

Communication Driver DSC

Driver for serial communication with Ordicom
DSC Reader using its RS-232 serial interface

Index

1 INTRODUCTION..... 2

2 GENERAL CHARACTERISTICS 3

2.1 DEVICE CHARACTERISTICS 3

2.2 LINK CHARACTERISTICS..... 3

2.3 DRIVER CHARACTERISTICS 5

2.4 INFORMATION ABOUT CONFORMANCE TESTING 6

3 INSTALLATION 7

3.1 INSTALLING THE DRIVER 7

3.2 OTHER SOFTWARE REQUIREMENTS 7

4 DRIVER CONFIGURATION 8

4.1 SETTINGS - COMMUNICATION PARAMETERS..... 8

4.2 STANDARD DRIVER WORKSHEET 10

4.3 STATION AND HEADER CONFIGURATION..... 11

4.4 ADDRESS CONFIGURATION 12

4.5 DEVICE CONFIGURATION 14

5 EXECUTION 14

6 TROUBLESHOOTING..... 15

7 APPLICATION SAMPLE 17

8 HISTORY OF VERSIONS..... 18

1 Introduction

The DSC driver enables communication between Studio system and some of the Ordicom devices using the proprietary protocol, in accordance with the characteristics covered in this document.

This document contains 8 parts, as follows:

- **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

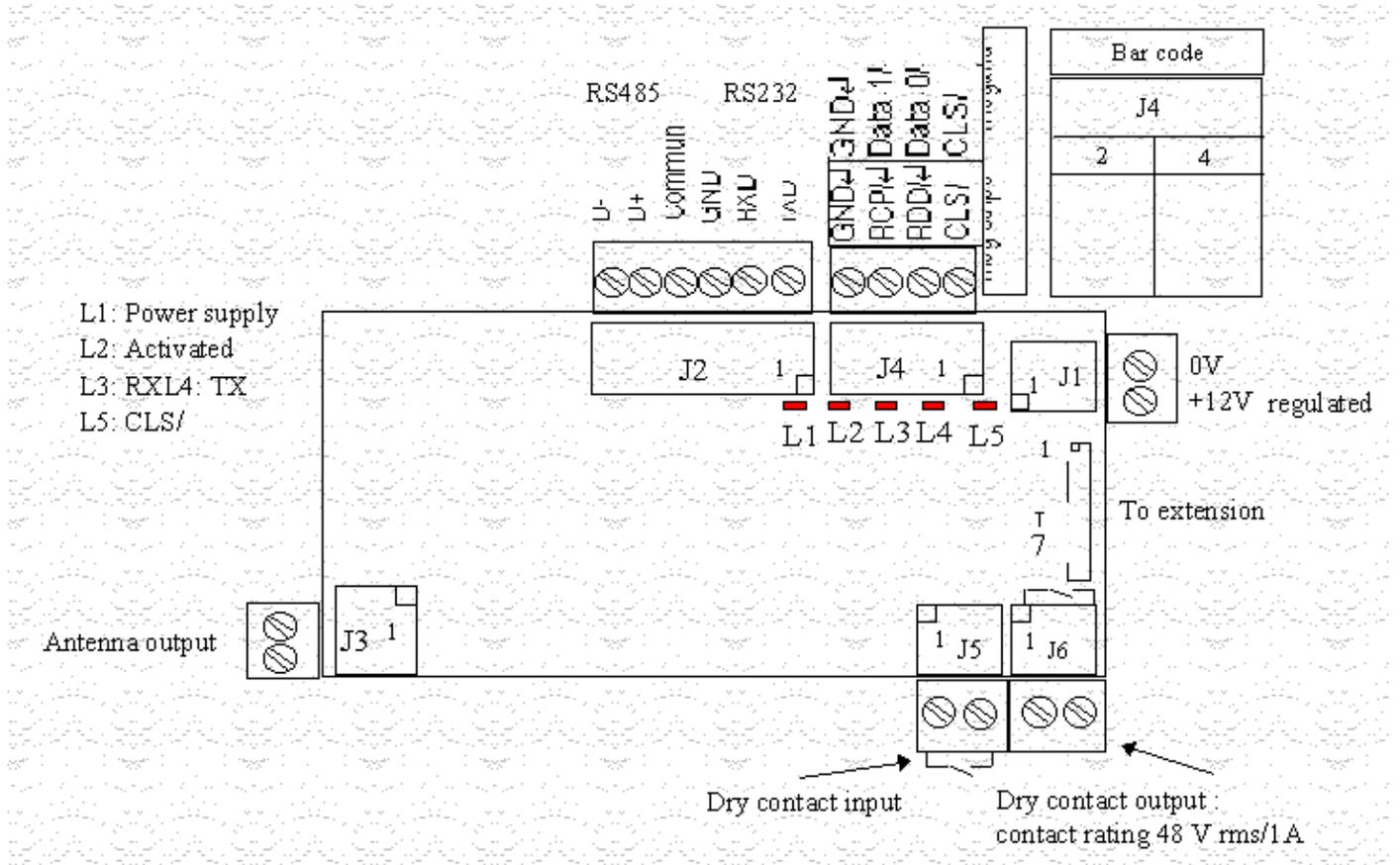
2.1 Device Characteristics

- **Manufacturer:** Ordicom R & D
- **Compatible Equipment:**
 - Reader DSC V5, V6 and V9

↳ **Tip:** 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::** Serial RS232 port
- **Physical protocol:** Serial RS232
- **Logic protocol:** Proprietary
- **Adapters/Convertes:** None.
- **Specific PC Board:** None
- **Cable:**



- Terminal block J1 for power supplies: two-pin binding-post terminal block
 - 1 -----> + 12 V regulated
 - 2 -----> 0 V
- Terminal block J2 for asynchronous links: six-pin binding-post terminal block
 - 1 -----> RS 232 TXD data output
 - 2 -----> RS 232 RXD data input
 - 3 -----> RS 232 common
 - 4 -----> RS 485 common
 - 5 -----> RS 485 D+ input/output
 - 6 -----> RS 485 D- input/output
- Terminal block J3 for the antenna: two-pin binding-post terminal block
- Terminal block J4 for the synchronous link: four-pin binding-post terminal block
- Terminal block J5 for the dry contact input : two-pin binding-post terminal block
- Terminal block J6 for the relay output : two-pin binding-post terminal block

2.3 Driver Characteristics

- **Operating System:**
 - Windows 9x
 - Windows 2000
 - Windows NT
 - Windows CE (x86 / SH3 / SH4 / MIPs / ARM / PPC)

↳ **Tip:** 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:

- **DSC.INI:** Internal file of the driver, it should not be modified by the user.
- **DSC.MSG:** Error messages for each error code. It should not be modified.
- **DSC.PDF:** Provides detailed documentation about the driver.
- **DSC.DLL:** Compiled driver.

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

- **Supported Commands:**

Register Type	Write	Read	Unsolicited	String	Integer	Float
V (Reader Version and Serial Number Read)	–	•	–	–	•	–
P (Reader Transmission Parameters Read)	–	•	–	–	•	–
T (Configuration of Reader Transmission Parameters Read)	•	–	–	–	•	–
C (Configuration of Reader Internal Timeouts)	•	–	–	–	•	–
K (Reader Self-Shutdown)	–	•	–	–	•	–
M (Reader Response Frame Shape)	•	–	–	–	•	–
B (Input/Output Management in the Sync Bus of a Reader)	•	•	–	–	•	–
Q (Module Detection Test)	–	•	–	–	•	–
S (Module Serial Number Read)	•	•	–	–	•	–

2.4 Information about conformance testing

- **Equipment:** DSC V6
- **Configuration:**
 - Baud Rate: 9600
 - Protocol: Proprietary
 - Data Bits: 7
 - Stop Bits: 1
 - Parity: Even
 - COM port: COM1
- **Cable:** According link specification, section 2.2.
- **Operating System (development):** Windows 2000 + Service Pack 1
- **Operating System (target):** Windows 2000 + Service Pack 1
- **Studio Version:** 4.1
- **Driver version:** 1.00

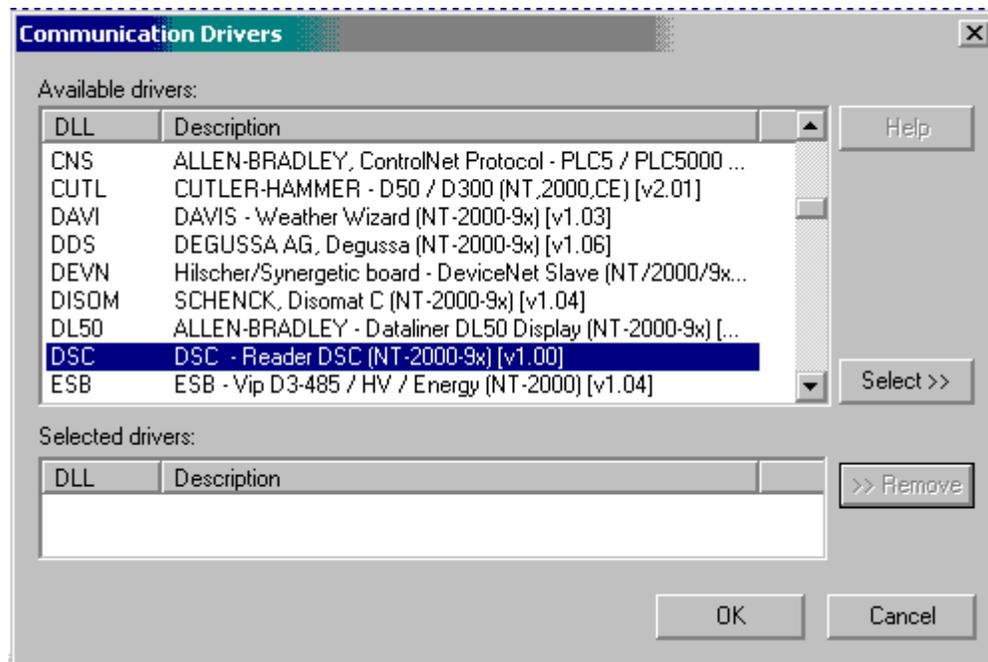
3 Installation

3.1 Installing the Driver

When you install the Studio, 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 **DSC Driver** and push the button **ADD>>>** (the driver DSC must appear in the column **Selected Drivers**).
4. Press **OK**.



3.2 Other software requirements

It is not necessary to install any other software in the PC to enable the communication between the Studio and the Device.

Attention: Special care 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.

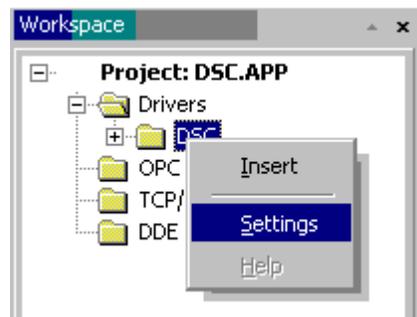
The driver configuration is two parts:

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

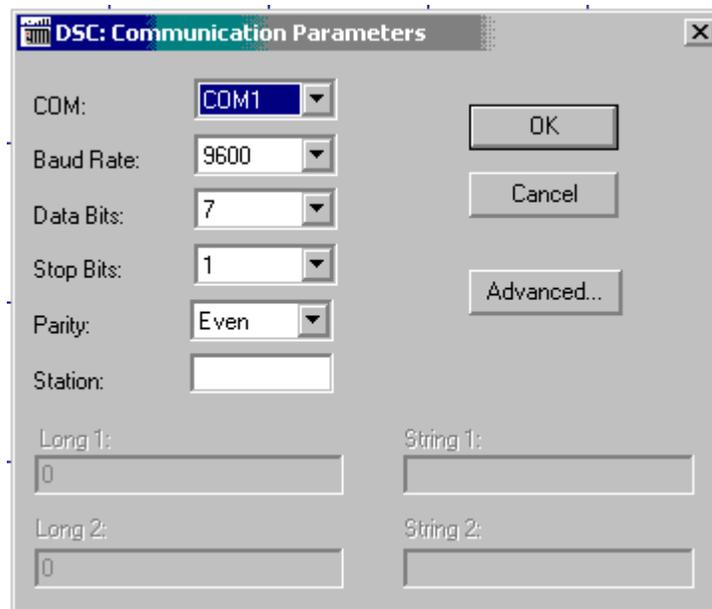
4.1 Settings - Communication Parameters

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 **DSC**.
3. Right click on the **DSC** subfolder and select the option **Settings**.



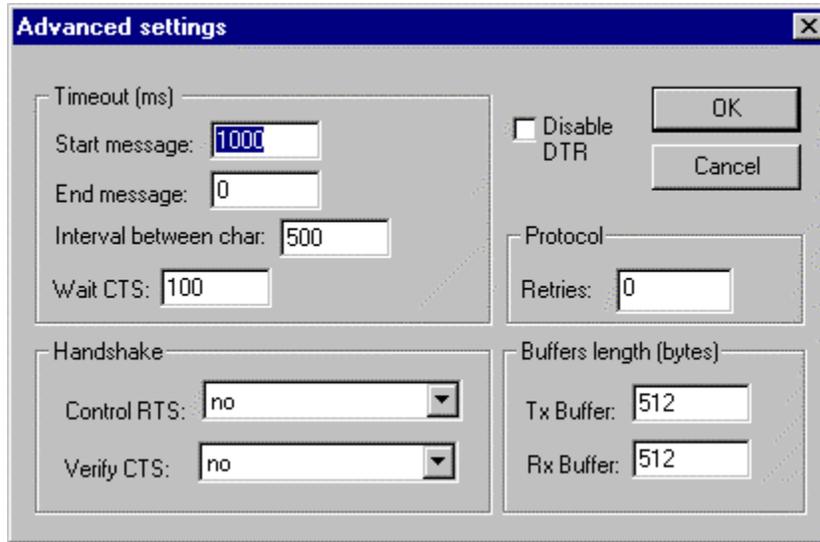
When selecting the Settings, there is the following dialog to configure:



Parameter	Default Value	Valid values	Description
Station	<Blank>	-	Not used for this driver

Note: The device **MUST** be configured with the **SAME** values defined in the **Communication Parameters** window of the DSC driver (Baud-rate=9600 ; Stop Bits=2 and Parity=Even).

By clicking on the button **Advanced...** in the windows **Communication Parameters**, you will be able to configure other serial communication parameters.



The Advanced setting parameters are explained at the Studio Technical Reference Manual, and you should keep the default values to all fields. Only the field described at the next table should be configured:

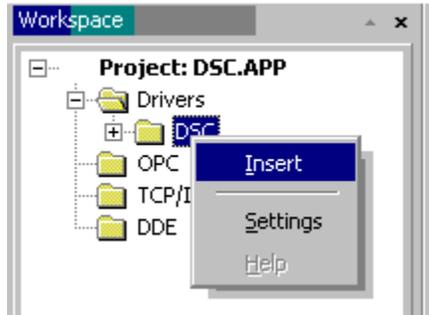
Parameter	Default Value	Valid values	Description
Control RTS	No	no, yes or yes + echo	Define if the handshake signal of RTS (Request to Send) is set before communication and if there is an echo in the communication. If we are using Windows 95 or CE with the right RS 232 – RS 485 Converter (without RTS Control), choose the option “no”. But, if we are under Windows NT and if the Cutler Hammer RS232 – 485 adapter is used, the choice must be “yes”. Important: Wrong settings on this field will not let the driver work, having the Timeout waiting start a message error.

Tip: Usually, these parameters must be changed when using a DCE (Data Communication Equipment) - converter (232/485, for example), modem, etc - between the PC and the device. It is necessary to know the characteristics of the DCE before adjusting these parameters.

4.2 Standard 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 **DSC**.
3. Right click on the **DSC** subfolder and select the option **Insert**.



Tip: 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.

Description:
 Increase 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	Add
1	nCmdPData[1]	0		
2	nCmdPData[2]	1		
3	nCmdPData[3]	2		
4	nCmdPData[4]	3		
5				
6				

All entries at the Driver Worksheet, exception by the **Station**, **Header** and **Address** are standard to all communication drivers. You should refer to Studio Technical Reference Manual 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	<None>	<Type>:<NetworkID>	Type: 1- Communication using network and Frame Validity Control using CRC-16. 2- Communication using network and Frame Validity Control using 16-bit byte by byte sum. 3- Communication without network and Frame Validity Control using CRC-16. 4- Communication without network and Frame Validity Control using 16-bit byte by byte sum. NetworkID: It is used when the TYPE 1 or 2 is used.
Header	B	See next table	Defines the command type

The **Header** parameter defines the command type. It complies with the following syntax:

<**Command**> (e.g.: B);

- **Command:** Command type: V (Reader Version and Serial Number Read), P (Reader Transmission Parameters Read), T (Configuration of Reader Transmission Parameters Read), C (Configuration of Reader Internal Timeouts), K (Reader Self-Shutdown), M (Reader Response Frame Shape), B (Input/Output Management in the Sync Bus of a Reader), Q (Module Detection Test), S (Module Serial Number Read);

After editing the **Header** field, the system will check if it is valid or not. If the syntax is incorrect, the default value will be automatically placed in this field.

You can configure a string tag name between curly brackets into this field (e.g.: {MyTag}). Make sure that the tag's value is correct, with the right syntax, or you will get the Invalid Header error. The correct syntax is described below:

Information regarding the parameter "Header"		
Type	Sample of syntax	Comment
Reader Version and Serial Number Read Command	V	Read data configured in the Address column.
Reader Transmission Parameters Read	P	Read data configured in the Address column.
Configuration of Reader Transmission Parameters Read	T	Write data configured in the Address column.
Configuration of Reader Internal Timeouts	C	Write data configured in the Address column.
Reader Self-Shutdown	K	Read data configured in the Address column.
Reader Response Frame Shape	M	Write data configured in the Address column.
Input/Output Management in the Sync Bus of a Reader	B	Read/Write data configured in the Address column.
Module Detection Test	Q	Read data configured in the Address column.
Module Serial Number Read	S	Read data configured in the Address column.

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 **Tag Name** column, 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 with the following syntax **<AddressValue>**. The table below shows the valid addresses values for each **Header** type:

Addressing Configuration				
Header	Address	Data Type	Range of Values	Comment
V	0	Integer	0 to 63	Reader version number.
	1	Integer	0 to 63	Reader sub-version number.
	2	Integer	-	Reader serial number.
P	0	Integer	0 to 7	If server does not acknowledge, the reader will make the following number of attempts to transmit a response frame.
	1	Integer	0 to 7	Response frame inter-character timeout: 0 – 10ms 1 – 20ms 2 – 50ms 3 – 100ms 4 – 200ms 5 – 500ms 6 – 1 second 7 – 2 seconds
	2	Integer	0 to 7	Acknowledge wait timeout for a response frame: 0 – 200ms 1 – 500ms 2 – 1second 3 – 2 seconds 4 – 5 seconds 5 – 10 seconds 6 – 20 seconds 7 – 50 seconds
	3	Integer	-	Reader network address.

T	0	Integer	0 to 3	Transmission rate: 0 – 1200 bauds 1 – 2400 bauds 2 – 4800 bauds 3 – 9600 bauds
	1	Integer	0 to 7	If server does not acknowledge, the reader will make the following number of attempts to transmit a response frame.
	2	Integer	0 to 7	Response frame intercharacter timeout: 0 – 10ms 1 – 20ms 2 – 50ms 3 – 100ms 4 – 200ms 5 – 500ms 6 – 1 second 7 – 2 seconds
	3	Integer	0 to 7	Acknowledge wait timeout for a response frame: 0 – 200ms 1 – 500ms 2 – 1second 3 – 2 seconds 4 – 5 seconds 5 – 10 seconds 6 – 20 seconds 7 – 50 seconds
	4	Integer	-	Reader network address.
C	0	Integer	0 to 62	Duration of Scan Cycle(200ms unit): 0 – no scan 62 – 12.4 seconds
	1	Integer	0 to 62	Duration of the reader self shutdown timeout: (1sec unit) 0 – absence of self-shutdown. 62 – 62 seconds
K	0	Integer	0 or 1	0 – Reader does not have a self-shutdown function. 1 – The command is valid and will be executed as soon as the reader receives an acknowledge from the server for this response frame.

M	0	Integer	0 to 3	Frame end character: 0 – none 1 – carriage return (default value) 2 – carriage return / line feed 3 – reserved
	1	Integer	0 or 1	Character inserts every 78 characters 0 – none (default value) 1 – frame end characters
	2	Integer	0 or 1	Reader EEPROM modification 0 – no 1 – yes
	3	Integer	0 to 3	Character inserts after each logic field of the CRP zone: 0 – none (default value) 1 – space 2 – tab 3 – reserved
B	0 to 7	Integer	-	In the Read Command the address value indicate the number of input type. In the Write Command the address value indicate the number of output type.
Q	0	Integer	0 or 1	Module detection test: 0 – no module present 1 – DSC/L module
S	0	Integer	0 to 1	Module Type: 0 – no module present 1 – DSC/L module
	1	Integer	-	Module Serial Number.

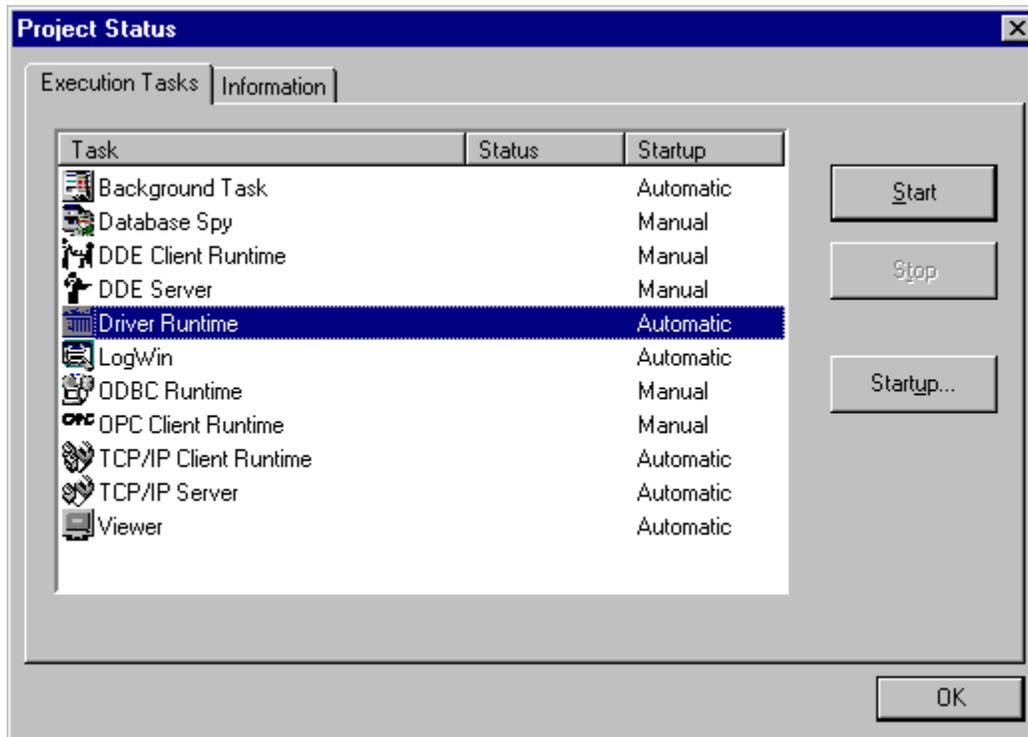
4.5 Device Configuration

The DSC Manual indicates the following serial communication settings:

- Baud Rate: 9600
- Data bits: 7
- Stop bits: 1
- Parity: Even

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	-
2	Invalid Command	The command configured in the header is only to read, write or unsolicited messages.	Verify the list of command in the 4.3 and 4.4 section and fix the operation with this command.
4	Block Size Error	Number of tags configured in the Address is minor of the required.	Verify the list of command in the 4.3 and 4.4 section and fix the number of Address configured.
5	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 4.2
6	Invalid Data	Content of tag is not correct.	Verify the list of command in the 4.3 and 4.4 section and fix the tag value.
8	Invalid Unsolicited Message	Driver sheet to receive a unsolicited message was not found.	Create a new driver sheet to receive the correct unsolicited message.
10	Timeout waiting a response.	Invalid data in the TX mssages	Verify the Address and the content of any tag value.
-15	Timeout waiting start a message.	<ul style="list-style-type: none"> - Disconnected cables - PLC turned off, or in Stop or error mode - Wrong Station number - Wrong RTS/CTS control settings. 	<ul style="list-style-type: none"> - Check the cable wiring - Check the PLC state. It must be RUN - Check the station number. - Check the right configuration. See on the section 2.2 the different RTS/CTS valid configurations.

-17	Timeout between rx char.	<ul style="list-style-type: none"> - PLC in stop or error mode - Wrong station number - Wrong parity - Wrong RTS/CTS configuration settings 	<ul style="list-style-type: none"> - Check the cable wiring - Check the PLC state. It must be RUN - Check the station number. - Check the right configuration. See on the section 2.2 the different RTS/CTS valid configurations.
-----	--------------------------	---	---

 **Tip:** The communication status can be verified by the **output** Window of the Studio's environment or by the **LogWin** module. 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, first of all establish the communication between the PLC Programming Tool and the PLC. Very frequently the communication 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 driver.

When testing the communication with the Studio, you should first use the application sample described at item 7 (if it's available), 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

Studio provides 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.

 **Tip:** 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	25-Oct-2001	▪ Initial version
1.01	Roberto V. Junior	29-Oct-2001	▪ Modified to CE Version