



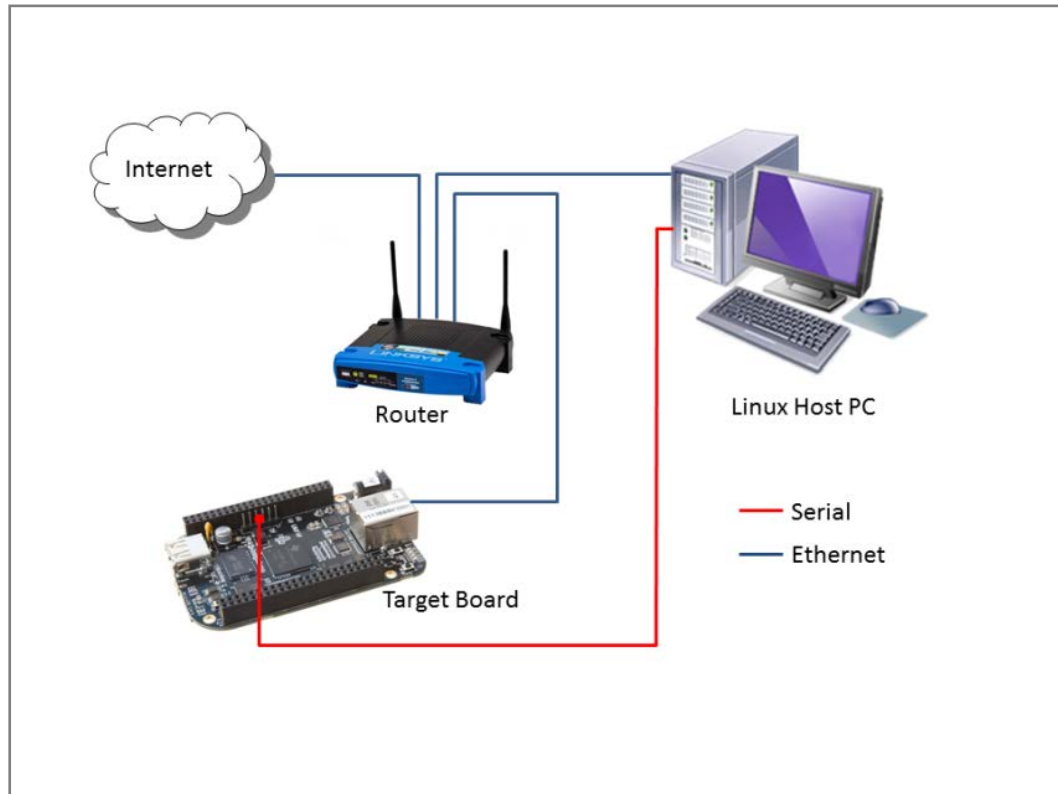
# **Linux Board Port Overview for Sitara AM-Class Devices: AM33x, AM43x, and AM57x**

# Elements of a Linux Board Port

# Section overview: Elements of a Linux board port

- Development environment using the TI Processor Linux SDK
- PinMux Tool
- Kernel image
- Design considerations and the DTS (Device Tree Source) file
- Root file system
- SDK lifecycle considerations: From initial board port to production

# Linux board port elements: Environment



Current 64-bit LTS Ubuntu Desktop  
(development machine)

PROCESSOR-SDK-LINUX-AMxx  
(installed on development machine)

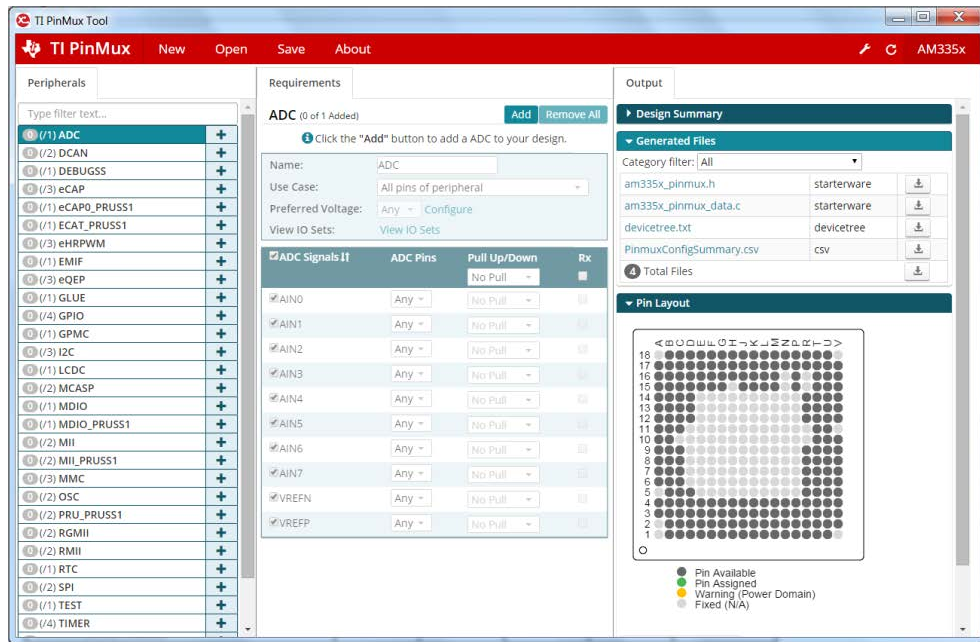
Texas Instruments EVM  
(for processor of target product)

# Linux board port elements: PinMux Tool

- Each peripheral requires a pinmux definition.
- The PinMux Tool assists with determining a mux configuration for a system based on use-case peripheral requirements.
- The output from the tool is then incorporated into DTS files for binding peripherals to a system application.
- Below is the PinMux Tool download link and associated Wiki resources.

<http://www.ti.com/tool/pinmuxtool>

[http://processors.wiki.ti.com/index.php/TI\\_PinMux\\_Tool\\_v4](http://processors.wiki.ti.com/index.php/TI_PinMux_Tool_v4)





# So... what is a pinmux?

IOSets for DCAN0

- Type filter text...
- DCAN0\_IOSet\_1
- DCAN0\_IOSet\_2
- DCAN0\_IOSet\_3

Signal	Device Pin
dcan0_tx	J18 ( gmii1_txd3 )
dcan0_rx	K15 ( gmii1_txd2 )

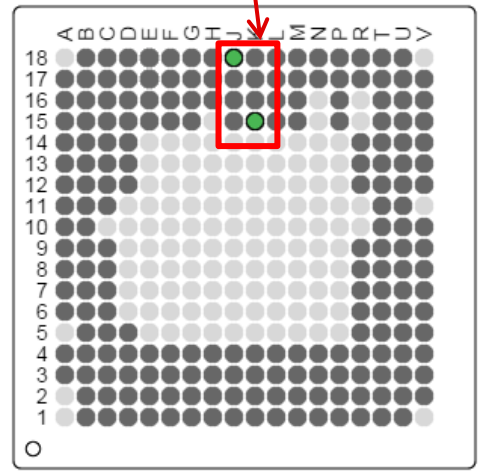
# So... what is a pinmux?

IOSets for DCAN0

- Type filter text...
- DCAN0\_IOSet\_1
- DCAN0\_IOSet\_2
- DCAN0\_IOSet\_3

Signal	Device Pin
dcan0_tx	J18 ( gmii1_txd3 )
dcan0_rx	K15 ( gmii1_txd2 )

IO Set 1



# So... what is a pinmux?

IOSets for DCAN0

Type filter text...

DCAN0\_IOSet\_1

DCAN0\_IOSet\_2

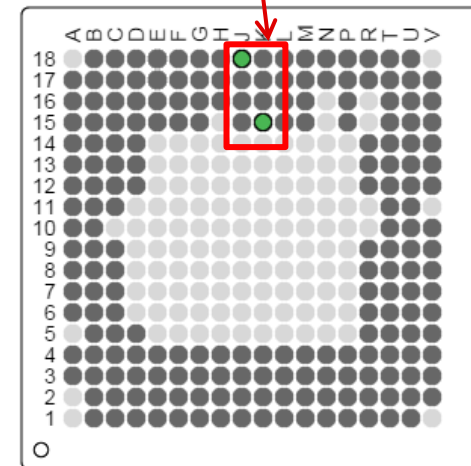
DCAN0\_IOSet\_3

Signal	Device Pin
dcan0_tx	J18 ( gmii1_txd3 )
dcan0_rx	K15 ( gmii1_txd2 )

Datasheet snippet showing signal names and mux modes

ZCZ BALL NUMBER [1]	PIN NAME [2]	SIGNAL NAME [3]	MODE [4]	TYPE [5]
J18	MII1_TXD3	gmii1_txd3	0	O
		dcan0_tx	1	O
		rgmii1_txd3	2	O
		uart4_rxd	3	I
		mcasp1_fsx	4	I/O
		mmc2_dat1	5	I/O
		mcasp0_fsr	6	I/O
		gpio0_16	7	I/O

IO Set 1





# So... what is a pinmux?

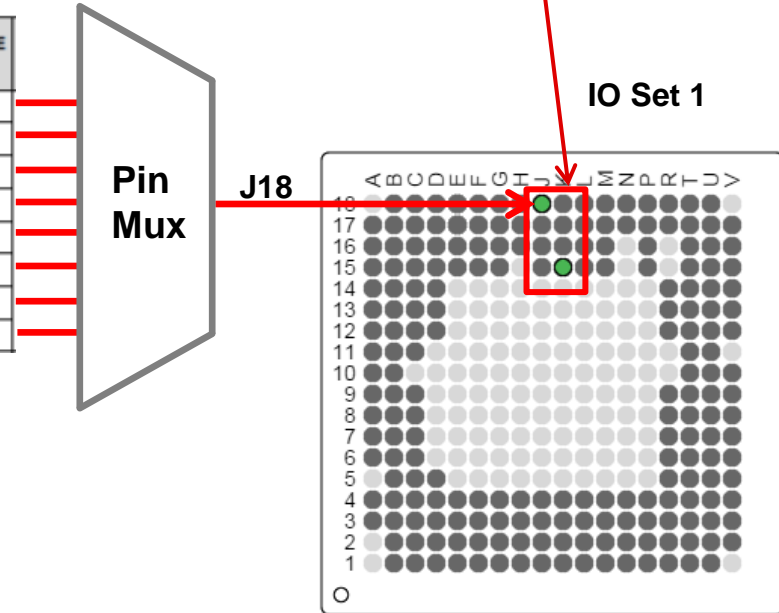
IOSets for DCAN0

- Type filter text...
- DCAN0\_IOSet\_1
- DCAN0\_IOSet\_2
- DCAN0\_IOSet\_3

Signal	Device Pin
dcan0_tx	J18 ( gmii1_txd3 )
dcan0_rx	K15 ( gmii1_txd2 )

Datasheet snippet showing signal names and mux modes

ZCZ BALL NUMBER [1]	PIN NAME [2]	SIGNAL NAME [3]	MODE [4]	TYPE [5]
J18	MII1_TXD3	gmii1_txd3	0	O
		dcan0_tx	1	O
		rgmii1_txd3	2	O
		uart4_rxd	3	I
		mcasp1_fsx	4	I/O
		mmc2_dat1	5	I/O
		mcasp0_fsr	6	I/O
		gpio0_16	7	I/O



# So... what is a pinmux?

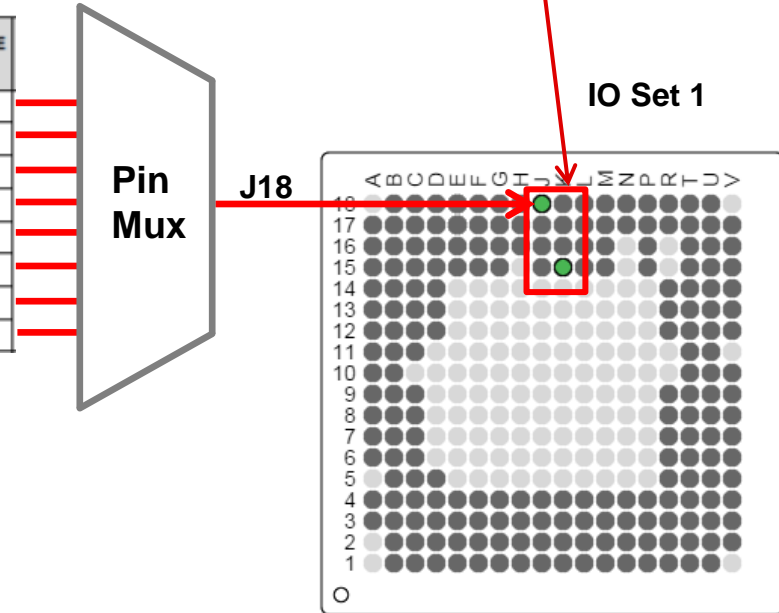
IOSets for DCAN0

- Type filter text...
- DCAN0\_IOSet\_1
- DCAN0\_IOSet\_2
- DCAN0\_IOSet\_3

Signal	Device Pin
dcan0_tx	J18 ( gmii1_txd3 )
dcan0_rx	K15 ( gmii1_txd2 )

Datasheet snippet showing signal names and mux modes

ZCZ BALL NUMBER [1]	PIN NAME [2]	SIGNAL NAME [3]	MODE [4]	TYPE [5]
J18	MII1_TXD3	gmii1_txd3	0	O
		dcan0_tx	1	O
		rgmii1_txd3	2	O
		uart4_rxd	3	I
		mcasp1_fsx	4	I/O
		mmc2_dat1	5	I/O
		mcasp0_fsr	6	I/O
		gpio0_16	7	I/O



# So... what is a pinmux?

IO Sets for DCAN0

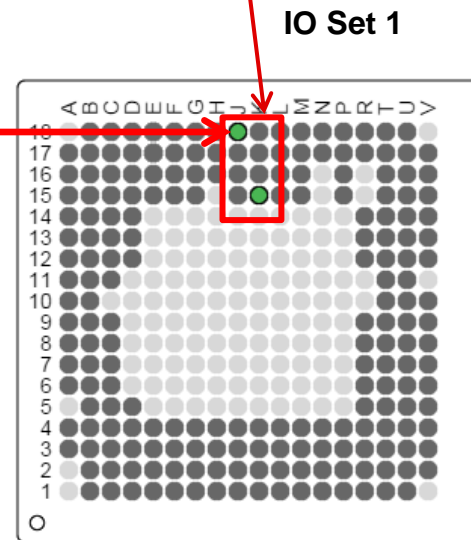
- Type filter text...
- DCAN0\_IOSet\_1
- DCAN0\_IOSet\_2
- DCAN0\_IOSet\_3

Signal	Device Pin
dcan0_tx	J18 ( gmii1_txd3 )
dcan0_rx	K15 ( gmii1_txd2 )

Datasheet snippet showing signal names and mux modes

ZCZ BALL NUMBER [1]	PIN NAME [2]	SIGNAL NAME [3]	MODE [4]	TYPE [5]
J18	MII1_TXD3	gmii1_txd3	0	O
		dcan0_tx	1	O
		rgmii1_txd3	2	O
		uart4_rxd	3	I
		mcasp1_fsx	4	I/O
		mmc2_dat1	5	I/O
		mcasp0_fsr	6	I/O
		gpio0_16	7	I/O

J18 = MII1\_TXD3.dcan0\_tx



# So... what is a pinmux?

IOSets for DCAN0

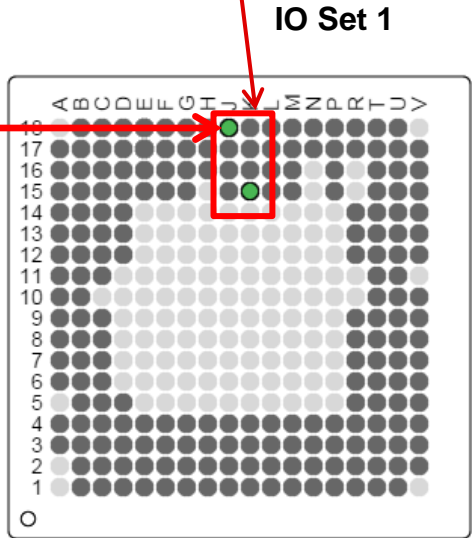
- Type filter text...
- DCAN0\_IOSet\_1
- DCAN0\_IOSet\_2
- DCAN0\_IOSet\_3

Signal	Device Pin
dcan0_tx	J18 ( gmii1_txd3 )
dcan0_rx	K15 ( gmii1_txd2 )

Datasheet snippet showing signal names and mux modes

ZCZ BALL NUMBER [1]	PIN NAME [2]	SIGNAL NAME [3]	MODE [4]	TYPE [5]
J18	MII1_TXD3	gmii1_txd3	0	O
		dcan0_tx	1	O
		rgmii1_txd3	2	O
		uart4_rxd	7	I
		mcaspl_fsx	4	I/O
		mmc2_dat1	5	I/O
		mcaspl_fsr	6	I/O
		gpio0_16	7	I/O

J18 = MII1\_TXD3.dcan0\_tx

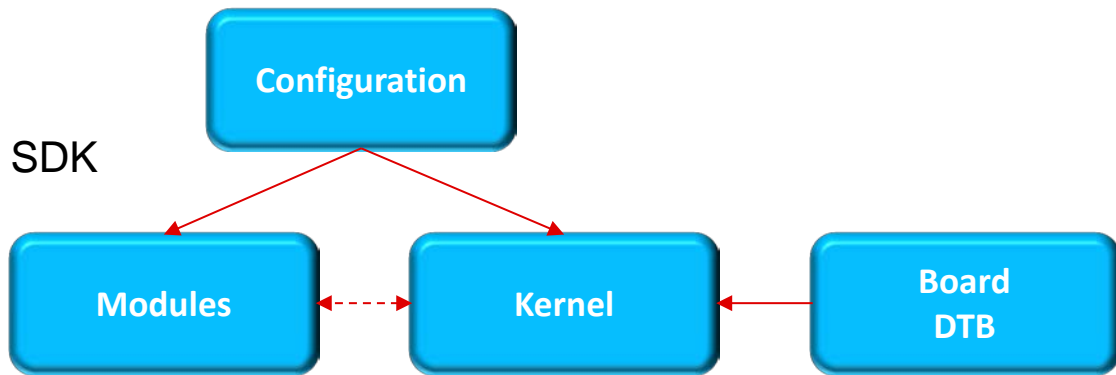


```
/* This file was auto-generated by TI PinMux on 1/24/2018 at 10:46:31 AM. */  
/* This file should only be used as a reference. Some pins/peripherals, */  
/* depending on your use case, may need additional configuration. */  
  
mydcan1_pins_default: mydcan1_pins_default {  
    pinctrl-single,pins = <  
        0x120 ( PIN_INPUT | MUX_MODE1 ) /* (K15) gmii1_txd2.dcan0_rx */  
        0x11c ( PIN_OUTPUT | MUX_MODE1 ) /* (J18) gmii1_txd3.dcan0_tx */  
    >;  
};
```

# Linux board port elements: Linux kernel

- The TI Kernel in Processor SDK Linux has at least these elements:

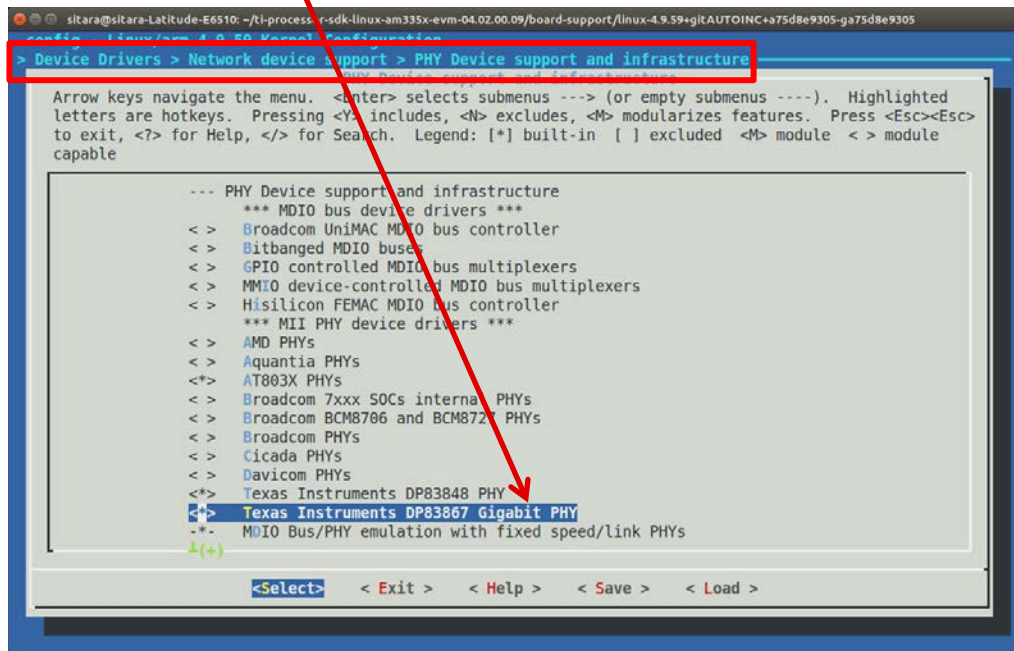
- The kernel configuration
- Modules compiled against the TI SDK kernel configuration
- A board DTB (Device Tree Blob) file



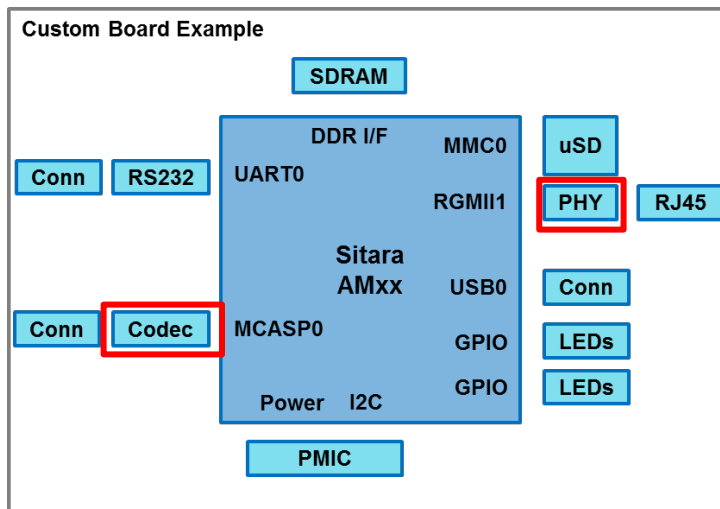
- The system developer should use the kernel from the TI SDK or a tagged release of the TI Kernel Tree.
- The SDK default kernel configuration is recommended to get started with the port.

# Linux system design considerations

```
grep -r dp83867 drivers/net/phy/  
drivers/net/phy/Makefile:obj-$(CONFIG_DP83867_PHY) += dp83867.o  
drivers/net/phy/dp83867.c:#include <dt-bindings/net/ti-dp83867.h>  
drivers/net/phy/dp83867.c:....
```

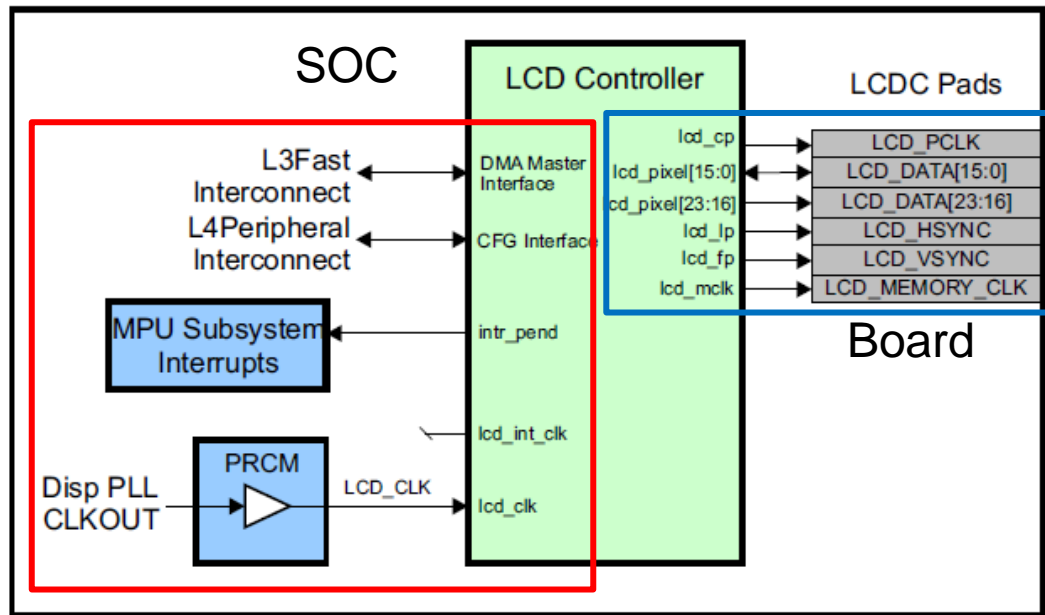


## Custom board



# Linux board port elements: DTS file

- What does the DTS file do?
- The DTS file enables a binding process of selected peripherals to a custom board.
- Do you need to account for all of the signals shown in the block diagram? YES
- Is the board port developer responsible for identifying “all” settings? NO



# DTS file components

TI EVM DTS files may include:

- DTSI (Device Tree Source Include) files
- Other DTS files
- Here the am33xx.dtsi is the processor include file

Arch

SOC

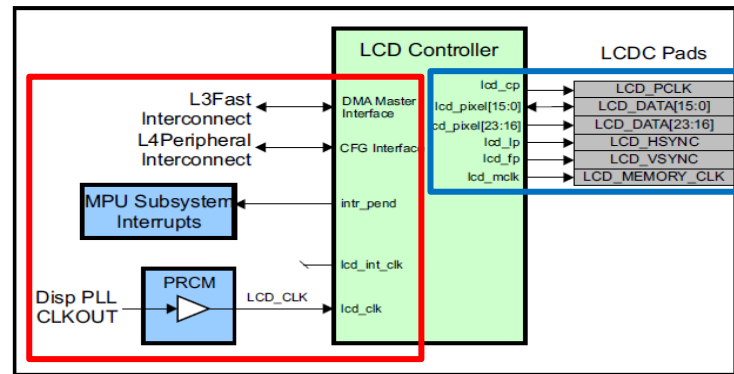
Board

**am33xx.dtsi**

**dt-bindings/gpio/gpio.h**  
**dt-bindings/pinctrl/am33xx.h**  
**skeleton.dtsi**  
**am33xx-clocks.dtsi**

**am335x-boneblack.dts**

**am335x-bone-common.dtsi**





# DTS file example structure

- HelloWorld-like minimal board DTS file (initramfs)
- Defines the Processor used
- UART node and supporting pinmux
- Can also use EVM DTS file and disable all the nodes ... except UART.

```
/dts-v1/;

#include "am33xx.dtsi"

/ {
    model = "TI AM3359 New Product";
    compatible = "ti,am3359-new-product", "ti,am33xx";

    memory@80000000 {
        device_type = "memory";
        reg = <0x80000000 0x10000000>; /* 256 MB */
    };
    chosen {
        stdout-path = &uart0;
    };
};

&am33xx_pinmux {

    uart0_pins: pinmux_uart0_pins {
        pinctrl-single,pins = <
            AM33XX_IOPAD(0x970, PIN_INPUT_PULLUP | MUX_MODE0) /* uart0_rxd. uart0_rxd */
            AM33XX_IOPAD(0x974, PIN_OUTPUT_PULLDOWN | MUX_MODE0) /* uart0_txd. uart0_txd */
        >;
    };
};

&uart0 {

    pinctrl-names = "default";
    pinctrl-0 = <&uart0_pins>;
    status = "okay";
};
```

# DTS file example structure

- HelloWorld-like minimal board DTS file (initramfs)
- Defines the Processor used
- UART node and supporting pinmux
- Can also use EVM DTS file and disable all the nodes ... except UART.

```
/dts-v1/;  
#include "am33xx.dtsi"  
  
/ {  
    model = "TI AM3359 New Product";  
    compatible = "ti,am3359-new-product", "ti,am33xx";  
  
    memory@80000000 {  
        device_type = "memory";  
        reg = <0x80000000 0x10000000>; /* 256 MB */  
    };  
    chosen {  
        stdout-path = &uart0;  
    };  
};  
&am33xx_pinmux {  
  
    uart0_pins: pinmux_uart0_pins {  
        pinctrl-single,pins = <  
            AM33XX_IOPAD(0x970, PIN_INPUT_PULLUP | MUX_MODE0) /* uart0_rxd. uart0_rxd */  
            AM33XX_IOPAD(0x974, PIN_OUTPUT_PULLDOWN | MUX_MODE0) /* uart0_txd. uart0_txd */  
        >;  
    };  
};  
&uart0 {  
  
        pinctrl-names = "default";  
        pinctrl-0 = <&uart0_pins>;  
        status = "okay";  
};
```

# DTS file example structure

- HelloWorld-like minimal board DTS file (initramfs)
- Defines the Processor used
- UART node and supporting pinmux
- Can also use EVM DTS file and disable all the nodes ... except UART.

```
/dts-v1/;  
#include "am33xx.dtsi"  
  
/ {  
    model = "TI AM3359 New Product";  
    compatible = "ti,am3359-new-product", "ti,am33xx";  
  
    memory@80000000 {  
        device_type = "memory";  
        reg = <0x80000000 0x10000000>; /* 256 MB */  
    };  
    chosen {  
        stdout-path = &uart0;  
    };  
};  
&am33xx_pinmux {  
  
    uart0_pins: pinmux_uart0_pins {  
        pinctrl-single,pins = <  
            AM33XX_IOPAD(0x970, PIN_INPUT_PULLUP | MUX_MODE0) /* uart0_rxd.uart0_rxd */  
            AM33XX_IOPAD(0x974, PIN_OUTPUT_PULLDOWN | MUX_MODE0) /* uart0_txd.uart0_txd */  
        >;  
    };  
};  
&uart0 {  
    pinctrl-names = "default";  
    pinctrl-0 = <&uart0_pins>;  
    status = "okay";  
};
```



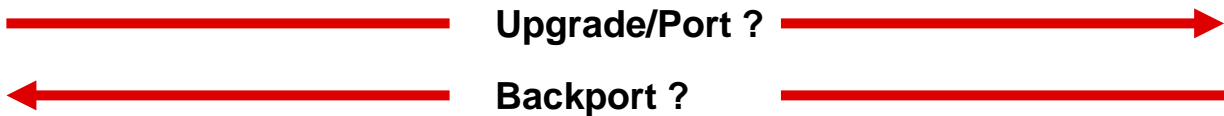
# Linux board port elements: Root filesystem

- “Everything is defined as a file in Linux.”
- The files are organized into a directory structure called the root filesystem, mostly for human consumption.
- This structure provides a natural hierarchy of abstraction between the kernel and the user applications.

```
/
bin
boot
dev
etc
home
include
lib
media
mnt
opt
proc
run
sbin
srv
sys
tmp
usr
var
www
```



# Linux board port elements: TI Linux SDK lifecycle



Current SDK Support Window



SDK release date

Initial board port

Issue detected during testing;  
Fixed in later SDK version

Production



# Summary: Elements of a Linux board port

Download  
SDK for  
EVM

Set up  
environment

Use pinmux  
for custom  
board

Create  
minimal DTB  
for new  
board

Boot SDK  
kernel with  
SDK RootFS

Iterate to add  
functionality



# Summary: Elements of a Linux board port

Download  
SDK for  
EVM

Set up  
environment

Use pinmux  
for custom  
board

Create  
minimal DTB  
for new  
board

Boot SDK  
kernel with  
SDK RootFS

PROCESSOR-SDK-LINUX-AMxx

# Summary: Elements of a Linux board port

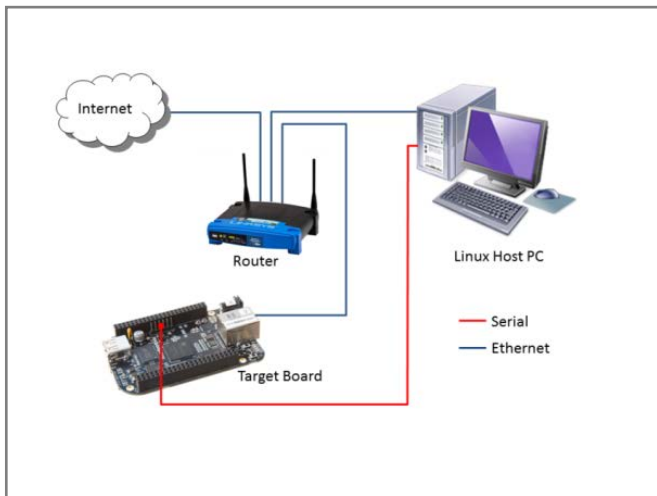
Download  
SDK for  
EVM

Set up  
environment

Use pinmux  
for custom  
board

Create  
minimal DTB  
for new  
board

Boot SDK  
kernel with  
SDK RootFS





# Summary: Elements of a Linux board port

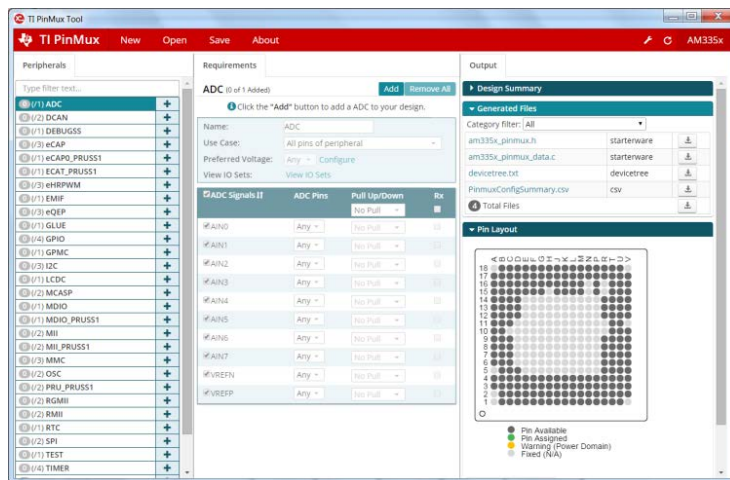
Download  
SDK for  
EVM

Set up  
environment

Use pinmux  
for custom  
board

Create  
minimal DTB  
for new  
board

Boot SDK  
kernel with  
SDK RootFS



# Summary: Elements of a Linux board port

Download  
SDK for  
EVM

Set up  
environment

Use pinmux  
for custom  
board

Create  
minimal DTB  
for new  
board

Boot SDK  
kernel with  
SDK RootFS

```
/dts-v1/;
#include "am33xx.dtsi"

/ {
    model = "TI AM3359 New Product";
    compatible = "ti,am3359-new-product", "ti,am33xx";

    memory@80000000 {
        device_type = "memory";
        reg = <0x80000000 0x10000000>; /* 256 MB */
    };
    chosen {
        stdout-path = &uart0;
    };
};

&am33xx_pinmux {

    uart0_pins: pinmux_uart0_pins {
        pinctrl-single_pins = <
            AM33XX_IOPAD(0x970, PIN_INPUT_PULLUP | MUX_MODE0) /* uart0_rxd.uart0_rxd */
            AM33XX_IOPAD(0x974, PIN_OUTPUT_PULLDOWN | MUX_MODE0) /* uart0_txd.uart0_txd */
        >;
    };

    uart0 {
        pinctrl-names = "default";
        pinctrl-0 = <uart0_pins_default>;
        status = "okay";
    };
};
```



# Summary: Elements of a Linux board port

Download  
SDK for  
EVM

Set up  
environment

Use pinmux  
for custom  
board

Create  
minimal DTB  
for new  
board

Boot SDK  
kernel with  
SDK RootFS



Linux Kernel Config  
Linux Kernel

Root  
File  
System

# Summary: Elements of a Linux board port

Download  
SDK for  
EVM

Set up  
environment

Use pinmux  
for custom  
board

Create  
minimal DTB  
for new  
board

Iterate to add  
functionality

```
/dts-v1/;
#include "am33xx.dtsi"

/ {
    model = "TI AM3359 New Product";
    compatible = "ti,am3359-new-product", "ti,am3359";

    memory@80000000 {
        device_type = "memory";
        reg = <0x80000000 0x10000000 256 MB * /
    };
    chosen {
        stdout-path = &uart0;
    };
};

&am33xx_pinmux {
    uart0_pins: pinmux_uart0_pins {
        pinctrl-single,pins = <
            AM33XX_IOPAD(0x970, PIN_INPUT_PULLUP | PUX_MODE0) /* uart0_rxd.uart0_rxd */
            AM33XX_IOPAD(0x974, PIN_OUTPUT_PULLDOWN | PUX_MODE0) /* uart0_txd.uart0_txd */
        >;
    };
};

&uart0 {
    pinctrl-names = "default";
    pinctrl-0 = <uart0_pins_default>;
    status = "okay";
};
```





# Conclusion: Elements of a Linux board port

- Development environment using TI Processor SDK Linux
- PinMux Tool
- Kernel image
- DTS file
- Root filesystem
- Product development cycle and TI SDK support window