

# つぎ、なにをする？

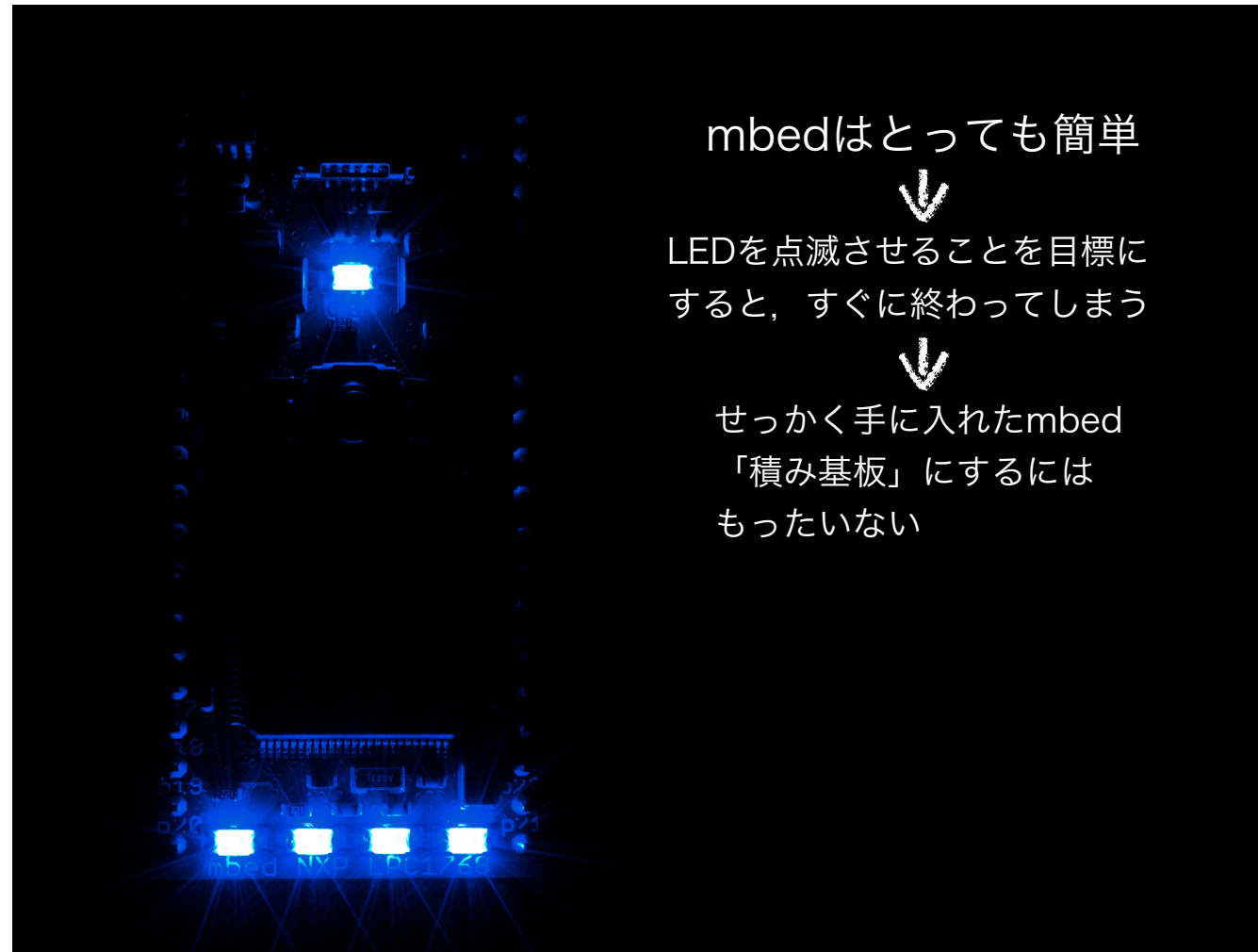
LEDのつぎ：セカンドステップ

「mbedのセカンドステップ」というお題で話をしようと決めたまでは良かったのですが、何が「セカンドステップ」となり得るかで悩みました。つばいさんからのPCA9629A<sup>※1</sup>(ステッピングモータ・コントローラ)の紹介にしてはどうかとアドバイスを頂き、取り組みはじめたのですが、これにはI<sup>2</sup>Cやステッピングモータなどの基本を押さえた上での解説が必要です。しかしこれを短時間で行なうのは非常に困難。私も仕事でそういう製品の担当であるわけですが、モータ関連は3年前まではよく知らない分野でした。初心者の方の中には当時の私と同じような人も居られるはず.. そんなわけで会社の名前を背負って前に出てきているにも関わらず、弊社製品の説明はしません(^^; ホントに簡単にステッピングモータを回す基本の所を解説するだけになります。プレゼン・スライドのページ数はびっくりするぐらい多いですが、もうそのほとんどがパラパラマンガ的に使うページなので内容は非常に薄いです..

※1 : PCA9629, PCA9629A → <http://ip.nxp-lpc.com/sm/motor/>

まずはこのスライドについての説明です。いろいろ前置きが長いのです(^^;

NXPの話はほとんどしません。mbedの話も後半になってやっと出てきます



mbedはフツートのマイコンではありません.

フツートのマイコンならLED点滅プログラムが動いただけで大きな満足感が得られます.

mbedでのLED点滅はものの5分もあれば動いてしまうので、これだけやって「積み基板」にしておくのはもったいないです.

フツートのマイコンは

開発ツールを探す→PCにインストール→ソフトを作れるようにする→出来上がった実行ファイルをマイコンに焼く方法を模索する→ツールをPCにインストールする→USB-シリアル変換ケーブルを買いに行く(またはデバッガ基板を買いに行く)→うまく焼けずにいろいろ困る→動作がおかしいのでサンプルコードを再確認にしてソフトウェアの設定も調整する→そんなことを繰り返しているうちにやっとLED点滅ソフトが動く

↓↓↓

もうこれだけで、おなかいっぱい ←ですね



mbedなら..

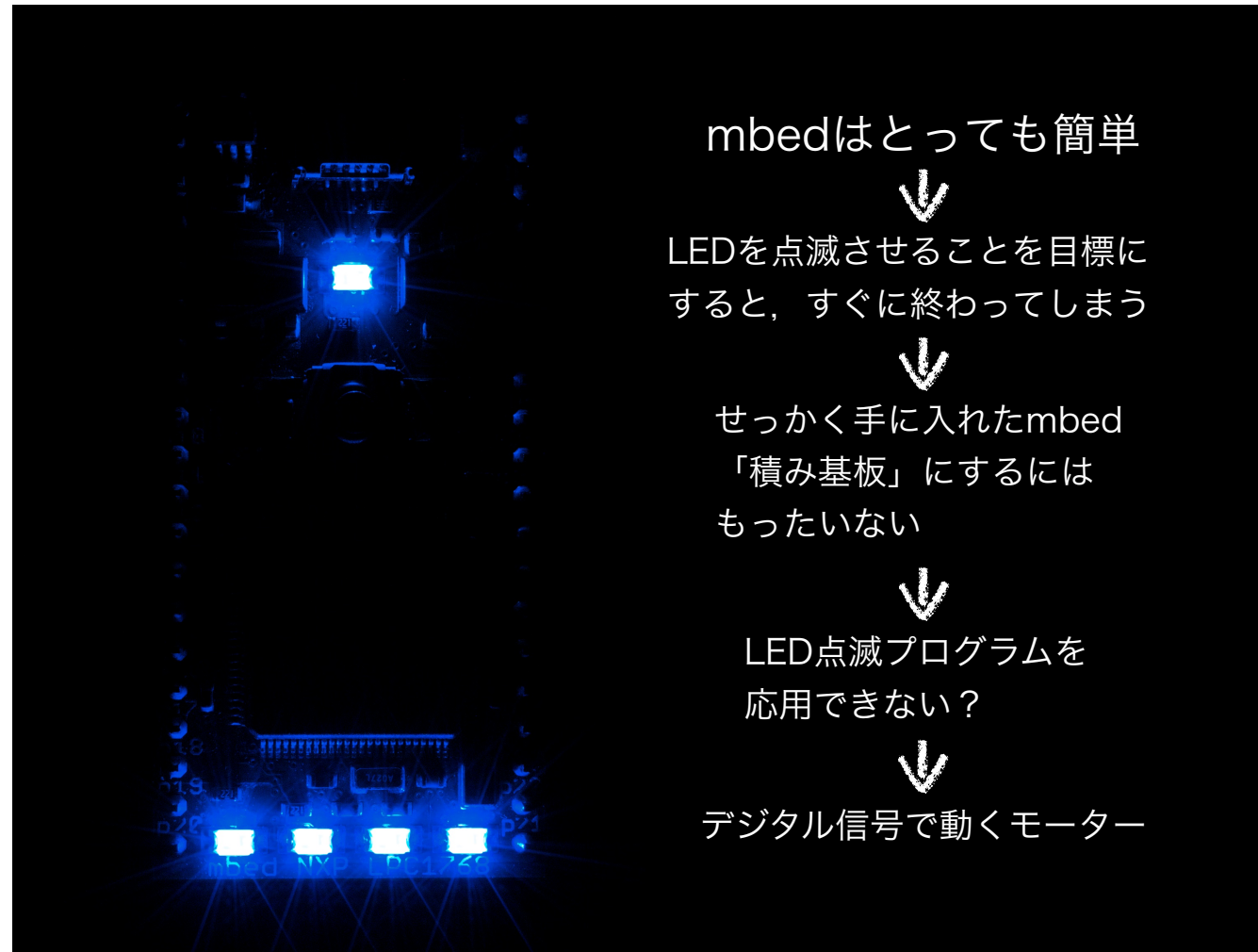
mbed.orgサイト内にmbedで使えるライブラリやハードの結線図などの情報がテンコ盛り。この中から興味のあるものを見つけて、試してみれそうなものからやってみると世界が広がります。

CookbookページやComponentsページを参考に

<https://mbed.org/cookbook/Homepage>

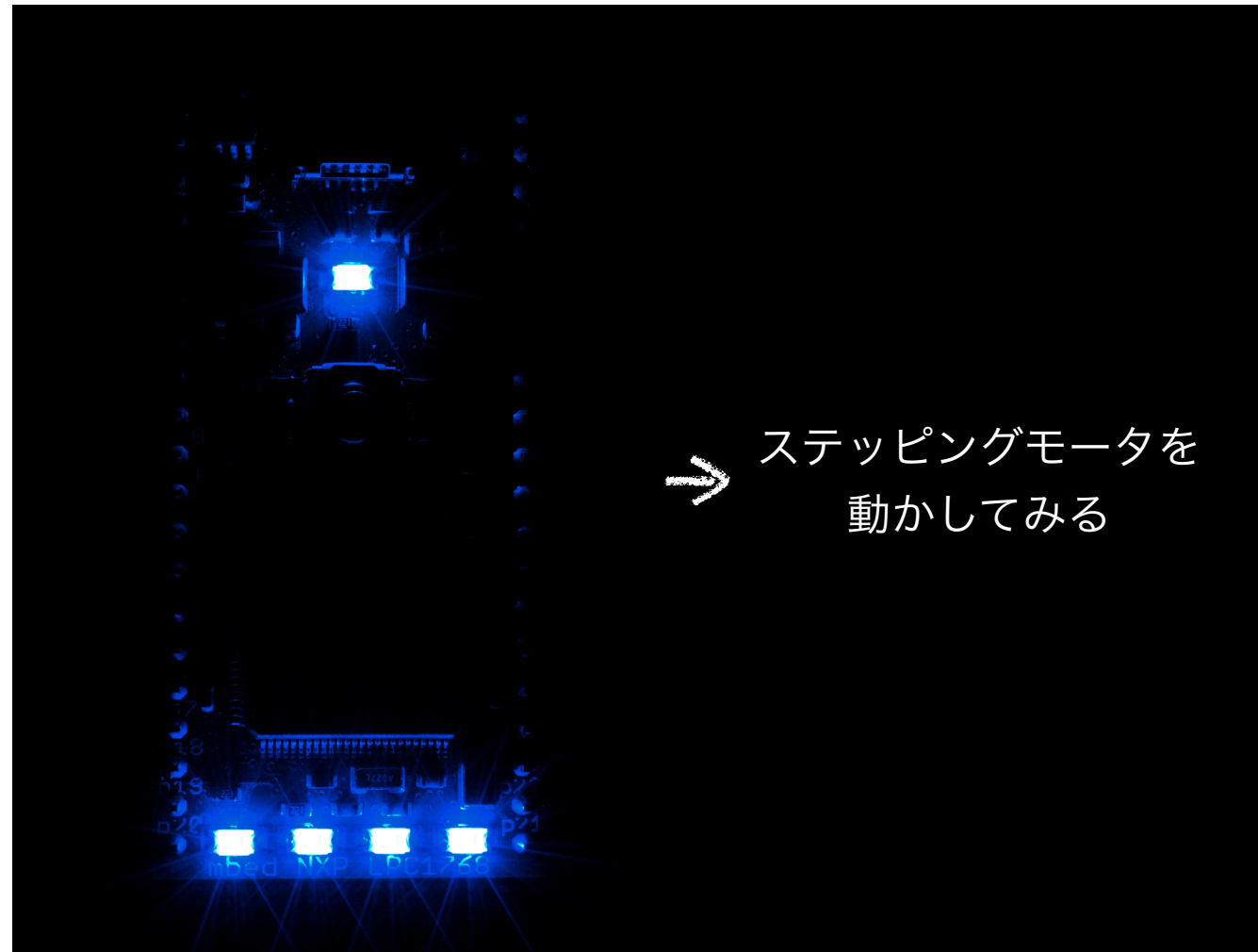
<https://mbed.org/components/>





LEDの順次点滅ができればそれがそのまま応用できるもの..

デジタル信号で動くモーターがあるので、それを説明してみたいと思います



ステッピングモータ(ステッパ・モータともいう)です

秋月電子通商

マイページ

注文書

お問い合わせ

かこの中身

トラ技広告

回路図集

クイック注文: 通版コードを入力  
(アルファベット+数字)  
- 数量 注文  
通版コード入力フォーム

商品カタログ

新商品

お知らせ

注文方法

振込先

よくある質問

ダウンロード

トラ技広告(PDF)

配送状況確認

ログイン

検索

トップ > パーツ一般 > 動力部品 > ステッピングモーター > ステッピングモータSPG20-1362



AA

この商品が友達に教える

お気に入り追加する

ステッピングモータSPG20-1362

[SPG20-1362]  
通版コード P-04241  
発売日 2010/11/09  
メーカーカテゴリ [日本電産コバル株式会社](#)

小型のギアヘッド付ステッピングモーターです。  
ギア比18:1、軸径3.0mm、360ステップ(ステップ角度1度)  
(上記の分解能はギアに拠るもので、モータ自身は18°ステップです)  
■コイル抵抗68Ω  
■平均消費電力2W以内  
■約24g  
■出力軸最大負荷19.6mN・m  
■「PICステッピングモータドライバキット」([K-00154](#))でドライブ可能です。

参考資料  
 [SPG20-1362.pdf](#)

関連商品 > [モータ](#) / [OPAMP](#) / [TR](#) / [FET](#) / [電源IC](#) / [オーディオ](#) / [ダイオード](#) / [抵抗](#) / [VR](#)

[P-04241] ステッピングモータSPG20-1362

1個 ¥250(税込)

在庫数 788個

購入数量 1 個

かこに入れる

かこの中身を見る

関連商品



ステッピングモータSPG20-1362

P-04241 [SPG20-1362]  
1個 ¥250(税込)

購入数量 個

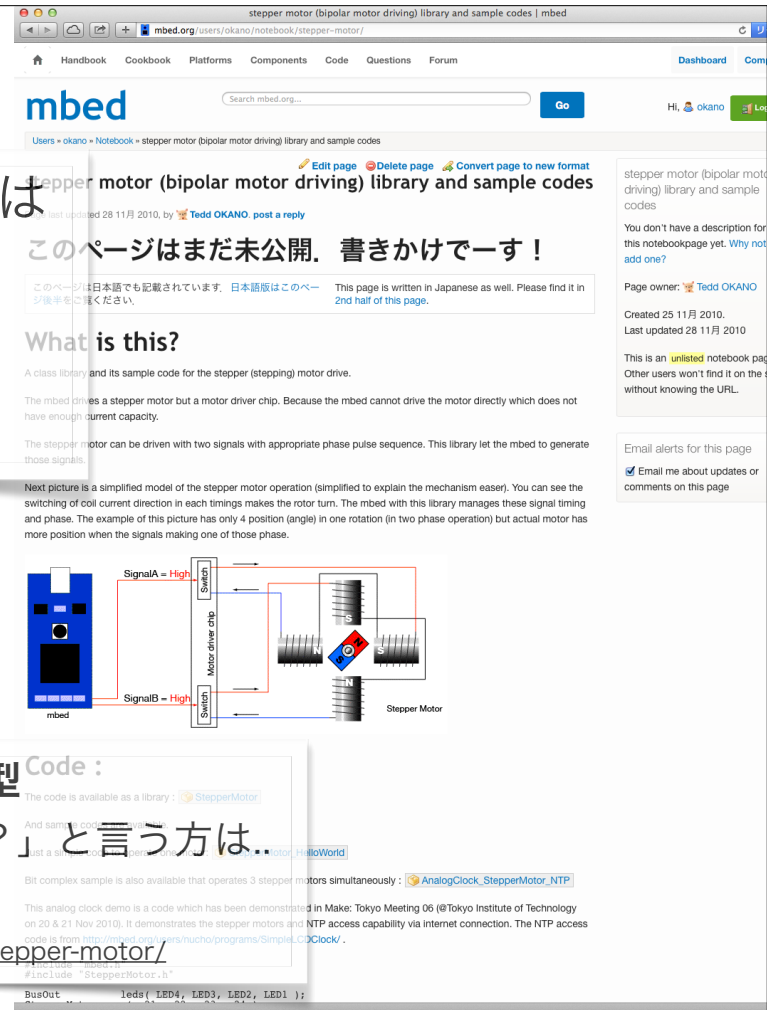
# ユニポーラ型 ステッピングモータ

入手が容易で簡単に使えるものを題材として選びました。  
「ユニポーラ型」のステッピングモータです。秋月電子で売っている小さなモータ

ステッピングモータには

- ・ユニポーラ型
- ・バイポーラ型

があります。



今回説明するのはユニポーラ型

「バイポーラ型ってどんなの？」という方は..

こちらをどうぞ

<http://mbed.org/users/okano/notebook/stepper-motor/>

ステッピングモータには「ユニポーラ型」と「バイポーラ型」があります。

今回の話はユニポーラ型の話だけ。

バイポーラモータについて興味があれば次のページを見てみてください

<http://mbed.org/users/okano/notebook/stepper-motor/>

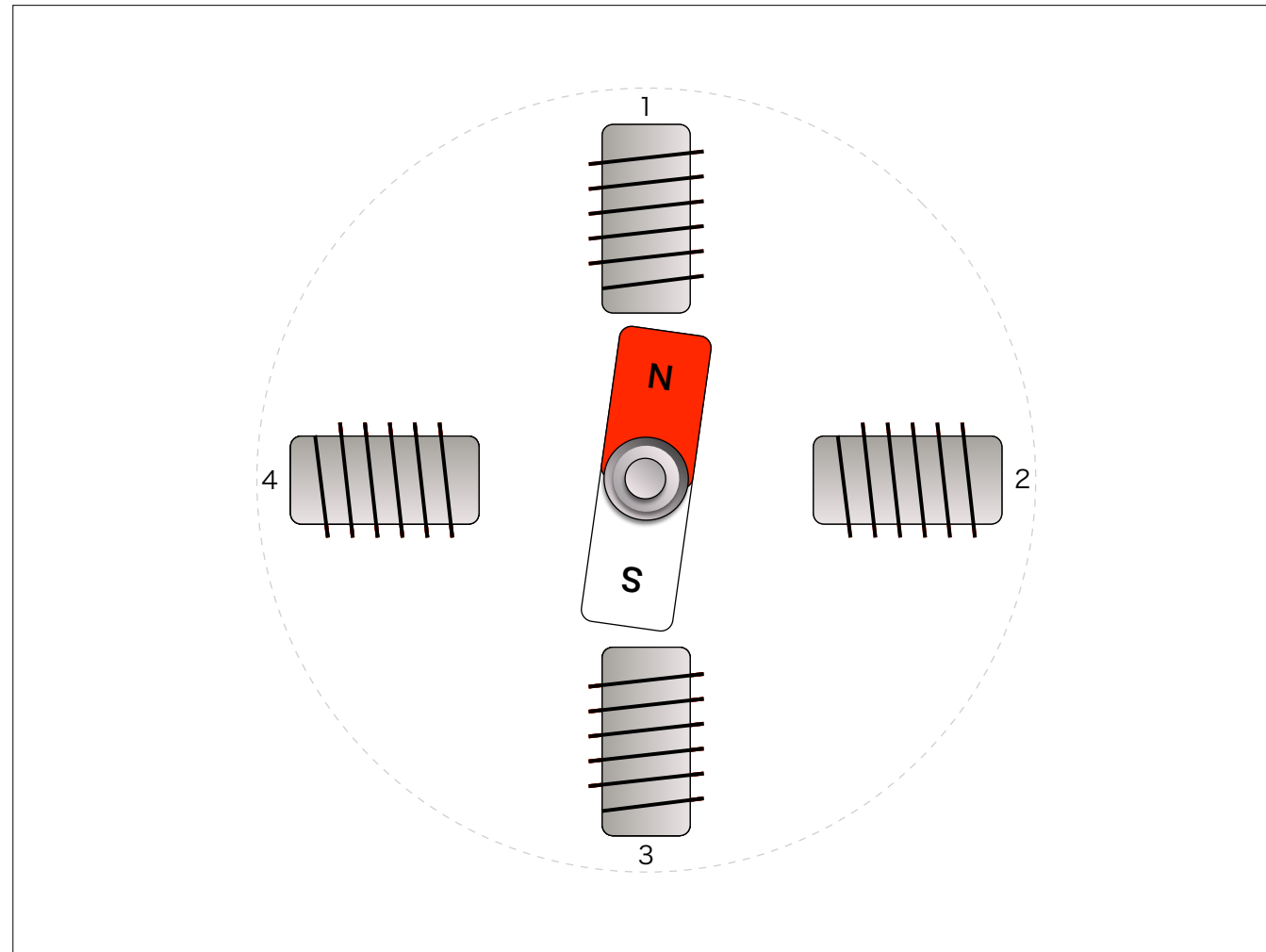


この他，大阪の共立電子さんでも面白そうなステッピングモータがたくさん。

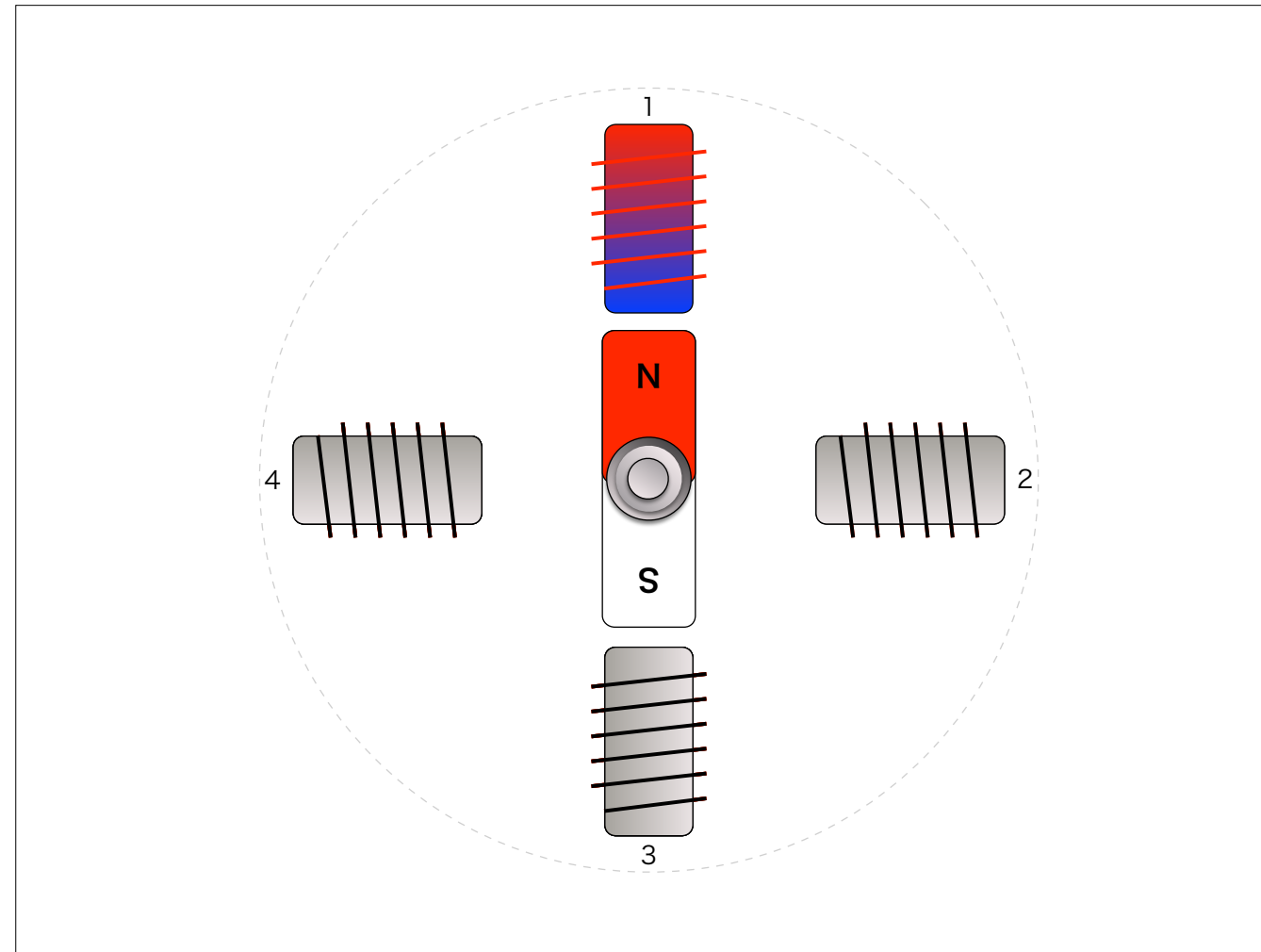
mbed祭りのデモにはこのスライドの一番下に表示されているPM25S-024も使ってみたりしています

( ー ー ? )

「ステッピングモータ」ってナニ

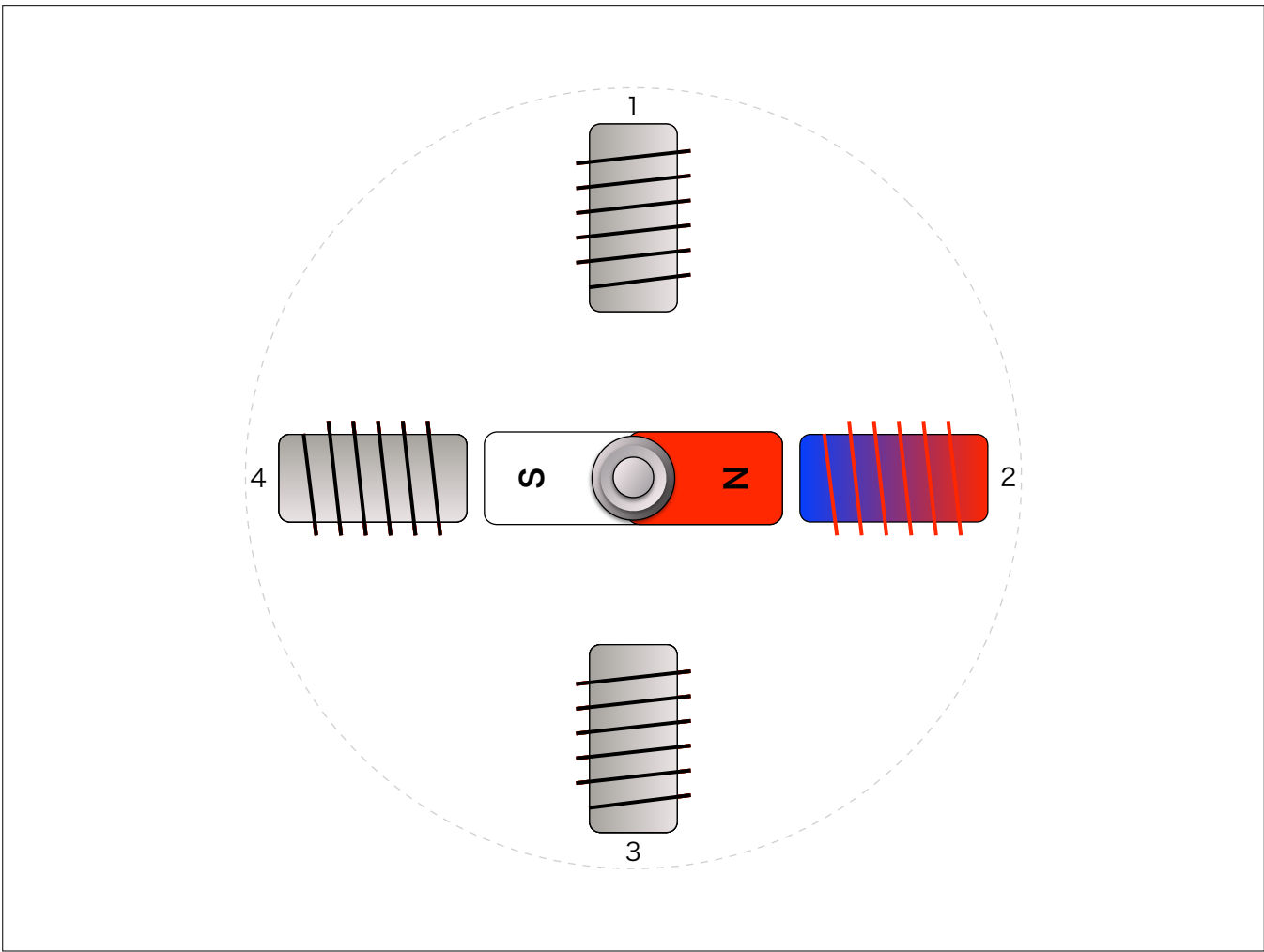


非常に簡略化した絵で，ステッピングモータの原理を説明します。  
まず4個のコイルが周りに置かれてあり，中に回転する磁石が置かれています

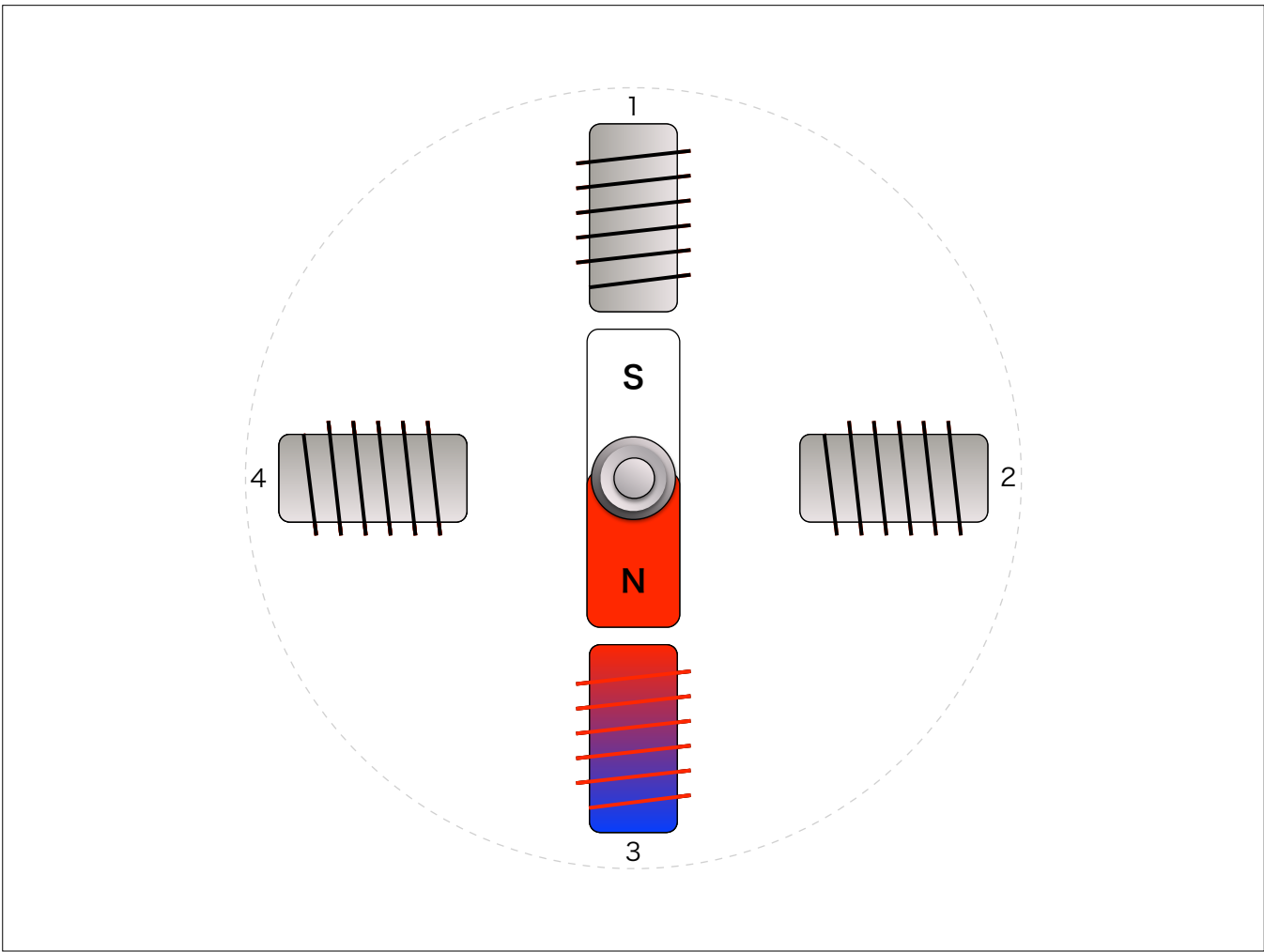


コイルのひとつに電流を流せば，磁石となって中の磁石が引き寄せられます.  
電流を流してコイルを磁石にすることを「励磁(れいじ)」といいます

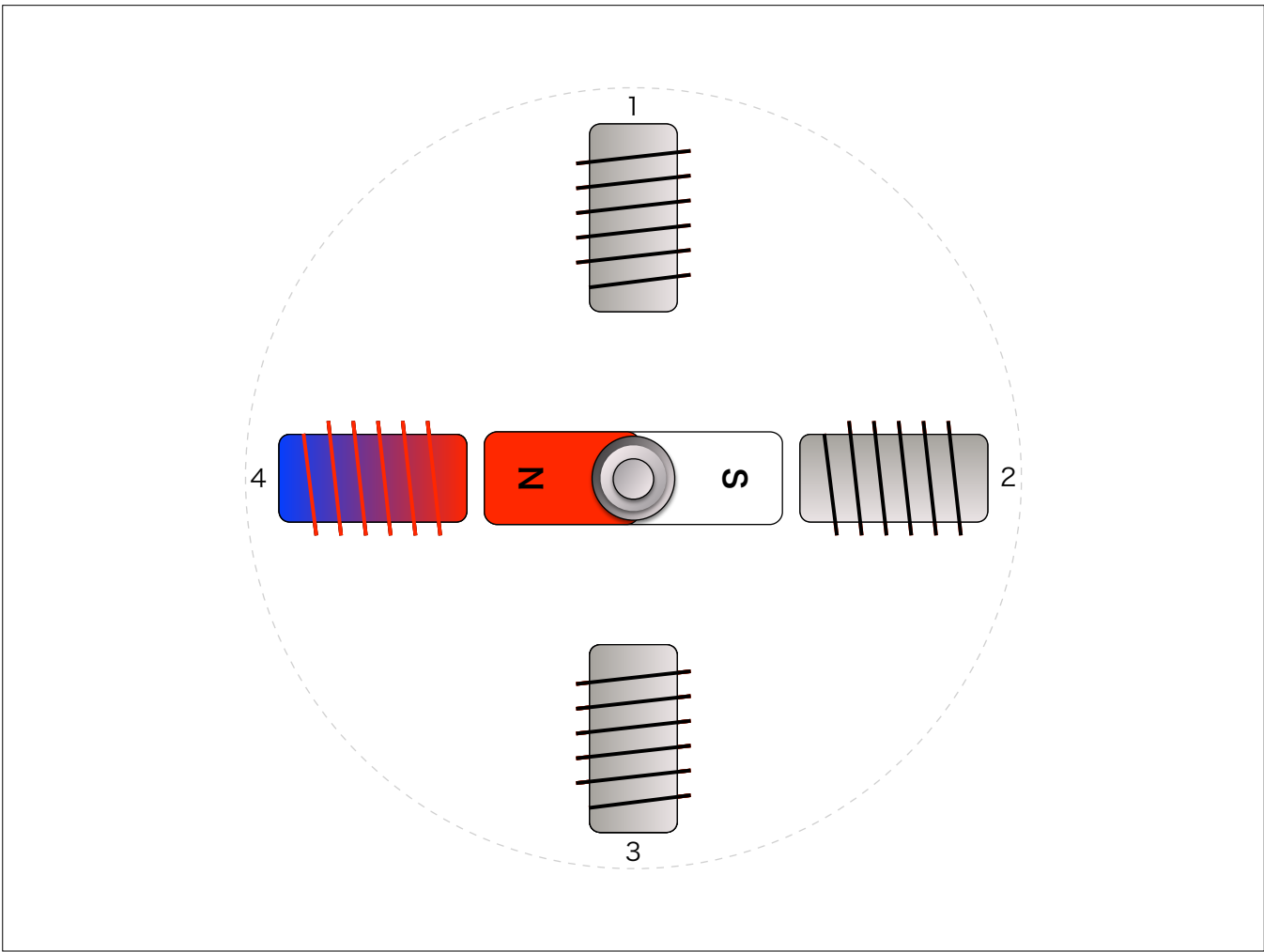




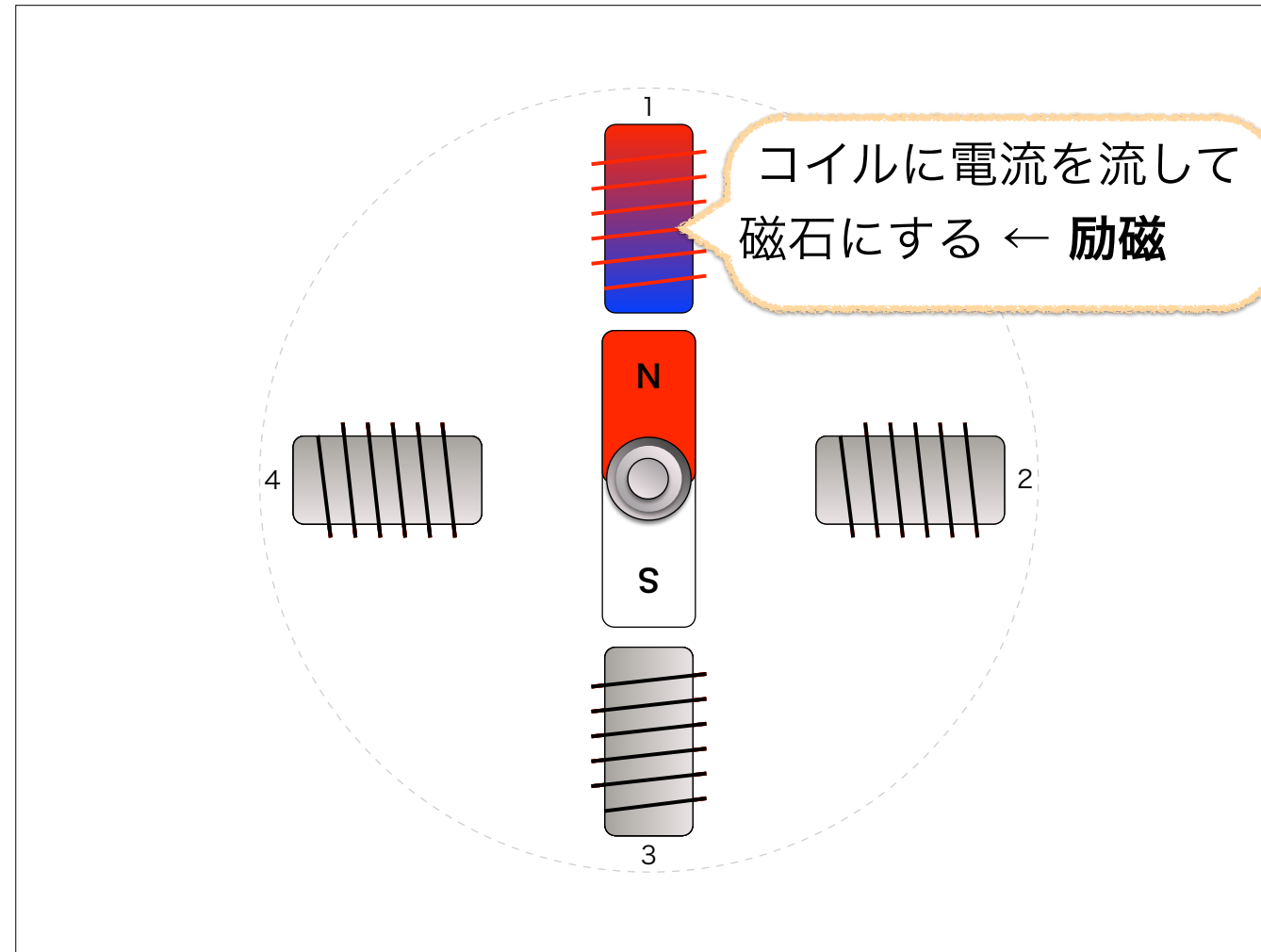
2番のコイルを励磁



3番のコイルを励磁



4番のコイルを励磁



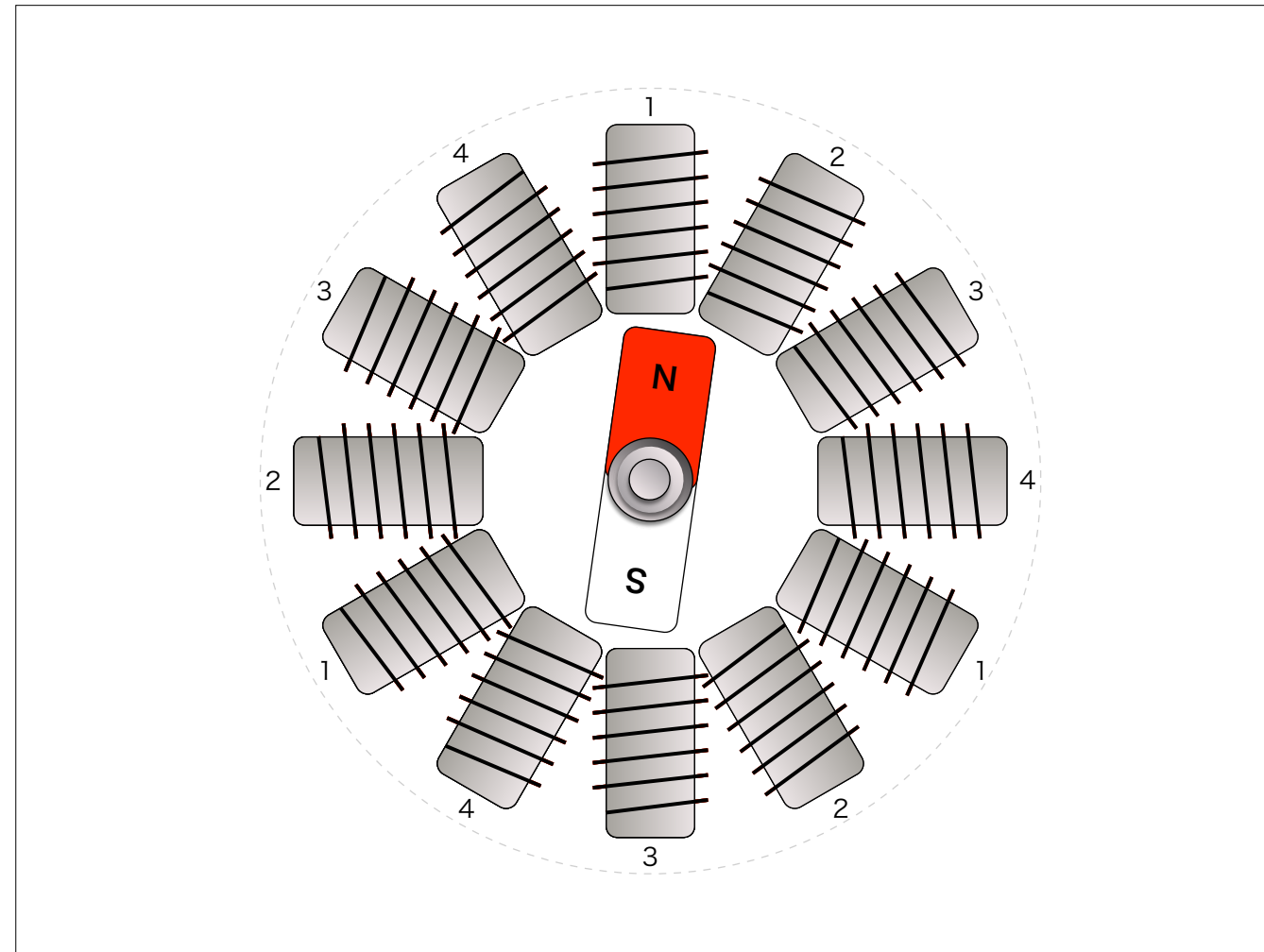
1→2→3→4→1の順でコイルを励磁すると中の磁石が回ります

1→4→3→2→1の順でコイルを励磁すると逆回転

## ステップで動くからステッピングモーター

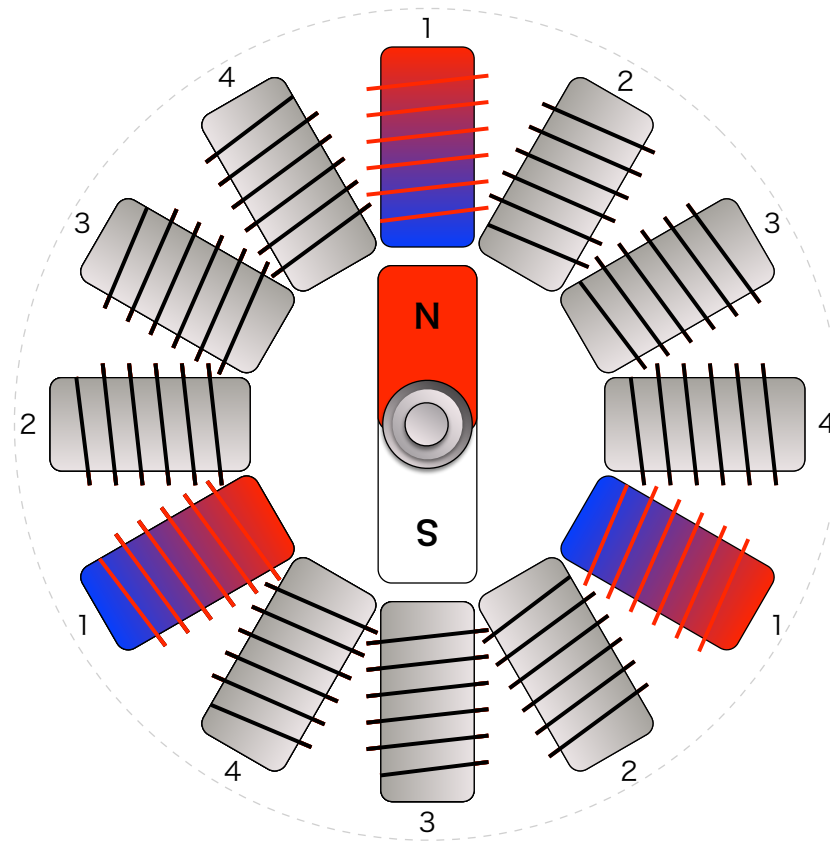
コイルを励磁させるたびに磁石がその分がけ動く←ステップ動作.  
ステップ動作するモータだから「ステッピングモータ」

$(\nabla^\circ)$   
なるほど！



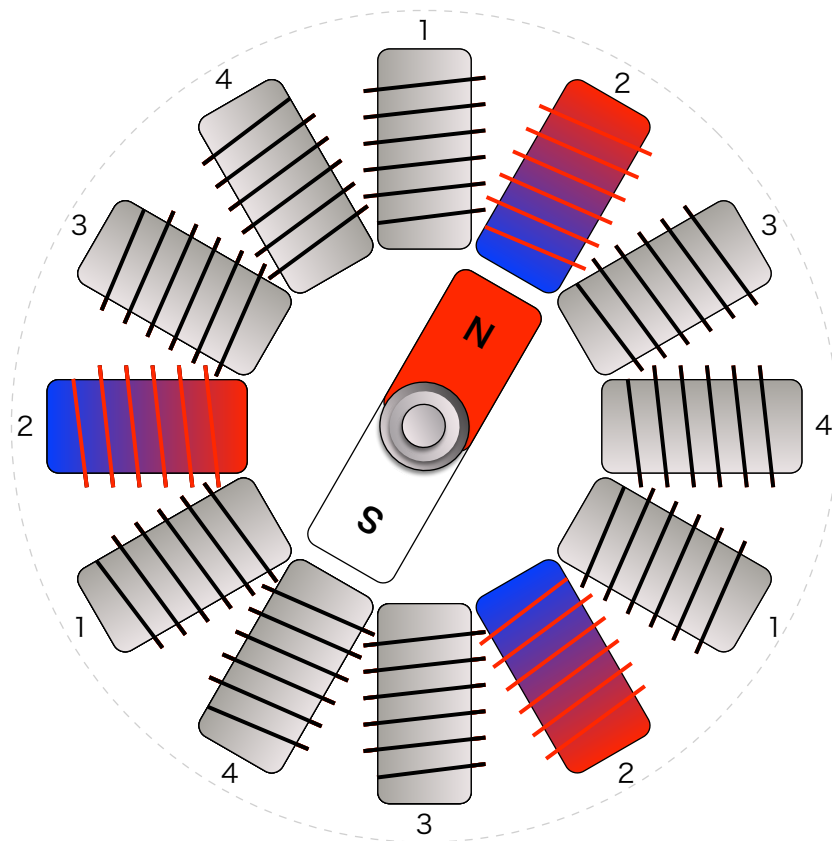
実際のモータにはもっとたくさんのコイルが入っています.

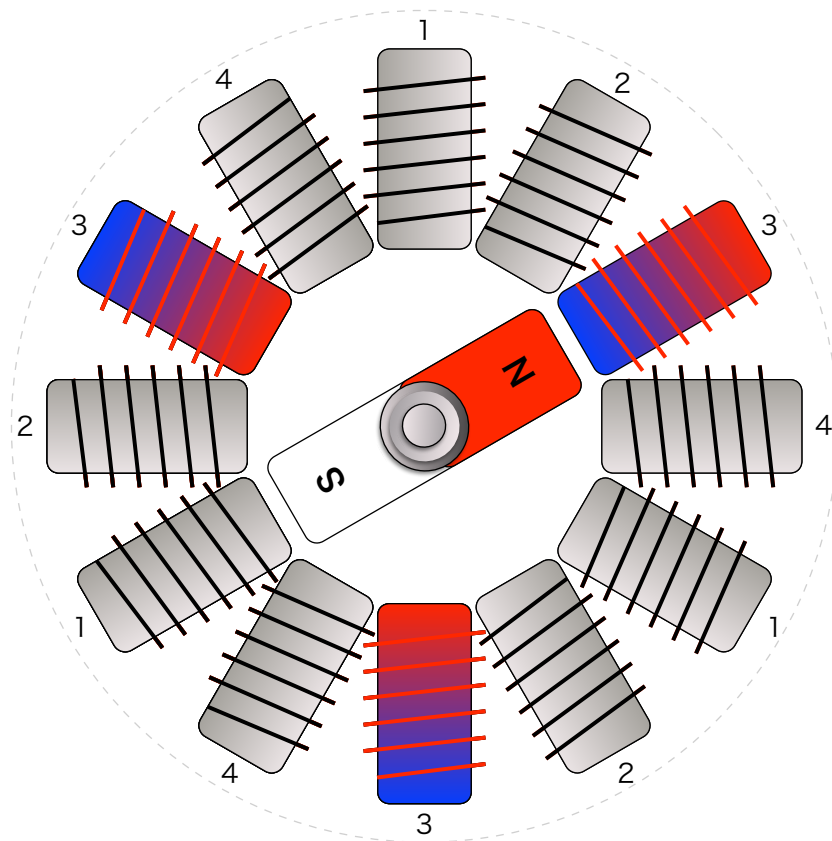
でも基本はさっきのと同じ. 4個で1セットのコイルが数セット(このスライドの例では3セット)入っています.

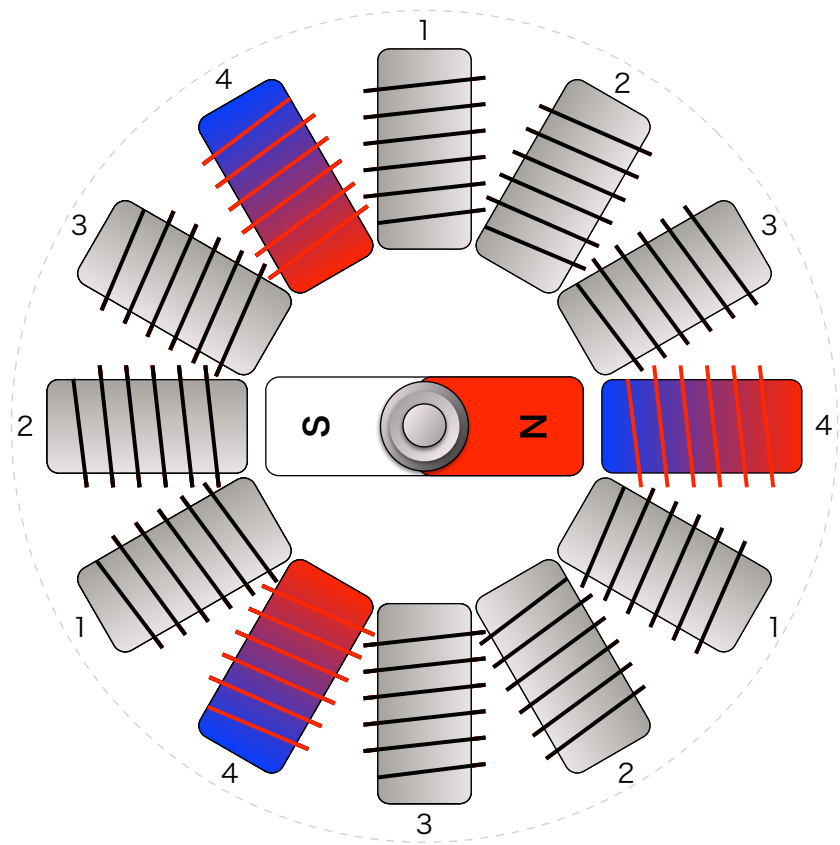


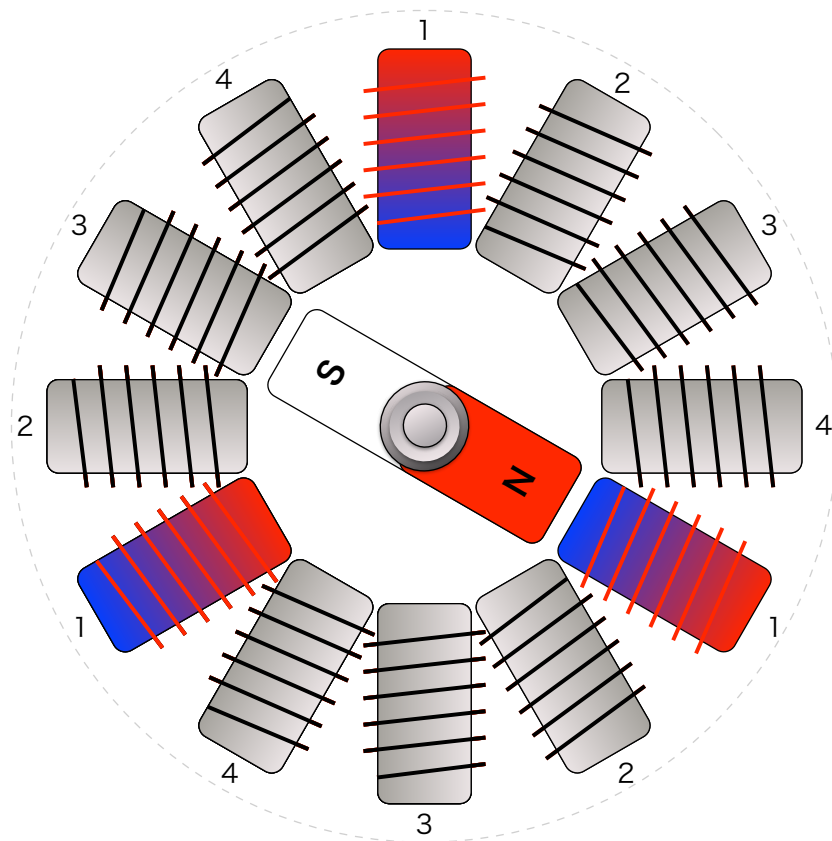
これも1→2→3→4→1→2→3→4のように励磁します

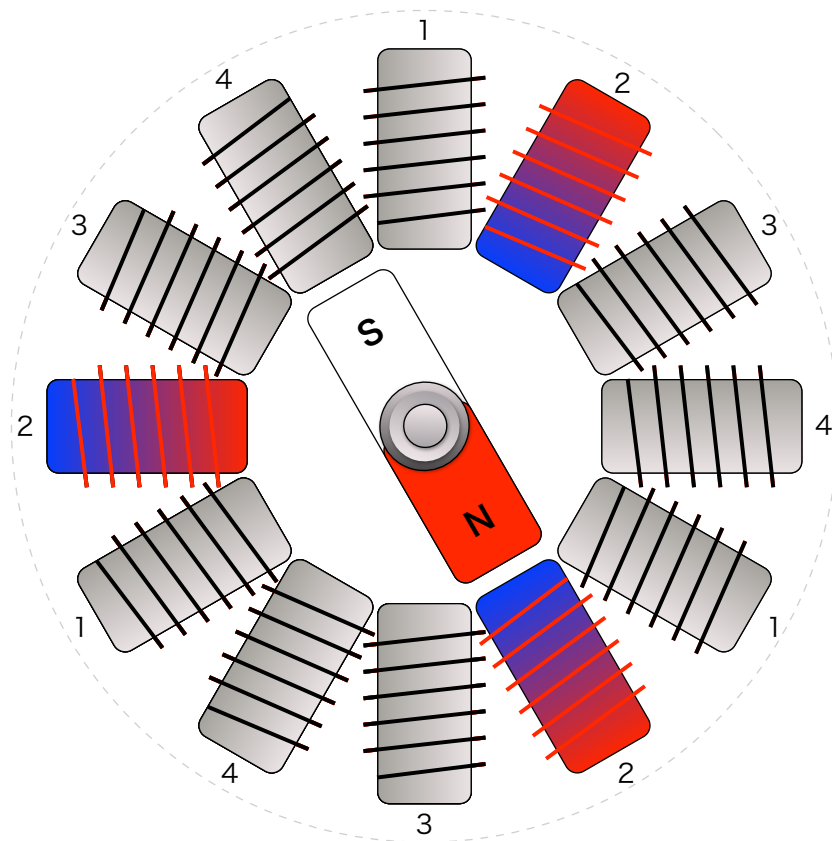


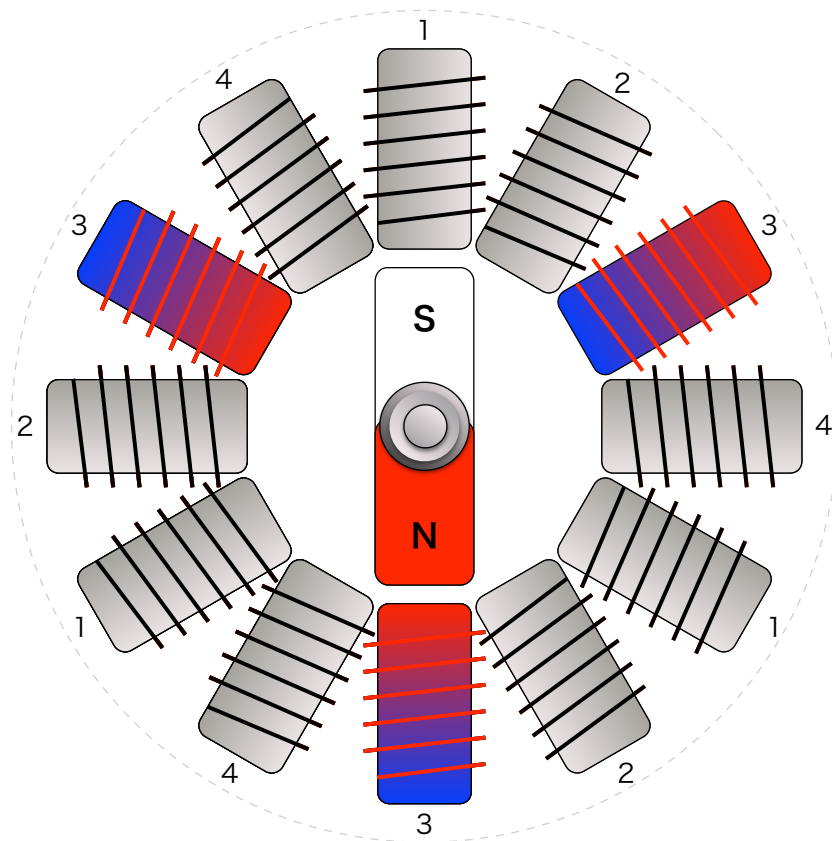


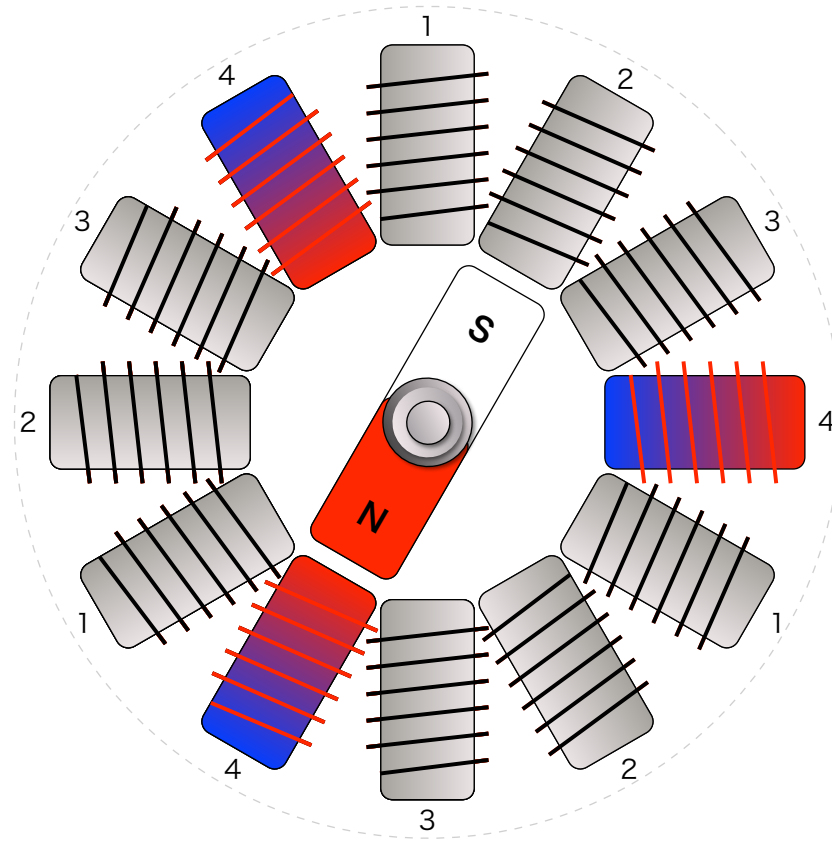


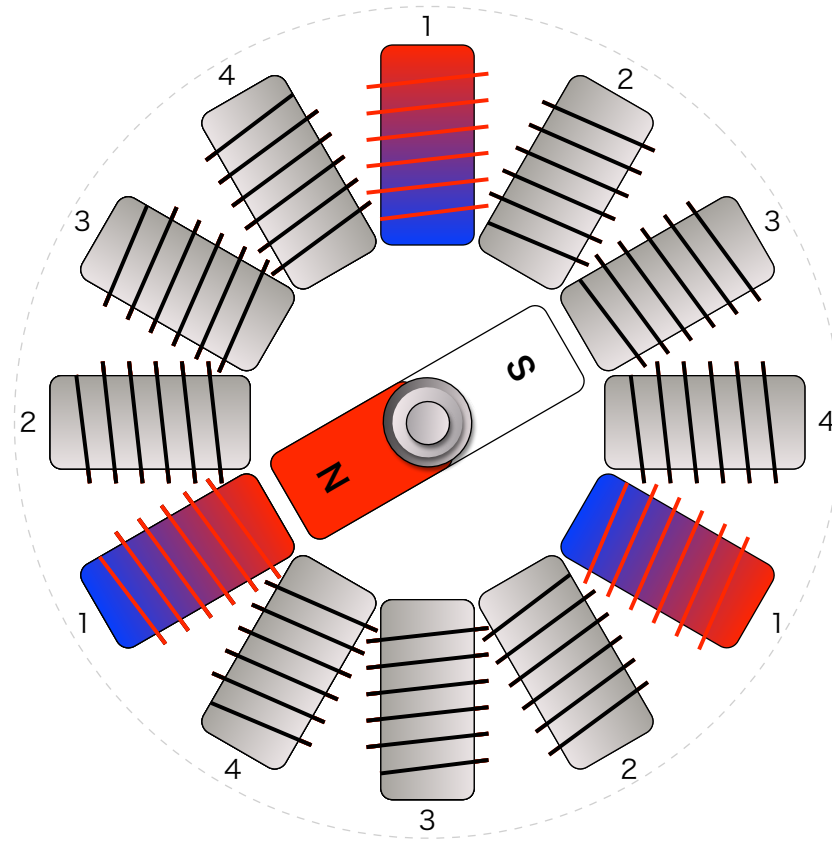




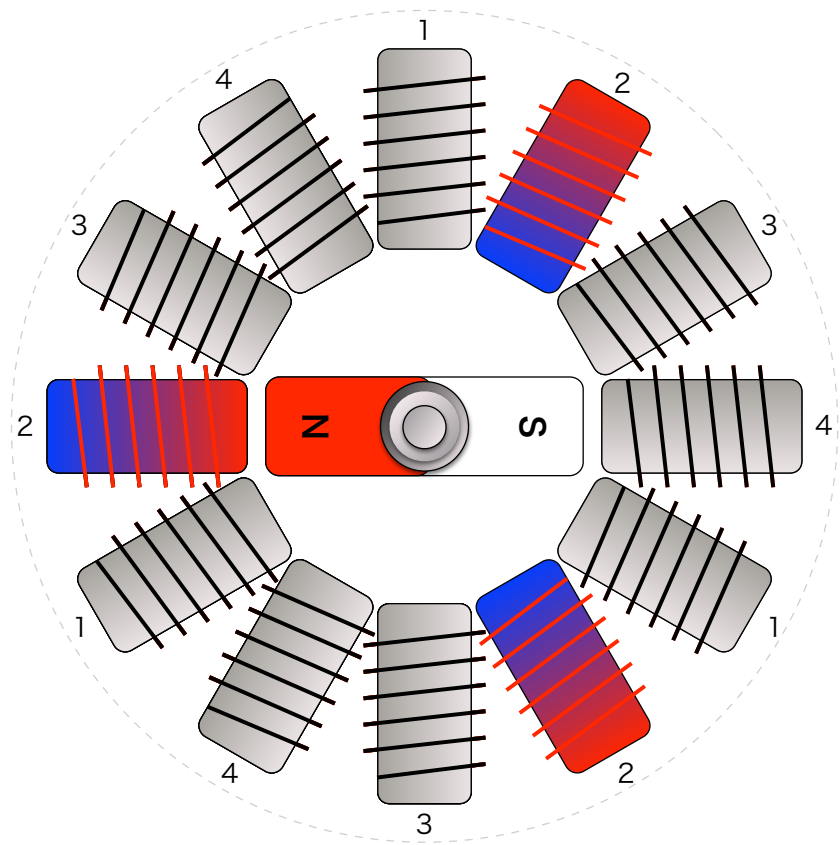


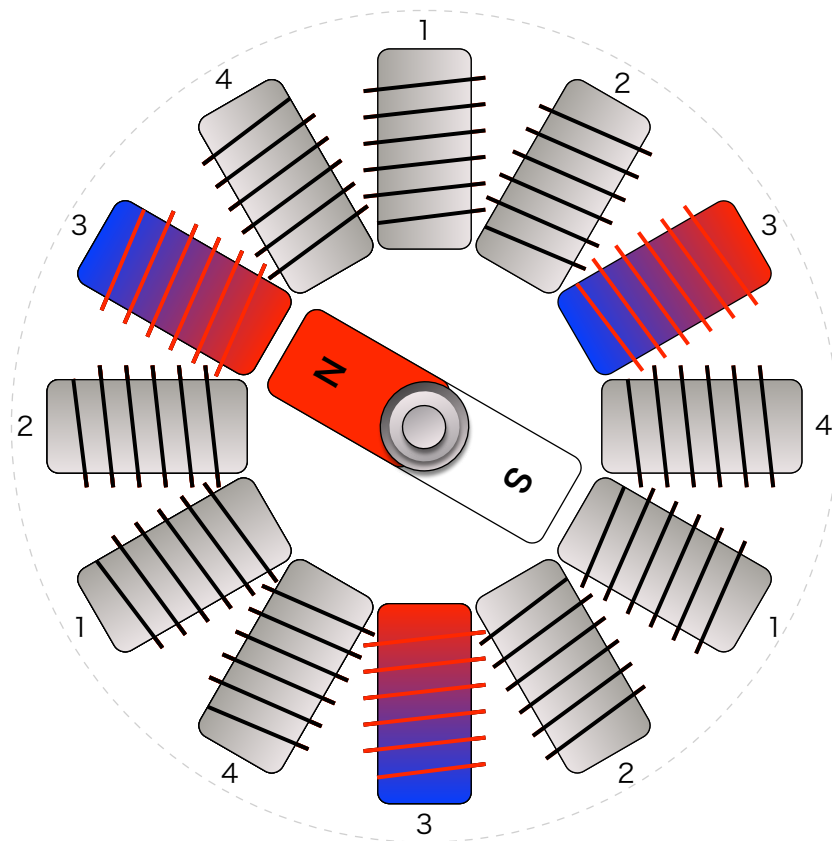


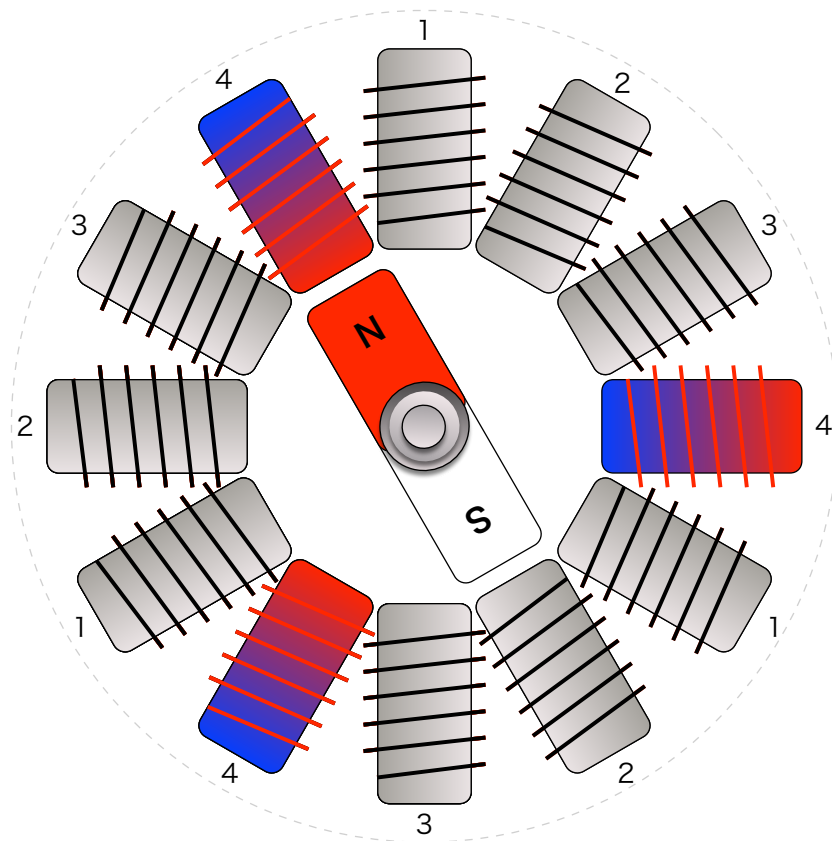


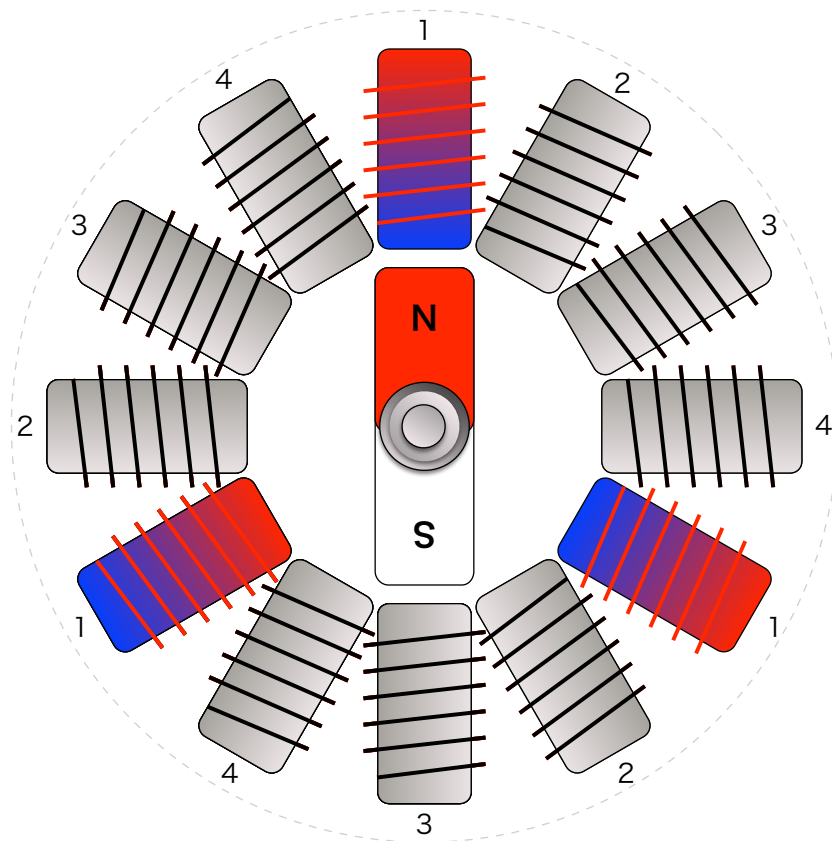


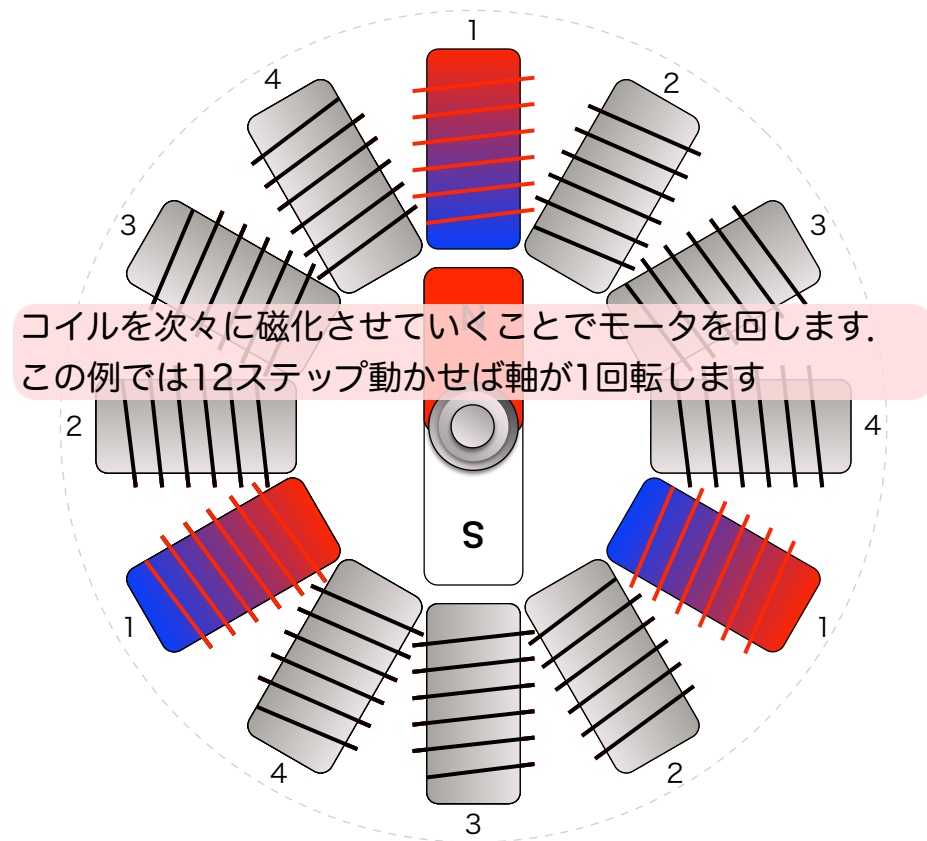




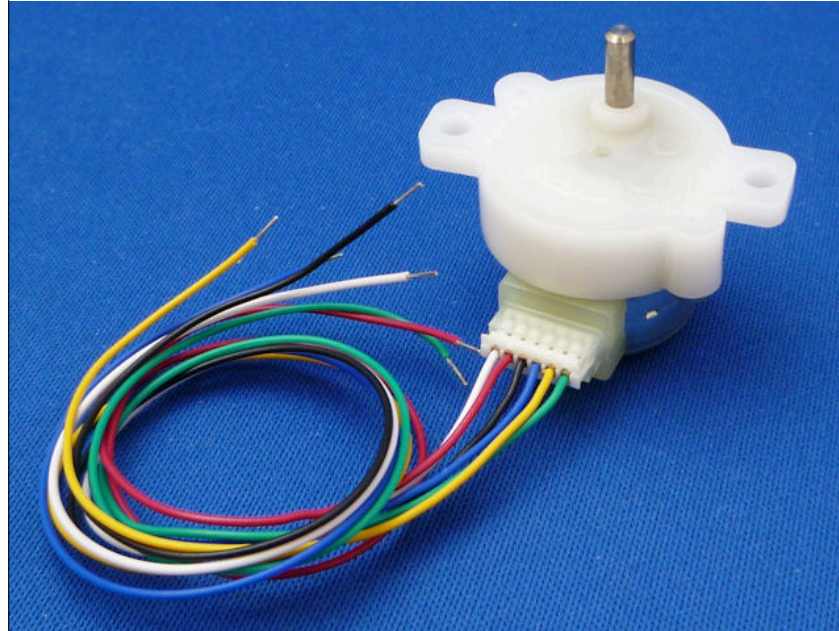








この例では12ステップで1回転



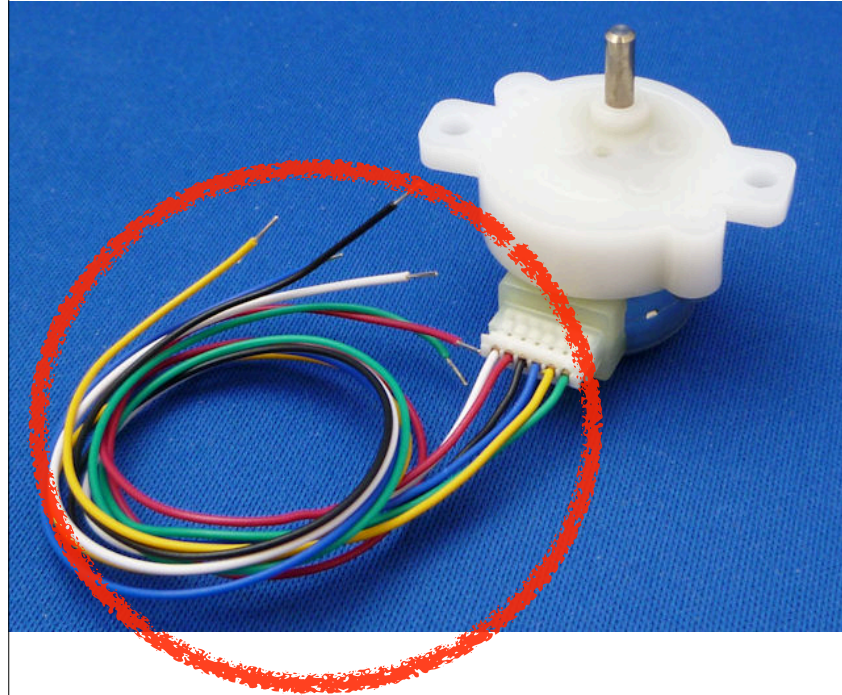
このモータでは内部に  
20ステップ/回転のモータが  
入っており、さらに1/18比  
のギアを介して軸が回るため  
360ステップで軸が1回転し  
ます

1ステップ=1°の動作

ここで試してみるモータは20ステップで1回転のモータ。しかしモータと軸の間には18:1のギアが入ってるので、この軸を1回転させるには360ステップ必要になります

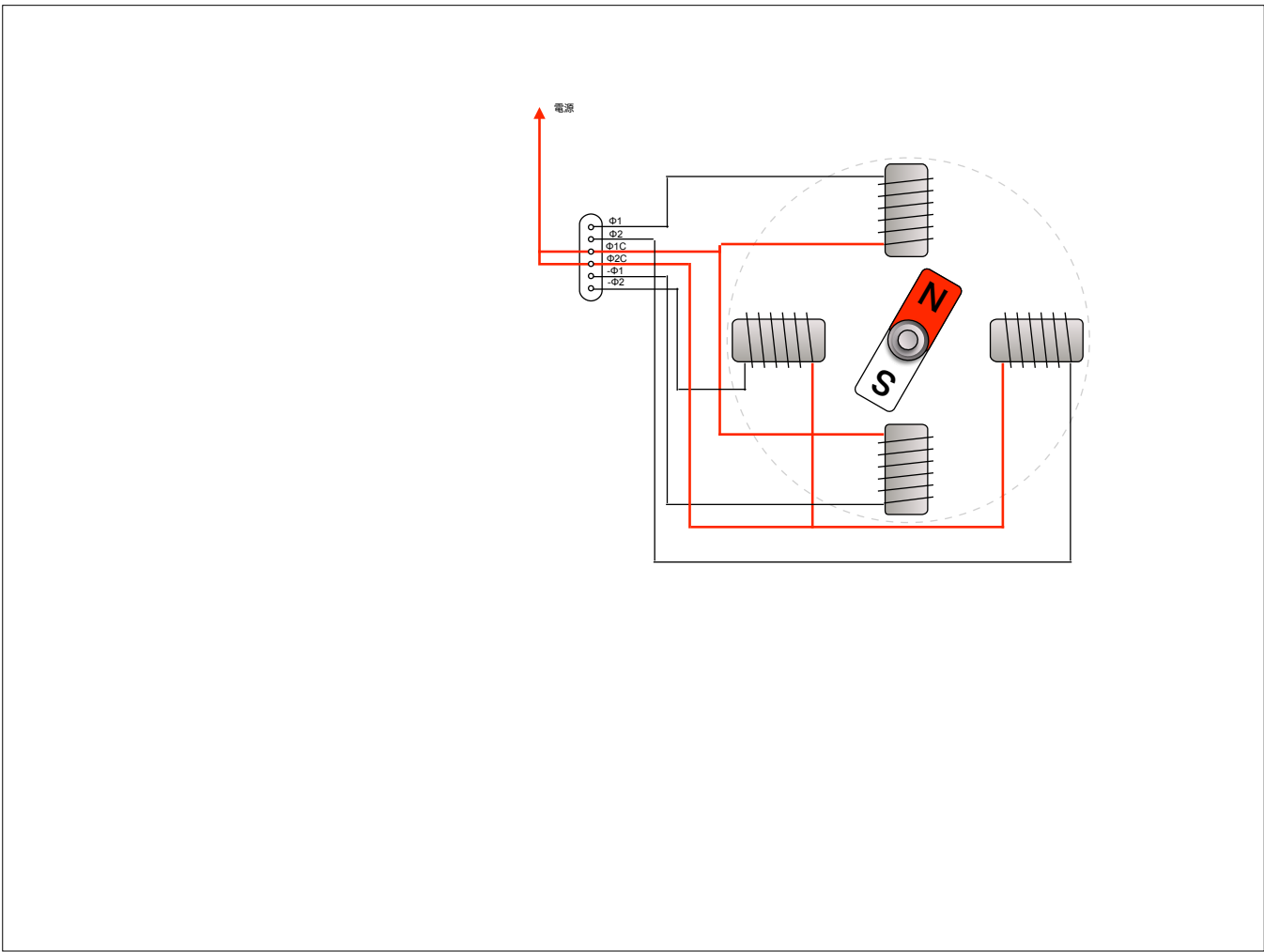
┐（´～`；）┐

線がいっぱい出てるけど？

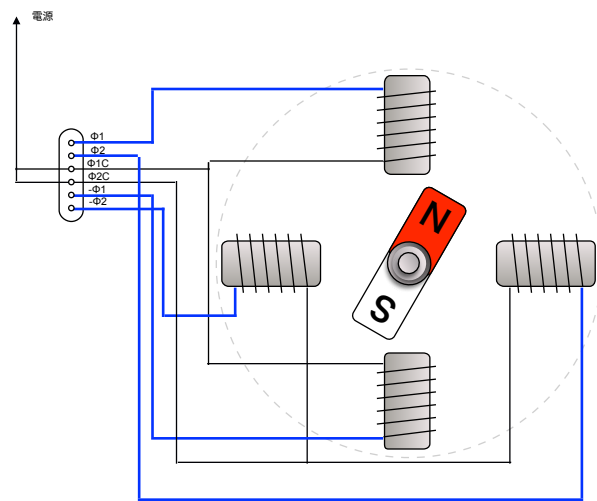


模型用のDCモータなら出ている線は2本だけ。電池の(+)と(-)に繋ぐだけで回ります。  
こんなのどうやって繋ぐのでしょうか？

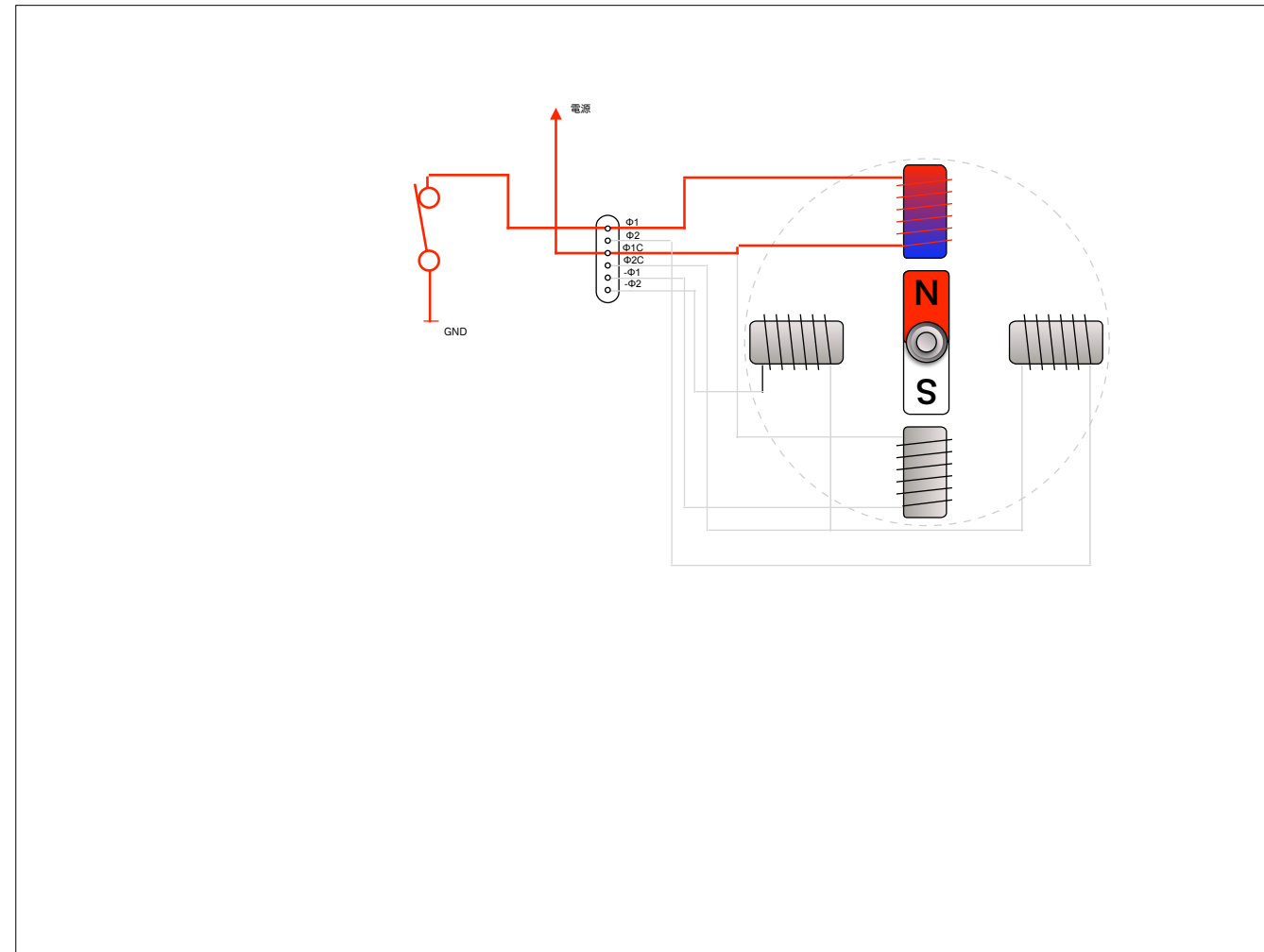




じつは内部は簡単な配線になっています。このモータでは6本のうち2本は共通端子として電源に繋ぐようにした線が2本

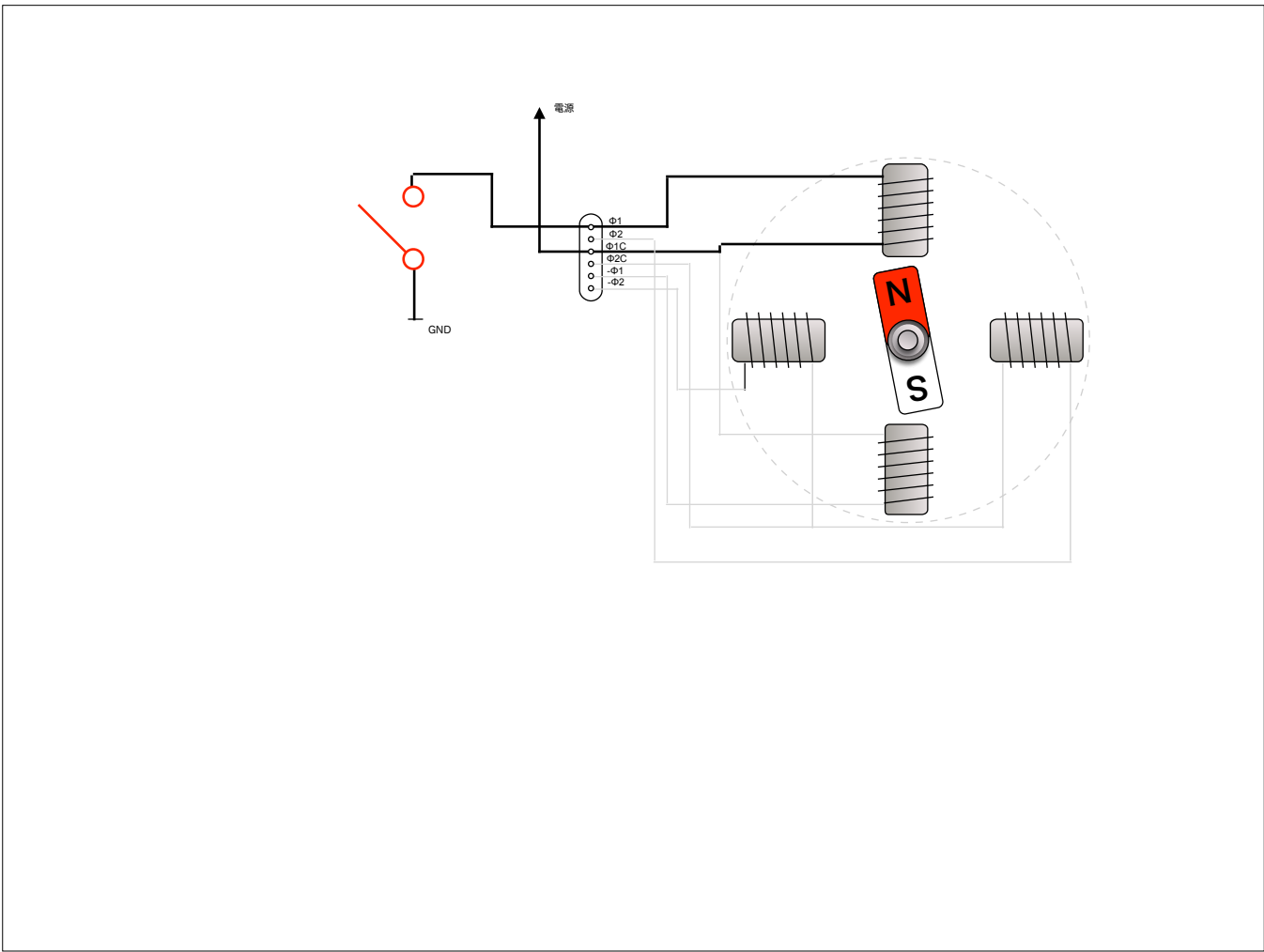


各コイルの反対端の4本がそれぞれの端子に出ています



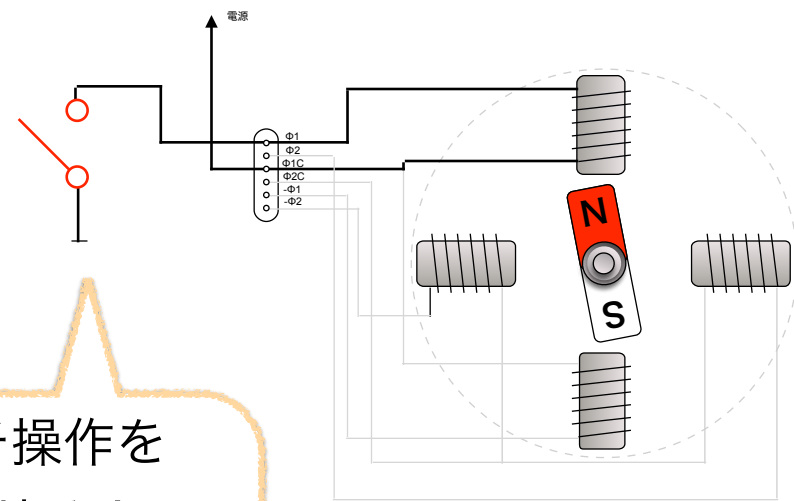
この共通端子を電源に繋いで、反対端を**GND**につなげば電流が流れて励磁されます。

ちなみにこの図の「電源」は電源の(+)側端子に、**GND**(グランド)は電源の0V側の端子に繋ぐことを省略した記号です



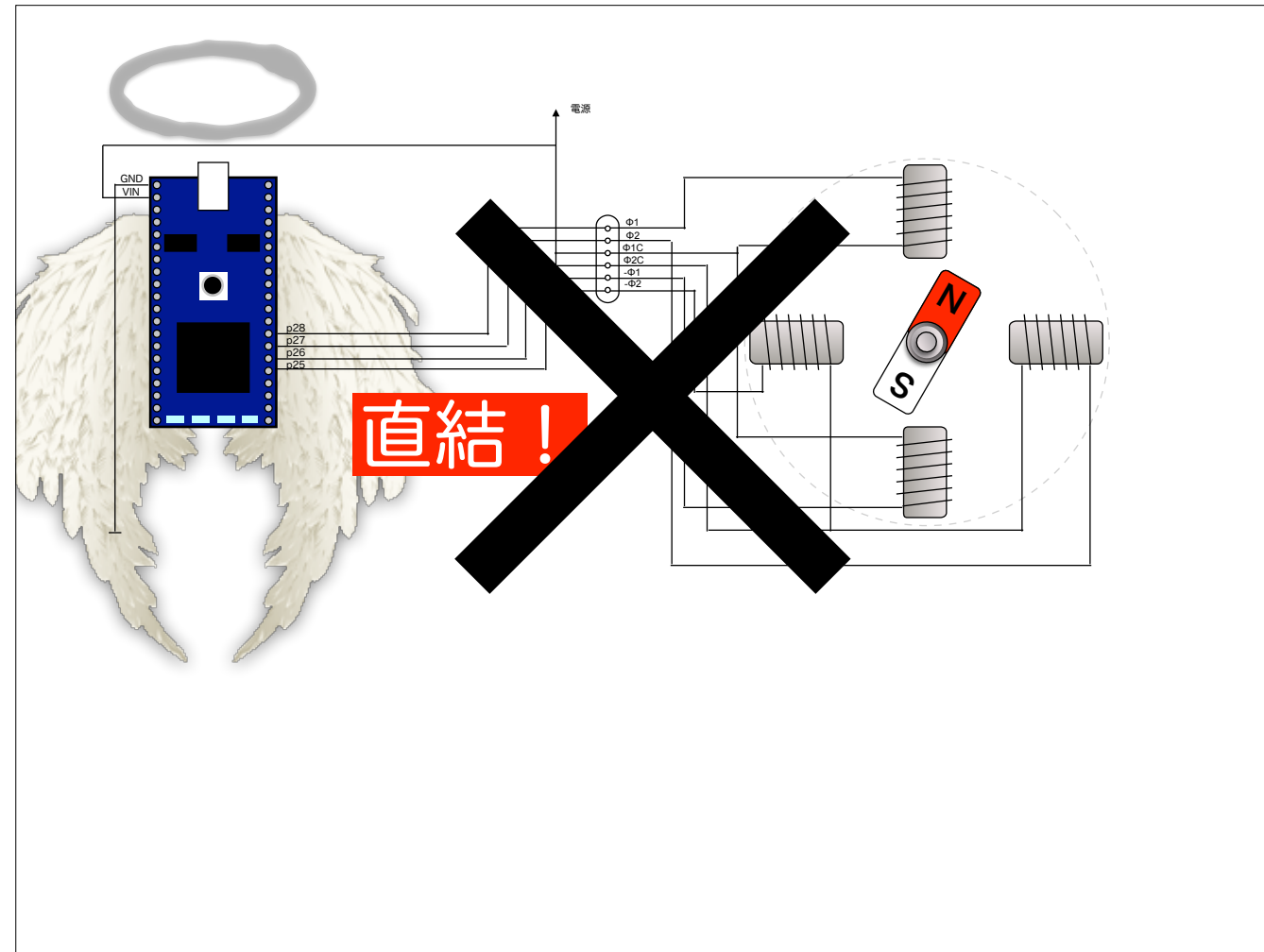
このGND側にスイッチを付けてON←→OFFできるようにしてやります

このスイッチ操作を  
mbedにやらせる！

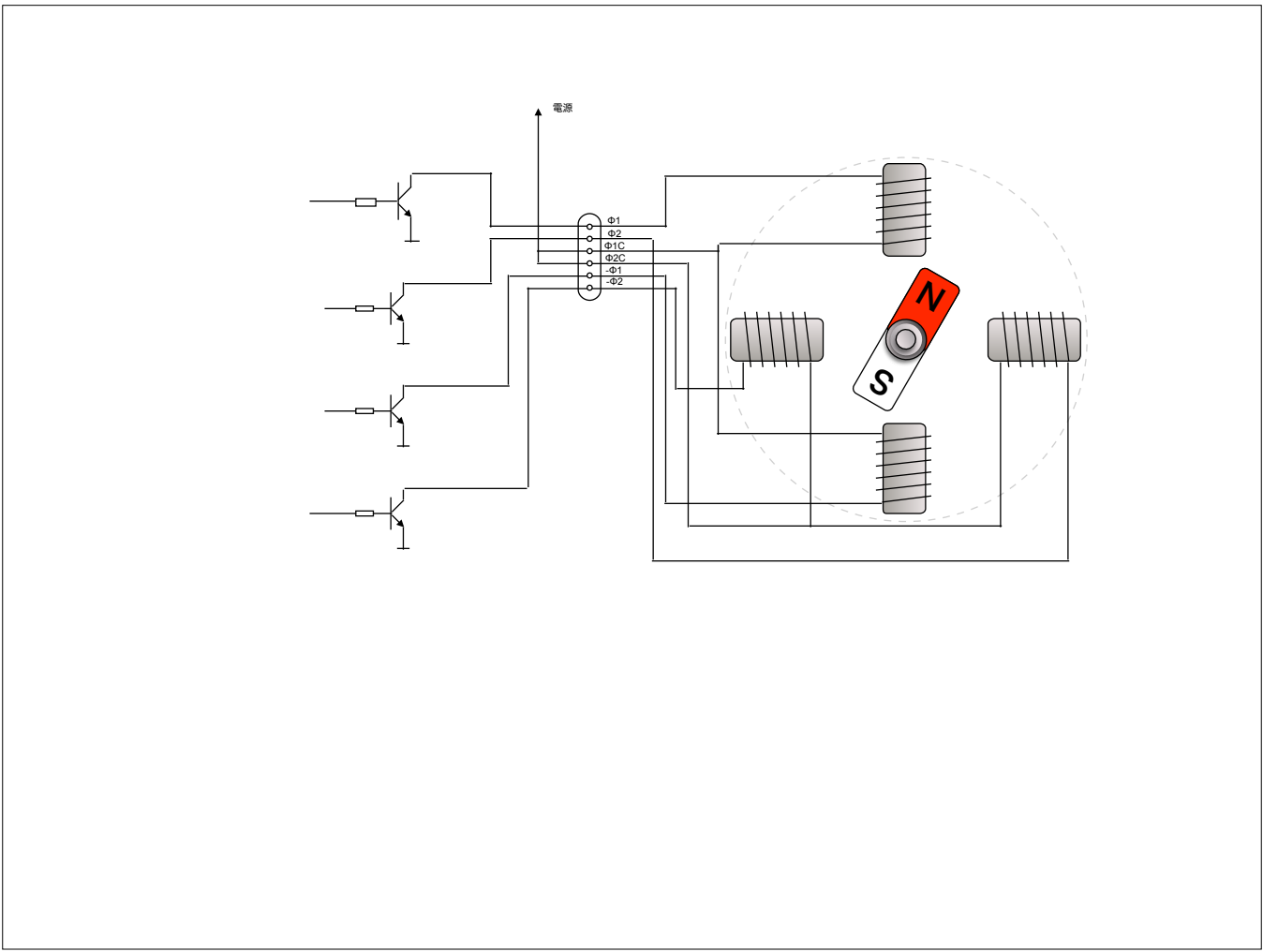


(<sup>◦</sup>∀<sup>◦</sup>)!

それなら簡単

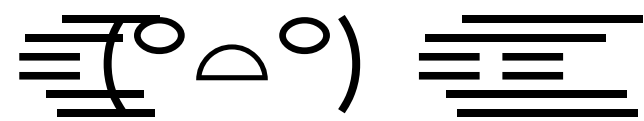


簡単ですがmbedを直接繋いではいけません。モータのコイルには大きな電流が流れます。  
mbedのピンはマイコンのピンに直結。大きな電流を流してしまうと壊れてしまいます。  
ちなみにmbedのデジタル出力ピンは4mAまでしか流せません

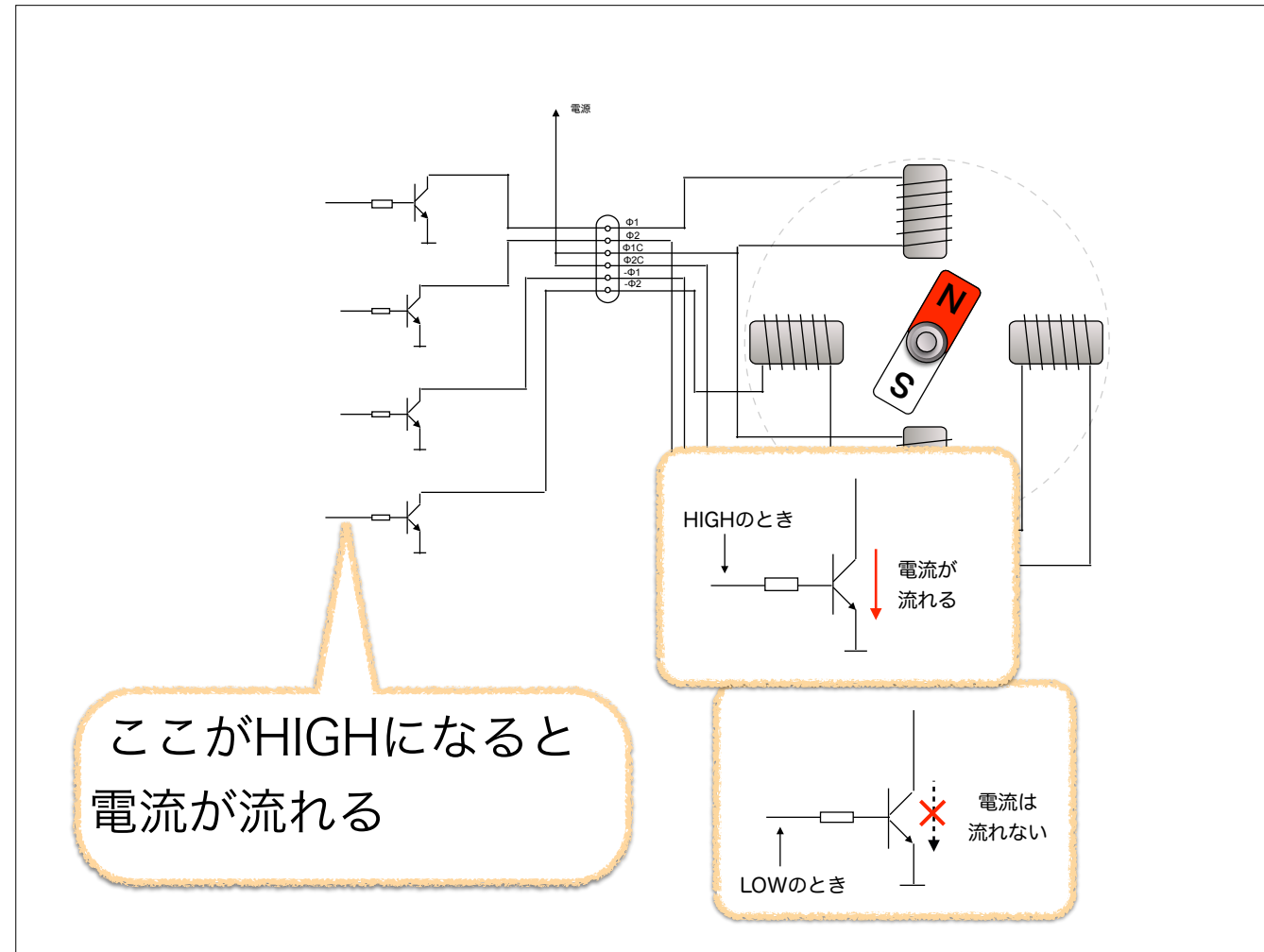


そこで、こんな回路を用意します



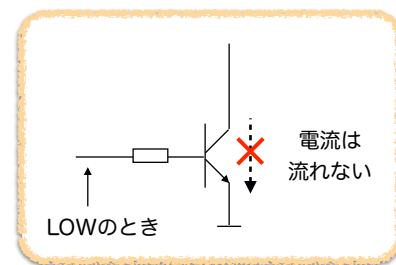
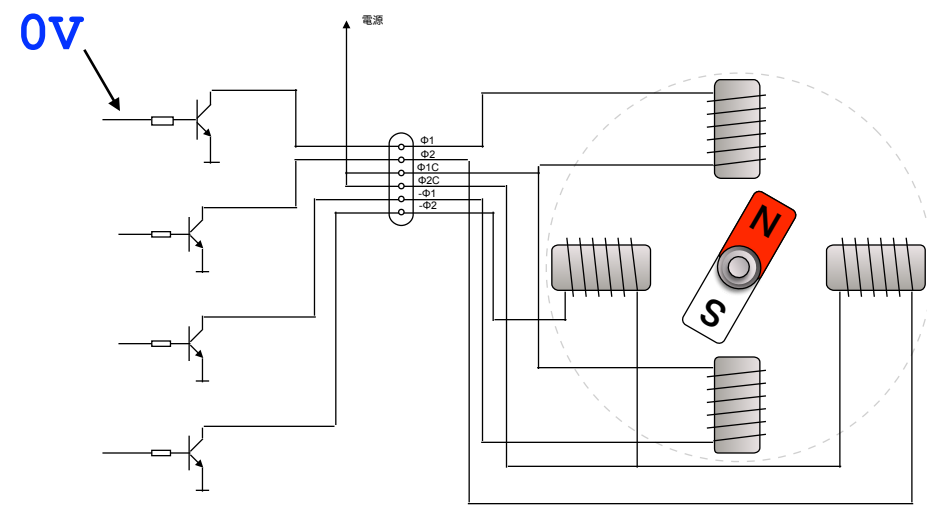


なにこれ



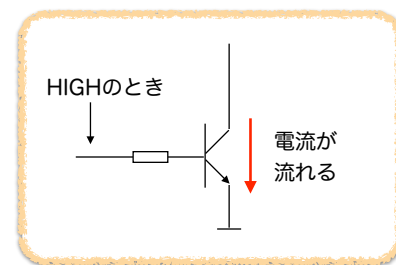
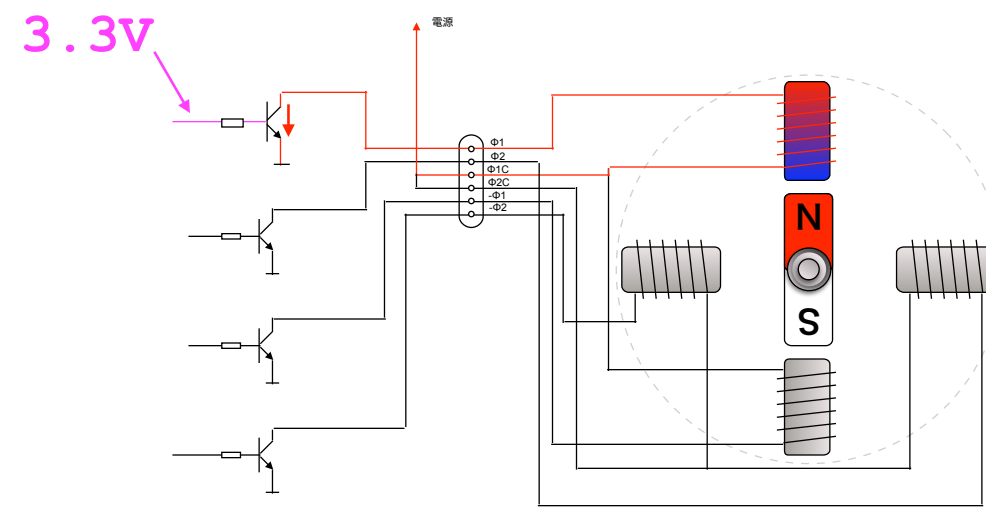
トランジスタをスイッチとして使います.

トランジスタの左側の線にHIGH(3.3Vの電圧)をかけるとトランジスタに電流が流れ、  
0Vにすると電流が流れなくなります.



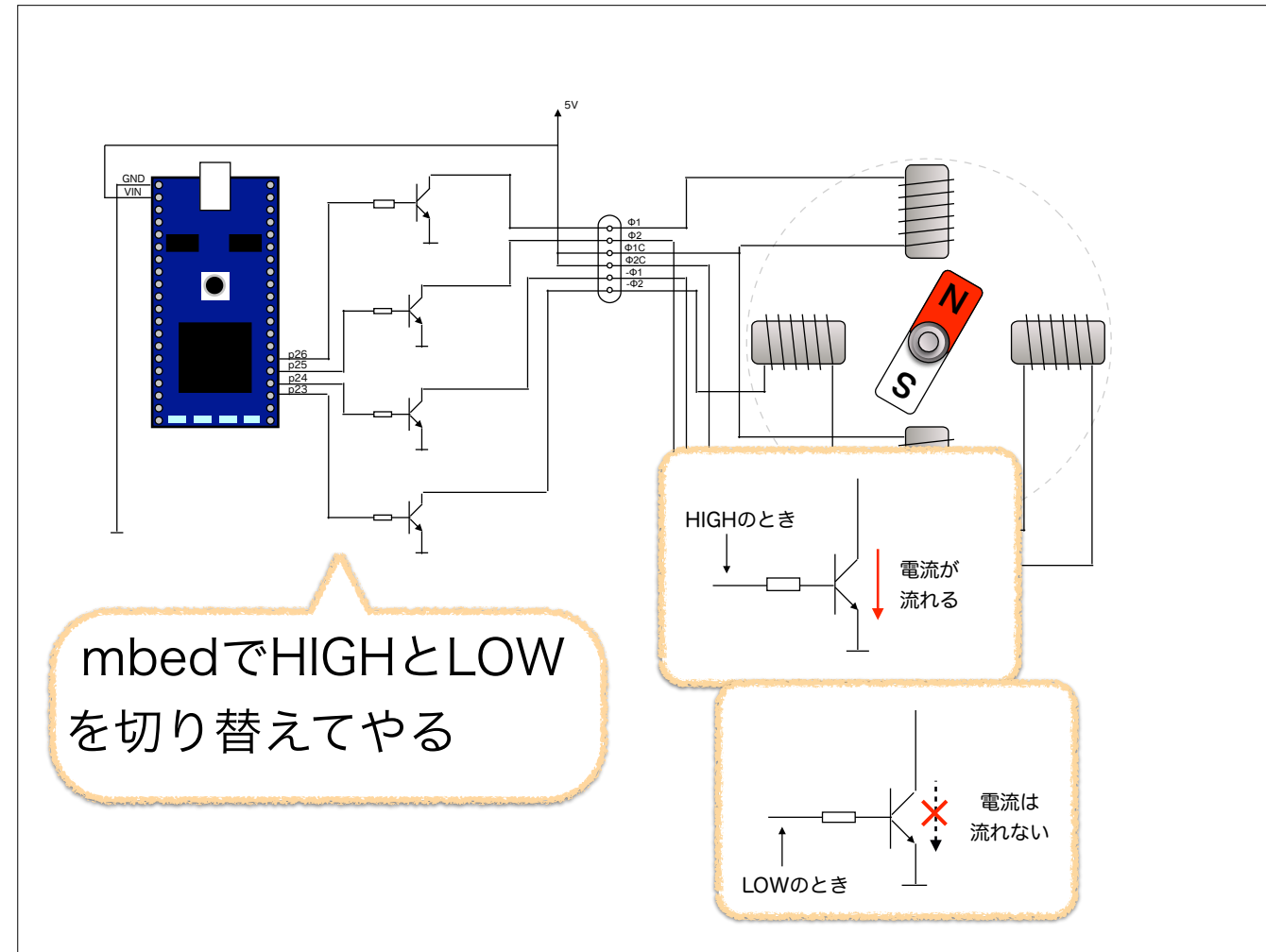
励磁なし

スイッチ(トランジスタ)がOFF状態



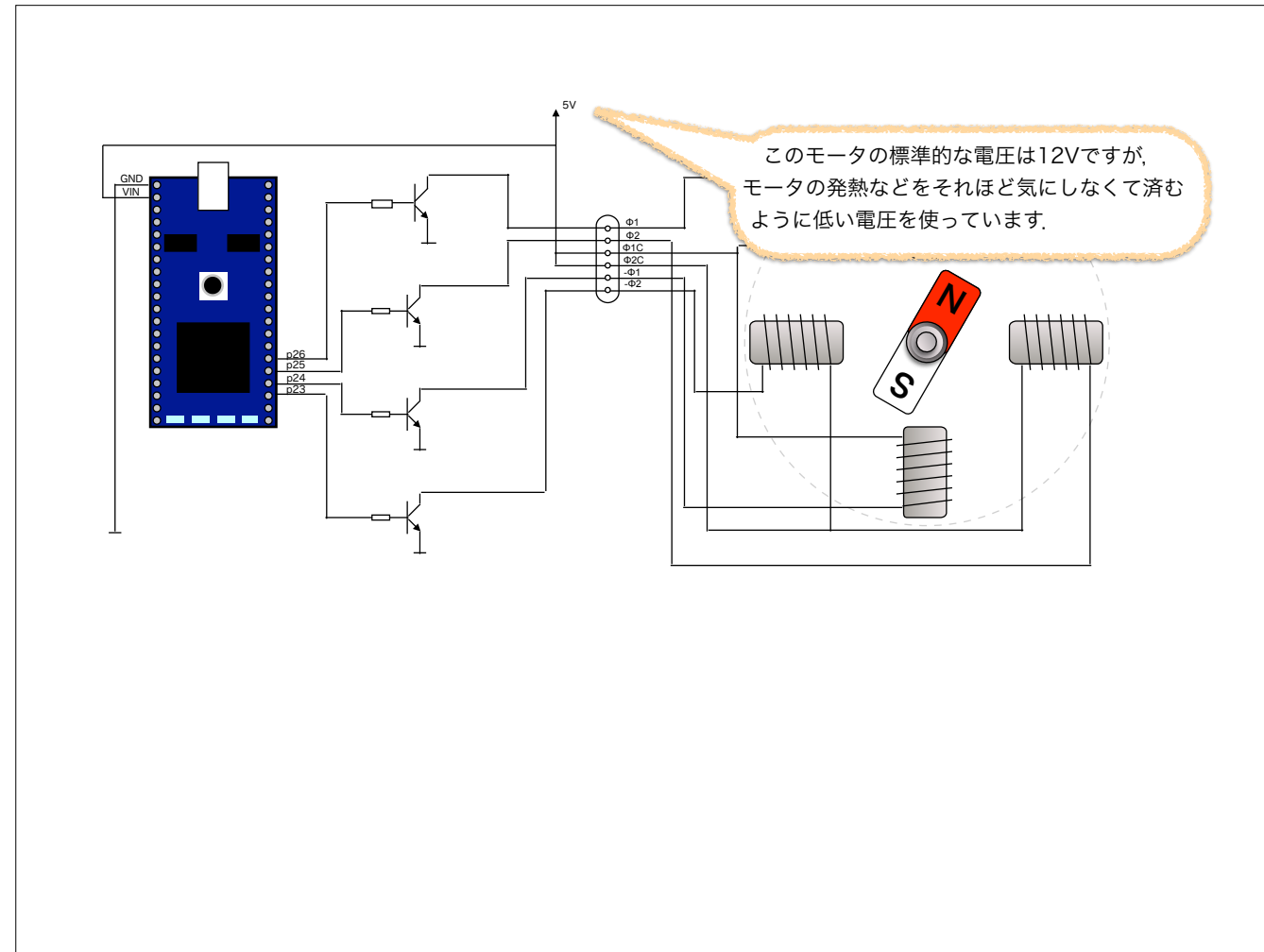
励磁

スイッチ(トランジスタ)がON状態



まとめるとこんな回路になります

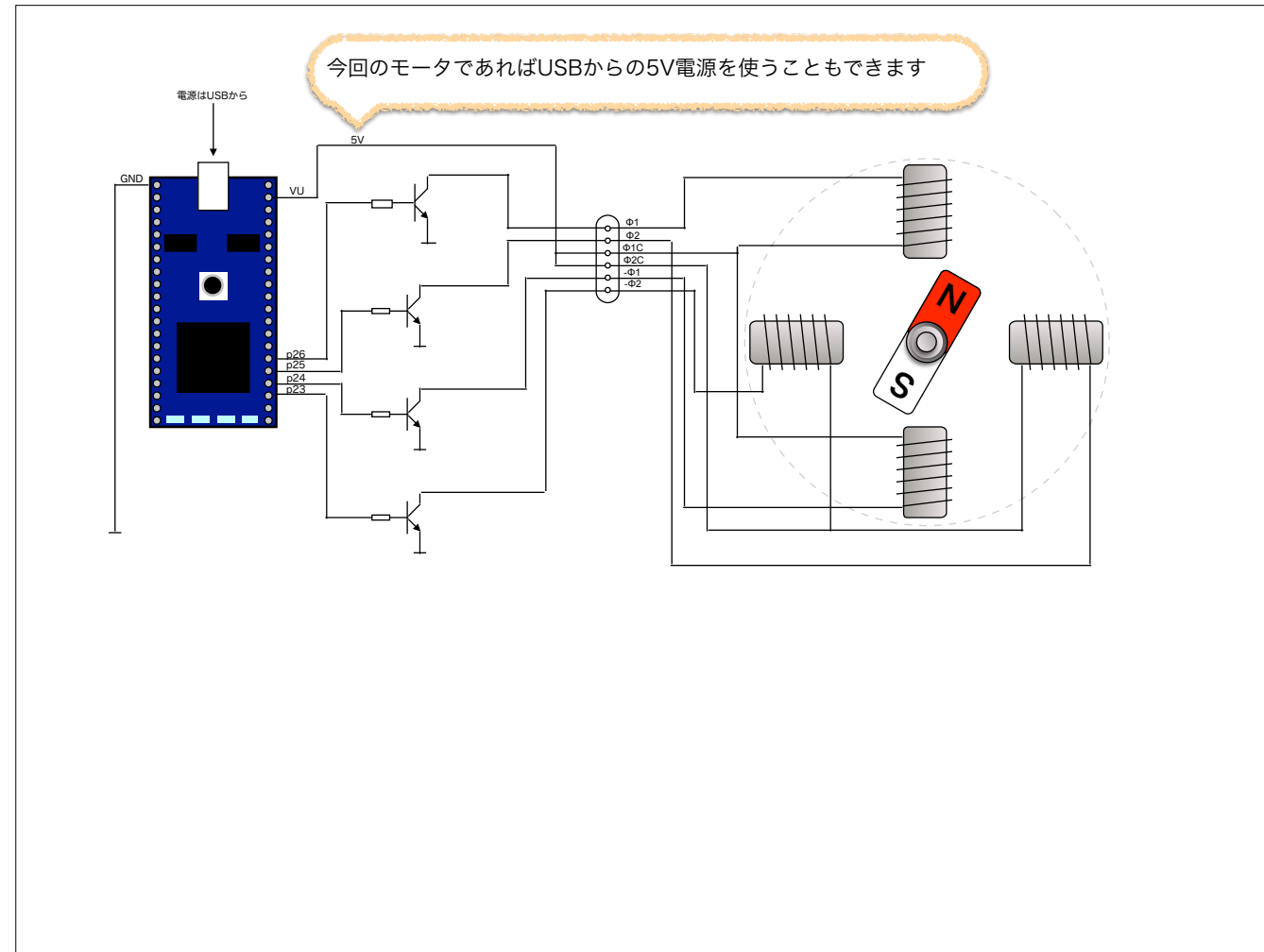
$(\nabla^\circ)$   
へえ！



ここでは電源に5Vを使うことにします.

このモータの標準的な電圧は12Vらしいのですが、それほどトルクがいらないというのならそれでも大丈夫です.

ちなみに5Vの電源の記号は(回路図表記の慣例で)上向きの矢印になっていますが、これは「5Vの電源に繋ぐ」という意味で、電流は5Vの方から流れてきます(当たり前ですけど)

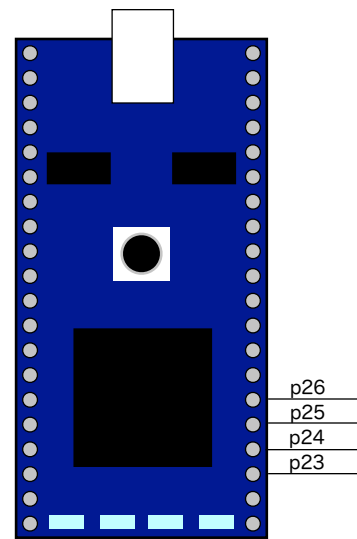


USBから電源を貰うことも可能です。

39番ピンはUSBの5V電源が出ています。電流制限素子を介した電源出力なので外部の回路がショートしてしまった時でも安心です。

ちなみに今回使うステッピングモータはコイルの直列抵抗分が約70Ω。5Vの直流を掛けた場合には約70mA程度が流れることになります





この4つのピンに  
HIGH(3.3V)とLOW(0V)を  
出力させたい

```
DigitalOut  motor_out0( p26 );  
DigitalOut  motor_out1( p25 );  
DigitalOut  motor_out2( p24 );  
DigitalOut  motor_out3( p23 );
```

mbedにHIGHとLOWの信号を出すのはとても簡単

```
DigitalOut  motor_out0( p26 );  
DigitalOut  motor_out1( p25 );  
DigitalOut  motor_out2( p24 );  
DigitalOut  motor_out3( p23 );
```

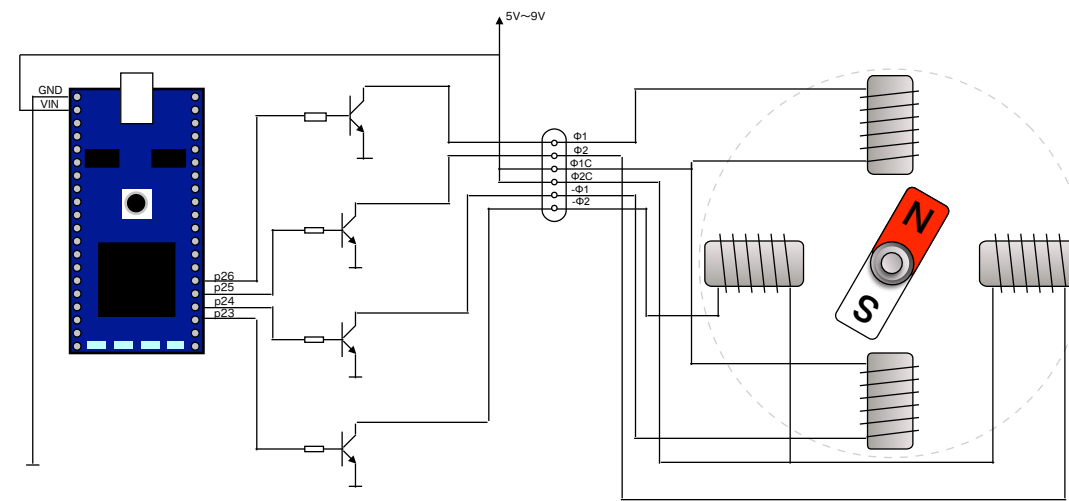
```
motor_out0  = 0;  
motor_out1  = 0;  
motor_out2  = 0;  
motor_out3  = 0;
```

LED1～LED4の代わりに  
p23～p26を指定してピンにデジタル信号を出力

0を代入するとLOWに  
1を代入するとHIGHに

詳しいコード例は  
あとでちょっと出てきます

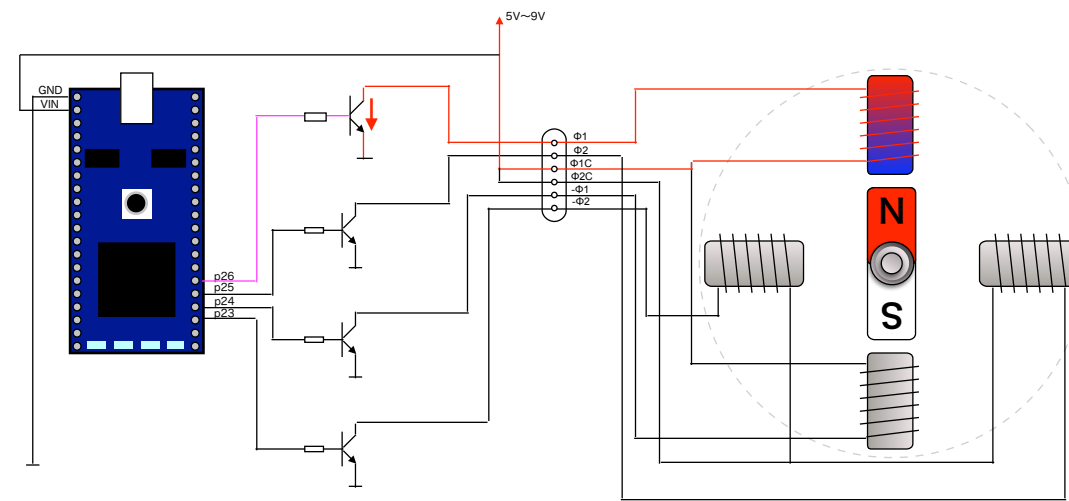
LED点滅のプログラムと同じ。ただし4個の出力があって、その出力はピンに出すこと



```
motor_out0 = 0;  
motor_out1 = 0;  
motor_out2 = 0;  
motor_out3 = 0;
```

励磁なし

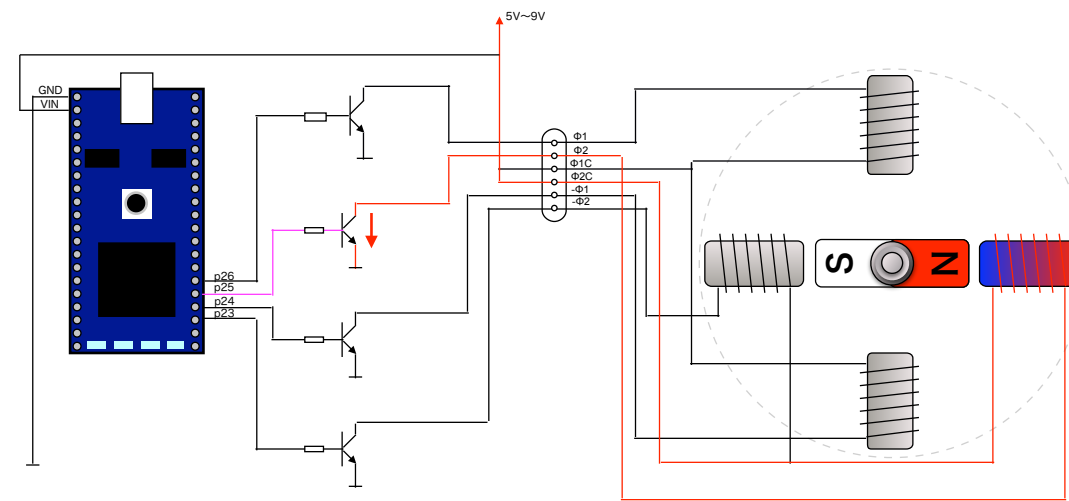
mbedからの4つのピンがすべてLOWなら，励磁されずモータはフリー状態です



```
motor_out0    = 1;  
motor_out1    = 0;  
motor_out2    = 0;  
motor_out3    = 0;
```

1相励磁(1/4)

コイルを1個ずつ励磁していく方法を1相励磁といいます。  
もっとも単純で流す電流も少なくて済む

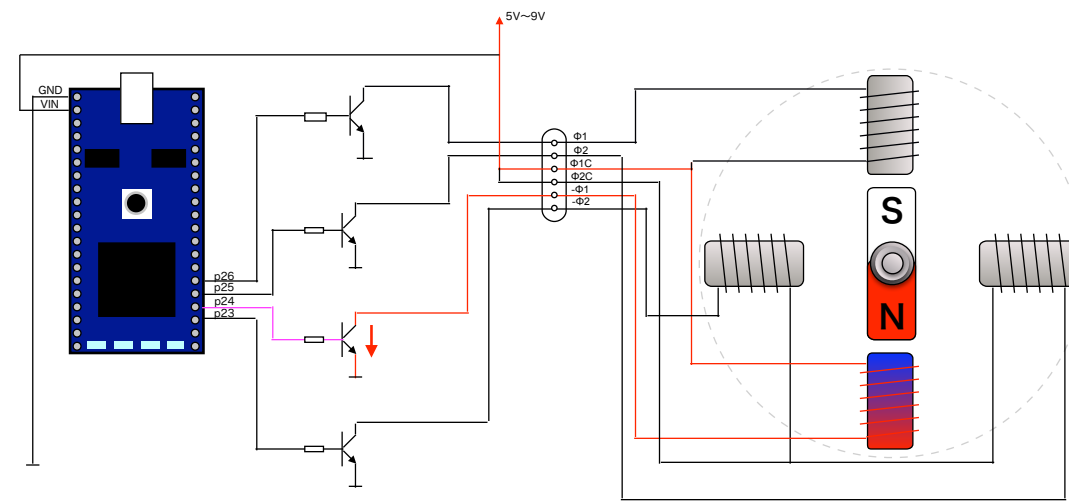


```

motor_out0 = 0;
motor_out1 = 1;
motor_out2 = 0;
motor_out3 = 0;

```

1相励磁(2/4)

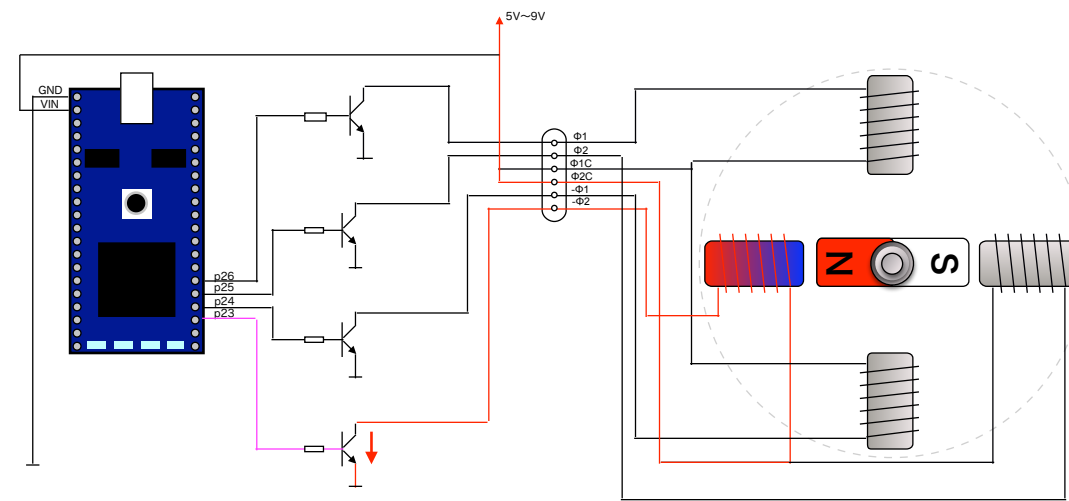


```

motor_out0 = 0;
motor_out1 = 0;
motor_out2 = 1;
motor_out3 = 0;

```

1相励磁(3/4)

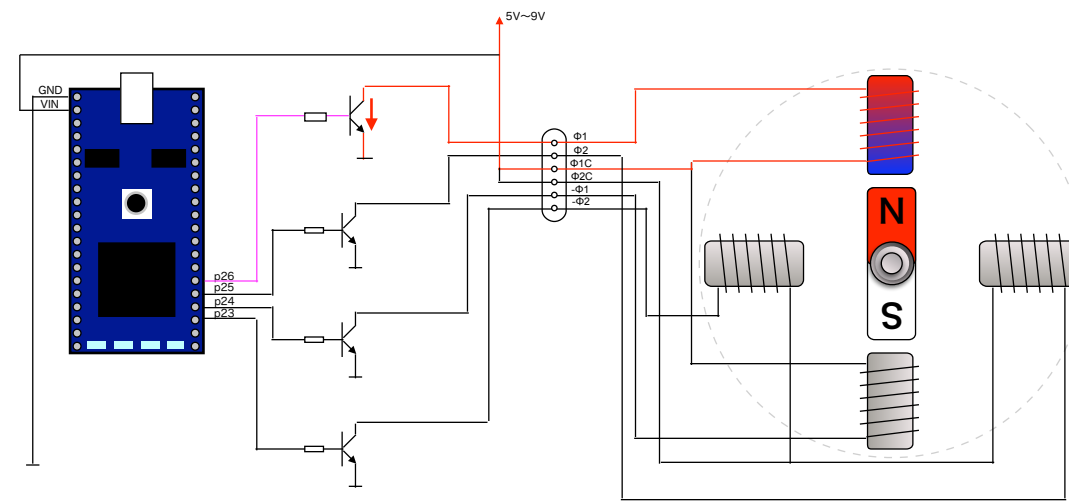


```

motor_out0 = 0;
motor_out1 = 0;
motor_out2 = 0;
motor_out3 = 1;

```

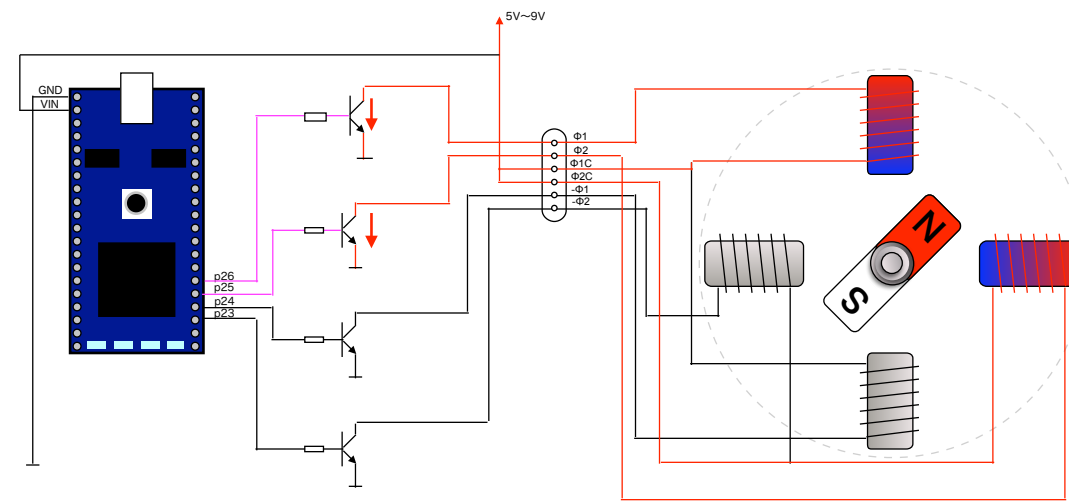
1相励磁(4/4)



```
motor_out0 = 1;  
motor_out1 = 0;  
motor_out2 = 0;  
motor_out3 = 0;
```

1相励磁(1/4)

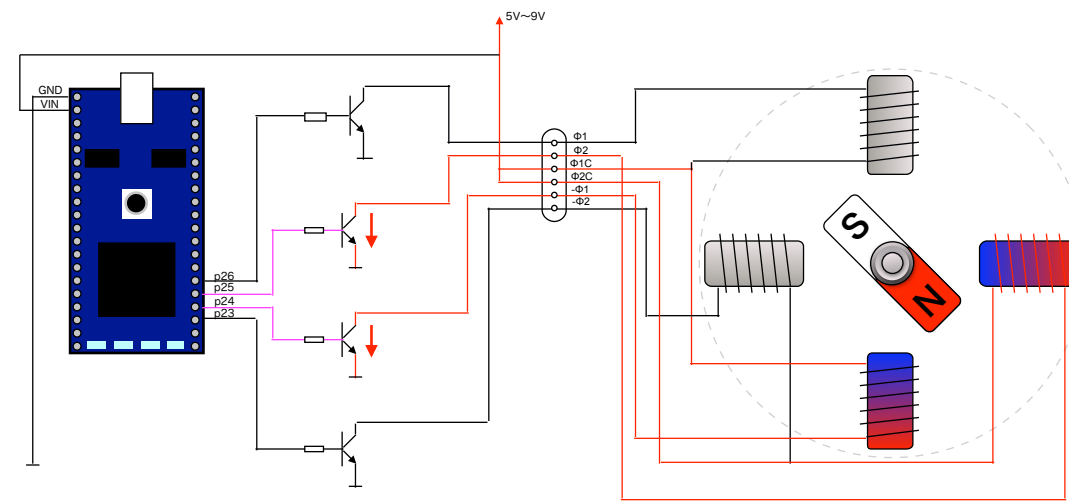




```
motor_out0 = 1;  
motor_out1 = 1;  
motor_out2 = 0;  
motor_out3 = 0;
```

2相励磁(1/4)

コイルを2個ずつ励磁していく方法もあります。2相励磁といいます。  
電流は流れますが、より強いトルクが得られる

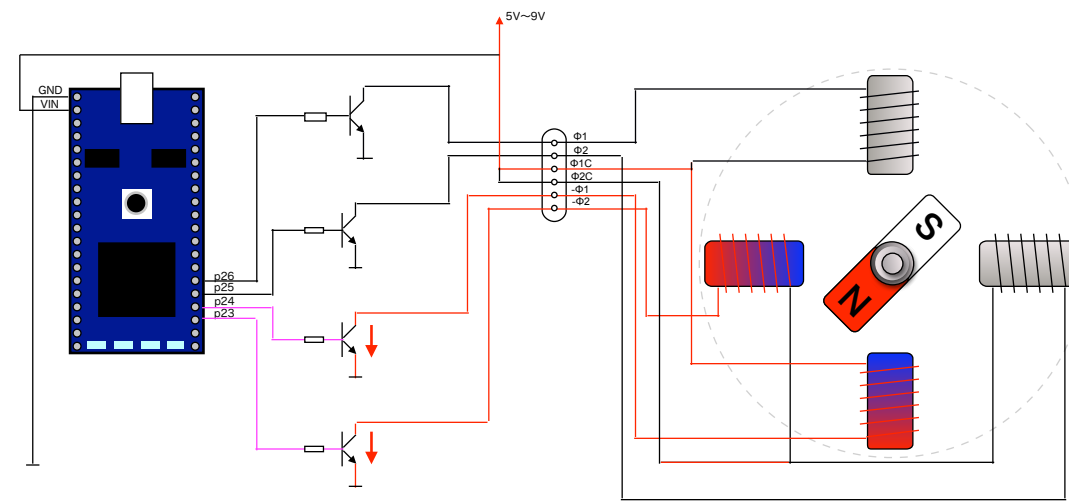


```

motor_out0 = 0;
motor_out1 = 1;
motor_out2 = 1;
motor_out3 = 0;

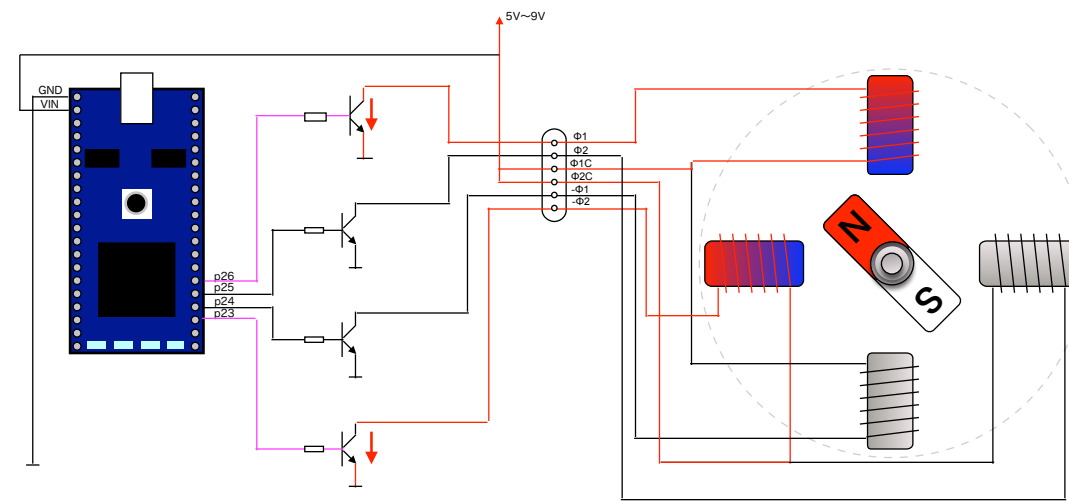
```

2相励磁(2/4)



```
motor_out0 = 0;
motor_out1 = 0;
motor_out2 = 1;
motor_out3 = 1;
```

2相励磁(3/4)

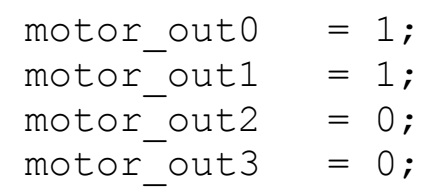


```

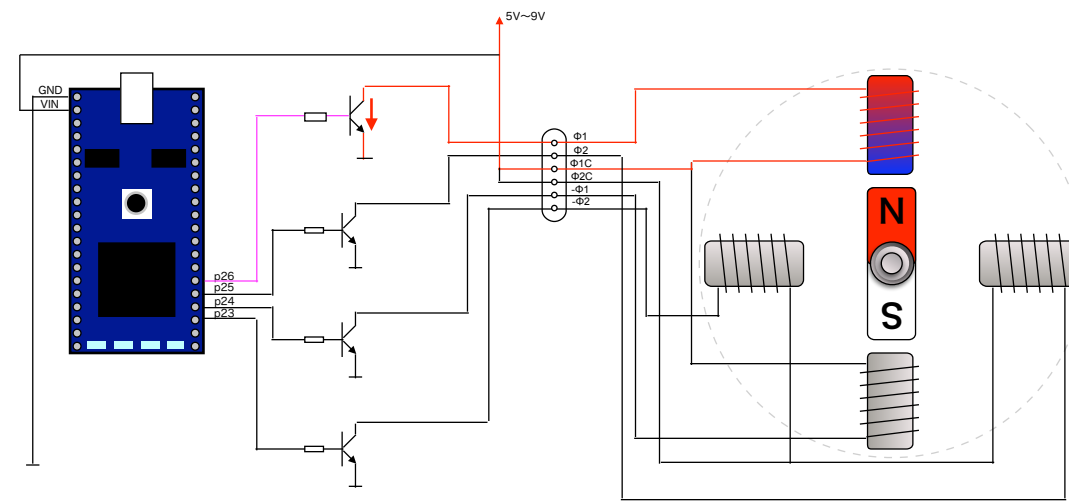
motor_out0 = 1;
motor_out1 = 0;
motor_out2 = 0;
motor_out3 = 1;

```

2相励磁(4/4)



## 2相励磁(1/4)



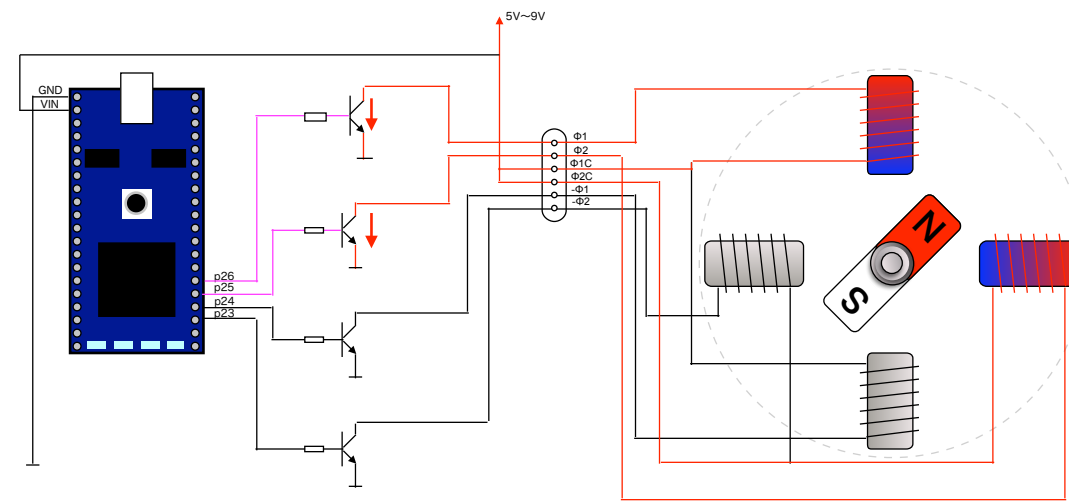
```

motor_out0    = 1;
motor_out1    = 0;
motor_out2    = 0;
motor_out3    = 0;

```

1-2相励磁(1/8)

この他に1-2相励磁と呼ばれる方法があります。ハースステップとも呼ばれ励磁を1個、2個を交互で繰り返すことによりよりステップを多く使う方法となります。(この他「マイクロステップ」と呼ばれる制御もあるのですが、アナログ的な制御方法となるため、ここでは説明しません)

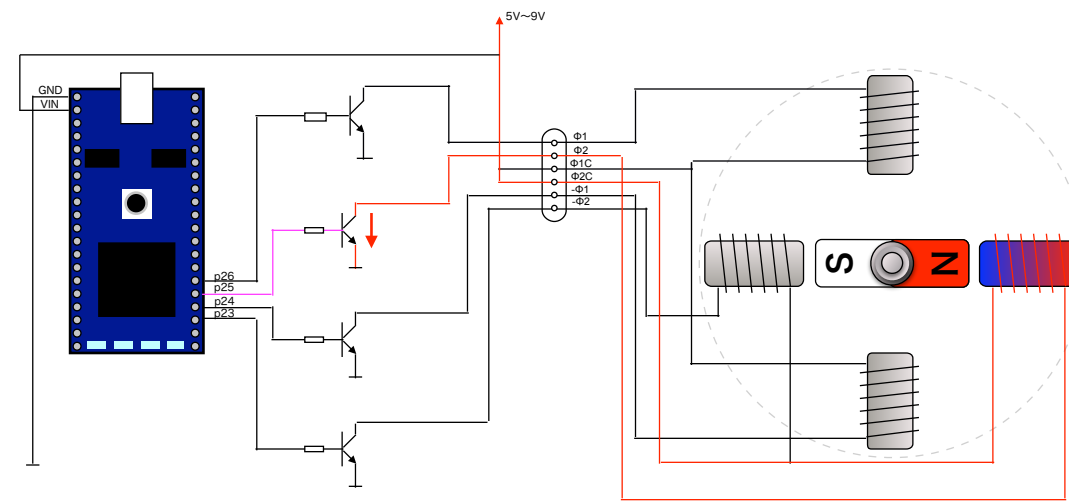


```

motor_out0 = 1;
motor_out1 = 1;
motor_out2 = 0;
motor_out3 = 0;

```

1-2相励磁(2/8)



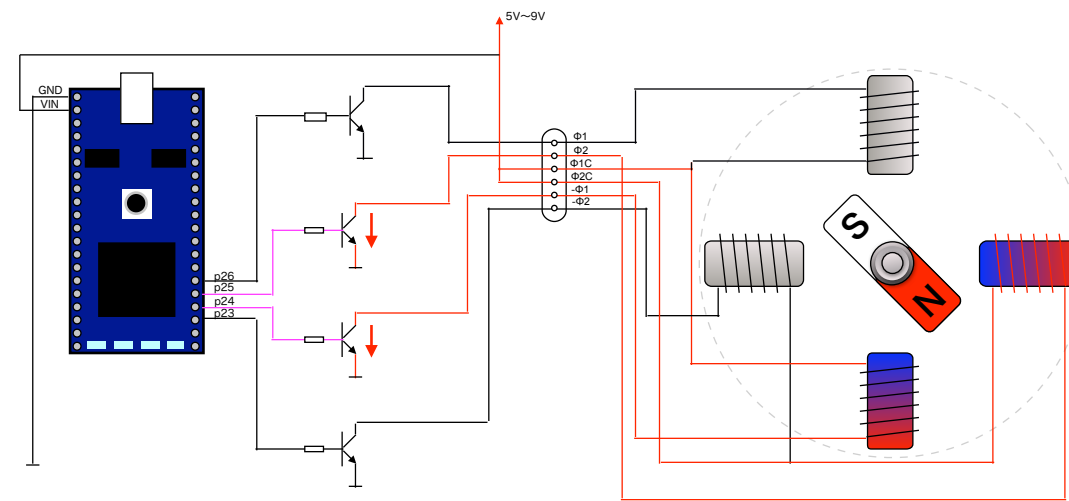
```

motor_out0 = 0;
motor_out1 = 1;
motor_out2 = 0;
motor_out3 = 0;

```

1-2相励磁(3/8)



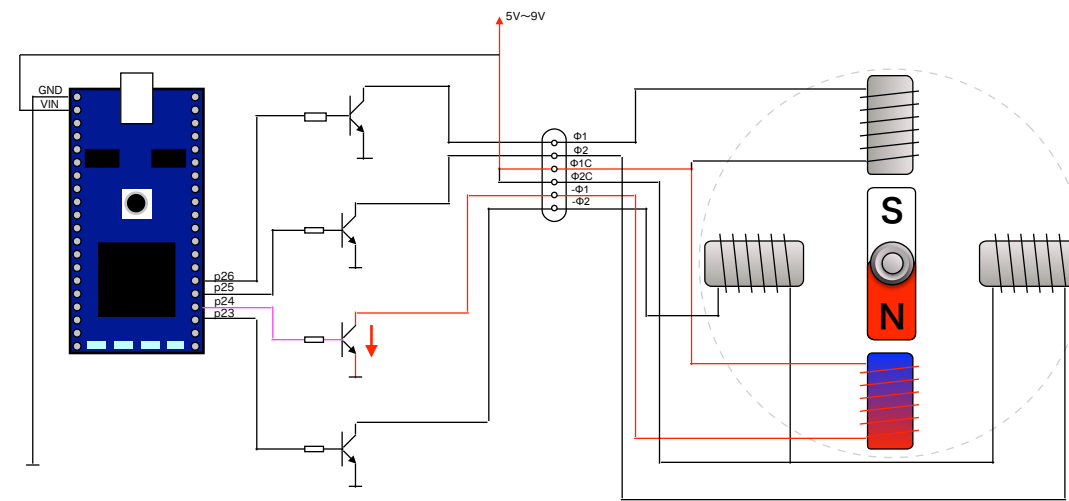


```

motor_out0 = 0;
motor_out1 = 1;
motor_out2 = 1;
motor_out3 = 0;

```

1-2相励磁(4/8)

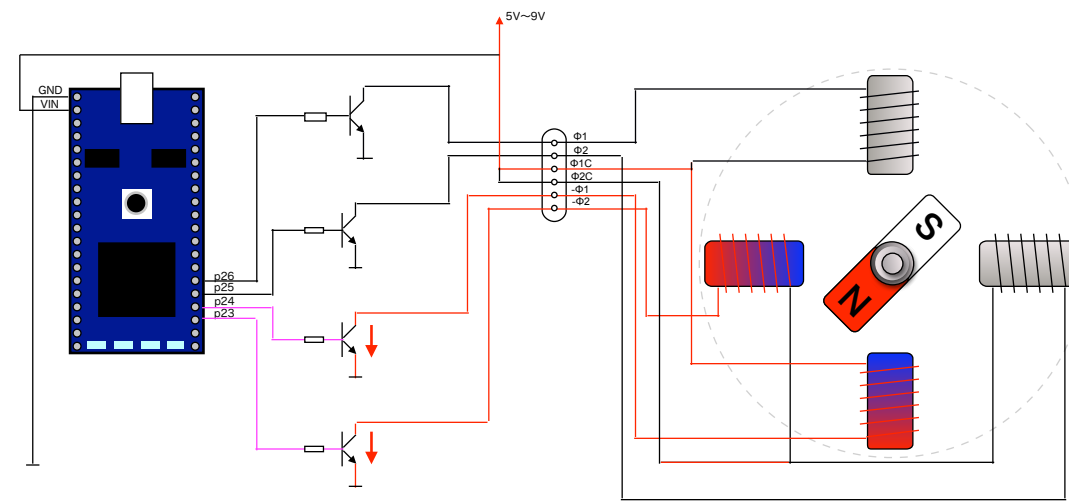


```

motor_out0 = 0;
motor_out1 = 0;
motor_out2 = 1;
motor_out3 = 0;

```

1-2相励磁(5/8)

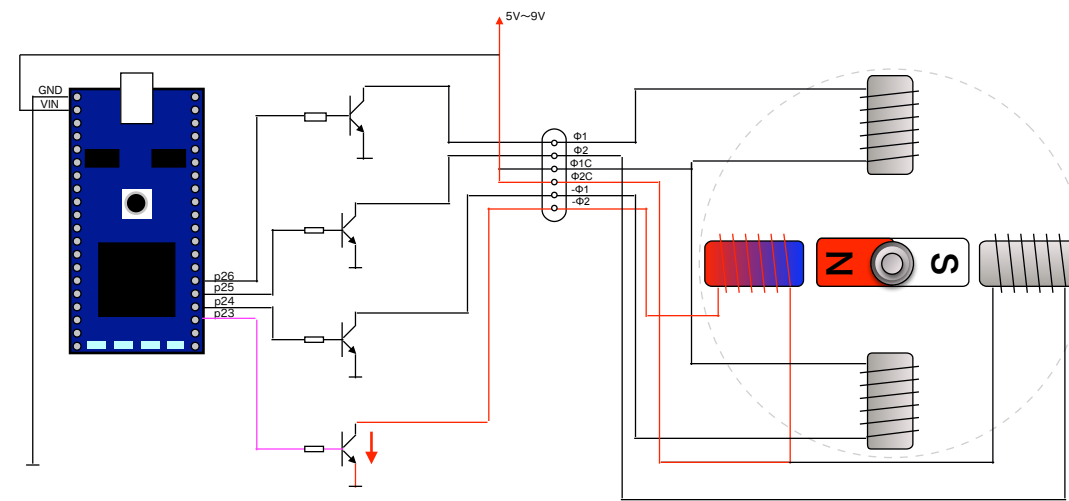


```

motor_out0 = 0;
motor_out1 = 0;
motor_out2 = 1;
motor_out3 = 1;

```

1-2相励磁(6/8)

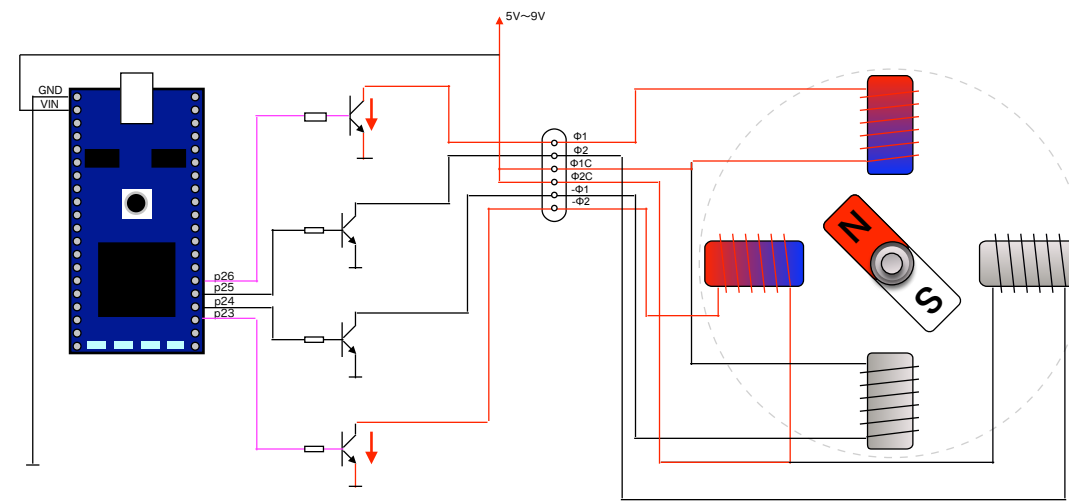


```

motor_out0 = 0;
motor_out1 = 0;
motor_out2 = 0;
motor_out3 = 1;

```

1-2相励磁(7/8)

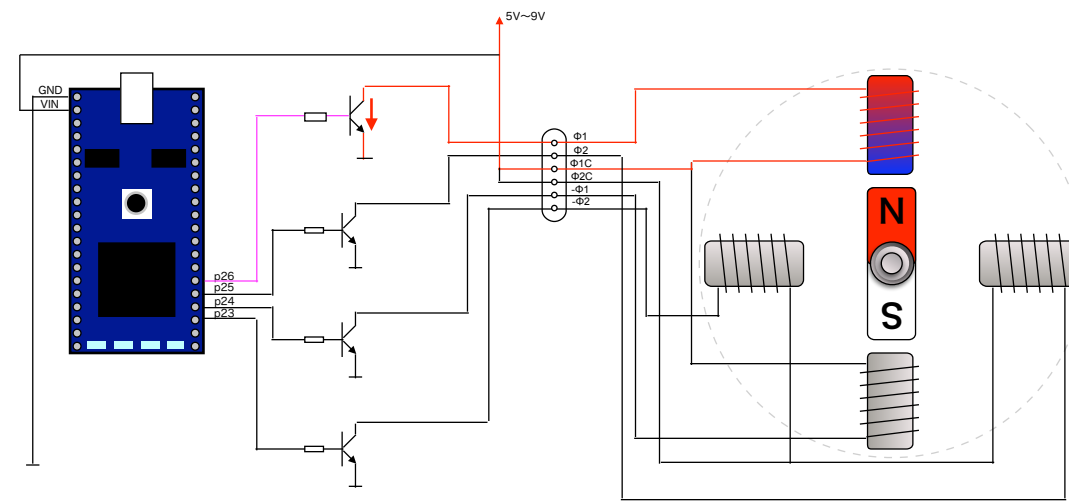


```

motor_out0 = 1;
motor_out1 = 0;
motor_out2 = 0;
motor_out3 = 1;

```

1-2相励磁(8/8)

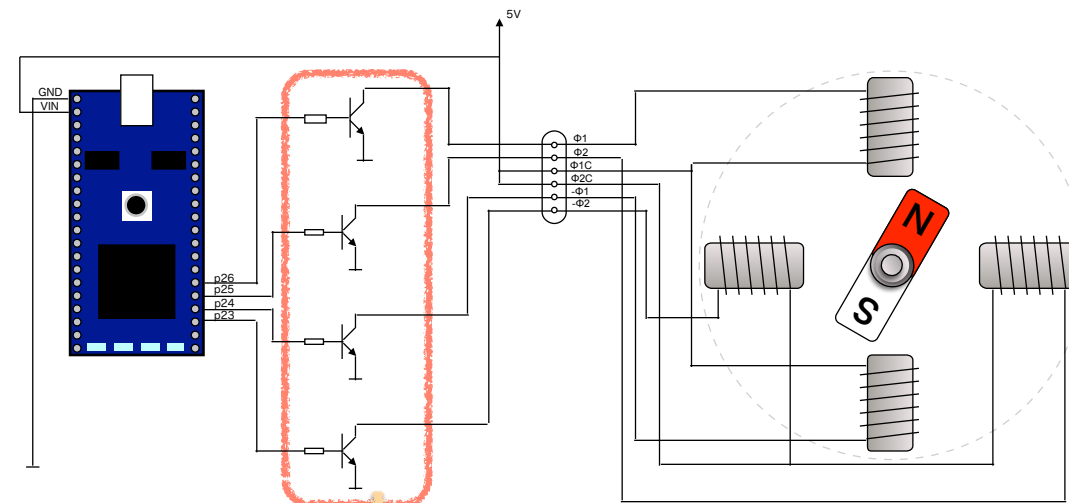


```

motor_out0 = 1;
motor_out1 = 0;
motor_out2 = 0;
motor_out3 = 0;

```

1-2相励磁(1/8)



ところでこの部分，面倒ですね。  
しかもこのままでは回路を壊しかねません

ところで，いまさらですがこの回路はオススメできません。

というのもコイルを駆動する場合にはON→OFFに切り替わった時に発生する高電圧からトランジスタを保護する部品が必要なのです。

もちろんトランジスタの種類や抵抗値の決め方なんて面倒ですね

( ° ◡ ° )

そうそう「トランジスタ」とかどうすれば？



トランジスタアレイ (8chダーリントンシンクドライバ) TD62083APG (2個入) : 半導体 秋月電子通商 電子部品 ネット通販

akizukidenshi.com/catalog/g/gI-01516/

net news weather computer mac man bike tv & radio life work えきから時刻表 駅時刻表 HTMLカラーコード

# 秋月電子通商

マイページ 注文書 お問い合わせ かごの中身 トラ技広告 回路図集

クイック注文: 通販コードを入力 (アルファベット+数字)  
数量 注文  
通販コード一括入力フォーム 検索

商品カタログ | 新商品 | お知らせ | 注文方法 | 振込先 | よくある質問 | ダウンロード | トラ技広告(PDF) | 配送状況確認 | ログイン

トップ > 半導体 > トランジスタ > トランジスタアレイ > トランジスタアレイ (8chダーリントンシンクドライバ) TD62083APG (2個入)

AAA



トランジスタアレイ (8chダーリントンシンクドライバ) TD62083APG (2個入)

[TD62083APG]  
通販コード I-01516  
発売日 2006/08/03  
メーカーカテゴリ [株式会社東芝セミコンダクター社\(TOSHIBA\)](#)

東芝の8回路入りトランジスタアレイです。  
■LEDドライブ、リレーのドライブなどに最適です。  
■18ピンDIPパッケージ  
■50V500mA/1ch

商品画像

[この商品を友達に教える](#)  
[お気に入り追加する](#)



TD62083AP.pdf

☆関連リンク  
[トランジスタアレイ](#)

関連商品>[TR \(TO-3P TO-220 TO-92 SC-59\)](#) / [コンデンサ \(フィルム 電解 セミコンダクター\)](#) / [LED \(5mm 3mm 10mm 12mm\)](#)

[I-01516] トランジスタアレイ (8chダーリントンシンクドライバ) TD62083APG (2個入)

AAA

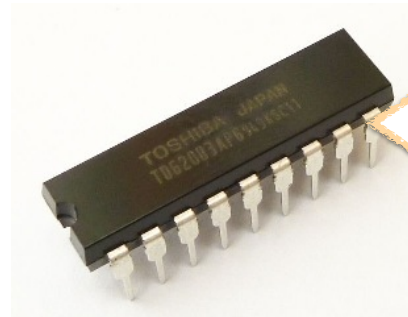
購入数量 1 パック 税100(税込)

かごに入れる 買うかごの中身を見る

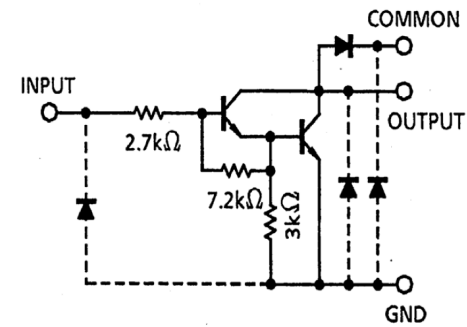
先ほどのスイッチ回路に  
**ドライバチップ**を使ってみます

そこで便利なチップを使ってみます

このチップ,  
中にこの様な回路が8個入ってます



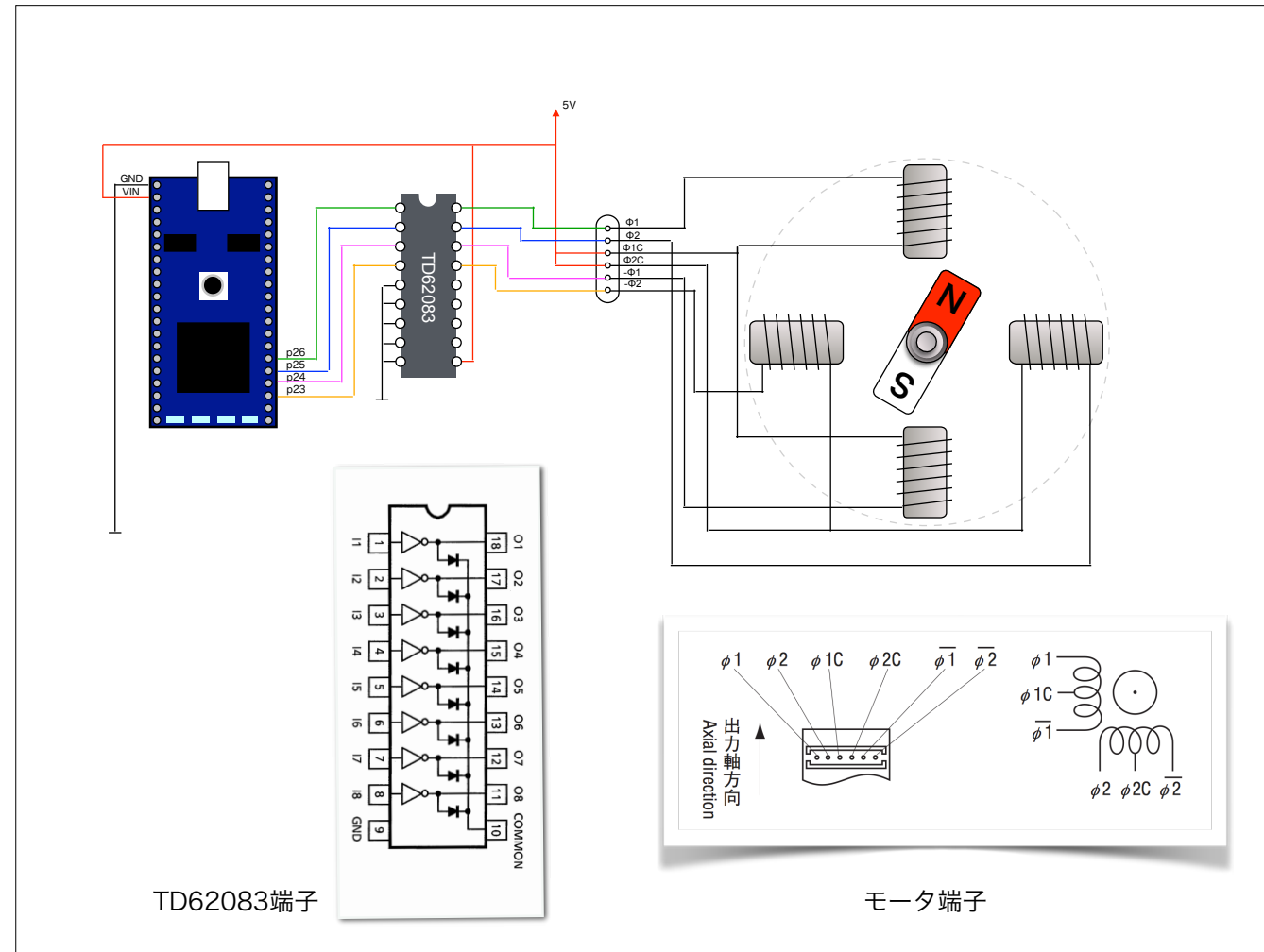
TD62083AP / AF



回路内には逆起電力クランプ用(回路保護用)  
ダイオードが入っているのでCOMMON端子を  
電源につなぐことを忘れないこと！

耐圧50V、チャンネルあたり0.5Aまでの駆動能力があるので、  
先ほどのモータに対して十分な余裕があります。  
たとえばモータに9Vをかけても、チャンネルあたりのモータの内  
部抵抗が68Ωなので、それほど大きな電流は流れません

さっきのトランジスタ8個分の回路が入っています。  
保護用のダイオードも入っているので便利

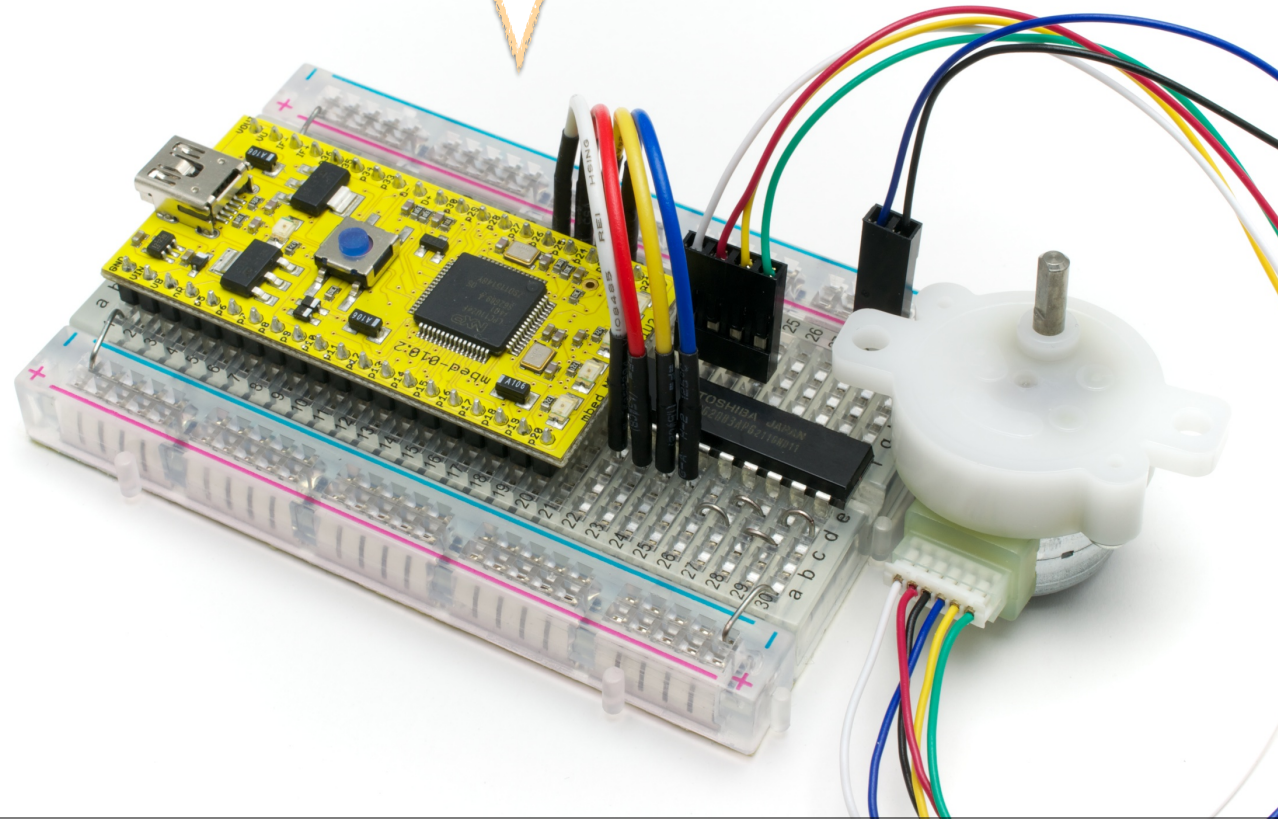


接続もとても簡単です.

5, 6, 7, 8ピンは使わない回路の入力なのでGNDに繋いでおきます.

10番ピンは保護用ダイオードが仕事してくれるように電源へ.

この例では黄色いmbedを使ってみました  
LPC1114でも同じことができます（'I'）



実際の接続を行ってみたらこんな感じ

(°▽°\*)

いがいにかんたん♡

```

#include "mbed.h"

DigitalOut motor_out0( p26 );
DigitalOut motor_out1( p25 );
DigitalOut motor_out2( p24 );
DigitalOut motor_out3( p23 );

int main() {
    while(1) {
        motor_out0 = 1;
        motor_out1 = 0;
        motor_out2 = 0;
        motor_out3 = 0;
        wait( 0.01 );

        motor_out0 = 0;
        motor_out1 = 1;
        motor_out2 = 0;
        motor_out3 = 0;
        wait( 0.01 );

        motor_out0 = 0;
        motor_out1 = 0;
        motor_out2 = 1;
        motor_out3 = 0;
        wait( 0.01 );

        motor_out0 = 0;
        motor_out1 = 0;
        motor_out2 = 0;
        motor_out3 = 1;
        wait( 0.01 );
    }
}

```

プログラムも簡単です.

LED点滅と同じように「wait」を入れながら各ピンの出力を変えてやるだけ

簡単なサンプルを作っておきました (^ ^)

[http://mbed.org/users/okano/code/unipolar\\_stepper\\_motor\\_operation\\_sample/](http://mbed.org/users/okano/code/unipolar_stepper_motor_operation_sample/)

サンプルコードがあるので、それですぐに試せます



前スライドのURLのページです。

このサンプルコードはLED点滅からモータを回してみるまで、順を追って説明するようにつくられました。

その各段階は「履歴」機能で見ることができます。Graphタブを押すと履歴の一覧が表示されます



HomeHistoryGraphAPI DocumentationWiki

Revision graph

11:436e74a3ec7a version 8 : motor control by function default

10:f00d0b8775d4 version 7 : speed setting by variable

9:e568dea69ab5 version 6 : 2-phase drive

8:6f3f8e5da87b version 5 : Smaller loop using array index

7:032ae28fae2e version 4 : Output pulse pattern in array

6:11b35048d384 version 3 : Operation by BusOut

5:28bbda0fe9b5 version 0-2 : Check BusOut by LED

4:3cb4e78e0846 version 2 : 360 steps clockwise and counter

3:db08e5009d0 version 1-1 : Rotate motor in reverse

2:e9fbf2c17789 version 1 : Output pulses to p23, p24, p25

1:40f906a4dbd version 0-1 : Speed changed

0:6e775c640f78 version 0 : Check DigitalOut by LED

HomeHistoryGraphAPI DocumentationWiki

Revision 2:e9fbf2c17789 day ago

Committer: Tedd OKANO

Date: a day ago

Revision: 2:e9fbf2c17789

Parent: 1:e49f906a4dbd

Child: 3:db08e5009d0

Commit message: version

Changed in this revision

main.cpp Show annotated file

```
1 1 // stepper motor operation sample (learn it step by step)
2 2 // showing how to control a unipolar stepper motor
3 3
4 4 // version 0-1 : Speed changed
5 5 // version 1 : Output pulses to p23, p24, p25 and p26
6 6
7 7
8 8 #include "mbed.h"
9 9
10 10 DigitalOut motor_out0( LED1 );
11 11 DigitalOut motor_out1( LED2 );
12 12 DigitalOut motor_out2( LED3 );
13 13 DigitalOut motor_out3( LED4 );
14 14
15 15 int main() {
16 16     while(1) {
17 17         motor_out0 = 1;
18 18         motor_out1 = 0;
19 19         motor_out2 = 0;
20 20         motor_out3 = 0;
21 21         wait( INTERVAL );
22 22
23 23         motor_out0 = 0;
24 24         motor_out1 = 1;
25 25         motor_out2 = 0;
26 26         motor_out3 = 0;
27 27         wait( INTERVAL );
28 28
29 29         motor_out0 = 0;
30 30         motor_out1 = 0;
31 31         motor_out2 = 1;
32 32         motor_out3 = 0;
33 33         wait( INTERVAL );
34 34
35 35         motor_out0 = 0;
36 36         motor_out1 = 0;
37 37         motor_out2 = 0;
38 38         motor_out3 = 1;
39 39         wait( INTERVAL );
40 40
41 41     }
42 42 }
43 43
44 44
45 45
```

HomeHistoryGraphAPI DocumentationWiki

main.cpp

Committer: Tedd OKANO

Date: a day ago

Revision: 2:e9fbf2c17789

Parent: 1:e49f906a4dbd

Child: 3:db008e5009d0

File content as of revision 2:e9fbf2c17789

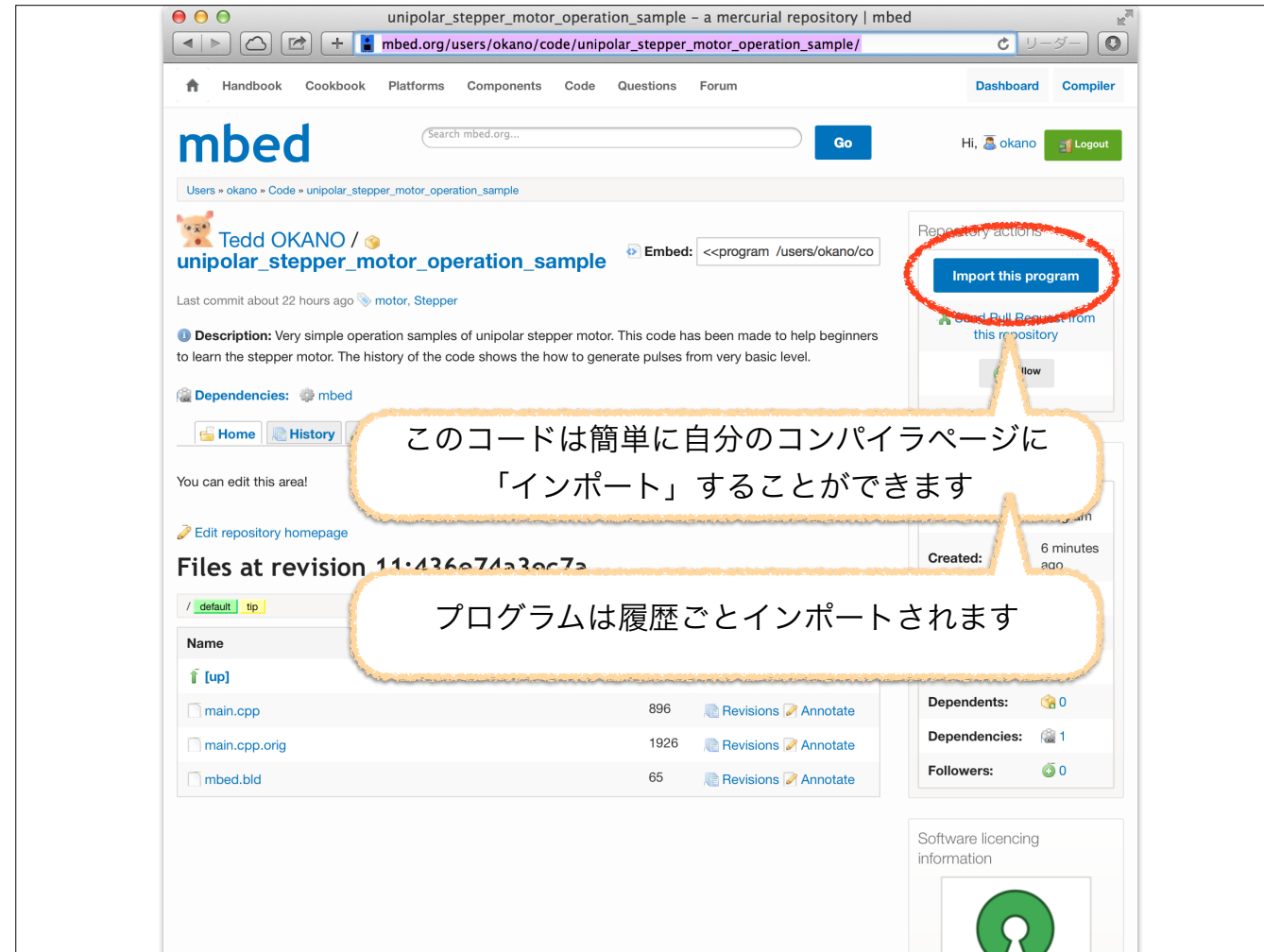
```
1 1 // stepper motor operation sample (learn it step by step)
2 2 // showing how to control a unipolar stepper motor
3 3
4 4 // version 1 : Output pulses to p23, p24, p25 and p26
5 5
6 6 #include "mbed.h"
7 7
8 8 DigitalOut motor_out0( p26 );
9 9 DigitalOut motor_out1( p25 );
10 10 DigitalOut motor_out2( p24 );
11 11 DigitalOut motor_out3( p23 );
12 12
13 13 #define INTERVAL 0.01
14 14
15 15 int main() {
16 16     while(1) {
17 17         motor_out0 = 1;
18 18         motor_out1 = 0;
19 19         motor_out2 = 0;
20 20         motor_out3 = 0;
21 21         wait( INTERVAL );
22 22
23 23         motor_out0 = 0;
24 24         motor_out1 = 1;
25 25         motor_out2 = 0;
26 26         motor_out3 = 0;
27 27         wait( INTERVAL );
28 28
29 29         motor_out0 = 0;
30 30         motor_out1 = 0;
31 31         motor_out2 = 1;
32 32         motor_out3 = 0;
33 33         wait( INTERVAL );
34 34
35 35         motor_out0 = 0;
36 36         motor_out1 = 0;
37 37         motor_out2 = 0;
38 38         motor_out3 = 1;
39 39         wait( INTERVAL );
40 40
41 41     }
42 42 }
43 43
44 44
45 45
```

リビジョン番号をクリックすると、  
前のリビジョンとの差分が表示されます

プログラムの履歴を見てみよう

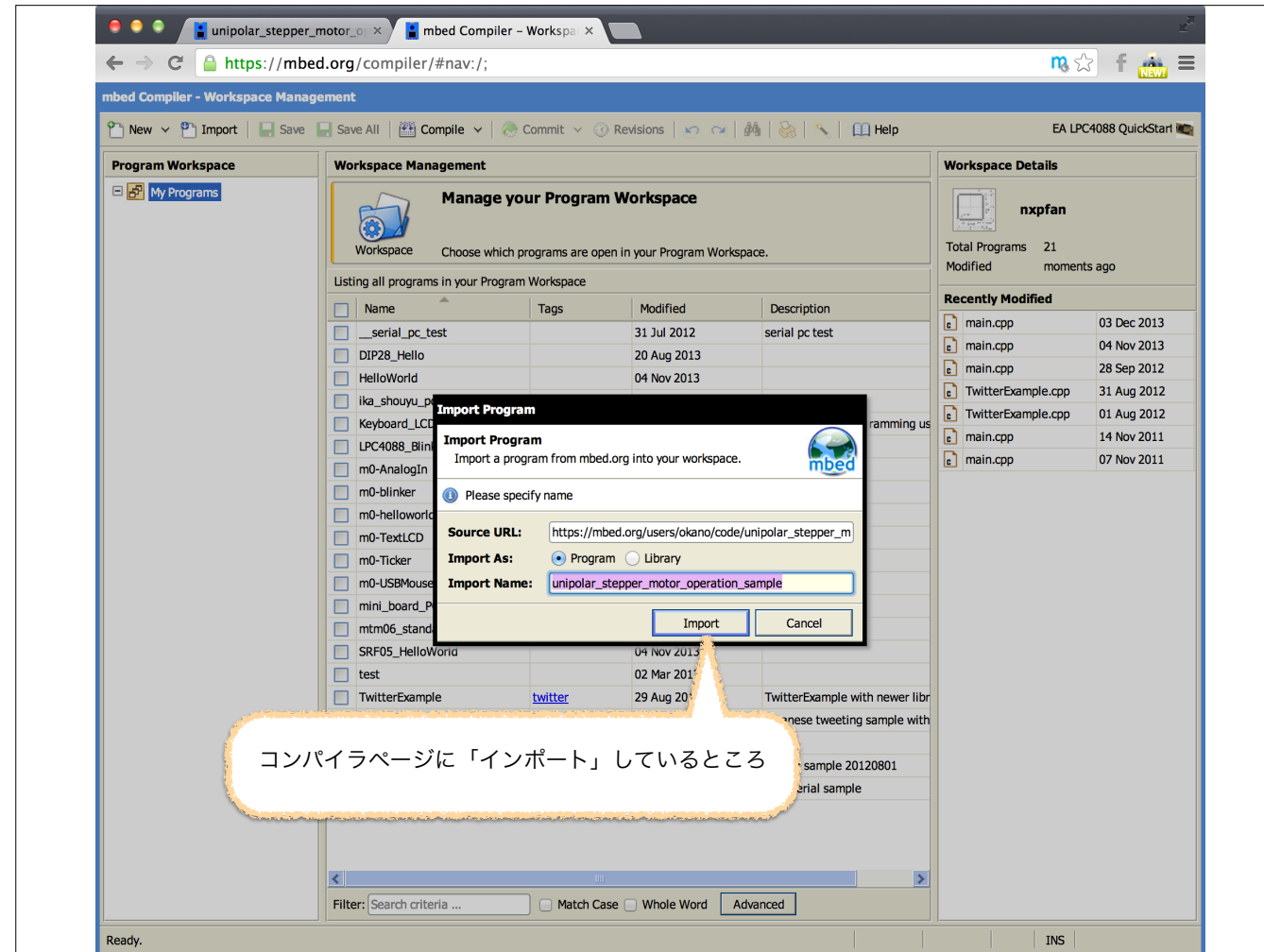
ファイル名をクリックすると  
そのリビジョンのファイルが  
表示されます

見てみたい履歴のリビジョンをクリックすると、その前のリビジョンとの差分を表示してくれます。さらにファイル名をクリックすると、コード全体がひょうじされます

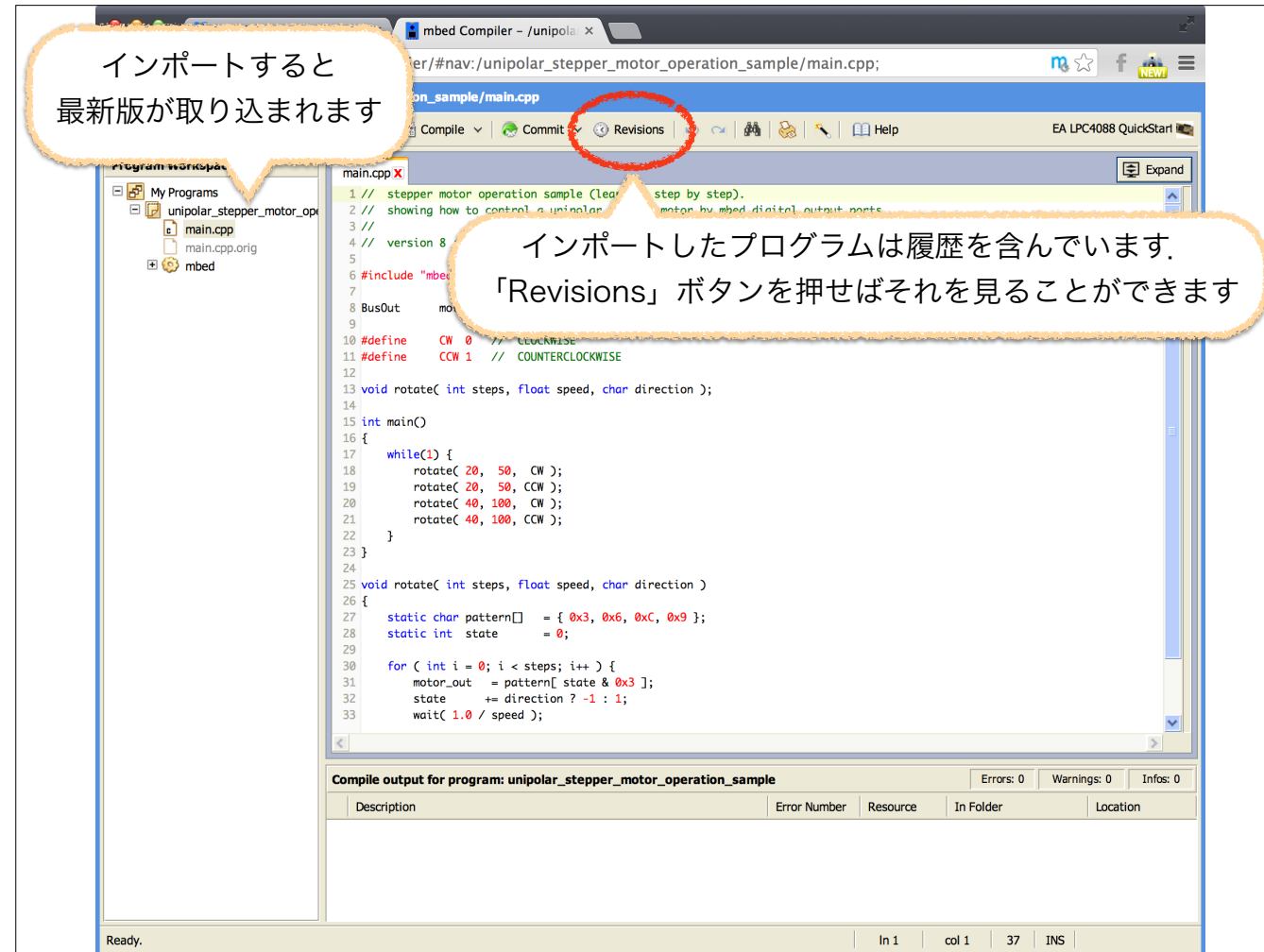


mbedなので、サンプルをインポートするのも簡単.

インポートしたプログラムには「履歴」の情報も含まれます



これは自分のコンパイラ・ページにプログラムをインポートしているところです



インポートが完了すると、そのプログラムの最新版が使えるようになります。

「Revisions」ボタンを使うと前のリビジョンに戻すことが可能です

インポートしたプログラムは履歴を含んでいます。  
「Revisions」 ボタンを押せばそれを見ることができます

① 「Revision」 ボタンを押す

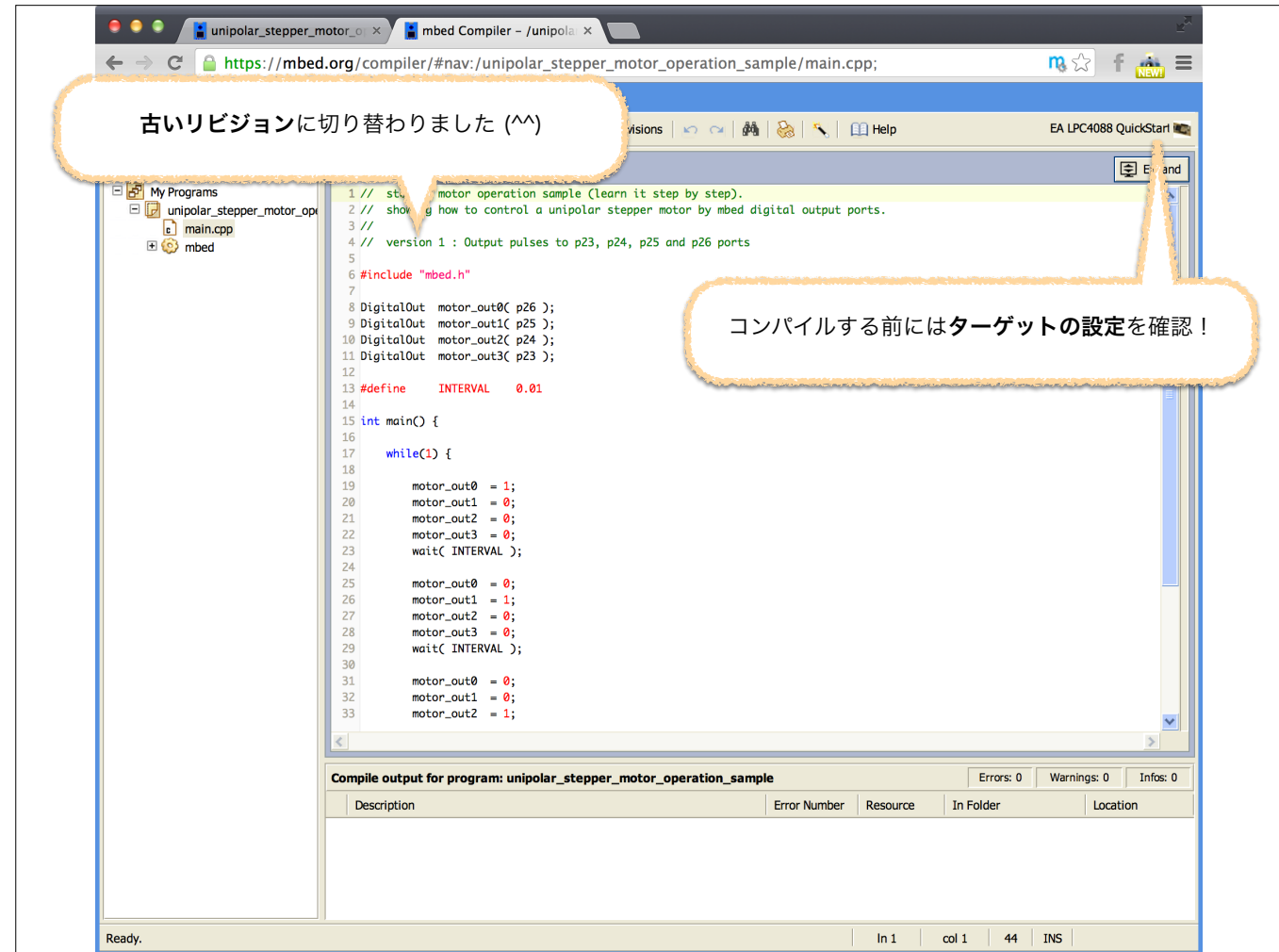
② 切り替えたい対象のリビジョンを選択

③ 「Switch」 を押して切り替え

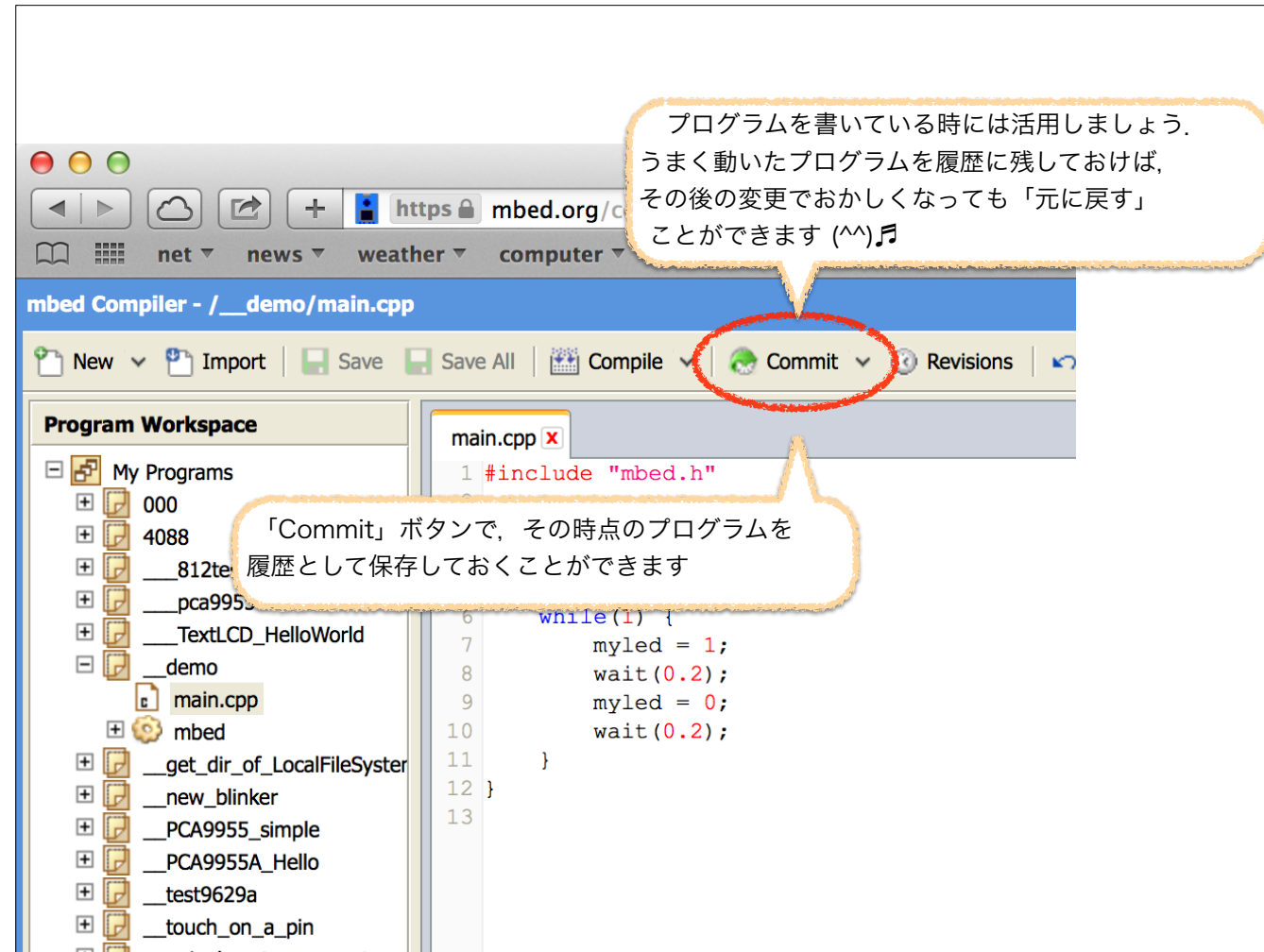
古いコードに切り替えるときは、その項目を選択しておいて「Switch」 ボタンを押します。 そうするとプログラムは保存された古いコードに置き換わります

Graph	Revisio	When	Who	Comment
	11	23 hours, 23 minutes	okano	version 1: Output pulses
	10	23 hours, 24 minutes ago	okano	version 0-1: Speed changed
	9	23 hours, 24 minutes ago	okano	version 0: Check DigitalOut by LED
	8	23 hours, 24 minutes ago	okano	version 1-1: Rotate motor in reverse direction
	7	1 day, 1 hour ago	okano	version 2: 360 steps clockwise and counter-clockwise
	6	1 day, 1 hour ago	okano	version 3: Operation by BlueOut
	5	1 day, 1 hour ago	okano	version 4: Output pulse pattern in array
	4	1 day, 1 hour ago	okano	version 5: Smaller loop using array index
	3	1 day, 1 hour ago	okano	version 6: 2-phase drive
	2	1 day, 1 hour ago	okano	version 1: Output pulses to p23, p24, p25 and
	1	1 day, 1 hour ago	okano	version 0-1: Speed changed
	0	1 day, 1 hour ago	okano	version 0: Check DigitalOut by LED

切り替えたいリビジョンを選択→「スイッチ」を押す。



そうすると履歴中の古いコードに戻すことができます



この「履歴」機能。一般に「リビジョン管理」といわれる機能は自分がプログラムを書いている時にも便利です。

うまく動いた時点のプログラムを「Commit」すれば、その状態を記憶させておくことができます。これを用いて段階的に開発を進めていけば、たとえば

「後から足したコードが動かず、一旦元に戻したい」というような時に、すぐに前の状態に戻すことができます。

mbedでのオンライン・コンパイラは、こんな機能も備えているので便利なのです



# 注意

- モータやドライバチップが過熱状態になることがあります。ONになりっぱなし状態には注意
- 電源電圧，モータ，ドライバチップを変更する際は慎重に
- 大きな電流を流すことで，大事な部品を壊したり，燃やしたりする可能性があります
- わからない時には，わかりそうな人に聞くこと
- 工作は自己責任の範囲で

最後ですが注意点をまとめました





- ・ ステッピングモータSPG20-1362

<http://akizukidenshi.com/catalog/g/gP-04241/>

- ・ トランジスタアレイ(8chダーリントンシンクドライバ)TD62083

<http://akizukidenshi.com/catalog/g/gI-01516/>

- ・ ステッピングモータの簡単なコード例

[http://mbed.org/users/okano/code/unipolar\\_stepper\\_motor\\_operation\\_sample/](http://mbed.org/users/okano/code/unipolar_stepper_motor_operation_sample/)

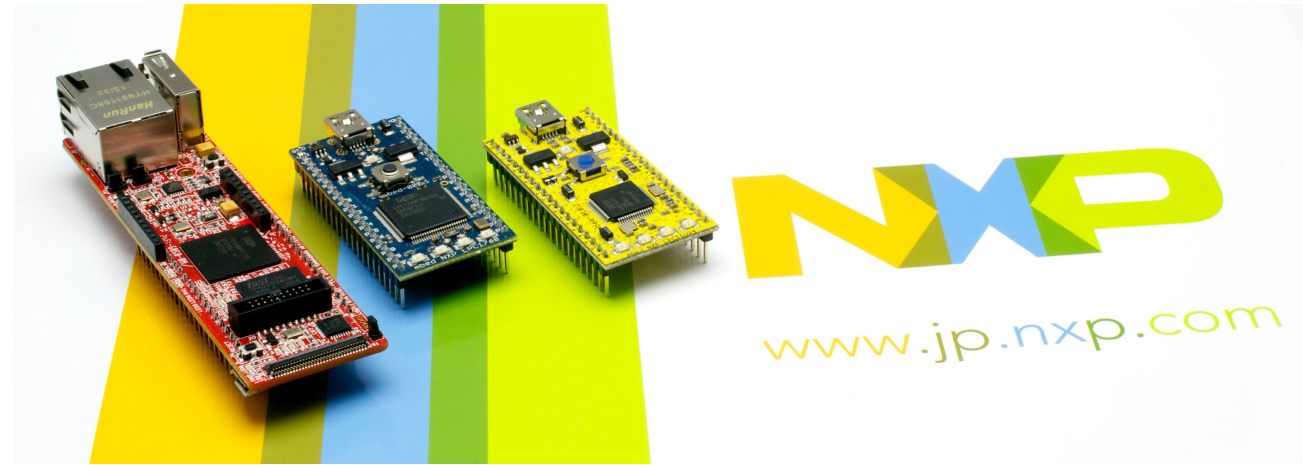
- ・ **NXPのマイコンについての日本語情報は**

<http://www.nxp-lpc.com>

- ・ mbed関連情報

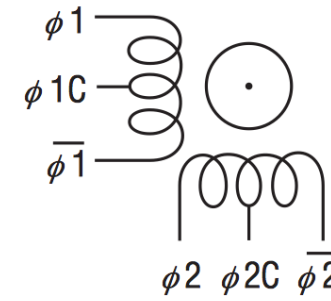
<http://mbed.org/users/nxpfan/notebook>

こちらは参考情報



## 補足情報

ステッピングモータの説明書にはよくつぎのような図が描かれています.  
このような場合,  $\phi 1C$ と $\phi 2C$ を電源に接続, ドライバチップからは  
 $\phi 1 \rightarrow \phi 2 \rightarrow \overline{\phi 1} \rightarrow \overline{\phi 2}$ の順で駆動します



mbed祭りで希望者に配布した「PM25S-024」  
には説明書がついていましたが, 上記を参考にし  
て繋いでみてください