

電気通信大学 工学研究部

音

報

第

45号

号



工学研究部 部報 第45号

もくじ

作者	題名	ページ
	部長挨拶	2
皆川 太志	時限爆弾を作ってみた	3
石岡 峻一	フルカラーコンサートライトの制作	5
本多 寿矢	2014年度エコラン参戦企画	8
河村 大輝	電子トリス	10
ポン☆デ☆リング☆村上	エレコン制作構想	13
般若 麗沙	PHPで遊んでみた ～作品紹介ページ自動再生スクリプトの作成～	15
山岸 大騎	アーケードコントローラー制作	18
はぐれウナギ	正しいグラウンド設計	19
本多 寿矢	スピーチタイマー制作計画	22
大川 修平	最初の工研部報	23
Mark.IU	疑似科学を科学する ～イオン水についての考察と効果～	24
しゅんりゅ &れいてんし	ドミネーター制作記 -Simple-	27
Hagi	びっこんなっ！の紹介とか	30

部長挨拶

はじめに

こんにちは。

工学研究部の部長、皆川です。

技術的ものづくり的なものを探していた読者の皆さん、この項は 100 %間に合わせ的に書かれたものなので、お気に障るようでしたら読み飛ばして頂いて全く結構です。

それでも、本項に興味を持って頂いた物好きな読者の方に感謝。

まえがき

趣味と勉強の分別をどこでつけるかという問題はこれまでどれほどの大学生の頭を悩ましてきたか検討もつきませんが、共通していることと言えばどちらを極めるにしても我々大学生には時間が圧倒的に足りないということでしょう。ましてや、趣味も勉強もということになれば、その達成確率は極端に下がることは言うまでもありません。迫り来る単位不可に怯え、恐怖し、趣味的活動はおろか人間的活動さえままならぬ学生を何人も見てきました。

しかしそういう状況にも関わらず、単位が犠牲になるかもしれないにも関わらず、己が趣味を磨き、鍛え上げ、創造の光を放つ立派な部員が、工学研究部にはいます。誤解を恐れずに言えば彼らは、おそらく学業重視な電気通信大学（そうではない大学なんて無いのでしょうか）において異端なのでしょうが、同時に誰よりも理工学を追い求める電気通信大学の学生らしくいるのだろうと思っております。そんな彼らの生み出す工学的作品物には、彼らの血と汗と涙と単位と、何より愛が詰まっています。本部報を含め、少しでも関心を寄せていただければ幸いです。今後共、工学研究部を宜しくお願ひ致します。

100 %間に合わせ的に書かれたものだけあってその内容には汗顔の限りですが、これにて工研部報 45 号のまえがきとさせていただきます。次ページから始まる作品群説明を読むかそっとページを閉じるかはお任せします。ここまで読んでくださった読者の皆さんに重ねて 100 %の感謝を。——ありがとうございました——

先進理工学科 3 年 工学研究部 部長 皆川太志

時限爆弾を作つてみた

先進理工学科 3 年 皆川太志

1 はじめに

さて、本記事を手にとって早速 110 番へ連絡しようしてくれたそこのアナタ。あなたの正義感は見上げたものですが、やめてください！これは決して、人に危害を加えるものではありません！本作品は「時限爆弾解除の感覚を楽しむ玩具」です。モチーフは映画とかで電線を切断することによって時限爆弾を解除する感じです。図 1 が完成した時限爆弾です。

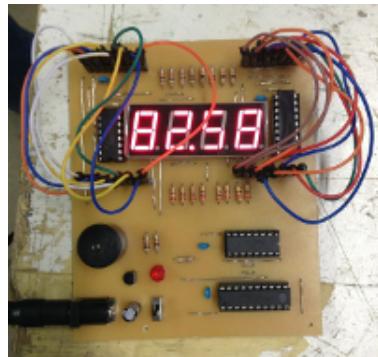


図 1 時限爆弾

2 遊び方

電源となる AC アダプタを差し込み、電源スライドスイッチを ON すると時限爆弾が起動します。起動後はピンソケットからジャンパー線を引き抜くことで、爆弾を解除していきます。表示されている残り時間が 0 になる前に、当たりのジャンパー線を引きぬいてタイマーを止める事が出来れば爆弾解除成功となります。もちろん、一発アウトや残り時間が 10 秒減るといったハズレ線も多数存在しているのでご用心を。もし解除に失敗した場合は、白い閃光と轟音とともに・・・ということではなく、けたたましくブザーが鳴り解除失敗を知らせます。

3 機構説明

図 2 が時限爆弾の構成の簡略図です。

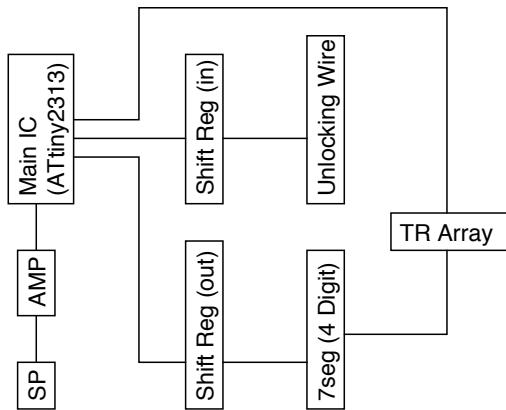


図 2 簡略図

メイン IC は ATtiny2313 です。出力用シフトレジスタは 4 枚 7 セグの文字の表示のデータを ATtiny2313 から受けて、7 セグを光らせます。入力用シフトレジスタは解除線の切断判定を ATtiny2313 に送ります。トランジスタアレイは四枚 7 セグのどの桁を光らせるのかを選択し、ダイナミック点灯させます。スピーカーは爆弾爆発時にブザー音を鳴らします。

4 感想（苦労したこととかいろいろ）

いまいの一言で製作が始まった時限爆弾ですが、いろいろあって完成が遅くなってしまいました。いやーほんとすいません。

苦労したことは、まず ATtiny2313 には ATmega328 のようにハードウェア SPI(Serial Peripheral Interface) が内蔵されていないので、他の機能で代替する必要がありました。じゃあ ATmega328 を使えよって思うかもしれません、茨の道を通るのが工研部員です。今回は ATtiny2313 にもハードウェア内蔵されている USI(Universal Serial Interface) を 3 線同期駆動することで、SPI と同じ動作をさせてみました。

結果として、ソフトウェアで実現する方が開発が早かったと思います。でもこれで ATtiny2313 でもハードウェアを使って高速に SPI 通信できるようになったよ！

フルカラーコンサートライトの製作

Creation of full-color concert light

石関 峻一
Shunichi ISHIZEKI

電気通信大学 工学研究部
(〒182-0021 東京都調布市調布ヶ丘 1-5-1, twitter id: ueckoken)

Abstract: It is one that is often used in live voice actor and full-color pen light. However, there is no agility and coordination very current. So, I, to produce a full-color study penlight with a corresponding property and coordination.

Key Words: LED, Full-color, Concert, Voice actor, cyalume

1. はじめに

コンサートライトとは、コンサートでよく用いられる長さ 25cm の LED により発光する棒状のアイテムである。主に声優系のライブ用いられ、公式グッズとしても、見ることが多々ある。2000 年代前半では、化学反応で発光するサイリュームやポケットネオンが主流であった。しかし現在は LED を用いたコンサートライトに主流が移った。ライブの現場（会場のこと）では、様々なコンサートライトを見ることが可能である。コンサートライトは単色ばかりではなく、フルカラー LED を用いたマルチカラーのコンサートライトもまた登場してきた。

そこで、なぜ自作するのかというと、市販されているフルカラーコンサートライトは、ボタンがひとつしかないため操作性にかけ、優れている構造とは呼べません。また、複数本を同時に持つなどをした時、個別に色を帰る必要があり、時間がかかり、コンサートを楽しむ時間の一部を、コンサートライトの色をかえる時間に奪われてしまう。よって、私は、複数本を目的の色に一操作で達することのできるフルカラーコンサートライトを開発を行う。

2. LED の明るさの制御方法

具体的な話をする前に LED の明るさを制御する方法について、語りたいと思います。LED の発光はほぼ、電流に比例します。

2.1 LED の基礎（制限抵抗）

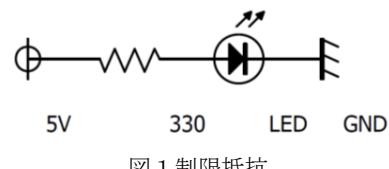


図 1 制限抵抗

みなさんは LED を光らせるときには必ずこの【制限抵抗】を入れていると思います。なぜなら LED は、ダイオードの一種であるため、電圧と電流が非線形性であり、ある程度の端子間電圧をかけると一気に大電流が流れ、素子を自ら破壊してしまうのだ。つまり、手間のかかるかわいい子なのだ。

一般的に LED の制限抵抗を決めるときの式は、

$$\text{制限抵抗 } R = \frac{\text{(電源電圧} - \text{LED の順方向電圧)}}{\text{流したい電流値}} \quad (1)$$

となる式を、使っているはずである。

この式の表している意味は下図のようになる。

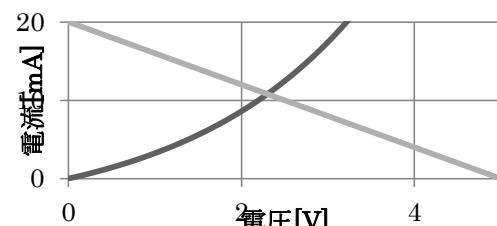


図 2 制限抵抗による LED の制御

0V から出ているのが、LED の I-V 特性、5V から出ているのが制限抵抗によって LED の逆起電力がその電圧だった時に流れる電流である。

つまり交点の電圧電流が LED にかかるのである。こんな感じで LED に流れる電流は決定され、LED の明るさが決まる。

この抵抗を入れるだけの方法の長所は、とても簡単な構成なことである。

欠点としては、LED は自らの温度によって I-V 特性が変化してしまうために、絶えず同じ電流が流れるわけではなく明るさも変化してしまうこと。電源電圧が変化すると電流も変化すること。電力を抵抗でも消費するため、効率が悪いことがあげられる。

2.2 定電流回路

電源電圧の変化によって LED の明るさが変化したり、LED の定格電流を超えていたりするのは、問題である。それを解決する方法が定電流制御である。これはどの電源電圧であっても、電流値が一定になるようにする制御方法である。

2.2.1 CRD

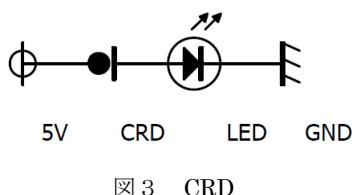


図3 CRD

この CRD は定電流ダイオードというものである。ある一定の電圧がかかっている時にその CRD に流れる電流は一定となる。

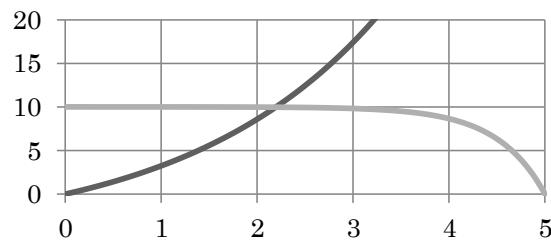


図2 CRD

これにより、電源電圧が変化しても明るさが変化しない回路になった。

しかし、CRD には 3V~の端子間電圧をかけないと、一定の電流を流さないなどがある。さらに~30mA までの商品がほとんどである。

2.3 トランジスタ

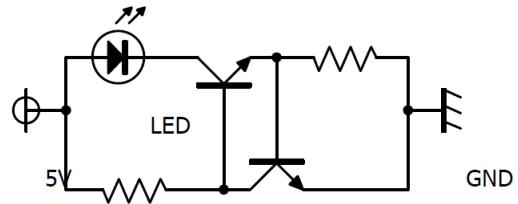


図5 トランジスタを用いた定電流回路

この回路では、CRD では出来なかったパワーLED のドライブが可能になる。

2.4 PWM 制御

さて、これまで紹介してきた LED の光らせ方では、LED の明るさを変化させるときには、抵抗を取り替えるか、可変抵抗などを手で操作するしかない。

これでは LED をマイコンなどから明るさを変化させることは出来ない。そこで、PWM 制御という方法で行う。

PWM 制御とは、素早く LED を ON, OFF させ、ON の時間と OFF の時間の割合を変化させることで、見かけ上の LED の明るさを変化させる方法である。

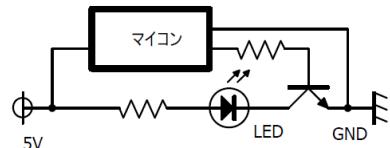


図6 PWM制御

この方法の長所は、マイコンと相性がよく、簡単に輝度制御が行えることです。また、抵抗で消費する電力を小さく出来ます。なぜかは考えてね。しかし、この方法でも抵抗などは使用しており、すべての電力を LED にまわせてはいません。

2.5 定電流制御

この方法では、抵抗器からの発熱は少ないので、とても高効率に電源からの電力をほぼ LED で消費する電力にすることが出来、発熱も LED からのみになります。

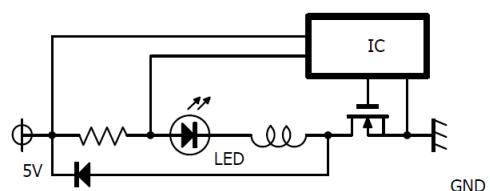


図7 定電流制御

低抵抗で LED に流れる電流値を測り FET をほどよくスイッチングすることで、一定の電流が LED に流れます。

この方法のメリットは、常時 LED が光っているのでちらつかない。とても高効率で光らせることができる。短所は回路が複雑になるなどです。この制御法でこそ LED の持っている特性を十分に活かすことができます。

3. 機能

4 本を同時に色切り替え

4 段階明るさ切り替え

とても丈夫

とても明るい

明るさが変わらない

4. 回路構成

4.1 LED 部

スイッチング制御により、定電流回路を作成した。抵抗だけでは効率や電流制限抵抗からの発熱が多くなってしまうため採用した。

4.2 制御部

AVR マイコンを使用し、全てのペンライトを同一マイコンから制御を行った。

4.3 入力部

スイッチマトリクスを用いて 20 個のスイッチによる入力がある。一つのボタンで一つの色であるため、一度の操作で、任意のプリセットカラーにすることができる。

さらにボリュームを 3 つ載せることにより、任意の色を登録し、使用することもできる。

5. 発光部構成

5.1 10W フルカラーLED

一般的なフルカラーぺんライトは 1.5W なのでかなり明るい。定電流制御を行っている。定電流制御 IC には PT4115 を用いた。

5.2 ヒートシンク

10W の LED には放熱板が必要となる。

5.3 ファン

自然風冷だとヒートシンクの大きさが大きくなってしまうため、ファンを取り付けることにより強制空冷を行った。

5.4 発光部

アクリルパイプで作成を行った。アクリルパイプ内で光が反射し、棒全体が光る。

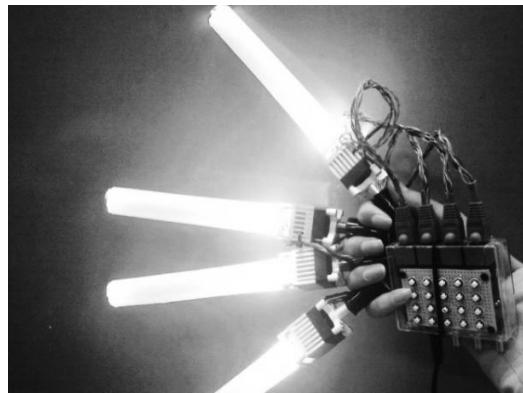


図 8 完成品

6. 考察と残された課題

色合いと明るさにおいては、どの既成品よりもいいものが出来た。操作性においては、スイッチが手首の位置に来るため、もう片方の手を用いて操作を行う必要がある。重量が片腕で 500g するため、ダンベルを振っているような運動となることである。次に 12V 以上の電源を必要とするため、ラジコンバッテリーなどを用いる必要があることである。

今後は全てのペンライトを別制御とし、グループの歌手を応援しやすいようにする。さらに出来れば軽量化を行う。またグラデーション機能などもあるといいと感じた。

7. 参戦実績

7.1 NANA MIZUKI LIVE JOURNEY 2011 埼玉

7.2 NANA MIZUKI LIVE CASTLE 2011 -QUEEN'S NIGHT-

7.3 NANA MIZUKI LIVE CASTLE 2011 -KING'S NIGHT-

7.4 NANA MIZUKI LIVE UNION 2012 群馬

7.5 NANA MIZUKI LIVE UNION 2012 千葉

7.6 アニメロミックス presents NANA MIZUKI LIVE GRACE 2013 -OPUS II - supported by JOYSOUND Calbee ポテリッヂ 1 日目

7.7 アニメロミックス presents NANA MIZUKI LIVE GRACE 2013 -OPUS II - supported by JOYSOUND Calbee ポテリッヂ 2 日目

7.8 魔法少女リリカルなのは THE MOVIE 2nd A's BD&DVD 発売記念イベント リリカル☆パーティー V 1 日目

7.9 魔法少女リリカルなのは THE MOVIE 2nd A's BD&DVD 発売記念イベント リリカル☆パーティー V 2 日目

7.10 アニメロミックス presents 田村ゆかり LOVE LIVE 2013*Cute'n Cute'n Heart* supported by JOYSOUND 1 日目

2014 年度エコラン参戦計画

知能機械工学科 2 年 本多寿矢

始めに

来年 5 月の GW に秋田で行われるワールド・エコノ・ムーブに向けて、工学研究部では新機体を製作し参戦する予定である。なので、その機体製作の計画についてまとめようと思う。

ワールド・エコノ・ムーブとは

大会側から支給されるバッテリーを用いて、2 時間のレース時間内に走る距離を競う省エネルギー競争である。秋田県大潟村のソーラースポーツラインで 1 周 6km のコースを利用して開催される。各チーム（ネタ枠を除く）空気抵抗を極力減らし低燃費に磨きをかけたマシンを製作しており、その走行距離が 90km にも及ぶチームもある。

設計

はしご型のフレーム構造をもつ平たい車体を設計中である。高さ方向への厚みがあまり無いため、角パイプでフレームの本数を増やして強度を上げようと計算中である。

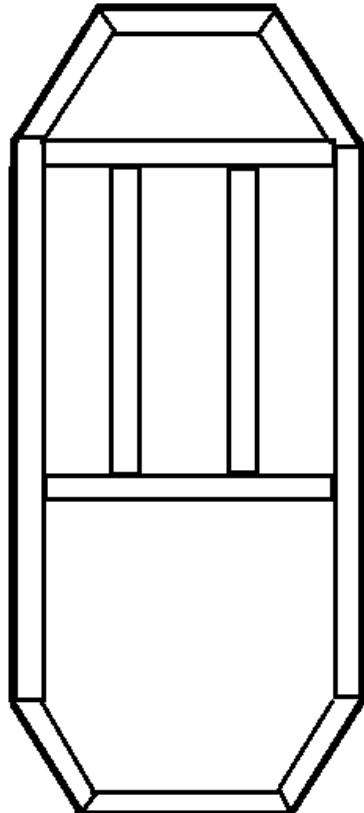


図 1 底面フレーム設計イメージ図

デザイン案

- 果たしてブームは過ぎ去っていないだろうか。駆逐艦島風型のデザインである。速きこと、島風の如し、です！唯一の問題はこんな画像のような再現性は出せないのであることである。



図 2 島風型エコランカー

- 比較的安価なプラスチックダンボールを用いた外装である。うまくカーブを出して空気抵抗の少ない車体を作れれば良いと思う。

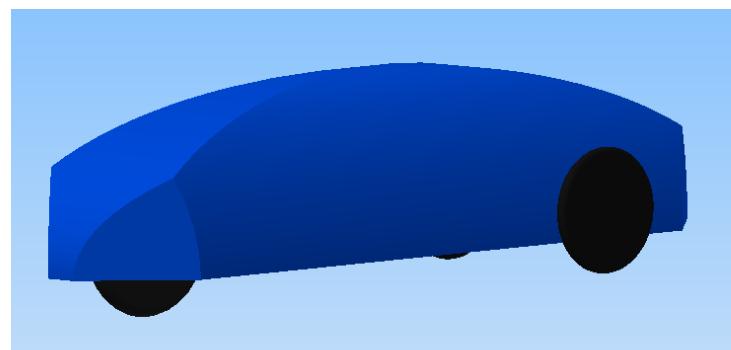


図 3 理想のプラダン製エコランカー

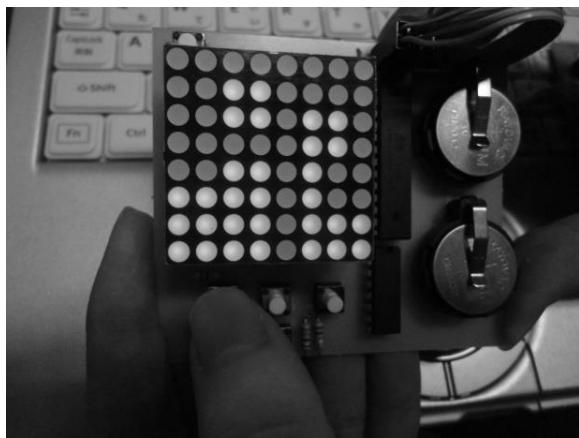
終わりに

現在は機体設計の段階であり、手探り状態で作業中である。断面 2 次モーメントやら何やらを思い出しつつ計算して、走行中に真ん中からボッキリということがないように設計している。来年の 5 月に向けて、皆と協力しながら何とか形にしていきたい。形になったらいいなあ。

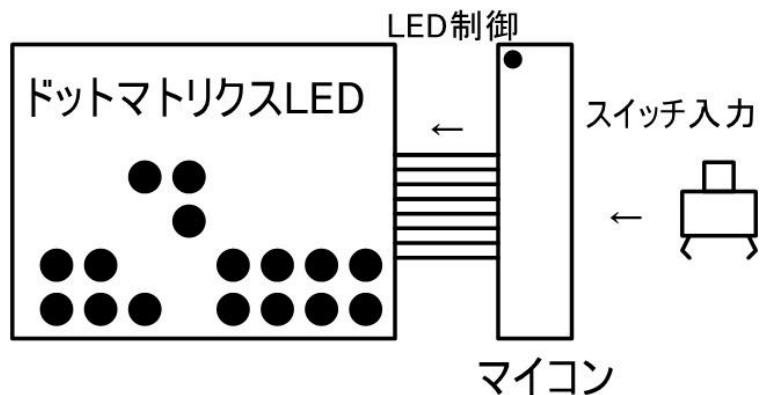
電子テトリス

河村 大輝

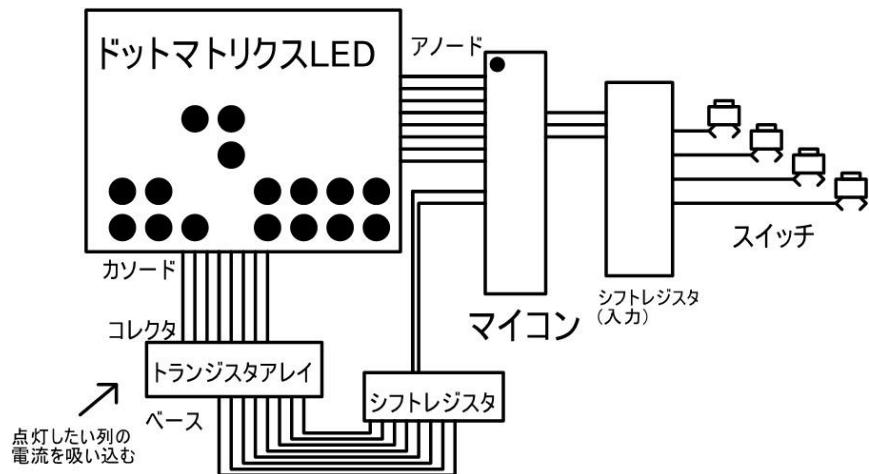
目黒会講習という小学生に電子工作を教える催しがあります、そこで使うために作ったテトリスについて紹介します。



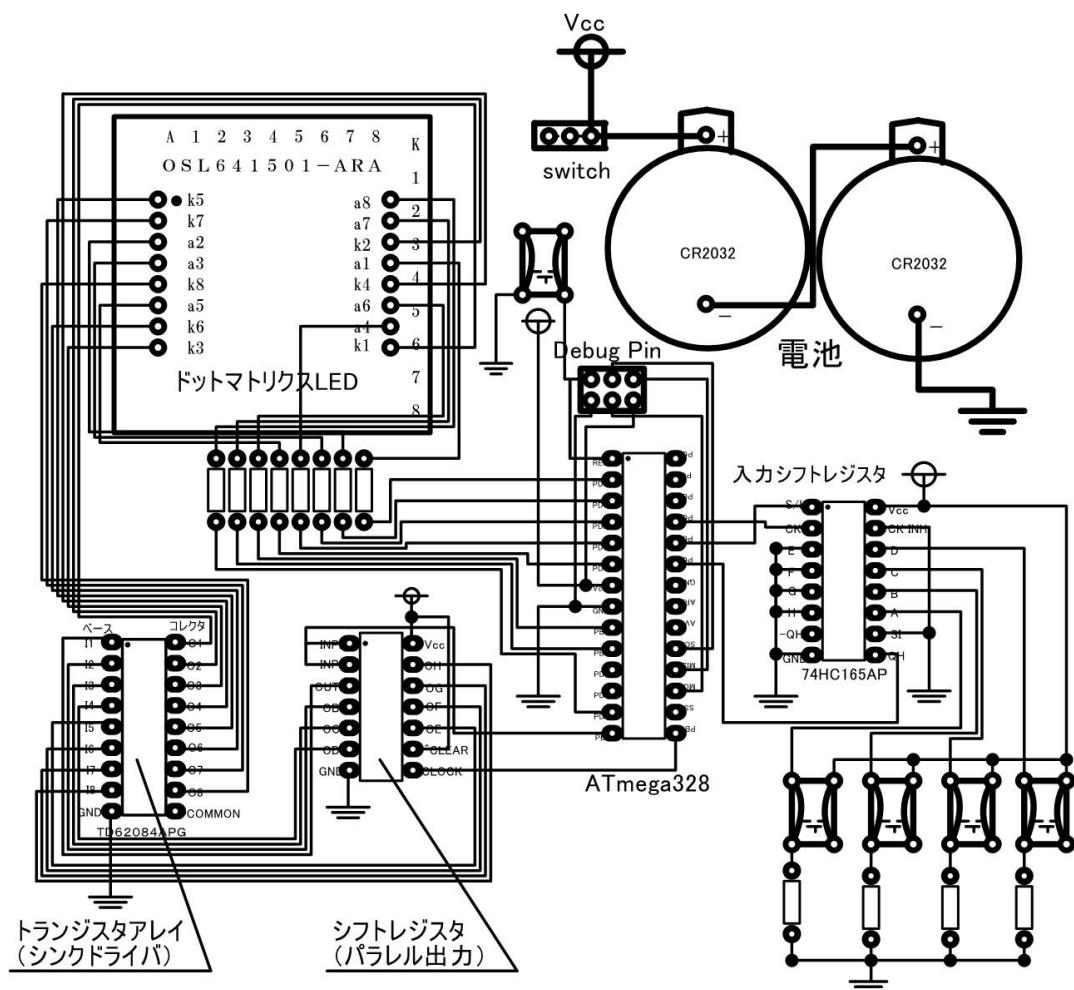
- ・ざっくりした概要



- もう少し詳しい概要



- 回路図みたいな何か



- ・ソフト面

LED 制御

- ・シフトレジスタをシリアル信号で制御してマイコンのピン数を節約しています。
- ・マイコンからの出力ピンで、LED のアノードに電圧をかけ、カソードをトランジスタアレイから吸い込むことで、任意の行を点灯できます。
- ・上記の操作をタイマー割り込みを使い、目にも止まらぬスピードで行うことで、ドットマトリクスを制御しています。

スイッチ入力

入力シフトレジスタはパラレル入力ピンの H/L をシリアル信号で出力してくれます。それをマイコンで受信してドットマトリクス LED に反映しています。

今後の予定

目黒会講習には間に合わないとは思いますが、シフトレジスタで節約した残りのマイコンのピンを使って、いろいろ追加機能を作つてみようと考えています。

1. テトリス基板同士の通信機能

テトリス同士で通信し、画面の拡張をできるようにしたい

2. 音声出力

PWM 信号を出力し、BGM を鳴らせるようにしたい。

3. 液晶画面

液晶ディスプレイなどを使って、点数や次に落ちてくるブロックの表示などをみたい。

ここまで本格的になるとドットマトリクス LED じゃ役者不足かもしれない。

おわり。

エレコン制作構想

1S ポン☆デ☆リング☆村上

平成 25 年 8 月 19 日

概要

エレクトロニクスコンテスト参加に向けて創作物の概要を記す。なお本記事は全て構想段階であり脳内設計であるため制作の変更は大いに起こり得る。

1 目的

自分がこれまで培った工作技術を復習、統合し、今後身につけるべきスキルを確認するためにエレクトロニクスコンテストに参加する。植物の栽培には定期的に水やりをすることが望ましいが、忙しい生活故、忘れてしまうことがある。そこで、自動で散水し、またデータ取得をすることで効率的な栽培を行うロボットを作成する。

2 経緯

自分の過去の主な制作物に以下のものがある。エレクトロニクスコンテストに向けて受験勉強によるブランクを払拭すべく、以下を踏まえた新作を制作する。

2.1 CANSAT

高校 2 年の夏、能代宇宙イベント・カンサット甲子園参加にあたり、電装班の一員として制作に参加した。自分はキャリアに搭載する小型データロガーの制作に携わったが、過剰な小型化に自身のスキルが追いつかず、その結果未完成となり非常に悔しい結果となった。カンサット甲子園で自分の成果物と言えるものは存在しないが、初めてプロジェクトに参加して制作を行つたこのイベントでの経験は有意義なものであり、自分の創作活動の原点となっている。

2.2 花の水やりき

初めて自身で設計を行つた作品である。高校 2 年次の文化祭で展示した。これはカメラ、エアポンプ、LED ライトを AVR マイコンで制御し、周期的に撮影、水やり、ライト照射を行う機械である。マイコンプログラミングや電子工作の初歩を習得、および自分の生活をより便利にするために制作を行つた。

2.3 ハイブリッドロケット

高校 2 年の冬から翌年の 6 月頃にかけて、ロケットガール＆ボーイ養成講座に参加し、電装班としてハイブリッドロケット内部電装及びペイロードの制作に携わった。ロケット内部電装の作成は当初サポートして下さった大学院の方の設計の下、基盤設計、切削、工作、

PIC プログラミング等を行なつていたが、打上げ 2 週間前前に急遽設計を我々が再設計することになり、そこでは SunSPOT を用いてデータロガーの制作を行つた。また、ロケットの搭載物としてローバーの制作を行つた。これはロケットから放出後、減速機構を用いて安全に着地、キャリアから離脱し地上を走行し、その後本体に内蔵された FM ビーコンを八木・宇田アンテナで受信し、回収を行うのが一連のミッションであった。自分はそこで、ローバー制作の主担当として各担当との調整やローバー内部電装の制作を行つた。

3 ミッションシークエンス

図 1 に本製作における動作の流れ図を示す。

まず、拠点にて気圧、湿度センサ等を用いて天候データを取得し、晴天／雨天で出動判定を行う。出動した場合、目的地まで GPS、方位センサ等を用いて誘導する。途中、障害物がある場合、距離センサを用いて検

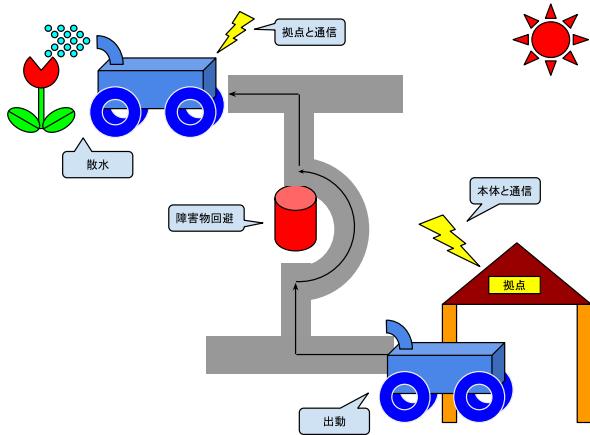


図1: ミッションシーケンス

出し、回避する。目的地に到着後、写真撮影、温度、湿度、気圧等のデータ取得後、散水し、同様に帰還する。拠点にて収集したデータを蓄積、解析し、効率的な栽培に役立てる。また、インターネットを用いてスマートフォンから遠隔操作、監視を行う。

4 サクセスレベル

前項のミッションをすべて達成するのは難しい。そこで、個別にサクセスレベルを設定する。サクセスレベルが低いものから順に制作を行う。

4.1 サクセスレベル1

- 走行
- 散水
- 物理データ取得

4.2 サクセスレベル2

- 自動走行
- 障害物回避
- 出勤判定

4.3 サクセスレベル3

- 拠点製作
- 無線通信
- データ解析

4.4 サクセスレベル4

- オンライン化
- スマートフォンアプリ製作

5 製作

現在、製作環境の構築段階であり、実際に製作が始まるのは9月頃を予定している。完成品はエレクトロニクスコンテストに出展する予定であるが、機体製作以外にもプレゼン作成等の必要があるため、そちらにも注力していきたい。

PHP で遊んでみた

～作品紹介ページ自動生成スクリプトの作成～

情報理工学部知能機械工学科 3 年 般若麗沙

1. はじめに

世の中には 2 種類の人間がいます。HTML を書きたい人間と書きたくない人間です。電気通信大学では 1 年次の必修授業で HTML を習うため、ほぼすべての電通生は HTML を理解し、使うことができるのだと思いますが、電通生は課題や研究に大忙しなので HTML を書くのが趣味っていう人でも無い限りあんまり書きたくないものだと思います。

ところで、工学研究部のホームページには部員が製作した作品を紹介するページが用意されていたのですが、誰も更新しないで長い間「準備中」と書かれているだけでした。工研の活躍を全世界に知らしめるために存在するホームページだというのにそれではあんまりです。工研では日々たくさんの作品が作られているのに、なぜみんな更新しないのかということを考えた結果、HTML を編集して色々書くのが面倒くさいのではないかという結論に至りました。そこで、工研の広報のためにも、手軽に自分の作品を紹介できる PHP スクリプト（以下「作品紹介ページ」）を作ってみました。

2. 概要

作品紹介ページを作るにあたっての条件として

1. 投稿・編集・削除は工研部員のみが実行できる（認証ページ）
2. 画像や動画も一緒に投稿できる
3. 大きいデータや怪しい拡張子のファイルは投稿できない
4. 紹介文などページの本文だけを書けば自動でページができる

ということを考えて作りました。PHP を書くのはほぼ初めてだったので、下手なプログラムを書いてセキュリティホールができたら怖いなと思っていたのですが、Basic 認証を信じることにしました。今のところ特に問題はなさそうなので、多分大丈夫だと思います。また、作品紹介は動画や画像があってこそだと思うので、画像・動画のアップロードにも対応しました。これは作品紹介ページを作る上で大変だったところの一つなので、部員にはぜひ利用してもらいたい機能です。

では、作品紹介ページの細かい機能を説明していきます。

2. 1 新規投稿

The screenshot shows a web form for creating a new post. It includes fields for 'Name' (名前), 'Work Name' (作品名), and 'Work Summary' (作品概要). Below these are two sections for attachments: 'Image (Up to 300KB, gif/jpg/png only)' and 'Movie (Up to 2MB, MP4 only)'. Each section contains a file input field and a list of attachments. A note at the bottom says 'Please check this if you do not want a link to your page to be generated.' (ページへのリンクを生成しない場合はチェックを入れてください。) and a 'Submit' button (送信).

Fig.1 新規投稿画面

新規投稿をするためには、新規投稿用のページにアクセスします。新規投稿用ページは認証が必要なページなので、IDとパスワードを知っている工研部員しか基本的に入れない仕組みになっています。

Fig.1を見ればわかる通り、名前・作品名・作品概要・画像(5枚まで)・動画(5本まで)を書くだけでHTMLが自動生成されます。プレビュー機能を実装するまでの元気がなかったので、「作品の紹介ページは作りたいけど、工研HPにページへのリンクは貼らない」という選択ができるチェックボックスもつけました。

投稿ボタンを押すと、ランダムに生成された文字列をファイル名とするHTMLファイルが出来て、そこには内容が書き込まれます。レイアウトに関するコード

はどのページも同じなので、その部分はPHPによって自動で書き、ほかのところは新規投稿画面で書き込んだデータを写すようにしています。

PHPには、テキストボックスで行った改行に対して自動的に改行タグをつけてくれたり、画像ファイルを生成してくれたり、便利な機能がたくさんあることが驚きました。PHPは動的なウェブページを実現するためのものなので当然といえば当然なのかもしませんが、なんだか甘やかされているような気がします…。

2. 2 修正

The screenshot shows the raw HTML code for a work introduction page. The code includes meta tags, a link to a CSS file, and various HTML elements like headings, paragraphs, and images. A note at the top says 'This page can be used as a homepage.' (このページはホームページとして使うことができます) and 'Please click the "Update" button after saving the changes.' (変更を保存した後、「変更」ボタンを押してください) followed by a note about browser compatibility.

Fig.2 修正・削除画面

いたってシンプルなHTML編集ページです。作品紹介ページのHTMLコードをまるごと持ってきて、まるごと書き換えるという算段です。修正ページを作る頃には既に疲れてしまっていたので、投げやりな感じになってしまいました。最初の目標である「HTML編集の手間をなくす」はどうしたというお叱りを受けそうですが、逆に考えると、気の向いたときに自分で好きなレイアウトに変更することができるのです。すべての編集を許す自由度の高さは他に類を見ないと思います。

この方法は下手なコードを書いてページの表示がおかしくなる危険性がありますが、工研部員ならそんなへマはしないだろうと信じてこのようにしました。

2. 3 削除

消す時はあつという間です。この記事のためにサムネイルを用意する間もなく、確認画面でOKを選択すればページは消えます。そのページに添付した写真や動画も一緒に消えます。立つ鳥跡を濁さず。

2. 4 サムネイル



Fig.3 サムネイル

工研 HP の作品/活動ページの様子です。それぞれの作品紹介ページへのリンクがサムネイルで表示されていることがわかります。このように、ページに投稿された最初の画像を縮小してサムネイル表示することで、より直感的に作品に出会うことができます。投稿する側としては、いかに視覚に訴えかける写真が撮れるかという勝負になりますね。

3. おわりに

作品紹介ページをせっかく作ったのに、まだ投稿数が少なくて私は悲しいです。みんなこの機能の存在を知らないのか、それともまだ面倒くさいのか…。きっとこの先もっと紹介できる作品数が増えると期待しているので、部員のみなさまよろしくお願いします。

なお、作品紹介ページは今も工研ホームページの「作品/活動」から閲覧することができる所以、見たことのない方はぜひ見てください。URLは奥付に書いてあります。

アーケードコントローラー制作

始めに

最近（といっても結構経ちますが）、某先輩の影響でSTGを始めることとなったので、なにかにかこつけて物を作りたがる私はアケコンを作ろうと思いました。そこで今回は、

- ・現行で最も普及しているVIEWLIX[TAITO]のボタン配列と、未だに稼働数の多いBLAST CITY[SEGA]他、旧筐体のボタン配列どちらにも対応出来る筐体

であることを念頭に作り始めました。

仕様

- ・ボタンレイアウトの置換可能 (VIEWLIX/ ブラスト / 他)
- ・レイシリーズを遊べるために、対応ハードは少なくともPS/SS
- ・新作のためにXbox360対応

筐体設計

ボタン配置する部分を別パーツにして制作します。他の形は、大体VIEWLIXを参考にしています。

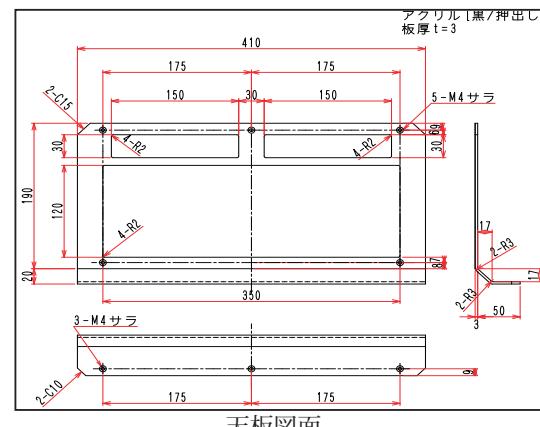
ボタンメンテナンス時に、ボタンへアクセスするために開けるネジを少なくするために、トルクヒンジを使って開閉式にしています。

また触った時の冷たい感じを無くすため（いつものレーザー加工機を使うために）、素材はアクリル板で作ります。

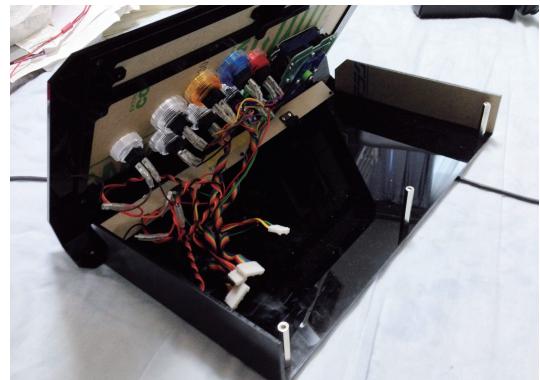
便利です。レーザー加工機。



外観



天板図面



パネル展開図

あとがき

あれっ？回路は？と思った方ごめんなさい。まだできてません。

回路の方は、過去部報に書いてあるPS互換コントローラー制作などを見ていただけだと、だいたいこんな感じのものが出来上がると言うのが解っていただけると思います。

とりあえず、決まっていることは、PS対応->自作 / SS対応->自作 / Xbox360対応->手持ちの360基板使用ということです。PSはSPI通信with Acknowledge。SSはページ切り替え方式と、自作が簡単なのですが、Xbox360はただのHIDジョイパッドというわけではないので、実装が難しいです。（そもそも自分でUSBの実装出来ませんが）なので、基板を乗っけようと思います。

まあ、プレイに関係する筐体までは出来上がっているので良し。

正しいグラウンド設計

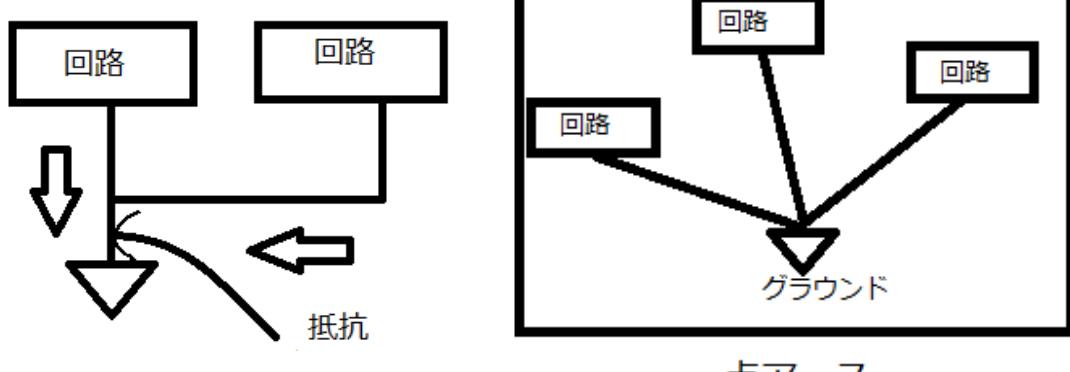
はぐれウナギ

初めに

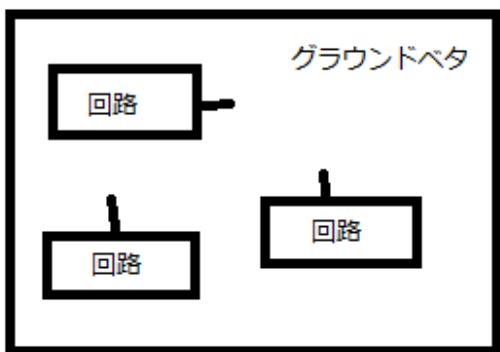
回路を設計する上で、気を付けなければならない点の一つにグラウンドの設計が有ります。何故なら、グラウンドは回路のすべての電位の基準となる点だからです。そこで、実際に回路を作る上でどのように気をつければ良いのかという事を説明したいと思います。ただし、ここで説明する内容はあくまで気を付けなければならない点の一部だという事に注意しなければなりません。今回説明する内容を守れば、必ずグラウンドのノイズが消えるわけでは有りません。しかし、守らなければグラウンドノイズの原因となる可能性が有ります。回路を作る時、頭の片隅でこの事を思い浮かべていただければ幸いです。

一点アースと全面アース

回路を作る上で、良く、一点アースが重要と言われます。この一点アースというのは、図の様にすべての回路のグラウンドを一点に集める事を言います。こうする事で、共通インピーダンスを抑える事が出来るためです。



共通インピーダンスというのは、上の図の様に回路をつないだ場合に発生する、二つの回路に共通するインピーダンス成分の事です。図の接続方法だと、二つの回路の線が有る点で合流してからグラウンドに流れ込んでいます。このように接続すると、例えば右の回路から電流が流れ込むとその電流によって共通部分に電圧が発生し、左の回路とグラウンドの間に電圧が発生します。つまり、右の回路によって左の回路の電位が変動する事になるのです。このように他の回路の電流によるノイズを避けるために一点アースが重要とされます。これに対して、全面アースと呼ばれるものも有ります。これは、基板の一面をすべてグラウンドにして、すべての回路を最短経路でそこと接続する方式です。この場合、グラウンドと接続される点が多くなり、多点アースになってしまいます。しかし、この場合



全面アース

グラウンドそのもののインピーダンスを低く抑える事が出来ます。そのため、共通インピーダンスが発生しても、抵抗値そのものが低いため、問題とならないです。

しかし、これら二つの方式は矛盾している様に見えます。全面アースと一点アースはそれぞれ利点を持ちます。一点アースは共通インピーダンスを抑えられる事。全面アースはインピーダンスそのものを低く抑える事が出来る事。これらの利点を、回路によつてどちらを取るかを選択する必要があります。

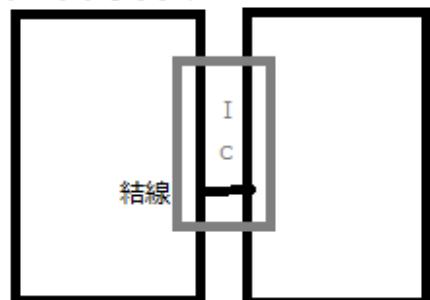
アナロググラウンドとデジタルグラウンド

これらの利点の使い分けとして、アナロググラウンドとデジタルグラウンドを分けるというものが有ります。アナログ IC はアナログ IC としてまとめ、広いグラウンドを取る。デジタルも同様にします。そしてそれぞれのグラウンドを有る一点でまとめます。この時、グラウンドは二つだけでなく、例えば大電流のアナログと小電流のアナログを分ける、ノイズ耐性の強いデジタル素子とノイズに弱い素子を分ける等、増やしてもかまいません。

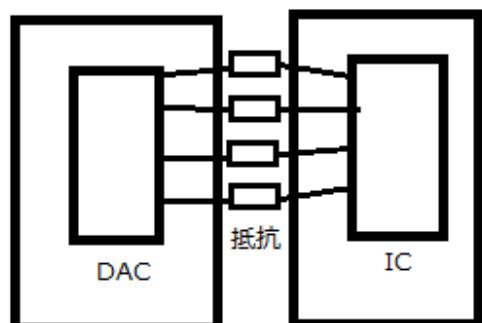
そして、線をまとめる点は電池や電源などのグラウンドのすぐ近くにしましょう。この様に、IC をある程度まとめ、それぞれのグラウンドを有る一点でまとめる事で、共通インピーダンスを抑えつつグラウンドを広く取る事が出来ます。

しかし、IC の中には DAC の様にアナログ回路とデジタル回路が含まれるものが有ります。これらの素子は、大抵アナロググラウンドとデジタルグラウンドが付いています。もしこの場合、図の様な接続をしない限り両方のグラウンドに接地する事ができません。このような場合は、素子をアナロググラウンドに接地しましょう。そして、デジタル IC と DAC の通信線にバッファを入れるなどしてインピーダンスを高くしましょう。こうする事で、DAC に流れ込む電流値が低くなり、DAC のデジタルグラウンドによるノイズを抑える事が出来ます。

アナロググラウンド デジタルグラウンド

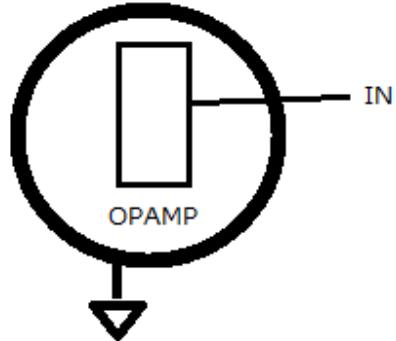


アナロググラウンド デジタルグラウンド



グラウンドループとシールド

オペアンプ等の入力を、周囲の雑音から守るためにシールドをする事が有ります。図の様にオペアンプの入力をグラウンドラインで囲み、電源等によるノイズを防ぐ事です。精密なオペアンプでは、回路の絶縁抵抗が不足する事があり、こうして保護する事が重要になります。しかし、この様にシールドを行うときにはシールドが吸収したノイズがすべて電流としてグラウンドに流れ込む事に注意をしなければなりません。特に、このようにグラウンドが一周すると、グラウンドループと呼ばれるものが発生します。有る点からグラウンドまでの経路が二つ以上有る時、その電位差が問題となるのです。また、このようにリングを作ると、その部分がリアクタンス成分を持つ事にもなります。そのほかにも、有るノイズを除去するときは、そのノイズは最終的にグラウンドに流れ込む事を忘れてはなりません。



結論

このように、グラウンドは電位の基準としてだけではなくいらない信号を捨てる場所としても利用されます。だからこそ、回路を作る時は常にどのように電流が流れるかを意識する必要があります。そして、大きなノイズが発生しそうな箇所、例えばモータードライバやLED ドライバ等の大電流が流れる部分はグラウンドを分離するなど注意する必要があります。ノイズの原因は常に電流です。電流が流れそうな部分のインピーダンスを低くして、共通インピーダンスが発生しないように気をつける事でグラウンドノイズを軽減する事が出来ます。今回の記事が、少しでも皆様のグラウンド設計に役立てば幸いです。

参考文献

「正しいグラウンドィングを守ること」

http://www.analog.com/jp/content/cu_ad4606jp/fca.html

岡村 延夫 (1990) 『定本 OP アンプ回路の設計』 CQ 出版

スピーチタイマー製作計画

知能機械工学科 2年 本多寿矢

始めに

我が大学では、某キャリア教育しかり研究発表しかり発表の場が数多く存在する。よほど徹底的に練習しない限り、残り時間が何分なのか非常に気になり不安になることだろう。そんなとき、残り時間が分からぬという悩みに答えるのがスピーチタイマーなのである。

作品コンセプト

マイコンで制御を行い、1分おきにヴァイブルーションを動作させることで経過時間を知らせる腕時計サイズの装置を制作する。

留意事項

- サイズをなるべく小さく。
- 1分→ツツ 2分→ツツツツ 3分→ツツツツツツ のように何分経過したのかを分かるようにする。
- 腕など目に見えないところに付けられるようゴムバンドでの固定ができるようにする。
- 服の上から操作できるように2,3ボタンの簡単操作。

終わりに

既に製品が存在するのかもしれない。AVRマイコンの使い方を色々と調べてしっかりと身につける。今まで企画したものが未だ実現できていないため、この作品はしっかり完成させるようにしたい。

最初の工研部報

1年 M 科 大川修平

1. 1年生前期の活動

自分は過去に電子工作をしたことが数回しかなく、技術はまだほとんどないので、今のところ工研内で作ったものといえば、新入生講習で教えてもらったマイコンライターぐらいです。ただ、技術を身に着けるために学内の電子工作的講座に行って、まだケースが完成していませんが、オーディオアンプを作成しました。外部電源方式のアンプです。この講座で主にパソコンを使った回路設計の方法、オシロスコープなどの計測機器の使い方、旋盤の注意点などを学習しました。

あと Raspberry Pi を購入しました。なにかといろいろと便利に使えそうなので。今のところ小さなモニタと 12V 出力対応のモバイルバッテリーや、小型キーボードなどを合わせて購入し楽に持ち運べるようにしたいと思っています。

2. エレコン

最初は個人でやろうと思ってましたが、他のサークルや学業との兼ね合いを考えた結果、他のチームに混ぜてもらうことにしました。エレコンに参加するにあたって、とにかく閃いたアイデアをどんどん貯めていたので、そのチームで作るものも自分のアイデアの中の一つに決定しました。そのせいというか、その流れでチームリーダーになってしまったのは

3. 今後の活動の展望

今後の活動は、エレコンの作業や他のサークル

の活動などを行っていくのはもちろんですが、ほかにも作りたいものがいっぱいあるし、エレコンのアイデアでボツになったものにもいいものがあるので、それらのものも細々と作っていきたいと思います。まあその前に今作ってるオーディオアンプを完成させることが先決ですが。

以下、湧き出たアイデアを箇条書きで列挙。

- ・超小型マウス
- ・トレース台
- ・モノクマ人形（ギミックを内蔵）
- ・一人用食洗器
- ・自動ベッドメイク装置
- ・本棚を整理する装置
- ・人形枕アラーム（膝枕）
- ・ロケット花火ランチャー
- ・個人用階段降機
- ・三人称視点カメラ
- ・多機能作業台（特に消しカスを何とかする）
- ・虫漬し（物理）
- ・エコー装置
- ・背後カメラ
- ・蚊の音のするアラーム
- ・拍手をすると時刻が音で分かる時計
- ・キャタピラ式移動装置
- ・集光機

自分で言うのもなんだが、はっきり言って 1 つでもできたらすごいと思う。

疑似科学を科学する ～イオン水についての考察と効果～

Mark.IU

世の中には科学的な根拠も何もなしに正しい、と信じられているものがあります。有名なもので言えば血液型占いや、ホメオパシーなどです。これらが所謂科学に見せかけた科学では無い「疑似科学」と呼ばれるものです。

今回は数年前にテレビ等で騒がれた「イオン水」について科学的に考察してみたいと思います。

1 イオンとは何か

「イオン水」について考える前にまずイオンとは何か、について考えなければなりません。化学をやっていない人の一般的な考えは「何か電気を帯びているもの」、だと思います。これは強ち間違っていないのですが、正しいとは言えません。

イオンとは原子や分子が電荷を持っている状態を示し、これらは Na^+ や CH_3COO^- のように書きます。また、イオンには陽イオンと陰イオンが存在し、陽イオンは正電荷を帯びていて陰イオンは負電荷をお帶びています。 Na^+ は陽イオンで CH_3COO^- が陰イオンです。

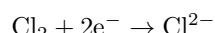
2 酸とアルカリ（塩基）

イオンについて説明したので次に酸とアルカリ（塩基）について説明しましょう。酸とアルカリ（塩基）には複数の定義、アレニウスの定義やブレンステッド・ローリーの定義等があるので、今回は一般的に用いられているルイスの定義で説明します。

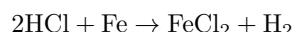
1. 酸とは

ルイスの定義では、酸とは電子を受け取る物質と定義されています。次のような

式（半反応式と呼びます）で表される物質が酸になります。



また、酸は主に金属などと反応し下式のように水素を発生させます。



また、酸がタンパク質と反応するとタンパク質が変性を起こし、様々な反応を示します。この「変性」は不可逆な反応なので、皮膚に触れた場合、触れた部分のタンパク質が火傷を起こしたような症状を示します。ただ、このような反応は塩酸のような強酸でしか起こりません。また、人の胃には rm HCl があり、胃酸と呼ばれています。 rm HCl は強酸なのに胃が痛まないのには理由があります。胃壁には粘膜がありその粘膜によって胃酸によって胃が荒らされるのを防いでいます。

2. アルカリ（塩基）とは

アルカリと塩基はほぼ同義なのでこれ以後は全て塩基に統一します。ルイスの定義では、塩基とは電子を与える物質と定義されています。次のような式で表される物質が塩基になります。



また、金属は基本的に塩基となります。NaOHのような強塩基はタンパク質を溶かす性質を持っています。なので指で強塩基に触れると指紋がとけてしまいます。さらに、目に入った場合などには失明する危険性もあります。ただ、このような反応はNaOHのような強塩基でしか起こりません。人の尿にもアンモニアNH₃という弱塩基が含まれています。

また、水溶液を電気分解することによって陰極に陽イオン（塩基）、陽極に陰イオン（酸）を集めることができます。

3 イオン水とは

Wikipediaによると、イオン水とは「飲用アルカリ性電解水の通称」だそうです。電解水とはイオン交換膜などを用いて水道水を電気分解した水で、弱塩基性の物質を混入してから電気分解する場合もあるそうです。また、Panasonicのホームページによると「ろ過した浄水を電気分解」と書いてあります。

つまり水道水（塩や塩基性物質を混ぜることもある）を電離させた物のようです。

4 イオン水についての科学的考察

弱塩基性であるイオン水が本当に体にいいのかどうかは専門家ではないのでわかりませんが、私はあまり体に良くないのではないかと思考します。理由としては、先に書いたように胃には胃酸があります。体にあまり良くない強酸がなぜ体内にあるかと言えば、胃の中にある消化酵素の為なのです。タンパク質等は強酸ではありません分解されません。そのため胃や腸の中には消化酵素があります。この消化酵素はまわりのpH（ペーハー）によって分解する速度が変わります。胃の中にある酵素はpHが低いとこ

ろで活発に反応します。なので胃酸によって胃の中のpHを低くしているのです。しかしそこに弱塩基性のイオン水が入ってしまうとpHが変わり、酵素の反応が鈍くなる可能性があります。なので体にあまり良くないのではないかと考えます。またアルカリイオン水の説明を読む限りだと、ただ水道水を電気分解していただけとわかります。日本の水道を流れている水は軟水であり、軟水はミネラル（塩基性物質）をあまり含んでいません。すると、電気分解しても陰極に集まつてくる陽イオンは非常に少ないと考えられます。なので体に影響を与えるだけのミネラルが集められるのか非常に疑問です。カルシウム等の塩基性物質を加えてから電気分解を行うのならば確かに何かしらの効果—それが良いか悪いかは置いておいて—is得られるかもしれません。

また、ただ設置するだけでイオン水ができると言う浄水器、これも正しいとは思えません。水道水に塩基性物質を加えた後に電気分解するタイプについてですが、買った当初は確かにアルカリイオン水ができるかもしれません。しかし、使い続けければいつかは必ず水道水に加える塩基性物質がなくなるはずです。しかしそのようなものを追加する必要がある、等とはどこにも書いてないのであります。そして水道水を電気分解するタイプは前述した通りです。

5 結論

私の結論としては「アルカリイオン水」による企業が謳っているような効果は望めない、でした。しかしこれはあくまであまりその分野について詳しくない私個人の考察なのでこれが正しい、というわけではありません。また日本には「信じるものは救われる」と言う言葉があるように信じることで心の安定を得てもいいのではないでしょうか。実際、人が信じれば影響がでるプラシーボ効果というものもあります。

私は信じませんが、信じる人は効果を信じても良いのではないでしょうか。

6 あとがき

科学的な考察をしてみると「言いながら肝心なところは「専門家ではないのでわかりません」とぶん投げてしまいすみません。しかしにわか知識で答えを決め付けるよりはいいのだと思います。もう一度書きますが「効果がない」というのはあくまで私個人の考えなので鵜呑みにしないでください。ただ私が伝えたいのは、偉い人が言っているから正しい、ではなく自分で考えて正しいかどうかを決める事は非常に重要なことです。

この記事が様々なことを自分で調べたり考えたりするきっかけになれば非常に嬉しく思います。最後に、このくだらない文章を読んでいただきありがとうございます。

7 参考資料

- wikipedia -アルカリイオン水-
<http://ja.wikipedia.org/wiki/%E3%82%A2%E3%83%AB%E3%82%AB%E3%83%AA%E3%82%A4%E3%82%AA%E3%83%B3%E6%B0%B4>
- Panasonic ホームページ -アルカリイオン整水器-
<http://panasonic.jp/alkaline/aboutalkaline/mechanism/index.html>

ドミネーター製作記 -Simple-

しゅんりゅ(回路・広報担当) & れいてんし(機構設計担当)

1. まえがき

この部報は、コミックマーケット 84 にて頒布した「ドミネーター製作記」の内容を一部抜粋・編集したものです。そのため、本誌をお持ちの方はあまり新鮮味がない記事となりますので、御了承下さい。

2. ドミネーターとは？

「PSYCHO-PASS サイコパス」という、2012 年 10 月から 2013 年 3 月に放送していた近未来 SF アニメに登場する特殊銃器。それが、「携帯型心理診断鎮圧執行システム“ドミネーター”」です。作中で法の適用・裁判・執行を一手に担い、作品のキャッチコピー「その銃口(システム)は、正義を支配する。」の元ともなっている、この作品を語る上で最初から最後まで外せない存在です。1 話からその特殊性や威力を存分に見せつけてくれたこの銃に、虜になつた人も多いことでしょう。そしてそれは、我々も同じでした。

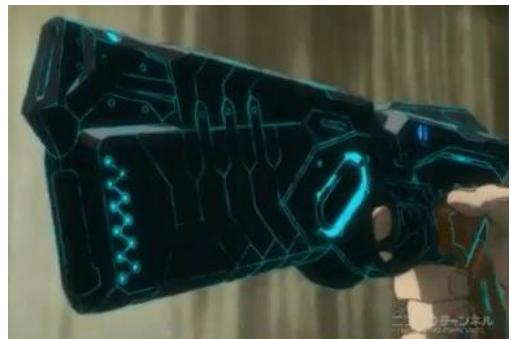


図 1. ドミネーター(アニメより)

ドミネーターのかっこいいところは、光り、喋るのも勿論のことなのですが、敵に合わせて銃自身が自動で変形すること、またその際の破壊力の高さにあります。さすがに破壊力は再現できそうにないですが、変形くらいは再現したい。そんな思いから、この製作企画が立ち上りました。

ドミネーターの変形形態は、麻痺銃「パラライザー」(基本形態)・殺人銃「エリミネーター」・分子分解銃「デコンポーザー」の 3 形態がありますが、このうち 3 形態目の「デコンポーザー」は現代技術の限界に相当する変形をしているため実現が厳しく、今回は 2 形態目との無差し替え変形を目標とします。

3. 電子回路部の設計 ~ 2013/02 月下旬

変形もさることながら、ドミネーターの有する機能は、できるだけ再現したい。ということで「人身感知、音声再生、発光ギミックの再現」を、内部に電子回路を組み込むことで実現しました。図 3 にブロック図を示します。基本的にシフトレジスタを使って LED 光らせたり、SD カードから mp3 データ読んで再生したりですが、一つ特殊なのは「人身感知」のセンサ。



図 2. 片方を隠した
焦電型赤外線センサ

これは私的に未開の地だったので、人身感知には焦電型赤外線センサを使うことにしました。人が通ると自動点灯する電灯などに使われている、人身感知センサです。原理としては、それぞれ見ていている範囲が異なる 2 対の赤外線センサがあり、その 2 つのセンサでの得られた温度差で人身を感知するそうです。センサの視野角を広くするために、通常はフレネルレンズと呼ばれる平面レンズと一緒に使うのですが、今回は向けた方向にしか反応してくれないと困るので視野角を広くとる必要はなく、使いませんでした。しかしこれだと問題があり、2 つのセンサが共に同じ対象を見てしまい、温度差が得られなくなってしまいました。これではセンサが反応しなくなるので、センサのうち一つを覆って見えなくして定値化し、それを基準とした温度センサ的な使い方をすることにしました。なんとか動きましたが、温度がある程度高ければ反応するので、ディスプレイなど温かいものに向けても反応してしまうのが玉に傷です。また検知距離も大したことがなく、だいたい 20cm がいいところです。が、パフォーマンスとしては十分だと判断しました。

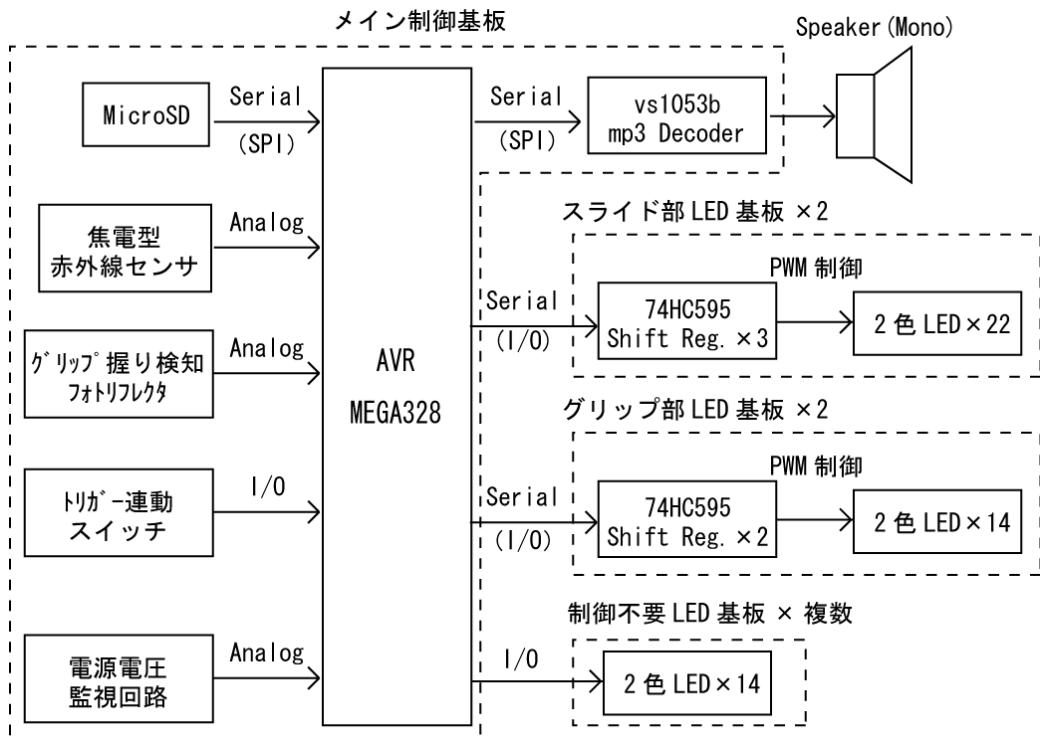


図3. 電子回路部 ブロック図

以上が、電子回路部のおおまかな紹介になります。これは大日本技研様のドミネーターキットに組み込めば、パラライザーモードの再現はできるので、製作しようと思う人は参考程度にどうぞ。

4. エリミネーターへの変形解説

基板設計と並行して、本体の具体的な設計にも手を入れ始めました。図4に、パラライザーからエリミネーターへ変形するときに、どのような事が起こるのかを書き込んだ図を示します。詳しくはサイコパス設定資料集をご覧下さい。おおまかになりますが、以下5つの手順でエリミネーターへの変形が完成します

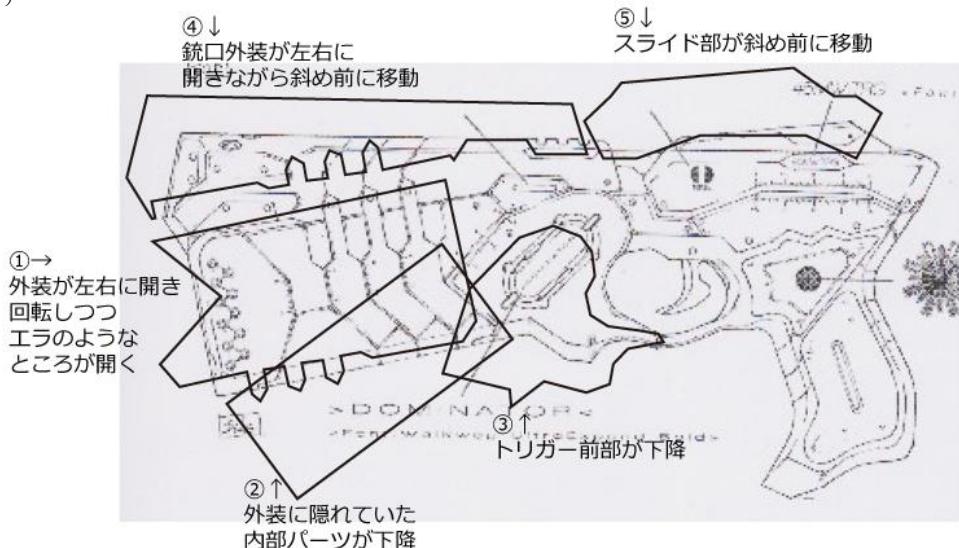


図4 エリミネーターへの変形

5. 試作版の完成と動画の投稿 ~ 2013/03 月下旬 - 2013/05 月中旬

そんなこんなで、動作確認用の試作版を3月末に完成させました。こちらについては、5月にニコニコ動画へ投稿しましたので、ご存知の方もいるのではないかと思います。片面しかない上、グリップは他のモデルガンの流用であったりするなど、あまり人に見せられたものではないのですが、我々はこういうことをやっているんだ、という思いが先走り、あのような動画を作りました。エゴの塊のような動画だったので不安でしたが、思いのほか反響がよくて、逆に焦りました。

6. 試作第二版の完成 ~ 2013/08 月上旬

試作版が完成してから約4か月、8/10に試作第二版が完成しました。試作版でお粗末だった部分を、大きく改善しました。なんとその翌日、C84で展示があったため、完成したのは本当にギリギリでした(汗)。機構設計のれいにてんくんに感謝。完成了第二版の画像を以下に紹介します。また、つい先日この試作第二版の紹介動画をニコニコ動画に投稿しました。まだ大きな反応がないため、今後どういった反応が得られるか、恐々と見ています(ピクンピクン……)。



試作第二版 ドミネーター仕様

【変形】パラライザー ⇄ エリミネーター

【電子機能】

- ・音声再生（随時）
(起動時、グリップを握った時、人に向けた時、トリガーを引いた時、etc)
- ・発光（2色）
適正ユーザー時：緑色、放射発光
不正ユーザー時：赤色。一切の電子機能が使用不可に。

紹介動画（ニコニコ動画）：[sm22149789](https://www.niconico.net/watch/sm22149789)

7. あとがき

この原稿ではさくっと進めましたが、ここまで道のりはとても険しいものでした。学業をおろそかにしてもいけないですし、限られた時間の中でここまで仕上げるに当たり、数々の挫折を味わいました。デコンポーザーへの変形を諦めたのも、そのうちの一つです。これについては、叶うならばやりたいと思っていますが、いかんせん現実味のない夢想になってしまっていて、具体案がないのが現状です。どうにかできないかなあ。

あ、そういえば、「PSYCHO-PASS サイコパス」二期と、劇場版が決定したそうで、すごく楽しみです。二期の主人公は、やっぱり一期の最後の娘なんですかね。劇場版は、ノベライズ版「名前のない怪物」の内容だとすごく嬉しいな。あ、もうページギリギリだ。ここまで読んで下さり、ありがとうございました！

ぴっこんなっ!の紹介とか

情報系男子院生 Hagi

URL: <http://www38.atwiki.jp/hagisoft/>

挨拶

こんにちは。情報系男子院生 Hagi と申します。せっかくなので工研部報に何か書こうということで、つい最近バージョンアップを行ったフリーソフト「ぴっこんなっ!」について、機能の紹介やプログラムについての話を書くことにしました。一部お見苦しい点があることをお許し下さい。

「ぴっこんなっ!」は、作成された経緯上使い方にかなり癖があるソフトです。実際に使ってみたい方は、<http://www38.atwiki.jp/hagisoft/pages/24.html> からダウンロード出来ます。よろしかったら使ってみてください。

1 背景

最近は様々な Web サービスで画像を投稿する機会も増えたり、また、様々なユーザーに画像を送る機会も増えたりしている。その際に、画像を縮小・他形式への変換を行う必要が出てくる場合がある。PC 上でこの作業を行う場合には、一般的に縮小ソフトやレタッチソフトで画像を縮小してから、その画像ファイルをアップロードする場合が多い。しかし、縮小するたびに保存するファイル名を指定しなければいけない場合や、アップロード時にアップロードするファイル名を指定するのに手間がかかる問題がある。また、アップロードしたファイルを手元に残す必要がない場合はそのファイルを削除するのに手間がかかったりする問題がある。

そこで、画像縮小・変換時にファイル保存やアップロードのファイル名指定の手間を省きつつ、また、使用後の画像ファイルの削除の手間を省く事もできる方法として、フリーソフト「ぴっこんなっ!」を作成し、初版を 2011 年に公開した。

2 提案手法

本ソフトを設計するにあたり、背景で述べた問題を解決すべく、以下の機能をもたらせた。

2.1 ドラッグ&ドロップだけで実行できる画像の拡大・縮小機能

図 1 のウインドウで、画像のサイズ・形式を指定する。

ウインドウに画像ファイルをドラッグすると、すぐに変換が始まる。変換された画像は、予め決まった場所へ自動的に保存される。



図 1: 「ピッコんなっ! Ver. 3.1」 画像を 1 枚変換した時の画面

2.2 変換したファイルのファイル名(フルパス)をクリップボードにコピー

図 1「最後に変換した画像のパス文字列」が有効な場合、フルパスがクリップボードにコピーされる。(複数を同時に変換した場合は最後に変換したファイルのフルパスがコピーされる。)

変換後に図 2 のようにブラウザ等でファイル選択画面のファイル名に変換後のパスを貼り付けると、変換したファイルをアップロード対象として直接指定することができる。

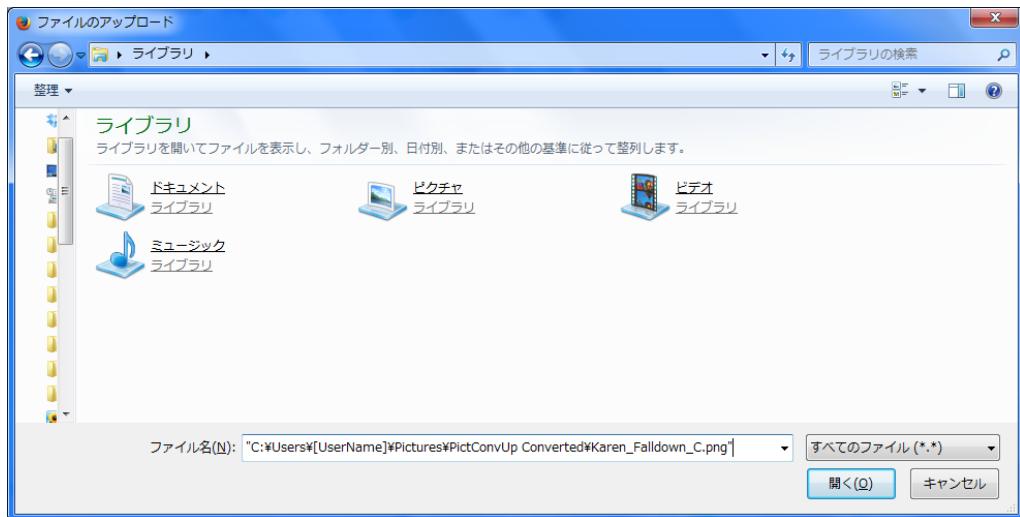


図 2: Firefox のファイルアップロード画面

2.3 変換した画像の自動削除

図 1「終了時に変換した画像を削除」が有効な場合、本アプリを終了時に変換後の画像ファイルを削除することができる。これによって、例えば、アップロードや他アプリへの取り込みなどが終わったあとに本アプリを終了すれば、変換後の画像を管理する必要がなくなる。

2.4 変換した画像をドラッグ&ドロップ

図1の変換した画像の一覧に表示されているアイテムをエクスプローラや他のアプリにドラッグ&ドロップすることにより、変換した画像を別の場所にコピーしたり、他のアプリで利用することができる。

3 設計

このソフトはC#で作成した。ソフトウェアの要となっている部分を幾つか紹介していく。

3.1 フォーム記述とドラッグ&ドロップ

このソフトでは、Windows Formsを利用してGUIを記述した。WPFも勉強してればよかった。
他のアプリ(エクスプローラとか)からドラッグ&ドロップを受け入れる場合は、メインフォームのクラスにDragEnterとDragDropイベントを設定する。[1]

また、変換した画像の一覧から画像をドラッグできるようにするには、ListViewクラスにItemDragイベントを設定し、その中でListView.DoDragDropメソッドを呼び出す事により、指定したアイテム(ここでは画像)を他のアプリにドラッグ&ドロップできる。[2]

ここで、以下にドラッグ&ドロップのコードの一部を示す。興味のある方は参考文献[1, 2]やその他WEB上の情報も参考にしつつ読んでみてほしい。一部メンバやクラスの命名がおかしくなってる部分があるのはご容赦願いたい。

```
public partial class MainForm : Form
{
    /*中略*/
    // ListViewからドラッグされたらフラグを立てる
    // ListViewからドラッグされたアイテムをこのフォーム自身にドロップできないようにする
    bool thisDraging = false;

    // ドラッグされたデータのチェック
    private void MainForm_DragEnter(object sender, DragEventArgs e)
    {
        // ドラッグデータとフラグの確認
        if (e.Data.GetDataPresent(DataFormats.FileDrop) && !thisDraging)
            e.Effect = DragDropEffects.Copy; // 正しいデータならコピーとして受け入れる。
        else
            e.Effect = DragDropEffects.None; // それ以外は受け入れない。
    }

    // ドラッグされたアイテムがドロップされた時の動作
    private void MainForm_DragDrop(object sender, DragEventArgs e)
    {
        // ドロップされたアイテムのファイル名を取得
        string[] fileName =
            (string[])e.Data.GetData(DataFormats.FileDrop, false);
        this.Activate();
    }
}
```

```

//入力されたファイルの変換を別スレッドで行う。
//メインスレッドの処理がこのメソッドを抜け出さないとエクスプローラが固まるため、
//時間がかかる操作はスレッドを分けている
BeginInvoke((MethodInvoker)delegate()
{
    //変換を行うメソッド
    ConvertPict(fileName, 0, fileName.Length);

});
}

//変換された画像の一覧から画像を変換した時の動作を記述
private void listViewConvertedHistory_ItemDrag
    (object sender, ItemDragEventArgs e)
{
    //変換するファイルのリスト
    System.Collections.Generic.List<string> col =
        new System.Collections.Generic.List<string>();

    //選択されたアイテムをドラッグするリストに登録
    foreach (ListViewItem selected in listViewConvertedHistory.SelectedItems)
    {
        //変換された画像の情報を listViewConvertedHistory に登録する際、
        // ListViewItem.Tag プロパティに、別途以下に定義した ConvertedImageInfo オブジェクトを登録、
        //ConvertedImageInfo.ConvertedPath に変換された画像のフルパスを書き込んでおく
        ConvertedImageInfo info = selected.Tag as ConvertedImageInfo;
        if (info != null)
        {
            col.Add(info.ConvertedPath);
        }
    }
    //実際にドラッグするデータの準備
    DataObject dobj = new DataObject(DataFormats.FileDrop,col.ToArray());

    //ドラッグドロップのアクションを定義
    //DragDropEffects.Copy を設定し、エクスプローラにアイテムをドラッグするとコピーされる
    //DragDropEffects.Move を設定してしまうと、ファイルが移動してしまうので設定しない
    //DragDropEffects.Link は、iTunes がドラッグを受け入れるのに必要
    DragDropEffects effect = DragDropEffects.Copy | DragDropEffects.Link;

    //フラグを立てることで、自分自身からドラッグされたアイテムを
    //自分自身が受け入れないようにする
    thisDraging = true;
}

```

```

//ドラッグドロップの実行
listViewConvertedHistory.DoDragDrop(dobj, effect);

}

//このアプリからドラッグしたアイテムをエクスプローラにドロップした時のイベント
private void listViewConvertedHistory_QueryContinueDrag
(object sender, QueryContinueDragEventArgs e)
{
    //ドラッグが完了したら
    if (e.Action != DragAction.Continue) {
        //フラグを降ろす
        thisDragging = false;
    }
}

//listViewConvertedHistory の定義とイベントハンドラの登録
//(通常は VisualStudio が自動的にやってくれる)
private System.Windows.Forms.ListView listViewConvertedHistory;
this.listViewConvertedHistory.ItemDrag +=
    this.listViewConvertedHistory_ItemDrag;
this.DragDrop += this.MainForm_DragDrop;
this.DragEnter += this.MainForm_DragEnter;
this.listViewConvertedHistory.QueryContinueDrag +=
    this.listViewConvertedHistory_QueryContinueDrag;

}

//変換された画像の情報
class ConvertedImageInfo : IDisposable {
    /*中略*/
    //変換後の画像のパス
    public string ConvertedPath
    {
        private set;
        get;
    }
}

```

3.2 画像の変換

.NET Framework では、簡単な画像の処理ができるライブラリが標準で含まれている。このアプリでは、.NET で使用出来る Image オブジェクトやそれを継承した Bitmap オブジェクトで描画キャンバスを定義し、その上で Graphics オブジェクトを用いて縮小後の画像を描画している。縮小された画像は、Image.Save メソッドで、PNG、JPG、BMP 形式などで保存することができる。

なお、画像を DrawImage で縮小描画する際、そのまま描画させると変換モードによっては縮小後に

画像の外周が乱れてしまう。それを防ぐために、図3のように、大きめの描画キャンバス(一時キャンバス)の中央に一旦画像を描画し、その周囲を画像の外周部と同じ色で塗っておき、縮小時に、大きめの描画キャンバスに描いた中央の元画像の部分を縮小後のキャンバスに描画するという手法をとっている。



図3: 一時キャンバスの描画

```
//インスタンスは作らない すべて static メソッド
static class PictureShrinker {
    /*中略*/

    //指定したサイズに画像を縮小(拡大縮小方法も指定)
    //拡大縮小時は、まずこのメソッドを呼び出す
    //返り値の Bitmap(Image クラスを継承) は呼び出し元で保存
    public static Bitmap Convert(Image toConvert, Size outputSize,
        System.Drawing.Drawing2D.InterpolationMode interpolation)
    {
        if (toConvert == null)
        {
            throw new ArgumentNullException("toConvert");
        }
        Bitmap newImg =
            new Bitmap(outputSize.Width, outputSize.Height, toConvert.PixelFormat);
        try
        { //縮小後の画像の描画キャンバスを用意
            DrawToNewImage(toConvert, newImg, interpolation);
        }
        catch (Exception)
        {
            if (newImg != null)
            {
                newImg.Dispose();
            }
        }
    }
}
```

```

//PixelFormat を 24bit にしてやり直し
newImg = new Bitmap
    (outputSize.Width, outputSize.Height,
     System.Drawing.Imaging.PixelFormat.Format24bppRgb);
//縮小後の画像を描画する
DrawToNewImage(toConvert, newImg, interpolation);
}
return newImg;
}

//newImg に toConvert の縮小イメージを描画させる
private static void DrawToNewImage
(Image toConvert, Bitmap newImg,
System.Drawing.Drawing2D.InterpolationMode interpolation)
{
    //一時キャンバスをもとの画像よりどれだけ大きくするか
    //縦クッションサイズ
    const int cussionSizeV = 100;
    //横クッションサイズ
    const int cussionSizeH = 100;
    using (Bitmap cussionBmp =
        new Bitmap(
            toConvert.Width + cussionSizeH*2, toConvert.Height + cussionSizeV*2)
    )
    {
        using (Graphics g = Graphics.FromImage(cussionBmp))
        {
            g.PixelOffsetMode = System.Drawing.Drawing2D.PixelOffsetMode.Half;
            g.CompositingMode =
                System.Drawing.Drawing2D.CompositingMode.SourceCopy;
            g.InterpolationMode =
                System.Drawing.Drawing2D.InterpolationMode.NearestNeighbor;

            //画像本体
            g.DrawImage(toConvert, new Rectangle(
                cussionSizeH, cussionSizeV, toConvert.Width, toConvert.Height));

            //以下で、画像の周囲を元画像の縁と同じ色で描画しておく
            //上
            g.DrawImage(toConvert,
                new Rectangle(cussionSizeH, 0, toConvert.Width, cussionSizeV),
                new Rectangle(0, 0, toConvert.Width, 1), GraphicsUnit.Pixel);
            //下
            g.DrawImage(toConvert, new Rectangle
                (cussionSizeH, cussionBmp.Height - cussionSizeV,
                 toConvert.Width, cussionSizeV),

```

```

        new Rectangle(
            0, toConvert.Height - 1, toConvert.Width, 1), GraphicsUnit.Pixel);
//左
g.DrawImage(toConvert,
new Rectangle(0, cussionSizeV, cussionSizeH, toConvert.Height),
new Rectangle(0, 0, 1, toConvert.Height), GraphicsUnit.Pixel);
//右
g.DrawImage(
toConvert, new Rectangle(
cussionBmp.Width - cussionSizeH, cussionSizeV,
cussionSizeH, toConvert.Height),
new Rectangle(toConvert.Width - 1, 0, 1, toConvert.Height),
GraphicsUnit.Pixel);
//左上
g.DrawImage(toConvert, new Rectangle(0, 0, cussionSizeH, cussionSizeV),
new Rectangle(0, 0, 1, 1), GraphicsUnit.Pixel);
//右上
g.DrawImage(toConvert, new Rectangle(
cussionBmp.Width - cussionSizeH, 0, cussionSizeH, cussionSizeV),
new Rectangle(toConvert.Width - 1, 0, 1, 1), GraphicsUnit.Pixel);
//左下
g.DrawImage(toConvert, new Rectangle(
0, cussionBmp.Height - cussionSizeV, cussionSizeH, cussionSizeV),
new Rectangle(0, toConvert.Height - 1, 1, 1), GraphicsUnit.Pixel);
//右下
g.DrawImage(toConvert, new Rectangle(
cussionBmp.Width - cussionSizeH, cussionBmp.Height - cussionSizeV,
cussionSizeH, cussionSizeV),
new Rectangle(toConvert.Width - 1, toConvert.Height - 1, 1, 1),
GraphicsUnit.Pixel);

}

#if DEBUG
cussionBmp.Save("DebugCussionBmp.png");
#endif

//縮小画像の描画
using (Graphics g = Graphics.FromImage(newImg))
{
    g.CompositingMode =
        System.Drawing.Drawing2D.CompositingMode.SourceCopy;
    g.InterpolationMode = interpolation;
    g.PixelOffsetMode = System.Drawing.Drawing2D.PixelOffsetMode.Half;
    g.DrawImage(cussionBmp,
    //一時キャンバスの元画像の部分を縮小描画する
    new Rectangle(0, 0, newImg.Size.Width, newImg.Size.Height),
    new Rectangle(

```

```
        cussionSizeH, cussionSizeV, toConvert.Width, toConvert.Height),  
        GraphicsUnit.Pixel);  
  
    }  
}  
}  
}  
}
```

4 まとめと展望

今回は、「ぴっこんなっ!」の紹介や設計について簡単な説明を行った。
インデックスカラー(256色)への変換対応や透過GIFや透過PNGの処理(透過部が黒になる)の改善・その他の形式への変換は今後の課題である。

参考文献

- [1] Drag&Dropを行う: .NET Tips: C#, VB.NET,
<http://dobon.net/vb/dotnet/control/draganddrop.html>
- [2] ファイルやディレクトリをエクスプローラへドラッグ&ドロップするには? - @ IT:
<http://www.atmarkit.co.jp/fdotnet/dotnettips/384expdragdrop/expdragdrop.html>

国立大学法人 電気通信大学
工学研究部 部報 第45号

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

発行 皆川 太志
編集人 住澤 庄秀
表紙 河村 大輝
発行 2013年10月31日
執筆 工学研究部 部員