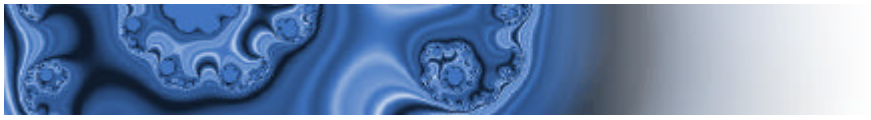
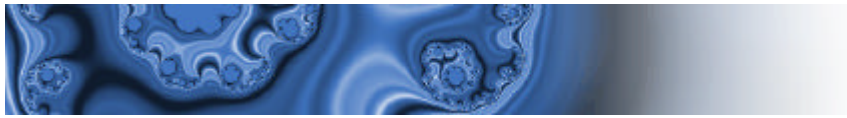


**AX-V**

**Serial Communication Protocol**

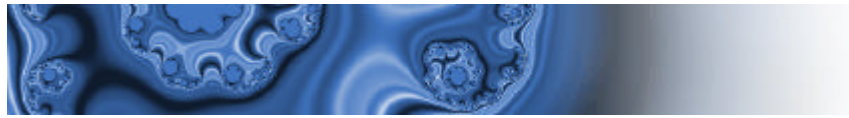
**SLINK3**





# Summary

<b>1</b>	<b>INTRODUCTION.....</b>	<b>4</b>
<b>2</b>	<b>GENERAL FRAME (DATA PACKET) DESCRIPTION.....</b>	<b>4</b>
<b>3</b>	<b>COMMAND CODES DESCRIPTION .....</b>	<b>6</b>
<b>4</b>	<b>RESPONSE CODE DESCRIPTION .....</b>	<b>6</b>
<b>5</b>	<b>PARAMETERS EXCHANGE BETWEEN MASTER AND SLAVE.....</b>	<b>6</b>
5.1	WRITE TYPED PARAMETER: MASTER->SLAVE .....	8
5.2	WRITE TYPED PARAMETER: SLAVE->MASTER .....	8
5.2.1	<i>OK Condition Answer.....</i>	<i>8</i>
5.2.2	<i>Error Condition Answer.....</i>	<i>8</i>
5.2.3	<i>Example.....</i>	<i>9</i>
5.3	READ TYPED PARAMETER: MASTER->SLAVE .....	10
5.4	READ TYPED PARAMETER: SLAVE->MASTER .....	10
5.4.1	<i>OK Condition Answer.....</i>	<i>10</i>
5.4.2	<i>Error Condition Answer.....</i>	<i>10</i>
5.4.3	<i>Example.....</i>	<i>11</i>



## 1 INTRODUCTION

SLINK3 protocol define the data exchange format between a Master device (PC or PLC) and a Slave device (AX-V Drive) over an RS485 serial line.

Scope of this document is to describe the implementation of SLINK3 protocol.

In the following pages the *frame* structure and the implemented data transfer functions will be described.

## 2 GENERAL FRAME (DATA PACKET) DESCRIPTION

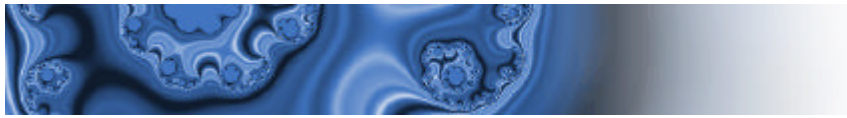
SLINK3 specify a data transfer organised in bytes. Each byte has a special meaning depending on the type of information to be exchanged.

The structure of a frame is shown in following table:

BYTE	NAME	VALUE	NOTES
1	START_CODE	01h	Always the same
2	ADDRESS	0 .. 127	Slave Address
3	CONTROL	00xxxx00	Frame identifier
4	LENGHT	8...255	Frame length in bytes
5	ORDER/INFOS	Bit0: 0 = Frame Master 1 = Frame slave	Frame Information
6	COMMAND/RESPONSE (LOW)	0...255	Command/Response lower byte
7	COMMAND/RESPONSE (HIGH)	0...255	Command/Response higher byte
8..Length-1	DATA	Dipende dal tipo di dati	Frame Content
Length	CHECKSUM	XOR dei Byte 2...(Length-1)	Vertical parity

### Columns description:

BYTE: Represent the order in which bytes are sent/received  
NOME : Byte mnemonic name  
VALORE: Where specified is the value of the byte.  
NOTE: Short description



**Fields description:**

START CODE:	Is the header byte of every frame. Its value is always 01.
ADDRESS:	Identify the slave device addressed by the frame. It is possible to specify up to 127 different slave devices. The special address 128 specify the broadcast communication that is all slave devices will receive the data. In this case Slave devices don't answer to Master message.
CONTROL:	Every frame transmitted by the master can be identified by a dedicated number in this field. The addressed slave will repeat the same number in the corresponding field of its answer. This can help the Master to recognise lost or overlapping frames.
LENGTH:	Complete frame length in bytes
ORDER/INFOS:	Frame information: 00h = Master Message; 01h = Slave Message
COMMAND/RESPONSE:	In Master messages represent the type of data exchange (ex. Read or Write parameter). In Slave messages represent the result of service required by the master. See Chapter 3 and 4 for a detailed list of possible values.
DATA:	Contain the actual data transmitted with the frame (for example contain the value to be written and the corresponding parameter index).
CHECKSUM:	XOR of bytes 2 ... (LENGTH-1)



### **3 COMMAND CODES DESCRIPTION**

Only two possible commands are implemented in AX-V drives:

COMMAND	VALUE
Read Typed Prameter	0026H
Write Typed Parameter	0027H

### **4 RESPONSE CODE DESCRIPTION**

Possible Response codes are:

RESPONSE	VALORE
OK no error	0000H
Parameter not exist	0001H
Unknown Command	0018H
Type Unknown	001EH
Frame error	0097H

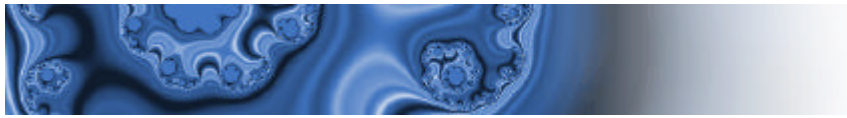
### **5 PARAMETERS EXCHANGE BETWEEN MASTER AND SLAVE**

The transmission of Typed Parameters allow to transfer parameters with a specified data format. Parameters are identified by an INDEX and a FORMAT. With this information the slave device can check the correspondence between received data type and internal database data type.

Following table show a list of supported data types:

TYPE	FORMAT FIELD VALUE	DESCRIPTION
VOID	00H	Tipo non specificato
BOOLEAN	01H	Parametro bit (1 byte)
SHORT_INT	03H	Intero 16 bit (2 byte)
LONG_INT	04H	Intero 32 bit (4 byte)
UNSIGNED_SHORT_INT	06H	Intero non segnato 16 bit (2 byte)
UNSIGNED_LONG_INT	07H	Intero non segnato 32 bit (4 byte)
FLOAT_SINGLE_PRECISION	08H	Virgola mobile (4 byte)

NOTE: for data types longer than 1 byte values are transmitted starting from the lower byte (see following examples).



**Examples:**

1. Short int (2 byte): value 1234h

The transmission order is:

Byte VALUE 0: 34H

Byte VALUE 1: 12H

2. Long int (4 bytes): value 12345678H

The transmission order is:

Byte VALUE 0: 78H

Byte VALUE 1: 56H

Byte VALUE 2: 34H

Byte VALUE 3: 12H

3. Floating point numbers (4 bytes) value 3.141592654

Floating point numbers are expressed according to IEEE-754 (single precision):

Byte 0	Byte 1	Byte 2	Byte 3
Mmmmmmm	mmmmmmmm	emmmmmmm	seeeeeee

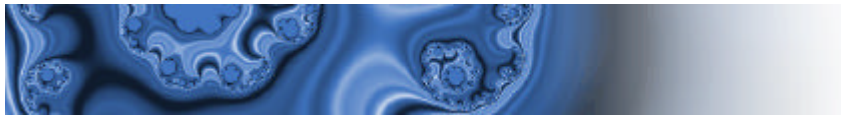
m = mantissa, e = exponent s = sign

Byte VALUE 0: DBH

Byte VALUE 1: 0FH

Byte VALUE 2: 49H

Byte VALUE 3: 40H



### 5.1 WRITE Typed Parameter: MASTER->SLAVE

BYTE	NAME	VALUE	NOTES
1	START_CODE	01h	Always the same
2	ADDRESS	0 .. 127	Slave Address
3	CONTROL	00xxxx00	Frame identifier
4	LENGHT	8...255	Frame length in byte
5	ORDER/INFOS	00h	Frame Information
6	COMMAND (LOW)	27h	Lower byte of Command
7	COMMAND (HIGH)	00h	Higher byte of Command
8	FORMAT	00h..08h	Parameter format
9	INDEX (LOW)	0..255	Lower byte of Parameter Index (IPA)
10	INDEX (HIGH)	0..255	Higher byte of Parameter Index (IPA)
11	VALUE (Byte 0)	0...255	Byte 0 of Value
12	VALUE (Byte 1)	0...255	Byte 1 of Value
13..Length-1	VALUE (Byte 2..n)	0...255	Byte 2..n of Value
Length	CHECKSUM	XOR of Bytes 2...(Length-1)	Vertical parity

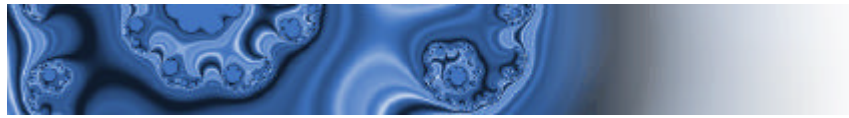
### 5.2 WRITE Typed Parameter: SLAVE->MASTER

#### 5.2.1 OK Condition Answer

BYTE	NAME	VALUE	NOTES
1	START_CODE	01h	Always the same
2	ADDRESS	0 .. 127	Slave Address
3	CONTROL	00xxxx00	Frame identifier
4	LENGHT	08h	Frame length in byte
5	ORDER/INFOS	01h = Slave Message	Frame Information
6	RESPONSE (LOW)	00h	Lower byte of Response
7	RESPONSE (HIGH)	00h	Higher byte of Response
8	CHECKSUM	XOR of Bytes 2...(Length-1)	Vertical Parity

#### 5.2.2 Error Condition Answer

BYTE	NAME	VALUE	NOTES
1	START_CODE	01h	Always the same
2	ADDRESS	0 .. 127	Slave Address
3	CONTROL	00xxxx00	Frame identifier
4	LENGHT	8...255	Frame length in byte
5	ORDER/INFOS	01h = Slave Message	Frame Information
6	RESPONSE (LOW)	Error Code (Low)	Lower byte of Response
7	RESPONSE (HIGH)	Error Code (High)	Higher byte of Response
8	CHECKSUM	XOR dei Byte 2...(Length-1)	Vertical Parity



### 5.2.3 Example

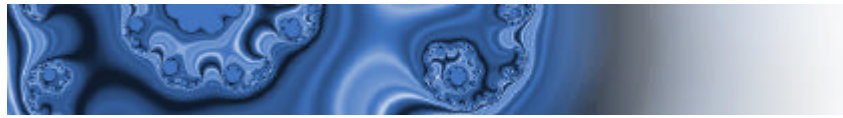
Write Parameter SHORT\_INT at IPA (INDEX) 1111 (value to write = 200)

#### Master Frame (Query):

byte1	01	Start Code (always 01)
byte2	00	Slave Address
byte3	3C	Frame Identifier. Can be any number: the slave will insert the same number in the corresponding field of answer message.
byte4	0D	Frame length in bytes (13 bytes)
byte5	00	00 = Master Message
byte6	27	Command 27 = Write Parameter
Byte7	00	Always 00
byte8	03	Parameter type: 03=SHORT_INT
byte9	57	Parameter Index (IPA) LOW Byte
byte10	04	Parameter Index (IPA) HIGH Byte 1011 in hexadecimal notation = 0457h
byte11	C8	Value to write (LOW byte): 200 = 00C8h
byte12	00	Value write (HIGH byte) : 200 = 00C8h
byte13	8E	Checksum: XOR of bytes 2...12

#### Slave Frame (Answer):

byte1	01	Start Code (always 01)
byte2	00	Slave Address
byte3	3C	Frame Identifier. The slave insert the same number found in the corresponding field of Master message.
byte4	08	Frame length in bytes (13 bytes)
byte5	01	01 = Slave Message
byte6	00	Always 00 in answer
byte7	00	Always 00 in answer
byte8	35	Checksum: XOR of bytes 2...7



### 5.3 READ Typed Parameter: MASTER->SLAVE

BYTE	NAME	VALUE	NOTES
1	START_CODE	01h	Always the same
2	ADDRESS	0 .. 127	Slave Address
3	CONTROL	00xxxx00	Frame identifier
4	LENGHT	0Bh	Frame length in byte
5	ORDER/INFOS	00h	Frame Information
6	COMMAND (LOW)	26h	Lower byte of Command
7	COMMAND (HIGH)	00h	Higher byte of Command
8	FORMAT	00h..08h	Parameter format
9	INDEX (LOW)	0..255	Lower byte of Parameter Index (IPA)
10	INDEX (HIGH)	0..255	Higher byte of Parameter Index (IPA)
11	CHECKSUM	XOR dei Byte 2...10	Vertical parity

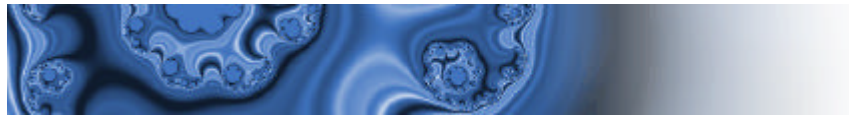
### 5.4 READ Typed Parameter: SLAVE->MASTER

#### 5.4.1 OK Condition Answer

BYTE	NAME	VALUE	NOTES
1	START_CODE	01h	Always the same
2	ADDRESS	0 .. 127	Slave Address
3	CONTROL	00xxxx00	Frame identifier
4	LENGHT	8...255	Frame length in byte
5	ORDER/INFOS	01h = Slave Message	Frame Information
6	RESPONSE (LOW)	00h	Lower byte of Response
7	RESPONSE (HIGH)	00h	Higher byte of Response
8	FORMAT	00h..08h	Parameter format
9	INDEX (LOW)	0..255	Lower byte of Parameter Index (IPA)
10	INDEX (HIGH)	0..255	Higher byte of Parameter Index (IPA)
11	VALUE (Byte 0)	0...255	Byte 0 of Value
12	VALUE (Byte 1)	0...255	Byte 1 of Value
13..Length-1	VALUE (Byte 2..n)	0...255	Byte 2..n of Value
Length	CHECKSUM	XOR dei Byte 2...(Length-1)	Vertical parity

#### 5.4.2 Error Condition Answer

BYTE	NAME	VALUE	NOTES
1	START_CODE	01h	Always the same
2	ADDRESS	0 .. 127	Slave Address
3	CONTROL	00xxxx00	Frame identifier
4	LENGHT	08h	Frame length in byte
5	ORDER/INFOS	01h = Slave Message	Frame Information
6	RESPONSE (LOW)	Error Code (Low)	Lower byte of Response
7	RESPONSE (HIGH)	Error Code (High)	Higher byte of Response
8	CHECKSUM	XOR of Bytes 2...7	Vertical parity



### 5.4.3 Example

#### Read Parameter SHORT\_INT at IPA (INDEX) 1009

##### Master Frame (Query):

byte1	01	Start Code (always 01)
byte2	02	Slave Address
byte3	00	Frame Identifier. Can be any number: the slave will insert the same number in the corresponding field of answer message.
byte4	0B	Frame length in bytes (11 bytes)
byte5	00	00 = Master Message
byte6	26	Command 26 = Read Parameter
byte7	00	Always 00
byte8	00	Format: 00=Generic (the slave device will specify the type of parameter in the answer);
byte9	F1	Parameter Index (IPA) LOW Byte
byte10	03	Parameter Index (IPA) HIGH Byte 1009 in hexadecimal notation = 03F1h
byte11	DD	Checksum: XOR of bytes 2...10

##### Slave Frame (Answer):

byte1	01	Start Code (always 01)
byte2	02	Slave Address
byte3	00	Frame Identifier. The slave insert the same number found in the corresponding field of Master message.
byte4	0D	Frame length in bytes (13 bytes)
byte5	01	01 = Slave Message
byte6	00	Always 00 in answer
byte7	00	Always 00
byte8	03	Format 03 = SHORT_INT (see table in chapter 5)
byte9	F1	Parameter Index (IPA) LOW Byte
byte10	03	Parameter Index (IPA) HIGH Byte 1009 in hexadecimal notation = 03F1h
byte11	7D	Parameter Value (LOW byte)
byte12	00	Parameter Value (HIGH byte) In this example the value is 125 (HEX 007Dh)
byte13	82	Checksum: XOR of bytes 2...12