



## **ELAN DIGITAL SYSTEMS LTD.**

LITTLE PARK FARM ROAD,  
SEGENSWORTH WEST,  
FAREHAM,  
HANTS. PO15 5SJ.

TEL: (44) (0)1489 579799

FAX: (44) (0)1489 577516

e-mail: [support@elandigitalsystems.com](mailto:support@elandigitalsystems.com)

website: <http://www.elandigitalsystems.com>

## **eSerial Device Driver Programmer's Guide ES317**



# PROVISIONAL

All trademarks are duly acknowledged.

### **REVISION HISTORY**

ISSUE	PAGES	DATE	NOTES
1	12	24.10.2001	FIRST ISSUE
2	12	03/12/2003	Updated OS support info

# Table of Contents

<b>OVERVIEW .....</b>	<b>4</b>
<b>OPERATING SYSTEM SUPPORT .....</b>	<b>4</b>
<b>DATA TYPES.....</b>	<b>5</b>
ELS_CARD_SOCKET_HANDLE .....	5
Declaration:.....	5
Description:.....	5
<b>ELSERIAL EXPORTED FUNCTIONS .....</b>	<b>6</b>
ELSERIAL_OPEN.....	6
Description:.....	6
Declaration:.....	6
Parameters:.....	6
Return parameters: .....	6
ELSERIAL_CLOSE.....	6
Description:.....	6
Declaration:.....	6
Parameters:.....	6
Return parameters: .....	6
ELSERIAL_READCSR.....	6
Description:.....	6
Declaration:.....	6
Parameters:.....	7
Return parameters: .....	7
ELSERIAL_WRITECSR .....	7
Description:.....	7
Declaration:.....	7
Parameters:.....	7
Return parameters: .....	7
ELSERIAL_SETUPDIO .....	7
Description:.....	7
Declaration:.....	7
Parameters:.....	7
Return parameters: .....	7
ELSERIAL_READDIO .....	8
Description:.....	8
Declaration:.....	8
Parameters:.....	8
Return parameters: .....	8
ELSERIAL_WRITEDIO.....	8
Description:.....	8
Declaration:.....	8
Parameters:.....	8
Return parameters: .....	8
ELSERIAL_SETVPUSERIALPORTMODE .....	8
Description:.....	8
Declaration:.....	8
Parameters:.....	8
Return parameters: .....	9
ELSERIAL_SETVPUPARALLELPORTMODE .....	9

Description:.....	9
Declaration:.....	9
Parameters:.....	9
Return parameters: .....	9
ELSERIAL_SETVPUEXTERNALFIFOMODE1 .....	9
Description:.....	9
Declaration:.....	9
Parameters:.....	9
Return parameters: .....	9
ELSERIAL_SETVPUEXTERNALFIFOMODE2.....	9
Description:.....	9
Declaration:.....	9
Parameters:.....	9
Return parameters: .....	10
ELSERIAL_READEXTFIFO1 .....	10
Description:.....	10
Declaration:.....	10
Parameters:.....	10
Return parameters: .....	10
ELSERIAL_WRITEEXTFIFO.....	10
Description:.....	10
Declaration:.....	10
Parameters:.....	10
Return parameters: .....	10
ELSERIAL_READEXTFIFO2 .....	11
Description:.....	11
Declaration:.....	11
Parameters:.....	11
Return parameters: .....	11
ELSERIAL_SETEXTFIFO1SIZE & ELSERIAL_SETEXTFIFO2SIZE.....	11
Description:.....	11
Declaration:.....	11
Parameters:.....	11
Return parameters: .....	11
ELSERIAL_SETEXTFIFO1TIMEOUTS & ELSERIAL_SETEXTFIFO2TIMEOUTS.....	11
Description:.....	11
Declaration:.....	12
Parameters:.....	12
Return parameters: .....	12

## Overview

The elSerial device driver is a serial device driver that enhances the standard serial driver functionality by allowing access to extended features of Elan's VPU 1655x chip.

*This document should be read in conjunction with the VPU1655x product data sheet and the elSerial.h C header file.*

## Operating System Support

Currently the elSerial device driver supports the following operating systems:

- Microsoft Windows CE
- Microsoft Windows PocketPC 2000/2002/2003

## Planned Operating System Support

In the future, we plan to support the enhanced features of the VPU chip on Windows 2000/XP operating systems with a WDM filter driver. Currently the Microsoft standard serial.sys driver works with serial and modem cards based on the VPU chip.

## Data Types

### ***ELS\_CARD\_SOCKET\_HANDLE***

#### **Declaration:**

```
typedef struct _ELS_CARD_SOCKET_HANDLE
{
    UCHAR uSocket;
    UCHAR uFunction;
} ELS_CARD_SOCKET_HANDLE, *PELS_CARD_SOCKET_HANDLE;
```

#### **Description:**

This structure forms the socket and function number of the PCMCIA card that contains the VPU 1655x chip.

- uFunction should be set to 0 for single function devices.
- uSocket should be set to the socket number of the PCMCIA card being opened. The first PCMCIA socket is number 0.

## **elSerial Exported Functions**

In addition to the standard Windows serial driver API, the elSerial driver exports additional IOControl calls to access the extended features available in Elan's VPU 1655x enhanced UART chip

### ***elSerial\_Open***

#### **Description:**

This function locates the COM port of the device (in a particular PCMCIA/CF socket) containing a VPU 1655x chip and returns an open handle to the COM port

#### **Declaration:**

```
HANDLE elSerial_Open(ELS_CARD_SOCKET_HANDLE hCardSock);
```

#### **Parameters:**

ELS\_CARD\_SOCKET\_HANDLE hCardSock  
Handle (socket and function) of the card.

#### **Return parameters:**

If the call was successful, the return value is the open COM port handle.  
If the call failed, the return value is NULL.

### ***elSerial\_Close***

#### **Description:**

Call this function to close the COM port.

#### **Declaration:**

```
void elSerial_Close(HANDLE hCOMPort);
```

#### **Parameters:**

HANDLE hCOMPort  
The open COM port handle returned from the call to elSerial\_Open.

#### **Return parameters:**

None.

### ***elSerial\_ReadCSR***

#### **Description:**

Reads from the Configuration and Status "CSR" register.

#### **Declaration:**

```
ULONG elSerial_ReadCSR(HANDLE hDevice, PCHAR pCSRReg);
```

**Parameters:**

HANDLE hDevice

The open COM port handle returned from the call to `elSerial_Open`.

PUCHAR pCSRReg

Pointer to a UCHAR for returning the value of the CSR register.

**Return parameters:**

Returns -1 if there is an error.

Otherwise \*pCSRReg contains the value in the CSR register.

***elSerial\_WriteCSR*****Description:**

Write to the Configuration and Status “CSR” register.

**Declaration:**

```
ULONG elSerial_WriteCSR(HANDLE hDevice, PCHAR pCSRReg);
```

**Parameters:**

HANDLE hDevice

The open COM port handle returned from the call to `elSerial_Open`.

PUCHAR pCSRReg

Pointer to a UCHAR containing the value to write.

**Return parameters:**

Returns -1 if there is an error.

***elSerial\_SetupDIO*****Description:**

Sets up the Digital IO port on the VPU16550 UART

**Declaration:**

```
ULONG elSerial_SetupDIO(HANDLE hDevice, PCHAR pDIODir);
```

**Parameters:**

HANDLE hDevice

The open COM port handle returned from the call to `elSerial_Open`.

PUCHAR pDIODir

Pointer to a UCHAR containing the bit pattern of digital IO direction bits to write to the DIO DIR register.

**Return parameters:**

Returns -1 if there is an error.

## ***elSerial\_ReadDIO***

### **Description:**

Reads the Digital IO port on the VPU1655x UART

### **Declaration:**

```
ULONG elSerial_ReadDIO(HANDLE hDevice, PCHAR pDIOR);
```

### **Parameters:**

HANDLE hDevice

The open COM port handle returned from the call to elSerial\_Open.

PCHAR pDIOR

Pointer to a UCHAR value for returning the DIOR register value.

### **Return parameters:**

Returns -1 if there is an error.

Otherwise \*pDIOR contains the DIOR register value.

## ***elSerial\_WriteDIO***

### **Description:**

Write to the Digital IO port on the VPU1655x UART

### **Declaration:**

```
ULONG elSerial_WriteDIO(HANDLE hDevice, PCHAR lpInBuffer);
```

### **Parameters:**

HANDLE hDevice

The open COM port handle returned from the call to elSerial\_Open.

PCHAR pDIOR

Pointer to a UCHAR value containing the value to write to the DIOR register.

### **Return parameters:**

Returns -1 if there is an error.

## ***elSerial\_SetVPUSerialPortMode***

### **Description:**

Sets the elSerial driver into standard serial port mode

### **Declaration:**

```
int elSerial_SetVPUSerialPortMode(HANDLE hDevice)
```

### **Parameters:**

HANDLE hDevice

The open COM port handle returned from the call to elSerial\_Open.

**Return parameters:**

Returns -1 if there is an error.

***elSerial\_SetVPUParallelPortMode*****Description:**

Sets the elSerial driver into parallel port mode

**Declaration:**

```
int elSerial_SetVPUParallelPortMode(HANDLE hDevice)
```

**Parameters:**

HANDLE hDevice

The open COM port handle returned from the call to elSerial\_Open.

**Return parameters:**

Returns -1 if there is an error.

***elSerial\_SetVPUExternalFifoMode1*****Description:**

Sets the elSerial driver into external fifo mode 1

(bi-directional, 2 fifo's) as explained in vpu16551.pdf Fig 7.3.2.2-1

**Declaration:**

```
int elSerial_SetVPUExternalFifoMode1(HANDLE hDevice)
```

**Parameters:**

HANDLE hDevice

The open COM port handle returned from the call to elSerial\_Open.

**Return parameters:**

Returns -1 if there is an error.

***elSerial\_SetVPUExternalFifoMode2*****Description:**

Sets the elSerial driver into external fifo mode 2

(dual inward facing fifo's) as explained in vpu16551.pdf Fig 7.3.2.2-2

**Declaration:**

```
int elSerial_SetVPUExternalFifoMode2(HANDLE hDevice)
```

**Parameters:**

HANDLE hDevice

The open COM port handle returned from the call to `elSerial_Open`.

**Return parameters:**

Returns -1 if there is an error.

***elSerial\_ReadExtFifo1***

**Description:**

Read from fifo 1 in `VPU_EXT_FIFO_MODE1` or `VPU_EXT_FIFO_MODE2`

**Declaration:**

`ULONG elSerial_ReadExtFifo1(HANDLE hDevice, PVOID pBuffer, ULONG BufferSize)`

**Parameters:**

`HANDLE hDevice`

The open COM port handle returned from the call to `elSerial_Open`.

`PVOID pBuffer`

Pointer to the buffer that is to be used to store the fifo data.

`ULONG BufferSize`

The size of the buffer (or number of bytes to read).

**Return parameters:**

Returns -1 if there is an error, or the number of bytes read into the supplied buffer.

***elSerial\_WriteExtFifo***

**Description:**

Write to the external output fifo in `VPU_EXT_FIFO_MODE1`

**Declaration:**

`int elSerial_WriteExtFifo(HANDLE hDevice, PVOID pBuffer, ULONG BufferSize)`

**Parameters:**

`HANDLE hDevice`

The open COM port handle returned from the call to `elSerial_Open`.

`PVOID pBuffer`

Pointer to the buffer that is to be used to store the fifo data.

`ULONG BufferSize`

The size of the buffer (or number of bytes to write).

**Return parameters:**

Returns -1 if there is an error.

## ***eISerial\_ReadExtFifo2***

### **Description:**

Read from fifo 2 in VPU\_EXT\_FIFO\_MODE2

### **Declaration:**

```
ULONG eISerial_ReadExtFifo2(HANDLE hDevice, PVOID pBuffer, ULONG  
BufferSize)
```

### **Parameters:**

HANDLE hDevice

The open COM port handle returned from the call to eISerial\_Open.

PVOID pBuffer

Pointer to the buffer that is to be used to store the fifo data.

ULONG BufferSize

The size of the buffer (or number of bytes to read).

### **Return parameters:**

Returns -1 if there is an error, or the number of bytes read into the supplied buffer.

## ***eISerial\_SetExtFifo1Size & eISerial\_SetExtFifo2Size***

### **Description:**

Sets the size of the hardware fifo's, this should be set to the number of bytes that are written to the fifo which will trigger an interrupt, and hence the number of bytes read from the fifo during the interrupt service routine. During this function call, the internal ring buffer is reset. Make sure the data stream running into the fifo is also synchronised at the point of calling this function otherwise the packets of data may be miss-aligned.

### **Declaration:**

```
int eISerial_SetExtFifo1Size(HANDLE hDevice, PULONG pFifoSize)
```

```
int eISerial_SetExtFifo2Size(HANDLE hDevice, PULONG pFifoSize)
```

### **Parameters:**

HANDLE hDevice

The open COM port handle returned from the call to eISerial\_Open.

### **Return parameters:**

Returns -1 if there is an error.

## ***eISerial\_SetExtFifo1Timeouts & eISerial\_SetExtFifo2Timeouts***

### **Description:**

Sets the timeouts values for reading the external fifo's.

**Declaration:**

```
int elSerial_SetExtFifo1Timeouts(HANDLE hDevice, COMMTIMEOUTS
*pCommTimeouts)
int elSerial_SetExtFifo2Timeouts(HANDLE hDevice, COMMTIMEOUTS
*pCommTimeouts)
```

**Parameters:**

HANDLE hDevice

The open COM port handle returned from the call to elSerial\_Open.

COMMTIMEOUTS \*pCommTimeouts

Filled COMMTIMEOUTS structure.

**Return parameters:**

Returns -1 if there is an error.