

**Communication Driver SL2DP**

Driver for Profibus DP communication  
with CP5412-A2 Siemens board.

## Index

<b>1</b>	<b>INTRODUCTION.....</b>	<b>2</b>
<b>2</b>	<b>GENERAL CHARACTERISTICS .....</b>	<b>3</b>
2.1	DEVICE CHARACTERISTICS .....	3
2.2	LINK CHARACTERISTICS.....	3
2.3	DRIVER CHARACTERISTICS .....	3
2.4	INFORMATION ABOUT CONFORMANCE TESTING.....	4
<b>3</b>	<b>INSTALLATION .....</b>	<b>4</b>
3.1	INSTALLING THE DRIVER .....	4
3.2	OTHER SOFTWARE REQUIREMENTS .....	5
<b>4</b>	<b>DRIVER CONFIGURATION .....</b>	<b>5</b>
4.1	SETTINGS - COMMUNICATION PARAMETERS.....	6
4.2	DRIVER WORKSHEET.....	7
4.3	STATION AND HEADER CONFIGURATION.....	8
4.4	ADDRESS CONFIGURATION .....	9
4.5	DEVICE CONFIGURATION .....	10
<b>5</b>	<b>EXECUTION .....</b>	<b>11</b>
<b>6</b>	<b>TROUBLESHOOTING.....</b>	<b>12</b>
<b>7</b>	<b>APPLICATION SAMPLE.....</b>	<b>13</b>
<b>8</b>	<b>HISTORY OF VERSIONS.....</b>	<b>14</b>

# 1 Introduction

The SL2DP driver enables communication between Studio system and the CP5412-A2 board from Siemens using the Profibus DP, in accordance with the characteristics covered in this document.

This document contains 8 parts, as follow:

- Introduction: Provides an overview of the driver documentation.
  - General characteristics: Provides information necessary to identify all the required components (hardware and software) necessary to implement the communication and global characteristics about the communication.
  - Installation: Explains the procedures that must be followed to install the software and hardware required for the communication.
  - Driver configuration: Provides the required information to configure the communication driver such as the different permutations for configuration and its default values.
  - Execution: Explain the steps to test whether the driver was correctly installed and configured.
  - Troubleshooting: Supplies a list of the most common error codes for this protocol and the procedures to fix them.
  - Application Sample: Provides a sample application for testing the configuration the driver.
  - History of versions: Provides a log of all the modifications done in driver.
- ☞ Note: This document presumes that the user has read the chapter *Driver Configuration* of the Studio's Technical reference manual.

## 2 General Characteristics

### 2.1 Device Characteristics

- Manufacturer: Siemens
- Compatible Equipment
  - CP5412-A2 (Profibus DP board) + Profibus compatible PLC
- Siemens programmer software, for example: COM PROFIBUS.

☞ Note: Please refers to section 2.4 to see the Equipment used in the standard conformance tests for this driver.

### 2.2 Link Characteristics

- Device communication port: Profibus DP
- Device Runtime software: “Simatic Net (Profibus DP CP5412-A2)” + “Seting of PG/PC Interface”
- Specific PC Board: CP 5412-A2
- Adapters / Converters: None
- Cable Wiring: Siemens Simatic Net Profibus 6XV1 – 830 – 0AH10

### 2.3 Driver Characteristics

- Operating System:
  - Windows 9x
  - Windows 2000
  - Windows NT

☞ Note: Please refer to section 2.4 to see the Operating System used in the conformance tests for this driver.

The driver is composed of the following files:

- SL2DP.INI: Internal file of the driver, it should not be modified by the user.
- SL2DP.MSG: This file contains the error messages for each error code. It is an internal file of the driver, the user should not modify it.
- SL2DP.PDF: This document provides detailed documentation about the driver.
- SL2DP.DLL: This is the compiled library for the driver.

☞ Note: All the files above must to be in the subdirectory /DRV of the Studio's installation directory.

## 2.4 Information about conformance testing

- Equipment: CP 5412-A2 (board) with a PLC S7-300.

### Configuration:

PLC program: None

LDB file Configuration COM PROFIBUS):

Master:	CP5412-A2
Slave:	PLC S7-300
Input:	10 words
Output :	10 words

PG/PC Configuration:

Access Point of Application:	CP_L2_1
Module Parameter Set Used:	CP5412A2(PROFIBUS)<Active>
- Parameters	
FMS/DP Protocol:	
Activate DP	
FMS/DP-Database:	<LDB File created by COM PROFIBUS>

Studio Configuration(communication parameters):

Device:	CP_L2_1
Slaves :	3
DPN Sys Central (0-No 1-Yes):	0
DPN Role Central (0-No 1-Yes):	0

- Operating System (development): Windows NT 4.0 + Service pack 4
- Studio Version: 3.0
- Driver version: 1.00

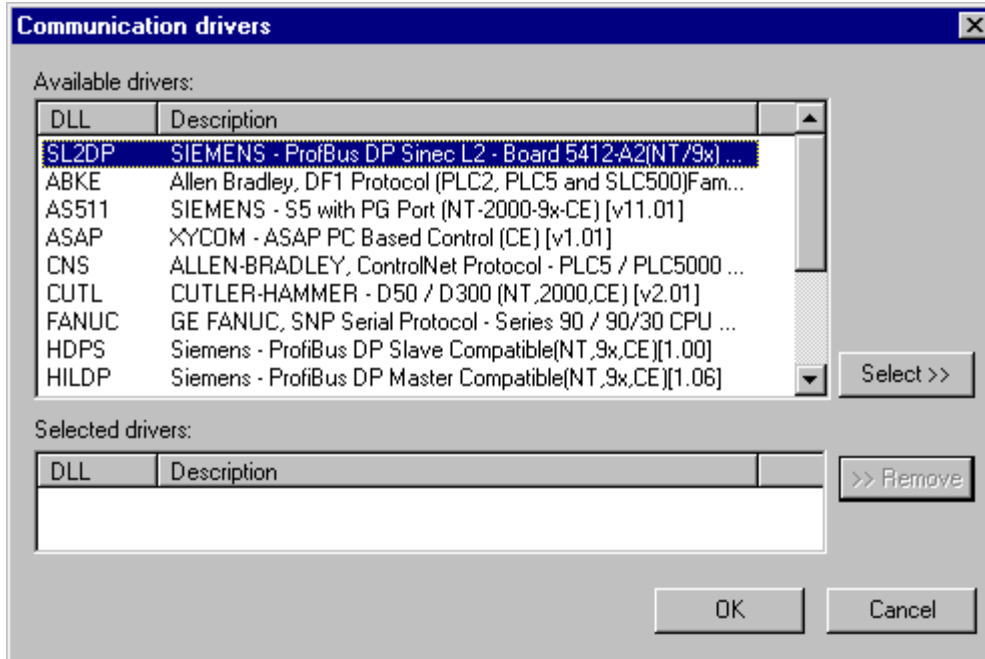
## 3 Installation

### 3.1 Installing the Driver

When you install the Studio v3.0 or higher, the communication drivers are already installed. You need now to select the driver at the applications where it will be used.

The steps to select the driver inside an application are:

1. Execute the Studio and select the proper application.
2. Select the menu *Insert + Driver...*
3. In the column **Available Drivers**, select the **SL2DP Driver** and push the button **ADD>>>** (the driver SL2DP must appear in the column **Selected Drivers**).
4. Press **OK**.



### 3.2 Other software requirements

You must install the SIMATIC NET software from Siemens. This software will install the DLLs of CP5412-A2 communication.

You will need of LDB file (configuration file). You must use the COM PROFIBUS from Siemens to create this file.

Note: Special precautions must be taken when installing the physical hardware. Refer to the hardware manufacturer documentation for specific instructions in this area.

## 4 Driver Configuration

After the driver is installed and selected in the Studio (see section 3.1), you should proceed to the driver configuration.

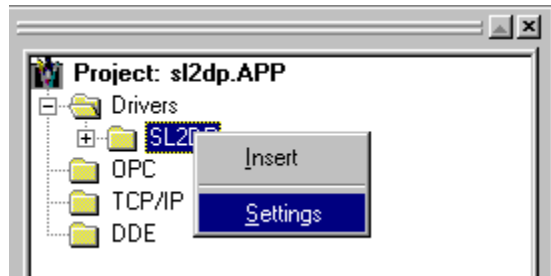
The driver configuration is two parts:

The Settings or Communication parameters, it is only one configuration to the whole driver, then you have the communication tables or Driver Worksheets, where the communication tags are defined.

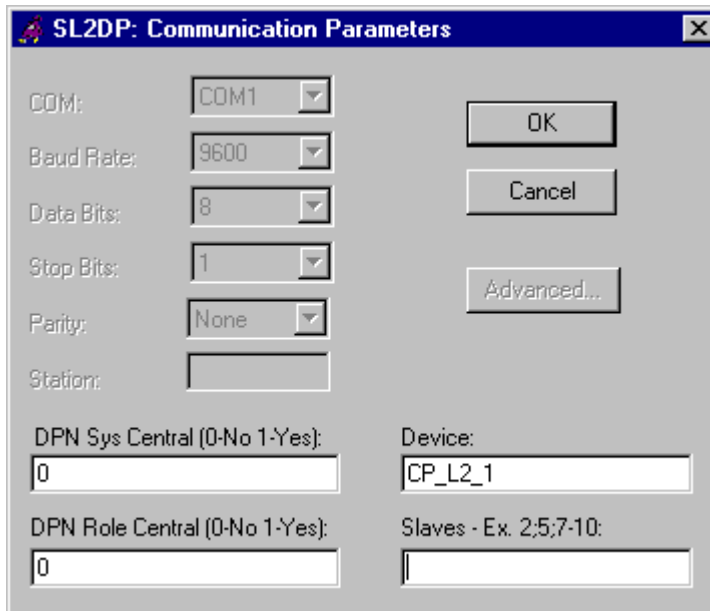
#### 4.1 Settings - Communication Parameters

These parameters are valid for all driver worksheets configured in the system. To open the window for configuring the **Communication parameters**, follow these steps:

1. In the **Workspace** of the Studio environment, select the **Comm** table.
2. Expand the folder **Drivers** and select the subfolder **SL2DP**.
3. Right click on the **SL2DP** subfolder and select the option **Settings**.



When selecting the Settings, there is the following dialog to configure:

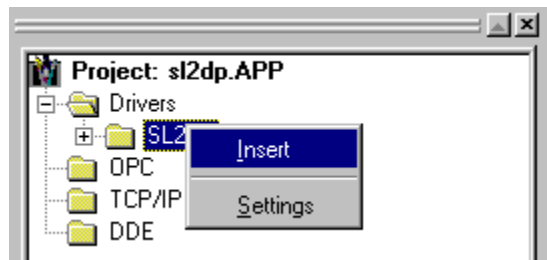


Parameter	Default Value	Valid values	Description
COM	COM2	None	Serial port of the PC used to communication with the device
Baud Rate	9600	None	Communication rate of data
Data Bits	8	None	Number of data bits used in the protocol
Stop Bits	1	None	Number of stop bits used in the protocol
Parity	None	None	Parity of the protocol
Station	0	None	Not used for the driver
Device	CP_L2_1	CP_L2_1 CP_L2_2 CP_L2_3 CP_L2_4	Name of device board. The name is the same of Setings PG/PC interface.
Slaves Ex. 2;5;7-10		0 to 255	Number of all slaves connected in the network Profibus DP.
DPN Sys Central (0-No 1-Yes)	0	0 or 1	DP application System Central.
DPN Role Central (0-No 1-Yes)	0	0 or 1	DP application Role Central.

## 4.2 Driver Worksheet

It is possible to configure many driver worksheets, each one will be composed of a Header and Body. To create a new driver worksheet, follow these steps:

1. In the **Workspace** of the Studio environment, select the table **Comm**.
2. Expand the folder **Drivers** and select the subfolder **SL2DP**.
3. Right click on the **SL2DP** subfolder and select the option **Insert**.



Note: To optimize communication and ensure better performance for the system, it is important to tie the tags in different driver sheets according to the events that must trigger the communication of each group of tags and the periodicity for which each group of tags must be written or read. In addition, it is recommended to configure the addresses of communication in sequential blocks.

When creating a communication table, you have the following window:

Description:  
  Increase read priority

Read Trigger:  Enable Read when Idle:  Read Completed:  Read Status:

Write Trigger:  Enable Write on Tag Change:  Write Completed:  Write Status:

Station:  Header:   Min:   
 Max:

	Tag Name	Address	Div	Ad
1	data[1]	0.0		
2	data[2]	0.1		
3	data[3]	0.2		
4	data[4]	0.3		

All entries at the Driver Worksheet, exception by the **Station**, **Header** and **Address** are standard to all communication drivers. You should refer to Studio Communication Driver documentation about the configuration of the standard fields. This document describes the Station, Header and Address fields, which are specific to each communication driver.

### 4.3 Station and Header configuration

Parameter	Default Value	Valid values	Description
Station	-	0 to 255	The CP5412-A2 Address.
Header	I:B:0	Vide next table	Defines the type of data to be read or written from or to the device and the reference of the initial address.

The **Header** field defines the type of variables that will be read or written from or to the device. It complies with the syntax: <Memory Type><Data Type><Initial Address Reference>. After editing the field **Header**, the system will check if it is valid or not. If the syntax were incorrect, the default value (I:B:0) will be automatically placed in this field.

You can type Tag between curly brackets into this field, but be sure that the Tag's value is correct, with the right syntax, or you will get the Invalid Header error. The right syntax, both for the field typing and Tag value is described bellow:

- **Memory Type:** Select the memory area to read or write. You can choose between I (input) , E(input), Q (output) or A (output).  
**Note:** Memory "I" is the same of "E" and "Q" is the same of "A".
- **Data Type:** Select data type of read and write. You can choose between W (word) or B (byte).  
**Note :** If you need read or writte bit data type configure it in the address field.
- **Initial address reference:** All you need to do to type a right device's variable address is to type here the initial address and on the *Address* cells (Driver Configuration's Body), just type the offset regarding this



initial address. The tables below have a lot of examples how to get the same variable's value using different headers.

Note: The bit reading and bit writing is possible only using the correct configuration in the address field.

Information regarding the parameter "Header"			
Type	Sample of syntax	Valid range of initial Address	Comment
Input	I:W:0 or E:W:0 I:B:0 or E:B:0	Configured Value in the CP5412-A2	<b>Input Memory.</b> Data received of external network. (READ ONLY) W – word data type B – Byte data type  The range of initial address reference is limited by the configuration in the CP5412-A2. The value of initial address can not be greater then configured in the CP5412-A2.
Output	Q:W:0 or A:W:0 Q:B:0 or A:B:0	Configured Value in the CP5412-A2	<b>Output Memory.</b> Data sent o external network. (READ and WRITE) W – word data type B – Byte data type  The range of initial address reference is limited by the configuration in the CP5412-A2. The value of initial address can not be greater then configured in the CP5412-A2.

#### 4.4 Address Configuration

The body of the driver worksheet allows you to associate each tag to its respective address in the device. In the column **Tag Name**, you must type the tag from your application database. This tag will receive or send values from or to an address on the device.

The address cells complies to the following syntax:

<offset>.<bit number>

*offset*: the **offset** is a number that when added to the **initial address reference** configured in the **Header** field defines an address in the device. If you're using the header W option (Word), it's the Word Offset. If you're using the B (byte) option, it's the byte offset.

To word data type the drivers limit is 127 words (255 bytes) and to byte data type the drivers limit is 255 bytes.

*Remember, the position of memory must exist in the CP5412-A2*

**Important:** if you configure an offset that reaches above the word register limit, it will be got a error status.

*bit*: The **bit** is optional and when used it specifies a bit, from 0 to 7 (bytes data type) and 0 to 15 (words data type), of the address configured.

Sample of Addressing Configuration		
Address on the Device	Header Field	Address Field
Input Area (address 0, byte access (read))	I:B:0	0
Input Area (address 0, byte access (read))	E:B:0	0
Input Area (address 0, word access (read))	I:W:0	0
Input Area (address 0, word access (read))	E:W:0	0
Input Area (address 10, byte access (read))	I:B:0	10
Input Area (address 10, byte access (read))	I:B:10	0
Input Area (address 20, word access (read))	E:W:10	10
Input Area (address 10,word access (read))	I:B:0	20(low byte)
Input Area (address 10,word access (read))	I:B:0	21(hi byte)
Input Area (address 20, byte access, bit 0 (read))	I:B:10	10.0
Input Area (address 12, byte access, bit 7 (read))	I:B:10	2.7
Input Area (address 15, word access, bit 10 (read))	I:W:15	0.10
Output Area (address 0, byte access (read/write))	Q:B:0	0
Output Area (address 0, byte access (read/write))	A:B:0	0
Output Area (address 0, word access (read/write))	QW:0	0
Output Area (address 0, word access (read/write))	A:W:0	0
Output Area (address 10, byte access (read/write))	Q:B:0	10
Output Area (address 10, byte access (read/write))	Q:B:10	0
Output Area (address 20, word access (read/write))	A:W:10	10
Output Area (address 10,word access (read/write))	Q:B:0	20(low byte)
Output Area (address 10,word access (read/write))	Q:B:0	21(hi byte)
Output Area (address 20, byte access, bit 0 (read/write))	Q:B:10	10.0
Output Area (address 12, byte access, bit 7 (read/write))	Q:B:10	2.7
Output Area (address 15, word access, bit 10 (read/write))	Q:W:15	0.10

Note: In the previous table there are several ways to set the same variable on the device because of the variable's number defined by the sum of the **initial address reference** defined in the field header and the **offset** defined in the Address Field.

#### 4.5 Device Configuration

LDB file Configuration(Created by COM PROFIBUS Siemens software):

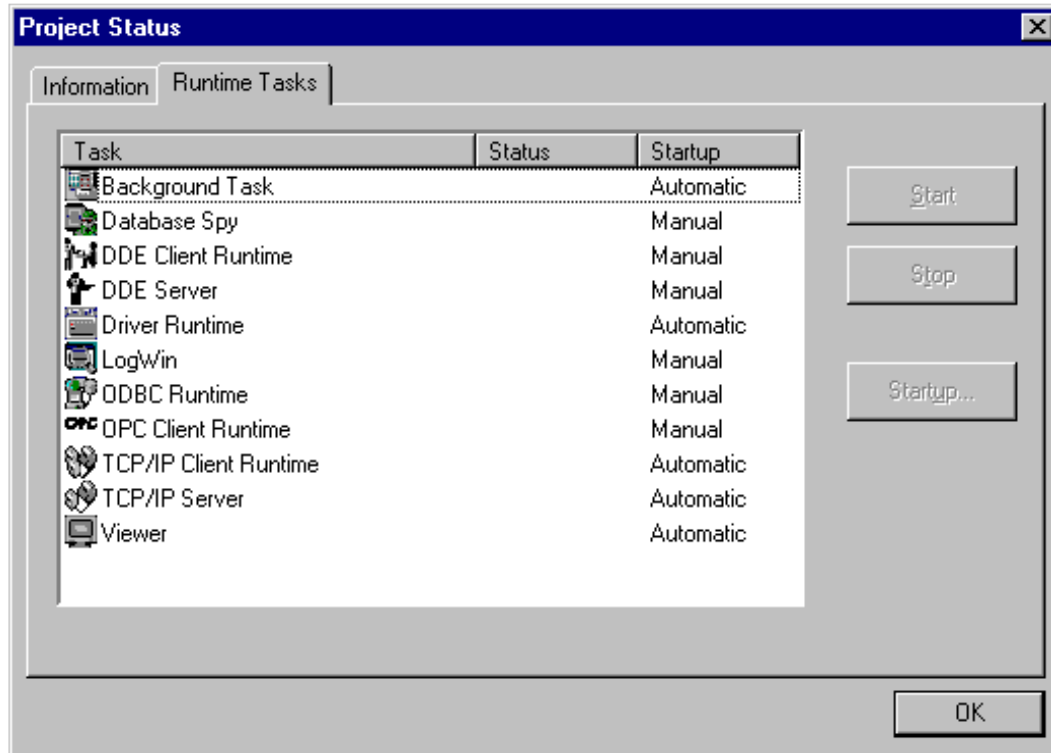
Master: CP5412-A2  
 Slave: Any Device  
 Input: Specific range data type  
 Output : Specific range data type

Settings of PG/PC interface:

Access Point of Application: CP\_L2\_1  
 Module Parameter Set Used: CP5412A2(PROFIBUS)<Active>  
 - Parameters  
 FMS/DP Protocol:  
 Activate DP  
 FMS/DP-Database: <LDB File created by COM PROFIBUS>

## 5 Execution

- When installing the driver, it is automatically selected to execute when you start-up the Runtime Environment. To verify the if the driver is correctly enabled to start, use the menu option **Project + Status...**, and verify the task Driver Runtime



## 6 Troubleshooting

After each attempt to communicate using this driver, the tag configured in the field **Read Status** or **Write Status** will receive the error code regarding the kind of failure that occurred. The error messages are:

Error Code	Description (*)	Possible causes	Procedure to solve
0	OK	Communication without problems	-
1	Invalid Header	An invalid Header has been typed or the tag that is inside this field has an invalid configuration.	Type a valid Header either on the header field or on the tag value. A lot of different valid headers are shown on the section 4.3
2	Invalid Address	An invalid Address has been typed.	Type a valid Address either on the address field. A lot of different valid Address are shown on the section 4.4
5	Block Size Error	Address offset greater then 127 to word data type or Address offset greater then 255 to byte data type.	Check on the driver configuration's worksheet that is getting this error if the configured offset s is greater then data type limit.
6	Invalid Command	An write command has been executed with INPUT memory type.	Execute only read command with INPUT memory type or change the configuration to OUTPUT memory type.
-15	Timeout waiting start a message.	<ul style="list-style-type: none"> <li>- Disconnected cables</li> <li>- PLC turned off, or in Stop or error mode</li> <li>- Wrong Station number</li> <li>- Wrong RTS/CTS control settings.</li> </ul>	<ul style="list-style-type: none"> <li>- Check the cable wiring</li> <li>- Check the PLC state. It must be RUN</li> <li>- Check the station number.</li> <li>- Check the right configuration. See on the section 2.2 the different RTS/CTS valid configurations.</li> </ul>
-17	Timeout between rx char.	<ul style="list-style-type: none"> <li>- PLC in stop or error mode</li> <li>- Wrong station number</li> <li>- Wrong parity</li> <li>- Wrong RTS/CTS configuration settings</li> </ul>	<ul style="list-style-type: none"> <li>- Check the cable wiring</li> <li>- Check the PLC state. It must be RUN</li> <li>- Check the station number.</li> <li>- Check the right configuration. See on the section 2.2 the different RTS/CTS valid configurations.</li> </ul>

Note: The results of the communication may be verified in the **output** Window of the Studio's environment. To set a log of events for **Field Read Commands**, **Field Write Commands** and **Serial Communication** click with the right button of the mouse on the output window and chose the option setting to select these log events. When testing under a Windows CE target, you can enable the log at the unit (Tools/Logwin) and verify the file celog.txt created at the target unit.

When you are not able to establish the communication with the PLC, you should first of all establish the communication between the PLC Programming Tool and the PLC. Very frequently the communication it is not possible due to a hardware or cable problem, or due an error or lack of configuration at the PLC. Only after the communication between the PLC Programming Software and the PLC is working fine, you can test again the supervisory.

When testing the communication with the Studio, you should first use the application sample described at item 7, instead of the new application that you are creating.

If is required to contact technical support, please have the following information available:

- Operating System (type and version): To find this information use the Tools/System Information option
- Project information: It is displayed using the option Project/Status from the Studio menu
- Driver version and communication log: Available from Studio Output when running the driver
- Device model and boards: please refer to hardware manufacture's documentation

## 7 Application Sample

The Studio contains a configured project to test the driver. It is strongly recommended to do some tests with this application before beginning the configuration of the customized project, for the follow reasons:

- To understand better the information covered in section 4 of this document.
- To verify that your configuration is working.
- To certify that the hardware used in the test (device + adapter + cable + PC) is in working conditions before beginning the configuration of the applications.

☞ Note: The Application Sample is not available for all drivers.

The Studio application is in the directory: **/COMMUNICATION EXAMPLES/<Driver Name>**

To perform the test, you need to follow these steps:

- Configure the device communication parameters using manufacturer programmer software..
- Open the application **/COMMUNICATION EXAMPLES/<Driver Name>**
- Execute the application
- To display the following screen with some information about the communication, please execute the Viewer module in the Studio.

☞ Note: The application for testing may be used like a maintenance screen for the custom application.

## 8 History of Versions

Version	By	Date	Description of changes
1.00	Roberto V. Junior	23-may-2000	▪ First driver version