TI Designs: TIDA-010032

Universal Data Concentrator Reference Design Supporting Ethernet, 6LoWPAN RF Mesh and More

Description

IPv6-based grid communications are becoming the standard choice in industrial markets and applications like smart meters and grid automation. The universal data concentrator design provides a complete IPv6-based network solution integrated with Ethernet backbone communication, 6LoWPAN RF mesh networking, RS-485 and more. The 6LoWPAN mesh networking addresses key concerns such as standard-based interoperability, reliability, security and long-distance connectivity. This design allows controlling and monitoring end devices remotely with a web server accessible via Ethernet backbone communication. It also provides 3.3-V and 5-V voltage rails and various peripheral interfaces to extend to additional connectivity such as broadband power-line communication (PLC), cellular and Wi-Fi®.

Resources

TIDA-010032	Design Folder
AM3358	Product Folder
CC1312R	Product Folder
THVD1500	Product Folder
ISOW7841	Product Folder
TPS65217	Product Folder
TL5209	Product Folder
TPS796	Product Folder

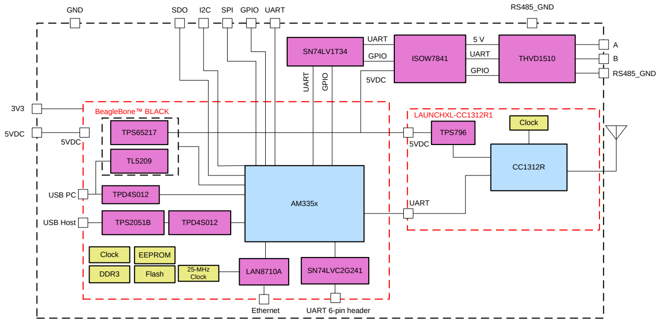
ASK Our E2E™ Experts

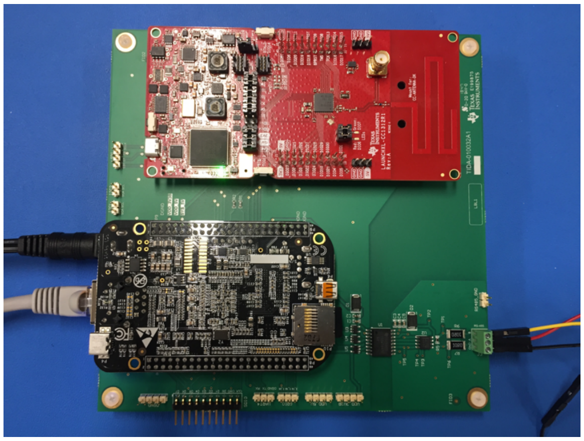
Features

- Implements universal data concentrator supporting Ethernet, 6LoWPAN mesh and RS-485 connectivity devices
- Capable of extending to other connectivity devices such as broadband PLC, cellular and Wi-Fi
- Allows Internet of Things (IoT) services with web server and Ethernet backbone connectivity
- IPv6 networking over low-power RF in the < 1-GHz ISM bands
- Implements 6LoWPAN RF mesh network protocols of 6LoWPAN, RPL, IPv6/ICMPv6and UDP
- Integrates with TI 15.4-Stack that supports frequency hopping (FH) and data encryption
- Fully compatible with the TIDA-010003 and TIDA-010024 end-node reference designs to provide a complete network solution

Applications

- Data Concentrators
- Wireless Communications





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

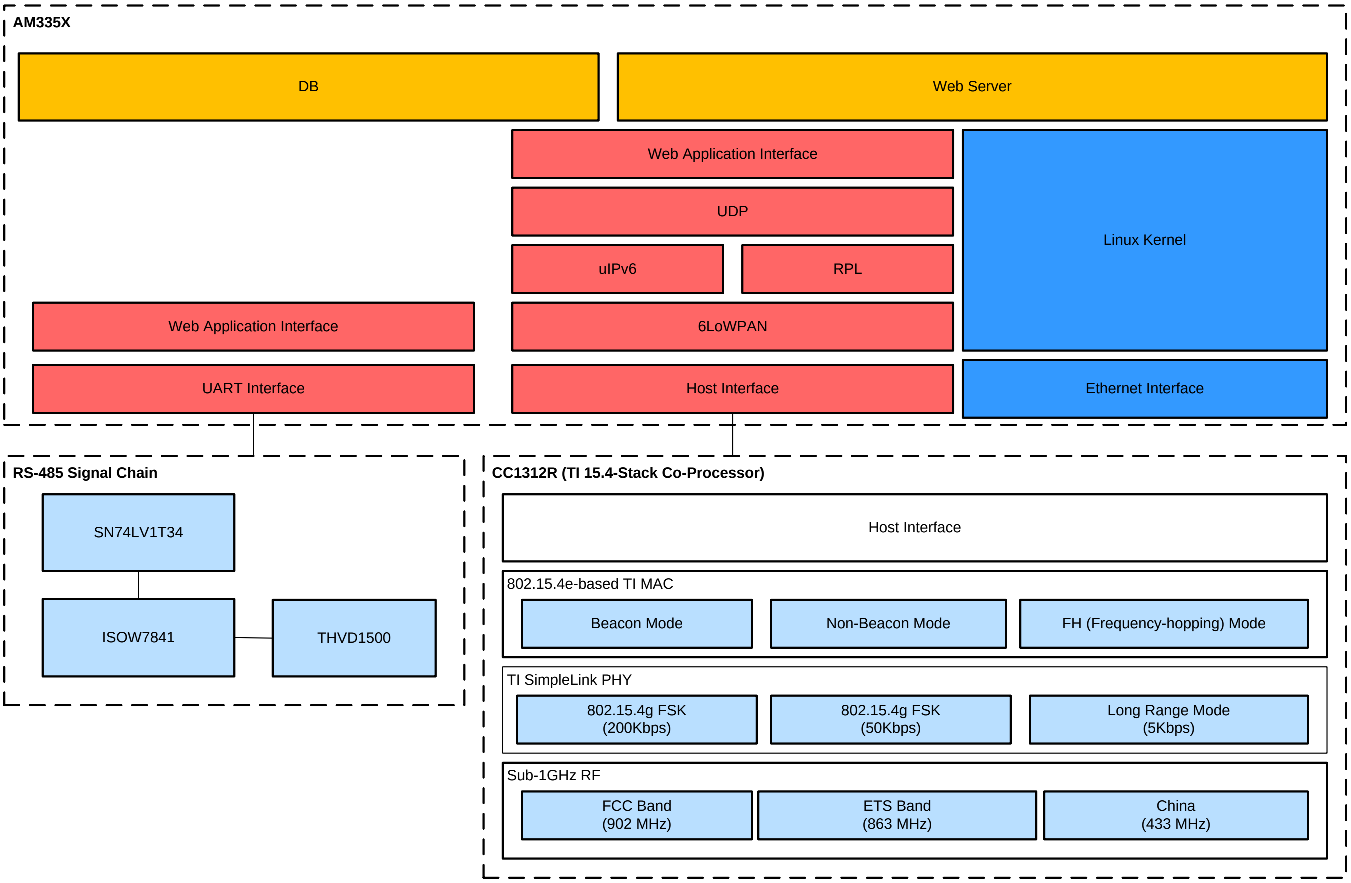
1 System Description

This reference design provides an open, IPv6 standard-based universal data concentrator design supporting various connectivity devices such as Ethernet, 6LoWPAN mesh networking, and RS-485.

Figure 1 shows the overall system architecture. This reference design provides a complete set of 6LoWPAN mesh stacks (6LoWPAN, RPL, IPv6, and UDP) integrated with TI 15.4-Stack, working with fully compatible 6LoWPAN mesh end-node designs of TIDA-010003 and TIDA-010024. The reference design improves advanced metering infrastructure (AMI) network performance by using frequency hopping (FH) techniques that increase robustness in noisy radio frequency (RF) environments.

This design provides a simple RS-485 gateway application connected to an isolated RS-485 transceiver via UART and GPIO interfaces, which can be a baseline software for developers to extend this design to support broadband PLC via UART. Furthermore, this design provides a working open-source based web server software that is connected to RS-485 and 6LoWPAN mesh applications via TCP sockets, which allows users to control and monitor end nodes connected to the data concentrator with different connectivity devices of 6LoWPAN mesh and RS-485. The Ethernet backbone connectivity allows IoT services by accessing the information from end nodes remotely.

Figure 1. TIDA-010032 System Architecture



1.1 Key System Specifications

Table 1. Key System Specifications

PARAMETER	SPECIFICATIONS	DETAILS
Connectivity Implemented	Ethernet, 6LoWPAN RF Mesh, and RS-485	
RS-485 Transceiver	250-kbps, half-duplex, RS-485 integrated with Bus I/O protection, 5-V supply power	Section 2.3.3
Digital Isolator	Integrated DC-DC converter with on-chip transformer, 3-V to 5.5-V wide input supply range, forward/reverse channels: 3/1	Section 2.3.4
6LoWPAN Mesh	UDP, IPv6, ICMPv6, RPL, and 6LoWPAN over TI 15.4-Stack based FH technique	
Ethernet	• 2-Port 10/100 PRU EMAC • 2-Port 1Gb Switch	Section 2.3.1
Peripheral Interfaces	UART, SPI, GPIO, USB, SDIO, and I^2C	Capable of supporting broadband PLC, cellular, and Wi-Fi

Table 2. 6LoWPAN Mesh System Specifications

PARAMETER	SPECIFICATIONS	DETAILS
Maximum number of route entries	1000	Section 3.2.1
Maximum number of neighbor nodes	500	
Maximum number of hops	64	
Maximum application data size	1576B	
Maximum FH neighbor entries (TI 15.4-Stack)	200	Section 3.1.2.2
MAC data encryption	• IEEE 802.15.4-based encryption • MIC-32, MIC-64, MIC-128 • ENC, ENC-MIC-32, ENC-MIC-64, and ENC-MIC-128 • MAC-level DTLS	Section 2.3.12
Delivery ratio	99.2%	Section 3.3.2

2 System Overview

2.1 Block Diagram

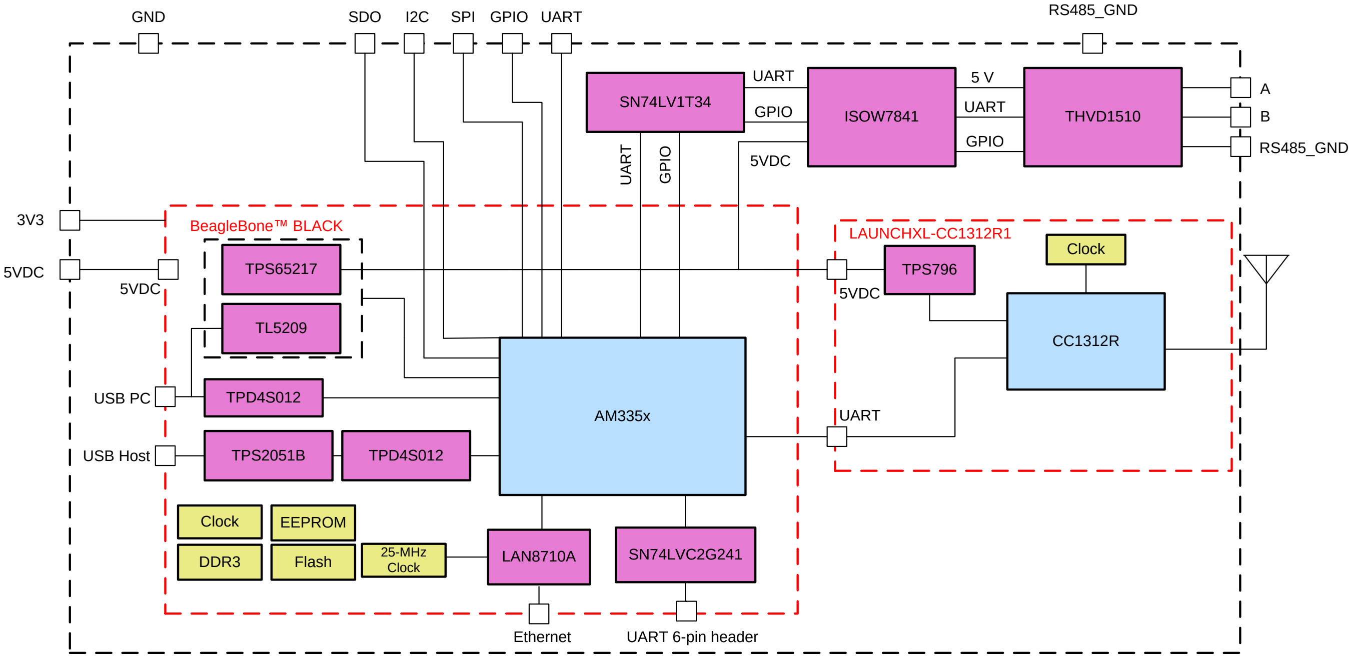
Figure 2 shows the system block diagram. The reference design is built with a TIDA-010032 docking board with two standard EVMs of BeagleBone Black (BBB) and LAUNCHXL-CC1312R1. The docking board includes an isolated RS-485 signal chain. With 5-V external DC power supply to the BBB, the BBB powers 5 V to the LAUNCHXL-CC1312R1 and embedded RS-485 transceiver. The AM3358 microprocessor unit (MPU) is the application processor running web server and gateway applications, connected to CC1312R and RS-485 transceiver via UARTs.

The CC1312R is the microcontroller unit (MCU) to run TI 15.4-Stack coprocessor over Sub-1 GHz RF. The TPS796, in the LAUNCHXL-CC1312R1, steps 5-V DC down to 3.3 V to power the CC1312R MCU.

The ISOW7841 isolates the RS-485 signal chain from AM335x MPU to reduce noise currents on a data bus and to protect damaging sensitive circuitry. The ISOW7841 is integrated with reinforced power converter to supply 5-V power to the RS-485 transceiver.

The THVD1500 is a cost optimized half-duplex RS-485 transceiver (up to 250 kbps) integrated with bus I/O protection. The THVD1500 operates from a single 5-V supply, which is provided by the ISOW7841. The level shifter (SN74LV1T34DBVR) installed in-between the ISOW7841 and AM335x adjusts the voltage level in the control (GPIO) and data (UART) paths to deliver the proper logical value to each device.

Figure 2. TIDA-010032 Block Diagram



2.2 Design Considerations

For this reference design, these devices perform the following:

- The AM335x microprocessors, based on the ARM® Cortex®-A8 processor, are enhanced with image, graphics processing, peripherals and industrial interface options.
- CC1312R wireless MCU combines an ARM Cortex®-M4 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
- THVD1500 is a cost-optimized half-duplex RS-485 transceiver integrated with bus I/O protection to satisfy ±8 kV IEC61000-4-2 contact discharge and ±2 kV IEC61000-4-4 Fast Transient Burst.
- ISOW7841 is a high-performance digital isolator with on-chip transformer and an integrated reinforced power converter up to 650 mW of isolated power.

- TPS65217x is a single-chip power management IC (PMIC) specifically designed to power the AM335x ARM Cortex-A8 processor in portable and 5-V line-powered applications.
- 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.

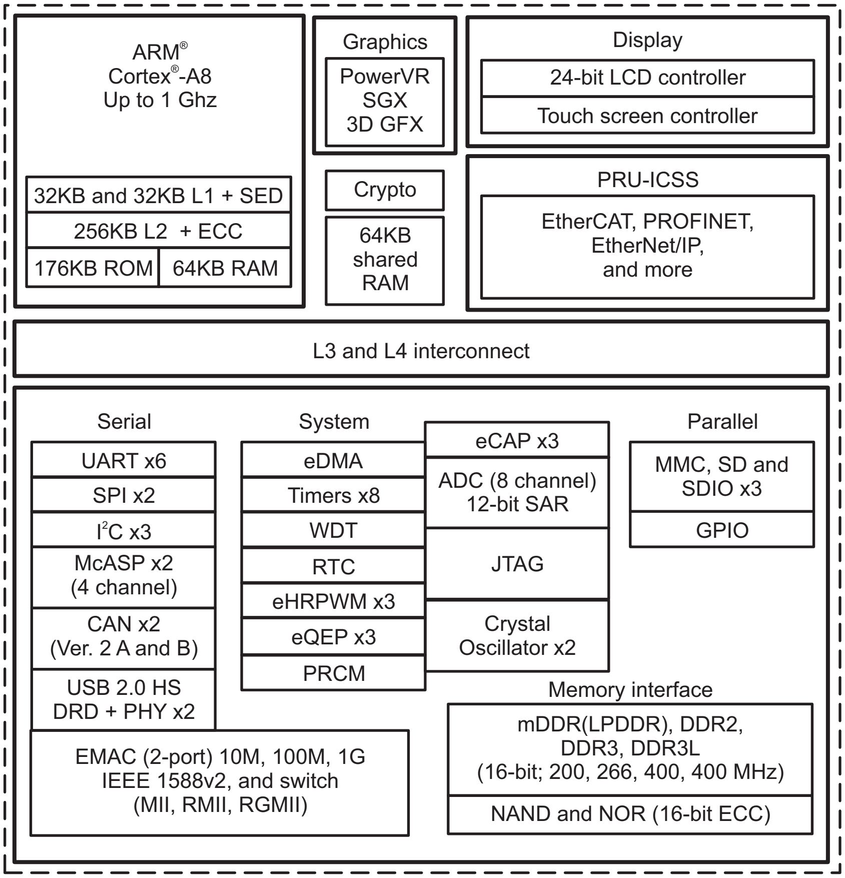
2.3 Highlighted Products

2.3.1 AM3358

The AM335x microprocessors, based on the ARM Cortex-A8 processor, are enhanced with image, graphics processing, peripherals and industrial interface options such as EtherCAT and PROFIBUS. The devices support high-level operating systems (HLOS). Linux® and Android™ are available free of charge from TI. The MPU subsystem is based on the ARM Cortex-A8 processor and the POWERVR® SGX Graphics Accelerator subsystem provides 3D graphics acceleration to support display and gaming effects. The PRU-ICSS is separate from the ARM core, allowing independent operation and clocking for greater efficiency and flexibility.

The PRU-ICSS enables additional peripheral interfaces and real-time protocols such as EtherCAT, PROFINET, EtherNet/IP, PROFIBUS, Ethernet Powerlink, Sercos, and others. Additionally, the programmable nature of the PRU-ICSS, along with its access to pins, events and all system-on-chip (SoC) resources, provides flexibility in implementing fast, real-time responses, specialized data handling operations, custom peripheral interfaces, and in offloading tasks from the other processor cores of SoC.

Figure 3. AM335x Functional Block Diagram



2.3.2 CC1312R

The CC1312R device is a Sub-1 GHz wireless MCU targeting Wireless M-Bus, IEEE 802.15.4g, IPv6-enabled smart objects (6LoWPAN), Zigbee®, KNX RF, Wi-SUN®, and proprietary systems.

The CC1312R device is a member of the CC26xx and CC13xx family of cost-effective, ultra-low power, 2.4-GHz and Sub-1 GHz RF devices. Very low active RF and MCU current, in addition to sub-µA sleep current with up to 80KB of RAM retention, provide excellent battery lifetime and allow operation on small coin-cell batteries and in energy-harvesting applications.

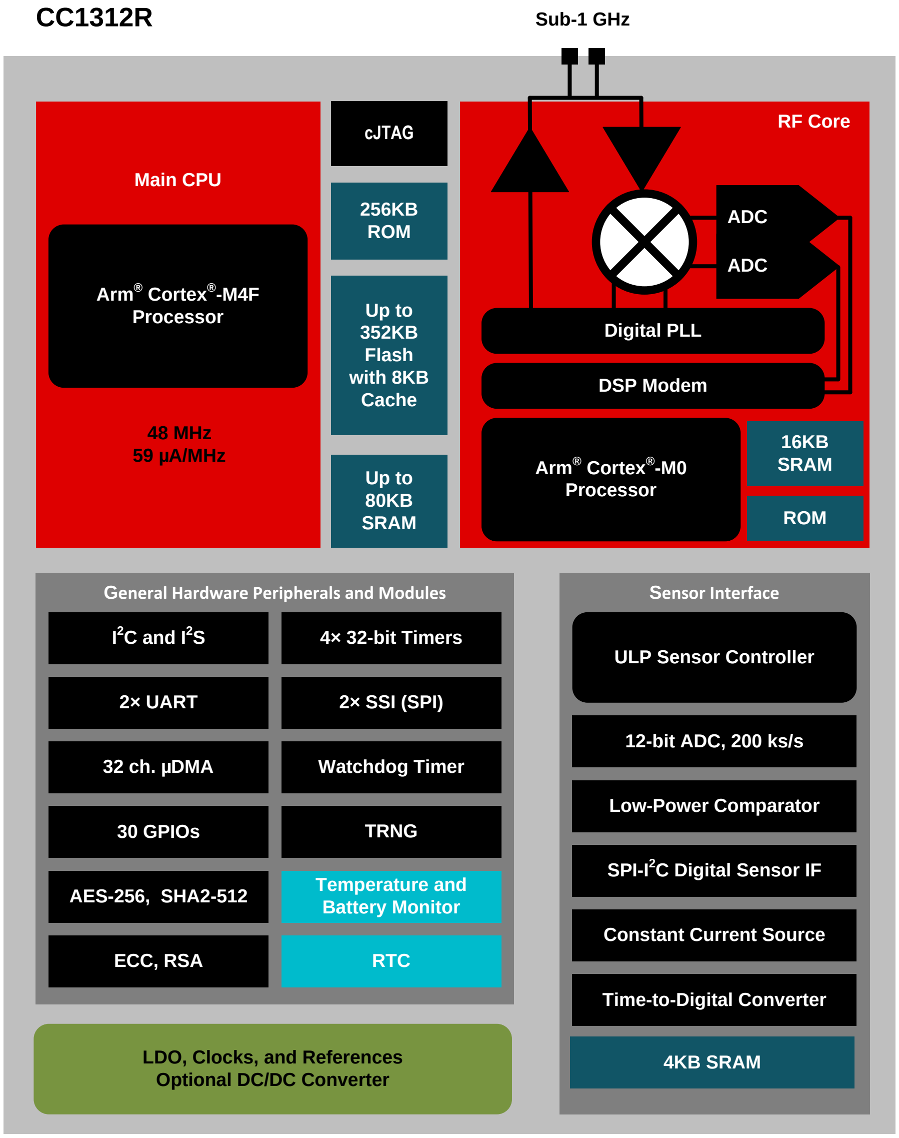
The CC1312R device combines a flexible, very low-power RF transceiver with a powerful 48-MHz ARM Cortex-M4F CPU in a platform supporting multiple physical layers and RF standards. A dedicated Radio Controller (ARM Cortex-M0) handles low-level RF protocol commands that are stored in ROM or RAM, thus ensuring ultra-low power and great flexibility. The low power consumption of the CC1312R device does not come at the expense of RF performance; the CC1312R device has excellent sensitivity and robustness (selectivity and blocking) performance.

The CC1312R device is a highly integrated, true single-chip solution incorporating a complete RF system and an on-chip DC/DC converter.

Sensors can be handled in a very low-power manner by a programmable, autonomous ultra-low power Sensor Controller CPU with 4KB of SRAM for program and data. The Sensor Controller, with its fast wake-up and ultra-low-power 2-MHz mode is designed for sampling, buffering, and processing both analog and digital sensor data; thus the MCU system can maximize sleep time and reduce active power.

The CC1312R device is part of the SimpleLink™ MCU platform, which consists of Wi-Fi, Bluetooth® low energy, Thread, Zigbee, Sub-1 GHz MCUs, and host MCUs, which all share a common, easy-to-use development environment with a single core SDK and rich tool set. A one-time integration of the SimpleLink platform enables you to add any combination of the portfolio's devices into your design, allowing 100 percent code reuse when your design requirements change. For more information, visit www.ti.com/simplelink.

Figure 4. CC1312R Functional Block Diagram



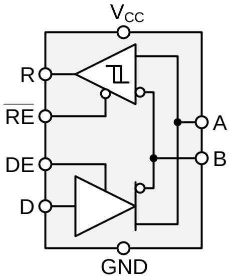
2.3.3 THVD1500

THVD1500 is a robust half-duplex RS-485 transceiver for industrial applications. The bus pins are immune to high levels of IEC Contact Discharge ESD events eliminating need of additional system level protection components.

The device operates from a single 5-V supply. The wide common-mode voltage range and low input leakage on bus pins make THVD1500 suitable for multi-point applications over long cable runs.

THVD1500 is available in industry standard 8-pin SOIC package for drop-in compatibility. The device is characterized from –40°C to 125°C.

Figure 5. THVD1500 Functional Block Diagram



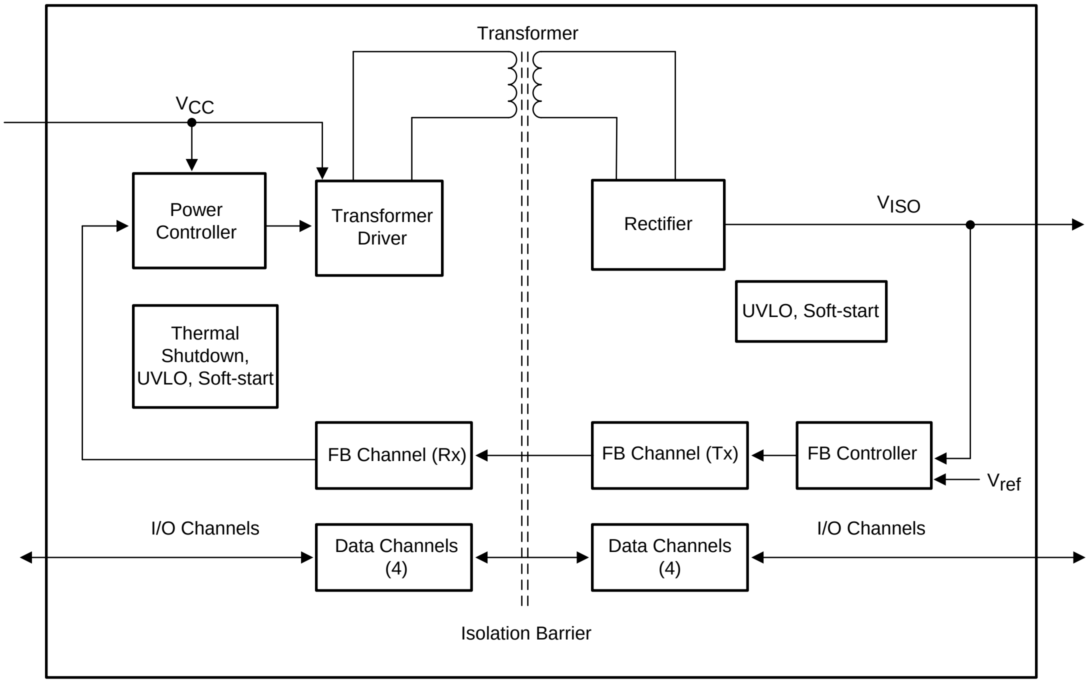
2.3.4 ISOW7841

The ISOW784x is a family of high-performance, quad-channel reinforced digital isolators with an integrated high-efficiency power converter. The integrated DC-DC converter provides up to 650 mW of isolated power at high efficiency and can be configured for various input and output voltage configurations. Therefore these devices eliminate the need for a separate isolated power supply in space-constrained isolated designs.

The ISOW784x family of devices provide high electromagnetic immunity and low emissions while isolating CMOS or LVCMOS digital I/Os. The signal-isolation channel has a logic input and output buffer separated by a silicon dioxide (SiO_2) insulation barrier, whereas, power isolation uses on-chip transformers separated by thin film polymer as insulating material. Various configurations of forward and reverse channels are available. If the input signal is lost, the default output is high for the ISOW784x devices and low for the devices with the F suffix.

These devices help prevent noise currents on a data bus or other circuits from entering the local ground and interfering with or damaging sensitive circuitry. Through innovative chip design and layout techniques, electromagnetic compatibility of the ISOW784x family of devices has been significantly enhanced to ease system-level ESD, EFT, surge and emissions compliance. The high-efficiency of the power converter allows operation at a higher ambient temperature. The ISOW784x family of devices is available in a 16-pin SOIC wide-body (SOIC-WB) DWE package.

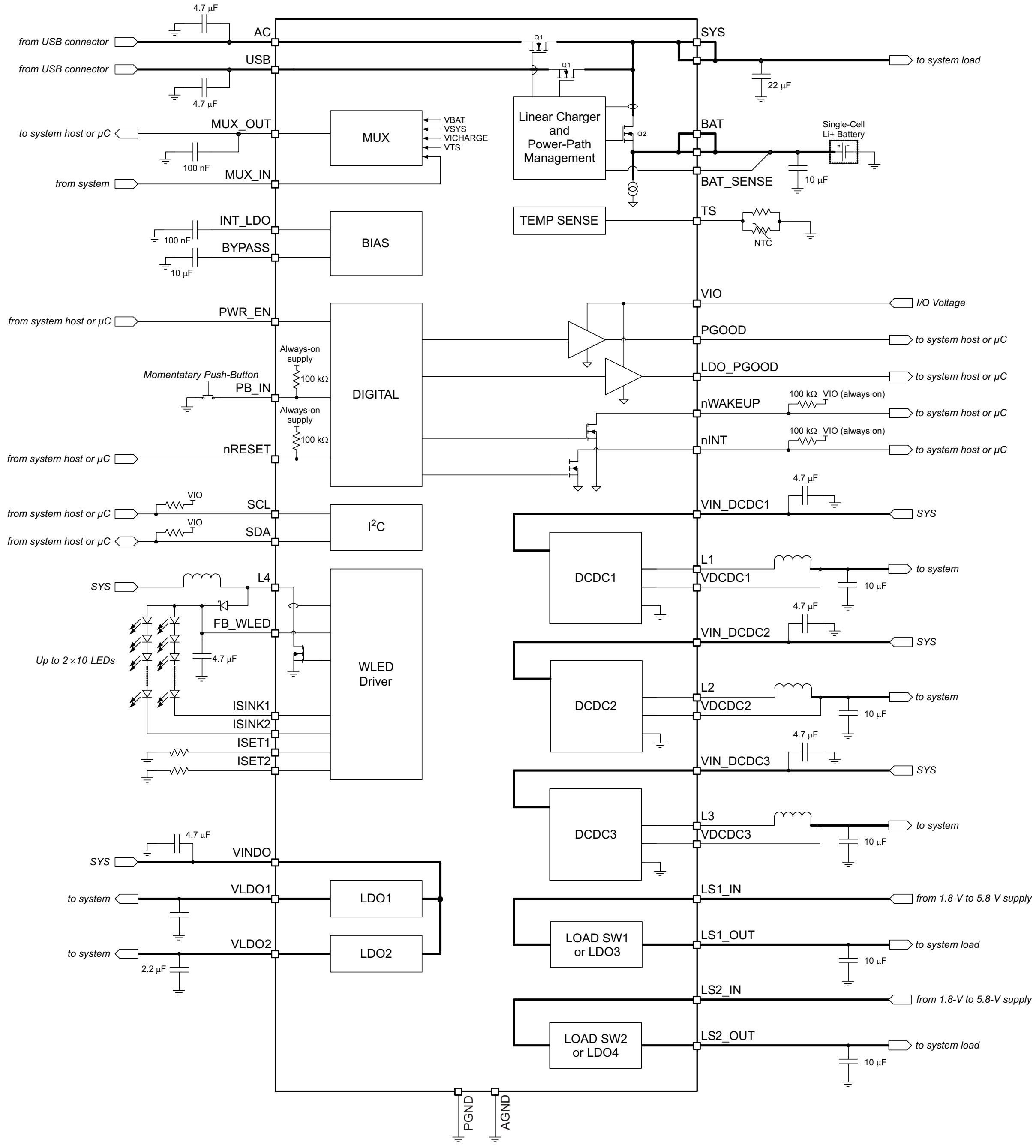
Figure 6. ISOW7841 Functional Block Diagram



2.3.5 TPS65217

The TPS65217x is a single-chip power management IC (PMIC) specifically designed to power the AM335x ARM Cortex-A8 processor in portable and 5-V line-powered applications. The PMIC device provides a linear battery charger for single-cell Li-ion and Li-polymer batteries, dual-input power path, three step-down converters, four low-dropout (LDO) regulators, and a high-efficiency boost converter to power two strings of up to 10 LEDs each. The system can be supplied by any combination of USB port, 5-V AC adaptor, or Li-Ion battery. The device is characterized across a –40°C to +105°C temperature range which makes it suitable for industrial applications. Three high-efficiency 2.25-MHz step-down converters can providing the core voltage, memory, and I/O voltage for a system. The TPS65217x device comes in a 48-pin leadless package (6-mm × 6-mm VQFN) with a 0.4-mm pitch.

Figure 7. TPS65217 Functional Block Diagram



2.3.6 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, 3×3 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 µA. The TPS79630 exhibits approximately 40 μV_{RMS} 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 8. Functional Block Diagram (Fixed Version)

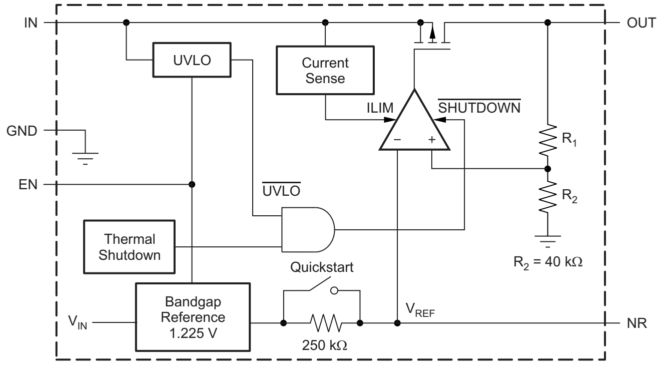
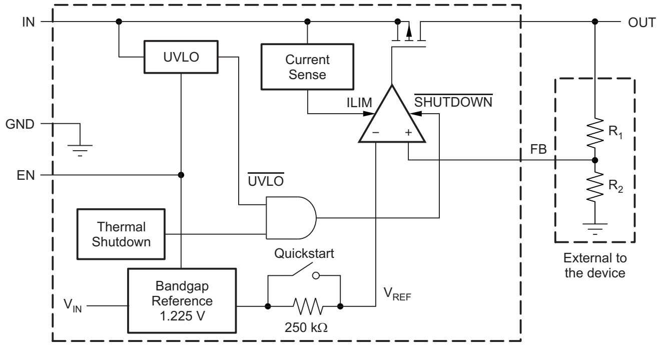


Figure 9. Functional Block Diagram (Adjustable Version)



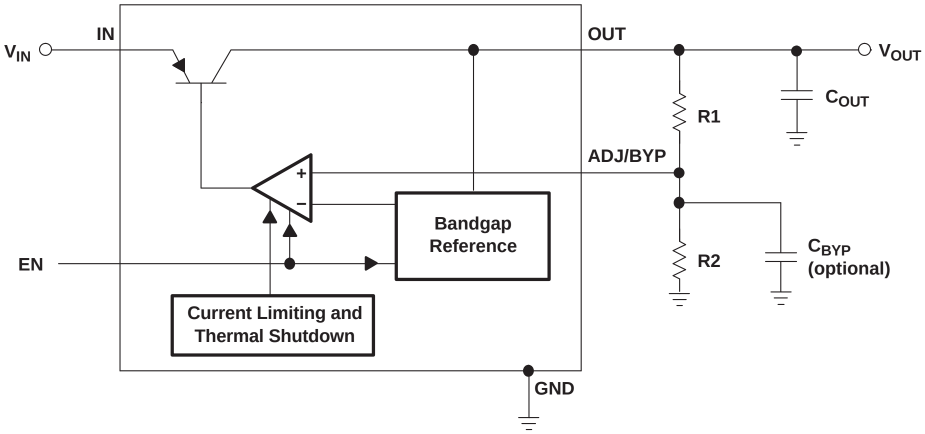
2.3.7 TL5209

The TL5209 device is 500-mA low-dropout (LDO) regulator that is well suited for portable applications. It has a lower quiescent current than most traditional PNP regulators and allows for a shutdown current of 0.05 µA (typical). The TL5209 also has very good dropout voltage characteristics, requiring a maximum dropout of 10 mV at light loads and 500 mV at full load. In addition, the LDO also has a 1% output voltage accuracy and very tight line and load regulation that is comparable to its CMOS counterparts.

For noise-sensitive applications, the TL5209 allows for low-noise capability through an external bypass capacitor connected to the BYP pin, which reduces the output noise of the regulator. Other features include current limiting, thermal shutdown, reverse-battery protection, and low temperature coefficient.

The TL5209 is available with adjustable output. Offered in an SOIC-8 surface-mount package, the TL5209 is characterized for operation over the virtual junction temperature ranges of –40°C to 125°C.

Figure 10. TL5209 Functional Block Diagram

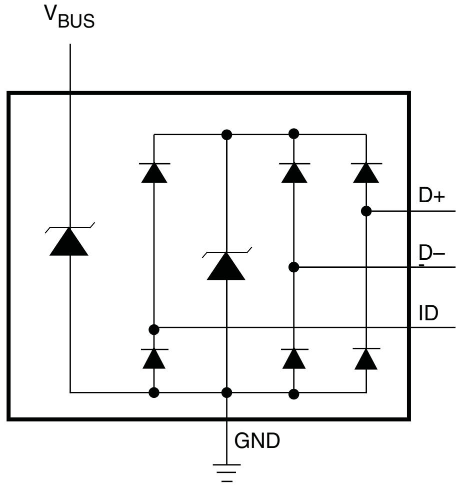


2.3.8 TPD4S012

The TPD4S012 is a four-channel Transient Voltage Suppressor (TVS) based Electrostatic Discharge (ESD) protection diode array for USB chargers and USB On-The-Go (OTG) interfaces.

The TPD4S012 provides IEC 61000-4-2 system level ESD Protection featuring 15-V tolerance on the VBUS line. The device is ideal for providing circuit protection for USB charger and OTG applications due to its high-voltage tolerance at the VBUS line and small flow-through package.

Figure 11. TPD4S012 Functional Block Diagram

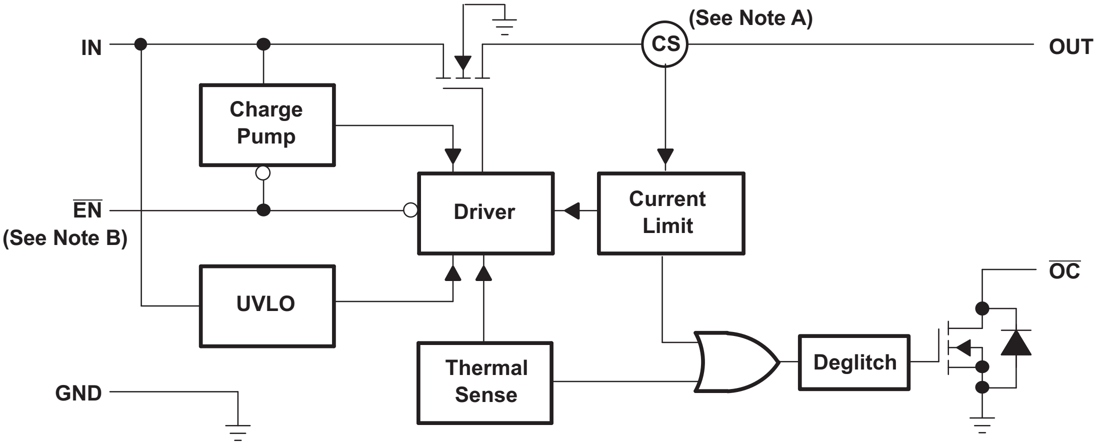


2.3.9 TPS2051B

The TPS20xxB power-distribution switches are intended for applications where heavy capacitive loads and short circuits are likely to be encountered. These devices incorporates 70-mΩ N-channel MOSFET power switches for power-distribution systems that require multiple power switches in a single package. Each switch is controlled by a logic enable input. Gate drive is provided by an internal charge pump designed to control the power-switch rise times and fall times to minimize current surges during switching. The charge pump requires no external components and allows operation from supplies as low as 2.7 V.

When the output load exceeds the current-limit threshold or a short is present, the device limits the output current to a safe level by switching into a constant-current mode, pulling the overcurrent (OCx) logic output low. When continuous heavy overloads and short circuits increase the power dissipation in the switch, causing the junction temperature to rise, a thermal protection circuit shuts off the switch to prevent damage. Recovery from a thermal shutdown is automatic once the device has cooled sufficiently. Internal circuitry ensures that the switch remains off until valid input voltage is present. This power-distribution switch is designed to set current limit at 1 A (typical).

Figure 12. TPS2051B Functional Block Diagram

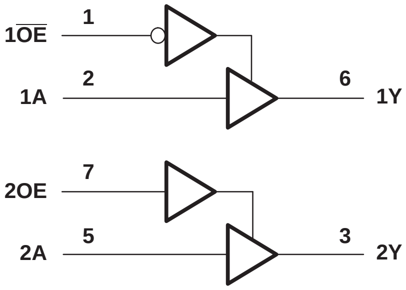


Note A: Current sense
Note B: Active low ($\overline{EN}$) for TPS2041B; Active high (EN) for TPS2051B

2.3.10 SN74LVC2G241

The SN74LVC2G241 device is designed specifically to improve both the performance and density of 3-state memory-address drivers, clock drivers, and bus-oriented receivers and transmitters. The SN74LVC2G241 device is organized as two 1-bit line drivers with separate output-enable (1OE, 2OE) inputs. When 1OE is low and 2OE is high, the device passes data from the A inputs to the Y outputs. When 1OE is high and 2OE is low, the outputs are in the high-impedance state. The SN74LVC2G241 is also an effective redriver, with a maximum output current drive of 32 mA.

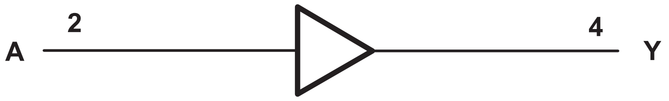
Figure 13. SN74LVC2G241 Functional Block Diagram



2.3.11 SN74LV1T34

The SN74LV1T34 device is a low voltage CMOS gate logic that operates at a wider voltage range for industrial, portable, and telecom applications. The output level is referenced to the supply voltage and is able to support 1.8-V, 2.5-V, 3.3-V, and 5-V CMOS levels. The input is designed with a lower threshold circuit to match 1.8 V input logic at VCC = 3.3 V and can be used in 1.8 V to 3.3 V level up translation. In addition, the 5 V tolerant input pins enable down translation (that is, 3.3 V to 2.5 V output at VCC = 2.5 V). The wide VCC range of 1.8 V to 5.5 V allows generation of desired output levels to connect to controllers or processors. The SN74LV1T34 device is designed with current-drive capability of 8 mA to reduce line reflections, overshoot, and undershoot caused by high-drive outputs.

Figure 14. SN74LV1T34 Functional Block Diagram



2.3.12 TI 15.4-Stack

TI's royalty-free TI 15.4-Stack is a complete software platform for developing applications that require extremely low-power, long-range, reliable, robust and secure wireless star-topology-based networking solutions. TI 15.4-Stack is a wireless stack component of the SimpleLink™ MCU platform that implements the standard IEEE 802.15.4e and 802.15.4g specification. For Sub-1 GHz frequencies TI 15.4-Stack also implements a frequency-hopping scheme adopted from Wi-SUN field area network (FAN) specification. This SDK also provides the required tools, real-time operating system (RTOS), and example applications for TI 15.4-Stack to help developers quickly get started developing their own star-topology-based wireless network products. The purpose of this document is to give an overview of TI 15.4-Stack to help developers run the out-of-box example applications and enable creation of custom TI 15.4-Stack based wireless star-topology networking solutions. This document introduces the essential need-to-know technology details for developing a wireless network based on the IEEE 802.15.4 and Wi-SUN FAN specification supported by TI 15.4-Stack.

For more details and to get the TI 15.4-Stack software, download the SimpleLink CC13x2 SDK, which includes the TI 15.4-Stack.

2.3.13 Processor SDK for AM335x Sitara Processors

The TI processor SDK is a unified software platform for TI embedded processors, which provides easy setup and fast out-of-the-box access to benchmarks and demonstrations. All releases of the processor SDK are consistent across TI's broad portfolio, which allows developers to seamlessly reuse and migrate software across devices. Developing scalable platform solutions has never been easier with the processor SDK and TI's embedded processor solutions.

TI processor Linux SDK highlights:

- Long-term stable (LTS) mainline Linux kernel support
- U-Boot bootloader support
- Linaro® GNU compiler collection (GCC) tool chains
- Yocto Project® OE Core compatible file systems

For more details and to get the processor SDK, see AM335x Processor SDK.

3 Hardware, Software, Testing Requirements, and Test Results

3.1 Required Hardware and Software

This section provides details on required hardware and software to run the universal data concentrator reference design with Ethernet, 6LoWPAN mesh, and RS-485 connectivity.

3.1.1 Hardware

This reference design is built with a TIDA-010032 docking board with standard TI EVMs of BeagleBone™ Black (BBB) and LAUNCHXL-CC1312R1. The TIDA-010032 docking board provides plug-in headers to mount the two standard TI EVMs to remove wires between the standard TI EVMs. The TIDA-010032 docking board includes a RS-485 signal chain and various peripheral header pin-outs (SDO, GPIO, UART, I^{2}C, and SPI) to allow extending this design with other connectivity.

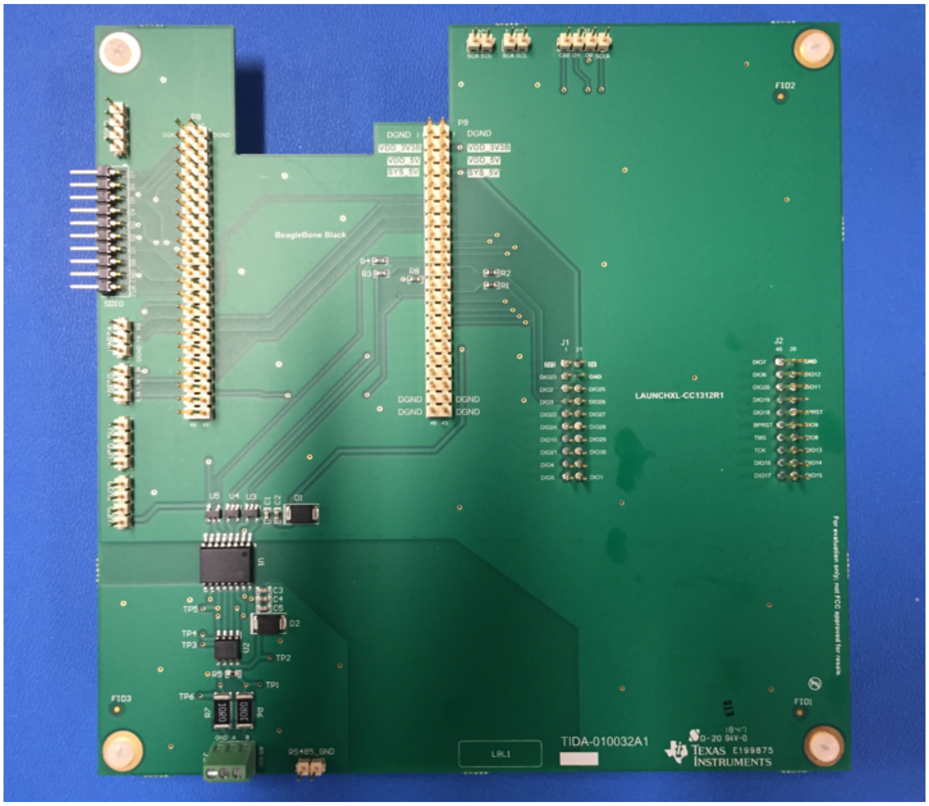
The following hardware is required to get the out-of-box application running and to develop applications:

- A TIDA-010032 docking board
  - Alternatively, wired connection with the TI standard EVMs for 5-V power supply, UART and GPIO. For the details, refer to Figure 16.
- An AM335x-based BeagleBone Black (BBB) to run gateway and web server applications
- A LAUNCHXL-CC1312R1 to run the TI 15.4-Stack coprocessor application
- RS-485 Half-Duplex Evaluation Module for RS-485 transceiver
  - In case when the TIDA-010032 docking board is not available
- An 16-GB micro SD card (the TI processor SDK image requires at least 8 GB of space)
- A 5-V power supply for the BeagleBone Black
- An Ethernet cable to connect the BeagleBone Black to the Internet
- A means to configure and set up the BeagleBone Black micro SD card (Microsoft® Windows® or Linux machine)
- A serial terminal program to connect to the BBB to run the web server and gateway applications in a PC

3.1.1.1 TIDA-010032 Docking Board

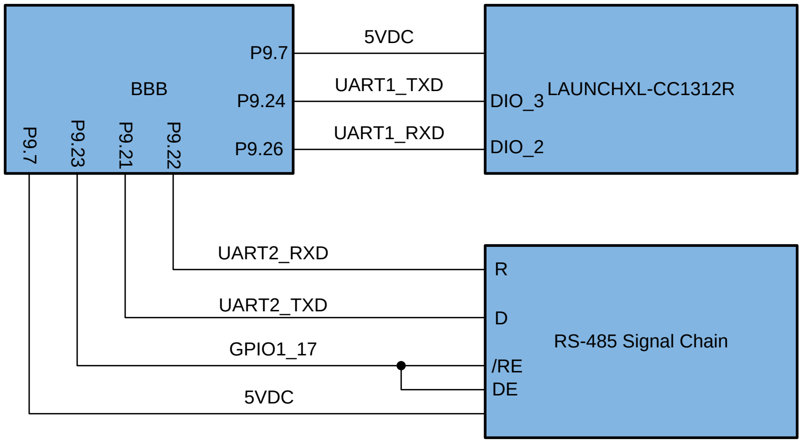
The TIDA-010032 docking board has two plug-in headers for LAUNCHXL-CC1312R and BBB EVMs, an isolated RS-485 signal chain and various peripheral pin-outs for VDD_3V3B, VDD_5V, GPIO, UART, SDO, I^{2}C, and SPI.

Figure 15. TIDA-010032 Docking Board



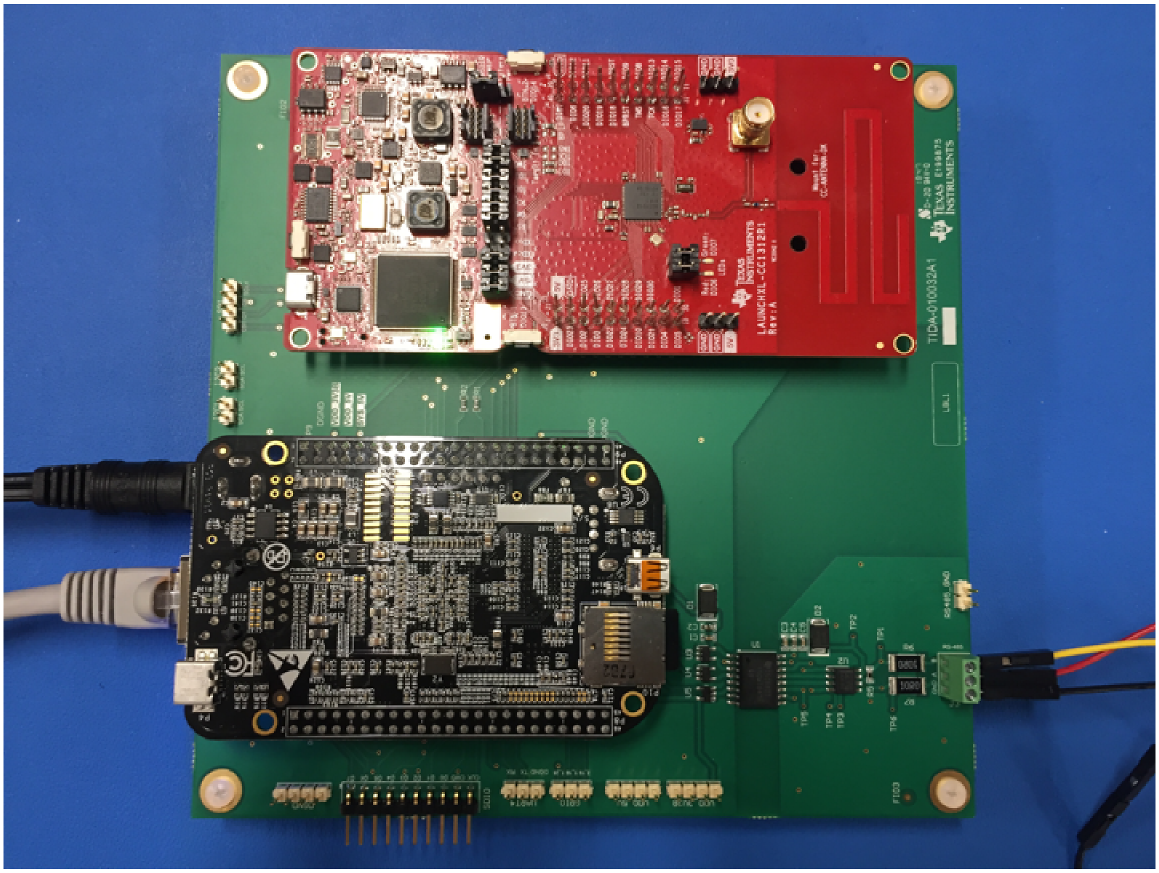
[Figure 16](#) shows the TIDA-010032 hardware block diagram with connections among BBB, LAUNCHXL-CC1312R and RS-485 signal chain. The 5-V power for the LAUNCHXL-CC1312R and RS-485 transceiver is supplied by the BBB. The LAUNCHXL-CC1312R is connected to the BBB via UART1 and the RS-485 transceiver is connected to the BBB via UART2. An additional GPIO connection between the RS-485 transceiver and BBB is needed to switch TRX mode.

Figure 16. TIDA-010032 Hardware Block Diagram



[Figure 17](#) shows the TIDA-010032 docking board plug-in with two standard EVMs of BBB and LAUNCHXL-CC1312R1. Check the pin labels on the TIDA-010032 docking board to connect the two EVMs in the right direction. Note that the LAUNCHXL-CC1312R1 EVM needs to remove the UART TX, and RX jumpers on the EVM.

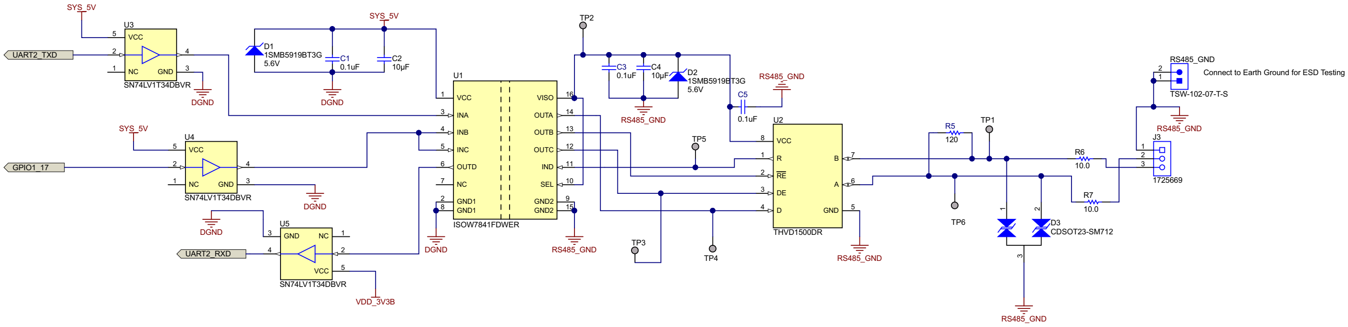
Figure 17. TIDA-010032 Docking Board With BBB and LAUNCHXL-CC1312R1 EVMs



3.1.1.2 Isolated RS-485 Signal Chain

The isolated RS-485 signal chain consists of a protection circuit against 1-kV surge (IEC 61000-4-5) transients, THVD1500, RS-485 transceiver, and ISOW7841 to isolate the RS-485 signal domain from the host processor as well as to power 5 V to the RS-485 transceiver. The two-pin jumper of RS485_GND can be connected to the earth ground when performing the ESD testing. Figure 18 shows the schematic of the isolated RS-485 signal chain.

Figure 18. Schematic: Isolated RS-485 Signal Chain



3.1.2 Software

The following software is required to get the out-of-box application running and to develop applications:

- AM335x Linux SDK (PROCESSOR-SDK-LINUX-AM335X)
- SimpleLink CC13x2 wireless MCU SDK (SIMPLELINK-CC13X2-SDK)

- Code Composer Studio™ v8.0 or above
- TIDA-010032 software example
  - Gateway applications (6LoWPAN mesh and RS-485)
  - Web server application
- Linux host machine to build a new AM335X Linux kernel device tree file and gateway applications

3.1.2.1 BeagleBone™ Black Setup

This section provides the details to set up the BeagleBone Black EVM with the AM335x Linux SDK.

3.1.2.1.1 BeagleBone™ Black SD Card

Program the SD card with the AM335x Linux processor SDK image using the following steps:

1. Download the pre-built TI Linux processor SDK SD card image.
2. Program the micro SD memory card (> 8GB). See the instructions in Processor SDK Linux Creating an SD Card with Windows.

3.1.2.1.2 SD Card Booting BeagleBone™ Black

Boot the BeagleBone Black EVM from the micro SD card using the following steps:

1. Power off the BeagleBone Black EVM
2. Insert the micro SD card into the BeagleBone Black
3. Press and hold the boot switch (S2). Note that the boot switch is detected only at initial power on.
4. Power the BeagleBone Black and then release the boot switch

NOTE: The BeagleBone Black includes an eMMC device on it which comes pre-flashed with an Angstrom distribution. Because eMMC is the default boot mode for this board we need to prevent it from being able to boot by either removing or renaming the MLO.

To do this you will need to wipe out the MLO file stored in the eMMC. To eliminate the MLO, first boot up the board with the USB mini cable connected to the board and your PC. Once the Angstrom kernel loads your host will mount the eMMC boot partition on your Linux host under /media/BEAGLEBONE. You can then erase or rename the MLO file here. You can also login to the BeagleBone Black and rename or remove /boot/MLO (for example, mv /boot/MLO /boot/OLDMLO).

With this change, steps #3 and #4 previously described can be skipped to boot from SD card.

3.1.2.1.3 Enable UARTs via the BBB Header Pins

Enabling the UART header pins on the BBB requires to update the Linux kernel device tree blob (dtb) file using the following steps:

1. Install the AM335x Linux SDK in a Linux host machine
2. Export the Linux tool chain in the host machine

   ```
   export PATH=<SDK_install_directory>/linux-devkit/sysroots/x86_64-arago-linux/usr/bin/:$PATH
   ```

3. Go to <SDK_install_directory>/board-support/linux-xx.xx.xx/arch/arm/boot/dts and modify "am335x-bone-common.dtsi" as the following code shows (changes marked in bold).
   a. This change activates UART1 (/dev/ttyS1), UART2 (/dev/ttyS2), and UART4 (/dev/ttyS4).

   ```
   .......
   uart0_pins: pinmux_uart0_pins {
   pinctrl-single,pins = <
   AM33XX_IOPAD(0x970, PIN_INPUT_PULLUP|MUX_MODE0)    /*uart0_rxd.uart0_rxd*/
   AM33XX_IOPAD(0x974, PIN_OUTPUT_PULLDOWN|MUX_MODE0) /*uart0_txd.uart0_txd*/
   >;
   ```

```
};
uart1_pins: pinmux_uart1_pins {
pinctrl-single,pins = <
AM33XX_IOPAD(0x980, PIN_INPUT_PULLUP|MUX_MODE0)    /*uart1_rxd.uart1_rxd */
AM33XX_IOPAD(0x984, PIN_OUTPUT_PULLDOWN|MUX_MODE0) /*uart1_txd.uart1_txd*/
>;
};
uart2_pins: pinmux_uart2_pins {
pinctrl-single,pins = <
AM33XX_IOPAD(0x950, PIN_INPUT_PULLUP|MUX_MODE1)    /*uart2_rxd.uart2_rxd*/
AM33XX_IOPAD(0x954, PIN_OUTPUT_PULLDOWN|MUX_MODE1) /*uart2_txd.uart2_txd*/
>;
};
uart4_pins: pinmux_uart4_pins {
pinctrl-single,pins = <
AM33XX_IOPAD(0x870, PIN_INPUT_PULLUP|MUX_MODE6)    /*uart4_rxd.uart4_rxd*/
AM33XX_IOPAD(0x874, PIN_OUTPUT_PULLDOWN|MUX_MODE6)    /*uart4_txd.uart4_txd*/
>;
}

.......

&uart0 {     pinctrl-names = "default";
          pinctrl-0 = <&uart0_pins>;
          status = "okay"; };
&uart1 {     pinctrl-names = "default";
          pinctrl-0 = <&uart1_pins>;
          status = "okay"; };
&uart2 {     pinctrl-names = "default";
          pinctrl-0 = <&uart2_pins>;
          status = "okay"; };
&uart4 {     pinctrl-names = "default";
          pinctrl-0 = <&uart4_pins>;
          status = "okay"; };
```

4. Go to the Linux Kernel top directory of <SDK_install_directory>/board-support/linux-xx.xx.xx/
   a. Build the Linux Kernel configuration file

      ```
      make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- tisdk_am335x-evm_defconfig
      ```

   b. Build a new am335x-boneblack.dtb

      ```
      make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- am335x-boneblack.dtb
      ```

5. Replace the new am335x-boneblack.dtb in the directory of <SDK_install_directory>/board-support/linux-xx.xx.xx/arch/arm/boot/dts/ in your host Linux machine with the old am335x-boneblack.dtb in the directory of /boot/ in the BBB EVM.

These steps are not necessary when the USB interface on the BBB EVM is used to connected to the LAUNCHXL-CC1312R1.

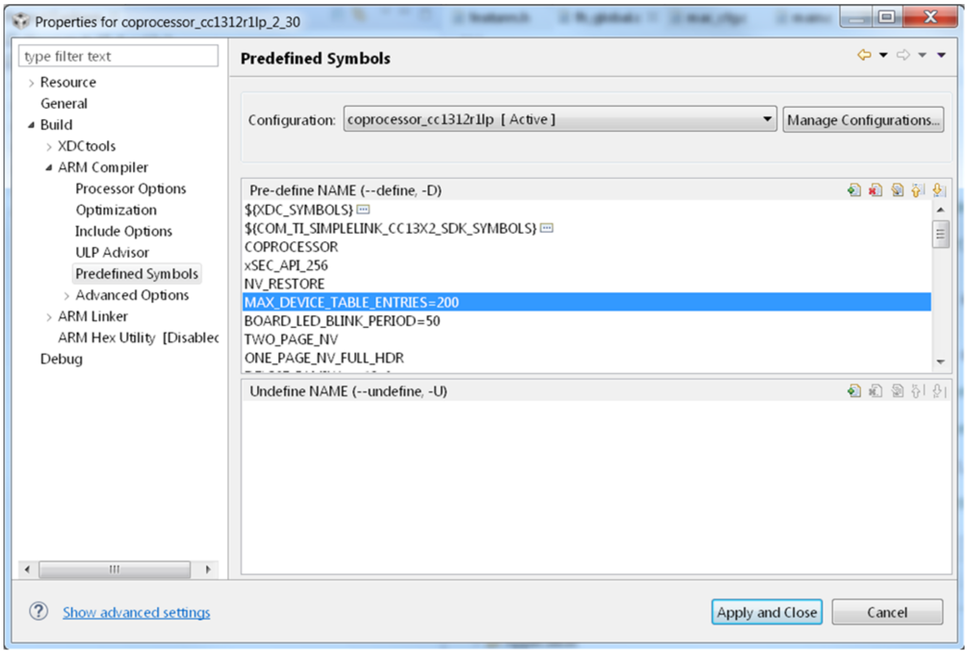
3.1.2.2 SimpleLink™ CC13x2 SDK

The SimpleLink CC13x2 SDK includes the TI 15.4-Stack co-processor which is the underlying low-level stacks of the 6LoWPAN RF mesh network stacks running on the AM335x MPU.

The SimpleLink CC13x2 SDK provides a pre-built TI 15.4-Stack co-processor that configured the maximum FH neighbor entries of 50 (based on SimpleLink CC13x2 SDK v2.30). On the other hand, the 6LoWPAN mesh gateway application, parts of TIDA-010032 software example, configures the maximum neighbor entries in the TI 15.4-Stack to 200 via a FH PIB command. Thus, with the pre-built TI 15.4-Stack, it will return error and the gateway application cannot complete its initialization procedure. To address this, the TI 15.4-Stack co-processor needs to be rebuilt with change of maximum FH neighbor entries to 200. This requires to re-compile the TI 15.4-Stack co-processor using the following steps:

1. Install SimpleLink CC13x2 SDK (verified with v2.30).
2. Open the TI 15.4-Stack co-processor project in the Code Composer Studio IDE.
3. Update "MAX_DEVICE_TABLE_ENTRIES" in the predefined symbols from 50 to 200 as Figure 19 shows.
4. Modify "fh_global.c" in the directory of Application/MAC/FreqHop/ as the following code shows:
5. Build the Code Composer Studio project.

Figure 19. MAX_DEVICE_TABLE_ENTRIES for TI 15.4-Stack Coprocessor



Configuration: coprocessor_cc1312r1lp [Active]

```
${XDC_SYMBOLS}
${COM_TI_SIMPLELINK_CC13X2_SDK_SYMBOLS}
COPROCESSOR
xSEC_API_256
NV_RESTORE
MAX_DEVICE_TABLE_ENTRIES=200
BOARD_LED_BLINK_PERIOD=50
TWO_PAGE_NV
ONE_PAGE_NV_FULL_HDR
```

```
/*!  FH PIB table  */
const FH_PibTbl_t FH_PibTbl[] = {
.......
/*!< maximum sleep node */
{offsetof(FHPIB_DB_t, macMaxNonSleepNodes), sizeof(uint16_t), 0, MAX_DEVICE_TABLE_ENTRIES},
/*!< maximum temp table node */
{offsetof(FHPIB_DB_t, macMaxSleepNodes), sizeof(uint16_t), 0, MAX_DEVICE_TABLE_ENTRIES},
.......
};
```

Alternatively, the configuration for the maximum FH neighbor entries can be reduced to 50 in the 6LoWPAN mesh gateway application running on the BBB EVM. Simply change the following line, as shown, in api_mac.h (in the directory of <TID_install_directory>/6lowpan_mesh/Components/gateway/include/).

```
#define CONFIG_MAX_DEVICES     200  /* change to 50 */
```

3.1.2.3 TIDA-010032 Gateway Examples

The TIDA-010032 reference design provides a web server application and two gateway examples: one is 6LoWPAN mesh and one is RS-485. This section provides the details of how to run, build, and debug the gateway examples. The details of the web server application will be discussed in Section 3.1.2.4.

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

The pre-built 6LoWPAN mesh and RS-485 gateway executable is located in the directory of <TID_install_directory>/6lowpan_mesh/Projects/gateway/proj/ and <TID_install_directory>/rs485/Projects/gateway/proj/, respectively. In case when developers need to modify the gateway applications, the application needs to be rebuilt using the following steps:

```
/* 6LoWPAN mesh gateway */
cd <TID_install_directory>/6lowpan_mesh/Projects/gateway/proj/
/* RS-485 gateway */
cd <TID_install_directory>/rs485/Projects/gateway/proj/

/* Build in Linux Host machine */
make clean
make
```

The gateway applications can be also built in the BBB EVM with a command line option of "local=y" as the following shows:

```
/* 6LoWPAN mesh gateway */
cd <TID_install_directory>/6lowpan_mesh/Projects/gateway/proj/
/* RS-485 gateway */
cd <TID_install_directory>/rs485/Projects/gateway/proj/

/* Build in BBB EVM */
root@am335x-evm:~#make clean
root@am335x-evm:~#make local=y
```

Note that, if the build process is never completed due to timestamp issue, execute the following commands and build again.

```
root@am335x-evm:~#date –s “2018-10-17 16:20:00” /*update to current date/time*/
root@am335x-evm:~#find /your/directory -type f -exec touch {} +
```

3.1.2.3.1 6LoWPAN Mesh Gateway

Running the 6LoWPAN mesh gateway application is simple using the following steps:

1. Copy "gwapp.exe" in the BBB EVM (if the application was not built in the BBB EVM).
2. Run "gwapp.exe"

```
root@am335x-evm:~# ./gwapp.exe
```

The application provides some command line options to configure the 6LoWPAN mesh gateway application without recompile. The "-s" option configures the UART interface. The command line options of "-f" and "-p" allows users to configure the FH network name and PAN ID, respectively. When multiple 6LoWPAN mesh networks run together within the transmission range to each other, it is strongly recommended to configure different FH network name and PAN ID. Note that with the change the FH network name of the 6LoWPAN mesh end-node should be changed accordingly.

```
root@am335x-evm:~# ./gwapp.exe -h
TIDA-010032 6LoWPAN Mesh Data Concentrator
./gwapp.exe [options] Options are:
-s serial dev for UART (default /dev/ttyS1)
-f TI-15.4 fh network name (default TI-6LOWPAN-MESH)
-p TI-15.4 fh PAN ID (default = 1234 in hex)
```

3.1.2.3.2 RS-485 Gateway

Running the RS-485 mesh gateway application is simple using the following steps:

1. Copy "rs485_gwapp.exe" in the BBB EVM (if the executable was not built in the BBB EVM).
2. Run "rs485_gwapp.exe".

```
root@am335x-evm:~# ./rs485_gwapp.exe
```

The RS-485 gateway application provides a command line option to configure the UART interface. By default, the interface is set to /dev/ttyS2.

```
root@am335x-evm:~# ./rs485_gwapp.exe -h
TIDA-010032 RS-485 Concentrator
./rs485_gwapp.exe [options] Options are:
-s serial dev for UART (default /dev/ttyS2)
```

3.1.2.3.3 Debugging Gateway Examples

There are two ways to debug the gateway examples: one is to add "printf" to debug the software examples and one is to debug with "gdb" tool. The gateway examples provides the built-in debug printfs that are disabled by default. The printfs in each file can be activated by changing the DEBUG macro value from "DEBUG_NONE" to "DEBUG_FULL" as the following shows:

```
#undef DEBUG
#define DEBUG DEBUG_NONE /* DEBUG_FULL to activate the debug logs */
```

Another way to debug the software example is to run the gateway examples in the gdb debug mode. A following simple instruction is given to run the examples with the gdb tool. There are various command line options to run in the debug mode. The details are found in GDB: The GNU Project Debugger.

```
root@am335x-evm:~# gdb gwapp.exe
(gdb) run
Starting program: /home/root/gwapp.exe
[Thread debugging using libthread_db enabled] Using host libthread_db library
"/lib/libthread_db.so.1".
```

3.1.2.4 Web Server Application

The web server application runs without compile. The entire sources in the directory of <TID_install_directory>/webapp need to move to the BBB EVM. The web application can launch by executing the following command in the BBB EVM.

```
root@am335x-evm:~/webapp# node app.js
```

NOTE: The web server application runs as a TCP socket client and thus the gateway applications need to start first before launching the web application. In addition, terminate the web server application before closing the gateway applications to avoid additional delay to close the TCP socket interfaces.

The web server application can run with both 6LoWPAN mesh and RS-485 gateways or one of them, which can be configured in the "webapp/setttings.json" as the following code shows. The "lowpanPort" denotes the port number of the 6LoWPAN mesh gateway and the "rs485Port" means the port number of the RS-485 gateway. The port numbers in the configuration were set based on the assigned TCP port numbers for the gateway applications.

```
{"httpPort": 8000,
"lowpanPort":40000,
"rs485Port": 50000}
```

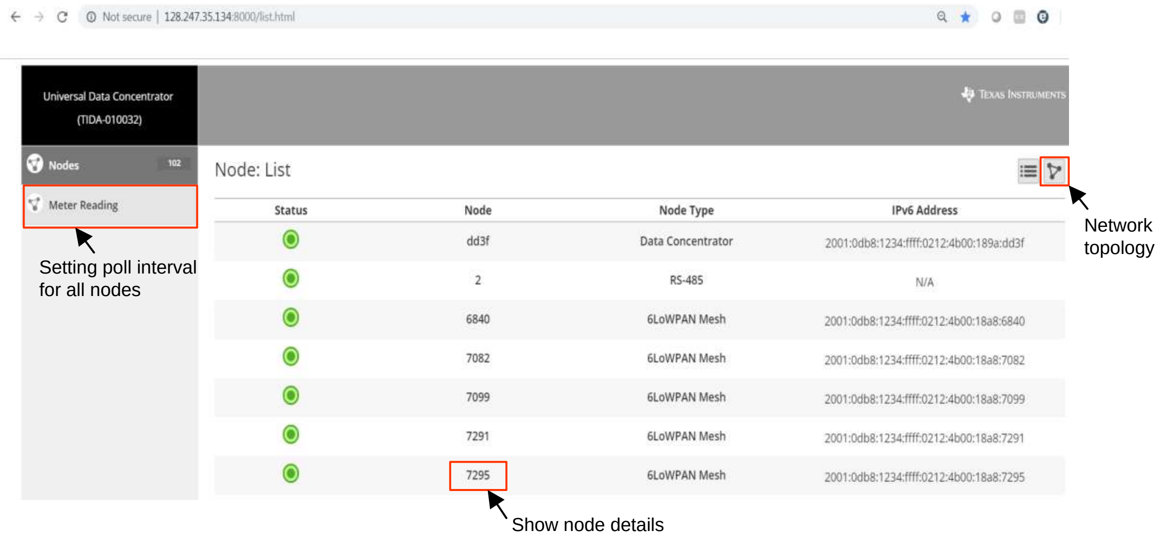
NOTE: In case when only a single gateway program (6LoWPAN or RS-485) is started, the port number of the other gateway should be set to zero. Without this change, the web server application will fail to run due to TCP connection errors.

3.1.2.4.1 Web GUI

Once the web server starts, the main web page can be loaded in the web browser by typing "<IPV4_address_of_BBB>:8000". The IPv4 address of the BBB EVM can be found by typing "ifconfig" in the BBB EVM console and 8000 is the assigned web port number.

Figure 20 shows the screen capture of the web page that shows 100 6LoWPAN mesh and one RS-485 end-node. The Node Type column shows if the end-node connected to the data concentrator uses 6LoWPAN mesh or RS-485. The IPv6 Address column is only applicable for the 6LoWPAN mesh nodes since they run over the IPv6 protocol. The Node column shows the end-node identifier (the last 2B of IPv6 address for 6LoWPAN mesh nodes and a 2B unique identifier for the RS-485 end-nodes). The Status column represents the connection status of each end-node. The green color is connected state and the red color means disconnected.

Figure 20. TIDA-010032 Web Pages-Main Page



Not secure | 128.247.35.134:8000/list.html

Universal Data Concentrator (TIDA-010032)

Node: List

Status	Node	Node Type	IPv6 Address
●	dd3f	Data Concentrator	2001:0db8:1234:ffff:0212:4b00:189a:dd3f
●	2	RS-485	N/A
●	6840	6LoWPAN Mesh	2001:0db8:1234:ffff:0212:4b00:18a8:6840
●	7082	6LoWPAN Mesh	2001:0db8:1234:ffff:0212:4b00:18a8:7082
●	7099	6LoWPAN Mesh	2001:0db8:1234:ffff:0212:4b00:18a8:7099
●	7291	6LoWPAN Mesh	2001:0db8:1234:ffff:0212:4b00:18a8:7291
●	7295	6LoWPAN Mesh	2001:0db8:1234:ffff:0212:4b00:18a8:7295

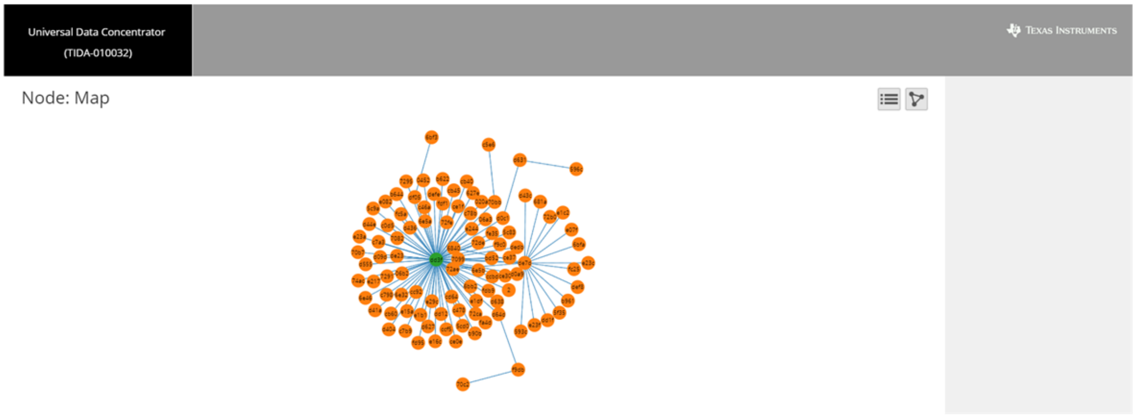
Setting poll interval for all nodes

Network topology

Show node details

Figure 21 shows a screen capture for Node:Map page by clicking the topology button on the top left in the main page. This page shows the complete route from end nodes to the data concentrator.

Figure 21. TIDA-010032 Web Pages-Node: Map

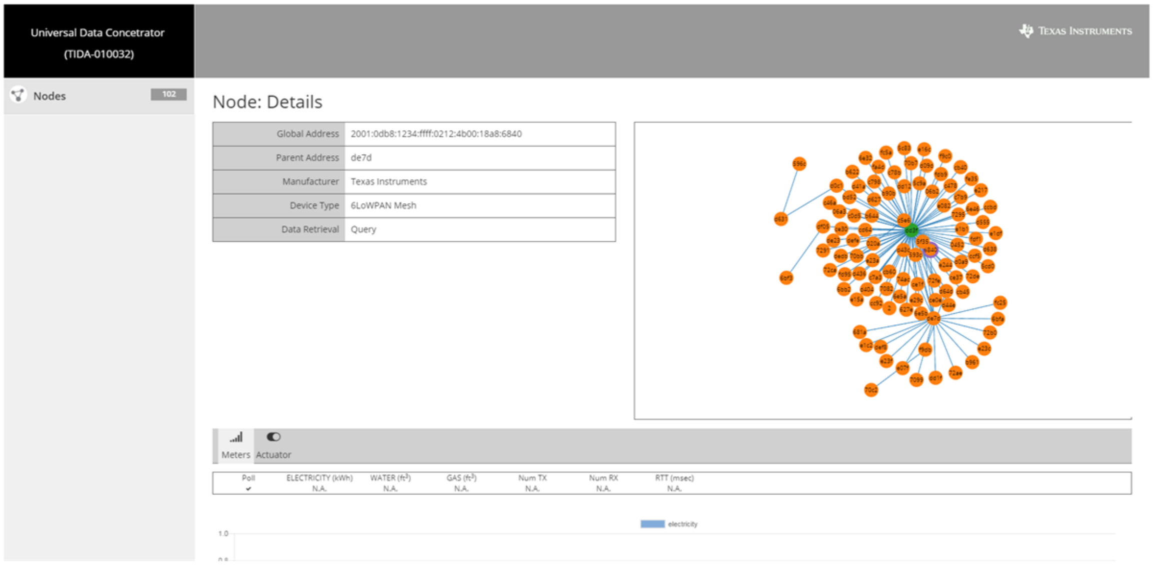


Universal Data Concentrator (TIDA-010032)

Node: Map

Clicking each node in the main page goes into the Node:Details page shown in Figure 22. In this page, users can perform one-time polling to read meter data and to control LED for the end node. This page also shows the historical data of meter reading and network performance of round-trip time (RTT) and number of TX and RX by clicking "Query".

Figure 22. TIDA-010032 Web Pages-Node: Details



Universal Data Concetrator
(TIDA-010032)

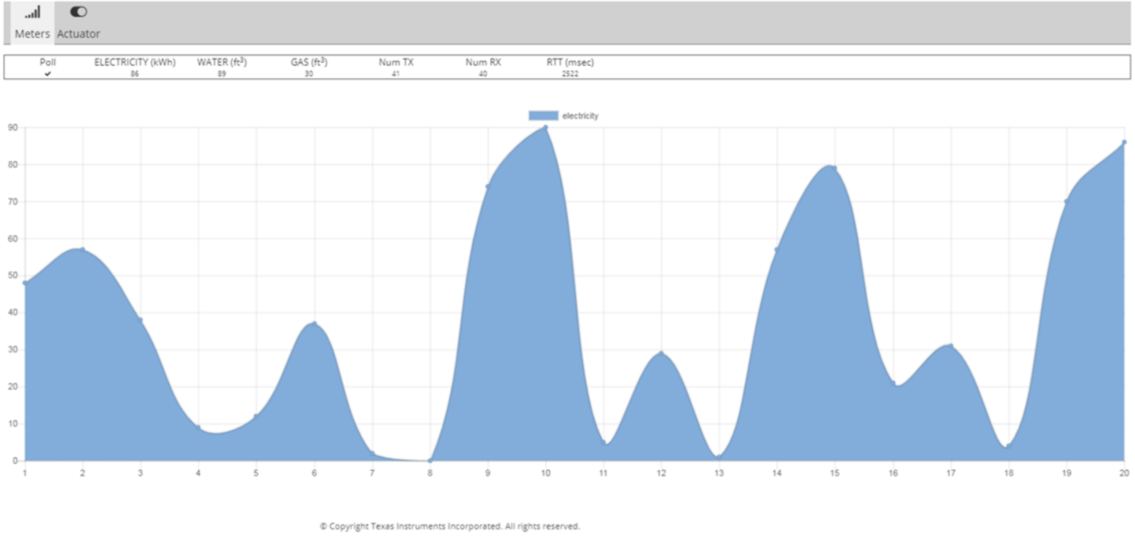
Node: Details

Global Address	2001:0db8:1234:ffff:0212:4b00:18a8:6840
Parent Address	de7d
Manufacturer	Texas Instruments
Device Type	6LoWPAN Mesh
Data Retrieval	Query

Poll	ELECTRICITY (kWh)	WATER (ft^3)	GAS (ft^3)	Num TX	Num RX	RTT (msec)
✓	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.

Figure 23 shows the screen capture of meter reading information, which is located in the bottom of the Node:Details page. The web page shows graphical representation with the latest 20 meter reading data.

Figure 23. TIDA-010032 Web Pages-Meter Reading Data

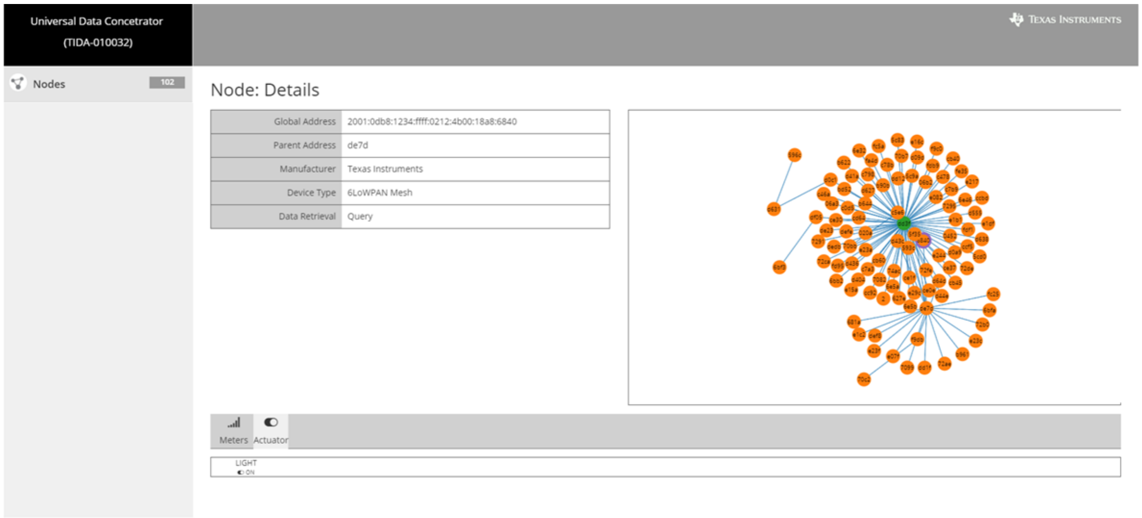


Poll	ELECTRICITY (kWh)	WATER (ft^3)	GAS (ft^3)	Num TX	Num RX	RTT (msec)
✓	86	89	30	41	40	2522

© Copyright Texas Instruments Incorporated. All rights reserved.

Figure 24 shows the LED control in the Actuator tab in the web page. The gateway application implements two-way handshake to synchronize the LED status in the web and the target node, which means that the LED status in the web page changes only when the data concentrator receives confirmation from the target end node.

Figure 24. TIDA-010032 Web Pages-LED Control



Universal Data Concetrator
(TIDA-010032)

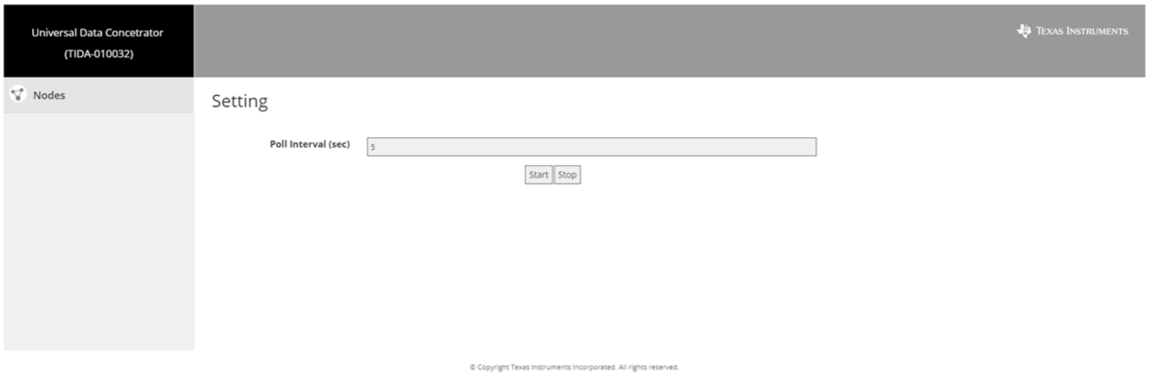
Node: Details

Global Address	2001:0db8:1234:ffff:0212:4b00:18a8:6840
Parent Address	de7d
Manufacturer	Texas Instruments
Device Type	6LoWPAN Mesh
Data Retrieval	Query

LIGHT
ON

In addition to one-time polling, the web server provides a feature to poll all the end nodes by clicking "Meter Reading" menu on the left side of the main page. Figure 25 shows the screen capture. The poll interval can be set any values greater than zero to start polling for each node connected to the data concentrator. The polling message is sent to each node one by one and the poll interval defines the time gap between two consecutive poll messages. Thus, assuming that there are 100 nodes connected to the date concentrator, the time duration to complete to poll to all the nodes will be 100*poll_interval.

Figure 25. TIDA-010032 Web Pages-Setting



Universal Data Concetrator
(TIDA-010032)

Setting

Poll Interval (sec): 5

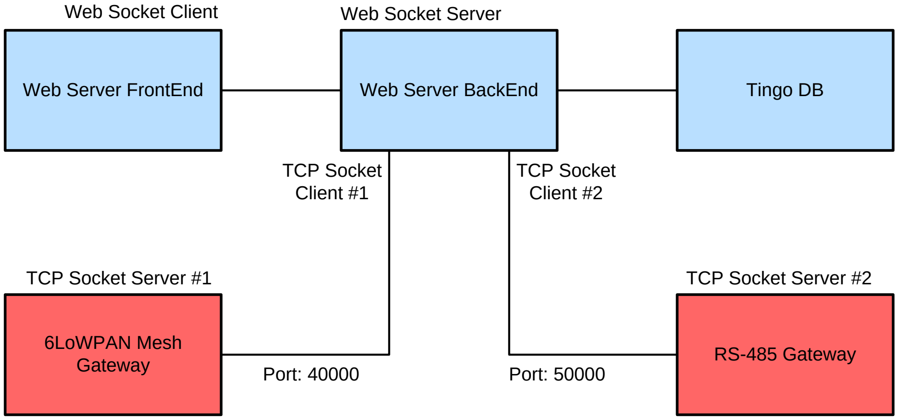
© Copyright Texas Instruments Incorporated. All rights reserved.

3.2 TIDA-010032 Software Architecture

Figure 26 shows the TIDA-010032 software architecture with a web application and gateway applications.

The web application consists of web server frontend, backend, and Tingo DB (data base). The web server frontend and backend are communicated via a web socket with port number of 5000. The gateway applications are connected to the web server backend via TCP sockets. The gateway applications run as TCP socket servers. The 6LoWPAN mesh gateway uses the TCP port of 40000 and the RS-485 gateway uses the TCP port of 50000.

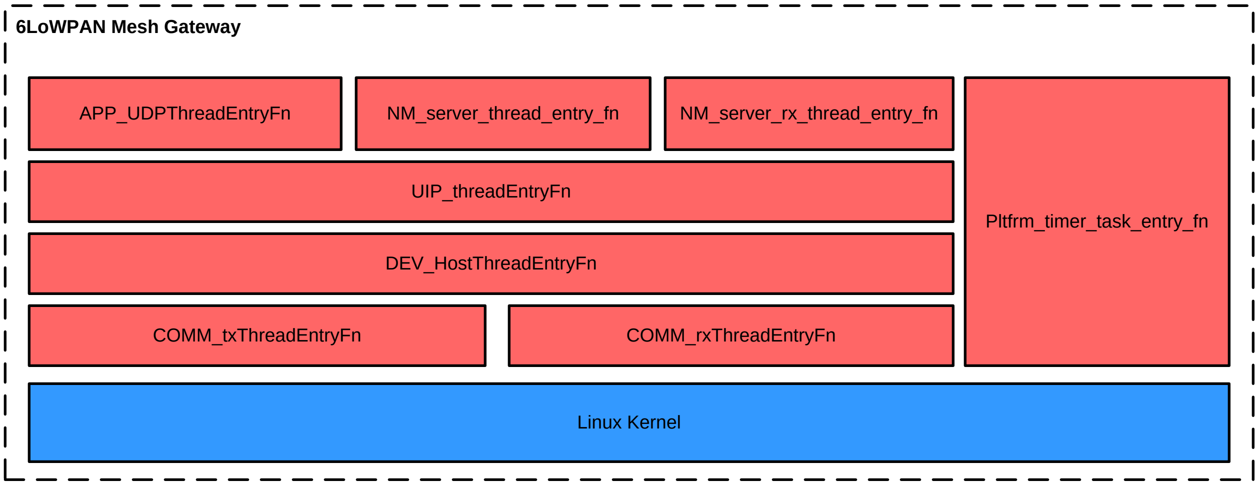
Figure 26. TIDA-010032 Software Overview



3.2.1 6LoWPAN Mesh Gateway Application

Figure 27 shows the 6LoWPAN mesh gateway software architecture running over Linux. The software example implements the TI 15.4-Stack host interface, 6LoWPAN mesh network stacks, and web application interface.

Figure 27. 6LoWPAN Mesh Gateway Software Architecture



3.2.1.1 COMM Threads

The COMM_txThread and COMM_rxThread are the UART interface between CC1312R and AM335x devices. The UART baud rate is configured to 115200 bps.

The COMM_rxThread reads the message header incoming from the UART. If the message header is valid, it continue to read the remaining bytes based on the length field in the header. If the CRC passes, the COMM_rxThread parses the message and, based on the message type, it passes the message to another thread or updates the receive status.

The COMM_txThread builds a message based on the host message format and sends it via UART. The host message format follows the TI 15.4-Stack co-processor message specification, which is defined in "ti-15.4-stack-cop-interface-guide.pdf" in the directory of <SimpleLink_CC13x2_SDK_install_directory>\docs\ti154stack.

3.2.2 DEV Thread

The DEV_HostThread configures the TI 15.4-Stack, maintains the host state machine, and acts as data interface between the 6LoWPAN network stacks and TI 15.4-Stack. Once the thread starts, it configures the MAC and FH PIB values for the TI 15.4-Stack. The operating frequency band and the number of active channels can be configured at this time. By default, the PHY ID is set to APIMAC_STD_US_915_PHY_1, which is the 50-kbps FSK over a 902-MHz frequency band with 129 sub-channels. Based on regional requirement, this configuration can be changed in host_config.h in the directory of /6lowpan_mesh/Components/gateway/include/. The available PHY ID options, shown in the following code, are defined in api_mac.h. In the given frequency band plan, a partial set of sub-channels can be activated by changing the FH channel masks (CONFIG_FH_CHANNEL_MASK and FH_ASYNC_CHANNEL_MASK) defined in host_config.h. The details to configure this are found in the *How to Configure Different Frequency Bands for the 6LoWPAN Mesh TI Designs* application report.

```
/*! 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
/*! 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
```

Once the initialization completes, the thread changes the host state to "JOINED" and starts a new FH network as a coordinator to maintain 6LoWPAN mesh end-nodes joined to the network.

3.2.3 UIP Thread

The UIP Thread runs the 6LoWPAN mesh network stacks of 6LoWPAN, IPv6, ICMPv6, RPL, and UDP. This thread processes messages incoming from APP and DEV Threads.

3.2.4 NM Server Threads

NM Server Threads is the interface between web server application and 6LoWPAN mesh gateway application. At initial, the NM Server TX thread creates a TCP socket as a server to interface to the web server backend. When the thread receives messages from other threads in the other threads in the gateway program, it formats the message in a JSON format and delivers to the web server backend via the TCP socket. The NM Server RX thread listens to the TCP socket and parses messages incoming from the web server backend. When the received message is in a valid form, it passes the message to the APP thread via mailbox_post.

3.2.5 APP Thread

The APP thread interfaces between the UIP thread and NM Server thread to pass messages received from the NM Server thread to the UIP thread or to deliver messages incoming from the UIP thread to the NM Server thread. When the APP thread receives a message from the NM Server thread, it formats the received message shown in Figure 28. For demo purpose, there are two message types defined: 1 for LED_CONTROL and 2 for POLL_REQ. For both message types, the data concentrator carries 1B payload. For LED_CONTROL, the 1B payload means the LED status to be changed (1:ON, 0:OFF). For POLL_REQ message, the 1B payload is always zero.

As a response, end-node sends 6B meter reading data (2B: electricity, 2B: water, 2B: gas) in the payload with the message type of POLL_REQ or 1B led status for the message type of LED_CONTROL. For demo, the meter reading data was chosen from rand(0, 100). The 6LoWPAN mesh end-node demo example (6LoWPAN mesh example with build configuration of "debug_poll_demo") can be found in TIDA-010024.

Figure 28. 6LoWPAN Mesh Application Message Format

1B	2B	2B	
Message Type	Sequence #	Payload Size	Payload

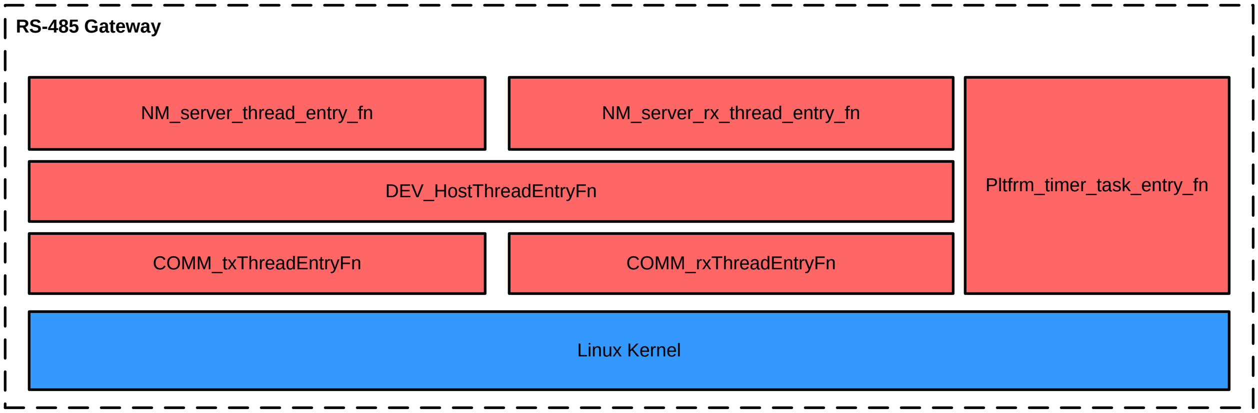
3.2.6 Pltfrm Timer Thread

The Pltfrm Timer thread maintains the timers running in the gateway application and handles the timeout events for the timer entries in the list.

3.2.7 RS-485 Gateway

The RS-485 gateway application has a similar architecture to the 6LoWPAN mesh gateway application. Since this example provides a simple application with UART and web server interfaces, developers can use this example as a baseline software to implement another application using UART and/or web server. The NW Server threads act as an agent to interface to the web application via TCP socket. The DEV Host thread runs as a middleware in-between COMM threads (UART interface) and the NW Server thread (web application interface). The Platform timer maintains the timers for the RS-485 gateway application.

Figure 29. RS-485 Gateway Software Architecture



3.2.7.1 *RS-485 Application Message Format*

For demonstration purposes, an application message format was defined as Figure 30 shows. There are two message types defined: 1 for LED_CONTROL and 2 for POLL_REQ. For both message types, the data concentrator conveys 1B payload. For LED_CONTROL, the 1B payload means the LED status to be requested (1: ON, 0:OFF). For POLL_REQ message, the 1B payload is always zero.

RS-485 end-nodes send 6B meter reading data (2B: electricity, 2B: water, 2B: gas) in the payload with the message type of POLL_REQ or 1B led status for the message type of LED_CONTROL. The meter reading data was chosen from rand(0, 100).

Figure 30. RS-485 Application Message Format

2B	2B	2B	2B	
Message Type	Source Address	Destination Address	Payload Size	Payload

3.2.8 Web Server Application

3.2.8.1 Web Server Backend

The web backend processes the data received from gateway applications and web frontend. Once the web backend starts, it connects to the TCP servers and then listens on the 6LoWPAN mesh and RS-485 TCP socket ports for any incoming messages, and it processes the messages based on the received message type.

If a button is pressed in the frontend, the web backend will trigger the corresponding handler function to send message via the 6LoWPAN mesh TCP socket or RS-485 TCP socket, or to perform DB query.

3.2.8.2 Web Server Frontend

The web frontend provides users network topology and meter reading information. User can connect to the web frontend in the web browser by typing “ipV4Address_of_BBB”:8000. The main web pages are as follows:

- /list.html – This is the default web page which shows all the nodes joined to the network.
- /details.html – This shows the node specific information. Users can view the meter reading data in this page.
- /map.html – This shows the network topology.
- /manage_gateway.html - This allows users to configure the polling interval setting.

The web page to retrieve DB information from web GUI as follows:

- /meterData – This retrieves the stored meter data for the entire network
- /meterData/id - This retrieves the stored meter data for a specific node id
- /nodes – This retrieves the node information for the entire network
- /nodes/id - This retrieves the node information for a specific node

3.2.8.3 Tingo DB

The Tingo DB stores meter data and node information. By default, up to 20,000 data items are stored in the DB, and items will be overwritten in a FIFO manner if the maximum number of items is reached. The maximum size can be changed in the following code in network_manager.js.

```
var MAX_NUM_DB_ITEMS = 20000;
```

Standard Tingo DB (or Mongo DB) APIs can be used to find, insert, and remove data. Refer to TingoDB API for the details.

3.2.8.4 Communication Interfaces

3.2.8.4.1 Web Backend and Gateway Applications

The web backend communicates with gateway applications via TCP sockets. After the web backend is started, it will connect to the 6LoWPAN mesh gateway server port and RS-485 gateway server port separately.

The following downlink messages are sent from web backend to gateway application:

- meter_data_pull: The web backend iterates through the existing nodes in the network and sends meter_data_pull message based on the specified poll interval. The pull message can be triggered from the web frontend as well. The message format is as follows: [2, dest_addr]
- led_put: When users click the LED toggle button in the frontend, a LED put message will be sent to the target node. The message format is as follows: [1, dest_addr, on_off_status]. The "on_off_status" can be 0 or 1.

The following uplink messages are sent from gateway applications to web backend. The messages are encoded in a JSON format. The codes can be found in nm.c in the gateway applications.

- dao_report (for 6LoWPAN mesh gateway): This contains the node information and parent information. It allows the web backend to construct network topology and display in the frontend.

```
{ "type": "dao_report",  "sender": %s,  "parent": %s, "lifetime": %d}
```

- add_node (for RS-485 gateway): This contains the 2B node information. It allows the web backend to construct network topology and display in the frontend.

```
{ "type": "add_node",  "addr": %d }
```

The sender address is in IPv6 address format for 6Lowpan mesh nodes, and in 2B address for RS-485 nodes. RS-485 nodes assume that the parent address is always the address of the data concentrator since it supports star topology.

- delete_node: This information is sent to remove a node from the network topology. The address format is string for 6LoWPAN mesh nodes and integer for RS-485 nodes.

```
{ "type": "delete_node",  "addr": %s }
```

- read_ info: This contains the meter reading data information. The meter reading data information is stored in Tingo DB, and it will broadcast to the web frontend via web socket.

```
{ "type": "read_info",  "addr": %s,  "data": { "rtt": %d,  "electricity": %d,  "water": %d,
"gas": %d } }
```

- led_control: This contains the LED status after a LED put message is sent.

```
{ "type": "led_control", "addr": %s, "data": { "led": %d } }
```

3.2.8.4.2 Web Backend and Web Frontend

After the web backend receives and parses the meter data from “read_info” message, it will broadcast the meter data to the web frontend via web socket port 5000 as defined in app.js.

When users select a node in the list page and go to the detail page, a web client of the selected node will be created. It will listen for broadcast messages from the web backend, and it will compare the received message id with its own address. If it matches with the address, the meter data message is displayed in the detail page. The message is in a JSON format as follows:

```
{"_id": %s,"txCount": %d,"rxCount": %d,"rtt": %d,"electricity": %d, "water": %d, "gas": %d,
"led": %d}
```

3.3 Testing and Results

3.3.1 Test Setup

The universal data concentrator reference design can be built with a TIDA-010032 docking board with AM335x-based BBB and LAUNCHXL-CC1312R1 EVMs. The AM335x runs gateway applications and web application over the AM335x Linux SDK. The CC1312R runs TI 15.4-Stack co-processor over TI-RTOS.

Table 3 summarizes the required hardware and software for the test setup with the data concentrator and the end nodes.

Table 3. Hardware and Software for Test Setup

NODE	HARDWARE	SOFTWARE	COMMENTS
Universal Data Concentrator	TIDA-010032 docking board	NA	Section 3.1.1.1
	BeagleBone Black	gwapp.exe	Section 3.1.2.3.2
		rs485_gwapp.exe	Section 3.1.2.3.3
		webapp	Section 3.1.2.4
	LAUNCHXL-CC1312R1	coprocessor_cc1312r1lp.hex	Section 3.1.2.2
6LoWPAN mesh end-node	LAUNCHXL-CC1312R1	6lowpan_ti_15_4_cc13x2.out	• TIDA-010024 • Code Composer Studio build configuration: debug_poll_demo
RS-485 end-node	TIDM-1005 or RS-485 Half-Duplex Evaluation Module	NA	Need to create a simple software described in Section 3.2.7.1

Table 4 summarizes the TI SDK versions used for testing TIDA-010032 reference design.

Table 4. SDK Version for TIDA-010032

SDK	VERSION
PROCESSOR-SDK-LINUX-AM335X	05.00
SIMPLELINK-CC13X2-SDK	2.30

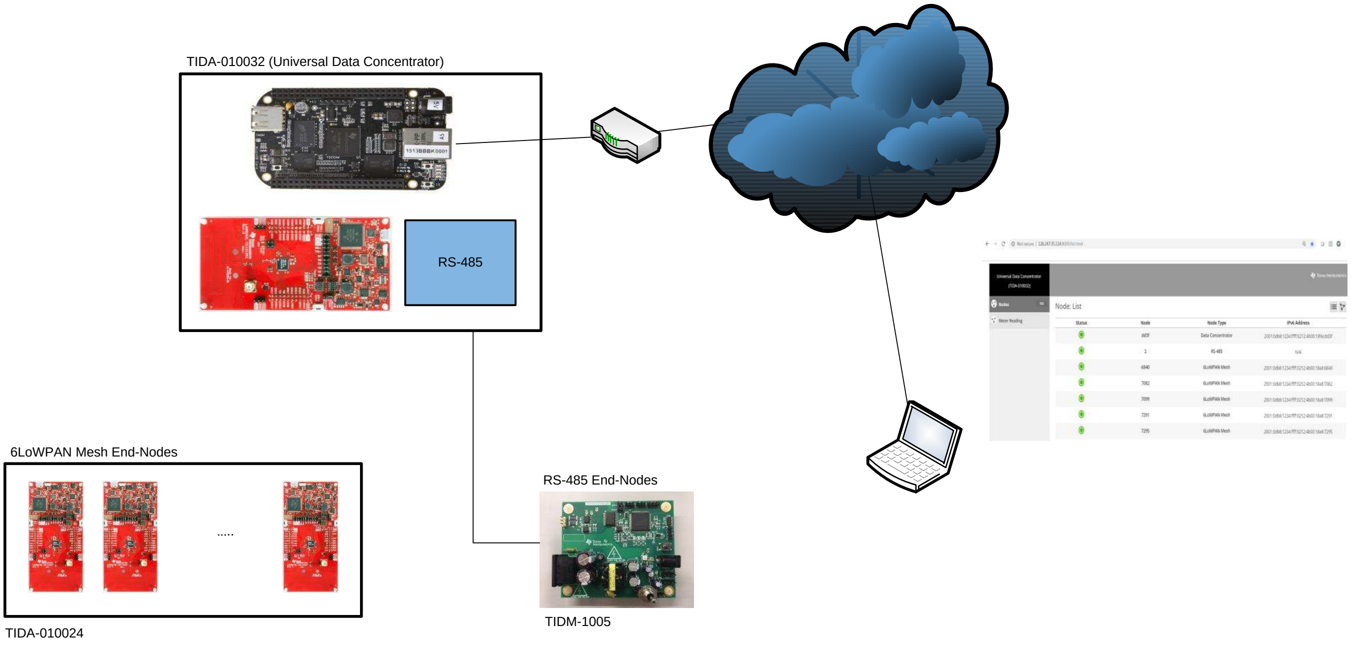
3.3.2 Test Results

The goal of this experiment is to measure delivery ratio performance as well as to verify the software reliability in a large network.

Figure 31 shows the test setup with 100 6LoWPAN mesh end-nodes and one RS-485 end-node. The TIDA-010032 data concentrator is connected to the backbone networks via Ethernet. A PC accessible to the internet can access to the web server to monitor and control the end nodes in the network.

Figure 32 shows the real setup for 100 6LoWPAN mesh end-nodes.

Figure 31. TIDA-010032 Test Setup

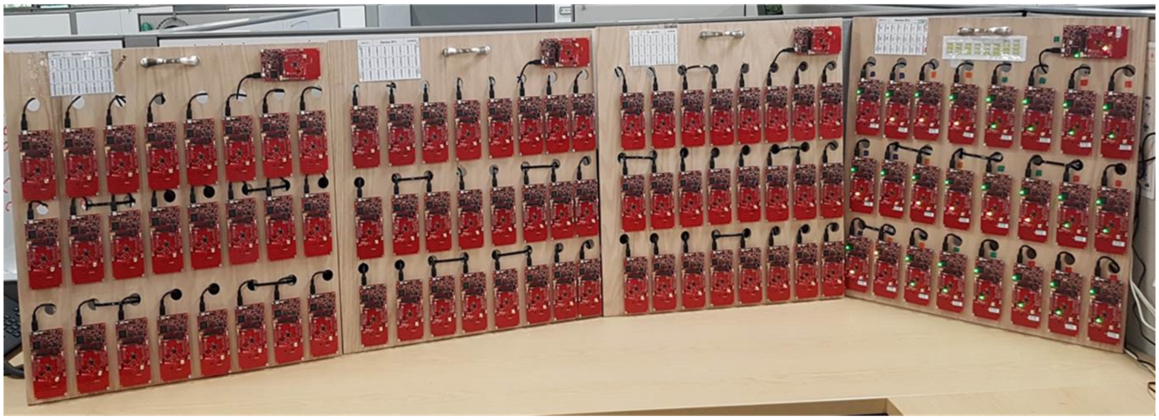


Universal Data Concentrator (TIDA-010032)

Node: List

Status	Node	Node Type	IPv6 Address
●	dd3f	Data Concentrator	2001:0db8:1234:ffff:0212:4b00:189a:dd3f
●	2	RS-485	N/A
●	6840	6LoWPAN Mesh	2001:0db8:1234:ffff:0212:4b00:18a8:6840
●	7082	6LoWPAN Mesh	2001:0db8:1234:ffff:0212:4b00:18a8:7082
●	7099	6LoWPAN Mesh	2001:0db8:1234:ffff:0212:4b00:18a8:7099
●	7291	6LoWPAN Mesh	2001:0db8:1234:ffff:0212:4b00:18a8:7291
●	7295	6LoWPAN Mesh	2001:0db8:1234:ffff:0212:4b00:18a8:7295

Figure 32. 100 6LoWPAN Mesh Network Setup

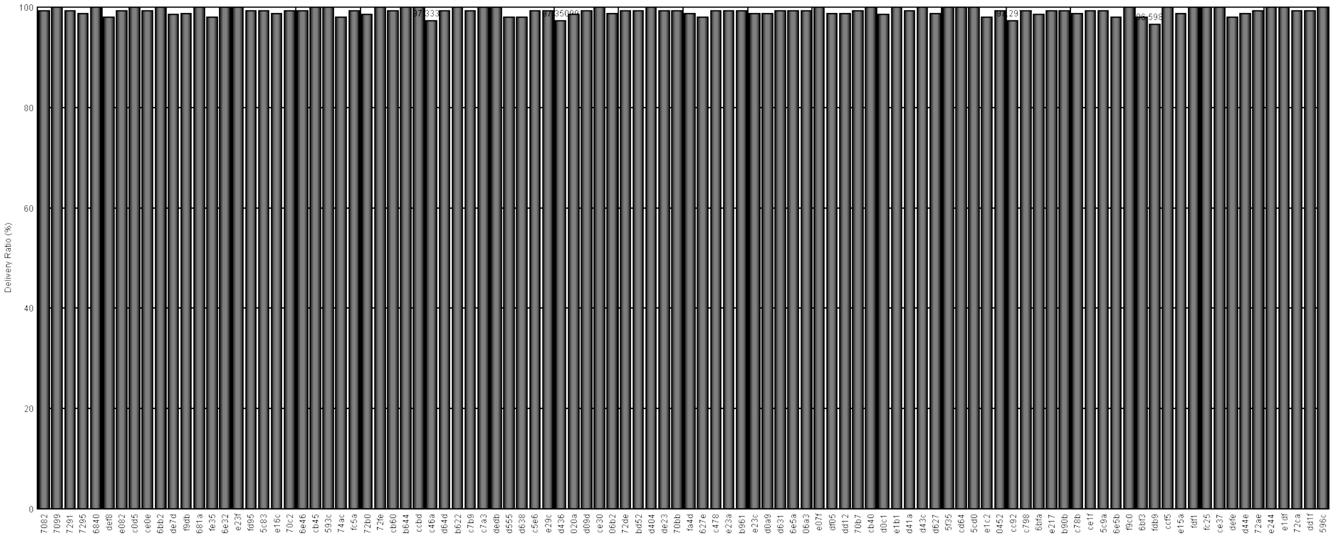


To evaluate the 100-node 6LoWPAN mesh network, the data concentrator polls each node every 5 seconds. The 6B polling message is sent to each end-node and as a return each node responds to the poll message with 11B meter reading data. Figure 28 shows the details of the message format. To verify that the data concentrator supports 6LoWPAN mesh as well as RS-485 connectivity, the RS-485 gateway application ran in parallel with the 6LoWPAN mesh gateway. The experimental data was collected by running the dataFetch.py script located in the directory of <TID-install-directory>/scripts/. To run the script, the following url code line has to be changed based on the IPv4 address of the web server.

```
url = 'http://128.247.35.134:8000/meterData';
```

Figure 33 shows the delivery ratio performance as a function of the MAC address of each node in the 100-node setup. The X-axis shows the MAC address of each 6LoWPAN mesh end node and the Y-axis shows the delivery ratio in percentage. The experimental result shows that all of the nodes achieve outstanding delivery ratio performance. The average delivery ratio with 100 nodes is 99.2%.

Figure 33. Delivery Ratio Performance With 100-Node Setup



To verify the TIDA-010032 software reliability, the network testing with 100 6LoWPAN mesh and one RS-485 end nodes ran for over 10 days. The TIDA-010032 gateway applications with web server ran without performance degradation or software stop issues.

4 Design Files

4.1 Schematics

To download the schematics, see the design files at TIDA-010032.

4.2 Bill of Materials

To download the bill of materials (BOM), see the design files at TIDA-010032.

4.3 PCB Layout Recommendations

4.3.1 Layout Prints

To download the layer plots, see the design files at TIDA-010032.

4.4 Altium Project

To download the Altium Designer® project files, see the design files at TIDA-010032.

4.5 Gerber Files

To download the Gerber files, see the design files at TIDA-010032.

4.6 Assembly Drawings

To download the assembly drawings, see the design files at TIDA-010032.

5 Software Files

To download the software files, see the design files at TIDA-010032.

6 Related Documentation

1. Texas Instruments, *AM335x Sitara™ Processors*
2. Texas Instruments, *CC1312R SimpleLink Sub-1 GHz CC1312R Wireless MCU*
3. Texas Instruments, *THVD1500 5V RS-485 Transceivers up to 500kbps With ±8kV IEC ESD Protection*
4. Texas Instruments, *Secured 6LoWPAN mesh end-node with enhanced network capacity reference design*
5. Texas Instruments, *Simple 6LoWPAN Mesh End-Node Improves Network Performance Reference Design*
6. Texas Instruments, *Data Collector With M-Bus And RS-485 Protocol Conversion Reference Design*

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.

JIANWEI ZHOU is a software engineer at Texas Instruments, where he focus on developing wireless mesh network solution with TI’s low power wireless microcontroller. He received his master degree in electrical engineering from the Texas A&M University, College Station, Texas.

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), 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, 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 (www.ti.com/legal/termsofsale.html) 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.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265