CHRP™ ISA Serial Port Device Binding

to:

IEEE 1275-1994

Standard for Boot

(Initialization, Configuration)

Firmware

Revision: 1.0 Unapproved DRAFT

Date: May 6, 1996

Table of Contents

1. Purpose of this Device Binding 3
2. Revision History 3
3. References. 3
4. Definition of Terms. 3
5. Device Characteristics (Informative) 3
6. Device-specific Configuration Variables 3
7. Device Nodes 4
7.1 Properties 4
7.1.1 Open Firmware-defined Properties for Device Nodes. 4
7.1.2 Device-specific Properties for Device Nodes 4
7.2 Methods. 5
7.2.1 Open Firmware-defined Methods for Device Nodes. 5
7.2.1.1 Device Arguments for “Open” Method 5
7.2.2 Device-specific Methods for Device Nodes 6
8. User Interface Commands. 6
8.1 Open Firmware-defined User Interface Commands 6
8.2 Device-specific User Interface Commands 6
9. Device State 6
9.1 Device State When Client is Started. 6
9.2 Device State Required When Client Calls Open Firmware 6
10. Other Commentary 7

1. Purpose of this Device Binding

This document specifies the application of Open Firmware to PowerPC Common Hardware Reference Platform (CHRP™) ISA serial port (UART Controller) devices, including device-specific requirements and practices for initialization, properties, and methods. This device shall be a child of an ISA or EISA Bus Node.

2. Revision History

Revision 1.0 Unapproved DRAFTMay 6, 1996	Initial revision. Jordan Brown, Sunsoft and John Kingman, IBM editors

3. References

This Open Firmware System binding standard *shall* be used in conjunction with the following publications. When the following standards are superseded by an approved revision, the revision *shall* apply.

[1] *IEEE Std 1275-1994 Standard for Boot (Initialization, Configuration) Firmware, Core Practices and Requirements*.

[2] *Core Errata, IEEE P1275.7/D4.*

[3] *ISA/EISA/ISA-PnP binding to: IEEE Std 1275-1994, Standard for Boot (Initialization, Configuration) Firmware*.

[4] *Device Support Extensions to: IEEE Std 1275-1994, Standard for Boot (Initialization, Configuration) Firmware*.

[5] *PowerPC Microprocessor Common Hardware Reference Platform: I/O Device Reference*. This document describes the PowerPC Common Hardware Reference Platform (CHRP™) System Standard I/O Devices; hardware registers, register locations, and hardware attributes.

[6] *Open Firmware Recommended Practice: Interrupt Mapping*.

[7] *Open Firmware Recommended Practice: Generic Names*.

[8] *PowerPC Microprocessor Common Hardware Reference Platform binding to: IEEE Std 1275-1994, Standard for Boot (Initialization, Configuration) Firmware*.

4. Definition of Terms

EISA: Extended Industry Standard Architecture

ISA: Industry Standard Architecture

UART: Universal Asynchronous Receiver/Transmitter

5. Device Characteristics (Informative)

The ISA serial port (UART Controller) is one of the most standardized pieces of hardware in the industry, mostly due to the relative lack of support for it in older operating systems and so the tendency for applications programs to directly manipulate it. See [5] for more information.

6. Device-specific Configuration Variables

None.

7. Device Nodes

7.1. Properties

As specified in [1], [3] and [8], with the following additions or modifications.

7.1.1. Open Firmware-defined Properties for Device Nodes

"name" **S**

Standard *property name*, specifies the generic name of the device.

The meaning of this property is as defined in Open Firmware core document [1], as modified by the *Generic Names Recommended Practice* [7]. The value for nodes described by this specification shall be "serial".

"device_type" **S**

Standard *property name* to define the device's implemented interface.

The meaning of this property is as defined in the Open Firmware core document [1]. The value for nodes described by this specification shall be "serial".

"compatible" **S**

Standard *property name*, specifies device names with which this device is compatible.

The meaning of this property is as defined in Open Firmware, as modified by the *Generic Names Recommended Practice* [7]. As described in those documents, the entries are a list of device names with which this device is compatible, starting with the name of the device itself and progressing through successively less precise and possibly less functional compatible devices.

For this device, the compatible property shall include "pnpPNP,501."

Additional entries may be supplied, at their appropriate position in the list, to describe devices with which this device is compatible.

"reg" **S**

Standard *property name* to define the package's registers.

The meaning of this property is as defined in the Open Firmware core document [1]. It describes the device's register set. The values which shall be assigned to this property are explained in the *ISA/EISA/ISA-PnP binding*[3] and the *I/O Device Reference*[5]

"interrupts" **S**

Standard *property name* to define the package's interrupts.

The meaning of this property is as defined in the *Interrupt Mapping Recommended Practice* [6]. The values assigned to this property are explained in the *ISA/EISA/ISA-PnP binding*[3] and the *I/O Device Reference*[5].

7.1.2. Device-specific Properties for Device Nodes

As specified in [3], with the following addition.

"clock-frequency" **S**

Standard property, encoded as with encode-int, that shall be the baud-rate generator's clock input frequency (in hertz).

Typically, the clock frequency is nominally 1,843,200 Hz. Some devices generate the baud rate input clock by dividing a higher-frequency clock source that is not an exact multiple of 1,843,200 Hz, resulting in a small but acceptable error in the nominal clock frequency. In such cases, the "clock-frequency" shall report the actual nominal frequency. For example, if the baud rate input clock is generated by dividing a 24 MHz clock by 13, the value of the "clock-frequency" property shall be 1846153.

7.2. Methods

7.2.1. Open Firmware-defined Methods for Device Nodes

As specified in [1] and [3], with the following addition or modification. This device shall be have its interrupt lines “tri-stated” when closed.

7.2.1.1. Device Arguments for “Open” Method

As specified in [1], [3] and [4], with the following addition or modification.

open is a Standard Method for the Serial Device. When the Serial Device is opened, several device-arguments can be passed as defined below:

driver-name@unit-address:**device-arguments**

device-arguments are defined to be of the form, e.g. 9600,8,n,1, and sets the UART parameters (baudrate, #data bits, parity, #stop bits, and handshake) accordingly.

The fields are (in order, left to right):

<baud rate>, <data bits>, <parity>, <stop bits>, <handshake>

Fields with empty values are not changed. Values for fields are whatever the hardware will support:

baud rates: various, including 110, 300, 1200, 2400, 4800, 9600, 19200, 38400

character bits: 5,6,7,8

parity:

char	means
n	none
e	even
o	odd
m	mark
s	space

stop bits:

char	means
1	1 stop bit
.	1.5 stop bits
2	2 stop bits

handshake:

char	means
-	none
h	hardware (rts/cts) \ Not supported
s	software (xon/xoff) \ Not supported

The default mode is 9600,8,n,1,-

7.2.2. Device-specific Methods for Device Nodes

As specified in [1] and [3], with the following additions.

set-mode **(** adr len **--** **)** **M**

Sets the device mode according to the string at *adr*, of length *len*, which is interpreted the same as the arguments to the "open" method (see Section 7.2.1.1. on page 5).

set-modem-control **(** bitmask **--** **)** **M**

Sets the device's modem control lines according to the following table:

bitmask	Modem control state
0	RTS off, DTR off
1	RTS off, DTR on
2	RTS on, DTR off
3	RTS on, DTR on

The response for other values of *bitmask* are implementation dependent.

Changing the bits associated with these lines shall have no effect on other bits not associated with this function.

8. User Interface Commands

8.1. Open Firmware-defined User Interface Commands

None.

8.2. Device-specific User Interface Commands

None.

9. Device State

9.1. Device State When Client is Started

For devices not selected as Open Firmware's “console input device” or “console output device” device, most of the initial state is typically undefined. This device shall be closed when the client is started, if it is not selected as Open Firmware's “console input device” or “console output device” device.

For devices which are selected as Open Firmware's “console input device” or “console output device” device, the device shall be initialized appropriately for the device to which it is attached.

Refer to [5] for more information on the state of this device when the client is started.

9.2. Device State Required When Client Calls Open Firmware

For devices not selected as Open Firmware's “console input device” or “console output device” device, there is no requirement.

For devices selected as Open Firmware's “console input device” or “console output device” device, the state should be unmodified from the initial state on client start-up.

Note: If the device is in a different state when the client calls Open Firmware, unpredictable behavior may result if Open Firmware accepts input or generates output. Clients

changing the device state should either restore the original state before calling Open Firmware or should avoid using Open Firmware features requiring user interaction. Changing the device state is likely to render Open Firmware unusable for debugging purposes.

10. Other Commentary