SDAQ measurement system, CAN-protocol specification

Document history

24.6.2019	Initial revision
28.7.2019	timestamp range changed from 65536 to 60000 (0…59999)
	bootloader commands added
7.8.2019	calibration related commands added
5.9.2019	Set Device Address (0x06) command added
18.9.2019	Added CAN config cmd (0x0b)
20.10.2019	Added payload types: 0x0c, 0x8b
4.11.2019	Added commands for System Variables (0x8d, 0x0d, 0x0e)
18.11.2019	Device info message: added max number of calib. Points
	Calibrated unit added to Calibration Date –message/command
	Calibration Point Data, added coefficients for polynomials
	Write Calibration Data renamed to Write Calibration Date
26.11.2019	Measurement units added.
2.12.2019	Calibration date format changed and calibration period added.
3.12.2019	Added “Out of calibrated range” bit to the measurement value (0x84) status byte
24.3.2020	Added “Overrange” bit to the measurement value (0x84) status byte
18.1.2021	Added firmware revision history
11.2.2021	Added bootloader commands
	Added list of systems variables
	Removed “Query flash data” command
	Added note to “Calib Point Data” about ordering of points
8.2.2022	Added extended version of Id/Status message
19.12.2023	Added description for LPC5504 bootloader (newer devices)

Basic operation

1. Power-up
2. Devices starts sending ID/status messages
3. Bus-master acquires ID/status messages and queries additional device info (device type, number of channels, serial number, calibration date…)
4. Bus-master enables measurement data streams by sending start-command to the devices
5. Devices streams measurement data until power-off or received stop-command

Optionally bus-master can synchronize measurements on all devices by sending sync-command. Synchronization command should be sent periodically to avoid clock drifting on bus-devices.

CAN 29 bit extended identifier usage

500 kbps (1000 kbps optional)

Priority (0..7)
protocol id (6 bits) = 0b110101 = 0x35
payload type (8 bits)
device address (0-63, 0=broadcast)
channel number (0-63)

	Priority			Protocol ID						Payload type								Device address						Channel number					
	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Payload type, messages from devices

ID	Name	Desc.
0x84	Measurement value	Send only when device is in running state
0x86	Device ID/status	Sent periodically (=20 sec), contains device type, serial & status, reply to 0x07 command
0x88	Device Info	Reply to 0x07 command
0x89	Calibration Date	Reply to 0x07 / 0x08 commands
0x8a	Calibration Point Data	
0x8b	Uncalibrated meas. value	
0x8d	System variable	(internal calibration, etc)
Bootloader replies		
0xa0	Bootloader Reply	
0xa1	Page Buffer Data	
Debug Data		
0xc0	Sync Info	
	raw measurement	
	debug data	
	calibration data	

Payload type, commands to devices (bit7=0)

ID	Name	Desc.
0x01	Synchronization command	
0x02	Start	Move to running state (starts streaming meas. data)
0x03	Stop	Move to standby mode (no meas. streaming)
0x06	Set Device Address	
0x07	Query Device Info	Device replies with messages 0x86, 0x88, 0x89
0x08	Query Calibration Data	Device replies with messages 0x89, 0x8a

0x09	Write Calibration Date	Device stores calibration data to its nonvolatile memory
0x0a	Write Calibration Point Data	Transfers calibration point data to device
0x0b	Write CAN-bus config	Writes new CAN-bus config and resets the device
0x0c	Configure Additional data	
0x0d	Query System Variables	
0x0e	Write System Variable	
Bootloader commands		
0x20	Jump to Bootloader	From application code
0x21	Erase Flash	
0x22	Write to Page Buffer	
0x23	Write Page Buffer to Flash	
0x25	Start Application Code	

Measurement value, Uncalibrated measurement value

Priority	= 0x03
payload type	= Measurement value (0x84), Uncalibrated Measurement value (0x8b)
device address	= 1..32
channel number	= 1..32

Payload

	Measurement value, 32bit float				Unit	Status	Timestamp	
	Lsb			Msb			lsb	Msb
byte	0	1	2	3	4	5	6	7

Base measurement unit types (for specific units see attachment 1)

value	Physical quantity	Unit
0	SDAQ_pSim Units	\Q/
1	SDAQ base unit for voltage	V (base)
2	SDAQ base unit for current	mA (base)
3	SDAQ base unit for temperature	°C (base)
4	Reserved for future SDAQ	
.	Reserved for future SDAQ	
.	Reserved for future SDAQ	
.	Reserved for future SDAQ	
19	Reserved for future SDAQ	

Status

Bit	Name	Desc.
0	Sensor Error	0 = ok, 1 = sensor disconnected
1	Out of Calibrated Range	0 = ok, =1 if device has calibration (num of points >=2) and the raw measurement value is lower than the input value of first calibration point or higher than the input value of last calibration point
2	Overrange	0 = ok, 1 = signal out of measurement range

Timestamp = 0..59999 ms

Uncalibrated values (0x8b) may use lower sampling rate.

Device ID / status

Priority	= 0x04
payload type	= Device ID/status (0x86)
device address	= 1..32
channel number	= 0

Payload

	Device serial number, uint32_t				Status	Device Type
	Lsb			Msb		
byte	0	1	2	3	4	5

Status

Bit	Name	Desc.
0	Run / Standby	0 = stdby, 1 = run
1	Sync status	0 = no sync, 1 = sync message received within 120 second
2	Error	0 = ok, 1 = error
3..6	reserved	
7	Bootloader	0 = application code, 1 = bootloader running

Device types

ID	Name	Desc.
1	SDAQ-TC1	1 channel thermocouple
2	SDAQ-TC16	16 channel thermocouple
3	SDAQ-RTD	1 channel Pt100 RTD
4	SDAQ-I	1 channel current meter
5	SDAQ-U	1 channel voltage meter

Extended ID/Status message

Payload

	Device serial number, uint32_t				Status	Device Type	Hw revision	reserved
	Lsb			Msb				
byte	0	1	2	3	4	5	6	7

Device Info

Priority	= 0x04
payload type	= Device Info (0x88)
device address	= 1..32
channel number	= 0

Payload

	Device Type	Sw revision	Hw revision	Num. of channels	Sample rate	Max calib. points / channel
byte	0	1	2	3	4	5

Device type

Check Device ID / status message for device types

Sw/Hw revision

0..255

Num of channels

1..32

Samplerate

n samples / second

Max calibration points per channel

Device type dependent, TC1 (=16), TC16 (=8)…

Calibration Date

Priority	= 0x04
payload type	= Calibration date (0x89), Write Calibration date (0x09)
device address	= 1..32
channel number	= 1..32

Payload

	Calibration date & calibration period, uint32_t				Number Of Points	Calibrated meas. unit
	lsb			Msb		
byte	0	1	2	3	4	5

Calibration date & calibration period

Byte	Description
0	Year of calibration. Years after 2000, for example value 19 means 2019.
1	Month of calibration.
2	Day of calibration.
3	Calibration period in months. Wärtsilä standard is 12 months.

Number Of Points = 0..8, number of calibration points

Calibrated measurement unit

Measurement unit in *measurement value* messages can be controlled with this value.

If number of points > 0 and calibrated unit >0, the unit field in measurement message is filled with this value.

Calibration Point Data

Priority	= 0x04
payload type	= Calibration Point Data (0x8a)
device address	= 1..32
channel number	= 1..32

Payload

	Point Value				Type	Point Number
	lsb			msb		
byte	0	1	2	3	4	5

Value	32bit float
Type	1 = input value 2 = output value 3 = a0 4 = a1 5 = a2 6 = a3
Point Number	0..7

Input value (1) is used to select which polynomial is used to calculate calibrated value. Output value (2) is not used by the device.

Calibrated value (y) is calculated from uncalibrated value (x) value using polynomial:

$$y = a3_n x^3 + a2_n x^2 + a1_n x + a0_n$$

Index n (point number) is selected by comparing input value (x) to saved input$_n$ values.

$$x < input_1 \rightarrow \quad n = 0$$

$$input_m < x \leq input_{m+1} \rightarrow \quad n = m$$

$x > input_{k-1} \rightarrow \quad n = k-1$, k = num of calib. Points

Calibration points must be saved in ascending order according to the input values. So point number 0 has the lowest input value.

Synchronization command

Priority = 0..7
payload type = Synchronization command (0x01)
device address = 0
channel number = 0

Payload

	Timestamp	
	Lsb	Msb
byte	0	1

Timestamp = 0..59999 ms

Start command

Priority = 0..7
payload type = start (0x02)
device address = 1..32 (broadcast address 0 should work too?)
channel number = 0

No Payload

Commands device(s) to run-state allowing measurement streaming.

Stop command

Priority = 0..7
payload type = stop (0x03)
device address = 1..32 (broadcast address 0 should work too?)
channel number = 0

No Payload

Commands device(s) to standby-state forbidding measurement streaming.

Set Device Address

Priority	= 0x04
payload type	= Set Device Address (0x06)
device address	= 0
channel number	= 0

Payload

	Device serial number, uint32_t				New device addresss
	Lsb			Msb	
byte	0	1	2	3	4

New device address = 1..32

Device with matching serial number changes its address and enters to idle state (no measurement streaming). Device confirms address change by sending id/status (0x86) message.

Query Device Info

Priority = 0..7
payload type = Query device info (0x07)
device address = 1..32 (broadcast address 0 should work too?)
channel number = 0

No Payload

Device responds by sending Device Info, Calibration date & Device ID messages

Write CAN-bus config

Priority	= 0x04
payload type	= Write CAN-bus config (0x0b)
device address	= 0, 1..32
channel number	= 0

Payload

	Bitrate						
byte	0						

Bitrate 0 = 1000 kbps
1 = 500 kbps
2 = 250 kbps

Writes new CAN-bus config and resets the device.

Configure Additional data

Priority	= 0x04
payload type	= Configure additional data (0x0b)
device address	= 0, 1..32
channel number	= 0

Payload

	Config						
byte	0						

Config bit 0, uncalibrated measurement values
bit 1, internal measurements

Controls additional data sent by the device. Write bit to 1 to enable, 0 to disable.

Query System variables

Priority = 0..7
payload type = Query system variables (0x0d)
device address = 1..32 (broadcast address 0 should work too?)
channel number = 0

No Payload

Device responds by sending all of its systems variables. Number of variables are specific for current device type and firmware revision.

System Variable / Write System Variable

Priority = 0x04
payload type = System Variable (0x8d) / Write System Variable (0x0e)
device address = 1..32
channel number = 0

Payload

	Variable Value				Variable Number
	lsb			msb	
byte	0	1	2	3	4

Number of variables and content of each variable are specific for each device type – or even device's firmware version.

Bootloader

There are two types of microcontrollers used in sdaq-devices. The older ones with LPC1519 and the newer ones with LPC5504. These two variants need different firmwares. Unfortunately there is no generic way to acquire the uC-type of device. So it must be done using HV-revision returned in DeviceInfo-message.

SDAQ-U

```
    HW 0..4     LPC1519
    HW 5        LPC5504
```

SDAQ-TC16

```
    HW 0..4     LPC1519
    HW 5..6     LPC5504
```

SDAQ-TC1

```
    HW 0..5     LPC1519
    HW 6        LPC5504
```

SDAQ-I

```
    HW 0..2     LPC1519
    HW 3..4     LPC5504
```

SDAQ-RTD

```
    HW 0..4     LPC1519
    HW 5..7     LPC5504
```

Preparing flash image

For LPC1519 devices the flash image is compiled and linked to start at address 0x10400. For LPC5504 devices this address is 0x8400.

Bootloader on device expects a header about the application code. In LPC1519 devices this header is located at 0x10000 and for LPC5504 at 0x8000 respectively. Bootloader host must generate and add this header when writing the code.

LPC1519

0x10000	0x27	Header word = 0x18281827
01	0x18	
02	0x28	
03	0x18	
04	LSB	Start Addr of code == 0x10400
05	..	Start Addr
06	..	Start Addr
07	MSB	Start Addr
08	LSB	End Addr
09	..	End Addr
0a	..	End Addr
0b	MSB	End Addr
0c	LSB	CRC of code aread
0d	..	CRC
0e	..	CRC
0f	MSB	CRC

LPC5504

0x8000	0x28	Header word = 0x18281828
01	0x18	
02	0x28	
03	0x18	
04	LSB	Start Addr of code == 0x8400
05	..	Start Addr
06	..	Start Addr
07	MSB	Start Addr
08	LSB	End Addr
09	..	End Addr
0a	..	End Addr
0b	MSB	End Addr
0c	LSB	CRC of code aread
0d	..	CRC
0e	..	CRC
0f	MSB	CRC

Programming sequence

If device is running the application code, send “Jump to bootloader” command and wait for an “device-id” message indicating that the loader is running.

Erase flash using command 0x21 (flash erase)

Fill the 256 byte buffer by sending 32 “Write to buffer” commands.

Write buffer to flash by sending “Write Buffer to Flash” command with appropriate flash address.

After successful write, device sends 32 “Bootloader buffer data” messages and one “bootloader reply” message.

Read all buffer data messages and after receiving reply message verify that the incoming data matches to the data just sent.

Increment address by 256 bytes and repeat the write buffer / buffer to flash sequence until the whole image is written to the device.

Jump to the application code by sending “Jump to App” command.

In case of timeout or verify error, restart the whole process (erase, buffer, write…).

Jump to Bootloader (from application)

Priority	= 0..7
payload type	= Jump to Bootloader (0x20)
device address	= 1..32
channel number	= 0

Payload none

Jump to Application (from bootloader)

Priority	= 0..7
payload type	= Jump to Application (0x25)
device address	= 1..32
channel number	= 0

Payload none

Bootloader Response

Priority	= 0
payload type	= Bootloader response (0xa0)
device address	= 1..32
channel number	= 0

Payload

	Error Code	Command	0	0	IAP return value			
byte	0	1	2	3	4	5	6	7

Error Code, 0 = no error

Erase Flash

Priority = 0..7
payload type = Erase Flash (0x21)
device address = 1..32
channel number =

Payload

	Start Addr, uint32_t				End Addr, uint32_t			
	Lsb			Msb	Lsb			Msb
byte	0	1	2	3	4	5	6	7

Device responds with 0xa0 (bootloader response) when erase is completed.

Write To Buffer / Bootloader buffer data

Priority = 0..7
payload type = Write to buffer (0x22), Bootloader buffer data (0xa1)
device address = 1..32
channel number = buffer addr / 8, (0..31)

Payload

	Data[8]							
byte	0	1	2	3	4	5	6	7

No response from device

Write Buffer to Flash

Priority	= 0..7
payload type	= Write buffer to flash (0x23)
device address	= 1..32
channel number	= buffer addr / 8, (0..31)

Payload

	Flash Addr			
	Lsb			Msb
byte	0	1	2	3

After successful write device reads the flash to the buffer and responds with 32 “Bootloader buffer data” messages containing 256 bytes of data.

Debug data

Priority	= 0x07
payload type	= Sync Info (0xc0)
device address	= 1..32
channel number	= 0

Payload

	Ref Time		Dev Time					
	Lsb	Msb	lsb	Msb				
byte	0	1	2	3	4	5	6	7

ATTACHMENT 1.

value	Physical quantity	Unit
20	Voltage	V
21	Voltage	uV
22	Voltage	mV
23	Voltage	kV
24	Current	A
25	Current	uA
26	Current	mA
27	Current	kA
28	Temperature	°C
29	Pressure	bar
30	Pressure	barg
31	Pressure	Pa
32	Pressure	kPa
33	Pressure	MPa
34	Pressure	GPa
35	Strain	um/m
36	Force	N
37	Force	kN
38	Force	MN
39	Displacement	m
40	Displacement	um
41	Displacement	mm
42	Displacement	cm
43	Displacement	dm
44	Velocity	m/s
45	Velocity	mm/s
46	Velocity	km/h
47	Acceleration	m/s2
48	Acceleration	g
49	Resistance	Ohm
50	Resistance	kOhm
51	Resistance	Mohm
52	Torque	Nm
53	Torque	kNm
54	Torque	MNm
55	Mass	kg
56	Mass	g
57	Mass	t
58	Angle	deg
59	Angle	rad
60	Frequency	Hz
61	Frequency	kHz
62	Frequency	MHz
63	Frequency	rpm
64	Angular acceleration	rad/s2
65	Angular acceleration	deg/s2
66	Angular velocity	rad/s
67	Angular velocity	deg/s
68	Mass flow	kg/s
69	Mass flow	kg/min
70	Mass flow	kg/h
71	Volumetric flow	m3/s
72	Volumetric flow	m3/min
73	Volumetric flow	m3/h
74	Volumetric flow	l/s
75	Volumetric flow	l/min
76	Volumetric flow	l/h
77	Humidity/percentage	%
78	Power	W
79	Power	kW
80	Power	MW
81	Energy	J
82	Energy	kJ
83	Energy	MJ
84	Energy	Wh
85	Energy	kWh
86	Energy	MWh
87	Ratio	mV/V
88	Ratio	mV/mA
89	Volume	l
90	Volume	m3

Firmware Revision history

SDAQ-U

```
    date       sw revision
    18.1.2021  8 * Fixed "Calibration Date msg" when replying "Query Device Info"
    2.10.2020  7 * support for hw4 (DIN)
     3.9.2020  6 * config led changed to fit case graphics
    30.7.2020  5 * more than 8 calibration equations bug fixed
     8.5.2020  4 * support for hw 1.3
    14.4.2020  3 * dev info msg fixed (last byte was missing, max num of calib points)
                 * LTC2440 power cycling removed
    15.3.2020  2 * (1st release candidate)
```

SDAQ-TC16

```
    date       sw revision
    18.1.2021  8 * Fixed "Calibration Date msg" when replying "Query Device Info"
    10.9.2020  7 support for board 1.3 & 1.4 (wide)
    8.6.2020   6 support for board 1.21
    14.3.2020  5 (1st release candidate)
```

SDAQ-TC1

```
    date       sw revision
    18.1.2021  9 * Fixed "Calibration Date msg" when replying "Query Device Info"
    24.8.2020  8 * new mode led definition (to fit case graphics)
    18.6.2020  7 * more than 8 calibration equations bug fixed
    8.5.2020   6 * support for hw 1.5
    6.4.2020   5 * dev info msg fixed (last byte was missing, max num of calib points)
    14.3.2020  4 * (1st release candidate)
```

SDAQ-I

```
    date       sw revision
    18.1.2021  4 * Fixed "Calibration Date msg" when replying "Query Device Info"
    10.7.2020  3 * support for board revision 1.2
                 * more than 8 calibration equations bug fixed
    14.4.2020  2 * dev info msg fixed (last byte was missing, max num of calib points)
                 * LTC2440 power cycling removed
                 * Added status bit SDAQ_MSTATUS_OVERRANGE
    27.3.2020  1 * (1st release candidate)
```

SDAQ-RTD

```
    date       sw revision
    18.1.2021  4 * Fixed "Calibration Date msg" when replying "Query Device Info"
    18.6.2020  3 * more than 8 calibration equations bug fixed
    14.4.2020  2 * dev info msg fixed (last byte was missing, max num of calib points)
                 * fixed ADC gain for Pt1000 sensor
```

```
             * added status bit SDAQ_MSTATUS_SENSOR
15.3.2020  1 * (1st release candidate)
```

System variables

System variables are considered as part of private interface of the sdaq-device and therefore they are subject to change with future firmware releases. Building external applications depending on system variables are highly undesirable.

SDAQ-U, sw rev 8

```
   0  : uint32 : input range 0 = 2V, 1 = 100V
   1  : float  : 2V offset mV
   2  : float  : 2V gain
   3  : float  : 100V offset mV
   4  : float  : 100V gain
```

SDAQ-TC16, sw rev 8

```
   0  : float  : Offset 1
   1  : float  : Gain 1
   2  : float  : Offset 2
   3  : float  : Gain 2
```

SDAQ-TC1, sw rev 9

```
   0  : uint32 : TC type : B(0), E, J, K, N, R, S, T(7)
   1  : float  : offset mV
   2  : float  : gain
```

SDAQ-I, sw rev 4

```
   0  : float  : Offset mA
   1  : float  : Gain
```

SDAQ-RTD, sw rev

```
   0  : uint32 : input mode Pt100  2 Wire (0), Pt100  3 Wire(1), Pt100  4 Wire(2)
                            Pt1000 2 Wire (3), Pt1000 3 Wire(4), Pt1000 4 Wire(5)
   1  : float  :  ref resistor value (ohm)
```