

**BUEP Communication Driver**

Driver for Serial Communication Between  
Studio and Bosch GmbH Devices

**Contents**

**INTRODUCTION .....2**

**GENERAL INFORMATION.....3**

    DEVICE CHARACTERISTICS .....3

    LINK CHARACTERISTICS .....3

    DRIVER CHARACTERISTICS .....3

    CONFORMANCE TESTING .....4

**INSTALLING THE DRIVER .....5**

**CONFIGURING THE DRIVER .....6**

    SETTING THE COMMUNICATION PARAMETERS .....6

    CONFIGURING THE DRIVER WORKSHEET .....7

**STATUS - SERVICE ..... 11**

**EXECUTING THE DRIVER ..... 12**

**TROUBLESHOOTING ..... 13**

**SAMPLE APPLICATION ..... 15**

**REVISION HISTORY..... 16**

# Introduction

The BUEP driver enables communication between Studio system and the Bosch GmbH devices, according to the specifications discussed in this publication.

This publication was designed to help you install, configure, and execute the BUEP driver to enable communication with the Bosch GmbH devices. The information in this publication is organized as follows:

- **Introduction:** Provides an overview of the BUEP driver documentation.
- **General Information:** Provides information needed to identify all the required components (hardware and software) used to implement communication between Studio and the BUEP driver. method
- **Installing the Driver:** Explains how to install the BUEP driver.
- **Configuring the Driver:** Explains how to configure the communication 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 driver configuration.
- **Revision History:** Provides a log of all modifications made to the driver and the documentation.

<p> <b>Notes:</b></p> <ul style="list-style-type: none"><li>• This document assumes that you have read the “Development Environment” chapter in the product’s <i>Technical Reference Manual</i>.</li><li>• 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 <b>Help</b> feature (available from the Windows desktop <b>Start</b> menu) as you work through this guide.</li></ul>
--

## General Information

This chapter explains how to identify all the hardware and software components used to implement communication between the BUEP driver and Bosch GmbH devices.

The information is organized into the following sections:

- Device Characteristics
- Link Characteristics
- Driver Characteristics

### Device Characteristics

This driver has been tested successfully with the following devices:

- **Manufacturer:** Bosch GmbH
- **Compatible Equipment:** CL200, CL300 and CL500
- **Bosch PLC Programmer Software:** WinSPS

### Link Characteristics

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

- **Device Communication Port:** Serial Port
- **Physical Protocol:** RS 232
- **Logic Protocol:** Buep19e Protocol
- **Adapters/Converters:** None

### Driver Characteristics

The BUEP driver is composed of the following files:

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

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

- Windows 9x
- Windows NT
- Windows 2000
- Windows XP
- Windows CE (x86, SH3, SH4, MIPs, ARM, PPC)

The BUEP driver supports the following registers:

Register Type	Write	Read	Bit	Byte	Word	DWord
Data-Block (D)	•	•	•	•	•	•
Data-buffer (C)	•	•	•	•	•	•
Data-field (d)	•	•	•	•	•	•
Input (E)	•	•	•	•	•	•
Output (A)	•	•	•	•	•	•
Memory (M)	•	•	•	•	•	•

### **Conformance Testing**

The following hardware/software was used for conformance testing:

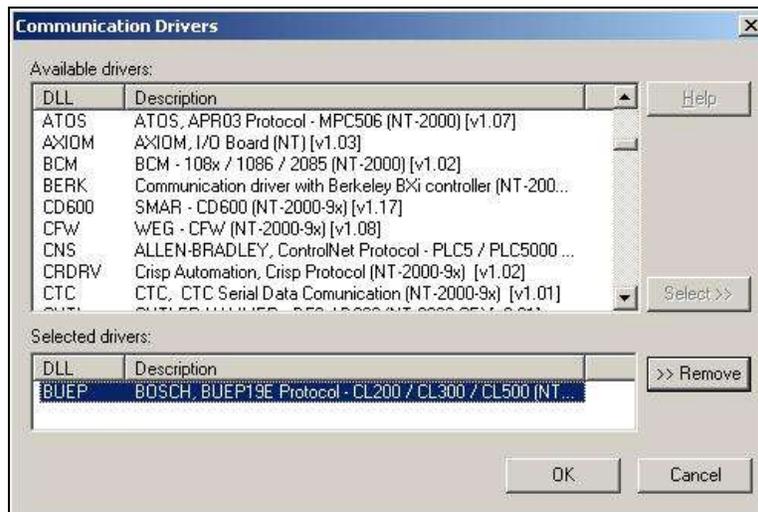
- **Equipment:** CL150
- **Driver Configuration:**
  - **PLC Program:** WinSPS v3.30
  - **Protocol:** Buep19e Protocol
- **Operating System:** Windows XP
- **Studio Version:** 6.0
- **Driver Version:** 1.00

## 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 **BUEP** driver from the *Available Drivers* list, and then click the **Select** button:



**Communication Drivers Dialog**

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

**Attention:**

For safety reasons, you must use care when installing the physical hardware. Consult the hardware manufacturer's documentation for specific installation instructions.

## Configuring the Driver

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

- Specifying communication parameters
- Defining communication tags and controls in the Communication tables or *Driver* worksheet

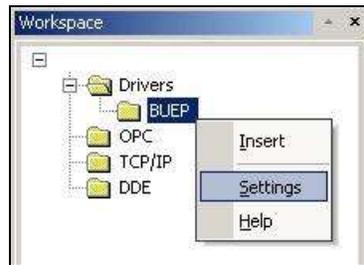
Worksheets are divided into two sections, a *Header* and a *Body*. The fields contained in these two sections are standard for all communications drivers — except the **Station**, **Header**, and **Address** fields, which are driver-specific. This document explains how to configure the **Station**, **Header**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

**Notes:**  
For a detailed description of the Studio *Standard Worksheets*, and information about configuring the standard fields, review the product's *Technical Reference Manual*.

### Setting the Communication Parameters

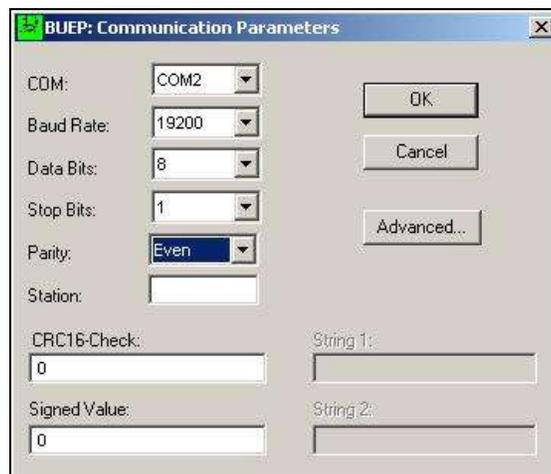
Use the following steps to configure the communication parameters, which are valid for all driver worksheets configured in the system:

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 *BUEP* subfolder and when the pop-up menu displays, select the **Settings** option:



Select Settings from the Pop-Up Menu

The *BUEP: Communications Parameters* dialog displays:



BUEP: Communication Parameters Dialog

4. Click **OK** to close the dialog.

**Notes:**

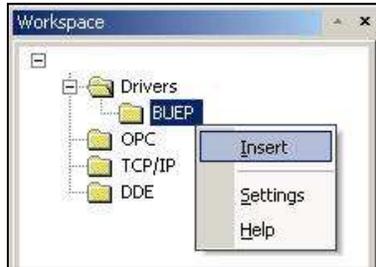
Currently, the LRC8 check sum of this protocol is used exclusively.

### **Configuring the Driver Worksheet**

This section explains how to configure a *Standard Driver Worksheet* (or Communication table) to associate application tags with the PLC addresses. You can configure multiple *Driver* worksheets — each of which is divided into a *Header* section and *Body* section.

Use the following steps to create a new *Standard Driver* worksheet:

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 *BUEP* subfolder.
3. When the pop-up menu displays, 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 `<drivername>.drv` dialog displays (similar to the following figure):

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

**BUEP Driver Worksheet**

In general, all parameters on the *Driver* worksheet (except the **Station**, **Header**, and **Address** fields) are standard for all communication drivers, and they will not be discussed in this publication. 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: No parameter will be filled here (the communication is point-to-point).
- **Header** field: This field contains five values, each separated by a colon:
  - The PLC's field
  - The number of the data block module (if applicable)\*
  - The address of the PLC's unit
  - Whether or not the data will be masked
  - The type for the size of the data (byte, word or dword).

The general sequence of a typical header is the following:

**<field>:<number of DB>:<PLC's unit>:<masking type>:<Size type>**

Where:

- **Field:** The possible values are **D** (data block), **C** (data buffer), **d** (data field), **E** (Input), **A** (Output) and **M** (Memory [Merkerbereich]).
- **Number of DB\*:** The value is the number of the data block to be written to or read from, if you entered **D** in the first position of the header. The accepted range is 0 - 255.  
 \*You will always enter a value in the **<number of DB>** position. If the PLC's field type is **D**, for data block, you will enter the data block number. If the PLC's Field type is not **D**, you must still populate this position, but you can use any digit, such as 0, as the placeholder.
- **PLC's Unit:** The value is the address of the unit of the PLC.
- **Masking type:** The possible values are **M** (Mask, if you wish to write bitwise) and **N** (Normal, if you wish to write byte- or wordwise).
- **Size:** The possible values are **B** (byte size), **W** (word size) and **D** (dword size).

 **Note:**  
 The two important points to remember about the **Header** field are:

1. Always enter a value in the *<number of DB>* position.
2. Use the Mask configuration only for writing bit values to the PLC.

**Example 1:** You are configuring a table to read values from data block number 5. The destination unit of the PLC is 0. All the data will be sent bitwise, i.e. the masking type is normal, and the size of the data will be in bytes. You type the header configuration as follows:

**D : 5 : 0 : N : B**

**Example 2:** You configure another table to write data into the data buffer of the PLC. This data will be sent as bitwise words into the unit number 5. You type the header configuration as follows:

**C : 0 : 5 : M : W**

- **Address** field: Contains the location of the data in the range defined in the header. To write and read data bitwise and wordwise, it is only necessary to fill the offset value that starts in position zero of the header's range. If the data will be sent or read bitwise, you must separate the main address and the bit indication with a . (point).

**Example 1:** The table below shows the configuration for a set of five (5) variables to be written, starting at address number 6 of the data block. These variables will be sent bitwise.

Tag Name	Address Field
Var1	6
Var2	7
Var3	8
Var4	9
Var5	10

**Example 2:** The following table shows a set of eight variables to be sent bitwise, starting at address number 5 of the data block. The header configuration is also shown. Note that the data is Masked, since it is written bitwise.

Header Field	Tag Name	Address Field
D : 5 : 0 : M : B	Var1	5.0
	Var2	5.1
	Var3	5.2
	Var4	5.3
	Var5	5.4
	Var6	5.5
	Var7	5.6
	Var8	5.7

 **Notes:**

- It is only possible to write either one byte bitwise, or a word bitwise at once. That is, if you configure a bitwise masking, you can only send eight Boolean variables, in the case of byte size, and sixteen Boolean variables otherwise. This is a limitation of the Buep19e protocol.
- In the case of Boolean reading, there is no limitation in the number of variables to be read from the

PLC.

- You must not configure a range of addresses greater than 300.

## Status - Service

It is possible to watch whether the PLC is in RUN or STOP state. To do so, it is necessary to configure a separate table, with the following parameters:

1. Enable **Read When Idle** by entering 1 or a chosen variable.

Header Field	Address Field
S:<PLC's unit>	A or a

The variable configured in this table will receive the status value that comes from the PLC. The possible values are:

- 11 - RUN State
- 10 - STOP State

 **Note:**  
This driver cannot show the service status of the CL200.

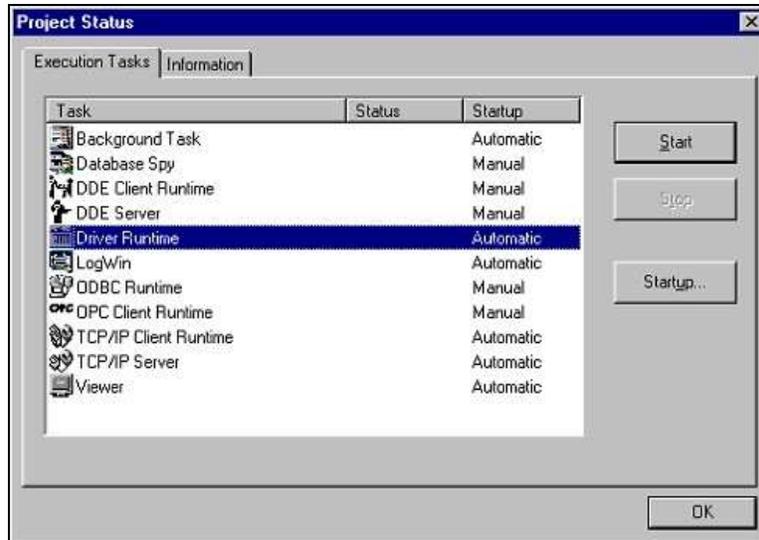
## Executing the Driver

After adding the BUEP 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:

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

The *Project Status* dialog displays:



*Project Status Dialog*

3. Verify that the *Driver Runtime* task is set to **Automatic**.
  - If the setting is correct, click **OK** to close the dialog.
  - 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**.
4. Click **OK** to close the *Project Status* dialog.
5. Start the application to run the driver.

## Troubleshooting

If the BUEP 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
-15	Timeout Start Message	<ul style="list-style-type: none"> <li>▪ Disconnected cables</li> <li>▪ PLC is turned off, in stop mode, or in error mode</li> <li>▪ Wrong station number</li> <li>▪ Wrong RTS/CTS control settings</li> </ul>	<ul style="list-style-type: none"> <li>▪ Check cable wiring.</li> <li>▪ Check the PLC state – it must be RUN.</li> <li>▪ Check the station number.</li> <li>▪ Check the RTS/CTS configuration (see <i>Studio Technical Reference Manual</i> for valid configurations).</li> </ul>
-17	Timeout between rx char.	<ul style="list-style-type: none"> <li>▪ PLC in stop mode or in error mode</li> <li>▪ Wrong station number</li> <li>▪ Wrong parity</li> <li>▪ Wrong RTS/CTS configuration settings</li> </ul>	<ul style="list-style-type: none"> <li>▪ Check cable wiring.</li> <li>▪ Check the PLC state – it must be RUN.</li> <li>▪ Check the station number.</li> <li>▪ Check the RTS/CTS configuration (see “Link Characteristics” for valid RTS/CTS configurations).</li> </ul>
0	No error	Communication without problems	None required.
1	Group with zero elements	Specified amount is an invalid Address number	Specify a valid Address amount
2	Error sending ENQ	ENQ error during communication	Check device and Studio Communication Parameters.
3	Error sending Message	Error during communication	<ul style="list-style-type: none"> <li>▪ Check cable wiring.</li> <li>▪ Check the PLC state – it must be RUN.</li> </ul>
4	Group too big	Offset greater than maximum allowed	Reduce number of elements
5	Check sum error	Invalid Check sum in response message	<ul style="list-style-type: none"> <li>▪ Check the cable wiring.</li> <li>▪ Check the station number.</li> <li>▪ Check the RTS/CTS configuration (see <i>Studio Technical Reference Manual</i> for valid configurations).</li> </ul>

⇒ **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 enable the log at the unit (**Tools** → **LogWin**) and verify the **celog.txt** file created at the target unit.

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**.
- **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 is provided in the `/COMMUNICATION EXAMPLES/<Driver Name>` directory. We strongly recommend that you use this sample application to test the BUEP 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.
3. Execute the *Viewer* module in Studio to display information about the driver communication.



**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
A	1.20	Eric Vigiani	15 Jun 04	First version