



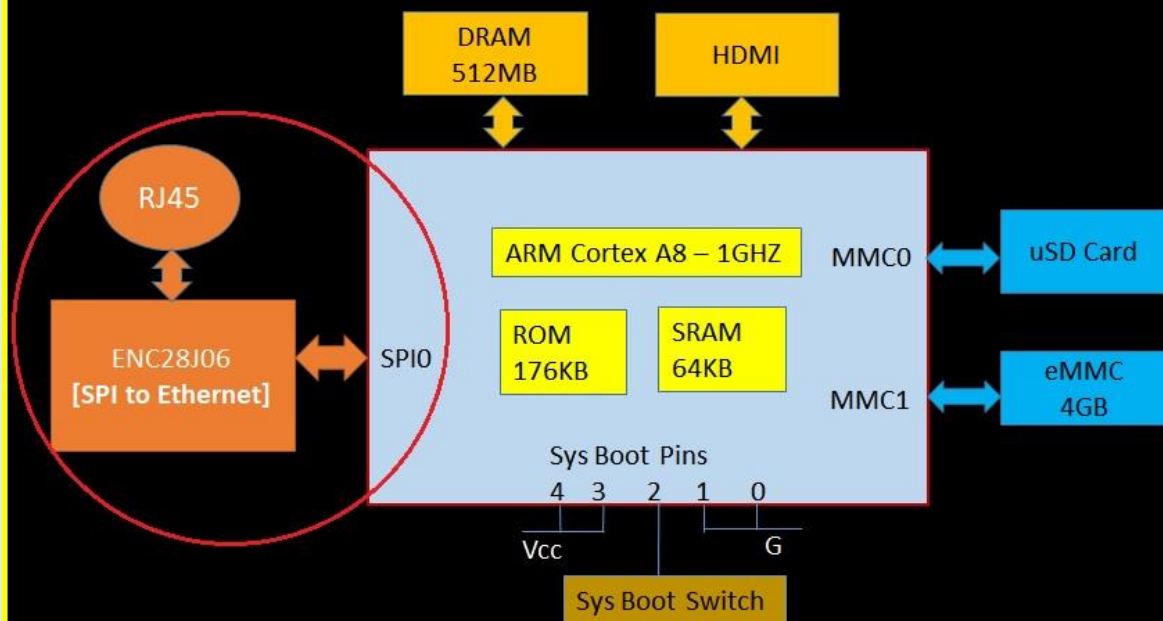
## Network & SPI Device Drivers

In U-boot & Kernel Space

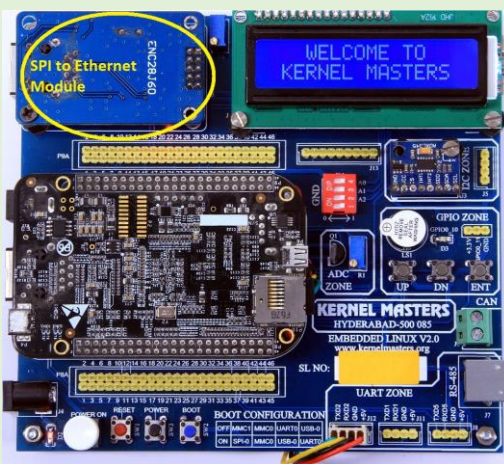
Device Example:

Add SPI to Ethernet Slave Device with AM3358

### Ethernet - AM335x Interface with Beagle Bone Black



Sysboot SW	Sysboot pins	1 <sup>st</sup> Order	2 <sup>nd</sup> Order	3 <sup>rd</sup> Order	4 <sup>th</sup> Order
OFF	11100	MMC1	MMC0	UART0	USB0
ON	11000	SPIO	MMC0	USB0	UART0



#### Hardware used

- The practical exercises will be run on a Beagle Bone Black (BBB) with a Cortex ARM.
- Add ENC28J06 SPI to Ethernet module with Beagle Bone Black.
- All exercises will be applicable to any other type of board supported by Linux.
- Online practical demonstration for Ethernet Driver, SPI Master & Slave drivers and User space Drivers on BBB. Later on you can buy and practise, support will be provided. Material will be provided with step by step procedure for lab guidance.

#### Prerequisites:

- Assumption zero knowledge in Network, SPI Communication Protocol.
- We assume that attendees are fully fluent in C, data structures and that the Linux/Unix command line is a familiar environment.
- Linux System Programming & Device Drivers.
- Embedded Linux Porting knowledge.

# KERNEL MASTERS

## Network & SPI Framework Detailed Agenda

### Linux Network & SPI Drivers Detailed Syllabus: Master, Slave & User Space Drivers

#### Session 1: Network & SPI High Level Analysis [Overview]

##### Getting Started with Network

- OSI & TCP/IP Layers
- Ethernet protocol format

##### Getting Started with SPI

- SPI communication protocol basics/terminology
  - SPI Modes
  - SPI Communication Protocol Format
- AM3358 Controller (master) SPI specifications & Block Diagram
- ENC28J06 (SPI to Ethernet) Specifications

#### Session 2: Network & SPI Low Level Analysis [Detailed Analysis]

- Understanding SPI Registers for AM3358 Controller.
- Understanding SPI to Ethernet [ENC28J06] Specifications, comm. Protocol format and register programming model.
- Ethernet - AM335X interface Block Diagram.

#### Session 3: Developing Network & SPI Drivers in u-boot space

##### U-boot Level

- Add SPI Mux/Pad Configuration in board specific mux file.
- Enable enc28j06, am335x SPI drivers and *sspi*, *ping* and *dhcp* commands in u-boot source code.
- Test basic communication with slave device using *sspi* command in u-boot command prompt.
- Test basic network communication using *ping* command in u-boot command prompt.

#### Session 4: Developing Network & SPI Drivers in Kernel Space

##### Kernel Level

- Add Ethernet Interrupt number & SPI master, slave driver platform data in to Device tree source code
- Integrating platform Ethernet & SPI Master and slave driver with framework.
- Network & SPI Driver & Device Registrations
- Network & SPI Kernel data structures
- Network & SPI Device Operations
- Check Probe and initialization functions both Master and slave drivers.
- Registering Ethernet & SPI master/adaptor and slave drivers

#### Session 5: Developing Network & SPI applications in User Space

##### User Level

- Disable SPI Slave driver and test SPI Master Driver with User space driver (spi-dev) absent of SPI Slave driver.
- Enable SPI Slave driver and test SPI slave driver with device specific applications.

#### Session 6: Understanding the complete flow [Network & SPI Framework]

- Understanding Network & SPI Master, slave initialization and device operation source code flow from bottom to top vice versa.
- Analyse network raw packets using Wireshark tool.
- Understanding SPI Interrupt flow with SPI Debug options.

**Note:** All Sessions are highly interactive hands-on-sessions with KM-BBB.

Authored and Compiled By: **Boddu Kishore Kumar**  
Email: [kishore@kernelmasters.org](mailto:kishore@kernelmasters.org)  
Reach us online: [www.kernelmasters.org](http://www.kernelmasters.org)  
Contact: 9949062828