

Interface Config

Jan 2020

Introduction

When using the M1 chassis, it should have been installed using the official .ISO, so this is applicable to both RX1 and MKSP installation.

Even though the interface configuration may look okay, there is an issue with the way the interfaces are installed, to remedy this you can run a small script that will recreate all of the configuration files.

The script `update_network_devices.sh` can be downloaded

here

:

To use the script, copy it to a folder that you have rights to use (/mfeng/home) and give the file execute rights:

```
chmod +x update_network_devices.sh
```

Now run the file using the following command:

```
./update_network_devices.sh
```

The script will create new ifcfg-ethx files depending on installed hardware (what NIC cards you have installed).

Navigate to /etc/sysconfig/network-scripts and edit ifcfg-eth0 (as sudo)

```
sudo /etc/sysconfig/network-scripts/ifcfg-eth0
```

Now add all of your IP Details.

Alternatively you can configure the interfaces via nmtui or the front panel (might have to reboot to use the front panel)

Interface Script

```
#!/bin/sh

# Script to assign names and hardware IDs to ifcfg-net* files as well as turn on
# consistent NIC naming. This resolves the issue of NIC ports swapping around
# due to boot-time race conditions. It should be run manually once after upgrade
# if you have multiple NICs. However it will delete all network configuration
# and rename NICs to be net0, net1, etc, which is not supported by all software.

echo "RUNNING SCRIPT: update_network_devices VERSION 0.2"

# recognised NICs
DEVICE_CODE_ONBOARD="8086:1533"
Connection
DEVICE_CODE_10G_CARD="8086:10fb|8086:1572|8086:1528"
SFI/SFP+ Network Connection

# from https://devicehunt.com/
# Intel Corporation - I210 Gigabit Network
# Intel Corporation - 82599ES 10-Gigabit
# Intel Corporation - Ethernet Controller X710
# Intel Corporation - Ethernet Controller 10-
# Intel Corporation - I350 Gigabit Network

for 10GbE SFP+
Gigabit X540-AT2
DEVICE_CODE_1G_CARD="8086:1521"
Connection

TMP_FILE="/tmp/nic_order_tmp.txt"
FILE_ETHX_TEMPLATE="/etc/sysconfig/network-scripts/ifcfg-
```

```
GRUB_FILE="/etc/default/grub"

function create_ifcfg() {
touch $1
cat > $1 << "EOF"
TYPE=Ethernet
BOOTPROTO="dhcp"
ONBOOT=yes
PERSISTENT_DHCLIENT=yes
IPV6INIT=no
PEERNTP=no
EOF
chmod 755 $1
}

function detect_nic_card() {
    device_code=$1
    lspci_id=$(lspci -nn | egrep "\[${device_code}\]" | egrep -o "[0-9]{2}:[0-9]{2}\.[0-9]{2}")
    for i in ${lspci_id};
    do
        mac_addr=$(dmesg | fgrep $i | egrep -o "([0-9a-f]{2}:){5}[0-9a-f]{2}")
        echo "eth$net_index $mac_addr $i" >> $TMP_FILE
        net_index=$(( $net_index + 1 ))
    done
}

# Check we have root privledges
if [[ $EUID != 0 ]]; then
    echo "root privileges needed, please re-run with sudo"
    exit 1
fi

# Delete existing ifcfg files
mv ${FILE_ETHX_TEMPLATE}lo /tmp/ifcfg-lo
rm -Rf ${FILE_ETHX_TEMPLATE}*
mv /tmp/ifcfg-lo ${FILE_ETHX_TEMPLATE}lo

rm -rf $TMP_FILE
touch $TMP_FILE

# Create entries in nic order file
net_index=0
detect_nic_card $DEVICE_CODE_ONBOARD
detect_nic_card $DEVICE_CODE_10G_CARD
detect_nic_card $DEVICE_CODE_1G_CARD

#iterate through nic order file and update ifcfg files
echo "Hard coding ethernet devices as follows:"
while read line; do

    echo $line

    name=`echo "$line" | awk -F ' ' '{print $1}`
    mac=`echo "$line" | awk -F ' ' '{print $2}`
    ifcfg_file=${FILE_ETHX_TEMPLATE}$name

    # Check if an ifcfg file already exists for this interface and if not, create it
    if [ ! -f $ifcfg_file ]; then
        create_ifcfg ${ifcfg_file}
    fi

    # Remove entries that we are going to set from the ifcfg file
    sed -i '/DEVICE/d' ${FILE_ETHX_TEMPLATE}$name
    sed -i '/NAME/d' ${FILE_ETHX_TEMPLATE}$name
    sed -i '/HWADDR/d' ${FILE_ETHX_TEMPLATE}$name
    sed -i '/UUID/d' ${FILE_ETHX_TEMPLATE}$name #Remove UUID entry to ensure we do not have a conflict
between current UUID and the MAC address
    sed -i '/NM_CONTROLLED/d' ${FILE_ETHX_TEMPLATE}$name
    sed -i '/^[[:space:]]*$/d' ${FILE_ETHX_TEMPLATE}$name # Remove blank lines
```

```
# Add fields to ifcfg file
echo "DEVICE=$name" >> ${FILE_ETHX_TEMPLATE}$name
echo "NAME=$name" >> ${FILE_ETHX_TEMPLATE}$name
echo "HWADDR=$mac" >> ${FILE_ETHX_TEMPLATE}$name
echo "NM_CONTROLLED=yes" >> ${FILE_ETHX_TEMPLATE}$name

done <${TMP_FILE}

# Finally, enable consistent naming which ensures ifcfg files are used
sed -i 's/net.ifnames=0/net.ifnames=1/g' $GRUB_FILE
grub2-mkconfig -o $(readlink -e /etc/grub2.cfg)

echo "Please review network configuration files ${FILE_ETHX_TEMPLATE}-X - THEN reboot"
```

From:
<http://cameraangle.co.uk/> - WalkerWiki - wiki.alanwalker.uk

Permanent link:
http://cameraangle.co.uk/doku.php?id=interface_config

Last update: **2023/03/09 22:35**

