IEEE 802.15.4 USB STICK WITH WIRESHARK™ FIRMWARE

REFERENCE MANUAL

ubisys®

1. Overview

Welcome to your ubisys IEEE 802.15.4 USB stick with Wireshark™ capture firmware!

This reference manual provides operating and maintenance instructions, command references etc. If you have any questions or need additional support, software or drivers, please visit our engineering support pages.

http://www.ubisys.de/engineering/support.html

2. Contents

3. Features

- Powerful IEEE 802.15.4 capture device for Wireshark™, the most widely used and actively maintained open-source network protocol analyzer software available to date
- Covers all channels in the 2.4 GHz band, i.e. channels 11-26 as specified in the IEEE 802.15.4 standard. Notice that one device is capable of capturing data on one channel at a time. Simultaneous multiple-channel capture is supported by using a number of ubisys IEEE 802.15.4 devices, each operating on a different channel. Diversity capture is supported by using more than one ubisys IEEE 802.15.4 stick on the same channel.
- With sixteen sticks, all channels in the 2.4GHz band are covered. This is very convenient, since you don't have to search for the channel the network is currently using. This is a must-have to observe frequency hopping systems like WirelessHART and useful for debugging frequency-agile systems like ZigBee PRO, ZigBee Green Power and ZigBee RF4CE
- On-board MCU: Advanced 32-bit ARM micro-controller running at 48MHz with 64KB SRAM – powerful enough to capture and buffer up to 128 packets (each comprising up to 127 bytes) until they are delivered to the host computer. Makes you not lose any packet due to buffer overruns, interrupt latencies or USB bus latencies – in contrast to other vendor's products based on slow 8-/16-bit controllers with limited RAM (typically 8KB)
- On-board PHY: Texas Instruments CC2520
- On-board meandered inverted-F antenna
- USB 2.0 full-speed device, bus-powered. Power consumption: 50mA in active mode. Thus, can be plugged into any USB port, even into passive hub ports, such as those integrated into keyboards
- Complies with Microsoft® RNDIS specification and is compatible with standard, pre-installed Windows drivers. Appears as a network adapter in device manager
- Creates ZigBee Encapsulation Protocol Version 2.0 Frames (ZEPv2), which can be immediately decoded by Wireshark's built-in dissectors. Includes channel information, link quality indication (LQI), received signal strength indication (RSSI) and a sequence counter
- Wireshark dissectors include: ZigBee, ZigBee PRO, ZigBee Green Power, 6lowpan. Wireshark can be extended with dissectors, including dissectors for your own proprietary protocols based on the IEEE 802.15.4 MAC
- Supports on-the-fly decryption of encrypted ZigBee network traffic (APS and NWK security)
- Exploit the networking capabilities of Wireshark to gather the data captured by a remote machine
- Create capture files and send them to colleagues, who can review the capture logs in Wireshark
- More convenient than Ethernet-based capture devices when used with mobile notebook computers etc.
- Extensible and future-proof design: Firmware updates via USB
- Supported on 32- and 64-bit Microsoft® Windows™ and Linux operating systems
- Timing accuracy: Approximately one IEEE 802.15.4 PHY symbol period (16 micro-seconds) in the time-stamps provided in the ZEPv2 header

4. Installation – Microsoft Windows

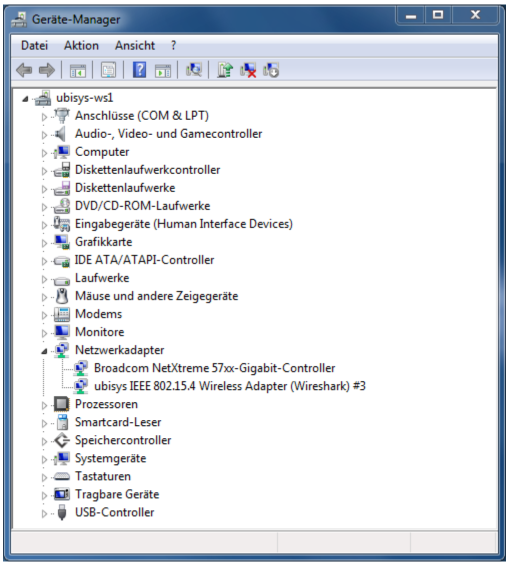
Download and install the Wireshark software from http://www.wireshark.org. The software installer package also includes WinPcap, a high-speed capture driver.

Download the ubisys IEEE 802.15.4 Wireshark USB stick driver package from here: http://www.ubisys.de/engineering/download-drivers.html.

Extract the files in the driver package into any folder on your hard disk.

Plug the device into any spare USB port on your PC. Windows will ask you for drivers. Point to the path where the extracted driver package files are stored. Follow the instructions on the screen.

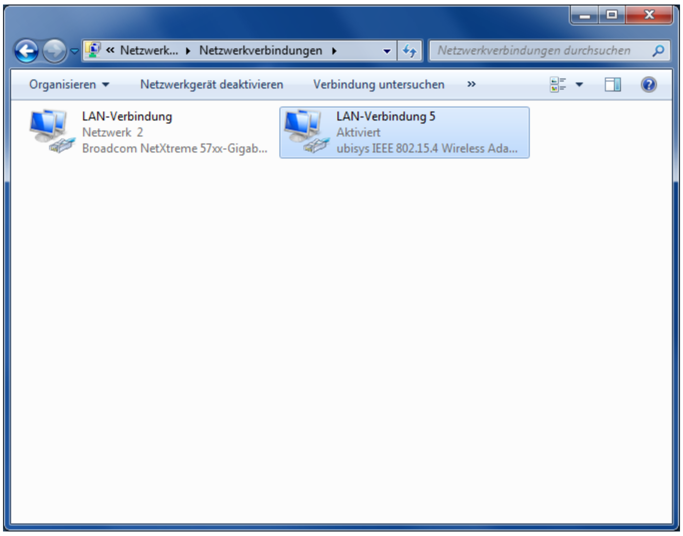
When you are done, verify that the device has been installed correctly by opening Windows Device Manager. Your ubisys IEEE 802.15.4 device with Wireshark capture firmware should appear under the network adapter section.



```
ubisys-ws1
  Anschlüsse (COM & LPT)
  Audio-, Video- und Gamecontroller
  Computer
  Diskettenlaufwerkcontroller
  Diskettenlaufwerke
  DVD/CD-ROM-Laufwerke
  Eingabegeräte (Human Interface Devices)
  Grafikkarte
  IDE ATA/ATAPI-Controller
  Laufwerke
  Mäuse und andere Zeigegeräte
  Modems
  Monitore
  Netzwerkadapter
    Broadcom NetXtreme 57xx-Gigabit-Controller
    ubisys IEEE 802.15.4 Wireless Adapter (Wireshark) #3
  Prozessoren
  Smartcard-Leser
  Speichercontroller
  Systemgeräte
  Tastaturen
  Tragbare Geräte
  USB-Controller
```

Figure 1: ubisys IEEE 802.15.4 Wireshark USB Stick in Windows Device Manager

We strongly recommend that you disable all network protocols that are linked with the new adapter in order to reduce traffic on the USB bus and the amount of data captured by Wireshark. In order to do so, open the adapter settings (via control panel, network connections).

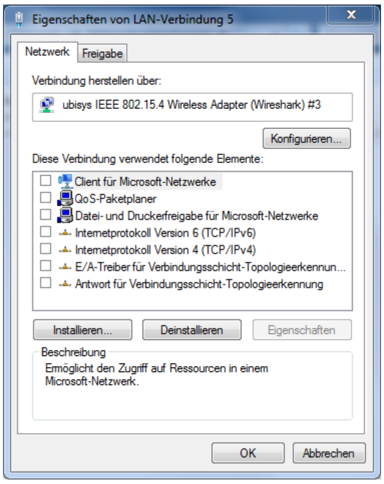

« Netzwerk... ▸ Netzwerkverbindungen ▸
Netzwerkverbindungen durchsuchen
Organisieren
Netzwerkgerät deaktivieren
Verbindung untersuchen
»

LAN-Verbindung
Netzwerk 2
Broadcom NetXtreme 57xx-Gigab...

LAN-Verbindung 5
Aktiviert
ubisys IEEE 802.15.4 Wireless Ada...

Figure 2: Network Connections

Right-click on the new ubisys IEEE 802.15.4 adapter and choose “Properties” from the pop-up menu that appears. Next, make sure that all protocol links are disabled.



Verbindung herstellen über:
ubisys IEEE 802.15.4 Wireless Adapter (Wireshark) #3

Diese Verbindung verwendet folgende Elemente:

- ☐ Client für Microsoft-Netzwerke
- ☐ QoS-Paketplaner
- ☐ Datei- und Druckerfreigabe für Microsoft-Netzwerke
- ☐ Internetprotokoll Version 6 (TCP/IPv6)
- ☐ Internetprotokoll Version 4 (TCP/IPv4)
- ☐ E/A-Treiber für Verbindungsschicht-Topologieerkennun...
- ☐ Antwort für Verbindungsschicht-Topologieerkennung

Beschreibung
Ermöglicht den Zugriff auf Ressourcen in einem Microsoft-Netzwerk.

Figure 3: Network Adapter Properties

You are done. Installation with recommended adapter settings is complete.

The device can be used for capturing packets now.

5. Installation – Linux

Download and install the latest Wireshark software source code from http://www.wireshark.org and compile it on your system, or simply use a pre-compiled package for your Linux distribution. A variety of Linux distributions and package managers is available, and we cannot provide information for all of them. But the general steps are the same or at least very similar. If you use debian or ubuntu Linux, you can download and install the pre-compiled package using the Advanced Packaging Tool. You need administrator privileges for installation.

```
# sudo apt-get install wireshark
```

Next, you need to add a suitable driver for the ubisys IEEE 802.15.4 Wireshark USB stick. Prepare your system for building kernel modules by downloading and installing the kernel headers that have been used to build your kernel, compiler tool-chains etc.

```
# sudo apt-get install linux-headers-$(uname -r) linux-libc-dev kernel-package
```

You are also going to need the kernel sources, since ubisys provides a patch to the rndis_host.c driver module. The community patch and accompanying shell script have both been provided by Marcus Ihde-Meister and have been tested by ubisys on different hardware platforms.

First, determine your kernel version using:

```
# uname -r
2.6.32-5-powerpc64
```

In this example, this is a 2.6 kernel. Substitute 2.6 with whatever is returned by uname –r up to the major release number.

```
# cd /usr/src
# sudo apt-get source linux-source-2.6
```

This will create a linux-xxx subdirectory with the complete, patched kernel sources under /usr/src.

Download and extract the ubisys IEEE 802.15.4 Wireshark USB stick driver package for Linux, which is provided as a gzip-compressed tar-ball.

```
# cd
# wget http://www.ubisys.de/downloads/ubisys-m7b-rndis.tgz
# tar -xzf ubisys-m7b-rndis.tgz
```

This creates a directory called ubisys-m7b-rndis under your home directory. Now, copy the original rndis_host.c file from your Linux source directory to this directory:

```
# cd ubisys-m7b-rndis
# cp /usr/src/linux-2.6_2.6.32/drivers/net/usb/rndis_host.c .
```

And apply the community patch:

```
# patch rndis_host.c rndis_host.c.patch
```

Notice: If certain hunks could not be applied, you should nevertheless continue to build.

Now, build the patched kernel module:

```
# make
```

This results in an output like this:

```
make -C /lib/modules/2.6.32-5-powerpc64/build/ M=/root/ubisys-m7b-rndis modules
make[1]: Entering directory `/usr/src/linux-headers-2.6.32-5-powerpc64'
  CC [M]  /root/ubisys-m7b-rndis/rndis_host.o
  Building modules, stage 2.
  MODPOST 1 modules
  CC      /root/ubisys-m7b-rndis/rndis_host.mod.o
  LD [M]  /root/ubisys-m7b-rndis/rndis_host.ko
```

Finally, install the modified kernel module:

```
# sudo make install
```

In case the module has been loaded previously, you must unload it first. You can use the tool lsmod to check if the module is active and rmmod to remove active modules. If rndis_wlan is also loaded, you have to unload it first, since it depends on rndis_host.

```
# lsmod
# sudo rmmod rndis_host
```

It is strongly recommended that you disable Internet Protocol version 4 and 6 bindings to that interface to prevent any traffic being generated on the sniffer interface.

Read the section on how to configure the capture channel to make the device operational, here we want to capture on channel 26:

```
# sudo ./ieee802154_options.sh –c 26
```

Now we need to check the interface that has been created for the capture device. Print the recent kernel messages to identify the interface:

```
# dmesg
```

Creates an output like this:

```
...
[2446022.499686] rndis_host ieee802154 channel is 26
[2446022.502352] rndis_host 1-1.4:1.0: eth3: register 'rndis_host' at usb-0000:00:1d.7-
1.4, RNDIS device, 00:1f:ee:00:01:84
...
```

Now, bring the interface up, such that it can be used by Wireshark:

```
# sudo ifconfig eth3 up
```

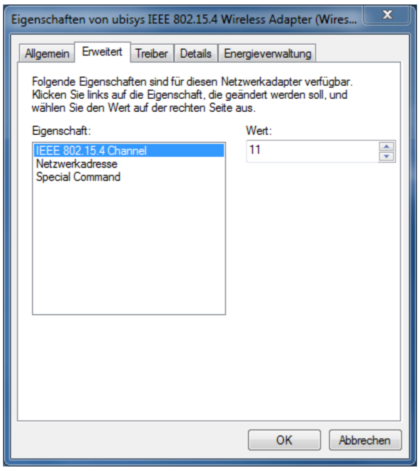
In the interface list shown in Wireshark pick eth3 as the capture interface.

Tested on ubuntu 12.04, i686, Kernel 3.2.0-32 and debian 6.0.6, ppc64, Kernel 2.6.32-5.

6. Setting the Channel for Capture – Microsoft Windows

Your ubisys IEEE 802.15.4 USB stick with Wireshark sniffer firmware is capable of capturing packets on any of the 16 channels in the 2.4GHz band, i.e. channels 11-26 according to the IEEE 802.15.4 standard. However, only one channel at a time can be captured. If you need to capture more channels concurrently, you will need one USB stick per channel, i.e. 16 sticks if you want to capture packets on all channels simultaneously, e.g. to analyze frequency hopping systems.

In order to select the channel for capture, open Windows Device Manager and right-click on your ubisys IEEE 802.15.4 Wireless Adapter for Wireshark and switch to the advanced settings tab:


Eigenschaften von ubisys IEEE 802.15.4 Wireless Adapter (Wires...
Allgemein
Erweitert
Treiber
Details
Energieverwaltung
OK
Abbrechen

Folgende Eigenschaften sind für diesen Netzwerkadapter verfügbar. Klicken Sie links auf die Eigenschaft, die geändert werden soll, und wählen Sie den Wert auf der rechten Seite aus.

Eigenschaft:
- IEEE 802.15.4 Channel
- Netzwerkadresse
- Special Command

Wert: 11

Figure 4: Network Adapter Properties, Advanced Settings

Select the IEEE 802.15.4 Channel property and set its value to the channel you want to capture, i.e. any decimal value in the range 11-26.

You can change the setting at any time. However, the adapter will disconnect and reconnect, so any live-capture currently in progress will be interrupted.

Notice that the Special Command property must be set to “no value” for normal operation.

7. Setting the Channel for Capture – Linux

Use the ieee802154_options.sh shell script, which is included in the Linux driver tar-ball, to start capturing IEEE 802.15.4/ZigBee frames on any of the 16 channels in the 2.4GHz band.

```
# sudo ./ieee802154_options.sh -c 26
```

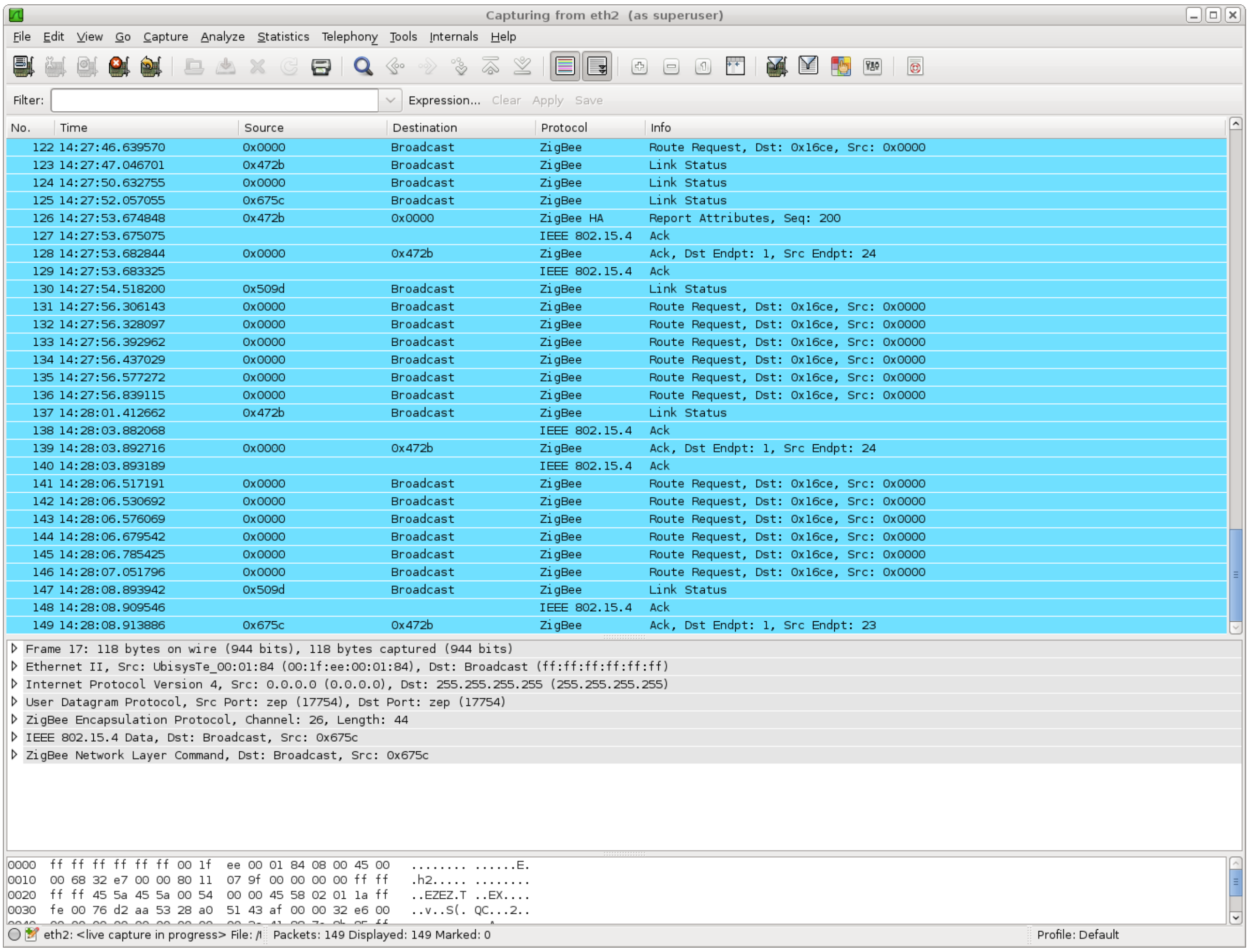
Instead of 26, you can enter a number in the range 11…26. If you want to make sure the command has been accepted, use dmesg to print the kernel message log and look out for a rndis_host message like this one:

```
...
[349673.652872] rndis_host ieee802154 channel is 26
...
```

Now you can start Wireshark:

```
# sudo wireshark &
```

The channel is also shown in the ZigBee encapsulation Protocol dissector:


Capturing from eth2 (as superuser)
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply Save

```
No.  Time             Source   Destination  Protocol     Info
 122 14:27:46.639570  0x0000   Broadcast    ZigBee       Route Request, Dst: 0x16ce, Src: 0x0000
 123 14:27:47.046701  0x472b   Broadcast    ZigBee       Link Status
 124 14:27:50.632755  0x0000   Broadcast    ZigBee       Link Status
 125 14:27:52.057055  0x675c   Broadcast    ZigBee       Link Status
 126 14:27:53.674848  0x472b   0x0000       ZigBee HA    Report Attributes, Seq: 200
 127 14:27:53.675075                        IEEE 802.15.4 Ack
 128 14:27:53.682844  0x0000   0x472b       ZigBee       Ack, Dst Endpt: 1, Src Endpt: 24
 129 14:27:53.683325                        IEEE 802.15.4 Ack
 130 14:27:54.518200  0x509d   Broadcast    ZigBee       Link Status
 131 14:27:56.306143  0x0000   Broadcast    ZigBee       Route Request, Dst: 0x16ce, Src: 0x0000
 132 14:27:56.328097  0x0000   Broadcast    ZigBee       Route Request, Dst: 0x16ce, Src: 0x0000
 133 14:27:56.392962  0x0000   Broadcast    ZigBee       Route Request, Dst: 0x16ce, Src: 0x0000
 134 14:27:56.437029  0x0000   Broadcast    ZigBee       Route Request, Dst: 0x16ce, Src: 0x0000
 135 14:27:56.577272  0x0000   Broadcast    ZigBee       Route Request, Dst: 0x16ce, Src: 0x0000
 136 14:27:56.839115  0x0000   Broadcast    ZigBee       Route Request, Dst: 0x16ce, Src: 0x0000
 137 14:28:01.412662  0x472b   Broadcast    ZigBee       Link Status
 138 14:28:03.882068                        IEEE 802.15.4 Ack
 139 14:28:03.892716  0x0000   0x472b       ZigBee       Ack, Dst Endpt: 1, Src Endpt: 24
 140 14:28:03.893189                        IEEE 802.15.4 Ack
 141 14:28:06.517191  0x0000   Broadcast    ZigBee       Route Request, Dst: 0x16ce, Src: 0x0000
 142 14:28:06.530692  0x0000   Broadcast    ZigBee       Route Request, Dst: 0x16ce, Src: 0x0000
 143 14:28:06.576069  0x0000   Broadcast    ZigBee       Route Request, Dst: 0x16ce, Src: 0x0000
 144 14:28:06.679542  0x0000   Broadcast    ZigBee       Route Request, Dst: 0x16ce, Src: 0x0000
 145 14:28:06.785425  0x0000   Broadcast    ZigBee       Route Request, Dst: 0x16ce, Src: 0x0000
 146 14:28:07.051796  0x0000   Broadcast    ZigBee       Route Request, Dst: 0x16ce, Src: 0x0000
 147 14:28:08.893942  0x509d   Broadcast    ZigBee       Link Status
 148 14:28:08.909546                        IEEE 802.15.4 Ack
 149 14:28:08.913886  0x675c   0x472b       ZigBee       Ack, Dst Endpt: 1, Src Endpt: 23
```

```
▷ Frame 17: 118 bytes on wire (944 bits), 118 bytes captured (944 bits)
▷ Ethernet II, Src: UbisysTe_00:01:84 (00:1f:ee:00:01:84), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
▷ Internet Protocol Version 4, Src: 0.0.0.0 (0.0.0.0), Dst: 255.255.255.255 (255.255.255.255)
▷ User Datagram Protocol, Src Port: zep (17754), Dst Port: zep (17754)
▷ ZigBee Encapsulation Protocol, Channel: 26, Length: 44
▷ IEEE 802.15.4 Data, Dst: Broadcast, Src: 0x675c
▷ ZigBee Network Layer Command, Dst: Broadcast, Src: 0x675c
```

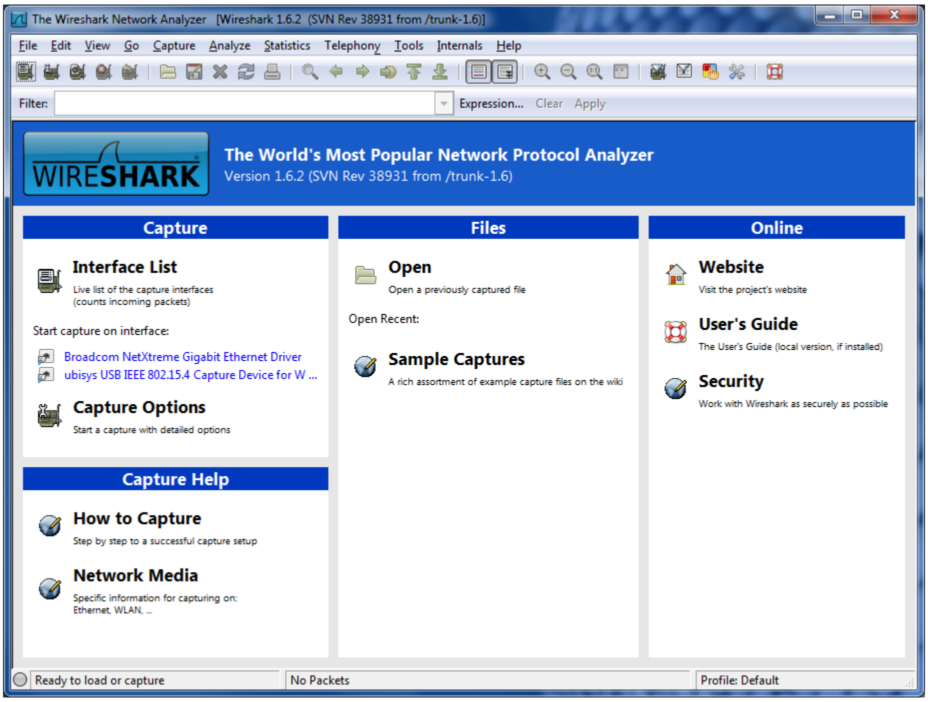
```
0000  ff ff ff ff ff ff 00 1f  ee 00 01 84 08 00 45 00   ........ ......E.
0010  00 68 32 e7 00 00 80 11  07 9f 00 00 00 00 ff ff   .h2..... ........
0020  ff ff 45 5a 45 5a 00 54  00 00 45 58 02 01 1a ff   ..EZEZ.T ..EX....
0030  fe 00 76 d2 aa 53 28 a0  51 43 af 00 00 32 e6 00   ..v..S(. QC...2..
```

eth2: <live capture in progress> File: /t Packets: 149 Displayed: 149 Marked: 0 Profile: Default

Figure 5: Wireshark Capturing ZigBee Traffic on a PowerMac G5 Running Debian Linux 6.0.6 for PowerPC 64-bit

8. Using Wireshark for IEEE 802.15.4 Captures (Quick Introduction)

Start Wireshark to begin a new live-capture.


The Wireshark Network Analyzer [Wireshark 1.6.2 (SVN Rev 38931 from /trunk-1.6)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply
WIRESHARK
Capture
Files
Online
Capture Help
Ready to load or capture
No Packets
Profile: Default

The World's Most Popular Network Protocol Analyzer
Version 1.6.2 (SVN Rev 38931 from /trunk-1.6)

Interface List
Live list of the capture interfaces (counts incoming packets)

Start capture on interface:

- Broadcom NetXtreme Gigabit Ethernet Driver
- ubisys USB IEEE 802.15.4 Capture Device for W ...

Capture Options
Start a capture with detailed options

How to Capture
Step by step to a successful capture setup

Network Media
Specific information for capturing on: Ethernet, WLAN, ...

Open
Open a previously captured file

Open Recent:

Sample Captures
A rich assortment of example capture files on the wiki

Website
Visit the project's website

User's Guide
The User's Guide (local version, if installed)

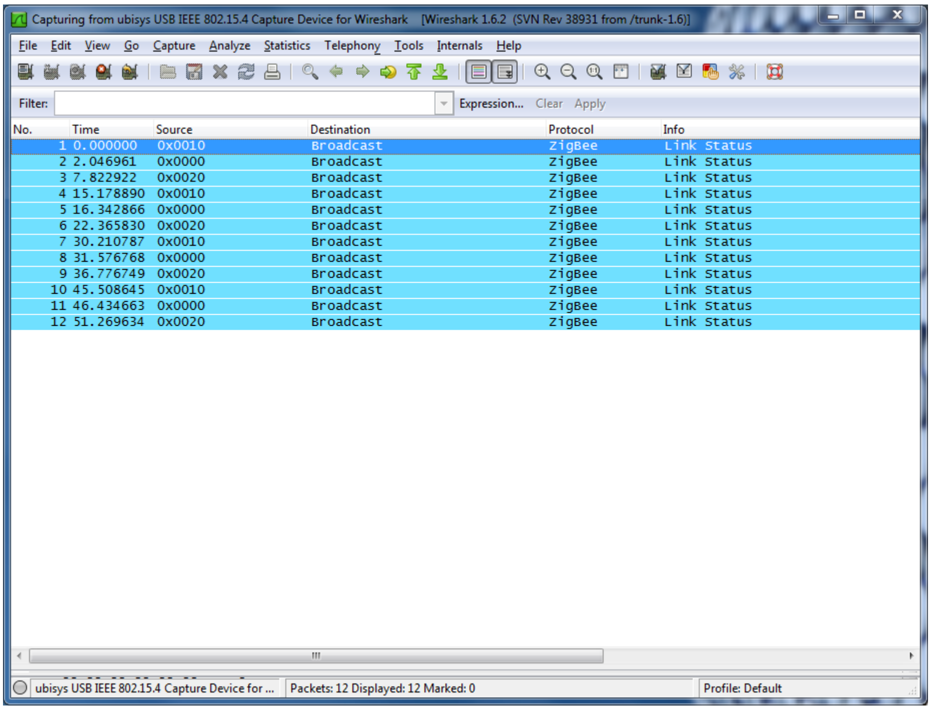
Security
Work with Wireshark as securely as possible

Figure 6: Wireshark Welcome Screen

Notice that the interface list includes the ubisys IEEE 802.15.4 capture device. If it does not appear on your system and you recently installed the driver, please restart the packet capture driver (Winpcap), first. You can either restart your computer or terminate Wireshark and then run the following commands from a command prompt with elevated user access rights (run as administrator):

```
C:\WINDOWS\system32>net stop npf
C:\WINDOWS\system32>net start npf
```

Click on the ubisys USB IEEE 802.15.4 Capture Device for Wireshark item to begin a new live-capture. The welcome screen disappears and a capture log appears.

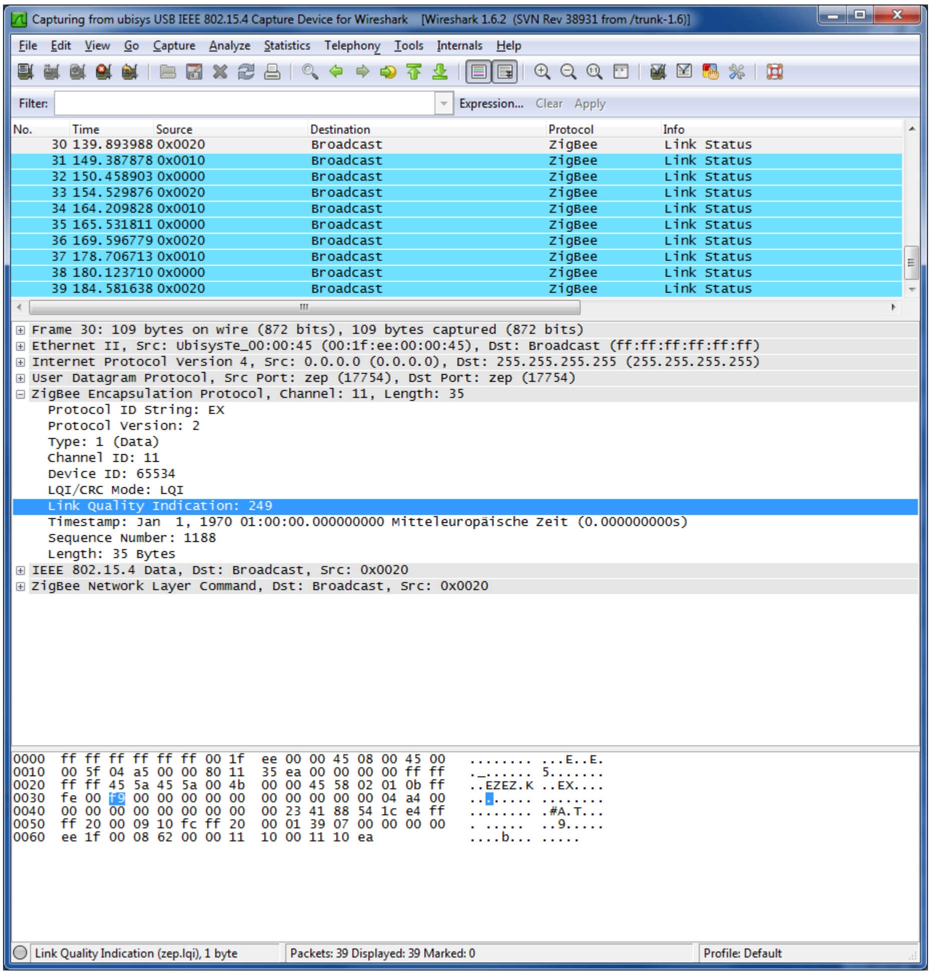

Capturing from ubisys USB IEEE 802.15.4 Capture Device for Wireshark [Wireshark 1.6.2 (SVN Rev 38931 from /trunk-1.6)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Info
1	0.000000	0x0010	Broadcast	ZigBee	Link Status
2	2.046961	0x0000	Broadcast	ZigBee	Link Status
3	7.822922	0x0020	Broadcast	ZigBee	Link Status
4	15.178890	0x0010	Broadcast	ZigBee	Link Status
5	16.342866	0x0000	Broadcast	ZigBee	Link Status
6	22.365830	0x0020	Broadcast	ZigBee	Link Status
7	30.210787	0x0010	Broadcast	ZigBee	Link Status
8	31.576768	0x0000	Broadcast	ZigBee	Link Status
9	36.776749	0x0020	Broadcast	ZigBee	Link Status
10	45.508645	0x0010	Broadcast	ZigBee	Link Status
11	46.434663	0x0000	Broadcast	ZigBee	Link Status
12	51.269634	0x0020	Broadcast	ZigBee	Link Status

ubisys USB IEEE 802.15.4 Capture Device for ... | Packets: 12 Displayed: 12 Marked: 0 | Profile: Default

Figure 7: Wireshark Live-Capture in Progress…

You can select any of the captured packets while the live-capture is still in progress. Two detail sections are available with decoded information as well as raw binary data:



Capturing from ubisys USB IEEE 802.15.4 Capture Device for Wireshark [Wireshark 1.6.2 (SVN Rev 38931 from /trunk-1.6)]
File
Edit
View
Go
Capture
Analyze
Statistics
Telephony
Tools
Internals
Help
Filter:
Expression...
Clear
Apply

```
No.   Time       Source      Destination                Protocol   Info
   30 139.893988 0x0020      Broadcast                  ZigBee     Link Status
   31 149.387878 0x0010      Broadcast                  ZigBee     Link Status
   32 150.458903 0x0000      Broadcast                  ZigBee     Link Status
   33 154.529876 0x0020      Broadcast                  ZigBee     Link Status
   34 164.209828 0x0010      Broadcast                  ZigBee     Link Status
   35 165.531811 0x0000      Broadcast                  ZigBee     Link Status
   36 169.596779 0x0020      Broadcast                  ZigBee     Link Status
   37 178.706713 0x0010      Broadcast                  ZigBee     Link Status
   38 180.123710 0x0000      Broadcast                  ZigBee     Link Status
   39 184.581638 0x0020      Broadcast                  ZigBee     Link Status
```

```
⊞ Frame 30: 109 bytes on wire (872 bits), 109 bytes captured (872 bits)
⊞ Ethernet II, Src: UbisysTe_00:00:45 (00:1f:ee:00:00:45), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
⊞ Internet Protocol Version 4, Src: 0.0.0.0 (0.0.0.0), Dst: 255.255.255.255 (255.255.255.255)
⊞ User Datagram Protocol, Src Port: zep (17754), Dst Port: zep (17754)
⊟ ZigBee Encapsulation Protocol, Channel: 11, Length: 35
    Protocol ID String: EX
    Protocol Version: 2
    Type: 1 (Data)
    Channel ID: 11
    Device ID: 65534
    LQI/CRC Mode: LQI
    Link Quality Indication: 249
    Timestamp: Jan  1, 1970 01:00:00.000000000 Mitteleuropäische Zeit (0.000000000s)
    Sequence Number: 1188
    Length: 35 Bytes
⊞ IEEE 802.15.4 Data, Dst: Broadcast, Src: 0x0020
⊞ ZigBee Network Layer Command, Dst: Broadcast, Src: 0x0020
```

```
0000  ff ff ff ff ff ff 00 1f  ee 00 00 45 08 00 45 00   ........ ...E..E.
0010  00 5f 04 a5 00 00 80 11  35 ea 00 00 00 00 ff ff   ._...... 5.......
0020  ff ff 45 5a 45 5a 00 4b  00 00 45 58 02 01 0b ff   ..EZEZ.K ..EX....
0030  fe 00 f9 00 00 00 00 00  00 00 00 00 00 04 a4 00   ........ ........
0040  00 00 00 00 00 00 00 00  00 23 41 88 54 1c e4 ff   ........ .#A.T...
0050  ff 20 00 09 10 fc ff 20  00 01 39 07 00 00 00 00   . .....  ..9.....
0060  ee 1f 00 08 62 00 00 11  10 00 11 10 ea            ....b... .....
```

Link Quality Indication (zep.lqi), 1 byte | Packets: 39 Displayed: 39 Marked: 0 | Profile: Default

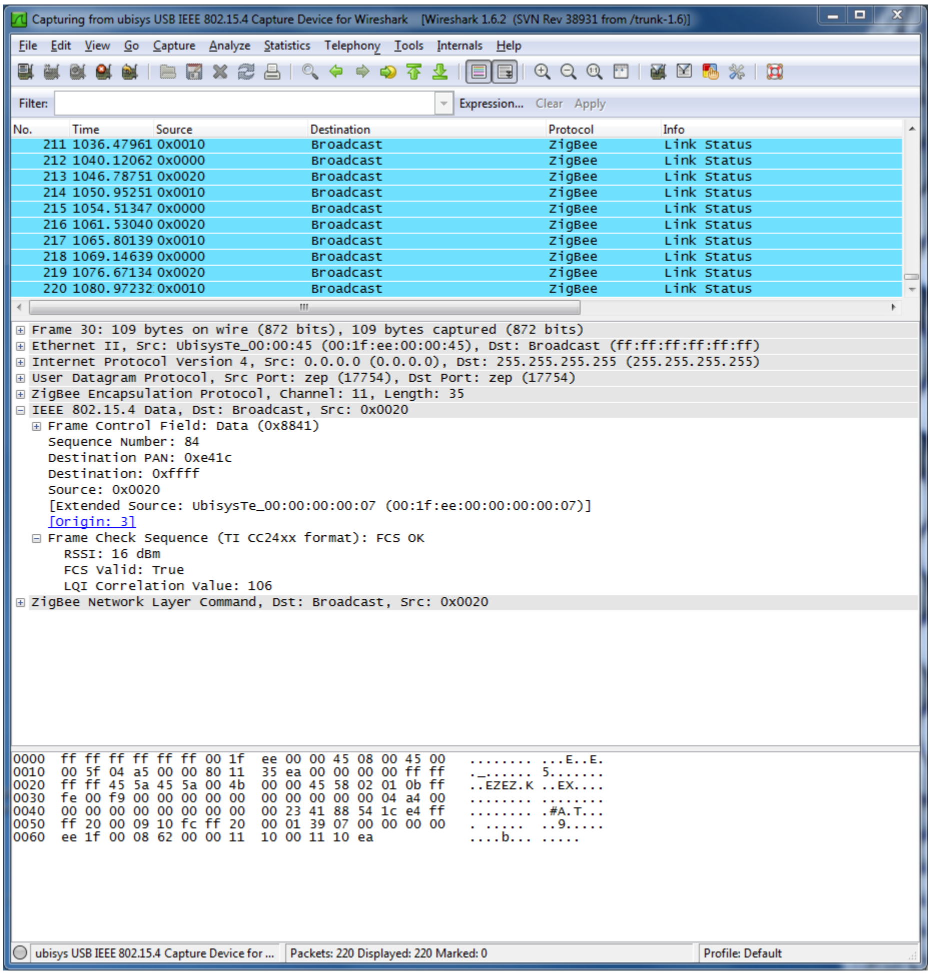
Figure 8: Dissector and Raw Binary Detail Views

Notice that the IEEE 802.15.4 frame is encapsulated in a ZEPv2 frame, which is transferred via UDP/IP, and Ethernet (RNDIS). The ZEP frame includes channel number information and an LQI value derived from individual correlation and RSSI values and a time-stamp[1] that is accurate to approximately one PHY symbol period (16 microseconds).

The time-stamp, despite being specified as an absolute NTP time-stamp, is rather meant as a relative time-stamp, which starts at a fixed time, e.g. January 1, 2012 in firmware revision 1.03 and below, or January 1, 2015, 00:00h in firmware revision 1.04, when the device is powered. The underlying hardware timer hosts a 32-bit register incrementing at a rate of 62.500Hz (corresponding to the PHY's symbol frequency), which results in a roll-over after approximately 19 hours of operation, when the time-stamp restarts at its pre-configured absolute start-time again.

Individual RSSI and LQI correlation values are available in the FCS field. Notice that this field is in CC2420 format, i.e. the frame check sequence is not the value actually transmitted over the air. Instead of the 16-bit CRC, there is only one bit that determines whether the FCS was correct. The remaining 15 bits are used to encode the output of the receiver's symbol correlation output and the RSSI value.

[1] The time-stamp is valid in firmware versions 1.04 and above



Capturing from ubisys USB IEEE 802.15.4 Capture Device for Wireshark [Wireshark 1.6.2 (SVN Rev 38931 from /trunk-1.6)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply

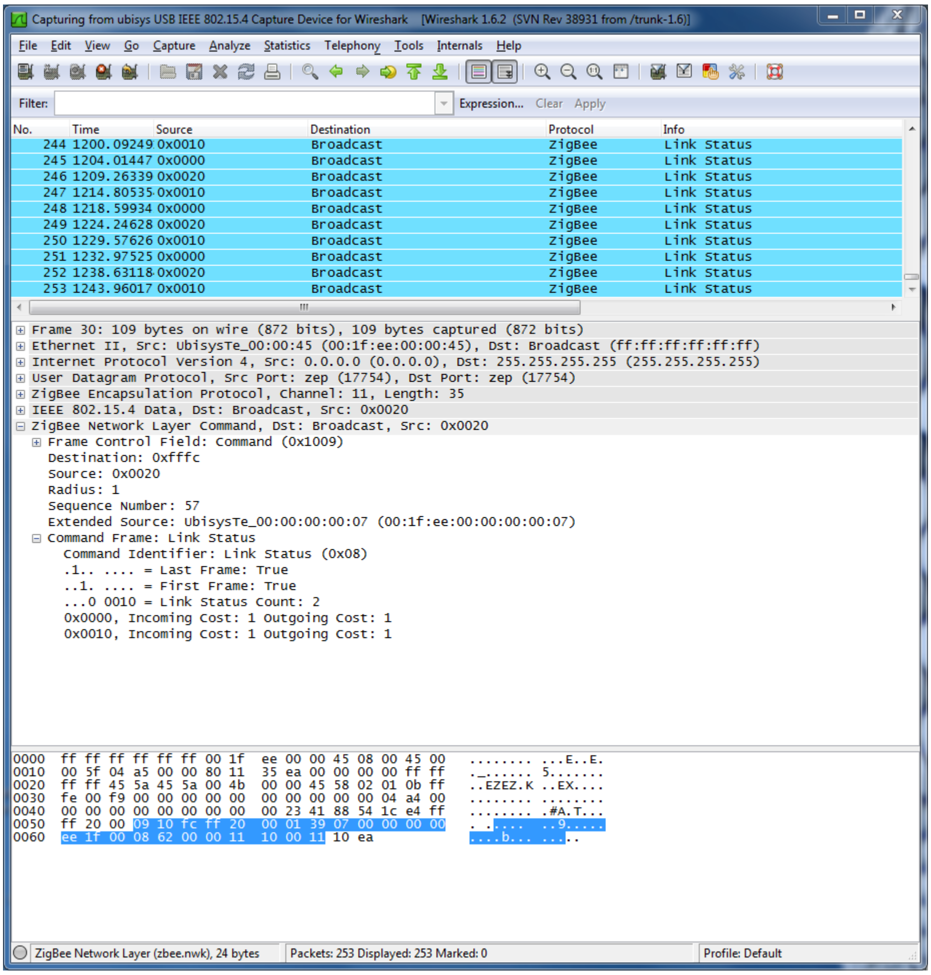
No.	Time	Source	Destination	Protocol	Info
211	1036.47961	0x0010	Broadcast	ZigBee	Link Status
212	1040.12062	0x0000	Broadcast	ZigBee	Link Status
213	1046.78751	0x0020	Broadcast	ZigBee	Link Status
214	1050.95251	0x0010	Broadcast	ZigBee	Link Status
215	1054.51347	0x0000	Broadcast	ZigBee	Link Status
216	1061.53040	0x0020	Broadcast	ZigBee	Link Status
217	1065.80139	0x0010	Broadcast	ZigBee	Link Status
218	1069.14639	0x0000	Broadcast	ZigBee	Link Status
219	1076.67134	0x0020	Broadcast	ZigBee	Link Status
220	1080.97232	0x0010	Broadcast	ZigBee	Link Status

```
⊞ Frame 30: 109 bytes on wire (872 bits), 109 bytes captured (872 bits)
⊞ Ethernet II, Src: UbisysTe_00:00:45 (00:1f:ee:00:00:45), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
⊞ Internet Protocol Version 4, Src: 0.0.0.0 (0.0.0.0), Dst: 255.255.255.255 (255.255.255.255)
⊞ User Datagram Protocol, Src Port: zep (17754), Dst Port: zep (17754)
⊞ ZigBee Encapsulation Protocol, Channel: 11, Length: 35
⊟ IEEE 802.15.4 Data, Dst: Broadcast, Src: 0x0020
  ⊞ Frame Control Field: Data (0x8841)
    Sequence Number: 84
    Destination PAN: 0xe41c
    Destination: 0xffff
    Source: 0x0020
    [Extended Source: UbisysTe_00:00:00:00:07 (00:1f:ee:00:00:00:00:07)]
    [Origin: 3]
  ⊟ Frame Check Sequence (TI CC24xx format): FCS OK
      RSSI: 16 dBm
      FCS Valid: True
      LQI Correlation Value: 106
⊞ ZigBee Network Layer Command, Dst: Broadcast, Src: 0x0020
```

```
0000  ff ff ff ff ff ff 00 1f  ee 00 00 45 08 00 45 00   ........ ...E..E.
0010  00 5f 04 a5 00 00 80 11  35 ea 00 00 00 00 ff ff   ._...... 5.......
0020  ff ff 45 5a 45 5a 00 4b  00 00 45 58 02 01 0b ff   ..EZEZ.K ..EX....
0030  fe 00 f9 00 00 00 00 00  00 00 00 00 00 04 a4 00   ........ ........
0040  00 00 00 00 00 00 00 00  00 23 41 88 54 1c e4 ff   ........ .#A.T...
0050  ff 20 00 09 10 fc ff 20  00 01 39 07 00 00 00 00   . .....  ..9.....
0060  ee 1f 00 08 62 00 00 11  10 00 11 10 ea            ....b... .....
```

ubisys USB IEEE 802.15.4 Capture Device for ...
Packets: 220 Displayed: 220 Marked: 0
Profile: Default

Figure 9: Decoded IEEE 802.15.4 Packet with RSSI and Correlation Values


Capturing from ubisys USB IEEE 802.15.4 Capture Device for Wireshark [Wireshark 1.6.2 (SVN Rev 38931 from /trunk-1.6)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply

```
No.    Time       Source   Destination   Protocol  Info
   244 1200.09249 0x0010   Broadcast     ZigBee    Link Status
   245 1204.01447 0x0000   Broadcast     ZigBee    Link Status
   246 1209.26339 0x0020   Broadcast     ZigBee    Link Status
   247 1214.80535 0x0010   Broadcast     ZigBee    Link Status
   248 1218.59934 0x0000   Broadcast     ZigBee    Link Status
   249 1224.24628 0x0020   Broadcast     ZigBee    Link Status
   250 1229.57626 0x0010   Broadcast     ZigBee    Link Status
   251 1232.97525 0x0000   Broadcast     ZigBee    Link Status
   252 1238.63118 0x0020   Broadcast     ZigBee    Link Status
   253 1243.96017 0x0010   Broadcast     ZigBee    Link Status
```

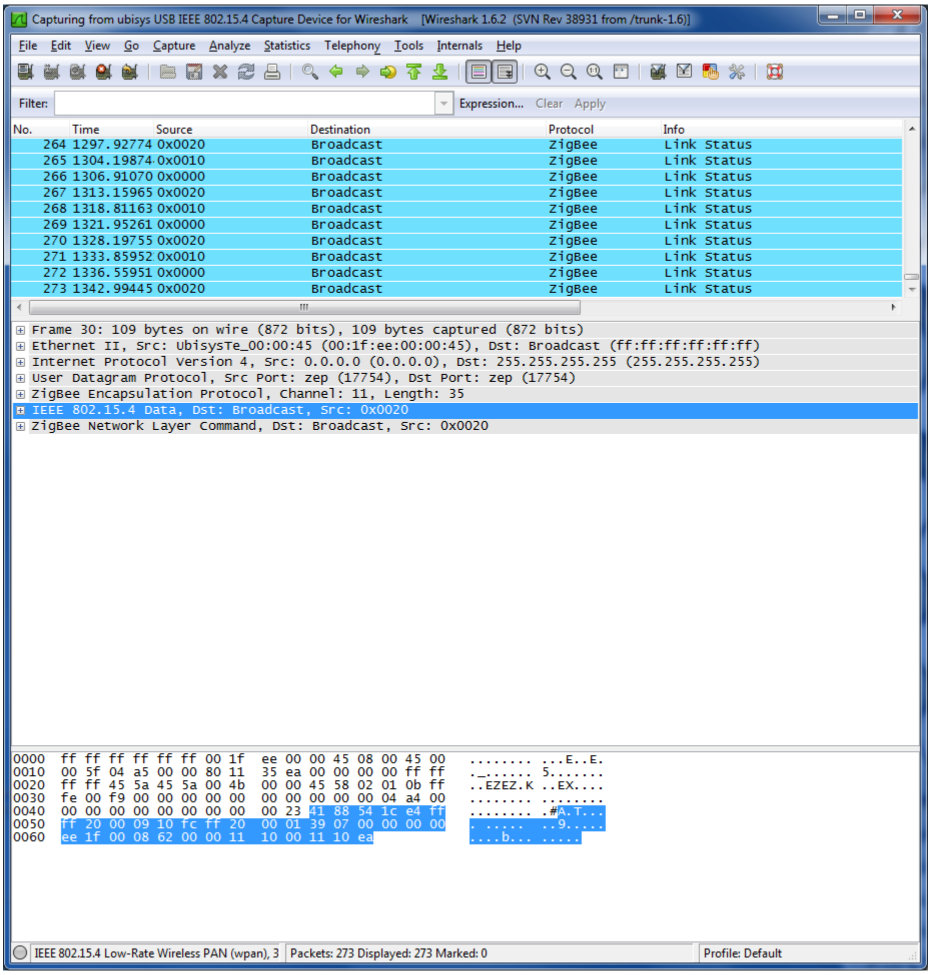
```
⊞ Frame 30: 109 bytes on wire (872 bits), 109 bytes captured (872 bits)
⊞ Ethernet II, Src: UbisysTe_00:00:45 (00:1f:ee:00:00:45), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
⊞ Internet Protocol Version 4, Src: 0.0.0.0 (0.0.0.0), Dst: 255.255.255.255 (255.255.255.255)
⊞ User Datagram Protocol, Src Port: zep (17754), Dst Port: zep (17754)
⊞ ZigBee Encapsulation Protocol, Channel: 11, Length: 35
⊞ IEEE 802.15.4 Data, Dst: Broadcast, Src: 0x0020
⊟ ZigBee Network Layer Command, Dst: Broadcast, Src: 0x0020
  ⊞ Frame Control Field: Command (0x1009)
    Destination: 0xfffc
    Source: 0x0020
    Radius: 1
    Sequence Number: 57
    Extended Source: UbisysTe_00:00:00:00:07 (00:1f:ee:00:00:00:00:07)
  ⊟ Command Frame: Link Status
      Command Identifier: Link Status (0x08)
      .1.. .... = Last Frame: True
      ..1. .... = First Frame: True
      ...0 0010 = Link Status Count: 2
      0x0000, Incoming Cost: 1 Outgoing Cost: 1
      0x0010, Incoming Cost: 1 Outgoing Cost: 1
```

```
0000  ff ff ff ff ff ff 00 1f  ee 00 00 45 08 00 45 00   ........ ...E..E.
0010  00 5f 04 a5 00 00 80 11  35 ea 00 00 00 00 ff ff   ._...... 5.......
0020  ff ff 45 5a 45 5a 00 4b  00 00 45 58 02 01 0b ff   ..EZEZ.K ..EX....
0030  fe 00 f9 00 00 00 00 00  00 00 00 00 00 04 a4 00   ........ ........
0040  00 00 00 00 00 00 00 00  00 23 41 88 54 1c e4 ff   ........ .#A.T...
0050  ff 20 00 09 10 fc ff 20  00 01 39 07 00 00 00 00   . ...... ..9.....
0060  ee 1f 00 08 62 00 00 11  10 00 11 10 ea            ....b... .....
```

```
ZigBee Network Layer (zbee.nwk), 24 bytes    Packets: 253 Displayed: 253 Marked: 0    Profile: Default
```

Figure 10: Example of a ZigBee PRO Link Status Frame

If you wish to examine the raw binary packet data, highlight the “IEEE 802.15.4 Data” line in the dissector view. The raw binary packet data will then be highlighted in the bottom area of the window.


Capturing from ubisys USB IEEE 802.15.4 Capture Device for Wireshark [Wireshark 1.6.2 (SVN Rev 38931 from /trunk-1.6)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply

```
No.     Time        Source   Destination    Protocol   Info
    264 1297.92774  0x0020   Broadcast      ZigBee     Link Status
    265 1304.19874  0x0010   Broadcast      ZigBee     Link Status
    266 1306.91070  0x0000   Broadcast      ZigBee     Link Status
    267 1313.15965  0x0020   Broadcast      ZigBee     Link Status
    268 1318.81163  0x0010   Broadcast      ZigBee     Link Status
    269 1321.95261  0x0000   Broadcast      ZigBee     Link Status
    270 1328.19755  0x0020   Broadcast      ZigBee     Link Status
    271 1333.85952  0x0010   Broadcast      ZigBee     Link Status
    272 1336.55951  0x0000   Broadcast      ZigBee     Link Status
    273 1342.99445  0x0020   Broadcast      ZigBee     Link Status
```

```
⊞ Frame 30: 109 bytes on wire (872 bits), 109 bytes captured (872 bits)
⊞ Ethernet II, Src: UbisysTe_00:00:45 (00:1f:ee:00:00:45), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
⊞ Internet Protocol Version 4, Src: 0.0.0.0 (0.0.0.0), Dst: 255.255.255.255 (255.255.255.255)
⊞ User Datagram Protocol, Src Port: zep (17754), Dst Port: zep (17754)
⊞ ZigBee Encapsulation Protocol, Channel: 11, Length: 35
⊞ IEEE 802.15.4 Data, Dst: Broadcast, Src: 0x0020
⊞ ZigBee Network Layer Command, Dst: Broadcast, Src: 0x0020
```

```
0000  ff ff ff ff ff ff 00 1f  ee 00 00 45 08 00 45 00   ........ ...E..E.
0010  00 5f 04 a5 00 00 80 11  35 ea 00 00 00 00 ff ff   ._...... 5.......
0020  ff ff 45 5a 45 5a 00 4b  00 00 45 58 02 01 0b ff   ..EZEZ.K ..EX....
0030  fe 00 f9 00 00 00 00 00  00 00 00 00 00 04 a4 00   ........ ........
0040  00 00 00 00 00 00 00 00  00 23 41 88 54 1c e4 ff   ........ .#A.T...
0050  ff 20 00 09 10 fc ff 20  00 01 39 07 00 00 00 00   . .....  ..9.....
0060  ee 1f 00 08 62 00 00 11  10 00 11 10 ea            ....b... .....
```

IEEE 802.15.4 Low-Rate Wireless PAN (wpan), 3 | Packets: 273 Displayed: 273 Marked: 0 | Profile: Default

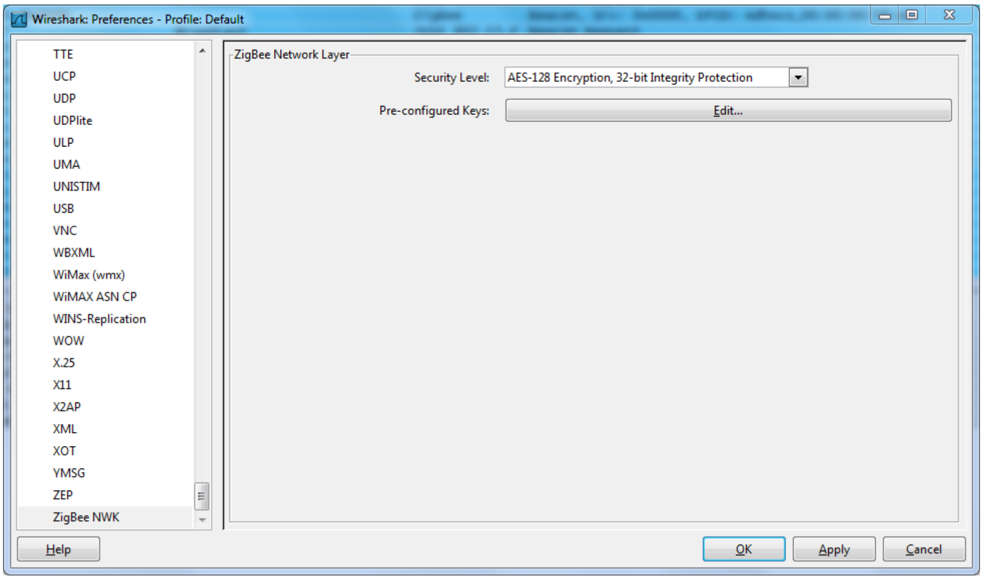
Figure 11: Raw Binary Packet Data

Notice that Wireshark is a powerful tool with various filtering capabilities, capture options etc. Please refer to the Wireshark documentation for a complete coverage of all features, including remote capture, merging capture files, etc.

9. Using Wireshark to Analyze Encrypted ZigBee PRO Traffic

You can use Wireshark to decrypt ZigBee PRO traffic on-the-fly. Both, secure NWK and APS frames, can be dissected, as well as ZigBee Green Power.

To set the AES-CCM* security level according to your particular network setup, open the Preferences for the ZigBee protocol. From the Edit menu, choose Preferences and expand the Protocols section. Locate and highlight “ZigBee NWK”. Select the appropriate security level. For example a ZigBee Home Automation Network is going to use security level 5, which means AES-128 encryption and 32-bit message integrity code.



Wireshark: Preferences - Profile: Default

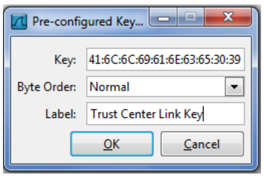
ZigBee Network Layer

Security Level: AES-128 Encryption, 32-bit Integrity Protection

Pre-configured Keys: Edit...

Figure 12: ZigBee NWK Preferences in Wireshark

For a ZigBee home automation network, you may use the default Trust Center link key “ZigBeeAlliance09” = 5A:69:67:42:65:65:41:6C:6C:69:61:6E:63:65:30:39 as long as it has not been changed via commissioning. For distributed security networks (like ZigBee Light Link) use the appropriate[2] global distributed security trust center link key, e.g. for uncertified products use the well-known key D0:D1:D2:D3:D4:D5:D6:D7:D8:D9:DA:DB:DC:DD:DE:DF. For other profiles, refer to the profile’s network security setup. Notice that the label is used to identify which key has been used by Wireshark to decode the frame. Notice that you may enter as many keys as you wish, for example the default Trust Center link-key, the distributed security link-key and any number of pre-configured link-keys (e.g. derived from installation codes) that you require in addition.



Pre-configured Key...

Key: 41:6C:6C:69:61:6E:63:65:30:39

Byte Order: Normal

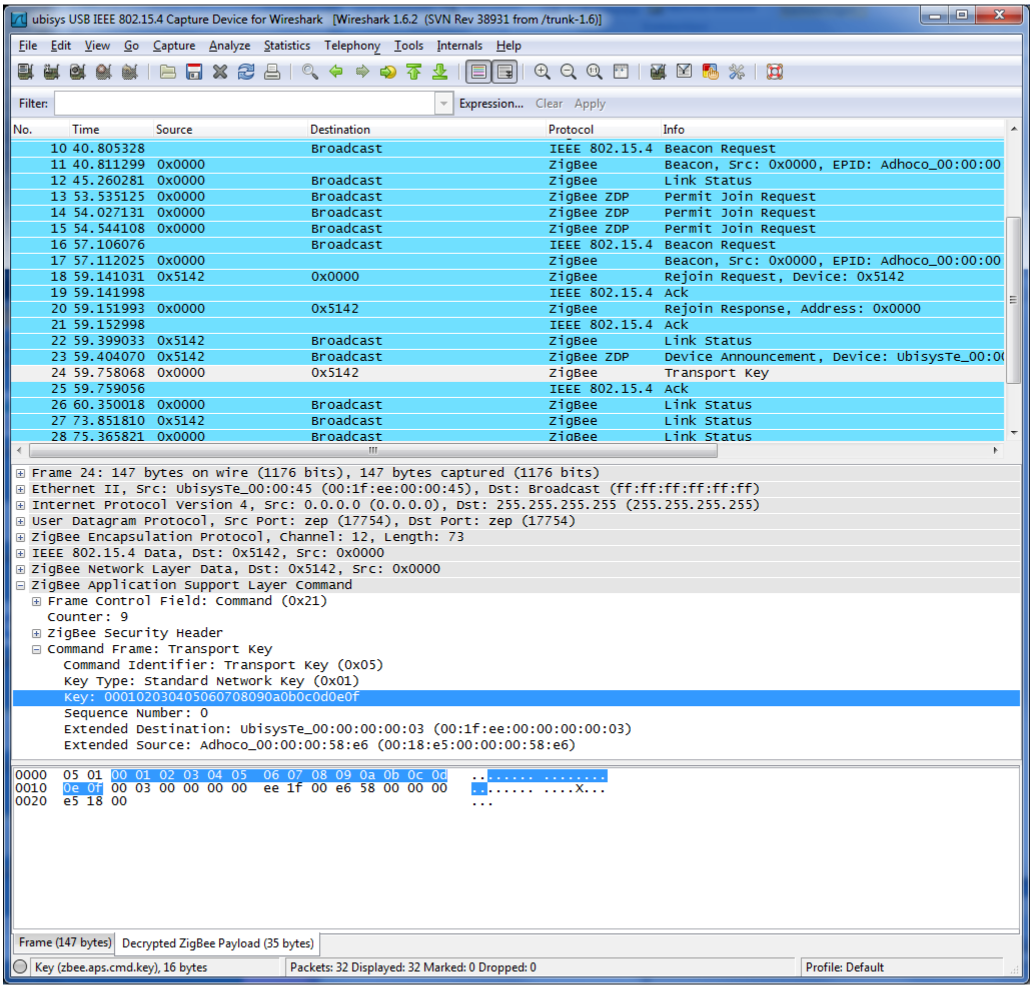
Label: Trust Center Link Key

Figure 13: Entering a ZigBee link or network key

[2] It is not recommended to use the confidential distributed security link key, i.e. the key used in certified devices, unless you are doing so in a secure production facility or laboratory environment for end-product testing – in accordance with all contracts, terms and conditions your company has accepted and signed.

Once you have entered the key, Wireshark is able to decrypt the Transport Key APS command.

Now, open the network for new devices, i.e. permit joining, and let a device join the network to trigger transmission of the transport key command from the trust center to the joining device.


ubisys USB IEEE 802.15.4 Capture Device for Wireshark [Wireshark 1.6.2 (SVN Rev 38931 from /trunk-1.6)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply
Frame (147 bytes)
Decrypted ZigBee Payload (35 bytes)

```
No.  Time         Source   Destination  Protocol      Info
  10 40.805328             Broadcast    IEEE 802.15.4 Beacon Request
  11 40.811299    0x0000                ZigBee        Beacon, Src: 0x0000, EPID: Adhoco_00:00:00
  12 45.260281    0x0000   Broadcast    ZigBee        Link Status
  13 53.535125    0x0000   Broadcast    ZigBee ZDP    Permit Join Request
  14 54.027131    0x0000   Broadcast    ZigBee ZDP    Permit Join Request
  15 54.544108    0x0000   Broadcast    ZigBee ZDP    Permit Join Request
  16 57.106076             Broadcast    IEEE 802.15.4 Beacon Request
  17 57.112025    0x0000                ZigBee        Beacon, Src: 0x0000, EPID: Adhoco_00:00:00
  18 59.141031    0x5142   0x0000       ZigBee        Rejoin Request, Device: 0x5142
  19 59.141998                          IEEE 802.15.4 Ack
  20 59.151993    0x0000   0x5142       ZigBee        Rejoin Response, Address: 0x0000
  21 59.152998                          IEEE 802.15.4 Ack
  22 59.399033    0x5142   Broadcast    ZigBee        Link Status
  23 59.404070    0x5142   Broadcast    ZigBee ZDP    Device Announcement, Device: UbisysTe_00:0(
  24 59.758068    0x0000   0x5142       ZigBee        Transport Key
  25 59.759056                          IEEE 802.15.4 Ack
  26 60.350018    0x0000   Broadcast    ZigBee        Link Status
  27 73.851810    0x5142   Broadcast    ZigBee        Link Status
  28 75.365821    0x0000   Broadcast    ZigBee        Link Status
```

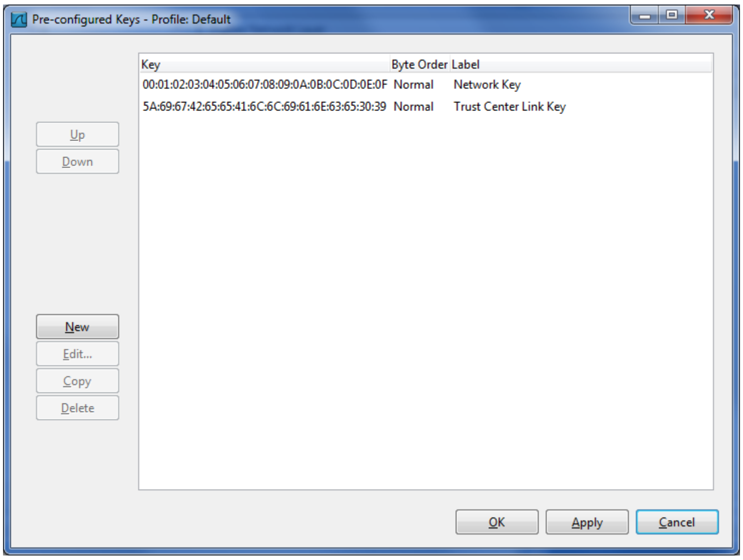
```
⊞ Frame 24: 147 bytes on wire (1176 bits), 147 bytes captured (1176 bits)
⊞ Ethernet II, Src: UbisysTe_00:00:45 (00:1f:ee:00:00:45), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
⊞ Internet Protocol Version 4, Src: 0.0.0.0 (0.0.0.0), Dst: 255.255.255.255 (255.255.255.255)
⊞ User Datagram Protocol, Src Port: zep (17754), Dst Port: zep (17754)
⊞ ZigBee Encapsulation Protocol, Channel: 12, Length: 73
⊞ IEEE 802.15.4 Data, Dst: 0x5142, Src: 0x0000
⊞ ZigBee Network Layer Data, Dst: 0x5142, Src: 0x0000
⊟ ZigBee Application Support Layer Command
  ⊞ Frame Control Field: Command (0x21)
    Counter: 9
  ⊞ ZigBee Security Header
  ⊟ Command Frame: Transport Key
      Command Identifier: Transport Key (0x05)
      Key Type: Standard Network Key (0x01)
      Key: 000102030405060708090a0b0c0d0e0f
      Sequence Number: 0
      Extended Destination: UbisysTe_00:00:00:00:03 (00:1f:ee:00:00:00:00:03)
      Extended Source: Adhoco_00:00:00:58:e6 (00:18:e5:00:00:00:58:e6)
```

```
0000  05 01 00 01 02 03 04 05  06 07 08 09 0a 0b 0c 0d   ........ ........
0010  0e 0f 00 03 00 00 00 00  ee 1f 00 e6 58 00 00 00   ........ ....X...
0020  e5 18 00                                           ...
```

Key (zbee.aps.cmd.key), 16 bytes | Packets: 32 Displayed: 32 Marked: 0 Dropped: 0 | Profile: Default

Figure 14: Transport Key Command in ZigBee PRO Home Automation

Check the contents of the Transport Key command to obtain the current network key. In the present example, the standard network key is 000102030405060708090a0b0c0d0e0f and can be added to the pre-configured keys just like the Trust Center link key. Depending on the version of Wireshark you are using, the software is also capable of learning the key automatically.

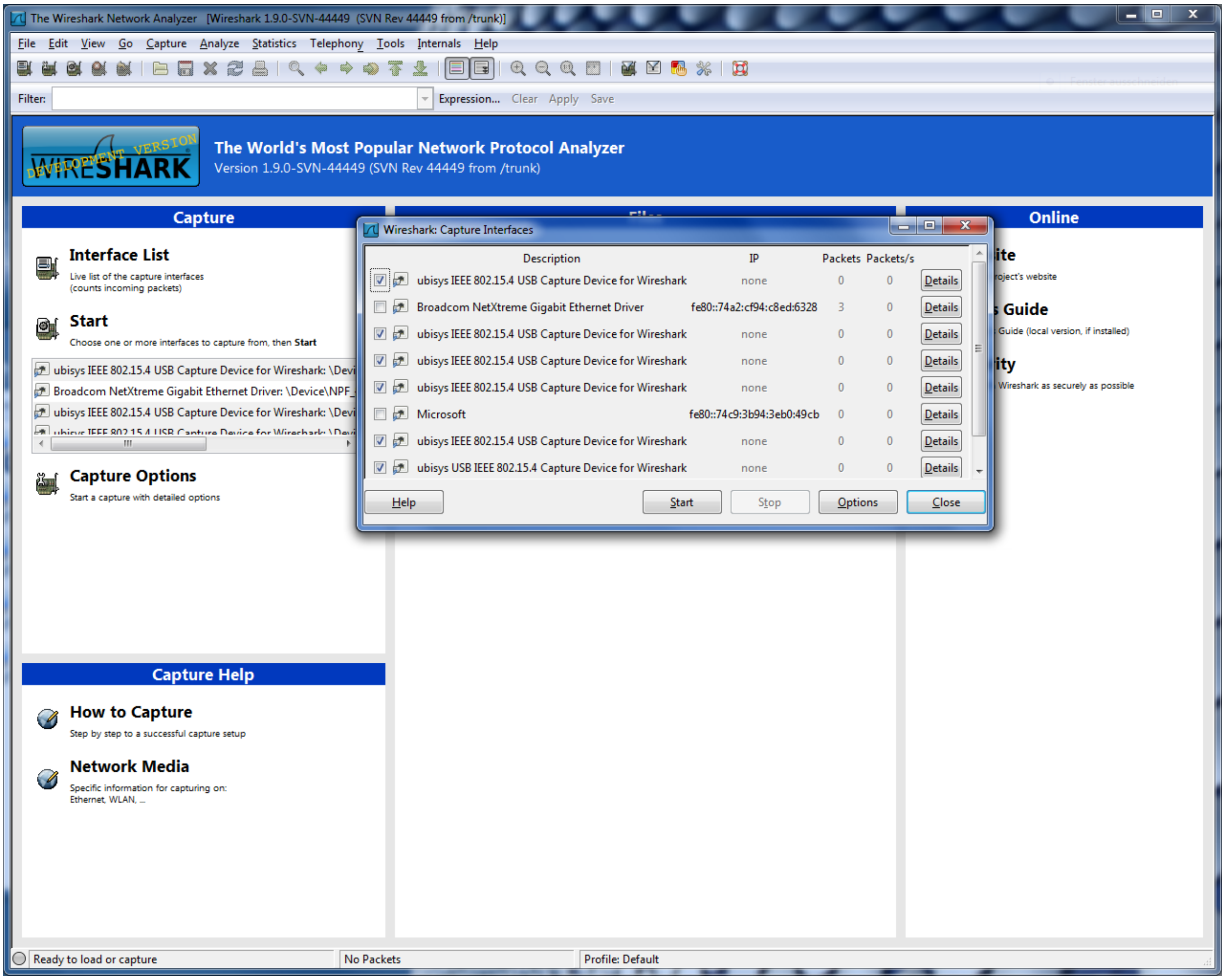


Key	Byte Order	Label
00:01:02:03:04:05:06:07:08:09:0A:0B:0C:0D:0E:0F	Normal	Network Key
5A:69:67:42:65:65:41:6C:6C:69:61:6E:63:65:30:39	Normal	Trust Center Link Key

Figure 15: Wireshark ZigBee pre-configured keys

It is possible to capture traffic on multiple channels simultaneously. This might be necessary to observe frequency hopping and frequency-agile systems. For each channel, a distinct ubisys IEEE 802.15.4 stick with Wireshark Firmware is required. Thus, to cover all channels in the 2.4GHz band, sixteen sticks are required, which can be ordered as a bundle. Additional sticks are also beneficial to mitigate the effects of multipath fading in indoor environments. In this case tune more than one stick to the same channel.

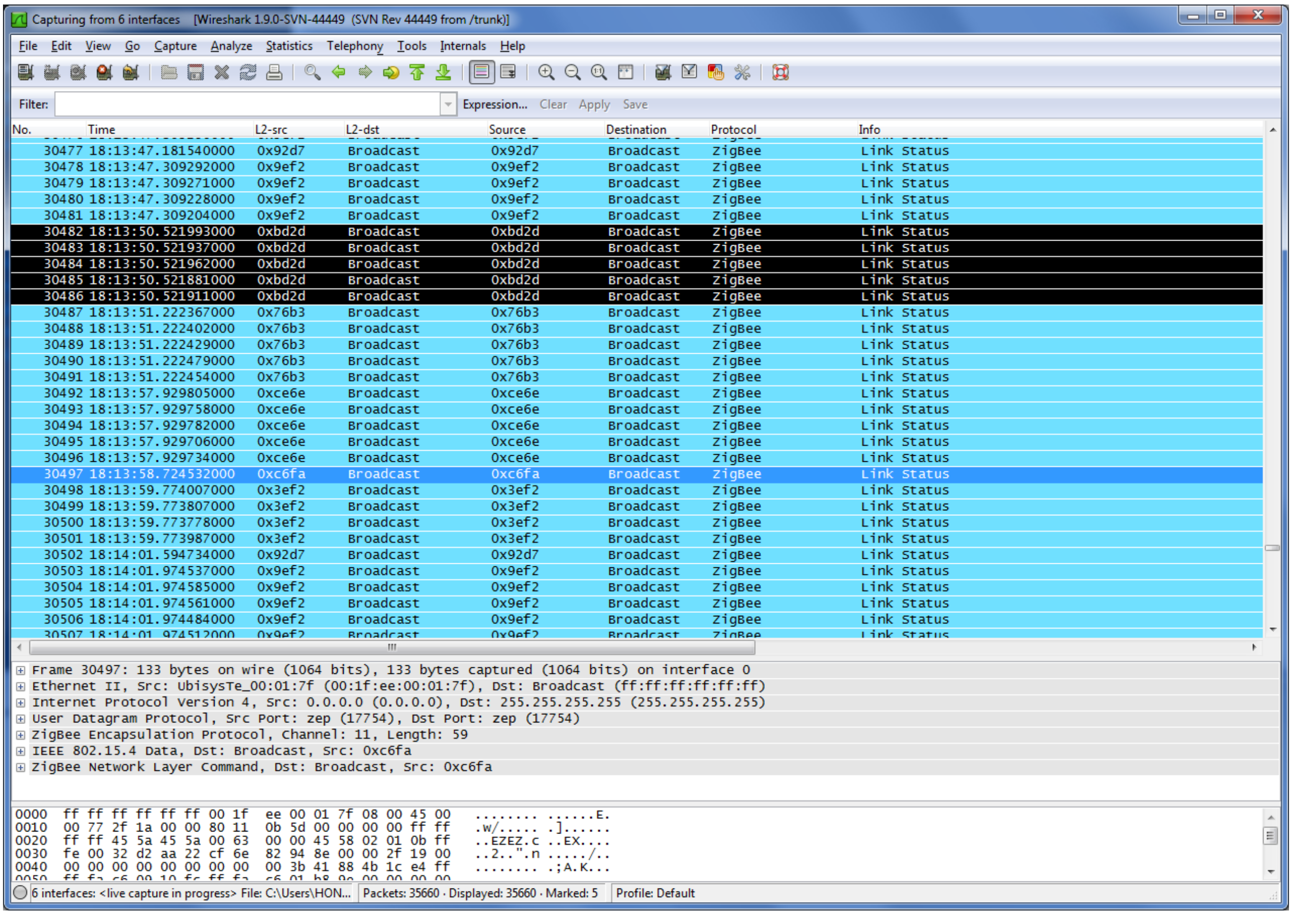
While you could use multiple instances of Wireshark in order to run multiple captures and then merge the captures files, it is often more convenient to group all sticks and run a single capture (on all sticks).


The Wireshark Network Analyzer [Wireshark 1.9.0-SVN-44449 (SVN Rev 44449 from /trunk)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply Save
DEVELOPMENT VERSION
WIRESHARK
The World's Most Popular Network Protocol Analyzer
Version 1.9.0-SVN-44449 (SVN Rev 44449 from /trunk)
Capture
Interface List
Live list of the capture interfaces (counts incoming packets)
Start
Choose one or more interfaces to capture from, then Start
ubisys IEEE 802.15.4 USB Capture Device for Wireshark: \Devi
Broadcom NetXtreme Gigabit Ethernet Driver: \Device\NPF_
ubisys IEEE 802.15.4 USB Capture Device for Wireshark: \Devi
Capture Options
Start a capture with detailed options
Capture Help
How to Capture
Step by step to a successful capture setup
Network Media
Specific information for capturing on: Ethernet, WLAN, ...
Online
Website
project's website
User's Guide
Guide (local version, if installed)
Security
Wireshark as securely as possible
Ready to load or capture
No Packets
Profile: Default

Wireshark: Capture Interfaces

		Description	IP	Packets	Packets/s	
☑		ubisys IEEE 802.15.4 USB Capture Device for Wireshark	none	0	0	Details
☐		Broadcom NetXtreme Gigabit Ethernet Driver	fe80::74a2:cf94:c8ed:6328	3	0	Details
☑		ubisys IEEE 802.15.4 USB Capture Device for Wireshark	none	0	0	Details
☑		ubisys IEEE 802.15.4 USB Capture Device for Wireshark	none	0	0	Details
☑		ubisys IEEE 802.15.4 USB Capture Device for Wireshark	none	0	0	Details
☐		Microsoft	fe80::74c9:3b94:3eb0:49cb	0	0	Details
☑		ubisys IEEE 802.15.4 USB Capture Device for Wireshark	none	0	0	Details
☑		ubisys USB IEEE 802.15.4 Capture Device for Wireshark	none	0	0	Details

Figure 16: Selecting multiple interfaces for capture in Wireshark (here, five ubisys IEEE 802.15.4 USB Sticks)


Capturing from 6 interfaces [Wireshark 1.9.0-SVN-44449 (SVN Rev 44449 from /trunk)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply Save

```
No.   Time                L2-src   L2-dst      Source   Destination  Protocol  Info
30477 18:13:47.181540000  0x92d7   Broadcast   0x92d7   Broadcast    ZigBee    Link Status
30478 18:13:47.309292000  0x9ef2   Broadcast   0x9ef2   Broadcast    ZigBee    Link Status
30479 18:13:47.309271000  0x9ef2   Broadcast   0x9ef2   Broadcast    ZigBee    Link Status
30480 18:13:47.309228000  0x9ef2   Broadcast   0x9ef2   Broadcast    ZigBee    Link Status
30481 18:13:47.309204000  0x9ef2   Broadcast   0x9ef2   Broadcast    ZigBee    Link Status
30482 18:13:50.521993000  0xbd2d   Broadcast   0xbd2d   Broadcast    ZigBee    Link Status
30483 18:13:50.521937000  0xbd2d   Broadcast   0xbd2d   Broadcast    ZigBee    Link Status
30484 18:13:50.521962000  0xbd2d   Broadcast   0xbd2d   Broadcast    ZigBee    Link Status
30485 18:13:50.521881000  0xbd2d   Broadcast   0xbd2d   Broadcast    ZigBee    Link Status
30486 18:13:50.521911000  0xbd2d   Broadcast   0xbd2d   Broadcast    ZigBee    Link Status
30487 18:13:51.222367000  0x76b3   Broadcast   0x76b3   Broadcast    ZigBee    Link Status
30488 18:13:51.222402000  0x76b3   Broadcast   0x76b3   Broadcast    ZigBee    Link Status
30489 18:13:51.222429000  0x76b3   Broadcast   0x76b3   Broadcast    ZigBee    Link Status
30490 18:13:51.222479000  0x76b3   Broadcast   0x76b3   Broadcast    ZigBee    Link Status
30491 18:13:51.222454000  0x76b3   Broadcast   0x76b3   Broadcast    ZigBee    Link Status
30492 18:13:57.929805000  0xce6e   Broadcast   0xce6e   Broadcast    ZigBee    Link Status
30493 18:13:57.929758000  0xce6e   Broadcast   0xce6e   Broadcast    ZigBee    Link Status
30494 18:13:57.929782000  0xce6e   Broadcast   0xce6e   Broadcast    ZigBee    Link Status
30495 18:13:57.929706000  0xce6e   Broadcast   0xce6e   Broadcast    ZigBee    Link Status
30496 18:13:57.929734000  0xce6e   Broadcast   0xce6e   Broadcast    ZigBee    Link Status
30497 18:13:58.724532000  0xc6fa   Broadcast   0xc6fa   Broadcast    ZigBee    Link Status
30498 18:13:59.774007000  0x3ef2   Broadcast   0x3ef2   Broadcast    ZigBee    Link Status
30499 18:13:59.773807000  0x3ef2   Broadcast   0x3ef2   Broadcast    ZigBee    Link Status
30500 18:13:59.773778000  0x3ef2   Broadcast   0x3ef2   Broadcast    ZigBee    Link Status
30501 18:13:59.773987000  0x3ef2   Broadcast   0x3ef2   Broadcast    ZigBee    Link Status
30502 18:14:01.594734000  0x92d7   Broadcast   0x92d7   Broadcast    ZigBee    Link Status
30503 18:14:01.974537000  0x9ef2   Broadcast   0x9ef2   Broadcast    ZigBee    Link Status
30504 18:14:01.974585000  0x9ef2   Broadcast   0x9ef2   Broadcast    ZigBee    Link Status
30505 18:14:01.974561000  0x9ef2   Broadcast   0x9ef2   Broadcast    ZigBee    Link Status
30506 18:14:01.974484000  0x9ef2   Broadcast   0x9ef2   Broadcast    ZigBee    Link Status
30507 18:14:01.974512000  0x9ef2   Broadcast   0x9ef2   Broadcast    ZigBee    Link Status
```

```
⊞ Frame 30497: 133 bytes on wire (1064 bits), 133 bytes captured (1064 bits) on interface 0
⊞ Ethernet II, Src: UbisysTe_00:01:7f (00:1f:ee:00:01:7f), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
⊞ Internet Protocol Version 4, Src: 0.0.0.0 (0.0.0.0), Dst: 255.255.255.255 (255.255.255.255)
⊞ User Datagram Protocol, Src Port: zep (17754), Dst Port: zep (17754)
⊞ ZigBee Encapsulation Protocol, Channel: 11, Length: 59
⊞ IEEE 802.15.4 Data, Dst: Broadcast, Src: 0xc6fa
⊞ ZigBee Network Layer Command, Dst: Broadcast, Src: 0xc6fa
```

```
0000  ff ff ff ff ff ff 00 1f  ee 00 01 7f 08 00 45 00   ........ ......E.
0010  00 77 2f 1a 00 00 80 11  0b 5d 00 00 00 00 ff ff   .w/..... .]......
0020  ff ff 45 5a 45 5a 00 63  00 00 45 58 02 01 0b ff   ..EZEZ.c ..EX....
0030  fe 00 32 d2 aa 22 cf 6e  82 94 8e 00 00 2f 19 00   ..2..".n ...../..
0040  00 00 00 00 00 00 00 00  00 3b 41 88 4b 1c e4 ff   ........ .;A.K...
```

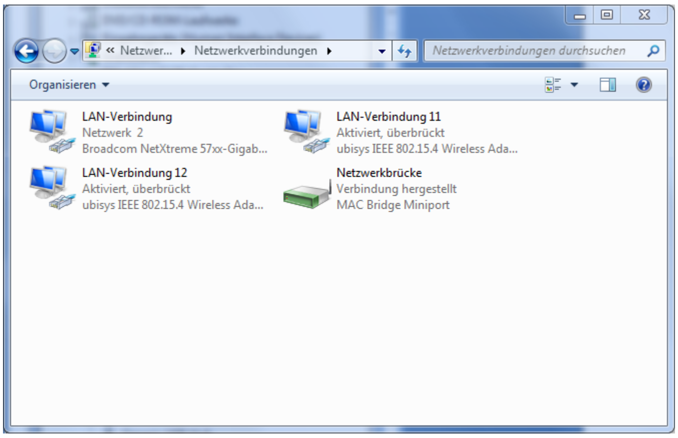
```
6 interfaces: <live capture in progress> File: C:\Users\HON...   Packets: 35660 · Displayed: 35660 · Marked: 5   Profile: Default
```

Figure 17: Diversity capture with five Sticks. Certain nodes (e.g. 0xc6fa), only received by one out of five sticks.

The figure above shows the benefit of diversity capture. The tagged group of link status messages from node 0xbd2d is received by all sticks concurrently, as expected. But only one out of five sticks was able to receive a message from node 0xc6fa at the border of wireless range.

Notice: Simultaneous capture on multiple interfaces is inherently supported by later versions of Wireshark as shown above. The information below remains available for reference, only.

This can be achieved with Windows built-in network bridge feature. Open the network connections view and select the ubisys IEEE 802.15.4 adapters you want to group. Right-click on one of them and select the Bridge Connections command from the context menu that appears. This will create a network bridge.


« Netzwer... ▸ Netzwerkverbindungen ▸
Netzwerkverbindungen durchsuchen
Organisieren ▾

LAN-Verbindung
Netzwerk 2
Broadcom NetXtreme 57xx-Gigab...

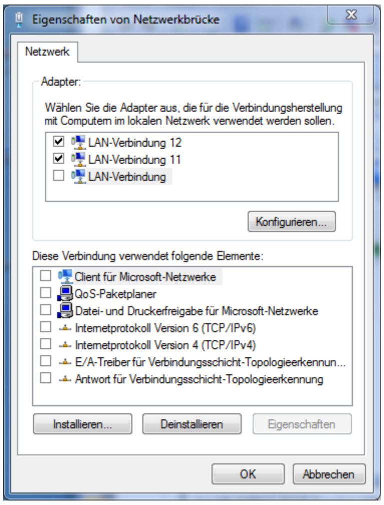
LAN-Verbindung 11
Aktiviert, überbrückt
ubisys IEEE 802.15.4 Wireless Ada...

LAN-Verbindung 12
Aktiviert, überbrückt
ubisys IEEE 802.15.4 Wireless Ada...

Netzwerkbrücke
Verbindung hergestellt
MAC Bridge Miniport

Figure 18: Network Bridge for Diversity or Multiple-Channel Capture

Edit the properties for the new connection and disable all protocols, as you have done for all the individual ubisys IEEE 802.15.4 adapters. You can add and remove other adapters from the group of bridged devices by adding or removing the check mark in the adapter selection area.



Adapter:

Wählen Sie die Adapter aus, die für die Verbindungsherstellung mit Computern im lokalen Netzwerk verwendet werden sollen.

- ☑ LAN-Verbindung 12
- ☑ LAN-Verbindung 11
- ☐ LAN-Verbindung

Diese Verbindung verwendet folgende Elemente:

- ☐ Client für Microsoft-Netzwerke
- ☐ QoS-Paketplaner
- ☐ Datei- und Druckerfreigabe für Microsoft-Netzwerke
- ☐ Internetprotokoll Version 6 (TCP/IPv6)
- ☐ Internetprotokoll Version 4 (TCP/IPv4)
- ☐ E/A-Treiber für Verbindungsschicht-Topologieerkennun...
- ☐ Antwort für Verbindungsschicht-Topologieerkennung

Figure 19: Network Bridge Properties

11. Troubleshooting

If you encounter problems acquiring packets, then walk through the following checklist. If you don't manage to solve the problem, feel free to contact ubisys support.

- Make sure that the driver is properly installed
- Make sure that you have disabled all networking protocols
- Make sure that the ubisys 802.15.4 networking adapter is enabled and has not been disabled by Windows' network discovery algorithm
- Make sure that no enterprise security software, firewall or anti-virus program blocks the network adapter. Contact your IT department if you are uncertain. Some of these applications require the IT administrator to authorize new hardware, in particular network adapters, before they are allowed to operate normally
- Make sure that there is actually wireless traffic on the channel that you have selected
- If you are trying to capture data from a single transmitter, you might be in a dead-spot (unlikely, but still)
- Make sure that there is no interference that prevents the sniffer from receiving data
- Unplug the USB stick and plug it in again, then restart packet acquisition

11.1. Known issues with firmware versions 1.04 – 1.00

Problem: When the host computer is put into sleep mode while the device is connect, this may cause a blue-screen in the Windows RNDIS driver a few seconds (about 15 to 30) after the computer is awake again.
Work-around: Detach the USB stick before entering sleep mode or before waking the computer. This will be fixed in a future firmware release.

11.2. Known issues with firmware versions 1.03 – 1.00

Problem: The time-stamp in the ZEPv2 frame is either fixed at zero or not always maintained correctly.
Solution: Please upgrade to firmware revision 1.04 or above.

Problem: Occasionally the capture could get stuck. The statistics show incoming traffic (the received packet count increases), but the incoming frames are not delivered to the host PC.
Solution: Please upgrade to firmware revision 1.04 or above.

12. Physical Dimensions

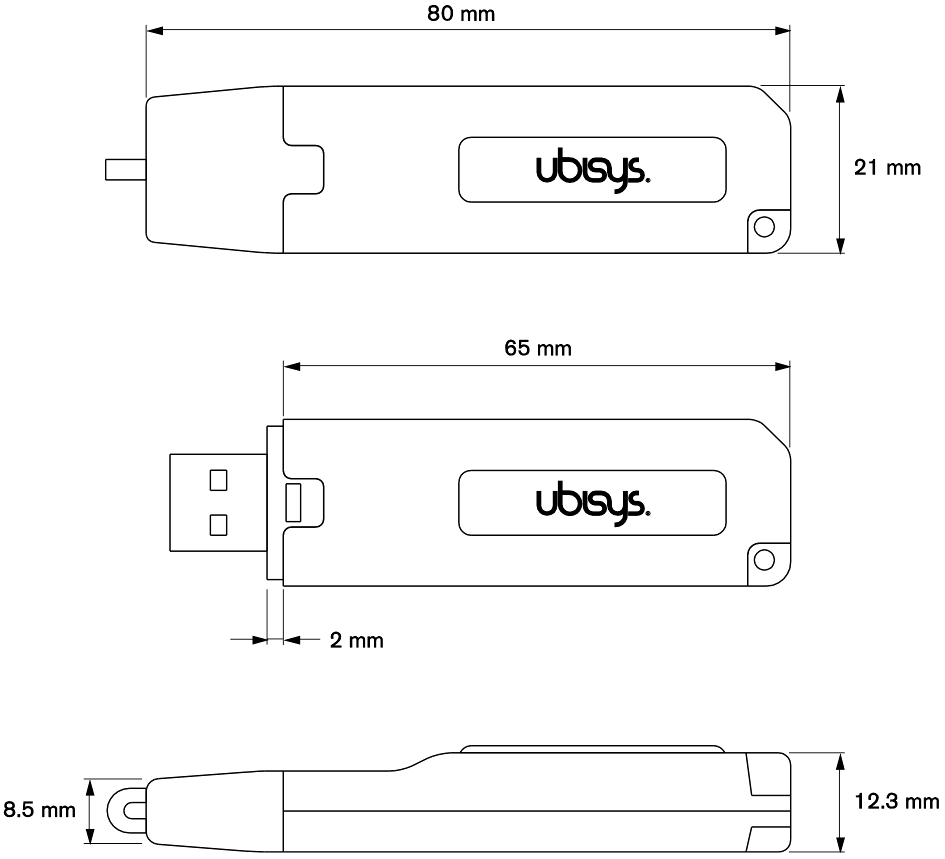


Figure 20: USB Stick with on-board PCB antenna

13. Ordering Information

The following tables list the product variants available. Use the specified order code for your orders. Please contact ubisys support if you require any customization.

Case	Firmware variant	Product Number	Order Code
Black	Wireshark/RNDIS	U0101-010110-02	**9010**
Light gray	Wireshark/RNDIS	U0101-010210-02	**9027**
Transparent	Wireshark/RNDIS	U0101-010310-02	**9034**

14. Declaration of Conformity

CE

We – ubisys technologies GmbH, Am Wehrhahn 45, 40211 Düsseldorf, Germany – declare under our sole responsibility that the ubisys IEEE 802.15.4/ZigBee USB Gateway stick with RNDIS/Wireshark Firmware with order codes as detailed in section 10 under the trade name “ubisys” to which this declaration relates are in conformity with the following directives and standards:

Directive/Standard	Description/Scope
1995/5/EC	Radio and Telecommunications Terminal Equipment Directive (R&TTE)
2004/108/EC	Electromagnetic Compatibility Directive (EMC)
2006/95/EC	Low Voltage Directive (LVD)
2002/96/EC	Waste Electrical and Electronic Equipment Directive (WEEE)
2002/95/EC	Restriction of Hazardous Substances Directive (RoHS)
EN 300 328	ERM; Wideband transmission systems; 2.4 GHz ISM band
EN 300 440	ERM; Radio equipment to be used in the 1 GHz to 40 GHz frequency range
EN 301 489	EMC
IEEE 802.15.4	IEEE Standard 802 – Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)

Düsseldorf, Germany
Place of issue

October 16, 2012
Date of issue

Dr.-Ing. Arasch Honarbacht
Full name of Authorized Signatory

Managing Director, Head of Research & Development
Title of Authorized Signatory

Arasch Honarbacht
Signature

ubisys.
UBISYS TECHNOLOGIES GMBH
HARDWARE UND SOFTWARE DESIGN
ENGINEERING UND CONSULTING
AM WEHRHAHN 45
40211 DÜSSELDORF
info@ubisys.de
www.ubisys.de
Seal

15. Revision History

Revision	Date	Remarks
1.0	25/09/2011	Initial Public Version
1.1	17/10/2011	Added ZigBee PRO Encrypted Traffic Chapter
1.2	18/10/2011	Added Multiple-Channel Capture Chapter
1.3	15/12/2011	Minor corrections
1.4	16/10/2012	Minor corrections. Added diversity capture example and updated multiple capture interface information. Conformity statement included.
1.5	20/12/2012	Added instructions for Linux
1.6	16/06/2014	Added instructions for manually restarting Winpcap
1.7	05/12/2014	Added ZigBee Green Power to the list of protocols supported by Wireshark “out-of-the-box” and added a trouble-shooting section.
1.8	02/18/2015	Included information about timing accuracy in firmware revision 1.04 and above and a list of known issues with various firmware versions. Added a note on distributed security Trust Center link-keys and pre-configured link-keys.

16. Contact

UBISYS TECHNOLOGIES GMBH
HARDWARE AND SOFTWARE DESIGN
ENGINEERING AND CONSULTING

AM WEHRHAHN 45
40211 DÜSSELDORF
GERMANY

T: +49 (211) 54 21 55 - 00
F: +49 (211) 54 21 55 - 99

www.ubisys.de
info@ubisys.de
support@ubisys.de