

RX62T_ADコンバータ関連_ADC.c

```
/*=====*/
/*          インクルードファイル          */
/*=====*/
#include <machine.h>

#include "../common/iodef.h" /* RX62T I/O定義ファイル */
#include "../common/vals_typedef.h" /* 変数型定義ヘッダファイル */

/* 制御プログラム関係          */
#include "fpmsm_gwork.h" /* グローバル変数定義ファイル */

#ifdef AC_SYSTEM
#include "fpmsm_user_ac.h" /* AC入力対応 */
#endif

#ifdef DC_SYSTEM
#include "fpmsm_user_dc.h" /* DC入力対応 */
#endif

/*=====*/
/* S12ADA0初期化関数          */
/* 引数： なし          */
/* 戻り値： 0:正常設定 1:設定異常終了          */
/*=====*/
/* 備考          */
/* 使用ADチャネル          */
/*       : P44/AN000          */
/*       : P45/AN001          */
/*       : P46/AN002          */
/*       : P47/AN003          */
/* 変換トリガ          */
/*       TRG4AN, TRG4BN          */
/* シングルスキャン          */
/* 本関数内で、1度だけ、自己診断を行う          */
/* 誤差を±10%までOKとする (2048±204)          */
/* 割り込み発生なし          */
/*=====*/

u1 S12ada0_init()
{
    u2 u2_temp_1;
    u1 u1_error;

    u1_error = 0;
}
```



```

/* モジュールストップ解除 */
SYSTEM.MSTPCRA.BIT.MSTPA24 = 0;
SYSTEM.MSTPCRA.BIT.MSTPA17 = 0;

/* AD停止 */
S12AD0.ADCSR.BIT.ADST = 0;
S12AD0.ADCSR.BIT.TRGE = 0;

/*==== 自己診断設定 =====*/

S12AD0.ADCSR.BIT.CKS = 3;
S12AD0.ADCSR.BIT.ADCS = 0;
S12AD0.ADANS.BIT.CH = 0;

S12AD0.ADCER.BIT.DIAGVAL = 2;
S12AD0.ADCER.BIT.DIAGLD = 1;
S12AD0.ADCER.BIT.DIAGM = 1;

/*==== 自己診断開始 =====*/
S12AD0.ADCSR.BIT.ADST = 1;
while ( 1 == S12AD0.ADCSR.BIT.ADST )
{
    ;;
}
nop();
nop();
u2_temp_1 = S12AD0.ADRD.BIT.RIGHT.DATA;
if ( ( u2_temp_1 > 2252 ) || ( u2_temp_1 < 1844 ) ) {
    u1_error = 1;
}

/* AD停止 */
S12AD0.ADCSR.BIT.ADST = 0;
S12AD0.ADCSR.BIT.TRGE = 0;

/* ADモード設定 */
S12AD0.ADCSR.BIT.CKS = 3;
S12AD0.ADCSR.BIT.ADCS = 1;
S12AD0.ADANS.BIT.CH = 3;

// S12AD0.ADCER.BIT.SHBYP = 0;
S12AD0.ADCER.BIT.SHBYP = 1;

/* MSTPA24 (S12ADA制御部)クロック供給開始 */
/* MSTPA17 (S12ADA0)クロック供給開始 */

/* トリガによる変換禁止 */

/* ADCLK = PCLK */
/* シングルモード */
/* AN000 */

/* 1/2VREFで自己診断 */
/* 自己診断電圧はローテーションしない */
/* 自己診断実施 */

/* トリガによる変換禁止 */

/* ADCLK = PCLK */
/* ワンサイクルスキャンモード */
/* AN000~AN003 */

/* S/H使用 */
/* S/H使用しない */

```



RX62T_ADコンバータ関連_ADC.c

```

S12AD0.ADCER.BIT.ADPRC          = 0;          /* 12Bit */
S12AD0.ADCER.BIT.DIAGVAL        = 1;          /* 0Vで自己診断(0は設定禁止?) */
S12AD0.ADCER.BIT.DIAGLD        = 1;          /* 自己診断電圧はローテーションしない */
S12AD0.ADCER.BIT.DIAGM         = 0;          /* 自己診断しない */
S12AD0.ADCER.BIT.ADRFMT        = 0;          /* 右詰め */
// S12AD0.ADCER.BIT.ACE          = 1;          /* 読み出しによるADDRレジスタクリア許可 */

/* サンプリング時間設定 */
S12AD0.ADSSTR                    = 0x14;        /* リセット値上書き */

/* トリガ選択 */
S12AD0.ADSTRGR.BIT.ADSTRSO      = 0x0B;      /* TRG4ANorTRG4BNを選択 */
S12AD0.ADCSR.BIT.TRGE          = 1;          /* トリガによる変換許可 */

// For デバック
// /* ---- 割り込み設定 ---- */
// S12AD0.ADCSR.BIT.ADIE          = 1;
// ICU.IER[12].BIT.IEN6          = 0;          /* 割り込み禁止 IEROC.IEN6=0 */
// ICU.IR[102].BIT.IR            = 0;          /* 割り込みフラグクリア IR102=0 */
// ICU.IPR[72].BIT.IPR           = 15;        /* 割り込み優先レベルIPR48 */
// ICU.IER[12].BIT.IEN6          = 1;          /* 割り込み許可 IEROC.IEN6=1 */

return(u1_error);
}

```

```

/*-----*/
/* S12ADA1初期化関数 */
/* 引数 : なし */
/* 戻り値 : 0:正常設定 1:設定異常終了 */
/*-----*/
/* 備考 */
/* 使用ADチャネル */
/* : P44/AN100 */
/* : P45/AN101 */
/* : P46/AN102 */
/* : P47/AN103 */
/* 変換トリガ */
/* TRG4AN, TRG4BN */
/* シングルスキャン */
/* 本関数内で、1度だけ、自己診断を行う */
/* 誤差を±10%までOKとする (2048±204) */
/* 割り込み発生なし */

```



```

/*=====*/
u1 S12ada1_init()
{
    u2 u2_temp_1;
    u1 u1_error;

    u1_error = 0;

    /* モジュールストップ解除 */
    SYSTEM.MSTPCRA.BIT.MSTPA24 = 0; /* MSTPA24 (S12ADA制御部)クロック供給開始 */
    SYSTEM.MSTPCRA.BIT.MSTPA16 = 0; /* MSTPA16 (S12ADA1)クロック供給開始 */

    /* AD停止 */
    S12AD1.ADCSR.BIT.ADST = 0;
    S12AD1.ADCSR.BIT.TRGE = 0; /* トリガによる変換禁止 */

    /*===== 自己診断設定 =====*/

    S12AD1.ADCSR.BIT.CKS = 3; /* ADCLK = PCLK */
    S12AD1.ADCSR.BIT.ADCS = 0; /* シングルモード */
    S12AD1.ADANS.BIT.CH = 0; /* AN100 */

    S12AD1.ADCER.BIT.DIAGVAL = 2; /* 1/2VREFで自己診断 */
    S12AD1.ADCER.BIT.DIAGLD = 1; /* 自己診断電圧はローテーションしない */
    S12AD1.ADCER.BIT.DIAGM = 1; /* 自己診断実施 */

    /*===== 自己診断開始 =====*/
    S12AD1.ADCSR.BIT.ADST = 1;
    while ( 1 == S12AD1.ADCSR.BIT.ADST )
    {
        ;;
    }
    nop();
    nop();
    u2_temp_1 = S12AD1.ADRD.BIT.RIGHT.DATA;
    if ( ( u2_temp_1 > 2252 ) || ( u2_temp_1 < 1844 ) ){
        u1_error = 1;
    }

    /* AD停止 */
    S12AD1.ADCSR.BIT.ADST = 0;
    S12AD1.ADCSR.BIT.TRGE = 0; /* トリガによる変換禁止 */
}

```



RX62T_ADコンバータ関連_ADC.c

```

/* ADモード設定 */
S12AD1.ADCSR.BIT.CKS = 3; /* ADCLK = PCLK */
S12AD1.ADCSR.BIT.ADCS = 1; /* ワンサイクルスキャンモード */
S12AD1.ADANS.BIT.CH = 3; /* AN100~AN103 */

// S12AD1.ADCGER.BIT.SHBYP = 0; /* S/H使用 */
S12AD1.ADCGER.BIT.SHBYP = 1; /* S/H使用しない */

S12AD1.ADCGER.BIT.ADPRC = 0; /* 12Bit */
S12AD1.ADCGER.BIT.DIAGVAL = 1; /* 0Vで自己診断(0は設定禁止?) */
S12AD1.ADCGER.BIT.DIAGLD = 1; /* 自己診断電圧はローテーションしない */
S12AD1.ADCGER.BIT.DIAGM = 0; /* 自己診断しない */
S12AD1.ADCGER.BIT.ADRFMT = 0; /* 右詰め */

/* サンプリング時間設定 */
S12AD1.ADSSTR = 0x14; /* リセット値上書き */

/* トリガ選択 */
S12AD1.ADSTRGR.BIT.ADSTRSO = 0x0B; /* TRG4ANorTRG4BNを選択 */
S12AD1.ADCSR.BIT.TRGE = 1; /* トリガによる変換許可 */

// For デバック
// /* ---- 割り込み設定 ---- */
// S12AD1.ADCSR.BIT.ADIE = 1;
// ICU.IER[12].BIT.IEN7 = 0; /* 割り込み禁止 IEROC.IEN7=0 */
// ICU.IR[103].BIT.IR = 0; /* 割り込みフラグクリア IR=0 */
// ICU.IPR[72].BIT.IPR = 15; /* 割り込み優先レベルIPR48 */
// ICU.IER[12].BIT.IEN7 = 1; /* 割り込み許可 IEROC.IEN7=1 */

return(u1_error);
}

```

```

/*-----*/
/* ADA初期化関数 */
/* 引数 : なし */
/* 戻り値 : 0:正常設定 1:設定異常終了 */
/*-----*/
/* 備考 */
/* 使用ADチャネル */
/* : P60/AN0 */
/* : P61/AN1 */
/* : P62/AN2 */

```



RX62T_ADコンバータ関連_ADC.c

```
/*          : P63/AN3          */
/*      変換トリガ          */
/*          TRG7AN          */
/*      本関数内で、1度だけ、自己診断を行う          */
/*      誤差を±10%までOKとする (512±51)          */
/*      割り込み発生なし          */
/*      3シャント用途          */
/*=====*/
```



```
u1 dpada0_init()
{
    u2 u2_temp_1;
    u1 u1_error;

    u1_error = 0;

    /* モジュールストップ解除 */
    SYSTEM.MSTPCRA.BIT.MSTPA23 = 0;

    /* AD停止 */
    ADO.ADCSR.BIT.ADST = 0;

    /*===== 自己診断設定 =====*/
    ADO.ADDIAGR.BIT.DIAG = 2;
    ADO.ADCR.BIT.CKS = 3;
    ADO.ADCR.BIT.MODE = 0;
    ADO.ADCSR.BIT.CH = 0x00;

    /* 1/2VREFで自己診断 */
    /* ADCLK = PCLK */
    /* シングルモード */
    /* ANO */

    /*===== 自己診断開始 =====*/
    ADO.ADCSR.BIT.ADST = 1;
    while ( 1 == ADO.ADCSR.BIT.ADST )
    {
        ;;
    }
    nop();
    nop();
    u2_temp_1 = ADO.ADDRA;

    if ( ( u2_temp_1 > 562 ) || (u2_temp_1 < 461) ){
        u1_error = 1;
    }

    /* AD停止 */
    ADO.ADCSR.BIT.ADST = 0;
}

/* MSTPA23 (ADA) クロック供給開始 */
```

RX62T_ADコンバータ関連_ADC.c

```

ADO.ADDIAGR.BIT.DIAG      = 0;          /* 通常動作（自己診断しない）      */
/* ADモード設定          */
ADO.ADCR.BIT.CKS         = 3;          /* ADCLK = PCLK                      */
ADO.ADCR.BIT.MODE        = 3;          /* 1サイクルスキャンモード          */

/* 変換チャンネル選択     */
ADO.ADCSR.BIT.CH         = 0x07;       /* AN0~AN7                            */

/* AD精度                 */
ADO.ADDPR.BYTE           = 0x00;       /* 10Bit/右詰                          */

/* サンプリング時間設定   */
ADO.ADSSTR               = 0x19;       /* リセット値上書き                    */

/* トリガ選択             */
// ADO.ADSTRGR.BYTE       = 0x0D;       /* TR7ANを選択                          */
ADO.ADSTRGR.BYTE         = 0x0B;       /* TRG4ANorTRG4BNを選択                */

// For debug
// /* ----- 割り込み設定 ----- */
// ADO.ADCSR.BIT.ADIE     = 1;
// ICU.IER[12].BIT.IEN2   = 0;
// ICU.IR[98].BIT.IR      = 0;
// ICU.IPR[68].BIT.IPR    = 15;
// ICU.IER[12].BIT.IEN2   = 1;
/* 割り込み禁止 IEROC.IEN2=0 */
/* 割り込みフラグクリア IR=0 */
/* 割り込み優先レベルIPR44 */
/* 割り込み許可 IEROC.IEN2=1 */

return(u1_error);
}

```

