

(THIS PROCEDURE IS FOR RASPBERRY Pi1 ONLY)

Step for kernel compilation:

```
1> enable CONFIG_KALLSYMS and CONFIG_DEBUG_INFO in your .config file.
2> set variable KERNEL=kernel
3> make ARCH=arm CROSS_COMPILE=/home/mehul/rpi/tools/arm-bcm2708/arm-bcm2708-
linux-gnueabi/bin/arm-bcm2708-linux-gnueabi- bcmrpi_defconfig
4> make ARCH=arm CROSS_COMPILE=/home/mehul/rpi/tools/arm-bcm2708/arm-bcm2708-
linux-gnueabi/bin/arm-bcm2708-linux-gnueabi- zImage modules dtbs
```

Step for U-boot compilation:

*(These commands only if you want to do some changes in U-boot , otherwise placing kernel related images in SDcard will work ,as raspbian has its own bootloader)

```
1> make ARCH=arm CROSS_COMPILE=/home/mehul/rpi/tools/arm-bcm2708/arm-bcm2708-
linux-gnueabi/bin/arm-bcm2708-linux-gnueabi- rpi_defconfig
2> make ARCH=arm CROSS_COMPILE=/home/mehul/rpi/tools/arm-bcm2708/arm-bcm2708-
linux-gnueabi/bin/arm-bcm2708-linux-gnueabi-
```

Putting image into SD card :

```
1> First format the SD card.
2> Then execute command sudo dd if=2017-06-21-raspbian-jessie.img of=/dev/sdd bs=4M .
3> insert the card in the board and boot it once , so that it will create partition.
4> now to get the log on minicom write "enable_uart=1" at the end of /boot/config.txt as mentioned
in https://elinux.org/RPi\_Serial\_Connection.
(in board RPi1-1551 Tx/Rx pin is not working , so please take board RPi1-1550)
```

go to kernel dirrctory where you compile the code and execute below command(below commands are given in raspberrry site also.)

```
mkdir mnt
mkdir mnt/fat32
mkdir mnt/ext4
sudo mount /dev/sdb1 mnt/fat32
sudo mount /dev/sdb2 mnt/ext4
```

```
sudo make ARCH=arm INSTALL_MOD_PATH=mnt/ext4 modules_install
```

```
sudo cp mnt/fat32/$KERNEL.img mnt/fat32/$KERNEL-backup.img
sudo cp arch/arm/boot/zImage mnt/fat32/$KERNEL.img
sudo cp arch/arm/boot/dts/*.dtb mnt/fat32/
sudo cp arch/arm/boot/dts/overlays/*.dtb* mnt/fat32/overlays/
sudo cp arch/arm/boot/dts/overlays/README mnt/fat32/overlays/
sudo umount mnt/fat32
sudo umount mnt/ext4
```

JTAG pin configuration program:

download JtagEnabler.cpp for Rpi1 from internet and crosscompile it and generate a.out and put executable(/a.out)in the SD card partition.

Enabling JTAG pins :

After SD card flashed , put it in the board's SD card slot. Then After when kernel boot and you get the prompt, go to the directory where ./a.out was kept and execute it.

username and password to login into board :

username : pi

password : raspberry

Trace32 :

After making connection as shown in the wiki image , launch trace32 and execute below command in t32 prompt.

SYSTEM.CPU ARM1176JZF

and then as shown in attached image tronchip.png , untick all the exceptions listed under "set" tab

.Then execute below command

SYSTEM.CPUACCESS Enable

SYSTEM.MODE ATTACH

DATA.LOAD.ELF c:\Users\Administrator\Documents\new\linux-rpi-3.10.y\vmlinux /NOCODE

Y.SPATH.SRD c:\Users\Administrator\Documents\new\linux-rpi-3.10.y\

Now , you are ready to set the breakpoint and debug ,but this will let you to debug only kernel code in which you have place

breakpoint and if application call that function but you will not able to debug the code which is executing in the start of the kernel

because we are enabling JTAG pins after kernel boots ; so to debug at kernel booting level ,

implement the JtagEnable.cpp logic

before kernel boots at uboot level.

USEFUL LINKS:

LINKS :

https://wiki.gloaledgesoft.com/index.php/JTAG_Debugging_on_Raspberry_Pi#Preparing_Raspberry_PI_for_JTAG_.26_README

<https://www.raspberrypi.org/documentation/linux/kernel/building.md>

<http://sysprogs.com/VisualKernel/tutorials/raspberry/jtagsetup/>

<https://github.com/raspberrypi/linux/tree/rpi-3.10.y>

https://elinux.org/RPi_U-Boot

<https://github.com/gonzoua/u-boot-pi>

https://wiki.gloaledgesoft.com/index.php/Raspberry_Pi

<https://devsidestory.com/build-a-64-bit-kernel-for-your-raspberry-pi-3/>