Embedded Linux Development with the Raspberry Pi

- Overview of the Raspberry Pi
- Setting up the QEMU cross chroot
- Exploring the QEMU cross chroot
- Building a sample SDL application
- Cross building the kernel
- Using GPIO in the shell
- Questions and Answers

Overview of the Raspberry Pi

- Power Supply
- SD Card
- USB Devices
- Ethernet Connection
- HDMI-to-DVI Connection
- Audio Connection
- raspi-config

QEMU Installation

- Install package or build from source?
- Ubuntu 12.04 package is qemu-user-static
- Ubuntu 12.04 package version is only

  1.0.50-2012.03-0ubuntu2
- Ubuntu package has known networking defects

  http://lists.nongnu.org/archive/html/qemu-devel/2012-07/msg03187.html
- Latest QEMU source is 1.2.0 from

  http://wiki.qemu.org/Download

Building QEMU from Source

- Remember to build static binaries
- For Raspberry Pi we only need ARM
- # ./configure –help | less
- # ./configure –target-list=”arm-linux-user” --static
- # make
- # sudo make install
- There is no "uninstall" target...
- ...but we can make our own

Setting up the QEMU cross chroot

- Create a fresh Raspberry Pi SD card
- Boot the Pi from the new card
- Configure the Pi using raspi-config
- Power off the Pi
- Create a tarball from the root file system
- Extract the tarball to a directory
- Chroot into the copy of the root file system

Exploring the QEMU cross chroot

- We can ping Google
- We can run aptitude
- We can run bash scripts
- We can customize our environment
- Is there anything we cannot do?
- No display!
- Tunneling X over SSH doesn't work either

Building a Sample SDL Application

- aliens demo
- http://www.libsdl.org/projects/aliens/
- Requires SDL_mixer and SDL_image
- We need libsdl1.2-dev, libsdl-mixer1.2-dev, libsdl-mixer1.2-dev
- aptitude to the rescue!
- # ./configure
- # make
- Copy to the target
- Enjoy the fun!

Cross Building the Kernel

- http://elinux.org/RPi_Kernel_Compilation
- Using the QEMU cross chroot
  - Very slow
  - Works for any kernel source type
- Using gcc-arm-linux-gnueabihf
  - Very fast
  - Very tricky to get right
- No fun bisecting a kernel using a slow build process

Using GPIO in the Shell

- Important: 3.3 V
- http://elinux.org/RPi_Low-level_peripherals
- Using standard Linux GPIO driver
- Using entries in the procfs virtual file system
- LED output example
- Button input example
- Explaining the scripts

Questions and Answers

- My name is David Cullen
- developer@kerneldriver.org