User's Guide
SLUUAO1A–April 2014–Revised February 2015

Gauge Development Kit

The Gauge Development Kit (GDK) is a complete evaluation system for any single-cell fuel gauge developed by Texas Instruments Incorporated. The GDK is a single, printed circuit board (PCB) equipped with a programmable load, programmable charger, an optional onboard fuel gauge, and an integrated EV2400 for PC interaction via Battery Management Studio (bqSTUDIO). A USB PC cable, 6-V 2.5-A DC jack (or equivalent power supply connection), and the latest version of bqSTUDIO is needed when using the GDK. The GDK is by default configured for an external single-cell fuel gauge EVM to be connected, but with minor adjustments to the GDK, the optional onboard fuel gauge can be used instead of an external EVM. Once the GDK is connected to a PC via USB cable, bqSTUDIO enables the user to do the following:

- Read the connected fuel gauge data registers
- Configure the connected fuel gauge
- Discharge the connected battery
- Charge the connected battery
- Log cycle data for evaluation
- Automate Learning Cycle(s)
- Evaluate the overall functionality of the connected fuel gauge solution under different charge and discharge conditions

Contents

List of Figures

List of Tables

1 Features

The GDK includes the following features:

- Complete evaluation system for connected single-cell fuel gauge
- Optional onboard fuel gauge (bq27421-G1)
- Integrated EV2400 for PC interface and downloadable software (bqSTUDIO) for easy evaluation and test setup
- bqSTUDIO allows various discharge and charge scenarios along with simultaneous data logging
- bqSTUDIO allows for automated learning cycles
- Programmable load allows currents up to 2.0 A
- Programmable load with constant current, constant power, and pulsed load modes
- Programmable fast charge current up to approximately 2.0 A

2 Kit Contents

- BQ27GDK000EVM Board (PWR568)

This device is used for the evaluation of any single-cell fuel gauge EVM supported by the Battery Management Studio (bqSTUDIO) tool.

3 Ordering Information

EVM Part Number	Firmware Version[1]	Configuration	Onboard Fuel Gauge
BQ27GDK000EVM	v1.04 (0x0104)	External EVM	bq27421-G1A

[1] Using the *FWVersion()* Standard Command returns 0x0104. See the Standard Commands section for more information.

4 Block Diagram

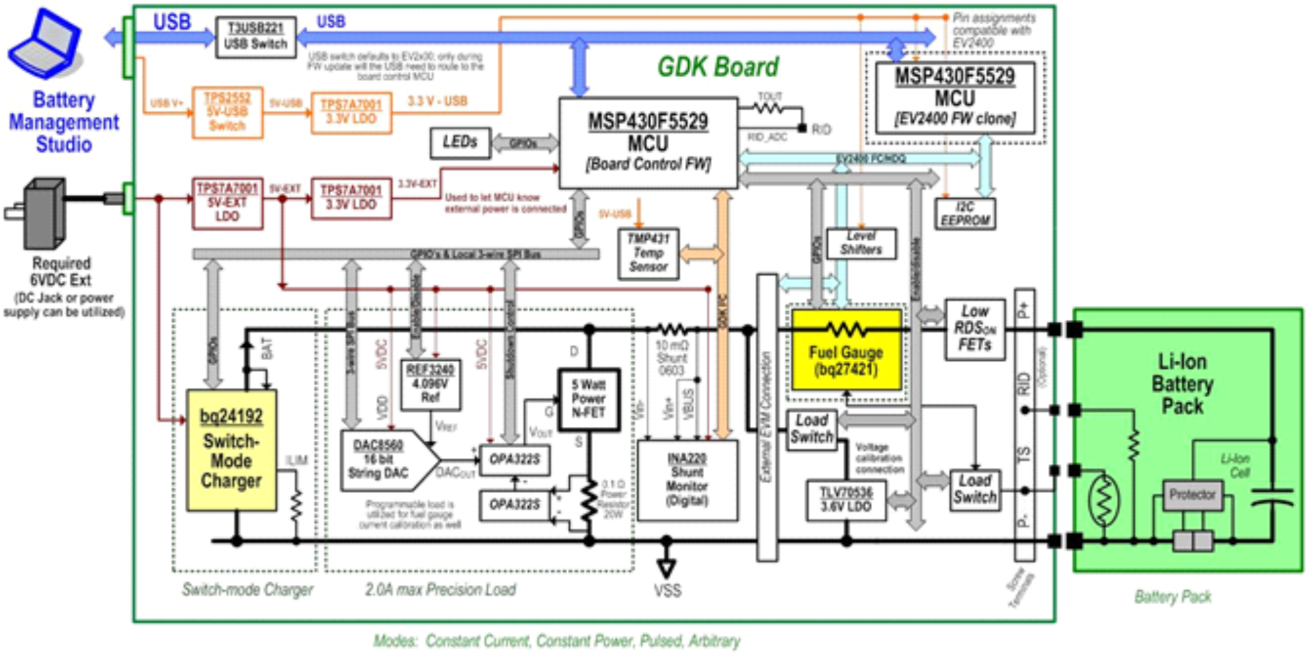


Figure 1. GDK Block Diagram

5 GDK Hardware and Software Setup

5.1 Board Connections

Contacts on the board provide the following connections:

- External power (J3): common, 2,1-mm, 6-V, 2.5-A DC jack
  - Recommended DC jack: EMSA060300K-P5P-SZ; Digi-Key: T1107-P5P-ND
  - Optionally a power supply (PS) can be connect to TP18 (6-V plug) and TP39 (PGND)
- USB power (J2)
- External temperature sensing (J1): DXP, DXN
- External EVM system load and charger (J4 and J8): EXT Load+ and EXT Load–
- External EVM serial communication (J13): SDA, SCL, and V_{SS}
- External EVM HDQ communication (J10): HDQ and V_{SS}
- Direct connection to battery pack (J9): Pack+, Pack–, BIN, RID
  Connection only used with optional onboard configuration. See the GDK Board Configuration section for more information.

5.2 Terminal Descriptions

Table 1. GDK Pin Descriptions

Terminal Name	Description
BIN	Battery insertion detection input
RID	Resistor identification input
DXP	External temperature sensing transistor positive terminal
DXN	External temperature sensing transistor negative terminal
EXT Load+	External EVM Load+ or Charger+
EXT Load–	External EVM Load– or Charger–
SDA	I^2C communication data line
SCL	I^2C communication clock line
V_{SS}	Signal return for communication line, shared with charger and ground
HDQ	HDQ communication data line
Pack+	Pack positive terminal
Pack–	Pack negative terminal

5.3 Test Point Descriptions

Table 2. GDK Test Point Descriptions

Test Point Designator	Test Point Description
TP4, TP13, TP17, TP39	PGND
TP18	6 V Plug (J3.1)
TP34	External Load+ or Charger+ (J4)
TP33	External Load– or Charge– (J8)
TP31	Pack+ (J9.4)
TP32	Pack– (J9.1)

5.4 Switch Descriptions

Table 3. GDK Switch Descriptions

Switch Name	Switch Description
SW1	EV2400 MSP reset
SW2	Firmware (FW) update
SW3	GDK MSP reset

5.5 Header Descriptions

Table 4. GDK Header Descriptions

Header Name	Header Description
J1	External temperature sensor connector
J2	USB connector
J3	DC jack connector
J4	External Load+ connector
J8	External Load– connector
J9	Battery pack connector
J10	External HDQ connector
J13	External I^2C connector

5.6 Jumper Descriptions

Table 5. GDK Jumper Descriptions

Jumper Name	Jumper Description
JP1	SMBPU1 pullup disable
JP2	SWPUE pullup disable
JP3	EV2400 SCL bypass
JP4	EV2400 SDA bypass
JP5	EV2400 HDQ bypass
JP6	Bypass; not connected
JP7	BIN pullup or pulldown selection
JP8	GPOUT pullup
JP9	External or onboard gauge SCL selection
JP10	External or onboard gauge SDA selection

5.7 Power

The GDK has two power rails: one rail from USB and one rail from an external DC jack or power supply. Both USB and external power are required to be connected for proper operation. A maximum of 6 V should be placed on the external rail to prevent damage to the components on the board. Because the GDK can provide a charge current up to 2.0 A, it is recommended to have an adapter that can provide at least 2.0 A of current.

5.8 Fuel Gauge

The GDK comes configured for the connection of an external single-cell fuel gauge EVM. Alternatively, the GDK can be configured to work with the onboard fuel gauge. For more information concerning the different configurations of the GDK, please see the GDK Board Configuration section.

5.9 Programmable Load

The GDK includes a 2.0-A programmable load that allows the user, through bqSTUDIO, to customize the type of load that should be placed on the battery under test. The GDK allows three types of discharge modes:

- Constant current
- Constant power
- Pulsed current

bqSTUDIO allows the user to customize various aspects of the load and when the load will terminate. For more information on the discharging operation, please see the GDK Operation section.

5.10 Programmable Charger

The GDK includes a bq24192 circuit model to serve as a programmable charger that allows the user, through bqSTUDIO, to customize the fast charge current, charge voltage, and charge termination conditions. For more information on the charging operation, please see the GDK Operation section.

5.11 External Temperature Measurement

The GDK, by default, reports temperature from the internal temperature of the onboard temperature sensor. The GDK uses the TMP431 temperature sensor from Texas Instruments Incorporated. The GDK also has the ability to measure external temperature by connecting a PNP-type or NPN-type diode-connected transistor to the external temperature sensor connector (J1) on the GDK board. The recommended small signal transistors to use with the GDK are the 2N3904 (NPN) and 2N3906 (PNP). For more details about external temperature measurement, please see the Remote Sensing section in the TMP431 data sheet (SBOS441).

5.12 PC Interface and Communication

The GDK includes an EV2400 on the board to provide the communication link between the fuel gauge and the PC software, bqSTUDIO. The GDK is connected to the PC via the USB interface on the PC and the USB port located on the top left of the GDK board (J2). bqSTUDIO controls the charge, discharge, and fuel gauge communication operations of the GDK. For more information on the operation of bqSTUDIO, please see the GDK Operation section.

5.13 System Requirements

The Battery Management Studio (bqSTUDIO) software requires Windows XP or later. Using earlier versions of the Windows operating system can have issues with the USB driver support.

5.14 Software Installation

The latest version of bqSTUDIO can be found at http://www.ti.com/tool/bq27gdk000evm.
Follow these steps to install bqSTUDIO:

1. Ensure that the GDK board is not connected to the PC via the USB cable before starting the procedure.
2. Open the archive containing the installation package, and copy its contents into a temporary directory.
3. Open the software file that was downloaded from the TI website.
4. Follow the instructions on the screen to continue the software installation.
5. Before starting bqSTUDIO, follow the recommended GDK power-up sequence in the Hardware Connection section.
6. The EV2400 should automatically complete installation, so drivers should not be needed.

6 GDK Operation

6.1 Hardware Connection

Figure 2 shows how to connect the GDK to the PC, external power, and an external EVM. The recommended power-up and power removal sequences for the GDK can be found in Power-up Sequence and Power Removal Sequence, respectively.

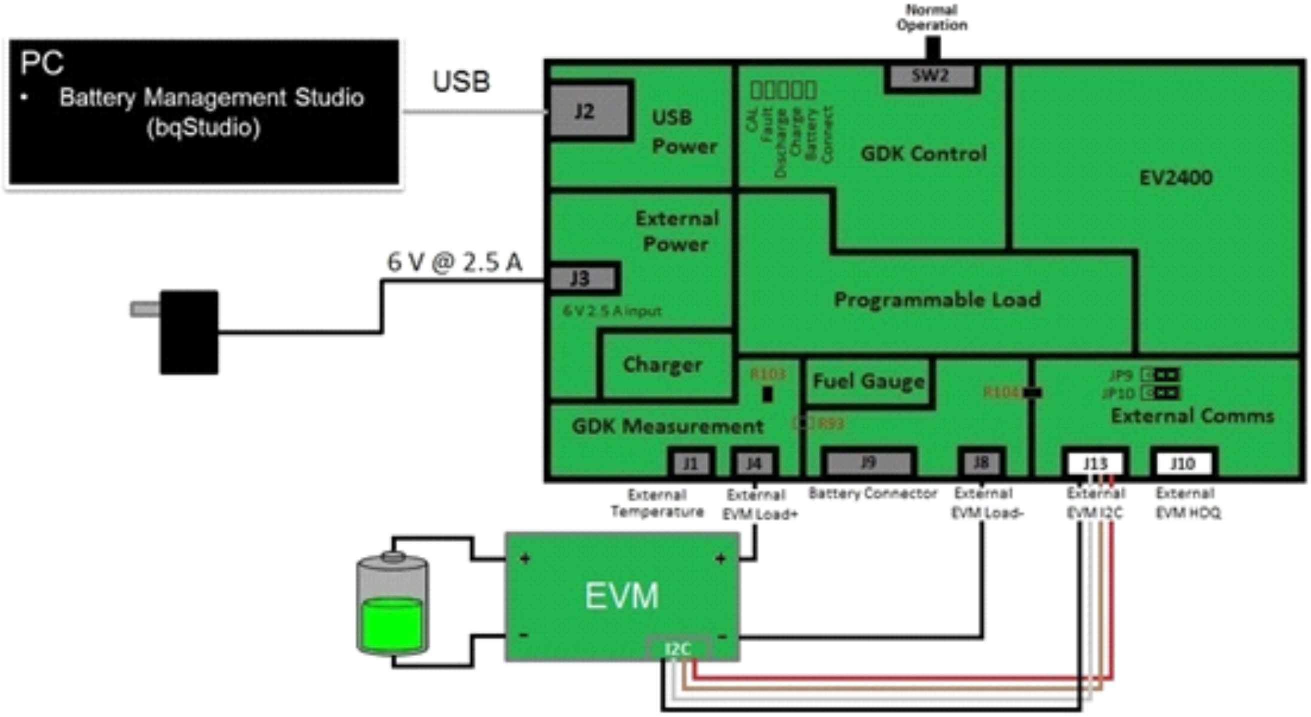


Figure 2. GDK Hardware Connections

6.1.1 Power-Up Sequence

The recommended power-up sequence is below:

1. Connect external power (either DC jack of the power supply) to J3 or the corresponding test points: TP18 (6V Plug) and TP39 (PGND).
2. Connect USB power to J2.

NOTE: If USB power is connected before the external power is connected, the fault LED will be lit upon initialization of the GDK due to lack of external power. To remove the fault condition, use bqSTUDIO or SW3 to reset the GDK control after external power has been connected.

3. Connect external EVM to External Load/Charger+ (J4) and External Load/Charger– (J8) and external I^{2}C (J13). The battery should be connected to the external EVM and not the GDK.

NOTE: If using the onboard fuel gauge, connect the battery to Pack+ (J9.4) and Pack– (J9.1) terminals of J9. Please see GDK Board Configuration section for further details.

4. Connect the external EVM I^{2}C bus to the GDK external EVM I^{2}C bus connector (J13). If the external EVM has jumpers for I^{2}C pullup resistors, it is recommended to remove those jumpers and use the I^{2}C pullup resistors on the GDK.

6.1.2 Power Removal Sequence

Recommended power removal sequence is below:

1. Remove external power.
2. Remove external EVM.

NOTE: If using onboard fuel gauge, remove the battery from J9.

3. Remove USB power.

NOTE: If only communication to the fuel gauge is desired, external power does not need to be connected, but the battery should be connected to power the fuel gauge. With external power not connected, the GDK will display a fault in bqSTUDIO and the fault LED (D14) will be lit.

6.2 Fuel Gauge Communication

6.2.1 External EVM

The GDK can connect to any single-cell fuel gauge EVM that is supported by bqSTUDIO. When using an external EVM with the GDK, communication to the fuel gauge will be established when the external EVM I^{2}C bus is properly connected to the GDK external I^{2}C bus (J13). Also, the external EVM should have the battery connected to the EVM so that the fuel gauge on the EVM has power. Figure 2 shows the connection of an external EVM to the GDK. For more information concerning the use of an external EVM with the GDK, please see the GDK Board Configuration section.

6.2.2 Optional Onboard Fuel Gauge

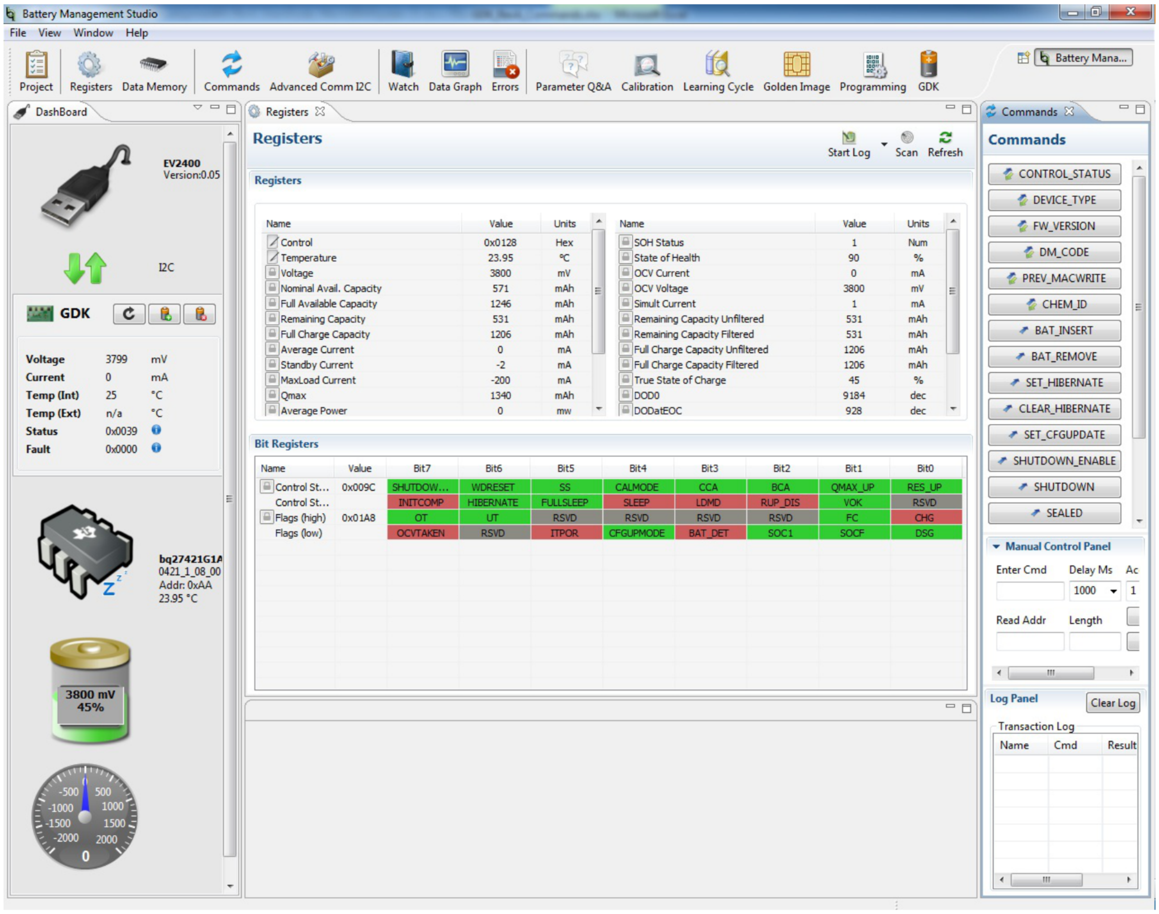
When using the optional onboard fuel gauge, communication to the fuel gauge will be established once the battery is connected to the Pack+ (J9.4) and Pack– (J9.1) terminals of J9 and the *CONNECT_BATTERY Control()* subcommand has been sent. For more information on the use of the onboard fuel gauge and on battery connection, see the GDK Board Configuration and Battery Connection and Disconnection sections, respectively.

6.3 Starting Battery Management Studio (bqSTUDIO)

After the GDK has been connected and powered properly (see Figure 2), bqSTUDIO can be launched. Launch bqSTUDIO from the Start > Programs > Texas Instruments > bqSTUDIO menu sequence. Once the launch is complete, the bqSTUDIO default GDK perspective (Figure 3) appears. The default GDK perspective consists of a dashboard panel on the left side of the window, a plug-ins panel at the top, and active plug-ins that can be hidden, docked in various positions, or allowed to float as separate windows. If bqSTUDIO launches for the first time, only a welcome window will be displayed. Simply close the welcome window tab to get to the default GDK perspective shown in Figure 3.

A bqSTUDIO perspective consists of the plug-ins (that is, Registers, Data Memory, GDK, and so on) a user has opened while using the program. Battery Management Studio will remember the previous perspective upon subsequent launches after the initial launch. A user also has the option to save and load a particular perspective.

If bqSTUDIO recognizes that a GDK is connected, the dashboard will reflect a GDK section and the GDK plug-in will be seen in the plug-ins panel as shown in Figure 3.



Voltage	3799	mV
Current	0	mA
Temp (Int)	25	°C
Temp (Ext)	n/a	°C
Status	0x0039	
Fault	0x0000	

Registers

Name	Value	Units
Control	0x0128	Hex
Temperature	23.95	°C
Voltage	3800	mV
Nominal Avail. Capacity	571	mAh
Full Available Capacity	1246	mAh
Remaining Capacity	531	mAh
Full Charge Capacity	1206	mAh
Average Current	0	mA
Standby Current	-2	mA
MaxLoad Current	-200	mA
Qmax	1340	mAh
Average Power	0	mw

Name	Value	Units
SOH Status	1	Num
State of Health	90	%
OCV Current	0	mA
OCV Voltage	3800	mV
Simult Current	1	mA
Remaining Capacity Unfiltered	531	mAh
Remaining Capacity Filtered	531	mAh
Full Charge Capacity Unfiltered	1206	mAh
Full Charge Capacity Filtered	1206	mAh
True State of Charge	45	%
DOD0	9184	dec
DODatEOC	928	dec

Bit Registers

Name	Value	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Control St...	0x009C	SHUTDOW...	WDRESET	SS	CALMODE	CCA	BCA	QMAX_UP	RES_UP
Control St...		INITCOMP	HIBERNATE	FULLSLEEP	SLEEP	LDMD	RUP_DIS	VOK	RSVD
Flags (high)	0x01A8	OT	UT	RSVD	RSVD	RSVD	RSVD	FC	CHG
Flags (low)		OCVTAKEN	RSVD	ITPOR	CFGUPMODE	BAT_DET	SOC1	SOCF	DSG

Enter Cmd: Delay Ms: 1000 Ac: 1

Read Addr: Length:

Name	Cmd	Result

Figure 3. bqSTUDIO Default GDK Perspective

6.4 Battery Connection and Disconnection

After the GDK has been connected to the PC and powered properly (see Hardware Connection section) and bqSTUDIO has been launched, bqSTUDIO should recognize the connected GDK and send the *CONNECT_BATTERY* (0x0001) *Control()* subcommand. Therefore, upon startup of bqSTUDIO, the connected fuel gauge will be recognized, the GDK section in the dashboard will be shown, the GDK plug-in icon will be shown in the plug-ins toolbar, and the Battery Connect LED should be lit on the GDK board.

The GDK section in the dashboard contains three buttons that send commands to the GDK. The buttons are described below:

- Reset—This button issues a *RESET Control()* subcommand to the GDK.
- Connect Battery—This button issues a *CONNECT_BATTERY Control()* subcommand to the GDK.
- Disconnect Battery—This button issues a *DISCONNECT_BATTERY Control()* subcommand to the GDK.

NOTE: When using either the default GDK configuration or the optional onboard configuration, the *CONNECT_BATTERY Control()* subcommand must be sent before the GDK will allow charging or discharging.

Once the battery connection process is complete, the GDK plug-in will allow for quick manual control over the programmable load and charger or allow a more detailed script to be created or loaded that can include any number of charge, discharge, and relaxation operations. More information on the charging and discharging process can be found in the Charging and Discharging sections below.

6.5 Charging

The GDK uses the bq24192 charger IC from Texas Instruments Incorporated to charge the battery connected to the external EVM. The GDK allows the user to specify the charging voltage, fast charge current, and charge termination method. Charge mode can be enabled through the manual control or scriptable control method in bqSTUDIO. More information about manual and scriptable control can be found in the Manual Control and Scriptable Control sections, respectively. For detailed information on the charger used by the GDK, please see the bq24192 data sheet (SLUSAW5).

For charge mode, the maximum allowable charge current is approximately 2000 mA and the minimum allowable fast charge current is approximately 105 mA. The maximum allowable charge voltage is 4400 mV and the minimum allowable charge voltage is 3504 mV.

NOTE: Even though the bq24192 allows up to a 4.5-A charge current, the GDK limits this maximum to approximately 2.0 A due to the configuration of the board.

The GDK allows multiple types of charge termination, each termination method is described in Table 6.

Table 6. Charge Termination Methods

Termination Method (Abbreviation)	Description	Termination Value Required	Termination Value Unit	Manual or Scriptable Control
Taper Current (TC)	Terminates the charge when the connected fuel gauge reports an *AverageCurrent()* greater than or equal to the specified value	Yes	mA	Manual or Scriptable
State of Charge (SC)	Terminates the charge when the connected fuel gauge reports a *StateOfCharge()* greater than or equal to the specified value	Yes	%	Manual or Scriptable
Remaining Capacity (RC)	Terminates the charge when the connected fuel gauge reports a *RemainingCapacity()* greater than or equal to the specified value	Yes	mAh	Manual or Scriptable
Passed Charge (PC)	Terminates the charge after the specified amount of passed charge has accumulated	Yes	mAh	Scriptable
Time (SS)	Terminates the charge after the specified number of seconds	Yes	s	Manual or Scriptable
Flags() [FC] bit (FC)	Terminates the charge when the connected fuel gauge sets the *[FC]* bit in the *Flags()* register	No	NA	Manual or Scriptable

6.6 Discharging

The GDK allows three different types of discharge modes, each described below. The GDK allows the user to specify the discharge mode and discharge termination method. Each mode can be enabled through the manual control or scriptable control method in bqSTUDIO. More information about manual and scriptable control can be found in the Manual Control and Scriptable Control sections, respectively.

- Constant Current—This discharge mode will discharge the battery at the specified load, in mA, until the specified termination condition is met.
- Constant Power—This discharge mode will discharge the battery at the specified power, in mW, until the specified termination condition is met.
- Pulsed Load—This discharge mode will discharge the battery at the high and low loads, in mA, with the specified pulse period and duty cycle until the specified termination condition is met.

For all discharge modes, the maximum allowable load is 2000 mA and the minimum allowable load is 5 mA with a resolution of 1 mA. For the pulsed-load discharge mode, the minimum period is 50 milliseconds (ms) and the maximum period is 45 seconds with a 1-ms resolution, the minimum duty cycle is 10% and the maximum duty cycle is 90% with a 1% resolution.

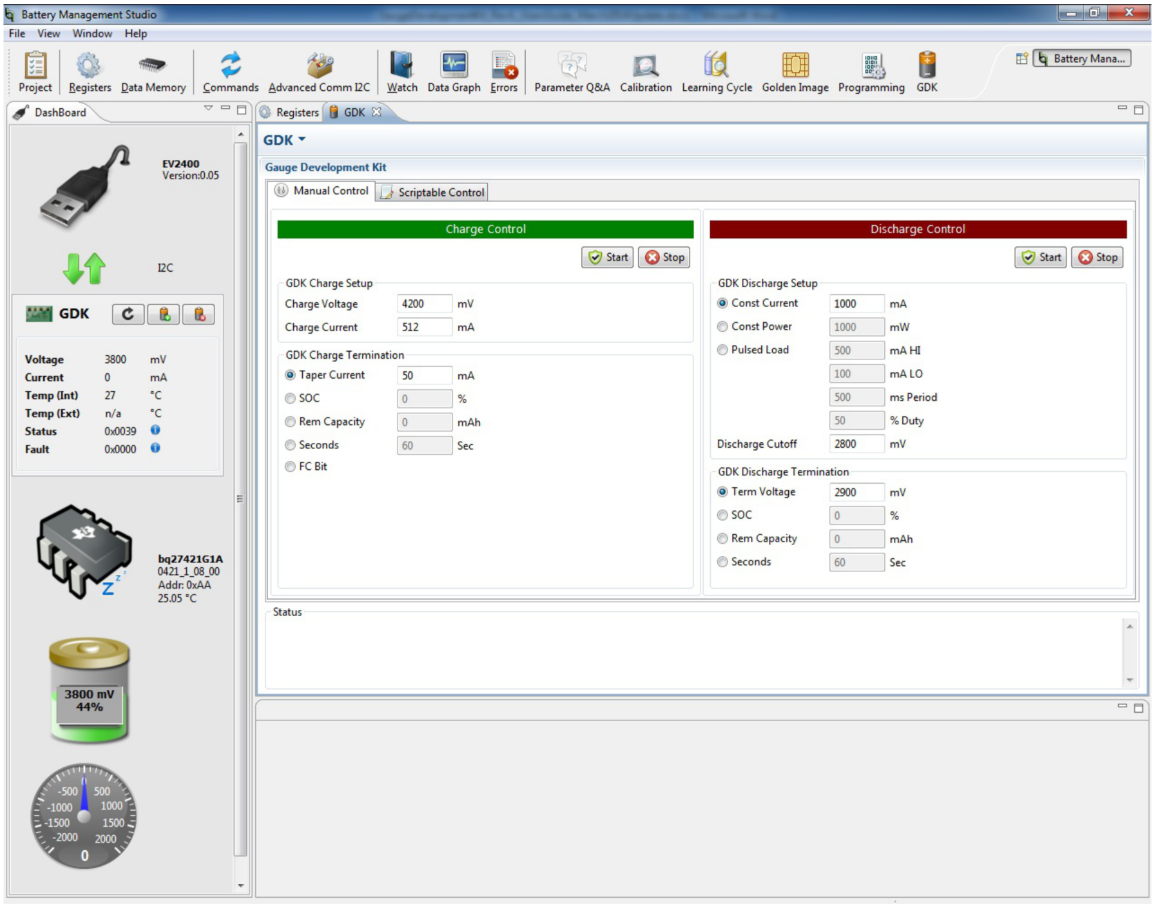
The GDK allows multiple types of discharge termination, each termination method is described in Table 7.

Table 7. Discharge Termination Methods

Termination Method (Abbreviation)	Description	Termination Value Required	Termination Value Unit	Manual or Scriptable Control
Terminate Voltage (TV)	Terminates the discharge when the connected fuel gauge reports a *Voltage()* less than or equal to the specified value	Yes	mV	Manual or Scriptable
State of Charge (SC)	Terminates the discharge when the connected fuel gauge reports a *StateOfCharge()* less than or equal to the specified value	Yes	%	Manual or Scriptable
Remaining Capacity (RC)	Terminates the discharge when the connected fuel gauge reports a *RemainingCapacity()* less than or equal to the specified value	Yes	mAh	Manual or Scriptable
Time (SS)	Terminates the discharge after the specified number of seconds	Yes	second	Manual or Scriptable
Passed Charge (PC)	Terminates the discharge after the specified amount of passed charge has accumulated	Yes	mAh	Scriptable
Flags() [SOC1] bit (S1)	Terminates the discharge when the connected fuel gauge sets the *[SOC1]* bit in the *Flags()* register	No	NA	Scriptable
Flags() [SOCF] bit (SF)	Terminates the discharge when the connected fuel gauge sets the *[SOCF]* bit in the *Flags()* register	No	NA	Scriptable

6.7 Manual Control

The Manual Control tab in the GDK plug-in allows a quick charge or discharge profile to be setup and executed. The Manual Control tab contains two panels, the Charge Control panel and the Discharge Control panel. Figure 4 shows the Manual Control Panel in the GDK plug-in.



EV2400
Version:0.05

I2C

GDK

Voltage	3800	mV
Current	0	mA
Temp (Int)	27	°C
Temp (Ext)	n/a	°C
Status	0x0039	
Fault	0x0000	

bq27421G1A
0421_1_08_00
Addr: 0xAA
25.05 °C

3800 mV
44%

GDK

Gauge Development Kit

Charge Control

GDK Charge Setup

Charge Voltage	4200	mV
Charge Current	512	mA

GDK Charge Termination

● Taper Current	50	mA
○ SOC	0	%
○ Rem Capacity	0	mAh
○ Seconds	60	Sec
○ FC Bit		

Discharge Control

GDK Discharge Setup

● Const Current	1000	mA
○ Const Power	1000	mW
○ Pulsed Load	500	mA HI
	100	mA LO
	500	ms Period
	50	% Duty
Discharge Cutoff	2800	mV

GDK Discharge Termination

● Term Voltage	2900	mV
○ SOC	0	%
○ Rem Capacity	0	mAh
○ Seconds	60	Sec

Status

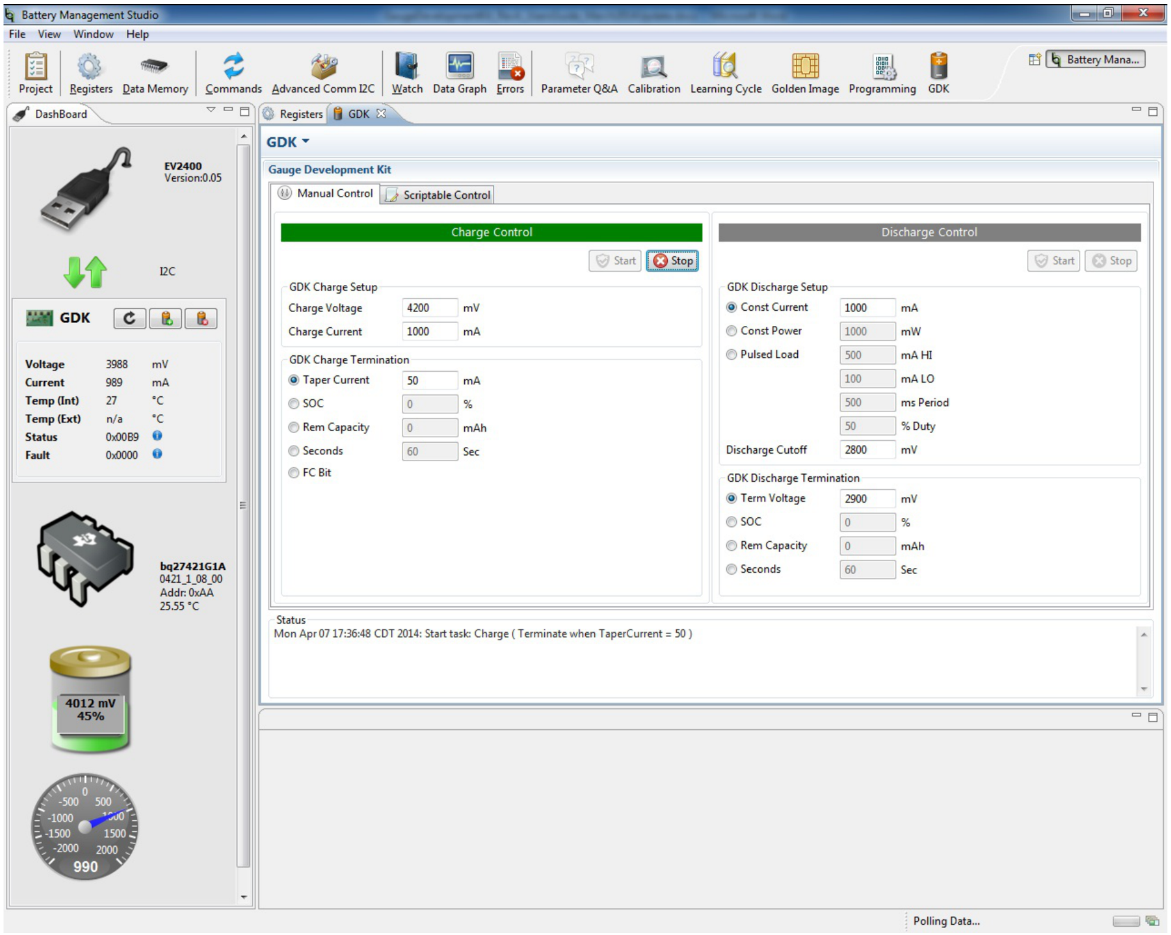
Figure 4. GDK Plug-in—Manual Control Tab

6.7.1 Manual Control–Charge Profile Panel

To begin a charge, the charge profile parameters should be filled out in the Charge Control panel. The user should enter the charge voltage and charge current for the charge profile in the GDK Charge Setup section of the Charge Control panel and also choose a charge termination method and specify a value for the charge termination, if applicable, in the GDK Charge Termination section of the Charge Control panel.

After the charge voltage, charge current, and charge termination method and value have been specified, the Start button in the Charge Control panel can be clicked to begin the charge. Figure 5 shows the Manual Control Tab in the GDK plug-in while a charge is in progress. The charge will terminate once the proper termination condition has been met or the user clicks the Stop button in the Charge Control panel.

NOTE: The user is responsible for inputting appropriate values for the charge profile based on the information provided by the battery manufacturer.



Voltage	3988	mV
Current	989	mA
Temp (Int)	27	°C
Temp (Ext)	n/a	°C
Status	0x00B9	
Fault	0x0000	

GDK Charge Setup

Charge Voltage	4200	mV
Charge Current	1000	mA

GDK Charge Termination

● Taper Current	50	mA
○ SOC	0	%
○ Rem Capacity	0	mAh
○ Seconds	60	Sec
○ FC Bit		

GDK Discharge Setup

● Const Current	1000	mA
○ Const Power	1000	mW
○ Pulsed Load	500	mA HI
	100	mA LO
	500	ms Period
	50	% Duty
Discharge Cutoff	2800	mV

GDK Discharge Termination

● Term Voltage	2900	mV
○ SOC	0	%
○ Rem Capacity	0	mAh
○ Seconds	60	Sec

Status

```
Mon Apr 07 17:36:48 CDT 2014: Start task: Charge ( Terminate when TaperCurrent = 50 )
```

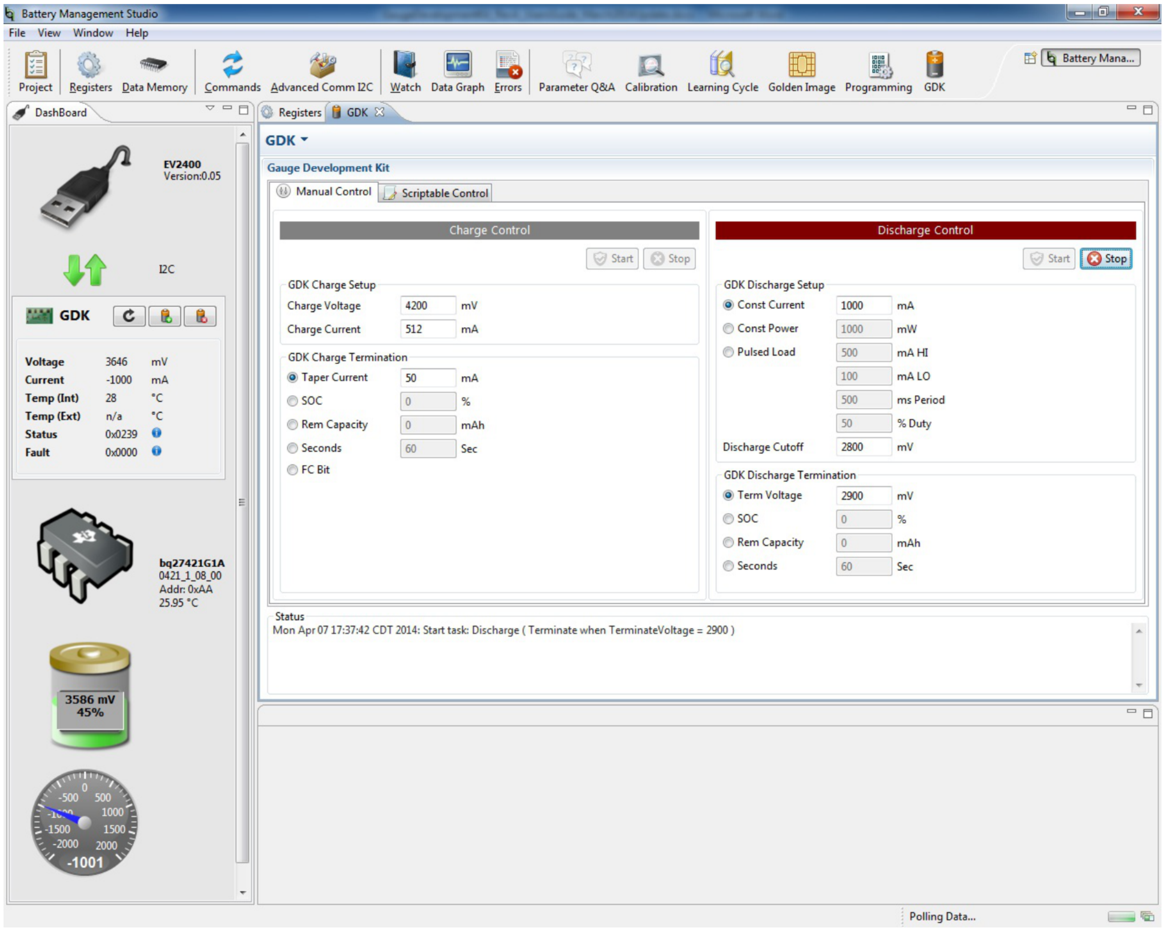
Figure 5. Manual Control Tab—Charge Control Panel While Charging

6.7.2 Manual Control–Discharge Profile Panel

To begin a discharge, the discharge profile parameters should be filled out in the Discharge Control panel. The user should select the discharge mode and specify the necessary values for the selected mode in the GDK Discharge Setup section of the Discharge Control panel. The user should also choose a discharge termination method and specify a value for the discharge termination, if applicable, in the GDK Discharge Termination section of the Discharge Control panel.

After the discharge mode has been selected, the necessary values for the selected discharge mode have been specified and the discharge termination method and value have been specified, the Start button in the Discharge Control panel can be clicked to begin the discharge. Figure 6 shows the Manual Control Tab in the GDK plug-in while a discharge is in progress. The discharge will terminate once the proper termination condition has been met or the user clicks the Stop button in the Discharge Control panel.

NOTE: The user is responsible for inputting appropriate values for the discharge profile based on the information provided by the battery manufacturer.



EV2400
Version:0.05

Voltage	3646	mV
Current	-1000	mA
Temp (Int)	28	°C
Temp (Ext)	n/a	°C
Status	0x0239	
Fault	0x0000	

bq27421G1A
0421_1_08_00
Addr: 0xAA
25.95 °C

3586 mV
45%

-1001

GDK Charge Setup

Charge Voltage	4200	mV
Charge Current	512	mA

GDK Charge Termination

● Taper Current	50	mA
○ SOC	0	%
○ Rem Capacity	0	mAh
○ Seconds	60	Sec
○ FC Bit		

GDK Discharge Setup

● Const Current	1000	mA
○ Const Power	1000	mW
○ Pulsed Load	500	mA HI
	100	mA LO
	500	ms Period
	50	% Duty
Discharge Cutoff	2800	mV

GDK Discharge Termination

● Term Voltage	2900	mV
○ SOC	0	%
○ Rem Capacity	0	mAh
○ Seconds	60	Sec

Status
Mon Apr 07 17:37:42 CDT 2014: Start task: Discharge (Terminate when TerminateVoltage = 2900)

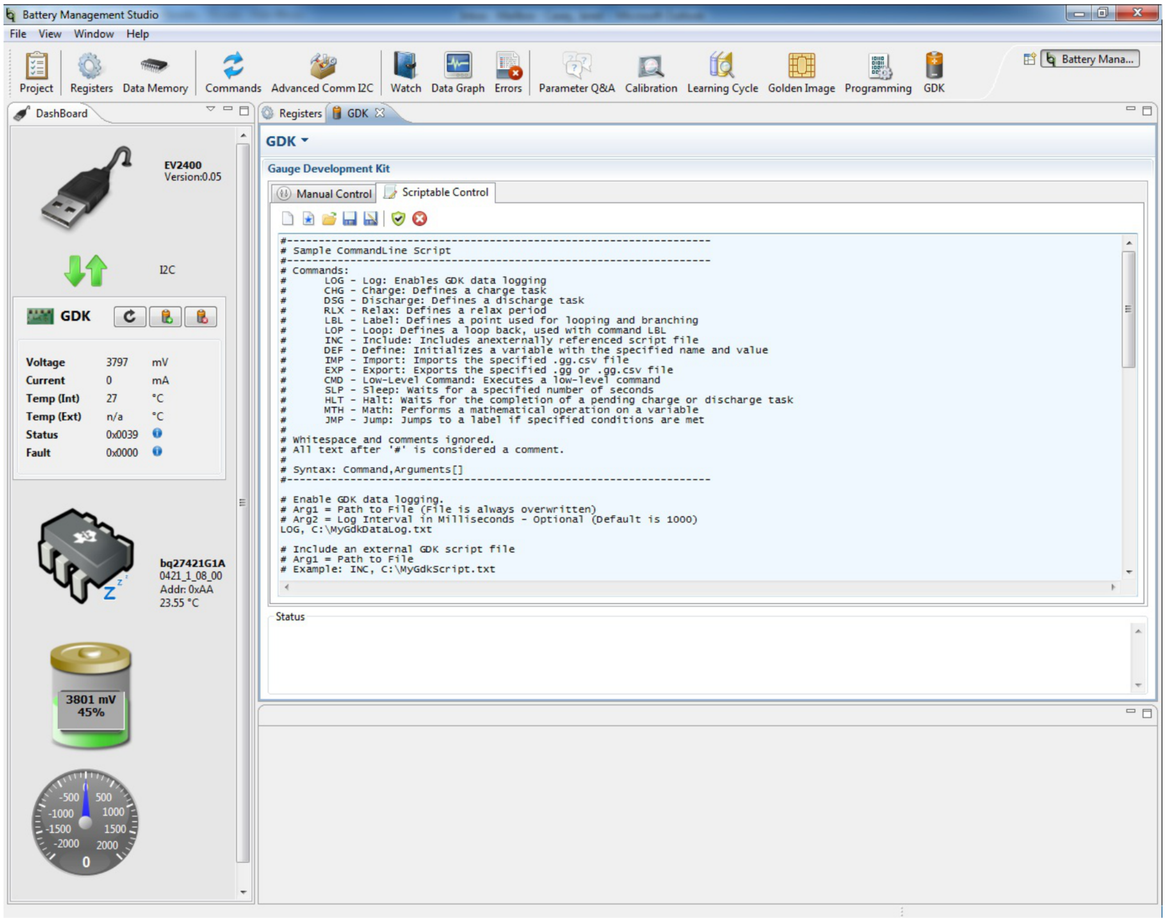
Figure 6. Manual Control Tab—Discharge Control Panel While Discharging

6.8 Scriptable Control

The Scriptable Control tab in the GDK plug-in allows a customizable script to be built and executed by bqSTUDIO in conjunction with the GDK. The script can consist of any number of charge, discharge, and relaxation operations with the characteristics of each operation being customizable. This mode allows users to execute specific tests relating to fuel gauge configuration (that is, chemistry identification, learning cycle, and so on) or to execute a specified test sequence that resembles what the battery and fuel gauge would see while in the actual system.

When the Scriptable Control tab is selected, the default script will appear in the script window. bqSTUDIO also allows the user to load a previously built script or save a current script. The default script contains detailed information on various commands allowed in the script and an example on how to use some of the commands. The Scriptable Control tab in the GDK plug-in can be seen in Figure 7.

NOTE: The default script is meant to provide an example of the syntax of the various script commands. The default script should only be executed if the charge and discharge characteristics fall within the allowed specifications of the attached battery.



Voltage	3797	mV
Current	0	mA
Temp (Int)	27	°C
Temp (Ext)	n/a	°C
Status	0x0039	
Fault	0x0000	

```
#----------------------------------------------------------------
# Sample CommandLine Script
#----------------------------------------------------------------
# Commands:
#       LOG - Log: Enables GDK data logging
#       CHG - Charge: Defines a charge task
#       DSG - Discharge: Defines a discharge task
#       RLX - Relax: Defines a relax period
#       LBL - Label: Defines a point used for looping and branching
#       LOP - Loop: Defines a loop back, used with command LBL
#       INC - Include: Includes anexternally referenced script file
#       DEF - Define: Initializes a variable with the specified name and value
#       IMP - Import: Imports the specified .gg.csv file
#       EXP - Export: Exports the specified .gg or .gg.csv file
#       CMD - Low-Level Command: Executes a low-level command
#       SLP - Sleep: Waits for a specified number of seconds
#       HLT - Halt: Waits for the completion of a pending charge or discharge task
#       MTH - Math: Performs a mathematical operation on a variable
#       JMP - Jump: Jumps to a label if specified conditions are met
#
# Whitespace and comments ignored.
# All text after '#' is considered a comment.
#
# Syntax: Command,Arguments[]
#----------------------------------------------------------------

# Enable GDK data logging.
# Arg1 = Path to File (File is always overwritten)
# Arg2 = Log Interval in Milliseconds - Optional (Default is 1000)
LOG, C:\MyGdkDataLog.txt

# Include an external GDK script file
# Arg1 = Path to File
# Example: INC, C:\MyGdkScript.txt
```

Figure 7. GDK Plug-in—Scriptable Control Tab

A GDK script contains a set of commands from which the user can build specific test scenarios. Each command has the following syntax:

```
Command, Argument1,Argument2,...,ArgumentN
```

The available GDK script commands are listed in Table 8. Details for each command can be found in the tables following Table 8.

Table 8. Script Command Summary

Command	Syntax	Description
Log	LOG, path to log file, logging interval (ms)	Logs GDK data in the specified file at the specified logging interval; if no logging interval is specified the default value of 1000 ms is used
Charge	CHG, charge voltage, charge current, termination method, termination value, wait for termination (optional)	Defines a charging operation
Discharge	**Constant Current or Power**: DSG, CC, or CP, discharge current or power, termination method, termination value, wait for termination (optional) **Pulsed Load**: DSG, PL, high current value, low current value, period, duty cycle, termination method, termination value, wait for termination (optional)	Defines a discharging operation
Relax	RLX, relaxation time (minutes)	Defines a relaxation period in minutes
Label	LBL, unique label	Defines a unique point in the script that can be referenced for looping and/or branching

Table 8. Script Command Summary (continued)

Command	Syntax	Description
Loop	LOP, unique label, number of times to loop	Transitions the script to the referenced unique label the specified number of times
Include	INC, path to script file	Includes an externally referenced GDK script
Define	DEF, variable name, variable value	Initializes a variable with the specified name and value
Import	IMP, path to *.gg.csv file	Imports the specified *.gg.csv file
Export	EXP, path to *.gg.csv file	Exports a *.gg.csv file to the specified location
Low-Level Command	CMD, low-level command, variable name (if applicable)	Executes the specified low-level command or stores the value returned from the specified low-level command in the specified variable
Sleep	SLP, sleep time (seconds)	Defines a sleep period in the script execution in seconds
Halt	HLT	Waits for the current charge or discharge operation to complete
Math	MTH, operation, variable name, operand (if applicable)	Performs the specified math command on the specified variable
Jump	JMP, condition, unique label, variable name, test value	Transitions the script to the referenced unique label if the specified condition is true

6.8.1 Log (LOG) Command

The log (LOG) command instructs the script to log the GDK data in the specified file at the specified logging interval. The GDK data consists of voltage, current, and temperature information from the GDK board perspective and is separate from the DataRAM registers in the fuel gauge. If no logging interval is specified, the default value of 1000 ms will be used.

Table 9. Log (LOG) Command Details

Log (LOG) Command			
Syntax: LOG, path to log file, logging interval			
Argument	**Argument Description**	**Argument Optional**	**Range**
path to log file	Log file path, including log name; file is always overwritten	No	NA
logging interval	Logging interval in milliseconds	Yes; default value = 1000 ms	≥ 250 ms

6.8.2 Charge (CHG) Command

The charge (CHG) command instructs the script to have the GDK execute a charge operation with the specified information in the charge command arguments. If a discharge operation is executing when the CHG command is issued, the discharge operation will terminate.

Table 10. Charge (CHG) Command Details

Charge (CHG) Command			
Syntax: CHG, charge voltage, fast charge current, termination method, termination value, wait for termination			
Argument	**Argument Description**	**Argument Optional**	**Range**
charge voltage	Voltage, in mV, to which the charger will regulate when in the constant voltage portion of the charge profile	No	3504 mV to 4400 mV
charge current	Current, in mA, the charger will use during the constant current portion of the charge profile	No	105 mA to 2000 mA
termination method	Method used to terminate the charge: [TC|SC|RC|PC|SS|FC]; see Table 6 for details of charge termination methods	No	NA

Table 10. Charge (CHG) Command Details (continued)

Charge (CHG) Command			
termination value	Value used for termination method; see Table 6 for details of charge termination methods	No, except FC termination; FC termination does not need a termination value	User specified; should be based on battery specifications
wait for termination	Determines if the script should wait for charge termination or not: [true|false]	Yes; default value = true	true or false

6.8.3 Discharge (DSG) Command–Constant Current (CC)

The discharge (DSG) command instructs the script to have the GDK execute a discharge operation with the specified information in the discharge command arguments. A discharge command allows three discharge modes: constant current, constant power, and pulsed load. Table 11, Table 12, and Table 13 provide details on the different discharge modes, respectively. If a charge operation is executing when the DSG command is issued, the discharge operation will terminate.

Table 11. Constant-Current Discharge (DSG) Command Details

Discharge (DSG) Command—Constant Current (CC)			
Syntax: DSG, CC, discharge current, termination method, termination value, wait for termination			
Argument	**Argument Description**	**Argument Optional**	**Range**
CC	Sets the discharge operation to be in constant-current discharge mode	No	NA
discharge current	Current, in mA, the GDK load will use during the discharge	No	5 mA to 2000 mA
termination method	Method used to terminate the discharge: [TV|SC|RC|PC|SS|S1|SF]; see Table 7 for details of discharge termination methods	No	NA
termination value	Value used for termination method; see Table 7 for details of discharge termination methods	No, except S1 and SF termination; S1 and SF termination methods do not need a termination value	User specified; should be based on battery specifications
wait for termination	Determines if the script should wait for discharge termination or not: [true|false]	Yes; default value = true	true or false

6.8.4 Discharge (DSG) Command–Constant Power (CP)

The discharge (DSG) command instructs the script to have the GDK execute a discharge operation with the specified information in the discharge command arguments. A discharge command allows three discharge modes: constant current, constant power, and pulsed load. Table 11, Table 12, and Table 13 provide details on the different discharge modes, respectively. If a charge operation is executing when the DSG command is issued, the discharge operation will terminate.

Table 12. Constant-Power Discharge (DSG) Command Details

Discharge (DSG) Command—Constant Power (CP)			
Syntax: DSG, CP, discharge power, termination method, termination value, wait for termination			
Argument	**Argument Description**	**Argument Optional**	**Range**
CP	Sets the discharge operation to be in constant-power discharge mode	No	NA
discharge power	Power, in mW, the GDK load will use to determine the discharge current during the discharge	No	Load current must be in the 5 mA to 2000 mA range
termination method	Method used to terminate the discharge: [TV|SC|RC|PC|SS|S1|SF]; see Table 7 for details of discharge termination methods	No	NA

Table 12. Constant-Power Discharge (DSG) Command Details (continued)

Discharge (DSG) Command—Constant Power (CP)			
termination value	Value used for termination method; see Table 7 for details of discharge termination methods	No, except S1 and SF termination; S1 and SF termination methods do not need a termination value	User specified; should be based on battery specifications
wait for termination	Determines if the script should wait for discharge termination or not: [true|false]	Yes; default value = true	true or false

6.8.5 Discharge (DSG) Command–Pulsed Load (PL)

The discharge (DSG) command instructs the script to have the GDK execute a discharge operation with the specified information in the discharge command arguments. A discharge command allows three discharge modes: constant current, constant power, and pulsed load. Table 11, Table 12, and Table 13 provide details on the different discharge modes, respectively. If a charge operation is executing when the DSG command is issued, the discharge operation will terminate.

Table 13. Pulsed-Load Discharge (DSG) Command Details

Discharge (DSG) Command—Pulsed Load (PL)			
Syntax: DSG, PL, high current value, low current value, period, duty cycle, termination method, termination value, wait for termination			
Argument	**Argument Description**	**Argument Optional**	**Range**
PL	Sets the discharge operation to be in pulsed-load discharge mode	No	NA
high current value	Current, in mA, the GDK load will use during the high portion of the pulsed load discharge	No	5 mA to 2000 mA
low current value	Current, in mA, the GDK load will use during the low portion of the pulsed load discharge	No	5 mA to 2000 mA
period	The period, in ms, of the pulsed load	No	50 ms to 45000 ms
duty cycle	The duty cycle, in %, of the pulsed load	No	10% to 90%
termination method	Method used to terminate the discharge: [TV|SC|RC|PC|SS|S1|SF]; see Table 7 for details of discharge termination methods	No	NA
termination value	Value used for termination method; see Table 7 for details of discharge termination methods	S1 and SF termination methods do not need a termination value	User specified; should be based on battery specifications
wait for termination	Determines if the script should wait for discharge termination or not: [true|false]	Yes; default value = true	true or false

6.8.6 Relax (RLX) Command

The relax (RLX) command instructs the script to have the GDK execute a relaxation operation for the specified amount of minutes.

Table 14. Relax (RLX) Command Details

Relax (RLX) Command			
Syntax: RLX, relaxation time			
Argument	**Argument Description**	**Argument Optional**	**Range**
Relaxation time	Amount of time, in minutes, the GDK will not apply either a charge or discharge to the connected battery	No	≥ 1

6.8.7 Label (LBL) Command

The label (LBL) command defines a unique point in the script that is used as a reference to transfer script execution for looping and/or branching. Both the loop and jump commands work in conjunction with the label command.

Table 15. Label (LBL) Command Details

Label (LBL) Command			
Syntax: LBL, unique label			
Argument	**Argument Description**	**Argument Optional**	**Range**
unique label	Creates a return point for the loop or jump command	No	NA

6.8.8 Loop (LOP) Command

The loop (LOP) command transfers the script execution to the referenced unique label the specified number of times. If the unique label does not exist, the script will halt execution.

Table 16. Loop (LOP) Command Details

Loop (LOP) Command			
Syntax: LOP, unique label, number of times to loop			
Argument	**Argument Description**	**Argument Optional**	**Range**
unique label	References a return point, defined by a LBL command, to transfer script execution	No	NA
number of times to loop	The number of times to execute the loop back to a return point	No	≥ 1

6.8.9 Include (INC) Command

The include (INC) command instructs the script to include an externally referenced GDK script. The INC command will execute all commands in the referenced script before moving onto further commands in the local script.

Table 17. Include (INC) Command Details

Include (INC) Command			
Syntax: INC, path to script file			
Argument	**Argument Description**	**Argument Optional**	**Range**
path to script file	Script file path, including script name	No	NA

6.8.10 Define (DEF) Command

The define (DEF) command initializes a variable with the specified name and value. The specified value can be user defined or come from a value returned by a low-level command.

Table 18. Define (DEF) Command Details

Define (DEF) Command			
Syntax: DEF, variable name, variable value			
Argument	**Argument Description**	**Argument Optional**	**Range**
variable name	Name used to reference a variable throughout the script	No	NA
variable value	Value to store in the specified variable	No	NA

6.8.11 Import (IMP) Command

The import (IMP) command instructs the script to import the specified *.gg.csv file and write the *.gg.csv file settings to the data memory of the connected fuel gauge.

Table 19. Import (IMP) Command Details

Import (IMP) Command			
Syntax: IMP, path to *.gg.csv file			
Argument	**Argument Description**	**Argument Optional**	**Range**
path to *.gg.csv file	*.gg.csv file path, including file name	No	NA

6.8.12 Export (EXP) Command

The export (EXP) command instructs the script to export the data memory settings of the connected fuel gauge to the specified *.gg.csv file.

Table 20. Export (EXP) Command Details

Export (EXP) Command			
Syntax: EXP, path to *.gg.csv file			
Argument	**Argument Description**	**Argument Optional**	**Range**
path to *.gg.csv file	*.gg.csv file path, including file name	No	NA

6.8.13 Low-level (CMD) Command

The low-level (CMD) command instructs the script to execute the specified low-level command or stores the value returned from the specified low-level command in the specified variable. See Table 22 through Table 25 for supported low-level commands.

Table 21. Low-level (CMD) Command Details

Low-level Command (CMD) Command			
Syntax: CMD, low-level command, variable name			
Argument	**Argument Description**	**Argument Optional**	**Range**
Low-level command	Name of low-level command to execute; see Table 22 through Table 25 for details on provided low-level commands	No	seeTable 22 through Table 25
variable name	Name used to reference the returned value of a specified low-level command, if applicable	Yes, only low-level commands that return a value to be stored need to specify a variable name	NA

6.8.13.1 GDK Standard Commands

Table 22 shows the GDK Standard Commands available in the bqSTUDIO scriptable control of the GDK.

Table 22. Scriptable Command Mapping—GDK Standard Commands

Standard Command	bqSTUDIO Read Command	bqSTUDIO Write Command
Status()	GDK_READ_STATUS	—
HWVersion()	GDK_READ_HW_VERSION	—
FWVersion()	GDK_READ_FW_VERSION	—
Fault()	GDK_READ_FAULTS	—
BusVoltage()	GDK_READ_BUS_VOLTAGE	—
BusCurrent()	GDK_READ_BUS_CURRENT	—

Table 22. Scriptable Command Mapping—GDK Standard Commands (continued)

Standard Command	bqSTUDIO Read Command	bqSTUDIO Write Command
AverageVoltage()	GDK_READ_BUS_VOLTAGE_AVERAGE	—
AverageCurrent()	GDK_READ_BUS_CURRENT_AVERAGE	—
Internal_Temp()	GDK_READ_TEMPERATURE_INTERNAL	—
External_Temp()	GDK_READ_TEMPERATURE_EXTERNAL	—
bq24192_Fault()	GDK_READ_BQ24192_FAULTS	—

6.8.13.2 Fuel Gauge Standard Commands

Table 23 shows the fuel gauge Standard Commands available in the bqSTUDIO scriptable control of the GDK.

Table 23. Scriptable Command Mapping—Fuel Gauge Standard Commands

Standard Command	bqSTUDIO Read Command	bqSTUDIO Write Command
Voltage()	GAUGE_READ_VOLTAGE	—
Temperature()	GAUGE_READ_TEMPERATURE	—
Current()	GAUGE_READ_CURRENT	—
StateOfCharge()	GAUGE_READ_STATE_OF_CHARGE	—
Power()	GAUGE_READ_POWER	—
RemainingCapacity()	GAUGE_READ_REMAINING_CAPACITY	—
Flags()	GAUGE_READ_FLAGS	—
Control_Status()	GAUGE_READ_CONTROL_STATUS	—

6.8.13.3 GDK Control() Subcommands

Table 24 shows the GDK *Control()* Subcommands available in the bqSTUDIO scriptable control of the GDK.

Table 24. Scriptable Command Mapping—GDK *Control()* Subcommands

***Control()* Subcommand**	bqSTUDIO Read Command	bqSTUDIO Write Command
STOP_CHARGE_DISCHARGE	—	GDK_STOP_CURRENT
CONNECT_BATTERY	—	GDK_CONNECT_BATTERY
DISCONNECT_BATTERY	—	GDK_DISCONNECT_BATTERY
START_CI_DISCHARGE	—	GDK_START_DISCHARGE
START_CP_DISCHARGE	—	GDK_START_CP_DISCHARGE
START_PULSED_LOAD	—	GDK_START_PULSED_LOAD_DISCHARGE
START_CHARGE	—	GDK_START_CHARGE
CHARGE_SETFULL	—	GDK_CHARGE_SET_FULL
CHARGE_SET20PEC	—	GDK_CHARGE_SET_20PERCENT
INSERT_BATTERY	—	GDK_INSERT_BATTERY
REMOVE_BATTERY	—	GDK_REMOVE_BATTERY
RESET_GDK	—	GDK_RESET

6.8.13.4 Fuel Gauge Control() SubCommands

Table 25 shows the fuel gauge *Control()* subcommands available in the bqSTUDIO scriptable control of the GDK. The user should check with the connected fuel gauge data sheet and/or technical reference manual to ensure that the specified *Control()* subcommand is supported. To support the wide range of potential *Control()* subcommands, the GAUGE_COMMAND_GENERIC can be used. The GAUGE_COMMAND_GENERIC can be used for any *Control()* subcommand bqSTUDIO supports for the connected fuel gauge. The supported *Control()* subcommands can be seen in the bqSTUDIO Commands Plug-in.

Table 25. Scriptable Command Mapping—Fuel Gauge Control() SubCommands

Control() SubCommand	bqSTUDIO Read Command	bqSTUDIO Write Command
CONTROL_STATUS	GAUGE_COMMAND_CONTROL_STATUS	—
DEVICE_TYPE	GAUGE_COMMAND_DEVICE_TYPE	—
FW_VERSION	GAUGE_COMMAND_FW_VERSION	—
CHEM_ID	GAUGE_COMMAND_CHEM_ID	—
BAT_INSERT	GAUGE_COMMAND_BAT_INSERT	—
SEALED	GAUGE_COMMAND_SEALED	—
SOFT_RESET	GAUGE_COMMAND_SOFT_RESET	—
RESET	GAUGE_COMMAND_RESET	—
	GAUGE_COMMAND_GENERIC	—

6.8.14 Sleep (SLP) Command

The sleep (SLP) command instructs the script to pause further execution for the specified amount of seconds. If a charge or discharge operation had been previously defined with the wait for termination argument set to false, the charge or discharge operation will continue to execute.

Table 26. Sleep (SLP) Command Details

Sleep (SLP) Command			
Syntax: SLP, sleep time			
Argument	**Argument Description**	**Argument Optional**	**Range**
sleep time	Amount of time, in seconds, to pause the script execution; will not pause a charge or discharge operation if wait for termination was set to false.	No	≥ 1

6.8.15 Halt (HLT) Command

The halt (HLT) command instructs the script to wait for further execution until the previously defined charge or discharge operation terminates.

Table 27. Halt (HLT) Command Details

Halt (HLT) Command			
Syntax: HLT			
Argument	**Argument Description**	**Argument Optional**	**Range**
NA	NA	NA	NA

6.8.16 Math (MTH) Command

The math (MTH) command instructs the script to perform the specified math command on the specified variable. See Table 29 for supported math operations.

Table 28. Math (MTH) Command Details

<table>
<tr><th colspan="4">Math (MTH) Command</th></tr>
<tr><td colspan="4">Syntax: MTH, operation, variable name, operand</td></tr>
<tr><th>Argument</th><th>Argument Description</th><th>Argument Optional</th><th>Range</th></tr>
<tr><td>operation</td><td>Specifies the math operation to be executed; see Table 29 for details on the provided math operations.</td><td>No</td><td>See Table 27</td></tr>
<tr><td>variable name</td><td>Specifies the variable the specified math operation will be performed on; the result of the math operation will be stored in the specified variable.</td><td>No</td><td>NA</td></tr>
<tr><td>operand</td><td>Specifies the second operand, if applicable, for the specified math operation; see Table 29 for details on the provided math operations.</td><td>Yes</td><td>NA</td></tr>
</table>

Table 29. Math Operation Details

Operation	Syntax	Description
ROUND	MTH, ROUND, variableName	Rounds the specified variable to the nearest integer
ADD	MTH, ADD, variableName, operand	Adds the integer value of the operand to the specified variable and stores the new value, rounded to the nearest integer, in the specified variable
ADD_FLOAT	MTH, ADD_FLOAT, variableName, operand	Adds the floating-point value of the operand to the specified variable and stores the new value in the specified variable
SUBTRACT	MTH, SUBTRACT, variableName, operand	Subtracts the integer value of the operand from the specified variable and stores the new value, rounded to the nearest integer, in the specified variable
SUBTRACT_FLOAT	MTH, SUBTRACT_FLOAT, variableName, operand	Subtracts the floating-point value of the operand from the specified variable and stores the new value in the specified variable
MULTIPLY	MTH, MULTIPLY, variableName, operand	Multiplies the specified variable by the integer value of the operand and stores the new value, rounded to the nearest integer, in the specified variable
MULTIPLY_FLOAT	MTH, MULTIPLY_FLOAT, variableName, operand	Multiplies the specified variable by the floating-point value of the operand and stores the new value in the specified variable
DIVIDE	MTH, DIVIDE, variableName, operand	Divides the specified variable by the integer value of the operand and stores the new value, rounded to the nearest integer, in the specified variable
DIVIDE_FLOAT	MTH, DIVIDE_FLOAT, variableName, operand	Divides the specified variable by the floating-point value of the operand and stores the new value in the specified variable
INCREMENT	MTH, INCREMENT, variableName	Increments the specified variable by 1
DECREMENT	MTH, INCREMENT, variableName	Decrements the specified variable by 1
AND_BYTE	MTH, AND_BYTE, variableName, operand	Performs a logical AND on the specified variable with the specified 1-byte operand value and stores the unsigned value in the specified variable
AND_16BIT	MTH, AND_16BIT, variableName, operand	Performs a logical AND operation on the specified variable with the provided 2-byte operand value and stores the unsigned value in the specified variable
AND_32BIT	MTH, AND_32BIT, variableName, operand	Performs a logical AND operation on the specified variable with the provided 4-byte operand and stores the unsigned result in the specified variable
OR_BYTE	MTH, OR_BYTE, variableName, operand	Performs a logical OR on the specified variable with the specified 1-byte operand value and stores the unsigned value in the specified variable

Table 29. Math Operation Details (continued)

Operation	Syntax	Description
OR_16BIT	MTH, OR_16BIT, variableName, operand	Performs a logical OR operation on the specified variable with the provided 2-byte operand value and stores the unsigned value in the specified variable
OR_32BIT	MTH, OR_32BIT, variableName, operand	Performs a logical OR operation on the specified variable with the provided 4-byte operand and stores the unsigned result in the specified variable
XOR_BYTE	MTH, XOR_BYTE, variableName, operand	Performs a logical XOR on the specified variable with the specified 1-byte operand value and stores the unsigned value in the specified variable
XOR_16BIT	MTH, XOR_16BIT, variableName, operand	Performs a logical XOR operation on the specified variable with the provided 2-byte operand value and stores the unsigned value in the specified variable
XOR_32BIT	MTH, XOR_32BIT, variableName, operand	Performs a logical XOR operation on the specified variable with the provided 4-byte operand and stores the unsigned result in the specified variable
NOT_BYTE	MTH, NOT_BYTE, variableName, operand	Performs a logical NOT on the specified variable with the specified 1-byte operand value and stores the unsigned value in the specified variable
NOT_16BIT	MTH, NOT_16BIT, variableName, operand	Performs a logical NOT operation on the specified variable with the provided 2-byte operand value and stores the unsigned value in the specified variable
NOT_32BIT	MTH, NOT_32BIT, variableName, operand	Performs a logical NOT operation on the specified variable with the provided 4-byte operand and stores the unsigned result in the specified variable
AND_BYTE_SIGNED	MTH, AND_BYTE_SIGNED, variableName, operand	Performs a logical AND on the specified variable with the specified 1-byte operand value and stores the signed value in the specified variable
AND_16BIT_SIGNED	MTH, AND_16BIT_SIGNED, variableName, operand	Performs a logical AND operation on the specified variable with the provided 2-byte operand value and stores the signed value in the specified variable
AND_32BIT_SIGNED	MTH, AND_32BIT_SIGNED, variableName, operand	Performs a logical AND operation on the specified variable with the provided 4-byte operand and stores the signed result in the specified variable
OR_BYTE_SIGNED	MTH, OR_BYTE_SIGNED, variableName, operand	Performs a logical OR on the specified variable with the specified 1-byte operand value and stores the signed value in the specified variable
OR_16BIT_SIGNED	MTH, OR_16BIT_SIGNED, variableName, operand	Performs a logical OR operation on the specified variable with the provided 2-byte operand value and stores the signed value in the specified variable
OR_32BIT_SIGNED	MTH, OR_32BIT_SIGNED, variableName, operand	Performs a logical OR operation on the specified variable with the provided 4-byte operand and stores the signed result in the specified variable
XOR_BYTE_SIGNED	MTH, XOR_BYTE_SIGNED, variableName, operand	Performs a logical XOR on the specified variable with the specified 1-byte operand value and stores the signed value in the specified variable
XOR_16BIT_SIGNED	MTH, XOR_16BIT_SIGNED, variableName, operand	Performs a logical XOR operation on the specified variable with the provided 2-byte operand value and stores the signed value in the specified variable
XOR_32BIT_SIGNED	MTH, XOR_32BIT_SIGNED, variableName, operand	Performs a logical XOR operation on the specified variable with the provided 4-byte operand and stores the signed result in the specified variable
NOT_BYTE_SIGNED	MTH, NOT_BYTE_SIGNED, variableName, operand	Performs a logical NOT on the specified variable with the specified 1-byte operand value and stores the signed value in the specified variable
NOT_16BIT_SIGNED	MTH, NOT_16BIT_SIGNED, variableName, operand	Performs a logical NOT operation on the specified variable with the provided 2-byte operand value and stores the signed value in the specified variable
NOT_32BIT_SIGNED	MTH, NOT_32BIT_SIGNED, variableName, operand	Performs a logical NOT operation on the specified variable with the provided 4-byte operand and stores the signed result in the specified variable

6.8.17 Jump (JMP) Command

The jump (JMP) command transfers script execution to the referenced unique label if the specified condition is true. If the unique label does not exist, the script will halt execution. See Table 31 for supported conditional operations.

Table 30. Jump (JMP) Command Details

Jump (JMP) Command			
Syntax: JMP, condition, unique label, variable name, test value			
Argument	**Argument Description**	**Argument Optional**	**Range**
condition	Specifies the condition, or comparison, to be performed on the provided variable and test value, see Table 31 for details on the provided conditions.	No	See Table 31
unique label	References a return point, defined by a LBL command, to transfer script execution	No	NA
variable name	Specifies the variable the specified condition will use with the test value	Yes	NA
test value	Specifies the test value the specified condition will use with the specified variable name	Yes	User specified

Table 31. Conditional Operation Details

Operation	Syntax	Description
UNCONDITIONAL	JMP, UNCONDITIONAL, Label	Transfers execution to the specified reference point automatically
EQUAL	JMP, EQUAL, Label, Variable, TestValue	Transfers execution to the specified reference point if the specified variable is equal to the specified test value
NOT_EQUAL	JMP, NOT_EQUAL, Label, Variable, TestValue	Transfers execution to the specified reference point if the specified variable is not equal to the specified test value
GREATER_THAN	JMP, GREATER_THAN, Label, Variable, TestValue	Transfers execution to the specified reference point if the specified variable is greater than the specified test value
LESS_THAN	JMP, LESS_THAN, Label, Variable, TestValue	Transfers execution to the specified reference point if the specified variable is less than the specified test value

6.8.18 Waiting for Charge or Discharge Termination

The CHG and DSG commands allow for the script execution to wait until the specified charge or discharge operation has terminated or to continue with script execution while the specified charge or discharge operation is executing. Allowing for the script to continue executing commands during a charge or discharge operation enables the user to develop a script that more closely resemble what might occur in a system.

6.9 GDK Board Configuration

The GDK can be configured in two ways: the default configuration (see Default Configuration) and the optional configuration (see Optional Configuration). To change the GDK configuration, the user will need to reroute specific GDK signals by adding or removing precision 0-Ω resistors located on the board and shorting different pins on the specified jumpers. Silk screen notes are located on the GDK board to reflect the steps outlined in this section on how to configure the GDK.

NOTE: Resistors R93, R103, and R104 are place holders or populated with precision 0-Ω resistors that are in a 1206 SMT package. Please see the Bill of Materials section for more information on specific part numbers for these resistors.

6.9.1 Default Configuration

The GDK is configured by default to allow the connection of any single-cell fuel gauge EVM that is supported by bqSTUDIO. The default configuration consists of the following connections:

- R103 is populated with a 0-Ω resistor. R103 can be found near the GDK Measurement section (see Figure 8). This connects the GDK load/charger to the external EVM.
- R104 is populated with a 0-Ω resistor. R104 can be found near the External Comms section (see Figure 8). This connects the GDK ground to the external EVM.
- JP9, pins 2 and 3 are connected together. This connects the SDA line of the GDK I^{2}C bus to the external EVM.
- JP10, pins 2 and 3 are connected together. This connects the SCL line of the GDK I^{2}C bus to the external EVM.
-

Figure 8 shows the GDK default configuration.

NOTE: R93 should not be populated for the default configuration.

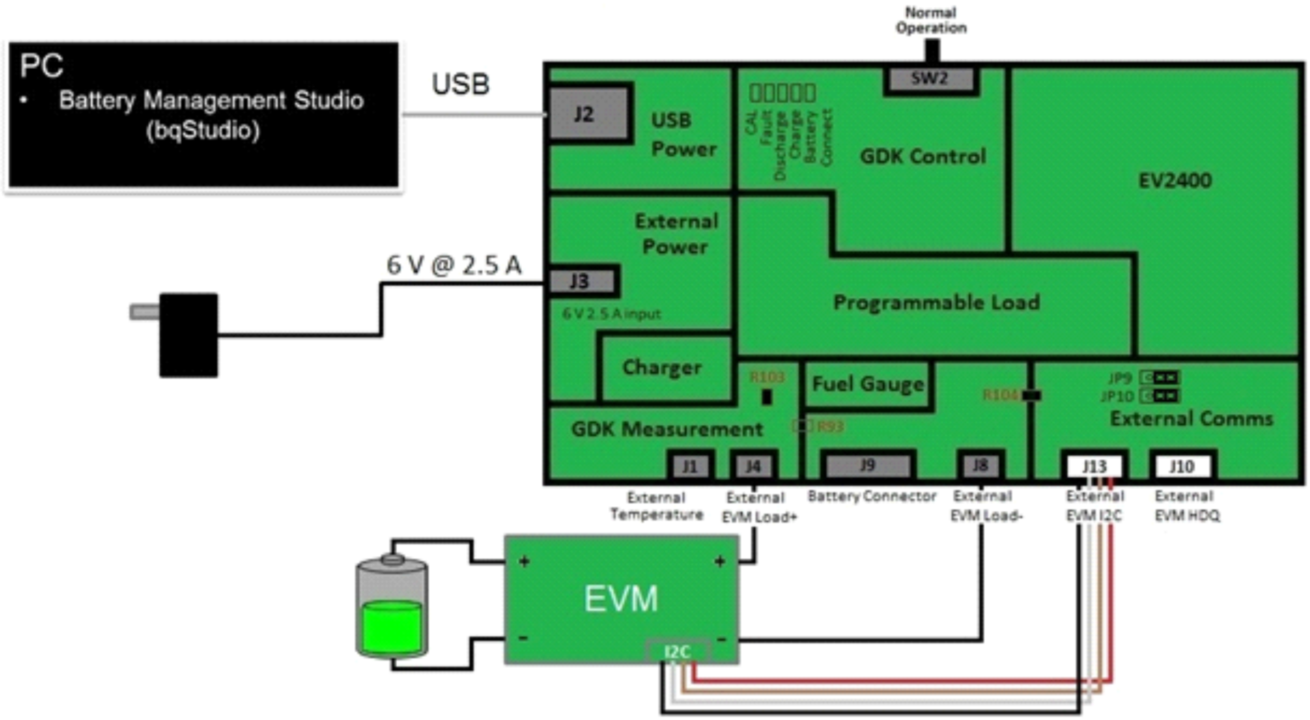


Figure 8. GDK Default Configuration—External EVM Connected

6.9.2 Optional Configuration

The GDK can also be configured to use the optional onboard fuel gauge (bq27421). The optional configuration consists of the following connections:

- R93 is populated with a 0-Ω resistor. R93 can be found near the GDK Measurement section (see Figure 9). This connects the GDK load/charger to the onboard fuel gauge.
- JP9, pins 1 and 2 are connected together. This connects the SDA line of the GDK I^2C bus to the onboard fuel gauge.
- JP10, pins 1 and 2 are connected together. This connects the SCL line of the GDK I^2C bus to the onboard fuel gauge.

Figure 9 shows the GDK default configuration.

NOTE: R103 should not be populated for the optional configuration. It is recommended that R104 is not populated for the optional configuration; however, it can remain populated without any issue.

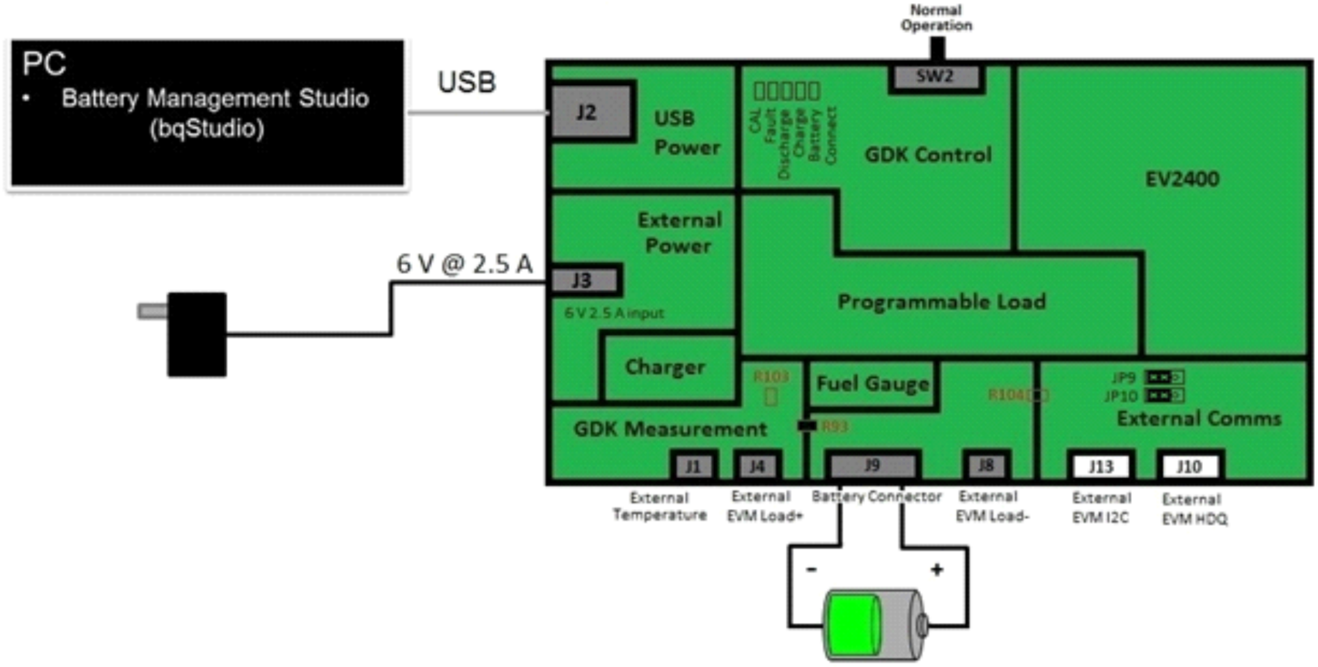


Figure 9. GDK Optional Configuration—Onboard Fuel Gauge

6.10 Firmware Updates

While the GDK comes with all of the necessary firmware to interact properly with the Battery Management Studio (bqSTUDIO) tool, future updates and enhancements to the GDK may be available periodically, and the GDK allows for these updates.

The GDK contains two separate MSP430 microcontrollers. One acts as the USB interface between the PC and the GDK/fuel gauge. This is equivalent to the standalone EV2400 module, which is also available on www.ti.com. To update this FW, see *Section 6.10.4*. The second MSP430 microcontroller acts as the controller for the charging and discharging functions of the GDK. To update this MSP430 FW, see *Section 6.10.1* through *Section 6.10.3*.

NOTE: Only proceed through the firmware update process if a newly released firmware version of the GDK has been released to the GDK product page, www.ti.com/tool/bq27gdk000evm.

6.10.1 Firmware Update PC Software

To update the GDK firmware, download and install the latest version of the *MSP430 USB Firmware Upgrade Example* PC software, which is located in the *USB Collateral Installers* section on the bottom left of the following page:

http://software-dl.ti.com/msp430/msp430_public_sw/mcu/msp430/MSP430_USB_Developers_Package/4_00_00/index_FDS.html

6.10.2 Firmware Update Image

Additionally, it is necessary to download the latest firmware image from the GDK product page, www.ti.com/tool/bq27gdk000evm.

6.10.3 Firmware Update Flow

To complete a firmware update of the GDK on the GDK board, follow these steps:

1. Download the *MSP430 USB Firmware Upgrade Example* software, as mentioned in Section 6.10.1.
2. Download the latest firmware image.
3. Verify that external power is removed.
4. Verify that the battery is disconnected from the GDK board.
5. Place SW2 in the far left position for a GDK firmware update. Refer to Figure 10 for the SW2 positions.

NOTE: To ensure that SW2 has a good connection to the proper position, push down slightly on the SW2 when moving the lever from the normal operation position.

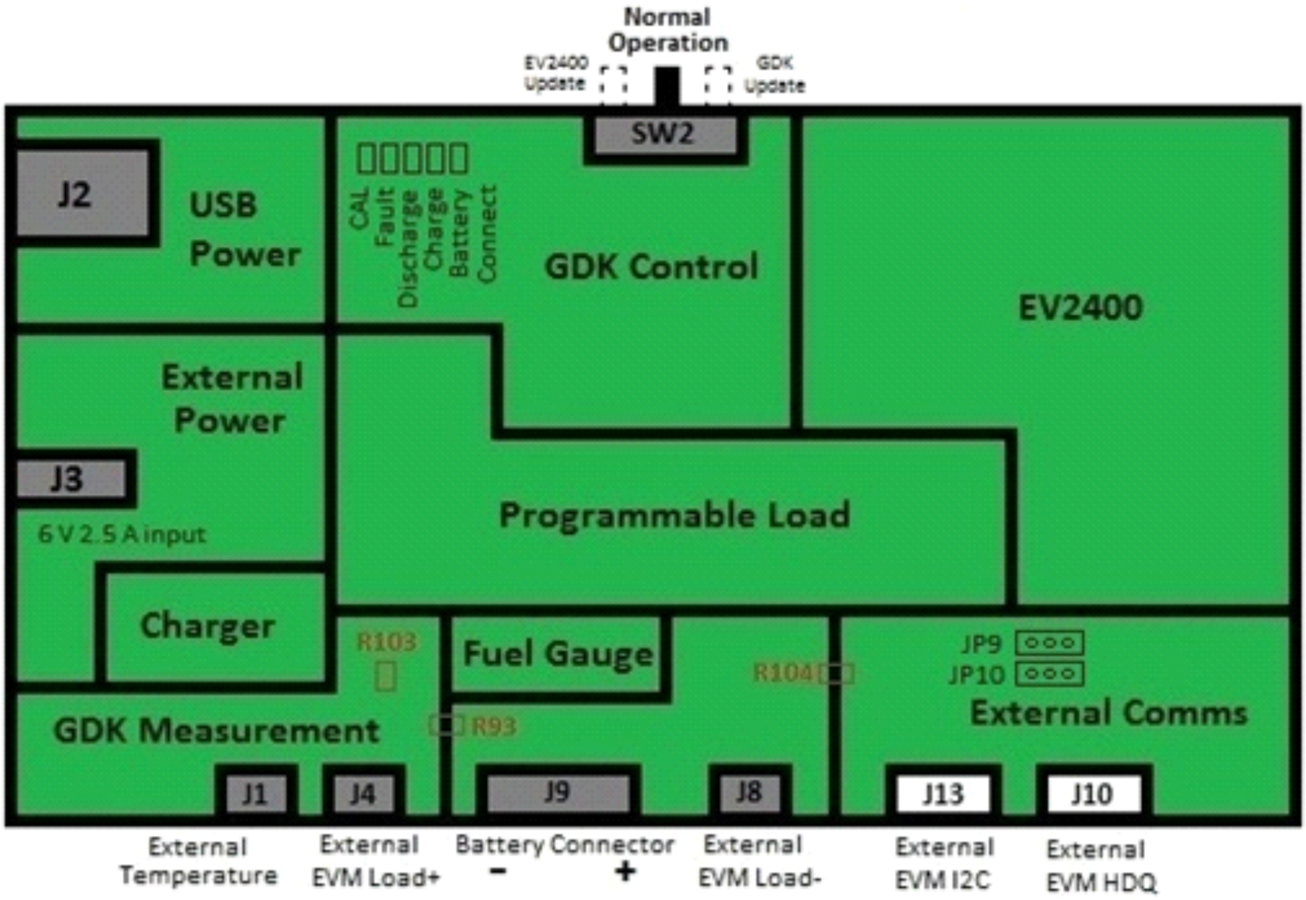
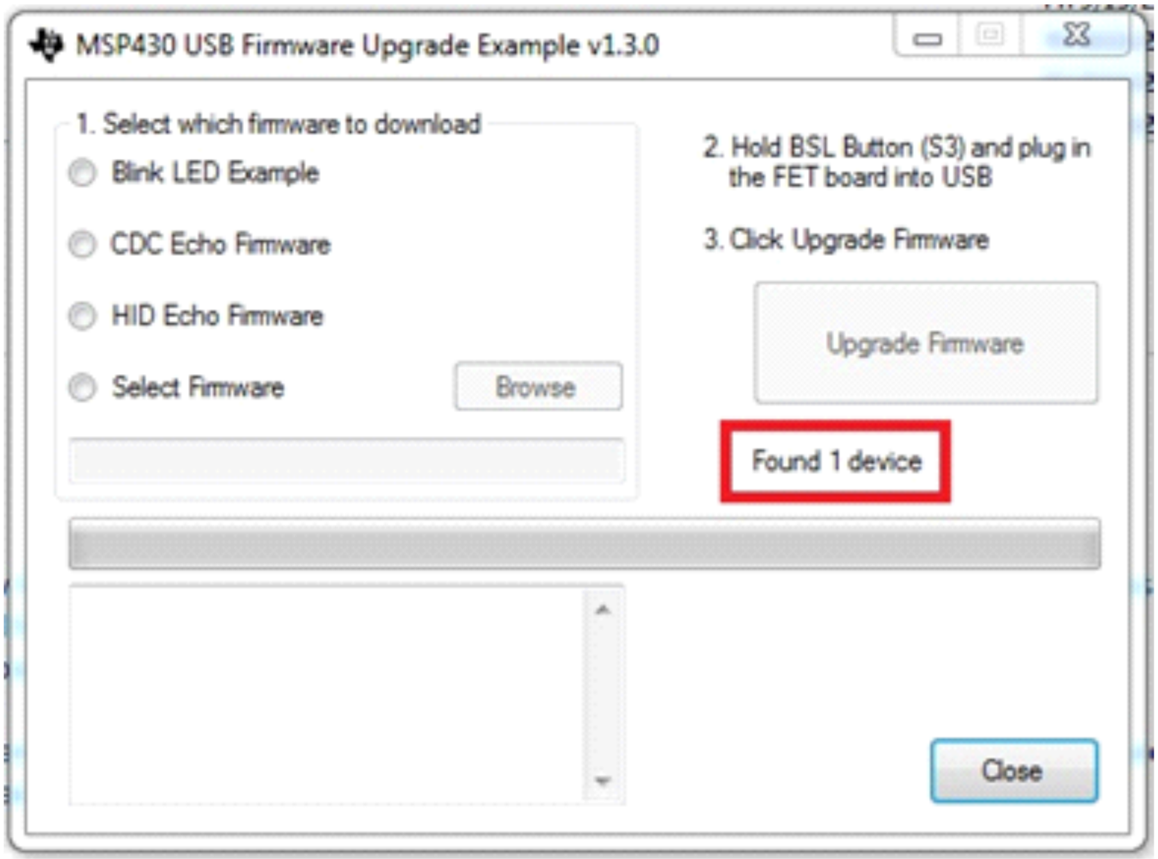


Figure 10. GDK Board Outline with FW Update Switch Positions

6. Plug in the USB power (J2).
7. Run the *MSP430 USB Firmware Upgrade Example* PC software.
8. After the software launches, click the **Next** button.
9. Accept the license agreement (the license agreement must be accepted to continue with the firmware update).
10. After the license agreement is accepted, the PC software should display the window seen in Figure 11. The software should recognize that one device was found. This is noted with the red box in Figure 11.



1. Select which firmware to download

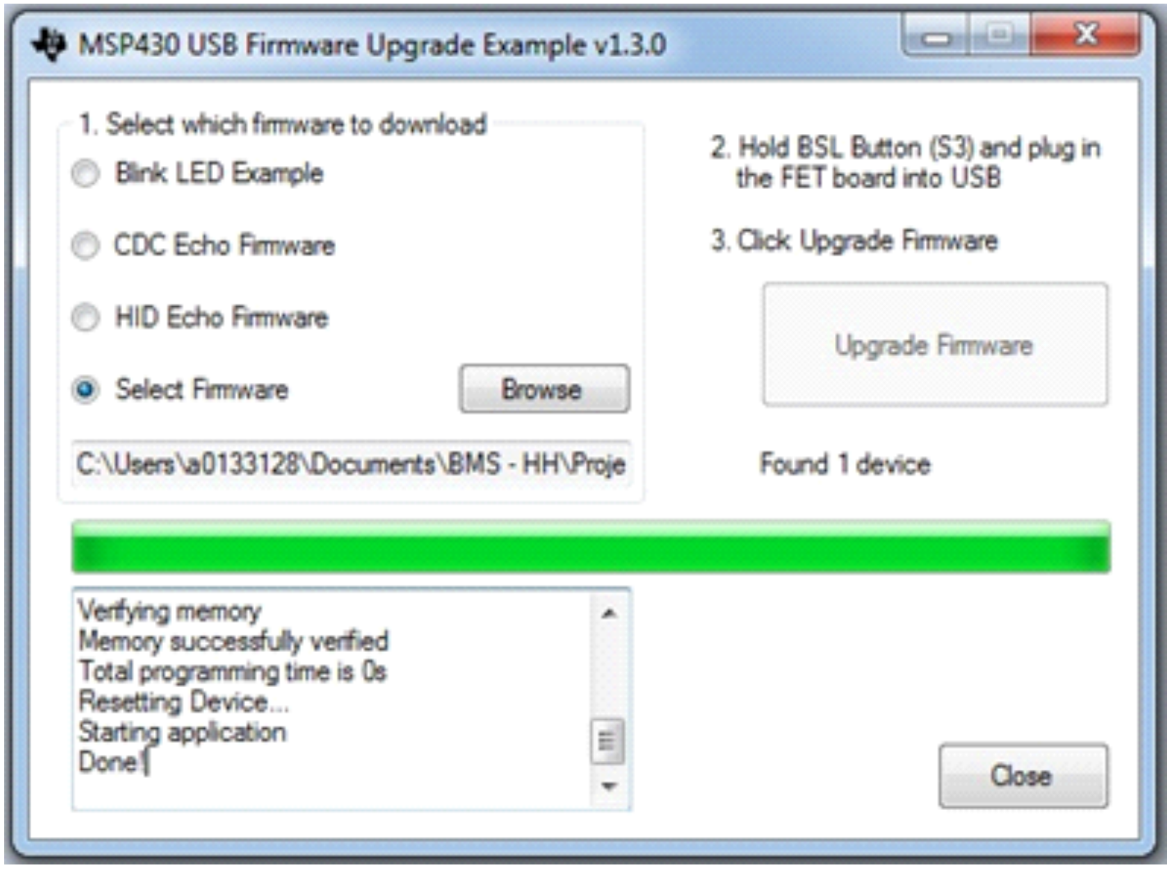
2. Hold BSL Button (S3) and plug in the FET board into USB

3. Click Upgrade Firmware

Found 1 device

Figure 11. MSP430 USB Firmware Upgrade Example Software

11. In the *Select which firmware to download* section of the screen, click on the **Select Firmware** radio button.
12. After clicking the **Select Firmware** radio button, click the **Browse** button and locate the firmware image downloaded in Step 2.
13. After selecting the firmware image, click the **Upgrade Firmware** button on the right side of the screen.
14. After the firmware update is completed, the *MSP430 USB Firmware Upgrade Example* software window should display the same content as Figure 12.



1. Select which firmware to download

○ Blink LED Example

○ CDC Echo Firmware

○ HID Echo Firmware

● Select Firmware

C:\Users\a0133128\Documents\BMS - HH\Proje

2. Hold BSL Button (S3) and plug in the FET board into USB

3. Click Upgrade Firmware

Found 1 device

```
Verifying memory
Memory successfully verified
Total programming time is 0s
Resetting Device...
Starting application
Done!
```

Figure 12. MSP430 USB Firmware Upgrade Example—Successful FW Update

15. Close the *MSP430 USB Firmware Upgrade Example* PC window.
16. Remove USB Power.
17. Place SW2 in the middle position for normal operation. See Figure 10 for reference on SW2 positions.
18. Follow the power-up sequence outlined in the *Power-Up Sequence* section.

6.10.4 EV2400 Firmware Updater

To update the firmware of the MSP430 used as an embedded EV2400, use the EV2400 Firmware Updater tool, which is located at http://www.ti.com/tool/ev2400, and follow these steps:

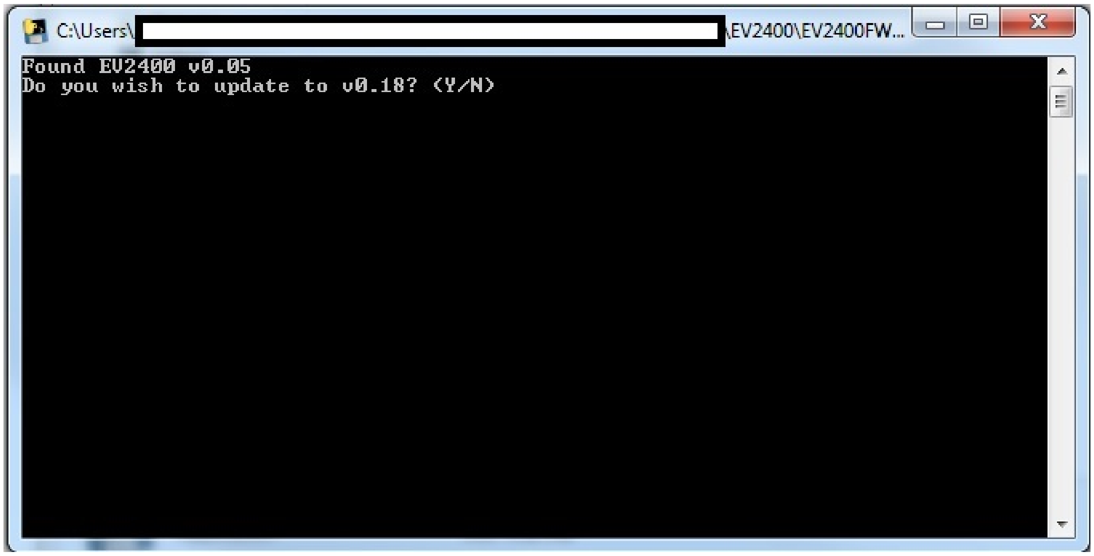
NOTE: The Battery Management Studio (bqSTUDIO) tool helps to determine the current version of the EV2400 firmware (FW) installed on a machine. To find the current version, plug in the EV2400 to a computer that has bqSTUDIO installed. After bqSTUDIO starts, the FW version is displayed in the top of the dashboard window next to the USB cable icon.

1. Download the latest EV2400 Firmware Updater tool from http://www.ti.com/tool/ev2400.
2. Open the archive with the update tool installer, and copy its contents to a temporary directory.
3. Run the installer. Take note of the location where the Firmware Updater tool is installed on the computer.
4. Connect the EV2400 that is to be updated to the computer with the EV2400 Firmware Updater tool.

NOTE: The EV2400 should remain plugged into the computer until the update is completed.

5. Ensure that no other EV2300 or EV2400 is connected to the computer being used for the firmware update.

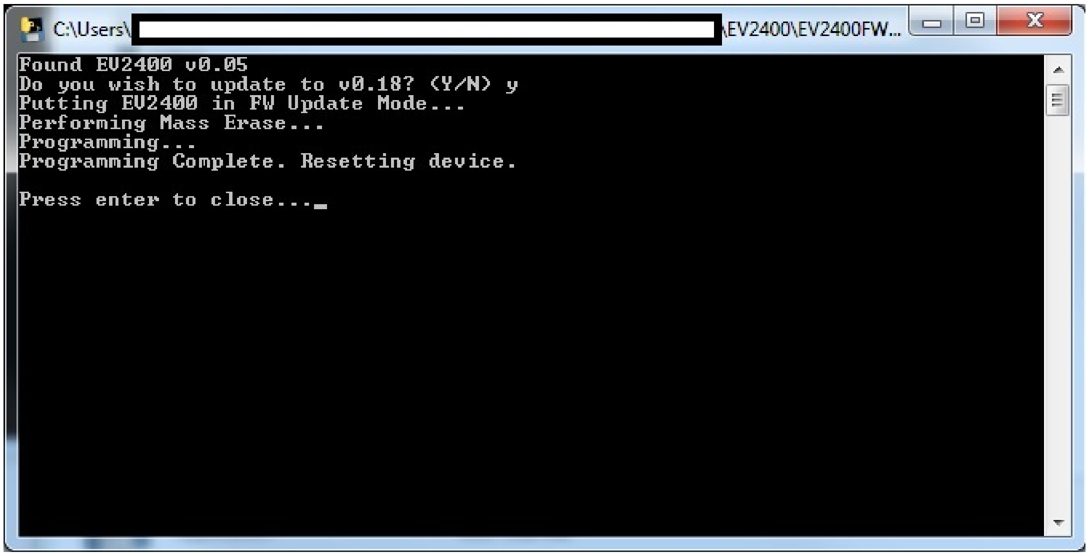
6. Go to the location of the Firmware Updater tool installed doing Step 3.
7. Run the Firmware Updater tool.
8. The upgrader tool should detect the connected EV2400, display the current firmware version, and prompt the user to continue to update the EV2400 firmware. See Figure 13.



```
Found EV2400 v0.05
Do you wish to update to v0.18? (Y/N)
```

Figure 13. Firmware Update Prompt Screen

9. Type **Y** and press **Enter**.
10. The Firmware Updater tool should place the EV2400 into FW Update mode, perform a mass erase of the older EV2400 version's firmware, program the EV2400, and then reset the device. The tool will prompt the user to continue when finished. See Figure 14.



```
Found EV2400 v0.05
Do you wish to update to v0.18? (Y/N) y
Putting EV2400 in FW Update Mode...
Performing Mass Erase...
Programming...
Programming Complete. Resetting device.

Press enter to close..._
```

Figure 14. Update Complete Screen

11. Press **Enter** to close the Firmware Updater tool.

NOTE: If during the update the EV2400 is disconnected, it is possible for the MSP on the EV2400 to remain in FW update mode. To update the device in this mode, run the Firmware Upgrader tool at the command line with the '-s' option (for example, EV2400_Updater_v0018.exe -s).

7 Standard Commands

The GDK uses a series of 2-byte standard commands to enable reading and writing of various GDK functions. Each standard command has an associated command-code pair, as indicated in Table 32. Because each command consists of two bytes of data, two consecutive I²C transmissions must be executed both to initiate the command function and to read or write the corresponding two bytes of data.

Table 32. Standard Commands

Standard Command	Address	RW	Unit	Supported in Scripting
Status()	0x00 and 0x01	R	NA	R
HWVersion()	0x04 and 0x05	R	NA	R
FWVersion()	0x06 and 0x07	R	NA	R
Fault()	0x08 and 0x09	R	NA	R
BusVoltage()	0x0A and 0x0B	R	mV	R
BusCurrent()	0x0C and 0x0D	R	mA	R
AverageVoltage()	0x12 and 0x13	R	mV	R
AverageCurrent()	0x14 and 0x15	R	mA	R
Internal_Temp()	0x16 and 0x17	R	°C	R
External_Temp()	0x18 and 0x19	R	°C	R
bq24192_Fault()	0x1A and 0x1B	R	NA	R
Control()	0x1C and 0x1D	RW	NA	RW

7.1 *Status(): 0x00 and 0x01*

Read command returns contents of GDK status register, depicting the current status of the GDK.

Table 33. Status Bit Definitions

Status								
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
High Byte	FAULT	RSVD	RSVD	DIS-CHARGING_TERM	DIS-CHARGING_PULSE	DIS-CHARGING_CP	DIS-CHARGING_CI	CHARGING_TERM
Low Byte	CHARGING	EXT_TEMP_CONN	FG_PWR_SOURCE	FG_ENABLED	EXT_PWR_GOOD	PREC_LDO_EN	LOW_BATT	BATTERY_CONNECTED

High Byte

FAULT = Indicates a fault has occurred. True when set. See Fault(): 0x08 and 0x09 for more details.

RSVD = Bits 6:5 are reserved.

DISCHARGING_TERM = Indicates that a discharge has stopped due to the discharge cutoff voltage (2800 mV) being reached. True when set.

DISCHARGING_PULSE = Indicates a pulsed-load discharge operation is executing. Active when set.

DISCHARGING_CP = Indicates a constant-power discharge operation is executing. Active when set.

DISCHARGING_CI = Indicates a constant-current discharge operation is executing. Active when set.

CHARGING_TERM = Indicates that a charge has stopped due to the charge cutoff current (–1 mA) being reached or if the onboard charger (bq24192) has terminated the charge. True when set.

Low Byte

CHARGING = Indicates a charge operation is executing. Active when set.

EXT_TEMP_CONN = Indicates that an external temperature sensor has been connected to GDK board (J1). Active when set.

FG_PWR_SOURCE = Not applicable. Always set

FG_ENABLED = Not applicable. Always set

EXT_PWR_GOOD = Indicates that the GDK recognizes the external power has been connected. Active when set.

PREC_LDO_EN = Indicates that the GDK has enable the onboard precision LDO used for onboard fuel gauge voltage calibration. Active when set.

LOW_BATT = Indicates that the GDK has taken a voltage measurement that falls below the discharge cut-off (2800 mV) voltage. Will prevent a discharge if set. This bit will clear when the voltage rises above the discharge cut-off (2800 mV) voltage. True when set.

BATTERY_CONNECTED = Indicates that the GDK has closed the isolation FETs that isolate the battery connected to the GDK board (J9) from the bus. True when set.

7.2 HWVersion(): 0x04 and 0x05

Read command returns the GDK board hardware version.

7.3 FWVersion(): 0x06 and 0x07

Read command returns the GDK board firmware (FW) version (0x0104).

7.4 Fault(): 0x08 and 0x09

Read command returns contents of the GDK fault register, depicting the current fault status of the GDK.

Table 34. Fault Bit Definitions

Fault								
	bit7	**bit6**	**bit5**	**bit4**	**bit3**	**bit2**	**bit1**	**bit0**
High Byte	RSVD	RSVD	RSVD	RSVD	EXT_PWR_REM	BATT_LDO	BQ24192_WRITE	USERDATA
Low Byte	RSVD	TEMP2	TEMP1	TMP431	DISCHG	FATAL_CHARGE	CHARGE	INIT

High Byte

RSVD = Reserved (bits 7:4)

EXT_PWR_REM = Indicates that external power has been removed from the GDK. True when set.

BATT_LDO = Indicates a fault after trying to connect the battery if the precision LDO is active or trying to connect the precision LDO if the battery has already been connected. True when set.

BQ24192_WRITE = Indicates a failed write to the onboard charger (bq24192) registers. True when set.

USERDATA = Indicates that the GDK has not had the user data previously written. True when set.

Low Byte

RSVD = Reserved (bit 7)

TEMP2 = Indicates that the temperature has gone out of –10°C to 60°C allowed temperature range for a discharge operation. True when set.

TEMP1 = Indicates that the temperature has gone out of 0°C to 45°C allowed temperature range for a charge operation. True when set.

TMP431 = Indicates the onboard temperature sensor (TMP431) has indicated a temperature, from either the internal or external sensor, that has exceeded the maximum temperature for the GDK board, 65°C. True when set.

DISCHG = Indicates a fault has occurred during discharge causing discharge to terminate. True when set.

FATAL_CHARGE = Indicates that the onboard charger (bq24192) has issued a fault and terminated charge. When set the GDK must be reset to allow for a charge operation. A charge operation cannot occur when this bit is set. True when set.

CHARGE = Indicates a fault occurred during charge and the charge operation has terminated. This fault can be cleared with the start of a new charge operation. True when set.

INIT = Indicates an error has occurred during GDK initialization. Operations should not be executed while this bit is set. True when set.

7.5 BusVoltage(): 0x0A and 0x0B

Read command returns an unsigned integer value of the bus voltage measured by the GDK, in mV.

7.6 BusCurrent(): 0x0C and 0x0D

Read command returns a signed integer value of the bus current measured by the GDK, in mA.

7.7 AverageVoltage(): 0x12 and 0x13

Read command returns an unsigned integer value of the bus voltage measured by the GDK, in mV, after a filter adjustment is applied to the *BusVoltage()* value.

7.8 AverageCurrent(): 0x14 and 0x15

Read command returns a signed integer value of the bus current measured by the GDK, in mA, after a filter adjustment is applied to the *BusCurrent()* value.

7.9 Internal_Temp(): 0x16 and 0x17

Read command returns a signed integer value of the measured onboard temperature measured by the GDK, in °C.

7.10 External_Temp(): 0x18 and 0x19

Read command returns a signed integer value of the measured external board temperature measured by the GDK, in °C, if applicable.

7.11 bq24192_Fault(): 0x1A and 0x1B

Read command returns the contents of the bq24192 Charger Fault (REG09) register; see the bq24192 data sheet (SLUSAW5) for more details.

7.12 Control(): 0x1C and 0x1D

Requires a subsequent 2-byte subcommand. These additional bytes specify the particular control function desired. The *Control()* command allows the host to control specific features of the GDK board. Read command returns the last command written. See Table 35 for details on the provided *Control()* subcommands.

Table 35. *Control()* **Subcommands**

***Control()* Subcommand**	**Address**	**RW**	**Unit**	**Supported in Scripting**
STOP_CHARGE_DISCHARGE	0x0000	W	NA	Yes
CONNECT_BATTERY	0x0001	W	NA	Yes
DISCONNECT_BATTERY	0x0002	W	NA	Yes
START_CI_DISCHARGE	0x0005	W	NA	Yes
START_CP_DISCHARGE	0x0006	W	NA	Yes
START_PULSED_LOAD	0x0007	W	NA	Yes
START_CHARGE	0x0008	W	NA	Yes
CHARGE_SETFULL	0x001A	W	NA	Yes
CHARGE_SET20PEC	0x001B	W	NA	Yes

7.12.1 STOP_CHARGE_DISCHARGE: 0x0000

Instructs the GDK to terminate a charge or discharge operation. Clears any charge- or discharge-related bits in the *Status()* register.

7.12.2 CONNECT_BATTERY: 0x0001

Instructs the GDK to close the isolation FETs that isolate the battery connected to the GDK board (J9) from the bus. Sets the *[BATTERY_CONNECTED]* bit in the *Status()* register.

7.12.3 DISCONNECT_BATTERY: 0x0002

Instructs the GDK to open the isolation FETs that isolate the battery connected to the GDK board (J9) from the bus. Clears the *[BATTERY_CONNECTED]* bit in the *Status()* register.

7.12.4 START_CI_DISCHARGE: 0x0005

Instructs the GDK to begin a constant current discharge. Sets the *[DISCHARGING_CI]* bit in the *Status()* register.

7.12.5 START_CP_DISCHARGE: 0x0006

Instructs the GDK to begin a constant power discharge. Sets the *[DISCHARGING_CP]* bit in the *Status()* register.

7.12.6 START_PULSED_LOAD: 0x0007

Instructs the GDK to begin a pulsed load discharge. Sets the *[DISCHARGING_PULSE]* bit in the *Status()* register.

7.12.7 START_CHARGE: 0x0008

Instructs the GDK to being a charge. Sets the *[CHARGING]* bit in the *Status()* register.

7.12.8 CHARGE_SETFULL: 0x001A

Instructs the GDK to clear the charger (bq24192) [FORCE_20PCT] bit in the Charge Current Control (REG02) register. See the bq24192 data sheet (SLUSAW5) for more details.

7.12.9 CHARGE_SET20PEC: 0x001B

Instructs the GDK to set the charger (bq24192) [FORCE_20PCT] bit in the Charge Current Control (REG02) register. See the bq24192 data sheet (SLUSAW5) for more details. This command allows charging currents < 512 mA.

7.12.10 INSERT_BATTERY: 0x001C

Instructs the GDK to enable the TS/BIN connection from the battery pack connector (J9) to the onboard fuel gauge. This command can simulate battery insertion or removal without removing power from the onboard fuel gauge.

7.12.11 REMOVE_BATTERY: 0x001D

Instructs the GDK to disable the TS/BIN connection from the battery pack connector (J9) to the onboard fuel gauge. This command can simulate battery insertion or removal without removing power from the onboard fuel gauge.

7.12.12 RESET_GDK: 0x00FF

Resets the GDK control.

8 Board Schematic, Layout and Bill of Materials

8.1 Schematic

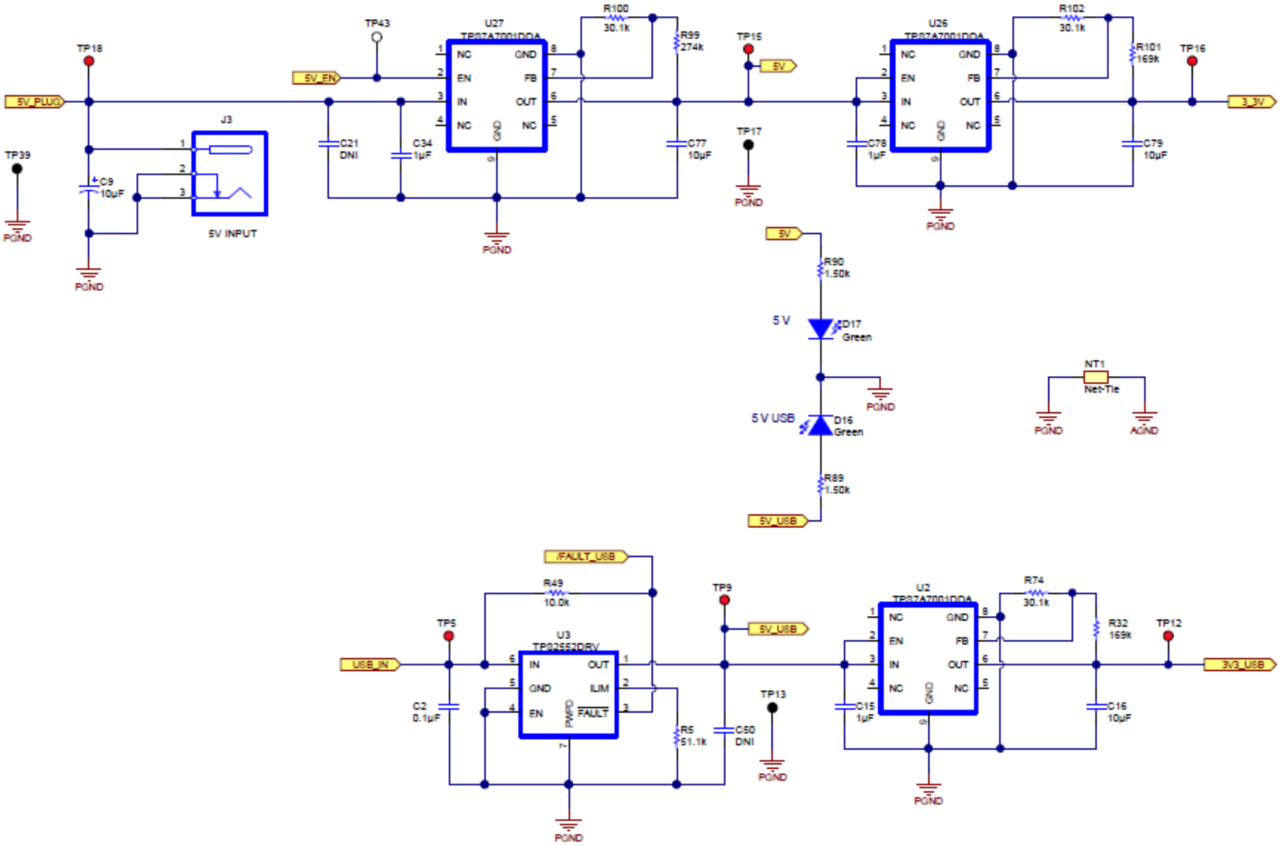


Figure 15. GDK Schematic—Power

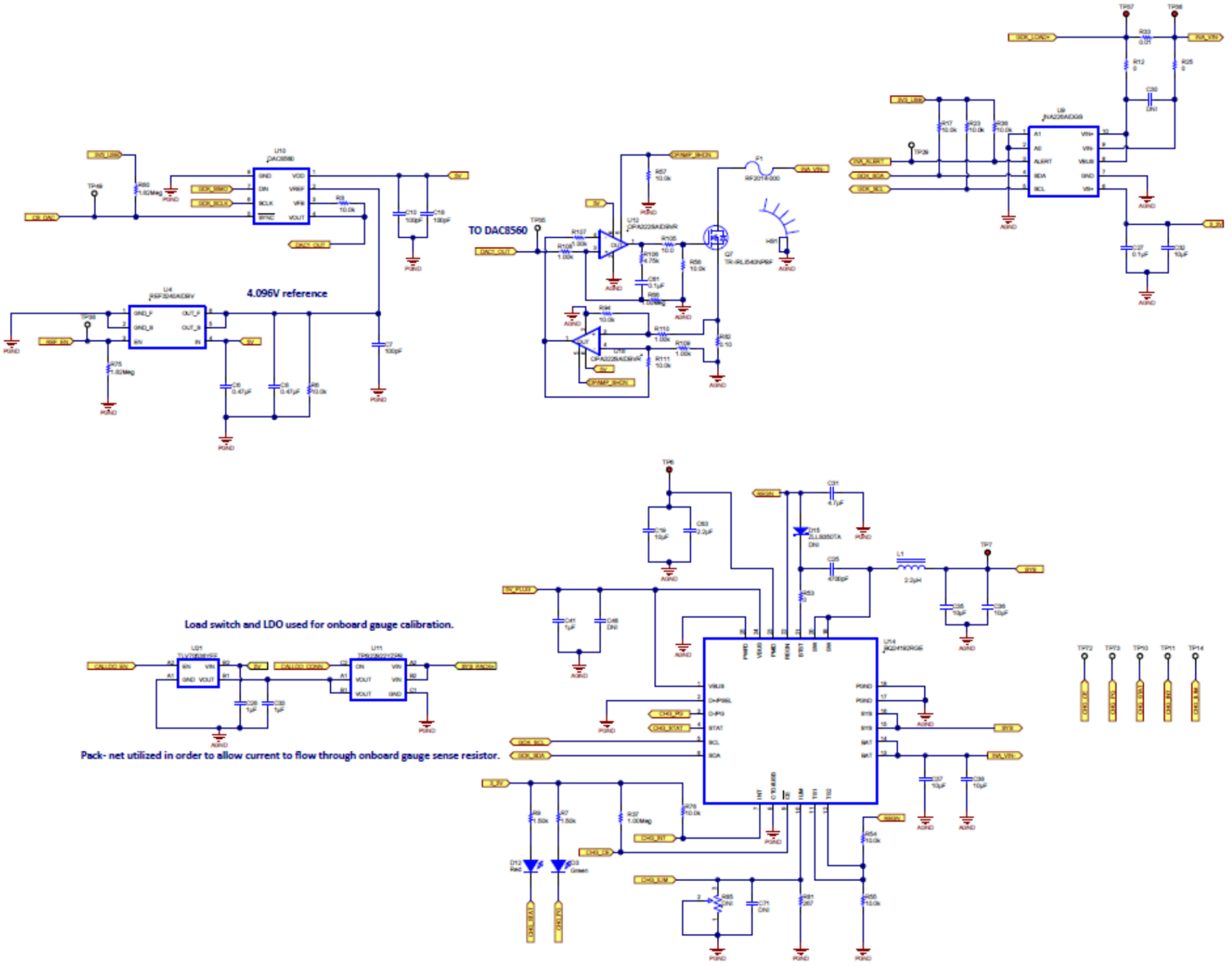
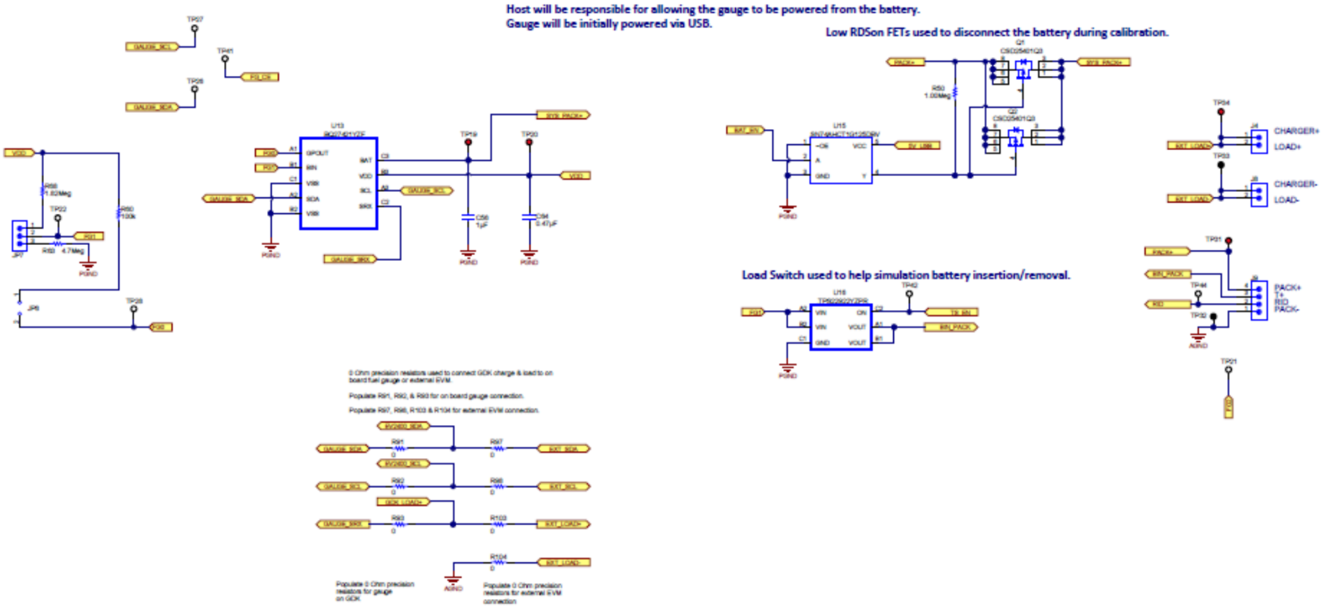

4.096V reference
TO DAC8560
Load switch and LDO used for onboard gauge calibration.
Pack- net utilized in order to allow current to flow through onboard gauge sense resistor.
U10
DAC8560
U4
REF3240AIDBV
U12
OPA2228AIDRR
U9
INA226AIDGS
U14
bq24192RGE
U11
TPS22922YZPR
U21
TLV70633YFF
Q7
F1
L1
2.2µH

Figure 16. GDK Schematic—Charger and Load



Host will be responsible for allowing the gauge to be powered from the battery.
Gauge will be initially powered via USB.

Low RDSon FETs used to disconnect the battery during calibration.

Load Switch used to help simulation battery insertion/removal.

0 Ohm precision resistors used to connect GDK charge & load to on board fuel gauge or external EVM.

Populate R91, R92, & R93 for on board gauge connection.

Populate R97, R98, R103 & R104 for external EVM connection.

Populate 0 Ohm precision resistors for gauge on GDK

Populate 0 Ohm precision resistors for external EVM connection

Figure 17. GDK Schematic—Battery Connection and Fuel Gauge

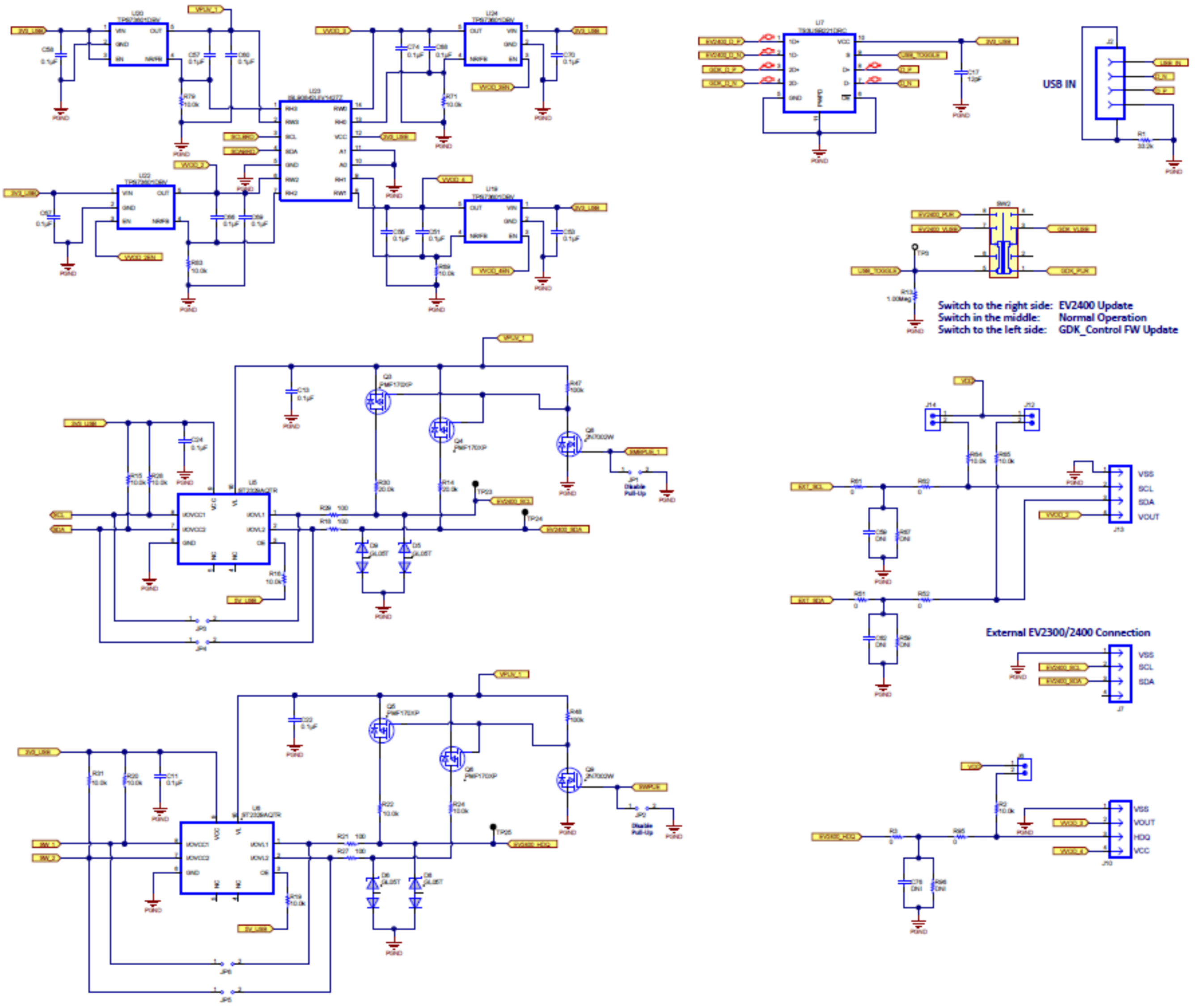

USB IN
Switch to the right side: EV2400 Update
Switch in the middle: Normal Operation
Switch to the left side: GDK_Control FW Update
External EV2300/2400 Connection
VSS
SCL
SDA
VOUT
VSS
SCL
SDA
VSS
VOUT
HDQ
VCC
TPS73601DBV
TS3USB221DRC
ST2329AQTR
PMF170XP
2N7002W
Disable Pull-Up
PGND

Figure 18. GDK Schematic—Communication

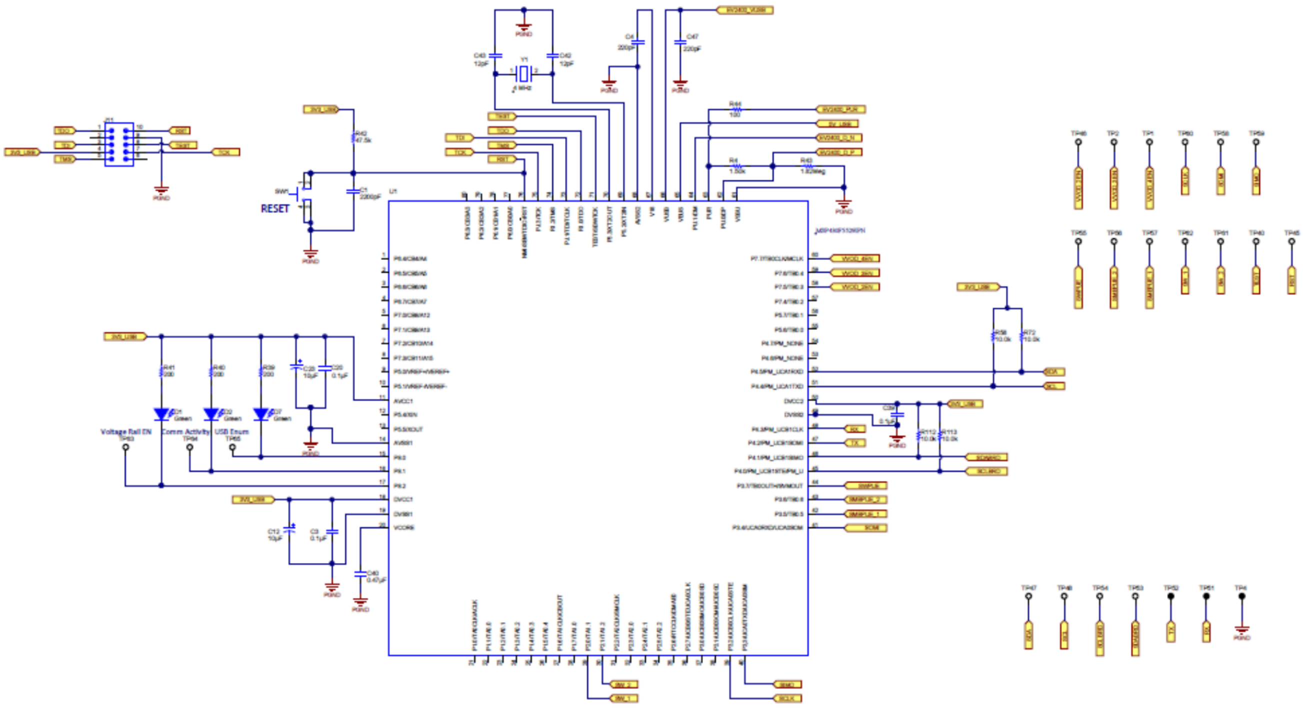


Figure 19. GDK Schematic—EV2400

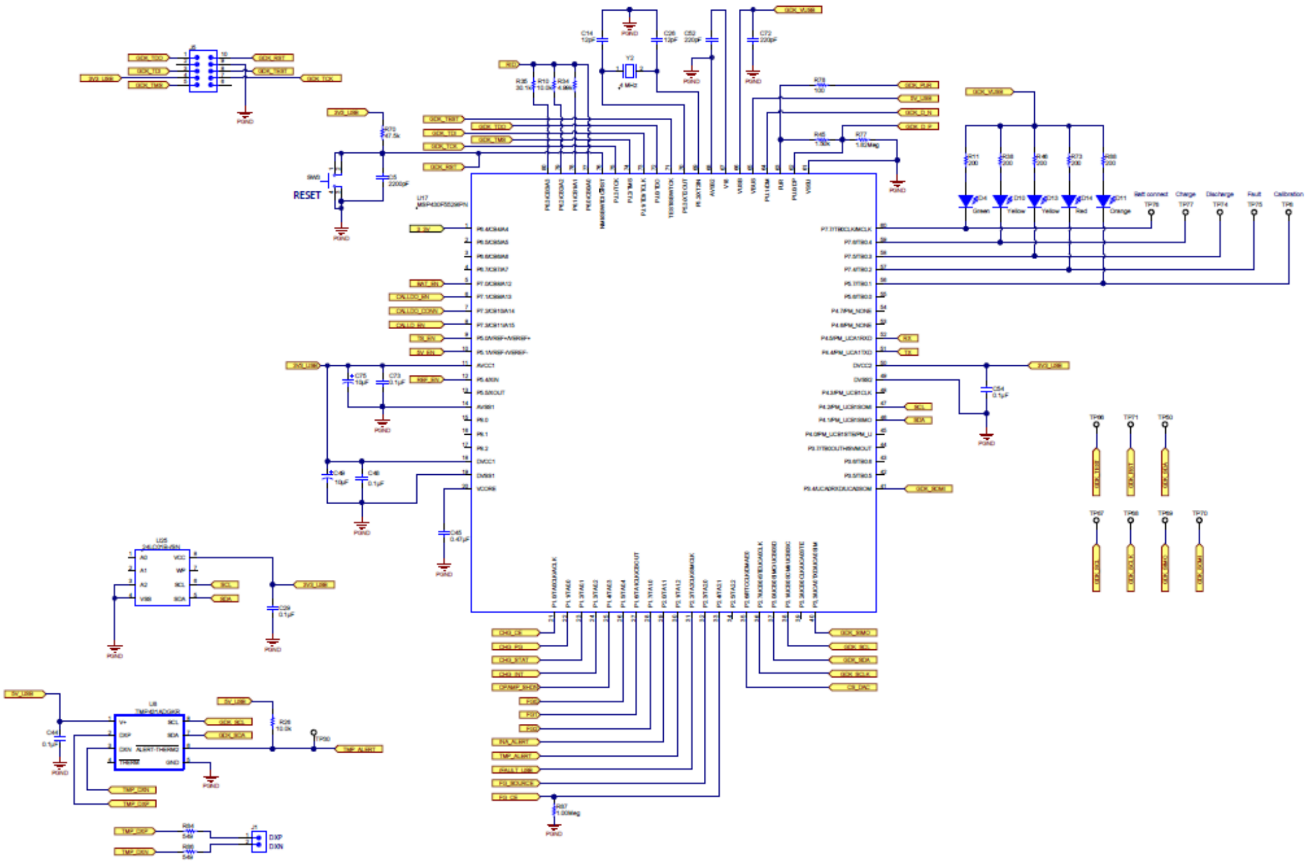


Figure 20. GDK Schematic—GDK Control

8.2 Layout

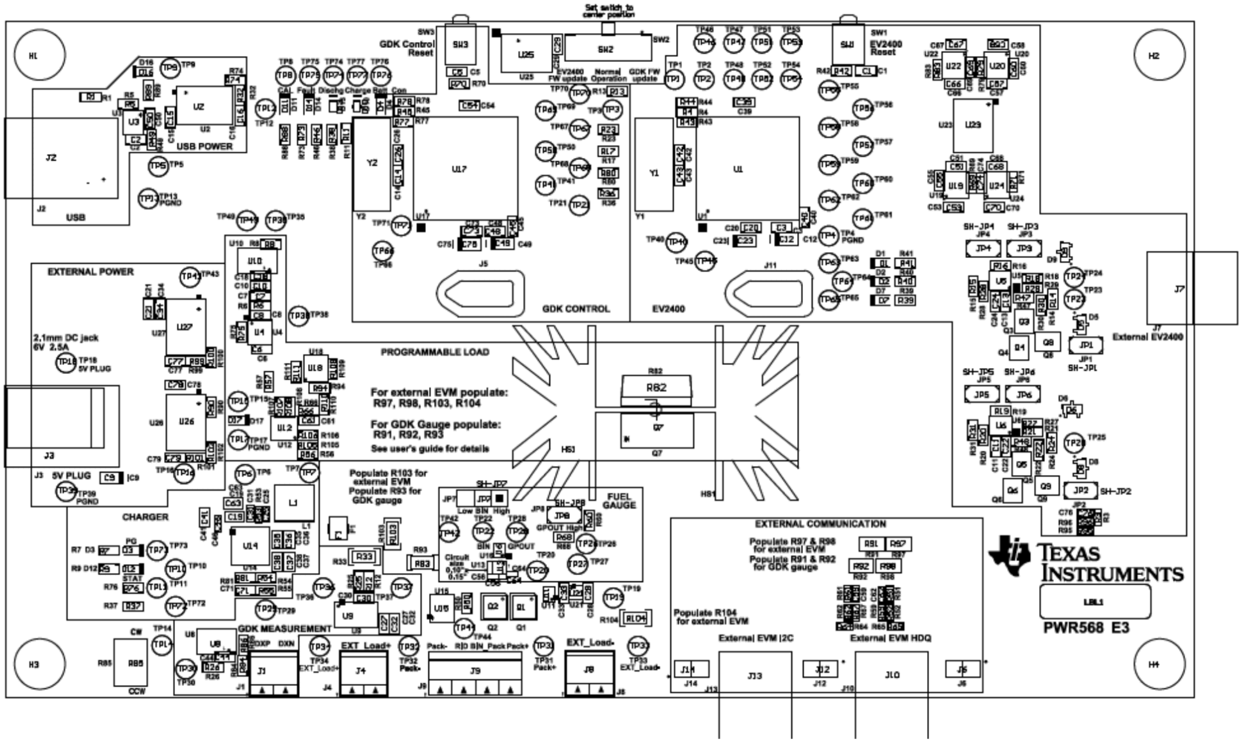


Figure 21. GDK Layout—Top Overlay and Assembly

8.3 Bill of Materials

Count	Reference Designator	Value	Description	Size	Part Number	Mfr
2	C1, C5	2200 pF	CAP, CERM, 2200pF, 50V, ±10%, X7R, 0603	0603	GRM188R71H222KA01D	MuRata
27	C2, C3, C11, C13, C20, C22, C24, C27, C29, C39, C44, C48, C51, C53, C54, C55, C57, C58, C60, C61, C66, C67, C68, C69, C70, C73, C74	0.1 uF	CAP, CERM, 0.1uF, 25V, ±10%, X7R, 0603	0603	GRM188R71E104KA01D	MuRata
4	C4, C47, C52, C72	220 pF	CAP, CERM, 220pF, 50V, ±5%, C0G/NP0, 0603	0603	GRM1885C1H221JA01D	MuRata
4	C6, C8, C40, C45	0.47 uF	CAP, CERM, 0.47uF, 10V, ±10%, X5R, 0603	0603	GRM188R61A474KA61D	MuRata
3	C7, C10, C18	100 pF	CAP, CERM, 100pF, 50V, ±5%, C0G/NP0, 0603	0603	GRM1885C1H101JA01D	MuRata
5	C9, C12, C23, C49, C75	10 uF	CAP, TA, 10uF, 10V, ±20%, 3.4 ohm, SMD	3216-18	293D106X0010A2TE3	Vishay-Sprague
5	C14, C17, C26, C42, C43	12 pF	CAP, CERM, 12pF, 50V, ±5%, C0G/NP0, 0603	0603	GRM1885C1H120JA01D	MuRata
5	C15, C28, C33, C34, C78	1 uF	CAP, CERM, 1uF, 25V, ±10%, X7R, 0603	0603	GRM188R71E105KA12D	MuRata
4	C16, C32, C77, C79	10 uF	CAP, CERM, 10uF, 6.3V, ±20%, X5R, 0603	0603	GRM188R60J106ME47D	MuRata
1	C19	10 uF	CAP, CERM, 10uF, 25V, ±10%, X5R, 0805	0805	C2012X5R1E106K125AB	TDK
1	C25	4700 pF	CAP, CERM, 4700pF, 50V, ±10%, X7R, 0603	0603	GRM188R71H472KA01D	MuRata
1	C31	4.7 uF	CAP, CERM, 4.7uF, 10V, +/-10%, X5R, 0603	0603	CGB3B1X5R1A475K055AC	TDK
4	C35, C36, C37, C38	10 uF	CAP, CERM, 10uF, 10V, +/-10%, X5R, 0805	0805	GRM21BR61A106KE19L	MuRata
1	C41	1 uF	CAP, CERM, 1uF, 25V, +/-10%, X7R, 0805	0805	GRM21BR71E105KA99L	MuRata

Count	Reference Designator	Value	Description	Size	Part Number	Mfr
1	C56	1 uF	CAP, CERM, 1uF, 10V, +/-10%, X5R, 0402	0402	GRM155R61A105KE15D	MuRata
1	C63	2.2 uF	CAP, CERM, 2.2uF, 25V, +/-10%, X5R, 0805	0805	GRM21BR61E225KA12L	MuRata
1	C64	0.47 uF	CAP, CERM, 0.47uF, 6.3V, +/-10%, X5R, 0402	0402	GRM155R60J474KE19D	MuRata
1	C65	100 pF	CAP, CERM, 100pF, 50V, +/-5%, C0G/NP0, 0603	0603	C0603C101J5GACTU	Kemet
7	D1, D2, D3, D4, D7, D16, D17	Green	LED, Green, SMD	1.6 x 0.8 x 0.8 mm	LTST-C190GKT	Lite-On
4	D5, D6, D8, D9	GL05T	Diode, TVS diode, Low Capacitance	SOT23	GL05T	Vishay
2	D10, D13	Yellow	LED, Yellow, SMD	1.6 x 0.8 x 0.8 mm	VLMY1300-GS08	Vishay-Semiconductor
1	D11	Orange	LED, Orange, SMD	1.6 x 0.8 x 0.8 mm	LTST-C190KFKT	Lite-On
2	D12, D14	Red	LED, Red, SMD	Red LED, 1.6 x 0.8 x 0.8 mm	LTST-C190CKT	Lite-On
3	D18, D19, D20	6.8 V	Diode, TVS, Uni, 6.8V, 1500W, SMC	SMC	1.5SMC6.8A	Littelfuse
3	F1, F2, F3		Fuse, Resettable, 1.9A, 6V, SMD	3.43 x 1.0 x 2.8 mm	RF1824-000	TE Connectivity
4	H1, H2, H3, H4		Machine Screw, Round, #4-40 x 1/4, Nylon, Philips panhead	Screw	NY PMS 440 0025 PH	B&F Fastener Supply
4	H5, H6, H7, H8		Standoff, Hex, 0.5"L #4-40 Nylon	Standoff	1902C	Keystone
1	HS1	529802B02500G	Heatsink, Vertical-mount With Solderable Pins	1.000 x 1.650 inch	529802B02500G	Aavid
3	J1, J4, J8		Terminal Block, 6A, 3.5mm Pitch, 2-Pos, TH	7.0 x 8.2 x 6.5 mm	ED555/2DS	On-Shore Technology, Inc.
1	J2		Connector, USB Type B, R/A, TH	USB Type B Receptacle	67068-8000	Molex
1	J3		Power Jack, mini, 2.1mm OD, R/A, TH	Jack, 14.5 x 11 x 9 mm	RAPC722X	Switchcraft
1	J9		Terminal Block, 6A, 3.5mm Pitch, 4-Pos, TH	14 x 8.2 x 6.5 mm	ED555/4DS	On-Shore Technology, Inc.
2	J10, J13	22-05-3041	Header, Friction Lock Ass'y, 4-pin Right Angle,	0.400 x 0.500 inch	22-05-3041	Molex
7	JP1, JP2, JP3, JP4, JP5, JP6, JP8	PEC02SAAN	Header, 2-pin, 100mil spacing	0.100 inch x 2	PEC02SAAN	Sullins
3	JP7, JP9, JP10	1x3	Header, TH, 100mil, 1x3, Gold plated, 230 mil above insulator	PBC03SAAN	PBC03SAAN	Sullins Connector Solutions
1	L1	2.2uH	Inductor, Shielded Drum Core, Powdered Iron, 2.2 µh, 5A, 0.0377 Ω, SMD	5.49 x 2 x 5.18 mm	IHLP2020BZER2R2M11	Vishay-Dale
1	LBL1		Thermal Transfer Printable Labels, 0.650" W x 0.200" H - 10,000 per roll	PCB Label 0.650"H x 0.200"W	THT-14-423-10	Brady
2	Q1, Q2	–20V	MOSFET, P-CH, -20V, -60A, SON 3.3x3.3mm	SON 3.3 x 3.3 mm	CSD25401Q3	Texas Instruments
4	Q3, Q4, Q5, Q6	PMF170XP	Transistor, PFET,20V, 1A, 200 mOhm	SC-70	PMF170XP	NXP
1	Q7	IRLI540npbf	MOSFET, N-ch, 100V, 23A, 44 mOhm	TO-220	IRLI540npbf	IR
8	Q8, Q9, Q10, Q11, Q12, Q13, Q14, Q15	2N7002W	MOSFET, Nch, 60V, 115mA	SOT323 [SC70]	2N7002W	Diodes
1	R1	33.2k	RES, 33.2k ohm, 1%, 0.1W, 0603	0603	CRCW060333K2FKEA	Vishay-Dale
6	R3, R51, R52, R61, R62, R95	0	RES, 0 ohm, 5%, 0.063W, 0402	0402	CRCW04020000Z0ED	Vishay-Dale
6	R4, R7, R9, R45, R89, R90	1.50k	RES, 1.50k ohm, 1%, 0.1W, 0603	0603	CRCW06031K50FKEA	Vishay-Dale
1	R5	51.1k	RES, 51.1k ohm, 1%, 0.1W, 0603	0603	CRCW060351K1FKEA	Vishay-Dale
31	R6, R8, R10, R15, R17, R20, R22, R23, R24, R26, R28, R31, R36, R49, R54, R55, R56, R57, R58, R69, R71, R72, R76, R79, R83, R94, R111, R112, R113, R116, R117	10.0k	RES, 10.0k ohm, 1%, 0.1W, 0603	0603	CRCW060310K0FKEA	Vishay-Dale
8	R11, R38, R39, R40, R41, R46, R73, R88	200	RES, 200 ohm, 1%, 0.1W, 0603	0603	CRCW0603200RFKEA	Vishay-Dale

Count	Reference Designator	Value	Description	Size	Part Number	Mfr
3	R12, R25, R53	0	RES, 0 ohm, 5%, 0.1W, 0603	0603	CRCW06030000Z0EA	Vishay-Dale
15	R13, R16, R19, R37, R50, R66, R87, R91, R92, R97, R98, R114, R115, R118, R119	1.00Meg	RES, 1.00Meg ohm, 1%, 0.1W, 0603	0603	CRCW06031M00FKEA	Vishay-Dale
2	R14, R30	20.0k	RES, 20.0k ohm, 1%, 0.1W, 0603	0603	CRCW060320K0FKEA	Vishay-Dale
6	R18, R21, R27, R29, R44, R78	100	RES, 100 ohm, 1%, 0.1W, 0603	0603	CRCW0603100RFKEA	Vishay-Dale
2	R32, R101	169k	RES, 169k ohm, 1%, 0.1W, 0603	0603	CRCW0603169KFKEA	Vishay-Dale
1	R33	0.01	RES, 0.01 ohm, 1%, 1W, 1206	1206	WSLP1206R0100FEA	Vishay-Dale
1	R34	4.99k	RES, 4.99k ohm, 1%, 0.1W, 0603	0603	CRCW06034K99FKEA	Vishay-Dale
4	R35, R74, R100, R102	30.1k	RES, 30.1k ohm, 1%, 0.1W, 0603	0603	CRCW060330K1FKEA	Vishay-Dale
2	R42, R70	47.5k	RES, 47.5k ohm, 1%, 0.1W, 0603	0603	CRCW060347K5FKEA	Vishay-Dale
5	R43, R68, R75, R77, R80	1.82Meg	RES, 1.82Meg ohm, 1%, 0.1W, 0603	0603	CRCW06031M82FKEA	Vishay-Dale
3	R47, R48, R60	100k	RES, 100k ohm, 1%, 0.1W, 0603	0603	CRCW0603100KFKEA	Vishay-Dale
1	R63	4.7Meg	RES, 4.7Meg ohm, 5%, 0.1W, 0603	0603	CRCW06034M70JNEA	Vishay-Dale
1	R65	10.0k	RES, 10.0k ohm, 1%, 0.063W, 0402	0402	CRCW040210K0FKED	Vishay-Dale
1	R81	267	RES, 267 ohm, 1%, 0.1W, 0603	0603	CRCW0603267RFKEA	Vishay-Dale
1	R82	0.1	RES, 0.1 ohm, 1%, 20W, TH	TO-220AC	PWR220T-20-R100F	Bourns
2	R84, R86	549	RES, 549 ohm, 1%, 0.1W, 0603	0603	CRCW0603549RFKEA	Vishay-Dale
1	R99	274k	RES, 274k ohm, 1%, 0.1W, 0603	0603	CRCW0603274KFKEA	Vishay-Dale
2	R103, R104	0	RES, 0 ohm, 5%, 0.25W, 1206	1206	CRCW12060000Z0EA	Vishay-Dale
1	R105	10.0	RES, 10.0 ohm, 1%, 0.1W, 0603	0603	CRCW060310R0FKEA	Vishay-Dale
1	R106	4.75k	RES, 4.75k ohm, 1%, 0.1W, 0603	0603	CRCW06034K75FKEA	Vishay-Dale
4	R107, R108, R109, R110	1.00k	RES, 1.00k ohm, 1%, 0.1W, 0603	0603	CRCW06031K00FKEA	Vishay-Dale
10	SH-JP1, SH-JP2, SH-JP3, SH-JP4, SH-JP5, SH-JP6, SH-JP7, SH-JP8, SH-JP9, SH-JP10	1x2	Shunt, 100mil, Gold plated, Black	Shunt	969102-0000-DA	3M
2	SW1, SW3		Switch, Tactile, SPST-NO, SMT	Switch, 6.1 x 1.8 x 4.6 mm	EVQ-PSD02K	Panasonic
1	SW2		Switch Slide DP3T 30V 0.2A RT Angle	12.6 x 4.7 x 6.8 mm	EG2315	E-Switch
6	TP4, TP13, TP17, TP32, TP33, TP39	Black	Test Point, TH, Miniature, Black	Keystone5001	5001	Keystone
3	TP18, TP31, TP34	Red	Test Point, TH, Miniature, Red	Keystone5000	5000	Keystone
2	TP22, TP28	White	Test Point, TH, Miniature, White	Keystone5002	5002	Keystone
2	U1, U17	MSP430F5529IPN	IC, Mixed Signal Microcontroller	TQFP-80	MSP430F5529IPN	TI
3	U2, U26, U27	TPS7A7001DDA	IC, Very Low Input, Very Low Dropout, 2-Amp Regulator With Enable	TPS7A7001xy DDAR	TPS7A7001DDAR	TI
1	U3	TPS2552DRV	IC, Power-Distribution Switch, Current-Limited	DRV	TPS2552DRV	TI
1	U4	REF3240AIDBV	IC, Series Voltage Reference, 4ppm °C Max, 100uA	SOT23-6	REF3240AIDBV	TI
2	U5, U6		Dual supply 2-bit voltage Translator/buffer/repeater/isolator for IC applications, MicroPak-8	SMD, 1.6 x 0.55 x 1.6 mm	FXMA2102L8X	Fairchild Semiconductor
1	U7	TS3USB221DRC	IC, High-Speed USB 2.0(480Mbps) 1:2 Multi/Demultiplexer Switch With Single Enable	SON-10	TS3USB221DRC	TI
1	U8	TMP431ADGKR	IC, ±1°C Temperature Sensor with Series-R, n-Factor, and Automatic Beta Compensation	MSOP	TMP431ADGKR	TI
1	U9	INA220AIDGS	IC, Bi-Directional Current/Power Monitor with I2C Interface	MSOP-10	INA220AIDGS	TI
1	U10	DAC8560	IC, 16-bit, Ultra-low glitch, Voltage output, A-D Converter, w/2.5V, 2ppm/°C Internal Reference	MSSOP-8	DAC8560ICDGKT	TI
2	U11, U16	TPS22922YZPR	IC, Low Input Voltage, Ultra-Low Ron Load Switches	SON	TPS22922YZPR	TI

Count	Reference Designator	Value	Description	Size	Part Number	Mfr
2	U12, U18	OPA322SAIDBVR	IC, Precision, 20MHz, 0.9pA, Low-Noise, RRIO, CMOS Op-Amp with Shutdown	SOT23-6	OPA322SAIDBVR	TI
1	U13	BQ27421YZF	IC,	uBGA	BQ27421YZF-G1A	TI
1	U14	BQ24192RGE	IC, I2C Controlled 4A Single Cell USB / Adaptor Charger With Narrow VDC Power Path Management and USB OTG	QFN-24	BQ24192RGE	TI
1	U15	SN74AHCT1G125 DBV	IC, Single Bus buffer gate with 3 state output	0.118 x 0.118	SN74AHCT1G125DBV	TI
4	U19, U20, U22, U24	TPS73601DBV	IC, Cap-Free, NMOS, 400mA LDO Regulator with Reverse Current Protection.	SOT23-5	TPS73601DBV	TI
1	U21		200-mA, Low IQ, Low-Noise, 3.6V LDO Regulator, YFP0004ABAB	YFP0004ABAB	TLV70536YFP	Texas Instruments
1	U23	ISL90842UIV1427 Z	IC, Quad Digitally Controlled Potentiometers	TSSOP	ISL90842UIV1427Z	Intersil
1	U25		IC, EEPROM, 8KBIT, 400KHZ, 8SOIC	SOIC-8	CAT24C08WI-GT3	ON Semiconductor
2	Y1, Y2	4 MHz	Crystal, SMT Quartz Crystal	0.484 x 0.190 inch	ATS040SM	CTS
1	W1		Cable assembly, 4 pin	Assembly	CBL002	Texas Instruments
0	C21, C30, C46, C50, C71	DNI	CAP, CERM, 10pF, 50V, +/-5%, C0G/NP0, 0603	0603	GRM1885C1H100JA01D	MuRata
0	C59, C62, C76	DNI	CAP, CERM, 1pF, 50V, +/-5%, C0G/NP0, 0402	0402	GRM1555C1H1R0CA01D	MuRata
0	D15	DNI	Diode, Schottky, 40V, 0.38A, SOD-523	SOD-523	ZLLS350TA	Diodes Inc.
0	FID1, FID2, FID3, FID4, FID5, FID6		Fiducial mark. There is nothing to buy or mount.	Fiducial	N/A	N/A
0	J6, J12, J14		Header, 100mil, 2x1, Tin plated, TH	Header 2x1	90120-0122	Molex
0	J7	22-05-3041	Header, Friction Lock Ass'y, 4-pin Right Angle,	0.400 x 0.500 inch	22-05-3041	Molex
0	R2, R64	10.0k	RES, 10.0k ohm, 1%, 0.063W, 0402	0402	CRCW040210K0FKED	Vishay-Dale
0	R59, R67, R96	DNI	RES, 0 ohm, 5%, 0.063W, 0402	0402	RC0402JR-070RL	Yageo America
0	R85	DNI	Potentiometer, 3/8 Cermet, Single-Turn	0.25 x 0.17 inch	3266W-xxx	Bourns
0	R93	0	RES, 0 ohm, 5%, 0.25W, 1206	1206	CRCW12060000Z0EA	Vishay-Dale
0	TP1, TP2, TP3, TP8, TP10, TP11, TP14, TP21, TP26, TP27, TP29, TP30, TP35, TP38, TP40, TP41, TP42, TP43, TP44, TP45, TP46, TP47, TP48, TP49, TP50, TP53, TP54, TP55, TP56, TP57, TP58, TP59, TP60, TP61, TP62, TP63, TP64, TP65, TP66, TP67, TP68, TP69, TP70, TP71, TP72, TP73, TP74, TP75, TP76, TP77	White	Test Point, TH, Miniature, White	Keystone5002	5002	Keystone
0	TP5, TP6, TP7, TP9, TP12, TP15, TP16, TP19, TP20, TP36, TP37	Red	Test Point, TH, Miniature, Red	Keystone5000	5000	Keystone
0	TP23, TP24, TP25, TP51, TP52	Black	Test Point, TH, Miniature, Black	Keystone5001	5001	Keystone

9 Safety

9.1 Supported Battery Types

Only single-cell batteries that have a charging voltage between 3.504 V and 4.4 V should be connected to the external EVM that is connected to the GDK in the default configuration or to the GDK board directly, via J9, when using the optional configuration.

The GDK is meant to be used with only rechargeable single-cell batteries. These rechargeable single-cell batteries should have a lithium ion component to their chemistry makeup. Other types of chemistries are not supported.

9.2 Battery Connection

Even though the GDK has protection to help prevent damage to the board if voltages greater than 4.4 V are connected or if reverse polarity battery connection is made, the user should pay attention to how the battery is connected. The only way to assure prevention of damage to the board is by following the GDK connections properly.

NOTE: When using the default configuration, the external fuel gauge will be damaged if the battery is connected improperly or if the incorrect type of battery is connected since the external EVM does not include protection for these types of events.

9.3 External Power Connection

A maximum of 6 V should be connected to the GDK external power connection. The recommended DC jack to use with the GDK is listed in the Board Connections section. If a power supply is being used, the voltage should be no greater than 6 V.

9.4 Charging

The user should be careful to specify charging options that fall within the manufacturer specifications of the connected battery. The GDK must rely on the information the user enters into bqSTUDIO for the charging operation. The GDK has a built-in charge temperature range of 0°C to 45°C. If the GDK recognizes a temperature outside of this range, it will immediately terminate the charge. However, it is the user’s responsibility to ensure the connected battery is charged under allowable specifications (that is, temperature, current levels, and voltage levels) provided by the battery manufacturer.

9.5 Discharging

The user should be careful to specify the discharging options that fall within the manufacturer specifications of the connected battery. The GDK must rely on the information the user enters into bqSTUDIO for the discharging operation. The GDK has a built-in discharge temperature range of –10°C to 60°C and a built-in discharge cut-off voltage of 2900 mV. If the GDK recognizes a temperature outside the specified range or a voltage below the specified value, the discharge will terminate immediately. However, it is the user’s responsibility to make sure the connected battery is discharged under allowable specifications (that is, temperature, current levels, and voltage levels) provided by the battery manufacturer.

10 Related Documentation from Texas Instruments

To obtain a copy of any of the following TI documents, go to www.ti.com.

1. *bq24192, I^{2}C Controlled 4.5A Single Cell USB/Adapter Charger With Narrow VDC Power Path Management and USB OTG* Data Sheet (SLUSAW5)
2. *bq27421-G1, System-Side Impedance Track™ Fuel Gauge With Integrated Sense Resistor* Data Sheet (SLUSB85)
3. *bq27421-G1, Technical Reference Manual* (SLUUAC5)
4. *bq27421-G1, EVM: Single-Cell Impedance Track Technology* User's Guide (SLUUA63)
5. *Quickstart Guide for Gauge Development Kit* (SLUUB24)

Revision History

Changes from Original (April 2014) to A Revision **Page**

- Changed "Battery Management Studio" to "Battery Management Studio (bqSTUDIO)" and "bqSTUDIO" 4
- Changed the *Firmware Updates* section 28
- Added the *EV2400 Firmware Update* section.. 32

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

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve 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 (also referred to herein as “components”) 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 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.

TI does not warrant or represent that any license, either express or implied, is granted under 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.

Reproduction of significant portions of TI information in TI data books or data sheets 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.

Resale of TI components or services with statements different from or beyond the parameters stated by TI for that component or service voids all express and any implied warranties for the associated TI component or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

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 which anticipate dangerous consequences of failures, monitor failures and their consequences, lessen the likelihood of failures that might cause harm 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 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 a special agreement specifically governing such use.

Only those TI components which 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 which have ***not*** been so designated is solely at the Buyer's risk, and that 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.

Products		**Applications**	
Audio	www.ti.com/audio	Automotive and Transportation	www.ti.com/automotive
Amplifiers	amplifier.ti.com	Communications and Telecom	www.ti.com/communications
Data Converters	dataconverter.ti.com	Computers and Peripherals	www.ti.com/computers
DLP® Products	www.dlp.com	Consumer Electronics	www.ti.com/consumer-apps
DSP	dsp.ti.com	Energy and Lighting	www.ti.com/energy
Clocks and Timers	www.ti.com/clocks	Industrial	www.ti.com/industrial
Interface	interface.ti.com	Medical	www.ti.com/medical
Logic	logic.ti.com	Security	www.ti.com/security
Power Mgmt	power.ti.com	Space, Avionics and Defense	www.ti.com/space-avionics-defense
Microcontrollers	microcontroller.ti.com	Video and Imaging	www.ti.com/video
RFID	www.ti-rfid.com		
OMAP Applications Processors	www.ti.com/omap	**TI E2E Community**	e2e.ti.com
Wireless Connectivity	www.ti.com/wirelessconnectivity		

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