# Design Guide: TIDA-010024 Secured 6LoWPAN mesh end-node with enhanced network capacity reference design

# **TEXAS INSTRUMENTS**

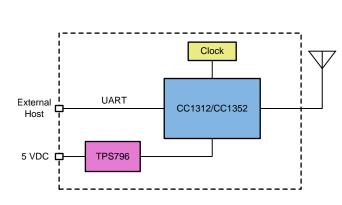
#### Description

This reference design implements a Radio Frequency (RF) mesh network end node with Datagram Transport Layer Security (DTLS) for smart meter Advanced Metering Infrastructure (AMI) networks. The network is an IPv6 over low-power wireless personal area networks (6LoWPAN) solution. The design implements this network in a single CC1312R SimpleLink™ wireless MCU to enhance performance and minimize system cost. The complete 6LoWPAN mesh network runs on top of the TI-15.4 stack based on IEEE 802.15.4e/g protocols and implements an Un-Slotted Channel Hopping (USCH) mode to provide protection against network interference. Network capacity is improved and the security is enhanced compared to the TIDA-010003 end-node design with the inclusion of larger neighbor and routing tables that optimize packet routing decisions.

#### Resources

TIDA-010024	Design Folder
CC1312R	Product Folder
TM4C1294NCPDT	Product Folder
LM4040	Product Folder
TPS796	Product Folder
SN74AVC4T245	Product Folder





ASK Our E2E<sup>™</sup> Experts

# A

An IMPORTANT NOTICE at the end of this TI reference design addresses authorized use, intellectual property matters and other important disclaimers and information.

#### Features

- IPv6 networking over low-power RF in the < 1-GHz ISM bands
- Implements IP mesh network protocols of 6LoWPAN, RPL, IPv6, ICMPv6, UDP, and DTLS
- Implements TI-15.4 stack with IEEE 802.15.4ebased frequency hopping and MAC data encryption
- Software layer architecture identical to Wi-SUN FAN v1.0
- Adds DTLS, larger neighbor and routing tables to the TIDA-010003 Simple 6LoWPAN Mesh End Node Improves Network Performance Reference Design
- Supports low-power mode configuration for batterypowered end nodes

#### Applications

- Wireless communications
- Electricity meter
- Water meter
- Gas meter





2

#### 1 **System Description**

This reference design provides a low-power RF mesh network end node that supports 6LoWPAN mesh protocols. A primary design goal is to add Datagram Transport Layer Security (DTLS) support plus improve network performance with larger neighbor and routing tables compared to the TIDA-010003 endnode reference design. This designs retains the Frequency Hopping (FH) techniques that increase robustness in noisy Radio Frequency (RF) environments.

FH is a technique of transmitting data by switching one of many channels where the channel is selected by a pseudo-random sequence known to both sender and receiver. This technique is known as robust versus interference and excellent in coexistence performance. Section 3.2 shows experimental results to measure and verify network performance in different scenarios. These scenarios are used to reproduce real life use cases and showcase how the system will perform in these special cases. The testings also show the negligible impact of DTLS on the overall performance. In addition, Table 2 summarizes key system performance.

Another segment of this design is the 6LoWPAN mesh stacks, which improves network coverage and supports IPv6-based applications. The increased network coverage reduces the total system cost by reducing the number of data collectors that are typically more expensive than smart meters. The smart meters are static in the Advanced Metering Infrastructure (AMI) networks. The mesh networking addresses the connectivity issue through multi-hop transmissions when data collectors and smart meters are not reachable with each other.

This design is based upon a SimpleLink<sup>™</sup> MCU of the CC13x2 wireless MCU. Figure 1 shows the overall system architecture. The TI-15.4 Medium Access Control (MAC) supports FH mode. The TI SimpleLink PHY supports 200-kbps and 50-kbps frequency-shift keying (FSK) mode, and 5-kbps long range mode. The sub-1 GHz RF on the CC1312R MCU can support three frequency bands: 902 MHz, 863 MHz, and 433 MHz.

This design is an evolution of the TIDA-010003 Simple 6LoWPAN End-Node Improves Network Performance Reference Design. Table 1 summarizes the key differences in terms of system features between the two reference designs.

	TIDA-010024	TIDA-010003
Wireless MCU	CC1312R	CC1310
Max neighbor entries	100	10
Max route entries	<ul><li> 200 for root node</li><li> 100 for end node</li></ul>	10 for end node
Security	DTLS     IEEE 802.15.4 MAC security	IEEE 802.15.4 MAC security
Max number of hop	64	64
Total flash size	352 KB	128 KB
Remaining flash memory	255 KB (debug_poll CCS configuration)	4 KB
Total RAM size	80 KB (+8 KB Cache)	20 KB (+8 KB Cache)
Remaining RAM	41 KB (debug_poll CCS configuration)	5 KB

#### Table 1. System Features of TIDA-010024 and TIDA-010003



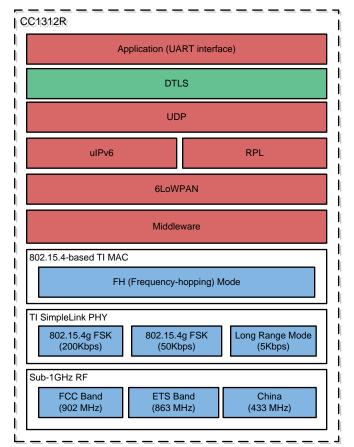


Figure 1. TIDA-010024 System Architecture

3

4

#### Key System Specifications 1.1

PARAMETER	SPECIFICATIONS	DETAILS
Maximum number of hops	<ul><li>64 hops (in software, configurable)</li><li>Tested up to 6-hop networks</li></ul>	
Maximum number of route entries	<ul><li> 200 for root node</li><li> 100 for end node</li><li> Tested up to 100-node networks</li></ul>	
Maximum number of neighbor nodes	<ul><li>100 for end node and route node</li><li>Tested up to 100-node networks</li></ul>	
Maximum application data size	<ul><li> 200 B (in software, configurable)</li><li> Tested up to 200 B</li></ul>	
Delivery ratio	<ul> <li>99.42% (average in 6-hop linear topology with DTLS)</li> <li>97.14% (average with 100 nodes network)</li> </ul>	
Round-trip time (RTT)	<ul><li>0.445 second (100B over 1-hop node with DTLS)</li><li>2.29 second (100B over 6-hop node with DTLS)</li></ul>	
MAC data encryption	<ul> <li>IEEE 802.15.4-based encryption supported</li> <li>MIC-32, MIC-64, MIC-128</li> <li>ENC, ENC-MIC-32, ENC-MIC-64, and ENC-MIC-128</li> </ul>	Section 3.1.2.2.1.3
DTLS encryption	DTLS v1.2     MAC-level DTLS	The MAC-level DTLS was activated and verified in this design.
Memory usage	<ul> <li>End node (debug_poll configuration): <ul> <li>Flash: 255 KB free</li> <li>RAM: 41 KB free</li> </ul> </li> <li>Root node (debug_root_poll configuration): <ul> <li>Flash: 248 KB free</li> <li>RAM: 34 KB free</li> </ul> </li> </ul>	

# Table 2. Key System Specifications



# 2 System Overview

#### 2.1 Block Diagram

Figure 2 shows the system block diagram. The CC1312R (or CC1352R) MCU is the 6LoWPAN mesh MCU that runs UDP applications, 6LoWPAN mesh network, and the TI 15.4-Stack over sub-1 GHz RF.

The external DC/DC converter, as shown in Figure 2, is needed when the external power source supplies the voltage level other than 3.3 V. In this reference design, because the evaluation modules (EVMs) are powered by USB, the TPS796 was chosen to convert 5 V to 3.3 V.

For end-equipment designs, the selection of the power supply depends on the input/output voltage level and required current consumption. The TI WEBENCH Power Designer provides the detailed design of the power supply with the given input requirements.

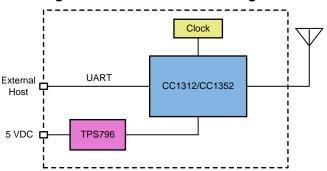


Figure 2. TIDA-010024 Block Diagram

## 2.2 Design Considerations

For this reference design, these devices perform the following:

- The CC1312R wireless MCU combines an Arm<sup>®</sup> Cortex<sup>®</sup>-M3 MCU with a flexible, ultra-low-power RF transceiver with excellent RX sensitivity to provide a robust link budget and execute the TI 15.4-Stack.
- The TPS796 low-power linear regulator offers high power-supply rejection ratio (PSRR), ultra-low noise, fast start-up, and excellent line and load transient responses.

## 2.3 Highlighted Products

## 2.3.1 CC1312R

The CC1312R wireless MCU is a member of the SimpleLink MCU platform. Its very low current consumption in both active and standby mode provides excellent lifetime when operating from batteries or super capacitors.

The CC1312R combines a flexible, very low-power RF transceiver with a powerful 48-MHz Cortex-M3 MCU in a platform supporting multiple physical layers and RF standards. A dedicated radio controller (Cortex-M0) handles low-level RF protocol commands that are stored in ROM or RAM, thus ensuring ultralow power and flexibility. Its RF subsystem offers an excellent link budget with receiver sensitivity with –110 dBm at 50 kbps and output power up to +14 dBm. The CC1312R is a highly integrated, true singlechip solution incorporating a complete RF system and an on-chip DC/DC converter. The CC1312R wireless MCU is supported by the SimpleLink Software Development Kit (SDK) that offers 100% application code compatibility across the entire SimpleLink MCU portfolio and includes the integrated TI-RTOS, complete peripheral driver libraries with POSIX-compatible APIs, and encryption-enabled security features.

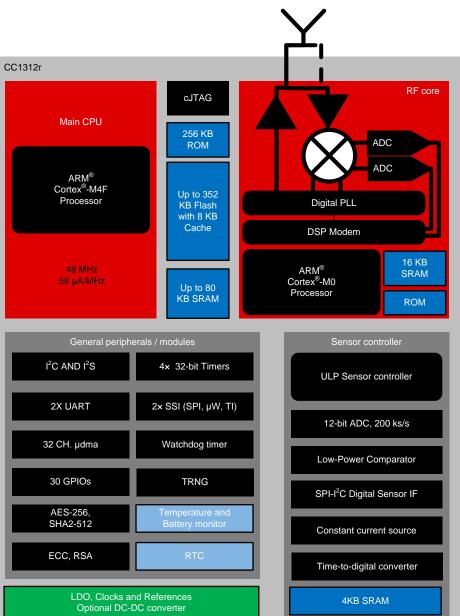
5



System Overview

6

Figure 3 shows the CC1312R functional block diagram.



# Figure 3. CC1312R Functional Block Diagram



#### 2.3.2 **TPS796**

The TPS796 family of low-dropout (LDO), low-power linear voltage regulators feature high power-supply rejection ratio (PSRR), ultra-low noise, fast start-up, and excellent line and load transient responses in small outline, 3x3 VSON, SOT223-6, and TO-263 packages. Each device in the family is stable with a small, 1-µF ceramic capacitor on the output. The family uses an advanced, proprietary BiCMOS fabrication process to yield extremely LDO voltages (for example, 250 mV at 1 A).

Each device achieves fast start-up times (approximately 50 µs with a 0.001-µF bypass capacitor) while consuming very low quiescent current (265 µA, typical). Moreover, when the device is placed in standby mode, the supply current is reduced to less than 1  $\mu$ A. The TPS79630 exhibits approximately 40  $\mu$ V<sub>RMS</sub> of output voltage noise at 3.0-V output with a 0.1-µF bypass capacitor. Applications with analog components that are noise sensitive, such as portable RF electronics, benefit from the high PSRR, low-noise features, and fast response time.

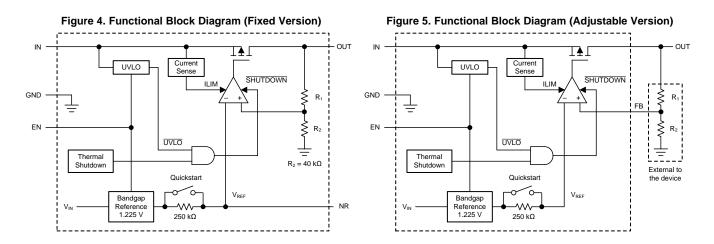


Figure 4 and Figure 5 show the TPS796 functional block diagrams.



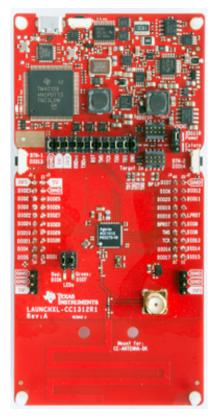
# 3 Hardware, Software, Testing Requirements, and Test Results

# 3.1 Required Hardware and Software

#### 3.1.1 Hardware

This reference design is built with a standard TI EVM of LAUNCHXL-CC1312, as shown in Figure 6.

Figure 6. TIDA-010024 Hardware Platform



8

#### 3.1.2 Software

#### 3.1.2.1 Getting Started

The 6LoWPAN mesh software examples were implemented based on the TI-15.4 sensor example from SimpleLink<sup>™</sup> CC13x2 and CC26x2 SDK v2.40. The software examples provided with the reference design run on the CC1312R MCU to support 6LoWPAN, RPL routing, IPv6, ICMPv6, UDP, DTLS and simple applications. The pre-requisite tools to build the software example are Code Composer Studio<sup>™</sup> v8.0 or above (CCSTUDIO) and SimpleLink<sup>™</sup> CC13x2 and CC26x2 SDK v2.40 (SIMPLELINK-CC13X2-26X2-SDK).

Table 3 shows the summary of software example with CCS build configuration options.

BUILD CONFIGURATION	EXAMPLE	ROLE	DATA ENCRYPTION
debug_poll	UDP Poll	End node	IEEE 802.15.4 MAC encryption
debug_push	UDP Push	End node	IEEE 802.15.4 MAC encryption
debug_poll_dtls	UDP Poll	End node	MAC-level DTLS +IEEE 802.15.4 MAC encryption
debug_poll_demo	UDP Poll	End node (demo with TIDA- 010032)	IEEE 802.15.4 MAC encryption
debug_push_leaf	UDP Push	End node (Low-power mode)	IEEE 802.15.4 MAC encryption
debug_root_poll	UDP Poll	Root node	IEEE 802.15.4 MAC encryption
debug_root_push	UDP Push	Root node	IEEE 802.15.4 MAC encryption
debug_root_poll_dtls	UDP Poll	Root node	MAC-level DTLS + IEEE 802.15.4 MAC encryption

#### Table 3. CCS Build Configurations

For the UDP poll example, end nodes send UDP data only when they receive the poll message from the root node. This example is popular in dense networks because this approach can control network traffic effectively regardless of the network size with the cost of polling overheads. For the UDP push example, end nodes send UDP data whenever they have data to send. Compared to the UDP poll-based approach, this technique reduces the polling overhead while it increases collision probability among transmissions of the end nodes in a dense network.

The UDP poll and push examples improve the level of security with the DTLS data encryption. The debug\_poll\_dtls build configuration provides an example of UDP poll example with the DTLS encryption. They will work the exact same way in terms of mesh stacks as UDP poll and push examples but with additional MAC-level DTLS encryption.

This design also allows the CC13x2 to be used as the root node. The root node in the UDP poll mode requests data from the nodes and sends the statistics of delivery ratio and round trip time (RTT) through a serial terminal. In the UDP push mode, the root node receives the data from the connected nodes in the network and sends the statistics through a serial terminal.

The leaf node configuration (debug\_push\_leaf) was designed to provide a low power operation mode for battery-operated devices. To achieve low power consumption, this mode disables routing capability, turns RX off when idle and runs on top of the TI 15.4-Stack sleep mode. An advantage is to extend end-nodes' coverage with the built-in mesh network while keeping power consumption low. A use case will be battery-powered flow meters or in-home display connected to an electricity meter acting as a router node.

Table 4 summarizes node configurations supported by this design.

#### **Table 4. Node Configurations**

Feature	ROOT	INTERMEDIATE	LEAF
Router Capability	Yes	Yes	No
RX ON when Idle	Yes	Yes	No
TI 15.4-Stack	FH non-sleep mode	FH non-sleep mode	FH sleep mode



Table 4. Node Configurations (continued)
--

Build Configuration	<ul> <li>debug_root_poll</li> <li>debug_root_push</li> <li>debug_root_poll_dtls</li> </ul>	<ul> <li>debug_poll</li> <li>debug_push</li> <li>debug_poll_dtls</li> <li>debug_poll</li> </ul>	debug_push_leaf
---------------------	--	---	-----------------

The TIDA-01547, TIDA-010003, and TIDA-010032 are companion reference designs that work with this design. These designs provide multiple options in terms of system performance or cost to build a complete 6LoWPAN mesh network solution.

Figure 7 and Figure 8 show Code Composer Studio screen captures to show the compiler, XDC tool and SimpleLink SDK versions for the software example.

Properties for 6lowpan_mesh_ti_	15_4_cc1312_tirtos_ccs	leras tatificas;	
type filter text	General		↓ → ▼
Resource     General     Build     XDCtools     ARM Compiler     Processor Options     Optimization     Include Options     ULP Activisor     Predefined Symbols     Advanced Options     Advanced Options     Advanced Options	Configuration: debug_pot Project Products Device Family: ARM Variant <select or<="" td=""><td>II [Active]   II [Active]  II [</td><td>Manage Configurations</td></select>	II [Active]   II [Active]  II [	Manage Configurations
ARM Hex Utility [Disablec	🗹 Manage	the project's target-configuration automatically	
Debug	Tool-chain Compiler version: Output type: Output form at Device endianness: Linker command file:	Trv18.1.1.LTS RTSC Application (Executable) eabi (ELF) little cc13x2lp.cmd	More      More      Browse
	Runtime support library		▼ Browse
Show advanced settings		Apply and	d Close Cancel

Figure 7. Code Composer Studio™ Compiler Version

10



Properties for 6lowpan_mesh_ti_	15_4_cc1312_tirtos_ccs	
type filter text	General	↓ → → ▼
<ul> <li>&gt; Resource</li> <li>General</li> <li>&gt; XDCtools</li> <li>&gt; ARM Compiler</li> <li>Processor Options</li> <li>Optimization</li> <li>Include Options</li> <li>ULP Advisor</li> <li>Predefined Symbols</li> <li>&gt; Advanced Options</li> <li>&gt; ARM Linker</li> <li>ARM Linker</li> <li>ARM Hex Utility [Disablec</li> <li>Debug</li> </ul>	Configuration: debug_poll [Active]   Project Project Products XDCtools version: 3.51.1.18_core  Products and Repositories Order  P = A Grace (MSP430)  P = A Inter-processor Communication  P = A SyS/BIOS  P = A SimpleLink CC13x0 SDK	Add Edit Edit Remove
<	▷ ■ A SimpleLink CC13x0 Source SDK         ▲ ▼ ▲ SimpleLink CC13x2 26x2 SDK         ▼ ▲ SimpleLink CC13x2 SDK         Target       ti.targets.arm.elf.M4F         Platform:       ti.platforms.simplelink:CC1312R1F3         Build-profile:       release	Select All Deselect All
(?) Show advanced settings	Apply an	d Close Cancel

Figure 8. Code Composer Studio<sup>™</sup> XDC Tool and SDK Versions

To implement the 6LoWPAN mesh design on the CC1352R wireless MCU, the same changes can be made with the CC1352R TI-15.4 sensor example from SimpleLink<sup>™</sup> CC13x2 and CC26x2 SDK v2.40. The step-by-step procedure is given below:

- 1. Import CC1352R TI-15.4 sensor example from SimpleLink<sup>™</sup> CC13x2 and CC26x2 SDK v2.40
- 2. Rename CCS project with your project
- 3. Copy and paste the entire directories of 6lowpan, Application and dtls into the project directory of the CC1352R TI-15.4 sensor example
- 4. Overwrite app.cfg, cc13x2lp.cmd and ccfg.c
- 5. Create build configurations and update the CCS property (Predefined symbols and Include options) based on the CC1312R-based 6LoWPAN mesh example project



# 3.1.2.2 6LoWPAN Mesh TI 15.4 Example

This section starts with a software overview followed by details of the software architecture and useful tips for debugging and optimizing the software.

**NOTE:** This reference design provides an open-source based working example that can be a baseline software to develop end-products. The software example is not optimized in RAM or Flash usage and does not guarantee product-level quality.

#### 3.1.2.2.1 Example Overview

This reference design implements a 6LoWPAN mesh network system working with the FH MAC over sub-1 GHz RF. The 6LoWPAN mesh network stacks run on TI-RTOS, which are implemented based on CONTIKI open source.

Figure 9 shows the end-node example software state machine. After power on, the end node starts with the *Jdllc\_deviceStates\_init* state and, after completing the initialization, goes to the *Jdllc\_deviceStates\_fh\_sync* state where the node starts FH synchronization, which consists of two steps, discussed in Section 3.1.2.2.1.1. Once the node completes the first step of the FH synchronization, the state moves to the *Jdllc\_deviceStates\_PA\_done* state. After the end node completes the FH synchronization, it moves to the *Jdllc\_deviceStates\_joined* state, which is a ready state to start RPL route discovery. The green LED (DIO7) on the EVM indicates that the node has completed the RPL-level joining process. The state machine transition has been implemented in /Application/middleware.c.

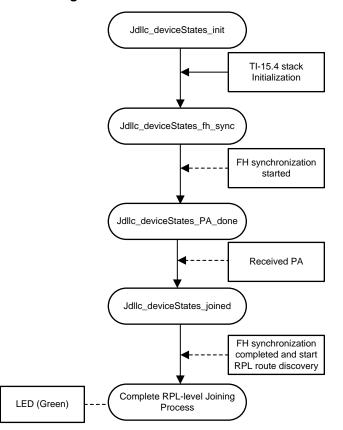


Figure 9. Software State Machine



#### 3.1.2.2.1.1 FH Synchronization

FH synchronization is required to discover the FH network and to synchronize the FH timing and schedule. The underlying TI 15.4-Stack adopts WI-SUN FAN v1.0-based mechanism using four command frames:

- PAN advertisement (PA)
- PAN advertisement solicit (PAS)
- PAN configuration (PC)
- PAN configuration solicit (PCS)

FH synchronization starts with discovering neighbors, or candidates of tracking parents, by performing active scan. The end nodes start with sending PAS commands at the time chosen by the trickle algorithm (RFC 6206). The PAS is sent over all the FH channels from the lowest channel number to the highest in sequence as the nodes do not know the FH timing and schedule at this time. As a response to the PAS, the PA is sent by the nodes that has already joined to the FH network. When the end nodes receive multiple PAs during the scan period (SCAN\_TIMEOUT\_VALUE), one will be chosen based on the link-level metric and then update unicast FH timing, schedule, and the PAN information with the tracking FH parent.

The next step is to send the PCS in the same way as the PAS. Once the nodes receive the PC as a response, they update the broadcast FH timing and schedule and the Group Transient Key (GTK) hash information. Receiving PC as a response of the PCS completes the FH synchronization process, which is ready to receive and send data at the network layers.

The FH timing and schedule correction is done with the received data packets that contain FH unicast and broadcast timing and schedule information elements (IEs).

**NOTE:** The FH synchronization mechanism implemented in this reference design is TI proprietary and is not WI-SUN FAN standard compliant.

Table 5 summarizes the trickle algorithm parameters used for FH synchronization. Depending on the network size, the parameters may need to be adjusted. These parameters are defined in /Application/middleware.h.

PARAMETER	VALUE	DESCRIPTION
TRICKLE_TIMEOUT_VALUE	6 seconds	Discovery trickle timer minimum timeout for PAS and PCS
TRICKLE_MAX_BACKOFF	3	Discovery trickle timer backoff exponent for PAS and PCS
SCAN_TIMEOUT_VALUE	20 seconds	Discovery trickle timer timeout for PA and PC

#### **Table 5. Trickle Algorithm Parameters**

#### 3.1.2.2.1.2 Keep-Alive Mechanism

The goal of the keep-alive mechanism is to detect FH sync loss throughout monitoring sync error conditions such as data transmission and reception failures. The keep-alive mechanism broadcasts keep-alive frames (a 10B TI proprietary message defined in the example) to the link-level neighbors periodically (KEEP\_ALIVE\_TX\_TIMEOUT\_VALUE) once end nodes join to the FH network. The keep-alive TX timer is reset when broadcast frames other than keep-alive frames are sent, which reduces the keep-alive traffic overheads to the network.

If end nodes do not receive broadcast frames from their tracking parents in series (KEEP\_ALIVE\_MAX\_ATTEMPTS), FH sync loss occurs and the end nodes move immediately to the *Jdllc\_deviceStates\_fh\_sync* state to restart the FH synchronization process.

In addition to the keep-alive transmissions, each node traces unicast transmission failures to the target parents. If TX attempts fail in series (MAXIMUM\_NUM\_DATA\_FAILURE), this indicates the FH sync loss, which results in moving to the *Jdllc\_deviceStates\_fh\_sync* state to re-start the FH synchronization process. Table 6 summarizes the keep-alive parameters. The keep-alive parameters are defined in /Application/middleware.h.



Hardware, Software, Testing Requirements, and Test Results

PARAMETER	VALUE	DESCRIPTION
KEEP_ALIVE_TX_TIMEOUT_VALUE	60 seconds	Keep-alive TX interval for parents
KEEP_ALIVE_MAX_ATTEMPTS	5 times	The maximum number of failed keep-alive RX in series to indicate FH sync loss
MAXIMUM_NUM_DATA_FAILURE	5 packets	The maximum number of TX failure to indicate FH sync loss

#### 3.1.2.2.1.3 MAC Data Encryption

The MAC data encryption follows IEEE 802.15.4 standard with the security level options of MIC-32, MIC-64, MIC-128, ENC, ENC-MIC-32, ENC-MIC-64 and ENC-MIC-128.

In the software example, the default mode is set to the security level of *ApiMac\_secLevel\_encMic32* and the key ID mode of *ApiMac\_keyIdMode\_8*, defined in the /Application/jdllc\_middleware.c. The TI 15.4-Stack supports the pre-shared key mechanism based on IEEE 802.15.4 standard. The security key should be pre-programmed in the software, and all of the nodes in the same network must share the same pre-shared key.

Because the MAC data encryption mechanism requires a node to register its neighbors that use the data encryption, it is required to have a discovery phase throughout unsecured data exchanges. The PAS and PA frames used for the FH synchronization must be sent without data encryption. In addition, the software example uses unsecured transmissions for PC, PCS, and broadcast frames to minimize a chance that some neighbors with a better route are not detected during the discovery phase.

#### 3.1.2.2.1.4 DTLS Encryption

This software example uses the tinyDTLS library for the DTLS. The current version of tinyDTLS supports DTLSv1.2 with SHA256. DTLS v1.2 is based on TLS 1.2 and runs over UDP to provide end-to-end secured transport.

This design implements a TI proprietary MAC-level DTLS mechanism. This consists of using the MAC encryption mechanism with the private key generated by DTLS that runs on application and negotiating the key with a two-way handshake mechanism. The negotiation process is performed between link-level neighbors and thus each pair of neighbors shares a different private key.

#### 3.1.2.2.1.5 RPL Routing

The 6LoWPAN mesh software example uses the RPL protocol for multi-hop routing. The network formation with the RPL routing is initiated by broadcasting Destination Oriented Directed Acyclic Graph (DODAG) information object (DIO) by the root node. Once child nodes receive the DIOs, they broadcast the DIOs and send back a unicast destination advertisement object (DAO) packet to the parents that provide the best route toward the root. The DIO and DAO transmission times are determined by the trickle algorithm (RFC 6206). For the RPL metric to decide the best route, the expected transmission count (ETX) is used by default. For details on RPL routing, see the RFC standard RFC 6550 or the TI training video on Wireless Network Challenges and Solutions for a Smarter Grid IoT.

#### 3.1.2.2.1.6 6LoWPAN

The goal of the 6LoWPAN protocol is to support the IP services by reducing the gap between IPv6 and lower stacks to serve IPv6 applications on the low-end devices typically restricted in processing power, memory, and energy. The primary tasks of the 6LoWPAN are fragmentation and reassembly, IPv6 and UDP header compression, stateless IPv6 address auto-configuration, and neighbor discovery optimization. For details of the 6LoWPAN protocol, see the RFC standards RFC 4944 and RFC 6282 or the TI training video on Wireless Network Challenges and Solutions for a Smarter Grid IoT.

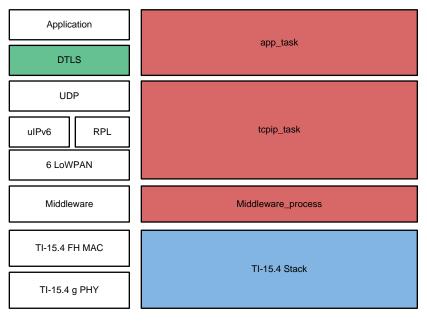


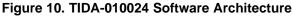
#### 3.1.2.2.2 Software Architecture

The 6LoWPAN mesh end-node software consists of TI 15.4-Stack, middleware layer interconnecting between the TI 15.4-Stack and upper layers of 6LoWPAN mesh stacks and the application layer. The middleware layer initializes and configures the TI 15.4-Stack and processes incoming data from the TI 15.4-Stack or mesh network stacks. The 6LoWPAN mesh stacks cover 6LoWPAN, RPL, IPv6/ICMPv6 and UDP protocols. The software example provides two types of UDP applications that can be configured at compile time: UDP poll and push examples.

This design also implements DTLS data encryption. A simplified two-way DTLS handshake mechanism (MAC-level DTLS) is implemented to minimize the transaction overhead on network performance.

Figure 10 shows the overall software architecture. The software example was implemented based on the sensor\_cc1312r1lp example available in the SimpleLink<sup>™</sup> CC13x2 and CC26x2 SDK v2.40. The middleware layer runs the middleware\_process that manages the FH synchronization and keep-alive mechanism, and interfaces between TI 15.4-Stack and mesh network stacks to handle incoming data. The middleware layer follows the same software architecture using the iCall service defined in the sensor\_cc1312r1lp example. The tcpip\_task covers the mesh network layers of 6LoWPAN, uIPv6, RPL and UDP layers. If a build configuration using DTLS, the encryption/decryption layer will be added between the UDP and the application layer. The UDP poll and push examples are parts of the app\_task as the application layer.





## 3.1.2.2.2.1 TI-15.4 PHY Configuration

The TI 15.4-Stack supports multiple options for the frequency band and mode to run the FH mode. The software example configures the PHY mode to APIMAC\_STD\_US\_915\_PHY\_1. The default configuration can be updated in the /Application/subg/config.h. The following codes list all the options for the PHY ID, which can be found in /Application/api\_mac.h.

**NOTE:** The TI design software example was verified with the FH operation over the PHY mode of APIMAC\_STD\_US\_915\_PHY\_1.

```
/*! PHY IDs - 915MHz US Frequency band operating mode # 1 */
#define APIMAC_STD_US_915_PHY_1 1
/*! 863MHz ETSI Frequency band operating mode #1 */
#define APIMAC_STD_ETSI_863_PHY_3 3
/*! 433MHz China Frequency band operating mode #1 */
#define APIMAC_GENERIC_CHINA_433_PHY_128 128
```

15



Hardware, Software, Testing Requirements, and Test Results

www.ti.com

/\*! PHY IDs - 915MHz LRM US Frequency band operating mode # 1 \*/
#define APIMAC\_GENERIC\_US\_LRM\_915\_PHY\_129 129
/\*! 433MHz China LRM Frequency band operating mode #1 \*/
#define APIMAC\_GENERIC\_CHINA\_LRM\_433\_PHY\_130 130
/\*! 863MHz ETSI LRM Frequency band operating mode #1 \*/
#define APIMAC\_GENERIC\_ETSI\_LRM\_863\_PHY\_131 131
/\*! PHY IDs - 915MHz US Frequency band operating mode # 3 \*/
#define APIMAC\_GENERIC\_US\_915\_PHY\_132 132
/\*! 863MHz ETSI Frequency band operating mode #2 \*/
#define APIMAC\_GENERIC\_ETSI\_863\_PHY\_133 133

# 3.1.2.2.2.2 Middleware Layer

The middleware layer initializes the TI 15.4-Stack, maintains MAC-level keep-alive mechanism, registers security entries to the TI 15.4-Stack, processes packets incoming from the TI 15.4-Stack or upper layers, handles message timeout and erroneous transmissions, performs FH synchronization and maintains the state machine which can be one of four states: *Jdllc\_deviceStates\_init*, *Jdllc\_deviceStates\_fh\_sync*, *Jdllc\_deviceStates\_PA\_done*, and *Jdllc\_deviceStates\_joined*.

Anend-node starts with the *Jdllc\_deviceStates\_init* state to reset the TI 15.4-Stack to configure initial MAC PIB and FH PIB values and to start the PAN as end node. The state changes to the *Jdllc\_deviceStates\_fh\_sync* state when the node initiates the FH synchronization process. As the first step of the FH synchronization, if the node receives PA as a response of the PAS command, it moves to the *Jdllc\_deviceStates\_PA\_done* state and proceeds to send the PCS command. After the FH synchronization completes by receiving PC commands, the node moves to the *Jdllc\_deviceStates\_joined* state, which is the ready state to send data at the mesh network layers.

Similar to the sensor\_cc1312r1lp example, the communications between TI 15.4-Stack and the middleware is done through iCall messages. Conversely, the message exchanges between the mesh network stacks and the middleware layer are done through mailbox posting or direct MAC API calls. For the details on the iCall framework, refer to ti-15.4-stack-users-guide in docs/ti154stack under the installation directory of SimpleLink<sup>™</sup> CC13x2 and CC26x2 SDK v2.40.

## 3.1.2.2.2.3 Network Layers (6LoWPAN, IPv6, RPL, and UDP)

The network stacks are implemented based on the CONTIKI open source. The tcpip\_task (in 6lowpan/uip\_rpl\_task.c) processes messages incoming from the application and lower layers through the mailbox. For details of the implementations, refer to the CONTIKI open source website.

## 3.1.2.2.2.4 DTLS Layer

The DTLS layer is used for the data encryption between the UDP layer and the Application layer. This layer is based on the tinyDTLS library. Two version of DTLS were implemented in the software : DTLS v1.2 and MAC-Level DTLS.

The MAC-level DTLS is the mechanism implemented in the software example. With the MAC-level DTLS, a node starts with the pre-shared key encryption based on IEEE 802.15.4 MAC encryption and then MAC-level DTLS handshake negotiates the private key with its neighbor. Once the handshake completes, the node encrypts data with the private key instead of the pre-shared key.

## 3.1.2.2.2.5 Application Layer

The example includes two types of UDP applications: UDP poll and UDP push. The app\_task opens the UDP socket (UDP client for the end-node) with known port numbers and starts UDP data transmissions. Depending on the UDP examples, the initiator of the UDP data is different. For the UDP poll example, the root node initiates the poll message to each node to read data. For the UDP push example, each node initiates data transmissions whenever there is application data to send.

The Device Language Message Specification (DLMS) and Companion Specification for Energy Metering (COSEM) for smart metering applications use the poll-based mechanism, and the Constraint Application Protocol (CoAP) uses the mix of UDP poll (GET command) and push (periodic OBSERVE command) mechanisms. The target end-product applications can be easily integrated on top of the given UDP application.



#### 3.1.2.2.2.6 LED Configuration

Table 7 summarizes the LED configuration in the software example. The board\_led\_type\_LED1 toggles when transmission or reception events happen. The board\_led\_type\_LED2 turns on when end nodes join to the RPL network.

NAME (PIN NUMBER)	EVENT	ACTION
board_led_type_LED1 (DIO6)	Data TX/RX	TOGGLE (Red)
board_led_type_LED2 (DIO7)	Complete RPL-level Joining	ON (Green)

Table 7. LED Configuration (CC1312R)

#### 3.1.2.2.3 Tips for Debugging and Optimization

This section provides useful tips with the Code Composer Studio tool to debug and optimize the software example in the end-product development phase.

### 3.1.2.2.3.1 Running in Debug Mode

To debug software, run the software in debug mode. The Code Composer Studio tool provides the debug mode operation. To run in debug mode with the Code Composer Studio tool:

- 1. Launch the target configuration for the target device as shown in Figure 11.
- 2. Connect the target EVM, load the program, and then run the software as shown in Figure 12.
- 3. Add global variables to debug in the *Expressions* tab as shown in the right top of Figure 12 to trace the variables in the debug mode.

# Figure 11. Code Composer Studio™ Debug: Launching Debug Window

	•  4, •  8  # •  8     0 0 • 0 •							uick Access
roject [ # " D	R mains II		- 0	* Target Configurations	0			X X 2 8 "
E 📚 🔻			^	type filter text				
6lowpan_ti_15_4_cc1				> @ Projects				
6 Binaries	3 @file main.c			v a User Defined				
Ø Includes	5 @brief main entry of the example application			E CC1312 L21000	OR.com			
i lowpan	6			R CC1312 L21000				
<ul> <li>Application</li> <li>debug poll</li> </ul>	7 Group: WCS LPC 8 Target Device: CC13xx			8 CC1312 L21000	HLcom			
<ul> <li>b debug_poll</li> <li>b debug_poll_dtls</li> </ul>	9			K CC1312_L21000	I0.coml			
debug_poi_ats	10			R CC1312_L21000	(1.comi			
debug push dtls	11 12 Copyright (c) 2016-2018, Texas Instruments Incorporated			R CC1312_L21000	K New Target Configu	ration		
> debug_push_root	13 All rights reserved.			R CC1312_L2100(		iguration		
e debug root poll	14			R CC1312_L2100		0	Delete	
a debug root poll e	15 Redistribution and use in source and binary forms, with or without 16 modification, are permitted provided that the following conditions			K CC1312_L2100;			F2	
b debug root push	10 modification, are permitted provided that the following conditions 17 are met:			K CC1312_L2100	📌 Refresh		F5	
e dtls	18			K CC1312_L2100	<ul> <li>Remean</li> <li>Launch Selected Co</li> </ul>	nfiguration		
targetConfigs	19 * Redistributions of source code must retain the above copyright			E CC1312_L2100/ E CC1312_L2100/				
Tools	20 notice, this list of conditions and the following disclaimer. 21			E CC1312_12100				
> > cc13x2lp.cmd	22 * Redistributions in binary form must reproduce the above copyright			K CC1312_L2100				
README.html	23 notice, this list of conditions and the following disclaimer in the			K CC1312 L21002		Alt+	Enter	
README.md	24 documentation and/or other materials provided with the distribution. 25			R CC1312 L21002				
	25 26 * Neither the name of Texas Instruments Incorporated nor the names of			R CC1312 L30020				
	27 its contributors may be used to endorse or promote products derived				1312 L2100011.com			
	28 from this software without specific prior written permission.			K TM4C.coml				
	29 30 THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"							
	31 AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO,			·				
	32 THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR							
	33 PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR							
	34 CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, 35 EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,							
	36 PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS;							
	37 OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,							
	38 WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR 39 OTHERMISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE.							
	40 EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.			·				
	41		~					
>	(		>	Click the New button to	create a new target con	iguration file. Click	sere to hide the	
Console II		• • • • • E Problems =						> * * *
consoles to display at t	his time.	0 items						
		Description	~	Resource	Path	Location	Type	



Secure Jerse     Abaced Step       To startio-decision Brown of onlygation about the tage).     Iggl Configation applies the tage).       Concent     Note tabelences to spratic about the tage).       Concent     Note tabelences to spratic about the tage).       Concent     Concent       Concent     Concent <tr< th=""><th>File Edit View Project Tools Run Scripts Window Help</th><th></th><th></th><th></th><th></th><th></th></tr<>	File Edit View Project Tools Run Scripts Window Help					
************************************	3 • 8 6 8 × 8 • 7 × 7 8 8 × 8 • 2 2 • • • • • 1 • • 1 7 × • € • 18					
** mining mining   ** mining </td <td></td> <td>R = = □ → Variables M Expressions II = Register</td> <td>15</td> <td></td> <td>6 🖪 B I</td> <td>· · · · · · · · · · · · · · · · · · ·</td>		R = = □ → Variables M Expressions II = Register	15		6 🖪 B I	· · · · · · · · · · · · · · · · · · ·
Image: Description of the sequence						
# 000000, 00000000000000000000000000000	Texas Instruments XDS110 US8 Debug Probe_0/Contex_M4_0 (Running)					
# disklability     # disklability    # disklability     # disklability     # disklability     # disklability     # disklability     # disklability      # disklability      # disklability      # disklability     # disklability         # disklability </td <td></td> <td></td> <td></td> <td></td> <td></td> <td></td>						
<pre>inter transmitter transmi</pre>						
<ul> <li></li></ul>						
# #dpdx.axtbd       usgued de[t]       0 \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$						
Add recognition    Children opposition Children opposit						
CCU12_U200ALconf # CCU12_U200ALconf   CCU12_U200ALconf # CCU			unsigned char[8]	[5 \x05',223 \xdf',154 \x9a',24 \x18',0 \x00']	0x20003CE0	
Also     Also <td></td> <td>Add new expression</td> <td></td> <td></td> <td></td> <td></td>		Add new expression				
alc   Construit   C						
all     All <td></td> <td></td> <td></td> <td></td> <td></td> <td></td>						
Also     Also <td></td> <td></td> <td></td> <td></td> <td></td> <td></td>						
alc   Construit   C						
abit       Amage Support Suppo						
alc   Construit   C						
all     All <td></td> <td></td> <td></td> <td></td> <td></td> <td></td>						
immediate     Advanced feety       Kine device the proper complycation double the trapt.       immediate     immediate       immediate     immediate <tr< td=""><td></td><td></td><td></td><td></td><td></td><td></td></tr<>						
ha usedine dropend ondyperior Adod the target. Second Target Second Tar	asic					10
Cancel In Instruments 2023112 USE blodge hole     Kanel or Deval     App Bar of        Calcified Deval     Calcified Deval     Calcified Deval        Calcified Deval        Calcified Deval </td <td>General Setup</td> <td></td> <td>Advanced Setup</td> <td></td> <td></td> <td></td>	General Setup		Advanced Setup			
Isonal or Dotto:	This section describes the general configuration about the target.					
Jend of Dole   Jend of Dole   C (2019928)    C (2019928)	Constant in the constant of th					
C (2)199/28       Image: C (2)199/28         C (2)198/28       Image: C (2)199/28         More C (2)199/28       Image: C (2)199/28         More C (2)199/28       Image: C (2)199/28         Image: C (2)199/28       Image: C (2)199/28		~	Target Configuration: lists the configuration option			
Image: Construint of the special conspecial consthe special construint of the special const		v		is for the target.		
C 130844       The C construint         C 130844       The C construint         C 13084713       The C construint         C 13384713       The C construint         D 1 C construint       The Construint         D 2 C construint       Th	Board or Device type filter text	v		ns for the target.		
Collability       Intel accession, all designs multi hard bees seed; the constraint one series of the constraint one seri	Board or Device type filter text	~	Save Configuration	ns for the target.		
	Board or Device type filter text	*	Save Configuration	ns for the target.		
Image: Construint in the second se	Board or Device type films text CC1310F28 CC1310F22 CC1310F44	~ 	Save Configuration Save Test Connection			
	Board or Divisio C C 13 0F 28 C C 13 0F 28 C C 13 0F 28 C C 13 0F 42 C C 13 0F 44 C C 13 0F 44		Save Configuration Save Test Connection To test a connection, all changes must have been s	avej, the		
Alerator Communication  Alerat	Board of Divisor Spee litter text CC13109728 CC13109728 CC1310974 CC13109742 CC1310974 CC1310	×	Save Configuration Save Test Connection To test a connection, all changes must have been s	avej, the		
Image: Construint of the structure of the s	Road on Device 'syse Blan fast C C1398128 C C1398128 C C1398128 C C1398149 C C139814 C C13	×	Save Configuration Sove Test Connection To be a connection, all changes must have been a configuration file occetains no errors and the conne	avej, the		
COAMP 128       Intel Communication       Intel Communication       Intel Communication         Net generation CC1310 device, the SingleCisin(M) Sub-1 Give CC1312 aimless MCU       Intel Communication one ULML trapt application reveloces yn to folktier data conner implementation one ULML trapt application reveloces in the trapt application r	Iterat or Device yape Bee Sed 0 00119738 0 00119738 0 00119738 0 00119748 0 00019748 0 00119748 0 00119748 0 00119748	×	Save Configuration Save Test Connection To test a connection, all changes must have been so configuration file contains no errors and the connec Test Connection	avej, the		
Code# 108       Image on additional state of the code of the c	Rand or Device 'spas film fast C C1399732 C C1399732 C C139974 C C1399743 C C1399743 C C1399743 C C1397473 C C1397473 C C13967473 C C13967473		Save Configuration Save Test Connection To test a connection, all changes must have been so configuration file contains no errors and the connec Test Connection	avej, the		
Note generation CCU3D device, the SingleLisic(MS Sub 1 Get CCU3D anives MCU  Information on ULMI, topic registration	Inand or Deniar		Save Configuration Save Test Connection The Connection To the a connection, all charges must have been is configuration for context no emory and the cone Test Connection Alternate Connection	avej, the		
Inter generative Color winds, is a sequelation of the Color is interest and provide and provide in the provide interest operation on how to provide interest operation on how to provide interest operation on how to provide interest operation of the Monte, disk the Add botton.       Nate: Support for more devices may be available from the update manager.     To ald a port of the least operation of the Monte, disk the Add botton.       Nate: Support for more devices may be available from the update manager.     To ald a port of the least operation of the Monte, disk the Add botton.       Nate: Support for more devices may be available from the update manager.     To ald a port of the least operation of the Monte, disk the Add botton.       UDL2/2000/Honord     To provide in the update generation of the Monte, disk the Add botton.	Inad or Device gaps Rev text CCU30738 CCU30738 CCU30748 CCU308448 CCU308448 CCU308448 CCU308448 CCU308448 CCU308448 CCU30849		Save Configuration Save Test Connection The Connection To the a connection, all charges must have been is configuration for context no emory and the cone Test Connection Alternate Connection	avej, the		
Note: Support for more devices may be available from the update manage.	Nand at Device 'gyet Bine test CC139732 CC139723 CC13974773 CC1397777 CC13977777777777777777777777777777777777		Save Configuration Save For Connection The Configuration necessar The Onether to table [a Proceeding necessar The Onether to table [a Proceding necessar The Onether to table [	and, for this type separat this function. Y to facilitate data		
Note: Sapport for more devices may be available from the update manager: A Manue Sapport for more devices may be available from the update manager: A Manue Sapport for more devices may be available from the update manager: A Manue Sapport for more devices may be available from the update manager: This Sapport for more devices may be available from the update manager: This Sapport for more devices may be available from the update manager: This Sapport for more devices may be available from the update manager: This Sapport for more devices may be available from the update manager: This Sapport for more devices may be available from the update manager: This Sapport for more devices manager: This Sapport for mo	Inand or Device	ine MOI	Save Configuration	end. Se ficin type supports the function.		
Nate Support for more devices may be available from the update manager. (a) Advanced Source: (a) Advanced Source: (b) Advanced So	Intered or Device	N No	Save Configuration Save Test Connection, all charges must have been is configuration for contain no more and the conve feet Connection. Alternatic Communication (Unit Communication Unit Communication Communicati	and, for data type supports this function. In the solution data for the solution another Theorem Englance Typesar		
State Support for more devices may be available from the update manager.     State of the second secon	Intered or Device	ies MCU	Save Configuration  Samp  For Consolidate  For Consolidat	and, for data type supports this function. In the solution data for the solution another Theorement Epidemet Typesor		
State Support for more devices may be available from the update manager.     State of the second secon	Intered or Device	ies MCU	Save Configuration  Samp  For Consolidate  For Consolidat	and, for data type supports this function. In the solution data for the solution another Theorement Epidemet Typesor		
Concle = 16 di 16 4 ° ° ° ° ° ° ° ° ° ° ° ° ° ° ° ° ° °	Issuel or Device QCUSIOT32 CCUS	New MCJ	Save Configuration Samp Test Connection, all charges must have been as configuration for contain no eners and the tones for Connection Material Connection (Lett Connection) Toget application beams of the second provided and the second	and, the class type supports this function. Y to facilitate data to include a monitor Theorem Equipore Type of Beed documentation on have to		
1012/2000/Honded # Provide the set of the se	Inter die Deven geschlasse das devensionen COSISIENTE COSISIENT	ies MCU 1	Save Configuration Samp Test Connection, all charges must have been as configuration for contain no eners and the tones for Connection Material Connection (Lett Connection) Toget application beams of the second provided and the second	and, the class type supports this function. Y to facilitate data to include a monitor Theorem Equipore Type of Beed documentation on have to		
rtex_M4_0: GEL Output: Memory Map Initialization Complete.	Isoad or Device Question of the fact COUSE of the	v	Save Configuration           Imm           Text Connection           Text connection           Text connection           Text connection           Text connection           Connection <td>and the tion type seports the function. Y to fulfither data B to Robel as monotor B to Robel as monotor then documentation on how to write, dock the Add buttor.</td> <td></td> <td></td>	and the tion type seports the function. Y to fulfither data B to Robel as monotor B to Robel as monotor then documentation on how to write, dock the Add buttor.		
rtex_M4_0: GEL Output: Memory Nap Initialization Complete.	Note of Device 'gene Bile load COSING COSING AND COSI	v	Save Configuration           Imm           Text Connection           Text connection           Text connection           Text connection           Text connection           Connection <td>and the tion type seports the function. Y to fulfither data B to Robel as monotor B to Robel as monotor then documentation on how to write, dock the Add buttor.</td> <td></td> <td>-84</td>	and the tion type seports the function. Y to fulfither data B to Robel as monotor B to Robel as monotor then documentation on how to write, dock the Add buttor.		-84
	Name of Device "specifies in the control of the con	v	Save Configuration           Samp           That connection, all charges must have been is configuration for contexts no eners and the tone for connection.           Matter Connection           Matter Connection           Matter Connection           Matter Connection           Interface Connection           Matter Connection           Sample Configuration secretary configuration secretary contexts for the start by the start of the secretary regulation by regulating the secretary project in the segret application by the secretary configuration is the term mode.           To add a port in the terms application for the Matter application for the Matte	end, the click hyse separate this function. Wy to facilitate data this social and another thesaure explores if your because explores if your social click the Add buttor.		-84

# Figure 12. Code Composer Studio™ Debug: Running Software in Debug Mode

Table 8 summarizes some global variables to debug the software example.

VARIABLES	DESCRIPTION
TCPIP_Dbg	Debug counts for UDP, IPv6, ICMPv6, RPL layers
LOWPAN_Dbg	Debug counts for 6LoWPAN layer
devInfoBlock	Device state machine and PAN information
mwDbg	Debug counts for middleware layer
macPib	TI-15.4 MAC PIB
FHPIB_db	TI-15.4 FH MAC PIB
joinedDODAG	Flag to indicate RPL DODAG join state
ApiMac_extAddr	8B extended MAC address

#### Table 8. Global Variables for Debugging

#### 3.1.2.2.3.2 Sniffer Setup

A sniffer can be used to read the code with Wireshark<sup>™</sup>. This is useful for debugging purposes such as encryption verification. This sniffer will require use of an extra CC13x2R LaunchPad configured as a sniffer and connected to a computer to log the packets.

Install and set up SmartRF Protocol Packet Sniffer 2 using the SmartRF Packet Sniffer 2 User's Guide.

The software must be configured correctly to facilitate the packet reception. The sniffing process may present challenges because this design uses frequency hopping, and the frequency that must be sniffed often changes... To avoid such complications, the TIDA-010024 software can be configured with the frequency hopping mode of using a single channel.

In the file Application/subg/config.h, use the following code to configure the software as one channel communication:

/\*! Use only one channel for sniffing purpose \*/
#define SINGLE\_FH\_CHANNEL

Once the software is configured and compiled, launch Wireshark to start sniffing as shown in Figure 13.

#### Figure 13. Wireshark<sup>™</sup> Sniffer Window

Captu	ring from \\\pipe\tiw	spc data				6	- 0
File			Telephony Wireless Tool	s Help			
		X C 4 + + = T .					
			• <u> </u>				
	y a display filter <ctrl< th=""><th></th><th></th><th></th><th></th><th></th><th>Expression</th></ctrl<>						Expression
0.	Time	Source	Destination	Protocol	Length Info		
	5 48.496882	00:12:4b:00:18:9a:df:05		TI 802.15.4GE	187 Data, Src: 00:12:4b:00:18:9a:df:05, Data Frame		
	6 48.537292	00:12:4b:00:18:a8:5c:d0		TI 802.15.4GE	187 Data, Src: 00:12:4b:00:18:a8:5c:d0, Data Frame		
	7 120.737753	00:12:4b:00:18:a8:74:ac		TI 802.15.4GE	187 Data, Src: 00:12:4b:00:18:a8:74:ac, Data Frame		
	8 120.778108 9 120.805694	00:12:4b:00:18:a8:59:3c 00:12:4b:00:18:9a:de:db		TI 802.15.4GE TI 802.15.4GE	187 Data, Src: 00:12:4b:00:18:a8:59:3c, Data Frame		
	9 120.805694	00:12:4b:00:18:9a:de:db 00:12:4b:00:18:a8:62:7e			187 Data, Src: 00:12:4b:00:18:9a:de:db, Data Frame		
	10 120.862239	00:12:40:00:18:88:62:7e 00:12:4b:00:18:88:5c:d0	00:12:4b:00:18:a8:57:c1	TI 802.15.4GE	187 Data, Src: 00:12:4b:00:18:a8:62:7e, Data Frame 235 Data, Dst: 00:12:4b:00:18:a8:57:c1, Src: 00:12:4b:00:18:a8:5c	10 Parts France	
	12 250.124335	00:12:40:00:18:a8:5C:00 00:12:4b:00:18:a8:57:c1	00:12:40:00:18:a8:5/:C1 00:12:4b:00:18:a8:5c:d0	TI 802.15.4GE	111 Ack, Dst: 00:12:40:00:18:a8:57:c1, Src: 00:12:40:00:18:a8:57: 111 Ack, Dst: 00:12:4b:00:18:a8:5c:d0, Src: 00:12:4b:00:18:a8:57:		
	12 250.124335	00:12:40:00:18:a8:57:01 00:12:4b:00:18:a8:5c:d0	00:12:40:00:18:a8:5C:00 00:12:4b:00:18:a8:57:c1		150 Data, Dst: 00:12:40:00:18:a8:57:c1, Src: 00:12:40:00:18:a8:57:		
	14 250.163951		00:12:40:00:18:a8:5/:C1 00:12:4b:00:18:a8:5c:d0		111 Ack, Dst: 00:12:40:00:18:a8:57:c1, Src: 00:12:40:00:18:a8:57:		
	14 250.105951	00.12.40.00.10.80.57.01	00.12.40.00.10.80.5C.00	11 002.15.400	111 ACK, DSC. 00.12.40.00.10.80.50.00, SFC. 00.12.40.00.10.80.57.	ci, Ack Frame	
		ersion 4, Src: 192.168.1. col. Src Port: 17760. Dst					
	adio Packet Inf		POPT: 17/60				
		o HY without Mode Switch					
	HR: 0x08bd with						
		= Mode Switch: False					
		= FCS Type: Four Byte:	(0-0)				
		= Data Whitening: True					
		101 = Frame Length: 189					
			c:d0, Src: 00:12:4b:00:18:	a8:57:c1			
Dat	(132 bytes)		-				
0	ata: 2ceaab00683	3c22180f5bd731a7d33f65d41e	e05e189bf41cf				
	Length: 132]						
000	45 00 00 eb 00 0	00 00 00 80 11 b6 ab c0 a	8 01 03 E				
		60 45 60 00 d7 cd df 00 c					
		3 33 33 00 00 e7 80 08 b					
		8 18 00 4b 12 00 c1 57 a					
		Lb ee ee ee ee ee ee ee					
		12 77 b4 07 15 02 9d 0a 9					
		ab 00 68 3c 22 18 0f 5b c 15 e1 89 bf 41 cf 93 4d 2					
		5 84 81 42 96 fa a9 66 b					
		b ff 6a 3c 21 78 2f 02 6					
eae	ea b0 12 ac 71 6	0 91 de bc f2 e8 54 25 2	0 57 76q`T% k	N			
		ed cc 56 02 9c 5b 57 78 6					
		46 8e 56 1e 10 ec 7d 00 a Fe 01 d7 48 cc 28 d3 e2 d					
		re 01 07 48 cc 28 03 e2 c 2b f6 28 75 a2 09	ld d1 70 '.i.c H.(	P			
090	ru oo za 98 04 2	10 10 20 75 82 09	+.( u				
7	ength (data.len)					Packets: 14 · Displayed: 14 (100.0%)	Profile:



#### 3.1.2.2.3.3 ROV Analysis

CCS provides a useful tool to debug the software, RTOS Object View (ROV). The ROV tool helps to address various software crash issues or to optimize software examples by analyzing peak memory usage. To debug software with the ROV tool:

- 1. Suspend debug mode and open ROV as shown in Figure 14. Figure 15 shows the screen capture of ROV (on the right-bottom).
- 2. As an example, go to the Detailed and CallStacks taps in the Task menu to optimize the stack size, to trace the call stacks, or to address the stack overflow issue.

#### Figure 14. Code Composer Studio™ Debug: Launch ROV

G debug_3 - C//Usen	s/a0923532/s7/CC	STargetConfigurations\CC1312_L210026N.coml - Code Com	nposer Studio							D)	-	ø	×
File Edit View Proje	ect Tools Run 1	Soripts Window Help											
Memory Map	1	SA-DDA-4-19-1895-1004	e •								Quick A	coess i 🕫	
GEL Files				Variables I Expressions II I Regist						6			
On-Chip Flash		and a Derive Delevation											
ARM Advanced F		omposer Studio - Device Debugging] Debug Probe_0/Cortex_M4_0 (Suspended)		Expression		Type struct cunnamed>		Value EntRandZero=0 'u00' kaRaTimeOut=2 'u02' ontKaRaInd.	Address				
Debugger Option		F134		# mwDbg # TCPIP_Dbg		struct_trpip_debug		httpip_tmout_cnt=0.tcpip_tmout_post_err=0.tcpip_input_					
Save Memory		() at PowerCC2EX2_tirtos.c157 0x00013CAA		B LOWPAN Dbg		struct (unnamed>		(total_rx_packet=3364.buf_full_drop_segment=0.frag_n_h.					
Load Memory		26X2 c:241 0x0001E406		# devinfoBlock		struct (unnamed)		[panID=4660.coordExtAddr=[193 'uc1'.87 'W.168 'ua8'.					
Fill Memory	-	e defined)		# macPib		struct <unnamed></unnamed>		(ackWaitDuration=54 %,associationPermit=0 \u00,auto					
				# FHPIB,db		struct <unnamed></unnamed>		(macTrackParentEUI+(202 *)aca),114 Y,168 *)xa8:24 *)x18.	0x20002900				
ROV Classic     Runtime Object V				== joinedDODAG		unsigned char		1 'w01'	0x20003228				
Hardware Trace A				ApiMac_extAddr		unsigned char[8]		[5 %05',223 %df',154 %9a',24 %18',0 %00']	0x20003CE0				
LE System Analyzer				Add new expression									
K RIOS Analyzer													
D Energy/Race**													
Li Graph	>												
di Image Analyzer													
Profile	>												
8 CC1312_L210026N	Looml = ROC13	112_L2100238.coml											° 0
Basic													E D .
General Setup						Advanced Setup						-	
	es the general cos	nfiguration about the target.				Novanceo setup							
		(DS110 US8 Debug Probe				Target Configuration: lists the configuration	untion methods for the l	local distance of the second					
		US110 USE Debug Probe				inget congristed and on compa	record options for the	a get					*
<b>Basic Advanced Sour</b>	rce												
X Target Configuratio	ens II											5 <b>X</b>   2 1	8 - 0
type filter text													
Projects													
- Dier Defined													
# CC1312 L210	000R.coml												
# CC1312,L210	000GA.coml												
B CC1312,1210													
B CC1312_L210													
K CC1312_L210													
# CC1312_L210													
H CC1312_L210 H CC1312_L210													
× CC1312 L210													
# CC1312 L210													
# CC1312 L210	0238.ccxml [Def	fault)											
B CC1312_L210													
8 CC1312_L210													
# CC1312_L210													
B CC1312_L210													
8 CC1312_L210 8 CC1312_L210													~
		arget configuration file. Click here to hide this message.											
CHOICE AND ANY DOLLOT	the original difference	and the second sec						•					
								<u>.</u>				U	



3 • 0 0 0 + 0 € x 0 x 8 0 x 8 0 0 0 € • 0 € • 0 € • 0 x 0 5 • 0 0 0 0 € • 0 € • 0 x 0 5 • 0 0 0 0 0 € • 0 0 0 0 0 € • 0 0 0 0 0	0 🛷 🕶											ccess 🔡 🖬
Debug II	ų ···	°⊡ ∞Var	iables 41	Expression	is II III Reg	gisters				0.00	♦ × ½ #   1	3814
# 0 CC1312_L210021H.comi [Code Composer Studio - Device Debugging]		Expre					Type	Value	Address			
# Texas Instruments XDS110 US8 Debug Probe_0/Contex_M4_0 (Suspended)		> 10 1	edGwm				struct <unnamed></unnamed>	jontRandZero=0 "u00"JkaRxTimeOut=2 "u02	ontKaRulind 0x20003334			
CPUwfi() at cpu/h;215 0x0001F134		> #1	CPIP Db	9			struct_trpip_debug	htpip_tmout_ont=0.tcpip_tmout_post_err=0.	cpip_input 0x20003280			
PowerCC26XX_standbyPolicy() at PowerCC26X2_tirtos.c157 0x00013CAA		> 69.0	OWPAN,	Dbg			struct <unnamed></unnamed>	(total_rx_packet=3364.buf_full_drop_segmen	=0,frag_n_h0x20003268			
Power_idleFunc() at PowerCC26X2.c241 0x0001E406		> 🙂 d	Sevin/oBi	ock			struct <unnamed></unnamed>	(panID=4660,coordExtAddr=[193 1,sc1),87 1	7,168 'ua8' 0x20003988			
0x10038226 (no symbols are defined)		> 🐲 🕫	nacPib				struct <unnamed></unnamed>	(ackWaitDuration=54 '6',associationPermit=0	'u00'.auto 0x20001A90			
			HPI8_db				struct <unnamed></unnamed>	(macTrackParentEUT=(202 *\sca*,114 Yr,168 *\s				
			oinedDO				unsigned char	1 'u01'	0x20003228			
			piMac_e				unsigned char[8]	[5 "xi05",223 "xid?,154 "xi9a",24 "xi18",0 "xi00	_] 0x20003CE0			
			ldd new i	expression								
# CC1312_L210026N.coml # CC1312_L2100238.coml # CPUwfi() at cpu.h:215 0x1f134	1											
Locate the file or edit the source lookup path to include its location.												- 8 - 7
Locate the file or edit the source lookup path to include its location.	A Basic )	Module Rea	dyQs R	24								. 8
Locate the file or edit the source lookup path to include its location.	A Basic ) address	Module Rea label state	dyQs R priority	ew fun argi	0 ang1 in	itligger our	ligger					
Loads the files or will the sources lookup path to include its location.	A Basic ) address 0x200	Module Rea label state idle	dyQs R priority 4	fin arg	Darg1 in Ox0 0	0	rigger					
Loots the file or eff the source lookap part to include its location.	Basic     J     address     0x200     0x200	Module Rea label state idle idle	dyQs R priority 4 5	tin arg 5_ 0x0 A_ 0x2	0 arg1 in 0x0 0 0x0 0	0	rigger					
Locate the file or edit the source lookup park to include its location. • Enget Configurations,  # ISOS Object View (ILOV)    # Ferrolits # Genome ISS 4_CITAL Cont • # Objects = EOS	Basic     address     0x200     0x200     0x200	Module Rea label state idle idle idle	dyQs R priority 4 5 0	fin arg 5, 0x0 A 0x2 R 0x0	0 arg1 in 0x0 0 0x0 0 0x0 0	0	ligger					
Loods the file or all the source looking path to include its location.  Exergit Configuration, IREOS Colject View (800) II  O Fanotine  O Fanotine  O Solid	A Basic address 0x200 0x200 0x200 0x200	Module Rea label state idle idle idle idle	dyQs R priority 4 5 0 0	fin arg 5 0x0 A_ 0x2 R_ 0x0 R_ 0x0	0 arg1 in 0x0 0 0x0 0 0x0 0 0x0 0	0 0 0 0	1939F					a <b>n</b> e
Look the file or all the score looked path to include its location.  Exergt Configurations: BICOS Object View (BOY) =  Exergt Configuration: BICOS Object View (BOY) =  Exercise  Exercise Exercise Exercise  Exercise Exer	Basic     address     0x200     0x200     0x200	Module Rea label state idle idle idle	dyQs R priority 4 5 0 0	fin arg 5 0x0 A_ 0x2 R_ 0x0 R_ 0x0	0 arg1 in 0x0 0 0x0 0 0x0 0	0 0 0 0	1939#					a <b>a</b> e
Loods the file or will the source looking path to include its location.  Exergit Configuration, III IDD Colgars View (800) II  P Browlin  P Bro	A Basic address 0x200 0x200 0x200 0x200	Module Rea label state idle idle idle idle	dyQs R priority 4 5 0 0	fin arg 5 0x0 A_ 0x2 R_ 0x0 R_ 0x0	0 arg1 in 0x0 0 0x0 0 0x0 0 0x0 0	0 0 0 0	Ngger					
Loost the file or ell the source looking path is include its location.  Expert Configuration, IRITOS Conject Vee (802) II  Foreitm  BITOS  BITOS BITOS  BITOS	A Basic address 0x200 0x200 0x200 0x200	Module Rea label state idle idle idle idle	dyQs R priority 4 5 0 0	fin arg 5 0x0 A_ 0x2 R_ 0x0 R_ 0x0	0 arg1 in 0x0 0 0x0 0 0x0 0 0x0 0	0 0 0 0	ligger					
Loost the file or all the score looked to house 5 Exerct Configuration; IRCOS Object View (BCV) = IRCOS Object View (BCV) = IR	A Basic address 0x200 0x200 0x200 0x200	Module Rea label state idle idle idle idle	dyQs R priority 4 5 0 0	fin arg 5 0x0 A_ 0x2 R_ 0x0 R_ 0x0	0 arg1 in 0x0 0 0x0 0 0x0 0 0x0 0	0 0 0 0	*00*					
Looot the file or edit the source looking path to include its location.  Exerget Configurations, IRROS Conject Wee (ROO) II  Exerget Configurations, IRROS Conject Wee (ROO) II  Exerce International III (IIII) (IIIII) (IIIIII) (IIIIII) (IIIIII) (IIIIII) (IIIIII) (IIIIIII) (IIIIII) (IIIIII) (IIIIII) (IIIIII) (IIIIII) (IIIIII) (IIIIIII) (IIIIIIII	A Basic address 0x200 0x200 0x200 0x200	Module Rea label state idle idle idle idle	dyQs R priority 4 5 0 0	fin arg 5 0x0 A_ 0x2 R_ 0x0 R_ 0x0	0 arg1 in 0x0 0 0x0 0 0x0 0 0x0 0	0 0 0 0	1934 					
Loost the file or all the score looked to looked its looked. Elegatic Configuration; MICOS Colject Week (800) = 2 Biological (51,5,4,5,12,2,2,0,1 2 Biological (51,5,4,5,12,2,2,0,1 2 Biological (51,5,4,12,2,0,1) 2 Biological (51,5,4,12,2,0,1) 2 Biological (51,5,4,12,2,0,1) 2 Biological (51,5,4,12,2,0,1) 2 Biological (51,5,4,12,2,0,1) 2 Biological (51,5,4,12,2,0,1) 2 Biological (51,5,4,12,12,0,1) 2 Biological (51,5,4,12,12,12,12,12,12,12,12,12,12,12,12,12,	A Basic address 0x200 0x200 0x200 0x200	Module Rea label state idle idle idle idle	dyQs R priority 4 5 0 0	fin arg 5 0x0 A_ 0x2 R_ 0x0 R_ 0x0	0 arg1 in 0x0 0 0x0 0 0x0 0 0x0 0	0 0 0 0	Ngger					
Loods the file or edit the source looking path to include its location. Exerget Configuration, IREOS Colpact Wee (800) 11    Exerget Configuration, IREOS Colpact Wee (800) 11	A Basic address 0x200 0x200 0x200 0x200	Module Rea label state idle idle idle idle	dyQs R priority 4 5 0 0	fin arg 5 0x0 A_ 0x2 R_ 0x0 R_ 0x0	0 arg1 in 0x0 0 0x0 0 0x0 0 0x0 0	0 0 0 0	ligger					
Loost the file or ell the score looked path is include its location.  Elegat Configuration; MICOS Object Wee (800) =  Elegat C	A Basic address 0x200 0x200 0x200 0x200	Module Rea label state idle idle idle idle	dyQs R priority 4 5 0 0	fin arg 5 0x0 A_ 0x2 R_ 0x0 R_ 0x0	0 arg1 in 0x0 0 0x0 0 0x0 0 0x0 0	0 0 0 0	Ngar					
Loost the file or all the score looked to house its Eleget Configuration: #EDS Object Vere (BOV) = #Elevent Configuration: #EDS Object Vere (BOV) = #Elevent Vere (BOV) = #Elevent Vere (BOV) = *Elevent	A Basic address 0x200 0x200 0x200 0x200	Module Rea label state idle idle idle idle	dyQs R priority 4 5 0 0	fin arg 5 0x0 A_ 0x2 R_ 0x0 R_ 0x0	0 arg1 in 0x0 0 0x0 0 0x0 0 0x0 0	0 0 0 0	ligger					
* 28 Groupset, 15.5, cct12.2ad * 805 * 805 * Cluck *	A Basic address 0x200 0x200 0x200 0x200	Module Rea label state idle idle idle idle	dyQs R priority 4 5 0 0	fin arg 5 0x0 A_ 0x2 R_ 0x0 R_ 0x0	0 arg1 in 0x0 0 0x0 0 0x0 0 0x0 0	0 0 0 0						
Look the file or all the sacra basis path is include its footbox Elegation (Englishing) = 2 Basis (Straget Versel (Straget	A Basic address 0x200 0x200 0x200 0x200	Module Rea label state idle idle idle idle	dyQs R priority 4 5 0 0	fin arg 5 0x0 A_ 0x2 R_ 0x0 R_ 0x0	0 arg1 in 0x0 0 0x0 0 0x0 0 0x0 0	0 0 0 0	Ngger					
Loods the file of all the score biology path is include its location.  Elegatic Configuration, IIII.000 Colpect Vere (IDO) II  Elegatic Colpect Vere (IDO) III  Elegatic Colpect Vere (IDO) III  Elegation  Elega	A Basic address 0x200 0x200 0x200 0x200	Module Rea label state idle idle idle idle	dyQs R priority 4 5 0 0	fin arg 5 0x0 A_ 0x2 R_ 0x0 R_ 0x0	0 arg1 in 0x0 0 0x0 0 0x0 0 0x0 0	0 0 0 0						
Loods the file or all the souro looked to houses Elegationage and the souro looked to house the all filesoften to the souro looked to the source of the source all houses the source of the source of the source of the source all houses the source of th	A Basic address 0x200 0x200 0x200 0x200	Module Rea label state idle idle idle idle	dyQs R priority 4 5 0 0	fin arg 5 0x0 A_ 0x2 R_ 0x0 R_ 0x0	0 arg1 in 0x0 0 0x0 0 0x0 0 0x0 0	0 0 0 0	Ngger					
Look the file or all the score looked path is include its footbis.  Eleged Configurations: #EDDS Object Vere (BDD) is  Eleged Configurations: #EDDS Object Vere	A Basic address 0x200 0x200 0x200 0x200	Module Rea label state idle idle idle idle	dyQs R priority 4 5 0 0	fin arg 5 0x0 A_ 0x2 R_ 0x0 R_ 0x0	0 arg1 in 0x0 0 0x0 0 0x0 0 0x0 0	0 0 0 0						
Locan the file or all the score locket put his include his location. Lengert Configurations, IRIOS Object Wee (BD7) = 2 Biospace, 19, 15, 4, ct 13-2, and 2 Biospace, 19, 15, 15, 15, 15, 15, 15, 15, 15, 15, 15	A Basic address 0x200 0x200 0x200 0x200	Module Rea label state idle idle idle idle	dyQs R priority 4 5 0 0	fin arg 5 0x0 A_ 0x2 R_ 0x0 R_ 0x0	0 arg1 in 0x0 0 0x0 0 0x0 0 0x0 0	0 0 0 0						
Locard the for well the score locked put his include its function.  Lengt Comparison # 8005 Object Week (8005 1)  Provide 1 (1990 1)  Provide 1 (1	A Basic address 0x200 0x200 0x200 0x200	Module Rea label state idle idle idle idle	dyQs R priority 4 5 0 0	fin arg 5 0x0 A_ 0x2 R_ 0x0 R_ 0x0	0 arg1 in 0x0 0 0x0 0 0x0 0 0x0 0	0 0 0 0						
Locan the file or all the score locket put his include his location. Lengert Configurations, IRIOS Object Wee (BD7) = 2 Biospace, 19, 15, 4, ct 13-2, and 2 Biospace, 19, 15, 15, 15, 15, 15, 15, 15, 15, 15, 15	A Basic address 0x200 0x200 0x200 0x200	Module Rea label state idle idle idle idle	dyQs R priority 4 5 0 0	fin arg 5 0x0 A_ 0x2 R_ 0x0 R_ 0x0	0 arg1 in 0x0 0 0x0 0 0x0 0 0x0 0	0 0 0 0						

#### Hardware, Software, Testing Requirements, and Test Results

#### 3.2 Testing and Results

#### 3.2.1 **Test Setup**

Table 9 summarizes the CCS build configuration to set up the end nodes and data collector. For these experiments, the end nodes and the data collector use the CC1312R LaunchPad as shown in Section 3.1.1. The pre-built binaries are provided in the reference design software package.

DEVICE	BUILD CONFIGURATION	NOTE
END NODE	<ul><li> debug_poll</li><li> debug_poll_dtls</li></ul>	_dtls configuration was used for DTLS verification shown in Section 3.2
DATA COLLECTOR	<ul><li>debug_root_poll</li><li>debug_root_poll_dtls</li></ul>	

#### Table 9. Firmware for Test Setup

#### 3.2.1.1 Creating Multi-hop Topology

It is challenging to create multi-hop networks in a small LAB area. For the experiments, the multi-hop topology was created with address filtering in the software. Each node has a list of entries where the node can accept the packet reception. To create a linear multi-hop topology, each node maintains two entries: one is the target parent and one is the target child.

```
#define NUM_ENTRIES 2 //Address filter to create multi-hop topology
//entry #0: target parent, entry #1: target child
//{0xBB, 0x63, 0xA4, 0x13, 0x00, 0x4B, 0x12, 0x00} (hop-1)
uint8_t whitelist[NUM_ENTRIES][8]={
{0x91, 0xC7, 0x27, 0x0A, 0x00, 0x4B, 0x12, 0x00},
{0x70, 0x63, 0xA4, 0x13, 0x00, 0x4B, 0x12, 0x00} };
```

In the software, this feature can be enabled with the "CONFIG MULTIHOP TESTING" macro defined in /Application/subg/config.h.

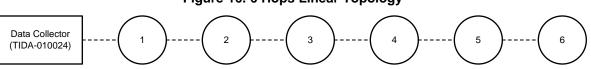
**NOTE:** For testing setup, modify the list of entries defined in /Application/middleware.c based on the extended address of the test node. For each node, the extended address set at the initial stage is stored in the global variable of "ApiMac\_extAddr".

The multi-hop network with the address filtering technique has pros and cons in terms of network performance because a node can communicate with a dedicated set of nodes only while all of the nodes can be seen with each other. A disadvantage is that a node limits the path diversity over multi-hop topology while an advantage is that this setup can avoid hidden node problems.

#### 3.2.2 Improved Security Performance

The goal of this experiment is to validate the security functionality implemented in the TIDA-010024 reference design. With DTLS, the TIDA-010024 offers a secured transmission mechanism of using a different private key per link.

To verify if the encryption works correctly, a packet sniffer was used to read the encrypted messages. To set up the packet sniffer, follow the instructions described in Section 3.1.2.2.3.2. This experiment was performed with the UDP poll example and uses over a 6-hop linear topology as shown Figure 16.



## Figure 16. 6 Hops Linear Topology



Hardware, Software, Testing Requirements, and Test Results

To test the encryption, a message is sent from node 6 to the data collector by passing through every node. The sniffer shows the data transfer between each of the nodes. At each hop, the message is decrypted and then re-encrypted with the private key shared for each link-level connection. Figure 17 shows the same message captured at different hops.



<pre>p /rmss 1946: 288 ptms owire (1948 bits), 288 bytes (captured (1948 bits)) on inter- bitement Frequent Vision 4, 9, 2013.08.13, 0, 001 1932.08.01.3 ) 10 miles frequent (1948 bits), 10 miles (19</pre>	) Free 199: 28 pts on vire (198 bits), 28 pts captured (198 bits) on inter Distance FreeDool Version 4, 5: CD 201.84.8.1.3, 0: VII 30: 48.1.3 ) un bringen Freiden, 3: CF PHT. 1796, DE HUT. 1776 ) UN Bringen Freiden, 3: CF PHT. 1796, DE HUT. 1776 ) TE BUJJAGE DEN PHY LICEN CARE DE NICH Y TE BUJJAGE DEN PHY LICEN CARE DE NICH Y TE BUJJAGE DEN, DET HUT. 1890 IE HITZICK, SFC: WIIIAHOWIER BID Security Field, 2018, DE HUT. 2018, DE HUT. 2018, Security Monla, Activating Re Sequence Number: 201 Press Control Historic West (2018), 4: CONTROL (2018), DE HUT. Destination: Teaching, WIIIAHITZICK, (WIIIAHOWIER BID Destination: Teaching, WIIIAHITZICK (WIIIAHITZICK)) V Explorition: EXPL D BID Destination: Teaching, WIIIAHITZICK (WIIIAHITZICK), BENCHY 2 BID TE (2015) DET (2015)	<pre>press 1341.218 bytes on vire (1988 bits), 228 bytes captured (1984 bits) on inter- interver frequency forces (1984 bits), 201 bits(1984 bits) ) the set of the s</pre>
C >> 00000 45 00 00 00 00 00 10 10 45 00 45 01 85 E	C >>	C >> 0000 45 00 00 e0 00 00 00 10 15 68 01 05 E
$ \begin{array}{c} \begin{array}{c} 1 \\ 1 \\ 1 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\$	$ \begin{array}{c} 1213 & 0 & 0 & 0 & 44 & 0 & 44 & 0 & 0 & 0 &$	$ \begin{array}{c} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\$

In Figure 17, the same data is sent between neighbors, but the value intercepted by the sniffer is different because the same data is encrypted with different keys. This mechanism makes this network more robust against tampering.

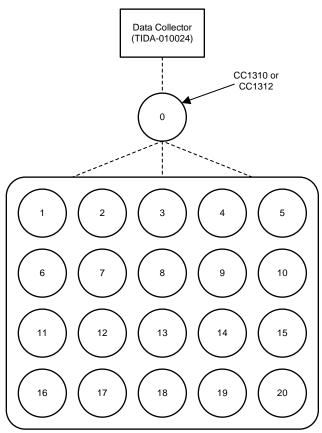
#### 3.2.3 Impact of Network Capacity on Performance: An Example of the Bottleneck Scenario

The goal of this experiment is to validate the impact of increased routing capacity on the network performance in a bottleneck scenario. Compared to CC1310, the CC1312R has more memory to support bigger neighbor and routing entries.

For this experiment, the UDP poll example is used over a bottleneck scenario where all the data goes through a single router (node 0) as shown in Figure 18. A data collector sends a 100B poll message every 5 seconds and, as a response, each node sends back the same size data packet to the data collector.



# Figure 18. A Bottleneck Scenario



This experiment was done with the CC1310 and the CC1312 at the bottleneck location of node 0. With the CC1312R, the data collector can reach at every node without losing connections due to the increased capacity.

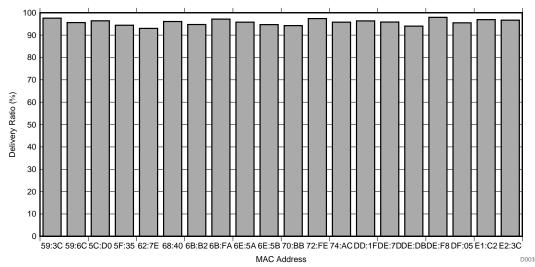


Figure 19. Delivery Ratio Performance With CC1312R Over a Bottleneck Scenario

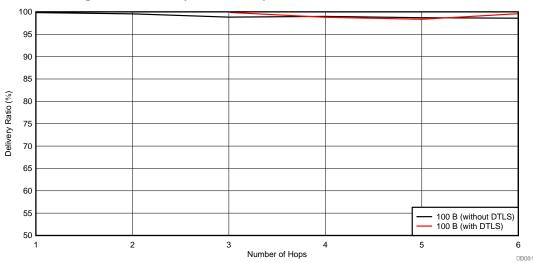
Figure 19 shows delivery ratio performance with CC1312R over the bottleneck scenario. The x-axis shows MAC address of each node and the y-axis is delivery ratio performance in percentage. The experimental result shows that the average delivery ratio achieves 95.75%.



The same experiment was performed by replacing CC1312R with CC1310 for node 0. The result shows that, as expected, with CC1310 the overall network performance is degraded due to network instability caused by the limited capacity.

#### 3.2.4 Impact of DTLS Security on Network Performance

It is important to ensure that the DTLS overhead does not degrade the network performance. To validate the impact of overhead on system performance, the round-trip time and delivery ratio were measured with and without DTLS over a 6-hop linear topology.



#### Figure 20. Delivery Ratio Comparison With and Without DTLS

Figure 20 shows the delivery ratio performance with and without DTLS. The experiment ran for a day. The experimental result shows that both achieve similar delivery ratio performance regardless of the number of hops. Without DTLS, the average delivery ratio achieves 99.08% and with DTLS, the delivery ratio is 99.42%.

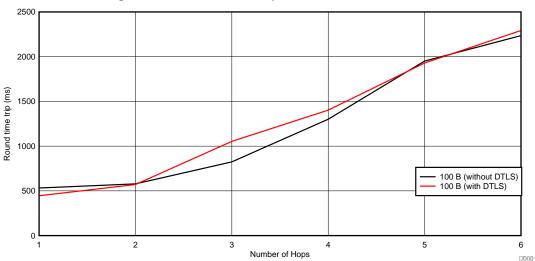


Figure 21. Round Time Trip With and Without DTLS

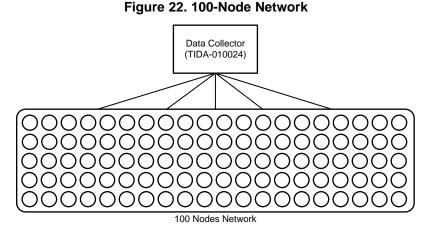
In Figure 21, with the same experiment, the average round-trip time was measured. For a six-hop distance, the round-trip time performance shows 2.294 seconds with DTLS and 2.236 seconds without DTLS.



In conclusion, the experimental results show that the impact of the DTLS overhead on network performance is negligible. This is because the simplified 2-way DTLS handshaking mechanism reduces the impact of the DTLS overhead on network performance and the transaction happens one time at the beginning.

#### 3.2.5 100-Node Network Testing

The goal of this experiment is to measure delivery ratio performance as well as to verify the software reliability in a large network. For the experiment, the UDP poll with DTLS example was used. The data collector polls each node every 10 seconds with 100B of data. Figure 22 shows the topology of the network used for this experiment.



This experiment runs for 3 days to collect enough data. A CC1312R LaunchPad was used as data collector and the 100 LaunchPads shown in Figure 23 are 50 CC1312R LaunchPads and 50 CC1352R LaunchPads.



Figure 23. 100-Node Network Test Setup

Figure 24 shows the delivery ratio performance as a function of the MAC address of each node in the 100node setup. The x-axis shows the MAC address of each node and the y-axis shows the delivery ratio in percentage. The experimental result shows that all of the nodes achieve good delivery ratio performance. The average delivery ratio with 100 nodes is 97.14%.



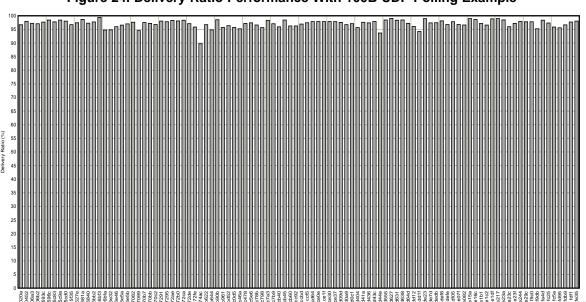


Figure 24. Delivery Ratio Performance With 100B UDP Polling Example

Table 10 summarizes test cases to verify the software reliability and self-healing feature with the 100-node network setup. The experiments use the UDP polling example with 100B of data. The experimental results show that the 6LoWPAN mesh software works reliably without losing connections in the 100-node setup, and the keep-alive mechanism implemented in the software is functional by detecting and recovering connection losses.

TEST CASE	DESCRIPTION	PASS or FAIL
Long-run data testing (> 3 days)	Run 100B UDP polling example in the 100-node setup	Pass (All 100 nodes stayed connection to the data collector)
Self-healing testing (data collector failure)	Power-cycle the data collector to see if all the 100 nodes are reconnected to the network.	Pass
Self-healing testing (end-node failure)	Power-cycle some end-nodes to see if all the 100 nodes are reconnected to the network.	Pass

#### Table 10. Software Reliability Test Summary



# 4 Design Files

# 4.1 Schematics

To download the schematics, see the design files at TIDA-010024.

# 4.2 Bill of Materials

To download the bill of materials (BOM), see the design files at TIDA-010024.

# 4.3 PCB Layout Recommendations

## 4.3.1 Layout Prints

To download the layer plots, see the design files at TIDA-010024.

## 4.4 Altium Project

To download the Altium Designer® project files, see the design files at TIDA-010024.

# 4.5 Gerber Files

To download the Gerber files, see the design files at TIDA-010024.

## 4.6 Assembly Drawings

To download the assembly drawings, see the design files at TIDA-010024.

# 5 Software Files

To download the software files, see the design files at TIDA-010024.



#### Related Documentation

#### 6 Related Documentation

- 1. Texas Instruments, Simple 6LoWPAN Mesh End-Node Improves Network Performance Reference Design
- 2. Texas Instruments, Simple 6LoWPAN Mesh Data Collector Improves Network Performance Reference Design
- Texas Instruments, Universal data concentrator reference design supporting Ethernet, 6LoWPAN RF mesh and more
- 4. Texas Instruments, SmartRF Packet Sniffer 2 v1.5.0 User's Guide

#### 6.1 Trademarks

SimpleLink, E2E, Code Composer Studio are trademarks of Texas Instruments. Altium Designer is a registered trademark of Altium LLC or its affiliated companies. Arm, Cortex are registered trademarks of Arm Limited. Wireshark is a trademark of Wireshark Foundation, Inc. All other trademarks are the property of their respective owners.

#### 6.2 Third-Party Products Disclaimer

TI'S PUBLICATION OF INFORMATION REGARDING THIRD-PARTY PRODUCTS OR SERVICES DOES NOT CONSTITUTE AN ENDORSEMENT REGARDING THE SUITABILITY OF SUCH PRODUCTS OR SERVICES OR A WARRANTY, REPRESENTATION OR ENDORSEMENT OF SUCH PRODUCTS OR SERVICES, EITHER ALONE OR IN COMBINATION WITH ANY TI PRODUCT OR SERVICE.

#### 7 About the Author

**WONSOO KIM** is a system engineer at Texas Instruments, where he is responsible for driving grid communication system solutions, defining future requirements in TI product roadmap, and providing system-level support and training focusing on communication systems for Smart Grid customers. He received a Ph.D. degree in electrical and computer engineering from the University of Texas, Austin, Texas.

**MICKAEL CHOUTEAU** is a field application engineer at Texas Instruments, where he is responsible for technical support for industrial customers. His design experience covers both hardware design (signal chain, battery monitoring) and software design (low level drivers, RTOS, and RF protocols). He received a master degree in embedded system engineering from ECE Paris University in France.



# **Revision History**

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

Cł	nanges from Original (October 2018) to A Revision	Page
•	Added list item to Features section	1
•	Deleted street lighting from Applications section	1
•	Added water meter and gas meter to Applications section	1
•	Changed information regarding what is supported by TI SimpleLink PHY	2
•	Changed information and values in the System Features of TIDA-010024 and TIDA-010003 table	
•	Changed the TIDA-010024 System Architecture image	3
•	Changed information and values in the Key System Specifications table	4
•	Changed CC1312 to CC1312R throughout document	5
•	Deleted information regarding the CC1352EVM from the Hardware section	8
•	Changed title of the Build Configurations of TIDA-010024 table to CCS Build Configurations	
•	Changed information in the CCS Build Configurations table	9
•	Changed Code Composer Studio XDC Tool and SDK Versions image	. 11
•	Changed title of the 6LoWPAN_TI_15_4_Example section to 6LoWPAN Mesh TI 15.4 Example	. 12
•	Changed title of the DTLS Data Encryption section to DTLS Encryption	. 14
•	Changed information in the DTLS Encryption section	. 14
•	Changed sensor_cc13x2r1lp to sensor_cc1312r1lp	. 15
•	Changed SimpleLink CC13x2 SDK v2.10.00.48 to SimpleLink CC13x2 and CC26x2 SDK v2.40	. 15
•	Changed information in the Firmware for Test Setup table	. 21
•	Changed title of Creating Multi-hop Linear Topology section to Creating Multi-hop Topology	. 21
•	Changed information in the Improved Security Performance section	. 21
•	Changed title of the Improved Security Performance section to Impact of Network Capacity on Performance: An Exan of the Bottleneck Scenario	nple . <mark>22</mark>
•	Changed information in the Impact of Network Capacity on Performance: An Example of the Bottleneck Scenario section	. 22
•	Changed title of the Bottleneck Scenario With CC1312 Delivery Ratio image to Delivery Ratio Performance With CC1312R Over a Bottleneck Scenario	. 23
•	Changed information in the Impact of DTLS Security on Network Performance section	. 24
•	Changed information in the 100-Node Network Testing section	. 25
•	Changed title of the 100 Nodes Network image to 100-Node Network	. 25
•	Changed title of the 100 Nodes Network Test Setup image to 100-Node Network Test Setup	. 25
•	Added Wonsoo Kim to About the Author section	. 28

# IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES "AS IS" AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI's products are provided subject to TI's Terms of Sale or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI's provision of these resources does not expand or otherwise alter TI's applicable warranties or warranty disclaimers for TI products.

TI objects to and rejects any additional or different terms you may have proposed.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265 Copyright © 2022, Texas Instruments Incorporated