SHORT RANGE

EVK-THEO-P1

Evaluation kit for THEO-P1 Host-based V2X transceiver modules

User Guide

Abstract

This document describes how to set up the EVK-THEO-P1 evaluation kit to evaluate the THEO-P1 series host-based IEEE 802.11p V2X modules.

www.u-blox.com

UBX-15013939 - R02

Document Information		
Title	**EVK-THEO-P1**	
Subtitle	Evaluation kit for THEO-P1 Host-based V2X transceiver modules	
Document type	User Guide	
Document number	UBX-15013939	
Revision, date	R02	12-Feb-2016
Document status	Advance Information	

Document status information	
Objective Specification	Document contains target values. Revised and supplementary data will be published later.
Advance Information	Document contains data based on early testing. Revised and supplementary data will be published later.
Early Production Information	Document contains data from product verification. Revised and supplementary data may be published later.
Production Information	Document contains the final product specification.

This document applies to the following products:

Product name	Type number	Firmware version	PCN / IN
EVK-THEO-P1	EVK-THEO-P173-00	LLC SDK 1.0.0 LLC driver 2016-01-07	N/A

Contents

1 Evaluation kit description

1.1 Overview

THEO-P1 is a compact, embedded transceiver module that facilitates development of electronics for Vehicle-to-Everything (V2X) communication systems. The module includes an integrated MAC/LLC/Baseband processor and the required RF front-end components. It connects to a host processor through the USB interface.

The EVK-THEO-P1 evaluation kit provides a simple way to evaluate the THEO-P1 series host-based V2X modules. The evaluation kit serves as an evaluation and development platform providing full access to the interfaces of the THEO-P1 radio module, and allows integration of the THEO-P1 module with an external PC or host processor development platform.

The main features of the EVK-THEO-P1 evaluation kit are:

- Micro-USB interface for host communication
- Two 5.9 GHz FAKRA RF connectors
- SMA-to-FAKRA adaptors and external antennas are included in the EVK
- 3.3 V and 5 V power supply

Table 1 lists the available versions of the evaluation kit:

Evaluation kit	Description	Suitable for evaluation of
EVK-THEO-P173	Evaluation kit for the THEO-P173 module	THEO-P173

Table 1: List of available EVK-THEO-P1 evaluation kit

☞ See the *THEO-P1 series Data Sheet [1]* and *THEO-P1 series System Integration Manual [2]* for the features supported by THEO-P1 series V2X modules.

Figure 1 shows the EVK-THEO-P173 evaluation board (EVB) with external antennas, power supply, and USB cable.

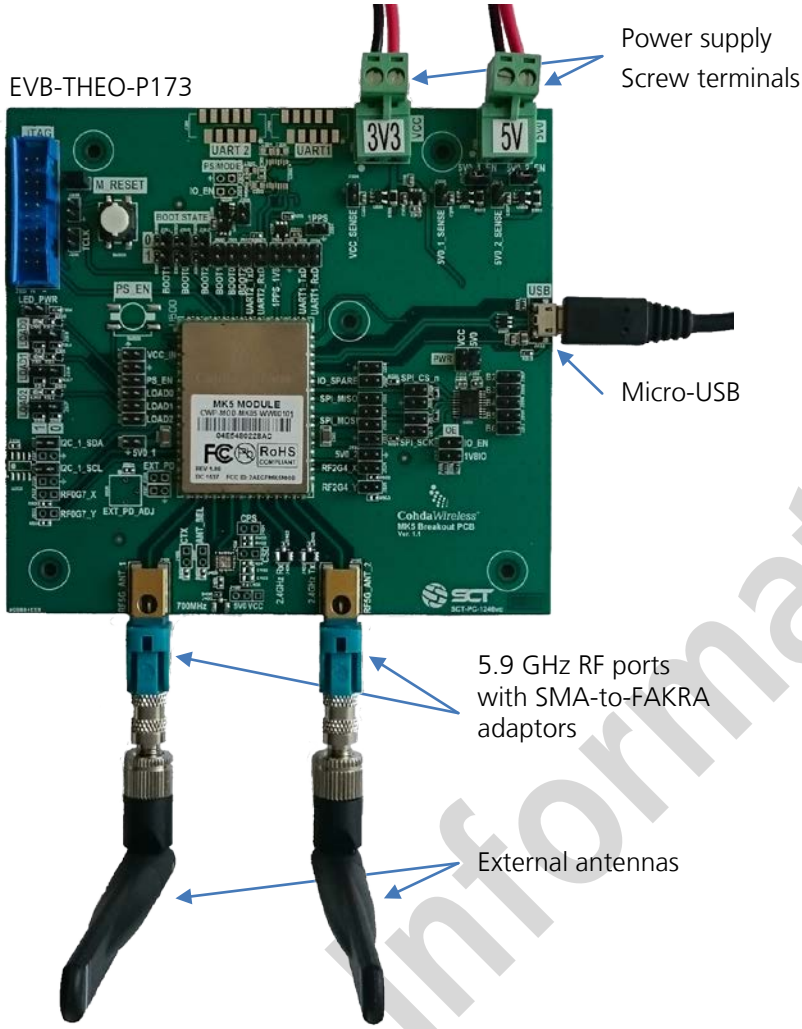

Figure 1: EVK-THEO-P173 evaluation board

☞ The THEO-P173 module was formerly known as the Cohda MK5. The first release of the EVK-THEO-P1 evaluation kit has the Cohda MK5 module mounted on the evaluation board.

1.2 Kit includes

The following is included in the EVK-THEO-P1 evaluation kits:

- EVB-THEO-P173 evaluation board
- Two external 5 GHz antennas and SMA-to-FAKRA adaptors
- Two screw terminals for connecting power supply cables
- Micro-USB cable
- Quick Start card

1.3 Software and documentation

The SDK, containing the Linux drivers, firmware, and evaluation tools for the THEO-P1 module series, is developed by Cohda Wireless and can be re-distributed by u-blox to customers free of charge.

☞ Please contact u-blox support to obtain the software package.

1.4 System requirements

- Host PC with USB 2.0 interface
- VMware Workstation Player/Pro™ (minimum version 7.1.0) for running the SDK

1.5 Specifications

Table 2 and Table 3 list the absolute maximum ratings and operating conditions for the EVB-THEO-P173 evaluation board.

Parameter	Description	Min.	Max.	Unit
3V3	Power supply voltage 3.3 V	-0.3	3.6	V
5V0	Power supply voltage 5 V	-0.3	6.0	V
1PPS	1PPS signal input, 1.8 V CMOS logic	-0.3	5.5	V
$T_{STORAGE}$	Storage temperature	-40	+85	°C

Table 2: Absolute maximum ratings for the EVB-THEO-P173

Parameter	Description		Min.	Typ	Max.	Unit
3V3	Power supply voltage 3.3 V		3.0	3.3	3.6	V
	Power consumption on 3V3 rail		-	550	700	mA
5V0	Power supply voltage 5 V		4.5	5.0	5.5	V
	Power consumption on 5V0 rail		-	-	1000	mA
1PPS	1PPS signal input, 1.8 V CMOS logic	Low	0	-	0.6	V
		High	1.2	-	5.0	V
T_A	Ambient operating temperature		-40	-	+85	°C
Ripple Noise	Peak-to-peak voltage ripple on 3V3 and 5V0 supply lines		-	-	10	mV

Table 3: Operating conditions for the EVB-THEO-P173

2 Getting started

The steps required to evaluate the THEO-P1 series modules using the THEO-P1 evaluation kit are provided below:

1. To connect the external antennas to the board, first connect the provided SMA-to-FAKRA adaptors to the two FAKRA RF ports on the board and then connect the antennas to these adaptors.

⚠ **If you are not using the antennas provided by u-blox with the evaluation kit, ensure that the RF ports are properly terminated to a 50 Ω load such as an antenna, spectrum analyzer or 802.11p receiver. If you directly connect the RF ports of two EVK-THEO-P1 evaluation kits, include a minimum attenuation of 50 dB, to avoid damage to the module.**

2. Connect cables fitting to your power supply to the screw terminals for 3.3 V and 5 V as shown in Figure 2:

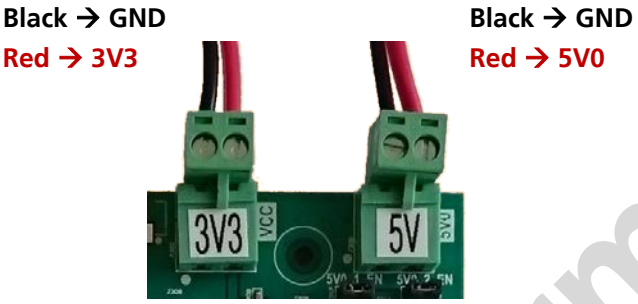

Figure 2: Power supply cable connection

⚠ **Ensure correct polarity while connecting the cables. Signal GND is connected to pin 1 and VCC to pin 2.**

3. Connect a 5 V and 3.3 V bench top power supply to power-up the module.

☞ The current drawn on the 3.3 V supply will be ~100 mA.

4. Connect the micro-USB on the board to a host processor or PC.

⚠ **Always provide power supply to 5 V first, followed by 3.3 V and then the micro-USB, as otherwise the module could be damaged. Reverse the power-up sequence to power-down the evaluation board; that is, first disconnect the USB and then remove 3.3 V and 5 V power supplies.**

5. Start the LLC SDK Virtual Machine *[3]* and connect the "NXP SAF510x DFU" USB device to it. The next steps for evaluating the THEO-P1 module are:
   - Build the Linux driver and applications in the SDK and
   - Download the firmware from the host to the module as described in chapter 4 – "Software development kit"

☞ When you have downloaded the firmware, the current drawn on the 3.3 V supply will rise to approximately 400 to 550 mA.

3 Board description

This section describes the EVB-THEO-P173 evaluation board and the available connectors and configuration settings.

3.1 Block diagram

Figure 3 shows a block diagram of the evaluation board and its main interfaces.

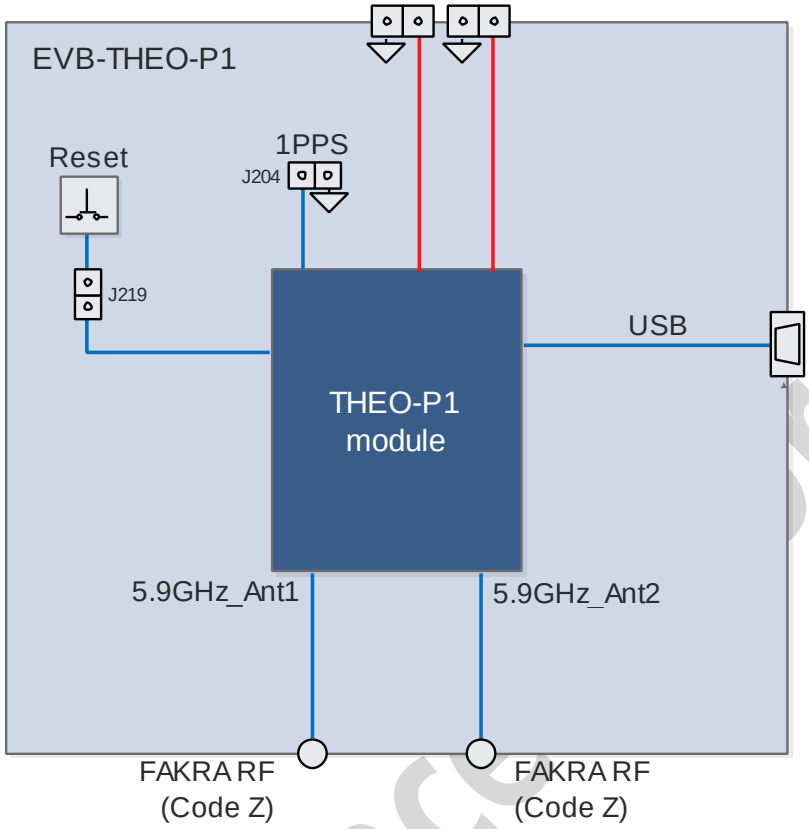

Figure 3: Block diagram of the EVB-THEO-P173 evaluation board

3.2 Connectors

Table 4 lists the available connectors on the EVB-THEO-P173 evaluation board and their functions.

Function	Description	Name
3.3 V Power input	Terminal block for 3.3 V main power supply	VCC
5 V Power input	Terminal block for 5 V RF power supply (Ant1 and Ant2)	5V0
Antenna 1	FAKRA RF connector (Code Z) for first 5.9 GHz antenna	RF5G_ANT_1
Antenna 2	FAKRA RF connector (Code Z) for second 5.9 GHz antenna	RF5G_ANT_2
Host USB	Micro-USB connector for host communication	USB
1PPS Signal input	Pin header for supplying a 1PPS UTC reference signal from an external GNSS receiver	1PPS

Table 4: EVB-THEO-P1 connector description

3.3 1PPS signal input

A 1PPS UTC reference signal is required by the IEEE1609.4 MAC inside the module to align transmissions during channel switching and for timekeeping. The pin header 1PPS/J204 on the evaluation board can be used to connect a 1PPS signal from an external GNSS receiver to the GPS_1PPS input of the THEO-P1 module.

3.4 Reset button

The reset button (M_RESET) on the evaluation board resets the THEO-P1 module. To use it, pin header J219 on the EVB must be bridged.

3.5 Jumpers

The required jumpers are already installed on the evaluation board of the EVK-THEO-P1 evaluation kit, except for the one that is required for the reset button.

4 Software development kit

A Software Development Kit (SDK) is provided in the form of a VMware virtual machine that runs a Linux OS (Ubuntu 14.04 LTS). The SDK contains the driver, firmware and low-level evaluation tools for operating the THEO-P1 module. This SDK is also referred to as the "LLC SDK"[1], as the contained drivers and tools provide access to the module at the Logical Link Control (LLC) layer.

The LLC SDK serves as an environment for the evaluation of the THEO-P1 module and can be used for the development and integration of upper layer V2X solutions on top of the provided LLC API. The driver and part of the tools are provided as source code and can be used as example implementations.

4.1 Software architecture overview

An overview of the LLC software architecture is shown in Figure 4. The THEO-P1 module implements the 802.11p MAC and PHY layers. A firmware runs on the THEO-P1 module, which needs to be downloaded from the host to the module through USB after each power-on. The driver on the host communicates to the module through USB and provides the API between the upper network layer and the radio. Refer to *CohdaMobility MKx Radio LLC remote API Specification [4]* for LLCremote API specification.

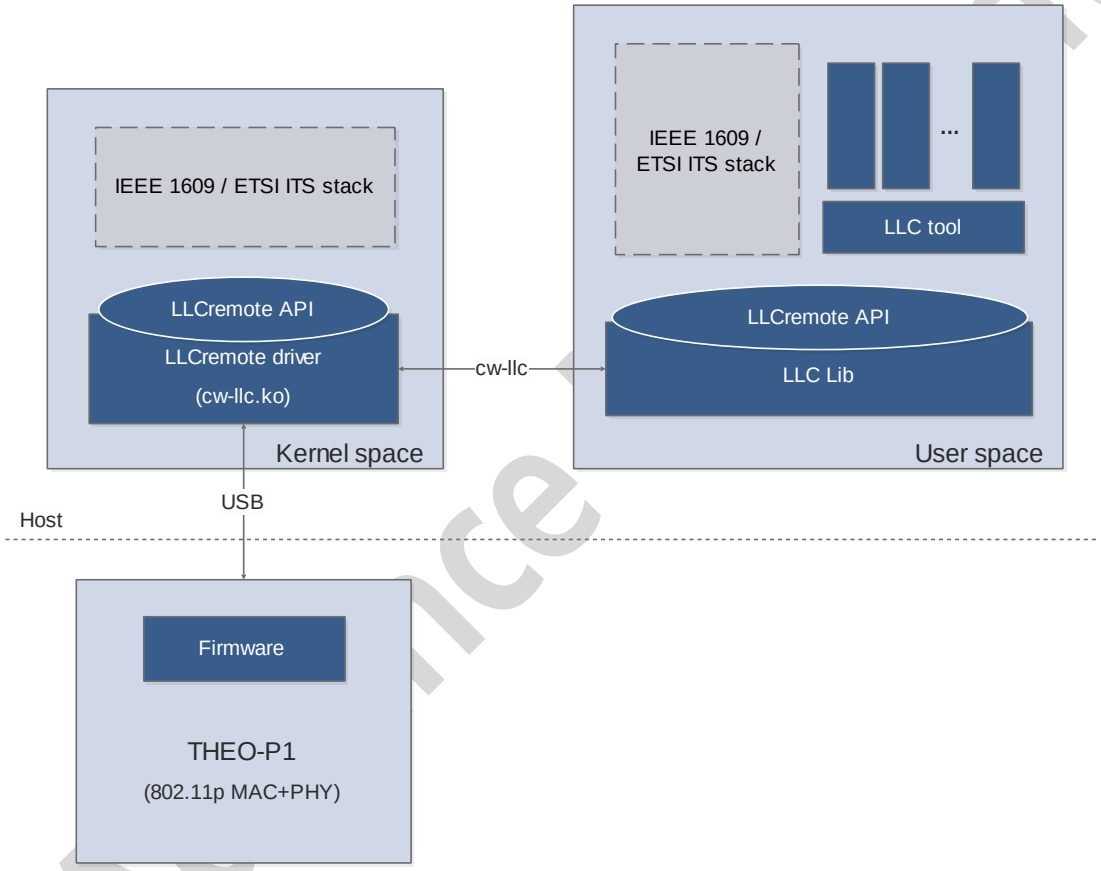

Figure 4: Overview of the LLC software architecture

[1] The LLC SDK, including driver and firmware for the THEO-P1 module, is developed by Cohda Wireless and can be provided free of charge. A full SDK supporting IEEE 1609 and ETSI ITS software stacks can be obtained separately from Cohda Wireless.

The description of the main components is provided in Table 5:

Component	Description	Location in the LLC SDK
Firmware	- Controls the baseband and radio modules - Implements the 802.11p MAC and PHY layers	~/x86/cohda/kernel/drivers/cohda/llc/SDRMK5Dual.bin
LLCremote driver	- Kernel device driver - Provides access to the THEO-P1 module through the USB interface - Provides the LLCremote API and an interface to the user space via the cw-llc network device	Sources: ~/x86/cohda/kernel/drivers/cohda/llc/ Kernel module: ~/x86/cohda/kernel/drivers/cohda/llc/cw-llc.ko
LLCremote API	- Interface between the network layer and the radio for o Configuring the radio o Sending and receiving of 802.11p data packets o Reading status and statistics o UTC time synchronization (timestamping of received packets) o Debugging	
LLC Lib	- Provides the LLCremote API in user space	Sources: ~/x86/cohda/app/llc/lib/
LLC tool	- Wrapper for plugins	Built Executable: ~/x86/cohda/app/llc/llc
LLC plugins	- Small programs for accessing the radio, For example, for : o Configuration o Reading statistics o Sending and receiving packets	~/x86/cohda/app/llc/plugin/ Sources for chconfig, test-tx, test-rx plugins: ~/x86/cohda/app/llc/plugin/simtdapi/

Table 5: Description of the software components

4.2 Installing the SDK

The SDK is an Ubuntu 14.04 virtual machine that can be opened in VMware Workstation Player™ or Pro™. The SDK is provided as a 7zip archive. Extract the files and open the virtual machine from within the VMware application.

4.3 Compiling the LLC driver and tool

Run the LLC SDK virtual machine and open a terminal window. Figure 5 shows an example screenshot of the virtual machine running with a terminal window opened.

To compile the LLC driver, run the following commands to build the driver's kernel module - "cw-llc.ko":

```
cd ~/x86/cohda/kernel/drivers/cohda/llc
make
```

To compile the LLC tool and plugins, run the following commands:

```
cd ~/x86/cohda/app/llc
make
```

This builds the LLC tool "llc", the LLC library "libLLC.so" and the LLC plugins[2] below the plugin directory.

[2] LLC plugins for channel configuration, sending, and receiving packets are provided as source code. Other plugins are provided as binary only.

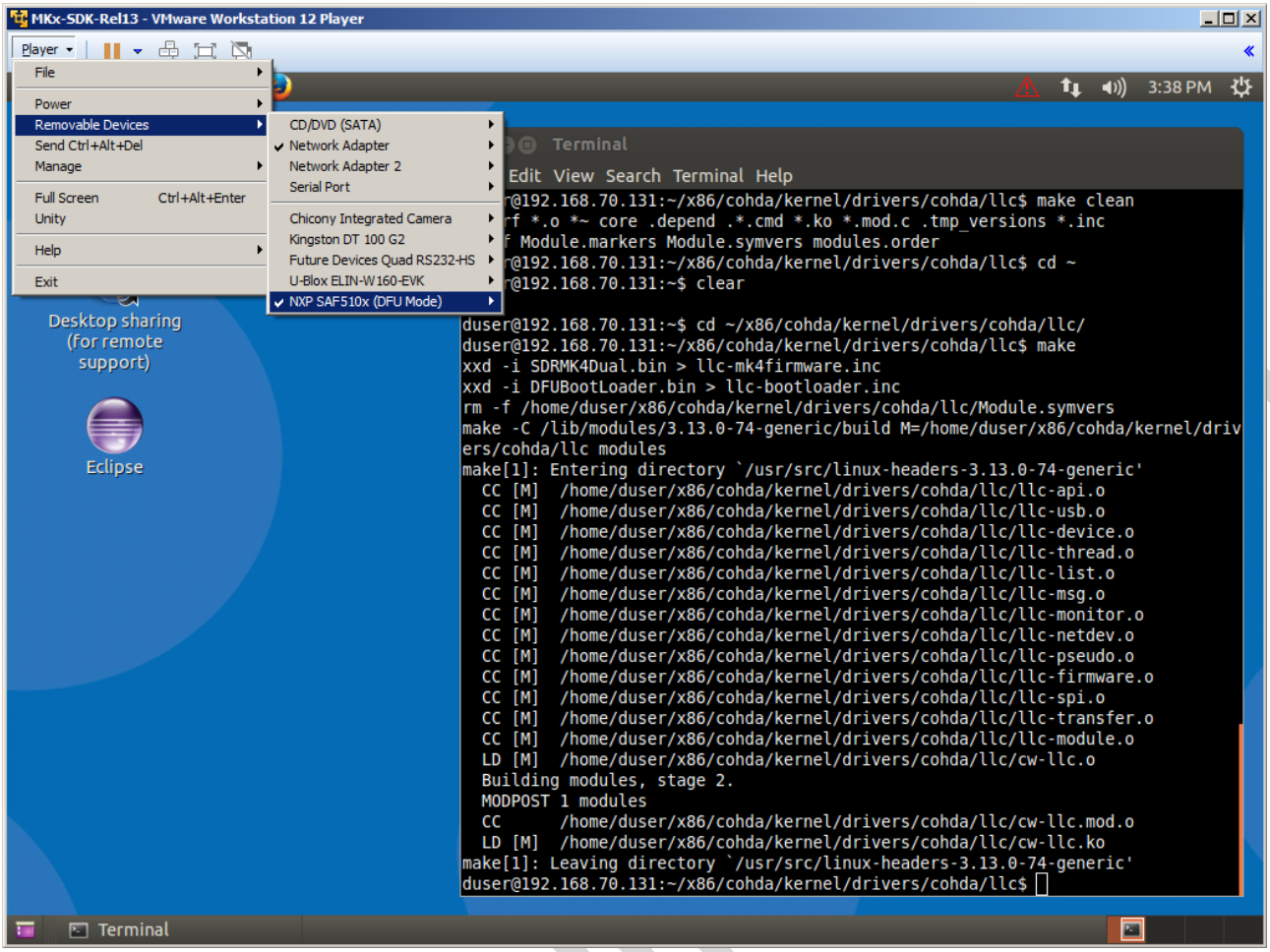

Figure 5: Screenshot of the LLC SDK virtual machine

4.4 Loading the firmware

Once you supply power to the evaluation board and connect the micro-USB to the PC, the THEO-P1 module enumerates itself on the USB as "NXP SAF510x DFU" device. Connect this to the virtual machine as shown in Figure 5. The USB device ID (vendor/product ID) of the device is 1fc9:0102, which can be checked using the "lsusb" command:

```
$ lsusb
Bus 001 Device 003: ID 1fc9:0102 NXP Semiconductors
[…]
```

Next, use "dfu-util" to download the firmware to the THEO-P1 module through USB via the Device Firmware Update (DFU) protocol:

```
sudo dfu-util -d 1fc9:0102 -R -D ~/x86/cohda/kernel/drivers/cohda/llc/SDRMK5Dual.bin
```

The output from the "dfu-util" command should be as shown below:

```
dfu-util 0.5

(C) 2005-2008 by Weston Schmidt, Harald Welte and OpenMoko Inc.
(C) 2010-2011 Tormod Volden (DfuSe support)
This program is Free Software and has ABSOLUTELY NO WARRANTY

dfu-util does currently only support DFU version 1.0

Filter on vendor = 0x1fc9 product = 0x0102
Opening DFU USB device... ID 1fc9:0102
Deducing device DFU version from functional descriptor length
Run-time device DFU version 0100
Claiming USB DFU Runtime Interface...
Determining device status: state = dfuIDLE, status = 0
WARNING: Runtime device already in DFU state ?!?
Found Runtime: [1fc9:0102] devnum=0, cfg=1, intf=0, alt=0, name="UNDEFINED"
Claiming USB DFU Interface...
Setting Alternate Setting #0 ...
Determining device status: state = dfuIDLE, status = 0
dfuIDLE, continuing
Deducing device DFU version from functional descriptor length
DFU mode device DFU version 0100
Device returned transfer size 4096
No valid DFU suffix signature
Warning: File has no DFU suffix
bytes_per_hash=8534
Copying data from PC to DFU device
Starting download: [##################################################] finished!
state(8) = dfuMANIFEST-WAIT-RESET, status(0) = No error condition is present
Done!
can't detach
Resetting USB to switch back to runtime mode
```

After this, the USB mode of the THEO-P1 module will switch over to a normal USB communication device "MK5 SDR 802.11p RADIO" with the USB device ID 1fc9:0103. If the new USB device is not detected automatically after the firmware download is complete, disconnect and reconnect the USB cable and then connect the USB device to the virtual machine again.

The "lsusb" command can be used to check for the new USB device:

```
Bus 001 Device 004: ID 1fc9:0103 NXP Semiconductors
[…]
```

The output of the "dmesg" command for the whole firmware download procedure is provided below:

```
usb 1-1: new high-speed USB device number 3 using ehci-pci
usb 1-1: New USB device found, idVendor=1fc9, idProduct=0102
usb 1-1: New USB device strings: Mfr=1, Product=3, SerialNumber=0
usb 1-1: Product: NXP SAF510x (DFU Mode)
usb 1-1: Manufacturer: NXP Semiconductors
usb 1-1: reset high-speed USB device number 3 using ehci-pci
usb 1-1: USB disconnect, device number 3
usb 1-1: new high-speed USB device number 4 using ehci-pci
usb 1-1: New USB device found, idVendor=1fc9, idProduct=0103
usb 1-1: New USB device strings: Mfr=1, Product=3, SerialNumber=0
usb 1-1: Product: MK5 SDR 802.11p RADIO - High Speed
usb 1-1: Manufacturer: CohdaWireless Pty Ltd
```

4.5 Loading the LLC driver

To load the LLC driver, simply insert the kernel module and enable the "cw-llc" network interface:

```
sudo insmod ~/x86/cohda/kernel/drivers/cohda/llc/cw-llc.ko
sudo ip link set cw-llc up
```

4.6 Using the LLC tool

After loading the firmware and the LLC driver, basic tests can be performed using the LLC tool and the provided plugins. The LLC tool can be found in the SDK in the directory ~/x86/cohda/app/llc/. A list of the available commands/plugins can be obtained by running the following command:

```
./llc -c
```

Some basic plugins and their purpose are described in Table 6.

Plugin	Description
count	Prints out transmit and receive counters for both the radios
config	Shows configuration of radio A or B
dmesg	Shows debug messages from various units inside the module
rxphylast	Shows details of last received frames
txphylast	Shows details of last transmitted frames
chconfig[3]	Configures and starts channels for radio A or B
test-tx[3]	Generates and transmits test packets
test-rx[3]	Receives test packets

Table 6: Description of LLC plugins

Instructions on how to use the individual commands are printed with the "--help" option as shown in the example below:

```
./llc <command-name> --help
```

[3] Source code for this plugin is provided in the plugins/simtdapi subdirectory.

4.6.1 Transmit and receive counters

Use the command "llc count" to print out the transmit and receive counters for both the radios as shown below:

☞ This command can be used to easily verify the access to the module.

	RadA Ch0	RadA Ch1	RadB Ch0	RadB Ch1
Tx MAC Unconfigured Frames	0	0	282	0
Tx MAC Failed Enqueue	0	0	0	0
Tx MAC Broadcast Enqueued Frames	5331	0	3	0
Tx MAC Broadcast Success Frames	5331	0	2	0
Tx MAC Broadcast Failed TTL	0	0	0	0
Tx MAC Broadcast Retired Frames	5331	0	2	0
Tx MAC Unicast Enqueued Frames	0	0	0	0
Tx MAC Unicast RTS/CTS Enq Frame	0	0	0	0
Tx MAC Unicast Success Frames	0	0	0	0
Tx MAC Unicast Failed Retry	0	0	0	0
Tx MAC Unicast Failed TTL	0	0	0	0
Tx MAC Unicast Retired Frames	0	0	0	0
Tx MAC Unicast Retry Frames	0	0	0	0
Tx MAC Unicast Ack Frames	0	0	0	0
Tx MAC Unicast RTS Frames	0	0	0	0
Tx MAC Unicast CTS Frames	0	0	0	0
Tx PHY Frames	5332	0	3	0
Tx PHY Aborted Frames	2	0	0	0
Tx PHY Aborted Resp (Ack/CTS/..)	0	0	0	0
Rx PHY AGC Firings	0	0	0	0
Rx PHY Acquisitions	0	2765	0	0
Rx PHY Failed SF	0	2	0	0
Rx PHY Valid SF Frames	0	2763	0	0
Rx MAC Failed FCS	0	0	0	0
Rx MAC Valid Matched RTS Frames	0	0	0	0
Rx MAC Valid Unmatched RTS Frame	0	0	0	0
Rx MAC Valid Matched CTS Frames	0	0	0	0
Rx MAC Valid Unmatched CTS Frame	0	0	0	0
Rx MAC Valid Data Frames	0	2763	0	0
Rx MAC Valid Matched Ack Frames	0	0	0	0
Rx MAC Valid Unmatched Ack Frame	0	0	0	0
Rx MAC Valid Matched Late Ack Fr	0	0	0	0
Rx MAC Broadcast Frames	0	2763	0	0
Rx MAC Address Matched Frames	0	0	0	0
Rx MAC Address Dropped Frames	0	0	0	0
Rx MAC Buffer Overflow Frames	0	0	0	0
Rx MAC Duplicate Frames	0	0	0	0
Rx MAC USB Transferred Frames	0	2763	0	0
Rx MAC USB Callbacks	2763	0	0	0
CS Active Firings	2	2751	0	0
CS Idle Firings	2	40	0	0
CCA Active Firings	0	51	0	0
CCA Idle Firings	0	2765	0	0
Rx Frame Firings	0	2765	0	0

4.6.2 Channel configuration

Two different channel configurations are available per radio on the THEO-P1, which can be configured by using the "chconfig" LLC plugin via the Control Channel (CCH) or Service Channel (SCH). The plugin uses the LLCremote API to set up, start, stop or get the CCH/SCH configurations.

The following example configures and starts the control channel (radio channel configuration 0) on radio A, channel 184, and both the antennas:

```
$ ./llc chconfig -s -w CCH -c 184 -r a -a 3
Interface:         wave-raw
Channel: CCH
Radio: A
ChannelNumber:  184
DefaultMCS:     10
DefaultTxPower: 40
DefaultTRC:     0
DefaultTPC:     0
Bandwidth:      10
DualTxControl:  0
ChannelUtilisationPeriod: 49
TxAntenna:      3
RxAntenna:      3
MACAddr: 04:e5:48:00:10:00
Filter:            0x88b5
MAC Address: 04:e5:48:00:10:00
```

4.6.3 Transmitter test

The "test-tx" LLC plugin is used to transmit a burst of packets. The CCH or SCH must be configured prior to transmitting a burst and their settings define the default configurations for the packets.

The following example transmits 100 test packets of 200 bytes length on channel 184 and both antennas, with the output power set to 40 and modulation set to ½ QPSK:

```
$ ./llc test-tx -c 184 -p 40 -a 3 -m MK2MCS_R12QPSK -n 100 -l 200
Mode:             CREATE
Number Of Packets:  100
Target Packet Rate: 10.00
Priority: 4
Service: 1
MCS[0]: 10
NMCS: 1
TxPwrCtrl: 0
TxPower: 40
TxAntenna[0]: 3
NTxAnt: 1
Expiry: 0
PayloadLength[0]: 200
Destination MAC Address: ff:ff:ff:ff:ff:ff
EtherType: 0x88b5
ChannelNumber:     184
PayloadMode:       increment
Source MAC Address: 04:e5:48:00:10:00
```

```
Packet Log File:
DumpPayload:       0
DumpToStdout:      0
Interface:         wave-raw
Tx: Last SeqNum:         10 [/100]. Packet rate: Current     9.6, Target    10.0
Tx: Last SeqNum:         20 [/100]. Packet rate: Current    10.0, Target    10.0
Tx: Last SeqNum:         30 [/100]. Packet rate: Current    10.0, Target    10.0
Tx: Last SeqNum:         40 [/100]. Packet rate: Current    10.0, Target    10.0
Tx: Last SeqNum:         50 [/100]. Packet rate: Current    10.0, Target    10.0
Tx: Last SeqNum:         60 [/100]. Packet rate: Current    10.0, Target    10.0
Tx: Last SeqNum:         70 [/100]. Packet rate: Current    10.0, Target    10.0
Tx: Last SeqNum:         80 [/100]. Packet rate: Current    10.0, Target    10.0
Tx: Last SeqNum:         90 [/100]. Packet rate: Current    10.0, Target    10.0
Tx: Last SeqNum:        100 [/100]. Packet rate: Current     9.8, Target    10.0
```

4.6.4 Receiver test

The "test-rx" command is used to receive test packets sent by another module with the "test-tx" command. The receiving application periodically informs about the number of received packets, the calculated packet error rate and also dumps per packet and statistics logs. The receive test should be running before starting the transmission to receive all the packets.

☞ The LLC driver supports only one THEO-P1 module being connected to the host system. In order to test packet transmission with two modules, two SDKs must be used in parallel.

The following example sets the channel configuration and starts receiving and transmitting of test packets with two different THEO-P1 modules:

On the receiving side:

```
./llc chconfig -s -w CCH -c 184
./llc test-rx -c 184
```

On the transmitting side, sends 1000 packets at a rate of 100 packets persecond:

```
./llc chconfig -s -w CCH -c 184
./llc test-tx -c 184 -a1 -p30 -n1000 -r100
```

The output on the receiver will look like the following:

```
Packet Log File:
Report File:       RxReport.txt
LogUnMatched:      0
DumpPayload:       0
DumpToStdout:      0
ReportPeriod:      100
Filter SA: None
Interface:         wave-raw
ChannelNumber:     184
Rx: Report [Last SeqNum: -1 (0xffffffff)]
Rx:   Approx PER: -nan% [Missed: 0, Payload Error: 0 / Tx: 0] (264.6% [-67108864, -1975612572 / -
772014096)
Rx:   Un/Matched Frames: 0/0
     Total          0          0
Rx: Report [Last SeqNum: 100 (0x00000064)]
```

```
Rx:   Approx PER: 0.0% [Missed: 0, Payload Error: 0 / Tx: 101] (0.0% [0, 0 / 100)
Rx:   Un/Matched Frames: 0/101
Rx:   Channel 184: 00000101 Matched Packets.
# MCS  Len    Matched Payload Err
   11  104        101          0
     Total        101          0

[…]

Rx: Report [Last SeqNum: 999 (0x000003e7)]
Rx:   Approx PER: 0.0% [Missed: 0, Payload Error: 0 / Tx: 1000] (0.0% [0, 0 / 99)
Rx:   Un/Matched Frames: 0/1000
Rx:   Channel 184: 00001000 Matched Packets.
# MCS  Len    Matched Payload Err
   11  104       1000          0
     Total       1000          0
```

4.7 Debugging

The following script runs a sequence of LLC debug commands to read diagnostic data from the firmware. Provide this data to u-blox support for analyzing firmware issues.

```
echo -e "\nCommand: llc config"
llc config
echo -e "\nCommand: llc raw 20"
llc raw 20
echo -e "\nCommand: llc count"
llc count
echo -e "\nCommand: llc txphylast"
llc txphylast
echo -e "\nCommand: llc rxphylast"
llc rxphylast
echo -e "\nCommand: llc raw 44"
llc raw 44
echo -e "\nCommand: llc raw 45"
llc raw 45
echo -e "\nCommand: llc raw 22"
llc raw 22
echo -e "\nCommand: llc txqueue a"
llc txqueue a
echo -e "\nCommand: llc txqueue b"
llc txqueue b
echo -e "\nCommand: llc dmesg arm"
llc dmesg arm
echo -e "\nCommand: llc dmesg vdsp1"
llc dmesg vdsp1
echo -e "\nCommand: llc dmesg vdsp2"
llc dmesg vdsp2
echo -e "\nCommand: llc reg r 0"
llc reg r 0
```

```
echo -e "\nCommand: llc reg r 1"
llc reg r 1
echo -e "\nCommand: llc reg r 2"
llc reg r 2
echo -e "\nCommand: llc reg r 3"
llc reg r 3
echo -e "\nCommand: llc raw 12"
llc raw 12
echo -e "\nCommand: llc raw 13"
llc raw 13
echo -e "\nCommand: llc raw 14"
llc raw 14
echo -e "\nCommand: llc raw 18"
llc raw 18
echo -e "\nCommand: llc raw 29"
llc raw 29
echo -e "\nCommand: llc raw 41"
llc raw 41
echo -e "\nCommand: llc raw 44"
llc raw 44
echo -e "\nCommand: llc dbg 28"
llc dbg 28
echo -e "\nCommand: llc error"
llc error
echo -e "\nCommand: llc raw 25"
llc raw 25

# rerun some of the commands to check if the radio is active (tx or rx)
echo -e "\nCommand: llc count"
llc count
echo -e "\nCommand: llc dmesg arm"
llc dmesg arm
echo -e "\nCommand: llc dmesg vdsp1"
llc dmesg vdsp1
echo -e "\nCommand: llc dmesg vdsp2"
llc dmesg vdsp2
```

Appendix

A Glossary

Name	Definition
1PPS	1 Pulse Per Second
API	Application Programming Interface
CMOS	Complementary Metal-Oxide-Semiconductor
DFU	Device Firmware Upgrade
ETSI	European Telecommunications Standards Institute
EVB	Evaluation Board
EVK	Evaluation Kit
IEEE	Institute of Electrical and Electronics Engineers
ITS	Intelligent Transport Systems
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
MAC	Medium Access Control
LLC	Logical Link Control
OS	Operating System
PC	Personal Computer
PHY	Physical Layer
QPSK	Quadrature Phase-Shift Keying
RF	Radio Frequency
SDK	Software Development Kit
USB	Universal Serial Bus
UTC	Coordinated Universal Time
V2X	Vehicle-to-Everything

Table 7: Explanation of abbreviations used

Related documents

[1] THEO-P1 series Data sheet, Document Number UBX-15023940

[2] THEO-P1 series System Integration Manual, Document Number UBX-15029954

[3] Cohda Wireless LLC SDK VM

[4] CohdaMobility MKx Radio LLCremote API Specification, CWD-MKx-0208

Revision history

Revision	Date	Name	Status / Comments
R01	27-May-2015	mzes	Preliminary release
R02	12-Feb-2016	mzes, kgom	The product name is changed from Cohda MK5 to THEO-P1 series. Major updates for the new evaluation kit. Modified document status to Advance Information.

Contact

For complete contact information visit us at www.u-blox.com.

u-blox Offices

North, Central and South America

u-blox America, Inc.

Phone: +1 703 483 3180
E-mail: info_us@u-blox.com

Regional Office West Coast:

Phone: +1 408 573 3640
E-mail: info_us@u-blox.com

Technical Support:

Phone: +1 703 483 3185
E-mail: support_us@u-blox.com

Headquarters
Europe, Middle East, Africa

u-blox AG

Phone: +41 44 722 74 44
E-mail: info@u-blox.com
Support: support@u-blox.com

Asia, Australia, Pacific

u-blox Singapore Pte. Ltd.

Phone: +65 6734 3811
E-mail: info_ap@u-blox.com
Support: support_ap@u-blox.com

Regional Office Australia:

Phone: +61 2 8448 2016
E-mail: info_anz@u-blox.com
Support: support_ap@u-blox.com

Regional Office China (Beijing):

Phone: +86 10 68 133 545
E-mail: info_cn@u-blox.com
Support: support_cn@u-blox.com

Regional Office China (Chongqing):

Phone: +86 23 6815 1588
E-mail: info_cn@u-blox.com
Support: support_cn@u-blox.com

Regional Office China (Shanghai):

Phone: +86 21 6090 4832
E-mail: info_cn@u-blox.com
Support: support_cn@u-blox.com

Regional Office China (Shenzhen):

Phone: +86 755 8627 1083
E-mail: info_cn@u-blox.com
Support: support_cn@u-blox.com

Regional Office India:

Phone: +91 80 4050 9200
E-mail: info_in@u-blox.com
Support: support_in@u-blox.com

Regional Office Japan (Osaka):

Phone: +81 6 6941 3660
E-mail: info_jp@u-blox.com
Support: support_jp@u-blox.com

Regional Office Japan (Tokyo):

Phone: +81 3 5775 3850
E-mail: info_jp@u-blox.com
Support: support_jp@u-blox.com

Regional Office Korea:

Phone: +82 2 542 0861
E-mail: info_kr@u-blox.com
Support: support_kr@u-blox.com

Regional Office Taiwan:

Phone: +886 2 2657 1090
E-mail: info_tw@u-blox.com
Support: support_tw@u-blox.com