

TI Designs Bluetooth Low Energy (BLE) Selfie Taker With MSP430™ Design Guide



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

[TIDM-SELFIETAKER](#)
[MSP430FR5969](#)

Tool Folder Containing Design Files
Product Folder



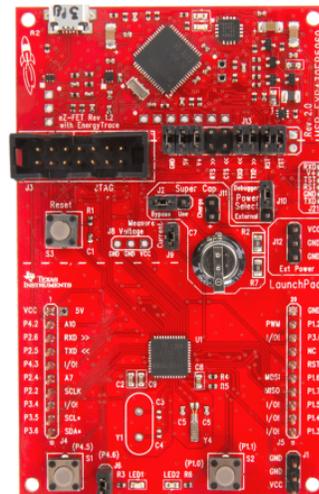
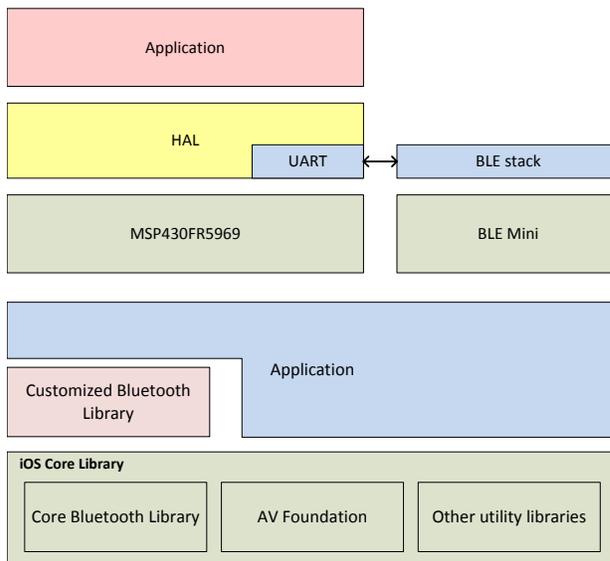
[ASK Our E2E Experts](#)
[WEBENCH® Calculator Tools](#)

Design Features

- TI's BLE-Stack™ 1.3 Software Stack
- FRAM
- Ultra-Low Power
- Smartphone Application

Featured Applications

- BLE Communication Application



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

All trademarks are the property of their respective owners.

1 System Description

Bluetooth® low energy (BLE) is a new wireless personal area network aimed toward low-energy consumption areas of application, such as healthcare, fitness, security, and home entertainment. The application utilizes the RedBearLab BLE Mini to add BLE capability to an MSP430™ LaunchPad™ and keep the low-power features at the same time.

The demo titled “SelfieTaker” provides an alternative for selfie enthusiasts. Without the limit of the length of the arm, the user can remotely capture a picture in a BLE-enabled smartphone by pressing a button on the MSP430 LaunchPad.

1.1 MSP430FR5969 LaunchPad

The MSP430FR5969 (Wolverine) LaunchPad is an easy-to-use evaluation module for the MSP430FR5969 microcontroller. The evaluation module includes on-board emulation for programming, debugging, and energy measurements. The module also features embedded FRAM, which is a non-volatile memory known for its ultra-low power, high endurance, and high-speed write access when comparing to flash memory. These features make the MSP430FR5969 an ideal LaunchPad for BLE communication applications.

1.2 RedBearLab BLE Mini

The RedBearLab BLE Mini is a small BLE development board incorporated with TI’s CC2410 Bluetooth Low Energy System-on-Chip (SoC). The RedBearLab BLE Mini only requires a serial port for communication and supports all of the major platforms that have universal asynchronous receiver and transmitter (UART) interfaces, including the MSP430. In addition, the firmware in the BLE Mini can be updated through a USB.

1.3 Apple—iPhone® 6

This design uses the Apple iPhone® 6 because it supports Bluetooth 4.0 Low Energy. The iPhone 4S, iPhone 5, iPhone 5C, and iPhone 5S are also applicable to this design.

1.4 Code Composer Studio™ v5.5

Code Composer Studio v5.5 is a main integrated development environment (IDE) tool for the MSP430FR5969 LaunchPad development including programming, downloading, and debugging.

1.5 XCode® 6

Apple's XCode® 6 is an IDE tool for the iPhone BLE application development. The IDE is free, but the user must purchase a digital certificate to download the application to the physical iPhone device.

1.6 BLE Stack

The BLE stack used in this design is TI's BLE stack v1.3.

2 Block Diagram

2.1 MSP430 Software Architecture

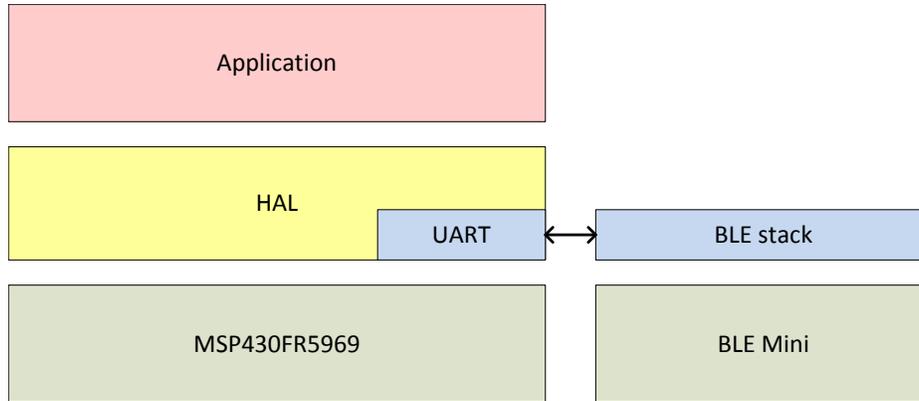


Figure 1. MSP430 Software Architecture

The basic software running on the MSP430FR5969 LaunchPad is divided into two layers: the hardware abstraction layer (HAL) layer and the application layer. The HAL layer encapsulates all of the hardware-related operations, such as serial peripheral interface, UART, and I²C communication; clock system initialization; and so forth. The UART communication in the HAL is specifically for BLE data sending and receiving to and from the BLE Mini. The uppermost layer is the application including all of the user logic codes.

2.2 iPhone Software Architecture

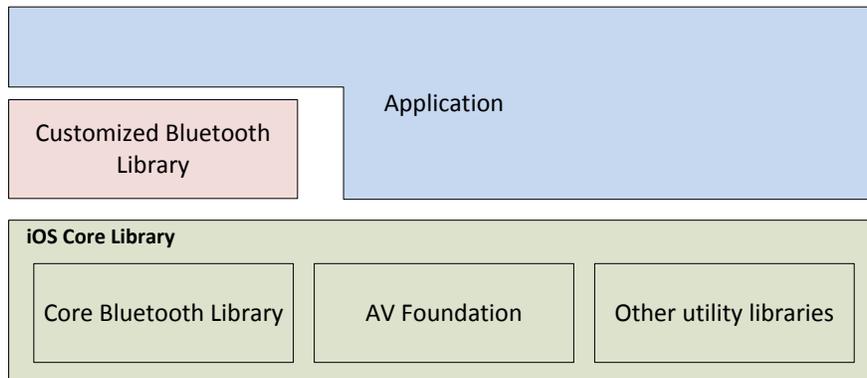


Figure 2. iPhone Software Architecture

The iPhone-side application has the two-layer architecture. The bottom layer includes all of the iPhone core libraries (AV Foundation for the camera feature, Core Bluetooth library, and network library for the HTTP GET query). The customized Bluetooth library sitting on the top of the Core Bluetooth library encapsulates a lot of the trivial Bluetooth communication work like scanning, services and characteristics discovery, reading and writing BLE data, and so forth. The application utilizes a customized Bluetooth library for Bluetooth communication and uses other Core libraries for remaining functions.

3 System Design Theory

3.1 BLE Message Protocol

The current BLE message protocol is to send out all the characters without padding any additional characters. For the purpose of extensibility, the protocol could be easily extended to add the data type, data length, and any other meta information besides the raw data. The packet size also is saved if sending integer/float type data because the packet size does not need to convert character format, which requires more space.

The selfie application uses two strings as the command to trigger the camera operation. The string “captureEvent” is to trigger the camera to take a picture, while string “cameraSwitchEvent” is to trigger the camera to switch between front and back camera.

3.2 MSP430 LaunchPad Software Design

3.2.1 Architecture

The MSP430 Launchpad software architecture has an infinite loop in the main function that listens to the event post. All of the interrupt service routines (ISR) post a corresponding event and wake up the CPU to process all of the pending events. This process minimizes the time consumed in the ISR functions to avoid delays when serving other ISRs. For example, the ISR button sends a button-pressed event to the global event container, which wakes up the CPU upon the ISR exit. Therefore, the main loop resumes checking the event container and executing the corresponding event handler, which is registered before the main loop.

3.2.2 Send the BLE message

The BLE message sending triggers by clicking two buttons (S1 and S2) on the MSP430FR5969 LaunchPad. As stated previously, when the user press one button, the ISR button wakes up the CPU to execute the UART transmission task. The content transferred through the UART is wrapped as a BLE packet and sent out to the smartphone. Each button corresponds to the different content to send out. Pressing S1 executes the send-camera switch command (“cameraSwitchEvent”), while pressing S2 executes the send camera capture command (“captureEvent”).

3.3 Smartphone Software Design

3.3.1 Architecture

The BLE smartphone-side programming requires the Apple iOS CoreBluetooth Library. The project also includes customized Bluetooth library wrapping CoreBluetooth library routines and provides more application-friendly application programming interfaces (API) to the upper layer. For example, users do not have to call `scanForPeripheralsWithServices` and then implement protocol function `didDiscoverPeripheral` to get all of the discovered peripherals. The user only needs to initialize the customized Bluetooth instance and then all of the discovered peripherals automatically fill in the public peripherals array.

3.3.2 Connect to a Device

The main user interface (UI) is a table view. The UI lists all of the nearby discoverable Bluetooth devices that provide `RBL_SERVICE_UUID` (defined in the `BLEDefines.h`). Clicking one Bluetooth device on the list connects the iPhone to the selected device. If the connection is successful, the user will see the camera view from the front camera.

3.3.3 Receive the BLE message

The Core Bluetooth library provides basic functions to receive the BLE message. The customized Bluetooth library improves the existing functionality by adding a notification mechanism so that any view that registers in the notification center can receive the BLE new message notification. The camera view, which registers in the notification center once loaded, executes the corresponding camera operations when the BLE message is received.

```
(BLE.c, didUpdateValueForCharacteristic())
//post an event to the notification center
NSDictionary * userInfoDict = [NSDictionary dictionaryWithObject:msgStr
forKey:@"cameraIndicator"];
[[NSNotificationCenter defaultCenter] postNotificationName:@"selfieNotification" object:nil
userInfo:userInfoDict];

(TIViewController.c, viewDidLoad())
//register to the notification center
[[NSNotificationCenter defaultCenter] addObserver:self selector:@selector(didReceiveNotification:)
name:@"selfieNotification" object:nil];

(TIViewController.c, didReceiveNotification())
//receive the notification
NSString *resourceName = [[notification userInfo] objectForKey:@"cameraIndicator"];

//capture the picture if the command string is "captureEvent"
if([resourceName isEqualToString:@"captureEvent"]) {
    [self capImage];
}
//switch the front/back camera if the command string is "cameraSwitchEvent"
else if([resourceName isEqualToString:@"cameraSwitchEvent"]) {
    FrontCamera = !FrontCamera;
    [self initializeCamera:FrontCamera];
}
```

4 Getting Started Hardware

The BLE Mini is connected with an MSP430FR5969 LaunchPad through a UART interface. The default UART setting is: baud rate 57600, 8 data bits, no parity, one stop bits, and no flow control.

5 Getting Started Firmware

The firmware of the BLE Mini is pre-loaded with TI's BLE Stack v1.3 and UART interface setting. Users can directly connect the development board and MCU with a BLE Mini without modifying the BLE Mini firmware. The firmware on the MSP430FR5969 LaunchPad is described above and provided in this design.

6 Test Setup

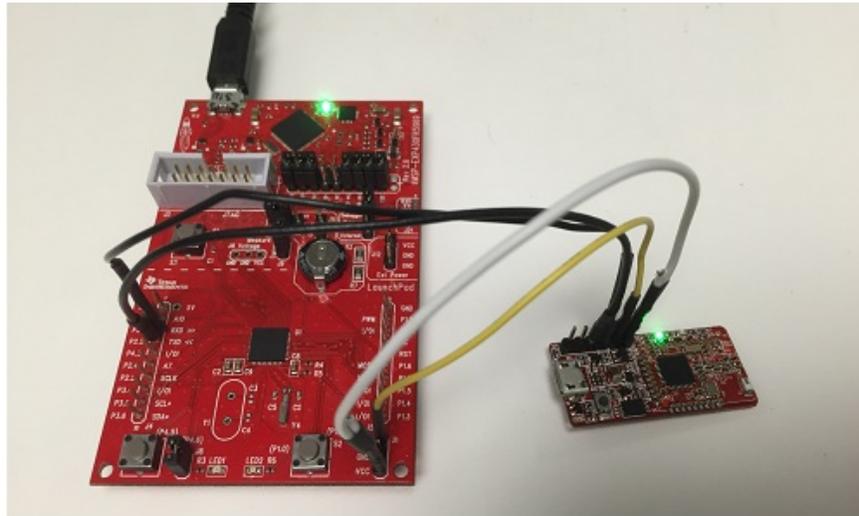


Figure 3. Hardware Setup

1. Connect the MSP430FR5969 LaunchPad with the BLE Mini through a UART interface. Make sure that the green light-emitting diode (LED) is on for the BLE Mini (Figure 3).

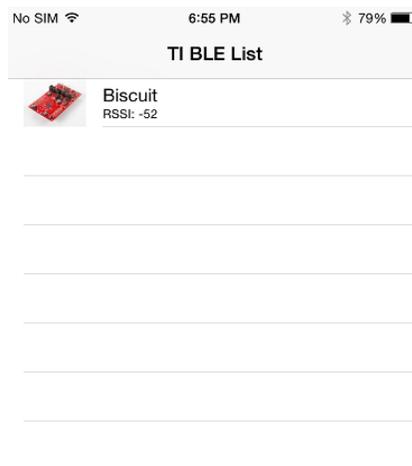


Figure 4. BLE List in Selfie Taker App

2. Open up the "Selfie Taker" application in the iPhone and select the current broadcasting BLE Mini (Figure 4).
3. Push the LaunchPad onboard button *S1* to capture the picture. Push the LaunchPad onboard button *S2* to toggle the front or back camera.

7 Power Consumption

The power consumption for the MSP430 LaunchPad is considerably low. In the selfie demo, the MSP430FR5969 LaunchPad is in the low-power mode (LPM0) most of time and is only triggered by UART communication when receiving a BLE message.

Table 1. Selfie Demo Power Consumption

	POWER (mW)			CURRENT (mA)			CR2032 BATTERY LIFE (DAYS)
	MEAN	MIN	MAX	MEAN	MIN	MAX	
Broadcast state	34	31	83	9	9	23	0.8
Connection state	34	32	60	9	9	17	0.8

8 Design Files

8.1 Schematics

To download the schematics, see the design files at [TIDM-SELFIETAKER](#).

8.2 Bill of Materials

To download the bill of materials (BOM), see the design files at [TIDM-SELFIETAKER](#).

8.3 Layout Prints

To download the layout prints for each board, see the design files at [TIDM-SELFIETAKER](#).

8.4 Software Files

To download the software files, see the design files at [TIDM-SELFIETAKER](#).

9 References

1. Texas Instruments, "MSP430FR58xx and MSP430FR59xx Family", User's Guide, ([SLAU367](#)).
2. Texas Instruments, "MSP-EXP430FR5969 LaunchPad™ Development Kit", User's Guide, ([SLAU535](#)).
3. RedBearLab, "BLE Mini", <http://redbearlab.com/blemini/>.
4. Apple, Inc, "iOS Developer Library", <https://developer.apple.com/library/ios/>.

10 About the Author

Yang Lu is a Software Design Engineer at Texas Instruments. He is responsible for developing the MSP430™ Driver Library and testing framework. Yang earned his Master of Science in Embedded Software Engineering from Carnegie Mellon University in Pittsburgh, PA.

IMPORTANT NOTICE FOR TI REFERENCE DESIGNS

Texas Instruments Incorporated ("TI") reference designs are solely intended to assist designers ("Buyers") who are developing systems that incorporate TI semiconductor products (also referred to herein as "components"). Buyer understands and agrees that Buyer remains responsible for using its independent analysis, evaluation and judgment in designing Buyer's systems and products.

TI reference designs have been created using standard laboratory conditions and engineering practices. **TI has not conducted any testing other than that specifically described in the published documentation for a particular reference design.** TI may make corrections, enhancements, improvements and other changes to its reference designs.

Buyers are authorized to use TI reference designs with the TI component(s) identified in each particular reference design and to modify the reference design in the development of their end products. HOWEVER, NO OTHER LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE TO ANY OTHER TI INTELLECTUAL PROPERTY RIGHT, AND NO LICENSE TO ANY THIRD PARTY TECHNOLOGY OR INTELLECTUAL PROPERTY RIGHT, IS GRANTED HEREIN, including but not limited to any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI components or services are used. Information published by TI regarding third-party products or services does not constitute a license to use such products or services, or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

TI REFERENCE DESIGNS ARE PROVIDED "AS IS". TI MAKES NO WARRANTIES OR REPRESENTATIONS WITH REGARD TO THE REFERENCE DESIGNS OR USE OF THE REFERENCE DESIGNS, EXPRESS, IMPLIED OR STATUTORY, INCLUDING ACCURACY OR COMPLETENESS. TI DISCLAIMS ANY WARRANTY OF TITLE AND ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, QUIET ENJOYMENT, QUIET POSSESSION, AND NON-INFRINGEMENT OF ANY THIRD PARTY INTELLECTUAL PROPERTY RIGHTS WITH REGARD TO TI REFERENCE DESIGNS OR USE THEREOF. TI SHALL NOT BE LIABLE FOR AND SHALL NOT DEFEND OR INDEMNIFY BUYERS AGAINST ANY THIRD PARTY INFRINGEMENT CLAIM THAT RELATES TO OR IS BASED ON A COMBINATION OF COMPONENTS PROVIDED IN A TI REFERENCE DESIGN. IN NO EVENT SHALL TI BE LIABLE FOR ANY ACTUAL, SPECIAL, INCIDENTAL, CONSEQUENTIAL OR INDIRECT DAMAGES, HOWEVER CAUSED, ON ANY THEORY OF LIABILITY AND WHETHER OR NOT TI HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, ARISING IN ANY WAY OUT OF TI REFERENCE DESIGNS OR BUYER'S USE OF TI REFERENCE DESIGNS.

TI reserves the right to make corrections, enhancements, improvements and other changes to its semiconductor products and services per JESD46, latest issue, and to discontinue any product or service per JESD48, latest issue. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All semiconductor products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its components to the specifications applicable at the time of sale, in accordance with the warranty in TI's terms and conditions of sale of semiconductor products. Testing and other quality control techniques for TI components are used to the extent TI deems necessary to support this warranty. Except where mandated by applicable law, testing of all parameters of each component is not necessarily performed.

TI assumes no liability for applications assistance or the design of Buyers' products. Buyers are responsible for their products and applications using TI components. To minimize the risks associated with Buyers' products and applications, Buyers should provide adequate design and operating safeguards.

Reproduction of significant portions of TI information in TI data books, data sheets or reference designs is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Buyer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products, and any use of TI components in its applications, notwithstanding any applications-related information or support that may be provided by TI. Buyer represents and agrees that it has all the necessary expertise to create and implement safeguards that anticipate dangerous failures, monitor failures and their consequences, lessen the likelihood of dangerous failures and take appropriate remedial actions. Buyer will fully indemnify TI and its representatives against any damages arising out of the use of any TI components in Buyer's safety-critical applications.

In some cases, TI components may be promoted specifically to facilitate safety-related applications. With such components, TI's goal is to help enable customers to design and create their own end-product solutions that meet applicable functional safety standards and requirements. Nonetheless, such components are subject to these terms.

No TI components are authorized for use in FDA Class III (or similar life-critical medical equipment) unless authorized officers of the parties have executed an agreement specifically governing such use.

Only those TI components that TI has specifically designated as military grade or "enhanced plastic" are designed and intended for use in military/aerospace applications or environments. Buyer acknowledges and agrees that any military or aerospace use of TI components that have **not** been so designated is solely at Buyer's risk, and Buyer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI has specifically designated certain components as meeting ISO/TS16949 requirements, mainly for automotive use. In any case of use of non-designated products, TI will not be responsible for any failure to meet ISO/TS16949.