

Purpose Of Code:

- This code serves the purpose of interfacing GPS(Global Positioning Module) serially with the Raspberry pi 2.
- This code is meant to show the uses of internal structures of serial Drivers present in the kernel.
- The main purpose of device drivers is to provide abstraction by acting as a translator between a hardware device and the applications or operating systems that use it.

How to use it:

This is the kernel driver to interface gps module serially with raspberry pi.

Steps to be performed to interface GPS module.

1) Download linux for raspberry pi from the link <https://github.com/raspberrypi/linux>. Store it in home directory

2) Run the following commands to build the kernel

(i) `cd linux`

(ii) `KERNEL=kernel7`

(iii) `make bcm2709_defconfig`

(iv) Download all the codes from our repository.

(v) Remove the existing driver from `/linux/drivers/tty/serial/pl011.c` because of resource conflict.

(vi) `make -j4 zImage modules dtbs` :- This step takes time.

(vii) `sudo make modules_install`

(viii) `sudo cp arch/arm/boot/dts/*.dtb /boot/`

(ix) `sudo cp arch/arm/boot/dts/overlays/*.dtb* /boot/overlays/`

(x) `sudo cp arch/arm/boot/dts/overlays/README /boot/overlays/`

(xi) `sudo cp arch/arm/boot/zImage /boot/$KERNEL.img`

(xii) reboot raspberry pi

3) The above Step build the kernel it creates the node named `BITS_PILANI0` under `/dev` directory.

4) To run gps module follow these steps.

(i) keep the GPS on open sky

(ii) `sudo cat /dev/BITS_PILANI0` : it will produce value of gps in unformatted pattern. The user will not understand this NMEA protocol.

(iii) install GPSD by using the command "`sudo apt-get install gpsd`".

(iv) `sudo nano /etc/default/gpsd.config`.

->Edit the file, so that it points to the node as device: `'/dev/BITS_PILANI0'` and save it

(v) run 'cgps -s' command to see the gps location.

For more information refer to reference manual.

Possible Errors:

Resource Conflict: This error occurs while inserting the serial driver module as linux has inbuilt serial driver which already uses the desired hardware. Remedy for this problem is that we have to compile the complete kernel as explained above. During compilation remove the existing serial driver(SERIAL0).