EUROM Communication Driver

Driver for Serial Communication (RS 232/485)
with Euromap 17-Compatible Devices

Contents

Introduction

The EUROM driver enables communication between the Studio system and Euromap 17-compatible devices according to the specifications discussed in this document. This document was designed to help you install, configure, and execute the EUROM driver to enable communication with EUROM 17-compatible devices. The information in this document 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 EUROM driver in the Studio system.
- **Configuring the Driver**: Explains how to configure the EUROM driver in the Studio system, including how to associate database tags with device registers.
- **Executing the Driver**: Explains how to execute the EUROM driver during application runtime.
- **Troubleshooting**: Lists the most common errors for this driver, their probable causes, and basic procedures to resolve them.
- **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 7/XP/Vista 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 explains how to identify all the hardware and software components used to implement communication between the Studio EUROM driver and the Euromap 17-compatible device. The information is organized into the following sections:

The information is organized into the following sections:

- Device Characteristics
- Link Characteristics
- Driver Characteristics
- Conformance Testing

Device Characteristics

Compatible Devices: **Gammaflux** GLC 2k or any device conforming to the Euromap 17 protocol, through a serial connection.

This driver has been tested successfully with the **Gammaflux** GLC 2k device

Link Characteristics

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

- **Device Communication Port**: Serial
- **Physical Protocol**: RS232/RS485
- **Logic Protocol**: Euromap 17 (ASCII)
- **Device Runtime Software**: None
- **Specific PC Board**: None

Driver Characteristics

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

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

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

- Windows 7
- Windows XP
- Windows Vista

For a description of the operating systems used to test driver conformance, see “Conformance Testing” below.

Conformance Testing

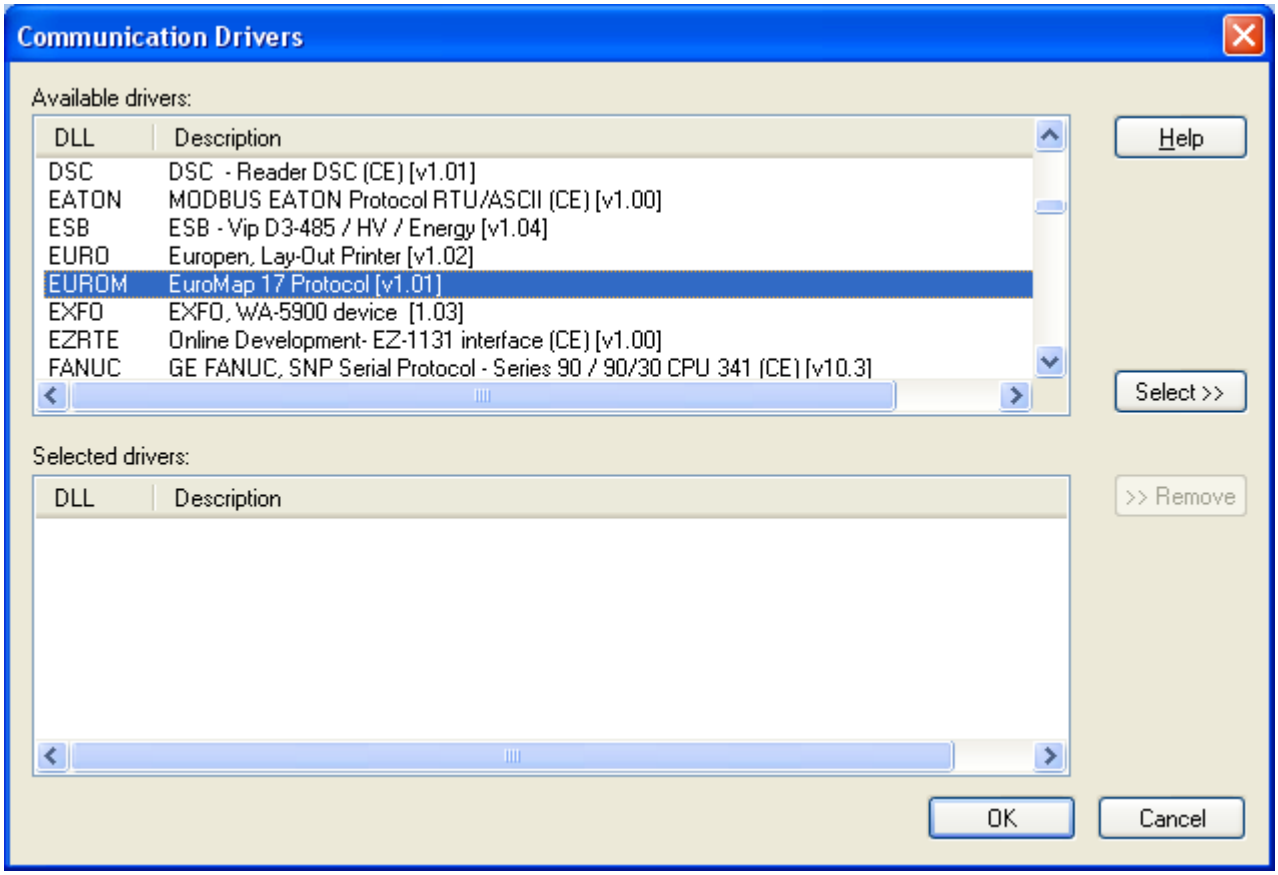
The following hardware and software was used for conformance testing:

- **Equipment**: Gammaflux GLC2k
- **Driver Configuration**:
  - Cable: 9-pin Serial cable
  - Baud rate: 9600
- **Operating System** (development/runtime): Windows 2000
- **Studio version**: 5.1
- **Driver version**: 1.01

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 EUROM driver for your Studio application:

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



Communication Drivers Dialog

3. When the **EUROM** 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.

> ➲ **Attention:**
> For safety reasons, you must take special precautions when installing any physical hardware. Please consult the manufacturer's documentation for specific instructions.

Configuring the Driver

After opening Studio and selecting the EUROM driver, you must configure the driver. For the EUROM driver, you simply define the communication tags by completing a Standard Driver Worksheet.

Configuring the Driver Worksheets

This driver currently does not support Main Driver Sheet. Standard Driver Worksheets must be inserted to define tag/register associations to be monitored, that are triggered by specific application behaviors.

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.

For the purposes of this document, only EUROM driver-specific parameters and procedures will be discussed here.

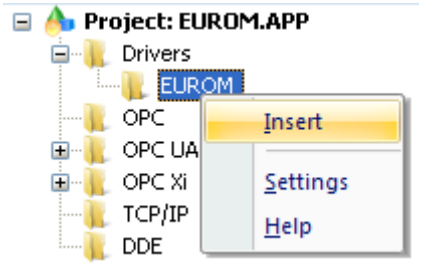
STANDARD DRIVER WORKSHEET

When you select the EUROM driver and add it to your application, it does not have any Driver Sheet added. To start communicating, you must insert Standard Driver Worksheets to define the tags/registers to be monitored and commands to be written. These services are specified by the header used on the driver sheet and the addresses.

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.

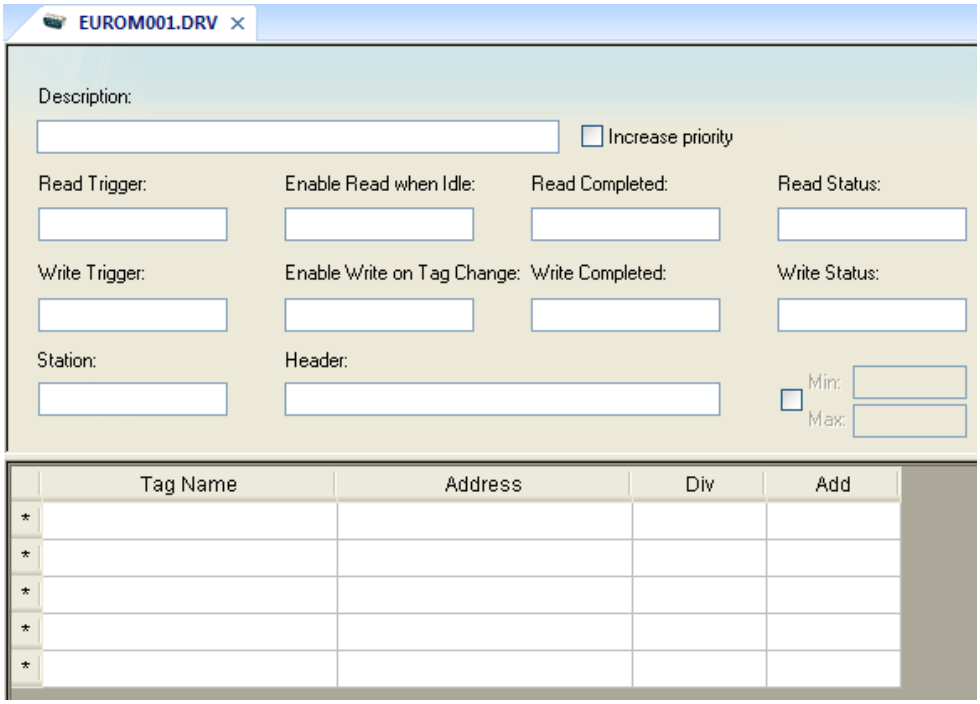
To insert a new driver worksheet:

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



Inserting a New Worksheet

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



EUROM Driver Worksheet

> **Note:**
> Worksheets are numbered in order of creation, so the first worksheet is **`EUROM001.drv`**.

3. Worksheets are divided into two sections, a Header and a Body, and 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.

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

   - **Station** field: Generally, you use this field to specify the IP address of the target device. You do not use this field for serial communications.

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

   - **Header** field: Use the following table to specify the type of command to be sent to the device. This table lists all of the headers that are valid for the EUROM driver, and which operations you can use with that header.

Header	Description	Valid Operations			
		Read Trigger	Write Trigger	Enable Read	Enable Write
SA	First set point	•	•	•	•
SB	Second set point	•	•	•	•

Header	Description	Valid Operations			
		Read Trigger	Write Trigger	Enable Read	Enable Write
UA	Upper limit alarm value	•	•	•	•
LA	Lower limit alarm value	•	•	•	•
UD	Upper deviation alarm value	•	•	•	•
LD	Lower deviation alarm value	•	•	•	•
PV	Actual value, process value	•	–	•	–
AC	Actual current	•	–	•	–
CO	Controller output	•	–	•	–
SW	Status word	•	–	•	–

- **Address** field: You must specify an address in this field, or the Driver Worksheet will not run. Use the following table to associate each command with a specific axis or group of axes. This table lists the valid address ranges and data types for each header.

Header	Valid Addresses	Description	Valid Tag Data Types
SA	0 thru 1000	First set point.	Real
SB	0 thru 1000	Second set point.	Real
UA	0 thru 1000	Upper limit alarm value.	Real
LA	0 thru 1000	Lower limit alarm value.	Real
UD	0 thru 1000	Upper deviation alarm value.	Real
LD	0 thru 1000	Lower deviation alarm value.	Real
PV	0 thru 1000	Actual value, process value.	Real
AC	0 thru 1000	Actual current.	Real
CO	0 thru 1000	Controller output.	Real
SW	0 thru 1000	Status word.	Integer

If the command returns a value, Studio will write that value to the tags in the **Tags** column of the Worksheet.

The **Address** field content must comply with the following syntax:

<Address Offset> or ***<Address Offset>.<Bit>*** Where:

– **Address Offset**: Use in combination with the **Address Reference** parameter (specified in the **Header** field) to specify the address to be read from/written to on the device.

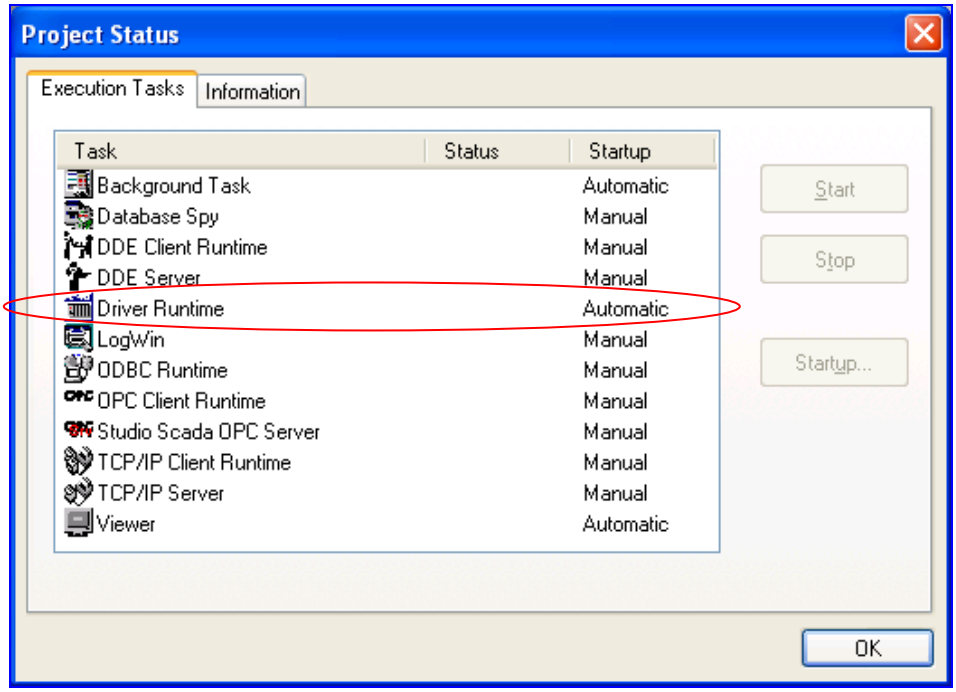
– **Bit**: Indicates a specific bit number (from 0 to 31) in the address. This parameter is *optional* and can be used with the SW header only.

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 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 EUROM 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
0	OK	N/A
3	Invalid Station	Invalid station IP address.
23	Invalid Answer	PLC reply was not recognized.
27	Write Violation	Header does not support the write function.
40	Negative Acknowledge	PLC received and understood the message but cannot comply due to protocol or program limitations.
41	BCC Failed	PLC responded but the message's BCC byte was incorrect.

⇨ **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 **Protocol Analyzer,** right-click in the *Output* window and select the desired options from the pop-up menu.

You can also use the *Remote LogWin* module (**Tools** → **LogWin**) to establish an event log on a remote unit that runs Windows CE

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. 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.

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

- **Operating System and Project Information** (type and version): To find this information, select **Help** → **Support Information**.
- **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.

Revision History

Doc. Revision	Driver Version	Author	Date	Description of Changes
A	1.00	Bryan Morgan	10-Dec-02	First driver version
B	1.01	Bryan Morgan	27-Jan-03	Altered write function and SW read
C	1.01	Andre Korbes	24-Jan-11	Documentation changes only