

USB Gadget Connections

Systems that have a USB device port can be connected to a PC through USB. A USB cable will need to be connected between the USB device port on the board and a USB host port on a PC. This section covers the process of establishing a connection using the gadget Ethernet driver and the gadget serial driver. More information regarding the USB Gadget drivers for Linux can be found here: <http://www.linux-usb.org/gadget> (<http://www.linux-usb.org/gadget>) .

USB Gadget Ethernet

On the board, an entry for the USB gadget device should be entered into the network interfaces file `/etc/network/interfaces` on the board. The entry below configures the interface for an IP address of 192.168.0.2, subnet mask of 255.255.255.0, and default gateway of 192.168.0.1.

```
allow-hotplug usb0
iface usb0 inet static
    address 192.168.0.2
    netmask 255.255.255.0
    gateway 192.168.0.1
```

Also, the USB Ethernet Gadget kernel module will need to be loaded. This can be done manually using the command `modprobe g_ether`. For automatic loading, add a line containing `g_ether` to the `/etc/modules` file. Additional options can be passed to the `g_ether` module if desired. See the output of `modinfo g_ether` for more information. To enable hotplugging and automatic configuration of the interface, `usb0` should be added to `/etc/ifplugd/ifplugd.conf`.

On the Linux PC, a configuration will need to be made for the USB device interface as well. The USB device should be detected after it has been connected and configured as a USB Ethernet Gadget. The appropriate module should be automatically loaded. If it is not, manually load the `usbnet` module using `modprobe`. The exact method of configuration depends on the Linux distribution. If the distribution uses the `/etc/network/interfaces` file, an entry like the one below should be added. The interface should be configured to match the addresses used for the board. The address for the Linux PC interface should be the same as the gateway setting for the board.

```
allow-hotplug usb0
iface usb0 inet static
    address 192.168.0.2
    netmask 255.255.255.0
    gateway 192.168.0.1
```

USB Gadget Serial

Serial device connections between the target board and host PC can be accomplished using the Linux gadget serial driver: `g_serial`. This allows for data transfer between the two systems using the standard serial programming API and existing terminal applications. To use the gadget serial driver, it must first be loaded using `modprobe`. Several parameters can be passed to the module on load, as reported by the `modinfo` command:

```

root@som9g20:~# modinfo g_serial
filename:      /lib/modules/2.6.28/kernel/drivers/usb/gadget/g_serial.ko
license:       GPL
author:        David Brownell
author:        Al Borchers
description:    Gadget Serial v2.4
license:       GPL
author:        Felipe Balbi
depends:        at91_udec
vermagic:      2.6.28 preempt mod_unload ARMv5
parm:          idVendor:USB Vendor ID (ushort)
parm:          idProduct:USB Product ID (ushort)
parm:          bcdDevice:USB Device version (BCD) (ushort)
parm:          iManufacturer:USB Manufacturer string (charp)
parm:          iProduct:USB Product string (charp)
parm:          iSerialNumber:SerialNumber string (charp)
parm:          use_acm:Use CDC ACM, default=yes (bool)
parm:          use_obex:Use CDC OBEX, default=no (bool)
parm:          n_ports:number of ports to create, default=1 (uint)

```

In most cases, the default settings will be acceptable, and the driver can be loaded by running the command `modprobe g_serial` with no arguments. After loading the module, connect a USB cable between the target board and a Linux PC. Once connected, the `dmesg` command should indicate that the connection was detected and associated with `g_serial`.

On the Linux PC, the `lsusb` command or `cat /proc/bus/usb/devices` should indicate that the device is connected and what vendor and product ID are used by the device. Once this has been determined, the `usbserial` driver can be registered and attached to the device. In this example, the vendor ID for the device is 0x0525 and the product ID is 0xA4A6.

```

root@emac-ldc:~# modprobe usbserial vendor=0x0525 product=0xA4A6
root@emac-ldc:~# dmesg | tail
[1462941.986961] usb 4-2: New USB device strings: Mfr=1, Product=2, SerialNumber=3
[1462941.986961] usb 4-2: Product: Gadget Serial
[1462941.986961] usb 4-2: Manufacturer: Linux 2.6.25 with at91_udec
[1462941.986961] usb 4-2: SerialNumber: 0
[1463015.408126] usbcore: registered new interface driver usbserial
[1463015.408147] drivers/usb/serial/usb-serial.c: USB Serial support registered for generic
[1463015.408174] usbserial_generic 4-2:1.0: generic converter detected
[1463015.408229] usb 4-2: generic converter now attached to ttyUSB0
[1463015.408241] usbcore: registered new interface driver usbserial_generic
[1463015.408243] drivers/usb/serial/usb-serial.c: USB Serial Driver core

```

To test the USB serial interface, use a standard terminal application such as `minicom`. The `microcom` application on the EMAC OE system can be used for the target board interface. The device on the Linux PC will be named `/dev/ttyUSB0` (assuming that this is the only USB serial device connected); the device on the target system will be `/dev/ttyGS0`. As long as the settings agree between the two systems, data will be transferred across the interface.

```

root@som9g20:~# microcom -s 115200 /dev/ttyGS0
....
developer@emac-ldc:~$ minicom -s
....

```

The baud rate and hardware flow control settings are not applicable to the USB serial interface and will be ignored.

It is also possible to connect to Microsoft Windows computers through this interface using a driver provided in the Linux kernel documentation and the ACM mode supported by the driver. The documentation can be found in the kernel source for your target device under `Documentation/usb/gadget_serial.txt`. For quick reference, you may

view the file for the 2.6.28 kernel for the EMAC AT91-based boards here: https://svn.emacinc.com/public/linux-2.6.28-at91/trunk/Documentation/usb/gadget_serial.txt (https://svn.emacinc.com/public/linux-2.6.28-at91/trunk/Documentation/usb/gadget_serial.txt) .

» qt » install » getting_started » eclipse » uboot_image_loading » emac_oe_fact » emac_oe_getting_started » boot_process » linux_start » emac_oe_gadget

-
- linux/emac_oe_gadget.txt · Last modified: 2011/02/08 23:03 by tstratman
 - Except where otherwise noted, content on this wiki is licensed under the following license: CC Attribution-No Derivative Works 3.0 Unported (cc-by-nd)