CANBus Specification

WST CANBUS SPECIFICATIONS – REV. 4.5

Intro

This specification outlines new functionality, that has been added to our firmware. Please enquire with us to learn which firmware version you need for your battery, to enable the new features.

The new functionality is an addition to the existing protocol, so we maintain backwards compatibility.

The new functionality will be referred to as Protocol 2, while the old functionality will be referred to as Protocol 1.

Please notice that the BMS will sleep to conserve power. The time before sleep varies with different firmware's usually 2-5 min.
Therefore, before doing any communication with the batteries, please ensure to "wake" them up by transmitting a frame like this:

	Description	ID	Data
MASTER	Wake up batteries	0x004	[0x00]

If this id is used by other nodes on the bus, another one can be chosen. Please use an ID that is not used for anything else on bus.

250kb/s is the default bitrate, we use 11bit frame ID's and we do not use remote frames.

Big endian

Unless specified, all multi-byte values are packed in big-endian.

Protocol 1

Get realtime data

The following data is available over CANBus and can be retrieved by sending a frame with the ID's seen below. The ID's can be changed by flashing the appropriate firmware or using protocol 2, if it is available. Assigning node_id's is relevant if communication with multiple packs on the same bus is desired.

#	Frame ID	Data Length	Data Description	Offset	Unit	Min value	Max value
1	0x201	1 frame - 8 bytes	Pack Voltage	byte 0 - 1	0.1V	0	65535
			Charge-Current	byte 2 - 3	0.1A	0	65535

WS Technicals A/S
Gjellerupvej 89A
8230 Åbyhøj
Denmark

+45 88 61 83 88
wstech@wstech.dk
CVR.: 37290785

wstech.dk

			Discharge-Current	byte 4 - 5	0.1A	0	65535
			SOC*	byte 6	1%	0	100
			Remaining Time to Full-Charge*	byte 7	0.1h	0	255
2	0x202	1 frame - 8 bytes	Remaining Capacity*	byte 0 - 1	1mAh or 10mAh if design capacity > 65000mAh	0	65535
			SOH*	byte 2		0	100
			Firmware Version	byte 3	0.1	0	255 (25.5)
			Full Capacity	byte 4 - 5	1mAh or 10mAh if design capacity > 65000mAh	0	65535
			Cycle Count	byte 6 - 7	1 cycle	0	65535
3	0x203	1 frame - 8 bytes	BMS Current Status 0x0001 discharge (bit 0) 0x0002 charge (bit 1) 0x0004 OV (bit 2) 0x0008 UV (bit 3) 0x0010 COC(bit 4) 0x0020 DOC (bit 5) 0x0040 DOT (bit 6) 0x0080 DUT (bit 7) 0x0200 SC (bit 9) 0x0400 COT(bit 10) 0x0800 CUT (bit 11)	Byte 0-1			
			BMS Temperature 1 is 1°C 0 is 0°C 255 is -1°C	Byte 2	1℃	-40℃	120℃
			N/A	Byte 3-7			

#	Frame ID	Data Length	Data Description	Offset	Unit	Min value	Max value
4	0x204	1 frame - 8 bytes	Cell1 Voltage	byte 0 - 1	1mv	0	4500
			Cell2 Voltage	byte 2 - 3	1mv	0	4500
			Cell3 Voltage	byte 4 - 5	1mv	0	4500
			Cell4 Voltage	byte 6 - 7	1mv	0	4500
5	0x205	1 frame - 8 bytes	Cell5 Voltage	byte 0 - 1	1mv	0	4500
			Cell6 Voltage	byte 2 - 3	1mv	0	4500
			Cell7 Voltage	byte 4 - 5	1mv	0	4500
			Cell8 Voltage	byte 6 - 7			
6-9	0x206		Cell 9 to Cell 24		1mv	0	4500

	- 0x209		Number of cells depend on the configuration of the battery.				
A	0x20A		N/A				
F	0x20F	Multiple frames – 8 byte each	Retrieve Log See further down for specifications				

Description of calculated values

Cycle

The BMS will increase he cyclecount when it detects that the Battery has been discharged for a total Ah equaling the current full-capacity value.

SoC

The SoC is the ratio between remaining capacity and the full-capacity.
The BMS uses two different methods for calculating the State of charge. If it has had time to gather enough data via the internal shunt, it will use this to calculate the SoC. If not, a simple voltage estimation will be used.

Remaining Capacity

The remaining capacity, or rem-cap, is the expected remaining capacity in the battery before UV protection kicks in. See SoC above.

SoH

SoH is calculated as full-cap/design-cap.

Design-cap is hardcoded, and full-cap is the current maximum capacity. The batteries will degrade over time, and this is reflected in full-cap. If the BMS sees that it is charged up to OV protection turns on and then discharged down to UV protection turns on, then it concludes that it has seen the new full-cap. on the battery and therefore updates this value.

Subsequently, full-cap is downgraded by 0.00025% for each cycle performed. It does not matter if the battery is repeatedly charged/discharged for only a fraction of its capacity – a cycle will be registered when the total discharged capacity since last cycle, is the same as the current full-cap. This compensates if the battery is never discharged and charged completely.

If the full-cap is never updated by a full charge and discharge cycle, then after 1000 cycles there will be: (1-0,00025) ^ 1000 = 0.779... = 78% SoH remaining. This is a slightly larger loss than we see in our cycle tests, but we feel it is better to be on the conservative side, when we do estimations.

If a full cycle, up to OV and down to UV, is again seen, then the full-cap will be updated.

Logged events

Some of the "events" are actually proxy events that are shown as states (state 1 - 3) together with an event byte, which can be considered as the "main" event.

Example:
When a charge OT "event" occurs, a "Turn off CHG-Mosfet event" will be triggered by the BMS and the Charge OT will be shown as a change in the state3 byte (0x80), in a response where the event byte set to 0x09(C-FET Off).

Log data specifications

When the log is requested the BMS will send a multiframe response with each frame organized like below.

Frame #	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
1	SOI (0xEA)	CID (0xD1)	ADR (0x01)	Data Length (eg 0x25)	0xFF	0x08	Record # (0x01 - 0x72)	Data 0
2	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	Data 7	Data 8
3	Data 9	Data 10	Data 11	Data 12	Data 13	Data 14	Data 15	Data 16
4	Data 17	Data 18	Data 19	Data 20	Data 21	Data 22	Data 23	Data 24
5	Data 25	Data 26	Data 27	Data 28	Data 29	Data 30	Data 31	Checksum
6	EOI(0xF5)	0x00	0x00	0x00	0x00	0x00	0x00	0x00

Data length in bytes is counted like this: all data bytes + byte 3-6 in frame 1 + checksum from frame 5.
For the current firmware this results in: 0x25 or $(37)_{10}$
Checksum: Frame1-Byte3^Frame1-Byte4^...^Frame5-Byte6 (Bitwise XOR)

The following frame will be sent as the last frame, to show the complete log has been sent:

Frame #	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
END	SOI (0xEA)	CID (0xD1)	ADR (0x01)	Data Length (0x04)	0xFF	0xFE	Checksum	0xF5

The checksum for this frame is calculated like this: Byte3^Byte 4^Byte5 (Bitwise XOR)

Datalog description

Please refer to the description of the data bytes 0-31 in protocol 2, as they are the same in the two protocols.

Protocol 2

In the new protocol we always communicate to the battery on frame id 0x00E, and the battery always responds on frame id 0x00D.

The protocol introduces the concept of a programmable NODE_ID, that is used to target the batteries, when sending commands.

The NODE_ID's are via a set of commands which rely on the battery's serial number. The serial number is stored inside the battery and persists.

Below is an example of the process, where we have 2 batteries attached to the bus, with the following serial numbers:

BATTERY #1 serial number: 001122
BATTERY #2 serial number: 112233

In the following, we reference the individual bytes in the frame data like this: [Byte 0, Byte 1, ..., Byte 7] and in short form: [B0, B1, ..., B7]

Get serials

First send an empty 0x001 frame to wake up the BUS.
Then we can get the serial numbers from all the attached batteries like below.

	Description	ID	Data
MASTER	Get Serials frame	0x00E	[0x02, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00]
BATTERY #1	Serial Response	0x00D	[0x02, 0x06, 0x00, 0x11, 0x22, 0xFF, 0xFF, 0xFF]
BATTERY #2	Serial Response	0x00D	[0x02, 0x06, 0x11, 0x22, 0x33, 0xFF, 0xFF, 0xFF]

Where [B0 = cmd, B1 = length of serial number (hex chars/nibble), B2 to B4 = Serial number, B5-B7 = N/A]

The BATTERY's will send their response within 1000ms, at a random delay as to prevent collisions on the BUS. The **MASTER** is expected to know how many batteries are attached.

Set Node id's

To program the NODE_ID on the two batteries, we send the desired NODE_ID and the serial number of the battery we wish to program. In the below example we choose NODE_ID 10 for **BATTERY #1**, and NODE_ID 20 for **BATTERY #2**.

	Description	ID	Data
MASTER	Assign NODE_ID 10 to **BATTERY #1**	0x00E	[0x03, 0x0A, 0x06, 0x00, 0x11, 0x22, 0xFF, 0xFF]
MASTER	Assign NODE_ID 20 to **BATTERY #2**	0x00E	[0x03, 0x14, 0x06, 0x11, 0x22, 0x33, 0xFF, 0xFF]

Where [B0 = cmd, B1 = NODE_ID, B2 = length of serial (max. 10 hex chars), B3 to B7 = serial number]

BMS's will reply with a confirmation, when a node_id is successfully assigned:

	Description	ID	Data
BATTERY #1	Confirm NODE_ID 10 assigned	0x00D	[0x0A, 0x03, 0x06, 0x00, 0x11, 0x22, 0xFF, 0xFF]
BATTERY #2	Confirm NODE_ID 20 assigned	0x00D	[0x14, 0x03, 0x06, 0x11, 0x22, 0x33, 0xFF, 0xFF]

Notice that NODE_ID 2-7 are treated specially. When a battery has a one of these NODE_ID's, it will also reply to the old 0x201 style frames from protocol 1.

E.g. If the NODE_ID is 7, you can get status from the battery on frame id's in the range 0x701 - 0x70F.

When a NODE_ID has been programmed, it will persist in the battery until changed.

Get status

When a battery has had a NODE_ID assigned, we can begin sending commands to it.

To get the status from **BATTERY #1**, which has NODE_ID 10, we send the following:

	Description	ID	Data
MASTER	Request status from NODE_ID=10	0x00E	[0x01, 0x0A, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01]

Where [B0 = standard-cmd-family, B1 = NODE_ID, B2 to B5 = N/A, B6 + B7 = Get status cmd]

	Description	ID	Data
BATTERY #1	Multi frame response	0x00D	See Below for explanation

Frame	ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0	0x00d	0x0A	0x00	0x01	0x13	NA	NA	NA	0x00
1	0x00d	0x0A	0x60	Data 0	Data 1	Data 2	Data 3	Data 4	0x01
2	0x00d	0x0A	Data 5	Data 6	Data 7	Data 8	Data 9	Data 10	0x02
3	0x00d	0x0A	Data 11	Data 12	Data 13	Data 14	Data 15	Data 16	0x03
4	0x00d	0x0A	Data 17	Data 18	Data 19	Data 20	Data 21	Data 22	0x04
5	0x00d	0x0A	Data 23	Data 24	Data 25	Data 26	Data 27	Data 28	0x05
6	0x00d	0x0A	Data 29	Data 30	Data 31	Data 32	Data 33	Data 34	0x06
7	0x00d	0x0A	Data 35	Data 36	Data 37	Data 38	Data 39	Data 40	0x07
8	0x00d	0x0A	Data 41	Data 42	Data 43	Data 44	Data 45	Data 46	0x08
9	0x00d	0x0A	Data 47	Data 48	Data 49	Data 50	Data 51	Data 52	0x09
10	0x00d	0x0A	Data 53	Data 54	Data 55	Data 56	Data 57	Data 58	0x0A
11	0x00d	0x0A	Data 59	Data 60	Data 61	Data 62	Data 63	Data 64	0x0B
12	0x00d	0x0A	Data 65	Data 66	Data 67	Data 68	Data 69	Data 70	0x0C
13	0x00d	0x0A	Data 71	Data 72	Data 73	Data 74	Data 75	Data 76	0x0D
14	0x00d	0x0A	Data 77	Data 78	Data 79	Data 80	Data 81	Data 82	0x0E

15	0x00d	0x0A	Data 83	Data 84	Data 85	Data 86	Data 87	Data 88	0x0F
16	0x00d	0x0A	Data 89	Data 90	Data 91	Data 92	Data 93	Data 94	0x10
17	0x00d	0x0A	Data 95	NA	NA	NA	NA	NA	0x11
18	0x00d	0x0A	0xFF	0xFF	0x60	0xFE	0xFF	0xFF	0x12

Frame 0, Byte 0: NODE_ID of Battery (Same for all frames)
Frame 0, Bytes 1-2: Command that Battery is responding to
Frame 0, Byte 3: Number of frames in the response. In the example above, there are 19 frames (0x13)
Frame 0, Bytes 4-6: Not Used
Frame 0, Byte 7: The frames number in the package.
Frame 1, Byte 1: Total length of the data. (data 0 to data 95 = 96 data frames (0x60))
Frame 17, Bytes 1,2,4,5 and 6 are the termination frame special bytes. They should always be 0xFF.
Frame 17, Byte 3: length of data part (Same as Frame 1, Byte 1)

Data

Data Byte #	Data Description	Unit	Additional information	Min value	Max value
0-1	Pack Voltage	0.1V		0	65535
2-3	Charge-Current	0.1A		0	65535
4-5	Discharge-Current	0.1A		0	65535
6	SOC	1%		0	100
7	Remaining Time to Full-Charge	0.1h		0	255
8-9	Remaining Capacity	1mAh	If design capacity > 65000mAh the unit changes to 10mAh	0	65535
10	SOH	1%		0	100
11	Firmware Version	0.1		0	255 (25.5)
12-13	Full Capacity	1mah	If design capacity > 65000mAh the unit changes to 10mAh	0	65535
14-15	Cycle Count	1 Cycle		0	65535
16-17	BMS Status		Bit flags, multiple can happen at the same time		
	0x0001 discharge (bit 0)				
	0x0002 charge (bit 1)				
	0x0004 OV (bit 2)				
	0x0008 UV (bit 3)				
	0x0010 COC(bit 4)				
	0x0020 DOC (bit 5)				
	0x0040 DOT (bit 6)				
	0x0080 DUT (bit 7)				
	0x0200 SC (bit 9)				
	0x0400 COT(bit 10)				
	0x0800 CUT (bit 11)				
18	BMS Temperature	1℃	1 is 1°C; 0 is 0°C; 255 is -1°C (default cell 1)	-40℃	120℃
19	BMS Temperature		(default cell 2)		
20-21	N/A				
22	BMS Temperature		(default FETS)		
23	BMS Temperature		(default ambient)		
24-25	Cell 1 Voltage	1mv		0	4500
26-71	Cell 2 to 24 cell voltages	1mv		0	4500
72-77	N/A				
78	N/A				
79	N/A				
80	Serial number length	#	Single Hex chars	1	10

81	Serial char 1&2	Hex Chars	Example If the serial is 123456, then the length is 6 and byte 81 is 0x12, byte 82 is 0x34 and byte 83 is 0x56		
82	Serial char 3&4				
83	Serial char 5&6				
84	Serial char 7&8				
85	Serial char 9&10				
86-95	N/A				

Relation to Protocol 1

	Protocol 1 frame ID	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
0	0x201	data 0	data 1	data 2	data 3	data 4	data 5	data 6	data 7
1	0x202	data 8	data 9	data 10	data 11	data 12	data 13	data 14	data 15
2	0x203	data 16	data 17	data 18	data 19	data 20	data 21	data 22	data 23
3	0x204	data 24	data 25	data 26	data 27	data 28	data 29	data 30	data 31
4	0x205	data 32	data 33	data 34	data 35	data 36	data 37	data 38	data 39
5	0x206	data 40	data 41	data 42	data 43	data 44	data 45	data 46	data 47
6	0x207	data 48	data 49	data 50	data 51	data 52	data 53	data 54	data 55
7	0x208	data 56	data 57	data 58	data 59	data 60	data 61	data 62	data 63
8	0x209	data 64	data 65	data 66	data 67	data 68	data 69	data 70	data 71
9	0x20A	data 72	data 73	data 74	data 75	data 76	data 77	data 78	data 79
10	0x20B	data 80	data 81	data 82	data 83	data 84	data 85	data 86	data 87
11	Reserved	data 88	data 89	data 90	data 91	data 92	data 93	data 94	data 95

Get Log

Like in the old protocol 1, we can retrieve the internal log from the battery over CANbus using the new protocol.

To do this we send the following frame,

	Description	ID	Data
MASTER	Request log from NODE_ID=10	0x00E	[0x04, 0x0A, 0x00, 0x00, 0x00, 0x00, 0x01, 0x01]

Where [B0 = log-cmd-family, B1 = NODE_ID, B2 to B5 = N/A, B6 + B7 = Get log cmd]

When a battery with the appropriate NODE_ID, gets this frame, it will respond by transmitting all the records in the log, where each record is split into 8 frames.

Frame	ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0	0x00d	0x04	CMD – 0x01	0x01	NODE_ID – 0x0A	FRAMES – 0x08	RECORD #	Total Records	0x00
1	0x00d	0x04	DATA LENGTH	Data 0	Data 1	Data 2	Data 3	Data 4	0x01
2	0x00d	0x04	Data 5	Data 6	Data 7	Data 8	Data 9	Data 10	0x02

3	0x00d	0x04	Data 11	Data 12	Data 13	Data 14	Data 15	Data 16	0x03
4	0x00d	0x04	Data 17	Data 18	Data 19	Data 20	Data 21	Data 22	0x04
5	0x00d	0x04	Data 23	Data 24	Data 25	Data 26	Data 27	Data 28	0x05
6	0x00d	0x04	Data 29	Data 30	Data 31	XOR CRC	NA - 0x00	NA - 0x00	0x06
7	0x00d	0x04	0xFF	0xFF	DATA LENGTH	RECORD #	0xFF	0xFF	0x07

Frame 0-7, Byte 0: LOG COMMAND FAMILY
Frame 0, Bytes 1-2: Command that the battery is responding to.
Frame 0, Byte 3: NODE_ID of the responding battery
Frame 0, Byte 4: Number of frames in the bundle for one record.
Frame 0, Byte 5: The records number – Same as Frame 7, Byte 4
Frame 0, Byte 6: The total number of records.
Frame 0, Byte 7: The frames number in the bundle
Frame 1, Byte 1: Total length of the data. (Data 0 to Data 31 = 32 bytes of data) – Same as Frame 7, Byte 3
Frame 6, Byte 4: XOR of Data 0 to Data 31
Frame 7, Bytes 1,2, 5 and 6 are the termination special bytes. They should always be 0xFF.

Please note that the DATA bytes are identical to the data that can be retrieved with the old protocol 1.

Below is a legend of the data, events and states

Data Byte	Description	Unit/notes
0	Year	Bit 7-4: tens
		Bit 3-0: ones
1	Month	Bit 4: tens
		Bit 3-0: ones
2	Day	Bit 5-4: tens
		Bit 3-0: ones
3	Hour	Bit 6-4: tens
		Bit 3-0: ones
4	Minute	Bit 6-4: tens
		Bit 3-0: ones
5	Second	Bit 6-4: tens
		Bit 3-0: ones
6	Pack V High Byte	10mV
7	Pack V Low Byte	-
8	Min Cell V High Byte	1mV
9	Min Cell V Low Byte	-
10	Max Cell V High Byte	1mV
11	Max Cell V Low Byte	-
12	Current High Byte	10mA
13	Current Low Byte	-
14	Max temperature	Celsius – offset by 40
15	Min Temperature	Celsius – offset by 40
16	SOC	Percent
17	Rem. Cap Byte 1 (highest)	1mAh
18	Rem. Cap byte 2	-
19	Rem. Cap byte 3	-
20	Rem. Cap byte 4 (lowest)	-
21	Cyclecount High Byte	
22	Cyclecount Low Byte	
23	State 1	See below
24	State 2	See below
25	State 3	See below
26	Charge/Discharge Flag	0x20: standby
		0x40: discharge
		0x80: charge
27	Event	See below
28	SOH	Percent
29	N/A	
30	N/A	
31	N/A	

Event	Description
0x03	UV Shutdown
0x04	Power Up
0x06	Full charge capacity update
0x07	Cycle count update
0x08	D-FET OFF
0x09	C-FET OFF
0x0A	D-FET ON
0x0B	C-FET ON
0x0C	Parameter Update
0x0D	Charge-Current Calibration
0x0E	Discharge-Current Calibration
0x0F	Voltage Calibration
0x20	Voltage Failure

0x23	Charging Start
0x24	Charging Stop
0x27	Discharge Begin
0x28	Discharge Stop
0x32	NA
0x33	NA
0x34	15s delayed current logging

State 1	**Code**	**Description**
	0x01	Pack UV Recovery
	0x02	Cell UV Recovery
	0x04	Pack OV Recovery
	0x08	Cell OV Recovery
	0x10	Pack UV
	0x20	Cell UV
	0x40	Pack OV
	0x80	Cell OV
State 2	**Code**	**Description**
	0x04	SC Recovery
	0x08	DOC Recovery
	0x10	COC Recovery
	0x20	SC
	0x40	DOC
	0x80	COC
State 3	**Code**	**Description**
	0x10	DOT Recovery
	0x20	COT Recovery
	0x40	DOT
	0x80	COT

Legend

UV: Under voltage
OV: Over voltage
DOC: Discharge Overcurrent
COC: Charge Over Current
SC: Short circuit
DOT: Discharge Over Temp
DUT: Discharge Under Temp.
COT: Charge Over Temp
CUT: Charge Under Temp.