RK512 Communication Driver

Driver for Serial Communication
using RK512 protocol.

Contents

Introduction

The RK512 driver enables Serial communication between the Studio system and devices using RK512 protocol, according to the specifications discussed in this document.

This document will help you to select, configure and execute the RK512 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 RK512 driver in the Studio system.
- **Configuring the Driver**: Explains how to configure the RK512 driver in the Studio system, including how to associate database tags with device registers.
- **Executing the Driver**: Explains how to execute the RK512 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 RK512 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 RK512 driver in Studio and devices.

The information is organized into the following sections:

- Device Specifications
- Network Specifications
- Driver Characteristics

Device Specifications

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

- **Compatible Equipment**:
    - Siemens S5 / S7 3964R
    - Deutz TEM controllers
    - Any device that is compatible with the RK512 protocol

- **Device Runtime Software**: None

Network Specifications

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

- **Device Communication Port**: COM Port
- **Physical Protocol**: Serial
- **Logic Protocol**: 3964R + RK512
- **Specific PC Board**: None

Driver Characteristics

The RK512 driver package consists of the following files, which are automatically installed in the **`/DRV`** subdirectory of Studio:

- **`RK512.INI:`** Internal driver file. *You must not modify this file*.
- **`RK512.MSG:`** Internal driver file containing error messages for each error code. *You must not modify this file*.
- **`RK512.PDF:`** This document, which provides detailed information about the RK512 driver.
- **`RK512.DLL:`** Compiled driver.

> ✍ **Note:**
> You must use Adobe Acrobat® Reader™ to view the **`RK512.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 RK512 driver on the following operating systems:

- Windows 2000/XP/Vista

- Windows CE

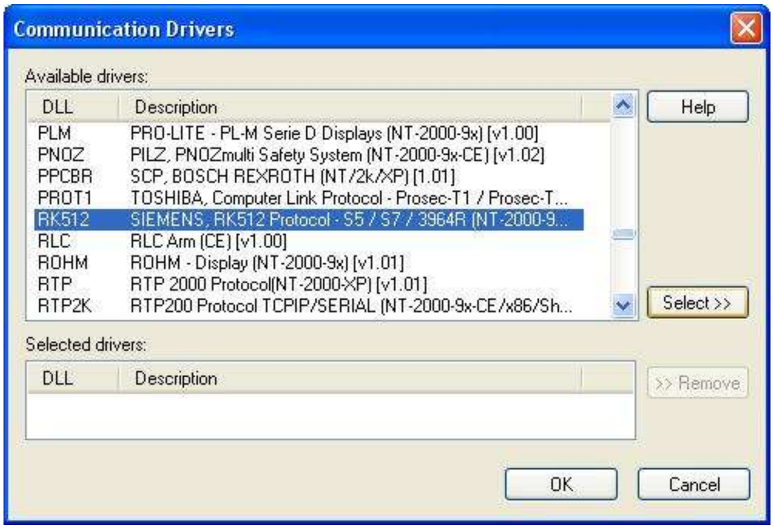
The RK512 driver supports the following registers:

Register Type	Length	Write	Read	Bit	Integer
DB (Data Block)	1 Word	●	●	●	●
C (Counter)	1 Word	●	●	●	●
T (Timer)	1 Word	●	●	●	●
F (Flag)	1 Byte	●	●	●	●
I (Input)	1 Byte	–	●	●	●
Q (Output)	1 Byte	●	●	●	●

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 RK512 driver for your Studio application:

1. From the main menu bar, select **Insert** → **Driver** to open the *Communication Drivers* dialog.
2. Select the **RK512** driver from the *Available Drivers* list, and then click the **Select** button.



Available drivers:

DLL	Description
PLM	PRO-LITE - PL-M Serie D Displays (NT-2000-9x) [v1.00]
PNOZ	PILZ, PNOZmulti Safety System (NT-2000-9x-CE) [v1.02]
PPCBR	SCP, BOSCH REXROTH (NT/2k/XP) [1.01]
PROT1	TOSHIBA, Computer Link Protocol - Prosec-T1 / Prosec-T...
RK512	SIEMENS, RK512 Protocol - S5 / S7 / 3964R (NT-2000-9...
RLC	RLC Arm (CE) [v1.00]
ROHM	ROHM - Display (NT-2000-9x) [v1.01]
RTP	RTP 2000 Protocol(NT-2000-XP) [v1.01]
RTP2K	RTP200 Protocol TCPIP/SERIAL (NT-2000-9x-CE/x86/Sh...

Selected drivers:

DLL	Description

Communication Drivers Dialog

3. When the **RK512** 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.

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

Once you have selected the RK512 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.

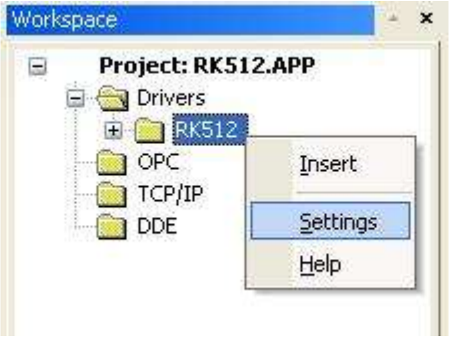
Configuring the Communication Settings

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 RK512 driver-specific settings and procedures will be discussed here. To configure the communication settings for the RK512 driver:

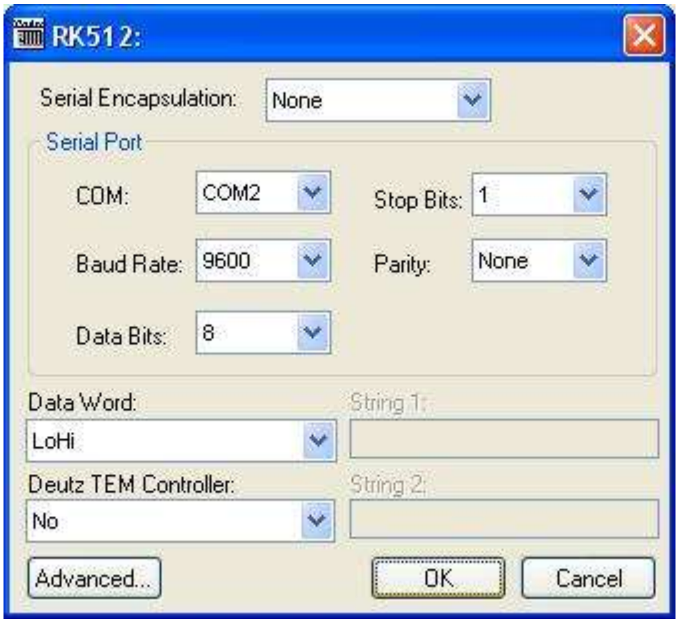
1. In the *Workspace* pane, select the *Comm* tab and then expand the *Drivers* folder. The RK512 driver is listed here as a subfolder.

2. Right-click on the *RK512* subfolder and then select the **Settings** option from the pop-up menu:



Select Settings from the Pop-Up Menu

The *RK512: Communication Settings* dialog is displayed:



RK512:

Serial Encapsulation: None

Serial Port

COM: COM2

Stop Bits: 1

Baud Rate: 9600

Parity: None

Data Bits: 8

Data Word: LoHi

String 1:

Deutz TEM Controller: No

String 2:

RK512: 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:

Parameter	Default Value	Valid values	Description
Data Word	**LoHi**	**LoHi** **HiLo**	Data Word: LoHi: Lo (Byte) Hi (Byte) HiLo: Hi (Byte) Lo (Byte)
DEUTZ TEM Controller	**NO**	**YES or** **NO**	Enable or disable DEUTZ TEM Controller.

✍ **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 RK512 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

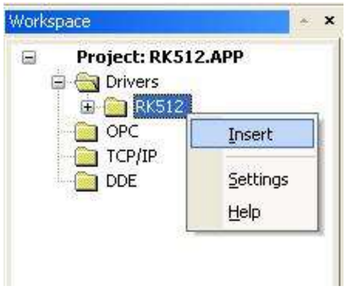
A selected driver includes one or more driver worksheets, which are used to associate database tags in Studio with registers on the target device. Each worksheet is triggered by specific application behavior, so that the tags / registers defined on that worksheet are scanned only when necessary – that 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.

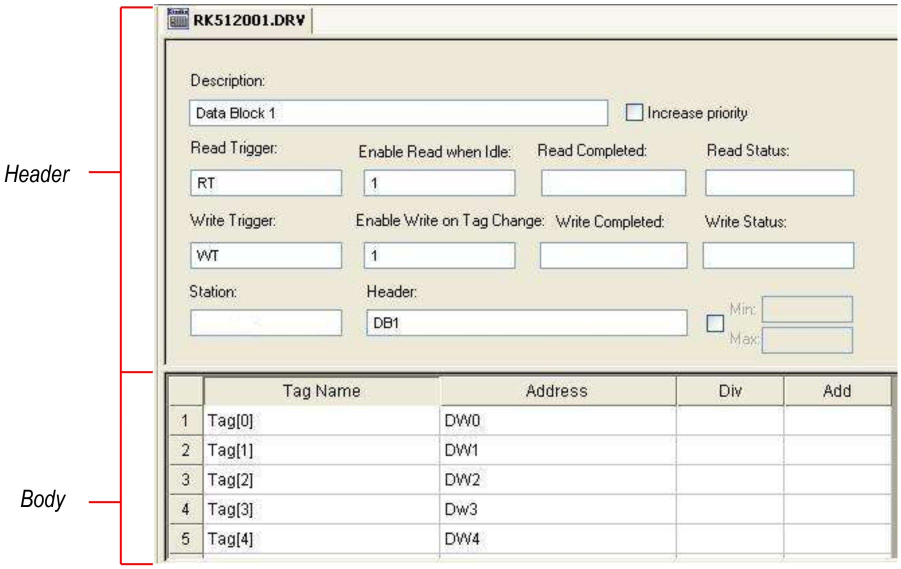
To insert a new driver worksheet:

1. In the *Comm* tab, open the *Drivers* folder and locate the *RK512* subfolder.
2. Right-click on the *RK512* subfolder, and then select **Insert** from the pop-up menu:



Inserting a New Worksheet

A new RK512 driver worksheet is inserted into the RK512 subfolder, and the worksheet is opened for configuration:



RK512001.DRV

Description: Data Block 1 □ Increase priority

Read Trigger:	Enable Read when Idle:	Read Completed:	Read Status:
RT	1		

Write Trigger:	Enable Write on Tag Change:	Write Completed:	Write Status:
WT	1		

Station:
Header: DB1
□ Min: Max:

	Tag Name	Address	Div	Add
1	Tag[0]	DW0		
2	Tag[1]	DW1		
3	Tag[2]	DW2		
4	Tag[3]	Dw3		
5	Tag[4]	DW4		

RK512 Driver Worksheet

> **Note:**
>
> Worksheets are numbered in order of creation, so the first worksheet is **`RK512001.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 RK512 driver.

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

   - **Station** field: Not used for the driver
   - **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:

     `DB <number>`

     Example — **`DB1`**

     `C <number>  (Counter)`

     Example — C1

     `T <number>  (Timer)`

     Example — T0

`D <number>  (Data Byte)`

Example — D0

`DL <number>  (Data Byte Left)`

Example — DL0

`DR <number>  (Data Byte Right)`

Example — DR0

`FW <number>  (Flag Word)`

Example — FW0

`F <number>  (Flag Byte)`

Example — F0

`IW <number>  (Input Word)`

Example — IW0

`I <number>  (Input Byte)`

Example — I0

`Q <number>  (Output Byte)`

Example — Q0

`QW <number>  (Output Word)`

Example — QW0

where:

– ***`<number>`*** is the Data Block number. The valid range depends of the PLC configuration.

After you edit the **Header** field, Studio checks the syntax to determine if it is valid.

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.

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

   <Data Type> <Address>.[Bit]

   Example — **DW1.2**

   Where:

   – ***<Data Type>***: Type of Data Block. Valid values are **DW** (Data Word) and **B** (Byte). If Byte is used the Header should be DB0 for DEUTZ device (Data Block 0 - Data Type Flag).

   – ***<Address>*** : Address of the Data Block.

   – ***[Bit]*** (optional): The bit number of the address. The valid range is 0 up to 15.

   For examples of how Data Block are specified using **Header** and **Address**, see the following table:

Address on the Device	Header	Address
Data Block 1, address number 2, bit number 0	**DB1**	**B2.0**
Data Block 2, address number 0	**DB2**	**DW0**
Data Block 3, address number 10	**DB3**	**B10**

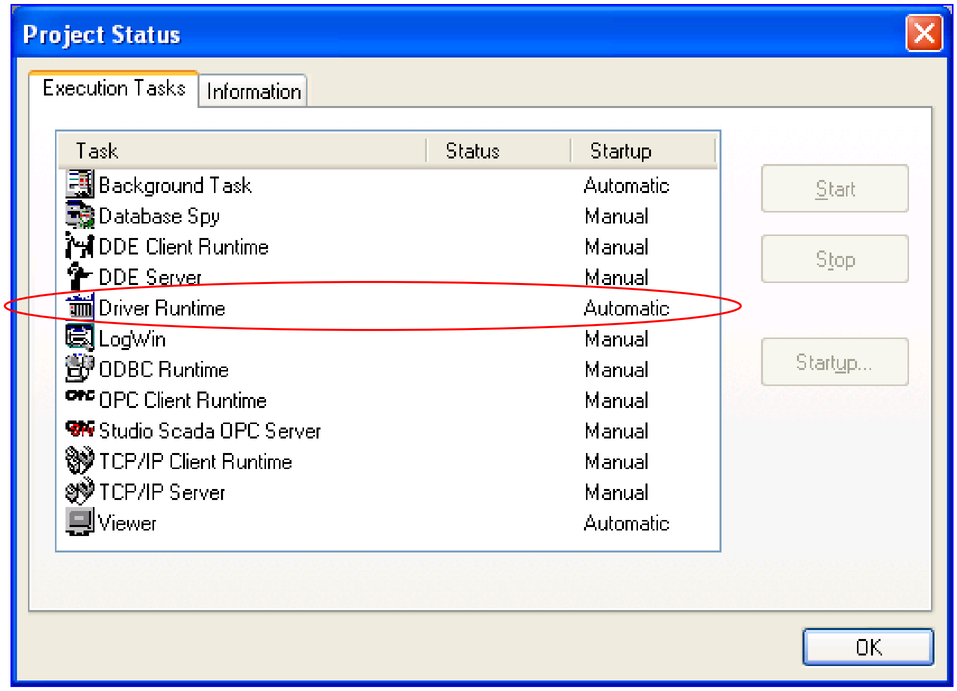
For more information about device Data Block and addressing, please consult the manufacturer's documentation.

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:



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

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 RK512 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 Main 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	None
2	Invalid Header	**Header** field with an invalid value	Check the valid **Header** configuration in the Header field. See "Configuring Driver" session.
3	Invalid Address	**Address** field with an invalid value	Check the valid **Address** configuration in the Address field. See "Configuring Driver" session.
8	Invalid Format Type	**Format Address** with an invalid value	Check the valid **Format Address** configuration in the Address field. See "Configuring Driver" session.
-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. Review the Communication Parameters section for valid RTS/CTS configurations.
-17	Timeout between rx characters.	▪ PLC in stop or error mode ▪ Wrong station number ▪ Wrong parity ▪ Wrong RTS/CTS configuration settings	▪ Check the cable wiring ▪ Check the PLC stat e (it must be RUN) ▪ Check the station number ▪ Check the configuration. Review the Communication Parameters section for valid RTS/CTS configurations.

Note:

For more information about device Error Code, please consult the manufacturer's documentation.

⇨ **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. 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

There was not an official sample application available for this driver by the timer that this document was written.

Revision History

Doc. Revision	Driver Version	Author	Date	Description of Changes
A	1.07	Eric Vigiani Graziane C. Forti	Apr 25 2008	▪ Added information about DEUTZ TEM Controller in the “Communication Parameters”.
B	1.08	Paulo Balbino	Jul 9, 2012	▪ Added new Headers
C	1.09	Charan Manjunath P	Apr 24, 2014	▪ Added table listing register types the driver supports.
C	1.9	Anushree Phanse	May 10 2017	▪ Driver released with version increment.