



Lesson #2: The Target Board & Class Software

Making it Real





Installation Steps

- Install ARM9 tool chain and sample code from class CD
- Configure minicom
- Configure Host Networking
- Connect the Target
- Run sample program



Installing Class Files

- Insert CD ROM and mount (should automount)
- cd to your home directory
- Become root user – su
 - Password:
- Execute
/run/media/<your_user>/EmbeddedLinux/install_tools.sh or
- <your_mount_point>/install_tools.sh <your_mount_point>
- Exit root user shell
- Execute /run/media/<your_user>/EmbeddedLinux/install.sh
or
- <your_mount_point>/install.sh <your_mount_point>



What Got Installed

- /opt/arm
 - Complete ARM9 cross tool chain
- /usr/local
 - eclipse – Eclipse IDE
- /home
 - target_fs – link to \$HOME/target_fs



Your Home Directory

- `busybox-1.29.2` – Busybox source tree
- `target_fs` – root file system for target board
 - `home/src` – sample source code
- `linux-rpi-4.19.y` – kernel source tree
- `boot` – boot partition from the micro SD card
- `u-boot` – source tree for u-boot boot loader



Configure minicom

- As root user, run `minicom -s`
- Select *Serial port setup*
 - Serial Device (`/dev/ttyUSB0`)
 - Bps/Par/Bits (115200 8N1)
 - No flow control
- Select *Modem and dialing*
 - Remove Init string
 - Remove Reset string



Configure minicom II

- Select *Screen and keyboard*
 - Type “b” to change backspace behavior to DEL
 - Type “r” to turn on line wrap
- Select *Save setup as dfl*
- Exit minicom



Add your user to dialout group

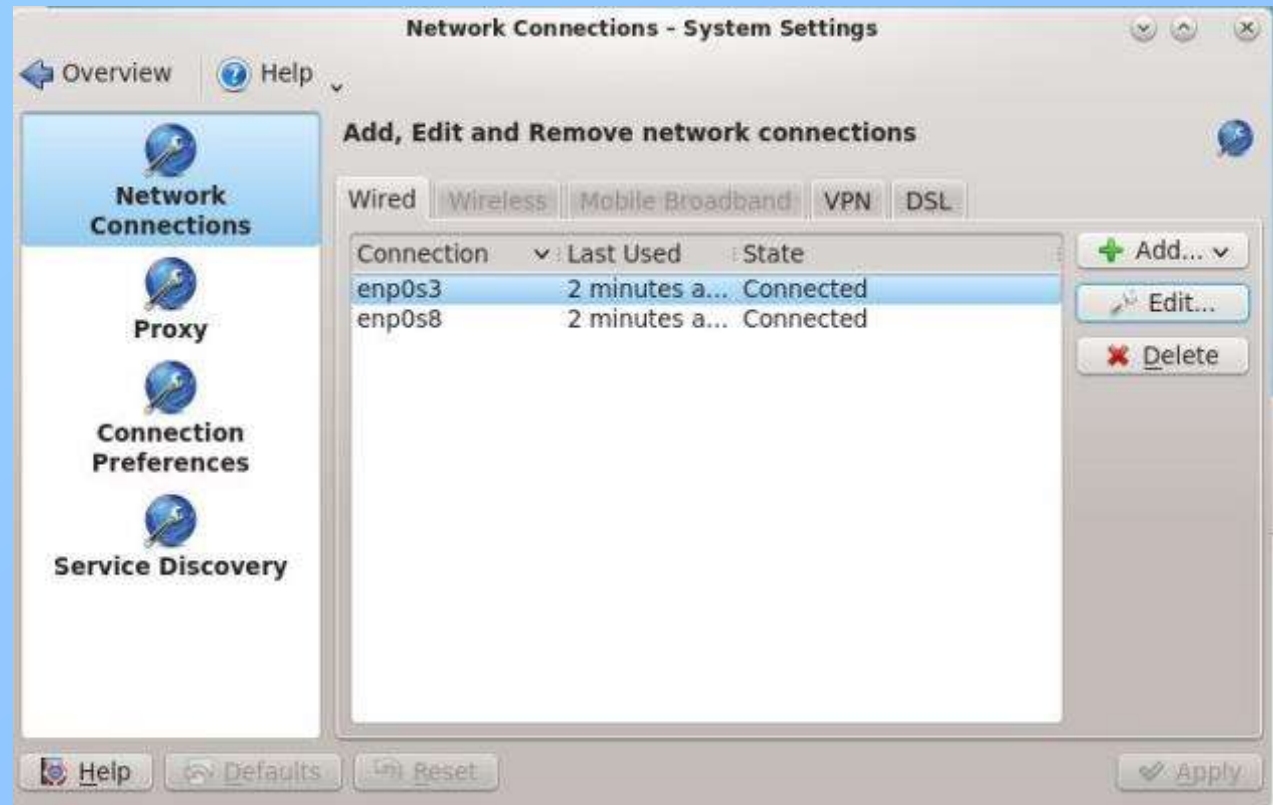
- As root user edit /etc/group
 - Add your user name to the line that begins “dialout”
 - dialout:x:18:<your_user_name>
 - Save file

root kwrite /etc/group



Configure Host Networking

Settings > System Settings > Network and Connectivity > Network Settings





Configure Host Networking 2

Edit Network Connection - System Settings

Connection name:

☒ Connect automatically

☒ System connection [Advanced Permissions](#)

IPv4 Address | IPv6 Address | Ethernet | 802.1x Security

Basic settings ▾

Method:

IP Address:

Subnet Mask:

Gateway:

DNS Servers: ...

Search Domains: ...

DHCP Client ID:

☒ IPv4 is required for this connection

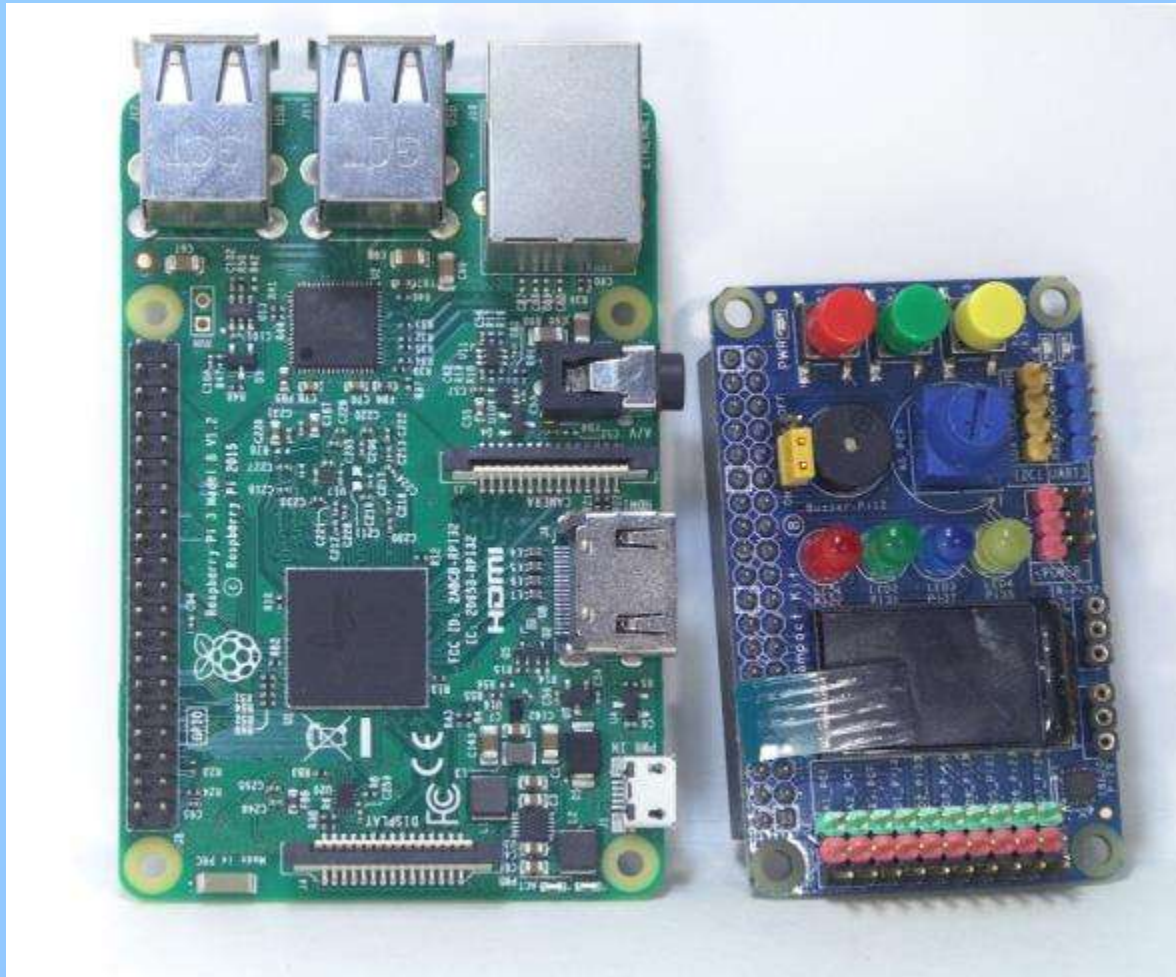


Configure NFS

- Edit /etc/exports (must be root)
 - Change <your_user_name> to your user name
 - Save
- In shell window as root:
 - systemctl enable nfs-server.service
 - systemctl start nfs-server.service
- If NFS not there:
 - yum install nfs-utils



The Target Board



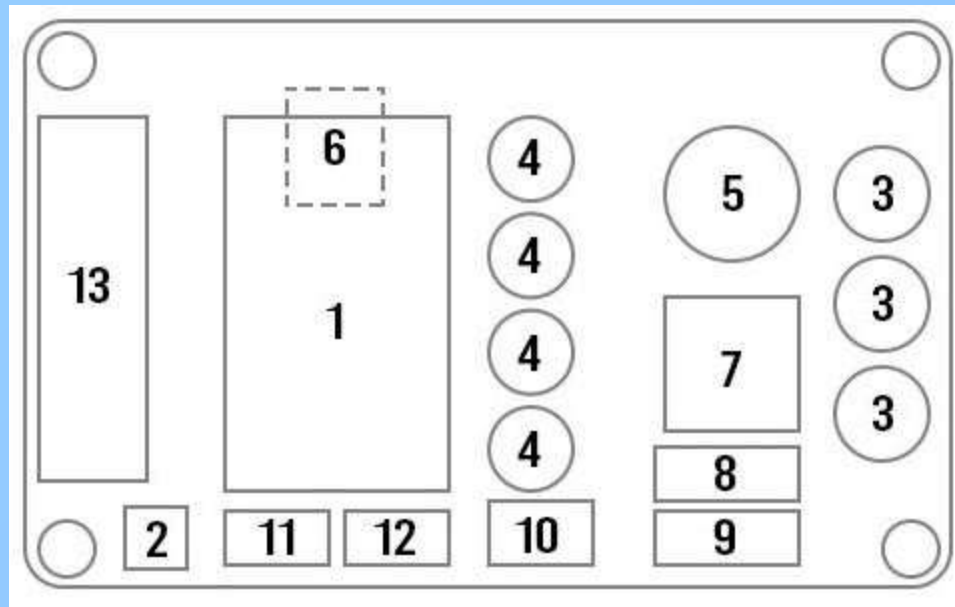


Specifications

- Quad core 1.2 GHZ Broadcom processor, 64-bit ARM Cortex A8
- 1 GB RAM
- Micro SD slot
- BCM43438 wireless LAN and Bluetooth Low Energy (BLE) on board
- 10/100 Ethernet port
- 4 USB 2.0 host ports, Type A
- Serial debug port, 3.3 volt signal levels
- Micro USB, primarily for power



Layout





Layout 2

1. 80x160 TFT display
2. Compass
3. Pushbuttons
4. LEDs
5. Buzzer
6. Analog to digital converter
7. Potentiometer connected to channel 0
8. 4-pin header for I2C bus
9. 4-pin header for UART
10. Double row power header: 3 pins 3V3, 3 pins GND
11. Socket for temperature transducer
12. Socket for IR receiver
13. 3x10 pin header
 - 3 analog in
 - 7 digital I/O: 2 can be configured for PWM, 4 for SPI



Debug Serial Port

Black	GND	1
Red	+5V	Not connected
White	RX	3
Green	TX	4



Changing Boot Parameters

- Pull micro SD card from R Pi
- Mount on workstation
 - `/run/media/<your_user>/boot`
- Edit `cmdline.txt`
 - Change values for `nfsroot=` and `ip=` as appropriate



Connect and Power Up Target

- Connect the network cable and serial adapter
- Run minicom
- Plug in the power supply
- Target boots into Linux
- `ssh root@192.168.15.50`
- Try some shell commands



The target Linux enviroment

- `ls /home`
 - This is where our sample code is stored
- `List the /proc directory`
 - `cat /proc/interrupts`
- `ls -l bin`
 - BusyBox!
- `ifconfig`
 - IP address was set by kernel command line



Our First Program

- On the workstation (from your home directory)
 - `source /opt/arm/environment-setup-cortexa7t2hf-neon-vfpv4-poky-linux-gnueabi`
 - `echo $CC`
 - `cd target_fs/home/src/pi-lib`
 - `make`
 - `cd ../led`
 - `make`

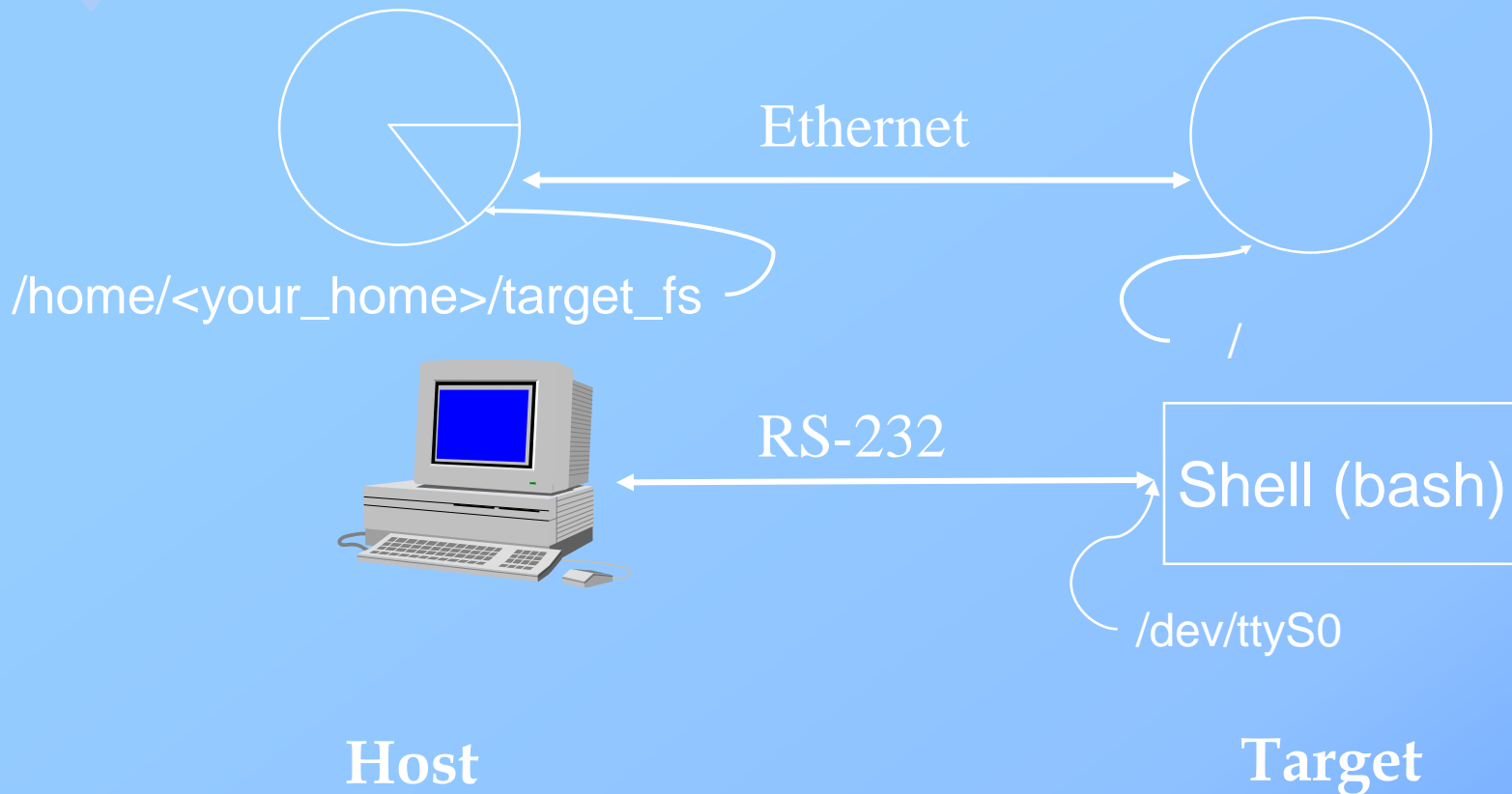


On the Target

- `cd /home/src/pi-lib`
- `cp lib* /lib`
- `ldconfig`
- `cd ../led`
- `./led`

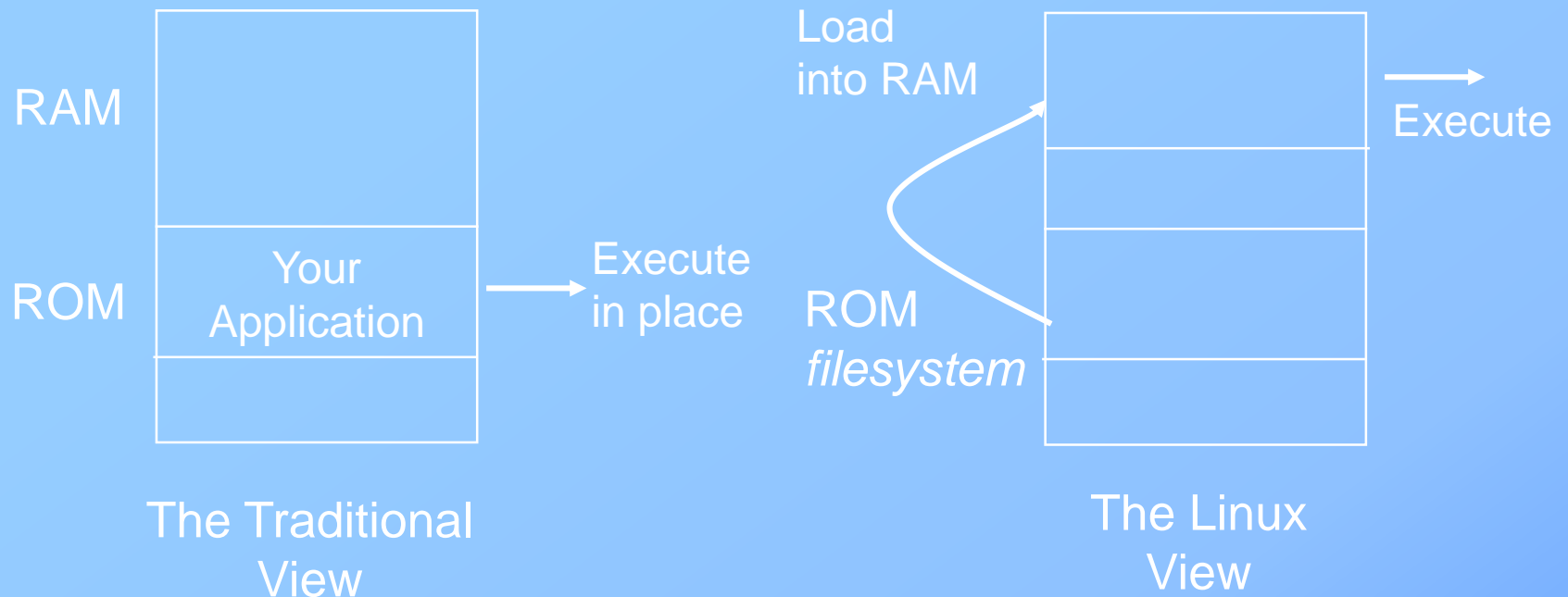


What's going on here?





Two Views of Embedded Programming





Review I

- Install class software
 - Cross build tools
 - Eclipse
 - Kernel source tree
 - Sample code
- Configure Workstation
 - Configure minicom



Review II

- Configure Workstation (cont)
 - Networking
 - Fixed IP address
 - NFS Server
 - Export directory
- Target
 - Fixed IP address specified on kernel command line
 - Kernel booted from micro SD card
 - Root file system mounted over NFS