
Serialized PRIME Network Manager on Linux

Summary

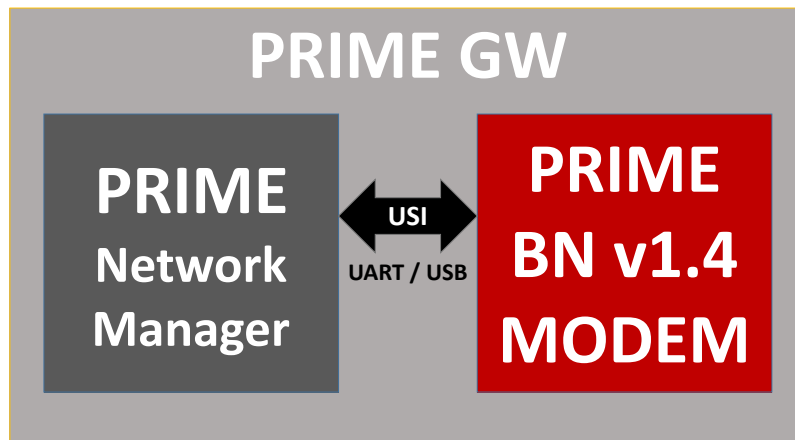
The PRIME Network Manager application is used to monitor and manage PRIME power line networks. This application requires a hardware device connected to the power line.

The PRIME Network Manager can be used for:

- PIB management
- Testing embedded firmware
- Monitorizing embedded Sniffer

The PRIME Network Manager application with command line interface (CLI) capabilities was developed as an example of use based on the Universal Serial Interface (USI) available on the `usi_host` package. It can be executed on any Linux™ Host connected to an embedded device running the PRIME Base Node Modem project with a simple serial port-like interface.

Figure 1. PRIME Network Manager General Schema



To evaluate the application, a complete Linux™ system is built on a SAMA5D27-SOM1-EK1 board connected through a mikroBUS to a PL360G55CF-EK board running the embedded Base Node Modem project present on the [PRIME v1.4 Base](#) package.

Table of Contents

Summary.....	1
1. Hardware Description.....	3
2. Software Description.....	5
2.1. PRIME Base Node Modem Embedded Project.....	5
2.2. Setting Up the Linux Environment.....	7
3. Microchip PRIME Network Manager Application.....	9
4. Building Microchip PRIME Base Node on Linux Application.....	11
5. Start-up PRIME Network Manager on Linux.....	12
6. Base Node Command Line Interface.....	14
6.1. SHOW commands.....	14
6.2. CONFIG commands.....	20
6.3. PIB commands.....	22
6.4. PHY commands.....	23
6.5. MAC commands.....	23
6.6. TARGET commands.....	24
6.7. RESET command.....	24
6.8. REBOOT command.....	24
6.9. NETWORK commands.....	24
6.10. FW UPGRADE commands.....	25
7. Revision History.....	28
7.1. Rev A - 12/2021.....	28
The Microchip Website.....	29
Product Change Notification Service.....	29
Customer Support.....	29
Microchip Devices Code Protection Feature.....	29
Legal Notice.....	29
Trademarks.....	30
Quality Management System.....	31
Worldwide Sales and Service.....	32

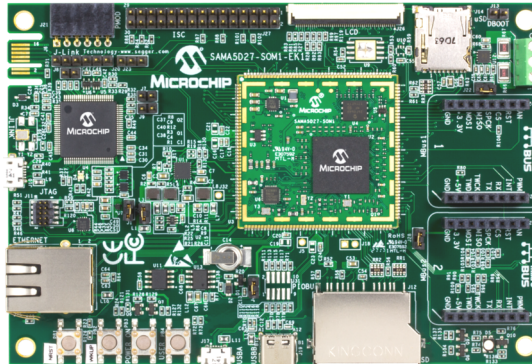
1. Hardware Description

Microchip provides several PRIME FW stack libraries for the Base Node. All of them are fully functional and the only difference resides in the number of supported nodes. Depending on the Hardware, the maximum number of Service Nodes to be supported by the PRIME Base Node are different. In addition, the only HW interface between the Host and the PRIME Base Node Modem is a Serial Port or a USB.

This application note is only for evaluation purposes so the most simple combination of boards was chosen. The Linux™ Host was implemented using a SAMA5D27-SOM1-EK1 and the Embedded Prime Base Node Modem uses a PL360G55CF-EK. The boards are connected through the MikroBUS interface, using the UART as a communication port.

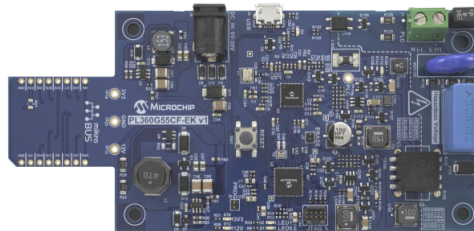
- SAMA5D27-SOM1-EK1
 - <https://www.microchip.com/developmenttools/ProductDetails/atsama5d27-som1-ek1>

Figure 1-1. SAMA5D27-SOM1-EK1



- PL360G55CF-EK
 - <https://www.microchip.com/developmenttools/ProductDetails/pl360g55cf-ek1>

Figure 1-2. PL360G55-CF EK



Both boards are available as evaluation kits from Microchipdirect.com and distributors, containing software, hardware schematics, Bill of Materials and PCB designs.

The SAMA5D27-SOM1-EK1 includes two mikroBUS™ connectors where PL360G55CF-EK could be connected. MikroBUS 1 is used by default. [Figure 1-3](#) and [Figure 1-4](#) show the mikroBUS schematic on both evaluation kits.

Figure 1-3. SAMA5D27-SOM1-EK1 mikroBUS 1

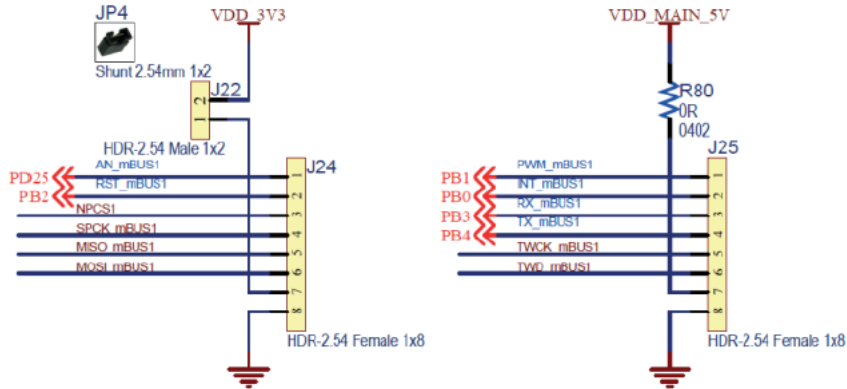
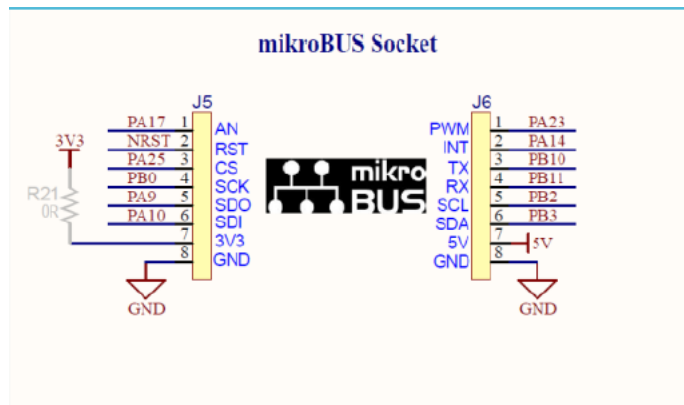


Figure 1-4. PL360G55-CF EK mikroBUS



The SAMA5D27-SOM1-EK1 is powered by USB and PL360G55CF-EK is powered by a 5V pin in the mikroBUS™. The project is using mikroBUS 1 of SAMA5D27-SOM1-EK1; therefore, verify in the SAMA5D27-SOM1-EK1 board that the resistor R80 is soldered to enable the 5 Volts pin to supply power (in case of using mikroBUS 2, the resistor to check is R109).

An SD card is also required to run the Linux™ image of the project. The SD card is connected to the SD connector of the SAM5D27-SOM1-EK1 (J12).

It is also possible to run the application with a native Linux™ machine or in a Windows environment (like CYGWIN or Windows 10 WSL) by connecting a USB cable directly to the microUSB port of the PL360G55 board. (It appears like a Serial Port on the Host side).

2. Software Description

The PRIME Network Manager application is used to monitor and manage PRIME power line networks. It is an example of use based on the Universal Serial Interface (USI) available on the `usi_host` package. It can be executed on any Linux Host connected to an embedded device running the PRIME Base Node Modem project with a simple serial port-like interface.

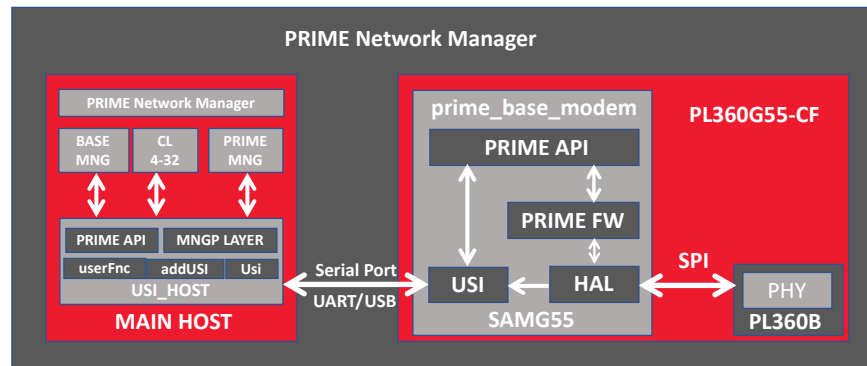
The main capabilities of the application are:

- Command Line Interface (CLI)
- PIB management
- Testing embedded firmware
- Network firmware upgrade
- Monitorizing embedded sniffer

This walkthrough will allow users to build, on a Linux environment, a PRIME Network Manager Linux example running on a SAMA5D27 microprocessor. The PL360G55CF-EK runs the application example `base_node_modem` included in the Microchip PRIME Base stack release and the communication interface between both boards is handled by the `usi_host` package.

Figure 2-1 shows the software and hardware relationships between the different elements.

Figure 2-1. PRIME Network Manager Software Blocks

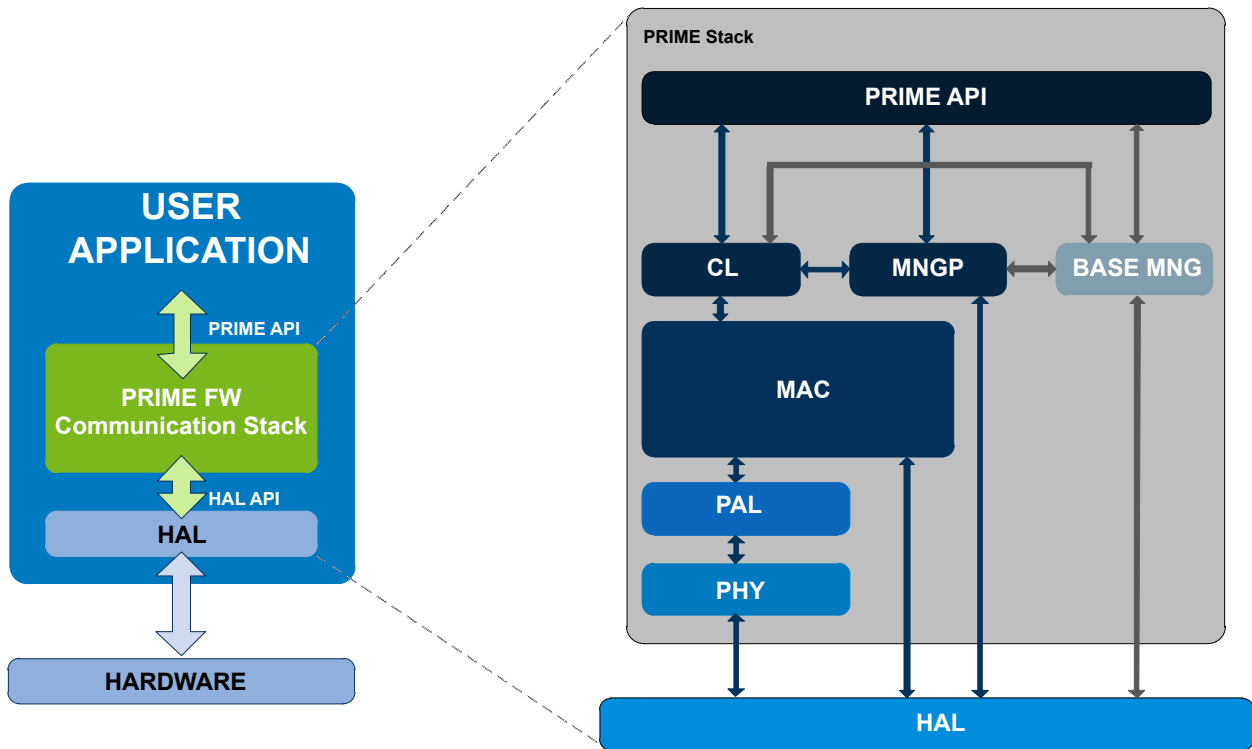


2.1 PRIME Base Node Modem Embedded Project

The PL360G55CF-EK is flashed with the example project `base_node_modem` from the last PRIME 1.4 Firmware package release. The original project is located on the following path of the PRIME Base 1.4 package:
`thirdparty\prime_ng\apps_1_4\prime_base_modem`.

The PRIME Firmware Stack follows a layered approach based on the PRIME specification. The following figure shows the stack architecture:

Figure 2-2. PRIME Firmware Stack on Base Node



Microchip provides a library that contains the MAC layer, the Convergence Layer and the Management Plane, as described in the PRIME specification. It also includes a proprietary Base Management module to handle the interface between the Base Node and the main DCU user application and enhances it with additional functionalities.

It is important to note that the only entry point to the PRIME FW stack from the user application is through the PRIME API, which contains the interfaces defined in the PRIME specification as well as stack control functions.

The Management Plane enables a local or remote control entity to perform actions on a node. These actions include providing access to internal parameters defined by PIB attributes as well as managing the firmware upgrade inside the stack.

The Base Management module is a Microchip proprietary extension over the PRIME specification to increase the functionalities of a Base Node. The PRIME specification does not always define the interfaces between the Base Node and the DCU user application, so new interfaces must be implemented as required.

The Base Management handles the Firmware Upgrade Protocol interface and access to the PRIME Profile. It also notifies about network events, such as node registrations and unregistrations, and manages the whitelist.

In the `base_node_modem` by default, the USI port is the USB interface of the PL360G55CF board. In this case, the mikroBUS UART (USART4) is used, which requires some modifications in two header files.

If working with a Linux or Windows Host connected directly to the PL360G55 board, the project MUST NOT be modified because, by default, the USI Port goes through the USB Port.

In the header file `conf_usi.h`, to use the mikroBUS UART:

```
#ifndef CONF_USI_H_INCLUDED
#define CONF_USI_H_INCLUDED

/* Port Communications configuration */
#define NUM_PORTS 1
/* In case of using USB_TYPE, refer to conf_usb.h in order to set baudrate by default */
/* #define PORT_0 CONF_PORT(USB_TYPE, 0, 115200, 1024, 1024) */
/* In case of using UART/USART TYPE */
#define PORT_0 CONF_PORT(USART_TYPE, 4, 115200, 1024, 1024)
#endif /* CONF_USI_H_INCLUDED */
```

In the header file `conf_project.h`:

```
#ifndef CONF_BOARD_H_INCLUDED
#define CONF_BOARD_H_INCLUDED

/* Enable Watchdog */
#define CONF_BOARD_KEEP_WATCHDOG_AT_INIT

/* Configuration for USI Port on MikroBUS */
#define CONF_BOARD_UART_MIKROBUS
#define CONF_BOARD_USART4

/* #define CONF_BOARD_UART_CONSOLE */
#define CONF_BOARD_CONSOLE_UART          USART0
#define CONF_BOARD_CONSOLE_UART_ID      ID_FLEXCOM0

/* Configure MIKROBUS_SPI */
/* #define CONF_BOARD_SPI_MIKROBUS */

/* Configure MIKROBUS_TWI */
/* #define CONF_BOARD_TWI1_MIKROBUS */

/* Configure USB */
#define CONF_BOARD_USB_PORT
#endif /* CONF_BOARD_H_INCLUDED */
```

Compile the project and program the board to be used as the PRIME Base Node modem.

2.2 Setting Up the Linux Environment

The Linux binaries that run in ATSAM5D27-SOM1-EK1 are generated using Buildroot (<https://buildroot.org/>), a tool to generate embedded Linux systems through cross-compilation. Microchip provides its own version of Buildroot which includes kernel, cross compiler and rootfs for ATSAM5D27-SOM1-EK1. More information can be found at:

- http://www.at91.com/linux4sam/bin/view/Linux4SAM/BuildRootBuild#How_to_build_Buildroot_for_AT91
- <https://www.at91.com/linux4sam/bin/view/Linux4SAM/Sama5d27Som1EKMainPage>
- ftp://www.at91.com/pub/demo/linux4sam_5.8/

Follow these instructions to build the SD image with the Linux binaries when using a PC running Ubuntu 19.10. If using a different version or distribution, the process may vary.

1. Install packages not included in the Linux distribution (For more information about the system requirements, see buildroot.uclibc.org/downloads/manual/manual.html#requirement):

```
$ sudo apt-get install sed make binutils build-essential gcc g++ bash patch
$ sudo apt-get install gzip bzip2 perl tar cpio python unzip rsync file bc wget
$ sudo apt-get install libncurses5-dev libncursesw5-dev bzip2 cvs git mercurial
$ sudo apt-get install subversion graphviz python-matplotlib
```

2. Download the sources of the Microchip Buildroot (this application note is based on version 5.8):

```
$ git clone https://github.com/linux4sam/buildroot-at91.git
$ git clone https://github.com/linux4sam/buildroot-external-microchip.git
$ cd buildroot-external-microchip
$ git checkout linux4sam_5.8 -b buildroot-external-microchip-linux4sam_5.8
$ cd ..
$ cd buildroot-at91
$ git checkout linux4sam_5.8 -b buildroot-at91-linux4sam_5.8
```

3. Build the Microchip Buildroot rootfs image:

```
$ BR2_EXTERNAL=../buildroot-external-microchip/ make
atmel_sama5d27_som1_ek_mmc_dev_defconfig
$ make
```

4. Customize the buildroot and save the configuration:

```
$make menuconfig
```

Enable the following configurations/packages if needed:

- \Target Packages\Networking applications\iputils (**BR2_PACKAGE_IPUTILS=y**)
- \Target Packages\Networking applications\openssh (**BR2_PACKAGE_OPENSSSH=y**)
- \Target Packages\Databases\sqlite3 (**BR2_PACKAGE_SQLITE=y**)
- \Target Packages\Text and terminal handling\ncurses (**BR2_PACKAGE_NCURSES=y**)
- \Target Packages\Text and terminal handling\readline (**BR2_PACKAGE_READLINE=y**)

5. Run the Buildroot compilation:

```
$make
```

Buildroot downloads and compiles all the packages required, so it may take quite a long time. After this point, the kernel, cross-compiler and rootfs are ready to flash your system.

6. If a USB serial port is used to connect the Linux Host to the PRIME BN modem, it is necessary to compile some additional Linux Drivers. To do that, in the kernel configuration:

```
$make linux-menuconfig
```

If using a PL360G55 board, enable the Atmel USB support as a module:

- \Device Drivers\usb\gadget\udc\“atmel_usba_udc”

If using an ATPL360 board, enable the CDC-ACM USB class support as module:

- \Device Drivers\usb\class\“cdc-acm”

If using an ATPL230BN board, enable the CP210X USB serial support as module:

- \Device Drivers\usb\serial\“cp210x”

```
$make linux-build
```

7. Create the SD image:

```
$make
```

8. After compiling all the binaries needed, update the SD card image to be mounted on SAMA5D27-SOM1-EK1 with the correct binaries located in:

```
buildroot-at91/output/images/sdcard.img
```

The Linux binaries are now ready to be used. Follow these steps to verify the proper operation of the kit:

1. Insert the SD card in the MMC/SD card connector (J12)
2. Connect the USB Micro-AB cable to connector J10 (JLINK-OB)
3. Connect the another end of the cable to a free port of your PC
4. Open a terminal (console 115200, N, 8, 1) on your PC
5. Reset the baseboard. A start-up message appears on the console

To use the Ethernet connector and easily access the SAMA5D27-SOM1-EK1, configure a new IP address:

```
$ ip address add 192.168.1.102/24 dev eth0  
$ ifconfig eth0 up
```

So, in this case, <target-ip-address> is 192.168.1.102 .

3. Microchip PRIME Network Manager Application

The PRIME Network Manager application example explained in this application note is included in the `usi-host-v2.2.2` package. It implements a basic Command Line Interface (CLI) for managing and monitoring a PRIME Network running over the Base Node Modem project present on PRIME 1.4 Firmware Stack package.

The PRIME Network Manager can be used for:

- PIB management: get counters and modify PRIME standard and manufacturer stack settings
- Testing embedded firmware: check the correct behavior of the embedded PRIME Base Node modem project
- Firmware upgrade of the network

It includes other capabilities like:

- Trace Logging: to debug and trace the program execution
- Configuration based on a file: to save and restore configurations without recompilation
- Local (file) / Remote (proxy) sniffing capabilities: to monitor the PRIME network in real time or save the information for further analysis.
- Network Events saved on SQLite database: to save network events (registrations, promotions...) for further analysis
- Simple CLI update: to add new commands in a simple way

This application uses the USI Host API to interact with the Base Node Modem; therefore, before continuing, read the documentation carefully for the PRIME Base Stack and USI Host for further information on the PRIME Serialization protocols and USI interface.

The application is structured according to the USI Host serialization, so, the first step requires you to consider the following points:

- USI Project Configuration (`PrjCfg.h` file): definitions required to enable `PROTOCOL_PRIME_API`, `PROTOCOL_PRIME_SNIFFER` and `PROTOCOL_PRIME_MNG` and serial port configuration to a specific serial port with speed 115200 bps. Serial port selection will be overwritten with a specific command.
- USI user functions: implement the functions `addUsi_Open()`, `addUsi_TxMsg()` and `addUsi_RxChar()`, which are required for the USI implementation.
- Call the USI host processing periodically (`addUsi_Process()` function). In this example, an independent thread in the Linux operating system calls “`usi_process_thread()`” every 1 millisecond. It could be improved using a select on serial port file descriptor.
- Synchronous mechanism implementation for PRIME API requests defining the `addUsi_WaitProcessing()` function with a “while loop” with timeout. This is not a problem working in Linux running a multi-process environment but could be improved using events.

The second step working with USI Host serialization is to identify which files are involved in the interface of the application:

- The `src\ifacePrime_api.c` supports the serialization of the PRIME API primitives for interacting with the PRIME Base Node Modem embedded project; it contains the functions and callbacks available for the USI protocol `PROTOCOL_PRIME_API`. This protocol includes the full functionalities needed to handle the Microchip PRIME Stack.
- The `src\ifaceMngLayer.c` refers to the different protocols defined in the Serial Communication Profile of the Management Plane described in the PRIME Specification and corresponds with the USI protocols `PROTOCOL_MNGP_PRIME_XXX`.
- The `src\ifacePrimeSniffer.c` supports the serialization of the PRIME FW Stack in terms of received PLC traffic from the PRIME Network.

The third step is to implement additional features for the board support package (BSP):

- Use of the Linux SYS Filesystem to act over microprocessor GPIOs (`primeBN_linux\source\port\common\gpio.c`) and LEDs (`primeBN_linux\source\port\common\led.c`)

Finally, the structure of the example itself includes these files:

- `userFnc.c`: specific implementation of user USI functions for this application

- `primeBN_linux\base_node_dlmsotcp.[c,h]`: DLMS over TCP functionality
- `primeBN_linux\prime_sniffer.[c,h]`: Sniffer capabilities in the Base Node, they include:
 - Logging to a file: on ATPL format, can be imported on the PLC Sniffer Tool
 - Logging to a TCP Server: for a connection from the PLC Sniffer Tool (only accepts 1 connection)
- `primeBN_linux\prime_log.[c,h]`: Logging capabilities on a file (the default filepath is `/tmp/prime.log`). There are four levels of messages depending on the relevance:
 - LOG_INFO: Information messages about normal execution of the application
 - LOG_ERR: Information about errors on application execution
 - LOG_DBG: Debugging information for greater knowledge of application execution
 - LOG_WARN: Warning messages
- `primeBN_linux\return_codes.h`: Common return values
- `primeBN_linux\prime_utils.[c,h]`: useful functions common to the whole application
- `primeBN_linux\base_node_network.[c,h]`: functions for saving information about what is happening on the PRIME network:
 - Registered device information
 - MAC connections information
 - 4-32 CL connections information
- `primeBN_linux\prime_bmng_network_events.[c,h]`: network events callback to save them in a SQLITE3 database (the default filepath is `/etc/config/prime_network_events.sql`)
- `primeBN_linux\base_node_mng_fw_upgrade.[c,h]`: functions related to the network firmware upgrade
- `primeBN_linux\base_node_mng.[c,h]`: callback functions definitions
- `primeBN_linux\base_node_vty.[c,h]`: specific commands for this application

4. Building Microchip PRIME Base Node on Linux Application

1. The PRIME Base Node application example is included in the `usi_host-2.2.2` package:

```
$ cd ~/010-usi-host/primeBN_linux
```

2. Select to configure the cross-compiler environment variable `CROSS_COMPILE` if needed:

```
$ export CROSS_COMPILE=~/buildroot-at91/output/host/bin/arm-buildroot-linux-gnueabihf-
```

3. Clean and compile the code:

```
$ make clean && make all
```

4. Copy the compiled application to the target:

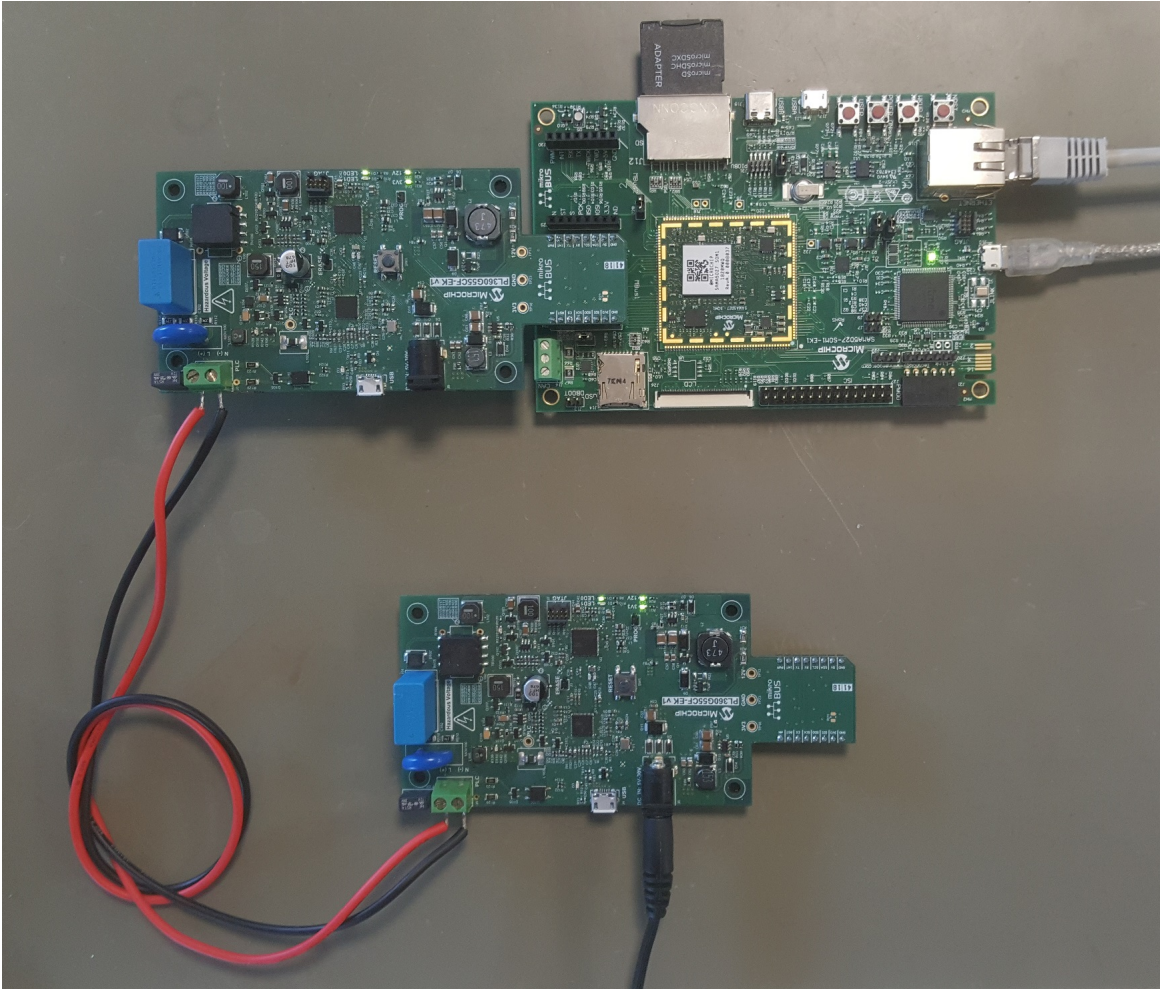
```
$ scp bn_prime root@<target-ip-addr>:/usr/bin
```

5. Start-up PRIME Network Manager on Linux

A fully compliant PRIME v1.4 network is required to run and evaluate the Linux PRIME Network Manager application. At least, it must be composed of:

- PRIME Network Manager: running in the SAMA5D27-SOM1-EK1 Linux board connected to a PL360G55CF-EK (which runs the PRIME Base Node Modem Embedded project modified as explained in [2.1. PRIME Base Node Modem Embedded Project](#)), it controls and manages the network formation
- PRIME Service Node: running on a PL360G55CF-EK, it joins the PRIME network and includes sniffer capabilities

Figure 5-1. Simple PRIME Network



The PRIME Network Manager Linux application “bn_prime” allows some arguments to be passed on execution:

```
$bn_prime -h
Usage : ./bn_prime [OPTION...]
-b, --boot Execute boot startup configuration
-e, --eval Execute argument as command
-c, --config Load the config file, default /etc/config/prime.conf
-l, --loglevel Loglevel <0-3>, default 3 (LOG_INFO)
-f, --logfile Logfile, default /tmp/prime.log
-v, --version Show the version
-h, --help Display this help and exit
```

The application configuration is based on a file saved by default in “/etc/config/prime.conf” but it can be overwritten with the argument “-c | --config PATH_TO_CONFIG_FILE” or with the environment variable PRIME_CONFIG. The most important configuration commands are:

- `config modemport`: configures the interface with Base Node Modem.
- `config mode`: configures the functionality of the Base Node
- `config tx_channel`: configures the PRIME Channel

This is an example of a configuration file:

```
#Writed on Thu Nov 19 11:37:16 2020
hostname MCHP-PRIME-BASE
!
prime
config modemport serial ttyS1 speed 115200
config mode base
config sniffer_log tcp port 4444
config sniffer_log tcp enabled
config sniffer_log logfile enabled
config sniffer_log enabled
config tx_channel 1
```

The path of the application log file is “/tmp/prime.log” by default. It can be overwritten with the argument “-f | --logfile PATH_TO_LOG_FILE” or with the environment variable PRIME_LOGFILE.

Default loglevel is LOG_INFO (according to [6.2.4. config loglevel](#)) but it could be overwritten with the argument “-l | --loglevel LOG_LVL”.

6. Base Node Command Line Interface

The Linux application includes a Command Line Interface (CLI), which is useful to interact with the PRIME Network in real time. This section describes the different commands implemented. Consider all of this functionality as an example of an application for testing purposes and not as a definitive piece of software.

The command line includes a PRIME node accessible from the configuration menu for everything related with the PRIME network and other simple commands (like hostname, write, ping, ip,...) located on the generic nodes of CLI (show/enable/config).

CLI offers the possibility to save the configuration when any modification is finished running the following:

```
MCHP-PRIME-BASE> enable
MCHP-PRIME-BASE# configure terminal
MCHP-PRIME-BASE(config)#write memory
Configuration saved to /etc/config/prime.conf
```

To access specific PRIME commands, go to the PRIME node running the following:

```
MCHP-PRIME-BASE> enable
MCHP-PRIME-BASE# configure terminal
MCHP-PRIME-BASE(config)# prime
MCHP-PRIME-BASE(prime)#
```

Keying '?' shows the specific commands for the PRIME node.

```
MCHP-PRIME-BASE(prime)#
config          PRIME configuration
end             End current mode and change to enable mode
exit           Exit current mode and down to previous mode
fw-upgrade     Firmware Upgrade
mac            MAC Attribute
mlme           PRIME MAC MLME Request Primitives
network        PRIME Network
phy            PRIME PHY PLME Request Primitives
pib            PIB Attribute
reboot         Reboot PRIME Modem
reset          Reset
show           Show PRIME
MCHP-PRIME-BASE(prime)#
```

6.1 SHOW commands

The commands to show the information/configuration about the PRIME PLC Base/Service Node and PRIME Network:

```
MCHP-PRIME-BASE(prime)# show
config          PRIME Configuration Information
dlmsotcp        PRIME DLMSoTCP
fw-upgrade     Firmware Upgrade
info           PRIME Information
mac            PRIME MAC
network        PRIME Network
phy            PRIME PHY
sniffer_log    PRIME Sniffer Log
status         PRIME Status
target         Target SN
```

6.1.1 show info

It shows the global PRIME Base Node information:

```
MCHP-PRIME-BASE(prime)# show info
PRIME Information
* Vendor ID    : 0x0000
* Product ID   : 0x3730
```

```
* FW Version      : B14.03.03
* PHY Version     : 0x23000303
* HOST Version    : 0x00000000
* MAC Version     : 0xC331460B
* 4-32 Version   : 0xC3300000
```

6.1.2 show status

It shows the global PRIME Base Node status:

```
MCHP-PRIME-BASE(prime)# show status
PRIME Status
```

6.1.3 show config

It shows the global configuration of the PRIME Base Node:

```
MCHP-PRIME-BASE(prime)# show config
PRIME Main Configuration
* EUI48          : 016020110014
* Mode           : BASE
* Sec Profile    : 0
* Band Plan     : 255
* Channel        : 1
PRIME Connection Configuration
* Modemport     : TCP/IP
* Remote IP     : 192.168.1.2
* Remote Port   : 8214
PRIME Sniffer Log Configuration
* Enabled       : Yes
* Logfile       : Yes
* TCP Server    : Yes
* TCP Port      : 4444
PRIME DLMSoTCP
* Enabled       : Yes
* TCP Port      : 4059
```

6.1.4 show dlmsotcp config

It shows the configuration of the DLMSoTCP service:

```
MCHP-PRIME-BASE(prime)# show dlmsotcp config
PRIME DLMSoTCP Configuration:
* Enabled       : Yes
* TCP Port      : 4059
```

6.1.5 show sniffer_log config

It shows the configuration of the sniffer log service:

```
MCHP-PRIME-BASE(prime)# show sniffer_log config
PRIME Sniffer Log Configuration:
* Enabled       : Yes
* Logfile       : Yes
* TCP Server    : Yes
* TCP Port      : 4444
```

6.1.6 show mac statistics

It shows the MAC layer statistics based on standard counters:

```
MCHP-PRIME-BASE(prime)# show mac statistics
PRIME MAC Counters
* macTxDataPktCount : 877168
* MacRxDataPktCount : 59
* MacTxCtrlPktCount : 197168
* MacRxCtrlPktCount : 774703
* MacCSMAFailCount  : 0
* MacCSMAChBusyCount : 20250
```

6.1.7 show phy statistics

It shows the PHY layer statistics based on standard counters:

```
MCHP-PRIME-BASE(prime)# show phy statistics
PRIME PHY Counters
* phyStatsCRCIncorrectCount: 45950
* phyStatsCRCFailCount      : 0
* phyStatsTxDropCount       : 1518
* phyStatsRxDropCount       : 0
* phyStatsRxTotalCount      : 774770
```

6.1.8 show network

The commands to show the network map:

```
MCHP-PRIME-BASE(prime)# show network
available_switches Available Switches
cl432_connections  4-32 Connections
level              PRIME Network Level
registered_devices Registered Devices
topology           Network Topology
```

6.1.8.1 show network cl432_connections

It shows the network Converge Layer 4-32 Connections:

```
MCHP-PRIME-BASE(prime)# show network cl432_connections
IEC 61334-4-32 ACTIVE CONNECTIONS:
-----
EUI48          ADDRESS  SERIAL/NUMBER
-----
0x013090320032 0x03E9  ATM3090320032
0x013090320029 0x03EA  ATM3090320029
0x013090320035 0x03EB  ATM3090320035
0x013090320030 0x03EC  ATM3090320030
0x013090320036 0x03ED  ATM3090320036
0x013090320031 0x03EE  ATM3090320031
0x010451210002 0x03EF  ATM0451210002
0x013090320044 0x03F0  ATM3090320044
0x013090320038 0x03F1  ATM3090320038
0x013090320043 0x03F2  ATM3090320043
[...]
```

6.1.8.2 show network available_switches

It shows the switches in the PRIME network:

```
MCHP-PRIME-BASE(prime)# show network available_switches
EUI48          Level  LSID  RxLvl  RxSNR
-----
0x013090320046 000  000128  000  000
0x013090320059 001  000001  000  000
0x013090320061 002  000002  000  000
0x013090320087 003  000003  000  000
```

6.1.8.3 show network level LEVEL registered_devices

It shows the registered nodes present in a specific PRIME network level:

```
MCHP-PRIME-BASE(prime)# show network level 0 registered_devices
EUI48          LNIID  State  LSID  SID  Level  TCap  SwCap
-----
0x013090320032 14001  Terminal  255  000  000  0x65  0x00
0x013090320029 14002  Terminal  255  000  000  0x65  0x00
0x013090320035 14003  Terminal  255  000  000  0x65  0x00
0x013090320030 14004  Terminal  255  000  000  0x65  0x00
0x013090320036 14005  Terminal  255  000  000  0x65  0x00
0x013090320031 14006  Terminal  255  000  000  0x65  0x00
0x010451210002 14007  Terminal  255  000  000  0x65  0x09
0x013090320038 14009  Terminal  255  000  000  0x65  0x09
0x013090320043 14010  Terminal  255  000  000  0x65  0x09
```

```

0x013090320037 14011 Terminal 255 000 000 0x65 0x09
0x013090320046 14012 Switch 128 000 000 0x65 0x09
[...]

```

6.1.8.4 show network registered_devices

It shows the registered devices in the PRIME Network:

```

MCHP-PRIME-BASE (prime) # show network registered_devices
EUI48          LNID      State   LSID   SID Level TCap SwCap
-----
0x013090320032 14001  TERMINAL 255 000 000 0x65 0x00
0x013090320029 14002  TERMINAL 255 000 000 0x65 0x00
0x013090320035 14003  TERMINAL 255 000 000 0x65 0x00
0x013090320030 14004  TERMINAL 255 000 000 0x65 0x00
0x013090320036 14005  TERMINAL 255 000 000 0x65 0x00
0x013090320031 14006  TERMINAL 255 000 000 0x65 0x00
0x010451210002 14007  TERMINAL 255 000 000 0x65 0x09
0x013090320038 14009  TERMINAL 255 000 000 0x65 0x09
0x013090320043 14010  TERMINAL 255 000 000 0x65 0x09
0x013090320037 14011  TERMINAL 255 000 000 0x65 0x09
0x013090320046 14012  SWITCH   128 000 000 0x65 0x09
0x013090320026 14013  TERMINAL 255 000 000 0x65 0x00
0x013090320027 14014  TERMINAL 255 000 000 0x65 0x00
0x013090320048 14015  TERMINAL 255 000 000 0x65 0x09
0x013090320025 14017  TERMINAL 255 000 000 0x65 0x00
0x013090320034 14018  TERMINAL 255 000 000 0x65 0x09
0x013090320044 14008  TERMINAL 255 128 001 0x65 0x00
0x013090320047 14016  TERMINAL 255 128 001 0x65 0x00
0x013090320051 14019  TERMINAL 255 128 001 0x65 0x09
0x013090320049 14020  TERMINAL 255 128 001 0x65 0x00
0x013090320059 14021  SWITCH   001 128 001 0x65 0x09
0x013090320058 14022  TERMINAL 255 128 001 0x65 0x09
[...]

```

6.1.8.5 show network topology

It shows the PRIME Network topology:

```

MCHP-PRIME-BASE (prime) # show network topology
State  EUI48          LNID      LSID   SID Level TCap SwCap
-----
BASE   202102120001  00000    0000  000  0    --  --
SWITCH 013090320046  14012    128  000  000  0x65 0x09
TERMINAL 013090320032  14001    255  000  000  0x65 0x00
TERMINAL 013090320029  14002    255  000  000  0x65 0x00
SWITCH 013090320059  14021    001  128  001  0x65 0x09
TERMINAL 013090320044  14008    255  128  001  0x65 0x00
TERMINAL 013090320047  14016    255  128  001  0x65 0x00
SWITCH 013090320061  14033    002  001  002  0x65 0x09
TERMINAL 013090320066  14028    255  001  002  0x65 0x09
TERMINAL 013090320053  14029    255  001  002  0x65 0x00
SWITCH 013090320087  14043    003  002  003  0x65 0x09
TERMINAL 013090320088  14039    255  002  003  0x65 0x09
TERMINAL 013090320075  14040    255  002  003  0x65 0x09
TERMINAL 013090320092  14057    255  003  004  0x65 0x00
TERMINAL 013090320098  14058    255  003  004  0x65 0x00
Generated topology file. Go to http://www.webgraphviz.com/ for visualization.

```

A graphviz file is generated in order to show the topology on a graph.

```

digraph Topology {
ranksep=3;
ratio=auto;
202102120001 [shape=diamond]
013090320046 [shape=box]
202102120001 -> 013090320046
202102120001 -> 013090320032
202102120001 -> 013090320029
013090320059 [shape=box]
013090320046 -> 013090320059
013090320046 -> 013090320044
013090320046 -> 013090320047
013090320061 [shape=box]

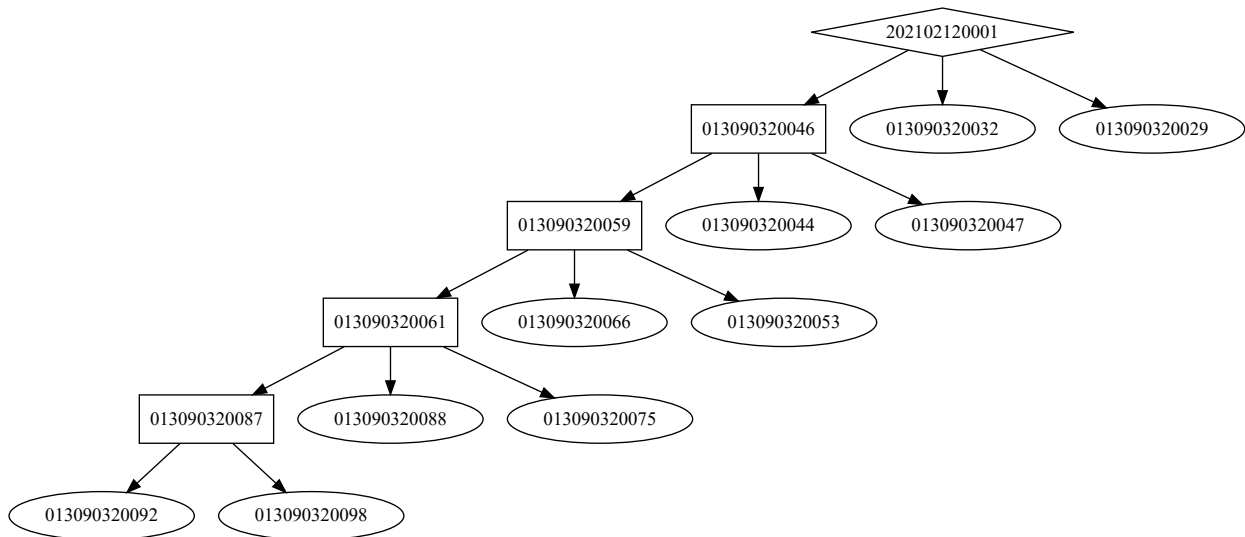
```

```

013090320059 -> 013090320061
013090320059 -> 013090320066
013090320059 -> 013090320053
013090320087 [shape=box]
013090320061 -> 013090320087
013090320061 -> 013090320088
013090320061 -> 013090320075
013090320087 -> 013090320092
013090320087 -> 013090320098
}

```

Figure 6-1. Network Topology



6.1.9 show fw-upgrade

It shows the FW Upgrade configuration and status:

```

MCHP-PRIME-BASE(prime)# show fw-upgrade
binarypath      FW Upgrade Binary File Path
data_info       Firmware Upgrade data info
match_rule      Firmware Upgrade match rule
options          Target List
signature_options FW Upgrade Signature Options
state           FW Upgrade State
target_list     Target List
version         FW Upgrade Version

```

6.1.9.1 show fw-upgrade target_list

It shows the list of PRIME Service Nodes included in a FW Upgrade process. The target list is sent to the Base Node only when the fw-upgrade download command is executed:

```

MCHP-PRIME-BASE(prime)# show fw-upgrade target_list
EUI48          Vendor      Model          Version
-----
0x013090320055 MCHP      ATPL360MB s140302
0x013090320049 MCHP      ATPL360MB s140302
0x013090320058 MCHP      ATPL360MB s140302
0x013090320054 MCHP      ATPL360MB s140302
0x013090320051 MCHP      ATPL360MB s140302
0x013090320052 MCHP      ATPL360MB s140302

```

6.1.9.2 show fw-upgrade state MAC

It shows the FW Upgrade progress state for a Service Node:

```

MCHP-PRIME-BASE(prime)# show fw-upgrade state 013090320056
EUI48          STATE      PAGES
-----

```

```

0x013090320056          IDLE 00000
MCHP-PRIME-BASE(prime)# show fw-upgrade state 013090320057
EUI48                   STATE      PAGES
-----
0x013090320056          RECEIVING 00085

```

The possible state values are:

- IDLE: No firmware upgrade in progress
- RECEIVING: Receiving FW Upgrade Binary from PRIME Base Node
- COMPLETE: Received complete FW Upgrade Binary
- COUNTDOWN: Waiting to execute FW Upgrade Binary
- UPGRADE: FW Upgrade process success
- EXCEPTION: FW Upgrade process exception
- UNKNOWN: FW Upgrade process state is unknown

The pages indicate the number of pages received in the FW Upgrade process. In case of an exception, they indicate the type of exception.

The exception can be:

- PERMANENT
- TEMPORARY

The possible exception values are:

- GENERAL: problem to get buffer for bitmap
- PROTOCOL: firmware upgrade image size is bigger than space available or page size is wrong
- CRC: error in firmware upgrade image CRC
- IMAGE: the firmware upgrade image is wrong
- SIGNATURE: the firmware upgrade signature is wrong
- TIMER_EXPIRED: timeout of firmware upgrade process without success
- UNKNOWN: firmware upgrade exception because of unknown situation

6.1.9.3 show fw-upgrade signature_options

It shows the Firmware upgrade signature options:

```

show fw-upgrade signature_options
Firmware Upgrade Signature 0 with length 0

```

The signature values are:

- 0: No signature
- 1: RSA-SHA signature
- 2: ECDSA-SHA signature

The signature length depends on the firmware binary size and signature (0 if no signature).

6.1.9.4 show fw-upgrade options

It shows the firmware upgrade options:

```

MCHP-PRIME-BASE(prime)# show fw-upgrade options
Firmware Upgrade Options
- Multicast      :Yes
- ARQ            :No
- Pagesize      :0 bytes
- Delay Restart  :60 s
- Safety Timer   :21600 s

```

where:

- Multicast: is multicasting to transfer the firmware image through the PLC enabled?
- ARQ: is ARQ enabled?
- Pagesize: size of the data chunks used during the PLC transmission phase of the FW Upgrade
- Delay Restart: time that a Service Node waits before restarting with the new image

- Safety timer: time a Service Node must wait before reverting to a former firmware image when the new image is not confirmed by the Base Node

6.1.9.5 show fw-upgrade version MAC

It shows the vendor, model and version for a specific MAC:

```
MCHP-PRIME-BASE (prime)# show fw-upgrade version 013090320056
EUI48          Vendor          Model          Version
-----
0x013090320056      MCHP          ATPL360MB    s140302
```

The information is requested in real time.

6.1.9.6 show fw-upgrade match_rule

It shows which nodes will be upgraded based on their Vendor and Model:

```
MCHP-PRIME-BASE (prime)# show fw-upgrade match_rule
Firmware Upgrade Match Rule:
- Model :No
- Vendor :No
```

6.1.9.7 show fw-upgrade data_info

It shows the firmware upgrade image data information:

```
MCHP-PRIME-BASE (prime)# show fw-upgrade data_info
Firmware Upgrade Data Info:
- Vendor :MCHP
- Model :ATPL360MB
- Version :S14.03.22
```

6.1.9.8 show fw-upgrade binarypath

It shows the path to the location of the firmware binary for an FW Upgrade:

```
show fw-upgrade binarypath
Firmware Upgrade Binary Path :/etc/config/firmwareS14.03.02.bin
```

6.2 CONFIG commands

Global configuration commands:

```
MCHP-PRIME-BASE (prime)# config
tx_channel    PRIME channel configuration
dlmsotcp     PRIME DLMS over TCP
log          PRIME log
loglevel     PRIME loglevel
mode         PRIME Configuration mode
modemport    PRIME Modem connection
sniffer_log  PRIME Sniffer Log
```

6.2.1 config tx_channel

It configures the active PRIME channel <1-8>. It depends on:

- Band Plan: set of bands that a device is configured to operate on. It is defined on the PRIME firmware as USER_BAND_PLAN (default all bands/channels enabled).
- Hardware capabilities: the band plan and the performance of the system depend on the coupling

6.2.2 config dlmsotcp

This command enables the DLMS over TCP extension for optimal multiplexing 4-32 connections over one connection. This allows a remote concentrator to connect to the PRIME Manager/Base Node and gather DMLS/COSEM data directly from the Service Nodes that implement such a layer. When enabled, the application opens a

TCP server in a port previously configured (default 4059) and waits for a connection from a remote concentrator. Only one remote connection is allowed.

```
MCHP-PRIME-BASE(prime)# config dlmsotcp
disabled PRIME DLMS over TCP disabled
enabled PRIME DLMS over TCP enabled
port DLMS over TCP Port
```

6.2.3 config log

It enables/disables the Global PRIME Base Node log:

```
MCHP-PRIME-BASE(prime)# config log
disabled PRIME log disabled
enabled PRIME log enabled
```

6.2.4 config loglevel

It configures the PRIME Base Node loglevel of logging. It is possible to configure different levels of the log for different features.

```
MCHP-PRIME-BASE(prime)# config loglevel
global PRIME Global loglevel
sniffer PRIME Sniffer loglevel
dlmsotcp PRIME DLMSoTCP loglevel
net_events PRIME Network Events loglevel
```

Available values are:

- 0: NONE
- 1: FATAL
- 2: ERROR
- 3: INFO
- 4: DEBUG

6.2.5 config mode

This command sets the configuration of the PRIME Application:

```
MCHP-PRIME-BASE(prime)# config mode
base PRIME Base Configuration
service PRIME Service Configuration
```

Only the Base Node is developed but a Service Node can be implemented too.

6.2.6 config modempport

It configures how the application is connected to the Base Node:

- Physical Serial Port connection (serial): the host where the application runs has a serial port connection to a Base Node identified by the physical device and the baudrate.

```
MCHP-PRIME-BASE(prime)# config modempport serial ttyS6 speed 115200
PRIME Modem connected to serial port ttyS6 at 115200 bps
```

- TCP Connection (tcp): the host where the application runs over a TCP port where USI frames are redirected. It allows remote connections to the PRIME Base Node. Remote serial port to TCP Port redirection can be done easily with applications like “socat” (<http://www.dest-unreach.org/socat/>) or Windows applications like COMbyTCP. For example, running the next socat command, it redirects the ttyS1 serial port to the TCP 3000 port:

```
- socat -d -x TCP-LISTEN:3000,reuseaddr /dev/ttyS1,B230400,raw,echo=0
```

```
MCHP-PRIME-BASE(prime)# config modempport tcp 192.168.137.1 port 8214
PRIME Modem connection to TCP host 192.168.137.1 port 8214
```

6.2.7 config sniffer_log

It configures the sniffer capabilities.

```
MCHP-PRIME-BASE (prime) # config sniffer_log
disabled PRIME Sniffer Log disabled
enabled PRIME Sniffer Log enabled
logfile PRIME Sniffer Log logfile
tcp PRIME Sniffer Log TCP
```

The sniffer log can be saved in a file or be redirected to a TCP Port.

- Logfile: created with the ATPL format, and can be opened with the PLC Sniffer application (File\Import ATPL Log...)
- TCP Port redirection: redirects Sniffer USI frames to be decoded in real time with the PLC Sniffer application in a remote computer

Figure 6-2. How to Import Sniffer Log on MCHP PLC Sniffer

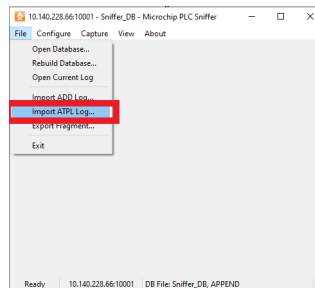
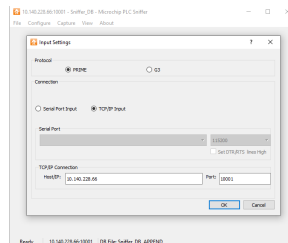


Figure 6-3. How to Connect MCHP PLC Sniffer to PRIME Base Node through TCP



6.3 PIB commands

The application implements a way to get/set PIBs from the Base Node. The following interfaces are available:

- Local Requests:
 - Serial Profile
 - PRIME API
- Remote Requests:
 - Base Management based on PRIME API

It allows access to PIBs from different layers:

```
MCHP-PRIME-BASE (prime) # pib
app APP Attribute
cl432 CL4-32 Attribute
mac MAC Attribute
mtp MTP Attribute
phy PHY Attribute
```

For read-write PIBs, it is possible to get and set the PIB value:

```
MCHP-PRIME-BASE (prime) # pib mac
get Get PIB Attribute
set Set PIB Attribute
```

Locally, it is possible to access the base node modem through the management plane on the serial profile (mngp) or PRIME_API (mlme); on the other hand, the base management plane (bmng) is used for remote requests.

```
MCHP-PRIME-BASE (prime) # pib mac get
mlme Request via MLME primitive
mngp Request via MNGP
bmng Request via Base Management PRIME Profile
```

6.4 PHY commands

It implements some of the PHY management primitives described on the PRIME specification.

```
MCHP-PRIME-BASE (prime) # phy
reset PRIME PLME Reset Request
resume PRIME PLME Resume Request
sleep PRIME PLME Sleep Request
```

6.4.1 phy reset

It resets the PRIME Physical Layer.

6.4.2 phy reset statistics

It resets the PRIME Physical statistics.

6.4.3 phy resume

It returns from sleep mode to normal mode on the PRIME Physical layer.

6.4.4 phy sleep

It enables the sleep mode on the PRIME Physical Layer.

6.5 MAC commands

It implements some generic MAC features:

```
MCHP-PRIME-BASE (prime) # mac
action Action PIB Attribute
demote PRIME MLME Demote Request
promote PRIME MLME Promote Request
reset PRIME MAC Reset Request
```

6.5.1 mac action

The MAC action commands are useful for testing MAC features on a PRIME Network:

```
MCHP-PRIME-BASE (prime) # mac action mlme|mngp
macActionAliveTime macActionAliveTime
macActionAppemuDataBurst macActionAppemuDataBurst
macActionBroadcastDataBurst macActionBroadcastDataBurst
macActionConnClose macActionConnClose
macActionDemote macActionDemote
macActionMgmtCon macActionMgmtCon
macActionMgmtDataBurst macActionMgmtDataBurst
macActionMgmtMul macActionMgmtMul
macActionProReject macActionProReject
macActionPromote macActionPromote
macActionRegReject macActionRegReject
macActionReject macActionReject
macActionSegmented432 macActionSegmented432
macActionTxData macActionTxData
macActionUnregister macActionUnregister
macActionUnregisterBN macActionUnregisterBN
```

6.5.2 mac demote

The MLME demote request primitive is used to trigger a demotion process in a Service Node that is in a Switch functional state. This primitive may be used by management entities to enforce demotion in cases where the Node's default functionality does not automatically perform the process.

Not useful in this example where a base node is implemented.

6.5.3 mac promote

This command corresponds with MLME promote. The MLME promote request primitive is used on the Base Node to ask a node to change its state from Terminal to Switch or, if the node is already in the Switch state, to adopt a new Beacon PDU modulation scheme. The modulation scheme options are:

- dbpsk_f - 4
- null - 0
- r_dbpsk - 8
- r_dqpsk - 9

6.5.4 mac reset

It resets the PRIME MAC Layer.

6.5.5 mac reset statistics

It resets the PRIME MAC statistics.

6.6 TARGET commands

This section includes some applications to get information from Service Nodes on the Base Node.

```
MCHP-PRIME-BASE(prime)# target MAC
zero-cross Zero Crossing PRIME Modem request
```

The zero-cross function allows access to information about the time difference between devices and the main frequency.

```
MCHP-PRIME-BASE(prime)# target 0x013090320098 zero-cross
- Status:UNKNOWN
- Terminal Block:1
- Direction:RAISING
- ZCT: 1591571730 us
- TIME_FREQ: 50 Hz
- TIME_DIFF: 19960 us
```

6.7 RESET command

It resets the PRIME Physical and MAC statistics.

6.8 REBOOT command

It reboots the local PRIME modem or the remote Service Node identified by a MAC address.

6.9 NETWORK commands

These commands allow administrative service nodes to be added/deleted in the PRIME network. To register the Service Nodes when the security profile is 1 or 2, it is necessary to configure the Device Unique Key (DUK) specific for every Service Node. This can be combined with PRIME whitelists (not implemented) for administrative management.

6.9.1 network add

This command adds an administrative Service Node EUI48 to the PRIME Network. It allows the Service Node in the Base Node to be registered. When security is disabled in the Network, all the Service Nodes are registered by default.

```
network add MAC key
    none      No security
    default   Security Key default DUK
    DUK      Security Key DUK value
```

- none – No security
- default – Security Key by default formed as EUI48+EUI48+FFFFFFFF (16 Bytes)
- DUK – Security Key DUK value

6.9.2 network del

It removes the service node from the PRIME network linked list.

6.10 FW UPGRADE commands

The commands related to the FW Upgrade of the PRIME Network:

```
MCHP-PRIME-BASE(prime) # fw-upgrade
  abort      Abort Target Firmware Upgrade
  add        Add Target to Firmware Upgrade mechanism
  binarypath FW Upgrade Binary File
  clear_target_list Clear Firmware Upgrade Target List
  download   FW Upgrade Download
  image_info Firmware Upgrade Image Info
  match_rule Firmware Upgrade Match Rule
  options    Firmware Upgrade Options
  signature_algorithm FW Upgrade Signature Algorithm
  start      FW Upgrade Start
  status     FW Upgrade Status
```

6.10.1 fw-upgrade add all

It adds all the MAC addresses of the registered service nodes in the Base Node to the FW Upgrade process.

6.10.2 fw-upgrade add MAC

It adds the MAC address to the fw-upgrade list in the Base Node. The MAC address must correspond with a registered device in the PRIME network.

6.10.3 fw-upgrade binarypath FILE

It configures the location in the file system of the firmware upgrade binary file (preferably on /etc/config - CONFIG_DIR):

```
fw-upgrade binarypath /etc/config/firmwareS14.03.02.bin
```



The binary file **MUST** exist, so it **MUST** be transferred to the local path before being used (through any mechanism like SCP, TFTP, FTP, USB Flash Disk...).

6.10.4 fw-upgrade clear_target_list

It clears the target list of service nodes to be upgraded.

6.10.5 fw-upgrade download

This command triggers the process to download the binary file programmed with the `fw-upgrade binarypath` command to the Base Node.

6.10.6 fw-upgrade image_info

It configures the parameters that define a firmware version. They can be found on the `conf_prime_stack.h` file of the `prime_service_bin` project:

- vendor (PRIME_FW_VENDOR): MCHP
- model (PRIME_FW_MODEL): [PL360MB|PL360G55CF|SAM4CP16BMB]
- version (PRIME_FW_VERSION): corresponds with the PRIME Stack Release

All of this information can be modified by customers to adapt it to their needs.

6.10.7 fw-upgrade match_rule

It configures the firmware upgrade matching rules to select which devices should be upgraded depending on the Model and Vendor values (the corresponding bits of 0000 0MV0 are set).

```
MCHP-PRIME-BASE(prime)# fw-upgrade match_rule
both    Firmware Upgrade filter by Model and Vendor
model   Firmware Upgrade filter by Model
none    Firmware Upgrade without filter
vendor  Firmware Upgrade filter by Vendor
```

6.10.8 fw-upgrade options

It configures the Firmware upgrade options:

```
MCHP-PRIME-BASE(prime)# fw-upgrade options multicast arq disabled page_size 0 delay 60 timer 21600
```

- Multicast: enables multicasting to transfer the firmware image through the PLC
- ARQ: enables ARQ
- Pagesize: size of the data chunks used during the PLC phase of the FU
 - automatic (0)
 - 32 bytes (1)
 - 64 bytes (2)
 - 128 bytes (3)
 - 192 bytes (4)
- delay: time that a Service Node waits before restarting with the new image (0-65536 seconds)
- timer: time that a Service Node must wait before reverting to a former firmware image when the new image is not confirmed by the Base Node; so, this is the time to test the new firmware (0-65536 seconds)

6.10.9 fw-upgrade signature_algorithm

This command configures the signature algorithm used in the generation of the firmware upgrade binary.

```
MCHP-PRIME-BASE(prime)# fw-upgrade signature_algorithm 0 size 0
```

Signature values are:

- 0: No signature
- 1: RSA-SHA signature
- 2: ECDSA-SHA signature

Signature size depends on the firmware binary length and signature (0 if no signature)

6.10.10 fw-upgrade start

This command starts the Firmware Upgrade process after the `fw-upgrade download` command.

6.10.11 fw-upgrade status

It shows the status of a fw-upgrade process:

- IDLE: no firmware upgrade process in progress
- RECEIVING: firmware upgrade process is in progress

- FINISHED: firmware upgrade process has finished

An example of the firmware upgrade process in progress:

EUI48	STATE	PAGES
0x013090320097	RECEIVING	00085
0x013090320098	RECEIVING	00091

6.10.12 fw-upgrade abort MAC

This command aborts a fw-upgrade process running for a specific Service Node (MAC).

6.10.13 fw-upgrade abort all

It aborts the fw-upgrade process for all the devices in the FW Upgrade target list.

6.10.14 fw-upgrade examples

The next codeblock shows the fw-upgrade command execution for a FW Upgrade Protocol when no security is implemented and no signature is configured:

```
fw-upgrade options multicast arq disabled page_size 0 delay 60 timer 21600
fw-upgrade clear_target_list
fw-upgrade add fcc23d018a46
fw-upgrade match_rule none
fw-upgrade image_info vendor MCHP model PL360MB version S14.03.03
fw-upgrade binarypath /etc/config/firmware_S14.03.03.bin
fw-upgrade signature_algorithm 0 size 0
fw-upgrade download
fw-upgrade start
show fw-upgrade state fcc23d018a46
```

If the security profile is 1 or 2, the Service Node has to be added to the network before. The process for signed firmware binary construction can be found on the PRIME Manager User manual:

```
network add fcc23d018a46 key fcc23d018a46fcc23d018a46ffffffff
fw-upgrade options multicast arq disabled page_size 0 delay 60 timer 21600
fw-upgrade clear_target_list
fw-upgrade add fcc23d018a46
fw-upgrade match_rule none
fw-upgrade image_info vendor MCHP model PL360MB version S14.03.03
fw-upgrade binarypath /etc/config/firmware_S14.03.03.ECDSA.bin
fw-upgrade signature_algorithm 2 size 71
fw-upgrade download
fw-upgrade start
show fw-upgrade state fcc23d018a46
```

7. Revision History

7.1 Rev A - 12/2021

Document	Initial document release.
----------	---------------------------

The Microchip Website

Microchip provides online support via our website at www.microchip.com/. This website is used to make files and information easily available to customers. Some of the content available includes:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQs), technical support requests, online discussion groups, Microchip design partner program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

Product Change Notification Service

Microchip's product change notification service helps keep customers current on Microchip products. Subscribers will receive email notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, go to www.microchip.com/pcn and follow the registration instructions.

Customer Support

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Embedded Solutions Engineer (ESE)
- Technical Support

Customers should contact their distributor, representative or ESE for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in this document.

Technical support is available through the website at: www.microchip.com/support

Microchip Devices Code Protection Feature

Note the following details of the code protection feature on Microchip products:

- Microchip products meet the specifications contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is secure when used in the intended manner, within operating specifications, and under normal conditions.
- Microchip values and aggressively protects its intellectual property rights. Attempts to breach the code protection features of Microchip product is strictly prohibited and may violate the Digital Millennium Copyright Act.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of its code. Code protection does not mean that we are guaranteeing the product is "unbreakable". Code protection is constantly evolving. Microchip is committed to continuously improving the code protection features of our products.

Legal Notice

This publication and the information herein may be used only with Microchip products, including to design, test, and integrate Microchip products with your application. Use of this information in any other manner violates these terms. Information regarding device applications is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. Contact your local Microchip sales office for additional support or, obtain additional support at www.microchip.com/en-us/support/design-help/client-support-services.

THIS INFORMATION IS PROVIDED BY MICROCHIP "AS IS". MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE, OR WARRANTIES RELATED TO ITS CONDITION, QUALITY, OR PERFORMANCE.

IN NO EVENT WILL MICROCHIP BE LIABLE FOR ANY INDIRECT, SPECIAL, PUNITIVE, INCIDENTAL, OR CONSEQUENTIAL LOSS, DAMAGE, COST, OR EXPENSE OF ANY KIND WHATSOEVER RELATED TO THE INFORMATION OR ITS USE, HOWEVER CAUSED, EVEN IF MICROCHIP HAS BEEN ADVISED OF THE POSSIBILITY OR THE DAMAGES ARE FORESEEABLE. TO THE FULLEST EXTENT ALLOWED BY LAW, MICROCHIP'S TOTAL LIABILITY ON ALL CLAIMS IN ANY WAY RELATED TO THE INFORMATION OR ITS USE WILL NOT EXCEED THE AMOUNT OF FEES, IF ANY, THAT YOU HAVE PAID DIRECTLY TO MICROCHIP FOR THE INFORMATION.

Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

Trademarks

The Microchip name and logo, the Microchip logo, Adaptec, AnyRate, AVR, AVR logo, AVR Freaks, BesTime, BitCloud, CryptoMemory, CryptoRF, dsPIC, flexPWR, HELDO, IGLOO, JukeBlox, KeeLoq, Klear, LANCheck, LinkMD, maXStylus, maXTouch, MediaLB, megaAVR, Microsemi, Microsemi logo, MOST, MOST logo, MPLAB, OptoLyzer, PIC, picoPower, PICSTART, PIC32 logo, PolarFire, Prochip Designer, QTouch, SAM-BA, SenGenuity, SpyNIC, SST, SST Logo, SuperFlash, Symmetricom, SyncServer, Tachyon, TimeSource, tinyAVR, UNI/O, Vectron, and XMEGA are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

AgileSwitch, APT, ClockWorks, The Embedded Control Solutions Company, EtherSynch, Flashtec, Hyper Speed Control, HyperLight Load, IntellIMOS, Libero, motorBench, mTouch, Powermite 3, Precision Edge, ProASIC, ProASIC Plus, ProASIC Plus logo, Quiet- Wire, SmartFusion, SyncWorld, Temux, TimeCesium, TimeHub, TimePictra, TimeProvider, TrueTime, WinPath, and ZL are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Adjacent Key Suppression, AKS, Analog-for-the-Digital Age, Any Capacitor, AnyIn, AnyOut, Augmented Switching, BlueSky, BodyCom, CodeGuard, CryptoAuthentication, CryptoAutomotive, CryptoCompanion, CryptoController, dsPICDEM, dsPICDEM.net, Dynamic Average Matching, DAM, ECAN, Espresso T1S, EtherGREEN, GridTime, IdealBridge, In-Circuit Serial Programming, ICSP, INICnet, Intelligent Paralleling, Inter-Chip Connectivity, JitterBlocker, Knob-on-Display, maxCrypto, maxView, memBrain, Mindi, MiWi, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, NVM Express, NVMe, Omniscient Code Generation, PICDEM, PICDEM.net, PICKit, PICTail, PowerSmart, PureSilicon, QMatrix, REAL ICE, Ripple Blocker, RTAX, RTG4, SAM-ICE, Serial Quad I/O, simpleMAP, SimpliPHY, SmartBuffer, SmartHLS, SMART-I.S., storClad, SQL, SuperSwitcher, SuperSwitcher II, Switchtec, SynchroPHY, Total Endurance, TSHARC, USBCheck, VariSense, VectorBlox, VeriPHY, ViewSpan, WiperLock, XpressConnect, and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

The Adaptec logo, Frequency on Demand, Silicon Storage Technology, Symmcom, and Trusted Time are registered trademarks of Microchip Technology Inc. in other countries.

GestIC is a registered trademark of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.

© 2021, Microchip Technology Incorporated and its subsidiaries. All Rights Reserved.

ISBN: 978-1-5224-9394-5

Quality Management System

For information regarding Microchip's Quality Management Systems, please visit www.microchip.com/quality.

Worldwide Sales and Service

AMERICAS	ASIA/PACIFIC	ASIA/PACIFIC	EUROPE
<p>Corporate Office 2355 West Chandler Blvd. Chandler, AZ 85224-6199 Tel: 480-792-7200 Tel: 480-792-7277 Technical Support: www.microchip.com/support Web Address: www.microchip.com</p> <p>Atlanta Duluth, GA Tel: 678-957-9614 Fax: 678-957-1455</p> <p>Austin, TX Tel: 512-257-3370</p> <p>Boston Westborough, MA Tel: 774-760-0087 Fax: 774-760-0088</p> <p>Chicago Itasca, IL Tel: 630-285-0071 Fax: 630-285-0075</p> <p>Dallas Addison, TX Tel: 972-818-7423 Fax: 972-818-2924</p> <p>Detroit Novi, MI Tel: 248-848-4000</p> <p>Houston, TX Tel: 281-894-5983</p> <p>Indianapolis Noblesville, IN Tel: 317-773-8323 Fax: 317-773-5453 Tel: 317-536-2380</p> <p>Los Angeles Mission Viejo, CA Tel: 949-462-9523 Fax: 949-462-9608 Tel: 951-273-7800</p> <p>Raleigh, NC Tel: 919-844-7510</p> <p>New York, NY Tel: 631-435-6000</p> <p>San Jose, CA Tel: 408-735-9110 Tel: 408-436-4270</p> <p>Canada - Toronto Tel: 905-695-1980 Fax: 905-695-2078</p>	<p>Australia - Sydney Tel: 61-2-9868-6733</p> <p>China - Beijing Tel: 86-10-8569-7000</p> <p>China - Chengdu Tel: 86-28-8665-5511</p> <p>China - Chongqing Tel: 86-23-8980-9588</p> <p>China - Dongguan Tel: 86-769-8702-9880</p> <p>China - Guangzhou Tel: 86-20-8755-8029</p> <p>China - Hangzhou Tel: 86-571-8792-8115</p> <p>China - Hong Kong SAR Tel: 852-2943-5100</p> <p>China - Nanjing Tel: 86-25-8473-2460</p> <p>China - Qingdao Tel: 86-532-8502-7355</p> <p>China - Shanghai Tel: 86-21-3326-8000</p> <p>China - Shenyang Tel: 86-24-2334-2829</p> <p>China - Shenzhen Tel: 86-755-8864-2200</p> <p>China - Suzhou Tel: 86-186-6233-1526</p> <p>China - Wuhan Tel: 86-27-5980-5300</p> <p>China - Xian Tel: 86-29-8833-7252</p> <p>China - Xiamen Tel: 86-592-2388138</p> <p>China - Zhuhai Tel: 86-756-3210040</p>	<p>India - Bangalore Tel: 91-80-3090-4444</p> <p>India - New Delhi Tel: 91-11-4160-8631</p> <p>India - Pune Tel: 91-20-4121-0141</p> <p>Japan - Osaka Tel: 81-6-6152-7160</p> <p>Japan - Tokyo Tel: 81-3-6880-3770</p> <p>Korea - Daegu Tel: 82-53-744-4301</p> <p>Korea - Seoul Tel: 82-2-554-7200</p> <p>Malaysia - Kuala Lumpur Tel: 60-3-7651-7906</p> <p>Malaysia - Penang Tel: 60-4-227-8870</p> <p>Philippines - Manila Tel: 63-2-634-9065</p> <p>Singapore Tel: 65-6334-8870</p> <p>Taiwan - Hsin Chu Tel: 886-3-577-8366</p> <p>Taiwan - Kaohsiung Tel: 886-7-213-7830</p> <p>Taiwan - Taipei Tel: 886-2-2508-8600</p> <p>Thailand - Bangkok Tel: 66-2-694-1351</p> <p>Vietnam - Ho Chi Minh Tel: 84-28-5448-2100</p>	<p>Austria - Wels Tel: 43-7242-2244-39 Fax: 43-7242-2244-393</p> <p>Denmark - Copenhagen Tel: 45-4485-5910 Fax: 45-4485-2829</p> <p>Finland - Espoo Tel: 358-9-4520-820</p> <p>France - Paris Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79</p> <p>Germany - Garching Tel: 49-8931-9700</p> <p>Germany - Haan Tel: 49-2129-3766400</p> <p>Germany - Heilbronn Tel: 49-7131-72400</p> <p>Germany - Karlsruhe Tel: 49-721-625370</p> <p>Germany - Munich Tel: 49-89-627-144-0 Fax: 49-89-627-144-44</p> <p>Germany - Rosenheim Tel: 49-8031-354-560</p> <p>Israel - Ra'anana Tel: 972-9-744-7705</p> <p>Italy - Milan Tel: 39-0331-742611 Fax: 39-0331-466781</p> <p>Italy - Padova Tel: 39-049-7625286</p> <p>Netherlands - Drunen Tel: 31-416-690399 Fax: 31-416-690340</p> <p>Norway - Trondheim Tel: 47-72884388</p> <p>Poland - Warsaw Tel: 48-22-3325737</p> <p>Romania - Bucharest Tel: 40-21-407-87-50</p> <p>Spain - Madrid Tel: 34-91-708-08-90 Fax: 34-91-708-08-91</p> <p>Sweden - Gothenberg Tel: 46-31-704-60-40</p> <p>Sweden - Stockholm Tel: 46-8-5090-4654</p> <p>UK - Wokingham Tel: 44-118-921-5800 Fax: 44-118-921-5820</p>