Linux Gadget Serial Driver v2.0

11/20/2004

(updated 8-May-2008 for v2.3)

License and Disclaimer

If you have questions, problems, or suggestions for this driver please contact Al Borchers at alborchers@steinerpoint.com.

Prerequisites

Versions of the gadget serial driver are available for the 2.4 Linux kernels, but this document assumes you are using version 2.3 or later of the gadget serial driver in a 2.6 Linux kernel.

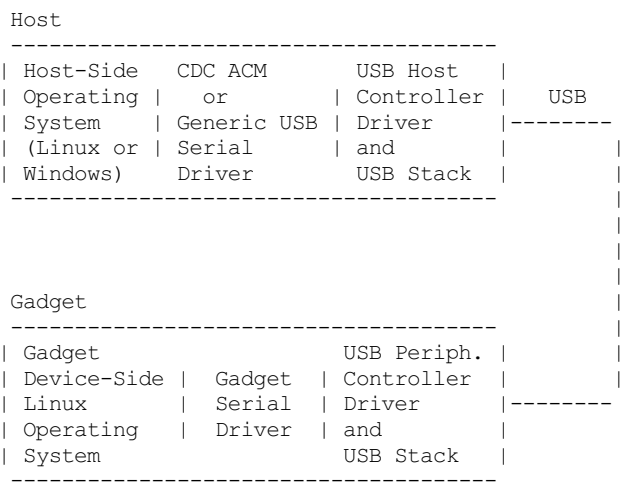
This document assumes that you are familiar with Linux and Windows and know how to configure and build Linux kernels, run standard utilities, use minicom and HyperTerminal, and work with USB and serial devices. It also assumes you configure the Linux gadget and usb drivers as modules.

With version 2.3 of the driver, major and minor device nodes are no longer statically defined. Your Linux based system should mount sysfs in /sys, and use "mdev" (in Busybox) or "udev" to make the /dev nodes matching the sysfs /sys/class/tty files.

Overview

The gadget serial driver is a Linux USB gadget driver, a USB device side driver. It runs on a Linux system that has USB device side hardware; for example, a PDA, an embedded Linux system, or a PC with a USB development card.

The gadget serial driver talks over USB to either a CDC ACM driver or a generic USB serial driver running on a host PC:



On the device-side Linux system, the gadget serial driver looks like a serial device.

On the host-side system, the gadget serial device looks like a CDC ACM compliant class device or a simple vendor specific device with bulk in and bulk out endpoints, and it is treated similarly to other serial devices.

The host side driver can potentially be any ACM compliant driver or any driver that can talk to a device with a simple bulk in/out interface. Gadget serial has been tested with the Linux ACM driver, the Windows usbser.sys ACM driver, and the Linux USB generic serial driver.

With the gadget serial driver and the host side ACM or generic serial driver running, you should be able to communicate between the host and the gadget side systems as if they were connected by a serial cable.

The gadget serial driver only provides simple unreliable data communication. It does not yet handle flow control or many other

features of normal serial devices.

Installing the Gadget Serial Driver

To use the gadget serial driver you must configure the Linux gadget side kernel for "Support for USB Gadgets", for a "USB Peripheral Controller" (for example, net2280), and for the "Serial Gadget" driver. All this are listed under "USB Gadget Support" when configuring the kernel. Then rebuild and install the kernel or modules.

Then you must load the gadget serial driver. To load it as an ACM device (recommended for interoperability), do this:

```
modprobe g_serial
```

To load it as a vendor specific bulk in/out device, do this:

```
modprobe g_serial use_acm=0
```

This will also automatically load the underlying gadget peripheral controller driver. This must be done each time you reboot the gadget side Linux system. You can add this to the start up scripts, if desired.

Your system should use mdev (from busybox) or udev to make the device nodes. After this gadget driver has been set up you should then see a /dev/ttyGS0 node:

```
# ls -l /dev/ttyGS0 | cat
crw-rw----    1 root     root     253,   0 May  8 14:10 /dev/ttyGS0
#
```

Note that the major number (253, above) is system-specific. If you need to create /dev nodes by hand, the right numbers to use will be in the /sys/class/tty/ttyGS0/dev file.

When you link this gadget driver early, perhaps even statically, you may want to set up an /etc/inittab entry to run "getty" on it. The /dev/ttyGS0 line should work like most any other serial port.

If gadget serial is loaded as an ACM device you will want to use either the Windows or Linux ACM driver on the host side. If gadget serial is loaded as a bulk in/out device, you will want to use the Linux generic serial driver on the host side. Follow the appropriate instructions below to install the host side driver.

Installing the Windows Host ACM Driver

To use the Windows ACM driver you must have the "linux-cdc-acm.inf" file (provided along this document) which supports all recent versions of Windows.

When the gadget serial driver is loaded and the USB device connected to the Windows host with a USB cable, Windows should recognize the gadget serial device and ask for a driver. Tell Windows to find the driver in the folder that contains the "linux-cdc-acm.inf" file.

For example, on Windows XP, when the gadget serial device is first plugged in, the "Found New Hardware Wizard" starts up. Select "Install from a list or specific location (Advanced)", then on the next screen select "Include this location in the search" and enter the path or browse to the folder containing the "linux-cdc-acm.inf" file. Windows will complain that the Gadget Serial driver has not passed Windows Logo testing, but select "Continue anyway" and finish the driver installation.

On Windows XP, in the "Device Manager" (under "Control Panel", "System", "Hardware") expand the "Ports (COM & LPT)" entry and you should see "Gadget Serial" listed as the driver for one of the COM ports.

To uninstall the Windows XP driver for "Gadget Serial", right click on the "Gadget Serial" entry in the "Device Manager" and select "Uninstall".

Installing the Linux Host ACM Driver

To use the Linux ACM driver you must configure the Linux host side kernel for "Support for Host-side USB" and for "USB Modem (CDC ACM) support".

Once the gadget serial driver is loaded and the USB device connected to the Linux host with a USB cable, the host system should recognize the gadget serial device. For example, the command:

```
cat /sys/kernel/debug/usb/devices
```

should show something like this::

```
T:  Bus=01 Lev=01 Prnt=01 Port=01 Cnt=02 Dev#=  5 Spd=480 MxCh= 0
D:  Ver= 2.00 Cls=02(comm.) Sub=00 Prot=00 MxPS=64 #Cfgs=  1
P:  Vendor=0525 ProdID=a4a7 Rev= 2.01
S:  Manufacturer=Linux 2.6.8.1 with net2280
S:  Product=Gadget Serial
S:  SerialNumber=0
C:* #Ifs= 2 Cfg#= 2 Atr=c0 MxPwr=  2mA
I:  If#= 0 Alt= 0 #EPs= 1 Cls=02(comm.) Sub=02 Prot=01 Driver=acm
```

```
E:  Ad=83(I) Atr=03(Int.) MxPS=   8 Ivl=32ms
I:  If#= 1 Alt= 0 #EPs= 2 Cls=0a(data ) Sub=00 Prot=00 Driver=acm
E:  Ad=81(I) Atr=02(Bulk) MxPS= 512 Ivl=0ms
E:  Ad=02(O) Atr=02(Bulk) MxPS= 512 Ivl=0ms
```

If the host side Linux system is configured properly, the ACM driver should be loaded automatically. The command "lsmod" should show the "acm" module is loaded.

Installing the Linux Host Generic USB Serial Driver

To use the Linux generic USB serial driver you must configure the Linux host side kernel for "Support for Host-side USB", for "USB Serial Converter support", and for the "USB Generic Serial Driver".

Once the gadget serial driver is loaded and the USB device connected to the Linux host with a USB cable, the host system should recognize the gadget serial device. For example, the command:

```
cat /sys/kernel/debug/usb/devices
```

should show something like this::

```
T:  Bus=01 Lev=01 Prnt=01 Port=01 Cnt=02 Dev#=  6 Spd=480 MxCh= 0
D:  Ver= 2.00 Cls=ff(vend.) Sub=00 Prot=00 MxPS=64 #Cfgs=  1
P:  Vendor=0525 ProdID=a4a6 Rev= 2.01
S:  Manufacturer=Linux 2.6.8.1 with net2280
S:  Product=Gadget Serial
S:  SerialNumber=0
C:* #Ifs= 1 Cfg#= 1 Atr=c0 MxPwr=  2mA
I:  If#= 0 Alt= 0 #EPs= 2 Cls=0a(data ) Sub=00 Prot=00 Driver=serial
E:  Ad=81(I) Atr=02(Bulk) MxPS= 512 Ivl=0ms
E:  Ad=02(O) Atr=02(Bulk) MxPS= 512 Ivl=0ms
```

You must load the usbserial driver and explicitly set its parameters to configure it to recognize the gadget serial device, like this:

```
echo 0x0525 0xA4A6 >/sys/bus/usb-serial/drivers/generic/new_id
```

The legacy way is to use module parameters:

```
modprobe usbserial vendor=0x0525 product=0xA4A6
```

If everything is working, usbserial will print a message in the system log saying something like "Gadget Serial converter now attached to ttyUSB0".

Testing with Minicom or HyperTerminal

Once the gadget serial driver and the host driver are both installed, and a USB cable connects the gadget device to the host, you should be able to communicate over USB between the gadget and host systems. You can use minicom or HyperTerminal to try this out.

On the gadget side run "minicom -s" to configure a new minicom session. Under "Serial port setup" set "/dev/ttygserial" as the "Serial Device". Set baud rate, data bits, parity, and stop bits, to 9600, 8, none, and 1--these settings mostly do not matter. Under "Modem and dialing" erase all the modem and dialing strings.

On a Linux host running the ACM driver, configure minicom similarly but use "/dev/ttyACM0" as the "Serial Device". (If you have other ACM devices connected, change the device name appropriately.)

On a Linux host running the USB generic serial driver, configure minicom similarly, but use "/dev/ttyUSB0" as the "Serial Device". (If you have other USB serial devices connected, change the device name appropriately.)

On a Windows host configure a new HyperTerminal session to use the COM port assigned to Gadget Serial. The "Port Settings" will be set automatically when HyperTerminal connects to the gadget serial device, so you can leave them set to the default values--these settings mostly do not matter.

With minicom configured and running on the gadget side and with minicom or HyperTerminal configured and running on the host side, you should be able to send data back and forth between the gadget side and host side systems. Anything you type on the terminal window on the gadget side should appear in the terminal window on the host side and vice versa.