THE RASPBERRY PI UARTS

The SoCs used on the Raspberry Pis have two built-in UARTs, a PL011 and a mini UART. They are implemented using different hardware blocks, so they have slightly different characteristics. However, both are 3.3V devices, which means extra care must be taken when connecting up to an RS232 or other system that utilises different voltage levels. An adapter must be used to convert the voltage levels between the two protocols. Alternatively, 3.3V USB UART adapters can be purchased for very low prices.

By default, on Raspberry Pis equipped with the wireless/Bluetooth module (Raspberry Pi 3 and Raspberry Pi Zero W), the PL011 UART is connected to the BT module, while the mini UART is used for Linux console output. On all other models the PL011 is used for the Linux console output.

In Linux device terms, by default, /dev/ttyS0 refers to the mini UART, and /dev/ttyAMA0 refers to the PL011. The primary UART is that assigned to the Linux console, which depends on the Raspberry Pi model as described above, and can be accessed via /dev/serial0.

MINI UART AND CPU CORE FREQUENCY

The baud rate of the mini UART is linked to the core frequency of the VPU on the VC4 GPU. This means that as the VPU frequency governor varies the core frequency, the baud rate of the UART also changes. This makes the UART of limited use in the default state. Also, when the Linux console uses the mini UART (Raspberry Pi 3, Raspberry Pi Zero W), as a consequence of the UART being disabled, the console is also disabled.

The Linux console can be re-enabled by adding `enable_uart=1` to config.txt. This also fixes the core_freq to 250Mhz (unless force_turbo is set, when it will fixed to 400Mhz), which means that the UART baud rate stays consistent.

The default value of the `enable_uart` flag depends on the actual roles of the UARTs, so that if ttyAMA0 is assigned to the BT module, `enable_uart` defaults to 0. If the mini UART is assigned to the BT module, then `enable_uart` defaults to 1. Note that if the UARTs are reassigned using a Device Tree Overlay (see below), `enable_uart` defaults will still obey this rule.

DISABLING LINUX'S USE OF CONSOLE UART

In a default install of Raspbian, the primary UART (serial0) is assigned to the Linux console. Using the serial port for other purposes requires this default behaviour to be changed. On startup, `systemd` checks the Linux kernel command line for any console entries, and will use the console defined therein. To stop this behaviour, the serial console setting needs to be removed from command line.

This can be done by using the raspi-config utility, or manually.

Select option 5, **Interfacing options**, then option P6, **Serial**, and select **No**. Exit raspi-config.

To manually change the settings, edit the kernel command line with `sudo nano /boot/cmdline.txt`. Find the console entry that refers to the serial0 device, and remove it, including the baud rate setting. It will look something like `console=serial0,115200`. Make sure the rest of the line remains the same, as errors in this configuration can stop the Raspberry Pi from booting.

Reboot the Raspberry Pi for the change to take effect.

UART OUTPUT ON GPIO PINS

By default, the UART transmit and receive pins are on GPIO 14 and GPIO 15 respectively, which are pins 8 and 10 on the GPIO header.

UARTS AND DEVICE TREE

Various UART Device Tree Overlay definitions can be found in the kernel github tree. The two most useful overlays are `pi3-disable-bt` and `pi3-miniuart-bt`.

`pi3-disable-bt` disables the Bluetooth device and restores UART0/ttyAMA0 to GPIOs 14 and 15. It is also necessary to disable the system service that initialises the modem so it doesn't use the UART: `sudo systemctl disable hciuart`.

`pi3-miniuart-bt` switches the Raspberry Pi 3 and Raspberry Pi Zero W Bluetooth function to use the mini UART (ttyS0), and restores UART0/ttyAMA0 to GPIOs 14 and 15. Note that this may reduce the maximum usable baudrate (see mini UART limitations below). It is also necessary to edit /lib/systemd/system/hciuart.service and replace ttyAMA0 with ttyS0, unless you have a system with udev rules that create /dev/serial0 and /dev/serial1. In this case, use /dev/serial1 instead because it will always be correct. If cmdline.txt uses the alias serial0 to refer to the user-accessible port, the firmware will replace it with the appropriate port whether or not this overlay is used.

There are other UART-specific overlays in the folder. Refer to `/boot/overlays/README` for details on Device Tree Overlays, or run `dtoverlay -h overlay-name` for descriptions and usage information.

For full instructions on how to use Device Tree Overlays see this page. In brief, add a line to the `config.txt` file to enable Device Tree Overlays. Note that the `-overlay.dts` part of the filename is removed.

```
...
dtoverlay=pi3-disable-bt
...
```

RELEVANT DIFFERENCES BETWEEN PL011 AND MINI UART

The mini UART has smaller FIFOs. Combined with the lack of flow control, this makes it more prone to losing characters at higher baudrates. It is also generally less capable than the PL011, mainly due to its baud rate link to the VPU clock speed.

The particular deficiencies of the mini UART compared to the PL011 are :

- No break detection
- No framing errors detection

- No DCD, DSR, DTR or RI signals

Further documentation on the mini UART can be found in the SoC peripherals document here.