工研部報

４６号

まえがき

工学研究部（略称：工研）というサークルです。
部室はサークル棟 2 階にある警告色のドアの部屋です。

工研では、いろいろなことをやっています。
マイコン
電子工作
機械工作
電気自動車
プログラミング
などなど……

どれか一つでも興味があれば、是非部室へ来てみて下さい。

経験のない人でも大丈夫。知識ある先輩方が教えてくれます！
入部前はハンダ付けをしたことないって部員だっています。

高専から来た人も大歓迎です。
持ってる知識・経験を存分に生かして下さい。
部員にも高専からの編入生がいます。

毎年工研では、新入生向けに講習会を開催しています。
何も知らない人向けの講習会なので、どうぞ参加してみて下さい。
内容としては、プログラミングの基礎や電子工作の基礎などをやる予定です。
詳しくは部室で部員に聞いてみて下さい。

工研では、各々が好きなことをしています。
やりたいことを、やりたときに、やりたい分だけやる。
そのための施設はそろっていますし、教えてくれる先輩もいます。

（去年は PSYCHO-PASS のドミネーターを作った部員もいます。）

また、定期的に合宿やイベント等もあります（参加は任意）

繰り返しになりますが、こんな工研に興味があれば是非部室までいらして下さい。
昼休みや放課後は部員がいると思います。

以下詳細

正式名称	: 工学研究部
部室	: サークル棟 2 階向かって左　警告色のドアが目印
部員	: 約 30 人。学部 1 年から院 2 年まで
部会	: 毎週 月、金 18 時から（どちらかに参加すれば OK）
講習会	: 部会の後

昨年度やったこと：エコラン（電気自動車）、サーバー増築、マイコン、回路設計・基盤加工、箱根合宿、アンプ作成、エレクトロニクスコンテスト参加、アーケードコントローラー作成、などなど

工学研究部　部報　第46号

もくじ

作者	題名	ページ
大川	AVR マイコン簡易まとめ	4
皆川 太志	トランジスタを用いた増幅回路の設計	7
池田 佳那	初心者向け CSS 解説	10
木村 敢	タッチスクリーンを使ってみた	12
住澤 柾秀	鍋でお湯沸かした時の蒸発量調べてみた	14
村上 大和	BeagleBone Black を用いた SoftEther VPN Server の構築（準備編）	16
横田 嶺	ブラシレス DC モーターのベクトル制御	18

AVR マイコン簡易まとめ

UEC 工学研究部 部報 46 号
知能機械工学科　１年 大川

自分は C 言語を使った Windows 上でのプログラムしか作ったことが無かったので、マイコン上で動作するプログラムについてはよく知りませんでした。だから同じような立場の人がわかるように、それについて書いておこうと思います。

まずマイコンのプログラムを書く上で必ず考慮に入れておかなければならないのが、使うマイコンの仕様です。マイコンの性能と機能を理解したうえでプログラムを書くと、そのマイコンを十分に利用してプログラムを書くことが出来、可能な動作の幅が広がります。AVR マイコンは普通のマイコンについている入出力の他に、「割り込み処理」や「タイマー」などのさまざまな機能が内蔵されています。特に「割り込み処理」はパソコン上で動作するプログラムには見られない特殊なもので、非常に利用価値があります。

１．タイマー

マイコンで使っているタイマーというのは設定した周波数に依存していて、処理回数をカウントしています。カウントに使われているレジスタはマイコンの種類によって数や特性が変わってきます（詳しくはデータシート)。カウントするタイミングは何種類か設定でき、n 周期ごとに 1 カウント、という風にできます（n 分周と言います)。カウントがオーバーフローすると 0 に戻ります。このカウンタを利用すると PWM(pulse width modulation)などが出来るようになります。PWM はマイコンの基本動作と言ってもいいくらいよく使うので習得しておくべきです。他にも使い方はいろいろとありますが、最低でも使用するカウンタの初期設定はする必要があります。

２．A/D 変換

A/D 変換とはアナログ情報をデジタル情報に変換する動作で、ADC(Analog to Degital Converter)とも書きます。センサーなどの電圧の変化をマイコン上で扱えるように数値に変換するのもこの処理をしなければなりません。これで注意することは、回路の組み方です。プログラムに関しては必要なレジスタを初期化して、変換の結果の入ったレジスタを利用するだけなので大して難しくはありません。AVR マイコンの A/D 変換で扱えるアナログ情報は電圧だけです。対象のピンの電圧を V_{PCn} とし、基準電圧を V_{AVCC} とすると、変換結果の数値は(V_{PCn}/V_{AVCC})に比例します。回路は、 VCC と AVCC をつなげ、２つある GND をつなぎ、調べたい電圧を A/D 変換用のピン（入力）につなぎます。また、必要に応じて RESET を基準とすることもできます。

３．割り込み処理

次に割り込み処理についてですが、マイコンはそれが動作する電子回路と密接に関係しているので、電子回路上で起きたマイコンとは関係のない変化にも反応して「処理」を「割り込ませる」ことができます。これを使うと、例えばある処理を実行している最中に A/D 変換が終了したとき、特定の処理を割り込ませたりすることができます。詳しいことはネット上で調べればすぐにわかることですが、簡単に書き方だけは書いておきます。

```
#include <avr/interrupt.h>

ISR ( /*監視するベクタ名*/ , /*割り込み属性*/ ){
        /*処理内容（返り値なし）*/
}
```

これを書いてから、 sei(); で割り込みを許可し、cli(); で割り込みを禁止します（これらの関数は片方を一度動作させると、もう一方で状態を変えない限り動作し続けます。）監視するベクタ名や割り込み属性については interrupt.h やデータシートに載ってます。ここで注意しておくべきことは、条件次第ではメインの処理ができなくなる可能性を考慮に入れなければならないということです。例えばタイマー比較一致で、一致したときに処理をするようにしたとき、一定時間ごとに割り込みが発生するのですが、その間のメイン処理で A/D 変換などの処理に時間のかかる動作をしようとすると、設定次第では正常に A/D 変換できなかったりします。他にも、割り込みの間隔が短すぎる場合は全体的な動作速度が遅くなったりもします。

４．UART

UART は Universal Asynchronous Receiver Transmitter の略で、USART とも書きます。シリアル通信の一種で、マイコンとパソコンの通信を可能とします。これを利用すると現在マイコンがどんな動作をしているのかをパソコンに転送できるので、デバッグに使うことが出来たりします。最近のパソコンにはシリアル通信ようの端子が付いていないものが多いので、大抵の場合シリアル変換モジュール（FT232R)を使います。これは USB をシリアル通信に変換するものです。UART では送信（TXD）・受信（RXD）・GND を回路でつかい、マイコン側の TXD（RXD）と RXD（TXD）を繋ぎます。初期設定として通信速度とデータビット数とパリティビットとストップビット数を設定します。また、必要に応じてマイコンの周波数（システムクロック）も設定します。データビット数とパリティビットとストップビット数は、通常の通信環境の場合は初期化する必要はありません。通信速度はマクロ定義で

```
#define BAUD 9600    //9600bps
```

のように設定でき、システムクロックも同様に

```
#define SYSCLK 20000000    //20MHz
```

のように設定できます。通信速度は、通信する際にパソコン上で起動するターミナルで設定する通信速度と同じにする必要があります。同じ設定にすることで同期させようというわけです。ターミナルは Windows なら「Tera Term」がいいと思います。通信して、データのやり取りをするためには基本的には割り込み処理を使います。プログラムを書くのは結構面倒なので、「ELM」という Web サイトにある UART 用のソースコードを使用することをお薦めします。

＜参考＞
UART 処理【http://okgnz.web.fc2.com/avrd/avrd07.htm】
電子工作の基礎知識＞通信規格＞UART その１基本【http://www.ele-lab.com/tips_uart.html】
ELM【http://elm-chan.org/index_j.html】
MONOist【http://monoist.atmarkit.co.jp/】

おまけ（スペースが余ったので。メモみたいなもんです。）

H8 マイコンを使うのに必要なもの
・本体（マイコンボード）
・シリアル通信環境
・ADM3202 とかいう IC

圧着端子を付けるとき気を付けること
・端子とコネクタのメーカーが同じで、対応していること。
・端子の種類によって使う工具も変わってくる。
・コードをはがすのは 5mm ぐらいがちょうどいい。長くても短くてもダメ。
・最終的にコードの長さがそろうようにする。不揃いだとコネクタに入れにくい。

ゼルダの伝説時のオカリナでのスタルフォスの攻略（神縛り）
・出現時にスタルフォルに急接近するとバックステップ後、ジャンプ切りをしてくる。
・スタルフォスのジャンプ切りを回転アタックでくぐると、背中ががら空きになり、攻撃するとスタルフォスがバックステップし、パターンに入る。ただし、バックステップした後のスタルフォスの位置がリンクと近すぎるとジャンプ切りをしてこない。
・リンクを中心に円を書くようにして徐々に接近してくる。
・たまに一直線に近づいてくる。一定時間追尾してきて、スタルフォスの射程距離に入ると確実に攻撃してくる。
・切り下げより切り上げのほうが、攻撃範囲が広い。
・攻撃は１回だけのときと２回のときがある。２回攻撃の後はスキが大きい。
・スタルフォスの動きが止まった時、相対したリンクから見て左側（スタルフォスの右足側）に回転アタックすると、スタルフォスの右後ろに回り込めることがある。攻撃すると、バックステップしてジャンプ切りしてくる。
・２体１のときは積極的に攻撃してくるが、１対１の時は慎重になる。
・体力は、折れたゴロンのナイフで１０回分。マスターソードなら５回。
・２体の距離が近い時に戦うとほぼ確実に攻撃をくらってしまう。

トランジスタを用いた増幅回路の設計

先進理工学科 3 年　皆川太志

1　はじめに

アナログ回路って難しいですよね、設計が。私も未だによくわからないです。というわけで入門的な感じで、今回は電流帰還バイアス増幅回路をトランジスタを用いて設計してみようと思います（唐突）。細かいところがかなり大雑把なうえ、私も素人なので、鵜呑みにはしないようにお願いします。

2　電流帰還バイアス増幅回路って？

電流帰還バイアス増幅回路ってのは図 1 みたいなやつです。入力 v_i に対して出力 v_o は反転増幅されます。この回路の特徴は、もし I_C が増加しようとすると、R_E による電圧降下 V_E が増加するため、ベースエミッタ間電圧 V_{BE} が減少し I_B も減少するということです。すなわち、R_1, R_2 によるバイアス電圧が、出力によって常に安定方向にフィードバック制御されるのです。これをバイアス電圧（ベース直流電圧）の安定化といいます。R_E と並行の C_E は、この出力安定（抑制）機能が交流（信号）について行われないように交流だけ通過させるものです。

専門的にはもっと色々あるんですが、基礎的にはこんなもので済まして早速設計してみようと思います。興味があれば専門書で調べてみて下さい。

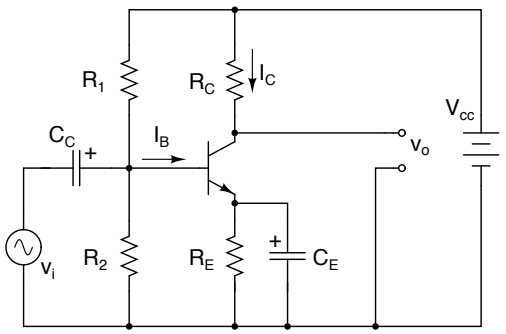


図 1: 電流帰還バイアス増幅回路

3　回路の設計仕様

まずは、仕様を決めましょう。今回は表 1 の仕様で設計してみたいと思います。設計目標を立てたら、それを余裕を持ってクリアできるように設計しておくといいです。

トランジスタとして 2SC1815Y を選んだのは、トランジスタの増幅率を表す h_{fe} が設計目標 160 より少しい大きいくらいだからです。入手性もいいですし。仕様に合うトランジスタを選びましょう。電源電圧 V_{CC} は実験で便利な電池で使えるように 1.5V の倍数で設計します。今回は 12V としました。出力は電源電圧 V_{CC} 以下にしましょう。周波数は増幅したい信号の周波数に合わせて作りましょう。

表 1: 設計仕様

値	設計目標	設計値
増幅率 $\lvert A_v \rvert$	160	2SC1815Y
出力 v_o	$5\mathrm{V_{pp}}$	$10\mathrm{V_{pp}}$
対応周波数 f	1kHz 以上	1Hz 以上
電源電圧 V_{CC}	1.5V の整数倍	12V

4　回路の設計

仕様が決まったら設計していきます。仕様から、オームの法則やらテブナンの定理やら先人の築いた経験則を使って素子の値などを求めていくのですが、その辺は大胆にカットして（専門書でも読んで下さい）、と

にかく設計できるように手順ごとに分けて書いていきます。

手順 1　電源電圧 V_{CC} の決定

電源電圧 V_{CC} は出力 v_o の pp 値に 2 倍の余裕をもって設計するといいです。本当は $V_{CC} = 1.1v_o$ となるようにすれば動くのですが、それでは波形が歪むので余裕を持って 2 倍に設定します。

手順 2　R_C の決定

トランジスタの $h_{ie}.h_{fe}$ と A_V から、R_C を式 1 のようにして決めます。$h_{ie}.h_{fe}$ は 2SC1815Y のデータシートの値か実測して使いましょう。ここでは、たまたまこないだ実測した値で計算してみます。

$$\begin{aligned} R_C &= |A_V| \cdot h_{ie}/h_{fe} = 160 \times 927.0/198.1 \\ &= 748.7\Omega \end{aligned} \tag{1}$$

このように求まりますが、ピッタシこの値の抵抗なんて存在しません。そこで近い値の抵抗をあてがいましょう。大丈夫、余裕を持って設計したんですから。

そんなこんなで $R_C = 748.7\Omega \rightarrow 680\Omega$ と値が決まりました。

手順 3　動作点 $\mathbf{Q}$（V_{CEQ}, I_{CQ}）を求める

動作点というのは、データシート等に書いてある交流負荷線というものの中点のことです。これは初心者には難しいものなんですが、ようは増幅動作をするときに出力が電源電圧より大きくなったり、GND 電位より低くなったりしないように、始点をなるべく真ん中にするということです。

測定で導くことも出来ますが、めんどくさいので適当に求めます（笑）。素子によるばらつきがありますが、経験的に 2SC1815Y は $I_B = 30\mu$A くらいになるといい感じに動作するんです。このときのコレクタ電流は

$$\begin{aligned} I_{CEQ} &= h_{fe} I_B = 198.1 \times 30\mu\text{A} \\ &= 6.589\text{mA} \end{aligned} \tag{2}$$

と求めることができます。このときデータシートから $V_{CEQ} = 5.583$V くらいであることが読み取れます。

これより動作点 Q は（$V_{CEQ} = 5.583$V, $I_{CEQ} = 6.589$mA）であることがわかりました。

手順 4　R_E の決定

センスある皆さんならもう気付いているかもしれないですが、ここまでで求めた $R_E, R_C, I_{CEQ}, V_{CEQ}$ から、R_E は簡単に求められますよね？I_{CEQ} は CE 間を流れているわけなので、オームの法則で次の式から求められます。

$$V_{CC} = I_{CEQ}(R_C + R_E) + V_{CEQ} \tag{3}$$

これを変形して R_E について解くと

$$\begin{aligned} R_E &= \frac{V_{CC} - V_{CEQ}}{I_{CEQ}} - R_C \\ &= \frac{12 - 5.583}{6.589 \times 10^{-3}} - 680 \\ &= 293.9\Omega \end{aligned} \tag{4}$$

これもピッタリの抵抗はないので近いものをあてがいます。$R_E = 293.9\Omega \rightarrow 270\Omega$ としました。なぜ、300Ω にしなかったのかというと、さっき R_C を実際より小さくしたことを考え、R_E と R_C に分圧される出力電圧 v_o がなるべく元の値からずれないようにするためです。

手順 5　バイアス回路の設計

続いてバイアス回路を設計します。まず、$R_B = R_1 // R_2$（R_1, R_2を 並列に接続した抵抗の合成値の意）とおきます。するとテブナンの定理でバイアス回路を等価した、以下の式から R_1 と R_2 を決定することが出来ます。

$$R_B = R_E h_{fe} 0.1 \tag{5}$$

$$V_{BB} = I_{CQ} R_E + 0.7\text{V} \tag{6}$$

$$R_1 = R_B/(V_{BB}/V_{CC}) \tag{7}$$

$$R_2 = R_B/(1 - V_{BB}/V_{CC}) \tag{8}$$

V_{BB} はテブナンの定理で等価されたバイアス電圧のことですが、よくわからずとも方程式は解けます。ま

た、等価回路を説明するとややこしくなりそうなので説明は省きますね。とにかく方程式を解いて下さい。

この 4 つの方程式に $V_{CC} = 12\text{V}$, $V_{CEQ} = 5.583\text{V}$, $I_{CEQ} = 6.589\text{mA}$, $R_E = 270\Omega$ を代入して R_1 と R_2 について解くと $R_1 = 26.37\text{k}\Omega$, $R_2 = 7.655\text{k}\Omega$ となります。値の近い抵抗をあてがうと、$R_1 = 27\text{k}\Omega$, $R_2 = 8.2\text{k}\Omega$ と決定することができます。

手順 6　C_E の決定

前述したように、キャパシタ C_E は「出力安定（抑制）機能が交流（信号）について行われないように交流だけ通過させる」ためについています。そのため C_E の決定は増幅回路の周波数特性の決定にほかならないというわけです。

通過させたい最低周波数を f_{min} とすると以下の条件を満たす C_E を選べばいいです。

$$\begin{aligned} C_E &\geq 1/(2\pi f_{min} R_C) = 1/(2\pi \cdot 1 \cdot 680) \\ &= 234.1\mu\text{F} \end{aligned} \tag{9}$$

もちろんピッタシのキャパシタなど無いので余裕を持って $C_E = 234.1\mu\text{F} \rightarrow 470\mu\text{F}$ とします。

4.1　手順 7　C_C の決定

まず入力インピーダンス $R_{in} = h_{ie}//R_1//R_2$ を計算します。

$$\begin{aligned} R_{in} &= \frac{h_{ie} \cdot R_1 \cdot R_2}{h_{ie} \cdot (R_1 + R_2) + R_1 \cdot R_2} \\ &= \frac{927.0 \times (27 \times 10^3) \times (8.2 \times 10^3)}{927.0 \times (27 \times 10^3 + 8.2 \times 10^3)} \\ &\quad \cdot \frac{1}{+(27 \times 10^3) \times (8.2 \times 10^3)} \\ &= 807.8\Omega \end{aligned} \tag{10}$$

入力インピーダンス R_{in} より C_C のインピーダンスが十分に小さいように設計すればいいので、次のように C_C を求めます。

$$\begin{aligned} C_C &\geq 1/(2\pi f_{min} R_{in}) = 1/(2\pi \times 1 \times 807.8) \\ &= 197.0\mu\text{F} \end{aligned} \tag{11}$$

これに合うキャパシタを選ぶと $C_C = 197.0\mu\text{F} \rightarrow 220\mu\text{F}$ と決定することができます。

5　完成した回路

ここまでで図 1 に示した電流帰還バイアス増幅回路の全てのパラメーターの値を決定することが出来ました。使用する素子についてまとめると表 2 のような感じです。これを基板に半田付けするなり、ブレッドボードで組めば動くはずですね。

表 2: 回路素子

抵抗	計算値	使用値
R_C/Ω	748.7	680
R_E/Ω	293.9	270
$R_1/\text{k}\Omega$	26.37	27
$R_2/\text{k}\Omega$	7.655	8.2
$C_E/\mu\text{F}$	234.1	470
$C_C/\mu\text{F}$	197.0	220

6　おまけ

オシロとファンクションジェネレータを使って増幅率 $|A_V|$ の周波数特性を図 2 にプロットしてみました。$f = 1\text{kHz}$ で $|A_V| = 140$ くらいですかね？仕様の 160 倍を満たせずちょっと残念でしたが、まあ満足です。

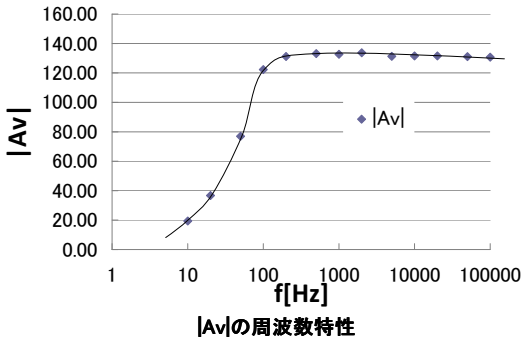


図 2: 増幅率 $|A_V|$ の周波数特性

参考文献

[1] CQ 出版,『定本 トランジスタ回路の設計』, 1991/12

初心者向け CSS 解説

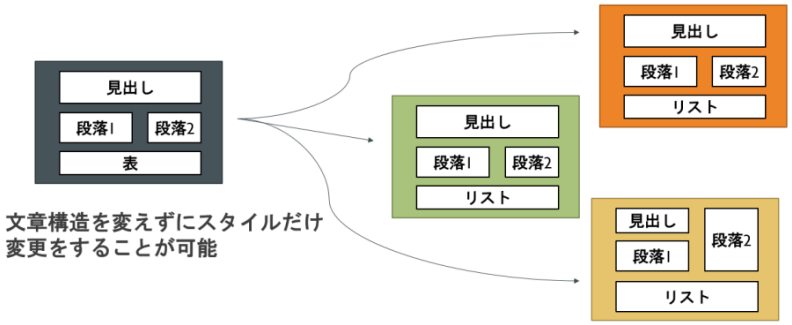
電気通信大学　工学研究部部報　46 号
知能機械工学科 1 年　池田佳那

１．はじめに

CSS とは、Cascading Style Sheets（カスケーディング・スタイル・シート）のことであり、HTML や XML などのスタイルをどのように修飾するかを指定するスタイルシートのことである。少し CSS に興味があるが手を出したことはないという人のために、軽く解説をしていこうと思う。

２．CSS の特徴

CSS はそもそも、HTML の文章構造とスタイルを分けるためにできたものである。HTML 文章にデザインまで全て記述してしまうと、デザインを変更したい際にその部分のコードを全ていちいち書き直す必要がでてきてしまう。そこで、CSS を用いることでスタイルを一括管理することができ、比較的簡単にデザイン変更をすることができるのである。ゆえに、ホームページ制作側はメンテナンスが楽になるのである。



もう一つの特徴として、SEO 対策をすることができるという点がある。これは、CSS を記述したファイルと HTML 文章ファイルを別にすることで、HTML 文章から余分なソースコードを省くことができるので、サーチエンジンがキーワードを拾いやすくなるのである。

３．CSS の書き方

まず、具体例を提示する。

```
p { color : red; }  …①
p { color : red;  font-size : 30pt; }  …②
```

CSS は①や②のような記述の仕方をしていく。まず①を見ていく。この p の部分に記述するもののことをセレクタと言う。これは{}（宣言ブロック）内のスタイルの適用対象を示している。color の部分にあたるのがプロパティ名で、red の部分が値である。①は、p というセレクタに対し文字色を赤くするというスタイルである。②は複数のスタイルを適用する場合の記述である。この場合、適用したいスタイルをセミコロンで区切って記述すれば良い。

スタイルにはいくつか種類がある。独断と偏見で比較的用いそうなものをいくつかピックアップした。

font-style…イタリック体や斜体などフォントのスタイルを指定

width:○○px、height:××px…長さの指定（単位は他にも%、em、ex、mm、cm、in、pt、pc などがある）

margin:△△px、padding:□□px…マージンやパディングの指定

border:☆☆px solid…罫線を指定

```
p,h1 { color : red; }  …③
.abc { color : red; }  …④
#abc{ color : red; }  …⑤
```

また、セレクタもいくつか紹介しようと思う。まず、①のようなものを要素型セレクタという。他にもセレクタが何種類か存在する

③…要素型セレクタを複数宣言したい場合の記述

④…class セレクタ。CSS で class セレクタを宣言し、HTML 内の要素に class を指定する。HTML のページ内で複数回用いることが可能。

⑤…id セレクタ。CSS で id セレクタを宣言し、HTML 内の要素に id を指定する。HTML のページ内で一回だけ用いることが可能。

CSS を適用する方法はいくつか存在する。HTML 文章の<head>の部分に<link rel="stylesheet" type="text/css" href="○○○.css">のような文を挿入する場合と、同じく HTML 文章の<head>の部分に<style type="text/css"><!--（スタイルをここに記述する） --></style>を挿入する方法、そして HTML のタグの要素内に直接書き込む方法である。尚、３つめの方法は推奨しない（これをしたら CSS としての意味が無い）。

４．おわりに

今回はだいぶ省いて CSS を紹介した。CSS はそこまで難しくないと思うので、これをきっかけに気になった人は調べてみると良いと思う。

[参考]

http://www.htmq.com/csskihon/index.shtml　CSS の基本

http://www.tohoho-web.com/css/　とほほのスタイルシート入門

http://allabout.co.jp/gm/gc/23897/2/　スタイルシートの class と id の使い分け

タッチスクリーンを使ってみた

情報 · 通信工学科 2 年　木村敢

aitendo で 7 インチのタッチスクリーンが 999 円だったのでつい買ってしまった。なので仕組みや制御方法について書こうと思う。

1　タッチスクリーン

様々なものに使われているタッチスクリーンだが、いくつかの種類がある。例として

- 抵抗膜方式
- 赤外線方式
- 静電容量方式

などがある。スマートフォンなどはマルチタッチ可能な静電容量方式が、NintendoDS は抵抗膜方式が使われている。今回購入した aitendo のタッチスクリーン [TP7-15486-4W] は抵抗膜方式で、その中の 4 線式抵抗膜方式が用いられている。

2　抵抗膜方式

抵抗膜方式は横から見ると下図のような構造をしていて押すと上下の電膜が接し導通する。導通した時位置によって抵抗値が変化する。

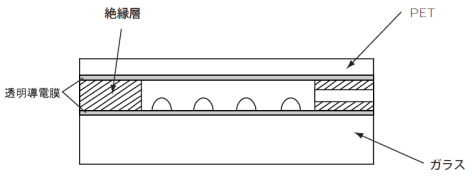


図 1: 抵抗膜式断面図

画像は http://gaje.jp/technotes/touch_screen/ のものを使用させて頂いた。

3　制御方法

長方形の各辺に電極がついていて対辺で 1 セットになっている。マイコンは電圧値を直接読み込めないので端子の電圧を読み取ることで場所を感知する。x 軸に電圧を印加した時、y 軸の電極からは触った場所の y 座標に応じた電圧が出力される。y 軸に電圧を印加した時は x 座標が x 軸の端子から電圧が出力される。なので同時に x,y 座標を感知することはできない。

4 端子を x+,x-,y+,y-とする。実際に使用するときは各端子をプルダウンし、x 軸を測りたい時は y 軸に電圧をかけ (y+に ADC の基準電圧,y-に gnd)、x 端子のどちらかで ADC を行う。y 軸を測りたいときは逆にする。高速で x 軸と y 軸を連続で読み取る事で現在座標が読み取れる。

4　実際に使用してみて

同じ場所を押し続けていても値が少しふらつき、さらに各軸の値を連続で読み込むときはより値がふらついた。これは切り替えたとき電圧が立ち上がるまで少し待つことによって改善した。また、電圧は基本的にはリニアに変化するのだが、誤差が大きいため押した場所と ADC の値が一致しなかった。これは DS のように始

めに四隅の電圧値を記憶させ、その後ソフトで補正するのが良いと考えられる。

制御自体は簡単だが実用に耐え得るレベルで使用するのは多大な苦労を伴うだろう。もし使用するならば、専用 IC が発売されているのでそれを使うのも一つの手である。しかし、この IC はやや値が張るので使わずに済ませたい。

これと液晶を組み合わせればかなり見栄えのいいものができるので是非ともしっかりと制御したいものである。

また、現在 [2014/2/28]aitendo で 8 インチのタッチスクリーンが 777 円で販売されているそうなので是非購入して制御してみてはいかがだろうか。

図 2: 今回使用したタッチスクリーン

5 参考文献

- aitendo

  http://www.aitendo.com/product/5276

- 妥協の電子工作

  http://npyo.net/pages/touch/touch.html

- タッチスクリーンを Arduino から使う

  http://gaje.jp/technotes/touch_screen/

- 電子工作教室/タッチスクリーンの制御

  http://www.picfun.com/PIC24F/AP/app24F16.html

電気通信大学 工学研究部 部報 46 号

鍋でお湯沸かした

時の蒸発量調べてみた

2 年 I 科 住澤 柾秀

注意 ： 他の人みたいな真面目な内容じゃないです m(_ _)m

日常生活の必須品といえば・・・そう、誰もがご存知「インスタントラーメン」です！カップラーメンじゃなくて鍋で作るほうです。我が家ではお昼ご飯によく出されます。

さて、このインスタントラーメンの作り方の最初の行程は「沸騰したお湯 500ml に麺を入れる」ですね。水 500ml ではなく沸騰水 500ml と書いてあります。ここで疑問。

水を沸騰させるとどれくらい蒸発するのか？

というわけで、簡単な実験をしたいと思います。まぁ、湿度とかに影響を受ける実験なので、「へぇ～そんなもんか～」的な感じで読み流しちゃって下さい。厳密なことは指摘しないで(泣

用意するもの

- 鍋 :我が家愛用 t-fal(直径 16cm)
- ガスコンロ :リンナイ RS38MS2R
- 容器 :てきとーなの
- 計量計 :TAMITA の家庭用の(デジタル表示)
- 温度計
- 水道水 :たくさん

最初に沸騰とは何かを定義します。沸騰=100℃と思われてる方もいると思いますが、実際は違います。沸騰した時の温度は、気圧に依存して変化します。今回、鍋に多量の水を入れ、加熱したところ、100℃で水温上昇がなくなりました。なので、今回の実験では沸騰=100℃とします。

では実験。有効数字は 3 桁です。

今回用いた水道水の温度は 15.0℃でしたので、この密度は 0.999g/ml になります。つまり、500ml の水とは $0.999\ [\mathrm{g/ml}] \times 500\ [\mathrm{ml}] \approx 500\ [\mathrm{g}]$ です。以下実験手順。

1） 計量計の上に乾燥させた容器を乗せ、500g になるように水を入れます。
2） これを乾燥した鍋に移し、水温を測りながら加熱します。
3） 100℃になった時点で乾燥させた容器に移し、お湯の重さを測ります。

3 回実験して、結果はこんなかんじになりました。

表 1:実験結果

実験回数	沸騰後の重さ g	重さの変化 g
1	473	−27
2	475	−25
3	474	−26

平均して 26g 減ですね。

この実験では容器→鍋→容器と水を移します。このとき、一部の水は元の容器または鍋に付着したままになります。この総量を調べてみると 1g だったので、先ほどの結果から差し引いて、平均 25g の水が蒸発したことになります。率にして 5%です。

意外と少ないですね。裏面の説明書き通りに作るより 5%味が濃くなる・・・気にならないわ！笑。
そもそも、説明書きに「水〇〇ml を沸騰させ・・・」って書いてくれればいいんですよ！と思って調べてみるとありました。

明星 評判屋 中華そば 新鶏ガラスープ塩味「お湯 450ml を沸騰させ・・・」

なんで「お湯」かは知りませんが、親切な説明ですね。

最後に。なんて雑な実験(?)レポートなんでしょう！一年生はこんなのまねしないで、もっとクオリティー高いの作って下さいね（一年生は基礎科学実験という実験が待ち受けてます）。じゃないと、単位落とします。留年の危機です。レポート書くの苦手な人は、早めに先輩に過去レポをもらいましょう。以上！

参考文献

・ Akira, 水の密度表, http://www.geocities.jp/leitz_house/benri/program/mitudo.htm

BeagleBone Black を用いた SoftEther VPN Server の構築（準備編）

先進理工学科 1 年 村上 大和

平成 26 年 2 月 28 日

1 はじめに

最近，Raspberry Pi や Galleo 等，手のひらサイズの Linux マイコンボードが流行している．これらの強みの一つに，低消費電力ながら Linux が動作することで，サーバー用途に適していることが挙げられる．今回は準備編として，BeagleBone Black(以下 BBB）の OS インストール方法を紹介する．筆者は Linux，サーバー，ネットワークについては素人であるため，足りない部分が多いと思われるが，そこは目を瞑ってほしい．

1.1 BeagleBone Black とは

Beagleboard 社が発売しており，電気通信大学生協で 5000 円弱程度で購入できる．また，Amazon 等の通販サイトや，秋月電子通商等の専門店でも購入できる．よく比較に挙げられる物に Raspberry Pi がある．それに比べると BBB は後発品であることから高スペックであるが，Web 日本語資料が少ないというデメリットもある．しかし，GPIO を操作する等の高機能マイコンとしての用途ではなく，低消費電力 Linux マシンとして扱うと，Raspberry Pi の資料を殆どそのまま流用できるため，このデメリットは気にならない．詳しい比較は Web 上で検索すれば多く出るため，ここでは割愛する．

1.2 VPN とは

VPN とは，Virtual Private Network（仮想プライベートネットワーク）の略称で，物理的に離れたネットワークを，インターネットなどのパブリックネットワークを介してあたかも直接同一プライベートネットワーク上にあるように拡張する技術である．これにより，専用回線を引くことなく，外出先で自宅 LAN 内の機器にアクセスしたり，身近なところでは学内有線接続による認証が必要な Microsoft Office の KMS によるライセンス認証が無線 LAN から可能になる．

1.3 SoftEther VPN とは

SoftEther VPN はオープンソースの、無償で、複数プラットフォームおよび複数 VPN プロトコル対応の VPN ソフトウェアであり、筑波大学 における研究プロジェクトとして開発されています。SoftEther VPN はソフトイーサ株式会社の PacketiX VPN 4.0 に相当するフリーバージョンです。(SoftEther VPN プロジェクトホームページより引用 : http://ja.softether.org/）

無料，インストールが簡単で，ファイアウォール貫通性が高く，管理ソフトが GUI であることから，素人にも扱いやすいと考え選択した．

2 必要なもの

- BBB
- LAN ケーブル
- microSD カード（2GB 以上）
- OS インストール用コンピュータ（今回は Windows）
- 5V AC アダプタ

3 OS のインストール

BBB は Angstrom Linux を内蔵しているが，これは資料が少なく，使い勝手が悪い．そのため，多くのユーザーが自分好みの OS を microSD カードにインス

トールし，それをブートして使用している．今回は手元に Linux マシンが無くてもブート用 microSD を用意する方法を紹介する．

3.1 OS の選択

今回は Debian 系 OS のひとつである Ubuntu を採用した．これは，個人的に使用経験があること，また資料が多く存在し，フォーラムが活発であることから初心者にも使用しやすいと考えたからである．近年，ARM を主要アーキテクチャとして扱うディストリビューションが増えているため，他にも Fedora や Debian 等が使用できる．後述する”start.htm”末尾に BBB で動作可能な OS の一覧がある．それらの内自分の気に入った OS をインストールするのが良いだろう．

OS については宗教戦争に発展しかねないため，多くは触れないでおく．

3.2 ドライバのインストール

BBB を付属の USB ケーブルを用いて PC と接続すると，リムーバブルメディアとして認識される．まずはじめに BBB のドライバを入れる必要がある．リムーバブルディスク ”BeagleBone Getting Started ”内の ”start.htm”を開き，それぞれの環境にあったドライバを選択し，インストールする．

3.3 接続の確認

Tera Term 等のターミナルエミュレータソフトを用いて，BBB に ssh 接続する．この時，デフォルトのアドレスは ”192.168.7.2”ユーザー名は ”root ”パスワードは空欄となっているが，環境によって異なる場合があるため，”start.htm”にて確認する．接続が確認できたら，BBB に LAN ケーブルを繋ぎ，同一 LAN 内に BBB を置く．ターミナルで **ifconfig** コマンドを実行し，eth0 の IP アドレス（inet addr）をメモする．環境によっては IP アドレスが再接続毎に変更する場合があるが，筆者の環境ではそのようなことがなかったため，今後はこのアドレスを用いて接続する．

3.4 Ubuntu のインストール

Image Writer for Windows（https://launchpad.net/win32-image-writer），BeagleBoradUbuntu イメージ（http://elinux.org/BeagleBoardUbuntu）をダウンロードし，適当なソフト（7zip 等）で解凍する．microSD を PC に接続し，先ほどのツールで microSD に書き込む．

念のため **shutdown** コマンドでシステム終了後 BBB の電源を外し，microSD カードを挿入する．S2 ボタン（LAN ケーブル等とは反対側にある）を押しながら電源をつなぎ，LED が点滅を始めるのを確認後，ボタンから手を離す．ターミナルから，先ほどメモしたアドレスに ssh 接続する．このとき，ユーザー名は”ubuntu”，パスは”temppwd”となっている．電源投入後，しばらく経っても接続がうまくいかない場合，IP アドレスが変わっている場合があるため，IgNetMap 等の IP アドレス調査ツールを用いるか，BBB とディスプレイを MicroHDMI ケーブルで接続し，確認する．

とりあえず **sudo su** コマンドを叩き，管理者ユーザーに切り替える．**apt-get update** コマンドでパッケージをアップデート，**apt-get upgrade** コマンドでアップグレードする．以上でインストールは完了だ．

4 終わりに

今回は OS のインストールまでの記事となったが，次号では SoftEther VPN のセットアップを解説する予定である．

このままでは microSD の約 2GB しか使用されていないため，パーティションを拡張する必要がある．他にも，ユーザ名，パスワード等も変更する必要があるが，それらは個人の用途により異なるため各自参照されたい．

5 参考

- beagleboard.org Getting Started (http://beagleboard.org/Getting%20Started)
- elinux.org (http://elinux.org/)

ブラシレス DC モーターのベクトル制御

2 年　横田　嶺

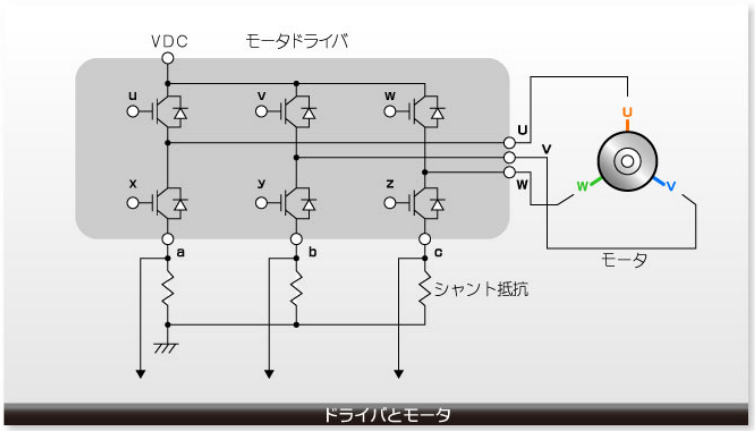
2014 年 2 月

1　ブラシレス DC モーターとは

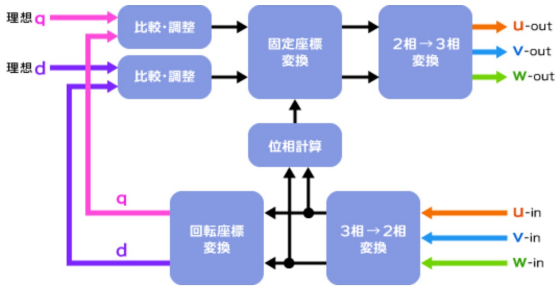
通常の DC モーターからブラシと整流子を取り除き，制御回路によって駆動させるモーター。永久磁石同期電動機。

DC モーターのブラシは回転とともに摺動し摩耗する。摩耗が進行すると整流時に異常な火花を生じ電蝕を起こしたり，整流作用が悪化し大電流が流れているにもかかわらず発生トルクが小さいなどのトラブルを引き起こす．そのため長期の運用では保守点検が必要である．

ブラシレスモータは機械的接点を有さないため様々なメリットがある．まず，メンテナンスフリーであり寿命が長いこと．整流時の機械・電気ノイズがないこと．機械的ロスがなく高効率であることなどがあげられる．



2　ベクトル制御



2.1　概要

ブラシレス DC モーターはインバータにより与えられる交流電流で駆動する．また，速度を変化させる場合はドライバによる可変速制御をする．

三相交流で駆動するモーターを最も効率よく回すためにベクトル制御というものがある．三相という複雑な電流を制御するのは容易ではない．座標変換を用いて三相交流を二相交流に，二相交流を二相直流に見立て扱いやすく正確な制御を行うものである．

2.2　原理

2.2.1　三相交流

ブラシレス DC モーターを駆動させる三相交流電流はそれぞれ位相が 120° ズレた正弦波である．3 つの相を u 相，v 相，w 相として相電圧と相電流を e_u，e_v，e_w，i_u，i_v，i_w とするとき，

$$\begin{cases} e_u = E\sin(\omega t) \\ e_v = E\sin\left(\omega t + \frac{2\pi}{3}\right) \\ e_w = E\sin\left(\omega t - \frac{2\pi}{3}\right) \end{cases} \tag{1}$$

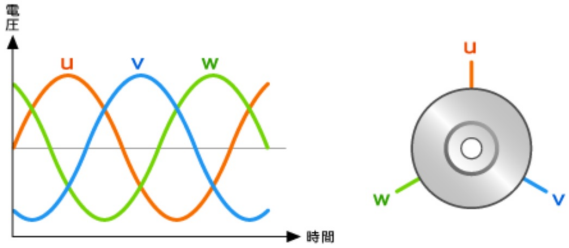
$$\begin{cases} i_u = I\sin(\omega t + \psi) \\ i_v = I\sin\left(\omega t + \psi + \frac{2\pi}{3}\right) \\ i_w = I\sin\left(\omega t + \psi - \frac{2\pi}{3}\right) \end{cases} \tag{2}$$

であるから，

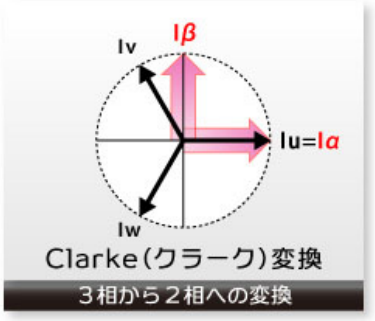
$$e_u + e_v + e_w = 0 \tag{3}$$

$$i_u + i_v + i_w = 0 \tag{4}$$

となる．



2.2.2　クラーク変換



三相交流を二相交流に変換することをクラーク変換という．α 軸と β 軸からなる直行座標系を考える．α 軸と u 相軸が合うように配置すると三相交流は位相のズレが 120° であるため，α 方向の電流 i_α と β 方向の電流 i_β は

$$\begin{cases} i_\alpha = i_u + i_v \cos\frac{2\pi}{3} + i_w \cos\frac{-2\pi}{3} \\ i_\beta = 0 + i_v \sin\frac{2\pi}{3} + i_w \sin\frac{-2\pi}{3} \end{cases} \tag{5}$$

である．これを行列で表現すると

$$\begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} = \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} i_u \\ i_v \\ i_w \end{bmatrix} \tag{6}$$

となる．この操作が三相から二相への座標変換であり，変換に使われた行列を変換行列という．

座標変換の前後で電力が変わってしまうと使いづらい．相電圧と相電流の列ベクトル $[v]$，$[i]$ を用いて電力は行列で

$$P = [v]^t [i] \tag{7}$$

と表すことができる．座標変換後の行列をそれぞれ v'，i' とすると電力は変わらないという条件

より，

$$\begin{aligned} P = [v]^t[i] &= [v']^t[i'] \\ &= [[c][v]]^t[c][i] \\ &= [v]^t[c]^t[c][i] \end{aligned} \tag{8}$$

となる．ただし $[c]$ は変換行列．以上より，

$$[c]^t[c] = [E] \tag{9}$$

であるから変換行列とその転置行列の積が単位行列である必要がある．ということは転置行列は元の行列の逆行列に等しい．逆行列が存在するためには正方行列である必要があるため，三相から二相への変換行列 $[C_T]$ に調整のための係数 k_1 と正方行列にするための k_2，k_3，k_4 をつける．

$$\begin{aligned} [C_T]^t[C_T] &= [C_T]^{-1}[C_T] \\ &= k_1 \begin{bmatrix} 1 & 0 & k_2 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} & k_3 \\ -\frac{1}{2} & -\frac{\sqrt{3}}{2} & k_4 \end{bmatrix} \times k_1 \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & \frac{\sqrt{3}}{2} \\ k_2 & k_3 & k_4 \end{bmatrix} \\ &= [E] \end{aligned} \tag{10}$$

これを解いて

$$k_1 = \sqrt{\frac{2}{3}} \tag{11}$$

$$k_2 = k_3 = k_4 = \frac{1}{\sqrt{2}} \tag{12}$$

となるからクラーク変換 $[C_T]$ は

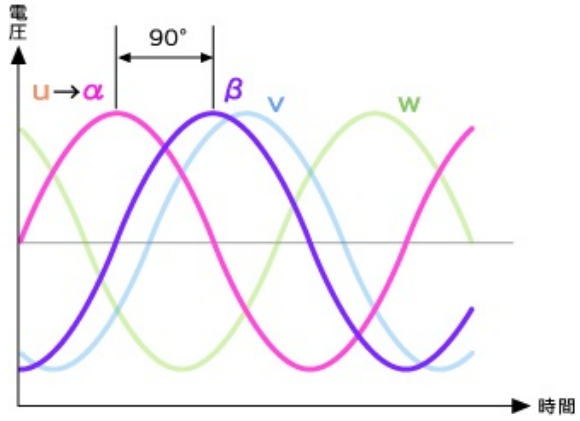
$$[C_T] = \sqrt{\frac{2}{3}} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \tag{13}$$

となる．追加した k_2，k_3，k_4 を削除し結局

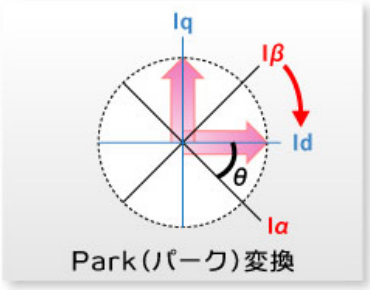
$$\begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} = \sqrt{\frac{2}{3}} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} I\sin(\omega t + \psi) \\ I\sin\left(\omega t + \psi + \frac{2\pi}{3}\right) \\ I\sin\left(\omega t + \psi - \frac{2\pi}{3}\right) \end{bmatrix} \tag{14}$$

となる．ここで $i_\alpha = i_u$ とすると

$$\begin{cases} i_\alpha = i_u = I\sin(\omega t + \psi) \\ i_\beta = \frac{\sqrt{3}}{3} i_u + \frac{2\sqrt{3}}{3} i_v = I\sin\left(\omega t + \psi + \frac{\pi}{2}\right) \end{cases} \tag{15}$$



2.2.3 パーク変換



二相静止座標 (i_α, i_β) を二相回転座標 (i_d, i_q) に変換することをパーク変換という．i_α と i_β はクラーク変換で得られ，ロータの回転角 θ を既知とすると，

$$\begin{cases} i_d = i_\alpha \cos\theta + i_\beta \sin\theta \\ i_q = -i_\alpha \sin\theta + i_\beta \cos\theta \end{cases} \tag{16}$$

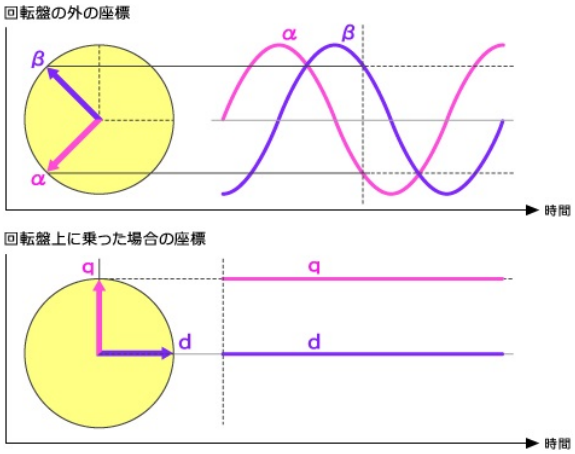
である．行列で表すと，

$$\begin{bmatrix} i_d \\ i_q \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} \tag{17}$$

となり，パーク変換 $[C_P]$ は

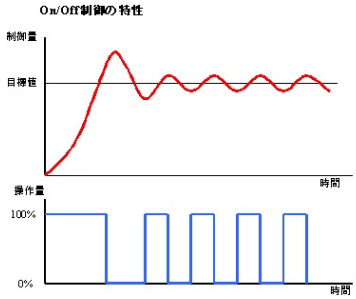
$$[C_P] = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \tag{18}$$

となる．逆行列による逆パーク変換ができるので電力は座標変換で変わらない．



2.2.4 PID 制御

フィードバック制御のひとつで入力の制御を出力値と目標値の偏差，積分，微分の三要素によって行う方式．オンオフ制御で目標値を中心に振動してしまうのを回避するために用いられる．



ブラシレス DC モーターの各相に抵抗成分とインダクタンス成分があるモデルを考える．相電圧は，

$$\begin{bmatrix} e_u \\ e_v \\ e_w \end{bmatrix} = R \begin{bmatrix} i_u \\ i_v \\ i_w \end{bmatrix} + \frac{d}{dt} \begin{bmatrix} \phi_u \\ \phi_v \\ \phi_w \end{bmatrix} \tag{19}$$

$$\begin{bmatrix} \phi_u \\ \phi_v \\ \phi_w \end{bmatrix} = \begin{bmatrix} L & M & M \\ M & L & M \\ M & M & L \end{bmatrix} \begin{bmatrix} i_u \\ i_v \\ i_w \end{bmatrix} \tag{20}$$

と表すことができる．ただし R は巻線抵抗，ϕ は電機子鎖交磁束，L は各相自己インダクタンス，M は各相間相互インダクタンス．

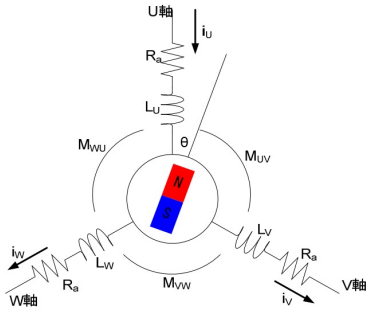
したがって dq 座標系での電圧方程式は

$$\begin{aligned}
\begin{bmatrix} e_d \\ e_q \end{bmatrix} &= [C] \begin{bmatrix} e_u \\ e_v \\ e_w \end{bmatrix} \\
&= [C_P][C_T] \begin{bmatrix} e_u \\ e_v \\ e_w \end{bmatrix} \\
&= \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \times \sqrt{\frac{2}{3}} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} e_u \\ e_v \\ e_w \end{bmatrix} \\
&= \sqrt{\frac{2}{3}} \begin{bmatrix} \cos\theta & \cos\left(\theta - \frac{2\pi}{3}\right) & \cos\left(\theta + \frac{2\pi}{3}\right) \\ -\sin\theta & -\sin\left(\theta - \frac{2\pi}{3}\right) & -sin\left(\theta + \frac{2\pi}{3}\right) \end{bmatrix} \begin{bmatrix} e_u \\ e_v \\ e_w \end{bmatrix}
\end{aligned} \tag{21}$$

これを計算すると

$$\begin{bmatrix} e_d \\ e_q \end{bmatrix} = \begin{bmatrix} R + L\frac{d}{dt} & -\omega L \\ -\omega L & R + L\frac{d}{dt} \end{bmatrix} \begin{bmatrix} i_d \\ i_q \end{bmatrix} \tag{22}$$

となり，uvw 座標系の相電流より dq 座標系の相電圧が求まる．この電圧を操作してモーターの回転を制御する．



比例 (Proportional) 動作

入力値を出力値と目標値の偏差の一次関数として制御する方法．入力値を $x(t)$，出力値を $y(t)$，目標値を y_0 とすると

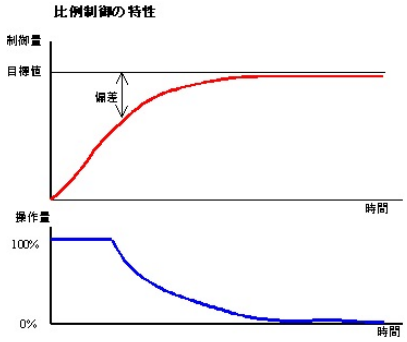
$$x(t) = K_p(y(t) - y_0) + x_0 \tag{23}$$

となる．ただし K_p は比例ゲイン，x_0 は $y(t) = y_0$ のときにそれを維持するために必要な入

力値．書き換えると

$$\Delta x(t) = K_p \Delta y(t) \tag{24}$$

P 動作でフィードバックすることを P 制御という．



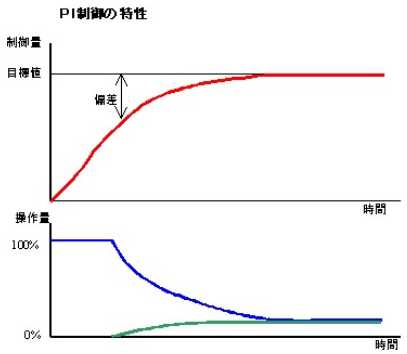
積分 (Integral) 動作

P 制御だけでは制御量が目標値に近づくと操作量が小さくなりすぎ，それ以上細かく制御できなくなりいつまでたっても目標値に到達できない．このようにして生じる出力値と目標値との差を残留偏差という．

そこで残留偏差の時間積分に比例して入力値を変化させる積分動作を加える．

$$\Delta x(t) = K_p \Delta y(t) + K_i \int_0^t \Delta y(\tau) d\tau \tag{25}$$

ただし K_i は積分ゲイン．P 動作と I 動作でフィードバックすることを PI 制御という．

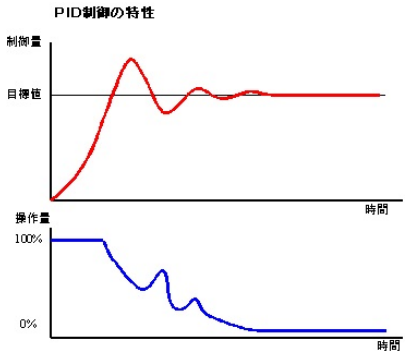


微分 (Differential) 動作

通常は PI 制御で目標値まで到達できる．しかし，I 動作は一定の時間経過がないと効果がない．そのため外乱があったときの応答性が悪い．そこで外乱に対し偏差を見て，その大きさに比例した入力を行う微分動作を加えると応答速度が増す．

$$\Delta x(t) = K_p \Delta y(t) + K_i \int_0^t \Delta y(\tau) d\tau + K_d \frac{d\Delta y(t)}{dt} \tag{26}$$

ただし K_d は微分ゲイン．P 動作，I 動作，D 動作によるフィードバック制御を PID 制御という．



モーターのトルクを考える．二相直流モーターでは d 軸電流はトルクに寄与しない．トルクを発生させる i_q を用いてトルク T は，

$$T = p\phi_a i_q \tag{27}$$

となる．ただし p は極対数，ϕ_a は電機子鎖交磁束実効値でどちらも定数．よってトルクは電流 i_q に比例する．

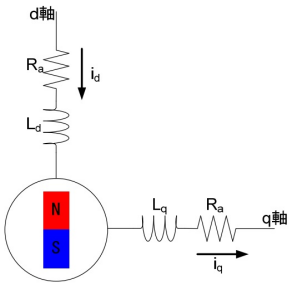
慣性モーメントを I'，負荷トルクを T_L とすれば運動方程式は以下で表される．

$$I' \frac{d\omega}{dt} = p\phi_a i_q - T_L \tag{28}$$

電圧の指令値 ${e_d}^*$，${e_q}^*$ はラプラス変換を用いて PI 制御で行い，

$$\begin{cases} {e_d}^* = \left(K_{P_{i_d}} + \dfrac{K_{I_{i_d}}}{s} \right) ({i_d}^* - i_d) \\ {e_q}^* = \left(K_{P_{i_q}} + \dfrac{K_{I_{i_q}}}{s} \right) ({i_q}^* - i_q) \end{cases} \tag{29}$$

となる．なお，d 軸電流は 0 になるよう制御するため ${i_d}^* = 0$ である．すなわち，q 軸電流のみ変化させトルクを発生させるということはただの直流モーターを制御しているのと等しい．



2.2.5 逆パーク変換

逆パーク変換により二相直流座標上の指令電圧を二相交流座標に変換する．

$$\begin{aligned}\begin{bmatrix} {e_\alpha}^* \\ {e_\beta}^* \end{bmatrix} &= [C_P]^{-1} \begin{bmatrix} {e_d}^* \\ {e_q}^* \end{bmatrix} \\ &= \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} {e_d}^* \\ {e_q}^* \end{bmatrix} \end{aligned} \tag{30}$$

2.2.6 逆クラーク変換

逆クラーク変換により二相交流座標上の指令電圧を三相交流座標に変換する．

$$\begin{aligned}\begin{bmatrix} {e_u}^* \\ {e_v}^* \\ {e_w}^* \end{bmatrix} &= [C_T]^{-1} \begin{bmatrix} {e_\alpha}^* \\ {e_\beta}^* \end{bmatrix} \\ &= \sqrt{\frac{2}{3}} \begin{bmatrix} 1 & 0 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} \\ -\frac{1}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} {e_\alpha}^* \\ {e_\beta}^* \end{bmatrix} \end{aligned} \tag{31}$$

3　実装方法

工学研究部で毎年出場している WEM ワールドエコノムーブ（通称：エコラン）を想定して実際にどのような構成で実装するか考える．

ベクトル制御には大きく分けて以下のものが必要である．

- ブラシレス DC モーター
- 電源
- インバータ
- マイコン

3.1　ブラシレス DC モーター

ブラシレスモータは減速機の付いていないインナーローター型のものを使用し，モーターの軸とタイヤに付けた歯車をチェーンで結ぶこととする．モーター軸の回転角 θ を知る必要があるためセンサを内蔵するか測定した相電流により推定する 2 つの方法がある．

3.2　電源

直流 12V の鉛蓄電池が 4 つ支給される．モーターが 24V 駆動であるため 4 つ並列で 12V×4 ににして昇圧するか，2 つ直列を並列で 24V×2 にするかがパターンとして考えられる．並列の方が電流を多く取り出せるが DCDC コンバータによる昇圧の変換効率も影響する．

加速時には電力を消費し，インバータによる減速時は電力を回生する．電力を回生して減速することを回生ブレーキという．双方向 DCDC コンバータを使えば電圧の昇降が自在であるため低速域から高速域の全域にわたって回生ブレーキを使用することができる．双方向 DCDC コンバータは昇圧チョッパ，降圧チョッパを組み合わせたもので，スイッチングは MOSFET などで行いマイコンで制御する．

3.3　インバータ

三相インバータ回路はハーフブリッジ 3 つで構成される．スイッチング素子には GTO サイリスタ，IGBT，パワー MOSFET などが考えられる．GTO サイリスタは大電力低速用，IGBT は中電力中低速用，パワー MOSFET は小電力高速用である．エコランを動かす程度のモーターであればパワー MOSFET が適している．

ハーフブリッジ回路を駆動させるには前段にゲートドライブ回路が必要である．ハーフブリッジは構造上，ハイサイドドライブが難しい．そこでハイサイドのゲートにブートストラップ方式による電圧の上乗せを行って駆動させる．ブートストラップ方式ではコンデンサに充電する時間を要す

るため，PWM のデューティ比を 100 %すなわちハイサイドを常時オンすることはできない．しかし三相交流を生成する場合，ハイサイドローサイドを交互にオンオフするためこの問題を考える必要はない．

ゲートドライブ回路はマイコンからのデジタル信号からゲート電圧を出力するものが IC 化されておりそれを用いることで回路を簡略できる．

3.4 マイコン

ベクトル制御をする場合は座標変換など高度な演算をさせるため性能の良いマイコンが必要である．現在 32bit マイコンが広く使われるようになり，三角関数演算や浮動小数点演算，デッドタイム付き相補三相 PWM 生成，高速 AD 変換などのベクトル制御に欠かせない機能があるものを用いる．

3.5 まとめ

ベクトル制御を実装するには知識と技術力が必要であることは明白である．座標変換などの高度な演算をやらせるソフトウェアと演算結果を出力するハードウェアが完璧に連動してはじめて動作するため，実装には相当な時間を要すると思われる．

4 参考文献

参考文献

[1] インバータ応用マニュアル三菱電機株式会社編，電気書院

[2] Interface2014 年 4 月号ワンチップでなめらかモーター制御，CQ 出版

[3] 電気技術解説講座　電動機ベクトル制御の基礎 1 4，日本電気技術者協会，
http://www.jeea.or.jp/course/contents/07118/

[4] ベクトルエンジンとベクトル制御，東芝，
http://www.semicon.toshiba.co.jp/product/micro/mcupark/vector/1323680_29030.html

[5] インバータ制御における座標変換，富士通，
http://edevice.fujitsu.com/jp/aplnote/ja-pdf/MB9B500-AN706-00032-1v0-J.pdf

[6] モーターの PID 制御，
http://www.picfun.com/motor05.html

[7] RX62T マイクロコントローラによるモータ制御　永久磁石同期モータのセンサレスベクトル制御編，RENESAS，
http://documentation.renesas.com/doc/products/mpumcu/apn/rl78/r30an0117jj0100_motor.pdf

[8] 絶対変換について，
http://motor.geocities.jp/workhard555/motor/zettaihenkan.pdf

国立大学法人 電気通信大学
工学研究部 部報 第46号

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

発行　横田 嶺
編集人　住澤 柾秀
表紙　河村 大輝
発行　2014年3月14日
執筆　工学研究部 部員