

2005年度
物理部無線班
回路図集

はじめに

『CPUは64 bit となり Athlon64
3400+ など……』と 今年の
回路図集 がありましたか、
とうとう 3GHz 超えは普通となり
後少して 4GHz にも達しそうな
勢いです。時代の最先端に追いつく
ことは できないのでしょうか。

この回路図集には 麻布学園物理部
無線班の部員が一年間 研究、
もしくは 製作したものが記されて
います。中には 未完成のものも
あるため、その通りに作ってもうまく
いかない場合もありますが、当方は
一切責任を負いかねます。

あらかじめご了承ください。

麻布学園物理部無線班
会計: 舟渡隆平

目次

・ はじめに			2
・ 目次			3
・ コンピューター系			
Bamboo stalkers	H1	浅井	6
ultimate weapon in the North	M3	宮田	11
・ ロボット系			
メーヴェ	H2	森	14
	M3	井上	
Titan	H2	舟渡	32
	H1	伊藤	
	M3	安部	
潜水停カッター	H1	西脇	43
CLIMBER	M2	竹島	47
	M2	高崎	
・ デジタル系			
弾出るロボット	M2	大塚	50
コボタン	M2	川崎	52
緑帽子を救え	M2	木村	55
成太の誤流娛13	M2	桐山	58

D K、改

M2 佐野 61

救助艇

M2 高崎 64

戦闘艇侵攻阻止

M2 竹島 67

逃げろ!! 五浪くん!!

M2 中嶋 70

百人斬

M2 中野 72

いさ いさ 棒

M2 大塚 75

M2 佐野

先生

M2 川崎 78

M2 木村

M2 中嶋

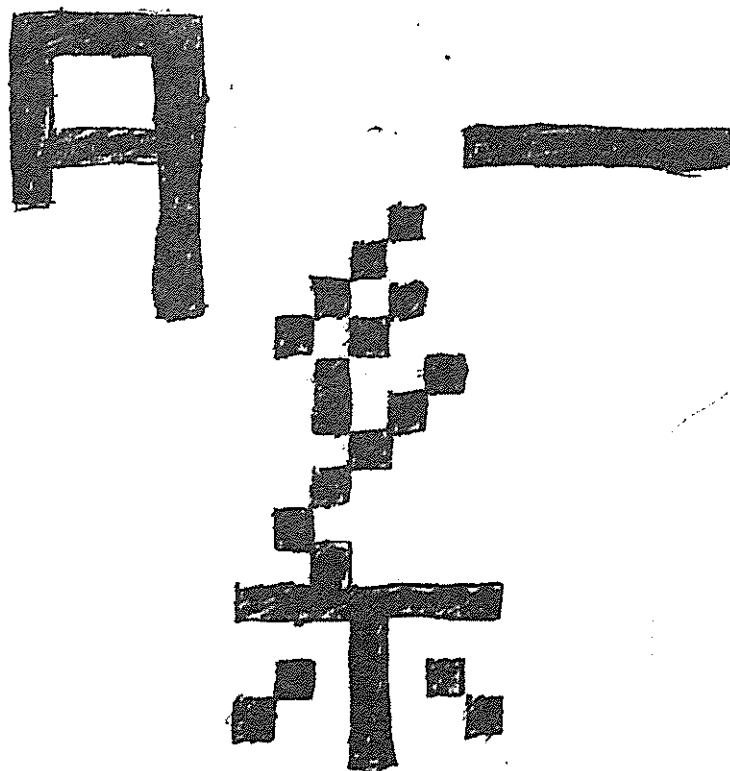
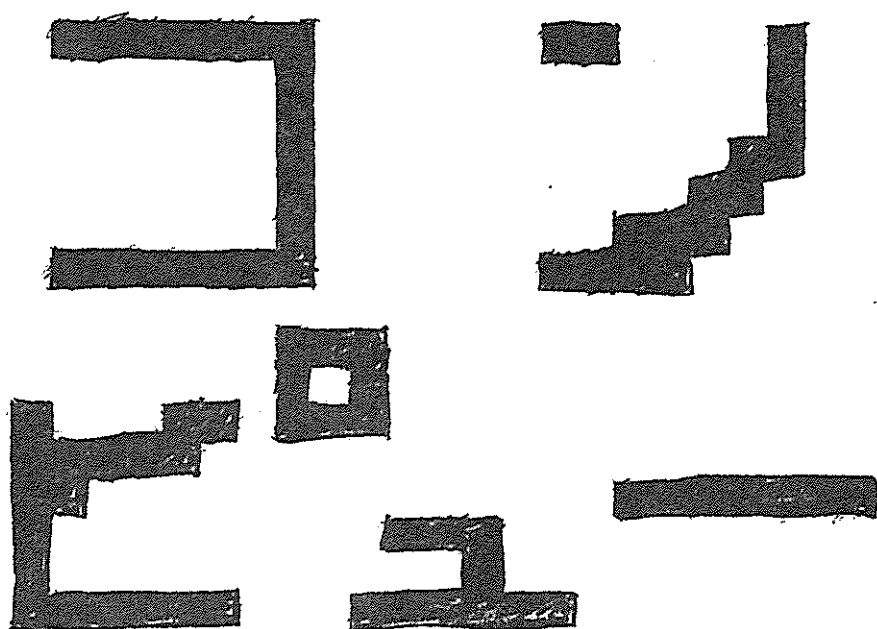
。売り物

the 反射

M3 井上 83

M2 竹島

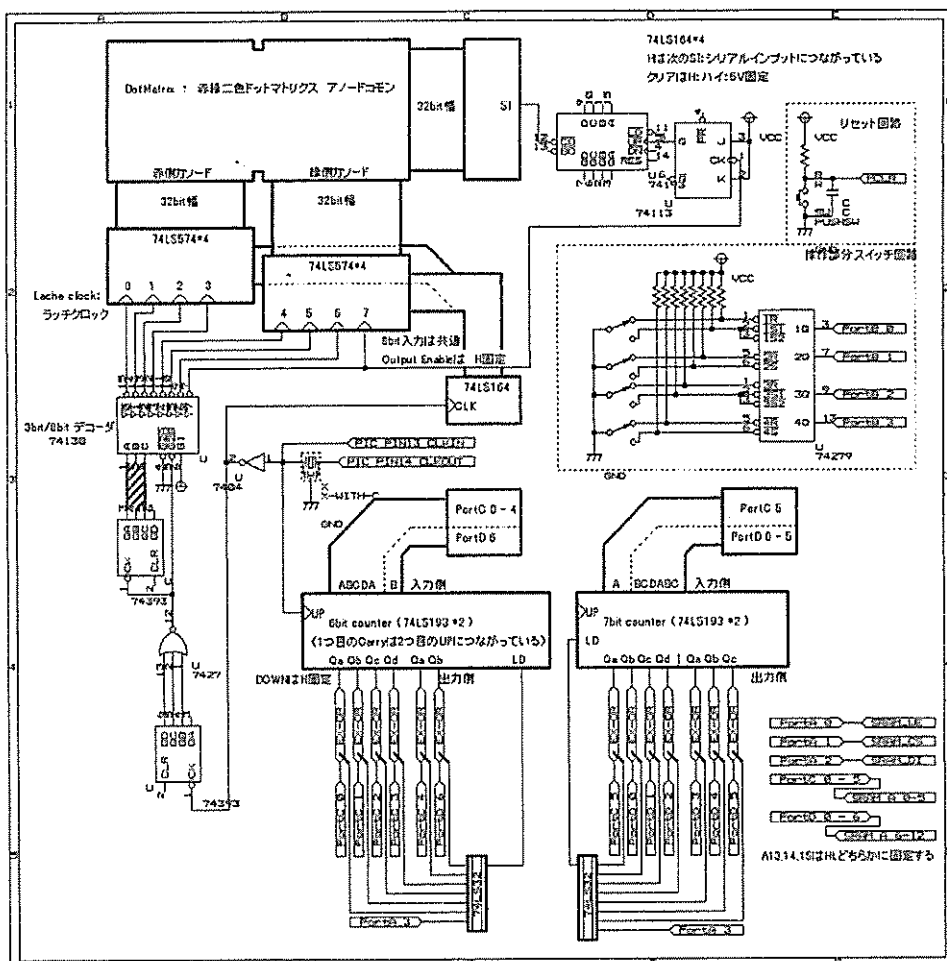
。後書き



Bamboo Stalkers !

設計・企画・作成：高1-7 浅井政太郎

・回路図(なかばブロック図になっているが…)



・どんなゲーム？

追いかけてくるストーカーたちを放射能や溶岩の壁に落として、あいつらから自分の身の体裁を守るゲームだ！ 僕は、64*64マスの危険な迷路に紛れ込んだかわいそうな子羊ちゃん。せいぜい逃げ回ることしかできないけど、相手は所詮ノーマンが腐ってる気違いストーカーだから、心配しなくても逃げ切ることができる。画面外から襲ってくる面倒な奴等もいるぞ！ 主人公は画面の中心にいて、ひたすら迷路を逃げるのだ。もちろん迷路の壁は放射性物質や溶岩でできていて、触るだけで自分の体力の八分の一が削がれてしまう!!

つまり、一回壁に触るごとに、八分の五体ずつ不満足になっていくんだ！まあ、敵はさらにやってくるから、地獄は終わらない。さて君は、この極限状態で何秒逃げ切れるかな？

・メインボードの性能・コンセプト

「抜群の性能をもった CPU」

CPU(?)は、有名な小型マイコンを生み出してきた Microchip 社の PIC16F877 だ！最新のパソコンに入っている CPU にも張り合えるすごいやつだ。

クロックは 20Mhz で動かしており、また SRAM には、M5M5258 を使っている。ちなみに電源は 5V で、IC はぜんぶ 74LS だ！F は使ってないぞ。

グラフィックは、32*32で、16*16のドットマトリクスが4個。マップは64*64*なのにご心配なく。高性能CPUのおかげで、画面外にいる敵を想定したゲームになっている。

「流行に乗ったつमりのスタイル」

躯体全体の構造は、某○天堂が売り始めた、タッチパネルがついていて画面が二つの某ポータブルゲーム機に似るように、真ん中で折り曲がるようになっている。基盤が2枚+1(制御パネル)なので、真ん中に蝶番がついているのだ。

「モバイル性能抜群」

充電電池を使えば、恥ずかしいだろうけど、外に持っていくこともできると思う、たぶん！重さは 100g 切っている。それに、がんばれば・・・ポケットにも入ると思う。

・回路の解説

PIC16F877 には、ポート A が六つ、ポート B,C,D がそれぞれ 8 つ、ポート E が 3 つの出入力ピンを持っている。今回の回路では、ポート A を入力に、ポート B をフラグ出力に、ポート C,D を SRAM へのアクセスのためのアドレスピンとした。

① ポート A

ここには、A0～A3 まで上下左右のスイッチの出力がつながっている。

押している間だけ L というスイッチである。

② ポート B

B0 には、SRAM の WE ピンがつながっている。

B1 には、SRAM の CS ピンがつながっている。

B2 には、SRAM の DI ピンがつながっている。

B3 には、74LS193 のロードにつながっている。

③ ポート CD

ポート C の 0～5 ビット目には、自分のマップ上での x 座標から 16 を引いたものが、

ポート C の 6～7 ビット目と D の 0～3 ビット目にはじぶんのマップ上の y 座標から 16 を引いたものが代入されているようにできている。

④ カウンタ(74LS193)

PIC が演算を行っている間、ドットマトリクスを光らせる機構のひとつである。

じぶんの座標の 16 個ずつ左上の部分から順々に出力していくよう、カウントアップした数値を SRAM のデータポートにつなげている。

⑤ シフトレジスタ(74LS164)

同じく PIC が演算を行っている間、ドットマトリクスを光らせる機構のひとつである。

SI(シリアルインプット)には 32 回に一回 H が来る様にされていたため、LED がひかるのは一列だけである。

・ プログラム解説

開発言語は C 言語。CCS 社の C コンパイラを使った。プログラムのフローチャートを下に示す。

SRAM にマップデータを送る

(以下繰り返し)

自分死亡判定 ……①

敵死亡判定

スイッチで自分の方向を決める ……②

AIに自分のほうを向かせる ……③

自分と敵を向いている方向に一マス進ませる

進んだ先に壁があるか判定(あるならダメージ、壁に跳ね返っているような方向を向く)

ディスプレイに自分・敵の座標を映す

じぶんの位置を中心に壁を表示する

(繰り返しここまで)

となっている。さらに解説するところには、印を入れた。

①… 敵も見方も、体力は8である。壁に当たるたびに-1され、体力がゼロになってしまうと、ゲームオーバーになってしまう。

②… スピードが上がると、自分も敵も、勢いがついているので曲がりにくいようになっている。応用すれば、プログラム次第でレースゲームのようなものを作り出すこともできることになるかも知れない。

③… まず、敵から見て自分がどの方向にいるのかを求める。そして今どちらを向いているのかをしらべ、今の速さに従って、曲がる角度も調整される。壁に当たって跳ね返ったときはどのように動いていくべきなのかもここで求められる。(垂直にあたれば垂直に跳ね返る。斜めからあたれば斜めに跳ね返る。)

つまりここは、プログラムの中でもっとも複雑な要素が絡んでくるような部分なのである。しかし、AIのコンピュータプレイヤーたちは、目の前に壁があっても壁をよけようとしなない。これは、このプログラムの仕様である。そうでもしなければゲームの難易度が難しすぎるのだ。

・ この製作物を作った感想と、反省点

んん！じつに楽しかった！この製作物には、とりあえずこれまで、私が活動した経験が全て使われている。復習にもなったし、技術の向上に自己満足したりもしたのだ。僕がこの「電子ゲーム」系の製作物を作った理由。それは、動かなかった中一・中二の製作物が、すべて稼動部があるメカ物であったからだ。「100%動く」物であり(去年は動かなかった製作物もたくさんあって、本当にむなしかった)、なおかつやりがいがあり、技術が必要だったのは、このようないわゆる「原点」に戻った、しかしPICを使っているために応用性も持つ「電子ゲーム」系であったのだった。反省点としては、個人的には、ポート・データ入出力の部分の仕様が、活動の後半期になるまではっきりと決まらず、その上で何度か変更になったのが、プログラムの開発にも、本体の製作にも悪影響をもたらしていたのではないかと考えている。実際4月になってデバックに躍起になりだしていたので、ともかく遅かった。ためしに表示するプログラムでも何でも、動く本体をまずは作らなくてはならなかった。何の原因もなく、PIC がいきなりの不調のせいで発熱したため、PIC16F877 を計 2 個も殺してしまった(PIC16F877 は 500 円で、秋葉原の秋月通商で買えます)。これは 877 自体のバグであるらしい。今この回路図集を書いているときは、もうすでに回路のデバッグは大方終わったところである。プログラムを難しいものにするたびに変なバグが怒るので、個人的にはうんざりである……。

Ultimate weapon in the



Vox

製作 m3
宮田 悠
協力 ④の皆様
OB様

1. ストーリー

20XX年、将軍様は窮地に立たされていた。アロカの手がせまり、すてに国内では脱出者が続出、ついに人口が半分にまで減った。絶体絶命と思いきや、将軍様には秘策があった。それは、全世界に低周波を流し、人々を洗脳しようというものだった。将軍様は早速工作員に、日本から技術者を拉致して来る事を命じ、ある日(国)のMは、北〇〇〇と拉致された。こうして「と名付けられたこの装置は世界を恐怖におとしいれたのだった。

2. 仕組み、機能

機能① ピアノのようにある鍵盤をみたしたスイッチを押すと、その音が出るということです。16個の好きな音を出せます。

② スピーカーから音を出して、つなげて曲を作ることもできます。(もちろん低周波も)

音の出る仕組み

まず、スピーカーが音を出す仕組みを説明します。

図1のように「H」が流れると、電磁石になった1が2に近づき「L」が流れるとはなれます。これをくり返すと、空気が振動し音が出ます。

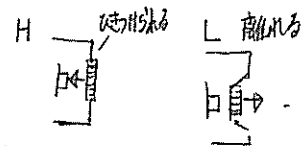


図1

そして、HとLを交互に流すのをPICでやります。ただHもLも数秒流してからLに変え数秒流す事をくり返すだけです。すると図2のように出カピンから出がされます。



図2

当初は図2の波形でけちゃんとした音は出ないと思い、段階的に電圧を変えて波形を整えるDA変換という物を使おうとしたが、図2でもちゃんと音が出たのでやめました。

3. プログラム

では、どのようにしてHとLを繰り返して出力するのでしょうか。それは簡単で、次のようなプログラムを組みればいいのです。そして、音を出す時、次のように指定音でプログラムが900行くらい使ってしまうので、(書きこめるのは1000行まで)、④のようにサブルーチンをCALLで呼び出すように、それでも900行くらいだったんで、④のようにしました。

その前に、一番最初のプログラムを。 MAIN BSF. PORTA, 0 すぐには頭悪いのでやめました。

```

NOP
...
BCF PORTA, 0
NOP
NOP
...
NOP
GOTO MAIN

```

次から縦に2行になります。

①は次のような感じですが、

```

MAIN BSF PORTA, 0
      MOVLW D'186
      MOVWF BCH ; BCHに186を格納
OTO   DECF BCH ; BCHをカウンタに
      GOTO OTO ; (1減らす)し、結果が
      BCF PORTA, 0 ; 0なら次の命令を
      MOVLW D'186 ; とはずす
      MOVWF BCH
SOUND DECF BCH
      GOTO SOUND
      GOTO MAIN

```

↑上ではちょうど“う”の音が出ます。

② CALL C4 → “ド”の音のことです
CALL E4 この場合
1ミで音が変ります。

```

C4 MOVLW D'200
   MOVWF SOUND1
C4CALL DECF SOUND1
      GOTO C4TOCYU
      RETURN
C4TOCYU BSF PORTA, 0
      ...

```

→これを何個も作り音の高さを決めておくので900行位のプログラムになってしまいます。

とあては①の要領です。

要は、①の動作を200回くり返すという事です。
これにより、音の長さが決まります。

③は、①の BCHに入る数を変えるだけで、サブルーチンの中で何一つサブルーチンを呼び出す事になります。こうして全体が500行に減り、曲が長く作れるようになりました。

4. 最後に

下手な説明の上、字が汚く、文字も細かくて、すいません。ですが苦情は受け付けません。御了承下さい。あと、御協力して下さいたすべての方々、ありがとうございます。進行状況ですが、今③のプログラムで音が変れないという問題があり、簡単なプログラムながら結構大変です。しかし、なんとか動きそうです。

ここまで読んでいただき、誠にありがとうございました。

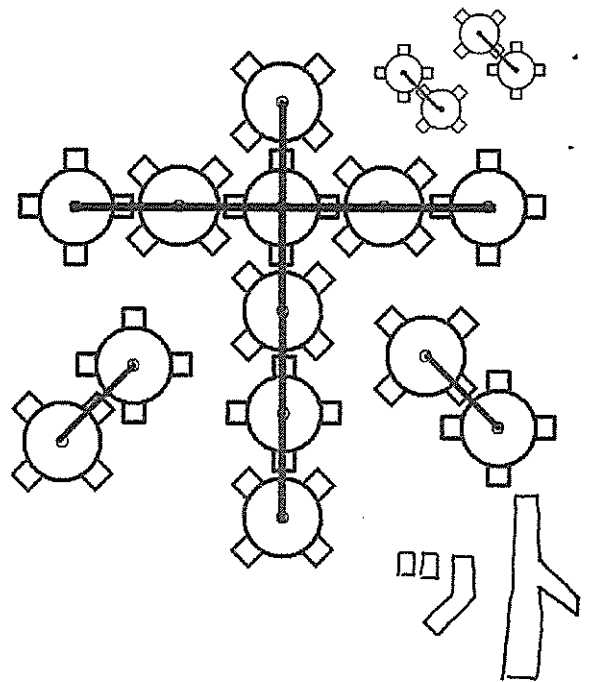
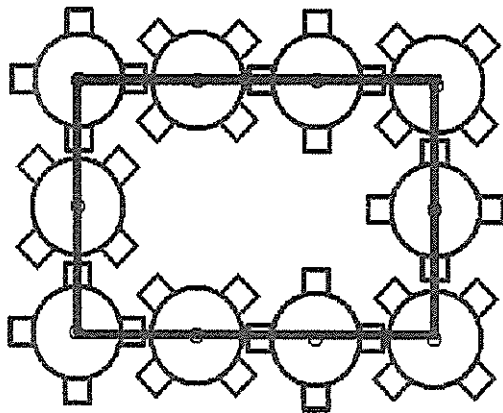
完

～よくある質問と答え～

Q. 低周波は出せるそうだが、超音波はどうなのか？

A. 上記のプログラムにありますように、高さはBCHに入る数字で決まり、大抵は低くなります。よって“1”を入れてためた所、超音波には達しませんでした。

Q. いつごろ完成するの？ A. 文化祭前までに動いているでしょう。



メーヴェ

～身を粉にして飛ぶ子供達～

製作者

H2 森 悠太

M3 井上 忠明

概要

鳥型ロボット

協力

④のOB様、現役のみなさん

これは、地面とメーヴェ家（鳥型ロボット）との半年にわたる決闘の歴史である。かつて、地面という地球上の頂点を極めた物体があった。地面はその身に降ってくる鳥たちをことごとく粉砕していき、数々の模型飛行機達を絶滅に追いやっていった。そこである一族が立ち上がった。その一族の名前こそ『メーヴェ家』である。

History

古田……言わずと知れた初代紙飛行機。自分たち一族がどこまでいけるかを試す。目標半ばで死す。

謎の鳥X(右下写真)……とにかく謎の鳥。古田が頑張っている間は刑務所にいたとかいないとか。人知れず死んでいた。

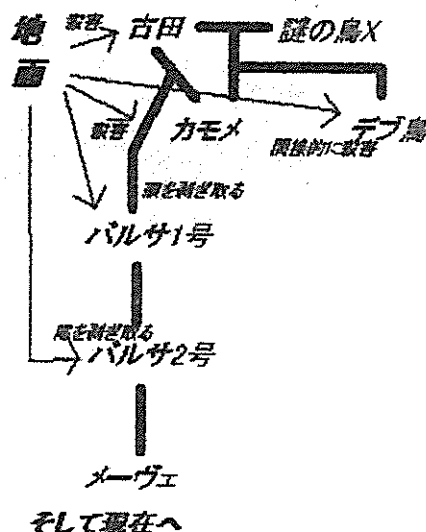
カモメ……古田の息子。一段と羽に強化が加えられ、パワーアップ。もはや図書館塔につっこむなどは当たり前。崇められたが、その後無残にも殺される…。

デブ鳥……古田の息子。突然変異か否か。飛べない鳥が生まれてしまった。最終的にはミサイルのように飛ぶことを覚えるが、無理がたたって全身骨折の末、再起不能。

バルサ1号……古田とカモメの親子の間の禁断の恋の間に生まれた。体がケント紙からバルサへと突然変異を起こし、強化若干太ったものの強度は抜群！！…に見えたが頭が胴体にめり込み死亡。

バルサ2号……頭がめり込んだだけで全身は無事だったバルサ1号が改造手術にかけられて復活。肉をそぎ落とされてスリムになり、羽の大改造によってある程度の飛行が可能に。しかし、体育館の屋根からのダイブでやられた尾が致命傷となり、幾度かの再改造を受けるも最後には尾は剥ぎ取られてしまった…。

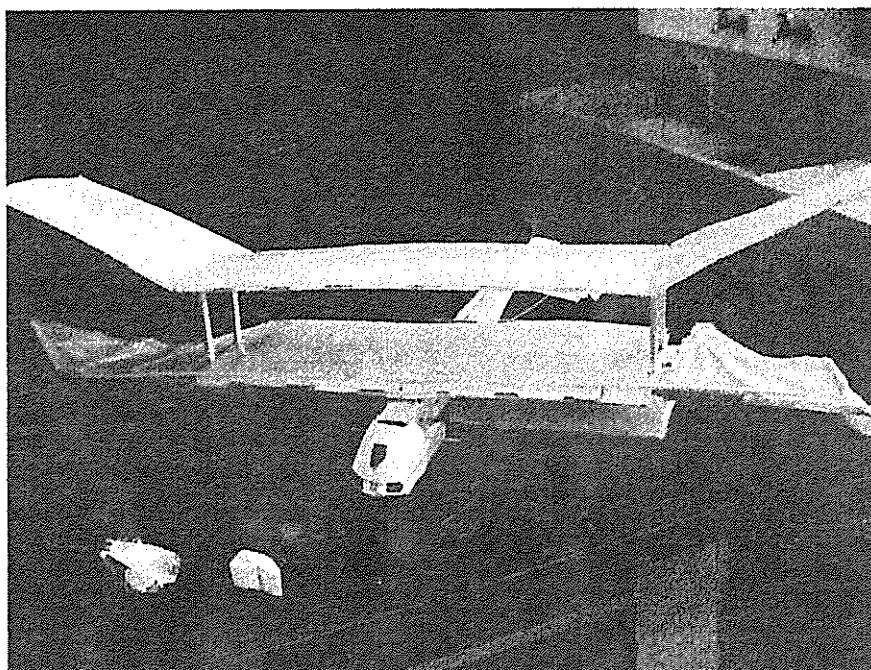
メーヴェ……バルサ2号の転生後の姿。地獄を見てきて大きく姿が変化した。まず、巨大な筋肉（モーター）が付けられて羽ばたけるようになり、尾羽の進化で進行方向も変えられるようになった。こうして、ついに地面と戦えるようになったのだ。メーヴェ家



の戦力は現在メーヴェである。しかし、近い将来また地面に殺害されるかもしれない。
その時は地面を倒せる鳥が再び生まれてくれるのだろうか…。

。 外観図

… とうり写真です。ちゃんと印刷できればいいのですが…



翼は全部で3段になっています。付けたいは、上から2段目の部分に行きます。

スペースに余ったので、デークを…

全長：約80 cm

横幅：170 cm

固定翼

翼面積：4600 cm² (46 dm²)

重さ、胴体：240 g (電池を含む)

その他：約600 g.

おそく、6~7 m/s で飛ぶはず…!!

Flap

羽ばたきの構造についてです。最初はいろいろと複雑なものを考えていたのですが、最終的(2005年3月17日現在)には、多少重くなるものの両方の羽それぞれにモーターをとりつけることで羽ばたかせています。一応図にすると下のような感じです。

一応解説します。まず、図2を見てください。これは図1で示されている軸受けの拡大

画像です。まずモーターは固定翼から伸ばした、アルミ

L字材を組み合わせたものに固定してあります。つまり

モーターは羽ばたく部分である可動翼の影響を受けない訳です。次に図2の上の方にあるL字材2つは可動

翼に固定されています。2枚の間には隙間が開いていて

そこにネジを通してスライドできるようにします。そのネ

ジを市販ギヤボックスに付

属していたプラスチック部品に取り付け、

そ反対側をモーターにつけることでモーターの回転がそのままネジのスライドに

伝わっていきます。そして、図2の状態

言えばこのままモーターが左回転するとプラスチック部品も左回転をして、ネジも

左回転をしようとします。このときネジは左上方向の力がかかっているため、ス

ライドしながらL字材が持ち上がります。

するとこのL字材が固定されている可動翼が上に上がっていきます。打ち下ろす

ときの動きも同じ原理です。

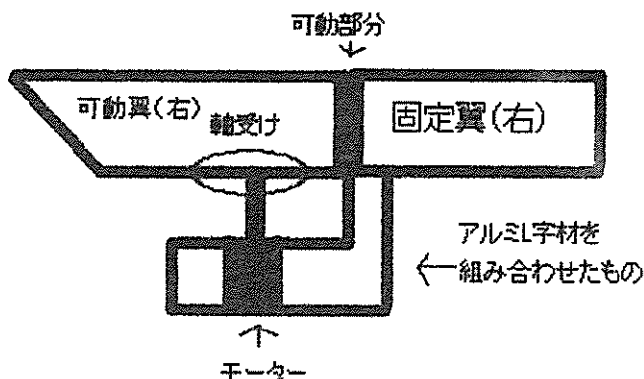


図1 上から見た図(右翼のみ)

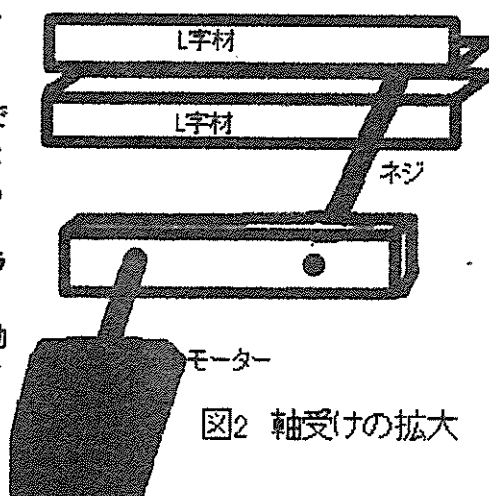


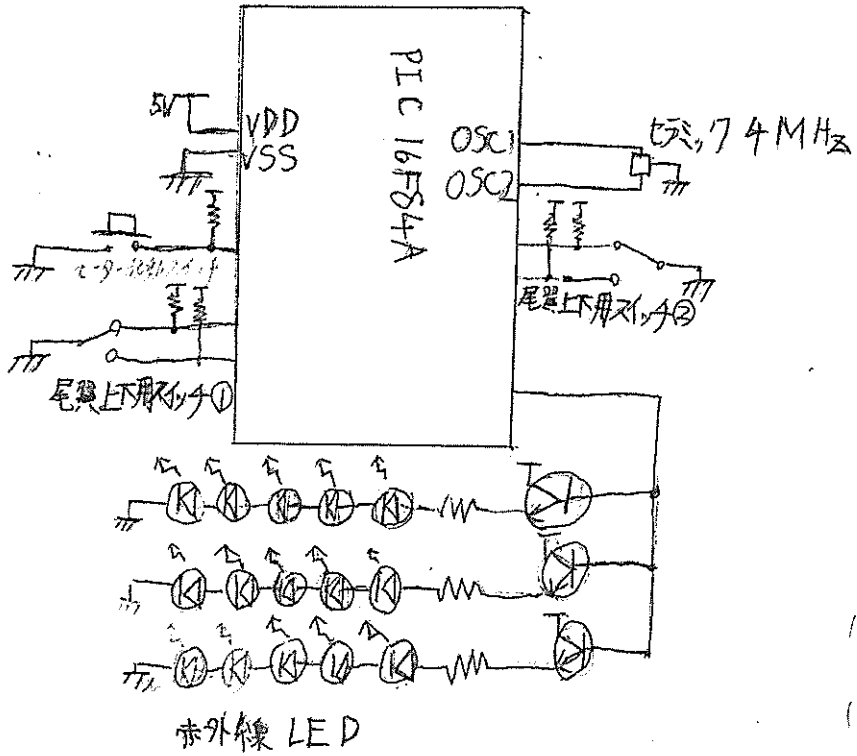
図2 軸受けの拡大

Controller

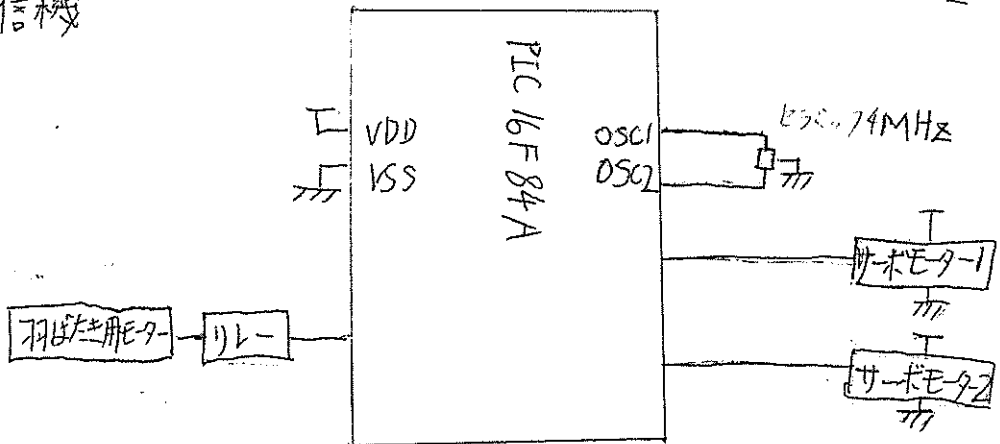
次にリモコンについてです。この『メーヴェ』は赤外線リモコンで動いています。赤外線というのは、目に見えないほど波長の短い光なのですが、これで信号を出し分けることで受信側は命令を受け取り、設定してある通りの動作をします。機械語は『0』と『1』だけで構成されている世界なので、この『0』と『1』の組み合わせで命令を発信します。簡単な例を挙げるならば、リモコンの送信部から0・0・1の順に赤外線を発信し、それを受信部で受け取ります。受け取る際、0と1は反転するので受信部には1・1・0と認識されます。この時に、受信部では1・1・0に定義されている命令を実行していきます。まあ、例というか仕組みみたいになってしまいましたが、基本はこんな感じです。全てを正確に書いていくとスペースを結構とってしまうのでこのぐらいで終わります。次ページより赤外線リモコンのプログラムと回路図を挙げていきます。

～回路図～

リモコン



受信機



・プログラム

```

*****送信プログラム*****
LIST      P=PIC16F84A
INCLUDE P16F84A.INC
_CONFIG_XT_OSC      &
_WDT_OFF & _PWRTE_ON &
_CP_OFF

INDATA EQU      0DH

SETUP
    BSF
    STATUS,RP0
    CLRF      TRISA
    CLRF      TRISB
    MOVLW     1FH
    MOVWF     TRISA
    BCF
    STATUS,RP0
    CLRF      PORTB
    CLRF      PORTA

;*****
**
;送信プログラム
;*****
**
MAIN
    CALL      WRY
    CALL      WRY
    CALL      WRY
    CALL      WRY
    CALL      WRY
    BTFSS
    PORTA,0
    GOTO      MAIN3
    CALL      WRY
    GOTO      MAIN4

MAIN3
    CALL      WRY

MAIN4
    BTFSC
    PORTA,1
    GOTO      MAIN5
    CALL      WRY
    GOTO      MAIN6

MAIN5
    CALL      WRY

MAIN6
    BTFSC
    PORTA,2
    GOTO      MAIN7
    CALL      WRY
    GOTO      MAIN8

MAIN7
    CALL      WRY
    BTFSC
    PORTA,3
    GOTO      MAIN9
    CALL      WRY
    GOTO      MAIN10

MAIN9
    CALL      WRY
    BTFSC
    PORTA,4
    GOTO      MAIN11
    CALL      WRY
    GOTO      MAIN12

MAIN11
    CALL      WRY

MAIN12
    CALL      WRY
    GOTO      MAIN

WRY
    MOVLW     017H
    MOVWF     0CH
    NOP

HL
    BCF
    PORTB,1
    BCF
    PORTB,0
    NOP
    NOP
    DECFSZ    0CH
    GOTO      LL
    RETURN

END

*****受信プログラム*****
LIST      P=PIC16F84A
INCLUDE P16F84A.INC
_CONFIG_XT_OSC      &
_WDT_OFF & _PWRTE_ON &
_CP_OFF

;////////////////////
;レジスタ設定
;////////////////////
COUNT1 EQU      0CH
COUNT2 EQU      0DH
INPUTDA EQU      0EH
GPR_4 EQU      010H

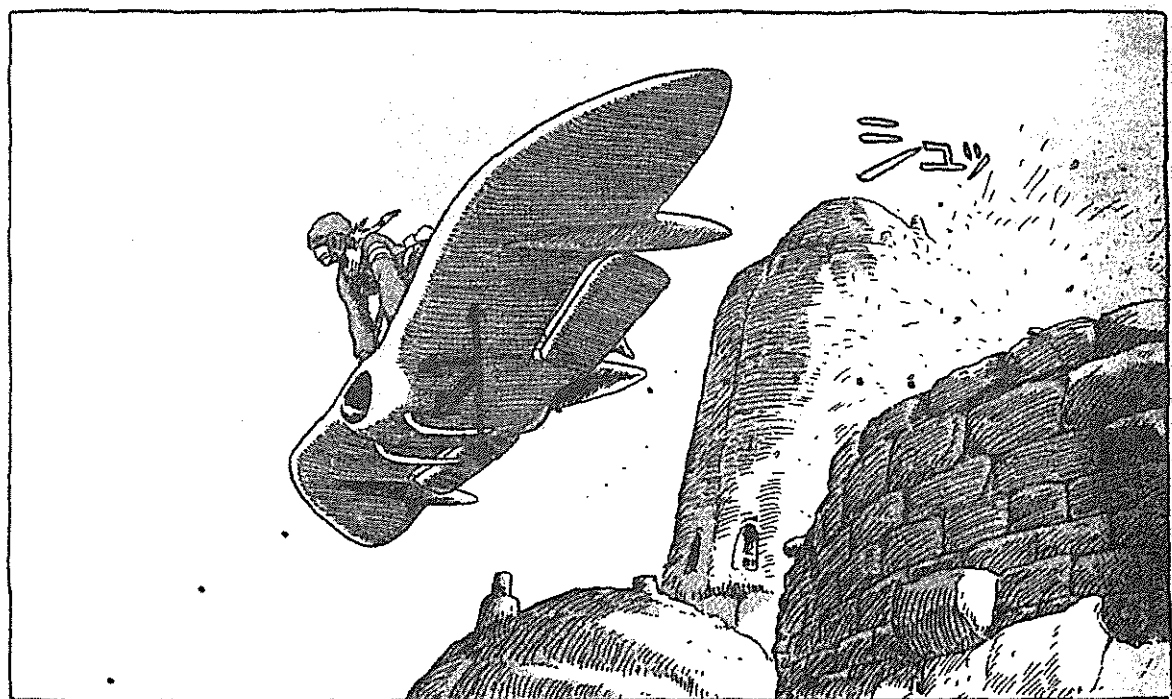
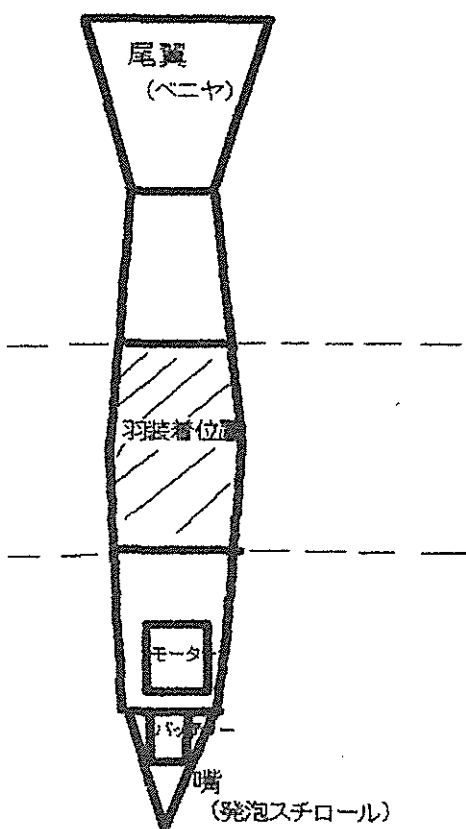
```


INPUTDA,4	MAININ6_TWO	GOTO	MAIN1
GOTO	MAININ5_TWO	MAININ14_TWO	
MAININ15	BSF	BSF	
MAININ14BSF	INPUTDA,1	INPUTDA,4	
INPUTDA,4	GOTO	GOTO	MAIN1
GOTO	MAININ6_TWO		
MAININ15	MAININ6_TWO	;;;;;;;;;;;;;;;;;	
CALL	CALL	:分岐プログラム	
WAIT600	WAIT600	;;;;;;;;;;;;;;;;;	
BTFSC	BTFSC	MAIN1	
PORTA,0	PORTA,1	BTFSC	
GOTO	GOTO	INPUTDA,5	
MAININ16	MAININ7_TWO	GOTO	
GOTO	GOTO	MOTORL	
MAININ17	MAININ8_TWO	GOTO	
MAININ16BCF	MAININ7_TWO	MOTORH	
INPUTDA,5	BCF	MAIN2	
GOTO	INPUTDA,2	BTFSC	
MAIN1	GOTO	INPUTDA,1	
MAININ17BSF	MAININ9_TWO	GOTO	
INPUTDA,5	MAININ8_TWO	MAIN0A	
GOTO	BSF	BTFSC	
MAIN1	INPUTDA,2	INPUTDA,2	
	GOTO	GOTO	
	MAININ9_TWO	MAIN180A	
	MAININ9_TWO	GOTO	
	CALL	MAIN90A	
	WAIT600	MAIN3	
	BTFSC	BTFSC	
	PORTA,1	INPUTDA,3	
	GOTO	GOTO	
	MAININ10_TWO	MAIN0B	
	GOTO	BTFSC	
	MAININ11_TWO	INPUTDA,4	
	MAININ10_TWO	GOTO	
	BCF	MAIN180B	
	INPUTDA,3	GOTO	
	GOTO	MAIN90B	
	MAININ12_TWO		
	MAININ11_TWO	;;;;;;;;;;;;;;;;;	
	BSF	:左のサーボ	
	INPUTDA,3	;;;;;;;;;;;;;;;;;	
	GOTO	MAIN0A	
	MAININ12_TWO	BSF	
	MAININ12_TWO	PORTB,1:0°	
	CALL	CALL	
	WAIT600	WAIT100	
	BTFSC	CALL	
	PORTA,1	WAIT100	
	GOTO	CALL	
	MAININ13_TWO	WAIT100	
	GOTO	CALL	
	MAININ14_TWO	WAIT100	
	MAININ13_TWO	BCF	
	BCF	PORTB,1	
	INPUTDA,4	CALL	

WAIT100		WAIT100		PORTB.2
CALL		CALL		CALL
WAIT100		WAIT100		WAIT100
CALL		CALL		CALL
WAIT100		WAIT100		WAIT100
CALL		BCF		CALL
WAIT100		PORTB.1		WAIT100
CALL		GOTO MAIN3		GOTO MAIN
WAIT100			MAIN180B	
CALL		:/:/:/:/:/:/:/:/:/		BSF
WAIT100		:右のサーボ		PORTB.2
GOTO MAIN3		:/:/:/:/:/:/:/:/		CALL
MAIN90A		MAIN0B		WAIT100
BSF		BSF		CALL
PORTB.1		PORTB.2:0°		WAIT100
CALL		CALL		CALL
WAIT100		WAIT100		WAIT100
CALL		CALL		CALL
WAIT100		WAIT100		WAIT100
CALL		CALL		CALL
WAIT100		WAIT100		WAIT100
CALL		CALL		CALL
WAIT100		WAIT100		WAIT100
CALL		BCF		CALL
WAIT100		PORTB.2		WAIT100
CALL		CALL		CALL
WAIT100		WAIT100		WAIT100
CALL		CALL		CALL
WAIT100		WAIT100		WAIT100
BCF		CALL		CALL
PORTB.1		WAIT100		WAIT100
CALL		CALL		BCF
WAIT100		WAIT100		PORTB.2
CALL		CALL		GOTO MAIN
WAIT100		WAIT100		
CALL		CALL		:/:/:/:/:/:/:/:/
WAIT100		WAIT100		:羽ばたきのモーター
GOTO MAIN3		GOTO MAIN		:/:/:/:/:/:/:/:/
MAIN180A		MAIN90B		MOTORH
BSF		BSF		MOVLW 0F0H
PORTB.1		PORTB.2		IORWF PORTB
CALL		CALL		GOTO MAIN2
WAIT100		WAIT100		MOTORL
CALL		CALL		CLRF PORTB
WAIT100		WAIT100		GOTO MAIN2
CALL		CALL		
WAIT100		WAIT100		:/:/:/:/:/:/:/:/
CALL		CALL		:ウェイト
WAIT100		WAIT100		:/:/:/:/:/:/:/:/
CALL		CALL		WAIT100
WAIT100		WAIT100		
CALL		CALL		MOVLW 042H
WAIT100		WAIT100		MOVWF 0CH
CALL		CALL		WAIT101
WAIT100		CALL		DECFSZ 0CH
CALL		WAIT100		GOTO
WAIT100		BCF		WAIT101
CALL				

～本体構造～

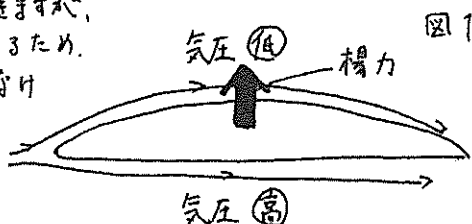
本体の大部分はパルサをエポキシ接着剤で固めて作ります。尾翼はベニヤで作し、これで旋回するために本体内部のサーボモーターからワイヤーを伸ばしてつないであります。また、バッテリーは嘴の中に搭載しており、モーターは本体内部に固定して設置してあります。さらに、モーターやサーボモーターを統括する回路は本体外部に露出していますが、これは嘴のつき方や尾翼の大きさから考えて、どんな落ち方をしてしても壊れないからです。



。飛行機が飛ぶ原理

1. 揚力

飛行機が飛んでいる際に、上に浮かぶ力、つまり飛ぶのに一番重要な力です。飛行機の主翼は図1のような断面をだいたいしています。飛んでいる時は前から風が当たりますが、その際に翼の上面は湾曲しているため、下面を通るよりも、空気は速く通るだけはいけません。そのため、上面は下面より気圧が下がり揚力が生じます。



揚力の大きさは次の式で表されます。

$$L = \frac{1}{16} C_z S V^2$$

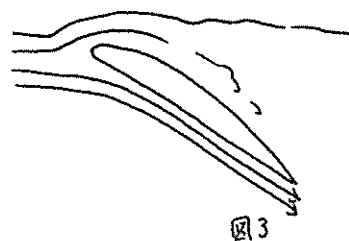
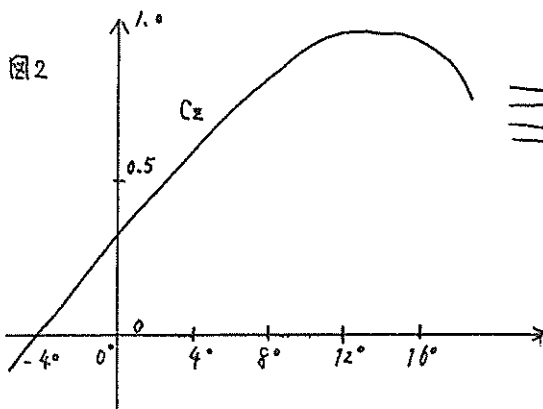
L : 揚力 (kg)

C_z : 揚力係数

S : 翼の面積 (cm²)

V : 速度 (m/s)

揚力係数とは、翼の形によって決まっている値のこと。図2のようなグラフで表されています。この値は風が当たる角度で変化する、10°くらいで最高になります。これ以上だと、図3のように、空気がうまく流れないため、失速



してしまいます。

この式から言えることは、揚力は、翼面積に比例し、速度の自乗に比例することです。ただ、後に述べる、つり合いが飛行機では重要なので、ただ単に翼を大きくし、速く飛ばせばいいというものではありません。

2. 抗力

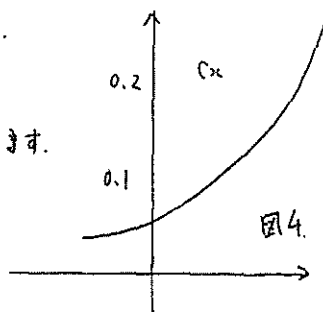
飛行機が空中を飛んでいると翼には揚力のほかに、進むのを妨げる力、抗力が生じます。空気との摩擦に因る、「摩擦抗力」、圧力の分布具合によって生じる「圧力抗力」、そして翼端の空気の移動によって生じる「誘導抗力」、これらの3つは、揚力を生むために翼が犠牲を払っているものなので、ある意味仕方ありません。他に「寄生抗力」などもあります。

翼の抗力も揚力と全く同じような式で計算できます。

$$D = \frac{1}{16} C_x S V^2 \quad C_x = \text{抗力係数}$$

抗力係数のグラフは、だいたい図4のようになっています。

抗力は小さいほうが良いものです。



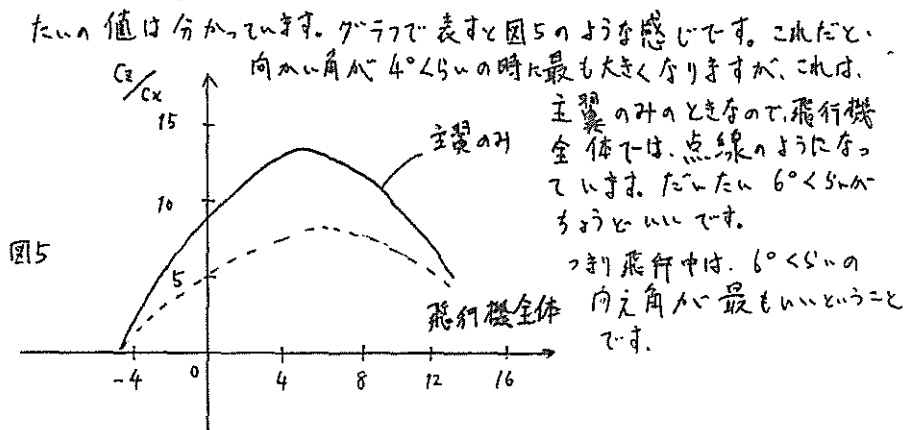
3. 揚抗比

揚抗比とは $\frac{\text{全体の揚力}}{\text{全体の抗力}}$ です。

つまり、揚力が抗力の何倍になるかということです。抗力は小さいほうが良いので、揚抗比は大きければ大きい方がよいです。

前に挙げた式で考えると、だいたいの揚抗比は、

$$\frac{L}{D} = \frac{\frac{1}{16} C_z S V^2}{\frac{1}{16} C_x S V^2} = \frac{C_z}{C_x} \quad \text{となり、つまり、翼型によってだいたいの値は分かっています。}$$



4. 飛行時のつり合い

水平飛行中は、重力と揚力が等しいはずなので、次のような式が成り立ちます。

$$W = L = \frac{1}{16} C_z S V^2$$

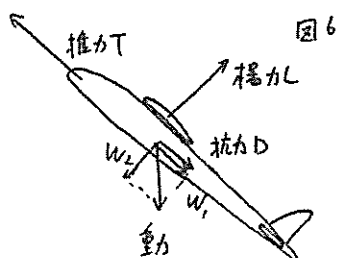
これにより飛行機の重さ、翼面積から決まっています。向い角が決まっていますと、速度から出ます。揚力係数はだいたい $0.6 \sim 0.7$ くらいなので

$$V = \sqrt{\frac{W}{S}} \quad k: 1.5 \sim 1.6 \text{ くらいの値}$$

上昇飛行中の場合は、推力が抗力に勝っているため、揚力が大きくなっています。この場合は、重力の進行方向と平行な向きの成分が抗力に合ったり、つり

合いが保たれます。(図6)

滑空飛行の場合は、これとは反対に、重力の成分の一部が、推力に合ったり、つり合いが保たれます。



これらのことから、揚力の所で書いた通り、速度と翼面積を大きくすれば、よく飛ぶようになるわけではなく、ただ、急な上昇をするだけになってしまいます。

6. 尾翼について.

飛行機は、重心を中心とした、2つの働きで姿勢を保っています。簡単に尾翼の面積や、主翼の位置について書きます。

まず、主翼の位置についてですが、主翼は、重心からちょっと後ろに、真ん中がくるのがいいです。ゆっくり飛ばしたい時は、重心の真上、速く飛ばしたい時は、それよりも後ろの所にすればいいです。

水平尾翼の面積は、

$$S_H = k \frac{S \times t}{l}$$

S_H : 水平尾翼の面積

k : $0.6 \sim 1.0$ くらいの尾翼容積に相当する値

t : 主翼の縦付け

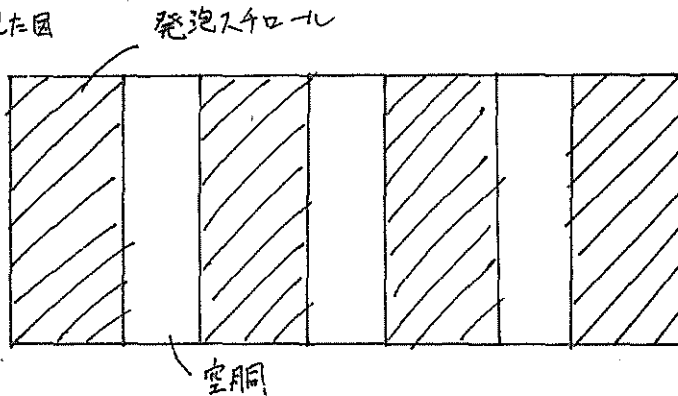
垂直尾翼は、方向の安定性を保つためにありますが、大きすぎると横すべりした時に機首を下げて螺旋降下をしてしまい、小さいと、方向が定まらず、きりぎりす状態になってしまい、調整をしながら、決める必要があります。

羽の仕様

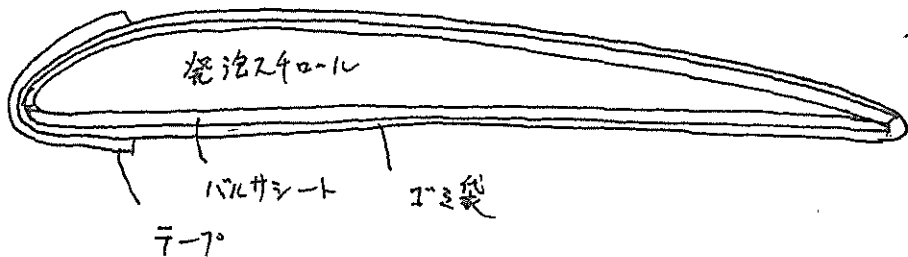
4月13日現在、主翼は3段になっています。色々な翼の製作法を試してみましたが、今の翼が一番軽く、良いです。

今の翼は、3つとも、発泡スチロールの芯と骨子部分が、3~5cmくらいの間隔を空けてあり、その表面を0.5mmの厚さのバルスシートでおおい、さらにその上に、0.02mm厚のゴミ袋をはめています。接着は、ぼんごをスプレーでしています。そして前縁部にはテープをはっています。胴体とは竹ひごを使ってゴム止めをしています。

- 上から見た図

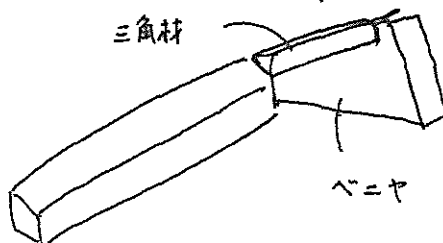


- 断面図



尾翼は、ベニヤ板を適度な大きさに切り、可動部をテープで止めています。サボとの接合などは、胴体のところを見て下さい。

胴体との接合は、三角材を削り削って、それを間に入れてつけています。



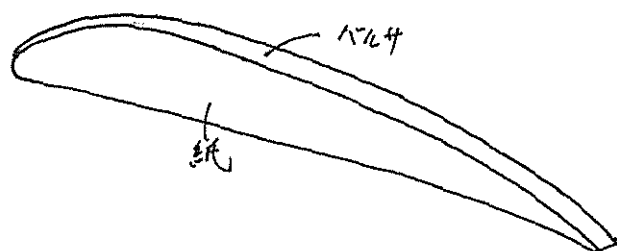
主翼の製作方法

きろんとした揚力を得られ、かつ抗力が少ない翼を作るには少し手間がかかります。ここでは手順を追ってそれを説明いたします。

① 翼の型を作る

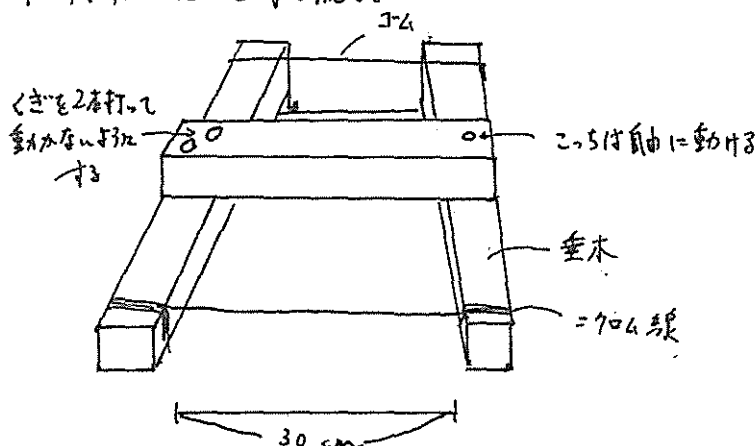
② で発泡スチロールを切り出す時の型です。

5mm ほどのバalsaに、翼型(ネットなどにある)を切った紙をはりつけ、それに合わせて、バalsaを切ります。下のようになります。



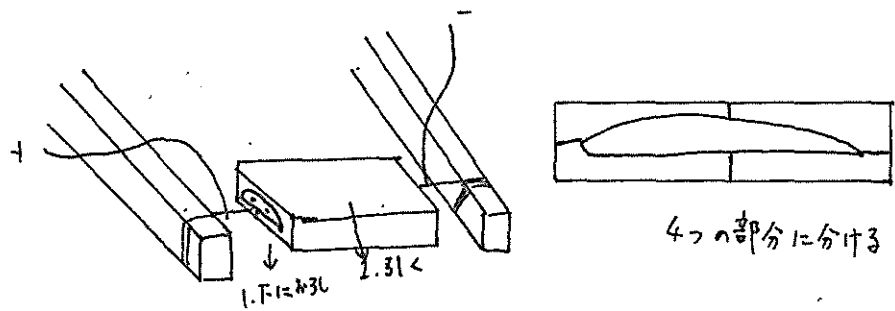
② 発泡スチロールカッターの製作

② で発泡スチロールを切るために使うものです。下の図のようになります。これにヒョウを流して使います。



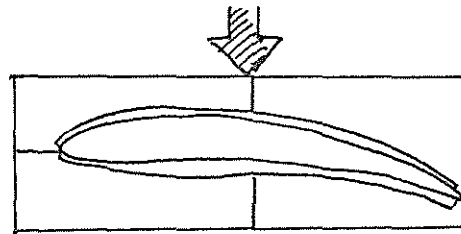
③ 発泡スチロールの切り方

まず適度な厚さ、長さの発泡スチロールの側面に①で作った型をまち針で止めます。そして②で作ったスチロールカッターで切ります。その時は4回に分けて、そと引きながら切るときれいにいきます。何回かやっていけばそのうちできるような気がします。



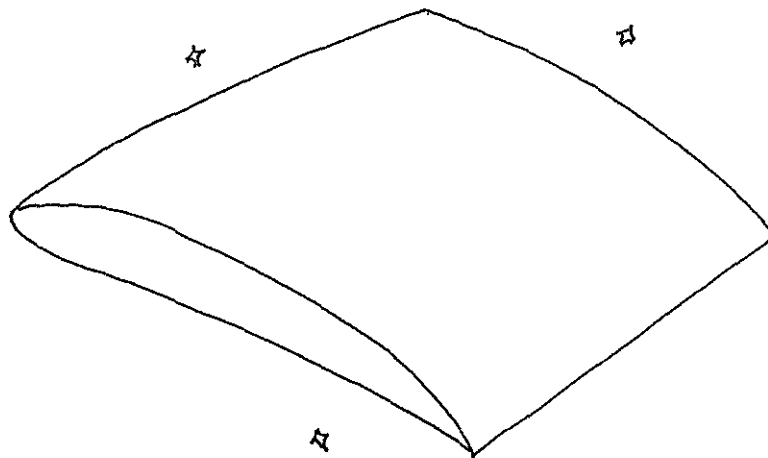
④ バルサバリ

設計図に合わせて、発泡スチロールを間をあけて置きます。それにスプレーを吹いて、0.5mmのバルサをはっていきます。はた後は⑤のスチロールの切り取った残りを合わせて、重しを置き、圧着させます。



⑤ ビニールはり

④で1時間ほど置いて、再びバルサの上にスプレーを吹きます。その上に、あらかじめ切っておいたビニールをはります。僕は学校で使っている、ゴミ袋を使いました。もし、補強したい所に梱包用テープをはって、完成です。

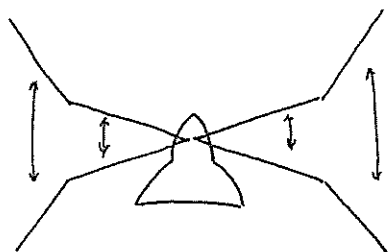


のはばたきの仕組み

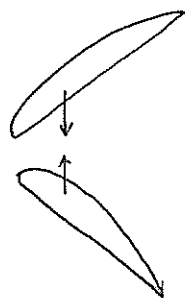
この飛行機は鳥型ですので、プロペラではなく、はばたきによって推力を得ます。

詳しくは、歴史の所を見てもうえば、わかると思うのですが、初めは普通の羽が途中で少ししなるようになっていて、それを振ろうとしていました。が、明らかに推力が足りず、今のような翼になりました。

元々のはばたきて、推力を得るには、「しなり」が重要です。図7のような感じに先端はより上下に動き、翼の断面は、下に打た落すときは向かい角を下げ、打た上げるときは、角度を上げることでより推力を得ます。

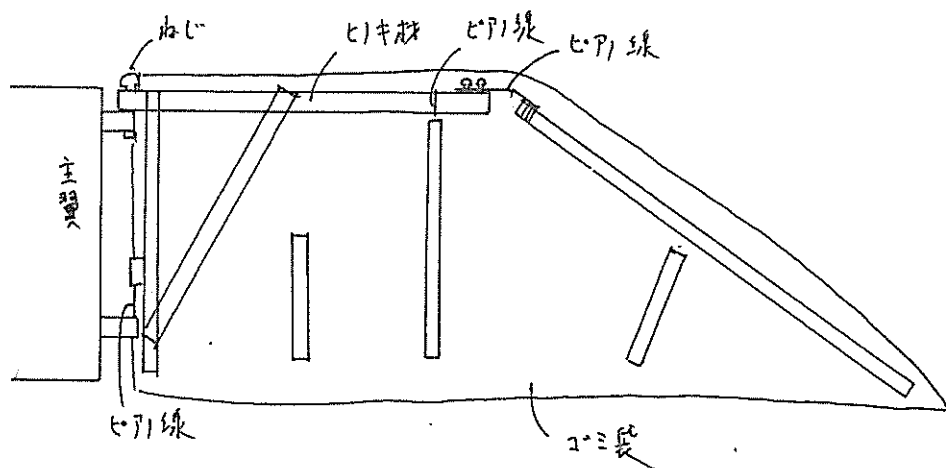


この2つを同時に行えると、翼端にいくにつれて根元にあった動きが遅れ、伝わり、本物の鳥のようにはばたくことができます。



今の翼は、骨組をヒコリ線で作って、ねじ止めを4つして、上下方向に動かすようにしています。また、一部の骨は、ビニールのシートに固定されているような形になっていて、うまく具合になわりか生まれます。

図7



。感想

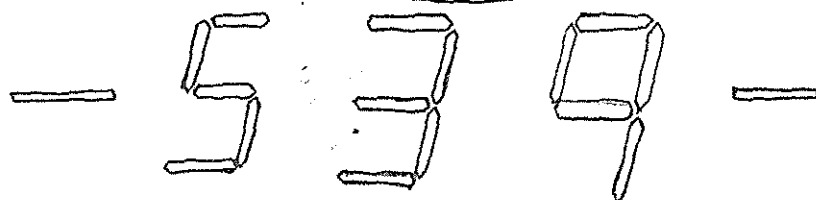
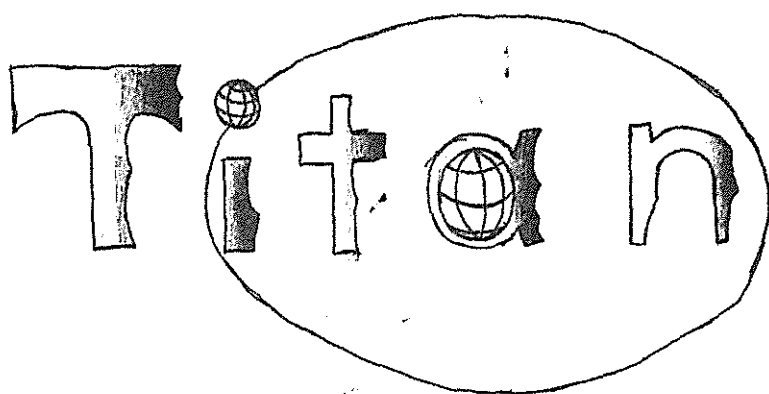
森…去年の冬休み、「風の谷のナウシカ」の原作のまんがを読んで、「マーウエ」にあこがれ、今年の製作物は決まりました。最初は紙飛行機作りから始めたのですが、自分で設計したものが、ゴールの午前まで飛んだときは、「絶対に完動する!!」と確信しました。現在、モーターをつけてはばたかせていると3分、もうすぐ飛ばせそうです。翼の軽量化の成功は大きかったです。

下の感想や、後書きでは、井上君はあんまり何もしないと言っていますが、実際のところ、明確な役割分担ができなかったこともあり、実はほとんど俺と井上で作りました。リモコンも、井上の調整しただけです。また修理などは、ほとんど井上にまかせてしまいました。本当に助かったよ。ありがとう。

今回かなり未知のジャンルとして「飛行機」を選んだわけですが、やっぱり飛行機の研究に時間を使いすぎ、無線班的なことや、ちとむききせんでした、そこが残念です。今までやってきた人がいないため、資料として使えるのに、この図路図集には、「飛行機の基礎知識」を喜きました。これを読んで、これからは飛行機を作りたいと思う人が増えると思うんです。小型、軽量機（試作機は、4階からゴールのネットまで飛んだ）あたりがいいと思います。

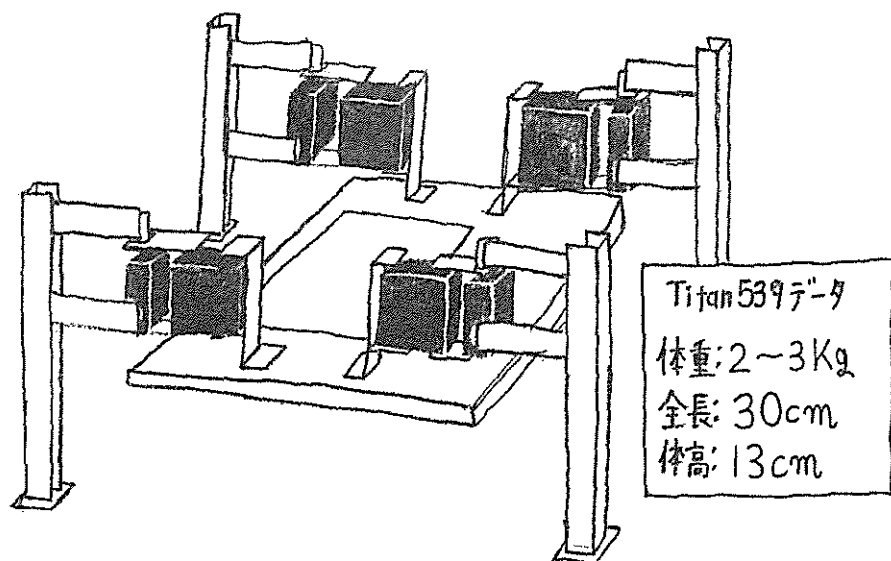
最後になりましたが、この見にくい文章を読んでいただき、ありがとうございます。みなさんのおかげで、いい製作物を最後に作れて、うれしいです。本当にありがとうございました。

井上…去年は中一がやる簡単なゲームを作り、今年は突然羽ばたいて空を飛ぶ鳥型ロボットにまで飛躍しました。今現在はやっと羽ばたきの機構ができたところなのでまだどうなるかは分かりませんが、正直できているものはスゴイと思います。製作物のアイデアがまだ決まっていなかったときに森さんにこれに誘われて共同製作となりました。できるだけたくさん部活に出るようにしてがんばって進めていこうとしました。しかし、僕に技術が無いのでほとんど森さんに頼ってしまってあまり製作のコアになる部分には触れてない気がします。強いて言えば羽ばたき部分のモーター固定部分の構造ぐらいでしょう（これも文化祭までには変わっているかもしれないが…）。ロボット製作は去年のゲームとはやはり全然難易度が違っていました。一番大きいところに本体を作らないといけないうところなんです。ゲームなら基盤にICを乗せればいいだけのところをその『基盤』になるところから作っていかなければいけません。これが大変でした。特に今回の製作はしょっちゅう壊れるため、修理も大変でした。今回のロボット製作は僕にとってはいいい経験になりました。僕が足を引っ張りながらも半年間一緒にやってくださった森さん、ありがとうございました。



— 外觀図 —

製作: H2 舟渡
 H1 伊藤
 M3 安部
 協力: 物理部の方々
 OB様



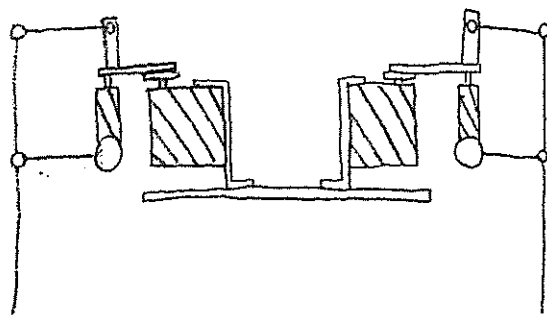
Titan 539 データ
 体重: 2~3Kg
 全長: 30cm
 体高: 13cm

— 略図 —

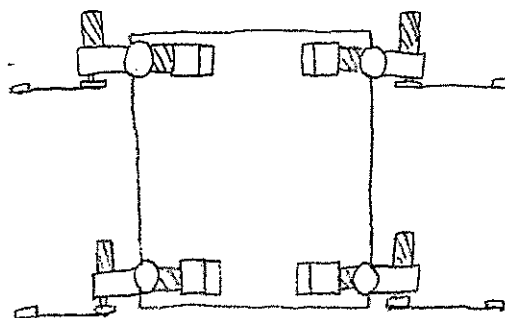
○・○…可動部

▨…サ-ボ-イ-タ-

正面



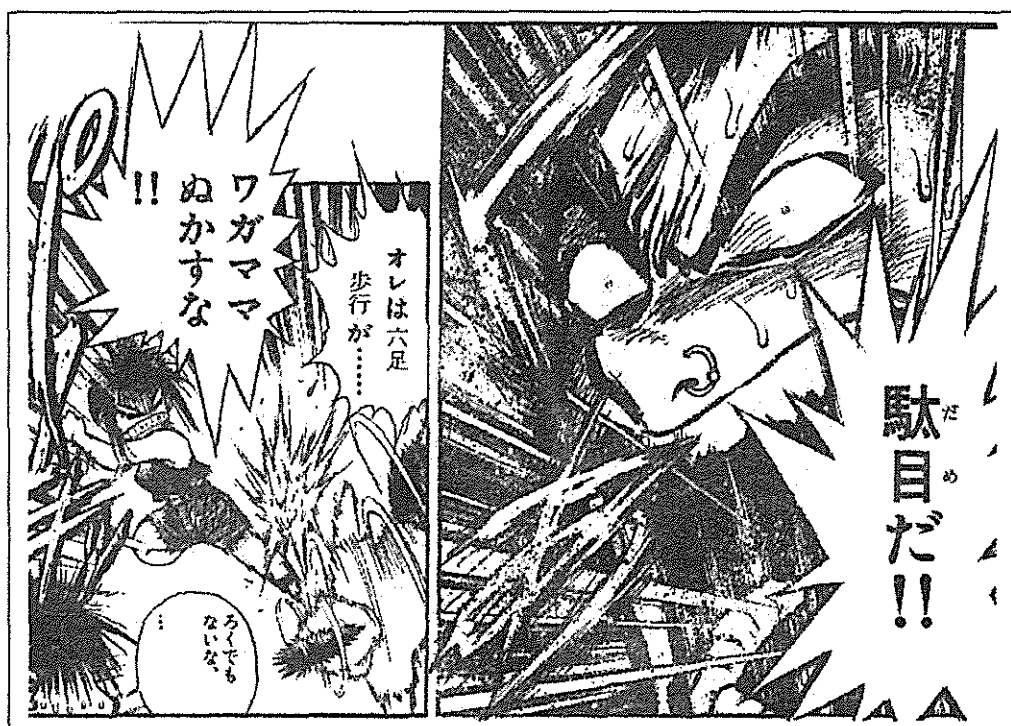
上面



一、Titan539 とは

Titan539 とはわが物理部のロボット技術の英知の結晶かもしれない4足歩行ロボットです。モデルは某T大のタイタンでそれを目標にして作りました。少ない関節数（足一つ辺り2つ）で4足歩行ができるかどうかと、サーボモーターと言う新たな技術がこの部で使えるかどうかを実験し、それらの技術を確認するためにこの製作に踏み切りました。それで、簡単に結果を言わせてもらえばとりあえず現段階（2005/04/20 現在）いくつかの歩行パターンで歩いています。4足で安定し、プログラムも基礎ができていますので応用次第では様々な歩行パターンのシミュレーションも可能です。あとは今本体には最低限の回路と赤外線受光モジュールとその周辺回路しか乗せていないので汎用性もありそうです。まあ、暇なら上になんか乗つけてもいいんじゃないですかね。

基本的な構成ですが、本体には主にアルミ板を制御には PIC16F877 を、操作には赤外線を使っています。アルミはいいです、軽くてそこそこ丈夫ですし、薄くても立体構造とらせれば強度も問題ありません。これからロボを作る人もアルミを使いましょう。



いろんな事情で六足は今回作れません。

小さい四足が作れたんです。

二、サーボモーターって何？

さて、聞きなれない単語かもしれないけれどこれについて説明します。

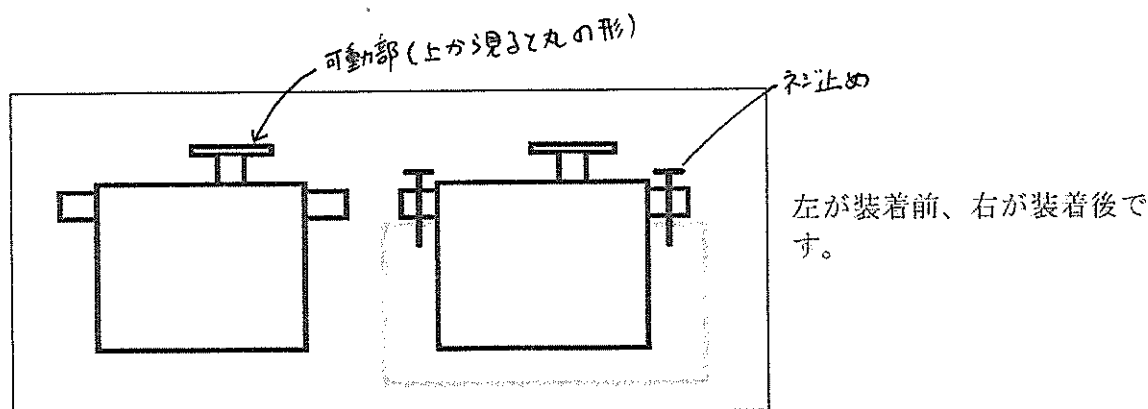
サーボモーターとは駆動速度が速く、トルクが強く、制御角度正確なすばらしいモーターです！！ただ少々値は張りますが・・・それでも1つ1600円ほどなのでギヤボックス買ったりギヤを買うよりは安上がりかもしれません。逆に駄目なところは、トルクは大きいのですが歩行ロボットの足などに使った場合内部のギヤがプラスチックのためトルクと同じ重さがかかるとギヤが壊れてしまうため、実際の力（本体を支える）はあまり期待できないということです。詳しい制御法はまたあとで書きます。

三、構造

このロボットの足にある2つの関節の構造について説明します。

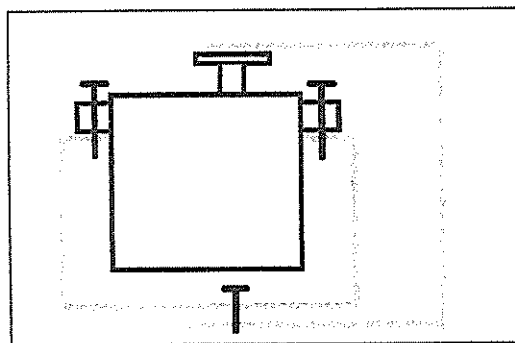
I、サーボブラケット

今回サーボモーターを使おうと思ったけど次のみたいな形なので使いにくい！！と思っただのでサーボブラケットを作りました。これはサーボモーターを使う上では必須で結構有名です。もしサーボを使うなら作りましょう。ちなみに材料は厚さ1mm、幅2cmのアルミ板（ハンズで購入）を使いました。サーボとちょうど横幅が合い、厚さも薄く軽かったのととても重宝しました、まげてサーボに付ければ耐久性も十分です。今「まげて」と言ったようにサーボブラケットの製作には主に折り曲げを使いました。やり方は折り曲げたいところを万力で固定して木槌で叩くだけで簡単でネジいらずで便利です！！なんかこれ以来物無で木槌の音を聞くことが多くなりましたが……やはり小さな物を加工するにはお勧めです。



II、関節1（足の付け根部分）

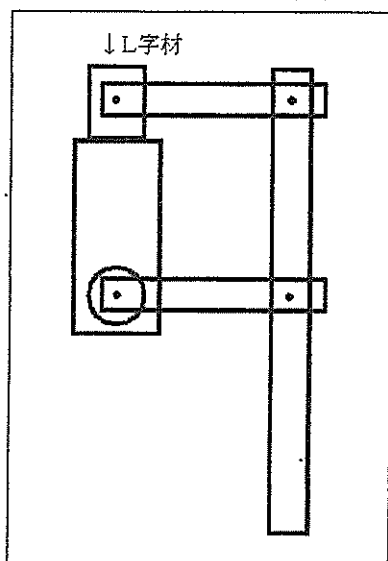
さて、前言ったようにサーボブラケットを付けたわけですが、これによりいろいろと物が（に）固定しやすくなったわけです。そこで付け根部分の関節ですが、下図のように



コの字型に折り曲げたアルミ板を取り付けます。で、その部分にパーツを付けることで、上から見たときの前後の動きができるようになるわけです。

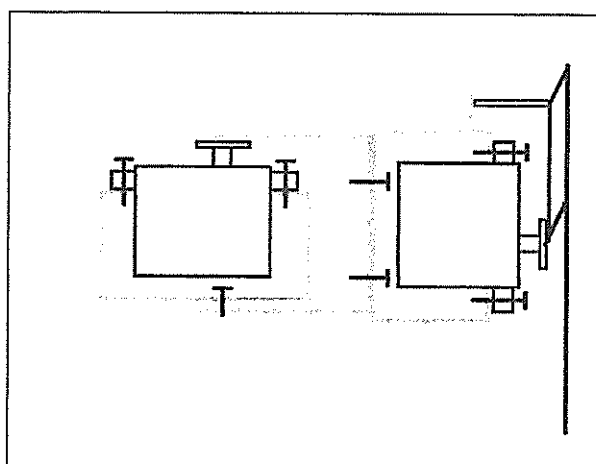
このとき下でブラケットとコの字型を固定している部分は上の可動部の中心と同じ位置にしましょう。まあびったり同じは難しいのですが・・・。

Ⅲ、関節 2 (足の先部分)



今度は左図のようなものになります。これは絵の●のところでネジ止めしてあり、可動部が回転するとそれに合わせてその右の長いアルミ板が上下に動くという平行リンクと呼ばれる方式を使っています。これは横向きの二本の同じ長さのアルミ板によって縦の二本が常に平行になるというものです。この場合左側が固定されているので右側が上下に平行に動きます。これにより足の上下運動が常に地面と垂直になるので接地する瞬間安定します。

上のL字材はサーボブラケットに固定してあります。



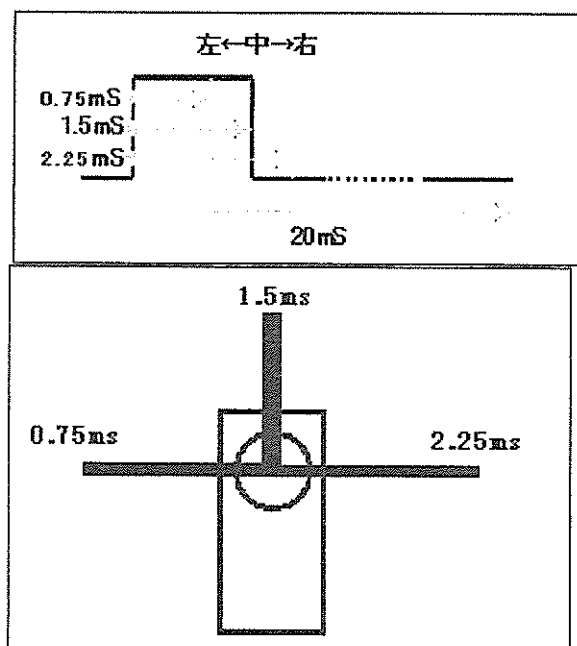
両方の関節を組み立てると左の図のようになります。本体にはこれを4つ付けています。



四、ソフトウェア

1、サーボモーターの制御法

サーボモーターには端子が三つ出ています。これは一つがGND、一つがモーター用電源、一つが制御用端子です。サーボモーターの角度はこの制御用端子に入れる信号によって制御することができます。



左の図は制御用端子に入れる信号をあらわしたものです。図のようにHの信号を1m秒～2m秒入れていて、その長さで角度が決まります。そして全体の一周期はそのHの長さによらず固定です。図では20m秒となっていますが、この周期は別にどうでもよく、単に20m秒のとき最もサーボモーターの出力が高くなるだけです。(まあ、高いほうがいいので普通は20m秒にしますが)

サーボモーターの対応は下図のように0.75m s のとき左に90度、1.5m s のとき真ん中、2.25m s のとき右に90度となります。

そしてこの制御をどうするかですが、今回はこの信号の時間を計るのにタイマ0割り込みというのを使いました。

このタイマ0割り込みとはPICに備わっている昨日の一つで、実行されているプログラムとは関係なく一定時間が立つと割り込みフラグが立ち、割り込み関数（ようは割り込んで実行するプログラム群のこと）を実行し、元の割り込む前のプログラムに戻るというものです。これで一定時間おきに割り込ませて、その割り込んだ回数で時間を計り、なおかつ割り込む回数を指定することで角度の制御が容易にできるようにしたものが今回のプログラムです。例えばタイマ0割り込みの時間を25μ秒としたときは30回割り込むよう指定すれば左に90度の角度まで動きます。

ですがこれらの設定には一定の時間がかかるため、角度制御信号のうち毎回必ず0.75m秒はHを出していることに注目し、その時間を利用して割り込む回数の指定やその他の設定を行うようにしました。他には時間が不定であるH信号の部分を最大の2.25m秒を一区切りとし、指定された時間Hを、2.25－その時間だけLを、そして20ms－2.25ms＝17.75m秒だけLを流すようにします。文字ばかりだと分かりにくいと思うので次に大まかな流れを載せておきます。

①出力をHに変更

750 μ 秒のタイマ0を設定、割り込みが入るまでに
角度を指定しあとは割り込みが入るまで待つ

↓↓

②25 μ 秒のタイマ0を設定、前に設定した回数割り込むまで
H信号を維持し、一定回数に達したらL信号にする。

↓↓

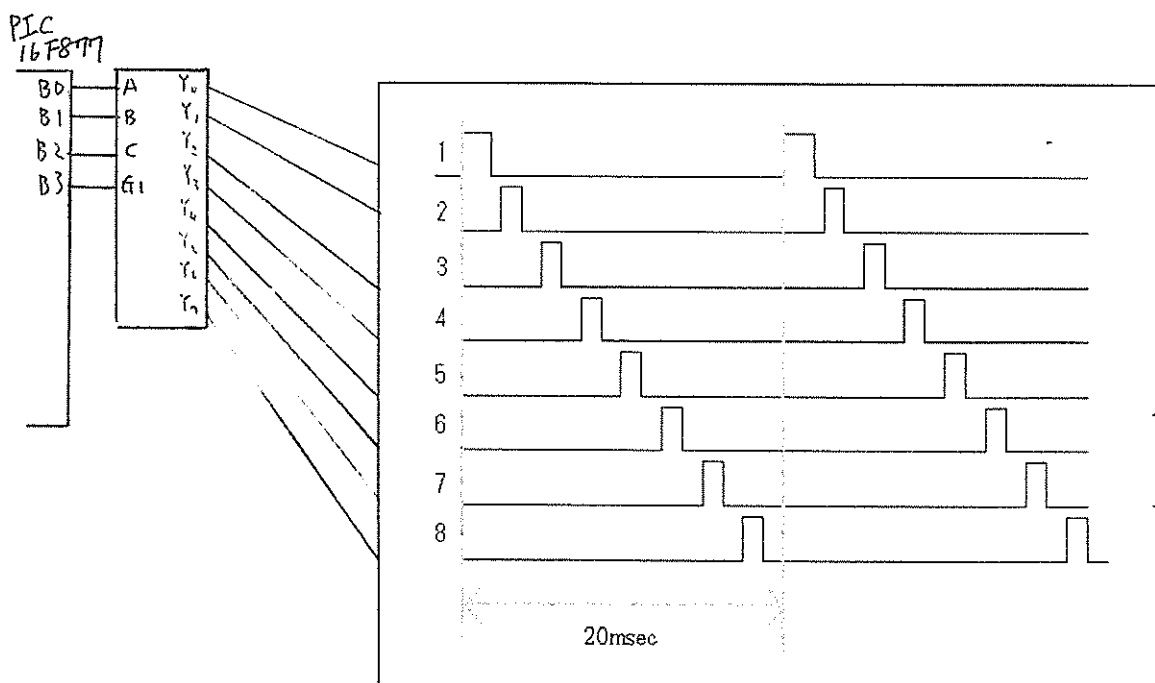
③Hにしてから2250 μ 秒、つまり25 μ 秒のタイマ0になってから
1500 μ 秒（割り込みが60回分）L信号を維持

↓↓

④長すぎるため半分にし8875 μ 秒のタイマ0を設定
2回割り込みが入るまでL信号を維持、
この時間を利用して様々な処理をすることができる。
今回は液晶を光らせている。

これが主な流れです。

そして次に、サーボモーターの複数同時制御ですが
これは③までは一つの場合と同じですが、④の長い時間を利用して今回は8個のサーボ
モーターを制御しています。このときH信号を出すのは同時間に1つのため74LS
138を使い3bitを8bitに変換、74LS540を通して信号を反転させた回路を使っていま
す。この時の信号と回路図は次のようになります。



これはどうなっているかという、①から③まで実行したあとHにする番号を変えてまた①から③の実行を8回繰り返して、その上で残った

$$20000 \mu s - 2250 \mu s \times 8 = 2000 \mu s$$

をタイマ0で待ち、次の周期に移ります。これがサーボモーターの複数同時制御法です。

II、プログラム

さすがにプログラムを全て載せるのは無理なのでここにまた概要を書きます。

ここまでの説明でサーボモーターの動かし方は分かったかと思いますが、まだこれをこのまま使うわけにはいきません。サーボモーターの角度制御についての記述に

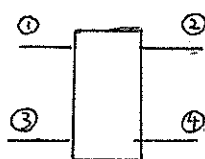
「サーボモーターは入力した信号に対応した角度に向けて回り続ける」とあるのです。ここまですべて分かった人もいないかもしれませんが、サーボモーターをある角度まで動かすためにはその角度に対応した信号をサーボモーターに入れ続けなければならないのです。

そのため、プログラム上ではサーボモーターの動く角度からそれだけ動くのに必要な時間を計算し、その時間だけ1周期20m秒を繰り返す。という物を作る必要があります。例えばこのサーボモーターは60度/0.6秒となっているので90度動かすには0.9秒、つまり5周期分繰り返す必要があるのです。これを短くし過ぎると思った角度まで動ききる前に次の角度まで回ってしまいます。

五、歩行パターン

さて、ここまですべて基本が説明できたので歩行パターンの説明に移ります。

ここでは今回Titan539にプログラムしてある4種の歩行パターンを上から見た図で説明していきます。



上から見た基本状態です。
何も動作をしていないときは

基本的にこの状態です。

便宜上、それぞれの足に①～④の番号を振っておきます

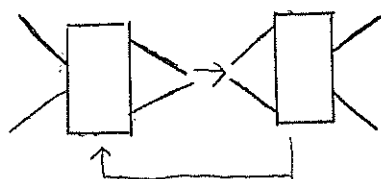
I、トロット歩行

いくつかある四足歩行パターンの一つ、

常に2足が地面に接地するようにして歩く

そのため、不安定だが移動速度は速い

動歩行と呼ばれるもので、重心が常に動いている



四角形の対頂にある足が同じ重みをする。
足を上げるタイミングが足を前に出すときだと
前進、後ろに下げるときだと後退する。

移動速度は20cm/sくらい

II. フロール歩行

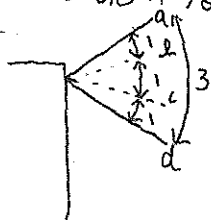
同じく、パターンの一つ、このニックスタにもいくつかあるが、

それはジャンプの必要があるため今回は無視。

常に3足が地面に接地しているためとても安定している。

重心が動かない静歩行と呼ばれるものである。

ニキを説明する前に足が一番前にあるときと後ろにあるときの角度を3とします。



つまり左図のようになります。

そして便宜上それぞれの時の位置を前からa, b, c, d.とおきます。

足の先に黒丸がついているものは接地している。

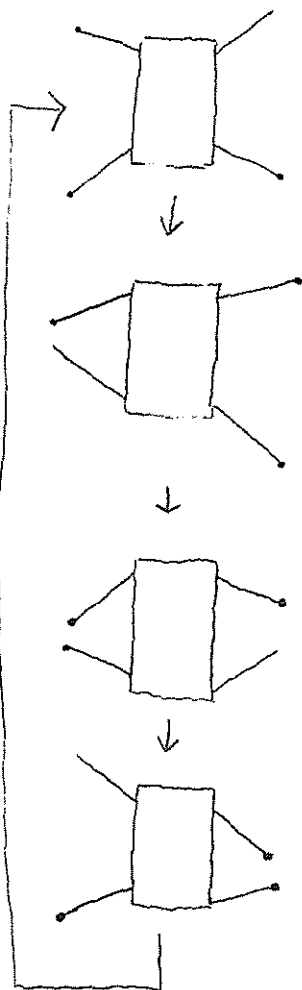
ついていないものは遊脚(接地にない浮いている足)です。

①はb

②はa

③はd

④はc



①は1下げてcに

②は1下げてbに

③は3前に出してaに

④は1下げてdに

①は1下げてdに

②は1下げてcに

③は1下げてbに

④は3前に出してaに

①は3前に出してaに

②は1下げてdに

③は1下げてcに

④は1下げてbに

図を見て分かるように それぞれの足が時間差で

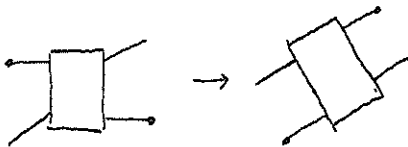
$a \rightarrow b \rightarrow c \rightarrow d \rightarrow a \dots$
足を
浮かべて

という動作を繰り返しています。

移動速度はトロットの半分くらいで 10cm/s といったところ。

Ⅲ. 旋回

その名の通りです。方向転換時に使用します。

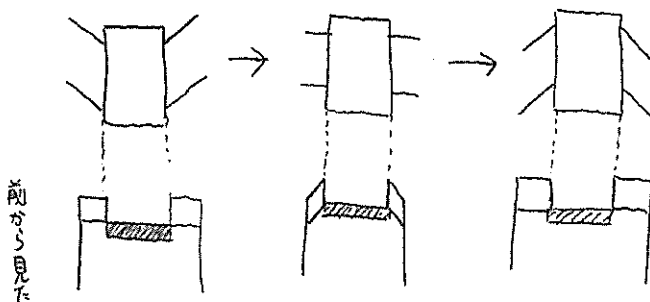


②、③を前、後に重かいたあと、
それを元に戻すときの逆歩脚を
変えることで回転します。
石も原理は同じです。

Ⅳ. ? 式歩行

中3実習部のオリジナルです。

まず 4足を全て 前に出します。



そして4足を全てを上に上げながら
前に出します。

なぜこうするかというと、この歩行本体を前に出すとき
全ての足が接地しているため足先と本体との間の幅を
一定に保つ必要があるため、足を上に上げ 図のように
ななめにして 幅をせまくした のです。

この歩行法、利点はとくにないのですが
重かきがコミカルで 見ていて面白いのでのせました。

六、最期に

今回は去年に続き 4 足歩行ロボットを作りました。

去年の反省を生かしモーターにパワーを求めた結果サーボを使うという結論に達しこのような物になりましたが。

動かし方さえ分かればほぼ確実に歩行まではいくであろうことは分かっていました。そのため今回の製作物はプログラムにかけた時間がかかなり長いです。ですがその分わかりやすく汎用性のある物を作ったつもりなので、よかったらこれからサーボを使う人は使ってください。

使ってくださいと言っておいてなんですがサーボモーターは危険です、制御法さえ確立してしまえば何よりも簡単に正確なモーターになってしまいます。なのでどうしても正確な制御がいる部分だけに他のモーターと組み合わせて使うぐらいがいいと思います。2 足を作るなら別にいいかもしれませんが、その場合ただ歩くだけでは物足りないと思うので、ダンスでもさせて下さい、それですぐに価値は上がります。

2 年ロボットを作りましたがその中で色々と学んだことがあります。が、それをここというには場所が足りないので、ロボット製作者専門の懺悔ノート「兄貴の懺悔録」に記したいと思います。ロボット作ろうと思ってるヒトはそれを読んでください（内部向けになってしまいましたが）。

これからの物無におけるロボット技術の発展と存続を祈って……



実際ロボット製作はこんなもんです。

失敗にめげずに頑張って続けていこう！

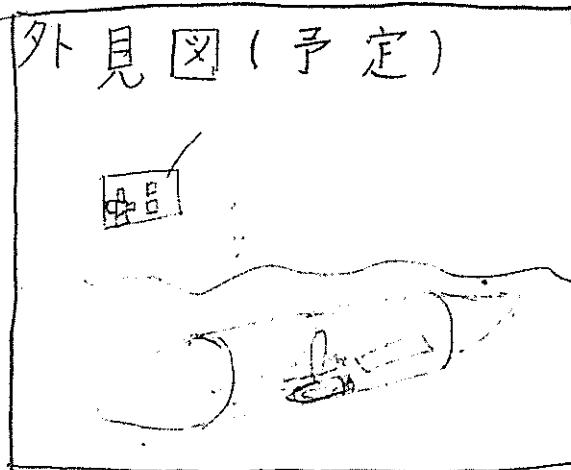
トローガ停水己夫

名前: 特に意味はあません

制作者: H1 西脇

276, 277, 278, 279

物無人

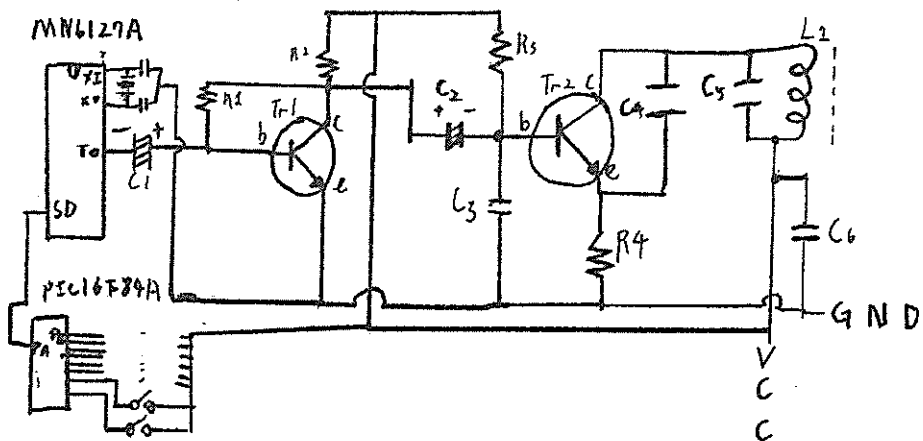


見てもわかりませんが、

これはラジコフ港水艦です。FMで電波を飛ばし、水中の本体を動かそうというものです。

回路

回路はリモコン(送信部)と本体(受信部)があります。まず送信部から



T1,2 : 2SC1815

L1 : 8μHのコア: 0.5mmの線3回巻く

R1 : 820kΩ

R2 : 10kΩ

R3 : 22kΩ

R4 : 100Ω

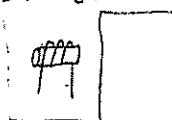
C1 : 10μF

C2 : 4.7μF

C3,6 : 0.0047μF

C4 : 10PF

C5 : 30PF

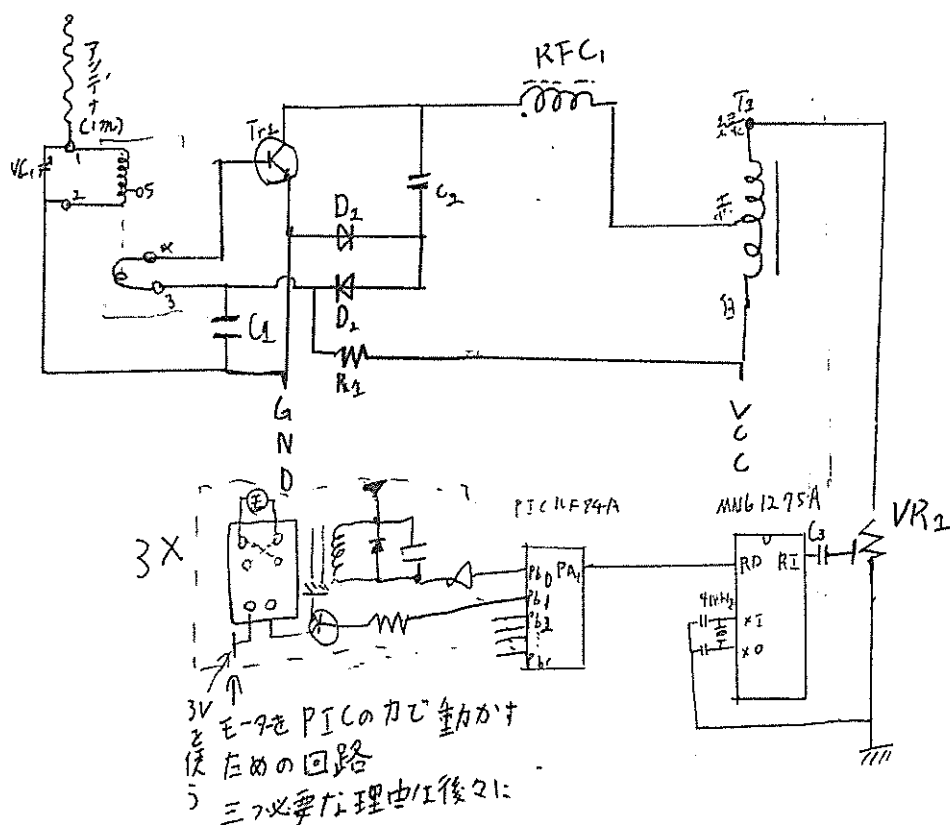


PICにはスイッチの入力に合わせて
出力するプロトタイプを入れた
玉の共振発振子の1/3072の速度で
出力させねばならない。

MN6127Aはデジタリ→アナログ
変換ICである。

これはPICからの出力
をアナログにかえ
約98MHzで出力する回路です。

受信部 (潜水艦本体)



T_{r2} : 2SC1815
 D_{12} : 1N60
 $RFC1$: 2mH
 $R1$: 1M Ω
 $L1$: SL-55GT
 VC : ホリバリコン
 $C1$: 0.04 μ F
 $C2$: 100pF
 $T1$: ST-30
 $VR1$: 1k

説明

PICには入力が付いたプロ
 グラムを書きこむ

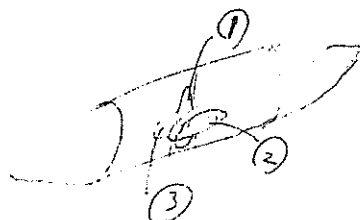
受信部では、MN6127Aを
 アログ→デジタルとに使う。

これは、電波を受信し、デジ
 タルに換え、PICでこれを処理して、
 三つのモーターを動作させる回路です。

この潜水艇の動き方

この潜水艇には 3つのモーターが
 ついています。かじはありません
 上下する時は ①のモーターを使
 われ、前に進む時、後ろにさがる時、
 右へまわる時、左へまわる時は、

②③のモーターを使います。

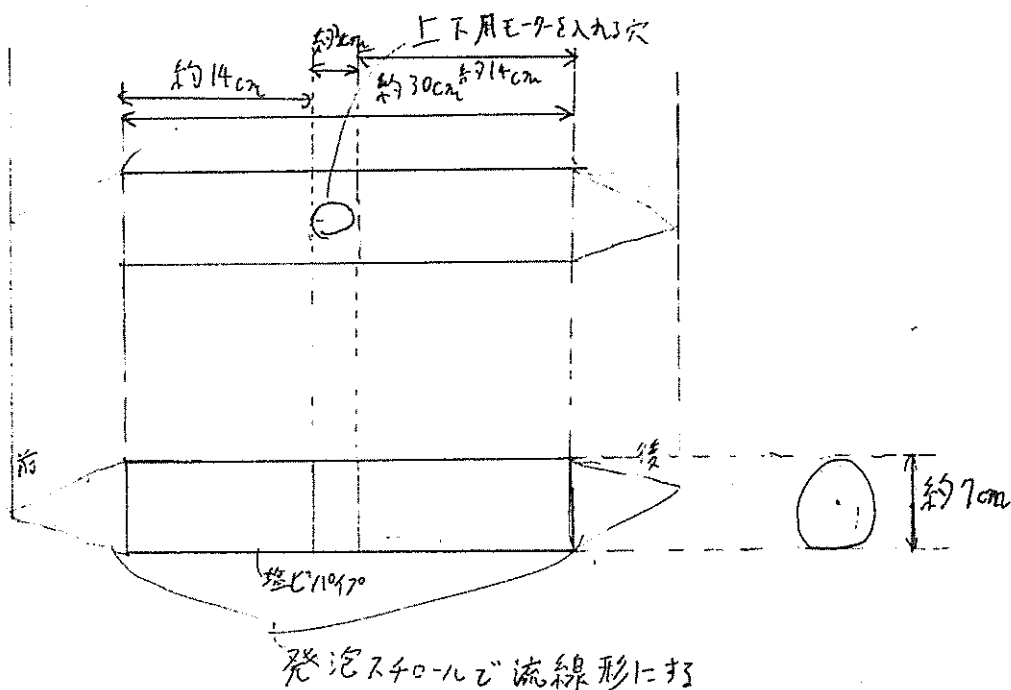


このモーターは、マダチ水中モーター
 を、外部電源型につくりかえ
 た物をつかいます。

この潜水艇は、内部電源を使わなくてはい
 ませんが、どれを使うかは、決まっておられません。せい
 だり15V~15V
 くらいのものを使う予定です。

設計図

はありませんかたまた等
 派のページへ。



この作り古の場合比重が水にくらべとこで軽くなって
しまうので重りを大量に入れないとすまたりと思わ
れます。

作ってみた(作っている)感想

今年は代替水艦を作っているわけですが、かなり難しいです。

それと部材の話になってしまっているのですが、週に3回出る
時の時間の足りなさを感じました。

なぶん、毎日出て先パイ方のアドバイスを
聞いていければ、すでに完動しているで(ふ)。

まあ、作って勉強になった所などがあるの
で、自分にとって正になったと思います。

なほ、この回路設計図通りに作って動かない場合がでてきますが、その際は一切責任は
とれませんのでご了承ください。

クライマー CLIMBER

制作者 M2 竹島啓純
協力者 M2 高崎尚人
物無の皆さん

こいつはLEGO MINDS TORMSという、
LEGO ブロックで、ロボットを作る物を使っ
た何かを登るロボットです。何を登るかはまだ決
まっています。

機構

まず登る物をつかむ手ですが、ウォームギアと
24歯ギアを2つづつ使い、24の2乗の57
6倍もの力がかかるようになっています。
次に腕ですが、40枚歯ギアとウォームギアを
かい40倍の力を出しています。

プログラム

両手を開いた状態から

左手を開く。←

腕を180度回す。

左手を開ける。

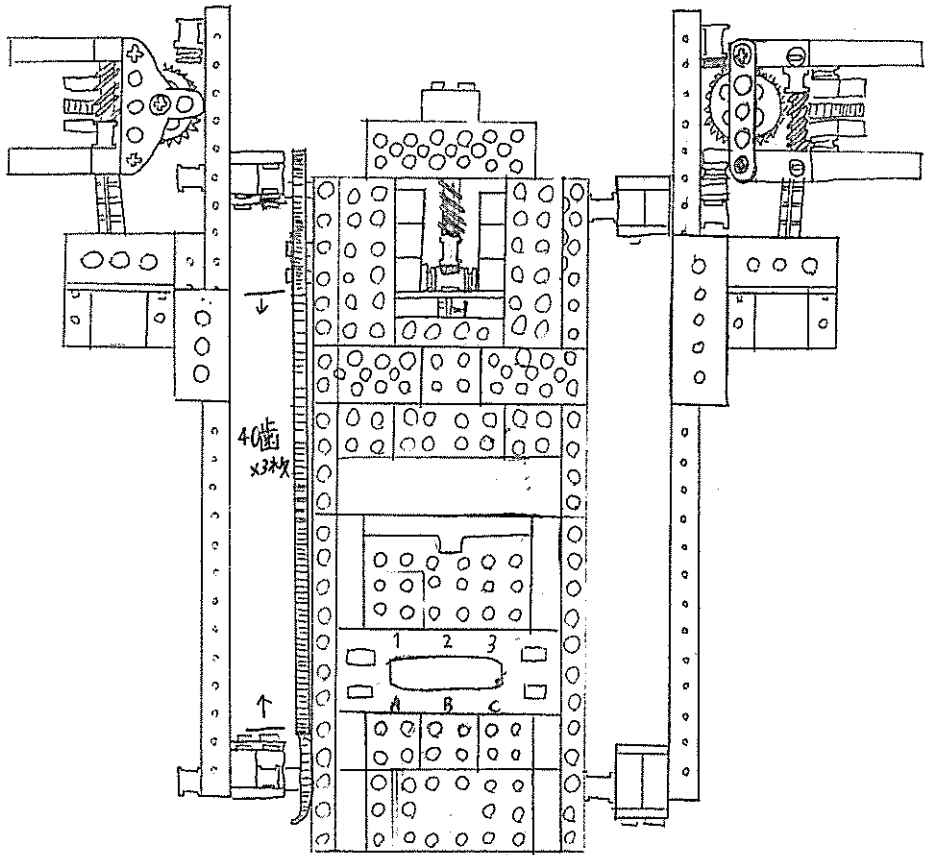
右手を開く。

腕を180度回す。

右手を開ける。

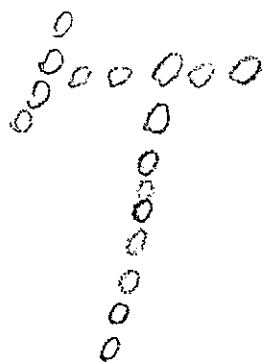
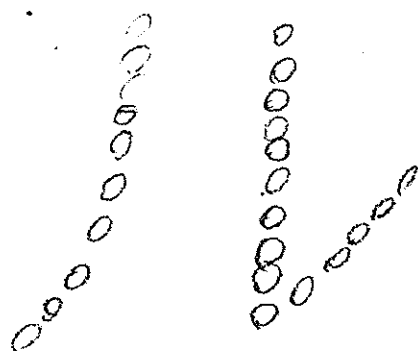
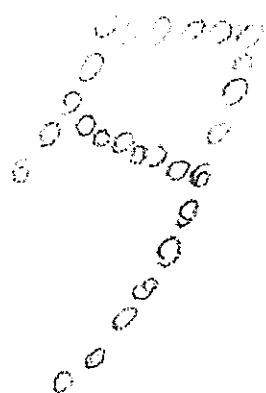
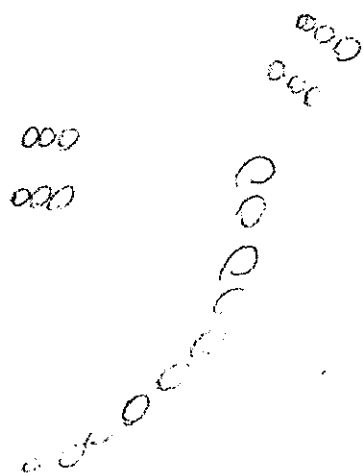
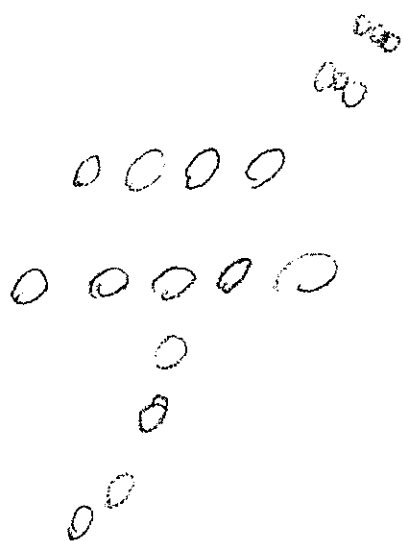
左のことをただくり返し続け
るだけです。
プログラミングにはNQCと
いう言語を使いました。

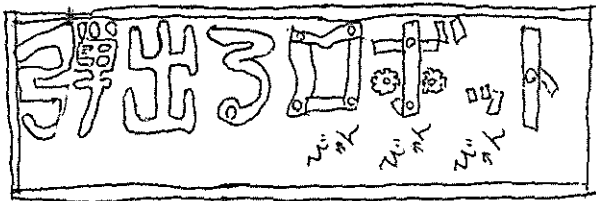
外見図



感想

当初はカーテンを登るはずだったのですが、
 カーテン→垂木→ロープと、登るものがかあていきました。
 作業は思ってた以上に川貞調にすみましたが、
 途中で部品が足りなくなったり、予想以上の負担でギアが害られては、
 たりもしました。
 パワー不足で登れなかったり、パワーをかせる為に
 減速しすぎて時間がかかったりしたもの、
 結局は楽しんで作業できたので良かったのではないかと思います。





製作者 M2 大塚 雅人
 回路図 舟渡 隆平さん
 協力者 無線班の 方々
 物理部

ストーリー

21XX年、地球上で第三次世界大戦が起こり、世界は壊滅的な状態になっていた。

そんなある日、ヲタニケニシマの父親が何者かに殺された。怒ったヲタニケニシマは父親を殺した人物を探すための手掛かりを探ることにした。その時、地下1の入口を見つけた。地下の入口を開くとそこには五体のロボットが!!

そのころ、地球防衛軍本部ではー。

「カーロボ「ロボットが宇宙から接近して来ます!!」

隊長「な、何!まさかインカーダーか?」

「え、どうもそのようだな。会議を開くぞ!」

隊長「会議が始まった。田中「うけです。どこかロボットを所有しているジャックボンをさがせませんか?」

隊長「それなら、確か『カムンダ隊』があるはずじゃ。」

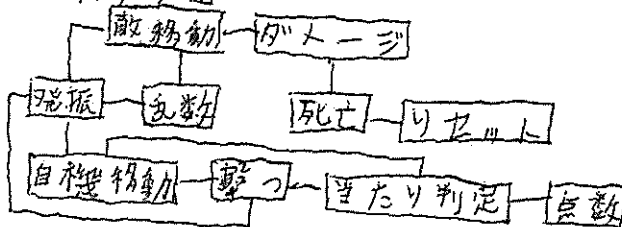
さらに兵士が四人、インダ隊を使わせてもらいます。」

インダ隊と戦うことになった。

作手法

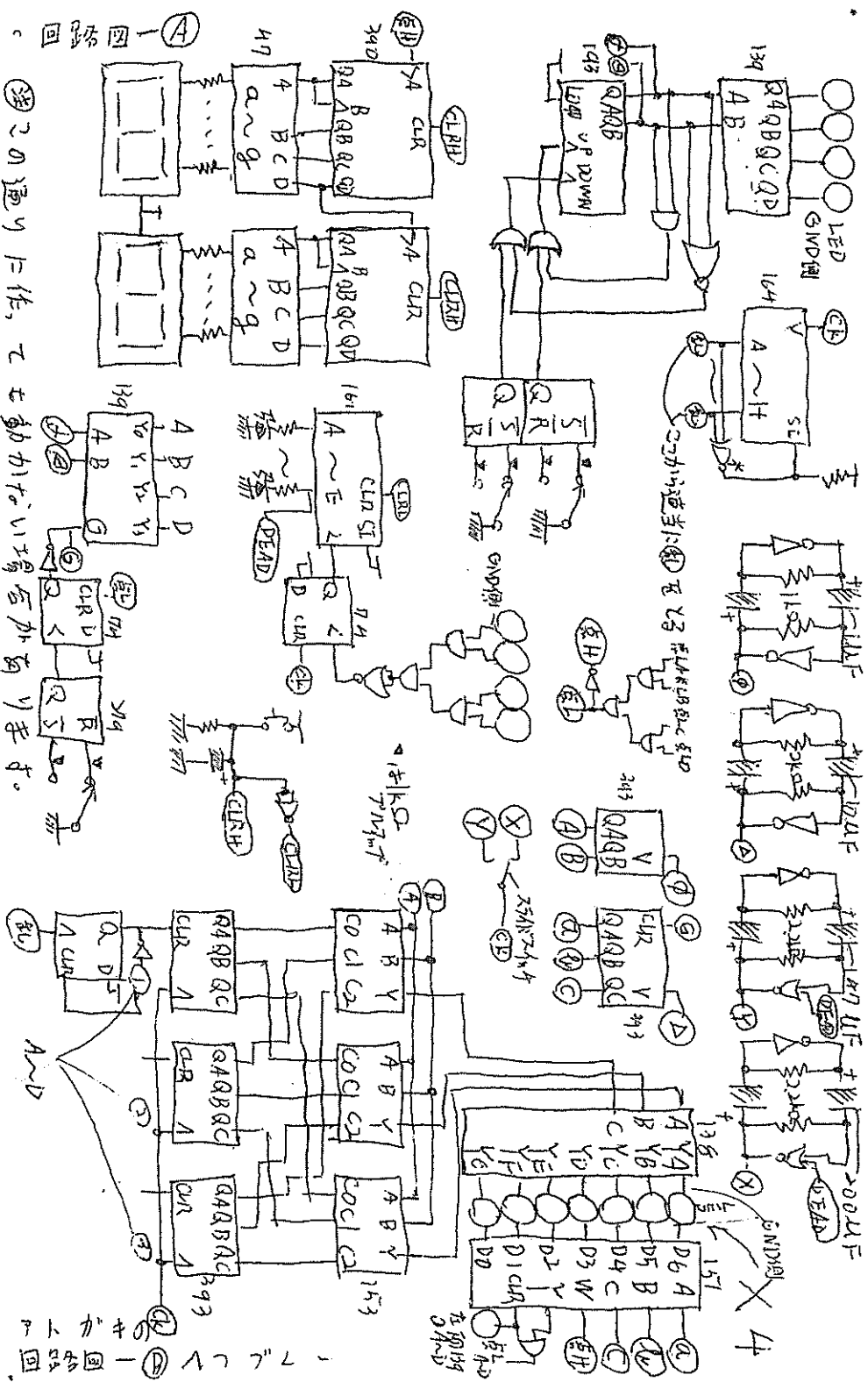
上の十段は敵移動、下の一段は自分の陣地で、左右に動かし、敵を倒します。また、自分の陣地に敵が入ると、ダメージ増え、ダメージになると、ゲームオーバーです。(この調節可能。)

ブロック図



外観





④この通りには、エ動かない場合があります。

④この通りには、エ動かない場合があります。

コボタン

制作者 M2 川崎 敦史
 回路図設計者 H2 森さん
 協力者 物無の皆様方

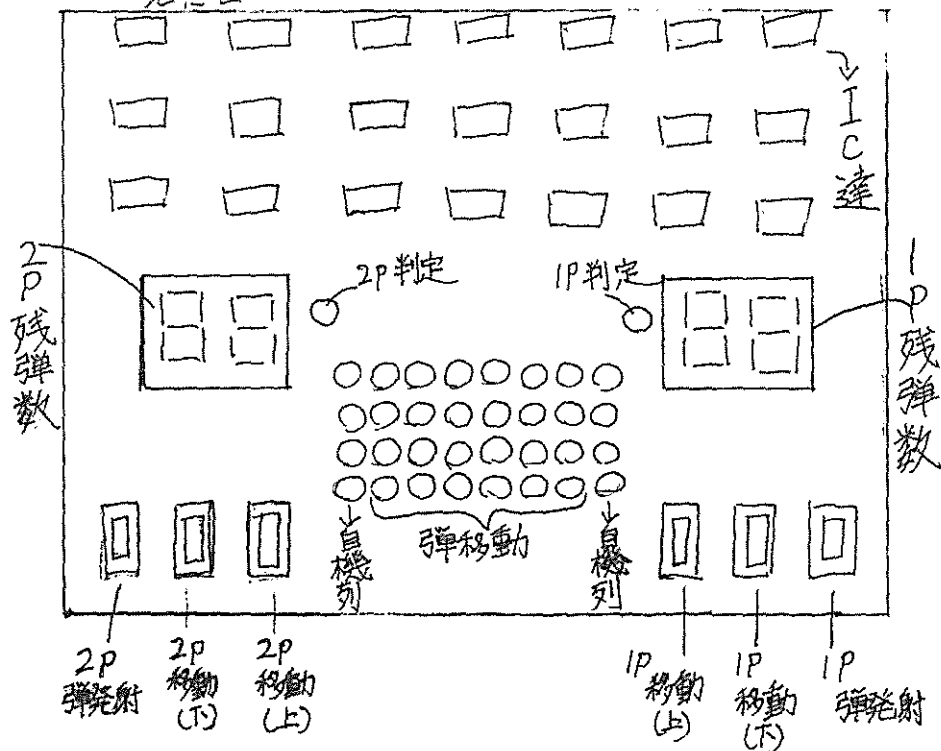
～ストーリー～

この世の中には自分とすごく似た人間が数人
 いるといわれており、人間は自分と似た人物を憎む。この戦
 いはその似た人物を倒すために始まったものである。

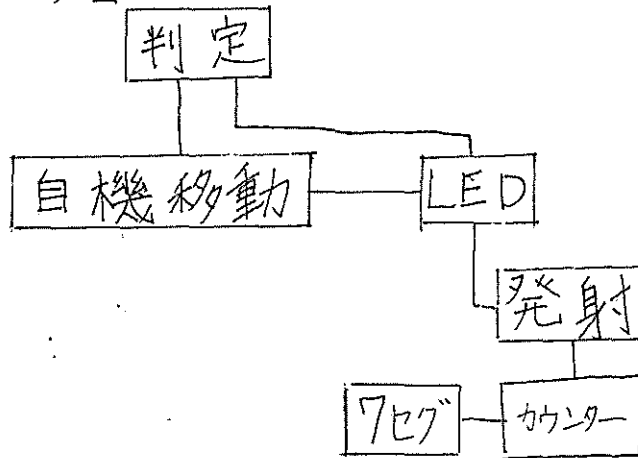
～ゲームについて～

このゲームは2人用ゲームで自機(緑)が相手目差
 して弾(赤)をバソバソうっていくゲームであり弾に
 当たると負けてしまう。残弾数もあるので注意して
 うっていくべし。

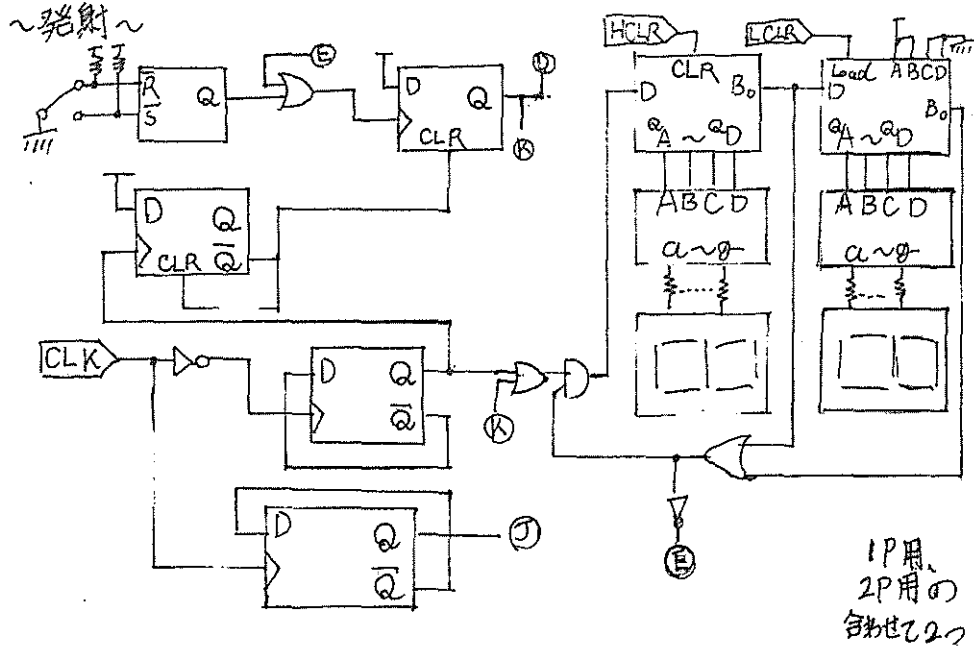
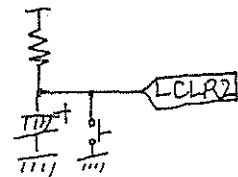
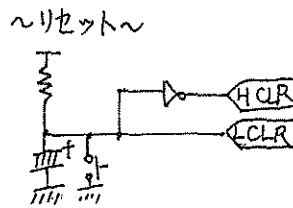
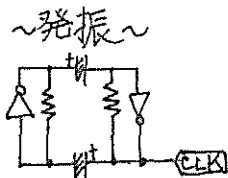
～見た目～

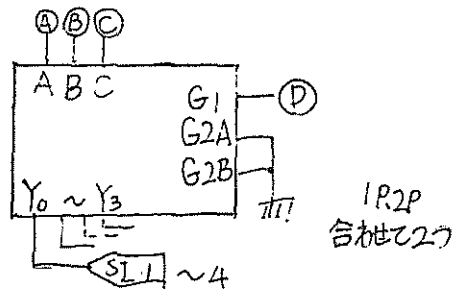


～ブロック図～

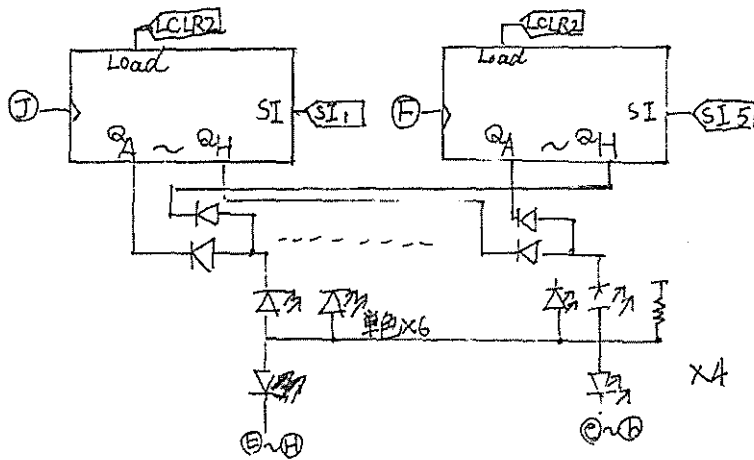


～回路図～

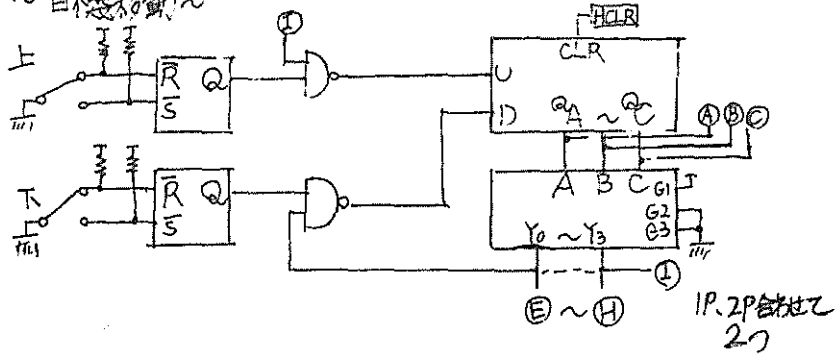




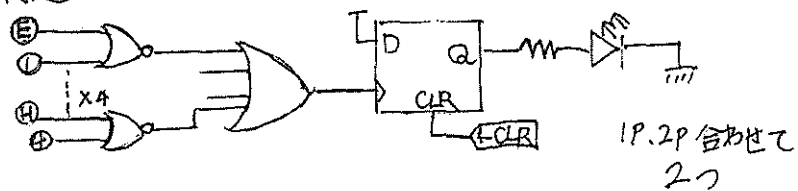
～単相検～



～自機検～



～判定～



録帽子を救え

製作者 M2 木村

回路図設計者 H2 森さん

協力者 森さんをはじめとする
物無の皆さん

銅^{※①}「あー疲れた。」

ベニ^{※②}「何故に?。」

銅「これ書くのに徹夜したんだぜ!」

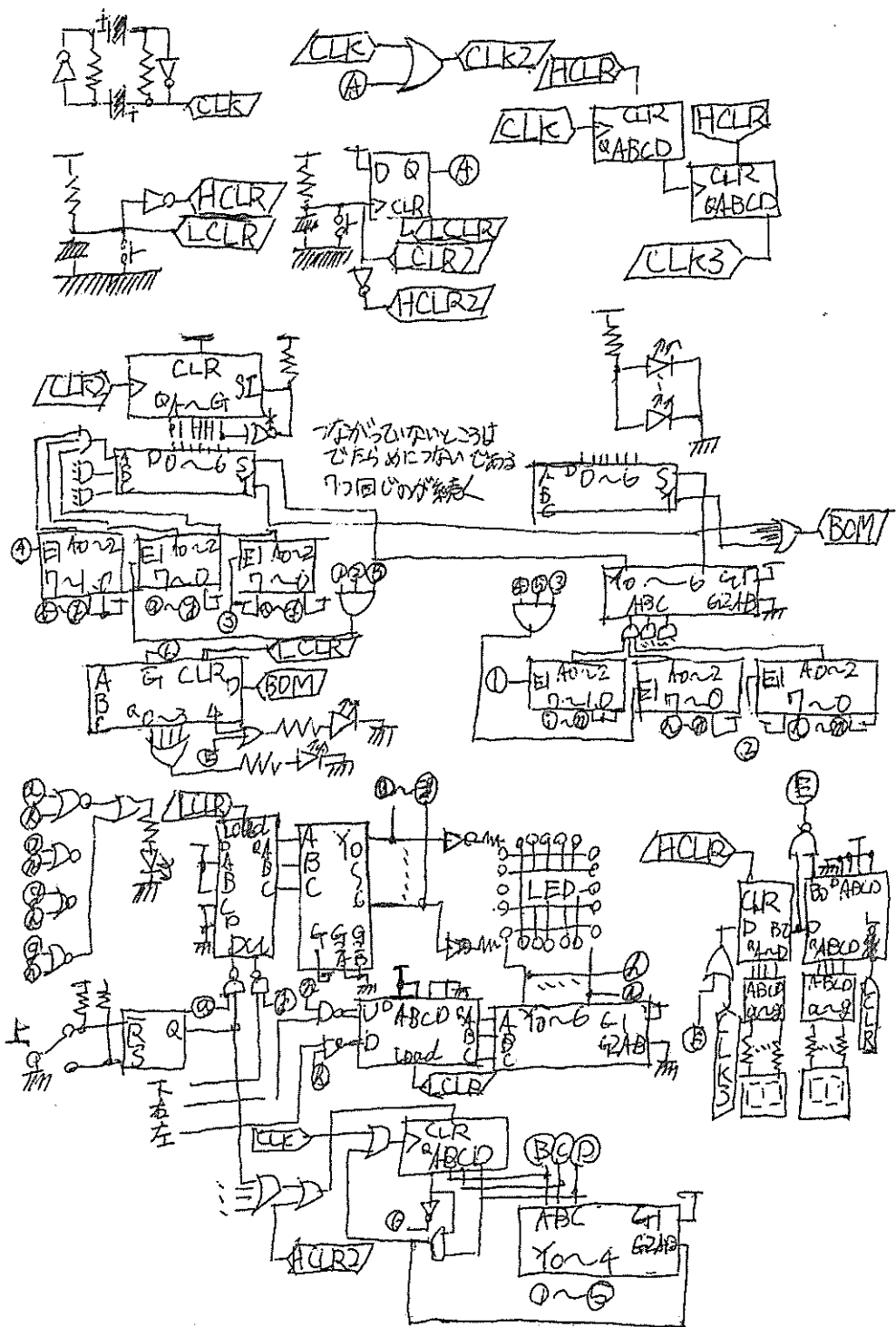
ベニ「相等昔の話じゃん。」

銅「いいの!それにこの製作物11月初めに完成
して12月終りに完動^{※③}しておと思ったら10月完成11月
完動だぜ!!俺様 自分の事寝めちゃう。」

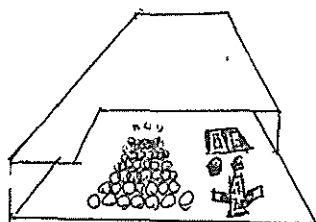
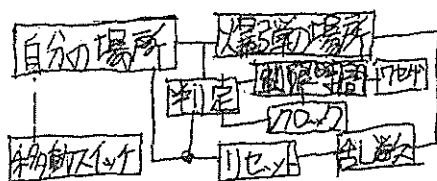
ベニ「はいはい判ったからちゃっとこの製作物の説明
しようぜ。」

※① 銅で溶かす係 ※② ベニヤで囲む係
※③ 完全に正常な動作にすること

銅「まずは回路図からいくよ!」



銅「まあ、こんなもん？ 次はブロック図と外觀だよ」



銅「このゲームのルール説明だぜい！」

最初は●にいて④に向か、て進みます
しかし④と●以外のところには爆弾があるか
わからないので進みます。到着するとチェンジで
死にます。死なないために自分は上下左右に
爆弾があるところを探るセンサーを持っています。
それとセンサーが隣接すると、危ないとき黄色
ゲームクリアすると緑になります。

銅「ひゃー やつと終わったぜ」

銅「でもホルペンでなぞると仕事はまたあったり」

あるんだよな 鬱, 鬱

ベニ「第二製作物のストーリーもあるぜ」

銅「やめてくれー鬱^{ふさ}ての画数の多い字書いただけでも鬱^{ふさ}になってたのに。」

銅「あと『発売バカ』好評発売中だよん♪」

へ「そんなもん売ってねー」へ「シッロ」

銅「スズ」

銅・ベニヤ「有り難うございましたあ。」

パチパチ...パチパチ

成太の謎流棋13

製作者 M2 桐山 隆成

回路図設計者 浅井サン

ストーリー

協力 物無のみなさま

A 中学校に在学中の成太君は誰もが認める迷スナイパーだ。ある日彼はこんな依頼を受けた。

④ はあ～ ポケモンやりてえー 他のゲーム買い過ぎちゃった。

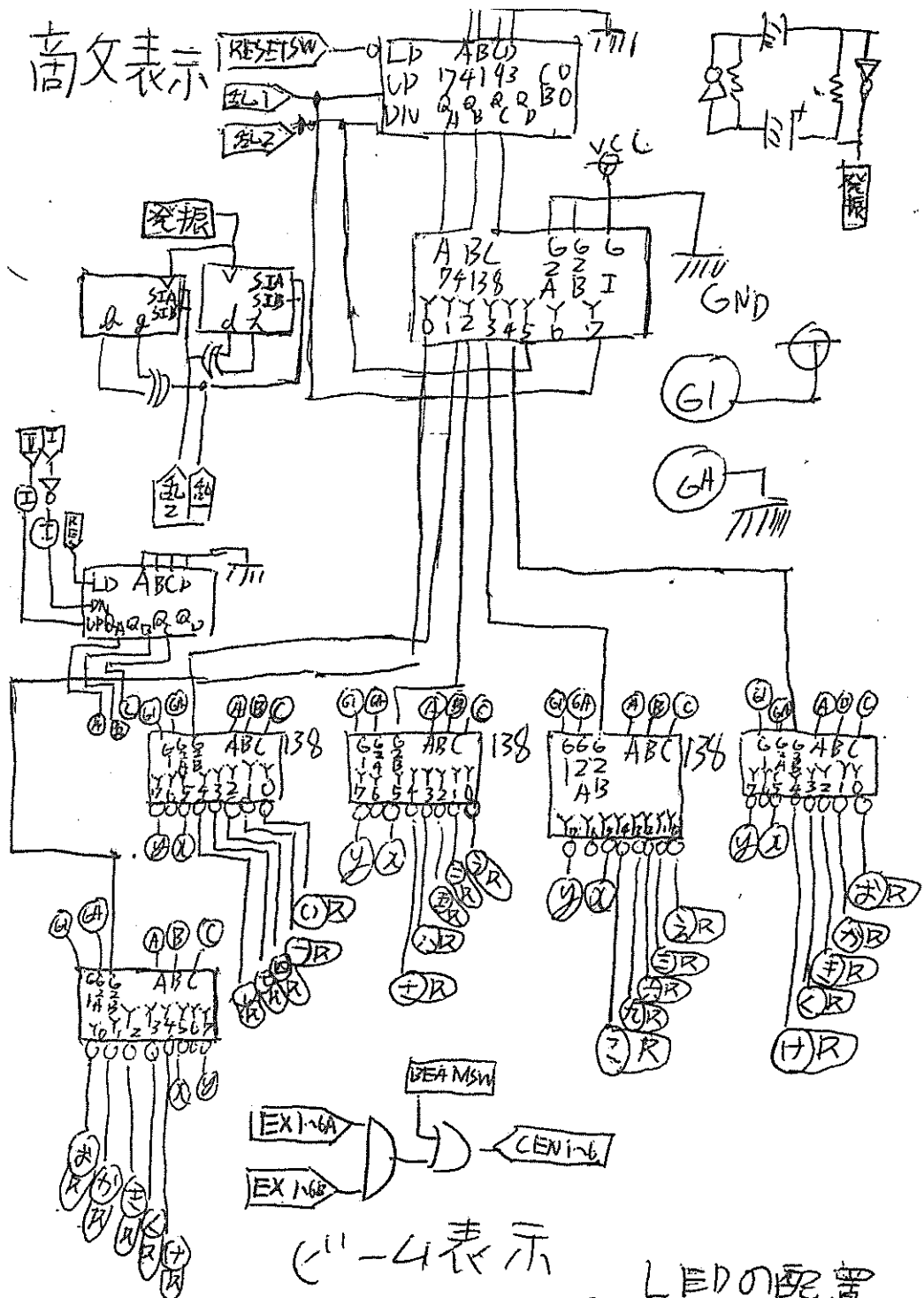
⑤ え？何がやりたいって？ポケモン？それなら俺が新発売のエメラルドバージョンを弄っているよ？でもただでかしてやるわけにはいかぬえな……。そうだ、担任のXXX最近調子のこるからさあ、適当にばらしておいてくれなかな……。エメラルドバージョンあげてもいいぜ。

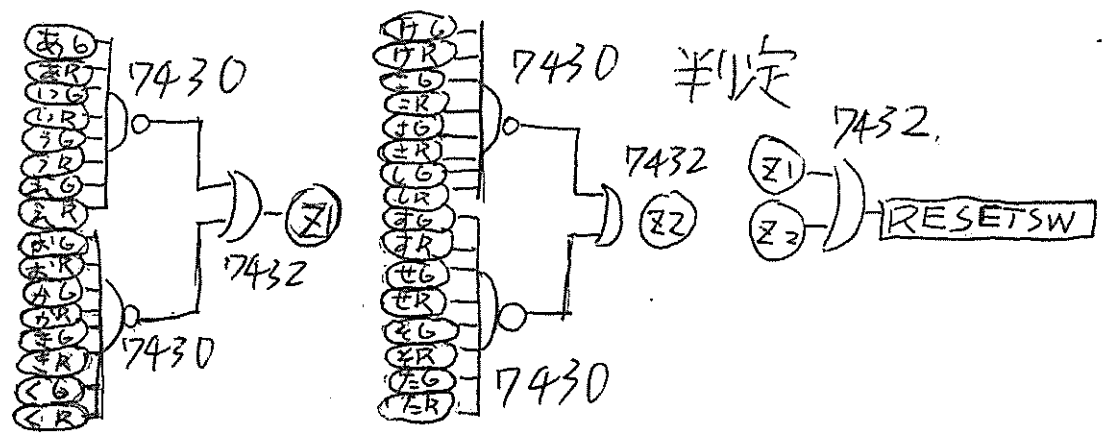
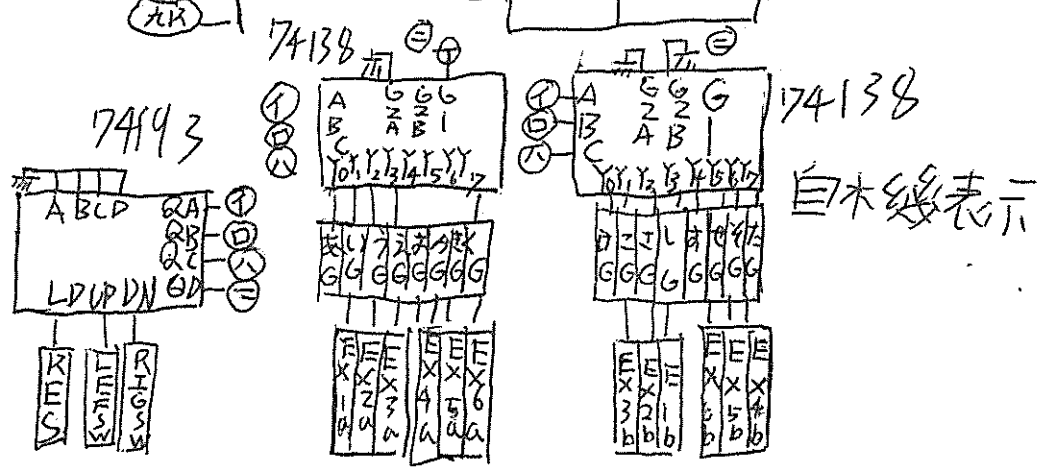
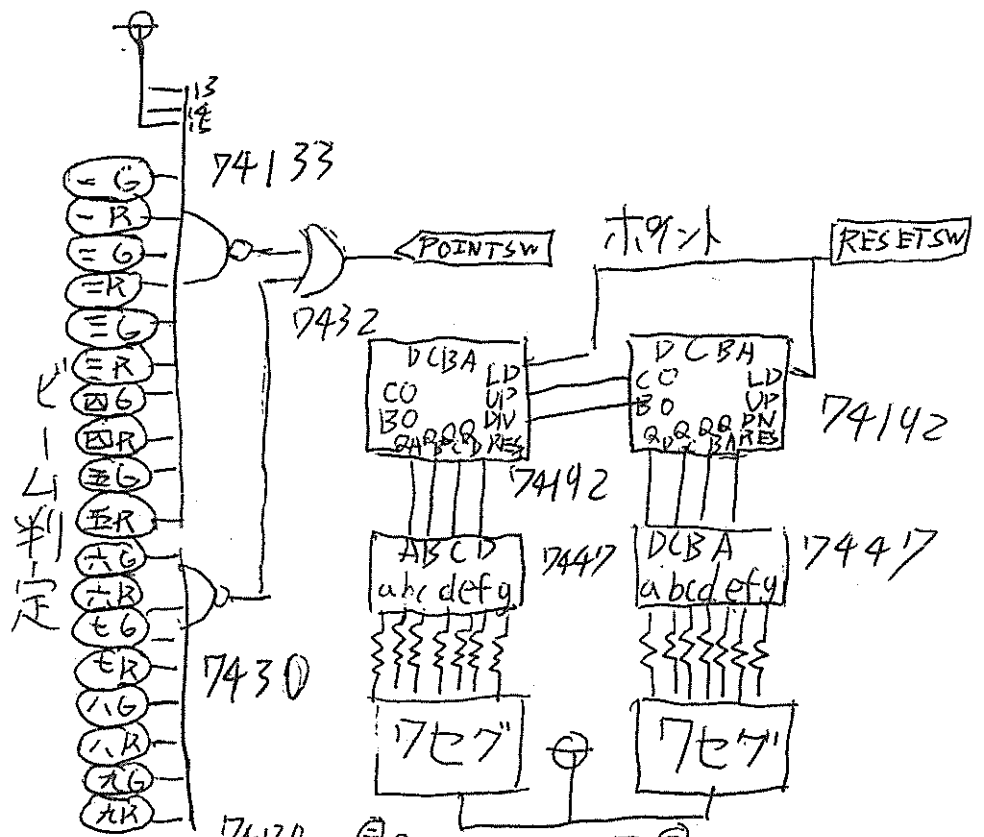
⑥ え？くれるのかい？！ほんとうだね？よし、俺のゲームガンでXXXなんて瞬殺だぜ！！！！でもこれは重しレクラスで一番デブで負弱な俺に扱いこなせるかなあ……。まあ大丈夫だろう。

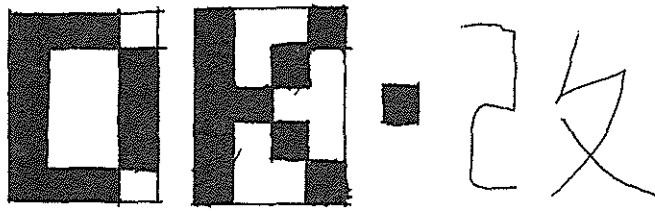
という訳でXXXを殺すことになったらしい。

ルール

このゲームは、5×5のフィールド内に成太（主人公？）と、XXX（ターゲット）の二つが存在し、それぞれ緑と赤で表示されます。主人公の成太をプレイヤーは移動ボタンで動かします。外側の16マスしか動けず、右まわりボタンと左まわりボタンがあります。ターゲットはおまを適当に動きます。ターゲットはどんどん動いてくるのでおまよく対峙しましょう。成太はゲームボタンでターゲットを倒せるゲームうつことが出来ます。しかしうちすぎるとポイントが増え死んでしまいます。





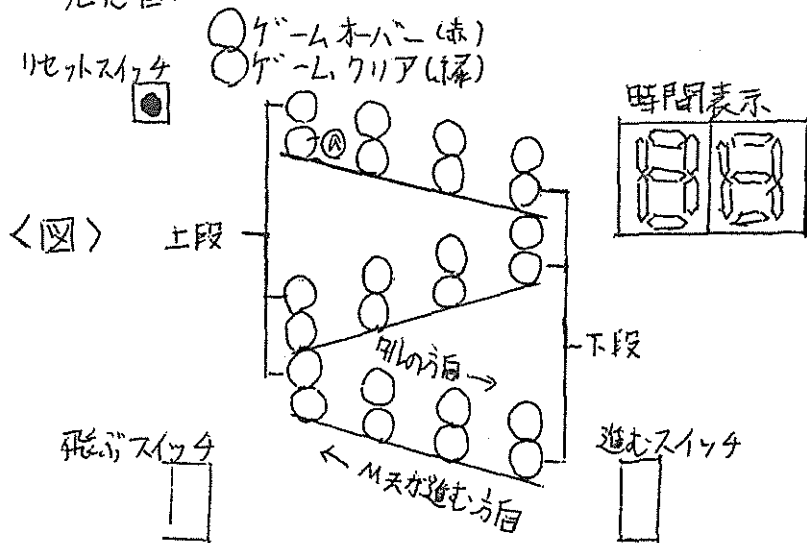


制作者：M2 佐野祐士
協力者：H2 毎渡さん、森さん、物無の方々

～ストーリー～

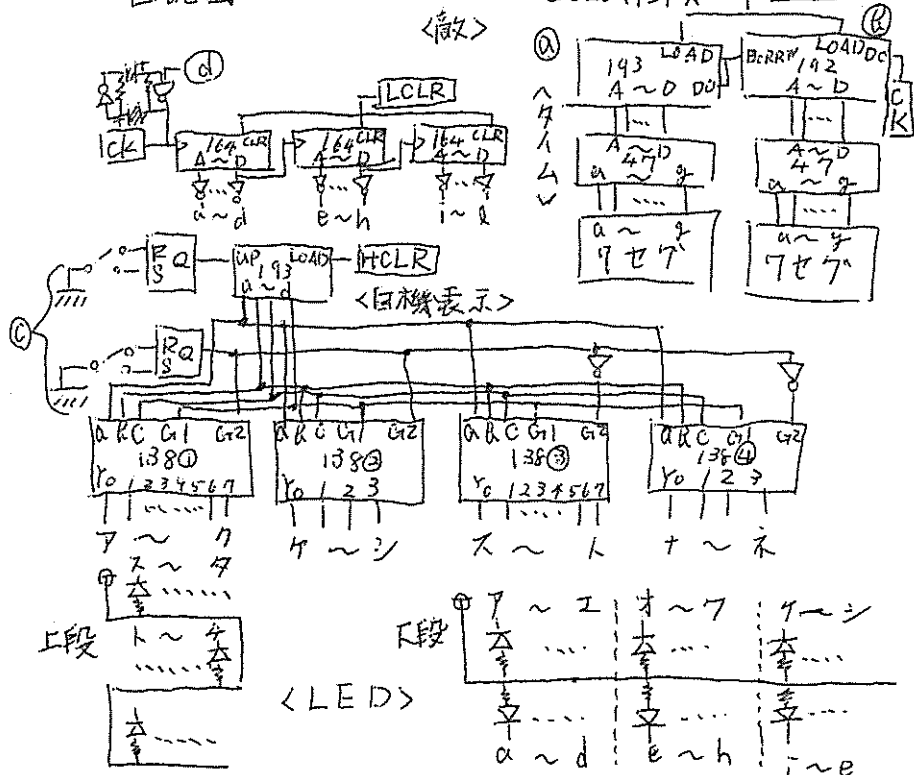
某N社のM夫は、桃姫と平和な生活を送っていました。しかし、ある日DKが突然やってきて、無理矢理桃姫をさらっていきました。M夫は桃姫をとり戻そうとしましたが、DKはタールを輾がしてきます。どうするM夫！

～見た目～



基本的に、ム夫は上段と下段(管段は下段 飛ぶと上段)も、タルは下段のみを進みます。
進むスイッチ4で進み、飛ぶスイッチ4で飛びます。
戻ることはできません。また、飛ぶながら進めません。そして緑の自機を赤いタルをよけながら、図中の④の場所まで移動させればゲームクリアでゲームクリアのLEDが光ります。
赤いタルに殺されてしまった場合、ゲームオーバーのLEDが光ります。
又、制限時間の50秒を過ぎた場合、時が止まり、動けなくなります。
ゲームクリア、ゲームオーバー、時間切れのいずれかになった場合、リセットスイッチでまたゲームを始められます。

* 全LS74シリーズ T-HCLR

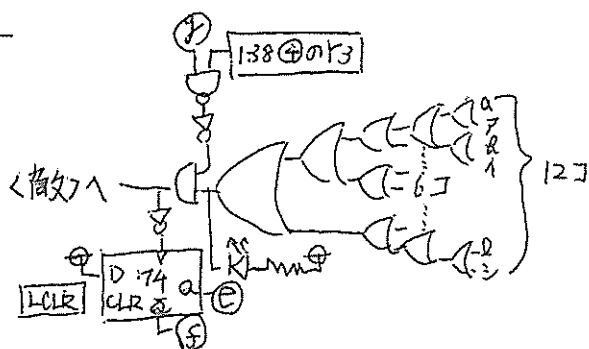
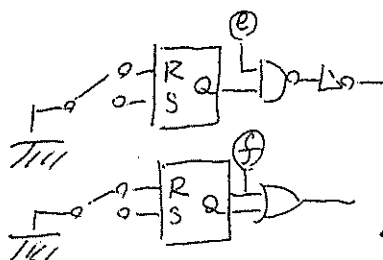


<ジャンプ制限>

<ゲームオーバー判定>

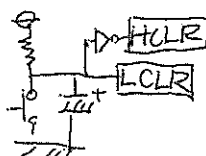
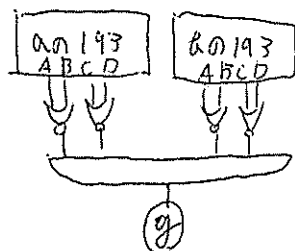
㉒の所の拡大図

㉓の所の拡大図



<時間切れ>

<リフ回路>



～その他、反省～
最初はタルが軽が、て来ないため、連打すれば3段目まで進めるので、その点を何とか...
あと、外のPCのスイッチをゲームのスイッチに接続しようと思っていましたが、面倒なのでやめました。
タイムも何かと速いのですか気にしない方針で。
もう言うことがなくなりました。

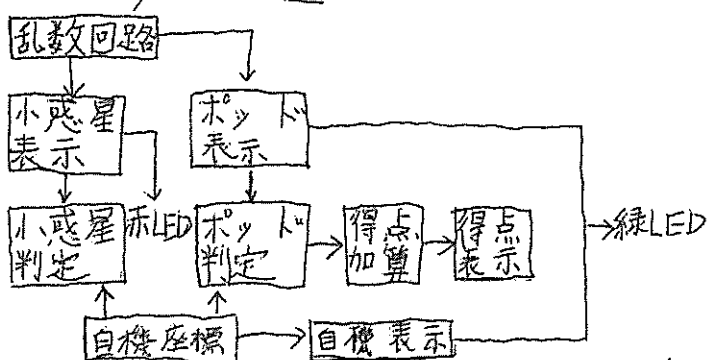
くらげくらげくらげ～

救命力艇

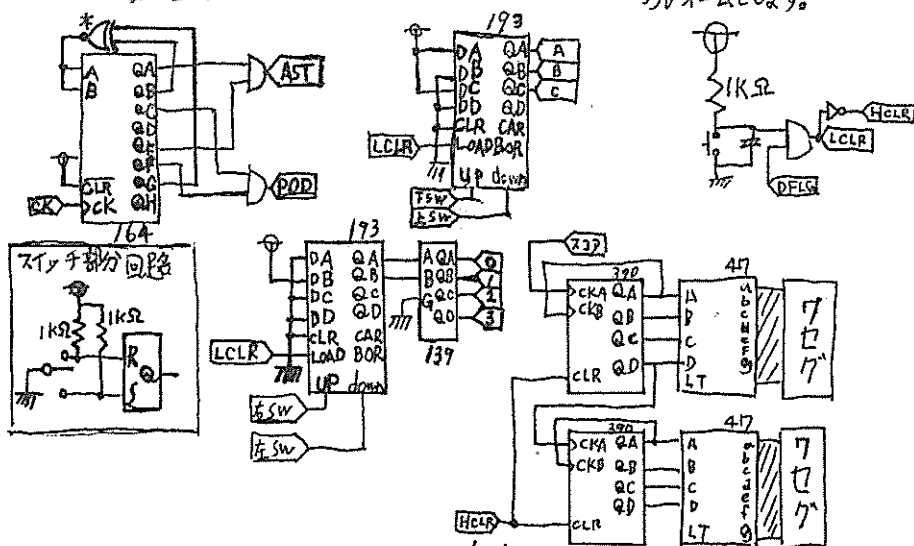
製作：高崎(中二) 回路設計：浅井氏(高一) 協力：物無の皆様
ストーリー

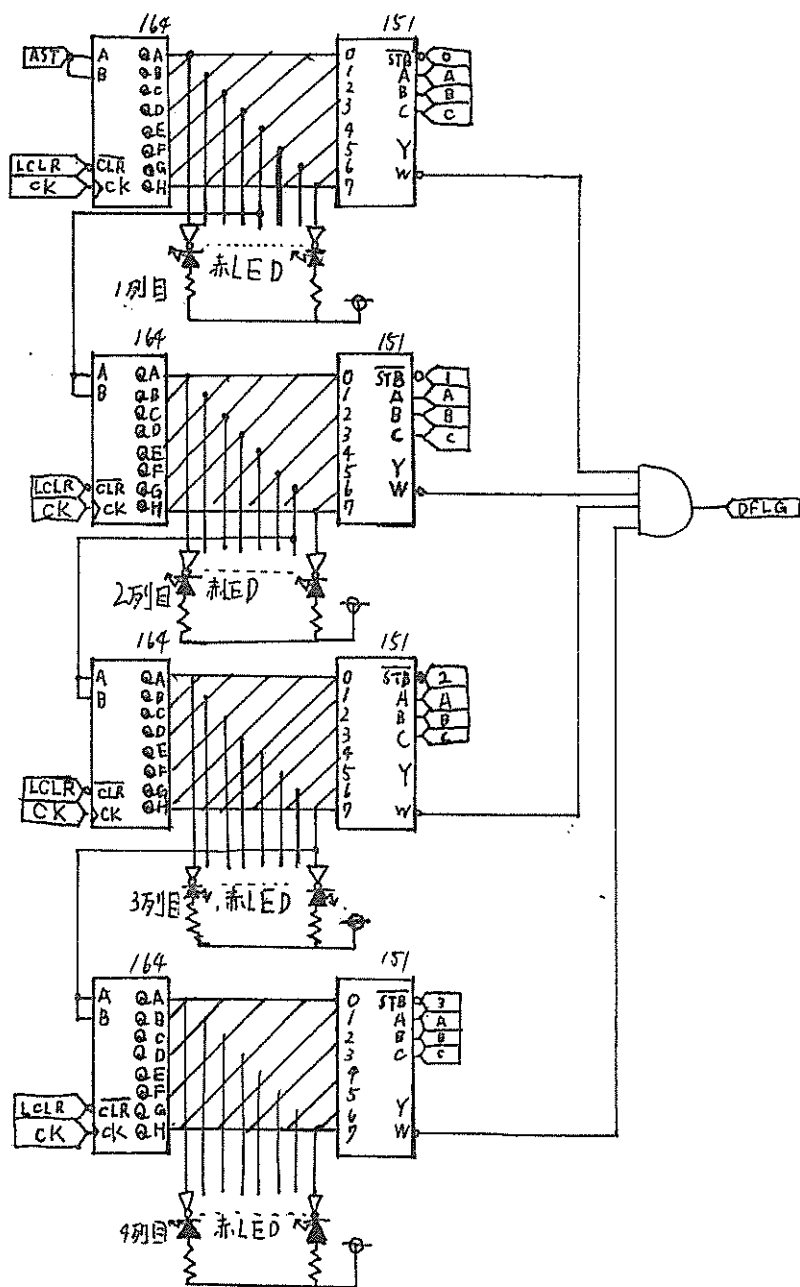
技術が進み、宇宙旅行が一般的になった時代...
ある宇宙豪華客船が小惑星帯につっこんでしまった。
けが人は出なかったものの、多数の救命ポットが漂流
している。
あなたは宇宙救助隊の一員となり、救助艇に乗り込んで
救命ポットの回収にむかうのだった...

ブロック図

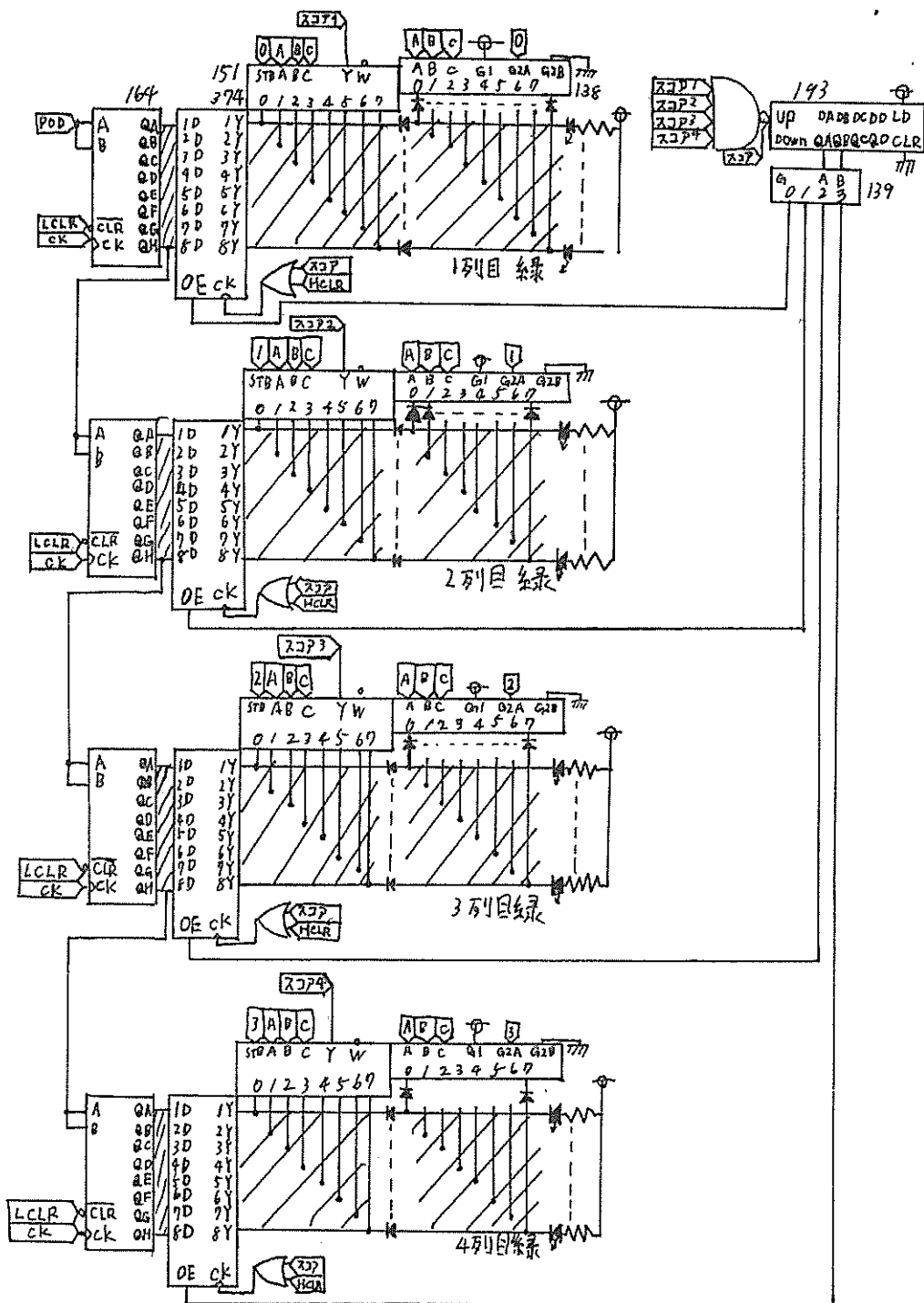


回路各図





CKはお好みの速度でどうぞ。

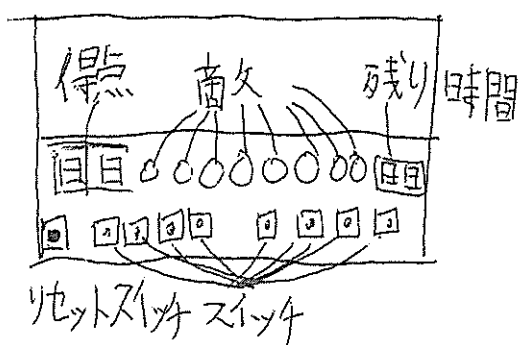


心 | とりあえず完動できたのですが、今ふりかえ、乙み
 木 | ると、初歩的なミスばかりだ。たぶん気がします。
 心 | 一時期全く初歩的なミスを無視していましたが、
 心 | 自分を過信するのはよくないと痛感しました。

単式闘艇侵入阻止

制作者 竹島 啓純(M2)
 回路図設計者 伊藤さん(H1)
 協力 物無の皆さん

%このゲームについて



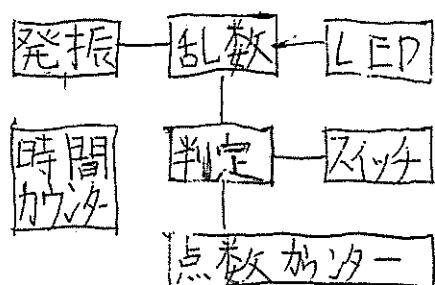
敵のLEDが光っている時に対応したスイッチを押せば点数が入ります。まちがえば点数が減ります。あと時々時間がわかるようになります。

その仕様上表示がおかしくなりますが、バグではありませんのであしからず。

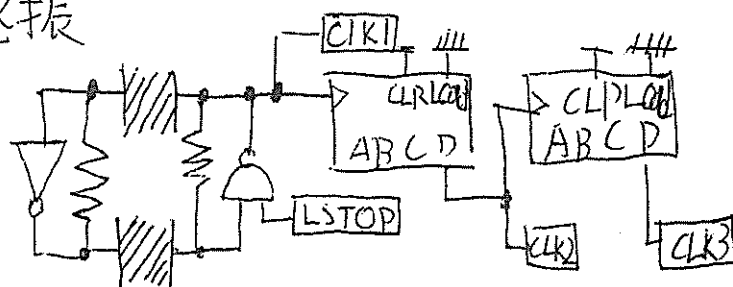
%ストーリー

あなたは雇兵である日次のような依頼がきました。
 「我が社の工場に突如となにものかが海から攻撃を仕掛けてくるといふ青幸が入った。一応、水雷をまいたがまだ心配なので狙撃でな。インタージをあたえてほしい。また弾薬費は自己負担だから無駄打ちすると得点が下がるぞ。また報酬は敵に与えたダメージによって、はらうことにする。」
 さて、かえって多くの報酬を得よう。

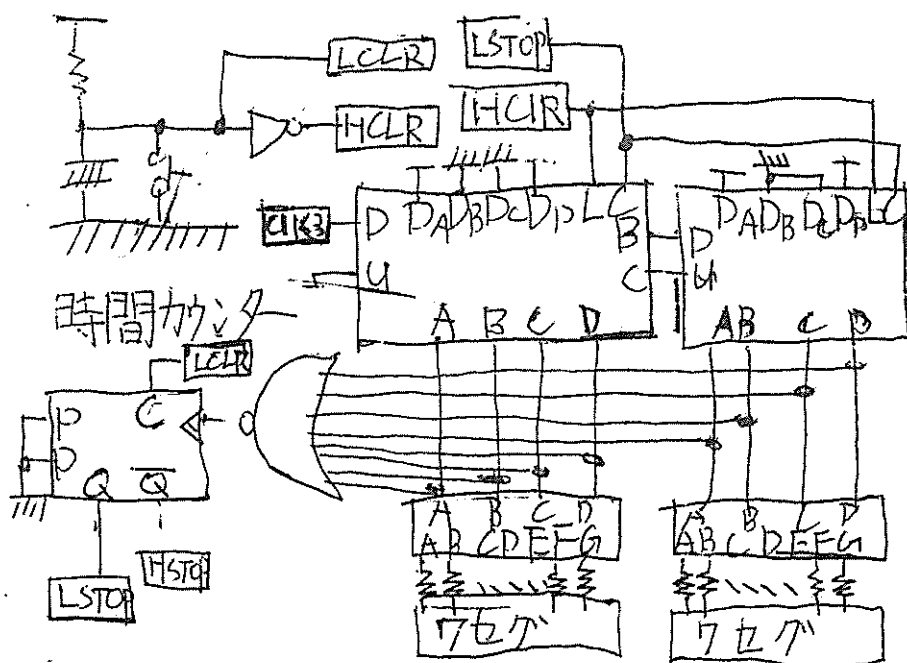
%ブロック図



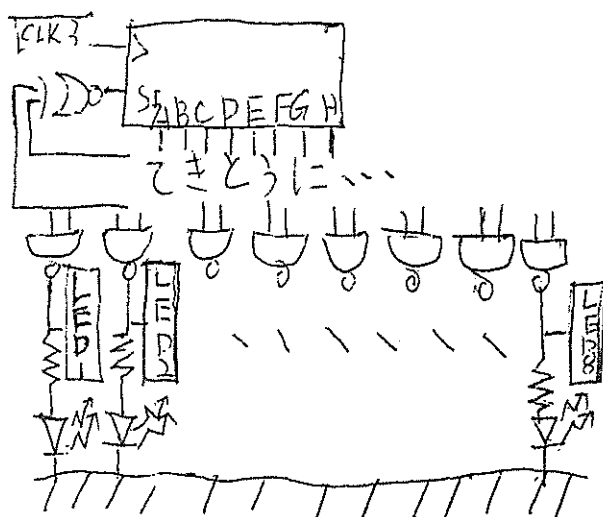
%回路図 発振



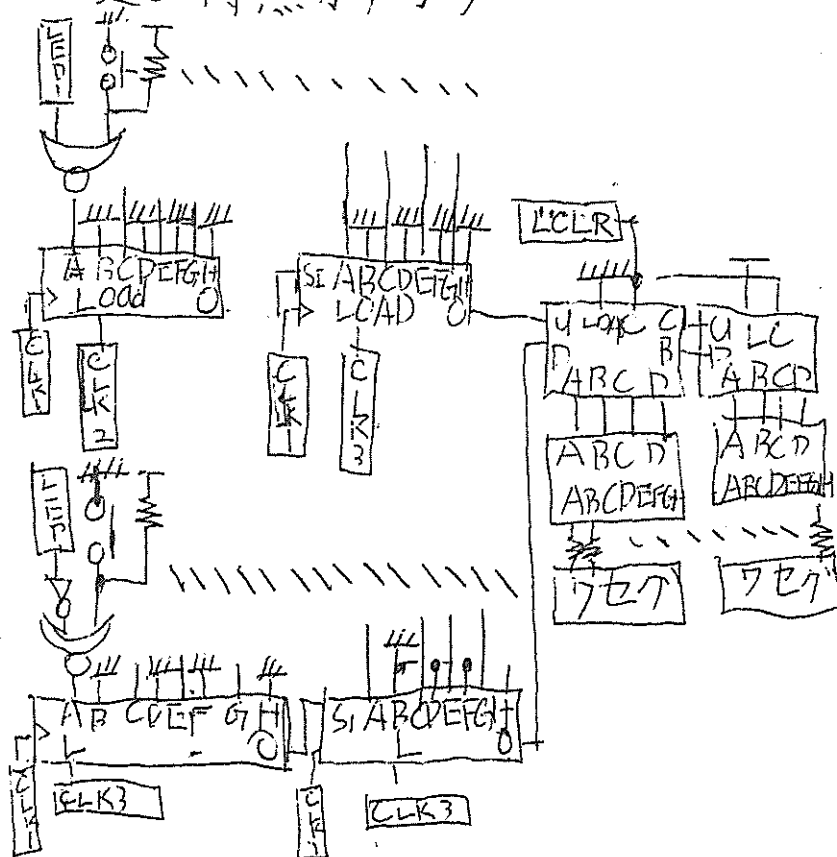
クリアスイッチ



乱数 Y LED



判定と得点カウンタ



逃げろ!!

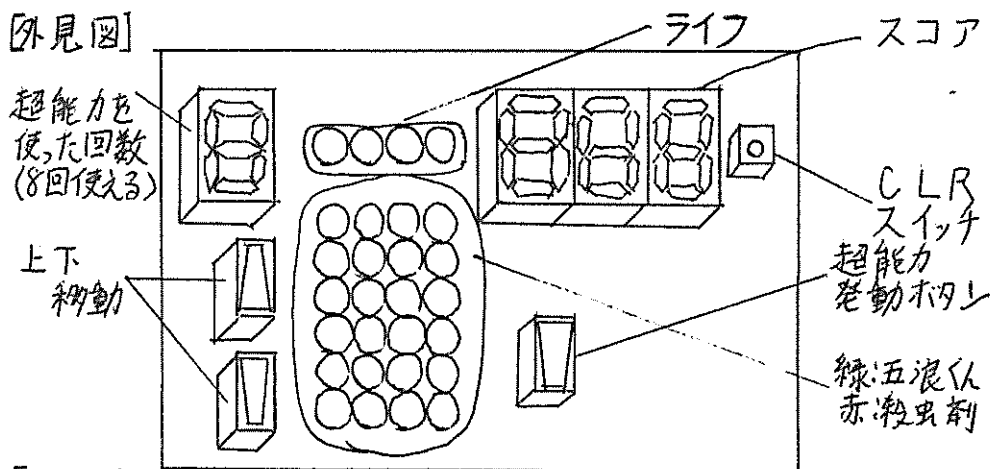
五浪くん!!

制作者 M2 中嶋剛大
 回路図設計者 H1 西脇さん
 協力者 物無の音様

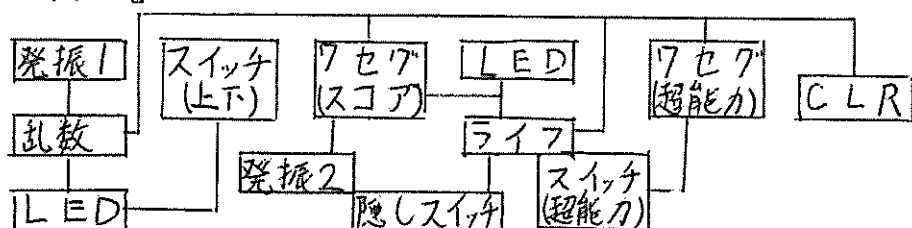
[ストーリー]

近未来、虫たちが人間に対抗すべく、超能力を身につけてゆくようになり、その能力は様々である。軽い物なら吹き飛ばす能力を持つた蚊の五浪くんは民家に入ってしまった。するとそこは作民の人間が殺虫剤を大量に吹きかけてきた。蚊の体力では超能力の回数も限られており、何度も殺虫剤をあひれば死んでしまう。そのため五浪くんは、民家を出て生きるために全力で、逃げ続けるのである!!

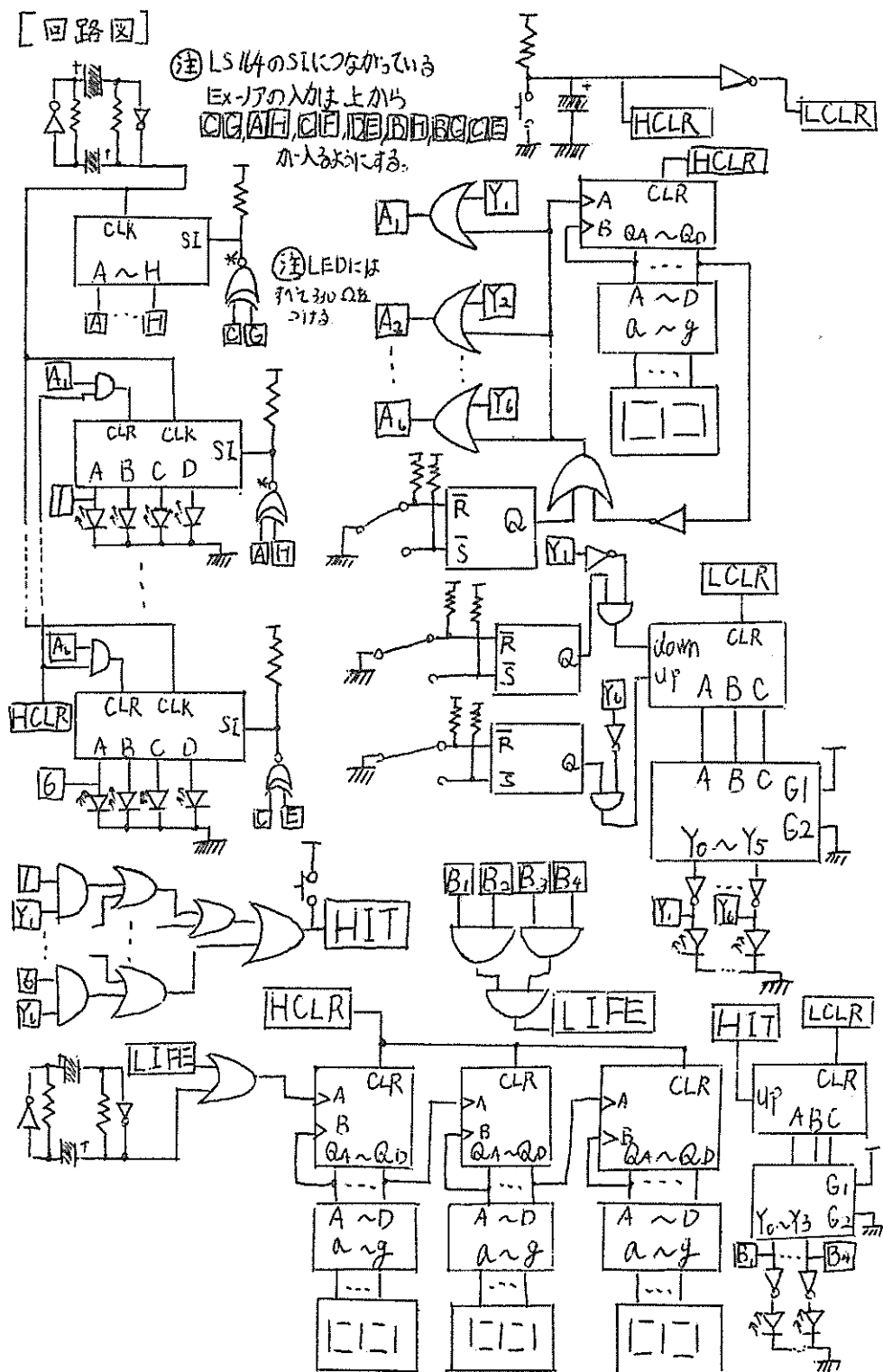
[外見図]



[ブロック図]



[回路図]





製作者 M2 中野

回路図設計者 H2

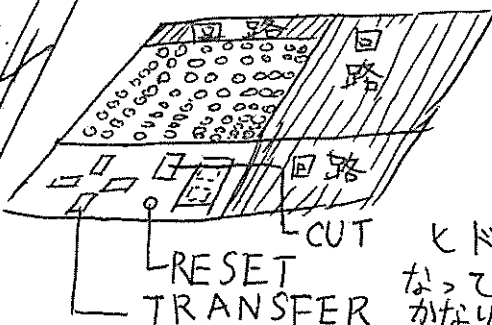
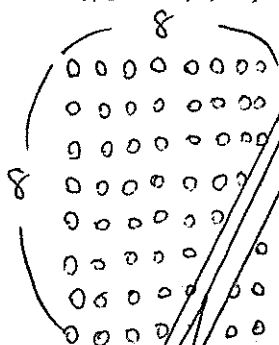
協力者 無線班の皆様 舟渡様

上でマスとりすきましたね。これはヒドいです。

① 内容

おそい来る落武者を片っ端から斬る! それだけです。
落武者は単細胞で一次元生物なんで一直線に向かっ
てきます。よけることはもちろん、回転斬りも可能!!!

② 外観図(理想)

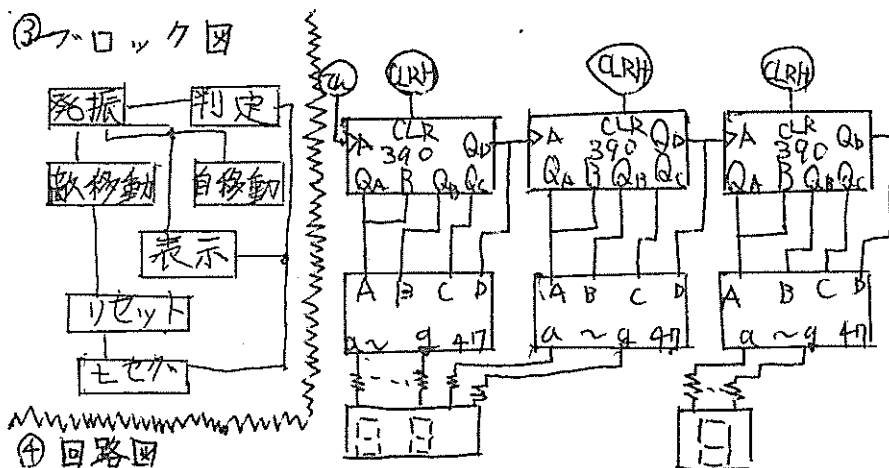


こんな風に
出来ると
いいんだけ
どなあ...
配線材が

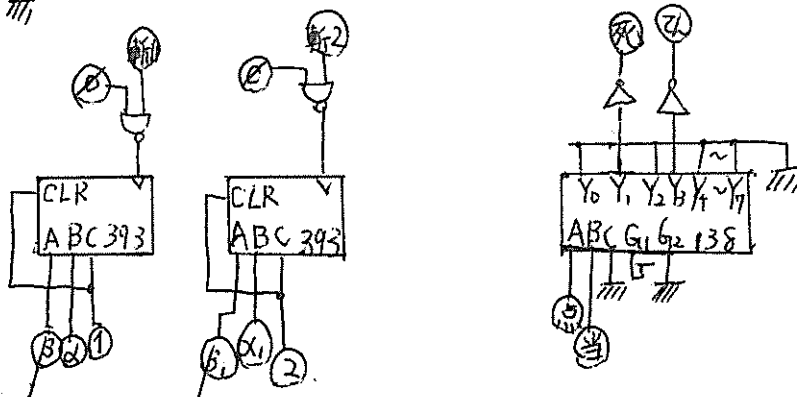
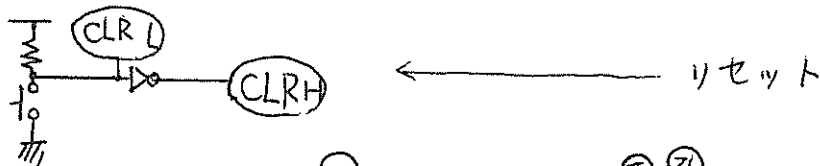
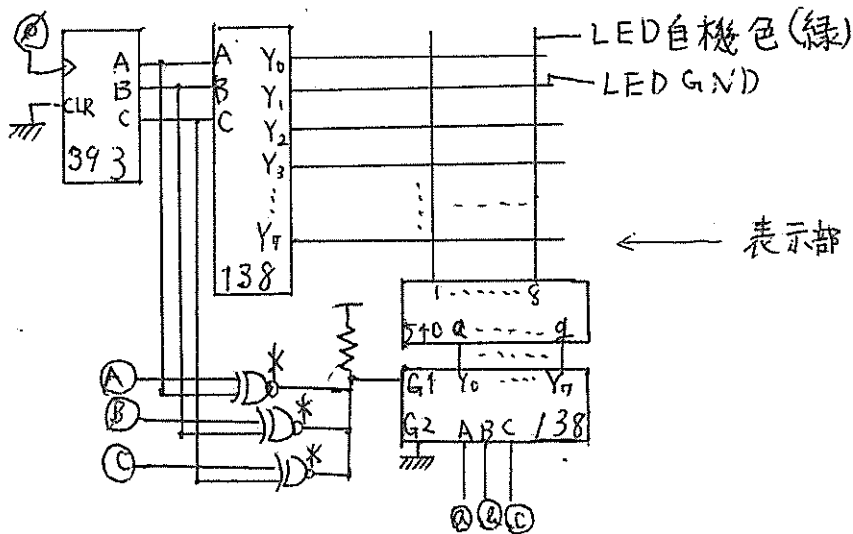
ヒドいことに
なっているよう
でかなりこの図と異

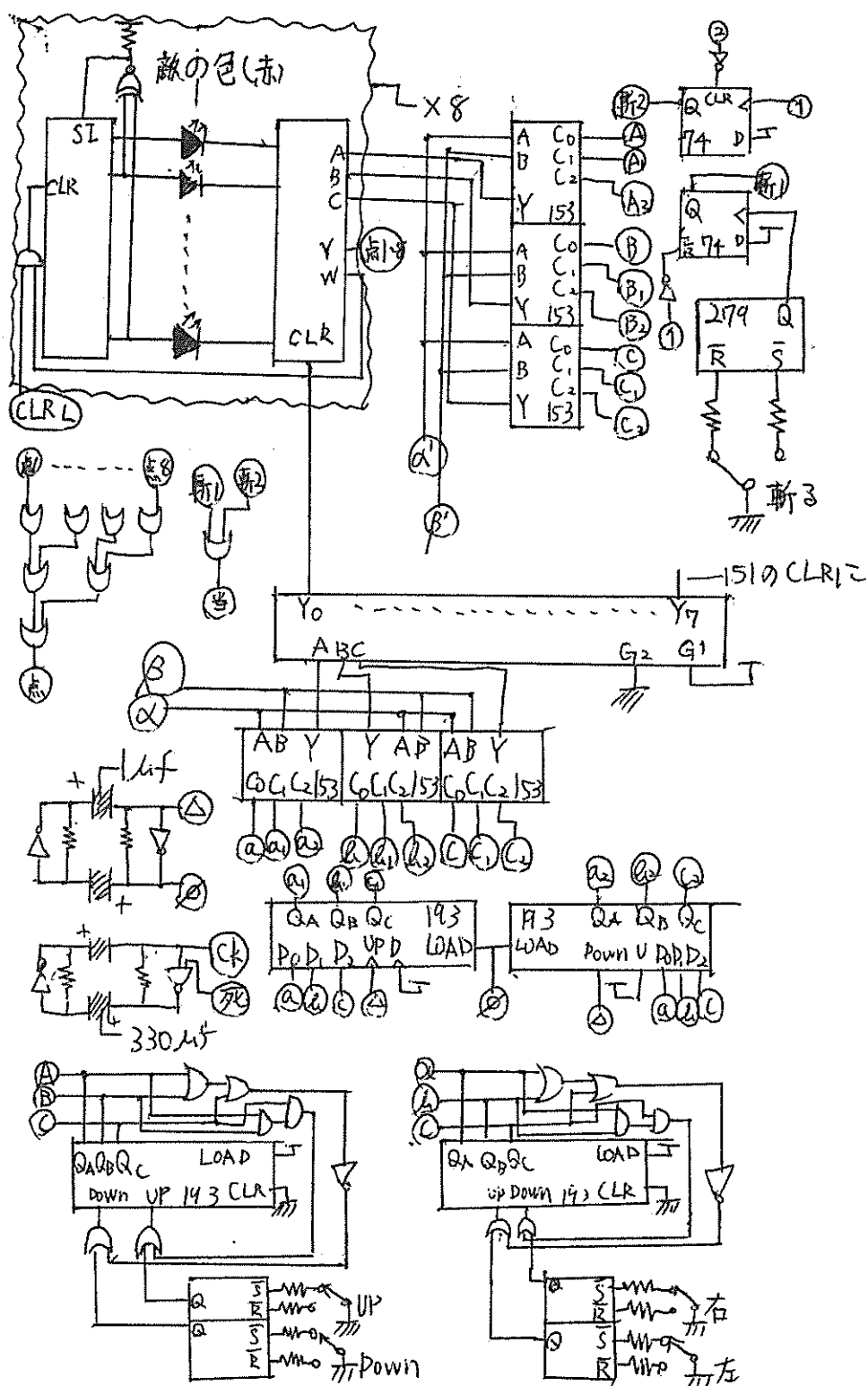
なってしまう感じがします。3月23日
現在、動いていません。絶対文化祭までに間
に合わせて表面をキレイにしたいです。

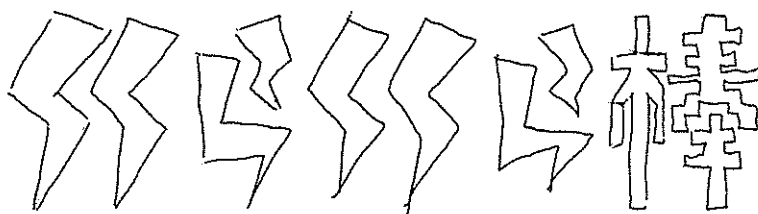
③ フロックス図



④ 回路図





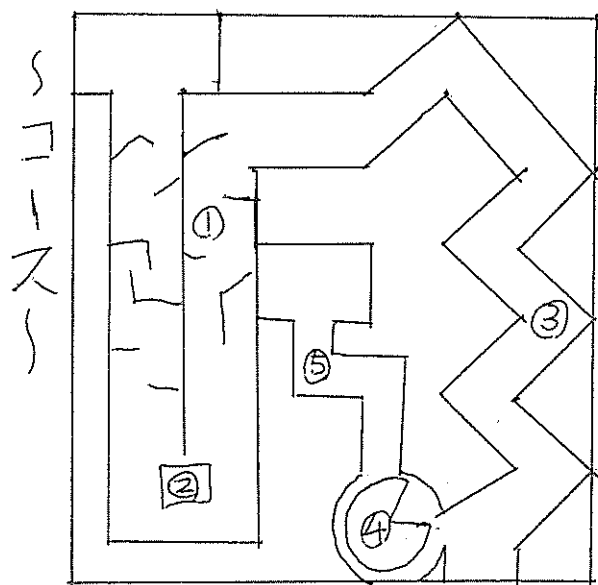


製作・回路
M2 雅人
大塚 祐士
佐野 祐士

協力
物理部無線班
の方々

PART I ~カントー編~

~ストーリー~
ポ○モンです。4チャンセオン目指してがんばれ。



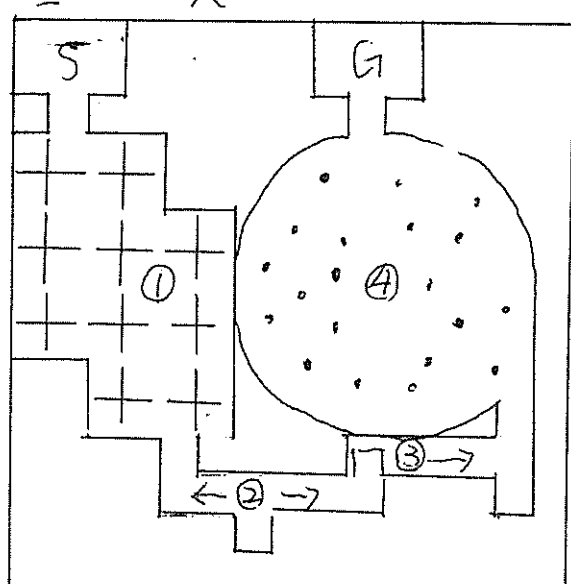
- ① ト○ワの森...方程式より簡単。迷えない。
- ② 往復ビンタ...当たると目とコースから火花が。
- ③ ピカ○ユウのしっぽ...十しより簡単。
- ④ ヒト○マン...ぐるぐるぐるぐるぐるぐるぐるぐるぐる。
- ⑤ チャンピオンロード...王の道。抜ければ君は
チャンピオン。

制限時間は100秒

PART II ~ジョウトーホウエン編~

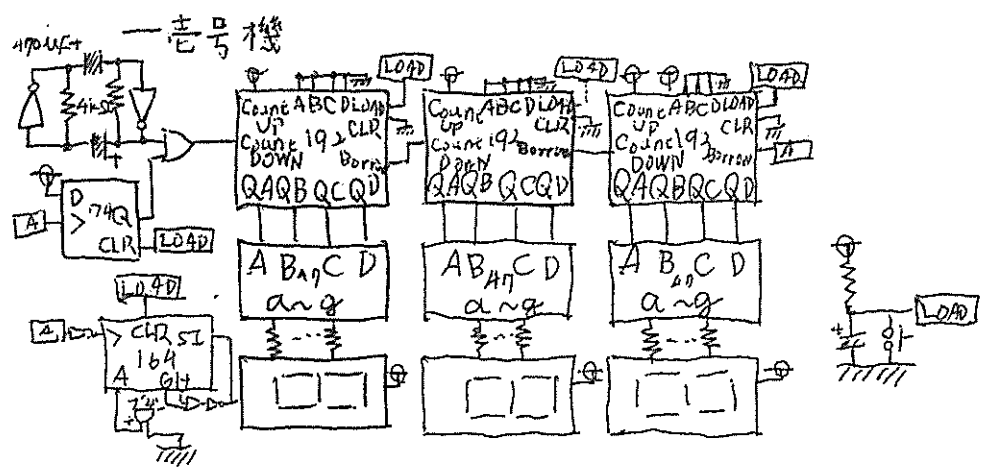
~局解説~
PART Iが小さくなって帰ってきた。
難易度が高い可能性があるかもしれないらしい、ぽい。
やってみれば分かるかも。

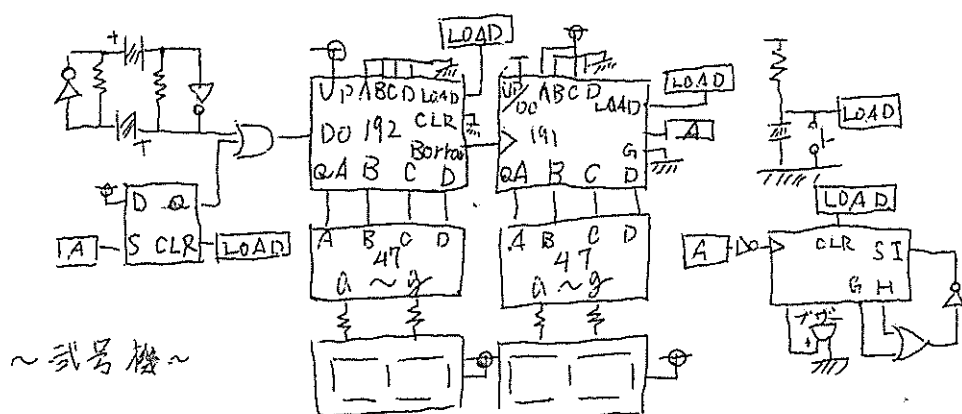
～ コース ～



- ① うずしお... 触わる所を間違えると、ばちっ。
 - ② イワツブテ... ぶされる。
 - ③ タマ○シのエレベーター... 何故かエレベーターに
高圧電流が。
 - ④ チャンピオンロードⅡ... 次々と来る敵をかかせ。
- 制限時間は50秒。

回路図 (4/21現在)





※この通りに作っても
動かない場合があります。

～コースの説明～

- ・まわりのわくはVPI3の塩ビパイプを使用。
- ・コースは8φ厚さ1mmのアルミパイプと、曲げた6φのアルミ丸棒を使用。



- ・スライド部分は、スライドワランワ機槽を使用。
- ・モーターは遊星ギヤボックス。
- ・回転部分は、切り替えスイッチを使って「回転」と「逆回転」する。

・火花がちらるということについて

インスタントカメラの中に入っている、「フラッシュ基板」を使う。充電をすると、電圧がコンデンサーにたまり、この二本の足をショートさせると火花がちらるのである。

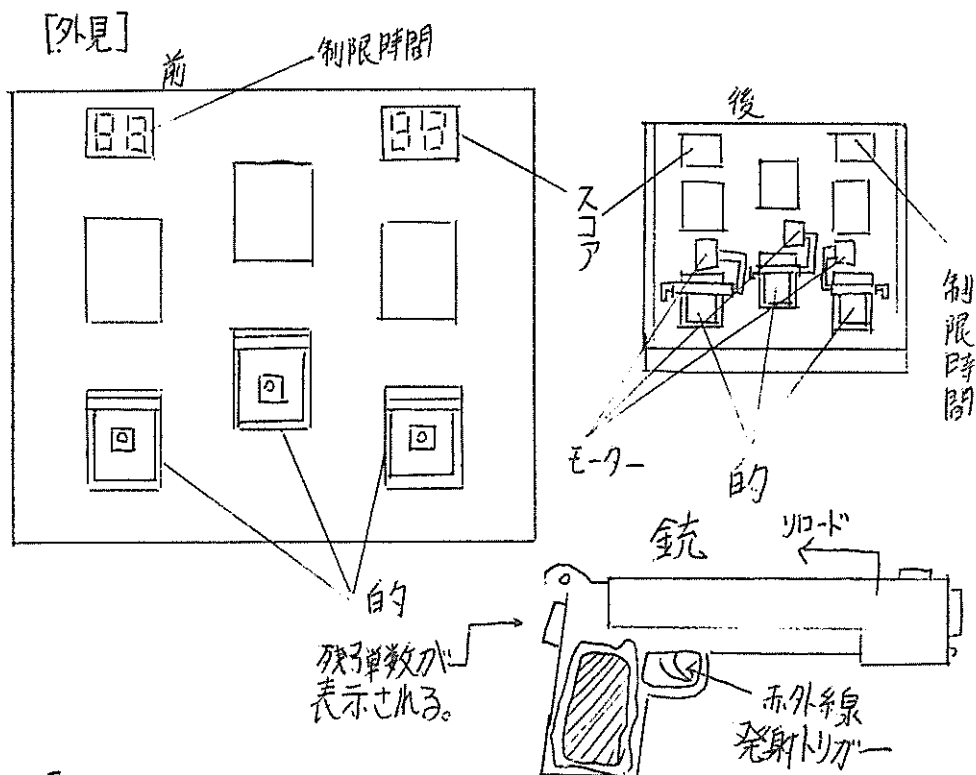
※インスタントカメラを分解するときは、

感電に注意しましょう。

下鎗@鎗

製作者 M2 木村 川崎 中嶋
協力者 物無の皆さん

これは赤外線を使った射的ゲームで
鎗はテレビのリモコンのようなものと思ってください
この製作物についてどうして中2ばかり集まって
作っているのか？題名に関することで語りたと思います。
まず中2ばかり集まっているのは第一製作
物が完動した当時中1が第二製作物を作る時に大
きい物を作るには同学年で集まらざるおえなかったからです。
しかし、そのおかげで上の学年で習う、PILのプログラミングの事な
ども学べて僕(木村)にとっては大変有意義だったと思
います。次に題名の事でか「これは本体を作っている
最中万がとめる時、傷を付けないために使った某
マンガのセリフ「先生/嘘だよな先生」によるもの
で、この製作物とはあんまり関係ないと言っちゃ
関係ないんですけどね。こんなくだらない話を最後まで
読んでいただいて有り難うございました。

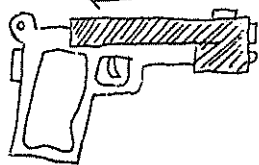


[ルール]

赤外線を発射する銃を使い、制限時間内に、多くの赤外線を的にあてるゲームです。

的に赤外線が当たると、裏についているモーターで、的が下にある場合は上へ、上にある場合は下へ移動します。

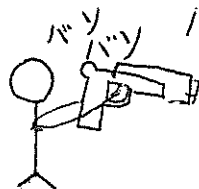
また、銃はトリガーを引くと赤外線が発射されます。弾数(赤外線を発射できる回数)は無限ですが、7発うつごとにリロード(下図の斜線部分を矢印の方向に引く)しなければ、赤外線を発射することはできません。



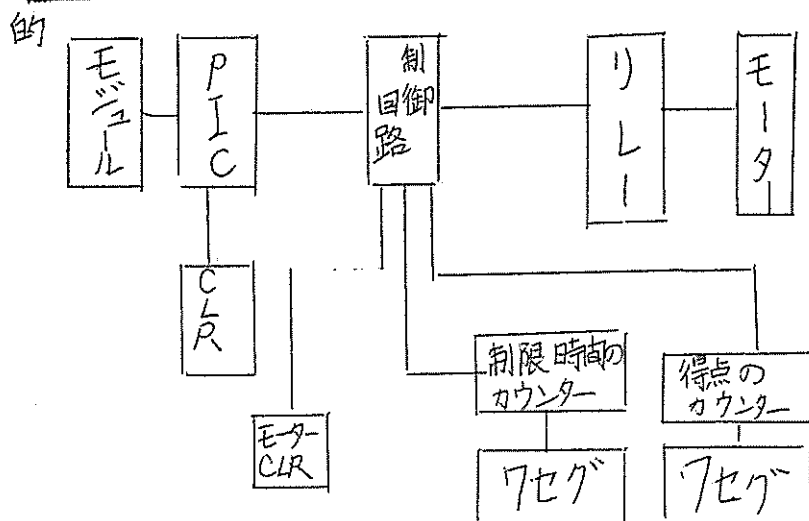
また、一回赤外線をうつと、少しの間、残り弾数を表示するワセグが点滅します。

この間はトリガーを引いても、赤外線は発射されません。

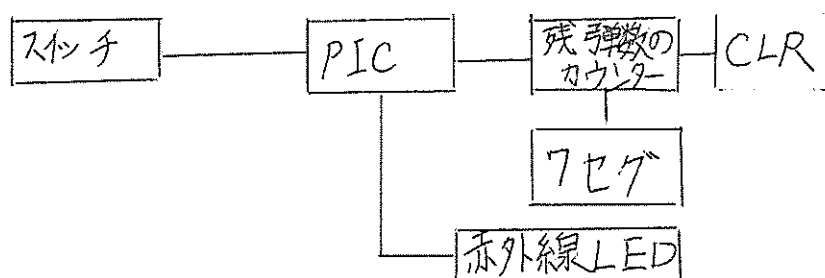
以上のことに注意して、的をバンバンうつして下さい。



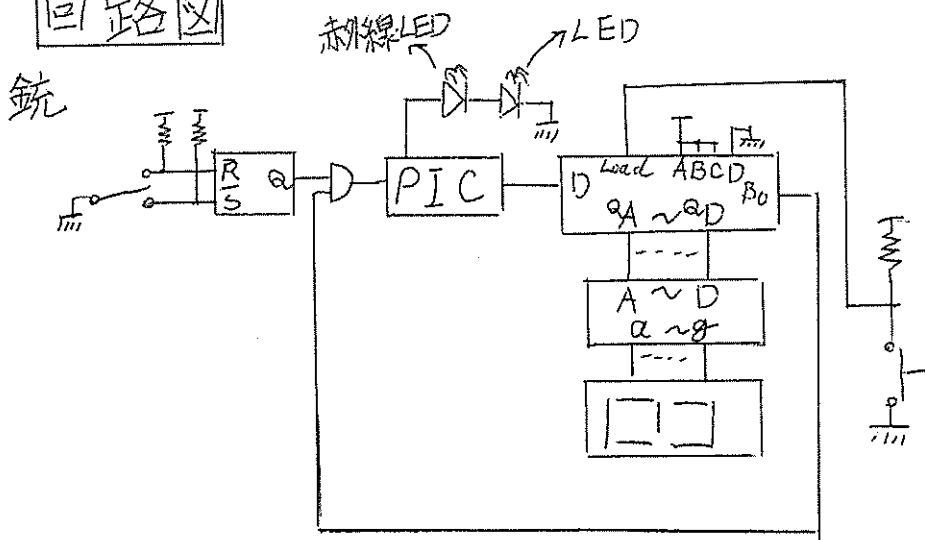
ブロック図

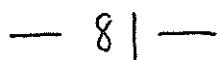


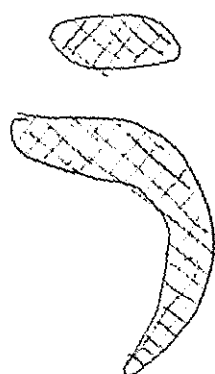
疏



回路図







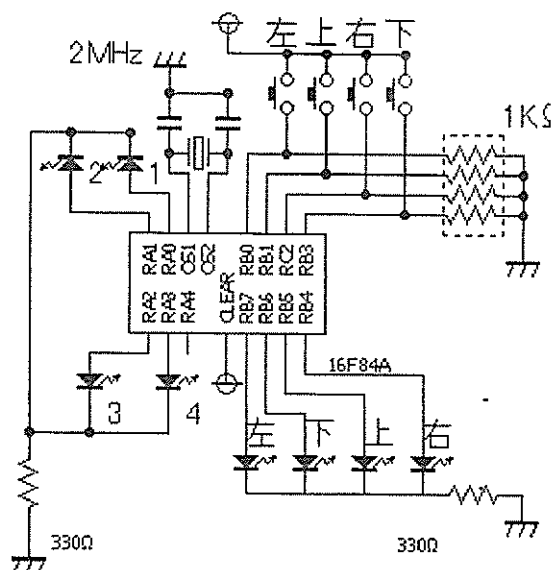
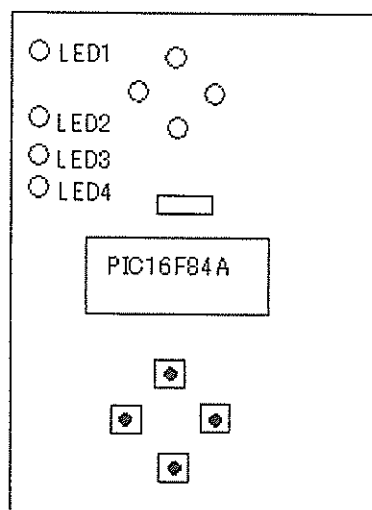
the 反射

プログラム作成 : M3 井上 忠明
回路設計 : M2 竹島 啓純
原案 : H2 舟渡 隆平氏

～ルール説明～

このゲームはプレイヤーの反射を試すゲームです。外見図を見れば分かる通り、十字型に LED とスイッチが配置されています。このうち、光った LED と逆の方向のスイッチを押せば成功、同じ方向、又は関係のない方向のスイッチを押したり、制限時間を越えてしまうとゲームオーバーです。レベルは 4 段階あり、1 つのレベルを 4 連続で成功することによって次のレベルに進めます。また、上のレベルにあがっていったところで失敗すると、またレベル 1 からやり直しになります。

～外観図&回路図～



～プログラム～

:売り物用プログラム(反射神経
ゲーム)

LIST P=P16F84A
INCLUDE P16F84A.INC
_CONFIG_XT_OSC & _WDT_OFF &
_PWRTE_ON & _CP_OFF

:レジスタ設定

KAR1	EQU	OCH
RANSU	EQU	ODH
TIME1	EQU	OEH
TIME2	EQU	OFH
TIME3	EQU	10H
LVL1	EQU	11H
LVL2	EQU	12H

:セットアップ

SETUP

BSF	STATUS, RPO
CLRF	TRISA
MOVLW	OFH
MOVWF	TRISB
BCF	STATUS, RPO
CLRF	PORTB
CLRF	PORTA
CLRF	LVL1
CLRF	LVL2

:乱数設定

RANSU1 :EEPROM読み出し

BCF	STATUS, RPO
MOVLW	00H
MOVWF	EEADR
BSF	STATUS, RPO
BSF	EECON1, RD
BCF	STATUS, RPO
MOVF	EEDATA, W
MOVWF	KAR1

RANSU2 :xnor1

BTFS	KAR1, 5
GOTO	RANSU3
GOTO	RANSU4

RANSU3 :xnor2

BTFS	KAR1, 6
GOTO	RANSU5
GOTO	RANSU6

RANSU4 :xnor3

BTFS	KAR1, 6
GOTO	RANSU7
GOTO	RANSU8

RANSU5 :xnor4

BSF	STATUS, 0
-----	-----------

RANSU6

RLF	KAR1, W
MOVWF	RANSU
GOTO	RANSU10

RANSU7 :xnor6

BCF	STATUS, 0
RLF	KAR1, W
MOVWF	RANSU
GOTO	RANSU10

RANSU8 :xnor7 (FFcheck)

BCF	STATUS, 0
RLF	KAR1, W
MOVWF	RANSU
GOTO	RANSU10

RANSU9 :xnor8

BTFS	KAR1, 0
GOTO	RANSU9
BTFS	KAR1, 1
GOTO	RANSU9

RANSU10

BTFS	KAR1, 2
GOTO	RANSU9
BTFS	KAR1, 3
GOTO	RANSU9

RANSU11 :EEPROM書き込み2

BTFS	KAR1, 4
GOTO	RANSU9
BTFS	KAR1, 7
GOTO	RANSU9

RANSU12 :LED点灯1

GOTO	STATUS, 0
BCF	KAR1, W
RLF	KAR1, W
MOVWF	RANSU
GOTO	RANSU10

RANSU13 :LED点灯2

BSF	STATUS, 0
RLF	KAR1, W
MOVWF	RANSU
GOTO	RANSU10

RANSU14 :LED点灯3

BCF	STATUS, RPO
MOVLW	00H
MOVWF	EEADR
MOVF	RANSU, W
MOVWF	EEDATA

RANSU15 :LED点灯4

BSF	STATUS, RPO
BSF	EECON1, WREN
MOVLW	55H
MOVWF	EECON2
MOVLW	0AAH
MOVWF	EECON2
BSF	EECON1, WR

RANSU16 :LED点灯5

BTFS	EECON1, WR
BTFS	EECON1, WR
GOTO	RANSU11
BCF	STATUS, RPO

RANSU17 :LED点灯6

BTFS	EECON1, WR
BTFS	EECON1, WR
GOTO	RANSU11
BCF	STATUS, RPO

RANSU18 :LED点灯7

BTFS	EECON1, WR
BTFS	EECON1, WR
GOTO	RANSU11
BCF	STATUS, RPO

RANSU19 :LED点灯8

BTFS	EECON1, WR
BTFS	EECON1, WR
GOTO	RANSU11
BCF	STATUS, RPO

RANSU20 :LED点灯9

BTFS	EECON1, WR
BTFS	EECON1, WR
GOTO	RANSU11
BCF	STATUS, RPO

RANSU21 :LED点灯10

BTFS	EECON1, WR
BTFS	EECON1, WR
GOTO	RANSU11
BCF	STATUS, RPO

RANSU22 :LED点灯11

BTFS	EECON1, WR
BTFS	EECON1, WR
GOTO	RANSU11
BCF	STATUS, RPO

RANSU16 :LED点灯5

BCF	PORTB, 7
CALL	SELECT1
GOTO	HANTE11

RANSU17 :LED点灯6

BCF	PORTB, 4
BSF	PORTB, 5
BCF	PORTB, 6
BCF	PORTB, 7
CALL	SELECT1
GOTO	HANTE12

RANSU18 :LED点灯7

BCF	PORTB, 4
BSF	PORTB, 5
BCF	PORTB, 6
BCF	PORTB, 7
CALL	SELECT1
GOTO	HANTE13

RANSU19 :LED点灯8

BCF	PORTB, 4
BSF	PORTB, 5
BCF	PORTB, 6
BSF	PORTB, 7
CALL	SELECT1
GOTO	HANTE14

RANSU20 :LED点灯9

BCF	PORTB, 4
BSF	PORTB, 5
BCF	PORTB, 6
BSF	PORTB, 7
CALL	SELECT1
GOTO	HANTE14

RANSU21 :LED点灯10

BCF	PORTB, 4
BSF	PORTB, 5
BCF	PORTB, 6
BSF	PORTB, 7
CALL	SELECT1
GOTO	HANTE14

RANSU22 :LED点灯11

BCF	PORTB, 4
BSF	PORTB, 5
BCF	PORTB, 6
BSF	PORTB, 7
CALL	SELECT1
GOTO	HANTE14

HANTE11

BTFS	PORTB, 0
GOTO	CLEAR
BTFS	PORTB, 1
GOTO	BAD
BTFS	PORTB, 2
GOTO	BAD
BTFS	PORTB, 3
GOTO	BAD
GOTO	DECFSZ TIME1
GOTO	HANTE11
MOVWF	TIME1
DECFSZ	TIME2
GOTO	HANTE11
MOVWF	TIME2
DECFSZ	TIME3
GOTO	HANTE11
GOTO	BAD

HANTE12

BTFS	PORTB, 0
GOTO	BAD
BTFS	PORTB, 1
GOTO	BAD
BTFS	PORTB, 2
GOTO	BAD
BTFS	PORTB, 3
GOTO	CLEAR
GOTO	DECFSZ TIME1
GOTO	HANTE12
MOVWF	TIME1
DECFSZ	TIME2
GOTO	HANTE12
MOVWF	TIME2
DECFSZ	TIME3
GOTO	HANTE12
GOTO	BAD

HANTE13

BTFS	PORTB, 0
GOTO	BAD
BTFS	PORTB, 1
GOTO	BAD
BTFS	PORTB, 2
GOTO	BAD
BTFS	PORTB, 3
GOTO	CLEAR
GOTO	DECFSZ TIME1
GOTO	HANTE13
MOVWF	TIME1
DECFSZ	TIME2
GOTO	HANTE13
MOVWF	TIME2
DECFSZ	TIME3
GOTO	HANTE13
GOTO	BAD

HANTE14

BTFS	PORTB, 0
GOTO	BAD
BTFS	PORTB, 1
GOTO	BAD
BTFS	PORTB, 2
GOTO	BAD
BTFS	PORTB, 3
GOTO	CLEAR
GOTO	DECFSZ TIME1
GOTO	HANTE14
MOVWF	TIME1
DECFSZ	TIME2
GOTO	HANTE14
MOVWF	TIME2
DECFSZ	TIME3
GOTO	HANTE14
GOTO	BAD

HANTE15

BTFS	PORTB, 0
GOTO	BAD
BTFS	PORTB, 1
GOTO	BAD
BTFS	PORTB, 2
GOTO	BAD
BTFS	PORTB, 3
GOTO	CLEAR
GOTO	DECFSZ TIME1
GOTO	HANTE15
MOVWF	TIME1
DECFSZ	TIME2
GOTO	HANTE15
MOVWF	TIME2
DECFSZ	TIME3
GOTO	HANTE15
GOTO	BAD

HANTE16

BTFS	PORTB, 0
GOTO	BAD
BTFS	PORTB, 1
GOTO	BAD
BTFS	PORTB, 2
GOTO	BAD
BTFS	PORTB, 3
GOTO	CLEAR
GOTO	DECFSZ TIME1
GOTO	HANTE16
MOVWF	TIME1
DECFSZ	TIME2
GOTO	HANTE16
MOVWF	TIME2
DECFSZ	TIME3
GOTO	HANTE16
GOTO	BAD

HANTE17

BTFS	PORTB, 0
GOTO	BAD
BTFS	PORTB, 1
GOTO	BAD
BTFS	PORTB, 2
GOTO	BAD
BTFS	PORTB, 3
GOTO	CLEAR
GOTO	DECFSZ TIME1
GOTO	HANTE17
MOVWF	TIME1
DECFSZ	TIME2
GOTO	HANTE17
MOVWF	TIME2
DECFSZ	TIME3
GOTO	HANTE17
GOTO	BAD

HANTE18

BTFS	PORTB, 0
GOTO	BAD
BTFS	PORTB, 1
GOTO	BAD
BTFS	PORTB, 2
GOTO	BAD
BTFS	PORTB, 3
GOTO	CLEAR
GOTO	DECFSZ TIME1
GOTO	HANTE18
MOVWF	TIME1
DECFSZ	TIME2
GOTO	HANTE18
MOVWF	TIME2
DECFSZ	TIME3
GOTO	HANTE18
GOTO	BAD

HANTE19

BTFS	PORTB, 0
GOTO	BAD
BTFS	PORTB, 1
GOTO	BAD
BTFS	PORTB, 2
GOTO	BAD
BTFS	PORTB, 3
GOTO	CLEAR
GOTO	DECFSZ TIME1
GOTO	HANTE19
MOVWF	TIME1
DECFSZ	TIME2
GOTO	HANTE19
MOVWF	TIME2
DECFSZ	TIME3
GOTO	HANTE19
GOTO	BAD

HANTE20

BTFS	PORTB, 0
GOTO	BAD
BTFS	PORTB, 1
GOTO	BAD
BTFS	PORTB, 2
GOTO	BAD
BTFS	PORTB, 3
GOTO	CLEAR
GOTO	DECFSZ TIME1
GOTO	HANTE20
MOVWF	TIME1
DECFSZ	TIME2
GOTO	HANTE20
MOVWF	TIME2
DECFSZ	TIME3
GOTO	HANTE20
GOTO	BAD

```

BTFSS PORTB, 1
GOTO CLEAR
BTFSS PORTB, 2
GOTO BAD
BTFSS PORTB, 3
GOTO BAD
DECFSZ TIME1
GOTO HANTE13
MOVWF TIME1
DECFSZ TIME2
GOTO HANTE13
MOVWF TIME2
DECFSZ TIME3
GOTO HANTE13
GOTO BAD

HANTE14
BTFSS PORTB, 0
GOTO BAD
BTFSS PORTB, 1
GOTO BAD
BTFSS PORTB, 2
GOTO CLEAR
BTFSS PORTB, 3
GOTO BAD
DECFSZ TIME1
GOTO HANTE14
MOVWF TIME1
DECFSZ TIME2
GOTO HANTE14
MOVWF TIME2
DECFSZ TIME3
GOTO HANTE14
GOTO BAD

;*****
;レベルセレクト
;*****

SELECT1
BTFSS LVL1, 0
GOTO SELECT2
GOTO SELECT3

SELECT2
BTFSS LVL1, 1
GOTO SELECT4
GOTO SELECT6

SELECT3
BTFSS LVL1, 1
GOTO SELECT5
GOTO SELECT7

SELECT4 :レベル1
BCF PORTA, 1
BCF PORTA, 2
BCF PORTA, 3
MOVLW 03H
MOVWF TIME3
MOVLW OFFH
MOVWF TIME1
MOVWF TIME2
BTFSS LVL2, 0
RETURN
BTFSS LVL2, 1
RETURN
CLRF LVL2
INCF LVL1
RETURN
:レベル2
BSF PORTA, 1

SELECT5
BTFSS PORTA, 2
GOTO 02H
MOVWF TIME3
MOVLW 0EEH
MOVWF TIME1
MOVWF TIME2
BTFSS LVL2, 0
RETURN
BTFSS LVL2, 1
RETURN
CLRF LVL2
INCF LVL1
RETURN

SELECT6 :レベル3
BCF PORTA, 1
BSF PORTA, 2
BCF PORTA, 3
MOVLW 01H
MOVWF TIME3
MOVLW 00DH
MOVWF TIME1
MOVWF TIME2
BTFSS LVL2, 0
RETURN
BTFSS LVL2, 1
RETURN
CLRF LVL2
INCF LVL1
RETURN

SELECT7 :レベル4
BCF PORTA, 1
BCF PORTA, 2
BSF PORTA, 3
MOVLW 01H
MOVWF TIME3
MOVLW 0BBH
MOVWF TIME1
MOVWF TIME2
BTFSS LVL2, 0
RETURN
BTFSS LVL2, 1
RETURN
CLRF LVL1
CLRF LVL2
RETURN

;*****
;終了 (成功・失敗)
;*****

CLEAR :成功
BCF PORTB, 4
BCF PORTB, 5
BCF PORTB, 6
BCF PORTB, 7
BSF PORTA, 0
CALL WAIT
BCF PORTA, 0
INCF LVL2
CALL WAIT_SW
GOTO WAIT_RAND

BAD :失敗
BSF PORTB, 4
BSF PORTB, 5
BSF PORTB, 6
BSF PORTB, 7
CALL WAIT_SW

CALL WAIT_SW2
BCF PORTB, 4
BCF PORTB, 5
BCF PORTB, 6
BCF PORTB, 7
CLRF LVL1
CLRF LVL2
CALL WAIT_SW
GOTO WAIT_RAND

BAD_TIME
BSF PORTB, 4
BSF PORTB, 5
BSF PORTB, 6
BSF PORTB, 7
CALL WAIT_SW2
BCF PORTB, 4
BCF PORTB, 5
BCF PORTB, 6
BCF PORTB, 7
CLRF LVL1
CLRF LVL2
CALL WAIT_SW
GOTO WAIT_RAND

;*****
;ウェイト
;*****

WAIT
MOVLW OFFH
MOVWF TIME1
MOVWF TIME2

WAIT2
DECFSZ TIME1
GOTO WAIT2
MOVWF TIME1
DECFSZ TIME2
GOTO WAIT2
RETURN

WAIT_SW
BTFSS PORTB, 0
GOTO WAIT_SW
BTFSS PORTB, 1
GOTO WAIT_SW
BTFSS PORTB, 2
GOTO WAIT_SW
BTFSS PORTB, 3
GOTO WAIT_SW
RETURN

WAIT_SW2
BTFSS PORTB, 0
RETURN
BTFSS PORTB, 1
RETURN
BTFSS PORTB, 2
RETURN
BTFSS PORTB, 3
RETURN

WAIT_RAND
BCF STATUS, RPO
MOVLW 00H
MOVWF EEADR
BSF STATUS, RPO
BSF EECON1, RD
BCF STATUS, RPO
MOVF EEDATA, W

```

	MOVWF	KAR1	DECFSZ	TIME1
WAIT_RANDOM_2	BTFSS	KAR1_2	GOTO	WAIT_RANDOM_2_2
	GOTO	WAIT_RANDOM_3	MOVWF	TIME1
	GOTO	WAIT_RANDOM_4	DECFSZ	TIME2
WAIT_RANDOM_3	BTFSS	KAR1_4	GOTO	WAIT_RANDOM_2_2
	GOTO	WAIT_RANDOM1	MOVWF	TIME2
	GOTO	WAIT_RANDOM2	DECFSZ	TIME3
WAIT_RANDOM_4	BTFSS	KAR1_4	GOTO	WAIT_RANDOM_2_2
	GOTO	WAIT_RANDOM3	GOTO	RANSU1
	GOTO	WAIT_RANDOM4		
WAIT_RANDOM1	MOVLW	03H		
	MOVWF	TIME3		
	MOVLW	0DDH		
	MOVWF	TIME1		
	MOVWF	TIME2		
WAIT_RANDOM1_2	DECFSZ	TIME1		
	GOTO	WAIT_RANDOM1_2		
	MOVWF	TIME1		
	DECFSZ	TIME2		
	GOTO	WAIT_RANDOM1_2		
	MOVWF	TIME2		
	DECFSZ	TIME3		
	GOTO	WAIT_RANDOM1_2		
	GOTO	RANSU1		
WAIT_RANDOM2	MOVLW	02H		
	MOVWF	TIME3		
	MOVLW	0B5H		
	MOVWF	TIME1		
	MOVWF	TIME2		
WAIT_RANDOM2_2	DECFSZ	TIME1		
	GOTO	WAIT_RANDOM2_2		
	MOVWF	TIME1		
	DECFSZ	TIME2		
	GOTO	WAIT_RANDOM2_2		
	MOVWF	TIME2		
	DECFSZ	TIME3		
	GOTO	WAIT_RANDOM2_2		
	GOTO	RANSU1		
WAIT_RANDOM3	MOVLW	02H		
	MOVWF	TIME3		
	MOVLW	0CCH		
	MOVWF	TIME1		
	MOVWF	TIME2		
WAIT_RANDOM3_2	DECFSZ	TIME1		
	GOTO	WAIT_RANDOM3_2		
	MOVWF	TIME1		
	DECFSZ	TIME2		
	GOTO	WAIT_RANDOM3_2		
	MOVWF	TIME2		
	DECFSZ	TIME3		
	GOTO	WAIT_RANDOM3_2		
	GOTO	RANSU1		
WAIT_RANDOM4	MOVLW	01H		
	MOVWF	TIME3		
	MOVLW	0FFH		
	MOVWF	TIME1		
	MOVWF	TIME2		
WAIT_RANDOM4_2				

END