OSIRE® E3731i - Open System Protocol 1.0

Application Note

Published by ams-OSRAM AG
Tobelbader Strasse 30,
8141 Premstaetten Austria
Phone +43 3136 500-0
ams-osram.com

OSIRE® E3731i - Open System Protocol 1.0

Application Note No. AN162

Valid for:
OSIRE® E3731i

Abstract

This document defines the basic communication protocol for single master - multiple slave topologies with focus on, but not limited to, automotive interior ambient lighting applications. The protocol description includes all basic characteristics of an electrical interface including the physical layer, the data link layer, and the network layer.

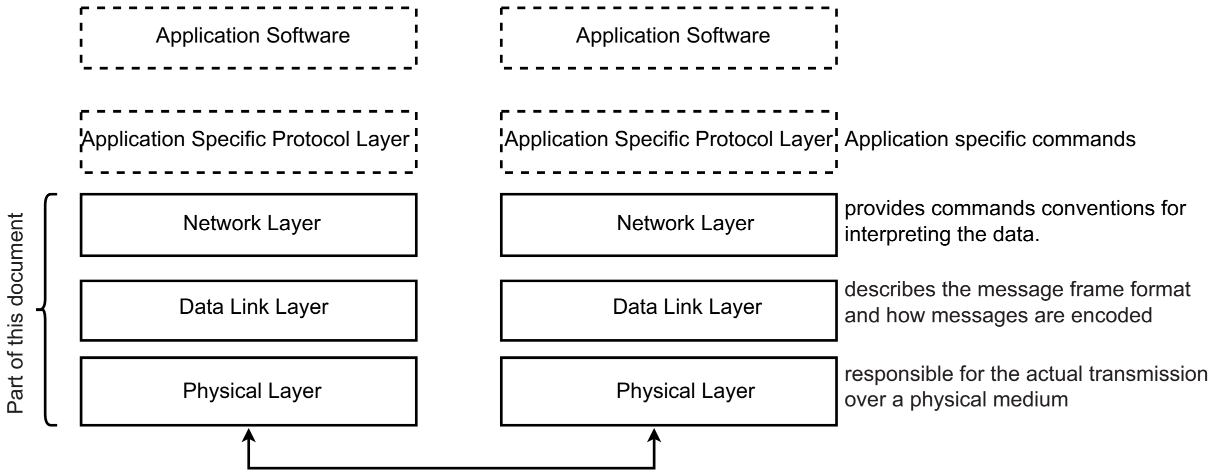
Note: Any application and/or device specific commands and registers are not part of this document but are provided elsewhere. For example, the application of the OSP for an LED requires LED-relevant commands and registers, such as PWM and current settings. These are part of a device specific document, e.g., datasheet or application note.

Table of contents

1 Basic information

Part of this document are the first three layers of the layer architecture described in Figure 1. The first layer, the physical layer, is responsible for the actual transmission over a physical medium. The second layer, the data link layer, describes the message frame format and how messages are encoded. And, the third layer, the network layer, provides commands and conventions for interpreting the data.

Figure 1: Layer architecture



1.1 Glossary and abbreviations

Table 1 shows the glossary for this documentation and explains the terms with a description.

Table 1: Glossary

Term	Description
Master	A unique node in the network that initiates communication and may be connected to a back-bone network (usually an MCU).
Slave	A node in the network that serves the communication requests of the master, e.g., an RGB LED with driver or a sensor.
Device specific	unique protocol implementation in hardware
0x"value"	Number in hexadecimal format, e.g., 0xff = 255
0b"value"	Number in binary format, e.g., 0b101 = 5

Table 2 shows the abbreviations for this documentation.

Table 2: Abbreviations

Term	Description
ACK / NACK	acknowledged / not acknowledged
BCU	body control unit
CAN (FD)	controller area network (flexible data-rate)
CRC	cyclic redundancy check
DMA	direct memory access
DUT	device under test
ECU	electronic control unit, e.g., microcontroller
EMC	electromagnetic compatibility
EMI	electromagnetic interference
EOL	end of line
ASIC	application-specific integrated circuit
LED	light emitting diode
LSB	least significant bit; also used as unit for the quantization step size
LVDS	low-voltage differential signaling
ME	Manchester encoded data
MCU	microcontroller (unit)
MSB	most significant bit
NOP	no operation
NRZ	non return to zero
OSP	open system protocol
PCB	printed circuit board
POR	power-on reset
PSI	payload size indicator
PWM	pulse-width modulation
RGB	red, green, blue
SE	single-ended
SPI	serial peripheral interface
USE	unidirectional single-ended
UTP	unshielded twisted pair
V_{pp}	Volt peak-to-peak, used for differential voltages

1.2 Mandatory and optional extended features

The implementation of these following features is mandatory for any compatible device:

- Unidirectional single-wire single-ended signaling (USE mode, separate Rx and Tx wires)
- 3.3 V and 5 V logic levels
- Fixed bit rate of 2.4 Mbit/s
- Daisy chaining of up to 1000 nodes
- Automatic addressing
- Frame-based protocol
- Bidirectional, half-duplex and unidirectional communication over the chain via loop back
- Flexible payloads up to 8 bytes
- 8-bit CRC for transmission errors
- Global broadcast configuration (write only)

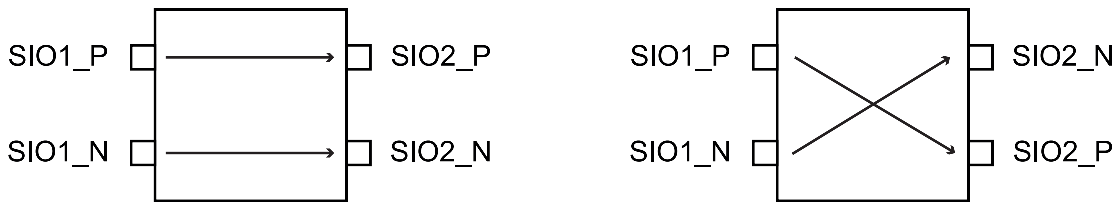
Optional extended features can be done device specific but are not mandatory:

- dedicated MCU mode to simplify integration
- differential signaling (LVDS mode) to reduce EMI

2 Physical layer

Each node provides two identical I/O ports with two pins each. In the following, these will be called SIOx_P and SIOx_N where x = 1, 2. Figure 2 shows two possible implementations with symmetric alignment (left, default) and crossed alignment (right). The crossed configuration is beneficial when the inter-node connectivity is realized via the USE-mode only. The symmetric configuration is recommended when the inter-node connectivity is realized via LVDS.

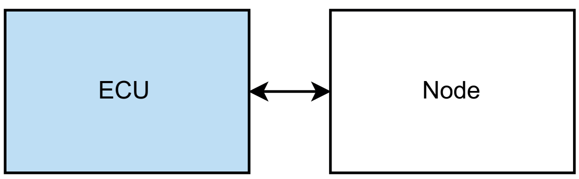
Figure 2: Symmetric and crossed alignment



2.1 Supported network topologies

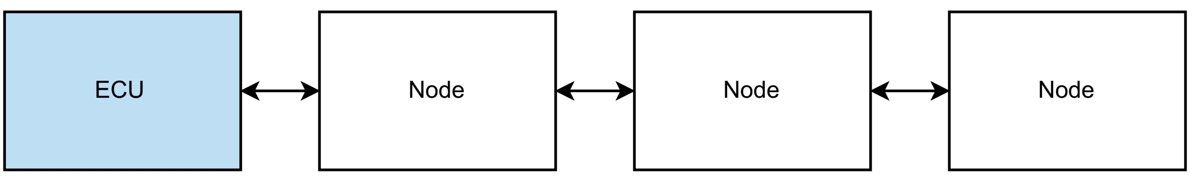
There are three supported network topologies. The point-to-point network, the daisy chain bidirectional and the daisy chain loop-back network. The point-to-point network is shown in Figure 3.

Figure 3: Point-to-point network



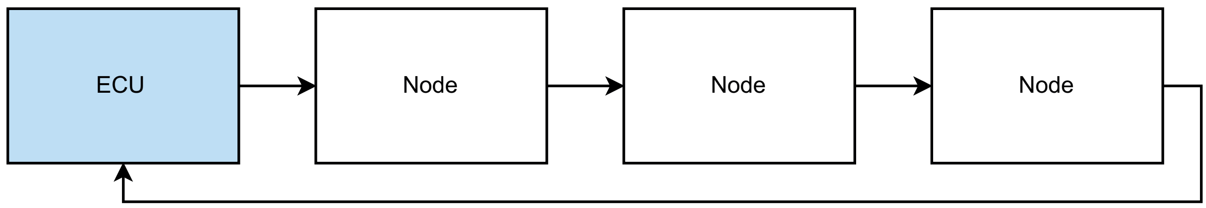
In the daisy chain bidirectional network, one network node is connected to the next in a line or chain. The communication runs through the chain once and then returns the answer along the same path. The first and last nodes are not directly connected (see Figure 4).

Figure 4: Daisy chain bidirectional network



The daisy chain loop-back network has also connected network nodes in a line or chain. But here the communication runs through the chain and, as the first and last nodes are connected to the master, the answer will return with this loop-back (see Figure 5).

Figure 5: Daisy chain loop-back network



2.2 Physical layer types and communication modes

Each of the two I/O interfaces independently supports different communication modes, that are selected through pull-up and pull-down resistors to VDD and GND, respectively.

2.2.1 Mode selection

Figure 6 and Table 3 show the possible resistor arrangements and I/O modes. Communication modes are selected directly after POR either through the pin voltage measurement or via an impedance measurement on the pins. The mode selection process has to be done within the mode selection delay time (see chapter "2.7 Parameters"). This voltage is set through pull-up and pull-down resistor combinations as indicated with each layer type. A typical resistance value of 10 kΩ is assumed. In the following the three different modes will be explained.

Figure 6: Possible resistor arrangements

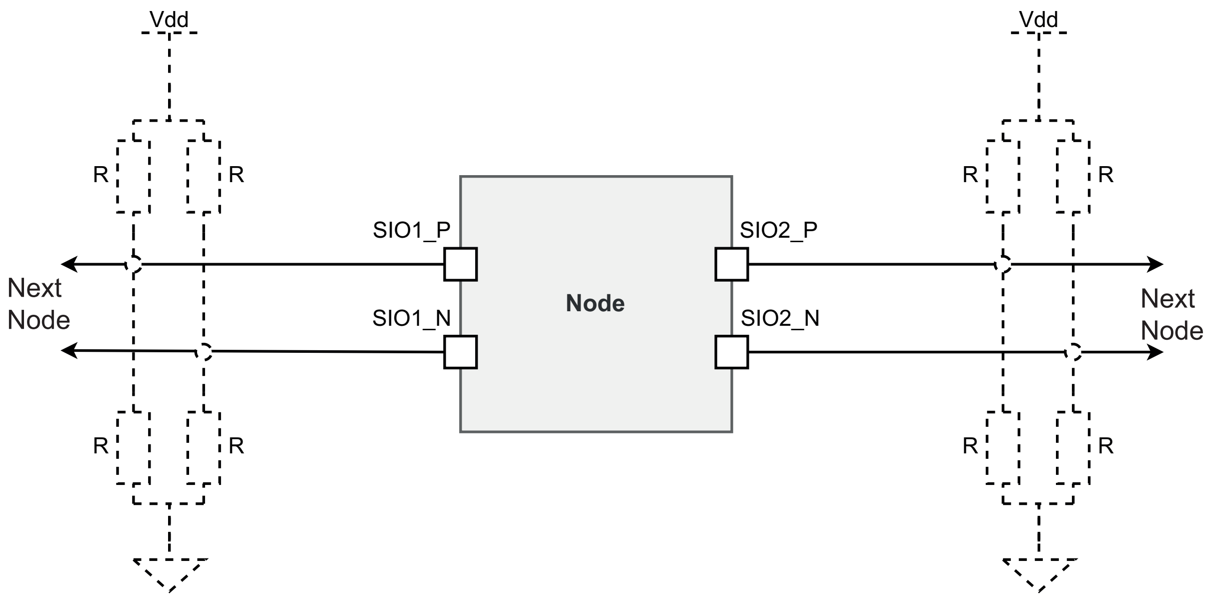


Table 3: Possible resistor arrangements

Mode	R @ SIOx_P	R @ SIOx_N	SIOx_P Input	SIOx_P Output	SIOx_N Input	SIOx_N Output
USE	up	up	-	ME	ME	-
LVDS	(down)	(down)	ME	ME	ME	ME
EOL	up	down	device specific			
	down	up	device specific			

Note: The pull-down resistors may be integrated into each node to provide a default setting. The recommended typical value for these pull-down resistors is 100 kΩ.

Note: The mode detection is implemented in a way that does not interfere with the MCU communication, i.e., does not actively pull the lines to some level that could be interpreted by the MCU as a valid transition.

Note: In EOL mode, the lines may be pulled to a value different from the idle value suggested by the resistor configuration after the mode detection has been successfully concluded. This may be used to simplify the MCU interaction on a device specific basis.

USE mode

This mode supports communication with MCUs, CAN-FD transceivers and node-to-node communication via two separate wires for transmission and reception (Tx and Rx). The communication over each wire is unidirectional.

Data is received via SIO_N only and is sent via SIO_P only. During receiving, SIO_P is switched to a high-Z state and during sending, data on SIO_N is ignored.

The bit stream is Manchester encoded. This mode is selected through two pull-up resistors and the idle state in this mode is high on SIO_P and low on SIO_N.

LVDS mode (optional)

This mode supports node-to-node communication and reduces EMI via differential signaling.

The bit stream is Manchester encoded. This mode is selected through two pull-down resistors (may be included in the node) and the idle state in this mode is $\Delta V = 0$ (recessive).

EOL mode / MCU mode (device specific)

This mode is used to indicate unit is at the end of the chain, i.e., either the first in line or the last in line and is necessary for proper initialization of the chain. The exact implementation of the end-of-line detection itself is device specific (see chapter "5 Appendix").

This mode may be selected through two different pull-up resistor configurations: either a pull-up resistor on SIO_P and a pull-down resistor on SIO_N, or vice versa. The idle state in this mode is consequently either high on SIO_P and low on SIO_N, or vice versa.

In addition to the end-of-line detection, this mode may be used to simplify communication with an MCU in a device specific fashion. Some examples for different implementations of this MCU mode are provided in the chapter "5.3 MCU mode implementation [optional, device specific]".

2.3 Timeout and idle condition

A static line, i.e., no signal change is detected on any wire for a duration of more than 1.5 bits is interpreted as timeout and communication is interrupted. The I/O port of this node will enter an idling state and be ready for a new transmission (both incoming and outgoing).

Note: The signal levels on each wire should be in the respective default states as determined by the pull-up and pull-down resistors.

2.4 Start and stop conditions

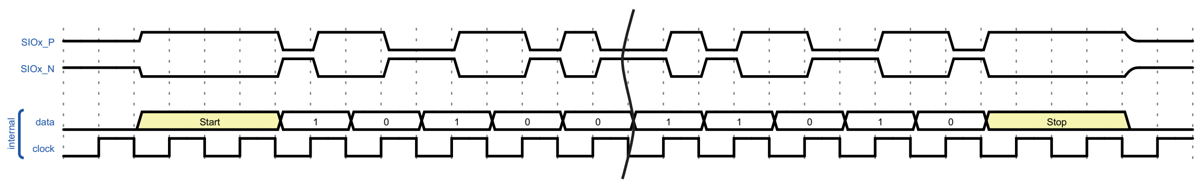
When a node is in idle state (no signal change for at least the timeout period) and any of the two wires changes its logic state, this is considered as the START condition. When a transmission has been finished and the line has been held static for at least the timeout period, this is considered as the STOP condition.

Note: In addition to the physical STOP condition, there is also a STOP condition defined on the data link layer (see chapter "End of message").

2.5 Start and stop sequence for LVDS mode

As the default condition for the LVDS mode ($\Delta V = 0\ V_{pp}$, $V_{CM} \sim 0$ V) does not correspond to the logic high or low states of the LVDS mode but is undefined, a special start and stop-sequence is required to ensure proper detection of start and stop conditions. These start and stop sequences consist of a logic-high held for a duration of 2 bits as shown in the Figure 7.

Figure 7: Start and stop sequence

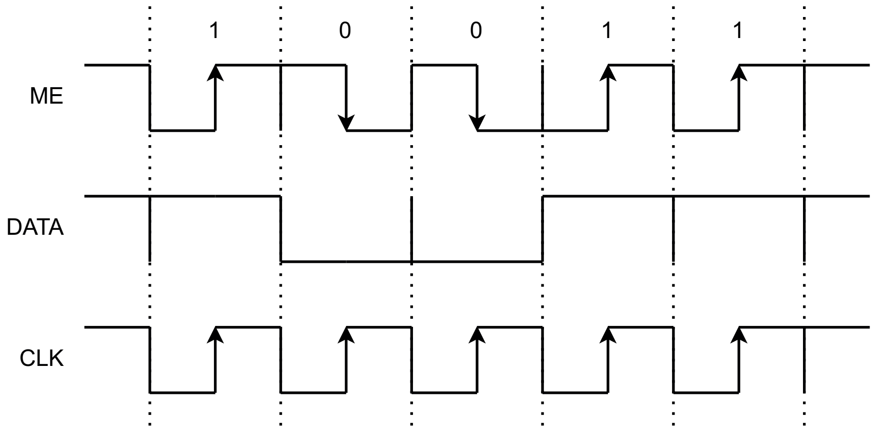


Note: There are no such start / stop sequences defined or necessary for the USE or EOL modes.

2.6 Manchester coding

The USE and LVDS modes use Manchester encoding based on the IEEE 802.3 standard, where a logical "1" is encoded by a rising edge and a logical "0" by a falling edge. Figure 8 shows DATA and CLK for illustration only.

Figure 8: Illustration DATA and CLK of Manchester encording



Note: Manchester encoding can also be used in the MCU mode depending on the device specific implementation.

2.7 Parameters

The parameters for the mode selection are shown in Table 4.

Table 4: Mode selection

Parameter	Unit	Min	Typ	Max	Comment
Resistor value	kΩ	2.2	10	25*	
Mode sel. high	V	2.5			
Mode sel. low	V			2	
Mode sel. delay	µs	5		500	after POR

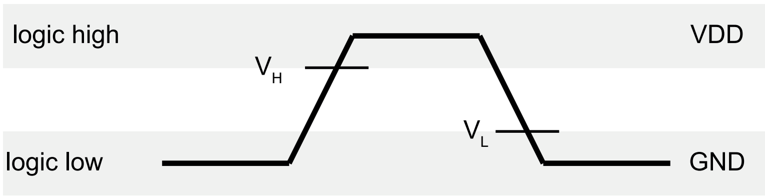
* Higher values up to 60 kΩ are possible when 3.3 V logic components are used

The single ended signaling (Table 5 and Figure 9) applies to USE, MCU and EOL modes from chapter "2.2.1 Mode selection".

Table 5: Single ended signaling

Parameter	Unit	Min	Typ	Max	Comment
Output low level	V		0	0.3	V_{OL}
Output high level	V	0.7 x VDD	VDD		V_{OH}
Input low level	V			0.7	V_{IL}
Input high level	V	2.4			V_{IH}
Load capacitance	pF			35	

Figure 9: Single ended signaling

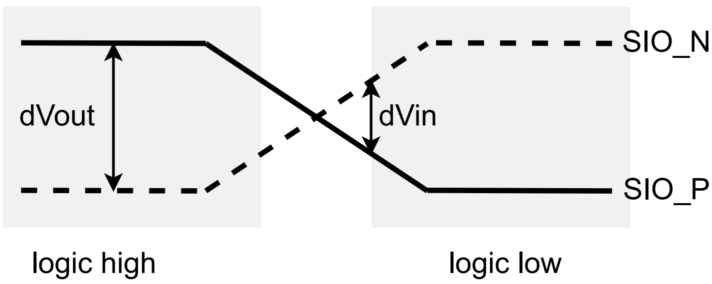


The differential signaling is shown in Table 6 and Figure 10. Logic high is encoded by a positive differential voltage, V(SIO_P) - V(SIO_N) > 0, while logic low is encoded by a negative differential voltage, V(SIO_P) - V(SIO_N) < 0.

Table 6: Differential signaling

Parameter	Unit	Min	Typ	Max	Comment
Line termination	Ω		200		
Tx current	mA		±1.5		
Common mode	V	-0.3	1.2	VDD + 0.3	
Input low level	V_{pp}	-0.15			V(SIO_P) - V(SIO_N)
Input high level	V_{pp}			+ 0.15	V(SIO_P) - V(SIO_N)
Output low level	V_{pp}		- 0.3	- 0.2	V(SIO_P) - V(SIO_N)
Output high level	V_{pp}	0.2	0.3		V(SIO_P) - V(SIO_N)
Load capacitance	pF			35	

Figure 10: Schematic differential signaling



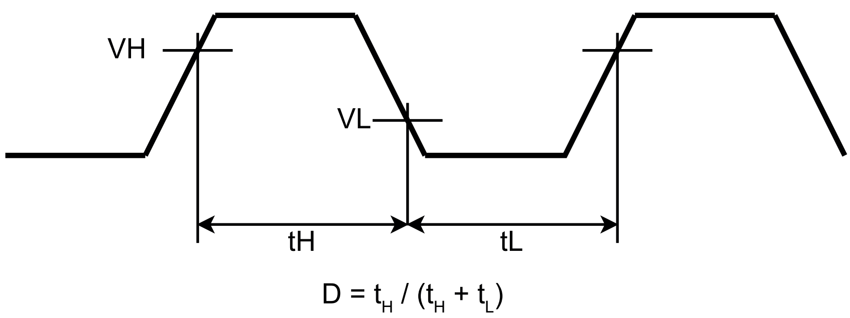
The communication timing is shown in Table 7 and the duty cycle for Manchester encoded signals is defined for a sequence of equal bits (0000…, 1111…) and for a sequence of alternating bits (010101…) as indicated in Figure 11.

Table 7: Communication timing

Parameter	Unit	Min	Typ	Max	Comment
Data rate	Mbit/s		2.4		
Rx rate variance	%	-8		8	
Tx rate variance	%	-5	0	5	
ME input duty cycle	%	40		55	H:L, see below
ME output duty cycle	%		50		H:L, see below
Timeout	µs	0.58			~ 1.5 bits
LVDS start / stop seq.	µs	0.77	0.83	0.90	~ 2 bits
Forward latency*	µs			8.5	~ 20 bits

* minimum gap between two consecutive messages sent into the chain

Figure 11: Duty cycle for Manchester encoded signals



Note: The duty cycle, data rate, and clock variance requirements together with the logic levels define the slew rate and I/O timing requirements. These are not listed separately.

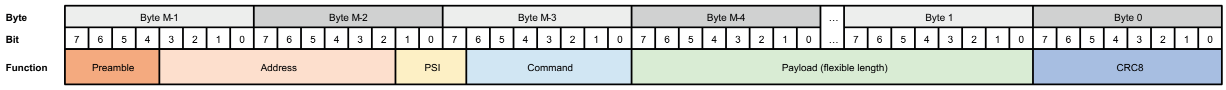
Note: Response latency is device specific and therefore not regulated in this specification.

3 Data link layer

3.1 Message frame format

The communication is message based and its frame format is shown in Figure 12.

Figure 12: Message frame format



The byte and bit numbering refers to the order for transmission. Bit order is MSB first. The message length is variable but limited to 12 bytes. The format is the same for upstream and downstream communication.

Preamble (4 bit): Fixed to 0b1010.

Address (10 bit): Allowed values for downstream messages are given in Table 8. Upstream messages always contain the address of the sender in the address field.

Table 8: Allowed values for downstream messages

Address value	Name	Tx	Rx	Description
0x000	BROADCAST	yes	no	Use this value to send the command to all devices, independent of the device address.
0x001 ... 0x3EF	<address>	yes	yes	Individual address to select a specific device 1-1007.
0x3F0 ... 0x3FR	RESERVED	yes	no	Reserved for future use.
0x3FF	INIT VALUE	no	no	Default value of an uninitialized node after POR or reset.

The initial value is listed for completeness only. It is not allowed as target address in a message. Messages with this address are silently discarded.

PSI (3 bit): The PSI field indicates the length of the payload in bytes. Its value (Table 9) must match the expected value for each command (may be different for up- and downstream directions).

Table 9: Allowed values for PSI field

PSI	0b000	0b001	0b010	0b011	0b100	0b101	0b110	0b111
Payload	0	1	2	3	4	reserved	6	8

The total message length is PSI + 4 bytes.

Command (7 bit): Commands are mostly device specific. See protocol layer description.

Payload (0 to 64 bit): Variable payload field. The length must match the value given in the PSI field.

CRC (8 bit): The CRC field contains the CRC8 check word calculated over the full message excluding the CRC field.

CRC Implementation: $x^8 + x^5 + x^3 + x^2 + x + 1$.

Polynomial: 0x2F (normal notation)

Initial value: 0x00

XOR value: 0x00

Direction: MSB first

Note: A device can select not to implement the CRC. In this case, the CRC field is still mandatory.

3.2 Message handling

Preamble

Messages without valid preamble are ignored, i.e., they are neither interpreted nor forwarded.

Forwarding and fast forwarding

Message forwarding refers to the direct forwarding of a message without modification. Messages are forwarded to the next device if the address in the message frame is different from the node's address (this includes the broadcast address).

Forwarding starts directly after the preamble and address field have been received, i.e., after 15 bits, but at most after 20 bits. This is referred to as fast-forwarding.

End of message

The PSI field is used to determine the STOP condition on the data link layer. Additional data received after the expected number of bits are silently discarded.

Note: When a transmission ends, the sending node / master should be able to directly release the lines. These are subsequently pulled to their respective idle state levels by the pull-up and pull-down resistors. Depending on their state at the end of the message, this will lead to an additional transition, which could be falsely interpreted as additional data.

Responses

Some commands trigger a response from the addressed node. The address field of the response message is set to the node's address to indicate the origin of the message.

Note: The OSP does not allow simultaneous up- and downstream messages on the same node, as this could lead to dropped or corrupted messages. Therefore, commands that trigger a response and status requests are not allowed in broadcast messages. Also, the next message should only be sent after the response has been received by the MCU. Unsolicited messages from slave nodes are not allowed.

Note: Response latency is device specific and therefore not regulated in this specification.

3.3 Communication errors on data link level

The OSP requires certain communication errors to be detected by each unit in a daisy chain independent of whether the unit is addressed (directly or via broadcast).

Incorrect preamble

Messages with incorrect preamble are silently discarded without raising an error flag. They will not be forwarded. The message will not be interpreted by the node.

Invalid address

Messages with an address 0x3FF are silently discarded without raising an error flag. They will not be forwarded. The message will not be interpreted by the node.

Incorrect PSI value [deprecated]

If the PSI value does not match with the payload (message) length, an error flag is raised. The message will not be interpreted by the node.

The message will not be forwarded. However, as fast forwarding starts before this error can be detected, fast forwarding will not be interrupted.

Note: This error is captured by chapter "Premature end of message" and chapter "Additional data after end of message".

Premature end of message

If the transmission ends before the expected number of bits (given by the PSI value) has been received, an error flag is raised. The message will not be interpreted by the node.

The message will not be forwarded. However, as fast forwarding starts before this error can be detected, fast forwarding will not be interrupted.

Incorrect CRC

If the CRC check word is incorrect, an error flag is raised. The message will not be interpreted by the node.

The message will not be forwarded. However, as fast forwarding starts before this error can be detected, fast forwarding will not be interrupted.

Additional data after end of message

If the transmission continues after the expected number of bits (given by the PSI value) has been received, these additional bits is silently discarded. No error flag is raised.

The message will not be forwarded. However, as fast forwarding starts before this error can be detected, fast forwarding will not be interrupted.

4 Network layer

With a few exceptions, the OSP does not define the protocol layer. These exceptions are required to ensure interoperability of mixed nodes in a daisy chain and are summarized as the network layer below.

Application and device specific commands are provided in specific device datasheets.

4.1 High level protocol features

4.1.1 Transaction acknowledgments

The purpose of an ACK mechanism is to signal the master that a transaction has been correctly received by a slave node.

The OSP does not directly define an ACK/NACK mechanism on physical or data link layer, but a similar mechanism can be implemented on the protocol layer on a device specific level using the non-OSP specific commands (> 0x10).

When a command triggers a response, the response message itself serves as ACK.

When a command does not trigger a response and was not sent via broadcast, a device is free to send a message as response, e.g., its status register, which again serves as ACK.

When a command is sent via broadcast, the last device in the chain (indicated by the EOL condition) could again send a message as response, which then serves as ACK for this specific device and indicates that the chain is intact.

The implementation should be done in a way that allows to enable and disable the ACK mechanism, either via a register setting (preferred) or a free bit in the command structure (for example bit 5, but not recommended as this reduces the number of possible commands).

4.1.2 Loop-back communication

By default, communication is bidirectional, half-duplex, i.e., responses to the master are returned over the same port that was used to receive the requests.

Devices additionally offer a loop-back mode. In the loop-back mode, the first and last units of the chain need to be connected with the master and responses are sent in the same direction along the chain as the request, i.e., responses are sent using the port opposite to the one used to receive the requests.

Loop-back mode may be activated through a dedicated command or via a register setting after the device has been initialized. The exact implementation is device specific.

Note: The INITLOOP command is mandatory to ensure backwards compatibility.

4.1.3 Node identification

In order to handle mixed chains, the master needs to be able to identify the device type of each node in the chain.

This is covered by a 32-bit, read-only device identification code (Figure 13), giving the following information:

- device type, e.g., sensor, LED, bridge, etc. (4 bits)
- manufacturer (10 bits)
- part identification (12 bits)
- part revision (6 bits)

Figure 13: Definition of device identification code

Byte 3								Byte 2								Byte 1								Byte 0							
31	30	29	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
Device Type				Manufacturer										Part Identification												Revision					

Refer to chapter "5 Appendix" for a list of device type and manufacturer codes.

The part identification and part revision codes are manufacturer-specific and not covered by this document. The respective values are stated in the device's datasheet.

4.1.4 Broadcasting

General broadcasting is done by address 0x000. This addresses all units in the chain with the lowest possible latency. The messages are always fast forwarded. Broadcast readouts are not allowed for the general broadcast address.

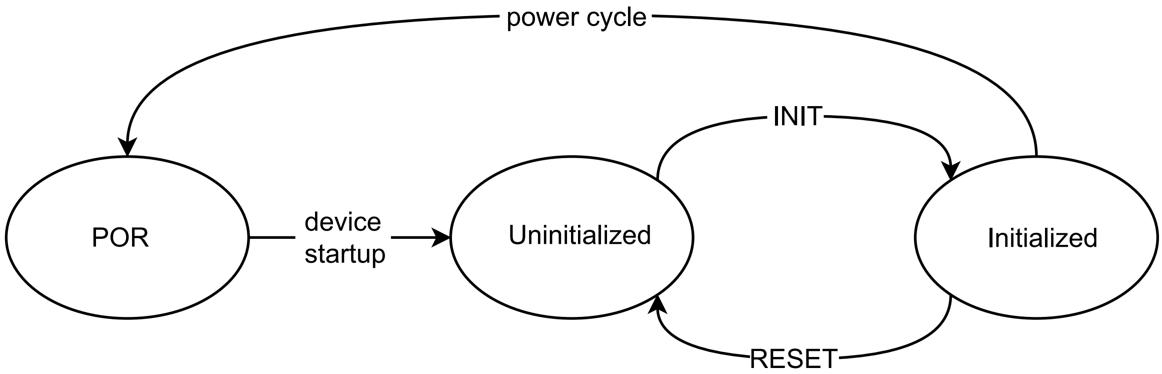
4.2 Node addressing

Node addressing is completely covered on protocol level through a dynamic initialization and automatic addressing procedure to support the high number of nodes with minimal effort on system integration side.

4.2.1 State diagram

A simplified state diagram for node addressing is shown in Figure 14.

Figure 14: State diagram for node addressing



The device is initially in the POR state after power has been applied. It automatically enters the uninitialized state after device startup. Nodes are responsive to (limited) communication (see next section). Once the node has received and executed the INIT command, its state changes to the initialized state, where it supports the full OSP command set including device specific commands. A RESET command will reset all registers to default values including the address state machine.

4.2.2 Uninitialized nodes

A node will reach this state automatically after power-up.

In this state, it only responds to a limited set of commands:

- INIT (bi-directional or loop back)
- RESET (via broadcast)
- CLEAREVENT (via broadcast)
- PING4EVENT (bi-directional or loop back)
- All write commands (via broadcast)

When a node receives the PING4EVENT command, it assumes the address in the message temporarily as its own address for the response.

Other messages are ignored. Broadcast messages are fast forwarded and addressed messages are forwarded after the message has been fully received.

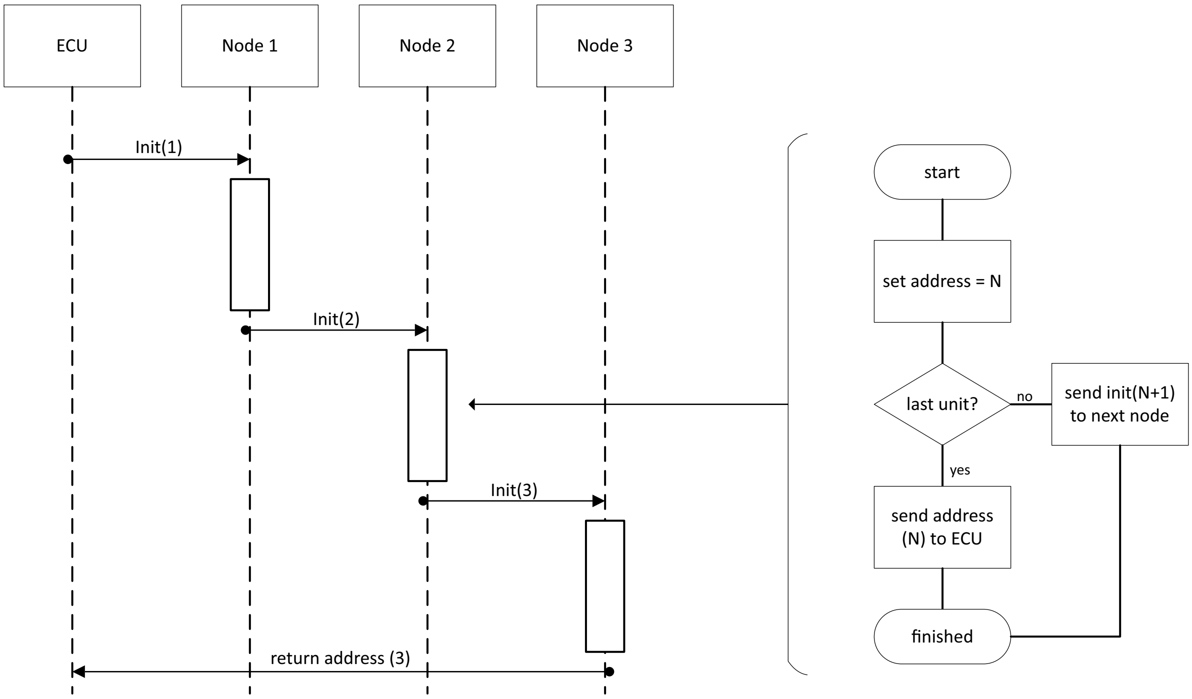
4.2.3 Chain initialization

The OSP requires dynamic addressing of units in a chain to support flexible arrangements. The addresses are assigned automatically after startup by an initialization command sent by the master (see Figure 15).

There are two flavors of INIT-commands, one for bidirectional communication and one for loop-back communication mode (see chapter "INIT" and chapter "INITLOOP").

A device receives the INIT-command with its new address set in the address field. It then sets its own address to the new address, increments the value by one and sends the INIT-command to the next unit in the chain. The last unit in the chain returns its own address to the master and stops the initialization procedure. The response is either sent in backward or forward direction depending on the command flavor used.

Figure 15: Illustration of automatic addressing



Note: If the last unit in the chain is a regular OSP node, the proper end-of-line mode selection has to be implemented at this node.

If the last node is connected to another component, for example, an MCU, this unit (MCU) needs to take care of returning the last address to the master.

Note: Every device will implement the procedure as outlined, independent of whether it was already initialized or not. This ensures proper incrementation of the address field along the chain.

Typical use cases include re-initialization of the chain if one or more nodes in the chain encountered a power cycle, for instance.

Note: An initialized node receiving an INIT-command with a target address different from its own will fast-forward the INIT-command without incrementing the address and without interpreting the command.

Thus, it is possible to initiate an uninitiated node somewhere downstream the chain with a specific address that could be different from the address it would receive by the automatic addressing. This procedure is not recommended due to the risk of creating duplicate addresses.

While not disturbing the chain, the second unit will not be accessible anymore directly through addressed commands.

4.3 Communication errors on network layer level

The following errors will only be detected by nodes when they are addressed by the message (directly or via broadcast) after the full message has been received and the basic checks have been passed (see chapter "3.3 Communication errors on data link level").

Note: Broadcast messages are always forwarded, independent of any potential protocol errors.

Unknown command

If a node receives an unknown command with code > 0x0F, an error flag is raised.

If a node receives an unknown command with code <= 0x0F (OSP specific command), no error is raised. The command will not be executed.

Incorrect length of argument

Some commands expect an argument delivered in the payload. If the length of the argument does not match the expectation, an error flag is raised. The command will not be executed.

Incorrect argument

If an argument is expected and the data in the argument does not meet the requirements, an error flag is raised. The command will not be executed.

Note: The decision whether an argument meets the expectations is device specific.

Broadcast readout attempt

If a readout command is received via the broadcast address 0x000, an error flag is raised. The command will not be executed.

Inappropriate broadcast attempt

If one of the following commands is received via the broadcast address 0x000, an error flag is raised: INIT, INIT_LOOP, PING4EVENT. The command will not be executed.

4.4 Commands

Table 10 shows a list of recommended and required commands. These commands are supported by each device to ensure the interoperability of mixed nodes in a daisy chain.

Table 10: List of recommended and required commands

Command	Code	Description	Required / Recommended
	0x00 … 0x0F	OSP Specific	required
RESET	0x00	Node reset	required
CLEAREVENT	0x01	Clear pending event / error flags	required
INIT	0x02	Initialize chain in bidirectional mode	required
INITLOOP	0x03	Initialize chain in loop-back mode	required
IDENTIFY	0x07	Returns node identification code	required
PING4EVENT	0x08	Returns pending events	required
	0x10 … 0x3F	Device Specific Commands	recommended
	0x40 … 0x7F	Register Access	recommended

The command range 0x00 to 0x0F is reserved for OSP specific commands. Nodes will not raise an error if they receive an OSP protocol command which is not implemented in this particular node.

Device specific commands and register access commands (0x10 to 0x7F) are free to be implemented by each node individually. It is recommended to use the range 0x40 to 0x7F for register access with a 5-bit register address and a read/write bit. For example, even commands could be used for reading and odd commands for write address.

Table 11: Recommended command field partitioning

Meaning	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
OSP Command	0	0	0	OSP Command [3:0]			
Device Specific Command	0	Device Command [5:0] (> 0x0F)					
Register Read	1	Register Address [4:0]					0
Register Write	1	Register Address [4:0]					1

Devices may choose to implement this differently.

RESET

Performs a complete reset of one or all devices and sets the node address to 0x3FF. The device is enter the uninitialized state with all values set to the device specific default values.

The properties of the RESET command:

- Takes no arguments
- No response
- Broadcast is possible

CLEAREVENT

Clears all pending events, e.g., failure flags, of a device. If the condition that led to the event still persists, the event will be raised again.

The properties of the CLEAREVENT command:

- Takes no arguments
- No response
- Broadcast is possible

INIT

Initiates the automatic addressing of the chain (see chapter "4.2.3 Chain initialization") and sets the communication direction to bidirectional. The command must be addressed to the first node of the chain.

The last unit in the chain sends a message with its address back to the master. The exact implementation of the return message is device specific (e.g., a node could return a status register).

The properties of the INIT command:

- Takes no arguments
- Broadcast is NOT possible

INITLOOP

Initiates the automatic addressing of the chain (see chapter "4.2.3 Chain initialization") and sets the communication direction to loop-back. The command must be addressed to the first node of the chain.

The last unit in the chain sends a message with its address via loop-back to the master. The exact implementation of the return message is device specific (e.g., a node could return a status register).

The properties of the INITLOOP command:

- Takes no arguments
- Broadcast is NOT possible

IDENTIFY

Returns the device type identifier (see chapter "4.1.3 Node identification") with a length of 32 bit.

The properties of the IDENTIFY command:

- Takes no arguments
- Broadcast is NOT possible

PING4EVENT

Poll the event status of each addressed node.

If new events are pending, the respective device sends a message with information on the events to the master. The type of information and length of payload is device specific. For example, it could include failure flags.

Uninitialized nodes send a response to the master indicating that they are not yet initialized. The exact response is device specific.

The properties of the PING4EVENT command:

- Takes no arguments
- Broadcast is NOT possible

5 Appendix

5.1 Identification codes

The 4-bit device type code (Table 12) allows to define groups of devices having an identical set of commands and registers defined beyond those defined in this standard.

Table 12: Identification codes

Code	Description	Comment
0x0	Generic Device	
0x1	3-channel LED	OSIRE® KRTBI D2LM31.31

Generic devices support all (mandatory) commands defined by the OSP technical specification.

The minimum command set for the other device types are described chapter "Device type LED — minimal command set". Table 13 lists assigned 10-bit manufacturer IDs. The manufacturer IDs are assigned by ams-OSRAM.

Table 13: 10-bit manufactors IDs

Code	Description	Comment
0x000	ams OSRAM	

Device type LED — minimal command set

All devices of device type 0x1 must support the set of commands in Table 14 in addition to the OSP base commands.

Table 14: Minimal command set

Command	Code	Payload	Description
Read Status	0x40	-	Return the device status register incl. error flags
Read LED Status	0x46	-	Return the LED error status register
Read Temperature	0x48	-	Return the node temperature
Read PWM Settings	0x4E	-	Return the PWM register (3x 16 bit)
Write PWM Settings	0x4F	3 x 16 bits	Set the PWM register (3x 16 bit)

Apart from the information given above, the exact implementation of each command is device specific.

5.2 EOL detection [device specific]

Each node needs to be able to detect whether it is the last node in the chain as required for proper initialization. The exact mechanism is device specific.

Some examples for possible implementations are given below.

Separate MCU / EOL mode with physical discrimination

Use the resistor configuration to create a unique discrimination between the first-in-line (MCU) and the last-in-line (EOL) device.

Table 15: Resistor configuration for unique discrimination

Mode	R @ SIOx_Po	R @ SIOx_N	SIOx_P Inout	SIOx_P Output	SIOx_N Input	SIOx_N Output
MCU	up	down	device specific			
	down	up	-	device specific	-	device specific

The EOL device allows sending messages as it is required for the loop-back mode.

Only the EOL device will terminate the enumeration procedure during the initialization phase and return its address to the master.

Combined MCU / EOL mode with link layer discrimination

In this case, there is a combined MCU/EOL mode that can be selected using either of the two possible resistor configurations as shown in Table 16.

Table 16: Resistor configuration for combined MCU/EOL

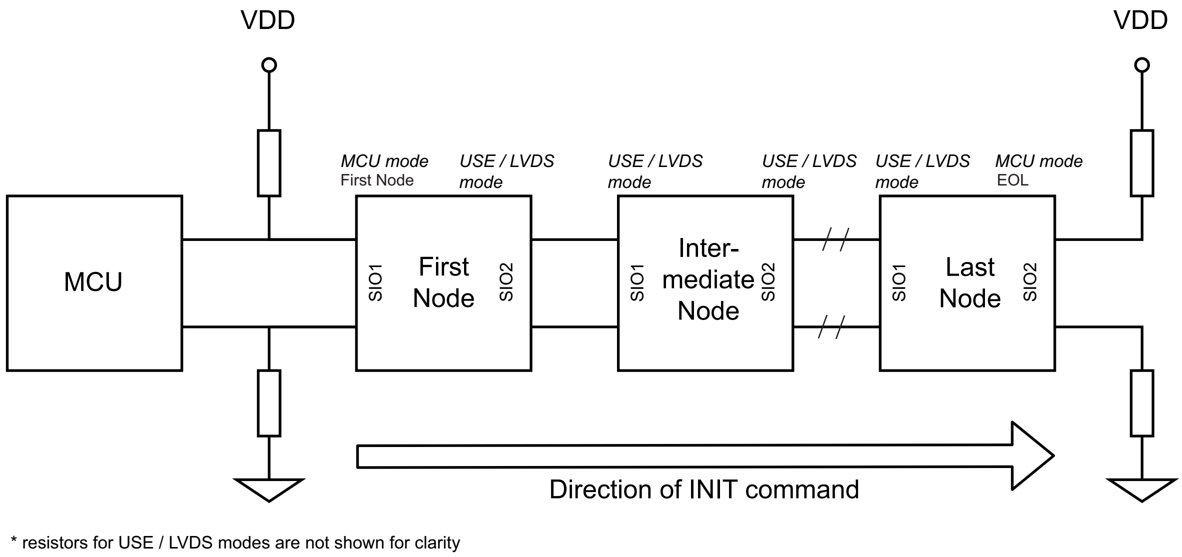
Mode	R @ SIOx_Po	R @ SIOx_N	SIOx_P Inout	SIOx_P Output	SIOx_N Input	SIOx_N Output
MCU	up	down	device specific			
	down	up	device specific			

The EOL detection is based on the direction of propagation of the initialization message along the chain relative to the MCU port as illustrated in Figure 16. The possible configurations are shown in Table 17.

Table 17: MCU / EOL configuration

Mode configuration		INIT received at	
SIO1	SIO2	SIO1	SIO2
MCU	x	first	EOL
x	MCU	EOL	first
MCU	MCU	EOL	EOL
x	x	x	x
x - other mode (USE, LVDS)			

Figure 16: Working principle



Note: The exact implementation is device specific. For example, a device might choose to support both options or only a specific option.

5.3 MCU mode implementation [optional, device specific]

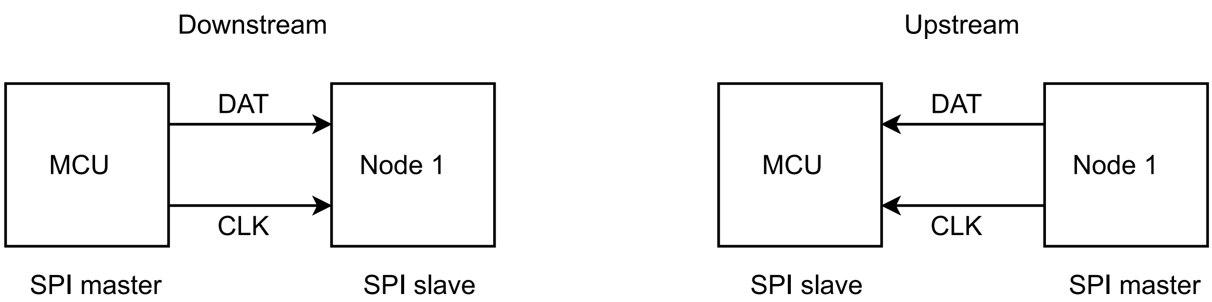
The following sections illustrate some possible implementations of the MCU mode.

The exact implementation is device specific.

SPI

Both downstream and upstream communication (Figure 17) use a regular SPI interface with clock and data signals with NRZ coding (no Manchester encoding).

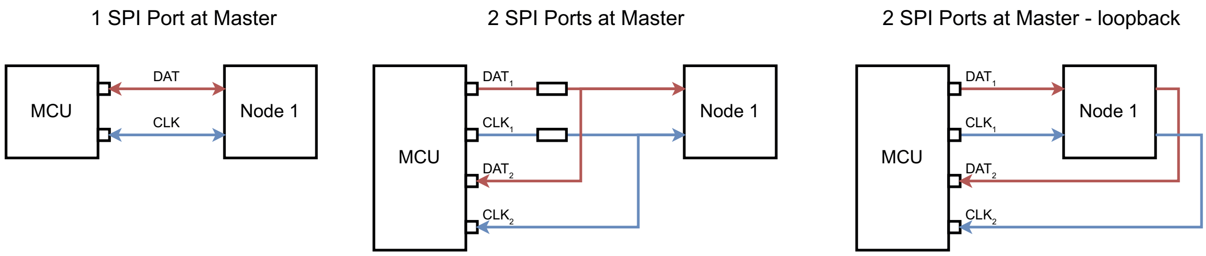
Figure 17: SPI interface for downstream and upstream



The SPI master role is always taken by the transmitting device, i.e., for downstream communication by the MCU and for upstream communication by the first node.

There are three possible options to realize this on MCU side like shown in Figure 18.

Figure 18: MCU options



The first two options work in bidirectional mode, the third option only for the loop-back mode.

When 2 SPI ports are used on the MCU, the first is always master and the second is always slave.The two resistors in the second case are needed to decouple the master and slave pins.

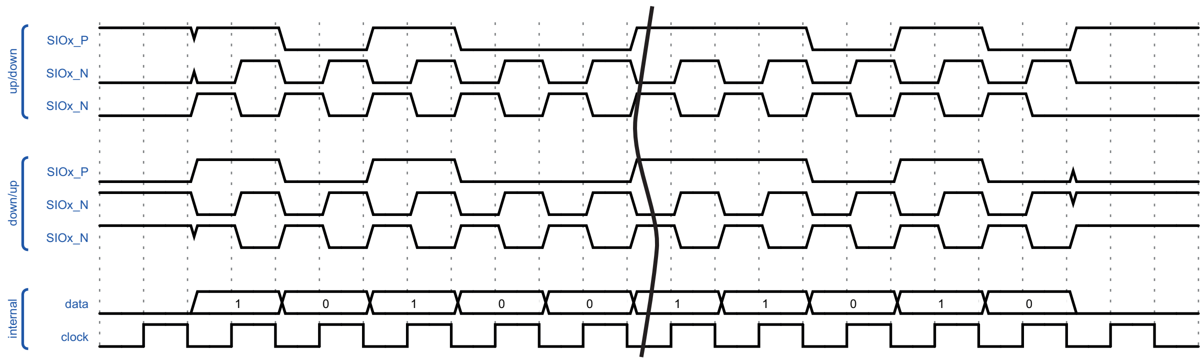
When only 1 SPI port is used, the MCU needs to toggle the SPI role between master and slave after a message has been sent and before the response is received.

There is a wait time of min. 5 µs between the end of an incoming message and the start of a response that is sent back to the master.

Ideally, the clock polarity on SIOx_N is selectable through register settings.

Figure 19 illustrates the possible signals for the two EOL / MCU modes (pull-up on SIO_P and pull-down on SIO_N and vice versa) and the two possible clock polarities, respectively.

Figure 19: Possible signal for EOL / MCU modes



These curves apply both to upstream and downstream communication.

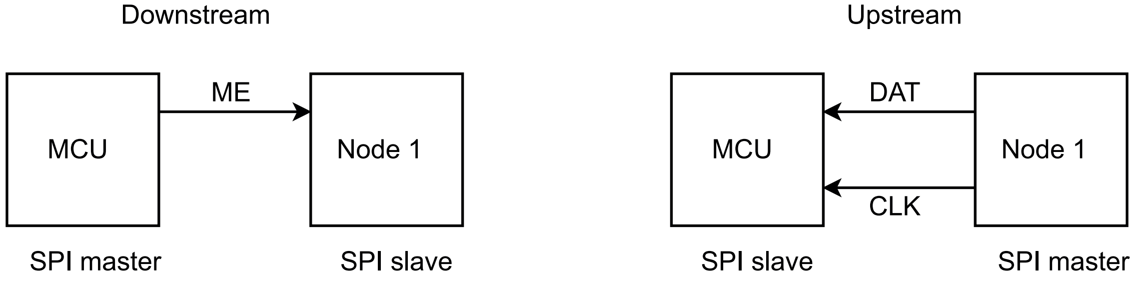
There is NO dedicated start / stop sequence.

The same timing requirements apply as given in chapter "2.7 Parameters" for communication timing.

Mixed SPI

This option is identical to the pure SPI given in the previous section with the modification that the downstream communication (Figure 20) uses a single-wire Manchester encoded signal.

Figure 20: Downstream communication



The advantage lies in a simpler design for the nodes as no additional blocks for receiving are needed (the receiving part is equal to the USE mode). The upstream direction still offers the advantage of easy decoding by the MCU (no Manchester decoder needed).

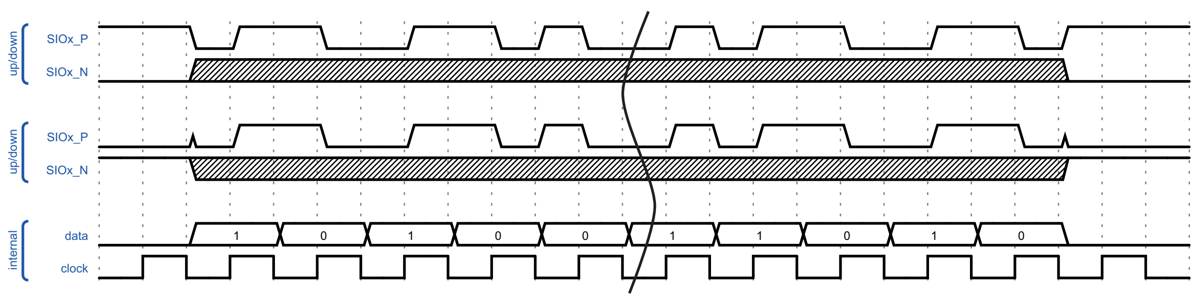
Refer to the previous section for possible connection options.

There is a wait time of min. 5 µs between the end of an incoming message and the start of a response that is sent back to the master.

The downstream signal follows the timing diagram below for the two possible EOL / MCU modes (pull-up on SIO_P and pull-down on SIO_N and vice versa). See chapter "2.6 Manchester coding" for the definition of the Manchester encoding.

During transmission, the signal on SIO_N is ignored. Thus, it is possible to use SPI with twice the nominal data rate to construct the ME signal in the MCU (see Figure 21). The clock signal transmitted on SIOx_N is ignored by the node.

Figure 21: ME signal in the MCU



The upstream signal is identical to the one given in the previous section.

There is NO dedicated start / stop sequence.

The same timing requirements apply as given in chapter "2.7 Parameters" for communication timing. The downstream bit rate needs to be twice the nominal rate, i.e., 4.8 Mbit/s.

Note: When the line is released by the MCU after the transmission, it will approach its idle state as defined by the resistor configuration. Depending on the RC constant of the interface, this transition might be interpreted as an additional data bit by the node. Therefore, it is recommended to either drive and hold the line in a defined state for a period of 4 bits (which is the minimum possible when using SPI with twice the data rate, or 2 bits when using GPIO), or ensure that the node is using the PSI value to determine the message length and ignores any additional bits (see chapter "Additional data after end of message").

ABOUT ams OSRAM Group (SIX: AMS)

The ams OSRAM Group (SIX: AMS) is a global leader in optical solutions. By adding intelligence to light and passion to innovation, we enrich people's lives. This is what we mean by Sensing is Life. With over 110 years of combined history, our core is defined by imagination, deep engineering expertise and the ability to provide global industrial capacity in sensor and light technologies. Our around 24,000 employees worldwide focus on innovation across sensing, illumination and visualization to make journeys safer, medical diagnosis more accurate and daily moments in communication a richer experience. Headquartered in Premstaetten/Graz (Austria) with a co-headquarters in Munich (Germany), the group achieved over EUR 5 billion revenues in 2021. Find out more about us on https://ams-osram.com

For further information on our products please see the Product Selector and scan this QR Code.

Published by ams-OSRAM AG
Tobelbader Strasse 30, 8141 Premstaetten, Austria

Published by ams-OSRAM AG
Tobelbader Strasse 30,
8141 Premstaetten Austria
Phone +43 3136 500-0
ams-osram.com