

# Adding Support for 16CXXX Based Devices to Linux

Linux already supports the direct use of 16CXXX UART based devices with drivers included in the kernel sources. Some of Sealevel's async serial devices do not currently have their vendor/product IDs included in this driver. It is necessary to either manually identify the UARTs to the driver or to modify the driver by adding the vendor/product IDs.

## Option 1: Using the setserial command line utility.

### NOTES

- This procedure is the ONLY method of accessing a PC/104 or ISA device by using standard device drivers.
- Since the <IO\_ADDRESS> and <IRQ\_NUMBER> are static in the case of PC/104 and ISA, a distribution's start-up scripts can easily be modified to enable driver attachment (this assumes that no new plug-and-play serial devices will be added later on, as they will be overwritten until the start-up scripts are modified).
- Unfortunately a recent change in the way the serial driver attaches to <MINOR> numbers may prevent more than 4 serial ports to be attached to the driver at any time. This can be remedied by changing the CONFIG\_SERIAL\_8250\_NR\_UARTS configuration variable and recompiling the kernel module (which means if this is a PCI device, it may be prudent to go ahead and enable Plug-and-Play capability)

The setserial program is an open source utility to aide in the configuration of the 8250 async serial port driver. It is included with most Linux distributions or available through that distributions package management system (check distribution documentation). Alternatively the source code can be found at the [project's website](#).

Once the setserial utility is available, it is necessary to determine the IO addresses and IRQ line for each of the devices UARTs. If this is an ISA or PC/104 device, these options are configured manually with jumpers on the board. If this is a PCI, PCI Express, or PC/104+ device, these are assigned through a combination of BIOS and OS. To determine what IO address and IRQ line has been assigned, use the lspci command as demonstrated. The d switch filters results to any devices with IDs VENDOR:PRODUCT. By leaving the PRODUCT ID blank, ALL Sealevel devices attached to the PCI bus will be displayed. The vvv option tells the command to be as verbose as possible giving us all the information we will need.

```
$ lspci -d 135e: -vvv
```

```
00:0f.0 Serial controller: Sealevel Systems Inc Unknown device 7203 (rev 01) (prog-if 02 16550)
Subsystem: Sealevel Systems Inc Unknown device 7203
Flags: medium devsel, IRQ 11
Memory at ee000000 (32-bit, non-prefetchable) [size=128]
I/O ports at b000 [size=128]
I/O ports at a800 [size=8]
I/O ports at a400 [size=8]
```

```
00:11.0 Serial controller: Sealevel Systems Inc Unknown device 7803 (rev 01) (prog-if 02 16550)
Subsystem: Sealevel Systems Inc Unknown device 7803
Flags: medium devsel, IRQ 10
Memory at ed800000 (32-bit, non-prefetchable) [size=128]
I/O ports at a000 [size=128]
I/O ports at 9800 [size=64]
```

The <IRQ\_NUMBER> can easily be extracted from the output of the lspci command. Next to each device is listed the assigned IRQ. This is one of the two pieces of information we need!

The next piece of information can be extracted from the PCI Regions listed in the lspci

output. Each PCI device is a little different and a certain amount of thinking must be done to determine which region(s) are mapped to the devices UARTs.

- Most Sealevel devices with 4 or less ports use individual PCI regions to map each UART's 8 IO registers. So look for IO regions that have sizes of 8. There should be X number of size 8 regions (where X is the number of UARTS on the device).
- Many newer devices and 16 port devices use one large PCI region to map all of a devices UARTs IO registers. So look for IO regions that have sizes of the product of the number of UARTs and 8.
- This information is usually contained within the device manual, but sometimes it is not. If you are unable to determine the correct IO space, please contact Sealevel's Technical Support for aid.

Once you know which region(s) correspond to which UARTs, you are interested in the <IO\_ADDRESS>. This is a 4 digit hexadecimal number.

Now that all the necessary information has been gathered, it is time to use the setserial utility to attach a device to the 8250 driver. Each UART must be assigned, one at a time, using setserial.

```
# setserial /dev/ttyS# port <IO_ADDRESS> irq <IRQ_NUMBER> baud_base X
* The baud_base parameter is very, very important. Without a value > 0, the
setserial command will fail. The proper baud_base is dependent on the particular
piece of hardware being used. The product manual should be consulted to identify the
proper baud_base.
```

The /dev/ttyS# (where # is an integer) must correspond to an existing device file. If there aren't enough device files, they can be created with the following command. The <MAJOR> is determined by the 8250 driver, and the <MINOR> is the next incremental integer. (\$ ls -al /dev/ttyS#-1 to see what they should be)

```
# mknod /dev/ttyS# c <MAJOR> <MINOR>
```

## Option 2: Modify 8250 driver to include Sealevel Product/Vendor IDs.

### NOTES

- This method requires kernel sources (NOT HEADERS). These can usually be obtained through the distribution's package management system. Alternatively the vanilla kernel sources can be downloaded from [kernel.org](http://kernel.org). Recompiling the Linux kernel goes beyond the scope of this document, and may be different depending on the distribution used. Consult the distribution's documentation concerning module building.
- This method could also be necessary if more than 4 8250 serial ports must be used at the same time. This requires modifying the kernel's configuration to support more ports.

Navigate to the directory containing kernel sources (usually /usr/src/linux).

First verify that the desired device is not already supported by the 8250 driver. Open the header file pci\_ids.h located in the include/linux subdirectory. Find the keyword "SEALEVEL". There will be a list of definitions assigning macro names like "PCI\_DEVICE\_ID\_SEALEVEL\_\*" to hexadecimal Sealevel product IDs. Scan through the list looking for your device's product number. If it is listed, then that device is already supported by the 8250 driver. If it is not listed, make a new entry with a unique macro name and the corresponding product ID.

```
.../include/linux/pci_ids.h
#define PCI_DEVICE_ID_SEALEVEL_COMM8    0x7801    //existing entry
#define PCI_DEVICE_ID_SEALEVEL_UCOMM8   0x7804    //new entry
```

Next add the device to the 8250's device detection code. Edit the 8250\_pci.c file located in the drivers/serial/ subdirectory. Again, find the keyword "SEALEVEL". There will be an array of pci\_device structs. Scroll to the last entry with a SEALEVEL vendor macro. Make a new entry using the product ID that was created in the previous step. The last element of this entry will be a descriptor to a specific board configuration which is hardware dependent.

Contact engineering for exact instructions to add a specific product's configuration.

```
.../drivers/serial/8250_pci.c
{PCI_VENDOR_ID_SEALEVEL, PCI_DEVICE_ID_SEALEVEL_COMM8, PCI_ANY_ID,
PCI_ANY_ID, 0, 0, pbn_b2_8_115200 },
{PCI_VENDOR_ID_SEALEVEL, PCI_DEVICE_ID_SEALEVEL_UCOMM8, PCI_ANY_ID,
PCI_ANY_ID, 0, 0, pbn_b2_8_115200 },
```

Next verify that a reasonable number of 8250 serial ports can be allocated by the device driver. Edit the .config file located in the root of the kernel source directory. Search for the keyword "CONFIG\_SERIAL\_8250". And verify that appropriate values are used for each variable.

```
.../.config
CONFIG_SERIAL_8250_NR_UARTS=48
CONFIG_SERIAL_8250_RUNTIME_UARTS=48
```

Now recompile the driver module, remove the old module, and insert the new one.