

How to setup an IEEE 802.1d MAC Bridge

The integrated Ethernet bridging tool of your DIL/NetPC ADNP/1520 Embedded Gateway Linux configuration implements the IEEE 802.1d (MAC Bridge) concepts.

A bridge is a device used to interconnect network segments and pass packets between those network segments. They operate at the layer 1 (Physical) and layer 2 (Data Link) of the OSI Model and their operation depends on the method they use to process the packets between the adjoining network segments.

The Application Board MB/1520-100 bridge software works as **Transparent** or **Learning Bridge**. Most of today's transparent bridges are more commonly called "Learning Bridges" because, in addition to buffering the packets at the port, they also examine information the source address field of a data packet. When a packet arrives at a port on a learning bridge, the bridge looks for its source address in its **Source Address Table** (SAT), an internal database.

If the source address is not in the SAT, it simply adds that source address and the bridge port number on which it arrived to the SAT. Then the bridge looks for the packet's destination in the SAT: 1. *If the destination exists in the SAT*, the packet is sent to the bridge port associated with the destination address. 2. *If the destination does NOT exist in the SAT*, the packet is sent out all of the bridge ports except for the port on which it arrived. A Learning Bridge learns from a packets source address and forwards it on the packets destination address.

For configuration and control of an Ethernet bridge the Embedded Gateway Linux of your Application Board MB/1520-100 offers a *brctl* command. The following steps shows how to create an Ethernet single port bridge with eth1 and eth2 and a multi port bridge with eth0, eth1 and eth2.

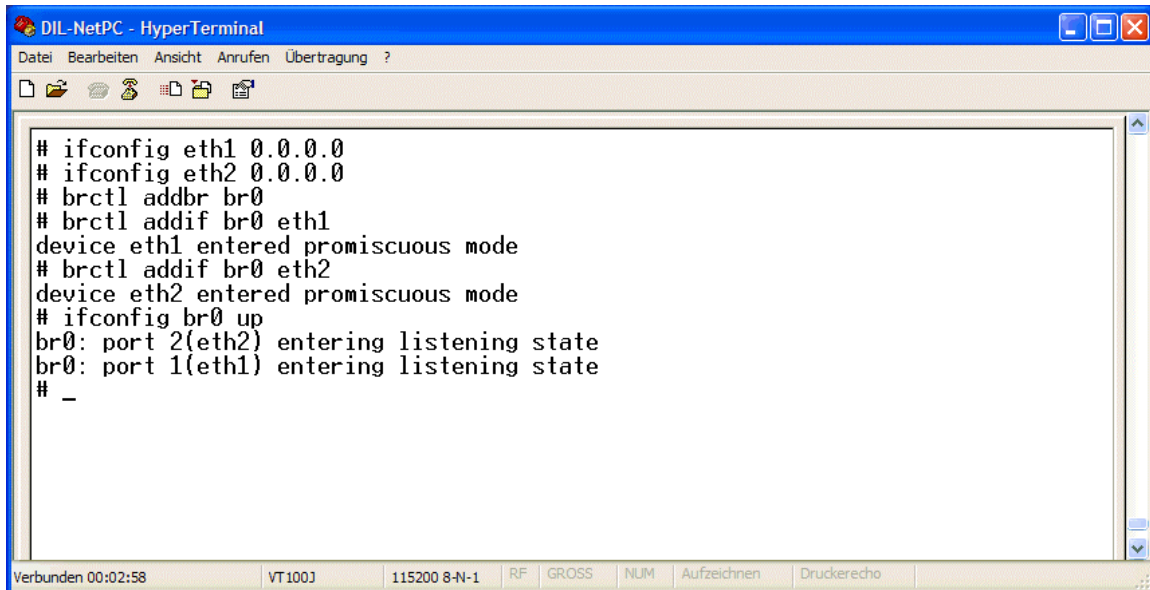
- **1. Step (Single Port Bridge):** Enter the following command sequence within an MB/1520-100 Telnet or RS232-based serial console session:

```
ifconfig eth1 0.0.0.0
ifconfig eth2 0.0.0.0
brctl addbr br0
brctl addif br0 eth1
brctl addif br0 eth2
ifconfig br0 up
```

The first two commands of this sample removes the default IP addresses from eth1 (MB/1520-100 LAN2 interface) and eth2 (MB/1520-100 LAN3 interface).

The three *brctl* commands defines a bridge with the name *br0* and adds the two interfaces eth1 and eth2 to this bridge.

The final *ifconfig* command brings the Ethernet bridge up to work. After this command the MB/1520-100 LAN2 and LAN3 Ethernet interfaces form a transparent dual-port switch. The LAN2 and LAN3 interfaces are now in promiscuous mode.



```

# ifconfig eth1 0.0.0.0
# ifconfig eth2 0.0.0.0
# brctl addbr br0
# brctl addif br0 eth1
device eth1 entered promiscuous mode
# brctl addif br0 eth2
device eth2 entered promiscuous mode
# ifconfig br0 up
br0: port 2(eth2) entering listening state
br0: port 1(eth1) entering listening state
# -

```

Please note: It is also possible to add the command sequence of the first step to the MB/1520-100 *autostart.sh* file. The MB/1520-100 comes with an Autostart feature. This feature allows you to start programs or execute Linux shell commands at boot time.

- **1. Step (Multi Port Bridge):** Enter the following command sequence within an MB/1520-100 RS232-based serial console session:

```

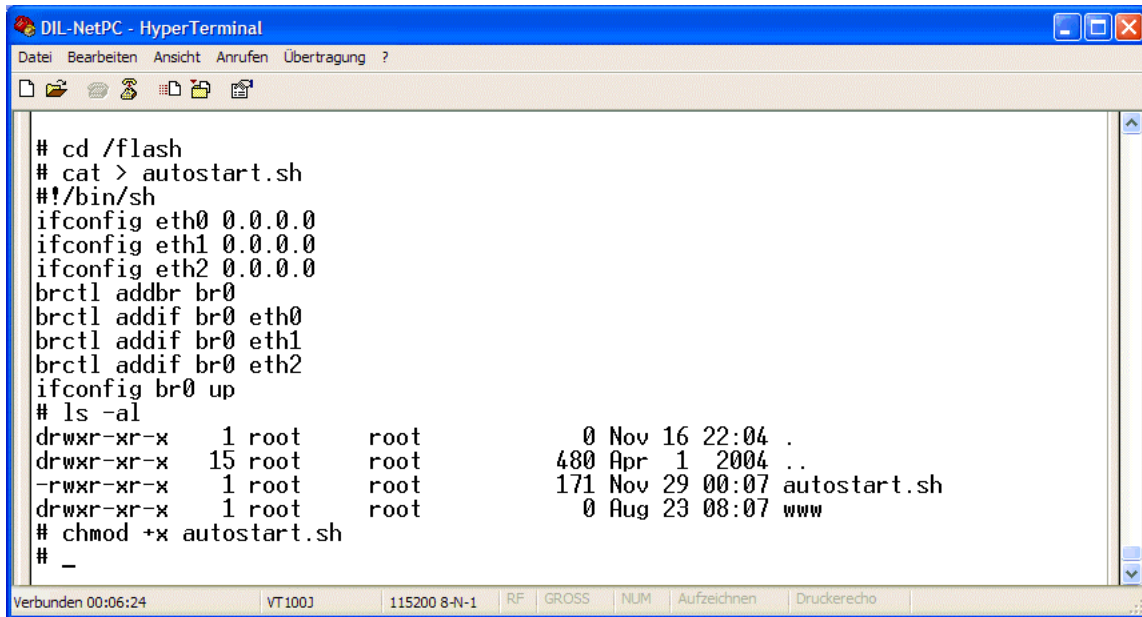
cd /flash
cat > autostart.sh
#!/bin/sh
ifconfig eth0 0.0.0.0
ifconfig eth1 0.0.0.0
ifconfig eth2 0.0.0.0
brctl addbr br0
brctl addif br0 eth0
brctl addif br0 eth1
brctl addif br0 eth2
ifconfig br0 up
CTRL-D (CTRL-D stops the Linux cat command)
chmod +x autostart.sh

```

The first three *ifconfig* commands of this sample removes the default IP addresses from eth0 (MB/1520-100 LAN1 interface), eth1 (MB/1520-100 LAN2 interface) and eth2 (MB/1520-100 LAN3 interface). The four *brctl* commands defines a bridge with the name *br0* and adds the three MB/1520-100 Ethernet LAN interface eth0, eth1 and eth2 to this bridge. The final *ifconfig* command brings the Ethernet bridge up to work.

After the four *brctl* commands and the final *ifconfig* command the MB/1520-100 LAN1, LAN2 and LAN3 Ethernet interfaces form a transparent three-port switch. The LAN1, LAN2 and LAN3 interfaces are now in promiscuous mode. In this operation mode the port of a network

device which enables that port to examine every packet being transmitted on the network segment they are attached to. This contrasts to the non-promiscuous mode of most network device ports that only examine those packets specifically addressed to them (unicast), or broadcast, or to those multicast address(es) to which the device port is a member. Promiscuous mode is set on bridge and router ports and used by those network devices, such as network sniffers, that must examine every packet on a network segment in order to process the packets or collect information about them.



```

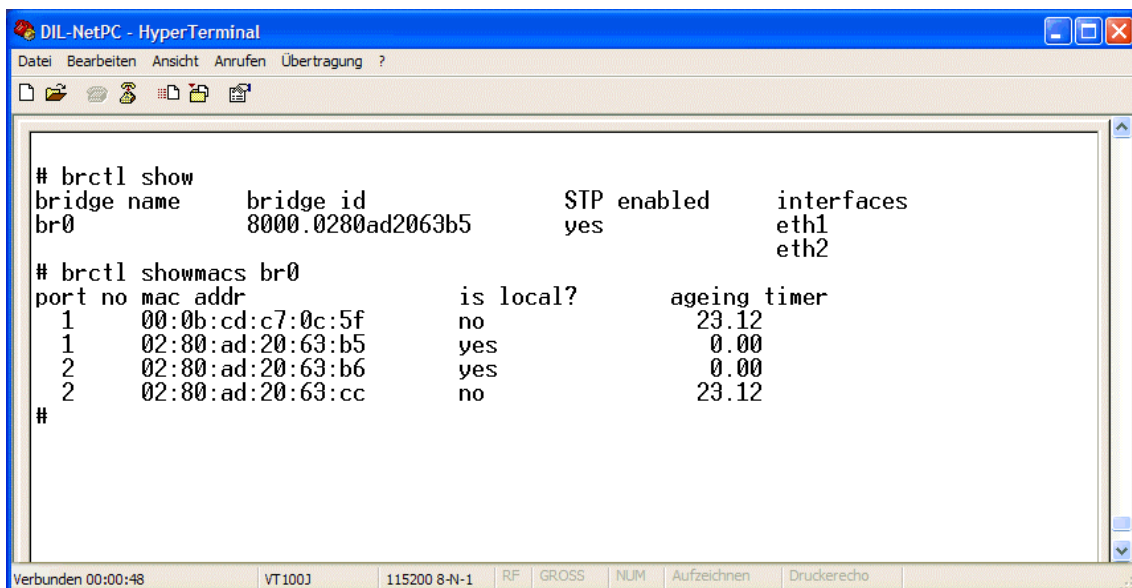
# cd /flash
# cat > autostart.sh
#!/bin/sh
ifconfig eth0 0.0.0.0
ifconfig eth1 0.0.0.0
ifconfig eth2 0.0.0.0
brctl addbr br0
brctl addif br0 eth0
brctl addif br0 eth1
brctl addif br0 eth2
ifconfig br0 up
# ls -al
drwxr-xr-x  1 root  root           0 Nov 16 22:04 .
drwxr-xr-x 15 root  root          480 Apr  1 2004 ..
-rwxr-xr-x  1 root  root          171 Nov 29 00:07 autostart.sh
drwxr-xr-x  1 root  root           0 Aug 23 08:07 www
# chmod +x autostart.sh
#

```

- **2. Step (Single/Multi Port Bridge):** Make sure that some LAN traffic goes over the Ethernet ports of the MB/1520-100. Then enter the following commands and watch the results:

```
brctl show
```

```
brctl showmacs br0
```



```

# brctl show
bridge name      bridge id      STP enabled    interfaces
br0              8000.0280ad2063b5  yes            eth1
                                                         eth2

# brctl showmacs br0
port no  mac addr          is local?    ageing timer
  1      00:0b:cd:c7:0c:5f  no           23.12
  1      02:80:ad:20:63:b5  yes          0.00
  2      02:80:ad:20:63:b6  yes          0.00
  2      02:80:ad:20:63:cc  no           23.12
#

```

The first command *brctl show* shows the devices in a bridge. The second command *brctl showmacs br0* displays some information about MAC addresses for a bridge. The following table gives an overview for the available *brctl* command.

MB/1520-100 <i>brctl</i> Command	Function
<i>brctl addbr</i> <bridge>	Creating a bridge device
<i>brctl addif</i> <bridge> <device>	Adding devices to a bridge
<i>brctl delbr</i> <bridge>	Deleting a bridge device
<i>brctl delif</i> <bridge> <device>	Deleting devices from a bridge
<i>brctl show</i>	Showing devices in a bridge
<i>brctl showmacs</i> <bridge>	Showing information about MAC addresses
<i>brctl showstp</i> <bridge>	Show bridge STP (Spanning Tree Protocol) info
<i>brctl setageing</i> <bridge> <time>	Set ageing time
<i>brctl setbridgeprio</i> <bridge> <prio>	Set bridge priority
<i>brctl setfd</i> <bridge> <time>	Set bridge forward delay
<i>brctl setgcint</i> <bridge> <time>	Set garbage collection interval
<i>brctl sethello</i> <bridge> <time>	Set hello time
<i>brctl setmaxage</i> <bridge> <time>	Set max message age
<i>brctl setpathcost</i> <bridge> <port> <cost>	Set path cost
<i>brctl setportprio</i> <bridge> <port> <prio>	Set port priority
<i>brctl stp</i> <bridge> <state>	Turn stp on/off

Table 1: MB/1520-100 *brctl* Command Overview

That's all.