



blaster.notes

back out rpi-update changes

```
sudo apt-get install --reinstall raspberrypi-bootloader raspberrypi-kernel
```

tweak some boot items

Speed limits - core/arm_freq/arm_freq_min - this goes in /boot/config.txt

- Pi3 - 400/1200/600
- Pi2 - 250/900/600
- PiZero - 250/1000/700

```
# edit to underclock pi3 to pi2 speeds
# core_freq=250
# arm_freq=900
# arm_freq_min=600
# end edit
```

temps/clocks

```
#!/bin/bash

# pistuff.sh looks at firmware/kernel, and clocks, temps for gpu/cpu
# create this file in the user dir, and make it executable

echo "$(date) @ $(hostname)"
echo "SW Rev => $(uname -vr)"
echo "FW Rev => $(vcgencmd version)"
echo "======"
echo "ARM Mem => $(vcgencmd get_mem arm)"
echo "GPU Mem => $(vcgencmd get_mem gpu)"
echo "======"
echo "Uptime =>$(uptime)"
echo "Pi Temp => $(vcgencmd measure_temp)"
echo "Pi Volts => $(vcgencmd measure_volts core)"
echo "Pi Clock => $(vcgencmd measure_clock arm)"
```

Enable Zram

New method - works across all Pi's

This script dynamically enables ZRAM on a Raspberry Pi or other Linux system.

Automatically detects the number of CPU cores to allocate to ZRAM computation, disables existing swap and enables ZRAM swap.

create the script (zram.sh) and copy to /usr/bin/ folder

make file executable

```
sudo chmod +x /usr/bin/zram.sh
```

edit /etc/rc.local file to run script on boot

```
sudo nano /etc/rc.local
```

add line before exit 0

```
/usr/bin/zram.sh &
```

script below

```
#!/bin/bash
cores=$(nproc --all)
modprobe zram num_devices=$cores
```

```
swapoff -a
```

```
totalmem=`free | grep -e "^Mem:" | awk '{print $2}'`  
mem=$(( ($totalmem / $cores) * 1024 ))
```

```
core=0  
while [ $core -lt $cores ]; do  
    echo $mem > /sys/block/zram$core/disksize  
    mkswap /dev/zram$core  
    swapon -p 5 /dev/zram$core  
    let core=core+1  
done
```

Reference - https://github.com/novaspirt/rpi_zram [https://github.com/novaspirt/rpi_zram]

Deprecated - Old Stuff

```
#!/bin/bash
```

```
# Raspberry Pi ZRAM script  
# Tuned for quad core, 1 GB RAM models  
# put me in /etc/init.d/zram.sh and make me executable  
# then run "sudo update-rc.d zram.sh defaults"
```

```
modprobe zram  
echo 3 > /sys/devices/virtual/block/zram0/max_comp_streams  
echo lz4 > /sys/devices/virtual/block/zram0/comp_algorithm  
echo 268435456 > /sys/devices/virtual/block/zram0/mem_limit  
echo 536870912 > /sys/devices/virtual/block/zram0/disksize  
mkswap /dev/zram0  
swapon -p 0 /dev/zram0  
sysctl vm.swappiness=10
```

And then enable it...

```
/etc/init.d/zram start|stop
```

SSH - rebuild keys

Raspbian sets the same pi user and ssh keys as part of the raspbian image - wise to reset all keys...

```
# sudo rm /etc/ssh/ssh_host_*  
# sudo dpkg-reconfigure openssh-server
```

Raspbian Jessie - Add new privileged user

As part of hardening the RPi - we add a new user with admin rights, deprecating the "pi" user (just set a long/robust password for pi, use the new account instead for daily stuff)

For SSH - review the SSH section - for allowusers on sshd, ensure that pi is not an allowed user to reduce the threat from brute force attacks

Add a new user, and give it the right perms/access

Below, we add "testuser" as the replacement user - once that userid is created, we need to include the same groups as the "pi" user...

```
# sudo useradd testuser -s /bin/bash -m -G adm,sudo,dialout,cdrom,audio,video,plugdev,games,users,input,netdev,gpio,i2c,spi  
# sudo passwd testuser  
Enter new UNIX password:  
Retype new UNIX password:  
passwd: password updated successfully
```

edit sudoers

With Raspbian - we need to add/modify sudoers file to ensure that the Pixel GUI still works for BT/WiFi config, along with the Raspi-Config GUI - this is a specific item for Raspbian-Jessie and the Pixel UI

NOTE - I'd like to not do this, as we still want to honor the rule of least privilege, and having to enter the sudo password helps out much, but as Raspbian is now, the GUI breaks if we don't do this...

edit the /etc/sudoers file - in that file, look for the line that says "ALL=(ALL:ALL) ALL" and comment and add new line like below;

```
# Allow members of group sudo to execute any command  
# %sudo    ALL=(ALL:ALL) ALL
```

```
%sudo    ALL=(ALL) NOPASSWD: ALL
```

Remove the pi user

This is not recommended as it will break things in the GUI config - better to leave pi user in place, and use a robust password - see pwgen

But if one must...

```
# sudo userdel pi
# sudo rm -rf /home/pi
```

snb_basics_-_pistuff.txt · Last modified: 2018/04/06 20:10 by admin