

Драйвер Elvees PCI

Общие сведения

Программный пакет Elvees PCI обеспечивает доступ к регистрам микросхемы на персональном компьютере через интерфейс PCI. Разработаны драйверы для операционных систем Windows (7, 8, 8.1) и Linux. Для написания прикладных программ, работоспособных в Windows и Linux, а также для упрощения интерфейса прикладного программирования, разработана кросс-платформенная библиотека libre. В данный программный пакет входят также примеры использования библиотеки.

Состав пакета

Поставка драйвера представляет zip-архив с именем elvees_pci_yyyy_mm_dd.zip, где

yyyy — год создания архива,

mm — месяц создания архива,

dd — день создания архива.

Архив содержит следующие директории:

- binaries_win — собранные драйверы для ОС Windows, примеры и динамические библиотеки, необходимые для запуска примеров;
 - binaries_win/Driver — содержит драйверы для различных версий ОС Windows, в том числе 32- и 64-разрядных вариантов;
- driver_src_linux — исходные коды драйвера Elvees PCI для ОС Linux;
- driver_src_win — исходные коды драйвера Elvees PCI для ОС Windows;
- libre — исходные коды кросс-платформенной библиотеки доступа к функциям драйвера Elvees PCI;
- test-mem — исходные коды теста памяти микросхемы;
- test-read-speed — исходные коды теста скорости чтения из микросхемы;
- test-write-speed — исходные коды теста скорости записи в микросхему;
- test-swic — исходные коды примера работы с SWIC.

Установка драйвера

Установка драйвера для ОС Windows

Драйвер Elvees PCI для ОС Windows не имеет официальной цифровой подписи, поэтому его использование возможно только в тестовом режиме.

Для перевода ОС Windows в тестовый режим необходимо открыть командную строку с правами администратора и выполнить команду:

```
bcdedit /set TESTSIGNING ON
```

Дальнейший процесс установки драйвера описан на пример Windows 8.1 x64, но во всех поддерживаемых драйвером версиях Windows установка осуществляется аналогичным образом.

Для начала необходимо установить тестовый сертификат драйвера package.cer. Его нужно выбрать из каталога binaries_win\Driver для используемой версии ОС Windows. Для установки необходимо в Проводнике щёлкнуть правой кнопкой мышки на файле и из контекстного меню выбрать пункт «Установить сертификат». Появится окно мастера импорта сертификатов. На первой странице (Рис. 1) необходимо выбрать вариант «Локальный компьютер» и нажать кнопку «Далее».

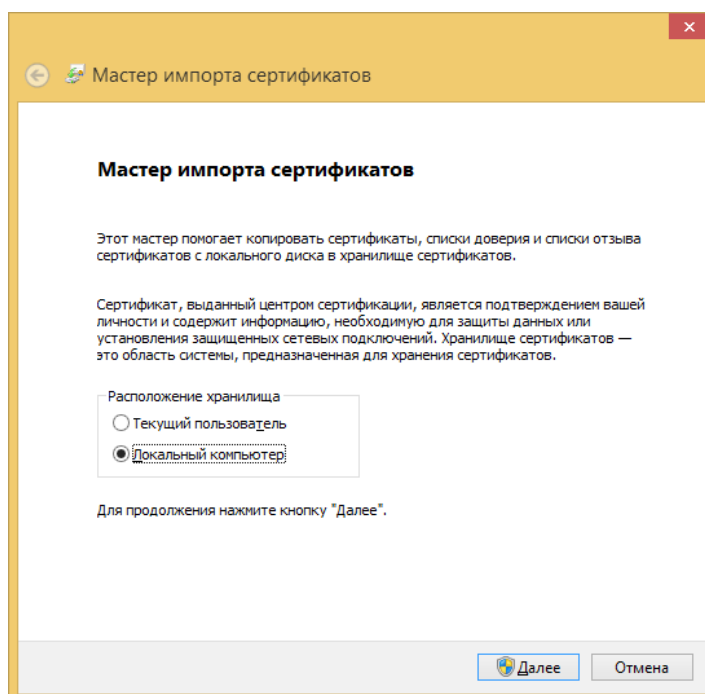


Рис. 1. Мастер импорта сертификатов.
Страница 1.

На второй странице выбрать пункт «Поместить все сертификаты в следующее хранилище» и нажать кнопку «Обзор». В появившемся окне необходимо выбрать группу «Доверенные корневые центры сертификации». Результат этих действий приведён на Рис. 2.

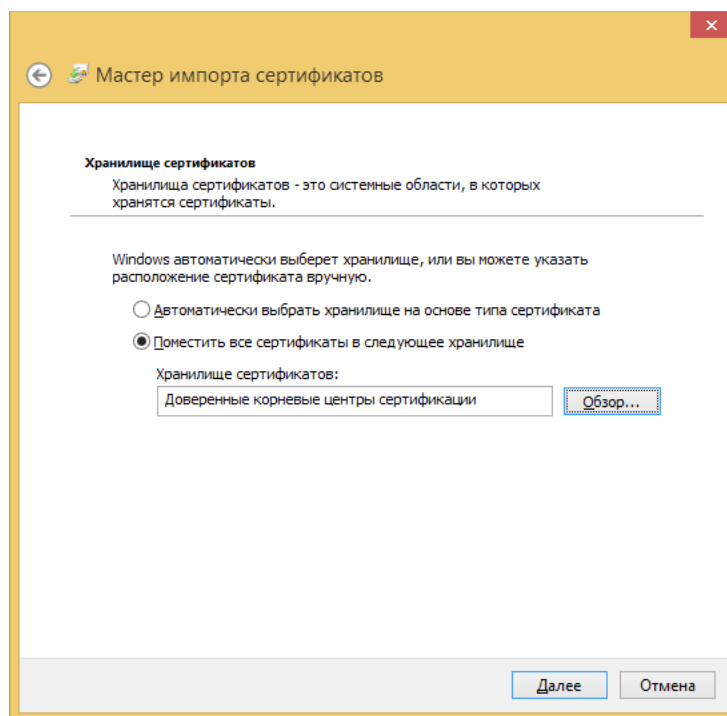


Рис. 2. Мастер импорта сертификатов. Страница 2.

После нажатия кнопки «Далее» Windows может выдать предупреждение безопасности, как показано на Рис. 3. Необходимо ответить «да».

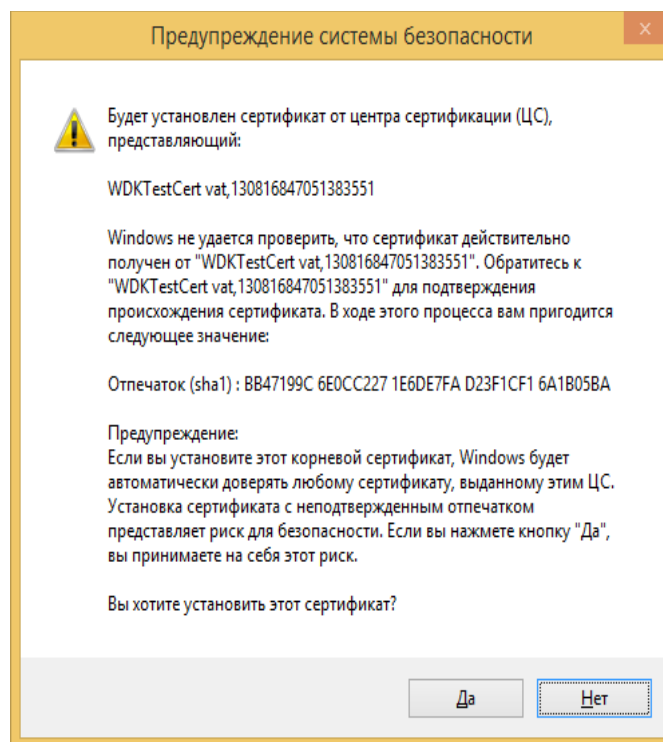


Рис. 3. Предупреждение

После установки сертификата необходимо установить сам драйвер Elvees PCI. Для этого необходимо открыть Диспетчер устройств (например, с помощью комбинации клавиш «Win

+ Pause», затем нажать строку/кнопку «Диспетчер устройств»). Окно диспетчера устройств будет выглядеть примерно так, как показано на Рис. 4.

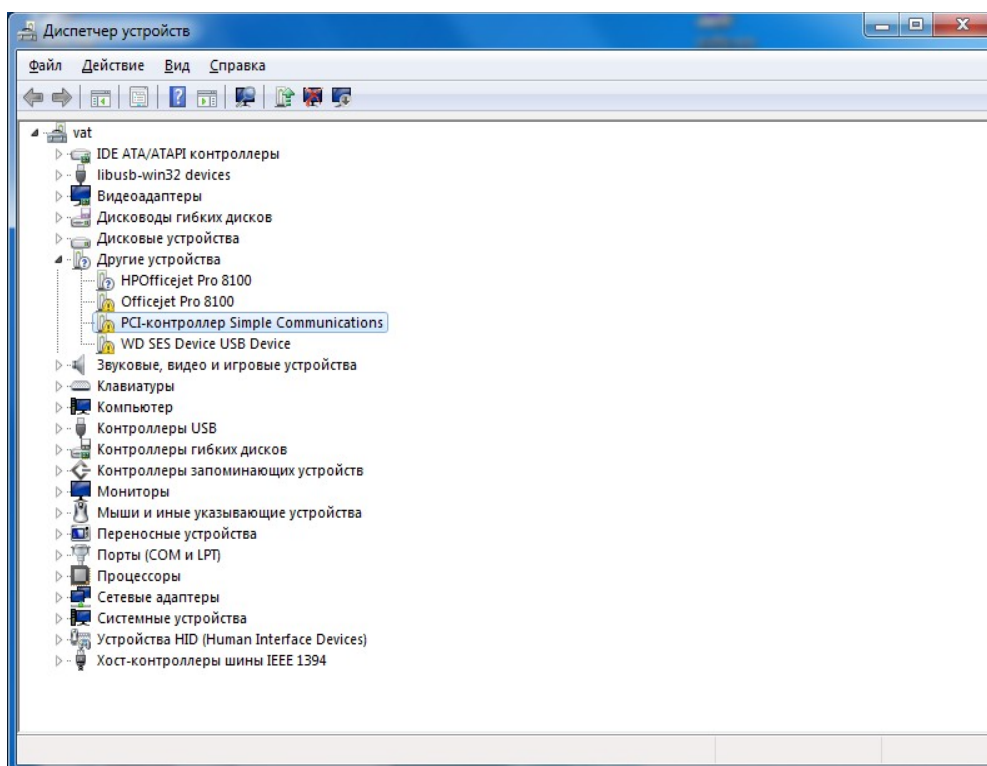


Рис. 4. Контроллер PCI до установки драйверов.

Необходимо щёлкнуть правой кнопкой мышки на узле «Контроллер PCI Simple Communications» и выбрать из контекстного меню пункт «Обновить драйверы...».

В появившемся окне мастера обновления драйверов (Рис. 5) необходимо выбрать пункт «Выполнить поиск драйверов на этом компьютере».

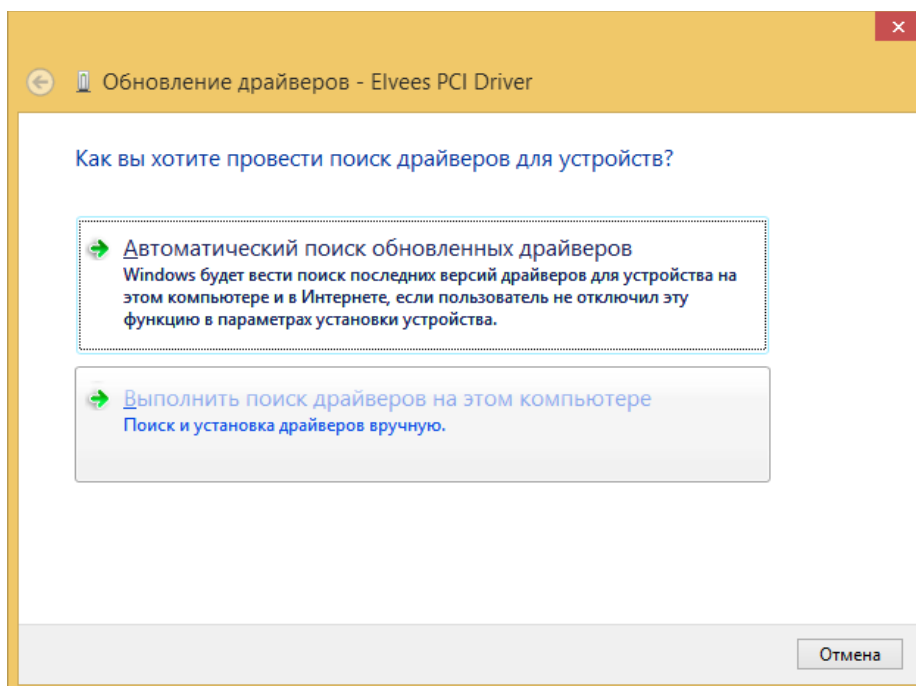


Рис. 5. Мастер обновления драйверов. Страница 1.

На следующей странице мастера (Рис. 6) необходимо указать путь к директории, содержащей драйверы для соответствующей версии Windows и нажать кнопку «Далее».

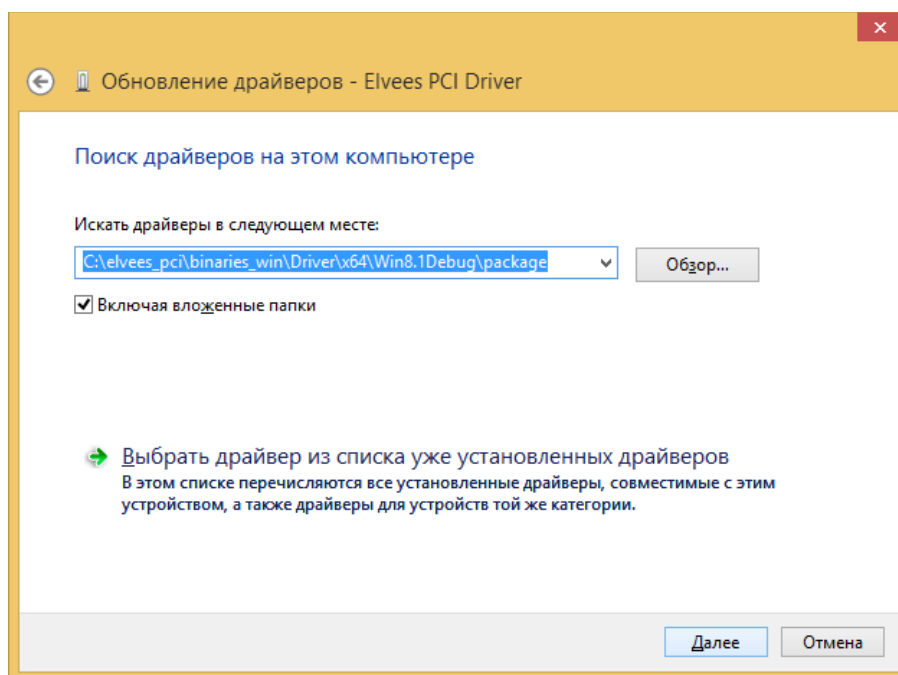


Рис. 6. Мастер обновления драйверов. Страница 2.

Если Windows сообщит о том, что драйвер не имеет постоянной цифровой подписи, выбрать пункт «Всё равно установить драйвер».

После установки драйвера в Диспетчере устройств появится узел Elvees Device\Elvees PCI Driver, как показано на Рис. 7.

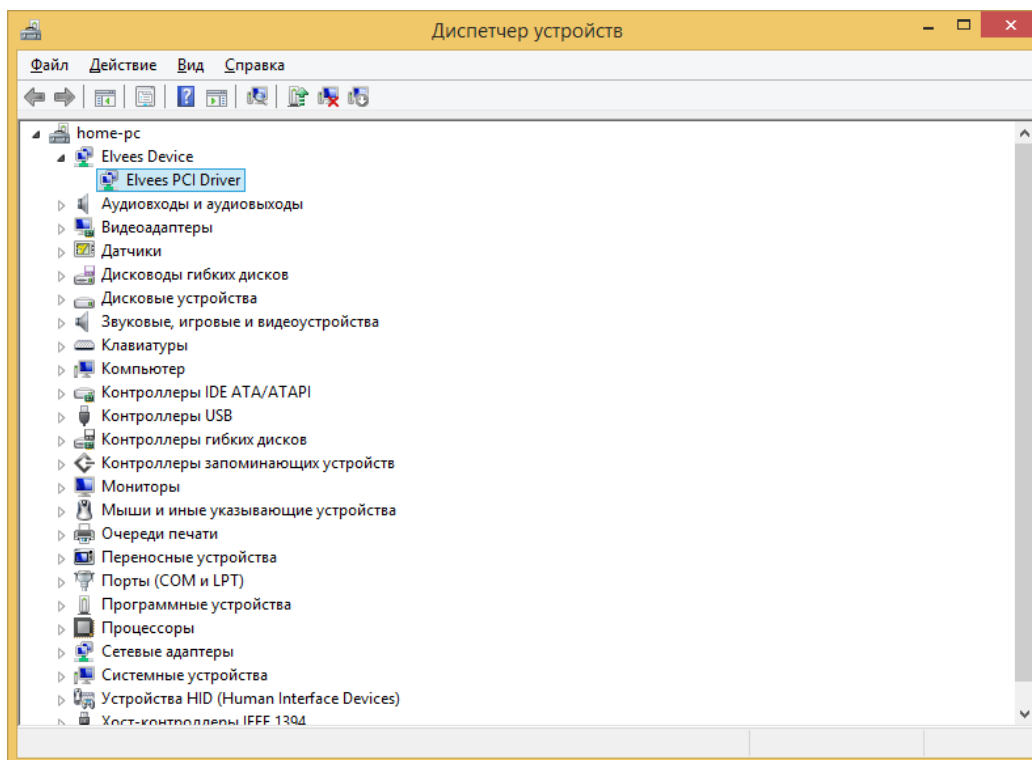


Рис. 7. Контроллер PCI после установки драйверов.

Установка драйвера для ОС Linux

Драйвер для Linux в виде бинарных кодов не поставляется, так как обычно требуется пересборка драйвера под конкретную версию ядра Linux (во всяком случае, безопаснее собрать драйвер из исходных кодов при его установке). Поэтому перед установкой драйвера Elvees PCI его необходимо собрать по инструкции, приведённой ниже, в главе «Сборка драйвера для ОС Linux».

Драйвер Elvees PCI для Linux в свою очередь использует драйвер uio. Он должен быть загружен до использования драйвера Elvees PCI. Проверить, загружен ли uio, можно с помощью команды:

```
lsmod | grep uio
```

Если вывод данной команды не пустой, значит модуль загружен.

Если uio не загружен, необходимо выполнить команду (с правами суперпользователя):

```
modprobe uio
```

Далее требуется определить физический адрес платы в системе с помощью команды:

```
lspci -vv
```

Вывод данной команды будет содержать информацию обо всех платах PCI, подключенных к системе. Среди этих строк нужно найти информацию о плате с микросхемой Элвис. Она будет иметь вид:

```
04:07.0 Communication controller: Temporal Research Ltd Device 680c (rev 01)
```

```
Subsystem: Device 0002:0000
```

```
Control: I/O- Mem+ BusMaster+ SpecCycle- MemWINV- VGASnoop- ParErr- Stepping-  
SERR- FastB2B- DisINTx-
```

```
Status: Cap- 66MHz- UDF- FastB2B+ ParErr- DEVSEL=medium >TAbort- <TAbort-  
<MAbort+ >SERR- <PERR- INTx-
```

```
Latency: 32 (8000ns min, 250ns max)
```

```
Interrupt: pin A routed to IRQ 3
```

```
Region 0: Memory at f8000000 (32-bit, prefetchable) [size=64M]
```

В последней строке указан физический адрес начала окна PCI платы в шестнадцатиричной системе исчисления (f8000000 в данном примере).

Теперь необходимо загрузить драйвер Elvees PCI (pe-uio) с помощью следующей команды от имени суперпользователя (команду необходимо выполнять из каталога, содержащего собранный файл pe-uio.ko):

```
insmod pe-uio.ko phys_addr_hi="0xF800"
```

В параметре `phys_addr_hi` необходимо указать старшие 16 бит физического адреса, полученного в предыдущем пункте.

Кросс-платформенная библиотека доступа libre

Кросс-платформенная библиотека `libre` скрывает различия в реализации драйвера Elvees PCI от прикладной программы и даёт возможность разрабатывать платформо-независимый прикладной исходный код для Windows и Linux.

Сборка библиотеки может быть осуществлена с помощью любого компилятора языка Си под выбранную платформу (Windows или Linux), но наиболее простой путь состоит в использовании свободно распространяемой программы Qt Creator (версия не важна), входящей в инструментарий Qt SDK.

При установке инструментария под ОС Windows при выборе состава устанавливаемых программ нужно обязательно отметить для установки пакет MinGW. Пример выбора устанавливаемых опций, достаточных для работы с `libre`, приведён на Рис. 8.

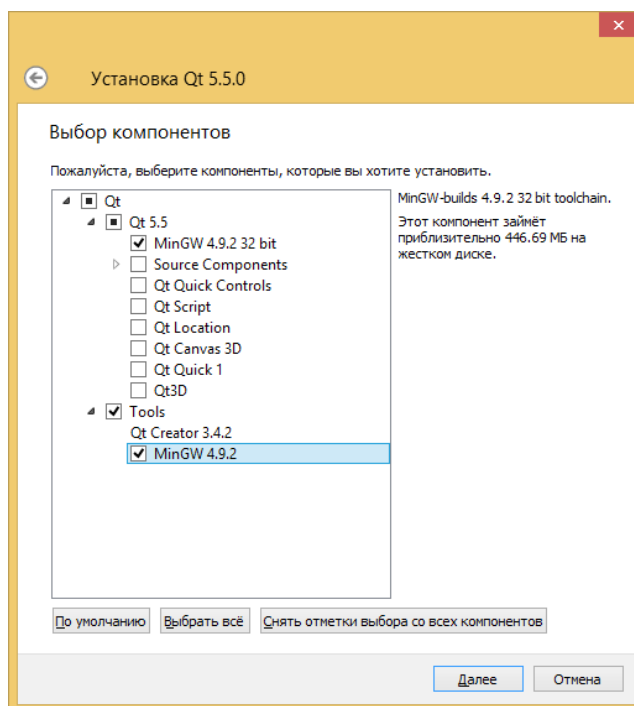


Рис. 8. Выбор опций установки QtSDK

В директории с исходными кодами `libre` прилагается файл проекта `libre.pro` для Qt Creator. Его необходимо открыть в Qt Creator, выбрав пункт главного меню Файл → Открыть файл или проект.

Далее для проекта рекомендуется установить тип сборки «Выпуск» (Release).

Собрать `libre` можно, например, выбрав команду главного меню Собрать → Собрать всё.

Ниже описаны функции, предоставляемые библиотекой `libre`.

1) Инициализация библиотеки доступа к микросхеме

```
int libpe_init(void);
```

2) Корректное отключение библиотеки доступа

```
void libpe_close(void);
```

3) Чтение участка памяти (в том числе регистров) из адресного пространства микросхемы.

Параметр `buffer` должен содержать указатель на буфер, в котором первым 32-хбитным элементом должен быть адрес начала участка памяти, остальная часть буфера может быть произвольной.

Параметр `size` должен содержать размер буфера в байтах, включая первый элемент (адрес). Из-за особенностей организации доступа через PCI параметр `size` должен содержать значение, выравненное на 4, два младших бита всегда игнорируются. Функция возвращает количество успешно считанных байтов.

```
int libpe_read(void *buffer, size_t size);
```

4) Запись участка памяти (в том числе регистров) в адресное пространство микросхемы.

Параметр `buffer` должен содержать указатель на буфер, в котором первым 32-хбитным элементом должен быть адрес начала участка памяти, остальная часть буфера должна содержать данные для записи.

Параметр `size` должен содержать размер буфера в байтах, включая первый элемент (адрес). Из-за особенностей организации доступа через PCI параметр `size` должен содержать значение, выравненное на 4, два младших бита всегда игнорируются. Функция возвращает количество успешно записанных байтов.

```
int libpe_write(void *buffer, size_t size);
```

5) Функция имеет реализацию только для ОС Linux. Для ОС Windows она возвращает нулевой указатель.

Функция возвращает указатель на начало адресного пространства микросхемы.

Особенностью драйвера под Linux является то, что память микросхемы отображается в виртуальное пространство процесса, благодаря чему возможен доступ к памяти микросхемы напрямую по указателю. Этот метод при правильном использовании может дать большую производительность пользовательской программе, при этом также могут использоваться функции `libpe_read` и `libpe_write`.

При использовании доступа по указателю необходимо также учитывать особенности работы с микросхемой через PCI: при считывании одного слова (или байта) всегда происходит считывание 16 слов. Таким образом могут быть сброшены некоторые биты в регистрах, которые пользователь не намеревался читать (имеются в виду статусные биты, сбрасываемые чтением). Сбрасываться иногда могут даже биты в одном регистре, если значение регистра копируется побайтно, например, с помощью функции `memscrw`.

```
void *libpe_raw_pointer_linux(void);
```

TEST-SWIC: пример использования библиотеки `libpe` на языке «Си»

Пример `test-swic` демонстрирует использование библиотеки `libpe` на языке программирования «Си», а заодно и позволяет проверить работоспособность платы отладочного модуля.

Сборка примера осуществляется в IDE Qt Creator. Необходимо открыть в IDE файл проекта `test-swic/test-swic.pro`, выбрать тип сборки проекта «Выпуск» (Release) и собрать проект.

Запускать программу `test-swic` необходимо с правами администратора (и в Windows, и в Linux).

Программа имеет следующий синтаксис вызова:

test-swic <кол-во повторов> <скорость обмена> <номер перед. канала> <номер прин. канала>

<кол-во повторов> - количество повторов теста; за один прогон выполняется передача одного пакета (131064 байт) с контролем содержимого;

<скорость обмена> - скорость обмена, устанавливаемая после установления соединения для передачи пользовательского пакета, в Мбит/с; значение должно быть больше или равно 10 и кратно 5;

<номер перед. канала> - номер канала, передающего пакет;

<номер прин. канала> - номер канала, принимающего пакет.

Для успешного прохождения теста передающий и принимающий каналы, указанные в параметрах программы должны быть соединены с помощью прилагаемого кабеля SpaceWire.

TEST-MEM: пример использования библиотеки libre на языке «Си++»

Пример test-mem демонстрирует использование библиотеки libre на языке программирования «Си++», а заодно и позволяет проверить работоспособность платы отладочного модуля. Программа сначала «расписывает» память DPRAM микросхемы счётчиком, а затем считывает записанные данные и проверяет их корректность.

Одновременно замеряется скорость записи и скорость считывания данных.

Сборка примера осуществляется в IDE Qt Creator. Необходимо открыть в IDE файл проекта test-mem\test-mem.pro, выбрать тип сборки проекта «Выпуск» (Release) и собрать проект.

Запускать программу test-mem необходимо с правами администратора (и в Windows, и в Linux).

Программа имеет следующий синтаксис вызова:

test-mem [размер блока]

[размер блока] — размер блока данных, записываемого/считываемого из DPRAM за одно обращение к функциям библиотеки libre; необязательный параметр, если он не задан, то используются блоки размером 4096 байт.

TEST-READ-SPEED: замер скорости чтения из DPRAM микросхемы

Пример test-read-speed позволяет замерить скорость чтения из DPRAM микросхемы.

Сборка примера осуществляется в IDE Qt Creator. Необходимо открыть в IDE файл проекта test-mem\test-mem.pro, выбрать тип сборки проекта «Выпуск» (Release) и собрать проект.

Запускать программу test-read-speed необходимо с правами администратора (и в Windows, и в Linux).

Программа имеет следующий синтаксис вызова:

test-read-speed [размер блока] [количество прогонов]

[размер блока] — размер блока данных, считываемого из DPRAM за одно обращение к функциям библиотеки libre.

[количество прогонов] — количество чтений всего блока памяти DPRAM.

TEST-WRITE-SPEED: замер скорости записи в DPRAM микросхемы

Пример test-write-speed позволяет измерить скорость записи в DPRAM микросхемы.

Сборка примера осуществляется в IDE Qt Creator. Необходимо открыть в IDE файл проекта test-mem\test-mem.pro, выбрать тип сборки проекта «Выпуск» (Release) и собрать проект.

Запускать программу test-read-speed необходимо с правами администратора (и в Windows, и в Linux).

Программа имеет следующий синтаксис вызова:

test-write-speed [размер блока] [количество прогонов]

[размер блока] — размер блока данных, записываемого в DPRAM за одно обращение к функциям библиотеки libre.

[количество прогонов] — количество записей всего блока памяти DPRAM.

Сборка драйвера из исходных кодов

Сборка драйвера для ОС Windows

Сборка драйвера для ОС Windows может быть осуществлена только в IDE Microsoft Visual Studio 2013 (самая свежая версия на момент написания документа). Условно-бесплатная 30-дневная версия программы доступна для скачивания в сети Интернет.

Помимо Visual Studio необходимо также установить инструментарий Windows Development Kit. Этот набор инструментов распространяется бесплатно, и также доступен для скачивания в Интернет. При разработке использовалась версия 8.1 данного инструментария.

Для сборки драйвера Elvees PCI необходимо открыть файл driver_src_win\elvees_pci.sln в Visual Studio, выбрать целевую операционную систему для сборки (например, Windows 8.1 Debug, x64) и выбрать пункт главного меню Сборка → Перестроить решение. В качестве целевой версии необходимо выбирать именно отладочную (Debug), поскольку при её сборке генерируется тестовый сертификат, используемый для подписи драйвера. Целевую операционную систему можно выбрать в ниспадающих списках на панели инструментов (они, как правило, присутствуют по умолчанию) либо в Диспетчере конфигураций, который можно вызвать командой главного меню Сборка → Диспетчер конфигураций...

Генерация файлов трассировки

Трассировка драйвера позволяет получить трассу выполнения драйвера, в том числе в процессе эксплуатации пользователем. Трасса генерируется, только если включена программа, «слушающая» трассировку. В противном случае, по заверениям специалистов компании Microsoft, накладной расход минимален и не оказывает влияния на время исполнения функций драйвера.

Драйвер Elvees PCI для Windows уже собран с возможностью трассировки. Файлы, необходимые для расшифровки трассировки, лежат в папке `binaries_win/Driver/trace`. Но в случае внесения изменений в драйвер, для корректной работы трассировки эти файлы необходимо регенерировать. Это выполняется командой:

```
tracedb -f <путь к файлу elvees_pci.pdb> -p <путь к папке с файлами расшифровки>
```

Программу `tracedb` необходимо взять из комплекта разработчика Windows (WDK), корневая директория которого после установки обычно находится в директории `Program files(x86)\Windows Kit\<номер версии>`.

Просмотр и запись трассы

Просмотр трассы выполняется программой `TraceView` из комплекта разработчика Windows (WDK).

После запуска `TraceView` необходимо выбрать пункт основного меню `File → Create New Log Session` (см Рис. 9).

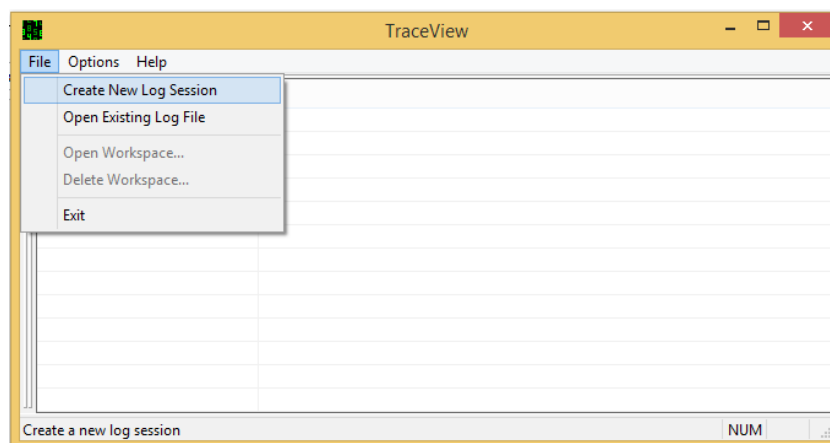


Рис. 9: Главное окно TraceView

В появившемся окне `Create New Log Session` (Рис. 10) необходимо нажать кнопку `Add Provider`.

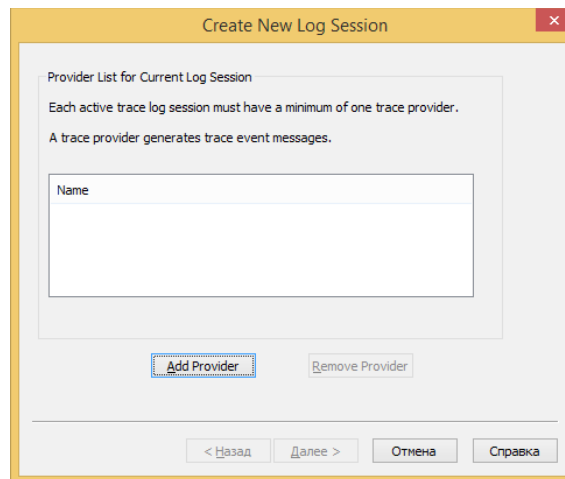


Рис. 10: Окно Create New Log Session

В появившемся окне Provider Control GUID Setup (Рис. 11) необходимо выбрать пункт CTL (Control GUID) File, указать путь к файлу elvees_pci.ctl и нажать кнопку ОК.

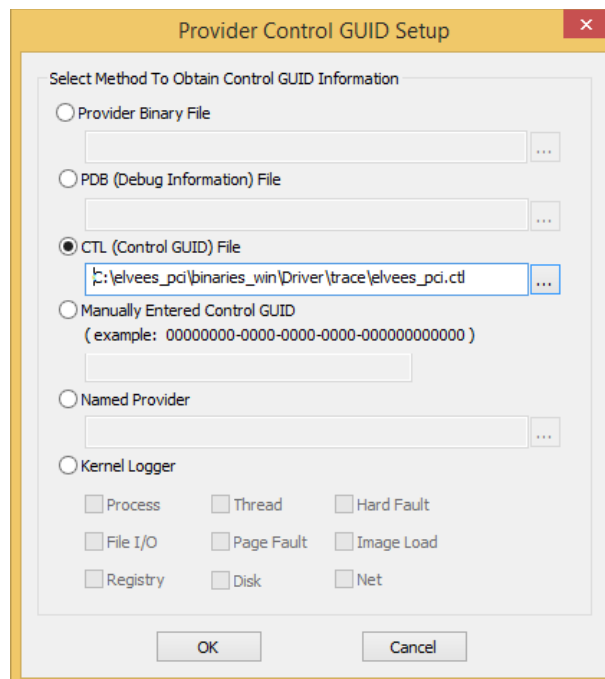


Рис. 11: Окно Provider Control GUID Setup

Появится окно Format Information Source Select (Рис. 12). В нём выбрать пункт Select TMF Files и нажать ОК.

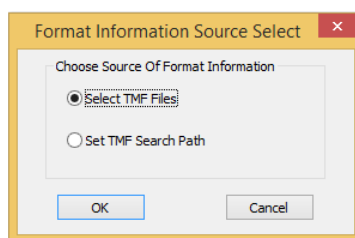


Рис. 12: Окно *Format Information Source Select*

В окне Trace Format Information Setup (Рис. 13) необходимо нажать на кнопку Add и указать файлы расшифровки трассировки, сгенерированные программой traserpdb (с расширением .tmf). Далее нажать кнопку Done.

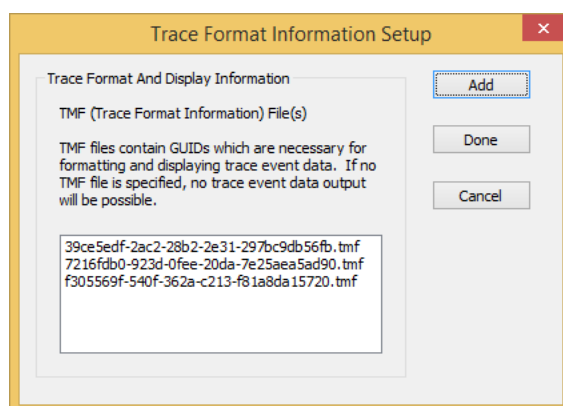


Рис. 13: Окно *Trace Format Information Setup*

Далее фокус вернётся в окно Create New Log Session, в котором нужно нажать кнопку «Далее».

В появившемся окне Log Session Options (Рис. 14) можно выбрать пункт «Log Trace Event Data To File» при необходимости сохранить журнал трассировки в файл (удобно, если трассировка выполняется удалённо, пользователем). Файл журнала трассировки можно в офф-лайн также просмотреть в программе TraceView.

Для завершения процесса подключения TraceView к трассировке драйвера необходимо нажать кнопку «Готово».

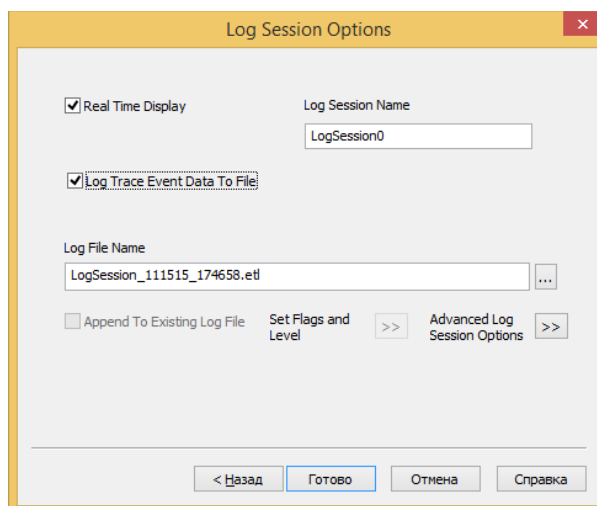


Рис. 14: Окно Log Session Options

Сборка драйвера для ОС Linux

Для сборки драйвера Elvees PCI для Linux необходимо дать команду make в директории driver_src_linux. В результате успешной сборки должен появиться в этой же директории файл re-urio.ko, который и является бинарной версией драйвера.