
Руководство по установке мультипортов KRON-ISA в ОС Linux.

Установки мультипорта

Установите на плате переключками параметры:

- Тип: **HOSTESS** (регистр вектора прерывания 107, положительный)
- Базовый адрес: **100**
- IRQ: любой

ВНИМАНИЕ! Убедитесь, что мультипорт не конфликтует с другими устройствами по диапазону адресов (100 - 13F) и IRQ, а также что его IRQ в BIOS Setup вашего компьютера определен как принадлежащий шине ISA. В любом случае после установки стоит проверить мультипорт поставляемым тестом.

Установки Linux

1. В файле `/etc/rc.d/rc.S` или `rc.sysinit` (в разных дистрибутивах это может быть помещено в разные файлы) раскомментировать строку типа:

```
/bin/sh /etc/rc.d/rc.serial
```

2. В файле `/etc/rc.d/rc.serial` определить плату как **Voca Board BB-1008**:

```
# VocaBoard 8 port (BB-1008) (base address 0x100),  
# or two BB-1004's (base addresses 0x100 and 0x120)  
#  
${SETSERIAL} /dev/ttyS16 ${AUTO_IRQ} autoconfig ${STD_FLAGS}  
${SETSERIAL} /dev/ttyS17 ${AUTO_IRQ} autoconfig ${STD_FLAGS}  
${SETSERIAL} /dev/ttyS18 ${AUTO_IRQ} autoconfig ${STD_FLAGS}  
${SETSERIAL} /dev/ttyS19 ${AUTO_IRQ} autoconfig ${STD_FLAGS}  
${SETSERIAL} /dev/ttyS20 ${AUTO_IRQ} autoconfig ${STD_FLAGS}  
${SETSERIAL} /dev/ttyS21 ${AUTO_IRQ} autoconfig ${STD_FLAGS}  
${SETSERIAL} /dev/ttyS22 ${AUTO_IRQ} autoconfig ${STD_FLAGS}  
${SETSERIAL} /dev/ttyS23 ${AUTO_IRQ} autoconfig ${STD_FLAGS}
```

Обычно последняя строка `rc.serial` (вызов `setserial` для вывода информации об обнаруженных портах) выглядит следующим образом:

```
${SETSERIAL} -bg /dev/ttyS?
```

В этом случае информация будет выводиться только для портов `ttyS1-ttyS9`, а мультипорт (`ttyS16-ttyS23`) в этот диапазон не попадает. Поэтому в данной команде вместо символа "?" следует использовать "*":

```
${SETSERIAL} -bg /dev/ttyS*
```

3. Убедитесь, что в `/dev` присутствуют спецфайлы устройств `cua16 - cua23` и `ttyS16 - ttyS23`. Если нет, выполните:

```
# cd /dev  
# ./MAKEDEV ttyS16 ttyS17 ttyS18 ttyS19 ttyS20 ttyS21 ttyS22 ttyS23
```

4. Опишите входные (*dial-in*) терминальные линии в */etc/inittab*. В зависимости от типа *getty* описание может выглядеть по-разному. Рекомендуется использовать один из:

- **agetty** - самый простой и небольшой
- **getty-ps** - стандартный
- **mgetty** - самый "навороченный", необходим, если планируется использовать линию и как входную, и как выходную, с поддержкой множества протоколов (терминальный, PPP, SLIP, FIDO, факс ...)

Учтите, что обычно используемый для консоли *mingetty* применять на последовательных линиях **НЕЛЬЗЯ**.

В случае *agetty* фрагмент инициализации терминальных линий файла */etc/inittab* может выглядеть следующим образом:

```
# Serial lines

# Первые 4 порта подключены по нуль-модемной линии на скорости
# 115200 бод к терминалам типа linux
sa:2345:respawn:/sbin/agetty -L 115200 ttyS16 linux
sb:2345:respawn:/sbin/agetty -L 115200 ttyS17 linux
sc:2345:respawn:/sbin/agetty -L 115200 ttyS18 linux
sd:2345:respawn:/sbin/agetty -L 115200 ttyS19 linux

# Оставшиеся 4 порта подключены по полномодемной линии на скорости
# 38400 бод к терминалам типа DEC VT-100
se:2345:respawn:/sbin/agetty -h 38400 ttyS20 vt100
sf:2345:respawn:/sbin/agetty -h 38400 ttyS21 vt100
sg:2345:respawn:/sbin/agetty -h 38400 ttyS22 vt100
sh:2345:respawn:/sbin/agetty -h 38400 ttyS23 vt100
```

5. При начальной загрузке в случае обнаружения мультипорта будет выведена информация:

```
Configuring serial ports....
/dev/ttyS0 at 0x03f8 (irq = 4) is a 16550A COM1
/dev/ttyS1 at 0x02f8 (irq = 3) is a 16550A COM2
/dev/ttyS16 at 0x0100 (irq = 12) is a 16550A Порт 1
/dev/ttyS17 at 0x0108 (irq = 12) is a 16550A Порт 2
/dev/ttyS18 at 0x0110 (irq = 12) is a 16550A Порт 3
/dev/ttyS19 at 0x0118 (irq = 12) is a 16550A Порт 4
/dev/ttyS20 at 0x0120 (irq = 12) is a 16550A Порт 5
/dev/ttyS21 at 0x0128 (irq = 12) is a 16550A Порт 6
/dev/ttyS22 at 0x0130 (irq = 12) is a 16550A Порт 7
/dev/ttyS23 at 0x0138 (irq = 12) is a 16550A Порт 8
```

Дополнительные настройки Linux

В большинстве случаев выполнения указанных действий достаточно для обеспечения нормальной работы мультипорта в ОС *Linux*. Однако при интенсивном обмене данными по последовательным портам могут иметь место сбои, вызванные так называемыми "потерями прерывания". В некоторой степени это явление сглаживается наличием встроенного в *UART FIFO*-буфера, и чем он больше - тем меньше вероятность возникновения сбоя. С другой стороны, в самой системе предоставляются некоторые механизмы для решения данной проблемы:

1. Использование регистра вектора прерывания мультипортовой платы.

Регистр вектора прерывания представляет собой 8-разрядный порт в адресном пространстве компьютера (в нашем случае - для режима *HOSTESS* с базовым адресом *0x100* - расположен по адресу *0x107*), каждый бит которого индицирует состояние одной из 8-ми линий мультипорта:

- 1 - линия запросила операцию ввода/вывода;
- 0 - линия не активна.

Драйвер *serial* может обрабатывать эту информацию и таким образом более аккуратно обслуживать запросы мультипорта. Для этого в файле */etc/rc.serial* после определения линий мультипортовой платы следует поместить строку следующего вида:

```
`${SETSERIAL} /dev/ttyS16 set_multiport port1 0x107 mask1 0xff match1 0
```

2. Запрет маскирования прерываний при выполнении дисковых операций ввода-вывода.

Как показала практика, первый метод решает проблему лишь в незначительной степени. Это вызвано тем, что при выполнении дисковых операций прерывания от последовательных портов маскируются, и операции с ними откладываются до завершения работы с диском. Поэтому, кроме вышеописанного, может понадобиться с помощью утилиты *hdparm* настроить систему так, чтобы прерывания от последовательных портов обрабатывались с более высоким приоритетом. Для этого необходимо выполнить (для автоматической отработки - поместить, например, в */etc/rc.local*) для каждого из имеющихся на компьютере жестких дисков команду вида:

```
/sbin/hdparm -u1 /dev/hda
```

Использование данной команды дает наиболее радикальный эффект в плане увеличения реактивности системы, но использоваться она должна с осторожностью, поскольку, как пишется в ее описании, некоторые комбинации диск/контроллер нетолерантны к возможному при включении этой функции увеличению времени ожидания ввода-вывода, что может привести к массивному краху файловой системы. В частности, опасны контроллеры (E)IDE CMD-640B и RZ1000 (из-за ошибки в аппаратуре), когда эта опция используется с ядрами версий ниже 2.0.13.