OMPLC Communication Driver

Driver for Serial Communication with
Omron PLCs Using Host Link Commands

Contents

Introduction

The OMPLC driver enables serial communication between the Studio system and certain Omron devices using Host Link commands, according to the specifications discussed in this document.

This document will help you to select, configure and execute the OMPLC driver, and it is organized as follows:

- **Introduction**: This section, which provides an overview of the document.
- **General Information**: Identifies all of the hardware and software components required to implement communication between the Studio system and the target device.
- **Selecting the Driver**: Explains how to select the OMPLC driver in the Studio system.
- **Configuring the Device**: Describes how the target device must be configured to receive communication from the OMPLC driver.
- **Configuring the Driver**: Explains how to configure the OMPLC driver in the Studio system, including how to associate database tags with device registers.
- **Executing the Driver**: Explains how to execute the OMPLC driver during application runtime.
- **Troubleshooting**: Lists the most common errors for this driver, their probable causes, and basic procedures to resolve them.
- **Sample Application**: Explains how to use a sample application to test the OMPLC driver configuration
- **Revision History**: Provides a log of all changes made to the driver and this documentation.

Notes:
– This document assumes that you have read the "Development Environment" chapter in Studio's *Technical Reference Manual*.
– This document also assumes that you are familiar with the Microsoft Windows NT/2000/XP environment. If you are not familiar with Windows, then we suggest using the **Help** feature (available from the Windows desktop **Start** menu) as you work through this guide.

General Information

This chapter identifies all of the hardware and software components required to implement serial communication between the OMPLC driver in Studio and Omron devices.

The information is organized into the following sections:

- Device Specifications
- Network Specifications
- Driver Characteristics
- Conformance Testing

Device Specifications

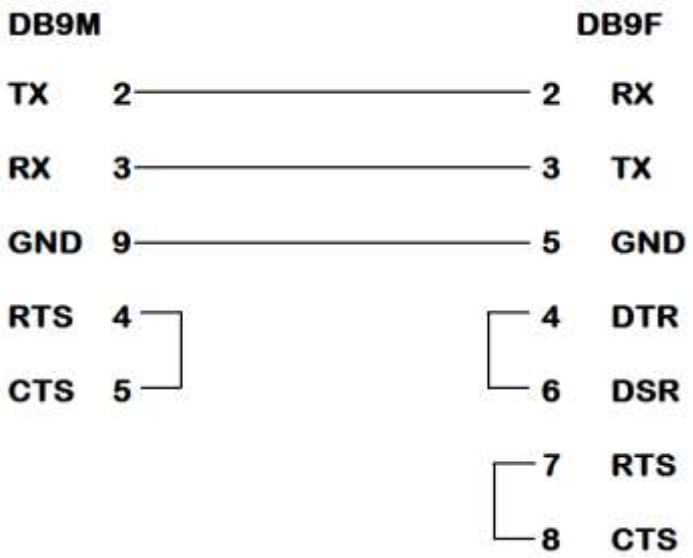
To establish communication, your target device must meet the following specifications:

- **Manufacturer**: Omron
- **Compatible Equipment**:
  - C Series Rack
  - PCs
  - Sysmac Way
  - Host Link Units
- **Programmer Software**: CX-Programmer

Network Specifications

To establish communication, your device network must meet the following specifications:

- **Device Communication Port**: Host Link Unit
- **Physical Protocol**: RS232
- **Logic Protocol**: Host Link
- **Device Runtime Software**: None
- **Specific PC Board**: None
- **Cable Wiring Scheme**: This scheme is based on the OMRON cable C200HS-CN220-EU



Driver Characteristics

The OMPLC driver package consists of the following files, which are automatically installed in the **\DRV** subdirectory of Studio:

- **OMPLC.INI:** Internal driver file. *You must not modify this file*.
- **OMPLC.MSG:** Internal driver file containing error messages for each error code. *You must not modify this file*.
- **OMPLC.PDF:** This document, which provides detailed information about the OMPLC driver.
- **OMPLC.DLL:** Compiled driver

> **Note:**
> You must use Adobe Acrobat® Reader™ to view the **OMPLC.PDF** document. You can install Acrobat Reader from the Studio installation CD, or you can download it from Adobe's Web site.

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

- Windows XP/Vista/7
- Windows CE

The OMPLC driver supports the following registers:

Register Type	Length	Write	Read	Bit	Integer	Float	BCD	String
IR Area (I/O and Internal Relays area)	2 Bytes	•	•	•	•	•	•	•
SR Area (Special Relays area)	2 Bytes	•	•	•	•	•	•	•
HR Area (Holding Relays area)	2 Bytes	•	•	•	•	•	•	•
AR Area (Auxiliary Relays area)	2 Bytes	•	•	•	•	•	•	•
LR Area (Link Relays area)	2 Bytes	•	•	•	•	•	•	•
TC Area (Timer and Counter Status area)	1 Bit	•	•	•	—	—	—	—
DM Area (Data Memory Area)	2 Bytes	•	•	•	•	•	•	•
PV Area (Timer and Counter Present Value Area)	2 Bytes	•	•	•	•	•	•	•

Conformance Testing

The following hardware/software was used for conformance testing:

- Equipment: C200H Version CP01 and CPM1A CPU10
- Cable: C200H-CN229-EU + CIF01, with link set to HOST

Configuration:

The Host Link Unit used with the C200H PLC LK201 was set as below:

- SW1: 0 (Unit Number)
- SW2: 0 (Unit Number)
- SW3: 6 (Transmission Speed: 19200)
- SW4: 2 (Command level 1, 2 and 3, parity Even, Transmission format ASCII, 7 bits, w/2 stop bits)
  Note: please notice on these tests, we did not use the Standard Communication parameters, since the baud rate and the parity are different

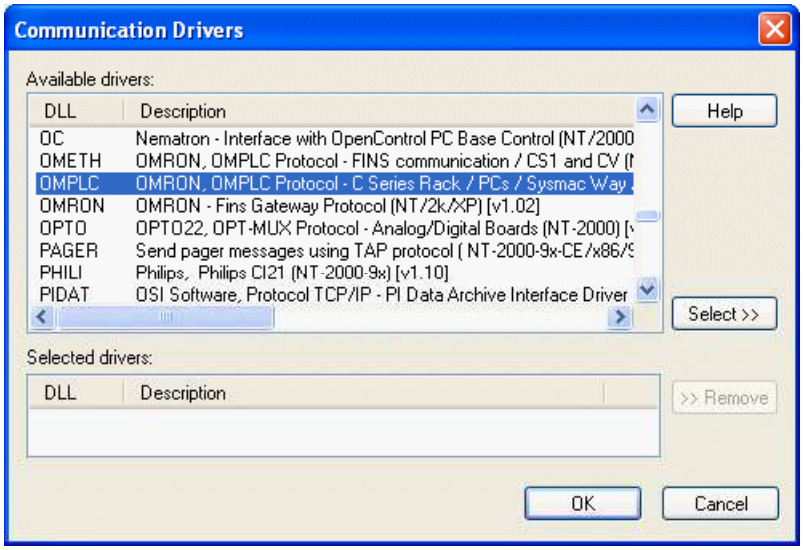
On the Rear of the Host Link Unit, there are 4 flip switches that need do be set like this:
SW1 through SW4 are 1 = off, 2 = off, 3 = on, 4 = off. The red light switch toward the right of the switch should be set to 0V.

- Operating System (development): Windows XP + Service pack 3
- Operating System (target): Windows XP + Service Pack 3; Windows CE v5.0
- Studio Version: 6.1 + Service Pack 6
- Driver version: 3.0

Selecting the Driver

When you install Studio, all of the communication drivers are automatically installed in the **\DRV** subdirectory but they remain dormant until manually selected for specific applications. To select the OMPLC driver for your Studio application:

1. From the main menu bar, select **Insert** → **Driver** to open the *Communication Drivers* dialog.
2. Select the **OMPLC** driver from the *Available Drivers* list, and then click the **Select** button.



Communication Drivers Dialog

3. When the **OMPLC** driver is displayed in the **Selected Drivers** list, click the **OK** button to close the dialog. The driver is added to the *Drivers* folder, in the *Comm* tab of the Workspace.

> **Note:**
>
> It is not necessary to install any other software on your computer to enable communication between Studio and your target device.
>
> However, this communication can only be used by the Studio application; it cannot be used to download control logic to the device. To download control logic to an Omron device, you must also install the Omron programming software (e.g., SYSMAC). For more information, please consult the documentation provided by the device manufacturer.

> **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

Once you have selected the OMPLC driver in Studio, you must properly configure it to communicate with your target device. First, you must set the driver's communication settings to match the parameters set on the device. Then, you must build driver worksheets to associate database tags in your Studio application with the appropriate addresses (registers) on the device.

Configuring the Communication Settings

The communication settings are described in detail in the "Communication" chapter of the Studio *Technical Reference Manual*, and the same general procedures are used for all drivers. Please review those procedures before continuing.

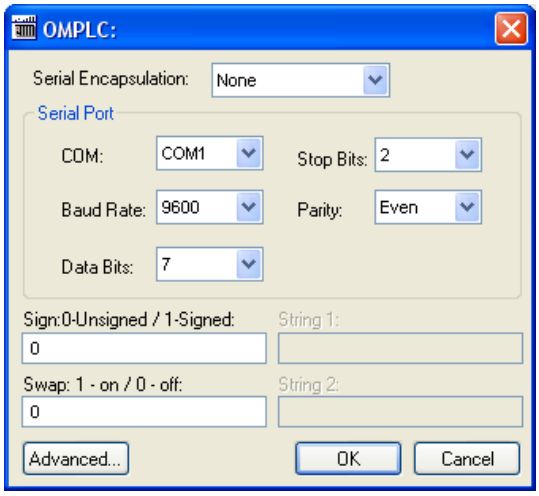
For the purposes of this document, only OMPLC driver-specific settings and procedures will be discussed here. To configure the communication settings for the OMPLC driver:

1. In the *Workspace* pane, select the *Comm* tab and then expand the *Drivers* folder. The OMPLC driver is listed here as a subfolder.

2. Right-click on the *OMPLC* subfolder and then select the **Settings** option from the pop-up menu:

Select Settings from the Pop-Up Menu

The *OMPLC: Communication Settings* dialog is displayed:



OMPLC: Communication Parameters Dialog

3. In the *Communication Settings* dialog, configure the driver settings to enable communication with your target device. To ensure error-free communication, the driver settings must *exactly match* the corresponding

settings on the device. Please consult the manufacturer's documentation for instructions how to configure the device and for complete descriptions of the settings.

Depending on your circumstances, you may need to configure the driver *before* you have configured your target device. If this is the case, then take note of the driver settings and have them ready when you later configure the device.

> **Attention:**
>
> For safety reasons, you **must** take special precautions when connecting and configuring new equipment. Please consult the manufacturer's documentation for specific instructions.

The communication settings and their possible values are described in the following table:

Parameters	Default Values	Valid Values	Description
COM	COM2	COM1 to COM8	PC Serial port used to communicate with the device
Baud Rate	19200	110 to 57600 bps	Data communication rate
Data Bits	7	5 to 8	Number of data bits used in the protocol
Stop Bits	2	1 or 2	Number of stop bits used in the protocol
Parity	Odd	Even, Odd, None, Space, or Mark	Protocol parity
0-Unsigned / 1- Singed	0	0 or 1	- **Unsigned**: The representation of the data is showed without the negative sign. (E.g.: 255) - **Signed**: The representation of the data is showed with the negative sign. (E.g.: -1)
Swap: 0 – off / 1 - on	0	0 or 1	- **off**: The driver doesn't make Byte Swap. - **on**: The driver makes Byte Swap.

> **Note:**
>
> Additional communication settings can be accessed in the *Advanced Settings* dialog. To open this dialog, simply click the **Advanced** button in the *Communication Settings* dialog.
>
> However, you should not need to change the advanced settings for the OMPLC driver. If you want to learn about the advanced settings for future reference, then consult the "Communication" chapter of the *Technical Reference Manual*.

Configuring the Driver Worksheets

A selected driver includes one or more driver worksheets, which are used to associate database tags in Studio with registers on the target device. Each worksheet is triggered by specific application behavior, so that the tags / registers defined on that worksheet are scanned only when necessary – that is, only when the application is doing something that requires reading from or writing to those specific tags / registers. Doing this optimizes communication and improves system performance.

The configuration of these worksheets is described in detail in the "Communication" chapter of the Studio *Technical Reference Manual*, and the same general procedures are used for all drivers. Please review those procedures before continuing.

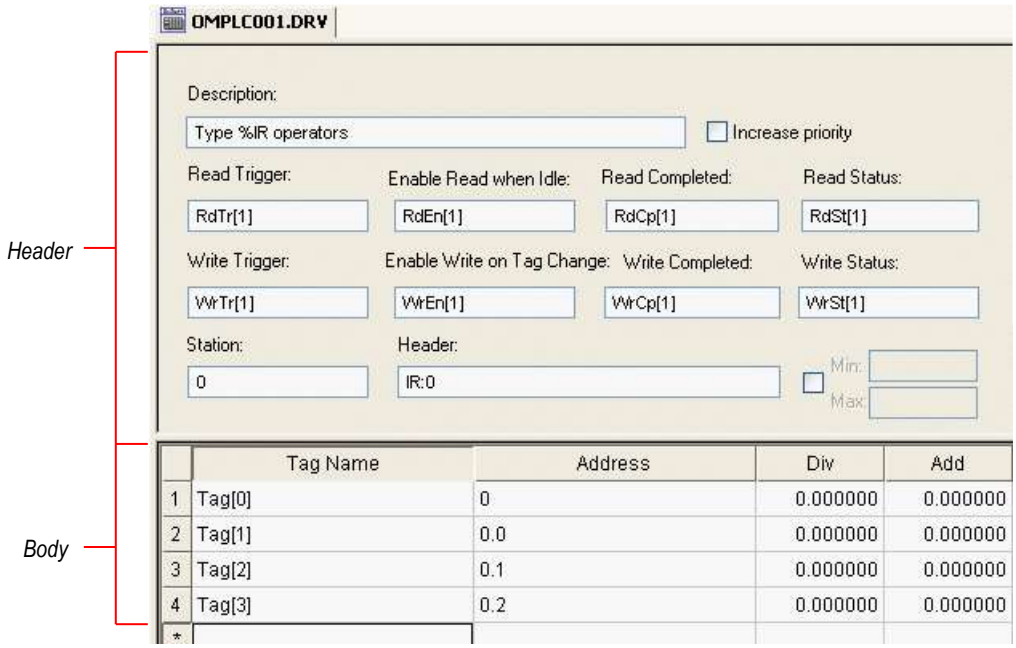
> **Note:**
> We recommend configuring device registers in sequential blocks in order to maximize performance.

To insert a new driver worksheet:

1. In the *Comm* tab, open the *Drivers* folder and locate the *OMPLC* subfolder.
2. Right-click on the *OMPLC* subfolder, and then select **Insert** from the pop-up menu:

Inserting a New Worksheet

A new OMPLC driver worksheet is inserted into the *OMPLC* subfolder, and the worksheet is opened for configuration:



OMPLC Driver Worksheet

> **Note:**
> Worksheets are numbered in order of creation, so the first worksheet is **`OMPLC001.drv`**.

Most of the fields on this worksheet are standard for all drivers; see the "Communication" chapter of the *Technical Reference Manual* for more information on configuring these fields. However, the **Station**, **Header**, and **Address** fields use syntax that is specific to the OMPLC driver.

3. Configure the **Station** and **Header** fields as follows:

   - **Station** field: Specify the ID number (node address) of the device, using the following syntax:

     <Node Address>

     For example, **0**

     Where ***<Node Address>*** is the device node address, from **1** to **31**.

   - **Header** field: Specify the address of the first register of a block of registers on the target device. The addresses declared in the *Body* of the worksheet are simply offsets of this **Header** address. When Read/Write operations are executed for the entire worksheet (see **Read Trigger** and **Write Trigger** above), it scans the entire block of registers from the first address to the last..

     The Header field uses the following syntax:

     <Area>:<AddressReference>

     For example, **IR:3**

     Where:

     – ***<Area>*** is the Memory Area (**IR**, **SR**, **HR**, **AR** and so forth); and
     – ***<AddressReference>*** is the initial (or reference) address of the configured group.

     After you edit the **Header** field, Studio checks the syntax to determine if it is valid. If the syntax is invalid, then Studio automatically inserts a default value of **IR:0**.

     You can also specify an indirect tag (e.g. **{header}**), but the tag that is referenced must follow the same syntax and contain a valid value.

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

Data Types	Sample Syntax	Valid Range of Initial Addresses per Worksheet	Comments
IR Area	**IR:0**	0 to 235	I/O and Internal Relays area
SR Area	**SR:0**	236 to 255	Special Relays area
HR Area	**HR:0**	0 to 99	Holding Relays area
AR Area	**AR:0**	0 to 27	Auxiliary Relays area
LR Area	**LR:0**	0 to 63	Link Relays area
TC Area	**TC:0**	0 to 511	Timer and Counter Status area
DM Area	**DM:0**	0 to 1999	Data Memory Area
PV Area	**PV:0**	0 to 511	Timer and Counter Present Value Area

Note:

The maximum address accepted is variable, according to the PLC.

4. For each table row (i.e., each tag/register association), configure the **Address** field using the following syntax:

   `[Format]<AddressOffset>.[Bit]` (For example, **`F3`**)

   `[Format]<AddressOffset>.[Len]` (For example, **`ST1.5`**)

   `Or`
   `[Signed/Unsigned]<AddressOffset>.[Bit]` (For example, **`U5`**)

   Where:

   – **`[Format]`** defines the format of the register. This is an *optional* parameter; if you leave this field blank, Studio provides the data in Word format (decimal). The **`[Format]`** options are as follows:

   - **BCD**: Binary Code Decimal format
   - **F**: Float format
   - **ST:** String format

   - **`[Signed/Unsigned]`** defines the representation of the data. This is an *optional* parameter; if you leave this field blank, Studio provides the representation of the data based on the parameters you set in the *Communication Parameters* dialog. The **`[Signed/Unsigned]`** options are as follows:

     - **U**: The data is showed without the negative sign (unsigned). (E.g.: 255).
     - **S**: The data is showed with the negative sign (signed). (E.g.: -1).

   – **`<AddressOffset>`** is a value that is added to the **`<AddressReference>`** configured in the **Header** field, to define the address of the group configured in the **Header** field.

   – **`[Bit]`** is the bit number in the device address. This is an *optional* parameter.

   – **`[Len]`** is the WORD string length. It is used with **ST** format.

> **Attention:**
> When reading an invalid BCD value from the PLC, the quality of the tag is set to BAD (0x00). The nibbles 1010, 1011, 1100, 1110 and 1111 are invalid for BCD format.

Sample of Addressing Configuration		
Register in the Device	**Header Field**	**Address Field**
IR0	**`IR:0`**	**`0`**
IR0 bit1	**`IR:0`**	**`0.1`**
IR10	**`IR:0`**	**`10`**
	`IR:10`	**`0`**
SR236	**`SR:236`**	**`0`**
SR236 bit1	**`SR:236`**	**`0.1`**
SR250	**`SR:236`**	**`14`**
	`SR:250`	**`0`**
HR0	**`HR:0`**	**`0`**
HR0 bit1	**`HR:0`**	**`0.1`**
HR10	**`HR:0`**	**`10`**

Sample of Addressing Configuration		
Register in the Device	**Header Field**	**Address Field**
	HR:10	**0**
AR0	**AR:0**	**0**
AR0 bit1	**AR:0**	**0.1**
AR10	**AR:0**	**10**
	AR:10	**0**
LR0	**LR:0**	**0**
LR0 bit1	**LR:0**	**0.1**
LR10	**LR:0**	**10**
	LR:10	**0**
TC0	**TC:0**	**0**
TC0 bit1	**TC:0**	**0.1**
TC10	**TC:0**	**10**
	TC:10	**0**
DM0	**DM:0**	**0**
DM0 bit1	**DM:0**	**0.1**
DM10	**DM:0**	**10**
	DM:10	**0**
PV0	**PV:0**	**0**
PV0 bit1	**PV:0**	**0.1**
PV10	**PV:0**	**10**
	PV:10	**0**

Note:

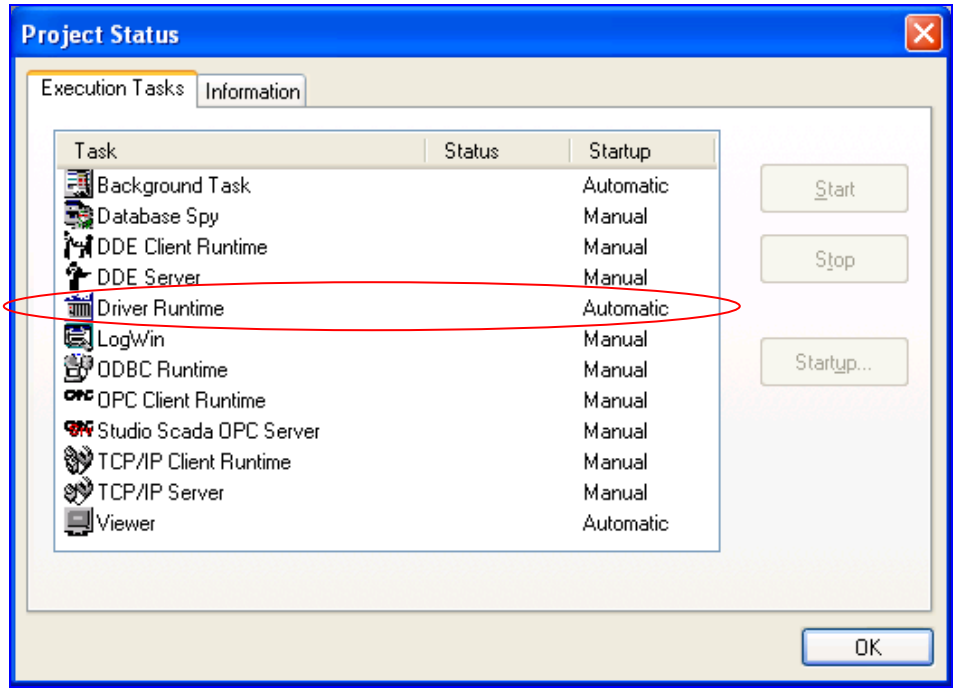
In the previous table is easy to see that there are many ways to set the same register in the device, because the address of the register is defined by the sum of the ***<AddressReference>*** defined in the **Header** field and the ***<AddressOffset>*** defined in the **Address** field. The highest offset allowed for all the areas is 28. Only the TC area has an offset 64. If you type an offset greater then these limits, an error will be generated.

Executing the Driver

By default, Studio will automatically execute your selected communication driver(s) during application runtime. However, you may verify your application's runtime execution settings by checking the *Project Status* dialog.

To verify that the the communication driver(s) will execute correctly:

1. From the main menu bar, select **Project** → **Status**. The *Project Status* dialog displays:



Project Status Dialog

2. Verify that the *Driver Runtime* task is set to **Automatic**.
   - If the setting is correct, then proceed to step 3 below.
   - If the **Driver Runtime** task is set to **Manual**, then select the task and click the **Startup** button to toggle the task's *Startup* mode to **Automatic**.
3. Click **OK** to close the *Project Status* dialog.
4. Start the application to run the driver.

Troubleshooting

If the OMPLC driver fails to communicate with the target device, then the database tag(s) that you configured for the **Read Status** or **Write Status** fields of the Standard Driver Sheet 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	-
1	Not executable in RUN mode	Try to write while the PLC is on the RUN mode	Using the SSS, set the PLC mode to MONITOR
2	Not executable in MONITOR mode	Protocol error	Protocol error
4	Address over (CPM1 PICs)	Protocol error	Protocol error
5	Invalid Header	TAG in the “Header” field with an Invalid value	Type a valid Header entry in this TAG
6	Invalid Address	Negative offset or bit greater then 15	Retype the address at the comm’s sheet.
7	Invalid block size	Offset greater then the operand area commands	Retype the address at the comm’s sheet.
11	Not executable in PROGRAM mode	Protocol error	Check all the hardware’s configuration
19	FCS error	Protocol error	Check all the hardware’s configuration
20	Format error	Protocol error	Check all the hardware’s configuration
21	Entry number data error	Address greater then the high limit of the Area	Retype the address at the comm’s sheet, so the limits must be obeyed.
22	Command not supported	Protocol error	Check all the hardware’s configuration
24	Frame length error	Protocol error	Check all the hardware’s configuration
25	Not executable	Protocol error	Check all the hardware’s configuration
35	User memory write-protected	Protocol error	Check all the hardware’s configuration
163	Aborted due to FCS error in transmit data	Protocol error	Check all the hardware’s configuration
164	Aborted due to format error in transmit data	Protocol error	Check all the hardware’s configuration
165	Aborted due to entry number data error in transmit data	Protocol error	Check all the hardware’s configuration
168	Aborted due to frame length error in transmit data	Protocol error	Check all the hardware’s configuration
-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.
-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.

⇨ **Tip:**

You can monitor communication status by establishing an event log in Studio's *Output* window (*LogWin* module). To establish a log for **Field Read Commands**, **Field Write Commands** and **Serial Communication,** right-click in the *Output* window and select the desired options from the pop-up menu.

You can also use the *LogWin* module (**Tools** → **LogWin**) to establish an event log on a remote unit that runs Windows CE. The log is saved on the unit in the **`celog.txt`** file, which can be downloaded later.

If you are unable to establish communication between Studio and the target device, then try instead to establish communication using the device's own programming software (e.g., SYSMAC). Quite often, communication is interrupted by a hardware or cable problem or by a device configuration error. If you can successfully communicate using the programming software, then recheck the driver's communication settings in Studio.

To test communication between Studio and the device, 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**.
- **Project Information**: To find this information, select **Project** → **Status.**
- **Driver Version** and **Communication Log**: Displays in the Studio *Output* window when the driver is running.
- **Device Model** and **Boards**: Consult the hardware manufacturer's documentation for this information.

Sample Application

A sample application that employs the OMPLC driver is provided on the Studio installation CD. We strongly recommend that you use this sample application to test the driver *before* you develop your own applications, for the following reasons:

- To better understand the information and instructions provided in this document;
- To verify that your driver configuration is working satisfactorily with the target device; and
- To ensure that the all of hardware used in the test (i.e. the device, adapter, cable, and PC) is functioning safely and correctly.

> **Note:**
>
> The following instructions assume that you are familiar with developing project applications in Studio. If you are not, then please review the relevant chapters of the Studio *Technical Reference Manual* before proceeding.

To use the sample application:

1. Configure the device's communication settings according to the manufacturer's documentation.
2. Run Studio.
3. From the main menu bar, select **File → Open Project**.
4. Insert the Studio installation CD and browse it to find the sample application. It should be located in the directory **\COMMUNICATION EXAMPLES\OMPLC**.
5. Select and open the sample application.
6. Configure and test the driver, as described in the rest of this document.

When you have thoroughly tested the driver with your target device, you may proceed with developing your own Studio application projects.

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

Revision History

Doc. Revision	Driver Version	Author	Date	Description of Changes
A	2.00	Roberto V. Junior	30-Jul-1999	▪ First driver version ▪ Driver available for Windows CE
B	2.02	Eric Vigiani	03-Sep-2002	Support BCD format for data exchange.
C	2.03	Eric Vigiani	30-Sep-2002	Fixed in the address parse for BCD data type.
D	2.05	Lidiane A. Moreira	05-Dec-2001	Support float point format for data exchange
E	2.07	Eric Vigiani	25-May-2006	Implemented to not change the tag TimeStamp when it reads values.
F	2.07	Michael D. Hayden	16-Aug-2006	Edited for language and usability.
G	2.08	Eric Vigiani	21-Aug-2006	Implemented the Signed and Unsigned value in the Address field.
H	2.09	Plínio M. Santana	Mar-15-2007	Made change to write negative float values, to sign invalid addresses for bigger addresses than the capacity, and, in DM memory area, to fill with zeros the not used addresses in the blocksize's gap.
I	2.09	Plínio M. Santana	May-14-2007	Made change in the DM operand to work like the others. Document corrections.
J	2.10	Plínio M. Santana	Nov -07-2007	Addresses warning included in the document.
L	2.11	Graziane C. Forti	Apr-18-2008	Inserted String Data Type.
M	2.11	Plinio M. Santana	May-09-2008	Inserted Byte Swap on Configuration Parameters.
M	2.12	Lourenço Teodoro	Dec-16-2008	Updated driver version, no changes in the contents.
N	3.00	Joel Nascimento Paulo Balbino	Jun-19-2009	Modified the driver to support communication with more than one device simultaneously via TCP/IP.
O	3.01	Andre Bastos	Jul-15-2011	Updated documentation: Changed cable configuration Settings Changed default parameters for Communication Settings No changes in the driver code