

1. 序論	- 1 -
1. 1 背景	- 1 -
1. 2 リモートセンシング	- 1 -
1. 3 目的	- 2 -
1. 4 ワンチップマイコン	- 2 -
2. 基本的なプログラムの作成	- 3 -
2. 1 ステッピングモーターを正・逆回転させるプログラム	- 3 -
2. 2 大小判定を行うプログラム	- 6 -
2. 3 可変抵抗のつまみを追従するプログラム	- 9 -
3. フォトダイオードを使用したプログラム	- 13 -
3. 1 システム概要	- 13 -
3. 2 1素子のフォトダイオードを使用した場合	- 14 -
3. 2 4素子を使用した場合	- 17 -
3. 3 許容範囲	- 21 -
4. まとめ・考察・課題	- 23 -
5. プログラムリスト	- 24 -
5. 1 プログラムリスト1	- 24 -
5. 2 プログラムリスト2	- 26 -
5. 3 プログラムリスト3	- 31 -
5. 4 プログラムリスト4	- 37 -
5. 5 プログラムリスト5	- 53 -
謝辞	- 63 -
付録1 2軸システムの設計図	- 64 -
付録2 1素子を使用した場合のラベルについてのフローチャートとラベルの説明	- 67 -
付録3 4素子を使用した場合のラベルについてのフローチャートとラベルの説明	- 71 -
参考文献	- 76 -

1. 序論

1. 1 背景

私たちの生活している現代社会は産業の発達により快適になっている。その反面、産業の発達が原因と考えられる大気汚染, 地球温暖化など地球環境の悪化が懸念されている。地球環境に影響を及ぼす物質のうちの1つにエアロゾルがあり、エアロゾルは黄砂やチョークの粉などの微粒子の事を指す。地表にとどく太陽光線は、エアロゾルによる光の散乱によって減少し、また地表から発生する熱を遮ってしまうので温室効果をもたらす。このためエアロゾルは地球の温暖化・寒冷化の双方に影響を及ぼす。そのため、エアロゾル濃度の計測はそれらのメカニズムを解明する上で重要である

1. 2 リモートセンシング

リモートセンシングとは、離れた場所から対象物を計測・分析する技術のことである。例を挙げると人工衛星ひまわりがある。

一般的な大気のリモートセンシングの概要を図1-1に示す。光源のスペクトルは大気などの影響を受けた後、計測装置によって観測され、応用分野に利用することができる。

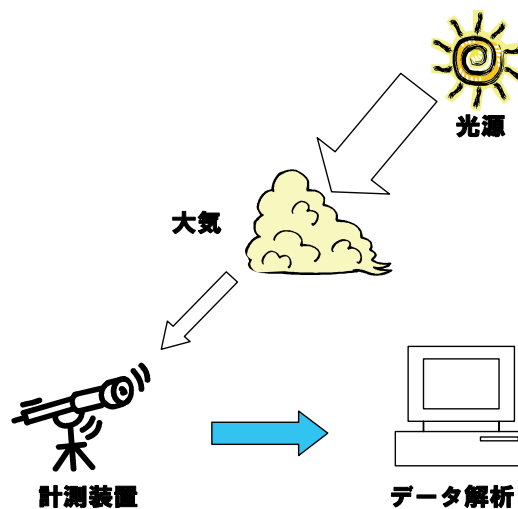


図1-1 大気のリモートセンシング

1. 3 目的

エアロゾル濃度を光学的に測定する際に太陽を光源とすると、人工の光よりも遠方にあるため広域でのリモートセンシングが可能で、海上や山中でも観測が可能であるといった利点がある。しかし、太陽は地球から見て移動しているため、計測装置を常に太陽に向けておくための装置、サントラッカーが必要となる。

本研究は、太陽を光源としたリモートセンシングを行うために必要となるサントラッカーの基礎プログラムの開発を行う。

1. 4 ワンチップマイコン

本研究では、ワンチップマイコンの中で A/D 変換機能を搭載している PIC16F874 を使用した。特徴を以下に示す。

- ・ 2 サイクルである分岐命令を除く全ての命令は 1 サイクルで実行される
- ・ 直接的、間接的あるいは相対的なアドレッシングモードを持つ。
- ・ A/D 変換の分解能は 10 bit
- ・ 命令速度 : 時間入力 DC-20MHz
: 指令サイクル DC-200ns

1	MCLR	RB7	40
2	RA0	RB6	39
3	RA1	RB5	38
4	RA2	RB4	37
5	RA3	RB3	36
6	RA4	RB2	35
7	RA5	RB1	34
8	RE0	RB0	33
9	RE1	V _{DD}	32
10	RE2	V _{SS}	31
11	V _{DD}	RD7	30
12	V _{SS}	RD6	29
13	OSC1	RD5	28
14	OSC2	RD4	27
15	RC0	RC7	26
16	RC1	RC6	25
17	RC2	RC5	24
18	RC3	RC4	23
19	RD0	RD3	22
20	RD1	RD2	21

図 1 - 2 PIC16F874 のピン配置

2. 基本的なプログラムの作成

2. 1 ステッピングモーターを正・逆回転させるプログラム

ステッピングモーターはパルスを入力することによって、軸をステッピングモーター固有のステップ角で回転させることができるので、位置制御の用途で多用されている。

使用したステッピングモーター (VEXTA PK243-01A) は 2 相励磁方式であり、正転 (軸を上から見て時計回り) の場合は図 2-1 の励磁シーケンスのように $\bar{Y}X, XY, Y\bar{X}, \bar{X}\bar{Y}$ の順にパルスを入力する。逆転 (反時計回り) の場合は図 2-1 のように、正転の場合での X と \bar{X} を逆に入力する。図 2-1・2-2 に示す励磁シーケンスは、2 相励磁方式のステッピングモーター励磁シーケンス図であり、図に示した流れのとおり黒く塗りつぶされた部分でステッピングモータードライブ回路に 5 V を印加することで、ステッピングモーターを正・逆転することができる。

図 2-3 にステッピングモータードライブ回路とワンチップマイコンを接続した回路図を示す。トランジスタにはモーター駆動のために大電流が必要となるため、ダーリントントランジスタを使用し、ダイオードはモーター ON 時に逆起電流が生じてしまうため、整流用に使用した。

正転させた場合のプログラムのフローチャートを図 2-4 に、ステッピングモーターを回転させるプログラムをプログラムリスト 1 示す。逆転させる場合は CALLKAI の中で呼び出すサブルーチンを GYAKUTEN にする。

また、パルスの幅を出すサブルーチンには HABA1, HABA2, HABA3 の 3 つを作成し、HABA1 を呼び出すと 0.1ms、HABA2 を呼び出すと 10ms、HABA3 を呼び出すと 500ms となるよう作成した。またそれぞれのサブルーチン内の変数に代入する値を決めることで、パルス幅を任意に設定することができる。

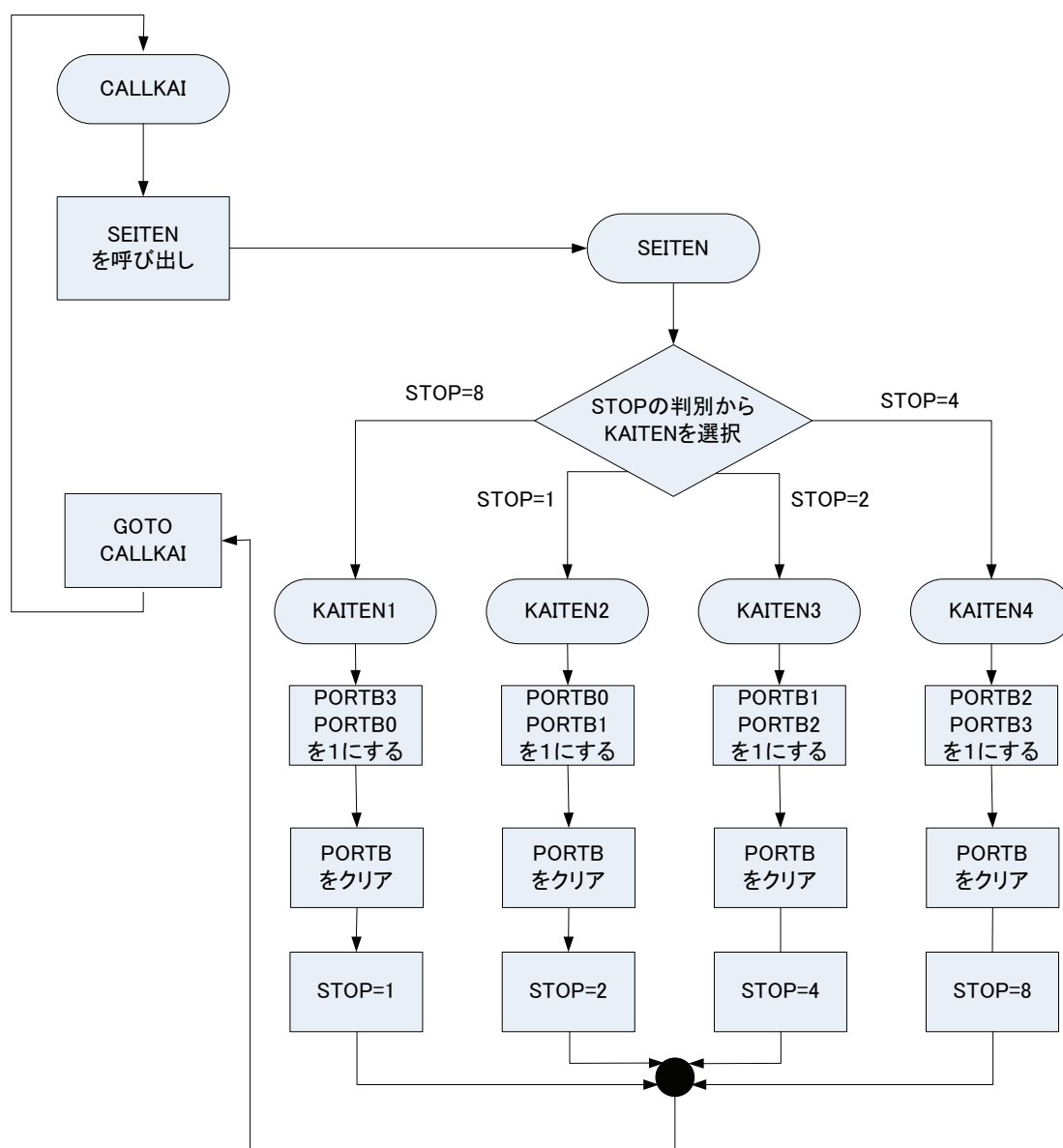


図 2-4 正転させた場合のフローチャート

ステッピングモーターを正・逆転するプログラムでは、呼び出された SEITEN, GYAKUTEN 内で KAITEN1~4 の中で適切なものを選んで移動し、移動したラベル内では PORTB からの出力を行ったのち、STOP という変数に値を代入する。ステッピングモータードライブ回路への出力を行う PORTB0~3 は、それぞれステッピングモータードライブ回路の X, Y, \bar{X}, \bar{Y} に対応する。

図 2-1 に示した正転の場合の励磁シーケンスのように、ステッピングモーターへの前回の出力を図 2-1 の “0” のパルス(KAITEN1)、として、正転させる場合、次に出力するパルスは図 2-1 の “1” のパルス、つまり KAITEN2 を呼び出してパルスを出力することになる。STOP に与える変数は 1, 2, 4, 8 で、1 bit のみの判定で KAITEN1~4 のラベルを選べる値とし

た。STOPは初期設定では1に設定する。

ステッピングモーターを正・逆回転させた場合のワンチップマイコンからの出力波形を図2-5・2-6に示す。図2-5・2-6の波形はそれぞれ2相励磁方式の励磁シーケンスと同様となった。またパルス幅が10msでプログラム通りであることが確認できた。

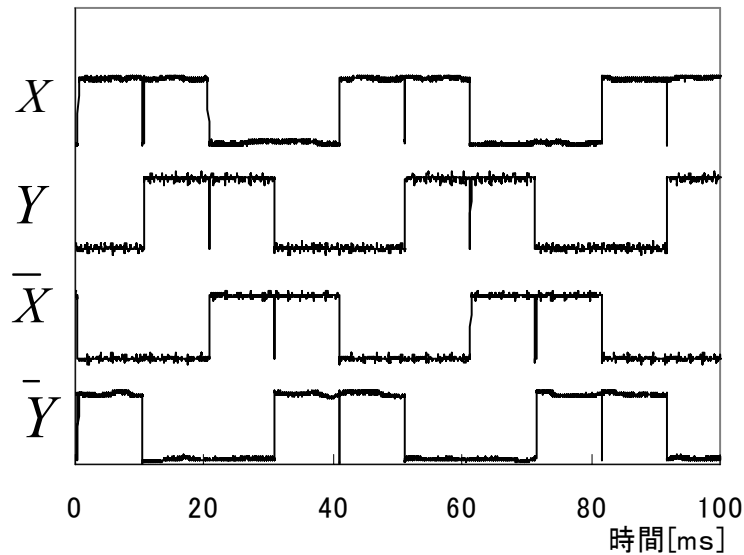


図2-5 正転の場合のワンチップマイコンからの出力波形

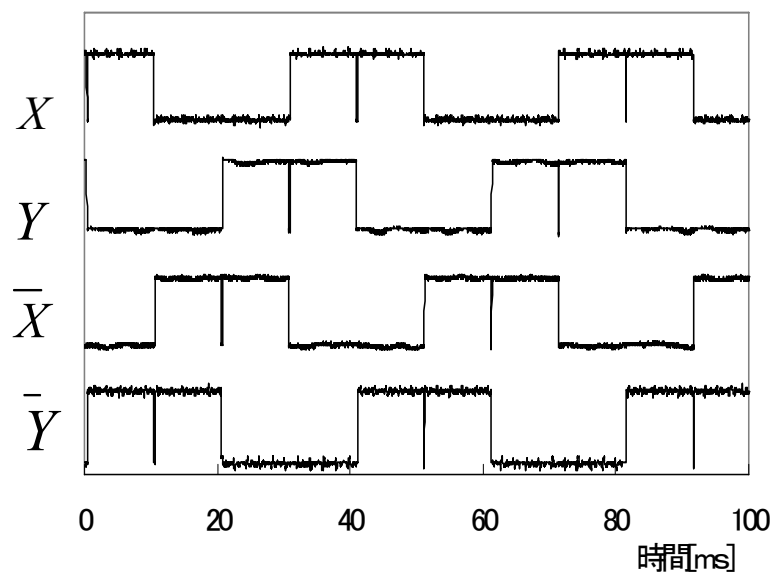


図2-6 逆転の場合のワンチップマイコンからの出力波形

2. 2 大小判定を行うプログラム

アナログポートRA0に可変電圧を入力する。入力電圧が以前の値より大きくなると正転, 小

さくになると逆転するプログラムを作成した。回路図を図 2-7 に、プログラムをプログラムリスト 2 に、プログラムのフローチャートを図 2-8 に示す。

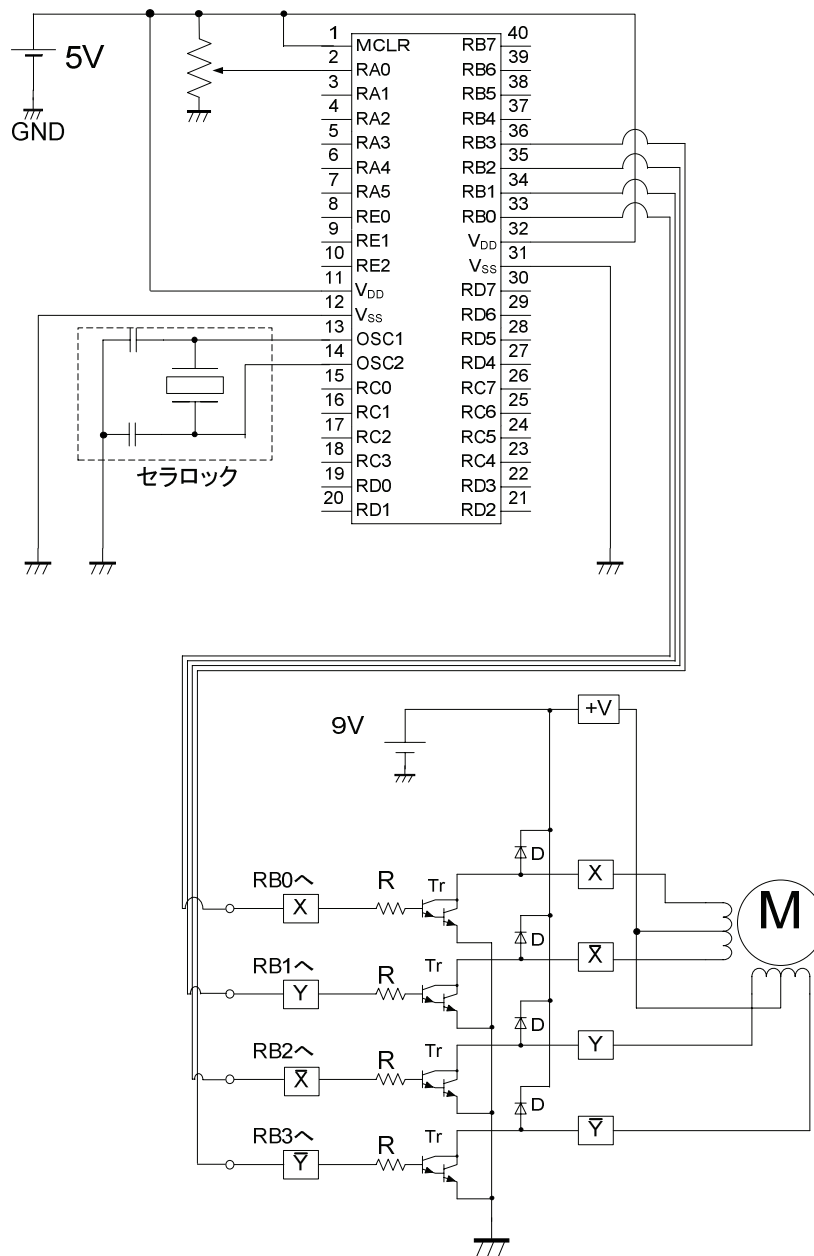


図 2-7 入力判定するステッピングモーター駆動回路

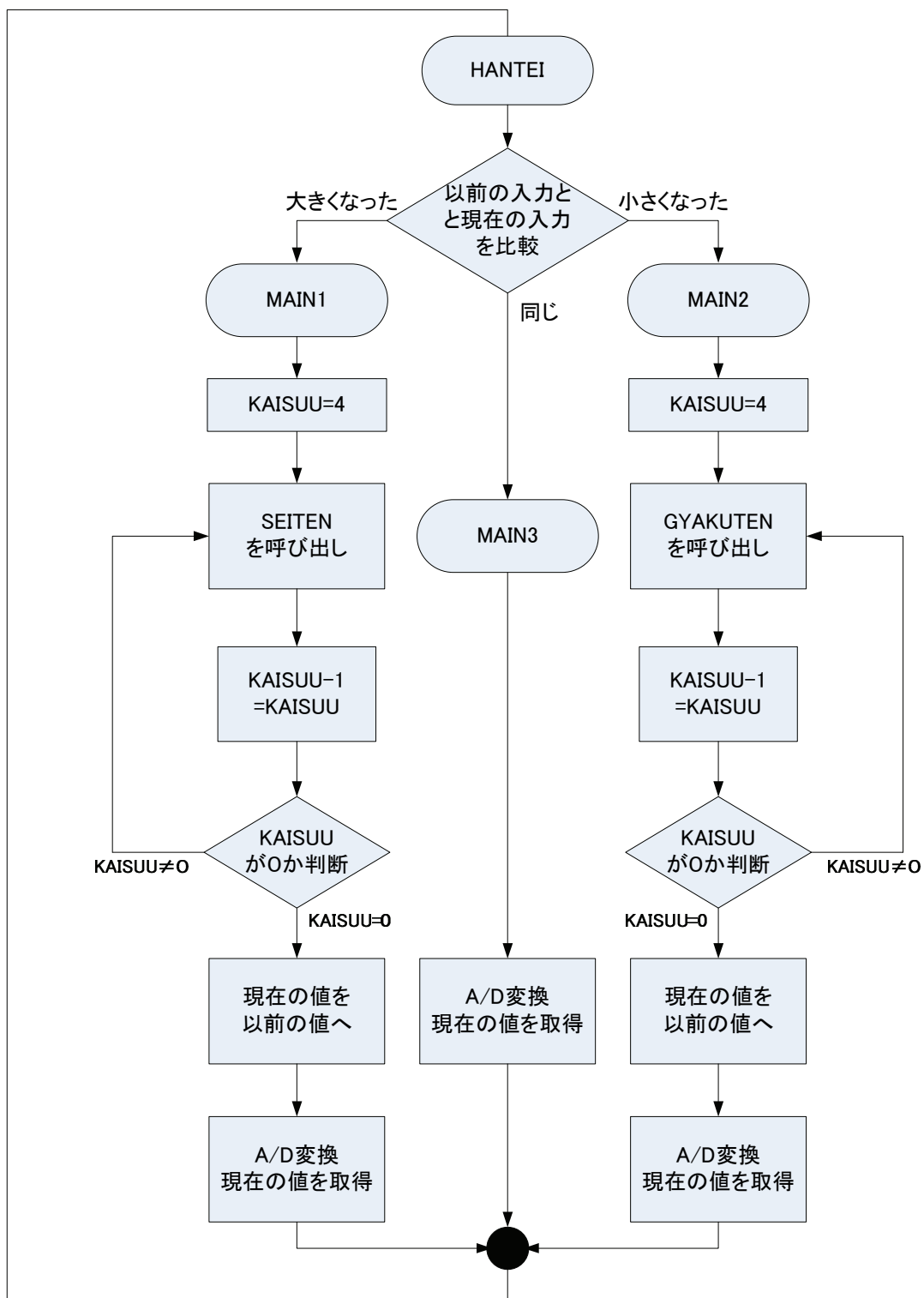


図 2-8 入力電圧の大小判定を行うプログラムのフローチャート

現在の入力電圧 (KEI) と以前の入力電圧 (KIOKU) の大小関係の判定は HANTEI で行っている。KEI が大きければ正転するので MAIN1 へ移動し、KIOKU が大きければ逆転するので MAIN2 へ移動する。また同じ場合は MAIN3 へ移動する。

KAISUU は SEITEN, GYAKUTEN を呼び出す回数を決定する変数である。

プログラムの大小判定部分 HANTEI で使用した変数は KEI と KIOKU で、結果によって移動するラベルは MAIN1 と MAIN2 および MAIN3 ある。作成した HANTEI は、判定したい 9 ～ 16 bit の 2 つの変数と、結果によって移動させたいラベルを 3 ヶ所与えると、任意の 2 変数で大小判定し、任意のラベルへ移動させることができる。

2. 3 可変抵抗のつまみを追従するプログラム

可変抵抗のつまみを約 180° 回転させると、ステッピングモーターを 0 ～ 360° 回転するプログラムを作成した。回路は図 2-7 を使用した。またプログラムをプログラムリスト 3 に示す。作成したプログラムはプログラムリスト 2 のプログラムの MAIN1, MAIN2 内に計算を行う過程を追加したプログラムである。MAIN1 内のフローチャートを図 2-9 に、MAIN2 内のフローチャートを図 2-10 に示す。

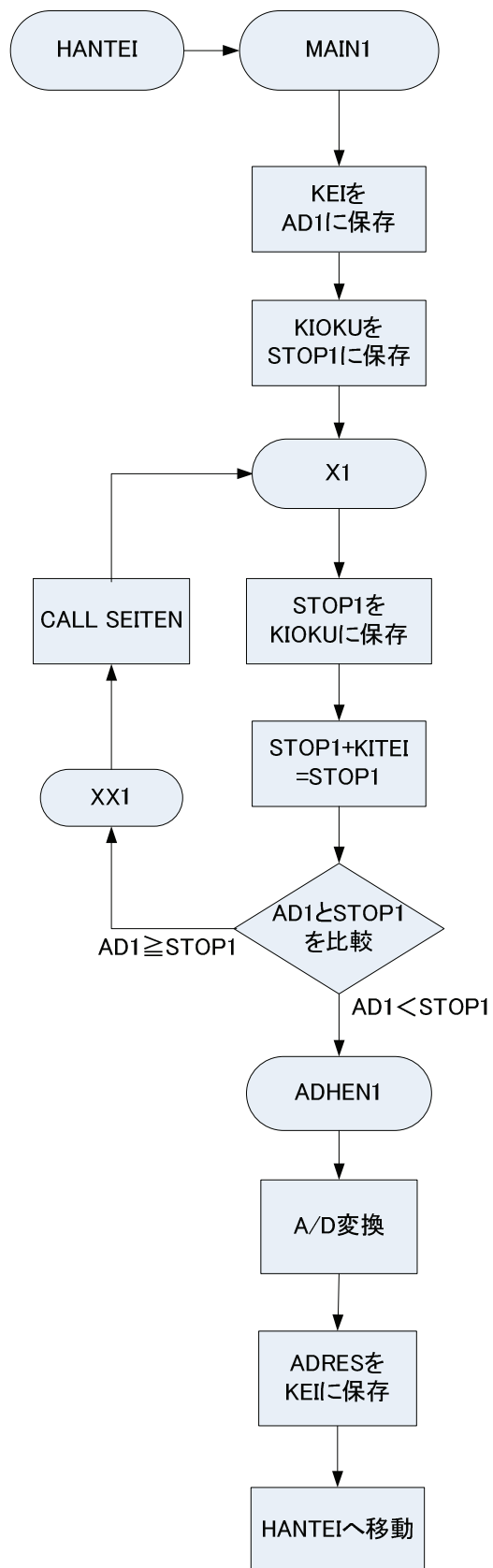


図 2 - 9 MAIN1 のフローチャート

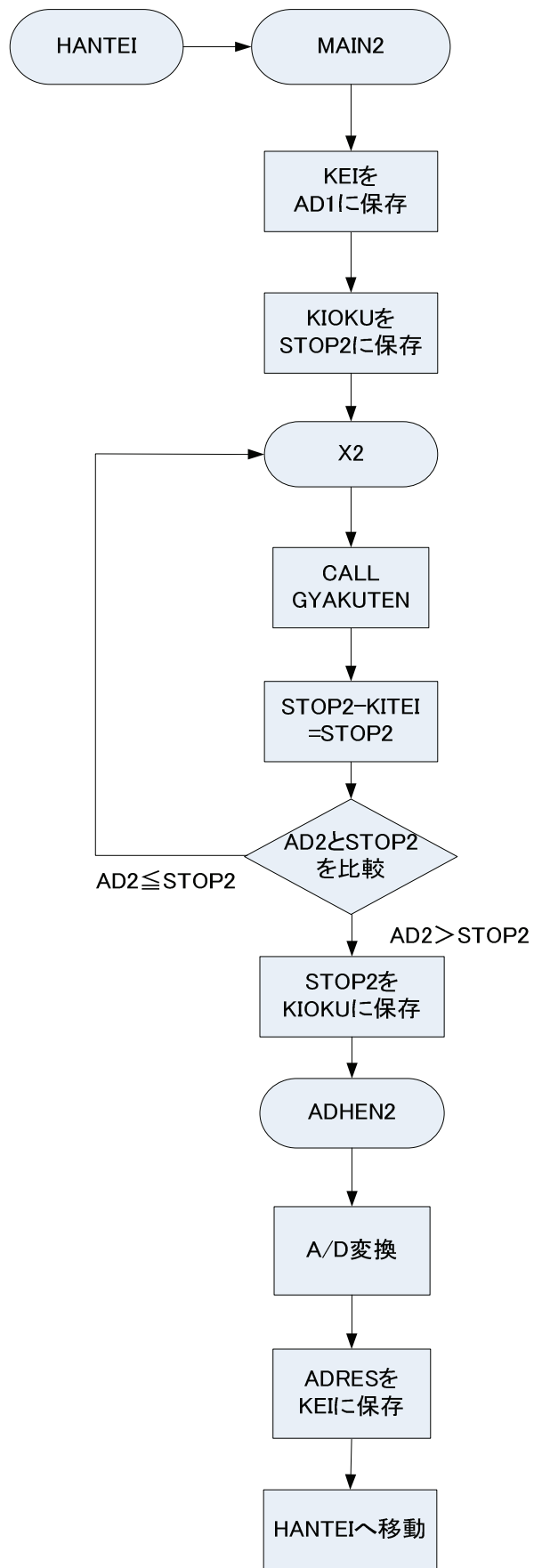


図 2 - 1 0 MAIN2 のフローチャート

16F874のADRESは10bit(10進数で1024)、入力電圧の最大値が5V、デジタル値1は入力電圧の約5mVに相当する。またステッピングモーターのステップ角が1.8°なので、360°回転させるためには200回パルスを送る。したがって、デジタル値が5変化すると、ステップ角1.8°移動することになるため、KITEI(1ステップに相当するデジタル値)を5に設定すればよいことになり、KIOKU、STOP1・2は常にKITEIの倍数となる。

STOP1・2はMAIN1・MAIN2内での加算・減算を行う際に使用する変数である。KIOKUはHANTEI内で大小判定を行うために使用すると同時に、STOP1・2の値を保存しておく変数である。

3. フォトダイオードを使用したプログラム

3. 1 システム概要

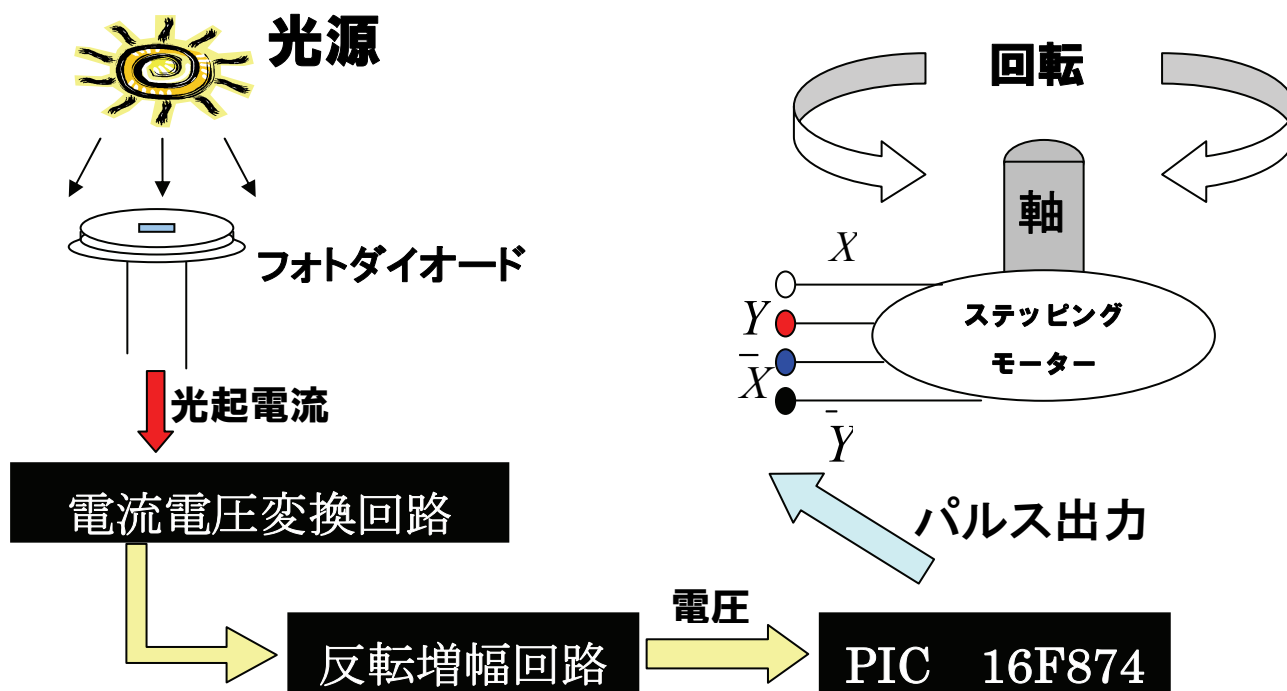


図3-1 システム概要

システムの概要を図3-1に示す。光がフォトダイオードの素子に入射することで光起電流が生じる。生じた光起電流を電流電圧変換回路と反転増幅回路によって変換・増幅し、得られた正電圧をワンチップマイコン（PIC16F874）に入力し、入力電圧から光源を追従するために必要な処理を判定し、ステッピングモーターにパルスを出力して軸を回転させる。

光起電流は入射する光の強度に比例し、フォトダイオードの感度は波長により異なる。フォトダイオードは、Si PIN 接合型4分割である浜松ホトニクス株式会社製のS4346を使用した。4分割フォトダイオードは1個分のサイズで4つの入力電圧を得ることができる。

図3-2に電流電圧変換回路と反転増幅回路を接続した回路図を示す。左のOPアンプを使用した回路が電流電圧変換回路で、入力された電流を変換係数 $-R_f$ で負電圧に変換する。右のOPアンプを使用した回路が反転増幅回路で、入力された電圧を $-R_2/R_1$ の増幅率で反転増幅する。電流電圧変換回路への入力電流と、反転増幅回路の出力電圧の関係は

$$V_{out} = \frac{R_2}{R_1} R_t I_{in}$$

となり、正電圧 V_{out} を得ることができる。

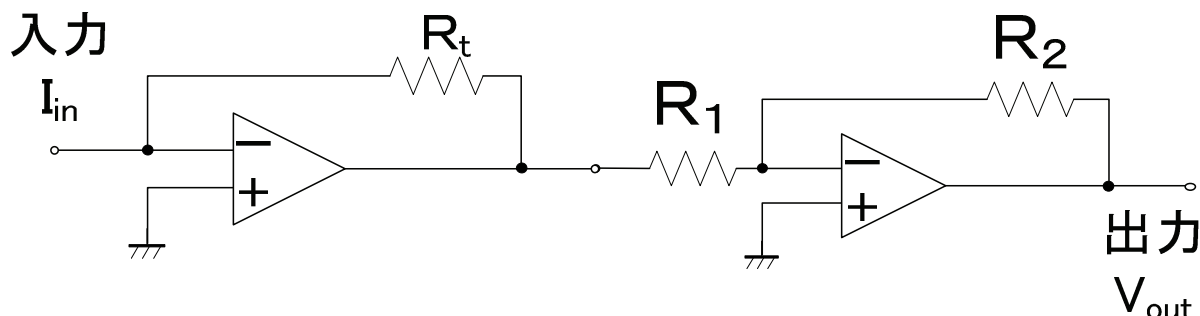


図 3 - 2 電流電圧変換回路と反転増幅回路

OP アンプは 1 つの IC にフォトダイオードの素子数と同じ 4 つの増幅回路をもつ LM324 を使用した。

光源を追従するための 2 軸システムを製作した。設計図を付録 1 に示す。またステッピングモーターは θ 軸方向の制御には VEXTA PH264-02, ϕ 軸方向の制御には VEXTA PK243-01A を使用した。

3. 2 1 素子のフォトダイオードを使用した場合

簡素なシステムのプログラム作成を行った。

プログラムの動作説明図を図 3 - 3 に示す。1 素子のフォトダイオードを使用する場合、強度の変化しか計測することができず、光源がどの位置にあるのか判別することができないため、いくつかの方向で入力電圧を得る必要がある。作成したプログラムではフォトダイオードを図のように 5 つの方向へ移動させ、それぞれの点で入力電圧を取得する。5 つの方向のうち光源に最も近い方向では、フォトダイオードの検出する光強度が最も強い。すなわち 5 つの方向のうち入力電圧が最も大きい方向が、光源の方向に近いということである。5 つの入力電圧を比較し、最も大きい入力電圧が得られた方向へ移動する。この動作を繰り返し行うことで、光源のある方向

へ移動する。この動作を 2 軸にわたって行い光源の位置を検出する。

光源の位置が移動した場合、フォトダイオードに入射する光の強度が変化し、入力電圧が小さくなる。静止した時の入力電圧と比べて、現在の入力電圧が許容範囲外であるとき、プログラムでは光源の位置が変化したとみなし、光源を探す動作を行うプログラムへ移動する。

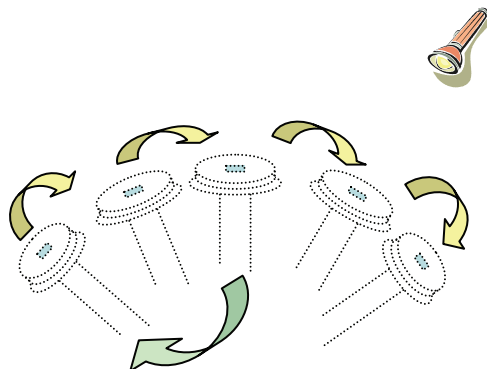


図 3 - 3 1 素子を使用した場合の動作説明図

使用した回路図を図 3 - 4 に、プログラムをプログラムリスト 4 に、プログラムの簡単なフローチャートを図 3 - 5 に、プログラムのラベルについてのフローチャートとラベルの説明を付録 2 に示す。

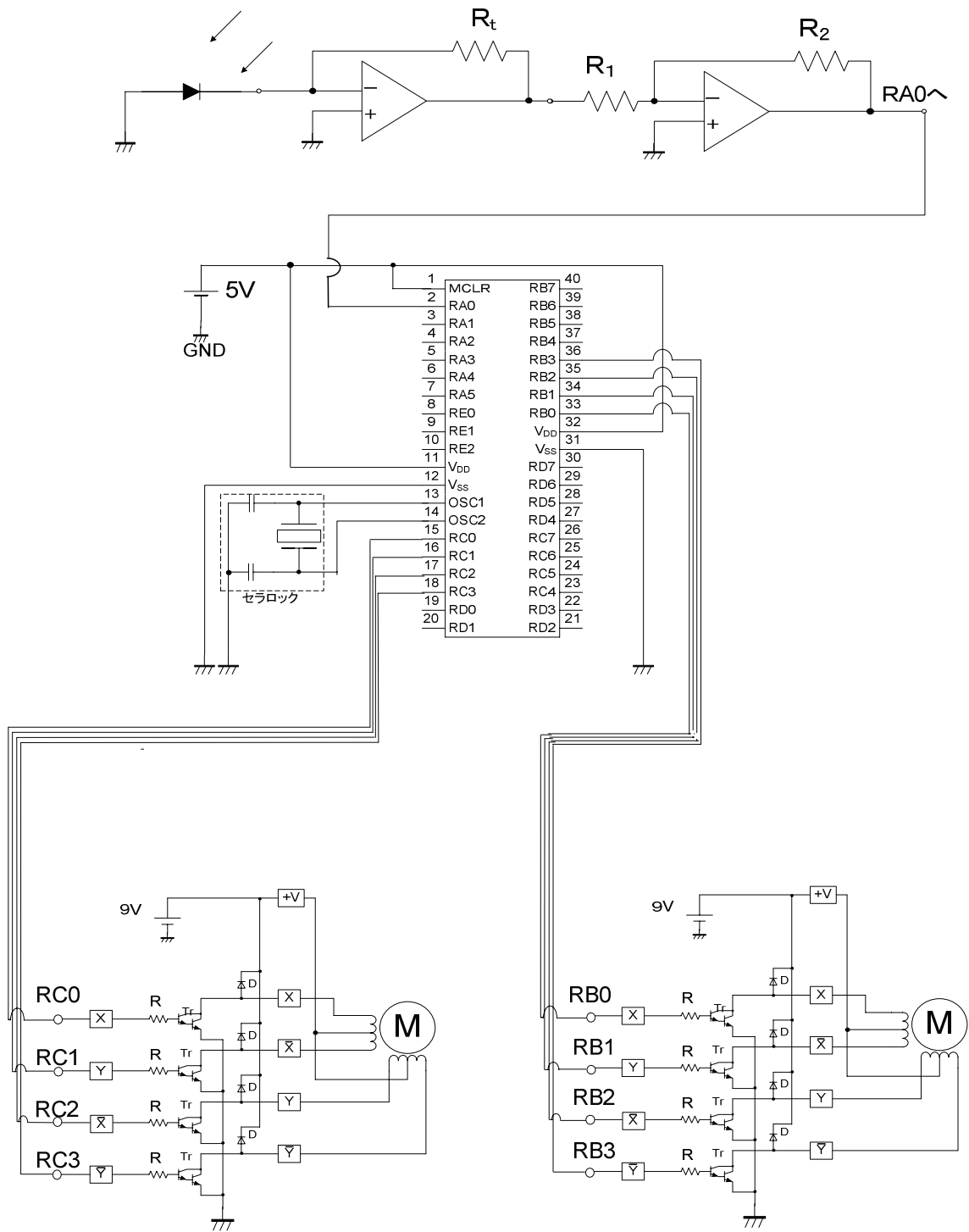


図 3 - 4 1 素子を使用した場合の回路図

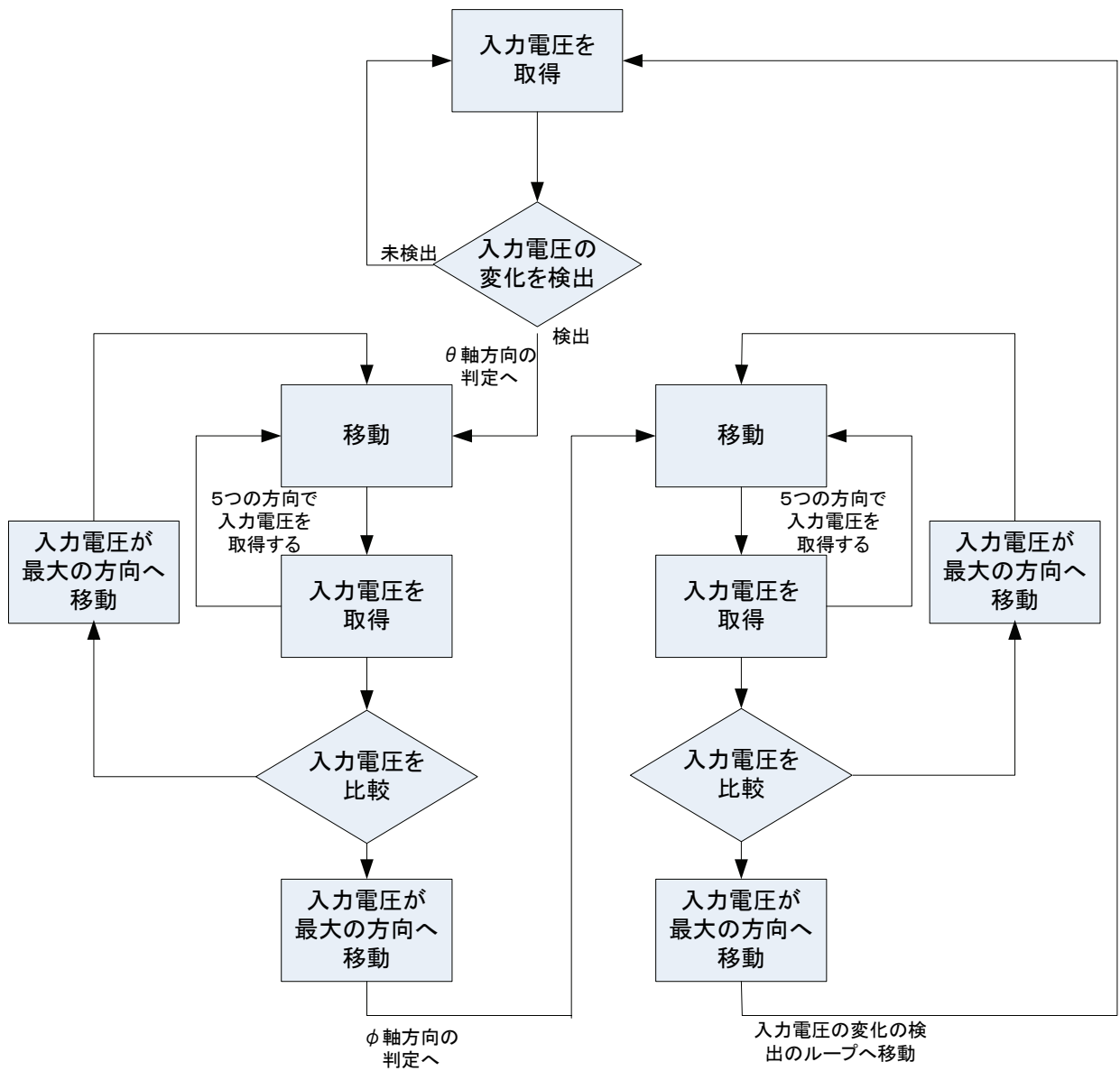


図 3-5 プログラムの簡単なフローチャート

3. 2 4 素子を使用した場合

付録 1 2 軸システムの設計図 の parts 1 にフォトダイオードを設置した際の図を図 3-6 に示す。M は、 ϕ 軸方向を制御するステッピングモーターで、素子番号 1・2 は θ 軸、3・4 は ϕ 軸に対応させる。

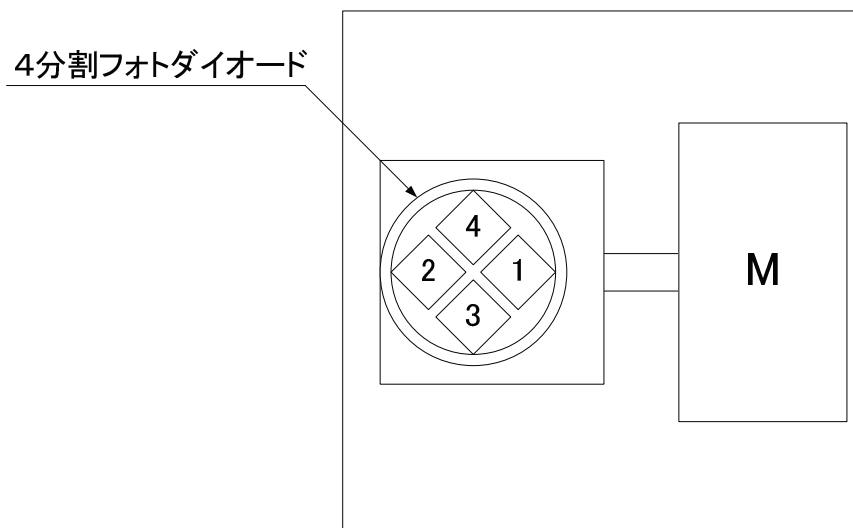


図 3-6 フォトダイオードの配置について

図 3-7 (a) のように、光源が ϕ 軸方向で移動したとする。光源からフォトダイオードまでの距離が近ければ、生じる光起電流も大きくなるため入力電圧が大きくなるため、その素子の方向へステッピングモーターを回転させればよいことになる。図 3-7 (a) の場合、3 と 4 の素子から得られた入力電圧を比較すると、4 の素子から得られる入力電圧のほうが大きいので、図 3-7 (b) のように 4 の素子のある方向にステッピングモーターを回転させる。

この動作を繰り返し行い、素子番号 3 と 4 で同じ入力電圧を得られることができた位置が、光源に向いている方向となる。

しかし、図 3-7 (c) のように、光源がステッピングモーターのステップ角の中間にあった場合、4 の素子の方向に動いた後は 3 の素子の入力電圧の方が大きく、3 の素子の方向に動いた後は 4 の素子の入力電圧の方が大きいという場合がある。その場合、両素子の入力電圧は同じにはならず、往復運動を繰り返してしまうことになり、光源が静止しているにもかかわらず、フォトダイオードが静止しない。このような場合、条件を、両素子から得られる入力電圧の差が許容範囲の値以内とすることで、往復運動を抑制することができる。許容範囲の幅が小さすぎると往復運動を抑制できず、また大きすぎると、両素子から得られる入力電圧の差が大きくなければ移動しないので、素早い追従が行えない。また、光源の種類によって、発生する光起電流が異なるので、適切な許容範囲は異なる。そのため、使用する条件のもとで、適切な許容範囲を設定しなければならない。

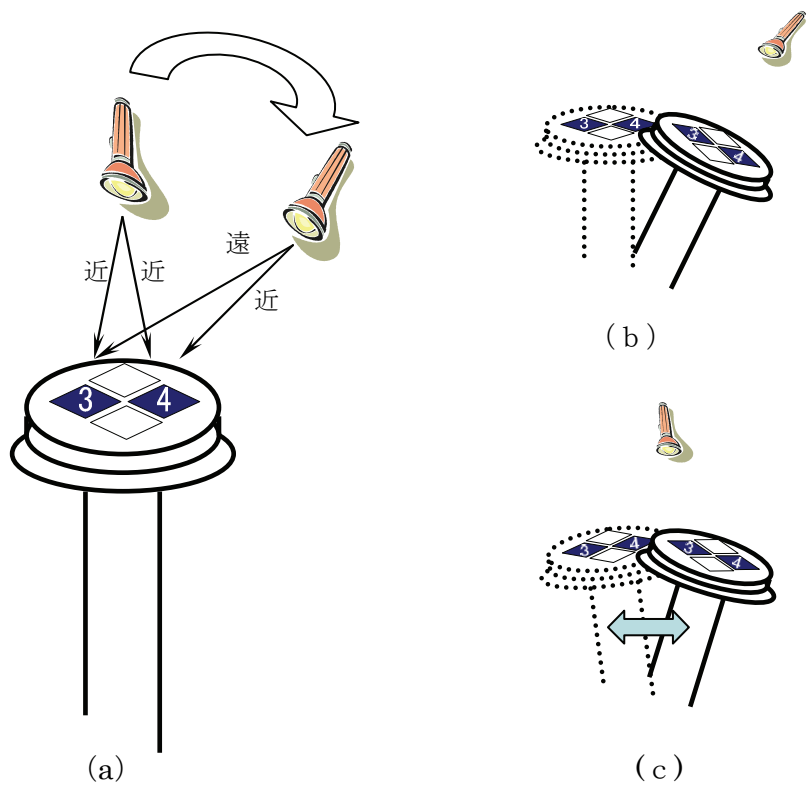


図 3-7 4素子を利用した場合の動作説明図

使用した回路図を図 3-8 に、プログラムの簡単なフローチャートを図 3-9 に、プログラムをプログラムリスト 5 に、プログラムのラベルについてのフローチャートとラベルの説明を付録 3 に示す。回路図のフォトダイオード付近に添付してある 1・2・3・4 は、図 3-6 の 1・2・3・4 に対応する。

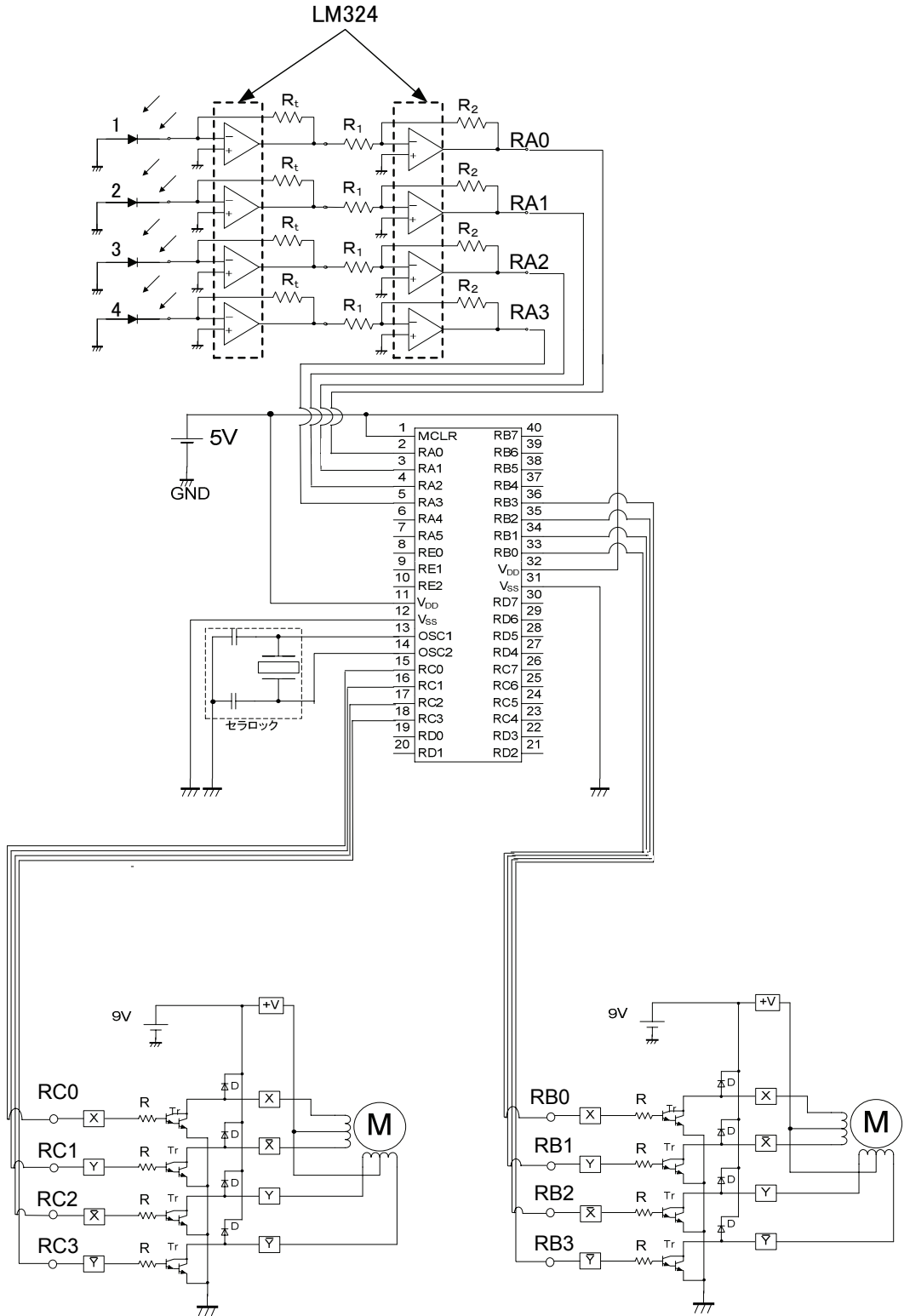


図 3 - 8 4 素子を利用した場合の回路図

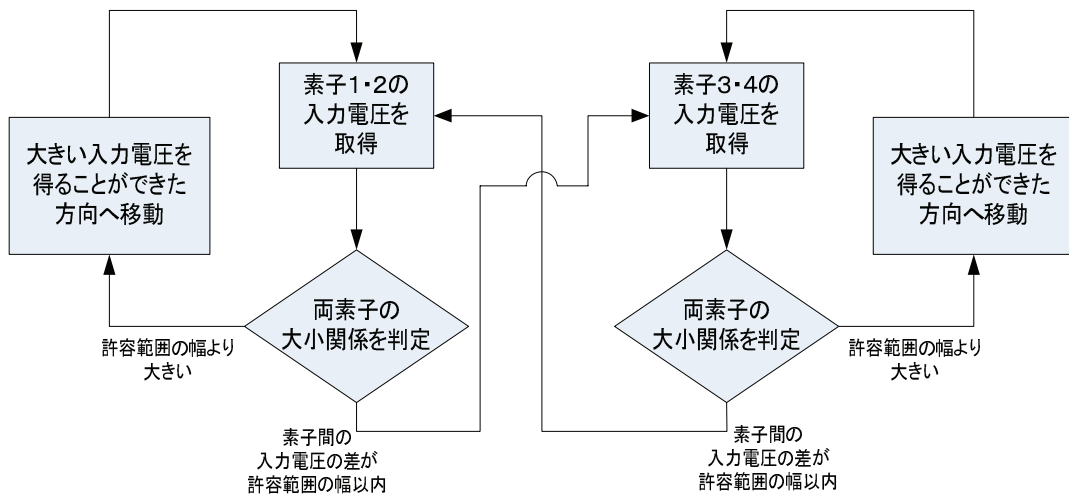


図 3-9 4素子を利用した場合のプログラムの簡単フローチャート

3.3 許容範囲

作成したプログラムで設定する許容範囲の幅は、使用する条件によって適切に選ばなければならない。

実験では、許容範囲の幅に対する静止状態の安定度を測定した。実験内容は、図 3-10 のように光源を静止させ、フォトダイオードが静止したと判断してから一定時間の間にフォトダイオードが移動した回数を 5 回計測し、平均をとる。光源には MINI MAGLITE を使用した。光を照射した際に生じた光起電流は約 $15 \mu\text{A}$ であったので、 $R_t = 270 \text{ k}\Omega$ 、 $R_1 = R_2 = 20 \text{ k}\Omega$ の抵抗を使用した。結果を図 3-11 に示す。

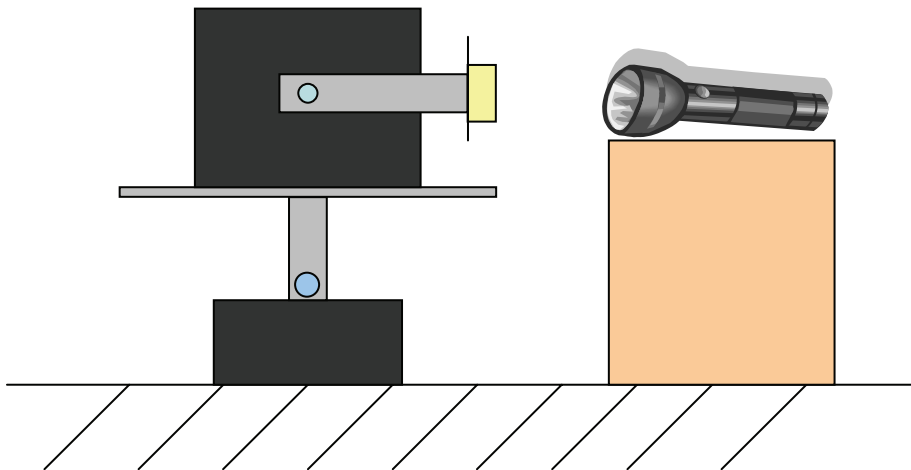


図 3-10 実験の様子

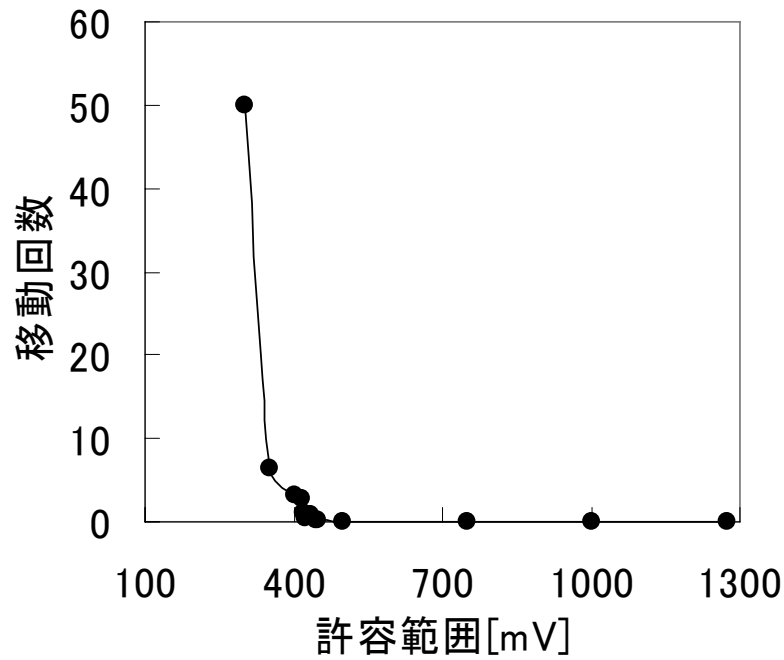


図3-11 設定した許容範囲の値に対する検出部の移動回数

図3-11から、許容範囲をある値より小さく設定すると、急激に移動回数が増加し始めることがわかる。増加し始める直前の許容範囲に設定したとき、静止状態の安定度と追従の速さとのバランスが良いということが実験から分かった。

光源の種類や光源からフォトダイオードまでの距離を変えてもグラフの形状は図3-11と同様であるが、移動回数が増加し始める許容範囲の値は条件によって変化するので、使用する条件下において実験を行う必要がある。

4. まとめ・考察・課題

サントラッカーの基礎的なプログラムを作成した。1素子でのプログラムは初期状態で、光源の位置を把握していない場合に位置検出ができ、4素子でプログラムは、光源の位置を把握した後の位置変化に対応させることができる。

1素子でのプログラムは、入力電圧を取得する方向を増やすことで、初期状態での位置検出のみに特化させることが可能である。そして、位置を特定した後に4素子でのプログラムに切り替わるようにすれば、太陽を追従することが可能である。

ただし太陽を追従する場合には、フォトダイオードにアイリス、筒、レンズ、フィルタを取り付ける必要がある。レンズを径 25mm、焦点距離 50mm とし、像の直径を 1mm とする場合、径 1.923mm のアイリスをフォトダイオードから 2 mm 離れた位置に置く。また使用するフィルタを透過率 0.5、中心波長 500nm、半値全幅 10nm とし、それらを取り付けた場合に生じる光起電流は、計算によって求めることができる。500nm のときの地上における分光放射照度は 1.6 W/m²nm、フォトダイオードの感度は 0.28A/W、入射する光の面積は 0.785mm² より、フォトダイオードで生じる光起電流は 1.8 μA となる。ワンチップマイコンの最大入力電圧が 5V であるので、変換率×増幅率を約 28・10⁵ にすればよいことになる。そして許容範囲を設定しなおすことで追従が可能になると考える。

本研究で製作した 2 軸システムは、実際に計測装置を搭載するシステムにはふさわしくないの
で、実際にリモートセンシングを行うためには、重量があり計測装置が搭載可能な 2 軸システム
の作成が必要である。

5. プログラムリスト

5. 1 プログラムリスト 1

```
LIST      P=PIC16F874
INCLUDE "P16F874.INC"
__CONFIG0X3D3A

CNT1      EQU      020H
CNT2      EQU      021H
HL1       EQU      022H
HL2       EQU      023H
HL3       EQU      024H
STOP      EQU      025H

                ORG      0

MAIN; _____ MAIN
    BSF        STATUS, RP0    ;バンク 1 を選択
    CLRF       TRISB          ;ポート B はすべてデジタル出力
    BCF        STATUS, RP0    ;バンク 0 を選択

    MOVLW     0x01
    MOVWF     STOP            ;STOP に 1 を代入

CALLKAI
    CALL      SEITEN          ;SEITEN を呼び出す
    GOTO     CALLKAI         ;CALLKAI へ移動

KAITEN; _____ 回転
:PORTB0=X
:PORTB1=Y
:PORTB2=X バー
:PORTB3=Y バー

SEITEN ; _____ 正転

    BTFSC     STOP,0          ;STOP が 1 なら
    GOTO     KAITEN2         ;KAITEN2 へ移動
    BTFSC     STOP,1          ;STOP が 2 なら
    GOTO     KAITEN3         ;KAITEN3 へ移動
    BTFSC     STOP,2          ;STOP が 4 なら
    GOTO     KAITEN4         ;KAITEN4 へ移動
    BTFSC     STOP,3          ;STOP が 8 なら
    GOTO     KAITEN1         ;KAITEN1 へ移動

GYAKUTEN ; _____ 逆転

    BTFSC     STOP,0          ;STOP が 1 なら
    GOTO     KAITEN4         ;KAITEN4 へ移動
    BTFSC     STOP,1          ;STOP が 2 なら
    GOTO     KAITEN1         ;KAITEN1 へ移動
    BTFSC     STOP,2          ;STOP が 4 なら
    GOTO     KAITEN2         ;KAITEN2 へ移動
    BTFSC     STOP,3          ;STOP が 8 なら
    GOTO     KAITEN3         ;KAITEN3 へ移動

: _____
KAITEN1
    BSF        PORTB,0        ;PORTB0 を 1 に設定
    BSF        PORTB,3        ;PORTB3 を 1 に設定
    CALL      HABA2           ;HABA2 を呼び出す
    CLRF       PORTB          ;PORTB をクリア
    CALL      HABA1           ;HABA1 を呼び出す
    MOVLW     0x01
    MOVWF     STOP            ;KAITEN1 が終わった後の STOP は 1
    RETURN

KAITEN2
```

BSF	PORTB,0	:PORTB0 を 1 に設定
BSF	PORTB,1	:PORTB1 を 1 に設定
CALL	HABA2	:HABA2 を呼び出す
CLRF	PORTB	:PORTB をクリア
CALL	HABA1	:HABA1 を呼び出す
MOVLW	0x02	
MOVWF	STOP	:KAITEN2 が終わった後の STOP は 2
RETURN		

KAITEN3		
BSF	PORTB,1	:PORTB1 を 1 に設定
BSF	PORTB,2	:PORTB2 を 1 に設定
CALL	HABA2	:HABA2 を呼び出す
CLRF	PORTB	:PORTB をクリア
CALL	HABA1	:HABA1 を呼び出す
MOVLW	0x04	
MOVWF	STOP	:KAITEN3 が終わった後の STOP は 4
RETURN		

KAITEN4		
BSF	PORTB,2	:PORTB2 を 1 に設定
BSF	PORTB,3	:PORTB3 を 1 に設定
CALL	HABA2	:HABA2 を呼び出す
CLRF	PORTB	:PORTB をクリア
CALL	HABA1	:HABA1 を呼び出す
MOVLW	0x08	
MOVWF	STOP	:KAITEN4 が終わった後の STOP は 8
RETURN		

;

;

;	パルス幅	
HABA1;	0. 1 ミリ S のループ	
MOVLW	0x3E	
MOVWF	HL1	

HABAL1		
NOP		
DECFSZ	HL1,F	
GOTO	HABAL1	
RETURN		

HABA2;	10 ミリ S のループ	
MOVLW	0x64	
MOVWF	HL2	

HABAL2		
NOP		
CALL	HABA1	
DECFSZ	HL2,F	
GOTO	HABAL2	
RETURN		

HABA3;	0. 5 秒のループ	
MOVLW	0x32	
MOVWF	HL3	

HABAL3		
NOP		
CALL	HABA2	
DECFSZ	HL3,F	
GOTO	HABAL3	
RETURN		

;

;

END

5. 2 プログラムリスト2

```
LIST      P=PIC16F874
INCLUDE "P16F874.INC"
__CONFIG0X3D3A
```

```
CNT1      EQU      020H
CNT2      EQU      021H
HL1       EQU      022H
HL2       EQU      023H
HL3       EQU      024H
KIOKUH    EQU      025H      ;KIOKU HIGH
KIOKUL    EQU      026H      ;KIOKU LOW
KEIH      EQU      027H      ;KEI HIGH
KEIL      EQU      028H      ;KEI LOW
KAISUU    EQU      029H
STOP      EQU      030H

          ORG      0
```

```
MAIN:_____
      BSF      STATUS, RP0      ;バンク 1 を選択

      CLRF     TRISB            ;PORTB はすべてデジタル出力
      MOVLW   0X80
      MOVWF   ADCON1          ;ADCON1 に 10000000 を代入

      BCF      STATUS, RP0      ;バンク 0 を選択

      CLRF     PORTB           ;PORTB をクリア
      MOVLW   0x01
      MOVWF   STOP            ;STOP に 1 を代入

      MOVLW   0X41
      MOVWF   ADCON0          ;ADCON0 に 01000001 を代入
      CALL    WAIT            ;蓄積時間の呼び出し
      BSF     ADCON0, GO       ;ADCON0 GO を 1 にして A/D 変換スタート
      CALL    TIMER1          ;A/D 変換終了!!!!

      BSF     STATUS, RP0      ;バンク 1 を選択
      MOVF    ADRESL, W        ;ADRESL を W レジスタへ
      BCF     STATUS, RP0      ;バンク 0 を選択
      MOVWF   KEIL            ;ADRESL を KEIL に代入

      MOVF    ADRESH, W
      MOVWF   KEIH            ;ADRESH を KEIH に代入

      GOTO    MAIN3           ;MAIN3 へ
```

```
_____;
```

```
MAIN1:_____メインプログラム(入力電圧が大きくなったとき)
```

```
      MOVLW   0x04
      MOVWF   KAISUU          ;KAISUU を 4 に設定
KLOOP1
      CALL    SEITEN          ;SEITEN を呼び出し
      DECFSZ KAISUU, F;KAISUU が 0 になるまで回転
      GOTO    KLOOP1          ;0 になると回転終了

      MOVF    KEIL, W         ;KEI を KIOKU に代入
      MOVWF   KIOKUL
      MOVF    KEIH, W
      MOVWF   KIOKUH

      MOVLW   0X41            ;A/D 変換
      MOVWF   ADCON0
      CALL    WAIT
      BSF     ADCON0, GO
      CALL    TIMER1          ;A/D 変換終了

      BSF     STATUS, RP0      ;ADRES を KEI に代入
```

```

MOVF      ADRESL,W
BCF       STATUS, RP0
MOVWF    KEIL

MOVF      ADRESH,W
MOVWF    KEIH

GOTO     HANTEI1      ;HANTEI1 へ

MAIN2;_____メインプログラム(入力電圧が小さくなったとき)

MOVLW    0x04
MOVWF    KAISUU      ;KAISUU を 4 に設定

KLOOP2
CALL     GYAKUTEN    ;GYAKUTEN を呼び出し
DECFSZ   KAISUU,F;KAISUU が 0 になるまで回転
GOTO     KLOOP2      ;0 になると回転終了

MOVF     KEIL,W      ;KEI を KIOKU に代入
MOVWF    KIOKUL
MOVF     KEIH,W
MOVWF    KIOKUH

MOVLW    0X41        ;A/D 変換
MOVWF    ADCON0
CALL     WAIT
BSF     ADCON0, GO
CALL     TIMER1      ;A/D 変換終了

BSF     STATUS, RP0 ;ADRES を KEI に代入
MOVF    ADRESL,W
BCF     STATUS, RP0
MOVWF   KEIL

MOVF    ADRESH,W
MOVWF   KEIH

GOTO   HANTEI1      ;HANTEI1 へ

MAIN3 ;_____ (入力電圧が同じであった場合)
MOVF   KEIL,W      ;KEI を KIOKU に代入
MOVWF  KIOKUL
MOVF   KEIH,W
MOVWF  KIOKUH

MOVLW  0X41        ;A/D 変換
MOVWF  ADCON0
CALL   WAIT
BSF   ADCON0, GO
CALL  TIMER1      ;A/D 変換終了

BSF   STATUS, RP0 ;ADRES を KEI に代入
MOVF  ADRESL,W
BCF   STATUS, RP0
MOVWF KEIL
MOVF  ADRESH,W
MOVWF KEIH

GOTO  HANTEI1      ;HANTEI1 へ

;_____

WAIT;_____蓄積時間

MOVLW  0FH
MOVWF  CNT1

LOOP1
NOP
NOP
NOP
NOP
NOP

```

```

NOP
NOP
NOP
CALL          SUB2
DECFSZ  CNT1,F
GOTO      LOOP1
RETURN
SUB2
MOVLW          03FH
MOVWF         CNT2
LOOP2
NOP
NOP
NOP
NOP
NOP
NOP
DECFSZ  CNT2,F
GOTO      LOOP2
RETURN
;
;
;
TIMER1
BTFSZ     ADCON0, GO      ;A/D 変換が終了するのを待つループ
GOTO      TIMER1        ;ADCON GO が 0 なら A/D 変換終了
RETURN
;
;
;
HABA1;      パルス幅
           0. 1 ミリ S のループ
MOVLW     0x34
MOVWF     HL1
HABAL1
NOP
DECFSZ   HL1,F
GOTO     HABAL1
RETURN
HABA2;      1 0 ミリ S のループ
MOVLW     0x64
MOVWF     HL2
HABAL2
NOP
CALL     HABAL1
DECFSZ   HL2,F
GOTO     HABAL2
RETURN
HABA3;      0. 5 秒のループ
MOVLW     0x32
MOVWF     HL3
HABAL3
NOP
CALL     HABAL2
DECFSZ   HL3,F
GOTO     HABAL3
RETURN
;
;
HANTEI;
;
HANTEI1
MOVF     KEIH,W          ;KIOKUH が小さければ MAIN1 へ
SUBWF    KIOKUH,W       ;KIOKUH - KEIH
BTFSZ    STATUS,C
GOTO     HANTEI2        ;KEIH <= KIOKUH → HANTEI2 へ
GOTO     MAIN1          ;KEIH > KIOKUH → MAIN1 へ
;
;
HANTEI2
           ;KEIH が小さければ MAIN2 へ
           ;そうでなければ HIGH は同じなので ADRESL の判定へ
MOVF     KIOKUH,W
SUBWF    KEIH,W          ;KEIH - KIOKUH

```

```

        BTFSC      STATUS,C
        GOTO      HANTEI3
        GOTO      MAIN2
;KEIH = KIOKUH → HANTEI3 へ
;KEIH < KIOKUH → MAIN2 へ
;
;-----
HANTEI3
        MOVF      KEIL,W
        SUBWF     KIOKUL,W
        BTFSC     STATUS,C
        GOTO      HANTEI4
        GOTO      MAIN1
;KIOKUL が小さければ MAIN1 へ
;KIOKUL-KEIL
;KEIL <= KIOKUL → HANTEI4 へ
;KEIL > KIOKUL → MAIN1 へ
;
;-----
HANTEI4
        MOVF      KIOKUL,W
        SUBWF     KEIL,W
        BTFSC     STATUS,C
        GOTO      MAIN3
        GOTO      MAIN2
;KEIL が小さければ MAIN2 へ
;そうでなければ同じ
;KEIL-KIOKUL
;KEIL = KIOKUL → MAIN3 へ
;KEIL < KIOKUL → MAIN2 へ
;
;-----

```

KAITEN; _____ 回転

```

;PORTB0=X
;PORTB1=Y
;PORTB2=X バー
;PORTB3=Y バー
SEITEN ; _____ 正転

```

```

        BTFSC     STOP,0
        GOTO     KAITEN2
        BTFSC     STOP,1
        GOTO     KAITEN3
        BTFSC     STOP,2
        GOTO     KAITEN4
        BTFSC     STOP,3
        GOTO     KAITEN1
;STOP が 1 なら
;KAITEN2 へ移動
;STOP が 2 なら
;KAITEN3 へ移動
;STOP が 4 なら
;KAITEN4 へ移動
;STOP が 8 なら
;KAITEN1 へ移動

```

GYAKUTEN ; _____ 逆転

```

        BTFSC     STOP,0
        GOTO     KAITEN4
        BTFSC     STOP,1
        GOTO     KAITEN1
        BTFSC     STOP,2
        GOTO     KAITEN2
        BTFSC     STOP,3
        GOTO     KAITEN3
;STOP が 1 なら
;KAITEN4 へ移動
;STOP が 2 なら
;KAITEN1 へ移動
;STOP が 4 なら
;KAITEN2 へ移動
;STOP が 8 なら
;KAITEN3 へ移動

```

;-----
KAITEN1

```

        BSF      PORTB,0
        BSF      PORTB,3
        CALL     HABA2
        CLRF     PORTB
        CALL     HABA1
        MOVLW    0x01
        MOVWF    STOP
        RETURN
;PORTB0 を 1 に設定
;PORTB3 を 1 に設定
;HABA2 を呼び出す
;PORTB をクリア
;HABA1 を呼び出す
;KAITEN1 が終わった後の STOP は 1

```

KAITEN2

```

        BSF      PORTB,0
        BSF      PORTB,1
        CALL     HABA2
        CLRF     PORTB
        CALL     HABA1
        MOVLW    0x02
        MOVWF    STOP
        RETURN
;PORTB0 を 1 に設定
;PORTB1 を 1 に設定
;HABA2 を呼び出す
;PORTB をクリア
;HABA1 を呼び出す
;KAITEN2 が終わった後の STOP は 2

```

KAITEN3

```

        BSF      PORTB,1
;PORTB1 を 1 に設定

```

BSF	PORTB,2	;PORTB2を 1 に設定
CALL	HABA2	;HABA2を呼び出す
CLRF	PORTB	;PORTBをクリア
CALL	HABA1	;HABA1を呼び出す
MOVLW	0x04	
MOVWF	STOP	;KAITEN3が終わった後のSTOPは 4
RETURN		

KAITEN4

BSF	PORTB,2	;PORTB2を 1 に設定
BSF	PORTB,3	;PORTB3を 1 に設定
CALL	HABA2	;HABA2を呼び出す
CLRF	PORTB	;PORTBをクリア
CALL	HABA1	;HABA1を呼び出す
MOVLW	0x08	
MOVWF	STOP	;KAITEN4が終わった後のSTOPは 8
RETURN		

;

END

5. 3 プログラムリスト3

```

LIST      P=PIC16F874
INCLUDE "P16F874.INC"
__CONFIG0X3D3A
CNT1     EQU      020H
CNT2     EQU      021H

HL1      EQU      022H
HL2      EQU      023H
HL3      EQU      024H

KIOKUH   EQU      025H
KIOKUL   EQU      026H

KEIH     EQU      027H
KEIL     EQU      028H

STOP     EQU      29H
KITEI    EQU      30H

STOPH1   EQU      31H
STOPL1   EQU      32H

STOPH2   EQU      33H
STOPL2   EQU      34H

ADH1     EQU      35H
ADL1     EQU      36H

ADH2     EQU      37H
ADL2     EQU      38H

MAIN     ORG      0
          ; _____ 初期設定
          BSF      STATUS, RP0
          CLRF     TRISB
          MOVLW    0X80
          MOVWF   ADCON1
          BCF      STATUS, RP0
          CALL    HABA3
          CLRF     PORTB
          MOVLW    0x05
          MOVWF   KITEI      ;KITEI は 5
          MOVLW    0x01
          MOVWF   STOP       ;STOP に 1 を代入

          MOVLW    0x00
          MOVWF   KIOKUL     ;初期状態の KIOKU は 0
          MOVF     0x00
          MOVWF   KIOKUH

          MOVLW    0x00
          MOVWF   STOPH1     ;初期状態の STOP は 0
          MOVLW    0x00
          MOVWF   STOPL1
          MOVLW    0x00
          MOVWF   STOPH2
          MOVLW    0x00
          MOVWF   STOPL2

          GOTO    MAIN3      ;MAIN3 へ

MAIN1    ; _____ メインプログラム(大きくなったとき)
          MOVF     KEIH,W     ;KEI を AD 1 に代入
          MOVWF   ADH1
          MOVF     KEIL,W
          MOVWF   ADL1

          MOVF     KIOKUH,W   ;KIOKU を STOP 1 に代入
          MOVWF   STOPH1
          MOVF     KIOKUL,W

```


X1	MOVWF	STOPL1	
	MOVF	STOPL1,W;STOP 1 を KIOKU に代入	
	MOVWF	KIOKUL	
	MOVF	STOPH1,W	
	MOVWF	KIOKUH	
	MOVF	KITEI,W	;STOP+KITEI=STOP
	ADDWF	STOPL1,F	
	BTFSC	STATUS,C	
	INCF	STOPH1, F	
	MOVF	ADH1,W	
	SUBWF	STOPH1,W	
	BTFSS	STATUS,C	
	GOTO	XX1	;STOPH-ADH < 0 ;STOPH-ADH ≥ 0
	MOVF	STOPH1,W	
	SUBWF	ADH1,W	;ADH-STOPH の結果を判定
	BTFSS	STATUS,C	
	GOTO	ADHEN1	;STOPH-ADH > 0 ;STOPH-ADH = 0
	MOVF	STOPL1,W	
	SUBWF	ADL1,W	;STOPL-ADL の結果を判定
	BTFSC	STATUS,C	
	GOTO	XX1	;ADL-STOPL ≥ 0
	GOTO	ADHEN1	;ADL-STOPL < 0
XX1	CALL	SEITEN	;SEITEN を呼び出し
	GOTO	X1	;X1 へ
ADHEN1	MOVLW	0X41	;A/D 変換
	MOVWF	ADCON0	
	CALL	WAIT	
	BSF	ADCON0, GO	
	CALL	TIMER1	;A/D 変換終了
	BSF	STATUS, RP0	;ADRES を KEI に代入
	MOVF	ADRESL,W	
	BCF	STATUS, RP0	
	MOVWF	KEIL	
	MOVF	ADRESH,W	
	MOVWF	KEIH	
	GOTO	HANTEI1	;HANTEI1 へ
MAIN2		; _____ メインプログラム(小さくなったとき)	
	MOVF	KEIH,W	;KEI を AD 2 に代入
	MOVWF	ADH2	
	MOVF	KEIL,W	
	MOVWF	ADL2	
	MOVF	KIOKUH,W	;KIOKU を STOP 2 に代入
	MOVWF	STOPH2	
	MOVF	KIOKUL,W	
	MOVWF	STOPL2	
X2	CALL	GYAKUTEN	;GYAKUTEN を呼び出し
	MOVF	KITEI,W	;STOP-KITEI=STOP
	SUBWF	STOPL2,F	
	BTFSS	STATUS,C	
	DECf	STOPH2,F	

	MOVF	STOPH2,W	
	SUBWF	ADH2,W	;ADH-STOPH の結果を判定
	BTFSS	STATUS,C	
	GOTO	X2	;ADH-STOPH < 0
			;ADH-STOPH ≥ 0
	MOVF	ADH2,W	
	SUBWF	STOPH2,W	;STOPH-ADH の結果を判定
	BTFSS	STATUS,C	
	GOTO	ADHEN2	;ADH-STOPH > 0
			;ADH-STOPH = 0
	MOVF	STOPL2,W	
	SUBWF	ADL2	;STOPL-ADL の結果を判定
	BTFSS	STATUS,C	
	GOTO	X2	;STOPL-ADL ≤ 0
	GOTO	ADHEN2	;STOPL-ADL > 0
ADHEN2			
	MOVF	STOPL2,W;STOP 2 を KIOKU に代入	
	MOVWF	KIOKUL	
	MOVF	STOPH2,W	
	MOVWF	KIOKUH	
	MOVLW	0X41	;A/D 変換
	MOVWF	ADCON0	
	CALL	WAIT	
	BSF	ADCON0, GO	
	CALL	TIMER1	;A/D 変換終了
	BSF	STATUS, RP0	;ADRES を KEI に代入
	MOVF	ADRESL,W	
	BCF	STATUS, RP0	
	MOVWF	KEIL	
	MOVF	ADRESH,W	
	MOVWF	KEIH	
	GOTO	HANTEI1	;HANTEI1 へ
MAIN3;			同じ場合の MAIN
	MOVLW	0X41	;A/D 変換
	MOVWF	ADCON0	
	CALL	WAIT	
	BSF	ADCON0, GO	
	CALL	TIMER1	;A/D 変換終了
	BSF	STATUS, RP0	;ADRES を KEI に代入
	MOVF	ADRESL,W	
	BCF	STATUS, RP0	
	MOVWF	KEIL	
	MOVF	ADRESH,W	
	MOVWF	KEIH	
	GOTO	HANTEI1	;HANTEI1 へ
;			
WAIT;			蓄積時間
	MOVLW	0FH	
	MOVWF	CNT1	
LOOP1			
	NOP		
	NOP		
	NOP		
	NOP		
	NOP		
	NOP		
	NOP		
	NOP		

```

CALL          SUB2
DECFSZ      CNT1,F
GOTO        LOOP1
RETURN

SUB2
MOVLW      03FH
MOVWF      CNT2

LOOP2
NOP
NOP
NOP
NOP
NOP
NOP
DECFSZ     CNT2,F
GOTO        LOOP2
RETURN

;
;
;
TIMER1
BTFSZ      ADCON0, GO      ;A/D 変換が終了するのを待つループ
GOTO        TIMER1        ;ADCON GO が 0 なら A/D 変換終了
RETURN

;
;
KAITEN;
;PORTB0=X
;PORTB1=Y
;PORTB2=X バー
;PORTB3=Y バー
SEITEN ;
;
BTFSZ      STOP,0          ;STOP が 1 なら
GOTO        KAITEN2        ;KAITEN2 へ移動
BTFSZ      STOP,1          ;STOP が 2 なら
GOTO        KAITEN3        ;KAITEN3 へ移動
BTFSZ      STOP,2          ;STOP が 4 なら
GOTO        KAITEN4        ;KAITEN4 へ移動
BTFSZ      STOP,3          ;STOP が 8 なら
GOTO        KAITEN1        ;KAITEN1 へ移動

GYAKUTEN ;
;
BTFSZ      STOP,0          ;STOP が 1 なら
GOTO        KAITEN4        ;KAITEN4 へ移動
BTFSZ      STOP,1          ;STOP が 2 なら
GOTO        KAITEN1        ;KAITEN1 へ移動
BTFSZ      STOP,2          ;STOP が 4 なら
GOTO        KAITEN2        ;KAITEN2 へ移動
BTFSZ      STOP,3          ;STOP が 8 なら
GOTO        KAITEN3        ;KAITEN3 へ移動

;
KAITEN1
BSF        PORTB,0          ;PORTB0 を 1 に設定
BSF        PORTB,3          ;PORTB3 を 1 に設定
CALL       HABA2            ;HABA2 を呼び出す
CLRF      PORTB            ;PORTB をクリア
CALL       HABA1            ;HABA1 を呼び出す
MOVLW     0x01
MOVWF     STOP              ;KAITEN1 が終わった後の STOP は 1
RETURN

KAITEN2
BSF        PORTB,0          ;PORTB0 を 1 に設定
BSF        PORTB,1          ;PORTB1 を 1 に設定
CALL       HABA2            ;HABA2 を呼び出す
CLRF      PORTB            ;PORTB をクリア
CALL       HABA1            ;HABA1 を呼び出す
MOVLW     0x02
MOVWF     STOP              ;KAITEN2 が終わった後の STOP は 2

```

```

RETURN

KAITEN3
BSF          PORTB,1      ;PORTB1 を 1 に設定
BSF          PORTB,2      ;PORTB2 を 1 に設定
CALL        HABA2        ;HABA2 を呼び出す
CLRF        PORTB        ;PORTB をクリア
CALL        HABA1        ;HABA1 を呼び出す
MOVLW      0x04
MOVWF      STOP          ;KAITEN3 が終わった後の STOP は 4
RETURN

KAITEN4
BSF          PORTB,2      ;PORTB2 を 1 に設定
BSF          PORTB,3      ;PORTB3 を 1 に設定
CALL        HABA2        ;HABA2 を呼び出す
CLRF        PORTB        ;PORTB をクリア
CALL        HABA1        ;HABA1 を呼び出す
MOVLW      0x08
MOVWF      STOP          ;KAITEN4 が終わった後の STOP は 8
RETURN
;
;
;
; _____ パルス幅
HABA1; _____ 0. 1 ミリ S のループ
    MOVLW      0x3E
    MOVWF      HL1
HABAL1
    NOP
    DECFSZ    HL1,F
    GOTO      HABA1
    RETURN

HABA2; _____ 1 0 ミリ S のループ
    MOVLW      0x64
    MOVWF      HL2
HABAL2
    NOP
    CALL      HABA1
    DECFSZ    HL2,F
    GOTO      HABAL2
    RETURN

HABA3; _____ 0. 5 秒のループ
    MOVLW      0x32
    MOVWF      HL3
HABAL3
    NOP
    CALL      HABA2
    DECFSZ    HL3,F
    GOTO      HABAL3
    RETURN
;
; _____
HANTEI; _____
;;
HANTEI1
    MOVF      KEIH,W      ;KIOKUH が小さければ MAIN1 へ
    SUBWF    KIOKUH,W     ;KIOKUH - KEIH
    BTFSC    STATUS,C
    GOTO     HANTEI2     ;KEIH <= KIOKUH → HANTEI2 へ
    GOTO     MAIN1       ;KEIH > KIOKUH → MAIN1 へ
;
; _____
HANTEI2
    MOVF      KIOKUH,W    ;KEIH が小さければ MAIN2 へ
    SUBWF    KEIH,W      ;そうでなければ HIGH は同じなので ADRESL の判定へ
    BTFSC    STATUS,C
    GOTO     HANTEI3     ;KEIH = KIOKUH → HANTEI3 へ
    GOTO     MAIN2       ;KEIH < KIOKUH → MAIN2 へ

```

```

;-----
HANTEI3                                ;KIOKUL が小さければ MAIN1 へ
      MOVF          KEIL,W
      SUBWF        KIOKUL,W           ;KIOKUL-KEIL
      BTFSC       STATUS,C
      GOTO        HANTEI4           ;KEIL <= KIOKUL  →  HANTEI4 へ
      GOTO        MAIN1           ;KEIL > KIOKUL  →  MAIN1 へ
;-----
HANTEI4                                ;KEIL が小さければ MAIN2 へ
      MOVF          KIOKUL,W
      SUBWF        KEIL,W           ;KEIL-KIOKUL
      BTFSC       STATUS,C
      GOTO        MAIN3           ;KEIL = KIOKUL  →  MAIN3 へ
      GOTO        MAIN2           ;KEIL < KIOKUL  →  MAIN2 へ
;-----

```

END

5. 4 プログラムリスト4

```

LIST      P=PIC16F874
INCLUDE "P16F874.INC"
__CONFIG0X3D3A
CNT1      EQU      020H
CNT2      EQU      021H

HL1       EQU      022H
HL2       EQU      023H
HL3       EQU      024H

KIOKUH    EQU      025H
KIOKUL    EQU      026H

KEIH      EQU      027H
KEIL      EQU      028H

STOP      EQU      29H
KITEI     EQU      30H

STOPH1    EQU      31H
STOPL1    EQU      32H

STOPH2    EQU      33H
STOPL2    EQU      34H

ADH1      EQU      35H
ADL1      EQU      36H

ADH2      EQU      37H
ADL2      EQU      38H

ADH3      EQU      39H
ADL3      EQU      40H

ADH4      EQU      41H
ADL4      EQU      42H

ADH5      EQU      43H
ADL5      EQU      44H

KAZU      EQU      45H
KAISUU    EQU      46H
KITEI05   EQU      47H
TEMP05    EQU      48H
ASOBI     EQU      49H

TEMP1     EQU      50H
TEMP2     EQU      51H

KLH       EQU      52H
KLL       EQU      53H

KHH       EQU      54H
KHL       EQU      55H

TSTOP     EQU      56H

START     ORG      0

BSF       STATUS, RP0      ;バンク 1 を選択
CLRF      TRISB            ;ポート B はすべてデジタル出力
CLRF      TRISC            ;ポート C はすべてデジタル出力
MOVLW    0X80
MOVWF    ADCON1
BCF      STATUS, RP0      ;バンク 0 を選択

CLRF      PORTB            ;PORTB をクリア
CLRF      PORTC            ;PORTC をクリア

MOVLW    0x01
MOVWF    STOP              ;STOP を 1 に設定

```

MOVLW	0x01	:TSTOP を 1 に設定
MOVWF	TSTOP	
MOVLW	0x01	:KITEI を 1 に設定
MOVWF	KITEI	
MOVLW	0x30	:TEMP1 を 48 に設定
MOVWF	TEMP1	:許容範囲の上限
MOVLW	0x30	:TEMP2 を 48 に設定
MOVWF	TEMP2	:許容範囲の下限
GOTO	MAIN	:MAIN へ

MAINXX: _____ あそびの範囲内か判定 (範囲内のうちはこのループ、範囲外なら MAIN へ)

MOVLW	0X41	:A/D 変換
MOVWF	ADCON0	
CALL	WAIT	
BSF	ADCON0, GO	
CALL	TIMER1	:A/D 変換終了
MOVF	ADRESH,W	:ADRES を KEI に代入
MOVWF	KEIH	
BSF	STATUS, RP0	
MOVF	ADRESL,W	
BCF	STATUS, RP0	
MOVWF	KEIL	

ASOBIH: _____ 上限以上かどうかを判定させる

MOVF	KIOKUL,W	:KIOKU を KH に代入
MOVWF	KHL	
MOVF	KIOKUH,W	
MOVWF	KHH	
MOVF	TEMP1,W	:KH+TEMP1=KH (許容範囲の上限)
ADDWF	KHL,F	
BTFSC	STATUS,C	
INCF	KHH,F	
GOTO	HANTEIX1	:HANTEIX1 へ

ASOBIL: _____ 下限以上かどうかを判定させる

MOVF	KIOKUL,W	:KIOKU を KL に代入
MOVWF	KLL	
MOVF	KIOKUH,W	
MOVWF	KLH	
MOVF	TEMP2,W	:KL-TEMP2=KL (許容範囲の下限)
SUBWF	KLL,F	
BTFSS	STATUS,C	
DECf	KLH,F	
GOTO	HANTEIX2	:HANTEIX2 へ

AS: _____ 許容範囲内

GOTO	MAINXX	:MAINXX へ
------	--------	-----------

;

;

HANTEIX1

:KH の方が大きければ上限以下→下限の判定へ

:KEI の方が大きければ上限以上→MAIN へ

;

HANTEIX11

MOVF	KEIH,W
SUBWF	KHH,W
BTFSC	STATUS,C
GOTO	HANTEIX12
GOTO	MAIN

;

HANTEIX12

MOVF	KHH,W
SUBWF	KEIH,W
BTFSC	STATUS,C

```

        GOTO      HANTEIX13
        GOTO      ASOBIL
;
;-----
HANTEIX13
        MOVF      KEIL,W
        SUBWF     KHL,W
        BTFSC     STATUS,C
        GOTO      HANTEIX14
        GOTO      MAIN
;
;-----
HANTEIX14
        MOVF      KHL,W
        SUBWF     KEIL,W
        BTFSC     STATUS,C
        GOTO      ASOBIL
        GOTO      ASOBIL
;
;-----
HANTEIX2
;KLの方が小さければ下限以上かつ上限以下→ASへ
;KEIの方が小さければ下限以下→MAINへ
;
;-----
HANTEIX21
        MOVF      KEIH,W
        SUBWF     KLH,W
        BTFSC     STATUS,C
        GOTO      HANTEIX22
        GOTO      AS
;
;-----
HANTEIX22
        MOVF      KLH,W
        SUBWF     KEIH,W
        BTFSC     STATUS,C
        GOTO      HANTEIX23
        GOTO      MAIN
;
;-----
HANTEIX23
        MOVF      KEIL,W
        SUBWF     KLL,W
        BTFSC     STATUS,C
        GOTO      HANTEIX24
        GOTO      AS
;
;-----
HANTEIX24
        MOVF      KLL,W
        SUBWF     KEIL,W
        BTFSC     STATUS,C
        GOTO      AS
        GOTO      MAIN
;
;-----

```

;シータ軸

```

MAIN      ; _____メインプログラム_____

```

```

        GOTO      MAIN1
;
;-----

```

```

MAIN1
        CALL      ADHENA      ;ADHENA を呼び出し
        CALL      HANTEI12345 ;HANTEI12345 を呼び出し
        CALL      MAINX1     ;MAINX1 を呼び出し
        GOTO      HANTEI05   ;HANTEI05 へ
;
;-----
;-----

```



```

MAIN2
    CALL      ADHENB      ;ADHENB を呼び出し
    CALL      HANTEI12345 ;HANTEI12345 を呼び出し
    CALL      MAINX2      ;MAINX2 を呼び出し
    MOVF      KAZU,W      ;4-KAZU=KAZU
    SUBLW     0x04
    MOVWF     KAZU
    GOTO      HANTEI05    ;HANTEI05 へ
;
;

```

```

HANTEI05; _____ KAZU の判定
;入力電圧取得前位置との最大入力的位置が同じ (KAZU=2) なら TMAIN へ
;0 1 なら MAIN1 へ
;3 4 なら MAIN2 へ
    MOVF      KAZU,W
    MOVWF     TEMP05
    MOVLW     0x01
    MOVWF     KITEI05
    MOVF      KITEI05
    SUBWF     TEMP05,F
    BTFSS     STATUS,C
    GOTO      MAIN1
    MOVF      KITEI05
    SUBWF     TEMP05,F
    BTFSS     STATUS,C
    GOTO      MAIN1
    MOVF      KITEI05
    SUBWF     TEMP05,F
    BTFSS     STATUS,C
    GOTO      TMAIN
    GOTO      MAIN2
;
;

```

```

MAINX1; _____ 移動
    MOVF      KAZU,W      ;KAISUU に KAZU を代入
    MOVWF     KAISUU
XXX1      ;KAISUU-KITEI=0 になるまで GYAKUTEN を呼び出す
    MOVF      KITEI,W
    SUBWF     KAISUU,F
    BTFSS     STATUS,C
    RETURN
    CALL      GYAKUTEN
    GOTO      XXX1
;
;

```

```

MAINX2; _____ 移動
    MOVF      KAZU,W      ;KAISUU に KAZU を代入
    MOVWF     KAISUU
XXX2      ;KAISUU-KITEI=0 になるま SEITEN を呼び出す
    MOVF      KITEI,W
    SUBWF     KAISUU,F
    BTFSS     STATUS,C
    RETURN
    CALL      SEITEN
    GOTO      XXX2
;
;

```

```

KAITEN; _____ 回転
;PORTB0=X
;PORTB1=Y

```

```

;PORTB2=X バー
;PORTB3=Y バー
SEITEN ; _____ 正転

```

```

    BTFSC      STOP,0      ;STOP が 1 なら
    GOTO       KAITEN2    ;KAITEN2 へ移動
    BTFSC      STOP,1      ;STOP が 2 なら
    GOTO       KAITEN3    ;KAITEN3 へ移動
    BTFSC      STOP,2      ;STOP が 4 なら
    GOTO       KAITEN4    ;KAITEN4 へ移動
    BTFSC      STOP,3      ;STOP が 8 なら
    GOTO       KAITEN1    ;KAITEN1 へ移動

```

```

GYAKUTEN ; _____ 逆転

```

```

    BTFSC      STOP,0      ;STOP が 1 なら
    GOTO       KAITEN4    ;KAITEN4 へ移動
    BTFSC      STOP,1      ;STOP が 2 なら
    GOTO       KAITEN1    ;KAITEN1 へ移動
    BTFSC      STOP,2      ;STOP が 4 なら
    GOTO       KAITEN2    ;KAITEN2 へ移動
    BTFSC      STOP,3      ;STOP が 8 なら
    GOTO       KAITEN3    ;KAITEN3 へ移動

```

```

; _____
KAITEN1

```

```

    BSF        PORTB,0     ;PORTB0 を 1 に設定
    BSF        PORTB,3     ;PORTB3 を 1 に設定
    CALL       HABA2       ;HABA2 を呼び出す
    CLRF       PORTB      ;PROTB をクリア
    CALL       HABA1       ;HABA1 を呼び出す
    MOVLW     0x01
    MOVWF     STOP        ;KAITEN1 が終わった後の STOP は 1
    RETURN

```

```

KAITEN2

```

```

    BSF        PORTB,0     ;PORTB0 を 1 に設定
    BSF        PORTB,1     ;PORTB1 を 1 に設定
    CALL       HABA2       ;HABA2 を呼び出す
    CLRF       PORTB      ;PROTB をクリア
    CALL       HABA1       ;HABA1 を呼び出す
    MOVLW     0x02
    MOVWF     STOP        ;KAITEN2 が終わった後の STOP は 2
    RETURN

```

```

KAITEN3

```

```

    BSF        PORTB,1     ;PORTB1 を 1 に設定
    BSF        PORTB,2     ;PORTB2 を 1 に設定
    CALL       HABA2       ;HABA2 を呼び出す
    CLRF       PORTB      ;PROTB をクリア
    CALL       HABA1       ;HABA1 を呼び出す
    MOVLW     0x04
    MOVWF     STOP        ;KAITEN3 が終わった後の STOP は 4
    RETURN

```

```

KAITEN4

```

```

    BSF        PORTB,2     ;PORTB2 を 1 に設定
    BSF        PORTB,3     ;PORTB3 を 1 に設定
    CALL       HABA2       ;HABA2 を呼び出す
    CLRF       PORTB      ;PROTB をクリア
    CALL       HABA1       ;HABA1 を呼び出す
    MOVLW     0x08
    MOVWF     STOP        ;KAITEN4 が終わった後の STOP は 8
    RETURN

```

```

; _____

```

```

;ファイ軸 _____
TMAIN; _____ メインプログラム _____

```

```

GOTO          TMAIN1
;
;
;
TMAIN1
CALL          TADHENA
CALL          HANTEI12345
CALL          TMAINX1
GOTO         THANTEI05
;
;
;
TMAIN2
CALL          TADHENB
CALL          HANTEI12345
CALL          TMAINX2
MOVF         KAZU,W
SUBLW       0x04
MOVWF       KAZU
GOTO         THANTEI05
;
;
;

```

THANTEI05; _____ KAZU の判定
;入力電圧取得前位置との最大入力的位置が同じ (KAZU=2) なら MAINXX へ
;0 1 なら TMAIN1 へ
;3 4 なら TMAIN2 へ

```

MOVF         KAZU,W
MOVWF       TEMP05

MOVLW       0x01
MOVWF       KITEI05

MOVF         KITEI05
SUBWF      TEMP05,F
BTFSS      STATUS,C
GOTO        TMAIN1

MOVF         KITEI05
SUBWF      TEMP05,F
BTFSS      STATUS,C
GOTO        TMAIN1

MOVF         KITEI05
SUBWF      TEMP05,F
BTFSS      STATUS,C
GOTO        MAINXX
GOTO        TMAIN2
;
;
;

```

```

TMAINX1; _____ 移動
MOVF         KAZU,W
MOVWF       KAISUU

TXXX1
MOVF         KITEI,W
SUBWF      KAISUU,F
BTFSS      STATUS,C
RETURN
CALL        TGYAKUTEN
GOTO        TXXX1
;
;
;

```

```

TMAINX2; _____ 移動
MOVF         KAZU,W
MOVWF       KAISUU

TXXX2
MOVF         KITEI,W

```

```

SUBWF      KAISUU,F
BTFSS     STATUS,C
RETURN
CALL      TSEITEN
GOTO     TXXX2
;_____

TKAITEN;_____縦方向の回転
:PORTC0=X
:PORTC1=Y
:PORTC2=X パー
:PORTC3=Y パー

TSEITEN ;_____正転
BTFSC     TSTOP,0
GOTO     TKAITEN2
BTFSC     TSTOP,1
GOTO     TKAITEN3
BTFSC     TSTOP,2
GOTO     TKAITEN4
BTFSC     TSTOP,3
GOTO     TKAITEN1

TGYAKUTEN ;_____逆転
BTFSC     TSTOP,0
GOTO     TKAITEN4
BTFSC     TSTOP,1
GOTO     TKAITEN1
BTFSC     TSTOP,2
GOTO     TKAITEN2
BTFSC     TSTOP,3
GOTO     TKAITEN3
;_____
TKAITEN1
BSF      PORTC,0
BSF      PORTC,3
CALL     HABA2
CLRF     PORTC
CALL     HABA1
MOVLW   0x01
MOVWF   TSTOP
RETURN

TKAITEN2
BSF      PORTC,0
BSF      PORTC,1
CALL     HABA2
CLRF     PORTC
MOVLW   0x02
CALL     HABA1
MOVWF   TSTOP
RETURN

TKAITEN3
BSF      PORTC,1
BSF      PORTC,2
CALL     HABA2
CLRF     PORTC
CALL     HABA1
MOVLW   0x04
MOVWF   TSTOP
RETURN

TKAITEN4
BSF      PORTC,2
BSF      PORTC,3
CALL     HABA2
CLRF     PORTC
CALL     HABA1
MOVLW   0x08
MOVWF   TSTOP
RETURN
;_____
;_____

```

ADHENA;	CALL	GYAKUTEN	:GYAKUTEN を 2 度呼び出し
	CALL	GYAKUTEN	
ADHEN11	MOVLW	0X41	:A/D 変換
	MOVWF	ADCON0	
	CALL	WAIT	
	BSF	ADCON0, GO	
	CALL	TIMER1	:A/D 変換終了
	MOVF	ADRESH,W	:ADRES を AD に代入
	MOVWF	ADH1	
	BSF	STATUS, RP0	
	MOVF	ADRESL,W	
	BCF	STATUS, RP0	
	MOVWF	ADL1	
	CALL	SEITEN	
ADHEN12	MOVLW	0X41	:A/D 変換
	MOVWF	ADCON0	
	CALL	WAIT	
	BSF	ADCON0, GO	
	CALL	TIMER1	:A/D 変換終了
	MOVF	ADRESH,W	:ADRES を AD に代入
	MOVWF	ADH2	
	BSF	STATUS, RP0	
	MOVF	ADRESL,W	
	BCF	STATUS, RP0	
	MOVWF	ADL2	
	CALL	SEITEN	
ADHEN13	MOVLW	0X41	:A/D 変換
	MOVWF	ADCON0	
	CALL	WAIT	
	BSF	ADCON0, GO	
	CALL	TIMER1	:A/D 変換終了
	MOVF	ADRESH,W	:ADRES を AD に代入
	MOVWF	ADH3	
	BSF	STATUS, RP0	
	MOVF	ADRESL,W	
	BCF	STATUS, RP0	
	MOVWF	ADL3	
	CALL	SEITEN	
ADHEN14	MOVLW	0X41	:A/D 変換
	MOVWF	ADCON0	
	CALL	WAIT	
	BSF	ADCON0, GO	
	CALL	TIMER1	:A/D 変換終了
	MOVF	ADRESH,W	:ADRES を AD に代入
	MOVWF	ADH4	
	BSF	STATUS, RP0	
	MOVF	ADRESL,W	
	BCF	STATUS, RP0	
	MOVWF	ADL4	
	CALL	SEITEN	
ADHEN15	MOVLW	0X41	:A/D 変換
	MOVWF	ADCON0	
	CALL	WAIT	
	BSF	ADCON0, GO	
	CALL	TIMER1	:A/D 変換終了

	MOVF	ADRESH,W	;ADRES を AD に代入
	MOVWF	ADH5	
	BSF	STATUS, RP0	
	MOVF	ADRESL,W	
	BCF	STATUS, RP0	
	MOVWF	ADL5	
	RETURN		
:			
ADHENB:	CALL	SEITEN	;SEITEN を 2 度呼び出し
	CALL	SEITEN	
ADHEN21	MOVLW	0X41	;A/D 変換
	MOVWF	ADCON0	
	CALL	WAIT	
	BSF	ADCON0, GO	
	CALL	TIMER1	;A/D 変換終了
	MOVF	ADRESH,W	
	MOVWF	ADH1	
	BSF	STATUS, RP0	
	MOVF	ADRESL,W	
	BCF	STATUS, RP0	
	MOVWF	ADL1	
	CALL	GYAKUTEN	
ADHEN22	MOVLW	0X41	;A/D 変換
	MOVWF	ADCON0	
	CALL	WAIT	
	BSF	ADCON0, GO	
	CALL	TIMER1	;A/D 変換終了
	MOVF	ADRESH,W	;ADRES を AD に代入
	MOVWF	ADH2	
	BSF	STATUS, RP0	
	MOVF	ADRESL,W	
	BCF	STATUS, RP0	
	MOVWF	ADL2	
	CALL	GYAKUTEN	
ADHEN23	MOVLW	0X41	;A/D 変換
	MOVWF	ADCON0	
	CALL	WAIT	
	BSF	ADCON0, GO	
	CALL	TIMER1	;A/D 変換終了
	MOVF	ADRESH,W	;ADRES を AD に代入
	MOVWF	ADH3	
	BSF	STATUS, RP0	
	MOVF	ADRESL,W	
	BCF	STATUS, RP0	
	MOVWF	ADL3	
	CALL	GYAKUTEN	
ADHEN24	MOVLW	0X41	;A/D 変換
	MOVWF	ADCON0	
	CALL	WAIT	
	BSF	ADCON0, GO	
	CALL	TIMER1	;A/D 変換終了
	MOVF	ADRESH,W	;ADRES を AD に代入
	MOVWF	ADH4	
	BSF	STATUS, RP0	
	MOVF	ADRESL,W	
	BCF	STATUS, RP0	
	MOVWF	ADL4	
	CALL	GYAKUTEN	
ADHEN25			

MOVLW	0X41	:A/D 変換
MOVWF	ADCON0	
CALL	WAIT	
BSF	ADCON0, GO	
CALL	TIMER1	:A/D 変換終了
MOVF	ADRESH,W	:ADRES を AD に代入
MOVWF	ADH5	
BSF	STATUS, RP0	
MOVF	ADRESL,W	
BCF	STATUS, RP0	
MOVWF	ADL5	
RETURN		

TADHENA:

CALL	TGYAKUTEN	
CALL	TGYAKUTEN	
TADHEN11		
MOVLW	0X41	:A/D 変換
MOVWF	ADCON0	
CALL	WAIT	
BSF	ADCON0, GO	
CALL	TIMER1	:A/D 変換終了
MOVF	ADRESH,W	:ADRES を AD に代入
MOVWF	ADH1	
BSF	STATUS, RP0	
MOVF	ADRESL,W	
BCF	STATUS, RP0	
MOVWF	ADL1	
CALL	TSEITEN	
TADHEN12		
MOVLW	0X41	:A/D 変換
MOVWF	ADCON0	
CALL	WAIT	
BSF	ADCON0, GO	
CALL	TIMER1	:A/D 変換終了
MOVF	ADRESH,W	:ADRES を AD に代入
MOVWF	ADH2	
BSF	STATUS, RP0	
MOVF	ADRESL,W	
BCF	STATUS, RP0	
MOVWF	ADL2	
CALL	TSEITEN	
TADHEN13		
MOVLW	0X41	:A/D 変換
MOVWF	ADCON0	
CALL	WAIT	
BSF	ADCON0, GO	
CALL	TIMER1	:A/D 変換終了
MOVF	ADRESH,W	:ADRES を AD に代入
MOVWF	ADH3	
BSF	STATUS, RP0	
MOVF	ADRESL,W	
BCF	STATUS, RP0	
MOVWF	ADL3	
CALL	TSEITEN	
TADHEN14		
MOVLW	0X41	:A/D 変換
MOVWF	ADCON0	
CALL	WAIT	
BSF	ADCON0, GO	
CALL	TIMER1	:A/D 変換終了
MOVF	ADRESH,W	:ADRES を AD に代入
MOVWF	ADH4	

```

BSF          STATUS, RP0
MOVF         ADRESL,W
BCF          STATUS, RP0
MOVWF       ADL4

CALL         TSEITEN

TADHEN15
MOVLW       0X41          ;A/D 変換
MOVWF       ADCON0
CALL        WAIT
BSF         ADCON0, GO
CALL        TIMER1      ;A/D 変換終了

MOVF        ADRESH,W     ;ADRES を AD に代入
MOVWF       ADH5
BSF         STATUS, RP0
MOVF        ADRESL,W
BCF         STATUS, RP0
MOVWF       ADL5

RETURN
;

```

TADHENB:

```

CALL        TSEITEN
CALL        TSEITEN

TADHEN21
MOVLW       0X41          ;A/D 変換
MOVWF       ADCON0
CALL        WAIT
BSF         ADCON0, GO
CALL        TIMER1      ;A/D 変換終了

MOVF        ADRESH,W     ;ADRES を AD に代入
MOVWF       ADH1
BSF         STATUS, RP0
MOVF        ADRESL,W
BCF         STATUS, RP0
MOVWF       ADL1

CALL        TGYAKUTEN

TADHEN22
MOVLW       0X41          ;A/D 変換
MOVWF       ADCON0
CALL        WAIT
BSF         ADCON0, GO
CALL        TIMER1      ;A/D 変換終了

MOVF        ADRESH,W     ;ADRES を AD に代入
MOVWF       ADH2
BSF         STATUS, RP0
MOVF        ADRESL,W
BCF         STATUS, RP0
MOVWF       ADL2

CALL        TGYAKUTEN

TADHEN23
MOVLW       0X41          ;A/D 変換
MOVWF       ADCON0
CALL        WAIT
BSF         ADCON0, GO
CALL        TIMER1      ;A/D 変換終了

MOVF        ADRESH,W     ;ADRES を AD に代入
MOVWF       ADH3
BSF         STATUS, RP0
MOVF        ADRESL,W
BCF         STATUS, RP0
MOVWF       ADL3

CALL        TGYAKUTEN

TADHEN24
MOVLW       0X41          ;A/D 変換
MOVWF       ADCON0

```


CALL	WAIT	
BSF	ADCON0, GO	
CALL	TIMER1	;A/D 変換終了
MOVF	ADRESH,W	;ADRES を AD に代入
MOVWF	ADH4	
BSF	STATUS, RP0	
MOVF	ADRESL,W	
BCF	STATUS, RP0	
MOVWF	ADL4	
CALL	TGYAKUTEN	
TADHEN25		
MOVLW	0X41	;A/D 変換
MOVWF	ADCON0	
CALL	WAIT	
BSF	ADCON0, GO	
CALL	TIMER1	;A/D 変換終了
MOVF	ADRESH,W	;ADRES を AD に代入
MOVWF	ADH5	
BSF	STATUS, RP0	
MOVF	ADRESL,W	
BCF	STATUS, RP0	
MOVWF	ADL5	
RETURN		
;		
HANTEI12345		
;		
HANTEI12		
;AD1 と AD2 のどちらが大きいかが判定		
;		
HANTEI121		
MOVF	ADH1,W	
SUBWF	ADH2,W	
BTFSC	STATUS,C	
GOTO	HANTEI122	
GOTO	H1	
;		
HANTEI122		
MOVF	ADH2,W	
SUBWF	ADH1,W	
BTFSC	STATUS,C	
GOTO	HANTEI123	
GOTO	H2	
;		
HANTEI123		
MOVF	ADL1,W	
SUBWF	ADL2,W	
BTFSC	STATUS,C	
GOTO	HANTEI124	
GOTO	H1	
;		
HANTEI124		
MOVF	ADL2,W	
SUBWF	ADL1,W	
BTFSC	STATUS,C	
GOTO	H2	;同じ場合は AD2 を優先
GOTO	H2	
;		
H1		
;AD1 が大きい場合		
MOVLW	0X04	;KAZU に 4 を代入
MOVWF	KAZU	
MOVF	ADH1,W	;KIOKU に ADH1 を代入
MOVWF	KIOKUH	
MOVF	ADL1,W	
MOVWF	KIOKUL	

H2	GOTO	HANTEI23	:HANTEI23 へ
;AD2 が大きい場合			
	MOVLW	0X03	:KAZU に 3 を代入
	MOVWF	KAZU	
	MOVF	ADH2,W	:KIOKU に ADH2 を代入
	MOVWF	KIOKUH	
	MOVF	ADL2,W	
	MOVWF	KIOKUL	
	GOTO	HANTEI23	:HANTEI23 へ

HANTEI23
;KIOKU と AD3 のどちらが大きい判定

HANTEI231

MOVF	KIOKUH,W
SUBWF	ADH3,W
BTFSC	STATUS,C
GOTO	HANTEI232
GOTO	HK2

HANTEI232

MOVF	ADH3,W
SUBWF	KIOKUH,W
BTFSC	STATUS,C
GOTO	HANTEI233
GOTO	H3

HANTEI233

MOVF	KIOKUL,W
SUBWF	ADL3,W
BTFSC	STATUS,C
GOTO	HANTEI234
GOTO	HK2

HANTEI234

MOVF	ADL3,W	
SUBWF	KIOKUL,W	
BTFSC	STATUS,C	
GOTO	H3	:同じ場合は AD3 を優先
GOTO	H3	

H3

;AD3 が大きい場合			
	MOVLW	0X02	:KAZU に 2 を代入
	MOVWF	KAZU	
	MOVF	ADH3,W	:KIOKU に ADH3 を代入
	MOVWF	KIOKUH	
	MOVF	ADL3,W	
	MOVWF	KIOKUL	

HK2

;KIOKU が大きい場合			
	GOTO	HANTEI34	:HANTEI34 へ

HANTEI34
;KIOKU と AD4 のどちらが大きい判定

HANTEI341

MOVF	KIOKUH,W
SUBWF	ADH4,W
BTFSC	STATUS,C
GOTO	HANTEI342
GOTO	HK3

HANTEI342

MOVF	ADH4,W
------	--------

```

SUBWF      KIOKUH,W
BTFSC     STATUS,C
GOTO      HANTEI343
GOTO      H4
;
HANTEI343
MOVF      KIOKUL,W
SUBWF     ADL4,W
BTFSC     STATUS,C
GOTO      HANTEI344
GOTO      HK3
;
HANTEI344
MOVF      ADL4,W
SUBWF     KIOKUL,W
BTFSC     STATUS,C
GOTO      HK3           ;同じ場合は KIOKU を優先
GOTO      H4
;
H4
;AD4 が大きい場合
MOVLW    0X01           ;KAZU に 1 を代入
MOVWF    KAZU

MOVF     ADH4,W         ;KIOKU に ADH4 を代入
MOVWF   KIOKUH
MOVF    ADL4,W
MOVWF   KIOKUL

HK3
;KIOKU が大きい場合
GOTO    HANTEI45       ;HANTEI45 へ
;
HANTEI45
;KIOKU と AD5 のどちらが大きい判定
;
HANTEI451
MOVF      KIOKUH,W
SUBWF     ADH5,W
BTFSC     STATUS,C
GOTO      HANTEI452
GOTO      HK4
;
HANTEI452
MOVF      ADH5,W
SUBWF     KIOKUH,W
BTFSC     STATUS,C
GOTO      HANTEI453
GOTO      H5
;
HANTEI453
MOVF      KIOKUL,W
SUBWF     ADL5,W
BTFSC     STATUS,C
GOTO      HANTEI454
GOTO      HK4
;
HANTEI454
MOVF      ADL5,W
SUBWF     KIOKUL,W
BTFSC     STATUS,C
GOTO      HK4
GOTO      H5
;
H5
;AD5 が大きい場合
MOVLW    0X00           ;KAZU に 0 を代入
MOVWF    KAZU

MOVF     ADH5,W         ;KIOKU に ADH5 を代入
MOVWF   KIOKUH
MOVF    ADL5,W
MOVWF   KIOKUL

```

```

HK4
;KIOKU が大きい場合
    RETURN                ;戻る
;

```

```

;----- パルス幅
HABA1;----- 0.1 ミリ S のループ
    MOVLW                0x3E
    MOVWF                HL1
HABAL1
    NOP
    DECFSZ HL1,F
    GOTO                HABAL1
    RETURN

```

```

HABA2;----- 10 ミリ S のループ
    MOVLW                0x40
    MOVWF                HL2
HABAL2
    NOP
    CALL                HABA1
    DECFSZ HL2,F
    GOTO                HABAL2
    RETURN

```

```

HABA3;----- 0.5 秒のループ
    MOVLW                0x32
    MOVWF                HL3
HABAL3
    NOP
    CALL                HABA2
    DECFSZ HL3,F
    GOTO                HABAL3
    RETURN
;
;-----
;

```

```

;-----
WAIT
    MOVLW                0FH
    MOVWF                CNT1
LOOP1
    NOP
    NOP
    NOP
    NOP
    NOP
    NOP
    NOP
    NOP
    CALL                SUB2
    DECFSZ CNT1,F
    GOTO                LOOP1
    RETURN
SUB2
    MOVLW                03FH
    MOVWF                CNT2
LOOP2
    NOP
    NOP
    NOP
    NOP
    NOP
    NOP
    DECFSZ CNT2,F
    GOTO                LOOP2
    RETURN
;
;-----
;

```

```

TIMER1
    BTFSC                ADCON0, GO
    GOTO                TIMER1
    RETURN

```

;A/D 変換が終了するのを待つループ

;

END

5. 5 プログラムリスト5

```

LIST      P=PIC16F874
INCLUDE "P16F874.INC"
__CONFIG0X3D3A

ADL1     EQU     020H
ADH1     EQU     021H
ADL2     EQU     022H
ADH2     EQU     023H
ADL3     EQU     024H
ADH3     EQU     025H
ADL4     EQU     026H
ADH4     EQU     027H
HKAZU    EQU     028H
SSTOP    EQU     029H
FSTOP    EQU     030H
KITEI    EQU     031H
KAZU     EQU     032H
HL1      EQU     033H
HL2      EQU     034H
HL3      EQU     035H
CNT1     EQU     036H
CNT2     EQU     037H

KEIL     EQU     038H
KEIH     EQU     039H
KL       EQU     040H
KH       EQU     041H
TEMP     EQU     042H
DOUSA    EQU     043H
TEMP0    EQU     044H

START    ORG     0

BSF      STATUS, RP0      ;バンク 1 を選択
CLRF     PORTB             ;PORTB はすべてデジタル出力
CLRF     PORTC             ;PORTC はすべてデジタル出力
CLRF     PORTD             ;PORTD はすべてデジタル出力
MOVLW   0X80
MOVWF   ADCON1
BCF      STATUS, RP0      ;バンク 0 を選択
CLRF     PORTB
CLRF     PORTC
CLRF     PORTD

MOVLW   0x01
MOVWF   SSTOP              ;SSTOP に 0X01 を代入

MOVLW   0x01
MOVWF   FSTOP              ;FSTOP に 0X01 を代入

MOVLW   0x01
MOVWF   KITEI              ;KITEI に 0x01 を代入

MOVLW   0xFF
MOVWF   KAZU               ;KAZU に 0xFF を代入

MOVLW   0x00
MOVWF   HKAZU              ;HKAZU に 0x00 を代入

MOVLW   0xFF
MOVWF   TEMP0

MOVLW   0x54
MOVWF   TEMP               ;TEMP に 84 を代入

MOVLW   0x00
MOVWF   DOUSA              ;DOUSA に 0 を代入

SMAIN; _____ θ 軸について
ADHEN1
MOVLW   0X41
MOVWF   ADCON0             ;RA0 の/AD 変換を開始

```

```

CALL          WAIT
BSF          ADCON0, GO
CALL        TIMER1          ;A/D 変換終了
MOVF        ADRESH,W
MOVWF       ADH1            ;ADRESH を ADH1 に格納
INCF        ADH1,F          ;ADH1+1=ADH1
BSF          STATUS, RP0
MOVF        ADRESL,W
BCF         STATUS, RP0
MOVWF       ADL1            ;ADRESL を ADL1 に格納

ADHEN2
MOVLW      0X49
MOVWF      ADCON0          ;RA1 の A/D 変換を開始
CALL        WAIT
BSF        ADCON0, GO
CALL        TIMER1          ;A/D 変換終了
MOVF        ADRESH,W
MOVWF       ADH2            ;ADRESH を ADH2 に格納
INCF        ADH2,F          ;ADH2+1=ADH2
BSF          STATUS, RP0
MOVF        ADRESL,W
BCF         STATUS, RP0
MOVWF       ADL2            ;ADRESL を ADL2 に格納
;
GOTO        SHANTEI        ;SHANTEI へ移動
;

SAAA
GOTO        SMAIN          ;SMAIN へ移動
;

;
SHANTEI
;θ 軸の素子間の大小判定
;AD1 が大きければ H1 へ
;AD2 が大きければ H2 へ
;同じなら SONAJI へ
;
SHANTEI1
MOVF        ADH1,W
SUBWF       ADH2,W
BTFSC      STATUS,C
GOTO        SHANTEI2
GOTO        H1
;
SHANTEI2
MOVF        ADH2,W
SUBWF       ADH1,W
BTFSC      STATUS,C
GOTO        SHANTEI3
GOTO        H2
;
SHANTEI3
MOVF        ADL1,W
SUBWF       ADL2,W
BTFSC      STATUS,C
GOTO        SHANTEI4
GOTO        H1
;
SHANTEI4
MOVF        ADL2,W
SUBWF       ADL1,W
BTFSC      STATUS,C
GOTO        SONAJI
GOTO        H2
;
SONAJI

```

	GOTO	FMAIN	:FMAIN (φ軸の判定) へ
;			
H1	;AD1の方が大きい場合		
	MOVF	ADL2,W	:KEIにAD2を代入
	MOVWF	KEIL	
	MOVF	ADH2,W	
	MOVWF	KEIH	
	MOVF	ADL1,W	; K にAD1を代入
	MOVWF	KL	
	MOVF	ADH1,W	
	MOVWF	KH	
	MOVF	TEMP,W	:K-TEMP=K
	SUBWF	KL,F	
	BTFSS	STATUS,C	
	DECf	KH,F	
	GOTO	SHANTEIX1	:AD2との差が許容範囲内かどうかの判定へ
H1KAI	CLRF	PORTD	:PORTDをクリア
	INCF	DOUSA,F	:DOUSAに1足してPORTDに出力
	MOVF	DOUSA,W	
	MOVWF	PORTD	
	BTFSS	HKAZU,0	
	GOTO	SHG	:HKAZUが0ならSHG
	GOTO	SHS	:HKAZUが1ならSHS
H2	MOVF	ADL1,W	:KEIにAD1を代入
	MOVWF	KEIL	
	MOVF	ADH1,W	
	MOVWF	KEIH	
	MOVF	ADL2,W	; K にAD2を代入
	MOVWF	KL	
	MOVF	ADH2,W	
	MOVWF	KH	
	MOVF	TEMP,W	:K-TEMP=K
	SUBWF	KL,F	
	BTFSS	STATUS,C	
	DECf	KH,F	
	GOTO	SHANTEIX2	:AD1との差が許容範囲内かどうかの判定へ
H2KAI	CLRF	PORTD	:PORTDをクリア
	INCF	DOUSA,F	:DOUSAに1足してPORTDに出力
	MOVF	DOUSA,W	
	MOVWF	PORTD	
	BTFSS	HKAZU,0	
	GOTO	SHS	:HKAZUが0ならSHS
	GOTO	SHG	:HKAZUが1ならSHG
SHS	CALL	SSEITEN	:SSEITENを呼び出し
	GOTO	SAAA	:SAAAへ
SHG	CALL	SGYAKUTEN	:SGYAKUTENを呼び出し
	GOTO	SAAA	:SAAAへ
SH	GOTO	FMAIN	:FMAIN (φ軸の判定) へ
;			
SHANTEIX1			
:KEIが大きければ許容範囲内→SHへ			
:Kが大きければ許容範囲外→H1KAIへ			


```

;
SHANTEIX11
    MOVF      KEIH,W
    SUBWF    KH,W
    BTFSC    STATUS,C
    GOTO     SHANTEIX12
    GOTO     SH

```

```

;
SHANTEIX12
    MOVF      KH,W
    SUBWF    KEIH,W
    BTFSC    STATUS,C
    GOTO     SHANTEIX13
    GOTO     H1KAI

```

```

;
SHANTEIX13
    MOVF      KEIL,W
    SUBWF    KL,W
    BTFSC    STATUS,C
    GOTO     SHANTEIX14
    GOTO     SH

```

```

;
SHANTEIX14
    MOVF      KL,W
    SUBWF    KEIL,W
    BTFSC    STATUS,C
    GOTO     SH
    GOTO     H1KAI

```

SHANTEIX2

;KEI が大きければ許容範囲内→SH へ
; K が大きければ許容範囲外→HKAI へ

```

;
SHANTEIX21
    MOVF      KEIH,W
    SUBWF    KH,W
    BTFSC    STATUS,C
    GOTO     SHANTEIX22
    GOTO     SH

```

```

;
SHANTEIX22
    MOVF      KH,W
    SUBWF    KEIH,W
    BTFSC    STATUS,C
    GOTO     SHANTEIX23
    GOTO     H2KAI

```

```

;
SHANTEIX23
    MOVF      KEIL,W
    SUBWF    KL,W
    BTFSC    STATUS,C
    GOTO     SHANTEIX24
    GOTO     SH

```

```

;
SHANTEIX24
    MOVF      KL,W
    SUBWF    KEIL,W
    BTFSC    STATUS,C
    GOTO     SH
    GOTO     H2KAI

```

FMAIN; _____ φ軸について

```

;
ADHEN3
    MOVLW    0X51
    MOVWF   ADCON0      ;RA2 の A/D 変換を開始
    CALL    WAIT

```

```

BSF          ADCON0, GO
CALL        TIMER1          ;A/D 変換終了
MOVWF      ADRESH,W
MOVWF      ADH3             ;ADRESH を ADH3 に格納
INCF       ADH3,F          ;ADH3+1=ADH3
BSF        STATUS, RP0
MOVWF      ADRESL,W
BCF        STATUS, RP0
MOVWF      ADL3           ;ADRESL を ADL3 に格納

ADHEN4
MOVLW     0X59
MOVWF     ADCON0          ;RA3 の A/D 変換を開始
CALL     WAIT
BSF      ADCON0, GO
CALL     TIMER1          ;A/D 変換終了
MOVWF    ADRESH,W
MOVWF    ADH4             ;ADRESH を ADH4 に格納
INCF     ADH4,F          ;ADH4+1=ADH4
BSF      STATUS, RP0
MOVWF    ADRESL,W
BCF      STATUS, RP0
MOVWF    ADL4           ;ADRESL を ADL4 に格納

;
;
FAAA      GOTO      FHANTEI      ;FNAITEI へ
          GOTO      FMAIN       ;FMAIN へ
;
;
;
FHANTEI
;φ 軸の素子間の大小判定
;AD3 が大きければ H3 へ
;AD4 が大きければ H4 へ
;同じなら FONAJI へ
;
FHANTEI1
MOVWF    ADH3,W
SUBWF    ADH4,W
BTFSC   STATUS,C
GOTO     FHANTEI2
GOTO     H3
;
FHANTEI2
MOVWF    ADH4,W
SUBWF    ADH3,W
BTFSC   STATUS,C
GOTO     FHANTEI3
GOTO     H4
;
FHANTEI3
MOVWF    ADL3,W
SUBWF    ADL4,W
BTFSC   STATUS,C
GOTO     FHANTEI4
GOTO     H3
;
FHANTEI4
MOVWF    ADL4,W
SUBWF    ADL3,W
BTFSC   STATUS,C
GOTO     FONAJI
GOTO     H4
;
FONAJI
          GOTO      SMAIN        ;SMAIN (θ 軸の判定) へ
;
H3

```

	MOVF	ADL4,W	;AD4 を KEI に代入
	MOVWF	KEIL	
	MOVF	ADH4,W	
	MOVWF	KEIH	
	MOVF	ADL3,W	;AD3 を K に代入
	MOVWF	KL	
	MOVF	ADH3,W	
	MOVWF	KH	
	MOVF	TEMP,W	; K -TEMP= K
	SUBWF	KL,F	
	BTFSS	STATUS,C	
	DECf	KH,F	
	GOTO	FHANTEIX3	;FHANTEIX3 へ
H3KAI	CLRF	PORTD	;PORTD をクリア
	INCF	DOUSA,F	;DOUSA に 1 足して PORTD に出力
	MOVF	DOUSA,W	
	MOVWF	PORTD	
	CALL	FSEITEN	;FSEITEN を呼び出し
	MOVF	KITEI,W	;KAZU-KITEI=KAZU
	SUBWF	KAZU,F	
	BTFSS	STATUS,C	
	DECf	HKAZU,F	
	GOTO	FAAA	;FAAA へ
H4	MOVF	ADL3,W	;AD3 を KEI に代入
	MOVWF	KEIL	
	MOVF	ADH3,W	
	MOVWF	KEIH	
	MOVF	ADL4,W	;AD4 を K に代入
	MOVWF	KL	
	MOVF	ADH4,W	
	MOVWF	KH	
	MOVF	TEMP,W	; K -TEMP= K
	SUBWF	KL,F	
	BTFSS	STATUS,C	
	DECf	KH,F	
	GOTO	FHANTEIX4	;FHANTEIX4 へ
H4KAI	CLRF	PORTD	;PORTD をクリア
	INCF	DOUSA,F	;DOUSA に 1 足して PORTD に出力
	MOVF	DOUSA,W	
	MOVWF	PORTD	
	CALL	FGYAKUTEN	;FGYAKUTEN を呼び出し
	MOVF	KITEI,W	
	ADDWF	KAZU,F	
	BTFSC	STATUS,C	
	INCF	HKAZU,F	;KAZU+KITEI=KAZU
	GOTO	FAAA	;FAAA へ
FH	GOTO	SMAIN	;SMAIN (θ 軸の判定) へ

;

FHANTEIX3

;KEI が大きければ許容範囲内→FH へ

;K が大きければ許容範囲外→H3KAI へ

```

;
FHANTEIX31
    MOVF          KEIH,W
    SUBWF        KH,W
    BTFSC        STATUS,C
    GOTO         FHANTEIX32
    GOTO         FH

```

```

;
FHANTEIX32
    MOVF          KH,W
    SUBWF        KEIH,W
    BTFSC        STATUS,C
    GOTO         FHANTEIX33
    GOTO         H3KAI

```

```

;
FHANTEIX33
    MOVF          KEIL,W
    SUBWF        KL,W
    BTFSC        STATUS,C
    GOTO         FHANTEIX34
    GOTO         FH

```

```

;
FHANTEIX34
    MOVF          KL,W
    SUBWF        KEIL,W
    BTFSC        STATUS,C
    GOTO         FH
    GOTO         H3KAI

```

FHANTEIX4

```

;KEI が大きければ許容範囲内→FH へ
;K が大きければ許容範囲外→H4KAI へ

```

```

;
FHANTEIX41
    MOVF          KEIH,W
    SUBWF        KH,W
    BTFSC        STATUS,C
    GOTO         FHANTEIX42
    GOTO         FH

```

```

;
FHANTEIX42
    MOVF          KH,W
    SUBWF        KEIH,W
    BTFSC        STATUS,C
    GOTO         FHANTEIX43
    GOTO         H4KAI

```

```

;
FHANTEIX43
    MOVF          KEIL,W
    SUBWF        KL,W
    BTFSC        STATUS,C
    GOTO         FHANTEIX44
    GOTO         FH

```

```

;
FHANTEIX44
    MOVF          KL,W
    SUBWF        KEIL,W
    BTFSC        STATUS,C
    GOTO         FH
    GOTO         H4KAI

```

```

; _____ パルス幅
HABA1; _____ 0. 1 ミリ S のループ
    MOVLW        0x3E
    MOVWF        HL1
HABAL1
    NOP
    DECFSZ      HL1,F
    GOTO        HABAL1
    RETURN
HABA2; _____ 1 0 ミリ S のループ

```

```

        MOVLW      0x64
        MOVWF     HL2
HABAL2
        NOP
        CALL      HABA1
        DECFSZ   HL2,F
        GOTO     HABAL2
        RETURN
HABA3;_____ 0. 5秒のループ
        MOVLW      0x32
        MOVWF     HL3
HABAL3
        NOP
        CALL      HABA2
        DECFSZ   HL3,F
        GOTO     HABAL3
        RETURN
;_____
;_____
WAIT;_____ 蓄積時間
        MOVLW      0FH
        MOVWF     CNT1
LOOP1
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        CALL      SUB2
        DECFSZ   CNT1,F
        GOTO     LOOP1
        RETURN
SUB2
        MOVLW      03FH
        MOVWF     CNT2
LOOP2
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        DECFSZ   CNT2,F
        GOTO     LOOP2
        RETURN
;_____
;_____
TIMER1
        BTFSC     ADCON0, GO      ;A/D 変換が終了するのを待つループ
        GOTO     TIMER1         ;ADCON GO が 0 なら A/D 変換終了
        RETURN
;_____
SKAITEN;_____ 回転
;PORTB0=X
;PORTB1=Y
;PORTB2=X バー
;PORTB3=Y バー
SSEITEN ;_____ 正転
        BTFSC     SSTOP,0
        GOTO     SKAITEN2
        BTFSC     SSTOP,1
        GOTO     SKAITEN3
        BTFSC     SSTOP,2
        GOTO     SKAITEN4
        BTFSC     SSTOP,3
        GOTO     SKAITEN1
SGYAKUTEN ;_____ 逆転

```

BTFSC	SSTOP,0
GOTO	SKAITEN4
BTFSC	SSTOP,1
GOTO	SKAITEN1
BTFSC	SSTOP,2
GOTO	SKAITEN2
BTFSC	SSTOP,3
GOTO	SKAITEN3

SKAITEN1

BSF	PORTB,0
BSF	PORTB,3
CALL	HABA2
CLRF	PORTB
CALL	HABA1
MOVLW	0x01
MOVWF	SSTOP
RETURN	

SKAITEN2

BSF	PORTB,0
BSF	PORTB,1
CALL	HABA2
CLRF	PORTB
CALL	HABA1
MOVLW	0x02
MOVWF	SSTOP
RETURN	

SKAITEN3

BSF	PORTB,1
BSF	PORTB,2
CALL	HABA2
CLRF	PORTB
CALL	HABA1
MOVLW	0x04
MOVWF	SSTOP
RETURN	

SKAITEN4

BSF	PORTB,2
BSF	PORTB,3
CALL	HABA2
CLRF	PORTB
CALL	HABA1
MOVLW	0x08
MOVWF	SSTOP
RETURN	

FKAITEN; _____ 回転

```
;PORTC0=X
;PORTC1=Y
;PORTC2=Xノバー
;PORTC3=Yノバー
```

FSEITEN ; _____ 正転

BTFSC	FSTOP,0
GOTO	FKAITEN2
BTFSC	FSTOP,1
GOTO	FKAITEN3
BTFSC	FSTOP,2
GOTO	FKAITEN4
BTFSC	FSTOP,3
GOTO	FKAITEN1

FGYAKUTEN; _____ 逆転

BTFSC	FSTOP,0
GOTO	FKAITEN4
BTFSC	FSTOP,1
GOTO	FKAITEN1

```

        BTFSC          FSTOP,2
        GOTO          FKAITEN2
        BTFSC          FSTOP,3
        GOTO          FKAITEN3

FKAITEN1
        BSF           PORTC,0
        BSF           PORTC,3
        CALL          HABA2
        CLRF          PORTC
        CALL          HABA1
        MOVLW         0x01
        MOVWF         FSTOP
        RETURN

FKAITEN2
        BSF           PORTC,0
        BSF           PORTC,1
        CALL          HABA2
        CLRF          PORTC
        CALL          HABA1
        MOVLW         0x02
        MOVWF         FSTOP
        RETURN

FKAITEN3
        BSF           PORTC,1
        BSF           PORTC,2
        CALL          HABA2
        CLRF          PORTC
        CALL          HABA1
        MOVLW         0x04
        MOVWF         FSTOP
        RETURN

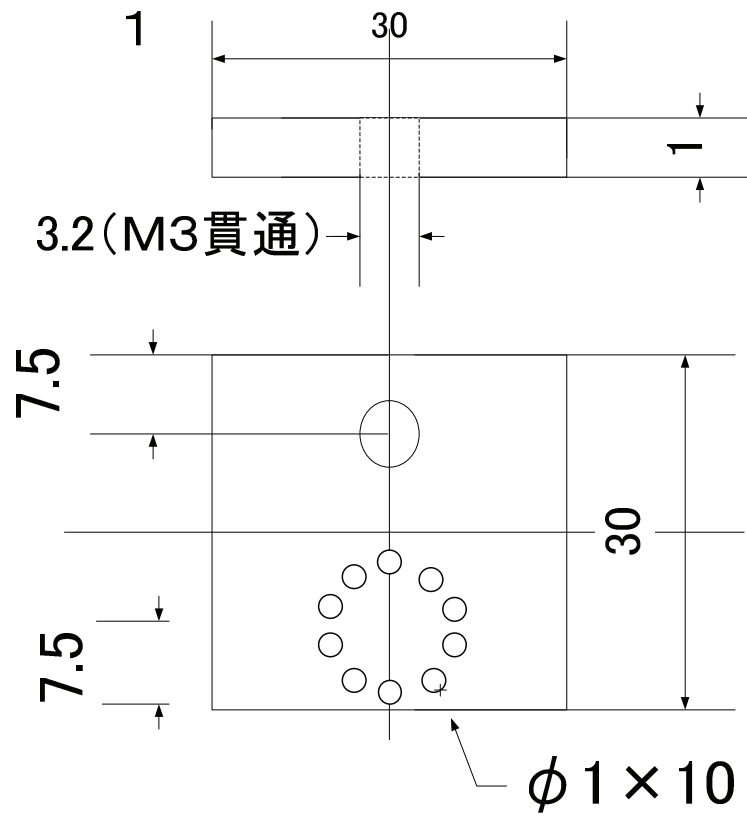
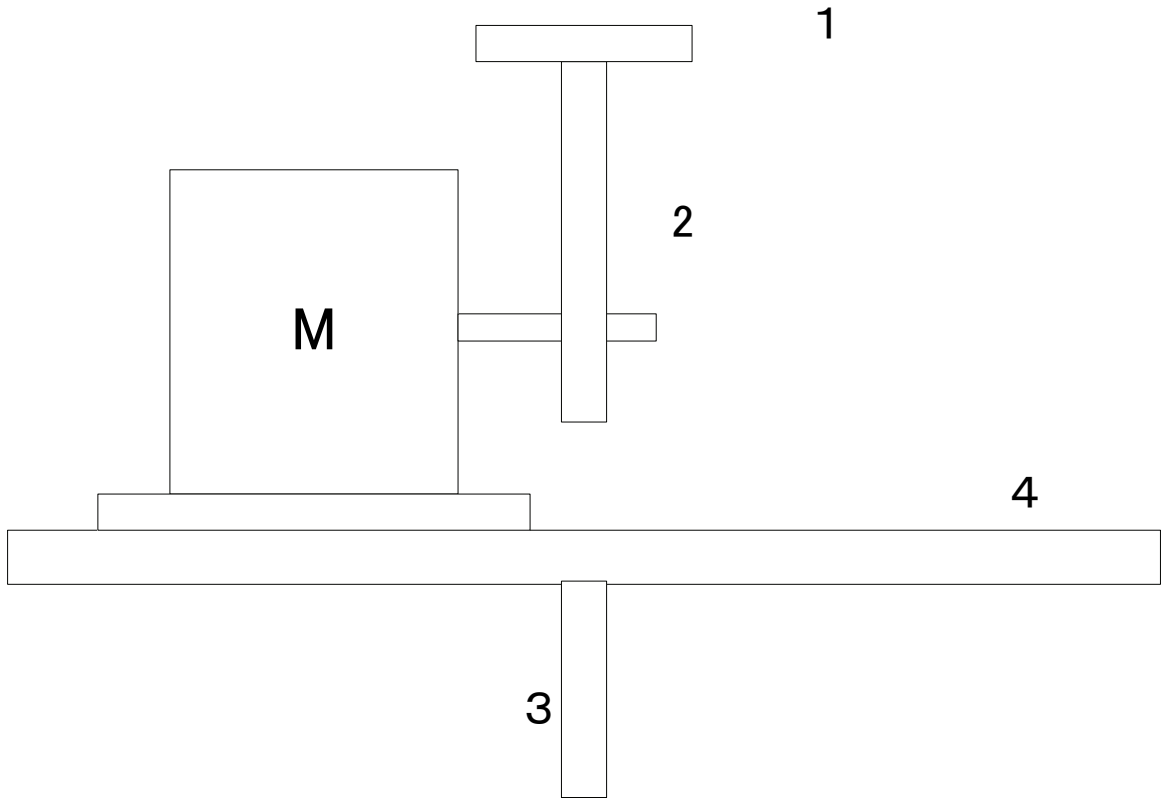
FKAITEN4
        BSF           PORTC,2
        BSF           PORTC,3
        CALL          HABA2
        CLRF          PORTC
        CALL          HABA1
        MOVLW         0x08
        MOVWF         FSTOP
        RETURN
;
;
END

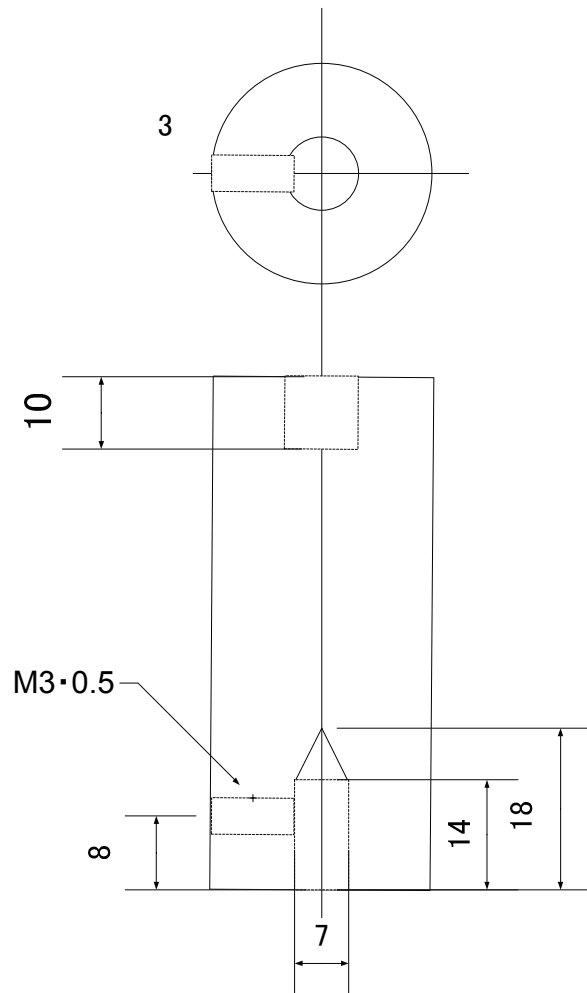
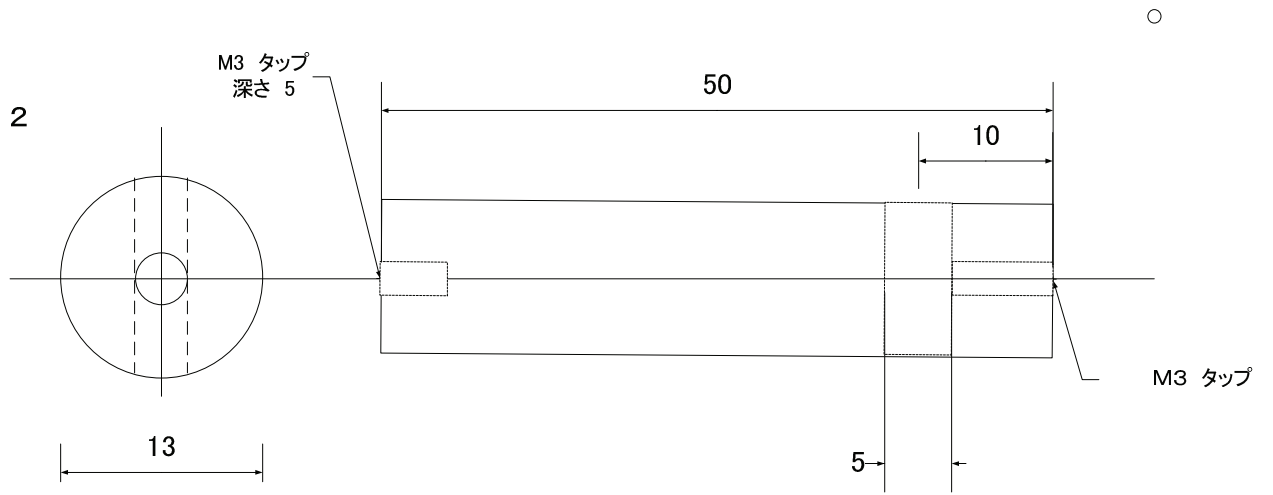
```

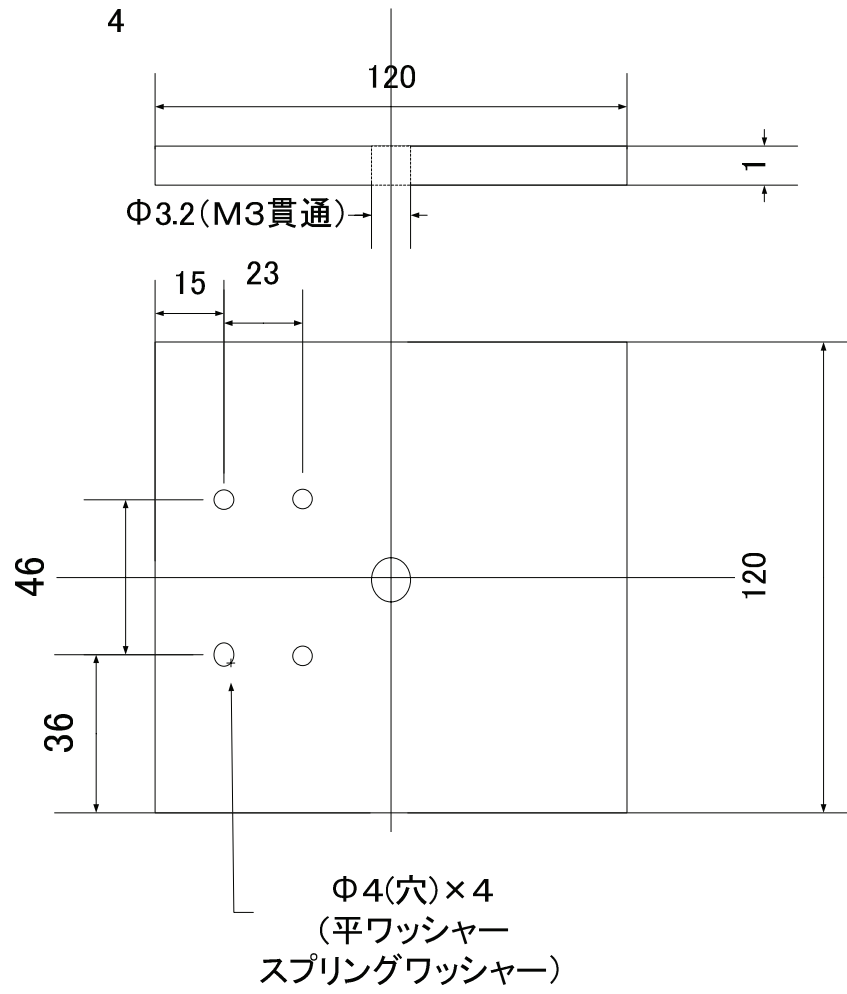
謝辞

本研究を遂行するにあたって、多数の文献をお借りし、研究においては丁寧なご指導、また、論文の推敲にも大変なお力添えを頂いた本校 電子制御工学科 由井四海 助手、制御情報システム工学専攻科 中谷健一 専攻科生に深く感謝いたします

付録1 2軸システムの設計図







付録2 1素子を使用した場合のラベルについてのフローチャートとラベルの説明

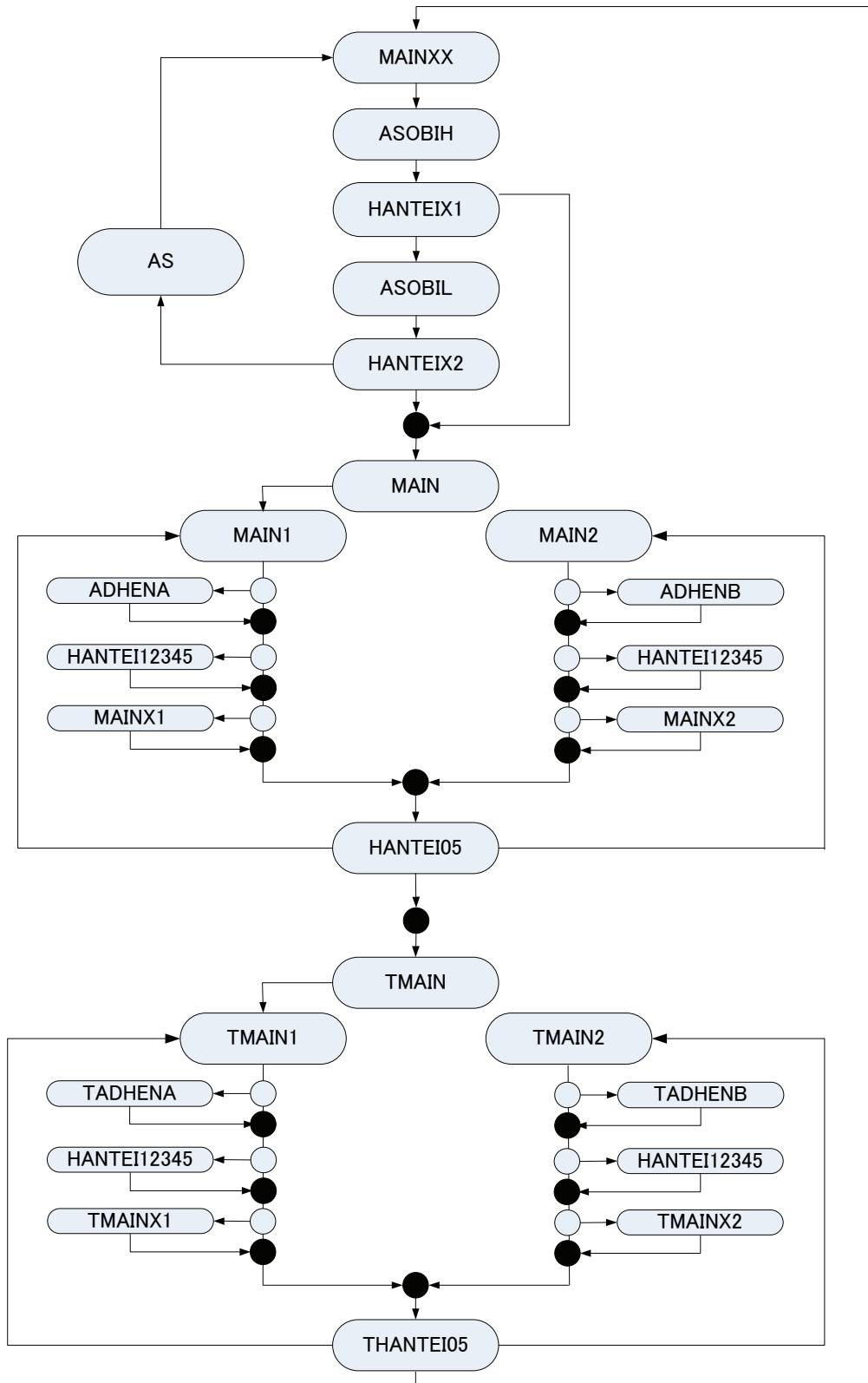


図5-1 1素子を使用した場合のラベルについてのフローチャート

ラベル名：内容の説明

START： 初期設定を行うラベル。

MAINXX： 縦方向での位置検出が F 終わった後に通るラベル。A/D 変換を行い、得た入力電圧を KEI に保存する。MAINXX で得た入力電圧 KEI と最後に得た入力電圧 KIOKU との差が許容範囲内であればこのループにとどまる。

ASOBILH： KEI に許容範囲の上限の幅 TEMP1 を加算し、許容範囲の上限 KH を得るラベル。KH を得た後、HANTEIX1 へ移動する。

HANTEIX1： KH と KEI の大小判定を行うラベル。KH が大きければ ASOBIL へ、KEI が大きければ MAIN へ、また同じであれば ASOBIL へ移動する。

ASOBIL： KEI に許容範囲の下限の幅 TEMP2 を減算し、許容範囲の下限 KL を得るラベル。KL を得た後、HANTEIX2 へ移動する。

HANTEIX2： KL と KEI の大小判定を行うラベル。KL が大きければ MAIN へ、KEI が大きければ AS へ、また同じ場合も AS へ移動する。

AS： HANTEIX1, HANTEIX2 で判定した結果、許容範囲内であれば通過するラベル。HABA2 を呼び出した後、MAINXX へ移動する。

θ 軸について光源を追従するラベルについて

MAIN： このラベルを通るのは、HANTEIX1, HANTEIX2 から移動した場合のみである。MAIN1 へ移動する。

MAIN1： KAZU が 0 あるいは 1 の場合に通るラベル。ADHENA, HANTEI12345, MAINX1 を呼び出した後、HANTEI05 へ移動する。

MAIN2： KAZU が 3 あるいは 4 の場合に通るラベル。ADHENB, HANTEI12345, MAINX2 を呼び出した後、KAZU を MAIN1 の場合と合わせる計算 ($4 - KAZU = KAZU$) を行い、HANTEI05 へ移動する。

ADHENA： θ 軸方向での 5 つの入力電圧を得るためのラベル。

ADHENB： θ 軸方向での 5 つの入力電圧を得るためのラベル。ADHENA とは入力電圧を得る順番が違う。移動効率を考慮して、このラベルを設けた。

HANTEI12345： AD1～AD5 のなかで最も大きい値を判定し、どの入力電圧が大きいかによ

って異なった KAZU を与えるラベル。入力電圧が同じ場合は AD3 を優先的に扱う。

MAINX1 : MAIN1 から呼び出され、回転数を制御するラベル。HANTEI12345 で与えられた KAZU の値だけ GYAKUTEN を呼び出し、終了すると RETURN する。

MAINX2 : MAIN2 から呼び出され、回転数を制御するラベル。HANTEI12345 で与えられた KAZU の値だけ SEITEN を呼び出し、終了すると RETURN する。

HANTEI05 : KAZU の値から、次にどのような動作を行うか判定するラベル。KAZU=2 であれば、位置が変わっていないので、θ 軸の判定が完了したとみなし、TMAIN へ移動する。KAZU が 0 あるいは 1 の場合は MAIN1 へ移動し、3 あるいは 4 の場合は MAIN2 へ移動する。

φ 軸について光源を追従するラベルについて

TMAIN : このラベルを通るのは、HANTEI05 から移動した場合のみである。MAIN1 へ移動する。

MAIN1 : KAZU が 0 あるいは 1 の場合に通るラベル。TADHENA, HANTEI12345, TMAINX1 を呼び出した後、THANTEI05 へ移動する。

TMAIN2 : KAZU が 3 あるいは 4 の場合に通るラベル。TADHENB, HANTEI12345, TMAINX2 を呼び出した後、KAZU を TMAIN1 の場合と合わせる計算 (4 - KAZU = KAZU) を行い、THANTEI05 へ移動する。

TADHENA : φ 軸方向での 5 つの入力電圧を得るためのラベル。

TADHENB : φ 軸方向での 5 つの入力電圧を得るためのラベル。TADHENA とは入力電圧を得る順番が違う。移動効率を考えて、このラベルを設けた。

TMAINX1 : TMAIN1 から呼び出され、回転数を制御するラベル。HANTEI12345 で与えられた KAZU の値だけ TGYAKUTEN を呼び出し、終了すると RETURN する。

TMAINX2 : TMAIN2 から呼び出され、回転数を制御するラベル。HANTEI12345 で与えられた KAZU の値だけ TSEITEN を呼び出し、終了すると RETURN する。

THANTEI05 : KAZU の値から、次にどのような動作を行うか判定するラベル。KAZU=2 で

あれば、位置が変わっていないので、 ϕ 軸の判定が完了したとみなし、MAINXX
へ移動する。KAZU が 0 あるいは 1 の場合は TMAIN1 へ移動し、3 あるいは 4
の場合は TMAIN2 へ移動する。

付録3 4素子を使用した場合のラベルについてのフローチャートとラベルの説明

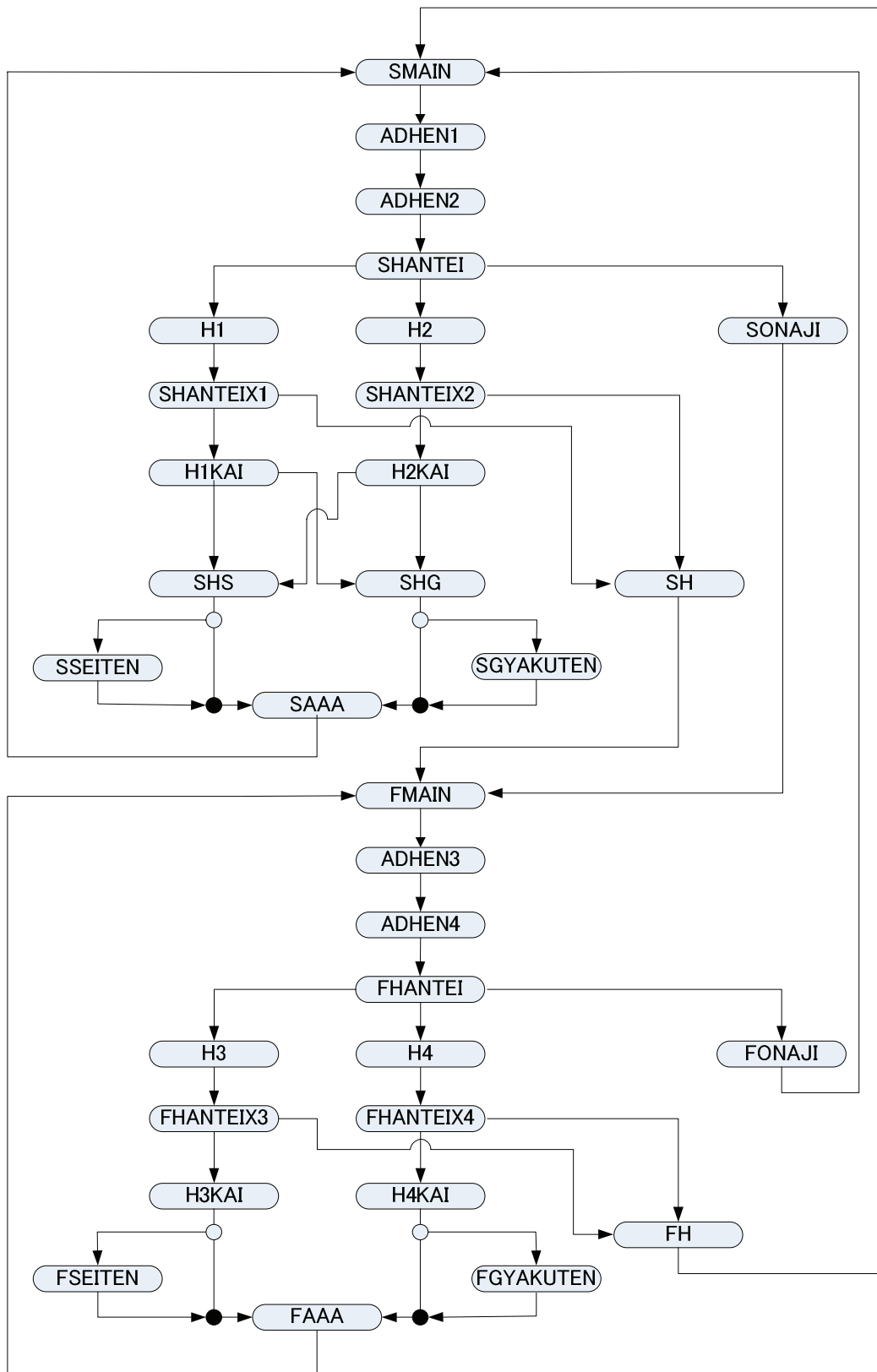


図5-2 4素子を使用した場合のラベルについてのフローチャート

ラベル名：内容の説明

- SMAIN :** θ 軸方向についてのラベル。ADHEN1, ADHEN2 を通った後、SHANTEI へ移動する。
- ADHEN1 :** RA0 に入力した電圧を A/D 変換し、取得したデジタル値を AD1 に保存し、ADH1 に 1 を足す。
- ADHEN2 :** RA1 に入力した電圧を A/D 変換し、取得したデジタル値を AD2 に保存し、ADH2 に 1 を足す。
- SHANTEI :** θ 軸方向の素子間の入力電圧 AD1, AD2 の大小判定を行うラベル。AD1 が大きければ H1 へ、AD2 が大きければ H2 へ、同じ場合は SONAJI へ移動する。
- SONAJI :** AD1 と AD2 が同じである場合に通るラベル。 θ 軸方向の各素子への入力電圧が同じ場合、フォトダイオードの θ 軸方向が光源の方向へ向いていると判断し、 ϕ 軸方向の位置検出を行うラベル FMAIN へ移動する。
- H1 :** AD1 のほうが大きい場合に通るラベル。AD1 を K に、AD2 を KEI に保存し、 $KEI - TEMP = KEI$ の計算を行った後、SHANTEIX1 へ移動する。
- H2 :** AD2 のほうが大きい場合に通るラベル。AD2 を K に、AD1 を KEI に保存し、 $KEI - TEMP = KEI$ の計算を行った後、SHANTEIX2 へ移動する。
- SHANTEIX1 :** KEI と K の大小判定により AD1 と AD2 の差が許容範囲内かを判断するラベル。KEI と K が同じ、あるいは KEI が大きければ許容範囲内と判断し SH へ、K が大きければ許容範囲外と判断し H1KAI へ移動する。
- SHANTEIX2 :** KEI と K の大小判定により AD1 と AD2 の差が許容範囲内かを判断するラベル。KEI と K が同じ、あるいは KEI が大きければ許容範囲内と判断し SH へ、K が大きければ許容範囲外と判断し H2KAI へ移動する。
- SH :** AD1 と AD2 の差が許容範囲内の場合に通るラベル。差が許容範囲内であると、 θ 軸方向での位置検出が完了したとみなし、 ϕ 軸方向の位置検出を行うラベル FMAIN へ移動する。
- H1KAI :** AD1 のほうが大きく、AD1 と AD2 の差が許容範囲外の場合に通るラベル。

PORTD をクリヤし、DOUSA+1=DOUSA の計算を行った後、DOUSA を PORTD に代入する。そして、HKAZU が 0 であれば SHG へ、1 なら SHS へ移動する。

H2KAI : AD2 のほうが大きく、AD2 と AD1 の差が許容範囲外の場合に通るラベル。PORTD をクリヤし、DOUSA+1=DOUSA の計算を行った後、DOUSA を PORTD に代入する。そして、HKAZU が 0 であれば SHS へ、1 なら SHG へ移動する。

SHS : SSEITEN を呼び出した後、SAAA へ移動する。

SHG : SGYAKUTEN を呼び出した後、SAAA へ移動する。

SSEITEN : θ 軸方向の位置を制御するステッピングモーターを正転させるラベル。

SGYAKUTEN : θ 軸方向の位置を制御するステッピングモーターを逆転させるラベル。

SAAA : SHS, SHG から移動するラベル。このラベルを通るということは、まだフォトダイオードが光源の向きへ向いていない可能性があるので、もう一度 SMAIN へ移動する。

FMAIN : ϕ 軸方向についてのラベル。ADHEN3, ADHEN4 を通った後、FHANTEI へ移動する。

ADHEN3 : RA2 に入力した電圧を A/D 変換し、取得したデジタル値を AD3 に保存し、ADH3 に 1 を足す。

ADHEN4 : RA3 に入力した電圧を A/D 変換し、取得したデジタル値を AD3 に保存し、ADH3 に 1 を足す。

FHANTEI : ϕ 軸方向の素子間の入力電圧 AD3, AD4 の大小判定を行うラベル。AD3 が大きければ H3 へ、AD4 が大きければ H4 へ、同じ場合は FONAJI へ移動する。

FONAJI : AD3 と AD4 が同じである場合に通るラベル。 ϕ 軸方向の各素子への入力電圧が同じ場合、フォトダイオードの ϕ 軸方向が光源の方向へ向いていると判断し、 θ 軸方向の位置検出を行うラベル SMAIN へ移動する。

H3 : AD3 のほうが大きい場合に通るラベル。AD3 を K に、AD4 を KEI に保存

し、 $KEI-TEMP=KEI$ の計算を行った後、FHANTEIX3 へ移動する。

H4 : AD4 のほうが大きい場合に通るラベル。AD4 を K に、AD3 を KEI に保存し、 $KEI-TEMP=KEI$ の計算を行った後、FHANTEIX4 へ移動する。

FHANTEIX3 : KEI と K の大小判定により AD3 と AD4 の差が許容範囲内かを判断するラベル。KEI と K が同じ、あるいは KEI が大きければ許容範囲内と判断し SH へ、K が大きければ許容範囲外と判断し H1KAI へ移動する。

FHANTEIX4 : KEI と K の大小判定により AD3 と AD4 の差が許容範囲内かを判断するラベル。KEI と K が同じ、あるいは KEI が大きければ許容範囲内と判断し SH へ、K が大きければ許容範囲外と判断し H2KAI へ移動する。

FH : AD3 と AD4 の差が許容範囲内の場合に通るラベル。差が許容範囲内であると、 ϕ 軸方向での位置検出が完了したとみなし、 θ 軸方向の位置検出を行うラベル SMAIN へ移動する。

H3KAI : AD3 のほうが大きく、AD3 と AD4 の差が許容範囲外の場合に通るラベル。PORTD をクリヤし、 $DOUSA+1=DOUSA$ の計算を行った後、DOUSA を PORTD に代入し、FSEITEN を呼び出す。そして $KAZU+1=KAZU$ を行い、FAAA へ移動する。

H4KAI : AD4 のほうが大きく、AD4 と AD3 の差が許容範囲外の場合に通るラベル。PORTD をクリヤし、 $DOUSA+1=DOUSA$ の計算を行った後、DOUSA を PORTD に代入し、FGYAKUTEN を呼び出す。そして $KAZU+1=KAZU$ を行い、FAAA へ移動する。

FSEITEN : θ 軸方向の位置を制御するステッピングモーターを正転させるラベル。

FGYAKUTEN : θ 軸方向の位置を制御するステッピングモーターを逆転させるラベル。

FAAA : H3KAI, H4KAI から移動するラベル。このラベルを通るということは、まだフォトダイオードが光源の向きへ向いていない可能性があるため、もう一度 SMAIN へ移動する。

変数について

- KAZU : KAZU は ϕ 軸の位置を表す変数である。初期設定で FF とし、始動時は ϕ 軸方向で頂点にフォトダイオードを設置する必要がある。
- HKAZU : HKAZU は KAZU HIGH である。 θ 軸方向で回転の種類を選ぶ際に、判定の対象となる。
- DOUSA : 回転の動作を行った回数を示す変数。静止状態の安定度を測る実験を行った際に使用した。
- TEMP : 許容範囲の幅を決める変数

参考文献

- [1] 日本リモートセンシング研究会 編. 図解リモートセンシング. 社団法人日本測量協会
- [2] 後閑哲也 著. 電子工作のための PIC 活用ガイドブック. 技術評論者