はじめに

昨年の回路図集では「6コア/12スレッドCPUなどと平列化を図っている」とあったが最近ではグラフィックチップとプロセッサを統合したCPUや、APUと呼ばれる物も登場し、並列処理効率の追求は新しいステージに進んでいる。我が部でも32bitマイコンが本格的に採用され始め、またPIC32、ARM Cortex M-3などの最新鋭のチップも使われている。

この回路図集は麻布学園物理部無線班の部員が1年間研究製作したものを記したものです。中には未完成のものもあるため、この本の通りに作ってもうまくいかない場合がありますのであらかじめご了承ください。

H21会計 鈴木 泰也

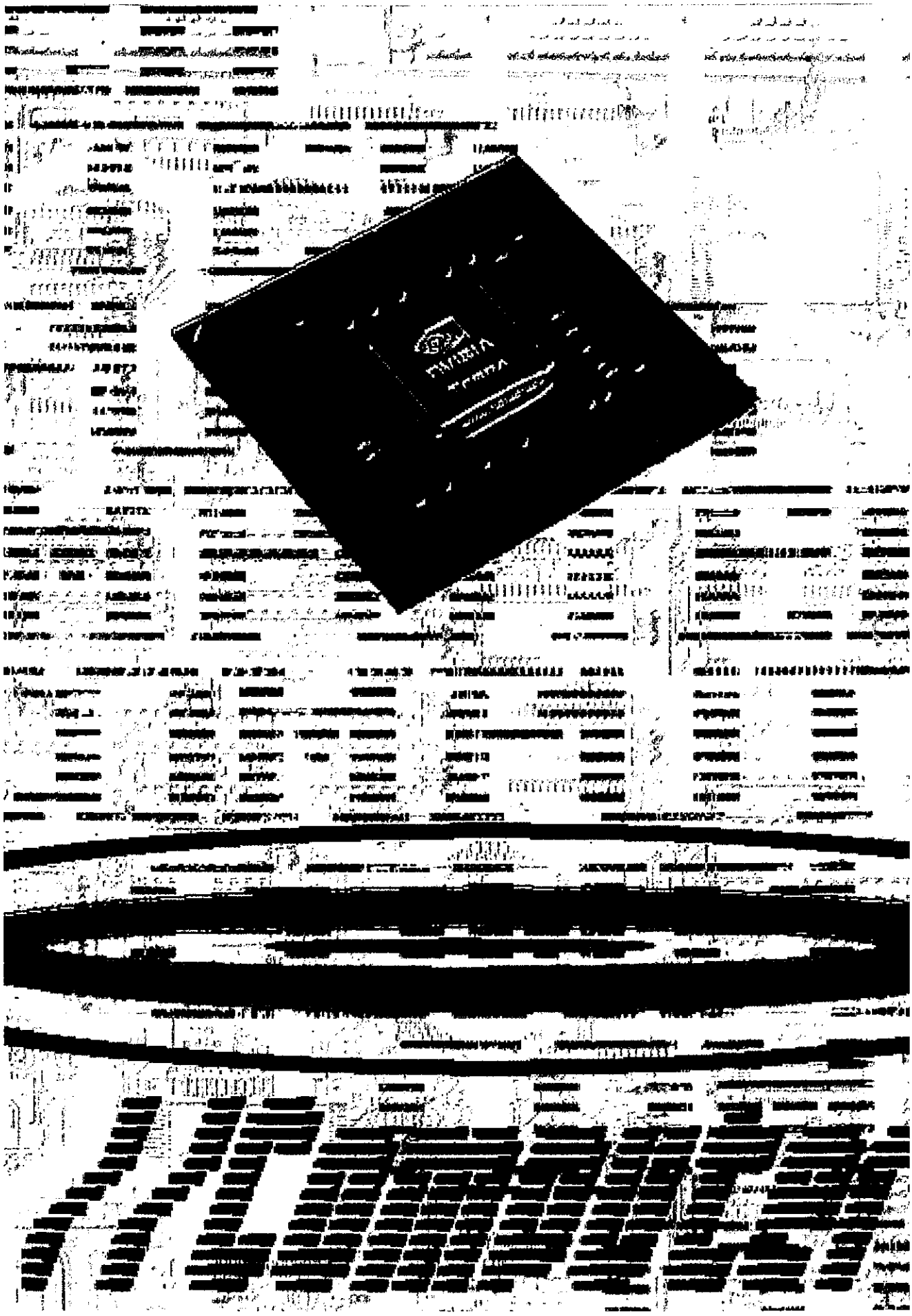
目次

2011

2011

ゲーム系

販売物



i-roid

製作者　H1　藤津直弥
M3　水野太郎

感謝　物無の諸氏
ネット上の諸氏

・概要

いわゆるデジタルフォトフレームです。分かりやすく言うなればiPo…ええ、林檎ですとも。
使用した液晶がタッチパネル付きなので、ジェスチャー機能も搭載させたいと思っています。(右へスライドでページ送り、長押しでバック等)タッチ機能を生かしたミニゲーム、その他最近の情報端末らしい機能も将来的に製作していく予定です。

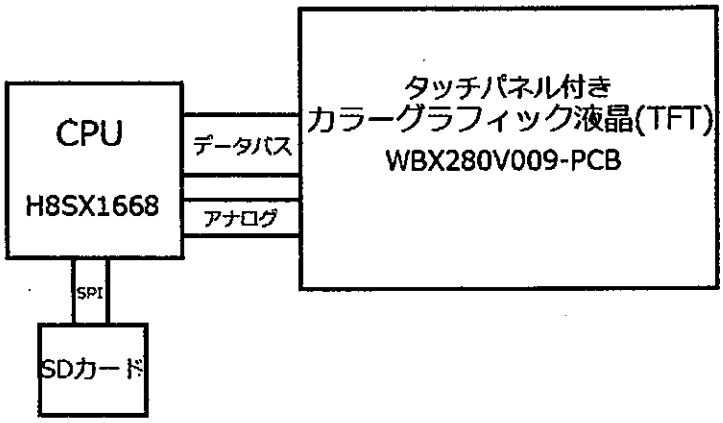
メイン部分

H1 藤津 直弥

SDカードとの通信を筆頭とする各種データ取り扱いを担うCPUを中心として構成された部分です。ここがこの制作物の中枢とも言える部位で、様々なインターフェースを制御し、人とCPUを結びつけます。

・大まかな構造

CPUを中心とした構造です。
SDカードとの通信はSPI通信、液晶パネルとの通信はデータバス、タッチパネルの制御はアナログ制御を使用しています。



・使用部品と大まかな説明

使用した部品について詳しく説明しても、うまく伝えきれないのと理解していただけるかわからないため、うわ

べだけサラッと説明いたします。

・CPU

・H8SX1668

ルネサス製マイコン「H8」シリーズ最高の性能を持つ「H8SX マイコン」のうちひとつ。基本命令を1クロックで行うなど下位モデルよりも処理性能に長けている。が、同社の「SH」シリーズの方が実は高性能だったりもする。
USB での書き込みが可能で、近年の PC でも手軽に最低限の開発環境を揃えられる。
残念ながら純正コンパイラは試用期間があり、過ぎると 64 キロバイトまでと制約がつく。

・SD カード

・microSD カード

この本を読んで下さっているあなたのケータイにもきっと刺さっているであろうおなじみのメモリ。広く普及していて独自規格だいすき S#NY 以外の製品でだいたい使える。
難しい制御方法と簡単な制御方法がある。難しい制御方法はライセンス取らないと教えてくれないなど制約が多い。逆に簡単な制御方法は SPI 通信という制御方法で、我ら物理部無線班員の使用するマイコンでも制御可能で、というか正直なところネット上にたくさんデータがあるのでまず困らない。
この microSD カードは FAT というファイルシステムを使用している(最も他にもファイルシステムは存在するのだが)。こちらもマイコン用に「FatFs」というファイルシステムモジュールがフリーソフトウェアとして公開されている。感謝しながら使用。一から組むと気が遠くなるような作業が待ち受けている。ここでは画像ファイルを保存する場所として使用している。ここに保存されたファイルを CPU が読み込み一覧を液晶に表示してタッチで選択、という寸法だ。

・液晶パネル

・WBX280V009-PCB

秋葉原の液晶パネルのお店、「CoCoNet」にて販売されていた液晶パネルの一つ。この液晶パネルのコントローラチップは他の違うサイズの液晶パネルにも使われていて、似たもの同士ならある程度というかほとんど使い回しが効く。安価なマイコンでの制御を容易としたインターフェースで、改造すれば 8+αピンでの制御が可能になり、PIC などピン数の少ないマイコンでも制御に余裕ができるようになる。最も、100 とちょっとの入出力ピンを持つ H8SX1668 でこの改造を施すのはまさにナンセンスなのだが。最大で 262,144 色表示できるのだが、設定が面倒なのと液晶から引き出されているバスの幅が足りないため一度にデータを送れず、送信に時間がかかってしまうため 65,536 色での表示を行なっている。「LCD&タッチセンサ活用の素」という書籍を参考にさせていただいております。

・その他精巧な部品たち

・microSD カードソケット

そのまんま microSD のソケット。直に基板にはんだづけしてコンパクトに収めてみた。差し込み状態を検出可能だが、動作中の差し込み・取り出しは規定していないので使っていない。特記事項はない。

・DIP スイッチ

H8SX は動作モードが多数ある。クロックの速さや USB の使用の有無、プログラムの書き込みなどだ。それらのモードを手で簡単に切り替えられるようにするために使用。

・水晶発振器

H8SX を動作させるために欠かせない部品の一つ。マイコンはクロックがないと永遠に仕事してくれないので、そのクロックを生成する部品が必要となる。このクロックを内部で逓倍し、高速での動作をしている。USB でのプログラムの書き込みをするため、本来なら逓倍後で 50MHz までサポートされている動作周波数を 48MHz に少し落としている。

・コンデンサ

最も有名な電子部品の一つにして電子部品らしい電子部品の一つ。電気を貯めておくことができる。その性質を利用して、ノイズやチャタリング(細かく言えばこれもノイズの一種)の抑制によく利用されている。バイパスコンデンサと言って、マイコンの動作の信頼性を高めるために使われる場合もある。(バイパスコンデンサという名前のコンデンサの種類があるわけではない)コンデンサの中でも電解コンデンサは他のコンデンサよりも容量が大きい。そのかわり極性があり、逆方向に電流を流し

過ぎると破裂する.たまにおもしろがってわざとやる者もいるが、良い子は真似してはいけないし、悪い子もぜひやめてほしい、危ないので.
余談だが電解コンデンサには皮膜に色分けが施されている.基本的な色は黒だが、他にも色がある.例えば、
茶　緑　青　橙　銀　金 etc...
と、リオ何とかを想起させる色ばかりだが、亜種というわけではない.しかし、基本的には金・銀色が一番質やら信頼性やらが高い.緑色は変わり種で、電解コンデンサのくせに極性がないため、使い方に幅が出る.

・抵抗

コンデンサと同じくポピュラーな電子部品.これのない電子回路はないと言っても過言ではない.基本的には電気を熱に変える部品.熱に変えると言ってもすなわち熱くなるわけではない.流れる電流によって変わるが、正しい使い方をしていればまず熱は持たず、実は気づかない程度に発熱していたという程度である.これまたいろいろな種類がある.しかし、長くなるのでネットを参照されたし.

・回路図

本当ならば掲載したいのはやまやまなのですが、ピン配置の盛り込まれた H8SX1668 の回路図データがないため割愛させていただきます.さらには、適切な入出力ピンなら、機能ごとにポートを振り分けるだけでソフトウェアで変更が効き、ほとんど動作するためあまり載せる意味もありません.ので、各々のポートに割り当てている役割だけを掲載させていただきます.

ポート	役割
P1H・P1	液晶のデータバス
P9	タッチパネル制御
P50・1	タッチパネル制御(アナログ部)
P2	SD カードとの通信部
PM・MD	モード変更

・ソフトウェア

FatFs はモジュールのため、物理ドライブへのアクセスのためのポートの設定以外変更点は特にありません.
液晶パネルも書籍のサンプルプログラムを元に初期化を行っています.そのためこの二つは割愛させていただきます.
現時点で完成している BMP ファイルの表示部分のみを掲載いたします.

```
FATFS fatfs;
FIL fil;
DIR dir;
FATFS *fs;
DWORD fre_clust;
DWORD fre_sect;
DWORD tot_sect;
DWORD res;
DWORD clust;
char buff1[952];        //適当
char buff2[960];        //1ピクセル分の BMP データ格納用
short yyyy;             //Y 座標格納用
short xxxx;             //X 座標格納用
UINT many;
f_mount(0,&fatfs);
f_open(&fil,"Sample.bmp",FA_OPEN_EXISTING | FA_READ);
```

```
f_read(&fil, buff1,54, &many)          //BMPファイルのヘッダ部分をすっとばすためのお茶濁し
for(yyyy=239; yyyy>=0; yyyy--){
        lcd_out(0x0020, yyyy);
        f_read(&fil, buff2,960, &many);
        for(xxxx=0; xxxx<320; xxxx++){
                lcd_out(0x0021, xxxx);
                LCD_CS = 0;     LCD_RS = 0;
                LCD_DH = (0x0022 >> 8) & 0x00FF;
                LCD_DL = 0x0022 & 0x00FF;
                LCD_WR = 0;     LCD_WR = 1;     LCD_RS = 1;
                LCD_DH = (buff2[(xxxx*3)+2] & 0xf8)|((buff2[(xxxx*3)+1] & 0xe0) >> 5);
                LCD_DL = ((buff2[(xxxx*3)+1] & 0x1c) << 3)|((buff2[(xxxx*3)] & 0xf8) >> 3);
                LCD_WR = 0;     LCD_WR = 1;     LCD_CS = 1;
        }
}
```

・BMPファイルとは？

前述のBMPファイルについて説明いたします。
早い話が無圧縮の画像ファイルです。ビットマップファイルと読みます。
無圧縮がゆえに扱いが簡単で劣化も無いですが、無圧縮ゆえにファイルサイズが馬鹿になりません。下手をすれば100メガバイト以上の画像ファイルも存在するとか。
と言っても320*240の画像では気にするほどでもなく、SDカードも1Gバイトあれば文化祭での展示用には十分なほどです。
本来ならばファイルの最初の方にヘッダ部分があるのですが、画像データさえ読み込めればとりあえず表示できますので、ここでは読み込んでおりません。

・タッチパネルとは？

タッチされた場所を電気信号として座標を出力するための部品です。これを画面の上に取り付けたものがタッチスクリーンだそうです。これを利用した情報端末やゲーム機が最近有名ですね。
ご存知かもしれませんが、代表的なものとしては感圧式と静電容量式があります。どのように違いがあるのでしょうか。

・感圧式

早い話「押した」所の座標が出力される。
タッチペンなどでの操作が可能。
マルチタッチは不可
耐久性に乏しい、経年劣化もあり(そのため普通はユーザーで座標補正ができる)。
Dsや安価なAndroid端末などで有名。

・静電容量式

早い話「電圧が下がった」所の座標が出力される(触らなくても反応することもある)。
特殊なものでない限りタッチペンの使用は不可。
マルチタッチ可能(ソフトウェア依存)
上にガラスなど固いもので覆うことができ、耐久性が高い。
iPhoneやメーカー製Android端末で有名。

ここでは制御方法が簡単な感圧式を使用しております。ネット上にある情報のほうが確実なので詳細は割愛させていただきます。

・余談

本当は音楽再生の方にも手をつける予定だったのですが、他の人に花を持たせるためという意

味合いも兼ねて割愛しました。

・感想

「今回の制作ではあまり知識もないままにこんな物を作り始めてしまい、正直行き詰まることが多々ありました。しかし、いろんな人に助けてもらい、自分の無力さを思い知ると同時に人に頼ることの大切さを学びました。回路図設計者としての役割と自分の制作を両立するのも大変でした。文化祭予定日に近づいても未だに画像表示すらできず、焦っていましたが、そんな中画像を表示できたときは心のそこから嬉しかったです。」

と言うと奥いテンプレですが、まさに上記の通りでした。反省点としては、思ったより扱いにくいマイコンを選んでしまったことと、「こんな形のPDAがどこにある」みたいな見かけになってしまったことです。初期の目標のスマートフォンに影が…みたいな形から大きくそれてしまったわけです。来年はここを改善。

何を作るにせよ後輩に向けて一言言うならば、「さっさとやれ」、です。何事も厄介なことは早期に対処してしまいましょう。

最後に、共同制作者の水野君、物無の皆様、そして文化祭に来てくださった皆様、ありがとうございました。

液晶ディスプレイ部

M3　水野　太郎

ここではもとからインターフェースが搭載された液晶パネルとは別に、同期信号を生成して画像を表示する必要のある液晶画面の画像表示について説明します。

・使った物

・LQ038Q5DR01

液晶パネルのお店「CoCoNet」にて販売されていた安価な液晶パネル。安価な割にはインバータ内蔵だったり、保護ケース同梱だったりと気の利いたセットになっています。

・コネクタ&ピッチ変換基盤

上記の液晶とユニバーサル基盤を接続するために使用しました。コネクタは特に壊れやすいので注意が必要です。

・CPLD(XC95144XL)

同期信号の生成やSRAMとデータ書き込み／読み込みを担う中心部分です。ピン数が多いのが取り柄であり悩みの種であり…

・ピッチ変換基盤

上記のCPLDとユニバーサル基盤を接続するために使用しました。変換基板上にパスコンを設置できるのが魅力です。

・SRAM(CY7C1041DV33)

液晶画面に表示するためのデータを一時的に保存しておくために使います。他のSRAMより高速です。

・ピッチ変換基盤

上記SRAMとユニバーサル基盤を接続するために使用しました。正直かさばるので出来れば使いたくありませんでした。

・回路図

続いて、CPLD内部の回路図を掲載します。大きいので次のページに掲載します。

液晶との配線は、CPLD側でピンアサインをいじれるので割愛します。

```
f_read(&fil, buff1,54, &many)          //BMPファイルのヘッダ部分をすっとばすためのお茶濁し
for(yyyy=239; yyyy>=0; yyyy--){
        lcd_out(0x0020, yyyy);
        f_read(&fil, buff2,960, &many);
        for(xxxx=0; xxxx<320; xxxx++){
                lcd_out(0x0021, xxxx);
                LCD_CS = 0;     LCD_RS = 0;
                LCD_DH = (0x0022 >> 8) & 0x00FF;
                LCD_DL = 0x0022 & 0x00FF;
                LCD_WR = 0;     LCD_WR = 1;     LCD_RS = 1;
                LCD_DH = (buff2[(xxxx*3)+2] & 0xf8)|((buff2[(xxxx*3)+1] & 0xe0) >> 5);
                LCD_DL = ((buff2[(xxxx*3)+1] & 0x1c) << 3)|((buff2[(xxxx*3)] & 0xf8) >> 3);
                LCD_WR = 0;     LCD_WR = 1;     LCD_CS = 1;
        }
}
```

・BMPファイルとは？

前述のBMPファイルについて説明いたします。
早い話が無圧縮の画像ファイルです。ビットマップファイルと読みます。
無圧縮がゆえに扱いが簡単で劣化も無いですが、無圧縮ゆえにファイルサイズが馬鹿になりません。下手をすれば100メガバイト以上の画像ファイルも存在するとか。
と言っても320*240の画像では気にするほどでもなく、SDカードも1Gバイトあれば文化祭での展示用には十分なほどです。
本来ならばファイルの最初の方にヘッダ部分があるのですが、画像データさえ読み込めればとりあえず表示できますので、ここでは読み込んでおりません。

・タッチパネルとは？

タッチされた場所を電気信号として座標を出力するための部品です。これを画面の上に取り付けたものがタッチスクリーンだそうです。これを利用した情報端末やゲーム機が最近有名ですね。
ご存知かもしれませんが、代表的なものとしては感圧式と静電容量式があります。どのように違いがあるのでしょうか。

・感圧式

早い話「押した」所の座標が出力される。
タッチペンなどでの操作が可能。
マルチタッチは不可
耐久性に乏しい、経年劣化もあり(そのため普通はユーザーで座標補正ができる)。
Dsや安価なAndroid端末などで有名。

・静電容量式

早い話「電圧が下がった」所の座標が出力される(触らなくても反応することもある)。
特殊なものでない限りタッチペンの使用は不可。
マルチタッチ可能(ソフトウェア依存)
上にガラスなど固いもので覆うことができ、耐久性が高い。
iPhoneやメーカー製Android端末で有名。

ここでは制御方法が簡単な感圧式を使用しております。ネット上にある情報のほうが確実なので詳細は割愛させていただきます。

・余談

本当は音楽再生の方にも手をつける予定だったのですが、他の人に花を持たせるためという意

味合いも兼ねて割愛しました。

・感想

「今回の制作ではあまり知識もないままにこんな物を作り始めてしまい、正直行き詰まることが多々ありました。しかし、いろんな人に助けてもらい、自分の無力さを思い知ると同時に人に頼ることの大切さを学びました。回路図設計者としての役割と自分の制作を両立するのも大変でした。文化祭予定日に近づいても未だに画像表示すらできず、焦っていましたが、そんな中画像を表示できたときは心のそこから嬉しかったです。」

と言うと良いテンプレですが、まさに上記の通りでした。反省点としては、思ったより扱いにくいマイコンを選んでしまったことと、「こんな形のPDAがどこにある」みたいな見かけになってしまったことです。初期の目標のスマートフォンに影が…みたいな形から大きくそれてしまったわけです。来年はここを改善。

何を作るにせよ後輩に向けて一言言うならば、「さっさとやれ」です。何事も厄介なことは早期に対処してしまいましょう。

最後に、共同制作者の水野君、物無の皆様、そして文化祭に来てくださった皆様、ありがとうございました。

液晶ディスプレイ部

M3　水野　太郎

ここではもとからインターフェースが搭載された液晶パネルとは別に、同期信号を生成して画像を表示する必要のある液晶画面の画像表示について説明します。

・使った物

・LQ038Q5DR01

液晶パネルのお店「CoCoNet」にて販売されていた安価な液晶パネル。安価な割にはインバータ内蔵だったり、保護ケース同梱だったりと気の利いたセットになっています。

・コネクタ&ピッチ変換基盤

上記の液晶とユニバーサル基盤を接続するために使用しました。コネクタは特に壊れやすいので注意が必要です。

・CPLD(XC95144XL)

同期信号の生成やSRAMとデータ書き込み/読み込みを担う中心部分です。ピン数が多いのが取り柄であり悩みの種であり…

・ピッチ変換基盤

上記のCPLDとユニバーサル基盤を接続するために使用しました。変換基板上にパスコンを設置できるのが魅力です。

・SRAM(CY7C10410V33)

液晶画面に表示するためのデータを一時的に保存しておくために使います。他のSRAMより高速です。

・ピッチ変換基盤

上記SRAMとユニバーサル基盤を接続するために使用しました。正直かさばるので出来れば使いたくありませんでした。

・回路図

続いて、CPLD内部の回路図を掲載します。大きいので次のページに掲載します。
液晶との配線は、CPLD側でピンアサインをいじれるので割愛します。

・CPLDのおはなし

最近活躍の場を広げてきているCPLDとFPGA、そして最近生産中止に追いやられつつあるロジックICについて少し話します。

・CPLDとFPGAとロジックIC

生い立ちは省略して、世の中にはCPLDとFPGAというICがあります。これらのICに何が出来るのでしょうか。具体的には、ロジックICで作ることの出来る回路のほとんどをCPLD·FPGAで再現できます。故に、ロジックICのうちマイナーな型番は次第に生産中止に追いやられています。

・CPLDとFPGAの違い

一般にCPLDにはフラッシュROMが内蔵されており、電源投入からすぐに動作が可能です。しかし、FPGAは外部にコンフィグレーションROMが必要です。その代わりFPGAはCPLDより大規模な回路を組むことが出来る上、電源を投入し、動作のデータ読み込み後は高速な動作をします。

・プログラミング方法

メーカーから提供される開発環境を使用してプログラミングします。回路図を描くだけでもプログラミングは出来ますが、専門的な言語を使用するなど多様な方法があります。しかし、書き込みに関しては個人ユーザーには手を出しにくいものとなっています。パラレルポートを使用したライタなら、安価な部品で自作することが出来ますが、近年のパソコンにはそんな旧世代の規格のポートは搭載しておりません。USBを用いた純正ライタは数万円もします。純正でなくとも1万円します。物理部無線班にあるパソコンは幸いにも容赦なく旧世代なので、すべてのパソコンにパラレルポートが搭載されており、悩むことはあれど困ることはありません。

・ロジックIC

物理部無線班では中一(文化祭時点では中2)がゲームを作る際に使用している他、高学年の製作でもマイコンだけでは不足する機能を補うために使われることが多いです。

・感想

他の仕事ができたり自分の力が足りなかったりしたせいで遅れをとってしまいました。来年はきちんと準備をし、計画を立てて活動したいです

DJ MANIAX ~type B~

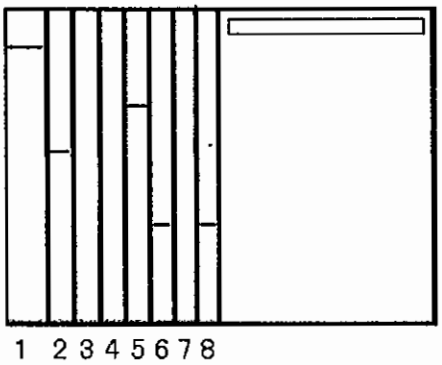
製作者：H1 川田 希望

協力者：物理部の皆様

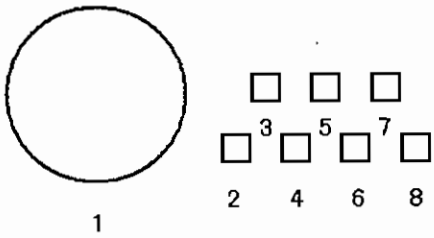
・どんなゲーム？

① スタートスイッチを押すと音楽が流れてきます。
② 音楽に合わせて各ライン上から線状のものが降ってきます。
③ 下の赤い線のところに線状のものが来たら、対応するボタン or ディスクを回します。
④ タイミングよく叩く事ができたら右上にあるノルマのゲージが増えていき、ミスをしてしまうとノルマのゲージが減っていきます。
⑤ 音楽が終わった時にノルマのゲージが一定量に満ちていればゲームクリア、満たしていなかったらゲームオーバー。
⑥ もう何も無いです

こんな感じのゲームです。
要するに某 KONAMI のゲームのパクリですね。



ゲーム画面

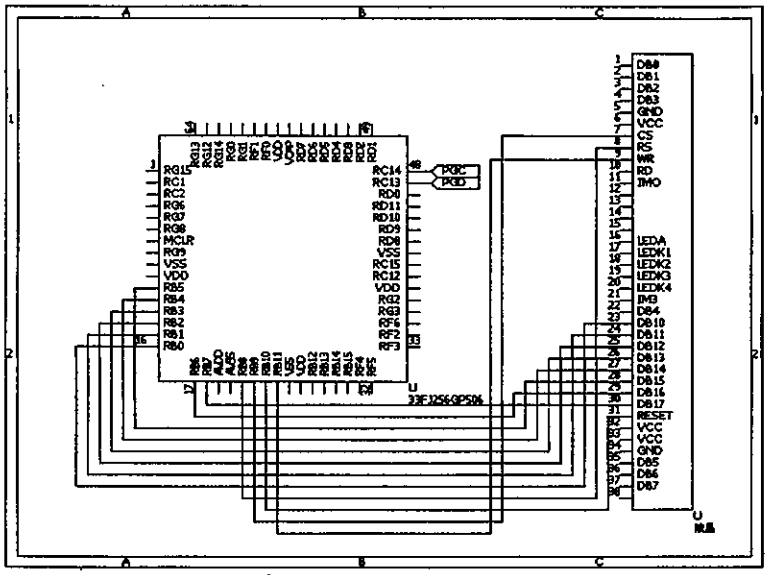


コントローラー配置

同じ数字のボタンが対応しているボタンです。
1番の丸いのは回転する仕様になっていて、回転させることによって反応させます。

・回路等の説明など

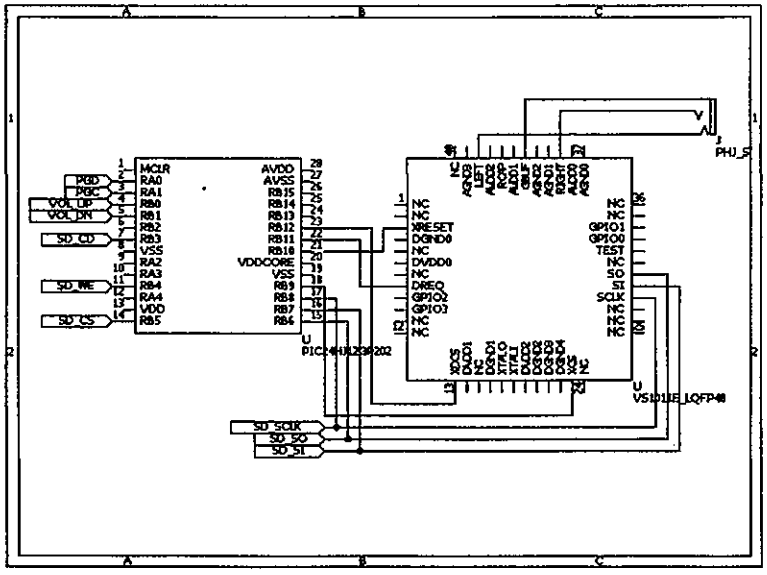
液晶部



※電源その他一部のピンを省略しています。

dsPIC33FJ256GP506を使用して液晶部分の制御とMP3の再生タイミングを決めています。

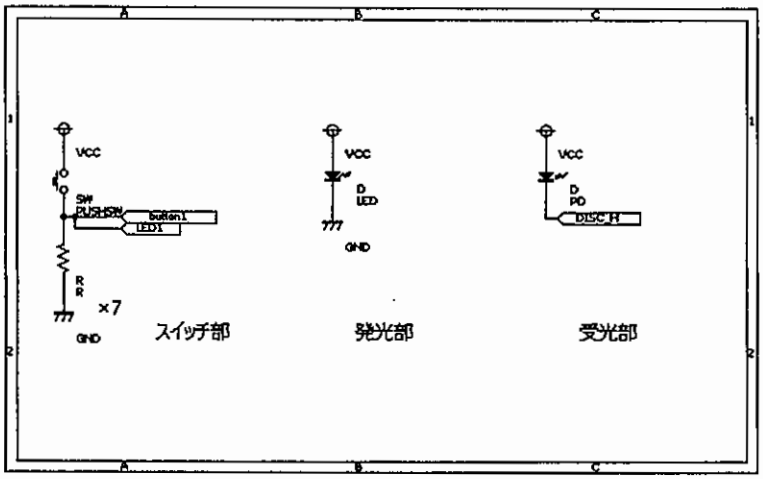
MP3部



※電源その他一部のピンを省略しています。

PIC24FJ12GP202 を使用して、MP3 部分に使用する VS1011e と SD カードを制御しています。

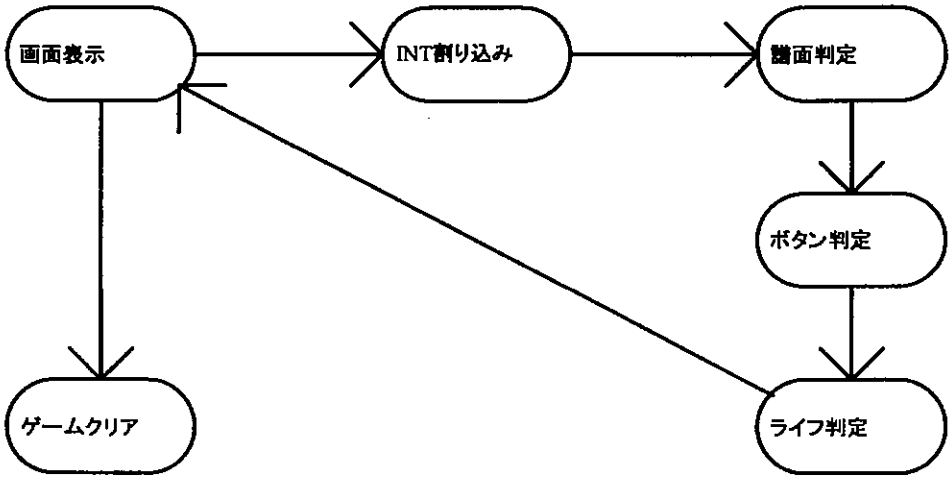
コントローラー部



回転部分の回転の認識は、受光モジュールを使用しています。
回転部の裏側に光を通す箇所と通さない箇所が出来るようにするために、黒いアクリルをつけています。回転することによって光が入ったり入らなくなったりするので、回転していることが認識できるようになっています。
言葉だけで言ってもあれなので画像を載せます。

こんな感じになっております。
これを LED と受光モジュールではさんでいるという事です。
スイッチにの方はいたって普通の光るスイッチを使用しています。スイッチを押すと光るように回路が組んであります。

・プログラム
プログラムは長すぎるためにここには書けません。
簡単なフローチャートを載せます。



感想

今回は液晶のゲームを作らせていただくことになりました。個人製作でわからない点等を他の人などに聞きながら進めていき、とりあえずカラー液晶の制御などの基本的なところは出来るようになりました。また、コントローラーの方も何とか動かす事ができました。

現在(3/29)残りはMP3部分の再生と、譜面を流すところが残っています。MP3は一昨年から動いている部分なのでなるべく早めに仕上げたいです。

文化祭までにはなんとか遊べる形までもって行きたいです。

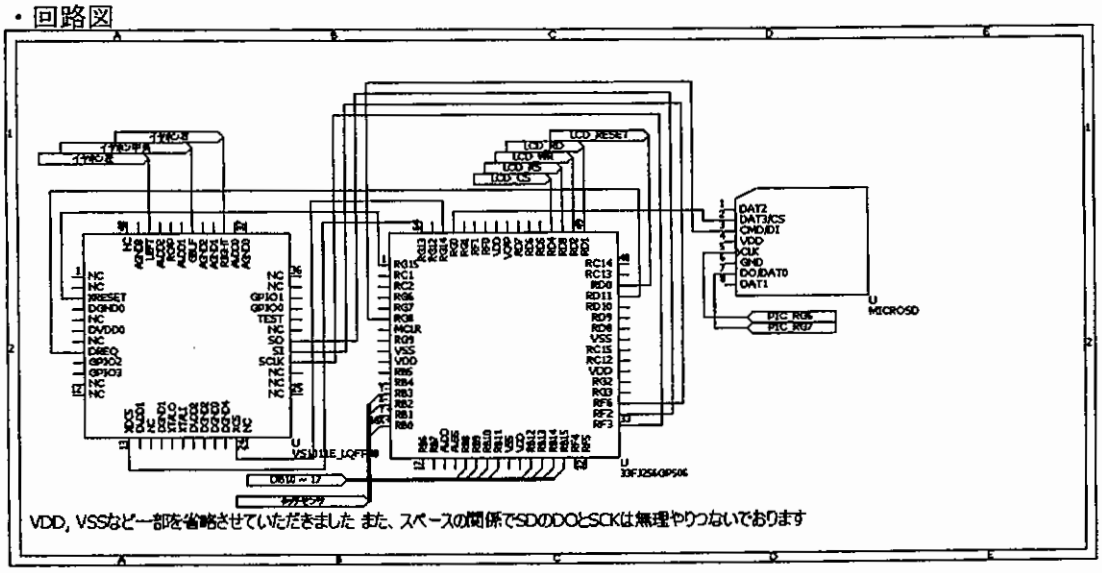
MP3Player and 太鼓の達人

高一　森田　晃平

・概要

この制作物は、MicroSD, VS1011E, タッチセンサ付き液晶(YHY024006A-PCB) をそなえ、
それを dspic33fj256GP506 が制御しています。　MP3Player になり、また液晶を使いゲームなども作れます。
液晶は coconet, その他部品は主に秋月で買いました
ちなみに液晶　http://www.aitendo.co.jp/product/1975

・回路図



・漢字表記

漢字がこの制作物では表示できますが、方法は Shift-Jis を変換し、MicroSD からフォントデータを読み込み、
それを表示しています。
ちなみに shift Jis から配列へのアクセスへの変換は

```
if (high < 0xa0) {
        high -= 0x81;
} else {
        high -= 0xc1;
}
if (low < 0x7f) {
        low -= 0x40;
} else {
        low -= 0x41;
}
data = high * 0xbc + low;
```

と変換しています

・主な動作

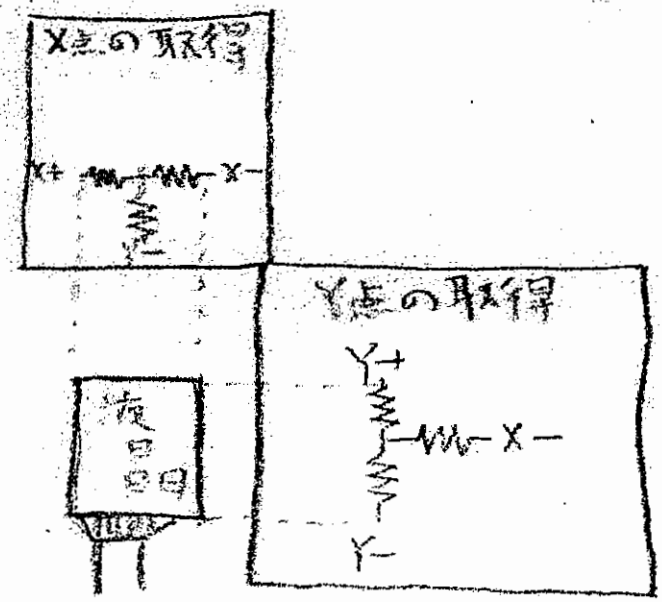
MicroSD との通信
MicroSD は、ファイルシステムが FAT12, 16, 32 のものに対応しています。　これには Chan 氏の Fatfs を使っています。　RAM が 16k あるので、たいがいのことは何とかなります。

VS1011e との通信
普通に SPI 通信をしています　vs1011e のデータリクエスト端子が立ち上がったら割り込みを入れるようにしていますので、ラグを少なくすることができます。

液晶との通信
液晶とは、8bit のパラレル通信でやり取りをしています。　この通信には電子工作の実験室にあった液晶のライブラリを使用しています。　GRAM を用意するほど RAM はないので、文字や図形データを主に扱います。
BMP 表示は実装しておりません

RGB6bit ずつの 18bit 表示が最高ですが、今回は 16bit でやっています 赤と青が 5bit づつです
普通液晶にはクロック制御などが必要なのですが、そこら辺の煩わしいことはすべて液晶内蔵のチップがやってくれます。
なのでこちらはコマンドとデータを送るだけですが、aitendo さんのサイトにサンプルプログラムが置いてあり、また後閑氏のサイトである「電子工作の実験室」にも使用例があり、簡単に動かせるはずです
液晶はアドレスを送ったあとにデータを送る仕組みになっており、代表的なアドレスは
0x21 縦
0x22 横
0x23 RGB データ
となっています
初期化は長すぎるので割愛します
アドレスかデータかは RS ピン
書き込みか読み込みかは RW ピン
データを送るクロックは E ピンです

タッチセンサ
上記の３つは何人か他にもやってる人がいるので僕はこのタッチセンサを重点的に説明しようと思います
まず僕が使ったタッチセンサは４線皮膜式と呼ばれるもので４本線が出ています
この４本は X+, X-, Y+, Y-とあり、　AD 変換をすることでタッチされた場所がわかるようになっています
タッチパネルには、無数の抵抗が並んでいる層が二層あり、互いにくっついていません。
タッチするとタッチされたところで二層が接触します。
そこで分圧したことになり、下の図のようになります　X 点と Y 点を測定すれば、タッチされた場所がわかります。



・感想

今年は、どこまで高い性能を引き出せるかというテーマで挑戦しました。　そのため秋月で買える最高性能の dspic をつかい、挑戦しました。　結果としてはなかなか満足いく出来になったと思っています。

来年で最後なので、来年は気楽にむしろ低い性能、消費電力の PIC や AVR で日常生活に使える小物をたくさん作ってみたりしたいものです。　また、来年は共同制作もいいなあと思っています。

PIC16Fシリーズ殲滅計画　～３０ドルで実現できる開発環境～

H1 朝倉

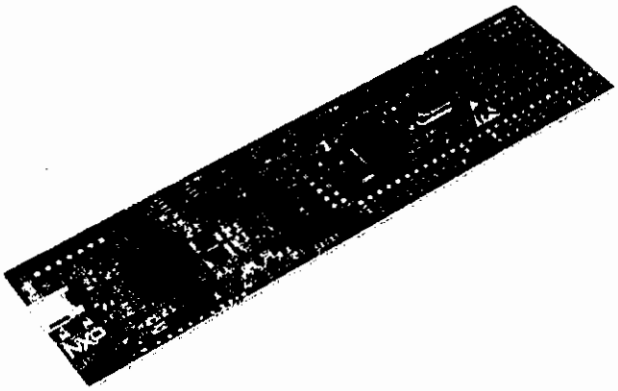
・イントロダクション

読んで字の通りである。今年は高１が無駄に沢山居たので、他人と同じものを使うのもどうかと思って新しいマイコンにチャレンジしようとした次第である。また、このところの技術の進歩に対応するためにも、32bit マイコンへの移行は不可欠なものであった。実際多くの高１が dsPIC などの 32bit マイコンへの移行を進めている。

しかしながら開発環境を自前で揃える様な気概は無かった訳でして、そんな私に丁度良い物があったのである。それがサブタイトルにある"３０ドルで実現できる開発環境"、そう、LPCXpresso だったのだ！

・LPCXpresso の概要

LPCXpresso は NXP セミコンダクター社製評価版マイコンボード LPC シリーズ用の eclipse ベースの統合開発環境（IDE）である。マイコンボードに標準搭載されている LPC-Link と呼ばれるハードウェアを介して、USB 経由でターゲットボードへのプログラムの書き込み、デバッグを行うことができる。…のだがこの機能を使っている人は専ら少数派のようで、大体の人が他社製のデバッグボードなんかを流用していました。とても負けた気がする…ぐぬぬ。



LPC1343

ちなみのこの LPCXpresso なる物はアクティベーションしておかないと書き込みできる容量に制限を食らうみたいなので一応登録してあげないといけないみたいです。

・LPC シリーズの概要

いくつか種類があって、それによって機能が少しづつ違います。

LPC11xx シリーズは Cortex-M0 コアを内蔵していて、最も性能が低い。簡単に言うと USB 通信機能を使えない。

LPC13xx シリーズは Cortex-M3 コアを内蔵していて、USB 通信が使えるけど、ホストにはなれない。

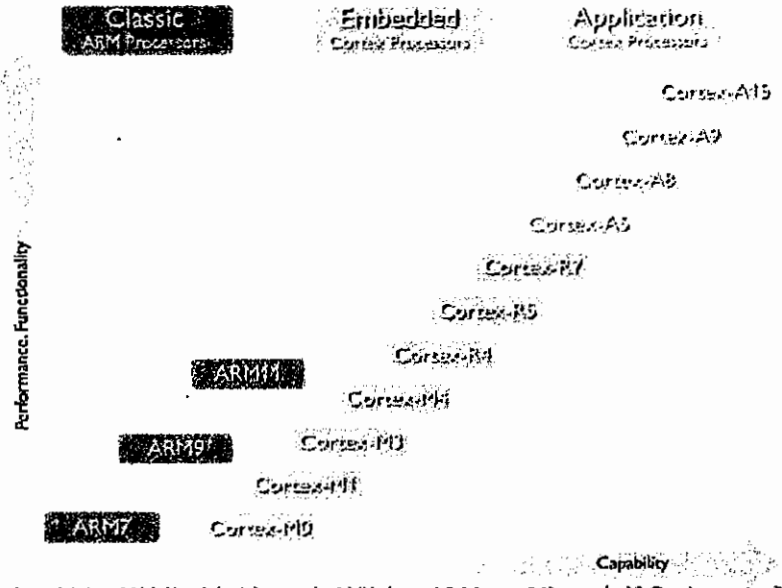
LPC17xx シリーズも Cortex-M3 コア内蔵していて、更に USB ホストまで出来ちゃうというスグレモノである。

すべて PIC 同様ハーバードアーキテクチャを採用しているのでアセンブラ言語による PIC プログラミングと同じ感覚でプログラミングすることが可能である。

実は更に上位機種の mbed なんてのも居ますけどこれは別の話。

・ARMCortex プロセッサの概要

ARMCortex シリーズはこれまでの製品からコンセプトを転換して開発された CPU である。三つのバリエーションがあり、スマートフォンにも搭載されているハイエンドモデルが Cortex-Ax シリーズ、影の薄いミドルエンドモデルが Cortex-Rx シリーズ、縁の下の力持ちの組み込み向けモデルが Cortex-Mx シリーズである。気付いている人も居ると思うが、これらのシリーズ名の頭文字は A,R,M、→ARM となっている。つまり ARM 社のフラグシップの製品群なのである。



右上へ行くほど性能が良くなる。左が従来の ARM ファミリー、右が Cortex ファミリー

余談になるが、以前Intel社員でプロセッサの開発に関わる方に話を伺う機会を持ったことがある。真偽の程は確かではないが、Intelも組み込み向けプロセッサの開発には意欲的で、今後組み込み向けプロセッサ市場の競争は更に加熱していくことが予想される。

・今のところ出来る事の詳細

・チカチカ

いわずもがな恒例のアイツです。残念なことにLPC13xxシリーズには一般的なwait関数が用意されていないためtimer16かtimer32を使って再現するしかない。幸いサンプルプログラムとしてtimer機能を用いたwait関数は用意されているので、これを別途インクルードして使うことになる。

入出力はメモリマップドI/O形式なので、アドレス空間を参照／代入すれば入出力も簡単に行うことが出来る。またgpio.hをインクルードすればピンごとのI／Oをし易くする事が出来る。

・コンソール入出力

なんとLPCXpressoにはターゲットボードとprintf()関数とputc()関数を用いてコンソール入出力を行うことが出来るのである。なんと言うゆとり機能。使い方は新規プロジェクト生成時に「Create NXP LPCxxxx C Project」ではなく、「Create NXP LPCxxxx Semihosting C Project」を選択するだけと、非常に簡便である。

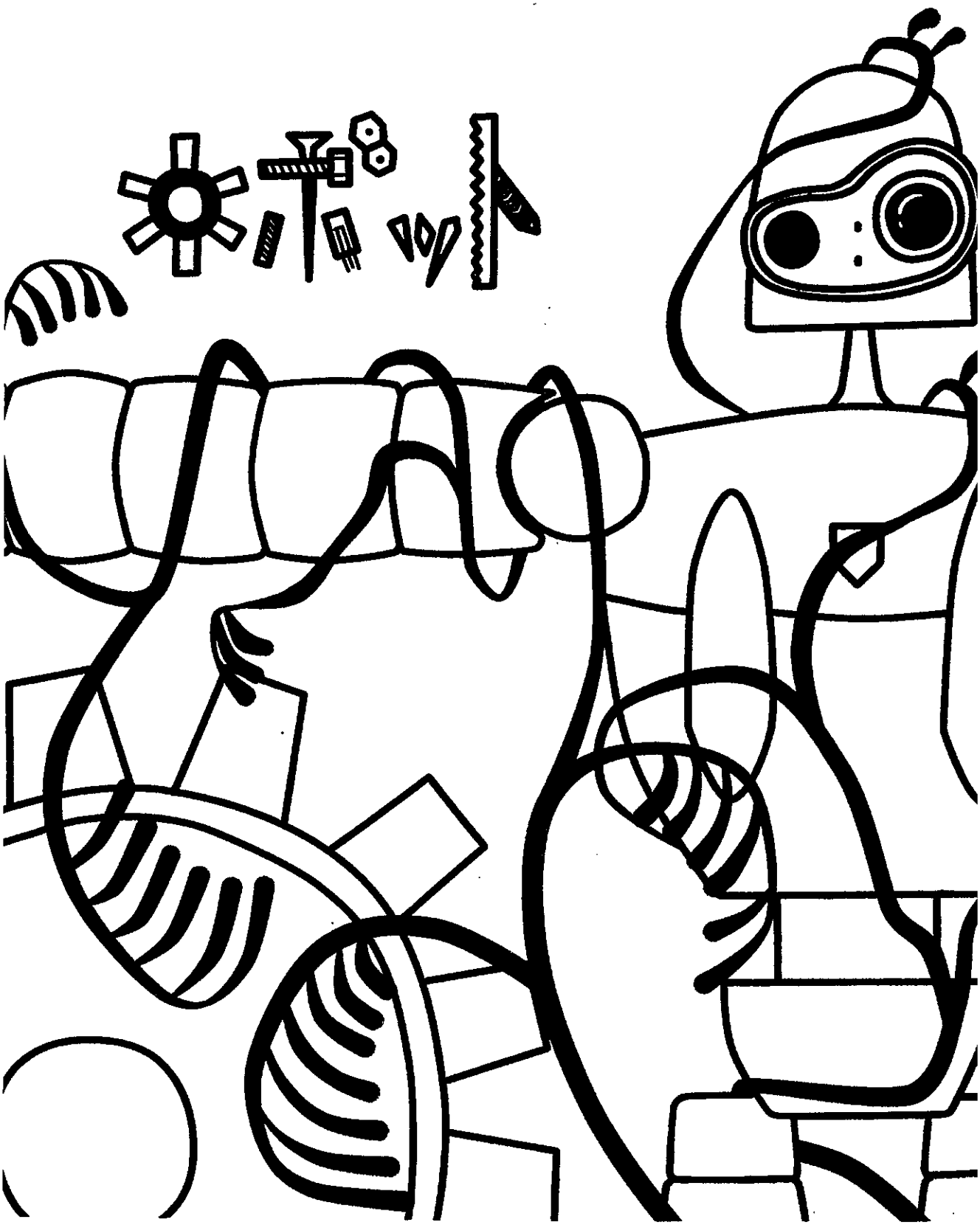
これを使えば簡単にPCとデータのやり取りを出来るようになるので、デバッグ作業の効率化を図ることができるであろう。

・被告人答弁の時間です

・・・ってあれッ!?これだけ!?・・・はい、これだけです。面目ない（泣…LPCXpressoのファイルのインクルードの仕方が意味わかんなかったんだもん。もちろんのこと現在未完動です。時間が許せば文化祭ではスタンドアロンで8bit音を垂れ流しているかもしれませんが、期待しないでください。

追記

やはりインクルードの仕様がよくわからない今日この頃。C言語を習い直すことも視野に入れて活動中です。6/11追記　音が鳴りました。やったね！



Stay away from me or …

製作者：Ｈ２　鈴木　舜也

Ⅰ：-Prologue-

ハァ、ハァ、ハァ――

どうしてこんなことになる！「アレ」の安全性は技術研究所によって保障されていたのではないのか！？

そもそもの発端は「アレ」をうちの小隊で運用することが上から通達されたことだ。

『この自律地上戦闘支援用ロボットは敵地に侵入し、目標だけを認識して排除することが可能で、従来の市街戦のキツイ、危ないという概念をクリーンかつ安全なものに変えることが可能です』クソッ、配備された時のあの研究員はこれ見よがしな笑顔で解説してたじゃねぇか。なにがクリーンで安全だ。「アレ」は仕掛け爆弾を至近であびただけで制御不能になった上、回収するために近づいた隊員を殺害、さらに勝手な動きを繰り返して一定距離内に入った隊員たちを殺害した。生き残ったのは自分ただ一人。

「司令部、どういうことだ！あれは―」

「ザァッ　状況は把握している。残念ながら技術研究所からは想定の範囲外の事態という報告が上がってきている。できることならアレをどうにかしてくれ小隊長」

「はっ！？」

「安心しろ小隊長。このまま「アレ」を放置しても戦局にただちに影響はなi―　ザザザアァ」

クソッ通信が切れた。特徴的な駆動音が聞こえてくる。覚悟を決めるか。「アレ」の姿が見えてくる。

その時、「アレ」との通信のためのPDAに文字が表示される。

「Stay away from me or …」

ガガガガガッ。威嚇射撃が飛んでくる。さぁ始めよう、最低でも部下の仇は取らせてもらおうか。

―to be continue

Ⅱ：概要

しょっぱなからストーリー詐欺をやらかして申し訳ございません。反省も後悔もしています。

このロボットは操縦者の操作により６輪で走行し、前部につけられた赤外線測距センサーによって周囲（主に前方180°）を監視し、一定距離内に接近した物体があると上部につけられたレーザーを目標に照射するものです。

当初は無線操縦化も検討していましたが時間的制約と技術的制約により残念ながら断念し、有線操作に切り替えました。

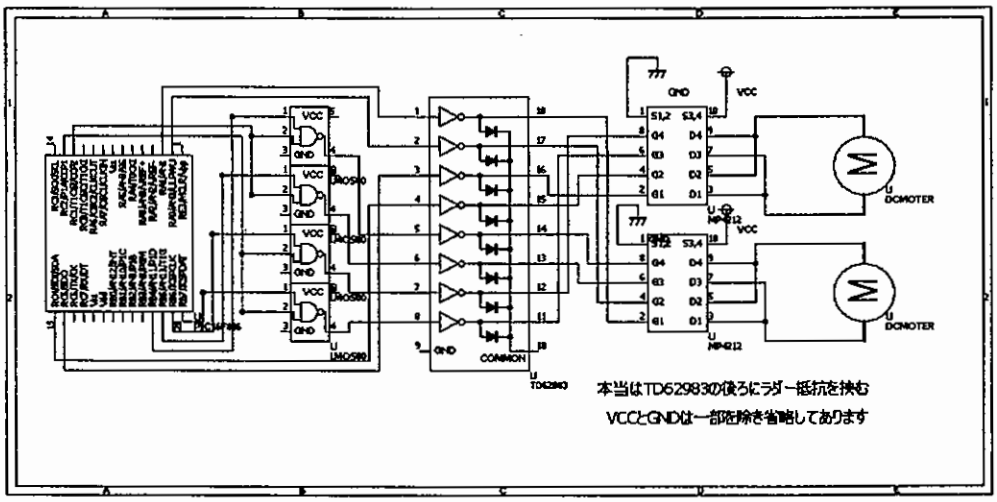
Ⅲ：外観図

あれ？外観図は？

ごめんなさいまだ本体が未完成です。おそらく箱型の簡易的なものとなる予定です。文化祭時に展示されていたら「間に合ったんだな～」とでも思ってあげてください

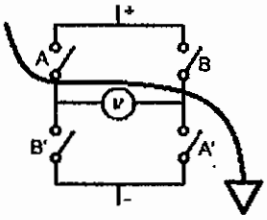
Ⅳ：ＤＣモーター制御

まずは回路図を



注）この回路は全体の回路より DC モーターの制御に関係するところのみを抜き出したものです。

DC モーターとはいわゆる小学校の理科の実験などで使うモーターのことで、最も一般的なモーターと言えるでしょう。通常、この DC モーターは配線をつなぎかえたり、流す電流の向きを逆転させたりしないと逆回転はできないのですが H ブリッヂ回路というものを使うことによりマイコンから信号を出すだけで DC モーターの正転逆転が自由にできるようになります。これが H ブリッヂの回路図となります。



太線はスイッチ A,A'をオンにした場合の電流の流れを示しています。逆に B,B'をオンにした場合は B から B'にむかって電流が流れます。これらの働きにより中央のモーターは正転逆転することができます。電子回路の場合ではこのスイッチを FET という電子的なスイッ

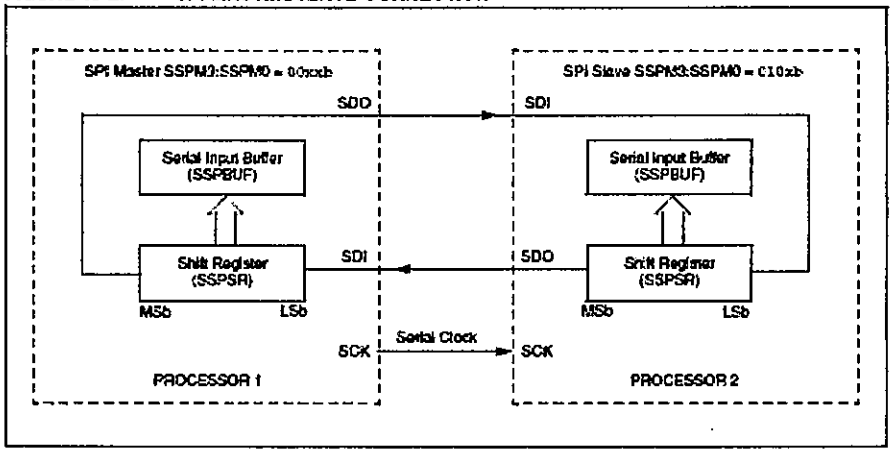
チのようなものに置き換えて回路を組みます。

なおこの制作物ではHブリッヂ回路をそのまま内蔵した便利なICのMP4212を使用しています。ただしMP4212を使用するときには注意が必要です。MP4212の2，4ピンはNチャンネル型のためH信号でONになりますが、6，8ピンはPチャンネル型のため一度信号を反転させる必要があります。さらにこのICはメーカが生産中止を発表しているため入手が困難かもしれません。

V：SPI通信

SPI通信とはモトローラ社（現フリースケール社）が提唱したオンボード（基盤の上での）マイコン間通信方法です。その特徴としては4本または3本の線でpicをつなぎ行う同期式全二重シリアル通信で、Master側とSlave側に分かれて通信することにある。また、Master側がすべての通信の主導権を握る。欠点としてはノイズに弱く、長距離通信には向かないということです。そのため基盤の上でのマイコンの間での通信に限定されてしまうのです。

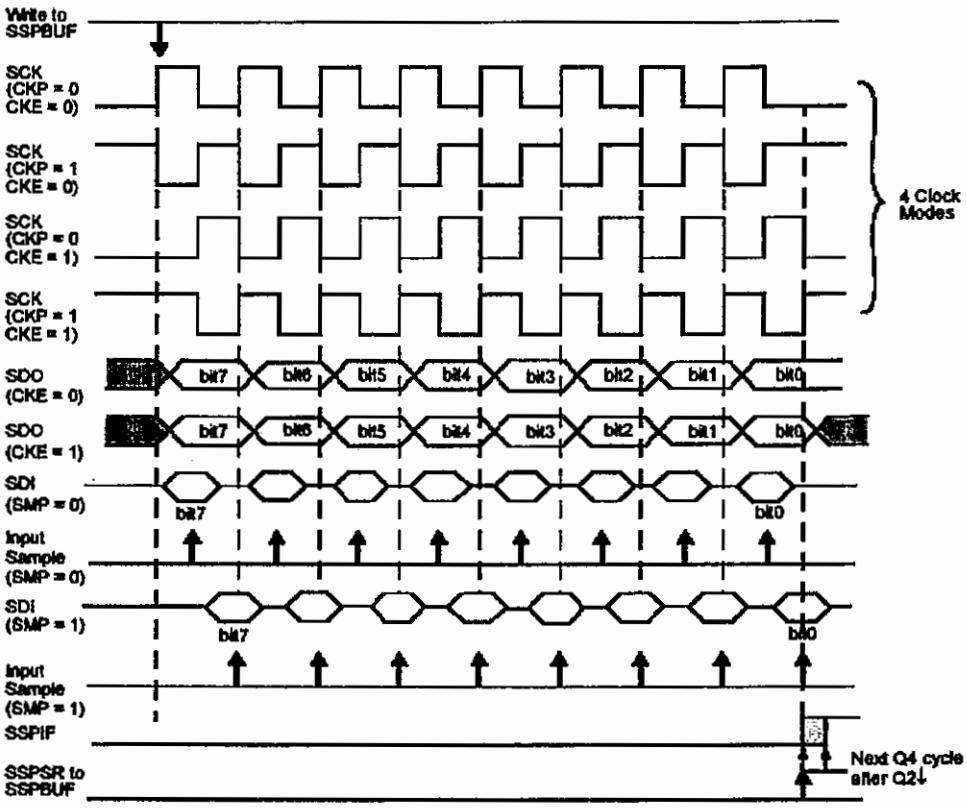
FIGURE 15-2: SPI MASTER/SLAVE CONNECTION



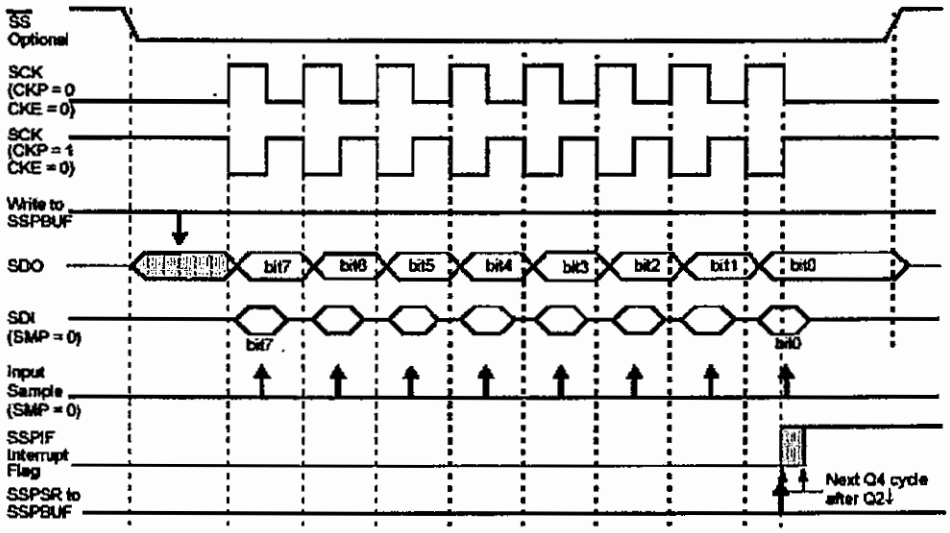
具体的な通信方法としてはまずMasterが同期用クロック（SCKピン）とMaster側の信号（SDOピン）を出力し、Masterの信号を受信した（SDIピン）Slaveが信号を返します。つまりデータのやり取りが上図の矢印のように循環するのです。ここで注意する必要があるのが通信は必ずMasterの主導で行われるということです。つまりSlaveの方から信号を送るには一旦Masterのほうからダミー信号を送らねばなりません。

次ページがその信号表です。

Master



Slave



また、プログラムも CCS 社が提供している関数を使えば簡単です。

Master 側プログラム（digit の数値を加算して順次送信する）

```
void main(){
    setup_spi(SPI_MASTER | SPI_L_TO_H | SPI_CLK_DIV_16 | SPI_SS_DISABLED);
    set_tris_b(0);
    output_b(0);
    set_tris_c(0x17);
    digit = 0;
    while(1) {
        digit = digit + 1;
        output_b(digit);
        spi_write(digit);
        data = spi_read();
        delay_ms(100);
    }
}
```

Slave 側プログラム（受信したデータを出力する）

```
#INT_SSP
void sspx(){
    data = spi_read();
    digit = data;
    output_b(digit);
    delay_ms(100);
}
main(){
    int data;
    setup_spi(SPI_SLAVE | SPI_L_TO_H | SPI_CLK_DIV_16 | SPI_SS_DISABLED );
    enable_interrupts(INT_SSP);
    enable_interrupts(GLOBAL);
    set_tris_b(0);
    output_b(0);
        while(1){
        }
        return 0;
}
```

Ⅵ：AD 変換、サーボモータ制御

このロボットではほかに赤外線測距センサーに AD 変換を、またレーザー砲塔とセンサー下部にサーボモータを使用していますが他の人に説明を委ねて割愛します。

Ⅶ：Epilogue

なんと今年で最後の制作物です。
いままで作った２機のロボットは完動してないから今年はするといいなぁ。
思い返すと色々あるけどまぁ湿っぽい話は後ろに回すとして。
技術的なことをいくつか。
・MP4212 は今のうちに買いためておいてもいいかもしれない。どうせ使うものだし。
・ロボット系一年目でもすぐにサーボ、PWM、SPI 通信、AD 変換などの基本的技術はできるように技術ソース集を作ればいいと思う。そうすればその他の発展的分野に専念することができるようになって制作物のレベルの底上げができるはず。
・次代の人向けの話だが、有望そうな新技術に対して予算を惜しまないほうがいいかもしれない。ただし、あくまで「有望そうな」だが。

―この回路図集を買っていただいたお客さまに―

おそらくこの回路図集を入手した受付にいて「展示大賞をぜひ投票してください」と言われたでしょう。おそらくその人が自分です。

自分はなるべく電子工作の電の字もしらないような方でもわかるように書きましたが、ページの都合及び製作者の力不足によりわかりづらい所も多々あると思います。その時はぜひ遠慮なく受付もしくは展示員までどうぞ。さらに詳しい説明をわかりやすくいたします。

展示大賞はぜひこの物理部無線班に投票願います！

スペースうめなんかではないんです！

MOGIX

製作者：H1 武子 真行・H1 新田 京太朗・M3 津田 信道
協力者：物無の皆様、OB の皆様

I. 概要

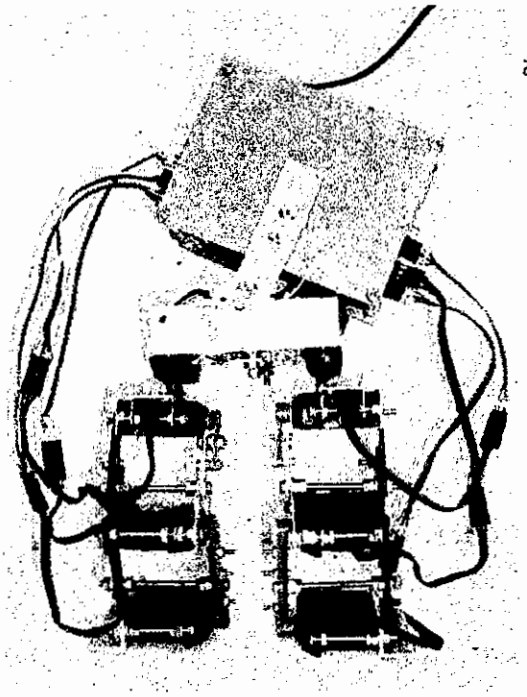
物無では未だ歩行できたという前例がない二足歩行ロボットです。
特にすごい動作をするわけでもありませんが、今回は歩かせるということだけに絞込み、なるべくすっきりとしたロボット設計です。
ロボット名は、津田君のあだ名をもじったものです。

II. 概観

＜仕様＞

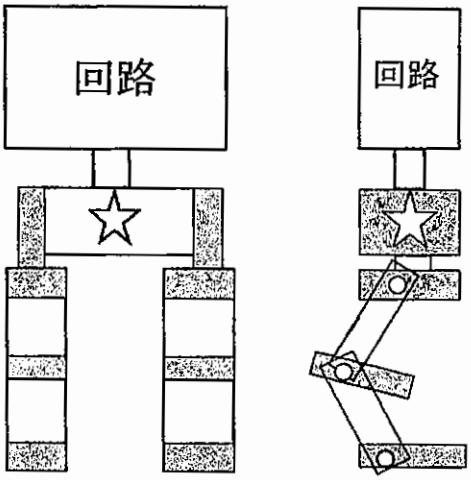
最高速度	：--m/h	関節自由度	：2×4
電源	：5V 電源	重量	：約 600g
高さ	：約 30cm	操縦	：有線式

＜写真＞



未完成なので、作り変える可能性があります。

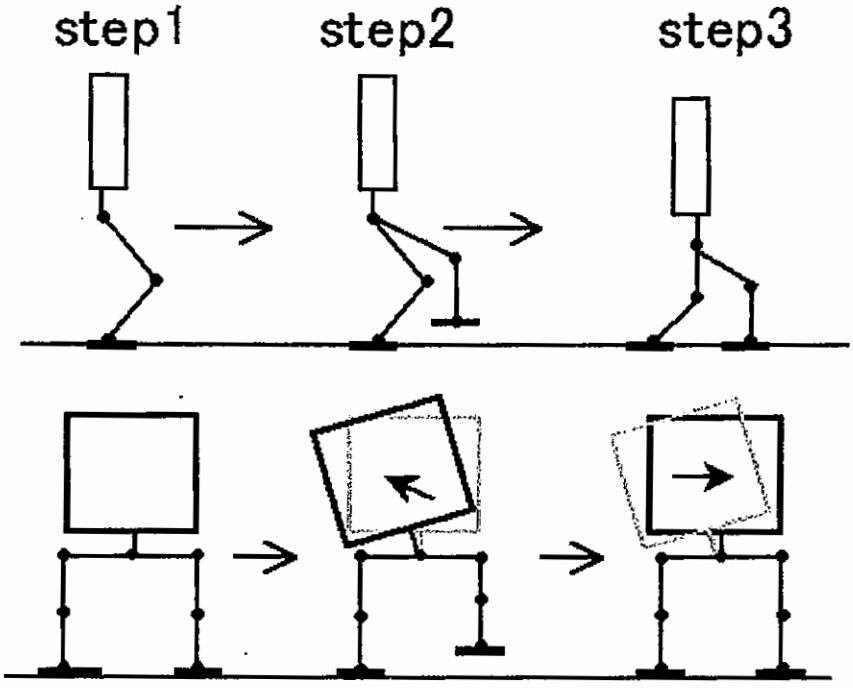
＜構造図＞



左側が、正面から見た図。右側が、側面から見た図。なるべく、簡単に歩行できるように単純な構造になっています。灰色になっている部分が、サーボモーターです(全部で 8 個)。星になっている部分が DC モーターです(1 個だけ)。ちなみに、中に埋め込んであります。ほかの部分は金属板です。

今回は、胴体があります。昔の先輩方の作品には、胴体はありませんでしたが、今回はつけました。胴体を DC モーターにつなぐことにより、重心が取りやすくなっています。片足を上げるときに、上げない側に胴体を振ることによって、重心を安定させます。

＜モーション＞



上が横から見た図で、下が正面からの図です。上と下で同期していて、足を上げると同時に腰を曲げます。

Step1：バランスがとりやすいよう少し足を「く」の字に曲げます。 これが基本の体勢です。

Step2：片足を上げます。この時に倒れないようにするために、腰のDCモーターを使って、　上半身（回路）を傾けて重心を動かします。

Step3：step2 で地面に着けていた足を後ろに曲げ体全体を低くして、上げていた足を地面　に着けて一歩前に踏み出します。また、この時両足が地面に着いたので、傾けてい　た上半身を元に戻します。

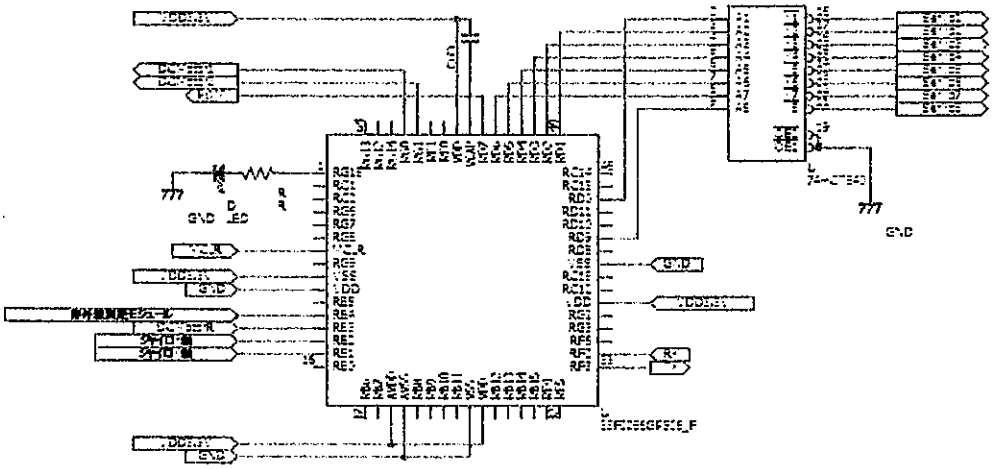
Ⅲ. 回路図

＜ブロック図＞

＜説明＞

CPU は dsPIC33F256GP506 で、センサーとリモコンによって、動作させます。dsPIC33F シリーズは 3.3V 動作なので、ロジック IC の HCT シリーズを使って 5V レベルに戻しています。

＜回路図＞



Ⅳ. DC モーター

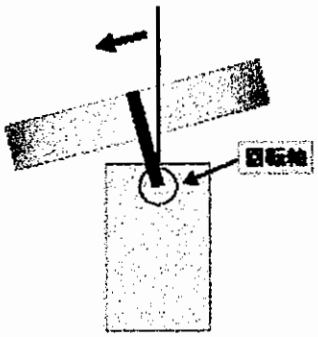
腰を動かすために使っているDCモーターの説明です。DCモーターとは電池につなげば回るよく見るモーターです。この DC モーターの制御には、FET を利用して回転の向きを制御できる H ブリッジを使っています。また、早さも制御できるように PWM というものも行っています。PWM とは、デューティー比と呼ばれる H の割合を変化させて、早さを制御する事です。詳しくは、栗本･河村参照です。今回は、DC モーターに歯車を通して可

変抵抗に接続し、PICのAD変換で可変抵抗の抵抗値を読み取ることで、DCモーターがどこの位置にいるかを把握できるようにしました。

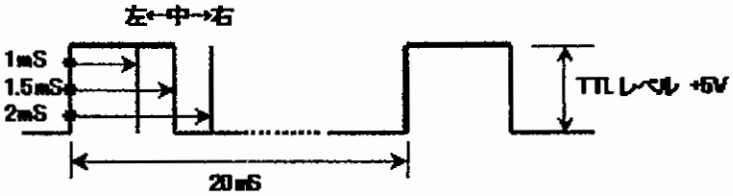
V. サーボモーター

サーボモーターを平然と書き続けてきましたが、『サーボモーターってなに？』っていう方のために、サーボモーターについて簡単に説明します。

サーボ(Servo)とは、体の位置、方位、姿勢などを制御量として、目標値に追従するように自動で作動するという意味です。つまり、モーターに信号を送ってやれば、モーターは信号によって動いてくれるというわけです。しかし、このサーボモーターは車のタイヤについているモーターのようにくるくる回るわけではありません。分度器1枚分、180度しか回ってくれないのです。不便にも見えますが、これがサーボモーターの特徴で、回らない代わりに、正確にその角度にいってくれるのです！ちなみに、右のイラストのように角度は、最初に決められた位置(0度)からの絶対値です。

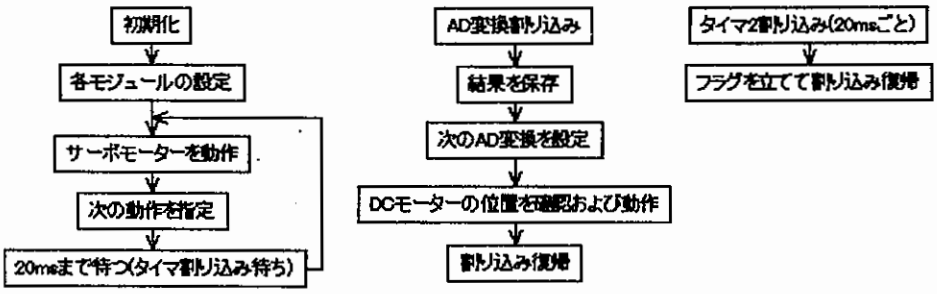


次に、入力信号について説明します。信号は20ms周期のパルス波(っていうらしい)を永遠と送り続ければいいのです。そして、角度に合せてHにする時間を変えればいいのです。しかし、このHにする時間はサーボによって違うので調べて見る必要がありますが、一般的には1ms(0度)～1.5ms(90度)～2ms(180度)です。例えばですが、このサーボを45度に曲げたいならば1ms(0度)と1.5ms(90度)の中間の1.25ms(45度)の間だけHを流せばいいって言うことです。もっとも、私のロボットに使ったサーボは、0.8ms～1.6ms～2.4msでしたが。



VI. プログラム

＜フローチャート＞



＜プログラム＞

プログラム自体は簡単なので割愛します。WEB にプログラムを一式あげています。

コンパイラは MPLAB C30 です。

≪Main.c≫

```
#include    <p33FJ256GP506.h>
#include    "config.h"    #include    "Wait.h"    #include    "data.h"
#include    "servomain.h"    #include    "pwm.h"    #include    "timer.h"
#include    "ADC33F.h"    #include    "ADC.h"    #include    "mode.h"
static int8 flag,servo[16];
static int16 flag,temp,tempx,tempy,templ,n,n_;
void __attribute__(( interrupt, auto_psv )) _T2Interrupt(void){
        IFS0bits.T2IF = 0;  WriteTimer2(54019);          flag = 1;  }
void __attribute__(( interrupt, auto_psv )) _ADInterrupt(void){
        省略。DC モーターに取り付けた可変抵抗の値を読み取り、目標の位置まで動かす}
int main(void){
        int8 co,f;
        ＜クロック設定＞
        ＜入出力設定＞
        T2_setup();          pwm_setup();          setup_adc();          //各モジュール設定
 while(1){
        for(co=0; co<f; co++){
                n = mode1(servo);                //各サーボの角度を設定=servo[16]に書き込む
                servo_con(servo);                //↑の書き込んだ値に対してサーボを動作
                while(flag == 0){}     flag = 0; //20ms になるまで待機
        }
        for(co=0; co<50; co++){
                ＜上同様サーボの設定＞
                servo_con(servo);     while(flag == 0){} flag = 0;
        }                                        //以下、同様サーボの角度を設定してループ
}}
```

≪servomain.h より。servo_con のみ。≫

```
void servo_con(int8 servo[]){              //ノットをかけるので、サーボの信号は逆に出力します
        int8 i,f,g,k,min,dif;   LATD = 0xFFFF;   k = 0;
        for(i=f=0,g=1; i<8; i++,f=f+2,g=g+2,LATD=0xFFFF,k=0){
                k |= (1<<f);          k |= (1<<g);          LATD ^= k;
                delay_us(799);
```

```
            if(servo[f]>servo[g]){
                min = servo[g];     dif = (servo[f]-servo[g]);
                delay_servo(min);   //2つのサーボのうちHが短いほうを動作
                k = (1<<g);         LATD ^= k;
            }else{
                min = servo[f];     dif = (servo[g]-servo[f]);
                delay_servo(min);   //長いほうを動作、短いサーボとの差分遅延
                k = (1<<f);         LATD ^= k;
            }
            delay_servo(dif);
        }
}
```

Ⅶ 周辺機器について

サーボモーター『GWSMICRO/2BBMG/J』(秋月)

今回のサーボモーターはすべてこれです。トルクが高く、小さく、価格もそれなりで扱いやすいです。

圧電振動ジャイロモジュール『AE-GYRO-SMD』(秋月)

角速度を測るもので2軸ついています。mogixの姿勢を見ます。

赤外線測距モジュール『GP2Y0A21YK』(秋月)

正面にある障害物までの距離をアナログ波で出力してくれる便利なモジュールです。

Ⅷ 感想

武子：

いやあ、ここまでの道のりは本当に長かったです。ほとんど何も出来なかったけど。津田と新田には本当に申し訳ないことをしました。多分、津田は新田のプレッシャーで他変だったと思う。新田は、残念な二人を持って大変だったと思う。本当に申し訳ない。

だけど、企画しているときしか一緒に活動してないけど、そのときは本当に楽しかったです。本当にありがとう！

Mogixは永遠に不滅です。

新田：

最初は何をしていいかわからなくて、回りを困らせてしまったけど、ようやくここまで来たと思ってる。ここまでこれたのは、武子、津田そして物無のみんなのおかげだと思う。本当にごめん。でも、本当にありがとう。そして、二足という未知の領域に踏み込めて本当に楽しかったし、この時間が幸せだった。

あと、少し、全力で突っ走ろうぜ！　Mogix Project成功させようぜ！

津田：

初めて作ったロボットは、いろいろと難しかったです。

金属加工やプログラムの全く分からなかった僕に、武子さんと新田さんは一から教えてくださいました。本当にこのお二人には、感謝しています。

また、二足歩行ロボットという難しい企画ができて、とても楽しかったです。

この Mogix Project に参加できて本当に良かったです。

おまけ『Mogix 誕生秘話』

筆者：武子 真行

そう、あれは夏のある暑い一日のときだった.....

物無では秋から制作物を作りだす。そのためには夏の間にいろいろ決めておかなければいけないことがある。それはその日偶然物無内でパンツだけを着用して歩いていた武子も例外ではなかった。

彼は内心苦悩していた。というのも彼にはセンスが常人より少しばかり足りていないため、物無のスペシャリストぐらいの人物と組まないと BIG な制作物が作れないのである。それは武子にとって色々遺憾ではあったが。

「はぁ、このままだと四角いアルミ箱を作るはめになっちまうよ」

その時、彼の前に一人の人物が立った。

「なっ.....お前は.....」「おっと、そこまでだ武子さん。男と男の間に余計な言葉は野暮ってもんだろ？」「茂木.....！」

その人物は茂木と呼ばれていた。人情と思いやりに溢れる熱血漢でまさに好青年の鏡だが、本名の影が薄いのが玉に傷な、まさに物無の『漢』だ。

T「どうしたんだ茂木。まさかお前も.....」M「おっと、そこまでだ武子さん。この俺に共同制作者がいないなんてこと、口にするのは野暮ってもんだろ？」T「そうだったな、悪かった。だったら俺と一緒に組んで.....」M「おっと、そこまでだ武子さん。別に組んでもいいんだが、例のスペシャリストに誘われてるから三人でやろうなんてこと、口にするのは野暮ってもんだろ？」

例のスペシャリストとは新田のことだ。コスプレが好きな頭のいい奴だぞ！

N「そんなわけであたいが協力してやるよ。まあ露出好きと熱血バカじゃあんまり頼りにはならないだろうけどな（彼はこのあと本当に頼りにできない状況になります。本当にお疲れ様です）」T「新田がくれば百人分の働きが期待できるぜ！制作物はどうする？」M「おっと、そこまでだ武子さん。男のロマンである二足歩行以外に選択肢があるわけないなんて口にするのは野暮ってもんだろ？」T「まあ確かにそうだな。じゃあそこらへんは新田工場長に一任するとして」N「あれ？工場長と言う名のリーダーに任命されてる上にさりげなくすべて押し付けられてね？（その通りです。本当にお疲れ様です）」T「ロボットの名前を決めようか。まあ妥当にガ「MOGIX」え？」M「MicomOsGraphicImage・Xsystem.頭文字を取って MOGIX。どの面からでも最高のロボットを作るという誓いを込めた名前だ」T「なるほど、いいな。採用！」N「語呂も意味もハチャメチャ www（お疲れ。）」

こうして、彼らの project.MOGIX が幕を開けたのだった.....！

改造アニマル、モモンガ／悲劇ソシテ絶望
-大空ニ憧レタ者達／末路-

製作者　H１　大森時生
M３　金子真太郎
協力者　物無員の方々

①序章　～プロローグ～

「びぎゃぁぁぁぁぁぁ！！！！」凄まじい小動物の悲鳴が、麻布動物改造研究所に響き渡った。
この動物研究所には、翼の生えたサル、カニの爪のようなものを何本も操っているイルカさらにはケンタウロスのように上半身人間で下半身が馬の生き物がいた。まさに、その生き物たちの姿は、人類科学が産み出した悲劇と言わざるを得なかった。
小動物の悲鳴は、またも新たな改造動物が創りだされたときに絞り出された声だったのだろう。

悲鳴とは、対局に一人笑い続けていた男がいる。名前はあつし。
そしてそれからこう呟いた。
「ついに、ついに、完成した。私が追い求めていたものに到達した。私は、幼いころからずっと望んでいた、大空を舞うことを・・・。飛行機のような無様な鉄の塊の力で空を浮かぶのでは無く、グライダーで醜態をさらして飛ぶのではなく、自らの力で・・・。」

そして、あつしの前には、普通の人間ならば目を伏せるような、改造動物が存在していた。
モモンガの体が、鋼鐵で覆われ、羽の部分には十数個にも及ぶモーターの数々が装着されているのだった。その改造されたモモンガの体は、無数に傷つけられ、目は死んだかのように閉ざされていた。
「今こそ目覚めの時だ！！」
男はこう叫び、改造モモンガを目覚めさせる電気ショックをかけた。
その瞬間モモンガは、目を見開き、憎しみに満ちた目であつしに焦点を合わした。その後体につけられた拘束具を引きちぎった。もうあつしの想定は、ゆうに超えていた。そして恐ろしいスピードであつしのことを・・・・。

そして世界中に、おびただしい数の、人間を憎んだ改造生物がばらまかれ、麻布動物改造研究所には何も残らなかった。ただ一人、細かい肉片になったあつしを除いては。
荒廃する世界、希望を失った人々、失われた自然、それなのに科学は引き返さないのか。

②概要

この章では、この製作物の概要を説明します。実際はストーリーで語られたように、生き物は使用していません。嘘ついてごめんなさい。文明が発展しすぎた人類に、警鐘を鳴らしたかったのです。
この製作物は、簡単に言ってしまえばグライダーのようなものです。
高いところ(麻布の非常階段3階)から放てば、風に乗って、遠く(多分麻布の砂場くらい)まで滑空していきます。
しかも、途中で障害物があると、超音波が反応してぶつからずに避けます。
次章からそれぞれのシステムについて語ります。

③本体と飛行システムについて

ここでは、モモンガ号がどのようなBODYを持ちあわしているのかということと、どうやってモモンガ号は空を飛んでいるのかということに関して記述したいと思います。

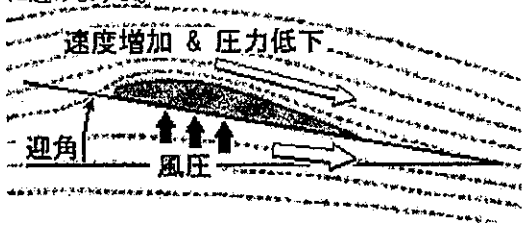
～揚力って何だろう～

みなさんは、何で飛行機が空を飛ぶのか知っていますか？
飛行機が空を浮かぶのは、「揚力」というものが働くからなんです。

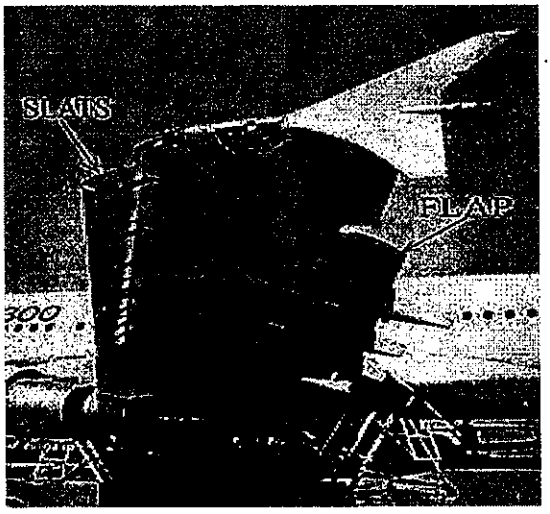
上の図を見ると分かると思いますが、みなさんが乗っているANAなどの飛行機の翼や、我らがモモンガ号も翼の断面図は、このように上面の緩やかな曲線と下面の平らな線で構成しています。何故このような形になっているかというと先述のとおり揚力を発生させるためです。今からそのメカニズムを説明します。
この翼が風を受けるとき、風は二手に分かれます。その後、翼の終わりの部分で集まります。つまり、翼の上側では、下側よりも速いスピードで風が吹き抜けたということです。
空気は、スピードが速ければ速いほど圧力は低くなるので、上側は下側より圧力は低くなります。
そして、皆さんご存じの通り、風は高い気圧のところから低い気圧へと吹くので、飛行機の翼は上へと機体を持ちあげるように風が吹きます。これが「揚力」です。
モモンガ号を始めとする世界中の飛行機は、これで空に飛び立ちます。

～フラップってなんだっけ～

ここでは、超音波で障害物を感知したとき、モモンガ号がどのように避けるかを説明します。華麗に避けますよ。



上の図を見てください。空中で揚力によって飛んでいるときに、さらなる上昇をする術のヒントがこの図には詰まっています。このように翼をかたむけることで、揚力が増大するのには二つの要因があります。
一つ目は、下側から上側へと風圧が加わるということです。この風圧が生まれるのは、傾けていることによって、下側では風が下方向に向かい、反作用としての風圧が上へと向うからなのですよ。
二つ目は、単純に迎角の増加によってカーブの長さが伸びたことです。それによって、当然の如く圧力差が増し、揚力もPOWERUPするわけなのです。
しかし、翼ごと傾けるのは難しい・・・。そこで、使われるのがフラップです。



はい、上の写真でフラップと書いてある場所がフラップです。そのまんま東ですね。
翼ごと傾けずとも、フラップだけを上下すれば同じ分の揚力は得られるのです。
モモンガ号の場合は、このフラップはサーボモーターで動かしています。
詳しくはのちほど回路の場所で語ります。
ちなみに、写真の飛行機の名前はエアバスＡ300です。

～材料のはなしとか～

この製作物は、翼をバルサ材で設計し、本体をヒノキで作っています。
理由としては、翼は揚力を受ける場所なので軽い素材で作る必要があり、本体は重心となる場所なのでしっかりとした素材で作る必要があるからです。
フラップもバルサ材で作られてあり、ワイヤーを使用してサーボモーターが上下させます。
また、揚力発生のためにつくられた丸みを帯びたカーブは、バルサ材で作った枠組みに合わして発砲スチロールを切り取りました。

そして、こんな形の発砲スチロールを大量生産してバルサ材の土台に装着します。
この作業が大変で大変で・・・、泣きそうでしたよ。

④回路について

ここでは、今回の製作で使用した超音波について記述したいと思います。

～超音波～

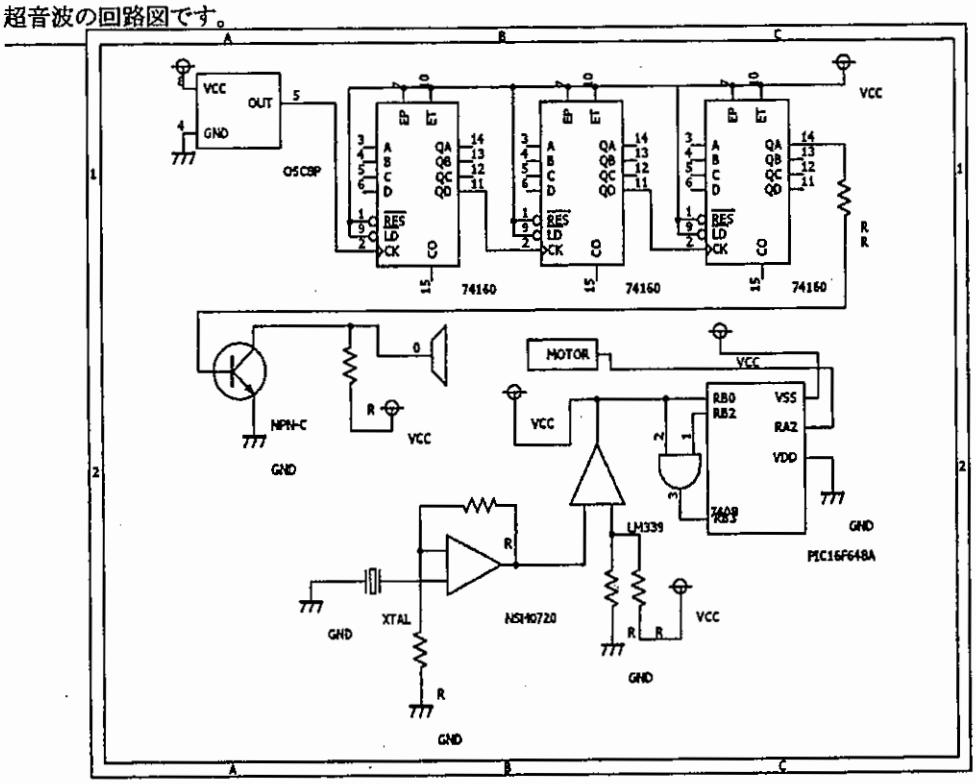
超音波とは、高い振動数のため人間には聴こえない音、のようなものです。
この超音波を本体側から発射します。その時、本体の前方に障害物があれば、超音波はその障害物にぶつかり、跳ね返ります。その跳ね返った超音波を受信する事によって、障害物があるという事を感知し、避ける事が出来るのです。

こうは言っても、超音波のイメージが湧きにくいと思いますので、身近にある超音波が使われているものを紹介していこうと思います。
まず、電動歯ブラシや骨伝導スピーカー、加湿器などと言ったものが挙げられます。

また、奥様方が使われる美顔機も、超音波が使われていますね。超音波は、美しい肌の味方です。
身近ではないですけど、魚群探知機にも超音波は使われています。
動物でも、超音波を利用している奴らがいますね。みなさんご存じコウモリ、そしてイルカもです。
コウモリは、超音波で獲物の位置を捉えているので、暗いところもへっちゃらなのです。
イルカは、超音波でイカなどを弱らして、食すのだそうです。怖いですね。

みなさんに、超音波のイメージが湧いていたら幸いです。

超音波の回路図です。



⑦参考文献

製作するにあたって、参考にしたホームページや書物を作成した方々に敬意を称し、ここに記しておこうと思います。

・2006年度物理部無線班回路図集
・2007年度物理部無線班回路図集
・2010年度物理部無線班回路図集

・http://www.yp1.yippee.ne.jp/launchers/hlg/hlgindex.html
・http://j7w1.info/jisakubu2/G-6.htm

後世の飛行機創る物無員とかに役立てばうれしいです。

⑥感想

金子：今年はチカチカ隊長として製作をしてきた訳ですが、今この文章を書いている3月後半、製作についてはサーボモーターさえ動いていない状態です。チカチカ隊長としての仕事があったとはいえ、ここまでの遅れは完全に自分の計画不足だと思います。その上、大切な春休みを地震の影響で全て潰されてしまいました。新学期には新入生が入って来る為、活動出来る時間は更に少なくなると思います。残されている時間、全力で製作、チカチカ共に進め、文化祭でこのモモンガが飛ぶ姿を見せたいと思います。

大森：初めての後輩との共同制作だったのですが、非常に楽しかったですね。そして最高でした。製作初期の9月ごろからすでに異彩を放ち、二人そろってインスピレーションを沸かしまくりました。しかし、3回目の製作だというのに反省するべきことは、たくさんありますね。やっぱり一番大きい反省は「妥協」です。妥協しないということは、自らが製作したことに満足するには必要不可欠だと思いますし、どんなことでも追い求められる物無員の口かは発してはいけない単語だとすら思います。金子は、チカチカも含めて本当に頑張ってくれたので、この場を借りて感謝します。ありがとう。そして、物無のみなさまやOBの方々にも非常にお世話になりました。本当にありがとうございました。今年一年金子とギャーギャーと共同制作したこととか、朽木とハチャメチャにゲーム作っていったことはナチュラルに一生忘れないんじゃないかな、なんて思ったりなんかもしてます。
二人も(もちろん、全ての物無員も)そうあることを望んでいます。
あと、文化祭当日一番人気が、モモンガ号であることも望んでいますよ笑

⑥終章　～エピローグ～

空を飛ぶことは；人類の夢だった。
僕らも例外では無かった。
素晴らしい夢を抱いていたのかもしれない。
ライト兄弟の気持ちを僭越ながら理解できた気がする。
この章は間埋めのことは、誰もが気付いていることでしょう。

HeVin mk-Ⅲ

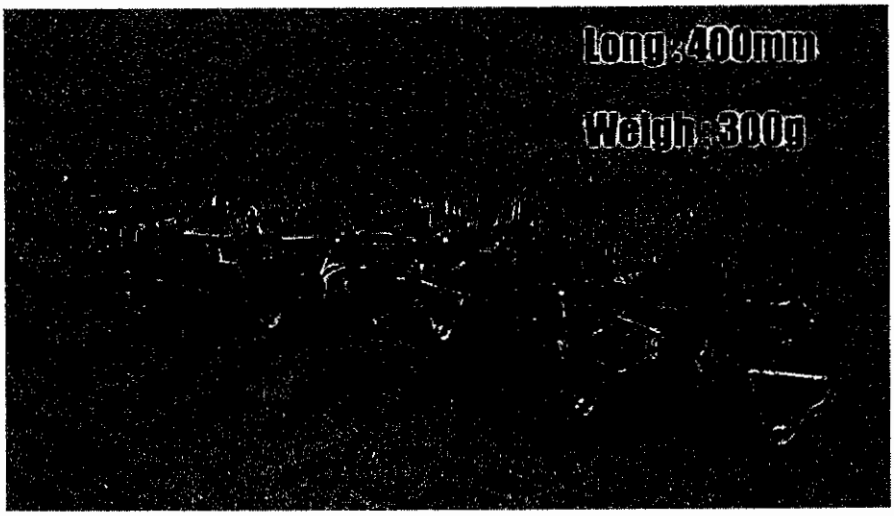
-Snake Mobile Automatic Wireless Shooting-

制作： H1 栗本 郁也 / M3 河村 洋一郎　　協力： 物無の皆様 / ネットの皆様

prologue

そう・・・その昔〇龍こと某ヘビ型ロボットを目指した計画が物無にはあった・・・
しかし、彼らは誓った。それはヘビ型ではないと・・・
というわけで連結車ではなく、体をくねらせて進むスマートなヘビ型ロボットの製作が始まったのである。

What's This?



ヘビ型ロボットです。車輪には動力を持たせずに関節のサーボモーターで体をくねらせて
推進力を発生する仕組みに挑戦しました。体全体が動くため、本体とコントローラーの通信は無線にしました。
ロボットの動きを大まかに制御するために赤外線通信でのコントローラーを製作しました。
マイコンは PIC32MX440F512(本体) & PIC18F2550(コントローラー)です。
どちらも USB 通信の機能を持ったマイコンです。

mk-Ⅲ は mk-Ⅱ までと違いサーボモーターを、アルミ板も 1mm 厚を使い、小型で軽量になりました。

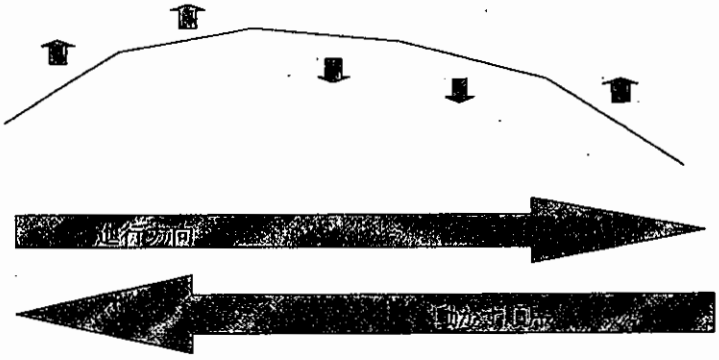
※平面方向限定です。鎌首もたげはご容赦ください。
※副題は理想です。現実は厳しいのでそこまで完成しないかもしれません。
※写真は 5/17 時点です。6 関節 7 車両です。

本体

☆動作

"ヘビ"のような進み方についてです。
実際のヘビの進み方は"サーペノイド曲線"と呼ばれる曲線に近似しているらしく、
人間たちはその曲線に基づいてロボットを作ってるわけです。
もっとも、蛇行の際に体の一部を持ち上げたりするらしくまだまだ研究されているとかいないとか。
まぁひも状のものが進むためにはそれなりに工夫する必要があるということです。

分かりやすく言うと車輪の横向きの摩擦を利用してエッジをたてるようにして関節で推進力を得て
前の車両を滑らせていく感じです。

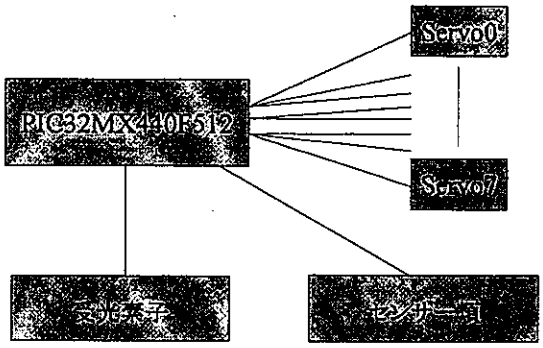


上の図のように先頭からだんだんと左右の一定の角度まで動かし、反転させていくことを繰り返して進みます。

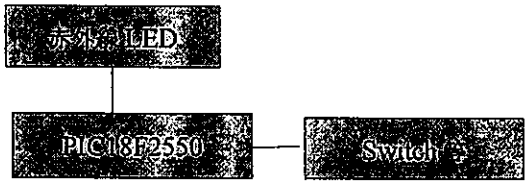
回路

☆ブロック図

本体側



コントローラー側



☆サーボモーター

サーボモーターは絶対角度を指定するとそこに向かって動いてくてるモーターです。

みなさんが思い浮かべるであろう、回転し続けるモーターとは違い、180°程度しか可動範囲がありません。

制御基板とギアボックス(モーターと歯車)とポテンションメーター(角度がわかる)が一緒になっています。

今回は左記のようなサーボモーターを使いました。

よく使われるタミヤのギアボックスは 2kg 程度のトルクですし、

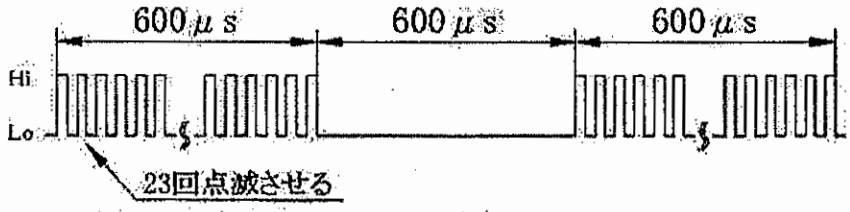
サイズもこちらのほうが小さいですね。

モーター用の+と-の他に信号線が 1 本ありまして、そこに 20Ms(ミリセカンド)中に H を出す長さによって絶対角度を決めます。このサーボは 800μs-2400μs です。

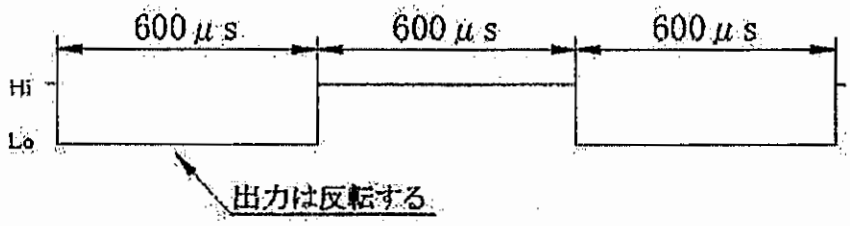
なお、サーボは起動時に大きな電流を流すため、複数のサーボを動かす際の、動かすタイミングをずらすようにしました。

☆赤外線通信

お茶の間でテレビのチャンネルをかえるそのリモコン。それも赤外線であることが多いでしょう。
また携帯電話の赤外線通信機能のお世話になった人は多いでしょう。
ここでは38khz受光素子と赤外線LEDを使って通信しています。



送信側:赤外線LED点滅状態



受信側:受信モジュールの出力状態

↑こんな感じです。受光素子は600μsごとに"38khzの信号"が入っていればL否ならHを出力します。
これを利用して通信するわけです。

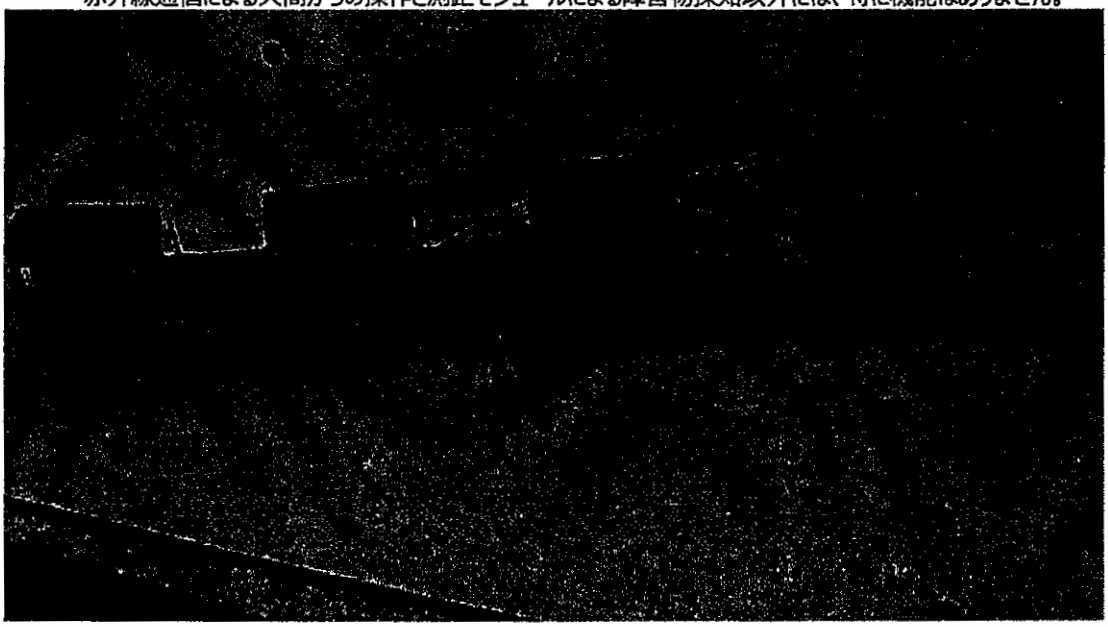
☆測距モジュール

これまた赤外線の測距モジュールですね。
赤外線LEDとPSD(position sensitive detector)を組み合わせた物です。
PSDとは光量に応じた電圧を発生する材料を利用して対象の距離とか位置を測る物です。
ここでは反射する赤外線の光量で電圧が変化します。
これをマイコン側でA/D変換して、電圧を数値化して距離を求めるわけです。
今回使ったものは10cm-80cm位の距離を測ることができます。
これを利用して壁にぶつからないようにすることができます。

試験機たち。

☆mk-Ⅰ

いきなり３次元運動をするのは困難に思えたため、
平面上での蛇のような動きを再現しようと製作したのが壱号機です。
赤外線通信による人間からの操作と測距モジュールによる障害物探知以外には、特に機能はありません。



タミヤの一般的にトルク2kgのギアボックス１つを使った関節が４つあります。
なんとか平面方向に進ませることができたものの、以下のような反省点が挙げられました。
・より人車両が長すぎて効率的に動けない。
・関節の消耗が激しく、故障することが多い。

☆mk-Ⅱ

３次元運動を前提にしたのが弐号機です。
こちらも同じギアボックスを使い、関節は縦運動と横運動のものを交互に入れました。
関節の殆どはモーターで埋まっており、基板を従来より縮小することで対策しました。
しかし、以下のような点から製作を断念しました。
・ギアボックスの大きさから、本体の制作に時間を要した。
・ギアボックスが部活に余ってただけで、実はサーボモーターの方がトルクも上で価格もそこまで変わらなかった。
　ならば角度制御がしやすいサーボモーターを使うべきだった。

☆Dcモーター

試験機でお世話になったモーターについて解説。
たくさんモーターを使うこと、大きなトルクが必要だと予想されることを理由にDCモーターを使いました。
しかし残念ながら角度制御にはサーボ or ステッピングのほうが優秀です。
トルクの高いサーボもありました。
制御方法はHブリッジという回路を使っています。
マイコンの出力にトランジスタアレイTD62083Aをかませて Hブリッジ回路を内蔵したMP4212というFETアレイを使いました。

☆PWM

PWM(パルス幅変調方式 Pulse Width Modulation)はモーターの速さを制御するのに使います。原理は簡単でモーターを高速でON、OFすることで擬似的に電圧を下げているというものです。
PWM制御をするためにはモータードライバにパルスをおくる必要があります。PICの内蔵タイマを二つ使ってやりました。

☆FET

モーター制御でお世話になったFETの解説。
ラジコンの性能向上と小型化や電源アダプタなどいろいろなところで役に立っています。電子回路の中での動作を比べると、トランジスタは入力信号の電流が増幅されて出力電流になりますが、FETは入力信号の電圧が増幅されて出力電力となります。
今回使っているのはMOS-FETという種類のFETです。
これを使うことで大きな電流が必要になるDCモーターを動かせたわけです。

☆感想

栗本

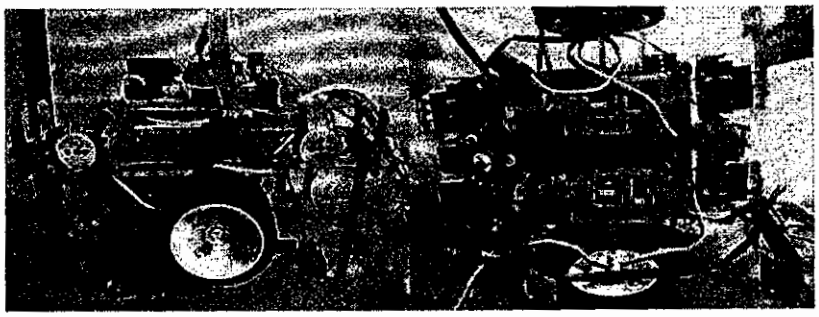
はい。今回の製作では序盤は河村君に教えつつ、dspic33を動かしました。mk-Ⅰは順調だったと言えるでしょう。しかし、mk-Ⅱを作るに当たってやる気がなくなりました。そしてPIC32とかSTM32とかに手を出してたらキャパごえした上、本体のほうを任せすぎて、互いにモチベ最低になった時期がありました。具体的な計画がなさすぎたせいだと思います。ごめんね。そしてmk-Ⅲ。とりあえず完動させんとしたら割とスムーズに進みました。
互いにあんまり新しいことをできなかったのが心残りです。

河村

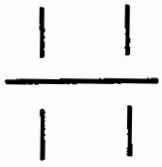
今回の制作は、本格的にプログラムを行った初めての制作で、戸惑うことも多くありましたが栗本さんを始め多くの先輩方に様々なことを教えていただき勉強になりました。最初は完璧に見えてもなかなか動かない「プログラム」には戸惑いましたが、またそこも醍醐味なのだと思います。Hevinでは微妙な調整がネックとなりましたが、まっすぐ動いたときはとても嬉しかったです。今回の制作で、おおよその基礎的な知識は習得できたと思うので、その部分も含めて今回の制作は有意義だったと思います。

箱丸～超効率的四足歩行ロボット～

製作者　H1 市村剛大,橋本佑由
太M3 村瀬空

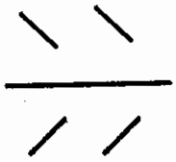


このロボットは効率的に歩行するためだけに作られた超近未来的ロボットで、地球上のどんな地形にも自ら状況判断して対応し、目的地にたどり着くことのできる万能ロボットなのである(パチ
本当の事を言いますと、四足歩行ロボットに車輪(二輪)をくっつけた四足二輪ロボットです。
普段は四足歩行しますが、何か危険(?)を察知して二輪走行に切り替え、猛スピードで逃げるロボットです。
赤外線測距モジュールによって障害物を感知して回避したり、赤外線による通信(テレビリモコンのような)やUSB通信を使ってロボットを無線操縦ができたりします。
まずは四足歩行の仕組みから解説しましょう。



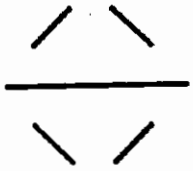
まず、左の図１は何を表わしているかというと、真ん中の線が本体(基盤類)を省略したもので、４本の縦棒が足を表わします。
これらの図の右方向(ロボットの前方向)に進む歩行法を説明します。

(図１)　　初めこのロボットは図１のような棒立ちの格好をしています。



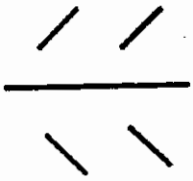
次に図２のように足４本を(上からみて)左に動かし、本体を右に移動させます。

(図２)



さらに図３のように(図で言う)右の２本(実際には前足２本)を固定しながら左２本(実際の後ろ足２本)を内側に曲げます。

(図３)



今度はさっきとは反対の左２本(実際の後ろ足)固定しつつ右２本(実際の前足)を外側に曲げます。

この後、４本をまっすぐにして図１のようにすれば、右に進んだ状態になります。

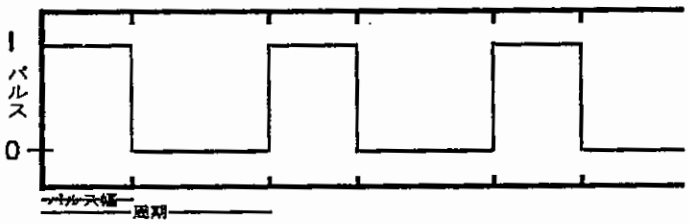
(図４)　これを繰り返すことで歩行ができる、というわけです。

～ここから村瀬～

このロボットはラジコンのようにタイヤで進んだり、曲がったり、戻ったりできるわけですが、タイヤを回すのにはモーターが必要です。このロボットにおいて、モーターはかなり一般的なＤＣモーターを使うわけでして、ここではそれをどう制御してるのか、少しふれてみたいと思います。

さて、ＤＣモーターを制御するにあたって、僕はPWMという制御方法を使いました。

PWM とは Pulse Width



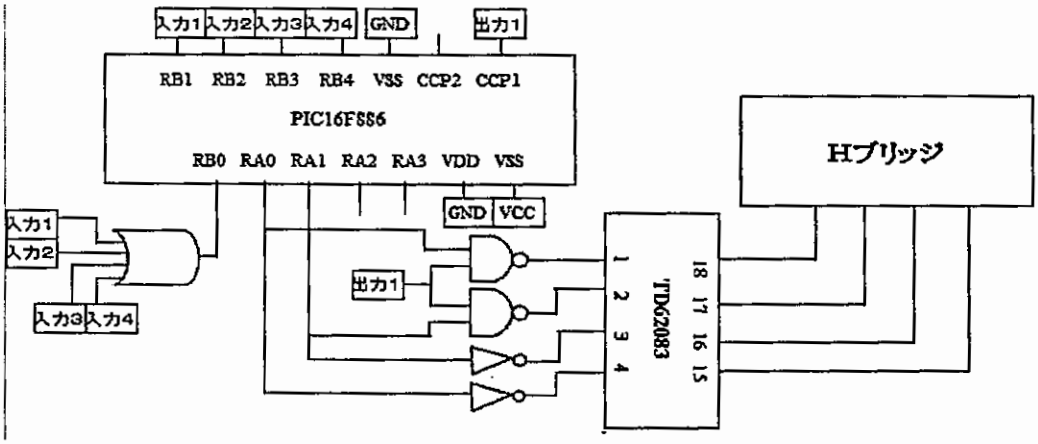
Modulationの略で、パルス波のデューティー比を変化させて変調する変調方法とのことです。また、デューティー比とは周期的なパルス波を出したときの周期とパルス幅の比のことで、パルス幅を周期で割ったものとなります。イメージは下の図です。

つまり、PWMは、一定時間内でどの程度、High（５Ｖなどの固定値、０ＶだとLow）を出力するかを制御することができる制御方法というわけです。これを利用してDCモーターの速さを制御するわけなのですが、いったいどう利用するのでしょうか？

さて、モーターの速さを制御するためには、単純に考えれば電圧の大きさを変えればいいということになりますが、それはＰＩＣだと難しく、デジタル（前述のHigh,Lowだけの世界）ではないために、効率も悪くなってしまうのです。そこでデジタルのPWMを使い、モーターを回しては止めるという作業を高速で繰り返すことによって、見た目上、速さを変えることができるのです。

また、向きを変えるのにはモーターを二つの方向に回すことができるＨブリッジという回路を使います。これを二つ使えば、曲がったりすることもできますね。

モータ制御に使う回路



ポートAの2,3は0,1と同じ様になるので省略します。他にもいくつか省略しています。
入力のいづれかが入ると割り込みが入り、モーターが回るようになっています。
もちろん、回る向きもどの入力が入るかによって変わります。

PWMのプログラム（アセンブラです）**SETUP**

```
        BSF     STATUS,RP0
        BSF     STATUS,RP1
        MOVLW   B'00000000'
        MOVWF   ANSEL
        MOVLW   B'00000000'
        MOVWF   ANSELH
        BCF     STATUS,RP1
        MOVLW   B'00000000'
        MOVWF   TRISA
        MOVLW   B'00111111'
        MOVWF   TRISB
        MOVLW   B'00000000'
        MOVWF   TRISC
        BCF     STATUS,RP0
SETUP_PWM
        MOVLW   00FH
        MOVWF   CCP1CON
        CLRF    CCPR1L
        MOVLW   00FH
        MOVWF   CCP2CON
        CLRF    CCPR2L
SETUP_TMR
        MOVLW   04H
        MOVWF   T2CON
        BSF     STATUS,RP0
        MOVLW   0FFH
        MOVWF   PR2
        BCF     STATUS,RP0
SETUP_INT
        BSF     INTCON,INTE
        BSF     INTCON,GIE
MAIN
        MOVLW   B'00000000'
        MOVWF   PORTA
        CLRF    CCPR1L
        CLRF    CCPR2L
        NOP
        NOP
BUNKI2
        BTFSS   PORTB,2
        GOTO    BUNKI3
        CALL    START2
        GOTO    $-3
BUNKI3
        BTFSS   PORTB,3
        GOTO    BUNKI4
        CALL    START3
        GOTO    $-3
BUNKI4
        BTFSS   PORTB,4
        GOTO    STOP
        CALL    START4
        GOTO    $-3
START1
        MOVLW   B'01111111'
        MOVWF   CCPR2L
        MOVWF   CCPR1L
        MOVLW   B'00001010'
        MOVWF   PORTA
        RETURN
START2
        MOVLW   B'01111111'
        MOVWF   CCPR2L
        MOVWF   CCPR1L
        MOVLW   B'00001001'
        MOVWF   PORTA
        RETURN
START3
        MOVLW   B'01111111'
        MOVWF   CCPR2L
        MOVWF   CCPR1L
        MOVLW   B'00000110'
        MOVWF   PORTA
        RETURN
START4
        MOVLW   B'01111111'
        MOVWF   CCPR2L
        MOVWF   CCPR1L
        MOVLW   B'00000101'
        MOVWF   PORTA
        RETURN
STOP
        CLRF    CCPR2L
        CLRF    CCPR1L
        CLRF    PORTA
        CLRF    PORTC
        BSF     PORTA,4
        RETFIE
        END
```

だいたいこんな感じです。

実際は測距モジュールをつけるのでもう少し複雑になりますが、これでも十分ラジコンになります。機会があればやってみてくださいね。回路図がこれだけでは不十分に感じるかもしれませんが…

ではここで、測距モジュールという言葉が出てきたのでそこを紹介しようとおもいます。

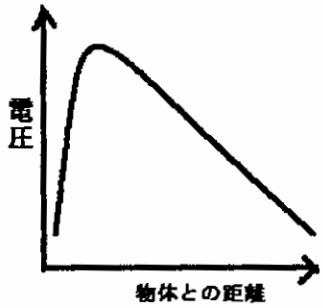
●測距機能

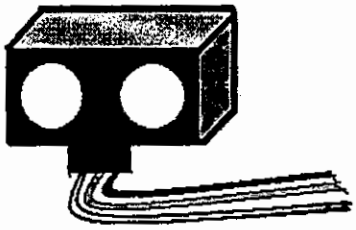
なんか堅苦しい見出しになってしまいましたが、やってることは簡単です。このロボットにはタイヤでの走行中に目の前に大きな障害物にぶつかると目の前で止まるか迂回できる機能をつけようと考えています。というか、実際に搭載します（３月現在）。そこで必要になるのが、物体と自機との距離を測る機能です。これが無いと、人間のように目があるわけでは無いので物にぶつかってしまいます。そして、この距離を測るのに必要になるのが測距モジュールであるというわけなのです。

○　測距モジュールとは

名前の通り物体との距離を測るモジュールですが、ここではどのようにして距離を測るかについて紹介します。まず、下の絵をみてください。パルタの顔みたいですね。これが測距モジュールの絵なのですが、このパルタの目に見えるものは実は赤外線 LED なんです。そして、ここから赤外線（目に見えない光のひとつ）が出て、物体に当たります。するとそれが反射して測距モジュールにかえってくるのですが、測距モジュールは、そのかえってくる量に応じて出力（白い導線）の電圧の大きさを変えてくれるのです。

つまり、その電圧の大きさによって物体との距離がわかるのです。ちなみにその距離と電圧の大きさの関係はだいたい下のグラフの様になります。





○　A/D 変換

さて、障害物を目の前で回避するにはさらに、この電圧の変化をＰＩＣでを読みとらなければなりません。そこで必要なのが、アナログの値の入力（電圧の変化）をデジタルに置き換えるということ、すなわち　A/D 変換　を行うことなのです。そして、ある電圧の値をきめておいて、それより高い電圧なら High、それより低い値なら Low を出力するようにしてやれば良いのです。この出力を先ほどの DC モーターの制御と組み合わせてやれば障害物の回避が可能になりますね。

○　A/D変換の回路

さき程の測距モジュールの出力をA/D変換に必要なポートにつなぎ、ある電圧の値より実際の電圧が上か下かをHighかLowで出力するポートを決めておけばそれでＯＫです。あとはパイロットランプをつけるくらいでしょうか。ＰＩＣは16F886を使いました。

○A/D変換のプログラム

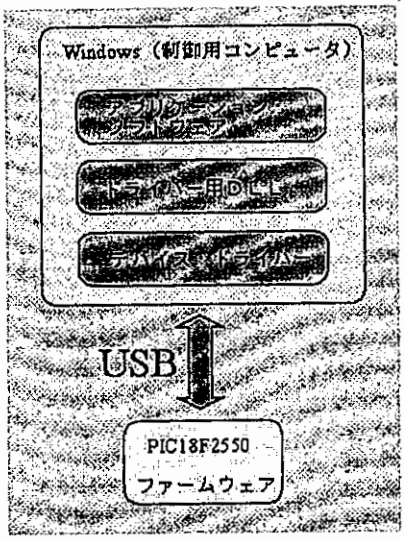
```
AD_DATA_H  EQU  25H
AD_DATA_L  EQU  26H
COUNT  EQU      30H
       ORG      0
       GOTO     START
START
       BSF      STATUS,RP1
       CLRF     ANSELH
       MOVLW    B'00000000'
       MOVWF    ANSELH
       BSF      STATUS,RP0
       MOVLW    B'00000000'
       MOVWF    ANSEL
       BCF      STATUS,RP1
       MOVLW    B'00000001'
       MOVWF    TRISA
       MOVLW    B'10001110'
       MOVWF    ADCON1
       MOVLW    B'00000000'
       MOVWF    TRISB
       MOVLW    B'00000000'
       MOVWF    TRISC
       BCF      STATUS,RP0
       MOVLW    B'10000001'
       MOVWF    ADCON0
       CLRF     PORTB
       CLRF     PORTC
       CALL     AD_CONVERT
       CALL     WAIT0.5S
       GOTO     $-2
AD_CONVERT
       BCF      PORTA,2
       NOP
TIME
       MOVLW    D'23'
       MOVWF    24H
       CALL     @WAIT_1LOOP
       NOP
       NOP
       NOP

       BSF      ADCON0, GO
       BTFSC    ADCON0, GO
       GOTO     $-1
       MOVF     ADRESH, W
       MOVWF    AD_DATA_H
       MOVWF    PORTC
       BSF      STATUS,RP0
       MOVF
       ADRESL, W
       MOVWF    AD_DATA_L
       BCF      STATUS,RP0
       MOVWF    PORTB
       BTFSS
       AD_DATA_H,1
       GOTO     $+8
       BTFSC
       AD_DATA_H,0
       GOTO     $+6
       MOVLW    B'01100110'
       BSF      STATUS,RP0
       SUBWF    AD_DATA_L,w
       BTHSS
       STATUS,C
       GOTO     $+3
       BCF      STATUS,RP0

       RETURN
       BCF      STATUS,RP0
       BSF      PORTA,2
       MOVLW    B'11111111'
       MOVWF    PORTB
       MOVLW    B'11111111'
       MOVWF    PORTC
       CALL     WAIT
       RETURN
@WAIT_1LOOP
       DECF     24H
       BTFSS    STATUS,Z
       GOTO     @WAIT_1LOOP
       RETURN
WAIT   ;お好みで決めてください
       RETURN
        END
```

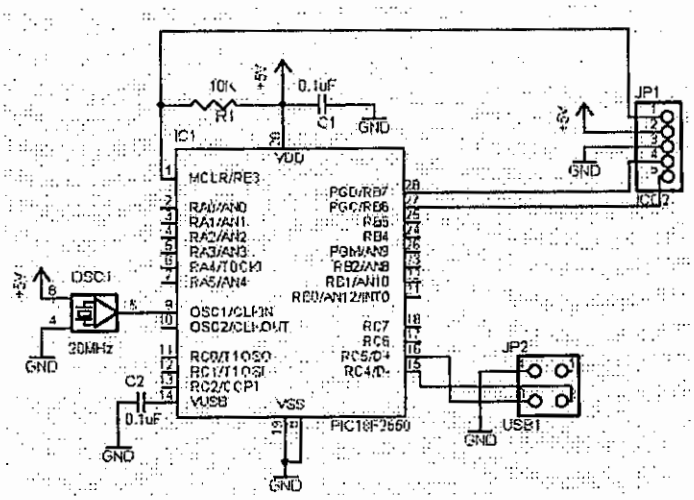
上で述べたことをやっていますが、それに加え、ポートＢとＣを使ってA/D変換が見えるようにしています。また、３月現在ちゃんと動作しています。

～ここから橋本～



僕は基本、本体製作を担当していましたが、すぐに本体を作り終えたので PIC による USB 通信にチャレンジしました。
USB 通信というのは PIC'(マイクロコンピュータ)と制御用コンピュータとを USB で接続してデータをやりとりする、というものです。
USB において、同じデバイス・ドライバーで動作するデバイス（device）をデバイス クラス（devices class）と言います。例えば、マウス、キーボード、ジョイスティックなどのデバイスは、ボタンなどの数などは違いますが、同一のデバイス・ドライバー で動作するので、Human Interface Devices（HID）クラスと呼ばれています。HID クラスのデバイス・ドライバーは、すでに Windows の OS で用意してあり、デバイスを制御用 コンピュータに接続する時に、自動的に組み込まれますので、手軽に利用できます。
なお、このクラスの規格については、http://www.usb.org/developers/hidpage にてどうぞ。
また、マウスやキーボードの模倣動作をするプログラムが microchip の方から公開されていますので、それを参考にプログラムを書きました。

テスト中の回路ではありますが、以下にあげておきます。



感想

今思っててることを言うと、僕にとってはこれからが勝負なので今から文化祭まで全力で頑張りたいということです。他の二人には感謝しています。今の形が出来てるのは二人のおかげでしかありません。なんかコミニケーションが取れていないとかの問題もありますが、文化祭が７月になるなんていう話もあり、もしそうなれば２ヶ月も増えるのでしっかり改善した２ヶ月と思います。現時点での感想は、共同制作はやっぱり難しいなということと、そして自分みたいに電子工作に長けていない人間はもっと活動の量・質・準備を改善する努力をすべきだったという後悔です。(市村)

自分の担当したのは本体と書きましたが、本当にあっという間にできてしまい、少し感動していました。しかし後述したUSB通信は回路は出来たものの、プログラムは全く動作せず、焦っている最中であります。

文化祭が延期であっても、４月中には完動させたいと思います。なかなか３人での活動は少なかったように感じますが、後輩(村瀬)に少しでも去年自分がやったことが伝わってものになってくれたことを嬉しく思います。(橋本)

はい、一応僕の担当したA/D変換とPWMは動いてるのですが、３月現在、A/D変換とPWMを組み合わせるのは、まだという状況…。しかし、というかもちろん動かします！とまあ、これが現実となっているといいのですが…。とはいっても独立してならちゃんと動いているんだから難しいわけないでしょう。文化祭延期ならなおさらですよね。とにかく動かすといったら動かさないと僕の心がつらいです。

とにかく、中二から中三にかけていろんなことを学びました。本当に力をかしてくださった先輩、同輩、後輩のみなさんありがとうございました！！あと節電がんばります。(村瀬)

Attention Please!!

製作：中３　岸田聖生

協力：物無員の皆様

■ストーリー

「もうこんなのテ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ーーーーーーーーー！」

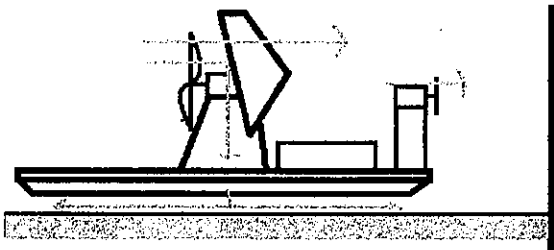
■概要

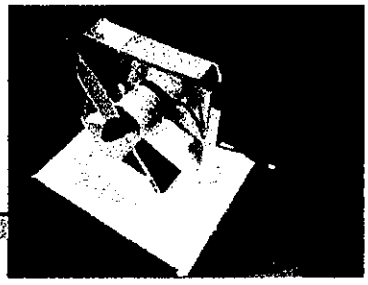
・去年に引き続き、兵器・・・じゃなくて高電圧なモノをつくろうと思ったんですが、大人の事情によりその夢は叶えられませんでした。諦めてはいませんのでどこかで作ってると思います。ヘリコプターなんてなかった。

・さて、簡単にこれがどういうモノなのか説明すると。ホバークラフトです。

夏休みの自由工作！とかに作ったあれ？そうですあれです。簡単じゃねーか。そう思った貴方は正しい。実はあんま難しくないんです。いや、Railgun やらヘリコプターやらをつくろうとしてたら時間がなくなっちゃって・・・・文化祭一ヶ月前から作り始めてたり部品流用したりで現在鋭意製作中です。

◆詳細・・・？　左は最終イメージ、右は現在の状況(見えにくいかも)





・重さ：400g 位　・速度：操縦者の足の速さまで　・モーター：GWS-300

・通信方式：有線リモコン制御(爆笑モン

・素材：プラ板、アルミ板 etc

◆仕組み

簡単に、ホバークラフトについて説明します。

① 上にある吸気口から空気を吸込み、下に吐き出す。

② スカートの中に貯まる

③ スカートと地面のあいだから常に空気が漏れてくる(浮く)　わー簡単

◆実を言うと

最初に話した通り、ちょっとした工作程度でもこれと同じ原理のモノは作れます。風船とCDとストローで作るのは有名だと思います。これは、モーターで空気を取り込んでいるだけです。めちゃくちゃクオリティ低いです。

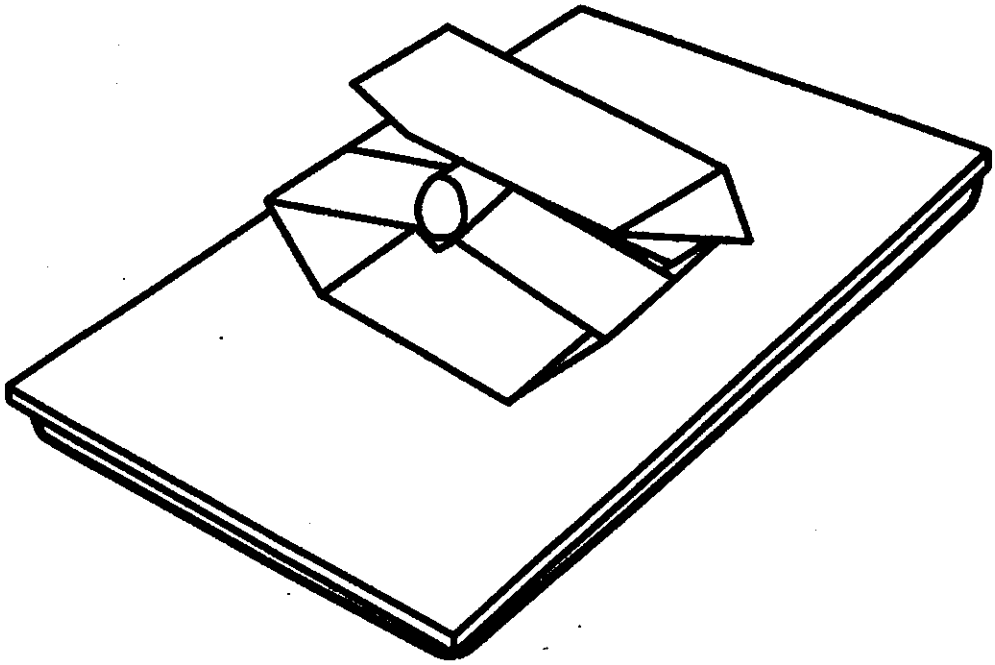
◆現在(11/6/1)

いろいろあってちょっと前につくり始めたホバークラフトです。今の機体は三号機。モーターの反動を消る、逆転プロペラがないので現在は推力と浮力を同時に得る構造になっています。反転プロペラさえあれば別のモーターで作れるのですが・・・。あと有線制御とか笑うしかないです。後々バッテリー搭載＆ラジコン化したいです。あと、プロペラは実は紙コップなんかで作れるって噂もあるとかないとか・・・・

■さいごの方に

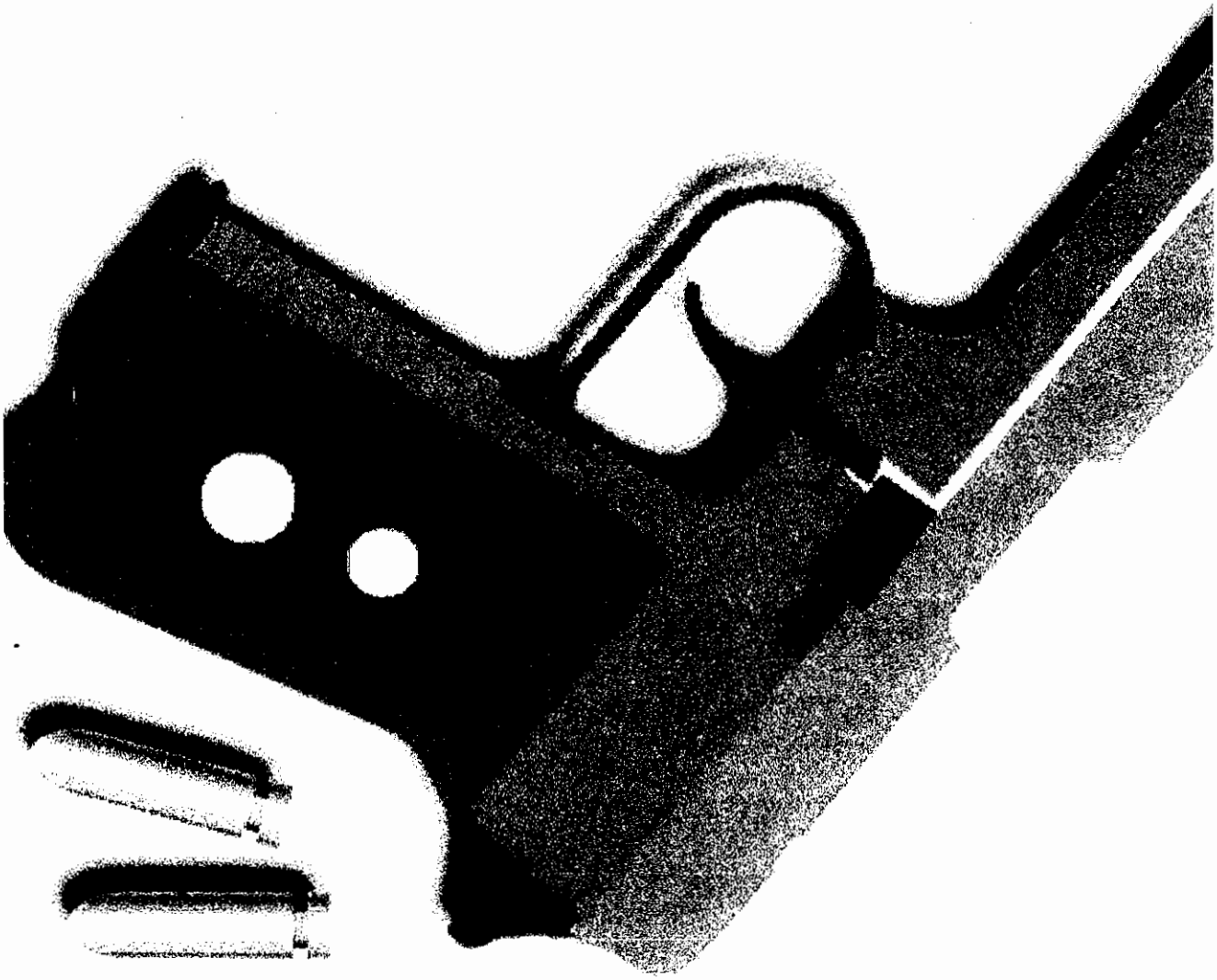
ここまで読んでくださってありがとうございます。

僕自身、なにやってたんだかよくわかりません。レールガンをつくって、止められ、ヘリコプターをつくろうとして諦め、今のに落ち着いた訳ですが。技術も大してなければやる気もでないという最低なループに陥っていました。文化祭まであと20日ちょっとなので、もうちょいとクオリティの高いものにしたいです。



はて。こうなるのだろうか。↑イメージ

GAME



コロンブスの卵

製作者　高2 伊藤 卓
中3 山本 諒一
協力者　物無の皆様

～ストーリー～

ここは今の世界とは違うどこか別の世界・・・・・・・

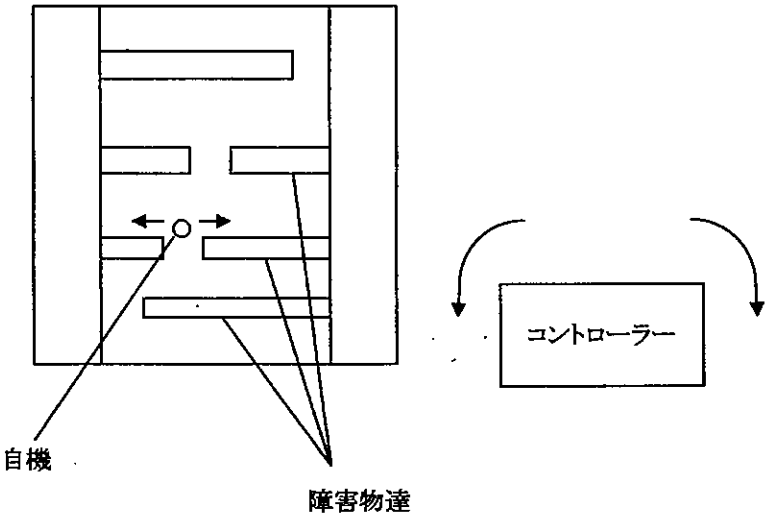
かの有名な新大陸を発見したコロンブスは、この世界では航海にでたものの何も見つけられなかったただのおちこぼれと化していた。

彼の有名な逸話である「コロンブスの卵」の話もこの世界では自分で話をふっておきながら、自分でそのことができなかったという恥ずかしい話として受け継がれてしまっている。

そんな中コロンブスが立てられなかった卵は華麗に障害物をよけながら地面を転がり続ける。人生の障害物に当たりまくってしまったコロンブスとは対称的に・・・・・

～ゲーム説明～

このゲームは障害物を避けて進みゴールを目指すゲームです。自機は緑色 LED、障害物は赤色 LED で表示されます。障害物に当たるとゲームオーバーになります。コースがいくつかあり、それぞれ最後まで進むことができればゲームクリアになります。そしてこのゲームの特徴として、コントローラーを傾けることによって自機を操作するという仕様になっています。



～ジャイロセンサ～

ここではジャイロセンサについて説明していきます。

・ジャイロセンサとは？

ジャイロセンサというのはその名の通りセンサーなわけですが、何のセンサーかというと物体の角度や角速度を検出するセンサーです。これを使うと物体が回転運動をした時、その物体がどれだけ回転したかとかどのぐらいの速さで回転したかとかがわかります。

今回使ったのは村田製作所の圧電振動ジャイロを表面に実装した、秋葉にある秋月という店で売られている圧電振動ジャイロモジュール（AE-GYRO-SMD）です。

このジャイロセンサは物体にかかるコリオリの力を検出することで角速度を電気信号として出力しています。コリオリの力というのは、簡単に言うと回転するものの上で移動したときにかかる力のことです。きっと中学2年生ぐらいに習うでしょう。

・ジャイロセンサの制御

ジャイロセンサは角速度（回転する速さ）を電気信号として出力します。たとえば右が正だとして右に回した後、最初の位置まで左に回すという動作をすると図1のような出力になります。しかしこの出力はあくまでも角速度であり、ゲームの仕様上角度を求める必要があります。そこで、しなければいけないのが積分です。この出力を積分することによって角度の出力がでます。図1の出力を積分してでた角度の出力は図2のようになります。

ただこれはジャイロセンサが理想の動きをした時にでき、実際のジャイロセンサは全く回転運動していない時出力が一定ではなく図3のように上下しているため、それを積分しただけでは誤差が出てしまいます。

図1

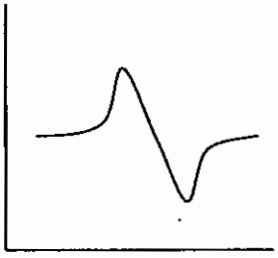


図2

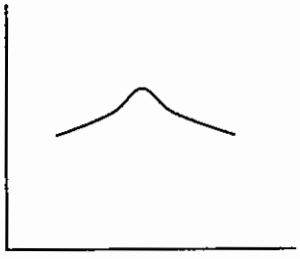
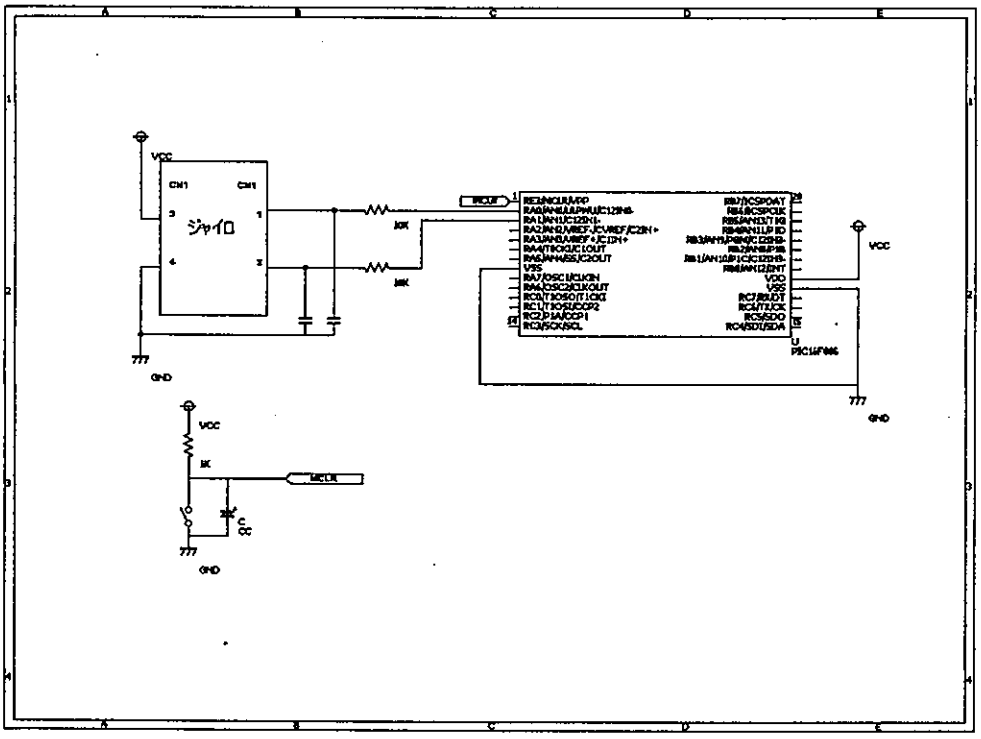


図3

・回路図



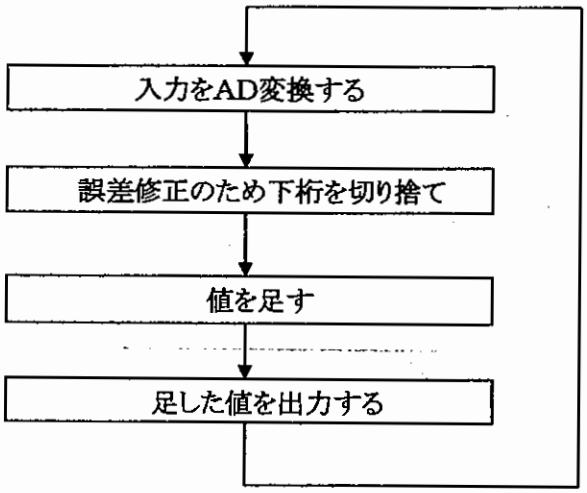
今回、積分を積分回路を使って行うのではなく、PICを使うことによって行っています。そのため回路はジャイロの出力をただPICにつなげるだけの回路となっております。使ったマイコンはPIC16F886です。

・プログラム

プログラムは、ジャイロの入力はアナログになっているので、PICで読み込むためにまずAD

変換をし、その数値をどんどん足していくという単純なものになっております。プログラムは長いのでフローチャートだけ書きます。

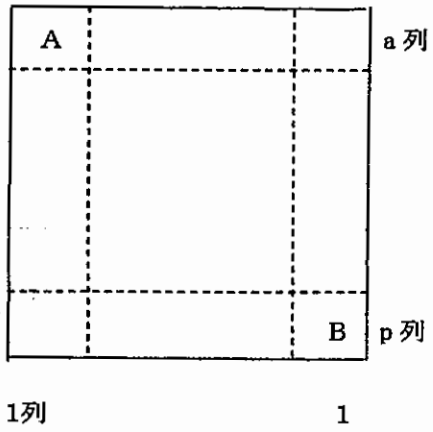
フローチャート



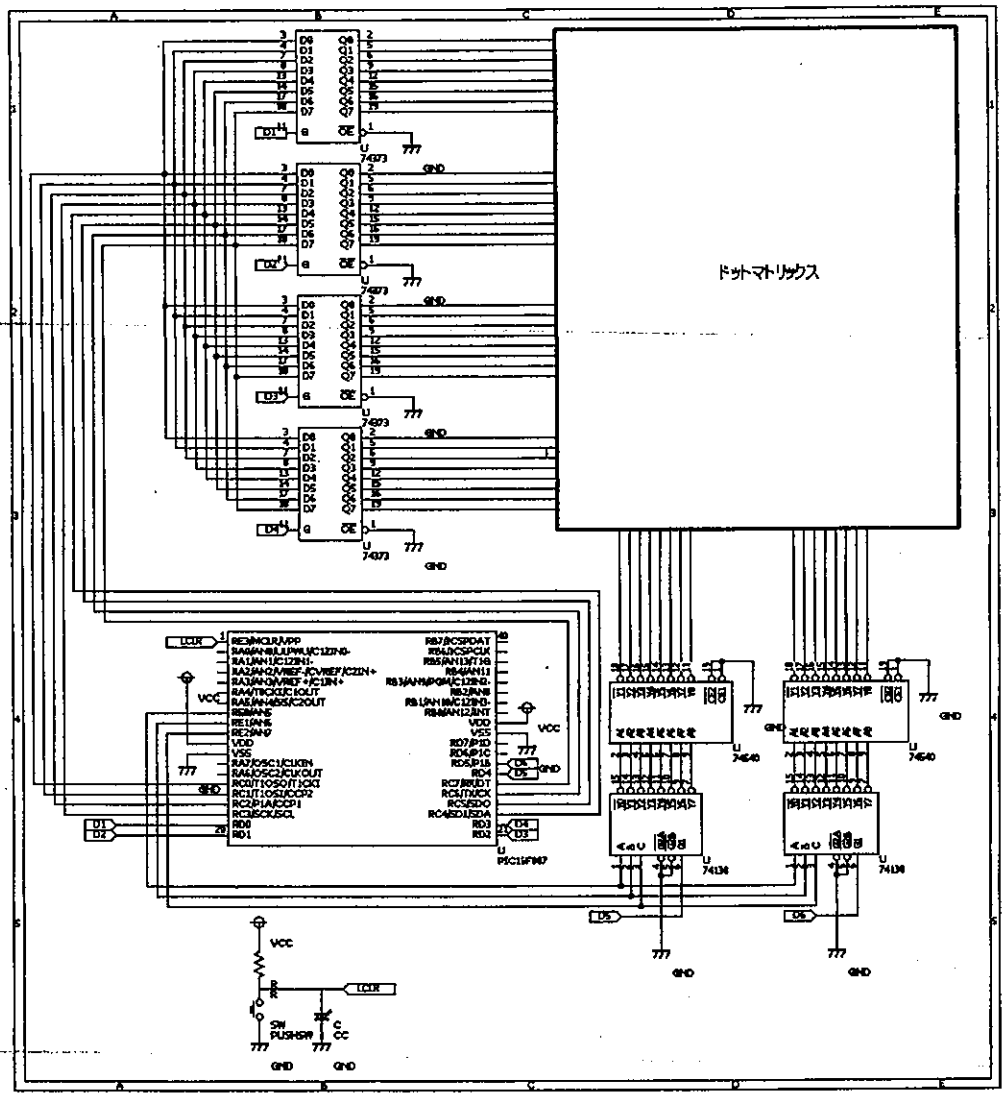
～ドットマトリックス～

・ドットマトリックスとは

このゲームの画面にはドットマトリックスが使われています。ドットマトリックスというのは、16x16の256個からなる LED の集合体で、これを使う人は、皆親しみを込めて「ドトマト」と呼んでいます。さて、このドトマトの表示方法 には、「ダイナミック点灯」を使っています。別に派手な転び方とかではありません。ダイナミック点灯というの は縦列と横列に光らせる信号を出し、その交わった点を光らせ、すぐにその点を消し、また別の点を光らせると いうことを高速で繰り返すことで複数の場所を同時に光 らせるという方法です。例えば、図の点 A,B を同時に光らせたい時には、まず、a列と1列に信号を出し、点Aを光らせ ます。その後、すぐに点Aを消し、16列とP列に信号を出 し点Bを光らせます。また、すぐに点Bを消し、点Aを光らせるということを高速で繰り返すことで点 A,B が同時に光っているように見えます。プログラムで、このようなことを何百回もやることによってこのゲームを表示させています。



・回路図

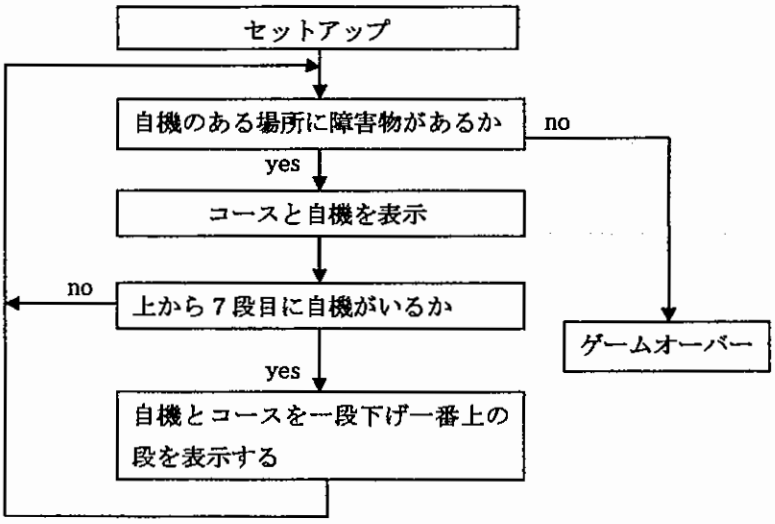


ドトマト表示部分の回路図です。74LS373を使って縦制御を、74LS138を使って横制御をしています。縦制御はポートCから信号を出しそれを各373のGピンの入力によってどう出力するか決めるという仕組みになっています。横制御も同じようにポートEからの信号をG1ピンによってどう出力するか決めています。また、373は4個のうち上2個が赤色LED用で、下2個が緑色LED用です。

・プログラム

このゲームは、コースが勝手に流れてくるのではなく、自機が動くと共にコースも流れるというプログラムになっています。プログラムは長いので、フローチャートを書きます。

フローチャート



・感想

伊藤

結局この3年間ずっとドットマトリックスに縁がある製作物でしたね。全くと言っていいほどロボットには縁がありませんでした。中2の終わりぐらいにはロボット一回ぐらいやってみたいと思っていたのになあ。

今年はジャイロをやってみたんですがこれはプログラムが難しいですね。最初はアセンブラでやっていてあまりのわからなさに愕然としていました。C言語でやったからわかったというわけでもないですが。

最近ゲーム系は似たようなものが出ていて地味になってきている気がしますね。まあゲーム系は変えるところが表示方法、操作方法、ゲーム内容ぐらいしかないから仕方ないのかもしれませんが、今後新感覚のゲームを作る発想がでてくるといいなあと思います。

山本

今回初めてプログラムを書いてプログラムの楽しさがわかりました。ドトマト制御はいまでこそ理解でき、説明できるくらいにはなりましたが、最初は伊藤さんの言うことが何も分からなくて大分迷惑をかけてしまいました。制作スピード自体は早かった方だと思うので、来年もこのぐらい余裕を持ってできればいいなと思います。

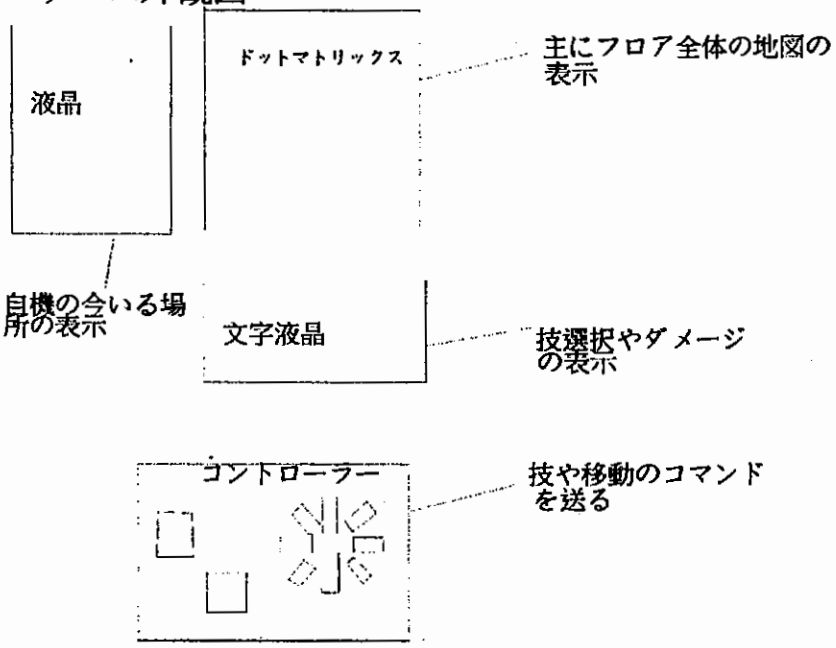
復活の朝

製作者 H1 石原 皓
M3 森 一斗
協力者 物無の皆様

＝ゲーム説明＝

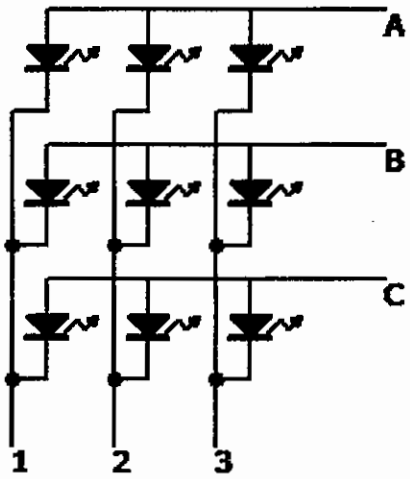
このゲームは、寄ってくる敵を倒す or 避けて階段を目指すというゲームです。また自分は八方位に動くことができます。また1ターンに1回までしか移動はできません。
そして、文字液晶に表示される4つの技でそれぞれ攻撃範囲が決まっており敵を撃破していくという感じです。勿論技には使用制限があって使える回数は決まってます。
ダンジョンの作りは階段を降りるごとに変わり、最下層のボスを倒すとゲームクリアとなります。ボスはかなり強めの予定なので頑張ってください。
どれが何を表しているか説明すると、
Ⅰドットマトリックスが全体の地図を表しています。液晶の拡小版みたいなものです。
Ⅱ液晶は自機(自分)の周りを表示させています。基本的には1つの部屋を表しています。
Ⅲ文字液晶は敵からのダメージや技の選択を表示させています

＝ゲーム外観図＝



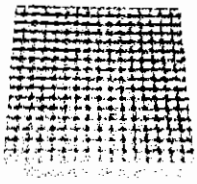
＝ダイナミック点灯の仕組み＝

まず、ドットマトリックスとは横の列に対応したピンが16、縦の列が32ピン（緑の LED で16ピン、赤の LED で16ピン）あり、1列ごとに高速で光らせることにより（人間の目に見えないくらい）16×16の LED を光らせています。このようにドットマトリックスはピン数が多いのでそこでダイナミック点灯を使います。ダイナミック点灯とは、ヒトの目では認識できないくらいの速度で部分的な表示を連続で繰り返して常時表示されているかのように見せかける表示です。

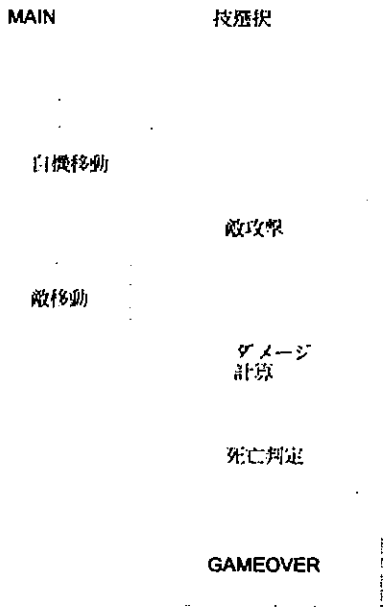


なお内部はこんな感じです

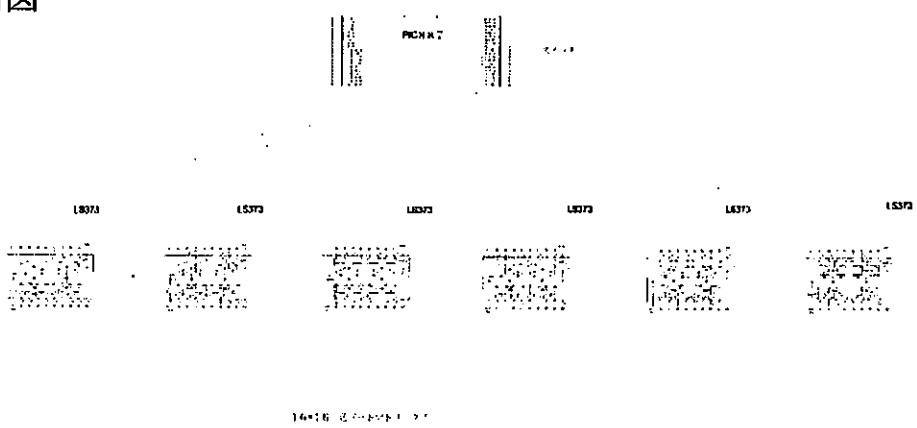
LS138 の場合、利点は「何行目か」のデータさえあればいいのでデータ量が少なくてすみます。しかし、不便な点として縦スクロールと横スクロールの関数を別々にしなければなりません。
また、LS373 の場合は行数のデータが増えてしまい使いづらくなってしまいます。しかし、LS138 とは逆に縦スクロールと横スクロールの関数を 1 つにできるという便利な点もあります。つまり、LS138 の方が便利というわけです。



フローチャート



回路図



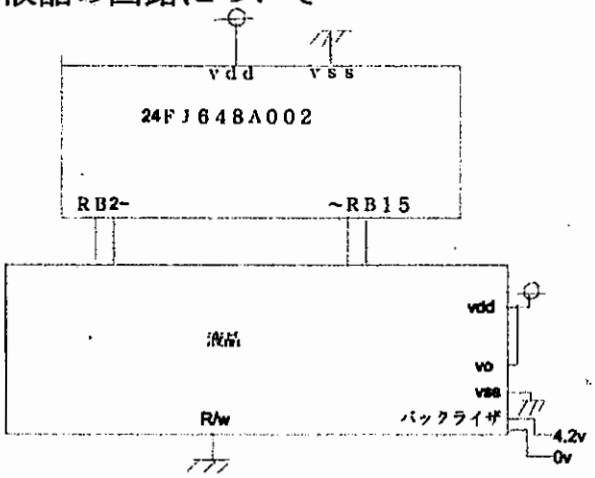
回路図について
この回路はPICから出た出力をLS373を入れ、ピン数を増やしてドットマトリックスにいれ入れています。液晶との接続はまだなので省略します。
今思うとLS373よりLS138の方を使うべきだったと思います。そのほうが接触不良も減りもっと快適になったのだと思っています。

＝感想＝

初めてPICやプログラムを書いたりしてドトマト部分を担当したわけですが、全くと言っていい程初めは何も分からず石原さんに聞いてばかりでその都度教えてもらってました。
それで、PICも1ヶ月くらい全く動かず石原さんといろいろ考えた末に動いた時は感動しましたしプログラムはおもしろいと思いました。来年はもっと知識をつけて製作していきたいと思います。

ここからは石原がお送りいたします。

＝液晶の回路について＝



詳しくは森田のを参照してください。

＝フローチャート＝

詳しくは森田のを参照してください。

＝制御について＝

この液晶は、縦64×横128ドットを横に2分割し縦8×横1の8ビットずつのデータを出力していくことで制御が可能である。
さまざまな列をずらすなどのコードは使わなくてもよい模様。

（筆者はこの液晶を実際に動作させたことがほぼ無いに等しくこれは決して信憑性のあるものではありません。）

感想

初めての共同製作で意思疎通が不足していた。もっと技術継承のため森に自力でやらせるべきだった。

Flash　a maze

H1 石井　裕太

協力　物無の皆様

【概要】

flash a maze とは、文字通りフラッシュ迷路である。具体的には、ドットマトリックスが迷路を表示し、少し経つと消える。その消えた跡を、迷路の壁に当たらないように進んでいく、というものである。

【ドットマトリックスとは】

たぶん他の人が語ってくれるでしょう。省略。

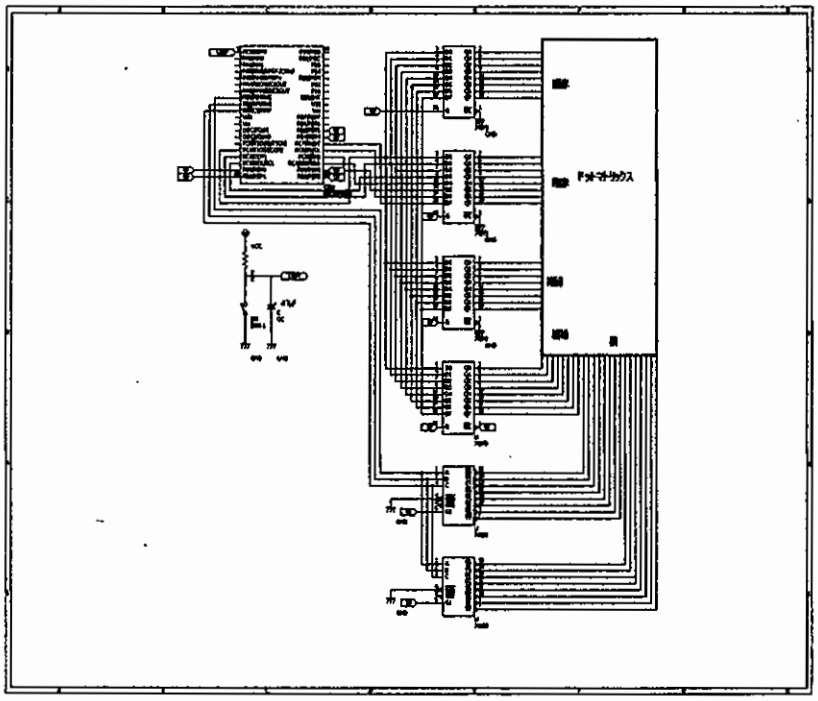
【ストーリー】

テロリストがとある研究所にたてこもっているという情報を受け、M 国の政府は急遽特別選抜部隊を設け、調査に向かわせる。そこで彼らが見たものは、テロリストではなかった。そこには大量の爪あとと、全滅したテロリストだった。部隊員たちは直ちに政府に連絡しようとしたが、通信が何者かに妨害され、救援を呼ぶことができない。彼らは連絡が可能な位置へと退避しようとしたその時、おぞましい奇声とともに、ゾンビの大群が現れた。ゾンビに襲われた隊員たちは、どんどん数が減っていき、ついに一人となってしまった。彼女の名前はチェ・ジュン。彼女は逃げるために入り組んだ道を通って、脱出をはかろうとする。通らなければならない道は暗く、ライトが無ければ何も見えない。彼女は長い道のりであるということをよくわかっていた。そのためライトを節約し、極力点点けずに節約しようとした。幸い、ライトを照らせば入り組んだ道のある程度はわかる。しかし、ゾンビに追われている今、最も効率よく、一度もぶつからずに進まなければならない。またゾンビが立ちはだかってきて、とっさに避けなければならないかもしれない。彼女の孤独な、生きるための戦いが始まる・・・

【感想】

やる気が著しく低かったために、効率も悪く、無駄なじかんを多く過ごしてしまいました。あとは技術不足ですね。もっと勉強しなければいけないようです。

【回路図】



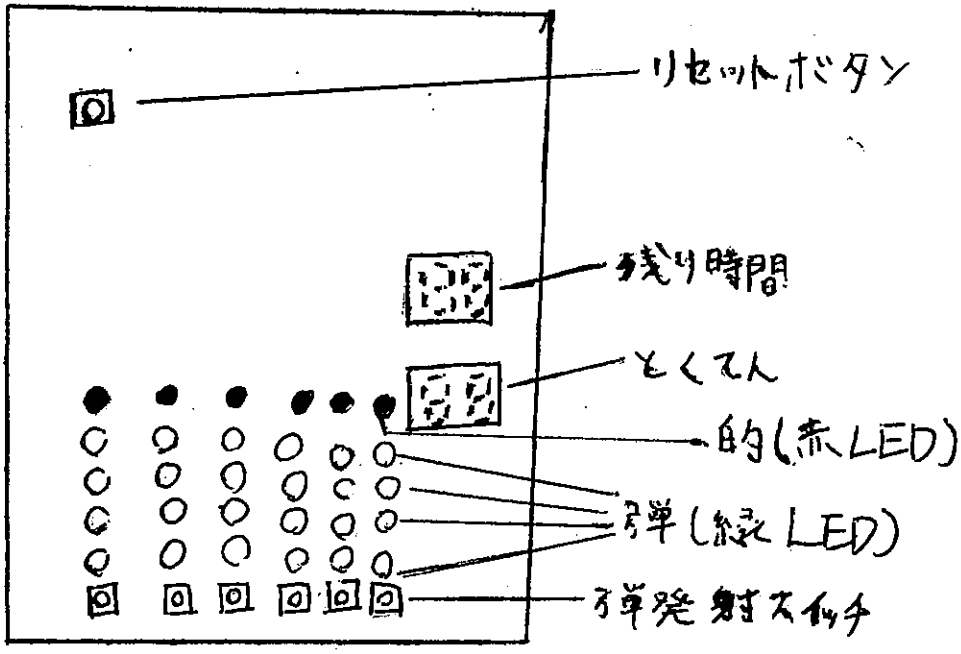
タイトル　もう度ったってええじゃないか!

製作者　小川佑太　中2　設計者　川田希望　高1

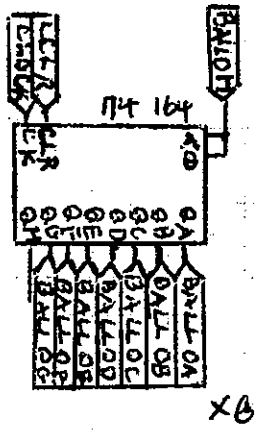
ジャンル　シューティング

ルール　制限時間は20秒
赤い的に当たると1点、
はずすと　−1点です。
長押ししても最初の1点
しか入らないので連打して下さい。

外見

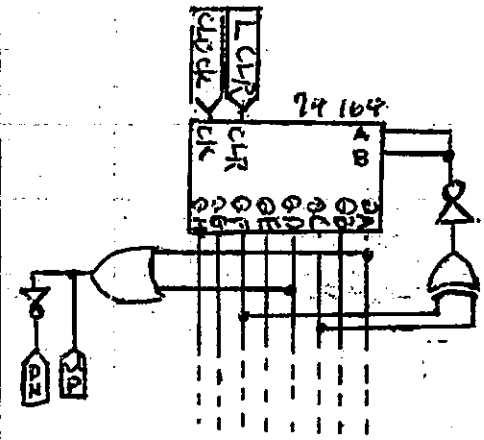


感想…ハンダづけがうまくいかなかったり、ICを何度も変えたので
とても大変でした。

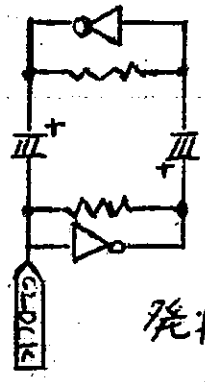


×8

弾発射スイッチ

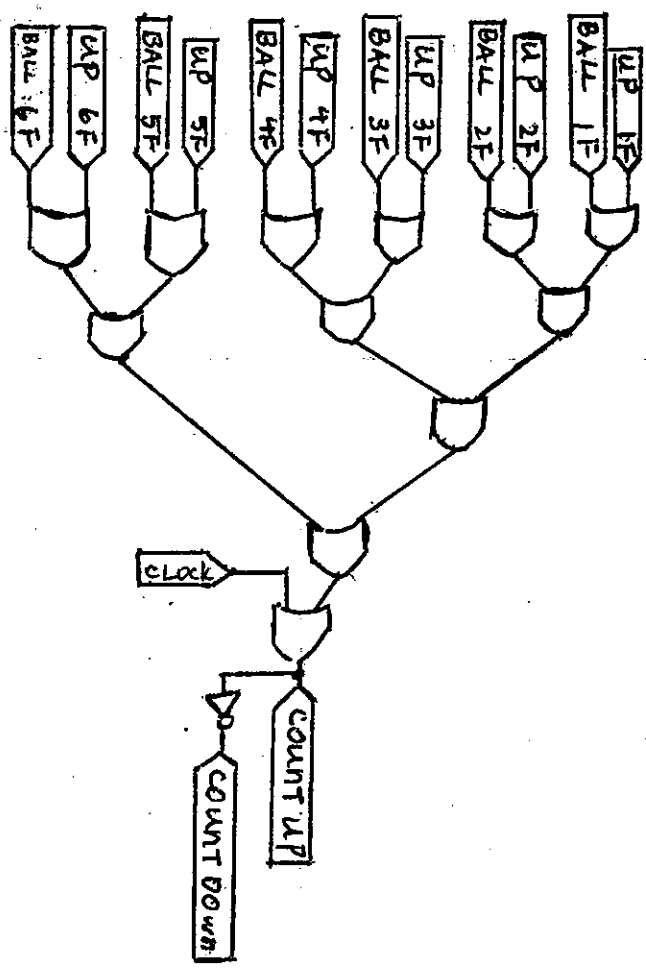


乱数+的判定 ×6

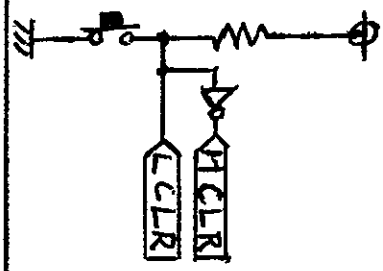


発振

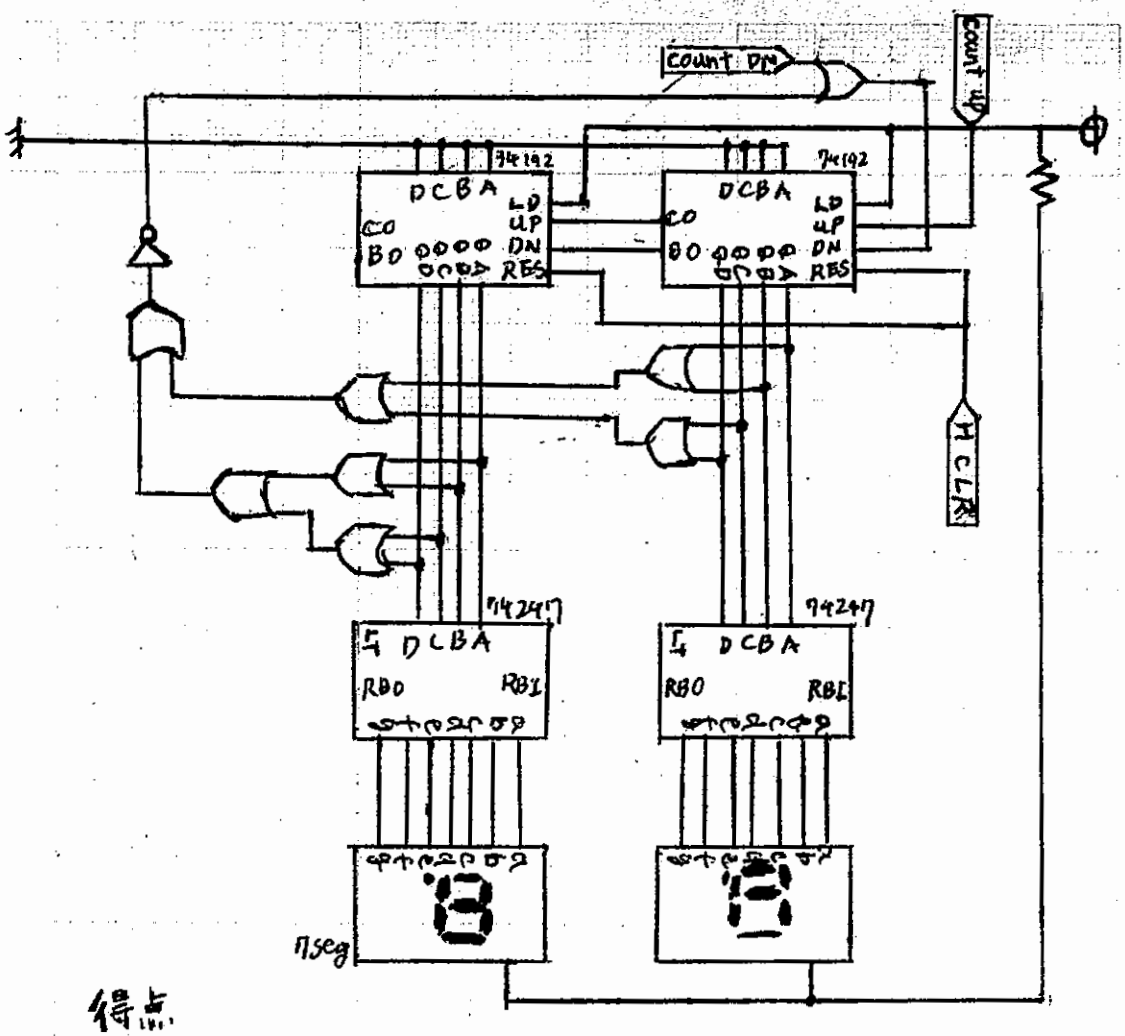
×2



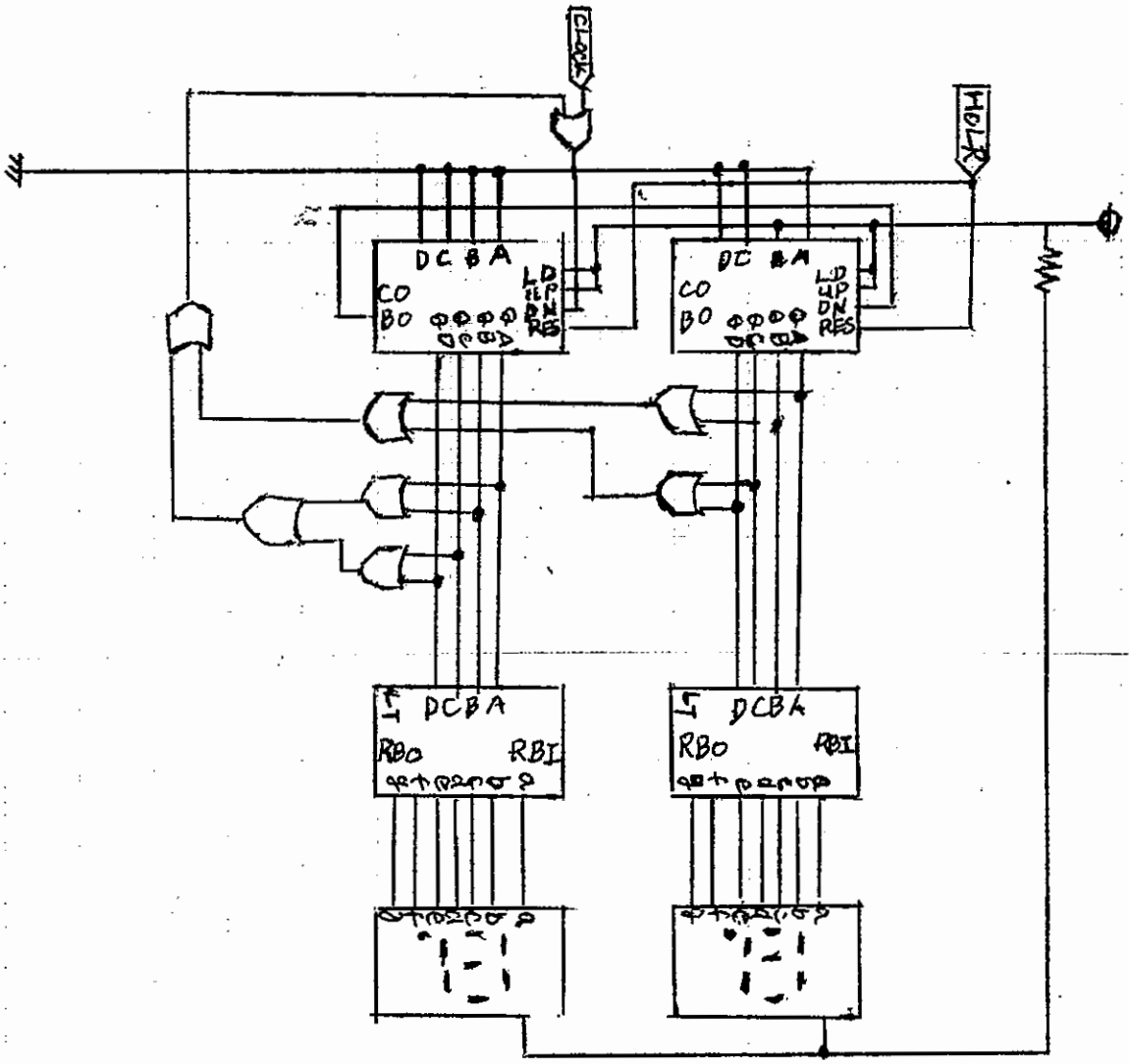
カウント UP、DN



クリアスイッチ


count DN
count up
74192
DCBA
LD
UP
DN
RES
CO
BO
74192
DCBA
LD
UP
DN
RES
CO
BO
HCLR
74247
DCBA
RBO
RBI
74247
DCBA
RBO
RBI
7seg

得点

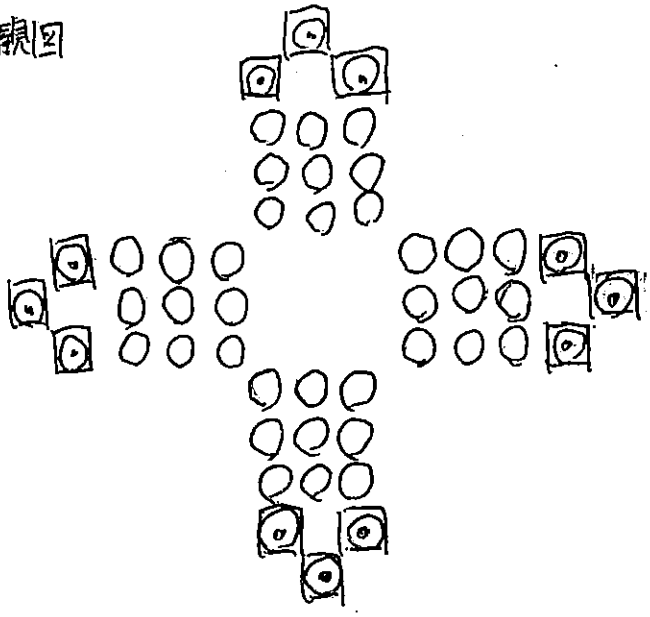


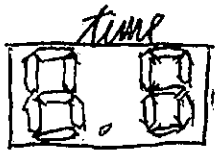
20秒タイマー

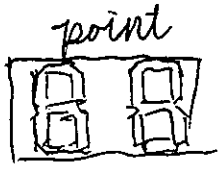
アクマはお好きですか

製作者 中2-6 木村
回路図設計者 H1-6 大森さん
協力者 物部のみなさん

1 概観図





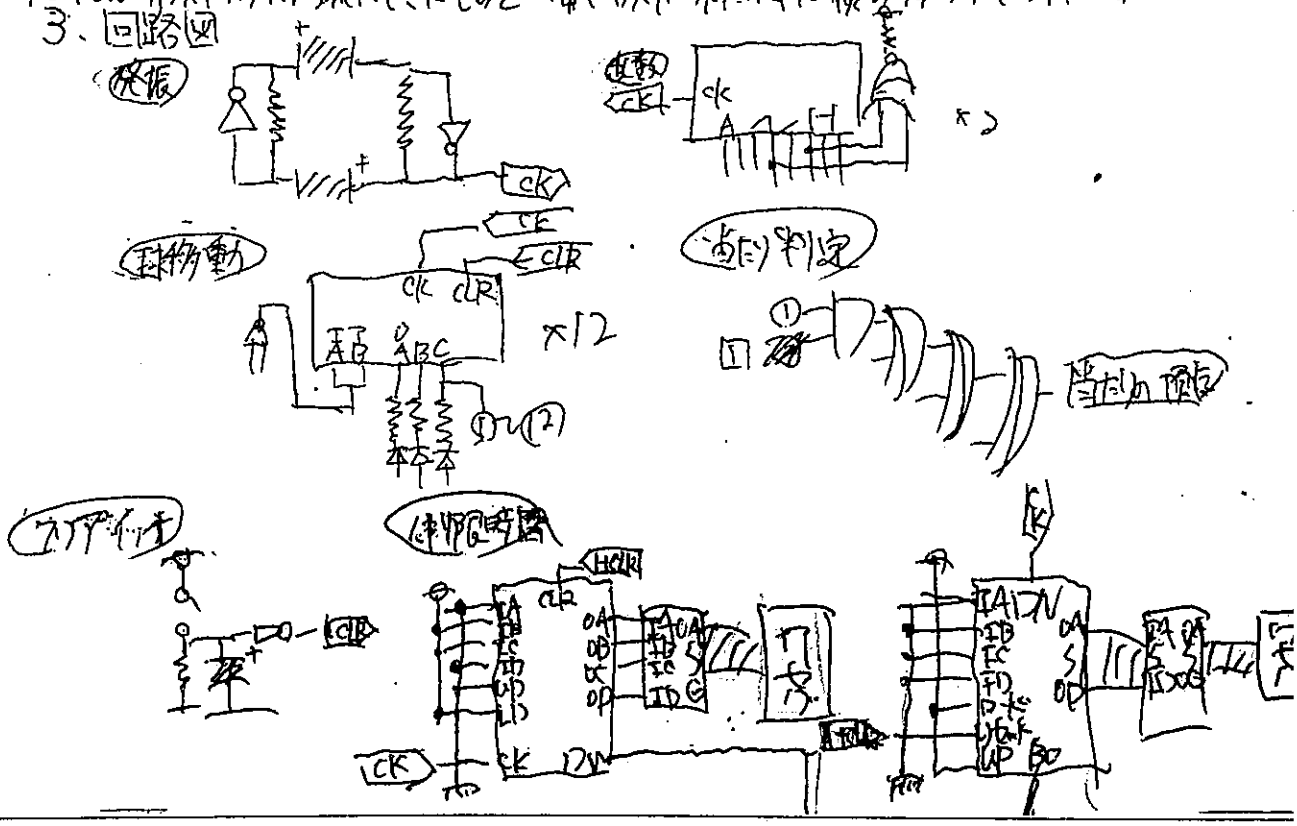


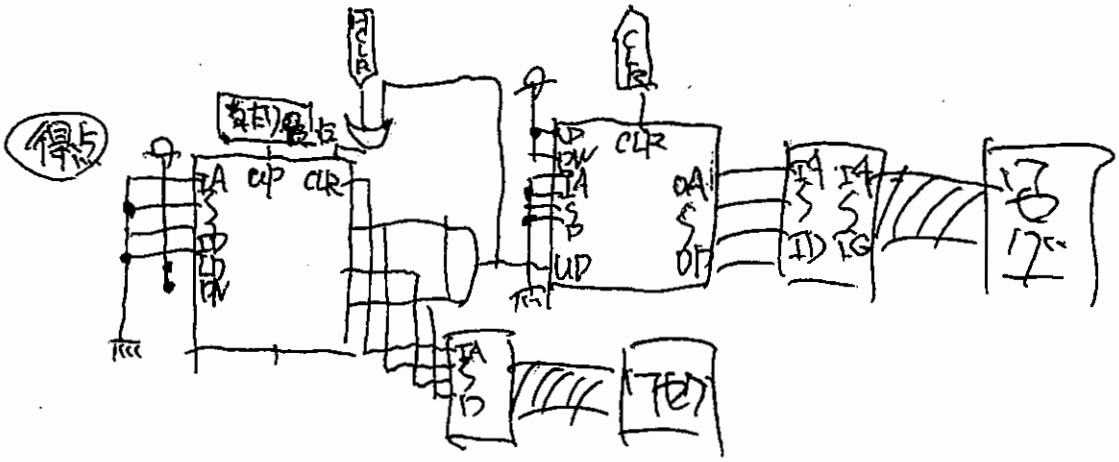
2 ゲームのストーリー

世は天使と悪魔の戦いの真っ只中。あなたは天使となっておそいくる悪魔をたおしてください。制限時間内にどれだけの悪魔を倒せるでしょうか

ルールは外側の列から流れてきたものを一番内側に来た時に横のスイッチで消してください

3、回路図





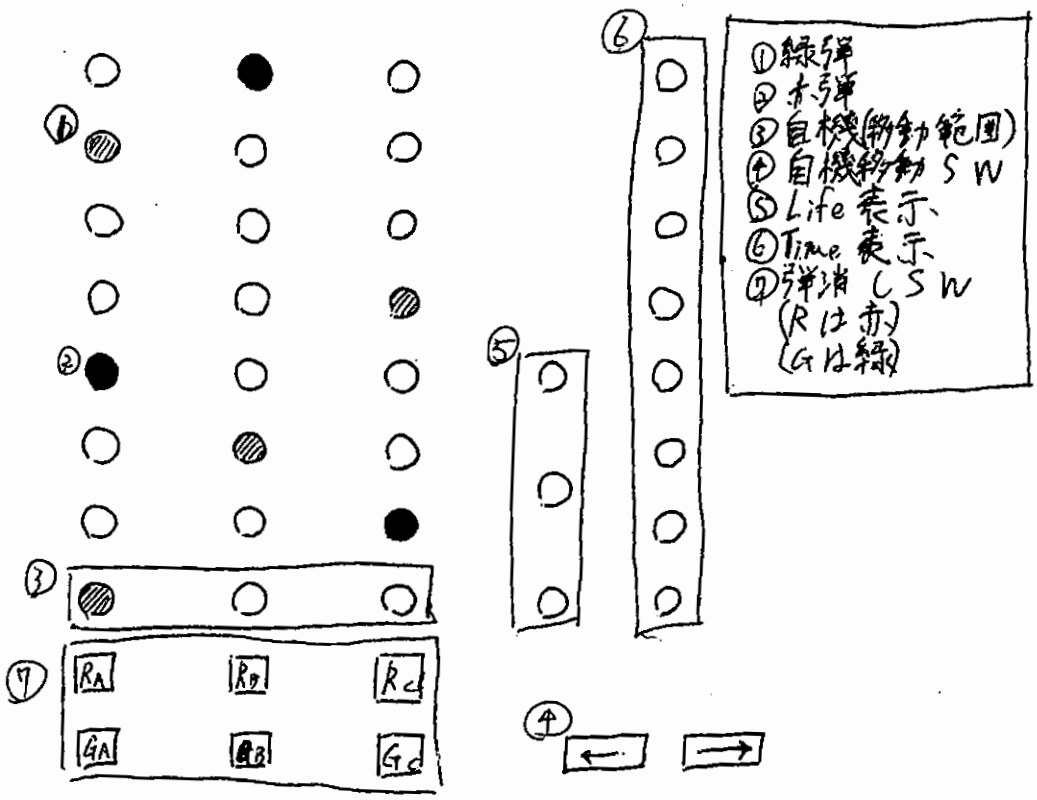
4、感想

製作はとても大変でむずかしかったです。
でも制作員としてとても楽しかったので、
来年もがんばりたいです。

パラシュートキャッチ

制作：M2 黒田
回路図：H1 東本さん
協力：物無の皆さん

1：外観図



2：ルール

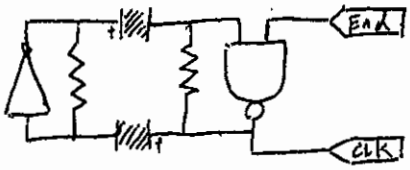
上から赤・緑の弾が流れてくるのでそこに自機を移動させて対応するSWを押して消していくゲームです。取り逃がし、押し忘れ、SW全押しをするとLifeが減ります。Timeが全て消えるまでにLifeが残っていればゲームクリアとなります。

3：ストーリー

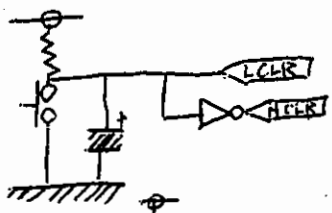
ある日A君は上からパラシュートで落ちてくる人を受け止める仕事に就いた。3回ミスするとクビになるため、A君はミスすることなく仕事を終えることができるのだろうか……

4:回路図 (一部省略あり)

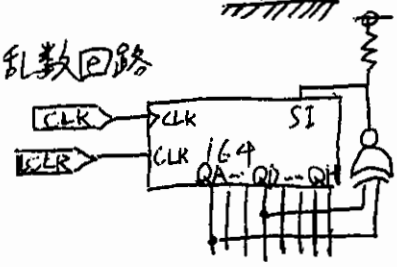
発振回路



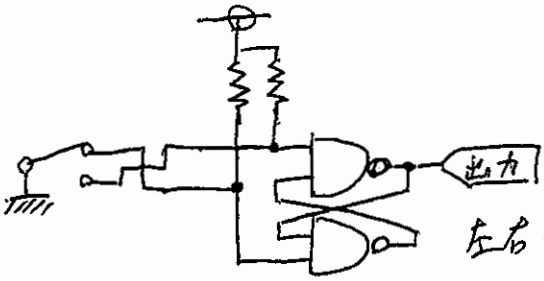
クリア回路



乱数回路

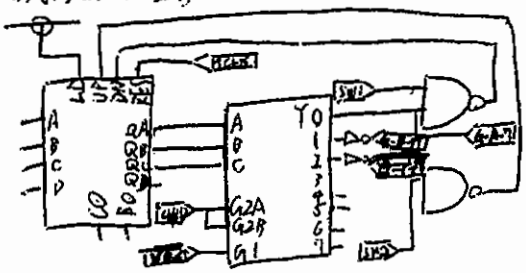


SW回路

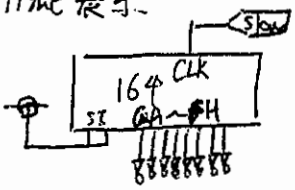


左右で2つ

自機移動回路

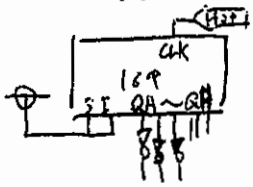


time表示



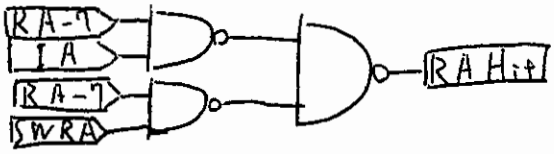
T0~T7

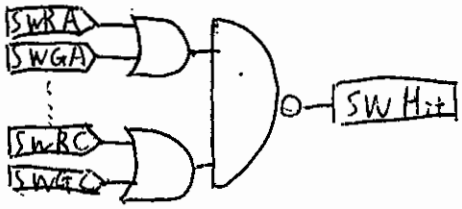
Life表示

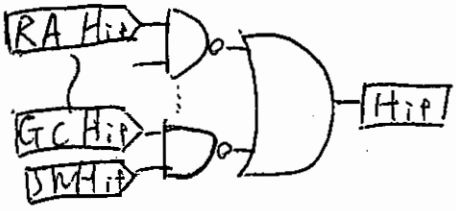


L0~L2

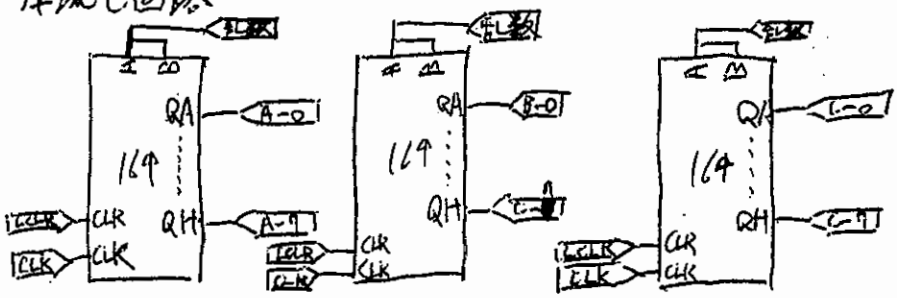
Hit







弾流し回路



RとGで1つずつ

DEAD

LED・SW配置

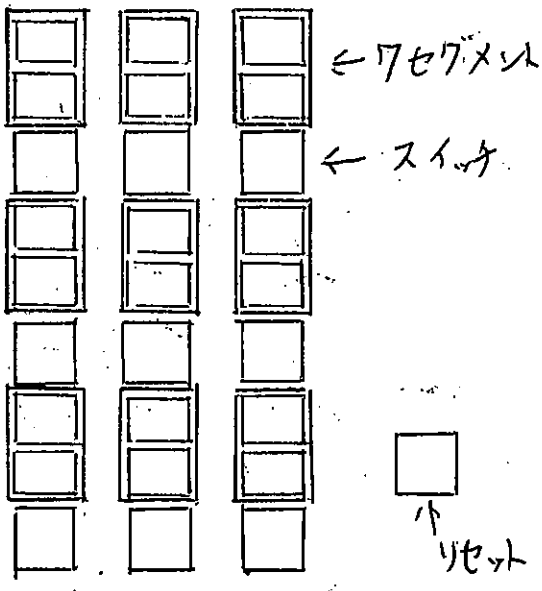
5.感想

ゲームというものを身近にあったが、一からゲームを作ったことで市販のゲームはどうやって作っているのかの興味も出てきた。
次の制作物ではもう少しがんばりたいと思う。

8×8

回路計計　高1　森田　晃平さん
　　高1　石井　　さん
協力　物無の皆さん
制作　中2　佐藤　惇

1 外観



2 ルール

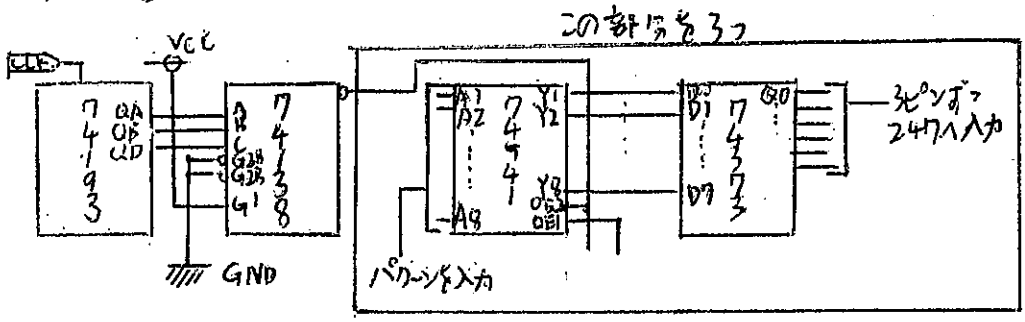
1 周りの7セグに数字を表示する。

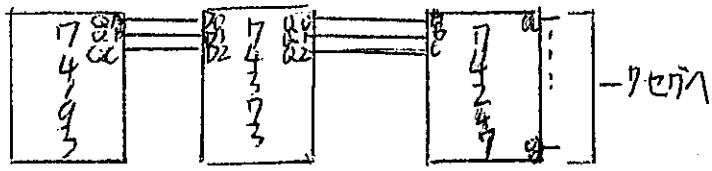
2 それが消える。

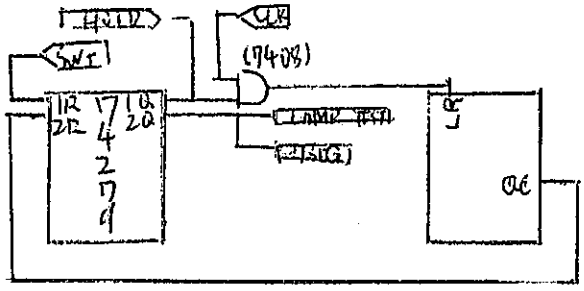
3 まん中にお題が表示される

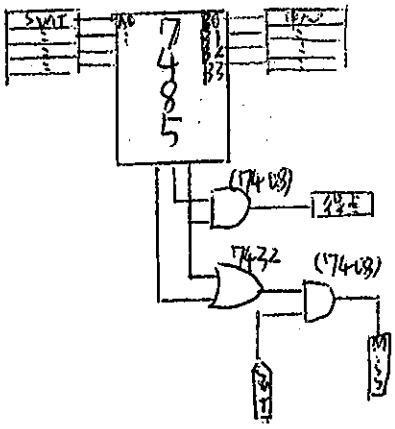
4 その数が出ていた場所のボタンを押す。くり返し。

三. 回路図









※ 実際の回路とは異なります。

74…と書いてあるのは74LSシリーズICとなります。
(373のみHCシリーズ)

4 説明、感想

① かんたんに説明

a 外側のワセグはパターンが決まっています。
その中から1つずつ出てきますので、
パターンを覚えることも可能です。
b お題はタイミングによって決まります。
1回目のお題は固定になります。
(電源投入時の値となるため。)
(ボタンが押された時、そこに表示されていた数と
お題を照合します。
同じなら正解、ちがえば不正解と判定します。

② 感想

本感的なゲームを目指して制作しました。そのコンセプト
は達成できたかと思います。
初めての制作となりましたが、先輩のサポートもあり、
楽しく制作できました。
配線なども異様に多く、あまり効率的な物とは言い難いこと
になっていますが、それは来年に生かしたいと思います。

あとがきのあも読んでいただければ幸いです。

ここまで読んでいただき、ありがとうございました。

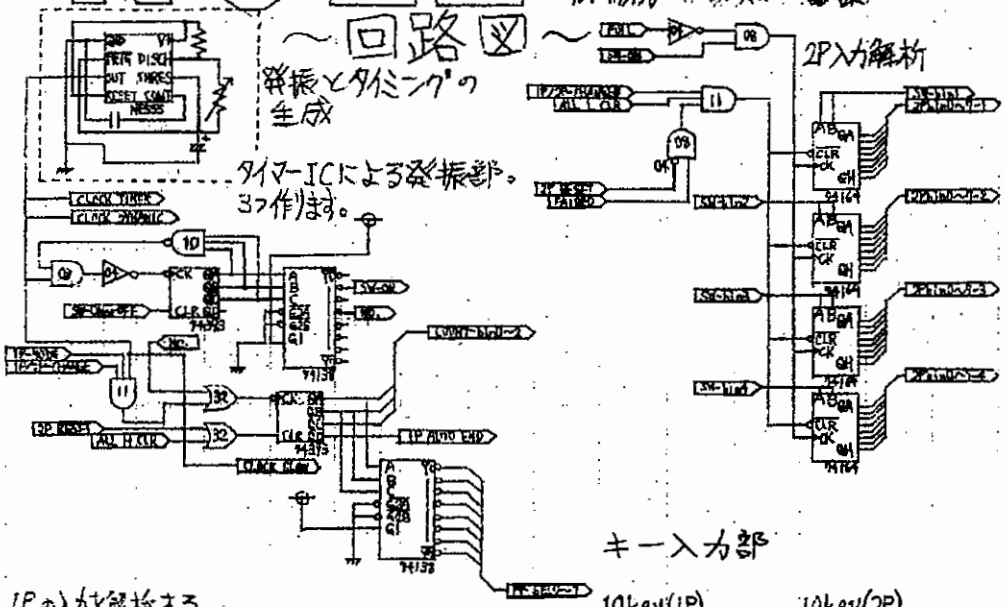
字が汚くて、すみませんでした(汗)

2011/3/31

NULL

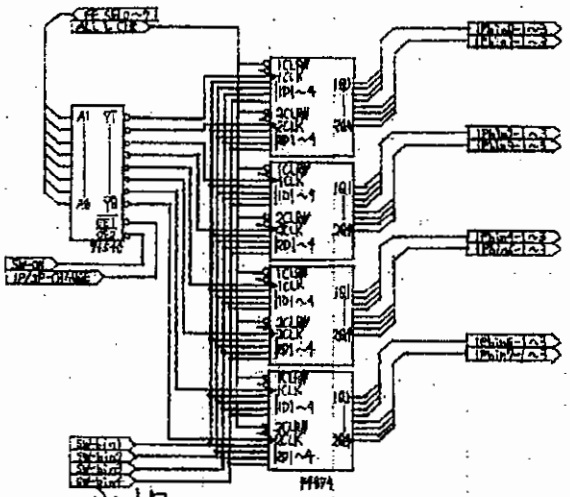
製作者　M2池田
回路図　H1梅津さん
御協力　物無の皆様

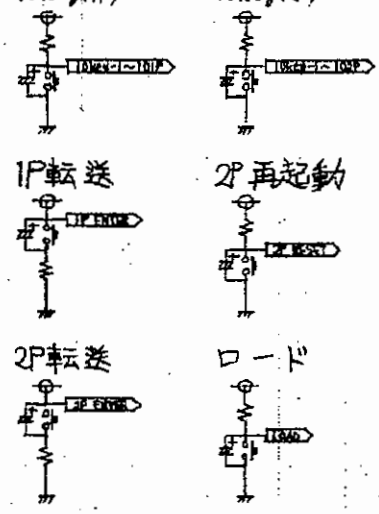
～回路図～



キー入力部

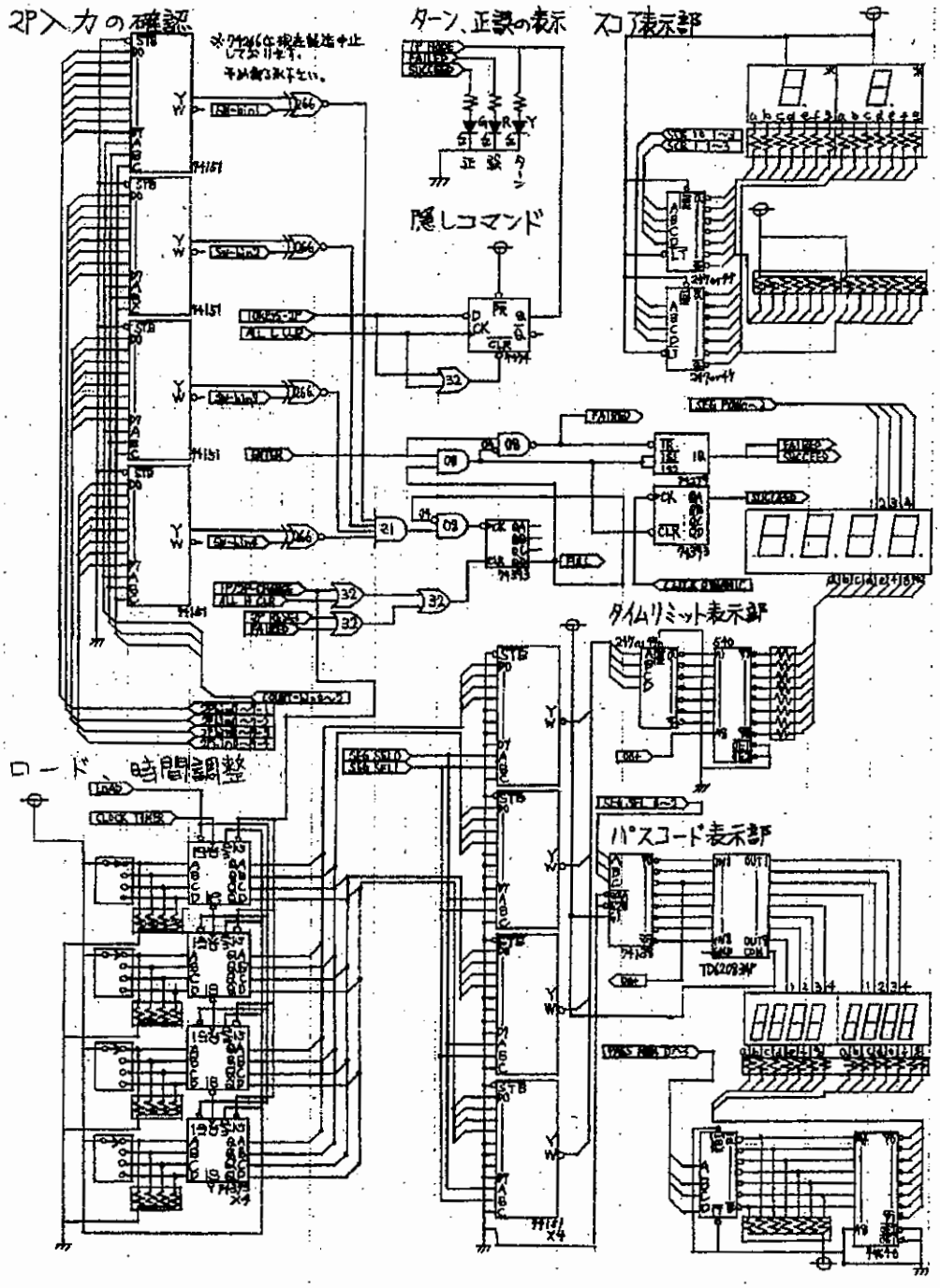
1Pの入力を解析する。

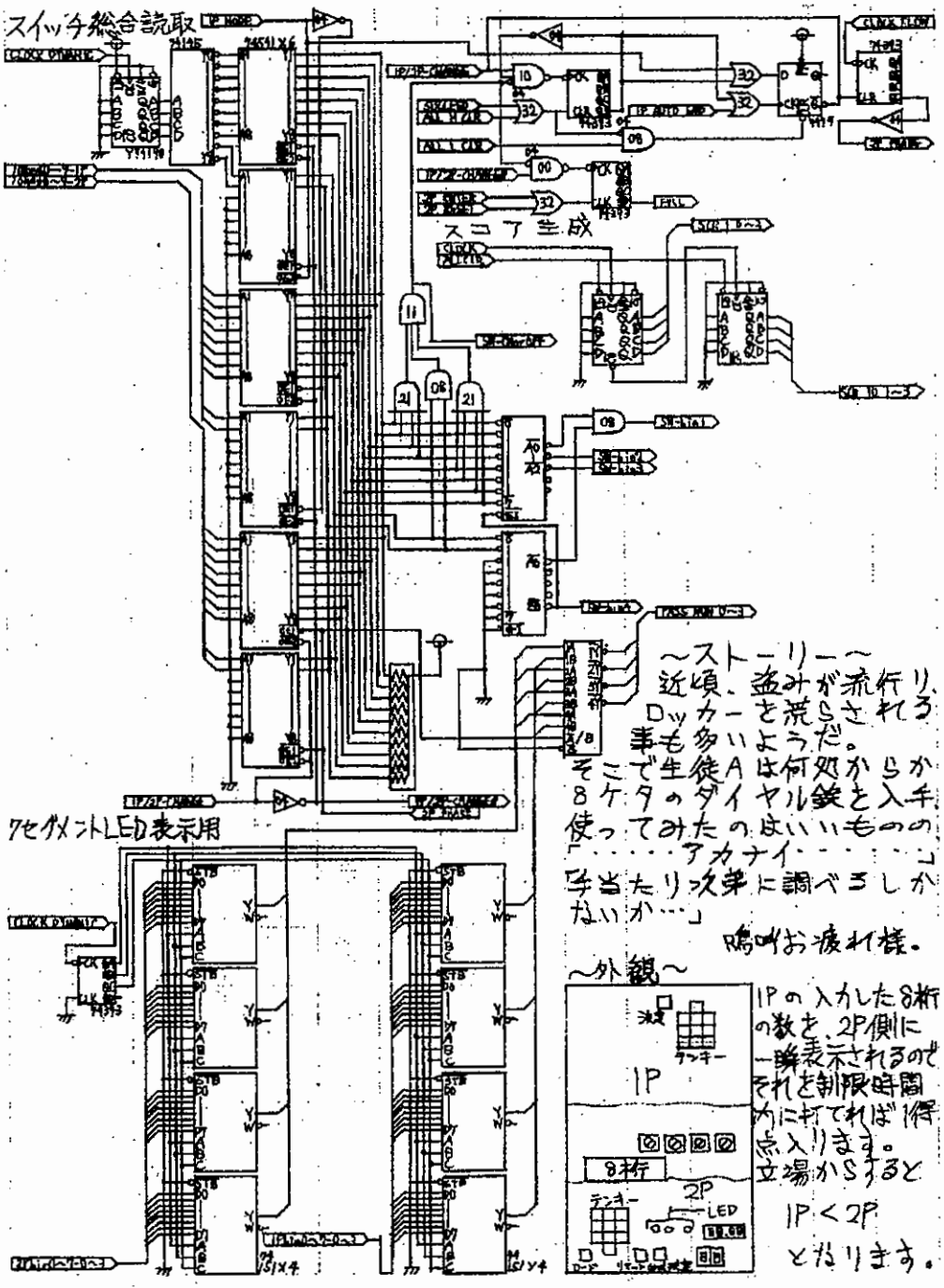




～感想～

このようなバグの多い、まだ未動作のゲームですが、文化祭で展示できるものに向け、より力を入れていきたいです。


2P入力の確認
ターン、正誤の表示
スコア表示部
隠しコマンド
タイムリミット表示部
ロード、時間調整
パスコード表示部

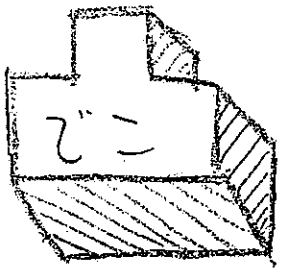


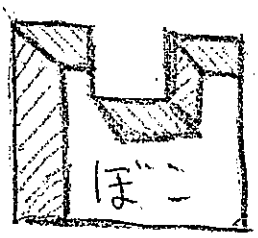
～ストーリー～
近頃、盗みが流行り、Dッカーを荒らされる事も多いようだ。
そこで生徒Aは何処からか8ケタのダイヤル錠を入手
使ってみたのはいいものの
「……アカナイ……」
「手当たり次第に調べるしかないか…」

鳴呼お疲れ様。

～外観～

1Pの入力した8桁の数を、2P側に一瞬表示されるのでそれを制限時間内に打てれば1得点入ります。
立場がSすると
1P＜2P
となります。





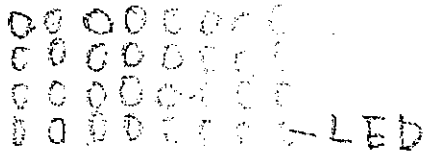
製作者
第2 森永 健

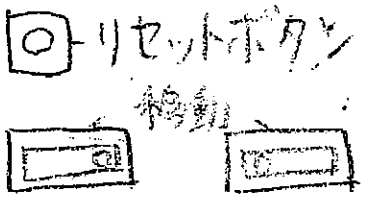
回路図設計者
高1 武子 真行 氏

協力
物無の皆様

壁あて

(外観)

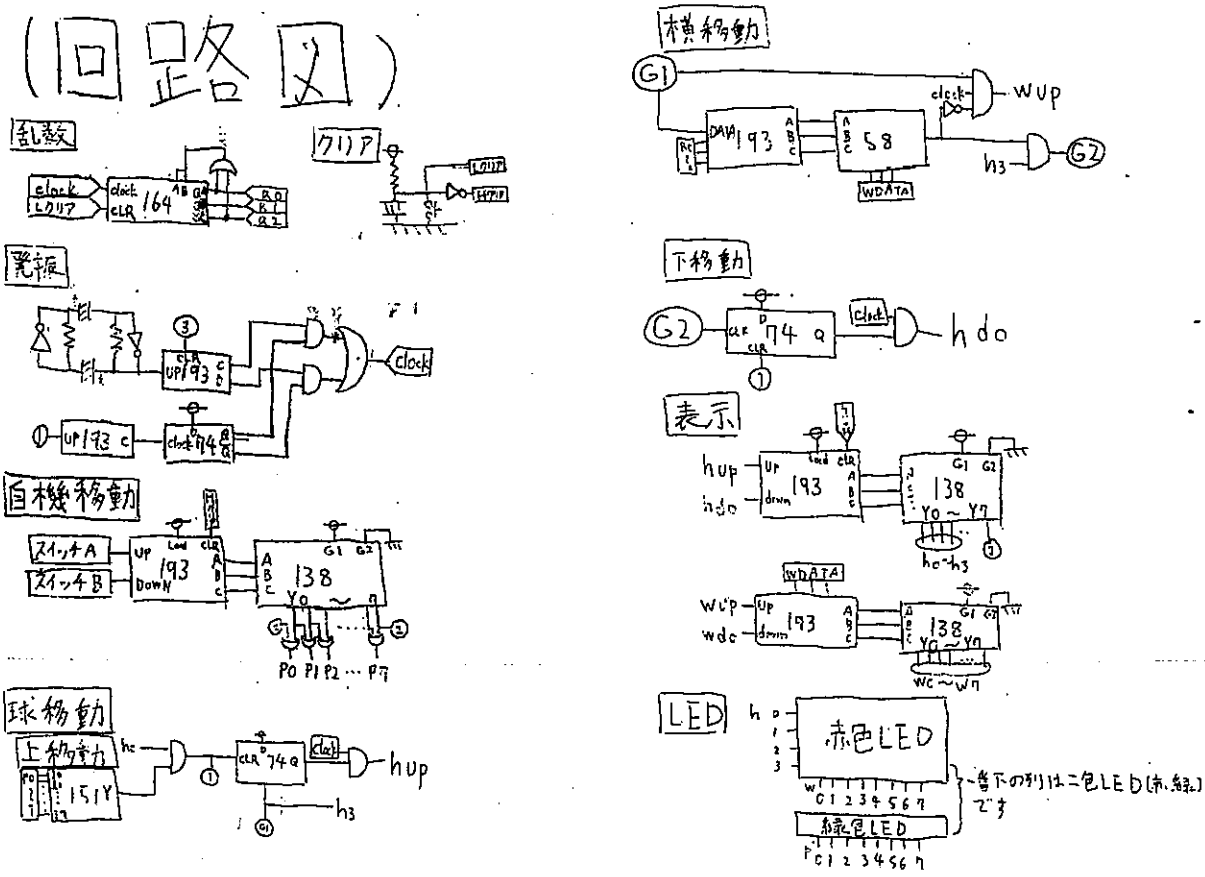




(ルール)

LEDは横8列、縦4列にならんでいて一番下の段を自機が移動して球を打ち返し壁あてをするゲームです。
球がまっすぐに戻ってくるとゲームとして成立しないので、壁に凸凹があると設定して壁にあたった球は斜めに進んでから戻るようにしました。

(回路図)



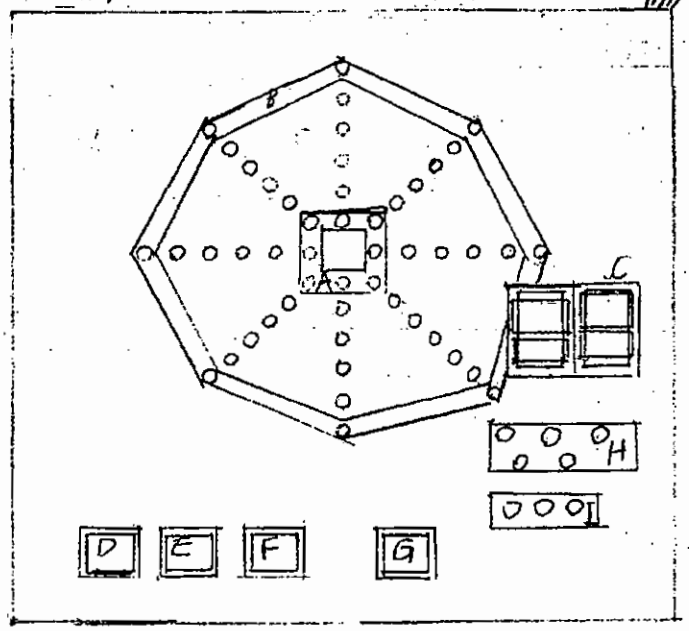
(感想)

僕は電子工作は初めてでした。
慣れていないので、基板は裏を見ると、ハンダゴテの熱でこげて(ひどいところでは溶けて)しまっています。
配線材も熱しすぎて被膜が変質しているところがあります。
失敗も多かったですが、電子工作は楽しかったです。

五面楚歌

制作 M2 花園
回路図 山口
協力 物部の者

・外観



A…自機移動範囲
B…敵移動範囲
C…残弾数
D…左自機移動スイッチ
E…右自機移動スイッチ
F…
G…弾アスイッチ
H…敵数
I…自機残数

・ルール

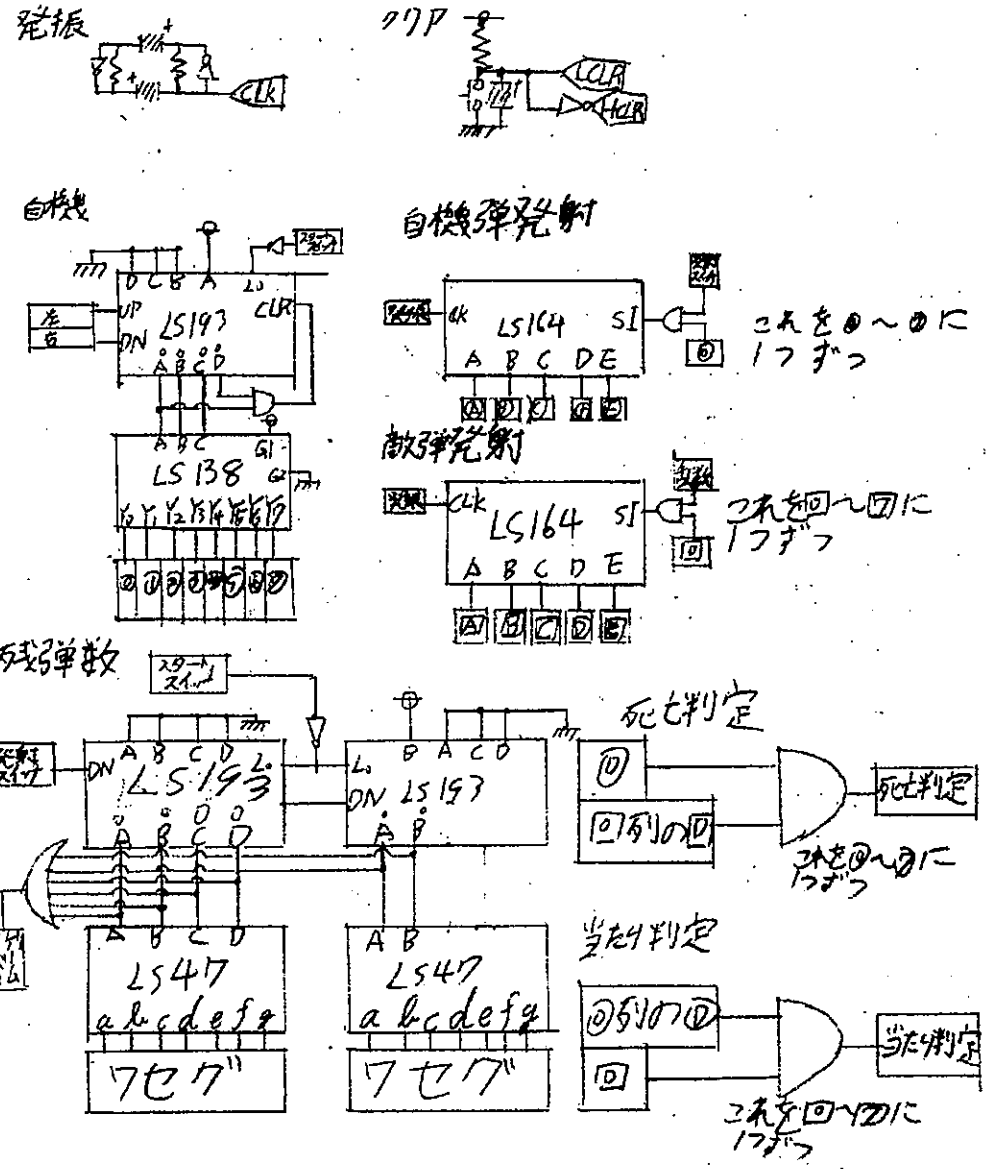
内側8マスを移動し、外側8マスを移動する5体の敵の弾をよけつつ、敵を全て倒せばゲームクリアです。
三回敵の弾に当たるか、20発弾を撃ち尽くすと、ゲームオーバーです。

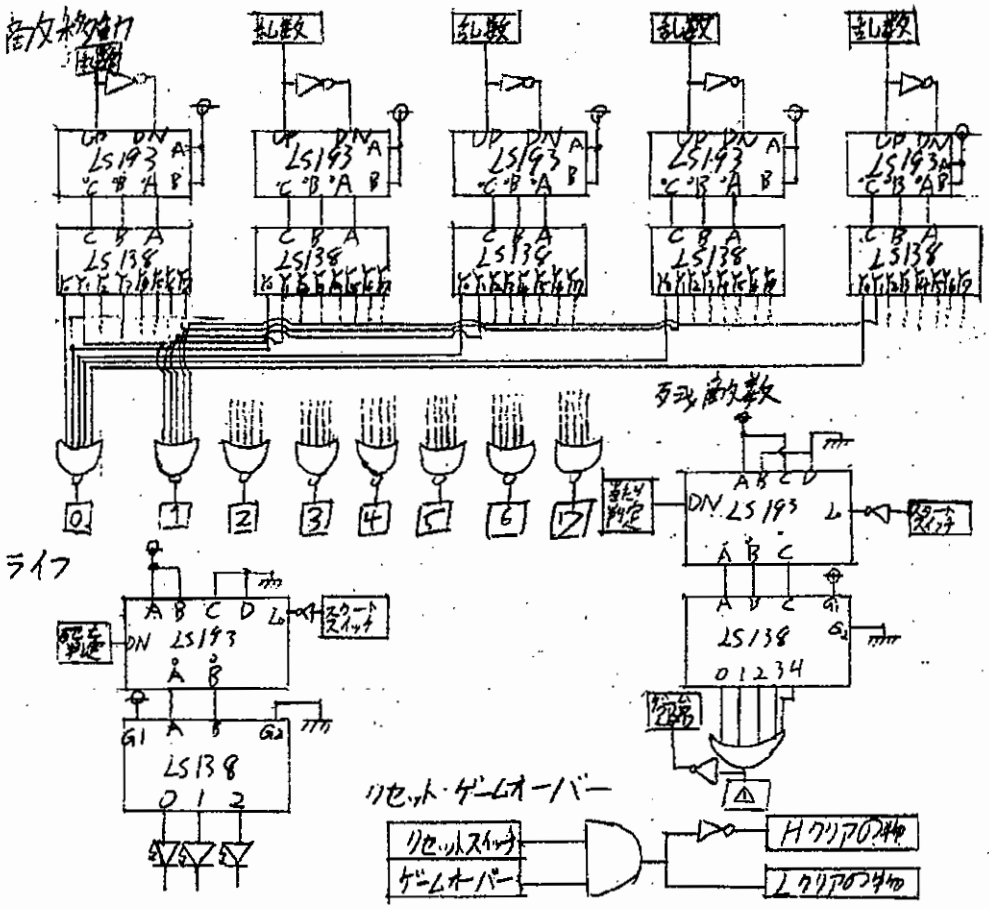
・ストーリー

―この星はもう終わりかもしれない―

五隻の宇宙空母に囲まれ、誰もが希望を捨てていた時、それは動き出した。
人類最後の戦線。それは圧倒的な力にも抵せず、20の砲弾と共に最後の防衛戦を繰り広げようとしていた。

回路


発振
CLK
クリア
CLR
自機
LS193
LS138
自機弾発射
LS164
これを0〜0に1つずつ
敵弾発射
LS164
これを回〜7に1つずつ
残弾数
LS193
LS193
LS47
LS47
7セグ
7セグ
死亡判定
死亡判定
これを0〜0に1つずつ
当たり判定
当たり判定
これを回〜7に1つずつ



感想

初めての電子工作は、バグやショートが多発して大変でした。
来年は少なくとも初歩的なバスを無くし、質の良いものを
早く作れるようになりたいです。

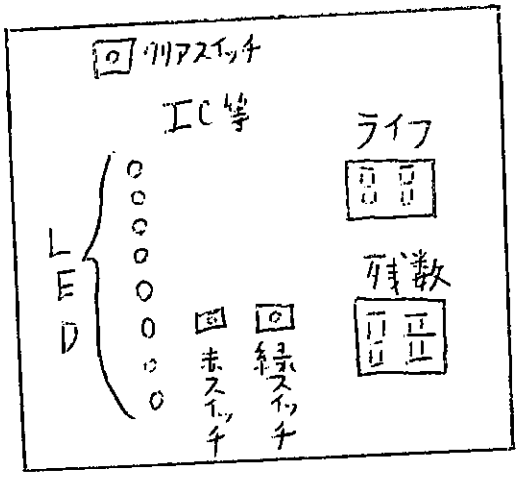
迎撃は爆発で

製作者、M2 四柳 雄太

回路図設計者、H1 橋本さん

協力、物無の皆様

1、外観



2、ストーリー

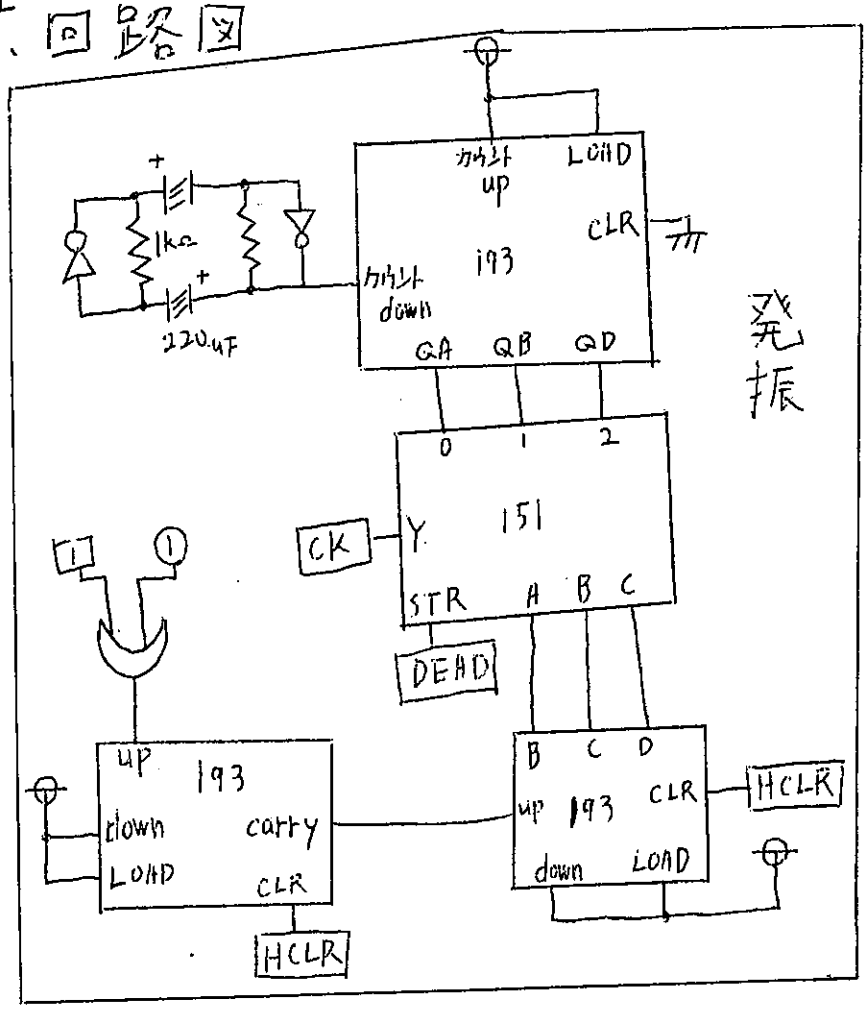
西暦22XX年

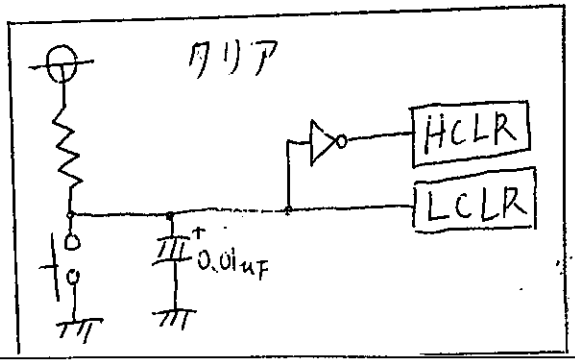
敵に追われていたあなたはロボットに乗って逃げています。
しかし、そこは一本道で行き止まり。
敵は100発近くの砲弾を持って担っています。
こうなったら迎撃しかないのです。しかしあなたは
目が悪いのでギリギリまで待つ必要があり、
早すぎると爆発で自分がやられてしまいます。
がんばって下さい。 ※あなたはLEDより上に居ます

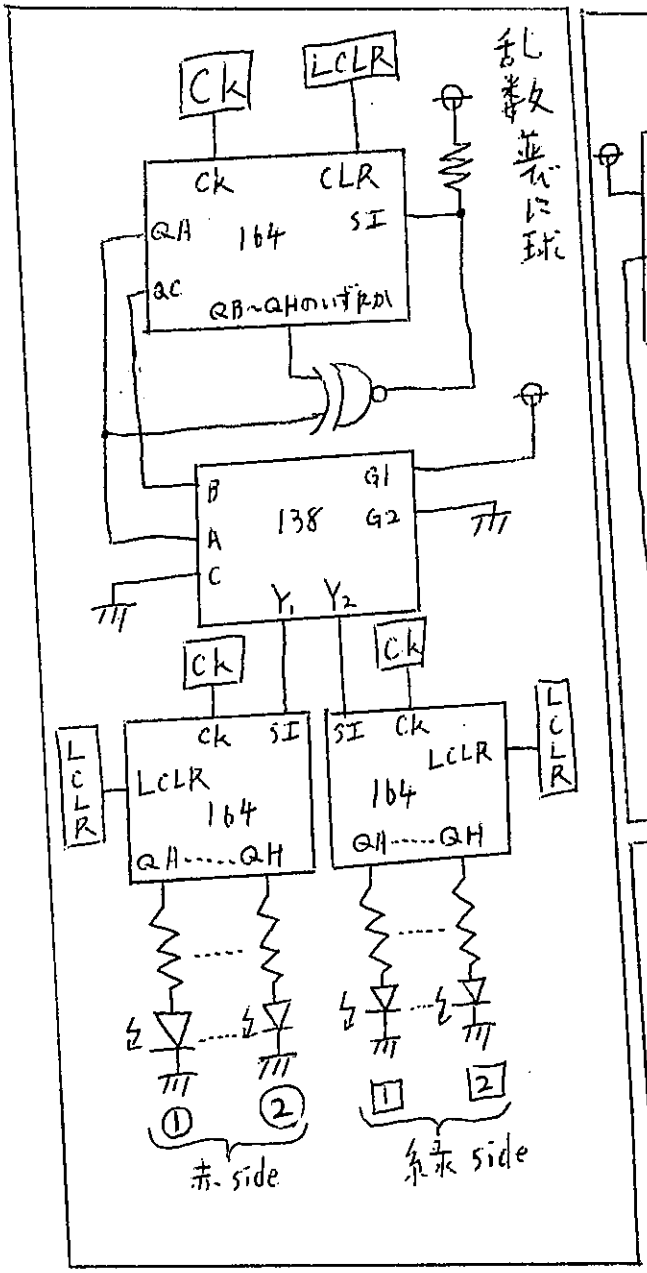
3. ルール

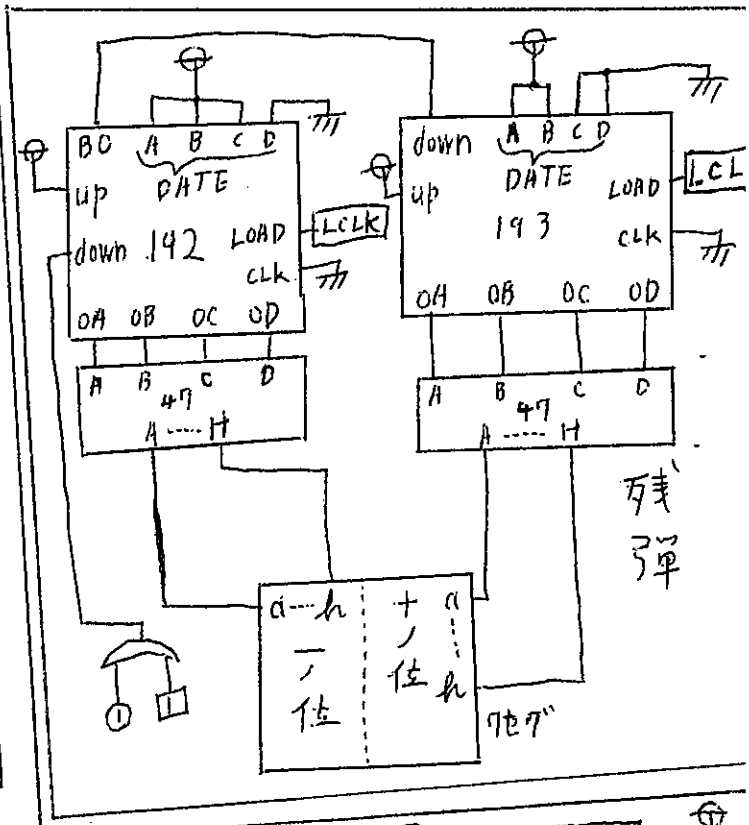
下のLEDから赤か緑の光が流れてくるので
一番上にくる時に同じ色のスイッチを押して下さい。
逃したり一番上が光っていない時にスイッチを押す、
違う色のスイッチを押すとライフが減って0になると終わりです。
光は全部で96個で32個ごとに速くなって流れてきます。

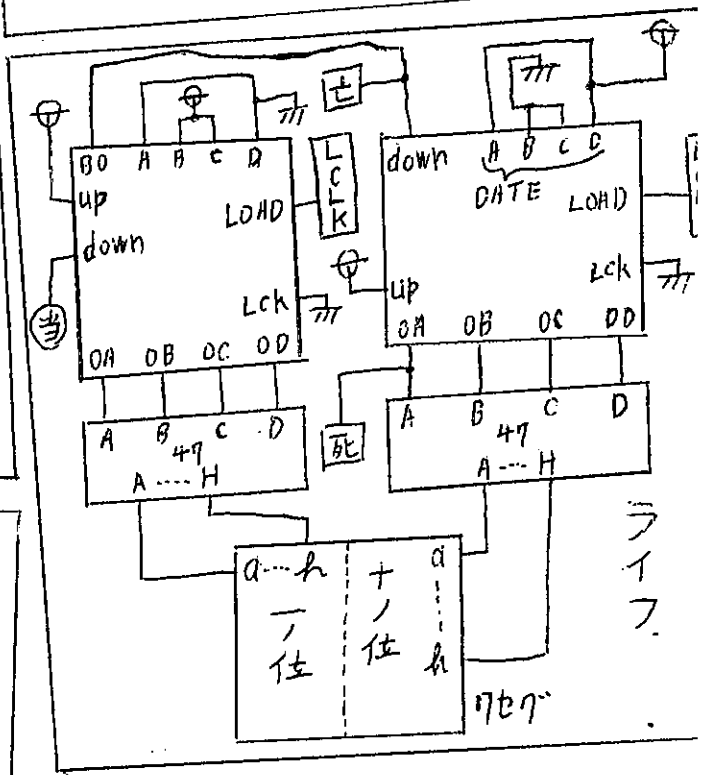
4. 回路図

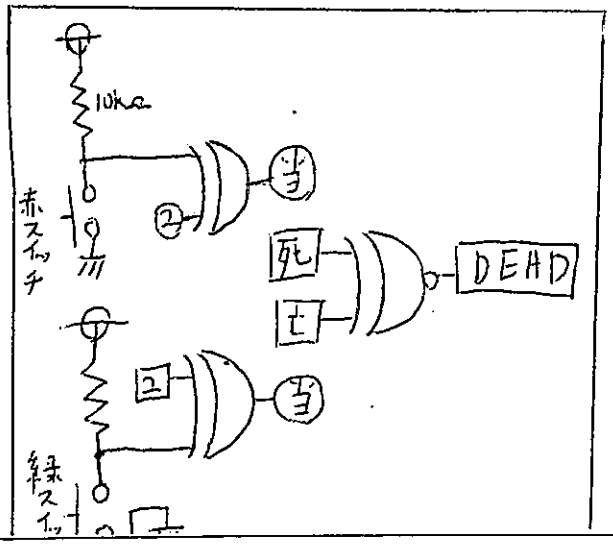












4.感想

今回は回路図は簡単ですが配線の量が僕としては多く、少し大変でした。にも関わらずゲームはどこか見覚えもあり単純な物です。でもそれを楽しんでもらえると幸いです。あと、もう少し配線をきれいにしたかったです。

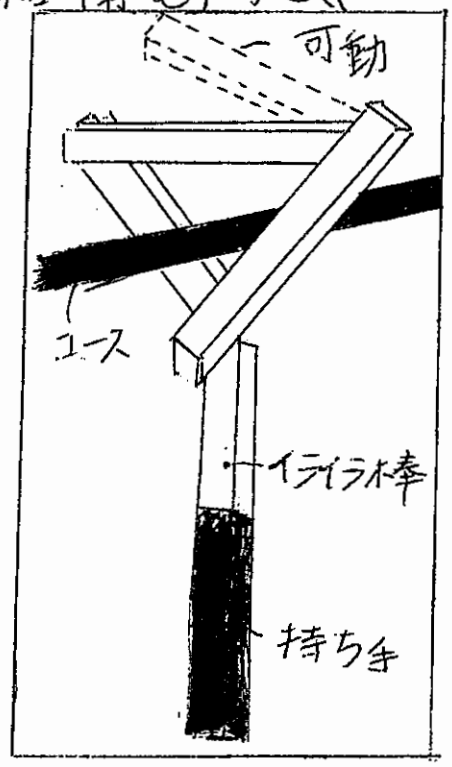
No.

Date

3D イライラ棒 2011

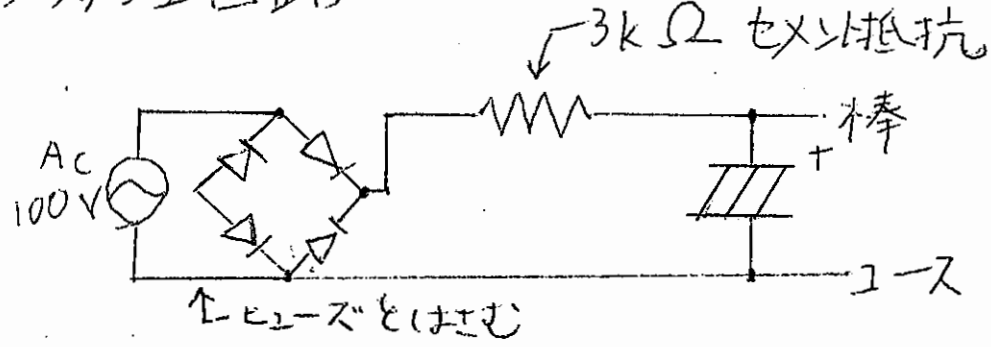
製作 M2 小川佑太
協力 M3 河村さん 岸田さん

今年は例年と違い、3D(立体)のコースです。
数週間で作ったにしては、結構むずかしく、
いい出来だと思います。



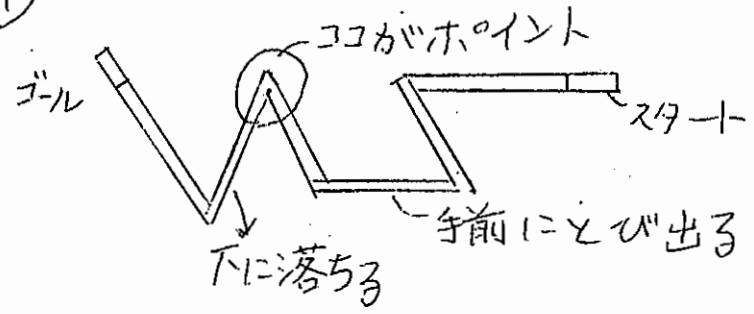
遊び方　イライラ棒の先の三角の部分にコースを通し、当たらないようにするだけです。
スタートとゴールにはビニールテープが巻いてあるので、可動の部分からコースにイライラ棒をひっかけたり、引き抜いて下さい。
コースは四種類あります。
上の二つは右側
下の二つは左側がスタートです。
コースと棒が接触すると火花が散ります。
注意して下さい。

フラッシュ回路

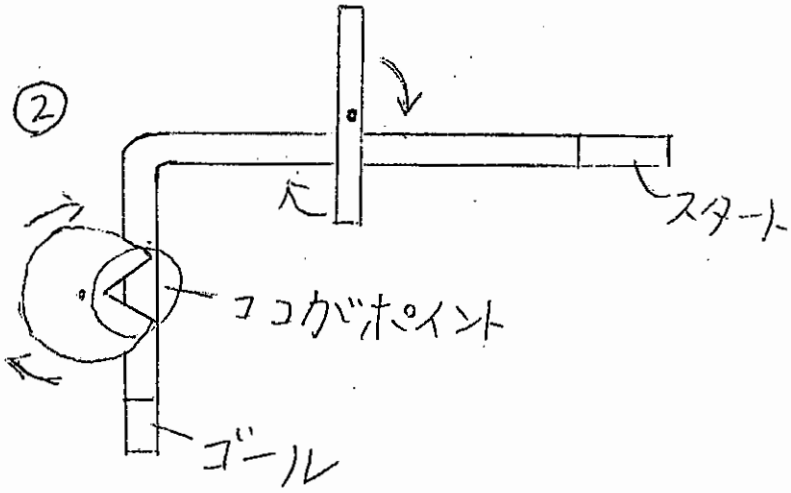


コース

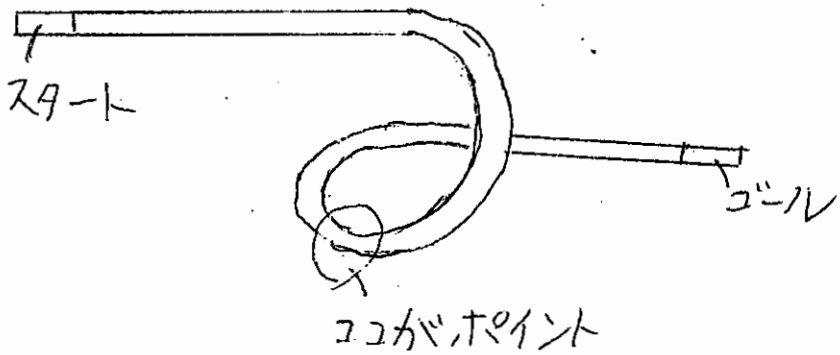
①



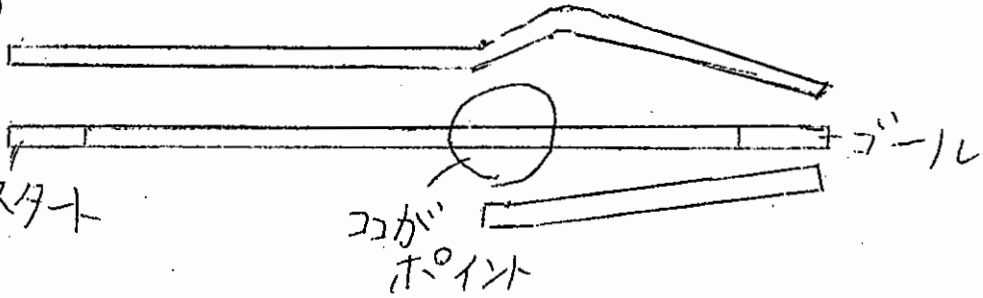
②



③



④



攻略

① 初級です。ポイントの所では手首をまげながら下ろすといいです。

② 丸い板のすきまにタイミングを合わせられればできます。下手したら、ここが一番難しいかも。

③ ポイントの所で持ち手を上からたにまわすのが大変です。少しぶれただけで当たるので、要注意です。

④ はっきり言うと、ここが一番楽かもしれません。ポイントの所で体勢をたて直しましょう。

No.

Date

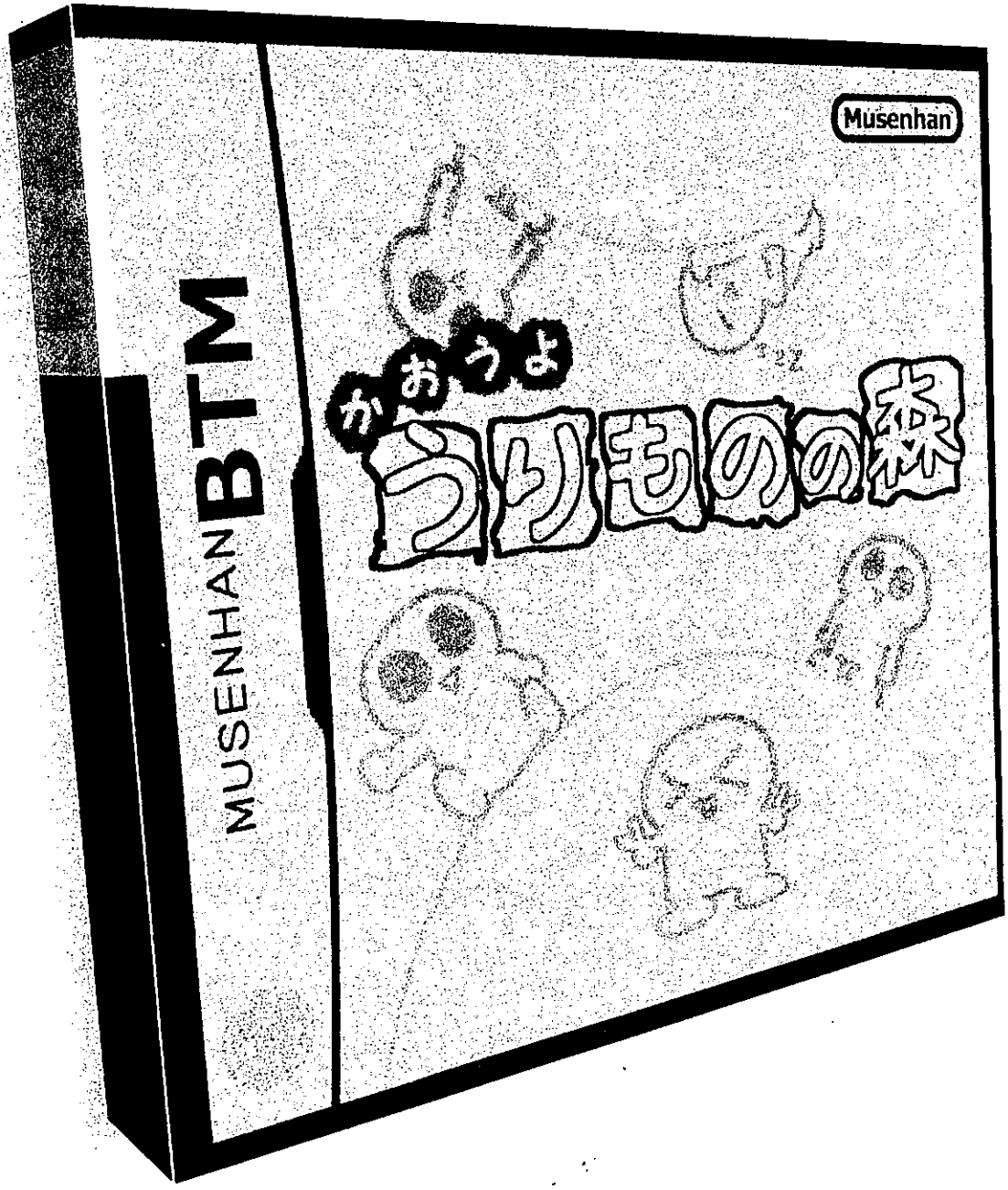
感想

小川　このイライラを作るのはとても大変でした。今まで、毎日ひたすらハンダ付けをしていたのに、アルミ棒を曲げてコースをつくったり、フレームをつくるのに木材を切って、くぎを打って木を切ってのくり返しをして、土木作業員のような2週間でした。でも、いい経験だったと思います。ノコギリの精度と早さが上がりました。来年も支えてくださった先輩方のようにがんばりたいです。

多村　いい何でイライラ作ってんだかまだ理解できないわww

[岸田　かメラの部品が神過ぎる。ブラックボックス回路に使いました。今年は乾電池駆動がもこれ不のおかげ。

再度小川　かなり難しめにつくったので、自分でもクリアしきれていない程です。これがすべてできる人は、人間じゃないかも、というくらいです。ぜひぜひ、たくさん参加人で全コースクリアを目指してください。


Musenhan
かおうよ
うり毛のの森
MUSENHAN BTM

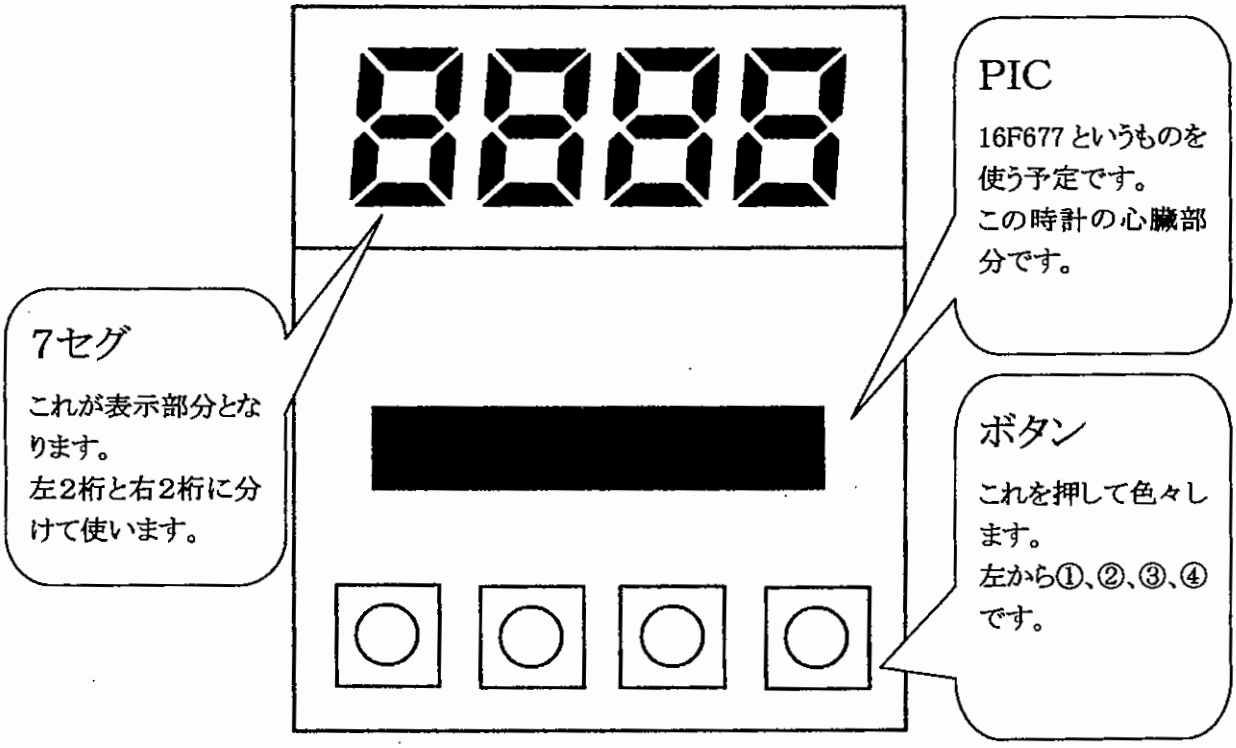
便利時計

製作者：水野太郎(中3)　協力：物理部無線班のみなさん

(1)便利時計とは

時計だけでなくタイマー、ストップウォッチとしても使える一石三鳥のデジタル時計です。

(2)外観　※イメージを模式的に表してあります。



(3)操作方法

・<u>モードⅠ：時計</u>　モード切り替え時に**1111**と表示されます。

普通のデジタル時計です。左2桁が時間、右2桁が分を表しています。

<u>ボタン説明</u>

①モード切り替え：モードⅡに切り替えます。

②桁数切り替え　：数値を変える桁を分、時間に切り替えます。

③＋(プラス)　：数値を＋1します。(例)**12:33→12:34　09:21→10:21**

④－(マイナス)　：数値を－1します。(例)**12:35→12:34　20:43→19:43**

・<u>モードⅡ:ストップウォッチ</u>　モード切り替え時に **2222** と表示されます。

普通のストップウォッチです。左2桁が分、右2桁が秒を表しています。

<u>ボタン説明</u>

①モード切り替え:モードⅢに切り替えます。
②スタート　　:計測を開始します。
③ストップ　　:計測を停止します。
④クリア　　:表示を0に戻します。

・<u>モードⅢ:タイマー</u>　モード切り替え時に **3333** と表示されます。

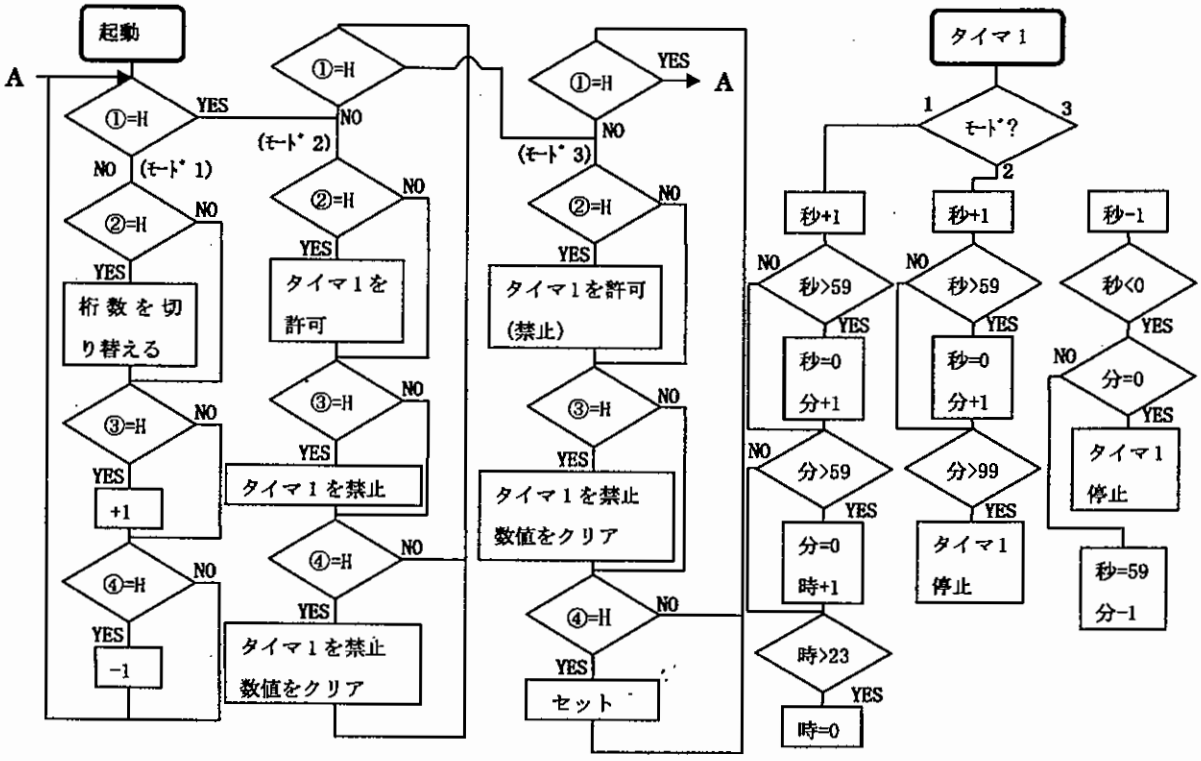
普通のタイマーです。左2桁が分、右2桁が秒を表しています。

<u>ボタン説明</u>

①モード切り替え:モードⅠに切り替えます。
②ｽﾀｰﾄ / ｽﾄｯﾌﾟ　:計測を開始(再開)、停止します。
③クリア　　:表示を0に戻します。
④セット　　:最初の時間をセットします。
　　セット出来る時間は 30 秒、1,2,3,4,5,10 分です。

(4)構造

プログラムは長すぎるのでフローチャートだけ載せます。表示はﾀｲﾏ 0 使用です。



(5)感想

初めての PIC での製作、初めての C 言語と初めてなことが多くて色々な方に聞きまくりました。もっと計画を立て、もっと勉強するべきだったのでこれからは気をつけたいです。

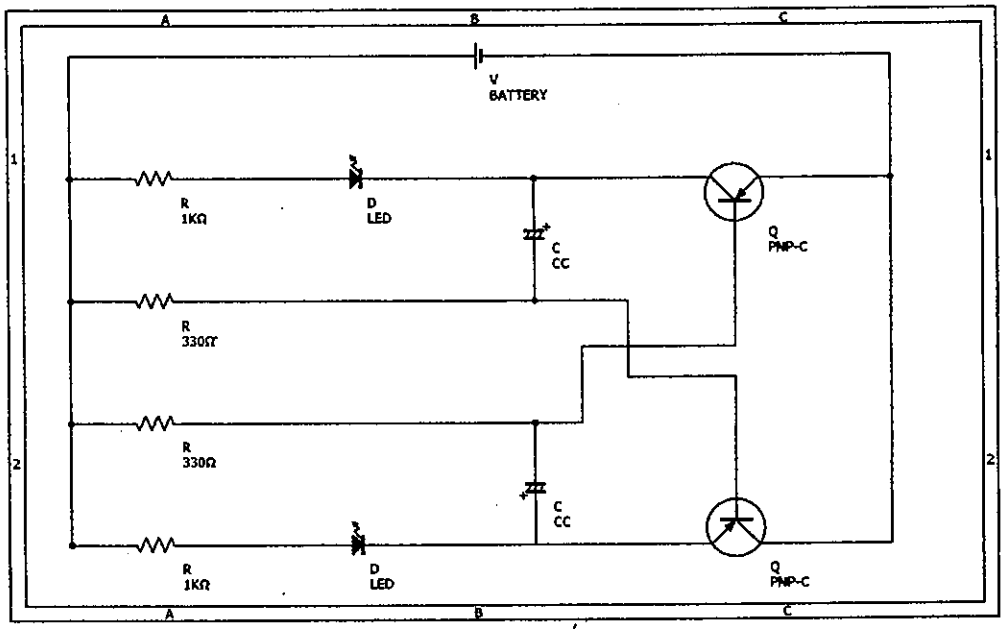
チカチカ

M３　金子　真太朗

・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、製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。

麻布学園物理部無線班
回路図集　２０１１

発行日：　2011/6/20

編集責任者：　伊藤　卓

販売責任者：　鈴木　舜也

デザイン：　朝倉　哲夫

市村　剛大

大森　時生

岸田　聖生

佐藤　惇

PRICED at 100 yen

We Love BUTSUMU

"Don't forget 64th festival in Butsumu"