Smart Battery Data Specification
Proposal for Update/Revision to V1.1

Contents:

SpecificationInfo() **(0x1a)**

Original: From Version 1.0 SBData Specification

Field	Bits Used	Format	Allowable Values
Revision	0...3	4 bit binary value	0 – 15
Version	4...7	4 bit binary value	1 – 15
VScale	8...11	4 bit binary value	0 - 3 (multiplies voltages* by 10 ^ VScale)
IPScale	12...15	4 bit binary value	0 - 3 (multiplies currents* and capacities by 10 ^ IPScale)

*Note: Except ChargingVoltage() and ChargingCurrent() values.

Example: The specification version supported by a particular battery is 1.0 and all current readings are to be scaled by a factor of 10. Power readings will be scaled by the voltage factor times the current factor (10^0 * 10 ^ 1) or 10 in this case. SpecificationInfo() will return 4112 (0x1010).

As Issued (Error): Version 1.1 SBData Specification

Field	Bits Used	Format	Allowable Values
Revision	0...3	4 bit binary value	Spec 1.0 and 1.1 = 0x1
Version	4...7	4 bit binary value	Spec 1.0 = 0x1 Spec 1.1 without optional PEC support = 0x2 Spec 1.1 with optional PEC support= 0x3
VScale	8...11	4 bit binary value	0 - 3 (multiplies voltages* by 10 ^ VScale)
IPScale	12...15	4 bit binary value	0 - 3 (multiplies currents* and capacities by 10 ^ IPScale)

*Note: Except ChargingVoltage() and ChargingCurrent() values.

Example: The specification version supported by a particular battery is 1.0 and all current readings are to be scaled by a factor of 10. Power readings will be scaled by the voltage factor times the current factor (10^0 * 10 ^ 1) or 10 in this case. SpecificationInfo() will return 4112 (0x1010).

NOTE: The above example is incorrect as written. According to the definition above, SpecInfo should be 0x1011.

Proposed Correction: Version 1.1a SBData Specification

5.1.25 SpecificationInfo() (0x1a)

Description:

Returns the version number of the Smart Battery specification the battery pack supports, as well as voltage and current **and capacity** scaling information in a packed unsigned integer. Power scaling is the product of the voltage scaling times the current scaling.

These scaling functions do NOT affect ChargingCurrent() and ChargingVoltage() values.

A Smart Battery Charger cannot be assumed to know this scaling information. (However, a 'Level 3' or 'Host Controlled' Smart Battery Charger may read this value if required for specific applications.)

This value may also indicate a version of SMBus error checking implementation. Refer to the SMBus Specification for actual implementation information.

The SpecificationInfo is packed in the following fashion: (~~major version number~~SpecID_H * 0x10 + ~~minor revision~~SpecID_L ~~number~~) + (~~voltage scaling~~ VScale + ~~current scaling~~IPScale * 0x10) * 0x100.

Purpose:

The SpecificationInfo() function is used by the SMBus Host's power management system to determine what information the Smart Battery can provide. It can be used by Smart Battery Systems where the defined 16-bit data values do not provide enough range for higher power applications.

SMBus Protocol: Read Word

Output: unsigned int -- packed specification number and scaling information

Field	Bits Used	Format	Allowable Values
~~Revision~~ SpecID_L	0...3	4 bit binary value	Spec 1.0 ~~and 1.1 = 0x1~~ 0x0 Spec 1.1 = 0x1
~~Version~~ SpecID_H	4...7	4 bit binary value	Spec 1.0 = 0x1 Spec 1.1 without optional PEC support = 0x2 Spec 1.1 with optional PEC support= 0x3
VScale	8...11	4 bit binary value	0 - 3 (multiplies voltages* by 10 ^ VScale)
IPScale	12...15	4 bit binary value	0 - 3 (multiplies currents* and capacities by 10 ^ IPScale)

*Note: Except ChargingVoltage() and ChargingCurrent() values.

Example 1:

The specification version supported by a particular battery is 1.0 and all current readings are to be scaled by a factor of 10. Power readings will be scaled by the voltage factor times the current factor (10^0 * 10 ^ 1) or 10 in this case. SpecificationInfo() will return 0x1010.

15							MSB 8	7							LSB 0
0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0

Example 2:

The specification version supported by a particular battery is 1.1, this battery does not support PEC and all current readings are to be scaled by a factor of 10. Power readings will be scaled by the voltage factor times the current factor (10^0 * 10 ^ 1) or 10 in this case. SpecificationInfo() will return ~~0x1012~~0x1021.

15							MSB 8	7							LSB 0
0	0	0	1	0	0	0	0	0	0	1~~0~~	0~~1~~	0	0	0~~1~~	1~~0~~

Example 3:

The specification version supported by a particular battery is 1.1, this battery supports PEC and all current and voltage readings are to be scaled by a factor of 10. Power readings will be scaled by the voltage factor times the current factor (10^1 * 10 ^ 1) or 100 in this case. SpecificationInfo() will return ~~0x1113~~0x1131.

15							MSB 8	7							LSB 0
0	0	0	1	0	0	0	1	0	0	1~~0~~	1	0	0	0~~1~~	1

BatteryMode() (0x03)

Description:
This function selects the various battery operational modes and reports the battery's capabilities, modes, and flags minor conditions requiring attention.

Defined capabilities include:

- Internal charge controller supported (INTERNAL_CHARGE_CONTROLLER bit)
- Internal primary battery control supported (PRIMARY_BATTERY_SUPPORT bit)
  Note: These capabilities listed are optional but their indicating flag bits must be supported.

Defined modes include:

- Battery's capacity information is specified to be reported in either mAh or 10 mWh (CAPACITY_MODE bit)
- Whether the ChargingCurrent() and ChargingVoltage() values are to be broadcast to the Smart Battery Charger when the Smart Battery requires charging (CHARGER_MODE bit)
- Internal charge controller enable (CHARGE_CONTROLLER_ENABLED bit) [Optional]
- Internal primary battery control enable (PRIMARY_BATTERY bit) [Optional]

Defined conditions requiring attention include:

- Battery requesting a ~~conditioning~~ capacity re-learn cycle (~~CONDITION~~RELEARN_FLAG bit)

Purpose:
To allow configuration of the Smart Battery for particular application requirements. (See individual bit definitions which follow.)
SMBus Protocol: Read or Write Word
Input/Output: unsigned int - bit mapped - see below

Units:	not applicable
Range:	0...1
Granularity:	not applicable
Accuracy:	not applicable

The BatteryMode() word is divided into two halves, the Most Significant Byte (MSB) which is read/write and the Least Significant Byte (LSB) which is read only. Attempts to set (write 1's) the reserved bits in the MSB are prohibited.

15			MSB				8	7			LSB				0
R/W	R/W	R/W	res	res	res	R/W	R/W	R	res	res	res	res	res	R	R

The following table summarizes the meanings of the individual bits in the BatteryMode() word and specifies the default values if any. Power-on default values, where applicable, are discussed in section 4.4. More detailed explanations can be found in the listing following the table below.

Field	Bits Used	Format	Allowable Values
INTERNAL_CHARGE_ CONTROLLER	0	read only bit flag	0 - Function Not Supported 1 - Internal Charge Controller Supported
PRIMARY_BATTERY_ SUPPORT	1	read only bit flag	0 - Function Not Supported 1 - Primary or Secondary Battery Support
Reserved	2-6		Undefined
~~CONDITION~~RELEARN _FLAG	7	read only bit flag	0 - Battery OK 1 - ~~Conditioning~~ Capacity Re-Learn Cycle Requested
CHARGE_CONTROLLER _ENABLED	8	r/w bit flag	0 - Internal Charge Control Disabled (default) 1 - Internal Charge Control Enabled
PRIMARY_BATTERY	9	r/w bit flag	0 - Battery operating in its secondary role (default) 1 - Battery operating in its primary role
Reserved	10-12		Undefined
ALARM_MODE	13	r/w bit flag	0 - Enable AlarmWarning broadcasts to Host and Smart Battery Charger (default) 1 - Disable AlarmWarning broadcast to Host and Smart Battery Charger
CHARGER_MODE	14	r/w bit flag	0 - Enable ChargingVoltage and ChargingCurrent broadcasts to Smart Battery Charger (default) 1 - Disable broadcasts of ChargingVoltage and ChargingCurrent to Smart Battery Charger
CAPACITY_MODE	15	r/w bit flag	0 - Report in mA or mAh (default) 1 - Report in 10mW or 10mWh

(Note: 'Reserved' bits are not defined and are intended for use in future revisions of the specification, therefore, their use for other purposes is not allowed.)

Specific Definitions for each bit flag condition are listed below:

INTERNAL_CHARGE_CONTROLLER bit set indicates that the battery pack contains its own internal charge controller. When the bit is set, this optional function is supported and the CHARGE_CONTROLLER_ENABLED bit will be available for activation and control of the actual internal charger.

The definition of an Internal Charge Controller is a device which accepts power from the battery terminals but may regulate or otherwise control the current and voltage that actually reaches the battery's cells. The INTERNAL_CHARGE_CONTROLLER bit simply indicates the presence of the internal charger while the CHARGE_CONTROLLER_ENABLED bit actually controls the on/off state of this internal charger. (See 'Examples' following this section.)

PRIMARY_BATTERY_SUPPORT bit set indicates that the battery pack has the ability to act as either the primary or secondary battery in a system. When the bit is set, this function is supported and the PRIMARY_BATTERY bit will be available for activation and control of this function.

The Primary/Secondary battery feature is used with batteries containing internal discharge control mechanisms to allow multiple batteries to be connected in parallel. The PRIMARY_BATTERY_SUPPORT bit simply indicates the presence of this internal control while the PRIMARY_BATTERY bit actually controls the on/off state of this internal control. (See 'Examples' following this section.)

~~**CONDITION**~~**RELEARN_FLAG** bit set indicates that the battery is requesting a ~~conditioning~~ capacity re-learn cycle. A ~~conditioning~~ capacity re-learn cycle may be requested because of the characteristics of the battery chemistry and/or the electronics in combination with the usage pattern.

The ~~CONDITION~~RELEARN_FLAG is the first signal from the Smart Battery that it has limited ability to determine the present state-of-charge. As a result other data values may be less accurate than required by this specification.

(A more serious flag is the INITIALIZED status bit flag found in the BatteryStatus() register. Refer to Section 5.1.21 for the BatteryStatus() register.)

Status Flag	Location	Smart Battery Performance	Action Required
~~CONDITION~~RELEARN_FLAG=1	BatteryMode() Bit 7	Useable, Safe, Reliable, but less accurate	Perform ~~Condition~~ Capacity Re-learn Cycle (see text)
INITIALIZED=0	BatteryStatus() Bit 7	Useable, Safe, but use data with caution (less reliable)	See User Manual

When the ~~CONDITION~~RELEARN_FLAG is set, the Smart Battery is still fully functional, reliable, and safe. However, the System Host may represent to the user that a ~~condition~~ capacity re-learn cycle should be performed as soon as possible to return the Smart Battery to full accuracy. While the ~~CONDITION~~RELEARN_FLAG is set, the Smart Battery Data values should be used with more tolerance.

The ~~condition~~ capacity re-learn cycle is pack specific, but typically will consist of a full-charge, full-discharge, and repeated to full-charge of the battery pack. The Smart Battery electronics will clear this flag after it detects that a ~~condition~~ capacity re-learn cycle has been completed. Refer to the Smart Battery electronics' supplier documentation for specific ~~conditioning~~ capacity re-learn cycle procedures required.

NOTE: Please refer to the INITIALIZED status bit flag in the BatteryStatus() register in Section 5.1.21 for a more detailed definition.

References to the "CONDITION_FLAG" will be changed to "RELEARN_FLAG" in the description of the "INITIALIZED" status flag in both the AlarmWarning() function (Section 5.4.1) and the BatteryStatus() function (Section 5.1.21). Mention of the "CONDITION_FLAG" in the description for the MaxError() function (Section 5.1.13) will also be updated to "RELEARN_FLAG".