

mx53 网络配置方法

这篇文章详解 mx53 的网络配置方法，其中一种是动态获取 IP，另一种是手工静态指定。使用内核传参进行 DHCP 时，如果在内核启动过程中向外发送的 DHCP 包未得到回复，则内核阻塞直到得到 DHCP 服务端的响应为止。**所以不是很建议使用这种方法**。强烈建议使用用户层配置脚本进行网络设置。

1、动态获取 IP 地址

内核传参的方法

setenv bootargs 'console=ttymx0,115200 **ip=dhcp** root=/dev/mmcblk0p1 rootwait rw'

```
eth0: Freescale FEC PHY driver [Generic PHY] (mii_bus:phy_addr=0:00, irq=-1)
Sending DHCP requests .
PHY: 0:00 - Link is Up - 100/Full
OK
IP-Config: Got DHCP answer from 0.0.0.0, my address is 192.168.2.3
IP-Config: Complete:
    device=eth0, addr=192.168.2.3, mask=255.255.255.0, gw=192.168.2.1,
    host=192.168.2.3, domain=, nis-domain=(none),
    bootserver=0.0.0.0, rootserver=0.0.0.0, rootpath=, mtu=576
EXT3-fs: barriers not enabled
kjournald starting. Commit interval 5 seconds
EXT3-fs (mmcblk0p1): using internal journal
EXT3-fs (mmcblk0p1): recovery complete
EXT3-fs (mmcblk0p1): mounted filesystem with writeback data mode
VFS: Mounted root (ext3 filesystem) on device 179:1.
Freeing init memory: 144K
/ #
```

```
OK
IP-Config: Got DHCP answer from 0.0.0.0, my address is 192.168.2.3
IP-Config: Complete:
    device=eth0, addr=192.168.2.3, mask=255.255.255.0, gw=192.168.2.1,
    host=192.168.2.3, domain=, nis-domain=(none),
    bootserver=0.0.0.0, rootserver=0.0.0.0, rootpath=, mtu=576
EXT3-fs: barriers not enabled
kjournald starting. Commit interval 5 seconds
EXT3-fs (mmcblk0p1): using internal journal
EXT3-fs (mmcblk0p1): recovery complete
EXT3-fs (mmcblk0p1): mounted filesystem with writeback data mode
VFS: Mounted root (ext3 filesystem) on device 179:1.
Freeing init memory: 144K
/ # ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:04:9F:00:EA:D3
          inet addr:192.168.2.3  Bcast:192.168.2.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:576  Metric:1
          RX packets:20 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2884 (2.8 KiB)  TX bytes:1180 (1.1 KiB)
          Base address:0x6000
/ #
```

用户层 DHCP 客户端方法

内核正常启动，通过下图可以看到，IP 地址、子网掩码等信息默认是没有默认。

```
/ # ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:04:9F:00:EA:D3
          BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
          Base address:0x6000
/ #
```

这里我们执行 `udhcpd` 程序，这是 DHCP 客户端程序，它会与 DHCP 服务器交互，获取 IP 地址。(这里可以使用开启 DHCP 服务的路由器试验)，通过下图，可以看到，成功获取 IP 地址及 DNS 服务器地址。

```
/ # udhcpd
udhcpd (v1.20.2) started
Setting IP address 0.0.0.0 on eth0
eth0: Freescale FEC PHY driver [Generic PHY] (mii_bus:phy_addr=0:00, irq=-1)
Sending discover...
PHY: 0:00 - Link is Up - 100/Full
Sending select for 192.168.2.3...
Lease of 192.168.2.3 obtained, lease time 7200
Setting IP address 192.168.2.3 on eth0
Deleting routers
route: SIOCDELRT: No such process
Adding router 192.168.2.1
Recreating /etc/resolv.conf
Adding DNS server 192.168.1.1
Adding DNS server 192.168.2.1
/ #
```

我们再查看一下网卡 0 的信息，发现设置正常。如下图所示。

```
/ # ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:04:9F:00:EA:D3
          inet addr:192.168.2.3  Bcast:255.255.255.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:12 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2116 (2.0 KiB)  TX bytes:656 (656.0 B)
          Base address:0x6000

/ #
```

这里我们 ping 一下路由器，发现网络是畅通着的。

```
/ # ping 192.168.2.1
PING 192.168.2.1 (192.168.2.1): 56 data bytes
64 bytes from 192.168.2.1: seq=0 ttl=64 time=8.983 ms
64 bytes from 192.168.2.1: seq=1 ttl=64 time=0.604 ms
64 bytes from 192.168.2.1: seq=2 ttl=64 time=0.562 ms
64 bytes from 192.168.2.1: seq=3 ttl=64 time=0.611 ms
64 bytes from 192.168.2.1: seq=4 ttl=64 time=0.580 ms
^C
--- 192.168.2.1 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 0.562/2.268/8.983 ms
/ #
```

2、手工指定 IP 地址

内核传参的方法

`setenv bootargs 'console=ttyMxc0,115200 ip=192.168.2.10 root=/dev/mmcbk0p1 rootwait rw'`

```
kjournald starting. Commit interval 5 seconds
EXT3-fs (mmcbk0p1): using internal journal
EXT3-fs (mmcbk0p1): recovery complete
EXT3-fs (mmcbk0p1): mounted filesystem with writeback data mode
VFS: Mounted root (ext3 filesystem) on device 179:1.
Freeing init memory: 144K
/ # PHY: 0:00 - Link is Up - 100/Full

/ # ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:04:9F:00:EA:D3
          inet addr:192.168.2.10  Bcast:192.168.2.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:7 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1108 (1.0 KiB)  TX bytes:0 (0.0 B)
          Base address:0x6000

/ #
```

通过上面的图可以看到，内核启动后，eth0 就直接有有效的 IP 地址信息了。下面我们 ping 一下路由器，网络也是畅通的。

```
/ # ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:04:9F:00:EA:D3
          inet addr:192.168.2.10  Bcast:192.168.2.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:7 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1108 (1.0 KiB)  TX bytes:0 (0.0 B)
          Base address:0x6000

/ # ping 192.168.2.1
PING 192.168.2.1 (192.168.2.1): 56 data bytes
64 bytes from 192.168.2.1: seq=0 ttl=64 time=3.214 ms
64 bytes from 192.168.2.1: seq=1 ttl=64 time=0.567 ms
64 bytes from 192.168.2.1: seq=2 ttl=64 time=0.580 ms
^C
--- 192.168.2.1 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 0.567/1.453/3.214 ms
/ #
```

用户层设置

在 shell 下，我们可以使用 ifconfig 进行 IP 地址的设置。内核启动后，起初是没有设置 IP 信息的，通过下图可以看出。

```
EXT3-fs (mmcblk0p1): using internal journal
EXT3-fs (mmcblk0p1): recovery complete
EXT3-fs (mmcblk0p1): mounted filesystem with writeback data mode
VFS: Mounted root (ext3 filesystem) on device 179:1.
Freeing init memory: 144K
/ # ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:04:9F:00:EA:D3
          BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
          Base address:0x6000

/ #
```

我们使用 ifconfig 设置 IP 和子网掩码，使用 route 设置网关。

```
/ # ifconfig eth0 192.168.2.22 netmask 255.255.255.0
eth0: Freescale FEC PHY driver [Generic PHY] (mii_bus:phy_addr=0:00, irq=-1)
/ # PHY: 0:00 - Link is Up - 100/Full

/ # route add default gw 192.168.2.1
/ # ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:04:9F:00:EA:D3
          inet addr:192.168.2.22  Bcast:192.168.2.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
          Base address:0x6000

/ #
```

现在我们 ping 一下路由器，看看通不通。通过下图可以看出，网络畅通。


```

/ # route add default gw 192.168.2.1
/ # ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:04:9F:00:EA:D3
          inet addr:192.168.2.22  Bcast:192.168.2.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
          Base address:0x6000

/ # ping 192.168.2.1
PING 192.168.2.1 (192.168.2.1): 56 data bytes
64 bytes from 192.168.2.1: seq=0 ttl=64 time=3.394 ms
64 bytes from 192.168.2.1: seq=1 ttl=64 time=0.613 ms
64 bytes from 192.168.2.1: seq=2 ttl=64 time=0.567 ms
^C
--- 192.168.2.1 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 0.567/1.524/3.394 ms
/ #

```

3、用户层配置脚本

通过上面的试验，我们发现，在用户层面进行配置更加方便。所以对应于用户层有 2 种方法：一是使用 udhcpd 客户端进行获取，这个可以加在开机启动脚本中；二是使用 ifconfig 和 route 命令进行配置。

① 开机脚本，DHCP

内核启动流程，init 进程→读取/etc/inittab→执行/etc/init.d/rcS，我们就将要执行的命令写在这个脚本文件中。如下：

```

/etc/init.d # cat rcS
#!/bin/sh
PATH=/sbin:/bin:/usr/sbin:/usr/bin
export PATH
#call /etc/fstab
mount -a
echo "Getting IP from DHCP Server..."
udhcpd
echo "Getting IP OK!"

/etc/init.d #

```

重新加电，看一下结果如何。获取到了 IP 和 DNS 信息。

```

kjournald starting.  Commit interval 5 seconds
EXT3-fs (mmcblk0p1): using internal journal
EXT3-fs (mmcblk0p1): recovery complete
EXT3-fs (mmcblk0p1): mounted filesystem with writeback data mode
VFS: Mounted root (ext3 filesystem) on device 179:1.
Freeing init memory: 144K
Getting IP from DHCP Server...
udhcpd (v1.20.2) started
Setting IP address 0.0.0.0 on eth0
eth0: Freescale FEC PHY driver [Generic PHY] (mii_bus:phy_addr=0:00, irq=-1)
Sending discover...
PHY: 0:00 - Link is Up - 100/Full
Sending select for 192.168.2.3...
Lease of 192.168.2.3 obtained, lease time 7200
Setting IP address 192.168.2.3 on eth0
Deleting routers
route: SIOCDELRT: No such process
Adding router 192.168.2.1
Recreating /etc/resolv.conf
Adding DNS server 192.168.1.1
Adding DNS server 192.168.2.1
Getting IP OK!
/ #

```

② 开机脚本，ifconfig 和 route

这种方法就是针对于静态指定，将需要执行的命令加到/etc/init.d/rcS 中就可以了。

```
/etc/init.d # cat rcS
#!/bin/sh
PATH=/sbin:/bin:/usr/sbin:/usr/bin
export PATH
#call /etc/fstab
mount -a
echo "Setting IP by manual....."
ifconfig eth0 down
ifconfig eth0 192.168.2.66 netmask 255.255.255.0
route add default gw 192.168.2.1
ifconfig eth0 up
echo "Setting IP OK!"

/etc/init.d #
```

开机试验一下，结果如下图。

```
EXT3-fs (mmcblk0p1): using internal journal
EXT3-fs (mmcblk0p1): recovery complete
EXT3-fs (mmcblk0p1): mounted filesystem with writeback data mode
VFS: Mounted root (ext3 filesystem) on device 179:1.
Freeing init memory: 144K
Setting IP by manual.....
eth0: Freescale FEC PHY driver [Generic PHY] (mii_bus:phy_addr=0:00, irq=-1)
Setting IP OK!
/ # PHY: 0:00 - Link is Up - 100/Full

/ # ping 192.168.2.1
PING 192.168.2.1 (192.168.2.1): 56 data bytes
64 bytes from 192.168.2.1: seq=0 ttl=64 time=3.384 ms
64 bytes from 192.168.2.1: seq=1 ttl=64 time=0.587 ms
64 bytes from 192.168.2.1: seq=2 ttl=64 time=0.575 ms
^C
--- 192.168.2.1 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 0.575/1.515/3.384 ms
/ #
```

4、完毕！

shell.albert
2012/12/19