**fdx-MYiR: DTS Consult**

One of the things we need to do is to redefine the functionality of some GPIO’s. As we aren’t using a display for example, we can use the multipurpose GPIO’s assigned to it for other features of our product. We are thinking of using overlays to do this.

After diving into the documentation (MYiR User Manual – System development guide, System-evaluation guide, T5 series datasheet, T5 series user manual, etc) devicetree files and kernel documentation, we’ve found some properties that we can’t figure out how to use:

* In *(MYiR - VMware virtualized dev. environment) :* ***myd-t507h/t507h-sdk/t507/kernel/linux-4.9/Documentation/devicetree/bindings/pinctrl/allwinner,sunxi-pinctrl.txt***

The pins, function, drive and pull properties are described. But we can’t find info on how to use the *allwinner,function* one. In the examples and devicetrees found it seems to be simply a string label. The file says:

“ *- allwinner,function: Function to mux the pins listed above to.* ”

**[1]** - Are they just labels defined by the consumer? Do they have some use or functionality to configure the pins?

Let’s say I want to redefine this pin:

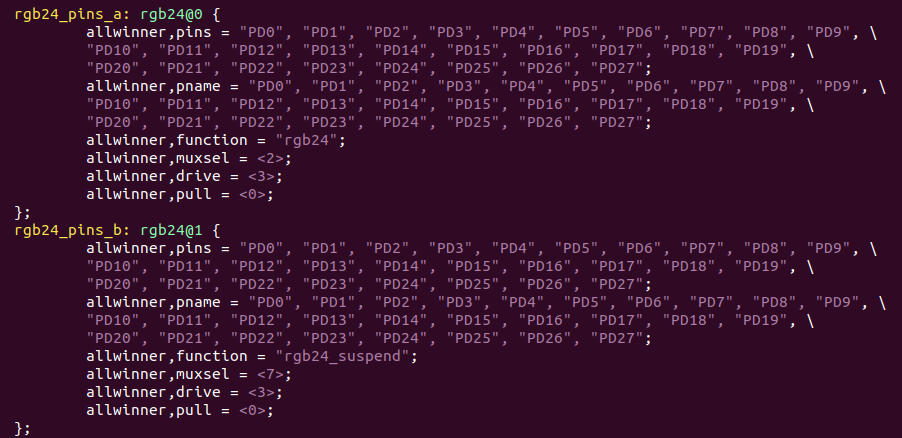


The datasheet says that function 0 and 1 are for input and output, respectively.

**[2]** - How can we assign the function property?

* In *(MYiR - VMware virtualized dev. environment) :* ***myd-t507h/t507h-sdk/t507/kernel/linux-4.9/arch/arm64/boot/dts/sunxi/sun50iw9p1-pinctrl.dtsi***

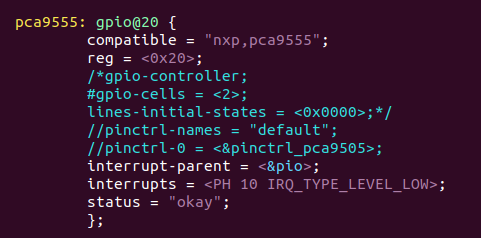
*We can find child nodes like this one:*



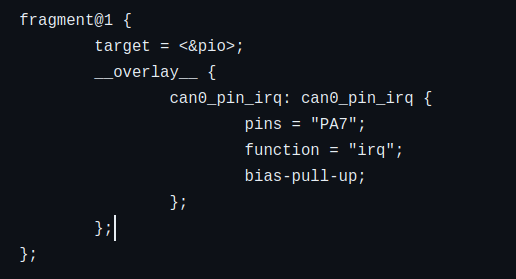
**[3]** - We weren’t able to find anything about the allwinner,muxsel property. It seems logical for this to have a 1:1 relation with the multiplex function described in the T5 datasheet, but it seems to contradict with the allwinner,function property

Could you point us in the right direction on where to look for this kind of information?

* In *(MYiR - VMware virtualized dev. environment) :*  ***myd-t507h/t507h-sdk/t507/device/config/chips/myir/configs/full***

******

We need to define a couple of interrupt-based GPIO’s, and this was the closest example of use that we could find. But it’s from the GPIO expander, and it doesn’t use directly mapped GPIO’s as we intend to. We were able to find an open source example:



But we’re not sure something like this would work since it’s compatible with a different architecture, and we don’t know how to use the function property.

**[4]** - Would something like this work?

