TI Designs

Low-Power Wi-Fi Enabled Electronic Paper Display (EPD)

TI Designs

TI Designs provide the foundation that you need including methodology, testing and design files to quickly evaluate and customize the system. TI Designs help *you* accelerate your time to market.

Design Resources

TIDC-CC3200_EPD_Design	Tool Folder Containing Design Files
CC3200	Product Folder

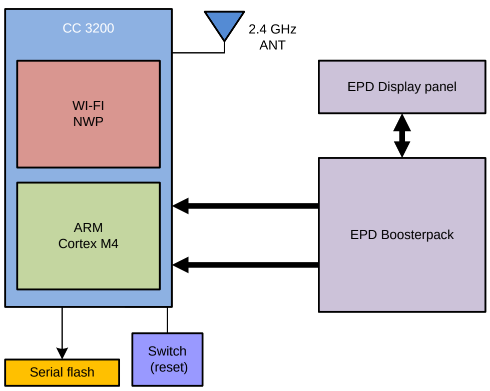
ASK Our E2E Experts

Design Features

- Low-Power, Battery Powered
- Cloud Connectivity
- Local Control
- Supports 5 Display Sizes
- Supports Custom Graphics
- Broadcast Contents Option

Featured Applications

- Electronic Shelf Labels
- Name Tags
- Indoor Signage
- Wearables



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

1 Key System Specifications

Table 1 shows the key system specifications.

Table 1. Key System Specifications

PARAMETER	SPECIFICATION	DETAILS
Operating voltage	2.3 V to 3.6 V	
Working environment	Indoor environment. For higher temperature and humidity range, compensation has to be made when writing to the EPD	
Average current	Highly dependant on use-case	See Section 5.3
Peak current	Achieved during power-ups	CC3200 Datasheet

2 System Description

2.1 Introduction

This design demonstrates the single-chip implementation of a wireless Electronic Paper Display (EPD) using Wi-Fi technology. Technologies that use EPD displays require low-power operation and a method for the user to update the display information. This design provides user control using remote maintenance, an important service in the growing Industrial Internet of things (IoT) market. The design uses the low-power TI SimpleLink Wi-Fi CC3200 wireless MCU, allowing the user to work on two AA batteries for several months, depending on the specific use-case. With an integrated micro-controller and Wi-Fi network processor, this CC3200-based design is a single-chip solution for an EPD display with remote control, locally or over the internet using the MQTT protocol.

3 Design Features

The low-power wireless EPD display allows the user to easily and wirelessly control the contents of any of five different EPD display panels, and demonstrates TI's SimpleLink capabilities of ease of use and low-power. Using either an internal HTTP page embedded in the device, or any MQTT client, the user has the ability to control the contents of the EPD display and configure many parameters. The parameters are then saved and loaded on next power-up. The display accepts text and icons as custom icons, and graphic files may be programmed into the device from a PC.

When remotely controlled, the user has the ability to update any number of EPD panels simultaneously, regardless of their physical location or size by giving them the same name. This use-case is crucial for some electronic shelf label applications, where pricing labels need to be updated at several locations at once, for example.

From a power perspective, this design utilizes one of EPD technology's main advantages of consuming no power between updates. In between updates, the Simplelink device is configured to remain connected, allowing for prompt response of the EPD panel, or intermittently connected at a configurable interval. Being connected to the configurable interval allows for an even lower power consumption and longer battery life. CC3200's Cortex M4 MCU directly drives the EPD display, fast and efficiently, allowing shorter update times compared to other implementations, contributing to the low power consumption of the entire system.

The display is comprised of the following parts:

- A title bar– shows the current operating mode, the device name, and Wi-Fi® reception level.
- An icon, typically 64×64 pixels– either a pre-loaded BMP, or an over-the-air sent graphics.
- Title text– appears next to the icon, typically in a larger font size than the body text. Not available in 1.44" display layout.
- Body text– appears at the bottom. Is typically more than one line long and automatically wraps.

Figure 1 shows the different EPD layouts.

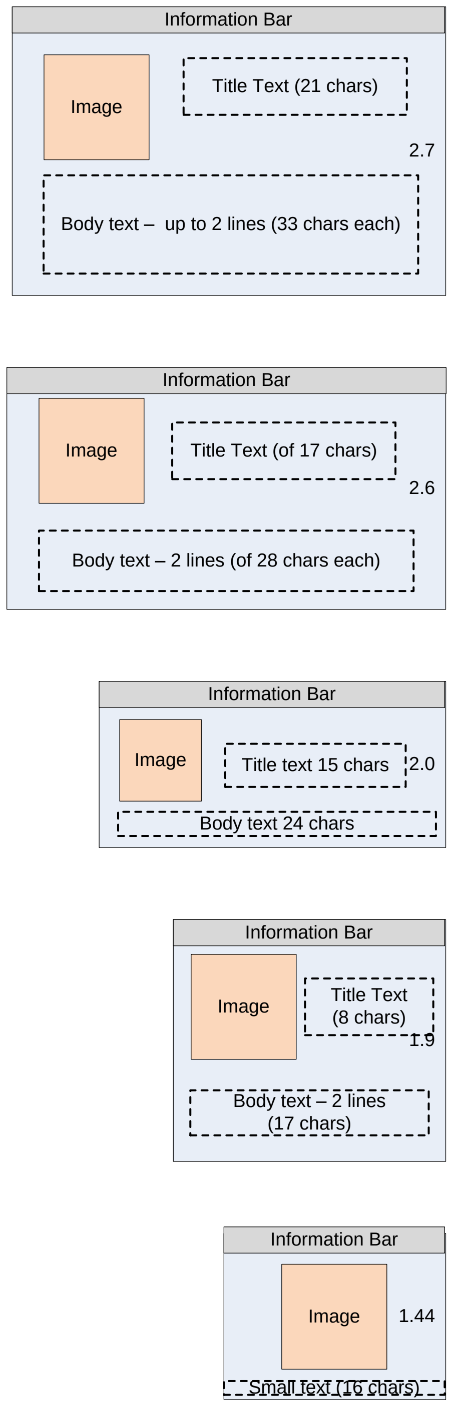


Figure 1. EPD Layout per Display Size

4 Block Diagram

Figure 2 shows a high level block diagram of the system. The hardware is comprised of the standard CC3200 LaunchPad™, the EPD BoosterPack™.

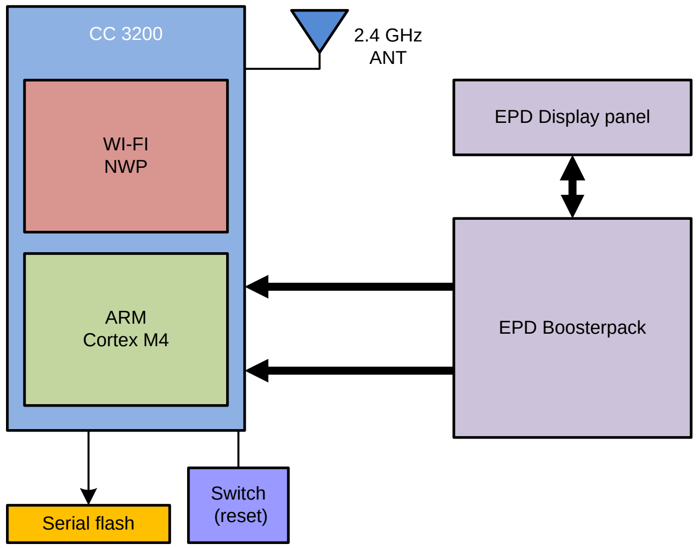


Figure 2. High Level System Block Diagram

4.1 Highlighted Products

This reference design features the CC3200. The SimpleLink Wi-Fi CC3200- wireless MCU integrates a high-performance Cortex-M4 MCU and peripherals. This design utilizes the CC3200-LAUNCHXL EVM that contains the CC3200 reference design along with emulation and a BoosterPack connector. The CC3200 MCU controls the EPD using an SPI interface. Figure 3 shows the CC3200–LAUNCHXL EVM.

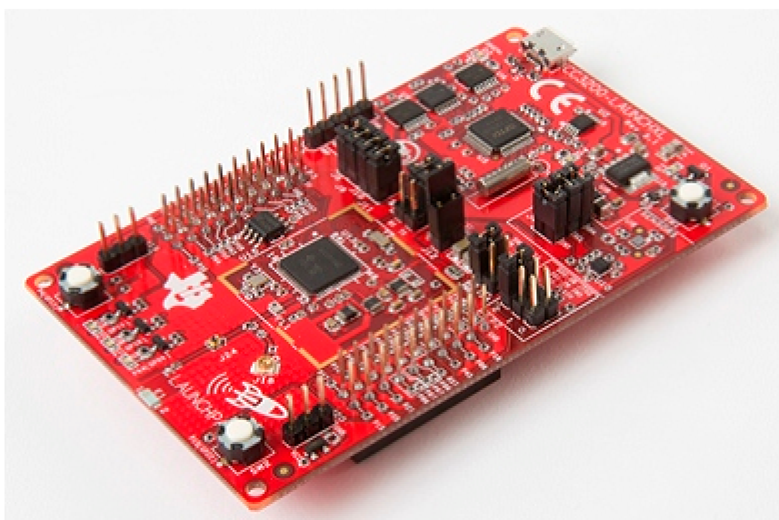


Figure 3. CC3200–LAUNCHXL EVM

5 System Design Theory

This design introduces two ways of controlling the EPD panel–locally, and remotely.

Locally, the EPD panel is controlled by any connected device on the same network. Remotely, the EPD panel is controlled by any internet connected device, anywhere in the world.

Controlling the EPD panel remotely is more power efficient, as the MQTT broker (server) buffers the data until it is polled by the SimpleLink device. Figure 4 shows the system design theory, case one–remote control and Figure 5 shows the system design theory, case two–local control.

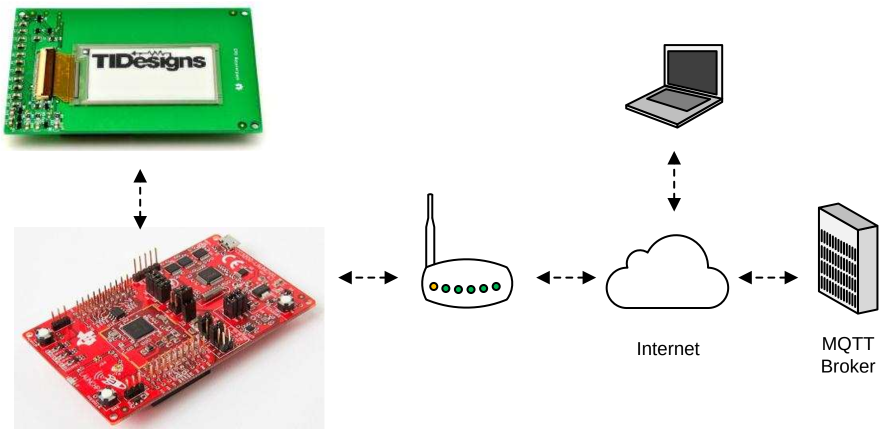


Figure 4. System Design Theory, Case 1–Remote Control

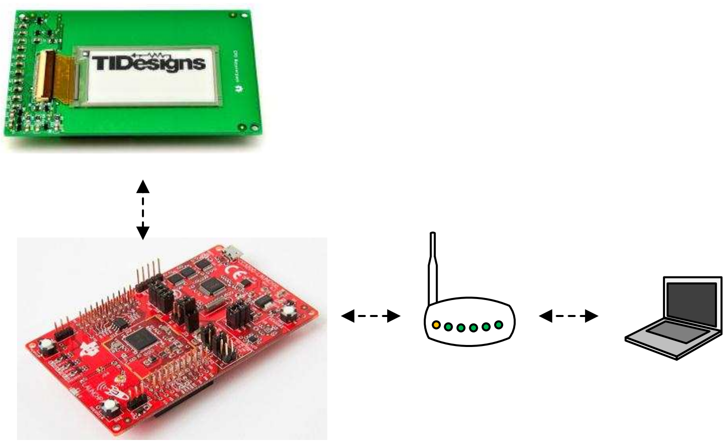


Figure 5. System Design Theory, Case 2–Local Control

5.1 Hardware

5.1.1 EPD BoosterPack

The EPD BoosterPack is designed to be compatible with the standard BoosterPack connectors of the Launchpad board, utilizing 2 of the 4 rows (to maintain an option for 2×10 connectors instead of the currently 2×20 mounted ones). Table 2 shows the mapping of the signals between the boards.

Table 2. BoosterPack Connector Mapping

	JI			J4		
Pin #	**LP Role**	**BP Role**	**Description**	**LP Role**	**BP Role**	**Description**
1	VBAT	VBAT		GND	GND	
2	ANA_IN	BUSY	EPD busy indication	GPIO	DISCHARGE	For fast charge pumps discharge
3	RX	RESET	EPD controller reset	SPI_CS	SPI_CS	SPI interface
4	TX			TEST		
5	GPIO	BORDER	For 2.7" displays only	RESET		
6	ANA_IN	PWM	For G1 displays only	SPI_MOSI	SPI_MOSI	SPI interface
7	SPI_CLK	SPI_CLK	SPI interface	SPI_MISO	SPI_MISO	SPI interface
8	NWP_LOG			SOP2		
9	SCL			FTDI_TX		
10	SDA			GPIO	PANEL_ON	Power switch for EPD

5.1.2 EPD Panels

The design supports all small display panels from pervasive displays, which are:

- 1.44"
- 1.9"
- 2.0"
- 2.6"
- 2.7"

The supported panel type is the Aurora Mb (V231) G2 by pervasive displays. There is backwards support for Vizplex (V230) or G1 panels using software compilation flags, but it is strongly recommended to only use V231 G2 panels.

EPD panels are sensitive to temperature. The lower the temperature is, the more time is required to drive the image to get similar optical results. This design is aimed for indoor applications, and it is assumed that the EPD panel is operating at room temperature.

In case the temperature drops below room temperature to a point in which there is visible optical degradation, the configurable display quality can be set to optimal quality to compensate for it. The software contains a hook for user temperature sensing function. It is currently statically set for 25 degrees. If required, a function that gets the temperature from the Launchpad’s onboard sensor may be implemented.

NOTE: Compensations for lower temperatures defeat the purpose of using partial update to shorten update cycles.

5.2 Software

The software supplied with this design is based on the SimpleLink CC3200 SDK MQTT example, combined with the pervasive display driver. The software works in a FreeRTOS environment, and has two threads. One for the SimpleLink driver, and the other for the application. Figure 6 shows how the software flow is divided into two main steps–initialization and the main state machine, and Figure 7 shows the software main state machine flow.

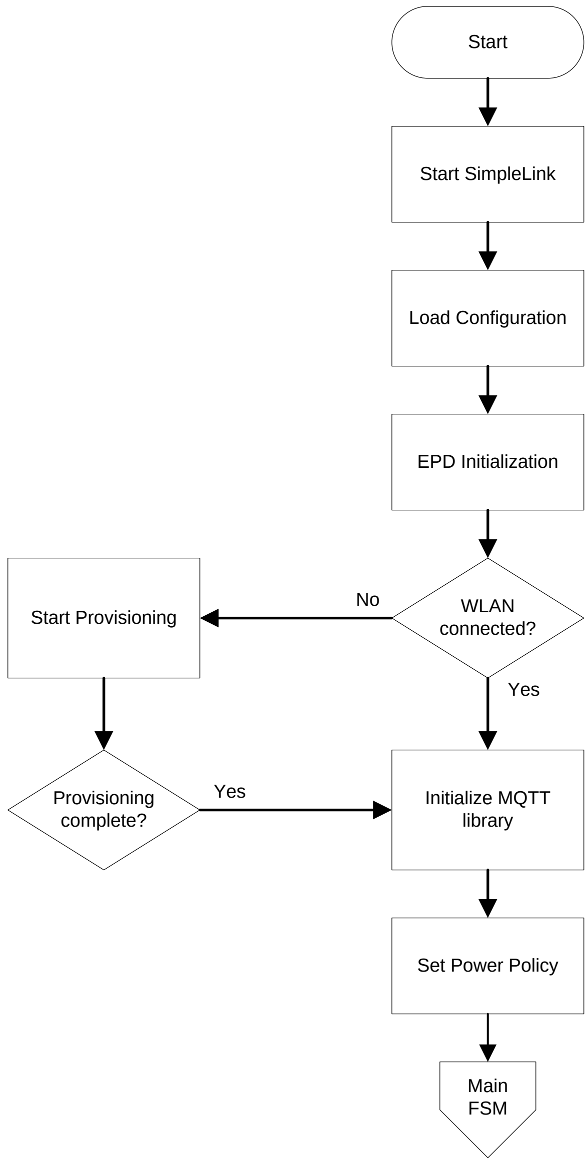


Figure 6. Software Initialization Flow

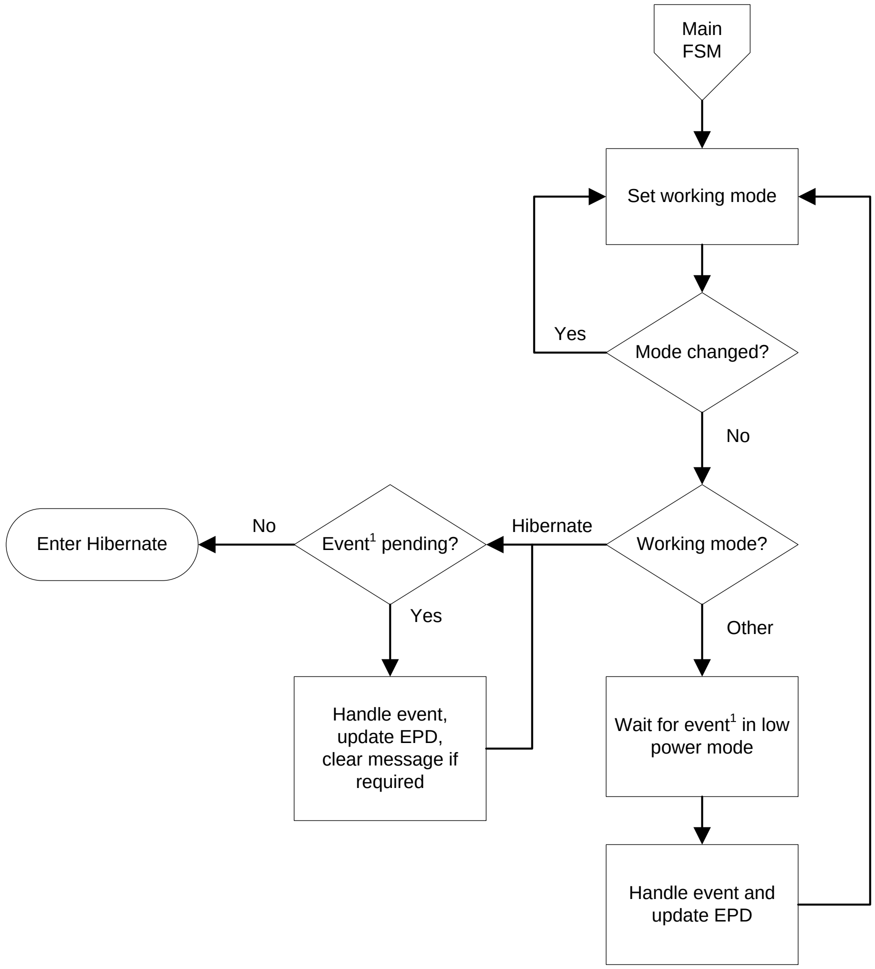


Figure 7. Software Main State Machine Flow

5.2.1 Provisioning

When the system boots (whether from power off or from a hibernate cycle), it will attempt to connect to a known AP. If it fails to find one in a predefined amount of time, the provisioning procedure is activated.

The purpose of this procedure is to connect the system to a new AP and add a profile for later use. When the provisioning procedure starts, a message will be displayed on the screen. Use TI's provisioning application for iPhone or Android to send the AP credentials to the device.

Once provisioning is complete, a message will be displayed on the screen, and the phone's provisioning application directs you to the device's web page. In case provisioning fails or times out, a message is displayed, and the device goes into hibernation to conserve current. The duration of the hibernation is the pre-configured interval, or default value in case it was never configured.

5.2.2 HTTP User Interface

The HTTP page is designed to reflect the currently displayed image on the EPD panel. This is true even when controlled by more than one simultaneously, and updates from one device will propagate to other connected controllers.

The main page allows the user to update the text fields (title and text) and select an icon from a drop-down list. Applying the changes will cause the sign to update.

- Click on the icon sign and load any image if a custom icon is required.

NOTE: The image will be scaled down to 64×64 pixels and is black and white. TI recommends to use an image with the correct size and color pallet to begin with.

- Select the icon number to save the image to the sign, or select volatile in order not to save it when a custom image is uploaded.

  Figure 8 shows the HTTP user interface window.

Figure 8. HTTP User Interface

- Press the three dots icon on the top right corner to open the configuration panel. There you can switch between modes, display types, display qualities, and configure timer intervals. These changes may be saved to the filesystem if required.

5.3 Power

The system is designed to work off 2×AA alkaline batteries, allowing a voltage range of approximately 3-V down to approximately 2.3-V, which is the lowest voltage allowed for the EPD. The system is capable of operating on voltages up to 3.6-V. The EPD driving circuitry is designed to disconnect the EPD (using a FET) whenever it is not in use, and is controlled by the *PANEL_ON* signal.

The current measurement in this episode are done in two parts (CC3200 and EPD Boosterpack) and are then superimposed.

The detailed analysis of power consumption per mode and element is described in the following subsections. For the lowest power consumption, the intermittently connected profile should be used. With this profile, the longest battery life is achieved.

Table 3 lists the projected battery lifetime for 2xAA alkaline batteries on a 2.7” display (worst-case), for various polling intervals and the percentage of new content available per polling cycle. The breakdown for these figures is detailed in the following chapters.

Table 3. Projected Battery Lifetime for Various Scenarios

	LIFETIME IN YEARS			
New Content Available	**Every Update**	**Every 2nd Update**	**Every 4th Update**	**Every 10th Update**
Polling every 10 minutes	0.42	0.52	0.57	0.63
Polling every 15 minutes	0.63	0.78	0.89	0.98
Polling every 30 minutes	1.23	1.52	1.73	1.94
Polling every 1 hour	2.36	2.89	3.26	3.53
Polling every 4 hours	7.5	Over 10	Over 10	Over 10
Polling twice daily	Over 10	Over 10	Over 10	Over 10
Polling once daily	Over 10	Over 10	Over 10	Over 10

For example, if the EPD panel is polling the MQTT broker once every 30 minutes. On average, a new message arrives every 2 hours, and the projected battery lifetime would be about 1.7 years.

With a smart scheduler, a combination of polling intervals may be configured. For example, updating every 15 minutes during working ours, and no updates for the rest of the time, with new data coming every second wake-up, will lead to 1.5 years, while still allowing a reasonably short response time.

5.3.1 WLAN Current Consumption

The design incorporates three main working modes, each with its own current consumption profile. The following sections describe these modes in depth.

NOTE: All figures given were measured in a moderately congested open-air environment, on a typical home AP with WPA2 personal security. The current consumption may vary greatly in both directions based on AP type, security type, distance from the AP, air congestion, traffic from peer clients, and more. Please treat these numbers as an estimate.

5.3.1.1 Local Control Mode

In this mode, the control is done locally within the network. This is aimed for home, office network, or maintenance, and is the most power-consuming mode.

To use SimpleLink's internal HTTP server, the long sleep interval is disabled, and the M4 MCU is kept in low power mode. This is as long as there is no activity except for polling of the device's status once per second.

Figure 9 shows how loading of the page consumes approximately 49-mA over a period of 1.3 seconds.

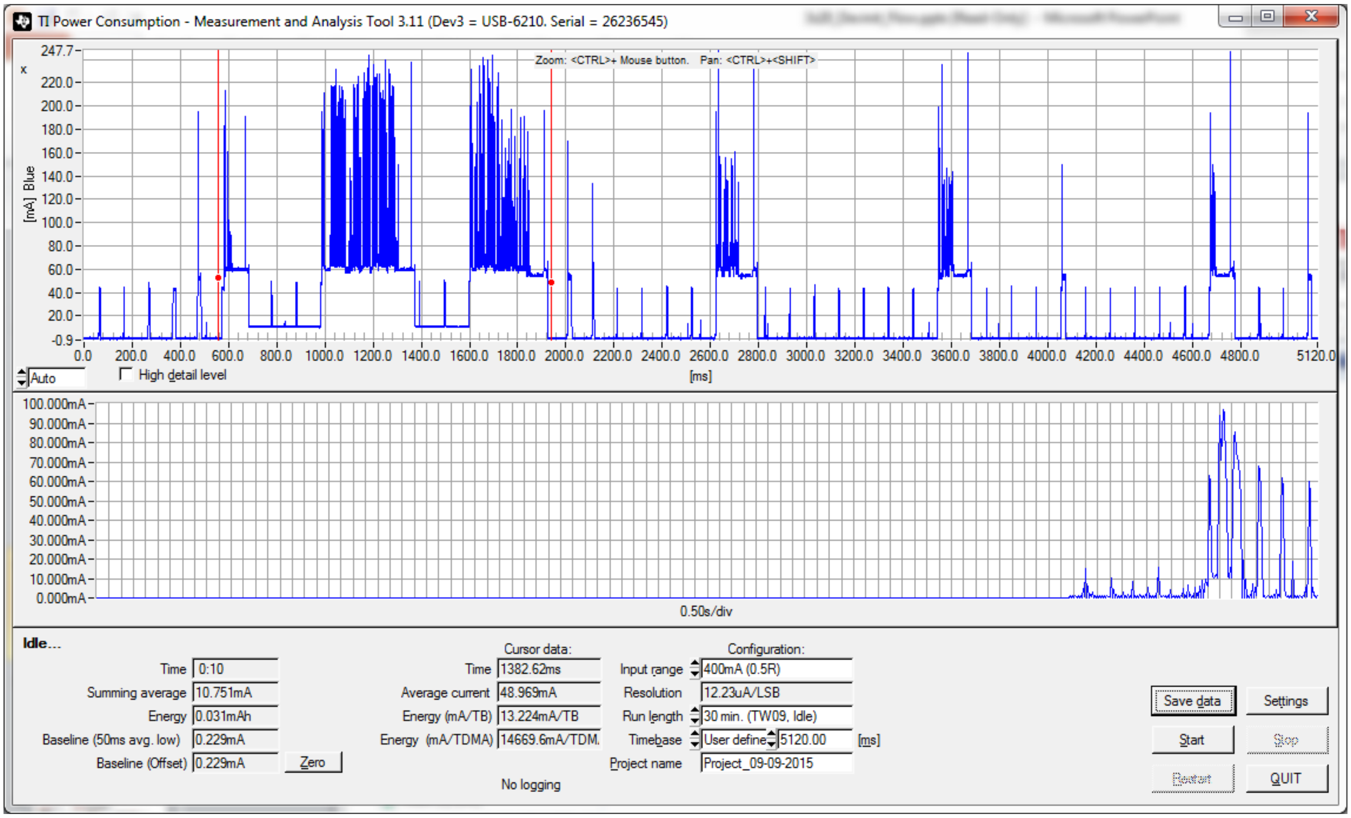


Figure 9. WLAN Current Consumption of HTTP Page Loading

An update of the EPD (sending content to the EPD via the local HTTP page, text and icon number only) consumes an average of 9.4-mA over a time of 1563 mSec. Figure 10 shows the WLAN current during an HTTP update.

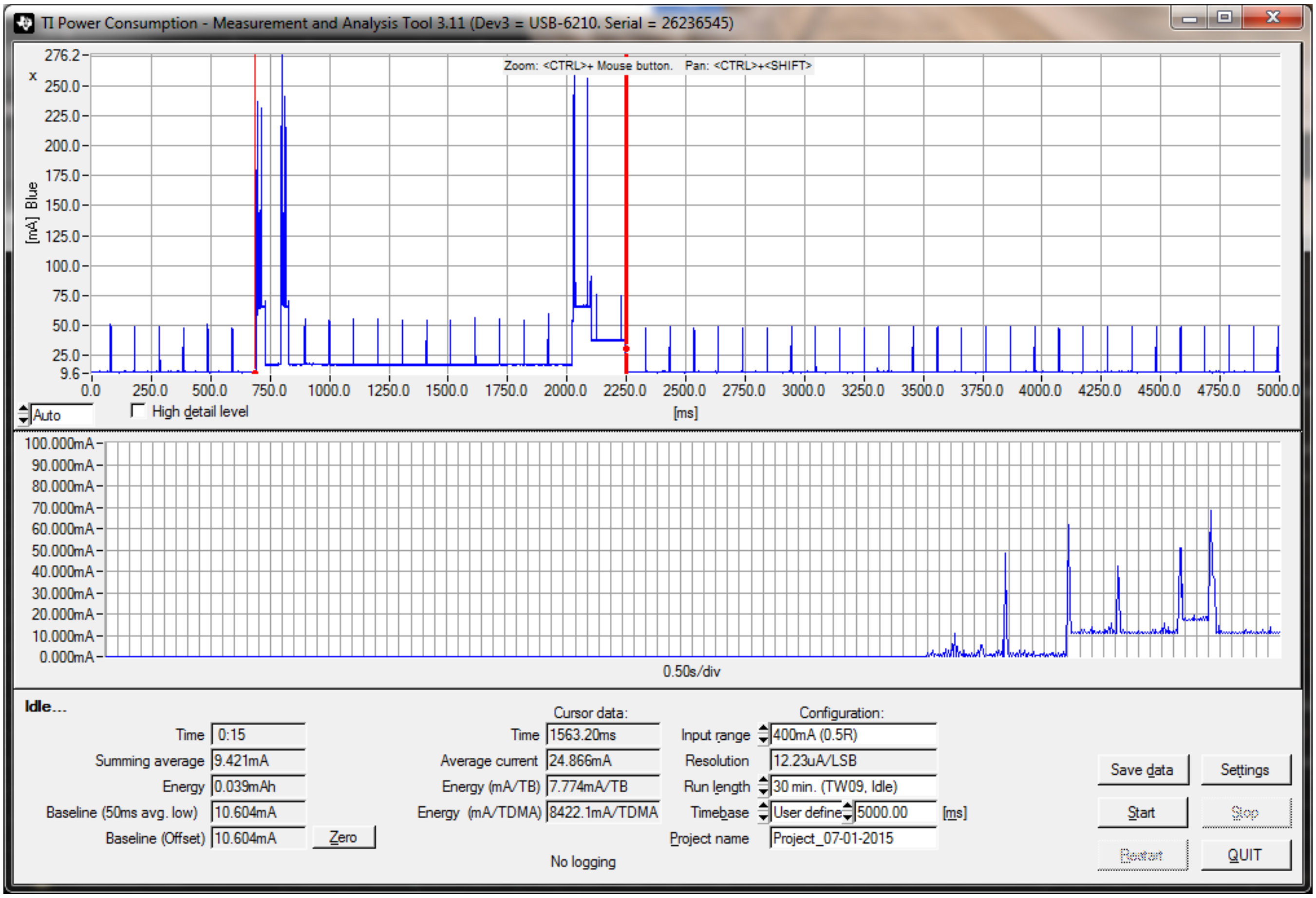


Figure 10. WLAN Current During an HTTP Update

As seen in Figure 10, these 9.4-mA include the baseline 1.5-mA of the WLAN connection, the transmission and reception of HTTP packets, and the M4 MCU being active to update the EPD. This profile does not include the EPD update itself.

Once the page has been loaded, the browser will poll on the device for any changes to the sign's contents, issued from other devices. Polling occurs once per second and helps keep all browsers in sync with the sign's actual contents. Polling consists of an RX/TX cycle once per second, as well as keeping the M4 MCU active for approximately 100 mSec. The average current in this mode is approximately 12.4-mA for one polling browser. Figure 11 shows the WLAN during HTTP idle polling.

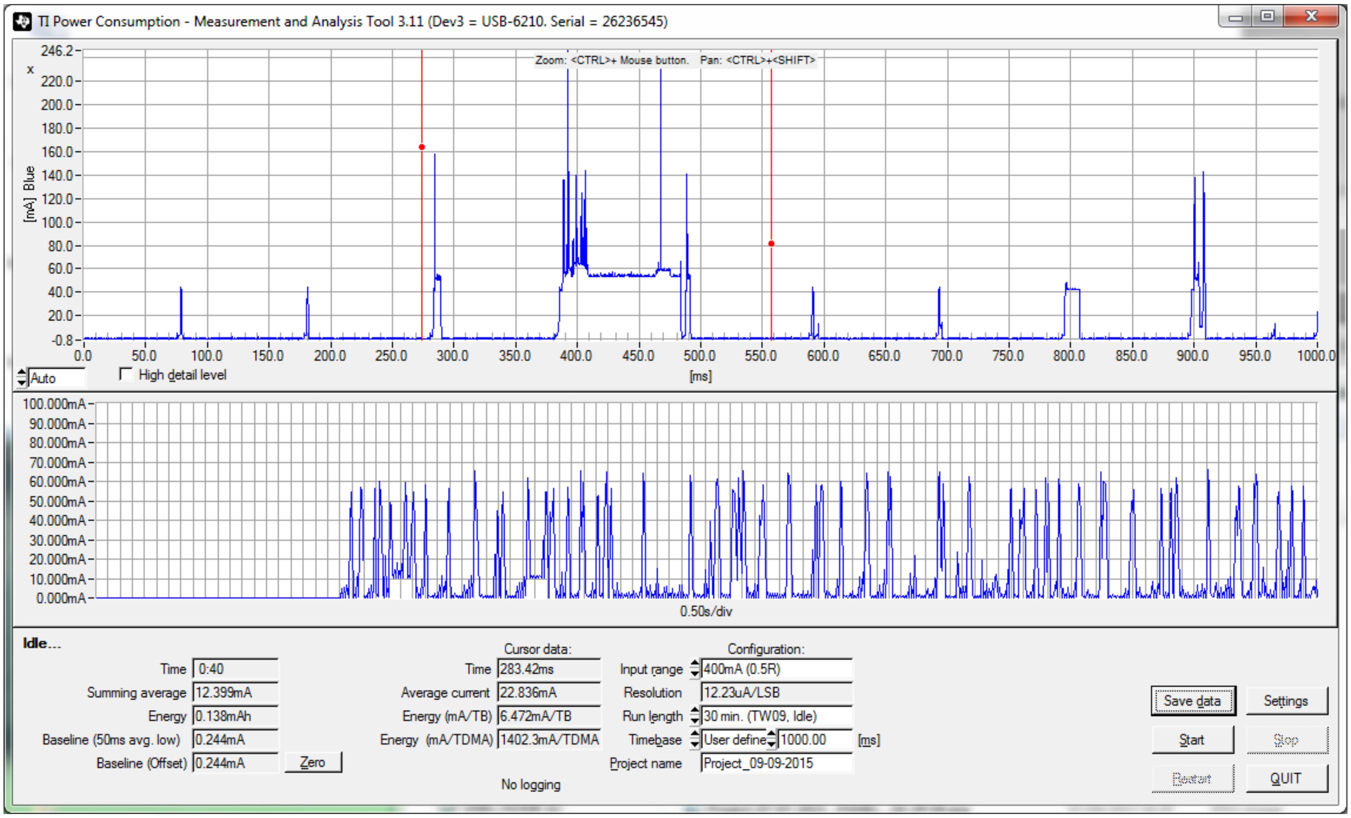


Figure 11. WLAN During HTTP Idle Polling

5.3.1.2 Remote Control Mode or Timer Mode

In this mode, the LSI (Long Sleep Interval) is enabled at an interval of 500-mSec to reduce current consumption while connected to the remote MQTT broker. The HTTP server is disabled in this mode.

While connecting to the remote broker, a keepalive exchange is done every 120 seconds. Averaging the current consumption over 10 minutes shows an average current consumption of slightly less than 1 mA.

An incoming update event from the MQTT broker consumes around 19-mA over a period of 1600 mSec. The update event includes the exchange with the MQTT broker, and the M4 up time for updating the EPD display, but does not include the update of the EPD itself, or the submission of the status update message. Figure 12 shows a single update after 10 minutes of idle connection, showing a summing average of 1.03mA, which gives an estimate for this type of case.

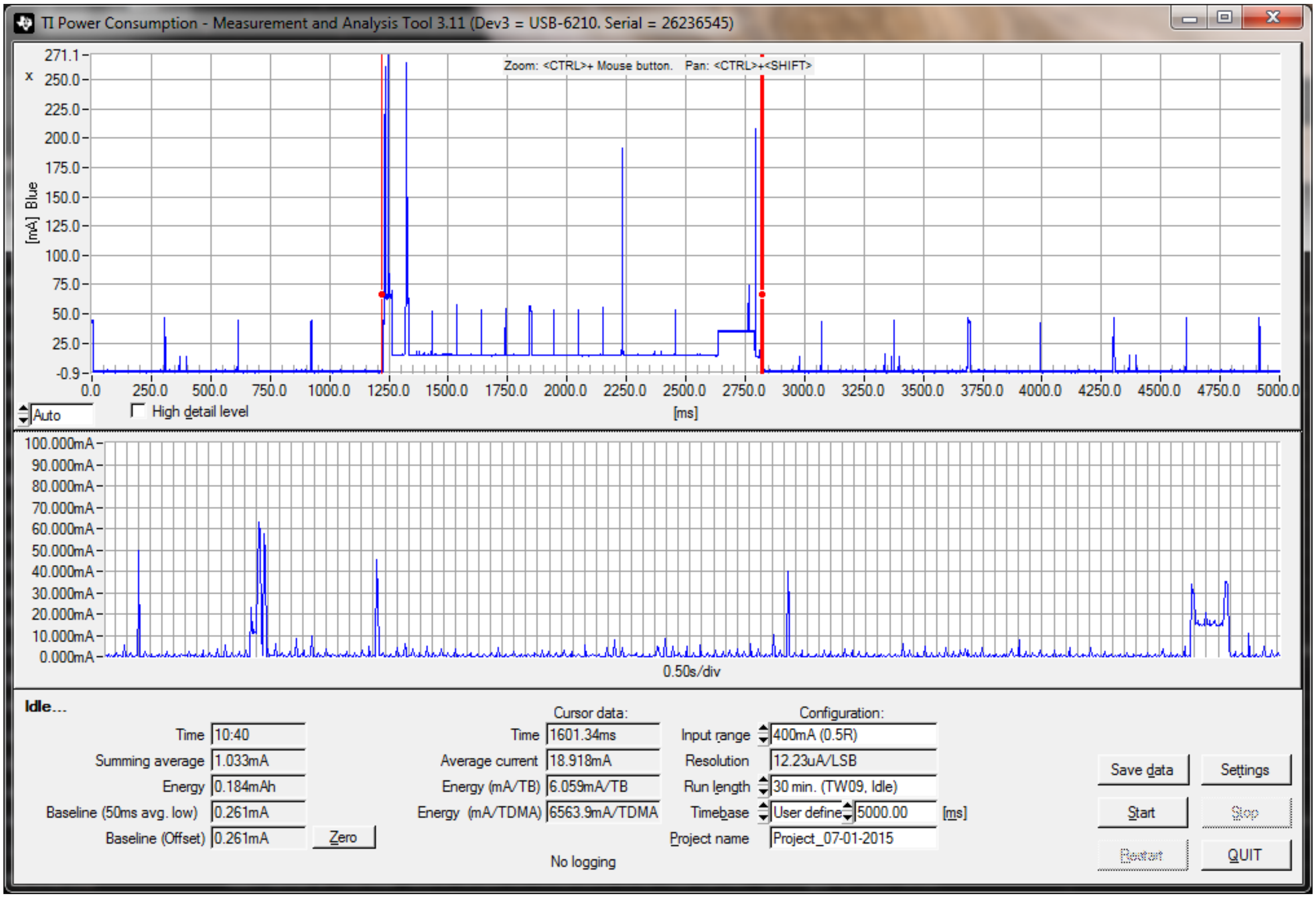


Figure 12. WLAN Current During an MQTT Update

5.3.1.3 Intermittently Connected Mode

This mode consumes the least amount of current. The SimpleLink device remains in hibernate mode between updates for the duration of the configured interval. During this period the EPD is turned off, consumes a negligible amount of current. The WLAN subsystem is also turned off, and the application MCU subsystem is suspended with a timer set for waking it up. The entire system consumes less than 10 uA in this mode (CC3200 + EPD BP). The exact figure varies according to temperature and process, refer to the datasheet for details.

For each hibernate cycle, the MCU wakes up, detects it is out of hibernate mode, begins the WLAN, and waits in power-saving mode until the WLAN is connected to the access point. The MCU then connects to the MQTT broker and waits in power-saving mode for 10 seconds to see if a message is pending. If there are no pending messages, the MCU will disconnect and re-enter hibernate. If there is a pending message, it will be grabbed and the screen will be updated with the new message. According to the settings, the message may be acknowledged (clear retained MQTT message).. According to the settings, the message may be acknowledged (clear retained MQTT message). Unacknowledged messages are useful for controlling several clients simultaneously.

If a message was received, a status update will be sent to signal to all sending devices that the display was updated. All current measurements in this design include the acknowledging of the messages (worst-case scenario). Figure 13 shows intermittently connected–polling without an update.

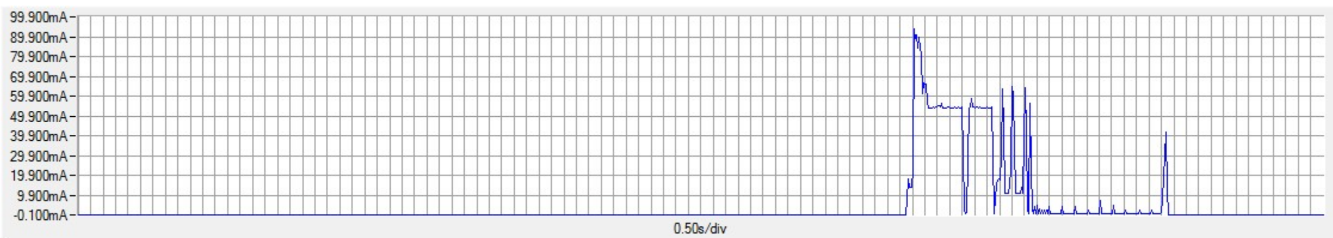


Figure 13. Intermittently Connected–Polling Without an Update

A cycle without an update takes slightly over 10 seconds (as the timer for re-entering hibernate is set for 10 seconds, plus the initialization delay before that timer is set). For such cycles, the EPD panel remains off. Table 4 lists the intermittently connected cycles without an update.

Table 4. Intermittently Connected Cycles Without an Update

EPD TYPE	TIME(S)	AVG. CURRENT (mA)	Q(mC)
2.7”	10.5	21.7	228

Figure 14 shows the intermittently connected–polling and updating a 2.7” panel.

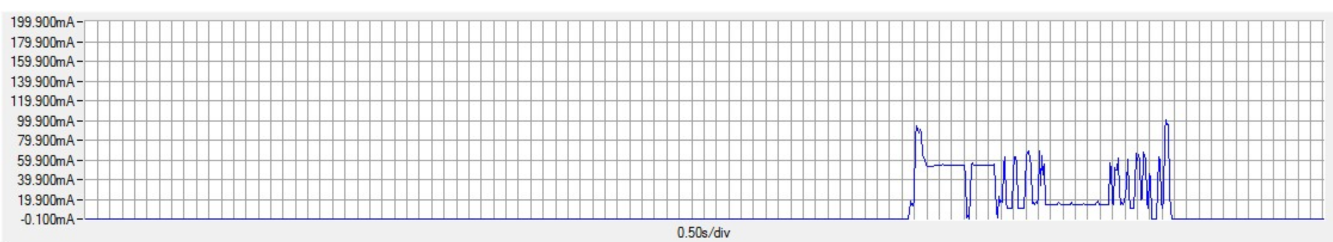


Figure 14. Intermittently Connected–Polling and Updating a 2.7” Panel

For cycles with an update, the extra current is derived from receiving the message, acknowledging it (when configured to), and sending a status update. The cycles' current figures do not vary much according to display size–only the update time variance between the panel types. The worst case (2.7” panels) measurements are taken for the rest of the current consumption measurements. Table 5 lists the intermittently connected cycles with a 2.7” panel update.

Table 5. Intermittently Connected Cycles With a 2.7” Panel Update

EPD TYPE	TIME(S)	AVG. CURRENT (mA)	Q(mC)
2.7”	10.5	35.5	372.8

5.3.2 EPD Current Consumption

In between updates, the supply rail to the EPD panel is cut by a MOSFET, and consumes a very low amount of power.

To minimize current consumption during updates, a partial update scheme is used, which only changes pixels that require change instead of flashing the entire screen (full update). This allows for a shorter update time, meaning less current wasted by the EPD, as well as by the M4 MCU. The M4 MCU is able to finish the update quicker and go back into power saving mode. The downside of this approach is ghosting of the previous image. The amount of ghosting is derived from the amount of time the update takes. This is user configurable, given in three steps (optimized for power, optimized for quality, or balanced). For each update, the user may force a full refresh to clean previous ghosting when required.

Full refresh is forced during the system initially (since the previous image is not kept), and in intermittently connected mode because the memory is not retained and the previous image is not available for comparison. Using the NVMEM to keep the previous image would save some current, but the system's lifetime would shorten due to frequent writes to the serial flash.

5.3.2.1 Stage Time Calculations

Stage time is defined as the time it takes to drive the required image. For each iteration, multiple frame drives are required (each one requires a certain amount of frame time). During the frame time, the ink particles are moved toward their destination (black to white, or vice versa). The frame time depends on the display size and the driving MCU.

Pervasive displays recommends driving at least five frames per stage to achieve an acceptable result. The hard coded stage times in the software are set for power optimized values, balanced values and image quality optimized values.

The stage times for full updates are fixed to the values defined by pervasive displays, which are 630-mSec for 2.7” and 2.6” displays, and 480-mSec for the rest of the displays.

For a partial update, the stage time is kept flexible in this design to allow flexibility of display quality v. current consumption. Stage time was measured per display size on CC3200 during partial updates, and different stage times were tried to achieve the stage time per display quality setting and display size. The addition in current per display profile is described in Section 5.2.2.

Table 6 shows the measured stage time per display size. Table 7 shows the hard coded stage times per display size, for each display quality.

Table 6. Frame Time per Display Size

DISPLAY SIZE	FRAME TIME (mSec)
1.44”	31
1.9”	91
2”	40
2.6”	117
2.7”	103

Table 7. Stage Time per Display Size and Quality

DISPLAY SIZE	BALANCED		POWER OPTIMIZED		QUALITY OPTIMIZED	
	Time (mS)	Frames	Time (mS)	Frames	Time (mS)	Frames
1.44”	350	12	250	9	500	17
1.9”	500	6	350	4	750	9
2.0”	350	9	250	7	500	13
2.6”	500	5	350	3	750	7
2.7”	600	6	350	4	750	8

5.3.2.2 EPD Current Consumption

EPD boosterpack measurements in this section is completed separately from the CC3200 LaunchPad. Figure 15 shows the 2.7” full update current consumption.

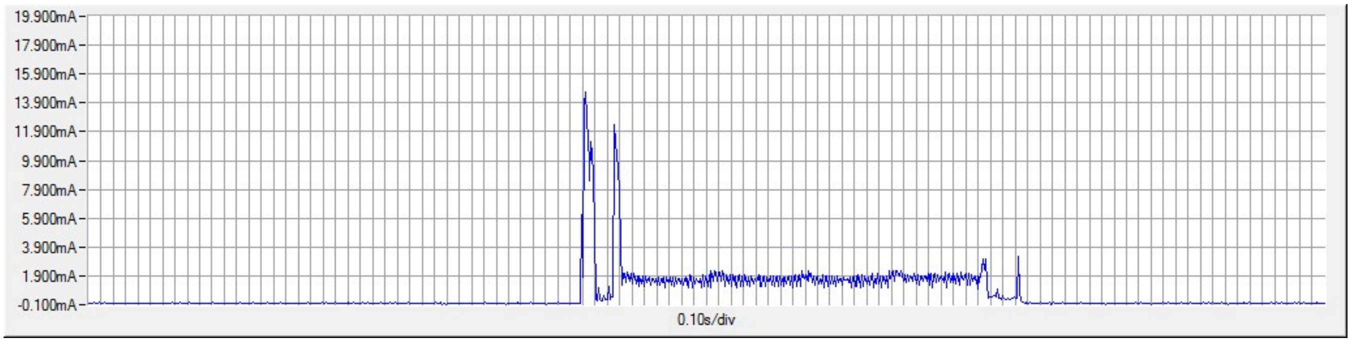


Figure 15. 2.7” Full Update Current Consumption

Table 8 lists the full update current consumption per display type.

Table 8. Full Update Current Consumption per Display Type

EPD TYPE	UPDATE TIME(S)	AVG. CURRENT (mA)	Q(mC)
2.7”	3.58	1.99	7.12
2.6”	3.47	2.06	7.14
2.0”	2.75	2.02	5.55
1.9”	2.71	2.05	5.55
1.44”	2.7	2.11	5.69

Table 9 lists the partial update current consumption per display type and display quality.

Table 9. Partial Update Current Consumption per Display Type and Display Quality

	POWER AND QUALITY OPTIMIZED			OPTIMIZED for DISPLAY QUALITY			POWER OPTIMIZED		
Type	Time	Avg Current	Q(mC)	Time	Avg Current	Q(mC)	Time	Avg Current	Q(mC)
2.7"	1.43	2.1	3.003	1.63	2	3.26	1.22	2.25	2.745
2.6"	1.4	2.08	2.912	1.64	1.96	3.2144	1.14	2.32	2.6448
2.0"	1.15	2.2	2.53	1.3	2.04	2.652	1.06	2.3	2.438
1.9"	1.36	2.08	2.8288	1.62	1.94	3.1428	1.16	2.27	2.6332
1.44”	1.14	2.3	2.622	1.3	2.12	2.756	1.03	2.4	2.472

6 Getting Started Hardware

6.1 Hardware

To get started, you are required to have:

- A CC3200 LaunchPad
- An EPD BoosterPack
- One of 5 EPD displays by Pervasive Displays (1.44", 1.9", 2.0", 2,6", or 2.7")

6.2 Power Optimizations

The EPD BoosterPack is designed to be power optimized, thus, no further optimization is required.

The CC3200 Launchpad requires an ECO to make it power optimized. This is not mandatory for running the application, just for being able to achieve the power figures described here.

The required changes (for LaunchPad Rev 4.1) are:

- Remove R129, R3 and R20
- Remove J12 jumper
- Disconnect the USB cable from the LaunchPad
- Connect the battery positive to J12 pin 1
- Connect the battery negative to J20 pin 3
- Remove R5
- Wire from R5 (the side near the LDO) to Battery +
- Wire J13 (bottom side, near the CC3200 chip) to Battery +
- Remove J6, J7, J8, J9, J10, J11, J2, and J3 jumpers

NOTE: In this configuration, LEDs will not be functional and programming will not be possible. The reset switch will not work either. This mode is for running the application once its preprogrammed into the serial flash, and not for debugging.

6.3 Using the Pervasive Displays BoosterPack

This design was made to work with the EPD BoosterPack. If such a board is not available, using the Pervasive Displays BoosterPack is possible, but some of the features of this design will not be functional.

When using the Pervasive Displays BoosterPack, low-power figures are not achieved, as the BoosterPack is not optimized for power. In addition, remote modes (remote active and remote low power) may not function properly, and they are redundant because low power cannot be achieved in this configuration. To use remote connection, select "always connected" mode instead.

To use the Pervasive Displays BoosterPack, make the following changes to the design software:

1. Undefine #TI_EPD_BOOSTERPACK and define #PERVASIVE_DISPLAYS_BOOSTERPACK in \EPD_Driver\conf_EPD.h and in \EPD_Driver\Pervasive_Displays_Small_EPD\EPD_hardware.c
2. In \EPD_Driver\Pervasive_Displays_Small_EPD\EPD_hardware.h, change the #define EPD_CS_GPIO from 40 to 28
3. In \EPD_Driver\Pervasive_Displays_Small_EPD\EPD_hardware.c, empty the functions EPD_flash_cs_high and EPD_flash_cs_low (make these functions empty). The design does not use the Pervasive Displays BoosterPack's onboard serial flash to store intermediate images.

In addition, on the LaunchPad, remove jumpers J6, J7, J15 (SoP2), J2, and J3.

Both version 1 and 2 of the Pervasive Displays BoosterPacks may be used using this method.

7 Getting Started Firmware

7.1 User Files

The design accepts custom user graphic files. All graphic files must be in Bitmap (.BMP) format, 1 bit (monochrome).

Two user graphic files are supported:

- **Icons**– Load 1 to 10 icons to the design. TI recommends the size of the icons are 64×64 pixels.
- **Splash screen**– This image is displayed when the system initializes from power off, and is removed once it is ready to receive messages. Any resolution up to the limit of the EPD panel is supported, with one restriction. The X aspect must be 32-bits aligned (for example, 32xY, 64xY, 96xY...). This is due to the BMP's format's padding. A small software change may enable 8-bit alignment.

If a missing icon file is requested by the user, a broken image icon (hard coded) will be displayed instead. If a splash screen is missing, a blank image will be displayed instead. Even if no user graphic file is loaded, the application will still run normally. TI recommends to program at least a couple of custom icons to get the best user experience.

Another user file is *display.txt*. This file contains the default display size connected to the system. The contents of this file is a single digit ASCII number of 0 through 4. If this file does not exist, the software will default to 2.7”. This value can be overridden and saved to the file system through the HTTP interface. The user may also switch display sizes while the system is in power-saving mode (between screen updates), since the EPD panel is shut down at those periods. Once the display has been replaced and reconfigured via the HTTP interface, the new display will refresh automatically.

Table 10 lists the display.txt values per display size.

Table 10. Display.txt Values per Display Size

DISPLAY TYPE	DISPLAY.TXT VALUE
1.44”	0
2.0”	1
2.7”	2
1.9”	3
2.6”	4

Warning: Defining a wrong display size will result in images appearing incorrectly.

7.2 Uniflash

The user files must be added to the existing Uniflash bundle. ***Without programming the Uniflash bundle, the application will not work***.

The Uniflash bundle includes the constant icons, application binary, display.txt file, HTTP page and its collaterals, and optional user graphics files– icons and splash screens.

To use, have Uniflash installed on your PC and run it. Load the .ucf file located at the root of the Uniflash files folder. Figure 16 shows the Uniflash folder example.

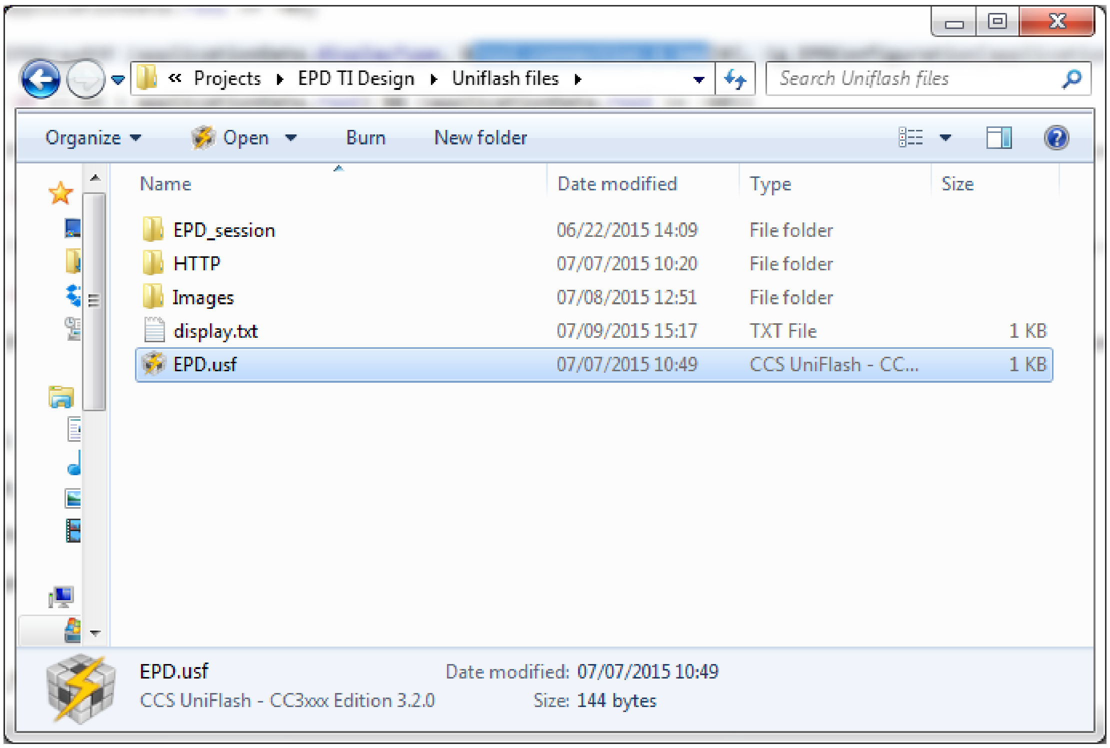


Figure 16. Uniflash Folder Example

1. Determine the COM port number assigned to the LaunchPad by your PC once Uniflash has loaded.
2. Refer to 7.3.1 for details (user logs are output on the same COM port).
3. Make sure the SOP2 jumper on the LaunchPad is assembled to enable programming.
4. Update the COM port field with the correct COM port number if required.
5. If this is the first usage of the LaunchPad, format the serial flash by pressing the format button (approve the default 1-MB serial flash value).
6. Press the reset button on the LaunchPad when prompted.
7. Remove the SOP2 jimper form the LaunchPad once all programming is completed.
8. Press the reset button.
9. The software begins to run.
10. The user is greeted by the splash screen graphics if one was programmed.

Figure 17 shows the Uniflash interface.

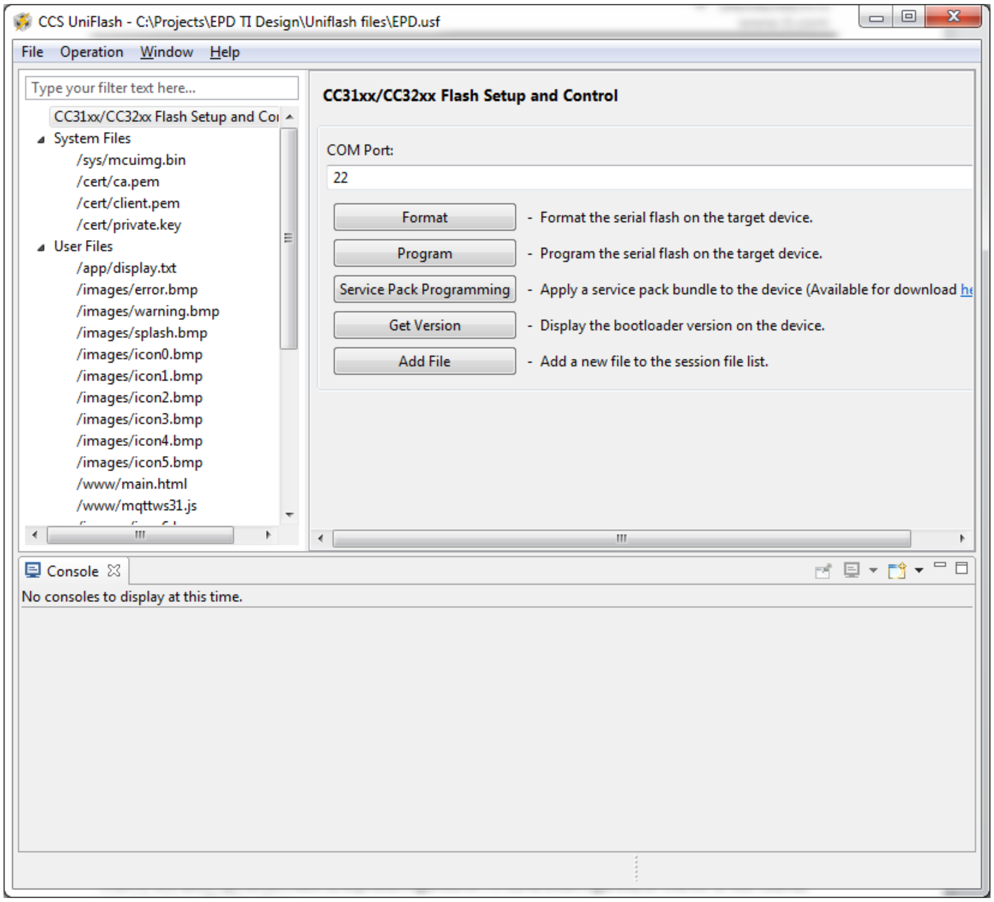


Figure 17. Uniflash Interface

7.3 MQTT Client

A different build of the MQTT client has to be used with this TI design. It is similar to the one supplied with the SDK of CC3200, but built with some different parameters, to comply with the usage of this application.

Make sure to use this custom build, and not the build from SDK. Otherwise, the application fails in all modes except local control.

7.4 Software

Assuming all required information is present in the SimpleLink’s filesystem (including application binary, display size) upon power-up. The EPD panel displays either the splash screen or clear the screen if the splash screen graphics is not present. Then, the embedded network processor attempts to connect to a pre-configured AP. If none is configured, or the AP is not found, the provisioning procedure starts.

NOTE: In case provisioning fails for any reason (such as time out), the application will enter an indefinite hibernate state to conserve power. A message will be displayed on the EPD to instruct the user to reset the device to re-attempt provisioning

Once AP connection is established, the MCU software enters one of the three supported modes. For each mode, an indication icon is displayed on the top left hand side. The reception indicator is located on the top right side, and the panel’s URN (or name) is displayed in the middle. Table 11 lists the displayed icons' description.

Table 11. Displayed Icons' Description

ICON	DESCRIPTION
	Disconnected from the AP. Provisioning may be required.
	Local control mode. Control the EPD panel using SimpleLink’s internal HTTP server. Fastest response time, highest current consumption. Internet connection is not required in this mode.
	Connected to remote MQTT broker. Fast response time, higher current consumption.
	Temporarily always connected. Once the user-configurable timer expires, mode will switch to Intermittently connected. While in this mode, attributes are the same as always connected.
	Intermittently connected. Display will update at configured intervals. This mode consumes the least amount of power, but response time is limited to the configured interval time.
	Excellent reception (above -45 dB)
	Good reception (between -45 dB and 60 dB)
	Average reception (between -60 dB and 80 dB)
	Poor reception (Below -80 dB)

8 Test Setup

Current measurements in this design were completed in two phases:

1. Measuring CC3200 current
2. Measuring the EPD panel's current

The total current consumption is the superposition of the two.

To measure the CC3200 current, a LaunchPad board is used. A current meter is connected instead of jumper 12. This method isolates the CC3200 (and its serial flash) current consumption from the rest of the board. Figure 18 shows the measuring current on the CC3200 LaunchPad.

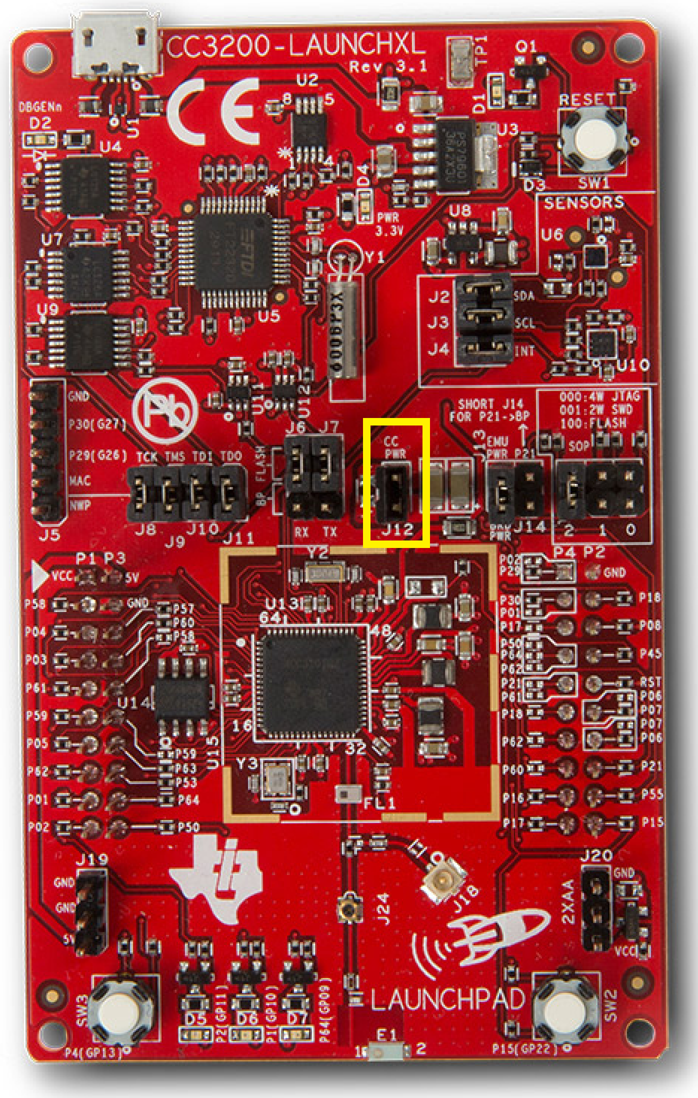


Figure 18. Measuring Current on the CC3200 LaunchPad

For the EPD BoosterPack, a current meter is connected between the LaunchPad's P1.1 (VCC to the BoosterPack) and its counterpart on the BoosterPack. Thus, measuring all current flow to the BoosterPack.

9 Test Data

9.1 EPD Current Consumption

EPD current measurements and analyses are described in Section 5.3.2.

9.2 WLAN Current Consumption

WLAN current measurements and analyses are described in Section 5.3.1.

10 Design Files

10.1 Schematics

To download the schematics for each board, see the design files at http://www.ti.com/tool/TIDC-CC3200_EPD_DESIGN.

shows the EPD BoosterPack Schematic.

10.2 Bill of Materials

To download the bill of materials (BOM), see the design files at http://www.ti.com/tool/TIDC-CC3200_EPD_DESIGN.

Table 12. EPD BoosterPack Bill of Materials

ITEM #	DESIGNATOR	QUANTITY	VALUE	PART NUMBER	MANUFACTURER	DESCRIPTION
1	!PCB 1	1		EPD_BP	Any	Printed Circuit Board
2	C1	1	2.2 uF	0805ZD225KAT2A	AVX	CAP, CERM, 2.2 µF, 10 V, +/- 10%, X5R, 0805
3	C2, C3, C5, C8, C9, C10, C11, C15, C16, C18	10	2.2 uF	0805YD225KAT2A	AVX	CAP, CERM, 2.2 µF, 16 V, +/- 10%, X5R, 0805
4	C4, C6, C7, C13, C14	5	2.2 uF	C2012X5R1H225K125AB	TDK	CAP, CERM, 2.2 µF, 50 V, +/- 10%, X5R, 0805
5	C12	1	1 uF	08053D105KAT2A	AVX	CAP, CERM, 1 µF, 25 V, +/- 10%, X5R, 0805
6	C17, C19	2	0.1 uF	06033C104KAT2A	AVX	CAP, CERM, 0.1 µF, 25 V, +/- 10%, X7R, 0603
7	D1	1	30 V	BAT54SWT1G	Fairchild Semiconductor	Diode, Schottky, 30 V, 0.2 A, SOT-323
8	FID1, FID2, FID3	3		N/A	N/A	Fiducial mark. There is nothing to buy or mount.
9	J1	1	Right Angle Socket	FH12-40S-0.5SH(55)	Hirose Electric Co. Ltd.	Connector, FFC/FFP Receptacle, 40-Pos, 1 Row, 0.5mm Pitch, SMT
10	J2, J3	2		SSW-110-22-F-D-VS-K	Samtec	Connector, Receptacle, 100mil, 10×2, Gold plated, SMD
11	Q1, Q2, Q3, Q7	4	60 V	2N7002KW	Fairchild Semiconductor	MOSFET, N-CH, 60 V, 0.31 A, SOT-323
12	Q4, Q6	2	-50 V	BSS84W-7-F	Diodes Inc.	MOSFET, P-CH, -50 V, -0.13 A, SOT-323
13	Q5	1	20 V	SI2377EDS-T1-GE3	Vishay-Siliconix	MOSFET, P-CH, 20 V, -4.4 A, SOT-23
14	R1	1	2.0 k	CRCW08052K00JNEA	Vishay-Dale	RES, 2.0 k, 5%, 0.125 W, 0805
15	R2, R3, R4	3	100 k	CRCW0805100KJNEA	Vishay-Dale	RES, 100 k, 5%, 0.125 W, 0805

Table 12. EPD BoosterPack Bill of Materials (continued)

ITEM #	DESIGNATOR	QUANTITY	VALUE	PART NUMBER	MANUFACTURER	DESCRIPTION
16	R5	1	33 k	CRCW080533K0JNEA	Vishay-Dale	RES, 33 k, 5%, 0.125 W, 0805

10.3 PCB Layout Recommendations

There is no special requirement for layout or routing. The form factor of the board was designed to match that of the LaunchPad beneath it, and the flat-cable connector was placed so that all display sizes will have a mechanical support from the bare-PCB. Mechanical support is also the reason SMT connectors were used.

10.3.1 Layout Prints

To download the layout prints for each board. see the design files at http://www.ti.com/tool/TIDC-CC3200_EPD_Design.

10.4 Altium Project

To download the Altium project files, see the design files at http://www.ti.com/tool/TIDC-CC3200_EPD_Design.

10.5 Assembly Drawings

To download the assembly drawings for each board, see the design files at http://www/ti.com/tool/TIDC-CC3200_EPD_Design.

10.6 Software Files

To download the software files, see the design files at http://www.ti.com/tool/TIDC-CC3200_EPD_DESIGN.

11 Terminology

WLAN: Wirelss LAN

EPD: Electrophoretic Display

AP: Access Point

12 About the Author

ASAF CARMELI is a systems engineer at Texas Instruments where he defines hardware modules and software flows for the SimpleLink devices. Asaf brings to this role 10 years in system engineering.

Revision History

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

Changes from Original (January 2016) to A Revision **Page**

IMPORTANT NOTICE FOR TI REFERENCE DESIGNS

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