Design Guide: TIDA-010007

Grid IoT Reference Design: Connecting Circuit Breakers and Sensors to Other Equipment Using Wi-Fi®

Description

This reference design showcases integrating Wi-Fi® capability to enhance the connectivity in grid equipment for asset monitoring using the CC3220 or CC3235 SimpleLink™ Wi-Fi and IoT, single-chip wireless MCUs with an integrated network processor and an applications processor. The design enables the capability to setup a 2.4G or 5G Wi-Fi network or connect to an external network, transfer data, and optimize the power. The data transfer scheme includes sending and receiving control, status, settings, and firmware updates over the air between devices or to the cloud. Switching the device between always connected, intermittently connected, and hibernate mode optimizes power consumption for grid application.

Resources

TIDA-010007	Design Folder
CC3220MODA	Product Folder
CC3235SF	Product Folder
TPS62162	Product Folder
OPA4342	Product Folder
TMP116	Product Folder

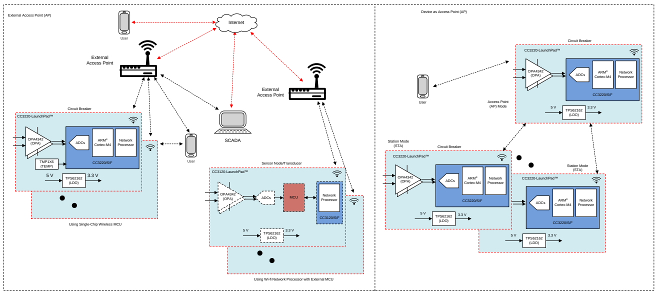
ASK Our E2E™ Experts

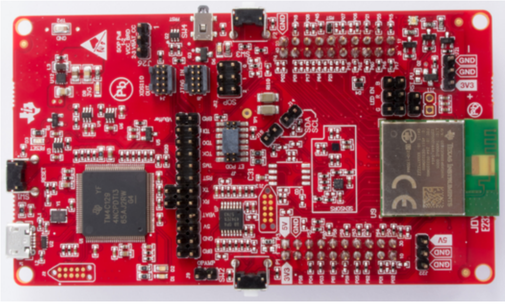
Features

- **Wi-Fi connectivity**: Single-band (CC3220) or dual-band (CC3235) Wi-Fi radio with integrated network and applications processor (Cortex®-M4)
  - CC3235 supports coexistence with BLE in addition to providing dual-band connectivity
  - Chip-level, Wi-Fi Alliance Wi-Fi® CERTIFIED™
  - Connection schemes: Connection to external AP (device as STA) or device as AP
  - Data transfer schemes: Always connected, intermittently connected
- **Power consumption**: Optimized for wired and battery operation
  - Always connected (beacon only): < 700 µA (2.4 GHz) or < 750 µA (5 GHz)
  - Intermittently connected: < 3 mA (5 s), < 2.1 mA (10 s) for 2.4 GHz or < 12.5 mA (5 s), < 5.1 mA (10 s) for 5 GHz
  - Device low power mode: < 10 µA (Hibernate), < 150 µA (LPDS), 20 mA peak
- **Security features**: 128-bit unique device ID, secure boot of MCU image, internal HTTPS server for AP provisioning, MQTT over TLS, secure over-the-air (OTA) update

Applications

- Circuit breakers: Residential breakers (MCB, AFCI, GFCI), Circuit breaker (ACB, MCCB, VCB)
- Renewable energy: PV inverter, EV charging
- Building automation and Factory automation equipment





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

To meet the future power demands, the electric grid needs modernization to improve reliability and reduce downtime. Predictive maintenance not only allows utilities and substations to monitor the health of expensive equipment and take action prior to system breakdown but also allows better load and source management by monitoring demand variation in real-time. With advancement in technology, grid modernization is lagging behind in terms of incorporating advanced connecting capabilities between equipment and operators (or end consumers) which improves the overall network reliability and resiliency. Steps have been taken to move towards digital grid where sensors are localized with data being processed (data concentrators) in a central location.

One of the key challenges in moving towards digitizing electric grid includes integrating safe and reliable wireless connectivity that is interoperable. For such a system, security is of high importance as unscrupulous users should not have access and control. Also standalone systems comprised of miniature sensors with wireless connectivity must have the means to scale power consumption as they are getting localized and efficient.

Asset management for grid

The Supervisory Control and Data Acquisition (SCADA) has been adopted and standardized over the past couple of decades to collect the information and control various end equipment in power generation, transmission, and distribution in generating stations and substations. Connectivity between equipment is achieved through wired communication or is lacking in some cases. Grid Internet of Things (GIoT) is an ever-growing system of computers, machines, or objects; all networked together with unique identifiers and sharing information within the network with limited human interaction. Communication between equipment, data concentrator, and SCADA is now being extended to the next level to connect to a sensor network of smart devices which can now be connected through multiple wireless technologies. An added benefit is improving reliability through adding a redundant data communication path.

- Systems, applications, and users can seamlessly share data and monitor and control equipment remotely through the network.
- On-demand asset health monitoring allows users to track equipment aging and to predict faults before they occur. This saves cost and expensive assets by scheduling maintenance before a grid breakdown.

Adding wireless connectivity

Multiple wireless communication technologies are available including near-field communication (NFC), Bluetooth® LE (BLE), ZigBee®, sub-1 GHz, Wi-Fi, and so forth. The selection of suitable technology for any equipment is dependent on the amount of data transferred, distance, number of nodes, power that is available, and response time. The TIDA-00816 TI Design shows integrating sub-1 GHz RF for short range (less than 1 km) while consuming very low power. This is useful when a limited amount of data needs to be transferred.

Wi-Fi offers an alternate solution for substations and residential breakers. Substations equipped with wireless communications are less susceptible to being disabled, enhancing the reliability of on-site sensors, security cameras, and other alert systems. [1]Wireless can also be used to back up traditional communications such as optic cables or ethernet and create system redundancy when wired communications are disrupted. It is also possible to retrofit the existing substation with wireless connectivity without investing on additional cables while providing redundancy.

Wi-Fi is based on the IEEE 802.11 standard developed as a wireless replacement for the wired IEEE 802.3 Ethernet standard. This standard is governed by the Wi-Fi Alliance which ensures interoperability across products and vendors providing native security protocols for a fast and secure wireless connection to the internet. [2]

[1] **Physical attack: Metcalf substation:** In the early hours of April 16, 2013, gunmen severely damaged 17 large power transformers at a major northern California sub-station. The attack was well-planned; the perpetrators scouted firing locations, targeted critical equipment, and cut fiber optic communication lines to disable on-site security and automation systems before the shooting began. The attack required several weeks and $15 million of repairs to correct, with only a fortunate series of factors keeping it from wreaking widespread and long lasting outages.

[2] "NOJA Power recently completed a project for the Ampla Electricity Utility in Brazil with an integration solution using optical fiber connections and Wi-Fi system. In addition, the users are able to connect (via cable or Wi-Fi) to the local area network to configure, download events and command the ACRs through an IP address configured in each device. An advantage of Wi-Fi is the possibility of using the recloser application, available for Android and Apple devices, which allows technicians to connect to the substation’s wireless network and control and monitor the devices from a smartphone and/or tablet in a practical, fast and reliable fashion."

The TIDA-010007 design connects industrial or residential equipment to the cloud through the CC3220SF (SimpleLink Wi-Fi wireless MCU) or CC3235SF (dual band SimpleLink Wi-Fi wireless MCU).

This reference design showcases:

- Two architectures to incorporate Wi-Fi capability that connects multiple grid equipment and to share data securely
- Setting up of network and transferring of data (control, status, settings, updates) between equipment, remote handheld device or cloud, or both, using secured socket connections
- Optimizing power consumption for multiple modes and data transfer schemes for efficient power source design

1.1 Circuit Breakers

A circuit breaker interrupts the flow of current in the event of any faults. A breaker reset is done manually to resume regular operation by attending to the breaker. By enabling Wi-Fi connectivity, a circuit breaker can now be reset by an authorized operator remotely (for example: an end user using their phone). Also, additional capabilities can now be integrated in a circuit breaker by monitoring the energy flow by taking it one step closer to smart home and building automation.

Adding Wi-Fi connectivity to industrial breakers simplifies the maintenance of a large number of breakers located in different corners of the building and identifying the exact fault location quickly. Sharing energy flow and sensor data through wireless connectivity can improve management of resources and early prediction of degradation. It also allows the settings of industrial breakers to be changed remotely by an authorized operator.

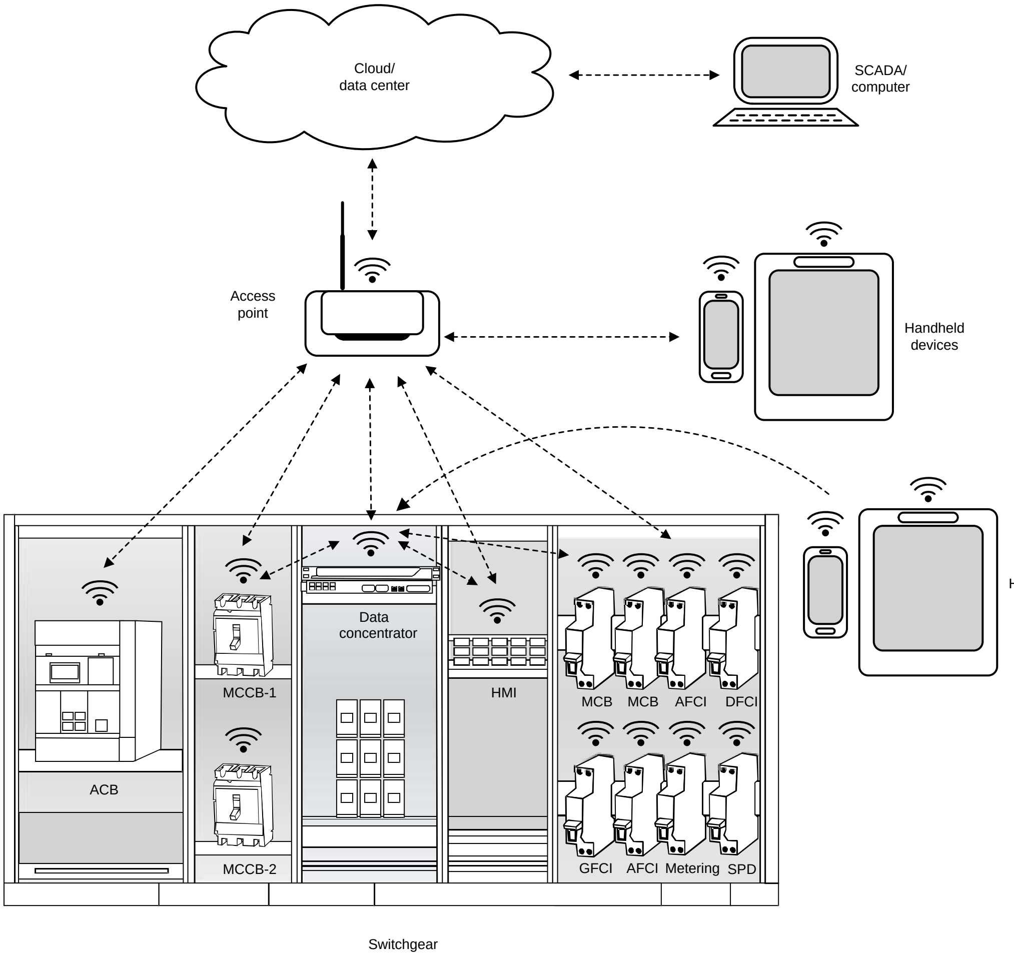


Figure 1. Connected Circuit Breakers

1.2 Substation Automation

A substation is comprised of multiple monitoring and protection equipment including protection relays, circuit breakers, merging units, terminal units, automation controllers, data concentrators, intelligent electronic devices (IED), transducers, and so forth. Substations can also include expensive assets such as transformers, generators, switchgear, busbars, motors, and so forth. These are interconnected through serial wired interface such as RS-232, RS-485, and Ethernet, for example. By enabling Wi-Fi connectivity between equipment modernizing of substation is easier as one scale sensor nodes quickly. This approach also is less expensive as compared to a traditional solution because no new wiring is required. Also, access to real-time asset health monitoring becomes easier since the operator can initiate this process remotely, and when required.

1.3 Key System Specifications

Table 1. Key System Specifications

PARAMETER	SPECIFICATIONS	DETAILS
Wi-Fi connectivity	Mode 1: Connecting equipment (STA) to external access point (External AP)	Section 2.4.1.1
	Mode 2: Connecting equipment (STA) to the master equipment (AP) in the network	Section 2.4.1.2
	Provisioning 1: Access-point provisioning	Section 2.4.4.1
	Provisioning 2: SmartConfig provisioning	Section 2.4.4.2
Data transfer scheme	Control signal: Breaker control (ON and OFF)	Section 2.4.5
	Data out: Time synchronization signal	
	Data In: Breaker status, voltage, current, temperature	
	Interrupt signal: Breaker fault detection	
	Firmware update: Over the air (OTA) update	Section 2.4.7
User interface	Device with Wi-Fi interface (Example: Smartphone, tablet, PC)	
Power consumption in 2.4-GHz mode (CC3220 and CC3235)	Always connected mode: < 700 µA (Beacon only)	Section 3.2.2
	Intermittently connected mode: < 3 mA (5 s interval), < 1.5 mA (10 s interval)	
	Device level modes: Hibernate: <10 µA, LPDS: < 150 µA (<20 mA peak), TX: 170 mA - 280 mA, RX: 70 mA	
Power consumption in 5-GHz mode (CC3235)	Always connected mode: < 750 µA (Beacon only)	
	Intermittently connected mode: <7 mA (5-s interval), < 12.5 mA (10-s interval)	
	Device level modes: Hibernate: <10 µA, LPDS: < 110 µA (< 10-mA peak)	
Sensor measurement	Temperature: –40°C to 125°C at ±1°C accuracy	Section 3.1.2.2
	ADC: 12-bit, 80 samples per cycle (4 ksps)	
DC power supply	5-V USB	Section 3.1.1

2 System Overview

2.1 Block Diagram

The block diagram in Figure 2 showcases two architectures to connect grid equipment and different ways of interfacing between the equipment and users.

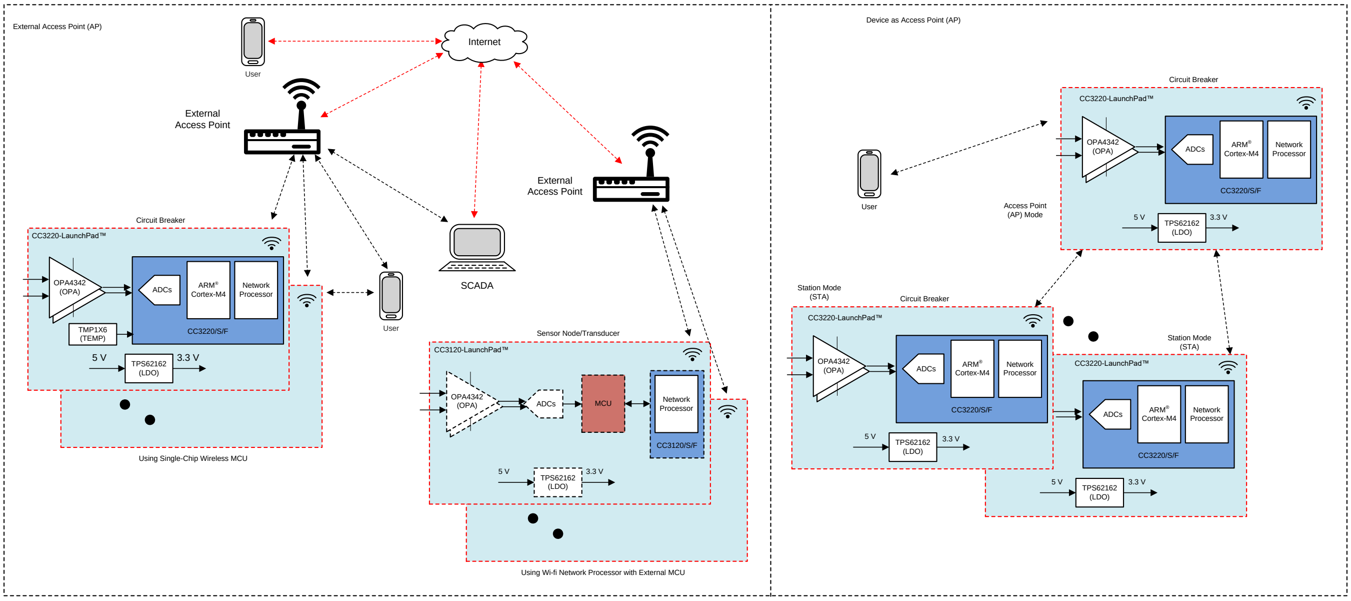


Figure 2. TIDA-010007 Block Diagram

From the hardware perspective, there are two methods of implementation.

- A single-chip solution using the CC3220 or CC3235 device for both the user application and a network processor as shown in the block diagram for the circuit breaker use case.
- If there is a need to retrofit the existing design or if high-end MCU or data acquisition is required, then external data acquisition can be used along with the network processor CC3120 or CC3135 device to add connectivity to the cloud.

NOTE: The CC3120 or CC3135 device contains only the networking subsystem and is driven by an external MCU host. The CC3220 or CC3235 device contains the same networking subsystem along with an internal MCU application processor.

This reference design showcases the following connectivity and data transfer schemes:

- Handheld device can be connected directly to the Wi-Fi device for provisioning
- Once the device is connected to the network, the device can communicate with the handheld device that is connected either to the same AP or through cloud access
- Multiple breakers connected to the same AP can also talk to each other and communicate the data or command
- SCADA connecting to the device through either local AP or through cloud

2.2 Design Considerations

Some of the key considerations for design of the TIDA-010007 are:

- Provisioning methods for easy and secured configuration of the network or device
- Transferring data in different modes between multiple devices and user interfaces such as handheld device, SCADA
- Optimizing power consumption for the given use-case
- Sampling analog inputs from the external sensors and transfer the data on-demand to cloud. For AC voltage and current, some of the applications require calculating root mean square (RMS) values and sending these values on a regular time interval

This is a generic design implementation that can be modified depending on the type of end equipment in the grid and suitable architecture is selected based on the power consumption for the corresponding data transfer schemes.

2.3 Highlighted Products

This section provides details of some of the TI products used in this TI design.

2.3.1 CC3220xx and CC3235x

The CC3220x device is part of the SimpleLink MCU platform, which consists of Wi-Fi, low-energy, sub-1 GHz, and host MCUs that all share a common, easy-to-use development environment, with a single-core SDK and rich tool set. The CC3220x SimpleLink Wi-Fi wireless MCU SoC is a single-chip with two separate execution environments: a user application dedicated ARM Cortex-M4 MCU, and a network processor MCU. Using this Wi-Fi CERTIFIED single-chip MCU with built-in Wi-Fi connectivity, it is easy to implement IoT design.

The CC3235x family offers the same features as CC3220x with the added capability of dual band Wi-Fi connectivity for 2.4 GHz and 5 GHz networks and supports BLE coexistence.

The CC3220/35x wireless MCU family is a part of the Internet-on-a-chip™ family of solutions from TI. This generation introduces new features and capabilities that further simplify the connectivity of devices to the internet. The new capabilities including the following:

- Enhanced Wi-Fi provisioning
- Enhanced power consumption
- Enhanced file system security (supported only by the CC3220/35S and CC3220/35SF devices)
- Wi-Fi AP connection with up to four stations
- IPv6
- More concurrently opened BSD sockets: up to 16 BSD sockets, of which 6 are secure
- HTTPS support RESTful API support
- Asymmetric keys crypto library

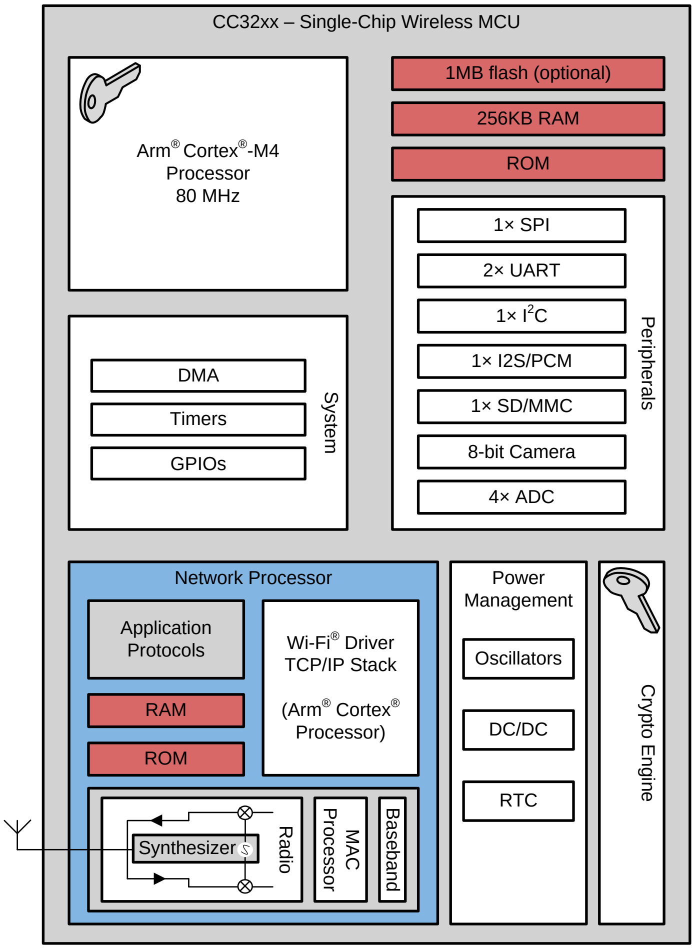


Figure 3. CC3220X Hardware Overview

The CC3220/35 wireless MCU family supports the following modes: station, AP, and Wi-Fi direct. The device also supports WPA2 personal and enterprise security. This subsystem includes embedded TCP/IP and TLS/SSL stacks, the HTTP server, and multiple Internet protocols. The device supports a variety of Wi-Fi provisioning methods including HTTP based on AP mode, SmartConfig Technology, and WPS2.0. The device includes a wide variety of peripherals, including a fast parallel-camera interface, I2S, SD, UART, SPI, I2C, and 4-channel ADCs.

The SimpleLink CC32xx family of devices is available in three different variants: CC3220/35R, CC3220/35S, and CC3220/35SF. The CC3220/35xR and CC3220/35S devices include 256KB of application-dedicated, embedded RAM for code and data, ROM with an external serial-flash bootloader, and peripheral drivers. The CC3220/35SF device includes an application-dedicated 1MB of XIP flash and 256KB of RAM for code and data, ROM with an external serial-flash bootloader, and peripheral drivers. The CC3220/35S and CC3220/35SF device options have additional security features, such as (1) encrypted and authenticated file systems, (2) user IP encryption and authentication, (3) secured boot (authentication and integrity validation of the application image at flash and boot time), and more.

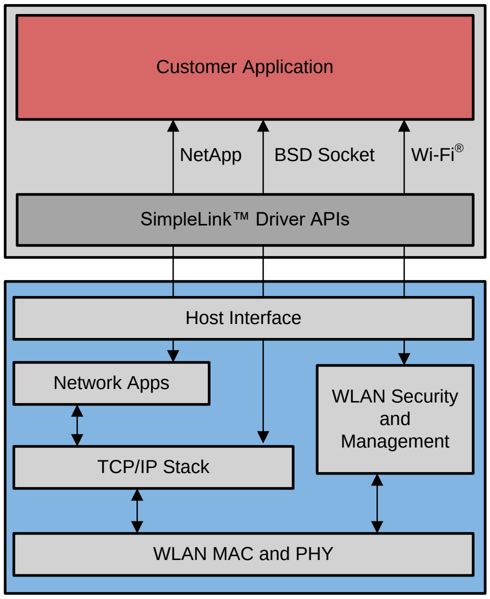


Figure 4. CC3200x Embedded Software Overview

2.3.2 OPA4342

For conditioning the analog input signal from sensors, the OPA4342 device is chosen which is optimized for single-supply operation (3.3-V supply). Rail-to-rail input/output and high-speed operation make them ideal for driving sampling analog-to-digital converters (ADC) available on the CC3220/35.

2.3.3 TPS62162

The TPS62162 device is synchronous step-down DC-DC converter optimized for smaller footprint and high power density. With its wide operating input voltage range of up to 17 V, the devices are ideally suited for systems powered from either a Li-Ion or other battery as well as from 12-V intermediate power rails. A high switching frequency of typically 2.25 MHz allows the use of small inductors and provides fast transient response as well as high output voltage accuracy by utilization of the DCS-Control™ topology to power up the CC3220/35.

2.3.4 TMP116

The TMP116 (TMP116, TMP116N) is a family of low power, high-precision temperature sensors with integrated EEPROM memory. The TMP116 device provides a 16-bit temperature result with a resolution of 0.0078°C and an accuracy of up to ±0.2°C with no calibration. The TMP116 is I2C and SM-Bus interface compatible, has programmable alert functionality, and can support up to four devices on a single bus. Across the device operating temperature range of –55°C to 125°C, the TMP116 exceeds the accuracy of a class A RTD, while consuming less than one fifth of the typical excitation current for a PT100 RTD. The TMP116 is easier to use than RTDs, eliminating the need for calibration, external circuitry, matched traces, and Kelvin connections.

2.4 System Design Theory

2.4.1 Architecture

There are two architectures possible using the CC3220/35 Wi-Fi device depending on the end equipment in grid and connectivity requirement:

- Connecting equipment (STA) to external access point (External AP)
- Connecting equipment (STA) to the master equipment (AP) in the network

2.4.1.1 Cloud Connectivity Through External AP

In this architecture, all the equipment with Wi-Fi devices are configured in STA mode connecting to an available external AP. The Wi-Fi device is connected to the internet through an external AP. Through cloud connection, devices can send or receive messages from remote devices such as handheld devices (cellphone, tablets, and so forth) or computers which are either connected to the same AP or internet as Figure 5 shows. Depending on the hardware capability of the external AP, at least hundreds of devices can be connected to a single AP.

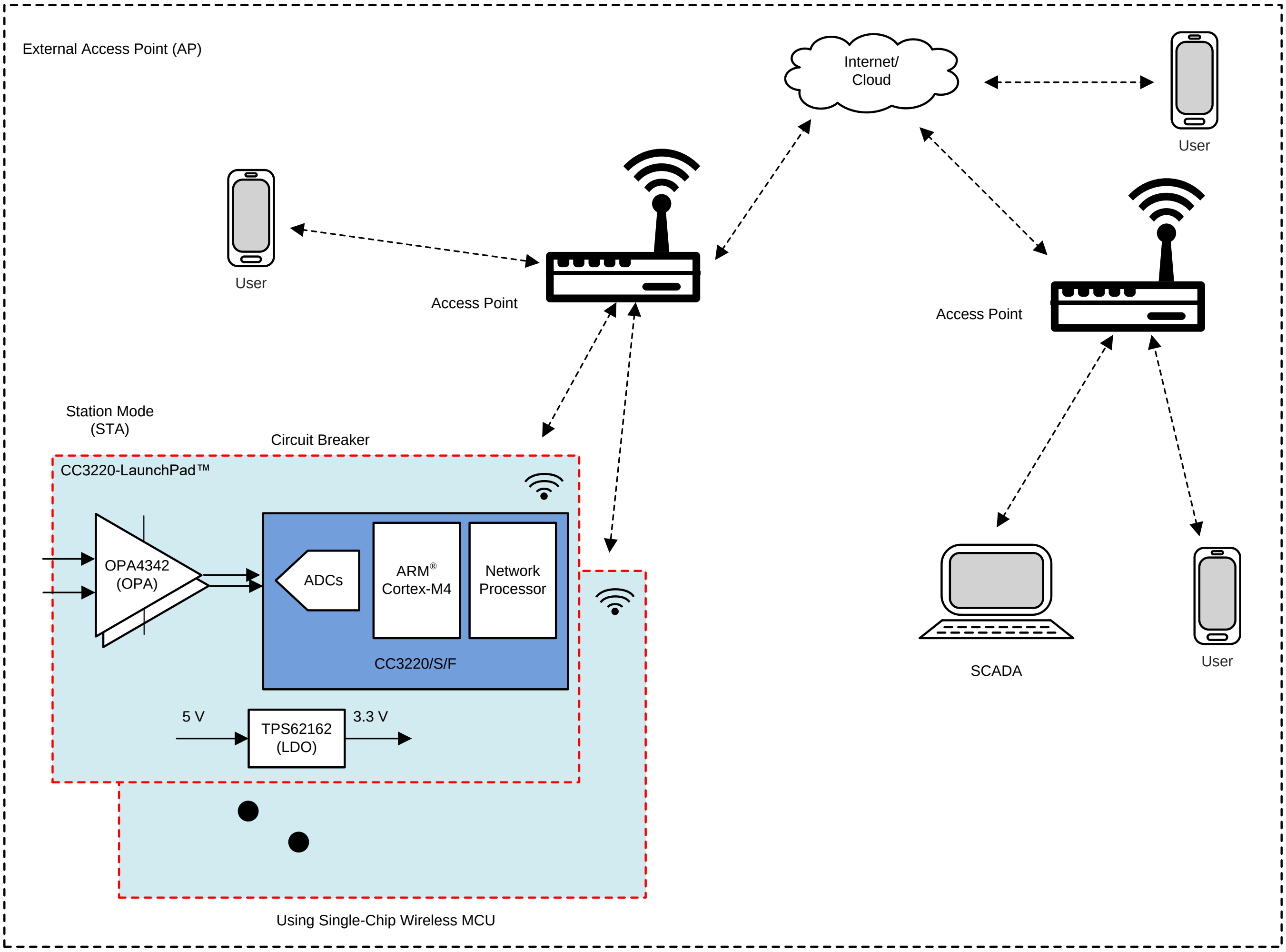


Figure 5. Wi-Fi® Connectivity Through External Access Point

2.4.1.2 SimpleLink™ AP

In some of the applications where an internet connection (that is, external AP) is unavailable, devices need to setup their own network and talk to each other to share the data between them. In such applications, one of the SimpleLink devices can be configured as the AP of the network and other devices can be connected to this network in STA mode as Figure 6 shows.

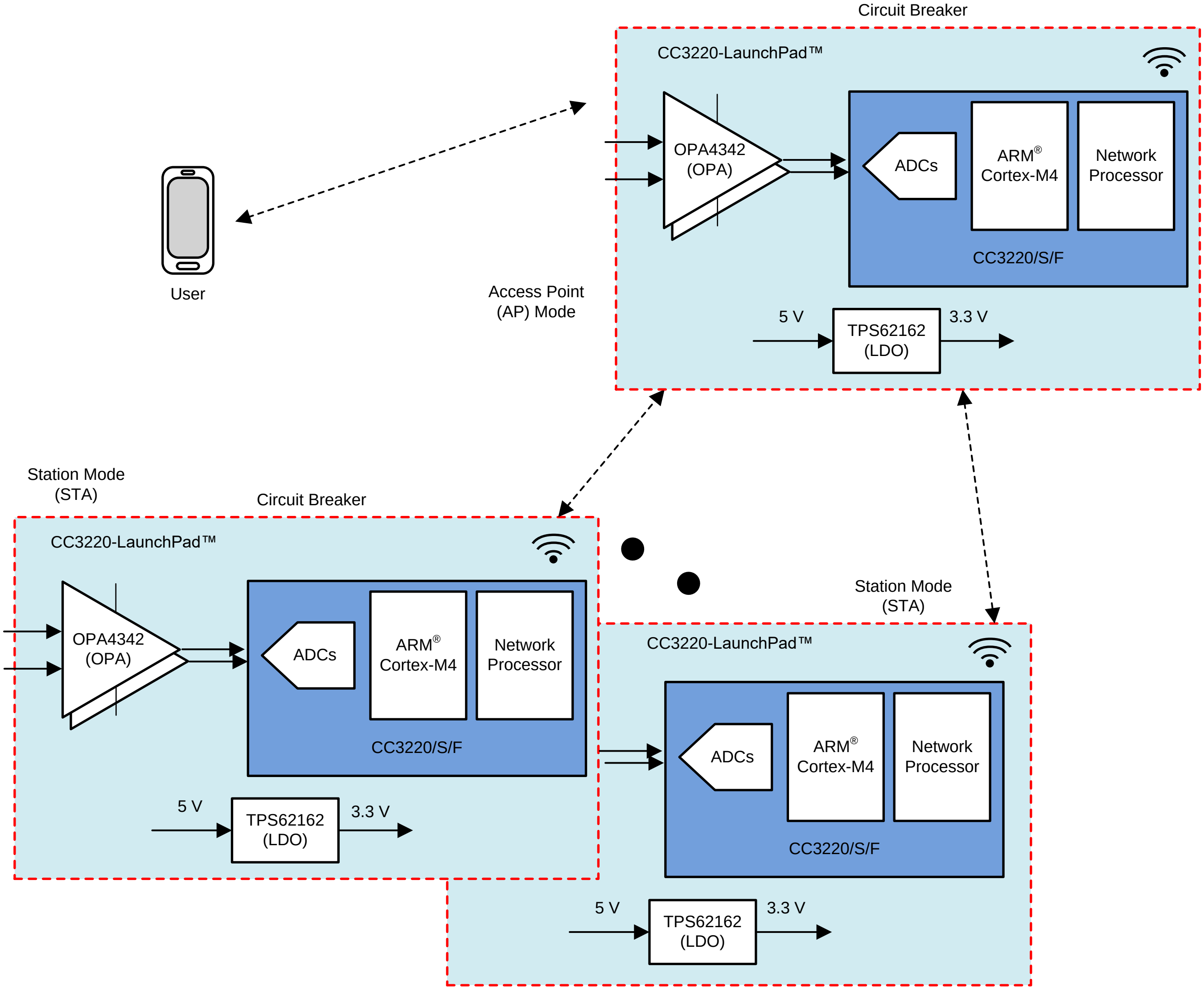


Figure 6. SimpleLink™ AP Architecture

2.4.2 Hardware

For adding Wi-Fi capability to any equipment has to be accomplished by including a network processor to run all Wi-Fi and internet logical layers. This can be implemented using either one of the following chipset solutions:

- **Discrete solution**: Wi-Fi wireless network processor, CC3120/35 consisting of power management subsystems along with a wireless network processor (NWP) can be used to add a connectivity solution to an external application microcontroller unit. For applications where a high-end processor or external ADC (analog front-end) with higher resolution or sampling rate is necessary, integrating the CC3120/35 device into the design is ideal. This is a perfect fit for retrofitting Wi-Fi connectivity to any grid equipment which has a host MCU.
- **Integrated solution**: Single-chip wireless MCU, CC3220/35 offers integration of two physically separated on-chip MCUs. In addition to NWP (same as CC3120/35), it has an ARM Cortex-M4 MCU-based application processor running at 80 MHz with a user-dedicated 256KB of RAM, and an optional 1MB of XIP flash.

2.4.3 Data Transfer Schemes

Two types of architectures and criteria for selection of Wi-Fi wireless MCU were discussed in earlier sections. Along with these two, it is equally important to evaluate the data transfer schemes necessary for the application which decides the overall power consumption assisting in budgeting power supply design for the Wi-Fi connectivity. Equipment across multiple grid applications can implement data transfer protocol suitable for that application. Based on the power consumption values, the following data transfer schemes are listed:

1. Status update with limited health update: For applications where basic parameters such as breaker status, average power consumption or number of events (surge, overvoltage, overloading or fault) over a span of time, needs to be sent occasionally, for example once a day or so, then average power consumption from the network processor is going to be insignificantly low. Since Wi-Fi has a very high data transfer capability, the device needs to be in active mode for a shorter proportion of the time interval while remaining in hibernate mode for the rest.
2. Status update with frequent health update: In some of the use cases, it is required to update the health of the system regularly and more frequently. These are a few of the applications where parameters that are being observed are expected to be dynamic and fluctuating. During normal operation of the system, timer with fixed frequency can be set to trigger the NWP to wake up from hibernate mode and start updating data such as peak or RMS values voltage/current, and so forth. It is also possible to monitor parameters such as voltage, current, or temperature and start sending updates whenever these parameters go beyond threshold levels to inform any remote operator. In addition to regular updates, this mode has ability to transfer data and updates asynchronously.
3. Network of multiple breakers: In applications where an external AP is not available, multiple (up to four) equipment can be connected to the master equipment (this could be similar to one in station mode or could be data concentrator). In this type of use case, applications may demand all the devices to be connected with each other always.
4. Combination of 2 and 3: In this mode, the equipment can be in hibernate mode for most of the duration and it can wake up based on detection of any hardware activity. Upon waking up from sleep, the equipment can build a connection with the network AP or its peers to start sending or receiving messages. The system can go back to hibernation mode if there is no activity for a specified time interval.

2.4.4 Provisioning

Wi-Fi provisioning is the process of providing an IoT device the information needed to connect to a wireless network for the first time such as network name or SSID, password, and so forth. Providing this information may be challenging, because not all IoT devices are equipped with conventional input peripherals such as keyboards or displays, and so forth. At the same time, it is important to secure the connection between the device sending the network credentials and the CC3220/35 device during AP provisioning. To protect the network credentials and ensure a user does not unknowingly transmit them to an unwanted device, this design uses WPA2 authentication when a station connects to the CC3220/35 AP.

In a SimpleLink Wi-Fi CC3220/35 device, end-users can configure the device wirelessly using a smartphone or tablet running a dedicated provisioning application. The provisioning capabilities can be easily embedded by developers on their own wireless applications. There are two methods by which an IoT device can be provisioned as described in Section 2.4.4.1 and Section 2.4.4.2.

2.4.4.1 AP Provisioning

The SimpleLink Wi-Fi device creates a wireless network of its own with a predefined network name, letting the user connect it with an external device (such as a smartphone, tablet, or PC) and add a profile through the internal HTTP web server. In AP mode, the unprovisioned SimpleLink Wi-Fi device wakes up initially as an AP with an SSID defined by the equipment manufacturer. Before trying to connect to the home network for the first time, the unprovisioned device creates a network of its own, allowing a PC or a smartphone to connect to it directly and facilitate its initial configuration. The mobile application connects to the device as a Wi-Fi station, and sends the configurations. By using this mode, the user should know which device to connect to according to its published SSID, while acting on the AP role. For security purposes, WPA2 authentication is used to connect a station to the AP network setup by the device.

2.4.4.2 SmartConfig™ Provisioning

SmartConfig is a proprietary provisioning method from TI that uses a smartphone or tablet to broadcast network credentials to an unprovisioned TI Wi-Fi device. The device can scan for SmartConfig broadcasts while operating in station mode or AP mode. Using the SmartConfig algorithm eliminates the need to know the device identity ahead of time. The process configures any listening device. Using this method, the user does not need to know the device name, or any other device identity.

In Figure 7, a profile is configured using SmartConfig. The provisioned device connects to the wireless network from the configured profile and waits for the smartphone application to contact its HTTP web server. When the confirmation result is delivered to the smartphone application, the device sends the successful result to the host, and stops the process.

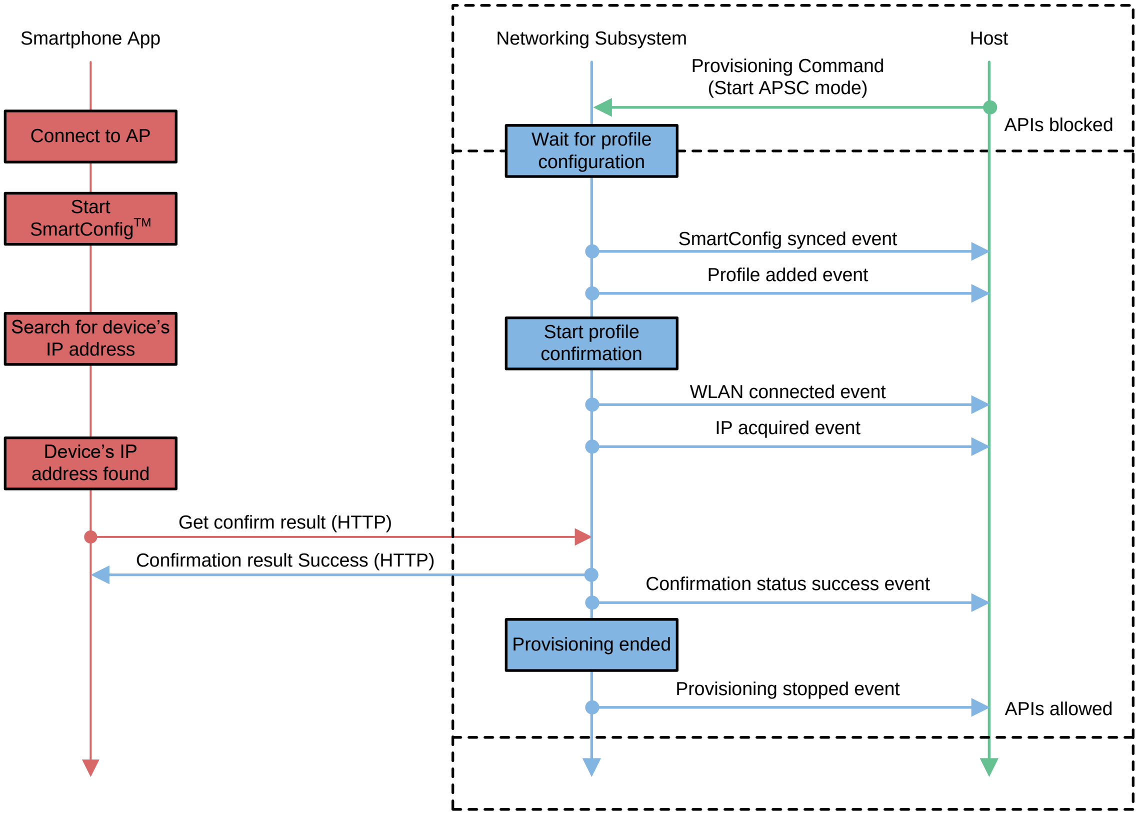

Smartphone App
Networking Subsystem
Host
Provisioning Command (Start APSC mode)
APIs blocked
Connect to AP
Wait for profile configuration
Start SmartConfig™
SmartConfig synced event
Profile added event
Search for device's IP address
Start profile confirmation
WLAN connected event
IP acquired event
Device's IP address found
Get confirm result (HTTP)
Confirmation result Success (HTTP)
Confirmation status success event
Provisioning ended
Provisioning stopped event
APIs allowed

Figure 7. Timing Diagram for Successful SmartConfig™ Provisioning

2.4.5 Data Transfer, Communication, and Sending and Receiving Messages

Once the Wi-Fi device is successfully connected to a network with cloud access, the device uses the Message Queue Telemetry Transport (MQTT) protocol to send messages to and receive messages from the cloud. This could be either a message or a data or parameter for monitoring the health of the asset. Sending and receiving messages through the cloud makes it possible for the system to be controlled and monitored from anywhere, as long as a user has access to the internet. To control the system, a user can send a message to the device through the cloud server from another client, such as a smartphone, tablet, or PC (SCADA). When the cloud server receives a message destined for a specific end device, the server forwards the message onto that particular device. This reference design also showcases status and data updates from the devices to clients (smartphone or tablet) that request them through the cloud server.

In the TIDA-010007 design, the basic requirement for the circuit breaker is to control the position of the breaker through the cloud and update the status to the remote user. In the event of any fault, the breaker should notify the opening of the breaker to all the clients subscribed for the corresponding breaker. In some of the applications, synchronizing multiple Wi-Fi devices with real time clock (RTC) is required. The health of the grid end-equipment can be monitored by sending parameters such as grid voltage, current, and temperature over to the cloud on request. Data could be periodic RMS value or raw data depending on the type of application and situation. Details of the data transfer schemes using MQTT are covered in Section 2.4.5.1 through Section 2.4.5.3.

2.4.5.1 Message Queue Telemetry Transport (MQTT) Protocol

MQTT is a light weight machine-to-machine (M2M), "Internet of Things" connectivity protocol. MQTT is based on the publish or subscribe messaging transport model, and it is designed to be used on the top of the TCP/IP protocol. The key benefits of MQTT are that it requires a small memory footprint and low network bandwidth. MQTT also provides fast response time and ease of scalability for applications with a low-power requirement. These features make it an ideal communication protocol for low power, embedded connectivity solutions such as circuit breakers and other health monitoring systems in the grid.

2.4.5.2 Message or Data From User to Device (Subscription Topics)

Subscription topics are the ones where the user can send message or data to the CC3220 or CC3235 device through the cloud. In the topic names, a unique id can be defined that indicates a specific single (or group of) device. These topics could be either sending inputs to the breaker or user requesting for data/feedback from the device. In this TI design, few example subscription topics are demonstrated. However, this list is not limited the following topics. Users can easily add subscription topics depending on the application use case.

Subscription Topics (from user to device):

- **Breaker Control_ON**: Using this topic, the user can turn-on the breaker from a remote location using any handheld device. When the breaker is open, it is possible to close the breaker remotely using this function. In applications other than the breaker, this could be activating some external hardware through the GPIO pins of the CC3220 or CC3235 device.
- **Breaker Control_OFF**: In case of any abnormal conditions or a user-defined event where the breaker needs to be de-asserted, the user can send a message to this subscription topic which corresponds to the corresponding device connected to the MQTT server.
- **RTC update**: Time stamping of the events and also the data from different devices is critical in many applications. Using this subscription topic, all the connected devices can update the clock synchronization by updating the device current local time by global NTP servers.
- **OTA update**: Through this topic, the user can trigger the firmware update on the Wi-Fi device using over-the-air update which performs an update of a full image over a secured channel.
- **Breaker Status Request**: Users can access the status of the breaker from anywhere at any time by connecting to the cloud and by subscribing to this topic. This can also be extended to request other parameters of the grid equipment such as breaker settings, number of surge events in surge protection device (SPD), power consumption, and an arc event in the arc fault circuit interrupter (AFCI), and so forth.
- **Temperature request**: The user can request temperature data from the device for monitoring its health. Again, only temperature data is shown in this design. However, this could be any other physical parameters like ambient light, humidity, and so forth, depending on the type of sensors interfaced with the Wi-Fi device.
- **Start periodic data:** For some of the applications like a health monitoring system, it is required to have periodic data being sent to cloud on a regular basis. Using this subscription topic, the user can connect to the network and trigger the device to start sending periodic data. The user can also specify the data capture and transmitting interval.
- **Stop periodic data**: If the user wants to stop receiving the periodic data, settings can be changed using this command to stop the device from capturing regular data and sending them through Wi-Fi.

2.4.5.3 Message or Data from Device to User (Publish Topics)

The device can send the message and data to any user connected to the internet and subscribed to the corresponding topic. This is implemented by publishing the data using MQTT which includes the topic and data as coded in the CC3220 or CC3235 device. It is possible to have multiple publish topics tuned for a specific application. These push notifications can be triggered by events such as hardware interrupts, software triggers such as based on signal levels in the ADC inputs, or timer triggered, and so forth. In this design, examples of publish topics are defined and implemented which could be either event-based transfer or based on the user request.

Publish Topics (from device to user interface):

- **Breaker OPEN**: This event is triggered when there is a fault and breaker opens to disconnect the power. A notification is sent to all the subscribed users upon any fault. It is also possible to append the time at the fault and the current or voltage data before the occurrence of fault. For some of the applications where the end equipment has limited backup power, it has to send the data within a specific time before the battery drains. In this scenario, it is recommended to send the data packets along with the breaker open notification. Time stamping of the fault events in multiple breakers helps in analyzing the cause of the fault.
- **Breaker Status (upon request)**: The breaker can update its status upon request from an authorized user. Using a handheld device, a user from a remote location can come to know the position of the breaker.
- **V, I measurement**: If the user wants to know more about the energy consumption or electrical parameter such as voltage and current through a specific breaker, then the RMS values of both voltage and current can be sent to the cloud through Wi-Fi. This could either be on a periodic basis or upon request. Using an internal timer base, it is possible to send the RMS values on a periodic basis and update the values. The time period of the data can be set by the user, depending on the type of application and use case.
- **Breaker status feedback**: This is acknowledgment back to the user when the user controls the breaker. Unlike breaker status, this message is sent to the user after closing or opening of the breaker after receiving a control command through Wi-Fi. This helps the user know that the control action has been executed.
- **Temperature**: Similar to voltage and current, the health of the equipment can be weighed by measuring the temperature within the equipment.

2.4.6 Power Optimization

This section describes how to optimize the power consumption depending on the type of data transfer schemes. By utilizing low power modes available in the CC3220 or CC3235 device, it is possible to optimize the average power consumption for the given use case. Overall power consumption depends on the amount of data transferred or received, latency in its response and the frequency of activity. The network processor subsystem of the CC3220/35S uses a policy called Long Sleep Interval (LSI) to increase the amount of time the Wi-Fi NWP spends in a low-power mode between AP beacons and broadcasts, which can significantly reduce the average power consumption.

2.4.6.1 Low Power Modes

The networking subsystems in the CC3220 and CC3235 devices have configurable power polices which affect the power consumption of the devices while in an idle state, which includes both hibernate mode and LPDS mode.

Hibernate mode: In this mode, the device enters immediately to hibernate which represents the lowest power state of the device. In this mode, all the voltage sources, like a DC/DC converter or LDOs, within the power management unit are shut off. Very few logic devices, which work directly on battery power are ON and they work on a 32-kHz clock. It takes at least 15 ms to up to 250 ms to wake up from the hibernate mode.

Low Power Deep Sleep (LPDS): In this mode, each subsystem processor requests the clock management unit for shutting off their subsystem. When both the subsystems request for this mode, the clock management unit will turn off the PLL, 40-MHz XTAL, and the power management unit will shut off the power to each subsystem and scale down the voltage of the always-on domain to 0.9 V. Active logic in this mode will work on a 32-kHz clock. Typical current consumption in this mode is 120 µA–140 µA. The device stays in low power deep-sleep mode if the Wi-Fi and NWP blocks have no immediate activity. Within the networking subsystem, entry and exit of the LPDS mode is dictated by activity. When one of the IPs has no immediate activity, it may go to lower power mode. When both IPs are in their lower power mode, the entire networking subsystem is in the LPDS mode. Less than 3 ms is required for the MCU to come out of LPDS mode to active mode.

Table 2. Current Consumption for CC3220SF (at 25°C and V = 3.6 V)

MCU/NWP State	Active	Sleep	LPDS	Hibernate	Shutdown
Active, TX	286 mA–174 mA	282 mA–170 mA	266 mA–154 mA	-	-
Active, RX	74 mA	70 mA	53 mA	-	-
Active	Varies	Varies	Varies	-	-
LPDS	25.2 mA	21.2 mA	135 µA	-	-
Hibernate	-	-	-	4.5 µA	-
Shutdown	-	-	-	-	1 µA

See the CC3235MODx SimpleLink TM Wi-Fi® CERTIFIED TM dual-band wireless MCU module data sheet for details on current consumption.

2.4.6.2 Always-Connected Mode

When an 802.11 station is connected to the access point, it must receive the beacons transmitted by the AP. APs generally transmit a beacon every 102.4 ms. 802.11 standards define the Delivery Traffic Indication Map (DTIM) as a specific beacon that contains information regarding incoming packets for the STA. The AP may choose its DTIM interval (such as 1- every beacon, 2-every other beacon, and so forth). The DTIM interval can be changed if the device is in AP mode. This special low-power policy instructs the networking subsystem to skip beacons and DTIM packets, and comes with a desired max sleep time parameter. Though the maximum sleep time can be 2 seconds, it is recommended to set to less than 0.5 seconds to ensure reliable communication.

In always connected mode, a device in STA mode is trying to connect to the access point then enters the entire system in LPDS with only the WLAN subsystem periodically waking up to service beacons.

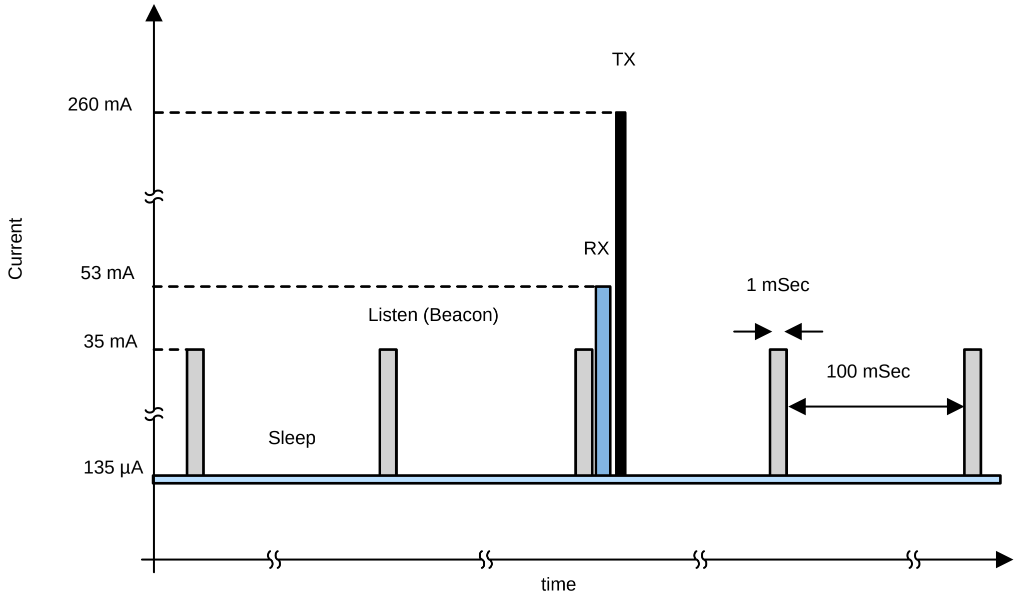


Figure 8. Current Consumption Pattern in Always-Connected Mode

Figure 8 shows typical current consumption by the Wi-Fi device during always-connected mode. For every fixed time interval, the network subsystem wakes up through timer interrupt and gets ready to receive beacon from the AP. It polls to see if there is any data from the AP. In case of any data, then it wakes the MCU and receives the message and transmits any data, if necessary. The device can listen only at the time of the beacon. The rest of the time, it is in LPDS mode. The time for which the current stays at peak value depends on the amount of data that needs to be transferred.

2.4.6.3 Intermittently-Connected Mode

Intermittently Connected mode can be used in applications where the transfer and reception of data is not so frequent and the device operates between long time intervals. Instead of staying connected to the network, it is possible to reduce the power consumption further by the device going to hibernate state. In this mode, the device is trying to connect to the access point then enters hibernate state between working cycles. Almost all the device components are shut down, hence when waking up a new connection needs to be established.

Figure 9 shows the sequence of events and typical power consumption during corresponding events. While in hibernate mode, the device wakes up based on either software event like timer interrupt or hardware interrupt such as GPIO interrupt (from external sensor or triggering event). Network processor (NWP) is initialized and attempts to connect with an AP. Once it establishes the connection, data is being communicated, and once it completes all the communication, the device goes back to hibernate mode.

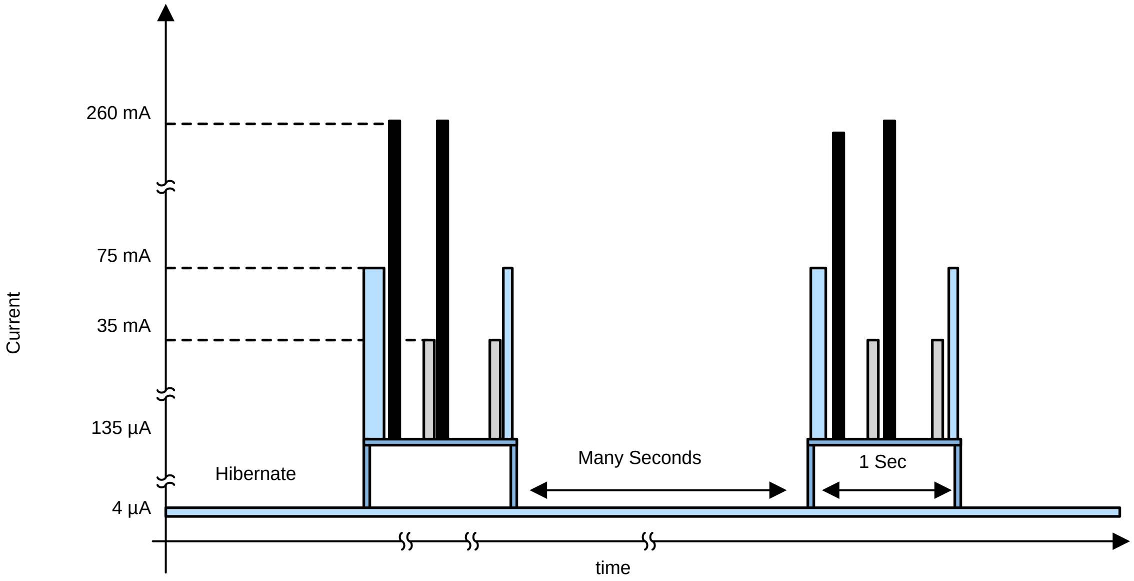


Figure 9. Current Drawn During Intermittent Connection

2.4.7 Over-the-Air Updates

An over-the-air update refers to a software update which is wirelessly transferred and installed on the equipment. Wi-Fi enabled systems can receive software updates directly from cloud servers, which make it easier for vendors to distribute software updates to a large number of devices installed across different locations. The SimpleLink Wi-Fi SDK includes an OTA library that can be used to quickly implement OTA updates in an application. The library includes support for both Dropbox and GitHub APIs; however, the library can also be modified to support alternate content delivery networks (CDN). This reference design demonstrates an OTA update solution for any equipment with the Wi-Fi device CC3X20 using the OTA library with the Dropbox API.

2.4.8 Security

While adding wireless connectivity to any grid equipment, it is very critical to maintain security of the hardware and the data that is being communicated through the air. Sensitive information includes passwords, keys, credentials, configurations, vendor intellectual property (IP), user data, and more. There are different layers of security that can be adopted for these type of end equipment.

Secure Boot: This refers to the process of verifying software that is loaded on to the device from an external memory. During the secure boot process, the CC3220/35S validates the integrity and authenticity of the runtime binary (MCU image). The UniFlash ImageCreator tool for CC3120/35 and CC3220/35 devices can be used to generate images with digital signatures that can be programmed to the external serial flash used by the CC3220/35S. This feature can enable developers to prevent the system from being compromised by malicious code, which could cause the system to stop functioning properly or expose the system to unauthorized access.

Secure Sockets: The SimpleLink Wi-Fi CC3x20/35 devices include embedded, standard-compliant, secure transport layer (TLS/SSL) stacks, which are network protocols that are designed to provide communication security over a TCP/IP connection. Information sent to peers through the Internet is being protected using secure sockets. Embedding the TLS/SSL stacks in the CC3x20/35 device helps in simplifying the code needed to create secure socket connections which frees up the MCU memory for the user application. In this design, secure socket connections are used during AP provisioning, when communicating with the Eclipse IoT broker for data transfer (MQTT client over TLS), and while performing OTA software updates through Dropbox (HTTPS client).

3 Hardware, Software, Testing Requirements, and Test Results

This section provides information on connecting this reference design for functional and performance testing. Users can setup this platform to develop the application and evaluate the performance such as power consumption and setup time for different modes of operation depending on the architecture and data transfer schemes.

3.1 Required Hardware and Software

3.1.1 Hardware

To set up the Wi-Fi network and validate the previously-mentioned functions, the following hardware platforms are used in the TIDA-010007:

- **CC3220MODASF-LAUNCHXL or CC3235REVASF-LAUNCHXL:** This development kit uses the CC3220MODASF or CC3235REVASF, a CERTIFIED single-chip wireless microcontroller module with onboard antenna. This features onboard emulation and temperature sensor. Along with the sensors, it also has push buttons, LEDs and analog input pins which could be used for validating different functionalities described in the system design theory.
- **IMETER BOOST:** This provides a simple and quick way to measure the power consumed by the CC3220/35 LaunchPad™. This power measurement tool is comprised of the INA226 and CC3200 devices to capture and monitor power accurately.
- **CC3200 LaunchPad (CC3200-LAUNCHXL Rev 4.1):** This is used along with the IMETER BOOST by plugging that on the CC3200 LaunchPad

Figure 10 shows connection between these three boards for measuring power consumption by the device under test (DUT). The jumper provided on the DUT to measure the current flowing into the CC3220 is used to connect with the connector J5 (VIN+, VIN–, VBUS and GND) on the IMETER BOOST board on the left. The IMETER board is plugged directly on top of the CC3200 LaunchPad as shown.

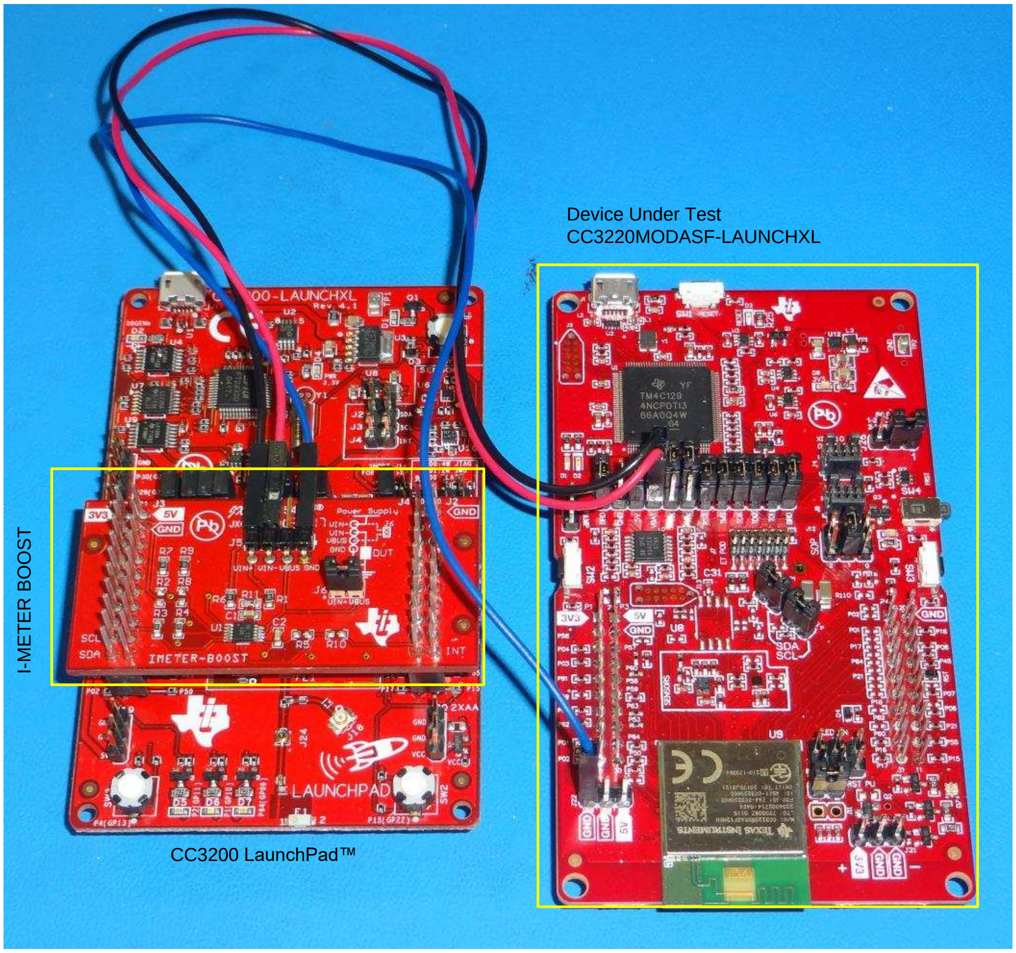


Figure 10. TIDA-010007 Setup for Power Measurement on CC3220

3.1.2 Software

The software created for this reference design is based on the SimpleLink SDK where developers can invest time in creating software once and then reuse the software across different MCUs with integrated wired or wireless technologies. Software required for editing and building the source using CCS (it is possible to move to other IEDs and compiler such as IAR or GCC) are:

- Code Composer Studio (Latest version)
- UniFlash 4.3 (with ImageCreator)
- SimpleLink Wi-Fi CC32xx SDK (v3.10.00.04)
- SimpleLink SDK Explorer: An application on the handheld device for provisioning

Flash using UniFlash ImageCreator: Flash the built MCU image using UniFlash ImageCreator. For step-by-step instructions, see CC3220 SimpleLink™ Wi-Fi® and Internet of Things Solution, a Single-Chip Wireless MCU Getting Started Guide.

To see the debug messages, open a serial terminal application on the computer that the LaunchPad is connected to, and connect to the COM port called XDS110 Class Application/User UART. Table 3 lists the settings for how the serial port is configured.

Table 3. Serial Port Settings

SETTINGS	VALUE
Baud Rate	115200
Data	8 bit
Parity	-
Stop	1 bit
Flow Control	-

3.1.2.1 Provisioning

Provisioning of the device can be done either by AP provisioning or the SmartConfig method using the SimpleLink Starter Pro mobile application for Android or iOS. Follow the instructions in the mobile application or see the instructions in Task 3 of the SimpleLink Academy Training for Wi-Fi Provisioning. Multiple statements are printed to the serial terminal during the provisioning process. The device prints when a new profile is added, a successful connection to the AP is made, an IP address is acquired, and when the mobile device used to provision the CC3220/35 confirms that provisioning succeeded.

NOTE: Upon successful provisioning, the profile is stored on the device. Resetting the CC3220/35S LaunchPad causes the application to restart and the device to attempt to connect to the stored profile.

3.1.2.2 Send and Receive Messages

The CC3220/35 connects to an MQTT broker over TLS, using the MQTT library from the SimpleLink SDK. The MQTT library implements a set of functions that can be used to operate as an MQTT client without dependency on connecting to a specific server. In this example, an open eclipse broker (iot.eclipse.org) is used.

In this demo, the Wi-Fi device subscribes to eight topics as discussed in Section 2.4.5.2.

- /cc3220/TIDA010007/<Unique Device ID>/BreakerON: Upon receiving a message from the user for this topic, the breaker is turned on. In this demo, one of the GPIOs is pulled high representing the input to the trip coil.
- /cc3220/TIDA010007/<Unique Device ID>/BreakerOFF: This topic can be used for turning off of the breaker or breakers.
- /cc3220/TIDA010007/<Unique Device ID>/UpdateRTC: Use this subscription topic to update the clock in all the connected devices for synchronization by updating the device current local time obtained from global NTP servers.
- /cc3220/TIDA010007/<Unique Device ID>/UpdateOTA: Use this topic to signal to the device to initiate an OTA update. When the device receives any message published on this topic, the OTA state machine runs.
- /cc3220/TIDA010007/<Unique Device ID>/ReqStatus: This topic can be used by others to obtain the current status of the breakers.
- /cc3220/TIDA010007/<Unique Device ID>/ReqTemp: Similar to breaker status, temperature near the breaker can be acquired using this topic.
- /cc3220/TIDA010007/<Unique Device ID>/StartPeriodicData: Using this subscription topic, the user can connect to the network and trigger the device to start sending periodic data.
- /cc3220/TIDA010007/<Unique Device ID>/StopPeriodicData: If the user wants to stop receiving the periodic data, use this topic to stop the service.

In the previous topic names, <Unique Device ID> indicates the 128-bit unique device identifier (in hexadecimal format) built into each CC3220/35S device. This ID makes the topics used by each device unique, which can prevent devices from receiving messages intended for other systems. The actual topic names used for a specific device are printed at the top of the serial terminal when the application restarts. The application dynamically sets the topic names by reading the device ID from the network processor and printing the ID value to the topic strings before they are used in the application.

Similarly, there are list of publish topics where the device can send data or messages to all the users subscribed to the corresponding publish topics. As an example, five publish topics are described here:

- /cc3220/TIDA010007/<Unique Device ID>/BreakerOPEN: This event is triggered when there is a fault and a breaker opens to disconnect the power. A notification is sent to all users subscribed to this topic upon any fault.
- /cc3220/TIDA010007/<Unique Device ID>/BreakerStatus: Breaker can publish its status upon request from the user whenever a message is received for '/cc3220/TIDA010007/<Unique Device ID>/ReqStatus' topic.
- /cc3220/TIDA010007/<Unique Device ID>/PeriodicData: Whenever the device receives a message on the topic to start periodic data (/cc3220/TIDA010007/<Unique Device ID>/StartPeriodicData), it initiates a timer and DMA to start collecting data from the ADC channels. These data are processed within the ARM core to compute RMS values. One more timer is used to send these RMS values over the Wi-Fi by publishing data on this topic. Any user subscribed to this topic will receive these parameters periodically. This service is stopped if there is any message to stop the periodic data.
- /cc3220/TIDA010007/<Unique Device ID>/BreakerAcknowledge: This is acknowledgment back to the user when the user controls the breaker either opening or closing of the breaker.
- /cc3220/TIDA010007/<Unique Device ID>/Temperature: Whenever the user sends a message to '/cc3220/TIDA010007/<Unique Device ID>/ReqTemp' topic, the device measures the temperature from TMP116 and publishes the data on this topic.

Every time the device receives a message on any topic that it is subscribed to or publishes to the targeted device, the device prints a debug message to the serial terminal. This can be used to monitor the system response while debugging and in the development phase. Any MQTT client that can connect to the Eclipse IoT broker can test out the functionality of the demo using the topic names. For example, the mobile applications mentioned in the Wi-Fi MQTT SimpleLink Academy Training can be used to test the behavior of a remote client that can control and monitor the state of the demo.

Every transaction through the external MQTT broker has a fee. Within the local (home) network, the controlling device (typically a handheld device or other device in the same network) should connect to a local server rather than the cloud broker. This can be achieved using the Client-Server method where the local network will eliminate the need for cloud broker access. In this implementation, the SimpleLink Wi-Fi device is running a MQTT server ("local broker") which allows local MQTT clients to communicate with each other. Simultaneously, it is also running a client which is connected to a cloud broker. This operation mode is also called "bridge mode." The interface between the onboard client and the server is such that the local clients can also communicate with the remote MQTT clients, which are connected to the same cloud broker as the onboard client. More details on this implementation can be found in task 4- task 9 in *Wi-Fi MQTT SimpleLink Academy Training*.

3.1.2.3 Obtaining Current Date and Time

The CC3220/35S acquires the current date and time using SNTP. Connecting to the NTP server, requesting the date and time, and interpreting the timestamp is handled by the Net Utils and SNTP libraries included in the CC3220/35 SDK. These libraries are built to use a predefined set of NTP servers to request the date and time.

3.2 Testing and Results

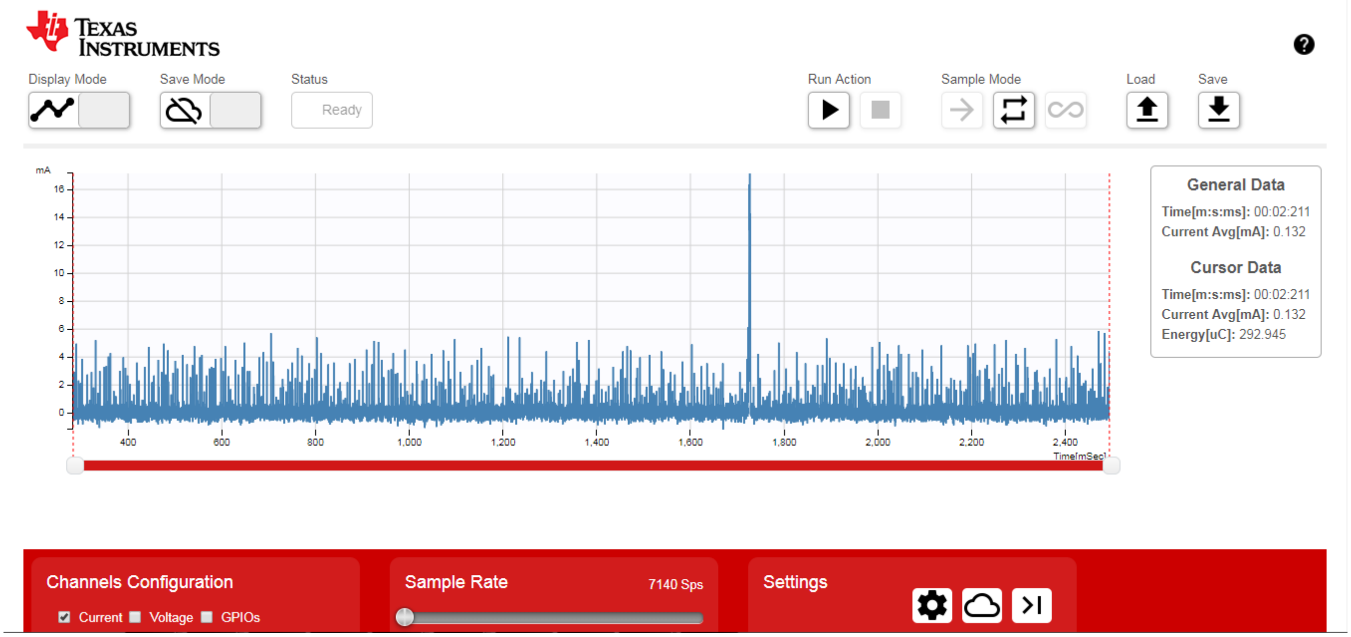
3.2.1 Functional Tests

The performance of the TIDA-010007 is evaluated using the CC3220 LaunchPad to validate the features discussed in Section 2.4:

- **Provisioning**: To connect the device to an external AP, provisioning is done and verified using both the AP configuration method as well as SmartConfig provisioning. While an external AP is not available, one of the devices is configured in AP mode setting up its own network. For validation, other devices are operated in STA mode connecting to the network of the master device.
- **MQTT setup, data transfer**: Once the device is connected to an external AP and established the MQTT connection over TLS, all the listed subscription topics and publish topics are validated. For verifying breaker control, one of the GPIOs is assigned and observed for HIGH and LOW, respectively, for breakers set to CLOSE and OPEN. The health of the system is transmitted by sending measured voltage, current, and temperature values over the Wi-Fi to the cloud. The breaker status is emulated by hardware triggering one of the pins.
- **Periodic data:** Upon request from the user, periodic data can be sent to the subscribed users over the Wi-Fi. To send the data, a timer is used to trigger a DMA to sample data from ADC channels and stored in a memory. Once it collects data for one periodic cycle, the RMS value is calculated. These data are published on MQTT on a periodic basis.

3.2.2 Power Consumption

To understand the power consumption for different application use cases, it is very important to know the power and energy consumption in individual low power and active modes. To analyze this, the CC3220 device is put into hibernate mode and current drawn by the device is measured using IMETER BOOST and captured on a power measurement GUI. These measurements were also performed on a CC3235 device in 5-GHz mode, and the results are recorded in Table 4. While the device is in hibernate mode, the current consumption is approximately 5 µA. Figure 11 shows a profile of current consumption of the device in Low Power Deep Sleep (LPDS) mode where the average current consumption is 130 µA to 150 µA.



General Data

Time[m:s:ms]: 00:02:211
Current Avg[mA]: 0.132

Cursor Data

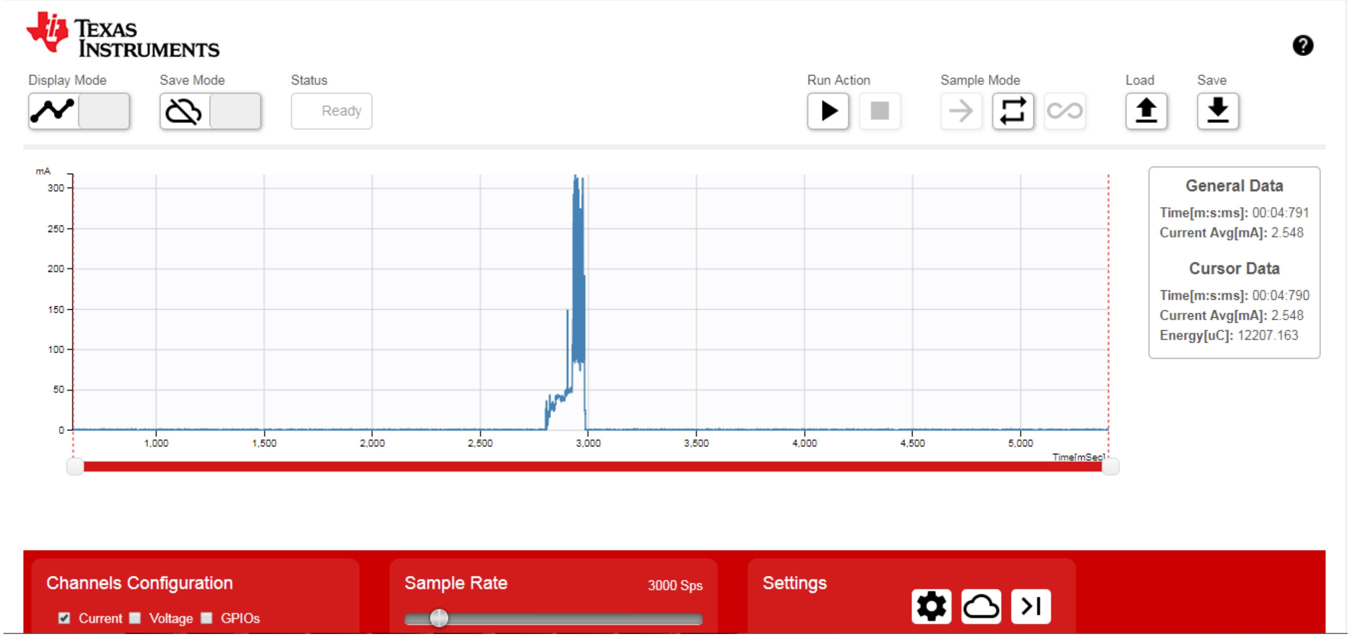
Time[m:s:ms]: 00:02:211
Current Avg[mA]: 0.132
Energy[uC]: 292.945

Channels Configuration: ☑ Current ■ Voltage ■ GPIOs

Sample Rate 7140 Sps

Figure 11. Power Consumption in Low-Power Deep Sleep (LPDS) Mode

In intermittently connected mode, the device is in hibernate mode while consuming current in the range of 5 µA and goes up to peak of 260 mA when connecting to the AP as Figure 12 shows. Typically, it takes 200 ms to 250 ms to reconnect after waking up from hibernation depending on the amount of data that needs to be transferred before going back to hibernation. Average current consumption between wake up and going back to hibernation mode is between 30 mA to 50 mA. Overall average current depends on the frequency of the connection. For the example in Figure 12, the average current is around 2.7 mA for a frequency of 5 seconds. As the time period goes above 5 seconds, the average current is going to reduce further.



General Data

Time[m:s:ms]: 00:04:791
Current Avg[mA]: 2.548

Cursor Data

Time[m:s:ms]: 00:04:790
Current Avg[mA]: 2.548
Energy[uC]: 12207.163

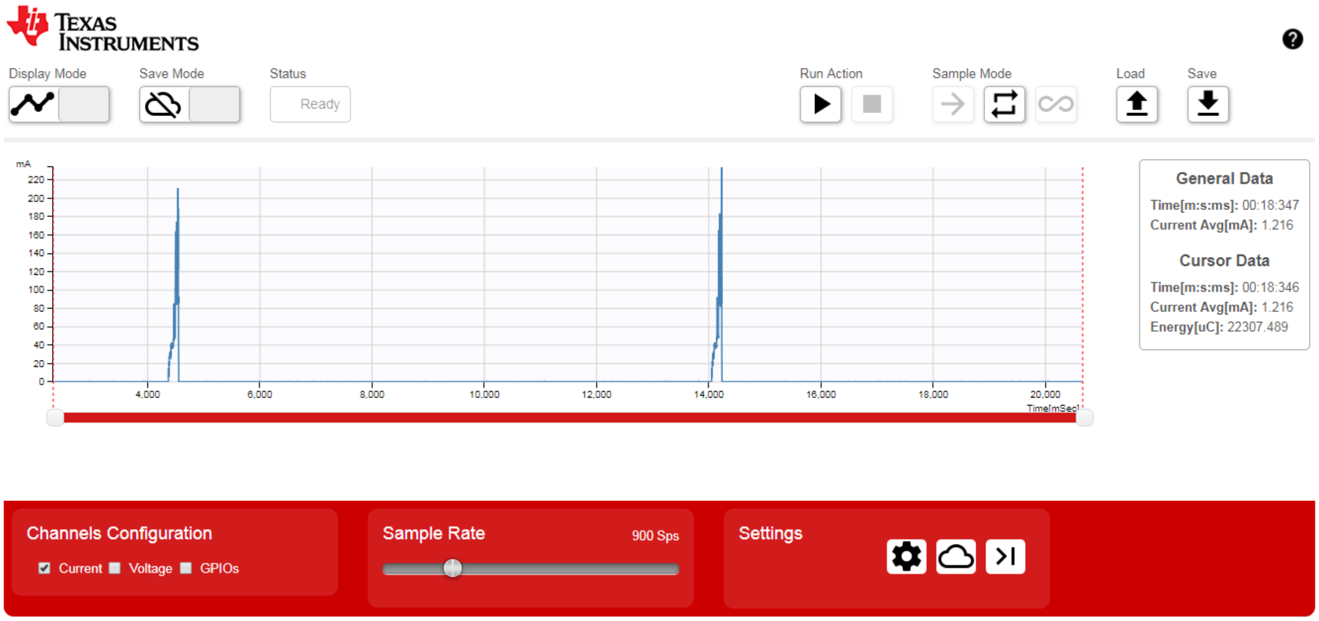
Channels Configuration

☑ Current □ Voltage □ GPIOs

Sample Rate 3000 Sps

Figure 12. Power Profile in Intermittently-Connected Mode (5-s Interval)

As compared to building up a connection every 5 s, the average power consumption reduces linearly as the intermittent connection interval is increased. When this is increased to 10 s as shown in Figure 13, the average current consumption goes down to 1.2 mA. For applications where the reconnection interval is more than 5–10 s, it is beneficial to go with an intermittent connection instead of always staying connected.



Status: Ready

General Data

Time[m:s:ms]: 00:18:347
Current Avg[mA]: 1.216

Cursor Data

Time[m:s:ms]: 00:18:346
Current Avg[mA]: 1.216
Energy[uC]: 22307.489

Channels Configuration

☑ Current □ Voltage □ GPIOs

Sample Rate 900 Sps

Settings

Figure 13. Power Profile in Intermittently-Connected Mode (10-s Interval)

Figure 14 demonstrates the timing and different mode while building the connection in intermittently-connected mode. This shows initialization of the device (MCU and NWP), connection to the AP, and active TX mode to transmit data. The overall average current consumption depends on the amount of data that needs to be sent in one wake-up interval which could go anywhere from 30 mA to 50 mA.

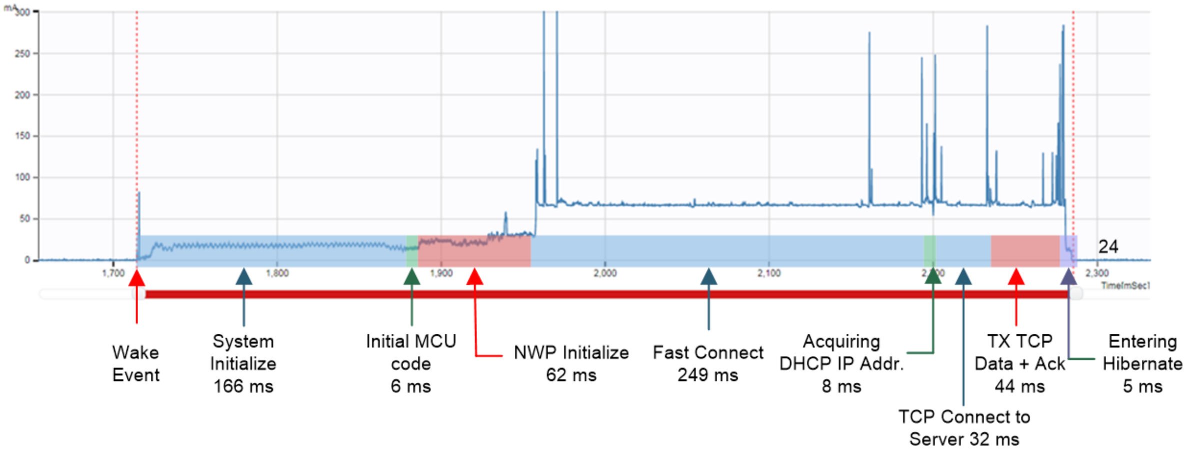
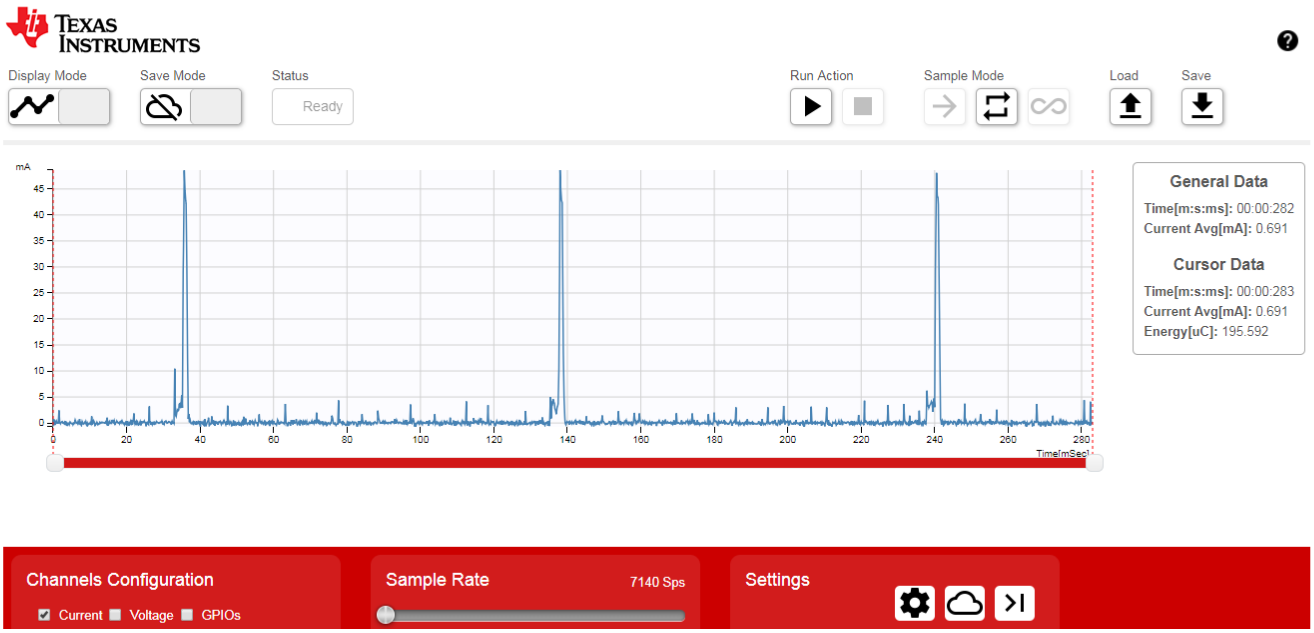


Figure 14. Current Consumption vs Timing in Intermittently-Connected Mode

In always-connected mode, the Wi-Fi device wakes up from LPDS mode and receives a beacon from the AP. When there is no data transfer, the connection between the AP and the device is maintained using this beacon at every fixed interval. Figure 15 shows the current consumption during this type of beaconing mode. A peak current of approximately 50 mA is observed while the average current is going to be less than 700 µA for an LSI of 0.1 s. It is possible to change the LSI by setting DTIM to further reduce the current consumption in this mode of operation.



General Data

Time[m:s:ms]: 00:00:282
Current Avg[mA]: 0.691

Cursor Data

Time[m:s:ms]: 00:00:283
Current Avg[mA]: 0.691
Energy[uC]: 195.592

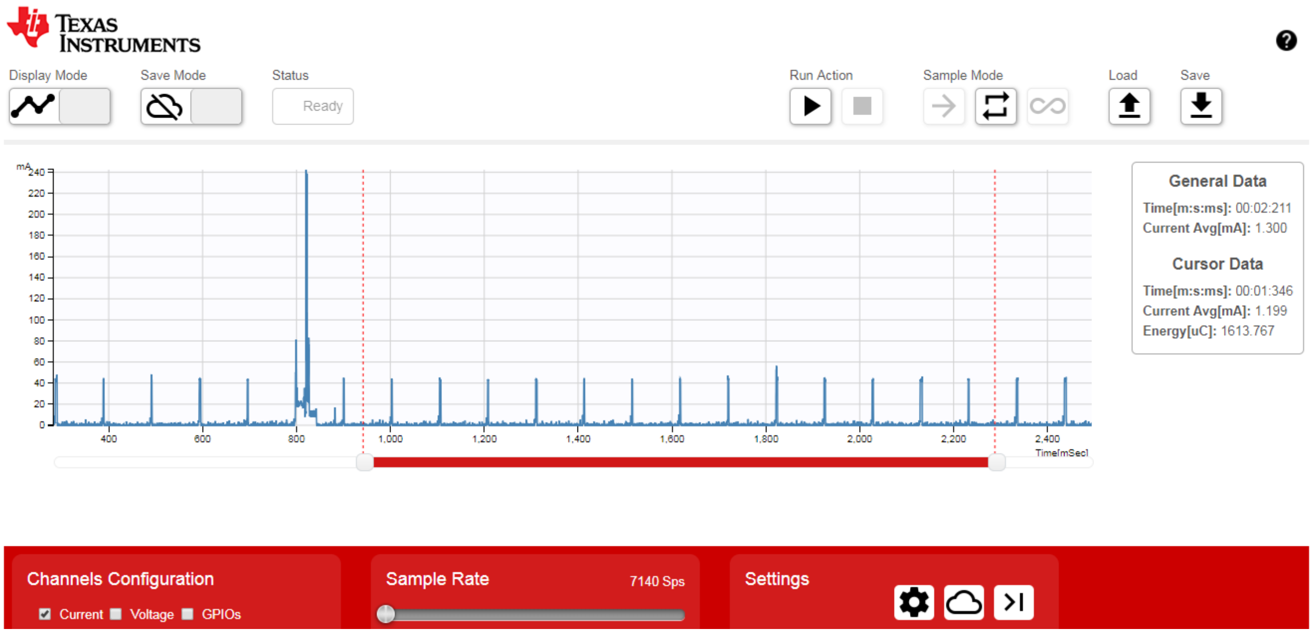
Channels Configuration ☑ Current □ Voltage □ GPIOs

Sample Rate 7140 Sps

Settings

Figure 15. Current Consumption: Always-Connected Mode Without Data Transfer

During the beacon mode, the device polls to see if there is any data from the AP. Whenever there is any data that needs to be either transferred or received, the device comes out of beacon mode and starts actively TX or RX. Figure 16 shows typical current consumption from the device while transferring data during always-connected mode. The duration at which it stays in peak current (approximately 250 mA) depends on the amount of data being sent through Wi-Fi. Henceforth, the average current consumption is going to vary according to the data packet size.



General Data

Time[m:s:ms]: 00:02:211
Current Avg[mA]: 1.300

Cursor Data

Time[m:s:ms]: 00:01:346
Current Avg[mA]: 1.199
Energy[uC]: 1613.767

Channels Configuration ☑ Current □ Voltage □ GPIOs

Sample Rate 7140 Sps

Settings

Figure 16. Current Consumption: Always-Connected Mode During Data Transfer

Table 4. Power Consumption Summary

Mode	AVERAGE CURRENT		PEAK CURRENT	
	2.4 GHz	5 GHz	2.4 GHz	5 GHz
Hibernate	5 µA	5 µA	-	-
Low power deep sleep	130 µA	105 µA	16 mA	6 mA
Transceiver	2 mA	-	200 mA	-
Intermittently connected mode	3 mA	7 mA	260 mA	300 mA
Always connected mode	2 mA	1 mA	220 mA	250 mA
			45 mA (beacon only)	60 mA (beacon only)

3.2.3 Performance Testing

Table 5 lists a summary of the performance testing.

Table 5. Summary

TEST	RESULT
Provisioning to a new Wi-Fi network using AP provisioning	OK
Provisioning to a new Wi-Fi network using SmartConfig	OK
Receive breaker OPEN or CLOSE command and execute	OK
Health status update: send V_{RMS}, I_{RMS}, and temperature on request	OK
Send breaker status	OK
Power consumption measurement	OK

4 Design Files

4.1 Schematics

To download the schematics, see the design files at TIDA-010007.

4.2 Bill of Materials

To download the bill of materials (BOM), see the design files at TIDA-010007.

4.3 Altium Project

To download the Altium Designer® project files, see the design files at TIDA-010007.

4.4 Gerber Files

To download the Gerber files, see the design files at TIDA-010007.

4.5 Assembly Drawings

To download the assembly drawings, see the design files at TIDA-010007.

5 Software Files

To download the software files, see the design files at TIDA-010007.

6 Related Documentation

1. Texas Instruments,*CC3220 SimpleLink Wi-Fi Wireless and Internet-of-Things Solution, a Single-Chip Wireless MCU Data Sheet*
2. Texas Instruments,*CC3220 SimpleLink™ Wi-Fi® LaunchPad™ Development Kit Hardware User's Guide*
3. Texas Instruments,*SimpleLink™ CC3120, CC3220 Wi-Fi® Internet-on-a-chip™ Solution Built-In Security Features Application Report*
4. Texas Instruments,*CC3120, CC3135, CC3220, CC3235 SimpleLink™ Wi-Fi® and Internet of Things Network Processor Programmer's Guide*
5. Texas Instruments,*CC3120, CC3220 SimpleLink™ Wi-Fi® Internet-on-a chip™ Networking Subsystem Power Management Application Report*
6. Texas Instruments,*SimpleLink™ Wi-Fi® CC3120 and CC3220 Provisioning for Mobile Applications User's Guide*
7. Texas Instruments,*SimpleLink™ Wi-Fi® CC3220 Out-of-Box Application User's Guide*
8. Texas Instruments,*A primer to Wi-Fi® provisioning for IoT applications White Paper*
9. Texas Instruments,*CC3120, CC3220 SimpleLink™ Wi-Fi Internet-on-a chip™ Solution Device Provisioning Application Report*
10. Texas Instruments,*CC3X20 SimpleLink™ Wi-Fi® and Internet-of-Things Over the Air Update Application Report*
11. Texas Instruments,*TIDC-01005 Battery-Powered, Smart-Lock Reference Design With Cloud Connectivity Using SimpleLink™ Wi-Fi® Design Guide*

6.1 Trademarks

SimpleLink, E2E, Internet-on-a-chip, DCS-Control, SmartConfig, LaunchPad are trademarks of Texas Instruments.
Altium Designer is a registered trademark of Altium LLC or its affiliated companies.
Cortex is a registered trademark of Arm Limited.
Bluetooth is a registered trademark of Bluetooth SIG, Inc.
CERTIFIED is a trademark of Wi-Fi Alliance.
Wi-Fi is a registered trademark of Wi-Fi Alliance.
ZigBee is a registered trademark of ZigBee Alliance.
All other trademarks are the property of their respective owners.

7 Terminology

AP: Access Point STA: Station

CDN: Content (or Cloud) Delivery Network (for OTA)

DTIM: Delivery Traffic Indication Map

DSSS: Direct-sequence spread spectrum

LPDS: Low Power Deep Sleep

LSI: Long Sleep Interval

MQTT: Message Queue Telemetry Transport

OTA: Over The Air

RMS: Root Mean Square

RTC: Real Time Clock

SC: SmartConfig

SNTP: Simple Network Time Protocol

SSID: Service Set Identifier

STA: Station

TBTT: Target Beacon Transmission Time

8 About the Author

PRASANNA RAJAGOPAL is a systems engineer at Texas Instruments Dallas where he is responsible for developing reference design solutions for Grid Infrastructure in Industrial Systems. Prasanna brings to this role his expertise in power electronics and mixed signal systems. Prasanna earned his PhD from IISc, Bangalore, India

AMIT KUMBASI is a systems architect at Texas Instruments Dallas where he is responsible for developing subsystem solutions for Grid Infrastructure within Industrial Systems. Amit brings to this role his expertise with defining products, business development, and board level design using precision analog and mixed-signal devices. He holds a Master’s in ECE (Texas Tech) and an MBA (University of Arizona). Reach Amit at amit-kg@ti.com.

Revision History

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

Changes from Original (August 2018) to A Revision	Page
• Added CC3235	1
• Added 2.4G or 5G	1
• Added CC3235SF to design resources	1
• Changed Features and Applications	1
• Added CC3235SF	3
• Added *Connected Circuit Breakers* image	3
• Added information to *Key Systems Specifications* table	4
• Added CC3235	5
• Added CC3135	5
• Changed device information in *Highlighted Products* for CC32200xx and CC3235x	6
• Added link to the CC3235MODx SimpleLink TM Wi-Fi® CERTIFIED TM dual-band wireless MCU module data sheet for details on current consumption for CC3220SF	15
• Added CC3235REVASF-LAUNCHXL	18
• Changed SimpleLink Wi-Fi CC32xx SDK software version	19
• Added information regarding the CC3235 device	22
• Added *Power Consumption Summary* table	26

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