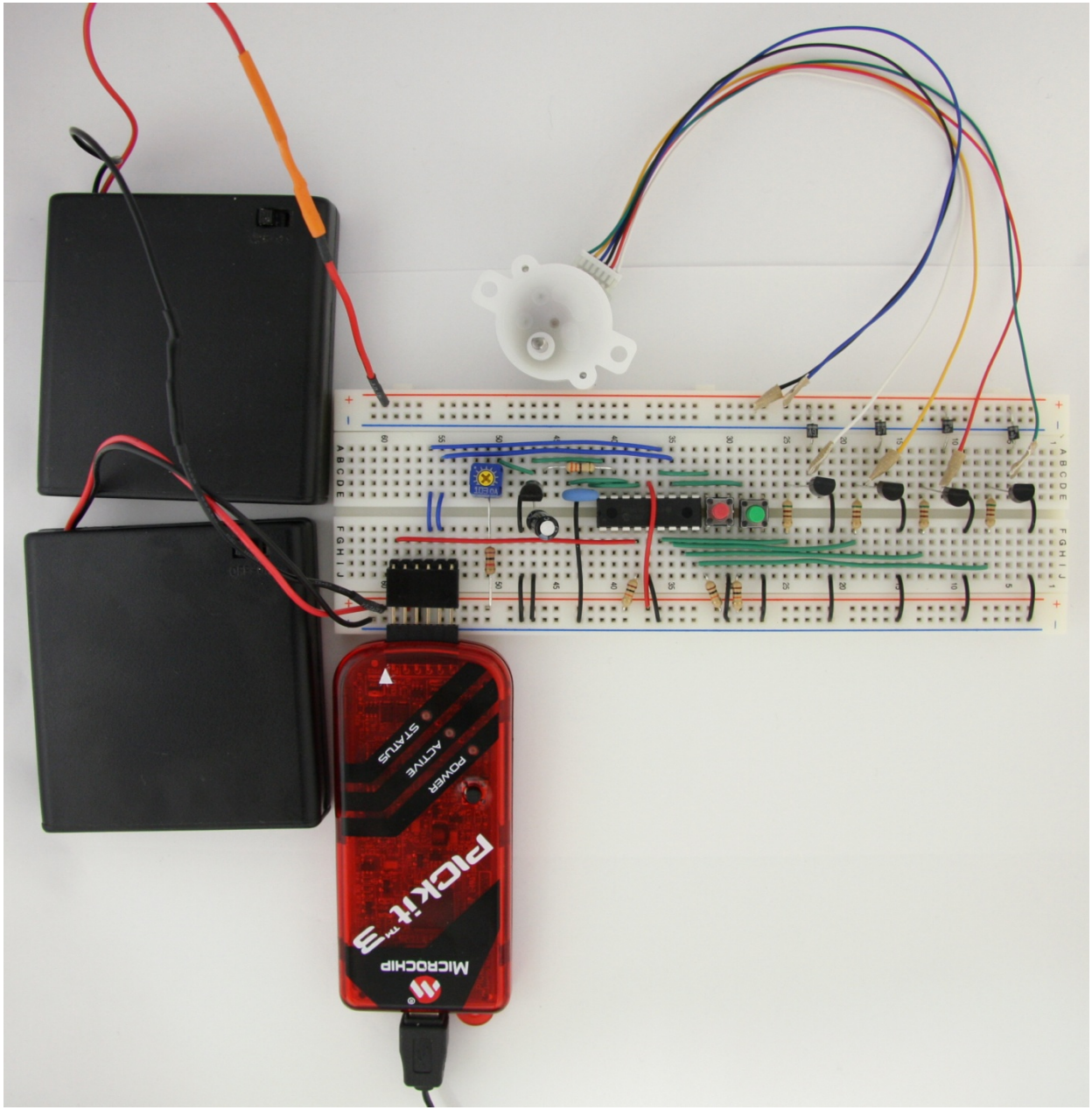
８．ステッピングモータの制御を学ぼう

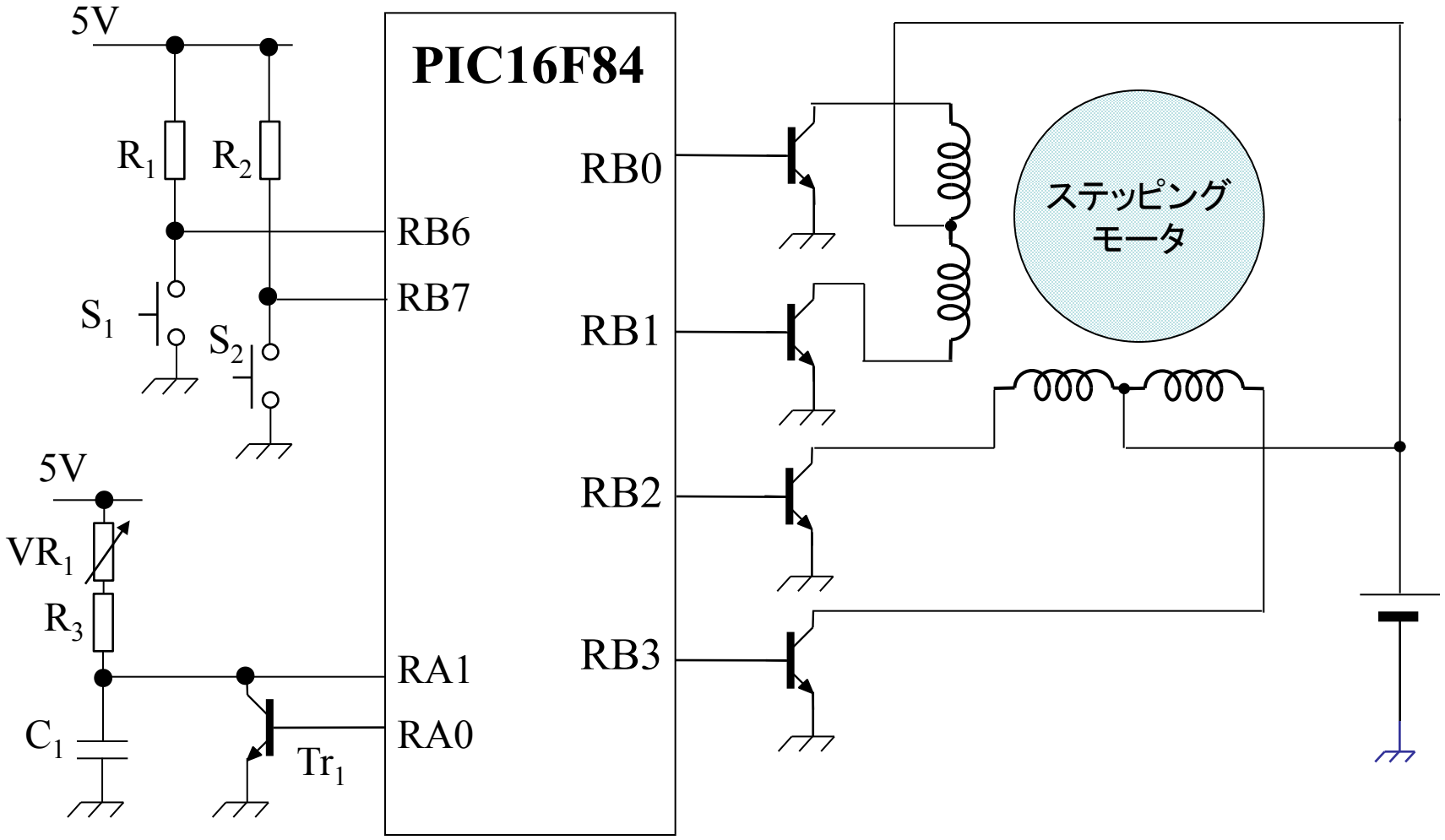
秋月電子通商　ＰＩＣステッピングモータドライバキット（小型モータ付き）を参照しました．

回路製作の詳細は第０章を参照してください．

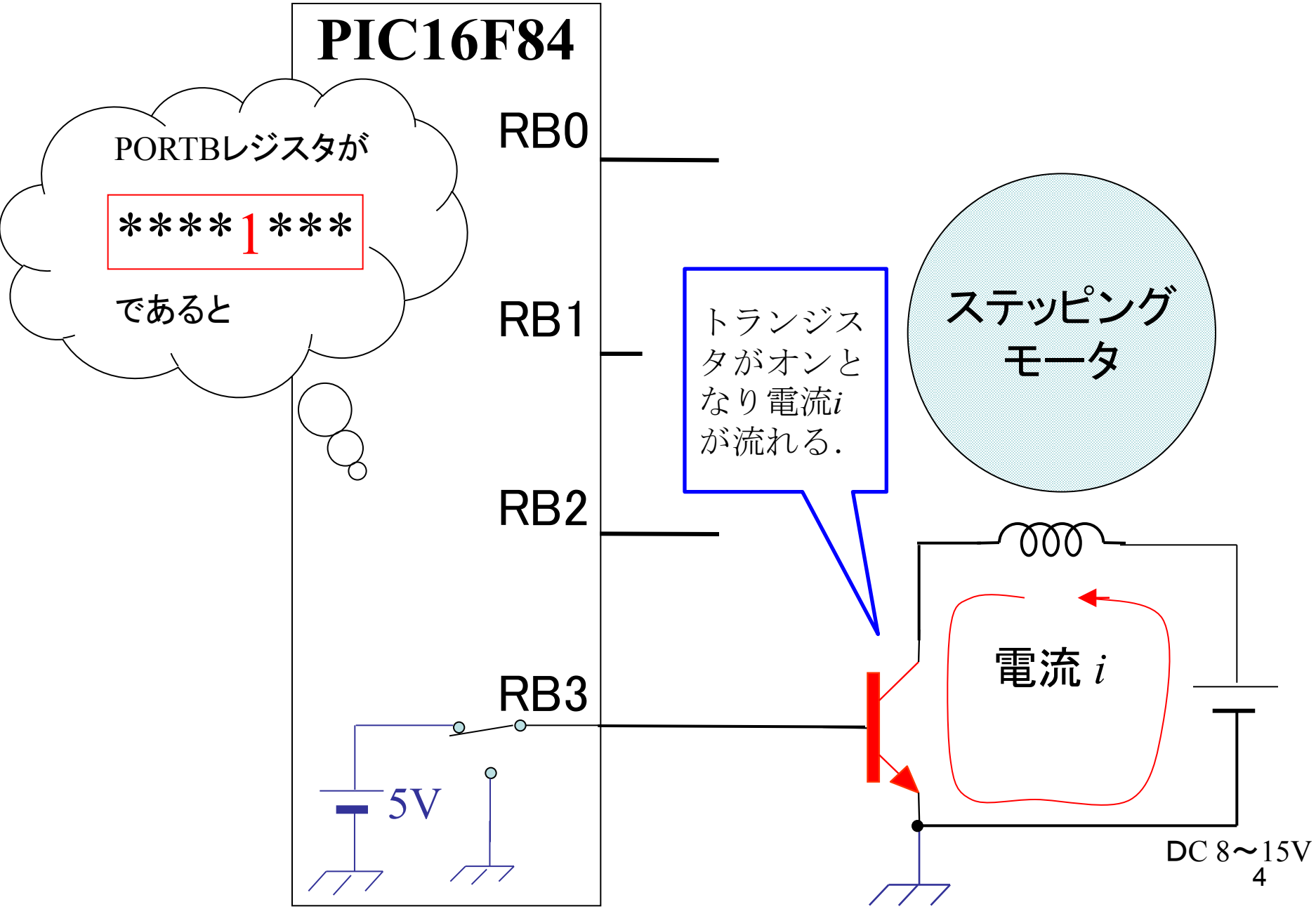


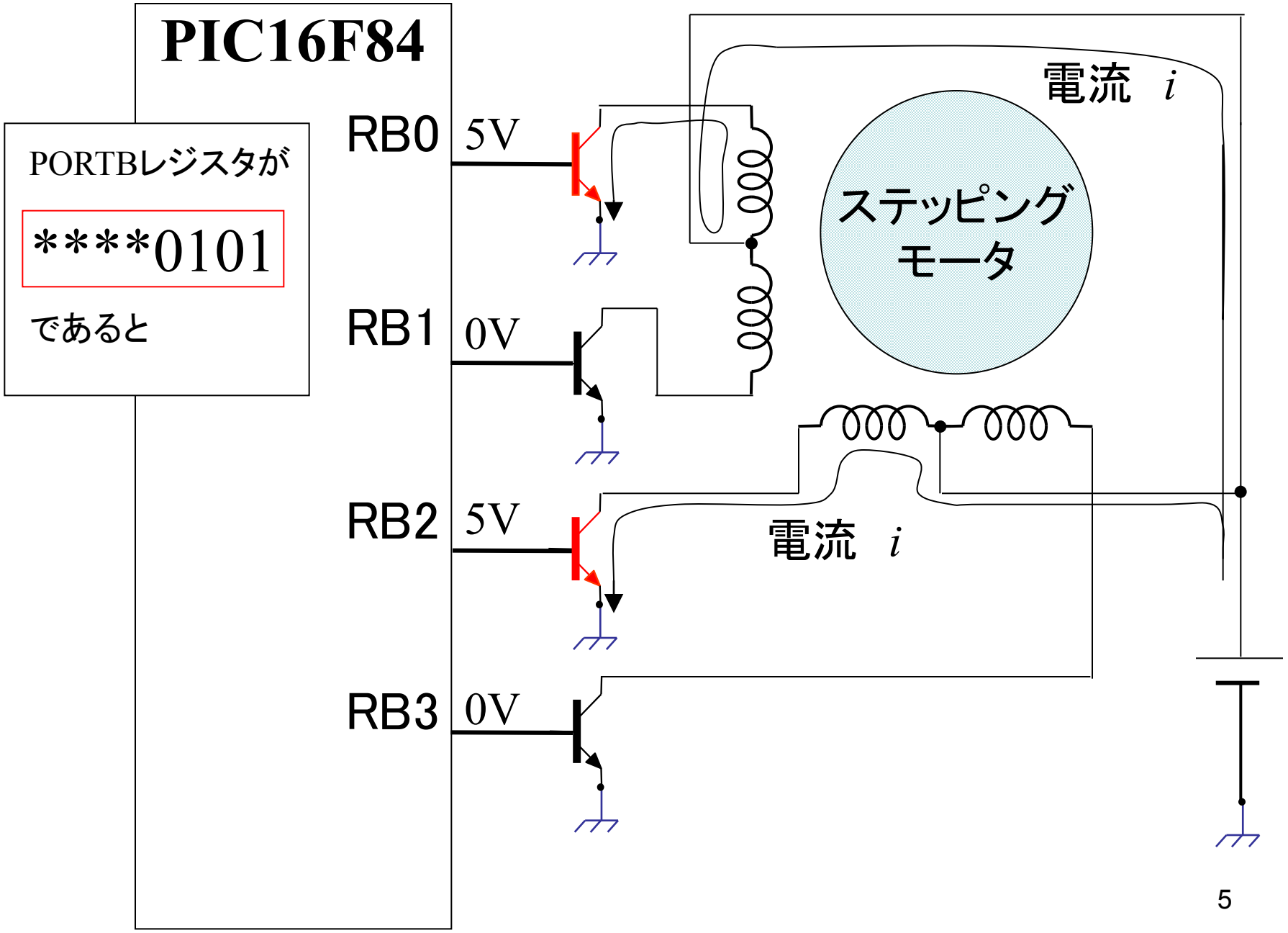
第０章 図28より　完成写真（マイコン回路＋ステッピングモータ駆動回路）

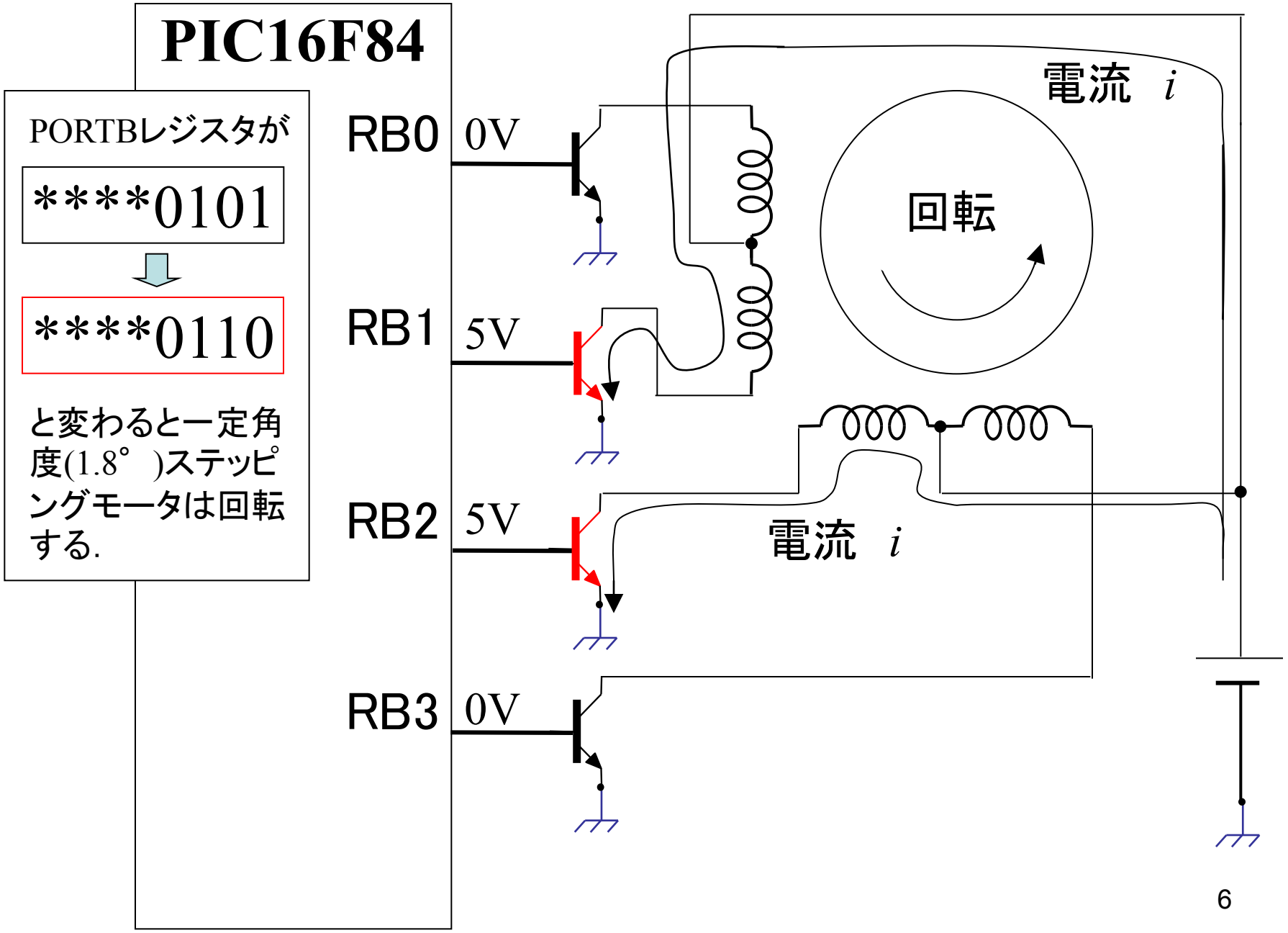
PICマイコンによるステッピングモータの制御

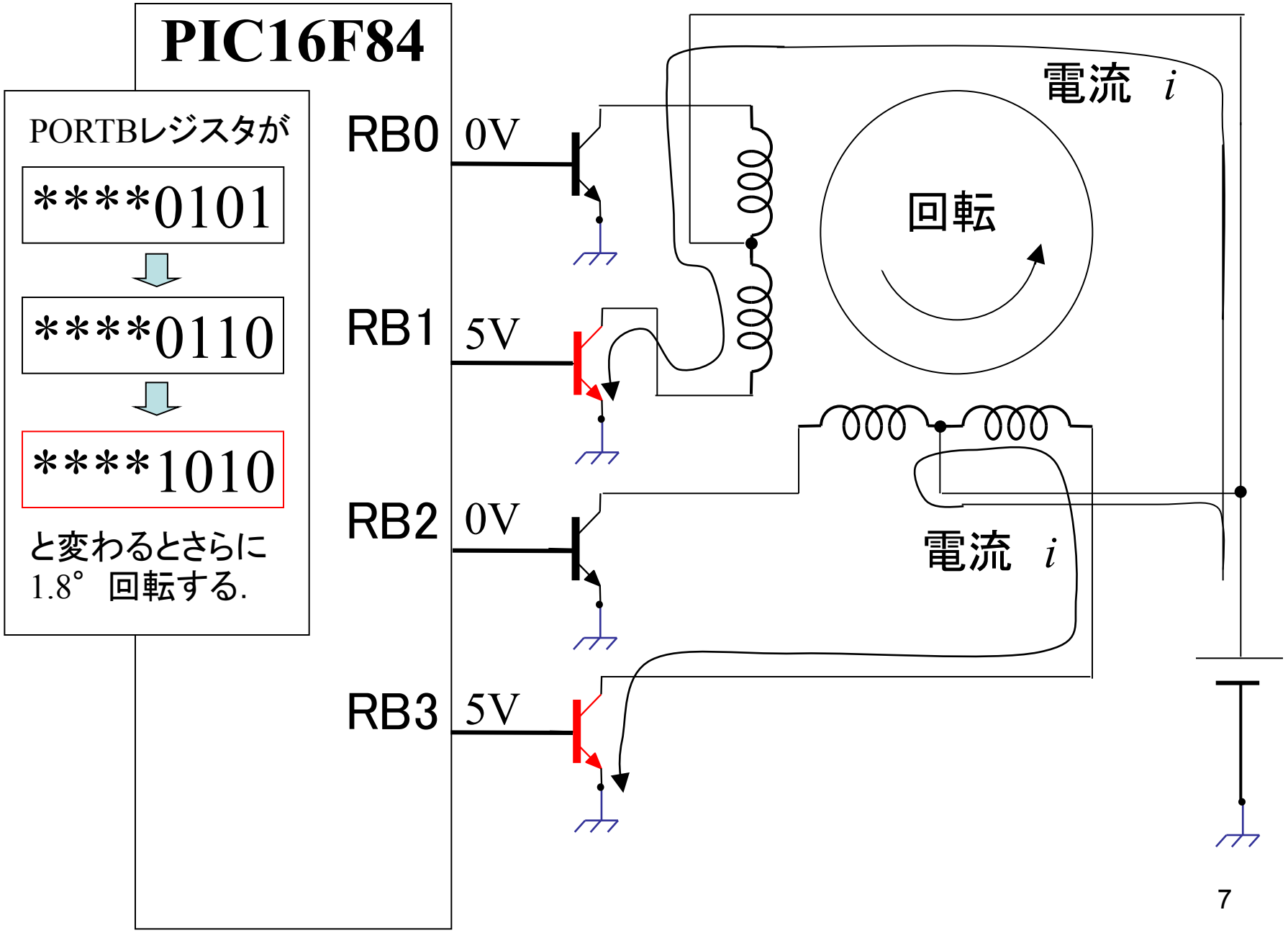


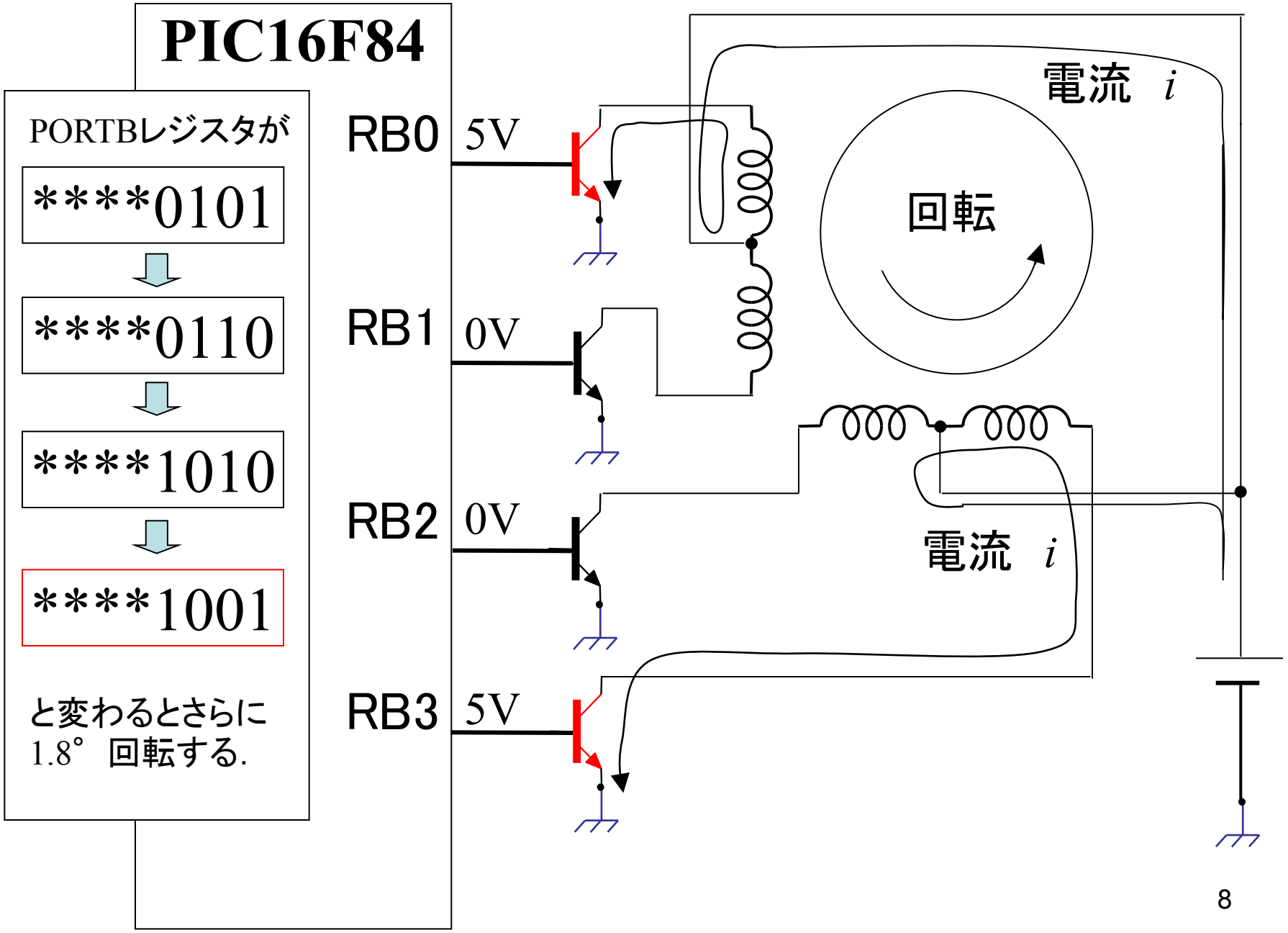
秋月電子通商
PICステッピングモータドライバキット（小型モータ付き）より

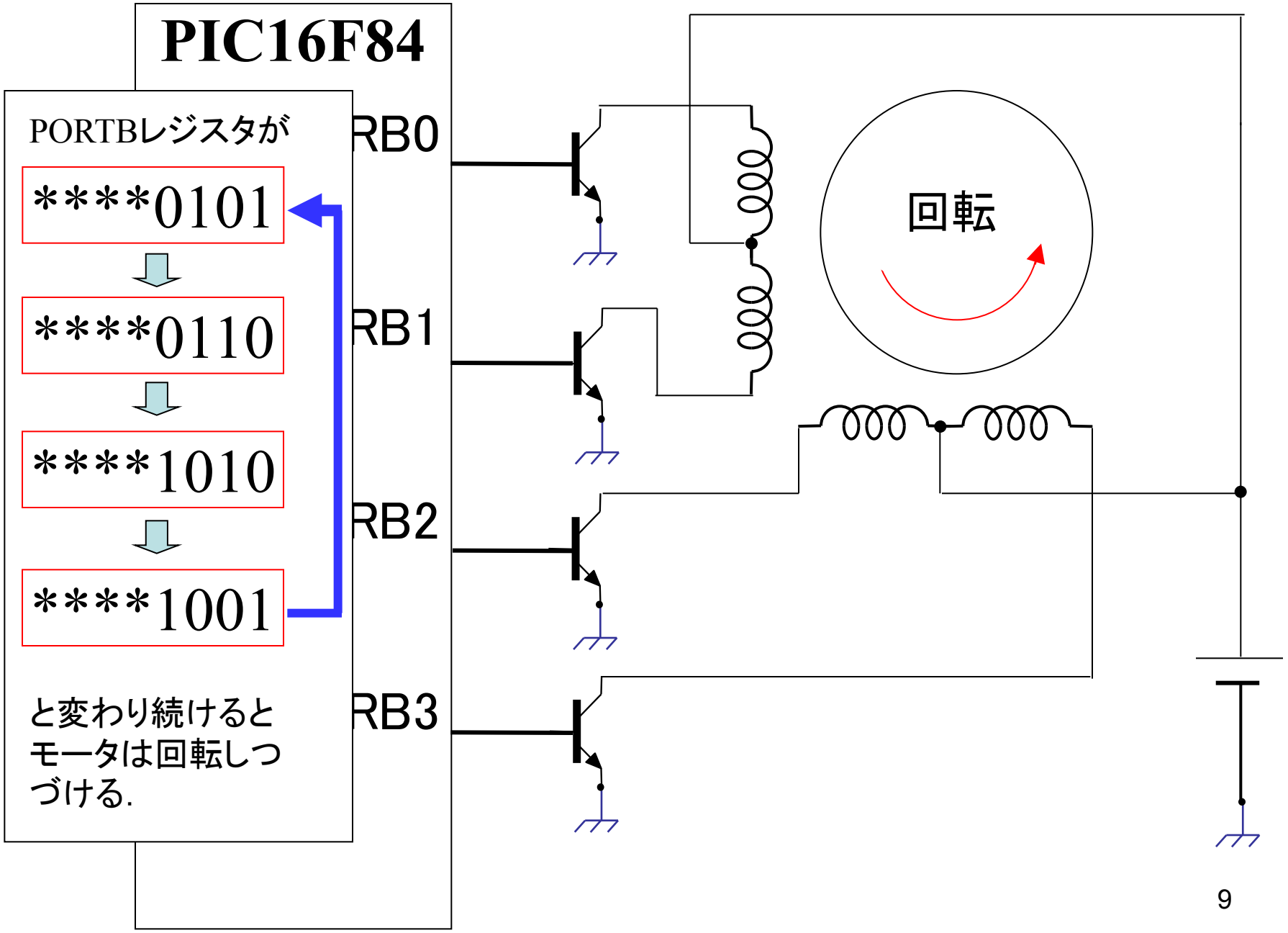

PIC16F84
PORTBレジスタが
*****1***
であると
RB0
RB1
RB2
RB3
トランジスタがオンとなり電流iが流れる.
ステッピングモータ
電流 i
5V
DC 8～15V

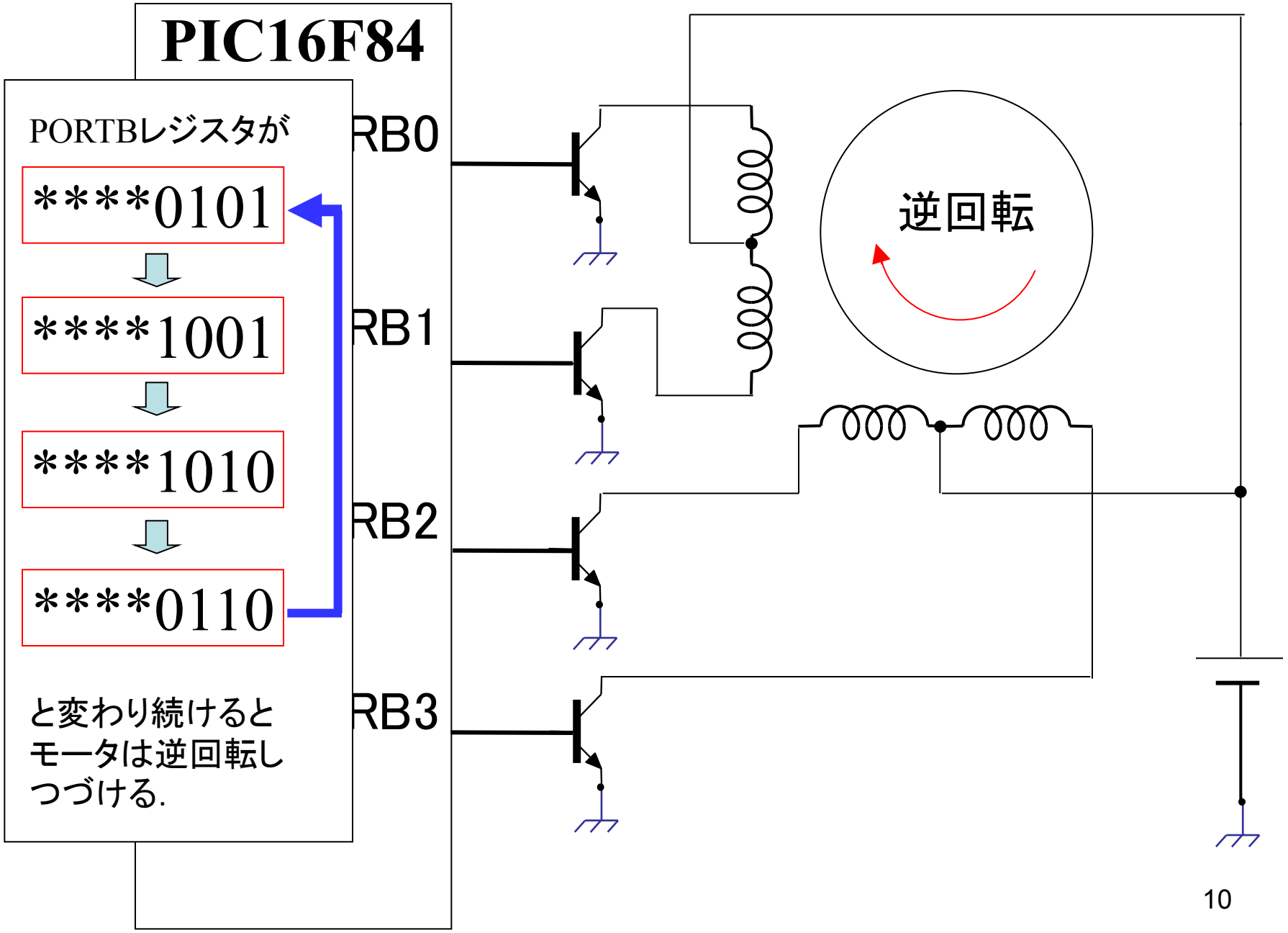

PIC16F84
PORTBレジスタが
****0101
であると
RB0
5V
RB1
0V
RB2
5V
RB3
0V
電流 i
ステッピング
モータ
電流 i


PIC16F84
PORTBレジスタが
****0101
****0110
と変わると一定角度(1.8°)ステッピングモータは回転する.
RB0
0V
RB1
5V
RB2
5V
RB3
0V
電流 i
回転
電流 i


PIC16F84
PORTBレジスタが
****0101
****0110
****1010
と変わるとさらに
1.8° 回転する.
RB0
0V
RB1
5V
RB2
0V
RB3
5V
電流 i
回転
電流 i


PIC16F84
PORTBレジスタが
****0101
****0110
****1010
****1001
と変わるとさらに
1.8° 回転する.
RB0
5V
RB1
0V
RB2
0V
RB3
5V
電流 i
回転
電流 i


PIC16F84
PORTBレジスタが
****0101
****0110
****1010
****1001
と変わり続けると
モータは回転しつ
づける.
RB0
RB1
RB2
RB3
回転


PIC16F84
PORTBレジスタが
****0101
****1001
****1010
****0110
と変わり続けると
モータは逆回転し
つづける.
RB0
RB1
RB2
RB3
逆回転

8.1 ステッピングモータの定速制御

このソースファイルを打ち込んで下さい.
注意 CONFIGの前のアンダーバーは2つあります.

```
;Stepping Motor Control Program

            INCLUDE"p16F84.inc"
            list p=16F84

            __CONFIG  _HS_OSC  &  _WDT_OFF  &  _PWRTE_OFF  &  _CP_OFF

Memory      EQU       0x0C
MEM1        EQU       Memory+0          ;MEM1  at 0C
TIME1       EQU       Memory+1          ;TIME1 at 0D
TIME2       EQU       Memory+2          ;TIME2 at 0E
TIME3       EQU       Memory+3          ;TIME3 at 0F

            ORG       0
            GOTO      START             ;Main Program starts at START

            ORG       4


START

;Setting of Port B
            BSF       STATUS,RP0        ;Selection of Bank 1
            MOVLW     B'11000000'
            MOVWF     TRISB             ;RB0-6 -> Output Port, RB6,7 -> Input Port
            BCF       STATUS,RP0        ;Selection of Bank 0

            MOVLW     B'00000101'       ;'00000101' -> (W)
            MOVWF     PORTB             ;(W) -> (PORTB)
```

ポートBの設定

ポートBの下位4桁に0101を出力

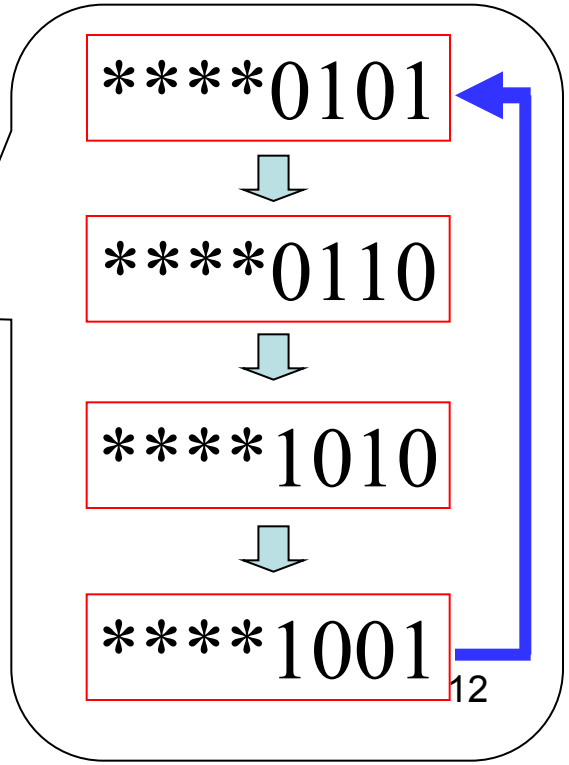
```
; Main Program

STEP1       MOVF      PORTB,0          ;(RB)->(W)
            ANDLW     B'11000000'      ;(W) and 11000000 -> (W)
            MOVWF     MEM1             ;(W) -> (MEM1)

            BTFSS     MEM1,7           ;If the 7th bit = 1, Then skip
            CALL      RotateR
            BTFSS     MEM1,6           ;If the 6th bit = 1, Then skip
            CALL      RotateL
            GOTO      STEP1

;End of Main Program


;Sub Routine1
RotateR     MOVLW     B'00000101'      ;'00000101' -> (W)
            MOVWF     PORTB            ;(W) -> (PORTB)
            CALL      COUNT1
            MOVLW     B'00000110'
            MOVWF     PORTB
            CALL      COUNT1
            MOVLW     B'00001010'
            MOVWF     PORTB
            CALL      COUNT1
            MOVLW     B'00001001'
            MOVWF     PORTB
            CALL      COUNT1
            RETURN
```

ポートBの上位2桁(RB6,7)をMEM1へ

SW1がon (RB7 = 0)でサブルーチン1をCALL

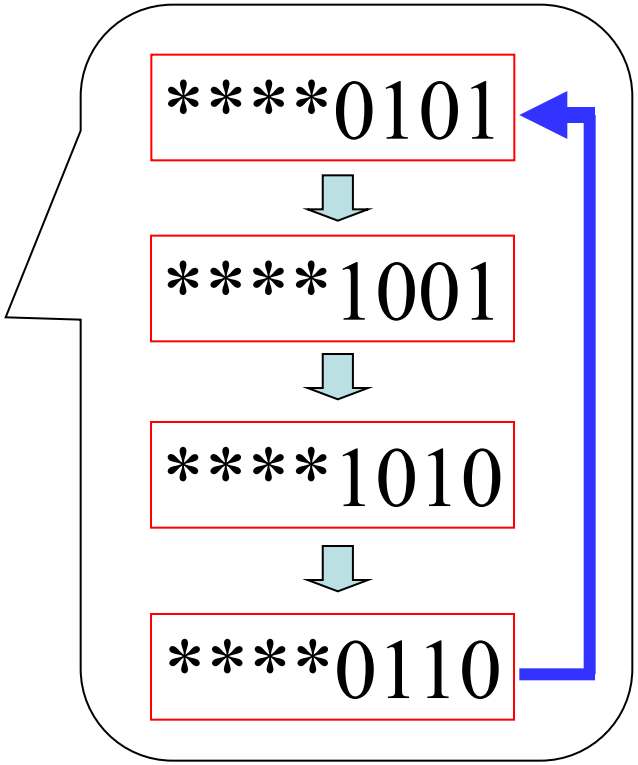
SW2がon (RB6 = 0)でサブルーチン2をCALL

サブルーチン1
右回転



```
;Sub Program2
RotateL     MOVLW     B'00000101‘
            MOVWF     PORTB
            CALL      COUNT1
            MOVLW     B'00001001'
            MOVWF     PORTB
            CALL      COUNT1
            MOVLW     B'00001010'
            MOVWF     PORTB
            CALL      COUNT1
            MOVLW     B'00000110'
            MOVWF     PORTB
            CALL      COUNT1
            RETURN
```

サブルーチン2
左回転




```
;Idling timer
COUNT1      MOVLW     0x16
            MOVWF     TIME1
STEPM       MOVWF     TIME2
STEPM1      MOVWF     TIME3
STEPM2      DECFSZ    TIME3,1
            GOTO      STEPM2
            DECFSZ    TIME2,1
            GOTO      STEPM1
            DECFSZ    TIME1,1
            GOTO      STEPM
            RETURN

            END
```

時間稼ぎルーチン

マイコンの動作が速すぎる（1サイクル = $1\mu s$）のでこれは時間を稼ぐだけのプログラム．
シミュレーションでは0x16を0x02として，このプログラムの動きを確認して下さい．
実際にステップモータを動かすときには，0x0F以上でないとマイコンの出力の変化が速すぎてステップモータは動かないので注意して下さい．

8.2 ステッピングモータの速度制御を学ぼう

8.2 ステッピングモータの速度制御を学ぼう

```
;Stepping Motor Control Program(Speed Control using RC)
            INCLUDE"p16F84.inc"
            list p=16F84

            __CONFIG  _HS_OSC  &  _WDT_OFF  &  _PWRTE_OI

Memory      EQU       0x0C
OutMode     EQU       Memory+0    ;OutMode
TIME1       EQU       Memory+1    ;TIME1

            ORG       0
            GOTO      START                   ;Main Program starts at START

            ORG       4

START

            CALL      PortSet                 ;Port setting routine

            CALL      IniSet                  ;Initial setting routine

; Main Program

STEP1       CALL      C5                      ;Discharge and charge of C5 routine

            CALL      Rotate                  ;Rotating Step Motor

            GOTO      STEP1

;End of Main Program
```

以下のプログラムを打ち込んで，PICマイコンに書き込み，実際にステップモータの回転数を制御できることを確認してください。

ポートの設定

レジスタ，ポートの初期値の設定

コンデンサC5の放電，充電

ステップモータの回転

```
;Port Setting Sub-Routine
PortSet     BSF     STATUS,RP0      ;Selection of Bank 1

            MOVLW   B'00000010'
            MOVWF   TRISA           ;RA0 -> Output Port, RA1->Input Port

            MOVLW   B'11000000'
            MOVWF   TRISB           ;RB0-6 -> Output Port, RB6,7 -> Input Port

            BCF     STATUS,RP0      ;Selection of Bank 0

            RETURN


;Initial Setting Sub-Routine
IniSet      MOVLW   B'00000101'     ;'00000101' -> (W)
            MOVWF   PORTB           ;(W) -> (PORTB)

            MOVLW   B'00000000'
            MOVWF   OutMode

            RETURN
```

RA0 → 出力ポート，RA1 → 入力ポート

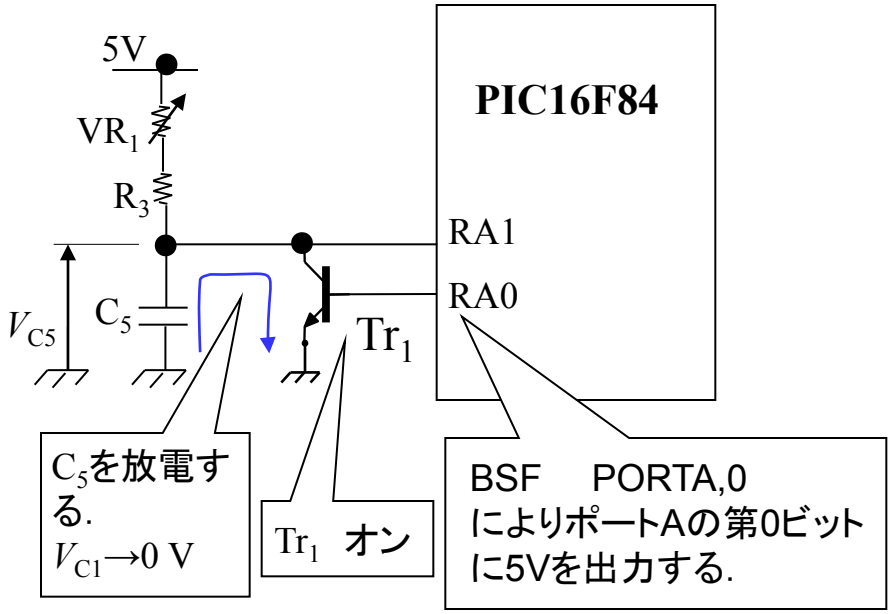
RB0～5 → 出力ポート，RB6, 7 → 入力ポート

ステップモータへの初期出力設定（4通りの出力のどれでも良いが，ここでは ****0101　としている．）

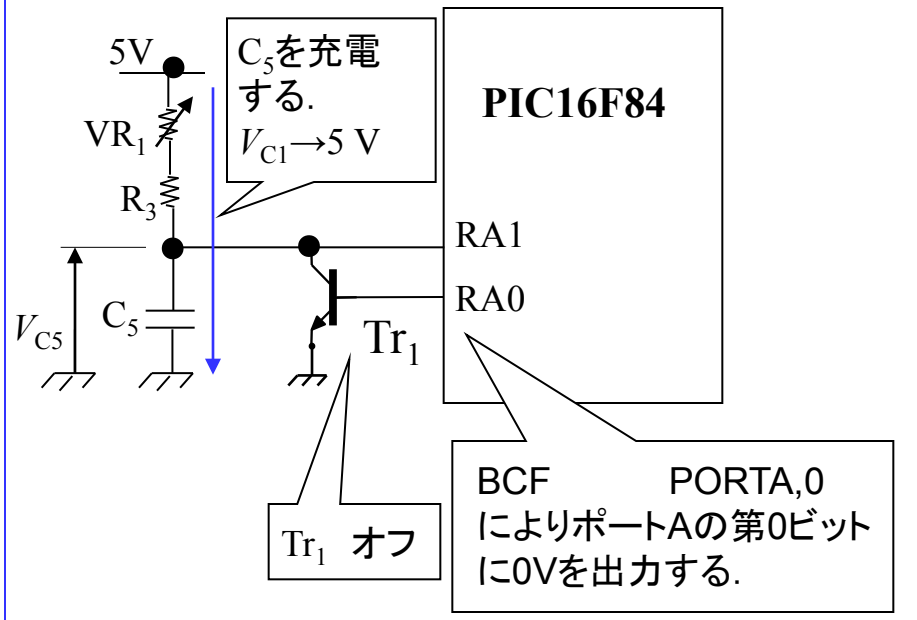
OutMode はステップモータの4通りの出力に番号を付けて，それを出力モードと呼んだもの．

モード番号	PotrBへの出力
0	0101
1	0110
2	1010
3	1001

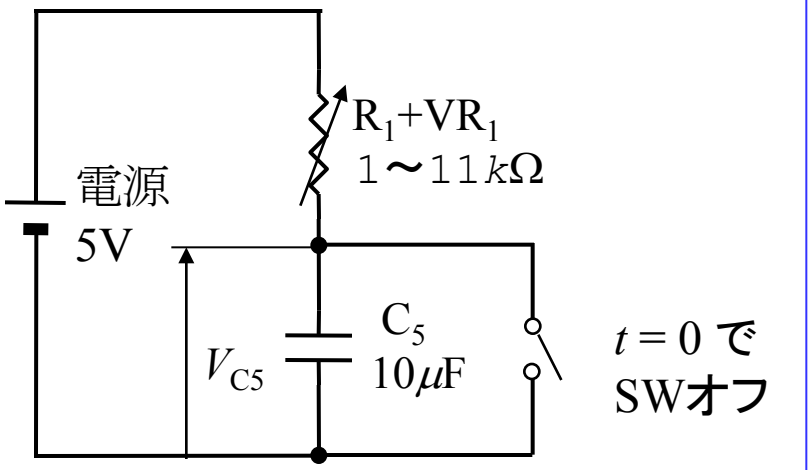
```
;Discharge and Charge of C5 Sub-routine
C5          BSF         PORTA,0         ;1 -> (RA0)
            CALL        Idtimer         ;(Discharge of C5)

            BCF         PORTA,0         ;0 -> (RA0) (Charge of C5)

STEP2       BTFSS       PORTA,1         ;If the 1-th bit = 1, then skip the next command line
            GOTO        STEP2

            RETURN
```



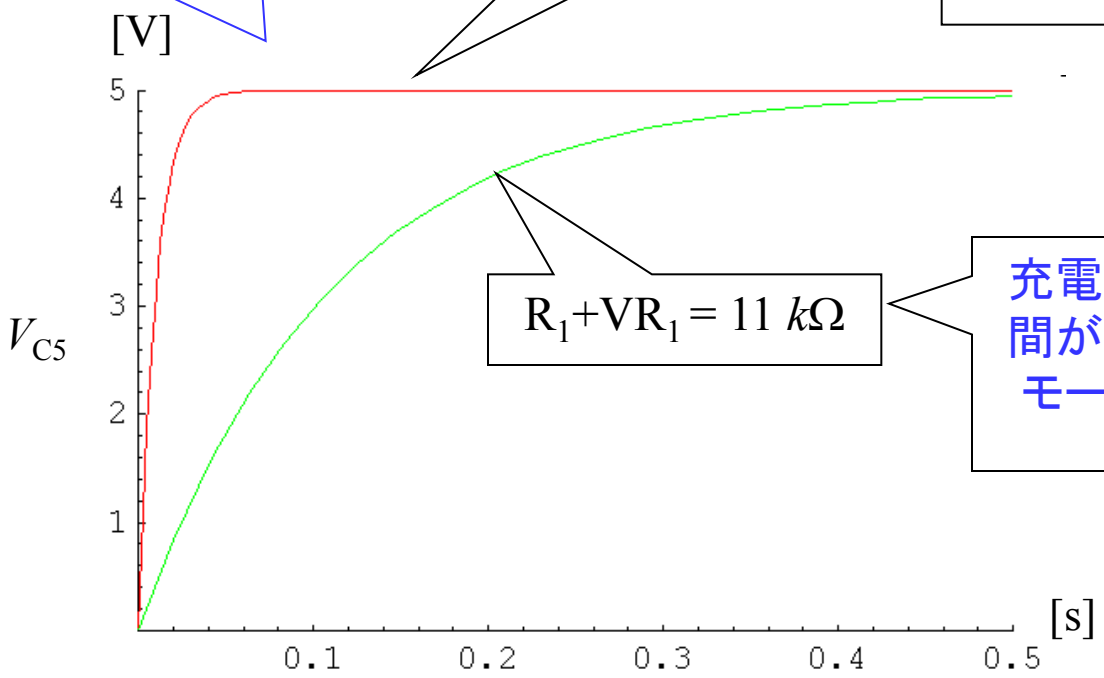
コンデンサC_5の電荷を放出して，コンデンサの両端電圧V_{C5}をゼロとする 命令



コンデンサC_5に電荷を充電して，コンデンサの両端電圧V_{C5}を上昇させる．ポートAの第1ビットが1(約5V)となるまで待つ命令

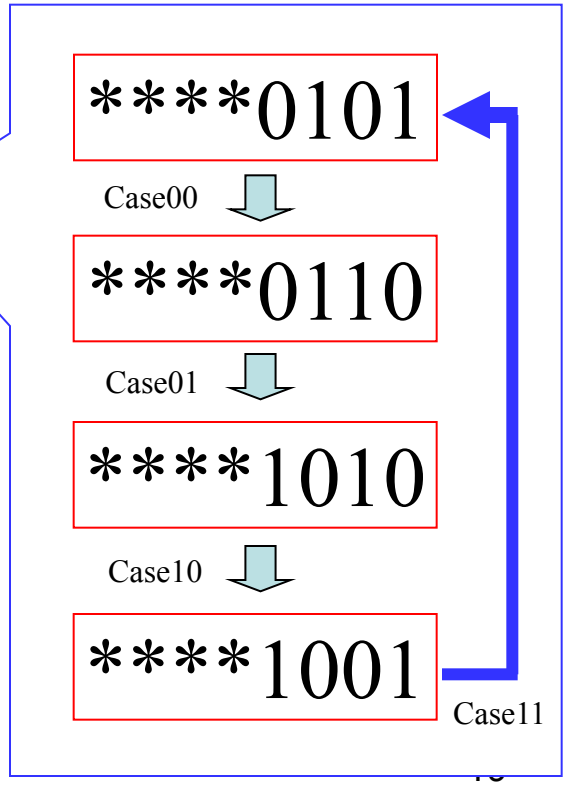


抵抗VR_1の値でコンデンサC_5の両端電圧V_{C5}が5Vになるまでの時間が変わる



V_{C5}のシミュレーション結果

```
;Rotating Step Motor routine
Rotate      BTFSS     OutMode,0    ;If the 0-th bit = 1, then skip the next command line
            GOTO      CaseX0
            GOTO      CaseX1

CaseX0      BTFSS     OutMode,1    ;If the 1-th bit = 1, then skip the next command line
            GOTO      Case00
            GOTO      Case10

CaseX1      BTFSS     OutMode,1    ;If the 1-th bit = 1, then skip the next command line
            GOTO      Case01
            GOTO      Case11

Case00      MOVLW     B'00000110'  ; '00000101' -> '00000110'
            GOTO      Fin

Case01      MOVLW     B'00001010'  ; '00000110' -> '00001010'
            GOTO      Fin

Case10      MOVLW     B'00001001'  ; '00001010' -> '00001001'
            GOTO      Fin

Case11      MOVLW     B'00000101'  ;'00001001' -> '00000101'

Fin         MOVWF     PORTB                    ;(W) -> (PORTB)

            INCF      OutMode,1    ;(OutMode) + 1 -> (OutMode)

            RETURN
```



モード番号の更新　OutModeの下2桁のみ意味がある.
INCFにより順次00→01→10→11→00→・・・と変化する.

0xFF = D’255’

```
;Idling timer
Idtimer     MOVLW     0xFF
            MOVWF     TIME1
STEPM       DECFSZ    TIME1,1
            GOTO      STEPM
            RETURN

            END
```

時間稼ぎルーチン

サイクル数

1		2サイクル
1		
1	Skipの時は2	254×（1＋2)+ 2 ＝764サイクル
2		
2		2サイクル

合計　　768サイクル

1サイクルに4クロックを使用する.

1クロックはセラミック発振子が約4MHzで動作するので

1/4MHz = 0. 25 μs

よって

768×4× 0.25 μs ≒ 770 μs

2004年8月

著者： 古橋武
名古屋大学工学研究科計算理工学専攻
furuhashi@cse.nagoya-u.ac.jp