[LEGACY] How do I setup usbboot (no SD cards in the Pi Zeros) for the Cluster HAT?

See the current documentation on booting without SD cards, this legacy information is kept here in case it helps anyone.

I have made an image available to boot the Pi Zeros using usbboot/rpiboot - see https://8086.support//content/23/88/en/guide-to-using-the-rpiboot-test-image-on-the-cluster-hat_zero-stem-or-just-a-usb-cable.html **so most of this guide is no longer needed if you want to try it yourself.**

The following is a ***first attempt rough guide*** to setting up the Cluster HAT + Pi Zeros to boot using usbboot meaning no SD card is needed in the Pi Zeros in the Cluster HAT.

It uses about 5.7GB of disk space for 4 NFS roots (~800MB each root) but I'd advise using at least a 16GB SD card to allow some room for the filesystems to grow.

There are issues with this setup :(- see the bottom of this guide for a list.

The USB path (1-1.2.4) assumes you're connecting the Cluster HAT using the top left USB connector (looking at the end with the connectors) if this isn't the case your path will be different.

How does it work?

After powering on the Pi Zeros (without SD cards) they boot up and show up as a device.
This is then picked up by a modified rpiboot (see "Modifications to rpiboot" below) which sends over the required files for the /boot directory.
The root filesystem is then mounted using NFS from the controller over the Internal network.
On the Pi Zero the network device usb0.10 (VLAN 10) is bridged to eth0 on the controller and obtains an IP address using the DHCP server on the network (see "Network Configuration" below)

Modifications to rpiboot

I have modified rpiboot to support "overlay" files based on the USB device path, this allows individual config.txt/cmdline.txt/etc. files to be sent to specific Pi Zeros using a single rpiboot process.

The new mode is enabled using the "-o" command line option and for example when used with "-d /boot" would first try to load files from /boot/USBPATH/ and if not found fallback to /boot/ . When using the Cluster HAT the USBPATH for P1 is "1-1.2.4", P2 is "1-1.2.3", P3 is "1-1.2.2" and P4 is "1-1.2.1" so when P1 is booting and looking for the config.txt file it will try to send /boot/1-1.2.4/config.txt but if the file isn't found will send /boot/config.txt as normal. This allows the

use of a custom config.txt/cmdline.txt for each Pi Zero.

The source for the hacked/modified rpiboot can be found on https://github.com/burtyb/usbboot

Network Configiration

The network now uses VLANs internally to separate the Internal and External networks.

Your local network doesn't need to support VLANs these are all handled within the Controller/Pi Zeros. With the configuration described here the Pi Zeros will still obtain an IP address from the DHCP server on your local network.

The original br0 interface which previously bridged eth0 (Controller) and ethpiX (Pi Zero) is removed and replaced by the Internal "brint" and External "brext" interfaces.

brint [Controller] is configured with the static IP address 172.19.180.254 and is bridged to the (untagged) ethpiX interface (usb0 on the Pi Zero size).

brext [Controller] is configured to obtain an IP address from the DHCP server on the local network (as br0 previously) and is bridged to eth0 on the Controller and the VLAN 10 interface ethpiX.10.

usb0 [Pi Zero] the untagged interface is used for NFSROOT (as initrd doesn't support VLAN). A static IP address 172.19.180.X is set for the Pi (so 172.19.180.1 for P1, ..180.2 for P2, etc). This is used for the NFSROOT but you can also use it for SSH/etc. access.

usb0.10 [Pi Zero] VLAN 10 is configured by dhcp (over **brext** to the eth0 interface on the controller) and will pickup an IP from the local network.

If you're using multiple Cluster HATs on the same network you just need to reconfigure the Controller by changing it's hostname (to controller1/controller2/etc.) and to prevent MAC address collisions the Pi Zeros need to be switched to different MAC addresses (and hostname changed to p11-p14/p21-p24/etc.). The internal 172.19.180.0/24 IP addresses will not conflict as they're isolated from the external traffic.

Setting up the software

Start by writing the Controller image to a new SD card (I'd advise using 16GB or bigger) as per the standard instructions - http://clusterhat.com/setup-software

I login using serial console on the Controller Pi (if you're using SSH you'll need to enable it first).

Login and do the normal first boot setup - set a ner password for the pi user, enable SSH, install the packages you'll need, download the Controller image (or copy it from your local machine) and expand the root filesystem.

```
sudo passwd pi
sudo touch /boot/ssh
sudo apt-get -y install git aria2 libusb-1.0-0-dev kpartx nfs-kernel-server
aria2c --seed-time 0 http://dist.8086.net/clusterhat/ClusterHAT-2017-03-02-lite-1-co
ntroller.zip.torrent
sudo raspi-config # Expand FS and reboot
```

The commands/edits below are ran as the root user.

```
sudo -i
```

Enable the kernel module for 802.1Q (VLAN) support.

```
echo 8021q >> /etc/modules
```

Edit /etc/network/interfaces and change "auto lo br0" to "auto lo brint brext"

Replace the existing "iface br0 config" section with the following config.

```
# Internal network (untagged on Pi Zeros)
auto brint
iface brint inet static
        bridge_ports none
        address 172.19.180.254
        netmask 255.255.255.0
        bridge_stp off
        bridge_waitport 0
        bridge_fd 0

# External network (VLAN10 on Pi Zeros)
iface brext inet manual
        bridge_ports eth0
        bridge_stp off
        bridge_waitport 0
        bridge_fd 0
```

Edit /etc/dhcpcd.conf and change the "denyinterfaces" line to

```
denyinterfaces eth0 ethpi1 ethpi2 ethpi3 ethpi4 ethpi*.10 brint
```

Alter Cluster HAT udev rules to only rename the device if it's a real USB device (and not the VLAN interface).

```
sed -i 's#"net", #"net", SUBSYSTEMS=="usb", #' /etc/udev/rules.d/90-clusterhat.rules
```

Generate the new network configuration for ethpiX and ethpiX.10 VLAN interfaces.

```
I=0
while [ $I -lt 256 ];do
echo "auto ethpi$I ethpi$I.10
```

```
allow-hotplug ethpi$I
allow-hotplug ethpi$I.10
# Internal network untagged
iface ethpi$I inet manual
        pre-up brctl addif brint ethpi$I
        up ifconfig ethpi$I up
        post-up ifup ethpi$I.10

# External network (VLAN 10)
iface ethpi$I.10 inet manual
        pre-up brctl addif brext ethpi$I.10
        up ifconfig ethpi$I.10 up
        post-up ifup ethpi$I.10"
echo
let I=$I+1
done > /etc/network/interfaces.d/clusterhat
```

Reboot the Controller Pi again to activate the changes.

```
reboot
```

Prepare directories for USBBOOT

```
mkdir -p /var/lib/clusterhat/nfs/{p1,p2,p3,p4} # NFS filesystems
mkdir /var/lib/clusterhat/boot # /boot filesystem for rpiboot tool
```

Extract the Controller image (all images are really a controller image set to reconfigure on first boot so downloading p1 would not alter the steps below).

```
cd /var/lib/clusterhat/
unzip ~pi/ClusterHAT-2017-03-02-lite-1-controller.zip
rm ~pi/ClusterHAT-2017-03-02-lite-1-controller.zip
mkdir mnt
LOOP=`losetup -f --show ClusterHAT-2017-03-02-lite-1-controller.img`
kpartx -av $LOOP
mount `echo $LOOP|sed s#dev#dev/mapper#`p2 mnt
mount `echo $LOOP|sed s#dev#dev/mapper#`p1 mnt/boot
tar -cC mnt . | tar -xvC nfs/p1/
umount mnt/boot
umount mnt
kpartx -dv $LOOP
losetup -d $LOOP
```

Setup files within NFS root for Pi Zeros

```
tar -cC nfs/p1/ . | tar -xvC nfs/p2/
tar -cC nfs/p1/ . | tar -xvC nfs/p3/
tar -cC nfs/p1/ . | tar -xvC nfs/p4/
cp -r nfs/p1/boot .
ln -s ../nfs/p4/boot/ boot/1-1.2.1
ln -s ../nfs/p3/boot/ boot/1-1.2.2
ln -s ../nfs/p2/boot/ boot/1-1.2.3
ln -s ../nfs/p1/boot/ boot/1-1.2.4
sed -i "s#^denyinterfaces.*#denyinterfaces eth0 ethpi1 ethpi2 ethpi3 ethpi4 usb0.10
usb0#" nfs/p*/etc/dhcpcd.conf
sed -i "s#^auto usb0#auto usb0.10#" nfs/p*/etc/network/interfaces
sed -i "s#^allow-hotplug usb0#allow-hotplug usb0.10#" nfs/p*/etc/network/interfaces
sed -i "s#^iface usb0 inet dhcp#iface usb0.10 inet dhcp#" nfs/p*/etc/network/interfa
ces
for I in 1 2 3 4 ; do
 sed -i "s/.*mmcblk0.*//" nfs/p$I/etc/fstab
 cp "/usr/share/clusterhat/cmdline.p$I" nfs/p$I/boot/cmdline.txt
 cp -f "/usr/share/clusterhat/interfaces.p" nfs/p$I/etc/network/interfaces
 cp -f "/usr/share/clusterhat/issue.p" nfs/p$I/etc/issue
 sed -i "s#^127.0.1.1.*#127.0.1.1\tp$I#g" nfs/p$I/etc/hosts
 sed -i "s/^#dtoverlay=dwc2$/dtoverlay=dwc2/" nfs/p$I/boot/config.txt
 echo "p$I" > nfs/p$I/etc/hostname
 sed -i "s#root=/dev/mmcblk0p2 rootfstype=ext4#root=/dev/nfs nfsroot=172.19.180.254:
/var/lib/clusterhat/nfs/p$I rw ip=172.19.180.$I:172.19.180.254::255.255.255.0:p$I:us
b0:static#" nfs/p$I/boot/cmdline.txt
 sed -i "s#MODULES=most#MODULES=netboot#" nfs/p$I/etc/initramfs-tools/initramfs.conf
 echo "BOOT=nfs" >> nfs/p$I/etc/initramfs-tools/initramfs.conf
 echo -e "dwc2\ng_cdc\nuio_pdrv_genirq\nuio\nusb_f_acm\nu_serial\nusb_f_ecm\nu_ether
\nlibcomposite\nudc_core\nipv6\n" >> nfs/p$I/etc/initramfs-tools/modules
 chroot nfs/p$I/ mkinitramfs -o /boot/initramfs.img 4.4.50+
 echo "initramfs initramfs.img" >> nfs/p$I/boot/config.txt
done
```

Setup NFS server

```
for I in 1 2 3 4 ; do
 echo "/var/lib/clusterhat/nfs/p$I 172.19.180.$I(rw,sync,no_subtree_check,no_root_sq
uash)" >> /etc/exports
done
systemctl enable rpcbind
systemctl restart rpcbind
systemctl enable nfs-kernel-server
systemctl restart nfs-kernel-server
```

Build the modified rpiboot tool

```
cd /var/lib/clusterhat
git clone https://github.com/burtyb/usbboot
cd usbboot
```

```
make
```

Enable SSH and set passwords for the Pi Zeros (this will ask for the password for each Pi Zero individually).

```
for I in 1 2 3 4; do
 touch nfs/p$I/boot/ssh
 echo "Updating password for Pi Zero p$I"
 chroot nfs/p$I/ passwd pi
done
```

Start rpiboot in a screen session (this could probably be setup as a service).

```
screen -S rpiboot -d -m /var/lib/clusterhat/usbboot/rpiboot -d /var/lib/clusterhat/b
oot/ -o -l -v
```

You should now be able to use the normal command to power up the Pi Zeros (with no SD cards) using.

```
clusterhat on
```

They will take quite a while to boot but you should be able to see some progress from "brctl show" and "ifconfig" when the interfaces are added and the internal 172.19.180.x IPs should start to ping after the initramfs is running.

Known Issues

The main issue with this setup is that after a couple of reboots the Pi Zeros stop showing up correctly on the Controller as USB devices. The work around for this is to disconnect/reconnect the USB cable to the ClusterHAT.

After rebooting the Controller Pi the NFS related services need to be restarted.

```
systemctl restart rpcbind
systemctl restart nfs-kernel-server
```

Also the rpiboot running in screen also needs to be restarted.

```
screen -S rpiboot -d -m /var/lib/clusterhat/usbboot/rpiboot -d /var/lib/clusterhat/b
```

```
oot/ -o -l -v
```

I'm also sure the network config is wrong for the Pi Zero as it sits there waiting for longer than it probably should.

Open to suggestions/fixes/ideas over on the Google Group.

Unique solution ID: #1084
Author: n/a
Last update: 2019-07-19