LLINK Communication Driver

Driver for Communication with
Enersafe LifeLink Devices

Contents

Introduction

The LLINK driver enables communication between the Studio system and Enersafe LifeLink devices using Enersafe proprietary protocol, according to the specifications discussed in this document.

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

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

The information is organized into the following sections:

- Device Characteristics
- Link Characteristics
- Driver Characteristics

Device Characteristics

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

- **Manufacturer**: Enersafe
- **Compatible Equipment**:
  - LifeLink 48T
  - Any Enersafe device that communicates using the same protocol that LifeLink 48T uses
- **Programmer Software**: Battery System Performance Monitor

For a list of the devices used for conformance testing, see "Conformance Testing."

Link Characteristics

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

- **Device Communication Port**: Serial Port
- **Physical Protocol**: RS-232
- **Logic Protocol**: Enersafe Proprietary
- **Device Runtime Software**: None
- **Specific PC Board**: Not Applicable

Driver Characteristics

The LLINK driver is composed of the following files:

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

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

- Windows 9x
- Windows 2000
- Windows NT
- Windows CE

For a list of the operating systems used for conformance testing, see "Conformance Testing" on page 4.

The LLINK driver enables the user to read and write the following information:

Register Type	Length	Write	Read	Bit	Integer	Float	DWord
Battery Voltage	4 bytes	–	•	–	–	•	–
Battery Temperature	4 bytes	–	•	•	–	•	–
Current Probe	4 bytes	–	•	–	–	•	–
System Information (Alarms, Discharge, Temperature, Time, Version, Serial Number, etc.)	-	–	•	•	•	•	–
Historical Logs	-	–	•	–	–	–	–
Device Configuration	-	•	•	–	–	–	–

Conformance Testing

The following hardware/software was used for conformance testing:

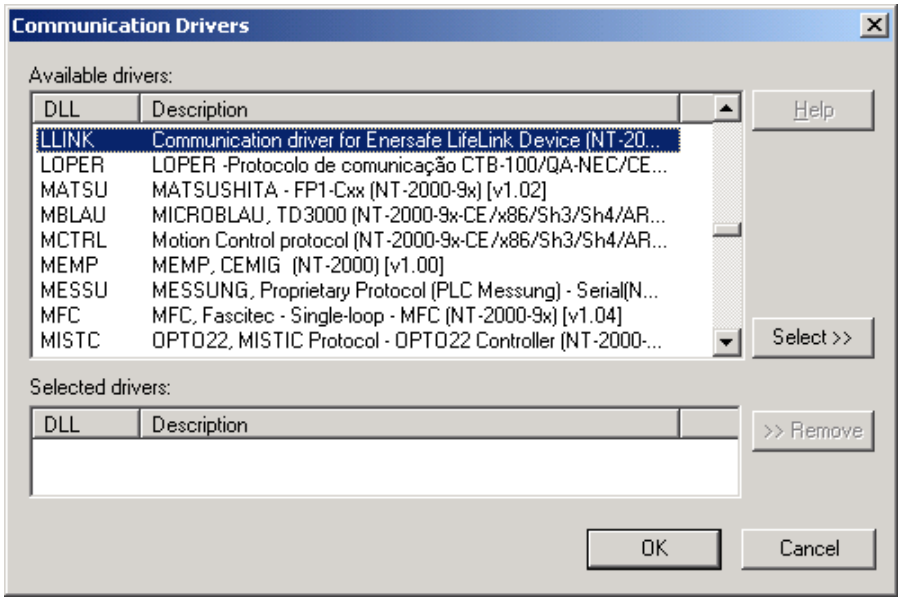
- **Equipment**: LifeLink 48T, Firmware T.2.1
- **Driver Configuration**:
  - **PLC Program**: Not Applicable
  - **Baud Rate**: 9600
  - **Data Bits**: 8
  - **Stop Bits**: 1
  - **Parity**: None
  - **COM Port**: COM1
- **Cable**: Enersafe Serial Port Insulation Adapter
- **Operating System (*development*):** Windows 2000 with Service Pack 4
- **Operating System (*runtime*):** Windows 2000 with Service Pack 4; Windows CE v4.1 – x86 processor
- **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 **LLINK** driver from the *Available Drivers* list, and then click the **Select** button.



Communication Drivers Dialog

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

> ➲ **Attention:**
> 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 Driver

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

- Specifying communication parameters
- Defining tags and controls in the *STANDARD DRIVER SHEET*s (or Communication tables)

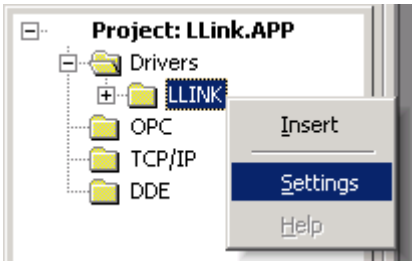
  Worksheets are divided into two sections, a *Header* and a *Body*. The fields contained in these two sections are standard for all communications drivers — except the **Station**, **Header**, and **Address** fields, which are driver-specific. This document explains how to configure the **Station**, **Header**,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

> **Note:**
> For a detailed description of the Studio *STANDARD DRIVER SHEET*s, and information about configuring the standard fields, review the product's *Technical Reference Manual*.

Setting the Communication Parameters

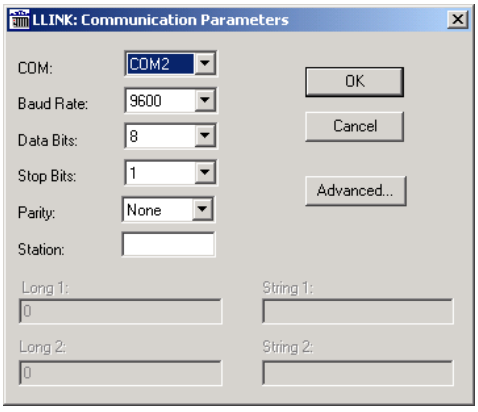
Use the following steps to configure the communication parameters, which are valid for all driver worksheets configured in the system:

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 LLINK subfolder and when the pop-up menu displays, select the **Settings** option.



Select Settings from the Pop-Up Menu

The *LLINK: Communications Parameters* dialog displays:



Communication Parameters Dialog

4. Driver custom parameters: LLINK driver does not have custom parameters.

> **Note:**
> The device must be configured with *exactly the same* parameters that you configured in the *LLINK Communication Parameters* dialog.

5. Click the **Advanced** button on the *Communication Parameters* dialog to open the *Advanced Settings* dialog and configure the settings as necessary.

> **Notes:**
> - Do not change any of the other *Advanced* 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 the *Advanced* parameter settings if you are using a DCE (Data Communication Equipment) converter (232/485 for example), modem, and so forth between the PC, driver, and the host. You must be familiar with the DCE specifications before adjusting these configuration parameters.

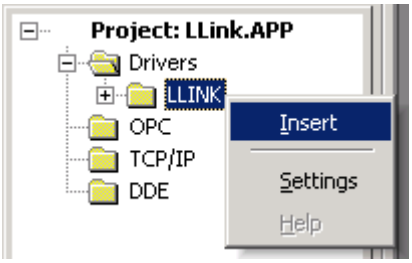
Configuring the Driver Worksheets

This section explains how to configure the *STANDARD DRIVER SHEETs* (or Communication tables) to associate application tags with the device addresses. You can configure multiple *Driver* worksheets — each of which is divided into a *Header* section and *Body* section.

Configuring the *STANDARD DRIVER SHEET*

Use the following steps to create a new *STANDARD DRIVER SHEET*:

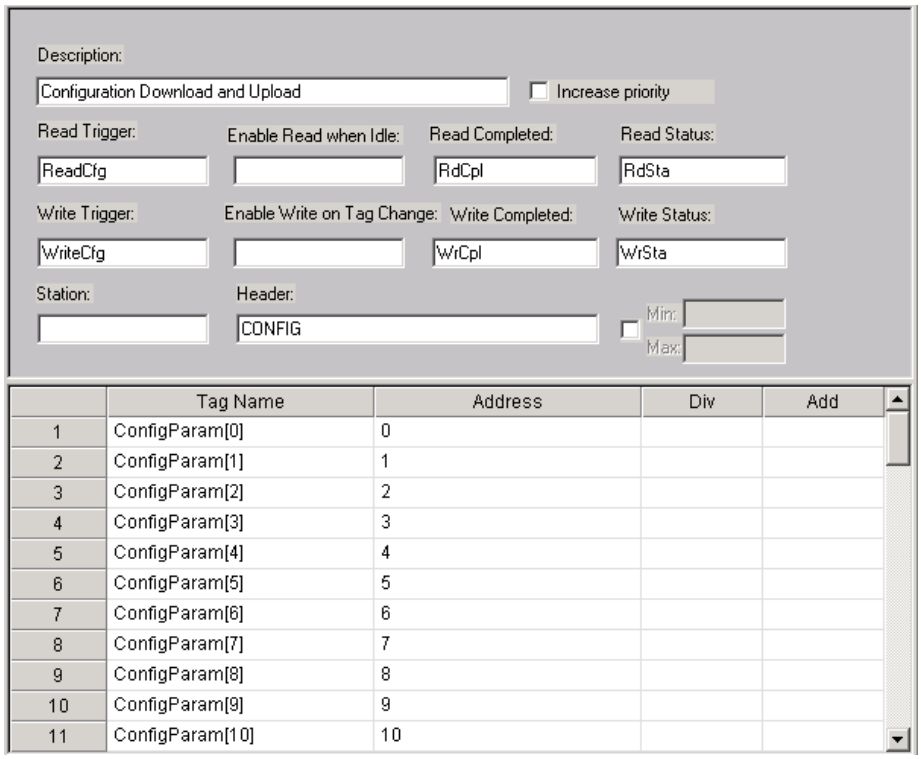
1. From the Studio development environment, select the **Comm** tab, located below the *Workspace* pane.
2. In the *Workspace* pane, expand the *Drivers* folder and right-click the <*Driver Name*> subfolder.
3. When the pop-up menu displays, select the **Insert** option.



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 *STANDARD DRIVER SHEET* displays (similar to the following figure):



	Tag Name	Address	Div	Add
1	ConfigParam[0]	0		
2	ConfigParam[1]	1		
3	ConfigParam[2]	2		
4	ConfigParam[3]	3		
5	ConfigParam[4]	4		
6	ConfigParam[5]	5		
7	ConfigParam[6]	6		
8	ConfigParam[7]	7		
9	ConfigParam[8]	8		
10	ConfigParam[9]	9		
11	ConfigParam[10]	10		

STANDARD DRIVER SHEET

In general, all parameters on the *Driver* worksheet (except the **Station**, **Header**, and **Address** fields) are standard for all communication drivers, but they will not be discussed in this document. 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: Not Applicable
   - **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. (Default value is *BTEMP*)

     These variables must comply with the following syntax:

     `<Data>[:<File name and path>]` (For example: *BTEMP or SNDHLOG:C:\HistoryLog.txt*)

Where:

- **Data** specifies the information that you want to retrieve. (**BTEMP, BVOLT, CPROBE, SYSINFO, ALRSTA, CMD, SNDHLOG, SNDMLOG, SNDDLOG, SNDDDAT, SNDIDAT, CONFIG**)
- **File Name and Path** should be used with the following headers: **SNDHLOG, SNDMLOG, SNDDLOG and SNDIDAT**. These commands save history data in the file specified.

After you edit the **Header** field, Studio checks the syntax to determine if it is valid. If the syntax is incorrect, Studio automatically inserts the default value in the **Header** field.

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 LLINK driver.

Header Field Information			
Data	**Sample Syntax**	**Valid Range of Initial Addresses per Worksheet**	**Address**
Battery Temperature	BTEMP	1-70	The address represents the battery number.
Battery Voltage	BVOLT	1-70	The address represents the battery number.
Current Probe Value	CPROBE	1-10	The address represents the Current probe number.
System Information	SYSINFO	0-14	0 – Voltage 1 – Temperature 2 – Energy 3 – Discharge 4 – Short Discharge 5 – Hour 6 – Minute 7 – Second (always 0 because the device does not return seconds) 8 – Day 9 – Month 10 – Year 11 – Version Number 1 12 – Version Number 2 13 – Version Number 3 14 – Serial Number
Probe Alarm	PROBEALR	1-70	The address represents the Current probe number. The value returned for this field is: ▪ Bit 0 - C/0 = CELL OVERCHARGE ALARM ▪ Bit 1 - C/U = CELL UNDERCHARGE ALARM ▪ Bit 2 - C/D = CELL DISCHARGE ALARM ▪ Bit 3 - FCC = FCC ALARM ▪ Bit 4 - T/R = THERMAL RUNAWAY ALARM ▪ Bit 5 - EQU = EQUIPMENT FAILURE ALARM ▪ Bit 6 - IMP = IMPEDANCE ALARM ▪ Bit 7 - RES = RESERVED
Alarm Status Information	ALRSTA	0-28	0 … 9 – Current probes 1 … 10 10 … 15 – Reserved 16 … 21 – Custom Alarms 1 … 6 22 and 23 – Reserved 24 – System Overcharge 25 – System Undercharge

Header Field Information			
Data	**Sample Syntax**	**Valid Range of Initial Addresses per Worksheet**	**Address**
Alarm Status Information (cont.)	ALRSTA	0-28	26 – System Discharge 27 – Ambient Temperature Alarm 28 – Equipment Failure 29 … 31 – Reserved 32 - Cell Overcharge 33 - Cell Undercharge 34 - Cell Discharge 35 - System Overcharge 36 - System Undercharge 37 - System Discharge 38 - Battery High Temperature 39 - FCC 40 - Ambient Temperature 41 - Cell Impedance 42 - Equipment Failure 43 - Customer (any one of six) 44 - Major rated 45 - Minor rated 46 – spare 47 - spare
Device Configuration	CONFIG	0-58	0 – System Voltage Present (0-Not Present, 1-Present) 1 – Current Transmitter Present (0-Not Present, 1-Present) 2 – Buzzer During Major Alarm (0-No, 1-Yes) 3 – Cell Overcharge Priority (0-Off, 1-Major, -1-Minor) 4 – Cell Undercharge Priority (0-Off, 1-Major, -1-Minor) 5 – Cell Discharge Priority (0-Off, 1-Major, -1-Minor) 6 – System Overcharge Priority (0-Off, 1-Major, -1-Minor) 7 – System Undercharge Priority (0-Off, 1-Major, -1-Minor) 8 – System Discharge Priority (0-Off, 1-Major, -1-Minor) 9 – Float Alarm Priority (0-Off, 1-Major, -1-Minor) 10 – Ambient Alarm Priority (0-Off, 1-Major, -1-Minor) 11 – Thermal Runaway Priority (0-Off, 1-Major, -1-Minor) 12 – Measurement Interval in Days (1-255) 13 – FCC Alarm Priority (0-Off, 1-Major, -1-Minor) 14 – FCC Quantity (0-10) 15 – Impedance Alarm Priority (0-Off, 1-Major, -1-Minor) 16 – Run Impedance Baseline (0-Off, 1-On) 17 – Reserved (1-255) 18 … 23 – Priority customer 1… 6 (0-Off, 1-Major, -1-Minor) 24 – System Temperature Sensor (0-Off, 1-On) 25 – Auto Measurement (0-Disable, 1-Enable) 26 – Temperature Unit (0-Fareneith, 1-Celcius) 27 – Auto CallOut (0-Disable, 1-Enable) 28 – Battery Probe Quantity (0-72) 29 – Current Probe Quantity (0-10) 30 – Batteries per String (0-72) 31 – Cells per Battery (1-255) 32 – Short Discharge in Minutes (1-255) 33 – Shunt Value (0-65535) 34 – Ambient Alarm Threshold (0-250) 35 – Thermal Runaway Threshold (0-250) 36 – Float Level (-100-100)

Header Field Information			
Data	**Sample Syntax**	**Valid Range of Initial Addresses per Worksheet**	**Address**
Device Configuration (cont.)	CONFIG	0-58	37 – Cell Overcharge Threshold (0-10) 38 – Cell Undercharge Threshold (0-10) 39 – Cell Discharge Threshold (0-10) 40 – System Overcharge Threshold (0-60) 41 – System Undercharge Threshold (0-60) 42 – System Discharge Threshold (0-60) 43 – Discharge Level (0-99) 44 – Temperature High Clamp (ANL) (0-250) 45 – Temperature Low Clamp (ANL) (0-250) 46 – Slope (0-1000) 47 – Voltage High Clamp (ANL) (0-1000) 48 – Impedance Threshold (0-100) 49 – FCC Threshold (0-100) 50 – Terminal Phone Number (String maximum 15 characters) 51 – Measurement Time (HH:MM) 52 – Priority Equipment Alarm (0-Off, 1-Major, -1-Minor) 53 – Low Temperature Threshold (0-250) 54 – Low Temperature Histeresis (0-250) 55 – Low Temperature Control (0-Disable, 1-Enable) 56 – High Temperature Threshold (0-250) 57 – High Temperature Histeresis (0-250) 58 – High Temperature Control (0-Disable, 1-Enable)
Alarm History Log Upload	SNDHLOG: C:\AlarmLog.csv	0	Any tag inserted in the worksheet will receive the amount of registers saved in the generated text file. The file is saved according to the following structure: <Alarm Description>, <On or Off>, < battery, string, FCC, customer or system number>, <Time>, <Date>, <Value>. Example: System Overcharge,On,01,01:00,01/03/03,52.586000
System Snapshot Log Upload	SNDMLOG: C:\Measurement.csv	0	Any tag inserted in the worksheet will receive the amount of registers saved in the generated text file. The file is saved according to the following structure: <Time>, <Date>, <Battery Voltage, Battery Temperature >, <Current Probes>, <Total Current>, <System Voltage>, <Ambient Temperature> Note that "Battery Voltage" and "Battery Temperature" will have the information for all the batteries, so you can have more than two values in this field. The same applies for the "Current Probes"
Discharge Event Log Upload	SNDDLOG: C:\Discharge.csv	0	Any tag inserted in the worksheet will receive the amount of registers saved in the generated text file. <Time>, <Date>, <Cumulative Number>, <Discharge Event Duration (seconds)>
Discharge Data Log Upload	SNDDDAT: C:\Discharge.csv	0	Any tag inserted in the worksheet will receive the amount of registers saved in the generated text file. The file is saved according to the following structure: <Time>, <Date>, <Battery Voltage, Battery Temperature >, <Current Probes>, <Total Current>, <System Voltage>, <Ambient Temperature> Note that "Battery Voltage" and "Battery Temperature" will have the information for all the batteries, so you can have more than two values in this field. The same applies for the "Current Probes

Header Field Information			
Data	**Sample Syntax**	**Valid Range of Initial Addresses per Worksheet**	**Address**
Impedance Data Upload	SNDIDAT	0-141	0 – Impedance Battery 1 1 – Error Code Battery 1 2 – Impedance Battery 2 3 – Error Code Battery 2 … 138 – Impedance Battery 70 139 – Error Code Battery 70 140 – Base Line Impedance 141 – Base Line Error Code 70
Send a Command to the device	CMD	RESET CLRHLOG CLRMLOG CLRDLOG RESDNUM DISCONNECT CLRDIS CLRSDIS CLREGY DDATOK EVENTOK LOGIN TRIGON TRIGOFF GOREMOTE GOLOCAL TEST1ON TEST2ON TEST3ON TEST4ON TESTOFF	RESET – Resets the device CLRHLOG – Clear alarms history log in the server CLRMLOG – Clear impedance history log CLRDLOG – Clear discharge event log RESDNUM – Reset discharge event cumulative number DISCONNECT – Forces the server to disconnect phone connection CLRDIS – Clear discharge counter in the server CLRSDIS – Clear short discharge counter in the server CLREGY – Clear energy counter in the server DDATOK – Clear discharge data buffer EVENTOK – Indicate that the alarm who generate call has been registered LOGIN – Request to the server to send his serial number TRIGON - Request the server to turn on the current transmitter TRIGOFF – Request the server to turn off the current transmitter GOREMOTE – Request the server to turn on the internet channel GOLOCAL – Request the server to turn on the local RS-232 channel TEST1ON – Request the server to go in test 1 mode TEST2ON – Request the server to go in test 2 mode TEST3ON – Request the server to go in test 3 mode TEST4ON – Request the server to go in test 4 mode TESTOFF – Request the server to go in test 1 mode

- **Address** field: Use the information in the previous table to associate each tag to its respective device address. The table below has some examples of address configuration:

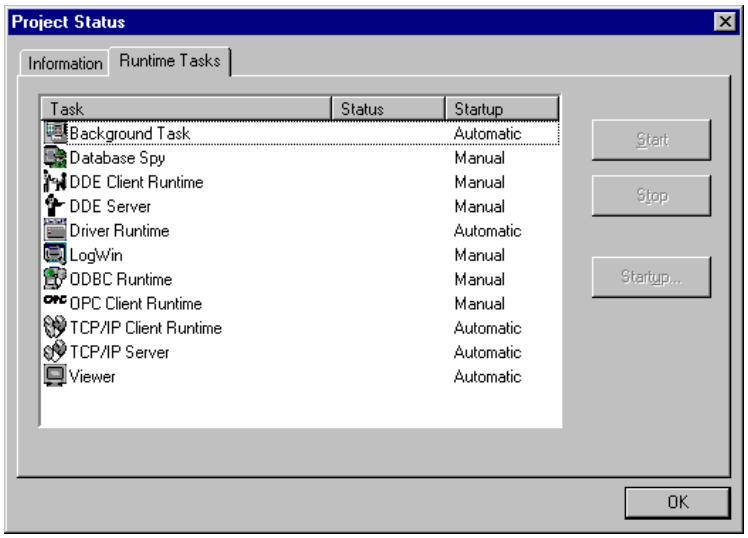
Address Configuration Sample		
Device Address	**Header Field**	**Address Field**
Temperature Battery 1	BTEMP	1
Temperature Battery 10	BVOLT	10
Current from probe 2	CRPROBE	2
Reset the device	CMD	RESET

Executing the Driver

After adding the LLINK 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 LLINK 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.
1	Out of memory	The computer where studio is running does not have enough memory available for the driver buffer	Increase the system memory
2	Invalid Header field	Invalid tag value in the Header field.	Specify a valid tag value in the Header field.
3	Invalid Address field	Invalid Address	▪ Check the initial address in the Driver Worksheet. ▪ Check the Holding register in the Driver Worksheet with bit configuration. This parameter cannot execute write triggers—it executes "Write on Tag Change" only. ▪ Retype the address in the Driver Worksheet.
4	Invalid Operation	The operation requested (read or write) is not valid for the header configured. This problem can happen if the user configures a CMD header and try to perform a read operation.	▪ Change the header or the operation to make them compatible.
5	Invalid Response	The response received from the device is not valid	▪ Verify the cables and the baud rate selected.
6	Invalid File	The file name passed in the header is not valid	▪ Verify if the file name does not contain any invalid character (?, *, etc.) ▪ If you specified subfolders, verify if they exist.
7	Invalid Parameter	When reading the configuration from the device, the driver expects parameters from 0 to 58. If by some reason, the device sends a parameter greater than 58, the driver will return this error code.	▪ Verify the firmware version that you are using and the one specified in the "Conformance Testing" section. If the one that you are using is newer, it can have new parameters that the driver does not expect to receive.
8	Invalid Configuration	One of the configuration parameters specified is not valid and the device responded configuration error.	▪ Verify the firmware version that you are using and the one specified in the "Conformance Testing" section. ▪ Check the values that you are passing to all the parameters. You can try to read the configuration and send it back to the device in order to test the command.
-15	Timeout Start Message	▪ Disconnected Cables ▪ PLC is turned off, in stop mode, or in error mode ▪ Wrong station number ▪ Wrong RTS/CTS control settings	▪ Check cable wiring ▪ Check the PLC state – it must be RUN. ▪ Check the station number ▪ Check the configuration. See *Studio Technical Reference Manual* for information about valid RTS/CTS configurations.
-17	Timeout between rx char	▪ PLC in stop mode or in error mode ▪ Wrong station number ▪ Wrong parity ▪ Wrong RTS/CTS configuration settings	▪ Check cable wiring ▪ Check the PLC state – it must be RUN. ▪ Check configuration. ▪ Check the configuration. See *Studio Technical Reference Manual* for information about 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 use the *Remote LogWin* of Studio (**Tools** → **Remote LogWin**) to get the log events from the target unit remotely.

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**.
- **Studio version**: To find this information, select **Help** → **About.**
- **Driver Version**: To find this information, read the full description of the driver on the *Communication Drivers* Dialog.
- **Communication Log**: Displays in the Studio *Output* window (or *LogWin* window) when the driver is running. Be sure to enable the **Field Read Commands**, **Field Write Commands**, and **Serial Communication** for the *LogWin* window.
- **Device Model** and **Boards**: Consult the hardware manufacturer's documentation for this information.

Sample Application

You will find a sample application for drivers in the **/COMMUNICATION EXAMPLES/<*Driver Name*>** directory. We strongly recommend that you check if there is a sample application for this driver and use it to test the 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.

> **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	Lourenço Teodoro	Aug/19/2003	First driver version
B	1.1	Paulo Balbino	Sep/16/2011	Changed Doc version