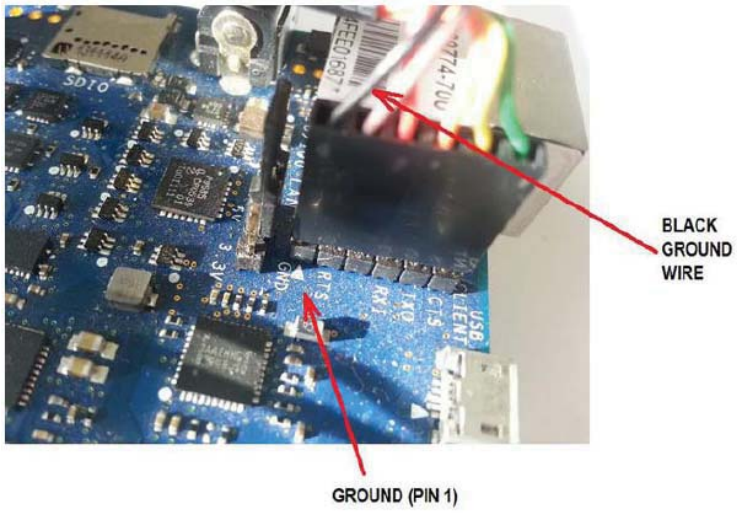
Class Web: WWW.MWFTR.COM/emblab.html

EECE456 Embedded Systems Design Lab
Electrical and Computer Engineering
Howard University
Instructor: Dr. Charles Kim

Accessing Linux Console of Galileo[1]

Galileo Gen 2:

Accessing Linux Console means we can run regular Linux commands directly on the Galileo board. A special serial cable is required to connect a computer to Galileo Gen 2. For Galileo Gen 1 Board, read the next page for the proper type of connector. This special serial cable is a FTDI cable with a 6-pin connector and +3.3 V serial to USB as illustrated below. Thankfully, this cable is included in the Grove Kit box. When we connect the special serial cable, we make sure the inline connector connects pin 1 (black) to pin 1 of the FTDI connector (which is indicated by a little white triangle) on Galileo Gen 2. We connect the other end of the cable to a computer.



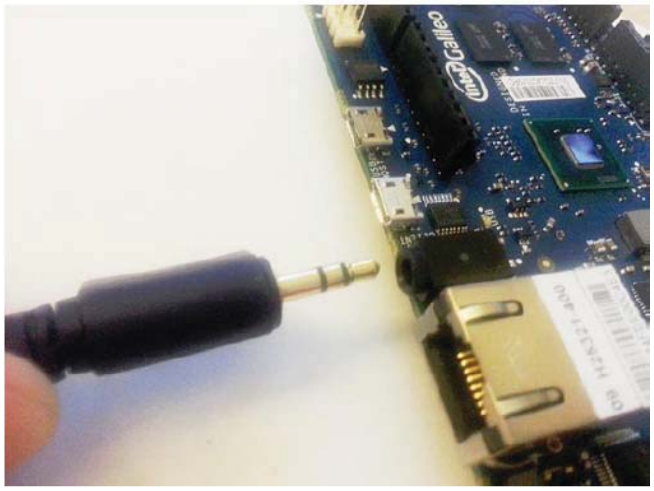
The next step tests the cable with Galileo Gen 2 board. By the way, to run a Linux console with the cable, we need to install the software on a computer in order to pen a serial channel with Galileo Gen 2. The software we need depends on our operational system and our preference. This step explores **putty** for Windows, **minimcom** for Linux, and **screen** for Max OS. Note that **screen** is also available on Linux, and **minicom** is also available in Mac OS.

Now jump to P3.

[1] This instruction is based on "Intel Galileo Gen 1 and Gen 2 - API Features and Arduino Projects for Linux Programming" by Ramoel Ramon, Apress Open.

Galileo Gen 1:

For Galileo (Gen 1), a very special type of connector is required: 3.5mm stereo jack to USB FTDI cable converter cable as shown in the left figure below. And the 3.5 mm jack is to be inserted to the Gen 1 board as indicated in the right figure below, and the other end (USB) to a computer.



The next step tests the cable with Galileo Gen 1 board. By the way, to run a Linux console with the cable, we need to install the software on a computer in order to pen a serial channel with Galileo Gen 1. The software we need depends on our operational system and our preference. This step explores **putty** for Windows, **minimcom** for Linux, and **screen** for Max OS. Note that **screen** is also available on Linux, and **minicom** is also available in Mac OS.

Go to the next page.

Windows

The following steps explain how to set up the Linux console on Windows:

1. After you insert the cable, wait for a few seconds so that the USB is enumerated. Then open the Windows Device Manager by choosing Start ➤ Control Panel ➤ Hardware and Sound ➤ Device Manager. You can also press the Windows key  and "R" at same time, and then type `devmgmt.msc`.

2. In the Device Manager, check the COM port available under the Ports section. Figure 1-20 shows an example of a COM port enumerated as COM5.

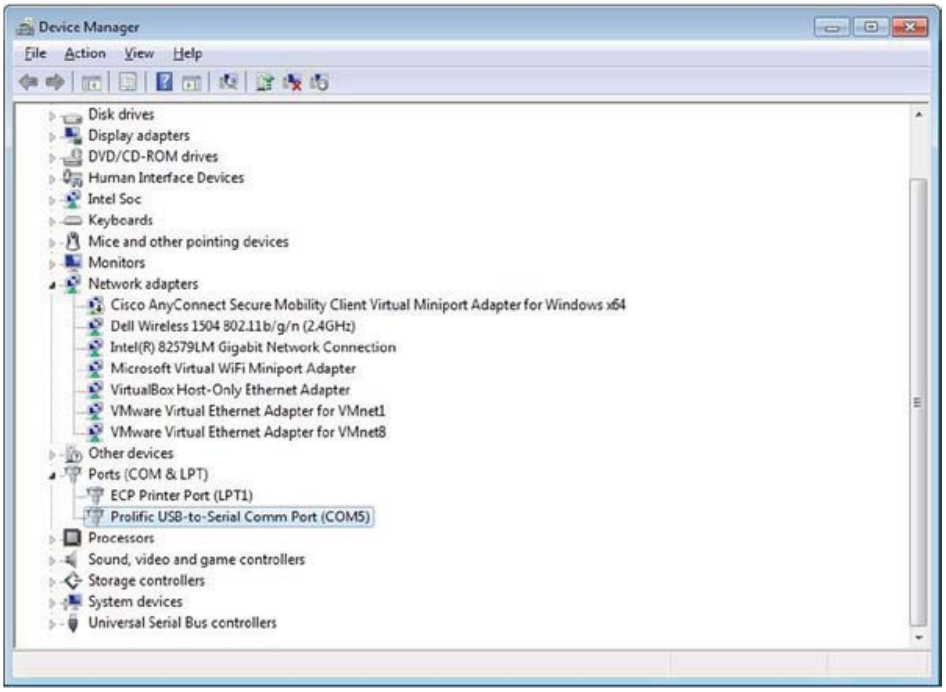


Figure 1-20. *Intel Galileo COM port on Windows*

3. Download putty and install it on your Windows machine. You can download it from `http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html`.

4. Execute putty. Select the Serial protocol and enter the COM port number. Then click the Open button, as shown in Figure 1-21.

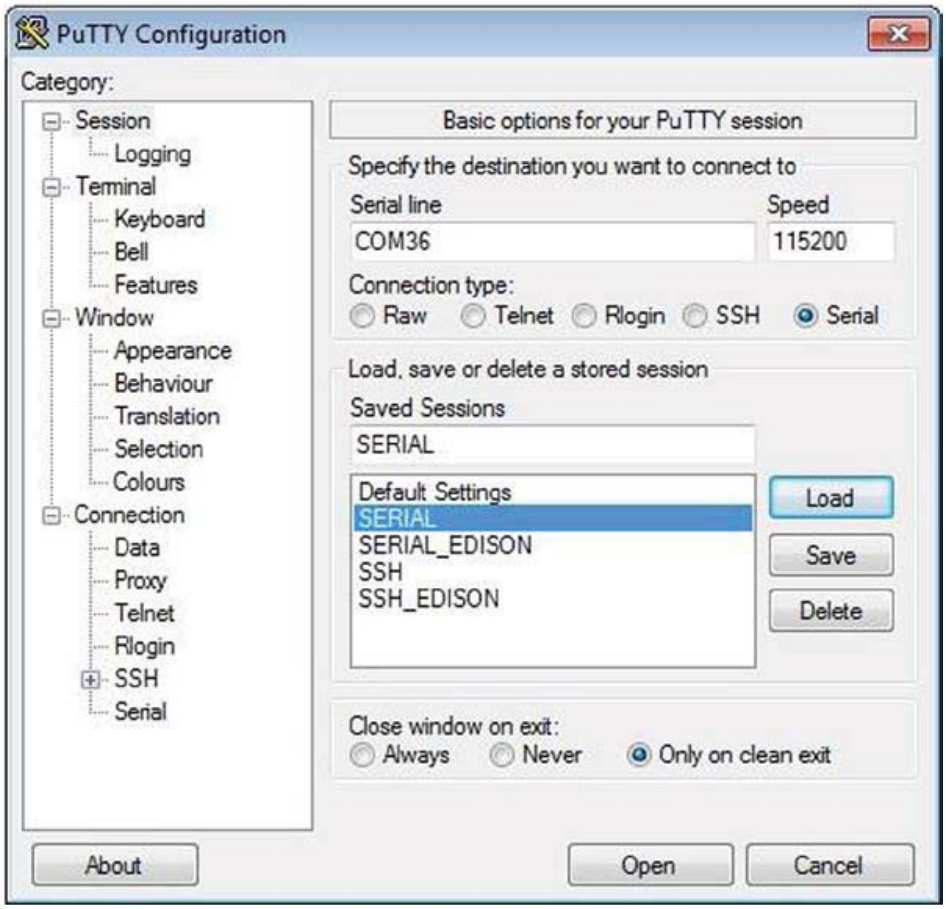


Figure 1-21. *Configuring putty to open the Linux console*

5. Finally, to access the Linux console, you type username `root` and press Enter. You will then have access to the Linux console, as shown in Figure 1-22.

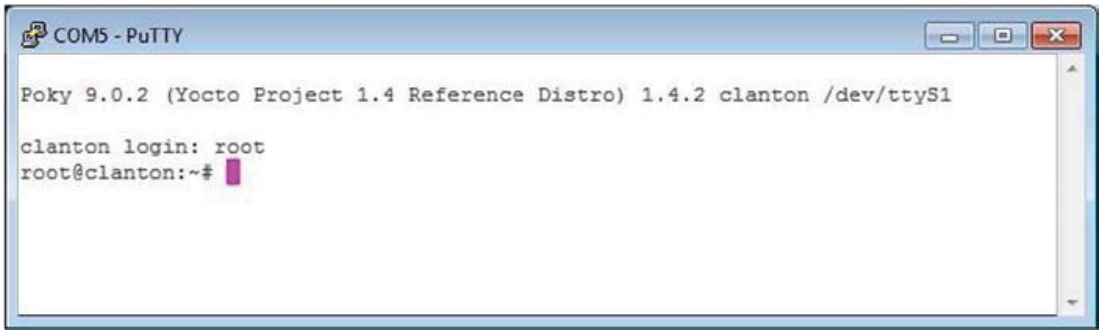


Figure 1-22. *The Linux serial console*

Ubuntu Linux

The procedure to set up Intel Galileo's Linux console on a Linux computer is easy. As mentioned, serial communication on Linux computers is based on the `minicom` software.

The following steps are necessary to get the Linux console working:

1. After you insert the cable, wait for a few seconds for the USB to be enumerated.
2. Open a Linux terminal. You can press Ctrl+T to do this.
3. Check the port by typing the command `dmesg|grep tty`. For example, Figure 1-23 shows the port enumerated as `ttyUSB0`.

Figure 1-23. *Checking the USB port enumerated on the Ubuntu terminal*

4. Finally, to access the Linux console, you typing the `sudo minicom --device /dev/ttyUSB0` command. The terminal will open. You need to use the username `root`.

Mac OSX

The following steps describe how to set up the Linux console on Mac OSX.

1. After you insert the cable, wait for a few seconds so that the USB is enumerated.
2. Open an OSX terminal. You can press ⌘ and the spacebar at same time to open the Spotlight text box. Then type `terminal` and press Enter.
3. In the terminal, check what is the serial port enumerated as following command:

```
~$ ls /dev/tty.usb*
/dev/tty.usbserial-A603HVUT
```

4. Finally, you can access the Linux console by typing `screen /dev/<YOUR SERIAL HERE> 115200` into the terminal. For example

   ~$ screen /dev/tty.usbserial-A603HVUT 115200

5. The console will open. You need to use the username `root`.

Exploring the Linux Console

Try to run some Linux commands. To check which board you have, you can run the following command:

```
root@clanton:~# cd /sys/firmware/board_data/
root@clanton:/sys/firmware/board_data# cat flash_version
0x01000300
```

As you can see, when this chapter was written, my board was using firmware 1.0.3. The hexadecimal sequence is decoded as 01.00.03.00, 01.00.03, or 1.0.3, as expected.

A second interesting test is to check your platform name. Run the following commands if your board is Intel Galileo:

```
root@clanton:~# cd /sys/devices/platform/Galileo
root@clanton:/sys/devices/platform/Galileo# cat modalias
platform:Galileo
```

If your board is Intel Galileo Gen 2, the following commands work:

```
root@clanton:~# cd /sys/devices/platform/GalileoGen2/
root@clanton:/sys/devices/platform/GalileoGen2# cat modalias
platform:GalileoGen2
```

The releases provided by Intel contain the `busybox` software utility. It provides about 300 commands that can be executed in the Linux console. If you type `busybox` and press Enter, you will be able to see the supported commands.

```
root@clanton:/sys/devices/platform/GalileoGen2# busybox
BusyBox v1.20.2 (2014-08-22 10:41:19 PDT) multi-call binary.
Copyright (C) 1998-2011 Erik Andersen, Rob Landley, Denys Vlasenko
and others. Licensed under GPLv2.
See source distribution for full notice.

Usage: busybox [function] [arguments]...
   or: busybox --list
   or: function [arguments]...
```

```
        BusyBox is a multi-call binary that combines many common Unix
        utilities into a single executable.  Most people will create a
        link to busybox for each function they wish to use and BusyBox
        will act like whatever it was invoked as.

Currently defined functions:
        [, [[, acpid, ar, arp, arping, ash, awk, basename, blkid, blockdev,
        bootchartd, brctl, bunzip2, bzcat, cat, chgrp, chmod, chown, chroot,
        chrt, clear, cmp, cp, cpio, cttyhack, cut, date, dc, dd, deallocvt,
        depmod, df, diff, dirname, dmesg, dnsdomainname, du, dumpkmap, echo,
        egrep, env, expr, false, fdisk, fgrep, find, findfs, flock, free, fsck,
        fsync, ftpd, ftpget, ftpput, fuser, getty, grep, gunzip, gzip, halt,
        hd, head, hexdump, hostname, hwclock, id, ifconfig, ifdown, ifup,
        insmod, ionice, iostat, ip, kill, killall, klogd, less, ln, loadkmap,
        logger, login, logname, logread, losetup, ls, lsmod, lsof, lspci,
        lsusb, md5sum, mdev, mkdir, mkfifo, mknod, mktemp, modprobe, more,
        mount, mv, nc, netstat, nice, nohup, nslookup, od, patch, pidof, ping,
        ping6, pivot_root, pmap, poweroff, printf, ps, pwd, rdate, readlink,
        realpath, reboot, renice, reset, resize, rm, rmdir, rmmod, route,
        run-parts, sed, seq, setconsole, setserial, setsid, sh, sleep, sort,
        start-stop-daemon, stat, strings, stty, sulogin, switch_root, sync,
        sysctl, syslogd, tail, tar, tcpsvd, tee, telnet, telnetd, test, tftp,
        time, timeout, top, touch, tr, traceroute, traceroute6, true, tty,
        udhcpc, umount, uname, uniq, unzip, uptime, usleep, vconfig, vi, watch,
        wc, wget, which, who, whoami, xargs, yes, zcat, zcip
```

If you need to execute one of these commands, you simply type the desired command.

Testing the Data Cables

The high-speed USB 2.0 male-to-micro USB cable, or simply the data cable, is used to transfer and debug the sketches. Testing this cable is covered in Chapter 3 because doing so requires the installation of the IDE.

File Edit View History Bookmarks Tools Help

Amazon.com: Ftdi USB to ...

www.amazon.com/3-5mm-Stereo-Cable-Module-Converter/dp/B00M3M1U2A

Search

amazon Try Prime

Your Amazon.com | Today's Deals | Gift Cards | Sell | Help

Shop by Department | Search | Electronics | Hello. Sign in Your Account

Computers | Best Sellers | Laptops & Tablets | Desktops & Monitors | Computer Accessories | PC Components | PC Gaming | All Electronics

Electronics › Computers & Accessories › Cables & Accessories › Cables & Interconnects › USB Cables

Click to open expanded view

Ftdi USB to TTL 3.5mm Stereo Jack Cable Module Converter 5v

by Connecting Cable

Be the first to review this item

Sale: $18.00 + $4.59 shipping

Only 20 left in stock.

Ships from and sold by JUDY STORE.

- 6ft (1.8m) FTDI TTL-232R-5V-AJ USB Programming Cable with 3.5mm stereo jack
- Support FT232R chip programming, Verdor ID, serial NO re-write
- Standard: Tip: TX, Ring-RX, Sleeve-GND
- USB TTL to 3.5mm stereo jack cable
- 300 ~ 1Mbps baud rate speed

2 new from $14.99

Great Deals on Computer Accessories

Shop now

What Other Items Do Customers Buy After Viewing This Item?

SF Cable, DB9 Female to 3.5mm Serial Cable-6 Feet