USB

PAGE CONTENTS

OVERVIEW

The Raspberry Pi Model B is equipped with two USB2.0 ports. These are connected to the LAN9512 combo hub/Ethernet chip IC3, which is itself a USB device connected to the single upstream USB port on BCM2835.

On the Model A, the single USB2.0 port is directly wired to BCM2835.

The USB ports enable the attachment of peripherals such as keyboards, mice, webcams that provide the Pi with additional functionality.

There are some differences between the USB hardware on the Raspberry Pi and the USB hardware on desktop computers or laptop/tablet devices.

The USB host port inside the Pi is an On-The-Go (OTG) host as the application processor powering the Pi, BCM2835, was originally intended to be used in the mobile market: i.e. as the single USB port on a phone for connection to a PC, or to a single device. In essence, the OTG hardware is simpler than the equivalent hardware on a PC.

OTG in general supports communication to all types of USB device, but to provide an adequate level of functionality for most of the USB devices that one might plug into a Pi, the system software has to do more work.

SUPPORTED DEVICES

In general, every device supported by Linux is possible to use with the Pi, subject to a few caveats detailed further down. Linux has probably the most comprehensive driver database for legacy hardware of any operating system (it can lag behind for modern device support as it requires open-source drivers for Linux to recognise the device by default).

If you have a device and wish to use it with a Pi, then plug it in. Chances are that it'll "just work". If you are running in a graphical interface (such as the LXDE desktop environment in Raspbian), then it's likely that an icon or similar will pop up announcing the new device.

If the device doesn't appear to work, then refer to the Troubleshooting section.

GENERAL LIMITATIONS

The OTG hardware on Raspberry Pi has a simpler level of support for certain devices, which may present a higher software processing overhead. The Raspberry Pi also has only one root USB port: all traffic from all connected devices is funnelled down this bus, which operates at a maximum speed of 480mbps.

The USB specification defines three device speeds - Low, Full and High. Most mice and keyboards are Low-speed, most USB sound devices are Full-speed and most video devices (webcams or video capture) are High-speed.

Generally, there are no issues with connecting multiple High-speed USB devices to a Pi.

The software overhead incurred when talking to Low- and Full-speed devices means that there are soft limitations on the number of simultaneously active Low- and Full-speed devices. Small numbers of these types of devices connected to a Pi will cause no issues.

PORT POWER LIMITS

USB devices have defined power requirements, in units of 100mA from 100mA to 500mA. The device advertises its own power requirements to the USB host when it is first connected. In theory, the actual power consumed by the device should not exceed its stated requirement.

The USB ports on a Raspberry Pi have a design loading of 100mA each - sufficient to drive "low-power" devices such as mice and keyboards. Devices such as Wi-Fi adapters, USB hard drives, USB pen drives all consume much more current and should be powered from an external hub with its own power supply. While it is possible to plug a 500mA device into a Pi and have it work with a sufficiently powerful supply, reliable operation is not guaranteed.

In addition, hotplugging high-power devices into the Pi's USB ports may cause a brownout which can cause the Pi to reset.

See Power for more information.

DEVICES WITH KNOWN ISSUES

1. Interoperability between the Raspberry Pi and USB3.0 hubs
There is an issue with USB3.0 hubs in conjuntion with the use of Full- or Low-speed devices (most mice, most keyboards) and the Raspberry Pi. A bug in most USB3.0 hub hardware means that the Raspberry Pi cannot talk to Full- or Low-speed devices connected to a USB3.0 hub.

USB2.0 high-speed devices, including USB2.0 hubs, operate correctly when connected via a USB3.0 hub.

Avoid connecting Low- or Full-speed devices into a USB3.0 hub. As a workaround, plug a USB2.0 hub into the downstream port of the USB3.0 hub and connect the low-speed device, or use a USB2.0 hub between the Pi and the USB3.0 hub, then plug low-speed devices into the USB2.0 hub.

2. USB1.1 webcams
Old webcams may be Full-speed devices. Because these devices transfer a lot of data and incur additional software overhead, reliable operation is not guaranteed.
As a workaround, try to use the camera at a lower resolution.

3. Esoteric USB sound cards
Expensive "audiophile" sound cards typically use far more bandwidth than is necessary to stream audio playback. Reliable operation with 96kHz/192kHz DACs is not guaranteed.
As a workaround, forcing the output stream to be CD quality (44.1kHz/48kHz 16-bit) will reduce the stream bandwidth to reliable levels.

4. Single-TT USB hubs
USB2.0 and 3.0 hubs have a mechanism for talking to Full- or Low-speed devices connected to their downstream ports called a Transaction Translator. This device buffers high-speed requests from the host (i.e. the Pi) and transmits them at Full- or Low-speed to the downstream device. Two configurations of hub are allowed by the USB specification: Single-TT (one TT for all ports) and Multi-TT (one TT per port).
Because of the OTG hardware limitations, if too many Full- or Low-speed devices are plugged into a single-TT hub, unreliable operation of the devices may occur. It is recommended to use a Multi-TT hub to interface with multiple lower-speed devices.
As a workaround, spread lower-speed devices out between the Pi's own USB port and the single-TT hub.

TROUBLESHOOTING

IF YOUR DEVICE DOESN'T WORK AT ALL

The first step is to see if it is detected at all. There are two commands that can be entered into a terminal for this: `lsusb` and `dmesg`. The first will print out all devices attached to USB, whether they are actually recognised by a device driver or not, and the second will print out the kernel message buffer (which can be quite big after booting - try doing `sudo dmesg -C` then plug in your device and retype `dmesg` to see new messages).

As an example with a USB pendrive:

```
pi@raspberrypi ~ $ lsusb
Bus 001 Device 002: ID 0424:9512 Standard Microsystems Corp.
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 001 Device 003: ID 0424:ec00 Standard Microsystems Corp.
Bus 001 Device 005: ID 05dc:a781 Lexar Media, Inc.
pi@raspberrypi ~ $ dmesg
... Stuff that happened before ...
[ 8904.228539] usb 1-1.3: new high-speed USB device number 5
using dwc_otg
[ 8904.332308] usb 1-1.3: New USB device found, idVendor=05dc,
idProduct=a781
[ 8904.332347] usb 1-1.3: New USB device strings: Mfr=1,
Product=2, SerialNumber=3
[ 8904.332368] usb 1-1.3: Product: JD Firefly
[ 8904.332386] usb 1-1.3: Manufacturer: Lexar
[ 8904.332403] usb 1-1.3: SerialNumber: AACU6B4JZVH31337
[ 8904.336583] usb-storage 1-1.3:1.0: USB Mass Storage device
detected
[ 8904.337483] scsi1 : usb-storage 1-1.3:1.0
[ 8908.114261] scsi 1:0:0:0: Direct-Access     Lexar    JD
Firefly       0100 PQ: 0 ANSI: 0 CCS
[ 8908.185048] sd 1:0:0:0: [sda] 4048896 512-byte logical blocks:
(2.07 GB/1.93 GiB)
[ 8908.186152] sd 1:0:0:0: [sda] Write Protect is off
[ 8908.186194] sd 1:0:0:0: [sda] Mode Sense: 43 00 00 00
[ 8908.187274] sd 1:0:0:0: [sda] No Caching mode page present
[ 8908.187312] sd 1:0:0:0: [sda] Assuming drive cache: write
through
[ 8908.205534] sd 1:0:0:0: [sda] No Caching mode page present
[ 8908.205577] sd 1:0:0:0: [sda] Assuming drive cache: write
through
[ 8908.207226]  sda: sda1
[ 8908.213652] sd 1:0:0:0: [sda] No Caching mode page present
[ 8908.213697] sd 1:0:0:0: [sda] Assuming drive cache: write
through
[ 8908.213724] sd 1:0:0:0: [sda] Attached SCSI removable disk
```

In this case, there are no error messages in `dmesg` and the pendrive is detected by the usb-storage driver. If your device did not have a driver available, then typically only the first 6 new lines will appear in the dmesg printout.

If a device enumerates without any errors, but doesn't appear to do anything, then it is likely there are no drivers installed for it. Search around, based on the manufacturer's name for the device or the USB IDs that are displayed in lsusb (e.g. 05dc:a781). The device may not be supported with default Linux drivers - and you may need to download or compile your own third-party software.

IF YOUR DEVICE HAS INTERMITTENT BEHAVIOUR

Poor quality power is the most common cause of devices not working, disconnecting or generally being unreliable.

- If you are using an external powered hub, try swapping the power adapter supplied with the hub for another compatible power supply with the same voltage rating and polarity.
- Check to see if the problem resolves itself if you remove other devices from the hub's downstream ports.
- Temporarily plug the device directly into the Pi and see if the behaviour improves.

VIEW/EDIT THIS PAGE ON GITHUB

READ OUR USAGE AND CONTRIBUTIONS POLICY