AVR459: SB200 Hardware User's Guide

Features

- **Full evaluation of Atmel smart battery reference design.**
- **Socket for SB20x smart battery reference designs.**
  - **Development platform for AVR based smart batteries.**
- **Communication gateway between PC and smart battery.**
  - **Single Wire UART for single and dual cell batteries.**
  - **TWI/ SMBus™ interface for larger batteries.**
- **Battery Charger.**
  - **Charge with up to 19V and 2A.**
  - **Auto-limitation of charging voltage and current.**
  - **Auto-termination of charging when charging period expires.**
- **Adjustable Constant Current Load for battery discharging.**
  - **Sink up to 4A continuous current.**
  - **Auto-shut down if too hot.**

1 Introduction

The SB200 is a development platform for SB20x smart battery reference designs, which offers an easy way to start evaluation and hence development of smart battery applications using Atmel® AVR® microcontrollers. The SB200 is not in itself a reference design, but is intended to be used with e.g. SB201-1, which is a reference design that demonstrates implementation of a single cell smart battery (Li-Ion).

16340/CR123A battery cells can be inserted in the SB200 battery sockets, and a SB20x is inserted in the dedicated daughter board socket. This forms a “battery pack” with a given number of cells, safety protection and full battery capacity monitoring capabilities. Alternately other cell types can be soldered into the SB200.

To simulate the use of the “battery pack” the SB200 features a Charger that can charge the battery pack, and a Constant Current Load that can simulate an application that draws power from the battery pack.

Figure 1-1. SB200 with two batteries and SB201-2 inserted.



8-bit AVR® Microcontrollers

Application Note

Rev. 8132A-AVR-10/08

2 Safety precautions when using Li-Ion batteries

The batteries provided with the SB200 have built in protection which might cause strange effects to occur if the battery is exposed to overstress situations, such as over-charging. The built in protection does affect the battery characteristics, especially in terms of resistance, but will ensure that even incorrect use and bugs in a smart battery application under development does not pose a safety hazard.

Nevertheless, if battery cells heat up, are physically damaged or shows other signs that can cause or be a consequence of malfunction, the battery cell should be discarded into a safe deposit.

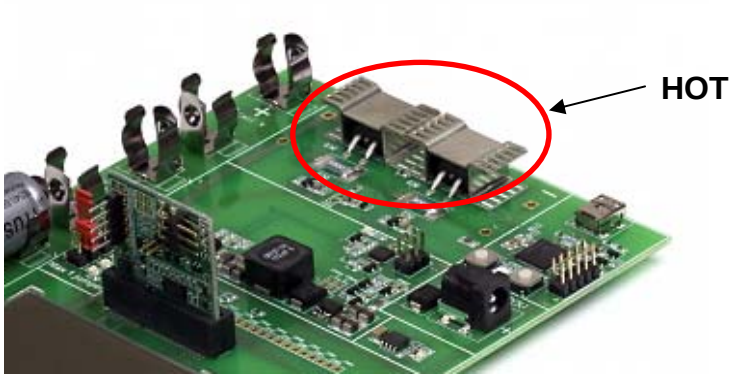
If other batteries without protection are used, please observe the safety guidelines following the batteries. Li-Ion and Polymer batteries & packs may explode and cause fire if misused or defective.

2.1 Warning – HOT!

Be aware that the heat sinks of the Constant Current Load can get **very hot**. The shunt resistors (0.5Ω) at the load and from the charger will also get very hot. Do not touch them!

The SB200 will limit the temperature to approximately 125°C by turning off the load if the temperature rises above this point.

Figure 2-1. Constant Current Load – hot heat sink.



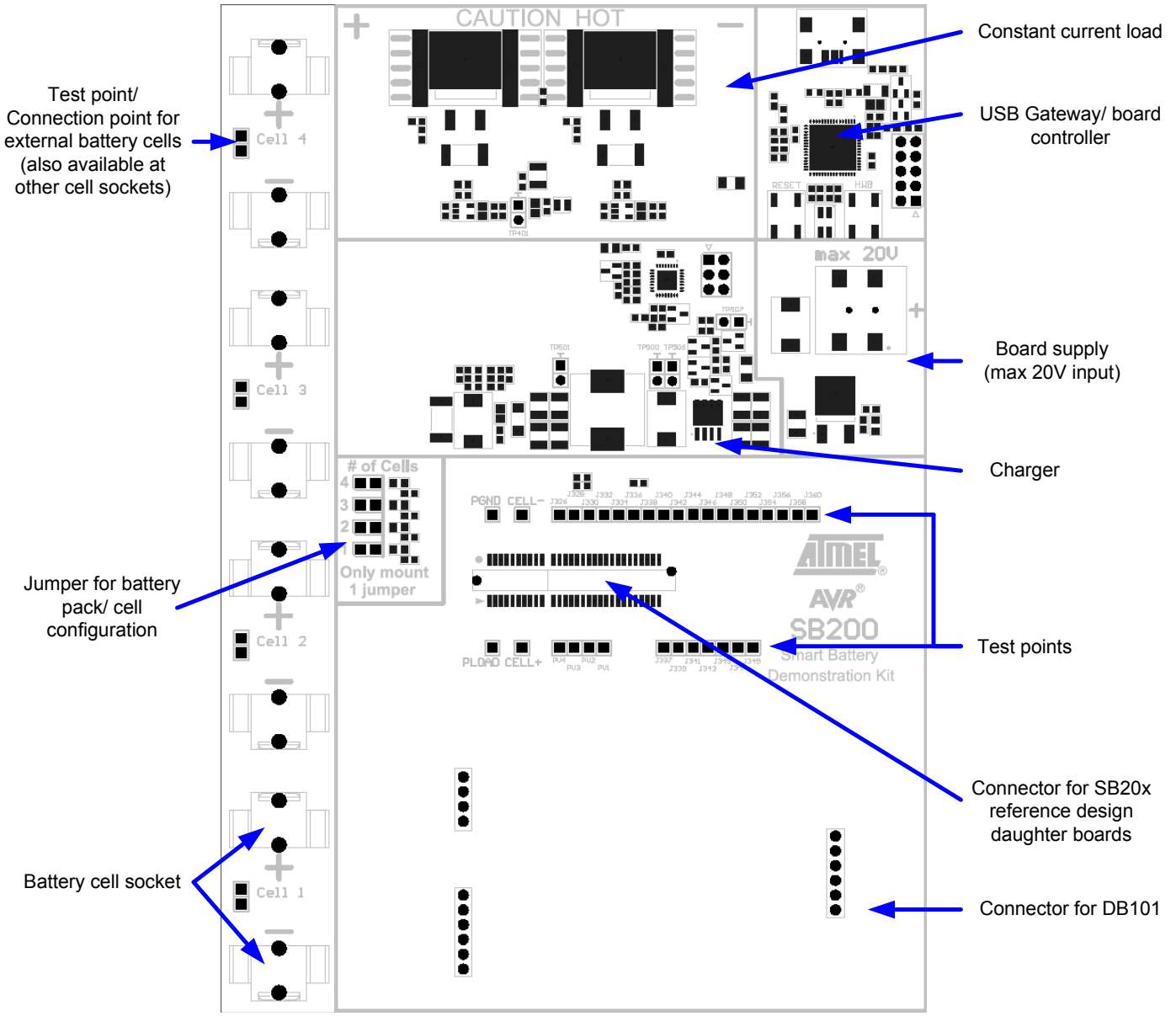
3 Functional overview

The SB200 is developed to ease development of Smart Battery applications: It offers a communication gateway between the SB20x reference designs which uses half duplex UART or SMBus to USB. It can charge the batteries connected or inserted in the board, and act as a load to investigate behavior during discharging of the battery.

The SB200 comes with a PC-software that controls the various features of the kit. An introduction to this software is found in application note “AVR491 – quick start guide for SB200”. Both the executable and the source code for the PC software are available from the Atmel web-site, and can be installed into AVR Studio.

An overview of the hardware is seen in Figure 3-1

Figure 3-1. Placement of hardware modules on SB200 (areas within silkscreen rectangles/shape).



4 Details of the SB200 blocks

The following sections describe the SB200 blocks on a functional level. Electrical schematics, a Bill of Materials and CAD (gerber) files are available as separate files, which can be downloaded from the Atmel web site http://www.atmel.com/sb200.

The source code for the SB200 firmware is available from the Atmel web site as well, but is provided as is. No support is provided on this part of the code, as it is made as a development tool for smart battery application only. SB200 firmware is based on demo application for AT90USBKEY. The supported code is therefore only the source code for the SB20x smart battery reference designs.

5 Insertion of battery cells and placement of the corresponding jumper

Insert only the number of battery cells that are intended to be used with the SB20x in the SB20x daughter board socket. Further, make sure to place the jumper that selects the number of cells correct – otherwise one of the battery cells will be shorted.

Recommended order

Step	Action	Remark
1	Power Off SB200	Remove Batteries !
2	Insert SB201-?	
3	Start PC SW	
4	Power On SB200	Check that correct SB201 is shown in status bar
5	# of Cells LED should light up	
6	Place Red Jumper correct	Only one jumper
7	Insert battery / batteries	
8	Press HWB button	One minute charge is started

6 Supply voltage

SB200 can be supplied with up to 20V through the DC jack. The DC jack has positive center. Minimum supply voltage is 7V.

The supply voltage need not be 20V, but needs to be a few volts higher than the maximum voltage of the battery stack on the board. The Charger and the opamp's in the Constant Current Load need that to function properly.

If a two cell smart battery application is being developed it is therefore sufficient to connect e.g. 10V. A higher voltage will only generate unnecessary heat in the board supply regulator, which is used to generate 5V for the USB1287, tiny861 and the DAC.

7 Charger

The Charger on the SB200 can deliver 0-20 volts and 0-2 ampere, using a constant current, constant voltage charging scheme. It is configured by setting maximum voltage, maximum current and timeout. The Charger will make sure that no limit is exceeded and will try to make sure one of them is reached.

7.1 How it works

A fast PWM output on the Charger is connected via a buck converter to the battery. By changing the duty cycle on the PWM, the voltage to the battery can be adjusted. To be able to measure voltage and current, two ADC pins is connected on both sides of a shunt resistor just before the battery.

The Charger continuously measures the voltage and current and adjusts the PWM output accordingly. The PWM is adjusted one step at a time and this will make it take approximately seven tenths of a second to go from minimum to maximum output. If either maximum current, maximum voltage or timeout is set to zero, charging will stop immediately.

When setting maximum current and voltage, the charging time in minutes is also set. This can be up to 255 minutes and the Charger will stop when the time is reached.

The HWB button on SB200 starts a charge cycle with a duration of 1 minute (or until you press it again). The charge voltage is set depending on which SB20x that is connected. This will also power up the SB20x, if it was off. The voltage, current and timeout will be sent to the PC.

7.2 Resolution

The ADC on tiny861 has an external reference of 2.5V, and with 10-bit ADC, this give a resolution of 2.44mV. The voltage input to the ADC is scaled down 8x, so max voltage possible to measure is 19.98V, and resolution is 19.5mV. The current input uses the differential input with 20x gain over a shunt resistor of 0.5Ω. This is also scaled down 8x, thus giving a resolution of 1,95mA, so max current possible to measure is 2A. The PWM output to the buck converter has 512 values, which means a resolution of “Input Voltage” divided by 512, e.g. at 10V input the resolution will be approximately 19.5mV.

Note that the battery cell voltage reported in the SB200 PC application may vary during charging. This is because the Charger is continually adjusting the voltage to match the voltage and current limits set.

7.3 Communication

The Charger has a simple communication interface on the SPI bus. It understands two commands: One command is “SetCharging” and it configures maximum current, maximum voltage and timeout. The other command is “ChargerRead”. This command to the Charger reads the voltage and current seen from the ATtiny861. (Please refer to section 14.4 for more information about the communication).

If it receives any other command, doesn’t get all parameters or notices a checksum error it will just discard the message.

The implementation is based on the BC100 reference design and application note AVR458. Please refer to this application note for more information about charging Li-Ion batteries.

8 Constant Current Load

The constant current is implemented with a 14-bit DAC, differential amplifiers and two MOSFET transistors operated in their linear mode. The reference is 5V, step size is approximately 305µV. Shunt resistor is two of 0.5Ω in parallel. Resolution is 1.22mA and max current is 20A.

The load can draw high currents for a short time and is mainly limited by thermal overstress. The current drawn by the load is controlled from the PC software. If the current is kept below 2A for single cell and 1A dual cells applications the load should not overheat (the higher the voltage of the battery stack the lower should the current be).

The USB1287 monitors the temperature of the load, and will shut it down if it reaches 125°C to avoid damage to the MOSFET’s.

The load has fairly good accuracy at higher currents, while it may be somewhat inaccurate at lower currents. The constant current stability is very good and it can

therefore be used to test and characterize battery cells and application. The load can be calibrated to get better accuracy and linearity by rewriting the firmware for SB200.

8.1 Short circuit load condition

The load is implemented as an integrating circuit. This means that it can be used to simulate high out-rush currents: if the load is enabled, but no current source is available, it will saturate and open fully (max load). When a voltage/current source is enabled the load will for a short moment draw maximum current, and can be seen as a short circuit load. As soon as the current starts flowing the load will start to adjust the current to reach the target level.

This can cause problems if one is not aware of this, since the smart battery AVR MCU will be detect this as a high current/over current/short circuit situation. However, this characteristic of the load can be used to test an implementation of a smart battery, as it can be used intentionally to trigger a high out-rush current situation and validate correct behavior.

9 Test and connection points

The SB200 offer a variety of test and connection points, some for easy access to relevant signals with scope probes, and others to make it possible to connect external units.

9.1 External connection

On each side of the heat sink at the Constant Current Load, connection points are available. These can be used to measure the battery stack voltage, or an external charger can be connected here. The connection points are designed to match a 4mm power connector such as Farnell no: 1101128 / 1101127. This makes it also easy to connect an external load if the on-board load is not sufficient. It also allows for the on-board load to be used towards an external target, for testing purposes.

9.2 External battery cells

If it is desired to connect battery cells to the SB200 that does not fit in the sockets, these can be placed on an external board and connected to the SB200 test/connection points at each of the battery cell sockets (see Figure 3-1).

These connections points can also be used to connect an external power supply instead of batteries. This can be used to more easily and safely test how the smart battery implementation reacts to extreme voltage/current conditions.

If working with cells that fit in the sockets these test/connection points can be used to measure the voltage over the battery cells.

9.3 Signal line test points

All signals available on the SB20x edge connector are available as test points. These are located next to the SB20x daughter board connector (see Figure 3-1). 2.54mm pin headers can be mounted to make it easier to hook a scope probe to these test points.

Table 9-1. Signals on the test points next to the SB20x daughter board connector.

Test point designator	Signal	Description
J323	PV2	Positive voltage cell 2
J325	PV1	Positive voltage cell 1
J326(1)		
J327	PV4	Positive voltage cell 4
J328	SCL	TWI/SMBus clock
J329	PV3	Positive voltage cell 3
J330	SDA	TWI/SMBus data
J332(1)		
J334(1)		
J336(1)		
J337	PA3	AVR Port A pin 3
J338(1)		
J339	PA2	AVR Port A pin 2
J340(1)		
J341	PA1	AVR Port A pin 1
J342(1)		
J343	PA0	AVR Port A pin 0
J344(1)		
J345	Vref	Voltage reference
J346	CELL1BAL_ON	Cell 1 cell balancing enable
J347	GND_ID	GND for ID circuitry
J348	CELL2BAL_ON	Cell 2 cell balancing enable
J349	Vreg	Regulated supply voltage
J350	UART1W	
J352	MISO	SPI master input slave output
J354	MOSI	SPI master output, slave input
J356	SCK	SPI clock
J358	SS	SPI slave select
J360	RESET	Reset to smart battery AVR

Notes: 1. Test point reserved for future use

10 Programming the USB1287 or the tiny861

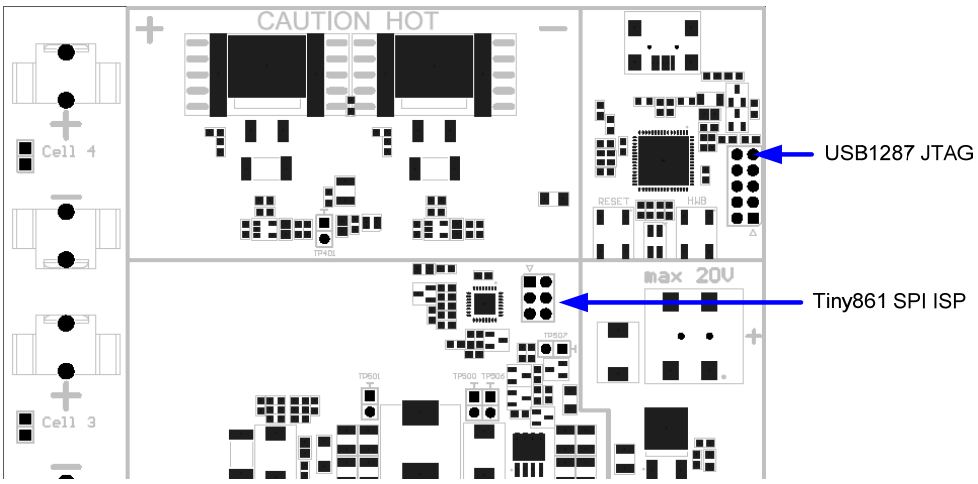
Reprogramming of the AT90USB1287 (USB gateway and board controller) and the ATtiny861 that controls the Charger is possible, and may be required to use the latest version of the firmware. It is recommended to always upgrade to the latest version of the firmware, found on www.atmel.com, to ensure that the firmware match this documentation and operates accordingly. The firmware source code running in these devices are available, but only as an extra service: These devices are part of the

development platform, not part of a reference design. Therefore the source code for these devices are provided as is, and with limited support. The product supported fully is the source code for the reference designs such as the SB201.

One case where update of the USB1287 firmware can be relevant is if a different communication protocol is desired to be used between the smart battery and the host. During the development it can be desired to use the USB1287 as a gateway to communicate with the smart battery via this interface/protocol. Changes and extensions to the communication interface/protocol are left to the users to implement.

To program the tiny861, the USB cable must be disconnected, and the USB1287 will release the SPI-lines which are used for inter-communication.

Figure 10-1. Programming interfaces for USB1287 and tiny861.



11 Buttons

Next to the USB1287 two buttons are present:

“Reset” will reset the USB1287 device and the Constant Current Load.

“HWB” will either enable the Charger for a short time to start the smart battery device, or it will stop the Charger next time the button is pressed.

SB20x	Max Voltage	Max Current	Timeout
SB201-1	4200mV	100mA	1min
SB201-2	8400mV	100mA	1min

12 Status LED

An RGB LED is located between the buttons. This indicates the status of the USB1287. When the LED is red the USB1287 is transmitting, and when it is green the USB1287 is receiving. As transmission is so fast that the human eye cannot distinguish the individual colors, the LED would appear yellow during transfers.

13 DB101 support for portable demonstration

A DB101 can be placed on the female pin header. This is an alternative to using a PC as host. The DB101 would typically only be used for demonstration, where standalone battery powered operation is desired. Atmel does not provide firmware for this.

13.1 Battery cells provided with SB200

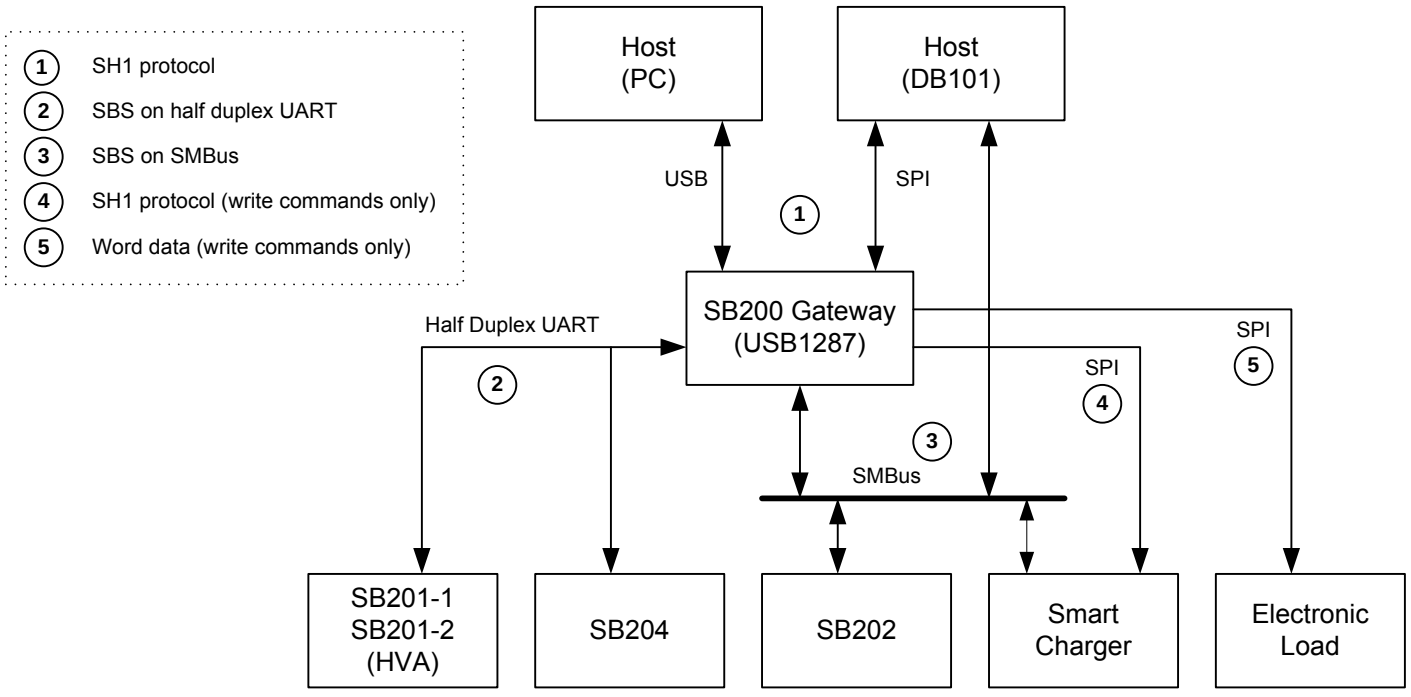
Though the capacity of the cells is specified to 880mAh they are actually closer to 700mAh.

The cells have built in protection, which affects the characteristic of the battery. It can be an advantage to use unprotected cells if a more correct behavior is desired. However, it is recommended to use protected cells until the implementation is verified to not allow overstress of the connected cells.

14 Communication Gateway

An AT90USB1287 AVR microcontroller acts as both communication gateway and board controller. It passes messages from its USB interface to other units on the SB200. The communication links on the SB200 is shown on Figure 14-1. The figure also shows reference designs SB204 and SB202 which are not available upon release of the SB200

Figure 14-1. Communication links on (and off) the SB200.



14.1 Communication protocol between the Host (PC) and the USB1287 gateway.

The SH1 protocol is built to be tested/invoked from a terminal, that's the reason for all the ASCII characters. The package uses a:

- Fixed character as preamble to indicate start of package
- Destination address byte
- Source address byte
- Command byte
- Size byte
- Data payload
- Simple checksum byte, the sum of previous bytes
- Carriage return character

The package is minimum 7 bytes and maximum 262 bytes in size. DATA is transmitted only if NBYTES is nonzero.

Example: Send 2 bytes from PC to Electronic Load.

SYNC	ADDRESS	SENDER	COMMAND	NBYTES	DATA	CS	CR
U	E	P	C	2	int mA	Calc	¬
0x55	0x45	0x50	0x43	0x02	0x04B0	0xE3	0x0D

14.1.1 Source and destination ADDRESS.

Address (ASCII)	Address (hex)	HW Module
C	0x43	Charger
D	0x44	DB101 Display
E	0x45	Electronic Load
M	0x4D	Main Controller
P	0x50	PC
S	0x53	SB20x

14.2 Global COMMAND

The commands for the different hardware modules are covered in the sections below. Commands can be uses in both read and write mode: if a command code (ASCII) is sent as upper case it is a WRITE, and if in lower case it is a READ. Each module has its own unique commands (commands not shared for different hardware modules).

In the following explanations, the COMMAND is marked yellow.

14.3 COMMAND to Constant Current Load.

The Constant Current Load only supports write commands: It is implemented with a 14-bit DAC and the Board Controller will process the requested current level to achieving the correct current level for the Constant Current Load.

The Board Controller will also monitor the PCB temperature and shut off the Constant Current Load if the PCB temperature is higher than 125°C.

Example: Set Constant Current Load to 2450mA

SYNC	ADDRESS	SENDER	COMMAND	NBYTES	DATA	CS	CR
U	E	P	C	2	int mA	Calc	¬
0x55	0x45	0x50	0x43	0x02	0x0992	0xCA	0x0D

14.4 COMMAND to/from Charger.

14.4.1 SetCharger.

The command to the Charger sets the maximum limits for charging voltage and current, and the charging time. The Charger will not exceed either of those, but will adjust the voltage and current levels to get as close to the limits as possible.

SetCharger: i.e. 8400mV, 450mA, 10 minutes

SYNC	ADDRESS	SENDER	COMMAND	NBYTES	DATA	DATA	DATA	CS	CR
U	C	P	S	5	int mV	int mA	char minutes	Calc	¬
0x55	0x43	0x50	0x53	0x05	0x20D0	0x01C2	0x0A	0xFD	0x0D

14.4.2 ChargingInfo.

The response from the Charger is sent when the HWB button is pressed. This will update the PC software.

ChargingInfo: i.e. 0000mV, 100mA, 1 minute

SYNC	ADDRESS	SENDER	COMMAND	NBYTES	DATA	DATA	DATA	CS	CR
U	P	C	s	5	int mV	int mA	char minutes	Calc	¬
0x55	0x50	0x43	0x73	0x05	0x0000	0x0064	0x01	0xC5	0x0D

14.4.3 ChargerRead.

This command to the Charger reads the voltage and current seen from the ATtiny861.

ChargerRead

SYNC	ADDRESS	SENDER	COMMAND	NBYTES	CS	CR
U	C	P	R	0	Calc	¬
0x55	0x43	0x50	0x52	0x00	0x3A	0x0D

14.4.4 ChargerReport.

The report from the Charger is sent when the PC has asked. This will update the PC software.

ChargerReport: i.e. 8385mV, 607mA

SYNC	ADDRESS	SENDER	COMMAND	NBYTES	DATA	DATA	CS	CR
U	P	C	r	4	int mV	int mA	Calc	¬
0x55	0x50	0x43	0x72	0x04	0x20C1	0x025F	0xA0	0x0D

14.5 List of COMMAND to/from Board controller.

14.5.1 Set the RGB LED.

Command to set the color of the RGB LED between the two switches at USB1287.

Leds_set_val(): i.e. SET RED Led on.

SYNC	ADDRESS	SENDER	COMMAND	NBYTES	DATA	CS	CR
U	M	P	F	1	char RGB	Calc	¬
0x55	0x4D	0x50	0x46	0x01	0x04	0x3D	0x0D

Where the bits in DATA that enabled the individual colors in the RGB LED are listed in table:

Table 14-1. Bits controlling RGB colors

LED color	Bit (set to enable)
Red	2
Green	1
Blue	0

14.5.2 Read the SB20x Hardware ID

When the Board controller has identified the hardware ID of the SB20x daughter card (refer to AVR455: SB201 Hardware User's Guide), the PC software can query that ID using the commands below.

Read_SB20x_HWid ()

SYNC	ADDRESS	SENDER	COMMAND	NBYTES	CS	CR
U	M	P	h	0	Calc	¬
0x55	0x4D	0x50	0x68	0x00	0x5A	0x0D

Response: Send_SB20x_HWid()

SYNC	ADDRESS	SENDER	COMMAND	NBYTES	DATA	CS	CR
U	P	M	h	1	char ID	Calc	¬
0x55	0x50	0x4D	0x68	0x01	0xB8	0x13	0x0D

14.5.3 Set the Battery Select LED.

The PC software can query SB20x_HWid, and then send a package to give a visual feedback to the user about where to place the red jumper. After reset of USB1287 this is done automatically by USB1287.

Batteryx_Led (): i.e. SET Cell2 Led on.

SYNC	ADDRESS	SENDER	COMMAND	NBYTES	DATA	CS	CR
U	M	P	M	1	char Led #	Calc	¬
0x55	0x4D	0x50	0x4D	0x01	0x02	0x42	0x0D

Led Value = 4,3,2,1

Table 14-2. Byte controlling the "# of Cells Leds"

# of Cells	DATA
4	4
3	3
2	2
1	1

14.5.4 Read the SB200 PCB Temperature.

The NTC placed close to the Constant Current Load can be read. Response is a temperature between -40°C to 125°C.

Read_Temperature ()

SYNC	ADDRESS	SENDER	COMMAND	NBYTES	CS	CR
U	M	P	t	0	Calc	¬
0x55	0x4D	0x50	0x74	0x00	0x66	0x0D

Response: Send_Temperature ()

SYNC	ADDRESS	SENDER	COMMAND	NBYTES	DATA	CS	CR
U	P	M	t	2	int Temp	Calc	¬
0x55	0x50	0x4D	0x74	0x02	0x0019	0x81	0x0D

14.6 Communication protocol between the USB1287 gateway and SB201/SB204.

The communication between the SB200 and SB201/ SB204 is based on a half duplex UART – here referred to as Asynchronous smart battery bus (ASBBus).

The SBS specification uses the SMbus protocol, but since SB201 uses software UART, some changes have been made to improve reliability. Addressing has also been removed.

SB201 supports the four different types of messages used by SBS:

- Read word
- Write word
- Read block
- Write block

A word command always has two bytes of data and a block command always 32 bytes, which is different from the SBS block commands that has variable length.

The host sends a command where the MSB determines if it's a write or read. 0 means write and 1 means read. This allows for 128 different commands that both support read and write. From the command itself it is not possible to determine if it is a word or block command; both sides have to agree what all commands are.

SB201 immediately sends back the command; this allows the host to detect if SB201 received the command correctly. What happens after that depends on the type of message used by that command.

A Packet Error-checking Code (PEC) is appended to all data to be able to detect errors. The PEC is a CRC8 with polynomial 0x8D (CRC8-CCITT) calculated from the command and all data bytes.

14.6.1 Read commands

On read commands, SB201 starts to send data after it has sent back the command. The PEC is also appended. However, if the command wasn't supported, nothing is sent back (except for the command byte).

14.6.2 Write command

On write commands, the host can start to send data to SB201 after it has received the replied command byte. After SB201 has received the whole message, it will execute the command if it was understood and the PEC was correct. It will send an ACK back to the host after it has finished executing it. If the command was supported but the PEC wrong, a NACK is sent back. On all other errors nothing is sent back.

ACK is 0xFF and NACK is 0x00.

Figure 14-2 A read word command

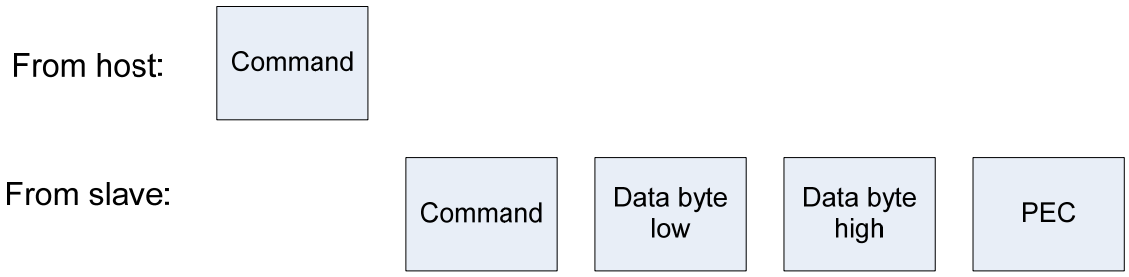


Figure 14-3 A write word command

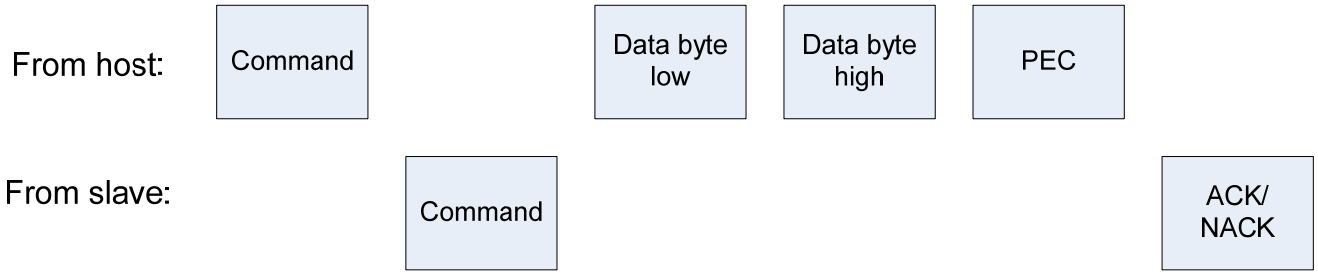


Figure 14-4 A read block command

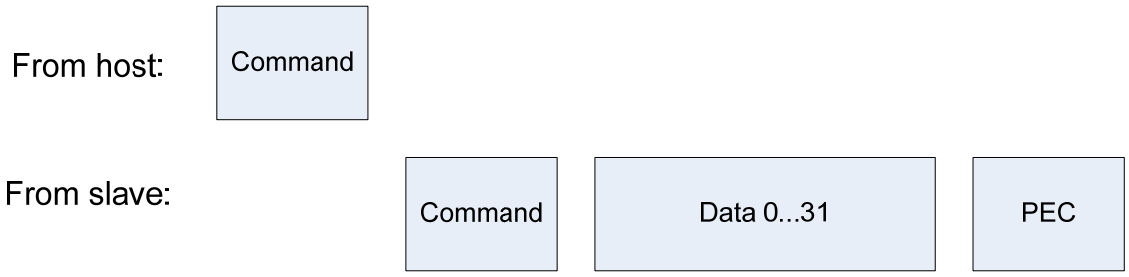
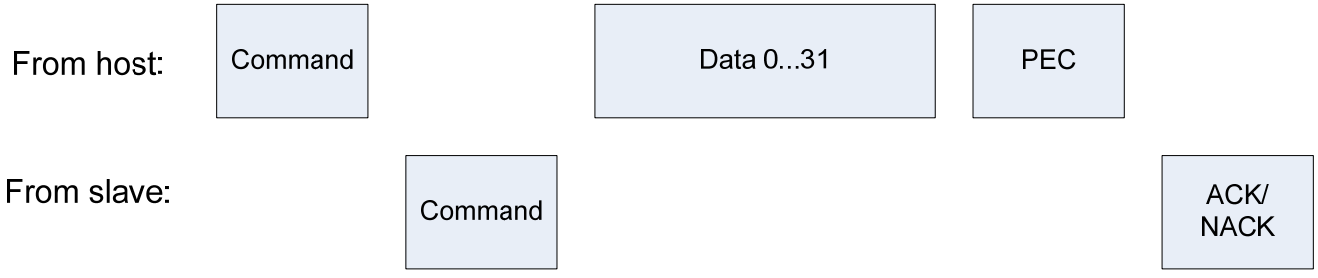


Figure 14-5 A write block command



14.6.3 Errors

At 4800 baud, there is only 208 clock cycles for SB201 between each bit. This means that to ensure error free communication all the time, all other interrupts would have to be less than 104 cycles, including the prologue of the timer interrupt. This is not always true for the interrupt service routines used by SB201. However, since those interrupts (CCADC & VADC conversion complete) only runs once per second, and since the blocking interrupts has to occur just a few cycles before the timer interrupt, a retransmission of the “broken” communication is sufficient and adequate to solve this.

The errors the host can detect include:

- Wrong command returned by SB201
- Frame error on data sent from SB201
- PEC incorrect on data sent from SB201
- SB201 does not respond within reasonable time (min 50ms)
- SB201 sends a NACK to indicate an error occurred

On all those errors the host should send a break/arbitration lock and retransmit. A break/arbitration lock is a minimum of 18 normal bit periods of low on the communication line. This resets the communication on SB201 and makes it ready to receive a new transmission. A break can be sent as soon as the error is detected.

If the same command fails many times in a row, that command is probably not supported by SB201 or something serious is wrong. It might be that the frequency of the clock on the host or SB201 has changed too much or maybe that SB201 have shut itself down.

15 EVALUATION BOARD/KIT IMPORTANT NOTICE

This evaluation board/kit is intended for use for **FURTHER ENGINEERING, DEVELOPMENT, DEMONSTRATION, OR EVALUATION PURPOSES ONLY.** It is not a finished product and may not (yet) comply with some or any technical or legal requirements that are applicable to finished products, including, without limitation, directives regarding electromagnetic compatibility, recycling (WEEE), FCC, CE or UL (except as may be otherwise noted on the board/kit). Atmel supplied this board/kit "AS IS," without any warranties, with all faults, at the buyer's and further users' sole risk. The user assumes all responsibility and liability for proper and safe handling of the goods. Further, the user indemnifies Atmel from all claims arising from the handling or use of the goods. Due to the open construction of the product, it is the user's responsibility to take any and all appropriate precautions with regard to electrostatic discharge and any other technical or legal concerns.

EXCEPT TO THE EXTENT OF THE INDEMNITY SET FORTH ABOVE, NEITHER USER NOR ATMEL SHALL BE LIABLE TO EACH OTHER FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES.

No license is granted under any patent right or other intellectual property right of Atmel covering or relating to any machine, process, or combination in which such Atmel products or services might be or are used.

Mailing Address: Atmel Corporation, 2325 Orchard Parkway, San Jose, CA 95131

Copyright © 2008, Atmel Corporation



Headquarters

Atmel Corporation
2325 Orchard Parkway
San Jose, CA 95131
USA
Tel: 1(408) 441-0311
Fax: 1(408) 487-2600

International

Atmel Asia
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimshatsui
East Kowloon
Hong Kong
Tel: (852) 2721-9778
Fax: (852) 2722-1369

Atmel Europe
Le Krebs
8, Rue Jean-Pierre Timbaud
BP 309
78054 Saint-Quentin-en-
Yvelines Cedex
France
Tel: (33) 1-30-60-70-00
Fax: (33) 1-30-60-71-11

Atmel Japan
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
Tel: (81) 3-3523-3551
Fax: (81) 3-3523-7581

Product Contact

Web Site
www.atmel.com

Technical Support
avr@atmel.com

Sales Contact
www.atmel.com/contacts

Literature Request
www.atmel.com/literature

Disclaimer: The information in this document is provided in connection with Atmel products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Atmel products. **EXCEPT AS SET FORTH IN ATMEL'S TERMS AND CONDITIONS OF SALE LOCATED ON ATMEL'S WEB SITE, ATMEL ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL ATMEL BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION, OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF ATMEL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.** Atmel makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice. Atmel does not make any commitment to update the information contained herein. Unless specifically provided otherwise, Atmel products are not suitable for, and shall not be used in, automotive applications. Atmel's products are not intended, authorized, or warranted for use as components in applications intended to support or sustain life.

© **2008 Atmel Corporation. All rights reserved**. Atmel®, logo and combinations thereof, AVR® and others, are the registered trademarks or trademarks of Atmel Corporation or its subsidiaries. Other terms and product names may be trademarks of others.