

```

/*****
Module
  SCI_Receive.c

Revision
  1.0.1

Description
  This is a template file for implementing a simple service under the
  Gen2 Events and Services Framework.

Notes

History
When          Who          What/Why
-----
01/16/12 09:58 jec          began conversion from TemplateFSM.c
*****/
/*----- Include Files -----*/
/* include header files for this state machine as well as any machines at the
   next lower level in the hierarchy that are sub-machines to this machine
*/
#include "ES_Configure.h"
#include "ES_Framework.h"
#include "Ultrason.h"
#include "ES_Timers.h"
#include <stdio.h>

#include <hidef.h>                /* common defines and macros */
#include <mc9s12e128.h>           /* derivative information */
#include <S12e128bits.h>         /* bit definitions */
#include <Bin_Const.h>           /* macros to allow specifying binary constants */
#include <termio.h>              /* to get prototype fo kbhit() */
// #include <s12vec.h>            /* E128 interrupt vectors */
#include "s12evec.h"

#define HighTime 20
#define LowTime 1

/*----- Module Defines -----*/

/*----- Module Functions -----*/
/* prototypes for private functions for this service. They should be functions
   relevant to the behavior of this service
*/

/*----- Module Variables -----*/
// with the introduction of Gen2, we need a module level Priority variable
static uint8_t MyPriority;
//static ES_Event PostEvent;
static unsigned int uPeriod;

/*----- Module Code -----*/
/*****
Function
  Init_SCI_Receive

Parameters

```

uint8\_t : the priority of this service

#### Returns

boolean, False if error in initialization, True otherwise

#### Description

Saves away the priority, and does any other required initialization for this service

#### Notes

#### Author

J. Edward Carryer, 01/16/12, 10:00

\*\*\*\*\*/

```
bool Init_Ultrason ( uint8_t Priority )
```

```
{
```

```
    ES_Event ThisEvent;
```

```
    MyPriority = Priority;
```

```
    TIMO_TSCR1 = _S12_TEN;
```

```
    TIMO_TSCR2 = _S12_PR2 | _S12_PR1 | _S12_PRO;
```

```
    TIMO_TIOS &= ~_S12_IOS4;
```

```
    TIMO_TCTL3 |= _S12_EDG4B;
```

```
    TIMO_TCTL3 |= _S12_EDG4A;
```

```
    TIMO_TFLG1 |= _S12_C4F;
```

```
    TIMO_TIE |= _S12_C4I;
```

```
    EnableInterrupts;
```

```
    //DDRT = 0x00;
```

```
    //DDRT = (DDRT | BIT4HI);    //Set T4 to be output
```

```
    //DDRT_DDRT7 = 1;    //T7 as an output
```

```
    PTT = 0x00;
```

```
    PTT = (PTT | BIT4HI);    // Raise the T4 to start triggering ultrasonic
```

```
    ES_Timer_SetTimer (UI_HiTime_TIMER, HighTime);
```

```
    ES_Timer_StartTimer(UI_HiTime_TIMER);
```

```
    //printf("%x\n\r", PTT);
```

```
    //printf("Ultrason  init!");
```

```
    ThisEvent.EventType = ES_INIT;
```

```
    if (ES_PostToService( MyPriority, ThisEvent) == true)
```

```
    {
```

```
        return true;
```

```
    }
```

```
    else
```

```
    {
```

```
        return false;
```

```
    }
```

```
}
```

```
*****
```

Function

Post\_SCI\_Receive

Parameters

EF\_Event ThisEvent ,the event to post to the queue

Returns

boolean False if the Enqueue operation failed, True otherwise

Description

Posts an event to this state machine's queue

Notes

Author

J. Edward Carryer, 10/23/11, 19:25

\*\*\*\*\*/

**bool** Post\_Ultrason( ES\_Event ThisEvent )

```
{  
    return ES_PostToService( MyPriority, ThisEvent);  
}
```

\*\*\*\*\*/

Function

Run\_SCI\_Receive

Parameters

ES\_Event : the event to process

Returns

ES\_Event, ES\_NO\_EVENT if no error ES\_ERROR otherwise

Description

add your description here

Notes

Author

J. Edward Carryer, 01/15/12, 15:23

\*\*\*\*\*/

ES\_Event Run\_Ultrason( ES\_Event ThisEvent )

```
{  
    ES_Event ReturnEvent;  
    ReturnEvent.EventType = ES_NO_EVENT; // assume no errors  
  
    if ((ThisEvent.EventType == ES_TIMEOUT) && (ThisEvent.EventParam == UI_HiTime_TIMER))  
    {  
        PTT = (PTT & BIT4LO);  
        //printf("LowT: %x\n\r", PTT);  
  
        ES_Timer_SetTimer (UI_LowTime_TIMER, LowTime);  
        ES_Timer_StartTimer(UI_LowTime_TIMER);  
    }  
  
    else if ((ThisEvent.EventType == ES_TIMEOUT) && (ThisEvent.EventParam == UI_LowTime_TIMER))  
    {  
        PTT = (PTT | BIT4HI);  
        //printf("HiT: %x\n\r", PTT);  
  
        ES_Timer_SetTimer (UI_HiTime_TIMER, HighTime);  
        ES_Timer_StartTimer(UI_HiTime_TIMER);  
    }  
}
```

```
    return ReturnEvent;
}
```

```
unsigned int Query_Ultrason ( void )
{
    return(uPeriod);
}
```

```
void interrupt _Vec_tim0ch4 Ultrasonic(void)
{
    static unsigned int LastEdge;
    //static int Distance;
    uPeriod = TIMO_TC4 - LastEdge;
    //printf("up:%i\n\r", uPeriod);
    LastEdge = TIMO_TC4;

    /*
    PTT_PTT7 ^= 1; // Toggle a pin

    if (uPeriod < 1000)
    {
        //printf("%i\n\r", uPeriod);

        Distance = uPeriod / 11;
        if (Distance < 10)
        {
            //printf("!C!\n\r");
        }
    }
    */

    TIMO_TFLG1 |= _S12_C4F; // CLEAR IC4 FLAG
}
```

```
/*-----
private functions
-----*/
/*unsigned char Receive_Data(void)
{

    return;

}

*/

/*----- Footnotes -----*/
/*----- End of file -----*/
```

