Troubleshooting

Jump to bottom

Lingzhu Xiang edited this page on 14 Mar 2017 · 18 revisions

Windows

Try upgrading USB 3 controller driver

Windows 7 does not have great USB 3.0 drivers. In our testing the above known working devices will stop working after running for a while. However, the official Kinect v2 SDK does not support Windows 7 at all, so there is still hope for Windows 7 users. Windows 8.1 and 10 have improved USB 3.0 drivers.

Disabling "USB selective suspend"

Might be helpful to ameliorate transfer problems.

LIBUSB_ERROR_NOTSUPPORTED

The libusb is not correctly built with libusbK backend. Use some prebuilt binary from https://github.com/OpenKinect/libfreenect2/releases/tag/v0.1.0

Also check the build log of `install_libusb_vs201?.cmd` to look for any errors that disable the libusbK backend.

Windows 7: Hangs after a while

Try setting environment variables to use different IR transfer pool sizes: `LIBFREENECT2_IR_PACKETS=64` (packets per transfer, default 8) and `LIBFREENECT2_IR_TRANSFERS=4` (number of transfers in the queue, default 60).

Multiple Kinects

Try to use less but larger transfers with the environment variables:

- `LIBFREENECT2_RGB_TRANSFER_SIZE=1048576` (default 0x4000)
- `LIBFREENECT2_RGB_TRANSFERS=3` (default 20)
- `LIBFREENECT2_IR_PACKETS=128` (or 64, default 8)
- `LIBFREENECT2_IR_TRANSFERS=4` (or 8, default 60)

This is because there is a limit on how many file descriptors that can be polled at once, and this limit is not accounted for in libusb.

Mac OS X

Verify Kinect is indeed connected to a USB 3 port.

Open "System Information" from Spotlight, go to the USB section, and verify "Xbox NUI Sensor" is under "USB 3.0 SuperSpeed Bus" not "High-Speed Bus". If this is not the case, try unplugging the Kinect from power source with the USB cable connected, and plug the power again, then verify.

`Abort trap: 6` on Macs with Nvidia GPUs using OpenGL

Use gdb or lldb to obtain a stack trace. Users report it happens in `gldCreateDevice()` in `GeForceGLDriver`.

This is a known bug in Nvidia OpenGL driver without resolution. You can use the workarounds: use the OpenCL depth processor, or use https://gfx.io/ to force OpenGL to use the Intel GPU.

Thunderbolt / USB 3.0 adaptor

A user reported the Kinect was able to work with the adaptor under OS X 10.10, but not under OS X 10.11. USB 3 hubs and adaptors generally add to hardware instability for USB isochronous transfers.

Linux

Things not to do:

- **Do not run `Protonect` with sudo**. If you do this, we don't accept your bug reports. If you have permission problem, set up the udev rules as instructed.
- Do not use scripts under `depends/` directory to build things unless instructed or you know what you are doing. If you have done so, remove any generated files under `depends/` directory.

Hardware issues

Verify Kinect is indeed connected to a USB 3 port.

```
lsusb -t
```

Example:

```
Bus 03.Port 1: Dev 1, Class=root_hub, Driver=xhci_hcd/4p, 5000M            <-- USB 3
controller driver
|__ Port 1: Dev 2, If 0, Class=Hub, Driver=hub/1p, 5000M                   <-- Kinect
adaptor
    |__ Port 1: Dev 8, If 0, Class=Vendor Specific Class, Driver=, 5000M   <-- Kinect
sensor, first interface
    |__ Port 1: Dev 8, If 1, Class=Vendor Specific Class, Driver=, 5000M
    |__ Port 1: Dev 8, If 2, Class=Audio, Driver=snd-usb-audio, 5000M
    |__ Port 1: Dev 8, If 3, Class=Audio, Driver=snd-usb-audio, 5000M
Bus 02.Port 1: Dev 1, Class=root_hub, Driver=xhci_hcd/9p, 480M
|__ Port 1: Dev 3, If 0, Class=Hub, Driver=hub/1p, 480M
|__ Port 8: Dev 2, If 0, Class=Video, Driver=uvcvideo, 480M
|__ Port 8: Dev 2, If 1, Class=Video, Driver=uvcvideo, 480M
Bus 01.Port 1: Dev 1, Class=root_hub, Driver=ehci-pci/3p, 480M
|__ Port 1: Dev 2, If 0, Class=Hub, Driver=hub/8p, 480M
    |__ Port 5: Dev 12, If 0, Class=Human Interface Device, Driver=usbhid, 12M
```

We have seen people trying to connect Kinect to USB 2 ports.

Check the connectors on the Kinect adaptor

The connector on the Kinect adaptor is easy to come loose.

Try replugging.

Sometimes the Kinect is not detected by the OS if plugged before or during boot.

Check the USB 3 controller support status

```
lspci | grep USB
```

Likely working:

- Intel Corporation 8 Series/C220 Series Chipset Family USB xHCI
- Intel Corporation 7 Series/C210 Series Chipset Family USB xHCI
- NEC Corporation uPD720200 USB 3.0 Host Controller

Probably not working:

- ASMedia Technology Inc. Device 1142
- ASMedia Technology Inc. ASM1042

OS and kernel issues

You must have access to the USB device.

`sudo cp platform/linux/udev/90-kinect2.rules /etc/udev/rules.d/` and replug if not done yet.

Try upgrading the kernel

For Ubuntu 14.04, `sudo apt-get install linux-generic-lts-wily` (4.2) or `linux-generic-lts-vivid` (3.19). (Or `linux-generic-lts-xenial` when it becomes available.)

Caveat: May interfere with Nvidia graphics driver.

Ubuntu kernel >=3.19.0-41 and <=3.19.0-47 had a bug that made some USB controllers (e.g. NEC uPD720200) get `bulk transfer failed: LIBUSB_ERROR_TIMEOUT` errors. Updating to later kernels solves this.

Check `dmesg` after running `Protonect`.

- `Not enough bandwidth for new device state`: Your hardware does not have required bandwidth for the Kinect, even if it supports USB 3.
- `ERROR Transfer event TRB DMA ptr not part of current TD. xHCI host not responding to stop endpoint command. Assuming host is dying, halting host`: Major bugs in the XHCI driver, your USB 3 controller is not well supported by the OS. (May appear as LIBUSB_ERROR_NO_DEVICE)
- `WARN Event TRB for slot x ep y with no TDs queued?`: Harmless warning.
- `xHCI xhci_drop_endpoint called with disabled ep`: Useless warning removed after kernel 4.0.
- `page allocation failure: order:7` in `proc_do_submiturb`: USB buffer allocation fails due to memory fragmentation. Reserve memory `sudo sysctl -w vm.min_free_kbytes=65536` or more (but always less than 5% of total memory) to mitigate this.

Try disabling USB autosuspend

`# for i in /sys/bus/usb/devices/*/power/autosuspend; do echo -1 >$i; done`. Then verify in `grep . /sys/bus/usb/devices/*/power/autosuspend` everything is `-1`.

Sometimes the USB power management can cause some issues. Do not keep doing this if this doesn't improve things.

(Intel OpenCL only) Check known issues at https://www.freedesktop.org/wiki/Software/Beignet/

Some 3.x kernels need `# echo 0 > /sys/module/i915/parameters/enable_cmd_parser`. If it works by default, don't use this.

(Multiple Kinects) Try increasing USBFS buffer size

`# echo 64 > /sys/module/usbcore/parameters/usbfs_memory_mb`, or maybe `128`. Don't set it too large.

Library and driver issues

Enable libusb debug logging

```
LIBUSB_DEBUG=3 ./Protonect
```

This outputs INFO and ERROR level messages from libusb to the console.

(CUDA 8.0) Fails to build third-party apps with `clReleaseMemObject@OPENCL_1.0`

See #804.

libfreenect2 is linked with `libOpenCL.so` from the standard OpenCL ICD loader `ocl-icd-libopencl1` for which it requires versioned dynamic symbols. CUDA 8.0 installs `/etc/ld.so.conf.d/cuda-8-0.conf` which puts `/usr/local/cuda-8.0/targets/x86_64-linux/lib` as the top search path for dynamic library and CUDA 8.0 also provides a worse copy of `libOpenCL.so` without versioned symbols in that path.

Therefore during build time GCC will fail with `undefined reference to clXXX@OpenCL_1.0` and during run time the app will report `no version information available (required by the app)` but still manages to continue.

Solutions:

- The solution to the root cause is `sudo mv /etc/ld.so.conf.d/cuda-8-0.conf /etc/ld.so.conf.d/zz_cuda-8-0.conf` but this will be reverted every time CUDA packages are updated. Nvidia must take action if this change is to persist.
- If you have a prebuilt libfreenect2 (or any library linked with ocl-icd) and can't rebuild it, build your own program with `LD_LIBRARY_PATH=/usr/lib/x86_64-linux-gnu make` to override CUDA's library path.
- (This workaround has been coded in) If you can rebuild libfreenect2 (or any library linked with ocl-icd), build it with `cmake .. -DCMAKE_INSTALL_RPATH=/usr/lib/x86_64-linux-gnu` to override CUDA's library path. Note this is an ugly hack which you should stop using after Nvidia properly resolves this problem.

(OpenGL only) Check driver version

```
glxinfo | grep OpenGL
```

Either the "OpenGL core profile version string" or the "OpenGL version string" must be greater than 3.1. If not, you don't have required graphics driver or you have not installed one correctly. Also, you should not see `llvmpipe` which is a software OpenGL renderer.

Insufficient version may appear as `GLFW error 65543 The requested client API version is unavailable.`

(OpenCL only) Smoke test OpenCL driver

`clinfo` should show the OpenCL setup information.

Jetson TK1 issues

External USB 3 extension cards or USB hub may interfere with the Kinect. The built-in blue USB 3 port is tested to be working.

USB 3.0 must be enabled and USB autosuspend must be disabled. Edit `/boot/extlinux/extlinux.conf` to change `usb_port_owner_info=0` to `usb_port_owner_info=2` and add `usbcore.autosuspend=-1` .

Udev rules must be set up. See above.

You may need to maximize CPU and GPU clock to maintain basic Kinect functionalities, see http://elinux.org/Jetson/Performance for instructions.

If the Kinect is plugged in during boot, it may not work. Reconnecting or plugging in Kinect2 after the system boots.

Jetson TX1 issues

Update: Nvidia has posted a preliminary firmware patch which resolves the following problem. The patch is here: https://devtalk.nvidia.com/default/topic/919354/jetson-tx1/usb-3-transfer-failures/post/4899105/#4899105

Jetson TX1 is known to not have working depth transfers. The symptoms are `low-level USB error -18` with `LIBUSB_DEBUG=4` , and a lot of `WARN Event TRB for slot 3 ep 8 with no TDs queued?` in dmesg.

The only working configuration reported so far is to have a uPD720202 extension card, and patch libusb to change `MAX_ISO_BUFFER_LENGTH` from `49152 * 128` to `49152` .

▾ Pages 5

Find a Page...

Home

Linux USB Notes

Performance

Test Plan

Troubleshooting

Clone this wiki locally

```
https://github.com/OpenKinect/libfreenect2.wiki.git
```