MATSU Communication Driver

Driver for Serial Communication
with Matsushita FP Devices using Mewtocol Protocol

Contents

Introduction

The MATSU driver enables communication between the Studio system and Matsushita FP devices using the Mewtocol protocol communicating over Serial, according to the specifications discussed in this document.

This document was designed to help you install, configure, and execute the MATSU driver to enable communication with these devices. The information in this document is organized as follows:

- **Introduction**: Provides an overview of the MATSU driver documentation.
- **General Information**: Provides information needed to identify all the required components (hardware and software) used to implement communication between Studio and the MATSU driver.
- **Installing the Driver**: Explains how to install the MATSU driver.
- **Configuring the Driver**: Explains how to configure the MATSU driver.
- **Executing the Driver**: Explains how to execute the driver to verify that you installed and configured the driver correctly.
- **Troubleshooting**: Lists the most common error codes for this protocol and explains how to fix these errors.
- **Sample Application**: Explains how to use a sample application to test the MATSU driver configuration.
- **Revision History**: Provides a log of all modifications made to the driver and the documentation.

Notes:

- This document assumes that you have read the “Development Environment” chapter in the Studio *Technical Reference Manual*.
- This document also assumes that you are familiar with the Windows NT/2000/XP environment.
If you are unfamiliar with Windows NT/2000/XP, we suggest using the **Help** feature (available from the Windows desktop **Start** menu) as you work through this guide.

General Information

This chapter explains how to identify all the hardware and software components used to implement communication between the Studio MATSU driver and the Matsushita FP Device.

The information is organized into the following sections:

- Device Characteristics
- Link Characteristics
- Driver Characteristics

Device Characteristics

To establish communication, you must use devices with the following specifications:

- **Manufacturer**: Matsushita
- **Compatible Equipment**:
  - FP series
- **Programming Software**: None
- **Device Runtime Software**: None

For a list of the devices used for conformance testing, see “Conformance Testing.”

Link Characteristics

To establish communication, you must use links with the following specifications:

- **Device Communication Port**: Serial Port
- **Physical Protocol**: Serial RS232
- **Logic Protocol**: Mewtocol
- **Specific PC Board**: None

Driver Characteristics

The MATSU driver is composed of the following files:

- **MATSU.INI**: Internal driver file. *You must not modify this file*.
- **MATSU.MSG**: Internal driver file containing error messages for each error code. *You must not modify this file*.
- **MATSU.PDF**: Document providing detailed information about the MATSU driver.
- **MATSU.DLL**: Compiled driver.

Notes:

- All of the preceding files are installed in the `/DRV` subdirectory of the Studio installation directory.
- You must use Adobe Acrobat® Reader™ (provided on the Studio installation CD-ROM) to view the *MATSU.PDF* document.

You can use the MATSU driver on the following operating systems:

- Windows NT/2K/XP
- Windows CE 5.x and higher

For a list of the operating systems used for conformance testing, see “Conformance Testing” on page 4.

The MATSU driver supports the following registers:

Register Type	Length	Write	Read	Bit	Integer	Float	Double
RCC	1 Word	–	●	●	●	●	●
WCC	1 Word	●	–	●	●	●	●
RD	1 Word	–	●	●	●	●	●
WD	1 Word	●	–	●	●	●	●
RK	1 Word	–	●	●	●	●	●
WK	1 Word	●	–	●	●	●	●
RS	1 Word	–	●	●	●	●	●
WS	1 Word	●	–	●	●	●	●
WCS	1 Word	●	–	●	●	–	–
RR	1 Word	–	●	●	●	●	●
WR	1 Word	●	–	●	●	●	●
RT	1 Word	–	●	–	●	–	–
RM	1 Word	●	–	–	●	–	–
RP	-	–	●	–	–	–	–
WP	-	●	–	–	–	–	–
RP&	-	–	●	–	–	–	–
WP&	-	●	–	–	–	–	–

Conformance Testing

The following hardware/software was used for conformance testing:

- **Driver Configuration**:
  - **Baud Rate**: 19200
  - **Data Bits**: 8
  - **Stop Bits**: 1
  - **Parity**: Odd
  - **COM Port**: COM1
- **Cable**: Serial RS232

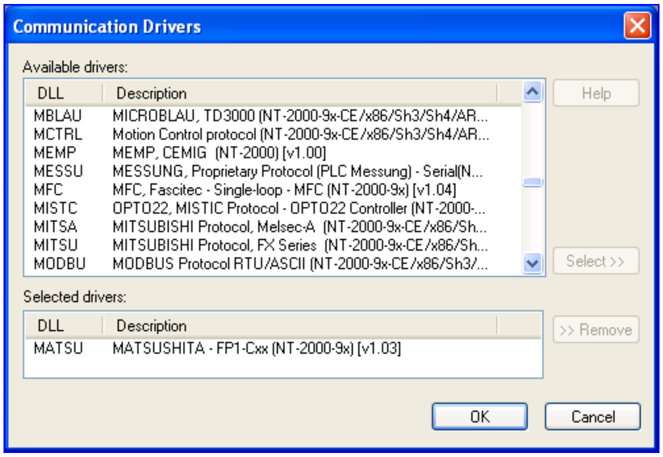
Driver Version	Studio Version	Operating System	Equipment
1.08	V7.1	WinCE 5.0 x86 WinXP	Matsushita FP0 Device

Installing the Driver

When you install Studio version 5.1 or higher, all of the communication drivers are installed automatically. You must select the driver that is appropriate for the application you are using.

Perform the following steps to select the driver from within the application:

1. Open Studio from the **Start** menu.
2. From the Studio main menu bar, select **File** → **Open Project** to open your application.
3. Select **Insert** → **Driver** from the main menu bar to open the *Communication Drivers* dialog.
4. Select the **MATSU** driver from the *Available Drivers* list (as shown in the following figure), and then click the **Select** button.



Available drivers:

DLL	Description
MBLAU	MICROBLAU, TD3000 (NT-2000-9x-CE/x86/Sh3/Sh4/AR...
MCTRL	Motion Control protocol (NT-2000-9x-CE/x86/Sh3/Sh4/AR...
MEMP	MEMP, CEMIG (NT-2000) [v1.00]
MESSU	MESSUNG, Proprietary Protocol (PLC Messung) - Serial(N...
MFC	MFC, Fascitec - Single-loop - MFC (NT-2000-9x) [v1.04]
MISTC	OPTO22, MISTIC Protocol - OPTO22 Controller (NT-2000-...
MITSA	MITSUBISHI Protocol, Melsec-A (NT-2000-9x-CE/x86/Sh...
MITSU	MITSUBISHI Protocol, FX Series (NT-2000-9x-CE/x86/Sh...
MODBU	MODBUS Protocol RTU/ASCII (NT-2000-9x-CE/x86/Sh3/...

Selected drivers:

DLL	Description
MATSU	MATSUSHITA - FP1-Cxx (NT-2000-9x) [v1.03]

Communication Drivers Dialog Box

5. When the **MATSU** driver displays in the **Selected Drivers** list, click the **OK** button to close the dialog.

Note:
It is not necessary to install any other software on your computer to enable communication between the host and the device.

Attention:
For safety reasons, you must use special precautions when installing the physical hardware. Consult the hardware manufacturer’s documentation for specific instructions in this area.

Configuring the Driver

After opening Studio and selecting the MATSU driver, you must configure the driver. Configuring the MATSU driver is done in two parts:

- Specifying communication parameters
- Defining tags and controls in the *STANDARD DRIVER SHEET*s (or Communication tables)

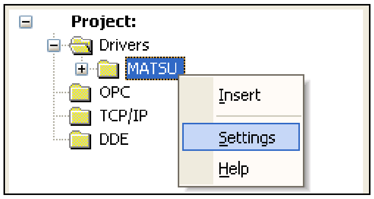
Worksheets are divided into two sections, a *Header* and a *Body*. The fields contained in these two sections are standard for all communications drivers — except the **Station**, **Header** and **Address** fields, which are driver-specific. This document explains how to configure the **Station**, **Header** and **Address** fields only.

> **Note:**
> For a detailed description of the Studio *STANDARD DRIVER SHEET*s, and information about configuring the standard fields, review the product's *Technical Reference Manual*.

Setting the Communication Parameters

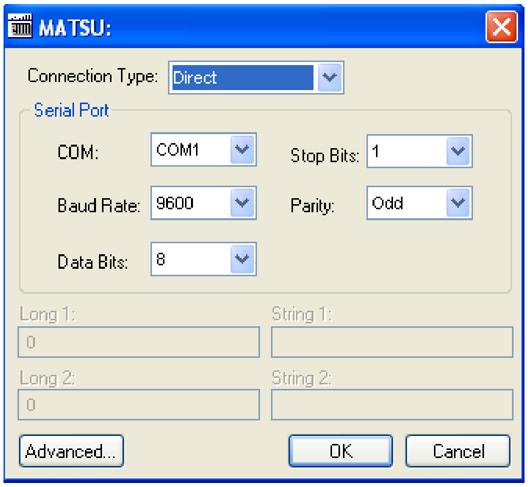
Use the following steps to configure the communication parameters, which are valid for all driver worksheets configured in the system:

1. From the Studio development environment, select the *Comm* tab located below the *Workspace*.
2. Click on the *Drivers* folder in the *Workspace* to expand the folder.
3. Right-click on the MATSU subfolder and when the pop-up menu displays (as shown in the following figure), select the **Settings** option.



Select Settings from the Pop-Up Menu

The *MATSU: Communication Parameters* dialog displays (as follows).


MATSU:

Connection Type: Direct

Serial Port

COM: COM1

Stop Bits: 1

Baud Rate: 9600

Parity: Odd

Data Bits: 8

Long 1: 0

String 1:

Long 2: 0

String 2:

Communication Parameters Dialog

This Driver does not have custom parameters.

4. Click the **Advanced** button on the *Communication Parameters* dialog to open the *Advanced Settings* dialog and configure the settings that are necessary.

✍ **Notes:**

- Do not change any of the other *Advanced* parameters at this time. You can consult the Studio *Technical Reference Manual* for information about configuring these parameters for future reference.
- Generally, you must change the *Advanced* parameter settings if you are using a DCE (Data Communication Equipment) converter (232/485 for example), modem, and so forth between the PC, driver and the host. You must be familiar with the DCE specifications before adjusting these configuration parameters.

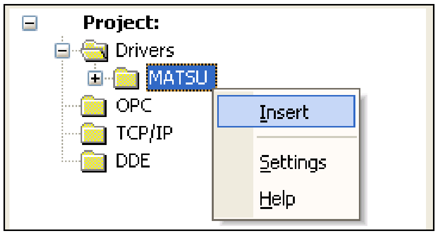
Configuring the Driver Worksheets

This section explains how to configure the *STANDARD DRIVER SHEETs* (or communication tables) to associate application tags with the device addresses. You can configure multiple Driver Worksheets — each of which is divided into a *Header* section and a *Body* section.

Configuring the *STANDARD DRIVER SHEET*

Use the following steps to create a new *STANDARD DRIVER SHEET*:

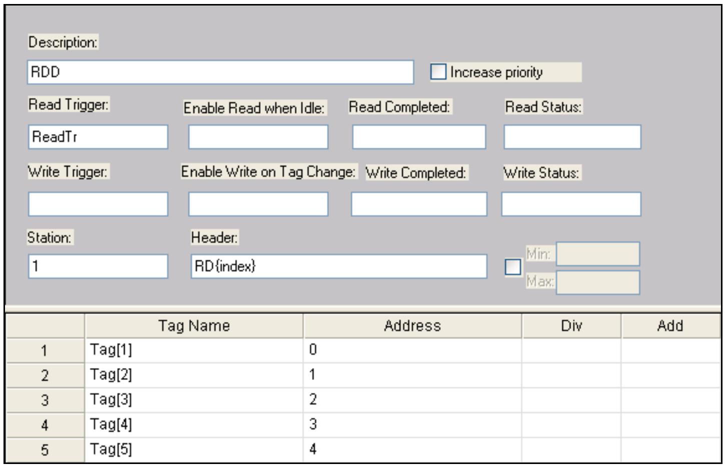
1. From the Studio development environment, select the *Comm* tab, located below the *Workspace* pane.
2. In the *Workspace* pane, expand the *Drivers* folder and right-click the *MATSU* subfolder.
3. When the pop-up menu displays (as shown in the following figure), select the **Insert** option.



Inserting a New Worksheet

> **Note:**
> To optimize communication and ensure better system performance, you must tie the tags in different driver worksheets to the events that trigger communication between each tag group and the period in which each tag group must be read or written. Also, we recommend configuring the communication addresses in sequential blocks to improve performance.

The *STANDARD DRIVER SHEET* displays (similar to the following figure).



Description: RDD
□ Increase priority
Read Trigger: ReadTr
Enable Read when Idle:
Read Completed:
Read Status:
Write Trigger:
Enable Write on Tag Change:
Write Completed:
Write Status:
Station: 1
Header: RD{index}
□ Min: Max:

	Tag Name	Address	Div	Add
1	Tag[1]	0		
2	Tag[2]	1		
3	Tag[3]	2		
4	Tag[4]	3		
5	Tag[5]	4		

STANDARD DRIVER SHEET

In general, all parameters on the Driver Worksheet (except the **Station**, **Header** and **Address** fields) are standard for all communication drivers, but they will not be discussed in this document. For detailed information about configuring the standard parameters, consult the *Studio Technical Reference Manual*.

4. Use the following information to complete the **Station**, **Header** and **Address** fields on this worksheet.

- **Station** field: Specify the device using the following syntax:

  <PLC ID>

  Where:

  – **PLC ID** is the PLC identification number.

- **Header** field: Use the information in the following table to define the type of variables that will be read from or written to the device and a reference to the initial address.

  These variables must comply with the following syntax:

  For **RCC, WCC, RD, WD, RK, WK, RS, WS, WCS, RR, WR** or **RT** commands **:**

  <Command>:[Contact]:<*AddressReference*> (For example: RCC*:X:10*)

  For **RP** and **WP** commands **:**

  <Command>:<Start Step>:<*End Step*>:<*Path to File*>

  (For example: RP:0:20: C:\PLC_PROGRAM.TXT)

  For **RP&** and **WP&** commands **:**

  <Command>:<*Path to File*>

  (For example: WP&:C:\PLC_PROGRAM.TXT)

  Where:

  – **Command** is the command to send to PLC.
  – **Contact** (*optional parameter)* is the contact type. This can be configured if the **Command** supports contacts (**X**, **Y, R or T**). The commands that support contacts generally are **RCC** (**X**, **Y, R or T**), **WCC** (**X**,**Y or R**) and **WCS** (**Y or R**).
  – **AddressReference** is the initial address (reference) of the configured **Command**.
  – **Start Step** and **End Step**, the difference between them can not exceed 23.

After you edit the **Header** field, Studio checks the syntax to determine if it is valid. If the syntax is incorrect, Studio automatically inserts the default value in the **Header** field.

Also, you can type a tag string in brackets **{Tag}** into the **Header** field, but you must be certain that the tag's value is correct and that you are using the correct syntax, or you will get an invalid Header error.

The following table lists all of the data types and address ranges that are valid for the MATSU driver.

Header Field Information			
Data Types	**Sample Syntax**	**Valid Range of Initial Addresses per Worksheet**	**Comments**
RCC	RCC:X:6	Varies according to the equipment	Read Contact Area (16 Contacts)
WCC	WCC:Y:2	Varies according to the equipment	Write Contact Area (16 Contacts)
RD	RD:10	Varies according to the equipment	Read Memory Data Area
WD	WD:25	Varies according to the equipment	Write Memory Data Area
RK	RK:23	Varies according to the equipment	Read Elapsed Values Area
WK	WK:13	Varies according to the equipment	Write Elapsed Values Area
RS	RS:12	Varies according to the equipment	Read Set Values Area
WS	WS:4	Varies according to the equipment	Write Set Values Area
WCS	WCS:R:4	Varies according to the equipment	Write Contact Single (1 Contact)
RR	RR:5	Varies according to the equipment	Read the contents of the system registers
RT	RT:0	Consult the table from page 12	Read the status of the PLC
RM	RM:0	0	Remote control of PLC operation mode
RP	RP:0:25:C:\PLC.TXT	0	Read a program stored in the PLC
WP	WP:0:25:C:\PLC.TXT	0	Write a program which was saved by using the "RP" command back into the PLC
RP&	RP&:C:\PLC.TXT	0	Read settings stored in the PLC
WP&	WP&:C:\PLC.TXT	0	Write settings which was saved by using the "RP&" command back into the PLC

- **Address** field: Use this field to associate each tag to its respective device address.

Type the tag from your application database into the **Tag Name** column. This tag will receive values from or send values to an address on the device. The address must comply with the following syntax:

[Format]<*AddressOffset*>.[*Bit or Contact*] (For example: *10*, *10.5*)

Where:

– **AddressOffset** is a parameter added to the **AddressReference** parameter (configured in the **Header** field) to compose the group address configured in the **Header** field.

– **Bit or Contact** (*optional parameter*) is the bit or contact number to be read from device.

– **Format** (*optional parameter*) is a float or double parameter. Must be configured if the values are in Float Point format or Double (2 Words). Must be configured with '**F**' or '**FS**' to read and write values in Float and Word Swap Float format respectively. '**D**' for double. '**S**' for signed 16-bit integers.

Notes:

- For RM, WP, RP, WP& and RP& commands, the value of the tag will not be considered, but there must be at least one address configured in the driver sheet.

- The maximum number of steps allowed for WP and RP commands is 23, because of the frame size limitation.

➲ **Attention:**

- The address 0 (with Header WCC:R or RCC:R) will access R0 to RF in the PLC, the address 1 will access R10 to R1F in the PLC. So, the last number (in hexadecimal format) means the bit number. For example: R1.1 will access R11 in the PLC.

➲ **Attention:**

- The Floating-point value is stored in two, consecutive registers, where the address value corresponds to the first register position. You must ensure that you do not configure a non-existent address, or a conflict will occur.

➲ **Attention:**

You must not configure a range of addresses greater than the maximum block size (data buffer length) supported by each PLC within the same worksheet. The maximum data buffer length for this driver is 128 words per *STANDARD DRIVER SHEET*.

Address Field Information						
Data Type	**Address Offset**	**Description**	**Value**			
RT	0	CPU Type	**Code**	**CPU Type**	**Code**	**CPU Type**
			05	FP0 2.7K	03	FP3 / C 10K
			06	FP0 5K, 10K	13	FP3 / C 16K
			04	FP1 / M 0.9K	02	FP5 16K
			05	FP1 / M 2.7K	12	FP5 24K
			06	FP1 / M 5K	20	FP10/10S 30K FP10 60K
			20	FP∑	20	FP10SH 30K, 60K, 120K
			50	FP2 16, 32K	20	FP2SH 60K, 120K
	1	CPU Version	**Code**	**CPU Version**		
			10	Version 1.0		
			11	Version 1.1		
			12	Version 1.2		
			…	…		
			35	Version 3.5		
			45	Version 4.5		
	2	Program Capacity (for FP-C/FP3/FP5)	**Code**	**Capacity**	**Code**	**Capacity**
			02	2K (1,534) steps	14	14k (13,822) steps
			04	4k (3,582) steps	16	16k (15,870) steps
			06	6k (5,630) steps	18	18k (17,918) steps
			08	8k (7,678) steps	20	20k (19,966) steps
			10	10k (9,726) steps	22	22k (20,014)steps
			12	12k (11,774) steps	24	24k (24,062) steps
	3	Operation Status	**Bit**	**Specification**	**Content**	
			7	Remote mode	0: NO	1: YES
			6	Message Exist	0: NO	1: YES
			5	Step execution	0: NO	1: YES
			4	Output refresh	0: NO	1: YES
			3	Break position	0: NO	1: YES
			2	Break mode	0: NO	1: YES
			1	Test mode	0: NO	1: YES
			0	Operation mode	0:PROG	1: RUN
	4	Link information	This information is meaningless for reading the status of the programmable controller, ignore this information			

Address Field Information						
Data Type	**Address Offset**	**Description**	**Value**			
			Bit	**Specification**	**Content**	
RT	5	Error Flags	7	Operation error	0: NO	1: YES
			6	Battery error hold	0: NO	1: YES
			5	Battery error real time	0: NO	1: YES
			4	I/O verify error	0: NO	1: YES
			3	Advanced unit error	0: NO	1: YES
			2	N/A	-	-
			1	Power voltage dip	0: NO	1: YES
			0	Diagnostic error	0:NO	1: YES
RM	0	Remote control of PLC operation mode	Value	Operation Mode		
			0	PROGRAM		
			1	RUN		

Notes:

- For RT command the addressing must follow the same line number in the driver sheet, and the data format will be always integer. For example, address 0 must be the first line in the driver sheet, address 2 must be the third line in the driver sheet.

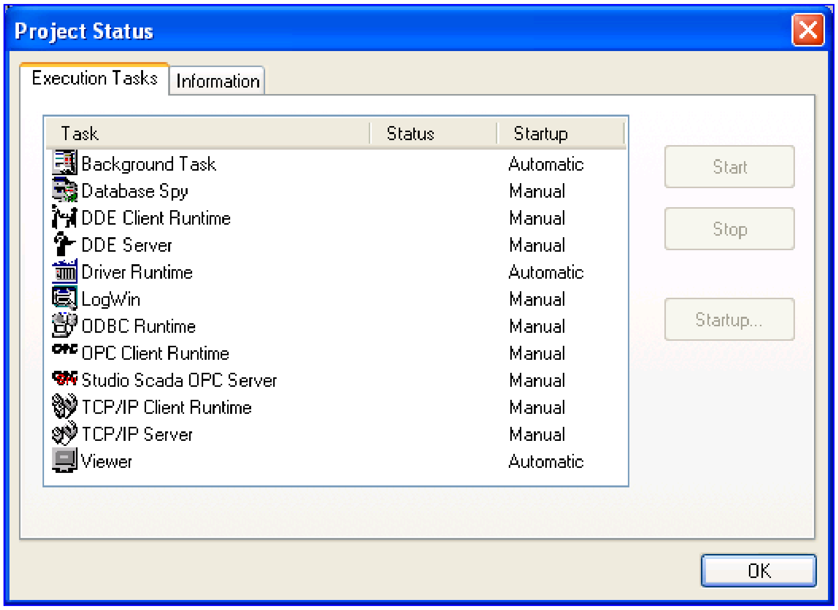
Executing the Driver

After adding the MATSU driver to a project, Studio sets the project to execute the driver automatically when you start the run-time environment.

To verify that the driver run-time task is enabled and will start correctly, perform the following steps:

1. Select **Project** → **Status** from the main menu bar.

   The *Project Status* dialog box displays, as follows.



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

Project Status Dialog Box

2. Verify that the *Driver Runtime* task is set to **Automatic**.
   - If the setting is correct, click **OK** to close the dialog box.
   - If the **Driver Runtime** task is set to **Manual**, select the **Driver Runtime** line. When the **Startup** button becomes active, click the button to toggle the *Startup* mode to **Automatic**.
3. Click **OK** to close the *Project Status* dialog.
4. Start the application to run the driver.

Troubleshooting

If the MATSU driver fails to communicate with the device, the tag you configured for the **Read Status** or **Write Status** fields will receive an error code. Use this error code and the following table to identify what kind of failure occurred.

Error Code	Description	Possible Causes	Procedure to Solve
0	OK	Communication without problems	None required
1	Protocol error	Protocol error	Contact your Studio technical support representative.
2	Invalid Block Size	Offset is greater than the maximum allowed. The maximum offset is usually 128.	Specify a valid offset or create a new Driver Worksheet.
3	Message Block does not end correctly.	Protocol error	Contact your Studio technical support representative
4	Wrong station number in response	Station number typed is invalid.	Check the number station.
6	Error of CheckSum.	Protocol error	Contact your Studio technical support representative
17	File Error	The file could not be opened	Check if file exists
18	Configuration no match	Information could not be found in the file, or the file is corrupted.	Be sure the file was created using WP or WP& command. If the problem persists, contact your Studio technical support representative.
20	Invalid Address	Wrong Address format.	Correct the Address field.
30	Invalid Header	Wrong Header format.	Correct the Header field.
121	NACK error	Data error such as parity error and framing error occurred	Check the communication format, cable connection and ambient noise level.
122	WACK error	Overflow of the receive–buffer occurred in the local node.	Contact your Studio technical support representative
123	Source station number overlap	Source station number overlapped with that for another node, then the communication was shut down.	Reconfigure the station number in the network without overlap.
124	Transmission error	Data not conforming to the transmission protocol format was sent. Or a frame overflow or data error occurred	Contact your Studio technical support representative
125	Link unit hardware error	Communication parts of the link unit did not work properly.	Check the cable connection and ambient noise level. Turn OFF the power for the system and then turn it ON. If communication cannot be performed well, probably a hardware abnormality
126	Station number setting error	The station number set for the source node was outside the specified range	Set the station number within the specified range for the network.
127	Frame–over error	Data over the specified limit was transmitted.	Contact your Studio technical support representative
128	No response error	No response was sent back to the source station from the destination node.	Re–send the same data again.
129	Buffer close error	Data was transferred to or from the source node when its buffer was closed.	Contact your Studio technical support representative.

Error Code	Description	Possible Causes	Procedure to Solve
130	Time–out error.	Data cannot be transferred.	Re–send the same data again.
140	BCC error	BCC error occurred in the command data.	Contact your Studio technical support representative.
141	Format error	The command message does not match the protocol format.	Check the driver sheet configuration. Check if the operation is valid for the operand
142	Not–support error	The command not supported by the source or destination node was transmitted.	Check if the command sent is supported by the source and destination nodes
143	Procedure error	Another series of messages was sent to one node when a series of messages in multiple frames was being sent.	Control the flux of commands so that another message series is not sent while one series is still in progress
150	Link setting error	The route number where no link unit existed was specified in the computer link function.	Check the route number and set the correct one.
151	Simultaneous operation error	The send–buffer overflowed while sending data to the local node in the computer link function.	Re–send data.
152	Sending disable error	The sending operation to another node cannot be performed in the computer link function.	Check the cable connection and ambient noise level. Turn OFF the power for the system and then turn it ON. If communication cannot be performed well, probably a hardware abnormality
153	Busy error	A new command was received from a local node while processing multiple frames.	Re–send the command again.
160	Parameter error	The area code specified is not available for the CPU or the command in the computer link function.	Reset the correct area code.
161	Data error	The specified data format, such as number system, data range, etc. was not correct.	Contact your Studio technical support representative.
163	Mode error	In the current operation mode of the PLC, operation of the command cannot be performed.	Change the operation mode.
165	Protect error	The program was written to the PLC when writing to memory was prohibited.	It is impossible to write program into the PLC when the memory is protected.
166	Address error	The address setting format, such as number system , address limitations, etc. was not correct.	Correct the address format
167	No data error	The area without data was specified for reading.	Specify the correct area for reading.
172	Time–out error	The CPU could not receive the answer within the specified time.	Re–send data.
173	Time–out error	The receive–buffer did not become available within the specified time.	Re–send data.
-15	Timeout Start Message	Disconnected cables PLC is turned off, in stop mode or in error mode Wrong station number Wrong RTS/CTS control settings	Check cable wiring. Check the PLC state – it must be RUN. Check the station number. Check the configuration. See *Studio Technical Reference Manual* for information about valid RTS/CTS configurations.
-17	Timeout between rx char	PLC is in stop mode or in error mode Wrong station number Wrong parity Wrong RTS/CTS configuration settings	Check cable wiring. Check the PLC state – it must be RUN. Check the station number. Check the configuration. See *Studio Technical Reference Manual* for information about valid RTS/CTS configurations.

⇨ **Tip:**

You can verify communication status using the Studio development environment *Output* window (*LogWin* module). To establish an event log for **Field Read Commands**, **Field Write Commands** and **Serial Communication**, right-click in the *Output* window. When the pop-up menu displays, select the option to set the log events. If you are testing a Windows CE target, you can use the Remote LogWin of Studio (**Tools** → **Remote Logwin**) to get the log events from the target unit remotely.

If you are unable to establish communication with the PLC, try to establish communication between the PLC Programming Tool and the PLC. Quite frequently, communication is not possible because you have a hardware or cable problem, or a PLC configuration error. After successfully establishing communication between the device's Programming Tool and the PLC, you can retest the supervisory driver.

To test communication with Studio, we recommend using the sample application provided rather than your new application.

If you must contact us for technical support, please have the following information available:

- **Operating System** (type and version): To find this information, select **Tools** → **System Information**.
- **Studio Version**: To find this information, select **Help** → **About.**
- **Driver Version**: To find this information, read the full description of the driver on the **Communication Drivers** dialog box.
- **Communication Log**: Displays in the Studio *Output* window (or *LogWin* window) when the driver is running. Be sure to enable the **Field Read Commands**, **Field Write Commands** and **Serial Communication** for the *LogWin* window.
- **Device Model** and **Boards**: Consult the hardware manufacturer's documentation for this information.

Sample Application

You will find a sample application for drivers in the **`/COMMUNICATION EXAMPLES/MATSU`** directory. We strongly recommend that you check for a sample application for this driver and use it to test the driver before configuring your own customized application, for the following reasons:

- To better understand the information provided in each section of this document.
- To verify that your configuration is working satisfactorily.
- To certify that the hardware used in the test (device, adapter, cable and PC) is working satisfactorily before you start configuring your own, customized applications.

> ✍ **Note:**
> This application sample is not available for all drivers.

Use the following procedure to perform the test:

1. Configure the device's communication parameters using the manufacturer's documentation.
2. Open and execute the sample application.

> ⇨ **Tip:**
> You can use the sample application screen as the maintenance screen for your custom applications.

Revision History

Doc. Revision	Driver Version	Author	Date	Description of changes
-	1.00	Marcos Taccolini	Nov/1996	Initial version
-	1.02	Roberto V. Junior	Mar/1997	Included WCS Command
A	1.03	Fabio H.Y. Komura	Jun/28/2004	Included Floating-point
B	1.04	Plínio M. Santana	Feb/05/2007	Included Double format and writing bits.
C	1.04	Plínio M. Santana	Feb/27/2007	Document corrections about accessing registers.
D	1.04	Plínio M. Santana	Mar/05/2007	Document corrections about valid commands.
E	1.05	Eric Vigiani / Rafael R. Fernandes	Aug/10/2007	Implemented the commands RR, WR, RT, RP, WP, RP& and WP& Included additional error codes to the manual
F	1.06	Rafael R. Fernandes	Dec/20/2007	Driver released for Windows CE
G	1.07	Eric Vigiani	Jan/05/2009	Fixed problems with check sum Fixed problems with RP& and WP& operands
H	1.8	André Körbes	Jan/7/2013	Fixed issue with FS address validation Added S format