HDQ Communication Basics for TI's Battery Monitor ICs

Battery Management

Introduction

Most battery monitor ICs from TI, including the bq2018 and bq2019, includes a single-wire HDQ serial data interface. Host controllers, configured for either polled or interrupt processing, use the interface to access various IC registers. The purpose of this application note is to cover most of the communication related topics for these devices.

HDQ BASICS

The HDQ pin requires an external pullup or pulldown resistor. The interface uses a command-based protocol, where the host processor sends a command byte to the battery monitor. The command directs the battery monitor either to store the next eight bits of data received to a register specified by the command byte or to output the eight bits of data from a register specified by the command byte. The communication protocol is asynchronous return-to-one and is referenced to Vss, which is typically tied to the battery pack's negative terminal.

There is no need to update register values in the battery monitor each time a run time computation is made. Local variables in the host processor can be kept up to date with pertinent data so that minimal communication is required for most computations. Update of registers in the battery monitor can be done at events like full or empty, after some predefined interval, when a self-discharge estimation is performed, or prior to a power down operation. The host may find that it only needs to read the CCR and DCR registers at frequent intervals (30 seconds, perhaps) to update the user with the latest runtime computation.

A GPIO or UART (on the host) can be used to communicate with the battery monitors. A microcode example is provided at the end of this application note.

Servicing High Priority Interrupts while Receiving HDQ Data

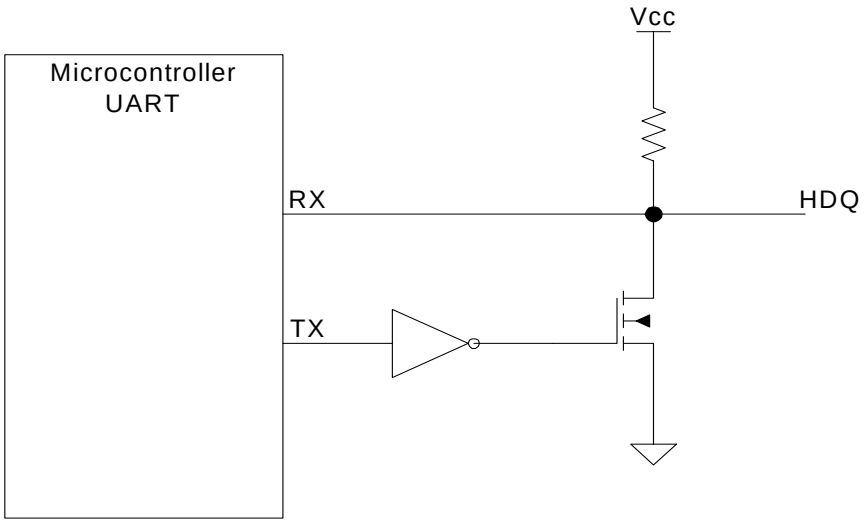
Normal host processing of communication with the battery monitors may encounter a high priority interrupt that needs to be serviced prior to completion of a communication transaction. The time between any bits sent from the host to the battery monitors is not time critical and as long as the HDQ line is high, the host processor may pause the communication and service an interrupt for any extended period and then resume the communication. However, when reading bits from the battery monitors, the host cannot allow any interruption, as the data is transmitted at a fixed rate and the host must know the exact timing to determine whether individual bits are ones or zeroes. Provision must be made for the host to flag any interruption

while receiving data. The host should then hold the HDQ line low, long enough to cause a break (no need to wait until transmission is complete from the battery monitor) and then retry the communication to avoid potential misreads of the data.

Using a UART for HDQ Communication

The host may be able to avoid the issue with communication repeats due to an inopportune interrupt if the host has a spare UART port to dedicate to the bq2019 communication. The UART may be set up to send and receive an 8-bit word for each bit of the communication. The UART may be set for a baud rate of 57,600 with no parity and 2 stop bits. This timing will send or receive a single word in 190 μsec and meet the maximum bit rate of the HDQ timing. The user could load the UART with 0xc0 to transmit a logic zero and with 0xfe to transmit a logic one. When receiving data, the host could interpret any data greater than or equal to 0xf0 as a logic one and any data less than 0xf0 as a logic zero.

Also note that the TX and RX of the UART must be tied together. In case the TX output is not an open-drain output, it will need to be inverted and then drive the gate of an NFET with the open drain tied to the RX input. See Figure 1.



HDQ Comunication Circuit For UART Without Open Drain Outputs

Sending a Break before Each Communication

If the battery pack is removed or some intermittent connection occurs on the HDQ line, the battery monitor may become out of sync with the host communication. A safe practice would be to issue a break prior to each communication to insure that communication was always in sync. An alternative procedure might be to always send a break at the start of a communication block and then not send it again unless there was a need to abort the communication or if the battery monitor did not respond within the expected time.

APPENDIX A: HDQ Assembly Code Example

```
;hdq routines:  receives and transmits 1 byte using HDQ protocall
;timed bit out routine
;timer 1 is setup as 1 micro second timer.
;this is written for a PIC 16c66 using a 8051 macro compiler.
;There are 2 HDQ ports on this example. Either one can be eliminated if not
required.

;hdq_d1           =       rc.3
;hdq_d           =       rc.4

hdq_read_byte
        call    hdq_break               ;send a break before each communication
        mov     hdq_buf,hdq_cmd         ;put hdq_cmd into hdq_buf
        clrb    hdq_buf.7               ;set for read
        call    hdq_tx                  ;transmit command
        jb      hdq_error_flg,hdq_finish_error  ;if error then jump
        mov     w,hdq_mode        ;
        xor      w,#11011000b     ;turn port to input
        mov    !rc,w              ;

        call    hdq_rx            ;read data
        jb      hdq_error_flg,hdq_finish_error  ;if error then jump
        mov     !rc,#11000000b                  ;done so reinit port (all outputs)
        ret


hdq_write_byte

        call    hdq_break               ;send a break before each communication
        mov     hdq_buf,hdq_cmd         ;put hdq_cmd into hdq_buf
        setb    hdq_buf.7               ;set for write
        call    hdq_tx                  ;transmit command
        jb      hdq_error_flg,hdq_finish_error  ;jump if error
        clr     tmr1l                           ;reset timer
        setb    tmr1on
hdq_write_byte1
        cjb     tmr1l,#190,hdq_write_byte1      ;wait correct time before transmit
byte
        clrb    tmr1on
        mov     hdq_buf,hdq_data
        call    hdq_tx                          ;transmit byte
        mov     !rc,#11000000b                  ;reinit port
        ret

;*****************************  Transmit  data  ******************8
hdq_tx
      mov   count,#8
hdq_txlp
        setb    do1
```

```
        sb     hdq_buf.0
        clrb    do1
        call    hdq_bit_out

        rrf     hdq_buf
      decfsz      count
        goto    hdq_txlp
        retlw   0



;reciev data routine
hdq_rx
;need to check for stop to start transition here
        mov     count,#8
        clr    hdq_data
      bcf   status,0
hdq_rxlp rrf    hdq_data
        call    hdq_bitin
        clrb    hdq_data.7
        snb     di1        ;skip if not a 1
        setb     hdq_data.7
      decfsz      count
        goto    hdq_rxlp
      retlw 0


hdq_bit_out
        clrb    tmr1on
        clr     tmr1h
        clr     tmr1l

        clrb    hdq_d
        clrb    hdq_d1
        mov     w,hdq_mode
;hdq_d1           =       rc.3
;hdq_d            =       rc.4 command 52

        xor      w,#11011000b
        mov    !rc,w
        setb    tmr1on

hdq_bitout_init_start_wait
        cjb     tmr1l,#14,hdq_bitout_init_start_wait
        sb      do1
        goto    hdq_bitout_low
        setb    hdq_d1
        setb    hdq_d
        goto    hdq_data_hold
hdq_bitout_low
        clrb    hdq_d
        clrb    hdq_d1
hdq_data_hold
        jnb     do1,hdq_data_hold1    ;if hdq is supposed to be set
```

```
        mov     w,rc
        and     w,hdq_mode
        jb     z,send_hdq_error  ;but its not then there is an error
hdq_data_hold1
        cjb     tmr1l,#107,hdq_data_hold     ;wait for 110 micro seconds
        setb    hdq_d
        setb    hdq_d1
        cje     count,#1,hdq_bit_out_end      ;if this is last bit then skip the
wait to make sure we get the input
stop_bit_hold
        mov     w,rc
        and     w,hdq_mode
        jb     z,send_hdq_error    ;if the stop bit did not go high then there is
an error
        cjb     tmr1l,#200,stop_bit_hold

hdq_bit_out_end
        clrb    tmr1on
        ret

hdq_bitin
        setb    di1
        mov     w,hdq_mode
        xor     w,#11000000b
        mov     !rc,w
        clr     tmr1l
        setb    tmr1on
hdq_init_start_wait
        cja     tmr1l,#250,send_hdq_error  ;if line stays high greater than 250
micros then there is an error
        mov     w,rc
        and     w,hdq_mode
        jnb      z,hdq_init_start_wait   ;wait for start bit to sync
        clr     tmr1l
hdq_start_wait
        cjb     tmr1l,#12,hdq_start_wait  ;test at about 12 micro seconds
        mov     w,rc
        and     w,hdq_mode
        jnb      z,send_hdq_error

hdq_data_wait
        cjb     tmr1l,#65,hdq_data_wait  ;test at 65 micro seconds

        mov     w,rc
        and     w,hdq_mode
        snb      z
        clrb    di1     ;clear data bit
hdq_stop_wait
        cjb     tmr1l,#155,hdq_stop_wait
        mov     w,rc
        and     w,hdq_mode
        jb      z,send_hdq_error

        clrb    tmr1on
        ret
```

```
hdq_break

;hdq_d1           =       rc.3
;hdq_d            =       rc.4 command 52
;        mov     this is command 52: hdq_mode,#00010000b  so 52 is rc.
        setb    hdq_d
        setb    hdq_d1
        mov     w,#11011000b
        mov     !rc,w
        clrb    hdq_d
        clrb    hdq_d1
        mov     w,hdq_mode
        xor      w,#11011000b     ;just make sure that the rs232 lines are enabled
and correct hdq is output while other
        mov    !rc,w              ;   is imput
        clrb    hdq_d
        clrb    hdq_d1

        clr     tmr1l
        setb    tmr1on
hdq_break_low_wait
        cjb     tmr1l,#220,hdq_break_low_wait
        setb    hdq_d
        setb    hdq_d1
        clr     tmr1l
hdq_break_low_wait1
        cjb     tmr1l,#60,hdq_break_low_wait1
        ret

hdq_finish_error
       mov     rc,#00000000b
        mov     !rc,#11011000b
        ret
send_hdq_error
        setb    hdq_error_flg
        ret
```

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

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

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI 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 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. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation.

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

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

Products		**Applications**	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DSP	dsp.ti.com	Broadband	www.ti.com/broadband
Interface	interface.ti.com	Digital Control	www.ti.com/digitalcontrol
Logic	logic.ti.com	Military	www.ti.com/military
Power Mgmt	power.ti.com	Optical Networking	www.ti.com/opticalnetwork
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
		Telephony	www.ti.com/telephony
		Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

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