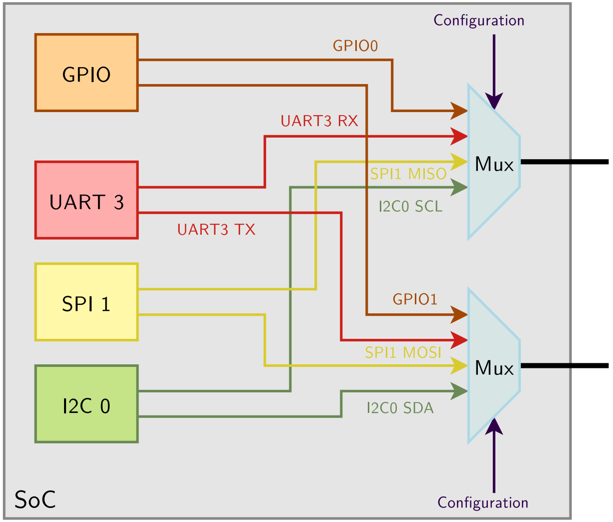
Introduction to pin muxing

What is pin muxing?

- Modern SoCs (System on Chip) include more and more hardware blocks, many of which need to interface with the outside world using *pins*.
- However, the physical size of the chips remains small, and therefore the number of available pins is limited.
- For this reason, not all of the internal hardware block features can be exposed on the pins simultaneously.
- The pins are **multiplexed**: they expose either the functionality of hardware block A **or** the functionality of hardware block B.
- This *multiplexing* is usually software configurable.

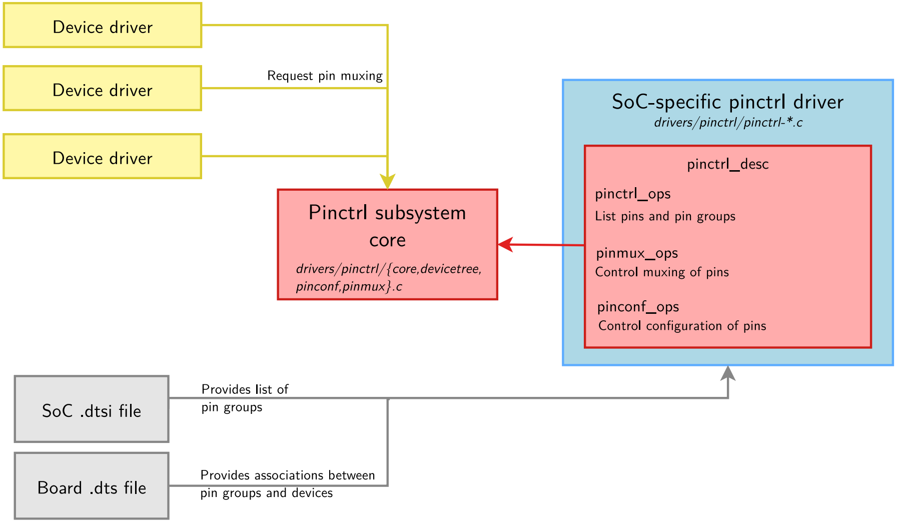
Pin muxing diagram



Pin muxing in the Linux kernel

- Since Linux 3.2, a `pinctrl` subsystem has been added.
- This subsystem, located in `drivers/pinctrl/` provides a generic subsystem to handle pin muxing. It offers:
  - A pin muxing driver interface, to implement the system-on-chip specific drivers that configure the muxing.
  - A pin muxing consumer interface, for device drivers.
- Most *pinctrl* drivers provide a Device Tree binding, and the pin muxing must be described in the Device Tree.
  - The exact Device Tree binding depends on each driver. Each binding is defined in `Documentation/devicetree/bindings/pinctrl`.

pinctrl subsystem diagram


Device driver
Device driver
Device driver
Request pin muxing
Pinctrl subsystem core
drivers/pinctrl/{core,devicetree, pinconf,pinmux}.c
SoC-specific pinctrl driver
drivers/pinctrl/pinctrl-*.c
pinctrl_desc
pinctrl_ops
List pins and pin groups
pinmux_ops
Control muxing of pins
pinconf_ops
Control configuration of pins
SoC .dtsi file
Provides list of pin groups
Board .dts file
Provides associations between pin groups and devices

Device Tree properties for consumer devices

The devices that require certains pins to be muxed will use the `pinctrl-<x>` and `pinctrl-names` Device Tree properties.

- The `pinctrl-0`, `pinctrl-1`, `pinctrl-<x>` properties link to a pin configuration for a given state of the device.
- The `pinctrl-names` property associates a name to each state. The name `default` is special, and is automatically selected by a device driver, without having to make an explicit *pinctrl* function call.
- See `Documentation/devicetree/bindings/pinctrl/pinctrl-bindings.txt` for details.

Device Tree properties for consumer devices - Examples

```
i2c0: i2c@11000 {
        ...
        pinctrl-0 = <&pmx_twsi0>;
        pinctrl-names = "default";
        ...
};
```

Most common case (arch/arm/boot/dts/kirkwood.dtsi)

```
i2c0: i2c@f8014000 {
        ...
        pinctrl-names = "default", "gpio";
        pinctrl-0 = <&pinctrl_i2c0>;
        pinctrl-1 = <&pinctrl_i2c0_gpio>;
        ...
};
```

Case with multiple pin states (arch/arm/boot/dts/sama5d4.dtsi)

Defining pinctrl configurations

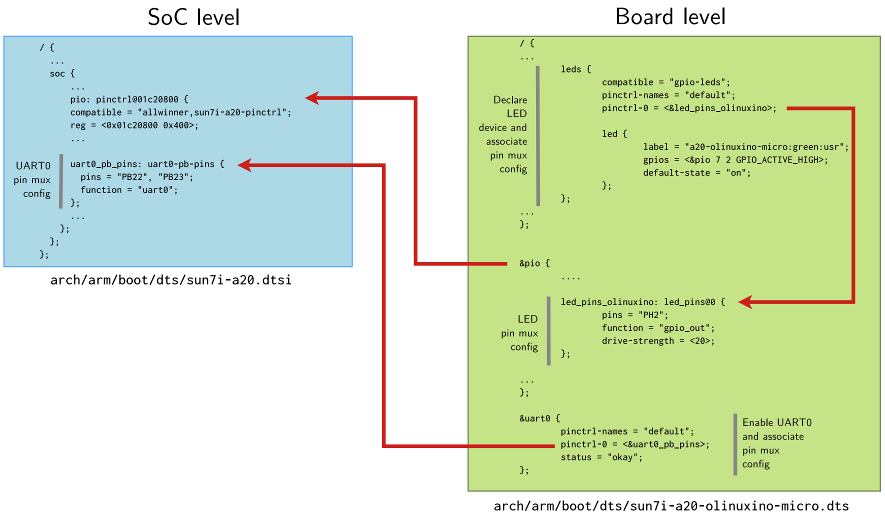
- The different *pinctrl configurations* must be defined as child nodes of the main *pinctrl device* (which controls the muxing of pins).
- The configurations may be defined at:
  - the SoC level (`.dtsi` file), for pin configurations that are often shared between multiple boards
  - at the board level (`.dts` file) for configurations that are board specific.
- The `pinctrl-<x>` property of the consumer device points to the pin configuration it needs through a DT *phandle*.
- The description of the configurations is specific to each *pinctrl driver*. See `Documentation/devicetree/bindings/pinctrl` for the pinctrl bindings.

Example on OMAP/AM33xx

- On OMAP/AM33xx, the `pinctrl-single` driver is used. It is common between multiple SoCs and simply allows to configure pins by writing a value to a register.
  - In each pin configuration, a `pinctrl-single,pins` value gives a list of *(register, value)* pairs needed to configure the pins.
- To know the correct values, one must use the SoC and board datasheets.

```
/* Excerpt from am335x-boneblue.dts */

&am33xx_pinmux {
   ...
   i2c2_pins: pinmux_i2c2_pins {
      pinctrl-single,pins = <
         AM33XX_IOPAD(0x978, PIN_INPUT_PULLUP | MUX_MODE3)
         /* (D18) uart1_ctsn.I2C2_SDA */
         AM33XX_IOPAD(0x97c, PIN_INPUT_PULLUP | MUX_MODE3)
         /* (D17) uart1_rtsn.I2C2_SCL */
      >;
   };
};

&i2c2 {
   pinctrl-names = "default";
   pinctrl-0 = <&i2c2_pins>;

   status = "okay";
   clock-frequency = <400000>;
   ...

   pressure@76 {
      compatible = "bosch,bmp280";
      reg = <0x76>;
   };
};
```

Example on the Allwinner A20 SoC


SoC level
UART0 pin mux config
Board level
Declare LED device and associate pin mux config
LED pin mux config
Enable UART0 and associate pin mux config

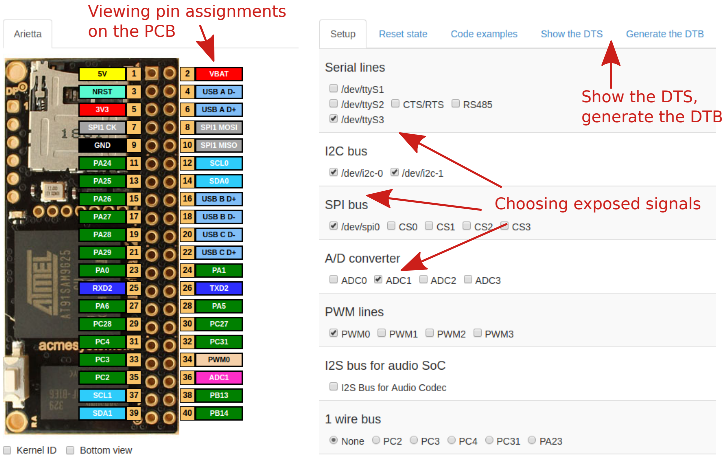
```
/ {
  ...
  soc {
      ...
      pio: pinctrl@01c20800 {
      compatible = "allwinner,sun7i-a20-pinctrl";
      reg = <0x01c20800 0x400>;
      ...

      uart0_pb_pins: uart0-pb-pins {
        pins = "PB22", "PB23";
        function = "uart0";
      };
      ...
    };
  };
};
```

arch/arm/boot/dts/sun7i-a20.dtsi

```
/ {
  ...
        leds {
                compatible = "gpio-leds";
                pinctrl-names = "default";
                pinctrl-0 = <&led_pins_olinuxino>;

                led {
                        label = "a20-olinuxino-micro:green:usr";
                        gpios = <&pio 7 2 GPIO_ACTIVE_HIGH>;
                        default-state = "on";
                };
        };
  ...
};


&pio {
        ....

        led_pins_olinuxino: led_pins@0 {
                pins = "PH2";
                function = "gpio_out";
                drive-strength = <20>;
        };

  ...
};

&uart0 {
        pinctrl-names = "default";
        pinctrl-0 = <&uart0_pb_pins>;
        status = "okay";
};
```

arch/arm/boot/dts/sun7i-a20-olinuxino-micro.dts

Illustration: live pin muxing configuration



5V	1	2	VBAT
NRST	3	4	USB A D-
3V3	5	6	USB A D+
SPI1 CK	7	8	SPI1 MOSI
GND	9	10	SPI1 MISO
PA24	11	12	SCL0
PA25	13	14	SDA0
PA26	15	16	USB B D+
PA27	17	18	USB B D-
PA28	19	20	USB C D-
PA29	21	22	USB C D+
PA0	23	24	PA1
RXD2	25	26	TXD2
PA6	27	28	PA5
PC28	29	30	PC27
PC4	31	32	PC31
PC3	33	34	PWM0
PC2	35	36	ADC1
SCL1	37	38	PB13
SDA1	39	40	PB14

☐ Kernel ID ☐ Bottom view

Serial lines
- ☐ /dev/ttyS1
- ☐ /dev/ttyS2 ☐ CTS/RTS ☐ RS485
- ☑ /dev/ttyS3

I2C bus
- ☑ /dev/i2c-0 ☑ /dev/i2c-1

SPI bus
- ☑ /dev/spi0 ☐ CS0 ☐ CS1 ☐ CS2 ☐ CS3

A/D converter
- ☐ ADC0 ☑ ADC1 ☐ ADC2 ☐ ADC3

PWM lines
- ☑ PWM0 ☐ PWM1 ☐ PWM2 ☐ PWM3

I2S bus for audio SoC
- ☐ I2S Bus for Audio Codec

1 wire bus
- ● None ○ PC2 ○ PC3 ○ PC4 ○ PC31 ○ PA23

Try ACME Systems' on-line pin-out generator: http://linux.tanzilli.com/