

```

#include <stdint.h>
#include <stdbool.h>
#include <stdio.h>
#include "inc/hw_types.h"
#include "inc/hw_memmap.h"
#include "driverlib/sysctl.h"
#include "driverlib/gpio.h"
#include "driverlib/interrupt.h"
#include "utils/uartstdio.h"

#include "ES_Configure.h"
#include "ES_Framework.h"
#include "ES_Port.h"
#include "termio.h"

#define clrScrn()          printf("\x1b[2J")
#define goHome()         printf("\x1b[1,1H")
#define clrLine()        printf("\x1b[K")

int main(void)
{
    // Set the clock to run at 40MHz using the PLL and 16MHz external crystal
    SysCtlClockSet(SYSCTL_SYSDIV_5 | SYSCTL_USE_PLL | SYSCTL_OSC_MAIN
                   | SYSCTL_XTAL_16MHZ);
    TERMIO_Init();
    clrScrn();

    ES_Return_t ErrorType;

    // When doing testing, it is useful to announce just which program
    // is running.
    puts("\rStarting Test Harness for \r");
    printf("the 2nd Generation Events & Services Framework V2.2\r\n");
    printf("%s %s\n", __TIME__, __DATE__);
    printf("\n\r\n");
    printf("Press any key to post key-stroke events to Service 0\r\n");
    printf("Press 'd' to test event deferral \n\r");
    printf("Press 'r' to test event recall \n\r");

    // Your hardware initialization function calls go here

    // now initialize the Events and Services Framework and start it running
    ErrorType = ES_Initialize(ES_Timer_RATE_1mS);
    if ( ErrorType == Success )
    {
        ErrorType = ES_Run();
    }
    //if we got to here, there was an error
    switch (ErrorType)
    {
    case FailedPost:
        printf("Failed on attempt to Post\n");
        break;
    case FailedPointer:
        printf("Failed on NULL pointer\n");
        break;
    case FailedInit:
        printf("Failed Initialization\n");

```

```
        break;
default:
    printf("Other Failure\n");
    break;
}
for (;;)
    i
}
```