

KLOCK Communication Driver

Driver for Serial Communication
with Devices using the SUCOM1 Protocol

Contents

INTRODUCTION 2

GENERAL INFORMATION..... 3

 DEVICE CHARACTERISTICS 3

 LINK CHARACTERISTICS 3

 DRIVER CHARACTERISTICS 3

 CONFORMANCE TESTING 4

INSTALLING THE DRIVER 5

CONFIGURING THE DEVICE 6

CONFIGURING THE DRIVER 6

 SETTING THE COMMUNICATION PARAMETERS 6

 CONFIGURING THE STANDARD DRIVER WORKSHEET 9

EXECUTING THE DRIVER..... 12

TROUBLESHOOTING 13

SAMPLE APPLICATION 15

REVISION HISTORY..... 16

Introduction

The KLOCK driver enables communication between the Studio system and a set of devices using the SUCOM1 protocol by serial link, according to the specifications discussed in this document.

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

- **Introduction:** Provides an overview of the KLOCK driver documentation.
- **General Information:** Provides information needed to identify all the required components (hardware and software) used to implement communication between Studio and the KLOCK driver.
- **Installing the Driver:** Explains how to install the KLOCK driver.
- **Configuring the Device:** Explains how to configure the Klockner-Moeller device.
- **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.



Notes:

- This document assumes that you have read the “Development Environment” chapter in the product’s *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 serial communication between the Studio KLOCK driver and devices using the SUCOM1 protocol.

The information is organized into the following sections:

- Device Characteristics
- Link Characteristics
- Driver Characteristics

Device Characteristics

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

- **Manufacturer:** Klockner-Moeller Devices
- **Compatible Equipment:**
 - PS316/PS306
 - PS32
 - PS4-201-MM1
 - PS416
 - Any device that is compatible with the SUCOM1 protocol

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

Link Characteristics

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

- **Device Communication Port:** Serial
- **Physical Protocol:** RS232
- **Logic Protocol:** SUCOM1
- **Device Runtime Software:** None
- **Specific PC Board:** None

Driver Characteristics

The KLOCK driver is composed of the following files:

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

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

- Windows XP/2K/XP
- Windows CE

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

Conformance Testing

The following hardware/software was used for conformance testing:

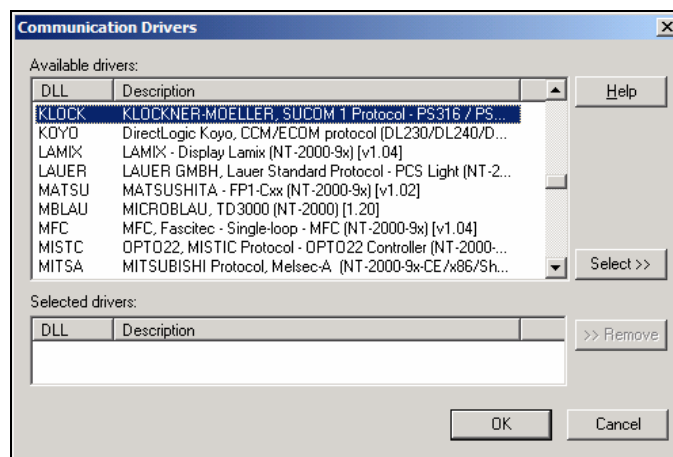
- **Equipment:** PS416
- **Configuration:**
 - **COM Port:** COM1
 - **Baud Rate:** 9600
 - **Data Bits:** 8
 - **Stop Bits:** 2
 - **Parity:** None
- **Cable:** Use specifications described in the “Link Characteristics” section
- **Operating System (*development*):** Windows XP with Service Pack 1
- **Operating System (*target*):**
 - Windows 2000/XP
 - Windows CE
- **Studio Version:** 6.0
- **Driver Version:** 1.07

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 or double-click the Studio shortcut icon on your desktop.
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 **KLOCK** driver from the *Available Drivers* list, and then click the **Select** button.



Communication Drivers Dialog

5. When the **KLOCK** 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 Studio and the device. However, to download the custom program to your device you must install a Klockner-Moeller programmer software package. Consult the Klockner-Moeller documentation for installation instructions.



Caution:

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 Device

Because there are several brands of equipment that use the SUCOM1 protocol, it is impossible to define a standard device configuration. Therefore, we suggest using the following default configuration:


- **Protocol:** RS232
- **Baud Rate:** 9600
- **Data Bits:** 8
- **Stop Bits:** 1 (To PS316/PS306 or PS32) or 2 (To PS4-201-MM1 or PS416)
- **Parity:** Even (To PS316/PS306 or PS32) or None (To PS4-201-MM1 or PS416)

Configuring the Driver

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

- Specifying communication parameters (only one configuration needed).
- Defining communication tags and controls in the Communication tables or *Driver* worksheets (Standard and Main Driver Worksheets).

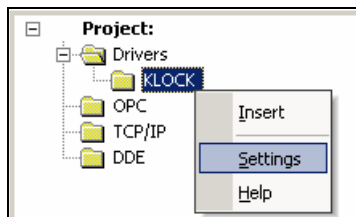
Worksheets are divided into two sections, a *Header* and a *Body*. The fields contained in these two sections are standard for all communications drivers — except the **Station**, **Header**, and **Address** fields, which are driver-specific. This document explains how to configure the **Station**, **Header**,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

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

Setting the Communication Parameters

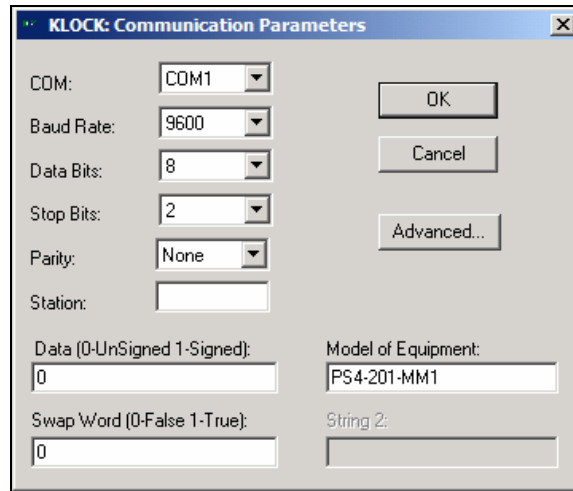
Use the following steps to configure the communication parameters, which are valid for all driver worksheets configured in the system):

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



Select Settings from the Pop-Up Menu

The KLOCK: Communications Parameters dialog displays:



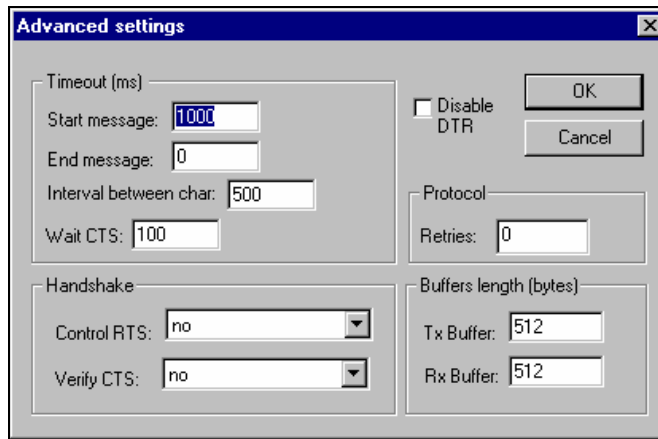
KLOCK: Communication Parameters Dialog

4. Specify the parameters as noted in the following table:

Parameters	Default Values	Valid Values	Description
Station	0	0	Not required for this driver
Data (0-Signed Value 1-Unsigned Value)	0	0 or 1	0: Signed values 1: Unsigned values
Model of Equipment	PS4-201-MM1	PS4-201-MM1 PS316/PS306 PS32	Equipment Type
Swap Word (0 - False 1 -True)	0	0 or 1	Swapping Words for Float Point data type

Note:
 These parameters must be configured *exactly the same* as those you configured for the KLOCK driver in the *Communications Parameters* dialog.

5. Click the **Advanced** button on the *Communication Parameters* dialog to open the *Advanced Settings* dialog:



Advanced Settings Dialog



Notes:

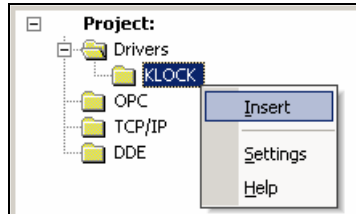
- Do not change any of the default Advanced settings parameters at this time. You can consult the *Studio Technical Reference Manual* for information about configuring these parameters for future reference.
- Generally, you must change these parameters only if you are using a DCE (Data Communication Equipment) converter (such as 232/485), modem, and so forth between your computer, the driver, and the host. However, before adjusting the Advanced communication parameters, you must be familiar with the characteristics of the DCE.

Configuring the Standard 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 *KLOCK* 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):

Description:				
<input type="text" value="Driver Test"/>				<input type="checkbox"/> Increase priority
Read Trigger:	Enable Read when Idle:	Read Completed:	Read Status:	
<input type="text" value="Rd"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	
Write Trigger:	Enable Write on Tag Change:	Write Completed:	Write Status:	
<input type="text" value="Wr"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	
Station:	Header:	<input type="checkbox"/> Min: <input type="text"/>		
<input type="text"/>	<input type="text" value="0"/>	Max: <input type="text"/>		
	Tag Name	Address	Div	Add
1	W[1]	0		
2	W[2]	1		
3	W[3]	2		
4	W[4]	3		
5	F[1]	F1		
6	F[2]	F2		

KLOCK Driver Worksheet

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

Station field: This field may be left blank.

Header field: Address of the operand in the PLC memory. It must be a HEXADECIMAL value corresponding to one word in the marker area or any valid area.

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 **Header** field:

After specifying the **Station** and **Header** parameters, proceed to the next section for instructions on how to configure the **Address** field.

CONFIGURING THE ADDRESS FIELD

The body of the *Driver* worksheet allows you to associate each tag to its respective address in the device. 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:

The address must comply with the following syntax:

<FloatFlag><AddressOffset> (for example: 10)

or

<FloatFlag><AddressOffset>.<Bit> (for example: 10.3)

Where:

- **FloatFlag** is an optional flag. When configured, this parameter must be **F** (Float Point) or **FP** (Float Point, swapping the high word with the low word).
- **AddressOffset** is a parameter added to the **Address Header** parameter (configured in the **Header** field) to compose the address to be read from or written to the device. Must be a decimal number.
- **Bit** is the bit number (from 0 – 15) of the word address. This parameter is *optional*.

➔ **Cautions:**

- In the *Driver* worksheet, the start address is defined by the sum of the Header (from the **Header** field) and the lowest <**AddressOffset**> parameter (in the **Address** field).
- The Bit Write is supported only when executing the “Enable Write On Tag Change” command.
- You must not configure a range of addresses greater than the maximum block size (*data buffer length*) supported by each PLC within the same *Driver* worksheet. The maximum data buffer length for this driver is 32 bytes per *Standard Driver Worksheet*.

Address Configuration Sample		
Device Address	Header Field	Address Field
1	1	0
10	0	10
1020	3E8	20
1020	0	1020

Address Configuration Sample		
Device Address	Header Field	Address Field
10	A	0
11	A	1
12	C	0
10,11	C	F0
10, 11 (with word swap)	C	FS0
10,11	0	F5
10,11	0	FS5
10,11	6	F2
10,11	6	FS2
1	1	0
10	0	10
1020	3E8	20
1020	0	1020
10	A	0
1	1	0
10	0	10
1020	3E8	20
1020	0	1020

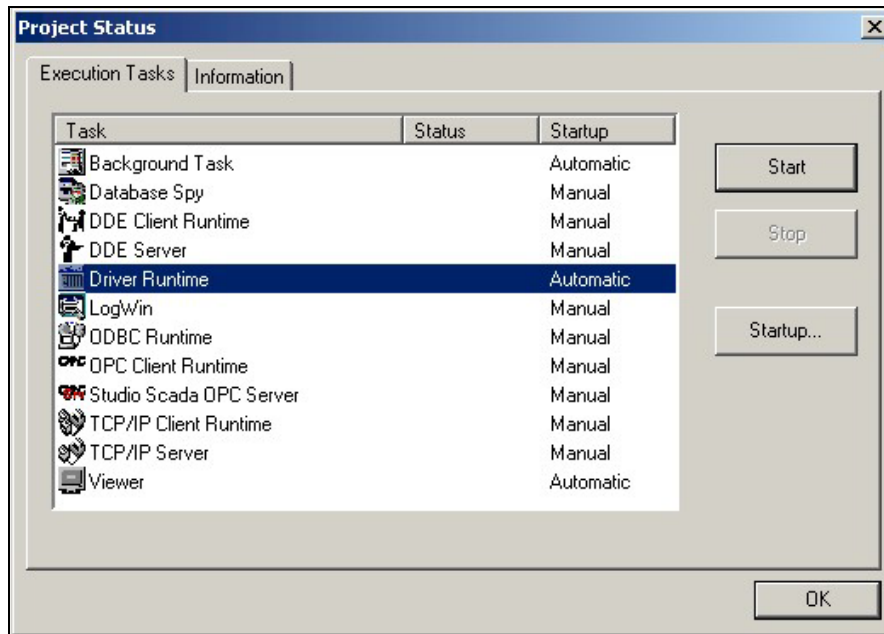
Executing the Driver

After adding the KLOCK 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 displays:



Project Status Dialog

2. 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**.
3. Click **OK** to close the *Project Status* dialog.
4. Start the application to run the driver.

Troubleshooting

If the KLOCK 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.
2	Invalid Memory Address	Invalid Header configured in Driver Sheet.	-
3	Invalid Offset	Invalid Address configured in Diver Sheet	-
4	Send Data Block Exceeded Limit	Range of address configured in Driver Sheet exceeded limit of 32 words	Specify a valid offset or create a new worksheet. Typically, maximum offset is 32.
6	Check Sum Error	Ack action error during communication	<ul style="list-style-type: none"> ▪ Check the cable wiring ▪ Check the station number ▪ Check the RTS/CTS configuration (see <i>Studio Technical Reference Manual</i> for valid configurations).
7	Invalid Number of received bytes	Unexpected length of the answer	Check driver Communication Parameters.
41	Incorrect Message to PLC	Receive Invalid Message	<ul style="list-style-type: none"> ▪ Check the cable wiring ▪ Check the station number ▪ Check the RTS/CTS configuration (see <i>Studio Technical Reference Manual</i> for valid configurations).
42	PLC is not Ready to Receive	PLC is not Ready to receive messages	<ul style="list-style-type: none"> ▪ Check cable wiring ▪ Check the PLC state – it must be RUN. ▪ Check the station number ▪ Check the RTS/CTS configuration (see <i>Studio Technical Reference Manual</i> for valid configurations).
43	Invalid Command	Try write bits without use "Enable write on Tag change"	-
-15	Timeout Start Message	<ul style="list-style-type: none"> ▪ Disconnected Cables ▪ PLC is turned off, in stop mode, or in error mode ▪ Wrong station number ▪ Wrong RTS/CTS control settings 	<ul style="list-style-type: none"> ▪ Check cable wiring ▪ Check the PLC state – it must be RUN. ▪ Check the station number ▪ Check the RTS/CTS configuration (see <i>Studio Technical Reference Manual</i> for valid configurations).
-17	Timeout between rx char	<ul style="list-style-type: none"> ▪ PLC in stop mode or in error mode ▪ Wrong station number ▪ Wrong parity ▪ Wrong RTS/CTS configuration settings 	<ul style="list-style-type: none"> ▪ Check cable wiring ▪ Check the PLC state – it must be RUN. ▪ Check the station number ▪ Check the RTS/CTS configuration (see "Link Characteristics" for 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 enable the log at the unit (**Tools** → **LogWin**) and verify the `ce1og.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


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

- To better understand the information provided in the 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
-	1.01		Jan/1997	▪ Inclusion of automatic offset calculation
-	1.02	Roberto V. Junior	Jan/1997	▪ Inclusion of String Field to define the Equipment in use.
-	1.03	Roberto V. Junior	Apr/1997	▪ Fixed Read of Long Value.
-	1.04	Roberto V. Junior	Mar/1998	▪ Fixed wrong EndAddress in WriteItemStart
-	1.05	Roberto V. Junior	Feb/1999	▪ Included read of short int (signed)
A	1.06	Fabio H. Y. Komura	Feb/2004	▪ Included Float Point Register Type
B	1.07	Fabio H. Y. Komura	Jul/2004	▪ Conformance to Windows CE.