You may not use or facilitate the use of this document in connection with any infringement or other legal analysis concerning Intel products described herein. You agree to grant Intel a non-exclusive, royalty-free license to any patent claim thereafter drafted which includes subject matter disclosed herein.

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest Intel product specifications and roadmaps.

The products described may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Copies of documents which have an order number and are referenced in this document may be obtained by calling 1-800-548-4725 or by visiting: http://www.intel.com/design/literature.htm

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Learn more at http://www.intel.com/ or from the OEM or retailer.

No computer system can be absolutely secure.

Intel, Atom and the Intel logo are trademarks of Intel Corporation in the U.S. and/or other countries.

*Other names and brands may be claimed as the property of others.

Copyright © 2016, Intel Corporation. All rights reserved.
Contents

1.0 Introduction .................................................................................................................. 6
  1.1 Terminology .................................................................................................................. 6
  1.2 Reference Documents .................................................................................................... 6

2.0 Before You Begin ........................................................................................................... 7

3.0 6LoWPAN IoT Software Overview ................................................................................. 8

4.0 Hardware and Software Compatibility ............................................................................ 10
  4.1 Software Packages ......................................................................................................... 10
  4.2 Hardware Supported ....................................................................................................... 10

5.0 Building the Firmware for CC2538 .............................................................................. 12

6.0 Build Yocto* Image with NBR Software Stack ............................................................... 14
  6.1 Build image with NBR Software Stack ......................................................................... 14
  6.2 Build Image with LWM2M on Top of NBR Software Stack ........................................ 14
  6.3 Build Procedure ............................................................................................................ 15
  6.4 Prepare USB or SD Card as Bootable Media ................................................................. 16

7.0 Running the 6LoWPAN IoT Software ............................................................................ 17
  7.1 Preparation .................................................................................................................... 17
  7.2 Starting the Network Border Router (NBR) ................................................................. 17
  7.3 Starting the Router Advertisement Daemon (radvd) .................................................... 18
  7.4 Detecting the Nodes Registered to Serial Radio ......................................................... 20

8.0 Accessing 6LoWPAN Network ..................................................................................... 22
  8.1 Access 6LoWPAN Network with CoAP Client .......................................................... 22
    8.1.1 Firefox Web Browser GUI CoAP Client .................................................................... 22
    8.1.2 Command-line Interface CoAP Client ..................................................................... 23
  8.2 Lightweight Machine to Machine (LWM2M) ............................................................... 24
    8.2.1 Configuring and Running the LWM2M Bootstrap Server ...................................... 24
    8.2.2 Running and Using LWM2M (Wakaama) Server to Access the LWM2M Clients ........................................................................................................ 26
     8.2.3 LWM2M Network with Datagram Transport Layer Security (DTLS) Encryption ........................................................................................................ 28
  8.3 IoT WebDemo Tool ....................................................................................................... 29
    8.3.1 Start the 6LoWPAN IoT Webdemo Tool ............................................................... 30
    8.3.2 Pinging a 6LoWPAN Node .................................................................................... 32
    8.3.3 Toggle LEDs ............................................................................................................ 32
    8.3.4 Reading Temperature ............................................................................................ 32
    8.3.5 HTTP Server ........................................................................................................... 33
  8.4 Network Topology View ............................................................................................... 33
  8.5 Configuration Page ....................................................................................................... 34
8.6 RSSI Scanner .................................................................................................................. 36
8.7 802.15.4 Sniffer ............................................................................................................... 36

9.0 Appendix ....................................................................................................................... 38
9.1 SPI Connection ............................................................................................................... 38

Figures

Figure 1. Overview of the Software Architecture on Target System ................................. 8
Figure 2. CoAP GUI Interface ............................................................................................. 23
Figure 3. LWM2M Protocol Stack ....................................................................................... 29
Figure 4. Prompt Message for Authentication .................................................................... 31
Figure 5. Device List for all 6LoWPAN Nodes ................................................................. 31
Figure 6. IPv6 Address of the Nodes on 6LoWPAN Network ............................................ 33
Figure 7. Network Topology View ....................................................................................... 34
Figure 8. Configuration Page .............................................................................................. 35
Figure 9. RSSI Scanner ....................................................................................................... 36
Figure 10. SPI Connection Block Diagram ......................................................................... 38
Figure 11. UART Connection Block Diagram ..................................................................... 39

Tables

Table 1. Terminology ............................................................................................................ 6
Table 2. Reference Documents ......................................................................................... 6
Table 3. Command Line Interface ....................................................................................... 23
Table 4. Objects Supported by IoT-U10 .......................................................................... 27
Table 5. Objects/Resources ............................................................................................... 27
## Revision History

<table>
<thead>
<tr>
<th>Date</th>
<th>Revision</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>August 2016</td>
<td>1.0</td>
<td>Initial release (Gold)</td>
</tr>
</tbody>
</table>
1.0 Introduction

This document, the IPSO 6LoWPAN IoT Build and Software User Guide, contains guides to create and integrate 6LoWPAN IoT features into Yocto BSP and other Linux Platforms.

The intended audience for this document are hardware/software engineers with experience in developing embedded applications.

1.1 Terminology

Table 1. Terminology

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LWM2M</td>
<td>Light Weight Machine to Machine</td>
</tr>
<tr>
<td>6LoWPAN</td>
<td>IPv6 Low power Wireless Personal Area Network</td>
</tr>
<tr>
<td>CoAP</td>
<td>Constrained Application Protocol</td>
</tr>
<tr>
<td>GUI</td>
<td>Graphical User Interface</td>
</tr>
<tr>
<td>SPI</td>
<td>Serial Peripheral Interface</td>
</tr>
<tr>
<td>UART</td>
<td>Universal Asynchronous Receiver/Transmitter</td>
</tr>
<tr>
<td>USB</td>
<td>Universal Serial Bus</td>
</tr>
</tbody>
</table>

1.2 Reference Documents

Table 2. Reference Documents

<table>
<thead>
<tr>
<th>Document</th>
<th>Document No./Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPSO 6LoWPAN IoT Software for Yocto Project* for Intel® Atom™ Processor E3800 Product Family Release Notes</td>
<td>334857</td>
</tr>
<tr>
<td>IPSO 6LoWPAN IoT Software for Yocto Project* for Intel® Atom™ Processor E3800 Product Family User Guide (this document)</td>
<td>334856</td>
</tr>
</tbody>
</table>

§
2.0 Before You Begin

- Prepare a Host PC system with the following requirements:
  - 64-bit Multicore System
  - Running Linux* Ubuntu* 14.04 LTS
  - Minimum of 4GB RAM
  - A high speed internet connection to download third party sources
  - Minimum of 100GB Storage

- If your PC is behind a corporate network with proxy settings, use the guide at [https://wiki.yoctoproject.org/wiki/Working_Behind_a_Network_Proxy](https://wiki.yoctoproject.org/wiki/Working_Behind_a_Network_Proxy) to configure your proxy settings.

- Install additional Linux* packages into your host PC:
  
  ```
  # sudo apt-get install gawk wget git-core diffstat unzip texinfo gcc-multilib build-essential chrpath socat
  # sudo apt-get install libSDL1.2-dev xterm
  # sudo apt-get install make xsltproc docbook-utils fop dblatex xmlto
  # sudo apt-get install autoconf automake libtool libglib2.0-dev
  ```

- The cmake that comes with Ubuntu 14.04 LTS has a problem building the LWM2M DTLS server. It is recommended that you replace it with the latest version of cmake:
  
  ```
  # sudo apt-get remove cmake
  # wget https://cmake.org/files/v3.5/cmake-3.5.1.tar.gz
  # tar -xvf cmake-3.5.1.tar.gz
  # cd cmake-3.5.1
  # ./bootstrap
  # make
  # sudo make install
  ```
3.0 6LoWPAN IoT Software Overview

Below are the main software components and tools for 6LoWPAN support:

- Network Border Router (NBR)
- Serial Radio (Yanzi* IoT-U10 set to AP mode)
- 6LoWPAN nodes (Yanzi* IoT-U10 set to node mode, or other devices)

Figure 1. Overview of the Software Architecture on Target System

- Radvd and the CoAP-client tool are not part of the Native Border Router. It is recommended that they be used together with Native Border Router.
The NBR and the serial radio acts as the root node and forms the 6LoWPAN network with all the 6LoWPAN nodes.

Serial radio must be attached to the Baytrail system's USB port for the NBR to work. To experiment with the 6LoWPAN network, at least one 6LoWPAN node is needed.

The Baytrail system running the Yocto* Project* based Linux image becomes the gateway between all the IP (Internet Protocol) network connections (WiFi, Ethernet, etc). It allows other software such as the web browser and the CoAP client tool (coap-client) to access the 6LoWPAN nodes through serial radio.

Users can also run the Linux IPv6 Router Advertisement Daemon (radvd) on the target system to broadcast the IPv6 prefix, which allows the local client to access the 6LoWPAN nodes by referring to their IPv6 address through CoAP protocol. However, radvd is not needed if users wish to access the 6LoWPAN nodes with the IPv4 address of the HTTP server, which is running on the target system. Details on how the radvd works is out-of-scope in this document. See http://www.litech.org/radvd/ or https://www.ietf.org/rfc/rfc2461.txt for more details.

Users can access the 6LoWPAN nodes through the CoAP protocol by using the web browser on their local client. This procedure only works if the users' local client operating system supports IPv6 and the radvd is running on the target system. Firefox web browser can be used to access the 6LoWPAN nodes by installing Copper(Cu), a CoAP through the GUI. For advanced users, they can also access the 6LoWPAN nodes by using coap-client, which is a command-line interface tool available in Linux. It is the users' responsibility to install the coap-client on their own machine (local client). Users can also use the pre-installed coap-client on the target system to access the 6LoWPAN nodes in the Linux shell.
4.0 Hardware and Software Compatibility

4.1 Software Packages

This release is comprised of:

1. 6LoWPAN_NBR_Linux_NC1.4_1.0.0.tar.gz
   - This is the software package to build image for Intel® Atom™ Processor E3800 Product Family (formerly Baytrail).

2. 6LoWPAN_Firmware-SR-Node_NC1.4_1.0.0.tar.gz
   - This is the firmware for Texas Instruments CC2538EM.

3. 6LoWPAN_IoT-Web-Demo_Linux_NC1.4_1.0.0.tar.gz
   - This is the sample application to access 6LoWPAN node.

4. 6LoWPAN_Wakaama-LWM2M_NC1.4_1.0.0.tar.gz
   - This is the software package to build LWM2M.

4.2 Hardware Supported

1. Intel® Atom™ Processor E3800 Product Family (Formerly Baytrail)
2. Yanzi* IoT-U10
   - Device with temperature sensor and LED that can act as Serial Radio or Node.
   - Contact your Intel representative for more details.
   - Only a USB connection is supported for this hardware.

3. Texas Instruments CC2538EM
   - Texas Instruments reference evaluation module that can act as Serial Radio or Node.
   - Off-the-shelf part from Texas Instruments.
   - Hardware supports USB connections.
   - Refer to Texas Instruments official website for details on the hardware.

4. Texas Instruments CC2538DK (SmartRF06 + CC2538 Evaluation Module)
   - Texas Instruments Development Kits
   - Off-the-shelf parts from Texas Instruments
   - Hardware supports SPI and UART connections
   - See the appendix for SPI and UART connection block diagrams.
• Refer to Texas Instruments official website for details on the hardware.

§
5.0 Building the Firmware for CC2538

The CC2538EM hardware is available from Texas Instruments. CC2538EM is not equipped with an LED and Temperature sensor as in IoT-U10 devices.

This section describes how to build the firmware for Yanzi* IoT-U10 and CC2538 devices.

The user will need to install the gcc ARM toolchain in the Linux build machine before proceeding with the firmware build process. Please go to <address> and register yourself before downloading the gcc ARM toolchain.

The 6LoWPAN Firmware build package is 6LoWPAN_Firmware-SR-Node_NC1.4_1.0.0. Setup the toolchains before proceeding with the firmware build setup process.

1. Copy the setup file in any directory and extract 6LoWPAN_Firmware-SR-Node_NC1.4_1.0.0.tar.gz (preferably in the /tmp directory).
   ```
   # tar zxf 6LoWPAN_Firmware-SR-Node_NC1.4_1.0.0.tar.gz
   # cp -r firmware-setup-tool /tmp/firmware-setup-tool
   # cd firmware-setup-tool
   ```
2. Ensure the firmware-setup.sh script has the correct permissions.
3. Execute the firmware-setup.sh script.
   ```
   # . firmware-setup.sh
   ```
4. This setup script will clone the Contiki source from upstream and apply the necessary patches with which to prepare the build environment for firmware generation.
5. Refer to toolchain_installation.txt attached with the software package to install the necessary toolchain.
6. Navigate to repo/netcontiki-v1.4-SR_Node/6LoWPAN-IoT/products/iot-u10-dual-mode/
   ```
   # cd repo/netcontiki-v1.4-SR_Node/6LoWPAN-IoT/products/iot-u10-dual-mode/
   ```
7. Build the firmware.

   For CC2538EM
   ```
   # make clean TARGET=felicia TARGET_VARIANT=felicia WITH_IPSO=1 WITH_NETSCAN=1 WITH_DTLS=1 WITH_LLSEC_LEVEL=0 images INC=0
   ```
   OR

   For IoT-U10

   ```
   # make clean TARGET=iotu10 TARGET_VARIANT=iotu10 WITH_IPSO=1 WITH_NETSCAN=1 WITH_DTLS=1 WITH_LLSEC_LEVEL=0 images INC=0
   ```
# make clean TARGET=felicia TARGET_VARIANT=iot-u10 WITH_IPSO=1
# make clean TARGET=felicia TARGET_VARIANT=iot-u10 WITH_NETSCAN=1 WITH_DTLS=1 WITH_LLSEC_LEVEL=0 images INC=0

For serial radio running in SPI mode:

# cd 6LoWPAN-IoT/products/iot-u10-serial-radio/
# make clean TARGET=felicia TARGET_VARIANT=felicia WITH_SPI=1 images INC=0

For serial radio running in UART mode:

# cd 6LoWPAN-IoT/products/iot-u10-serial-radio/
# make clean TARGET=felicia TARGET_VARIANT=felicia WITH_UART=1 WITH_RTSCTS=1 images INC=0

8. Create the rescue image (execute the following command from iot-u10-dual-mode folder itself):

# cd ../../../cpu/cc2538/bootloader/
# make clean all IMAGE=0
# ../../../tools/yanzi/createRescueImage.sh -B bootloader -m mfg-felicia-felicia.mhx -i ../../../6LoWPAN-IoT/products/iot-u10-dual-mode/felicia.1.flash

For SPI and UART

# cd ../../../cpu/cc2538/bootloader/
# make clean all IMAGE=0
# ../../../tools/yanzi/createRescueImage.sh -B bootloader -m mfg-felicia-felicia.mhx -i ../../../6LoWPAN-IoT/products/iot-u10-serial-radio/felicia.1.flash

rescueimage.bin is created.

9. Copy rescueimage.bin to your Windows machine and flash it to your CC2538EM board.

Refer to the Texas Instruments website for detailed guidelines to flash the CC2538EM node using the flash upgrade tool.

§
6.0 **Build Yocto* Image with NBR Software Stack**

For the user who is planning to use Native Border Router on a Baytrail system, there is a shell script provided to help build the complete Linux* file system and kernel image that contain 6LoWPAN IoT Software components. Running this shell script will download all the Linux* components through an internet connection and set up the build environment for subsequent build instructions.

6.1 **Build image with NBR Software Stack**

1. Unzip the tar ball to get the nbr-byt-setup folder.
   ```
   # tar -xvf 6LoWPAN_NBR_Linux_NC1.4_1.0.0.tar.gz
   ```

2. Create a directory called 6LoWPAN-development and copy the content of nbr-byt-setup.
   ```
   # mkdir -p ~/6LoWPAN-development
   # cp -r nbr-byt-setup/* ~/6LoWPAN-development
   # cd ~/6LoWPAN-development
   ```

3. Run the shell script.
   ```
   # . byt-netcontiki-setup.sh
   ```

4. The **byt-netcontiki-setup.sh** script takes no parameters.

   During the setup process, the user will be prompted to name the build directory. The default is “build-6LoWPAN”.

   If you choose the default, the build directory will be ./build/build-6LoWPAN.

   The user will be prompted to choose a 32-bit system or 64-bit system. The default is a 64-bit system.

5. If the setup procedure is somehow interrupted, execute the setup script again.

   Once the setup procedure is complete, the user will be at the build folder./build/build-6LoWPAN. If you would like integrate LWM2M into the image, proceed to the next section. Otherwise, go to the build procedure.

6.2 **Build Image with LWM2M on Top of NBR Software Stack**

1. un-tar the package to get the wakaama-byt-setup folder.
   ```
   # tar -xvf 6LoWPAN_Wakaama-LWM2M_NC1.4_1.0.0.tar.gz
   ```

2. Copy the contents of wakaama-byt-setup into ~/6LoWPAN-development that was created in a previous section.
# cp -r wakaama-byt-setup/* ~/6LoWPAN-development
# . wakaama-setup.sh

3. The `wakaama-setup.sh` script takes no parameters.

   During the setup process, the user will be prompted to name the build directory.
   The default is “build-6LoWPAN”.

   If choose the default, the build directory will be `/build/build-6LoWPAN`.

   The user will be prompted to choose a 32-bit system or 64-bit system. The default
   is a 64-bit system.

4. If the setup procedure is somehow interrupted, execute the setup script again.

   Once the setup procedure is complete, the user will be at the build
   folder ./build/build-6LoWPAN. The user may now proceed with the build procedure
   in the next section.

### 6.3 Build Procedure

1. A complete build may take several hours to complete, depending on the internet
   connection speed and your build PC’s specifications.

   This command should be executed under the build folder ./build/build-6LoWPAN.
   
   ```
   # bitbake -c cleanall core-image-sato
   # bitbake core-image-sato
   ```

2. If the build procedure is somehow interrupted, execute the bitbake command
   again.

   Depending on whether a 32-bit or 64-bit system is selected, the image is built to
   `tmp/deploy/images/valleyisland-32/` or `tmp/deploy/images/valleyisland-64/`,
   respectively.

   The output image file is `core-image-sato-*hddimg`.

   If the current terminal is closed or the user wishes to execute a concurrent build,
   executing the setup script again is not required. Instead, execute:

   ```
   # cd ~/6LoWPAN-development
   # source repo/poky/oe-init-build-env build/build-6LoWPAN/
   # bitbake -c cleanall core-image-sato
   # bitbake core-image-sato
   ```
6.4 Prepare USB or SD Card as Bootable Media

To write the bootable image to an SD card or USB stick (assuming it is at /dev/sdc):

```
# cd tmp/deploy/images/valleyisland-64
# sudo dd if=core-image-sato-*hddimg of=/dev/sdc
# sync
# eject /dev/sdc
```
7.0 Running the 6LoWPAN IoT Software

This section guides you step-by-step on running the 6LoWPAN software on the target system platform. We assume users already have access to the target system Linux shell through an SSH (Ethernet port) or serial console (UART port).

7.1 Preparation

This is not an introductory document for 6LoWPAN. Many online resources cover the concepts and specifications of 6LoWPAN. This document walks you step-by-step in setting up the gateway or network border router (NBR) to enable communication between the local/remote PC on the Ethernet network and the nodes on the 6LoWPAN network. It also covers different methods to access the 6LoWPAN nodes with a local PC through HTTP and CoAP protocols.

This document assumes you have some experience with IPv4 and IPv6 networking, Linux*/Windows* OSes, and using software/hardware tools to flash firmware into the hardware platform.

What you need:

- A host PC as local/remote client running: Ubuntu* Linux with IPv6 support is recommended.
- An Intel Architecture target PC
- A web browser installed on the host PC. A Firefox* browser is recommended.
- An Ethernet network with a DHCP server. This network connects the local client and target system through the LAN. Connection to the internet and remote client are optional.
- A 6LoWPAN serial radio device.
- One or more 6LoWPAN node devices.

7.2 Starting the Network Border Router (NBR)

1. Attach the serial radio to the USB port on the target system. You should be able to see ttyACM0 in the /dev directory. Enter the following command to confirm the serial radio is detected by the OS:

```bash
# ls /dev/ttyA*
/dev/ttyACM0
```

2. If the serial radio is present, then you can start the NBR. The NBR executable binary is located at /opt/netcontiki/sbin. Enter the command as follows:

```bash
# cd /opt/netcontiki/sbin/
```
a. For USB, enter the command:

```bash
# ./border-router.native -s /dev/ttyACM0 aaaa::1/64
```

b. For SPI, provide the GPIO number that is being connected to the serial radio for SPI interrupt purposes, e.g. GPIO82:

```bash
# echo 82 > /sys/class/gpio/export
# ./border-router.native -g82 -s /dev/spidev0.0 aaaa::1/64
```

c. For UART, with baud rate 921.6kbps:

```bash
# ./border-router.native -B921600 -H -s /dev/ttyS1 aaaa::1/64
```

3. A tunnel interface tun0 with IPv6 address aaaa::1 is created by NBR as shown below:

```bash
# ifconfig
...
tun0 Link encap:UNSPEC  HWaddr 00-00-00-00-00-00  inet6 addr: aaaa::1/64 Scope:Global UP POINTOPOINT RUNNING NOARP MULTICAST MTU:1500  Metric:1 RX packets:0 errors:0 dropped:0 overruns:0 frame:0 TX packets:0 errors:0 dropped:0 overruns:0 carrier:0 collisions:0 txqueuelen:500 RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

**Note:** The NBR needs to run continuously. Users are advised to run other applications in new terminals to avoid disrupting the NBR. Please kill the NBR (# Ctl + c) before dismantling the device connection.

4. The NBR is up and running. Next, start the radvd.

### 7.3 Starting the Router Advertisement Daemon (radvd)

For users who wish to access the 6LoWPAN network through a local client with IPv4 address only, this step is not required. The Router Advertisement Daemon (radvd) is required if users need to access the 6LoWPAN nodes with IPv6 addresses through the local client. Before the radvd can be executed, users are required to set IPv6 as a static address.

1. Edit the `/etc/network/interfaces` configuration by entering

```bash
vi /etc/network/interfaces
pre-up modprobe ipv6
iface eth0 inet6 static
address bbbb::6a05:caff:fe1c:12d3
netmask 64
```
2. Make sure the radvd configuration file located at /etc/radvd.conf exists. If /etc/radvd.conf does not exist, create it by issuing the command “vi /etc/radvd.conf” and enter the following recommended configuration settings and save it:

```plaintext
interface eth0
{
    AdvSendAdvert on;
    prefix bbbb::/64
    {
        AdvOnLink on;
        AdvAutonomous on;
    }
    route aaaa::/64
    {
    }
};
```

With the settings above, radvd sends advertisements to all clients on the same LAN. When the clients receive the router advertisement message, they automatically configure addresses with the received prefix (bbbb::/64) and the default route. This allows the clients on the LAN network to access the nodes on the 6LoWPAN network. For more details on radvd advanced configurations, visit http://linux.die.net/man/5/radvd.conf and http://www.tldp.org/HOWTO/Linux+IPv6-HOWTO/hints-daemons-radvd.html.

3. Enter the following command to start the radvd:

```
# radvd
```

4. If radvd is running successfully, you will see the Ethernet interface eth0 is auto configured with IPv6 address with prefix bbb::/64. Note that the interface name could be different.

```
# ifconfig
Eth0 Link encap:Ethernet  HWaddr 98:4F:EE:01:6E:30
    inet addr:192.168.0.100  Bcast:172.30.66.255
    Mask:255.255.255.0
    inet6 addr: bbb::984f:eeff:fe01:6e30/64 Scope:Global
    inet6 addr: fe80::984f:eeff:fe01:6e30/64 Scope:Link
    UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
    RX packets:37083 errors:0 dropped:0 overruns:0 frame:0
    TX packets:3907 errors:0 dropped:0 overruns:0 carrier:0
    collisions:0 txqueuelen:1000
    RX bytes:3118774 (2.9 MiB) TX bytes:554533 (541.5 KiB)
    Interrupt:41 Base address:0x8000
```
The Ethernet interface on the target system as shown above is configured with IPv6 address bbbb::984feeff:fe01:6e30/64.

All the local clients located in the same LAN are auto configured with prefix bbbb::/64. For example, in Figure 1, the local client is configured with IPv6 address bbbb::20c:30ff:fe0c:1b23/64.

5. To enable routing in IPv6, issue the following command:

   `# sysctl -w net.ipv6.conf.all.forwarding=1`

   Now the target system acts as an IPv6 router, which enables the local client to access the 6LoWPAN nodes with IPv6 addresses.

7.4 Detecting the Nodes Registered to Serial Radio

There are several ways to identify the nodes that are joining the NBR. These methods require the http web address obtained when the NBR was started.

Example:

IPv6 Address of NBR is aaaa::212::4b00::60e:943

Method 1:

When the NBR is started, nodes that are powered up will begin to join the network one by one. This information is visible at the terminal where the NBR was started.

Method 2:

When the NBR is started, the NBR will establish an http link. The user will be able to access the nodes via browser. The webpage displays the nodes that are connected to the network border router.
Method 3:

When the NBR is started, the NBR will establish an http link. The user will need to wget the ipv6 address using the following command:

```
# wget -qO- http://[NBR_IPV6_ADDR]/r
```

![Image of network status](image-url)
8.0 **Accessing 6LoWPAN Network**

This document covers two methods to access the 6LoWPAN Network:

- CoAP protocol between CoAP server and CoAP client
- IoT WebDemo Tool

8.1 **Access 6LoWPAN Network with CoAP Client**

In this section, we focus on accessing the 6LoWPAN nodes directly with IPv6 addresses through CoAP protocol. Keep in mind that every node in 6LowPAN is a CoAP server. The user will access the nodes as a CoAP client either through a Firefox Web Browser GUI or command line.

8.1.1 **Firefox Web Browser GUI CoAP Client**

Using your Firefox web browser visit [https://addons.mozilla.org/en-US/firefox/addon/copper-270430/](https://addons.mozilla.org/en-US/firefox/addon/copper-270430/) to install the CoAP plugin **Copper (Cu)** into it. Make sure the plugin is installed properly and enabled in the Firefox web browser. Then, find out the IPv6 address of the 6LoWPAN node you wish to access. For example, if the 6LoWPAN node's IPv6 address is `aaaa::212:4b00:60e:db5`, enter the following CoAP URI into the Firefox address bar:

```
coop://[aaaa::212:4b00:60e:db5]/.well-known/core
```

A CoAP GUI interface is shown in the screenshot below.
Use the buttons **GET**, **POST**, **PUT**, **DELETE** to perform an action on the resources listed in the left pane on the web page. For example, at a resource location (coap://[aaaa::212:4b00:60e:db5]:5683/temperature), use the **GET** button to read the temperature. The temperature reading is displayed in the browser. To toggle the LED on the 6LoWPAN node, click on the link “0” under the “led” in the left pane. You are redirected to URI coap://[aaaa::212:4b00:60e:db5]:5683/led/0. Click on the **POST** button to toggle the state of LED. The state of LED is displayed in the browser.

### 8.1.2 Command-line Interface CoAP Client

For users who prefer a command-line interface, they can use the **coap-client** preinstalled in the target system. Below are some examples for using the command-line interface to access the 6LoWPAN node:

<table>
<thead>
<tr>
<th>CoAP Resource Name</th>
<th>Method</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>./well-known/core</td>
<td>GET</td>
<td>root@valleyisland-64:~# coap-client -m get coap://[aaaa::212:4b00:60e:db5]/./well-known/core v:1 t:0 tkl:0 c:1 id:49657</td>
</tr>
<tr>
<td>CoAP Resource Name</td>
<td>Method</td>
<td>Command</td>
</tr>
<tr>
<td>--------------------</td>
<td>--------</td>
<td>---------</td>
</tr>
<tr>
<td>&lt;/well-known/core&gt;;ct=40,;/led&gt;;title=&quot;LED(default): led=toggle ?len=0..&quot;;rt=&quot;Text&quot;,;/led/0&gt;;title=&quot;LED0: led=toggle ?len=0..&quot;;rt=&quot;Text&quot;,;/led/1&gt;;title=&quot;LED1: led=toggle ?len=0..&quot;;rt=&quot;Text&quot;,;/ipv6/neighbors&gt;;title=&quot;IPV6 Neighbors: ?len=0..&quot;;rt=&quot;Text&quot;,;/rpv:1 t:0 tkl:0 c:1 id:49658 l-info&gt;;title=&quot;RPL Information&quot;;obs,;/rpv:parent&gt;;title=&quot;RPL Parent&quot;;obs,;/rpv:rank&gt;;title=&quot;RPL Rank&quot;;obs,;/rpv:link-metric&gt;;title=&quot;RPL Link Metric&quot;;obs,;/push-button&gt;;title=&quot;Push event&quot;;obs,;/temperature&gt;;title=&quot;Temperature: ?len=0..&quot;;rt=&quot;Tv:1 t:0 tkl:0 c:1 id:49659 ext&quot;</td>
<td>GET</td>
<td>root@valleyisland-64:~# coap-client -m get coap://[aaaa::212:4b00:60e:db5]/temperature v:1 t:0 tkl:0 c:1 id:19162 Temperature (C): 28.5</td>
</tr>
<tr>
<td>/temperature (only applicable for IoT-U10)</td>
<td>GET</td>
<td>root@valleyisland-64:~# coap-client -m get coap://[aaaa::212:4b00:60e:db5]/temperature v:1 t:0 tkl:0 c:1 id:19162 Temperature (C): 28.5</td>
</tr>
<tr>
<td>/led/0 (only applicable for IoT-U10)</td>
<td>POST</td>
<td>root@valleyisland-64:~# coap-client -m post -e led=toggle coap://[aaaa::212:4b00:60e:db5]/led/0 v:1 t:0 tkl:0 c:2 id:46220 LED0 Toggle: ON</td>
</tr>
</tbody>
</table>

### 8.2 Lightweight Machine to Machine (LWM2M)

Lightweight Machine to Machine (LWM2M) is a protocol from the Open Mobile Alliance for M2M or IoT device management. It is a new, still on-going effort to create a new technical standard for remote management of machine-to-machine devices, service enablement, and application management. For more detailed information about how this protocol works, obtain specification documents from the URL below:

http://openmobilealliance.hs-sites.com/lightweight-m2m-specification-from-oma

To manage 6LoWPAN nodes using this protocol, the nodes should support the LWM2M protocol. The nodes act as LWM2M clients and communicate with the LWM2M server running on the target system. Commands and queries can be sent to the nodes through the LWM2M server's command prompt interface. The following sections describe the required steps to configure the LWM2M clients and manage them through the LWM2M server.

### 8.2.1 Configuring and Running the LWM2M Bootstrap Server

The LWM2M (Wakaama) Bootstrap Server provides a Bootstrap Interface for provisioning the essential information into the LWM2M clients to enable the LWM2M clients to register themselves with one or more LWM2M servers.
The LWM2M Bootstrap Server's executable binary (bootstrap_server) and its configuration file (bootstrap_server.ini) can be found under directory /opt/netcontiki/sbin/.

To start the LWM2M (Wakaama) Bootstrap Server, perform the following:

```
# cd /opt/netcontiki/sbin
# ./bootstrap_server -f bootstrap_server.ini
```

Messages are displayed in the console when provisioning happens between the LWM2M clients and the LWM2M Bootstrap server. The LWM2M clients are ready to connect to LWM2M server after successfully configured by the LWM2M Bootstrap Server.

Here is an example of bootstrap_server.ini:

```
# Standard non-encrypted LWM2M server
[Server]
id=1
uri=coap://[aaaa::1]:5683
bootstrap=no
lifetime=300
security=NoSec

# DTLS encrypted LWM2M server with Pre-Shared key
# using identity 'OurIdentity' and password 'OurSecret'.
# The identity and password are specified in hexadecimal form.
[Server]
id=2
uri=coaps://[aaaa::1]:5684
bootstrap=no
lifetime=300
security=PSK
public=4f75724964656e74697479
secret=4f7572536563726574

# For any client, we delete all server accounts and
# provision all the server info.
[Endpoint]
Delete=/0
Delete=/1
Server=1
```

The settings above provision the LWM2M clients to connect to the first entry of the LWM2M server listed in the configuration file, which is a non-DTLS encrypted LWM2M server with URI (coap://[aaaa::1]:5683). This is the ipv6 address of the target system.
8.2.2 Running and Using LWM2M (Wakaama) Server to Access the LWM2M Clients

Start the LWM2M (Wakaama) Server in another terminal:

```
# cd /opt/netcontiki/sbin
# ./lwm2mserver 2> messages.log
```

This starts the LWM2M (Wakaama) Server with all the low level messages received from the LWM2M clients being redirected to the file `messages.log`.

Information as shown below are displayed on the screen automatically when a LWM2M client has successfully registered itself to the LWM2M server:

New client #0 registered.
Client #0:
  name: "IoT-U104B00060E0DB5"
  binding: "Not specified"
  lifetime: 86400 sec
  objects: /0/3, /1/0, /3/0, /3200/0, /3303/0, /3311/0, /3311/1,

Depending on the type of services or functions supported by the LWM2M clients, different object information may be shown.

Enter the "help" command in the LWM2M server's command prompt to get a list of commands supported by the server:

```
> help
help   Type 'help [COMMAND]' for more details on a command.
list   List registered clients.
read   Read from a client.
disc   Discover resources of a client.
write  Write to a client.
time   Write time-related attributes to a client.
attr   Write value-related attributes to a client.
clear  Clear attributes of a client.
exec   Execute a client resource.
del    Delete a client Object instance.
create create an Object instance.
observe Observe from a client.
cancel Cancel an observe.
q      Quit the server.
```

Enter the "list" command to get a list of LWM2M clients registered to the server.

```
> list
Client #0:
  name: "IoT-U10plus4B00060E0DB5"
```
From the example above, there is only one LWM2M client registered to the server and the client’s name is “IoT-U10plus4B00060E0DB5”. The ID of the client is 0. There are seven IPSO objects reported by the LWM2M client.

IPSO objects are presented with {Object ID}/{Object Instance}.

The following objects are supported by IoT-U10.

**Table 4. Objects Supported by IoT-U10**

<table>
<thead>
<tr>
<th>Object ID</th>
<th>Description</th>
<th>Instance ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>/0/1</td>
<td>LWM2M Security Instance 1</td>
<td></td>
</tr>
<tr>
<td>/1/0</td>
<td>LWM2M Server Instance 0</td>
<td></td>
</tr>
<tr>
<td>/3/0</td>
<td>Device Instance 0</td>
<td></td>
</tr>
<tr>
<td>/3200/0</td>
<td>Digital Input Instance 0</td>
<td>e.g., switch/button</td>
</tr>
<tr>
<td>/3303/0</td>
<td>Temperature Sensor Instance 0</td>
<td></td>
</tr>
<tr>
<td>/3311/0</td>
<td>Light Control Instance 0</td>
<td>e.g., LED/light bulb</td>
</tr>
<tr>
<td>/3311/1</td>
<td>Light Control Instance 1</td>
<td>e.g., LED/light bulb</td>
</tr>
</tbody>
</table>

CC2538EM doesn’t support LED and Temperature interface. The first three IPSO objects (/0/1, /1/0, and /3/0) are only supported by CC2538EM. From the information above, the LWM2M client has one digital, one temperature sensor, and two light controls.

An Object is a collection of Resources. A Resource is an atomic piece of information or interface that can be accessed or manipulated (for example, Read, Written or Executed) by a user. Objects/Resources can be accessed with simple URI /{Object ID}/{Object Instance}/{Resource ID} as indicated in the following table.

**Table 5. Objects/Resources**

<table>
<thead>
<tr>
<th>URI</th>
<th>Object ID</th>
<th>Instance ID</th>
<th>Resource ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>/3200/0/5500</td>
<td>Digital Input</td>
<td>Instance 0</td>
<td>Digital Input State</td>
</tr>
<tr>
<td>/3303/0/5700</td>
<td>Temperature Sensor</td>
<td>Instance 0</td>
<td>Sensor Value</td>
</tr>
<tr>
<td>/3311/0/5850</td>
<td>Light Control</td>
<td>Instance 0</td>
<td>On/Off</td>
</tr>
</tbody>
</table>

For more information about the definition of Object and Resource, refer to the following URL: [http://technical.openmobilealliance.org/Technical/technical-information/omna/lightweight-m2m-lwm2m-object-registry](http://technical.openmobilealliance.org/Technical/technical-information/omna/lightweight-m2m-lwm2m-object-registry)

Below are some of the basic examples on how the LWM2M client is accessed through the URIs as shown above.
To read the state of the input button of the LWM2M client (Client ID 0), enter the following command in LWM2M (Wakaama) server's command prompt:

```
> read 0 /3200/0/5500
OK
> Client #0 3200/0/5500 : 2.05 (COAP_205_CONTENT)
4 bytes received:
   E1 15 7C 00 ..
```

To get the reading of the temperature sensor of the LWM2M client:

```
> read 0 /3303/0/5700
OK
> Client #0 3303/0/5700 : 2.05 (COAP_205_CONTENT)
7 bytes received:
   E4 16 44 41 FF 7E 00 ..DA~.
```

To turn on the first LED of the LWM2M client:

```
> write 0 /3311/0/5850 1
OK
> Client #0 3311/0/5850 : 2.04 (COAP_204_CHANGED)
```

To turn off the first LED of the LWM2M client:

```
> write 0 /3311/0/5850 0
OK
> Client #0 3311/0/5850 : 2.04 (COAP_204_CHANGED)
```

For more advance operations on LWM2M object's resources, refer to the LWM2M specification document.

All data received from LWM2M client are in raw format (Hexadecimal). Refer to the 6LoWPAN device (for example, IoT-U10) manual for the interpretation of the data.

### 8.2.3 LWM2M Network with Datagram Transport Layer Security (DTLS) Encryption

Datagram Transport Layer Security (DTLS) is a protocol for securing network transmission for datagram-oriented transport such as UDP. It provides security for data interchanged between the LWM2M server and LWM2M client. Refer to Figure 3 below.
DTLS can be enabled easily by changing the configuration file of LWM2M (Wakaama) Bootstrap Server. Just change the settings under **Endpoint** in `bootstrap_server.ini` as indicated below:

```
[Endpoint]
Delete=/0
Delete=/1
Server=2
```

The settings above provision the LWM2M clients to connect to LWM2M server with id = 2. Refer to Section 8.2.1. The 2nd entry of the server in `bootstrap_server.ini` is configured using the DTLS Pre-Shared Key security.

Start the LWM2M (Wakaama) Bootstrap Server:
```
# cd /opt/netcontiki/sbin
# ./bootstrap_server --f bootstrap_server.ini
```

Then, start the LWM2M (Wakaama) server with DTLS:
```
# cd /opt/netcontiki/sbin
# ./lwm2mserver -dtls > messages.log
```

The LWM2M clients should connect to the LWM2M (Wakaama) server with secure encryptions and be available for access through the LWM2M (Wakaama) server as described in Section 8.2.2.

For more information about enhancing 6LoWPAN network security, please refer to 6LoWPAN IoT Security Guide.

### 8.3 IoT WebDemo Tool

This section describes the use of the 6LoWPAN IoT WebDemo Tool. This is a sample application for the user to be able to access 6LoWPAN nodes.

There are known security risks when using this tool under certain system configurations. The user owns the responsibility of ensuring that system security is not compromised by operating this tool.
Before you begin:

- You need a target PC running 6LoWPAN IoT Software
- The PC is connected in a LAN/WLAN configuration
- Extranet connection is optional

### 8.3.1 Start the 6LoWPAN IoT Webdemo Tool

1. Unzip the tarball into `/opt/netcontiki/sbin`.
   
   ```bash
   # cd /opt/netcontiki/sbin
   # tar -xvf 6LoWPAN_IoT-Web-Demo_Linux_NC1.4_1.0.0.tar.gz
   # cp -r iot-webdemo-tool/iotwebdemo /opt/netcontiki/sbin
   ```

2. Start the 6LoWPAN Demo Server (`wsdemoserver.py`) to allow the local client and remote client to access the 6LoWPAN network through HTTP protocol with the IPv4 address.
   
   ```bash
   # cd /opt/netcontiki/sbin/iotwebdemo/demo
   # python wsdemoserver.py &
   ```

   The above command starts the 6LoWPAN Demo Server as a background process. Now the local client and remote client are able to access the HTTP server through port 8000.

3. After the 6LoWPAN Demo Server has started, you can browse to the IP address of the target system using any modern browser (with support for HTML5 and JavaScript).

4. The default port is 8000, so the link to the index page is (assuming the target system's IP address is 192.168.0.100) the following:

   ```
   http://192.168.0.100:8000/index.html
   ```

   a. When the IoT-Webdemo tool is started, the user will be prompted to insert Username and Password.

   Username: admin
   Password: iotwebdemo
b. The 6LoWPAN Demo page contains a device list where all the 6LoWPAN nodes can be viewed and all the basic functions can be demonstrated by clicking on the buttons. The device list is shown in the screenshot below.

Figure 4. Prompt Message for Authentication

![Authentication Message]

In the screenshot above, you can see there are two 6LoWPAN nodes discovered by the NBR. Users can click on the "Update list" button to refresh the device list if the 6LoWPAN nodes are not listed on the page. Each of the devices is listed with IPv6 address together and their type. CC2538EM Node is identified as Felicia-Node and IoT-U10 node is identified as IoT-U10.

Figure 5. Device List for all 6LoWPAN Nodes

![Device List Screenshot]
CC2538EM does not contain LED and temperature sensors by default. Any features related to temperature and LED feature will be disabled by default. These hardware features are only available in IoT-U10.

### 8.3.2 Pinging a 6LoWPAN Node

Any node that connects to the 6LoWPAN network is listed in the device list of the page. Each of the nodes has a ping button so that it is possible to ping the nodes from the web application.

After clicking on the ping button, a window pops up within the web page and the response from the ping command is shown.

#### 8.3.3 Toggle LEDs

If the node is a Yanzi IoT-U10 radio device, there are two Toggle LED buttons on the same row as the IPv6 address of the node. Clicking on these buttons toggles the corresponding LED on the IoT-U10 node.

### 8.3.4 Reading Temperature

If the node is a Yanzi IoT-U10 radio device, there is a button for reading the temperature of the device. If clicked, the request is sent to the 6LoWPAN web server, which then sends a request for reading the temperature variable. The temperature is displayed at the end of the same row.
8.3.5 HTTP Server

In the IoT-U10 node, there is also a tiny web server that shows some information on the node. If your OS supports IPv6 and you are directly connected to the same network, you can either browse to [http://IPv6-address-of-node] or do a wget to the same address to access the node’s information through HTTP protocol.

The screenshot below shows that we can browse to the IPv6 address of the nodes on the 6LoWPAN network with a web browser. A simple web page with some links is shown in the browser. Click on the links displayed in the web page to retrieve basic network information about the node.

**Figure 6. IPv6 Address of the Nodes on 6LoWPAN Network**

![IPv6 Address of the Nodes on 6LoWPAN Network](image)

8.4 Network Topology View

The screenshot shown below is the network topology view that shows two 6LoWPAN nodes that are connected to the NBR. All these nodes support the network statistics functional instance. This instance makes it possible to read out routing layer data such as Rank (position in the network) and Parent (the default route of the node) and other information. This information is used to create a network topology graph.
8.5 Configuration Page

The configuration page of the 6LoWPAN demo server page shows some information about the network border router (NBR) and serial radio. It also allows configuration of channel, PAN ID, and the link layer security key as shown in the screenshot below.
Figure 8. Configuration Page

Note: Currently, the channel, PAN ID, security level, and encryption key of the nodes in the 6LoWPAN network are set during firmware compilation time. Therefore, changing these configuration parameters on the NBR will risk losing communication between the 6LoWPAN nodes and the NBR.
8.6 RSSI Scanner

The RSSI Scanner demo is started by clicking on “RSSI Scan” on the “NBR and Serial Radio” web page. The NBR then re-configures the serial radio to act as a RSSI Scanner and pops up a window within the web page that shows the RSSI scanning results in a 3D surface plot. The graph is updated as long as the sniffer mode is running. While the scanner is running, the NBR will not manage the 6LoWPAN network.

To exit the scanner mode, click outside the window or on the closing button in the upper right corner.

Figure 9. RSSI Scanner

8.7 802.15.4 Sniffer

The sniffer demo is started by clicking on the “Sniffer” button on the “NBR and Serial Radio” web page. It sniffs a set of packets for a short time and displays the packets in hexadecimal in a pop-up window. While the sniffer is running, the NBR will not manage the 6LoWPAN network.
To exit the sniffer mode, click outside the window or on the closing button in the upper right corner.
9.1 SPI Connection

CC2538DK (acts as serial radio) supports an SPI connection with NBR (Intel® Atom™ Processor E3800m™ Processor E3800). Refer to the schematic for each of the products for more details. Note that there is an additional connection required from the RF1.17 pin header (SmartRF06) to any one of the available GPIO pins in the Intel® Atom™ Processor E3800 for SPI interrupt. The interrupt line is used by serial radio to notify NBR when data is available for NBR.

Figure 10. SPI Connection Block Diagram
9.2 UART Connection

CC2538DK (acts as serial radio) supports a UART connection with NBR (Intel® Atom™ Processor E3800m™ Processor E3800). Refer to the schematic for each of the products for more details.

Figure 11. UART Connection Block Diagram

![UART Connection Block Diagram](image-url)