bq500xxx
Wireless Power Transmitter I2C Interface

Application Note

Literature Number: SLUxxx
June 2015

BQ500xxx Wireless Power Transmitter I2C Interface

Oettinger, Eric

1. Introduction

The bq500xxx family of devices supports an I2C interface which can be used to program the device, perform tuning, and monitor various system parameters. This may be useful in an end product for device control or an advanced status display. During development it may provide feedback useful for system calibration and design guidance to improve system performance. It also provides a mechanism for field updates on most platforms. All devices described in this application note use a common I2C hardware block except the cost optimized bq500511 which has been designed around a different core and uses different peripherals. The behavior of the I2C block is comparable in the two devices for most applications; notes highlight features and specification limits which are unique to one device or the other.

2. Physical Bus Interface

The communication protocol is based on the SMBus definition. All bq500xxx controllers function as SMBus slave devices. For electrical details of the communication please refer to the SMBus specification which can be found on the web at: http://smbus.org/specs/smbus110.pdf. The slave address assigned to all bq500xxx wireless power devices has been hardcoded to 20 (decimal). The hardware can support 100 kHz, 400 kHz, or 1 MHz operation. (The bq500511 is limited to 100 kHz operation.)

3. Texas Instruments Supported Tools

The easiest method to access the basic features of the I2C interface is using TI's USB Serial Interface Adapter http://www.ti.com/tool/USB-TO-GPIO in combination with the "bqTesla TX Tuning Tool" which can be obtained by contacting Texas Instruments support.

Note one exception is the bq500511; with a new microprocessor core, it is incompatible with the USB Serial Interface Adapter. The EV2300/EV2400 USB adapter will be used for production support tools when released.

Ground, Clock and Data are the only three signals needed to interface between the bq500xxx device and the USB Serial Interface Adapter (or any other I2C host). In all 48-pin bq500xxx controllers, Clock is pin#10 and Data is pin#11; in 64-pin controllers, Clock is pin#15 and Data is pin#16. On the bq500511 Clock is pin#29; Data is pin#28.

4. Supported Commands by Device Part Number

Some commands have been added or removed as device requirements changed, and others have been modified to provide slightly different behavior based on changing system needs and customer feedback. The following table provides a summary of which commands are supported by each device in the family. Details on each of the commands listed here are provided in section 5.

bq500xxx:	cmd	210	211	211A	212A	215	410A	412	414Q	511
CLEAR_STATS	0xE1	No	No	No	No	Yes	No	No	No	No
DEVICE_ID	0xFD	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
DIAG_MODE	0xDE	Yes	Yes	Yes	Yes	Yes[1]	Yes	Yes	Yes	Yes
FREQ_SHIFT	0xD8	No	No	No	No	No	No	No	Yes	No
PLD_MONITOR	0xD5	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
PLD_THRESHOLD	0xD6	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
RX_PROP	0xD3	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
RX_PROP_COUNT	0xD4	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
RX_STATS	0xD0	No	Yes	Yes	Yes	Yes[2]	Yes	Yes	Yes	Yes
SHUTDOWN	0xD7	No	No	No	Yes	No	No	Yes	Yes	No
SLEEP_DISABLE	0xD2	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
TX_STATS	0xD1	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
TX_STATS2	0xDD	No	No	No	No	Yes	No	Yes	Yes	No

(1) For fixed frequency devices, DIAG_MODE specifies rail voltage instead of frequency/duty.

(2) In proprietary 10W mode, the 16-bit format of received power is recorded.

5. Command definitions

Commands may be Read Only, Write Only, or Read/Write and are further categorized into BYTE, WORD, or BLOCK types. BYTE commands specify an 8-bit operand (both read/write). WORD commands are 16-bits. BLOCK commands have variable length which is specified as part of the block message.

5.1 CLEAR_STATS (Write only BYTE – Command 0xE1)

The transmitter stores a copy of all data it receives from the receiver being charged (see command RX_STATS). It may be useful to be able to clear this data for display purposes or to detect a change in the system state. Note that none of the cleared parameters are used as "working variables", so this command will have no impact on transmitter operation.

In addition the RX data, it clears the error counters which are used to report the number of good and bad received messages (see command TX_STATS).

CLEAR_STATS is a write-byte command type. The data byte sent is completely arbitrary; it will have no effect on the command operation.

5.2 DEVICE_ID (Read only BLOCK – Command 0xFD)

The transmitter firmware build information is returned in a string containing the device number, the firmware version (major.minor.sub.build), and a date-code (YYMMDD).

Ex: BQ500210|2.1.4.3548|110714

5.3 DIAG_MODE (Read/Write BLOCK – Command 0xDE)

WARNING: The DIAG_MODE command has the potential to damage receiver devices and should be used with caution.

DIAG_MODE puts the transmitter in an open loop operating mode driving a constant frequency and duty cycle. This capability can be useful to examine system behavior during design and development.

While operating in DIAG_MODE the demodulator is disabled, and commands from a receiver on the pad will not be decoded. Specifically – any Control Error Packet commands for less power or End Power Transfer commands which may be intended to prevent system damage are not recognized and not responded to.

After DIAG_MODE has been disabled, normal system operations should be restored. HOWEVER, this is intended as an internal development tool and as such it has not been thoroughly tested for proper restoration under all conditions. To ensure not adverse side effects linger from the DIAG_MODE operation, a power-cycle is recommended.

The block written to DIAG_MODE comprises the following bytes in order:

byte(s)	parameter	units / scaling
0	number of bytes to follow	constant = 3
1	enable(1) / disable(0)	binary
2	frequency	kHz (0-255)
3	duty-cycle	percent on time (0-50)

5.4 FREQ_SHIFT (Write only BLOCK – Command 0xD8)

In automotive applications, the keyfob (which typically operates around 130 kHz) may experience interference from wireless chargers. Temporarily changing the transmitter's operating frequency may minimize such interference.

The FREQ_SHIFT command requires two arguments: the new frequency and the length of time it should continue operating there.

FREQUENCY is specified by a single byte indicating the new frequency in kHz. Care should be taken when specifying frequency, as no boundary limits are checked. **As with DIAG_MODE commanding operation at the resonant frequency can generate a large magnetic field which is potentially damaging to a receiver.**

DURATION is specified as a 16-bit integer number of milliseconds.

The block written to FREQ_SHIFT comprises the following bytes in order:

byte(s)	parameter	units / scaling
0	number of bytes to follow	constant = 3
1	frequency	kHz
2	duration msb	msec * 256
3	duration lsb	msec

The FREQ_SHIFT command will change the operating point to the new frequency immediately upon receiving the command. When the duration expires, normal Control Error Packets from the receiver will restore the power transfer point (this is done rather than jumping immediately back to the original frequency because system conditions may have changed). This may cause the duration to appear somewhat longer than what is specified.

5.5 PLD_MONITOR (Read only BLOCK – Command 0xD5)

The parasitic loss monitor command returns data related to the FOD or PMOD operation. The block returned in response to PLD_MONITOR comprises the following bytes in order:

byte(s)	parameter	units / scaling
0	number of bytes to follow	constant = 31
1	reported received power	128ths of max power
2	raw reported max power	*500 = mW
3-6	threshold set from resistor	mw (q19,13)
7-10	calculated parasitic loss	mw (q19,13)
11-12	DC input voltage	V (q6,10)
13-14	DC input current	mA (q13,3)
15-18	Calculated input power	mw (q19,13)
19-22	COMM+ amplitude	V (arbitrary)
23	PLD state machine	0=clear 1=triggered 2=set
24-27	PLD timer	msec (q16,16)
28-29	output frequency	kHz (q10,6)
30-31	spare	

Note: For V1.0 receivers, the value in byte 1 is “rectified power” and reported in 100ths of max power.

5.6 PLD_THRESHOLD (Read/Write WORD – Command 0xD6)

The parasitic loss threshold can be used to overwrite the resistor determined value. Note that this value will be reset when the device is power cycled. Adjusting the threshold is useful during development and characterization to determine an appropriate value for the MOD_THRESH resistor or to evaluate friendly losses in the system. The threshold value specified defines the mW threshold; it is expressed as an integer. A threshold value of -1 will disable FOD/PMOD.

5.7 RX_PROP (Read only BLOCK – Command 0xD3)

The WPC specification allows the receiver to send “proprietary packets”. There are several header numbers designated as proprietary which containing a various number of bytes of data. These packets may be used for some wireless power transfer related function, or they could be used to convey information from the device being charged to the host controlling the wireless charger. i.e. the wireless charging system can provide a conduit to pass information. When a proprietary packet is received, it is stored in the transmitter memory, and a counter containing the total number of proprietary packets received is incremented.

The RX_PROP command will return 25 bytes. Byte 0 specifying the length of block to follow (24), followed by 24 bytes containing the data from the most recently received proprietary packet. Twenty-four bytes is large enough to contain all of the data from any of the proprietary packets.

After RX_PROP is read, the header byte in memory is reset to 0. This allows the host to detect when a new proprietary packet is received.

5.8 RX_PROP_COUNT (Read only WORD – Command 0xD4)

The RX_PROP_COUNT command returns the total number of proprietary packets that have been received.

Note: Power cycling or entering low-power mode will clear this counter.

5.9 RX_STATS (Read only BLOCK – Command 0xD0)

The RX_STATS command returns data which was communicated from a wireless power receiver placed on the transmitter. Presently there are nine message types defined by the WPC specification that the RX can send. The most recent value for each of these messages is returned in a block when the RX_STATS command is issued. Following is the list of WPC defined packets:

Header	Message	Bytes	Comment
0x01	Signal Strength	1	Sent only once when RX placed on pad.
0x02	End Power Transfer	1	Will contain the most recent EPT code.
0x03	Control Error	1	Latest value only – updated frequently.
0x04	V1.0 Rectified Power V1.1 Received Power	1	Used for PMOD / FOD
0x05	Charge Status	1	Optional packet from RX.
0x06	PID Holdoff	1	Optional – will contain default if not received.
0x51	Configuration	5	Needs post processing to decode.
0x71	Identification	7	Needs post processing to decode.
0x81	Extended Identification	8	Presently undefined.

The block returned in response to RX_STATS comprises the following bytes in order:

Byte	Message	
0	28	(number of bytes to follow)
1	signal_strength	
2	end_power_transfer	
3	control_error	
4	8-bit rectified_power / received power (V1.0 receivers will send a rectified power message)	
5	charge_status	
6	holdoff	
7	configuration [0]	** For multi-byte messages, the order matches the order
...	...	** sent by the recevier. i.e. configuration[0] is the first
18	identification [6]	** byte received in a configuration message.
19	extended_identification [0]	
…	…	
26	extended_identification [7]	

For all low-power (<=5W) transmitters

27	*0*
28	*0*

For medium power (>5W) transmitters, the received power message is sent with higher resolution

27	*16-bit received power (msb)*
28	*16-bit received power (lsb)*

5.10 SLEEP_DISABLE (Read/Write BYTE – Command 0xD2)

When configured to operation with the low–power supervisor MSP430, the BQ500xxx attempts to reduce system power consumption by turning itself off when not in use (when no receiver is present, after a fault, or when the receiver sends an “End Power Transfer” message). While off, communication is not possible, and issued commands have no effect. Additionally statistical counters such as those for good and bad messages will be reset on wake–up.

When set to 1, SLEEP_DISABLE, will prevent the BQ500xxx from entering sleep mode. This disables the low-power standby mode which is normally implemented by periodically turning off the 3.3V input power to the processor. When power is removed, the processor is unable to communicate, which can make debug and system diagnosis more difficult. The SLEEP_DISABLE function prevents the low-power mode from operating which leaves communication with the devices possible during system idle periods when normally it would be prevented.

Note 1: Communication must be currently active in order to issue this command. This can be due to the presence of a functioning receiver, or forced by shorting the LED_MODE selection resistor to ground.

Note 2: SLEEP_DISABLE should never be set to 0 in system configurations which do not include an MSP430 low–power supervisor or the second generation power saving SLEEP and SNOOZE circuitry. Unpredictable behavior may result.

5.11 TX_STATS (Read only BLOCK – Command 0xD1)

The transmitter status command returns data related to the operating status of the transmitter. The data includes recent results from the ADC converter, statistics regarding the communication channel, the present power operating point (frequency and duty cycle), status indicators, and parasitic loss parameters.

The block returned in response to TX_STATS comprises the following bytes in order:

byte	parameter	description	units	scaling
0	31	number of bytes to follow		
1	voltage_in msb	Input voltage	volts	(q6,10)
2	voltage_in lsb			
3	iout msb	I_SENSE current	mA	(q13,3)
4	iout lsb			
5	temp_ext msb	external temperature	raw ADC result.	
6	temp_ext lsb			
7	temp_int msb	internal temperature	degC	(q9,7)
8	temp_int lsb			
9	good_msg_cnt msb	good message counter		
10	good_msg_cnt ..	count of successfully received messages		
11	good_msg_cnt ..			
12	good_msg_cnt lsb			
13	bad_msg_cnt msb	bad message counter		
14	bad_msg_cnt ..	count of detected errors: checksum, invalid values,		
15	bad_msg_cnt ..	timeouts, hotswaps, and unknown message types		
16	bad_msg_cnt lsb			
17	frequency msb	operating frequency	kHz	(q10,6)
18	frequency lsb			
19	duty_cycle msb	operating duty_cycle	percent	(q1,15)
20	duty_cycle lsb			
21	led_mode	resistor selected led mode		
22	led_out	present LED indication 4–bits per LED 0=off,1=slow,2=fast,3=on		
23	mod_threshold msb	resistor set threshold	mW	(q19,13)
24	mod_threshold ..			
25	mod_threshold ..			
26	mod_threshold lsb			
27	pld msb	parasitic loss detected	mW	(q19,13)
28	pld ..			
29	pld ..			
30	pld lsb			
31	cs100_latched	indicator of CS100 detection		

Note: The “Q-notation” used in the scaling convention is a fixed point representation of a floating point number comprising the number of integer bits and the number of fractional bits. Ex. (q9,7) denotes 9 integer bits and 7 fractional, and the conversion can be made by dividing by 2 raised to the fractional count. If the internal temperature variable returned is 0x0F14 = 3860 (decimal) the internal temperature of the device is 3860 / 2^7 = 30.16^{o}C.

5.12 TX_STATS2 (Read only BLOCK – Command 0xDD)

The block size is limited to 32 bytes; to accommodate a few more fields with useful information, a second TX_STATS command was required. Presently mostly filled with 0's, there is room for expansion.

The block returned in response to TX_STATS2 comprises the following bytes in order:

byte	parameter	description	units	scaling
0	31	number of bytes to follow		
If device supports FOD calibration				
1	fod_correction msb	Value read from FOD_CAL input	mW/A	1024/1000
2	fod_correction lsb			
Else				
1	0			
2	0			
If device supports DPL				
3	dpl_state	value >0 indicates DPL triggered	state machine status	
Else				
3	0			
4	0	Padding until needed.		
...	...			
31	0			

5.13 SHUTDOWN (Read/Write BYTE – Command 0xD7)

(bq500x14 only) SHUTDOWN provides a method to manually start or stop the transmitter. While SHUTDOWN is set to 1, the transmitter will remain idle and will not ping. If the transmitter is actively delivering power when SHUTDOWN is set to 1, it will cease power transfer.

When SHUTDOWN is set to 0, the transmitter will start pinging and when a device is detected it will begin (or resume) power transfer normally. By default the SHUTDOWN is set to 1; this means that a command clearing SHUTDOWN is required in order to establish power transfer with a receiver.

6. BQ500xxx Field Updates

The executable firmware in the bq500xxx devices can be updated through the I2C interface, with the exception of the lower cost bq500511 device which does not have this capability.

From an environment where a PC is available, the most straightforward method of programming is via the Firmware Download Tool, part of the Fusion GUI suite of tools, available at http://www.ti.com/fusiongui.

In some PC programming environments, the command line version of the downloader may be preferred. After installing the Fusion tools the environment path should be configured correctly and from a console window the following command will program the device:

```
FusionFirmwareDownload --state auto --pflash download --dflash download --pflash-checksum calc
--execute-program --infile c:/temp/wp410.x0
```

The response should appear as follows:

```
Found BQ500410 Firmware v2.3.1.5064 @ Address 20d, sending it to ROM mode ...
Reading PKGID version ...
Starting download ...
Parsing firmware file ...
Firmware is in Tektronix Extended format
Mass erasing program flash
Downloading program flash ...
Mass erasing data flash
Downloading data flash ...
Verifying program flash ...
Having ROM calculate checksum for program flash 0x10000 through 0x17FFB ...
Program Flash checksum match (0x0033F79C)
Verifying data flash ...
Reading data flash ...
Data Flash verified!
Writing program flash checksum 0x33F79C ...
Validating checksum written through read back ...
Program flash checksum AOK
Executing program ...
```

A third method of downloading firmware is through an I2C master. Detailed programming information for this approach can be found here:

Firmware Upgrade of UCD Devices by 3rd Party Hosts Over I2C

Revision History

Changes from Original (July 2014)			Page(s)
•	Removed table footnote stating 215 and 414Q were pre-release		3
•	16-Apr-2015	Added command number for TX_STATS2	11
•	22-Jun-2015	Added CMD number and bq500511 capabilities to table.	3
		Interface changes addressing bq500511.	2, 12
•	23-Jun-2015	Description of bq500511 uniqueness, bus speed, and I2C pins.	1

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