Linux Based Embedded Systems

Kevin Martin - AD/EES

16 August 2018

- A little history
- PC104
- MitySOM Cyclone V (FPGA w/ ARM HPS)
- Embedded Linux
- Real-Time Application Interface (RTAI)

Embedded Systems at Fermilab in the year 2000

- VME Chassis
- PowerPC based CPU Board
- Various other IO Boards
- VxWorks
- Cost at least $10K

Linux Based Embedded Systems

The challenge

- Embedded Power Supply controller
    - Should only need power and ethernet. All other cabling is eliminated.
    - Must provide for Control System connectivity
    - < $1000 for incremental Hardware/Software costs. This will allow for extensive justifiable use.
    - Sixteen 16-bit ADC Channels @ 1Khz per channel
    - Four 16-bit DAC channels
    - 16 bits of Digital I/O for digital status and control
    - Should have small footprint to enable embedding
    - Built-in Transient Recorder to provide PS diagnostics
    - PS Ramp generation functionality possibly to be added.

The Solution

- PC/104 (ISA) and/or PC/104+ (ISA & PCI)
  - Small formfactor PC – 3.5” x 3.75”
  - Relatively inexpensive – commodity pricing
  - A widely supported standard that still being updated/upgraded
  - Many manufacturers/options.
  - Stackable expansion IO
- x86 based CPU board
- Multi-Function DAQ Board
- Linux OS with RTAI Realtime Extension
- Implemented Front-Ends: PSRAC, BuLB, RBEX, LiLens/Pmag (MC), HWR/SSR1 QPM (PIP2-IT)

Linux Based Embedded Systems

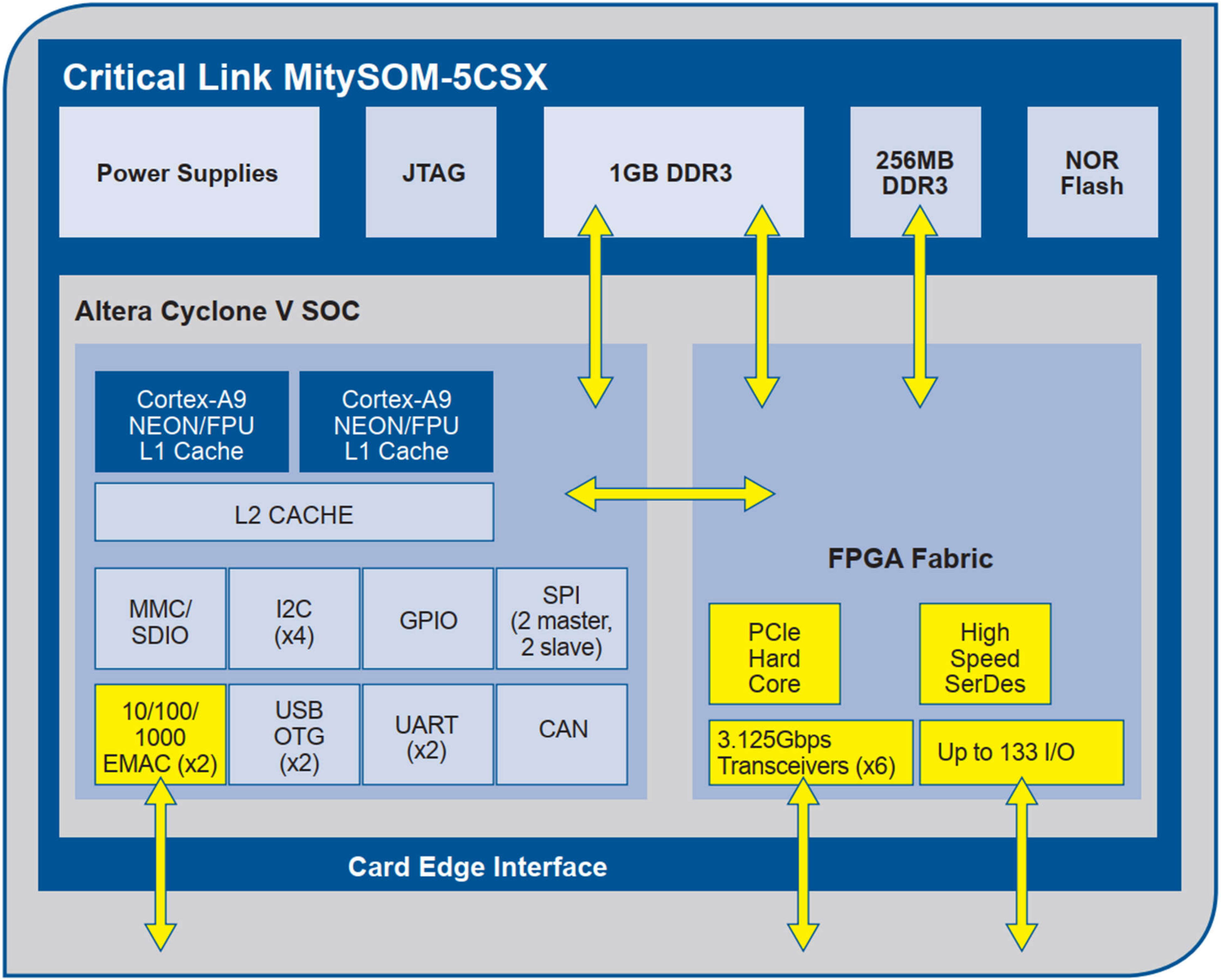
Linac Marx Modulators

- Unique application never done before at Fermilab
- This meant we needed a flexible dev platform to be able try various approaches
- Application demanded an FPGA to do high speed Marx Cell fire commands and real-time feedback
- Lots of IO and high speed state machines needed
- Application also need a embedded CPU to do Modulator pulse learning and interface to ACNET
- High speed FPGA to CPU communication was needed
- Only “soft” real-time processing needed (15Hz)

Linux Based Embedded Systems

Critical Link MitySOM

- Since EES was new to FPGAs, the MitySOM helped us through the FPGA learning curve
- We didn’t have to implement everything, just the application specific things
- Critical Link provides documentation and development board design schematics


Critical Link MitySOM-5CSX
Power Supplies
JTAG
1GB DDR3
256MB DDR3
NOR Flash
Altera Cyclone V SOC
Cortex-A9 NEON/FPU L1 Cache
Cortex-A9 NEON/FPU L1 Cache
L2 CACHE
MMC/ SDIO
I2C (x4)
GPIO
SPI (2 master, 2 slave)
10/100/ 1000 EMAC (x2)
USB OTG (x2)
UART (x2)
CAN
FPGA Fabric
PCIe Hard Core
High Speed SerDes
3.125Gbps Transceivers (x6)
Up to 133 I/O
Card Edge Interface

Linux Based Embedded Systems

Embedded Linux

- Customizable kernel (real-time patches, drivers)
- Minimal footprint – size, robustness
- Network bootable
- Support multiple platforms – x86, ARM, PowerPC
- Minimal disk space requirement
- Maintainable in the long term – OS/Application upgrades and patches

Linux Based Embedded Systems

Embedded Linux

- EES went through several iterations before settling on our current solution
- Looked into SLF: 1.5GB minimum size, x86 only
- TimeSys Linux – embedded Linux with proprietary development environment and real-time OS patches: didn’t provide acceptable hard real-time
- Real Time Application Interface (RTAI)
- Linux From Scratch (LFS): customizable, tedious, little or no automation, hard to replicate, not small enough (200-300MB)

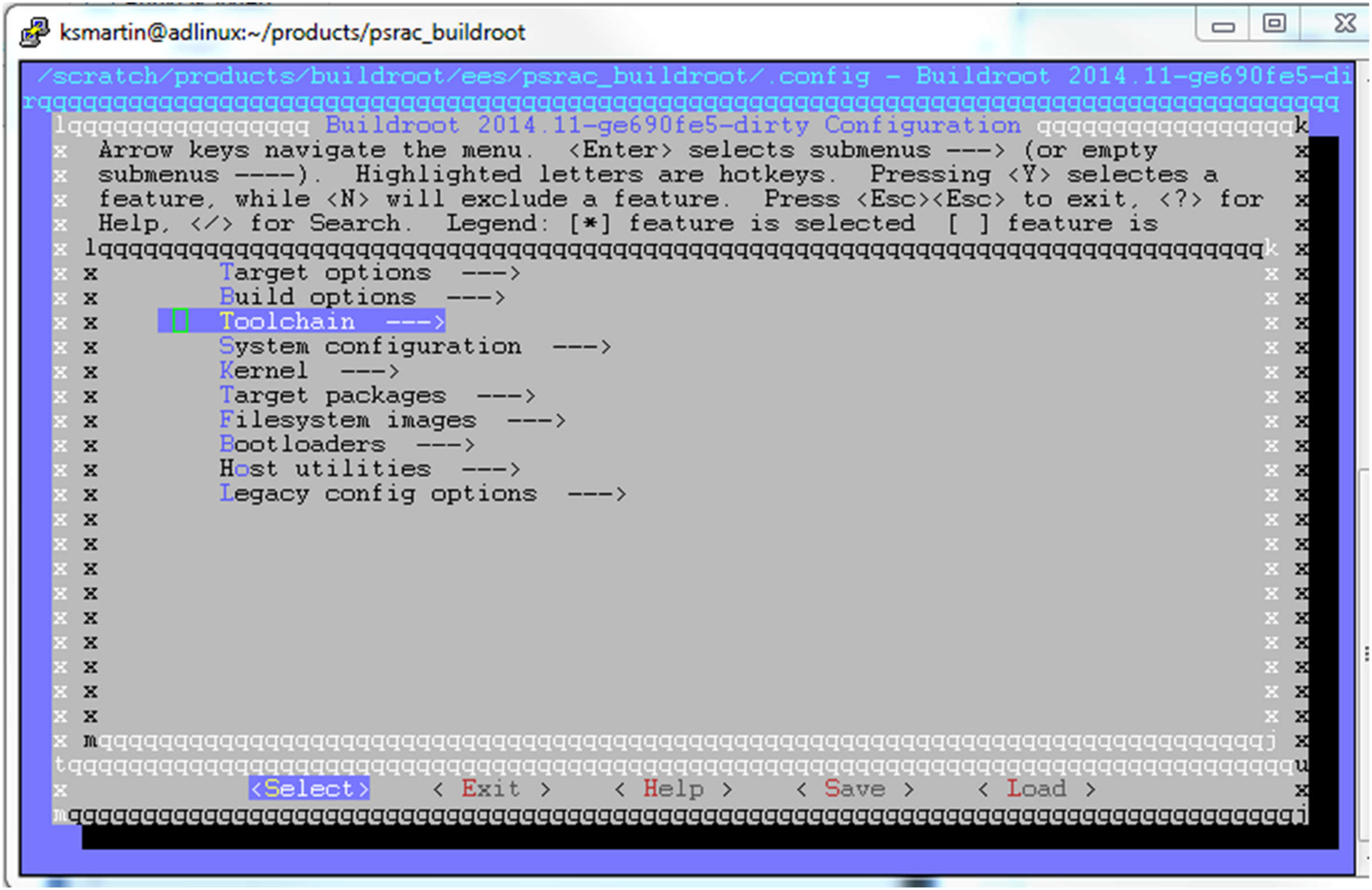
Embedded Linux - Buildroot

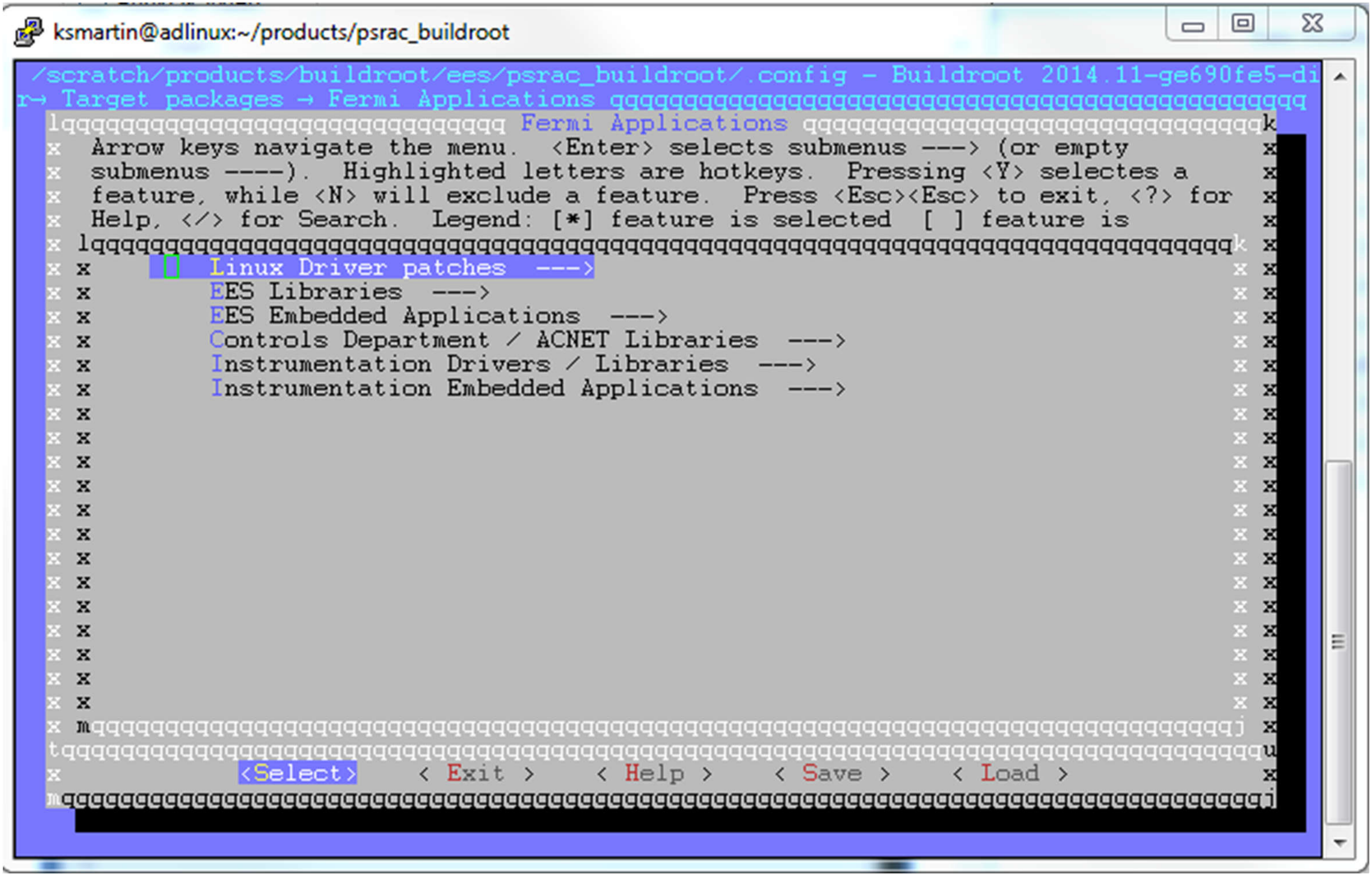
- Automated system for building Linux kernels and root filesystems
- Supports many toolchains, target architectures, and hardware platforms
- Integrated system for building and deploying application software
- Supports ucLibc and Busybox for reducing the size footprint of our systems
- Supports hundreds of optional software packages
- Creating custom packages (our applications) is supported

Linux Based Embedded Systems

Embedded Linux – Buildroot (cont.)

- Our bzImage files which contain the Linux Kernel, the root filesystem, and Fermi software (application, ACNET FE etc) are ~25MB
- We are able to do network booting using the GRUB or uBoot bootloaders
- Boots into a RAM disk. The local flash disk is only used for node specific information (IP Name/Number etc)
- New Buildroot Development Node (adlinux.fnal.gov) has been recently created (CD VM)


ksmartin@adlinux:~/products/psrac_buildroot
/scratch/products/buildroot/ees/psrac_buildroot/.config - Buildroot 2014.11-ge690fe5-di
Buildroot 2014.11-ge690fe5-dirty Configuration
Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty
submenus ----). Highlighted letters are hotkeys. Pressing <Y> selectes a
feature, while <N> will exclude a feature. Press <Esc><Esc> to exit, <?> for
Help, </> for Search. Legend: [*] feature is selected [ ] feature is
Target options --->
Build options --->
Toolchain --->
System configuration --->
Kernel --->
Target packages --->
Filesystem images --->
Bootloaders --->
Host utilities --->
Legacy config options --->
<Select> < Exit > < Help > < Save > < Load >


ksmartin@adlinux:~/products/psrac_buildroot
/scratch/products/buildroot/ees/psrac_buildroot/.config - Buildroot 2014.11-ge690fe5-di
r→ Target packages → Fermi Applications
Fermi Applications
Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty
submenus ----). Highlighted letters are hotkeys. Pressing <Y> selectes a
feature, while <N> will exclude a feature. Press <Esc><Esc> to exit, <?> for
Help, </> for Search. Legend: [*] feature is selected [ ] feature is
Linux Driver patches --->
EES Libraries --->
EES Embedded Applications --->
Controls Department / ACNET Libraries --->
Instrumentation Drivers / Libraries --->
Instrumentation Embedded Applications --->
<Select> < Exit > < Help > < Save > < Load >

Real-Time Application Interface (RTAI)

- Provides hard Real-Time (low latency, deterministic interrupts) on Linux systems even if they have only modest hardware
- x86 only (for the most part)
- Requires specific versions of “vanilla” kernels (i.e. from kernel.org)
- In a nutshell, RTAI provides a small RTOS that runs Linux as it’s lowest priority task.
- Minimal changes to the Linux kernel allow access to the RTAI API in order to start RTAI tasks and to hook them to interrupts

Linux Based Embedded Systems

Real-Time Application Interface (RTAI)

- Depending on your platform and potential Linux Kernel version limitations, it is now possible that a properly configured Linux kernel with the latest PREMPT-RT patches makes the use of RTAI unnecessary.
- In other words Linux can be (depending on your definition and your application) hard real-time.