Алексей Уваров

Методика тестирования функционирования процессорных плат фирмы Fastwel под ОС QNX

Введение

Говоря о QNX, мы подразумеваем прежде всего надёжную ОС, используемую в ответственных задачах. Это означает, что к аппаратуре и, в частности, к процессорной плате, на которой будет работать эта ОС РВ, предъявляются высокие требования по совместимости, надёжности и отказоустойчивости. Подобные требования накладывают обязательства по проведению полноценного тестирования всей системы.

Фирма Fastwel проводит обязательное тестирование всей своей продукции под ОС РВ QNX 4 и QNX 6 различных версий. Результаты этого тестирования вы можете посмотреть на странице технической поддержки ОС QNX: http://www.fastwel.ru/support/qnx/. Описанная в статье методика тестирования поможет вам самостоятельно убедиться в приведённых результатах. Все необходимые для этого утилиты вместе с исходными текстами вы найдёте на странице технической поддержки ОС QNX.

Также в этой статье даны рекомендации по сборке образа ОС QNX Momentics вместе с графической оболочкой Photon. А на сайте технической поддержки есть утилиты, позволяющие значительно упростить этот процесс.

О методике тестирования

Используемые в статье тесты выполнены преимущественно в виде shell-скриптов. Это сделано для наглядности и для упрощения их возможной доработки для другой процессорной платы или другой конфигурации стенда для тестирования. Этот подход позволяет использовать данные тесты под ОС РВ QNX 4.

Поскольку разработчика интересует работоспособность системы целиком, то тестирование осуществляется для каждого функционального узла процессорной платы в комплексе. Каждая возникающая ошибка требует индивидуального изучения. В случае её обнаружения обращайтесь в службу технической поддержки ОС QNX: qnx@fastwel.ru.

Общий подход к тестированию и проверке совместимости драйверов ОС QNX и контроллеров процессорной платы можно сформулировать следующим образом:

1) для начала смотрим список поддерживаемого в ОС QNX оборудования и находим там нужный драйвер. Этот список вы найдете по ссылке: http://www.qnx.com/develoers/ hardware_support/index.html. Если в списке нужного вам драйвера не оказалось, то ищите наиболее подходящий;
2) затем пробуем запустить драйвер, подбирая подходящие аргументы запуска;
3) после запуска делаем предварительную оценку совместимости;
4) если всё работает, запускаем на длительное время специальный тест, позволяющий выявить различные сбои в программной или аппаратной части. Для повышения эффективности тест должен максимально нагрузить контроллер и соответствующий драйвер. В зависимости от сложности проверяемого функционального узла платы и от его важности для системы время тестирования может меняться от нескольких часов до нескольких суток.

Конфигурация стенда

Стенд для проведения тестирования процессорной платы (рис. 1) включает следующие составные части:

- инструментальный компьютер с предустановленной ОС QNX Momentics SE или PE (self hosted) и всей необходимой периферией;

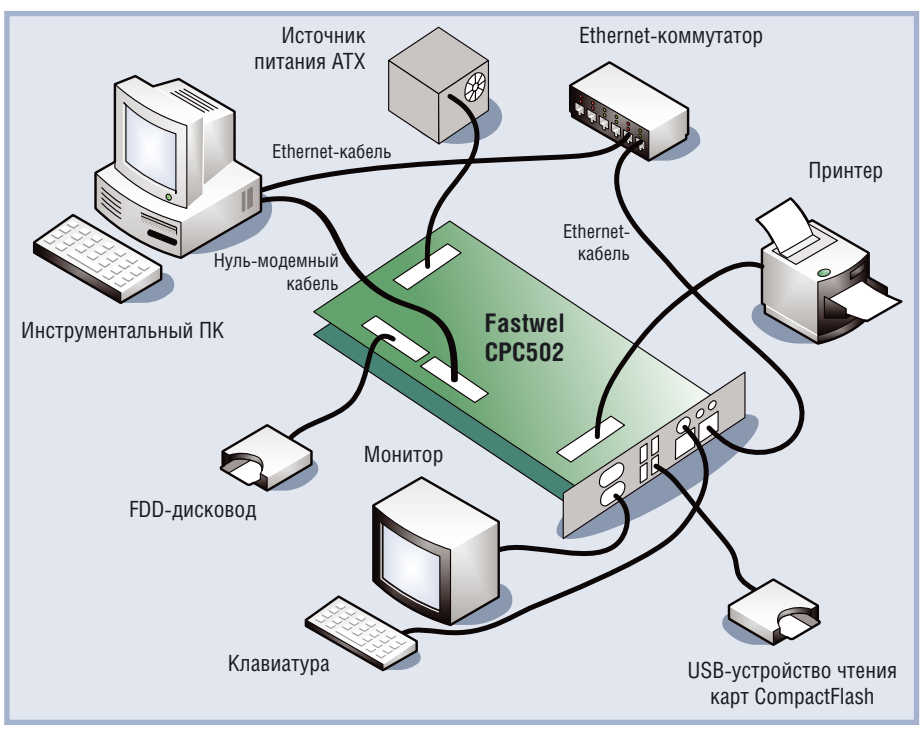

Рис. 1. Схема конфигурации стенда

© 2007, СТА Тел.: (495) 234-0635 Факс: (495) 232-1653 http://www.cta.ru

- процессорная плата фирмы Fastwel как объект тестирования;
- монитор;
- клавиатура;
- мышь;
- принтер для проверки печати через параллельный порт;
- USB-устройство чтения карт CompactFlash;
- USB-мышь и USB-клавиатура для проверки USB-портов;
- FDD-дисковод;
- набор кабелей.

Тестирование проводилось под ОС QNX 6.3.0 SP3. В качестве примера проверки методики тестирования взята процессорная плата Fastwel CPC502 вместе с необходимыми платами расширения MIC580 и RIO582. Эта компактная плата имеет все стандартные интерфейсы для специализированных плат с большим количеством портов. В статье проведено тестирование следующих её функциональных узлов:

- VGA-контроллера,
- EIDE-интерфейса,
- Ethernet-портов,
- COM-портов,
- USB-портов,
- FDD-интерфейса,
- параллельного порта,
- часов реального времени.

Создание образа QNX

Итак, приступим. Сначала соберём образ ОС. Для этого нужен файл построения образа. Он может быть простым и универсальным для всех процессорных плат x86 архитектуры с BIOS, либо специально разработанным для тестируемой платы.

В первом случае можно использовать самый простой образ, запускающий только основные драйверы. Пример такого образа вы можете найти на вашем инструментальном ПК: `/boot/build/bios.build`.

Во втором случае требуется образ, включающий в себя все или практически все драйверы устройств тестируемой процессорной платы, запускаемые с определёнными аргументами, то есть нужен уже заранее проверенный файл построения образа. Его можно получить, обратившись в отдел технической поддержки фирмы-разработчика, в данном случае в службу технической поддержки ОС QNX фирмы Fastwel. Из двух поставляемых к каждой процессорной плате Fastwel файлов построения для задачи тестирования лучше подойдёт «альтернативный» – файл с обозначением alt. Для выбранной платы это `cpc502-alt.build`. В него добавлены утилиты, необходимые для диагностики ошибок платы и их исправления.

На заметку: в каждом образе ОС рекомендуется использовать технологическую консоль, подключённую через COM1. Она может быть использована как основная консоль доступа к компьютеру или как резервная на случай блокирования основной. Для такой цели оптимально подойдёт командный интерпретатор `fesh`. В нём содержатся основные утилиты для работы с файлами и файловой системой, что может оказаться необходимым в случае сбоя во встроенной файловой системе.

Сборка образа осуществляется утилитой `mkifs`. Вот пример:

```
# mkifs -v cpc502-alt.build cpc502-alt.ifs
```

Создание файловой системы на CompactFlash

Все необходимые файлы и утилиты для тестирования платы можно включить в образ ОС и позже запускать их из файловой системы образа, либо записать их на диск и запускать уже оттуда. Пожалуй, второй вариант является более простым. Рассмотрим, как можно легко его реализовать, на примере с использованием CompactFlash.

Возьмем CompactFlash ёмкостью 128 Мбайт или больше. При желании для целевой системы QNX Neutrino вместе с графической оболочкой Photon можно уместить в 16 Мбайт, а без Photon и в 2 Мбайт. Конечно, требуемая ёмкость диска зависит от поставленной задачи. В нашем случае экономия места на диске не требуется и неоправданно отнимет драгоценное время на процесс сборки. Тем более, что в процессе тестирования вы ещё не знаете, с какими проблемами придётся столкнуться. Поэтому рекомендуется скопировать всё, что может пригодиться в процессе работы с платой.

Итак, подключим CompactFlash к инструментальному ПК через USB-устройство чтения карт CompactFlash. Если на нём нет файловой системы QNX4, создадим её. Для начала запустим драйвер флэш-диска:

```
# io-usb -dehci
# devb-umass cam pnp
```

В системе должен появиться префикс этого устройства, предположим, */dev/hd1*, если у вас в компьютере установлен всего один жёсткий диск. Проверяем:

```
# ls /dev/hd?
/dev/hd0      /dev/hd1
```

Затем очищаем таблицу разделов этого диска:

```
# dd if=/dev/zero of=/dev/hd1 count=100
```

Создаём раздел QNX, делаем его загружаемым и записываем QNX-загрузчик:

```
# fdisk /dev/hd1 delete -a
# fdisk /dev/hd1 add -s 1 qnx all
# fdisk /dev/hd1 boot -s 1
# fdisk /dev/hd1 loader
```

Проверим результат:

```
# fdisk /dev/hd1 show
    _____OS_____      Start       End
______Number______    Size     Boot
   name    type    Cylinder  Cylinder  Cylinders
Blocks
1. DOS       77          0         497        497
254944    124 MB    *
2. ---    --    ----    ----    ----   ----   ---
3. ---    --    ----    ----    ----   ----   ---
4. ---    --    ----    ----    ----   ----   ---
```

Подключаем только что созданные разделы:

```
# mount -e /dev/hd1
```

Проверяем:

```
# ls /dev/hd1*
/dev/hd1       /dev/hd1t77
```

Инициализируем файловую систему QNX4 в появившемся разделе:

```
# dinit -h /dev/hd1t77
```

Подмонтируем только что созданную файловую систему в `/fs/uf`:

```
# mount /dev/hd1t77 /fs/uf
```

Теперь можно скопировать сюда все файлы, которые вам могут пригодиться:

```
# cp -cAR /etc /fs/uf/
# cp -cAR /usr/qnx630/target/qnx6/x86/bin /fs/uf/
# cp -cAR /usr/qnx630/target/qnx6/x86/lib /fs/uf/
# cp -cAR /usr/qnx630/target/qnx6/x86/sbin /fs/uf/
# cp -cAR /usr/qnx630/target/qnx6/x86/usr /fs/uf/
# cp -A /usr/photon/config/crtc-settings /fs/uf/usr/photon/config/
```


© 2007, СТА Тел.: (495) 234-0635 Факс: (495) 232-1653 http://www.cta.ru

```
# cp -cAR /usr/photon/config/pterm /fs/uf/usr/photon/config/
# cp -cAR /usr/photon/config/wm /fs/uf/usr/photon/config/
# cp -cAR /usr/photon/font_repository /fs/uf/usr/photon/
# cp -cAR /usr/photon/keyboard /fs/uf/usr/photon/
```

Не забудьте добавить образ ОС:

```
# cp cpc502-alt.ifs /fs/uf/.boot
```

Или собранный из файла `bios.build`:

```
# cp bios.ifs /fs/uf/.boot
```

Теперь отмонтируем диск и выгрузим из памяти драйвер, чтобы можно было отключить USB-устройство чтения карт CompactFlash и вынуть CompactFlash:

```
# umount /fs/uf
# umount /dev/hd1
# slay -f devb-umass
# slay -f io-usb
```

Все то же самое вы можете сделать, воспользовавшись всего тремя специально разработанными для этой цели скриптами:

```
# ufinit cpc502-alt.ifs
# ufcpfs
# umass stop
```

Первый создаёт раздел на диске, инициализирует его и подключает в `/fs/uf`. Если указано имя файла образа ОС, лежащего в этом каталоге, утилита копирует его. В противном случае собирает и копирует `bios.ifs`, используя стандартный образ `/boot/build/bios.build`. Второй копирует на этот диск файлы. Третий отключает CompactFlash.

Эти и другие полезные утилиты вы можете найти на странице технической поддержки ОС QNX.

На этом пока все. Теперь можно вставить только что созданный загрузочный диск с QNX Neutrino в целевой компьютер и приступить к тестированию.

Тестирование EIDE-интерфейса

Перейдём к тестированию EIDE-контроллера – ключевой части для надёжности работы вашей системы, если планируется использовать CompactFlash или другой диск с EIDE-интерфейсом в качестве системного диска.

Запуск драйвера с нужными аргументами

Итак, у нас есть готовая загрузочная карта CompactFlash с QNX, которую вставляем в нашу целевую плату и загружаемся. Если после меню выбора разделов для загрузки появляется сообщение «Missing Operating System», измените в настройках BIOS тип используемого диска с LBA на Normal или наоборот.

После включения, посмотрев содержимое папок в файловой системе, убеждаемся, что все файлы на своих местах. Проверяем, не отрапортовал ли EIDE-драйвер об ошибках при запуске в системный журнал. Для этого запускаем `sloginfo`.

В случае если драйвер `devb-eide` не стартовал или стартовал с ошибками, следует попробовать его запустить с другими аргументами, например, отключив поддержку DMA:

```
# devb-eide eide nobmstr blk automount=hd0t77:/
```

Для подобных экспериментов необходимо, чтобы в файловой системе образа содержались следующие утилиты (необходимые драйверы и библиотеки там уже есть): `ls`, `sin` или `pidin`, `rm` или `slay`, `mount` и `umount`.

В противном случае для каждой попытки запуска EIDE-драйвера нужно будет менять файл построения образа, заново собирать его, перезаписывать образ на CompactFlash и снова пробовать запустить. Поэтому рекомендуется использовать специально разработанные для подобных случаев образы альтернативной загрузки, те что с обозначением alt.

Начальная проверка

После того, как EIDE-драйвер запущен и файловая система с CompactFlash подключена, можно приступить к проверке. Для начала воспользуемся стандартными средствами проверки диска на наличие сбойных блоков:

```
# dcheck -mrwV /
```

и утилитой проверки файловой системы:

```
# chkfsys /
```

Теперь проверим сохранность файлов в файловой системе диска. Для этого посчитаем контрольную сумму файлов, хранящихся в самом большом каталоге, и запишем результат на диск:

```
# cksum `find /usr -type f` > /cksum0.res
```

Перезапускаем компьютер по команде shutdown и повторяем команду, записывая результат уже в другой файл:

```
# cksum `find /usr -type f` > /cksum1.res
```

Перезагрузка – это простой и эффективный способ перезапустить драйвер, проинициализировать EIDE-контроллер и сбросить кэш диска. Сравниваем результаты:

```
# diff /cksum0.res /cksum1.res
```

Если утилита ничего не вывела на экран, значит, предварительный тест по выявлению явных ошибок прошел успешно. Но расслабляться ещё рано.

Удалим уже ненужные файлы с результатами:

```
# rm cksum0.res /cksum1.res
```

Полная проверка

Переходим к полной проверке файловой системы EIDE-диска. Для этого нужна утилита, позволяющая циклически выполнять проверку записи и чтения файлов, перезапуская драйвер `devb-eide` в конце каждого цикла проверки. Такая утилита есть и называется `eiderstst`. Скачать её вы можете со страницы технической поддержки ОС QNX.

Рассмотрим её алгоритм работы.

1. Скопировать необходимые файлы и утилиты в `/dev/shmem` (файловая система, располагающаяся в оперативной памяти):

```
# cp /sbin/devb-eide /dev/shmem/devb-eide
# cp /bin/mount /dev/shmem/mount
```

Если используется образ `cpc502-alt.ifs`, то этот пункт можно пропустить, поскольку все необходимые утилиты и драйверы уже включены в файловую систему образа.

2. Подсчитать контрольную сумму всех (или большей части) файлов файловой системы на диске. Например, возьмем каталог `/usr`:

```
# cksum `find /usr -type f` > /dev/shmem/cksum0.res
```

3. Переместить эти файлы в архив, расположенный в ОЗУ:

```
# tar -czf /dev/shmem/usr.tgz usr
# rm -R /usr
```

4. Выгрузить из памяти драйвер `devb-eide`:

```
# slay devb-eide
```

5. Запустить EIDE-драйвер:

```
/dev/shmem/devb-eide blk cache=256k,
automount=/dev/hd0t77:/:qnx4 &
```

6. Распаковать архив:

```
# tar -xzf /dev/shmem/usr.tgz
```

7. Пересчитать контрольную сумму файлов этого каталога:

```
# cksum `find /usr -type f` > /dev/shmem/cksum1.res
```

8. Сравнить полученный результат и ответ сохранить в журнале на диске:

```
# diff /dev/shmem/cksum0.res /dev/shmem/cksum1.res >> /eidetst.res
```

9. Перейти к п.3.

© 2007, СТА Тел.: (495) 234-0635 Факс: (495) 232-1653 http://www.cta.ru

После запуска тест будет работать до нажатия клавиш Ctrl+C или до завершения работы. Тестирование CompactFlash в составе специализированной процессорной платы следует проводить в течение нескольких суток.

Тестирование FDD-интерфейса

Тестирование FDD-интерфейса будет осуществляться по схожему с тестированием EIDE алгоритму. Поэтому мы не будем повторно рассматривать описанный ранее алгоритм с перезапуском драйвера, а сразу перейдём к запуску теста. Утилита для тестирования FDD называется `fddtst`. Скачать её можно со страницы технической поддержки ОС QNX.

Перед началом теста отформатируем дискету. Для этого запускаем драйвер:

```
# devb-fdc &
```

И начинаем форматировать:

```
# fdformat /dev/fd0
```

Теперь можно запускать тест:

```
# fddtst
```

Если в течение нескольких часов тестирования не произошло ни одной ошибки, то тест можно считать успешно пройденным. В случае возникновения ошибки попробуйте сменить дискету и повторить тест – скорее всего ошибка была именно в ней.

Тестирование USB-портов

Проведем тестирование USB-портов. Для начала запускаем драйверы всех поддерживаемых USB-контроллеров:

```
# io-usb -dehci -dohci -duhci
```

Определяем, какие устройства подключены в систему, с помощью утилиты `usb`. Для процессорной платы CPC502 и подключённых к ней USB-клавиатуры, USB-мыши и CompactFlash через USB-устройство чтения карт CompactFlash результат будет следующий:

```
# usb
USB 0 (EHCI) v1.10, v1.01 DDK, v1.01 HCD

Device Address         : 1
Vendor                 : 0x0424
Product                : 0x2502
Class                  : 0x09 (Hub)
Subclass               : 0x00
Protocol               : 0x01
Hub Number Ports       : 2
Hub Characteristics    : 0x0010 (Ganged power,No overcurrent)
Hub Power On->Good     : 100 ms
Hub Power Requirements : 100 mA

Device Address         : 2
Vendor                 : 0x1267
Product                : 0x0201 (USB Mouse)
Class                  : 0x00 (Independant per interface)

Device Address         : 3
Vendor                 : 0x0aec (Generic )
Product                : 0x3260 (USB Storage Device)
Class                  : 0x00 (Independant per interface)

USB 1 (UHCI) v1.10, v1.01 DDK, v1.01 HCD

Device Address         : 1
Vendor                 : 0x0566
Product                : 0x3002
Class                  : 0x00 (Independant per interface)

USB 2 (UHCI) v1.10, v1.01 DDK, v1.01 HCD
```


© 2007, СТА Тел.: (495) 234-0635 Факс: (495) 232-1653 http://www.cta.ru

Как видно из результата, в системе обнаружено два типа USB-контроллеров: EHCI и UHCI. Причем USB-устройство чтения карт CompactFlash и USB-мышь подключены к EHCI-контроллеру, а USB-клавиатура – к UHCI-контроллеру. Теперь включаем эти устройства в другие USB-порты и убеждаемся, что они тоже работают. Таким образом мы проверили совместимость USB-контроллеров с драйверами ОС QNX.

Теперь проверим стабильность работы USB-портов. Для этого воспользуемся специально разработанным скриптом `usbmasstst`, который вы можете найти на странице технической поддержки ОС QNX. Алгоритм работы этого скрипта аналогичен схеме тестирования работы EIDE-контроллера, с той разницей, что теперь потоки данных файловой системы будут пропускаться через USB-контроллер, а не EIDE, поэтому повторно рассматривать алгоритм мы не будем.

Итак, подключаем USB-устройство чтения карт CompactFlash вместе с CompactFlash в порт тестируемой платы. На CompatcFlash уже должна быть проинициализирована файловая система QNX 4. Запускаем скрипт:

```
# usbmasstst
```

По истечении многочасового тестирования смотрим журнал результатов тестирования. Если в нём не содержится ошибок, то значит, тест на стабильность работы USB-порта при его интенсивной загруженности прошёл успешно. Аналогичным образом проверяем все остальные порты.

Тестирование COM-портов

COM-порт в промышленной автоматизации до сих пор является чуть ли не основным интерфейсом для подключения периферийных устройств (датчиков, контроллеров, модемов и т.п.). По этой причине его тестированию следует уделить особое внимание.

В зависимости от типа используемого трансивера порт может функционировать как интерфейс RS-232, RS-422 или RS-485. Во всех случаях используется один и тот же драйвер `devc-ser8250`, совместимый с 8250, 14450 и 16550 UART-контроллерами. Метод тестирования меняется только в случае с RS-485, поскольку это уже полудуплексный интерфейс. Для него потребуется использование специально разработанных программ диагностики. Интерфейсы RS-232 и RS-422 являются полнодуплексными, а значит, полностью аналогичными, с программной точки зрения.

Начальная проверка работоспособности COM-портов

Для начальной проверки работоспособности порта следует сделать «заглушку», которая соединяет приёмник трансивера с передатчиком. Обычно это контакты 2 и 3 разъемов DB-9 или IDC-10. Таким образом, можно будет проверить связь целиком, принимая все передаваемые в порт данные. Это так называемый loopback. Далее нужно запустить драйвер в текстовом режиме (используя опцию `-e`). Строка запуска драйвера для платы CPC502 (со всеми необходимыми модулями расширения для COM1, COM2 и COM3) выглядит следующим образом:

```
# devc-ser8250 -t14 -T14 -e -b115200 -u1 3F8,4 -u2 2F8,3 -u3 3E8,4 &
```

На заметку: опции `-t 14` и `-T 14` включают буферы FIFO приёмника и передатчика с уровнем срабатывания 14 байт (максимально допустимое значение), что значительно снижает нагрузку на процессор при работе драйвера. Рекомендуется всегда включать буферы FIFO контроллера UART хотя бы уровнем срабатывания 4 байта. В противном случае при значительной нагруженности последовательного интерфейса на высоких скоростях обмена и при низкой производительности процессора драйвер может полностью загрузить процессор и не успеть обработать принятые или переданные данные. К счастью, на практике такой режим работы маловероятен. И все же...

Следующая строка запускает утилиту `qtalk` на COM1:

```
# qtalk -m /dev/ser1
```

Теперь, если на плате нет ошибок и драйвер был запущен с нужными параметрами, всё, что вы наберёте на клавиатуре, сразу отобразится на экране как «эхо». Выньте «заглушку» и убедитесь, что связь пропала. Проделайте тот же тест с остальными портами платы.

Таким образом вы проверили работоспособность всего передающего и приёмного тракта COM-портов в ненагруженном режиме.

Простая проверка связи с другим компьютером

Теперь проверим совместимость COM-портов между целевой платой и другим компьютером, в данном случае с инструментальным ПК с предустановленным QNX Momentics. Для этого нужно использовать нуль-модемный кабель. Достаточно использовать трёхпроводной кабель – первые два провода перекрёстно соединяют приёмники и передатчики COM-портов, а третий соединяет их земли. Подключаем кабель. На инструментальном ПК запускаем драйвер UART, который автоматически подключает существующие COM-порты из двух стандартных (2f8,3 3f8,4):

```
# devc-ser8250 -t14 -T14 -e -b115200 &
```

Запускаем `qtalk`, чтобы использовать ПК в качестве терминала связи с целевой платой. Допустим, ПК подключён через COM2:

```
# qtalk -m /dev/ser2
```

Запускаем на целевой плате `devc-ser8250` и `qtalk`, как это показано в предыдущем тесте. Набираем в терминале любой текст и видим его вывод на экране другого компьютера. Теперь меняем компьютеры и проделываем то же самое на другом терминале.

Для усложнения теста можно закрыть `qtalk` и отправить в порт текстовый файл или результат работы любой текстовой утилиты:

```
# hogs -ns1 / > /dev/ser2
```

Её вывод вы увидите на другом компьютере.

Таким образом мы проверили связь с другим компьютером в ненагруженном режиме, исключив основные проблемы в работе COM-портов.

Тестирование стабильности связи с другим компьютером

Продолжим тестирование – проверим стабильность связи между компьютерами. Для этого используем специально разработанную программу тестирования для последовательных портов `sertst`. Скачать её можно со страницы технической поддержки ОС QNX.

Принцип работы утилиты следующий. В режиме передачи (опция `-T`) утилита постоянно генерирует пакеты и передаёт их драйверу UART-контроллера. Эти пакеты имеют определённый формат и содержание. При этом может меняться их длина (опция `-l`) и скважность передачи (опция `-d`). В режиме приёма (режим по умолчанию или опция `-R`) утилита принимает все пакеты от драйвера UART-контроллера, подсчитывает и отображает статистику. Таким образом с помощью этой

© 2007, СТА Тел.: (495) 234-0635 Факс: (495) 232-1653 http://www.cta.ru

утилиты можно эмулировать пакетный обмен между устройствами через COM-порт, причём помимо обычных полнодуплексных интерфейсов RS-232 и RS-422 возможно протестировать ещё полудуплексные типа RS-485 (опция -h).

Дополнительную гибкость в построении тестового стенда даёт режим работы обратной петли (опция -L). В этом режиме утилита работает как программная обратная петля, передавая в порт все принятые байты. В результате с помощью sertst можно протестировать стабильность связи COM-портов в различных схемах подключения.

Рассмотрим простой метод тестирования COM-порта. Вставляем в порт «заглушку». Задаём скорость обмена 57600 бод и переводим драйвер в режим работы RAW:

```
# stty 57600 </dev/ser1
# stty raw </dev/ser1
```

Теперь запускаем тест на 3600 секунд (опция -w):

```
# sertst -T -l 1024 -d 1 -w 3600 &
# sertst -v -l 1024 -d 1 -w 3600
```

Если требуется больше отладочной информации (опция -v), то утилиты лучше запускать на разных консолях. По истечении 1 часа утилита, работающая в режиме приёмника, выдаст накопленную статистику:

- количество принятых байтов,
- количество потерянных байтов,
- количество изменённых байтов,
- количество принятых пакетов,
- количество пропущенных пакетов,
- количество пакетов, содержащих ошибки.

Если ошибок нет, то тестирование COM-порта в интенсивно нагруженном режиме работы можно считать успешным. Аналогичным образом тестируем все остальные порты.

Тестирование Ethernet-портов

Для начала проведём диагностику Ethernet-портов и работы сетевых протоколов с этими контроллерами. Запускаем Ethernet-драйвер и два стека протоколов, Qnet и TCP/IP, включив программную проверку CRC в Qnet:

```
# io-net -di82544 -pqnet host=cpc502,do_crc=1 &
```

Проверяем, что запустилось:

```
# ls /dev/io-net
en0         en1         ip0         ip_en       qnet_en
```

Как мы видим, для Fastwel CPC502 драйвер обнаружил и подключил оба Ethernet-контроллера.

На всякий случай проверяем, какие библиотеки запущены в сетевой системе менеджера io-net:

```
# pidin -P io-net mem
pid tid name          prio STATE         code  data     stack
11   1 sbin/io-net    10r SIGWAITINFO   64K 4132K    24K(516K)*
11   2 sbin/io-net    21r RECEIVE       64K 4132K    4096(132K)
11   3 sbin/io-net     9r RECEIVE       64K 4132K    8192(68K)
11   4 sbin/io-net    10r RECEIVE       64K 4132K    4096(68K)
11   5 sbin/io-net    21r RECEIVE       64K 4132K    4096(132K)
11   6 sbin/io-net    10r RECEIVE       64K 4132K    4096(132K)
11   7 sbin/io-net    20r RECEIVE       64K 4132K    4096(132K)
11   9 sbin/io-net    10r RECEIVE       64K 4132K    4096(132K)
11  10 sbin/io-net     9r RECEIVE       64K 4132K    4096(68K)
11  11 sbin/io-net    10r RECEIVE       64K 4132K    4096(68K)
       ldqnx.so.2         @b0300000          348K   20K
       devn-i82544.so     @b8200000           32K   4096
       npm-qnet.so        @b8209000          144K   36K
       npm-tcpip.so       @b8236000          276K   28K
       /dev/mem           @40100000 (e0200000)      128K
       /dev/mem           @40120000 (e0220000)      128K
```


© 2007, СТА Тел.: (495) 234-0635 Факс: (495) 232-1653 http://www.cta.ru

Проверяем функционирование Ethernet-драйвера с помощью утилиты nicinfo, запуская её для каждого порта в отдельности:

```
# nicinfo -r en0
INTEL 82544 Gigabit (Copper) Ethernet Controller

  Physical Node ID ..... 0008B3
00011E
  Current Physical Node ID ..... 0008B3
00011E
  Current Operation Rate ..... 100.00
Mb/s full-duplex
  Active Interface Type ..... MII
    Active PHY address ..... 0
  Maximum Transmittable data Unit ..... 1514
  Maximum Receivable data Unit ..... 0
  Hardware Interrupt ..... 0xb
  Memory Aperture ..... 0xe0200000
- 0xe021ffff
  Promiscuous Mode ..... Off
  Multicast Support ..... Enabled

  Packets Transmitted OK ..... 12379
  Bytes Transmitted OK ..... 4685339
  Broadcast Packets Transmitted OK ..... 324
  Multicast Packets Transmitted OK ..... 0
  Memory Allocation Failures on Transmit ..... 0

  Packets Received OK ..... 25803
  Bytes Received OK ..... 2684312
  Broadcast Packets Received OK ..... 13798
  Multicast Packets Received OK ..... 1957
  Memory Allocation Failures on Receive ..... 0

  Single Collisions on Transmit ..... 0
  Deferred Transmits ..... 0
  Late Collision on Transmit errors ..... 0
  Transmits aborted (excessive collisions) ... 0
  No Carrier on Transmit ..... 0
  Receive Alignment errors ..... 0
  Received packets with CRC errors ..... 0
  Packets Dropped on receive ..... 0

# nicinfo -r en1
INTEL 82544 Gigabit (Copper) Ethernet Controller

  Link is DOWN

  Physical Node ID ..... 0008B3
00011F
  Current Physical Node ID ..... 0008B3
00011F
  Current Operation Rate ..... 0 kb/s
half-duplex
  Active Interface Type ..... MII
    Active PHY address ..... 0
  Maximum Transmittable data Unit ..... 1514
  Maximum Receivable data Unit ..... 0
  Hardware Interrupt ..... 0xb
  Memory Aperture ..... 0xe0220000
- 0xe023ffff
  Promiscuous Mode ..... Off
  Multicast Support ..... Enabled

  Packets Transmitted OK ..... 0
  Bytes Transmitted OK ..... 0
  Broadcast Packets Transmitted OK ..... 0
  Multicast Packets Transmitted OK ..... 0
  Memory Allocation Failures on Transmit ..... 0

  Packets Received OK ..... 0
  Bytes Received OK ..... 0
  Broadcast Packets Received OK ..... 0
  Multicast Packets Received OK ..... 0
  Memory Allocation Failures on Receive ..... 0

  Single Collisions on Transmit ..... 0
  Deferred Transmits ..... 0
  Late Collision on Transmit errors ..... 0
  Transmits aborted (excessive collisions) ... 0
  No Carrier on Transmit ..... 0
  Receive Alignment errors ..... 0
  Received packets with CRC errors ..... 0
  Packets Dropped on receive ..... 0
```

Обратите внимание на счетчики успешно принятых и переданных пакетов. Если они нулевые, то либо драйвер работает неправильно, либо проблемы в физическом соединении сетевого кабеля. В данном случае специально не был подключён сетевой второй кабель.

Отладочную информацию протокола Qnet можно посмотреть в файле /proc/qnetstats. Часть этой информации поступает в системный журнал в менеджер slogger. Проверить её можно утилитой sloginfo.

Теперь смотрим, какие компьютеры есть в сети:

```
# ls /net
cpc502      uvarov
```

В рассматриваемом примере это инструментальный компьютер с именем uvarov и тестируемая процессорная плата с именем cpc502.

Проверяем доступность удалённого компьютера:

```
# ls /net/uvarov
```

Если ошибок в связи нет, то сразу должна вывестись корневая файловая система компьютера uvarov. Переключаем сетевой кабель во второй порт. Через несколько секунд связь с удалённым компьютером должна восстановиться. На этом проверку совместимость драйвера Ethernet и протокола Qnet с Ethernet-контроллером можно считать успешно выполненной.

Оценить стабильность работы сетевого соединения и узнать эффективную скорость связи можно путём перекачивания больших файлов в файловом менеджере mc или mqc.

Перейдем к тестированию работы сетевого соединения через стек TCP/IP. Если в локальной сети находится DHCP-сервер, то следует поправить файл конфигурации /etc/net.cfg и запросить свободный IP-адрес у сервера:

```
# netmanager -s
```

Если DHCP-сервера нет или не удаётся получить IP-адрес автоматически, то его можно назначить вручную. Например, назначим второму порту следующий IP:

```
# ifconfig en1 192.168.5.144
```

Узнать IP-адреса можно той же командой:

```
# ifconfig
lo0: flags=8009<UP,LOOPBACK,MULTICAST> mtu 33212
    capabilities=7<IP4CSUM,TCP4CSUM,UDP4CSUM>
    enabled=0<>
    inet 127.0.0.1 netmask 0xff000000
en0: flags=8c43<UP,BROADCAST,RUNNING,OACTIVE,SIMPLEX,MUL-
TICAST> mtu 1500
    capabilities=7<IP4CSUM,TCP4CSUM,UDP4CSUM>
    enabled=0<>
    address: 00:08:b3:00:01:1e
    inet 192.168.5.48 netmask 0xffffff00 broadcast
192.168.5.255
en1: flags=8c43<UP,BROADCAST,RUNNING,OACTIVE,SIMPLEX,MUL-
TICAST> mtu 1500
    capabilities=7<IP4CSUM,TCP4CSUM,UDP4CSUM>
    enabled=0<>
    address: 00:08:b3:00:01:1f
```


© 2007, СТА Тел.: (495) 234-0635 Факс: (495) 232-1653 http://www.cta.ru

```
    inet 192.168.5.144 netmask 0xffffff00 broadcast
192.168.5.255
```

Убеждаемся, что `ftp` и `telnet` разрешены в файле конфигурации `/etc/inetd.conf`. Запускаем соответствующие «демоны»:

```
# inetd
```

Теперь можно проверить доступ к портам с удалённого компьютера, обращаясь по их адресам:

```
# ping 192.168.5.48
# ping 192.168.5.144
```

Если доступ есть, запускаем удалённый терминал. Поработайте в нем для проверки:

```
# telnet 192.168.5.48
```

Затем подключаем ко второму порту сетевой кабель и делаем для него то же самое:

```
# telnet 192.168.5.144
```

Оценить стабильность связи и узнать эффективную скорость соединения можно путём перекачивания больших файлов по ftp-протоколу. Создадим большой файл:

```
# tar -cf usr.tar usr
```

Чтобы не заводить новых пользователей в системе, разрешим доступ администратору, закомментировав поле `root` в файле `/etc/ftpusers`. В случае если у пользователя `root` ранее не был заведён пароль, нужно задать его с помощью утилиты `passwd`.

На удалённом компьютере запустим `ftp`-клиента и войдём как пользователь `root`:

```
# ftp 192.168.5.144
```

В диалоговом окне введём команду загрузки файла:

```
ftp> get /usr.tar
```

Для процессорной платы CPC502 усреднённая эффективная скорость закачивания данных составляет около 10 Мбайт/с.

Тестирование параллельного порта

Как правило, параллельный порт под управлением ОС QNX Momentics используется для решения следующих задач:

- подключения принтера и осуществления печати через драйвер параллельного порта `devc-par`;
- подключения специализированных устройств, не предназначенных для вывода текстовой информации, то есть требующих разработки собственного драйвера для обмена с ним.

Второй вариант является нестандартным решением, но не менее популярным. Параллельный порт – это удобный интерфейс для подключения множества различных устройств. Это могут быть устройства отображения и сохранения информации, контактные датчики и наборы реле, устройства связи или специализированные контроллеры собственной разработки. При этом порт может работать в одном из трёх режимов: SPP, ECP или EPP — с использованием прерываний. Такое множество вариантов подключения усложняет задачу всестороннего тестирования. Здесь можно рекомендовать тестирование порта в работе с помощью прикладного ПО в связке со специализированным адаптером. Мы же ограничимся тестированием стандартного решения и проверим печать на принтере. Этот тест должен исключить большую часть потенциальных ошибок функционирования порта.

Подключаем к порту принтер. Включаем его. Запускаем драйвер для одного порта с адресом 0x378:

```
# devc-par -p0x378
```

Для печати тестовой страницы используем команды для стандартного вывода `echo` или `print`. Используем любые утилиты для вывода информации, например, `date` и `uname`. Вы

© 2007, СТА Тел.: (495) 234-0635 Факс: (495) 232-1653 http://www.cta.ru

можете использовать по аналогии любые другие утилиты или отправить на печать заранее приготовленный текстовый файл.

```
# echo «DATE:  `date`\r» | tee /dev/par1
# echo «QNX VERSION: `uname -a`\r\n» | tee /dev/par1
# echo «\f» > /dev/par1
```

На принтере должна отпечататься страничка с той же информацией, что отображена на экране. Если так, то считаем тест выполненным успешно.

Проверка часов реального времени

Проверку совместимости часов реального времени с ОС QNX проведём путём изменения их значения. Для этого используется утилита `rtc`.

Для начала узнаем текущее время ОС:

```
# date
```

Устанавливаем новую дату и время. Например, 11:00:00 09.01.2007:

```
# date 0701091100.00
```

Записываем системное время в микросхему часов реального времени, используя автоматическое определение типа микросхемы RTC:

```
# rtc -s hw
```

Перезапускаем компьютер и входим в BIOS. Убеждаемся в том, что аппаратное время было изменено. При необходимости восстанавливаем исходное значение времени. На этом проверку можно считать выполненной.

Конфигурация Photon и тестирование VGA-контроллера

Перейдем к тестированию графического контроллера. Удаляем текущие настройки графического драйвера, которые скопировали из инструментальной машины при сборке файловой системы:

```
# crttrap clear
/etc/system/config/graphics-modes removed
```

С помощью утилиты `pci` определим идентификатор фирмы-разработчика (VID), идентификатор устройства (DID) и индекс на шине PCI. Для CPC502 результат будет следующий:

```
Class          = Display (VGA)
Vendor ID      = 8086h, Intel Corporation
Device ID      = 3582h, Unknown Unknown
PCI index      = 0h
```

Вводим полученные данные о графическом контроллере в файл конфигурации `/etc/system/config/graphics-traplist`. Явно задаем драйверы, совместимые с ним. Для CPC502 получаем файл следующего содержания:

```
devgt-iographics -dldevg-i830.so -IO -d0x8086,0x3582
devgt-iographics -dldevg-vesabios.so -IO -d0x8086,0x3582
devgt-iographics -dldevg-svga.so -IO -d0x8086,0x3582
```

Запускаем утилиту автоматической конфигурации:

```
# crttrap trap
/usr/photon/bin/devgt-iographics -dldevg-i830.so -IO -d0x8086,0x3582
/usr/photon/bin/devgt-iographics -dldevg-vesabios.so -IO -d0x8086,0x3582
/usr/photon/bin/devgt-iographics -dldevg-svga.so -IO -d0x8086,0x3582
crttrap: wrote config file as /etc/system/config/graphics-modes
```

В результате получаем файл `/etc/system/config/graphics-modes`, содержащий все возможные варианты запуска заданных графических драйверов, поддерживаемых данным контроллером. Выберем наиболее подходящий для нас вариант. Например, следующая строка запускает драйвер с разрешением 1280×1024 точек на 32 бит:

```
io-graphics -di830 vid=0x8086,did=0x3582,index=0,photon,
xres=1280,yres=1024,bitpp=32 -pphoton;#1280,1024,32,100,0CBDr,
i830 - Intel 82830
```

Добавляем эту строчку с аргументом `refresh=85` и `hwcursor` в ваш сценарий запуска графической оболочки Photon целевой платы – файл `/usr/bin/ph`. Получаем следующий результат:

```
export PATH=$PATH:/usr/photon/bin
export PHOTON=/dev/photon
export PHFONT=/dev/phfont
export PHOTON_PATH=/usr/photon
export TMPDIR=/tmp
export HOME=/root
export PHWM=pwm
export USER_NAME=/dev/photon
Photon &
waitfor /dev/photon 10
io-graphics -di830 vid=0x8086,did=0x3582,index=0,photon,
xres=1280,yres=1024,bitpp=32,refresh=85 -pphoton hwcursor &
waitfor /dev/phfont 10
devi-hirun kbd fd -d/dev/kbd ps2 mousedev &
pwm &
pterm &
```

В конце сценария можно добавить вызовы разных графических утилит либо ограничиться вызовом графического терминала `pterm`. Остальные утилиты можно будет вызвать из него.

Не забудьте создать домашний каталог, куда будут записаны текущие конфигурации графической оболочки:

```
# mkdir /root
```

Наконец, можно запустить скрипт:

```
# /usr/bin/ph
```

Теперь все должно работать, либо ищите, где вы допустили ошибку.

Для тестирования следует запустить несколько различных графических программ и уделить им какое-то время. Большую их часть вы найдете в каталоге `/usr/photon/bin/`. Если в процессе работы на экране монитора вы не заметите никаких артефактов, значит, можно считать, что тестирование прошло успешно.

Для выхода из графической оболочки в окне терминала `pterm` следует вызвать следующую строку:

```
# slay Photon
```

Этот скрипт запуска Photon вы можете написать сами по аналогии или найдёте в архиве BSP, поставляемом для процессорных плат фирмы Fastwel, предварительно запросив его в службе технической поддержки ОС QNX.

Заключение

Описанная в статье методика тестирования процессорных плат фирмы Fastwel не является окончательной. Она постоянно дорабатывается. Последнюю версию утилит для проведения тестирования вы можете загрузить со страницы технической поддержки ОС QNX.

Если в процессе чтения этой статьи у вас возникли замечания, вопросы или появились советы по доработке тестового ПО или доработке методики тестирования, обратитесь в службу технической поддержки ОС QNX фирмы Fastwel. ●

Автор – сотрудник фирмы Fastwel
119313, Москва, а/я 242
Тел.: (495) 234-0639 Факс: (495) 232-1654
E-mail: info@fastwel.ru

© 2007, СТА Тел.: (495) 234-0635 Факс: (495) 232-1653 http://www.cta.ru