

RLC Communication Driver

Driver for the Analog Input, Analog Output and Parallel Port of the ARM-GPIO Expansion Board for the RLC-ARM Embedded Systems

Contents

INTRODUCTION	2
GENERAL INFORMATION.....	3
DEVICE CHARACTERISTICS	3
DRIVER CHARACTERISTICS	3
CONFORMANCE TESTING	3
INSTALLING THE DRIVER	4
CONFIGURING THE DRIVER	5
CONFIGURING THE DRIVER WORKSHEET	5
EXECUTING THE DRIVER	7
TROUBLESHOOTING	8
SAMPLE APPLICATION	9
REVISION HISTORY.....	10

Introduction

The RLC driver enables communication between the ARM-GPIO boards with CEView™ system based on the ARM-GPIO Demo, according to the specifications discussed in this publication.

This publication was designed to help you install, configure, and execute the RLC driver to enable communication with the ARM-GPIO board. The information in this publication is organized as follows:

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

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 communication between the Studio RLC driver and ARM GPIO boards.

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:** Flowtronics

For a list of the devices used for conformance testing, see “Conformance Testing” on page 3.

Driver Characteristics

The RLC driver is composed of the following files:

- In the **/DRV** subdirectory:
 - **RLC.INI**: Internal driver file. *You must not modify this file.*
 - **RLC.MSG**: Internal driver file containing error messages for each error code. *You must not modify this file.*
 - **RLC.PDF**: Document providing detailed information about the RLC driver.
 - **RLC.DLL**: Compiled driver.
- In the **/BIN** subdirectory:
 - **RLC_ARM.DLL**: Third-party library file. *You must not modify this file.*

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 **RLC.PDF** document.

You can use the RLC driver on the **Windows CE** operating system only.

Conformance Testing

The following hardware/software was used for conformance testing:

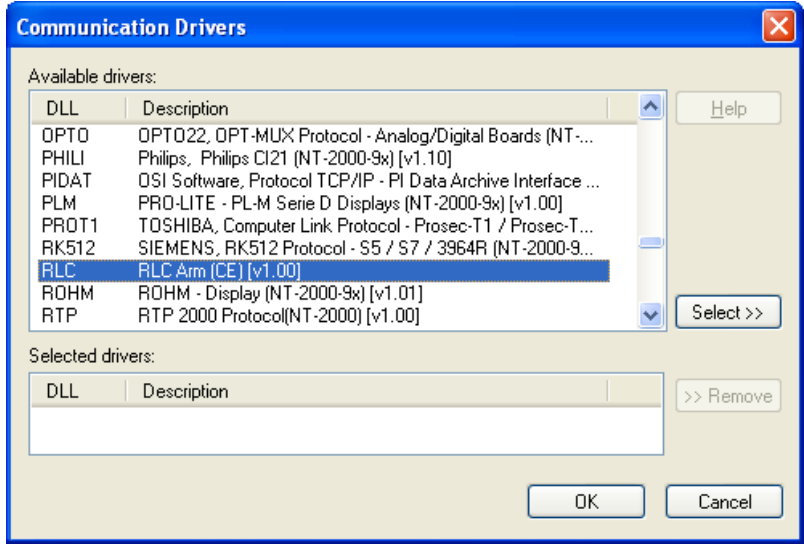
- **Equipment:** Flowtronics HMI
- **Operating System (development):** Windows XP
- **Operating System (runtime):** Windows CE 3.0
- **Studio Version:** 5.1
- **Driver Version:** 1.00

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



Communication Drivers Dialog

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

Attention:
For safety reasons, you must take precautions when you install the physical hardware. Consult the hardware manufacturer's documentation for specific installation instructions.

Configuring the Driver

After opening Studio and selecting the RLC driver, you must configure the driver. Configuring the RLC 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:

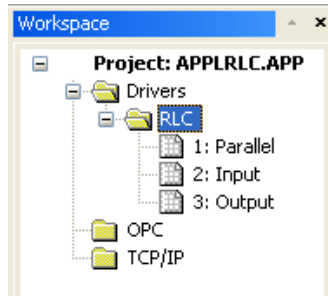
For a detailed description of the Studio *Standard* and *MAIN Driver Worksheets*, and information about configuring the standard fields, review the product's *Technical Reference Manual*.

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 *RLC* subfolder.
3. When the pop-up menu displays, select the **Insert** option:

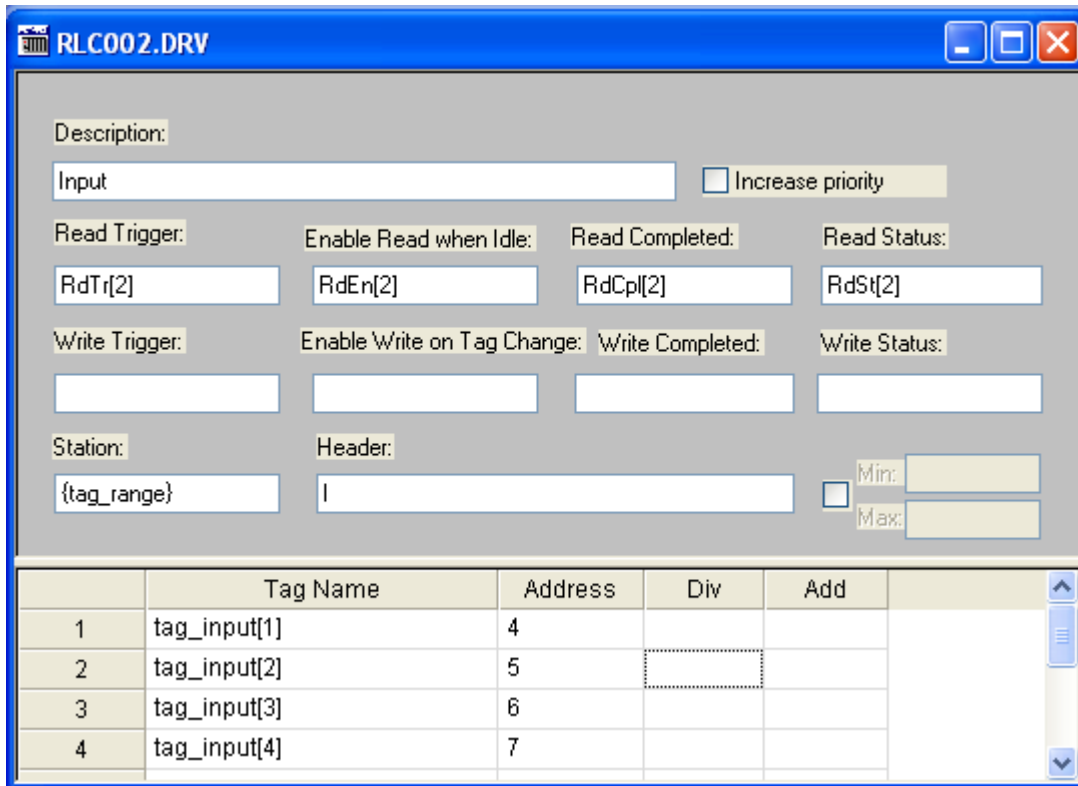


Inserting a New Worksheet

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, it is recommended that you configure the communication addresses in sequential blocks to improve performance.

The <drivername>.drv dialog displays (similar to the following figure):



RLC 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.
 - **Header** field: Use the information in the following table to define the type of variables that will be read from or written to the device and a reference to the initial address. These variables must comply with the following syntax:

<Type>

Where:

Type is the register type (P = Parallel, I = Input, o = Output).

- **Station** field: Use this field to specify the voltage range to read the data. Valid values are 0–3 (0 = 0–5 volts, 1 = ± 5 volts, 2 = 0–10 volts, 3 = ± 10 volts).
- **Address** field: Use the information provided in the following table to associate each tag to its respective device address. For the header Input the valid values are 0–7, for the Outputs 0–3 and for the Parallel port this information is not used.

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.

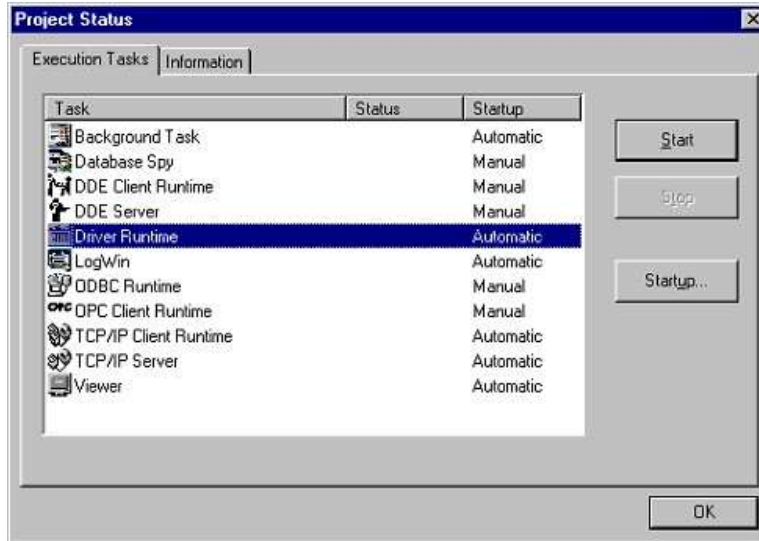
Executing the Driver

After adding the RLC 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 RLC 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
10	Cannot load RLC_ARM.dll	The RLC_ARM.dll is not in the \bin folder of the CEView installation folder	Copy the file RLC_ARM to the \bin folder of the Studio installation folder.
11	Invalid Header	Wrong Header configuration settings	Type a valid header (I, O or P).
12	Invalid Address	Wrong Address	Type a valid address (an integer value).
13	Function not valid for this platform	Trying to run driver on a Windows NT/2000/XP platform	Send the driver to Windows CE platform.

⇒ **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, it is recommended that you use the sample application provided rather than your new application.

If you need 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

Studio provides a sample application in the `/COMMUNICATION EXAMPLES/<Driver Name>` directory. We strongly recommend that you use this sample application to test the RLC 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
A	1.00	Lidiane A. Moreira	19 Jan 04	First version