



Application Note

Creating a Ethernet to WiFi Bridge with ConnectCard Wi-i.MX28

© Digi International Inc. 2014, All Rights Reserved.

The Digi logo is a registered trademark of Digi International, Inc.

All other trademarks mentioned in this document are the property of their respective owners.

Information in this document is subject to change without notice and does not represent a commitment on the part of Digi International.

Digi provides this document “as is,” without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of fitness or merchantability for a particular purpose. Digi may make improvements and/or changes in this manual or in the product(s) and/or the program(s) described in this manual at any time.

This product could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes may be incorporated in new editions of the publication.

Digi International Inc.

11001 Bren Road East

Minnetonka, MN 55343 (USA)

☎ +1 877 912-3444 or +1 952 912-3444

<http://www.digi.com>

History

Date	Version	Author	Description
Aug 2013	1.0	Leonid Makhnovskiy	First version
Mar 2014	1.1	Paolo Chiarlone	Minor cosmetic changes, updated graphics, updated information for DEL 5.9
May 2014	1.2	Paolo Chiarlone	Changed some wording, more consistent graphic layout
September 2014	1.3	Paolo Chiarlone	Extended encryption options, minor corrections
November 2014	1.4	Paolo Chiarlone	Added a new command sequence, minor corrections, verified WPA mode works fine
November 2014	1.5	Paolo Chiarlone	Minor corrections, added comment

Acronyms

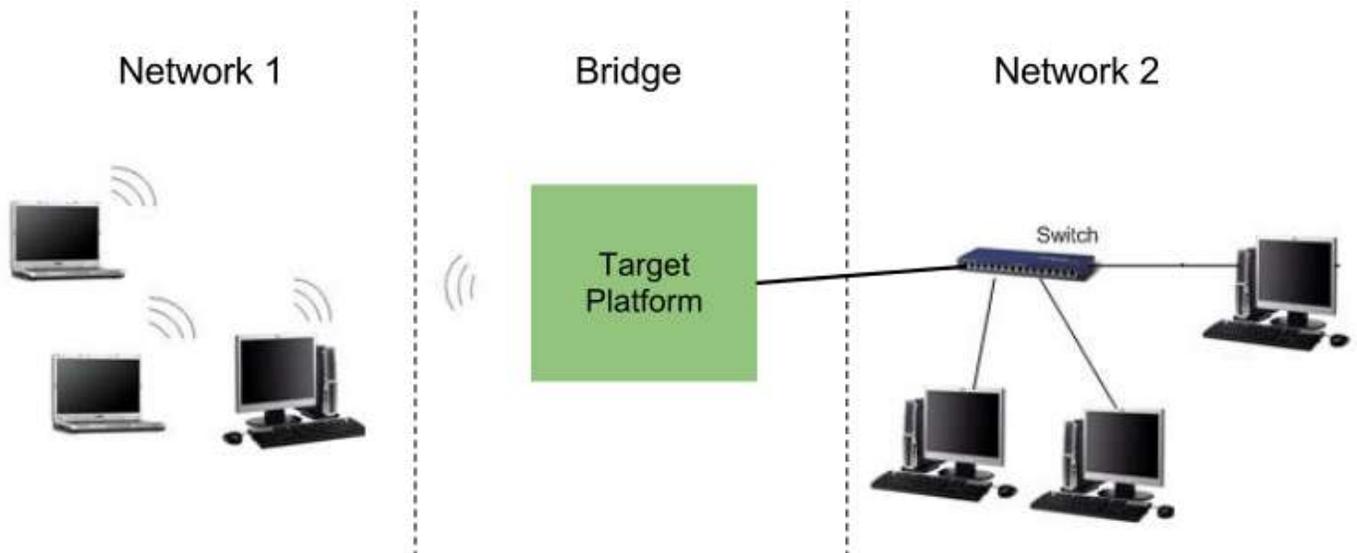
Name	Description
OS	Operating System
DEL	Digi Embedded Linux
CC	ConnectCard
LAN	Local Area Network
WAN	Wide Area Network

Network Bridging

This application note contains guidelines for configuring the Digi platform as a network bridge between two separate networks.

The basic idea is to have the system behaving like an Access Point or a WiFi Router.

It is targeted to systems running Digi DEL Linux.



1. Requirements

To make a target platform act as a network bridge between two networks the following is required:

- Target platform needs to have at least **two network interfaces** (either wired or wireless)
- Kernel image of the target platform requires to have support for bridging (explained later)
- The target platform root filesystem needs to have some special network applications

2. Methods

There are two methods for connecting a pair of networks:

- **Bridging:** The platform device acts like a tunnel that connects the two networks (or two network segments) forming one big subnet thus allowing devices to connect to each other without the need of routers
- **Port forwarding:** The platform device translates the address of a packet to a new destination according to a routing table

This application note explains the first one: **Bridging**

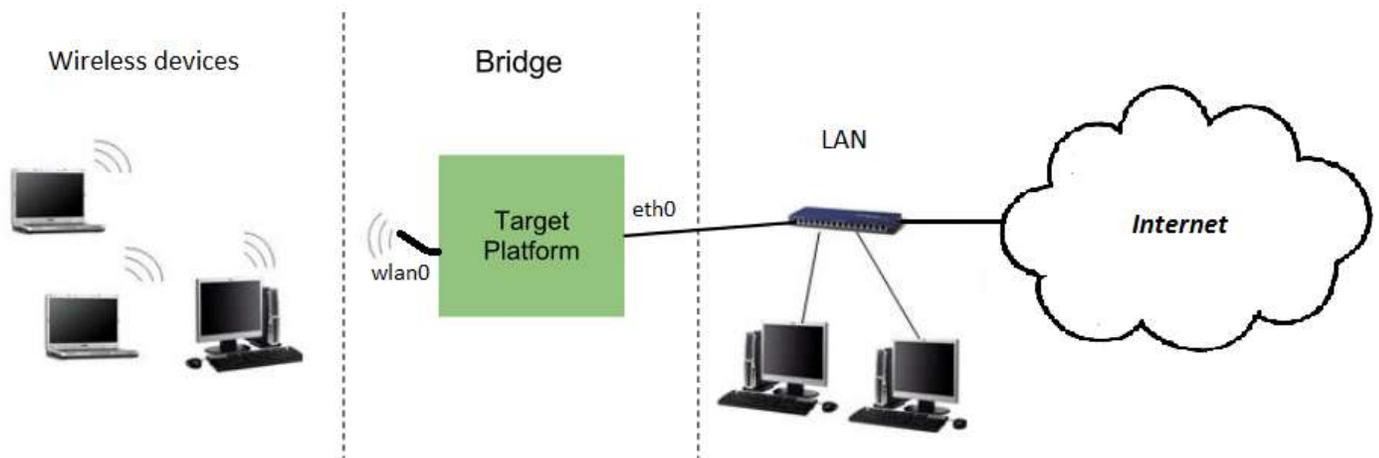
For information about port forwarding there are plenty of examples in the Internet about how to tweak the routing table to that end.

3. Bridging

This chapter illustrates how to configure a target platform to act as a network bridge.

Consider the following scenario:

- We have a target platform **ConnectCard for i.MX28** to act as bridge
- The target is connected through one wired Ethernet interface to a wired LAN that has access to the Internet
- We have three wireless devices
- We don't have any wireless AP, so our wireless devices cannot access the LAN



By configuring our platform as a bridge, we will be able to make the three wireless devices connect to the LAN and have access to the Internet.



A network bridge can be established between two network interfaces of any kind: two wired Ethernets, a wired Ethernet and a USB-Ethernet gadget, a USB-Ethernet gadget and a Wireless interface...



For a platform to act as a bridge using a wireless interface, the wireless interface needs to be able to operate in SoftAP mode

STEP1: Add support for network bridging to the kernel of the target platform

Go to your target platform DigiEL project and configure the kernel. Under **Networking Options**, on the right side panel, scroll down and enable the option **802.1d Ethernet Bridging**:

The screenshot shows the 'Digi Embedded Linux Configuration' window. The 'Networking options' section is expanded, and the '802.1d Ethernet Bridging (NEW)' option is checked and highlighted in orange. Below this, the '802.1d Ethernet Bridging BRIDGE' section is visible, containing the following text:

802.1d Ethernet Bridging BRIDGE

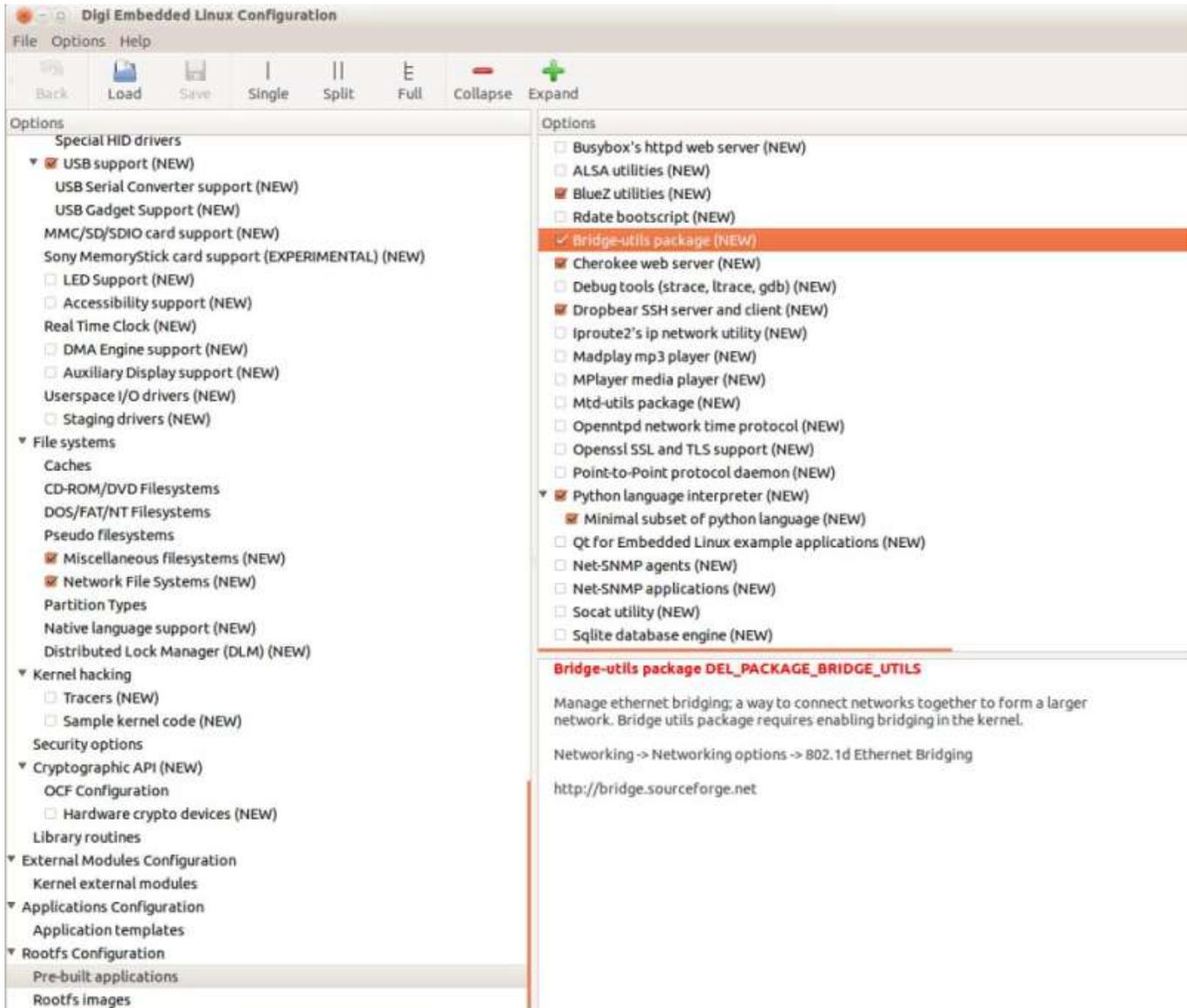
If you say Y here, then your Linux box will be able to act as an Ethernet bridge, which means that the different Ethernet segments it is connected to will appear as one Ethernet to the participants. Several such bridges can work together to create even larger networks of Ethernets using the IEEE 802.1 spanning tree algorithm. As this is a standard, Linux bridges will cooperate properly with other third party bridge products.

In order to use the Ethernet bridge, you'll need the bridge configuration tools; see <file:Documentation/networking/bridge.txt> for location. Please read the Bridge mini-HOWTO for more information.

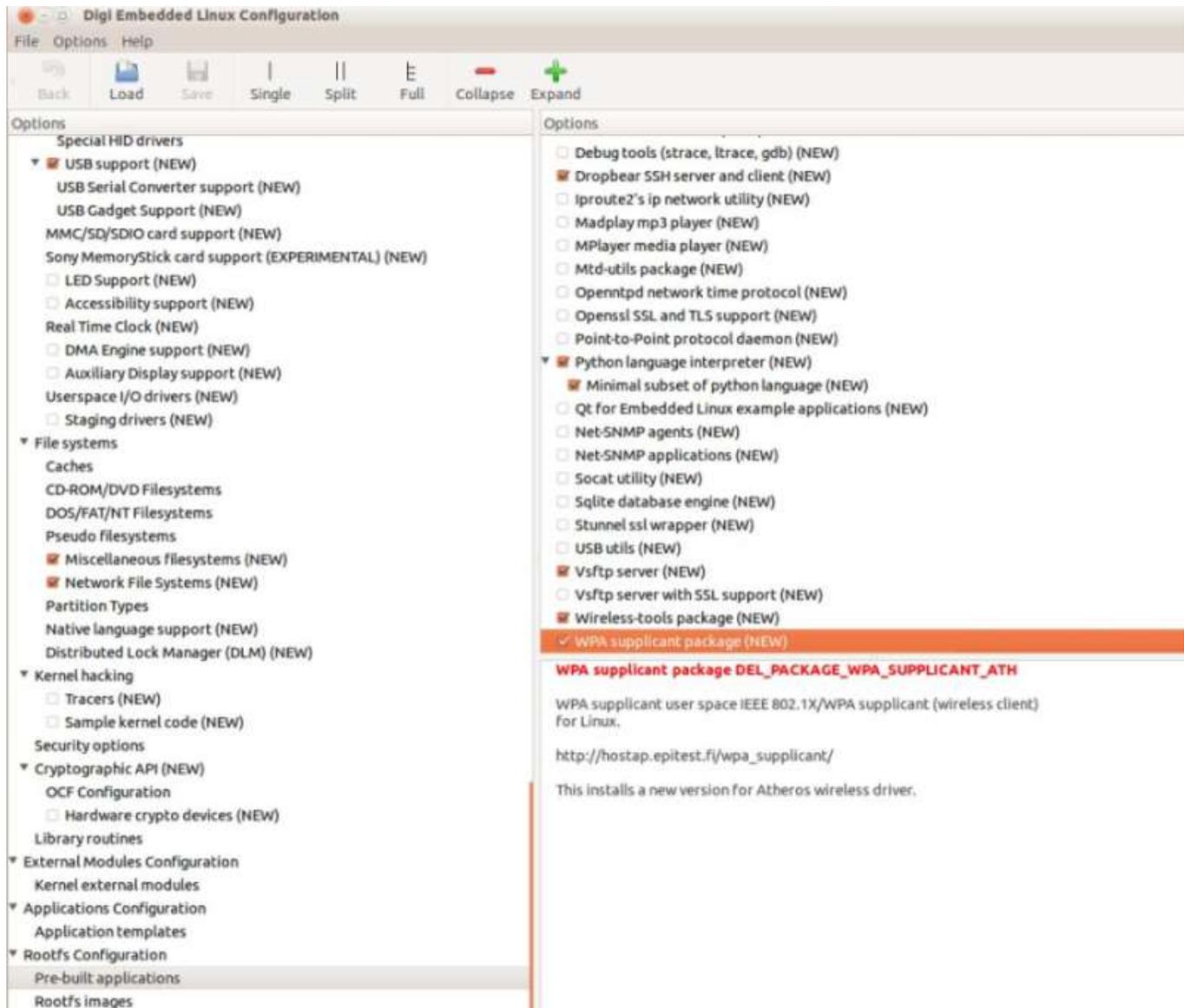
If you enable iptables support along with the bridge support then you turn your bridge into a bridging IP firewall. iptables will then see the IP packets being bridged, so you need to take this into account when setting up your firewall rules. Enabling arptables support when bridging will let arptables see bridged ARP traffic in the arptables FORWARD chain.

STEP 2: Add support for Bridge-utils to the root file system of the target platform

Go back to the options on the left panel, go to your project **RootFS Configuration** and then **Pre-built applications** to configure the rootfs. On the right side panel, enable the option **Bridge-utils package**:



Also, make sure the **Wireless-tools** package and the **WPA supplicant** package are selected as well (they should be already selected by default!):



STEP 3: Build and update the system of your platform

After all the configuration changes, save the configuration, (re)build the system and install it. You can then boot the Digi target platform via TFTP or transfer the code to the FLASH.

STEP 4: Configure the i.MX28 wireless interface to operate in SoftAP mode

In the Digi target platform, you have to edit the file `/etc/wpa_supplicant.conf` to have the correct network configuration. Below are various examples of how to configure the file (values here are an example, you could use a different configurations).

The Digi platform will work in SoftAP mode (for information about this mode, refer to the Digi DEL Help at paragraph **10. WLAN adapter guide**).

The operating frequency channel will be set as 2.412 MHz (Channel 1)

Configuration with no encryption (OPEN)

- The name of our target's SoftAP will be **OPENimx28**
- No password needed

```
# SoftAP mode (open/WEP encryption)
ctrl_interface=/var/run/wpa_supplicant
ctrl_interface_group=0
fast_reauth=1
update_config=1

# -- SoftAP mode OPEN
ap_scan=2
network={
    ssid="OPENimx28"
    mode=2
    frequency=2412
    key_mgmt=NONE
}
```

Configuration with WEP encryption

- The name of our target's SoftAP will be **WEPimx28**
- It will use WEP encryption
- WEP password will be "abcde"

```
# SoftAP mode (open/WEP encryption)
ctrl_interface=/var/run/wpa_supplicant
ctrl_interface_group=0
fast_reauth=1
update_config=1

# -- SoftAP mode WEP
ap_scan=2
network={
    ssid="WEPimx28"
    mode=2
    frequency=2412
    key_mgmt=NONE
    wep_key0="abcde"
    wep_tx_keyidx=0
    priority=5
}
```

Configuration with WPA encryption

- The name of our target's SoftAP will be **WPAimx28**
- It will use WPA-PSK encryption
- WPA password will be "12345678"
- To generate the encrypted password for the configuration file, it is necessary to use the following command:
- `wpa_passphrase <SSID name> <your PSK>`

e.g. in this case: `wpa_passphrase WPAimx28 12345678`

The output of the command can be copied into the configuration file directly.

```
# SoftAP mode (open/WEP encryption)
ctrl_interface=/var/run/wpa_supplicant
ctrl_interface_group=0
fast_reauth=1
update_config=1

# -- SoftAP mode WPA
ap_scan=2
network={
    ssid="WPAimx28"
    mode=2
    frequency=2412
    key_mgmt=WPA-PSK
    proto=RSN
    pairwise=CCMP
    #psk="12345678"
    psk=d0928a11b698fadf4510bea4b2da4dd5d35d73c1aa80658c6b5811610a1630eb
}
```

STEP 5: Start the i.MX28 wireless interface as SoftAP

Restart the **wpa_supplicant** to use the new configuration file, previously edited (Step 4) and add a new bridge interface:

NOTE: below, it is a "NL", not a "N1"

NOTE: if you change the name of your bridge (e.g. from br0 to br1) that has to be reflected into the wpa_supplicant command as well!



```
# killall wpa_supplicant
# brctl addbr br0

# wpa_supplicant -i wlan0 -D nl80211 -c /etc/wpa_supplicant.conf -b
br0 -B
```

You should receive a message similar to:



```
Successfully initialized wpa_supplicant
Using interface wlan0 with hwaddr 00:40:9d:53:xx:xx and ssid
"xxxxxxxx"
wlan0: CTRL-EVENT-CONNECTED - Connection to 00:40:9d:53:xx:xx
completed [id=0 id_str=]
```

Upon receiving the successful message, the WPA supplicant is correctly configured and the new wireless network should be visible but not yet ready.

STEP 6: Enable bridge on the target platform

Now you need to configure each of the three interfaces in the SoftAP just created on your target platform. The interfaces are: **eth0** (Ethernet), **wlan0** (WiFi) and **br0** (the bridge interface).

The steps are:

- Remove the current IP addresses of the two network interfaces that will be joined together by the bridge:



```
# ifconfig eth0 0.0.0.0
# ifconfig wlan0 0.0.0.0
```

- Create the bridge interface between the two network interfaces:



```
# brctl addif br0 wlan0
# brctl addif br0 eth0

# brctl show
bridge name      bridge id                STP enabled        interfaces
br0              8000.0004f328xxxx       no                 eth0
                                                         wlan0
```

- The last command (“**brctl show**”) should correctly list **eth0** and **wlan0** in the same **br0** bridge. If that does not happen, or it is showing something else, then bridge has not been created correctly, and should be deleted before trying to re-create it. To remove the bridge and restart the procedure, you can do:



```
# ifconfig br0 down
# brctl delbr br0
# killall wpa_supplicant
```

At this point, the command “**brctl show**” should list nothing and you can restart the procedure from **STEP 5**

- If the bridge has been correctly created, you can now assign a free IP address of the Ethernet LAN to the bridge. This is optional but it will allow you to reach the bridge (the target platform) in the network.



```
# ifconfig br0 192.168.1.5 netmask 255.255.0.0 up
...or...
# ifconfig br0 up (without IP)
```



*The bridge will be working fine and your client(s) will be able to access both LAN and WAN, but it might not be possible to ping or telnet the IP address assigned to br0 from WiFi if the address is not in your subnet. **This is an expected behavior.***



*In case the Ethernet cable gets disconnected while the bridge is active, upon reconnection the bridge functionality won't restart working. **This is a known limitation.***

To correct that, after reconnection, issue the command:

```
# ifconfig eth0 up
```

That will restore the correct functionality of the bridge.

That's it!

Your wireless clients, connected to the Digi target platform (CC Wi-i.MX28), acting as SoftAP, should now be able to reach any device in the network, and vice versa as well as browse the Internet (if the CC Wi-i.MX28 is connected to the Internet, of course).

[end of document]