

```

/*****
Module
    LEDWriteService.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
*/
// the common headers for C99 types
#include <stdint.h>
#include <stdbool.h>

// the headers to access the GPIO subsystem
#include "inc/hw_gpio.h"
#include "inc/hw_types.h"
#include "inc/hw_pwm.h"
#include "inc/hw_memmap.h"
#include "driverlib/sysctl.h"
#include "inc/hw_sysctl.h"
#include "ES_Port.h"

// the headers to access the TivaWare Library
#include "driverlib/sysctl.h"
#include "driverlib/pin_map.h"
#include "driverlib/gpio.h"
#include "driverlib/timer.h"
#include "driverlib/interrupt.h"

#include "BITDEFS.H"

#include "ES_Configure.h"
#include "ES_Framework.h"
#include "LEDWriteService.h"
#include "PWM16Tiva.h"
#include "ShiftRegisterWrite.h"

/*----- 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 uint16_t WATER_LO = BIT0LO & BIT1LO & BIT2LO;
static uint16_t LIGHT_LO = BIT3LO & BIT4LO & BIT5LO;
static uint16_t HEALTH_LO = 0x00FF;

```

```

static uint16_t ALL_LO = 0x0000;
static uint16_t ALL_HI = 0xFFFF;

/*----- Module Code -----*/
/*****
Function
    InitLEDWriteService

Parameters
    uint8_t : the priority of this service

Returns
    bool, 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 InitLEDWriteService(uint8_t Priority)
{
    ES_Event_t ThisEvent;
    MyPriority = Priority;
    /*****
    in here you write your initialization code
    *****/
    // post the initial transition event
    ThisEvent.EventType = ES_INIT;

    SR_Init();
    SR_Write(0x00);

    if (ES_PostToService( MyPriority, ThisEvent) == true)
    {
        return true;
    }else
    {
        return false;
    }
}

/*****
Function
    PostLEDWriteService

Parameters
    EF_Event ThisEvent ,the event to post to the queue

Returns
    bool 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 PostLEDWriteService(ES_Event_t ThisEvent)
{
    return ES_PostToService(MyPriority, ThisEvent);
}

```

```

}

/*****
Function
    RunLEDWriteService

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_t RunLEDWriteService(ES_Event_t ThisEvent)
{
    ES_Event_t ReturnEvent;
    ReturnEvent.EventType = ES_NO_EVENT; // assume no errors

    static uint16_t SRData = 0;
    uint16_t command = 0;
    printf("LED: %d\r\n",SRData);
    switch (ThisEvent.EventType) {
        case LIGHT_CMD:
            command = (uint16_t) ThisEvent.EventParam;
            SRData &= LIGHT_LO;
            SRData |= (command << 3);
            break;
        case WATER_CMD:
            command = (uint16_t) ThisEvent.EventParam;
            SRData &= WATER_LO;
            SRData |= command;
            break;
        case HEALTH_CMD:
            command = (uint16_t) ThisEvent.EventParam;
            SRData &= HEALTH_LO;
            SRData |= (command << 8);
            break;
        case ALL_ON:
            SRData = ALL_HI;
            break;
        case ALL_OFF:
            SRData = ALL_LO;
            break;
        default:
            break;
    }

    SR_Write(SRData);

    return ReturnEvent;
}

/*****
private functions
*****/

```

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