# Linux Installation Guide: LAVA Cards

## Installation of Serial Cards

***PRODUCT(S):***

- SSerial-PCI
- SSerial-PCI/LP
- LavaPort-650
- RS422 SS-PCI
- DSerial-PCI
- DSerial-PCI/LP
- DSerial-PCI Powered
- LavaPort-PCI
- Quattro-PCI
- Quattro-PCI/LP
- Quattro-PCI Powered
- LavaPort-Quad
- Octopus-550

***INSTALLATION:***

At bootup the Linux hardware discovery utility should automatically detect the PCI cards installed in the system and enumerate them as ttySx devices in the system's /dev directory. The actual ttyS numbers will depend on the number of serial ports that are currently present on the system.

To verify the actual allocation of the ports you must first determine the I/O range and IRQ that have been assigned to each serial port. To retrieve this information, query the PCI bus for a list of all LAVA PCI devices present.

To query the PCI bus from the terminal enter:

```
# lspci –v –d 1407:*
```

Depending on the number of cards present in the system, and whether or not your card supports multiple ports, you may receive a list of several items detected on the PCI bus. However, most should resemble the example below:

```
06:00.0 Serial controller: Lava Computer mfg Inc … … …
        Subsystem: Lava Computer mfg Inc … … …
        Flags: medium devsel, IRQ 21
        I/O ports at 3000 [size=8]
        Kernel driver in use: serial
```

The example above was returned by an SSerial-PCI card where the serial port has been configured for IRQ 21 and an I/O range of 0x3000. Every system is different and therefore will return a different IRQ and I/O range; however, the important thing to obtain from the command is the IRQ and I/O range of all ports on the card.

Once the IRQ and I/O range of all ports have been determined, you will need to verify that all the ports have been installed correctly. Any serial port installed on the system will be located in the /dev directory and can be listed from the terminal by entering:

```
# ls /dev/ttyS*
```

This command should display:

```
/dev/ttyS0   /dev/ttyS1   /dev/ttyS2   /dev/ttyS3
```

**NOTE:** By default certain installations of Linux may only support up to four serial ports (i.e. ttyS0-ttyS3), with the first port (ttyS0) allocated to the onboard serial port. Therefore it may be necessary to enable additional ports manually on the system. To do this refer to "APPENDIX A: Manually Adding Serial Ports" (see page 8).

Now, given the IRQ and I/O range, as well as the list of ports present, you can verify if the port was installed correctly, by entering:

```
# setserial –g /dev/ttyS[0123]
```

Where: **`[0123]`** indicates the ports you would like to display the information from.

This command should display:

```
/dev/ttyS0, UART:16550A, Port: 0x03f8, IRQ: 4
```

```
        /dev/ttyS1, UART:16550A, Port: 0x3000, IRQ: 21
        /dev/ttyS2, UART:unknown, Port: 0x03e8, IRQ: 4
        /dev/ttyS3, UART:unknown, Port: 0x02e8, IRQ: 4
```

The above output was returned by an SSerial-PCI card; therefore, the output returned by your system will likely vary based on the I/O range and IRQ of the port. However, it is important to note that in the output above the resources assigned to /dev/ttyS1 match the resources collected earlier and the UART type matches the UART present on the card, indicating that the LAVA card has been installed correctly on ttyS1. If the resources are correct then the installation of the card has been successful and you can either obtain more detailed information on the port and/or attempt to test the port.

**NOTE:** If the settings do not match the resources collected earlier then it may be necessary to manually configure the card. If so, please refer to "APPENDIX B: Manually Setting Serial Port Resources" (see page 9).

To obtain more detailed information on the port you can use the setserial command again by entering:

```
        # setserial –a /dev/ttyS1
```

This command should display:

```
        /dev/ttyS1, Line 1, UART: 16550A, Port: 0x3000, IRQ: 21
                Baud_base: 115200, close_delay: 50, divisor: 0
                closing_wait: 3000
                Flags: spd_normal skip_test
```

***TESTING:***

Information on testing serial ports can be found in: "APPENDIX D: Testing Your Installations" (see page 11).

## Installation of Parallel Cards

***PRODUCT(S):***

- Parallel-PCI
- Parallel-PCI/LP
- Dual Parallel-PCI

***INSTALLATION:***

At bootup the Linux hardware discovery utility should automatically detect any PCI parallel ports installed in the system and create both a parportX and lpX device in the system's /dev directory. The actual parport and lp numbers will depend on the number of parallel ports currently present on the system.

To verify the actual allocation of the ports you must first determine the I/O range and IRQ that have been assigned to each parallel port. To retrieve this information you will need to query the PCI bus for a list of all LAVA PCI devices present.

To query the PCI bus from the terminal, enter:

```
# lspci –v –d 1407:*
```

Depending on the number of cards present in the system, and whether or not your card supports multiple ports, you may receive a list of several items detected on the PCI bus. However, the output should resemble the example output below:

```
03:00.0 Parallel controller: Lava Computer mfg Inc … … …
        Subsystem: Lava Computer mfg Inc … … …
        Flags: medium devsel, IRQ 20
        I/O ports at cf00 [size=8]
        Kernel driver in use: parport_pc
```

The example above was returned by the Parallel-PCI card where the parallel port  has been configured for IRQ 20 and an I/O range of 0xCF00. Every system is different and therefore will return a different IRQ and I/O range; however, the important thing to obtain from the command is the IRQ and I/O range of all ports on the card.

Once the IRQ and I/O range of all ports have been determined you will need to verify that all the ports have been installed correctly. Any parallel port installed on the system will be located in the /dev directory and can be listed from the terminal by entering:

```
# ls /dev/parport*
```

This command should display:

```
/dev/parport0       /dev/parport1
```

In addition, each port should also have a line printer associated with it in the /dev directory that can be viewed by entering:

```
# ls /dev/lp*
```

This command should display:

```
/dev/lp0     /dev/lp1
```

Now given the IRQ and I/O range, as well as the list of ports present, you can verify whether the port was installed automatically by viewing the kernel's message buffer, by entering:

```
# dmesg | grep par
```

This command should display, among other things:

```
… … …
parport_pc 00:0a: reported by Plug and Play ACPI
parport0: PC-style at 0x378, irq 7 [PCSPP,TRISTATE]
PCI Parallel port detected: 1407:8000, I/O at 0xcf00(0x0), IRQ 20
parport1: PC-style at 0xcf00, irq 20 [PCSPP,TRISTATE,EPP]
parport1: Printer, Hewlett-Packard HP LaserJet 5MP
lp0: using parport0 (interrupt-driven).
lp1: using parport1 (interrupt-driven).
… … …
```

The above output was returned by the Parallel-PCI card. The output returned by your system will likely vary based on the I/O range and IRQ of the port. However, it is important to note that in the output above the resources assigned to /dev/parport1 match the resources collected earlier, indicating that the LAVA card has been installed correctly on parport1. If the resources are correct then the installation of the card has been successful and you may begin using the port.

**NOTE:** If the port was not automatically detected by the system or the settings don't match the resources collected earlier then it may be necessary to manually remove and re-add all parallel ports. If this is the case then refer to "APPENDIX C: Manually Adding Parallel Ports" (see page 10).

***TESTING:***

Information on testing parallel ports can be found in: "APPENDIX D: Testing Your Installations" (see page 11).

## Installation of Combo Cards

***PRODUCT(S):***

- SP-PCI
- 2SP-PCI

***INSTALLATION – Serial Ports:***

At bootup the Linux hardware discovery utility should automatically detect the PCI cards installed in the system and enumerate them as ttySx devices in the system's /dev directory. The actual ttyS numbers will depend on the number of serial ports that are currently present on the system.

To verify the actual allocation of the ports you must first determine the I/O range and IRQ that have been assigned to each serial port. To retrieve this information you will need to query the PCI bus for a list of all LAVA PCI devices present.

To query the PCI bus from the terminal, enter:

```
# lspci –v –d 1407:*
```

Depending on the number of cards present in the system, and whether or not your card supports multiple ports you may receive a list of several items detected on the PCI bus. However, most should resemble the example output below:

```
03:00.0 Parallel controller: Lava Computer mfg Inc … … …
         Subsystem: Lava Computer mfg Inc … … …
         Flags: medium devsel, IRQ 20
         I/O ports at cf00 [size=8]
         Kernel driver in use: parport_pc
03:00.1 Serial controller: Lava Computer mfg Inc … … [16550])
         Subsystem: Lava Computer mfg Inc … … …
         Flags: medium devsel, IRQ 21
         I/O ports at 3000 [size=8]
         Kernel driver in use: serial
```

The example above was returned by an SP-PCI card where the serial port has been configured for IRQ 21 and an I/O range of 0x3000. Every system is different and therefore will return a different IRQ and I/O range; however, the important thing to obtain from the command is the IRQ and I/O range of all ports on the card.

**NOTE:** The SP-PCI and 2SP-PCI cards will return the resources for both the parallel and serial ports present on the card; however, at the present we are only interested in the resources of the serial port.

Once the IRQ and I/O range of all ports have been determined, you will need to verify that all the ports have been installed correctly. Any serial port installed on the system will be located in the /dev directory and can be listed from the terminal by entering:

```
# ls /dev/ttyS*
```

This command should display:

```
/dev/ttyS0   /dev/ttyS1   /dev/ttyS2   /dev/ttyS3
```

**NOTE:** By default certain installations of Linux may only support up to four serial ports (i.e. ttyS0-ttyS3), with the first port (ttyS0) allocated to the onboard serial port. Therefore it may be necessary to enable additional ports manually on the system. To do this please refer to "APPENDIX A: Manually Adding Serial Ports" (see page 8).

Now, given the IRQ and I/O range, as well as the list of ports present, you can verify if the port was installed correctly, by entering:

```
# setserial –g /dev/ttyS[0123]
```

Where: [0123] indicates the ports you would like to display the information from. This command should display:

```
/dev/ttyS0, UART:16550A, Port: 0x03f8, IRQ: 4
/dev/ttyS1, UART:16550A, Port: 0x3000, IRQ: 21
/dev/ttyS2, UART:unknown, Port: 0x03e8, IRQ: 4
/dev/ttyS3, UART:unknown, Port: 0x02e8, IRQ: 4
```

The above output was returned by an SP-PCI card. The output returned by your system will likely vary based on the I/O range and IRQ of the port. However, it is important to note that in the output above the resources assigned to /dev/ttyS1 match the resources collected earlier and the UART type matches the UART present on the card, indicating that the LAVA card has been installed correctly on ttyS1. If the resources are correct then the installation of the card has been successful and you can either obtain more detailed information on the port or attempt to test the port.

**NOTE:** If the settings do not match the resources collected earlier then it may be necessary to manually configure the card. If so, please refer to "APPENDIX B: Manually Setting Serial Port Resources" (see page 9).

To obtain more detailed information on the port you can use the setserial command again by entering:

```
# setserial –a /dev/ttyS1
```

This command should display:

```
/dev/ttyS1, Line 1, UART: 16550A, Port: 0x3000, IRQ: 21
        Baud_base: 115200, close_delay: 50, divisor: 0
        closing_wait: 3000
        Flags: spd_normal skip_test
```

### *TESTING:*

Information on testing serial ports can be found in: "APPENDIX D: Testing Your Installations" (see page 11).

### *INSTALLATION – Parallel Ports:*

At bootup the Linux hardware discovery utility should automatically detect parallel ports installed in the system and create both a parportX and lpX device in the system's /dev directory. The actual parport and lp numbers will depend on the number of parallel ports currently present on the system.

To verify the actual allocation of the ports you must first determine the I/O range and IRQ that have been assigned to each parallel port. To retrieve this information you will need to query the PCI bus for a list of all LAVA PCI devices present.

To query the PCI bus from the terminal enter:

```
# lspci –v –d 1407:*
```

Depending on the number of cards present in the system, and whether or not your card supports multiple ports, you may receive a list of several items detected on the PCI bus. However the output should resemble the example below:

```
03:00.0 Parallel controller: Lava Computer mfg Inc … … …
          Subsystem: Lava Computer mfg Inc … … …
          Flags: medium devsel, IRQ 20
          I/O ports at cf00 [size=8]
          Kernel driver in use: parport_pc
03:00.1 Serial controller: Lava Computer mfg Inc … … [16550])
          Subsystem: Lava Computer mfg Inc … … …
          Flags: medium devsel, IRQ 21
          I/O ports at 3000 [size=8]
          Kernel driver in use: serial
```

The example above was returned by the SP-PCI card where the parallel port has been configured for IRQ 20 and an I/O range of 0xCF00. Every system is different and therefore will return a different IRQ and I/O range; however, the important thing to obtain from the command is the IRQ and I/O range of all ports on the card.

**NOTE:** The SP-PCI and 2SP-PCI cards will return the resources for both the parallel and serial ports present on the card; however, at the present we are only interested in the resources of the parallel port.

Once the IRQ and I/O range of all ports have been determined you will need to verify that all the ports have been installed correctly. Any parallel port installed on the system will be located in the /dev directory and can be listed from the terminal by entering:

```
# ls /dev/parport*
```

This command should display:

```
/dev/parport0       /dev/parport1
```

In addition each port should also have a line printer associated with it in the /dev directory that can be viewed by entering:

```
# ls /dev/lp*
```

This command should display:

```
/dev/lp0     /dev/lp1
```

Now, given the IRQ and I/O range, as well as the list of ports present, you can verify if the port was installed automatically by viewing the kernel's message buffer, by entering:

```
# dmesg | grep par
```

This command should display, among other things:

```
… … …
parport_pc 00:0a: reported by Plug and Play ACPI
parport0: PC-style at 0x378, irq 7 [PCSPP,TRISTATE]
PCI Parallel port detected: 1407:8000, I/O at 0xcf00(0x0), IRQ 20
parport1: PC-style at 0xcf00, irq 20 [PCSPP,TRISTATE,EPP]
parport1: Printer, Hewlett-Packard HP LaserJet 5MP
lp0: using parport0 (interrupt-driven).
lp1: using parport1 (interrupt-driven).
… … …
```

The above output was returned by the SP-PCI card. The output returned by your system will likely vary based on the I/O range and IRQ of the port. However, it is important to note that in the output above the resources assigned to /dev/parport1 match the resources collected earlier, indicating that the LAVA card has been installed correctly on parport1. If the resources are correct then the installation of the card has been successful and you may begin using the port.

**NOTE:** If the port was not automatically detected by the system or the settings don't match the resources collected earlier then it may be necessary to manually remove and re-add all parallel ports. If this is the case then please refer to "APPENDIX C: Manually Adding Parallel Ports" (see page 10).

### *TESTING:*

Information on testing parallel ports can be found in: "APPENDIX D: Testing Your Installations" (see page 11).

## APPENDIX A: Manually Adding Serial Ports

Certain distributions of Linux may limit default the number serial ports created at boot time. This restriction usually limits the number to four ttyS devices. In those cases if you require additional ttyS ports to be created then it will be necessary to reconfigure the system in order to create additional ports. This can be done either manually or automatically.

### *Manually Creating Ports:*

Once the system has booted it's possible to create additional ttyS devices by using the setserial command.

Now, assuming ttyS4 hasn't already been created by the system, we can create a fifth port by entering the following command:

```
# setserial /dev/ttyS4 port 0x3000 irq 21 uart 16550a baud_base 115200
```

If the setserial command was executed without error then a listing of the ttyS devices (ls /dev/ttyS*) as well as a query of serial settings (**`setserial –g /dev/ttyS4`**) should return results very similar to those set out in the previous installation section.

It should be noted that this port will need to be created every time the system boots up, though it is possible to automate the creation of this port by adding the setserial command to the system's local startup script; however, should the card's resources change then the command in the startup script will also need to modified to reflect the change.

### *Automatically Adding Ports:*

An alternative to manually creating the ports is to configure the system to increase the number of default ports created at boot time, which can either be done when compiling the kernel or by overriding the system defaults at boot time (i.e. boot-prompt or kernel command).

The simplest of these options is to override the system defaults via the kernel command line (e.g. **`8250.nr_uarts=x`**; where: x=the total number of ports to be created on boot) which can be enabled through your system's boot loader.

The benefit to increasing the number of default ports is that the system should automatically detect the addition of, or changes to, any PCI cards in the system and enumerate them automatically.

## APPENDIX B: Manually Setting Serial Port Resources

Once the system has booted it's possible to manually set the resources of an existing ttyS devices by using the setserial command.

Using the information from our previous query of the PCI bus (i.e. I/O range=0x3000, IRQ=21 and UART=16550a), we can modify the fourth port by entering the following command:

```
# setserial /dev/ttyS3 port 0x3000 irq 21 uart 16550a baud_base 115200
```

If the setserial command was executed without error then a query of serial settings (**`setserial –g /dev/ttyS3`**) should return results very similar to those set out in the previous installation section.

It should be noted that the port resources will need to be set every time the system boots up, though it is possible to automate the creation of this port by adding the setserial command to the system's local startup script. However, should the card's resources change then the command in the startup script will also need to modified to reflect the change.

## APPENDIX C: Manually Adding Parallel Ports

In certain situations the system may not detected the presence of an additional parallel port automatically and therefore it may be necessary to manually add parallel ports after the system has booted. To add ports you must first remove both the lp and paraport_pc modules associated with the existing parallel ports on the system. This can by done from the terminal by entering:

```
# modprobe –r lp
# modprobe –r parport_pc
```

Now, utilizing the information returned from our previous query of the PCI bus (i.e. I/O range=0xcf00 and IRQ=20) as well as the information returned from the kernel's message buffer regarding the onboard parallel port (i.e. I/O range=0x378 and IRQ=7), we can manually re-add both parallel ports by entering:

```
# modprobe parport_pc io=0x378,0xcf00 irq=7,20
# modprobe lp
```

**NOTE:** The order that the ports are enumerated will follow the order provided in the I/O range and IRQ fields of the previous command.

Assuming the previous commands were executed without any errors the system will create the requested number of parallel ports and corresponding line printers in the /dev directory, which can list be listed by entering:

```
# ls /dev/lp* /dev/parport*
```

This should display:

```
/dev/lp0     /dev/lp1     /dev/parport0      /dev/parport1
```

It should be noted that the parallel port resources will need to be set every time the system boots up, though it is possible to automate the creation of this port by adding the above commands to the system's local startup script. However should the card's resources change then the command in the startup script will also need to be modified to reflect the changes.

## APPENDIX D: Testing Your Installations

### *Testing Serial*

To verify the serial port is working correctly you can use a terminal program and either a loopback connector or a serial cable attached to any appropriate serial device.

For the purpose of this example, Minicom was used along with a loopback connector attached to the serial port. The loopback connector simply shorts the Rx and Tx pins of the serial port's DB9 connector, allowing any data sent from the serial port to be "echoed" back and displayed in Minicom. If a loopback connector isn't available then any means of shorting the Rx and Tx pins (pins 2 and 3) will be sufficient.

If Minicom wasn't included in the base installation, it can be can be obtained from your distribution's package manager.

Once Minicom has been installed on the system you can open the application's setup configuration by entering:

```
# minicom –s
```

This will allow you to select which COM port and settings (i.e. baud rate, flow control, etc) you would like to use. These settings can be reached through the "Serial Port Setup" options.

Since this example uses a loopback connector, there is no need to configure the baud rate. However, the "Serial Device" will need to be configured to match the newly installed port and all "Flow Control" will need to be disabled.

Once all the setup and configuration changes have been made you can select "Exit", which will take you to the terminal interface. With the loopback connector attached to the serial port all data typed into the terminal should be echoed back. In addition, removing the loopback connector should prevent the echoing of data. At this point you've performed a fairly basic but sufficient test of the serial port and can exit Minicom by accessing the control menu (Ctrl+A) and selecting Exit (X).

### *Testing Parallel*

By far the best way to verify the parallel port is working correctly is to connect a printer to the parallel port. In the example above a printer was connected the parallel port before the system was powered up and as a result detection of the printer was displayed as being connected to parport1 in the message buffer as follows:

```
… … …
parport1: Printer, Hewlett-Packard HP LaserJet 5MP
… … …
```

Regardless of whether or not the printer was present at boot time the system should be able to detect the presence of a printer, assuming the port is correctly configured, and as such should be able to print a test page.

## APPENDIX E: Listing of Tested Operating Systems

| LINUX Family | Distribution Release | Kernel | ISO Image Tested |
|---|---|---|---|
| Debian | Ubuntu 13.10 | 3.11.0-12-generic | ubuntu-13.10-desktop-i386.iso |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| Fedora | Fedora 19 | 3.9.5-301.fc19.i686 | Fedora-Live-Desktop-i686-19-1.iso |
| | CentOS 6.5 | 2.6.32-431.el5.i686 | CentOS-6.5-i386-LiveCD.iso |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| openSUSE | openSUSE 13.1 | 3.11.6-4-default | openSUSE-13.1-KDE-Live-i686.iso |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |