

Drive System SD2

LabVIEW Connection



P-TD-0000357.2 2016-02-22



Copyright

Translation of the original instructions, Copyright © 2016 SIEB & MEYER AG

All rights reserved.

This manual or extracts thereof may only be copied with the explicit authorization of SIEB & MEYER AG.

Trademarks

All product, font and company names mentioned in this manual may be trademarks or registered trademarks of their respective companies.

SIEB & MEYER worldwide

For questions regarding our products and technical problems please contact us.

SIEB & MEYER AG Auf dem Schmaarkamp 21 21339 Lüneburg Germany

Phone: +49 4131 203 0 Fax: +49 4131 203 2000 <u>support@sieb-meyer.de</u> http://www.sieb-meyer.de

SIEB & MEYER Asia Co. Ltd. 4 Fl, No. 532, Sec. 1 Min-Sheng N. Road Kwei-Shan Hsiang 333 Tao-Yuan Hsien Taiwan

Phone: +886 3 311 5560 Fax: +886 3 322 1224 <u>smasia@ms42.hinet.net</u> http://www.sieb-meyer.com SIEB & MEYER Shenzhen Trading Co. Ltd. Room 306, 3rd Floor, Building A1, Dongjiaotou Industrial Area, Houhai Dadao, Shekou, Nanshan District, Shenzhen City, 518067 China

Phone: +86 755 2681 1417 / +86 755 2681 2487 Fax: +86 755 2681 2967 sm.china.support@gmail.com http://www.sieb-meyer.cn

SIEB & MEYER USA 3975 Port Union Road Fairfield, OH 45014 USA

Phone: +1 513 563 0860 Fax: +1 513 563 7576 info@sieb-meyerusa.com http://www.sieb-meyer.com



Introduction	1
Preparations	2
Structure of the DLL File	3
Example	4





1	Introduction	· <u>7</u>
2	Preparations	. 9
2.1	Set Module Address	9
2.2	install drivemaster2	. 9
2.3	Set Communication Connection	. 9
2.4	Set Environment Variables	10
2.5	Set Drive Parameters	11
3	Structure of the DLL File	12
0.4		10
3.1		<u>13</u>
3.2		<u>13</u>
3.3	Read Object	<u>14</u>
3.4	Close Communication	<u>14</u>
3.5	Description of the Arguments	<u>14</u>
3.6	Communication Errors	<u>15</u>
4	Example	<u>17</u>
4.1	Graphic Display	17
4.2	Sequence	18
4.3	Used Objects	18
4.3.1	Object 101 – ICTRL_IQ_ACTUAL	. <u>19</u>
4.3.2	Object 168 – VCTRL_VELOCITY_ACTUAL_VALUE	. <u>19</u>
4.3.3	Object 398 – VCTRL_VELOCITY_ACTUAL_VALUE_UUNIT	. <u>19</u>
4.3.4	Object 68 – DEV_CTRL_CONTROL_WORD	. <u>19</u>
4.3.5	Object 210 – SPG_TARGET_VELOCITY_VL	. <u>19</u>
4.3.6	Object 395 – SPG_ TARGET_VELOCITY _VL_UUNIT	<u>20</u>
4.4	Implementation	<u>20</u>
4.5	Initial Operation	<u>23</u>

Content





Introduction

This document describes the control of the SD2 drive amplifiers by SIEB & MEYER using the graphic programming system LabVIEW by National Instruments. LabVIEW controls the drive amplifier SD2 via a DLL file.

A simple example is used to demonstrate initial operation. All examples used in this document are programmed with LabVIEW-version 6.1. You can download them from the SIEB & MEYER web page.

System requirements

The following hardware and software components are required:

- National Instruments LabVIEW version 6.1 and higher
- SIEB & MEYER software *drivemaster2* version V1.13 and higher
- PC (Windows XP or Windows 7) with RS232 interface or USB port
- SIEB & MEYER drive amplifier SD2

The examples used in this document were tested with Windows XP and Windows 7 only. They might also run on other Windows operating systems.

Introduction



2 Preparations

Before the communication between LabVIEW and the drive amplifier SD2 can be established the following preparations are required:

- 1. set module address
- 2. install drivemaster2
- 3. set communication connection
- 4. set environment variables
- 5. set drive parameters

2.1 Set Module Address

In order to simplify the following examples the drive amplifier is always addressed with the module address 0. Of course it is possible to address the drive amplifiers with different addresses or to operate several drives simultaneously – this, however, is not described in this document.

- → Set the ID switch on the front panel of the drive amplifier to 0.

For more information on addressing the drive, please refer to the documentation "drivemaster2 - User Manual" (chapter "Communication").

2.2 Install *drivemaster2*



When you use the programming system LabVIEW, do not install the software *drivemaster2* in the default path "C:\Program Files\".

For the following instructions we assume that *drivemaster2* is installed in the path "C: \SM_AG\drivemaster2". If you chose a different installation path, you must adapt the following paths accordingly.

2.3 Set Communication Connection

After a successful installation start the software *drivemaster2*.



Open the dialog for setting the communication parameters via the menu "Settings
 Setup connection to the device".

🗞 Setup conr	nection to the device
Setup communication interface	
 Local connection 	O Network connection
Select connection	
Туре U58	-
S Keep project + connect	<i>F</i> + ^C Search devices + connect
	Y Cancel

- Select the used communication channel and confirm the conection by click on one of the two "Connect" buttons.
- ✓ Now, the *drivemaster2* software establishes the connection with the drive amplifier.

For more information on the software installation and the communication, please refer to the documentation "drivemaster2 - User Manual".

2.4 Set Environment Variables

The environment variable "Path" of the Windows operating system must be extended by the additional directory "SM_Exe" from the *drivemaster2* installation.

In addition, the new system variable QT_PLUGIN_PATH must be created. For this variable the directory "SM_Exe" from the *drivemaster2* installation must be specified as well.

The required settings are described in the following using the example of Windows 7:

- Right-click on "My Computer" in the start menu and select the entry "Properties" in the corresponding context menu.
- Click the entry "Advanced System Settings" in the left column.



In the window "System Properties" select the tab page "Advanced". Click the button "Environment Variables " there.

Variable	Value
TEMP	%USERPROFILE%\AppData\Local\Temp
TMP %USERPROFILE%\AppData\Local\Temp	
	New Edit Delete
stem variables	
vstem variables Variable	Value
vstem variables Variable OS	Value Windows_NT
vstem variables Variable OS Path	Value Windows_NT C:\cygwin\bin;C:\Ruby\bin;C:\Windows
vstem variables Variable OS Path PATHEXT	Value Windows_NT C:\cygwin\bin;C:\Ruby\bin;C:\Windows .COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;

- Extend system variable "Path":
 - Select the category "Path" in the system variables and click the button "Edit".
 - Add the directory path to "SM_Exe" (C:\SM_AG\drivemaster2\SM_Exe) at the end of the variable values. Use a semicolon as separator.



Be careful not to delete or change any of the existing variable values.

- Click the button "OK" to apply the changes.
- - Click the button "New" below the system variables.
 - Enter the name "QT_PLUGIN_PATH" for the variable.
 - Enter the directory path to "SM_Exe" (C:\SM_AG\drivemaster2\SM_Exe) as variable value.
 - Click the button "OK" to apply the new system variable.

2.5 Set Drive Parameters

During parameterization in the software *drivemaster2* the following settings must be made for the LabVIEW connection:

- Parameters → Drive control → Control": The option "Serial interface / RS485 / USB" must be set as control channel and as setpoint channel.
- ► "Parameters → Digital signals → Digital inputs": All digital inputs must be set to "No function".



↔ When you have finished the parameterization and written the parameters to the drive, check the connection via "Diagnosis → Drive actual values".



- When the online connection is established, the connection status in the bottom right corner is displayed in green color [1]. Furthermore, the drive address "Drv: 0A Online" must be indicated there. This drive address means that the ID switch of the drive amplifier is set to 0.
- The drive signals a quick stop in the status field [2]. This message will be reset later.

The status field must not display an error. In case of an error the status field would be displayed in red color. If an error is displayed, you need to reset it at first. Then you can continue with the following steps.

✓ Now the preparations for the LabVIEW-connection are finished. The software *drivemaster2* is not required for the following steps and you can close it.



3 Structure of the DLL File

The drives are controlled via the file sd2ObjAccess1.dll. The DLL file is part of the *drivemaster2* installation and is also configured by the *drivemaster2* software.

Generally, *drivemaster2* makes the settings for the communication (local/network, serial/USB). The DLL file uses the DNC communication servers of *drivemaster2* to communicate with the drives. For this reason the software *drivemaster2* and the DLL file can communicate with a drive simultaneously, e.g. via a serial interface.

The file sd2ObjAccess1.dll is stored in the following directory:

C:\SM_AG\drivemaster2\SM_Exe.

The DLL file reads the settings from the registry of the Windows system. Path settings are read, too. Therefore, do not copy the DLL file into another directory.

The file sd2ObjAccess1.dll provides 4 access functions:

- 1. Initialize communication
- 2. Write object
- 3. Read object
- 4. Close communication

3.1 Initialize Communication

This function reads the current communication settings from *drivemaster2* (local/ network, serial/USB) and starts the corresponding DNC server for local communication.

- Function name: initComm
- Calling conventions: stdcall

Argument	Туре	Data type	Description
return type	Numeric	Signed 32-bit Integer	Error code

3.2 Write Object

►

This function writes a value into an object.

- Function name: writeObj
- Calling conventions: stdcall

Argument	Туре	Data type	Pass	Description
arg1	Numeric	Unsigned 8-bit Integer	Value	PowAddr: Address of power supply unit (07)
arg2	Numeric	Unsigned 8-bit Integer	Value	DrvAddr: Drive address (015)
arg3	Numeric	Unsigned 8-bit Integer	Value	DrvNo: Drive number (1, 2)
arg4	Numeric	Unsigned 16-bit Integer	Value	ObjIndex: Number of the object
arg5	Numeric	Unsigned 16-bit Integer	Value	SubIndex: Subindex of the object (0255)
arg6	Numeric	Unsigned 32-bit Integer	Value	Data: The value to be written into the object
arg7	Numeric	Unsigned 8-bit Integer	Value	Size: Number of valid bytes in Data (14)



Argument	Туре	Data type	Pass	Description
return type	Numeric	Signed 32-bit Integer	-	Error code

3.3 Read Object

This function reads the value from an object.

- Function name: readObj
- Calling conventions: stdcall

Argument	Туре	Data type	Pass	Description
arg1	Numeric	Unsigned 8-bit Integer	Value	PowAddr: Address of power supply unit (07)
arg2	Numeric	Unsigned 8-bit Integer	Value	DrvAddr: Drive address (015)
arg3	Numeric	Unsigned 8-bit Integer	Value	DrvNo: Drive number (1, 2)
arg4	Numeric	Unsigned 16-bit Integer	Value	ObjIndex: Number of the object
arg5	Numeric	Unsigned 16-bit Integer	Value	SubIndex: Subindex of the object (0255)
arg6	Numeric	Unsigned 32-bit Integer	Pointer to Value	Data: The value read from the object
arg7	Numeric	Unsigned 8-bit Integer	Pointer to Value	Size: Number of valid bytes in Data (14)
return type	Numeric	Signed 32-bit Integer	-	Error code

3.4 Close Communication

This function closes the communication channel to the DNC communication server. For local communication the used DNC server is closed as well.

- ► Function name: initComm
- Calling conventions: stdcall

Argument	Туре	Data type	Description
return type	Numeric	Signed 32-bit Integer	Error code

3.5 Description of the Arguments

PowAddr	Multi-axis applications often communicate via the power supply unit that supplies electricity to the drives. In this case the address of the power supply unit must be defined in PowAddr. If the drive is directly connected to the PC (serial or USB), PowAddr must be set to 0. For more information on addressing the drives, please refer to the documentation "drivemaster2 - User Manual" (chapter "Communication").
DrvAddr	The address is set via the rotary switch on the front panel of the device.
DrvNo	Double-axis drives share one address selection switch. In this case the desired drive is selected via DrvNr. For single-axis drives you must set DrvNr to 1.
ObjIndex	ObjIndex indicates the number of the object to be accessed.



SubIndex

Size

For more information on the object access, please refer the documentation "Drive System SD2 – DNC Object Access".
Objects with data fitting into a 32-bit value can be read or written with a single access. For these objects the SubIndex must be set to 0.

Objects with more than 32 bits of data are so-called arrays or strings. Arrays and strings must be split into several transmissions. Each transmission can transmit 32 bits or 4 bytes. The SubIndex indicates the number of the byte at which reading the array or string is started.

Information on the SubIndex:

- Strings have a header with a size of 1 byte. The header contains the length of the string. Therefore, the user data of the string start with SubIndex 1.
- Arrays have a header with a size of 4 bytes. The header contains the size of the array. Therefore, the array data start with SubIndex 4.
- The SubIndex passed to the file sd2ObjAccess1.dll is a 16bit integer. But the current version of the DLL file allows only a value in the range of 0...255.
- Data Depending on the object 8-bit, 16-bit or 32-bit data are transmitted here. Note: A 32-bit value is always passed. Therefore, the value must

be extended according to the number of valid bits.

Depending on the object this argument must indicate, whether 8bit, 16-bit or 32-bit data are transmitted.

Error code • 0 = no error • Lower 16 bits <> 0 = communication error (see se

- Lower 16 bits <> 0 = communication error (see <u>section 3.6</u> <u>"Communication Errors", p. 15</u>)
- Upper 16 bits <> 0 = drive error (see documentation "Drive System SD2 - Device Control", chapter "Fault Codes of the Service Data Channel")

3.6 Communication Errors

The DNC server can generate the following errors:

Error code	Error
1	Server Timeout
2	Server Send Error
3	Server Receive Error
4	Server Command Size
5	Server Response Size
6	Server Busy
7	Server Power Offline
16	Server Command Open
17	Server Command Config
18	Server Not Implemented
19	Server Access Denied
20	Server Toggle Error



Error code	Error
32	Parameter Module Address
33	Parameter Memory Address
34	Parameter Host Address
35	Parameter Illegal Command
36	Parameter Size
48	Client Host Unknown
64	Client Timeout
65	Client Send Error
66	Client Receive Error
80	Application Timeout
81	Object Dictionary Size Min
82	Object Dictionary Size Max
83	Object Dictionary Object Not Found
84	No Client
85	Object Dictionary Object Type
86	Object Dictionary Object Unit
87	Unsupported Object
100	Device Is Offline
101	Firmware Not Ready



The following LabVIEW implementation demonstrates the functionality of the file sd2ObjAccess1.dll.

4.1 Graphic Display

The example shows a simple control of the drive, i.e. the motor is started and a reference speed value can be set.

The following figure shows the screen layout of the example VI:

Datei Bearbeiten Ausführen Werkzeuge Durchsuchen Eenster Hilfe	
Operation Enable Switch On Error Reset cur Reset Speed reference 0 800 1000 1800 2000 2800 3000 3800 3FFF Stop Simulation STOPP	Speed actual Current actual speed setpoint 16383,0 12500,0 12500,0 5000,0 2500,0 0,0 79031 Time

Fig. 1: Example VI

- After starting the VI, start the drive amplifier via th button "Switch On".
- ◇ Click the button "Operation Enable" to activate the reference values.
- 今 Use the slide control "Speed Reference" to set a reference speed value.



4.2 Sequence

The example has the flowing sequence:

When the VI is started, a communication channel with the DNC server is established.

The following steps are executed one after the other in a program loop:

- Read actual current value from drive. The object 101 ICTRL_IQ_ACTUAL is read.
- Read actual speed value from drive. The object 168 VCTRL_VELOCITY_ACTUAL_VALUE is read.
- Write control word to drive.
 The object 68 DEV_CTRL_CONTROL_WORD is written.
- Write reference speed value to drive.
 The object 210 SPG_TARGET_VELOCITY_VL is written.

The program loop is stopped, when the button "Stop Simulation" was clicked.

Then the communication with the DNC server is closed and the VI is exited.



Fig. 2: VI sequence

4.3 Used Objects

The following sections provide brief descriptions for each object used in this example as well as a few objects that can be used alternatively.

For further information on the use of objects, please refer to the documentation "Drive System SD2 – DNC Object Access" or to the "Object browser" in the software *drive*-*master2*.

4.3.1 Object 101 – ICTRL_IQ_ACTUAL

Actual value of the current Format: Signed 16-bit value

Unit: 0.01 A Value range: -327.68 A ... 327.67 A

4.3.2 Object 168 – VCTRL_VELOCITY_ACTUAL_VALUE

Actual value of the speed

Format:	Signed 16-bit value
Unit:	The value 16383 equals the drive scaling or 100 % speed. In the software <i>drivemaster2</i> you set the drive scaling under "Parameters \rightarrow Motor measurement system \rightarrow Velocity scaling".
Value range:	−16384 … 16384 (equals −100 % … 100 %)

4.3.3 Object 398 – VCTRL_VELOCITY_ACTUAL_VALUE_UUNIT

Actual value of the speed in user units (physical units)

Format:	Signed 32-bit value	
Unit:	0.001 rpm for rotative drives (1000 = 1 rpm)	
Value range:	-drive scaling drive scaling	

4.3.4 Object 68 – DEV_CTRL_CONTROL_WORD

Control word of the dr	ive
Format:	Unsigned 16-bit value
Unit:	See documentation "Drive System SD2 - Device Control"
Value range:	0x0000 0xFFFF

4.3.5 Object 210 – SPG_TARGET_VELOCITY_VL

Reference value of the speed

Format:	Signed 16-bit value
Unit:	The value 16383 equals the drive scaling or 100 % speed. In the software <i>drivemaster2</i> you set the drive scaling under "Parameters \rightarrow Motor measurement system \rightarrow Velocity scaling".
Value range:	−16384 … 16384 (equals −100 % … 100 %)



4.3.6 Object 395 – SPG_ TARGET_VELOCITY _VL_UUNIT

Reference value of the speed in user units (physical units)

Format:	Signed 32-bit value	
Unit:	0.001 rpm for rotative drives (1000 = 1 rpm)	
Value range:	-drive scaling drive scaling	

4.4 Implementation

In order to integrate the file sd2ObjAccess1.dll in a LabVIEW VI you must place the LabVIEW function "Call Library Function" in the VI diagram.

The first call of the file sd2ObjAccess1.dll must start the communication with the SD2 DNC server ("Start communication" in Fig. 3).



Fig. 3: Start communication and read actual current value

When you set the parameters of the function, you define the file sd2ObjAccess1.dll as library name. The DLL file is to find in the directory SM_Exe of the *drivemaster2* installation.



tion start:			
E Call Library Function			
Library Name or Path	sd2ObjAccess1.dll	Browse	
Function Name	initComm@0	Reentrant 💌	
Calling Conventions	stdcall (WINAPI) 🛛 💌		
Parameter	return type		
Туре	Numeric 💌	Add a Parameter Before	
Data Type	Signed 32-bit Integer 💟	Add a Parameter After	
		Delete this Parameter	
Function Prototype:			
long initComm@0(void);			
		K Cancel Help	

As an example, the following figure shows the parameterization of the communica-

The second call of the DLL file in Fig. 3 reads the actual current value. For this purpose the object 101 is written. The data are saved temporarily.

Fig. 4 shows the third call of the DLL file that reads the actual speed value. For this purpose the object 168 is written. The data along with the saved data of object 101 are entered in a diagram.



Fig. 4: Read actual speed value

With the fourth call of the DLL file the control word (object 68) is written. The control word uses the bits 0, 1, 2, 3 and 7. As shown in Fig. 5 the bits 1 and 2 have the fix value '1'. This corresponds to the shutdown command of the control word (see documentation "Drive System SD2 - Device Control"). The button "Switch On" controls the



bit 0 of the control word. The button "Operation Enable" controls the bit 3 of the control word. An error reset is triggered, when the bit 7 of the control word changes from '0' to '1'.



Fig. 5: Write control word

The last call of the DLL file writes the reference speed value to the drive. For this purpose the object 210 is written. If the button "Stop Simulation" was not clicked, the program loop starts again. If "Stop Simulation" was clicked, the exitComm method of the file sd2ObjAccess1.dll starts and the communication with the SD2 DNC server is closed as shown in Fig. 6.



Fig. 6: Write reference speed value

4



4.5

Initial Operation

In order to test the application, start the software *drivemaster2* at first. When started, the software *drivemaster2* automatically establishes a connection with the drive. For this purpose the SD2 DNC server is started. The DNC server is

indicated by the symbol **1** in the taskbar of the Windows system. Double-click the symbol to open the diagnosis window of the SD2 DNC server. Depending on the used communication connection (see <u>section 2.3 "Set Communication</u> <u>Connection", p. 9</u>) either the window "S&M USB Server" or "S&M Serial Server" appears. In the diagnosis window the DNC server signals that it is connected to 1 client. This client is *drivemaster2*.

- Now start the VI in LabVIEW by click on the button "Execute". The VI starts and establishes a connection with the SD2 DNC server. Now the server signals that it is connected to 2 clients. Simultaneously the status display in the software *drivemaster2* changes from the state "Switch On Disabled, Software Funktion 'Quick Stop'" to the state "Ready To Switch On", since the VI has written the command "Shutdown" into the control word (see <u>figure 5 "Write control word"</u>, p. 21).
- Now set a reference value via the VI and allow the motor to drive the new speed. If you exit the VI via the button "Stop Simulation", the connection with the SD2 DNC server is also closed. The server signals this by resetting the number of connected clients to 1.



Fig. 7: Initial operation via LabVIEW-VI

Example

