

## Serial Communication in Raspberry Pi 3

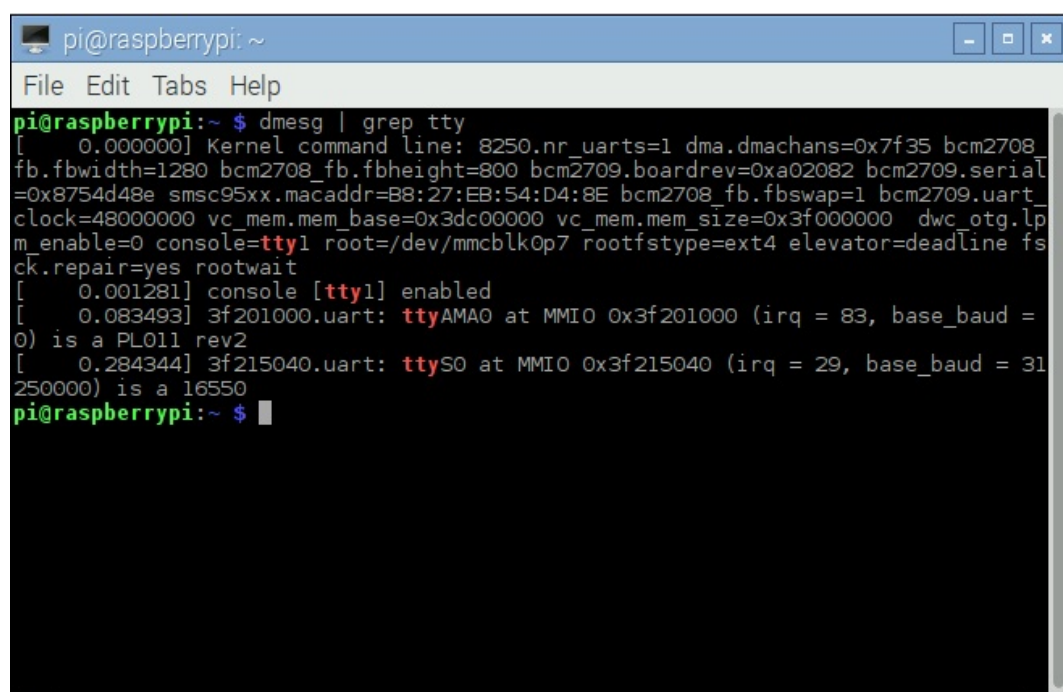
We had an exciting first week of March this year, yeah playing with the all new Raspberry Pi 3. We had the Pi 3, and we were selling it to all Pi enthusiasts out there. And one day, we got a call. A frustrated customer, he was unable to use the UART of his Pi 3. Thats right! Without a proper work around, UART on the GPIO header wont function well.

The previous versions of Raspberry Pi had UART0 brought out on the GPIO header. But the design was changed with Raspberry Pi 3, which features UART1 on the GPIO header while UART0 is dedicated for Bluetooth. And therefore, the UART on GPIO header will be available at /dev/ttyS0 instead of /dev/AMA0. To meet with with change, we will have to configure the OS accordingly.

Open the LX teminal and type:

***dmesg | grep tty***

This command will list the available UART modules, as shown in the figure:



```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi:~$ dmesg | grep tty  
[ 0.000000] Kernel command line: 8250.nr_uarts=1 dma.dmachans=0x7f35 bcm2708_fb.fbwidth=1280 bcm2708_fb.fbheight=800 bcm2709.boardrev=0xa02082 bcm2709.serial=0x8754d48e smsc95xx.macaddr=B8:27:EB:54:D4:8E bcm2708_fb.fbswap=1 bcm2709.uart_clock=48000000 vc_mem.mem_base=0x3dc00000 vc_mem.mem_size=0x3f000000 dwc_otg.lpm_enable=0 console=tty1 root=/dev/mmcblk0p7 rootfstype=ext4 elevator=deadline fsck.repair=yes rootwait  
[ 0.001281] console [tty1] enabled  
[ 0.083493] 3f201000.uart: ttyAMA0 at MMIO 0x3f201000 (irq = 83, base_baud = 0) is a PL011 rev2  
[ 0.284344] 3f215040.uart: ttyS0 at MMIO 0x3f215040 (irq = 29, base_baud = 31250000) is a 16550  
pi@raspberrypi:~$
```

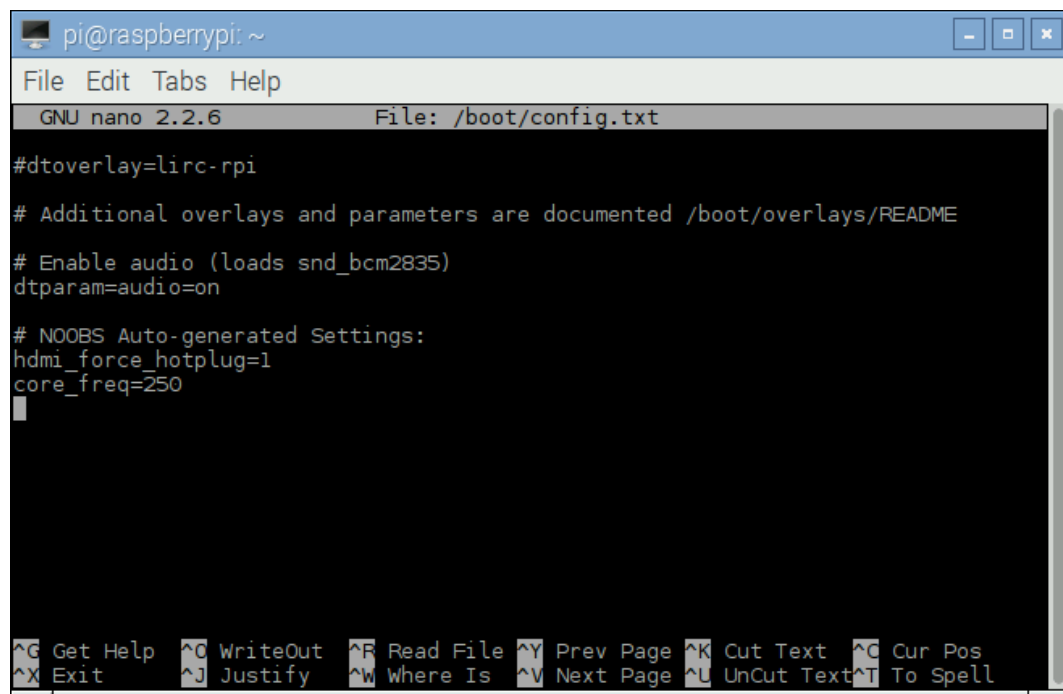
The list will show, ttyS0 – UART corresponding to the GPIO header.

To operate the GPIO UART at the correct baud rate, enter ***sudo nano /boot/config.txt*** in the terminal and add the two lines

**`core_freq=250`**

**`enable_uart=1`**

Save the file and reboot the Pi.



```
pi@raspberrypi: ~
File Edit Tabs Help
GNU nano 2.2.6 File: /boot/config.txt

#dtoverlay=lirc-rpi

# Additional overlays and parameters are documented /boot/overlays/README

# Enable audio (loads snd_bcm2835)
dtparam=audio=on

# NOOBS Auto-generated Settings:
hdmi_force_hotplug=1
core_freq=250

```

The GPIO UART now operates at the correct baud rate, and is

available at `/dev/ttyS0`, and NOT `/dev/ttyAMA0` like before.

So while writing the code in python has to be changed accordingly:

**`port = serial.Serial("/dev/ttyS0", baudrate=9600, timeout=1)`**

So what makes the difference here? How is anything different with UART1 of BCM2835/6. This is how it is:

- Unlike UART0, UART1 is dependent on Core Clock Frequency
- Unlike UART0, UART1 has smaller FIFOs

When the core clock changes, the baud rate will also change!