LinMot®

DeviceNet™

Documentation of the DeviceNet Interface of the following Controllers:

- **E1100-DN (-HC, XC)**
- **E1100-GP (-HC, XC)**
- **E1130-DP (-HC, XC)**
- **B1100-GP (-HC, XC)**

DeviceNet Interface V3.12

User Manual

© 2011 NTI AG
This work is protected by copyright.
Under the copyright laws, this publication may not be reproduced or transmitted in any form, electronic or mechanical, including photocopying, recording, microfilm, storing in an information retrieval system, not even for didactical use, or translating, in whole or in part, without the prior written consent of NTI AG.
LinMot® is a registered trademark of NTI AG.
Note
The information in this documentation reflects the stage of development at the time of press and is therefore without obligation. NTI AG reserves itself the right to make changes at any time and without notice to reflect further technical advance or product improvement.

Document version 3.12a / May 2011

1. System Overview

The LinMot DeviceNet controllers E1100-DN(-HC) and E1100-GP(-HC) supports the DeviceNet communication profile. Further information on DeviceNet can be found under: http://www.odva.org/

The LinMot DeviceNet controller is a UCMM Group 3 capable slave. And supports polled IO runtime data transmission. With the B1100 device the Servo act as group 2 only server.

2. Installation on Servo Controller

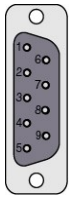
For installing the DeviceNet firmware on the servo controller, start the LinMot-Talk software and press the install firmware button  . Choose the file “Firmware_Buildxxxxxxxx.sct” and press “Open“. The wizard will guide you through the installation. When asking for the application software choose DeviceNet”:

Press ok and follow the rest of the wizard.

3. Connecting the CAN bus

3.1. Pin Description of the COM Connector (X5):

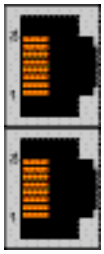
DSBU 9 male:



Pin 1	RS-485 Y	Pin 6	RS-485 B
Pin 2	RS-232 TX	Pin 7	RS-485 Z
Pin 3	RS-232 RX	**Pin 8**	**CAN L**
Pin 4	RS-485 A	**Pin 9**	**CAN H**
Pin 5	**GND**		

3.2. Pin Description of the CMD Connector (X7, X8):

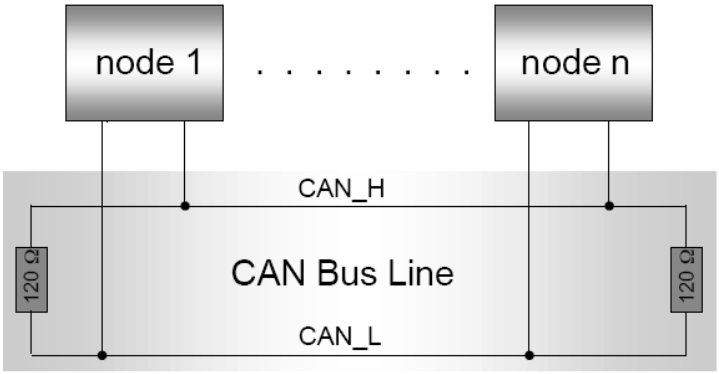
The CMD connector exists only on the E1100-RS(-HC/-XC), E1100-DP(-HC/-XC) and B1100-GP(-HC/-XC) controllers, 2xRJ45 with 1:1 connected signals. Standard twisted pairs: 1/2, 3/6, 4/5, 7/8. Use Ethernet cables according the EIA / TIA 568A standard.



Pin 1	RS485 A
Pin 2	RS485 B
Pin 3	RS485 Y
Pin 4/5	**Ground**
Pin 6	RS485 Z
Pin 7	**CAN H**
Pin 8	**CAN L**

3.3. CAN Termination

The CAN bus must be terminated by two 120 Ohm resistors at both ends of the bus line, according the following scheme:



For easy installation, the LinMot DeviceNet controllers have built in termination resistors, which can be activated, if the LinMot controller is at the end of the bus line, and if there is no termination in the connector.

3.3.1. E1100

S3
ON – OFF
Interface
CAN Term
RS485 Term
RS485/232

S3

The built in termination resistor for the CAN bus can be activated by setting the DIP switch “CAN Term” to “ON”.

ATTENTION: For normal operation S3.4 (Interface) has to be set to ON!

3.3.2. B1100

S4
ON – OFF
Bootstrap
CAN Term
RS485 Term
RS485/232

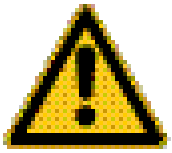
S4

The built in termination resistor for the CAN bus can be activated by setting the DIP switch “CAN Term” to “ON”.

ATTENTION: For normal operation S4.4 (Bootstrap) has to be set to OFF!

4. Power Up Behaviour

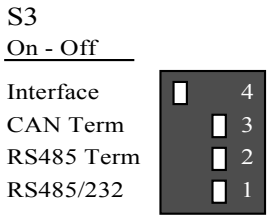
The power up behaviour can be defined over the S3 switches and the S1 and S2 hex switches and the parameter configuration.



With the B1100 there are no switches to define the Baud rate, ID and interface Enable, for this reason all this configuration has to be done by Parameter with LinMotTalk 4 software over CAN-Bus. This CAN bus configuration parameters for the B1100 lay in the OS parameter section.

4.1. Activating and Deactivating the DeviceNet

Over the Interface Switch on the S3.4 switches the DeviceNet protocol can be activated (Switch On) or deactivated (Switch Off).



4.2. ID and Baud Rate Selection

For E1100 controllers, with the default parameterization the baud rate is selected over S1 and the MACID is selected over S2.
For B1100 controllers, the default parameterization sets the baud rate and MACID selector to parameters. Baud Rate is set to 500kBaud, MACID is set to 63dec (3Fh).

4.2.1. Baud Rate Selection

The baud rate can be defined over the S1 hex switch (default setting) or by parameter value.

S1 Baud Rate Code Table	
S1 Value	**Selected Baud Rate**
0	Undefined Baud Rate (set to 125 kBaud)
1	**125 kBaud**
2	**250 kBaud**
3	**500 kBaud**
4	1000 kBaud (Invalid for DeviceNet
5	Undefined Baud Rate (set to 125 kBaud)
.	Undefined Baud Rate (set to 125 kBaud)
7	
.	Undefined Baud Rate

4.2.2. MACID Selection

Like the baud rate the MACID can be defined over the S2 hex switch (default setting), by parameter value or by the S1&S2 hex switches.

S2 ID code table	
S2 Value	**Selected MACID**
0	MACID = 0x00h
1	MACID = 0x01h
2	MACID = 0x02h
.	.
F	MACID = 0x0Fh

S1&S2 ID code table		
S1 Value	**S2 Value**	**Selected MACID**
0	0	MACID = 0x00h
1	1	MACID = 0x01h
2	2	MACID = 0x02h
.	.	.
1	0	MACID = 0x10h
.	.	
3	F	MACID = 0x3Fh
4	0	Invalid MACID
.	..	Invalid MACID
F	F	Invalid MACID

5. DeviceNet Parameters

The DeviceNet Servo Controllers have an additional parameter tree branch, which can be configured with the distributed LinMot-Talk software. With these parameters, the DeviceNet behaviour can be configured. The LinMot-Talk software can be downloaded from http://www.linmot.com under the section download, software & manuals.

Dis-/Enable With the Dis-/Enable parameter the LinMot servo controller can be run without the DeviceNet going online.

DeviceNet\ Dis-/Enable	
Disable	Servo controller runs without DeviceNet.
Enable	Servo controller runs only with a DeviceNet connection. (Default)

IMPORTANT: To activate the DeviceNet Interface, the Dip-Switch “Interface” at the bottom of the drive has to be set to “ON”.

Baud Rate This directory contains the baud rate definition parameters.

Baud Rate Source Select

Defines the source of the baud rate definition.

E1100: CANopen Interface\ Baud Rate \Baud Rate Source Select B1100: OS\Communication\ CAN Configuration\ Baud Rate\ Baud Rate Source Select	
By Hex Switch S1 [1]	E1100 only: CAN bus baud rate dependent on S1 0 = By BTR 1 = 125 kBit/s 2 = 250 kBit/s 3 = 500 kBit/s 4 = 1 Mbit/s
By Parameter	The CAN bus baud rate is selected by the “Baudrate Parameter”: - 125 kBit/s [1] - 250 kBit/s [2] - 500 kBit/s [3] - 1 Mbit/s [4]
By BTR	CAN bus baud rate is defined according to the Bit Timing Register
By DigIn 6 & 5	B1100 only: The baud rate is defined through the state of DigIn5 and DigIn6 at startup. DigIn6 is the most, DigIn5 the least significant bit. DigIn6 DigIn5 BaudRate 0 0 125kBaud 0 1 250kBaud 1 0 500kBaud 1 1 1MBaud

[1] Parameter not available on controllers of the B1100 series.

Baud Rate Parameter Definition

The Baud rate parameter defines the CAN bus baud rate for the DeviceNet connection.

DeviceNet\ Baud Rate Selection\ Baud Rate Parameter Def	
125 kBit/s [1]	CAN bus baud rate = 125 kBit/s
250 kBit/s [2]	CAN bus baud rate = 250 kBit/s
500 kBit/s [3]	CAN bus baud rate = 500 kBit/s (default)

MACID In this section the MACID (controller number) can be configured.

MACID Source Select

The MACID parameter defines the source of the MACID (Node Address).

E1100: CANopen Interface\ MACID\ MACID Source Select B1100: OS\ Communication\ MACID\ MACID Source Select	
By Hex Switch S2	E1100 only:The MACID is determined by the hex switch S2
By Hex Switches S1 and S2	E1100 only: The MACID is determined by the two hex switches S1 and S2
By Parameter	The MACID is determined by parameter setting
By Dig In 1	B1100 only: The MACID is defined by DigIn1 (X13.14) at power up. 0V = ID 0, 24V = ID 1
By Dig In 2..1	B1100 only: The MACID is defined by DigIn2 .. 1 (X13.2 and X13.14) at power up. DigIn2 is the most, DigIn1 the least significant bit. (00b = ID 0, 11b = ID 3)
By Dig In 3..1	B1100 only: The MACID is defined by DigIn3 .. 1 (X13.15, X13.2 and X13.14) at power up. DigIn3 is the most, DigIn1 the least significant bit. (000b = ID 0, 111b = ID 7)
By Dig In 4..1	B1100 only: The MACID is defined by DigIn4 .. 1 (X13.3, X13.15, X13.2 and X13.14) at power up. DigIn4 is the most, DigIn1 the least significant bit. (0000b = ID 0, 1111b = ID 15)
By Dig In 5..1	B1100 only: The MACID is defined by DigIn5 .. 1 (X13.16, X13.3, X13.15, X13.2 and X13.14) at power up. DigIn5 is the most, DigIn1 the least significant bit. (00000b = ID 0, 11111b = ID 31)
By Dig In 6..1	B1100 only: The MACID is defined by DigIn6 .. 1 (X13.4, X13.16, X13.3, X13.15, X13.2 and X13.14) at power up. DigIn6 is the most, DigIn1 the least significant bit. (000000b = ID 0, 111111b = ID 63)
By Dig In 1 + Offset	B1100 only: The MACID is defined by DigIn1 (X14.14) at power up plus the value of 6081h (MACID Parameter Value) as offset. 0V = ID 0, 24V = ID 1 (plus offset).
By Dig In 2..1 + Offset	B1100 only: The MACID is defined by DigIn2 .. 1 (X14.2 and X14.14) at power up plus the value of 6081h (MACID Parameter Value) as offset. DigIn2 is the most, DigIn1 the least significant bit.

	(00b = ID 0, 11b = ID 3 (plus offset))
By Dig In 3..1 + Offset	B1100 only: The MACID is defined by DigIn3 .. 1 (X14.15, X14.2 and X14.14) at power up plus the value of 6081h (MACID Parameter Value) as offset. DigIn3 is the most, DigIn1 the least significant bit. (000b = ID 0, 111b = ID 7 (plus offset))
By Dig In 4..1 + Offset	B1100 only: The MACID is defined by DigIn4 .. 1 (X14.3, X14.15, X14.2 and X14.14) at power up plus the value of 6081h (MACID Parameter Value) as offset. DigIn4 is the most, DigIn1 the least significant bit. (0000b = ID 0, 1111b = ID 15 (plus offset))
By Dig In 5..1 + Offset	B1100 only: The MACID is defined by DigIn5 .. 1 (X14.16, X14.3, X14.15, X14.2 and X14.14) at power up plus the value of 6081h (MACID Parameter Value) as offset. DigIn5 is the most, DigIn1 the least significant bit. (00000b = ID 0, 11111b = ID 31 (plus offset))
By Dig In 6..1 + Offset	B1100 only: The MACID is defined by DigIn6 .. 1 (X14.4, X14.16, X14.3, X14.15, X14.2 and X14.14) at power up plus the value of 6081h (MACID Parameter Value) as offset. DigIn6 is the most, DigIn1 the least significant bit. (000000b = ID 0, 111111b = ID 63 (plus offset))
Parameter Value	The MACID, when “Parameter” is selected

MACID Parameter Value

The ID, when “By Parameter” is selected as source.

Polled IO Config

These parameters define the mapping of the exchanged polled IO data. The configuration is split into the Command Configuration (the input to the Servo controller). And Response Configuration (the output of the servo controller).
The polled IO timeout value normally is configured from the master with the expected package rate, but is defaulted at startup.

DeviceNet\ Polled IO Config	
Command Config	Definition of the command data, exchanged through the polled IO telegrams. Master -> Slave
Response Config	Definition of the response data, exchanged through the polled IO telegrams. Slave -> Master
Polled IO Time Out	This parameter defines the polled data exchange timeout at startup.

Command Config

The command configuration determines the data that is sent from the PLC to LinMot servo controller. The length of the default configured command data is 20 bytes.

DeviceNet \ Polled IO Config \ Command Configuration	
Control Word	Control Word is sent from PLC. (Default Selection On)
Motion Cmd Intf	Motion Command Interface, with 3 different length: • 3 Words • 6 Words • 9 Words (Default Selection On)
RAM Parameter Channel	RAM Parameter Channel (Default Selection Off) Memory Mapping of Parameter Channel: • 1. Word UPID • 2. Word Parameter Value Low Word • 3. Word Parameter Value High word

Response Config The response configuration determines the data that is responded from the LinMot servo controller to the PLC. The length of the default configured response data is 18 bytes. Each direct variable needs 4 bytes data space in the response telegram.

DeviceNet \ Polled IO Config \ Response Configuration	
Status Word	Status Word (Default Selection On)
State Var	State Variable (Default Selection On)
Error Code	Error Code (Default Selection Off)
Warn Word	Warn Word (Default Selection On)
Echo MC Intf Header	Send back MC interface header. (Default is Off)
Monitoring Channel 1	Monitoring Channel 1 Selection (Default On)
Channel 1 UPID	Monitoring Channel 1 UPID
Monitoring Channel 2	Monitoring Channel 2 Selection (Default On)
Channel 2 UPID	Monitoring Channel 2 UPID
Monitoring Channel 3	Monitoring Channel 3 Selection (Default On)
Channel 3 UPID	Monitoring Channel 3 UPID

Slave Config The LinMot servo controller offers a UCMM Grp 3 Service opened explicit message channel. And a Group 2 Master/Slave allocable explicit message channel.

DeviceNet\ Dis-/Slave Config	
Enable Grp 3 UCMM	Group 3 UCMM service is enabled
Force Group 2 Only Server	Group 3 UCMM service is disabled (Default)

IMPORTANT: Turn on the UCMM behaviour only if needed (second master to serve at the same time). Otherwise the Group 2 only server capabilities should be enough to serve the master at start up.

6. Memory Mapping Of The Default IO Configuration

6.1. Default Configured Command Data

Below the default configured receive data memory mapping is listed. The size of the consumed data is 10 words. One motion command parameter may use two words of the motion command parameter word.

Memory Mapping Default Configured Consumed Data		
Word Offset	Name	Description
0	Control Word	Bit mapped word to control the state machine of the servo
1	Motion command Header	Defines the command to execute, split into three parts: • Main ID (8 bit) • Sub ID (4 bit) • Execution count/toggle (4bit)
2	Motion Cmd 1. Par Word	1. Word Motion command Parameter
3	Motion Cmd 2. Par Word	2. Word Motion command Parameter
4	Motion Cmd 3. Par Word	3. Word Motion command Parameter
5	Motion Cmd 4. Par Word	4. Word Motion command Parameter
6	Motion Cmd 5. Par Word	5. Word Motion command Parameter
7	Motion Cmd 6. Par Word	6. Word Motion command Parameter
8	Motion Cmd 7. Par Word	7. Word Motion command Parameter
9	Motion Cmd 8. Par Word	8. Word Motion command Parameter

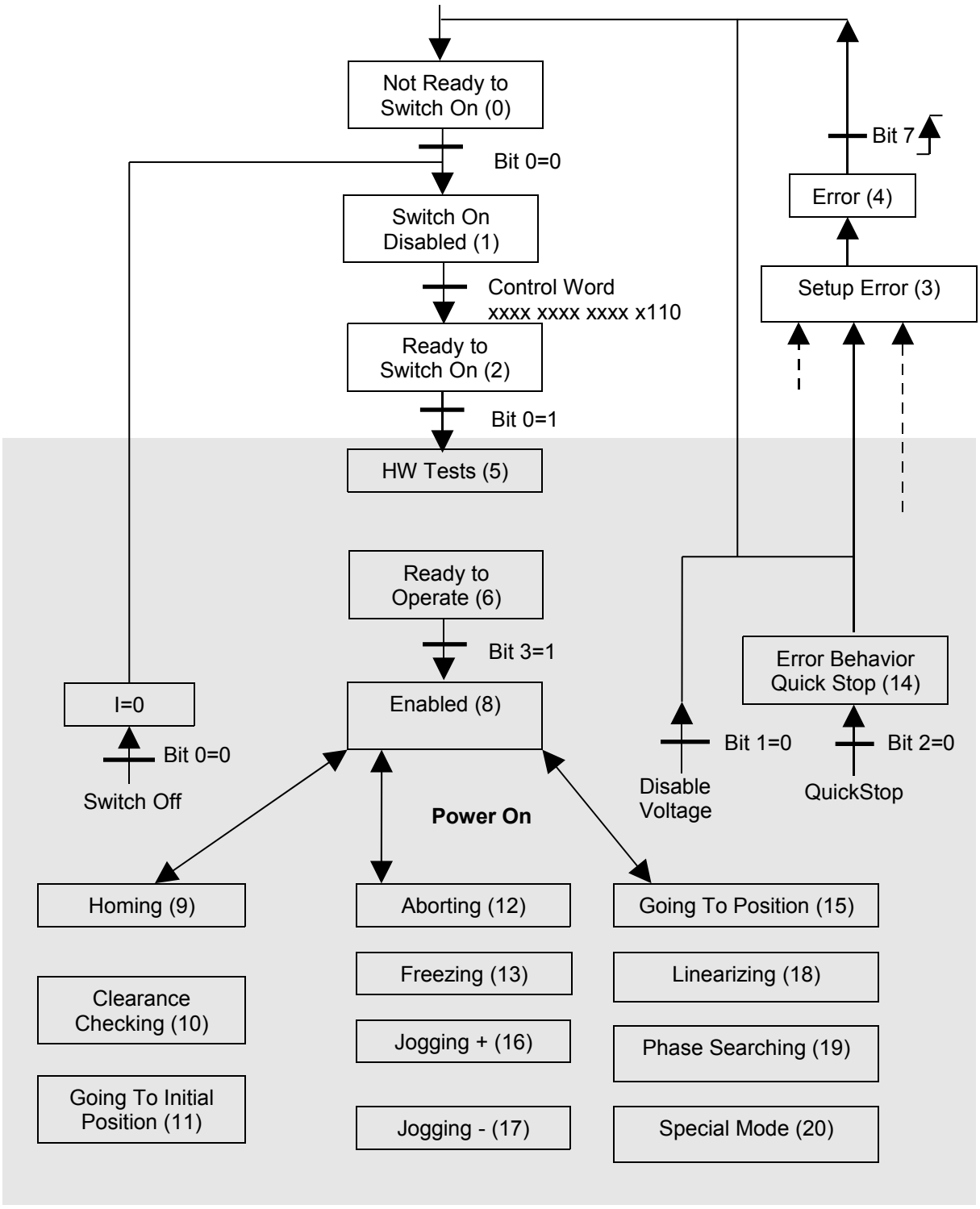
6.2. Default Configured Response Data

With the default configured response data the Servo can be supervised and monitored.

Memory Mapping Default Configured Produced Data		
Word Offset	Name	Description
0	Status Word	Bit mapped word, to monitor some important events/states.
1	State Var	Mirror of the main state machine, split into high and low byte: • Main State ID (high byte) • Sub State ID (low byte)
2	Warn Word	Bit mapped word, to monitor the warnings.
3	Monitoring Channel 1 Data Low Word	
4	Monitoring Channel 1 Data High Word	
5	Monitoring Channel 2 Data Low Word	
6	Monitoring Channel 2 Data High Word	
7	Monitoring Channel 3 Data Low Word	
8	Monitoring Channel 3 Data High Word	

7. State Machine

The main behavior of the axles is controlled with the control word, it's shown in the following state diagram.



8. Control Word

With the Control Word (16Bit) the main state machine of the servo controller can be accessed. Following table shows the meaning of each bit:

Bit Name	Val	Meaning	Remark
0 Switch On	0	OFF1	A-Stop, -> Current = 0, power switches disabled
	1	ON	State change from switch on disabled to ready to switch on
1 Voltage Enable	0	OFF2	Power switches disabled without microcontroller action
	1	Operation	
2 /Quick Stop	0	OFF3	Quick Stop -> Current = 0 -> H-Bridges disabled
	1	Operation	
3 Enable Operation	0	Operation disabled	Position controller active Motion Commands disabled
	1	Operation enable	Position controller active Motion Commands enabled
4 /Abort	0	Abort	Quick Stop position control rests active, motion command is cleared.
	1	Operation	
5 /Freeze	0	Freeze motion	Quick Stop position control rests active, Target position not cleared, curves motions are aborted
	1	Operation	Rising edge will reactivate motion command
6 Go To Position	0		
	1	Go To Position	Go to fixed parameterized Position. Wait for release of signal.
7 Error Acknowledge	0		
	1	Error Acknowledge	Rising edge of signal acknowledges error
8 Jog Move +	0		
	1		Jog Move +
9 Jog Move -	0		
	1		Jog Move -
10 Special Mode	0		
	1	Special Mode	Special Mode
11 Home	0	Stop Homing	
	1	Homing	At startup bit 11 Status word is cleared, until procedure is finished.
12 Clearance Check	0	Stop Clearance Check	
	1	Clearance Check	Enable Clearance Check Movements
13 Go To Initial Position	0		
	1	Go To initial Position	Rising edge will start go to initial position
14 Reserved	0		
	1		Reserved
15 Phase Search	0	Stop Phase Search	
	1	Phase Search	Enable Phase Search Movements

9. Status Word

Following table shows detailed meaning of the single bits:

Bit Name	Val	Meaning	Remark
0 Operation Enabled	0		State Nr < 8
	1	Operation Enabled	State Nr 8 or higher (copied to Controller EN LED)
1 Switch On Active	0	Switch On Disabled	Control Word Bit 0
	1	Switch On Enabled	
2 Enable Operation	0	Operation Disabled	Control Word Bit 3
	1	Operation	
3 Error	0	No Error	
	1	Error	Acknowledge with Control word Bit 7 (Reset Error)
4 Voltage Enable	0	Power Bridge Off	Control Word Bit 1
	1	Operation	
5 /Quick Stop	0	Active	Control Word Bit 2
	1	Operation	
6 Switch On Locked	0	Not Locked	
	1	Switch On Locked	Release with 0 of Control word bit 0 (Switch On)
7 Warning	0	Warning not active	No bit is set in the Warn Word
	1	Warning active	One or more bits in the Warn Word are set
8 Event Handler Active	0	Event Handler Inactive	Event Handler cleared or disabled
	1	Event Handler Active	Event Handler setup
9 Special Motion Active	0	Normal Operation	
	1	Special Command runs	Special motion commands (Homing, ..) runs
10 In Target Position	0	Not In Pos	Motion active or actual position out of window
	1	In Pos	Actual position after motion in window
11 Homed	0	Motor not homed	Incremental sensor not homed (referenced)
	1	Motor homed	Position sensor system valid
12 Fatal Error	0		
	1	Fatal Error	A fatal error could not be acknowledged!
13 Motion Active	0	No Motion	Setpoint generation inactive
	1	Motion active	Setpoint generation (VAI, curve) active
14 Range Indicator 1	0	Not In Range 1	Defined UPID is not in Range 1
	1	In Range1	Defined UPID is in Range 1
15 Range Indicator 2	0	Not In Range 2	Defined UPID is not in Range 2
	1	In Range2	Defined UPID is in Range 2

10. Examples IO Messages

With the following examples the first steps in programming should be explained.

10.1. Reset Control Word

Memory Mapping Default Configured Consumed Data			
Word Offset	Name	Value	Description
0	Control Word	0000h	Reset all bits in Ctrl Word
1	Motion Cmd Header	0000h	No Motion Cmd
2	Motion Cmd 1. Par Word	0000h	Not used
3	Motion Cmd 2. Par Word	0000h	Not used
4	Motion Cmd 3. Par Word	0000h	Not used
5	Motion Cmd 4. Par Word	0000h	Not used
6	Motion Cmd 5. Par Word	0000h	Not used
7	Motion Cmd 6. Par Word	0000h	Not used
8	Motion Cmd 7. Par Word	0000h	Not used
9	Motion Cmd 8. Par Word	0000h	Not used

After this command the high byte of the state variable changes to 2.

10.2. Set Control Word Switch On

Memory Mapping Default Configured Consumed Data			
Word Offset	Name	Value	Description
0	Control Word	003Fh	Set bits 0..5 in Ctrl Word
1	Motion Cmd Header	0000h	No Motion Cmd
2	Motion Cmd 1. Par Word	0000h	Not used
3	Motion Cmd 2. Par Word	0000h	Not used
4	Motion Cmd 3. Par Word	0000h	Not used
5	Motion Cmd 4. Par Word	0000h	Not used
6	Motion Cmd 5. Par Word	0000h	Not used
7	Motion Cmd 6. Par Word	0000h	Not used
8	Motion Cmd 7. Par Word	0000h	Not used
9	Motion Cmd 8. Par Word	0000h	Not used

After this command the high byte of the state variable changes to 8.

10.3. Set Control Word Home Request

Memory Mapping Default Configured Consumed Data			
Word Offset	Name	Value	Description
0	Control Word	083Fh	Set bits 0..5 and 11 in Ctrl Word
1	Motion Cmd Header	0000h	No Motion Cmd
2	Motion Cmd 1. Par Word	0000h	Not used
3	Motion Cmd 2. Par Word	0000h	Not used
4	Motion Cmd 3. Par Word	0000h	Not used
5	Motion Cmd 4. Par Word	0000h	Not used
6	Motion Cmd 5. Par Word	0000h	Not used
7	Motion Cmd 6. Par Word	0000h	Not used
8	Motion Cmd 7. Par Word	0000h	Not used
9	Motion Cmd 8. Par Word	0000h	Not used

After this command the high byte of the state variable changes to 9. Wait until bit 11 in the status word occurs, then release bit 11 in the control word again.

10.4. Reset Control Word Home Request

Memory Mapping Default Configured Consumed Data			
Word Offset	Name	Value	Description
0	Control Word	003Fh	Set bits 0..5 and reset bit 11 in Ctrl Word
1	Motion Cmd Header	0000h	No Motion Cmd
2	Motion Cmd 1. Par Word	0000h	Not used
3	Motion Cmd 2. Par Word	0000h	Not used
4	Motion Cmd 3. Par Word	0000h	Not used
5	Motion Cmd 4. Par Word	0000h	Not used
6	Motion Cmd 5. Par Word	0000h	Not used
7	Motion Cmd 6. Par Word	0000h	Not used
8	Motion Cmd 7. Par Word	0000h	Not used
9	Motion Cmd 8. Par Word	0000h	Not used

After this command the high byte of the state variable changes to 8. Now the servo controller is ready for motion commands.

10.5. Motion Command Go To Absolute Position 50mm

Memory Mapping Default Configured Consumed Data			
Word Offset	Name	Value	Description
0	Control Word	003Fh	Set bits 0..5 in Ctrl Word
1	Motion Cmd Header	0101h	VAI Go To Pos, Cmd Count = 1
2	Motion Cmd 1. Par Word	A120h	Target position (50mm) low word
3	Motion Cmd 2. Par Word	0007h	Target position (50mm) high word
4	Motion Cmd 3. Par Word	4240h	Maximal Velocity (1m/s) low word
5	Motion Cmd 4. Par Word	000Fh	Maximal Velocity (1m/s) high word
6	Motion Cmd 5. Par Word	4240h	Acceleration (10m/s^2) low word
7	Motion Cmd 6. Par Word	000Fh	Acceleration (10m/s^2) high word
8	Motion Cmd 7. Par Word	4240h	Deceleration (10m/s^2) low word
9	Motion Cmd 8. Par Word	000Fh	Deceleration (10m/s^2) high word

After this command the motor moves to the defined target position with the defined Maximal Velocity, Acceleration and Deceleration.

10.6. Motion Command Go To Absolute Position 0mm

Memory Mapping Default Configured Consumed Data			
Word Offset	Name	Value	Description
0	Control Word	003Fh	Set bits 0..5 in Ctrl Word
1	Motion Cmd Header	0102h	VAI Go To Pos, Cmd Count = 2
2	Motion Cmd 1. Par Word	0000h	Target position (0mm) low word
3	Motion Cmd 2. Par Word	0000h	Target position (0mm) high word
4	Motion Cmd 3. Par Word	4240h	Maximal Velocity (1m/s) low word
5	Motion Cmd 4. Par Word	000Fh	Maximal Velocity (1m/s) high word
6	Motion Cmd 5. Par Word	4240h	Acceleration (10m/s^2) low word
7	Motion Cmd 6. Par Word	000Fh	Acceleration (10m/s^2) high word
8	Motion Cmd 7. Par Word	4240h	Deceleration (10m/s^2) low word
9	Motion Cmd 8. Par Word	000Fh	Deceleration (10m/s^2) high word

After this command the motor moves to the new defined target position with the defined Maximal Velocity, Acceleration and Deceleration.

11.Explicit Message Services

For configuration and program handling purposes different classes and object class specific services are implemented.

11.1. Vendor specific Classes

Following LinMot specific classes are implemented for configuration and program handling reasons.

LinMot Specific Classes			
Class	E1100	B1100	Description
0040h	X	X	LinMot Parameter Class
0041h	X	-	LinMot Curve Class
0042h	X	-	LinMot Command Table Class
0043h	X	X	LinMot Program Handling Class

To simplify the access with the LinMot specific classes and services, only two message sizes of requests are implemented 6 bytes (non fragmented) and 10 bytes (fragmented). The response is always transmitted in a 8 byte long (non fragmented) message, if no data is needed, dummy data is transmitted.

11.2. LinMot Parameter Class

Within the LinMot parameter class services are supported to access the LinMot specific parameters identified by the LinMot specific UPID. The UPID is mapped to the instance ID of the explicit message service.

Services within the LinMot Parameter Class			
Service	E1100	B1100	Description
50h	X	X	Read RAM value
51h	X	X	Write RAM value
52h	X	X	Read ROM value
53h	X	X	Write ROM value
56h	X	-	Default SW instance

Read RAM Value Request

Read RAM Value Request (6 bytes)		
	Data	Description
0..63	Master ID	Master MACID (1 byte)
50h	Service ID	Read RAM value (1 byte)
0040h	Class	LinMot Parameter Class (2 bytes)
0..FFFFh	Instance	The parameter UPID is mapped to the instance (2 bytes)

Read RAM Value Response

Read RAM Value Response (8 bytes)		
	Data	Description
0..63	Master ID	Master MACID (1 byte)
D0h	Service ID	Read RAM value response (1 byte)
0000h	Trap State	LinMot Trap response: 0000h: ok 00C0h: parameter doesn't exist
0..FFFFFFFFh	Value	RAM Value of the requested parameter (4bytes)

Write RAM Value Request

Write RAM Value Request (10 bytes)		
	Data	Description
0..63	Master ID	Master MACID (1 byte)
51h	Service ID	Write RAM value (1 byte)
0040h	Class	LinMot Parameter Class (2 bytes)
0..FFFFh	Instance	The parameter UPID is mapped to the instance (2 bytes)
0..FFFFFFFFh	Value	(4 bytes)

Write RAM Value Response

Write RAM Value Response (8 bytes)		
	Data	Description
0..63	Master ID	Master MACID (1 byte)
D1h	Service ID	Write RAM value response (1 byte)
0000h	Trap State	LinMot Trap response: 0000h: ok 00C0h: parameter doesn't exist
0..FFFFFFFFh	Value	Echo set value (4 bytes)

Read ROM Value Request

Read ROM Value Request (6 bytes)		
	Data	Description
0..63	Master ID	Master MACID (1 byte)
52h	Service ID	Read ROM value (1 byte)
0040h	Class	LinMot Parameter Class (2 bytes)
0..FFFFh	Instance	The parameter UPID is mapped to the instance (2 bytes)

Read ROM Value Response

Read ROM Value Response (8 bytes)		
	Data	Description
0..63	Master ID	Master MACID (1 byte)
D2h	Service ID	Read ROM value response (1 byte)
0000h	Trap State	LinMot Trap response: 0000h: ok 00C0h: parameter doesn't exist
0..FFFFFFFFh	Value	RAM Value of the requested parameter (4bytes)

Write ROM Value Request

Write ROM Value Request (10 bytes)		
	Data	Description
0..63	Master ID	Master MACID (1 byte)
53h	Service ID	Write RAM value (1 byte)
0040h	Class	LinMot Parameter Class (2 bytes)
0..FFFFh	Instance	The parameter UPID is mapped to the instance (2 bytes)
0..FFFFFFFFh	Value	(4 bytes)

Write ROM Value Response

Write ROM Value Response (8 bytes)		
	Data	Description
0..63	Master ID	Master MACID (1 byte)
D3h	Service ID	Write RAM value response (1 byte)
0000h	Trap State	LinMot Trap response: 0000h: ok 00C0h: parameter doesn't exist
0..FFFFFFFFh	Value	Echo set value (4 bytes)

Default SW instance Request

Default SW instance Request (6 bytes)		
	Data	Description
0..63	Master ID	Master MACID (1 byte)
56h	Service ID	Default SW instance (1 byte)
0040h	Class	LinMot Parameter Class (2 bytes)
0..FFFFh	Instance	Instance Mapping: 1: default OS SW layer 2: default MC SW layer 3: default Interface SW layer 4: default Application SW layer

Default SW instance Response

Default SW instance Response (8 bytes)		
	Data	Description
0..63	Master ID	Master MACID (1 byte)
D6h	Service ID	Default SW instance response(1 byte)
0000h	Trap State	LinMot Trap response: 0000h: ok
0..FFFFFFFFh	Value	Echo of the defaulted SW layer (4bytes)

11.3. LinMot Curve Class

Within the LinMot curve class services are supported to read and write motion profiles of the servo controller. The curve ID is mapped to the instance ID of the explicit message service.

Services within the LinMot Curve Class			
Service	E1100	B1100	Description
50h	X	-	Read curve info block and data size
51h	X	-	Read curve info block
52h	X	-	Read curve data
54h	X	-	Write curve info block and data size
55h	X	-	Write curve info block
56h	X	-	Write curve data
58h	X	-	Delete single curve in RAM
59h	X	-	Delete all curves in RAM
5Ah	X	-	Write curves from RAM to FLASH

Read Curve Info Block and Data Size Request

Read Curve Info Block and Data Size Request (6 bytes)		
	Data	Description
0..63	Master ID	Master MACID (1 byte)
50h	Service ID	Read Curve Info Block and Data Size (1 byte)
0041h	Class	LinMot Curve Class (2 bytes)
0..FFFFh	Instance	The curve ID (1..99) is mapped to the instance (2 bytes)

Read Curve Info Block and Data Size Response

Read Curve Info Block and Data Size Response (8 bytes)		
	Data	Description
0..63	Master ID	Master MACID (1 byte)
D0h	Service ID	Read Curve Info Block and Data Size response (1 byte)
0000h	Trap State	LinMot Trap response: 0000h: ok
0..FFFFh	Info Block Size	Size of the curve info block
0..FFFFh	Data size	Size of the curve data block

Read Curve Info Block Request

Read Curve Info Block Request (6 bytes)		
	Data	Description
0..63	Master ID	Master MACID (1 byte)
51h	Service ID	Read Curve Info Block (1 byte)
0041h	Class	LinMot Curve Class (2 bytes)
0..FFFFh	Instance	The curve ID (1..99) is mapped to the instance (2 bytes)

This Request contains no data and could be transmitted non fragmented (6 bytes long).

Read Curve Info Block Response

Read Curve Info Block Response (8 bytes)		
	Data	Description
0..63	Master ID	Master MACID (1 byte)
D1h	Service ID	Read Curve Info Block response (1 byte)
0000h	Trap State	LinMot Trap response: 0000h: ok, end of info block reached 0004h: ok, more data follows
0..FFFFFFFFh	Data	Info block data (4bytes)

Read Curve Data Request

Read Curve Data Request (6 bytes)		
	Data	Description
0..63	Master ID	Master MACID (1 byte)
52h	Service ID	Read Curve Data Request (1 byte)
0041h	Class	LinMot Curve Class (2 bytes)
0..FFFFh	Instance	The curve ID (1..99) is mapped to the instance (2 bytes)

Read Curve Data Response

Read Curve Data Response (8 bytes)		
	Data	Description
0..63	Master ID	Master MACID (1 byte)
D2h	Service ID	Read Curve Data Response (1 byte)
0000h	Trap State	LinMot Trap response: 0000h: ok, end of data reached 0004h: ok, more data follows
0..FFFFFFFFh	Data	Curve data (4bytes) = 1 position point.

Write Curve Info Block and Data Size Request

Write Curve Info Block and Data Size Request (10 bytes)		
	Data	Description
0..63	Master ID	Master MACID (1 byte)
54h	Service ID	Read Curve Info Block and Data Size (1 byte)
0041h	Class	LinMot Curve Class (2 bytes)
0..FFFFh	Instance	The curve ID (1..99) is mapped to the instance (2 bytes)
0..FFFFh	Info Block Size	Size of the curve info block
0..FFFFh	Data size	Size of the curve data block

Write Curve Info Block and Data Size Response

Write Curve Info Block and Data Size Response (8 bytes)		
	Data	Description
0..63	Master ID	Master MACID (1 byte)
D4h	Service ID	Write Curve Info Block and Data Size response (1 byte)
0000h	Trap State	LinMot Trap response: 0000h: ok
0..FFFFh	Info Block Size	Echo of the curve info block size
0..FFFFh	Data size	Echo of the curve data block size

Write Curve Info Block Request

Write Curve Info Block Request (10 bytes)		
	Data	Description
0..63	Master ID	Master MACID (1 byte)
55h	Service ID	Write Curve Info Block (1 byte)
0041h	Class	LinMot Curve Class (2 bytes)
0..FFFFh	Instance	The curve ID (1..99) is mapped to the instance (2 bytes)
0..FFFFFFFFh	Info Block Data	Curve info block data (4bytes)

Write Curve Info Block Response

Write Curve Info Block Response (8 bytes)		
	Data	Description
0..63	Master ID	Master MACID (1 byte)
D5h	Service ID	Write Curve Info Block response (1 byte)
0000h	Trap State	LinMot Trap response: 0000h: ok, end of data reached 0004h: ok, more data follows
0..FFFFFFFFh	Info Block Data	Echo of the curve info block data (4bytes)

Write Curve Data Request

Write Curve Data Request (10 bytes)		
	Data	Description
0..63	Master ID	Master MACID (1 byte)
56h	Service ID	Write Curve Data (1 byte)
0041h	Class	LinMot Curve Class (2 bytes)
0..FFFFh	Instance	The curve ID (1..99) is mapped to the instance (2 bytes)
0..FFFFFFFFh	Info Block Data	Curve info block data (4bytes)

Write Curve Data Response

Write Curve Data Response (8 bytes)		
	Data	Description
0..63	Master ID	Master MACID (1 byte)
D6h	Service ID	Write Curve Data response (1 byte)
0000h	Trap State	LinMot Trap response: 0000h: ok, end of data reached 0004h: ok, more data follows
0..FFFFFFFFh	Curve Data	Echo of the curve data (4bytes)

Delete Single Curve in RAM Request

Delete Single Curve in RAM Request (6 bytes)		
	Data	Description
0..63	Master ID	Master MACID (1 byte)
58h	Service ID	Delete Single Curve in RAM (1 byte)
0041h	Class	LinMot Curve Class (2 bytes)
0..FFFFh	Instance	The curve ID (1..99) is mapped to the instance (2 bytes)

Delete Single Curve RAM Response

Delete Single Curve RAM Response (8 bytes)		
	Data	Description
0..63	Master ID	Master MACID (1 byte)
D8h	Service ID	Delete Single Curve in RAM response (1 byte)
0000h	Trap State	LinMot Trap response: 0000h: ok
0h	Data	Dummy data always 0 (4bytes)

Delete All Curves in RAM Request

Delete All Curves in RAM Request (8 bytes)		
	Data	Description
0..63	Master ID	Master MACID (1 byte)
59h	Service ID	Delete All Curves in RAM (1 byte)
0041h	Class	LinMot Curve Class (2 bytes)
0..FFFFh	Instance	The curve ID (1..99) is mapped to the instance (2 bytes)

Delete All Curves in RAM Response

Delete All Curves in RAM Response (8 bytes)		
	Data	Description
0..63	Master ID	Master MACID (1 byte)
D9h	Service ID	Delete All Curves in RAM response (1 byte)
0000h	Trap State	LinMot Trap response: 0000h: ok
0h	Data	Dummy data always 0 (4bytes)

Write Curves from RAM to FLASH Request

Write Curves from RAM to FLASH Request (6 bytes)

	Data	Description
0..63	Master ID	Master MACID (1 byte)
5Ah	Service ID	Write Curves from RAM to FLASH (1 byte)
0041h	Class	LinMot Curve Class (2 bytes)
0..FFFFh	Instance	0: no meaning not evaluated

Write Curves from RAM to FLASH Response

Write Curves from RAM to FLASH Response (8 bytes)		
	Data	Description
0..63	Master ID	Master MACID (1 byte)
DAh	Service ID	Write Curves from RAM to FLASH (1 byte)
0000h	Trap State	LinMot Trap response: 0000h: ok
0h	Data	Dummy data always 0 (4bytes)

11.4. LinMot Command Table Class

Within the LinMot Command Table class services are supported to read and write command table entries of the servo controller. The command table ID is mapped to the instance ID of the explicit message service.

Services within the LinMot Command Table Class			
Service	E1100	B1100	Description
50h	X	-	Command Table Get Entry List
51h	X	-	Setup Read Command Table Entry
52h	X	-	Read Command Table Entry Data
53h	X	-	Setup Write Command Table Entry
54h	X	-	Write Command Table Entry Data
55h	X	-	Delete Single Command Table Entry In RAM
56h	X	-	Delete All Command Table Entries In RAM
57h	X	-	Write Command Table Entries from RAM to FLASH

Command Table get entry list Request

Command Table get entry list Request (6 bytes)		
	Data	Description
0..63	Master ID	Master MACID (1 byte)
50h	Service ID	Command Table get entry list Request (1 byte)
0042h	Class	LinMot Command Table Class (2 bytes)
0..FFFFh	Instance	Instance mapping: 0: get entry list 0..31 1: get entry list 32..63 . 7: get entry list 224..255

Command Table get entry list Response

Command Table get entry list Response (8 bytes)		
	Data	Description
0..63	Master ID	Master MACID (1 byte)
D0h	Service ID	Command Table get entry list Response (1 byte)
0000h	Trap State	LinMot Trap response: 0000h: ok
0..FFFFFFFFh	Entry List bit field	Entry List for 32 command Table entries bit = 0 entry exists, bit = 1 entry not defined

Setup Read Command Table Entry Request

Setup Read Command Table Entry Request (6 bytes)		
	E1100	Description
0..63	Master ID	Master MACID (1 byte)
51h	Service ID	Setup Read Command Table entry Request (1 byte)
0042h	Class	LinMot Command Table Class (2 bytes)
0..FFFFh	Instance	The Command table Entry ID (1..255) is mapped to the instance (2 bytes)

Setup Read Command Table Entry Response

Setup Read Command Table Entry Response (8 bytes)		
	Data	Description
0..63	Master ID	Master MACID (1 byte)
D1h	Service ID	Setup Read Command Table entry Response (1 byte)
0000h	Trap State	LinMot Trap response: 0000h: ok
0.00000040h	Size	Command Table Entry size (40 bytes)

Read Command Table Entry Data Request

Read Command Table Entry Data Request (6 bytes)		
	Data	Description
0..63	Master ID	Master MACID (1 byte)
52h	Service ID	Read Command Table Entry Data Request (1 byte)
0042h	Class	LinMot Command Table Class (2 bytes)
0..FFFFh	Instance	The Command table Entry ID (1..255) is mapped to the instance (2 bytes)

Read Command Table Entry Data Response

Read Command Table Entry Data Response (8 bytes)		
	Data	Description
0..63	Master ID	Master MACID (1 byte)
D2h	Service ID	Read Command Table Entry Data Response (1 byte)
0000h	Trap State	LinMot Trap response: 0000h: ok, end of data reached 0004h: ok, more data follows
0..FFFFFFFFh	Data	Command Table Entry data (4bytes)

Setup Write Command Table Entry Request

Setup Write Command Table Entry Request (10 bytes)		
	Data	Description
0..63	Master ID	Master MACID (1 byte)
53h	Service ID	Setup Write Command Table Entry (1 byte)
0042h	Class	LinMot Command Table Class (2 bytes)
0..FFFFh	Instance	The Command table Entry ID (1..255) is mapped to the instance (2 bytes)
0..FFFFh	Info Block Size	Size of the curve info block
0..00000040h	Data size	Size of the Command Table entry Data 40h

Setup Write Command Table Entry Response

Setup Write Command Table Entry Response (8 bytes)		
	Data	Description
0..63	Master ID	Master MACID (1 byte)
D3h	Service ID	Setup Write Command Table Entry response (1 byte)
0000h	Trap State	LinMot Trap response: 0000h: ok
0h	Data	Dummy data always 0 (4bytes)

Write Command Table Entry Data Request

Write Command Table Entry Data Request (10 bytes)		
	Data	Description
0..63	Master ID	Master MACID (1 byte)
54h	Service ID	Write Command Table Entry Data (1 byte)
0042h	Class	LinMot Command Table Class (2 bytes)
0..FFFFh	Instance	The Command table Entry ID (1..255) is mapped to the instance (2 bytes)
0..FFFFFFFFh	Data	Command Table Entry data (4bytes)

Write Command Table Entry Data Response

Write Command Table Entry Data Response (8 bytes)		
	Data	Description
0..63	Master ID	Master MACID (1 byte)
D4h	Service ID	Write Command Table Entry Data response (1 byte)
0000h	Trap State	LinMot Trap response: 0000h: ok, end of data reached 0004h: ok, more data follows
0h	Data	Dummy data always 0 (4bytes)

Delete Single Command Table Entry In RAM Request

Delete Single Command Table Entry In RAM Request (6 bytes)		
	Data	Description
0..63	Master ID	Master MACID (1 byte)
55h	Service ID	Delete Single Command Table Entry In RAM Request (1 byte)
0042h	Class	LinMot Command Table Class (2 bytes)
0..FFFFh	Instance	The Command table Entry ID (1..255) is mapped to the instance (2 bytes)

Delete Single Command Table Entry In RAM Response

Delete Single Command Table Entry In RAM Response (8 bytes)		
	Data	Description
0..63	Master ID	Master MACID (1 byte)
D5h	Service ID	Delete Single Command Table Entry In RAM Response (1 byte)
0000h	Trap State	LinMot Trap response: 0000h: ok
0h	Data	Dummy data always 0 (4bytes)

Delete All Command Table Entries In RAM Request

Delete All Curves in RAM Request (6 bytes)		
	Data	Description
0..63	Master ID	Master MACID (1 byte)
56h	Service ID	Delete All Command Table Entries In RAM RAM (1 byte)
0042h	Class	LinMot Command Table Class (2 bytes)
0..FFFFh	Instance	The Command table Entry ID (1..255) is mapped to the instance (2 bytes)

Delete All Command Table Entries In RAM Response

Delete All Curves in RAM Response (8 bytes)		
	Data	Description
0..63	Master ID	Master MACID (1 byte)
D6h	Service ID	Delete All Command Table Entries In RAM response (1 byte)
0000h	Trap State	LinMot Trap response: 0000h: ok
0h	Data	Dummy data always 0 (4bytes)

Write Command Table Entries from RAM to FLASH Request

Write Command Table Entries from RAM to FLASH Request (6 bytes)		
	Data	Description
0..63	Master ID	Master MACID (1 byte)
57h	Service ID	Write Command Table Entries from RAM to Flash (1 byte)
0042h	Class	LinMot Command Table Class (2 bytes)
0..FFFFh	Instance	0: no meaning not evaluated

Write Command Table Entries from RAM to FLASH Response

Write Command Table Entries from RAM to FLASH Response (8 bytes)		
	Data	Description
0..63	Master ID	Master MACID (1 byte)
D7h	Service ID	Write Command Table Entries from RAM to Flash Response (1 byte)
0000h	Trap State	LinMot Trap response: 0000h: ok
0h	Data	Dummy data always 0 (4bytes)

11.5. LinMot Program Handling Class

Within the LinMot program handling class services are supported, which allows the handling of different SW layers.

Services within the LinMot Program Handling Class			
Service	E1100	B1100	Description
50h	X	X	Reset Servo Controller
51h	X	X	Stop MC and Application SW layers

Reset Servo Controller Request

After configuration data has been written to ROM or FLASH areas, reset the controller to affect the changes.

Reset Servo Controller Request (6 bytes)		
	Data	Description
0..63	Master ID	Master MACID (1 byte)
50h	Service ID	Reset Servo Controller Request (1 byte)
0043h	Class	LinMot Program Handling Class (2 bytes)
0..FFFFh	Instance	0: no meaning not evaluated

Reset Servo Controller Response

Reset Servo Controller Response (8 bytes)		
	Data	Description
0..63	Master ID	Master MACID (1 byte)
D0h	Service ID	Reset Servo Controller Response (1 byte)
0000h	Trap State	LinMot Trap response: 0000h: ok
0h	Data	Dummy data always 0 (4bytes)

Stop MC and Application SW layers Request

Before accessing the FLASH areas the MC SW and Application SW has to be stopped.

Stop MC and Application SW layers Request (6 bytes)		
	Data	Description
0..63	Master ID	Master MACID (1 byte)
51h	Service ID	Stop MC and Application SW layers (1 byte)
0043h	Class	LinMot Program Handling Class (2 bytes)
0..FFFFh	Instance	0: no meaning not evaluated

Stop MC and Application SW layers Response

Stop MC and Application SW layers Response (8 bytes)		
	Data	Description
0..63	Master ID	Master MACID (1 byte)
D1h	Service ID	Stop MC and Application SW layers (1 byte)
0000h	Trap State	LinMot Trap response: 0000h: ok
0h	Data	Dummy data always 0 (4bytes)

12. Contact Addresses

SWITZERLAND

NTI AG
Haerdlistr. 15
CH-8957 Spreitenbach

Sales and Administration:	+41-(0)56-419 91 91 office@linmot.com
Tech. Support:	+41-(0)56-544 71 00 support@linmot.com
Tech. Support (Skype) :	skype:support.linmot
Fax:	+41-(0)56-419 91 92
Web:	http://www.linmot.com/

USA

LinMot, Inc.
5750 Townline Road
Elkhorn, WI 53121

Sales and Administration:	877-546-3270 262-743-2555
Tech. Support:	877-804-0718 262-743-1284
Fax:	800-463-8708 262-723-6688
E-Mail:	us-sales@linmot.com
Web:	http://www.linmot-usa.com/

Please visit http://www.linmot.com/ to find the distribution near you.

Smart solutions are…

driven by LinMot® www.LinMot.com