```
/***************************************************************************
    RTM_MSP430(for MSP430F2013) C-Compiler Version

        June     27,1998    Kenji Arai
        January   1,1999    Change starting method for Cyclic task
        November  4,2001    Start porting to H8/3664F
        July     30,2004    Add user timer
        August    5,2006    Start porting to MSP430F2013
        August    6,2006    looks like running!

    Copyright (C) 1998,'99,2000,'01,'04,'06 Kenji Arai/JH1PJL
    All rights reserved. Permission is granted to use, modify,or redistribute
    this software so long as it is not sold or exploited for profit.

    THIS SOFTWARE IS PROVIDED AS IS AND WITHOUT WARRANTY OF ANY KIND,
    EITHER EXPRESSED OR IMPLIED.
***************************************************************************/
#include "rtm_msp430.h"

#define RTM_ON      1
#define RTM_OFF     0

/*****  Extern function prototype **************/
extern void ed0_main( void );
extern void ed1_main( void );
extern void ed2_main( void );
extern void ed3_main( void );
extern void cy0_main( void );
extern void cy1_main( void );
extern void cy2_main( void );
extern void cy3_main( void );
extern void cy4_main( void );
extern void cy5_main( void );
extern void cy6_main( void );
extern void cy7_main( void );

/*****  RAM assign  ****************************/
unsigned int
    tim_usr0,   /* user timer no.0 */
    tim_usr1,   /* user timer no.1 */
    tim_usr2,   /* user timer no.2 */
    tim_usr3;   /* user timer no.3 */

static unsigned char
    tim_cyc0,   /* cyclic task no.0 period timer */
    tim_cyc1,   /* cyclic task no.1 period timer */
    tim_cyc2,   /* cyclic task no.2 period timer */
    tim_cyc3,   /* cyclic task no.3 period timer */
    tim_cyc4,   /* cyclic task no.4 period timer */
    tim_cyc5,   /* cyclic task no.5 period timer */
    tim_cyc6,   /* cyclic task no.6 period timer */
    tim_cyc7;   /* cyclic task no.7 period timer */

struct {
    unsigned char ed0_req:  1;  /* event driven task no.0 request flag */
    unsigned char ed1_req:  1;  /* event driven task no.1 request flag */
    unsigned char ed2_req:  1;  /* event driven task no.2 request flag */
    unsigned char ed3_req:  1;  /* event driven task no.3 request flag */
    unsigned char not_used: 3;
    unsigned char cyc_req:  1;  /* Cyclic task check flag */
}status;
```

```
static union {
    unsigned char request;      /* cyclic task request flag */
    struct {
        unsigned char cy0:  1;
        unsigned char cy1:  1;
        unsigned char cy2:  1;
        unsigned char cy3:  1;
        unsigned char cy4:  1;
        unsigned char cy5:  1;
        unsigned char cy6:  1;
        unsigned char cy7:  1;
    }req;
}r;

/*****  External data *************************/
extern const unsigned char cyclic_period[8];
extern const unsigned char cyclic_init_period[8];

/*********************** Basic routine ************************************/
void rtm_core( void )
{
    /* event driven task dispatch */
    if( status.ed0_req ){
        /* event driven task no.0 check and go */
        status.ed0_req = RTM_OFF;
        ed0_main();
    }else if( status.ed1_req ){
        /* event driven task no.1 check and go */
        status.ed1_req = RTM_OFF;
        ed1_main();
    }else if( status.ed2_req ){
        /* event driven task no.2 check and go */
        status.ed2_req = RTM_OFF;
        ed2_main();
    }else if( status.ed3_req ){
        /* event driven task no.3 check and go */
        status.ed3_req = RTM_OFF;
        ed3_main();
    }
    if( status.cyc_req != 0 ){
        status.cyc_req = 0;
        rtm_msp430();
        if( r.request != 0 ){
            /* cyclic task dispatch */
            if( r.req.cy0 ){
                /* cyclic task no.0 check and go */
                r.req.cy0 = RTM_OFF;
                cy0_main();
            }else if( r.req.cy1 ){
                /* cyclic task no.1 check and go */
                r.req.cy1 = RTM_OFF;
                cy1_main();
            }else if( r.req.cy2 ){
                /* cyclic task no.2 check and go */
                r.req.cy2 = RTM_OFF;
                cy2_main();
            }else if( r.req.cy3 ){
                /* cyclic task no.3 check and go */
                r.req.cy3 = RTM_OFF;
                cy3_main();
            }else if( r.req.cy4 ){
                /* cyclic task no.4 check and go */
```

```
                r.req.cy4 = RTM_OFF;
                cy4_main();
            }else if( r.req.cy5 ){
                /* cyclic task no.5 check and go */
                r.req.cy5 = RTM_OFF;
                cy5_main();
            }else if( r.req.cy6 ){
                /* cyclic task no.6 check and go */
                r.req.cy6 = RTM_OFF;
                cy6_main();
            }else if( r.req.cy7 ){
                /* cyclic task no.7 check and go */
                r.req.cy7 = RTM_OFF;
                cy7_main();
            }
        }
    }
}

/*********************** Interrupt routine **********************************/
void rtm_msp430( void )
{
    if( !--tim_cyc0 ){
        tim_cyc0 = cyclic_period[0];
        r.req.cy0  = RTM_ON;
    }
    if( !--tim_cyc1 ){
        tim_cyc1 = cyclic_period[1];
        r.req.cy1  = RTM_ON;
    }
    if( !--tim_cyc2 ){
        tim_cyc2 = cyclic_period[2];
        r.req.cy2  = RTM_ON;
    }
    if( !--tim_cyc3 ){
        tim_cyc3 = cyclic_period[3];
        r.req.cy3 = RTM_ON;
    }
    if( !--tim_cyc4 ){
        tim_cyc4 = cyclic_period[4];
        r.req.cy4  = RTM_ON;
    }
    if( !--tim_cyc5 ){
        tim_cyc5 = cyclic_period[5];
        r.req.cy5  = RTM_ON;
    }
    if( !--tim_cyc6 ){
        tim_cyc6 = cyclic_period[6];
        r.req.cy6 =  RTM_ON;
    }
    if( !--tim_cyc7 ){
        tim_cyc7 = cyclic_period[7];
        r.req.cy7 =  RTM_ON;
    }
    if( tim_usr0 )
        --tim_usr0;
    if( tim_usr1 )
        --tim_usr1;
    if( tim_usr2 )
        --tim_usr2;
    if( tim_usr3 )
        --tim_usr3;
```

```
}

/************************ Event Driven Task request ***************************/
void reqed0()
{
    status.ed0_req = RTM_ON;
}

void reqed1()
{
    status.ed1_req = RTM_ON;
}

void reqed2()
{
    status.ed2_req = RTM_ON;
}

void reqed3()
{
    status.ed3_req = RTM_ON;
}

void req_cyctsk()
{
    status.cyc_req = 1;
}

/************************ Initialize routine ***********************************/
void rtm_msp430_init()
{
    tim_cyc0 = cyclic_init_period[0];
    tim_cyc1 = cyclic_init_period[1];
    tim_cyc2 = cyclic_init_period[2];
    tim_cyc3 = cyclic_init_period[3];
    tim_cyc4 = cyclic_init_period[4];
    tim_cyc5 = cyclic_init_period[5];
    tim_cyc6 = cyclic_init_period[6];
    tim_cyc7 = cyclic_init_period[7];

    status.ed0_req = RTM_OFF;
    status.ed1_req = RTM_OFF;
    status.ed2_req = RTM_OFF;
    status.ed3_req = RTM_OFF;
    status.cyc_req = 0;

    r.request = 0;

    tim_usr0 = 0;       /* idle */
    tim_usr1 = 0;       /* idle */
    tim_usr2 = 0;       /* idle */
    tim_usr3 = 0;       /* idle */
}
```