Introduction

This Application Note covers how to load custom PCI/PCIe UART drivers, and install and run the Minicom test utility to test the UART ports using an Ubuntu Linux distribution. Custom drivers can be downloaded from www.exar.com/design-tools/software-drivers.

The following steps are required for the setting up and using Minicom:

1. Unbind the native serial driver.
2. Make and install the custom (MaxLinear) Linux driver.
3. Install Minicom.
5. Using Minicom / test the serial port.

Step 1: Unbind the native serial driver

Step 1.1: Enable Root Mode
Enable the root mode by typing in the following command and the root password:

```
exas@exas-desktop:~$ sudo -i
```

Step 1.2: Native Serial Driver
In recent Linux kernels, there is a native “serial” driver built in. This driver is loaded by default. It can be determined if it is loaded by using lspci:

```
lspci -vd 13a8:
```

Upon running the above command, something like this will be displayed if an XR17xxx device is present:

```
03:00.0 Serial controller: Exar Corp. Device 0358 (rev 03) (prog-if 02 [16550])
    Flags: fast devsel, IRQ 30
    Memory at f3ffc000 (32-bit, non-prefetchable) [size=16K]
    Capabilities: [50] MSI: Enable- Count=1/1 Maskable- 64bit+
    Capabilities: [78] Power Management version 3
    Capabilities: [80] Express Endpoint, MSI 01
    Capabilities: [100] Virtual Channel
    Kernel driver in use: serial (See Note 1)
```

**NOTE:**
1. In newer kernels, “exar_serial” may be seen and need to be used instead of “serial”.

The native serial driver should be sufficient for basic operation (transmitting and receiving data). When the serial driver is loaded, the ports in the XR17V35x device will be listed in /dev as ttySn.
Step 1.3: Unbind Native Serial Driver
To use the custom driver, the easiest mechanism is to unbind the serial driver by entering:

```
ls /sys/bus/pci/drivers/serial/
```

This command returns something like:

```
0000:03:00.0 bind new_id remove_id uevent unbind
```

Use:
```
echo -n "0000:03:00.0" > /sys/bus/pci/drivers/serial/unbind
```

(replacing value in quotes with value from ls /sys/bus/pci/drivers/serial/)

NOTE:
1. In newer kernels, “exar_serial” may be seen and need to be used instead of “serial”.

Step 2: Make and install the custom (MaxLinear) Linux driver

Step 2.1: Make and Install Driver
Make and install the driver per the readme file included with the driver.
With the custom driver, the ports in the XR17V35x device will be listed in /dev as ttyXRn.

Step 2.2: List TTY Devices
List all of the tty devices by typing in the following command. See Figure 1 as an example:
```
root@exar-desktop:~# ls /dev/tty*
```

Figure 1: Example List of TTY Devices
Step 2.3: Compilling and Installing Drivers
Compile the XR17V35x drivers from the directory containing the driver files and install them using the insmod command:

In Figure 2 below:
- `ls` lists the driver folder contents
- `make` makes the driver
- `insmod ./xr17v35x.ko` installs the driver

```
root@exar-desktop:~/exar/Desktop/xr17v25x_35x-lnx3.x-x-pak# ls
Makefile readme xr17v35x.c
root@exar-desktop:~/exar/Desktop/xr17v25x_35x-lnx3.x-x-pak# make
make -C /lib/modules/`uname -r`/build SUBDIRS=/home/exar/Desktop/xr17v25x_35x-lnx3.x-x-pak modules
make[1]: Entering directory `/usr/src/linux-headers-3.2.0-43-generic`
  CC [M] `home/exar/Desktop/xr17v25x_35x-lnx3.x-x-pak/xr17v35x.o`
  CC [M] `home/exar/Desktop/xr17v25x_35x-lnx3.x-x-pak/xr17v35x.c`
make[1]: Warning: `int_offset' defined but not used [-Wunused-variable]
  Building modules, stage 2.
  MODPOST 1 modules
  CC `home/exar/Desktop/xr17v25x_35x-lnx3.x-x-pak/xr17v35x.mod.o`
  LD [M] `home/exar/Desktop/xr17v25x_35x-lnx3.x-x-pak/xr17v35x.ko`
make[1]: Leaving directory `/usr/src/linux-headers-3.2.0-43-generic`
root@exar-desktop:~/exar/Desktop/xr17v25x_35x-lnx3.x-x-pak# insmod ./xr17v35x.ko
```

Figure 2: Driver Installation

Step 2.4: XR17v35x devices in /dev/tty*
Confirm that the driver is loaded for the XR17V35x by running ls /dev/ttyX* as shown in Figure 3:

```
root@exar-desktop:~/exar/Desktop/xr17v25x_35x-lnx3.x-x-pak# ls /dev/ttyX*
/dev/ttyXR0  /dev/ttyXR1  /dev/ttyXR4  /dev/ttyXR3  /dev/ttyXR6  /dev/ttyXR9
/dev/ttyXR1  /dev/ttyXR2  /dev/ttyXR15 /dev/ttyXR4  /dev/ttyXR7  /dev/ttyXR10
/dev/ttyXR10 /dev/ttyXR13 /dev/ttyXR2  /dev/ttyXR5  /dev/ttyXR8
```

Figure 3: Searching for XR Devices

Step 3: Install and Run Minicom

Step 3.1: Install Minicom

```
sudo apt-get install minicom
```

Step 3.2: Run Minicom Utility
To test the first PCIe UART port, type in the Minicom command per Figure 4.

```
root@exar-desktop:~/exar/Desktop/xr17v25x_35x-lnx3.x-x-pak# minicom -s /dev/ttyXR0
```

Figure 4: Running the Minicom Utility
Step 4: Configure Minicom

Step 4.1: Serial Port Setup
In the Minicom configuration menu shown in Figure 5, scroll to “Serial port setup” using the ↓ or ↑ keys and then press enter.

![Configuration Menu](image)

Figure 5: Configuration Menu

Step 4.2: Settings
The settings can be changed as desired by typing the corresponding letter on the left as shown in Figure 6. For example, to select the desired serial port, select ‘a’ or ‘A’ (not case-sensitive) or to change the baud rate, select ‘e’ or ‘E’. After the selection, ‘Enter’ must be pressed.

![Settings Menu](image)

Figure 6: Settings Menu

Note in Figure 6 that port 0, 115200 bps, 8-bit character, no parity, 1 stop bit, hardware flow and no software flow control are selected. For an example of changing these settings, press F to disable Hardware Flow Control. Then see that the Hardware Flow Control is automatically disabled as shown in Figure 7. Press ‘Enter’ to exit this menu and return to the configuration menu.

![Hardware Flow Control (F) Disabled](image)

Figure 7: Hardware Flow Control (F) Disabled
Step 4.3: Saving the Settings (optional)
To save the settings, now scroll to Save setup as shown in Figure 8.

![Figure 8: Saving the Settings](image)

Step 4.4: Opening the Port
Once the settings are saved, select ‘Exit’ from the menu. The port has been successfully configured and opened, and is ready to send and receive data.

![Figure 9: Welcome to Minicom](image)

Note that CTRL A Z for the help menu can be used. For example:

A sets the line wrap
X exits from Minicom

Step 5: Using Minicom / Test Serial Port

Serial Port Testing Examples:
Example 1: Loopback testing on a single port
With a loopback connector on a serial port, typed / transmitted data is echoed back in the same Minicom window.
Example 2: Testing between different ports
With a null modem connection between two serial ports, typed / transmitted data in one Minicom window is received in a second Minicom window and vice versa.
Example 3: Testing with a remote UART
With a null modem connection between one of the local serial ports and a remote serial port, typed / transmitted data in the Minicom window will be transmitted to the remote UART. Data received from the remote UART will be displayed in the Minicom window.
Revision History

<table>
<thead>
<tr>
<th>Revision</th>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1A</td>
<td>5/1/18</td>
<td>Initial release</td>
</tr>
<tr>
<td>1B</td>
<td>7/31/18</td>
<td>Updated title to include PCI.</td>
</tr>
</tbody>
</table>