Getting Started with the BQ79606A-Q1 GUI

This document provides a guide to install, setup and use the BQ79606A-Q1 graphical user's interface (GUI). In this document only the basic functionalities of the GUI will be documented with the intent of reducing the time needed to become familiar with the GUI.

Contents

List of Figures

Trademarks

All trademarks are the property of their respective owners.

1 Getting Started

1.1 Download the GUI

bqStudio version 1.3.70 or higher is needed to support the BQ79606A GUI. During the initial sampling period, bqStudio is available on TI.com as “bqStudioTest”, which is version 1.3.87. Please contact your local TI representative to receive the .bqz file required to run the bq79606 GUI. The GUI is compatible with Windows 10/8/7/XP.

Use the following steps to download and install the GUI:

1. Receive the latest bqStudio (Version 1.3.87 or higher) from the web. The file “Monitor_1_03-bq79606.bqz” should be provided by your local TI representative.
2. Run the installer and follow the wizard
3. Once bqStudio is installed, locate the BatteryManagementStudio folder in the directory it was installed into (typically the C: Drive)
4. Place the Monitor_1_03-bq79606.bqz file into the config directory: (C://->ti->BatteryManagementStudio->config) as shown in Figure 1

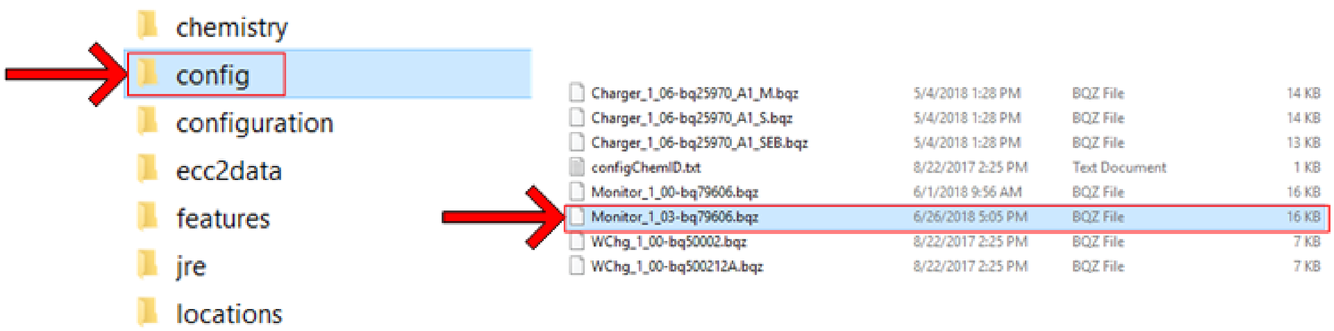

Figure 1. Placing .bqz filte into the BatteryManagementStudio/config Director

Once completed, continue to Section 1.2

1.2 Setup and Connect the BQ79606A EVM

Before connecting the bq79606EVM, please refer to the EVM user’s guide to ensure the EVM is configured correctly.

To ensure the base device is connected correctly, there are a few points of interest:

- SW1 must be set to “Base”
- A minimum of 5 V must be applied to PWR or VSTACK(Jumper J2 connects them together)
  - For this document, a resistor ladder is used to provide correct voltages at each Cell_X
- Connect the FTDI-USB serial cable from the PC to the EVM Header J8. Header J8 has 6 through-hole pads, with the square pad denoting GND and Pin 0. To properly connect the FTDI cable, the black wire denotes GND and the exposed metal on the connector should be facing “up”. An example of this cable can be seen in Figure 2.
  - FTDI Cable part number: TTL-232R-5V
  - The latest drivers can be found on the FTDI chip website here

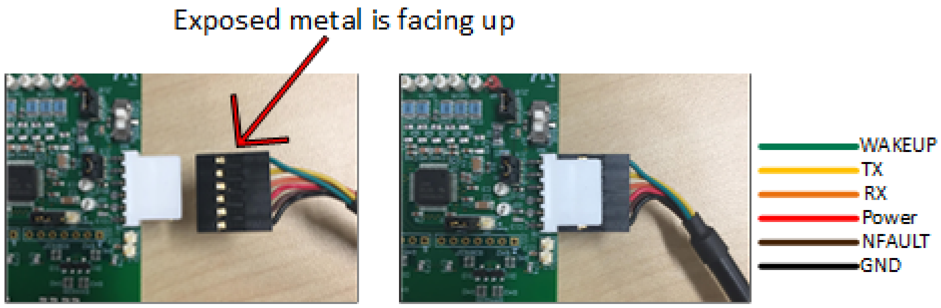

Figure 2. FTDI Cable

To ensure stack devices are connected correctly, there are a few points of interest:

- SW1 must be set to “Stack”
- The bq79606EVM should be connected to a minimum of 3 cells, with all unused VC connections shorted to the highest cell.
- Connect COMMH header J6 of the base device to COMML header J5 of the next highest stack device. Repeat for all stacked devices.

2 Communications Tab

2.1 Opening the GUI

After installing follow these next steps to open the GUI.

1. Launch bqStudio and select "Monitor" as seen in Figure 3.
   a. If a "Monitor" option is not available, check bqStudio version. It should be 1.3.70 or higher.

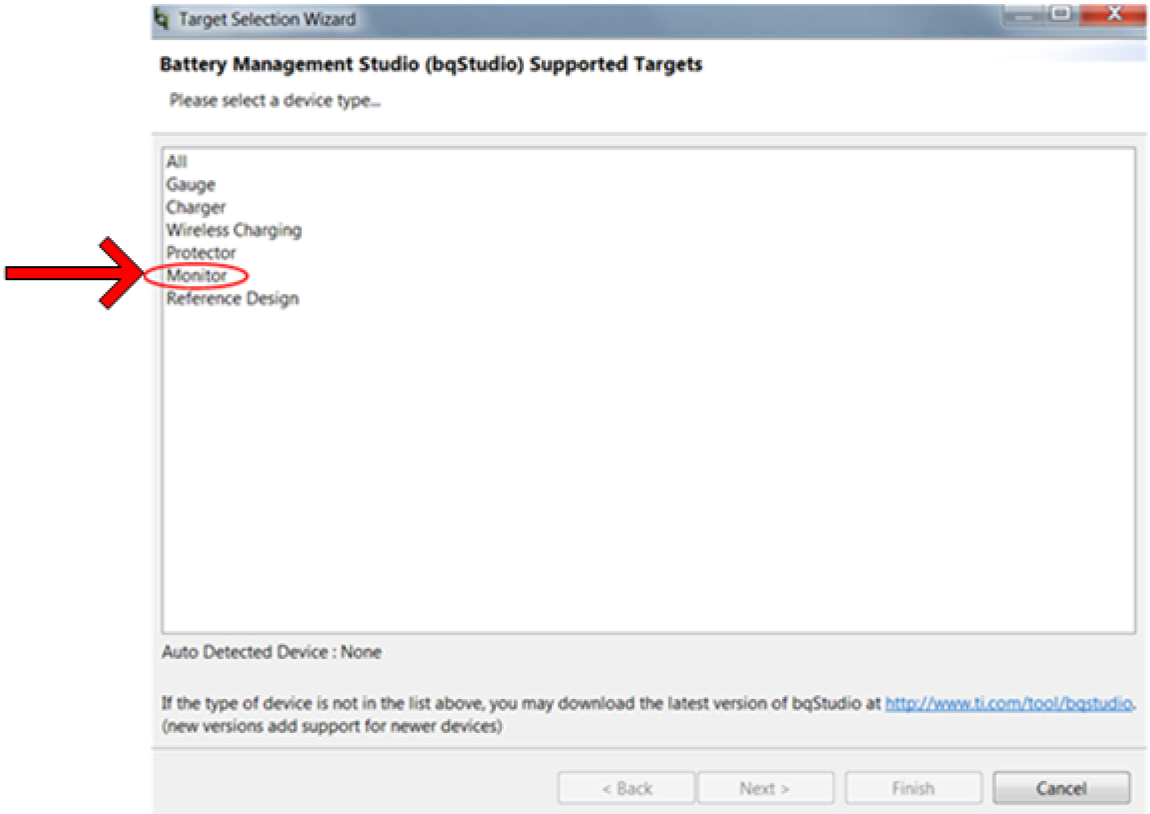

Figure 3. Selecting Monitor After Opening bqStudio

2. Select bq79606 as seen in Figure 4.

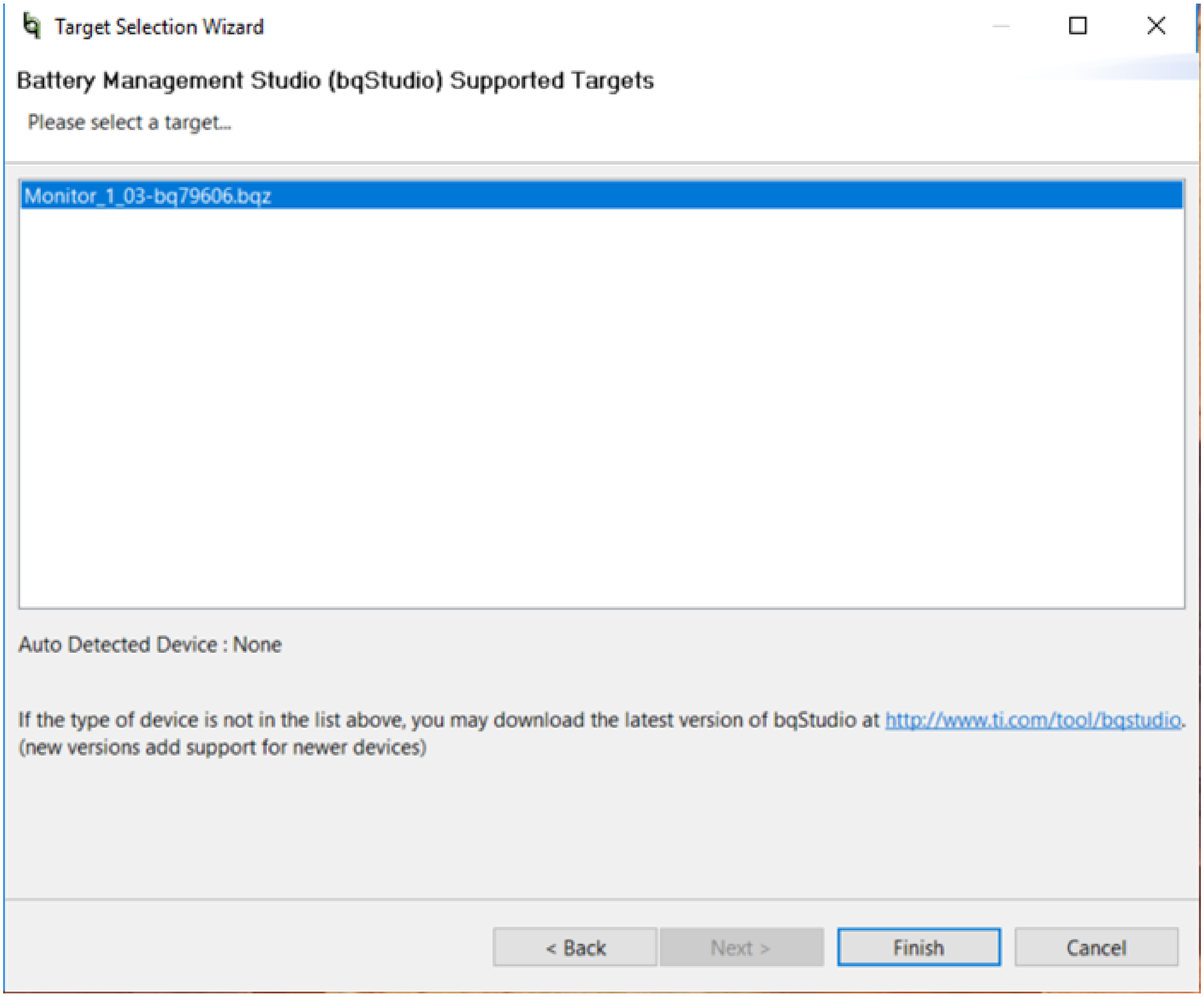

Figure 4. Selecting bq79606.bqz File

1. Reach communication tab (opening screen) shown in Figure 5. If the screen is showing this then the GUI has successfully been opened. However before running "auto addressing" there will be no data in the "Device Communication Information" section.

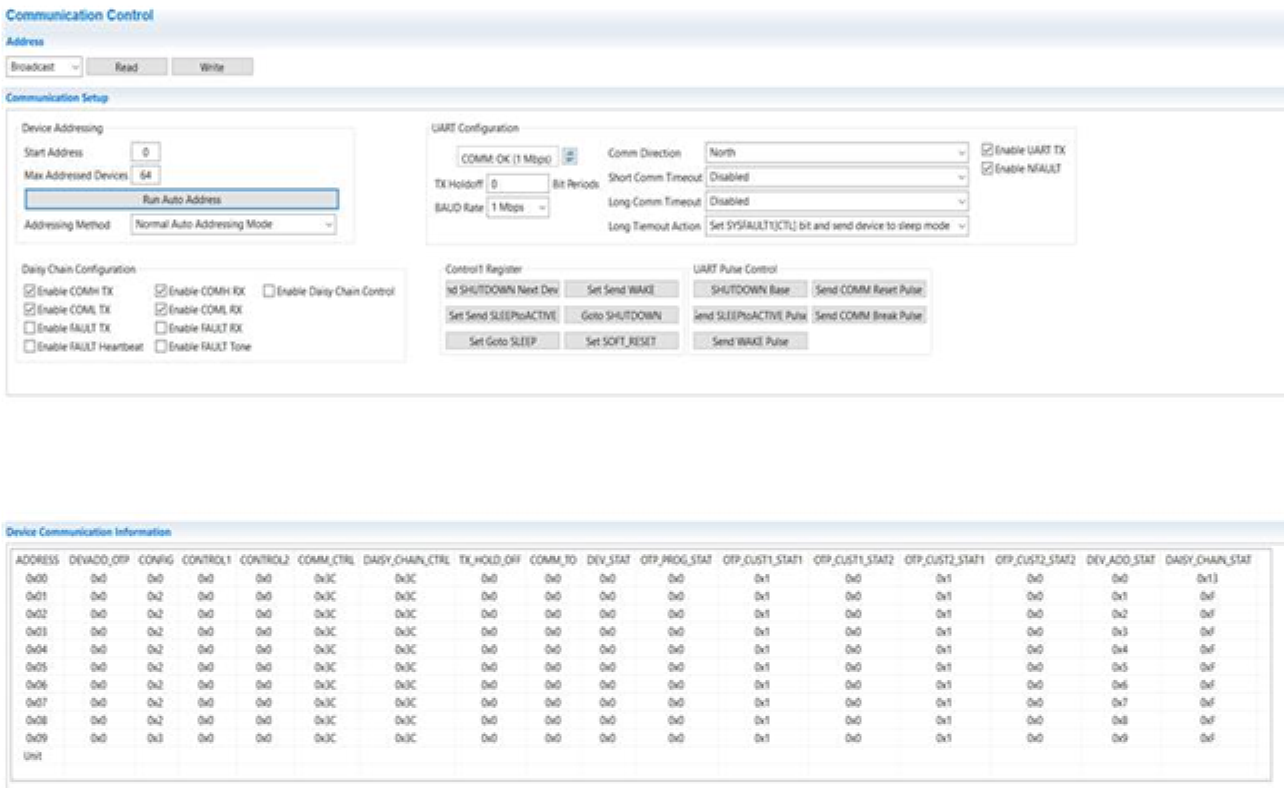

Figure 5. Communications Tab

2.2 Basics

The following section describes basic actions to communicate to the BQ79606A EVM's.

2.2.1 Address

Allows the user to select broadcast read/write, stack read/write, or select the device address for a single read/write.

Use the dropdown to choose the communication type to be used. For most cases, the default “broadcast” choice will suffice (it will work for a single device or stacked devices.

2.2.2 Device Addressing

Device addressing configures the start and end address for “auto” addressing, or configures GPIO addressing.

Before beginning any other portion of the guide, ensure that the device(s) have been correctly auto-addressed using the “Run Auto Address” button. This ensures that all devices are started properly and all future commands are sent to the appropriate devices. Figure 6 shows how to do this. After running auto addressing the "Device Communication Information" section should be filled out like Figure 5.

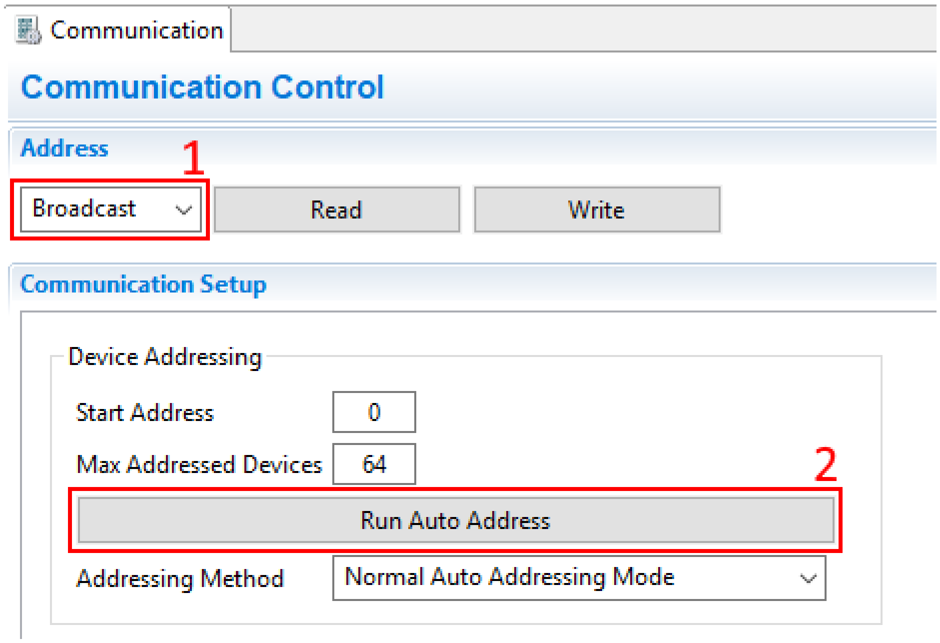

Figure 6. Auto Addressing

2.2.3 Control1 Register and UART Pulse Control

Sends commands to base or stack devices to transition to shutdown, sleep, or active mode. Also sends Comm reset/break command.

Stack Shutdown

To send a shutdown call to all devices, use the Control1 Register's "Send SHUTDOWN Next Device" command, followed by the UART Pulse Control's "SHUTDOWN Base" command.(Figure 7). This sets the shutdown command bits and then calls the shutdown pulse.

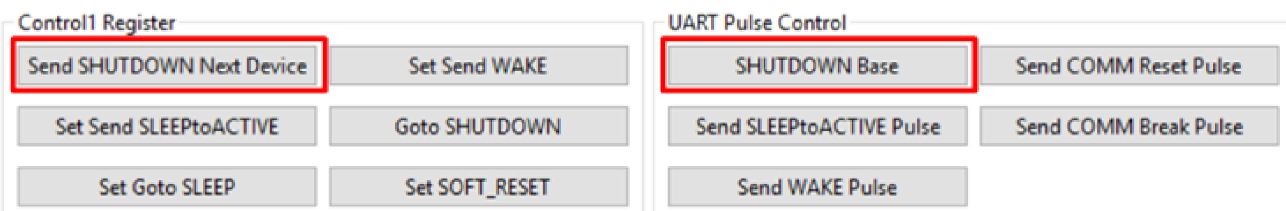

Figure 7. Stack Shutdown

Sleep to Active

If the device is left active without any commands for an extended period of time, the device will enter sleep mode by default. To wake up all devices, press the "Send SLEEPtoACTIVE Pulse" button as shown in Figure 8.

Figure 8. Sleep To Active

Soft Rest

When using the "Set SOFT_RESET" command, remember to use the "Run Auto Address" button again in order to re-establish addressing for all devices. Figure 9 shows the correct order to perform a soft reset in.

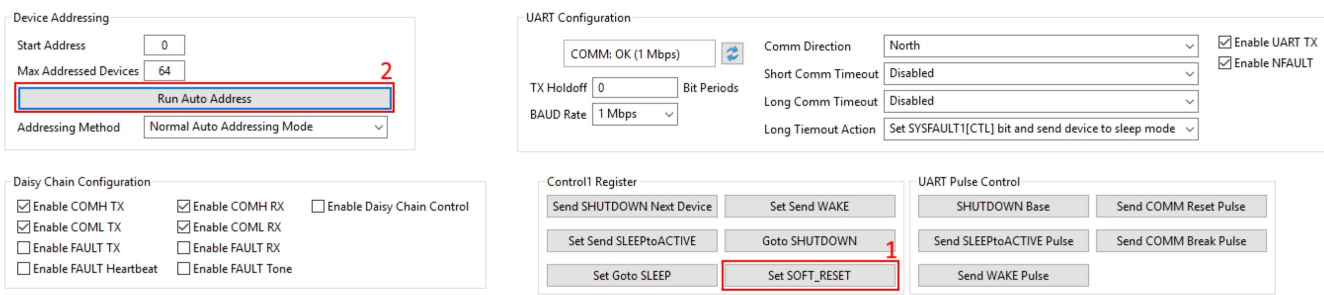

Figure 9. Soft Reset

Sleep and Exiting Sleep

When using the "Set Goto SLEEP" command, you can wake the device using the "Send SLEEPtoACTIVE Pulse." Number 1 in Figure 10 is the button to set the device into sleep mode while 2 is used for waking the devices back up.

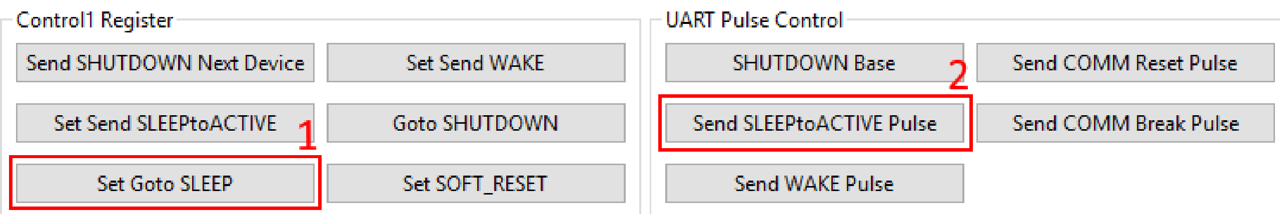

Figure 10. Sleep and Exiting Sleep

2.2.4 UART Configuration

The UART configuration displays GUI communication status (COMM OK), baud rate, and UART communication setting as shown in Figure 11.

To check the UART communications status of the device(s) at any point, the refresh button can be pressed and status read. BAUD Rate, Comm Direction, Comm Timeouts, NFAULT enable, and TX configuration are all provided in this section of the GUI.

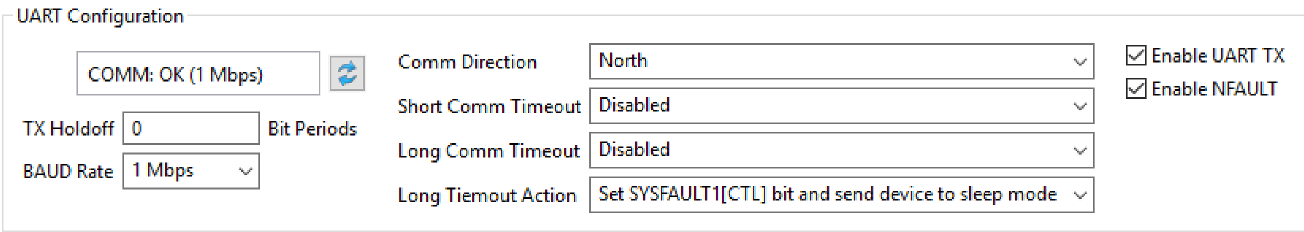

Figure 11. UART Configuration

2.2.5 Daisy Chain Configuration

Daisy chain configurations allows for the configuration of daisy chain communication settings.

Desired Daisy Chain communications lines can be enabled or disabled using the provided boxes, and written to the device with the “Write” command button. By default, COMH/L TX/RX are enabled as shown in Figure 12. Once the desired boxes are selected then press the "Write" button.

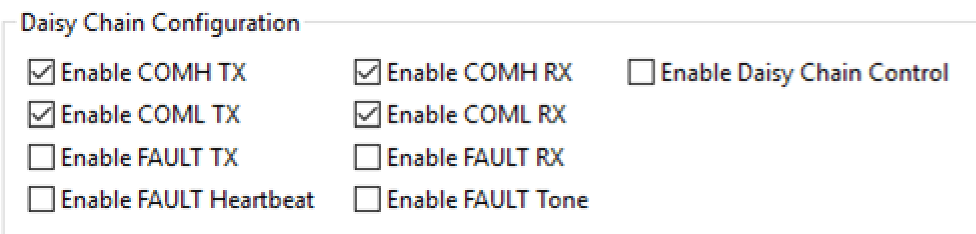

Figure 12. Daisy Chain Configuration

3 Cell Voltage Monitor Tab (ADC Config)

3.1 One Shot ADC Configuration and Readings Configuration

1. Configure CELL channels and the ADC in available fields (Figure 13)
   a. Select ADC Delay, Decimation Ratio, ADC Frequency, Cell Conversion Interval, and LP Corner Frequency
   b. Select desired cells to sample under CELL SELECTION
   c. Write to Register (Labeled 1 in Figure 13)
   d. Press START ADC MEASUREMENT (Labeled 2 in Figure 13) to begin sampling cells.

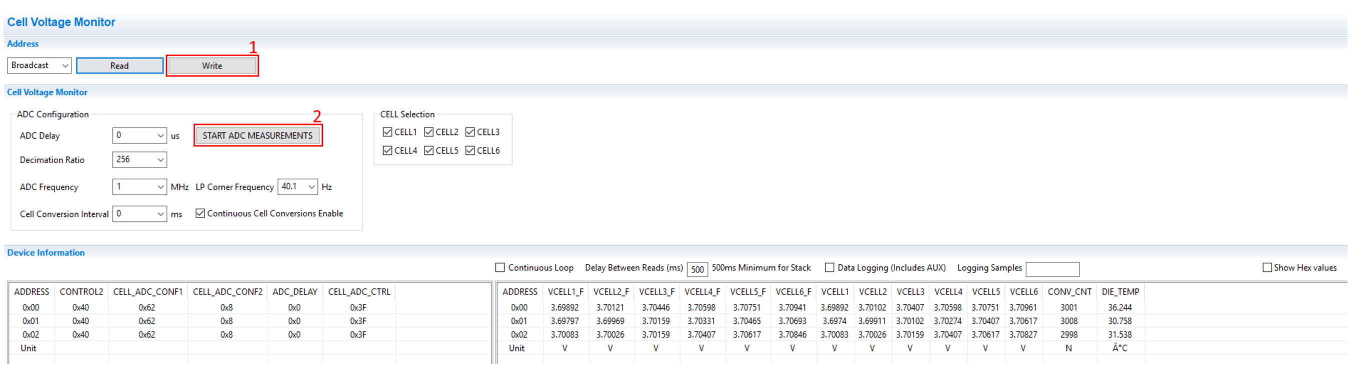

Figure 13. One-Shot ADC Measurement

2. After completing step 1, the ADC values will be stored in their respective registers and are ready to be

read (Figure 14)

a. Click “read” to update cell values.
b. Values will be read in as a voltage, or alternatively as a two bye Hex value
c. VCELLX_F contains cell values after the ADC output filter
d. VCELLX contains corrected cell values before the ADC output filter
e. To continue reading cell voltages, click “START ADC MEASUREMENT” and read the registers again to update cell voltages.

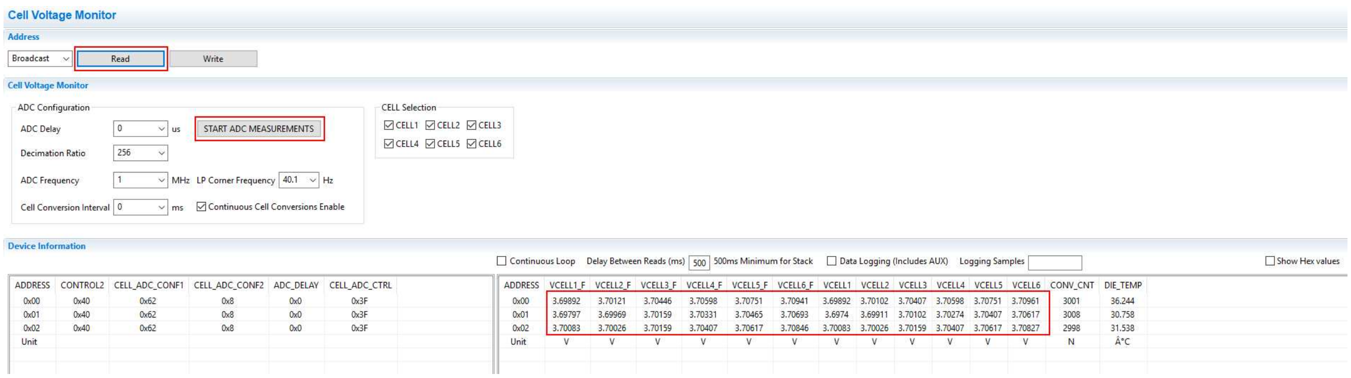

Figure 14. Updating Cell Voltage Registers

3.2 Continuous ADC Configuration and Readings Configuration

1. Configure CELL channels and the ADC in available fields
   a. Select ADC Delay, Decimation Ratio, ADC Frequency, Cell Conversion Interval, and LP Corner Frequency
   b. Select desired cells to sample under CELL SELECTION
   c. Check “Continuous Cell Conversions Enable” (Labeled 1 in Figure 15)
   d. Click “Write” (Labeled 2 in Figure 15)
   e. Click” START ADC MEASUREMENT” (Labeled 3 in Figure 15) to begin sampling cells
   f. Click “Read” to update voltage registers.
   g. Repeat step f as needed to update voltage registers

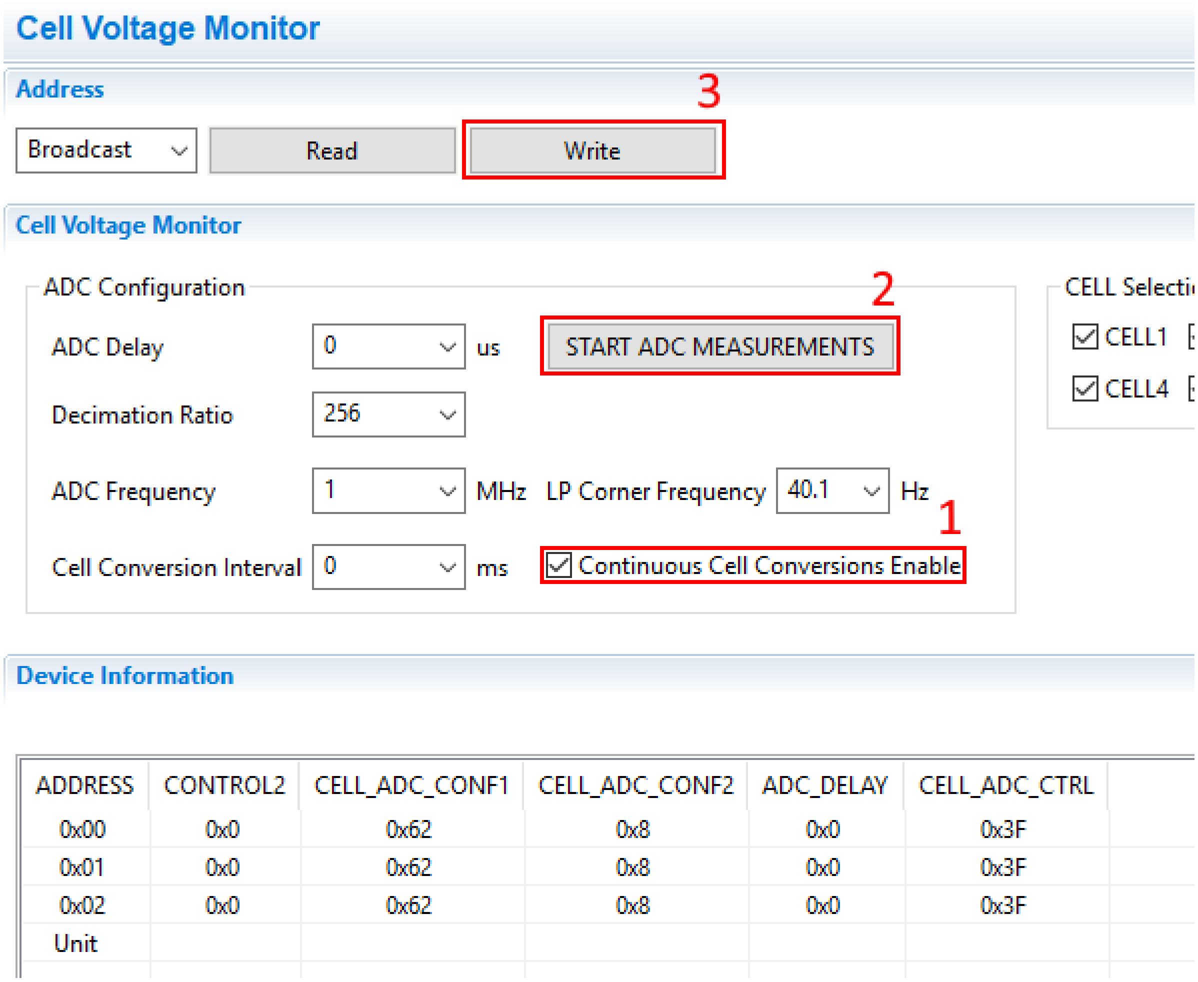

Figure 15. ADC Config for Continuous Readings

2. After completing step 1, the ADC values will be stored in their respective registers and are ready to be read.
   a. Click “read” to update cell values.
   b. Values will be read in as a voltage, or alternatively as a two bye Hex value
   c. VCELLX_F contains cell values after the ADC output filter
   d. VCELLX contains corrected cell values before the ADC output filter
   e. To continue updating cell voltages, click “Read” again to update cell voltages as seen in Figure 16.
   f. Repeat step e as needed to update cell voltage registers

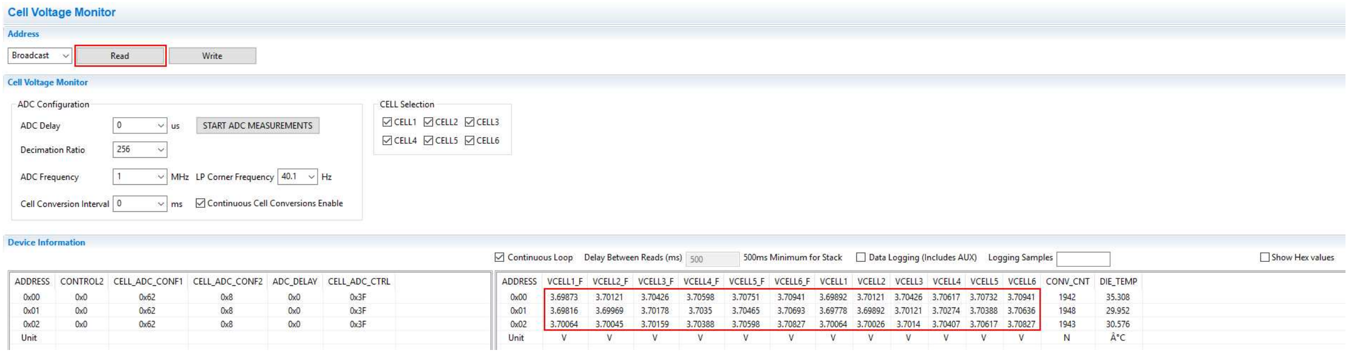

Figure 16. Polling ADC Readings in Continuous Mode

3.3 Cell Voltage Data Logging

To log cell voltage readings (includes AUX readings):

1. Setup Continuous ADC Measurements as explained above
2. Type the number of samples in the "Logging Samples" text-box in the "Device Information" section of the Cell Voltage Monitor tab
3. Click on the "Data Logging" checkbox
4. Choose the file name and location to store the ".log" file, then click "Save"

The samples will be stored in a ".log" file.

4 AUX ADC Monitor Tab

4.1 ADC Configuration

The ADC configuration configures the AUX ADC frequency and decimation ratio.

Only press the “START ADC MEASUREMENTS” button after the desired AUX Selection and Absolute/Ratiometric Selections have been chosen and written to the devices. Decimation Ratio is 256 by default, and ADC Frequency is 1 MHz by default.

4.2 AUX ADC Selection

The AUX ADC selects channels for the AUX ADC to measure.

All AUX ADC channels available for measurement are available from this selection. Select the desired AUX measurements to be made. (# 1 in Figure 17). Once selections have been made, use the “Write” command button (# 2 in Figure 17), followed by the “START ADC MEASUREMENTS” button (# 3 in Figure 17), and finally the “Read” command button(# 4 in Figure 17).

The results can then be found in the bottom right, within the “Device Information” section of the GUI. To continue updating the values in this table, press “START ADC MEASUREMENTS” followed by “Read” for each desired read.

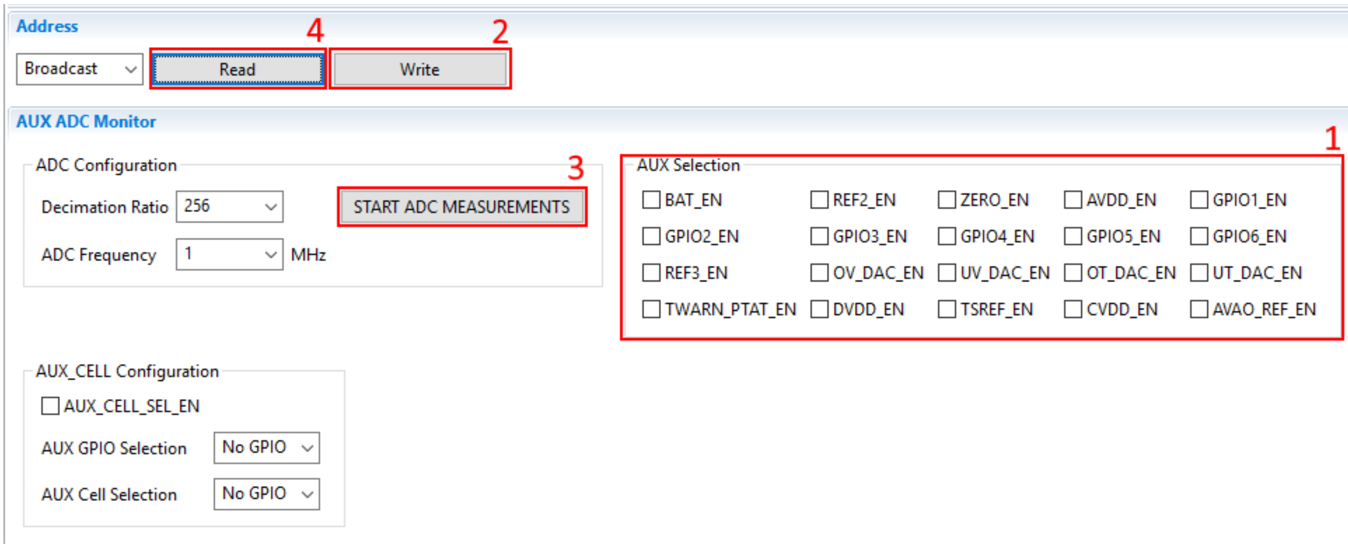

Figure 17. AUX ADC Selection

4.3 Absolute/Ratiometric Selection

Absolute and Ratiometric Selection configures GPIO ADC readings as Absolute (referenced to GND) or Ratiometric (percentage of TSREF)- generally used for thermistor readings.

To display the absolute voltage of a GPIO's specific AUX value, simply check the desired GPIO* box. Unchecked boxes display ratiometric values.

4.4 AUX_CELL Configuration

Configures the device to provide a secondary cell measurement system.

To allow for secondary measurement of a battery cell, first choose the desired "AUX Cell Selection" from the dropdown, press the "Write" command button, select the "AUX_CELL_SEL_EN" checkbox, and finally press the "Write" command button once more. This should allow for the AUX_CELL column in the "Device Information" section at the bottom of the GUI to display the chosen cell's voltage.

5 Cell Balancing

This section will discuss the basic blocks of the Cell Balancing Tab.

5.1 Balancing Configuration

To configure cell balancing, use the drop boxes to configure the duty cycle time, cell balancing timers, and balancing sequence shown in Figure 18. If desired, the cell balancing done comparator can be enabled and the threshold set.

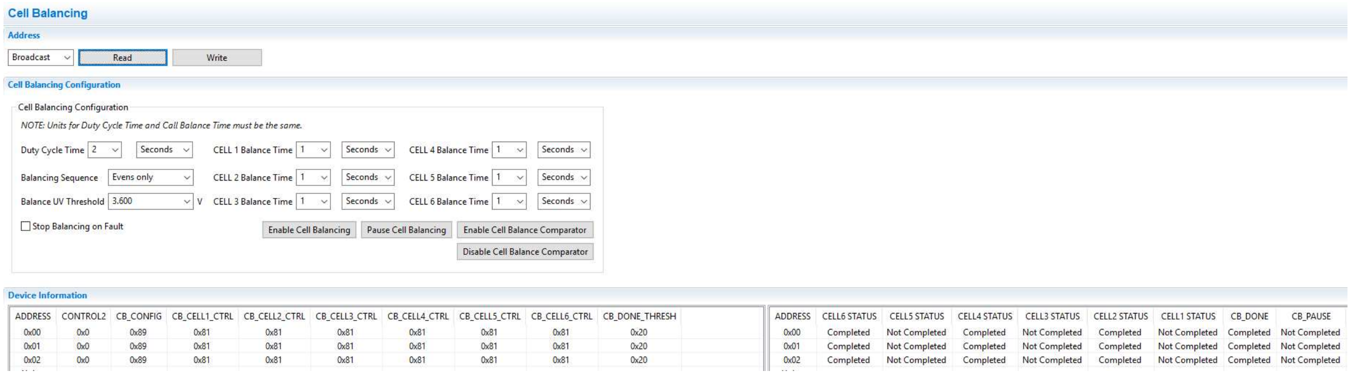

Figure 18. Cell Balancing Tab

The Cell X Balancing timers are independently configurable, and dictate how long each cell is to be balanced before CELLX Status is changed from “Not Completed” to “Completed”. The duty cycle time dictates the how often the balancing sequence logic will switch states.

A “Write” command will update the BQ79606A registers with the configured options, Following this with an “Enable Cell Balancing” command will set the CONTROL2[BAL_GO] bit, and begin the cell balancing algorithm. If desired, the part can be put into sleep mode at this point to achieve sleep mode balancing.

For example, each Cell X Balance Time can be set to a total of 50 seconds, and the Duty Cycle Time programmed to 30 seconds, with a balancing sequence of Odds then Evens. Programming the Balance UV Threshold and clicking the “Enable Cell Balance Comparator” button will set the CB_DONE_THRESH register to stop a cell from balancing once it reaches the desired UV Threshold. Once cell balancing is enabled, the BQ79606A will switch between the odd and even banks of cells every 30 seconds. For the first duty cycle period, each bank will be balanced for the full 30 seconds. On the second duty cycle period, each bank will only be balanced for the remaining 20 seconds, as the Cell X Balance Timer will expire before the Duty Cycle timer. If at any point in time an individual cell reaches the Balance UV Threshold, that particular cell will no longer balance when it’s respective bank turns on. For example, each Cell X Balance Time can be set to a total of 50 seconds, and the Duty Cycle Time programmed to 30 seconds, with a balancing sequence of Odds then Evens. Programming the Balance UV Threshold and clicking the “Enable Cell Balance Comparator” button will set the CB_DONE_THRESH register to stop a cell from balancing once it reaches the desired UV Threshold. Once cell balancing is enabled, the BQ79606A will switch between the odd and even banks of cells every 30 seconds. For the first duty cycle period, each bank will be balanced for the full 30 seconds. On the second duty cycle period, each bank will only be balanced for the remaining 20 seconds, as the Cell X Balance Timer will expire before the Duty Cycle timer. If at any point in time an individual cell reaches the Balance UV Threshold, that particular cell will no longer balance when it’s respective bank turns on.

5.2 Balancing Status

Once the cell balancing options have been configured, and cell balancing has been configured, the host can monitor the cell balancing status by reading the CELLX Status registers.

If the balancing algorithm has not been started, or is still running, the status registers will read as “Not Completed”. If a cell timer has expired, or the cell has reached the Balance UV Threshold, the register will read as “Completed", as shown in Figure 19. To monitor these registers through the GUI, the user must poll the “Read” button to manually read the registers.

ADDRESS	CELL6 STATUS	CELL5 STATUS	CELL4 STATUS	CELL3 STATUS	CELL2 STATUS	CELL1 STATUS	CB_DONE
0x00	Completed	Not Completed	Completed	Not Completed	Completed	Not Completed	Completed
0x01	Completed	Not Completed	Completed	Not Completed	Completed	Not Completed	Completed
0x02	Completed	Not Completed	Completed	Not Completed	Completed	Not Completed	Completed
Unit							

Figure 19. Cell Balancing: Completed

6 GPIO/SPI Master

6.1 GPIO Configuration and Set GPIO Output High/Low

Configures GPIO fault settings, I/O settings, pull up/pull down, or if used for addressing, sets GPIO output as high or low.

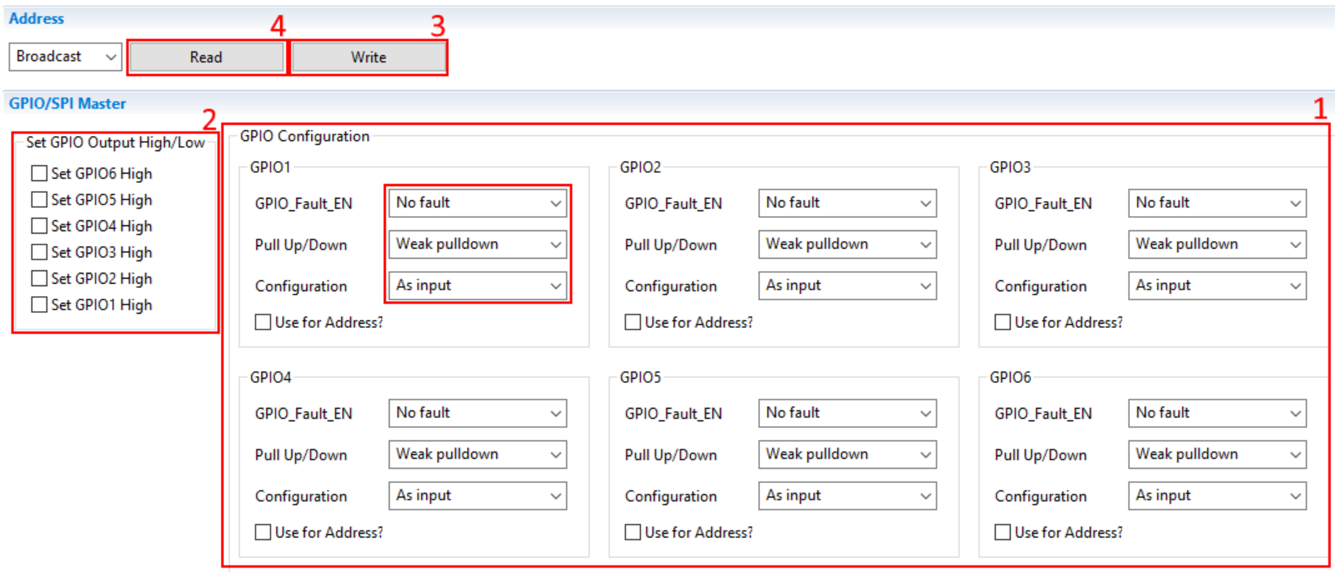

Figure 20. GPIO High/Low Testing

For GPIO output High/Low testing, first set all GPIOs to "No Fault", "Push-Pull (Output only)," and "As output"(#1 in Figure 20). This will ensure that the GPIO are ready as outputs to be set high and low by the GUI.

GPIO can now be enabled and disabled using the checkboxes in the "Set GPIO Output High/Low" section of the GUI (#2 in Figure 20). Press the "Write" command button at the top of the screen to update the devices (#3 in Figure 20). The resultant GPIO values can be viewed using the "GPIO_STAT" (0x262) GPIO Input Status section at the bottom of the GUI after the "Read" button has been pressed (#4 in Figure 20).

The enable for TSREF is on the far right side of the page. Click the box if configuring for NTC monitoring.

Note from data sheet: "Pull-downs must not be used in output mode. Additionally, push-pull mode must not be used in input mode. If either of these configurations is selected, correct operation is not guaranteed and undesirable operation may occur.

6.2 SPI Master

The SPI master will configure SPI settings for when using GPIO 3-6 as a SPI Master.

First, ensure that only one device is selected in the top left dropdown menu when using SPI Master as shown in Figure 21.

Figure 21. SPI Master Command Frame Change

To enable SPI across GPIO3-GPIO6, check the “Enable SPI” checkbox provided (#1 in Figure 22). Ensure the desired settings for CPOL, CPHA, SS, and SPI Transaction length (1 to 8 bits long) are correct. There is loopback functionality provided by the “Enable Loopback” checkbox, as well as boxes for transmitting/receiving the SPI data (#2 in Figure 22). You may send 1 byte of data over SPI by writing the desired byte in the “SPI Transmit” box, in hex (#3 in Figure 22). Then use the “Execute SPI (Single Device Only)” button to send/receive the data (#4 in Figure 22).

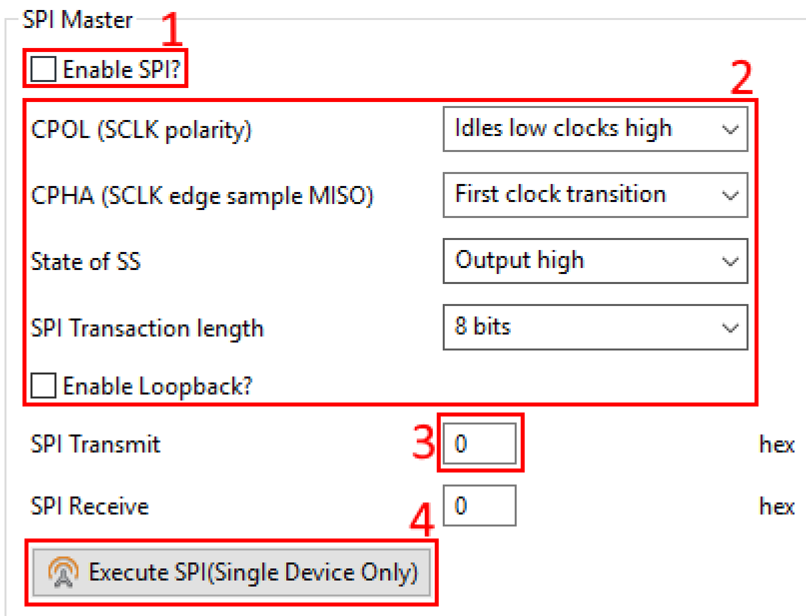

Figure 22. SPI Master Configuration

7 Registers

7.1 Bit Level Register Map

Displays a register map of all user addressable registers in the BQ79606A, as well as allowing for bit level read and writes.

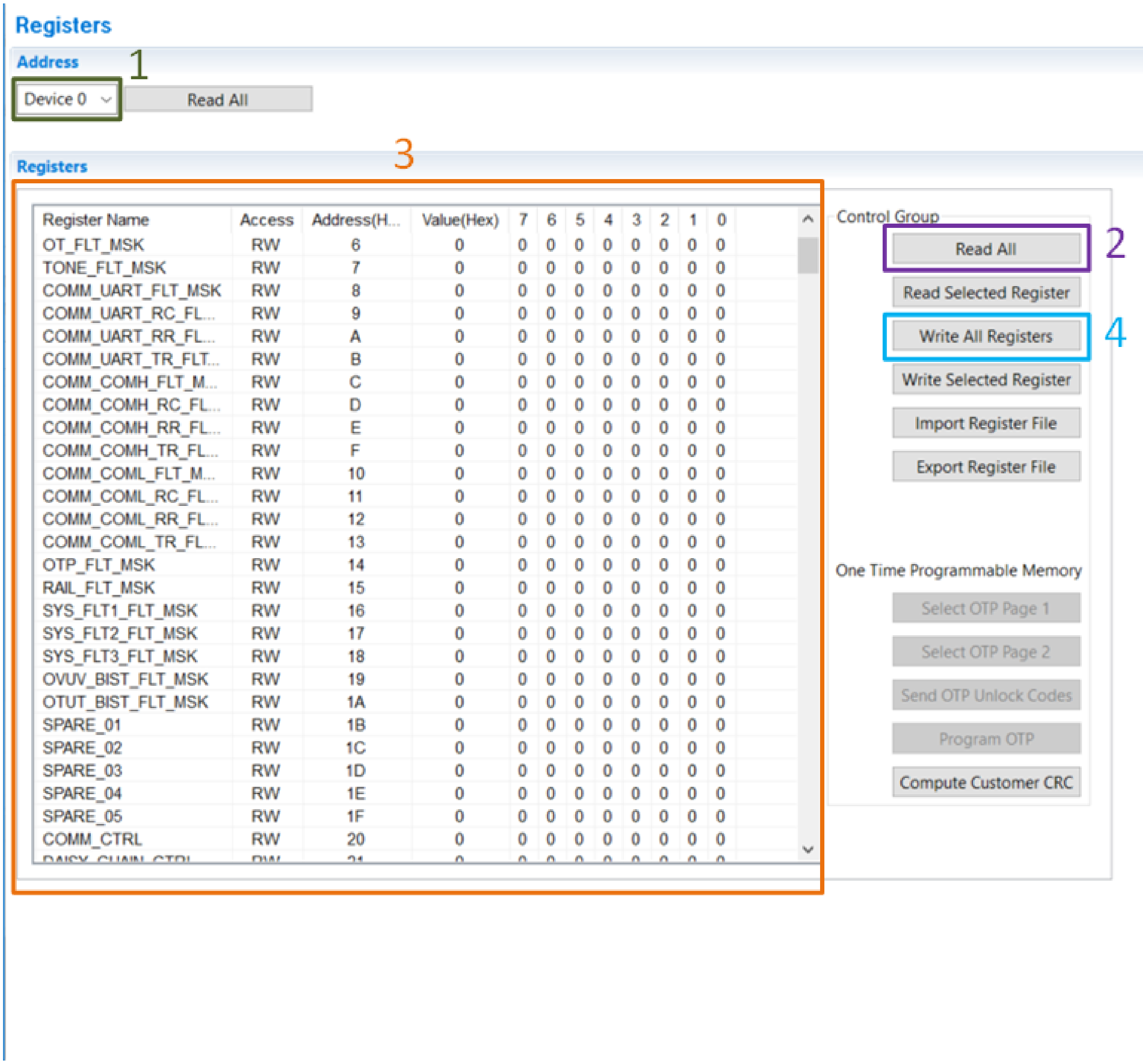

Figure 23. Registers

Access to a device's current register map is available upon choosing a device and sending a "Read All" command (#1 in Figure 23). Individual registers can be updated by clicking on a specific register within the "Registers" section, then performing a "Read Selected Register" request (#2 in Figure 23). Registers can be updated on a bit-by-bit basis by simply clicking on the desired bits to flip. Registers can also be updated by entering specific hex values in the "Value(Hex)" column (#3 in Figure 23). Once the desired registers are modified, the device can be updated by using the "Write All Registers" button, or for an individual register change, by using the "Write Selected Register" (#4 in Figure 23).

The entire register map can be saved by simply exporting the current register file using the "Export Register File" button. This register map can then be applied and used on other devices by first using "Import Register File" while using the GUI on that device, then using "Write All Registers" to update that device.

8 Protectors

The protector tab allows for manipulation of a separate set of voltage and temperature thresholds, which act independent of the main ADC function. This allows for redundancy in the case of an ADC failure.

8.1 Protector Configuration and OV/UV/OT/UT Selection

Configures OT/UT/OV/UV thresholds, modes, and deglitch times.

Instructions to properly configure the Protector Tab:

1. Enable TSREF
   a. Go to "Advanced Comm" tab
   b. Choose "Broadcast Write" for "Command"
   c. Write "106" for "Register Address (Hex)"
   d. Write "10" for "Bytes to Write (Hex)"
   e. Press "Send Command"

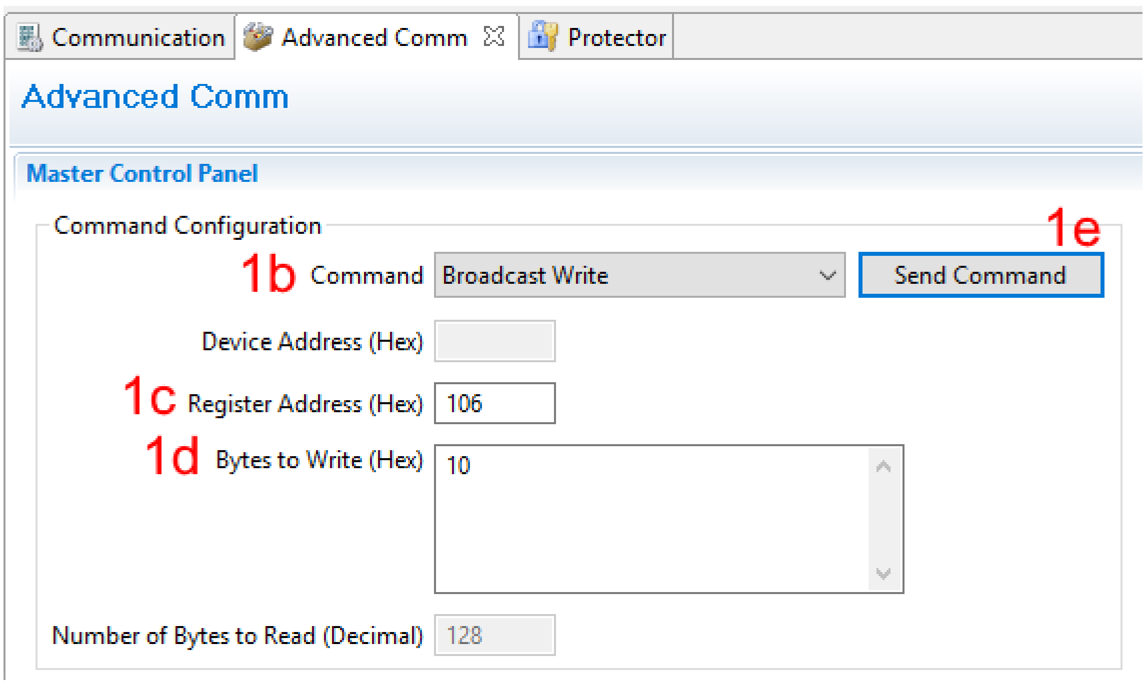

Figure 24. Enable TSREF

2. Disable OVUV and OTUT (if they are enabled) - return to "Protector" tab:
   a. Use the “Read” button (a in Figure 25)
   b. Uncheck “Enable OVUV” and “Enable OTUT” if they are enabled (b in Figure 25)
   c. Use the “Write” button (c in Figure 25)

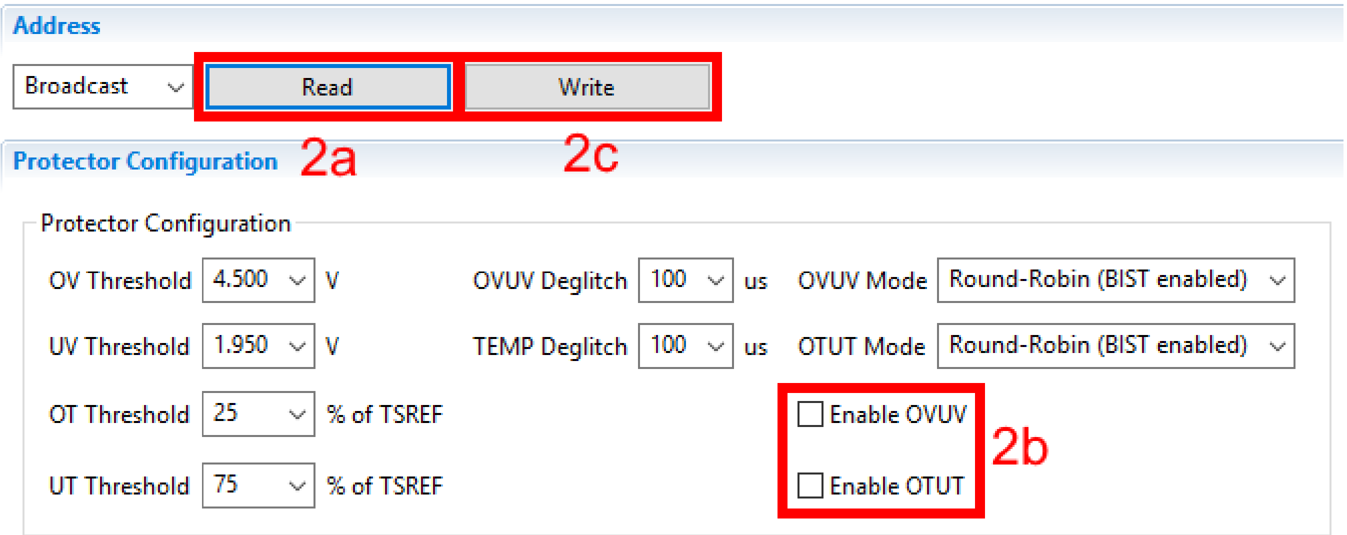

Figure 25. Disable OVUV and OTUT

3. Configuration
   a. Set the OV, UV, OT, and UT Threshold values from the dropdown menu. Also set the OVUV and TEMP Deglitch duration times, as well as the OVUV and OTUT Modes (#2 in Figure 26)
   b. Set the OVUV CELL and OTUT GPIO Selection boxes (#3 in Figure 26)
   c. Write the configuration to the device using the “Write” button at the top (DO NOT check “Enable OVUV” or “Enable OTUT” until you have written your configuration to the device(s) (#3 in Figure 26)

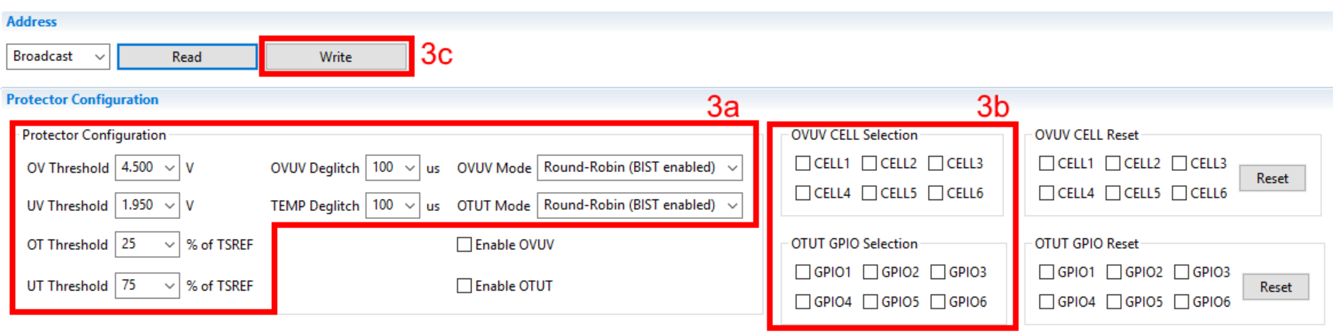

Figure 26. Configuration of OVUV/OTUT

4. Enable OVUV/OTUT
   a. Check “Enable OVUV” and “Enable OTUT” (#5 in Figure 27)
   b. Use the “Write” button again (#6 in Figure 27)

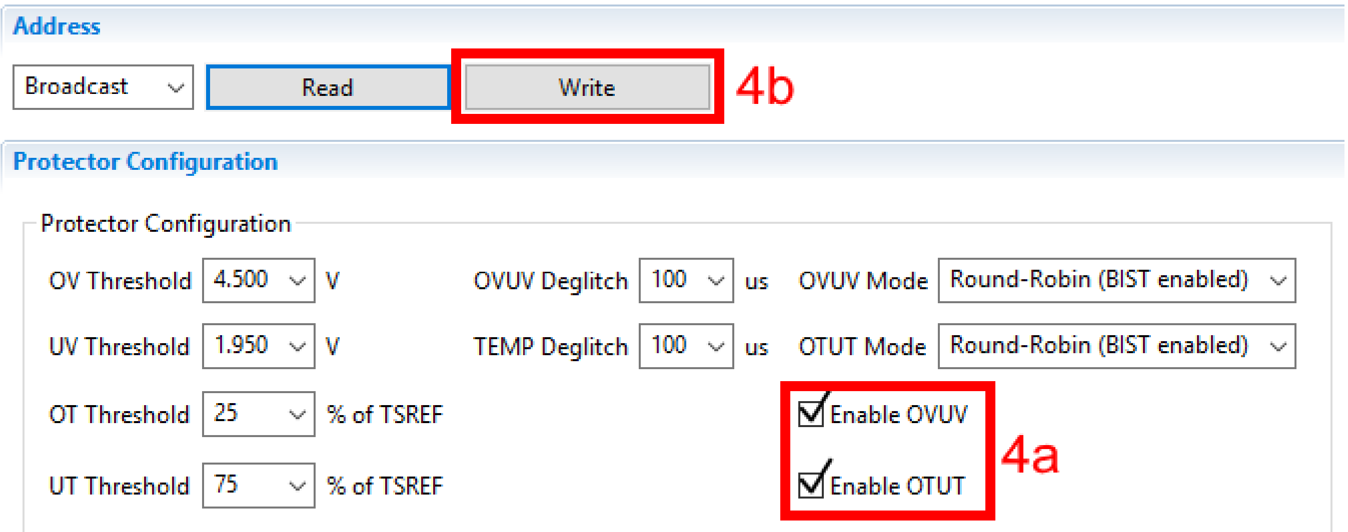

Figure 27. Enable OVUV and OTUT

The device should now be able to detect if there is an over/under threshold event. You may view these over/under threshold events in the "Device Information" section of the GUI by pressing "Read" manually.

The OV threshold allows values from 2 V to 5 V, and the UV threshold allows values from 0.7 V to 3.875 V, both in steps of 25 mV.

8.2 OV/UV/ and OT/UT Reset

Selects channels to reset faults.

To reset OVUV/OTUT faults:

1. Disable OVUV and OTUT (if they are enabled):
   a. Use the "Read" button (a in Figure 25)
   b. Uncheck "Enable OVUV" and "Enable OTUT" if they are enabled (b in Figure 25)
   c. Use the "Write" button (c in Figure 25)
2. Check the OVUV Cells and OTUT GPIOs you would like to reset under the "OVUV CELL Reset" and "OTUT GPIO Reset" sections respectively (#2 in Figure 28)
3. Use the "Write" button again (#3 in Figure 28)
4. Press "Reset" under both "OVUV CELL Reset" and "OTUT GPIO Reset" (#4 in Figure 28)
5. Use the "Read" button to ensure that the desired CELL and GPIO faults have been reset.

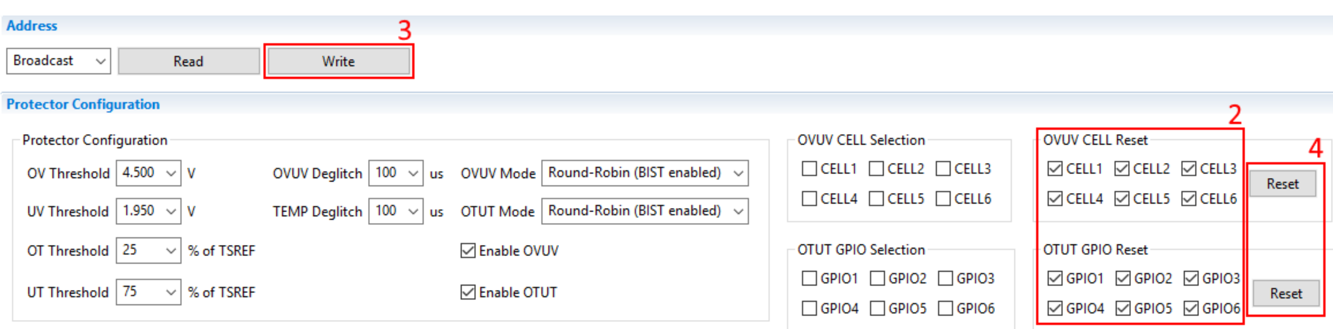

Figure 28. OVUV and OTUT Reset

9 Faults

A “Read” displays a list of device addresses, and faults associated with each device which can be seen in Figure 29.

Once “Read” is pressed, all devices are shown on the left side of the GUI, along with an overview of the major faults on each device. Pressing “Show All” next to a device provides all faults for that device, in detail, on the right side of the GUI. Here, each fault register address and value is provided. Each fault can be masked, unmasked, or reset on the right side of the GUI by pressing the corresponding checkbox, the using the “Write All” command at the top of the GUI.

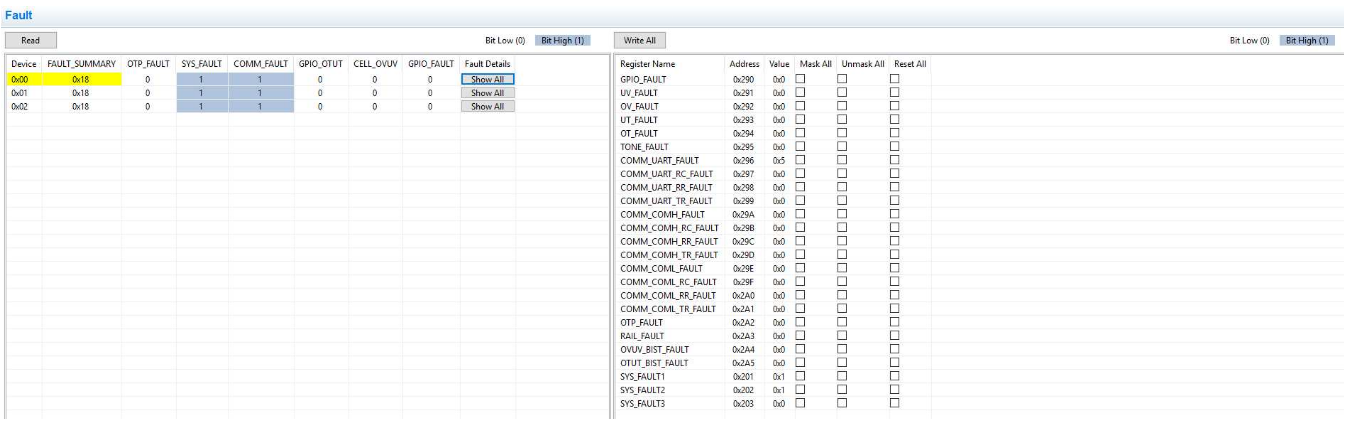

Figure 29. Fault Summary

10 Status

A “Read” displays a list of device addresses, and the status register for each device as shown in Figure 30.

The Status tab provides an in-depth look at a device’s registers of importance (such as several of the communications status registers). Select a single device from the dropdown and then use the “Read” command button. The address of each register, hex values, and bit values of each register are provided here for quick viewing of the device’s status.

Status

Select Device: Device 0x00 | Read

Register Name	Address	Value	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
DEV_STAT	0x204	0xC5	CRC_DONE	CB_DONE	CB_PAUSE	CB_RUN	AUX_STAT	CELL_STAT	DRDY_AUX	DRDY_CELL
LOOP_STAT	0x205	0x0	RSVD[3]	RSVD[2]	RSVD[1]	RSVD[0]	OTUT_BIST_DONE	OVUV_BIST_DONE	OTUT_LOOP_DONE	OVUV_LOOP_DONE
CBVC_COMP_STAT	0x263	0x0	RSVD[1]	RSVD[0]	CELL6	CELL5	CELL4	CELL3	CELL2	CELL1
CBVC_VCLOW_STAT	0x264	0x0	RSVD[1]	RSVD[0]	CELL6	CELL5	CELL4	CELL3	CELL2	CELL1
COMM_UART_RC_STAT3	0x265	0x0	DISCARD[7]	DISCARD[6]	DISCARD[5]	DISCARD[4]	DISCARD[3]	DISCARD[2]	DISCARD[1]	DISCARD[0]
COMM_COML_RC_STAT3	0x266	0x0	DISCARD[7]	DISCARD[6]	DISCARD[5]	DISCARD[4]	DISCARD[3]	DISCARD[2]	DISCARD[1]	DISCARD[0]
COMM_COMH_RR_STAT3	0x267	0x0	DISCARD[7]	DISCARD[6]	DISCARD[5]	DISCARD[4]	DISCARD[3]	DISCARD[2]	DISCARD[1]	DISCARD[0]
COMM_COML_RR_STAT3	0x268	0x0	DISCARD[7]	DISCARD[6]	DISCARD[5]	DISCARD[4]	DISCARD[3]	DISCARD[2]	DISCARD[1]	DISCARD[0]
COMM_COMH_RC_STAT3	0x269	0x0	DISCARD[7]	DISCARD[6]	DISCARD[5]	DISCARD[4]	DISCARD[3]	DISCARD[2]	DISCARD[1]	DISCARD[0]
COMM_UART_RR_STAT3	0x26A	0x0	DISCARD[7]	DISCARD[6]	DISCARD[5]	DISCARD[4]	DISCARD[3]	DISCARD[2]	DISCARD[1]	DISCARD[0]
COMM_UART_RC_STAT1	0x26B	0x0	VALIDL[7]	VALIDL[6]	VALIDL[5]	VALIDL[4]	VALIDL[3]	VALIDL[2]	VALIDL[1]	VALIDL[0]
COMM_UART_RC_STAT2	0x26C	0x2	VALIDL[7]	VALIDL[6]	VALIDL[5]	VALIDL[4]	VALIDL[3]	VALIDL[2]	VALIDL[1]	VALIDL[0]
COMM_COML_RC_STAT1	0x26D	0x0	VALIDL[7]	VALIDL[6]	VALIDL[5]	VALIDL[4]	VALIDL[3]	VALIDL[2]	VALIDL[1]	VALIDL[0]
COMM_COML_RC_STAT2	0x26E	0x0	VALIDL[7]	VALIDL[6]	VALIDL[5]	VALIDL[4]	VALIDL[3]	VALIDL[2]	VALIDL[1]	VALIDL[0]
COMM_COMH_RR_STAT1	0x26F	0x0	VALIDL[7]	VALIDL[6]	VALIDL[5]	VALIDL[4]	VALIDL[3]	VALIDL[2]	VALIDL[1]	VALIDL[0]
COMM_COMH_RR_STAT2	0x270	0x0	VALIDL[7]	VALIDL[6]	VALIDL[5]	VALIDL[4]	VALIDL[3]	VALIDL[2]	VALIDL[1]	VALIDL[0]
COMM_UART_TR_STAT1	0x271	0x0	SENTL[7]	SENTL[6]	SENTL[5]	SENTL[4]	SENTL[3]	SENTL[2]	SENTL[1]	SENTL[0]
COMM_UART_TR_STAT2	0x272	0x2	SENTL[7]	SENTL[6]	SENTL[5]	SENTL[4]	SENTL[3]	SENTL[2]	SENTL[1]	SENTL[0]
COMM_COML_TR_STAT1	0x273	0x0	SENTL[7]	SENTL[6]	SENTL[5]	SENTL[4]	SENTL[3]	SENTL[2]	SENTL[1]	SENTL[0]
COMM_COML_TR_STAT2	0x274	0x0	SENTL[7]	SENTL[6]	SENTL[5]	SENTL[4]	SENTL[3]	SENTL[2]	SENTL[1]	SENTL[0]
COMM_COMH_RC_STAT1	0x275	0x0	VALIDL[7]	VALIDL[6]	VALIDL[5]	VALIDL[4]	VALIDL[3]	VALIDL[2]	VALIDL[1]	VALIDL[0]
COMM_COMH_RC_STAT2	0x276	0x0	VALIDL[7]	VALIDL[6]	VALIDL[5]	VALIDL[4]	VALIDL[3]	VALIDL[2]	VALIDL[1]	VALIDL[0]
COMM_COML_RR_STAT1	0x277	0x0	VALIDL[7]	VALIDL[6]	VALIDL[5]	VALIDL[4]	VALIDL[3]	VALIDL[2]	VALIDL[1]	VALIDL[0]
COMM_COML_RR_STAT2	0x278	0x0	VALIDL[7]	VALIDL[6]	VALIDL[5]	VALIDL[4]	VALIDL[3]	VALIDL[2]	VALIDL[1]	VALIDL[0]
COMM_COMH_TR_STAT1	0x279	0x0	SENTL[7]	SENTL[6]	SENTL[5]	SENTL[4]	SENTL[3]	SENTL[2]	SENTL[1]	SENTL[0]
COMM_COMH_TR_STAT2	0x27A	0x0	SENTL[7]	SENTL[6]	SENTL[5]	SENTL[4]	SENTL[3]	SENTL[2]	SENTL[1]	SENTL[0]
COMM_UART_RR_STAT1	0x27B	0x0	VALIDL[7]	VALIDL[6]	VALIDL[5]	VALIDL[4]	VALIDL[3]	VALIDL[2]	VALIDL[1]	VALIDL[0]
COMM_UART_RR_STAT2	0x27C	0x0	VALIDL[7]	VALIDL[6]	VALIDL[5]	VALIDL[4]	VALIDL[3]	VALIDL[2]	VALIDL[1]	VALIDL[0]
OTP_PROG_STAT	0x27D	0x0	RSVD	UNLOCK	UVERR	OVERR	SUVERR	SOVERR	PROGERR	DONE
OTP_CUST1_STAT1	0x27E	0x1	RSVD	LOADED	LOADWRN	LOADERR	FMTERR	PROGOK	RETRY	FREE
OTP_CUST1_STAT2	0x27F	0x0	RSVD[1]	RSVD[0]	TRY2	UV2OK	OV2OK	TRY1	UV1OK	OV1OK
OTP_CUST2_STAT1	0x280	0x1	RSVD	LOADED	LOADWRN	LOADERR	FMTERR	PROGOK	RETRY	FREE
OTP_CUST2_STAT2	0x281	0x0	RSVD[1]	RSVD[0]	TRY2	UV2OK	OV2OK	TRY1	UV1OK	OV1OK
CB_SW_STAT	0x282	0x0	RSVD[1]	RSVD[0]	CELL6	CELL5	CELL4	CELL3	CELL2	CELL1
DEV_ADD_STAT	0x2BB	0x0	RSVD[1]	RSVD[0]	ADD[5]	ADD[4]	ADD[3]	ADD[2]	ADD[1]	ADD[0]
COMM_STAT	0x2BC	0x3	RSVD[3]	RSVD[2]	RSVD[1]	RSVD[0]	COMH_TONEBUSY	COML_TONEBUSY	BAUD_STAT[1]	BAUD_STAT[0]
DAISY_CHAIN_STAT	0x2BD	0x13	RSVD[2]	RSVD[1]	RSVD[0]	HW_DRV	COMLTX	COMLRX	COMHTX	COMHRX

Figure 30. Status Registers

11 Command Sequence and Advanced Comms

11.1 Advance Comm Tab

Master Control Panel

Allows the user to send raw hex commands to the base device.

Read Command

Reads can be up to 128 bytes, and called as a single device read, broadcast read, or stack read. A range of registers can be read by simply choosing the starting register address, the number of bytes to read, and sending the command. Follow the order found in Figure 31.

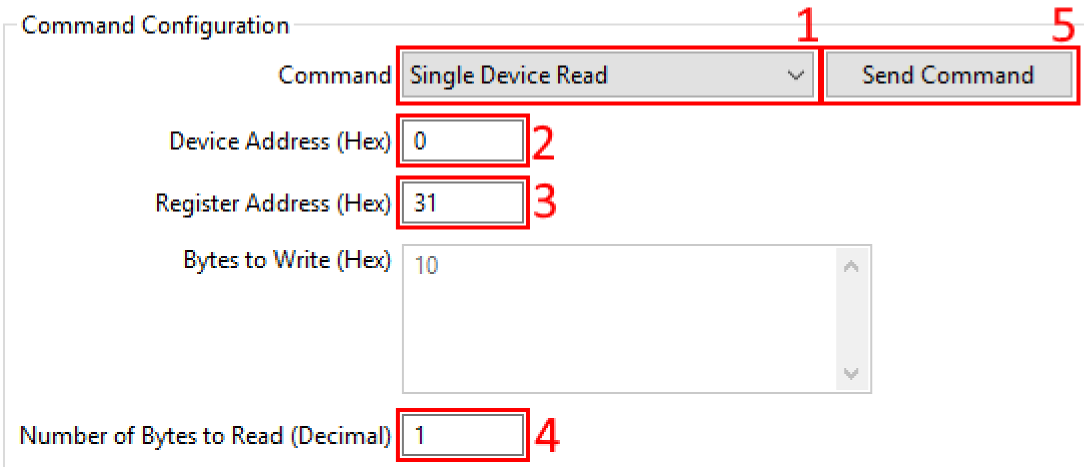

Figure 31. Read Command

Write Command

Writes can be up to 128 bytes, and called as a single device write, broadcast write, stack write, or reverse direction broadcast write.

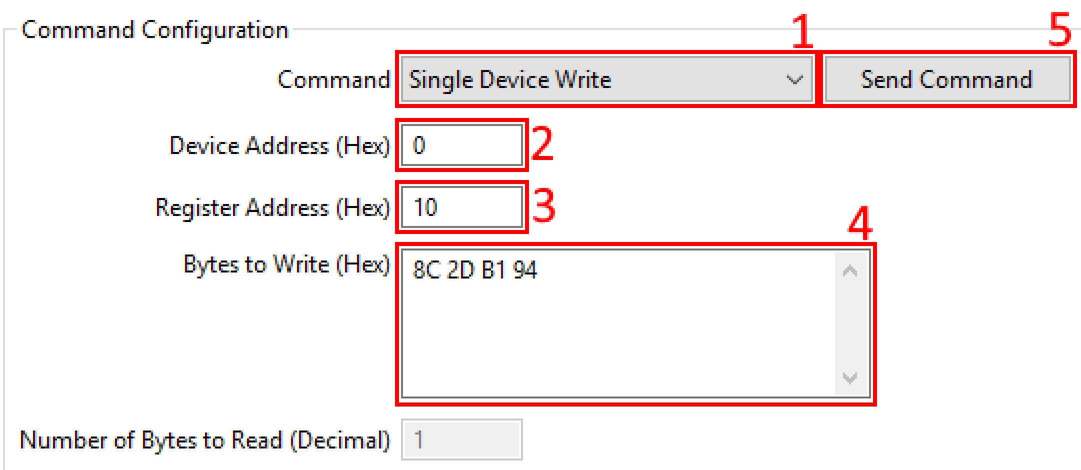

Figure 32. Write Command

A range of registers can be written to by simply choosing the starting register address, then writing enough bytes in the "Bytes to Write" section to fill the desired range, as seen in Figure 32 or Figure 33. For instance, if you choose "Register Address" 10E, then provide "82828282" in the "Bytes to Write" section, registers 10E to 111 (4 registers total) will be overwritten with "82".

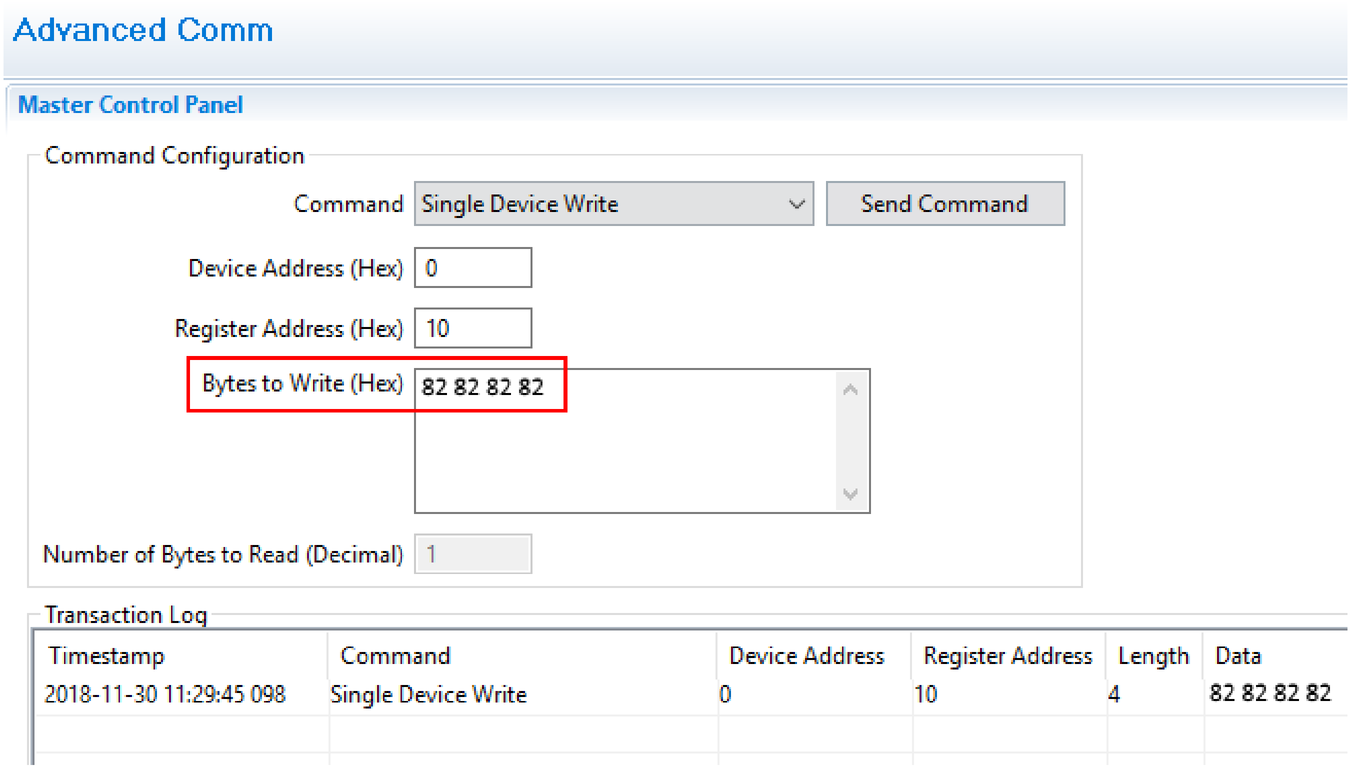

Figure 33. Write Command - With Transaction Log

11.2 Command Sequence Tab

Command Frame

Allows the user to send raw hex commands to the base device, as a host MCU would.

The Command Sequence tab works in the same manner as the Advanced Comm tab, by sending commands in hexadecimal form to the devices. However, the Command Sequence tab takes this functionality a step further by allowing for sequences of commands to be executed, saved, and executed again at a later time.

To begin generating command sequences, start by choosing a command type from the dropdown menu (#1 in Figure 34), typing the number of bytes to read/write, and press the “Generate Command Frame Header”. If a write command has been chosen, add the desired hex values to the end of the generated “command frame header” (#2 in Figure 34). Ensure the number of bytes added to the end of a write command matches the “Number of Bytes to Send” (#3 in Figure 34). Finally, send the command to the device(s) by pressing the “Send Command Frame to stack” button (#4 in Figure 34). This will also add the command to a stack of commands within the GUI. This stack of commands is located under the “Command Dialog” section. Repeat this procedure for as many commands as desired.

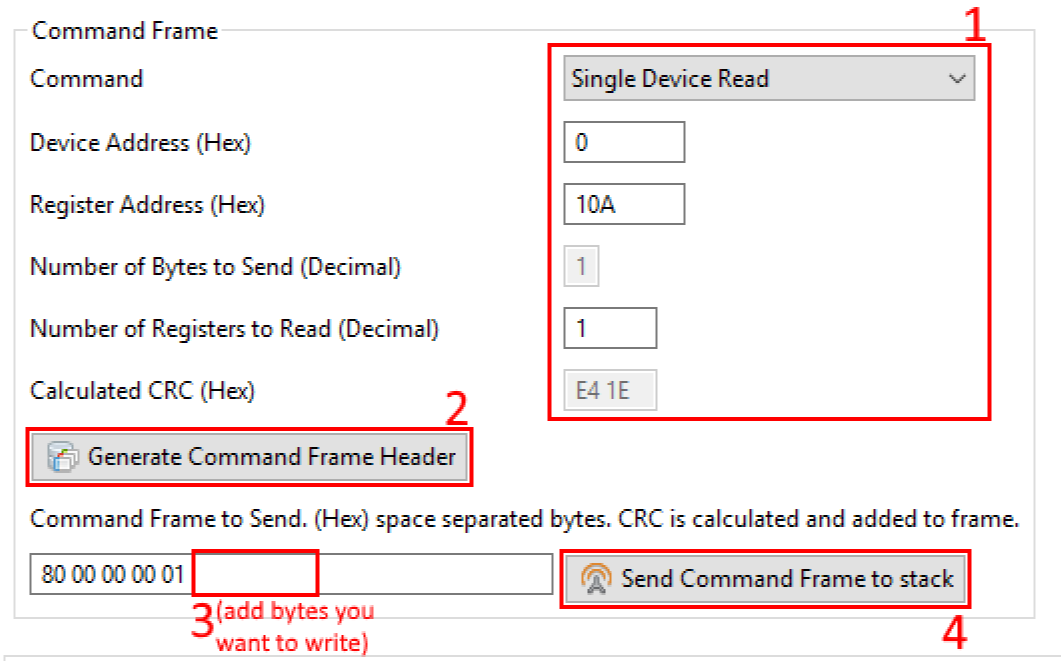

Figure 34. Generating Command Frame

Once the command dialog contains all of the desired commands, save the command sequence to a text file using the “Save” button to the right of the “Command Dialog” section. This file can later be loaded so that the sequence can be quickly run again.

Sequence File Assignment Buttons

Allows a script file containing several raw hex commands to be assigned to specific GUI button.

Saved command sequences can also be stored within the GUI as a button under the “Command Frame Sequence File Assignment Buttons” section. This allows complex user-created command sequences to be loaded with a single button press.

Once a user has generated a command sequence file (as explained in the above section), this file can be saved to one of the 8 buttons provided. Simply press an “Unassigned” button, and the GUI will ask for a command sequence file and name for the button. Fill these in and the button will be updated with the newly added command sequence and name.

Left-clicking the button will now load the command sequence of the saved file to the “Command Dialog”. Simply press “Run” to run the file’s command sequence.

Users can also delete saved buttons by right-clicking the undesired button and pressing “Clear”.

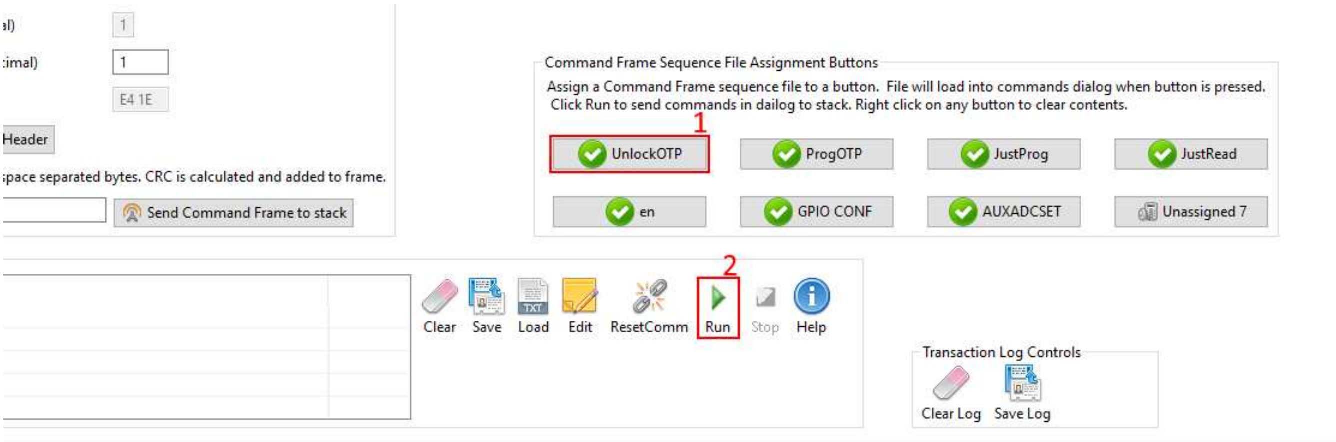

Figure 35. Sequence File Assignment

12 Troubleshooting

12.1 Troubleshooting for GUI communications

1. Remove all power from device
   a. Cell Power/ Resistor Ladder Power
   b. USB-Serial cable power
2. Unplug USB-Serial cable from PC
3. Close bqStudio
4. Apply power to device and reconnect cables
5. Open bqStudio

12.2 General troubleshooting for device recognition

1. Ensure all jumpers are correctly placed as outlined in the EVM User's Guide. This includes ensuring jumpers J2 and J3 are connected if using the battery stack/resistor ladder to supply power for the EVM. Also ensure that SW1 is set properly to base or stack.
2. Ensure that the device lower in the stack is connected using its "High Side Comm" connector to the higher device in the stack's "Low Side Comm."
3. Run auto-addressing and see if all devices show up in the top left dropdown menu. If a device does not, the device may not be configured properly as described above and in the EVM User's Guide.

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