WordAccApp

1. Description

The application Word Access Application (WAP) was developed to control the LinMot Servo Controller by any serial protocol, which only has to be able to access the internal RAM of the Controller in words (2 bytes). At the moment it is used together with the LinMot RSTALK and CANTalk protocol.

It's a virtual interface within the internal RAM of the LinMot Servo Controller.
It runs on MT Controllers with at least hardware revision 2.

2. Commands

The WAP supports in the actual version the following commands:

Command	ID (Hex)	Par 1	Par 2	Par 3	Par 4	Description
Bitset	0x0100	Address	Mask	Value	-	Copies the masked bits from value to address
Startcurve	0x020m	Curve A	Curve B	Curve C	Curve D	Runs the curves with the given number. There's one byte reserved for each motor
Goto Position	0x030m	Pos A	Pos B	Pos C	Pos D	Commands the motors to an absolute position.
Move HomePosition	0x040m	Val A	Val B	Val C	Val D	The actual position is moved to the given value. The motor will not jump.
Start ME Cam	0x050m	Curve A	Curve B	Curve C	Curve D	Runs the curves with the given number as CAM. There's one byte reserved for each motor

The „m“ within the ID means the motor mask. Bit 0 is motor A, Bit 1 is motor B, … This means, that with a single command, all motors can be commanded simultaneously.

The commands should always be used as followed:
1.) Set the parameters (Par 1 to Par 4)
2.) Set the Command ID
3.) Poll the Command ID until it is either:
0x0000 (OK),
0x00FE (DATA OUT OF RANGE) or
0x00FF (UNKNOWN COMMAND)

3. Adresses

The following table shows the used addresses:

Address	Name	Description
0xF702	Command	Command ID (with motor mask)
0xF704	CMD Par 1	Command Parameter 1
0xF706	CMD Par 2	Command Parameter 2
0xF708	CMD Par 3	Command Parameter 3
0xF70A	CMD Par 4	Command Parameter 4
0xF70C	Control	Control Word: Bit 0: reserved Bit 1: Run Request Flag Bit 2: Stop Request Flag (Emergency Stop) Bit 3: Main Init Request Flag Bit 4: Freeze Request Flag Motor A Bit 5: Freeze Request Flag Motor B Bit 6: Freeze Request Flag Motor C Bit 7: Freeze Request Flag Motor D Bit 8: reserved Bit 9: reserved Bit 10: reserved

		Bit 11: reserved Bit 12: Init Request Flag Motor A (Run State only)[1] Bit 13: Init Request Flag Motor B (Run State only) Bit 14: Init Request Flag Motor C (Run State only) Bit 15: Init Request Flag Motor D (Run State only)
0xF70E	Status	Status Wort: Bit 0: Init done Bit 1: Controller is in state RUN Bit 2: Controller is in state ERROR Bit 3: Controller is in state INIT Bit 4: Controller is in state DISABLE Bit 5: Controller is in ESTOP Bit 6: Error Pending Flag Bit 7: Warning Pending Flag Bit 8: Trigger Level A Bit 9: Trigger Level B Bit 10: Trigger Level C Bit 11: Trigger Level D Bit 12: In Position Flag of Motor A Bit 13: In Position Flag of Motor B Bit 14: In Position Flag of Motor C Bit 15: In Position Flag of Motor D
0xF712	Setpos Mot A	Direct Set Position for Motor A without handshake (0x7FFFh = Feature disabled)
0xF714	Setpos Mot B	Direct Set Position for Motor B without handshake (0x7FFFh = Feature disabled)
0xF716	Setpos Mot C	Direct Set Position for Motor C without handshake (0x7FFFh = Feature disabled)
0xF718	Setpos Mot D	Direct Set Position for Motor D without handshake (0x7FFFh = Feature disabled)
0xF71A	TrigPos Mot A	Stored position of Motor A at any rising 0 1 edge of the Trigger Input A
0xF71C	TrigPos Mot B	Stored position of Motor B at any rising 0 1 edge of the Trigger Input B
0xF71E	TrigPos Mot C	Stored position of Motor C at any rising 0 1 edge of the Trigger Input C
0xF720	TrigPos Mot D	Stored position of Motor D at any rising 0 1 edge of the Trigger Input D
0xE040	APOS Mot A	Actual position of Motor A
0xE100	APOS Mot B	Actual position of Motor B
0xE1C0	APOS Mot C	Actual position of Motor C
0xE280	APOS Mot D	Actual position of Motor D
0xE030	Max Speed Mot A	Max. Speed of Motor A
0xE0F0	Max Speed Mot B	Max. Speed of Motor B
0xE1B0	Max Speed Mot C	Max. Speed of Motor C
0xE270	Max Speed Mot D	Max. Speed of Motor D
0xE032	Max Acceleration A	Max. Acceleration of Motor A
0xE0F2	Max Acceleration B	Max. Acceleration of Motor B
0xE1B2	Max Acceleration C	Max. Acceleration of Motor C
0xE272	Max Acceleration D	Max. Acceleration of Motor D
0xE046	Max Current Mot A	Max. Current of Motor A
0xE106	Max Current Mot B	Max. Current of Motor B
0xE1C6	Max Current Mot C	Max. Current of Motor C
0xE286	Max Current Mot D	Max. Current of Motor D
0xF722	Spec. Max Current Mot A	Max. Current of Motor A (in the Run State with activated Feature “Spec Current on E-Stop”
0xF724	Spec. Max Current Mot B	Max. Current of Motor B (in the Run State with activated Feature “Spec Current on E-Stop”
0xF726	Spec. Max Current Mot C	Max. Current of Motor C (in the Run State with activated Feature “Spec Current on E-Stop”
0xF728	Spec. Max Current Mot D	Max. Current of Motor D(in the Run State with activated Feature “Spec Current on E-Stop”
0xF72A	Spec. Max Current Mot A E-Stop	Max. Current of Motor A (On E-Stop with activated Feature “Spec Current on E-Stop”
0xF72C	Spec. Max Current Mot B E-Stop	Max. Current of Motor B (On E-Stop with activated Feature “Spec Current on E-Stop”
0xF72E	Spec. Max Current Mot C E-Stop	Max. Current of Motor C (On E-Stop with activated Feature “Spec Current on E-Stop”
0xF730	Spec. Max Current Mot D E-Stop	Max. Current of Motor D (On E-Stop with activated Feature “Spec Current on E-Stop”
0xE050	Error Mot A	Error state of Motor A
0xE05E	Buffered Error Mot A	Buffered Error state of Motor A

[1] The single Init flags for each motor can only be used within the RUN state. With each rising edge the configured homing routine will be executed. The speed and acceleration can be controlled with the parameters Max. Speed and Max. Acceleration as it is done with a normal positioning commands.

0xE052	Warnings Mot A	Warnings of Motor A
0xE110	Error Mot B	Error state of Motor B
0xE11E	Buffered Error Mot B	Buffered Error state of Motor B
0xE112	Warnings Mot B	Warnings of Motor B
0xE1D0	Error Mot C	Error state of Motor C
0xE1DE	Buffered Error Mot C	Buffered Error state of Motor C
0xE1D2	Warnings Mot C	Warnings of Motor C
0xE290	Error Mot D	Error state of Motor D
0xE29E	Buffered Error Mot D	Buffered Error state of Motor D
0xE292	Warnings Mot D	Warnings of Motor D
0xE35A	Error System	Error state of the system
0xE3E0	Buffered Error System	Buffered Error state of the system
0xE35C	Warnings System	Warnings of the system
0xF738	“0” Position Mot A	“0” Position for Spec Two Point Mode (0x7FFFh = Feature inactive)
0xF73A	“1” Position Mot A	“1” Position for Spec Two Point Mode (0x7FFFh = Feature inactive)
0xF73C	“0” Position Mot B	“0” Position for Spec Two Point Mode (0x7FFFh = Feature inactive)
0xF73E	“1” Position Mot B	“1” Position for Spec Two Point Mode (0x7FFFh = Feature inactive)
0xF740	“0” Position Mot C	“0” Position for Spec Two Point Mode (0x7FFFh = Feature inactive)
0xF742	“1” Position Mot C	“1” Position for Spec Two Point Mode (0x7FFFh = Feature inactive)
0xF744	“0” Position Mot D	“0” Position for Spec Two Point Mode (0x7FFFh = Feature inactive)
0xF746	“1” Position Mot D	“1” Position for Spec Two Point Mode (0x7FFFh = Feature inactive)
0xA0198	Via Position Mot A	When crossing this position, the max_speed is set to Via Speed Mot X. This is only used when Set Pos Mot X is active. (0x7FFFh = Feature inactive)
0xA019A	Via Speed Mot A	Is used together with Via Position Mot X
0xA01B8	Via Position Mot B	When crossing this position, the max_speed is set to Via Speed Mot X. This is only used when Set Pos Mot X is active. (0x7FFFh = Feature inactive)
0xA01BA	Via Speed Mot B	Is used together with Via Position Mot X
0xA01D8	Via Position Mot C	When crossing this position, the max_speed is set to Via Speed Mot X. This is only used when Set Pos Mot X is active. (0x7FFFh = Feature inactive)
0xA01DA	Via Speed Mot C	Is used together with Via Position Mot X
0xA01F8	Via Position Mot D	When crossing this position, the max_speed is set to Via Speed Mot X. This is only used when Set Pos Mot X is active. (0x7FFFh = Feature inactive)
0xA01FA	Via Speed Mot D	Is used together with Via Position Mot X
0xA01A0	Time1 Trig to Pos Mot A	Time measured from Trigger to Measure Position (1 = 1s/control frequency, standard is 400us) for Motor A
0xA01A2	Time2 Trig to Pos Mot A	Time measured from Trigger to Measure Position (1 = 1s/control frequency, standard is 400us) for Motor A
0xA019E	Measure Pos Mot A	Measure position for “Time Trig to Pos” feature Motor A
0xA01C0	Time1 Trig to Pos Mot B	Time measured from Trigger to Measure Position (1 = 1s/control frequency, standard is 400us) for Motor B
0xA01C2	Time2 Trig to Pos Mot B	Time measured from Trigger to Measure Position (1 = 1s/control frequency, standard is 400us) for Motor B
0xA01BE	Measure Pos Mot B	Measure position for “Time Trig to Pos” feature Motor B
0xA01E0	Time1 Trig to Pos Mot C	Time measured from Trigger to Measure Position (1 = 1s/control frequency, standard is 400us) for Motor C
0xA01E2	Time2 Trig to Pos Mot C	Time measured from Trigger to Measure Position (1 = 1s/control frequency, standard is 400us) for Motor C
0xA01DE	Measure Pos Mot C	Measure position for “Time Trig to Pos” feature Motor C
0xA0200	Time1 Trig to Pos Mot D	Time measured from Trigger to Measure Position (1 = 1s/control frequency, standard is 400us) for Motor D
0xA0202	Time2 Trig to Pos Mot D	Time measured from Trigger to Measure Position (1 = 1s/control frequency, standard is 400us) for Motor D
0xA01FE	Measure Pos Mot D	Measure position for “Time Trig to Pos” feature Motor D

In the following table the bit definitions of the errors and warnings can be found:

Bit definitions for the error state of the motors and the system:

Bit 0	Motor too hot calculated	The calculated temperature is too high (I^2t protection)
Bit 1	Motor too hot sensor	The measured temperature of the motor is too hot (PTC sensors)
Bit 2	Following Error	Following Error
Bit 3	Slider missing	The Hall sensors don’t have reasonable signals, because there is either no slider in the stator or the distance between magnetic strip and sensor head of the external sensor system is too big.
Bit 4	Reserved	
Bit 5	Init failt	There was an error during the INIT procedure (check position)
Bit 6	Wrong motor type	Wrong or no motor is attached
Bit 7	Curve missing	Error when starting a curve (the curve is not loaded into the Flash

		memory)
Bit 8	Reserved	
Bit 9	DCLV power too small	The DC Link Voltage is too low
Bit 10	DCLV power too high	The DC Link Voltage is too high. (Extensive breaking)
Bit 11	DCLV signal too small	The logic power supply voltage is too low
Bit 12	DCLV signal too high	The logic power supply voltage is too high
Bit 13	Electronic fault	The Heat sink of the controller is too hot or the short circuit protection of the power stage was triggered
Bit 14	Reserved	
Bit 15	Application error	Application specific error

Bit definitions for the system warnings:

Bit 0	Motor too hot calculated	The calculated temperature is high (I^2t protection)
Bit 1	Motor too hot sensor	The measured temperature of the motor is too hot (PTC sensors)
Bit 2	Following Error	Following Error
Bit 3	Slider missing	The Hall sensors don't have reasonable signals, because there is either no slider in the stator or the distance between magnetic strip and sensor head of the external sensor system is too big.
Bit 4	Reserved	
Bit 5	Reserved	
Bit 6	Init not done	The Init procedure hasn't been done yet
Bit 7	Reserved	
Bit 8	Low free capacity	Low free cpu capacity
Bit 9	DCLV power too small	The DC Link Voltage is too low
Bit 10	DCLV power too high	The DC Link Voltage is too high. (Extensive breaking)
Bit 11	DCLV signal too small	The logic power supply voltage is too low
Bit 12	DCLV signal too high	The logic power supply voltage is too high
Bit 13	Electronic too hot	The Heat sink of the controller is too hot or the short circuit protection of the power stage was triggered
Bit 14	Reserved	
Bit 15	Reserved	

Bit-Definitionen der Motor Warnungen:

Bit 0	Motor too hot calculated	The calculated temperature is high (I^2t protection)
Bit 1	Motor too hot sensor	The measured temperature of the motor is too hot (PTC sensors)
Bit 2	Following Error	Following Error
Bit 3	Slider missing	The Hall sensors don't have reasonable signals, because there is either no slider in the stator or the distance between magnetic strip and sensor head of the external sensor system is too big.
Bit 4	Reserved	
Bit 5	Reserved	
Bit 6	Init not done	The Init procedure hasn't been done yet
Bit 7	Reserved	
Bit 8	Mot not in pos range	Motor is not in PosRange (to be configured with LinTalk)
Bit 9	DCLV power too small	The DC Link Voltage is too low
Bit 10	DCLV power too high	The DC Link Voltage is too high. (Extensive breaking)
Bit 11	DCLV signal too small	The logic power supply voltage is too low
Bit 12	DCLV signal too high	The logic power supply voltage is too high
Bit 13	Electronic too hot	The Heat sink of the controller is too hot or the short circuit protection of the power stage was triggered
Bit 14	Reserved	
Bit 15	Reserved	

During runtime all parameters, which are marked as „Live“ with an „**L**“-symbol in the parameter inspector, can be changed. If the needed addresses are not listed in the above table, please contact support@linmot.com. The units can be found in the addendum to the release 1.3.9 on chapter 3.7.

4. Variable Block

To have flexible and efficiant access to different variables, there are two blocks of 32 words with an addresslist each (32 bit per word). This allows to bring the variables together so, that they can be transferred with the block read and block write commands of the used serial protocol. There should only be used even addresses. The mapping of these blocks can be changed during runtime.

Read Block:

From this block can be read up to 32 words. The mapping can be configured with “Read Block Address” memory area. For each word there is a 32 Bit address. The mapping can be changed during runtime.

Addresses:	Read-Block:	0x0A’0006h – 0x0A’0044h
	Read-Block Address:	0x0A’0046h – 0x0A’00C4h

Write Block:

Into that block there can be written up to 32 words. The mapping can be configured with the “Write Block Address” memory area:

Addresses:	Write-Block:	0x0A’00C6h – 0x0A’0104h
	Write-Block Address:	0x0A’0106h – 0x0A’0184h

5. Cyclic actual values

There are 4 cyclic PDO’s with 4 word data each available. This allows the PLC to get the actual values of the LinMot Controller without polling them. They are cyclic sent over CAN bus with a configurable cycle time.
The PDO’s have to be set up by the Talk protocol. The following addresses are relevant:

- Inhibit Time:	PDO_Wait	0x0A’0284h	(Wait time, in 400us)
-PDO1:	PDO1 Identifier:	0x0A’0268h	(11 Bit Identifier)
	PDO1 Sync:	0x0A’026Ah	(Sync raster)
	PDO1 Addresses:	0x0A’0228h-0x0A’0236h (Four 32 Bit addresses, that are mapped to the PDO)	
-PDO2:	PDO2 Identifier:	0x0A’026Eh	(11 Bit Identifier)
	PDO2 Sync:	0x0A’0270h	(Sync raster)
	PDO2 Addresses:	0x0A’0238h-0x0A’0246h (Four 32 Bit addresses, that are mapped to the PDO)	
-PDO3:	PDO3 Identifier:	0x0A’0274h	(11 Bit Identifier)
	PDO3 Sync:	0x0A’0276h	(Sync raster)
	PDO3 Addresses:	0x0A’0248h-0x0A’0256h (Four 32 Bit addresses, that are mapped to the PDO)	
-PDO4:	PDO4 Identifier:	0x0A’027Ah	(11 Bit Identifier)
	PDO4 Sync:	0x0A’027Ch	(Sync raster)
	PDO4 Addresses:	0x0A’0258h-0x0A’0266h (Four 32 Bit addresses, that are mapped to the PDO)	

The Sync raster is initialised with a value, so that the four PDO’s are sent with a burst (400 us for one PDO)
If the identifier or the Sync raster of a PDO is set to zero, it is deactivated (this is as well the start up value).
If no addresses are mapped, the data is stuffed with zero.

6. Monitoring Variables

To simplify the monitoring of a LinMot system there are the following variables available, which show the maximal set current and the maximal following error. They are reset on each transition of the state machine to “RUN”.
They are mapped to the following addresses:

- Max Current Motor A:	0x0A’0194h
- Max Pos Error Motor A:	0x0A’0196h
- Max Current Motor B:	0x0A’01B4h
- Max Pos Error Motor B:	0x0A’01B6h
- Max Current Motor C:	0x0A’01D4h
- Max Pos Error Motor C:	0x0A’01D6h
- Max Current Motor D:	0x0A’01F4h
- Max Pos Error Motor D:	0x0A’01F6h

7. Installation

The installation requires an already installed LinTalk Release 1.3.10 or higher.

a.) Start LinTalk.exe

b.) Select 'Special' -> 'Install Package'

c.) In the following dialog enter as ID „Install“. The password is “NTI” (in capital letters)

d.) In the open dialog select the file „wapxxx.ik“ (or mwapxxx.ipk, if the master encoder functionality is used) e.) Wait until the installation has finished successfully

8. Example

This example shows how a positions and curves are commanded using this software.

a.) Install the firmware according to the previous chapter.

b.) Log into the Controller and open the parameter inspector

c.) Set the parameter „Command Interface“ in the directory „System“ to application

d.) Set up the attached motors, and set the “run mode” to serial.

e.) Push the 'Start' button in the control panel.

f.) Write the following variables with CANTalk or RSTalk:

Address	Value	Description
0xF70C	0x0002	Set the RUN REQUEST flag. The controller should now be in the state RUN. Bit 1 on the Address 0xF60E (STATUS) should be set.
0xF704	Pos A	The desired Set Position for motor A
0xF706	Pos A	The desired Set Position for motor B
0xF702	0x0303	GotoPos Command for motor A and B.
0xF70E	Status	shows the state of the controller.