Communication Driver TFLUX

Driver for serial communication with Toledo scale module using its RS-422 serial interface

Index

1   Introduction

The TFLUX driver enables communication between Studio system and some of the TOLEDO devices using the proprietary TLP2 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**: TOLEDO

- **Compatible Equipment**:
  - TOLFLUX 9300

✍ **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 RS422 port

- **Physical protocol**: Serial RS422

- **Logic protocol**: TLP2

- **Adapters/Convertes**: A RS232/RS422 converter must be used to connect the PC (RS232) serial port to the TOLFLUX9300 (RS422) serial port.

- **Specific PC Board**: None

- **Cable**: According to TOLFLUX9300 and RS232/422 converter manuals.

2.3   Driver Characteristics

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

✍ **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:

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

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

- **Supported Registers:**

Register Type	Write	Read	Unsolicited	String	Integer	Float
GP (General Program)	●	●	–	●	●	–
BP (Scale Program)	●	●	–	●	●	●
CP (Container Program)	●	●	–	–	●	●
SP (Auto-Simulation Program)	●	●	–	●	●	–
TC (User Field Title Proram)	●	●	–	●	–	–
TA (Subttotalizer Title Program)	●	●	–	●	–	–
TR (Report Header Program)	●	●	–	●	–	–
SS (Subtotalizer Sequence Program)	●	●	–	●	●	–
CU (User Field Content Program)	●	●	–	●	–	–
DH (Date and Hour)	●	●	●	–	●	–
ZG (Reset General Total)	●	–	–	–	–	–
ZS (Reset Subtotalizer)	●	–	–	–	–	–
CG (General Commands)	●	–	–	●	–	–
PP (Weighing Ready)	–	–	●	–	●	–
EV (Error Messages and Events)	–	–	●	–	●	–
SG (General Status)	–	●	●	●	–	–
TT (Accumulated General Total)	–	●	–	–	●	–
ST (Accumulated Subtotal)	–	●	–	–	●	–
WT (Liquid Weight and Tare)	–	●	●	–	●	–

2.4 Information about conformance testing

- **Equipment**: TOLFLUX 9300 – 4.0

- **Configuration:**
- Baud Rate: 9600
- Protocol: TOLEDO TLP2 (Asynchronous Half-Duplex)
- Data Bits: 7
- Stop Bits: 2
- 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 **TFLUX Driver** and push the button **ADD>>>** (the driver TFLUX must appear in the column **Selected Drivers**).
4. Press **OK**.
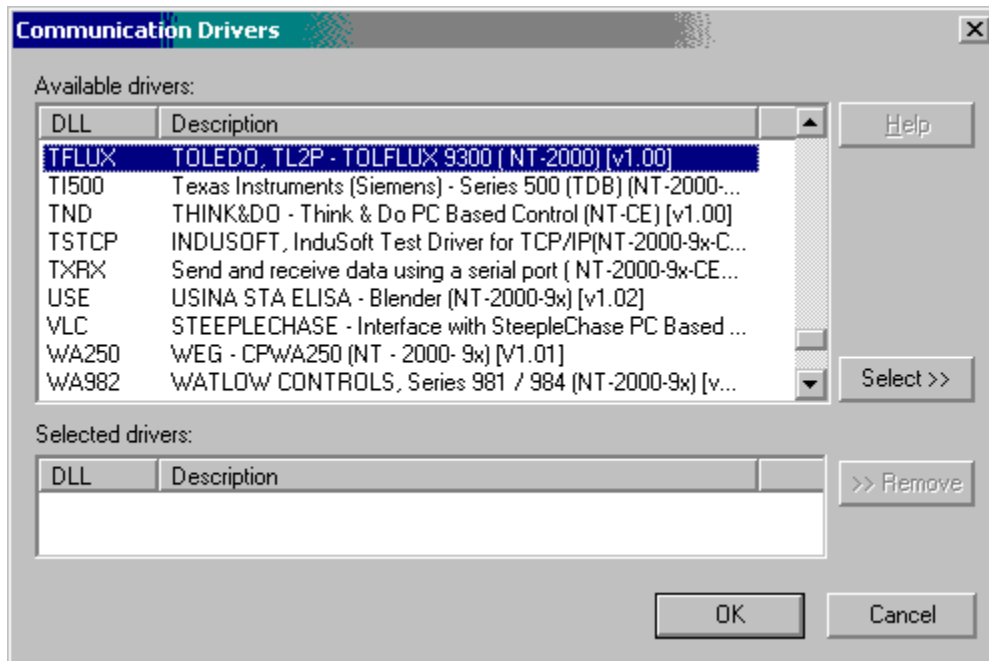
3.2   Other software requirements

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 **TFLUX**.
3. Right click on the **TFLUX** subfolder and select the option **Settings**.
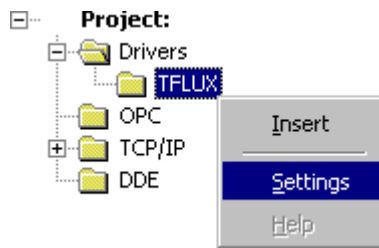
When selecting the Settings, there is the following dialog to configure:
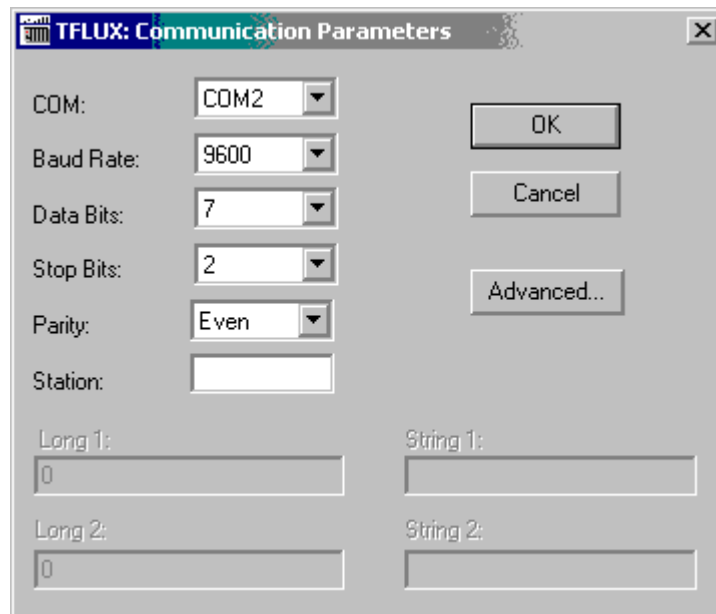
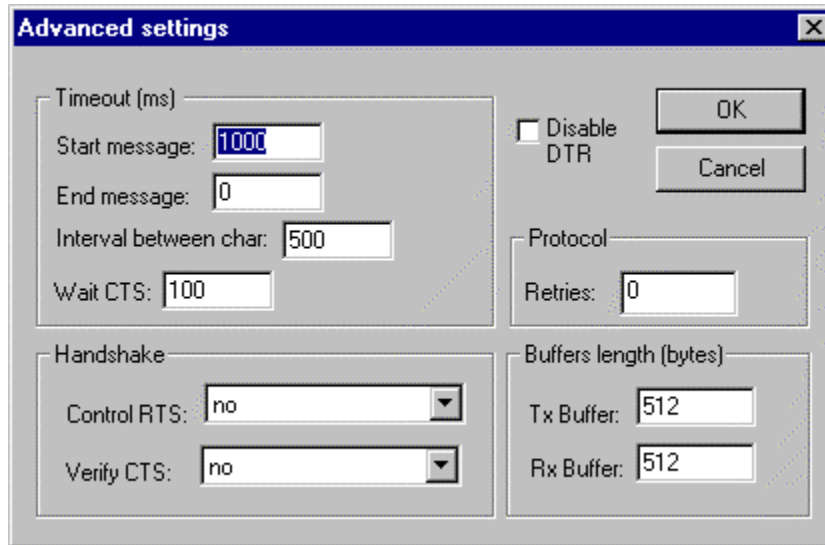| Parameter	Default Value	Valid values	Description
Station	<Blank>	-	Not used for this driver

✎ **Note:** The device MUST be configured with the SAME values defined in the **Communication Parameters** window of the TFLUX 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 **TFLUX**.
3. Right click on the **TFLUX** subfolder and select the option **Insert**.
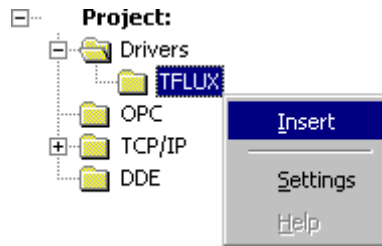
✋**Tip:** To optimize communication and ensure better performance for the system, it is important to tie the tags in different driver sheets according to the events that must trigger the communication of each group of tags and the periodicity for which each group of tags must be written or read. In addition, it is recommended to configure the addresses of communication in sequential blocks.

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>	-	Not used for this driver
Header	WT	See next table	Defines the command type

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

- To GP, BP, CP, SP, DH, ZG, CG, PP, EV, SG, TT and WT:
  <Command> (e.g.: GP);
- To TC, TS, TR, SS, CU, ZS and ST:
  <Type><TypeGroup>:<AddressReference> (e.g.: N7:0)
- To Unsolicited Message:
  <Command><SubCommandNumber> (e.g.: TC01)

- Command: Command type (GP=General Program, BP=Scale Program, CP=Container Program, SP=Autosimulation Program, TC=User Field Title Program, TS=Subttotalizer Title Program, TR=Report Header Program, SS=Subtotalizer Sequence Program, CU=User Field Content Program, DH=Date and Hour, ZG=Reset General Total, ZS=Reset Subtotalizer, CG=General Commands, PP=Weighing Ready, EV=Error Messages and Events, SG=General Status, TT=Accumulated General Total, ST=Accumulated Subtotal or WT=Liquid Weight and Tare);
- SubCommandNumber: The meaning of this parameter varies according to the Command type:
  - To TC command: SubCommandNumber = Title Number of the User field
  - To TS command: SubCommandNumber = Title Number of the Subtotalizer
  - To TR command: SubCommandNumber = Line Number of the Report header
  - To SS command: SubCommandNumber = Sequence Step
  - To CU command: SubCommandNumber = Title Number of the User field
  - To ZS command: SubCommandNumber = Title Number of the Subtotalizer

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

You can type string 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 below:

Information regarding the parameter "Header"			
Type	**Sample of syntax**	**Valid range of SubCommandNumber**	**Comment**
General Program	GP	-	Read/Write data configured in the Address column
Scale Program	BP	-	Read/Write data configured in the Address column
Container Program	CP	-	Read/Write data configured in the Address column
Autosimulation Program	SP	-	Read/Write data configured in the Address column
User Field Title Proram	TC01	01 to 06	Read/Write data configured in the Address column
Subttotalizer Title Program	TS01	01 to 20	Read/Write data configured in the Address column
Report Header Program	TR01	01 to 06	Read/Write data configured in the Address column
Subtotalizer Sequence Program	SS01	01 to 30	Read/Write data configured in the Address column

Information regarding the parameter "Header" (cont.)			
Type	**Sample of syntax**	**Valid range of SubCommandNumber**	**Comment**
User Field Content Program	CU01	01 to 06	Read/Write data configured in the Address column
Date and Hour	DH	-	Read/Write and receives unsolicited data configured in the Address column. Whenever the device (TOLFLUX 9300) is powered on it requires the Date and Time from the PC.
Reset General Total	ZG	-	Reset the "General Total".
Reset Subtotalizer	ZS01	01 to 20	Reset data configured in the Address column
General Commands	CG	-	Write data configured in the Address column
Weighing Ready	PP	-	Read unsolicited message with the weigh value whenever the device (TOLFLUX 9300) is ready to send it.
Error Messages and Events	EV	-	Read unsolicited messages with the event code as soon as it happens.
General Status	SG	-	Read and receive unsolicited messages with the status whenever they change of value in the device (TOLFLUX 9300).
Accumulated General Total	TT	-	Read the accumulated general total value.
Accumulated Subtotal	ST01	01 to 20	Read the accumulated subtotal value.
Liquid Weight and Tare	WT	-	Read and receive unsolicited messages with the liquid weight and tare values. The device (TOLFLUX 9300) send these values each three seconds.

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 **<AddressValue>**.

Addressing Configuration				
Header	**Address**	**Data Type**	**Range of Values**	**Comment**
GP	0	String	**S** (Yes) ; **N** (No)	General program ok
	1	Integer	1 to 31	Tolflux number, report and network
	2	String	**S** (Yes) ; **N** (No)	Network operation / peer to peer
	3	String	**S** (Yes) ; **N** (No)	Remote display
	4	String	**H** (Hour) ; **M** (Minute) ; **S** (Second)	Unit of flow rate
	5	String	**T** ; **V** ; **B**	Analog exit
	6	String	**S** (4/20mA) ; **N** (2/10V)	Signal type
	7	String	**<String Value>**	Supervisor password
	8	String	**<String Value>**	Operator password
	9	String	**S** (Yes) ; **N** (No)	Keyboard supplier program
	10	String	**S** (Yes) ; **N** (No)	Continue transmitting weight
BP	0	String	**S** (Yes) ; **N** (No)	Scale Program Ok
	1	Integer	**<Integer Value>**	Decimal point position.
	2	Integer	**<Integer Value>**	Increase size.
	3	Integer	**<Integer Value>**	Scale capacity (kg)
	4	Integer	**<Integer Value>**	Detection zero range (kg)
	5	Float	**<Float Value>**	Detection zero time (9,9 sec)
	6	Float	**<Float Value>**	Authorization feeding time (9,8 sec)
	7	String	**S** (Yes) ; **N** (No)	Verify feeding
CP	0	Integer	**<Integer Value>**	Liquid weght of container (kg)
	1	Real	**<Float Value>**	Impact time (9,9 sec)
	2	Integer	**<Integer Value>**	Dosing time (998 sec)
	3	Integer	**<Integer Value>**	Discharge time (998 sec)
	4	Real	**<Float Value>**	Stability time (9,9 sec)
	5	Integer	**<Integer Value>**	Stability range (kg)
	6	Integer	**<Integer Value>**	Container anticipation (kg)
SP	0	String	**S** (Yes) ; **N** (No)	Output turn on flag
	1	String	**S** (Yes) ; **N** (No)	Input simulation flag
	2	String	**S** (Yes) ; **N** (No)	Weighting simulation flag
	3	Integer	**<Integer Value>**	Fast feeding flow (kg/sec)
	4	Integer	**<Integer Value>**	Slow feeding flow (kg/sec)
	5	Integer	**<Integer Value>**	Discharge flow (kg/sec)
	6	Integer	**<Integer Value>**	Anticipation (kg/sec)
TC	0	String	**<String Value>**	User field title (max. length = 20 bytes)
TS	0	String	**<String Value>**	Subtotalizer title (max. length = 20 bytes)
TR	0	String	**<String Value>**	Report header line (max. length = 60 bytes)

Addressing Configuration (cont.)				
Header	**Address**	**Data Type**	**Range of Values**	**Comment**
SS	0	String	**S** (Yes) ; **N** (No)	Sequence configured
	1	String	**S** (Yes) ; **N** (No)	Restart sequence
	2	Integer	**0** to **20**	Subtotalizer number (00=End Sequence)
	3	Integer	**<Integer Value>**	Subtotalizer value
CU	0	String	**<String Value>**	User field content (max. length = 40 bytes)
DH	0	Integer	**0** to **31**	Day
	1	Integer	**0** to **12**	Month
	2	Integer	**0** to **9999**	Year
	3	Integer	**0** to **23**	Hour
	4	Integer	**0** to **59**	Minute
ZG	0	-	**-**	Reset general total.
ZS	0	-	**-**	Reset subtotalizer.
CG	0	String	**S** (Yes) ; **N** (No)	Start/ Restart automatic cycle
	1	String	**S** (Yes) ; **N** (No)	Stop automatic cycle
	2	String	**S** (Yes) ; **N** (No)	Silent alarm
	3	String	**S** (Yes) ; **N** (No)	Last charge
	4	String	**S** (Yes) ; **N** (No)	Verify charge
	5	String	**S** (Yes) ; **N** (No)	Discharge container and finalize automatic cycle
	6	String	**S** (Yes) ; **N** (No)	Abort automatic cycle
PP	0	Integer	**<Integer Value>**	Sequence of weighting
	1	Integer	**<Integer Value>**	Tare (kg)
	2	Integer	**<Integer Value>**	Weight (kg)
	3	Integer	**<Integer Value>**	Subtotatiler number
	4	Integer	**<Integer Value>**	Subtotal accumuled (kg)
	5	Integer	**<Integer Value>**	Flow rate (ton/hour)

| Addressing Configuration (cont.)				
Header	**Address**	**Data Type**	**Range of Values**	**Comment**
EV	0	Integer	**1** to **61**	Error or event code: 1 – Feeding authorization fail 2 – Discharge authorization fail 3 – Feeding flow fail 4 – Power supply fail 5 – Scale fail 6 – Scale tare fail 7 – Scale autozero fail 8 – Scale clean fail 9 – Scale overload fail 10 – Scale out of zero fail 11 – Network communication fail 12 – Input and output boards (opto22) fail 13 – Container hi level fail 14 – Air pressure fail 20 –Feeding timeout 21 – Discharge timeout 30 – Cycle stoped 31 – Verify charge 32 – Waiting 33 – End subtotal configured in the sequence 40 – Configure batch 41 – Printing fail 50 – Aborted batch 51 – End batch 52 – Last container of batch 60 – Start of configured sequence 61 – Scale ready
SG	0	String	**S** (Yes) ; **N** (No)	Automatic operation
	1	String	**S** (Yes) ; **N** (No)	Automatic cycle stoped
	2	String	**S** (Yes) ; **N** (No)	Alarme started
	3	String	**S** (Yes) ; **N** (No)	Manual operation
TT	0	Integer	**<Integer Value>**	General total value.
ST	0	Integer	**<Integer Value>**	Subtotalizer value.
WT	0	Integer	**<Integer Value>**	Liquid weight value.
	1	Integer	**<Integer Value>**	Tare value.

4.5   Device Configuration

The TOLFLUX 9300 Manual indicates the following serial communication settings:

- Baud Rate: 9600
- Data bits: 7
- Stop bits: 2
- 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**
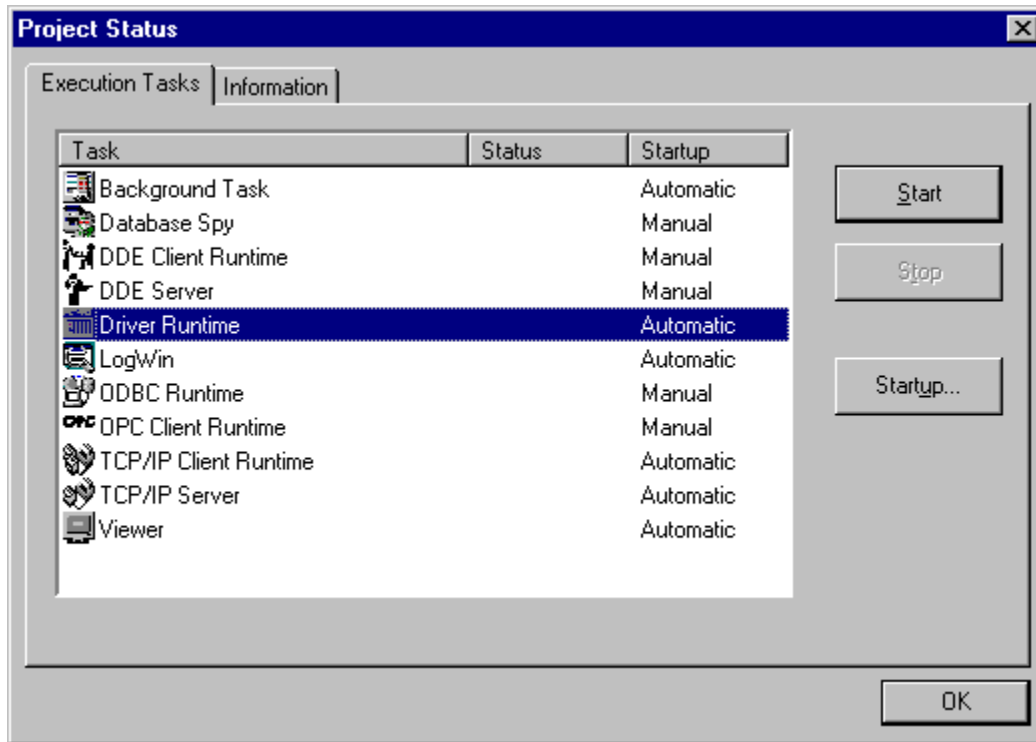
6   Troubleshooting

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 numbet 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.	- Disconnected cables - PLC turned off, or in Stop or error mode - Wrong Station number - Wrong RTS/CTS control settings.	- 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.	- PLC in stop or error mode - Wrong station number - Wrong parity - Wrong RTS/CTS configuration settings	- 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
1.01	Roberto V. Junior	07-Jan-2003	▪ Modified treat of protocol.
1.00	Roberto V. Junior	05-Jul-2001	▪ Initial version