```
/* A newly opened port has the default configuration. The default port configuration is:
*
*   * A receive queue of 512 bytes
*   * A default CTS timeout (currently 5 seconds) set
*   * 1 stop bit
*   * 8 data bits
*   * Hardware handshaking on input
*   * Flow control enabled
*   * For RS-232 connections, the baud rate you specified when you opened the port.
*/
#include "serial.h"


/********************
*
*   functions
*
********************/

void SerialPortOpenError(Err error, UInt16* portId)
{
    switch(error){
        case serErrAlreadyOpen:
            FrmCustomAlert(SerialPortAlert,"Serial port already in use.","","");
            SrmClose(*portId);
            break;
        case serErrBadParam:
            FrmCustomAlert(SerialPortAlert,"The configP parameter is null.","","");
            break;
        case serErrBadPort:
            FrmCustomAlert(SerialPortAlert,"The port you are trying to open does not 
exist.","","");
            break;
        case memErrNotEnoughSpace:
            FrmCustomAlert(SerialPortAlert,"Not enough memory available to open port.","",
"");
            break;
        default:
            FrmCustomAlert(SerialPortAlert,"Unknown error opening serial port.","","");
            break;
    }
}

void SerialPortCloseError(Err error)
{
    switch(error){
        case serErrBadPort:
            FrmCustomAlert(SerialPortAlert,"The port you are trying to close does not 
exist.","","");
            break;
        case serErrNotOpen:
            FrmCustomAlert(SerialPortAlert,"The port you are trying to close is not open."
,"","");
            break;
        case serErrNoDevicesAvail:
            FrmCustomAlert(SerialPortAlert,"No serial devices could be found.","","");
            break;
        default:
            FrmCustomAlert(SerialPortAlert,"Unknown error closing serial port.","","");
            break;
    }
}

void SerialPortReceiveError(Err error, UInt16* portId)
{
    switch(error){
        case serErrBadPort:
            FrmCustomAlert(SerialPortAlert,"The port you are trying to receive data from
does not exist.","","");
            break;
        case serErrNotOpen:
            FrmCustomAlert(SerialPortAlert,"The port you are trying to receive data from
is not open.","","");
            break;
        case serErrTimeOut:
            FrmCustomAlert(SerialPortAlert,"Unable to receive data within specified
timeout period.","","");
            break;
        case serErrNotSupported:
            FrmCustomAlert(SerialPortAlert,"Port is not foreground port.","","");
            break;
        case serErrBadParam:
            FrmCustomAlert(SerialPortAlert,"Bytes Parameter exceeds the size of receive
queue.","","");
            break;
        case serErrLineErr:
            FrmCustomAlert(SerialPortAlert,"A line error occured during the receipt of
data.","","");
            SrmClearErr(*portId);
            break;
        case serErrNoDevicesAvail:
            FrmCustomAlert(SerialPortAlert,"No serial devices could be found.","","");
            break;
        //specific to SrmReceive
        case serErrConfigurationFailed:
            FrmCustomAlert(SerialPortAlert,"The port could not configure itself.","","");
            break;
        default:
            FrmCustomAlert(SerialPortAlert,"Unknown error receiving from serial port.","",
"");
            break;
    }
}
```

```
void SerialPortSendError(Err error)
{
    switch(error){
        case serErrBadPort:
            FrmCustomAlert(SerialPortAlert,"The port you are trying to send data to does
not exist.","","");
            break;
        case serErrNotOpen:
            FrmCustomAlert(SerialPortAlert,"The port you are trying to send data to is
not open.","","");
            break;
        case serErrTimeOut:
            FrmCustomAlert(SerialPortAlert,"Unable to send data within specified time
period.","","");
            break;
        case serErrNoDevicesAvail:
            FrmCustomAlert(SerialPortAlert,"No serial devices could be found.","","");
            break;
        case serErrConfigurationFailed:
            FrmCustomAlert(SerialPortAlert,"The port could not configure itself.","","");
            break;
        case serErrNotSupported:
            FrmCustomAlert(SerialPortAlert,"The specified port is not the foreground
port.","","");
            break;
        default:
            FrmCustomAlert(SerialPortAlert,"Unknown error sending to serial port.","","");
            break;
    }
}
```