

stallion.txt

* NOTE - This is an unmaintained driver. Lantronix, which bought Stallion technologies, is not active in driver maintenance, and they have no information on when or if they will have a 2.6 driver.

James Nelson <james4765@gmail.com> - 12-12-2004

Stallion Multiport Serial Driver Readme

Copyright (C) 1994-1999, Stallion Technologies.

Version: 5.5.1
Date: 28MAR99

1. INTRODUCTION

There are two drivers that work with the different families of Stallion multiport serial boards. One is for the Stallion smart boards - that is EasyIO, EasyConnection 8/32 and EasyConnection 8/64-PCI, the other for the true Stallion intelligent multiport boards - EasyConnection 8/64 (ISA, EISA, MCA), EasyConnection/RA-PCI, ONboard and Brumby.

If you are using any of the Stallion intelligent multiport boards (Brumby, ONboard, EasyConnection 8/64 (ISA, EISA, MCA), EasyConnection/RA-PCI) with Linux you will need to get the driver utility package. This contains a firmware loader and the firmware images necessary to make the devices operate.

The Stallion Technologies ftp site, [ftp.stallion.com](ftp://ftp.stallion.com), will always have the latest version of the driver utility package.

<ftp://ftp.stallion.com/drivers/ata5/Linux/ata-linux-550.tar.gz>

As of the printing of this document the latest version of the driver utility package is 5.5.0. If a later version is now available then you should use the latest version.

If you are using the EasyIO, EasyConnection 8/32 or EasyConnection 8/64-PCI boards then you don't need this package, although it does have a serial stats display program.

If you require DIP switch settings, EISA or MCA configuration files, or any other information related to Stallion boards then have a look at Stallion's web pages at <http://www.stallion.com>.

2. INSTALLATION

The drivers can be used as loadable modules or compiled into the kernel. You can choose which when doing a "config" on the kernel.

All ISA, EISA and MCA boards that you want to use need to be configured into the driver(s). All PCI boards will be automatically detected when you load the driver - so they do not need to be entered into the driver(s)

stallion.txt

configuration structure. Note that kernel PCI support is required to use PCI boards.

There are two methods of configuring ISA, EISA and MCA boards into the drivers. If using the driver as a loadable module then the simplest method is to pass the driver configuration as module arguments. The other method is to modify the driver source to add configuration lines for each board in use.

If you have pre-built Stallion driver modules then the module argument configuration method should be used. A lot of Linux distributions come with pre-built driver modules in /lib/modules/X.Y.Z/misc for the kernel in use. That makes things pretty simple to get going.

2.1 MODULE DRIVER CONFIGURATION:

The simplest configuration for modules is to use the module load arguments to configure any ISA, EISA or MCA boards. PCI boards are automatically detected, so do not need any additional configuration at all.

If using EasyIO, EasyConnection 8/32 ISA or MCA, or EasyConnection 8/63-PCI boards then use the "stallion" driver module. Otherwise if you are using an EasyConnection 8/64 ISA, EISA or MCA, EasyConnection/RA-PCI, ONboard, Brumby or original Stallion board then use the "istallion" driver module.

Typically to load up the smart board driver use:

```
modprobe stallion
```

This will load the EasyIO and EasyConnection 8/32 driver. It will output a message to say that it loaded and print the driver version number. It will also print out whether it found the configured boards or not. These messages may not appear on the console, but typically are always logged to /var/adm/messages or /var/log/syslog files - depending on how the klogd and syslogd daemons are setup on your system.

To load the intelligent board driver use:

```
modprobe istallion
```

It will output similar messages to the smart board driver.

If not using an auto-detectable board type (that is a PCI board) then you will also need to supply command line arguments to the modprobe command when loading the driver. The general form of the configuration argument is

```
board?=<name>[,<ioaddr>[,<addr>]][,<irq>]]
```

where:

board? -- specifies the arbitrary board number of this board, can be in the range 0 to 3.

name -- textual name of this board. The board name is the common board name, or any "shortened" version of that. The board type number may also be used here.

stallion.txt

- ioaddr -- specifies the I/O address of this board. This argument is optional, but should generally be specified.
- addr -- optional second address argument. Some board types require a second I/O address, some require a memory address. The exact meaning of this argument depends on the board type.
- irq -- optional IRQ line used by this board.

Up to 4 board configuration arguments can be specified on the load line. Here is some examples:

```
modprobe stallion board0=easyio,0x2a0,5
```

This configures an EasyIO board as board 0 at I/O address 0x2a0 and IRQ 5.

```
modprobe istallion board3=ec8/64,0x2c0,0xcc000
```

This configures an EasyConnection 8/64 ISA as board 3 at I/O address 0x2c0 at memory address 0xcc000.

```
modprobe stallion board1=ec8/32-at,0x2a0,0x280,10
```

This configures an EasyConnection 8/32 ISA board at primary I/O address 0x2a0, secondary address 0x280 and IRQ 10.

You will probably want to enter this module load and configuration information into your system startup scripts so that the drivers are loaded and configured on each system boot. Typically the start up script would be something like /etc/modprobe.conf.

2.2 STATIC DRIVER CONFIGURATION:

For static driver configuration you need to modify the driver source code. Entering ISA, EISA and MCA boards into the driver(s) configuration structure involves editing the driver(s) source file. It's pretty easy if you follow the instructions below. Both drivers can support up to 4 boards. The smart card driver (the stallion.c driver) supports any combination of EasyIO and EasyConnection 8/32 boards (up to a total of 4). The intelligent driver supports any combination of ONboards, Brumbys, Stallions and EasyConnection 8/64 (ISA and EISA) boards (up to a total of 4).

To set up the driver(s) for the boards that you want to use you need to edit the appropriate driver file and add configuration entries.

If using EasyIO or EasyConnection 8/32 ISA or MCA boards,

In drivers/char/stallion.c:

- find the definition of the stl_brdconf array (of structures) near the top of the file
- modify this to match the boards you are going to install (the comments before this structure should help)
- save and exit

If using ONboard, Brumby, Stallion or EasyConnection 8/64 (ISA or EISA)

stallion.txt

boards,

In drivers/char/istallion.c:

- find the definition of the stli_brdconf array (of structures) near the top of the file
- modify this to match the boards you are going to install (the comments before this structure should help)
- save and exit

Once you have set up the board configurations then you are ready to build the kernel or modules.

When the new kernel is booted, or the loadable module loaded then the driver will emit some kernel trace messages about whether the configured boards were detected or not. Depending on how your system logger is set up these may come out on the console, or just be logged to /var/adm/messages or /var/log/syslog. You should check the messages to confirm that all is well.

2.3 SHARING INTERRUPTS

It is possible to share interrupts between multiple EasyIO and EasyConnection 8/32 boards in an EISA system. To do this you must be using static driver configuration, modifying the driver source code to add driver configuration. Then a couple of extra things are required:

1. When entering the board resources into the stallion.c file you need to mark the boards as using level triggered interrupts. Do this by replacing the "0" entry at field position 6 (the last field) in the board configuration structure with a "1". (This is the structure that defines the board type, I/O locations, etc. for each board). All boards that are sharing an interrupt must be set this way, and each board should have the same interrupt number specified here as well. Now build the module or kernel as you would normally.
2. When physically installing the boards into the system you must enter the system EISA configuration utility. You will need to install the EISA configuration files for *all* the EasyIO and EasyConnection 8/32 boards that are sharing interrupts. The Stallion EasyIO and EasyConnection 8/32 EISA configuration files required are supplied by Stallion Technologies on the EASY Utilities floppy diskette (usually supplied in the box with the board when purchased. If not, you can pick it up from Stallion's FTP site, ftp.stallion.com). You will need to edit the board resources to choose level triggered interrupts, and make sure to set each board's interrupt to the same IRQ number.

You must complete both the above steps for this to work. When you reboot or load the driver your EasyIO and EasyConnection 8/32 boards will be sharing interrupts.

2.4 USING HIGH SHARED MEMORY

The EasyConnection 8/64-EI, ONboard and Stallion boards are capable of using shared memory addresses above the usual 640K - 1Mb range. The ONboard ISA and the Stallion boards can be programmed to use memory addresses up to

stallion.txt

16Mb (the ISA bus addressing limit), and the EasyConnection 8/64-EI and ONboard/E can be programmed for memory addresses up to 4Gb (the EISA bus addressing limit).

The higher than 1Mb memory addresses are fully supported by this driver. Just enter the address as you normally would for a lower than 1Mb address (in the driver's board configuration structure).

2.5 TROUBLE SHOOTING

If a board is not found by the driver but is actually in the system then the most likely problem is that the I/O address is wrong. Change the module load argument for the loadable module form. Or change it in the driver stallion.c or istallion.c configuration structure and rebuild the kernel or modules, or change it on the board.

On EasyIO and EasyConnection 8/32 boards the IRQ is software programmable, so if there is a conflict you may need to change the IRQ used for a board. There are no interrupts to worry about for ONboard, Brumby or EasyConnection 8/64 (ISA, EISA and MCA) boards. The memory region on EasyConnection 8/64 and ONboard boards is software programmable, but not on the Brumby boards.

3. USING THE DRIVERS

3.1 INTELLIGENT DRIVER OPERATION

The intelligent boards also need to have their "firmware" code downloaded to them. This is done via a user level application supplied in the driver utility package called "stlload". Compile this program wherever you dropped the package files, by typing "make". In its simplest form you can then type

```
./stlload -i cdk.sys
```

in this directory and that will download board 0 (assuming board 0 is an EasyConnection 8/64 or EasyConnection/RA board). To download to an ONboard, Brumby or Stallion do:

```
./stlload -i 2681.sys
```

Normally you would want all boards to be downloaded as part of the standard system startup. To achieve this, add one of the lines above into the /etc/rc.d/rc.S or /etc/rc.d/rc.serial file. To download each board just add the "-b <brd-number>" option to the line. You will need to download code for every board. You should probably move the stlload program into a system directory, such as /usr/sbin. Also, the default location of the cdk.sys image file in the stlload down-loader is /usr/lib/stallion. Create that directory and put the cdk.sys and 2681.sys files in it. (It's a convenient place to put them anyway). As an example your /etc/rc.d/rc.S file might have the following lines added to it (if you had 3 boards):

```
/usr/sbin/stlload -b 0 -i /usr/lib/stallion/cdk.sys  
/usr/sbin/stlload -b 1 -i /usr/lib/stallion/2681.sys
```

```
stallion.txt
/usr/sbin/stlload -b 2 -i /usr/lib/stallion/2681.sys
```

The image files `cdk.sys` and `2681.sys` are specific to the board types. The `cdk.sys` will only function correctly on an EasyConnection 8/64 board. Similarly the `2681.sys` image will only operate on ONboard, Brumby and Stallion boards. If you load the wrong image file into a board it will fail to start up, and of course the ports will not be operational!

If you are using the modularized version of the driver you might want to put the `modprobe` calls in the startup script as well (before the download lines obviously).

3.2 USING THE SERIAL PORTS

Once the driver is installed you will need to setup some device nodes to access the serial ports. The simplest method is to use the `/dev/MAKEDEV` program. It will automatically create device entries for Stallion boards. This will create the normal serial port devices as `/dev/ttyE#` where `#` is the port number starting from 0. A bank of 64 minor device numbers is allocated to each board, so the first port on the second board is port 64, etc. A set of callout type devices may also be created. They are created as the devices `/dev/cue#` where `#` is the same as for the `ttyE` devices.

For the most part the Stallion driver tries to emulate the standard PC system COM ports and the standard Linux serial driver. The idea is that you should be able to use Stallion board ports and COM ports interchangeably without modifying anything but the device name. Anything that doesn't work like that should be considered a bug in this driver!

If you look at the driver code you will notice that it is fairly closely based on the Linux serial driver (`linux/drivers/char/serial.c`). This is intentional, obviously this is the easiest way to emulate its behavior!

Since this driver tries to emulate the standard serial ports as much as possible, most system utilities should work as they do for the standard COM ports. Most importantly `"stty"` works as expected and `"setserial"` can also be used (excepting the ability to auto-configure the I/O and IRQ addresses of boards). Higher baud rates are supported in the usual fashion through `setserial` or using the CBAUDEX extensions. Note that the EasyIO and EasyConnection (all types) support at least 57600 and 115200 baud. The newer EasyConnection XP modules and new EasyIO boards support 230400 and 460800 baud as well. The older boards including ONboard and Brumby support a maximum baud rate of 38400.

If you are unfamiliar with how to use serial ports, then get the `Serial-HOWTO` by Greg Hankins. It will explain everything you need to know!

4. NOTES

You can use both drivers at once if you have a mix of board types installed in a system. However to do this you will need to change the major numbers used by one of the drivers. Currently both drivers use major numbers 24, 25 and 28 for their devices. Change one driver to use some other major numbers,

stallion.txt

and then modify the mkdevnodes script to make device nodes based on those new major numbers. For example, you could change the istallion.c driver to use major numbers 60, 61 and 62. You will also need to create device nodes with different names for the ports, for example ttyF# and cuf#.

The original Stallion board is no longer supported by Stallion Technologies. Although it is known to work with the istallion driver.

Finding a free physical memory address range can be a problem. The older boards like the Stallion and ONboard need large areas (64K or even 128K), so they can be very difficult to get into a system. If you have 16 Mb of RAM then you have no choice but to put them somewhere in the 640K -> 1Mb range. ONboards require 64K, so typically 0xd0000 is good, or 0xe0000 on some systems. If you have an original Stallion board, "V4.0" or Rev.0, then you need a 64K memory address space, so again 0xd0000 and 0xe0000 are good. Older Stallion boards are a much bigger problem. They need 128K of address space and must be on a 128K boundary. If you don't have a VGA card then 0xc0000 might be usable - there is really no other place you can put them below 1Mb.

Both the ONboard and old Stallion boards can use higher memory addresses as well, but you must have less than 16Mb of RAM to be able to use them. Usual high memory addresses used include 0xec0000 and 0xf00000.

The Brumby boards only require 16Kb of address space, so you can usually squeeze them in somewhere. Common addresses are 0xc8000, 0xcc000, or in the 0xd0000 range. EasyConnection 8/64 boards are even better, they only require 4Kb of address space, again usually 0xc8000, 0xcc000 or 0xd0000 are good.

If you are using an EasyConnection 8/64-EI or ONboard/E then usually the 0xd0000 or 0xe0000 ranges are the best options below 1Mb. If neither of them can be used then the high memory support to use the really high address ranges is the best option. Typically the 2Gb range is convenient for them, and gets them well out of the way.

The ports of the EasyIO-8M board do not have DCD or DTR signals. So these ports cannot be used as real modem devices. Generally, when using these ports you should only use the cueX devices.

The driver utility package contains a couple of very useful programs. One is a serial port statistics collection and display program - very handy for solving serial port problems. The other is an extended option setting program that works with the intelligent boards.

5. DISCLAIMER

The information contained in this document is believed to be accurate and reliable. However, no responsibility is assumed by Stallion Technologies Pty. Ltd. for its use, nor any infringements of patents or other rights of third parties resulting from its use. Stallion Technologies reserves the right to modify the design of its products and will endeavour to change the information in manuals and accompanying documentation accordingly.