Q.link protocol

1 Revision history

Version	Date	Responsible	Description
1.0	03-10-16	JUK	Document created

Table of Contents

2 References

Ref. 1 Integrated Servo Motors – Technical Manual LB0048-05GB by JVL Industri Elektronit A/S

3 Introduction

This document contains the description of the Q.link protocol. This protocol is currently use for communication between the Q.control target and the MAC00FR02 motor interface.

4 General

The Q.link protocol is a master/slave based protocol. It uses request/reply for data direction control. Electrical and physical the protocol uses a RS485 based network as data carrier. The network consists of a master and 1-16 slave nodes. Each slave must have their own unique node number.

Not all slave nodes may support all the described commands.

5 Protocol format

The format of the packets:

Header	Command data	CRC

5.1 Header

Preample high byte	Preample low byte	Length (1 byte)	Command (1 byte)	Node (1 byte)

Name	*Description*
Preample high/low bytes	Two bytes of preample. Both must be 0xA5
Length	This is the length of the packet. The length is min. 4 (Node, Command and CRC) plus the size of the Command specific data
Command	The command varies. Please see the ”Protocol commands” chapter
Node	The slave node number. Valid values are 1-16

5.2 Command Data

Command specific data (0-26 bytes)

Name	*Description*
Command specific data	This is the data attached to the command. See ”Protocol commands” chapter for more information. The size of this data may not exceed 26 bytes

5.3 CRC

CRC high byte	CRC low byte

Name	*Description*
CRC high/low bytes	Two byte used for the CRC16. The CRC16 must follow the CITT-16 standard for calculation. The data used for the CRC calculation are: Length, Command, Node and Command specific data

6 Communication

The master issues a request to a node on the network. The slave responces to the master with data according to the command issued by the master. The slave node uses it's own node number in the response.

6.1 Timeout

The slave node must send a response within the time of 3 bytes or else a timeout error has occured. If the network is running at 115200 bit/sec with 8 data bit, 1 stop bit (10 bits pr. transferred byte) the timeout time wil be:
1 / ((115200 bit/sec / 10 bit/byte) * 3 = 260 usec

In case of a timeout error occures the master expects no response form the current slave. The master can choose to retry the request to the current slave or proceed to the node.

Note: The Node program load 0x06 command has a response timeout of 10 ms.

6.2 CRC16

To each packet is attached a 16 bit CRC. This CRC is calculated according to the CITT-16 standard and is based upon the command, node and the command specific data. If the calculated CRC of a recieved packet is not equal, with the CRC contained in the recieved packet, a CRC error has occured. If such error has occured the whole packet is discarded. If a master recieves a packet with CRC error it can retry the request. If a slave recieves a packet with CRC error it will ignore the packet.

6.3 Byte stuffing

To ensure that the preample (two bytes with both 0xA5) is unique in a packet byte stuffing is used. In case of an occurance of a 0xA5 a 0x00 byte is inserted after the 0xA5 byte exept from the preample.

The stuffing should be done just before transmission careless of the command, length, etc. and the destuffing should be done just after recieving before the command, lengt, CRC etc. are examined.

Example with a write/read request from the master:

Before byte stuffing at the master:

0xA5	0xA5	0x07	0x83	0x01	0xA5	0xFF	0xA5	0xE3	0x05

After byte stuffing at the master (stuffed bytes marked with gray):

0xA5	0xA5	0x07	0x83	0x01	0xA5	0x00	0xFF	0xA5	0x00	0xE3	0x05

And after byte destuffing at the slave:

0xA5	0xA5	0x07	0x83	0x01	0xA5	0xFF	0xA5	0xE3	0x05

6.4 Broadcast

For broadcasting to slave use node number 0x00. Please see the specific command for more information about possible broadcast.

7 Protocol Commands

This chapter contains the different commands in the serial protocol. The command specific data is marked grey.

7.1 Node setup 0x01

This command is used for setting up the slave node. The command byte in the packet must be 0x01.

Note: This packet can be sent as a broadcast and the slave responds to confirm setup. If the request is sent as at breadcast only **one** node may be connected to the master.

7.1.1 Request

Header	Node number (1 byte)	T1 delay (1 byte)	T2 delay (1 byte)	Position type (1 byte)	CRC

Name	*Description*
Node number	This is the desired slave node number. The values 1-16 is valid. The default value is 1
T1 delay	This byte contains the T1 delay time. The time is in 100 ms steps so that up to 25,5 sec delay is possible. The T1 is used to delay the time from when the I1 is set high to the O1 is set high. Default value is 10 (1 sec delay)
T2 delay	This byte contains the T2 delay time. The time is in 100 ms steps so that up to 25,5 sec delay is possible. The T2 is used to delay the time from when the I1 is set low to the O1 is set low. Default value is 10 (1 sec delay)
Position type	This is used to define which position request should be used for requesting the position serial from the motor. Following types is supported: 0: No position is requested (default) 1: JVL MacTalk position request 2: Metronix position request

7.1.2 Response

Header	CRC

Note: Command 0x01 response contains no command specific data.

7.2 Node info 0x02

This command is used for retrieving slave node information. The command byte in the packet must be 0x02.

Note: It's possible to send this command as broadcast. In such case only **one** node may be connected to the master.

7.2.1 Request

Header	CRC

7.2.2 Response

Header	Version major	Version minor	Product ID (2 bytes)	T1 delay (1 byte)	T2 delay (1 byte)	Position type (1 byte)	Serial number (4 bytes)	CRC

Name	*Description*
Version major	This byte contains the major number of the firmware version
Version minor	The minor number of of the firmware version is written in this byte
Product ID	This is a product ID for the slave
T1 delay	This byte contains the T1 delay time. The time is in 100 ms steps so that up to 25,5 sec delay is possible. The T1 is used to delay the time from when the I1 is set high to the O1 is set high. Default value is 10 (1 sec delay)
T2 delay	This byte contains the T2 delay time. The time is in 100 ms steps so that up to 25,5 sec delay is possible. The T2 is used to delay the time from when the I1 is set low to the O1 is set low. Default value is 10 (1 sec delay)
Position type	This is used to define which position request should be used for requesting the position serial from the motor. Following types is supported: 0: No position is requested (default) 1: JVL MacTalk position request 2: Metronix position request
Serial number	These 4 bytes is the serial number. The MSB is the most left byte

7.3 Node start 0x03

This command is used for starting one or all nodes at the same time. The packet can be broadcasted to all slave nodes and none of the slaves may respond to this request. The command byte in the packet must be 0x03.

7.3.1 Request

Header	CRC

7.3.2 Response

There's no response to the request.

7.4 Node stop 0x04

This command is used for stopping one or all nodes at the same time. The packet can be broadcasted to all slave nodes and none of the slaves may respond to this request. The command byte in the packet must be 0x04.

7.4.1 Request

Header	CRC

7.4.2 Response

There's no response to the request.

7.5 Node sync 0x05

This command is used for syncronisation all nodes. The packet is broadcasted to all slave nodes and none of the slaves may respond to this request. The command byte in the packet must be 0x05.

7.5.1 Request

Header	Counter (1 byte)	CRC

Name	*Description*
Counter	The Counter is used for the slaves to see if it missed a sync packed by checking the increased counter. The master is responsible for sending the sync request with a defined interval. The Counter is increased in each packet

7.5.2 Response

There's no response to the request.

7.6 Node program load 0x06

This command is used for loading a Intel HEX program file to the node. To initiate the program load the first request sent to the slave must be empty. When the slave replies to the request it's ready for receiving the program data. The program data may consist of one or more packets. The request and reponse contains a Counter to keep track of which program data has been received. The first request with the program data the Counter value is 0. The master increases the Counter each time a request is succesfully received.

The command byte in the packet must be 0x06.

Note: The response timeout for the Node program load is 10 ms which differs from the other commands.

7.6.1 Request

Header	Counter (1 byte)	Program data (0-26 bytes)	CRC

Name	*Description*
Counter	Program load message counter.
Program data	This is the Intel HEX file program data

7.6.2 Response

Header	Counter (1 byte)	CRC

Name	*Description*
Counter	This byte muest be the counter value received from the master

7.7 Node reset 0x07

This command is used for resetting the node. After the slave node has send the response it resets. The command byte in the packet must be 0x07.

7.7.1 Request

Header	CRC

7.7.2 Response

Header	CRC

Note: Command 0x07 response contains no command specific data.

7.8 Motor response poll 0x15

This command is used for cheking the completion of the previous issued MACTalk or a Motor request command. The response contains the command number of the last completed request and the data which the request to the motor returned.

7.8.1 Request

Header	CRC

7.8.2 Response

Header	Last completion (1 byte)	Motor data (0-25 bytes)	CRC

Name	*Description*
Last completion	If this is zero the last issued MACTalk or Motor request command is not yet completed. If the last command was completed this byte is none zero
Motor data	This is depended on the MACTalk or Motor request command. See ref. 1 or the specification for the used motor for more information

7.9 Motor request 0x16

This command is used for sending a request for the connected motor node. The data sent to the motor is depended on the specific motor. The data defined in ”Motor specific data” is sent directly to the motor. The request is not of this command is not an indication that the motor request is completed but only that the request has been recieved by the node. The real response from the motor must be read with the Motor response poll command

7.9.1 Request

Header	Motor specific data (1-26 bytes)	CRC

Name	*Description*
Motor specific data	Data which contains the data for the motor request

7.9.2 Response

Header	CRC

Note: Command 0x16 response contains no command specific data.

7.10 Motion 0x20

This command is used for providing a connected motor with segment data for its motion path and reading and writing the I/Os. Its possible to send from 0 to 12 words of segment data in each request. The response will contain the general inputs, latest motor position, status and the number of remaining segment data spaces left in the FIFO.
The request and response contains a Counter value to keep track of which Motion has been successfully received. In case of the response from the slave is lost, the master will timeout on the current response. The master will resend the request with the previous Counter value but the slave will ignore the message because the previous message with the same Counter value had already been received.

7.10.1 Request

Header	Counter (1 byte)	Output (1 byte)	1st segment data high byte	1st segment data low byte	2nd - 12th segment data high byte	2nd - 12th segment data low byte	CRC

Name	*Description*
Output	This byte is used for wrting the outputs at the motor (if any). The least significant bit is output 1 and the most significant bit is output 8
Counter	This byte used for keeping track of the motion messages. The value is increased by the master each time it has received a response from the slave. The value can be 0-255 and is set to 0 when the 255 value is increased
1st segment data high byte	The MSB of the first word of segment data
1st segment data low byte	The LSB of the first word of segment data
2nd - 12th segment data high byte	This is the MSB of the following segment data. All segment data word must contain a MSB followed by a LSB
2nd - 12th segment data low byte	This is the LSB of the following segment data words

7.10.2 Response

Header	Counter (1 byte)	Input (1 byte)	Position (4 bytes)	Motor status high byte	Motor status low byte	Remain space in FIFO	CRC

Name	*Description*
Counter	This byte muest be the counter value received from the master
Input	This is the 8 general inputs at the motor (if present). The least significant bit is input 1 and the most significant bit is input 8
Position	This is the 4 bytes of the motor position. The most left byte is the MSB. Its value is depended on the encoder used for the motor. See Node setup 0x01 command for position type select
Motor status high byte	This is the MSB of the status word. Its value has to be defined
Motor status low byte	This is the LSB of the status word. Its value is defined as following: Bit 0: Motor running Bit 1: Position valid
Remain space in FIFO	This value indicates the number of remaining segment data words in the FIFO buffer

7.11 Motor position 0x21

This command is used for reading the current motor position.

7.11.1 Request

Header	CRC

7.11.2 Response

Header	Position (4 bytes)	CRC

Name	*Description*
Position	This is the 4 bytes of the motor position. The first byte is the MSB. Its value is depended on the encoder used for the motor. See Node setup 0x01 command for position type select

7.12 Clear FIFO 0x22

This command is used for clearing the FIFO containing the segment data.

7.12.1 Request

Header	CRC

7.12.2 Response

Header	CRC

7.13 Preset motor position 0x23

This command is used for setting the position in the connected motor. It is depended on the position type set with Node setup command. The Preset command may not be sent when the motor is running (Node start command).

7.13.1 Request

Header	Position (4 bytes)	CRC

Name	*Description*
Position	This is the 4 bytes of the motor position. The first byte is the MSB. Its value is depended on the encoder used for the motor. See Node setup 0x01 command for position type select

7.13.2 Response

Header	CRC

Note: Command 0x82 response contains no command specific data.

7.14 Read 0x81

This command reads a defined number of inputs. The command byte in the packet must be 0x81.

7.14.1 Request

Header	Data size (1 byte)	Offset (1 byte)	CRC

Name	*Description*
Data size	The size of the requested data. Max requested size is 26 bytes
Offset	This offset is used to address specific bytes relative to least significant byte. If this value is 2, the first data byte that will be read, is byte number 3.

7.14.2 Response

Header	Data (1-26 byte(s))	CRC

Name	*Description*
Data	The data which is read from the inputs. The first byte, counting from left, is the least significant byte. The most significant byte is the last byte when counting from left

7.15 Write 0x82

This command makes both an output write and an input read in the same request/response. The command byte in the packet must be 0x82.

7.15.1 Request

Header	Offset (1 byte)	Data (1-25 bytes)	CRC

Name	*Description*
Offset	This offset is used to address specific bytes relative to least significant byte. If this value is 2 this first data byte will be written to byte number 3.
Data	The data which is written to the outputs. The first byte, counting from left, is the least significant byte. The most significant byte is the last byte when counting from left

7.15.2 Response

Header	CRC

Note: Command 0x82 response contains no command specific data.

7.16 Write/Read 0x83

This command makes both an output write and an input read in the same request/response. The command byte in the packet must be 0x83.

7.16.1 Request

Header	DO1 (1 byte)	DO2 (1 byte)	DO3 (1 byte)	CRC

Name	*Description*
DO1	The least significant byte of the outputs. Contains digital output 1-8. The least significant bit is output 1 and the most significant bit is output 8
DO2	Contains digital output 9-16. The least significant bit is output 9 and the most significant bit is output 16
DO3	The most significant byte of the outputs. Contains digital output 17-24. The least significant bit is output 17 and the most significant bit is output 24

7.16.2 Response

Header	DI1 (1 byte)	DI2 (1 byte)	DI3 (1 byte)	CRC

Name	*Description*
DI1	The least significant byte of the inputs. Contains digital input 1-8. The least significant bit is input 1 and the most significant bit is input 8
DI2	Contains digital input 9-16. The least significant bit is input 9 and the most significant bit is input 16
DI3	The most significant byte of the inputs. Contains digital input 17-24. The least significant bit is input 17 and the most significant bit is input 24

7.17 Set outputs 0x84

This command sets outputs specified by a mask. The command byte in the packet must be 0x84.

Note: This command is limited to 24 outputs for the fast I/O unit.

7.17.1 Request

Header	Mask1 (1 byte)	Mask2 (1 byte)	Mask3 (1 byte)	CRC

Name	*Description*
Mask1	The least significant mask byte of the outputs. Contains mask for output 1-8. The least significant bit sets output 1 and the most significant bit sets output 8
Mask2	Contains mask output 9-16. The least significant bit sets output 9 and the most significant bit sets output 16
Mask3	The most significant mask byte of the outputs. Contains mask for output 17-24. The least significant bit sets output 17 and the most significant bit sets output 24

7.17.2 Response

Header	CRC

Note: Command 0x84 response contains no command specific data.

7.18 Clear outputs 0x85

This command clears outputs specified by a mask. The command byte in the packet must be 0x85.

Note: This command is limited to 24 outputs for the fast I/O unit.

7.18.1 Request

Header	Mask1 (1 byte)	Mask2 (1 byte)	Mask3 (1 byte)	CRC

Name	*Description*
Mask1	The least significant mask byte of the outputs. Contains mask for output 1-8. The least significant bit clears output 1 and the most significant bit clears output 8
Mask2	Contains mask output 9-16. The least significant bit clears output 9 and the most significant bit clears output 16
Mask3	The most significant mask byte of the outputs. Contains mask for output 17-24. The least significant bit clears output 17 and the most significant bit clears output 24

7.18.2 Response

Header	CRC

Note: Command 0x85 response contains no command specific data.

7.19 Loop back 0xA4

This command is for testing purpose. The data specified in the data field in the request is returned in the response.

7.19.1 Request

Header	Offset (1 byte)	Data (1-25 bytes)	CRC

Name	*Description*
Offset	For testing purpose only. Is returned in the response
Data	Data for the request. This data is returned in the response

7.19.2 Response

Header	Offset (1 byte)	Data (1-25 bytes)	CRC

Name	*Description*
Offset	Supplied by the request packet
Data	Data for the response. This data is supplied by the request packet