

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

TI PinMux New	Open	Save About	i anti anti anti anti anti anti anti ant				F C	AM3
Peripherals		Requirements				Output		
Type filter text	-	ADC (0 of 1 Added)		Add Re	emove All	• Design Summary		
() (/1) ADC	+	Click the "A	dd" button to ac	id a ADC to your desis	εn.	➡ Generated Files		
(/2) DCAN	+		17			Category filter		
(/1) DEBUGSS	+	Name:	ADC			category inter. An		1
(/3) eCAP	+	Use Case:	All pins of peri	pheral	+	am335x_pinmux.h	starterware	<u>±</u>
(/1) eCAP0_PRUSS1	+	Preferred Voltage:	Any - Confi	gune		am335x_pinmux_data.c	starterware	<u>+</u>
(/1) ECAT_PRUSS1	+	View IO Sets:	View IO Sets			devicetree.txt	devicetree	Ŧ
(/3) eHRPWM	+					PinmuxConfigSummary.csv	CSV	Ŧ
(/1) EMIF	+	ADC Signals It	ADC Pins	Pull Up/Down	Rx	Total Elles		-
(/3) eQEP	+			No Pull 👻		Total Files		<u> </u>
(/1) GLUE	+	M AINO	Any ~	No Pull +	ļi.	- Pin Layout		
(/4) GPIO	+	Ø ∆INT	Anv -	No bull e	11			
(/1) GPMC	+	and an a	City				2011-22	
🕐 (/3) I2C	+	MAIN2	Any -	No Pul	53	18	00000	
(/1) LCDC	+	✓AIN3	Any -	No Pull				
(/2) MCASP	+	(FAIL)	Ante -			15 000000 0000		
(/1) MDIO	+	INAIN4	Any *	No Pull *	61	13 000000000000000000000000000000000000		
(/1) MDIO_PRUSS1	+	≮AIN5	Any -	No Puli	2.2	11 000000000000000000000000000000000000		
(/2) MII	+	MAING	Anv -	MACPUIL *	11			
(/2) MII_PRUSS1	+	12002	and the second s			800000000000000000000000000000000000000	0000	
(/3) MMC	+	MAIN7	Any +	No Pull 👻	111	600000000000000000000000000000000000000		
(/2) OSC	+	WVREFN	Any -	No Pull	11	5 000 000000000000000000000000000000000		
(/2) PRU_PRUSS1	+	ØVRFEP	Any -	I No Dull	111	300000000000000000000000000000000000000		
(/2) RGMII	+		a start a	100.110		100000000000000000000000000000000000000		
(/2) RMII	+					0		
(/1) RTC	+					Pin Available		
() (/2) SPI	+					Pin Assigned Warning (Power Dr	omain)	
(/1) TEST	+					Fixed (N/A)		
(/4) TIMER	+							



So... what is a pinmux?

IOSets for	DCAN0

Ŧ

Type filter text	Signal	Device Pin
DCAN0_IOSet_1	dcan0_tx	J18 (gmii1_txd3)
DCAN0_IOSet_2	dcan0_rx	K15 (gmii1_txd2)
DCAN0_IOSet_3		

🔱 Texas Instruments

So... what is a pinmux?

	Signal	Device Pin
	dcan0_tx	J18 (gmii1_txd3)
	dcan0_rx	K15 (gmii1_txd2)
		IO Set 1
ſ	∢поошгот	JEZGRHON
		000000000
	16	
	15 • • • • • • • • • • • • • •	
	13 ••••	
	9	
	°7	
	6 5	
	4 0000000	
	2	
	0	
C C)

IOSets for DCAN0

Type filter text... DCAN0_IOSet_1 DCAN0_IOSet_2 DCAN0_IOSet_3 -



So... what is a pinmux?

Type filter text	Signal	Device Pin
DCAN0 IOSet 1	dcan0_tx	J18 (gmii1_txd3)
DCAN0 IOSet 2	dcan0_rx	K15 (gmii1_txd2)
DCAN0_IOSet_3		
odes		
[4] TYPE [5]		IO Set 1
0		
0		
1		
VO		
VO		
VO	13	
VO	11	
	10	
	8	
	6	
	5 4 • • • • • • •	
	0	

DCAN0_IOSet_3 Datasheet snippet showing signal names and mux modes

ZCZ BALL NUMBER [1]	PIN NAME [2]	SIGN	AL NAME [3]	MODE [4]	TYPE [5]
J18	MII1_TXD3	gmii1_txd3		0	0
		dcan0_tx		1	0
	•	rgmii1_td3		2	0
		uart4_rxd		3	I.
		mcasp1_fsx		4	I/O
		mmc2_dat1		5	I/O
		mcasp0_fsr		6	I/O
		gpio0_16		7	I/O



















Linux board port elements: Linux kernel



- 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







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
- SOC LCD Controller LCDC Pads lcd_cp LCD PCLK L3Fast DMA Master Icd pixel[15:0] LCD DATA[15:0] Interface Interconnect LCD DATA[23:16] cd pixel[23:16] L4Peripheral lod lp LCD HSYNC CFG Interfac Interconnect LCD VSYNC lcd fp LCD MEMORY CLK led melk MPU Subsystem intr_pend Board Interrupts lcd int clk PRCM Disp PLL LCD CLK Icd clk CLKOUT
- 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



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



Texas Instruments

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 = <</pre>
           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 = <</pre>
           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 = <</pre>
           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





Iterate to add functionality





PROCESSOR-SDK-LINUX-AMxx







Set up

environment

Download SDK for EVM Use pinmux for custom board Create minimal DTB for new board

Boot SDK kernel with SDK RootFS

TI PinMux Tool								
😽 TI PinMux 🛛 Ne	ew Open	Save About					1	C AM335
Peripherals		Requirements				Output		
Type filter text	-	ADC (0 of 1 Added)		Add R	emove All	Design Summary		
(/1) ADC	+	Click the "A	dd" button to ac	id a ADC to your desis	in.	- Generated Eler		
(/2) DCAN	+				500			
(/1) DEBUGSS	+	Name:	ADC			Category hiter: All	1.1.0.000000000000000000000000000000000	
(/3) eCAP	+	Use Case:	All pins of peri	pheral		am335x_pinmux.h	starterware	<u>+</u>
(/1) eCAP0_PRUSS1	+	Preferred Voltage:	Any + Confi	gure .		am335x_pinmux_data.c	starterware	Ŧ
(/1) ECAT_PRUSS1	+	View IO Sets:	View ID Sets			devicetree.txt	devicetree	£
(/3) eHRPWM	+				00/00/000	PinmuxConfigSummary.csv	CSV	±
(/1) EMIF	+	CADC Signals IT	ADC Pins	Pull Up/Down	Rox	Total Files		4
(/3) eQEP	+	4		No Pull -		U Totar Pries		
(/1) GLUE	+	MAINO	Any -					
(/4) GPIO	+	PAIN1	Any -					
(/1) GPMC	+					<	20.00+2>	
(/3) I2C	+	MAIN2	Any -			18 000000000000000000000000000000000000	00000	
(/1) LCDC	+	RAIN3	Any -					
(/2) MCASP	+	(Restance)	Arrit in 1			15 0000000 0000	0 000	
(/1) MDIO	+		Party -			13 000000000000000		
(/1) MDIO_PRUSS1	+	MAINS	Any -			110000000000000000000000000000000000000	00000	
(/2) MII	+	RAINS :	Arty -			10 000000000000000000000000000000000000		
(/2) MII_PRUSS1	+	WART.				800000000000000000000000000000000000000		
(/3) MMC	+	T. Autor	wild a			600000000000000000000000000000000000000		
(/2) OSC	+	MVREFN	Any -			400000000000000000000000000000000000000		
U(2) PRU_PRUSS1	+	RVREFP	Any -			300000000000000000000000000000000000000		
(/2) RGMII	+					100000000000000000000000000000000000000	00000	
(/2) RMII	+					0		
0 (/1) RTC	+					Pin Available		
(/2) SPI	+					Pin Assigned Warning (Power Dr.)	omain)	
(/1) TEST	+					Fixed (N/A)	5-5650 # C	
(/4) TIMER	+ .							











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" 11 model = "TI AM3359 New Product"; compatible = "ti,am3359-new-product" memory@80000000 { device type = "memory"; reg = <0x80000000 0x10000000 256 MB */ 1: chosen stdout-path = &uart0; };); Gam33xx_pinmux (uart0_pins: pinmux_uart0_pins { pinctrl-single,pins = < AM33XX IOPAD (0x970, PIN INPUT PUL (03. /* uart0 rxd.uart0 rxd */ AM33XX IOPAD (0x974, PIN OUTPUT FULLDOWN MUX MODEO) /* uart0 txd.uart0 txd */ >; 1; &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

