



Общество с ограниченной ответственностью
«СВД Встраиваемые Системы» (ООО «СВД ВС»)
ул. Кузнецовская, д. 19, г. Санкт-Петербург, 196128

тел. (812) 346-89-56, факс (812) 346-89-53
www.kpda.ru sales@kpda.ru

Руководство пользователя

Руководство пользователя пакета поддержки платы OrangePi-PC на основе интегральной микросхемы Allwinner-H3 под ЗОСРВ «Нейтрино»

2 февраля 2024 г.

Содержание

Руководство пользователя	2
Содержание	2
BSP для OrangePi-PC Allwinner-H3	3
Основные термины и сокращения	4
Общие сведения	5
Состав и структура	6
Установка и настройка	7
Загрузка образа ОС	8
Описание компонентов	11
startup-orangepi	12
devnp-sun8i.so	14
Проверка функционирования	16
Известные особенности	24
Журнал изменений BSP для OrangePi-PC Allwinner-H3	25

BSP для OrangePi-PC Allwinner-H3

Аннотация

Данный документ представляет собой руководство пользователя для пакета поддержки платы OrangePi-PC на основе интегральной микросхемы Allwinner-H3 под ЗОСРВ «Нейтрино».

Процессорная архитектура: ARMv7 Little Endian

Поддерживаемые устройства:

- Одноплатный компьютер [OrangePi-PC\(v1.3\)](#)
- Одноплатный компьютер [OrangePi-PC-Plus\(v1.1\)](#)

Документ включает следующие разделы:

- Основные термины и сокращения
- Общие сведения
- Состав и структура
- Установка и настройка
- Загрузка образа ОС
- Описание компонентов
- Проверка функционирования
- Известные особенности
- Журнал изменений

В разделе «Основные термины и сокращения» приведена расшифровка сокращений, используемых в данном документе.

В разделе «Общие сведения» приведены назначение и функции ППМ, а также описание используемых технических средств.

В разделе «Состав и структура» приведены сведения о структуре ППМ, его составных частях и назначении составных частей.

В разделе «Установка и настройка» приведены указания по установке и настройке компонентов сред разработки и сред исполнения, необходимых для функционирования ППМ и сборки загружаемого образа ЗОСРВ «Нейтрино».

В разделе «Загрузка образа ОС» приведены сведения о вариантах запуска загружаемого образа ЗОСРВ «Нейтрино» на целевой системе.

В разделе «Описание компонентов» приведены описания компонентов ППМ.

В разделе «Проверка функционирования» приведены указания по проверке работоспособности пакета поддержки.

В разделе «Известные особенности» приведены известные особенности и ограничения текущей версии пакета поддержки.

Раздел	Описание
Основные термины и сокращения	Основные термины и сокращения, используемые в данном руководстве
Общие сведения	Общие сведения
Состав и структура	Состав и структура
Установка и настройка	Установка и настройка
Загрузка образа ОС	Загрузка образа ОС
Описание компонентов	Описание компонентов, используемых в ППМ
Проверка функционирования	Проверка функционирования ППМ OrangePi-PC Allwinner-H3 под ЗОСРВ «Нейтрино».
Известные особенности	Известные особенности и ограничения текущей версии пакета поддержки
Журнал изменений BSP для OrangePi-PC Allwinner-H3	Журнал и история изменений компонентов пакета поддержки

Основные термины и сокращения

Основные термины и сокращения, используемые в данном руководстве

ЗОСРВ

защищенная операционная система реального времени

ПМ

процессорный модуль

ППМ

BSP

пакет поддержки микропроцессора

Общие сведения

Общие сведения

Пакет поддержки платы OrangePi-PC (ППМ) предназначен для обеспечения запуска и функционирования ЗОСРВ «Нейтрино» на процессорном модуле (ПМ) на основе интегральной микросхемы Allwinner-H3.

Для работы с ППМ необходим инструментальный ПК с установленным комплектом разработчика для ЗОСРВ «Нейтрино» в варианте `_Linux Host_` и целевая плата OrangePi-PC на основе интегральной микросхемы Allwinner-H3.

ППМ совместим со следующими редакциями ЗОСРВ «Нейтрино»: **2021, 2020** (ограниченно).

ППМ обеспечивает поддержку и функционирование следующих блоков и интерфейсов интегральной микросхемы Allwinner-H3:

- процессорные ядра Cortex-A7 в режиме SMP
- контроллер прерываний GICv2
- контроллер оперативной памяти DDR
- UART
- USB (EHCI/OHCI)
- Ethernet
- SD/eMMC

Состав и структура

Состав и структура

ППМ представляет собой архив, имеющий формат `.tar.gz` и имя вида «`kpda-bsp-allwinner-h3-orangepipc-(версия-ОС)-(формат)-(дата)-(хэш-git).tar.gz`». Назначение полей:

- (версия-ОС) - версия ЗОСРВ для которой предназначен BSP
- (формат) - формат ППМ: `src` для версии с исходными кодами, `bin` для версии с бинарными компонентами
- (дата) - дата сборки и запаковки BSP в формате ГГГГММДД
- (хэш-git) - номер или хэш коммита системы контроля версий git

Архив ППМ содержит следующие компоненты:

```
kpda-bsp-allwinner-h3-orangepipc/ # Каталог верхнего уровня
|--> images/                       # Каталог с загрузочным образом (IFS) и файлами построения
|   |--> ifs-h3-orangepi.bin       # Загружаемый образ ЗОСРВ «Нейтрино»
|   `--> *.build                   # Файлы построения загрузочных образов
|--> install/                      # Бинарные компоненты, собираемые из исходных кодов
|   `--> usr/include/              # Заголовочные файлы
`--> src/                           # Исходные коды компонентов ППМ
```

Установка и настройка

Установка и настройка

Распаковывать архив с ППМ следует на инструментальной машине *x86_64* с ОС *Linux* с установленным комплектом разработчика ЗОСРВ «Нейтрино».

Для распаковки ППМ следует скопировать архив в рабочий каталог и выполнить команду:

```
$ tar xzf kpda-bsp-allwinner-h3-orangepipc-*.tar.gz
```

Для сборки ППМ выполните следующие команды:

```
$ cd kpda-bsp-allwinner-h3-orangepipc  
$ make
```

В результате будет произведена сборка ППМ и подготовлен загрузочный образ ЗОСРВ «Нейтрино» `ifs-h3-orangepi.bin`.

Загрузка образа ОС

Загрузка образа ОС

- Загрузка образа по сети
- Загрузка образа с SD карты памяти
- Результат загрузки образа
- Сохранение варианта загрузки по-умолчанию

Загрузка образа по сети

Для загрузки образа ЗОСРВ «Нейтрино» на ПМ по сети используется протокол TFTP. На инструментальной системе необходимо настроить TFTP сервер и скопировать полученный загрузочный образ `ifs-h3-orangepi.bin` в каталог, использующийся TFTP сервером.

Затем следует подключить отладочный USB-UART ПМ к USB порту инструментального ПК, проверить наличие устройства `‘/dev/ttyUSB*‘` и запустить терминальную программу, например, `minicom` со следующими параметрами: скорость 115200, 8 бит, без контроля четности, 1 стоп-бит, порт `‘/dev/ttyUSB0‘`.

После включения питания ПМ в терминальной программе должен присутствовать отладочный вывод загрузчика U-Boot. Например:

```
U-Boot SPL 2020.04-orangepi (Oct 26 2020 - 16:27:23 +0800)
DRAM: 1024 MiB
Trying to boot from MMC1

U-Boot 2020.04-orangepi (Oct 26 2020 - 16:27:23 +0800) Allwinner Technology

CPU: Allwinner H3 (SUN8I 1680)
Model: Xunlong Orange Pi PC
DRAM: 1 GiB
MMC: mmc@1c0f000: 0
Loading Environment from FAT... Unable to use mmc 0:1... In: serial@1c28000
Out: serial@1c28000
Err: serial@1c28000
Net: phy interface0
eth0: ethernet@1c30000
starting USB...
Bus usb@1c1a000: USB EHCI 1.00
Bus usb@1c1a400: USB OHCI 1.0
Bus usb@1c1b000: USB EHCI 1.00
Bus usb@1c1b400: USB OHCI 1.0
Bus usb@1c1c000: USB EHCI 1.00
Bus usb@1c1c400: USB OHCI 1.0
Bus usb@1c1d000: USB EHCI 1.00
Bus usb@1c1d400: USB OHCI 1.0
scanning bus usb@1c1a000 for devices... 1 USB Device(s) found
scanning bus usb@1c1a400 for devices... 1 USB Device(s) found
```

```
scanning bus usb@1c1b000 for devices... 1 USB Device(s) found
scanning bus usb@1c1b400 for devices... 1 USB Device(s) found
scanning bus usb@1c1c000 for devices... 1 USB Device(s) found
scanning bus usb@1c1c400 for devices... 1 USB Device(s) found
scanning bus usb@1c1d000 for devices... 1 USB Device(s) found
scanning bus usb@1c1d400 for devices... 1 USB Device(s) found
    scanning usb for storage devices... 0 Storage Device(s) found
Autoboot in 1 seconds, press <Space> to stop
=>
```

Для загрузки образа ЗОСРВ «Нейтрино» следует остановить загрузку и в консоли загрузчика U-Boot выполнить следующие команды:

- Указать IP-адрес TFTP сервера:

```
=> setenv serverip 172.16.0.16
=> setenv gatewayip 172.16.0.16
```

- Установить IP-адрес модулю, для организации обмена по протоколу TFTP:

```
=> setenv ipaddr 172.16.10.44
```

- Загрузить образ по протоколу TFTP и запустить его:

```
=> tftpboot 0x41000000 ifs-h3-orangepi.bin
=> go 0x41000000
```

Загрузка образа с SD карты памяти

Для загрузки образа ЗОСРВ «Нейтрино» с карты памяти SD/microSD, скопируйте заранее подготовленный загрузочный образ `ifs-h3-orangepi.bin` на FAT раздел карты памяти и после включения питания ПМ в консоли загрузчика U-Boot выполните следующие команды:

```
=> fatload mmc 0 0x41000000 ifs-h3-orangepi.bin
=> go 0x41000000
```

Результат загрузки образа

В результате выполнения приведенных выше команд должен запуститься образ ЗОСРВ, выведены сообщения о ходе загрузки и запущен командный интерпретатор. Пример вывода:

```
System page at phys:40022000 user:fc410000 kern:fc410000
Starting next program at vfe0820ac
cpu_startnext: cpu0 -> fe0820ac
cpu_startnext: invoking vstart:4100a720 sysp:fc410000 eip:fe0820ac cpu:00000000
VFPv3: fpsid=41023075
```

```
coproc_attach(10): attach fe099c78 (fe09b750)
coproc_attach(11): attach fe099c78 (fe09b750)
Welcome to KPDA Neutrino on the OrangePi PC Allwinner H3 (ARMv7 Cortex-A7 QuadCore) - Board
Starting common servers...
Starting UART serial driver...
Starting USB host driver...
Starting SD/eMMC driver...
Starting Network driver...
#
```

Сохранение варианта загрузки по-умолчанию

Для сохранения параметров загрузки и выполнения требуемого вариант загрузки по-умолчанию автоматически при включении питания модуля, в консоли U-Boot установите переменную окружения **bootcmd** и сохраните значение переменных с помощью команды **saveenv**, например:

```
=> setenv kpdaboot_net 'tftpboot 0x41000000 ifs-h3-orangepi.bin; go 0x41000000'
=> setenv bootcmd run kpdaboot_net
=> saveenv
```

Описание компонентов

Описание компонентов, используемых в ППМ

- [Список компонентов, входящих в состав ППМ](#)
- [Список стандартных компонентов, используемых в ППМ](#)

Список компонентов, входящих в состав ППМ

Описание компонентов, входящих в состав ППМ.

Раздел	Описание
startup-orangepi	Модуль инициализации и запуска системы OrangePi-PC Allwinner-H3
devnp-sun8i.so	Драйвер Ethernet контроллеров семейства процессоров Allwinner sun8i (на базе Cortex-A7)

Список стандартных компонентов, используемых в ППМ

Описание стандартных компонентов, не входящих в состав ППМ.

Компонент	Описание
devc-ser8250	Менеджер последовательных 8250-совместимых каналов связи
devu-ehci.so	Драйвер для USB-контроллеров с усовершенствованным интерфейсом хост-контроллера (EHCI)
devu-ohci.so	Драйвер для USB-контроллеров с открытым интерфейсом хост-контроллера (Open Host Controller Interface, OHCI)
devb-sdmmc	Драйвер для SD/eMMC контроллеров



Опции запуска по умолчанию для указанных компонентов можно посмотреть на странице "[Проверка функционирования](#)".

startup-orangepi

Модуль инициализации и запуска системы OrangePi-PC Allwinner-H3

Синтаксис:

```
startup-orangepi [универсальные опции библиотеки libstartup]
```

Платформы:

ЗОСРВ «Нейтрино»

Целевые архитектуры:

armv7

Опции:

Все модули инициализации систем в ЗОСРВ «Нейтрино» слинкованы с одной из стандартных библиотек *libstartup* и поддерживают её набор универсальных опций. Также у любого из них могут существовать дополнительные эксклюзивные опции. Ниже рассматриваются универсальные опции библиотек:



Различные модули инициализации могут перекрывать универсальные опции, а также поддерживать дополнительные, специфичные для конкретных систем, опции. Приоритет действия следующий:

1. Опции модулей инициализации для конкретных архитектур/систем.
2. Опции библиотеки для конкретных архитектур.
3. Универсальные опции библиотеки.

Универсальные опции библиотеки libstartup

Стандартные опции библиотеки *libstartup* описаны на странице описания модуля *startup* в разделе Универсальные опции библиотеки *libstartup*.

Описание:

Модуль инициализации *startup-orangepi* обеспечивает запуск и начальную инициализацию отладочной платы OrangePi-PC на базе микропроцессора Allwinner-H3.

Примеры:

Базовый вариант запуска с включением вывода отладочной информации:

```
startup-orangepi -v
```

Классификация:

Базовые подсистемы ЗОСРВ «Нейтрино», Драйверы

devnp-sun8i.so

Драйвер Ethernet контроллеров семейства процессоров Allwinner sun8i (на базе Cortex-A7)

Синтаксис:

```
io-pkt-* -d sun8i [опция[,опция ...]] ...
```



Для разделения опций используется запятая (но не пробелы).

Платформы:

ЗОСРВ «Нейтрино»

Целевые архитектуры:

armv7

Описание:

Драйвер *devnp-sun8i.so* управляет Ethernet контроллерами семейства процессоров Allwinner sun8i (на базе Cortex-A7). Это драйвер io-pkt-*; имена интерфейсов назначаются в виде *sunX*, где *X* – целое число.

Поддерживаемые устройства:

- Ethernet контроллер процессора Allwinner H3

Примеры:

Запуск io-pkt-* с использованием драйвера *devnp-sun8i.so* и полного стека TCP/IP:

```
io-pkt -d sun8i  
ifconfig sun0 192.168.0.10
```

Классификация:

Базовые подсистемы ЗОСРВ «Нейтрино», Драйверы

Тематические ссылки:

devn-*, devnp-*, ifconfig, io-pkt-*, nicinfo

Проверка функционирования

Проверка функционирования ППМ OrangePi-PC Allwinner-H3 под ЗОСРВ «Нейтрино».

Список проверок:

- Базовые проверки функционирования ОС
 - Проверка запуска на ПМ ядра ЗОСРВ в составе загружаемого образа
 - Проверка запуска компонентов
 - Проверка функционирования контроллера DDR памяти
 - Проверка функционирования контроллера прерываний
 - Проверка функционирования таймера
 - Проверка работы ЗОСРВ на ЦП в режиме симметричной мультипроцессорности (SMP) с поддержкой всех CPU ядер типа Cortex-A7
- Проверки интерфейсов
 - Проверка драйвера контроллера UART
 - Проверка драйвера контроллера Gigabit Ethernet
 - Проверка драйвера контроллера USB
 - Проверка драйвера контроллера SD/MMC

Тестовый стенд

Тестовый стенд представляет из себя отладочный модуль OrangePi-PC, соединённый с инструментальным ПК по интерфейсам RS-232 и Ethernet.

На инструментальный ПК должно быть установлено:

- ОС семейства GNU Linux, например, Ubuntu 20.04
- Комплект разработчика для ЗОСРВ «Нейтрино»
- ППМ OrangePi-PC Allwinner-H3
- Терминальная программа для отладочной консоли модуля OrangePi-PC, например, minicom

Подготовка

Перед проведением проверок необходимо запустить ЗОСРВ «Нейтрино» на отладочном модуле OrangePi-PC в соответствии с главой документации [Загрузка образа ОС](#).

Дополнительные тестовые программы *rr*, *ticksize* и другие не входят в состав ППМ и должны быть самостоятельно загружены из [публичного git-репозитория](#) и собраны командой:

```
$ cd utils
$ make install
```

Затем следует скопировать *armle-v7* версии тестовых программ на SD карту памяти и подключить её к OrangePi-PC или передать файлы по сети с помощью FTP или NFS.

Базовые проверки функционирования ОС

Проверка запуска на ПМ ядра ЗОСРВ в составе загружаемого образа

В последовательном терминале выполнить команду:

```
# uname -s
```

Проверка считается выполненной успешно, если в терминале будет выведено сообщение вида:

```
KPDA
```

Проверка запуска компонентов

Для вывода информации о запущенных компонентах выполнить команду `pidin ar`:

```
# pidin ar
pid Arguments
 1 procnto-smp -vvv -ae
 2 slogger -s 16384k
 3 pipe
 4 devc-ser8250 -e -F -b115200 -c24000000/16 0x01C28000\^{}2,32
 5 io-usb -dehci ioport=0x01c1b000,irq=106 -dohci ioport=0x01c1b400,irq=107
    -dehci ioport=0x01c1c000,irq=108 -dohci ioport=0x01c1c400,irq=109
    -dehci ioport=0x01c1d000,irq=110 -dohci ioport=0x01c1d400,irq=111
 6 random -t
 7 io-pkt-v4-hc-ksz -dsun8i
4103 sh
4104 pidin ar
```

Проверка функционирования контроллера DDR памяти

Для вывода информации о процессоре и доступной оперативной памяти выполнить команду `pidin info`:

```
# pidin info
CPU:ARM Release:trunk FreeMem:992Mb/1024Mb BootTime:Jan 01 00:00:00 UTC 1970
Processes: 8, Threads: 28
Processor1: 1091551349 Cortex A7 228MHz FPU
Processor2: 1091551349 Cortex A7 228MHz FPU
Processor3: 1091551349 Cortex A7 228MHz FPU
Processor4: 1091551349 Cortex A7 228MHz FPU
```

Для проверки функционирования контроллера оперативной памяти используется утилита *memcheck*. Ее требуется запустить и дождаться завершения проверки:

```
# /tmp/memcheck
```

Проверка считается выполненной успешно, если выведено сообщение:

PASS: all memory allocated

Проверка функционирования контроллера прерываний

Для вывода информации о используемых прерываниях выполнить команду `pidin ir`:

```
# pidin ir
 pid tid name
  1   1 procnto-smp
  1   2 procnto-smp
  1   3 procnto-smp
  1   4 procnto-smp
      0          0x1b    0 -P- @0xfe08a2c4:0x0
  1   5 procnto-smp
  1   6 procnto-smp
  1   7 procnto-smp
  1   8 procnto-smp
  1   9 procnto-smp
  1  10 procnto-smp
  1  11 procnto-smp
  1  12 procnto-smp
  1  13 procnto-smp
  1  14 procnto-smp
  1  15 procnto-smp
  1  17 procnto-smp
  2   1 slogger
  3   1 pipe
  3   2 pipe
  3   3 pipe
  4   1 devc-ser8250
      1          0x20    0 --- @0x804ab6c:0x805d1c0
  5   1 io-usb
      2          0x6a    0 TP- =PULSE 0x40000008:21 0:0
      3          0x6b    0 TP- =PULSE 0x4000000b:21 0:0
      4          0x6c    0 TP- =PULSE 0x4000000e:21 0:0
      5          0x6d    0 TP- =PULSE 0x40000011:21 0:0
      6          0x6e    0 TP- =PULSE 0x40000014:21 0:0
      7          0x6f    0 TP- =PULSE 0x40000017:21 0:0
  5   2 io-usb
  5   3 io-usb
  5   4 io-usb
  5   5 io-usb
  5   6 io-usb
  5   7 io-usb
  5   8 io-usb
  5   9 io-usb
  5  10 io-usb
  5  11 io-usb
  6   1 random
```

```

6 2 random
6 3 random
7 1 io-pkt-v4-hc-ksz
7 2 io-pkt-v4-hc-ksz
      8          0x72    0 T-- @0x78002d5c:0x818f2c0
7 3 io-pkt-v4-hc-ksz
4104 1 sh
4105 1 pidin

```

Проверка считается пройденной успешно, если вывод утилиты `pidin` содержит информацию об обработчиках прерывания, принадлежащих выполняемым процессам, в формате

```

pid tid Имя процесса
      id прерывания  vector  mask_count  flags  handler:area

```

Проверка функционирования таймера

Для проверки функционирования системного таймера необходимо воспользоваться программой `ticksize`:

```
# /tmp/ticksize
```

Проверка считается выполненной успешно, если вывод данной программы содержит аналогичные строки:

```

System timer ticksize: 1000 mks
Measuring system timer interrupt latency for 10 sec...

-----
System Timer Latency =          0.54 mks
-----

```

Проверка работы ЗОСРВ на ЦП в режиме симметричной мультипроцессорности (SMP) с поддержкой всех CPU ядер типа Cortex-A7

Для вывода информации о количестве процессоров выполнить команду `pidin info`:

```

# pidin info
CPU:ARM Release:trunk FreeMem:992Mb/1024Mb BootTime:Jan 01 00:00:00 UTC 1970
Processes: 8, Threads: 28
Processor1: 1091551349 Cortex A7 228MHz FPU
Processor2: 1091551349 Cortex A7 228MHz FPU
Processor3: 1091551349 Cortex A7 228MHz FPU
Processor4: 1091551349 Cortex A7 228MHz FPU

```

Для проверки функционирования на всех ядрах выполнить следующее:

```
# for cpu in 0 1 2 3; do on -C$cpu pidin -P pidin -fl; done
```

Проверка считается выполненной успешно, если вывод содержит следующие строки:

```
сри
 0
сри
 1
сри
 2
сри
 3
```

Проверки интерфейсов

Проверка драйвера контроллера UART

Для работы с последовательным портом используется драйвер `devc-ser8250`. Настройки последовательного порта по умолчанию: скорость 115200, 8 бит, без контроля четности, 1 стоп-бит. В составе загрузочного образа драйвер запускается автоматически.

В случае успешного запуска драйвера в директории `/dev/` будет создано устройство `ser1`.

```
# ls /dev/ser*
/dev/ser1
```

Для проверки функционирования драйвера UART подключить ПМ к ПК через преобразователь по USB в один из разъемов и запустить терминальную программу `minicom`. В последовательном или удалённом терминале выполнить следующую команду:

```
# echo Test message > /dev/ser1
```

Проверка считается выполненной успешно, если в терминальной программе будет выведено сообщение:

```
Test message
```

Проверка драйвера контроллера Gigabit Ethernet

Для работы с контроллером интерфейса Gigabit Ethernet используется драйвер `devnp-sun8i.so` и менеджер ресурсов `io-pkt-v4-hc-ksz`. В составе загрузочного образа драйвер запускается автоматически. Для вывода информации о сетевых интерфейсах выполнить команду `ifconfig`.

В случае успешного запуска драйвера будет создан сетевой интерфейс `xzynq0` и в результате выполнения команды `ifconfig` на экран будут выведены данные, аналогичные следующим, с указанием текущего `ip`-адреса, маски и широковещательного адреса:

```
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 33192
    inet 127.0.0.1 netmask 0xff000000
sun0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    address: f2:b4:15:84:4c:8b
    media: Ethernet autoselect (100baseTX full-duplex)
    status: active
    inet <ip адрес> netmask <маска> broadcast <широковещательный адрес>
```

Для проверки функционирования сетевого обмена выполнить следующую команду, инициирующую отправку ICMP пакетов существующему сетевому узлу:

```
# ping -c 3 <ip получателя>
PING <ip получателя> (<ip получателя>): 56 data bytes
64 bytes from <ip получателя>: icmp_seq=0 ttl=64 time=0 ms
64 bytes from <ip получателя>: icmp_seq=1 ttl=64 time=0 ms
64 bytes from <ip получателя>: icmp_seq=2 ttl=64 time=0 ms

----<ip получателя> PING Statistics----
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 0/0/0 ms    variance = 0 ms\^{ }2
```

Проверка считается выполненной успешно, если все 3 пакета были успешно получены (процент потерь равен нулю).

Проверка драйвера контроллера USB

Для работы с контроллером интерфейса USB используется драйвера `devu-ehci.so`, `devu-ohci.so` и менеджер ресурсов `io-usb`. В составе загрузочного образа драйвер запускается автоматически. Для вывода информации о доступных интерфейсах выполнить команду [Проверка драйвера контроллера USB](#).

В случае успешного запуска драйвера в директории `/dev/` будет создано устройство `io-usb`.

```
# ls /dev/io-usb
devu-ehci.so      devu-ohci.so      io-usb
```

Для проверки функционирования драйвера UART подключить ПМ к ПК через преобразователь по USB в один из разъемов и запустить терминальную программу `minicom`. В последовательном или удалённом терминале выполнить следующую команду:

```
# usb
```

Проверка считается выполненной успешно, если в терминальной программе будет выведено сообщение:

```
USB 0 (EHCI) v1.10, v1.01 DDK, v1.01 HCD
USB 1 (OHCI) v1.10, v1.01 DDK, v1.01 HCD
USB 2 (EHCI) v1.10, v1.01 DDK, v1.01 HCD
USB 3 (OHCI) v1.10, v1.01 DDK, v1.01 HCD
USB 4 (EHCI) v1.10, v1.01 DDK, v1.01 HCD
USB 5 (OHCI) v1.10, v1.01 DDK, v1.01 HCD
```

После подключения должно появиться новое устройство:

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

```
Device Address      : 1
Vendor              : 0x8564 (JetFlash)
Product             : 0x1000 (Mass Storage Device)
Class               : 0x00 (Independent per interface)
```

```
USB 1 (OHCI) v1.10, v1.01 DDK, v1.01 HCD
```

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

```
USB 3 (OHCI) v1.10, v1.01 DDK, v1.01 HCD
```

```
USB 4 (EHCI) v1.10, v1.01 DDK, v1.01 HCD
```

```
USB 5 (OHCI) v1.10, v1.01 DDK, v1.01 HCD
```

Проверка драйвера контроллера SD/MMC

Для работы с контроллером SD/MMC используется драйвер `devb-sdmmc`. В составе загрузочного образа драйвер запускается автоматически. В случае успешного запуска драйвера в директории `/dev/` будет создано устройство `sd0`, соответствующее SD-карте, а также устройства, соответствующие разделам. Для управления устройствами могут быть использованы стандартные утилиты для управления и чтения/записи блочных устройств. Проверка считается выполненной успешно, если при выполнении команды `ls /dev/sd0*` отображаются устройства, соответствующие SD-карте и разделам, например:

```
# ls /dev/sd0*
/dev/sd0          /dev/sd0t131     /dev/sd0t177
```

В выводе утилиты `mount` увидеть точку монтирования одного из разделов SD-карты, например:

```
# mount
/dev/sd0t177 on /fs/qnx6 type qnx6
```

Для измерения производительности операции записи и чтения необходимо создать файл и заполнить его случайными значениями с помощью утилиты [Проверка функционирования контроллера DDR памяти](#), при этом измеряя время данной операции через `time`:

```
# dd if=/dev/urandom of=/dev/shmem/test_file bs=1000 count=256
# time dd if=/dev/shmem/test_file of=<точка монтирования>/test_file \
    bs=1000 count=256
```

Выводом данной команды будет являться время и скорость записи:

```
256+0 records in
256+0 records out
256000 bytes transferred in 0.038 secs (6736842 bytes/sec)
    0m00.04s real    0m00.00s user    0m00.00s system
```

Для измерения скорости чтения необходимо скопировать созданный тестовый файл в `/dev/null` с помощью утилиты [Проверка функционирования контроллера DDR памяти](#):

```
# time dd if=<точка монтирования>/test_file of=/dev/null bs=1000 count=256
```

Выводом данной команды будет является время и скорость чтения:

```
256+0 records in  
256+0 records out  
256000 bytes transferred in 0.010 secs (25600000 bytes/sec)  
  0m00.01s real    0m00.00s user    0m00.00s system
```


Известные особенности

Известные особенности и ограничения текущей версии пакета поддержки

Известные особенности отсутствуют

Журнал изменений BSP для OrangePi-PC Allwinner-H3

Журнал и история изменений компонентов пакета поддержки