CSD File Format Instruction
-Version : 1.34

Update history

Date	*Updated By*	*Version*	*Description*
Aug 5, 2011	BE	1.1	- Add CSD file structure - Add relation description of pheader, channel and data
Nov 18, 2011	HL	1.21	- Version updated to 3 - Add pointer to first sample in protocol header table - Add average setting in Protocol header table - Add Version of logger to Protocol header table - Add device description and sensor description in channel header table
Mar 5, 2012	HL	1.22	- Fix bugs in V1.1 - Adjust position of sensor description and device description.
Apr 19, 2012	HL	1.23	- add channel config
Jul 18, 2012	HL	1.30	- add tester name, company name, company address ...
Aug 02, 2012	HH	1.31	- Update the tester name length to 32
May 07, 2013	HH	1.32	- Add device type to "**Protocol Header Table"** - Add slave address to **"Channel Header Table"** - Updatet the VERSION to 5 of "**File Information Table**"
Aug 19, 2013	HH	1.33	- Add the definition for special macro in CSD file
Jan 17, 2014	BE	1.34	- Update version to 6 of "**File Information Table**" - Add origin to "**Protocol Header Table**"

Data format: All multi-byte data type is MSB first unless specified.

- CSD file is structured in following blocks:

File Information
Protocol Header
Channel Headers
Measurement Datas

- File Information Table

Field Name	Type	Length	Remark
VERSION	int	4	Now is 7
FILE_IDENTIFIER	chars	10	Now default CSMDF
System.currentTimeMillis()	long	8	
Dummy	long	8	
recordPosition	int	4	Start record value's position

- Protocol Header Table

offset: 34

Field Name	Type	Length	Remark
Pref	long	8	An unique identification for the csd file
DeviceID	int	4	serial number
Length of descriptions	short	2	
Description(location name)	string	128	Description of the protocol
Length of tester name	short	2	
Tester name	string	32	
Length of company name	short	2	
Company name	string	32	
Length of company short address	short	2	
Company short address	string	128	
Length of company name	short	2	
Service company name	string	32	
Length of service company short address	short	2	
Service company short address	string	128	
Length of device name	short	2	
Device name	string	32	
Calibration date	double	8	LSB first
Reserved	Byte[]	2466	
NumOfDevices (offset: 0xbe6)	int	4	Number of devices in the protocol
NumOfChannels	int	4	Number of channels in the protocol
NumOfSamples	int	4	Number of samples in the protocol
SampleRate	int	4	The number represents second
SampleRateFactor	int	4	Default 1000
TimeOfFirstSample (offset: 0xbfa)	long	8	The long integer is one in millisecond since Jan 1st 1970 00:00:00
StopTime	long	8	No use
Status (offset: 0xc0a)	int	4	From logger setting of device: Stop, Wait, Log, wrap around...
Version of logger firmware	short	2	100 is shown as 1.00
Pointer to first sample	int	4	Used for PC software to search first sample inside the

(offset: 0xC10)			protocol. It's useful when wrap around is occurred.
CRC(offset: 0xC14)	short	2	CRC check sum of protocol header
Device type(offset: 0xC16)	Unsigned int short	2	Device type saved when Version of logger firmware is 125 or higher
Origin	byte	1	Just used for maintain. Tip where the csd file come from. Value define: - Device created: 1; - CSM-2G read single device stop logger created: 2; - CSM-2G read single device running logger created: 3; - CSM-2G import csd file created: 4; - CSM-2G import csv file created: 5; - CSM-2G online logging running created: 6; - CSM-2G online logging stop created: 7; - CSM-S read stop logger created: 8; - CSM-S read running logger created: 9; - CSM-S online recoding created: 10; - CSM-M read stop logger created: 11; - CSM-M read running logger created: 12; - CSM-M online recording created: 13; - CSM-2G merge file created: 14; - CSM-2G deleted : 15; - CSM-2G export file : 16; - CSM-2G import xlsx file : 17; - CSM-2G import xls file : 18;
Reserved	Byte[]	489	Note length

- Channel Header Table (length 918)

Field Name	Type	Length	Remark
Pref (offset: 0xe02)	long	8	An unique identification for the csd file. Same as one in Protocol Header Table
Length of channel desp.	short	2	
Channel descriptions	string	128	
Length of sub. Desp.	short	2	
SubDevice descriptions	string	128	
Length of device desp.	short	2	
Device descriptions	string	19	
Length of sensor desp.	short	2	
Sensor descriptions	string	19	
Reserved	Byte[]	470	
ChannelNumber (offset 0x110e)	int	4	Channel index starts from 0
Unit	int	4	An integer number representing the unit
Length of unit text	short	2	
UnitText	string	58	Note length
Resolution (offset 0x1152)	int	4	
Min	double	8	
Max	double	8	
Device ID	int	4	
SubDevice ID	int	4	
Sensor ID	int	4	
Chanel ID	int	4	

Channel config	Byte	1	Bit 0: 1 the channel is counter
Slave address	Unsigned char	1	Save slave address of MODBUS when Version of logger firmware is 125 or higher
Device type	Unsigned int short	2	Device type saved when Version of logger firmware is 125 or higher(Add at version 7)
Device Unique id	byte[]	8	When origin is CSM2G and device type is 3rd party this field used with device ID together to construct the SN. For CSM2G this field store mac address of the server.
Reserved	Byte[]	22	Note length

Note: If Reserved buffer is used up, the length of Channel description can be cut to reserve space for new fields.

- Data Table (offset 3586 + channel number *918)

Field Name	Type	Length	Remark
ID	int	4	
Value	double	8	

Note: ID starts from 0.

E.g: Format of one record (One Measurement Data)

id	Value1 from CH1	Value2 from CH2	……	ValueN from CHN

Note: N is the number of recorded channels

CSD file header information length:

4+4+5*2+8+8=34

Protocol header length:

8+4+3*1000+4*5+8+8+4+500=3552

The first channel header position is:

CSD file header length+ Protocol header length=34+3552=3586

One channel header length:

256*3+8+4+4+4+3*20+4+8+8+50=918

One record length

4+8*number of channel

Calculate record start position:

3586+918 * number of channel + value's id*(4+8 * number of channel)

- CSD special macro definition:

```
#define DATA_INVALID                -9999
#define DATA_OVERRANGE              -8888
#define DATA_SENSOR_CHANGE          -8887
#define DATA_UNIT_CHANGE            -8886

#define DATA_OUTPUT_VALUE_TYPE_INVALID      -8885 // only used in PC SW CSM.
                                                  // check data out put type if valid
```

// when getting online measurement
//value, if invalid show this value

- Relation of pheader, channel and data's table