USB Netconsole

Bring-up Netconsole and USB without a Serial Debug Net

Jason Kridner, *Sitara Apps*, *Texas Instruments*
Board Member, BeagleBoard.org Foundation

Bring-up Netconsole and USB without a Serial Debug Net

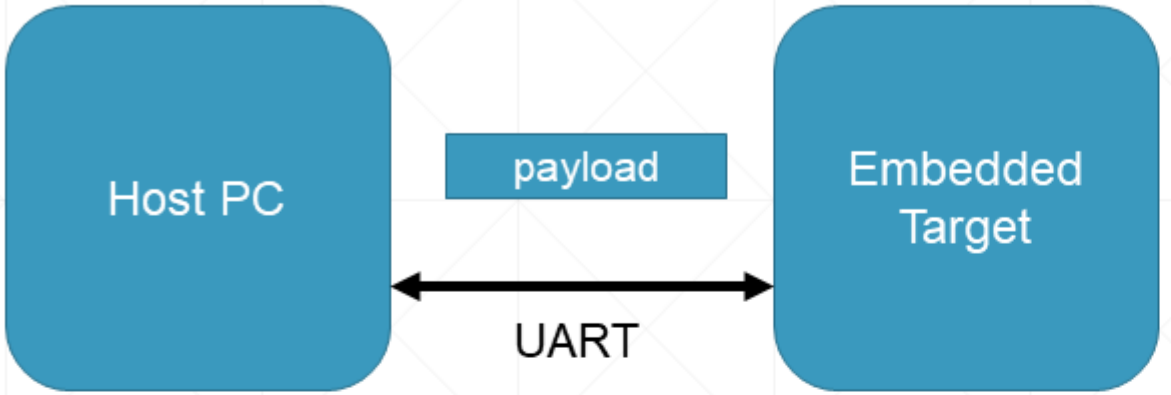
A serial debug console and even JTAG are often assumed for new board bring-up. This presentation will illustrate opportunities for bringing up some designs with neither a serial UART interface or JTAG when doing new bootloader and kernel development. This will include usage of Buildroot to do a complete build management. Utilization of bootloader provided debug resources to the kernel, such as EFI, will also be explored. Demonstration and reproduction materials will be provided on PocketBeagle boards using only a USB connection.

Outline

- What is USB NETCONSOLE?
- What examples can I show?
- Why might it be helpful?
- How could it be better?

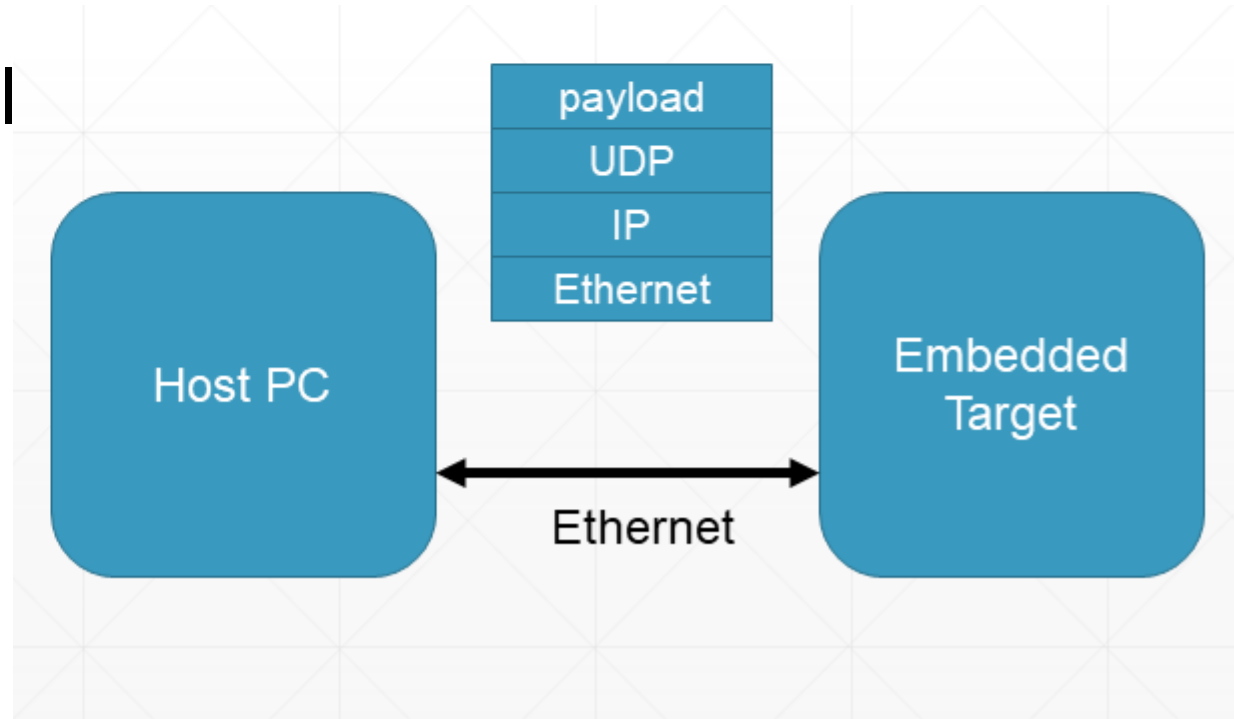
Traditional serial debug

- Typically RX/TX, sometimes CTS/RTS, rarely few others
- Configure baud, stop bits, parity, etc. and go!



What is NETCONSOLE?

- Essentially networked serial
- Nothing but UDP packets, typically over port 6666
- Specify IP addresses and go!



How to get the IP addresses?

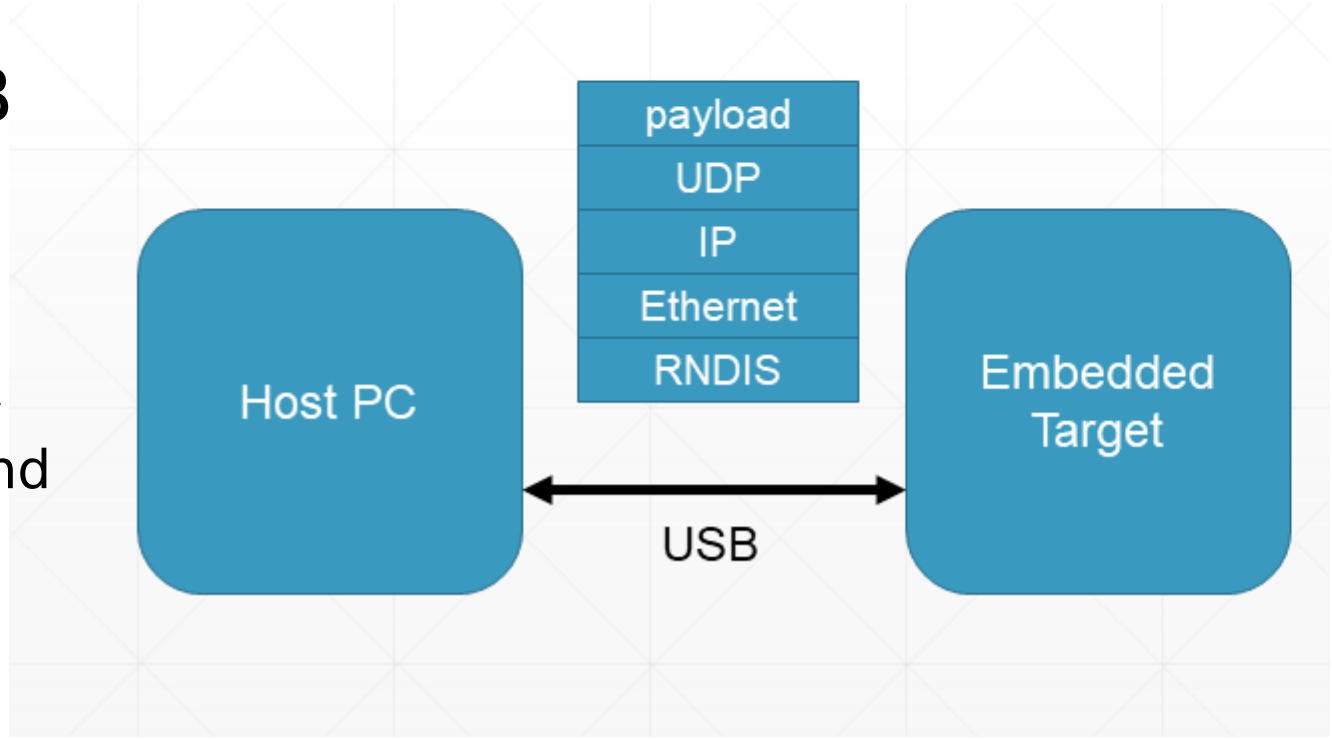
- ARP – get server IP
  - Provides IP addresses for devices on the LAN
- BOOTP/DHCP – get own IP from server
  - Provides IP address (and more) for the target device
  - Can specify a file to boot

What are USB “gadgets”?

- USB client drivers running on target, ie., slave not host
  - **Act** like the USB storage disk, not **read** the USB storage disk
- USB clients are grouped in classes to simplify host
- RNDIS is one class driver that provides USB Ethernet
  - CDC ECM is another, but we aren’t using that one today

What is USB NETCONSOLE?

- NETCONSOLE over USB network
  - We're using RNDIS for this example
  - RNDIS+TFTP is supported by AM335x boot ROM, u-boot and Linux



Testing PocketBeagles

- My original application of USB NETCONSOLE in u-boot
- Used in production today
  - Provides serial numbers
  - Performs basic tests
  - Indicates success via GPIO LEDs
- Gist at bbb.io/pbncgist

Reference server – node-beagle-boot

- Implemented in node.js using node-usb
- Target cross-platform environments
  - Doesn’t change host configuration, ie. network stack
  - Targeting Electron apps today
  - Relatively easy to migrate to WebUSB in the future
- Implements RNDIS, ARP, BOOTP/DHCP and netconsole

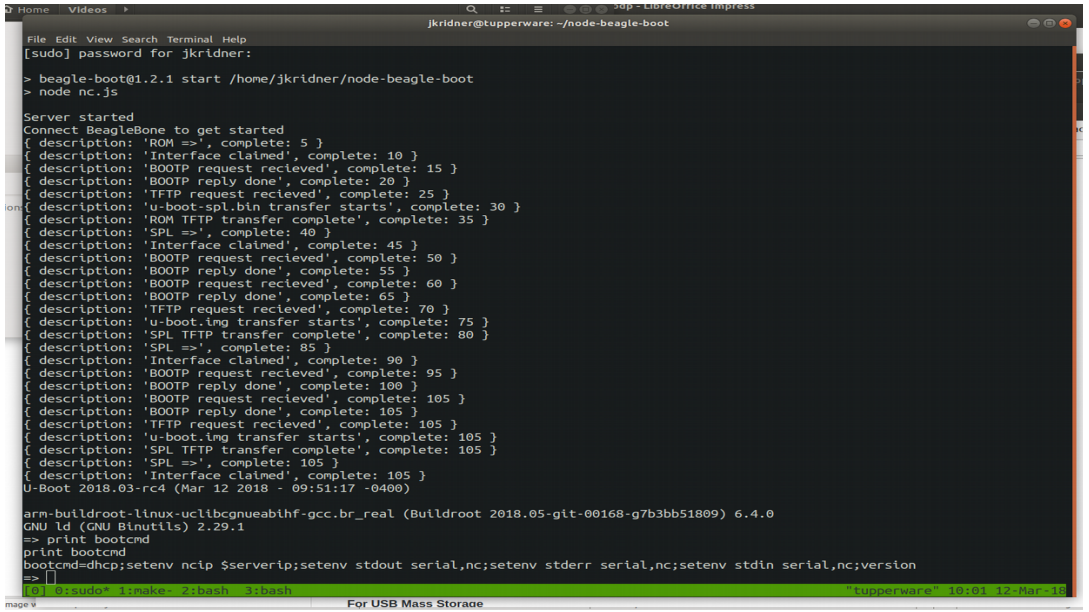
My goals for node-beagle-boot

- Interactivity as early as possible
- Minimize hardware target dependencies
  - While minimizing host software dependencies
  - Running in browser would be ideal, Etcher is also a target
- Eventually, keeping interactivity into userspace

What can I do in u-boot?

- A lot!
  - Memory reads/writes
  - General I/O: GPIO, I2C, SPI
  - Storage: MMC/SD, USB, SPI
- Boot kernel
  - setenv stdin serial;setenv stdout serial;tftp ${loadaddr} zImage;tftp ${rdaddr} rootfs.cpio.uboot;tftp ${fdtaddr} am335x-pocketbeagle.dtb;setenv stdout serial,nc;setenv stdin serial,nc
  - run ramargs;bootz ${loadaddr} ${rdaddr} ${fdtaddr}

node-beagle-boot with u-boot and Buildroot



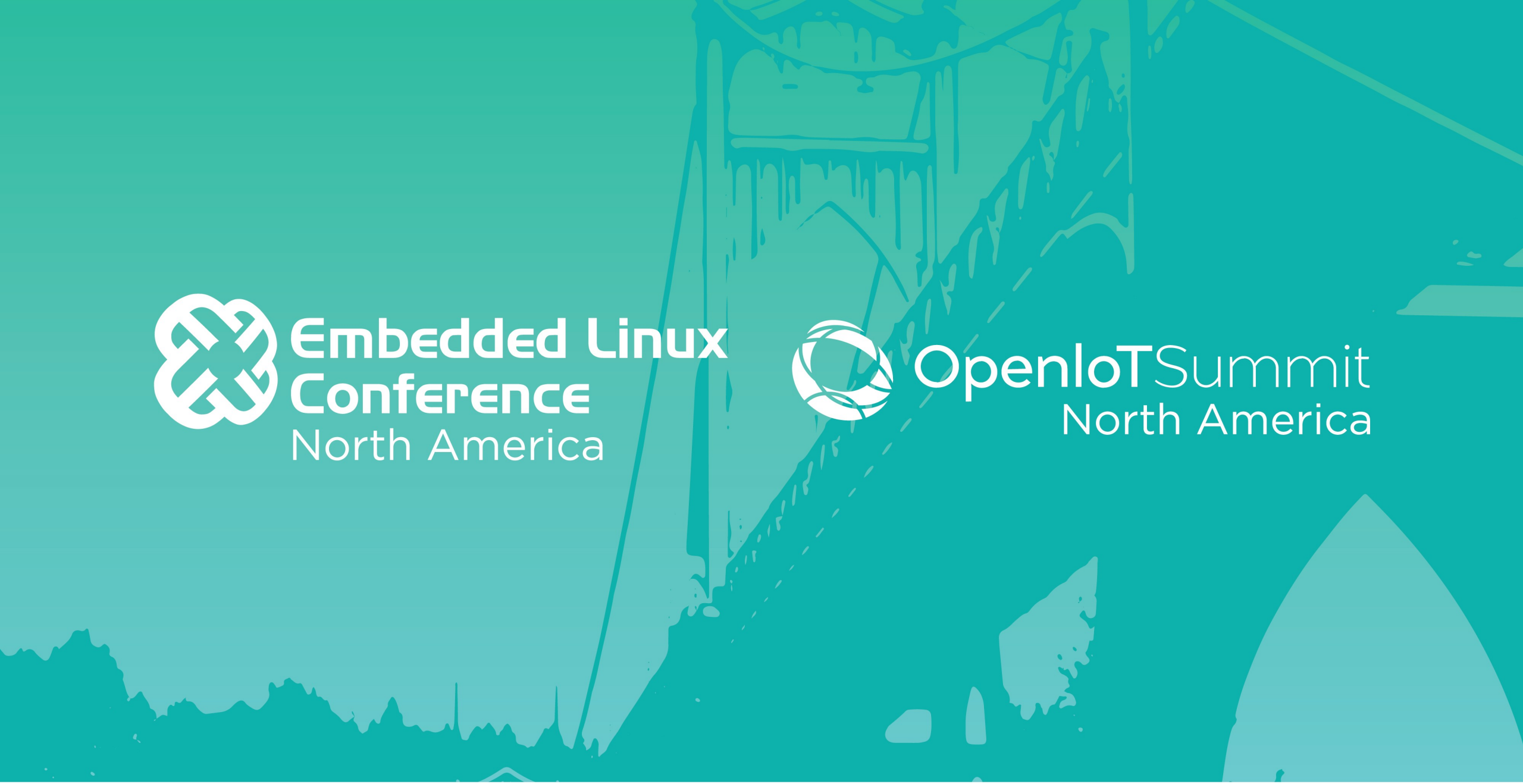
- jadonk/node-beagle-boot
- jadonk/buildroot
- npm install
- sudo npm start
- Connect PocketBeagle
- At u-boot netconsole

Next steps

- Configure kernel to use NETCONSOLE
  - libcomposite and ConfigFS not up until *way* late
  - Planning to try g_multi from kernel command-line
  - Eventually, use UEFI and transition
- U-boot crashes when NETCONSOLE is on and you TFTP
- Porting and integration
  - Resolve any remaining cross-platform issues
  - Etcher.io integration
  - WebUSB

UEFI

- Last year's ELC presentation: Marrying U-Boot, uEFI and grub2 - Alexander Graf, SUSE
  - https://www.youtube.com/watch?v=qJAkJ3nmWgM
  - slides/Marrying_U-Boot_UEFI_and_grub.pdf
- Currently, we lose connection between NETCONSOLE and the target when Linux boots
- Linux has a provision to use existing console


Embedded Linux
Conference
North America
OpenIoTSummit
North America