

MuntsOS Embedded Linux

Application Note #12: pinctrl

Revision 1
26 January 2024

by Philip Munts
dba Munts Technologies
<http://tech.munts.com>

Introduction

Traditionally [Raspberry Pi](#) boards have managed GPIO pin alternate function multiplexing with the main device tree and device tree overlays. For example, the device tree overlay `pwm.dtbo` both enables the PWM controller subsystem `pwmchip0` and configures `GPIO18` as a PWM output.

In contrast, the [BeagleBone Black](#) and its kin enable *all* peripheral subsystems with a single device tree overlay, `BB-GPIO.dtbo`, without configuring *any* GPIO pin alternate functions. In order to use a PWM channel, you have to configure one of the GPIO pins to be its output, using a command named `config-pin`.

On a BeagleBone board, the **MuntsOS Embedded Linux** startup script `/etc/rc` feeds a GPIO pin multiplexing configuration text file `/etc/pinmux.conf` to `config-pin` at system boot time. This **BeagleBone Style** of GPIO pin configuration management is often easier to understand and more flexible than the **Traditional Raspberry Pi Style**.

In late 2023 a new command named `pinctrl` was added to **Raspberry Pi OS**. It functions very similarly to `config-pin` albeit with a somewhat different syntax. **MuntsOS Embedded Linux** has imported `pinctrl` from **Raspberry Pi OS** and `/etc/rc` has been modified to feed `/etc/pinmux.conf` to `pinctrl` at system boot time, exactly as it happens on a BeagleBone board. All of this now makes it possible to do **BeagleBone Style** GPIO pin management on a 64-bit Raspberry Pi running **MuntsOS Embedded Linux**.

Two things are required to begin **BeagleBone Style** GPIO pin management. First you must install a default `pinmux.conf` using the `sysconfig` command. After that you can edit `/etc/pinmux.conf` as desired. Remember to run `sysconfig --save` after editing any system configuration file to save the configuration to the boot medium.

After stripping comments and whitespace, each line in `/etc/pinmux.conf` is appended to `pinctrl set` and the result passed to `/bin/sh` for execution. You can run `pinctrl help` to see its command syntax.

The second requirement is for device tree overlays that do not configure GPIO pins. There does not (yet) exist an equivalent to `BB-GPIO.dtbo` that enables all of the peripheral subsystems without configuring any GPIO pins. **MuntsOS Embedded Linux** does now include some selected device tree overlays for this purpose:

<code>pwmchip0.dtbo</code>	enables the PWM controller subsystem.
<code>Pi3ClickShield.dtbo</code>	enables the PWM controller subsystem.
<code>MUNTS-0018.dtbo</code>	enables the PWM controller subsystem.

More such overlays will be added in the future.