```c
#include "ES_Configure.h"
#include "ES_Framework.h"
#include "EventCheckers.h"
#include "CheckValidSM.h"
#include "sci.h"
#include "ES_Port.h"
#include "TransmitXbeeSM.h"
#include "ES_ServiceHeaders.h"

/* Checks if Asynchronous Transmit Register is empty*/
boolean TransmitBufferEmpty(void)
{
  static unsigned char LastTXIFState = 1;
  static unsigned char CurrentTXIFState = 0;
  boolean ReturnVal = False;
  CurrentTXIFState = TXIF;
  if ( (CurrentTXIFState != LastTXIFState) &&(CurrentTXIFState == HIGH) )
  {
    ES_Event ThisEvent;
    ThisEvent.EventType = ES_TRMT_REG_EMPTY;
    PostTransmitXbeeSM(ThisEvent);
    ReturnVal = True;
  }
  LastTXIFState = CurrentTXIFState;
  return ReturnVal;
}

/* Checks if Balloon Monitor has set the Balloon Popping flag high*/
boolean Check4BalloonPopped(void)
{
  static unsigned char LastPinState = 0;
  unsigned char CurrentPinState;
  boolean ReturnVal = False;
  CurrentPinState = RC4;
  if ( (CurrentPinState != LastPinState) && (CurrentPinState == HIGH) )
  {
    ES_Event ThisEvent;
    ThisEvent.EventType = ES_BALLOON_POPPED;
    ThisEvent.EventParam = 1;
    PostInterpretSM(ThisEvent);
    ReturnVal = False;
  }
  LastPinState = CurrentPinState;
  return ReturnVal;
}

/* This is a system event defined in Framework.c*/
/* Checks if a byte has been received through Asynchronous port*/
static boolean CheckSystemEvents( void )
{
  if ( kbhit() != 0 ) // new key waiting?
  {
```

```
    ES_Event ThisEvent;
    ThisEvent.EventType = ES_XB_Msg_Rec;
    ThisEvent.EventParam = getche();
    (*pPostKeyFunc)( ThisEvent );
    ES_Timer_InitTimer(_1_SEC_TIMER_CV, ONE_SEC );
    return True;
  }
  return False;
}

/* Following are the functions used directly from the Framework*/
void USART_Init( void )
{
    // TXSTA = set TXEN and BRGH, all others clear
    TXEN = 1;
    BRGH = 1;

    // RCSTA = Set SPEN & CREN, all others clear
    SPEN = 1;
    CREN = 1;

    // BAUDCTL = set BRG16, all others clear
    BRG16 = 1;

    // Baudrate, assuming 20MHz & the prior register settings
    // Setting 9600 baud rate
    SPBRGH = 0x02;
    SPBRG = 0x08;

   // disable the analog function on the RX pin
   ANS11 = 0;
}

bit kbhit ( void )
{
    if ( RCIF == 1)
        return (1 );
    else
        return ( 0 );
}
```