LinMot®

LabVIEW® LinRS Programming Example

Control of LinMot® E1100 or B1100-GP controllers over RS232

Absolute Positioning, Homing, Error Acknowledge and QuickStop

Version 0.1 SL, March 24, 2010

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. Please refer to the latest edition of our "General business terms"

NTI AG
LinMot®
Haerdlistrasse 15
CH-8957 Spreitenbach

Tel.: +41 (0)56 419 91 91
Fax: +41 (0)56 419 91 92
Email: office@LinMot.com
Homepage: www.LinMot.com

Content

Content 3
Use of the example 3
Recommended documentation 3
General 4
Hardware configuration 5
Interface 6
Test program 10
Contact 11

Use of the example

This LabVIEW project is a programming example for LinMot E1100 or B1100-GP controllers. Communication is done using the LinRS protocol over RS232.

The example is provided by LinMot free of charge. There is no demand of warranty and updates for the user. Also liability is explicit excluded for damages that may be caused by using this examples.

Controller:	E1100 / B1100-GP
Classification:	[ ] LinMot internally [x] Dissemination to customers allowed
Approval:	[x] Programming example [ ] Use in productive environments

Recommended documentation

Reading the following user manuals is needed for the understanding of the communication between the PLC and the LinMot controller. The manuals are included in LinMot-Talk1100 or can be downloaded from www.linmot.com "Download > Servo Controllers"

- user manual "Motion Control SW"
- user manual "LinRS Interface"
- user manual "LinMot-Talk1100"

General

The LinMot controllers can be connected over RS232 (LinRS protocol) to a PLC (or PC). To simplify the integration of the controller into the PLC program and to show the general control the available example is presented in this document.

The package consists of the function block „LinMotFBAbsoluteMove“, a communication block, a visualisation and the LinMot hardware configuration.

The function block „LinMotFBAbsoluteMove“ has the following functionality:

- Move to any target position with definable maximal velocity, acceleration and deceleration.
- Automatic homing
- Error acknowledge
- QuickStop

The project has been realised with LabVIEW version 8.5.

The available documentation and project is made available free of charge from LinMot.
The software is to be understood as an example program.

Hardware configuration

Configuration of the LinMot controller

The LinMot controller is configured with LinMot-Talk1100. Download from www.linmot.com
It is supposed that the motor attached to the controller is already configured with the motor wizard.

The LabVIEW is using the LinRS protocol to communicate with the controller. Therefor the LinRS interface has to be installed on the controller. Below the settings are listed that have to be done on the controller..

RS Select:
E1100: By default the RS source is selected by the S3.1 switch on the controller. Alternatively it can be set by parameter. In that case „RS Parameter Def“ (UPID 200Dh) has to be set to „RS 232 and „RS Source Select“ (UPID 200Fh) to „By Parameter“.
B1100-GP: On the B1100-GP controller the RS source has to be set with „RS Parameter Def“ (UPID 6380h) to RS 232.

Baud Rate:
E1100: By default the baud rate is selected with the rotary hex switch S1. But it can also be set with the parameter „Baud Rate Parameter Definition“ (UPID 2012h) and „Baud Rate Source Selection“ (UPID 200Eh) set to „By Parameter“.
B1100-GP: „Baud Rate Parameter Definition“ (UPID 6381h) sets the baud rate on B1100-GP controllers.

MACID:
E1100: By default the MACID is selected with the rotary hex switch S2. But it can also be set with the parameter „MACID Parameter Definition“ (UPID 2076h) and „MACID Source Select“ (UPID 206Ch) set to „By Parameter“.
B1100-GP: „MACID Parameter Definition“ (UPID 6385h) defines the MACID of the B1100-GP controller.

MC Response Configuration:
Additionally to the default mapping Monitoring Channel 2 (*E1100:* UPID 20A4h / *B1100-GP*: UPID 4416h) has to be set to „True”.
Monitoring Channel 2 is used in this example to show the actual current of the motor. Therefore the „Channel 2 UPID“ (*E1100:* UPID 20A9h / *B1100-GP*: 638Bh) value has to be set to 1B93h (*E1100: Demand Current)* or E9E7h (*B1100-GP*: *Demand Current*).

Control Word:
In order that the function block is able to activate a QuickStop in the “Control Word Parameter Force Mask” the /QuickStop“ (*E1100:* UPID 13EEh / *B1100-GP*: UPID 4382h) hast to be set to “False”.

Any other parameter:
Any other parameters of the LinRS interface should be left on default value.

Note:
The above mentioned settings of the LinRS interface are part of this example and saved as LinMot configuration file (*.lmc) and can be found in the ZIP archiv:

--> „LinMot_Beispielprogramm_RS232_AbsolutePositionierung_v0.1\LinMot_Configuration“

Interface

The LabVIEW project consists of the following components:

Main:

LinMotFBAbsoluteMove	Call an instance of LinMotFBAbsoluteMove to control the LinMot drive.
Communicationloop	Function block to communicate with the LinRS-protocol over RS232.
GUI	Visualisation to control the motor.
Queues	A Queue is a data register. It is used to communication between function block and communication loop. Saves inputs and outputs of the function block.

SubVI's:

Several functions that are called by the example.

LinMotFBAbsoluteMove

With the inputs and outputs of LinMotFBAbsoluteMove the LinMot drive is controlled and monitored. On the next page they are described in Table 1 and Table 3.

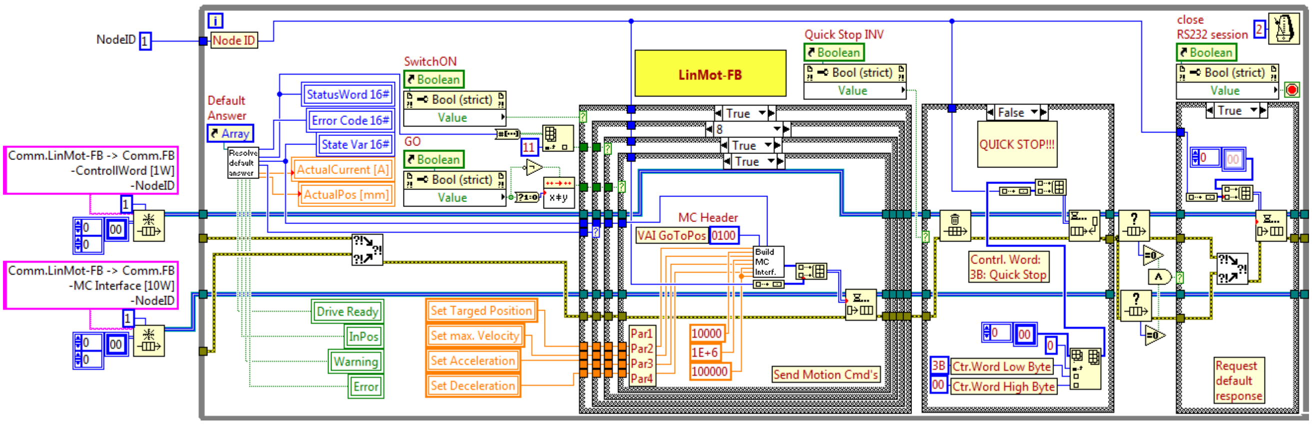


Figure 1: LinMotFBAbsoluteMove

Inputs			
Name	**Data type**	**Range**	**Description**
NodeID	U8	0..255	The NodeID is the MAC-ID of the controller.
ControllerAnswer	Array of Word	0..65535	The ControllerAnswer is the answer of the controllers to a command. Includes information about the controller / motor.
EnableDrive	Bool	0,1	Enables the drive. Is the motor not homed yet the homing is performed and afterwards switched to run mode, which is signalised with "TRUE" at output "DriveReady". Is the motor homed already it goes directly to run mode. An other function of this input is the error acknowledge. Is the input set low (False) an occurred error is acknowledged.
QuickStopINV	Bool	0,1	This inverted input activates a QuickStop if it goes low. As soon as the QuickStop is activated the motor is decelerated and stopped. To go back to run mode this input has to be set again and the EnableDrive input has to be switched off and on again. After a QuickStop the motor is current less.
Go	Bool	0,1	On a rising edge a motion with the following parameters is executed.
TargetPosition	Real	+/- 214748	Target position in mm
maxVelocity	Real	0…2147	Maximal velocity in m/s
Acceleration	Real	0…21474	Acceleration in m/s^2
Deceleration	Real	0...21474	Deceleration in m/s^2

Table 1: Inputs LinMotFBAbsoluteMove

In / Out		
Name	**Data type**	**Description**
Queue1 „ControlWord“	2D-Array of Word	Queue1 is a data register for singular communication between function block and communication loop. This Queue contains the ControlWord and the NodeID.
Queue2 „MC Interface“	2D-Array of Word	Queue1 is a data register for singular communication between function block and communication loop.. This Queue contains the Motion Command Interface (MC Interface) and the NodeID.

Table 2: InOutputs LinMotFBAbsoluteMove

Outputs		
Name	**Data type**	**Description**
DriveReady	Bool	Shows if drive is ready to operate.
InPosition	Bool	Is set if the motor is in target position. It is only set if the motor is at the position that has been set with the last evaluated command.
Warning	Bool	Active if warning is pending.
Error	Bool	Active if error is pending
ErrorCode	Byte	Error code of a pending error. (See manual „Motion Control SW“).
ActualPosition	Real	Actual position in mm
ActualCurrent	Real	Actual current in A (Ampère)

Table 3: Outputs LinMotFBAbsoluteMove

States of LinMotFBAbsoluteMove

Figure 2 shows the different states of the function block. If the controller is started/restarted the function block is in state “Switch On Disabled”. If the input “EnableDrive” is set it changes to state “Ready To Switch On”. Thereby the “QuickStopINV” input has to be set to high! Is the motor already homed, this is the case for example after an error acknowledge, it directly goes to operating state “Operation enabled”. Otherwise the homing procedure is done.

If the “QuickStopINV” input goes low a QuickStop is immediately activated. The motor is decelerated and as soon as it has stopped the motor becomes current less (Attention in case of vertical applications).

To go back to “Operation Enabled” after a QuickStop has been executed “EnableDrive” has to be set to 0 and “QuickStopINV” to 1. After that “EnableDrive” can be set back to 1 again.

If an error occurs it changes to state “Error”. The error is acknowledged by setting the “EnableDrive” input to 0. After the error output goes low the “EnableDrive” can be set to 1 to go back to “Operation Enabled”.

Attention: If the “EnableDrive” input is low the motor is current less!

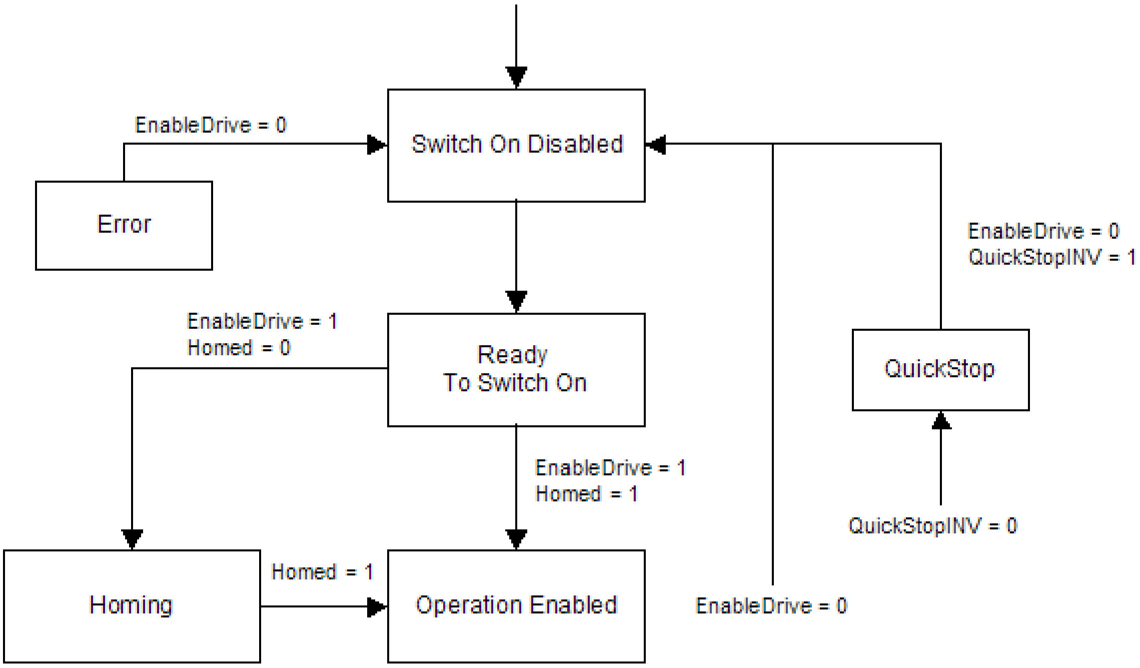


Figure 2: States of LinMotFBAbsoluteMove

Communication loop

With the inputs and outputs of the communication loop the communication with the LinMot controller over LinRS is realised. ControlWord and MC Command are sent and the status of the controller is received. Figure 3 shows the communication loop.

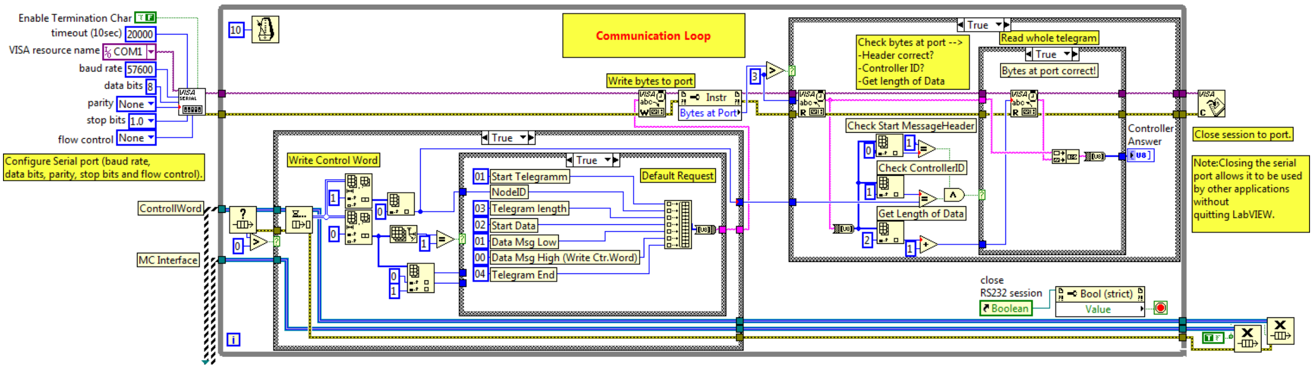


Figure 3: Communication loop

Table 4 and 5 explain the inputs and outputs.

Inputs		
Name	**Data type**	**Description**
Visa Configure Serial Port	VISA Ressource	This LabVIEW VI defines and opens the serial port. The configuration of the VI has to fit the configuration of the LinMot controller (See chapter „Hardware configuration“).
Queue1 „ControlWord“	2D-Array of Word	Queue1 is a data register for singular communication between function block and communication loop. This Queue contains the ControlWord and the NodeID.
Queue2 „MC Interface“	2D-Array of Word	Queue1 is a data register for singular communication between function block and communication loop.. This Queue contains the Motion Command Interface (MC Interface) and the NodeID.

Table 4: Inputs communication loop

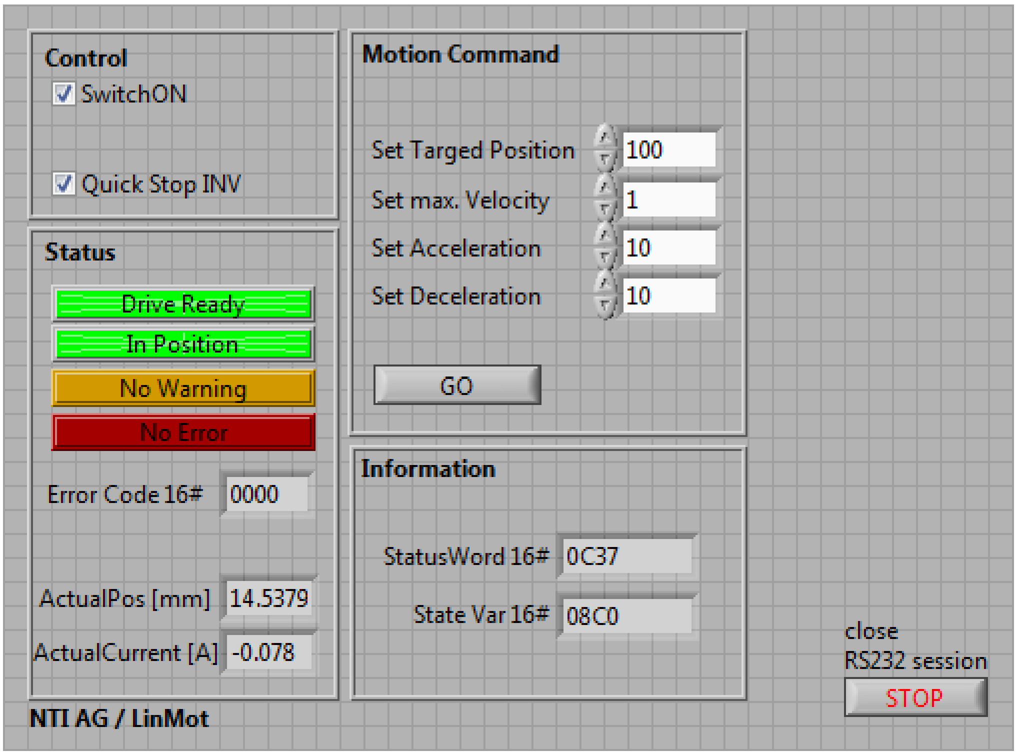
Outputs		
Name	**Data type**	**Description**
ControllerAnswer	Array of Word	The controller answers on a command with a default response that contains the StatusWord, StateVar, ActualPosition and DemandCurrent. The received response is directly forwarded to the LinMot function block

Table 5: Outputs communication loop

Test program

The function block can be controlled using the integrated visualisation.

One can move to any target position with definable maximal velocity, acceleration and deceleration.
Start motion with GO button.



☑ SwitchON
☑ Quick Stop INV

Error Code 16# 0000
ActualPos [mm] 14.5379
ActualCurrent [A] -0.078

Set Targed Position 100
Set max. Velocity 1
Set Acceleration 10
Set Deceleration 10

StatusWord 16# 0C37
State Var 16# 08C0

Abbildung 4: GUI Testprogramm

Default setting in the block diagram:

COM-Port: COM1
Baud Rate: 57600
Data Bits: 8
Parity: None
Stop Bits: 1
Flow Control: None

NodeID / MACID: 1

Contact

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