Application Report

SLVA413–April 2010

# MSP430™ Microcontroller Software for Handheld Fuel Gauges and Battery Authentication Products

Randy Wu
Digital Field Applications

**ABSTRACT**

The Texas Instruments (TI) battery management portfolio includes fuel gauges and authentication ICs for the handheld and mobile device market. To assist in the development of this market, this application note provides a set of software drivers for interfacing the one and two-cell battery management devices with the TI MSP430™ ultra-low power microcontroller.

The examples provided with this application note give the user a basic functioning MCU host software for communicating with the following products: bq26100, bq26150, bq26500, bq27000, bq27010, bq27200, bq27210, bq27500, bq27505, bq27510, and bq27541. Various combinations of standard MSP430 and bq26xxx/bq27xxx Evaluation Modules (EVMs) were used as the hardware for the development of the code examples provide with this application note, giving users of these products working hardware and software reference designs to use for creating custom products.

**Contents**



**List of Figures**

**List of Tables**



# 1 Introduction

This application note provides a set of individual code examples which run on a standard pair of MSP430 and bq Evaluation Modules connected to each other. Having a basic reference design covering both hardware and software enables user of the bq devices with functioning microcontroller code which implements the communications interface of the specific bq device. Each code example consists of a stand-alone set of project files for a cost-effective device from the Texas Instruments MSP430 microcontroller family.

Each specific example can be modified to work with similar MSP430 and bq devices. It is recommended to first select the example which targets the specific bq device to be used, since the microcontroller code can be ported to any MSP430 device. The complete set of examples demonstrates all of the following serial communications busses between the MSP430 (Master) and the bq device (Slave):

- SDQ™ (single wire)
- HDQ™ (single wire)
- I2C Master using the MSP430 Universal Serial Interface (USI) module
- I2C Master using the MSP430 Universal Serial Communications Interface (USCI) module
- I2C Master using GPIO software driver for bit-banging

All of the code provided with this application note can be used for quick prototyping as well as in production-quality code in systems where the MSP430 is used as the host microcontroller interfacing to various bq26xxx and bq27xxx devices.

The complete source code and project files have been provided for the two embedded firmware Integrated Development Environments for the MSP430 microcontroller platform – IAR Systems Embedded Workbench and Texas Instruments Code Composer Studio.

# 2 MSP430 + bq26xxx/bq27xxx Code Examples

## *2.1 MSP430F23x0 + bq26100 via SDQ™ Interface*

The bq26100 provides a method to authenticate battery packs, ensuring that only packs manufactured by authorized sub-contractors are used in the end application. The security is achieved using the SHA-1 hash function inside the HMAC construction. A unique 128-bit key is stored in each bq26100 device, allowing the host to authenticate each pack.

The bq26100 communicates to the system over a one-wire bi-directional serial interface called SDQ. The 5-kbits/sec SDQ bus interface reduces communications overhead in the external microcontroller. The bq26100 also derives power over the SDQ bus line via an external capacitor. The SDQ interface is a one-wire bus designed for high-speed two-way communications between the bq device and a host microcontroller. The MSP430F23x0 device is used as the microcontroller due to the need for a decent amount of RAM required to perform SHA-1/HMAC authentication routines.

There are several types of MSP430 development boards available. The MSP430 device used for this example is the MSP430F23x0, a 40-pin ultra-low power microcontroller with up to 32 I/O pins, 32KB flash memory, and 2KB RAM. This device was chosen because it has sufficient RAM to execute the SHA-1/HMAC algorithm. The part number for the development kit is MSP-FET430U23X0, which includes an MSP430F2370 device. The standard bq26100EVM is connected to the MSP430 EVM board.

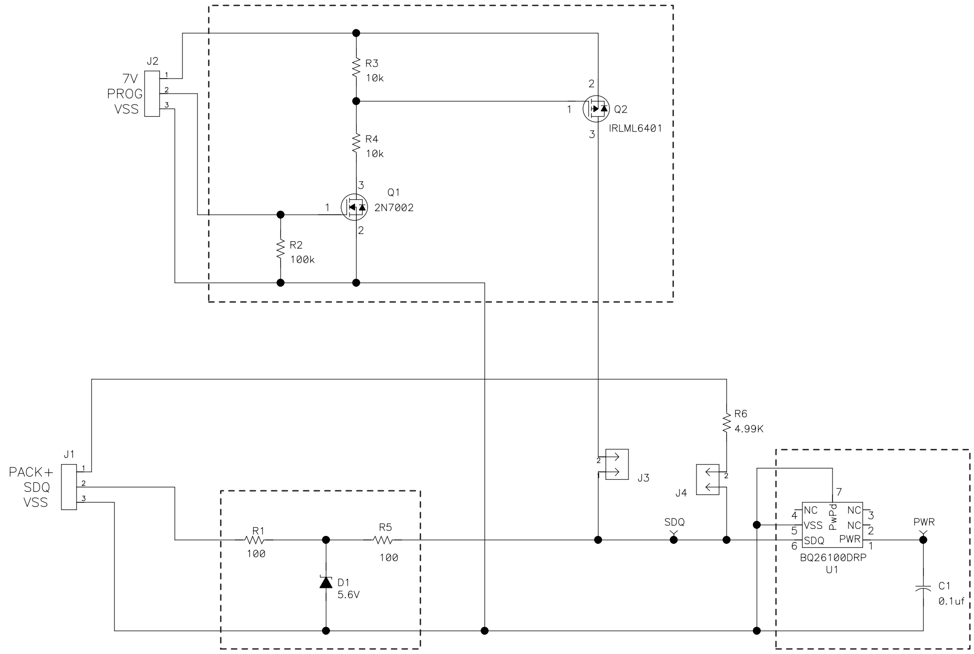


**Figure 1. bq26100EVM Board Schematic**

A total of three direct-wire connections must be made between the MSP430 and bq26100 demo boards. See Table 1 for the required connections and jumper setting.

**Table 1. MSP430 and bq26100 ConnectionsTable 1**

| MSP430 | CONNECT | bq26100 (EVM) |
|---|---|---|
| P1.1 | ←→ | SDQ (J1-2) |
| VCC | ←→ | PACK+ (J1-1) |
| VSS | ←→ | VSS (J1-3) |

## WARNING

**The bq26100 can operate at up to 7.7 VDC; however, the MSP430 maximum operating voltage is 3.6 VDC. When connecting the MSP430 board to the bq26100 EVM, ensure that the MSP430 never exceeds more than 3.6 VDC.**

When the two Evaluation Modules have been connected, the user can load and run the MSP430 code. The MSP430 will generate 20 bytes of random values and then write them into the bq26100. The MSP430 will calculate its own SHA-1/HMAC value based on the random values which were generated.

Once the MSP430 has computed its own SHA-1/HMAC value, then the bq26100 can compute its own SHA-1/HMAC value. The MSP430 will send a command to the bq26100 to run the calculation. The SHA-1/HMAC result is read from the bq26100 and compared to the MSP430's value. If the two values match, authentication is successful and the LED on the MSP430 EVM will light up as a visual indicator of success.

By default, the private 128-bit key should be all 0's for a fresh (unprogrammed) bq26100 device, so the example code assumes that this is the case. If the 128-bit key has been re-programmed in the bq26100 device, make sure to modify the code in the main function so that the keys match in both the MSP430 and bq26100; otherwise, authentication will not be successful.

### 2.2 MSP430F20xx + bq26150 via HDQ Interface

The bq26150 provides a method to authenticate battery packs, ensuring that only packs manufactured by authorized sub-contractors are used in the end application. The bq26150 uses a 96-bit unique device ID, device unique 16-bit seed, and a 16-bit device specific CRC to provide security. The device ID, CRC seed, and CRC polynomial coefficients are stored securely in each bq26150 device, allowing the host to authenticate each pack.

The bq26150 communicates to the system over a one-wire, bidirectional serial interface. The 5 Kbits/sec HDQ bus interfaces reduces communications overhead in the external microcontroller. The bq26150 also uses the HDQ bus to charge an external capacitor that provides power to the bq26150.

There are several types of MSP430 development boards available. The MSP430F20xx is a 14-pin microcontroller with up to 10 I/O pins, 2KB flash memory, and 128B RAM. The part number for the development kit is MSP-FET430U14, which includes an MSP430F2013 device. The standard bq26150EVM is connected to the MSP430 EVM board.

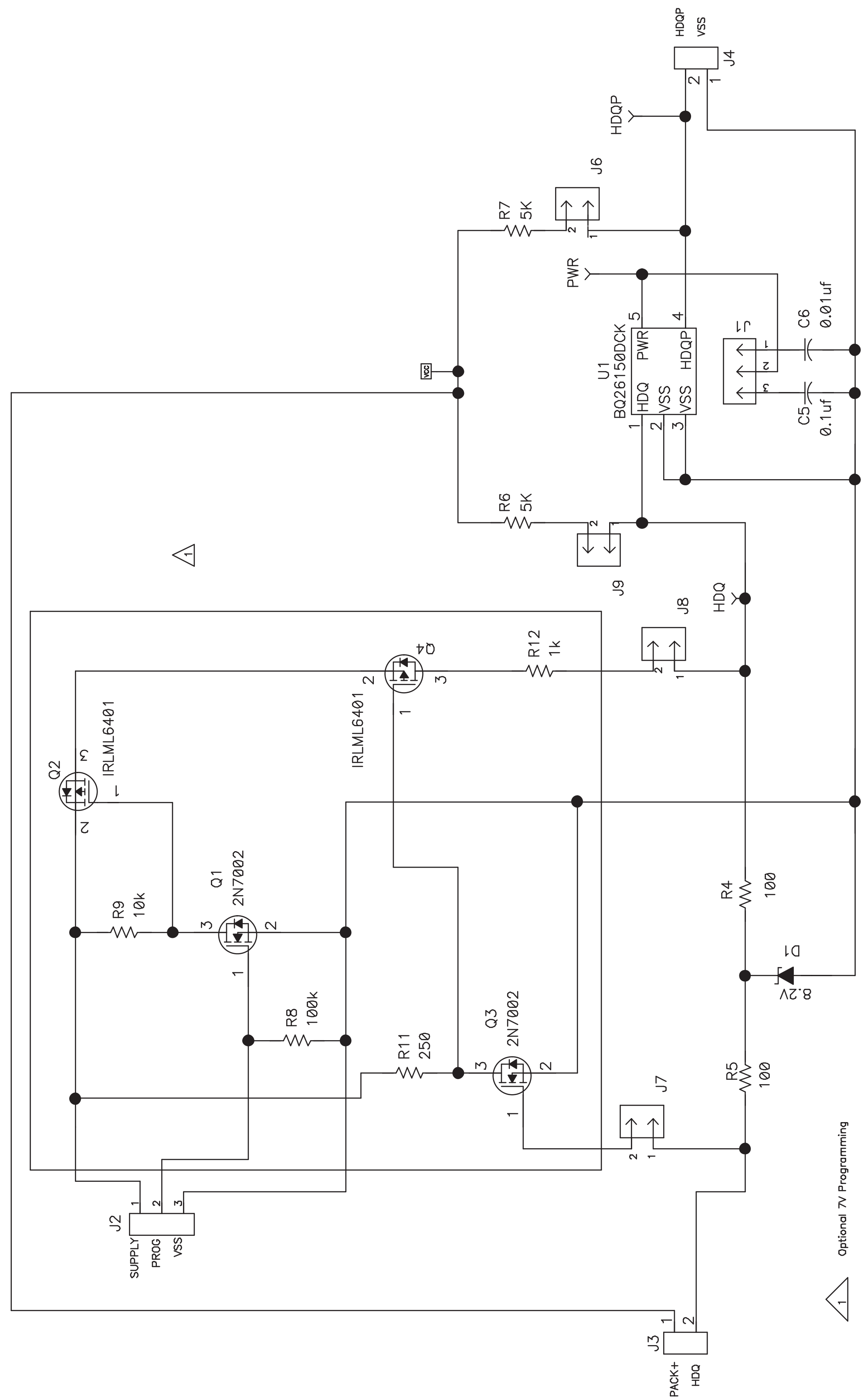


**Figure 2. bq26150EVM Board Schematic**

A total of three direct-wire connections must be made between the MSP430 and bq26150 demo boards. See Table 2 for the required connections, bq26150EVM jumper setting, and pull-up resistor for HDQ.

**Table 2. MSP430 and bq26150 Connections**(1)(2)

| MSP430 | CONNECT | bq26150 (EVM) |
|---|---|---|
| P1.1 | ←→ | HDQ (J3-2) |
| VCC | ←→ | PACK+ (J3-1) |
| VSS | ←→ | VSS (J4-3) |

(1) The J9 jumper on the bq26150EVM must be installed to enable the HDQ pull-up resistor

(2) A jumper must be installed at J1 to short J1-1 and J1-2 to provide power to the bq26150 device via the HDQ bus

**WARNING**

**The bq26150 can operate at up to 7.7 VDC; however, the MSP430 maximum operating voltage is 3.6 VDC. When connecting the MSP430 board to the bq26150 EVM, ensure that the MSP430 never exceeds more than 3.6 VDC.**

**Before the bq26150 device can be used with the MSP430, *all* encrypted *and* plaintext Device ID, polynomial seed, and polynomial coefficient bytes must be programmed using the EV2300 programmer module and bq Evaluation Software. The provided MSP430 code example assumes that the plaintext values are encrypted by XOR'ing each plaintext byte with 0xFF. Similarly, decryption also involves XOR'ing with the 0xFF crypto key, since XOR'ing any value with 0xFF twice results in the same value. If a different encryption/decryption scheme is desired, the MSP430 source code must be modified accordingly; otherwise, authentication will not be successful.**

When the two Evaluation Modules have been connected, the user can load and run the MSP430 code. The MSP430 will generate 4 bytes of random values and then write them into the bq26150. The MSP430 will read the encrypted Device ID, polynomial coefficients, and polynomial seed bytes and then decrypt each byte to obtain the plaintext values. The MSP430 will use the decrypted plaintext data to calculate its own CRC16 value based on the random values which were generated.

Once the MSP430 has computed its own CRC16 value, then the bq26150 can compute its own CRC16 value. The MSP430 will send a command to the bq26150 to run the calculation. Then the CRC16 result is read from the bq26150 and compared to the MSP430's value. If the two values match, authentication is successful and the LED on the MSP430 EVM will light up as a visual indicator of success.

### *2.3 MSP430F20xx + bq26500 via HDQ Interface*

The bq26500, the first in the bqJUNIOR family of advanced gas gauge device for handheld applications, is a highly accurate standalone single-cell Li-Ion and Li-Pol battery capacity monitoring and reporting device targeted at space limited portable applications. The device monitors a voltage drop across a small current sense resistor connected in series with the battery to determine charge and discharge activity of the battery. Compensations for battery temperature, self-discharge, and rate of discharge are applied to the charge counter to provide available capacity across a wide range of operating conditions.

Battery capacity is automatically recalibrated, or learned, in the course of a discharge cycle from full to empty. Internal registers include available capacity, cell temperature and voltage, state-of-charge, and status and control registers. The bq26500 can operate directly from single-cell Li-Ion and Li-Pol batteries and communicates to the system over a one-wire bi-directional serial interface. The 5-kbits/s HDQ bus interface reduces communication overhead in the external microcontroller.

There are several types of MSP430 development boards available. The MSP430F20xx is a 14-pin microcontroller with up to 10 I/O pins, 2KB flash memory, and 128B RAM. The part number for the development kit is MSP-FET430U14, which includes an MSP430F2013 device. The standard bq26500EVM is connected to the MSP430 EVM board.

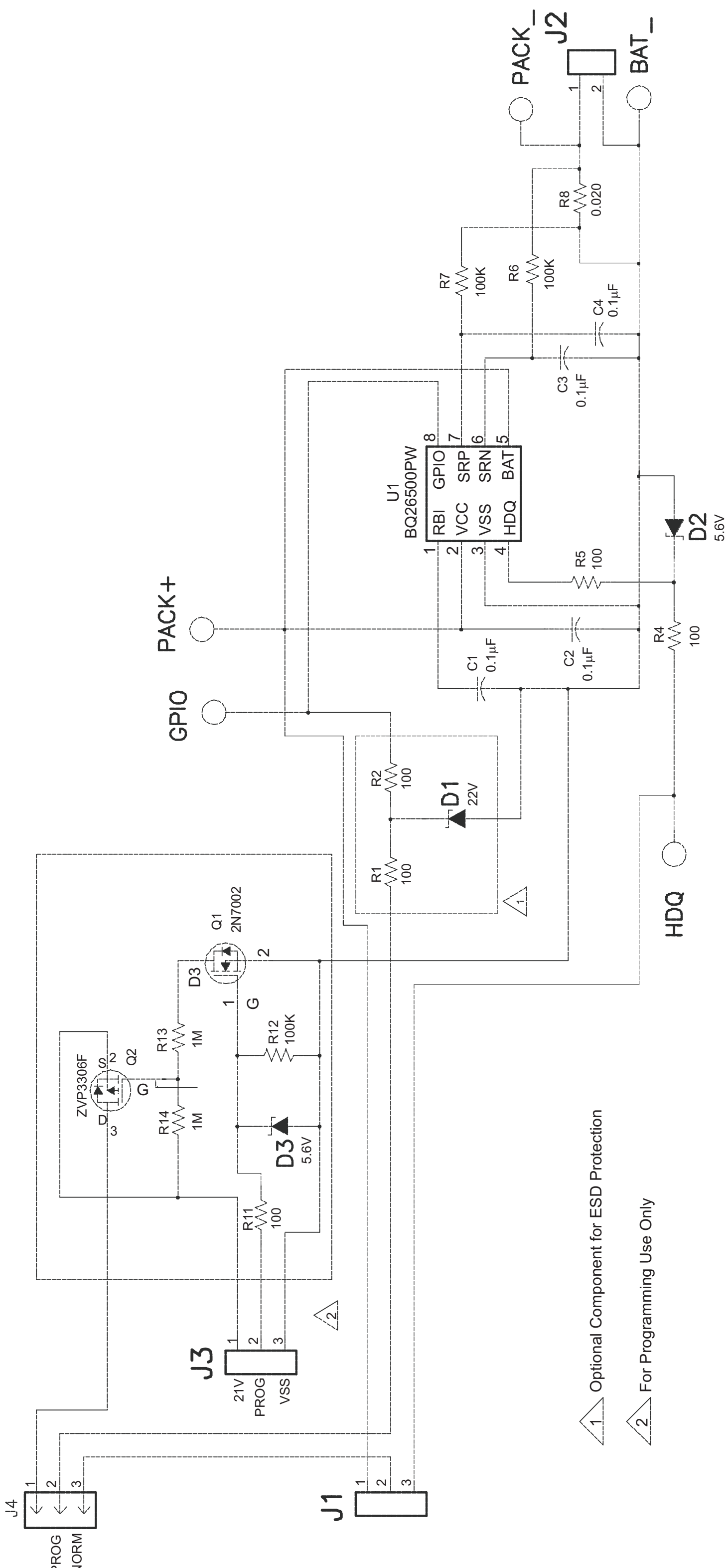

**Figure 3. bq26500EVM Board Schematic**

A total of three direct-wire connections must be made between the MSP430 and bq26500 demo boards. SeeTable 3 for the required connections and HDQ pull-up resistor.

**Table 3. MSP430 and bq26500 Connections(1)**

| MSP430 | CONNECT | bq26500 (EVM) |
|---|---|---|
| P1.1 | ← → | HDQ (J1-3) |
| VCC | ← → | BAT (J1-1) |
| VSS | ← → | PACK– (J2-1) |

(1) A 10kΩ pull-up resistor must be installed between PACK+ (J1-1) and HDQ (J1-3)

**WARNING**

**The bq26500 can operate at up to 7.0 VDC; however, the MSP430 maximum operating voltage is 3.6 VDC. When connecting the MSP430 board to the bq26500 EVM, ensure that the MSP430 never exceeds more than 3.6 VDC.**

When the two Evaluation Modules are connected, the user can load and run the MSP430 code. The MSP430 will periodically read out the temperature and command the bq26500 to toggle a GPIO pin to indicate changes in the reading. If the change between consecutive temperature readings is greater than 1 Kelvin, the GPIO pin will be set to high; otherwise, it will be set to low. In addition, the AtRate is set to 100 steps which yields 357 µV (since each steps corresponds to 3.57 µV).

### 2.4 MSP430F20xx + bq27010 via HDQ Interface

The bqJUNIOR series are highly accurate stand-alone single-cell Li-Ion and Li-Pol battery capacity monitoring and reporting devices targeted at space-limited, portable applications. The IC monitors a voltage drop across a small current sense resistor connected in series with the battery to determine charge and discharge activity of the battery. Compensations for battery age, temperature, self-discharge, and discharge rate are applied to the capacity measurements to provide available time-to-empty information across a wide range of operating conditions. Battery capacity is automatically recalibrated, or learned, in the course of a discharge cycle from full to empty. Internal registers include current, capacity, time-to-empty, state-of-charge, cell temperature and voltage, status, and more.

The bqJUNIOR can operate directly from single-cell Li-Ion and Li-Pol batteries and communicate to the system over a HDQ one-wire interface.

There are several types of MSP430 development boards available. The MSP430F20xx is a 14-pin microcontroller with up to 10 I/O pins, 2KB flash memory, and 128B RAM. The part number for the development kit is MSP-FET430U14, which includes an MSP430F2013 device. The standard bq27010EVM is connected to the MSP430 EVM board.

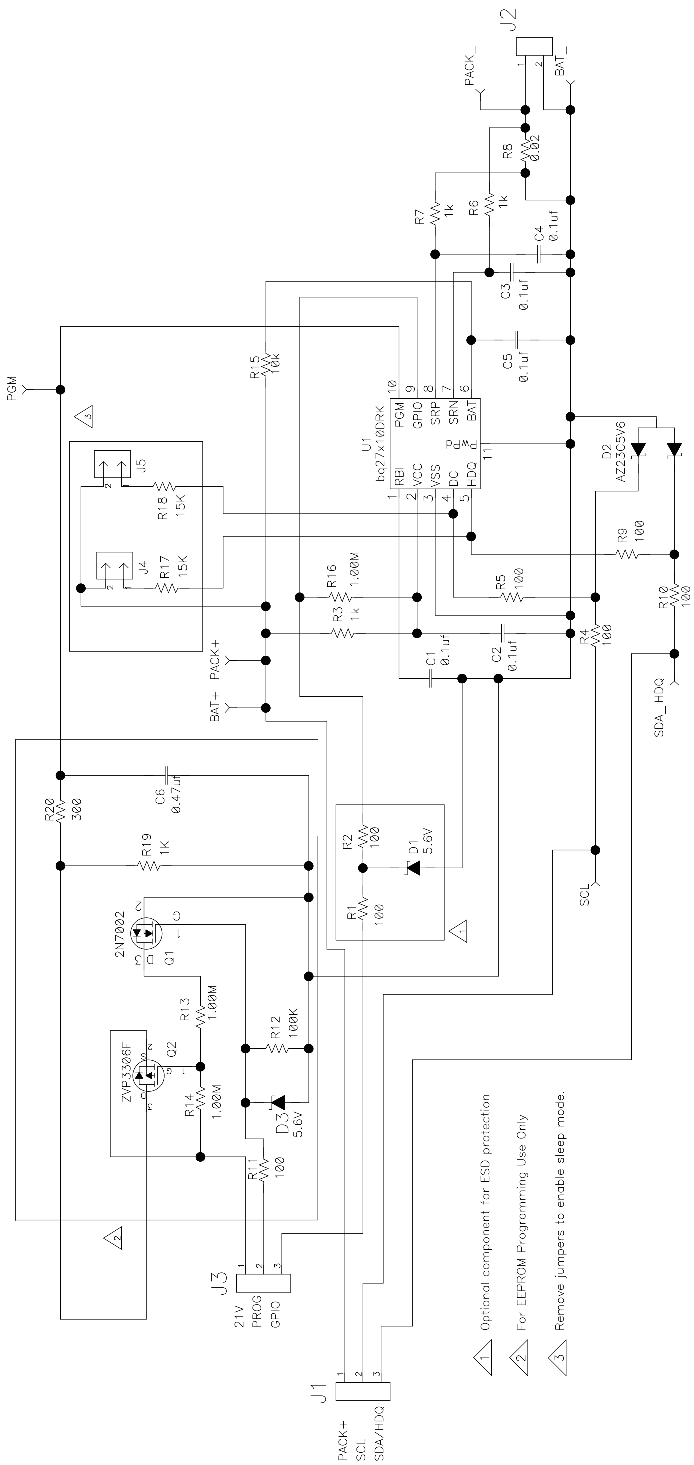


**Figure 4. bq27010EVM Board Schematic**

A total of three direct-wire connections must be made between the MSP430 and bq27010 demo boards. See Table 4 for the required connections and HDQ pull-up resistor.

**Table 4. MSP430 and bq27010 Connections**[1]

| MSP430 | CONNECT | bq27010 (EVM) |
|---|---|---|
| P1.1 | ←→ | HDQ (J1-3) |
| VCC | ←→ | PACK+ (J1-1) |
| VSS | ←→ | PACK– (J2-1) |

(1) A 10kΩ pull-up resistor must be installed between PACK+ (J1-1) and HDQ (J1-3)

**WARNING**

**The bq27010 can operate at up to 7.0 VDC; however, the MSP430 maximum operating voltage is 3.6 VDC. When connecting the MSP430 board to the bq26100 EVM, ensure that the MSP430 never exceeds more than 3.6 VDC.**

When the two Evaluation Modules have been connected, the user can load and run the MSP430 code. The MSP430 will perform the following commands:

- Turn ON the bq27010 GPIO open drain output
- Read Status Flags, Relative State of Charge, Voltage, Temperature
- Set AtRate to 357 µV
- Read AtRate to verify the setting
- Read AtRate Time To Empty
- Perform and Read Offset Measurement
- Turn OFF the bq27010 GPIO open drain output

If every command is successful, the MSP430 will turn on the LED by setting P1.0 high.

### 2.5 MSP430F20xx + bq27210 via I²C Interface

The bqJUNIOR series are highly accurate stand-alone single-cell Li-Ion and Li-Pol battery capacity monitoring and reporting devices targeted at space-limited, portable applications. The IC monitors a voltage drop across a small current sense resistor connected in series with the battery to determine charge and discharge activity of the battery. Compensations for battery age, temperature, self-discharge, and discharge rate are applied to the capacity measurements to provide available time-to-empty information across a wide range of operating conditions. Battery capacity is automatically recalibrated, or learned, in the course of a discharge cycle from full to empty. Internal registers include current, capacity, time-to-empty, state-of-charge, cell temperature and voltage, status, and more.

The bqJUNIOR can operate directly from single-cell Li-Ion and Li-Pol batteries and communicate to the system over I²C interface.

There are several types of MSP430 development boards available. The MSP430F20xx is a 14-pin microcontroller with up to 10 I/O pins, 2KB flash memory, and 128B RAM. The part number for the development kit is MSP-FET430U14, which includes an MSP430F2013 device. The standard bq27210EVM is connected to the MSP430 EVM board.

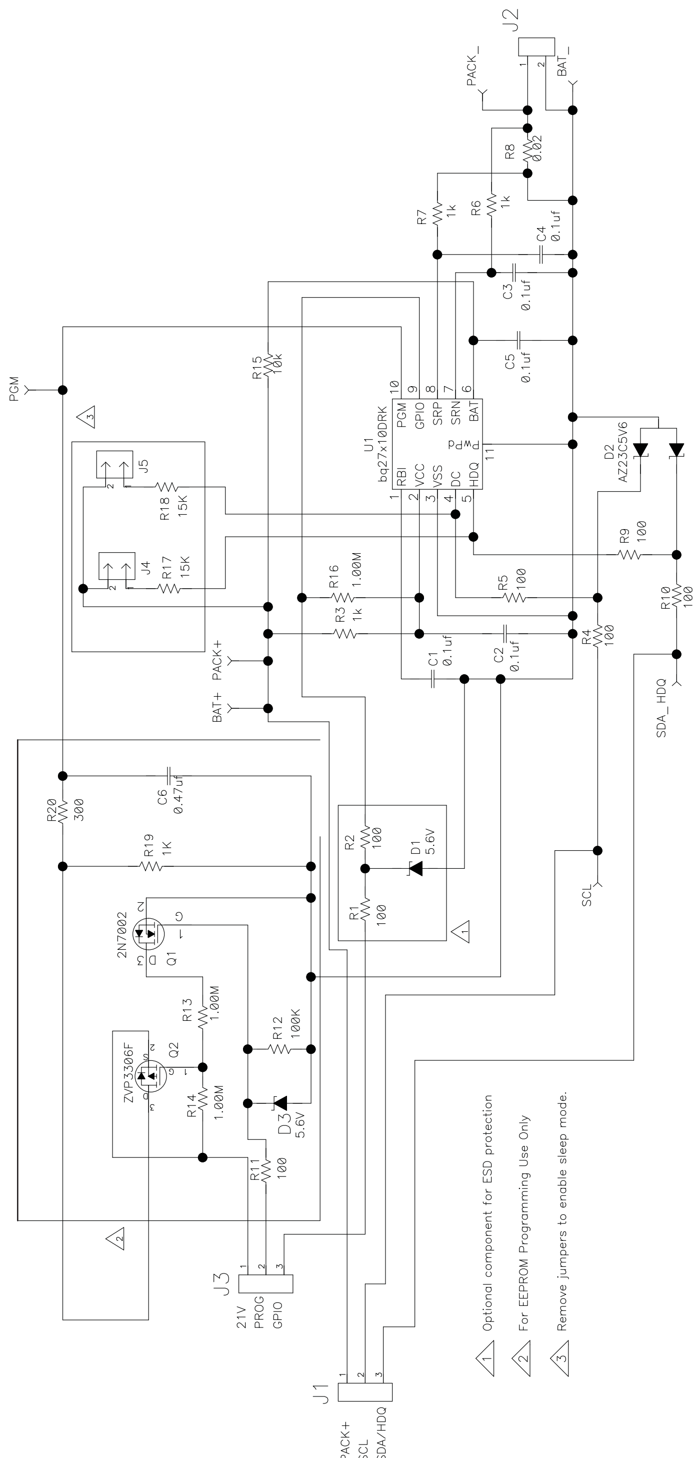


**Figure 5. bq27210EVM Board Schematic**

A total of four direct-wire connections must be made between the MSP430 and bq27210 demo boards. See Table 5 for the required connections and jumper settings.

**Table 5. MSP430 and bq27210 Connections(1)**

| MSP430 | CONNECT | bq27210 (EVM) |
|---|---|---|
| P1.6 | ←→ | SCL (J1-2) |
| P1.7 | ←→ | SDA (J1-3) |
| VCC | ←→ | PACK+ (J1-1) |
| VSS | ←→ | PACK– (J2-1) |

(1) The J4 and J5 jumpers on the bq27210EVM must be installed to enable the I2C pull-up resistors

**WARNING**

**The bq27210 can operate at up to 7.0 VDC; however, the MSP430 maximum operating voltage is 3.6 VDC. When connecting the MSP430 board to the bq26100 EVM, ensure that the MSP430 never sees more than 3.6 VDC.**

When the two Evaluation Modules have been connected, the user can load and run the MSP430 code. The MSP430 will perform the following commands:

- Turn ON the bq27210 GPIO open drain output
- Read Status Flags, Relative State of Charge, Voltage, Temperature
- Set AtRate to 357 µV
- Read AtRate to verify the setting
- Read AtRate Time To Empty
- Perform and Read Offset Measurement
- Turn OFF the bq27210 GPIO open drain output

If every command is successful, the MSP430 will turn on the LED by setting P1.0 high.

The MSP430 code to implement an I$^2$C™ Master interface can be configured to either use the MSP430 Universal Serial Interface (USI) for I$^2$C support in hardware, or it can be configured to use two General Purpose I/O pins to emulate an I$^2$C Master interface in the software via bit-bang fashion. To find out which scheme is being used, and/or to change between one scheme and the other, locate the following lines of code in the "msp430_bq27500.c" source file:

```
-------------------------------------------------------------------------------------
/**** USER CONFIGURATION: Select ONE of the following I2C physical layers *****/
#define _I2C_USI                                      // OPTION #1: Use MSP430 USI module
//#define _I2C_BITBANG                                // OPTION #2: Use Standard GPIO pins
-------------------------------------------------------------------------------------
```

Whichever line of code that is not commented out at compile time enables the specified mode of operation. In the above case, the MSP430 USI module is being used as the I$^2$C Master interface.

## 2.6 MSP430F20xx + bq27500 via I$^2$C Interface

The bq27500 system-side Li-Ion battery fuel gauge is a microcontroller peripheral that provides fuel gauging for single-cell Li-Ion battery packs. The device requires little system microcontroller firmware development. The bq27500 resides on the system main board, and manages an embedded battery (non-removable) or a removable battery pack.

The bq27500 uses the patented Impedance Track™ algorithm for fuel gauging, and provides information such as remaining battery capacity (mAh), state-of-charge (%), run-time to empty (min.), battery voltage (mV), and temperature (°C).

Battery fuel gauging with the bq27500 requires only PACK+, (P+), PACK-, (P-), and Thermistor (T) connections to a removable battery pack or embedded battery.

There are several types of MSP430 development boards available. The MSP430F20xx is a 14-pin microcontroller with up to 10 I/O pins, 2KB flash memory, and 128B RAM. The part number for the development kit is MSP-FET430U14, which includes an MSP430F2013 device. The standard bq27500EVM is connected to the MSP430 EVM board.

**WARNING**

**Before proceeding, please update the bq27500 firmware to "V130" or higher. See the bq27500-V130 home page to download the "Updating Firmware with the bq2750x and EVM" application note as well as the "bq27500-V130 Senc File".**

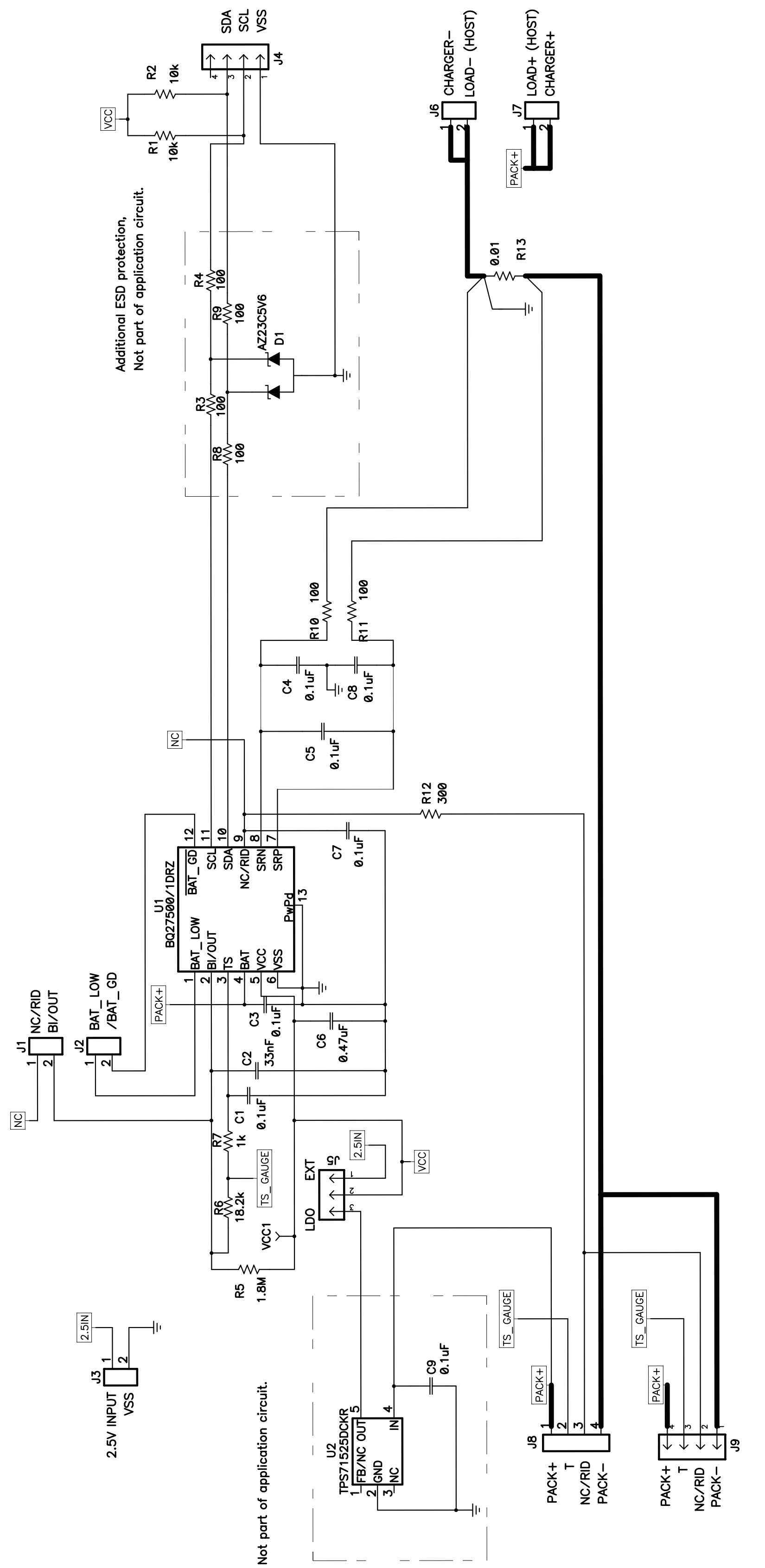


**Figure 6. bq27500EVM Board Schematic**

A total of four direct-wire connections must be made between the MSP430 and bq27500 demo boards. See Table 6 for the required connections.

**Table 6. MSP430 and bq27500 Connections**

| MSP430 | CONNECT | bq27500 (EVM) |
|---|---|---|
| P1.6 | ←→ | SCL (J4-2) |
| P1.7 | ←→ | SDA (J4-3) |
| VCC | ←→ | PACK+ (J9-1) |
| VSS | ←→ | PACK– (J9-4) |

When the two Evaluation Modules have been connected, the MSP430 code can be loaded and run. The MSP430 will perform the following commands:

- Read Temperature, Voltage, AtRate Time To Empty, State of Charge, Design Capacity, Device Name Length, Device Name
- Set AtRate to -100 mA
- Read AtRate to verify the setting
- Write 32 bytes to Manufacturer Info Block A
- Read back 32 bytes from Manufacturer Info Block A to verify previous write command

If every command is successful, the MSP430 will turn on the LED by setting P1.0 high.

The MSP430 code to implement an $I^2C$ Master interface can be configured to either use the MSP430 Universal Serial Interface (USI) for $I^2C$ support in hardware, or it can be configured to use two General Purpose I/O pins to emulate an $I^2C$ Master interface purely in software via bit-bang fashion. To find out which scheme is being used, and/or to change between one scheme and the other, locate the following lines of code in the "msp430_bq27500.c" source file:

```
-------------------------------------------------------------------------------
/**** USER CONFIGURATION: Select ONE of the following I2C physical layers *****/
#define _I2C_USI                                  // OPTION #1: Use MSP430 USI module
//#define _I2C_BITBANG                            // OPTION #2: Use Standard GPIO pins
-------------------------------------------------------------------------------
```

Whichever line of code that is not commented out at compile time enables the specified mode of operation. In the above case, the MSP430 USI module is being used as the $I^2C$ Master interface.

## 2.7 MSP430F23x0 + bq27541 via HDQ and I²C Interfaces

The bq27541 Li-Ion battery fuel gauge is a microcontroller peripheral that provides fuel gauging for single-cell Li-Ion battery packs. The device requires little system microcontroller firmware development for accurate battery fuel gauging. The bq27541 resides within the battery pack or on the system's main-board with an embedded battery (non-removable).

The bq27541 uses the patented Impedance Track algorithm for fuel gauging, and provides information such as remaining battery capacity (mAh), state-of-charge (%), run-time to empty (min.), battery voltage (mV), and temperature (°C).

The bq27541 also features integrated support for secure battery pack authentication, using the SHA-1/HMAC authentication algorithm.

There are several types of MSP430 development boards available. The MSP430 device used for this example is the MSP430F23x0, a 40-pin ultra-low power microcontroller with up to 32 I/O pins, 32KB flash memory, and 2KB RAM. This device was chosen because it has sufficient RAM to execute the SHA-1/HMAC algorithm. The part number for the development kit is MSP-FET430U23X0, which includes an MSP430F2370 device. The standard bq27541EVM is connected to the MSP430 EVM board.

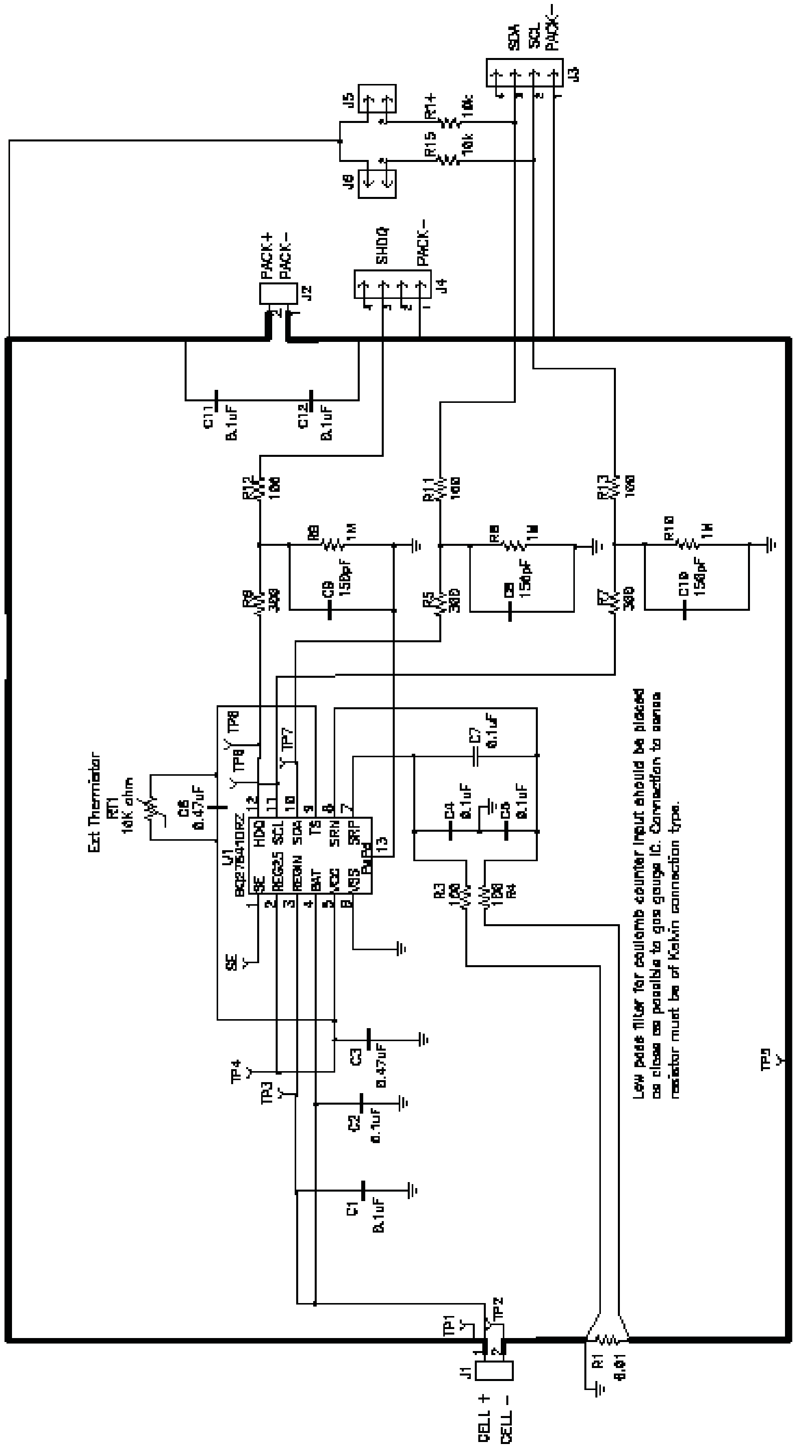


**Figure 7. bq27541EVM Board Schematic**

When the two Evaluation Modules have been connected, the user can load and run the MSP430 code. The MSP430 will perform the following commands:

- Read Temperature, Voltage, AtRate Time To Empty, State of Charge, Design Capacity, Device Name Length, and Device Name
- Set AtRate to -100 mA
- Read AtRate to verify the setting

- Write 32 bytes of data to Manufacturer Info Block A
- Read back 32 bytes from Manufacturer Info Block A to verify previous write command
- Generate 160-bit random challenge (including checksum) and upload the bytes
- Issue authentication command and read back the resulting 160-bit digest
- Calculate what the digest should be based on knowledge of the 128-bit private key
- Compare the calculated "host" digest vs. the "slave" digest provided by the bq device

If every command is successful, the MSP430 will turn on the LED by setting P1.0 high.

### 2.7.1 HDQ Interface for bq27541

A total of three direct-wire connections must be made between the MSP430 and bq27541 demo boards. See Table 7 for the required connections and pull-up resistor for HDQ.

**Table 7. MSP430 and bq27541 Connections for HDQ(1)**

| MSP430 | CONNECT | bq27541 (EVM) |
|---|---|---|
| P1.1 | ←→ | SHDQ (J4-3) |
| VCC | ←→ | PACK+ (J4-1) |
| VSS | ←→ | PACK– (J3-4) |

(1) A 10kΩ pull-up resistor must be installed between PACK+ (J3-1) and SHDQ (J4-3)

### 2.7.2 HDQ Interface for bq27541

The MSP430 code to implement an I²C Master interface can be configured to either use the MSP430 Universal Communications Serial Interface (USCI) for I²C support in hardware, or it can be configured to use two General Purpose I/O pins to emulate an I²C Master interface in the software via bit-bang fashion. To find out which scheme is being used, and/or to change between one scheme and the other, locate the following lines of code in the "msp430_bq27541_I2C.c" source file:

```
-------------------------------------------------------------------------------------
/**** USER CONFIGURATION: Select ONE of the following I2C physical layers *****/
#define _I2C_USCI                                      // OPTION #1: Use MSP430 USCI module
//#define _I2C_BITBANG                                 // OPTION #2: Use Standard GPIOs
-------------------------------------------------------------------------------------
```

Whichever line of code that is not commented out at compile time enables the specified mode of operation. In the above case, the MSP430 USCI module is being used as the I²C Master interface.

A total of four direct-wire connections must be made between the MSP430 and bq27541 demo boards. See Table 8 for the required connections and based on whether the USCI module or GPIO pins are to be used.

**Table 8. MSP430 and bq27541 Connections for I²C**

| MSP430 | CONNECT | bq27541 (EVM) |
|---|---|---|
| P1.6 (GPIO) or P3.2 (USCI) | ←→ | SCL (J3-2) |
| P1.7 (GPIO) or P3.1 (USCI) | ←→ | SDA (J3-3) |
| VCC | ←→ | PACK+ (J4-1) |
| VSS | ←→ | PACK– (J3-4) |

## 3 Authentication

This section provides the user with a brief introduction to the authentication functionality of the bq26100, bq26150, and bq27541 which is demonstrated by the included sample code. While the bq26100 and bq26150 are standalone authentication ICs (with additional features such as user-programmable non-volatile memory), the bq27541 is a full-featured pack-side fuel gauge which also has a SHA-1/HMAC authentication feature built-in.

Any of these ICs can be used in battery packs, battery chargers, adapters, dongles, or any peripheral requiring a level of security to prevent counterfeit replacements from being introduced. For batteries in particular, requiring such authentication using a secret key can protect users from potentially unsafe packs that may not have the typically included safety or protection circuits. Counterfeit packs may also not meet the performance or functionality standards of the original OEM batteries and could result in a degraded user experience. Any safety mishap or unsatisfactory user experience would damage the reputation of the OEM, and not the counterfeit pack maker.

To identify authentic battery packs or peripherals, one option is to use an EPROM (such as the bq2022A) which is programmed with a specific code or ID. The host can read this code or ID and if it matches the expected value it will allow the battery or peripheral to be used. The use of an EPROM chip can deter counterfeiters based on the additional cost required to implement it, but a sufficiently motivated counterfeiter can easily snoop the code or ID and reproduce it with their own EPROM or microcontroller.

A more secure method is to use an encrypted authentication architecture using an IC such as the bq26100, bq26150, or bq27541. An IC used for this architecture is programmed with secret values known only to the OEM. The host processor will send a random challenge to the authentication IC in the peripheral each time it must be authenticated. The authentication IC will use the secret values and its authentication transform to calculate a response and return this message to the host. If the response matches the host's own calculations, the peripheral is successfully authenticated. If the response does not match, the host can take appropriate action such as disallowing the use of the peripheral.

In the sample code provided, the MSP430 acts as the challenger/host and sends a random challenge to the authentication IC. If the bq26100 or bq27541 are used as the authentication IC, the authentication transform is the SHA-1 algorithm with the specific implementation detailed in application note SLUA389. The basic scheme used is shown in Figure 8.

The OEM must program the bq26100 or bq27541 with the desired SHA-1 key at their factory, or for high volume production, arrangements can be made directly with TI for the key to be pre-programmed using TI's Secure B-to-B Protocol.

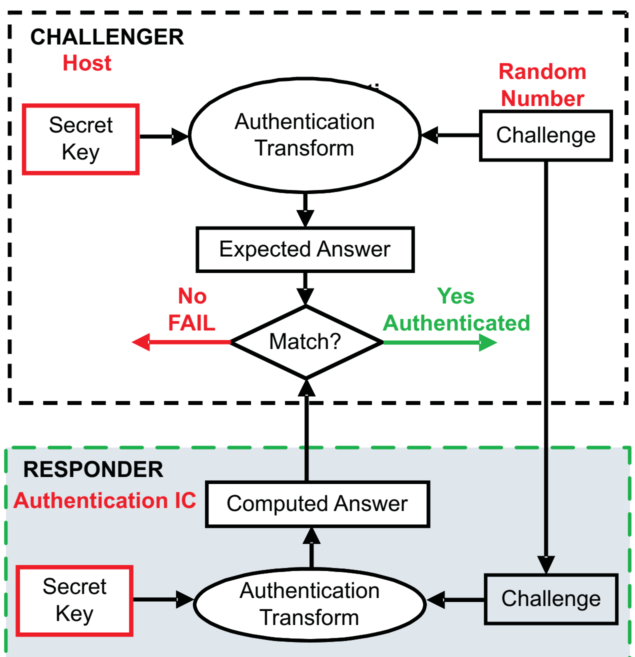


**Figure 8. Authentication Scheme for bq26100 or bq27541 (SHA-1/HMAC)**

If the bq26150 is used as the authentication IC, the authentication transform is done using a CRC algorithm and the device ID, CRC seed, and CRC polynomial coefficients are encrypted before being read by the host. The host decrypts these to obtain the plaintext ID to be used by the authentication transform. This is illustrated in Figure 9.

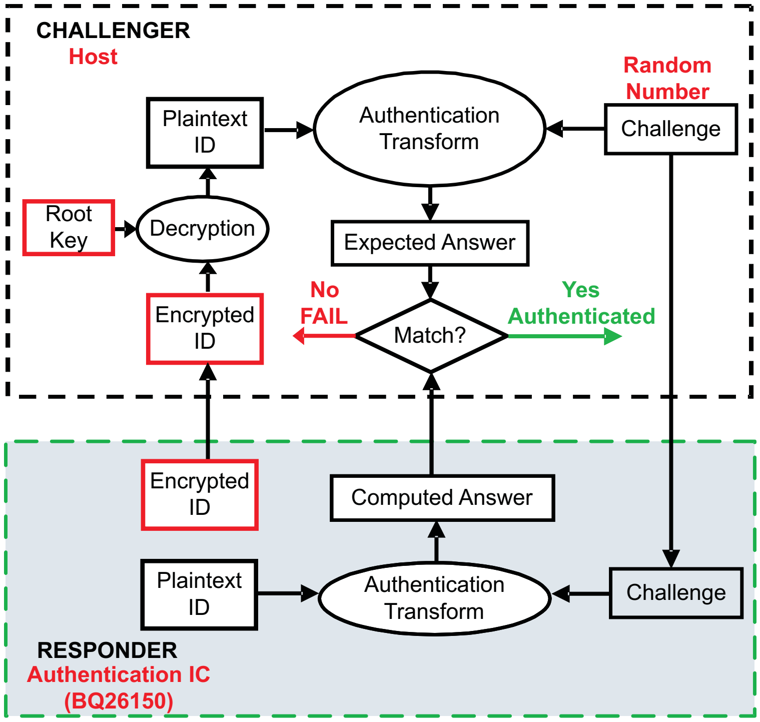


**Figure 9. Authentication Scheme for bq26150 (CRC)**

The OEM must program the bq26150 with the desired encrypted and plaintext values at their factory, or for high volume production, arrangements can be made directly with TI for this data to be pre-programmed using TI's Secure B-to-B Protocol.

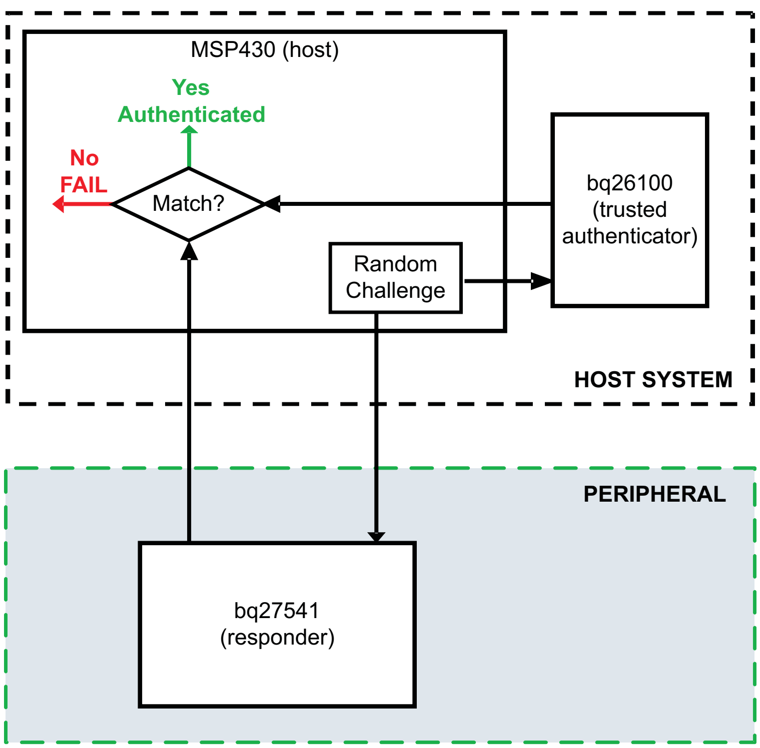


**Figure 10. Authentication Scheme (host does not perform authentication transform)**

A final scheme using both a bq26100 and a bq7541 is shown in Figure 10. Such a scheme might be desirable for a number of reasons.

1. An economic reason might arise in the case where a lower-cost host processor is used which does not have the memory or CPU horsepower required to perform the SHA-1/HMAC authentication transform.
2. A security reason might arise for the case where the OEM prefers the SHA-1 key to be stored and accessed off-board from the host processor.

Since the bq26100 and bq27541 both have secure (unreadable) memory locations and busses for the SHA-1 key and calculations, the security of the key is higher when isolated to these dedicated ICs than when stored in a general-purpose microcontroller with typical flash memory. The confidentiality of the key can also be promoted by having the bq26100 and bq27541 keys programmed together at another location than the general PCB assembly/test site. More details on the bq26100, bq26150, and bq27541 functions, authentication transforms, and usage can be found in their respective data sheets and application notes at the Texas Instruments web site.

## 4 Conclusion

The Texas Instruments (TI) battery management portfolio includes a number of fuel gauges and authentication ICs for the handheld and mobile device market. To assist in the fast-paced development environment of this market, this application note provides a set of software drivers for interfacing the one and two-cell battery management devices with TI's MSP430 ultra-low power microcontroller.

The examples provided with this application note will give users basic functioning MCU host software for communicating with the following products: bq26100, bq26150, bq26500, bq27000, bq27010, bq27200, bq27210, bq27500, bq27505, bq27510, and bq27541. Various combinations of standard MSP430 and bq26xxx/bq27xxx Evaluation Modules (EVMs) were used as the hardware vehicles for the development of the code examples provide with this application note, giving users of these products working hardware and software reference designs to use for creating custom products.

All of the code provided with this application note can be used for quick prototyping as well as in production-quality code in systems where the MSP430 is used as the host microcontroller interfacing to various bq26xxx and bq27xxx devices.

The complete source code and project files have been provided for the two most popular embedded firmware Integrated Development Environments for the MSP430 microcontroller platform – IAR Systems Embedded Workbench and Texas Instruments Code Composer Studio.

## 5 References


1. *SHA-1/HMAC Based Security and Authentication IC with SDQ Interface* (SLUS696)
2. *How to Implement SHA-1/HMAC Authentication for bq26100* (SLUA389)
3. *bq26100 Evaluation Software* (SLUU244)
4. *Battery Pack Security and Authentication IC for Portable Applications* (SLUS641)
5. *bq26150 Evaluation Software* (SLUU223)
6. *HDQ Communications Basics* (SLUA408)
7. *HDQ Protocol Implementation with MSP430* (SLAA196)
8. *bq26500 Single Cell Battery Fuel Gauge Evaluation Module* (SLUU174)
9. *Single-Cell Li-Ion and Li-Pol Battery Gas Gauge IC for Handheld Applications* (SLAS567)
10. *Single-Cell Li-Ion and Li-Pol Battery Gas Gauge IC for Portable Applications* (SLUS707)
11. *bq27x10EVM Single-Cell Battery Fuel Gauge Evaluation Module* (SLUU259)
12. *System-Side Impedance Track™ Fuel Gauge* (SLUS880)
13. *bq2750xEVM System-Side Single-Cell Impedance Track™ Technology EVM* (SLUS287)
14. *Updating Firmware With the bq2750x and EVM* (SLUA453)
15. *Single-Cell Li-Ion Battery Fuel Gauge for Battery Pack Integration* (SLUS861)
16. *bq27541EVM Single-Cell Impedance Track™ Technology Evaluation Module* (SLUU273)
17. *MSP430x2xx Family User's Guide* (SLAU144)
18. *MSP430x20x1, MSP430x20x2, MSP430x20x3 Mixed Signal Microcontroller* (SLAS491)
19. *MSP430x23x0 Mixed Signal Microcontroller* (SLAS518)

# Appendix A Source Code Files Required for Each Example

## A.1 MSP430 + bq26100

### A.1.1 IAR Systems Embedded Workbench

| SOURCE FILES | HEADER FILES |
|---|---|
| MSP430_bq26100.c | |
| MSP430_SDQLIB.c | MSP430_SDQLIB.c |
| SDQ_writeByte_IAR.s43 | SDQ_writeByte.h |
| SHA1_HMAC.c | SHA1_HMAC.h |
| vlo_rand_IAR.s43 | vlo_rand.h |

### A.1.2 TI Code Composer Studio

| SOURCE FILES | HEADER FILES |
|---|---|
| MSP430_bq26100.c | |
| MSP430_SDQLIB.c | MSP430_SDQLIB.h |
| SDQ_writeByte_CCS.asm | SDQ_writeByte.h |
| SHA1_HMAC.c | SHA1_HMAC.h |
| vlo_rand_CCS.asm | vlo_rand.h |

## A.2 MSP430 + bq26150

### A.2.1 IAR Systems Embedded Workbench

| SOURCE FILES | HEADER FILES |
|---|---|
| MSP430_bq26150.c | bq26150.h |
| HDQ.c | HDQ.h |
| vlo_rand_IAR.s43 | vlo_rand.h |

### A.2.2 TI Code Composer Studio

| SOURCE FILES | HEADER FILES |
|---|---|
| MSP430_bq26150.c | bq26150.h |
| HDQ.c | HDQ.h |
| vlo_rand_CCS.asm | vlo_rand.h |

## A.3 MPS430 + bq26500

### A.3.1 IAR Systems Embedded Workbench

| SOURCE FILES | HEADER FILES |
|---|---|
| MSP430_bq26500.c | |
| HDQ.c | HDQ.h |

### A.3.2 TI Code Composer Studio

| SOURCE FILES | HEADER FILES |
|---|---|
| MSP430_bq26500.c | |
| HDQ.c | HDQ.h |

## A.4 MSP430 + bq27010

### A.4.1 IAR Systems Embedded Workbench

| SOURCE FILES | HEADER FILES |
|---|---|
| MSP430_bq27010.c | bq27x10.h |
| HDQ.c | HDQ.h |

### A.4.2 TI Code Composer Studio

| SOURCE FILES | HEADER FILES |
|---|---|
| MSP430_bq27010.c | bq27x10.h |
| HDQ.c | HDQ.h |

## A.5 MSP430 + bq27210

### A.5.1 IAR Systems Embedded Workbench

| SOURCE FILES | HEADER FILES |
|---|---|
| MSP430_bq27210.c | bq27x10.h |
| MSP430_SWI2C_Master.c | MSP430_SWI2C_Master.h |
| USI_I2CMaster_IAR.s43 | USI_I2CMaster.h |

### A.5.2 TI Code Composer Studio

| SOURCE FILES | HEADER FILES |
|---|---|
| MSP430_bq27210.c | bq27x10.h |
| MSP430_SWI2C_Master.c | MSP430_SWI2C_Master.h |
| USI_I2CMaster_CCS.asm | USI_I2CMaster.h |

## A.6 MSP430 + bq27500

### A.6.1 IAR Systems Embedded Workbench

| SOURCE FILES | HEADER FILES |
|---|---|
| MSP430_bq27500.c | bq27500.h |
| MSP430_SWI2C_Master.c | MSP430_SWI2C_Master.h |
| USI_I2CMaster_IAR.s43 | USI_I2CMaster.h |

### A.6.2 TI Code Composer Studio

| SOURCE FILES | HEADER FILES |
|---|---|
| MSP430_bq27500.c | bq27500.h |
| MSP430_SWI2C_Master.c | MSP430_SWI2C_Master.h |
| USI_I2CMaster_CCS.asm | USI_I2CMaster.h |

## A.7 MSP430 + bq27541 (HDQ)

### A.7.1 IAR Systems Embedded Workbench

| SOURCE FILES | HEADER FILES |
|---|---|
| MSP430_bq27541_HDQ.c | bq27541.h |
| HDQ.c | HDQ.h |
| SHA1_HMAC.c | SHA1_HMAC.h |
| vlo_rand_IAR.s43 | vlo_rand.h |

### A.7.2 TI Code Composer Studio

| SOURCE FILES | HEADER FILES |
|---|---|
| MSP430_bq27541_HDQ.c | bq27541.h |
| HDQ.c | HDQ.h |
| SHA1_HMAC.c | SHA1_HMAC.h |
| vlo_rand_CCS.asm | vlo_rand.h |

## A.8 MSP430 + bq27541 ($I^2C$)

### A.8.1 IAR Systems Embedded Workbench

| SOURCE FILES | HEADER FILES |
|---|---|
| MSP430_bq27541_I2C.c | bq27541.h |
| MSP430_SWI2C_Master.c | MSP430_SWI2C_Master.h |
| TI_USCI_I2C_master.c | TI_USCI_I2C_master.h |
| SHA1_HMAC.c | SHA1_HMAC.h |
| vlo_rand_IAR.s43 | vlo_rand.h |

### A.8.2 TI Code Composer Studio

| SOURCE FILES | HEADER FILES |
|---|---|
| MSP430_bq27541_I2C.c | bq27541.h |
| MSP430_SWI2C_Master.c | MSP430_SWI2C_Master.h |
| TI_USCI_I2C_master.c | TI_USCI_I2C_master.h |
| SHA1_HMAC.c | SHA1_HMAC.h |
| vlo_rand_CCS.asm | vlo_rand.h |

## IMPORTANT NOTICE



Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

| **Products** | | **Applications** | |
|---|---|---|---|
| Amplifiers | amplifier.ti.com | Audio | www.ti.com/audio |
| Data Converters | dataconverter.ti.com | Automotive | www.ti.com/automotive |
| DLP® Products | www.dlp.com | Communications and Telecom | www.ti.com/communications |
| DSP | dsp.ti.com | Computers and Peripherals | www.ti.com/computers |
| Clocks and Timers | www.ti.com/clocks | Consumer Electronics | www.ti.com/consumer-apps |
| Interface | interface.ti.com | Energy | www.ti.com/energy |
| Logic | logic.ti.com | Industrial | www.ti.com/industrial |
| Power Mgmt | power.ti.com | Medical | www.ti.com/medical |
| Microcontrollers | microcontroller.ti.com | Security | www.ti.com/security |
| RFID | www.ti-rfid.com | Space, Avionics & Defense | www.ti.com/space-avionics-defense |
| RF/IF and ZigBee® Solutions | www.ti.com/lprf | Video and Imaging | www.ti.com/video |
| | | Wireless | www.ti.com/wireless-apps |

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