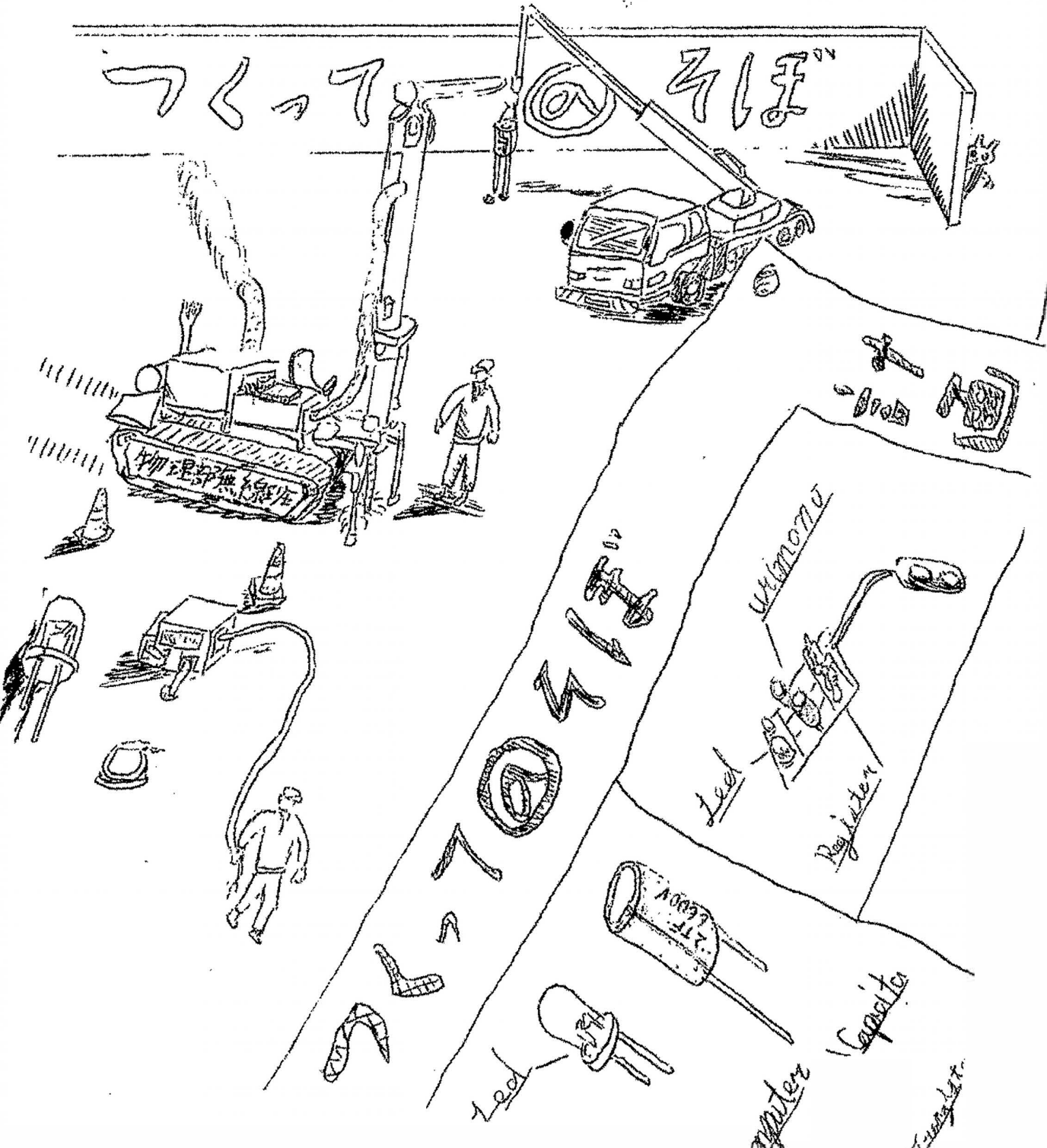


路口图集

2013





はじめに

昨年の回路図集では「よりプロセスルールを微細化したCPUが発表される」とありますが、限界の見え始めた平面上の微細化に加えて三次元の集積技術も活発に研究されています。

我が部でもARMによるOS設計や、32bitマイコンによる並立動作を行うなど時代に追いつこうと日進月歩しています。

この「回路図集」は麻布学園物理部無線班の部員が1年をかけて研究・製作したものについての詳細が記されているものです。唯し、未完成のものも含まれており、必ずしも本書の通りに作っても上手く動作する保障はありません。予め御了承下さい。

H2 会計 水野 太郎

はじめに 01

目次 02

コンピュータ系 04

A-tab3 05

SOUNDPAD64ver.3 11

ロボット系 14

ATTENTION PLEASE!!! 15

食べ砲台 20

Alto-2 25

>STRIDER-"0"WD 30

Three-way standoff 36

Ro.ver 39

WALKING!! 44

H1 佐藤 博

H2 岸田 聖生

H2 岸田 聖生

M3 石倉 匠

H2 水野 太郎

M3 宮嶋 優大

M3 山本 涼太郎

H2 河村 洋一郎

H1 花園 佳月

M3 市川 詠亮

M3 佐伯 大地

H1 黒田 健太

M3 横尾 幸丸

H1 四柳 雄太

M3 渡辺 悠太

アナログ系	46	
Catch the WAVE	47	H2 岸田 聖生
ゲーム系	49	
GAME TUBE	50	H2 山本 涼一
My home wars	56	M2 磯村 隆正
超跳躍役	59	M2 伊藤 颯
猛獣へのえさ投げ	63	M2 川本 博之助
激突!グルメレース	66	M2 佐藤 誓斗
ピアノの達人	68	M2 関根 史人
SKY WALKER	71	M2 中村 航
ジグザグマを追え	74	M2 福島 弘也
メテオ襲来	77	M2 和田 直輝
イライラ棒	80	M2 関根 史人 M2 中村 航
売り物系	82	
チカチカ	83	M3 宮嶋 優大
SPACE MATE	84	M3 佐伯 大地

COMPUTER

	36	37	38	39	3A	3B	3C	3D
CD	empty	empty	empty	1	1	0	1	1
CE	1	1	0				1	1
CF	0	1	1				0	1
DD	1	1		1	0	1	1	1
	1	1	1	1	1	1	0	1
	1	1	1					

The image features a grid of binary data (0s and 1s) overlaid on a construction site illustration. The grid has columns labeled 36 through 3D and rows labeled CD, CE, CF, and DD. The data values are as follows:

	36	37	38	39	3A	3B	3C	3D
CD	empty	empty	empty	1	1	0	1	1
CE	1	1	0				1	1
CF	0	1	1				0	1
DD	1	1		1	0	1	1	1
	1	1	1	1	1	1	0	1
	1	1	1					

The construction site includes a crane, a concrete mixer truck, and a small cartoon rabbit character.

A-tab 3.2

制作 H1 佐藤 惇
協力 物無の皆様

◆ これはなに？

一言でいえば多機能プレイヤーです。

PDA、というかスマートフォンを目指して作成したデバイスです。まあバッテリーを積む予定も無線機能も付けるつもりはないのでただのパームトッププレイヤーですね。ゲームや音楽再生、動画ファイルの再生、3Dポリゴンモデルの展開を目指しています。

外観は現状本体絶賛再制作中なので省略ということで。スマートフォンのような基本の形態+拡張で取り外し式のハードウェアキーとサブモニタを取り付けることを目標にしています。

現状ゲーム三種(パックオン、テトオス、インベ〇ダー)は完成。動画再生と3Dポリゴン展開用のプログラムはベータ版が完成済みです。

当初STM32F4Discoveryという開発用のボードを使おうと思っていたのですが、無駄に周辺回路が充実しているせいで思うように機能が使えず、Digi-Keyで輸入したSTM32F407V/ZGT6というARMCPUで基盤を組みなおしているところです。

これらARMが中心となって液晶、SD、電源、周辺ICを制御することで上の機能を実現可能とします。

またもう一つの特徴として簡易OSの搭載があります。

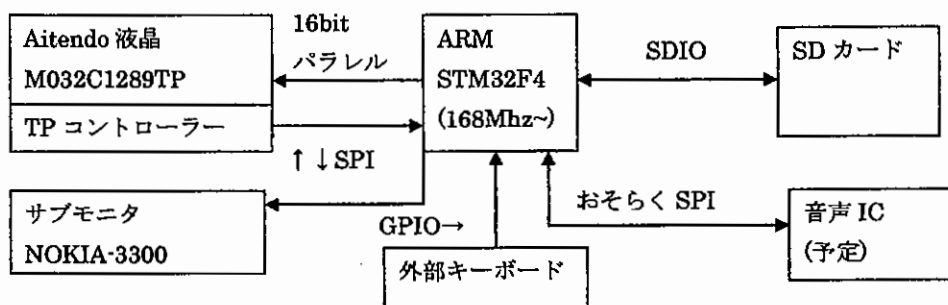
RTOSなどのフリーOSもありましたが今回はタスク管理がそこまでシビアでなく、かつハードウェアの性能を殺さないことが求められるため、ハードウェアとソフトウェアの関係が疎になることを防ぐため、セマフォなどの機能を一部排除したあくまで簡易的なOSとして複数タスクを管理するソフトウェアの設計を行いました。

時間管理タスク	タスク管理タスク	UI	上位ソフトウェア1	上位ソフトウェア2
液晶制御	電源管理	SD制御	その他	
ハードウェア				

下位レイヤが上位レイヤにハードウェアから提供される機能の中継ぎをします。上位レイヤは割り当てられた CPU 作業時間の中で下位レイヤとやり取りし、動画再生やゲームなどの機能を実現します。

ハードウェア

上に記したように ARM CPU である STM32F4 を中心としています。



構成は恐ろしくシンプルです。

ARM が SD カードからデータを読み出し、メモリ操作を経て液晶に情報を表示します。ユーザーからの操作があれば TP コントローラーから取得し、それに応じた読み込みや出力を行います。

STM32F4 はデフォルト 168Mhz 動作ですが 240Mhz までのオーバークロック方法は確立されているらしいので、333Mhz での動作を目指しています。

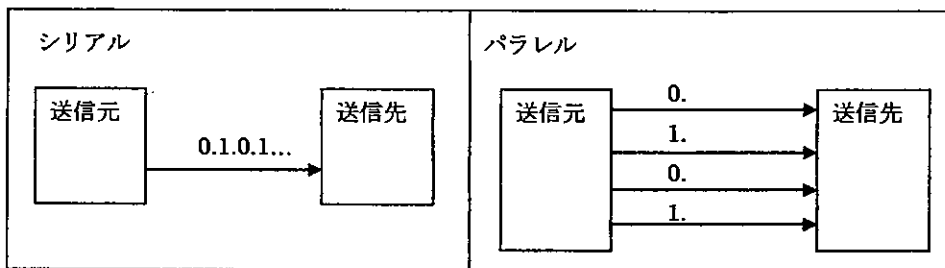
音声 IC は ARM cortex-M4F の内部命令にある音声デコード命令の実用性如何では搭載しないかもしれません。

◆ SPI とかパラレルって？

前の節で「SPI」という言葉がありましたね。この制作物のほかでもこの単語は様々なところで出てくることと思います。SPI のほかにも「16bit パラレル」や「SDIO」もそうなのですが、これらはすべて有線方式の通信(バスと言います)方法です。

バスの方式には大きく分けて「シリアル」と「パラレル」の二つの方式があります。

この二つの方式には大きな差があります。



上の図で言うと右側は「4bit パラレル」になります。

要するに「同じ線」の上を「時間」で区切って順番にデータを送っていくのがシリアル。「同じタイミング」で「配線」を分けてデータを送るのがパラレルです。

このとき何本のデータを同時に送るかで「Obit パラレル」かが決まるわけですね。

シリアルは送受信に必要な配線が(最低)1本で、そのためにケーブル間で通信の内容が干渉しあう「漏話」が発生しないというメリットがある一方、同時にいくつものデータをやり取りすることができないため、どうしても通信が低速になりがちという欠点を持ちます。

SPI というのはシリアルを規格化したもので、SD カードはこの方式も(SDIO については後述)利用できます。

SPI の特徴として通信を行う基本ピンを二つにし、送受信を同時に行えるということのほか、通信のタイミングを決めるクロックピンや通信先を選択するチップセレクトピンがあることで複数の機器を簡単に同期し、通信が可能という特徴を持ちます。

この制作物ではサブモニタにこの通信方式の液晶を利用しています。NOKIA3300 液晶は SPI にてコマンドを受け取り、色情報を順番に受信することで画面が描かれていきます。その他についてはパラレル液晶と一切変わりません。こちらの液晶はタッチパネル日搭載です。

SD カードは基本的に SPI を使って通信するものということになっているのですが、さらに上位の SDIO という通信方式を使い、高速にアクセスが可能です。ARM Cortex-M4F(STM32F4)ではこの通信の支援機能が追加されたので、それを使ってデータの読み書きを行っています。

ARM って?

この制作物の核になっているのが ARM というのは説明しましたね? 他の制作物には PIC という言葉がちらほら見えることだろうと思いますが、この制作物でその役割を担ってい

るのが ARM です。

では ARM とはなんなのかをこの節で説明しようと思います。

技術的な解説はインターネットを見ればいくらでも乗っていますので、「ARM ってなんなのか」だけに焦点を置いて解説していきましょう。

ARM は 32bitCPU アーキテクチャの一種です。

「Obit」というのは CPU が一度に作業できる計算の量。「8bit」であれば一回に 0~255、「32bit」であれば 4,294,967,296、「64bit」ならばなんと 18,446,744,073,709,551,616 まで一度に計算できます。一度に計算できなければそれだけ繰り返して計算することになるので、どうしても動作が遅くなります。

CPU は「中央演算処理装置」。PC などに入っているもので、計算を行うものです。アーキテクチャを一言で表すなら「設計図」です。設計図といっても CPU の形が決まっているわけではなく、CPU の設計思想やどんなことができるのか、そういったことが決められています。それを「ARM 社」という管理会社が「IP コア」という形で様々なメーカーに提供しています。それらのメーカーが実際の生産を受け持つわけです。日本企業では東芝や富士通が生産をしています。今回の STM32F4 はスイス STMicroelectronics 社が生産しているものです。

重要なのは CPU がどんなものに利用されているのかということですね。



この↑全部 ARM で動いているものです。

そう、iPhone や PS Vita やルンバも ARM で動いているんですね。最近 CM や広告で見る「TeOra 搭載スマホ」とかいうのはもろに ARM で動いています。TeOra というのも ARM の一つの形なのです。

さて、ここまで ARM と呼んできましたが、その意味はいったい何なのでしょう。

インターネットなんかで調べると「Advanced RISC Machines」という綴りが出てきます。

「Advanced」は「高度な」といった意味。「Machines」は「機械(ここでは計算機)たち」という意味です。この二つを当てはめると「高度な RISC 計算機」ということになります。RISC という単語の意味は分からずとも「高性能な計算機なの・・・かな?」ぐらいに思っ

てもらえれば十分なのです。が、この RISC というのこそが ARM の最大の特徴でもあるのです。どうせなのでここで RISC について少し解説したいと思います。

◆ RISC とか CISC って?

「じゃあ RISC ってナンだ?」

って話ですね。綴りだけで翻訳すれば「危険」の意味ですが、略語です。で、対義語が「CISC」。もとは「Reduced Instruction Set Computer」と「Complex Instruction Set Computer」。意味が分かりませんね。

翻訳すると「縮小命令セットコンピューター」と「複合命令セットコンピューター」(Google 翻訳談)らしいです。これで少し意味がお分かりでしょうか。

簡単に言ってしまうえば「ちいさな命令セット」と「おおきな(複合)命令セット」を持つコンピューターです。ここでいうのは命令というのは CPU にさせる作業の内容です。

✓ 「ちいさな命令」というのは簡単な計算とか作業。一回の時間は短く済みます。

✓ 「おおきな命令」というのは複雑な計算とか作業。一回の時間は長くなります。

これでなんとなく意味がわかってきたでしょうか。たとえて言うなら

↓ RISC コンピューターは簡単な道具を多く繰り返して使い物を作る。



CISC コンピューターは複雑な道具を少しの回数使って物を作る。↑

結果としてできるものは同じでも、それにかける時間や行程は全く異なるものになります。それぞれの特徴を挙げると下のようになります。

	RISC	CISC
構造	単純	複雑
1 作業の量	少ない	多い
1 作業の行程	多い	少ない
周波数	上げやすい	上げにくい
コード	単純	複雑(になりやすい)
電力消費	抑えやすい	抑えづらい
アーキテクチャ	ARM, Super-H, AVR, MIPS	x86(x64), H8

このように互いに得意不得意があり、その優劣は決め難いのです。

できるだけ構造を簡単にして省電力化を図りたい携帯電話や携帯ゲーム機や、そこまでの瞬間的判断を求められない家電等には ARM や Super-H のような RISC プロセッサ。

なるだけ計算効率を求めたい PC やサーバーなどには x86 や x64 といった CISC プロセッサが使われています。X86 とか x64 という名前になじみがないかもしれませんが PC に入っている Core-i3/5/7 や Core2 、サーバーに使われる Xeon などの CPU はこの x86 や x64 といったアーキテクチャを採用しています。

物無員大好き PIC の中身は RISC に近い CISC となっており、明確な区分はありません。x86 アーキテクチャの中にも RISC 的な構造が含まれていたり、ARM や Super-H も命令が複雑化していたりと、その区切りは薄まっています。

上に書いたような理由で一概には言えませんが、プロセッサ業界全体でみれば近年どうも CISC よりから RISC よりへ向かっていると感じます。電子部品の反応速度自体の限界が近づき、これ以上の性能向上が狙いづらくなってきているという現状があり、RISC 的なプロセッサを並列化することで作業一段一段の遅延を最小限に抑え、その繰り返しをすることで全体での高速化につなげようという狙いがあるのでしょう。また、近年デバイスの小型化が進み、バッテリーとともに持ち運ぶという形態が増えました。バッテリー駆動である以上電力消費は抑えなければなりません。このとき、デバイス全体に占める CPU の電力消費というのは予想以上に大きなものなのです。そんな場面でも RISC の低消費電力という長所が生きているのです。スマートフォンが軒並み ARM を採用しているのはこんな理由もあるのです。

◆ 感想とか

もうすこし面白く書きたかったんですけどやっぱりコンピューター系は理論理論しちゃいますね。まあ僕の文章力が根本的に足りないのですが。

目下の目標は新 ARM 基板の動作ですね。何しろ回路構成は勉強から開始。コンパイラもこれまで無償でつかえていた物が有料になってしまったので 1 から勉強しつつ自分でコンフィグなどを書く破目になりました。なんであんなにコンパイラって高いんでしょうかね使っている人の少ないマイコンは経験不足がダイレクトに響くのでつらいですね。PIC とは構成も考え方も違うアーキテクチャの制作を通じて CPU の設計思想の変化を感じ取ることができました。(といっても ARM もかれこれ開発開始から 30 年も経っているのですが。)

一人でプログラミングしているとコードが汚いものになりがちですが、それは無いようにできるだけ意識してプログラミングしたつもりです。OS を運用することでできるだけ高級な(人間よりな)プログラミングを目指しました。

数年前より ARM を使っている先輩はいらっしゃいましたが、十分に広まっているとは思いません。この制作物を一つの足がかりにして物無全体が脱 PIC を図り、ARM のような新しいマイコンへ移行してくれることを祈るばかりです。

SOUNDPAD⁶⁴ VER.3

高2 岸田 聖生

HISTORY

SOUNDPAD⁶⁴とは、僕が中3の時から作っている「Nintendo64のコントローラーの形をした楽器」のことです。

Ver.1はPIC内部で数マイクロ秒待機→出力をON/OFFすることによって方形波を生成、待機する時間によって発音周波数の変更を行うものでした。これは前回の文化祭で展示していました。

Ver.2ではYAMAHAのYMZ294という音源ICを用い、方形波+ノイズを発音させるものでした。これは冬休み入って3日程度で作り終わりましたが、前作ver.1と大した差が無いということでボツに。

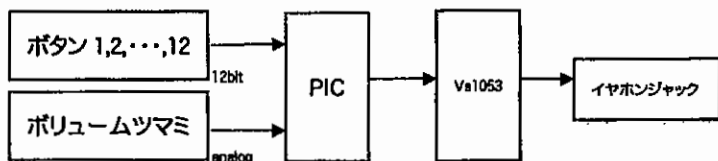
Ver.3では今までのものから大きく進化するためにまず「音色変更」と「打楽器の発音」を目標として掲げています。これらを満たすにはMIDIへと手を伸ばすのは必然的でした……

OUTLINE

この製作物は主に3つの部分より構成されています。

- ・入力部分 ……………コントローラー基盤の流用など
- ・送られてきたデータを元に波形を生成する音源IC ……………vs1053b
- ・上記2つの制御を行うPIC ……………PIC16F1938

ブロック図は



ボタン8つはド、レ、ミ、ファ、ソ、ラ、シ、ドといった風に8音階分の発音に使われます。

ここで注意していただきたいのは、ドを60といった値においた時、ド#は61、レは62、ファは65という値に置き換えることが可能ということです。

残りのボタンのうち2つは押されている間だけ

- ・上記の値を+1つまり半音上げる機能
 - ・上記の値を+12つまり一オクターブ上げる機能
- に振り当てられています。

さらに、残りの2つは押した瞬間に上記の値を+1/-1する機能に振り当てられています。移調された音階と同様になります。

ボリュームつまみは、位置によって音色をドラム含め129種類に変えることができます。

以上より、12のボタンとつまみを利用することによってあらゆる音をだすことが可能となります。

ボタンの機能配置は、大合奏バンドブラザーズというゲームを参考に作られています。

WHAT IS MIDI ?

MIDI について解説します。

MIDI とは電子楽器の演奏データを機器間でデジタル転送するための世界共通規格で、物理的な送受信回路・インターフェース、通信プロトコル、ファイルフォーマットなど複数の規定から成り立ちます。今回はその中でも「通信プロトコル」を使用します。

MIDI データというのは、「どのような音が鳴っているか」といういわゆる音声ファイルのような情報群ではなく、「どのタイミングで」「何を鳴らすのか」という楽譜のような情報群のことです。

MIDI データの送受信は USART という PIC の機能によって行われます。MIDI の場合、通信速度が規格されているので非同期式、なおかつ今回は音を出すだけなので送信のみ扱います。

MIDI の通信速度は 31.25kbps (31250 個の bit (=0 か 1) の信号を遅れるという意味) です。つまり、 $1/31250 = 0.000032$ 秒 = 32us が 1bit 当たりに必要な時間数となります。

信号の規格としてスタートビットに 1bit、データビットに 8bit、エンドビットに 1bit の計 10bit が送られます。

この中のデータビット、つまり 0x00~0xFF までの信号を何個か送ることによって MIDI のデータとなります。一例としては、0x9X 0xYY 0xZZ と 3つ送ることによって「Xトラックで YY 番目の音階を ZZ の強さで鳴らす」といったデータとなります。

MIDI には 16 個のトラックが存在しており、16 個のパート分けができるとイメージしていただくと思います。それぞれのトラックには自由に楽器を振り当てることが可能ですが、トラック 10 のみはドラムセットとして定義されています。

このような楽譜的データを元に、音を発したり止めたりといった動作を行うものとして今回は vs1053b というデコーダーを用いています。

INLINE

特徴としては、ツマミの位置を AD 変換で読み取って 129 個に分割し、ドラムか否か、否ならどの音色 (128 種) かで動作を分けている点、やボタンの立ち上がりと立下りどちらも読み取るようにしている点などです。MIDI の規格上、発音停止のデータを送らないと鳴りっぱなしになってしまうのであらゆるボタンの状態変化によってこまめに発音停止のデータを送るようになっています。

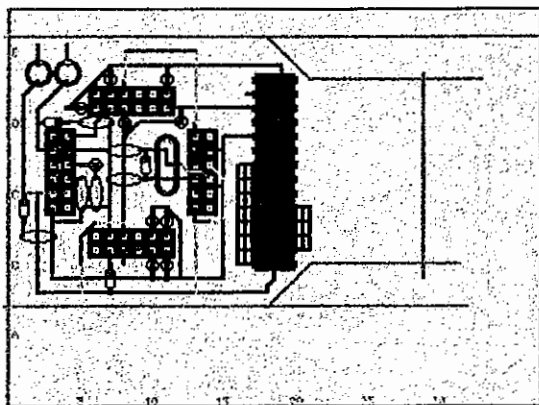
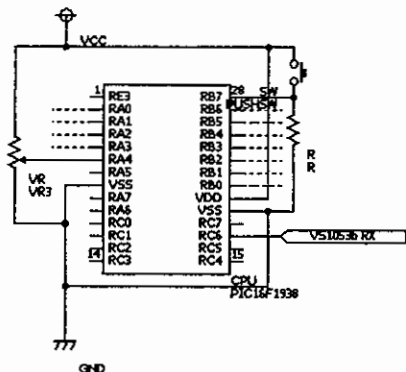
ドラムの場合は送るデータを 0x99... とすることでトラック 10 へ送信し、それ以外の場合は 0x90 に対してプログラムチェンジ (音色変更) の信号を送っています。

ボタンの状態判定には、「現在の値を記録→入力と記録値が変わるまでボタンの変更なし」という風にしてあります。PIC 内部には立下り又は立ち上がりエッジ判定機能がついているのですが、どちらか一方しか選べないためプログラム面で解決しました。

また、一度に 8bit*3 個送るのに対し、PIC 内部のバッファに相当する部分は 1 個分しかないのデータを送っている間の 320us の間は何もできないようになっています。

回路図は次のページのとおり (実は VR の出力は RA4 ではなく RA5 というミスあり) です。RB0~7、RA0~3 はすべて RB7 のようにスイッチが配線されています。Vs1053b 側の回路図はデータシートや「からくり工房「ききょうや」」様のを参考にほぼそのまま作っています。

配線自体はコントローラーの内部にほとんど収めるため、秋月電子で販売されている C 型基盤に



すべて収まるよう設計しました。実体配線図は右の様です。

ONE'S IMPRESSION

Oh…気づいたら文字ばかりのつまらない製作物説明に…。

二週間かそこらで作り上げた製作物ですが、なかなかおもしろい事ができて楽しかったかと思いません。基板サイズを調べて実体配線図を組み、その通りに配線して一発で動いた時の感動はなかなかのものでした。「作るって楽しい！」そういう初心者心を蘇らせてくれた一製作物でもありました。

ちゃんとドラムが鳴るように作れて、元々興味あったMIDIも触れて、音という視覚以外の刺激で楽しめるものが作れてと満足しています。本製作には影響の少ないようにやっただけです。迷惑だったらごめんね、石倉。

いつぞやの単純な方形波な製作物よりはよっぽど多彩なことができるようなものを作りましたがその反面、面白みと言いますか反応と言いますか、ギャラリーの興味津々度が低くなったような気がして少しさみしいところもあります。単純なピコピコ音のほうが受けやすい気がするのはなぜなのでしょう？

文化祭の時のお客様の反応が気になるところです。

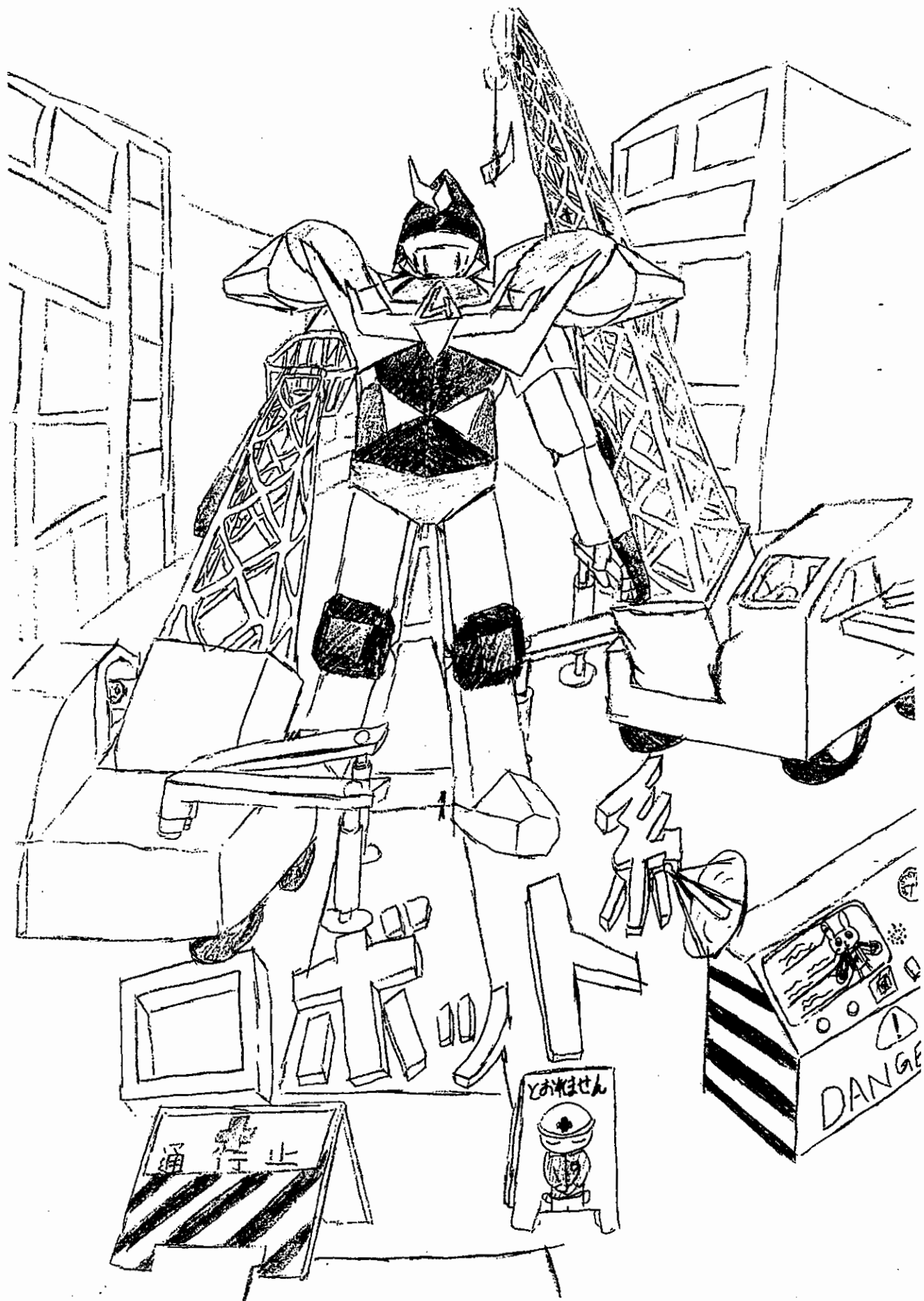
犠牲者 - 追悼 -

圧電スピーカー…Ver.1 についていたのはいいもののスペースの都合で追放。ずっと材料箱の中で転がっています。

YMZ294…調べてプログラムをバッチ書きした上で購入。配線してすぐ動いたのはいいのだが限界を感じ4日目あたりに vs1053b へ以降することに決定。

水晶発振子…5v 駆動だったので使えず。3.3v用のを頼んだつもりだったのだが…

コントローラー本体…分解され、スタートボタンとスティックとZボタンは引剥がされ、基板は真っ二つになり…と様々な改造を受けた一番の被害者。彼がいたからこそこの製作物は成り立つのであり、彼無しには完成し得なかった。黙祷。



通行止

おかしん

DANGER

ATTENTION PLEASE!!!

製作：高2 岸田 聖生

中3 石倉 匠

協力：物無員の皆様

■ストーリー

はッ

私は知っている。この情景を。

デジャヴだろうか

いや、たしかに昔、こんなことが… そうだ！あれは二年前のことだ。

「もうこんなのデモンションで操縦しようぜ？！なっ！」

「いややめろって危ないって危険だからやめて！はい注目！」

「知るか！出航だ！」

「ちゃんと聞いてくれ—————！」

そう、あの時の私はまだ若かった。

どれくらい若かったかというと XXXXX と聞いても首を傾げていたくらい若かった。

今では立派な XXXXX となり、私もまた注目を必要とする側へなったのであった。

■概要

この本を手にとってるあなたは二年前の回路図集を持っているでしょうか

そう！今回は二年前のもののバージョンアップ(?)です

持っていない方は、昔、卓上サイズのホバークラフトがあったということを頭の隅っこの方にボツリとおいていただければ幸いです

さて、簡単にこれがどういふモノなのか説明すると、ホバークラフトです。

しかし、今回ののはなんと人が乗れちゃうんです！

操作もホバーに乗りながらタッチ液晶で操縦できちゃうんです！

◆仕組み

タッチ液晶パートと本体パートに分けて説明しようと思います

まずは、タッチ液晶です

担当は、皆さんに笑顔をお届けする石倉です

「液晶光らせるのなんて簡単じゃね？！」と思われそうですが、一般の人に頑張りが理解されないのが物理部の悲しいところでもあります

重い物を持ち上げることが可能なのです。(＊ただし、空気は流体と言えども簡単に圧縮されてしまいますし、空気はスカートと地面の間から流れ出ていくので密閉空間ではありません)

本製作では、吸気口にブローと呼ばれる強力な送風機を取り付け、スカート側にうきわを使用しています。もともと二人乗りで計画していましたが、ブローの力不足か本体の大きさのせいかで一人乗用となってしまいました。

VI 何が優れているのか？

前述のとおり、ホバークラフトは地面とスカートの間隙があります。つまり、浮いているので地面からの摩擦を原理的には受けません。実際にはやはり摩擦がありますがうまくバランスを取れば総重量 60kg をおよそ 3kg の張力だけで滑りだすくらいにはなっています。なので、今回のように小型モーターを2つ使用しただけでもおよそ 2cm/s^2 という低加速度ながら一応動くことが可能です。

VII 何に劣るのか？

ブローというのは駆動抵抗たっぷりのモーターをぶん回しているわけですから、暖房器具並みに電力を食います。そしてそのエネルギーは音へも変換されています。騒音が大きく、なかなかフルパワーで回す気になれません。さらに、熱としてもロスするのでこの制作物では回し続けることができません。よって、駆動時間が短くなってしまいました。おまけにモーター用の電源が必要なためコードレスにはなりません。

IX この製作物の特徴

本製作ではスカート部に外直径 80cm のうきわを使用しています。浮き輪を利用したのは、「安価である」と「扱いやすい」という二点の理由があります。もちろん防水性の布を利用したもので良かったかもしれませんが、ブローの出力に不安があったため半固形状の浮き輪を選択しました。

実際に乗ってみるとわかるのですが、浮き輪一個だと非常に不安定で、重心の位置をほぼ真ん中に持ってこないで浮き輪が完全に地面と接触してしまい、進まなくなってしまう。ホバークラフトの全長は全高の7倍くらいがいいみたいな話を聞いたことがあるのですが、展示することを考えるとあまり大きなものは作れず、小さい浮き輪を使用するに留まっています。また、大きくすると空気が流れだす部分も増えるのでやはりブローの能力の限界が訪れてしまいます。

今回は、液晶をタッチして方向転換という仕様になりましたが、おそらくバランスを取るのに精一杯で方向転換なんてしていただいれなんでしょう(笑)。またまた、プロペラが完全にむき出しなので当たると非常に痛い。血が出ます。

■この先文化祭まで

安定化を狙うなら大型化が必然となってしまいますがそうなる展示スペースなどの話は既に述べた通り。試してみる価値は十分あると思いますので展示するしないは別としてやってみるかもしれません。が、まずはプロペラに保護カバー等を付けないと死人が出る勢いなので格子か何かを組まなければいけません。あとは搭乗者のバランス感覚次第で進まないこともないと思います

■感想

石倉

初めて「液晶」を扱いましたが、光らせるまでが一苦勞でした。まだ文化祭一か月前の時点では完動していませんが、手伝ってくださったみなさんありがとうございました

自分の中では、関数を組み合わせて新しい関数を作るのが楽しかったです

ケーブルの先にソケットを付ける作業が、自分が思っていたよりうまくできたので、自分の配線技術が上達していたことを実感しました

プログラミングも、先輩に聞くだけでなく自分でも基礎的なことはできるようになりました

私は、これからも裏配線はしないつもりです

岸田

いやーこれで製作ともお別れかと思うと寂しいですねー。最後の最後にロボット系のある種の極みと言える「搭乗」を一部達成できたのは良かったような気がします。

製作中は選辺の二足歩行ロボットの設計を全て担当したり(多分名前は載ってない。残念。)佐藤琴斗のラジコンを全面的に教えたりなどまーやりたいただけやらせてもらいましたが、もっと、こう「創ッ！」ってのを全身で体感できるようなことがしたかったですね。パワードスーツとか。自分はプログラムは教えたばかりでほとんど回路製作と本体設計をしていました。何か革新的な技術を導入できたらなんて思っていましたけど触ったのはせいぜい無線モジュールのみ。先輩がさらなる新技術を取り入れてくれることを願うばかりです。

ここまで読んでくれた方にはきつといいことばかり待ってます。さすがに FET の中身などの説明はどこにも書いていませんが全部の製作物の説明を読めばたいいのことはさらっと書いてあるはず！

■終章

石倉「岸田さん」

岸田「ヤダ」

石倉「僕の話聞いてください！」

そう、彼もまたひとりの“XXXXX”だったのだ……

Fin

食べ砲台

中3 宮嶋 山本

高2 水野

協力 みなさん

概要

このロボットは本体から撮っている映像をタッチすることで、その方向にピンポン球を発射するという砲台です。

ストーリー

僕は今日食べ放題に来ていた。

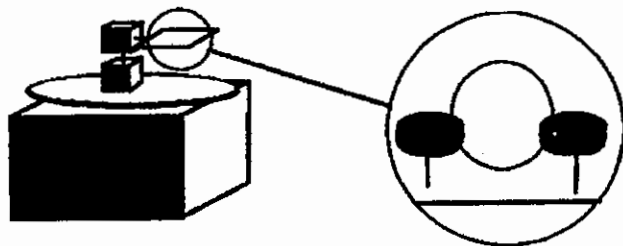
そして仲良く食べていたところに宇宙人が攻めてきた。逃げ遅い悲鳴を上げる人々、飛び交う光線銃。

そこに一筋の光とともに食べ砲台が現れた。僕は食べ砲台のコントローラーを持ち、食べ砲台とともに食べ放題を守るために立ち上がったのだった。

3ねんせい やまもと りょうたろう

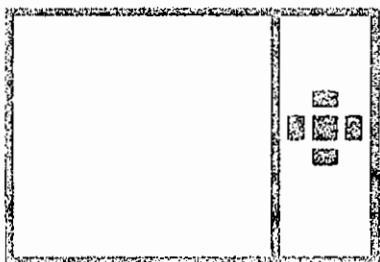
外観

左図の立方体部分がステッピングモーターで、右図の発射部分拡大部にはサーボモーターとDCモーターが使われています。



ステッピングモーターはそれぞれ、土台の左右、銃口の上下左右と連動しています。またDCモーターはミニ四駆などに使われるウルトラダッシュモーターを左右に一個ずつつけることで、軽くて速い回転力を出しています。そのDCモーターの電源は単二電池二個(3V)から供給しています(その他の電源は、コンセントからの電力供給による12V電源です)。

コントローラー



全体がタッチできる液晶になっていて、左にはカメラから送られている画像、右には上下左右移動ボタンと発射ボタンがついています。画像をタッチするとその方向にピンポン球が発射されます。

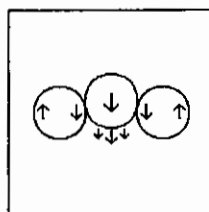
ロボット部門

ブロック図



発射方法

打つ瞬間は DC モーターをバットイングマシンのように回すことでピンポン球を打ち出します(中心がピンポン球で左右が DC モーターです。)。基本 DC モーターは回りっぱなしなので、発射時にサーボモーターを使うことでピンポン球を DC モーター部分まで誘導します。



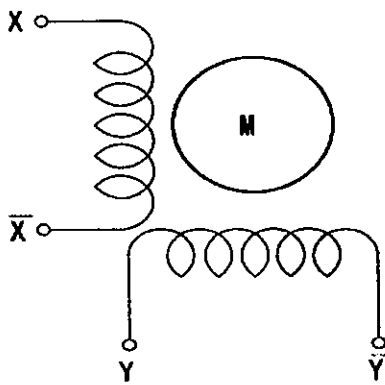
ステッピングモーター

ステッピングモーターとは回す角度を調整することができるモーターで、高速回転するDCモーターとは性質が異なります。同様に角度の調節できるサーボモーターより優れている点は、

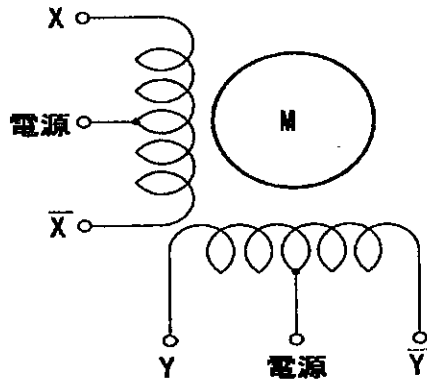
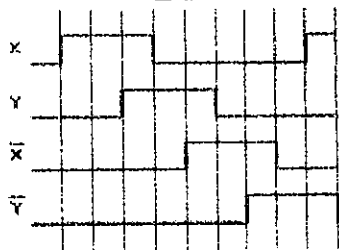
- ・何回転でもできる。
- ・信号の速さで回転速度も調節できる。
- ・静止トルク(静止中の外力への抵抗能力)が強い

などがあり、逆に振動が大きい・動トルクが小さいといった欠点もあります。

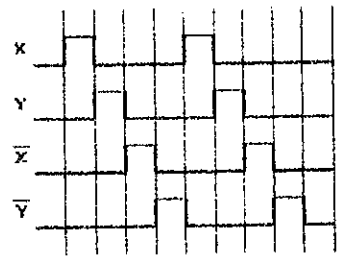
またステッピングモーターにはユニポーラ型とバイポーラ型がありますが、仕入れの関係上両方使用しました。



正転



正転



モーター左図がバイポーラ型で右図がユニポーラ型です。それぞれ下の1相励磁のパルス波、1-2相励磁のパルス波を出すことによって、回転させています。それぞれ違う理由はトルクと電力どちらを優先するかです

コントローラー部門

ブロック図



カメラ

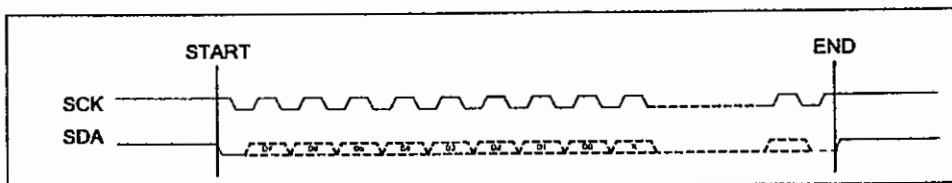
OV7670 というカメラを使用しています。aitendo というお店で購入可能です。ロボットに設置することでロボットから見た映像を読み出しています。

ピン配置は下表です。今回は色データを RGB565 という形式で行うため、2 BYTE で一点の色決定を、すなわち通信二回ごとに1ピクセルの色を決定しました。

3V3, GND	電源です。
HREF	行が変わるごとに H になります
VSYNC	一画面表示し終わると H になります
PLCK	1 BYTE 分のデータごとに発振します
XCLK	動作クロックです。外部供給
SCL	SCCB 通信という通信に使用
SDA	SCCB 通信という通信に使用
D0-D7	色のデータを 1 BYTE 分送ります

SCCB 通信

カメラとの、色データに関する通信はバラシル通信(たくさんの信号ピン)によって行なっていますが、カメラの初期設定を行うための通信は SCCB 通信というシリアル通信(少ない信号ピン)によって行う仕様になっています。この初期設定によってカメラの色の強さ、明るさ、コントラストなどを決めます。初期設定の内容はトランジスタ技術の記事を参考にしました。



上図のようにデータ通信を行います。SPI 通信などと違い一度に通信できるデータの量は決まっています。そのため、通信開始の合図に「SCK が H のときに、SDA を H から L にする」という動作を行う必要が、通信終了の合図に「SCK が H のときに、SDA を L から H にする」という特別な信号を行う必要があります。よって途中のデータを送る場面の間は、SCK が H の間に SDA の HL が変わらないように注意する必要があります。今回は 8bit ずつ通信を行いました。

その他

液晶は aitendo の CP2401T というタッチパネル付きの物を使いました。タッチパネルと液晶の詳細については石倉くんを参照にしてください。

PIC32MX460F512L は microchip 社の 32bit マイコンで、最大クロックが 80MHz とあまり速くないので、本格的にカメラを行うにはもう少し速めのものを使えばよかったかと思えます。

感想

山本 初めてのプログラムはいろいろと一人でできず大変だった。けれどいろいろな人に教えてもらい何とかなった。なんだかんだで楽しかった。

宮嶋 僕はチカチカのため、あまり砲台を作る時間がありませんでした。少しでも砲台を作れて楽しかったです。これから頑張って、早くチカチカを終わらせて本製作に戻りたいです。

水野 今までずっとマイコンと液晶に向かってばかりだったので、今年のロボットは新鮮でした。ロボット大変ですね、プログラムが含んでも動かないし。でもロボットの部分は中二がふたりとも頑張ってくれたので、結構カメラに集中できたかなと思います。もうちょっと液晶とかカメラとかを教えたかったな。

初めは、またひとりでコンピューター系をしようと思いましたが後輩らが待っていたので一緒に組みました。コンピューター系に巻き込もうとしましたが、二人がロボット推しだったので砲台を作ることにしました。お陰でコンピューター系とロボット系の融合という新しい挑戦ができたのではないかと思います。二人には感謝です。

昔コンピューター系は一人じゃなきゃ難しいと言われましたが、コンピューターとロボットを一つの製作物に織り交ぜる場合は複数人のほうが楽だなと思います。

Alto-2

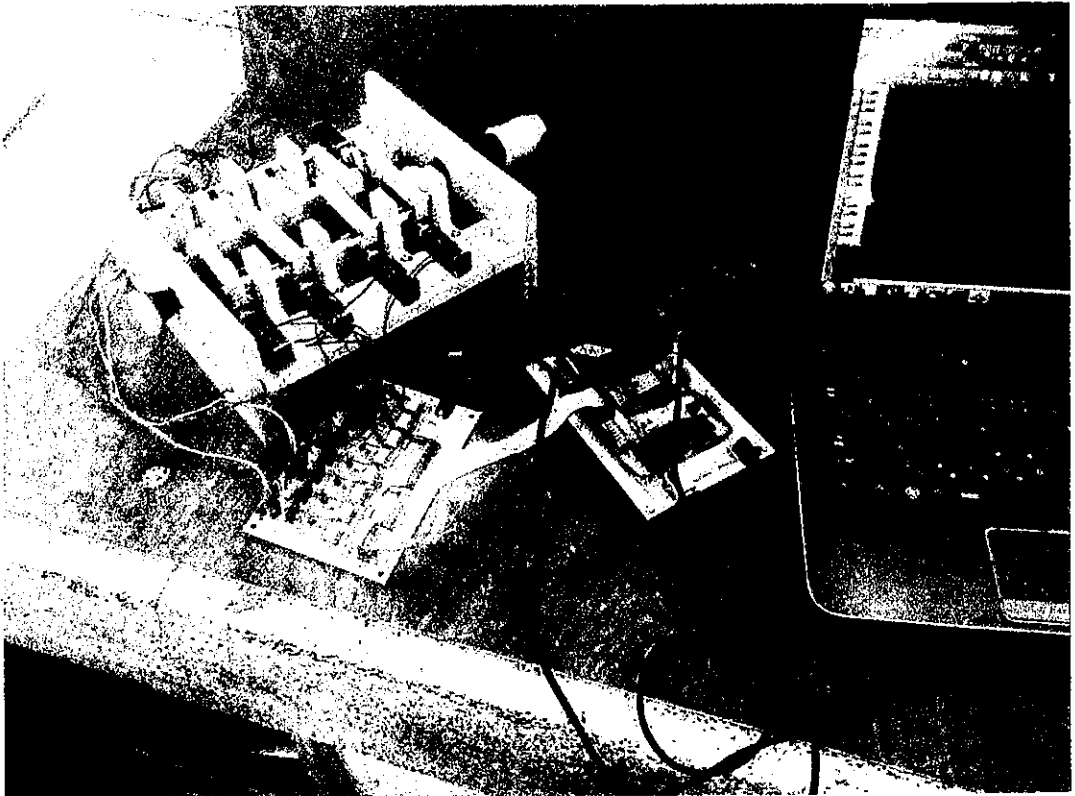
制作:河村 洋一郎
協力:物無のミナサマ

♪概要

去年の Alto に引き続き、リコーダーの自動演奏ロボット第2弾です。ソプラノリコーダーを演奏しますが、名前は Alto を踏襲して Alto-2 です。この間の試作機などはありません。

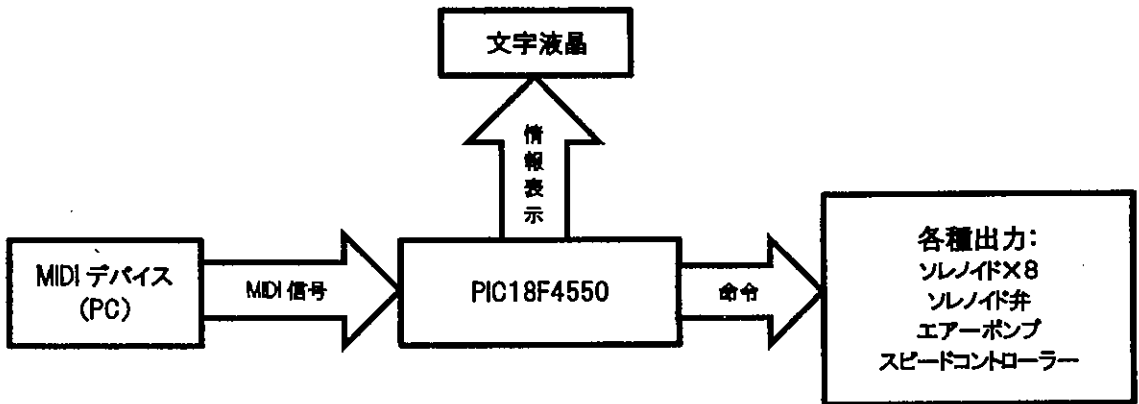
今年の Alto-2 も Alto と同様に、空気を送るポンプ、穴をふさぐ指部分、それらの制御回路によって構成されています。去年同様タンギング、サミング機能を実装していて、これに加え、1.駆動機をサーボモーターからソレノイドに、2.MIDI 通信機能の追加、3.風量調節をポンプ単体から、ソレノイドバルブ、スピードコントローラーの併用、といった大きく3つの機能が追加変更されています。これにより、基盤に乗っているマイクロスイッチを押したらチャルメラを演奏するというだけのお粗末なものから、今年は PC と接続してどんな曲でも演奏できたり、常人には不可能な超高速タンギングができたり、ロボットならではのといった感じの演奏が可能になりました。もはや名曲喫茶を風前の灯火ですね、ハイ。

え？音楽の授業が無くなってリコーダーロボットつくってまで赤点回避する必要がなくなったなら別に第2弾作らなくても良かったって？ほ、ほら歌手とか気に入るまで何回もレコーディングするとかあるじゃないですか、これはそういう職人のこだわりてきなやつなんですよ、ハハハ。



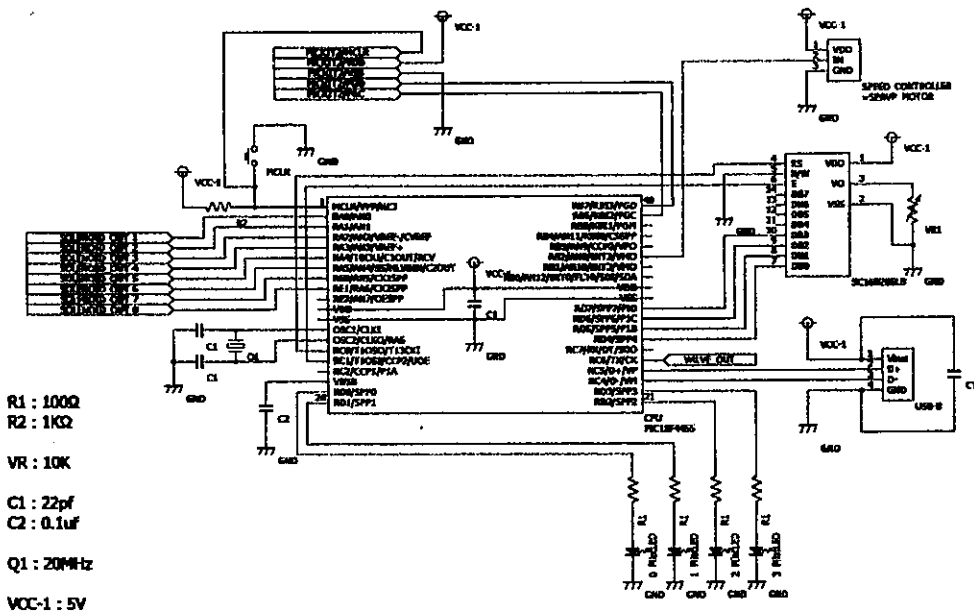
♪回路図

♪ブロック図

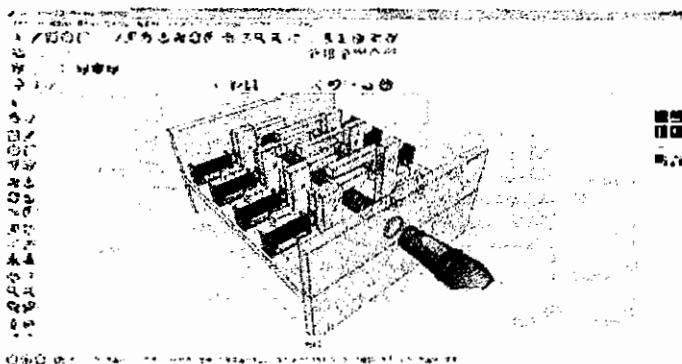


ブロック図はこんなかんじです。MIDI デバイスは MIDI 信号を送れば電子ピアノだろうと PC だろうとなんでもいいです。

♪回路図



具体的な回路図は上図のようになっています。ソレノイド駆動回路はもう少し後のページに書いてあります。PCとの通信には18F4550内蔵のUSB機能を使っています。これも詳しくは後述。この回路は3/15日現在のものに変更される場合があります。



今回は GoogleSketchUp を用いて 3D 設計図を書きました。設計図といってもただの 3D モデルなのでそこら辺はあしからずってことで、GoogleSketchUp は建築なんかで使われる 3D モデリングソフトです。3DCAD というには物足りないようですが、扱いが非常に簡単で、自分の雑魚 PC でもさくさく動いてくれ、しかも**無料**です。(←ここ重要。3DCAD は本格的なものになるとウン十万円とか普通にするんで学生には手が届きにくい存在です。*SketchUp Pro は有料) また、Trimble ギャラリーから世界中のユーザーが作ったモデルがダウンロード可能で、様々な素材を使用することができます。自分も Sketchycat 様の Recorder を利用させて頂きました。丸っぽくて作りにくい形をしているリコーダーも、他の人が作ったものを改造すれば過ぎに使えるというわけです。因みに最近の建築業界では思いついた建築アイデアを紙に書くより前に SketchUp で起こす人も居るんだとか、それほど使い勝手がいいってことなのでしょうね。操作方法はココには書きませんが、解説動画を見れば 1 時間ほどで基本的な操作がマスターできるので是非お試し下さい。
<http://www.sketchup.com/itl/ja/> から無料でダウンロード出来ます。
 上図は設計中の Alto-2 です。

♪ 指

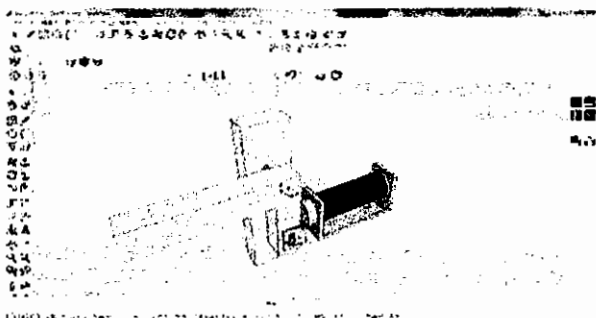


図1

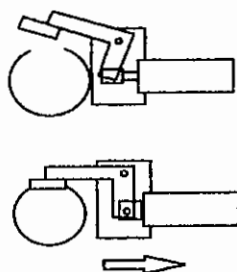
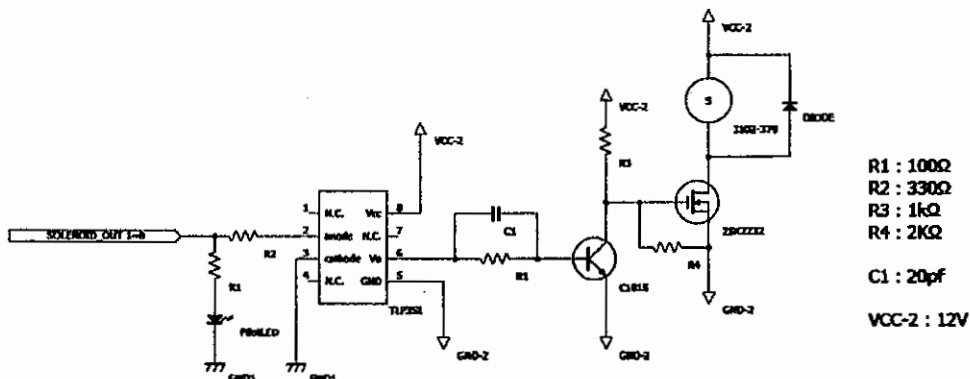


図2

Alto-2 の本体は木材で作られています。木材は安く加工も家で使い勝手はいいですが、強度的な問題があり負荷がかかる所には向いていません。(技術料の某 0 田先生談)この穴をふさぐ指部分も同様に木材を加工して作りました。図1で茶色の部分が木材です。
 また指部分はアクチュエーターとしてソレノイドを用いています。ソレノイドは固定コイルと可動鉄心によって構成されていて固定コイルに電流を流した時に、固定コイルが可動鉄心を惹き付け力を利用して可動鉄心をスライドさせる運動器です。電流を流した時に鉄心を引っ込めるものをプル・ソレノイド、押し出すものをプッシュ・ソレノイドといいます。電気を流した時だけ鉄心が引っ込んだり押し出されたりということで自動車や家電など様々な所で使われている便利な奴だそう。バネソニックとかも作ってるらしいです。トルクはそこまで大きくありません。10tとか欲しい人は油圧でも使って下さい。
 Alto-2 で使っているのはプル・ソレノイドで、CKD 社製の DS-06E J102-122(日米無難で1個700円)で図1の水色と灰色の部分に相当します。
 穴を高く掘るのは簡単で図2のように行なっています。ソレノイドは指の下部分を引っ張って、指全体を回すというやり方ですね。ソレノイドを使う利点は大きく2つあり、1.電流を流す、流さないだけの操作なので制御が簡単、2.スイッチングを高速度できる、という2点があります。発熱するとコイルが壊れてしまうため、長時間の使用はできませんが、どうせ一曲2、3分しか演奏しないんで、今回はあまり気になりませんでした。Alto がサーボモーターを用いた機構で穴の開閉にかかった時間が 0.4 秒だったのに対し、ソレノイドは 0.1 秒以内での動作が可能で Alto-2 の高速演奏を可能にしています。
 コイルの巻数とかめんどっつい計算をしなくてもメーカーの仕様書通りに動かせばいいので、楽っちゃ楽です。お陰でコイルのことは何も分からず終いでしたけど...



これがソレノイドの駆動回路です。TLP351はフォトカプラといって電氣的絶縁をしてPIC側の回路を保護しています。つまり、マイコン側と駆動回路側を別系統の回路にしているってことです。R2がないとフォトカプラが爆発ファイヤーします、ご注意ください。ソレノイドの周りに付いているダイオードは逆起電力からの保護用です。逆起電力とは鉄心が動くときにソレノイド自体が発電しちゃうことによる電流で、回路に悪影響なやつです。普通のDCモーターとおなじですね。因みにTLP351も2SK2232も両方1個100円です。これを8個作るわけですから、その他部品と合わせて2000円相当の部品たちが居ることになります。2000円あったら製作者の家から小田原までいけちゃいますね。

♪送風機構

Altoでエアープンプのみの制御で行っていた送風機構を大幅に改善して、エアープンプ制御+ソレノイド弁+スピードコントローラーの3つの部品による送風機構をつりました。これにより、より細かな流量の調節や、ソレノイド弁の高速動作によるタンギングが可能となりました。エアープンプとスピードコントローラーの制御で流量の調節を、ソレノイド弁で空気のON-OFFをしています。

(1) エアープンプ、スピードコントローラー制御

エアープンプ制御はDCモーターの制御とここでは同じなので割愛させていただきます。スピードコントローラーは水道の蛇口みたいなもので、回すと空気の流れる量が変わるというものです。(左図)スピードコントローラーには日本ビスコ社製のユニオンストリート JSU12(楽天市場で2000円)を使っていて、サーボモーターで可動部を回転させて制御しています。ここではサーボモーターの制御についても割愛。因みに「〇〇社製の安物を買って壊れた」ってレビューがあったんで安物(社名は出しませんよ)は買いませんでした。Alto-2ではエアープンプで大きな流量の調整、スピードコントローラーで細かな流量の調整を行い、組み合わせることで高音、低音にも対応した流量調節を可能にしています。

(2) ソレノイド弁

ソレノイド弁は空気弁の一種で、先ほど説明したソレノイドで開閉させる弁のことです。エアープンプより高速に空気をON-OFFできるので便利です。CKD社製 圧縮空気用パイロット式2ポート電磁弁 EXAC602C3(アマゾンで2700円)を使いました。メーカーの仕様書通りに動かせばなんにも問題ありません。仕様書はちゃんと熟読しましょう、高電圧かけると弁が高速で振動して死にそうな音を発します。え、すぐ電源抜けば問題ないけど(爆)



スピードコントローラー



ソレノイド弁

♪プログラム

Alto-2ではMIDI通信による譜面読み込みを実現するために、PIC18FシリーズのUSB機能を使いました。MIDIは簡単に言うと電子楽器みたいなもので、YAMAHAとかが作った規格らしいです。これを使えば、煩わしい楽譜入力をしなくても、PCから譜面を受信するだけで演奏できるようになるというわけですね。USB通信は敷居が高いように思われがちかもしれませんが、マイクロチップ社が無料で提供しているUSBフレームワークを用いれば誰でも簡単にUSBアプリケーションを作ることができます。具体的には、C¥Microchip Solutions v2012-04-03¥USB¥Device - Audio - MIDI¥Firmwareのところにあるファームウェアを改造して作りました。デフォルトのプログラムはC4(低いド?)からC5(高いド?)までMIDI信号を出力し続けるプログラムです。プログラムの全てを載せることはできないので、改変箇所だけ乗っけておきます。

1.サーボモーター、DCモーターのPWM用にタイマ割り込みの追加

```

void YourHighPriorityISRCode(){
    USBDeviceTasks();

    /*タイマ割り込みとUSB割り込みの優先順位に関して問題が発生(1/12現在)
    USBDeviceTasks()に位置をここに持つことで解決、原因はUSBの割り込み仕様か?
    割り込みベクタが見当たらないんだけどどうなってんの?とどにかく動いたのでこれでいいや*/

    if(INTCONbits.TMRQIF){
        割り込み命令記述部分
    }
}

```

YourHighPriorityISRCode()にタイマ割り込みを追加しました。タイマの設定はInitializeSystem()で行なっています。コメントで製作者の適当な性格が見え隠れしていますが、これは本当に謎のバグだったんで動いたからいいやってことになりました。

2.MIDI データの受信と格納

```
if((USBDeviceState < CONFIGURED_STATE)&&(USBSuspendControl==1)) return;
if(!USBHandleBusy(USBRxHandle))
{
    for(int i = 0; i<64;i++){
        MIDI_RECEIVED_DATA[i] = ReceivedDataBuffer[i];
    }
    USBRxHandle = USBRxOnePacket(MIDI_EP,(BYTE*)&ReceivedDataBuffer,64);
    TRIS0bits.TRIS0=0;
}
```

ProcessIO()に上のものを追加することで MIDI_RECEIVED_DATA[]に MIDI データが格納されます。因みに MIDI_RECEIVED_DATA[1]が note_on/note_off 情報、MIDI_RECEIVED_DATA[2]が音階情報、MIDI_RECEIVED_DATA[3]がベロシティ(音の強さ)情報です。

3.ソレノイド制御、文字液晶制御用のライブラリの追加

文字液晶に関しては例の lcd.lib.c です。ちょっと改変させて頂きました。ソレノイド制御は渡された音階データを指運表に変換して出力するというものです。特に難しいわけでもないので割愛。

4.その他

PC 側に表示されるデバイス名を変更したり、それぞれの指運表データ、それぞれの音に対応する風量データのヘッダファイルの作成など細かなことはいろいろ追加、変更しています。

USB ファームウェアの中には、PIC をマウスデバイスにしたり、MIDI ホストをしたり(USB ホスト機能があるのは上位種のみ)、HID として認識させているんな事ができたり、と様々なファームウェア、もといサンプルプログラムがあります。英語さえ何とかできれば(最悪 Google 翻訳にぶち込めばおk)基本的に何とかなるんで是非色々試してみてください。

♪感想

これが最後の制作になるわけです。中1ゲーを除いた過去3回の制作は全部ロボット系でしたね。元々、小学生の頃からロボットとかプログラミングとかが好きでよく日本科学未来館とかに連れて行ってもらって ASIMO とか見ていて、その延長線上で感じてロボット作ってたわけです。最近久しぶりに日本科学未来館に行ったんですが、ロボットとか超電導の展示とかに興味津々で目をキラキラ(中にはギラギラ?)させているちびっ子たちがいっぱいいて、なんか過去の自分を見ているようで、将来麻布入って物理部無縁班でロボット作らないかなあとか思ってしまった。中には本当にそうなる人もいるのかなと思うと胸が熱くなりますな。

でロボットなんですけど、やっぱり実物が動くってのはいいもので、プラモデルとかとは違う楽しみがあると思います。(お金かかるけどね)。初めてリコーダーの音になった時の感動は忘れません。ベヤング自動配膳ロボットも作れたかったんですけど、~~夫~~高校生の事情のより断念してしまったのが、そこが唯一の心残りといったところでしょうか。

反省点はちょっと妥協が多かったところです。去年のを超えればそれでいいやって思っていた時期もありましたねハイ。ひとり制作は妥協しようと思えばどこまでも堕ちていくんで注意ですね。その分、出来ることの自由度は高いんで精刃の剣って感じですね。要は自律の精神が大事と。(ワ麻布ッポイ)

最後に、ここまで呼んでくださった方々(居るのかそんな人!?)、World で作っていると緑とか赤の波線だらけで、「ココ日本語おかしいだろ」って突っ込まれまくりな、読みにくい文章でしたが最後まで目を通して頂きありがとうございました。

♪参考

<https://www.google.co.jp/> 一番お世話になりました。そこら辺の麻布教師より物知りです。

<http://translate.google.co.jp/> グーグル翻訳。正直コレなしでは辛かった。たまに変な訳が出てくるのはご愛嬌。

<http://esyokumin.deo.cc/index.html> この方、すごいです。18F の USB なんかにとどまらず色々やられてるので是非他の物無員も参考に。

『PIC で楽しむ USB 機器自作のすすめ』後関 哲也著 一度は聞いたことがあるであろう後関さんの出版されている本のうちの一つ。参考になりました。

>STRIDER-“0”WD



製作者:H1花園、M3市川、M3佐伯

~STORY~

ここは私の住む池だ。人間の視点からみれば、水溜りと言うかもしれない。しかし、私からすれば立派な池であり、住処なのだ。

私が何者か？自分の名など考えたことも無いが、君たち人間の間では私は“アメンボ”という名で呼ばれているらしい。

私の種族は、生涯を水と共に過ごす。それが種族の習性であり、使命なのだ。と周りは言っていた。しかし、私はとてもじゃないがそう思うことができない。地球では、陸が広がるのに対し、水は無くなる一方らしい。現にこの間の大きな水溜りは、干上がって無くなってしまったではないか。

「今こそ、陸に進出するべきなのだ。みんななぜそれが分からない。」私は今日も呟く。聞く者など居ないのだが。

「その望みを叶えようか？」・・・声？ 見上げると、そこには人間が居た。人間には私の声など分からないはずだが・・・

「私の名前は博。この辺りではDr.アツシーの通り名で知られている科学者だ。今は、虫と会話ができる装置を用いて君とコンタクトを取っている。私の元にこないか？」・・・この人間はどうやら信用できそうだ。私は彼に連れられ、研究所へと向かった。

「ではこの台の上に乗ってくれ。すぐに終わる。」何をすると言うのだろうか。まあここまで着たからには彼に身を委ねるとしよう。

「・・・フフフ、今回もいい出来だ・・・」・・・？「まさかこんなにも簡単に実験体が手に入るとはな・・・」・・・実験体？「アメンボの推進方法は素晴らしい。この推進力を利用した機動隊を組織すれば、世界征服も可能だ・・・」気がつくと、自分の体が金属に覆われていることに気づいた。足には車輪がつき、見るからに醜い姿となっている。「今に見てる・・・あの池のアメンボどもはみんな改造だ・・・Dr.アツシーの名を轟かせる時が来た・・・」

漸くすべてを理解した。ひよんな思い付きのために騙されて利用され、さらに池の仲間までも危険に晒してしまったのだ。自分は半分望んだような形でこのようになったが、何も知らない池の仲間に出しをさせるわけには行かない。

・ ・ 戦うしか、無いようだな。いいだろう。

兵器となったこの体の力を、逆に故郷の池に轟かせてやろう。

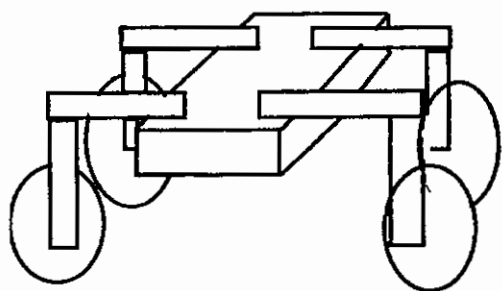
～概要～

ストーリーは適当に読み飛ばしてください。長文申し訳ございません。

言うなれば、アメンボ型4足走行ロボットというものです。4足歩行ロボットではないです。

足の先にはタイヤがついていて、これによって地面をすべるように移動します。

外観図↓



←理想

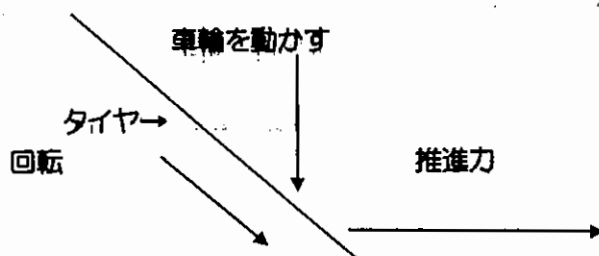
イラストクオリティ低いですね。ペイントに慣れませんでした。

このイラストでは、楕円がタイヤ、足の継ぎ目が関節を表しています。

関節にはサーボモーター(後述)を、タイヤは薄くて大径なレゴタイヤの一種を使用しています。

このロボットの走行方法は、スケートに似ています。

車輪を斜めに傾けつつ横向きに移動させることによって、前後への推進力を得ます。

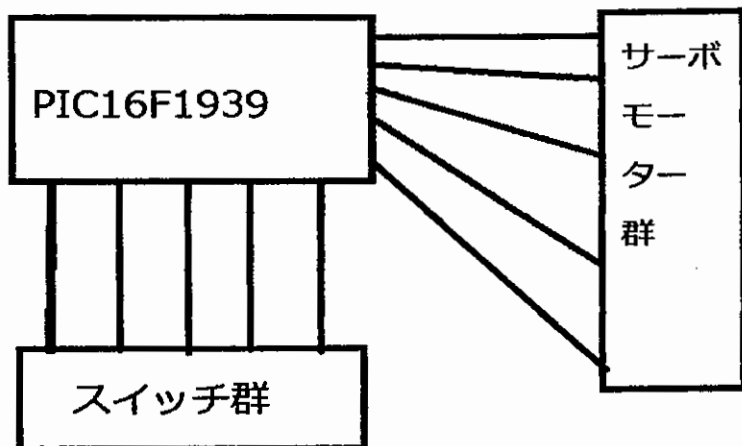


～部品、回路～

>PIC16F1939 . . . このロボットを動かす頭脳にあたる IC。“16F”と“1939”で中間クラスの最新型であることを示す。

>サーボモーター . . . 関節を動かすモーター。回らない代わりに角度を決定、固定できる。

>回路 . . . ブロック図



～感想～

今回の製作では主に本体のほうを担当しましたが、本体はすぐにできてしまったため、結局プログラムを二人で勉強していくと言う形になってしまいました。おまけにその本体も1,2度壊れ、仕様変更が存在して作り直しなんてこともあり、自分の計画性の無さを痛感しました。そして自分もプログラムをあまり理解していないため教えることもろくにできない酷い有様でした。漸く後輩(主に市川)とのコミュニケーションもとれ、双方が作業に理解をもち順調に進んできているので、この勢いで文化系に出せるものにしていきたいです。(花●)

～ここから後輩(市川)に交代します。フォントの違い等ご了承下さい。

これからはプログラム担当の 市川 がお送りします。

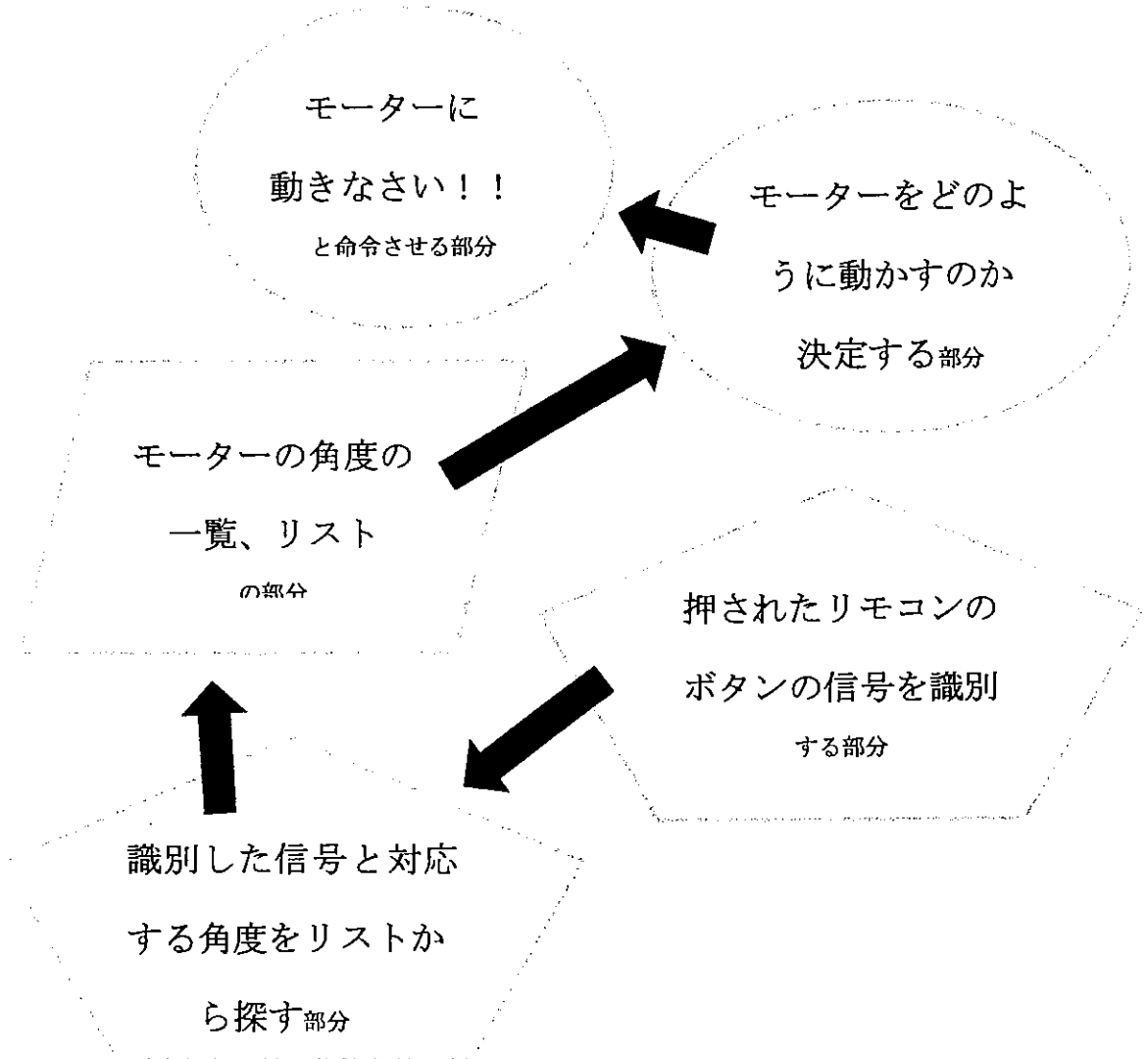
プログラム解説

ここでは、ロボットのプログラムについて簡単に解説してしまいます。まずロボットは、本体解説でも言ったように

モーターと金属板だけで動いているんです

だから、プログラムも単純明快で少な目なんです（別に簡単なわけではないですよ）

ロボットは以下の要素で動いています。



C言語を使って書いています。C言語とはプログラムの言語の一つで、iPhoneアプリを作るための言語はC言語が進化したモノです。つまり、一番メジャーなんです。

ペルセウスの武器も上に書いた方法と同じ方法で動いています。全部サーボモーターというモーターを使っているの、基本プログラムはかえなくて済みました。

さて、本題に入ります。プログラムがどうなっているのか、大体わかっていただけたかともいいます。スイッチからの信号を読み取り、モーターの角度を決定、モーターを動かす、という一連の動作ですね。ここでもっと詳しく説明します。

まずメインプログラムと割り込みについて。

その名の通り、メインプログラムはプログラムの骨組みです。割り込みもその名の通り、メインプログラムの途中で割りこんで動作します。この二つを改造していくことで細かな設定ができます。

僕は、メインプログラムでリモコン、モーターの管理をしていて、いざモーターを動かすぞ！の部分で割り込みで書いてます。イメージ的には、リモコンを操作するとメインプログラムでモーターを動かす設定が作られて、割り込みが入るとその位置にモーターが動くといった感じです。

次に角度のリストですが、僕は1(押されたボタン)と2(モーターの角度)の対応表を全て打ち込みました。つまり、前進ボタンが押されたら、モーターはAからBへ動いてCを経由してDに行きなさい、という一連の動作を全てリスト化しました。

その次は割り込みですが、ここでは一番悩みました。14個のモーターを同時に動かすので、「電力」が足りなくなってしまうんです。そこで4つずつモーターをグループわけして、動かす瞬間を少しずつずらしてみました。

大体こんな感じでプログラムは構成されています。もっともっと書きたいところですが、長くなってしまったのでここまでにしておきます。また、少ない文章で分かりにくいと思います。解らないことがあったら、遠慮なく部員に聞いてください。

つくった後の感想

【市川】

物無員の皆さんのおかげでここまで来れました。

モーターが動かなかったり、割り込みがうまくいかなかったりと今まで沢山有りましたが、皆さん、特に花園さんや佐伯が丁寧に教えてくれたり時には叱ってくれたりして(アメンボー←保留)はできました。

今年の経験やもらった知識は来年に生かしていきたいと思います。

最後になりますが、共同制作者の花園さん、佐伯、そして物理部の皆さん、ありがとうございました！

【佐伯】

このロボットの製作には僕は売り物担当だったので、あまり関われませんでした。でも完動が見えてきたのでよかったです。そして初の共同製作でしたが、とても楽しかったです。

共同制作者の花園さん、市川、物無員の皆さん、ありがとうございました。

早く売り物の量産を終わらせ、完動に貢献したいです

Three-way Standoff

製作: H1黒田 健太、M3 横尾 幸丸
協力: 物無の皆様方

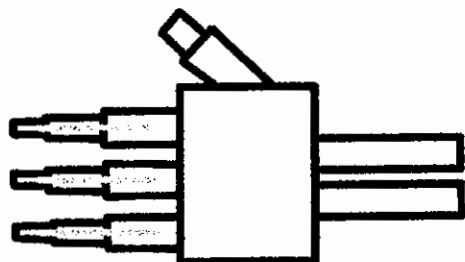
～始まり～

「何故だ！ どうして勝てない！」 そう、彼は一切じゃんけんに勝つことができないのだ。彼の名前はケンタ。とことん運のないごく普通の高校生…のはずだった。しかし、そんな彼の前に現れた謎の人物、ユキマルが彼の運命を変えることとなった…。ユ「そのあんた、ちょっとこいつを使ってみないかい？」 ケ「誰だあんたは？」 ユ「そんなことはどうでもいい。ただあんたを勝たせてやろうと思ってな。」 ケ「本当にそんなことができるのか？」 ユ「ああ、こいつならそれが出来る。そう、この“Three-way Standoff” 通称TSならな！」 ケ「これが…TS…」 ユ「あんたが勝ちたい相手にこの手袋をはめるといい。そうすれば必ず勝てる。」 そうしてケンタはTSを携え旅に出る。こんなに怪しい手袋をはめてくれる親切な相手など少ないということにも気づかずに…

～概要～

単純に言うと同じじゃんけんロボットってやつです。金属でできた手部分と相手の手を眺む手袋部分に分かれています。本当は手袋なんて使いたくなかったんですが、無理でした。使用上の都合ってやつです。※ここから先は現在試験段階にあるため実際の使用と異なる場合があることをご了承ください。

右が手部分の内部構造です。四本指なのは内緒です。全部開いた状態がパー、全部閉じた状態がグー、一番下の指だけを曲げた状態がチョキを表します。

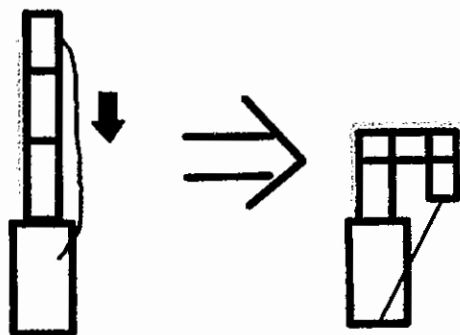


実際は手袋をしている(はず)なので、いかにも金属、という感じにはなっていません。どうでもいいことですが、なぜ左手かというところ製作者が左利きだからです。(はい、どうでもいいですね)

次に、指を曲げる仕組みを説明します。指の背側にゴムバンドを張り、指の腹側からテグス(下図細線)で指先を引っ張ることにより曲げています。白黒なのでわかり辛いかもしれませんが、右図指近くの色違い線がゴムバンドです。テグスはラックとモーターを使って引っ張っています。ラックというのはギアの円運動を左右運動に変換する部品です。モーターを回し、その力でラックを引き寄せる感じです。

手袋部分では圧力センサーを内蔵し、相手が出す手を認識します。どうやっているかというところ、手を握ると手の甲で出っ張る部分がありますよね？ 手袋のそこに当たる部分に圧力センサーを内蔵し、曲がっている指がどれかを判別しちゃおうってことなんです。

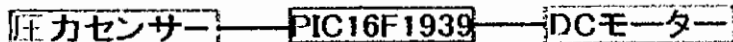
ただし、この方法だと案外高い確率で誤作動を起こすので、この製作物のコンセプトである「絶対じゃんけんに勝てるロボット」ではなくなりました。でもそれもご愛嬌ということで…^^;



本体の方の説明もひととおり終わったところで次は横尾君がプログラムとかの説明をしてくれるそうです。以上H1黒田がお送りしました。あとはよろしく。

はい、中三の横尾です。ここからはプログラム(おもにAD変換)について説明します。

～ブロック図～



圧力センサーで相手が曲げた指を検知しそれをPICに伝えそこでA/D変換を行います。その結果によってどの指を曲げるかを判定し、対応するDCモーターへ信号を送るといった簡単な構造です。

～A/D変換について～

A/D変換のAとDとはアナログとデジタルの略称です。アナログとはなめらかにつながっている数値のことを、デジタルとは離散化された数値のことを言います。わかりやすくいうとアナログが小数、デジタルが整数といったところでしょうか。そのアナログをデジタルに変換するということなので瞬間の数値を出せるわけです。今回のプログラムでは圧力センサーからアナログ数値を出す為、それをデジタルの数値にするためにA/D変換を用いました。実はここが一番難しかったです。物無の皆様にご手伝ってもらいました。ありがとうございます。

～プログラム～

コピペです。ですがこの通りにプログラムしたからといって必ず正常動作するというわけではありませんのでご了承ください。

```
#include <htc.h>
#define _XTAL_FREQ 32000000
__CONFIG (FOSC_INTOSC & WDTE_OFF & PWRTE_ON &
MCLR_OFF & CP_OFF & CPD_OFF & BOREN_OFF &
FCMEN_OFF & IESO_OFF);
//__CONFIG (FOSC_INTOSC & WDTE_OFF & PWRTE_ON &
MCLR_OFF & CP_OFF & CPD_OFF &
BOREN_OFF & FCMEN_ON);
__CONFIG (WRT_OFF & PLEN_ON & STVREN_ON);

long light;
long yokoo1;
long yokoo2;
long yokoo3;

long counter=0;

int touch1;//中指
int touch2;//薬指
int slate=0;//0:グー 1:チョキ 2:パー

void main(void){
  OSCCON=0b01111010;
  TRISB=0b00000000;
  ANSELB=0b00000000;
  TRISC=0b00000000;
  // ANSELC=0b00000000;
  TRISA=0b11111111;//入出力設定 0=出力
  ANSELA=0;//入力をデジタルとして参照
  TRISB=0b00001100;
  ANSELB=0b00001100;
  ADCON0=0b00001001;//AN2を使用、AD変換回路有効
  ADCON1=0b00000000;//左詰め、Focs/2、Vss、VDD
  GIE=0;PEIE=1;ADIE=0;ADIF=0;
  RB0=1;
  __delay_ms(500);
  RB0=0;

  while(1){
    counter++;
    if(counter%1000==0){
      light+=1;
    }

    ADCON0=0b01000001;//AN8を使用、AD変換回路有効
    GO_nDONE=1;
    //while(GO_nDONE);
    yokoo1=ADRESH;

    ADCON0=0b01001101;//AN9を使用、AD変換回路有効
    GO_nDONE=1;
    //while(GO_nDONE);
    yokoo2=ADRESH;
    //しきい値を2v程度と設定
    //2v/5v = 0.4 255*0.4 = 52
    if(yokoo1>=52){ touch1=1;}
    else{RB0=0;}
    if(yokoo2>=102){ touch2=1;}
    else{RB0=0;}
  }
}
```

```

switch(light){
case 0:
RD0=1;
break;
case 1:
break;
RD1=1;
case 2:
RD2=1;
break;
}

!(counter%3000){
!(touch1==1){
!(touch2==1){
else{
}
}
else{
!(touch2==1){//チヨキ

RA0=1;
RA1=0;
RA2=1;
RA3=0;
RA4=0;
RA5=0;
}

else{//←
RA0=1;
RA1=0;
RA2=1;
RA3=0;
RA4=1;
RA5=0;
}
}
counter=0;
}
}

```

～感想～

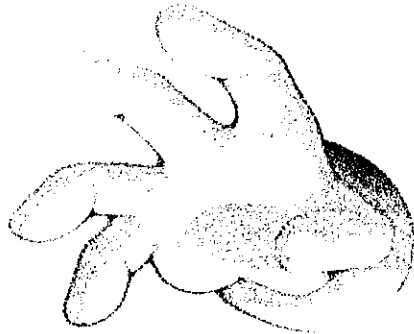
H1黒田

正直めんどくさかつ…失礼。今回このTSを立案したのは僕なんですけれども簡単な気持ちで「こんな人作れたら面白いよね」とか言ってた案が通っちゃってびっくりしました。でもまあ満足はしてる…はず。(まだこれ書いてるの製作途中なので) あ、製作物名のTSってのは「三すくみ」って意味らしいです。後輩の横尾君が結構働いてくれました。正確には横尾を手伝った物無員の方々ですが。ほんとに足を向けて眠れませんね。感謝してます。来年とかはもっと簡単なものでもつくりましようかね(笑) お読みいただきありがとうございました。

M3横尾

未だやることは残っていますが楽しくやらせてもらっています。楽しすぎてあまり進んでないのも現状ですが…。プログラムはまだまだ未熟者ですがこれからもよろしくお願いします。

将来的にはこんな感じに→



資源探索収集機試作呂号型

(O R C A S M E E M
= Ore Resources Collection Autonomy Search
MachinE Experimental Model)

コードネーム~Ro.ver~

製作者: H1 四柳 雄太

協力者: 川本を始めとする物無の皆様

0、 Ro.ver についての調査レポート

私、四柳は α 国への諜報活動により α 国の機密情報を入手した。以下はその全文である。

1、 開発背景

19xx 年、長引く戦争による資源不足に悩まされていた α 国では小さな鉱石を効率的に精製・抽出・融合させ一塊にする技術が開発された。しかしそのためには鉱石を一か所に集めなくてはならない。そんな作業を大規模にすれば敵機に見つかることは確実だ。そこで α 国は資源収集のためのロボットを作ることになった。それがまさにこの ORCASMEEM なのである。

さらに今回私は ORCASMEEM についての機密文書を入手した。以下はその一部である。

2、 機密文書~概要書~

機体の概要について。

Ro.ver は鉱石採取並びに分布推定のための機体である。本機体は自律往復運動による鉱石の収集と分布推定のための収集情報の確保を目的としたものであり、走行経路の記憶等の自律制御により目的を達成するものである。

持ち出し厳禁

3、機密文書~内部仕様書~

機体内部の動作等について。

Ro.ver は機体制御のためのコンピューターと駆動用モーター、外部情報収集のための測距・感圧センサーを搭載しセンサーの情報に基づいて鉱石の収集活動を行うように設計されたプログラムで動いている。尚、機体の簡易化並びに活動範囲拡大のため分布推定に関連する装置・プログラムは搭載せずに推定は本部で行うものとし、機密保持のためこれ以上の詳細は伏せられるものとする。

持ち出し厳禁

注釈

原文には内部仕様の詳細が書かれていなかったが別ルートの調査により情報入手したため以下に記載する。

- ・コンピューター…マイコン(32MX440F512H)を使用。

家庭向け電化製品の制御などにも使用されているものの一つで、使用者が自由にプログラムを書き込むことで様々な動作を可能にする。また多様な動作を実現するために電圧測定など多くの機能が内蔵されプログラム実行速度も最大で秒間 80M とかなり高性能になっている。

- ・モーター…走行用の DC モーターとアーム駆動用のサーボモーターの二つがある。

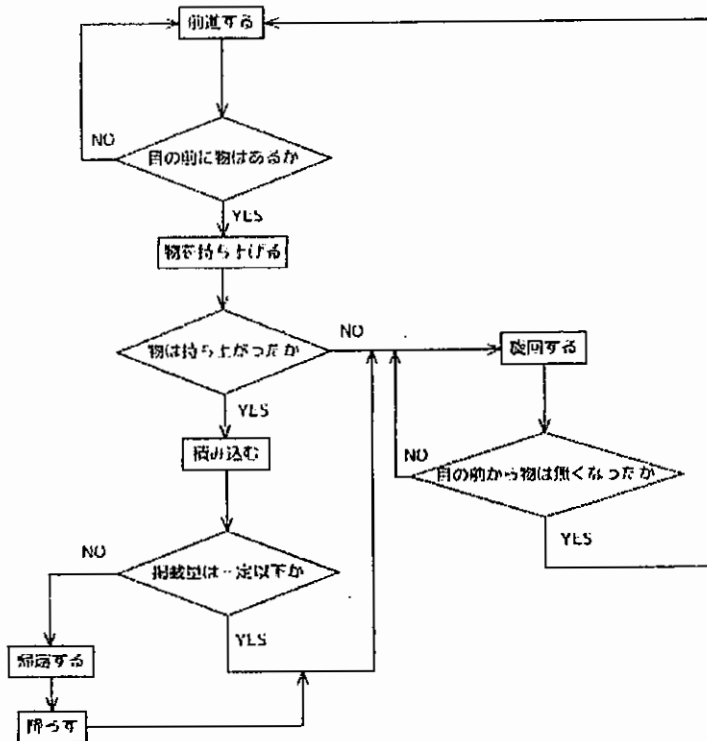
直流電流で回転し電流の向きにより回転方向を変える DC モーターと、信号のパルス幅によって角度を変化させるサーボモーターによりそれぞれ走行とアームを制御している。

- ・センサー…正面の物体への測距用モジュールと収集した量を量る感圧センサーの二つがある。

赤外線を照射し反射をもとに距離を測定する測距モジュールを複数使用することによりモジュール正面の壁・物体の存在と距離を把握し、圧力を測定する感圧センサーにより掲載済の収集物の重量を把握している。

どちらも情報の出力は電圧変化によるアナログ情報のためマイコン内の A/D 変換機能を使用することで情報をデジタル化している。

・動作時のフローチャート



ORCASMEEM はこれによって採取可能な鉱石類を片っ端から取っていき、また旋回時に極力反転を防がせ掃選時には内部に保存している DC モーターの制御の逆を効率的に辿ることで、自律走行を成し遂げている。

注釈終わり***

4、機密文書~回路設計書~

Ro.ver は簡易化のため主だった回路にあたるものとしては DC モーター制御のた

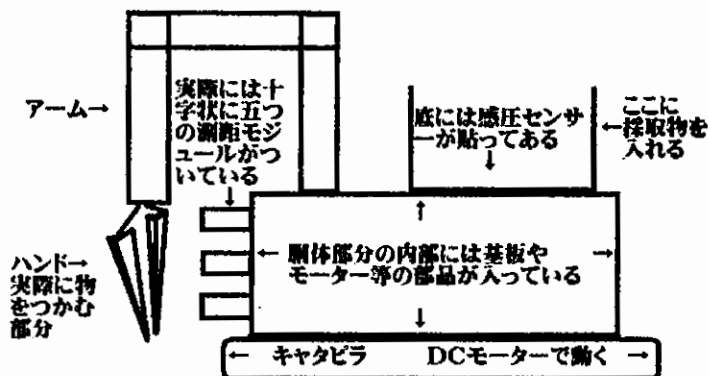
めの H ブリッジ回路とマイコンの動作の安定化ためのコンデンサ等の取り付け以外は有しない。他の動作は全て一つ当たり 2~4 本程度の部品の接続とマイコン内の動作のみで構成されている。

持ち出し厳禁

掲載する機密文書は以上である

5、 外観について

残念ながら私はこのレポート作成時に ORCASMEEM の画像を入手することができなかった。だが、簡易的な絵ならば掲載できるのでそれを残しておくことにする。



6、 開発者(Y氏)からの証言

私は実際に開発を行ったというY氏から話を聞くことができたので以下に載せる。

これが大勢の人に見られる頃、Ro.ver は表舞台からは消えているかもしれない、むしろ表舞台に立っていないかもしれない。何故ならばハードウェアよりソフトウェアの開発を優先してしまったからだ。形と性質のどちらが欠けてもいけないが人が何かを見るためには形が必要だ。その点で私は致命的なミスを犯していたのかもしれない…。或いは軍によって存在が消されるかもしれないが…

7、 …という壮大に誇張したストーリーにお付き合い下さいまし

てありがとうございました。

無駄に大きな背景世界にお付き合い下さいまして本当にありがとうございました。そして上で開発者が嘆いていた通り執筆中の今現在の時点で呂号君はまだ形になっていません。が、形が間に合いさえすればいいだけの話なのでようはつじつまが合えばいいのです。

そんなところで少しばかり今回の過程を振り返ってみると正直に告白してロボットなめてました。ごめんなさい。って感じです。おそらく本体を先に作れば今とは違った現実があったでしょう。

と、そういった経緯の中でとりあえず呂号君の形を作っていくために再び製作に取り掛かる開発者ですが完成するかわからないので完成しても単に消されるかもしれないので、呂号君が表舞台に立っていることを願いつつ改めて作業に取り掛かるのでした。

WALKING!!

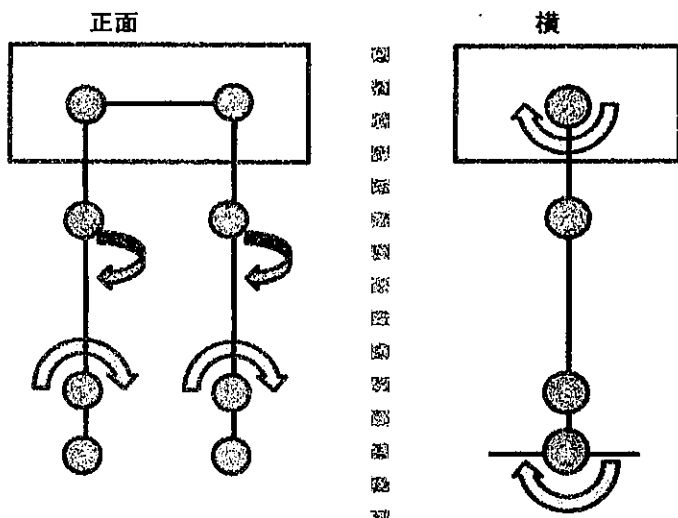
製作: M3 渡辺悠太

協力: 物無の皆様

～ストーリー～ 「彼」は歩き出す。
僕らの思いに答えるために…
「彼」は歩き出した。
僕らの期待に添うために…
僕は「彼」に叫んだ。
「WALKING!!」

～概要～ はい、見てわかる通り二足歩行ロボットです。8個のサーボモーターを使っています。
サーボモーターについては、あとで説明したいと思います。
現状では歩くかどうかすら不安ですが、こいつは自由に歩き回れるロボットなのです。
それでは、構造について説明しようと思います。

～構造～



図のように動く様、●の部分にサーボモーターが取り付けられています。

- ①: 足を振る為のサーボ
- ②: 回転する為のサーボ
- ③: 重心を傾ける為のサーボ
- ④: 地面につける角度を合わせるサーボ

～本体の様子～



本体は今3号機です。1、2号機は適当に作りすぎました。
下の画像が現在のロボットの様子です。
上に被さってる発泡スチロールの半球はデザインで
基板を中に収納しています。

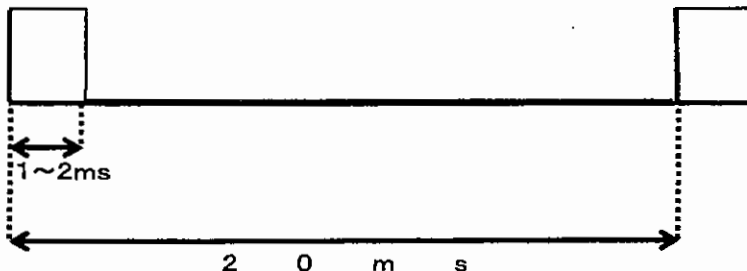
～動作～

前進用の動作だけ書いておきます。

③を使って、片足を上げる。→①で脚を前に出す。→③を戻し、④で両足をつける→
③と④で重心をずらす。→①で脚を前に出す。...

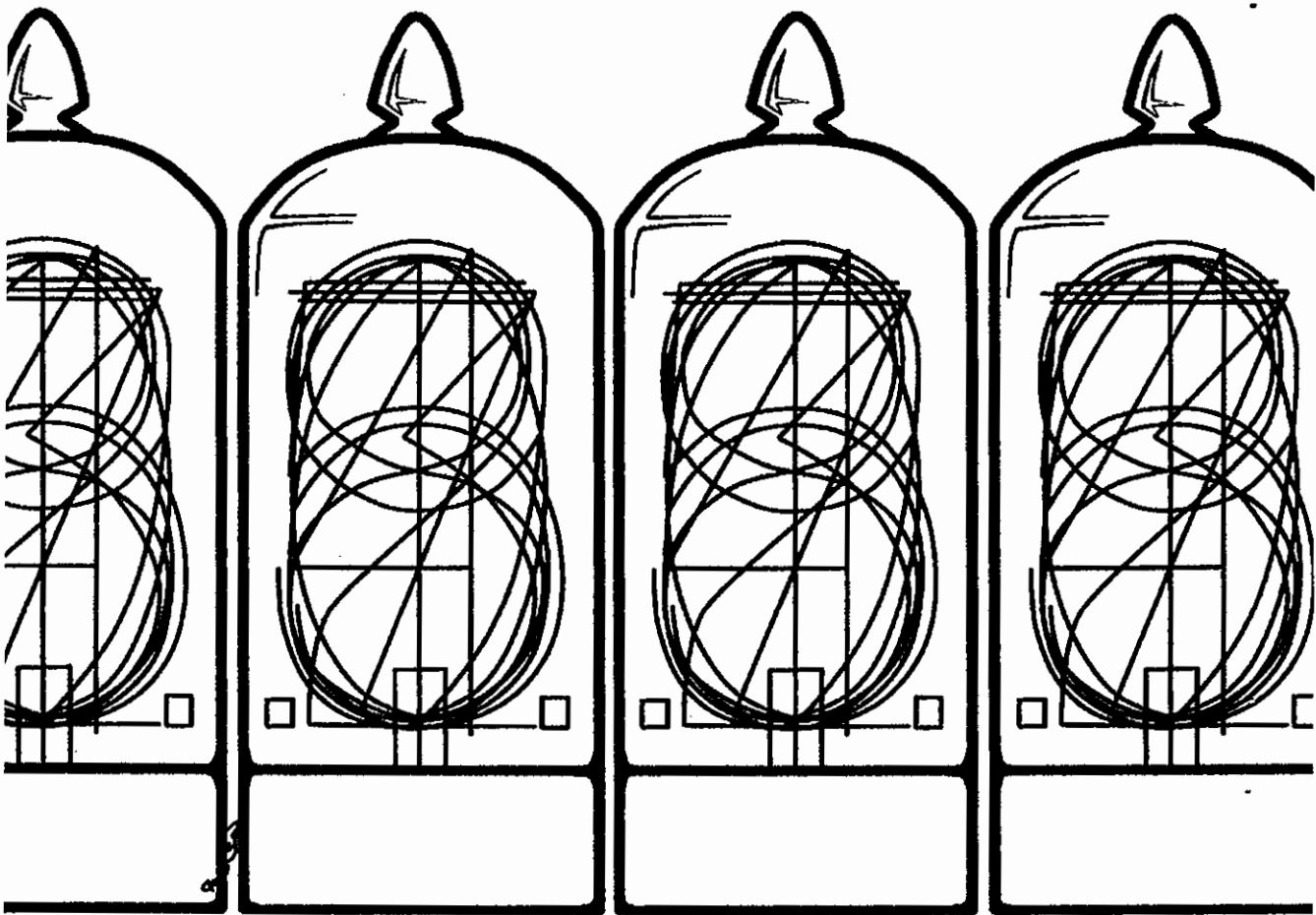
～サーボモーターとは～

サーボモーターには、+と-の入力線ともう一本信号線があり、
その信号線にHとLを入れることによって角度を制御するモーターです。
20ms周期の中でHをどれくらい出すかによって角度を制御出来ます。
一般的には、1ms(0°)～2ms(180°)位ですが、サーボによって違うので
確かめる必要があります。
例えば、そのサーボに1.5msを入れると90°、1.25msを入れると45°になります。



～感想～

今回はいきなり「二足歩行をやろう。」と思い、一年間やってきましたが
ロボットはやはり難しかったです。助けてくれた皆さんありがとうございます。
金属加工は苦手です。ミスもたくさんしてしまいました。
しかし、ロボットの本体はとても重要だということが1、2号機の段階でわかり
立派な3号機ができ、良かったです。来年に生かそうと思います。
まだ現状では当日満足いく製作物になるかどうかはわかりませんが、
あと少し全力で活動したいと思います。
歩いてほしい...いや、歩かせます。歩かせて見せます！



ANALOG

Catch the WAVE!!

高2 岸田 聖生

---Story---

夢をただひたむきに追いかけていたころが懐かしい。手が届きそうと思うほど離れていく夢。あの頃、自分に足りなかったのは現実を現実的に魅せつけてくれる“相棒”だったのかもしれない…

---概要---

~~ひどいストーリー詐欺です。~~この製作物は製作者の「スペクトラムアナライザって可愛いよね」と「今年ってアナログ系の製作物なくね」という発想によって作られたものです。

この製作物は、「オーディオスペクトラムアナライザ」と呼ばれるもので、一番わかり易い？例でいうと Windows の Media Player にある視覚エフェクトの中の「バー」です。つまり、入力された音楽を周波数ごとに分解、それぞれの強さ(音量)を視覚的に表示するというものです。

といっても、高性能な PIC で高速フーリエ変換なんてやっていられるほど僕の頭は数学に長けていないので、電子工作では割りと初歩的な RC フィルタというものを使いました。

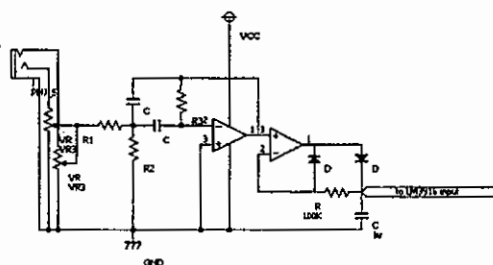
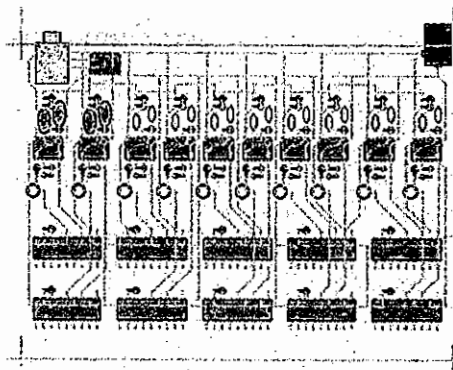
---原理---

コンデンサと抵抗器などをつかってある回路を組むと、特定の周波数以外の交流は流しにくくなるという変わった特性をもつものができます。他にもコイルとコンデンサを使ったものやコイルと抵抗器を使ったものがあり、それぞれ AM ラジオや電源などで使われています。

この特定周波数のみを透過させる部分を BPF(バンドパスフィルタ)と呼んだりするのですが、この透過周波数を適当に可聴域に分布させるように BPF を複数設置することによって可聴域をそれとなく分解してスペクトラムアナライザを作っています。BPF を通過したそれぞれの周波数の交流は整流回路と平滑回路を通して、電圧を LED の数に置き換えてくれる便利な子で表示しています。

電圧をLEDに置き換える子は、コンパレータをいくつも内蔵していて、電源電圧をいくつにも分圧し、各位置で入力電圧と比較するというものです。こちらは秋月電子で売っているLM3916というものを使いました。

---回路---



上：一周波数ごとの回路

左：純粋の実体配線図

---特徴---

といてもほとんどがインターネットで調べててきた回路のバクリなので特徴といった特徴はありませんね。10ch分も実際に作ろうとしているのが一番のアピールポイントかもしれません。

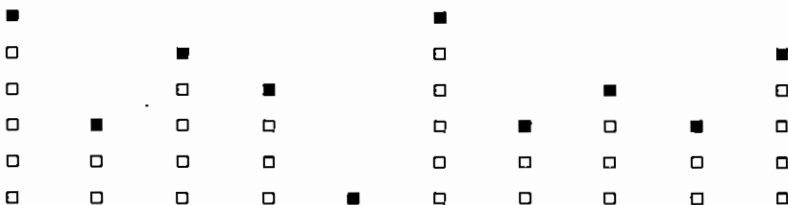
一番すごいところはLM3915から表示用のバーLEDに伸びる配線が10本×10個で100本もあることです。

---感想---

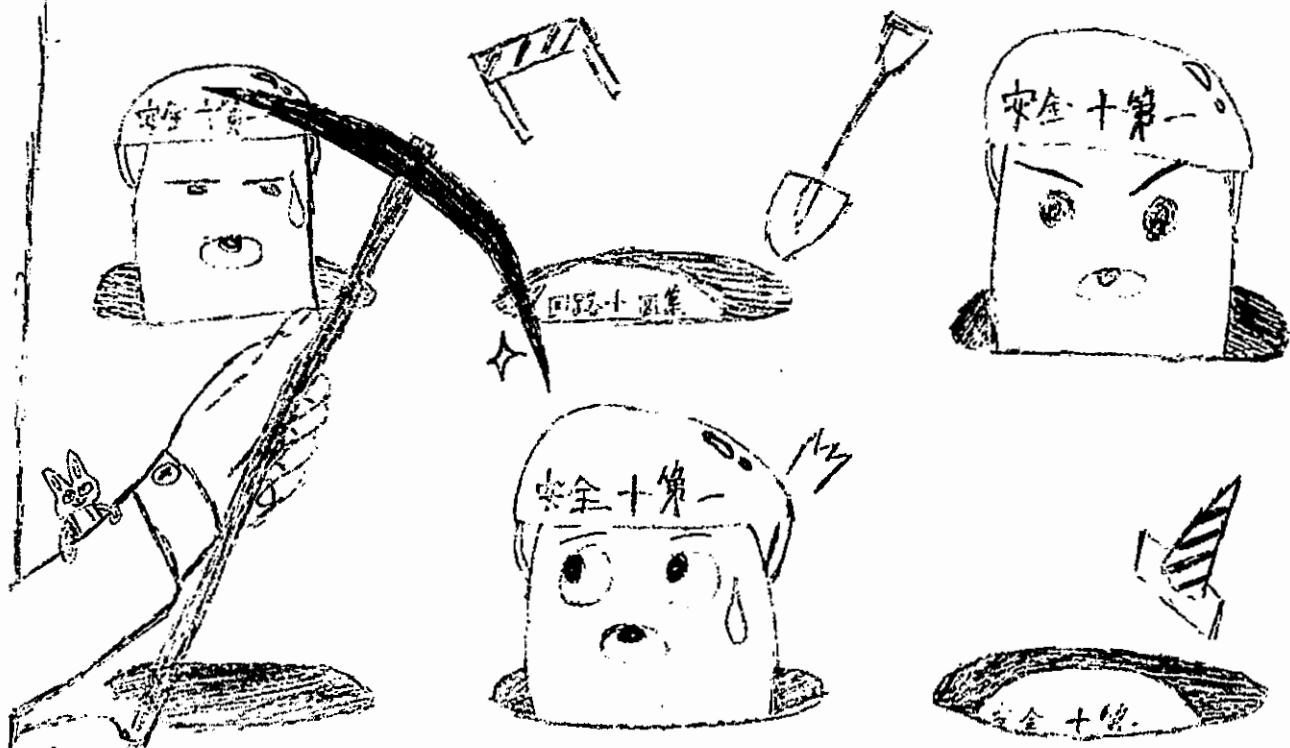
感想もなにも、全く完成していません。

アナログ系の先駆者(笑)として完成させる気は満タンですが、いかんせん配線量が多すぎて辛い現実と闘いながらやっています。

アナログ回路ってのは回路図通り組んでも動かないことがあるので厄介ですよ。ラジオ作った時もそうでした。これも二の舞にならないよう気をつけたいところです。



GAME



MOGLIRA
TATAKI

WARI



GAME TUBE

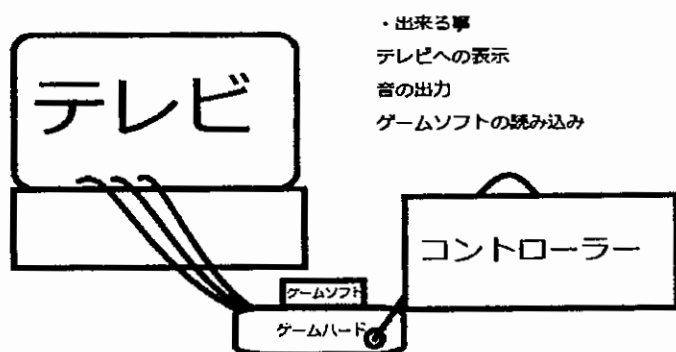
製作者：H2 山本 涼一

協力者：物無の皆様

1, 概要

どうもゲーム系なのに恒例のぐだぐだストーリーから始めなくていいのは日頃の行いがいいせいでしょうか。いつもあれ考えるために深夜まで原稿書くの待って、深夜テンションで乗り切ってたんですが今回はその必要は無さそうです。深夜に書いてるのは変わりませんがね。てか結局ぐだってますね。

はい、それはともかく早速本体の概要に移りたいと思います。今回は一応構想としてメモリーカードを差し替えると違うゲームが楽しめるファミコンみたいなのをめざしていました。とりあえず機能の紹介として↓の簡単なイメージ図を。



これ図いるのかな・・・

① テレビへの表示

今回 NTSC のコンポジット信号で表示しています。解像度は 240×216 で色数は 8 色。液晶よりも表示が早いのでアクションとかシューティングとかに向いてるかも。

② 音の出力

方形波を出して音を出力しています。1ch しかないので BGM か効果音かどちらかにはなりますが、あのピコピコ音はファミコンみたいでなにか懐かしいのを感じます。自分ファミコン世代ではないですが。

③ ゲームソフトの読み込み

マイクロ SD からプログラムを読み込んでゲームができます (といいな)。個人的にはあのゲームでよくある「LOADING・・・」というのをやりたかっただけ。

2, 回路図

回路図という題ですが回路図はピンアサインをいじれば変えられるので掲載を割愛させて頂きます。その代わりに使った部品について説明したいと思います。

・PIC32mx440f512h

使っているマイコンです。去年使って味をしめ、性懲りもなくまた今年も使うことになりました。発振が 80MHz までなのでとても高速でデータ通信、コンポジット信号出力に適しているかなと思いいこれにしました。

・水晶発振子

周波数は 3.579545MHz です。(以後 3.58MHz) 後述しますがコンポジット信号の出力にこの周波数が必要です。後、さらに言えばオシレーターは精度が必要なため水晶発振子を使う必要があるそうです。自分はこれを 16 倍した 57.2727MHz で動かしています。

・RCA 端子

ご家庭でよく見るあの赤、黄色、白の 3 本のコードのメス側です。音と表示と 2 つテレビに出力しているので 2 つつけています。

3, コンポジット信号

コンポジット信号は今回の製作でテレビ表示に使った信号方式です。この章ではコンポジット信号について説明したいと思います。

コンポジット信号は同期信号、カラーバースト信号、映像信号などカラー表示に必要なデータを一本の線にまとめた信号で主に RCA 端子などが使われています。アナログ放送にも使われてましたし、その他テレビとの接続でご家庭でもよく目にすると思われます。

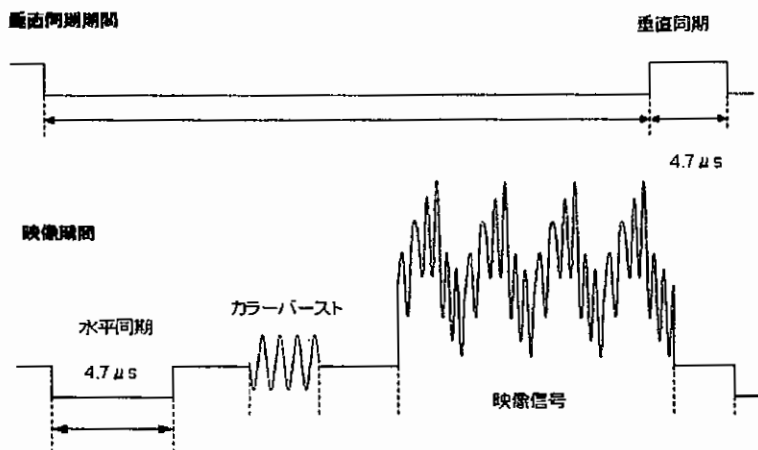
・コンポジット信号を出力するには

当たり前のことかもしれませんがコンポジット信号はテレビからの出力はなく映像データを出力する側が一方的に送りつけているだけです。なので、きっちりと定められたタイミングを守ってデータを送らないと表示されません。

簡単に下図に記していますが表示の流れだけ挙げながら説明したいと思います。もう少し詳しく知りたいという方は参考サイトがお勧めです。

・垂直同期期間

アナログテレビの表示は 60 分の 1 秒に一回と定められているので通信も 60 分の一秒に一回することになります。この表示一回の全面分の通信を 1 フィールドと呼び、1 フィールドは 262 ラインとなっています。最初 1 フィールドのはじめに 10 ラインほど垂直同期期間を取ります。下にも書いてありますが垂直同期期間は 4.7 μ s のパルスを出力します。



・映像期間

次に映像期間に入ります。映像期間の頭には映像信号を出力しない空白行を数行入れます。空白行といっても映像信号を出力しただけで水平同期信号とカラーバースト信号は出力する必要があるので、水平同期信号については4.7 μs出力、カラーバースト信号については振幅0.2の3.579545MHzの正弦波を9周期出力します。で、カラーバースト信号については更にあとで詳しく説明する予定ですが、簡単に言うとその後の映像信号で出力する色の信号の基準となる信号です。このカラーバースト信号がちゃんと出ていないと色が正しく表示されません。

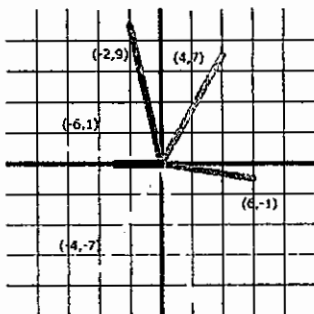
そしてこの空白行の後に映像信号を出力し始めます。映像信号は主に色差信号と輝度信号に分けられます。色差信号というのはRGBから輝度を引いた信号のことで、輝度信号というのは単に黒と白の明暗を分けるだけでなく、人間の目にも自然に見えるように緑は明るく、青は暗くというように調節するために必要な明るさの信号です。この2つによって映像信号はできています。さてこの色のデータですがこれらを一本の線で同時に送るにはどのような工夫が必要でしょうか。ただ単にそのまま足して送ってしまったら受け取り側(テレビ)でこの信号を元の2つの信号に戻すことができなくなってしまいます。これらの信号を一つにまとめて、かつ受け取り側で正しく2つの信号に戻せるようにするために送る前に「カラーサブキャリア」による変調しておく必要があります。具体的に方法としてはまずRGBから輝度信号と色差信号を割り出します。計算式は下記のとおりです。

$$Y = 0.587G + 0.114B + 0.299R$$

$$Cb \quad (B-Y) = -0.587G + 0.886B - 0.299R$$

$$Cr \quad (R-Y) = -0.587G - 0.114B + 0.701R$$

Y信号というのが輝度信号、RGBがそれぞれ赤、緑、青を表しています。Cb、Crというのは青、赤から輝度を引いたものです。この色差信号、CbとCrをx、y座標にとった色空間では色が全て点で表せます。原点からその色空間でとった点まで引いた線分は長さで表すことができます。つまりすべての色をこの色空間上での長さで角度で表すことができるということです。サブキャリアというのは3.58MHzの変調をかける波のことなので、カラーサブキャリアによる変調というのはこの色空間での長さで角度のデータをサブキャリアの波に乗せることをいいます。具体的には長さを波の振幅に角度を波の位相にすることで色ごとの波が作れそうです。こうして、変調してできた色差信号C信号と輝度信号Y信号をそれぞれ単純に足しあわせて映像信号はできています。そして受け取る側ではこの映像信号から3.58MHzの波だけを取り出すとC信号が得られ、3.58MHzの波を除去するとY信号が得られることとなります。やっとなしあわせても分離可能な波ができました。またカラーサブキャリアで変調することで信号を波にすることが出来るのでこれを放送用の電波で変調して飛ばせば遠く離れた場所にも信号を送ることができます。

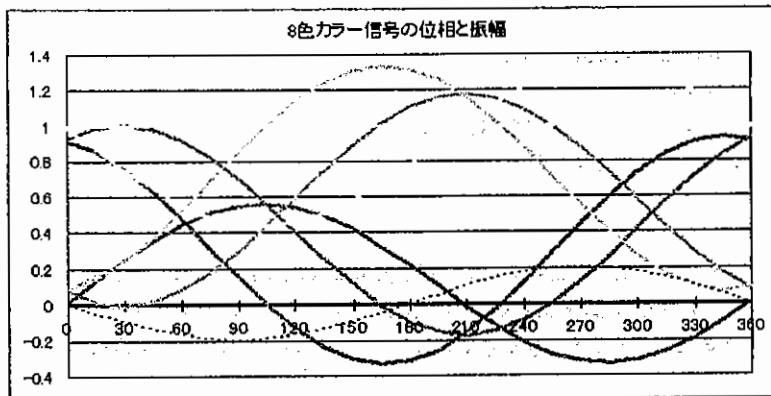


←色空間の図

長さは濃さを表しています。長ければ色が濃く、黒や白などの色のないのは長さがゼロなのでこの図で言うと(0,0)に当たります。左に伸びている黒い短い線がカラーバースト信号です。

↓変調した色差信号

色空間の長さで角度を振幅と位相に置き換えた図です。破線がカラーバースト信号です、



・カラーバースト信号

しかし、この波から3.58MHzだけ取り出すといっても基準が必要です。そこで出てくるのがさっき話に出たカラーバースト信号です。カラーバースト信号を送るとその信号をテレ

ピ側が記憶していて、その延長をずっと暗記し続けてくれます。忘れないようにずっと同じメロディーをくちずさんでるような感じです。それを使ってテレビ側は映像信号からRGBを割り出しています。カラーバースト信号は毎ラインごとに送り直すのでずれる心配もありません。

4, プログラム

さてこの章では前章で説明したことを実際にプログラムで表現するにはどうすればいいかというのを説明したいと思います。今回の製作では1フィールドの表示にはタイム2で60分1秒に1回割り込みを入れて、OC3で垂直同期信号、OC1で水平同期信号とカラーバースト、OC2で映像信号を出力しています。基本的に同期信号を出すところそのままH、Lを出せばいいんですが、カラーバースト信号や映像信号など正弦波を出さないとダメです。しかし、正弦波を出そうと思ったらとてもめんどくさい上に別に正弦波じゃなくてもある程度なら読みとってくれるので今回は90度に1回出力して3回出力で1ドットとしています。つまり1ドット270度でカラーサブキャリア4分の3周期です。精度よく表示しようと思ったら最低これくらいの出力は必要らしいです。

コンボジット信号の出力のプログラムの設計はケンケンさんのサイトを丸パクリ参考にさせて頂いています。

5, 参考サイト

・ <http://www.ze.em-net.ne.jp/~kenken/index.html>

趣味の電子工作とプログラミング ケンケンのホームページ様

様々などとも面白い製作をしている方です。どれも簡単で応用がしやすく作ってみたいという気になってしまいます。コンボジット信号出力と音出力、回路図をお借りしています。

・ <http://picavr.uunyan.com/index.html>

PIC AVR 工作室様

本当に多くのことをやっている方で説明がわかりやすいです。このサイトでコンボジット信号について勉強させて頂きました

6, 感想

なんだかんだ文化祭まで来てしまいました。3/21現状では32mx440f512hでの表示ができてなくて、すでに32mx120f032bで文化祭を臨もうとしている所です。メモリーカードの読み出しもできるころまで行きませんでしたし、当初の目標はぜんぜん達成出来ませんでした。最初の方はかなり早かったはずなんですけどまあ、来年はもっと早くに準備したいです。まあ来年ないんですけどね。というわけで終始ぐだぐだした物無生活でした。死んでも悔やみきれないものをいっぱい作ってしまった4年間でしたが、今はそういうものは全部そら辺の戸棚にでもおいておきましょう。最高学年ということもあって自分が作

りたいものがあつた時にすぐ先輩や他の人に聞けるような環境を物無に作りたと思つていたのですが少しは技術的なことが共有できるようになつたんじゃないかと思つます。自分としても今年はかなり製作とかプログラミングとかに興味があつた一年だつたので、出来れば引退後も製作は続けたいです。いつの日か変なもの作つてる人見つけたと思つたら山本だつたみたいになつたらいいなと思つます。

7, オマケ

BIRDS DAY

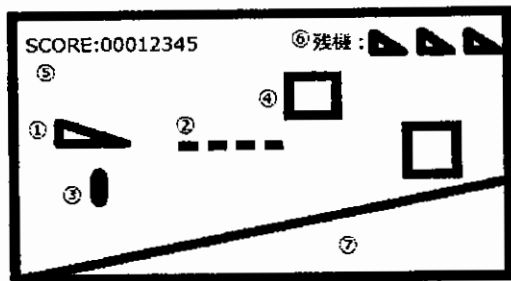
あの世界を股にかける男達が帰つてきた!!・・・鳥に乗つて!!

・ストーリー

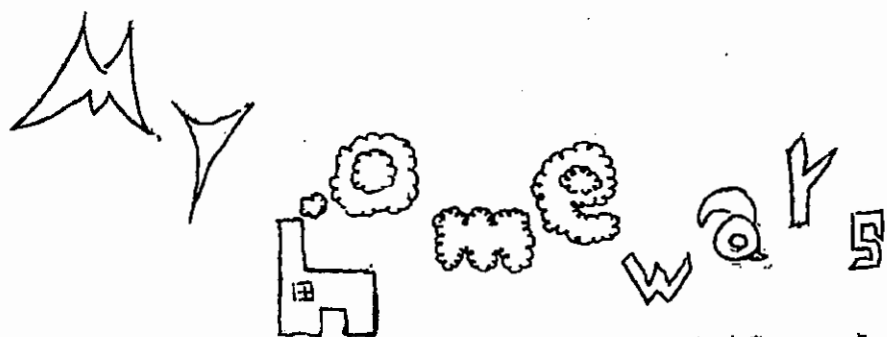
19XX年、十数年前の戦争はすでに過去のものとなり世界は平穏に進歩を続けていた。そんな中戦場にしか身をおくことのできない一人の男が未開の地アマゾンを目指していた。彼の目的はアマゾンの奥地に眠るといわれるピンク色のひよこ通称「はやロク」。明日誕生日を迎える息子にせがまれたのだ。さて、彼はアマゾンに眠る強敵たちを倒し、念願のはやロクを手に入れることはできるのか?それとも自然の脅威にあつさり負けてしまうのか?相棒の大鳥にのつて彼は今日も空を駆る!

・ゲーム概要

ハイ、オマケです。このゲームは一応本体でプレイできるゲームとなっております。アーケードみたいな普通のシューティングゲームです。パロパロパロディウスくらいになるんでしょうか。全三面構成で面の最後にボスが出てきます。一応イメージ図です↓

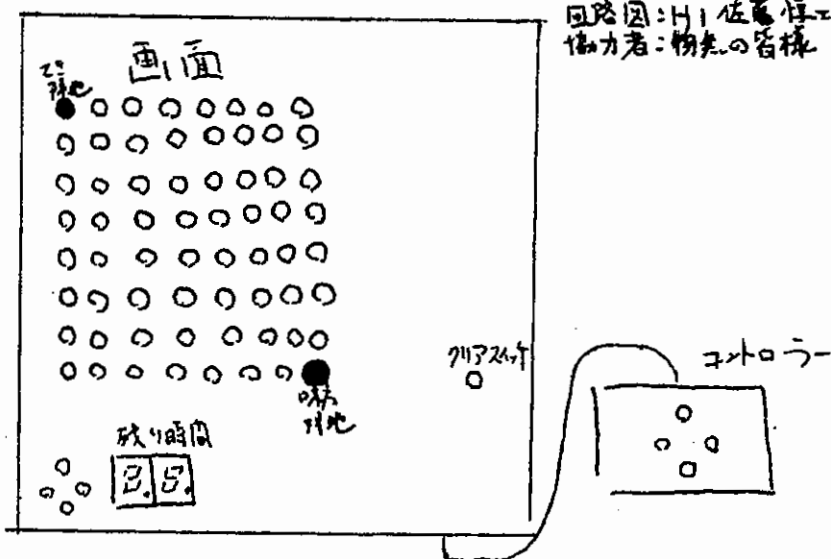


- ① 自機：縦横無尽に動いて敵を倒してください。
- ② Aショット：前方に進む威力は弱いが連射出来る射撃です。
- ③ Bショット：下方に進む威力は高いが一発ずつしか打てない射撃です。
- ④ 敵：敵も射撃をしてきます。できるだけ避けて敵を倒してください。
- ⑤ スコア：点数です。これが最終的な評価になります。
- ⑥ 残機：残り機数を表します。三回死ねるね。
- ⑦ 地形：当たると死にます。避けてください。



・外観

制作者: M2 磯村隆正
 回路図: H1 佐藤 博之
 協力者: 物売の管様



・ルール

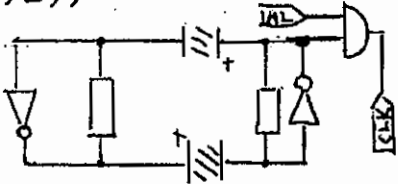
- 敵(赤)が敵陣地から出てくるので、自機(緑)に体当たりします。
- 体当たりすると、敵一列と自機が、消えます。自機はまた味方陣地に飛ぶので、同じことを繰り返します。
- 敵陣地まで自機を移動させれば勝利、味方陣地に敵が来るまたは、時間終了で敗北となります。
- 自機の移動範囲…画面全体。ただし、緑の部分に入ると敵が見えない。
- 敵の移動範囲…画面の左右。緑の部分には入れない。

・ストーリー

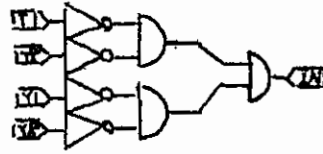
※シュールな話ではありません

2013年夏 家にゴキブリが大量発生!
 ゴキブリを撃退して、筆筒のむこうの奥にたまたまつづら!

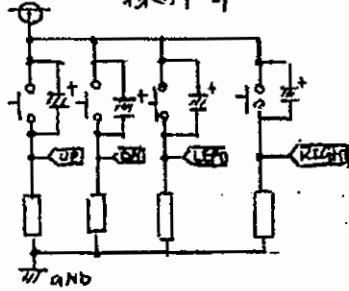
回路
ブロック



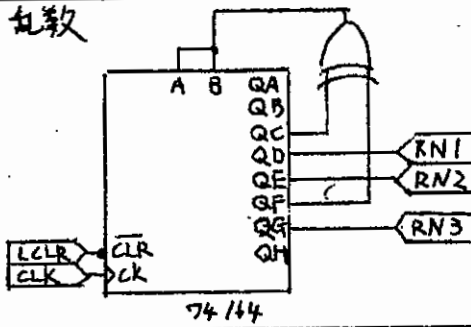
プレイヤー位置判定



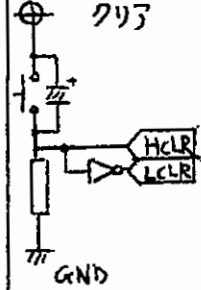
操作部



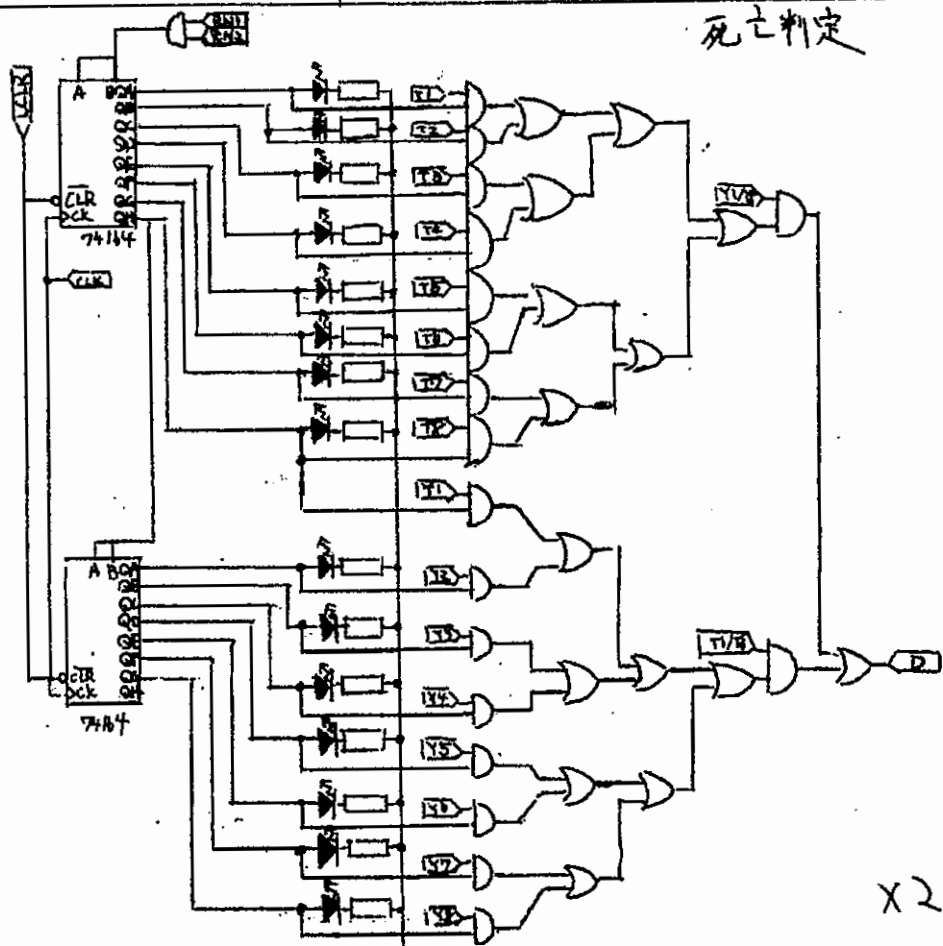
乱数



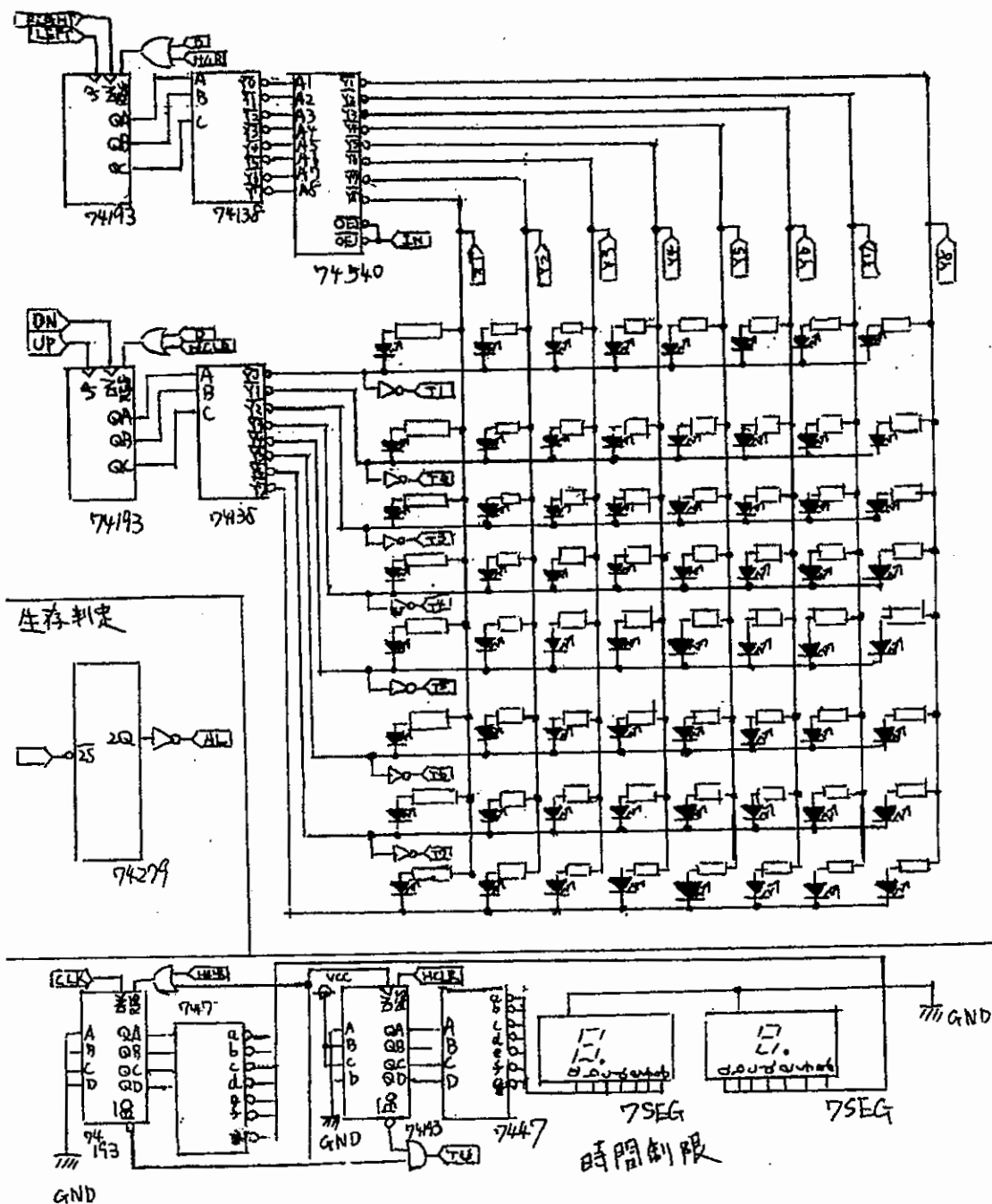
クリ



死亡判定



X2

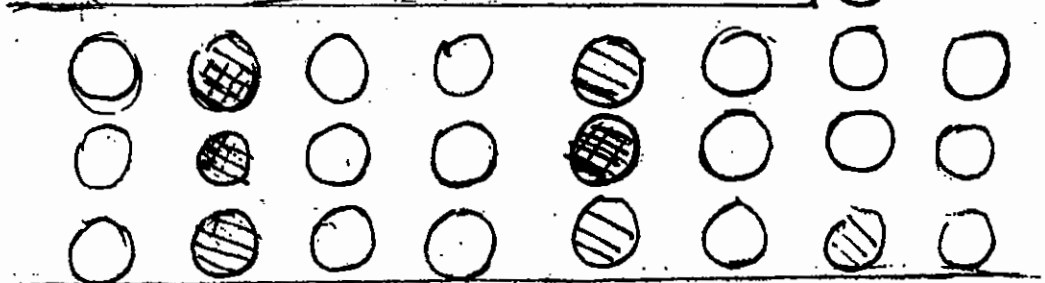


感相
心心

最初は回路の意味をよく考えず、ひらすら作業をするという感じ
でしたが、後半からはきちんと考えで作れるようになった。しかし、
回路ミスや接触不良が多く、エラーやバグが頻繁に発生し、後半の出席
率が上がらないこともあり、思うように動いていません。来年は最初からきちんと考えたことと
思っています。

走超跳躍役

M2
伊藤 楓
柳 さん



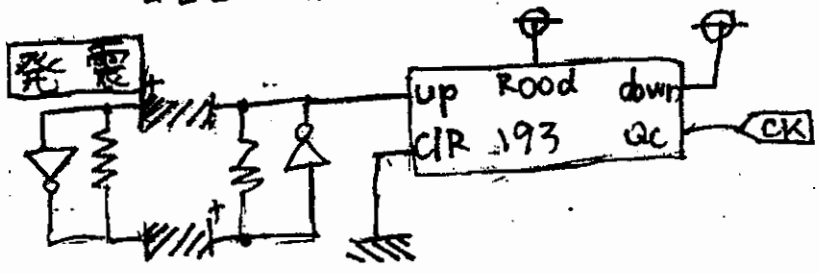
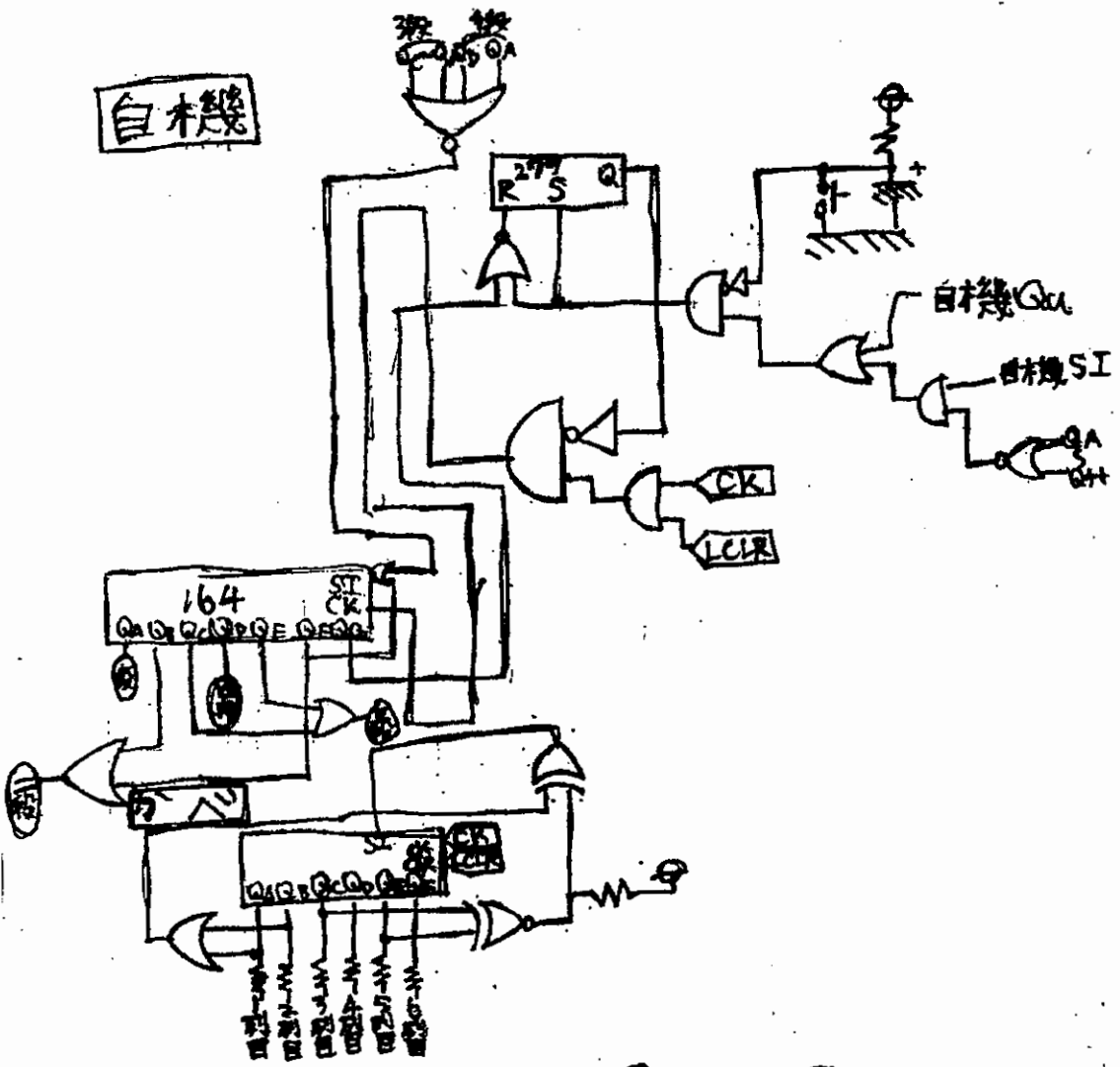
ルール

せまっ てくるかべをとんでよける
ゲーム。かべは2段と3段の2種
類で、自機移動は3段ジャンプと
4段ジャンプの2種類。

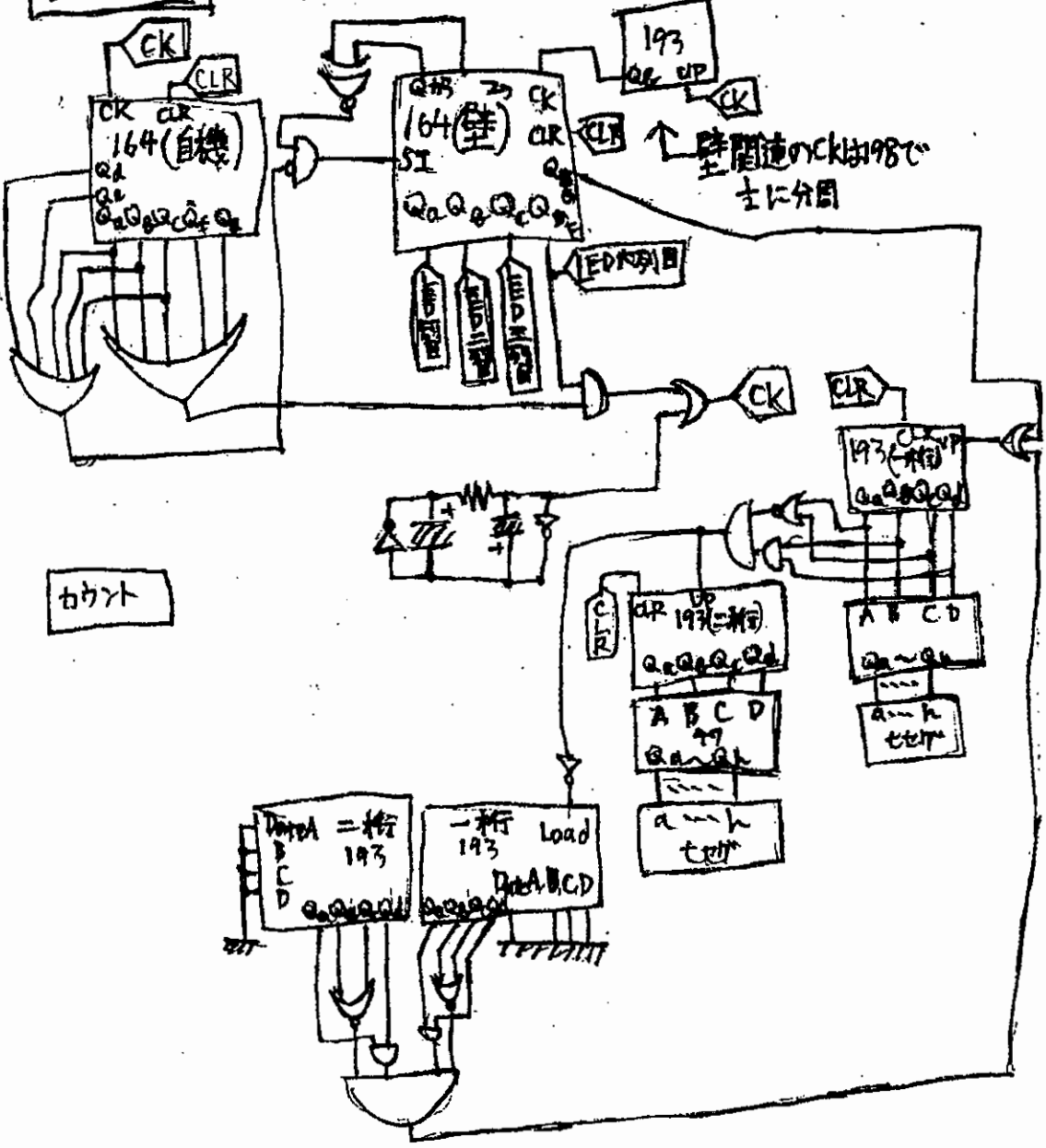
設定

A君が道をふつうに歩いている時、
なぜかかべが飛んできた！
A君は走り高とびの選手であるた
め、かべをとびこえることは簡単
だが、かべは次々とせまってくる。
A君になつて、かべをとびこえま
くろう！

自機



当たり判定



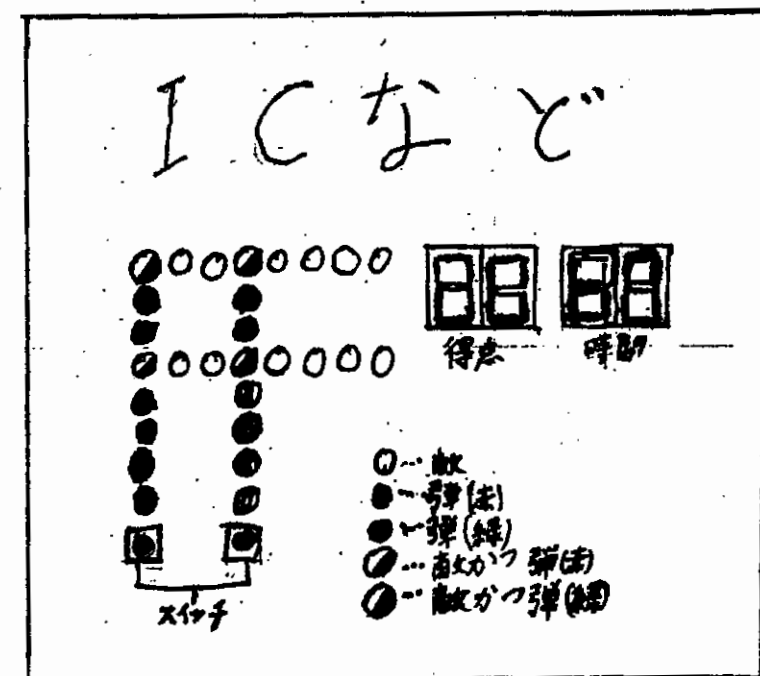
あとがき

3月に入っても自機が動かなくて、文化祭までに完動できるか絶望的な状況でしたが、最近ようやく自機が動いたため、もしかしたら完動できなくもないかもしれないという感じですか。ぼくのゲームは、自機が一番難しくて他のものは結構単純なので、これからは陸上部が終わり、た後物無に来て製作をしたいと思います。

猛獣への えさ投げ

製作
藤本博之助
監修
HI 四柳さん
協力
物無の皆様

○外観



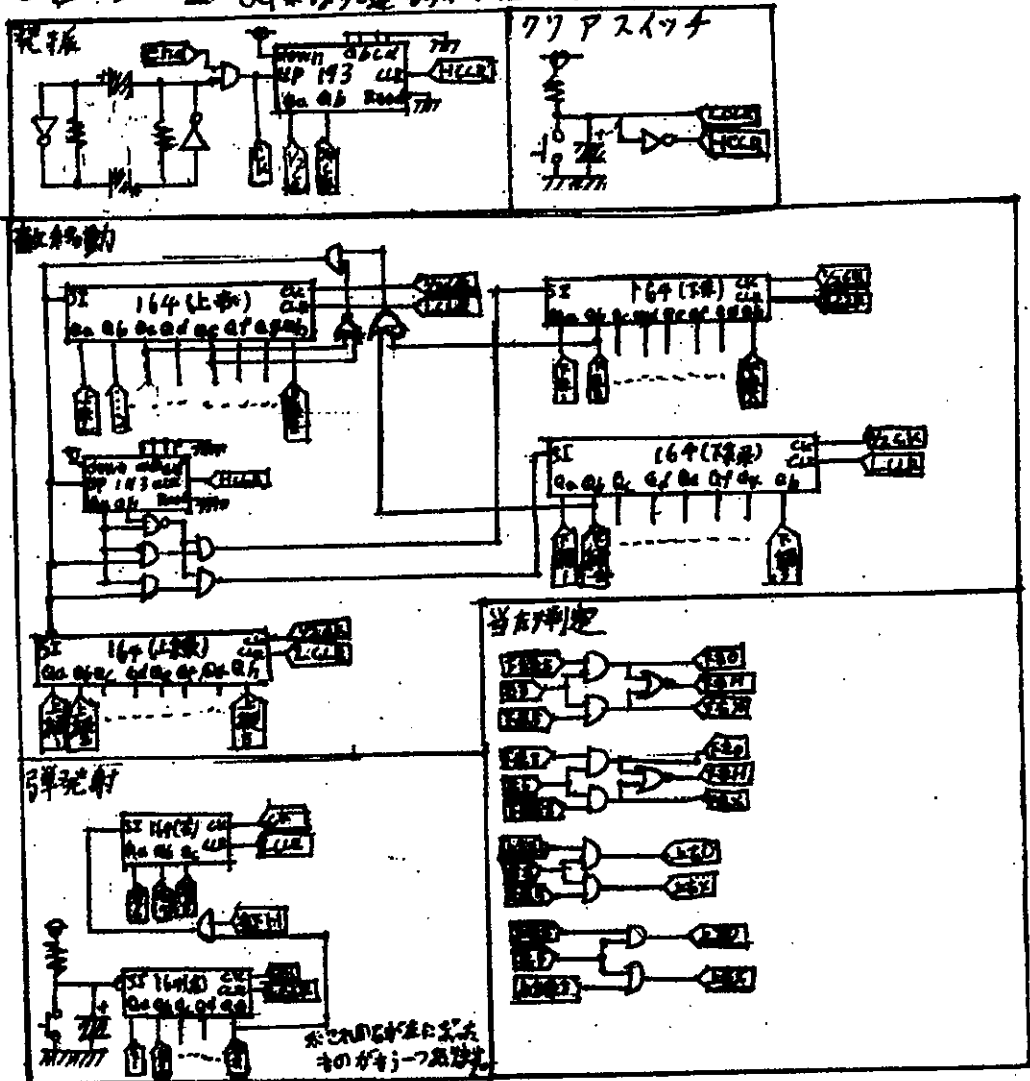
○ルール

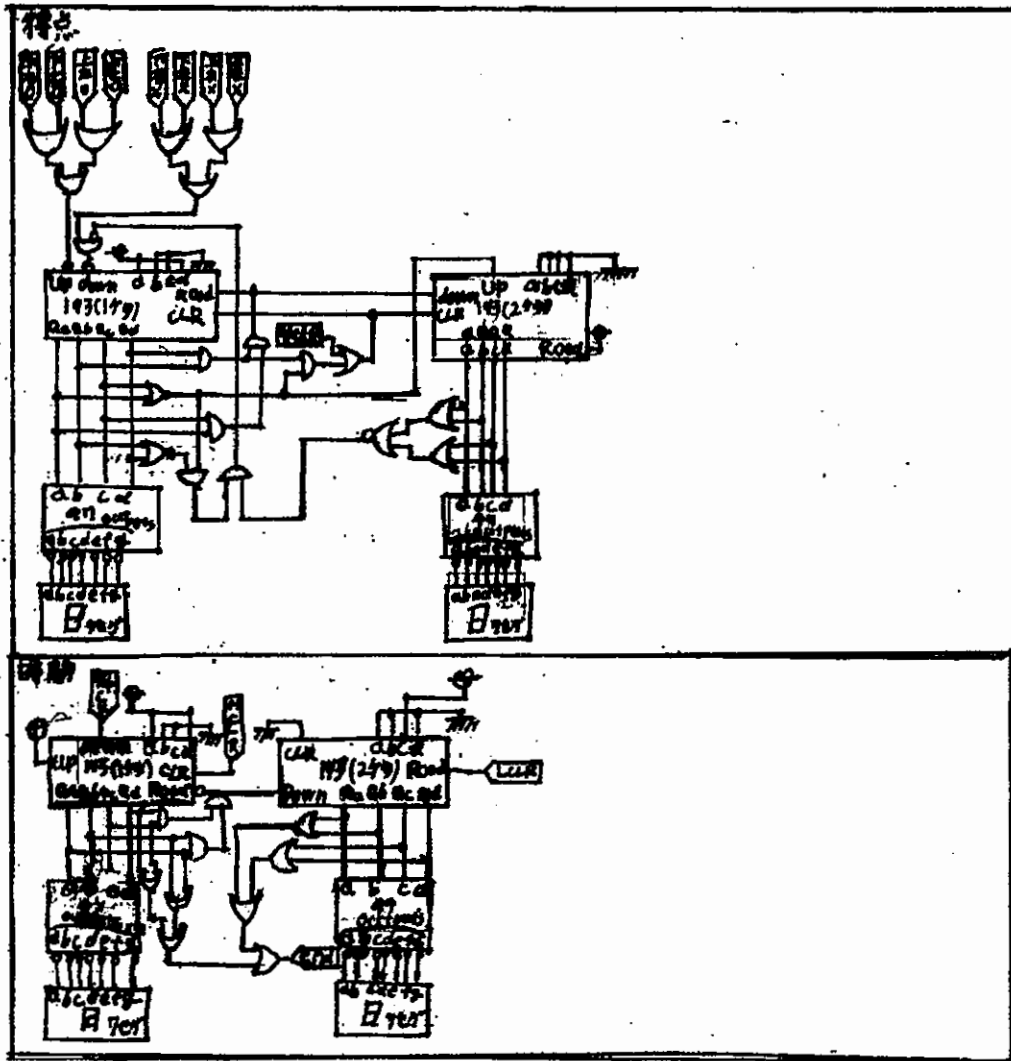
横2列に、右から左へ赤と緑の敵が流れてきます。スイッチを押すとスイッチの色と同じ色の弾が発射されるので、同じ色同じの敵と弾を当ててください。同じ色同じを当てると+1点で、違う色同じを当てると-1点です。また、敵とぶつかった弾は色に関係なく消えます。時間が尽きたらゲーム終了です。(スイッチはいつでも押さないと弾が出ない場合があつます。)

○ ストリーター

こゝは様々管理センター。そこに1人の新米の飼育員がいた。彼は2種類の猿のえさヤクを頼まれたのだが、その2種類の猿の好みが真逆で、嫌いなものを流すはずがつけられないほど。登ってしまい、おりの中にいたが殺さずにはおかしくない。なので、えさヤクをしようと思っていえおかしすぎ、悩んでいたら、おりの外から投げればいいんだ。」 ※この話はフィクションです。

○ 回路図 (実際は少し違いがありますが、問題はありません)





感想

僕は配線材をむくのが下手で、むいたら中の線が二本しか残って
 いないということがありました。今は少しは上手になりましたが今でもたまに
 むくのは失敗する事が多いです。まあ、むくの作業は作業しただけを見たり
 下手なせいでおどろきました。
 製作では年計なものをつけたせいでバグが起きるようになりました。
 来年は、なるべく年計なものを作らず、バグを減らしたいです。

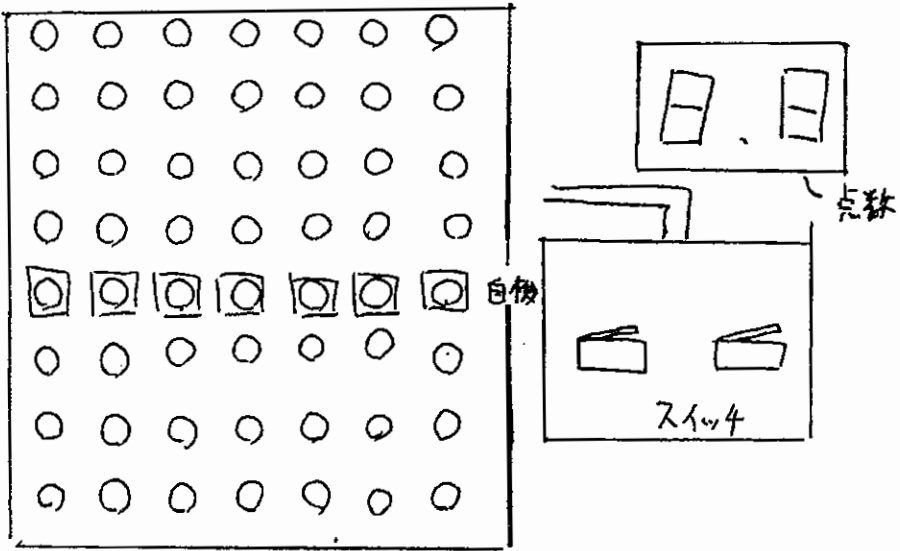
製作者 M2 佐藤

激突! グル×レース

ルール

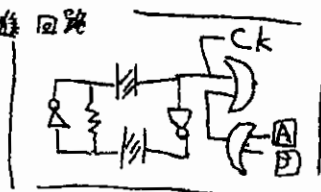
赤いLEDはと、てしまうとゲームオーバーにな。てしま。
緑のLEDをとると点数が増えていく。
白旗は左右に動き、時期は無制限、君は何点で夫子か...

外観

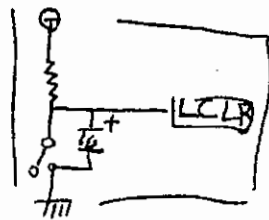


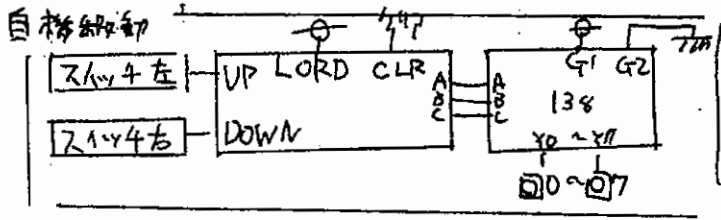
回路図

点灯回路

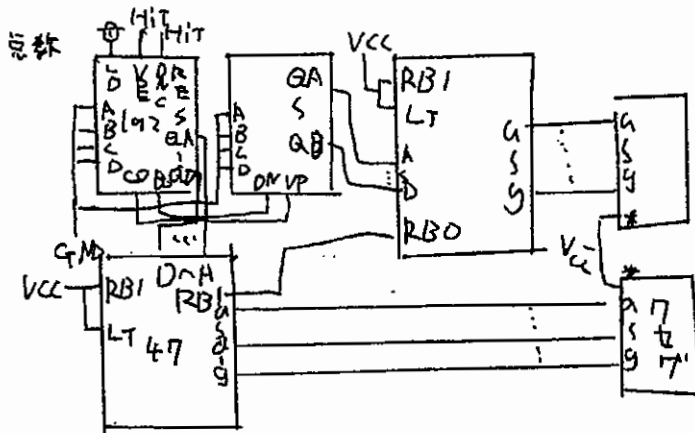
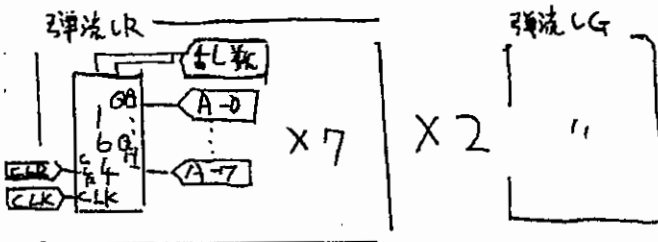
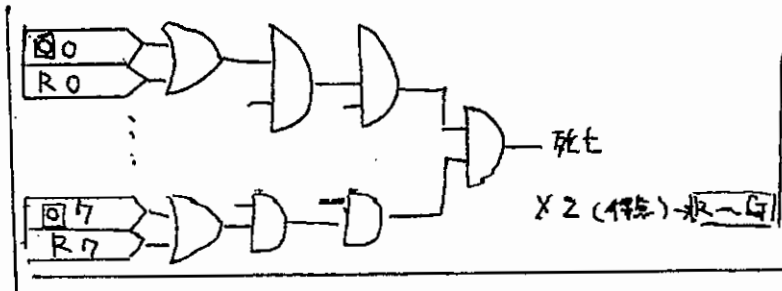


LCLR





死亡判定と得点判定

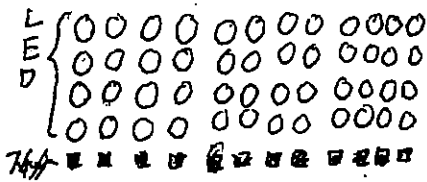


感想

製作の途中で、3人たハブニングがめつたし、せよふりょうめつた。たつたで、したか3番にたつた。本当に協力してこられた方に感謝申し上げます。

ピアノの達人

外観～



時間

得点

製作 M2
回路図 HI
協力

関根史人

佐藤さん

物無の皆様

ルール～

一段目が降ってきた光っている部分が四段目についたときに、真下のスイッチを押します。このとき、LEDは1列で最大2個光っているので同時に押すと得点が1点引かれます。時間が99になると終了です。

ストーリー～

ここはある音楽インクルの裏側。

「何だ、音面がないだもッ！」

演奏開始まで残り30分を知ら

「もう...おしまいだ...」

ガチャ

ドアの開く音がした。

「先生ッ、これを見せてください」

「もういいんだよ...ん」

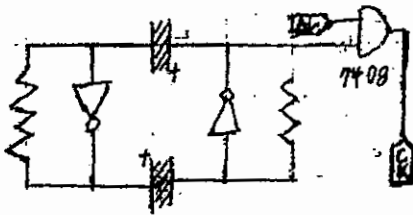
彼の目に浮かぶのは無数の空間が流れ、落ちてゆく無数の希望の光ッ！」

「ま、まさか...音面を自動で生成しているのかッ！」

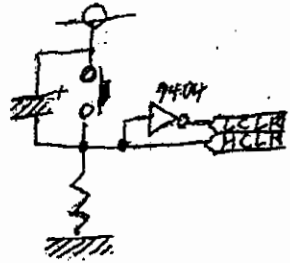
「これだ、俺は演奏で来るッ！」

そして、彼は一枚の鍵盤を片手にピアノへと歩いていった。

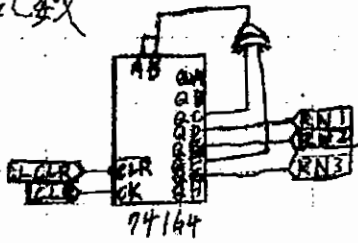
回路
クロック



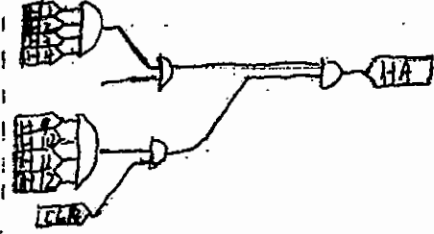
出力



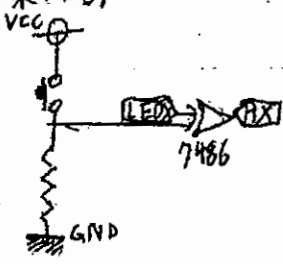
乱数



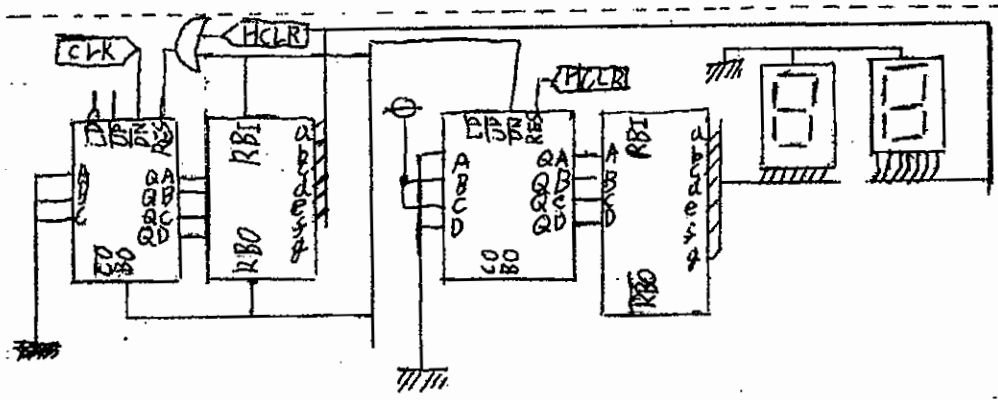
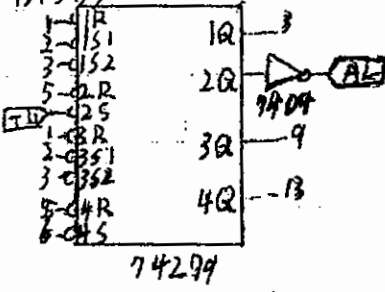
成否判定

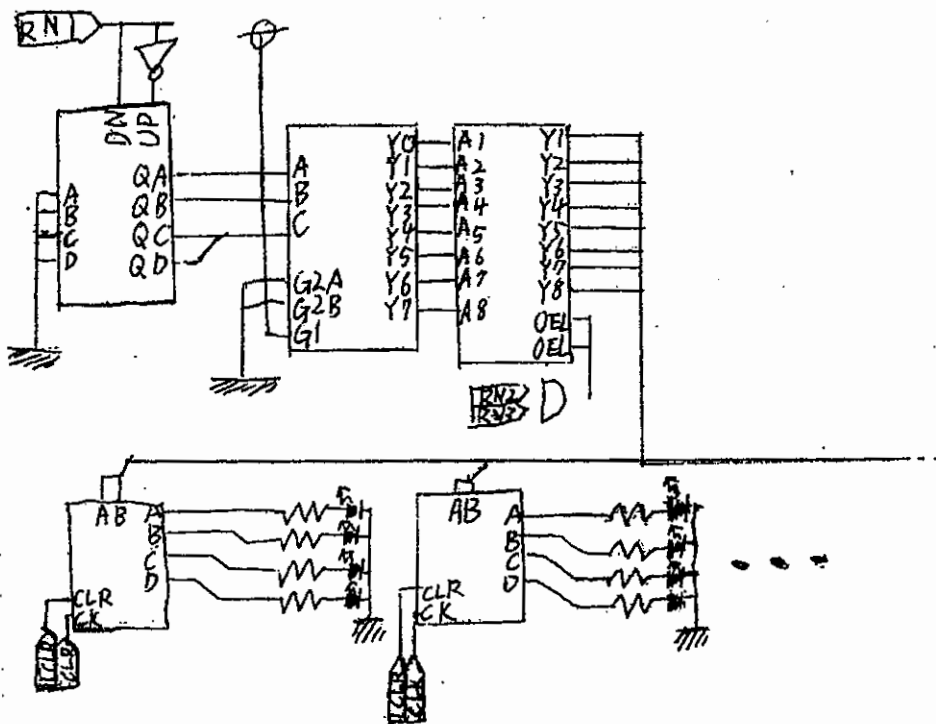


操作部



74299





感想

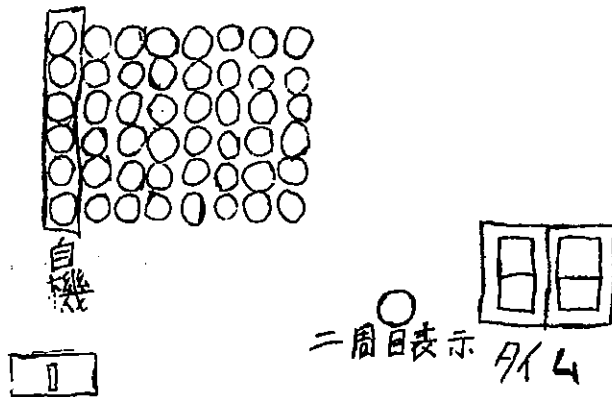
製作途中で焦りすぎて逆向きに電流を流し、約12個のICを取り替えることになったので、とても反省しています。

しかし、はんだ技術が最初とくらべてかなり上達したので良かったと思います。

SKY WALKER

制作者: M2 中村 航
 回路図設計者: H1 花園 十人
 協力者: 物無の 目 さま

一. 外観



自機移動スイッチ

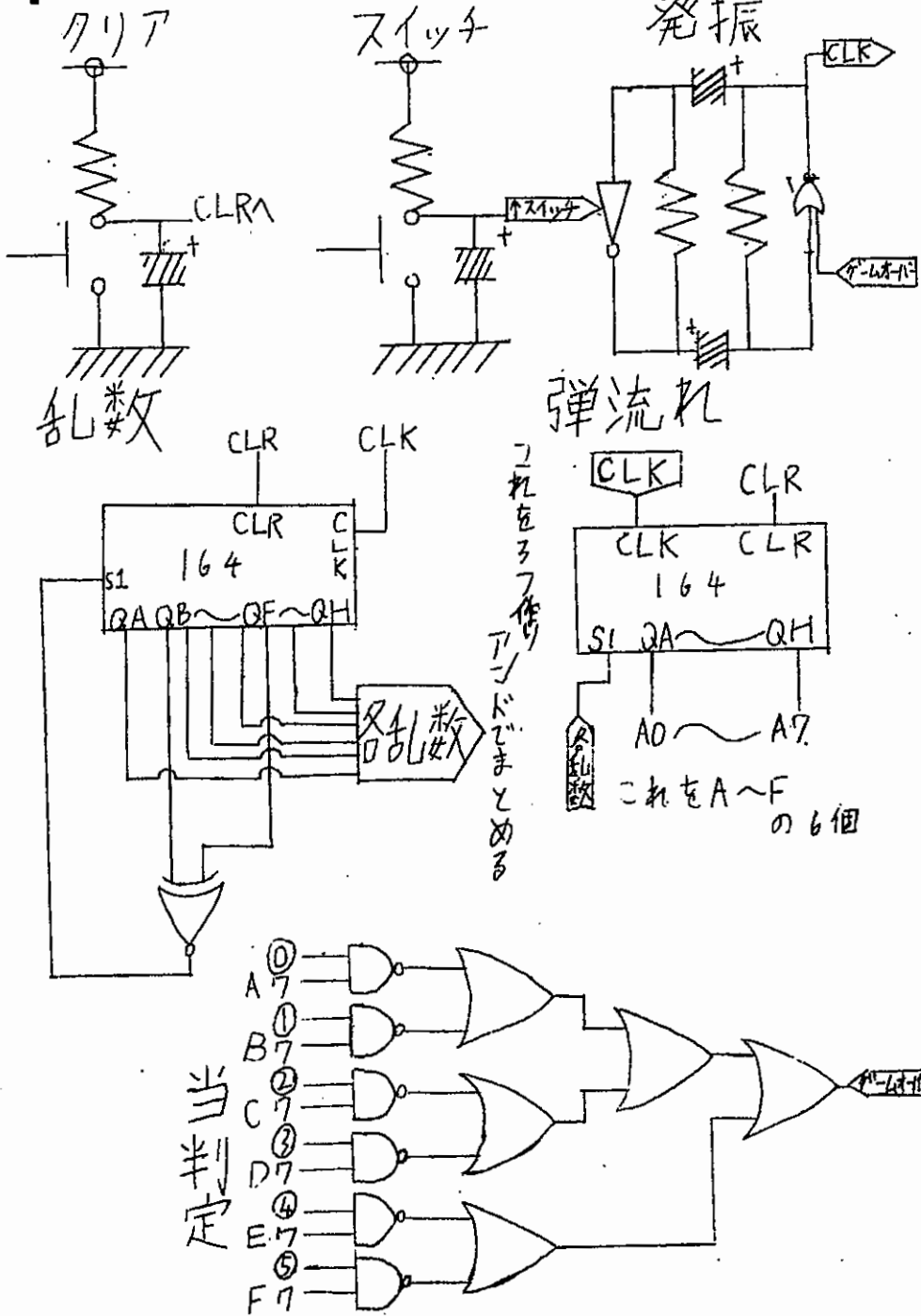
二. ルール

右からくる赤の光を緑の自機を動かしてよけるゲーム
 自機は一番下にいない限り少しずつ落ちてくる。
 自機の初期位置は一番下でスイッチを押すと一つ上の段に上がる。

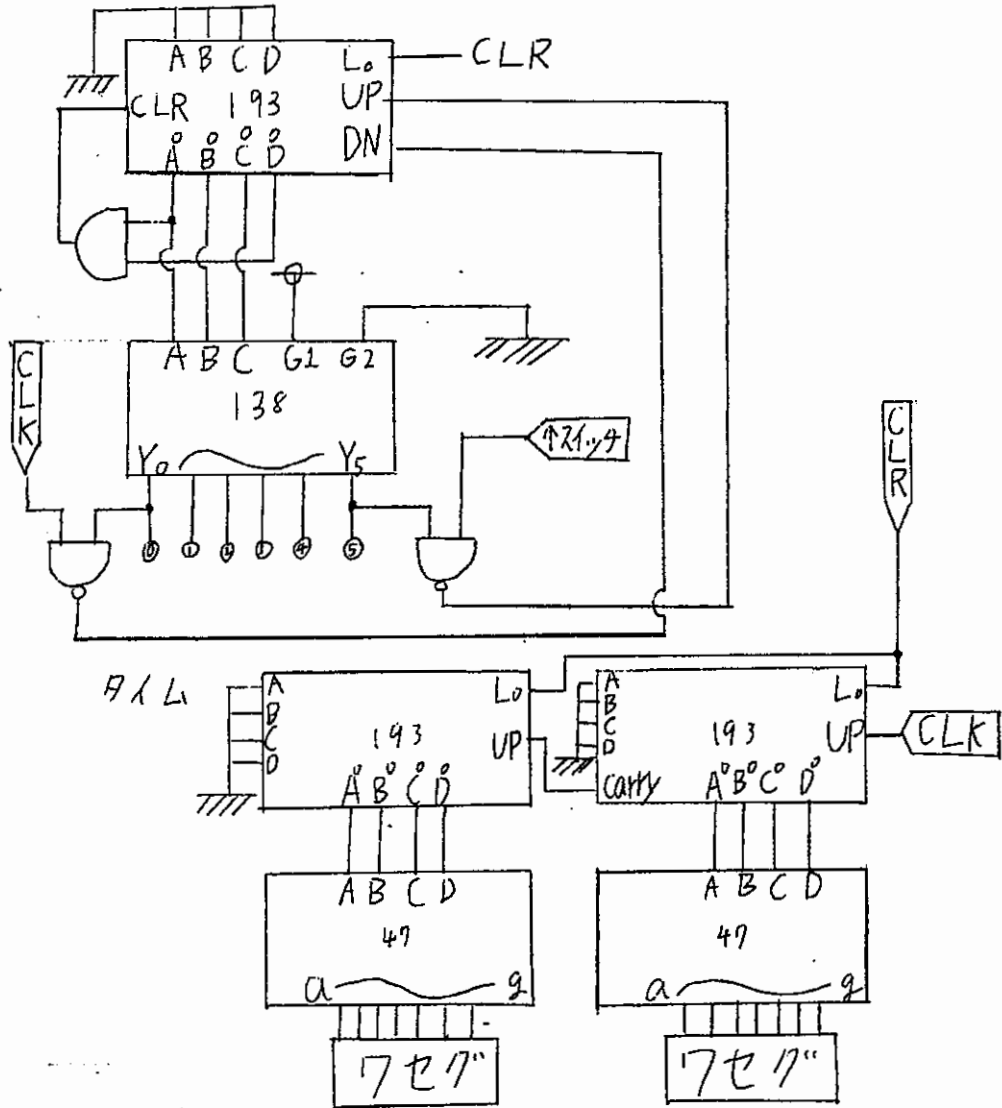
三. ストーリー

ここは口△星である。ここでは空を歩く人間が文明を築いていた。
 そして今日は口△星の祭りの日広場の中心ではパイよけ大会が行われていた。君も挑戦しよう!!
 Are you ready? オ——!!!!

四. 回路



自機



五. 感想

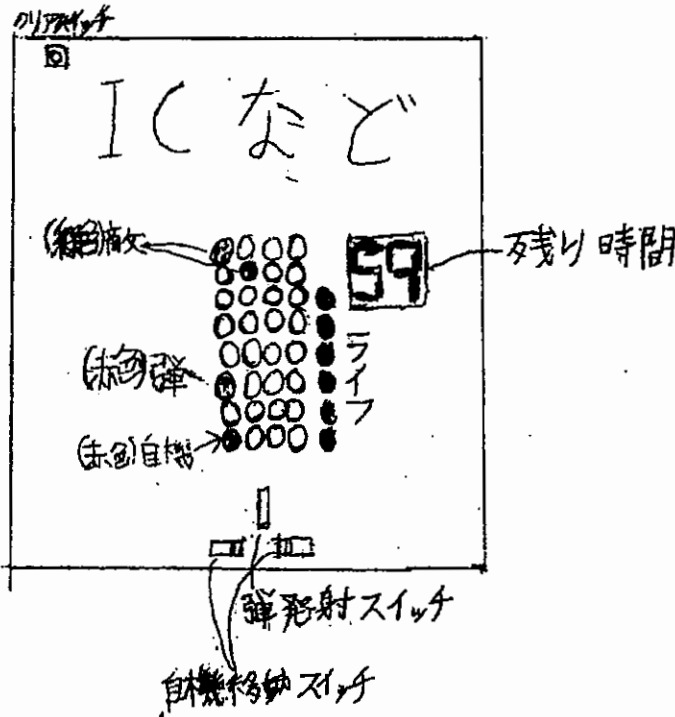
横から流れてくる弾を少なくするための配線とLED
 につなげる配線は多くて疲れました。
 一番こつた部分は基盤の形をアーケード的なものに
 したところですね。

ジグザグマ を追え

回路図設計者さん
高工 花園さん
協力 物無の皆さん
制作 中2 福島 弘也

ルール
上の4列のLEDから敵(緑色LED)が流れてきます。その敵に弾発射スイッチを押してまっすぐに流れる弾を自機発射スイッチで発射してきてください。また敵が一番下の段にいくと、ライフが一つ減り、ライフがなくなるとゲームオーバーとなります。残り時間が00になるとゲーム終了となります。

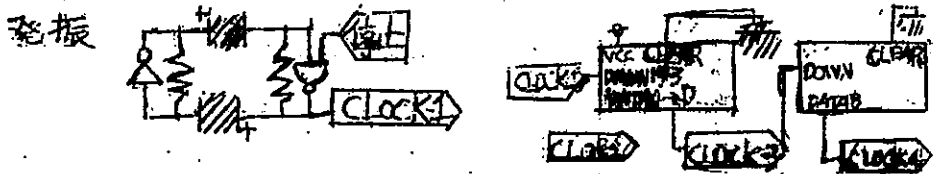
外観



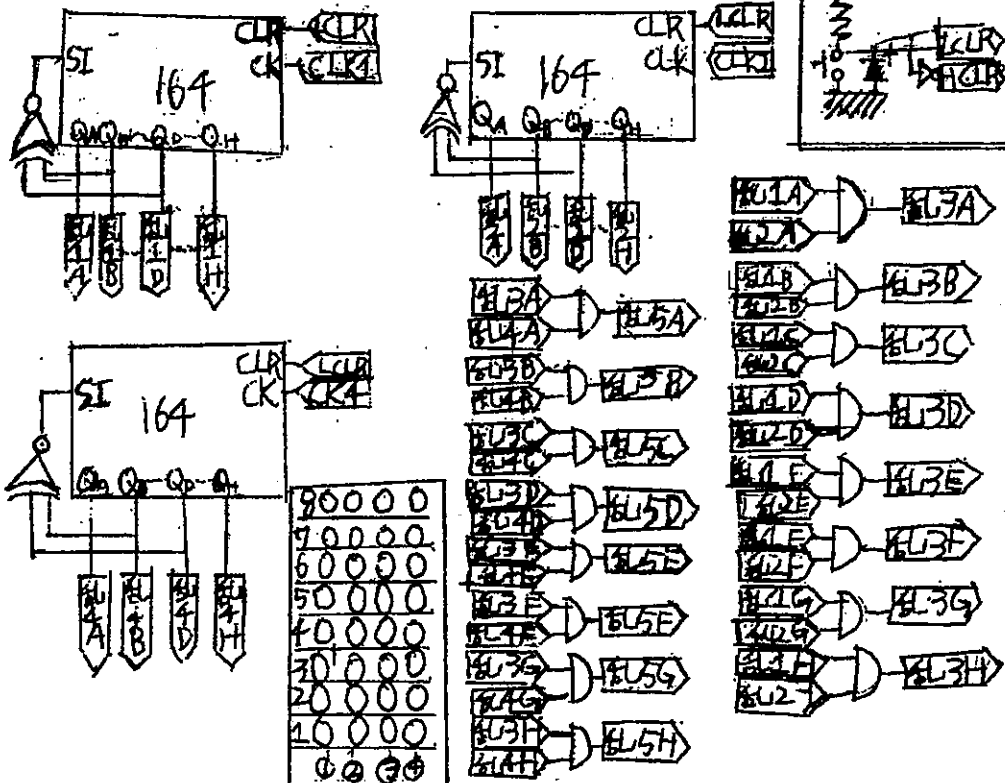
0ストーリー

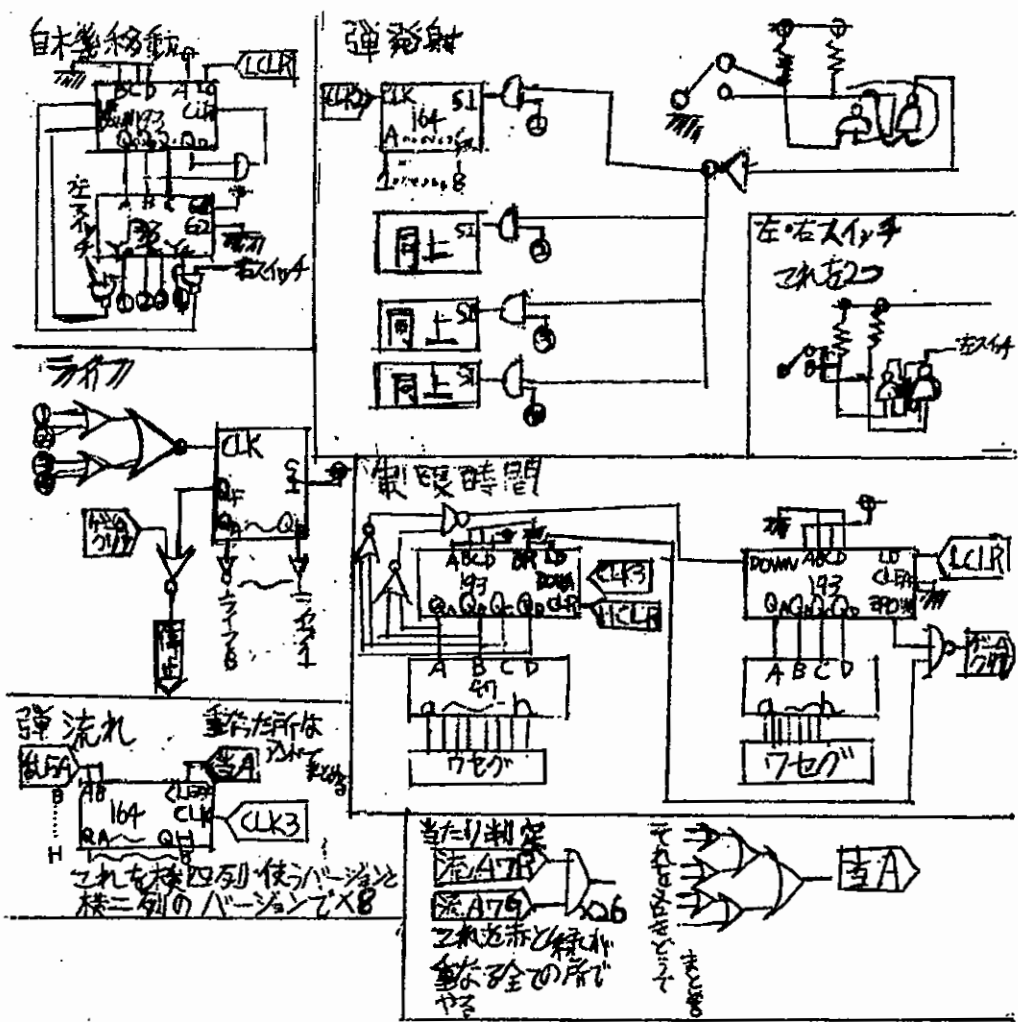
突い、まっすぐな人間になりたくなった君は
 マスガマを捕まえようと思った。しかし、野生
 のマスガマはいなくなりました。ため
 シガサガマを捕まえてLVアップさせなくては
 ならない。そこで草むらに行き向かってくるシガサガマ
 を捕まえなくてはならない。

○回路図 (実際の回路と少し違う可能性があります)



乱数





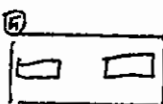
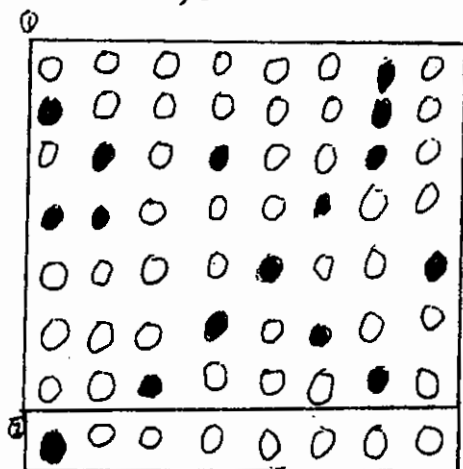
感想

初めての制作でまさかしかたですか。
 楽しかったです。あともう少しで完成しそうなので
 が大はります。

メテオ龍巻

制作: M2 和田
 回路図: HI 黒田 さん
 協力: 物無の皆さん

1 外観図



- ① この一番上から弾(メテオ)が落ちてくる。
- ② この列を自機が重かる。
- ③ ライフ二つある。
- ④ 時間制限。
- ⑤ コントローラー。左や右に自機を重かる。

2 ルール

① 一番上から弾が落ちてくるのでそれをコントローラーを馬鹿使して弾を60秒間逃げきれればクリア。
 ライフは2つあり弾に2回あたると死んでしまうのでよけよう。

3 ストーリー

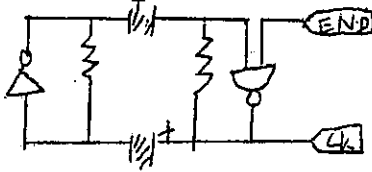
20XX年 遠い銀河の果てからメテオがや、てきた
 そのメテオはなんと、A君が住む日本の東京着陸に降るといふ
 しかしい回ぐらいいは超最先端技術をもついまの人間なら
 防ぐことができる。
 さて、A君はメテオが降るといふ魔の60秒間の間2回
 以上メテオに当たらずに逃げることができるのか。

4 感想

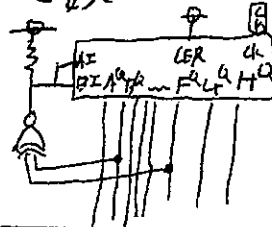
いままでは電子工作というものを一回もやったことがなかったの
うまくできずがどうかしても不安だった。しかし物販会にしろ 右しんの人
たちは手伝ってもらえたので、いろいろできたと感じる。しかし羊田ブツ
がうまくすることができなかったの、少ししゃしいと思った。

5 回路図

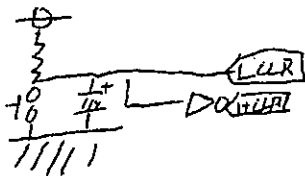
① 発振



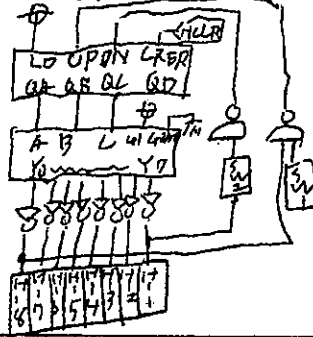
② 乱数



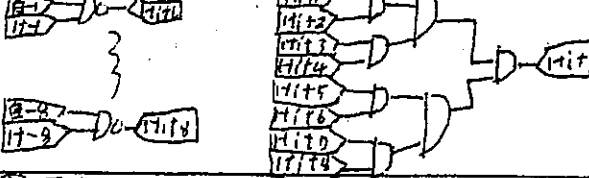
③ LLR



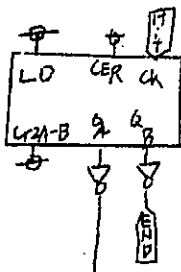
④ 自木線移動重力



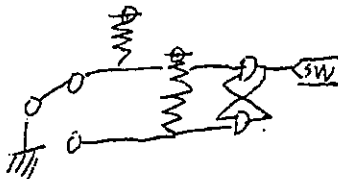
⑤ あたり判定



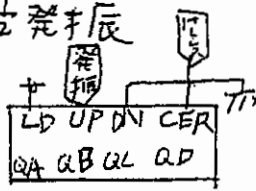
⑥ ライフ



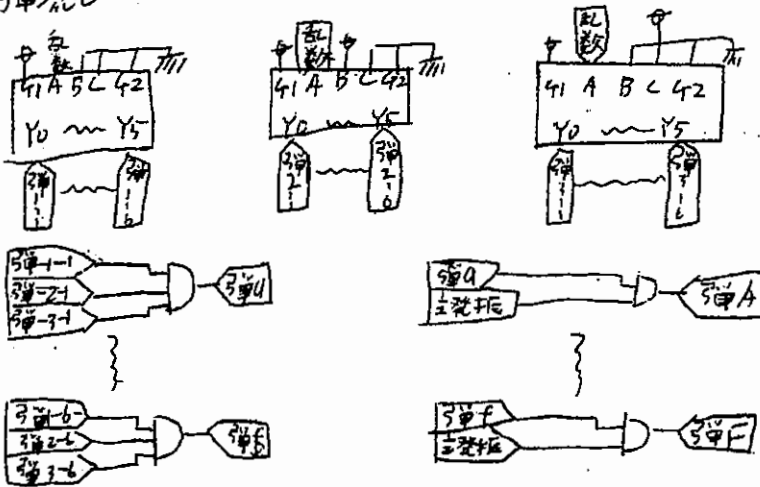
⑦ SW回路x2



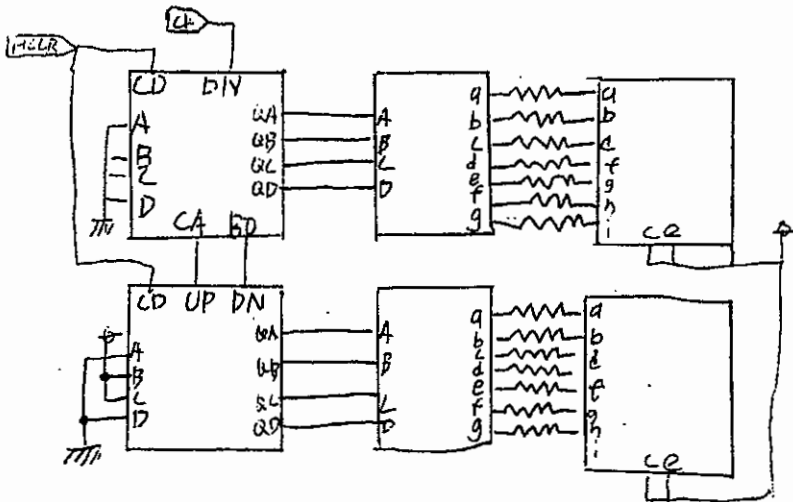
⑧ 主発振



⑨ 引当流し



⑩ 時間制限(60秒)



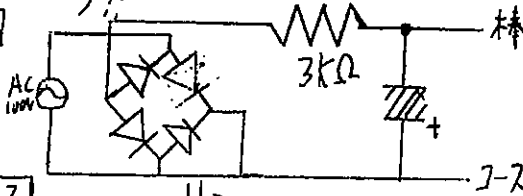


やり方

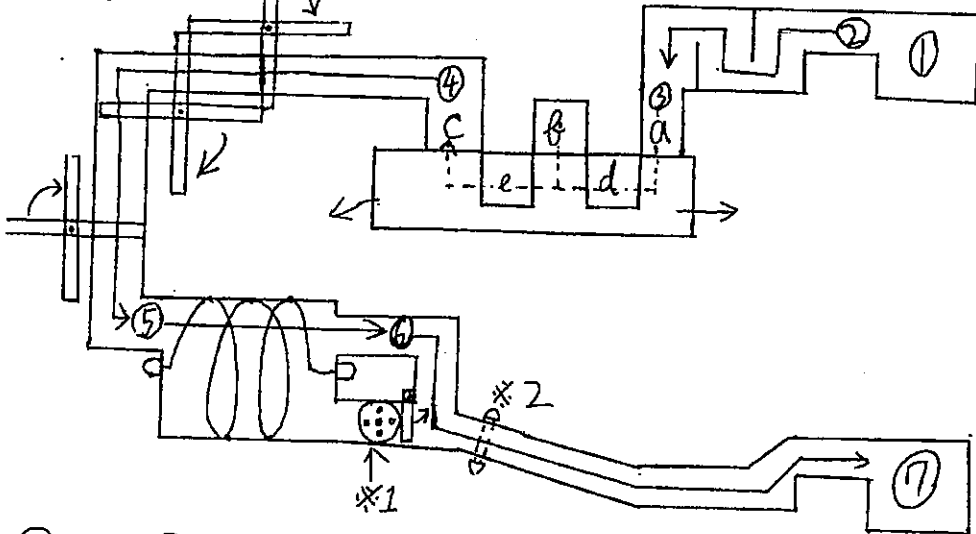
金属棒を金属部分に当たらないように金属の閉を通りぬかせるゲームです。

もし、金属部分が当たった... (当たらないでゴールまでいく勇者は多くはない) フラッシュして火花をちぎります!!
次々と現れる障害物を乗り越え、ゴールへと進んで行こう!!

回路



コース



① スタート

イライア棒の出発地点周りは木で作られているので、まあ、まあ大丈夫安心してスタートしよう!!

② ねじ

なんと... 太いネジが二本出た...
ゆ、くりこいねいにやれば心配ありません。動きませんからね。

③ 重く床

a → d → e → 2 → c の順に移動しましょう。右左に動く金属板の動きにびたりあわせないとダメです。今回のイライラ棒びー、ニを争う難しさをね。

④ 風車

上から下への移動地点。しかし、ただでは通してくれません。風車の間をうまく通りぬきましょう。③よりは簡単なはず。

⑤ バネ

床屋さんの赤と青と白のくるくる回ってる木のようなものです。線と線の間に入れ続ければだんだん右へ移動ができます。

⑥ The インディー

あの有名なあれです。そう、あれそれだよ!! ※2のセンサーに反応して※1の円板が回転しだします。スピードが大事です。しかし最後に上に上がるなければアウトです。気を付けましょう!

⑦ ジョール

おめでとら!!
ここまじくしたのは2,3人しかいない。

感想

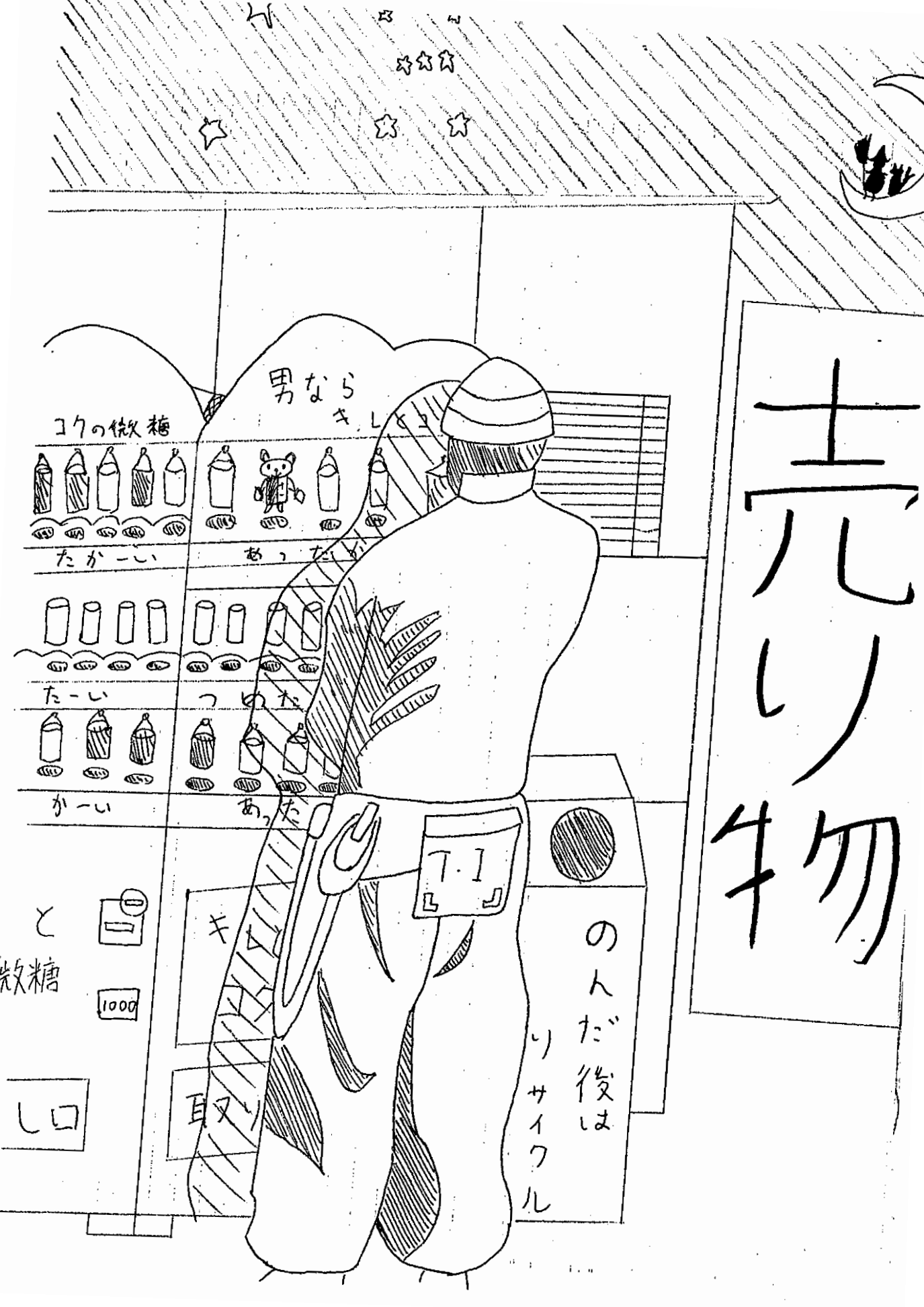
関根

今年は第一完動が1ヶ月ほど遅く、その遅れを取り戻そうとプログラミングを終わらせたのですが、また遅れたままで終わる気がしながら、たのびすが、⑥のギミックが何とかかなりそうだったの良かったです。

プログラミングは楽しかったの、やはり回路を選んじ良かったです。

中村

ぼくは本体担当でした。木を切断する作業と木をさじで固定する作業が一番大変でした。木は多いし、おじ山はつぶれし(つぶ)した、材料は少ないし、と幸か、たですが一番楽しかったです。



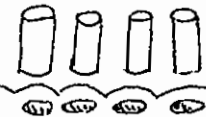
男なら
女し

コクの微糖



たがーい

あーた



たーい

つめた



かーい

あーた

と
微糖



1000

し口

キ

取

の
人
だ
後
は

リ
サイ
クル

士
売
り
物

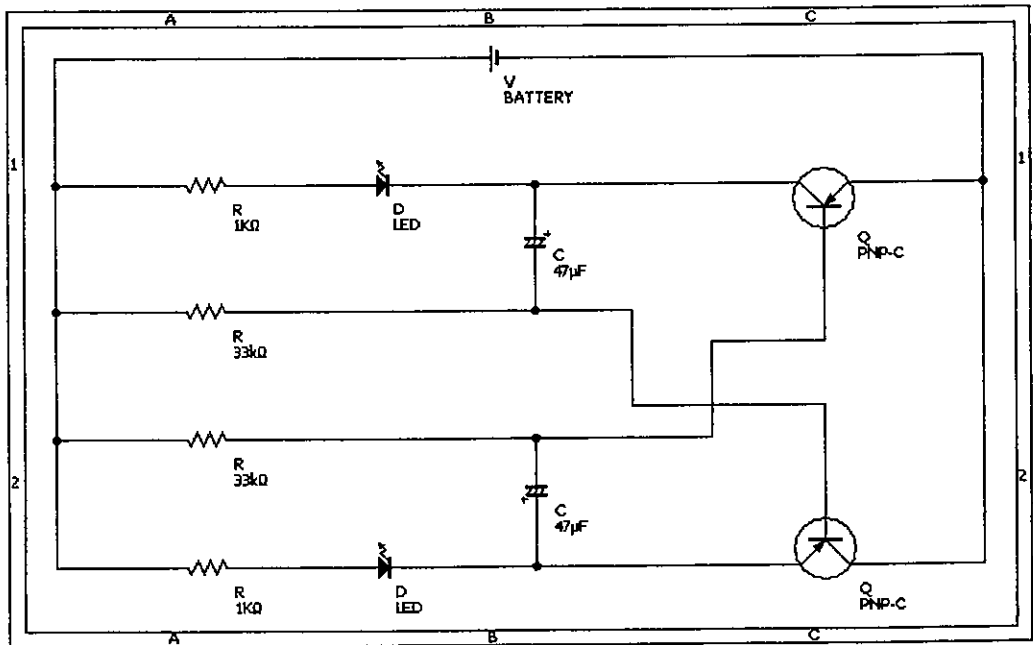
チカチカ

M3 宮嶋 優大

・チカチカとは

チカチカ(本名=自走マルチバイブレータ)とは、二つのLEDという物が交互に輝く、その名の通りの物です。その原理は、電池から流れた電気がコンデンサーという物に溜まり、その電気がLEDに流れ、それが二つ交互に行われる、というものです。容量の違うコンデンサーを使えば、点滅の速度も変えられる訳です。

仕組みは何だか難しそうに見えますが、制作は半田ごてを持った事の無い方でも可能です。

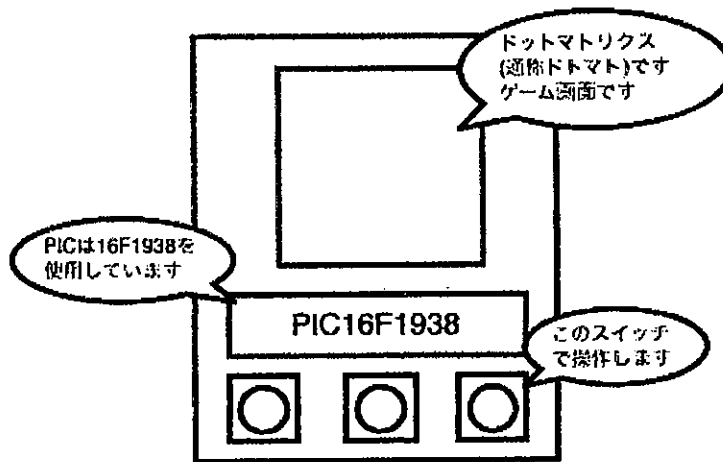


↑
回路図です。

EDGE MDTE

M3 佐伯大地
協力 物無のみなさま

・外観・本体 注イメージ図です



スイッチは左から順に1・2・3です。

・ストーリー

私の名前はキャプテン・エイスケ。

故郷のストーンスターでは、結構名前の知られたベテラン宇宙航海士だ。

あるときは小型貨物を、またあるときは、宇宙資源を、愛機の宇宙船スーア号で星から星へ運ぶ商売をしている。

普段となにも変わらないある夜、仕事の依頼が入った。

出発の前夜、愛する妻が私の好物のスープを作ってくれた。ストーンスターの特産である、真っ赤なトマトがたっぷり入った、とてもおいしいスープだ。満腹になった私は、あたたかいベッドにもぐりこみ、明朝からの航海のことを考えた。やがて私は、幸せな深い眠りへと落ちていった……。

航海は順調だった。スーア号は宇宙という大海原を軽快に突き進んでいた。この航海が、すばらしいものになる予感を覚えた。私はスーア号を自動操縦に切り替え、少しラノベを読もうと操縦席を立った。その時！とてつもない衝撃がスーア号の機体をつらぬいた。機器類にたたきつけられる。

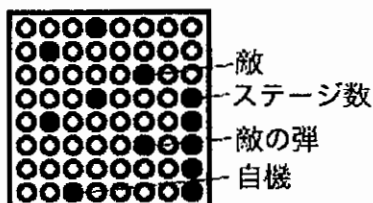
なんだこれは！外を見ると大量の戦艦が通ってきている、
スーア号はまたあの衝撃を受ければ壊れてしまうだろう。

ともかく、私は無事だった。だが、宇宙船の壊れかたはすさまじく、前進ができなくなってしまった。おまけにやつらの狙いはスーア号の荷物のようなのだ。

やつらさえおせば、宇宙船の修理が可能になり、ストーンスターへ行くことができるのだが・・・。

うまく攻撃をかわしながらやつらを倒し、私は無事に故郷に帰ることができるのだろうか？

・ゲーム説明



このゲームはシューティングゲームです。全8ステージあります。4と8ステージにはボスがいいます。弾を発射し、敵に当ててたおします。敵は次々と現れ、一定数倒すと次のステージへと進みます。ドトマトの一番右の列はステージ数を表していますが、第8ステージではボスの体力を表します。一番下の列にいるのが自機、上の方にいるのが敵です。第8ステージのボスは弾を斜めにも撃ってくるので注意が必要です。

・操作方法

- 1: 自機を左へ移動させる
- 2: 弾を発射する・次へ進む
- 3: 自機を右へ移動させる

・ドットマトリクス

このゲームでは表示にドットマトリクス、通称ドトマトを使用しています。

ドトマトとは、縦横にLEDが並んでいるディスプレイです。

ドトマトはダイナミック点灯という方式で制御します。

これは人の目では認識できないほどの速度で1列ずつ順番に光らせ、残像効果を利用して常時表示されているかのように見せるものです。

こうすることで少ないデータ量で通常と同じような表示をすることができます。

具体的には、このゲームに使用しているドトマトの場合、通常はLEDの数分、つまり64本のPINが必要となるわけです。しかしダイナミック点灯を使うと16本で済みます。

内部構造は次のページにのせておきます。

麻布学園物理部無線班

回路図集2013年

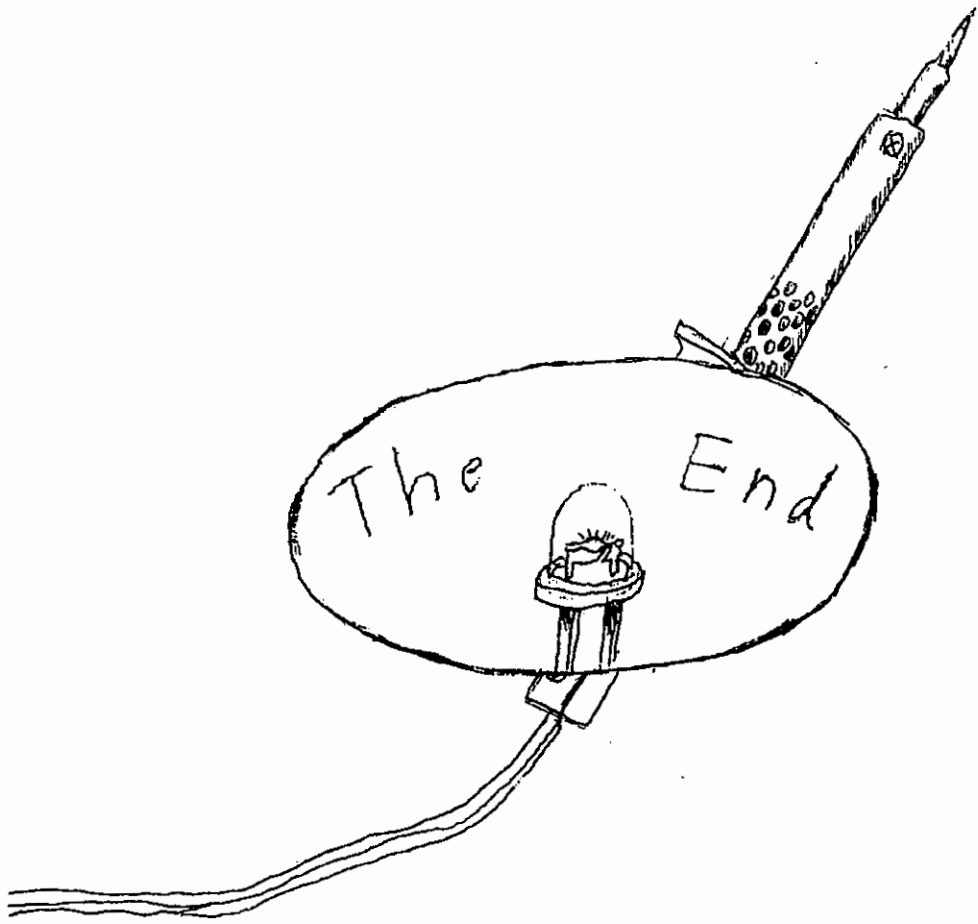
発行日	2013/04/13
編集責任者	岸田 聖生
販売責任者	水野 太郎
デザイン	山本 涼一 黒田 健太 佐藤 惇 花園 佳月 石倉 匠 横尾 幸丸 関根 史人

物理部無線班公式HP

URL <http://azabubtm.web.fc2.com/>



落丁乱丁はお取り替え致します。転載複製は自由です



羊200-

2013年物理部無線班