Serial IOP Driver ERS

Network Systems Development

S S
B O'C
J L

Introduction

This ERS describes the SCC asynchronous serial driver written for the SCC Input Output Processors (IOP). The IOP driver provides the functionality of the Macintosh serial driver as described in **Inside Macintosh** volumes 1 and 4. Combined with the appropriate driver interface on the Macintosh, the IOP driver relieves the Macintosh of the overhead of directly dealing with the SCC.

The IOP serial driver runs as a task under the IOP Kernel. For a description of the IOP Kernel see the **IOP Kernel ERS**. All communication with the driver is through a message technique described in the **IOP Manager ERS**.

There are two separate drivers, one for the modem port (or port A) and one for the printer port (or Port B). These drivers are contained in standard Macintosh resources of type 'SERD' and id's of 60 and 61. The first two bytes of each driver contains the length of the driver, the following bytes contain the driver ready to be downloaded into the IOP.

Driver Overview

Under the IOP Kernel each driver has six message boxes for its use. Three Host to IOP message boxes and three IOP to Host message boxes. These message boxes will be referred to as message boxes 1, 2, and 3. When sending messages to or receiving messages from the IOP driver the real message box number must be used. For the port A driver these are message box numbers 2, 3, and 4. For the port B driver these are message box numbers 5, 6, and 7.

Host to IOP message box 1 is used for read messages, message box 2 is used for write messages, and message box 3 is used for open/control/close messages. IOP to Host message box 1 is used for event messages.

The IOP driver is allocated and initialized through the standard techniques as described in the **IOP Kernel ERS**. After initialization, the driver is ready to accept open, read, write, control, and close

messages. In addition to accepting these messages, the driver may send an event message to the host computer. There is no status message for the driver, the information needed for status can be determined by reading the statistics buffer from IOP memory. The address of the statistics buffer is returned in the open message. The technique for reading the statistics buffer if covered later in this document.

Message Descriptions

Open Message - Host to IOP Message box 3

Offset	Length	Direction	Description
$00	1	Output	Result code
$01	1	Input	Open opcode ($01)
$02	2		Unused
$04	2	Input	Serial configuration word
$06	2	Input	SCC baud rate divisor
$08	2	Output	Address of IOP write buffer
$0A	2	Output	Length of IOP write buffer
$0C	2	Output	Address of IOP statistics buffer

The open message causes the IOP driver to initialize the SCC and the internal data structures with the configuration supplied in the serial configuration word. The format of the configuration word is described in Inside Macintosh volume II page 250. Bits 0-9 of the configuration word as shown in Inside Macintosh is actually the SCC baud rate divisor for the original Macintosh. The host must convert the divisor in the configuration word to the correct divisor for the IOP SCC and place that value in the SCC baud rate divisor field in the message. The formula for the conversion is:

Baud Divisor = ((Old Divisor + 2) * 36864 / 36707) - 2

Where Old Divisor = Bits 0-9 of the configuration word. The results of the division must be rounded, not truncated.

Read Message - Host to IOP Message box 1

Offset	Length	Direction	Description
$00	1	Output	Result code
$01	3		Unused
$04	2	Input	Bytes taken from IOP read buffer
$06	1	Input	Abort acknowledged
$06	2	Output	Address of IOP read ring buffer
$08	2	Output	Address of byte past end of ring buffer
$0A	2	Output	Address of first byte in ring buffer
$0C	2	Output	Number of bytes in IOP ring buffer
$0E	2	Output	Number of bytes in buffer before an abort condition occurred

The read message will complete when at least one byte is in the IOP ring buffer or an abortable error condition has occurred. The read message will return the address of the IOP ring buffer, the number of bytes in the buffer, and the address of the first byte to remove from the buffer. If there was an abortable error condition, the message will contain the number of bytes in the buffer before the error condition occurred. A read completing with zero bytes in the buffer signifies that an error condition occurred with an empty buffer.

After the bytes are read by the host, the IOP driver must be notified how many bytes were read by placing the number read in the input parameter of the next read message. In addition, if the host has read up to or past the abort condition, the abort acknowledged field should contain a non zero number.

Write Message - Host to IOP Message box 2

Offset	Length	Direction	Description
$00	1	Output	Result code
$01	3		Unused
$04	2	Input	Length of data to write

The open message return the IOP address of the write buffer and the

length of the write buffer. The data must be moved into the write buffer by the host before the write message is sent to the IOP. When the last byte of the write data has cleared the SCC transmit buffer the write message will complete. If the data the host wants to write is longer than the IOP write buffer, the data must be broken into multiple write messages.

Control Messages - Host to IOP Message box 3

The opcodes and the parameters to the control messages are the same as the Macintosh serial driver control calls except for the control calls that contain baud rate information. For an explanation of control calls and the parameters see **Inside Macintosh** volumes 1 and 4.

Serial Reset

Offset	Length	Direction	Description
$00	1	Output	Result code
$01	1	Input	Control opcode ($00)
$02	1	Input	Serial Reset opcode ($08)
$03	1		Unused
$04	2	Input	Serial configuration word
$06	2	Input	SCC baud rate divisor

The parameters to the Serial reset control message are the same as the parameters to the open message. See the description of the open message for a further explanation of the parameters.

Serial Handshake

Offset	Length	Direction	Description
$00	1	Output	Result code
$01	1	Input	Control opcode ($00)
$02	1	Input	Serial Handshake opcode ($0A)
$03	1		Unused
$04	1	Input	XOn/XOff output flow control flag
$05	1	Input	CTS output flow control flag

$06	1	Input	XOn character
$07	1	Input	XOff character
$08	1	Input	Errors that cause abort
$09	1	Input	Status changes that cause events
$0A	1	Input	XOn/XOff input flow control flag

Clear Break Mode

Offset	Length	Direction	Description
$00	1	Output	Result code
$01	1	Input	Control opcode ($00)
$02	1	Input	Clear Break opcode ($0B)

Set Break Mode

Offset	Length	Direction	Description
$00	1	Output	Result code
$01	1	Input	Control opcode ($00)
$02	1	Input	Set Break opcode ($0C)

Set Baud Rate

Offset	Length	Direction	Description
$00	1	Output	Result code
$01	1	Input	Control opcode ($00)
$02	1	Input	Set baud rate opcode ($0D)
$03	1		Unused
$04	2	Input	SCC baud rate divisor

The formula for determining the SCC baud rate divisor is:

SCC baud rate divisor = (115200 / baud rate) - 2

Extended Serial Handshake

Offset	Length	Direction	Description

$00	1	Output	Result code
$01	1	Input	Control opcode ($00)
$02	1	Input	Extended Serial Handshake opcode ($0E)
$03	1		Unused
$04	1	Input	XOn/XOff output flow control flag
$05	1	Input	CTS output flow control flag
$06	1	Input	XOn character
$07	1	Input	XOff character
$08	1	Input	Errors that cause abort
$09	1	Input	Status changes that cause aborts
$0A	1	Input	XOn/XOff input flow control flag
$0B	1	Input	DTR input flow control flag

Control Options

Offset	Length	Direction	Description
$00	1	Output	Result code
$01	1	Input	Control opcode ($00)
$02	1	Input	Control options opcode ($10)
$03	1		Unused
$04	1	Input	Options byte

Assert DTR

Offset	Length	Direction	Description
$00	1	Output	Result code
$01	1	Input	Control opcode ($00)
$02	1	Input	Assert DTR opcode ($11)

Clear DTR

Offset	Length	Direction	Description
$00	1	Output	Result code
$01	1	Input	Control opcode ($00)
$02	1	Input	Clear DTR opcode ($12)

Set PE Substitution Char

Offset	Length	Direction	Description
$00	1	Output	Result code
$01	1	Input	Control opcode ($00)
$02	1	Input	PE substitution opcode ($13)
$03	1		Unused
$04	1	Input	PE substitution character

Set Alternate PE Substitution Char

Offset	Length	Direction	Description
$00	1	Output	Result code
$01	1	Input	Control opcode ($00)
$02	1	Input	PE substitution opcode ($14)
$03	1		Unused
$04	1	Input	PE substitution character
$05	1	Input	Alternate PE substitution character

Set XOff

Offset	Length	Direction	Description
$00	1	Output	Result code
$01	1	Input	Control opcode ($00)
$02	1	Input	Set Xoff opcode ($15)

Clear XOff

Offset	Length	Direction	Description
$00	1	Output	Result code
$01	1	Input	Control opcode ($00)
$02	1	Input	Clear Xoff opcode ($16)

Send XOn

Offset	Length	Direction	Description
$00	1	Output	Result code
$01	1	Input	Control opcode ($00)
$02	1	Input	Send XOn opcode ($17)

Unconditionally Send XOn

Offset	Length	Direction	Description
$00	1	Output	Result code
$01	1	Input	Control opcode ($00)
$02	1	Input	Unconditional Send XOn opcode ($18)

Send XOff

Offset	Length	Direction	Description
$00	1	Output	Result code
$01	1	Input	Control opcode ($00)
$02	1	Input	Send XOff opcode ($19)

Unconditionally Send XOff

Offset	Length	Direction	Description
$00	1	Output	Result code
$01	1	Input	Control opcode ($00)
$02	1	Input	Unconditional Send XOff opcode ($1A)

Reset SCC

Offset	Length	Direction	Description
$00	1	Output	Result code
$01	1	Input	Control opcode ($00)
$02	1	Input	Reset SCC opcode ($1B)

Close Message - Host to IOP Message box 3

Offset	Length	Direction	Description
$00	1	Output	Result code
$01	1	Input	Close opcode ($02)

Event Message - IOP to Host Message box 1 (2)

Offset	Length	Direction	Description
$04	1	Output	SCC Read register 0 at interrupt time
$05	2	Output	SCC Read register 0 bits that have changed

The event message will be sent to the host when a SCC status change has occurred that the host has specified in the Serial Handshake control call.

Determining Status Information

The Macintosh serial driver has a status call to retrieve information about the state of the serial driver. This information is contained in six bytes described in **Inside Macintosh** volume 2 page 253. Since status information can be obtained with an immediate status call, a message can not be sent to the IOP driver to obtain the information. The IOP driver maintains the status information in memory and can be accessed by the following procedure:

1. Read the seven bytes from the IOP memory at the location returned by the open message.

2. If the first byte is non zero, place a zero into the second byte read from the IOP memory.

3. Write an $FF into the first memory location returned by the open message.

4. Bytes 2-7 of the bytes from the IOP memory contain the status information described in Inside Macintosh.