SHDLC
Protocol Definition

CONTENT

This document contains the protocol definition fort he Sensirion High-Level Data Link Control.

CONFIDENTIAL

SENSIRION
THE SENSOR COMPANY

RECENT CHANGES ON THIS DOCUMENT

Date	Version	Author	Why
11.11.2011	2	LWI	Initial draft with version
17.01.2013	3	LWI	7.1 Removed block transfer

SENSIRION
THE SENSOR COMPANY

1 Table Of Contents

2 GLOSSARY

MOSI	Master Out Slave In. Frame direction from master to slave.
MISO	Master In Slave Out. Frame direction from slave to master.
SHDLC	Sensirion High-Level Data Link Control
MSb	Most significant bit
LSb	Least significant bit
MSB	Most significant byte
LSB	Least significant byte

3 INTRODUCTION

The intention of defining the SHDLC protocol was to have one protocol for different Sensirion devices as massflow controllers, meters, and other microcontroller devices. The communication is typically between a PC and a device bus where one or several Sensirion devices are connected.

4 SHDLC KEY FEATURES

The following list of features should help to decide if the SHDLC protocol is suitable for an application (or new product) or not:

- SHDLC is based on an **byte orientated** hardware
- The bus operates as **half-duplex** system (no transmit and receive at the same time)
- The protocol is a **Master/Slave** protocol without the need for bus arbitration
- Bus transfers allow the transfer of 255 bytes Tx and 255 bytes Rx
- The protocol supports **Broadcasting**

5 HARDWARE

The protocol is based on a byte oriented, bidirectional interface without hardware handshaking. Typically this will be an UART interface (RS232, RS485, RS422,…).

Because the protocol is a halfduplex protocol, Rx and Tx lines can be shared.

In the following there are some conventions listed for the different hardware options.

5.1 RS485 FULL/HALF DUPLEX DEFINITION

The UART hardware uses following settings:

- 115'200 baud
- Full/Half Duplex
- 8 Data bits (LSb first)
- No Parity
- 1 Stop Bit

6 FRAME DEFINITION

In the following, the composition of the frame body is shown. This body is used for every transfer in an SHDLC system.

MOSI Frame

The following diagram shows the data flow in relation to time for a MOSI frame:

	Frame Content					
Start (0x7E)	Adr 1 Byte	CMD 1 Byte	L 1 Byte	Tx Data 0…255 Bytes	CHK 1 Byte	Stop (0x7E)

The start signalizes the begin of a new frame. The following address field defines the receiver of the frame. Then a device command follows. The length (L) defines the size of the transferred data. After data a checksum (over the frame content) and a frame stop signature follows.

MISO Frame

The following diagram shows the data flow in relation to time for a MISO frame:

	Frame Content						
Start (0x7E)	Adr 1 Byte	CMD 1 Byte	State 1 Byte	L 1 Byte	Rx Data 0…255 Bytes	CHK 1 Byte	Stop (0x7E)

The start signalizes the begin of a new frame, followed by an address field which contains the address of the slave. The CMD is the same as received from the master. Additional to the MOSI frame the MISO frame contains state byte to signalize the master communication or command execution errors. The length (L) defines the size of the transferred data. After the data a checksum (over the frame content) and a frame stop signature follows.

The bit order in the byte is defined separately for the different hardware protocols in chapter 5.

6.1 FRAME START AND STOP

Because there is not hardware handshaking, the frame start and stop is signalized by a unique data content:

- Start: 0x7E (01111110b)
- Stop: 0x7E (01111110b)

If this byte (0x7E) occurs anywhere else in the frame, it will be replaced by another two bytes (byte stuffing: first send 0x7D, than the original data byte with bit 5 inverted → 0x5E). This will also be done for Escape (0x7D), XON (0x11) and XOFF (0x13) bytes:

Original data byte	Transferred data bytes
0x7E	0x7D, 0x5E
0x7D	0x7D, 0x5D
0x11	0x7D, 0x31
0x13	0x7D, 0x33

6.2 Address Field

The address field in the MOSI frame (1 Byte) defines the receiver of the frame (slave device address). The address range is divided as follows:

- 0…254 slave addresses
- 255 broadcast address

In a MISO frame the address field contains the slave address (sender address).

6.3 Command

Typically (in a MOSI frame), this field contains the application command which defines for the specific application what to do with the given data. There are some reserved commands which are used for special frame transfers (see Chapter 7.1). In the MISO frame the slave will return the received command in this field.

The following Table shows the command space:

Command ID (Hex)	Size	Usage
0x00 0x7F	128	Individual device command space Commands which are defined individual for every device
0x80 0xCF	80	Device command pattern Common commands, if available they are implemented similar.
0xD0 0xEF	32	SHDLC common command space Commands which operate with every SHDLC device
0xF0 0xFF	16	Special frame identifiers space (Chapter 7.1) With this identifiers, some special frames as block transfers can be marked.

The size of the command is 1 byte.

6.4 Length

The Length byte defines the number of transferred bytes in data field (Rx or Tx). It is the length of the data field before byte stuffing, not the number of bytes which are transferred over the bus.

Example: The sender will transmit data [0xA7, 0xB4, 0x7E, 0x24]. Because of byte stuffing, it needs to transmit the stream [0xA7, 0xB4, 0x7D, 0x5E, 0x24]. The transmitted size information in this case is 0x04.

6.5 State

The MISO frame contains a state byte, which allows the master to detect communication and execution errors.

The following shows the composition of the Status byte:

b7	b6						b0
Device Error Flag	Execution error code						

6.5.1 Execution Error Code

The execution error code signalizes all errors which occur while processing the frame or executing the current command. The following table shows the error mapping:

Error Code	Meaning
0x00	No Error.
0x01 0x1F	Common error codes (see chapter 10) This codes are the same for every SHDLC device.
0x20 0x7F	Device error codes (see device documentation) Errors which are defined for a specific device

6.5.2 Device Error Flag

This flag notifies the master that an error occurred on the device during operation. If this flag is set, the master can read the device error state with the "Get Device Error State". For example a supply under voltage condition can cause the setting of the error flag.

6.6 Data

The data has a usable size of [0…255] bytes (original data, before byte stuffing). The meaning of the data content is defined in the devices command reference.

6.7 Checksum

The checksum is built before byte stuffing and checked after removing stuffed bytes from the frame. The checksum defines as follows:

1. Sum all bytes between start and stop (without start and stop bytes)
2. Take the LSB of the result and invert it. This will be the checksum.

Example (MOSI frame without start/stop and without byte stuffing):

Adr	CMD	L	Tx Data 4 Bytes	CHK
0x02	0x43	0x04	0x64, 0xA0, 0x22, 0xFC	0x94

The checksum calculates as follows:

Adr	0x02
CMD	0x43
L	0x04
Data 0	0x64
Data 1	0xA0
Data 2	0x22
Data 3	0xFC
Sum	0x26B
LSB of Sum	0x6B
Inverted (=Checksum)	**0x94**

7 PROTOCOL DEFINITION

This chapter describes the frame communication protocol in a SHDLC frame. There are some basic rules:

1. On every master request (MOSI frame) slave will respond with a slave response (MISO frame). There are two exclusions where the slave should not send a response:
   - If the checksum of a MOSI frame does not match
   - If the MOSI frame was a broadcast
2. Between receiving a MOSI frame and sending slave response, the slave will not accept any other frame from master. In case of a broadcast, the master has to wait the specified command execution time.

7.1 TRANSFER TYPES

By default, the master sends a standard frame which contains up to 255 bytes Tx and Rx data. This is called a standard frame transfer. Additionally there are some special frame transfers defined. They are marked with a special frame identifier in the CMD field of the frame. The following chapters describe the different transfer types

7.1.1 STANDARD TRANSFER

In this transfer, the Master initiates a transfer with a MOSI frame containing command and up to 255 bytes of data. After executing the command, the slave will respond with a MISO frame containing state and up to 255 bytes of data. The command field of the frame is used for an application command.

The transfer looks as follows:

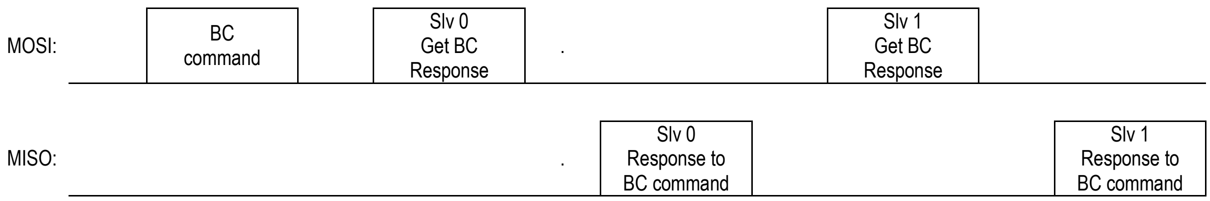
MOSI:	Start	Adr	CMD	L	Tx Data (max 255 Bytes)	CHK	Stop

MISO:	Start	Adr	CMD	State	L	Tx Data (max 255 Bytes)	CHK	Stop

7.1.2 GET BROADCAST RESPONSE TRANSFER

After sending a broadcast command, the slave executes the command but does not send the generated response. The "Get Broadcast Response" frame allows you to get the slave response on a previous broadcast command.

The following shows an example with two slaves:



If the next addressed transfer is a "Get Broadcast Response" frame, the slave will send the buffered answer. If any other frame is sent, the buffered response is discarded.

The frame to get the broadcast response (MOSI) looks as follows:

MOSI:	Start	Adr	0xF2	L	CHK	Stop

The slave answers with the same response as on an addressed command.

7.2 ERROR RESPONSE

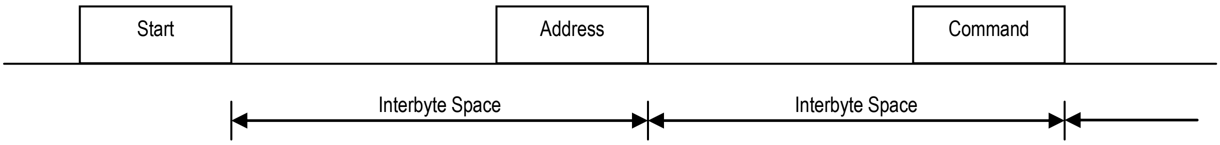
In case of a command execution error, the device will return an error response. This response may be transmitted without data (L=0). That means that a simple error response looks alike for any transfer type (normal and multiframe):

MISO:	Start	Adr	CMD	State	L (0)	CHK	Stop

8 PROTOCOL TIMINGS

8.1 INTERBYTE TIMEOUT

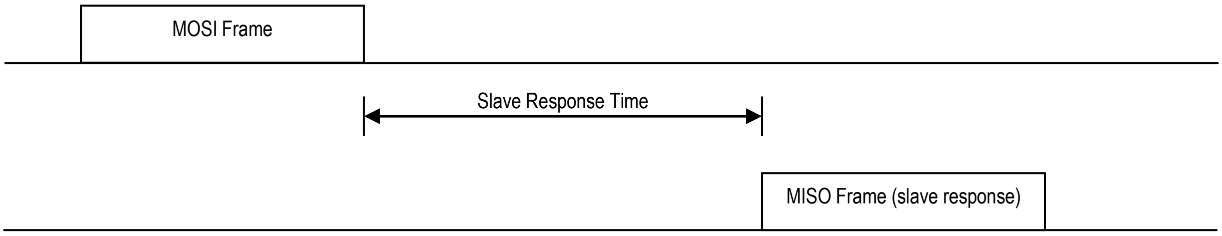
The interbyte time defines the time between two bytes in the same frame. After reception of a frame byte, the receiver waits for the next frame byte. This time is limited by the interbyte timeout. See the following timing diagram which defines the interbyte time:



The interbyte timeout is set to **200ms**.

8.2 SLAVE RESPONSE TIMEOUT

The slave response time is the time between the MOSI frame has left the master port and the begin of the reception of the MISO frame. This time is defined in the command reference.



The timeout should be at least **2 * 'Slave Response Time max'** and in none real time systems it should not be smaller than **200ms** due to possible system side delays.

9 DATA TYPES AND REPRESENTATION

The data in the frames is transmitted in big-endian order (MSB first).

9.1 INTEGER

Integers can be transmitted as signed or unsigned integers. If signed, use the two's complement. The following types of integers are known:

Integer Type	Size	Range
u8t	1 Byte	0 … 2^8-1
u16t	2 Byte	0 … 2^16-1
u32t	4 Byte	0 … 2^32-1
u64t	8 Byte	0 … 2^64-1
i8t	1 Byte	-2^7 … 2^7-1
i16t	2 Byte	-2^15 … 2^15-1
i32t	4 Byte	-2^31 … 2^31-1
i64t	8 Byte	-2^63 … 2^63-1

9.2 BOOLEAN

A boolean is represented by 1 byte:

- False = 0
- True = 1…255

9.3 FLOAT (32-BIT SINGLE PRECISION)

For floating-point representation, the IEEE 754 format is used which has the following structure:

31	30						24	23							16	15							8	7							0
S	E	E	E	E	E	E	E	E	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
	Exponent								Fraction																						

Use the following coding to signal invalid float, positive or negative infinity:

Value	Coding (hex)
invalid float (NaN)	0xFFFFFFFF
+ infinity	0x7F800000
- infinity	0xFF800000

9.4 STRING

Strings will be transferred as C-strings. This means in ASCII coding, one byte per character and terminated with a final null-character (0x00). The first letter will be sent first.

10 ERROR STATE REFERENCE

In the following, the common error codes are listed. These errors are used in the slave state response and as device state errors.

Code	Name	Meaning
0x00	no error	No error occurred on device/command execution
0x01	wrong data size	A MOSI frame had the wrong size for selected command
0x02	unknown command	Command not supported from device
0x03	no access rights for command	You need higher access rights to execute command
0x04	invalid parameter	One of the parameters for command execution was illegal or out of range
0x05	Wrong checksum	The checksum in MOSI was wrong
0x06	Firmware update operation failed	Couldn't write to flash or validation failed.
0x07		
0x08		
0x09		
0x0A		
0x0B		
0x0C		
0x0D		
0x0E		
0x0F		
0x10		
0x11		
0x12		
0x13		
0x14		
0x15		
0x16		
0x17		
0x18		
0x19		
0x1A		
0x1B		
0x1C		
0x1D		
0x1E	Idle	Ready for receive a new Command
0x1F	Busy	Slave is executing a command