

USB HOST — тестирование и проверка

Этот документ описывает, **как работает USB HOST тест** в данном проекте и **какие команды можно выполнить вручную в Linux**, чтобы проверить то же самое через консоль.

Тест ориентирован на **embedded Linux плату** и предназначен для проверки: - USB HOST аппаратной части - USB стека Linux - работы USB Mass Storage - базового чтения/записи данных

1. Назначение теста

USB HOST тест проверяет:

- что USB Host контроллер корректно работает
- что на плате определяется USB HUB
- что USB порты могут определять USB Mass Storage устройства
- что данные могут быть записаны и считаны с USB накопителя

Тест **не пытается поддерживать все возможные USB устройства и файловые системы**.
Используется **фиксированный и воспроизводимый сценарий**.

2. Условия тестирования (важно)

Тест выполняется при следующих условиях:

- Тестовое устройство: **SD-карта через USB кардридер**
- Файловая система: **FAT32**
- SD-карта **заранее отформатирована**
- Одно и то же устройство вручную переставляется между портами
- Порты тестируются **по очереди**

Логические порты, используемые в тесте:

Логический порт	Описание
USB1	Нижний USB разъём
USB2	Верхний USB разъём
USB3	UIO1 (линии USB, без разъёма)
USB4	UIO2 (линии USB, без разъёма)

USB3 и USB4 могут быть пропущены, если отсутствует адаптер или разъём.

3. Как работает тест в GUI

Шаг 1. Подключение устройства

Вставьте SD-карту (через USB кардридер) в **любой USB порт** платы.

Шаг 2. Кнопка Detect device

При нажатии выполняются следующие действия:

- Чтение последних сообщений ядра (`dmesg`)
- Определение USB топологического пути (например `usb 2-1.2`)
- Сопоставление пути с логическим портом USB1-USB4
- Автоматическое переключение радиобаттона
- Проверка наличия блочного устройства `/dev/sdX1`
- Проверка, что файловая система — **FAT32**

Если устройство корректно определено, появляется сообщение:

Filesystem OK (FAT32). Press Run test.

Шаг 3. Кнопка Run test

Выполняется следующий сценарий:

1. Размонтирование `/mnt/usb`, если он был смонтирован
2. Создание каталога монтирования
3. Монтирование устройства **явно как FAT32**
4. Запись тестового файла
5. Чтение тестового файла
6. Сравнение данных
7. Размонтирование устройства

Содержимое тестового файла:

```
test 123456789
```

Шаг 4. Результаты

- **PASS** — порт помечается зелёной точкой
- **FAIL** — порт помечается красной точкой
- **Не тестирован** — серая точка

После завершения теста выводится список **непроверенных портов** и сообщение:

Переставьте устройство в другой порт и повторите тест

4. Проверка USB вручную из консоли Linux

Все действия, которые выполняет GUI тест, можно повторить вручную.

4.1 Проверка USB HOST и HUB

```
lsusb
```

Ожидается: - Root HUB Linux (1d6b:0001, 1d6b:0002) - USB HUB на плате (например 0424:2514)

4.2 Отслеживание подключения USB

```
dmesg -w
```

При подключении устройства должны появиться сообщения:

- new high-speed USB device
 - usb-storage: USB Mass Storage device detected
 - sdX: sdX1
-

4.3 Проверка блочных устройств

```
ls /dev/sd* 2>/dev/null
```

или

```
cat /proc/partitions
```

Ожидается появление:

```
/dev/sda  
/dev/sda1
```

4.4 Определение файловой системы

```
file -s /dev/sda1
```

Для данного теста ожидается:

```
FAT (32 bit)
```

4.5 Монтирование USB (FAT32)

```
mkdir -p /mnt/usb  
mount -t vfat /dev/sda1 /mnt/usb
```

Проверка:

```
mount | grep /mnt/usb  
ls /mnt/usb
```

4.6 Тест записи и чтения

```
echo "test 123456789" > /mnt/usb/usb_test.txt  
sync  
cat /mnt/usb/usb_test.txt
```

Ожидаемый вывод:

```
test 123456789
```

4.7 Безопасное размонтирование

```
mount | grep -q " /mnt/usb " && umount /mnt/usb
```

5. Ограничения и примечания

- В тесте **явно используется FAT32** (`mount -t vfat`)
- Автоопределение файловой системы не используется
- Другие ФС (ext4, NTFS, exFAT) не входят в сценарий теста
- USB3 / USB4 могут быть пропущены в зависимости от аппаратной разводки
- Логическая нумерация USB портов не обязана совпадать с топологией Linux

6. Итог

USB HOST тест предоставляет:

- повторяемый и детерминированный сценарий проверки
- понятное разделение аппаратных и программных проблем
- простой способ диагностики как через GUI, так и через консоль Linux

Этот подход удобен для демонстраций, отладки и базового производственного тестирования.