(0,0)からX方向に1画素ずつ順にCLKの立ち下がり画素値を入力していき、右端まで入力できたら、左端の一つ下(0,1)にいきます。またそこからX方向に1画素ずつ画素値を入力していき、下端まで入力することが出来たら左上(0,0)に戻って再び、X方向に画素値を入力するということを繰り返すことで、液晶を表示していきます。



～同期～

まず同期信号を入れない場合、次の図のようになります。水平同期信号を入れなくとも（ $107+400+53$ ）クロック毎にXアドレスが勝手にリセットされ、Yアドレスは勝手にインクリメントされます。垂直同期信号を入れなくとも、 $Y=(107+400+53)+1$ になったらXアドレスとYアドレスが勝手にリセットされます。このやり方でも表示をすることは出来ますが、アドレスの初期値が不明のため、画像がずれて表示されてしまいます。

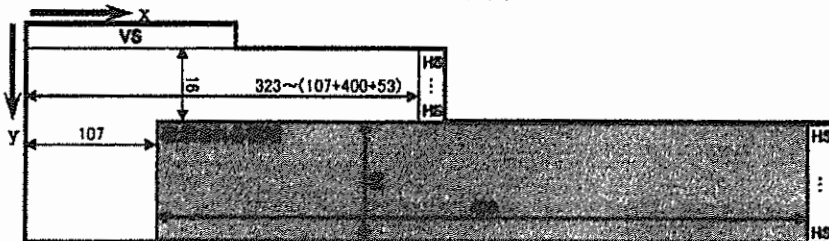


次に同期信号を入れる場合、下図のようになります。同期信号を入れると、アドレスがリセットされるので同期が確実にとれます。

- ・水平同期信号はXアドレスをリセットして、Yアドレスをインクリメントします。

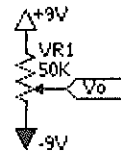
- ・垂直同期信号はXアドレスとYアドレスをリセットします。

また表示される領域の最後の画素値を入力したらすぐに同期信号を入れてしまうようにすれば、入力するクロック数を減らしてほんの少し、フレームレートをあげることが出来ます。

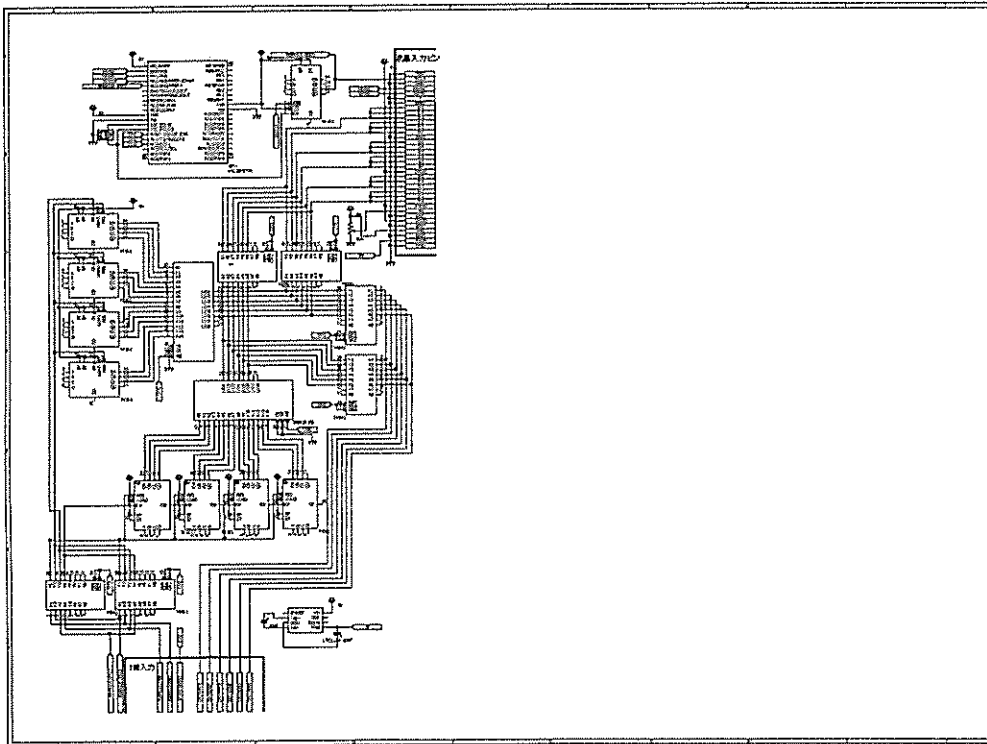


③Vo

右図のようにアナログな電圧を入力して、コントラストを調整していきます。50kΩの半固定抵抗を使えば大丈夫だと思います。±9Vの電圧を使うのがベストですが、9Vの電流を使うのがもったいなければ、5V-GND間で生成してもよいかもしれません。大体2.2Vくらいの電圧がきれいに写るとおもいます。他のネットの資料によれば、L固定でも一応平気らしいです。



III. 回路図



まあ回路図はこんな感じになりそうです。では何箇所か説明をしていきます。

①74161 (液晶入力用)

この161は何の為にあるのかというと、20M発振を8倍に遅くしてNCLKに入れるためです。このことは1サイクル(コンピュータの命令単位)が4クロックのPICに対して、2サイクルがNCLKへの1CLKになるので、色データの読み出しが簡単に出来ます。

②-9V

-9Vなんて電圧をどのように作るかというと、秋月に頼ります。秋月には「LTC1144」というICが売っていて、何をやるものかということ、入力電圧をそのまま負極性にするものです。これを使えば9V電源を用意するだけで大丈夫です。ちなみに僕は12V電源を使いましたが余裕で動きました。

IV. プログラム

最後にプログラム例を載せたいと思います。先ほど書いたとおりにHSYNC・VSYNCをH・Lさせることが出来れば誰でも表示できると思います。今回は青・緑色に2分割するだけのプログラムを載せることにします。

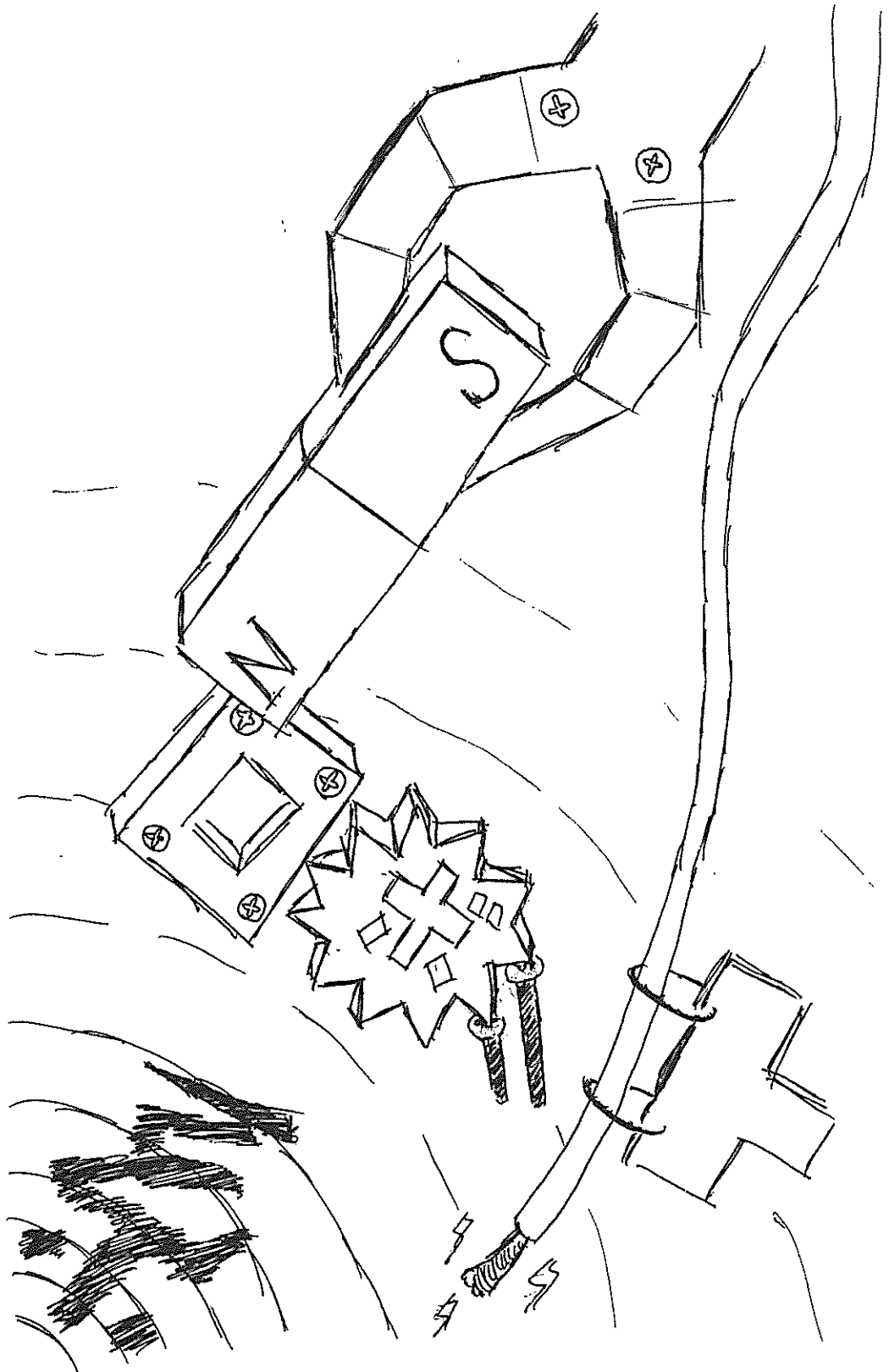
<pre>LIST P=PIC16F877A INCLUDE P16F877A.INC ORG 0 GOTO SETUP SETUP COUNT1 EQU 20H COUNT2 EQU 21H COUNT3 EQU 22H BCF ADCON0,0 BCF STATUS,RP1 BSF STATUS,RP0 CLRF TRISA CLRF TRISB CLRF TRISC CLRF TRISD CLRF TRISE MOVLW B'00000111' MOVWF ADCON1 BCF STATUS,RP0 MOVLW B'00000100' ;VSYNCを L、HSYNCをL、DCKを開始 MOVWF PORTA MAIN</pre>	<pre>MOVLW D'016' MOVWF COUNT2 MOVLW D'096' MOVWF COUNT3 ;****(0,0)**** BSF PORTA,0 MOVLW D'100' MOVWF COUNT1 DECFSZ COUNT1 GOTO \$-1 NOP NOP NOP NOP NOP NOP BCF PORTA,0 ;****(0,+1)**** BCF PORTA,1 ;HSYNCをL にする。 MOVLW D'215' MOVWF COUNT1 DECFSZ COUNT1</pre>
---	---

GOTO \$-1	NOP
NOP	NOP
BSF PORTA,1 ;HSYNCをHにする。	NOP
DECFSZ COUNT2	MOVLW B'00000010'
GOTO \$-8	MOVWF PORTC
NOP	;**200DCK**
MAIN2	MOVLW D'131'
BCF PORTA,1 ;↑4サイクル	MOVWF COUNT1
;****107DCK****	DECFSZ COUNT1
MOVLW D'69'	GOTO \$-1
MOVWF COUNT1	NOP
DECFSZ COUNT1	NOP
GOTO \$-1	NOP
MOVLW B'00000001'	NOP
MOVWF PORTC	BSF PORTA,1 ;HSYNCをHにする。
;****400DCK****	DECFSZ COUNT3
;**200DCK**	GOTO MAIN2
MOVLW D'131'	BCF PORTA,1 ;HSYNCをLにする。
MOVWF COUNT1	GOTO MAIN
DECFSZ COUNT1	
GOTO \$-1	
NOP	END

感想

今回の製作物は今まで4年間の中で一番楽しかったですね。その要因としては池上君と共同制作をしたからかもしれません。二人で同じ目標を持ってやるって言うのはやっぱり、一人で製作すること断然面白いですね。一回は共同制作を試みることをお勧めしますよ。

液晶技術も今現在完成とはいえないところにいるので、文化祭までに完璧に引き継げるとこまでいきたいと思います。将来中1のゲームがこの液晶で動いていると信じています(笑)。まあ最後に、製作物名を'AZABAC'にしましたが今のままではOBの人の怒られてしまうかもしれないので、今後胸を張って'AZABAC'といえるものが出来ていることを祈っているので、皆さんこれからも頑張ってください。



超電磁式鈍速列車

製作者 H2 中嶋剛大

H1 西村陽輝

M3 竹下 諒

協力 物無の皆様、OBの皆様

1. はじめに

この製作物は、リニアモーターカー (Linear Motor Car)です。

リニアモーターカーと言う単語は誰もが一度は耳にしたことがあると思います。何と言うか、次世代っぽい響きでカッコいいですね。

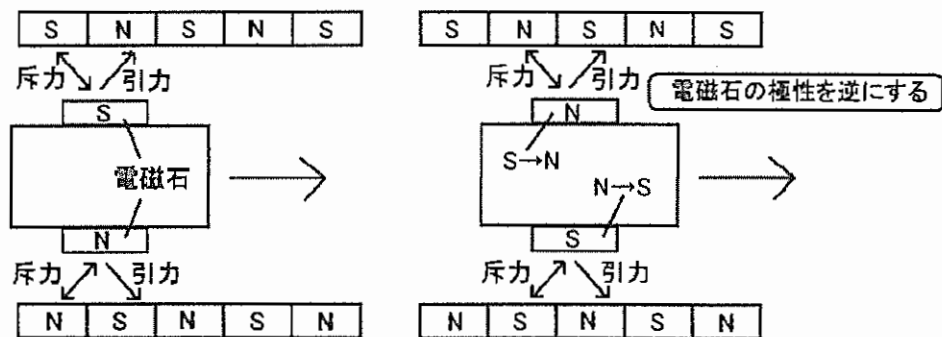
僕たちもその響きに憧れ、リニアモーターカーの製作をすることに決めました。

電子工作経験者で無い方には、わかりづらい所もあるとは思いますが、リニアモーターカーの魅力を少しでも知って貰えたら非常に嬉しく思います。

2. What is the Linear Motor Car?

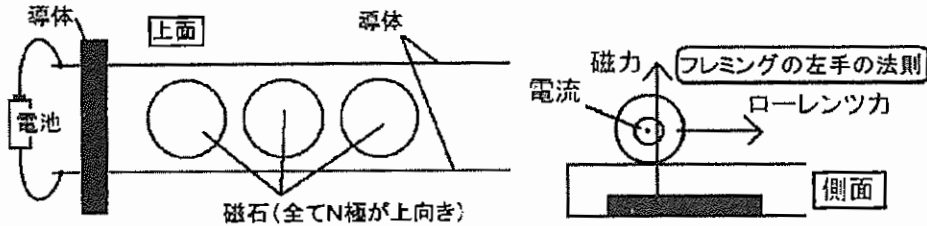
① リニアモーターカーの仕組み

リニアモーターの仕組みは「普通のモーターの中身を広げたもの」と良く表現されます。詳しくは下図をご覧ください。(磁力が反発するときの力を斥力といいます。)



電磁石のS極(N極)とレールのN極(S極)がくっついて、ストップする前に、電磁石の極性を逆向きにする(電磁石に流す電気を逆向きにする)ことで前に進み続けると言うものです。正にモーターと同じ動作です。実は単純なんですね～。ちなみに、このタイプのリニアモーターは同期型と呼ばれます。僕たちが使用したのはこのタイプのリニアモーターです。

同期型のリニアモーター以外にも、ローレンツ力(フレミングの左手の法則で出てくる“カ”の事)を使った誘導型というものもあります。(下図の通りの物は火花が出るので注意)



大雑把ですが上図ようになっており、ローレンツ力で進むリニアモーターのことです。世の中のリニアモーターカーは金銭的な面からこのタイプのリニアモーターを使っているものが多いです。(上図とは異なり、車体側に磁石がついている。)

—— コラム : リニアモーターのメリットは? ——

- ①「非常に薄く、省スペース」
- ②「鉄道に使う場合、タイヤの摩擦に頼る必要がなく、勾配やカーブに強い」
- ③「ギアボックスと違って修理頻度が少なく、維持が楽」

以上のように、沢山のメリットをもっています。

しかし、デメリットとして、「同じパワーを出す場合、従来のモーターに比べて電気の消費量が格段に多い」という特徴を持っています。温暖化防止には向いていませんね…。

② リニアモーターカーの種類

続いて、リニアモーターカーの種類です。浮上型と車輪型の2種類に大別されます。

< 浮上型 >

その名前のとおり、磁力で浮上するタイプで、JRマグレブや愛知県のリニモなどが有名です。すぐ頭に浮かぶのはこのタイプのものだと思います。

車輪による摩擦がなくなり、とても早いスピードで走る事ができますが、浮上させるのは難しく技術的問題が残っています。

飛行機と新幹線の中間に位置する速さを持ち、騒音や振動が少ないというメリットがあります。

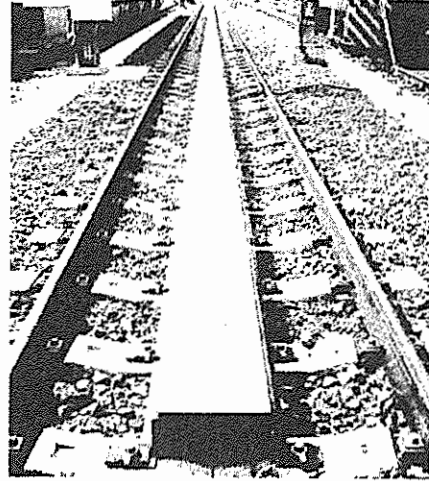


＜車輪型＞

動力にリニアモーターを使い、通常のレールと車輪によって走行するタイプで、主に地下鉄で使われています。身近なところで言うと、都営大江戸線がこのタイプです。

レール間にリアクションプレートという金属の板がおいてあります。(右図参照)

麻布十番からお越しの方は、ぜひ帰りにレール間を覗いてみてください。



3. 本体製作について

ここでは、車体とレールの両方を合わせて本体と呼んでいます。(車体とレール、両方の磁石を使って進むので、当然といえば当然ですね。)

全て、同期型リニアモーターを使用した車輪型のリニアモーターとなっています。

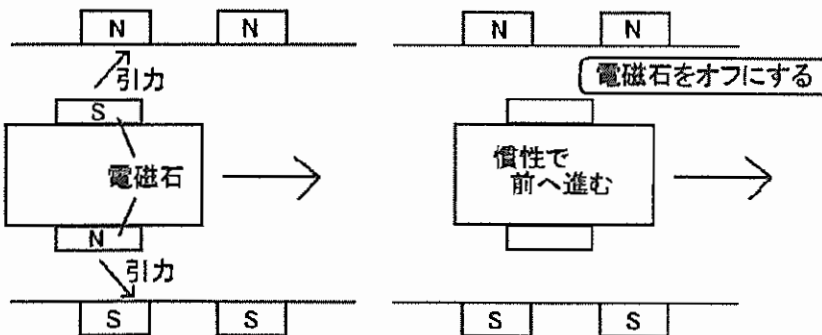
1号機 (レール:電磁石 車体:永久磁石)

レール側が電磁石であるため、加速と減速が自由自在。しかし、電磁石の同時制御数が40個を越え、制御が面倒になったため断念。

2号機 (レール:永久磁石 車体:電磁石)

電磁石が1つになったため、制御が大幅に簡単になった。しかし、斥力が弱く、働いていない時と変わらないため、実質的に下図のように移動している。

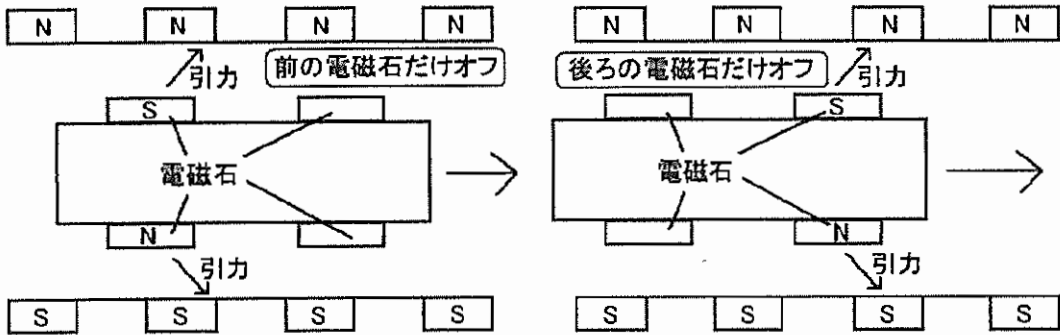
タイヤの摩擦や加速性能の低さによって、慣性による移動中に止まってしまうため、ポツになった。



3号機（レール:永久磁石 車体:電磁石）

2号機の電磁石を二つに増やしたことで、慣性のみによる移動時間をゼロにした。(下図参照)

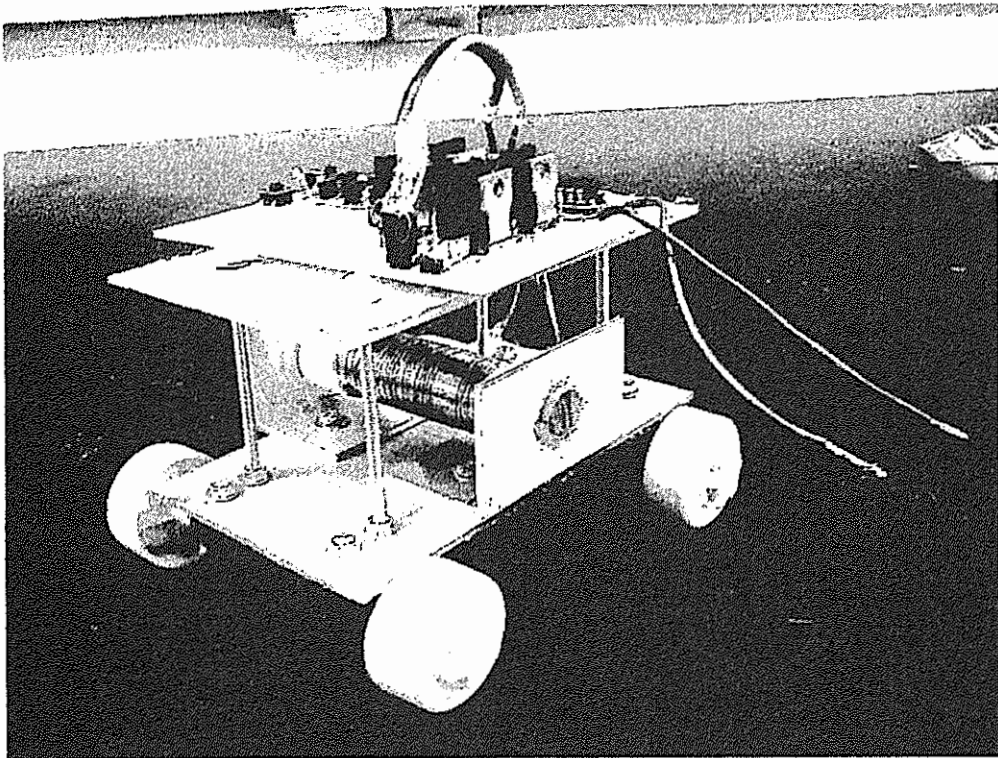
また、タイヤをベアリングに変更したことで、摩擦を大幅に減らし、途中で止まってしまうことの無いようにした。



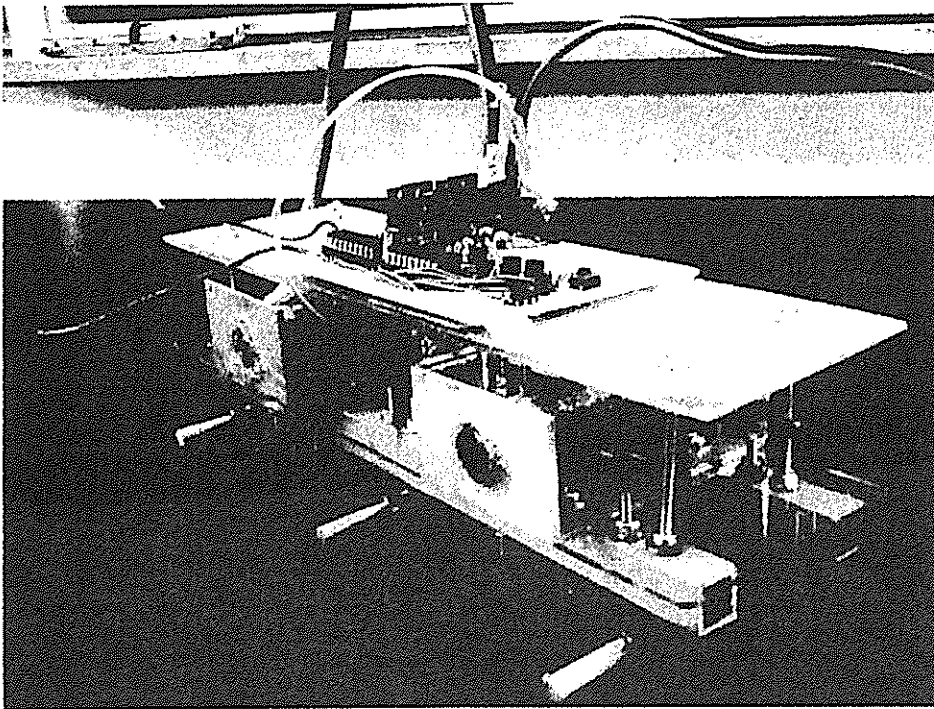
車体の写真

車体の写真を載せます。見つらかったら申し訳ありません。

2号機



3号機



以上、H2 中嶋がお送りしました！

レール（ここからはH1の西村がお送りします。）

ではニアの密かな要であるレールについて説明します。

電磁石の極を入れ替えるためには、車体の現在地を知る事が必要になるため、高輝度LEDをレールの側面の電磁石切り替えポイントに設置しています。

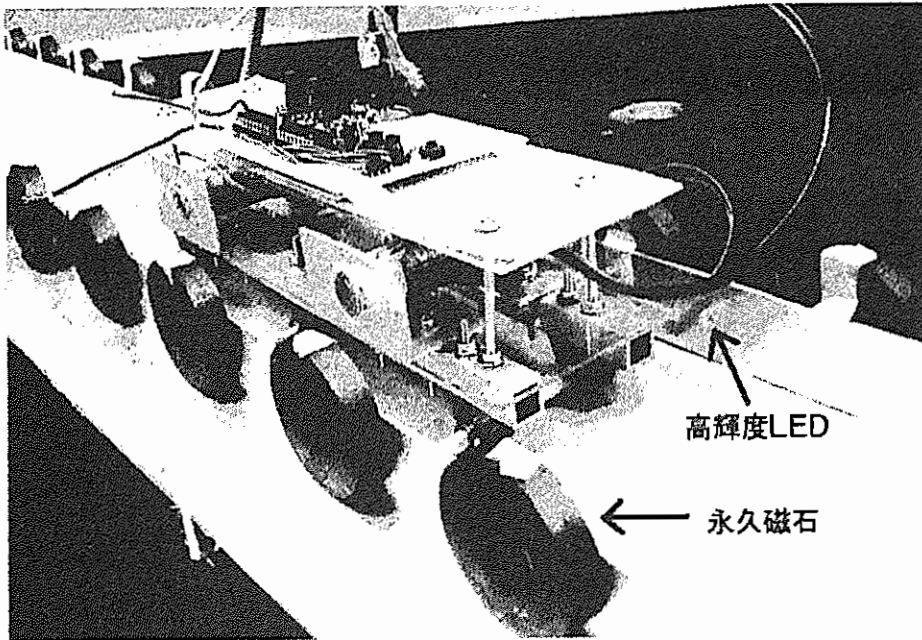
その光を車体の側面のフォトランジスタで受光する事でPICの信号が変化し、電磁石を切り替えます。

レールは、一定間隔でアルミのL字材に磁石とLEDを貼り付けただけの、シンプルなものとなっています。(LEDがついているのは片側のレールのみ)

また、車体は浮上せずタイヤ(ベアリングで代用)で走るため、レールに浮上するための仕組みはついていません。

レールの幅などがずれてしまわないよう、アクリル板と口の字材を使ってしっかりと固定しています。(次ページ写真参照)

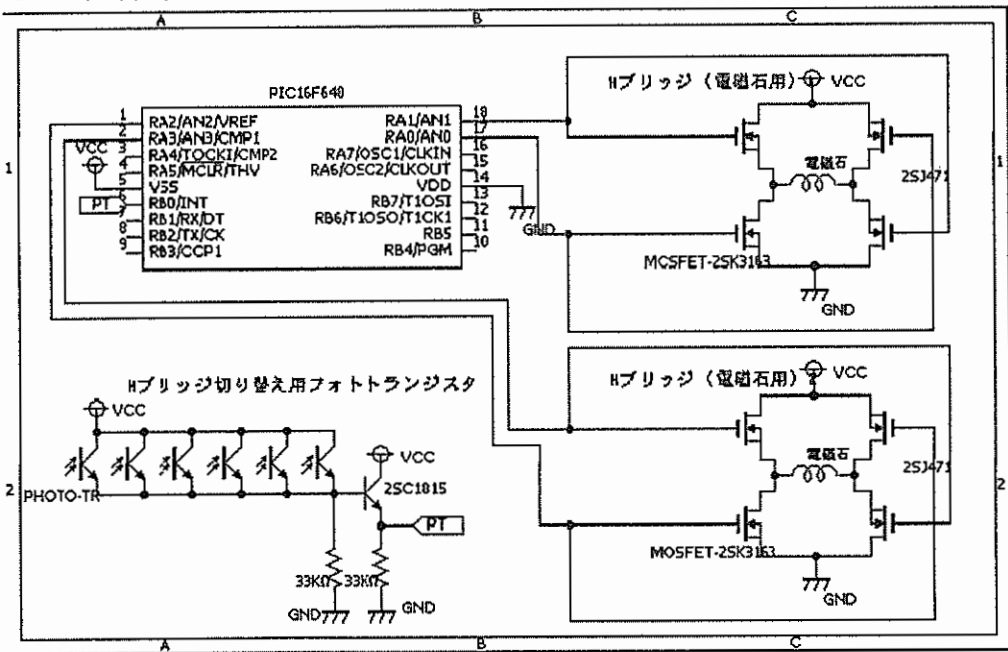
レールの写真



4. 回路について

① 回路図

ではまず今回使用した回路の回路図をご紹介します。



② 回路の構成について

・Hブリッジ

Hブリッジとは、一言で言えば「単一の電源でモータに加える電圧の向きを変えられる回路」の事です。回路図で見るとHの形をしていることから「Hブリッジ」と呼ばれています。

ではHブリッジの基本動作について説明します。まず左上と右下の FET だけを同時に On とすると、図の左から右に電流が流れ、電磁石が磁力を発生します。逆に右上と左下だけを On とすれば、図の右から左に電流が流れ、電磁石の磁極が反転します。仕組みも動作もそう難しくはありませんが、右上と右下、または左上と左下を同時に On にしてしまうと、ショートして FET が壊れてしまうため、その制御はそのようなことの無いようにしなくてはなりません。

現在は FET に全て P チャンネルである 2SK3163 を使っていますが、Hブリッジでは VCC につなぐ FET の 2 つは N チャンネルである 2SJ シリーズを使ったほうが、よりパワーが出るそうなので、2SJ シリーズを使おうかなとも思っています。

・フォトトランジスタ

フォトトランジスタとは、光信号によって電流を制御するトランジスタです。光信号はある程度の強さを持つ必要があります。また、指向性(光を感知する角度)が種類によって決まっています。

今回は、リニアモーターカーが揺れてもレール上に付けた高輝度 LED の光を感知しやすいように指向性が低い(光を感知する角度が大きい)ものを使用しています。また 7 個もつけているのは、円状に並べることで揺れても光を感知しやすくするためです。

フォトトランジスタとトランジスタは直接 5V をかけると壊れてしまう恐れがあるため、フォトトランジスタの出力とトランジスタのコレクタの所にそれぞれ 33K Ω をつけています。

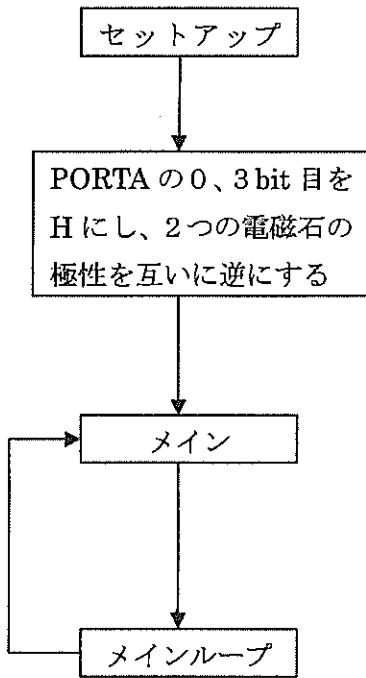
回路の動作については、「プログラム」の章でまとめて説明します。

5. リニアのプログラム

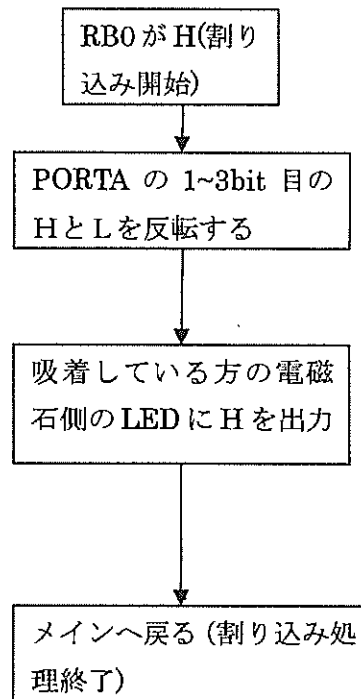
① プログラムと回路の動作目標

この回路の最終動作目標は、『PWMによって(コイルの抵抗値の低さによる)FETの発熱を防ぎ、PICがフォトトラの出力を認識すること(磁石の位置に来る度に)、「Hブリッジが On(引力)⇒一定期間 Off(惰性走行)⇒Hブリッジ On(磁極が逆転し斥力)』となり、電磁石と回路を取り付けた本体が前進することです。ではプログラムです。

メイン、セットアップルーチン



INT 割り込みルーチン



現在、上のプログラムで PIC は、フォトトラの入力があるごとに（本体が切り替えポイントに来るごとに）、PORTA が上のような動作を繰り返しています。

これによってどのような動作が起きるかという、最初は前の電磁石がレールの永久磁石に吸着するように本体が前に進みます。そしてフォトトラの所まで来ると今度は後ろの電磁石が永久磁石に吸着しようとして、さらに前に進み、もう 1 度フォトトラの所まで来ると前が吸着する・・・、という様にして前進していきます。

しかし実際このプログラムでは、吸着する必要のない電磁石が、永久磁石と反発するようになってしまっています。これは当初の案では電磁石を OFF にする予定だったのですが、実験をしてみたところ電磁石の軟鉄心が永久磁石の影響を受けて引き合ってしまったため、反発にすることにしました。

ここからは幽霊のM3竹下が執筆します

ここに記載されたことには誤りが多数ありますがなにとぞごかんべんのほどを

・コイルについて

「インダクタ」などとも呼ばれるコイルですが、一概に言っても見た目も用途もさまざまで表面実装型とかバネ状のものなどいろいろあって種類も限りなく多く大変扱いにくい代物です。もし使うなら事前にちゃんと調べることをお勧めします。

i 原理…導線に電流を流すと導線を中心とした同心円状に磁界ができます。図1

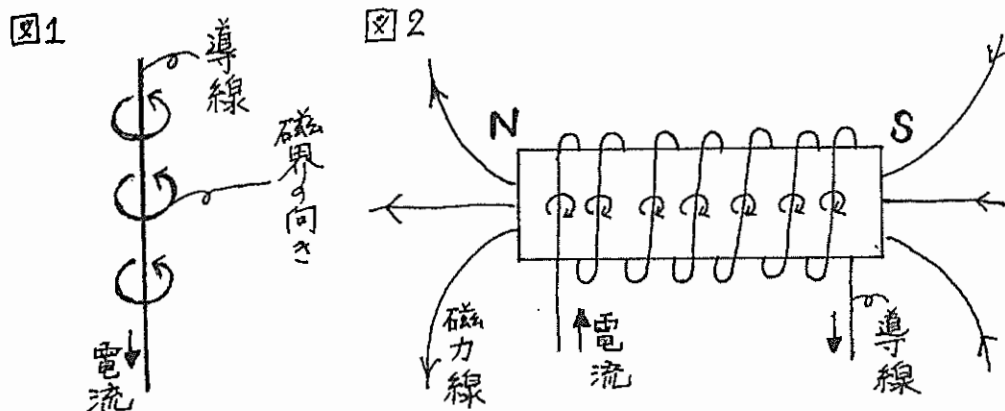
これをバネ状に成型すると内部で磁束が同一方向に強められます。

これによって擬似的な磁石のようになります。これが原理です。図2

また周囲の磁界が変化するとコイルに電流が流れます。(電磁誘導*1)

ii 電磁石…コイルに軟鉄芯*2などの強磁性体を通したものです

ON/OFF 切り替えのできる磁石みたいなものです。



はいここまではまじめにかいてきたわけですが

コイルを理解しようなんて僕のような低能では無理ってことらしくさっぱりわかりませんそれでもコイルをきれいに巻くだけならできたのだから問題無いと思えますが、さすがにそうはいかないので今後の優秀な後輩たちの努力に大いなる期待を寄せることにします

*1 電磁誘導

コイル内部の磁界が変化するとコイルに電流がながれる性質のことです

詳しくは…いつか習うと思います

*2 軟鉄芯

軟鉄はゆっくりと冷やされた鉄であり。

鋼鉄と違って一度磁化しても元の鉄に戻りやすい。

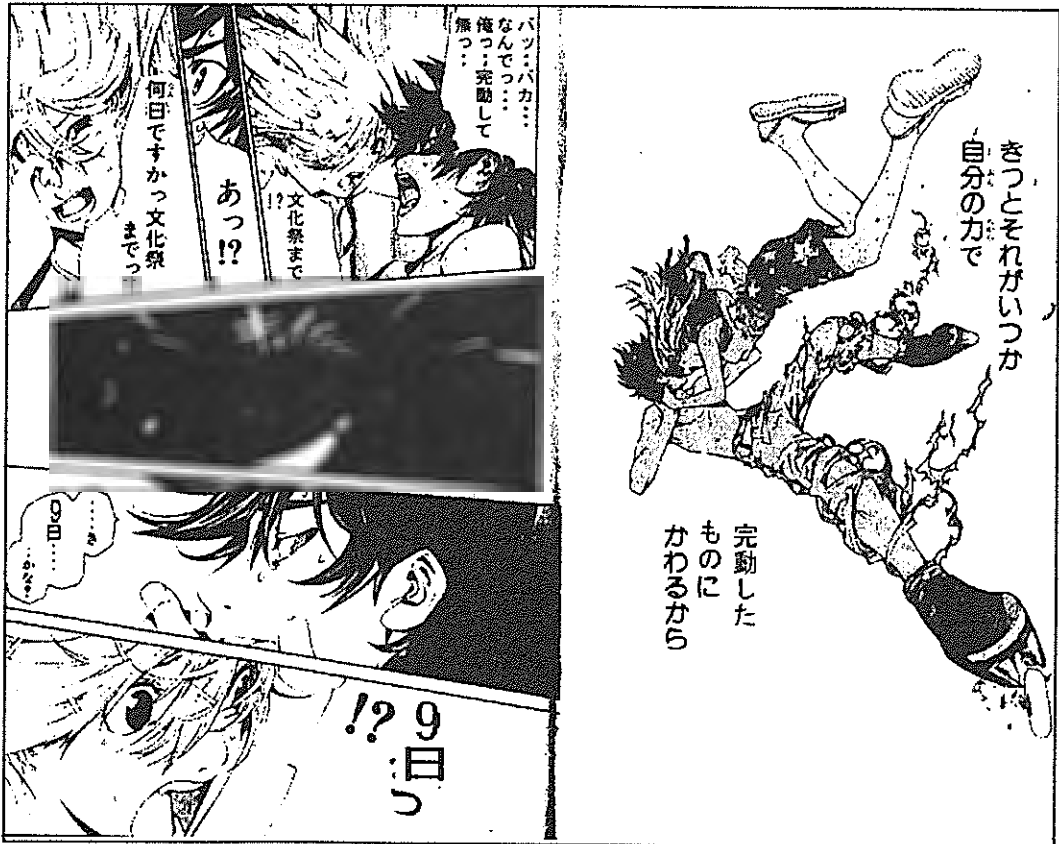
電磁石に鋼鉄を使うと ON/OFF できなくなります

はっきり言って才能で 95% 決まるコイル巻き講座

- 1 中に入れる軟鉄芯を作る。パワーがほしかったら太くしとくいいですよ。
- 2 芯を回転台に固定する。しっかりやらないとやりにくいです。
- 3 巻きはじめを瞬着でしっかり固定。ずれたら死にます
- 4 きれいに巻き巻き。ここは才能
- 5 折り返しを瞬着でしっかり固定。ずれたら死にます
- 6 4・5を繰り返す
- 7 巻き終わりを瞬着でしっかり固定。

つまりコイルとは…線が細くて太くて短くて巻き数が多けりゃ強いというおおよそ人間なら理解できるものです。ただし巻きすぎるとコイルの抵抗が大きくなりすぎ電流が流れにくくなって逆に弱くなります。またあんまり太すぎるコイルは固定しにくいのでご注意をあと巻くのは才能だからできないやつは……努力しろということで

注意、エナメル線がおれないように気をつけて下さい(折れると泣けます)



最後に駄文でサーセン 4月24日

7. 感想

H2 中嶋 剛大

『リニアモーターカー』にはロマンが詰まっています。カッコイイです。・・・ということで、作ることにしました。今更ですが、単純な理由ですね～(笑)。

まずは反省を・・・やはり、作り始める前に知識を集めておく事と計画性を立てる事は必要でした。もっと実験にたくさん時間を割くべきだったと思います。

共同製作者の西村と竹下へ。君たちのおかげで製作が進みました。最後に素晴らしい思い出をありがとう！初めて前に進んだときのテンションはやばかったです。

読んでくださった方へ。全体的に日本語が下手ですみませんでした。

でも、ここまで読んでくださって本当にありがとうございます。少しでも興味を持ってもらえたり、タメになったと思っていただけたら嬉しいです。

H1 西村 陽樹

僕がリニアモーターカーを作ると決まったとき、今まで何人もの人が挑戦しているのだから、車輪で走るだけならすごく簡単だろうと思っていた。しかし実際は、Hブリッジも、フォトトラもほとんど扱ったことも無かったし、それができても、コイルの抵抗値が小さすぎたせいでFETもコイルも熱くなって・・・という感じで、考えている分には単純に見えても、やるのが次から次へと出てきて、それにどんな対応をとるのが一番いいのかも見当がつかなかった。正直他の人の流れについていくのがやっと、という感じで、他の人たちが言っていることを、記憶をあちこちから引っ張りだしながら聞いている感じだった。

でも、その分学んだものはあったと思う。おとしは741Cを使っていたし、去年はステッピングモータを制御しているだけだったので、センサー系や、Hブリッジ、そして電磁石について学ぶことができて、有意義な時間をすごせたと思う。なんとか今年中にそれらをマスターして、来年は人に頼りすぎるのではなくできるだけその技術を使っていきたいものだ。

中嶋さん、竹下、いろいろとわからないことも多く迷惑をかけたと思いますが、一緒に製作できて本当によかったです。ありがとうございました。

M3 竹下 涼

コイル巻き機の製作に半端じゃない時間を掛けた割に結局動かずそして使わないという結果に陥り途中僕自身が部活に来ないという事態にもなってしまいなかなかいいものが出来上がったとは思えません。来年はもっと見通しを立ててきちっと製作します。僕は回路とかがいまひとつわかっていないので来年は回路についての理解を深め知識を蓄えて後輩に教えられる立場になりたいと思っています。

最後に物無の方には多大な迷惑をおかけしましたそれでも皆さんとは有意義に過ごせたと思いますのでこれからもよろしくお願いします。



Climber-3 The Final Edition

高2 竹島 啓純
協力者:OBの皆さん&
無線班のみんな

OUTSIDE



Width: 300mm
Span: 300mm
Height: 300mm
Weight: 500g

What's This?

このClimber-3は登り系ロボットの第三弾です。世の中の摂理で大体のシリーズものは3作で終わるようなのでこれでこのシリーズもも終わりでしょう。(3の次は3-Xという案もありますが・・・)

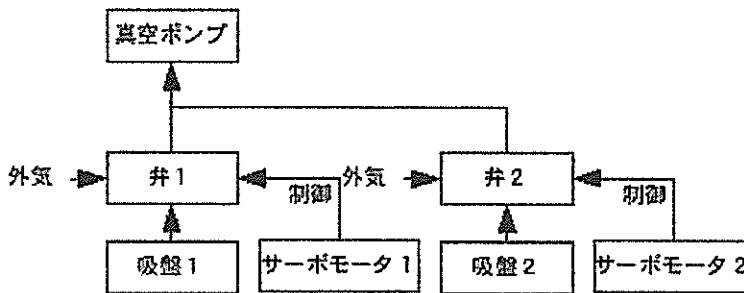
昨年の2に引き続き、また壁を登るわけですが、2で最大の問題であった本体の歪みを解消すべく本体を全て2mm厚のアルミ合金板で構成することにより2とは比べ物にならない強度を持たせたんですが、当然のように重量が2とは比べ物のならないものになってしまいました。また、左右のぐらつきを抑えるために吸盤を左右3つづつの計六個に増やしました。

そこで、メインモーターの駆動電圧を2.4倍の12Vにアップして巨体を支えるだけのパワーを持たせてます。

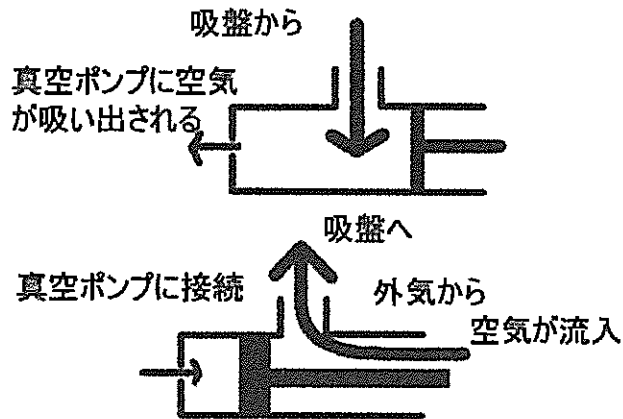
次のページから各部分の詳細を解説していきます。

さてClimber-3の空気系統は2を世襲しており、真空ポンプにより吸盤内の空気を吸い出し、吸盤の吸着、非吸着を切り替えるようになっている。下の図は機体の空気の流れ(サーボモータと弁の矢印を除く)を示す図です。

左右に一つづつついた弁で左右の3つづつの吸盤の内圧を制御します。吸盤と真空ポンプをつなげると吸盤内の圧力は減少して吸盤は壁に吸着し、外気とつなげると穴の開いた吸盤状態になり壁から吸盤は離れます。



弁は下のようになっておりシリンダーの片側が開放、もう片方が真空ポンプにつながっている。その中を動くシリンダーにより、吸盤とどちらかをつなぐのが選択される。シリンダーはサイズの関係上サーボモータを使用してしまった。



真空ポンプは、普通のピストンと金魚用の逆流防止弁を使用して製作しています。逆流防止弁があまり高圧用のものではないので、そこまでの引圧は作りないでしょうが、それでも十分な吸着力を発生させているのでOKということで。

ちなみに、この空気周りに限ってはいまだに未完成なので今後変わる可能性大ということで...

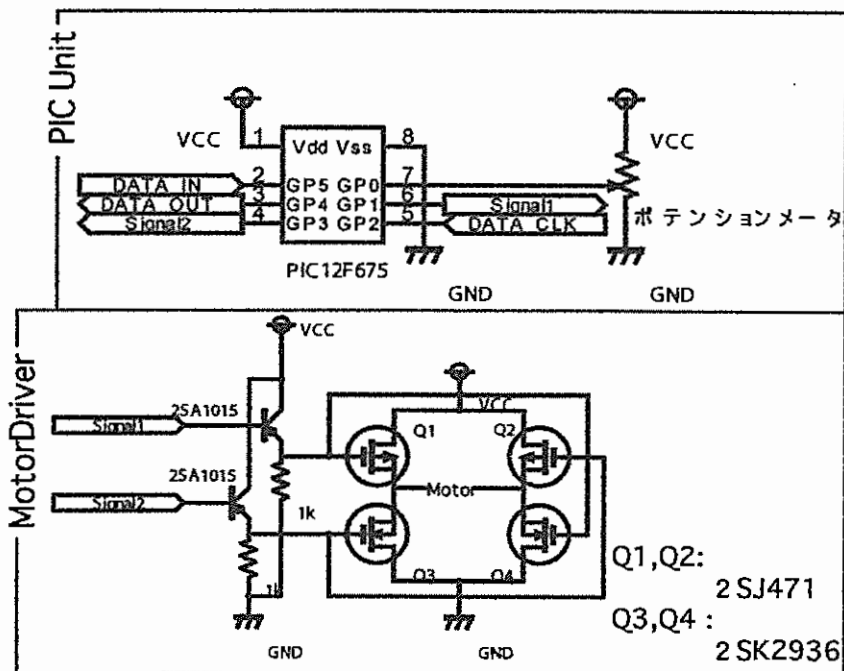
さて、回路です。ついに、3年に及ぶ長きにわたり開発を行っていたモーター制御が完成したのですよ。

まず、回路の全体像ですが、本体の各部各部（肩駆動系、弁駆動系、ポンプ駆動系etc...）を一つずつ独立させ、その各部をスレーブとした簡易SPI（スレーブセレクトとクロックを合わせた物）を使って接続。という形になっています。今のところメインマイコンにはPIC16F648を使用していますが、ここにもっと高性能なマイコンを使えばカメラや無線LANなどのオプションもつけることが可能な、発展性のある回路にしています。

と、ということで各部の回路の詳細な解説を・・・

まずはロボットの動力源である肩部分のモーター制御回路から。この部位はPIC12F675を中心とし、FETによるモータードライバと、可変抵抗によるポテンションメーターを接続しメインマイコンからのデータに従い、モーターをPWM制御してます。

回路図は下のです。モータードライバ前段の2sa1015はベース前に抵抗入れるかもしれません。2つのモーターを制御するので、下のは2つあります。



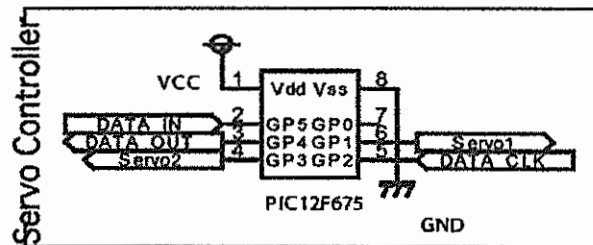
プログラムは極めて簡単です。通常時は

- 1 : GP0に接続されたポテンションメーター（以下PM）の値を読み込む。
- ↓
- 2 : 記録されている目標値との差を出す。
- ↓
- 3 : その差が32以上ならモーターは常時オン。それ以下ならその差×1msだけモーターオン。
- ↓
- 4 : 1に戻る。

を繰り返し、GP2に立下りが入るたびにINT立下り割り込みを利用して、マスター側からのシリアルデータをGP5から読み込むようにしています。

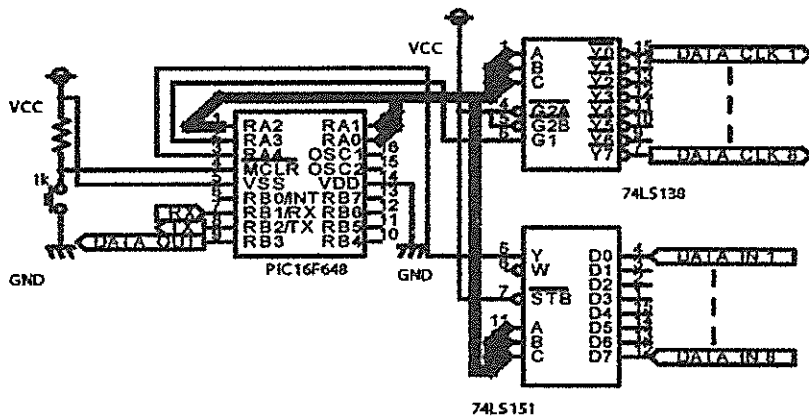
また、PMの値と、目標値の差が32以内の場合、GP4をHighにすることでマスター側に角度調整終了を伝えるようにしています。

弁開閉用アクチュエータはさらに簡単です。メインアクチュエータと同じように簡易SPIで受信したデータを弁制御用サーボにTimer0を使って送信するだけです。ここには特にすごいことをしていないので割愛します。回路図は下のものですが・・・何の面白味もありませんね



最後に、全てを統括するメインユニットと、簡易SPIの解説です。

SPI通信は米モトローラ社の提唱した3本の信号線(DataClock, DataOut, DataInput)を使用するシリアル通信形式です。通常は複数の素子を接続する場合、CS(ChipSelect)信号を使用して接続する素子を選択します。しかし、今回の用途では信号受信がわの使用ピン数を減らすため、CS信号とDC信号を統合し、単にクロックの有無によって素子の選択をすることにしました。回路図は以下のものです。



これが回路です。74LS138のイネーブル信号をいじることにより、クロックを生成。74LS151で入力を切り替えています。また、RS-232C接続のためのRxとTxを引き出しています。

ついに、最期の製作物が終わってしまいました。この3年間ひたすらモーター制御について研究してきましたが、CPLDでも持ち出さない限り、この12F675+FETでの制御が一番いいようです。

これを参考にしてロボットを作る人たちに一つ。ロボットの各部ごとに制御は分けてそれをバスでつなぐようにしたほうが、デバッグや拡張性において有利になると思うので今回の簡易SPIとかI2Cバスとかを積極的につけてください。

最後に、外部の方で買って下さりしかもこの文章を最期まで読んでくださった方、本当にありがとうございました。

追記: PIC12F675のGP3は入力専用だったのでGP5と入れ替えて使用しています。

Lorelai2008

製作者 H1 菅原 寛生 M3 中岡 勇太

前半部分は M3 菅原がお送りします

1 概要

—潜水艦—

この乗り物について皆さんはどのような考えを抱かれていますでしょうか？もっとも、もともとが隠密に敵を攻撃することを目的に作られ、目立つこともなくあまり知られるところが少ないと思います。

しかし、今では深海調査艇などが天然ガスなどの油田を見つけたり、科学的な調査などに役立ったりして新聞などにその成果が載せられることも珍しくありません。

この製作物はそんな潜水艦の仕組みをリアルに再現しようとしたものです。

2 本体概要

ここでは本体の構造などについて述べていきます。

この製作物は次の 3 つの部分で本体が構成されています。

・外殻 ……外側の構造体。内殻との間に錘や浮力材を入れてバランス調整をする。実はなくてもあっても変わらないのでまだ製作されていません。

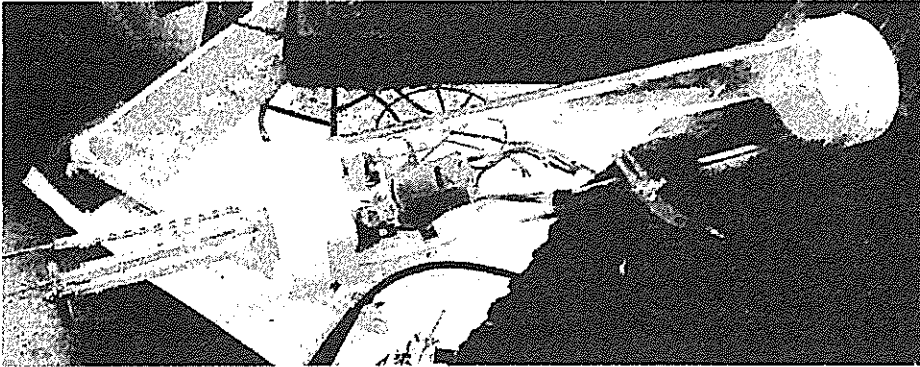
・内殻 ……本体において最も重要な部分。水圧に耐えるため頑丈な構造をしている。中に回路、モーターなどがある。

・バラスタंक……魚の浮き袋に相当し、中に水を入れたり出したりし浮上したり、沈んだりする。

この中でメインとなる内殻とバラスタंकの制作方法について説明します。

3 内殻

とりあえずどんな感じのものかを写真で乗せます。



先ほども書いたとおり今回の内殻の構造はWTB(ウォータータイトボトル、水密ボトル)という水圧に耐える頑丈な構造になっています。

基本的な構造材として使われているのは直径 6cm のアクリルパイプです。

そのアクリルパイプの両端に軟膏などを入れるようなポリプロピレン製の広口ピ

ンの蓋の部分をはめ込み、防水パテで固定兼防水をしています。

また、軸部の防水は左の図のよう

に 1ml のいわゆる予防接種用の

注射器の内部にシリコングリス

という防水用のグリスを満ち、そ

の中にスクリューの軸をいれ防水

する、という方式をとりました。サ

ーボモーターのリンゲージ(舵を動

かすための棒)も同じように防水しています。

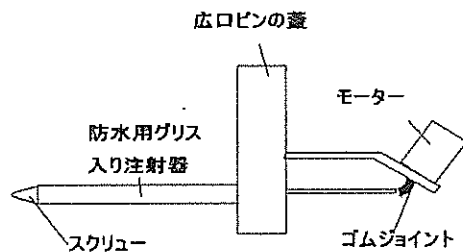
4 バラストタンク

ラジコン潜水艦におけるバラストタンクは代表的なものが 3 種類あります。

I; ガスボンベタイプ

実際の潜水艦と同じように船体内にガスボンベを入れ、ガスをタンク内に送り込んで浮上する

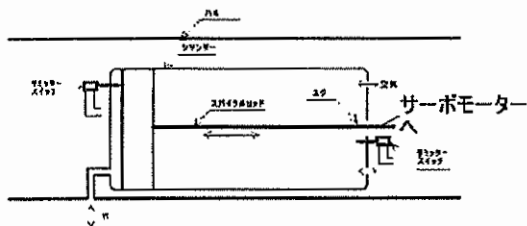
II; ポンプタイプ



タンク内にポンプを使って水を注廃水するタイプ。

III:シリンダータイプ

注射器のようなタンクでシリンダーを押したり引いたりすることで注排水するタイプ。



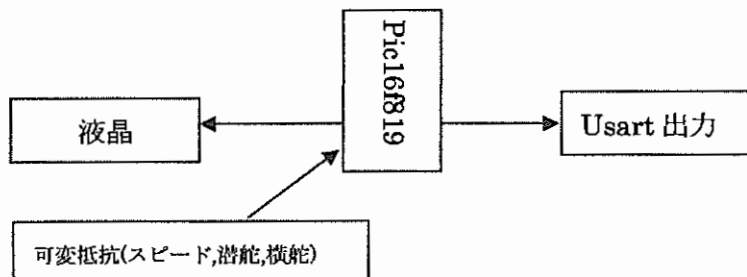
今回はいづらか構造が複雑になっても浮上、潜水が確実なシリンダータイプを選びました。右の図のように棒がシリンダにつながっており、それをサーボモーターで前後させて注排水するわけです。

5回路及びプログラム～コントローラー部分～

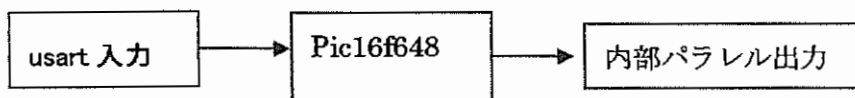
このコントローラー部分では可変抵抗からの電圧データを AD コンバーターで読み込み、そのデータを液晶に表示し USART 通信で送信するという形式をとりました。USART とは非同期式のシリアル通信で、送信線と受信線のみで通信ができます。そのため無線通信などの単線の通信回路をつけるときに簡単に組み込むことができます。以下にブロック図とプログラムを載せます。

【ブロック図

<送信側>



<受信側>



II プログラム

今回は C 言語を使用しました。因みにコンパイラーは CCS 社のものです。
プログラムと入っても usart の受信割り込みが入らないため実際はまだ動いて
いません。そのため今現在動作している送信側のもプログラムを載せます。

```
#include <16f819h>
//定数/変数指定
long speed;
int serialout=0;
//tmr1 割り込み関数
#define timer1
void isr1()
{
    //定時シリアル出力
    putc(serialout);
}
//メイン関数
void main()
{
    //ADコンバーター設定
    setup_adc_ports(RA0_RA1_RA3_A
    NALOG);
    setup_adc(ADC_CLOCK_DIV_32);
    //MAX
    //lcd設定
    set_tris_b(0);
    lcd_init();
    //LCD初期化
    lcd_clear();
}
//クロック周波数指
#define
//液晶表示ライブラリ設定
#define mode 0
#define input_x input_B
//ポート
B使用
#define output_x output_B
#define set_tris_x set_tris_B
#define stb PIN_B3
#define rs PIN_B2
#include "stdlib.h"
#include "lcd_lib.h"
#define
rs232(BAUD=2000)XMIT=pin_B0,RCV=pin_B1,
BRGH10K(INVERT)//出力B0、入力B1
```

```

//LCD全消去
printf(lcd_data,"MAIDAINABUTU
MU"); //初期メッセージ
//tmr1 設定
setup_timer_1(T1_INTERNAL
T1_DIV_BY_8);
set_timer1(0x00);
// 割り込み
enable_interrupts(INT_TIMER1);
enable_interrupts(GLOBAL);
while(1)
//永久ループ
{
speed=scaleout;
speed=0;
set_adc_channel(0);
delay_us(50);
speed = read_adc();
lcd_cmd(0xC0);

printf(lcd_data,"speed=%lu",speed);
}

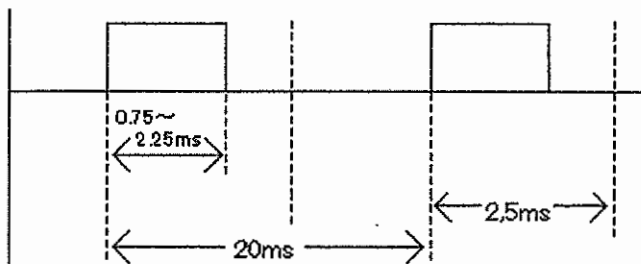
```

後半は M3 の中岡です。

先ほど説明していない内殻に入る部分、サーボモーター、モーター制御の Hブリッジ、PWM とバラスタタンクに使用する真空ポンプについての説明をします。

6. サーボモーター

サーボモーター(後サーボと表記)とは、外部からの入力で、自由な角度を指定できるモーターのこと。大きな力の要る作業は不可能だが、制御が簡単なので使用している。今回はこれを使い、舵、バラスタタンクの制御を行う。



左の図のように 0.75ms ~ 2.25ms の信号により、角度を指定する。基本は 1.50ms で 0°。角度の変化は 1.50ms から

の差に比例し、0.75ms で -90°、2.25ms で +90°、と 180° の間を自由に設定できる。サーボモーターは 20ms 毎に信号を送った時に一番安定して動作するため、信号は 20ms 毎に出す。又、そうする事で最大 8 つを平行して制御するこ

とが出来る。角度の指定の方法は、菅原さんの製作のコントローラーを使い、コントローラー部の pic16f648 とパラレル通信を行うことで指定する。

今回は TMR1 を 0,25ms に設定し、舵は -90° ~ $+90^{\circ}$ まで 30° 刻みに計 7 段階の角度設定を行う。バラスタックは開閉のみの操作のため、 -90° と $+90^{\circ}$ の 2 段階のみ使用する。又、各箇所にも 2 個ずつ使うため、計 4 つのサーボを平行して制御する。1 サイクルが、割り込み 80 回で構成されているので、はじめの 3~9 回の割り込みの間だけ H の信号を出し、H にしていた時間を 80 から引く。その後引いた差の時間だけ L を出力する。L を出している間に別のサーボの制御を行う。このように 4 つのサーボを制御する。

サーボのプログラムの流れを表にして説明する。

- ・TMR1 を 0,25ms 毎に設定する。
- ・バラスタックはどちらも割り込み 3 回、舵はどちらも 6 回に指定し H 信号を維持。そして、指定した回数を 80 から減算する。この回数は、別途で汎用レジスタに保存。
- ・指定した回数が終わったら、汎用レジスタに保存した回数だけ L 信号を維持。
- ・4 つ全てのサーボの指定が終わったらコントローラーから、角度の指定などのデータを受信する。これも汎用レジスタに保存する。
- ・合計で 80 回割り込みが行われたら、3 で受信したデータ通りにサーボの H の割り込み回数を指定。2 と同じように、H の回数を 80 から減算し、数値を保存する。
- ・2 に戻り、この動作を繰り返す。

7,Hブリッジ

Hブリッジとは、トランジスタ又は FET を用いてモーターを制御する回路の事。普通、モーターにスイッチを付けたのみだと、正転と停止だが、Hブリッジを使えば、正転、逆転、停止、そして、ブレーキも出来る。今回は 4 つの動作の内、正転、逆転、停止を使い前進後退用のモーターを動作させる。

Hブリッジの使用効率を上げるため今回は、FET の 2SJ377 と 2SK2231 という型番の理由は特性の真逆な、FET を使ってる。

Hブリッジの詳しい説明は、リニアモーターカーの西村さんの所に記載してあ

る。

8,PWM

PWMとは、モーターに流れる電流を細かくH、Lさせ、HとLの時間を変える事で動いたり止まったりを繰り返す時間を変え、擬似的にモーターの速度を変えするという技法の事。このPWMを使い最低速度～最高速度を16段階に分け、潜水艦の移動速度を決める。

HブリッジとPWMにより、前進、後退とその速度そして停止をする訳である。

今回はPWMの間隔を決めるのに、サーボモーターに使ったTMR1を流用して制御を行う。サーボモーターの関係上、プログラムのサイクルは、20ms 毎になる。0,25ms 毎に割り込みが入り、1 サイクルは割り込み 80 回で構成されている事となる。速度は 16 段階なので 1 サイクルを 5 つに分け、割り込み 16 回の小サイクルにわけると。この小サイク中何回 H 信号を出すかで速度が決まる。この処理は、サーボの制御と平行して行われる。

9,真空ポンプ

真空ポンプとは、モーターと弁を使い、ある密閉された空間から空気を抜く装置の事。このポンプを使い、バラストタンク内の圧力を変え、水の出し入れを行います。

一見制御が難しそうに見えるが、動作するのはモーターでありさらに一方向のみの動作で良い為、片側のみの H ブリッジのみで良い訳である。速度も変える必要も無い為、ただ、動くか止まるかを指定すればよい。

10プログラム

```
LIST    P=PIC16F648A;LIST  宣言で使用する          ORG    0
PICを16F648Aと定義する。          GOTO   SETUP
                                     INCLUDE P16F648A.INC;設定ファイル
                                     p16f648a.incを読み込む。          ORG    4
                                     _CONFIG _INTOSC_OSC_CLKOUT          BCF    INTCON,GIE
                                     & _BODEN_OFF & _BOREN_OFF & _CP_OFF &          GOTO   WARIKOMI
                                     _DATA_CP_OFF & _PWRTE_ON & _WDT_OFF &
                                     _LVP_OFF & _MCLR_OFF          SETUP
```

;基本的なセットアップでござんす。

```

CLRF    PORTA
;ポート A の出力を L にする。

MOVLW  B'10000000'      MAIN
MOVWF  INTCON           ;メインのプログラム
;イントコンの設定(割り込み許可)。

BSF    STATUS,RP0      BCF    PORTB,1
;ページを 1 にする。   BTFSS  PORTB,2

CLRF   TRISA           GOTO   A1
;ポート A を出力にする。
BSF    PORTB,1

MOVLW  B'11111101'    GOTO   A7
MOVWF  TRISB
;ポート B の入出力設定。

MOVLW  B'00000001'
MOVWF  T1CON          ;TIMER1 設  KS
定(プリスケラー1 倍)           ;角度指定のアレ

BCF    STATUS,RP0
;ページを 0 にする      A1

CLRWF  ;w レジスタをクリアする。
BSF    PORTA,2
MOVLW  D'3'           MOVWF  D'3'
;待ち時間
BCF    PIR1,TMR1IF   MOVWF  50H
MOVLW  B'11111111'   CALL   TM
MOVWF  TMAR1H        BCF    PORTA,2
;
MOVWF  TMAR1H        MOVWF  D'7'
;動く時間
MOVLW  B'00001001'   MOVWF  50H
;タイマ 1 を 250 に設定(250 カウン
MOVWF  TMAR1H        CALL   TM
;ト)0.25ms
MOVLW  D'2'           CALL   AATM
MOVLW  D'30'          GOTO   MAIN
MOVWF  50H

```

```

                                RETURN
A7                                WARIKOMI
                                ;割り込み
BSF    PORTA,2
MOVLW  D'9'                       BTFSS   PIE,TMR1IF
;待ち時間                         GOTO   WARIKOMI1
MOVWF  50H
CALL   TM
MOVLW  D'1'                       WARIKOMI1
;動く時間                         ;タイマ 1 割り込み
MOVWF  50H                       MOVLW  B'11111111'
BCF    PORTA,2                   MOVWF  TMAR1H
CALL   TM                         ;
CALL   AATM                      MOVLW  B'00001001'
GOTO   MAIN                      MOVWF  TMAR1H
                                ;タイマ 1 を 250 に設定(250 カウン
                                ト)0.25ms
TM                                OECFSZ  50H
NOP                                GOTO   WARIKOMI00
BTFSS  INTCON,T0IF              MOVLW  B'10100000'
GOTO   TM                       MOVWF  INTCON
BCF    INTCON,T0IF              ;イントコンの設定(割り込み許可)。
RETURN
                                RETFIE
AATM
MOVLW  D'70'
MOVWF  50H                       WARIKOMI00
NOP                                ;タイマ 0 割り込み その2
BTFSS  PIR1,TMR1IF             BCF    PIE,TMR1IF
GOTO   $-2                      MOVLW  B'10100000'
BCF
                                MOVWF  INTCON

```

11感想

H1 菅原

今回は二度目の共同制作だったわけですが、省みるとなかなか反省すべき点が多かったように感じます。1つ目として共同制作にもかかわらず中岡との製作における係わり合いなどをほとんど持たず、進度がまちまちになってしまったという点。2つ目は予定や分担の割合が甘く機能を大幅に削減せねばならなくなったということ。来年はこれらの教訓を生かしてより納得のいく製作物を作っていきたいと感じた。

M3 中岡

今回の制作では、反省する点が、いくつもある。まず、共同制作であるはずなのに菅原さんと殆どコンタクトを取っていなかったということ。次に、Hブリッジを作るのにかなりの時間をかけてしまった事。またそのために、プログラムを書き始めるのが、遅くなってしまったことなどがあげられる。来年は、今年の反省を生かして制作をして行きたい。

製作物名「Teacher4 -先生-」 本体—物理部無線班		
破壊力… D	スピード… A	射程距離… B
持続力… A	精密動作性… D	成長性… C
Teacher1, 2の“銃”という概念と、 Teacher3の“ロボットが的”という概念を引き継ぎ、 人間のように、自ら学習し、成長していくAIを持つ。 Teacherシリーズ、ここに完結！		
Teacher4 -先生- 誕生		製作者 H2 中嶋 剛大 H2 竹島 啓純 協力者 物無のみんな

A—超スゴイ B—スゴイ C—人間並 D—ニガテ E—超ニガテ

What is "Teacher4"?

早速、製作物の概要を説明したいと思います。

下の四角の中は製作物と直接関係は無いので読み飛ばして頂いても結構です。

Teacher シリーズとは、中嶋が中一から毎年作り続けている『射的ゲーム』のことで、名前の由来は、作業する際に机などを傷つけないために雑誌を用いるのですが、その雑誌の漫画の1コマ、「先生！嘘だよな？先生！」という台詞によるものです。ちなみに、未だにその元ネタとなった漫画は読んでいません。

Teacher4 は、自動で動く的をプレイヤーが銃で撃つというシンプルなルールの射的ゲームです。

～銃の仕様について～

エアガンの中身をくりぬき、回路を入れたもの。また、ショットガンを使っているため、1発撃つごとにリロードする必要がある。(これは竹島の趣味)

銃口には赤外線LEDが埋め込まれていて、常に信号が発射されています。赤外線による信号パターンは「通常時」と「時弾発射」の2種類あり、的側のほうで“通常”、“銃口が向いた”、“弾が当たった”の3種類の状態を識別できるようにしています。

～的の仕様について～

的は、前輪駆動で後輪がボールキャスターの三輪車となっています。

赤外線受光モジュールを前後左右に一つずつ設置し、どの方向からでも信号を受信できるようにしています。

また、AIによる自動操縦で動きます。(データの記録にはEEPROMを、演算にはH8を、モーター駆動や赤外線受信にはPICを使っています)

～AIのアルゴリズムについて～

このAIのアルゴリズムこそが、今回の製作物の要となっています。

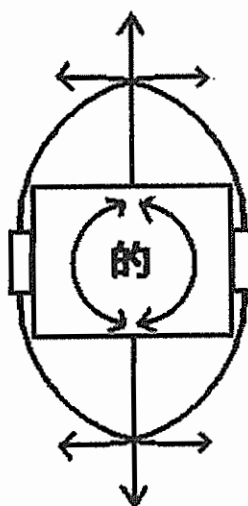
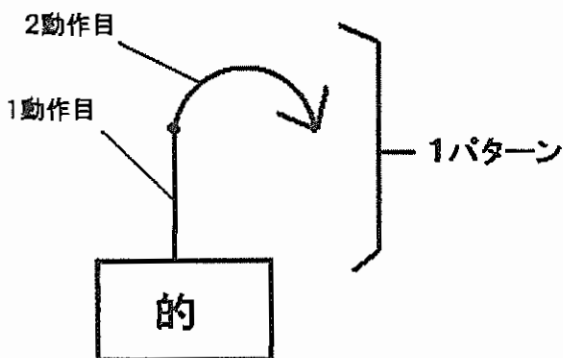
『モーターの駆動をパターン化して整理し、一番安全度の高いパターンを実行する』
と言う、動作をさせています。

①モーターの駆動パターン

モーターが回ってから、止まるまでを“1動作”とします。その2動作分を合わせて、1パターンとします。

つまり、AIは『次の2動作の組み合わせの中で、もっとも安全なものを探し出す』という働きをする事になります。

また、モーターの駆動パターン数は、全部で70通り(10通り×7通り)となっています。



静止
前進
後退
旋回
その場×2
右を軸×2
左を軸×2
前後に小刻みに動く

※注意※
旋回は1つにつき、時計回りと
反時計回りの2種類がある

1動作目は左図のように、
10通りありますが、

2動作目は、左図の「その
場旋回×2」と「前後に小刻み
に動く」を除く7通りとなりま
す。

便宜上、各駆動パターンに
は番号がついており、駆動パ
ターン[0]、駆動パターン[35]な
どと区別しています。(番号は
[0]～[69]まで)

②駆動パターンを選出する基準

Teacher4のAIは、パターン毎に1バイトの安全度データを持ちます。(全部で70バイト)

安全度は(選択回数÷パターン実行時のHIT回数)で算出しています。

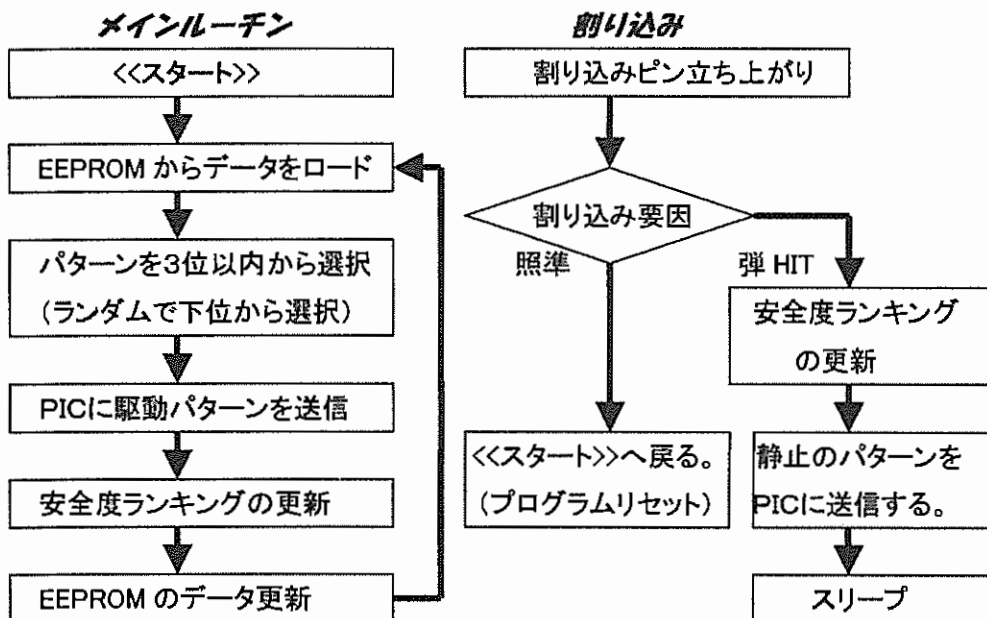
また、安全度の高い順にランキングを作成し、基準として保存しています。(全部で70バイト)

それぞれ、70バイトずつ使用し、合計140バイトのデータを用いて、駆動パターンを選出させています。

各駆動パターンの安全度		安全度ランキング	
[0]	3	1位	[2]
[1]	19	2位	[67]
[2]	59	3位	[33]
⋮		⋮	
[67]	46	67位	[0]
[68]	26	68位	[59]
[69]	36	69位	[28]

③AIのプログラム(H8を使用)

割り込みの要因は「銃の照準(銃口)が向いた」と「弾がHITした」の2通り。

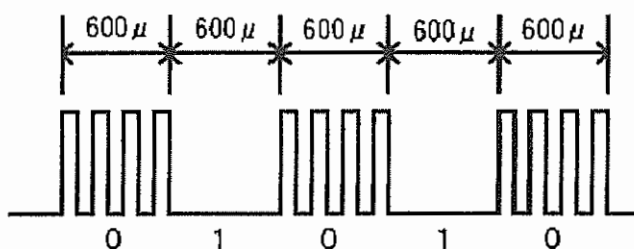


以上のように、シンプルな内容となっております。

パターンを完了した時と、照準を合わせられたときのみ、パターンを選出します。

～赤外線について～

38kHzの周波数で下図のように赤外線を送信します。

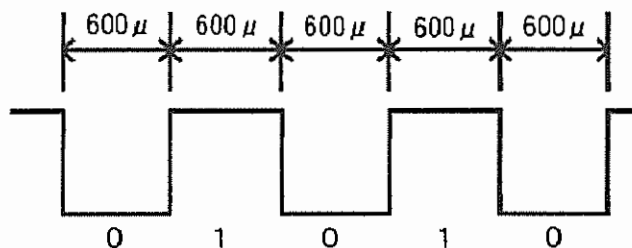


“0”送信のプログラム

```
int i=23;
while(i>0){
    output_high(PIN_B1);
    delay_us(10);
    output_low(PIN_B1);
    delay_us(10);
    i--;}

```

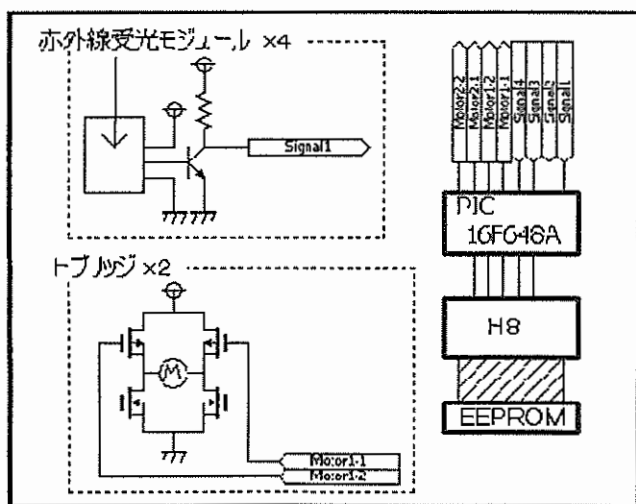
これを、受光モジュールで受信すると、下図のようになります。



今回の赤外線の信号は、通常時(トリガーを引いていないとき)は“1110100”、トリガーを引いているとき(発射時)は“1110110”となっています。

赤外線は簡単な上に遠くまで届くので、データ送信(シリアル通信)に便利です。

～回路図～

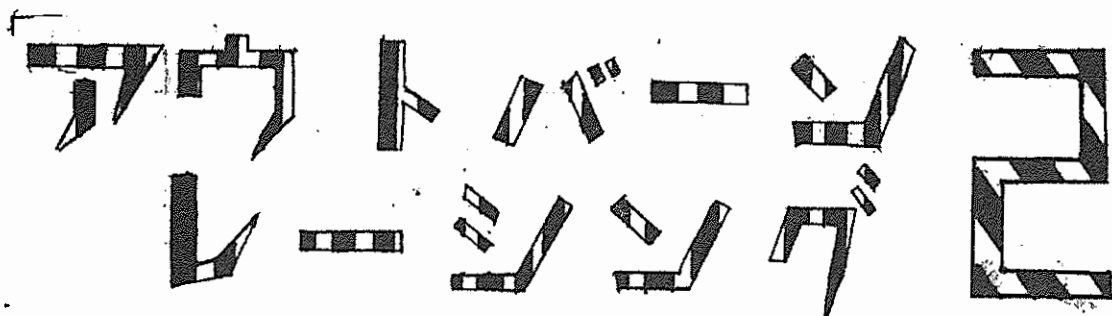


特に、複雑なところは無いはずです。

PICは、赤外線の受信(H8へ、割り込み信号送信)とH8からのデータ受信によるモーター駆動を行っています。

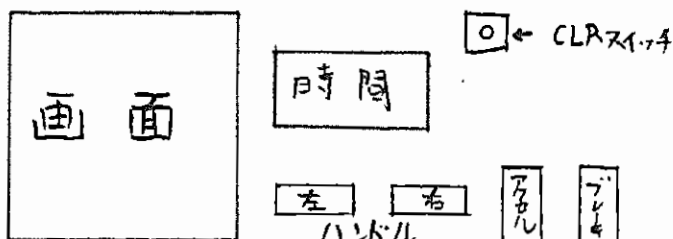
PICとH8の間のデータ通信は、基本的にパラレル通信で行っています。(シリアルよりも楽ですしね～。)





制作者：飯田 亮
 協力者：物原の比呂様

1. 外観



2. ストーリー

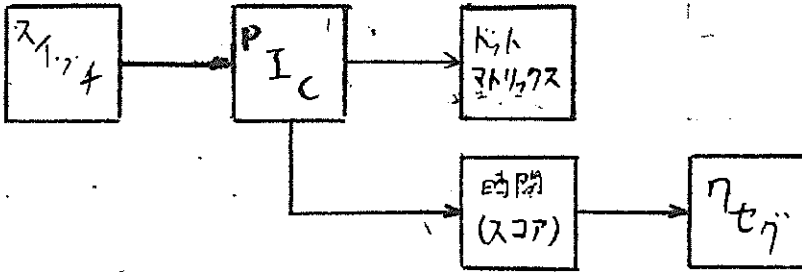
彼はアウトバーンで行われたレースで優勝した。しかし、ゴールで待っていたのはドイツ警察だ。警察はここで行われた違法レースのことを知って来ていたのだ。彼は逃げた。彼は警察の視界から消え去ることに成功したが、いつ何時やらが追って来ると分らない。車もカーチェイスでボロボロだ。彼は、ただひたすら走り続けた。

3. ルール + 説明

ゲームは上から見た視点で行われます。下の方に緑色の自機がいます。要は壁にぶつかると走ってネという事です。時間は一定です。アウトオブレンジ制で、それがそのままスコアとなります。一定以上のスコア以上になるとストップウォッチは動きません。車はダメージを受けているため、一回ぶつかっただけで壊れます。外観についての説明ですが、画面はリボX/6の2色のドットマトリックスを置き、時間は3ケタ表示、その上CLRスイッチはゲームのリセットボタンです。

4. ブロック図

このゲームはほとんどPICで作業を行っています。従ってPICメインになります。



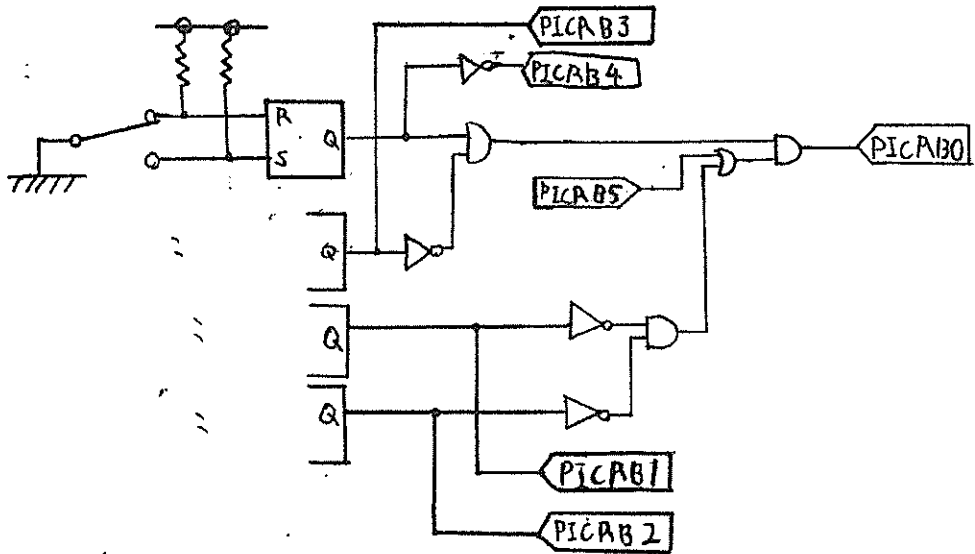
5. 回路図

スイッチ

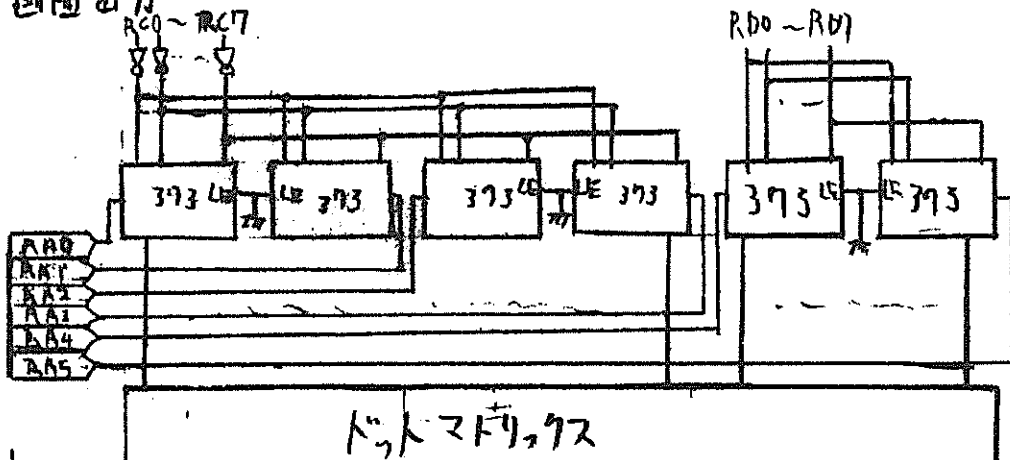
上セル

下セル

左



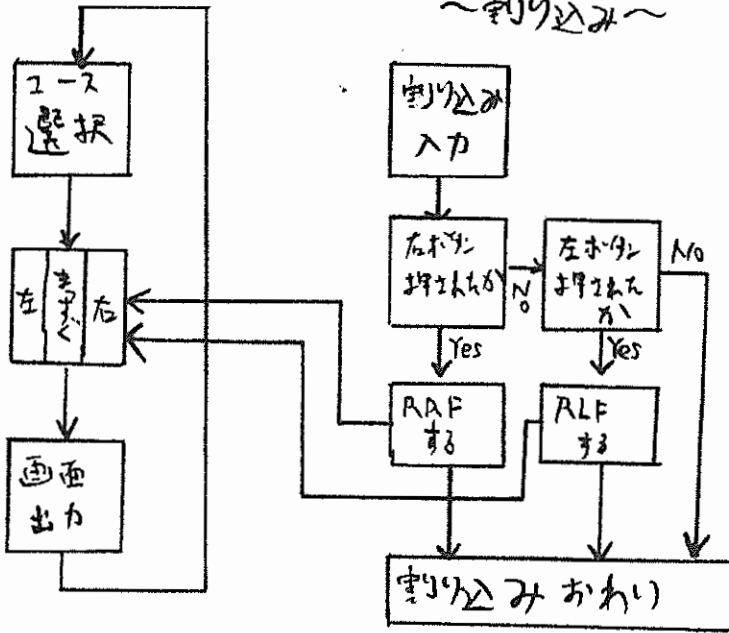
画面出力



6. チャート

PIC内部の命令の順番です。

～切り込み～



7. プログラム

《説定は省略》

```

MOVLW B'00001111'
MOVWF 20H
MOVWF 21H
MOVWF 22H
MOVWF 23H
MOVWF 24H
MOVWF 25H
MOVWF 26H
MOVWF 27H
MOVWF 28H
MOVWF 29H
MOVWF 2AH
MOVWF 2BH
MOVWF 2CH
MOVWF 2DH
MOVWF 2EH
MOVWF 2FH
MOVWF 30H
CLRF PORTC
CLRF PORTD
MOVLW B'00111111'
MOVWF PORTA
NOP
CLRF PORTA
MOVLW 20H
MOVWF FSR
BCF PORTB, 5
GOTO GAME
    
```

GAME

```

CALL HIDARIHAJIMARI
CALL HIDARITUKAN
    
```

このプログラムは、PIC 16F877Aの汎用レジスタの20Hから2FHまでデータを代入し、それをボタンマトリックスの1列目、2列目...とあてていき、そのレジスタをRLF、RRFすればそれぞれの列も丸LF、RRFされ、コースが変化するという仕組みです。(コースの左側は、右側のデータをCOMFして表示するため、コースのははははコマいつでも通じます。)

具体的な説明としては、GAMEの所でコースの表示する順を決め、HIDARIHAJIMARIやHIDARITUKANで20H～2FHのデータのRLF、RRFを行い、UGOKUでその20H～2FHのデータをそれぞれの列の所でそれぞれのデータをロード、出力します。

F < 1/2/82 >

*****表示*****

UGOKU

```

MOVLW B'00111111'
MOVWF PORTA
CLRF PORTD
CLRF PORTC
NOP
CLRF PORTA
MOVLW B'00000111'
MOVWF 41H
BCF STATUS, IRP

```

```

MOVLW 20H
MOVWF FSR
MOVLW B'00000001'
MOVWF PORTD
BCF STATUS, C
BSF PORTA, 4
CALL UGOKUX
BCF PORTA, 4
NOP
MOVLW B'00111111'
MOVWF PORTA
CLRF PORTC
CLRF PORTD
NOP
CLRF PORTA
MOVLW B'00000111'
MOVWF 41H
MOVLW B'00000001'
MOVWF PORTD
BCF STATUS, C
BSF PORTA, 5
CALL UGOKUX
BCF PORTA, 5
NOP
RETURN

```

UGOKUX

```

MOVF INDF, W
MOVWF PORTC
BSF PORTA, 2
NOP
NOP
BCF PORTA, 2
MOVF INDF
MOVWF 40H
COMF 40H, W
MOVF 40H, W
MOVWF PORTC
BSF PORTA, 3
NOP
NOP
BCF PORTA, 3
BCF STATUS, IRP
INCF FSR
RLF PORTD
DECFSZ 41H
GOTO UGOKUX
RETURN :共通

```

< 1/2/82 >

END

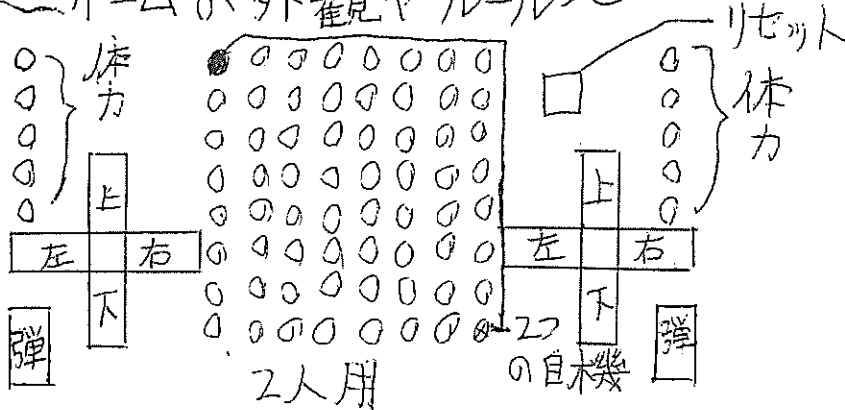
將軍をたおせ!

制作者 M2 伊藤 卓
 回遊図設計者 川原 さん
 協力者 物無の皆様

〜ストーリー〜

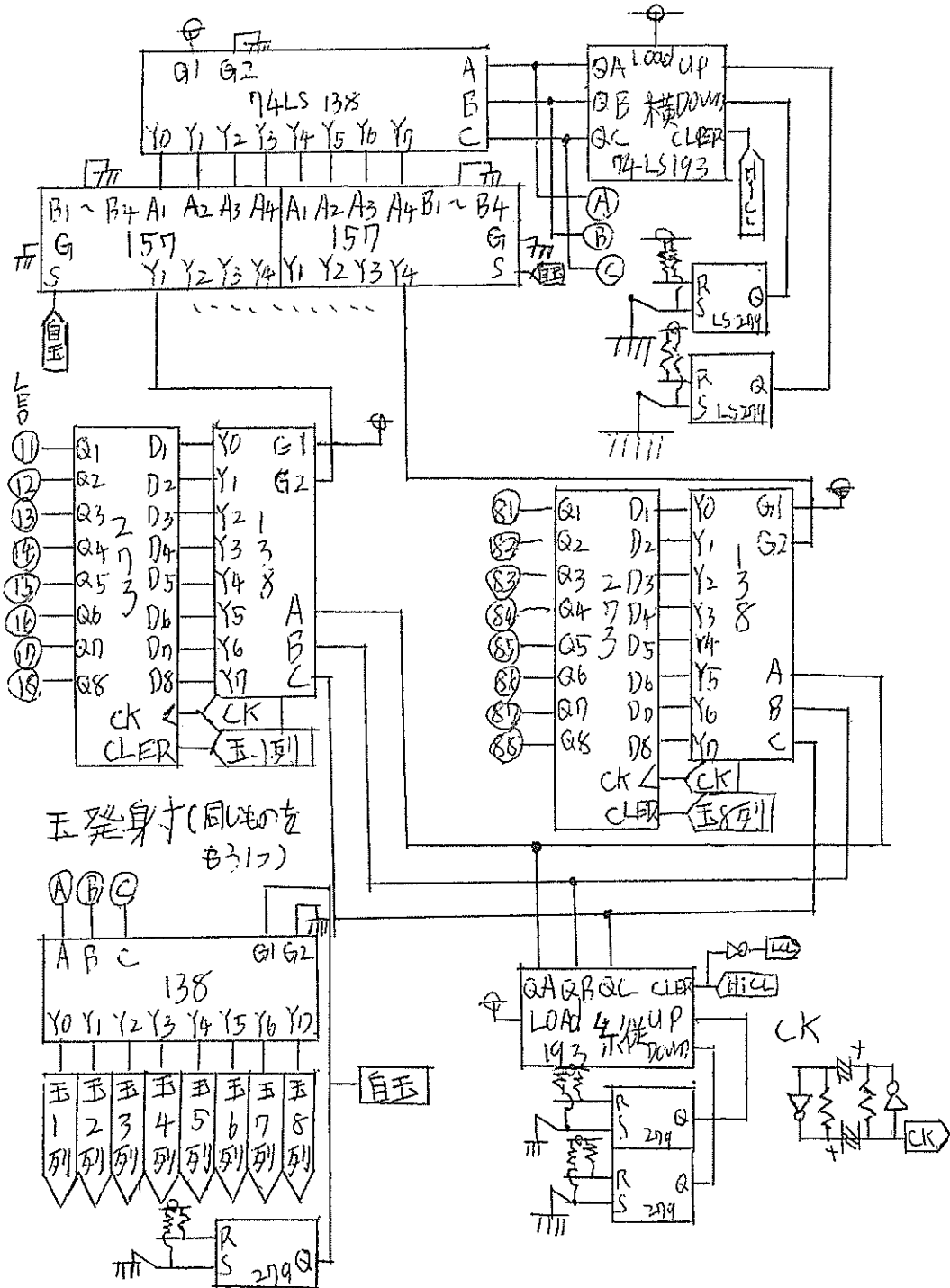
1622年 ある2つの軍が闘っていた。その2つの軍の強さは五角、露いさ長く続いた。そんな時片方の軍の將軍がもう片方の軍の將軍に話を打ちかける。それは將軍同士が闘い、勝ったほうの將軍の軍が勝利とするものだった。意見が合ひそれをやることになった。そして2つの軍の將軍の闘いが始まった。

〜ゲームの外観やルール〜



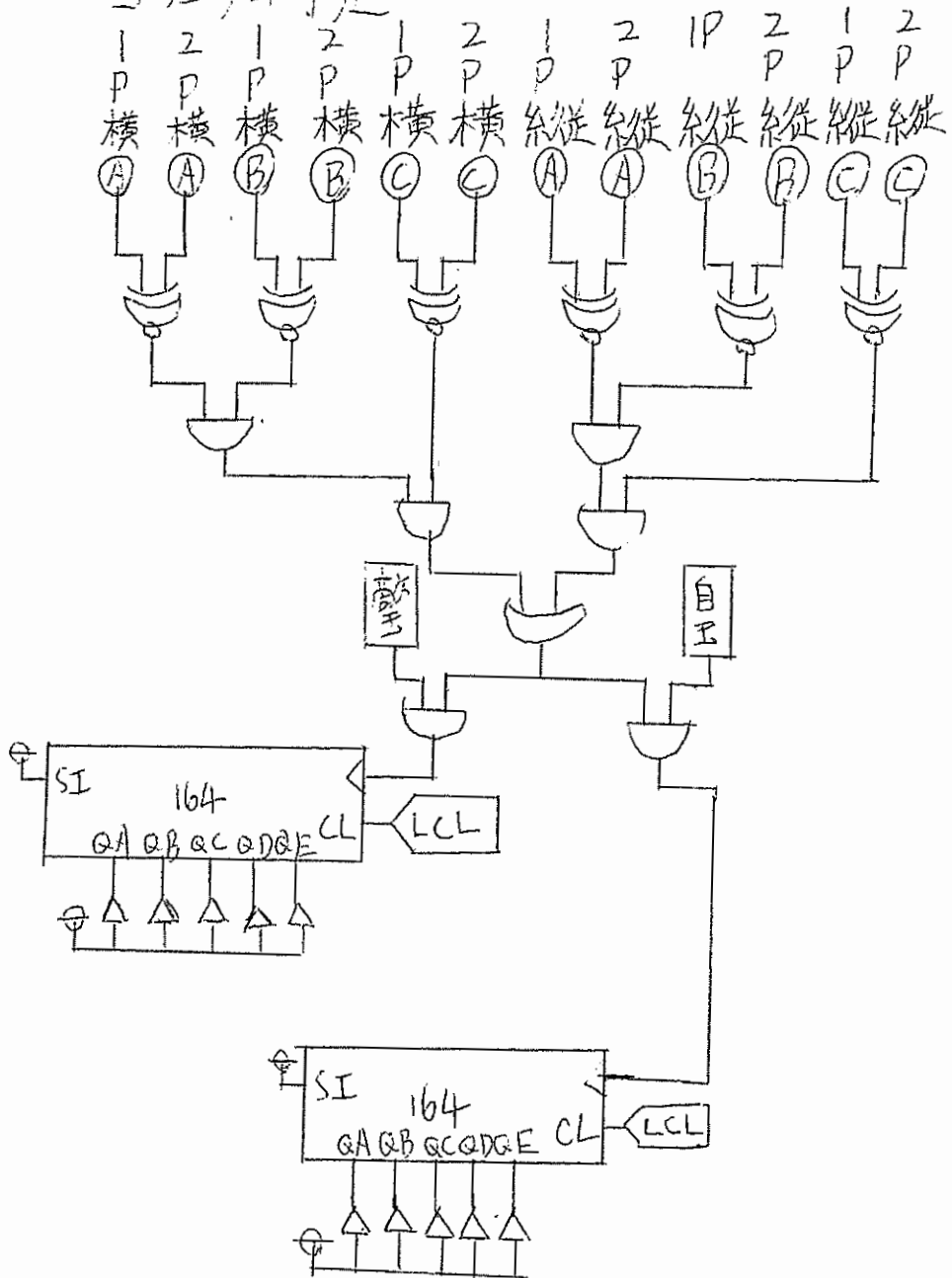
自機は上下左右にスズク動くことができ、弾は弾発射のボタンを4回押し4方向に弾を発射し、発射した後はまたボタンを4回押しないと弾が出ばなしで動くことができず、体力がらつて弾を5回敵に当てると敵が消えて勝ちとなるが逆に5回弾を当てられると負けとなる。

～ 回路図 ～ ICは全て74LSである
 自機制御力(同じものをもう1つ)



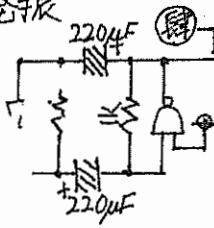
回路図 続き

当たり判定

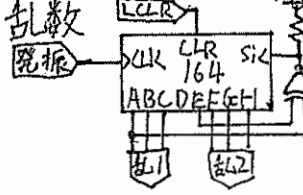


回路図: ICは全て74LS系です。
Vcc, Gndは省略してあります

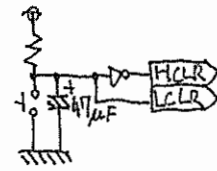
発振



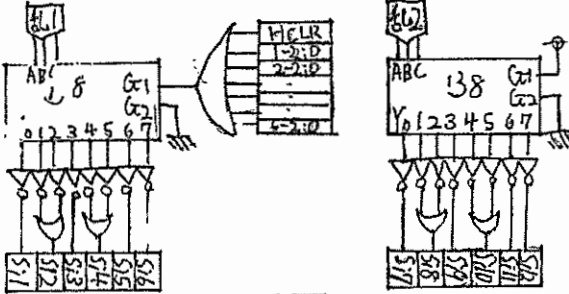
乱数



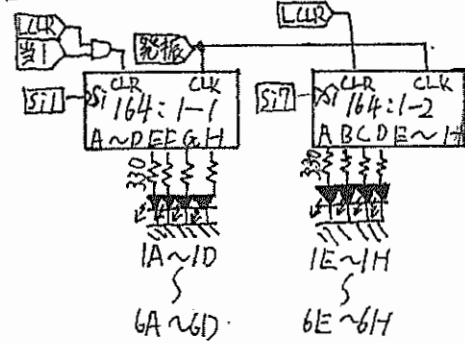
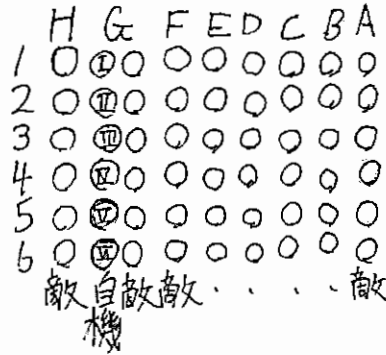
リセット



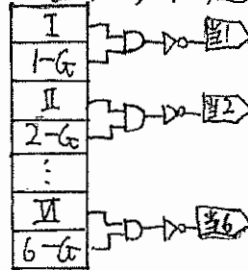
敵の力



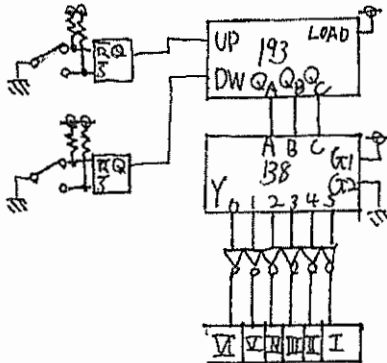
フィールドのLEDの配置



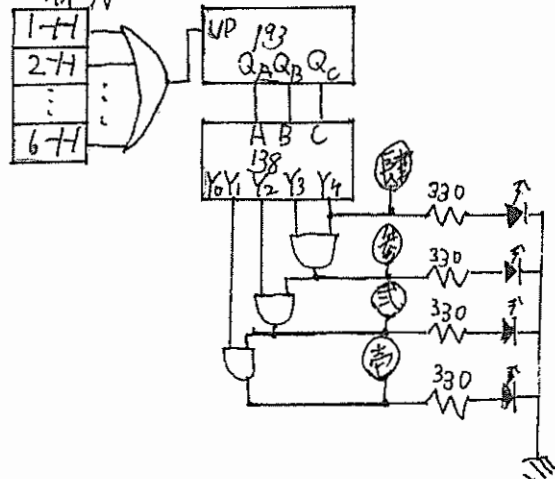
当たり判定



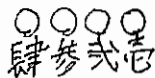
自機移動



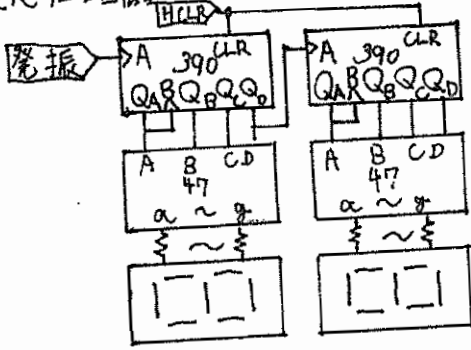
体力



体力LEDの配置



進んだ距離



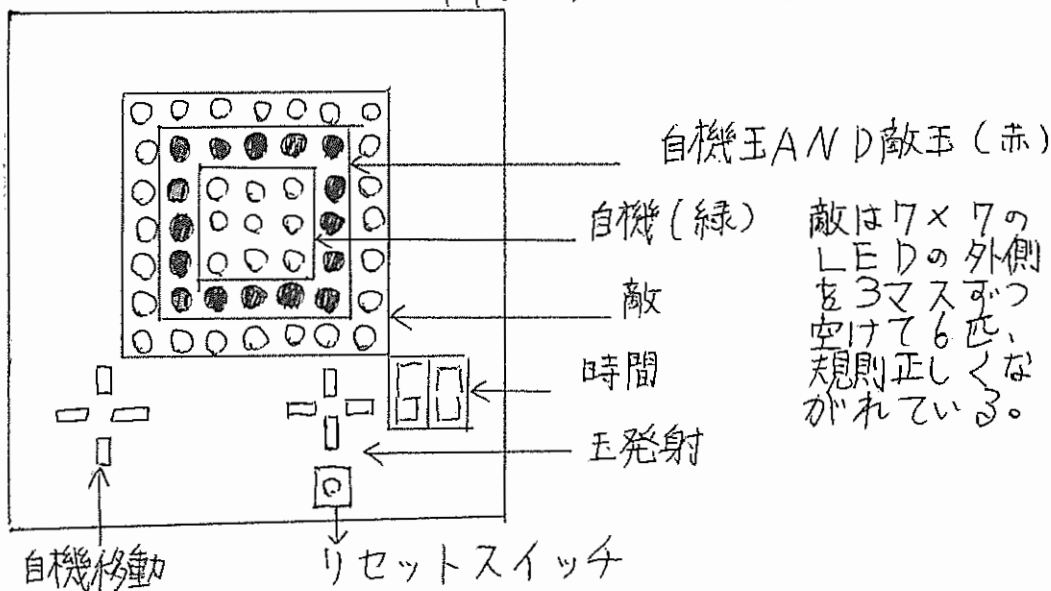
↑ 画素数が多ければ、こうな。ていとかも… (上移動時)

空手戦隊 バカサンジヤー

製作者 M2 梅本
 回路設計 H1 西村さん
 協力者 物無の皆さん

～ゲーム外観やルール

↑ 限りない IC 達

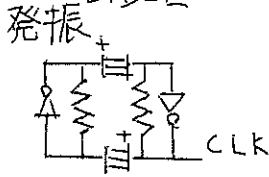


自機は 3x3 のマス を上下左右に動いて玉も発射して打つ。
 敵も打つ玉をよけつつ、60秒のあいだ、6匹ずつの
 敵をたばせばクリアとなります。

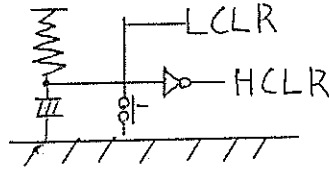
～感想～

配線材が IC の上を平気で横気っているので
 ノイズがのりまくりです。
 来年からは 気をつけます。

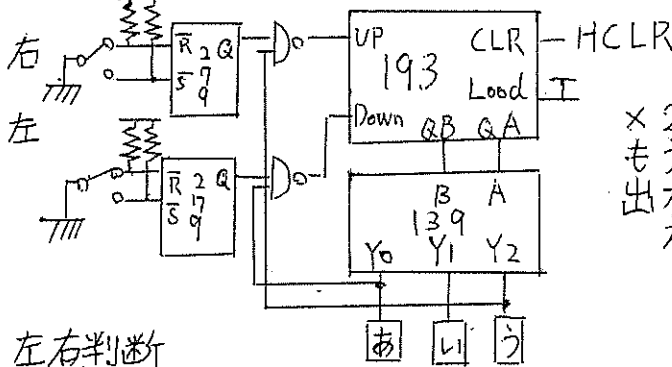
~回路図~



クリア

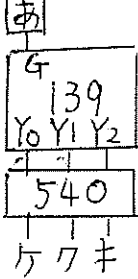


自機移動 4方向

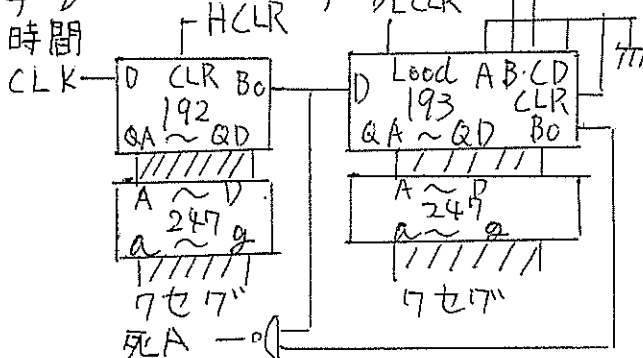
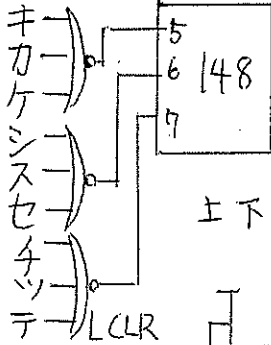
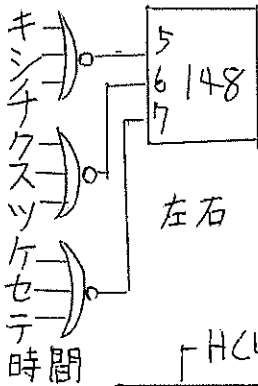


×2 (上下)
もう1つの139の
出力は
え、お、かとする。

左右判断

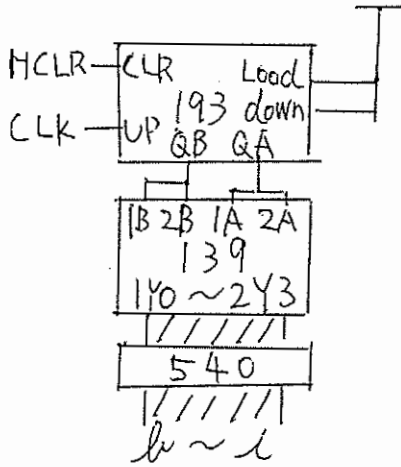


×6 139のGにあ~か6を作る。
出力は順に
(セスシ) (テツチ) (チシキ)
(ツスク) (テセケ)である。
ここでいうカタカナをLE17の
入力として扱う。



左が1の位
右が10の位

敵移動

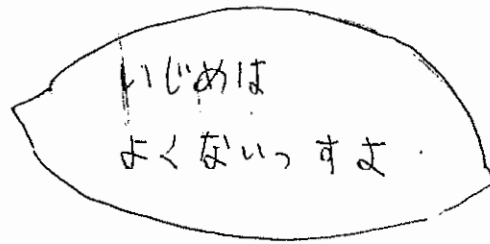
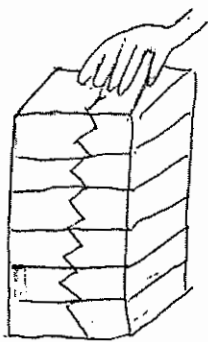


これを j ~ m n ~ q
 r ~ u v ~ a
 の 3 セット 7 くる

LED 表
 a b c d e f g
 x h
 w i
 v j
 u k
 t l
 s r q p o n m

ストーリー

みんなからの暴言をよけ続け
 がんばって生きていきましょう。

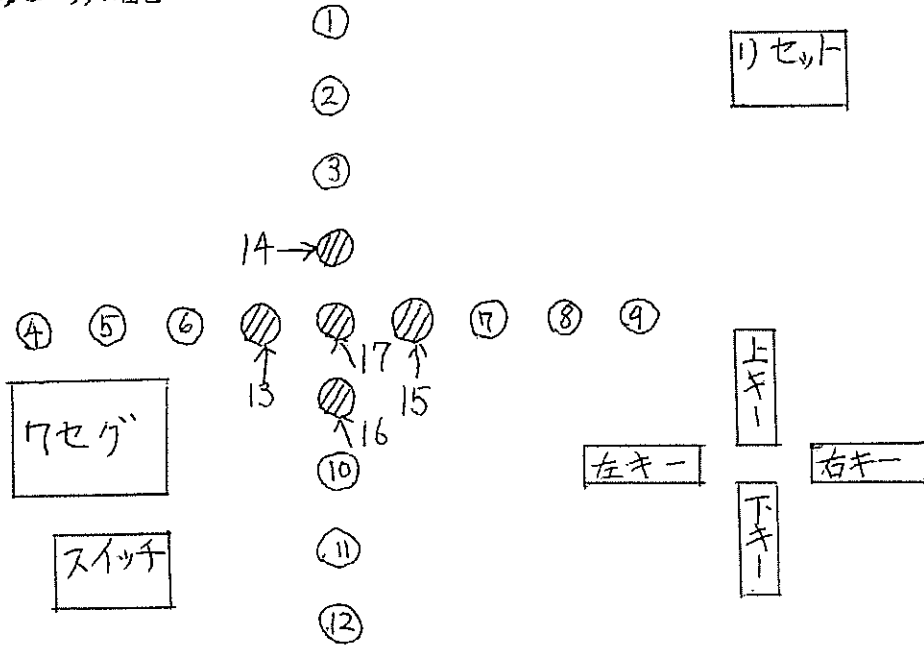


X城危機一発！！

製作者
回路図設計者
協力者

中学二年白水 陽久
高校二年川崎 敦史さん
物理部の皆さま

～回路の外観～



※図上の○はLEDと呼ばれる赤、緑、黄など様々な色に光る(一色)物質を示す。今回は赤。②は二色LEDと呼ばれる、二色に光ることが出来る物を示す。今回は緑と赤。

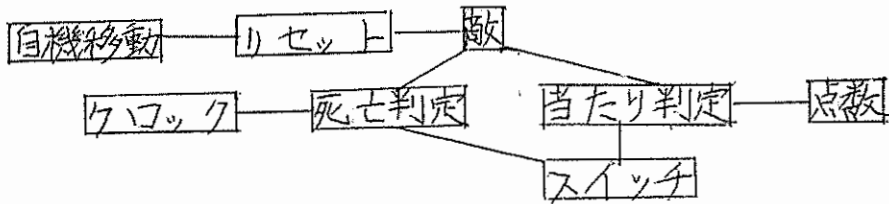
～ルール説明～

自機が動ける範囲は二色LEDの部分の合計5マス。敵は①→③、④→⑥というように真ん中にマスを一つ向かって動いて来る。プレイヤーは上キー、下キー等を押しながら、二色LEDの部分を一マスずつ移動し、向かって来る敵を倒して行く。敵を倒す手段は敵と重なった瞬間に「スイッチ」の部分を押すこと。例えば⑤と⑬に敵が来ており、自分が⑬にいて「スイッチ」を押した時⑤と⑬にいた敵は消える。ただし、⑫に敵が来た時はゲームオーバーになる。自分のスコアは「7セグ」の部分に表示され、「リセット」を押すことで始めの状態に戻る。

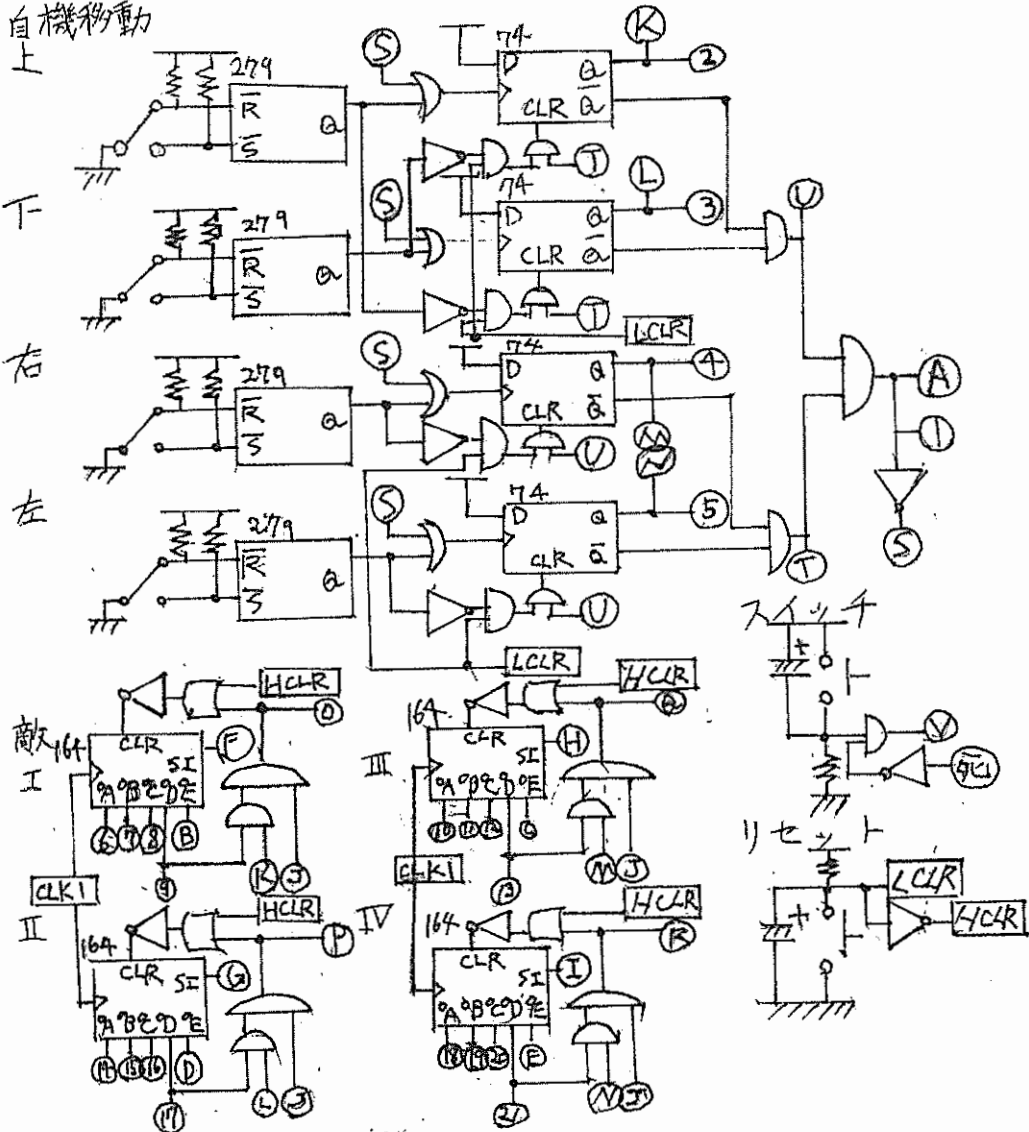
～ストーリー～

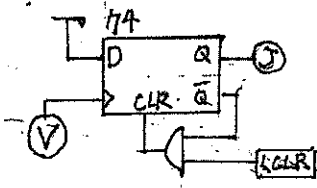
時ははるか昔Xの乱が激しいX城でのこと。X城には美しいX姫がいた。そのため、X姫と結婚したいと思う男は多くいた。ある時、暴力的に結婚をせよと男が現われた。X城の頭主Xである。このXは四方を山に囲まれたX城に四方から攻めこ

うと考えた。X城にスパイ活動をしていたX城のYはこの事をX城の頭主Xに伝えた。Xは山により、四方に分裂させられたX城の戦力には勝てるぞと踏んでV将軍に四方からくる敵を軍勢を率いて倒すように命じた。ゲームをプレイするあなたはV将軍、四方から来る敵を倒せ！！
 ~ブロック図~



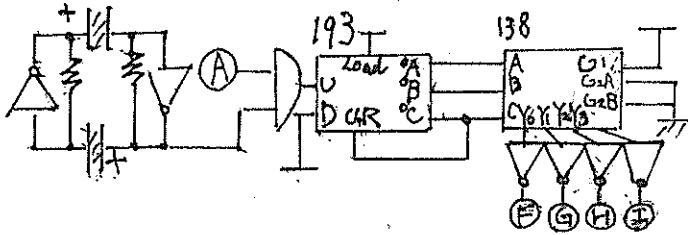
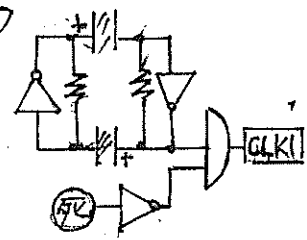
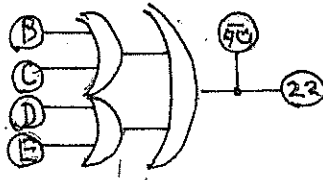
~回路図~



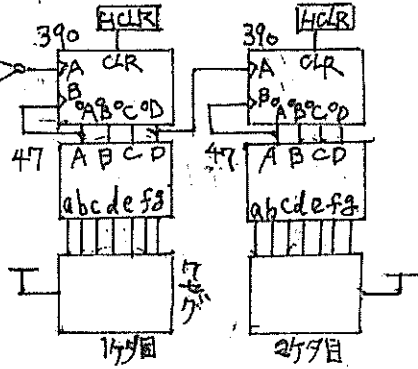
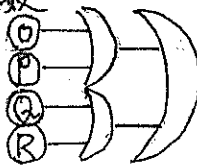


死七判定

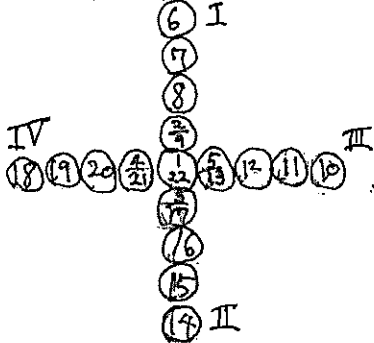
クロック



点数

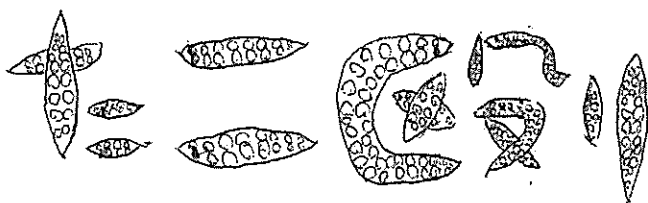


LED配置表



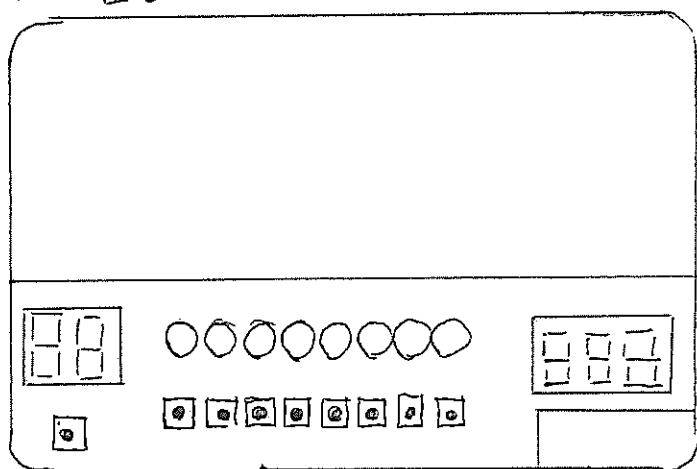
*①は2色LED。上は緑で下は赤に接続。

*回路图中的数字①などはLEDに対応し、アルファベット(A)などは同じ記号の部分がつながっていることを示す。



製作者 M2 鈴木 啓也
 設計者 H2 竹島 純
 協力者 物無 の 様

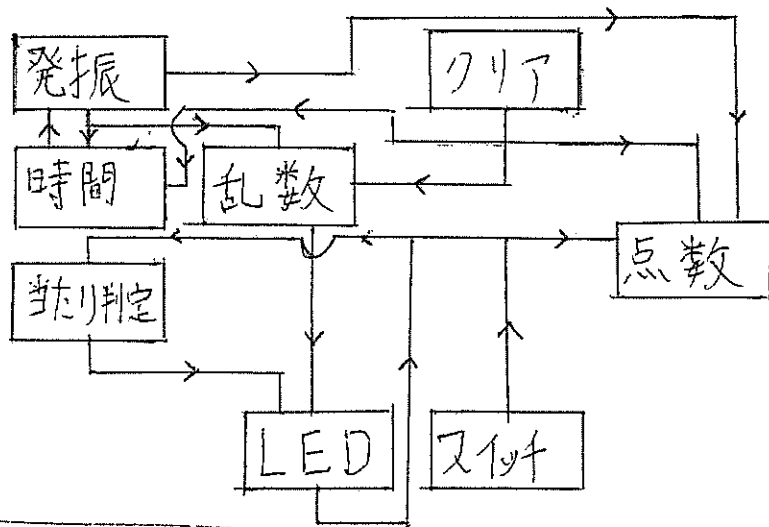
～ 外 観 ～



～ ルール ～

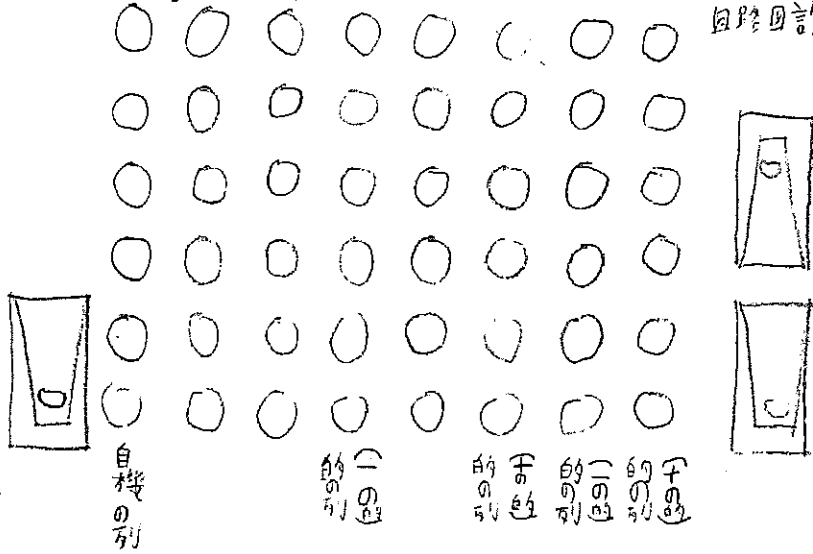
「もぐるたたき」です。点灯したLEDの下のスイッチを
 押して下さい。点灯した瞬間に押すと4点入ります。

～ ブロック図 ～



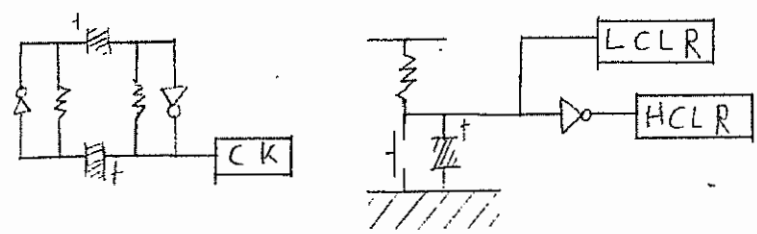
～白勺当てゲーム～

制作者: 藤森
 図案設計者: 川崎さん

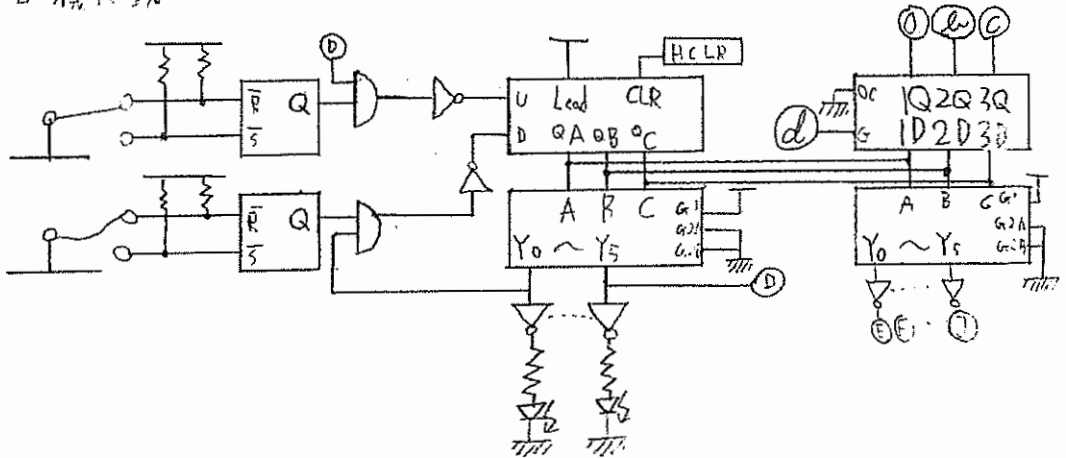


遊び方: 右にある2つのスイッチで自機が右側の列で上下に動き、左のスイッチで右に向かって弓弾を打ちます。その弓弾を右側に流れている敵に当てて行き、1の敵に当てたらマイナス1点、10の敵に当てたらプラス1点になり点数を上げていくゲームです。

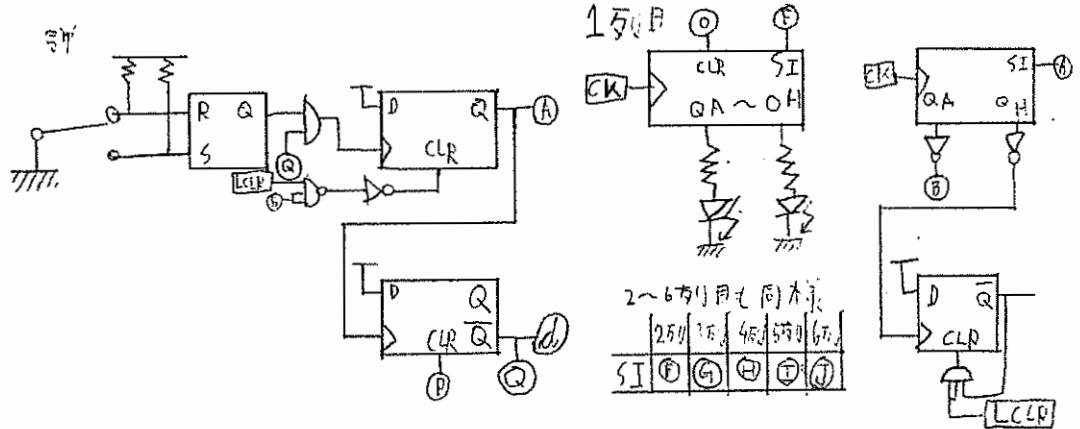
発振



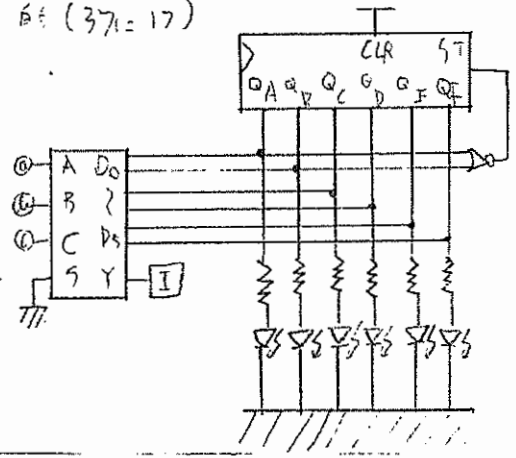
自動作動



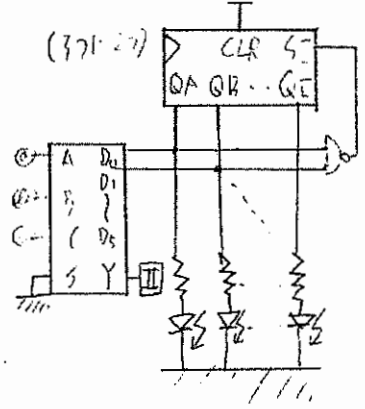
計測

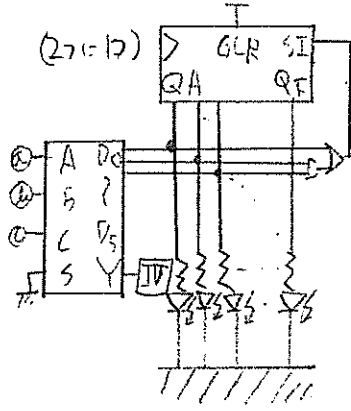
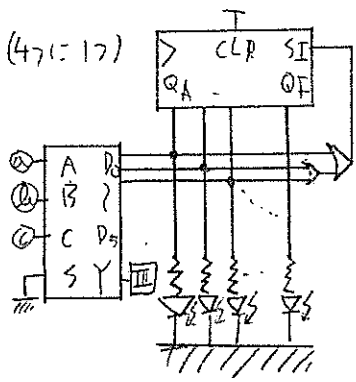


① (371=17)



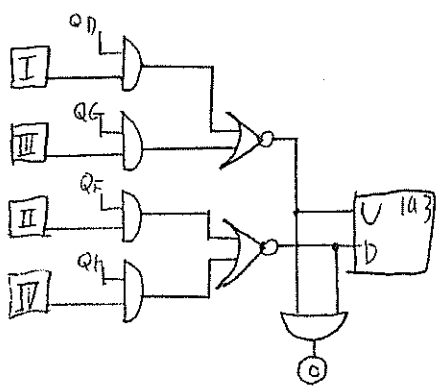
(371=27)



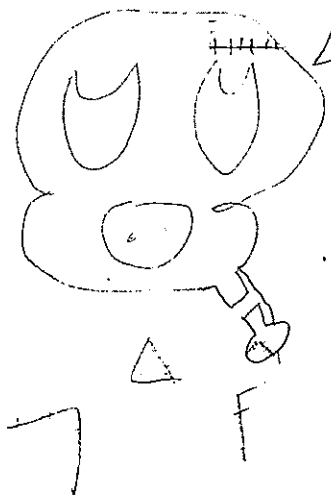


使用したIC

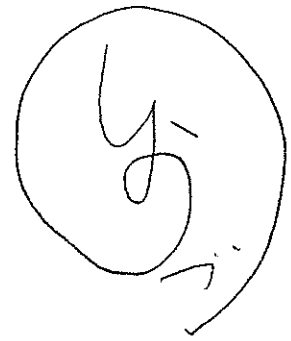
74LS00	6
(74LS02)	02
(74LS08)	08
74LS175	10
	32
	04
	136
	151
	164
	193
	279
	373
	540

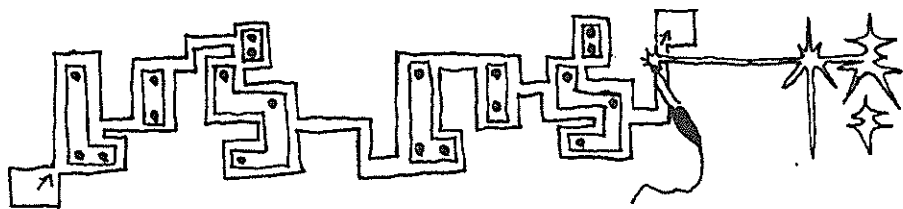


スペースが
足りず



ボクのサインです

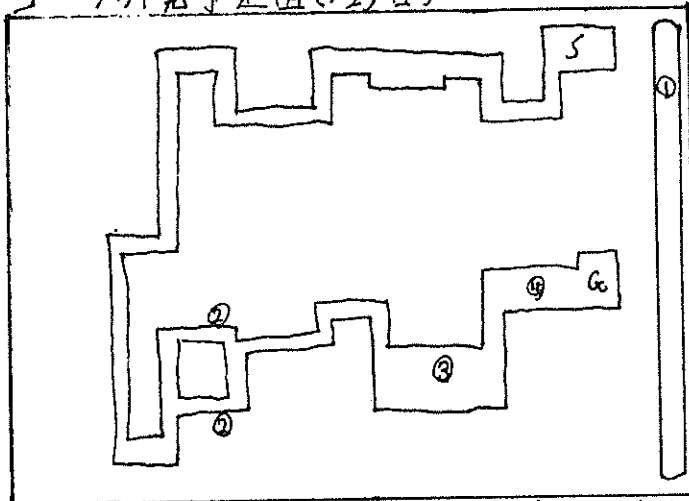




製作者・伊藤 卓 (回路担当)
 ・岩下 義明 (コース担当)

イライラ棒とは、コースの端にぶれないようにしながらスタートからゴールまで行くというもので、コースの端に当たると音が鳴る物とかあります。そしてただ通るだけではつまらないので、ギミックが途中にあります。

コース外見予定図(4/25日)



- ギミック(4/25日)
- ①…棒が追ってくる
 - ②…片方がランダムに止まる
 - ③…通路が少しずつせまくなっていく
 - ④…引きつけられる(電磁石)



← ゴールまで行けてこれくらいの賞金がでたらな。

ギミック・コースは変更されている場合があります。

ギミック

制作者: M2 伊藤 卓

協力者: 物無のみなさま

今年のイラハ棒には、ギミックが3つあります。

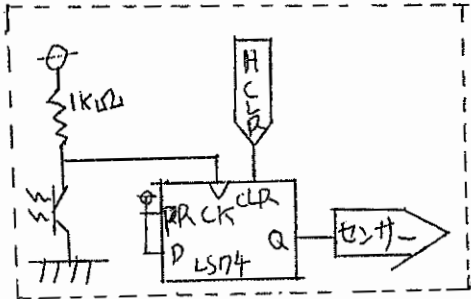
ギミック1 | フォトラという部品を使って、ある一定の場所を通ると、電流が流れている棒が動きだし、ある場所で止まり違う一定の場所を通るとまた棒が反対に動き出すという回路です。

ギミック2 | これにもフォトラという部品を使っています。最初に2つのスイッチのどちらかを押してもらい、途中の別れ道の正解の道とはずれの道が決まります。正解の道を通ると何事もなく進めるが、はずれの道を進むと道が閉じて反対の道を進まなければならなくなる、という回路です。

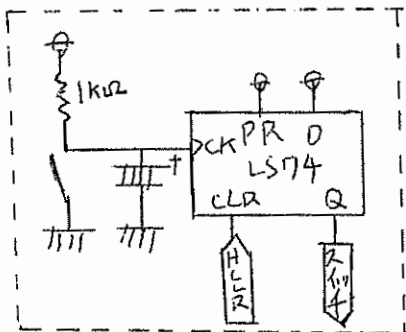
ギミック3 | これもフォトラを使っています。ある一定の場所を通ると、出口の前の道が狭くなり、女台め最終的にはとても狭くなり、難易度が上がるという回路です。

回路各図

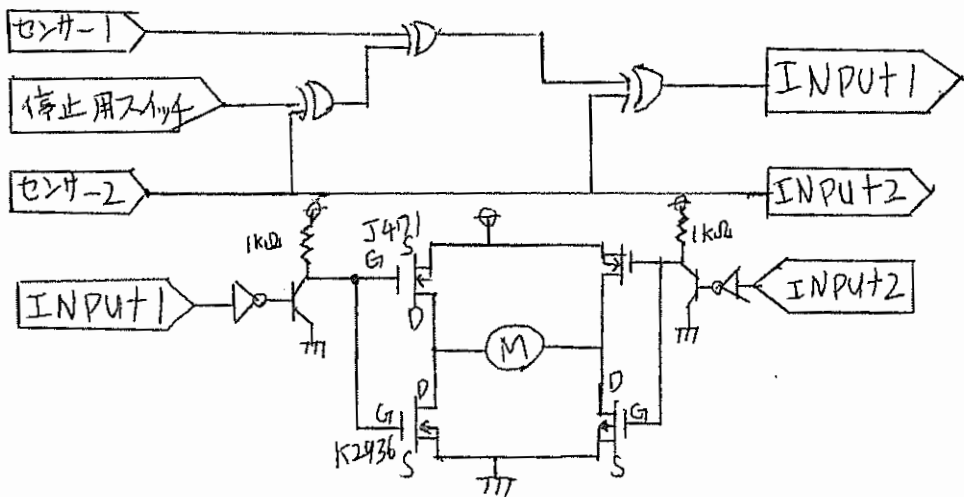
ギミック1



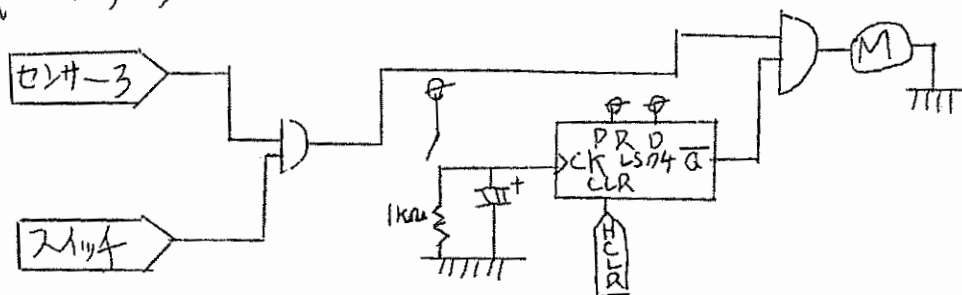
この回路をセンサーの回路とする。



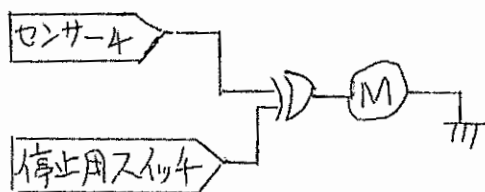
この回路をスイッチの回路とする。



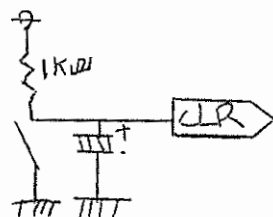
ギミック2



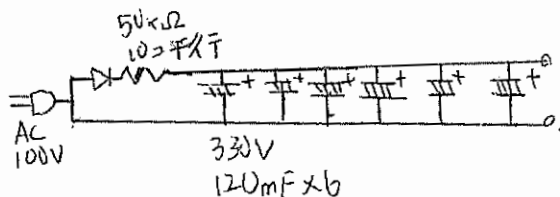
ギミック3



CLRスイッチ

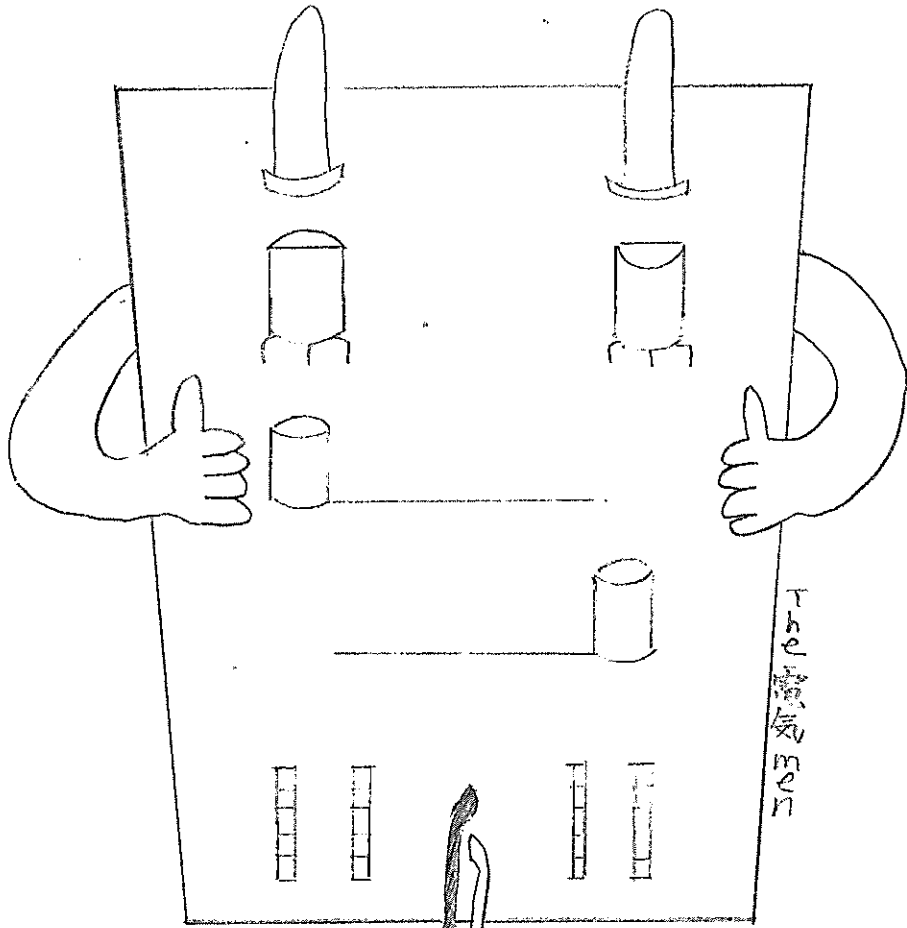


フラッシュ回路... フラッシュ回路とは棒が壁に当たった時に火花を散らす回路です。

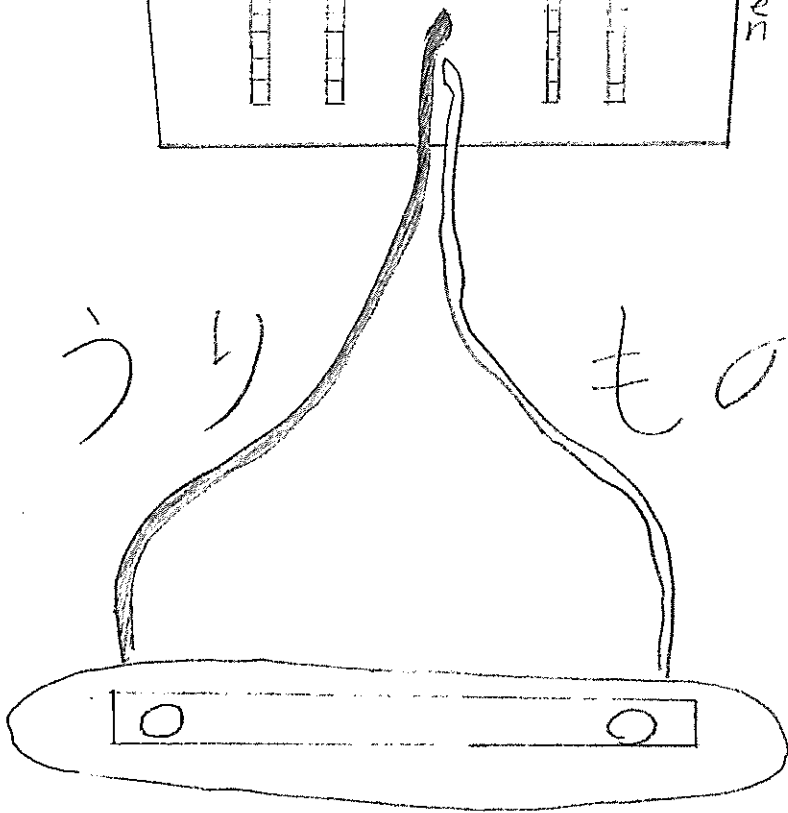


感想

FETが熱くなって、モーターが動かなかったりと苦労します。
この回路は作るのが難しいと思いました。



うりもの

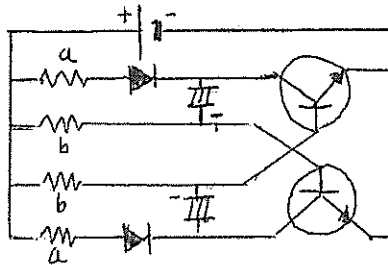


ちかちか

ちかちかとは？

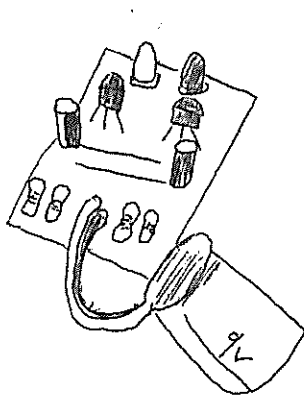
ちかちかとは、「自走マルチバイブレーター」の事である。
 コンデンサーという電子部品に電気をため、左右について
 いるLEDに交互に電気を流し、交互に点滅します。
 非常に簡単なので初心者でも作れることのできるです。

回路



トランジスタは 2SC1815を
 使っています。
 抵抗値は $1k\Omega$ ・ 330Ω です。
 コンデンサーの容量を
 変えると点滅の速度も
 変わります。

外観



基板

これに用する基板は、
 物理部無線班の
 お手製です。
 大事に扱っていただけれ
 ば光栄です。

Rennda Game

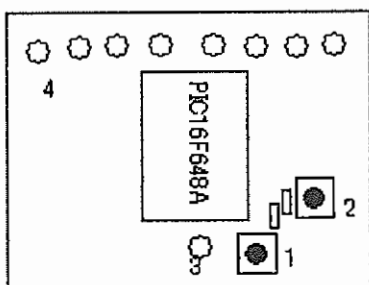
M3 中岡 勇太

今回の売り物は、一言で言うと連打ゲーです。

—ゲーム概要—

一つの LED が光っている間にいったいどれだけの回数連打できるかというものです。レベルが8つあり、レベルが上がるほど連打しなければならない回数が増えていきます。そして、連打回数が足りないとゲームオーバーになってしまいます。さて、あなたは、いったいどのレベルまでいけるでしょうか？

—外観—



1、連打スイッチ

このスイッチを連打します。

2、クリアスイッチ

このスイッチは、ゲームオーバーになったときに初めからやり直すときに使います。

3、指定 LED

この LED が光っているときに連打します。

4、レベル表示 LED

この LED が光っている数が現在のレベルを表します。

—プログラム—

池上氏に書いていただきました。

```
//16F627A を使う
#include "16F627a.h"
//モード
#define HS, NOWDT, PUT, MCLR, NOLVP
//ディレイ設定
#define delay(clock=4000000)
//fast_io を使う
#define fast_io(A)
#define fast_io(B)

//起動時の待機時間(ms)
#define first_wait_time 4096
//入力のチェック回数(ms)
#define game_wait_time 4096

//スコアとレベルの対応 (各レベルのスコア最低値)
#define score_level_1 10
#define score_level_2 12
#define score_level_3 15
#define score_level_4 20
#define score_level_5 28
#define score_level_6 40
#define score_level_7 50
```

```
//関数
void init();
long game0;
void level_correct(long score);
void level_print();
void freeze();
void game_over();

//レベルの数値
//初期は 1、最大は 8、0 でゲームオーバー
int level = 1;

//main 関数
void main(){
    //初期設定
    init();
    //ゲームを開始
    while(1){
        level_correct(game0);
    }
}

//初期設定と起動確認
```

```

void init(){
    //ポート A を全て出力に
    set_tris_a(0);
    //ポート B を全て出力に
    set_tris_b(0);
    //ポート A を全て L に
    output_a(0);
    //タイマ 0 の設定 RA4 の立ち
    上がりでカウント
    setup_timer_0(RTCC_EXT_L_
    TO_H);
    //ポート B 全てを H に
    output_b(0xFF);
    //0.5s 待機する
    delay_ms(500);
    //ポート B 全てを L に
    output_b(0);
    ///レベルを 1 に
    //level = 1;
    //レベルを表示
    level_print();
}

//連打を数える
long game0(){
    //タイマ 0 の初期化
    set_timer0(0);
    //起動時の待機
    delay_ms(first_wait_time);
    //LED を光らせる
    output_high(PIN_A0);
    //タイマ 0 の入力を許可
    output_high(PIN_A4);
    //game_wait_time ms 待つ
    delay_ms(game_wait_time);
    //タイマ 0 の入力を停止
    output_low(PIN_A4);
    //LED を消す
    output_low(PIN_A0);
    //タイマ 0 の値を返す
    return get_timer0();
}

//スコアに応じてレベルを変更する
void level_correct(long score){
    //各レベルにおいてスコアが最
    小値に達しているかどうかを判定
    //達していればレベル 1 アッ
    プ、達していなければゲームオーバー
    switch(level){
        case 0:
            break;
        case 1:
            if(score <
score_level_1){
                game_over();
            }else{
                level++;
            }
            break;
        case 2:
            if(score <

```

```

score_level_2){
            game_over();
        }else{
            level++;
        }
        break;
        case 3:
            if(score <
score_level_3){
                game_over();
            }else{
                level++;
            }
            break;
        case 4:
            if(score <
score_level_4){
                game_over();
            }else{
                level++;
            }
            break;
        case 5:
            if(score <
score_level_5){
                game_over();
            }else{
                level++;
            }
            break;
        case 6:
            if(score <
score_level_6){
                game_over();
            }else{
                level++;
            }
            break;
        case 7:
            if(score <
score_level_7){
                game_over();
            }else{
                level++;
            }
            game_over();
        }
        break;

```

```

        default:
        break;
    }
    //LEDにレベルを表示
    level_print0;
}

//出されたレベルを表示
void level_print0{
    //ポート B に出力するビット
    int printbit = 0;

    int i = 0;
    //printbit を 1 左シフトして 1 足すのを、レベルの数だけ繰り返す
    while(i < level){

        shift_left(&printbit, 1, 1);
        i++;
    }
    //printbit をポート B に出力
    output_b(printbit);
}

```

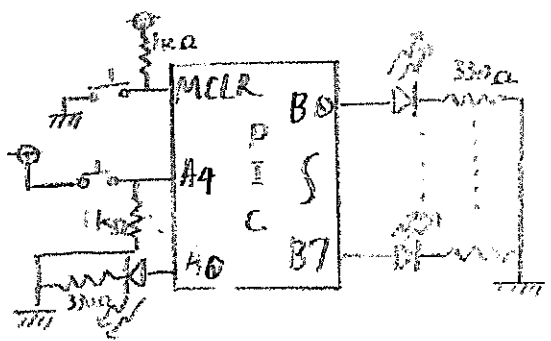
```

//クリア後の処理
void freeze0{
    while(1){
        //待機
        delay_ms(1);
    }
}

//ゲームオーバー時の処理
void game_over0{
    while(1){
        level_print0;
        delay_ms(500);
        output_b(0);
        delay_ms(500);
    }
}

```

—回路図—



それでは最後に、定番の『みんなへのメッセージ』を。

M1・・・ようこそ物理部無線班へ！この部活には、夢が詰まっています。素直に、楽しんでください。

M2・・・やりたい事をどんどん見つけて、みんなに提案して行くと、物無がもっと楽しくなると思います。先輩に対して調子に乗りすぎるのはやめようね。

M3・・・これから、責任者とかで大変になってくると思います。だからこそ、やりがいがあるというものです。楽しむという事を忘れず、自然体で頑張っていきましょう。

H1・・・あっという間に、最高学年は終わります。やりたい事があるなら、それを一番叶えられる学年なので頑張ってください。最も楽しい1年を駆け抜けてください。

H2・・・なんか、あっという間にここまで来てしまいました。なんか色々あったけど、川崎と竹島がいてくれて、俺の物無人生はすっごい充実したものになった。本当にかげがえのない友達です。本当にありがとう！残りわずかだけど、いっしょに頑張っていこう！

俺は、この物理部無線班を忘れない！

みんな、本当にありがとう！

総務 中嶋剛大

麻布学園 物理部無線班
回路図集 2008

- Electric Circuits Collection 2008 -

2008年 4月 28日 発行

編集責任者 : 中嶋 剛大

発行者 : 麻布学園物理部無線班

発行所 : 麻布学園第二応接室

落丁・古丁 など ありましたら おとりかえいたします。

また、この本に関するご質問などは 麻布学園物理部無線班まで お願いします。

～ 無断転載・複写を禁ず～

Design

表紙・裏表紙 : 池上 涼平

他デザインなど : 中嶋 剛大

渡邊 翔

竹下 諒

相迎 駿介

岩下 義明

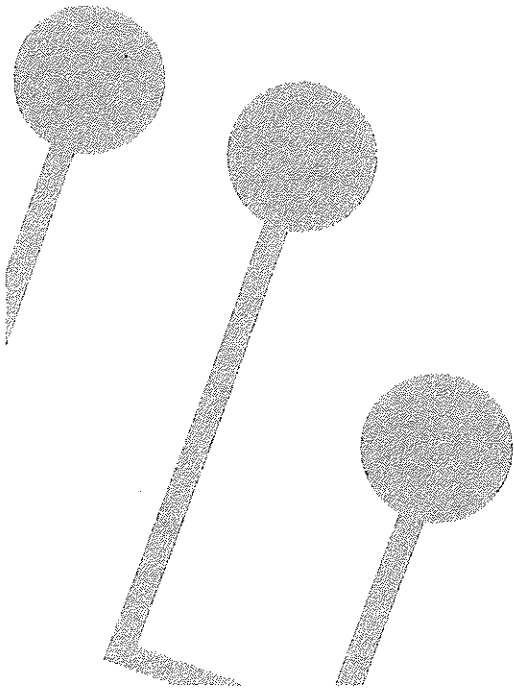
白水 陽久

Schematic

Collections

2008年度 回路図集

本書は、麻布学園物理部無線班の部員が1年間研究、
制作した作品の内容についてがまとめてあります。
ただし、未完成のものも多く含まれており、書いてあ
る内容については保証できません。あらかじめご了承ください。



定価：
100円