Draft Standard of SPI protocol(s) for space

Draft Standard 2017-11-24

Doc. No SPI-DS-13

Issue 1.0 Contract 4000114112/15/NL/LF, deliverable D13

This document represents a draft of an ECSS standard which is in development. The document has been written as far as possible according to ECSS drafting rules. The document should not be considered an ECSS standard.

Change log

Issue	Date	Section / Page	Description
0.4	2017-11-14		First draft submitted to ESA. Annexes are to be added.
0.5	2017-11-14	Annex A and B	Added.
1.0	2017-11-24	5.2.2.1 5.3.2.2 5.3.3.3	Suggestions for Backplane connectors. LVCMOS included. Additional requirements for maximum rise/fall time.

TABLE OF CONTENTS

Introduction

General

The Serial Peripheral Interface Bus or SPI bus is a synchronous serial data link de facto standard, named by Motorola that operates in full duplex mode. Devices communicate in master/slave mode where the master device initiates the data frame. Multiple slave devices are allowed with individual slave select (Chip Select) lines.

The positive features of SPI bus that have contributed to its success in the embedded World are several:

- Full duplex communication in the default implementation
- Higher throughput than other serial busses (I²C or SMBus)
- Complete protocol flexibility for the bits transferred (no protocol is defined!)
- Not limited up to to 8-bit words (as RS 232)
- Arbitrary choice of message size, content, and purpose
- Extremely simple hardware interfacing
- Lower power dissipation
- Slaves use the master's clock, (not need of precision oscillator/quartz)
- Slaves do not need a unique address — unlike I²C or GPIB or SCSI
- Transceivers are not needed (SPI is typically used to connect devices on the same board or inside an unit)
- Uses only four pins on IC packages, and wires in board layouts or connectors, much fewer than parallel interfaces
- At most one unique bus signal per device (Chip Select); all others are shared
- Signals are unidirectional allowing for easy Galvanic isolation
- Not limited to any maximum clock speed, enabling potentially high throughput, implementations up to 70Mbps (peak mode) have been developed

SPI bus is characterized also by some weaknesses:

- Requires more pins on IC packages than I²C, even in the three-wire variant
- No in-band addressing; out-of-band Chip Select signals are required on shared buses
- No hardware flow control by the slave (but the master can delay the next clock edge to slow the transfer rate)
- No hardware slave acknowledgment (the master could be transmitting to nowhere and not know it)
- Supports only one master device
- No error-checking protocol is defined
- Without a formal standard, validating conformance is not possible
- Only handles short distances compared to RS-232, RS-485, or CAN-bus

- Many existing variations, making it difficult to find development tools like host adapters that support those variations
- SPI does not support hot plugging (dynamically adding nodes)
- Interrupts must be either be implemented with out-of-band signals or be faked by using periodic polling similarly to USB 1.1 and 2.0

Interesting applications for SPI bus are for example as internal bus for RTU/RIU or used for the configuration or re-programmability of FPGA on board, or as low-medium speed interface for new microcontrollers and microprocessors under development at ESA or already available.

Purpose

Based on development of a demonstrator and a simulation environment, and by tests and simulations proving SPI-related characteristics, the purpose of this draft standard is to propose one or several SPI protocol(s). This draft standard is intended to be an input to future ECSS workgroups.

Guide to this Standard

This Standard begins with clause 1 which introduces the scope of the Standard. Clause 2 then gives a list of applicable documents. Clause 3 provides the necessary definitions of terms and abbreviations and explains the notation used throughout the document. A brief overview of SPI and the Standard is given in clause 4 to familiarize the reader with the basic SPI concepts, prior to the detailed specification of subsequent clauses.

The body of this Standard is presented in clauses 5 to 8, which ascend through the various normative levels of the Standard.

Clause 5 (Physical Layer) covers cables, connectors, cable assemblies and printed circuit board tracks.

Clause 6 (Data Link Layer) describes the different levels of protocols the standard covers.

Clause 7 (Network Layer) describes how data and control command packages are defined.

Clause 8 (Redundancy) gives a description of redundancy concepts.

Clause 9 includes a list of informative references is included in the Bibliography.

There are two annexes:

- Annex A: Terminations in SPI Networks
- Annex B: SPI Bus Topologies

1 Scope

The purpose of this document is to define a formal standard for SPI. The definition of a formal standard helps to construct reliable systems, interoperable units, reusability and for validating conformance.

The following layers for the SPI communication are defined in this standard:

- Physical layer
- Data link layer
- Network layer

Physical layer

This layer defines connectors, cables, cable assemblies, voltage levels, noise margins, PCB tracks, signal encoding, data transfer rate and redundancy solutions.

Three different alternatives for the electrical signals to be transferred, called routing cases, are defined:

- PCB internal communication
- PCB to PCB via backplane
- Cable, using LVDS

Data link and Network layer

Different communication protocols are defined to provide a generic solution for existing devices and a separate protocol for high or medium traffic load with detailed data and network layer definition.

This layer provides data word formats, message formats defining commands and responses, data integrity and redundancy solutions.

This standard may be tailored for the specific characteristic and constrains of a space project in conformance with ECSS-S-ST-00 [1].

2 Normative references

The following normative documents contain provisions which, through reference in this text, constitute provisions of this ECSS Standard. For dated references, subsequent amendments to, or revision of any of these publications do not apply. However, parties to agreements based on this ECSS Standard are encouraged to investigate the possibility of applying the more recent editions of the normative documents indicated below. For undated references, the latest edition of the publication referred to applies.

[1]	ECSS-S-ST-00-01	ECSS system - Glossary of terms

3 Terms, definitions and abbreviated terms

3.1 Terms from other standards

For the purpose of this Standard, the terms and definitions from ECSS-S-ST-00-01 [1] apply.

3.2 Terms specific to the present standard

3.2.1 byte

eight bits of data

3.2.2 master

The device issuing the clock, Chip Select, driving the data using MOSI, receiving the data using MISO and managing the communications speed.

3.2.3 slave

The device receiving the clock and Chip Select, driving the data using MISO, receiving the data using MOSI.

3.2.4 command token

The master transmits a message header which specifies the action need to be performed in slave.

3.2.5 response token

The slave transmits a message header which consist of status of module and details of error occurred.

3.3 Abbreviated terms

Abbreviation	Meaning
A/D	analogue-to-digital
ADC	analogue to digital converter
CLK	clock
CPOL	clock polarity
CPHA	Clock phase
CRC	cyclic redundancy check
CS	Chip Select

Abbreviation	Meaning
DAC	digital to analogue converter
EDAC	error detection and correction
EEPROM	electrically erasable programmable read-only memory
ESA	european space agency
FPGA	field programmable gate array
I/F	interface
kbps	Kilobits per second
LVDS	low voltage differential signalling
LVTTL	low voltage transistor-transistor logic
Mbps	megabits per second
MRAM	magnetoresistive random-access memory
MOSI	master out slave in
MISO	master in slave out
MSB	most significant bit
ns	nanosecond
NVRAM	non-volatile random access memory
OSI	open system interconnection
ps	picosecond
RTU	remote terminal unit
SCK	serial clock
SPI	serial peripheral interface
SW	software
TBC	to be confirmed
TBD	to be defined

3.4 Conventions

3.4.1 Bit Numbering

The following conventions are used for bit numbering:

- The Most Significant Bit (MSB) of a vector has the leftmost position.
- The Least Significant Bit (LSB) of a vector has the rightmost position.
- Unless otherwise indicated, the MSB of a vector has the highest bit number and the LSB the lowest bit number.

3.4.2 Radix

The following conventions is used for writing numbers:

- Binary numbers indicated by the subscript b, e.g. 1b, 1011_1110b, 110b.
- Hexadecimal numbers indicated by the prefix 0x, e.g. 0x0E16BABE.
- Decimal number have no prefix.
- Unless a radix is explicitly declared, the number should be considered a decimal.

4
Overall description

4.1 Use cases

4.1.1 Overview

A number of possible use cases for SPI bus in space is foreseen. These are used to derive the requirements for the protocol(s) and physical layer(s).

The following cases for SPI use in Space are considered:

- Internal backplane bus inside a unit,
- Interface using cable
- Serial bus for reprogrammable logic on board (FPGA, NVRAM),
- Interface with sensors,
- Interface with standard ICs (ADC, DAC).

These are described further in the subsections below.

4.1.2 Internal backplane bus inside a unit

The single master multiple slave configuration of SPI fits to connect multiple systems inside a backplane unit. The host system in the unit may only have a single SPI master and each of the peripherals can have multiple slaves.

The host and peripheral system can take advantage of the full duplex mode of communication and transfer simultaneously. This can for example be used for transferring status interface simultaneously between two redundant systems performing the same work.

The SPI in the backplane can be used to cater medium to high traffic load between two processor nodes or to quickly initialize and configure the peripheral units.

The operations to be performed over the backplane bus vary greatly depending on the design of the unit and include:

- Simple peripheral control via memory mapped registers
- Data transfers between memories connected to different devices within the unit
- Handling of in-memory descriptors in the target system to target system communication controllers.

In order to allow efficient implementation of target devices within the unit, the protocol should allow efficient hardware implementation of the target device with full protocol decoding in hardware. The SPI in the backplane can also be connected in a way to achieve redundancy between the masters and the slaves. The masters from different host systems and the slaves in the peripheral system can be connected in redundant fashion.

The Onboard time from masters can be distributed and synchronized to slaves using dedicated SPI commands for initialization and distribution. The time required for a transfer to complete can be accurately calculated and utilized to perform accurate time keeping between the modules.

4.1.3 Interface using cable

The SPI can be used to interface two units using a cable. A point-to-point interface transmitting from one master to another slave can make use of SPI. Several devices have dedicated master and slave SPI controllers where a master send streams of data to the slave.

For the cable lengths the following are considered:

- 10 cm (simulating unit interconnection cases)
- 1 m (simulating unit to unit cases)
- 10 m (simulating cases where SPI sensor nodes are located in far distance)

4.1.4 Serial bus for reprogrammable logic on board

Several memory devices (Flash, MRAM, EEPROM...) already available make use of SPI for communication. The serial interface for memory with less pin count for connection clearly provides an advantage when connecting the memories in redundant configuration. By defining a suitable protection scheme (EDAC without software intervention) SPI Flash devices can also provide the reliability required. There are also Field Programmable Arrays where the fabric can be configured via SPI.

4.1.5 Interface with sensors

Extremely simple interface is only required for communication when SPI is used. The Sensor interface can be designed to perform only half duplex mode of operation. The master only need to send the clock periodically and the slave sends the data back to the master. The periodic fetching of data and minimum logic needed will result in low power consumption.

SPI is an attractive bus for interfacing multiple sensors. Several existing sensors (Temperature, Pressure, Altitude, Humidity, Accelerometers, Position, Touch, and Current) make use of SPI. A microcontroller/FPGA can easily interface these sensor units as slave devices. SPI interface also allows use of multiple independent sensors for applications requiring redundancy.

A typical sensor interface using SPI is shown in Figure 1. Also by interfacing an ADC several other sensors can be interfaced.

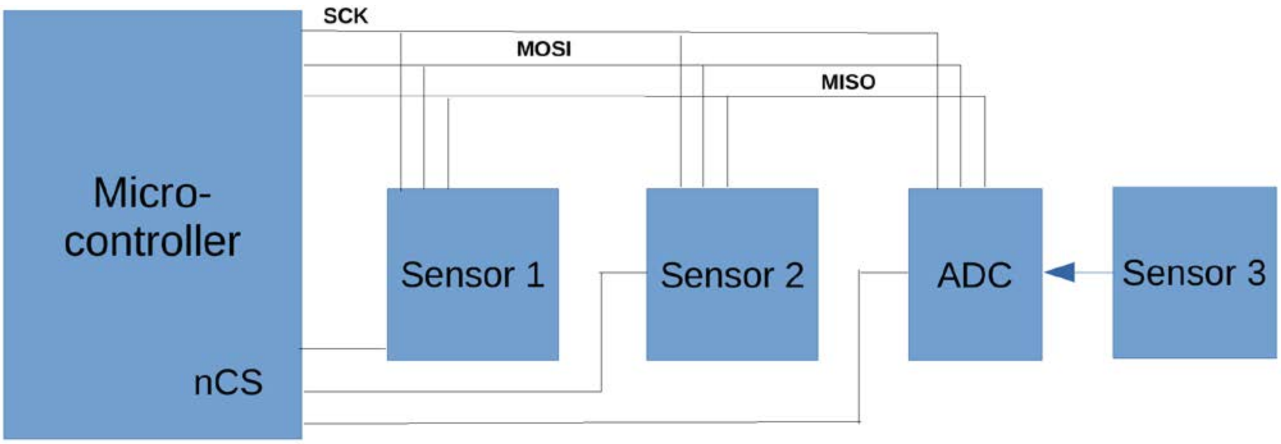


Figure 1 Typical SPI Configuration

4.1.6 Interface with standard ICs

Several components (Ethernet controller with SPI interface, Real-time clock, Switches Counters, LED drivers) make use of SPI. Multifunction ICs (Real-time clock+ SRAM) with SPI are also available. The SPI interface only require four pins to interface with IC packages, further reduction in the pin count is possible when acting in half duplex mode. For example in a DAC the master can send only the clock and data to the slave. Similarly in an ADC the master can send only the clock and acquire data from the slave.

The standard ICs have fixed electrical characteristics and existing protocols, the approach while defining the protocol would be to develop a separate SPI for space protocol which is generic enough so that they can be used to interface existing devices.

4.2 SPI signals

4.2.1 Introduction

The SPI bus is a full-duplex synchronous serial bus. Transmission starts when a master selects a slave through the slave’s chip (or slave) select signal and the clock line (SCK) transitions from its idle state. Data is transferred from the master through the Master-Output-Slave-Input (MOSI) signal and from the slave through the Master-Input-Slave-Output (MISO) signal.

The signals of the SPI protocols are:

- Serial Clock (SCK)
- Master Out Slave In (MOSI)
- Master In Slave Out (MISO)
- Chip Select(s) (nCS#)

These are described more in detail in the subsections below.

4.2.2 Serial Clock

The SCK is driven by a master providing the serial clock to all the slave devices. Based on this SCK the master transfers (using MOSI) and receives (using MISO) data. Similarly the slave transfers (using MISO) and receives (using MOSI) data. All communication is initialized by master.

4.2.3 Chip Select

An independent Chip Select (nCS#) signal is provided to each slave that is driven only by the master device. The Chip Select signal marks the start and end of a transmission. The master deselects the slave using Chip Select after the end of the transmission.

If the SPI slave is not activated using Chip Select, it ignores the clock and MOSI signal from the master. The SPI slave which is not activated using Chip Select, does also not drive the MISO signal.

4.2.4 MOSI and MISO

The implementation supports full duplex communication. In the master, the MOSI line is driven to communicate data to the selected slave. The MISO line is driven by the selected slave to communicate its output data back to the master device.

4.3 SPI protocol variants

Different protocols are defined to cover the different applications or use cases. The purpose of this strategy is to make communication possible with the existing devices using a generic protocol and define a separate protocol for high or medium traffic load (CPU communication or backplane bus) with data and network layer definition [RD1].

The protocol SPI-0 is generic and to be used for interfacing existing devices. New devices can be implemented with only the required features using this protocol.

The protocol SPI-1 targets 8, 16 and 24 bit data word communications. To guarantee the quality of data transmission a Parity bit is included at SPI word level.

The protocol SPI-2 targets 16-bit data word communications. This protocol provides detailed data and network layer definition. Error detection bits are included in the message transferred between the master and slave. A CRC checksum is appended at the end of the SPI messages.

4.4 Physical layer

The SPI bus signals shall be routed along PCB traces and/or over harness (cable). The following table lists the alternative routing cases.

Routing case	Type of connection
1	PCB internal
2	PCB to PCB via backplane
3	Cable using LVDS

Table 1 Routing cases

5
Physical Layer Specifications

5.1 General

In this section, three different alternatives for the electrical signals to be transferred, called routing cases, are defined.

5.1.1 Routing

a. The different routing scenarios considered by this standard are defined in the table below.

Routing case	Type of connection
1	PCB internal
2	PCB to PCB via backplane
3	Cable using LVDS

Table 2 Routing cases

5.2 Mechanical Specifications

5.2.1 Connectors

5.2.1.1 Micro-D (Routing case 3)

a. Connectors between PCB and cable shall be D-SUB connectors of Micro D type.

b. The cable assemblies shall have connectors with male contacts.

c. The circuit boards to which cables are expected to be attached shall have connectors with female contacts.

5.2.1.1.1 Pin assignments

a. The connector contact identification given in the tables and figure below shall be used.

b. Note: The shield should be connected to signal ground according to the EMC requirements of the mission.

Contact number	Signal name
1	SCK+
2	MOSI+
3	Circuit Ground
4	MISO-
5	CS-

Contact number	Signal name
6	SCK-
7	MOSI-
8	MISO+
9	CS+

Table 3 Connector with signal identification

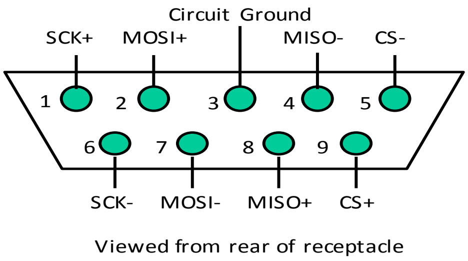


Figure 2 Connector viewed from rear of receptacle

Contact Number	Signal (Master)	Direction (Master)	Direction (Slave)
1	SCK+	Output (Tx)	Input (Rx)
2	MOSI+	Output (Tx)	Input (Rx)
3	Circuit Ground	NA	NA
4	MISO-	Input (Rx)	Output (Tx)
5	CS-	Output (Tx)	Input (Rx)
6	SCK-	Output (Tx)	Input (Rx)
7	MOSI-	Output (Tx)	Input (Rx)
8	MISO+	Input (Rx)	Output (Tx)
9	CS+	Output (Tx)	Input (Rx)

Table 4 Signals from SPI master and slave devices

Note 1 The SPI Master interface has three output pair and one input pair, which corresponds to three input pair and one output pair at the slave. The assignment given allows a 1:1 cable correspondence for the cabling between a Master and Slave, which gives a clear and logical mapping.

5.2.1.1.2 Cables and cable assemblies

a. A SPI cable shall be of twisted pair cable type with an outer shield.

Note 1 Using a cable with an outer shield, the SPI signals will be less susceptible to disturbances from external sources compared to an unshielded cable. The radiated emissions from the SPI cable are also largely reduced through the use of an outer shield.

b. No internal shield is used between signal pairs.

Note 1 Not using inner shield results in lower cable cost, which is important for many applications of the SPI protocol.

c. The different twisted pairs should have separate twist rates.

Note 1 Different twist rates minimize cross talk between adjacent signals.

5.2.1.1.3 Cable length

a. The maximum length of the cable assembly shall be 10 m.

Note 1 Longer cables will limit the maximum operational SPI frequency.

5.2.2 Backplane connectors (Routing case 2)

5.2.2.1 Pin assignments

a. Different types of connectors can be used, no matched impedance connectors are needed for SPI, in case of 2mm cPCI connector the following pin function is suggested below.

	Contact Number	**Signal**	**Contact Number**	**Signal**	**Contact Number**	**Signal**	**Contact Number**	**Signal**
Row1	A1	GND	B1	GND	C1	VCC	D1	GND
Row2	A2	GND	B2	**CS**	C2	GND	D2	GND
Row3	A3	GND	B3	GND	C3	**MISO**	D3	GND
Row4	A4	GND	B4	**MOSI**	C4	GND	D4	GND
Row5	A5	GND	B5	GND	C5	**SCK**	D5	GND
Row6	A6	GND	B6	VCC	C6	GND	D6	GND

Table 5 Signal arrangement for a typical Backplane connector

Note 1 SPI signals should be assigned to the inner section of the connector in order to be separated from each other by adjacent GND pins and the outer rows can also assigned to GND. It is expected that this potentially reduce any cross talk between the signals. Two of the outer most pins on the connectors are assigned to +3V3 in order to provide a power supply for the daughter cards.

Note 2 The proposed pin function relies on use of 2mm - 4 columns type of connectors (e.g. 2mm/cPCI from smith's connector).

5.2.3 Printed circuit board tracks

b. The maximum total trace length, per unit including stubs in a backplane configuration, shall be according to the table below.

Routing case	Max length
1	1 m
2	1 m

Table 6 Routing lengths for different routing cases

5.3 Electrical Specifications

5.3.1 General

In this section the electrical specifications of the physical layer are defined.

5.3.1.1 Topology

a. A device communicating on the SPI bus shall be one of the following:

i. A bus controller, also referred to as "master" (the device issuing the clock and managing the communications speed)

ii. A "slave"

b. Only one master device shall be present on a SPI bus.

c. The total number of slave devices present on a SPI bus shall be according to the table below.

Routing case	Max number of slaves
1	Application-specific, set by user
2	Application-specific, set by user
3	1 (Point to point)

Table 7 Max number of slaves for different routing cases

d. A linear multi-drop topology shall be adopted for the SPI bus (routing case 1-2).

Note 1 Note See Figure 3 and Figure 4 below for SPI slave device arrangements.

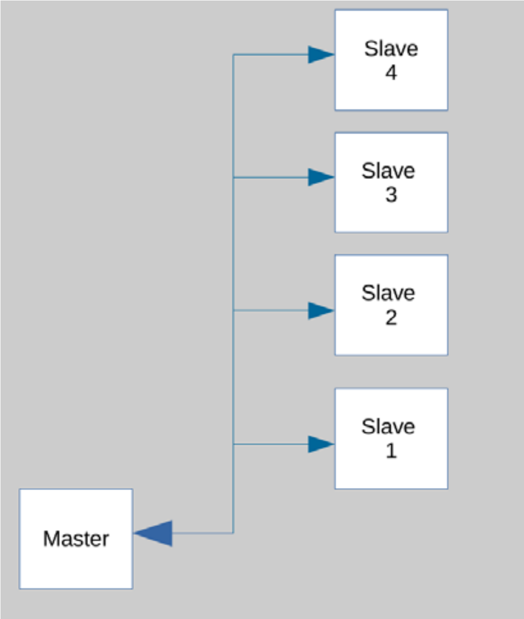


Figure 3 Connection from SPI master to network of SPI devices

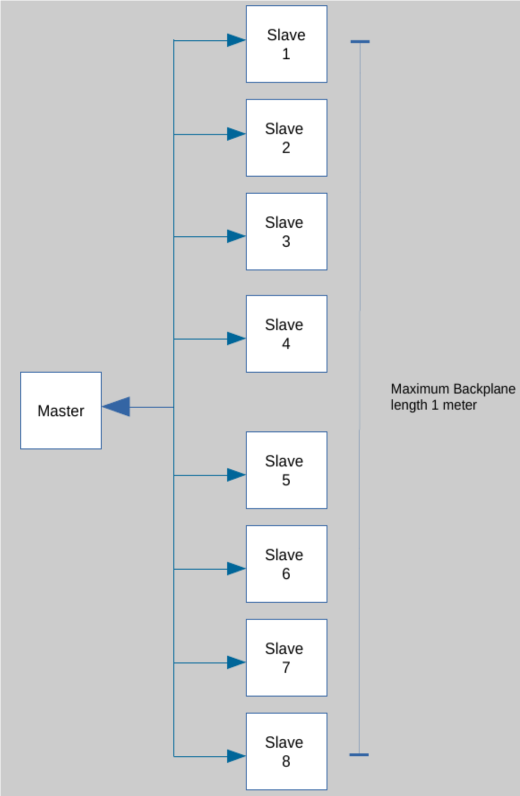


Figure 4 Connection from SPI master to network of SPI devices (T-topology)

5.3.1.2 Bus signals

a. The signals of the SPI protocols shall be

  i. Serial Clock (SCK)

  ii. Master Out Slave In (MOSI)

  iii. Master In Slave Out (MISO)

  iv. Chip Select(s) (nCS#)

  Note 1 Number of Chip Select shall be equal to the number of slaves connected to the master.

b. All the SPI signals shall be unidirectional.

c. The SCK shall only be driven by the master device providing the serial clock to the slave device(s).

d. The MOSI line shall be driven by the master to communicate data to the selected slave.

e. The MISO line shall be driven by the selected slave to communicate its output data back to the master device.

f. An independent Chip Select (nCS#) signal shall be provided to each slave which is driven by the master device.

g. The Chip Select signal shall mark the start and end of a transmission.

h. A slave which is not activated using Chip Select shall disregard the clock and MOSI signal from the master.

i. A slave which is not activated using Chip Select shall not drive the MISO signal.

j. The message shall be discarded if the Chip Select goes inactive in the middle of a transfer.

5.3.1.3 Configuration

a. The master shall configure the SCK frequency less than or equal to the frequency that a respective slave can support.

b. The SCK frequency shall not be changed during a single message transfer between the master and a slave.

c. The master device shall have the possibility to configure active high or active low Chip select.

d. During a transmission on the SPI bus data shall be either changed or read at a transition of SCK.

  i. If data read at edge n, data shall be changed at edge n+1.

ii. If data read at the first transition of SCK the transfer has CPHA 0, and if data is changed at the first transition of SCK the transfer has CPHA 1.

e. The idle state of SCK shall be either high or low.

i. If the idle state of SCK is low then the transfer has CPOL is 0

ii. If the idle state of SCK is high then the transfer has CPOL is 1.

iii. The combined values of CPOL and CPHA determine the mode of the SPI bus.

Mode	CPOL	CPHA
0	0	0
1	0	1
2	1	0
3	1	1

Table 8 Possible modes of operation

f. The SCK, MOSI and MISO signals shall be possible to configure in the four different modes (Mode 0, 1, 2, 3) using parameters CPOL and CPHA, as illustrated in Table 8 and Figure 5,

Note 1 The figure illustrates an example of one byte (0x55) being transferred MSB first (to the left) over the SPI bus under the four different modes. The behaviour of MISO shall be the same as for the MOSI signal.

Note 2 The CPHA = 0 means the devices shall have data ready before the first transition of SCK.

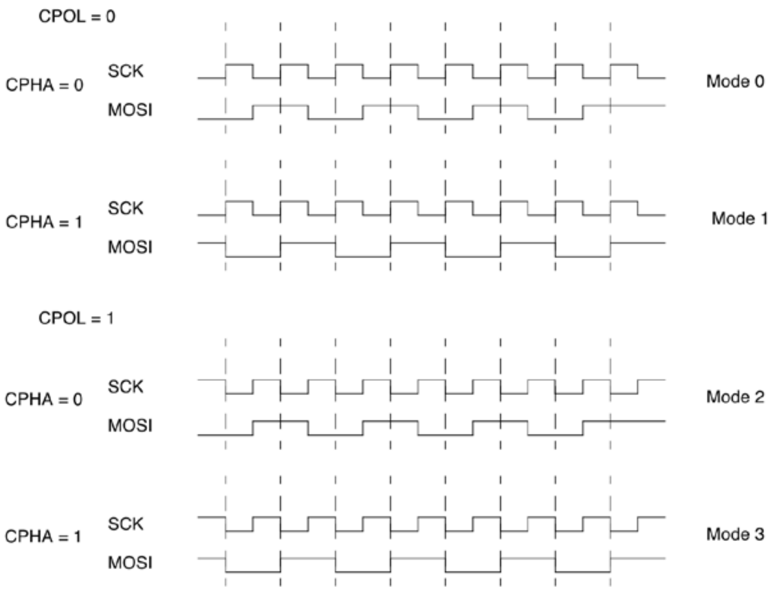


Figure 5 SPI modes, example 0x55 being transferred MSB first

g. The master and slave shall use the same SCK frequency, CPOL and CPHA.

h. If multiple slaves are connected the master shall address each slave when configuring the parameters (SCK frequency, CPOL and CPHA).

5.3.1.4 Timing

a. The in-message signalling rate absolute fluctuations, i.e. variation of serial clock period time, shall be limited to ±10%.

b. A SPI slave device shall respond to a power-on event and become operative within 500 ms, where a power-on event is either when power supply is being applied to the device, or when an external reset signal is being asserted.

5.3.1.5 Performance

a. The reciprocal cross-talk between the PCB lines of the SPI bus shall be kept below -20 dB in the frequency range 100 kHz to 200 MHz.

5.3.2 DC Specifications

5.3.2.1 Power Supply

a. Nominal power supply for the SPI bus shall be 3.3V ± 10%.

5.3.2.2 Signal Levels

a. All SPI lines shall use unipolar NRZ coding.

b. The Chip Select signal active polarity must be either

   i. driven low to activate a slave and high to deactivate

   ii. driven high to activate a slave and low to deactivate

c. The SPI lines SCK, MOSI and MISO shall adopt active-high logic levels.

d. Referring to section 5.1.1, SPI bus signals shall adhere to the DC levels in the table below, corresponding to standard 3.3 V LVTTL/LVCMOS logic levels.

Parameter	**Route Case 1-2**
V_{OL}	≤ 0.4 V
V_{OH}	≥ 2.4 V
V_{IL}	≤ 0.8 V
V_{IH}	≥ 2.0 V

Table 9 SPI bus signal levels different Routing Cases 1-2

e. Referring to section 5.1.1, SPI bus signals shall adhere to the DC levels in the table below, corresponding to standard LVDS levels.

Parameter	Route Case 3
V_{OL}	$\leq$ -0.25 V_{diff}
V_{OH}	$\geq$ 0.25 V_{diff}
V_{IL}	$\leq$ -0.10 V_{diff}
V_{IH}	$\geq$ 0.10 V_{diff}

Table 10 SPI bus signal levels Routing Cases 3

f. An SPI receiver shall tolerate the presence of a common mode noise signal in the range ±1V, peak-to-peak in the frequency range 0-30 MHz, without misinterpreting the signal.

Note 1 This requirement is valid for Routing Case 3.

g. An SPI receiver shall tolerate the presence of a peak-to-peak noise signal with the amplitude of -10 mV to +10 mV peak-to-peak in the frequency range 0-30 MHz, without misinterpreting the signal.

Note 1 This requirement is valid for Routing Case 3.

h. When any of the following fault condition occurs, the voltage level at the receivers input shall be comprised in the range -10 mV to +10 mV.

i. Driver not operated

ii. Driver disabled

iii. Driver not connected to receiver

iv. Receiver input open circuit

v. Receiver inputs shorted together

Note 1 This requirement is valid for Routing Case 3.

5.3.2.3 Loads

a. When a driver is not powered, its output shall be DC high impedance (> 100 kΩ).

b. When a receiver is not powered, its input shall be DC high impedance (> 100 kΩ).

5.3.3 AC Specifications

5.3.3.1 Bit Rate

a. Minimum bit rate for the SPI bus shall be according to the table below.

Routing case	Bit rate
1	100 kbps
2	100 kbps
3	100 kbps

Table 11 Bit rates for different routing cases

Note 1 Actual maximum bit rate is not defined by this standard. The limit is determined by a slave read action. The maximum SCK frequency depends on delays in the master, delays between master and slave (including buffer delays – if used, distance between the master and slave and media characteristics), slave response time and master set-up time.

Note 2 Sizing of an SPI network should take all these delays in account, observing that the criteria for functional SPI operation is that the MISO signal from the slave returns to the master before the reading edge of SCK.

5.3.3.2 Clock accuracy

a. The clock frequency shall have an absolute error of maximum ±5%.

b. The clock frequency shall have a temperature dependency of maximum ±2% over a temperature range of at least from -40°C to +60°C.

c. Ageing degradation of the clock frequency shall be maximum ±50 ppm/month.

d. Any SPI clock signal shall present a positive duty cycle of 40-60%.

5.3.3.3 Timing

a. For single-ended applications (i.e. Routing Cases 1-2) any signal at the receiver end should fulfil the rise/fall time characteristics in the table below, where TUI – the unit interval, refers to half clock period time.

Parameter	Min (ns)	Max (ns)
t_{rise} / t_{fall}	N/A	0.3 T_{UI}

Table 12 Rise and fall times for Routing Cases 1-2

Note 1 Due to a single ended network not being matched in driver and receiver ends, the signal at the receiver end will be dependent on network characteristics and driver termination.

b. For LVDS applications (i.e. Routing Cases 3) the SPI drivers should fulfil the rise/fall time characteristics in the table below, where TUI – the unit interval, refers to half clock period time.

Parameter	Min (ns)	Max (ns)
t_{rise} / t_{fall}	N/A	0.3 T_{UI}

Table 13 Rise and fall times for Routing Case 3

Note 1 As LVDS applications utilize an impedance matched driver and receiver, the signal quality is primarily determined by driver characteristics.

c. Max rise/fall time shall not be larger than 50 ns TBC in order to avoid glitches and burst at the receiver input.

d. Setup and hold times shall be respected with margins with respect to the clock.

5.3.3.4 Conductor impedances

a. Each single ended SPI PCB trace shall have impedance, with respect to ground in the range 45 Ω to 55 Ω in the frequency range 100 kHz to 200 MHz.

b. Each SPI differential trace and cable shall have differential impedance in the range 85 to 115 Ω in the frequency range 100 kHz to 200 MHz.

Note 1 The PCB traces for the Routing case 3 circuit board shall have a characteristic single-ended impedance of 50 Ω. After the LVDS circuits these are converted to differential signals and shall have a differential impedance of 100 Ω. To reach a differential impedance of 100 Ω, the each of the +ve and -ve traces shall have single-ended impedance equal to about half of the differential impedance i.e. 50 Ω.

c. Each SPI driver output should be source impedance terminated in the frequency range100 kHz to 200 MHz.

d. Each SPI driver output shall be short circuit protected to avoid failure propagation in the case of a short circuit at the receiver input.

Note 1 Resistive series termination can be used as an effective means of short circuit protecting the SPI driver.

6 Data Link Layer Specifications

6.1 Protocols

A master or a slave device can make use of any of the protocol variant defined below.

6.1.1 SPI-0 protocol

a. Word width shall be as per the requirements of the implementers.

Note 1 Recommended word width shall be 8 bits.

b. Word bit ordering (MSB first or LSB first transferred) shall be as per the implementers.

Note 1 Recommended bit ordering: MSB transferred first and LSB last.

Direction of transfer

←

D0	D1	D2	D3	D4	D5	D6	D7
1	0	1	0	1	0	1	0

Table 14 Example 8 bit data 0x55 transfer, LSB first

Direction of transfer

←

D7	D6	D5	D4	D3	D2	D1	D0
0	1	0	1	0	1	0	1

Table 15 Example 8 bit data 0x55 transfer, MSB first

c. The contents of the message, type of message, and number of words per message shall be as per the requirements of the implementers.

d. The implementers can choose any data integrity solutions.

i. Recommended solution is to use parity at word level (odd or even).

ii. Recommended solution is to use checksum (CRC) (at message level) appended at the end of the SPI message. Checksum covers the group of words (message).

iii. The implementation of any data integrity solution is optional.

6.1.2 SPI-1 protocol

a. Word width shall be 8, 16 or 24 bits.

b. Word bit ordering shall be MSB transferred first and LSB transferred last.

c. Parity bit shall be included at word level. The master device shall have the option to choose odd or even parity.

Direction of transfer
←

D7	D6	D5	D4	D3	D2	D1	D0	P
0	1	0	1	0	1	0	1	1

Table 16 Example 8 bit data 0x55 transfer, odd parity, MSB first

6.1.3 SPI-2 protocol

a. Word width shall be 16 bits.

b. Word bit ordering shall be MSB transferred first and LSB transferred last.

Direction of transfer
←

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Table 17 Example 16 bit data 0x5555 transfer, MSB first

6.2 Loop Back

a. A simple loop back internal to the master device may be provided.

b. MOSI should be connected to MISO, when conditions for transmitting the data are set in master, and also loop back is enabled, the transmit register content should be transferred to receive register in the master.

7
Network Layer Specifications

7.1 SPI-2 protocol

7.1.1 Overview

The message format transferred between a master and a slave SPI device is defined below. The message header composed of Command token for the master and Response token for the slave. The message also contains optional data words and CRC checksum appended at the end which are calculated for the data words transferred. The CRC is mandatory, if the message contains payload data then the message shall be appended with one word of CRC. The received messages shall be processed by the SPI slave device and response and data shall be transferred as per the received command. Some of the status bits in the response token are status for the previously received command.

Message format

The information between a master and slave is transferred simultaneously. The SPI requires a word transmitted to receive a word. The example message format for a write and read is shown below.

Signal	Message Header		Payload	Payload CRC
MOSI	Command #1	Command #2	Data	CRC-16
MISO	Response #1	Response #2	0x0000	0x0000

Table 18 Example Message format (Write data)

Signal	Message Header		Payload	Payload CRC
MOSI	Command #1	Command #2	0x0000	0x0000
MISO	Response #1	Response #2	Data	CRC-16

Table 19 Example Message format (Read data)

7.1.2 Message Header - Command Token

The master transmits a message header which specifies the action need to be performed in slave. The message header sent by master device is called command token which consist of two 16-bit words. The message header content details are explained below.

MSB	Command Token Word #1														LSB
Prefix		Command Code						Spare		Message Length					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
'0'	'1'	C5	C4	C3	C2	C1	C0	'1'	'1'	L5	L4	L3	L2	L1	L0

Table 20 Command word #1

MSB	Command Token Word #2														LSB
Prefix		Sub-address								Spare		CRC-4			
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
'0'	'1'	SA7	SA6	SA5	SA4	SA3	SA2	SA1	SA0	'1'	'1'	CRC3	CRC2	CRC1	CRC0

Table 21 Command word #2

7.1.2.1 Prefix and spare

a. The prefix bits shall be transmitted initially.

b. In the command word#1 and #2 the prefix and spare bits shall have fixed value.

7.1.2.2 Command Code

7.1.2.2.1 General

a. The command code shall specify the operating instruction for the receiving slave.

7.1.2.2.2 Reset command

Code	Command	Length	Sub-address	Payload	Description
0x00	RESET_SPI	0x00	0x00	None	This command shall be used to reset the controller to a power up initialized state.

Table 22 Reset command

a. The SPI slave resets the system only if a valid command is received.

b. If the prefix and spare bits does not match the expected value, then the RESET_SPI command shall be discarded.

c. If the calculated CRC-4 for the command token does not match, then the RESET_SPI command shall be discarded.

7.1.2.2.3 Synchronisation command

Code	Command	Length	Sub-address	Payload	Description
0x07	SYNCH	0x04	0x00	MOSI: <SYNC1><SYNC2> <SYNC3><SYNC4> <CRC-16> MISO: <all zeros>	The slave shall receive a command token followed by payload words of length 4 containing synchronization information.

Table 23 Synchronisation command

a. The words containing timing information for the slave shall be transferred.

b. These words shall be copied into dedicated time register implemented in the SPI slave device after validation.

c. The time register is of 4 words (64 bits), the most significant time is transferred first in SYNC1 followed by SYNC2, SYNC3 and SYNC4.

Note 1 By providing 64 bits for time register four bytes of coarse and four bytes of fine time can be implemented [RD2].

d. The time shall be synchronized only when all the 4 SYNC words are received and also the command token CRC-4 and payload words CRC-16 are valid.

e. The time shall be synchronized only when the command token CRC-4 and payload words CRC-16 are valid.

f. The RESET_SPI command may not reset the time register.

7.1.2.2.4 Tick command

Code	Command	Length	Sub-address	Payload	Description
0x08	TICK	0x00	Used as index for increment	None	The command shall be used to advance the time register in the SPI slave device.

Table 24 Time increment command

a. A valid command increments the implemented time register.

b. The sub-address shall specify the time register bit (index) which shall be incremented.

7.1.2.2.5 Read back command

Code	Command	Length	Sub-address	Payload	Description
0x0A	READ-BACK_CMD	0x02	0x00	MOSI: <all zeros> MISO: <CMDTOKEN> < CRC-16>	The command shall be used to verify the correct reception of the previous command. Upon reception of this command the slave shall respond with previous command token.

Table 25 Read back previously sent command

a. The slave device after receiving the READBACK_CMD shall send the previous command token transmitted by the master.

Note 1 This command is useful only when commands other than RESET_SPI was previously transmitted. If the previous command is RESET_SPI, the slave device may transfer only zeros or token with RESET_SPI command in the payload section of this command.

7.1.2.2.6 Read data command

Code	Command	Length	Sub-address	Payload	Description
0x0E	READ_SA	Number of words to be read	SA	MOSI: <all zeros> MISO: <DW1> <DW2>…. <DWN> <CRC-16>	The command shall be used to read data words from a slave. Dedicated "Length" and "Sub-address" field of the command token shall provide the payload length and the target address.

Table 26 Read data command

a. When a valid READ_SA command is received, the payload data shall be transferred from the address specified.

b. The address for reading the data shall be calculated using the sub-address.

c. The payload CRC shall be calculated for payload data and sent after the payload data is transferred by the slave device.

7.1.2.2.7 Write data command

Code	Command	Length	Sub-address	Payload	Description
0x0D	WRITE_SA	Number of words to be written	SA	MOSI:<DW1> <DW2> …<DWN> <CRC-16> MISO: <all zeros>	The command shall be used to write data words into a slave. Dedicated “Length” and “Sub-address” field of the command token shall provide the payload length and the target address.

Table 27 Write data command

a. When a valid WRITE_SA command is received, the payload data shall be transferred to the address specified.

b. The address for writing the data shall be calculated using the sub-address.

c. The payload CRC shall be calculated for payload data and sent after the payload data transferred by the master device.

d. The payload CRC shall be calculated for the received words and compared with the received payload CRC from the master

e. If a payload CRC error is detected, then message error status shall be enabled and transmitted to master as part of the next response token.

7.1.2.2.8 Configure access address commands

Code	Command	Length	Sub-address	Payload	Description
0x20	CONFIG-WRITE_ADDR	0x02	0x00	MOSI: <CW1><CW2> <CRC-16> MISO: <all zeros>	The command shall be used to notify the slave about the address to which the data from the master shall be written, used for WRITE_SA command.
0x21	CONFIG READ_ADDR	0x02	0x00	MOSI: <CR1> <CR2> <CRC-16>	The command shall be used to notify the slave about the address from which the data to the master shall be read, used for READ_SA command.

Code	Command	Length	Sub-address	Payload	Description
				MISO: <all zeros>	

Table 28 Configure access address command

a. The most significant word CW1 (or CR1) shall contain the most significant byte of the target address.

Note 1 The purpose of this command is to provide dedicated write and read access address.

Note 2 This enables the master device to access 32 bits of address space in a slave device.

7.1.2.2.9 Redundancy commands

Code	Command	Length	Sub-address	Payload	Description
0x24	ACTIVATE	0x00	0x00	None	The command shall be used to activate the redundant slave interface. This command shall not activate the interface in which it is receiving.
0x25	DEACTIVATE	0x00	0x00	None	The command shall be used to deactivate the redundant slave interface. This command shall not deactivate the interface in which it is receiving.

Table 29 Redundant port - activate and deactivate command

Note 1 See clause 8.2.2 for detailed usage of these commands

7.1.2.2.10 Illegal commands

Code	Command	Length	Sub-address	Payload	Description
All others	N/A	0x00			These command codes are reserved for future issue of the SPI in-space specification. Upon reception of these codes the slave shall discard the incoming data and set ILLEGAL_COMMAND status bit which is delivered in the next response token.

Table 30 Illegal commands

7.1.2.3 Message Length

a. The number of payload data words that shall be transmitted in the current message.

b. The number shall not include the command token and the CRC checksum appended at the end of the message.

7.1.2.4 Sub-address

a. This field shall provide sub-address location for write and read commands.

> Note 1 This field is also used for other purposes which are explained in clause 7.1.2 (Message Header - Command Token)

7.1.2.5 Command CRC

Command Token Word #1				Command Token Word #2			
Prefix	Command code	Spare	Message length	Prefix	Sub-address	Spare	CRC-4

Bits protected by CRC-4

Table 31 Command CRC

a. The final four bits of the command token shall consist of checksum for all the previous command token bits transmitted in this message.

b. The CRC-4 shall be computed for the 28 bits of command token,

   i. All 16 bits of Command word #1

   ii. Excluding CRC-4 field of Command word #2

c. The Prefix and Spare fields shall be included in the CRC calculation.

d. The generator polynomial used shall be $X^4 + X + 1$.

e. In the slave the CRC-4 shall be calculated for the received command token, if the CRC-4 calculated in the receiver does not match the expected value (this field) the corresponding command token shall be discarded in the slave.

> Note 1 When the command token is discarded because of incorrect CRC-4 value message error status shall be enabled and transmitted to master as part of the next response token.

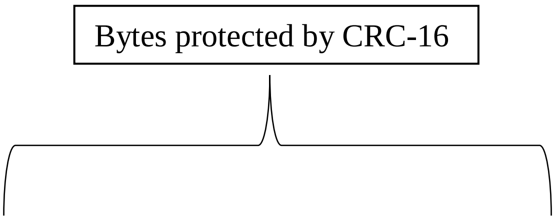
7.1.2.6 Payload data

a. The payload consists of the data need to be transferred from the master to slave.

b. Depending on the command executed the data transferred through the MOSI signal shall include valid data or string of zeros.

Note 1 SPI requires a word transmitted to receive a word. For a write data command the MOSI signal shall have the data to be written as the payload but the read command shall have data in the form of string of zeros.

7.1.2.7 Payload CRC



Bytes protected by CRC-16

Signal	Message Header		Payload Data			Payload CRC
MOSI	Write Command Token #1	Write Command Token #2	Data1	Data2	Data3	CRC-16
MISO	Response #1	Response #2	0x0000	0x0000	0x0000	0x0000

Table 32 Payload CRC

a. When valid information is delivered in the payload data section of the message a Payload CRC (CRC-16) shall be appended at the end of the message.

b. The generator polynomial used shall be $x^{16}+ x^{15}+ x^{2}+ 1$.

Note 1 When information in the form of string of zeros is included then no Payload CRC is required, the payload CRC field shall be all zero (0x0000).

c. In the slave the CRC-16 shall be calculated internally for the received payload data, if the calculated CRC-16 does not match the expected value (this field) the message error status shall be enabled and transmitted to master as part of the next response token.

7.1.3 Message Header -Response Token

7.1.3.1 Overview

The slave transmits a message header which consist of status of module and details of error occurred. The message header sent by slave device shall be called response token which consist of two 16-bit words.

The message header content details are explained below.

MSB	Response Token Word #1														LSB
Prefix		Status					Spare					Module State			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
'1'	'0'	STF	ME	AR	IC	‘0’	'0'	'0'	'0'	'0'	'0'	MS3	MS2	MS1	MS0

Table 33 Response word #1

MSB	Response Token Word #2														LSB
Prefix		Data	Spare									CRC-4			
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
'1'	'0'	‘0’	'0'	'0'	'1'	'1'	'1 '	'1'	'0'	'0'	'0'	CRC3	CRC2	CRC1	CRC0

Table 34 Response word #2

7.1.3.2 Prefix and spare

a. The prefix bits shall be transmitted initially.

b. In the response word#1 and #2 the prefix and spare bits shall have fixed value.

7.1.3.3 Status bits

7.1.3.3.1 SPI terminal fault condition

Bit	Identifier	Type	Value	Description	Clear Condition
13	SPI_TERMINAL_FAULT	Error	‘0’=no fault ‘1’=fault	The bit shall flag a SPI terminal fault condition.	According to the module current state.

Table 35 SPI terminal fault condition

7.1.3.3.2 Message error

Bit	Identifier	Type	Value	Description	Clear Condition
12	MESSAGE_ERROR	Error	‘0’=no fault ‘1’=fault	This bit shall be utilized to indicate that the previous message received from the master has failed to pass	Always related to the previous command. Reception of a valid command shall clear

Bit	Identifier	Type	Value	Description	Clear Condition
				the validity tests.	the message error.

Table 36 Message error

a. This status bit shall be enabled when the received message fails to pass the command token and payload data CRC checks.

b. The next valid command shall clear this status bit.

7.1.3.3.3 Address error

Bit	Identifier	Type	Value	Description	Clear Condition
11	ADDRESS_ERROR	Error	‘0’=no fault ‘1’=fault	This bit shall flag that an unaligned sub address, which did not match the used block in this module, was used in the previous command.	Always related to the previous command. Reception of a valid command shall clear the message error.

Table 37 Address error

Note 1 For example, the slave device can use this status bit to intimate the master about AMBA error response triggered for write or read from a specific address space.

7.1.3.3.4 Illegal error

Bit	Identifier	Type	Value	Description	Clear Condition
10	ILLEGAL_COMMAND	Error	‘0’=no fault ‘1’=fault	This bit shall flag that the previous received command was not compatible with the state of the module at the reception’s instant.	Always related to the previous command. Reception of a valid command shall clear the message error.

Table 38 Illegal error

a. This status bit shall be enabled for the following cases

   i. The prefix and spare bits in the received command token do not match the intended value.

   ii. An unimplemented command is received.

   iii. The next valid command clears this status bit.

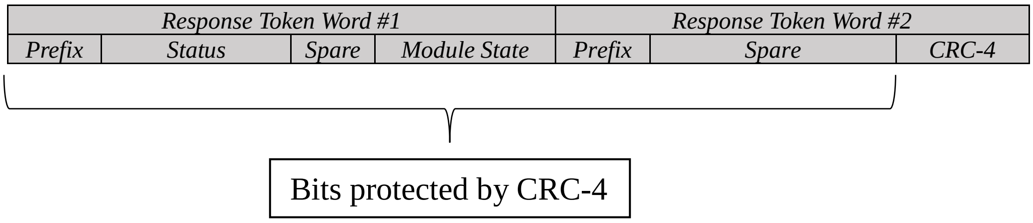
7.1.3.4 Module state

Bit	Identifier	Type	Value	Description	Clear Condition
0 to 3	Module State	Status	Implementation Specific	Optional bits used to express module state/condition. Details left to implementers.	Implementation Specific

Table 39 Explanation for module state.

Note 1 These bits can be used by the slave device to provide additional status to the master.

7.1.3.5 Response CRC



Response Token Word #1				*Response Token Word #2*		
Prefix	*Status*	*Spare*	*Module State*	*Prefix*	*Spare*	*CRC-4*

Table 40 Response CRC

a. The final four bits of the response token shall consist of checksum for all the previous response token bits transmitted in this message.

b. The CRC-4 shall be computed for the 28 bits of command token,

   i. All 16 bits of Response word #1

   ii. Excluding CRC-4 field of Response word #2

c. The Prefix and Spare fields shall be included in the CRC calculation.

d. The generator polynomial used shall be $X^4 + X + 1$.

e. In the master the CRC-4 shall be calculated for the received response token, if the CRC-4 calculated in the receiver does not match the expected value (this field) the corresponding response token shall be discarded in the master.

7.1.3.6 Payload data

a. The payload consists of the data need to be transferred from the slave to the master.

b. Depending on the command executed the data transferred through the MISO signal shall include valid data or string of zeros.

Note 1 SPI requires a word transmitted to receive a word. For a read data command, the MISO signal shall have the data to be written as the payload to the master but the write command shall have data in the form of string of zeros.

7.1.3.7 Payload CRC

Signal	Message Header		Payload Data			Payload CRC
MOSI	Read Command Token #1	Read Command Token #2	0x0000	0x0000	0x0000	0x0000
MISO	Previous Command Response #1	Previous Command Response #2	Data1	Data2	Data3	CRC-16

Bytes protected by CRC-16

Table 41 Payload CRC

a. When valid information is delivered in the payload data section of the message a Payload CRC (CRC-16) shall be appended at the end of the message.

b. The generator polynomial used shall be $x^{16}+ x^{15}+ x^{2}+ 1$.

> Note 1 When information in the form of string of zeros is included then no Payload CRC is required, the payload CRC field shall be all zero (0X0000).

c. In the master the CRC-16 shall be calculated internally for the received payload data, if the calculated CRC-16 does not match the expected value (this field) the payload data shall be discarded.

8 Redundancy

8.1 Introduction

The following solutions are provided to mitigate failures. The redundancy scheme which should be selected shall be based on the actual use of SPI. The intention in the definition below is to keep the system as simple as possible and less complexity shall be added to achieve redundancy.

8.2 Overview

8.2.1 Simple duplication of sensors

As per the implementers need only the interfacing slaves can be duplicated.

8.2.2 Two independent lanes

All the components of the system master, slaves and bus structures shall be duplicated. This option tolerant to any single failures.

8.2.3 Cross-strapped slaves

A possible redundancy scheme with dual port slaves is provided in Figure 6. The nominal and redundant masters are connected to each slave device by two serial interfaces. The slave device must have internal logic to select one of the redundant ports based on the input provided by master device or using configuration from an intelligent device. If the inputs are provided by the master, the commands shall be similar to the redundancy commands described in clause 7.1.2.2.9.

8.2.4 Failure detection and recovery using SPI-2 protocol commands

Initially both the SPI port interfaces are enabled to receive commands, when the communication between the nominal master and slave interface fails then the redundant master can deactivate the nominal interface using its dedicated redundant interface.

An example switchover scenario from nominal to redundant interface is described in the following text.

- The nominal master communicates with its dedicated interface to a slave device, a fault occurred can be detected by the master using several options,
- The status received by the master have invalid values (using the response token),
- The Read back command sent does not provide appropriate values in the received payload,

- Error bits are enabled in the status received by the master (using the response token),

Based on any of the fault detection methods above, the master can send deactivate command in the redundant interface to deactivate the nominal interface of the slave. The master can send Read back command (using redundant port) to check if the previous deactivate command was received by the slave, and it can check the status of the response token as well. After conforming a proper communication has been established the master can use the redundant interface to perform its normal operations.

Requirements for this case is defined in this standard in clause 8.3.

8.3 Redundant system

See example in Figure 6 below.

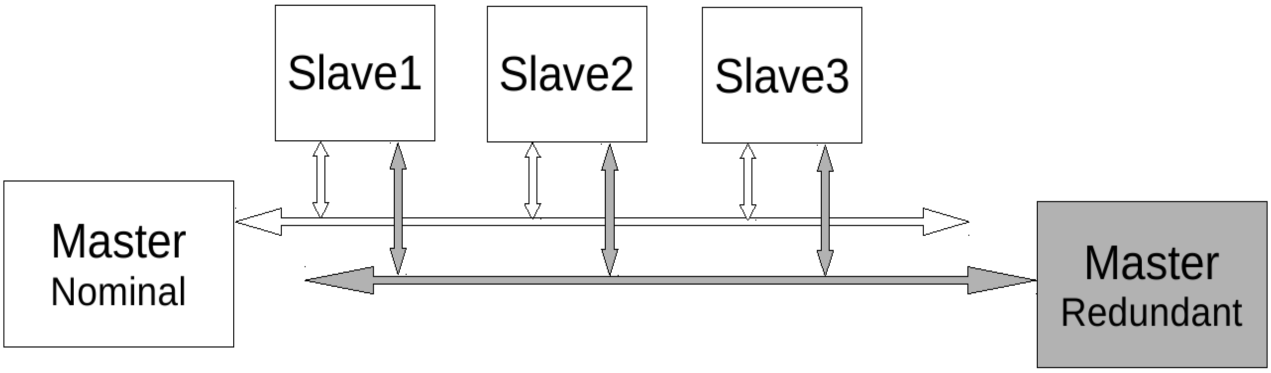


Figure 6 Redundant system

a. The nominal and redundant master shall connect to each slave device by two serial interfaces.

b. The slaves shall have two serial interfaces.

c. The slave device port selection shall be performed by external configuration or using the command messages from master.

Note 1 Dedicated command codes are defined for SPI-2 protocol.

d. Using external configuration, only one port shall be active at any time.

Note 2 The external configuration takes complete control of the device port selection. The switch over shall be signaled by the external configuration to the slave device.

Note 3 One master using its dedicated port shall perform communication with the slave at any time.

e. Using command messages for port selection, the slave shall have the possibility to receive switch over commands from both the interfaces.

Note 4 The normal operations (other than switch over commands) are carried out by one master using its dedicated port, to take control of the device the redundant master shall send deactivate command (to disable the nominal interface) before performing its normal operations.

f. The commands delivered from a port shall not activate or deactivate itself.

Note 5 The commands shall only activate and deactivate the other redundant port.

9 Bibliography

[RD1] Open Systems Interconnection, ISO/IEC 7498-1
[RD2] Time Code Formats, CCSDS 301.0-B-4

Annex A (informative)
Terminations in SPI Networks

A.1 Introduction

SPI networks can either be implemented using differential signalling or, more commonly, single ended signalling. These two different implementation schemes place different requirements on the implementation of transmission line terminations. While the differential network is commonly terminated in both driver and receiver ends, the single ended network is only terminated in the driver end. Differential SPI networks can be utilized both for point-to-point communications and in multi-drop configuration, however point-to-point is considered the default configuration. Single ended SPI networks are commonly used in one of two configurations, either a large multi-drop bus using a low bit-rate (≤1 Mbit/s) or a short point-to-point connection working at a high (≥1 Mbit/s) bit-rate.

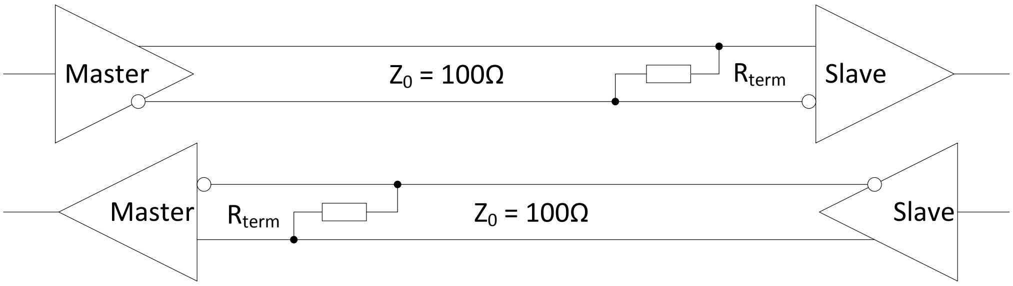


Figure 7 Differential point-to-point SPI network (simplified)

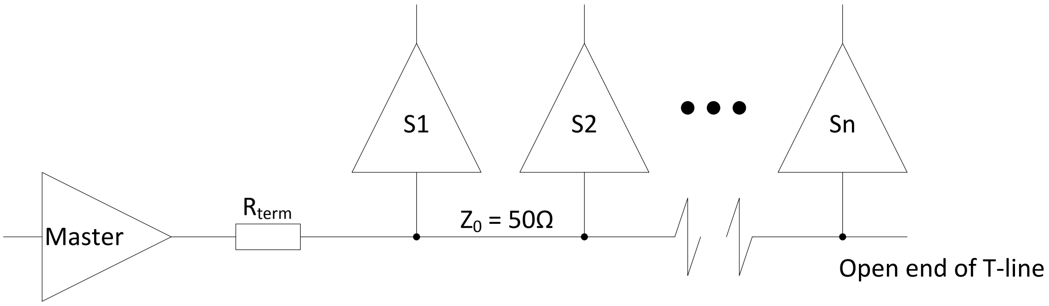


Figure 8 Example multi-drop single ended SPI network (simplified)

A.2 Differential point-to-point networks

Differential point-to-point networks are typically impedance matched in both ends. This is accomplished through the use of matched drivers, controlled impedance transmission lines and receiver end termination. Selection of terminations in differential point-to-point networks is normally straight forward as components used are intended for use in impedance matched networks.

a. Drivers used in differential should have a known output impedance of 100±10 Ω:

  1. In the case of an integrated driver this is normally specified by the manufacturer.

     NOTE Driver end termination might be necessary for matched integrated drivers to handle down-stream short circuits.

  2. For a discrete driver circuit it is necessary to ensure that compliant signal levels are achieved even when output termination is added.

b. Transmission lines or cables used in differential point-to-point SPI networks should have a differential impedance of 100±10 Ω and be of twisted pair type if implemented as a cable:

  1. For cables, the use of external shields is dependent on spacecraft EMC requirements.

  2. Due to comparatively low data rates, no shields between pairs is necessary.

c. Receivers should be fitted with parallel termination resistors matching the impedance of the transmission media (PCB or cable):

  1. The receiver input impedance should be high (≥100 kΩ) to avoid interaction between any discrete termination and the receiver input stage.

     NOTE Termination matching the transmission media avoids reflections which negatively impact signal integrity.

A.3 Single ended point-to-point networks

Single ended point-to-point networks are normally utilized for very short (≤5 cm) connections between a single Master and Slave device. These networks typically do not require impedance control due to the short electrical length of the traces.

NOTE Series resistors might be required in single ended point-to-point networks to handle receiver short circuit conditions.

A.4 Single ended multi-drop networks

Single ended multi-drop networks are normally utilized as either a daisy chain of SPI slaves on a single PCB or in a backplane configuration connecting one master to many SPI slave PCBs. While conceptually different, both these network implementations place identical requirements on the termination in driver and receiver ends.

a. Termination in single ended multi-drop networks should only be done at the driver end of the line to avoid loss of signal amplitude.

  1. Drivers used in single ended multi-drop networks are not commonly specified for driving a controlled impedance and must therefore be

resistively series terminated. Adding any receiver end termination would result in voltage division and loss of signal amplitude.

b. The termination resistance used should be selected to as accurately as possible match the driver to the SPI network.

1. This ensures a 50% amplitude incident wave giving full swing at the first full reflection.

2. Selection of termination resistance should take into account the input impedance of the SPI-network, the apparent output impedance of the driver, and handling of fault conditions.

   NOTE Apparent driver output impedance is not commonly specified for single ended chips. Determination of apparent output impedance can be done through the use of simulation models or direct measurements.

c. Transmission lines used in single ended multi-drop networks should have as high impedance as is practical considering the technology used.

   NOTE A high impedance transmission line is easier to drive but comes at the expense of very narrow trace widths when implemented on a PCB. Additionally, high impedance transmission lines can allow for better matching as the minimum value of the termination resistor increases when taking the discussion in b. into account.

d. No termination should be fitted at receivers in the network.

1. Leaving the trace “open” ensures full reflection at the far end of the network, this is the quickest possible rise time of the signal without overdriving the transmission line.

2. In practical circuits a pull-up or pull-down resistor might be necessary in the receiver to ensure deterministic behaviour if the driver is lost.

   NOTE The choice of pull-up or pull-down depends on configuration of CPOL and chip-select polarity.

e. Normally the Master should be placed in one end of a multi-drop SPI network.

1. Placement of the Master in the middle of a network halves the effective impedance of the network as seen by the Master, this makes the transmission line more difficult to drive, degrading signal quality.

Annex B (informative)
SPI Bus Topologies

B.1 Introduction

SPI bus topology is in this Annex defined as the way the different nodes in a SPI network are connected to each other.

The nodes in a SPI network can either be a master or a slave. Only one master can be part of a bus, while several slaves can be connected in the network. The master initiates the transfer of data between the master and the slaves.

Two different SPI bus topologies are possible:

a. A topology with independent slaves, where all the slaves are directly connected to the master and all slaves share the same SCLK, MOSI and MISO signals. The CS signal though is unique for each slave. Only one slave can be addressed at the time, see Figure 9. An advantage with this topology is that the data delay to a specific slave is low.

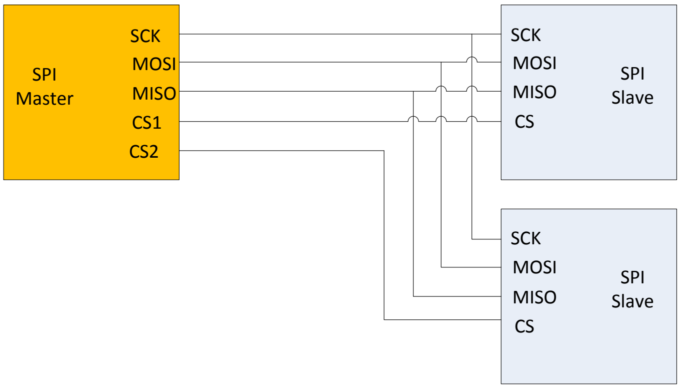


Figure 9 Schematic of a SPI topology with independent slaves.

b. A topology with cooperative slaves, where all the slaves are directly connected to the master with the SCK and CS signals that are in common for all devices while the MISO and MOSI signals are connected to one slave respectively, see Figure 10.
NOTE The MISO and MOSI signals are daisy-chained between the slaves and could be regarded as a large shift register with e.g. 8 bits in each slave.
An advantage with this topology is that the CS is in common for all slaves. This is especially favorable in a network with many slaves, where the number of chip selects output from the master is limited. The data delay is dependent on the number of slaves in the network.

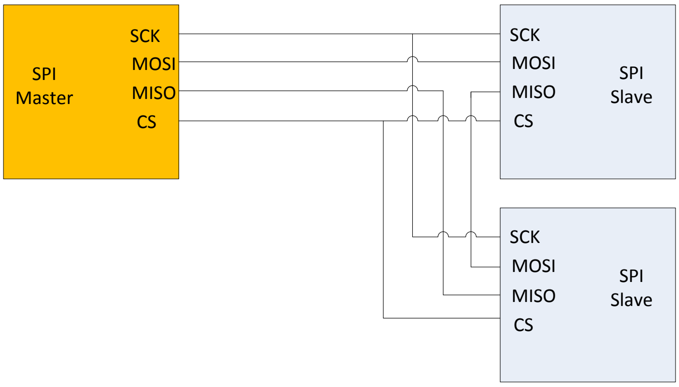


Figure 10 Schematic of a SPI topology with cooperative (daisy chained) slaves.

The topology with cooperative slaves are not covered further in this Annex.

B.2 Bus topology with independent slaves

In order to maximize clock speed, considerations when implementing the topology should be made to minimize driver load, reflections and flight time in traces. In practice some general rule of thumbs should be considered:

a. The number of slaves in the bus should be kept as few as possible.

NOTE This is especially true when the signals to the different slaves are routed in separate traces that branch out close to the master.

b. The traces should be kept as short as possible.

NOTE The flight time for a signal is directly proportional to the length of the traces.

c. A signal common for all slaves should to as large extent as possible be routed in a common trace with short stubs close to the individual slaves.

NOTE This decreases the load on the driver (less fan-out) and minimizes the reflection time from the different branches, compare the left illustration in Figure 11, where mostly separate traces are used for the common signals to the different slaves, with the right illustration, where to a large extent common signals are used.

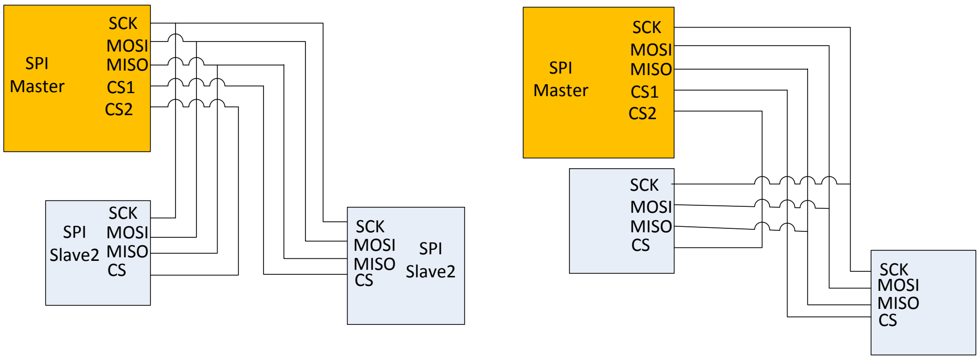


Figure 11 Schematics illustrating different routing alternatives for independent slaves. To the left, the common signals are routed in separated traces from close to the master. To the right, the common signals are to a large extent routed in a common trace, with small stubs at the slave.

d. In order to get as high clock frequency as possible with consideration to the topology, the special case with one master and one slave (point-to-point) routed at a close distance is optimal.

e. If the required clock frequency cannot be achieved in a network with a specific number of slaves, the option to divide the network into smaller networks should be considered.