

```

/* ===== */
/* 第7回 熊本県高校生ものづくりコンテスト大会 電気系 電子回路組立部門 実施：2010年06月13日(日) */
/* ===== */

/** ヘッドファイルの定義 **/
#include "3048f.h" /* ヘッドファイル：CPU関連 */

/** 外部参照関数の定義 **/

/** 内部参照関数の定義 **/
void main(void); /* メイン関数 */
void InitializeI0(void); /* マイクン初期設定 */
void Wait1msec(void); /* タイマー(待ち時間：1msec)処理 */
void WaitMsec(int); /* タイマー(待ち時間：?msec)処理 */
int GetSW1(void); /* トグルスイッチ状態取得 */
int GetSW3(void); /* トグルスイッチ状態取得 */

/** 記号文字定数の定義(タイマー+割込処理用) **/
#define COUNT_START 1 /* カウント開始 */
#define COUNT_STOP 0 /* カウント停止 */

/** 記号文字定数の定義(入出力制御用) **/
#define LED_AL_OFF 0xff /* LED表示：全消灯 */
#define LED_AL_ON 0x00 /* LED表示：全点灯 */

#define OE_ENABLE 0 /* 許可(負論理) */
#define OE_DISABLE 1 /* 非許可(負論理) */

#define OE_7SEG PB.DR.BIT.B7 /* OE:IC1/Output Enable/7SEGMENT LED */
#define OE_LED PB.DR.BIT.B6 /* OE:IC2/Output Enable/LED */

#define SW1 P7.DR.BIT.B1 /* トグルスイッチ：押上時( )=Hi、押下時( )=Lo */
#define SW3 P7.DR.BIT.B5 /* トグルスイッチ：押上時( )=Lo、押下時( )=Hi */
#define SW_ON 0 /* スwitchの状態：Lo(電圧=0V)=オフ */
#define SW_OFF 1 /* スwitchの状態：Hi(電圧=5V)=オン */

/** 記号文字定数の定義(AD変換) **/
#define VREF10BIT 0.00488281 /* AD変換：電圧変換係数(10bit) = 基準5(V) ÷ 1024 */

/** 記号文字定数の定義(モーター制御) **/
#define MTR1 PB.DR.BIT.B3 /* モーター駆動信号1：TA7257.IN1(Pin1) */
#define MTR2 PB.DR.BIT.B1 /* モーター駆動信号2：TA7257.IN2(Pin2) */
#define CTR0 0 /* モーター制御：0 */
#define CTR1 1 /* モーター制御：1 */

/* ===== */
/* 関数名：main */
/* 引数：なし(void) */
/* 戻り値：なし(void) */
/* 内容：メインプログラムの実行 */
/* ===== */
void main(void)
{
    /* 変数の定義 */
    float fVofAD; /* AD変換の結果：電圧換算(浮動小数点値) */
    int iVofAD; /* AD変換の結果：電圧換算(整数値) */

    /* 処理 */
    InitializeI0(); /* マイクン初期化 */
    OE_7SEG = OE_DISABLE; /* 7SEG LEDの点灯許可=禁止(IC1.19pin:~OE) */
    OE_LED = OE_DISABLE; /* LEDの点灯許可=禁止(IC2.19pin:~OE) */
    for(;;){ /* 無限ループ(開始) */
        if( GetSW1() == SW_ON /* SW1とSW3の状態取得～両方がオン状態の確認： */
            && GetSW3() == SW_ON ){ /* 両方ともオンなら以下.. */

            /* ----- */
            /* ここにプログラムし、課題を完成させる。 */
            /* ----- */

```

```

    } else {
        MTR1 = CTRO;
        MTR2 = CTRO;
    }
}

/* SW1、SW3のいずれか、或いは両方がわなら以下.. */
/* ストップモード指示(IN1) */
/* ストップモード指示(IN2) */

/* 無限ループ(終端) */

/*
=====
関数名: InitializeI0
引数: なし(void)
戻り値: なし(void)
内容: マイコン初期値設定
=====
*/
void InitializeI0(void)
{
    /* 入出力端子の定義 */
    PA.DR.BYTE = 0; /* PA: データクリア */
    PA.DDR = 0xff; /* PA: 全ポートアウトプット */
    PA.DR.BYTE = LED_AL_OFF; /* PA: データセット(LEDオフ) */
    PB.DR.BYTE = 0; /* PB: データクリア */
    PB.DDR = 0xff; /* PB: 全ポートアウトプット */
    /* P7xは、8bitの入力専用ポート。入出力設定はない。 */

    /* ADコンバータの定義 */
    AD.DRD = 0; /* データレジスタモードをクリア */
    AD.CSR.BIT.ADF = 0; /* A/D変換終了フラグをクリア */
    AD.CSR.BIT.ADIE = 0; /* A/D変換終了割り込みを禁止 */
    AD.CSR.BIT.ADST = 0; /* A/D変換スタートフラグをクリア */
    AD.CSR.BIT.SCAN = 0; /* A/D変換は単一モード */
    AD.CSR.BIT.CKS = 1; /* 変換時間は、134テスト(高速) */
    AD.CSR.BIT.CH = 7; /* A/D変換は、AN7ポート(J1-2ピン)を使用 */

    /* タイマーの定義 */
    ITU1.TCR.BYTE = 0x23; /* タイマー1 設定 */
    ITU1.GRA = 0x0c00; /* タイマー1 1/100秒(0x7800)、1/1000秒(0x0c00) : 24.576MHz時 */
    ITU1.TSTR.BIT.STR1 = COUNT_START; /* タイマー1 カウント開始 */
}

/*
=====
関数名: ADConv
引数: なし(void)
戻り値: A/D変換値
内容: A/D変換の本体ルーチン
=====
*/
int ADConv(void)
{
    /* 引数変数の定義 */
    int ADdata; /* A/D変換値 */

    AD.CSR.BIT.ADST = 1; /* A/D変換開始 */
    while( !AD.CSR.BIT.ADF ); /* 変換終了待ち */
    ADdata = AD.DRD >> 6; /* 変換された値を変数に格納: 10ビット左詰 */
    AD.CSR.BIT.ADF = 0; /* A/D変換終了フラグをクリア */
    return( ADdata ); /* 戻り値(A/D変換値)のセット */
}

/*
=====
関数名: Wait1sec
引数: なし(void)
戻り値: なし(void)
内容: 待ち時間処理(1/1000秒)
=====
*/
void Wait1msec(void)
{
    /* 処理 */
    for(;;){ /* カウンタ確認(無限)ループ開始 */
        if( ITU1.TSR.BIT.IMFA != 0 ) break; /* 1/1000秒待つ */
    } /* カウンタ確認(無限)ループ終了 */
    ITU1.TSR.BIT.IMFA = 0; /* カウンタレジスタをクリアする */
}

/*
=====
関数名: WaitMsec
=====
*/

```

```

/* 引数：int msec 秒数の指定(単位：msec) */
/* 戻り値：なし(void) */
/* 内容：指定秒数の待ち時間処理 */
/* ===== */
void WaitMsec(int msec)
{
    /* 引数変数の定義 */
    int iCnt; /* カンタ */

    /* 処理 */
    if( msec < 0 ) return; /* 指定秒数が負の時は、処理しない */
    for( iCnt = 0 ; iCnt < msec ; iCnt++ ) Wait1msec(); /* 1msecを指定回数繰り返す */
}

/* ===== */
/* 関数名：GetSW1 */
/* 引数：なし(void) */
/* 戻り値：SW_OFF、SW_ON */
/* 内容：トグルスイッチ(SW1)の状態取得：レバ-を上向きに倒した時にo、下向きに倒した時にoとする。 */
/* ===== */
int GetSW1(void)
{
    return( SW1 ); /* トグルスイッチ状態の取得 */
}

/* ===== */
/* 関数名：GetSW3 */
/* 引数：なし(void) */
/* 戻り値：SW_OFF、SW_ON */
/* 内容：トグルスイッチ(SW1)の状態取得：レバ-を上向きに倒した時にo、下向きに倒した時にoとする。 */
/* ===== */
int GetSW3(void)
{
    return( SW3 ); /* トグルスイッチ状態の取得 */
}

/* ===== */
/* ===== EndOfProgram ===== */
/* ===== */

```