

```

/* ===== */
/* 第8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17日(金) */
/* ===== */
/* ルネサ製マイコンH8用基板(RY3048F-ONE TypeH：北斗電子製マイコンカー用CPUボード)制御プログラム用ヘッダファイル */
/* ケーブル接続：基板.CN1(LED/7セグメント点灯用)～H8側J2、基板.CN2(LED/7セグメント点灯制御+オプション制御用)～H8側J3 */
/* ===== */

/** 内部参照関数の定義 **/
void InitializeIO(void); /* マイコン初期設定 */
int ADConv(int); /* A/D変換ルーチン(入力チャンネル指定：単一モード) */
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 GetSW2(void); /* フッシュスイッチ状態取得 */
int GetSW3(void); /* フォトセンサー状態取得 */
void Led7Ctl(char, char); /* LED1～2表示制御関数：7セグメントLED */
void LedCtl(char); /* LED3～10表示制御関数：LED */

/** 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 FET_ENABLE 1 /* 許可(正論理) */
#define FET_DISABLE 0 /* 非許可(正論理) */

#define LED7SEG_ALL PA.DR.BYTE /* LED/7SegLED出力 */
#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 LED7SEG_2 PB.DR.BIT.B5 /* 7Segment LED 2 */
#define LED7SEG_1 PB.DR.BIT.B4 /* 7Segment LED 1 */
#define A 0x01 /* 7Segment LED : a 7Seg Port */
#define B 0x02 /* 7Segment LED : b LED PA */
#define C 0x04 /* 7Segment LED : c a 0 */
#define D 0x08 /* 7Segment LED : d f b 5 1 */
#define E 0x10 /* 7Segment LED : e g 6 */
#define F 0x20 /* 7Segment LED : f e c 4 2 */
#define G 0x40 /* 7Segment LED : g d 3 */

#define SW1 P7.DR.BIT.B3 /* (J1-6ピン)トグルスイッチ：押上時=5V、押下時=0V */
#define SW2 P7.DR.BIT.B5 /* (J1-4ピン)フッシュスイッチ：押下時=0V、解放時=5V */
#define SW3 P7.DR.BIT.B7 /* (J1-2ピン)フォトセンサー：透過時=0V、遮光時=5V */
#define SW_LO 0 /* スwitchの状態：電圧=0V */
#define SW_HI 1 /* スwitchの状態：電圧=5V */

/** 記号文字定数の定義 **/
#define True 1 /* フラグ制御：真 */
#define False 0 /* フラグ制御：偽 */
#define MAX_OF_WAIT 2400 /* 初期化最大待ち時間：2400 x 3msec = 7.2秒 */

/** 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 CTRO 0 /* モーター制御：0 */
#define CTR1 1 /* モーター制御：1 */
#define DIR_CLOCKWISE 1 /* モーター回転方向：正転 */
#define DIR_REVERSE -1 /* モーター回転方向：逆転 */
#define DIR_STOP 0 /* モーター回転方向：停止 */
#define DSP_CLOCKWISE 12 /* モーター回転方向：正転表示文字指定 c */
#define DSP_REVERSE 16 /* モーター回転方向：逆転表示文字指定 r */
#define DSP_STOP 17 /* モーター回転方向：停止表示文字指定 - */

/** 記号文字定数の定義(7segLEDの表示) **/
#define DSP_C 12 /* 表示文字指定 c */

```

```

#define DSP_D      13      /* 表示文字指定 d */
#define DSP_E      14      /* 表示文字指定 E */
#define DSP_F      15      /* 表示文字指定 F */
#define DSP_R      16      /* 表示文字指定 r */
#define DSP_O      18      /* 表示文字指定 o */

/** グローバル変数の定義 **/
const char _7segVal[19] = { ~ (A+B+C+D+E+F), /* 注：これがあるのでセクション「C」が必要になる。 */
                             ~ (B+C), /* 「0」表示 */
                             ~ (A+B+G+E+D), /* 「1」表示 */
                             ~ (A+B+G+C+D), /* 「2」表示 */
                             ~ (F+G+B+C), /* 「3」表示 */
                             ~ (A+F+G+C+D), /* 「4」表示 */
                             ~ (A+F+G+C+D+E), /* 「5」表示 */
                             ~ (A+B+C), /* 「6」表示 */
                             ~ (A+B+C+D+E+F+G), /* 「7」表示 */
                             ~ (A+B+C+D+F+G), /* 「8」表示 */
                             ~ (A+B+C+E+F+G), /* 「9」表示 */
                             ~ (C+D+E+F+G), /* 「A」表示 */
                             ~ (D+E+G), /* 「b」表示 */
                             ~ (B+C+D+E+G), /* 「c」表示 */
                             ~ (A+D+E+F+G), /* 「d」表示 */
                             ~ (A+E+F+G), /* 「E」表示 */
                             ~ (E+G), /* 「F」表示 */
                             ~ (G), /* 「r」表示 */
                             ~ (C+D+E+G) }; /* 「-」表示 */
                                           /* 「o」表示 */

/* ===== */
/* 関数名：Initialize0
/* 引数：なし(void)
/* 戻り値：なし(void)
/* 内容：マイコン初期値設定
/* PA=出力ポート(8bit)：LED+7セグメントLED点灯用
/* PB=出力ポート(8bit)：LED+7セグメントLED点灯制御用
/* P7=入力専用ポート(8bit)：AD変換(AN1)+DI(P77,P75,P73)用
/* AD変換(AN1)の設定：以下の通り。
/* AD.CSRは、ADコントロール/ステータスレジスタを設定するもの。次の様な構成となっている。
/* ADF(A/Dイベントフラグ)は、A/D変換が終了したことを示すフラグ。A/D変換が終了した時に「1」になる。
/* ADIE(A/Dインタラプトイネーブル)は、A/D変換の終了で割り込み処理を発生させるかどうかを設定する。今回は、割り込みを使用しないので「0」とする。
/* ADST(A/Dスタート)は、A/D変換の開始/停止を設定する。「0」で停止、「1」で開始する。今回は、停止から始めるので「0」に設定する。
/* SCAN(スキャンモード)は、スキャンモードとしてA/D変換をするのか、指定チャンネルを単一モードでA/D変換するのかを指定する。今回は、単一モードなので「0」に設定する。
/* CKS(クロックレクタ)は、A/D変換時間を設定する。「0」で266ステート、「1」で134ステートの時間になる。今回は、「1」に設定する。
/* CH(チャンネル)には、次の様な意味づけがある。
/* CH2 CH1 CH0 単一モード スキャンモード
/* 0 0 0 AN0 AN0
/* 0 0 1 AN1 AN0+AN1
/* 0 1 0 AN2 AN0+AN1+AN2
/* 0 1 1 AN3 AN0+AN1+AN2+AN3
/* 1 0 0 AN4 AN4
/* 1 0 1 AN5 AN4+AN5
/* 1 1 0 AN6 AN4+AN5+AN6
/* 1 1 1 AN7 AN4+AN5+AN6+AN7
/* | チャンネル選択
/* | グループ選択
/* タイマ(TU1)
/* TU1.TCR.BYTE = 0x23
/* | CCLR1|CCLR0|CKEG1|CKEG0|TPSC2|TPSC1|TPSC0|
/* | 0 1 1 →クロック周波数φ÷8(TCNT(タイマカウンタ)の基準となるクロック)
/* | 0 0 →カウントするエッジの選択：設定は、クロックの立ち上がり
/* | 0 1 →TCNTのクリアの仕方：カウント値の設定はGRAに書込む(GRAコンパリアマツとインプットキャプチャでクリア)
/* TU1.GRA = 0x7800
/* | どこ迄カウントしたらTCNTをクリアして、知らせてもらうかを設定する。
/* | 今回のTCNTのクロックはφ(24.576MHz)/8に設定したので、3.072MHzが基準クロックになる。
/* | 3.072MHzの周期なので、30720倍するとちょうど10ミリ秒(1/100秒)になるので、
/* | GRAを30720(16進数で7800h)に設定すると、カウント開始から1/100秒後にフラグが立つ。
/* TU.TSTR.BIT.STR1 = COUNT_START →TU1のカウント開始指示
/* 注：システムクロックφ=24.576MHz(RY3048Foneの場合：現物と北斗電子の資料から確認)
/* =====
void Initialize0(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スタート(高速)     */
AD. CSR. BIT. CH = 0;        /* A/D変換は、利用する時に指定       */

/* 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 */
ITU. TSTR. BIT. STR1 = COUNT_START; /* タイマー1 カウント開始           */
}

/* ===== */
/* 関数名 : ADConv          */
/* 引数 : int ch = AD変換ポートの指定(0~7) */
/* 戻り値 : A/D変換値      */
/* 内容 : A/D変換の本体ルーチン */
/* A/D変換関数の使い方は、AN1ポート(J1-8ピン)を使用する場合、ADConv(1)となる。 */
/* ===== */
int ADConv(int ch)
{
    /* ローカル変数の定義 */
    int ADdata;                /* A/D変換値                          */

    AD. CSR. BIT. CH = ch;     /* A/D変換は、AN1ポート(J1-8ピン)を使用 */
    AD. CSR. BIT. ADST = 1;    /* A/D変換開始                          */
    while( !AD. CSR. BIT. ADF ); /* 変換終了待ち                          */
    AD. CSR. BIT. ADF = 0;     /* A/D変換終了フラグをクリア           */
    switch(ch & 0x03) {
        case 0:                /* AN0/AN4の場合は、ADDR=ADDRA(10ビット左詰) */
            ADdata = AD. DRA >> 6;
            break;
        case 1:                /* AN1/AN5の場合は、ADDR=ADDRB(10ビット左詰) */
            ADdata = AD. DRB >> 6;
            break;
        case 2:                /* AN2/AN6の場合は、ADDR=ADDRC(10ビット左詰) */
            ADdata = AD. DRC >> 6;
            break;
        case 3:                /* AN3/AN7の場合は、ADDR=ADDRD(10ビット左詰) */
            ADdata = AD. DRD >> 6;
            break;
    }
    return( ADdata );         /* 戻り値(A/D変換値)のセット          */
}

/* ===== */
/* 関数名 : Wait1msec      */
/* 引数 : なし(void)       */
/* 戻り値 : なし(void)     */
/* 内容 : 待ち時間処理    */
/* 参考 : IMFA - TCNT = GRAの時、1になる */
/*        : TSR - Timer Status Register */
/* ===== */
void Wait1msec(void)
{
    /* 処理 */
    for(;;) {
        if( ITU1. TSR. BIT. IMFA != 0 ) break;
    }
    ITU1. TSR. BIT. IMFA = 0;
}

/* ===== */
/* 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)
{
    /* 0-加変数の定義 */
    int iCnt; /* カウンタ */

    /* 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 : ハブを上向きに倒した時に1、下向きに倒した時に0とする。 */
/* ===== */
int GetSW1(void)
{
    return( SW1 ); /* トグルスイッチ状態の取得 */
}

/* ===== */
/* 関数名 : GetSW2 */
/* 引 数 : なし(void) */
/* 戻り値 : SW_OFF、SW_ON */
/* 内 容 : プッシュスイッチ(SW2)の状態取得 : 押した時に1、放した時に0とする。 */
/* ===== */
int GetSW2(void)
{
    return( SW2 ); /* プッシュ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 : フォトセンサー(SW3)の状態取得 : 透過状態で1、遮光状態で0とする。 */
/* ===== */
int GetSW3(void)
{
    return( SW3 ); /* フォトセンサー状態の取得 */
}

/* ===== */
/* 関数名 : Led7Ctl */
/* 引 数 : char led1=LED1の表示データ(点灯=0、消灯=1)
          char led2=LED2の表示データ(点灯=0、消灯=1)
/* 戻り値 : なし(void)
/* 内 容 : LED1~2表示制御関数(7セグメントLED)
          表示制御時間は、凡そ2msecである。表示しながらタイミングを計る時は要注意。
/* ===== */
void Led7Ctl(char led1, char led2)
{
    OE_7SEG = OE_ENABLE; /* 7seg LEDの点灯許可=許可(IC1. 19pin:~OE) */
    OE_LED = OE_DISABLE; /* LEDの点灯許可=禁止(IC2. 19pin:~OE) */
    LED7SEG_ALL = led1; /* 7segLED(LED1)点灯 : 10の位 */
    LED7SEG_1 = FET_ENABLE; /* 7segLED1点灯許可 */
    LED7SEG_2 = FET_DISABLE; /* 7segLED2点灯禁止 */
    Wait1msec(); /* 7segLED1点灯保持(1msec) */
    LED7SEG_ALL = led2; /* 7segLED(LED2)点灯 : 1の位 */
    LED7SEG_1 = FET_DISABLE; /* 7segLED1点灯禁止 */
    LED7SEG_2 = FET_ENABLE; /* 7segLED2点灯許可 */
    Wait1msec(); /* 7segLED2点灯保持(1msec) */
}

/* ===== */
/* 関数名 : LedCtl */
/* 引 数 : char leds=LED3~10の表示データ(点灯=0、消灯=1)
/* 戻り値 : なし(void)

```

```
/* 内容 : LED3~10表示制御関数(LED) */
/*      表示制御時間は、ない。 */
/* ===== */
void LedCtl(char leds)
{
    OE_7SEG = OE_DISABLE; /* 7SEG LEDの点灯許可=禁止(IC1. 19pin:~OE) */
    OE_LED = OE_ENABLE; /* LEDの点灯許可=許可(IC2. 19pin:~OE) */
    LED7SEG_ALL = leds; /* LED3~10の点灯 */
}
```