

```

/*
 * File:   kadai10.c
 * Author: mono2kuri
 * 反射板の移動方向と距離により、(DM) と (SM) の回転を変化させる。tgSWが下側のとき、
 * (DM) と (SM) は動かない。tgSWが上側のとき、(DM) と (SM) は動く。(DM) も (SM)
 * も反射板の距離が遠のくにつれて、回転が速くなる。反射板の距離が10cmを越え15cm以内
 * のとき、(LED) = 赤、15cmを越え25cm以内のとき、(LED) = 緑、25cmを越え40cm以内
 * のとき、(LED) = 青となる。(LED) が赤色のとき、tcSW (黄) を押下すると、(DM) も
 * (SM) も停止し、(SP) から高音を1秒間出力する。その状態から復帰するには、tcSW
 * (青) を押下する。尚、距離を10cmより近づけることはしない。また、同様に40cmから離
 * すこともしない。
 * Created on 2019/06/13, 10:11
 */

```

```
#include <xc.h>
```

```

// クロック周波数指定
// ( __delay_ms(), __delay_us() )関数が必要としているため
#define    _XTAL_FREQ    1000000

```

```
//記号文字定数：入出力
```

```

#define TGSW    (RA3)    // tgSW : トグルスイッチ
#define TCSW_Y    (RA5)    // tsSW : タクトスイッチ(黄色)
#define TCSW_B    (RE1)    // tsSW : タクトスイッチ(青色)
#define BEEP_YES    1    // 音を鳴らす
#define BEEP_NO    0    // 音を鳴らさない
#define BEEP_HIGH    1    // 高音出力

```

```
// 記号文字定数：3色LED
```

```

#define RED    1    // 赤色
#define BLUE    2    // 青色
#define GREEN    3    // 緑色
#define LED_OFF    0    // 消灯

```

```
// 記号文字定数：DCモータ、ステップングモータ関係
```

```

#define CW    0    // 正転指示
#define CCW    1    // 逆転指示
#define NOWAIT    10    // 待ち時間指定なし
#define MT_GO    1    // 両モータ回転指示
#define MT_STOP    0    // 両モータ停止指示

```

```
// 外部参照変数の宣言
```

```
extern char Seg7Data[18];
```

```
// 外部参照関数の宣言
```

```

extern unsigned int adConv();
extern void LedEachColor( int selcolor );
extern void BeepBuzzer( int tone );
extern void dcMotorTon( int rotate );
extern void dcMotorToff( int rotate );

```

```
//関数の宣言
```

```

void kadai10(void);
void StepAndDCMtrStop(void);
void DCAndStepnMtrPhase14(void);
void DCAndStepnMtrPhase23(void);
void DCAndStepnMtrPhase32(void);
void DCAndStepnMtrPhase41(void);

```

```
/* 関数名 : kadai10()
```

```
* 引数 : なし
```

```

* 戻り値：なし
* 概要：
* 入力：RA1/AN1＝測距センサの入力、RA3＝tgSW、tcSW（黄）＝RA5、tcSW（青）＝RE1
* 出力：
* 3色LEDの点灯＝AN1のAD変換結果を距離に見立て、赤/緑/青を表示する。
* DCモータ、ステップモータの速度制御をしながら、回転させる。
*/

```

```

void kadai10(void)
{
    // ローカル変数の定義
    unsigned int iUpper;    // AD変換後の電圧値の1位と小数点第1位の値を格納する
    double fValue;         // AD変換後の電圧値
    int doRotate;          // モータ回転指示

    // pic16f874aの端子定義/初期化
    PORTA = 0x00;          // PORTA = All 0 = Low
    PORTE = 0x00;          // PORTE = All 0 = Low
    PORTB = 0x00;          // PORTB = All 0 = Low
    PORTC = 0x00;          // PORTC = All 0 = Low
    PORTD = 0x00;          // PORTD = All 0 = Low
    TRISA = 0xFF;          // PORTA = 全て入力(AnalogもDigitalも)
    TRISE = 0x07;          // PORTE = 全て入力(RE0?RE3のみの設定、AnalogもDigitalも)
    TRISB=0x00;           // PORTB を出力に設定
    TRISC=0x00;           // PORTC を出力に設定
    TRISD=0x00;           // PORTD を出力に設定
    ADCON1 = 0b00000110;  // PORTA/E = Digital I/O, not A/D inputs

    //処理本体
    doRotate = MT_GO;
    while(1) {
        unsigned int value = adConv();    // AD変換結果の取得
        fValue = (double)value * 5 / 1023; // AD変換結果を電圧値に換算し、
        if( 1.6 < fValue ) {              // およそ10cm?15cmとなったら、
            LedEachColor (RED);           // LED＝赤色に点灯し、
            if( TCSW_Y == 0 ) {           // その時に、tcSW(黄)を押下したら、
                StepAndDCMtrStop();       // 両モータを停止し、
                BeepBuzzer (BEEP_HIGH);    // ブザーを1秒間鳴らし、
                doRotate = MT_STOP;        // 両モータ停止指示をセットする。
            }
            if( TCSW_B == 0 ) doRotate = MT_GO; // 両モータ停止指示をリセットするのは、tcSW(青)を押下した時
        } else if( 1.1 < fValue && fValue <= 1.6 ) LedEachColor (GREEN); // 15cm?25cmなら、LED＝緑色に点灯
        else if( fValue <= 1.1 ) LedEachColor (BLUE); // 25cm?40cmなら、LED＝青色に点灯
        if( TGSW == 1 ) {                  // tgSWが下側だと、
            StepAndDCMtrStop();           // 両モータは停止状態にする..
        } else if ( TGSW == 0 && doRotate == MT_GO ) { // tgSWが上側で両モータの回転指示なら、
            value = 511 - (value & 511);    // DCモータ：デューティ比/遠のくと高速
            value = value / 10;
            iUpper = (unsigned int)(value / 10); // DCモータのデューティ元は変換結果から直接に
            DCAndStepnMtrPhase14(); // DCモータ+ステップモータ：時計回り(CW)回転指示
            switch( iUpper ) {              // DCモータのデューティを変化させる..
                case 5:
                    __delay_ms(9); // 回転時間/5ms
                    break;
                case 4:
                    __delay_ms(8); // 回転時間/8ms
                    break;
                case 3:
                    __delay_ms(7); // 表示時間/7ms
                    break;
                case 2:
                    __delay_ms(6); // 回転時間/6ms
                    break;
            }
        }
    }
}

```

```

        default: // case 1:
            __delay_ms(5); // 回転時間/5ms
            break;
    }
    DCAndStepnMtrPhase23(); // DCモータ+ステップングモータ: 時計回り(CW)回転指示
    __delay_ms(5); // 回転時間/5ms
    DCAndStepnMtrPhase32(); // DCモータ+ステップングモータ: 時計回り(CW)回転指示
    __delay_ms(5); // 回転時間/5ms
    DCAndStepnMtrPhase41(); // DCモータ+ステップングモータ: 時計回り(CW)回転指示
    __delay_ms(5); // 回転時間/5ms
} else {

}
}
return;
}

```

```

/* 関数名: StepAndDCMtrStop()
 * 引数: なし
 * 戻り値: なし
 * 概要: DC/ステップングモータ停止
 */

```

```

void StepAndDCMtrStop(void)
{
    RD0 = 0; // ステップングモータ停止: RD0?RD3へセット
    RD1 = 0;
    RD2 = 0;
    RD3 = 0;
    RD6 = 1; // DCモータ停止: RD6、RD7へセット
    RD7 = 1;
    RC0 = 1; // モータ駆動回路へセット: RC0のPGEでセット
    RC0 = 0;
    return;
}

```

```

/* 関数名: DCAndStepnMtrPhase14()
 * 引数: なし
 * 戻り値: なし
 * 概要: ステップングモータ回転フェーズ = 正転時/1、反転時/4
 */

```

```

void DCAndStepnMtrPhase14(void)
{
    RD0 = 1; // ステップングモータ回転#4: RD0?RD3へセット
    RD1 = 0;
    RD2 = 0;
    RD3 = 0;
    RD6 = 1; // DCモータ: Hi-Z状態で停止指示 = RD6、RD7
    RD7 = 0;
    RC0 = 1; // モータ駆動回路へセット: RC0のPGEでセット
    RC0 = 0;
    return;
}

```

```

/* 関数名: DCAndStepnMtrPhase23()
 * 引数: なし
 * 戻り値: なし
 * 概要: ステップングモータ回転フェーズ = 正転時/2、反転時/3
 */

```

```

void DCAndStepnMtrPhase23(void)
{

```

```
RD0 = 0;          // ステッピングモータ回転#4 : RD0?RD3へセット
RD1 = 1;
RD2 = 0;
RD3 = 0;
RD6 = 0;          // DCモータ : Hi-Z状態で停止指示 = RD6、RD7
RD7 = 0;
RC0 = 1;          // モータ駆動回路へセット : RC0のPGEでセット
RC0 = 0;
return;
}
```

```
/* 関数名 : StepMtrPhase32 ()
```

```
* 引数 : なし
```

```
* 戻り値 : なし
```

```
* 概要 : ステッピングモータ回転フェーズ = 正転時/3、反転時/2
```

```
*/
```

```
void DCAndStepnMtrPhase32 (void)
```

```
{
RD0 = 0;          // ステッピングモータ回転#4 : RD0?RD3へセット
RD1 = 0;
RD2 = 1;
RD3 = 0;
RD6 = 0;          // DCモータ : Hi-Z状態で停止指示 = RD6、RD7
RD7 = 0;
RC0 = 1;          // モータ駆動回路へセット : RC0のPGEでセット
RC0 = 0;
return;
}
```

```
/* 関数名 : DCAndStepnMtrPhase41 ()
```

```
* 引数 : なし
```

```
* 戻り値 : なし
```

```
* 概要 : ステッピングモータ回転フェーズ = 正転時/4、反転時/1
```

```
*/
```

```
void DCAndStepnMtrPhase41 (void)
```

```
{
RD0 = 0;          // ステッピングモータ回転#4 : RD0?RD3へセット
RD1 = 0;
RD2 = 0;
RD3 = 1;
RD6 = 0;          // DCモータ : Hi-Z状態で停止指示 = RD6、RD7
RD7 = 0;
RC0 = 1;          // モータ駆動回路へセット : RC0のPGEでセット
RC0 = 0;
return;
}
```