HE910 SPI Port Application Note

80000NT10053A Rev. 1 – Preliminary – 2013-02- 04

Making machines talk.

APPLICABILITY TABLE

HE910 Family	SW Version
HE910 [1]	12.00.xx3
HE910-GA	
HE910-D	
HE910-EUR/ HE910-EUD	
HE910-EUG/ HE910-NAG	
HE910-NAR/ HE910-NAD	

[1] HE910 is the “type name” of the products marketed as HE910-G & HE910-DG.

SPECIFICATIONS SUBJECT TO CHANGE WITHOUT NOTICE

Notice

While reasonable efforts have been made to assure the accuracy of this document, Telit assumes no liability resulting from any inaccuracies or omissions in this document, or from use of the information obtained herein. The information in this document has been carefully checked and is believed to be entirely reliable. However, no responsibility is assumed for inaccuracies or omissions. Telit reserves the right to make changes to any products described herein and reserves the right to revise this document and to make changes from time to time in content hereof with no obligation to notify any person of revisions or changes. Telit does not assume any liability arising out of the application or use of any product, software, or circuit described herein; neither does it convey license under its patent rights or the rights of others.

It is possible that this publication may contain references to, or information about Telit products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that Telit intends to announce such Telit products, programming, or services in your country.

Copyrights

This instruction manual and the Telit products described in this instruction manual may be, include or describe copyrighted Telit material, such as computer programs stored in semiconductor memories or other media. Laws in the Italy and other countries preserve for Telit and its licensors certain exclusive rights for copyrighted material, including the exclusive right to copy, reproduce in any form, distribute and make derivative works of the copyrighted material. Accordingly, any copyrighted material of Telit and its licensors contained herein or in the Telit products described in this instruction manual may not be copied, reproduced, distributed, merged or modified in any manner without the express written permission of Telit. Furthermore, the purchase of Telit products shall not be deemed to grant either directly or by implication, estoppel, or otherwise, any license under the copyrights, patents or patent applications of Telit, as arises by operation of law in the sale of a product.

Computer Software Copyrights

The Telit and 3rd Party supplied Software (SW) products described in this instruction manual may include copyrighted Telit and other 3rd Party supplied computer programs stored in semiconductor memories or other media. Laws in the Italy and other countries preserve for Telit and other 3rd Party supplied SW certain exclusive rights for copyrighted computer programs, including the exclusive right to copy or reproduce in any form the copyrighted computer program. Accordingly, any copyrighted Telit or other 3rd Party supplied SW computer programs contained in the Telit products described in this instruction manual may not be copied (reverse engineered) or reproduced in any manner without the express written permission of Telit or the 3rd Party SW supplier. Furthermore, the purchase of Telit products shall not be deemed to grant either directly or by implication, estoppel, or otherwise, any license under the copyrights, patents or patent applications of Telit or other 3rd Party supplied SW, except for the normal non-exclusive, royalty free license to use that arises by operation of law in the sale of a product.

Usage and Disclosure Restrictions

License Agreements

The software described in this document is the property of Telit and its licensors. It is furnished by express license agreement only and may be used only in accordance with the terms of such an agreement.

Copyrighted Materials

Software and documentation are copyrighted materials. Making unauthorized copies is prohibited by law. No part of the software or documentation may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language or computer language, in any form or by any means, without prior written permission of Telit

High Risk Materials

Components, units, or third-party products used in the product described herein are NOT fault-tolerant and are NOT designed, manufactured, or intended for use as on-line control equipment in the following hazardous environments requiring fail-safe controls: the operation of Nuclear Facilities, Aircraft Navigation or Aircraft Communication Systems, Air Traffic Control, Life Support, or Weapons Systems (High Risk Activities"). Telit and its supplier(s) specifically disclaim any expressed or implied warranty of fitness for such High Risk Activities.

Trademarks

TELIT and the Stylized T Logo are registered in Trademark Office. All other product or service names are the property of their respective owners.

Contents

Figures

Tables

1. Introduction

1.1. Scope

The present Application Note provides the reader with a guideline concerning the use of the SPI port supported by the modules belonging to the HE910 family.

1.2. Audience

This document is intended for those users that need to develop a synchronous connection between an Application Processor and an HE910 module using the SPI port.

1.3. Contact Information, Support

For general contact, technical support, to report documentation errors and to order manuals, contact Telit's Technical Support Center (TTSC) at:

TS-EMEA@telit.com
TS-NORTHAMERICA@telit.com
TS-LATINAMERICA@telit.com
TS-APAC@telit.com

Alternatively, use:

http://www.telit.com/en/products/technical-support-center/contact.php

For detailed information about where you can buy the Telit modules or for recommendations on accessories and components visit:

http://www.telit.com

To register for product news and announcements or for product questions contact Telit's Technical Support Center (TTSC).

Our aim is to make this guide as helpful as possible. Keep us informed of your comments and suggestions for improvements.

Telit appreciates feedback from the users of our information.

1.4. Text Conventions

Danger – This information MUST be followed or catastrophic equipment failure or bodily injury may occur.

Caution or Warning – Alerts the user to important points about integrating the module, if these points are not followed, the module and end user equipment may fail or malfunction.

Tip or Information – Provides advice and suggestions that may be useful when integrating the module.

1.5. Related Documents

[1] HE910 AT Commands Reference Guide, 80378ST10091A

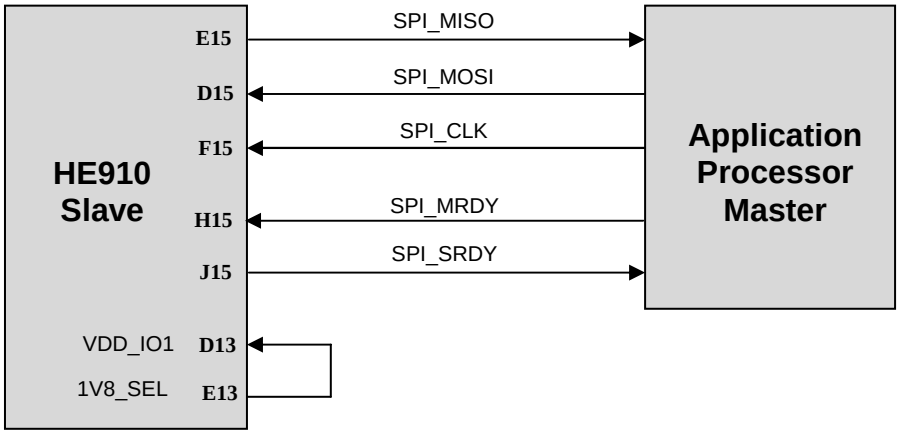
[2] HE910 Hardware User Guide, 1vv03700925

[3] Telit EVK2 User Guide, 1vv0300704

[4] HE910 Family Ports Arrangements, User Guide, 1vv0300971

2. SPI Interface Overview

The HE910 Module provides an SPI interface that shares the hardware resources with the AUX_UART port, to have more detailed information refer to [2]. It shall be noted that the SPI port of the module differs slightly from the standard SPI. The SPI module interface supports two handshake lines for flow control and mutual wake-up: SRDY (slave ready) and MRDY (master ready). The user Application Processor (AP) has the master role so it supplies the clock signal (up to 13MHz). The figure below shows the involved signals and the table summarizes the hardware characteristics of the signals.



PAD	Signal	I/O	Function	Type	Comment
D15	SPI_MOSI	I	SPI MOSI (Data is transferred from Master to Slave)	CMOS 1.8V	SHARED with AUX TX
E15	SPI_MISO	O	SPI MISO (Data is transferred from Slave to Master)	CMOS 1.8V	SHARED with AUX RX
F15	SPI_CLK	I	SPI Clock	CMOS 1.8V	
H15	SPI_MRDY	I	SPI_MRDY (Master is active and ready to transfer data. Similar to Select Slave (SS) on standard SPI)	CMOS 1.8V	
J15	SPI_SRDY	O	SPI_SRDY (Slave is active and ready to transfer data)	CMOS 1.8V	

PAD	Signal	I/O	Function	Type	Comment
D13	VDD_IO1	I	IO1 SUPPLY Input		This pin must always connected to E13
E13	1V8_SEL	O	1V8 SEL for VDD_IO1		

2.1. SPI Setting

The Fig. 1 shows the SPI Test Bench used to describe how the module SPI interface works. Two aspects are taken in consideration: the signals timing of the involved lines and the protocol format used to exchange the information between the two devices.

The SPI Master represents the generic user Application Processor and in this example it is built up by:

- a Window-PC running the Telit's AT Controller (it is available on the Telit's Site) that allows to connect the PC to the module via different hardware interfaces (COMx, USB, SPI);
- a Telit's USB/SPI Adapter that translates the USB interface into an SPI interface. The USB/SPI Adapter is installed on the interface board of the module. In turn, the interface board is installed on the EVK2 board. To have detailed information about the interface board and the EVK2 board refer to [3]. Telit's USB/SPI Adapter is used only internally to Telit's LAB to simulate the generic user Application Processor.

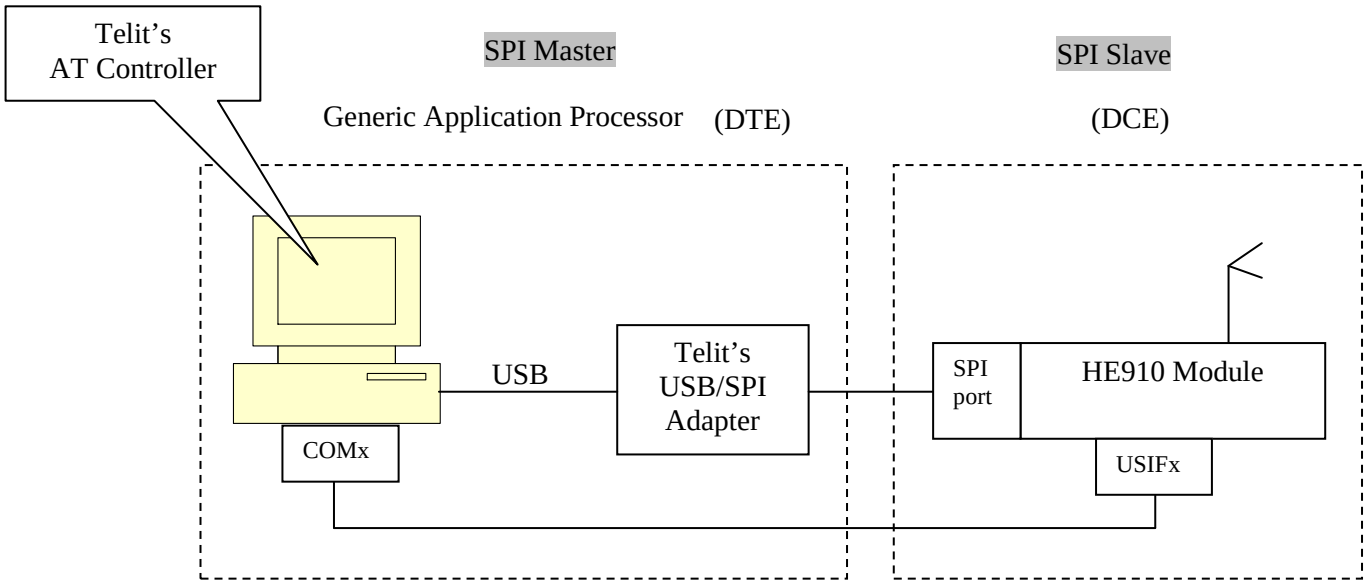


Fig. 1: Telit's SPI Test Bench

Before using the SPI port the HE910 module must be configured as described hereafter.

- It is assumed that the Telit's AT Controller is configured to use the COMx.
- Via COMx the AT terminal of the Telit's AT Controller sends the AT#PORTCFG=2 command to configure the SPI port of the module. To have detailed information about this step refer to [4].
- To make active the new SPI configuration turn off/on the module.
- Configure the Telit's AT Controller in order to use the SPI port. The COMx/USIFx connection can be disconnected.

Now, the Application Processor (SPI Master) can send on its initiative to the module (SPI Slave) data (e.g. AT commands) via the SPI interface, or can receive data (e.g. URC) on initiative of the module.

2.1.1. SPI Mode

It is worth noting that the standard SPI interface provides 4 modes of clock phase (CPH) and clock polarity (CPOL).

For CPOL = 0, the inactive level of CLK is 0, in this case:

- When CPHA = 0, data will be read in on the rising edge of CLK, and data will be clocked out on the falling edge of CLK.
- When CPHA = 1, data will be read in on the falling edge of CLK, and data will be clocked out on the rising edge of CLK.

For CPOL = 1, the inactive level of CLK is 1, in this case:

- When CPHA = 0, data will be read in on the falling edge of CLK, and data will be clocked out on the rising edge of CLK.
- When CPHA = 1, data will be read in on the rising edge of CLK, and data will be clocked out on the falling edge of CLK.

The following table summarizes the standard four modes and shows the mode supported by the HE910 modules

Mode	CPOL	CPHA	Supported by HE910
0	0	0	YES
1	0	1	/
2	1	0	/
3	1	1	/

Tab. 1: SPI Mode

2.1.2. RS-232 Interface Software Emulation

A software emulation of the RS-232 interface is realized via the SPI protocol. Control information about the RS-232 status line can be exchanged between the master and slave by setting the relevant flags in the SPI header, for example refer to Fig. 6 and Fig. 8. For reader convenience the figure above shows the V.24 serial interface.

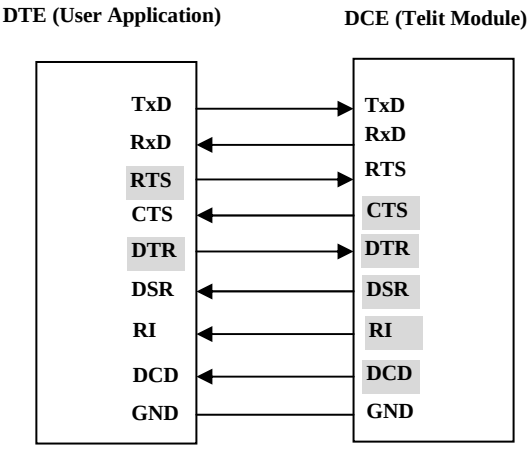


Fig. 2: V.24 Serial Interface

3. HE910 SPI Protocol Description

Before describing the HE910 SPI protocol examples obtained using the Telit's SPI Test Bench shown in Fig. 1, it is worth noting the following SPI line usage assumption:

1. The frame size is known by both sides before the beginning of the packet transfer of each packet. The same amount of data is exchanged in both directions simultaneously.

2. Both sides set their RDY lines independently when they are ready to transfer data. The other side has to wait for the activating interrupt in order to allow the transfer of the other side.

3. The Master starts the clock shortly after MRDY and SRDY are set to active. The amount of clocks is exactly that one of the frame size to be transferred.

4. The SRDY line will be set down after the end of the clock.

5. Usually the MRDY line is also set inactive with the end of the clock, but in case of a big transfer containing multiple packets, the MRDY lines stays active.

The HE910 SPI protocol is described by the examples illustrated on the following two chapters.

3.1. Data Exchange Initiated by the SPI Master

It is assumed that the AP (SPI-Master) needs to send to the module (SPI-Slave) the simple command: AT<CR>. The figure below shows the basic concept.

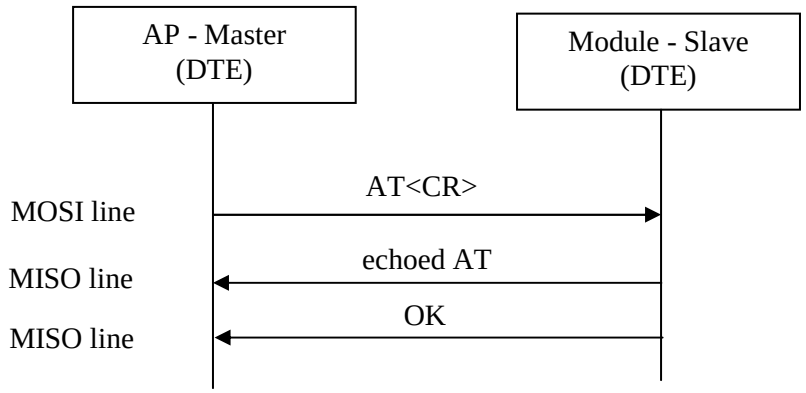


Fig. 3: The Basic SPI Connection: AT<CR>

The basic SPI data flow shown above is hereafter described in detail taking in consideration all the involved data and control SPI signals. Going through the next steps refer to Fig. 4. The figure was not expanded over time in order to show the entire interested timing. To help the reader, on the figure there are the indications of the data type transmitted in both directions, in fact a single vertical line or a narrow impulse can include several bytes building up the SPI messages. The black impulses represent the clock generated by the SPI Master; its representation is compressed over the time.

- The AP sets the MRDY line high to advise the module that it is going to send data. The module is in sleep mode.
- After a while, the module sets the SRDY line high to communicate to the AP that it is ready to receive data.
- The AP recognizes that the module is ready to receive data and in accordance starts the clock on the CLK line and sends the AT<CR> command across the MOSI line. At the same time the module sends “dummy” data on the MISO line (as previously state in step 1 of the chapter 3). When both messages of equal and fixed size are received, the AP stops the clock, and the SRDY and MRDY lines are pulled down respectively by the module and AP.
- The module verifies the correctness of the just received AT command, pulls up the SRDY line, after a while the AP activates the MRDY line and starts the clock. When the module detects on the CLK line the clock signal it sends back, across the MISO line, the AT command (echoed AT). At the same time the AP sends “dummy” data on the MOSI line (as previously state in step 1 of the chapter 3). When both messages of equal and fixed size are received, AP stops the clock, and the SRDY and MRDY lines are pulled down respectively by the module and AP.

- The module executes the AT command and sends to the AP the OK response using the procedure described in the previous step.

On the top of the Fig. 4 are shown some time values relating to the example carried out using the configuration indicated in Fig. 1. Refer to chapter 4.1 to have more timing information concerning MRDY and SRDY lines.

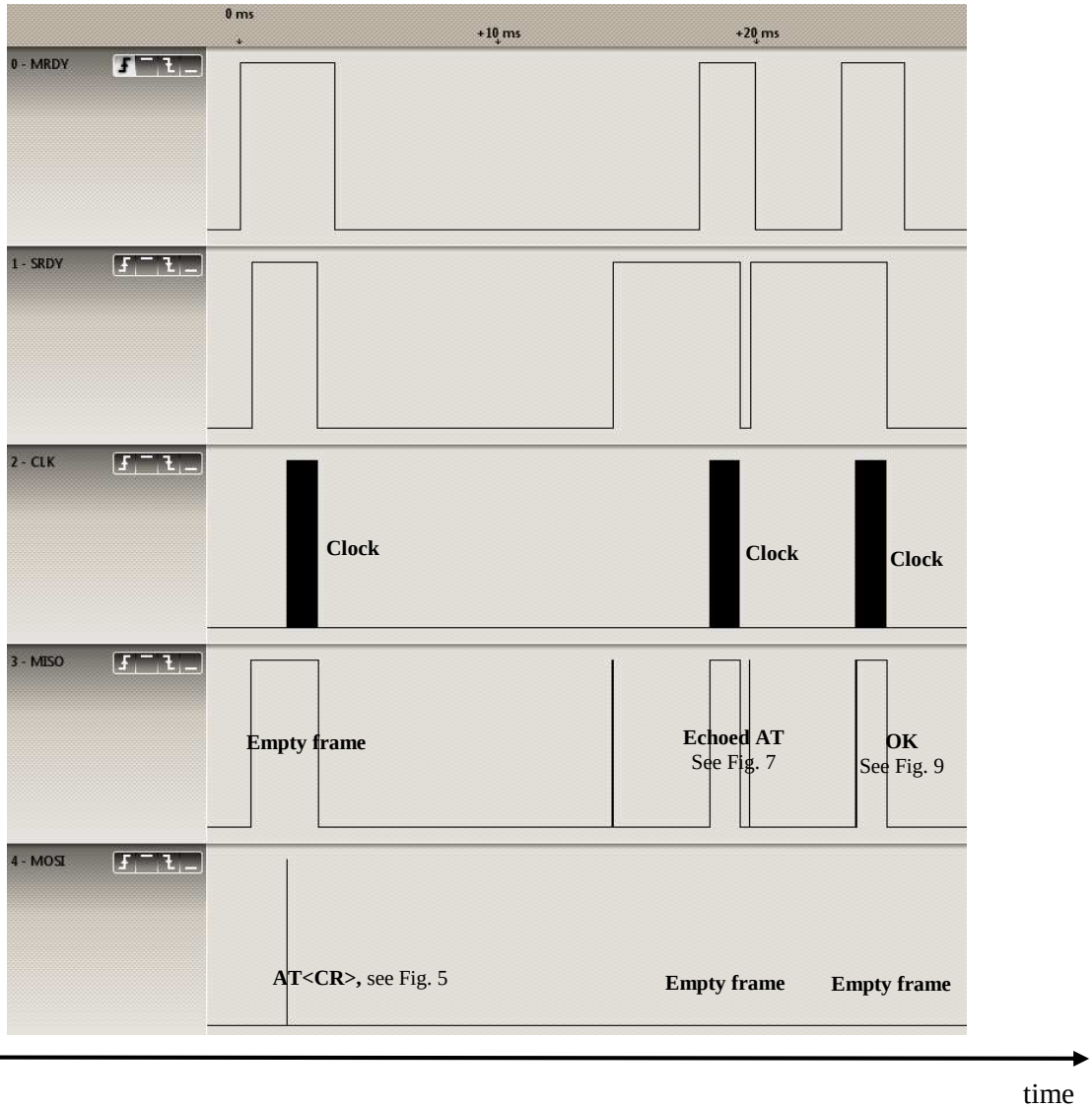


Fig. 4: SPI Timing: AT<CR>

The Fig. 5 shows the signal on MOSI line that forms the header and a part of the payload building up the message AT<CR>. The section of the payload that does not include characters (information) is filled with 0x00.

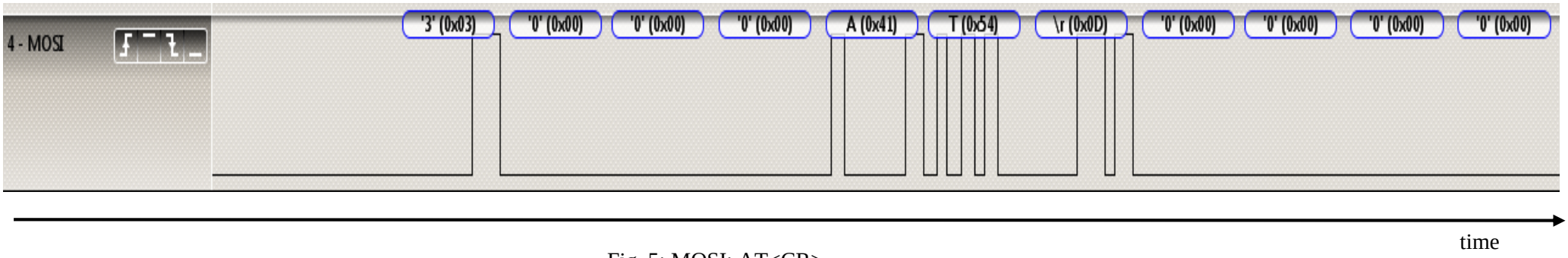
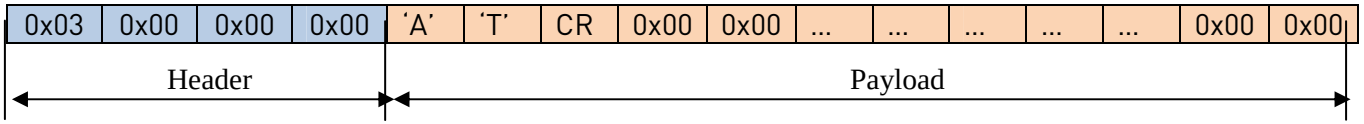


Fig. 5: MOSI: AT<CR>

The SPI Frame format sent by the Master on the MOSI line is the following:



The header size is always long 4 bytes. To explain the header bytes refer to Fig. 6.

The payload-length is set to a fixed value of 2044 bytes. In this example only 3 characters are transported by the payload section, but all 2044 bytes are transmitted.

The Fig. 6 shows the header format and the order in which the bytes are transmitted by the Master on the MOSI line. Depending on Slave device, data is shifted MSB first.

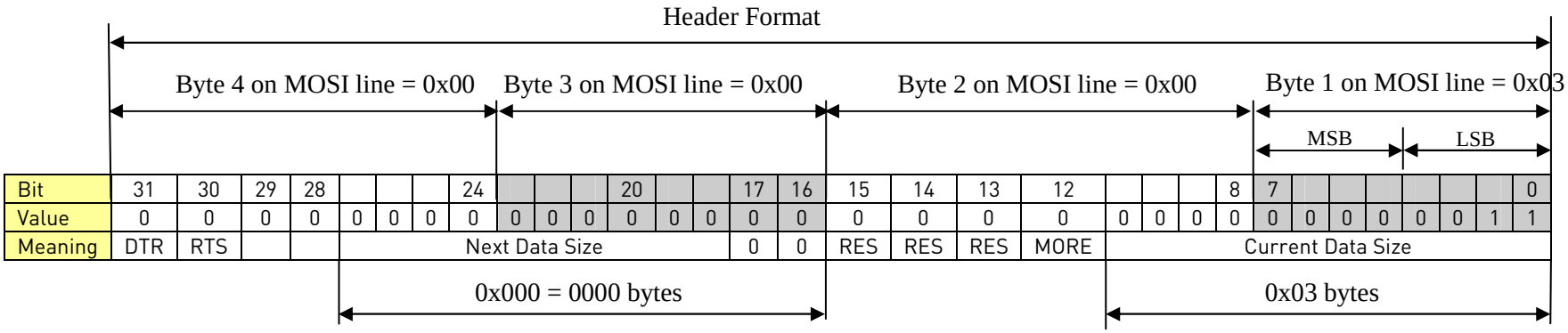


Fig. 6: Header Format

- Current Data Size: characters number inserted into the payload section (regardless the characters number inserted, the payload is always long 2044 bytes)
- MORE: indicates if new data will be sent (MORE = 0 means no more data)
- RES: reserved
- Next Data Size: The Application Processor (Master) sets Next Data Size to 0 bytes.
- RTS: supported
- DTR: supported

The Fig. 7 shows the signal on MISO line that forms the header and a part of the payload building up the echoed message AT<CR>. The section of the payload that does not include characters (information) is filled with 0xFF.

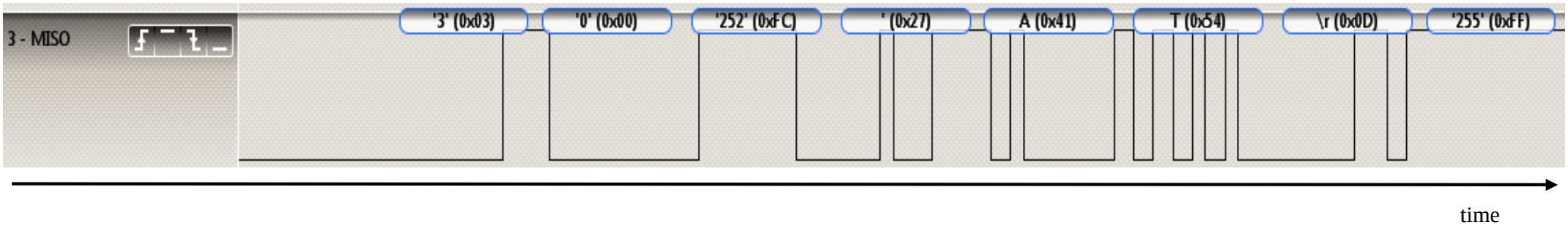
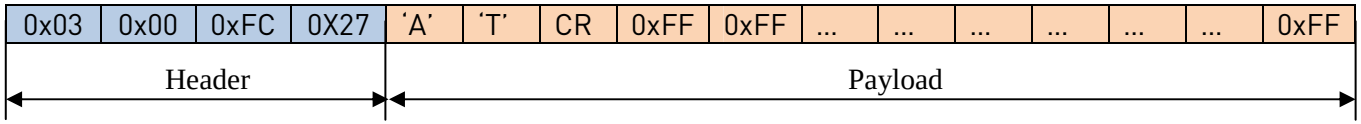


Fig. 7: MISO: Echoed AT<CR>

The SPI Frame format on the MISO line is the following:



The header size is always long 4 bytes. To explain the header bytes refer to Fig. 8.

The payload-length is set to a fixed value of 2044 bytes. In this example only 3 characters are transported by the payload section, but all 2044 bytes are transmitted.

The Fig. 8 shows the header format and the order in which the bytes are transmitted on the MISO line.

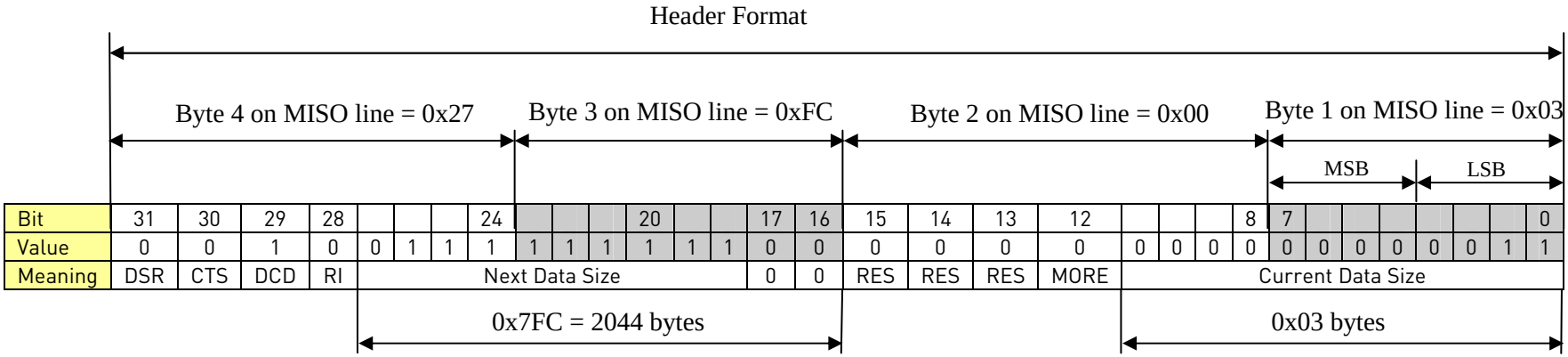


Fig. 8: Header Format

- Current Data Size: characters number inserted into the payload section (regardless the characters number inserted, the payload is always long 2044 bytes)
- MORE: indicates if new data will be sent (MORE = 0 means no more data)
- RES: reserved
- Next Data Size: the module sets Next Data Size to 2044 bytes.
- RI: supported
- DCD: supported
- CTS: supported
- DSR: supported

The Fig. 9 shows the signal on MISO line that forms the header and a part of the payload building up the message OK<CR>. The section of the payload that does not include characters (information) is filled with 0xFF.

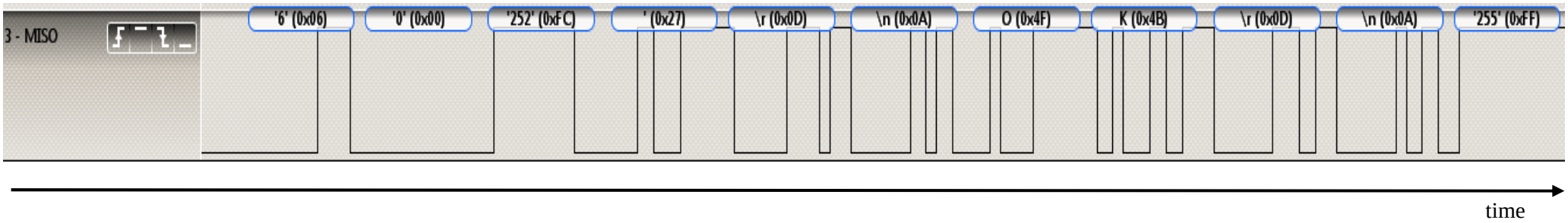
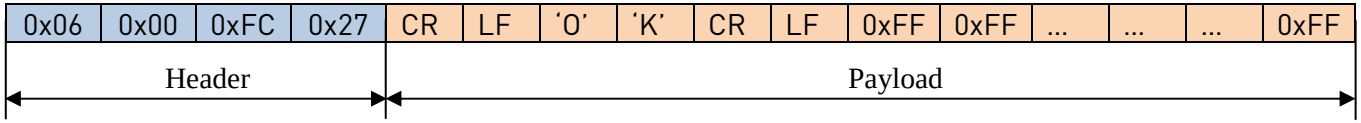


Fig. 9: MISO: OK<CR>

The SPI Frame format on the MISO line is the following:



The header size is always long 4 bytes. To explain the header bytes refer to Fig. 10.

The payload-length is set to a fixed value of 2044 bytes. In this example only 6 characters are transported by the payload section, but all 2044 bytes are transmitted.

The Fig. 10 shows the header format and the order in which the bytes are transmitted on the MISO line.

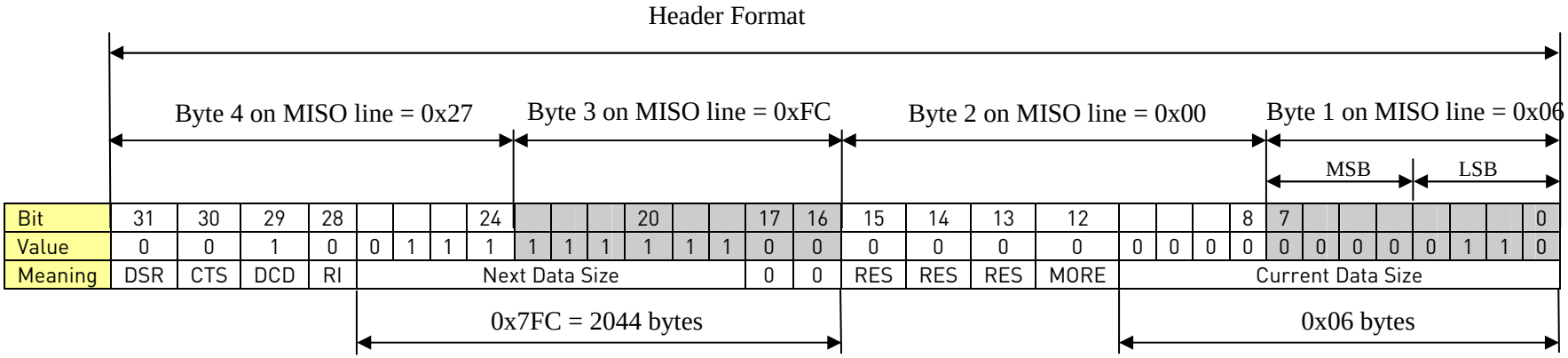


Fig. 10: Header Format

- Current Data Size: characters number inserted into the payload section (regardless the characters number inserted, the payload is always long 2044 bytes)
- MORE: indicates if new data will be sent (MORE = 0 means no more data)
- RES: reserved
- Next Data Size: the module sets Next Data Size to 2044 bytes.
- RI: supported
- DCD: supported
- CTS: supported
- DSR: supported

3.2. Data Exchange Initiated by the SPI Slave

It is assumed that the module (SPI Slave) receives a call. The example illustrates only the sending of the unsolicited RINGs toward the AP, the figure below shows the basic concept.

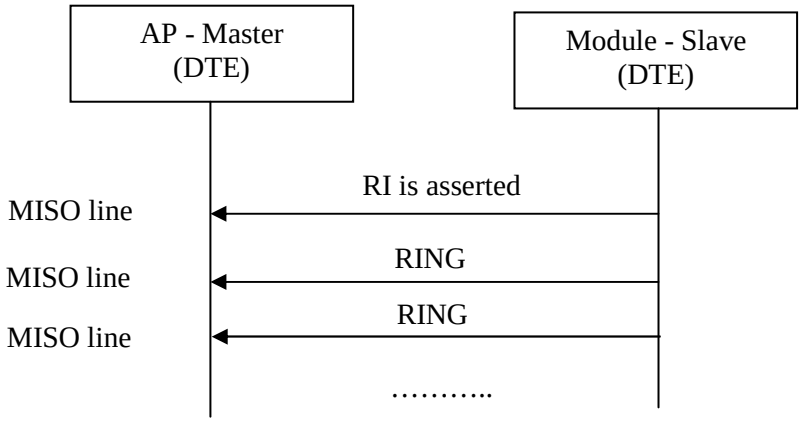


Fig. 11: The Basic SPI Connection: RINGs

The basic SPI data flow shown above is hereafter described in detail taking in consideration all the involved data and control SPI signals. Going through the next steps refer to Fig. 12. The figure was not expanded over time in order to show the entire interval of the interested timing. To help the reader, on the figure there are the indications of the data type transmitted on MISO line, in fact a narrow impulse can include several bytes building up the SPI messages. The black impulses represent the clock generated by the SPI Master; its representation is compressed over the time.

- The module (SPI Slave) sets the SRDY line high to advise the AP that it is going to send data. It is supposed that the AP is in sleep mode.
- After a while, the AP recognizes that the module is ready to exchange data, sets the MRDY line high and sends across the CLK line the clock signal.
- The module detects the clock coming from the AP and starts to send on the MISO line the message (RI asserted). At the same time the AP sends “dummy” message to the module (as stated in step 1 of chapter 3). When both messages of equal and fixed size are received, AP stops the clock, and the SRDY and MRDY lines are pulled down respectively by the module and AP.
- The module to send the next message (RING) starts from the first step previously described.

On the top of the Fig. 12 are shown some time values relating to the example carried out using the configuration indicated in Fig. 1. Refer to chapter 4.1 to have more timing information concerning MRDY and SRDY lines.

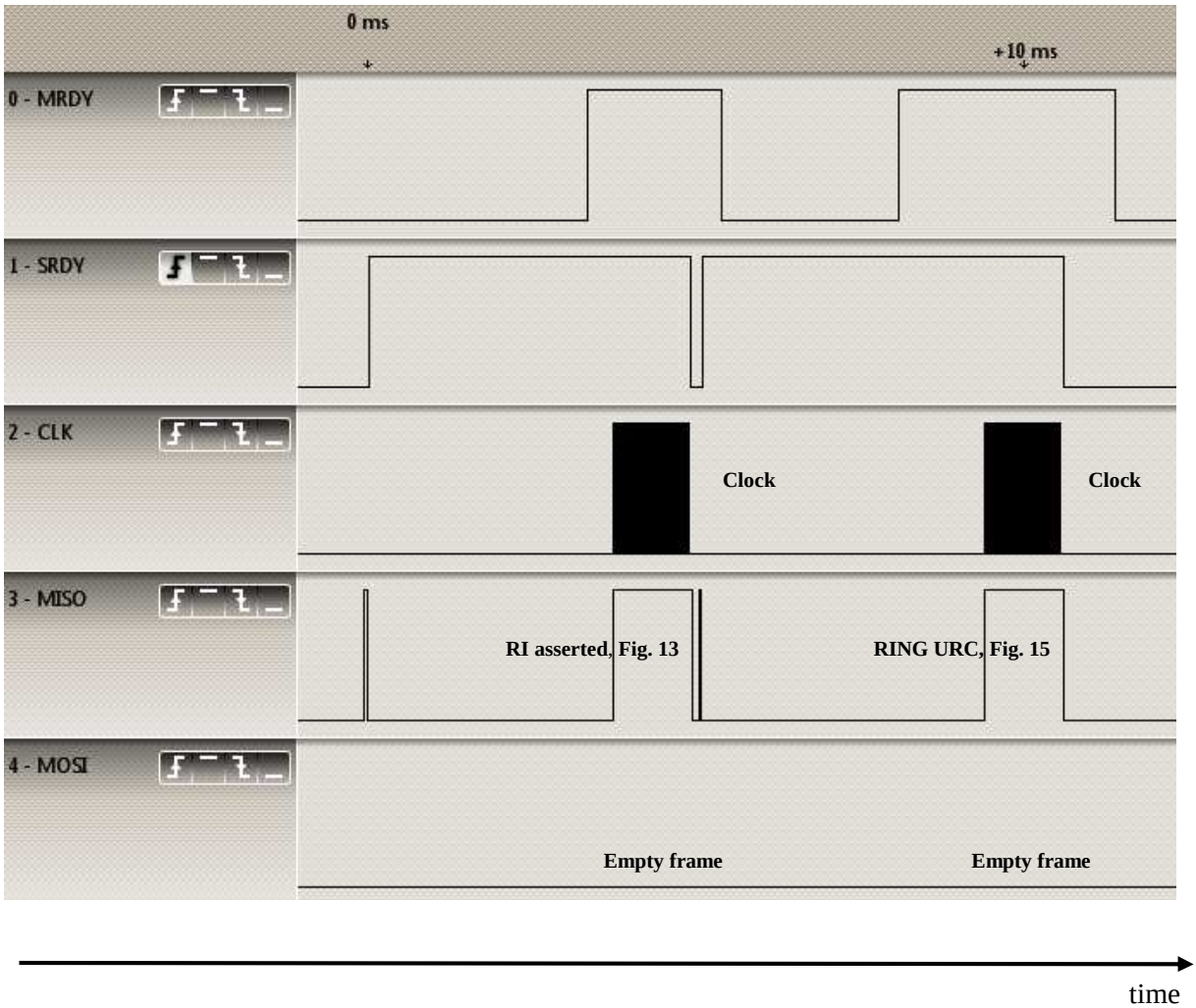


Fig. 12: SPI Timing: RINGs

The Fig. 13 shows the signal on MISO line that forms the header and a part of the payload building up the message RI asserted. The payload does not include characters (information), it is filled with 0xFF.

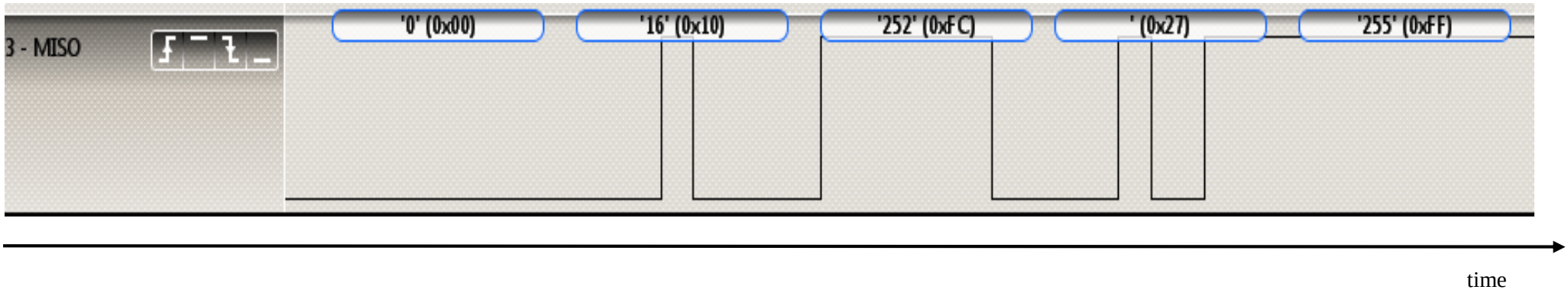
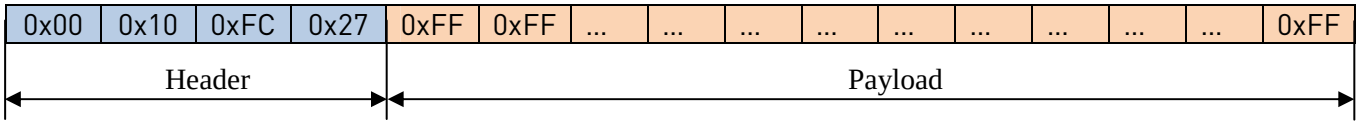


Fig. 13: MISO: RI Asserted

The SPI Frame format sent by the module (Slave) on the MISO line is the following:



The header size is always long 4 bytes. To explain the header bytes refer to Fig. 14.

The payload-length is set to a fixed value of 2044 bytes. In this example no characters are transported by the payload section, but all 2044 bytes are transmitted.

The Fig. 14 shows the header format and the order in which the bytes are transmitted on the MISO line.

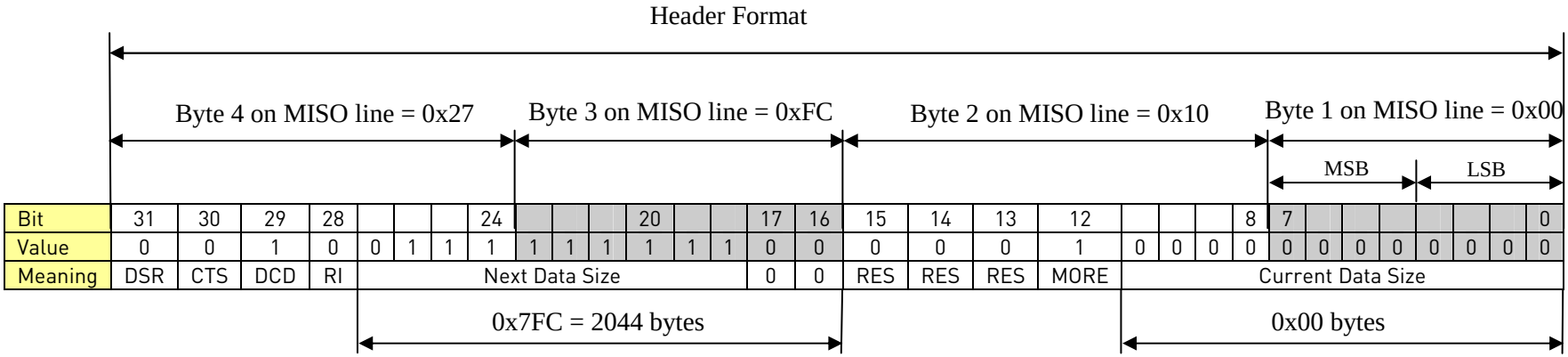


Fig. 14: Header Format

- Current Data Size: characters number inserted into the payload section (regardless the characters number inserted, the payload is always long 2044 bytes)
- MORE: indicates if new data will be sent (MORE = 0 means no more data)
- RES: reserved
- Next Data Size: the module sets Next Data Size to 2044 bytes.
- RI: supported
- DCD: supported
- CTS: supported
- DSR: supported

The Fig. 15 shows the signal on MISO line that forms the header and a part of the payload building up the message RING. The section of the payload that does not include characters (information) is filled with 0xFF.

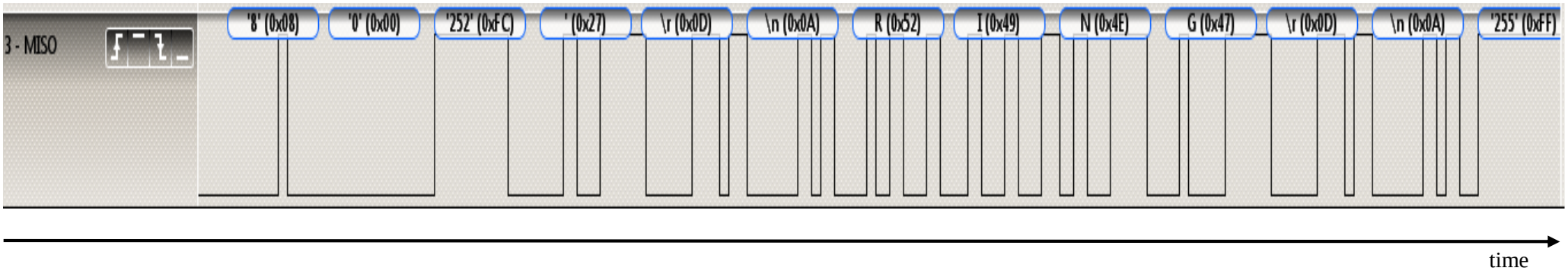
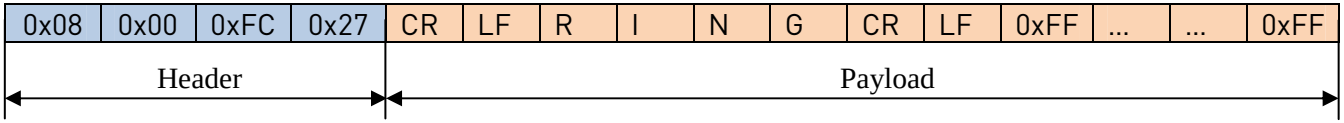


Fig. 15: MISO: Unsolicited RING Message

The SPI Frame format sent by the module (Slave) on the MISO line is the following:



The header size is always long 4 bytes. To explain the header bytes refer to Fig. 16.

The payload-length is set to a fixed value of 2044 bytes. In this example 8 characters are transported by the payload section, but all 2044 bytes are transmitted.

The Fig. 16 shows the header format and the order in which the bytes are transmitted on the MISO line.

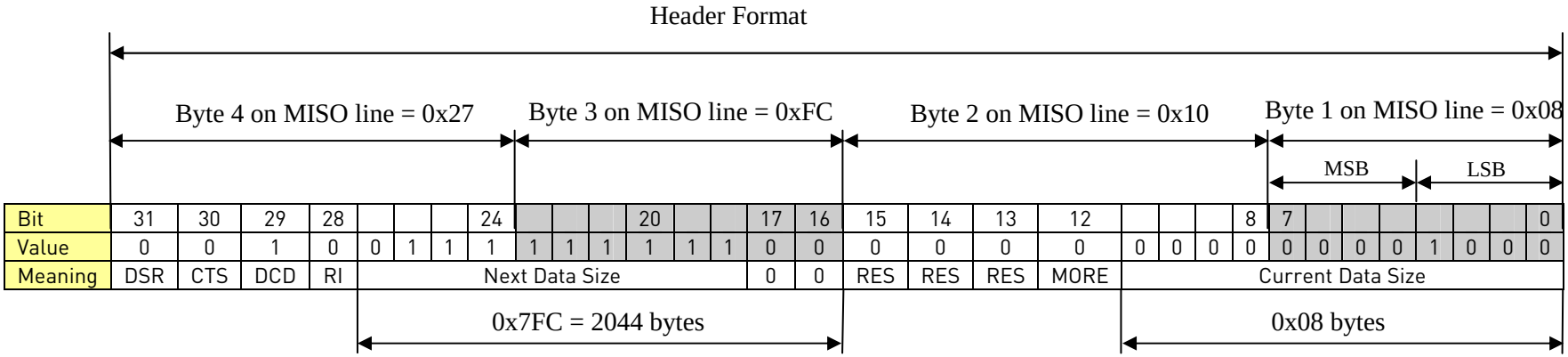


Fig. 16: Header Format

- Current Data Size: characters number inserted into the payload section (regardless the characters number inserted, the payload is always long 2044 bytes)
- MORE: indicates if new data will be sent (MORE = 0 means no more data)
- RES: reserved
- Next Data Size: the module sets Next Data Size to 2044 bytes.
- RI: supported
- DCD: supported
- CTS: supported
- DSR: supported

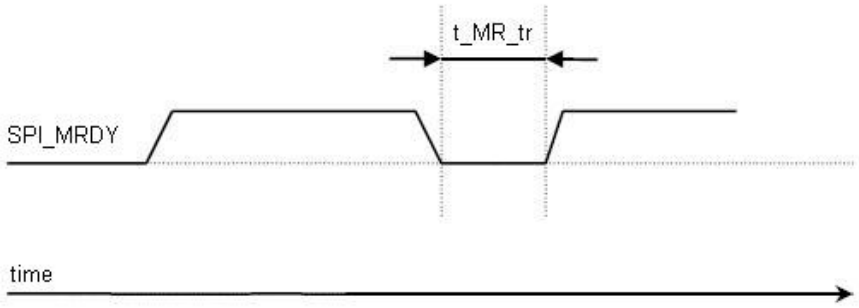
4. Appendix

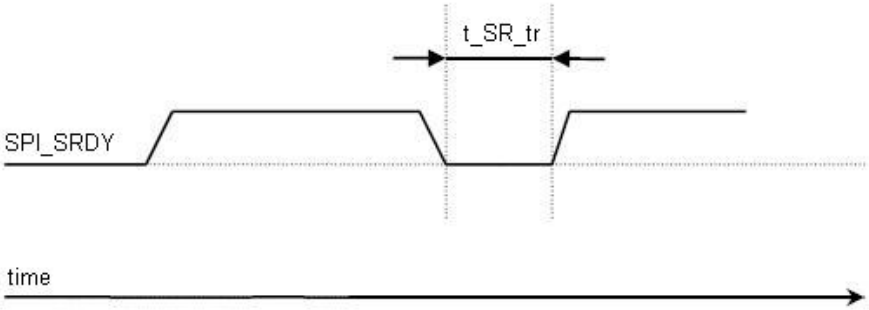
4.1. MRDY & SRDY Timing

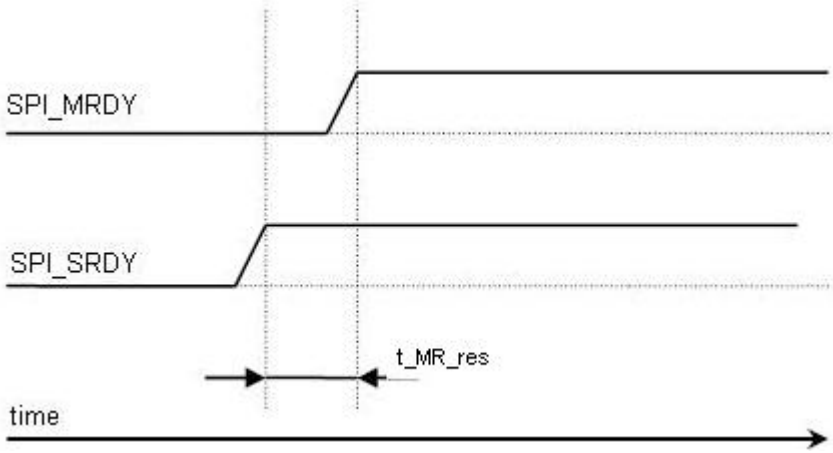
The table below summarizes the timing values of the SPI_MRDY and SPI_SRDY lines.

Timer	Description	Value	Comment
t_MR_tr	SPI_MRDY min transition time	8 ms	typical
t_SR_tr	SPI_SRDY min transition time	1.5 ms	typical
t_MR_res	SPI_MRDY response time	Depending on AP	
t_SR_res	SPI_SRDY response time	430 us	typical

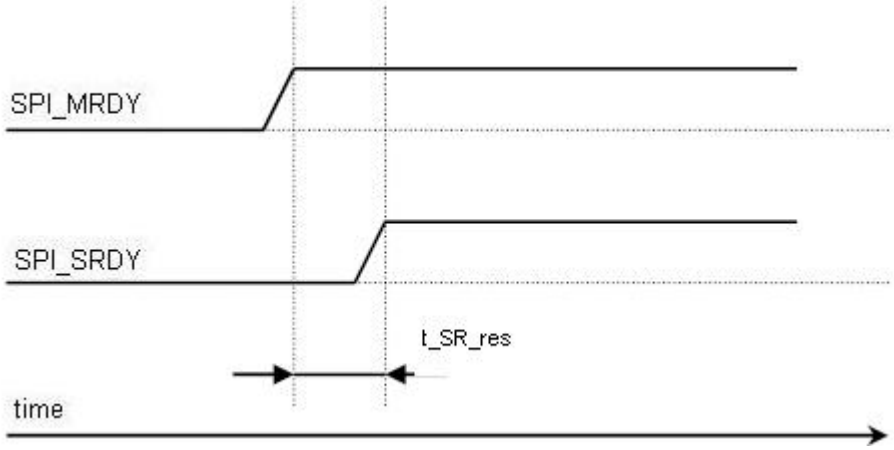
The following figures show the timings of the SPI_MRDY and SPI_SRDY lines.







MRDY response



SRDY response

5. Document History

Revision	Date	Changes
ISSUE#0	2011-09-26	Preliminary release
ISSUE#1	2013-02-04	Overall revision of the document

6. Abbreviation and acronyms

CR	Carriage Return
CTS	Clear To Send
DCD	Data Carrier Detect
DCE	Data Communication Equipment
DSR	Data Set Ready
DTE	Data Terminal Equipment
DTR	Data Terminal Ready
HW	Hardware
LF	Line Feed
LSB	Least Significant Byte
MISO	Master In Slave Out
MOSI	Master Out Slave In
MSB	Most Significant Byte
RI	Ring Indicator
RTS	Request To Send
SPI	Serial Peripheral Interface
URC	Unsolicited Result Code