Communication Driver Messung

Driver for serial communication with devices using proprietary ASCII protocol

Index

1 Introduction

It is recommended that you read this document in its entirety before implementing the communication between the Studio and the device. If after reading this document, you are still unable to establish communication, then contact Studio Technical Support Group. The Technical Support Group will need the following information from you to better assist you:

- Operating System (Type and Version): To find this information, click on the start icon. Select the menu option for *settings* and the folder for *Control Panel*. In the *Control Panel* window, double-click on the icon for System. This action should provide you the operating system type and version number.
- Studio version: To find this information, select *Project* from the main menu bar and then select the *status option*. The information is located under the *system information* tab.
- Driver version: Double click in the driver icon that appears in the bottom-left side of the screen after the driver is initialized.
- Device's model and boards: Please refer to the hardware manufacturer's documentation.

This document contains 7 parts, as follow:

- Introduction: Provides an overview of the driver documentation.
- General hardware and software pre-requisites: Provides information necessary to identify all the required components (hardware and software) necessary to implement the communication such as global characteristics about the communication.
- Installation: Explains the procedures that must be followed to install the software and hardware required for the communication.
- Configuration: Provides the necessary information to configure the Studio driver such as the different permutations for configuration and its default values.
- Execution Procedure: Explain the steps to test whether the driver was correctly installed and configured.
- Error codes: Supplies a list of the most common error codes for this protocol and the procedures to fix them.
- History of versions: Provides a log of all the modifications done in the MESSU driver since version 1.00.

☞ Note: This document presumes that the user has read the chapter *Driver Configuration* of the Studio's TECHNICAL REFERENCE that is part of the Studio product.

2 General Characteristics

2.1 Device Characteristics

- Manufacturer: Messung. (PLC Messung devices and compatibles with proprietary protocol ASCII through serial communication.)

2.2 Link Characteristics

- Device communication port: COM1, COM2, COM3…
- Physical protocol: RS485 (RS232 x RS485 converter needed).
- Logic protocol: Proprietary Protocol
- Device Runtime software: None

2.3 Studio Driver Characteristics

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

The Studio driver is composed of the following files:

- MESSU.INI: This file contains the default values for various parameters of this driver.
- MESSU.MSG: This file contains the error messages for each error code. In case of communication errors from commands such as **Field Read Commands** , **Field Write Commands and Enable Write on Tag Change,** these messages will be shown in the output windows of the Studio environment. 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 for this driver are:

0;Ok.
1;Invalid address.
2;Invalid station.
3;Protocol error.
4;Invalid checksum.
5;Invalid block size.
90;EC Invalid command.
91;EC Check sum error.
92;EC Invalid variable.
93;EC Invalid address out of range.
94;EC Page not programmed.
95;EC Read only page.
96;EC Invalid characters.

☞ Note: The information above uses the syntax **error code;message** . Section VI covers a full description of each error message and the procedures to follow to correct this error.

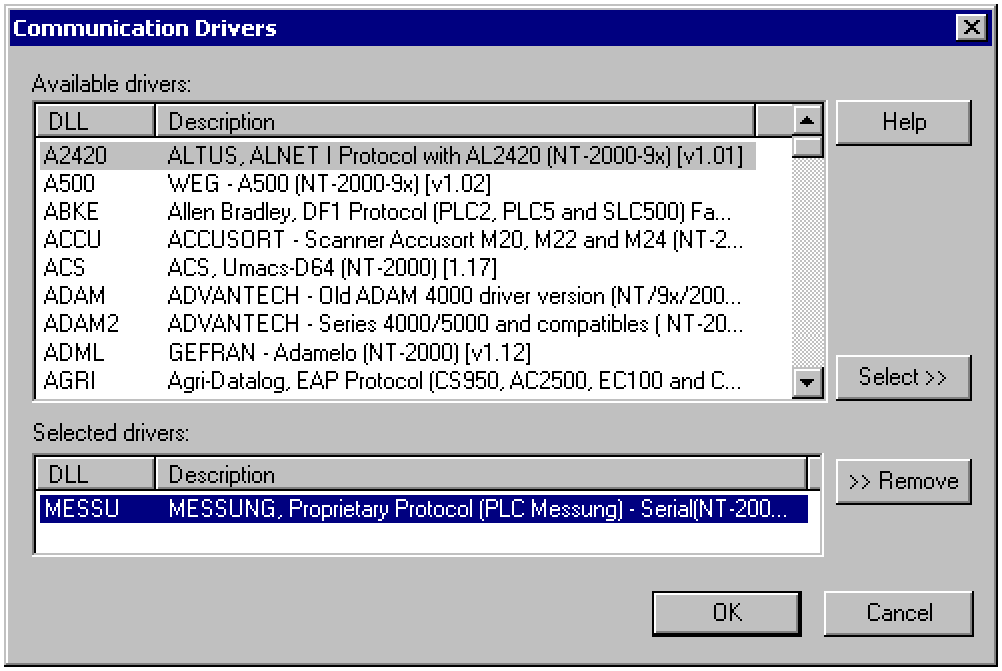
- MESSU.DOC: This document provides detailed documentation about the MESSU driver.
- MESSU.DLL: This is the compiled library for the MESSU driver.

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

3 Installation

3.1 *Installing Studio Driver*

Select the menu *Insert* + *Driver...*
In the column **Available Drivers**, select the **MESSU Driver** and push the button **Select>>>** (the driver MESSU must appear in the column **Selected Drivers**).
Push the button **OK**.



DLL	Description
A2420	ALTUS, ALNET I Protocol with AL2420 (NT-2000-9x) [v1.01]
A500	WEG - A500 (NT-2000-9x) [v1.02]
ABKE	Allen Bradley, DF1 Protocol (PLC2, PLC5 and SLC500) Fa...
ACCU	ACCUSORT - Scanner Accusort M20, M22 and M24 (NT-2...
ACS	ACS, Umacs-D64 (NT-2000) [1.17]
ADAM	ADVANTECH - Old ADAM 4000 driver version (NT/9x/200...
ADAM2	ADVANTECH - Series 4000/5000 and compatibles (NT-20...
ADML	GEFRAN - Adamelo (NT-2000) [v1.12]
AGRI	Agri-Datalog, EAP Protocol (CS950, AC2500, EC100 and C...

DLL	Description
MESSU	MESSUNG, Proprietary Protocol (PLC Messung) - Serial(NT-200...

3.2 *Other software requirements*

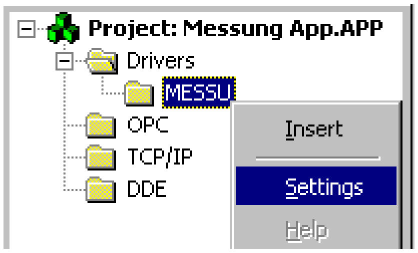
It is not necessary to install any other software in the PC to enable the communication between the Studio and the Device.

4 Studio Driver Configuration

After the MESSU driver is installed and selected in the previous steps (see section III.1), you will see that its configuration is composed of two parts: communication parameters and driver sheets.

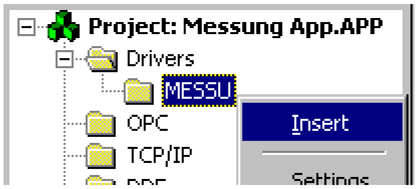
- Communication parameters: *Settings of the serial communication.* These parameters are valid for all driver worksheets configured in the Studio system. To open the window for configuration of the **Communication parameters**, follow these steps:

1. Run In the window **Workspace** of the Studio environment, select the table **Comm**.
2. Expand the folder **Drivers** and select the subfolder **MESSU**.
3. Click the right button of the mouse on the **MESSU** subfolder and select the option **Settings**.



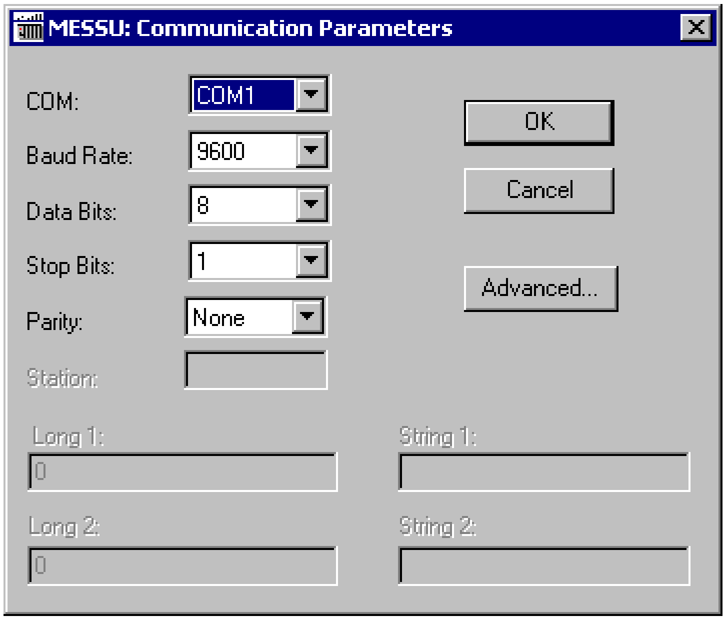
- Driver Sheets: It is possible to configure many driver worksheets, each one will be composed of a Header and Body. To create a new driver worksheet, follow these steps:

1. In the window **Workspace** of the Studio environment, select the table **Comm**.
2. and the folder **Drivers** and select the subfolder **MESSU**.
3. Click the right button of the mouse on the **MESSU** 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.

4.1 Communication Parameters



MESSU: Communication Parameters

COM: COM1

Baud Rate: 9600

Data Bits: 8

Stop Bits: 1

Parity: None

Station:

Long 1: 0

String 1:

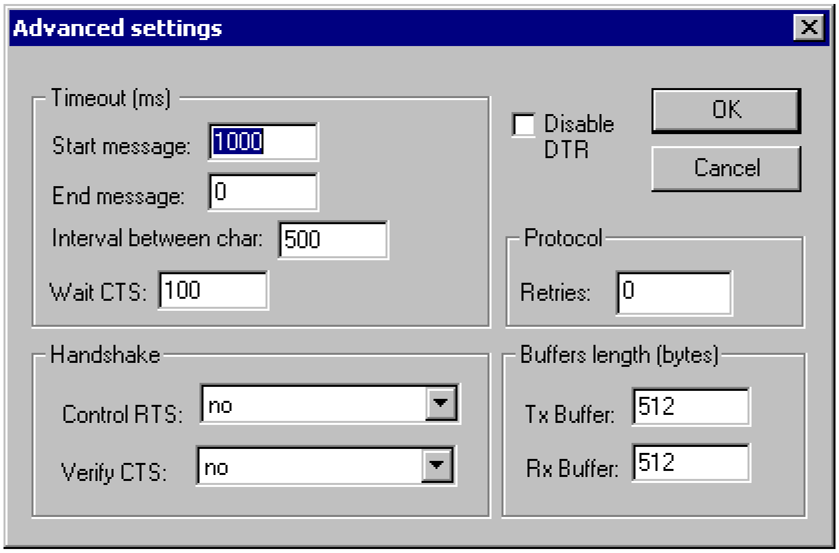
Long 2: 0

String 2:

Parameter	Default Value	Valid values	Description
COM	COM1	COM1 to COM8	Serial port of the PC used to communication with the device
Baud Rate	9600	9600	Communication rate of data
Data Bits	8	8	Number of data bits used in the protocol
Stop Bits	1	1	Number of stop bits used in the protocol
Parity	None	Even,Odd,None,Space,Mark	Parity of the protocol

☞ Note: The device MUST be configured with the SAME values defined in the **Communication Parameters** window of the MESSU driver.

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



Advanced settings

Timeout (ms)

Start message: 1000

End message: 0

Interval between char: 500

Wait CTS: 100

□ Disable DTR

Protocol

Retries: 0

Handshake

Control RTS: no

Verify CTS: no

Buffers length (bytes)

Tx Buffer: 512

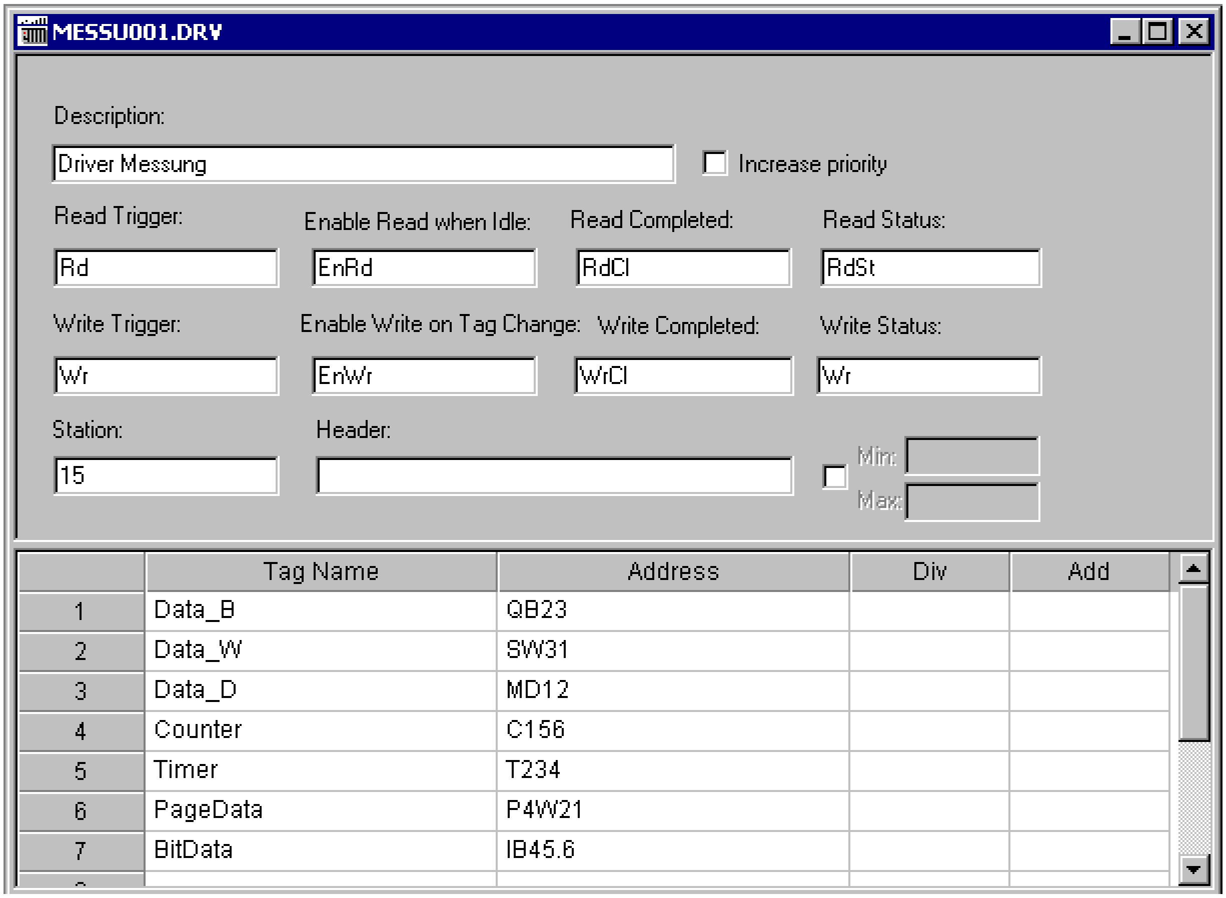
Rx Buffer: 512

Parameter	Default Value	Valid values	Description
Start message (ms)	1000	0 to 10000	Maximum time to receive the beginning of the message from the device (time-out time)
End message (ms)	0	0 to 10000	Maximum time to receive the end of the message from the device from the beginning of the message. (Note: the value zero mean that the driver will not check these times.)
Interval between char	500	0 to 10000	Maximum time between the characters sent from the device
Wait CTS (ms)	100	0 to 10000	Maximum time to receive the CTS signal after setting the RTS signal (Note: valid only if the parameter **Verify CTS** has the value **yes**)
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
Verify CTS	No	no or yes	Define if the driver must wait for the handshaking signal of CTS (Clear to Send) before send a

			message
Disable DTR	Not checked	Not checked or checked	If checked, the driver will not set the DTR signal before starting the communication
Retries	0	0 to 9	The number of retries by each tag configured in the driver worksheet if failure occurs.
Tx Buffer (bytes)	512	0 to 512	Maximum size of the buffer of information to be sent from the driver
Rx Buffer (bytes)	512	0 to 512	Maximum size of the buffer of information to be received from the device

☞ Note: Generally, these parameters must be changed when using a DCE (Data Communication Equipment) - converter (232/485, for example), modem, etc - between the PC, driver and the host. It is necessary to know the characteristics of the DCE before adjust these parameters.

4.2 Driver Worksheet



MESSU001.DRV

Description: Driver Messung

□ Increase priority

Read Trigger: Rd
Enable Read when Idle: EnRd
Read Completed: RdCl
Read Status: RdSt

Write Trigger: Wr
Enable Write on Tag Change: EnWr
Write Completed: WrCl
Write Status: Wr

Station: 15
Header:

□ Min: Max:

	Tag Name	Address	Div	Add
1	Data_B	QB23		
2	Data_W	SW31		
3	Data_D	MD12		
4	Counter	C156		
5	Timer	T234		
6	PageData	P4W21		
7	BitData	IB45.6		

4.2.1 Header of the Driver Worksheet:

Parameter	Default Value	Valid values	Description
Description	-	Text (up to 80 characters)	Documentation of the driver sheet
Read Trigger	-	Tag (boolean, integer, real or string)	When the tag configured in this field changes value, all the addresses configured in the worksheet are read from the device once.
Enable Read When Idle	-	Numerical value or Tag (boolean, integer, real or string)	While the tag or numerical value configured in that field are higher than zero (0), the addresses configured in the worksheet are continually read from the device.
Read Complete	-	Tag (boolean)	The tag configured in this field toggles its value automatically each time that the tags of the worksheet are read from the device.
Read Status	-	Tag (boolean)	The tag configured in this field will receive the error code of the communication of the worksheet (see section II.3)
Write Trigger	-	Tag (boolean, integer, real or string)	When the tag configured in this field changes value, all the tags configured in the worksheet are written to the device once.
Enable Write on Tag Change	-	Numerical value or Tag (boolean, integer, real or string)	While the tag or numerical value configured in this field is higher than zero (0), any changes in the configuration of the tag in the worksheet will be written to the device.
Write Complete	-	Tag (boolean)	The tag configured in this field toggles its value automatically each time the tags of the worksheet are written to the device.
Write Status	-	Tag (boolean)	The tag configured in this field will receive the error code of the communication of the worksheet (see section 2.3)
Station	-	See the table below	The ID (node) of the device (Unit name).
Increase read priority	Not checked	Not checked or checked	Enhances the priority of read to the same level of write commands. If this

			field is not checked, then the write commands will have priority over the read commands.
Min/Max	Not checked	Not checked or checked	When checked, it enables a range of values to be converted to engineering format (more details ahead).
Header	-	This field is not used.	This field is not used.

- Station Field

The valid range is 01 to 31.

4.2.2 Body of the Driver Worksheet:

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

Address Field

The address field complies to the following syntax: Variable<PageNumber><DataType>Offset<.Bit>

Where:

- Variable: I (Input), Q (Output), M (Memory), S (System), T (Timer), C (Counter) or P (Pages). See table below.
- <PageNumber>: Valid only with Pages Variable. Valid Pages is 0 to 31.
- <DataType>: See table below. Valid only with Input, Output, Memory, System and Pages Variables.
- Offset: Address Variable in the device. Each address has 1 byte (except for Timer and Counter variable). If use datatype Word (2 bytes) for example QW12, the address Q24 and Q25 will be accessed in the Device.
- <.Bit>: Bit of address. 0 to 7 (Byte), 0 to 15 (Word) or 0 to 31 (Double).

Attention: You can use the Bit Writing function only with the "`Write on tag chang`e" driver tag enabled, which
means that you cannot use the "`Write trigge`r" tag for the Bit Writing function.

Sample of Addressing Configuration

Address Field	Address on the Device
QB12	Q12
QW12	Q24, Q25
QD12	Q48, Q49, Q50, Q51
QW12.10	Q24 bit 2
QW12.5	Q25 bit 5
QW12.0	Q25 bit 0
T123	T123
MB97	M97
ID4	I16, I17, I18, I19
SW31	S62, S63
C56.13	C56 bit 13
C56.0	C56 bit 0
P30W367.13	P30.P734 bit 5
P30W367.1	P30.P735 bit 1
P30W367	P30.P734, P30.P735

Variable	Range	DataType Support
I (Input) 1 Byte	I0 to I63	All
Q (Output) 1 Byte	Q0 to Q63	All
M (Memory) 1 Byte	M0 to M255	All
S (System) 1 Byte	S0 to S63	All
T (Timer) 2 Bytes	T0 to T255	None
C (Counter) 2 Bytes	C0 to T255	None
P (Pages) 1 Byte	P0.P0 to P0.P1023 … P31.P0 to P31.P1023	All

Data Type	Size	Range
B (Byte)	1 Byte	0 to 255
W (Word)	2 Bytes	0 to 65535
D (Double)	4 Bytes	0 to 2^32 –1

The next columns of the body worksheet can be Div and Add or Min and Max, according to the check box Min/Max of the Header worksheet. The meaning of each column is explained in the follow table:

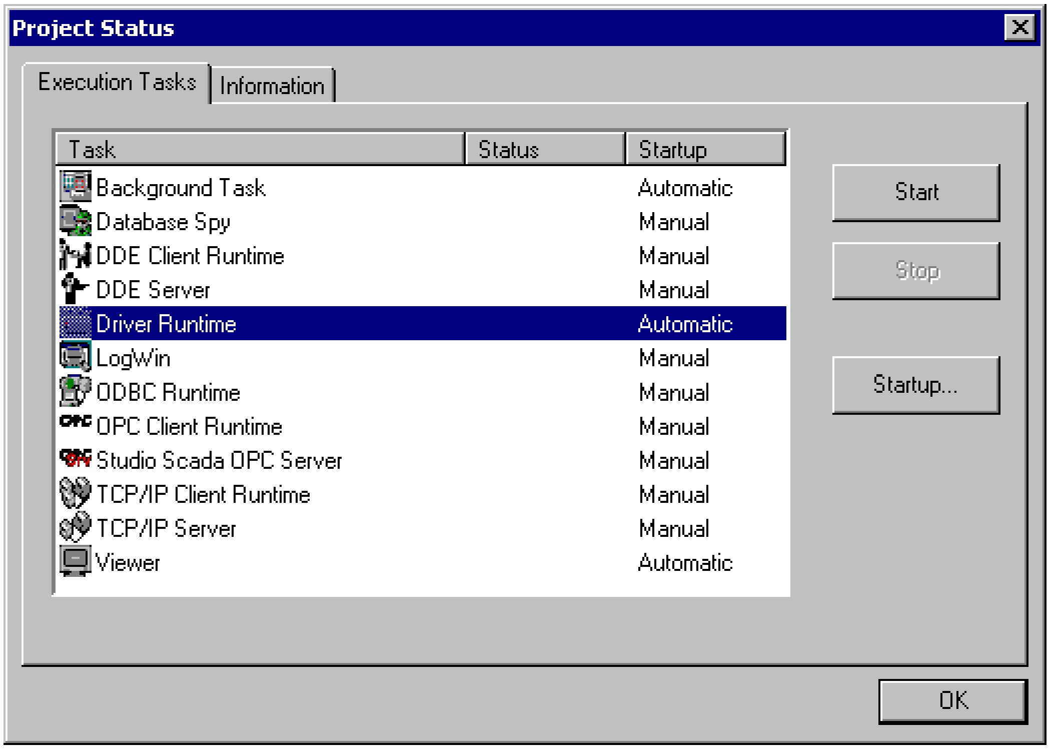
Column	Range of Values	Meaning
Div	Any numeric value	In read commands: Tag = (Device value) / **DIV** In write commands: Device value = Tag * **DIV**
Add	Any numeric value	In read commands: Tag = (Device value) + **ADD** In write commands: Device value = Tag – **ADD**
Min	Any numeric value	Defines the minimum value assigned for the tag when the corresponding device's value is equal to the value defined in the field Min of the Driver Worksheet Header.
Max	Any numeric value	Defines the maximum value assigned for the tag when the corresponding device's value is equal to the value defined in the field Max of the Driver Worksheet Header.

☞ Note: If the columns of the table above are configured with null, the tags of the drive sheet will receive the same value as the address configured.

5 Execution

In the previous sections we provided you with the explanations and procedures for installing and configuring the MESSU and Studio (please see section III and IV), now you need to start the communication between the Mussung's device and the Studio system.

- Turn on the device and connect it to the configured serial port in the PC, where the Studio will be running;



Task	Status	Startup
Background Task		Automatic
Database Spy		Manual
DDE Client Runtime		Manual
DDE Server		Manual
Driver Runtime		Automatic
LogWin		Manual
ODBC Runtime		Manual
OPC Client Runtime		Manual
Studio Scada OPC Server		Manual
TCP/IP Client Runtime		Manual
TCP/IP Server		Manual
Viewer		Automatic

- To verify if the module Runtime Driver is running, use the menu option **Project** + **Status…**

☞ Note: If the task Runtime Driver is not started, select it and click on the button **Start** shown in the windows above.

- To simulate commands to read or write values from or to the host, this action is done according to the configuration of each driver sheet.

☞ 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.

6 Troubleshooting

Error Code	Description (*)	Possible causes	Procedure to solve
0	OK	Communication without problems	-
1	Invalid address.	Bit Write done with “Write Trigger”	Bit write must be done with “Enable Write on Tag Change”
2	Invalid station.	Station Invalid	Enter a valid Station
3	Protocol error.		
4	Invalid checksum.		
5	Invalid block size.	Too many line configured in the sheet.	
90	EC Invalid command.		
91	EC Check sum error.		
92	EC Invalid variable.		
93	EC Invalid address out of range.		
94	EC Page not programmed.		
95	EC Read only page.		
96	EC Invalid characters.		

7 History of Versions

The objective of this section is to log and maintain a history of all the changes made in the driver since version 1.00.

Version	By	Date	Description of changes
1.00	Fabio H. Y. Komura	12-jul-2002	▪ First version of the driver