SITIA

Contents

SITIA Driver

SITIA SIEMENS S7-1200/S7-1500 Driver (version 1.7, last revised Dec 05 2018).

The SITIA driver enables communication between the Studio system and remote devices using the Siemens S7 Protocol protocol, according to the specifications discussed in this document.

This document assumes that you have read the "Development Environment" section in the main Studio documentation.

This document also assumes that you are familiar with the Microsoft Windows 7/8/10 environment. If you are not familiar with Windows, then we suggest using the **Help and Support** feature (available from the Windows **Start** menu) as you work through this document.

Driver specifications

This section identifies all of the software and hardware components required to implement communication between the SITIA driver in Studio and remote devices using the Siemens S7 Protocol protocol.

Driver files

The SITIA driver package comprises the following files, which are automatically installed in the `Drv` folder of the Studio application directory:

- `SITIA.DLL`: Compiled driver.
- `SITIA.INI`: Internal driver file. *You must not modify this file.*
- `SITIA.MSG`: Internal driver file defining error messages for the possible error codes. (These error codes are described in detail in the Troubleshooting section.) *You must not modify this file.*
- `SITIA.PDF`: This document, which provides complete information about using the driver.

> **Note:** You must use a compatible PDF reader to view the `SITIA.PDF` file. You can install Acrobat Reader from the Studio installation CD, or you can download it from Adobe's website.

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

- Windows 7/8

Device specifications

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

- Manufacturer: Siemens
- Compatible Equipment: S7-1500 PLCs, S7-1200 PLCs with firmware v4.0 and higher
- Programmer Software: Siemens TIA portal v13 and higher

Network specifications

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

- Device Communication Port: 102
- Physical Protocol: Ethernet
- Logic Protocol: Siemens S7 Protocol
- Device Runtime Software: None
- Specific PC Board: None
- Cable Wiring Scheme: Regular Ethernet cable

Additional specifications

For using SITIA driver v1.4 and lower:

To ensure successful communication between this driver and the PLC program, use Siemens TIA Portal (i.e., the PLC programming software) to do the following:

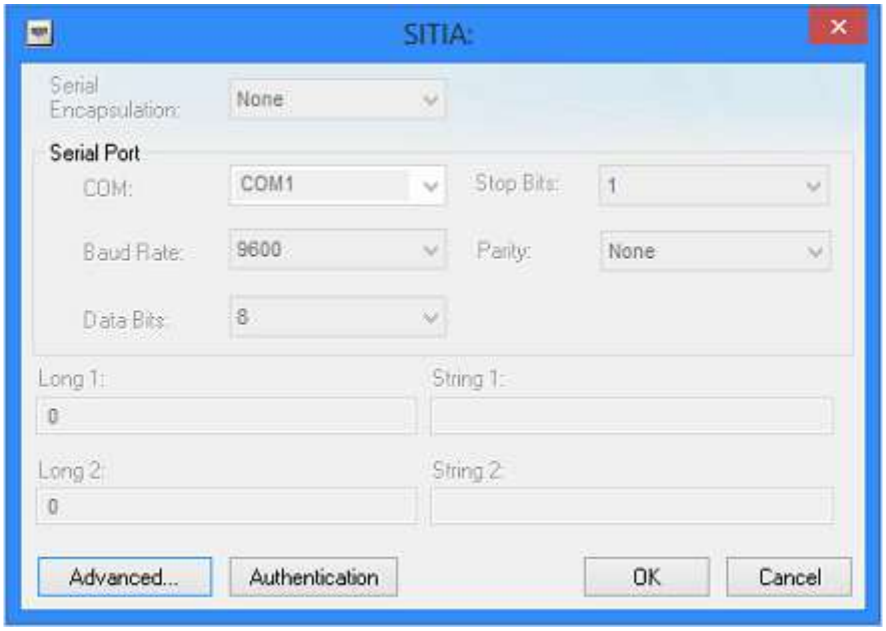
1. Make sure the program is not password protected. In other words, in the program's **Protection** properties, set the access level to **Full access (no protection)**, with HMI, Read, and Write access enabled for all users.
2. Also in the program's **Protection** properties, under **Connection mechanisms**, select **Permit access with PUT/GET communication from remote partner (PLC, HMI, OPC, ...)**.
3. Make sure **Visible in HMI** and **Accessible in HMI** are selected for all program tags that you want this driver to be able to read/write.

4. Whenever new tags are created in the program, recompile the program and then download it to the target device.

For using SITIA driver v1.5 and higher with support for authentication:

To ensure successful communication between this driver and the PLC program, use Siemens TIA Portal (i.e., the PLC programming software) to do the following:

5. Make sure you know the password for the program that password protected. In other words, in the program's **Protection** properties, set the access level to **No access (complete protection)**, with HMI, Read, and Write access enabled for appropriate levels and with specific passwords
6. Also in the program's **Protection** properties, under **Connection mechanisms**, the **Permit access with PUT/GET communication from remote partner (PLC, HMI, OPC, ...)** will be grayed out after any password is enabled.
7. Make sure **Visible in HMI** and **Accessible in HMI** are selected for all program tags that you want this driver to be able to read/write.
8. In the driver settings of SITA, Select Authentication and enter in the station field the IP of the PLC exactly as would be entered in the Main Driver Sheet. In the user use any generic value like user, user1 etc. since there is no user field in this protocol spec of the Siemens PLCs (this interface is used by some other drivers). In the password field specify the appropriate password for any specific level of access as specified on the PLC. If different access levels have different passwords and will be used by the driver, create multiple entries with the same station IP and different users/passwords.



Serial Encapsulation: None

Serial Port

COM: COM1

Stop Bits: 1

Baud Rate: 9600

Parity: None

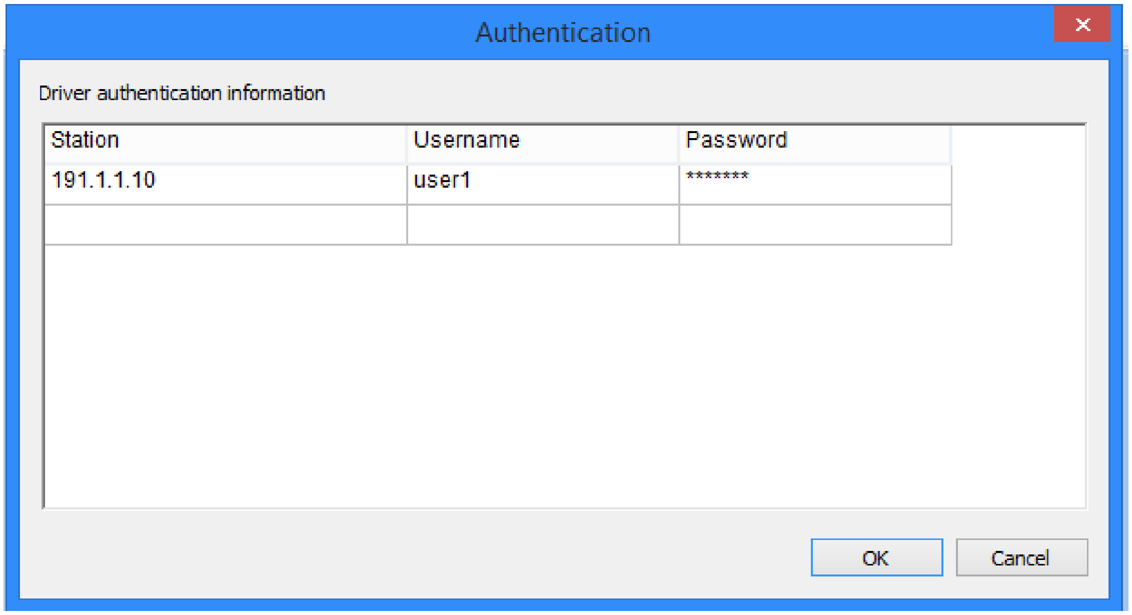
Data Bits: 8

Long 1: 0

String 1:

Long 2: 0

String 2:



Driver authentication information

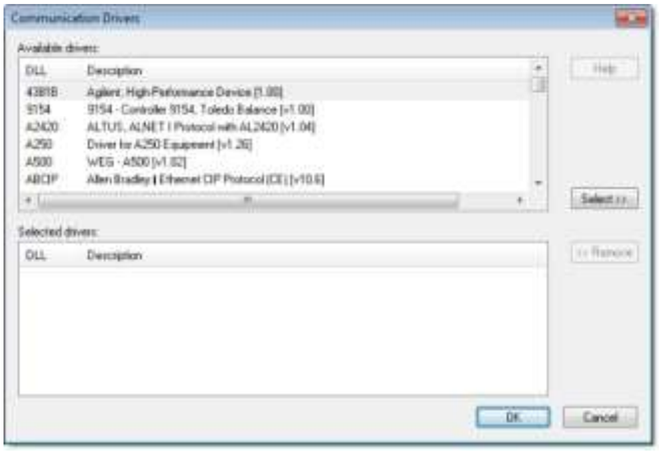
Station	Username	Password
191.1.1.10	user1	*******

9. Whenever new tags are created in the program, recompile the program and then download it to the target device.

Adding a communication driver to your project

This section explains how to add a communication driver to your project.

1. On the **Insert** tab of the ribbon, in the **Communication** group, click **Add/Remove Driver**.
   The *Communication Drivers* dialog is displayed.



Available drivers:

DLL	Description
43818	Agilent; High-Performance Device [1.00]
9154	9154 - Controller 9154, Toledo Balance [v1.00]
A2420	ALTUS, ALNET I Protocol with AL2420 [v1.04]
A250	Driver for A250 Equipment [v1.26]
A500	WEG - A500 [v1.02]
ABCIP	Allen Bradley	Ethernet CIP Protocol (CE) [v10.6]

Selected drivers:

DLL	Description

Communication Drivers dialog

2. In the *Available drivers* list, click the communication driver that you want to add.
3. Click **Select**.
   The driver is added to the *Selected drivers* list.
4. Click **OK**.
   The *Communication Drivers* dialog is closed and the selected driver is inserted in the **Drivers** folder in the Project Explorer.

About driver worksheets

Like the other parts of your project, communication with remote devices is controlled by worksheets. This section explains how to add worksheets to your project and then configure them to associate project tags with device registers.

Each selected driver includes a Main Driver Sheet (MDS) and one or more Standard Driver Sheets (SDS). The Main Driver Sheet is used to define tag/register associations and driver parameters that are in effect at all times, regardless of project behavior. In contrast, Standard Driver Sheets can be inserted to define tag/ register associations that are triggered by specific project behaviors.

The configuration of these worksheets is described in detail in the "Communication" chapter of the *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 SITIA driver-specific parameters and procedures are discussed here.

Adding and configuring a Standard Driver Sheet

By default, a communication driver does not include any Standard Driver Sheets. This section explains how to add a Standard Driver Sheet to your project and then configure it.

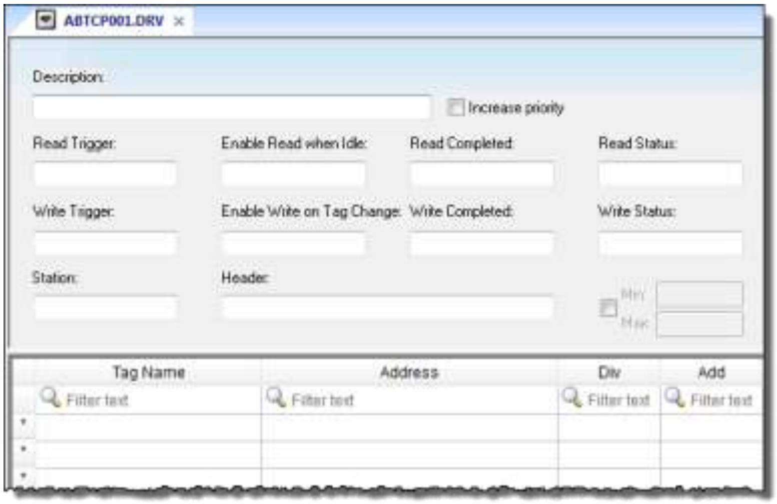
The SITIA driver must be added to the project before you can configure any of its worksheets. For more information, see Adding a communication driver to your project on page 6.

Standard Driver Sheets can be inserted to define additional tag/register associations that are triggered by specific project behaviors.

> **Note:** Most of the settings on this worksheet are standard for all drivers; for more information about configuring these settings, see the “Communication” chapter of the *Technical Reference Manual*. The **Station** and **I/O Address** fields, however, use syntax that is specific to the SITIA driver.

1. Do one of the following.
   - On the **Insert** tab of the ribbon, in the **Communication** group, click **Driver Sheet** and then select **SITIA** from the list.
   - In the **Comm** tab of the Project Explorer, right-click the **SITIA** folder and click **Insert** on the shortcut menu.

   A new SITIA driver worksheet is inserted into the **SITIA** folder, and then it is automatically opened for configuring.



Standard Driver Sheet

> **Note:** Worksheets are numbered in order of creation, so the first worksheet is `SITIA001.drv`.

2. Configure the Station and Header fields as described below.

Station

Specify the IP Address of the PLC, port number is optional. Examples are given in the table below.

IP Address : [*Port Number*]

Where:

IP Address: is the IP address of the device. (This value is mandatory)

[*Port Number*]: This is the port number for the SITIA protocol (default: 102). (Optional)

You can also specify a tag in curly brackets to change the station during the runtime (e.g. `{Station}`), but the tag that is referenced must follow the same syntax and contain a valid value.

Station Formats

Station	Examples
IP Address:[*Port Number*]	192.1.168.125 192.1.168.125:102

Header

The only header supported is Header TAG

3. For each tag/register association that you want to create, insert a row in the worksheet body and then configure the row's fields as described below.

Tag Name

Type the name of the project tag.

Address

Specify the tag name you want to communicate with. Where: *tag name* is the name of the tag configured on the PLC via TIA Portal

> **Note:** Tag names can contain alphanumeric characters and "_" (underscore).Other characters in the tag name may not communicate successfully.

> **Note:** Each Standard Driver Sheet can have up to 4096 rows. However, the **Read Trigger**, **Enable Read When Idle**, and **Write Trigger** commands attempt to communicate the entire block of addresses that is configured in the sheet, so if the block of addresses is larger than the maximum block size that is supported by the driver protocol, then you will receive a communication error (e.g., "invalid block size") during run time. Therefore, the maximum block size imposes a practical limit on the number of rows in the sheet.

For examples of how device registers are specified using **Header** and **Address**, see the following table.

Examples of Address fields in Standard Driver Sheet

Address on Studio
motorSpeedRPM
PLC_Tag
currentMotor[100]
WaterLevelTank2000
Tank[5].waterLevel.Average
Datablock_1.Classtag.Array[1].temperatureRoom1

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

4. Save and close the worksheet.

Configuring the Main Driver Sheet

When you add the SITIA driver to your project, the Main Driver Sheet is automatically included in the **SITIA** folder in the Project Explorer. This section describes how to configure the worksheet.

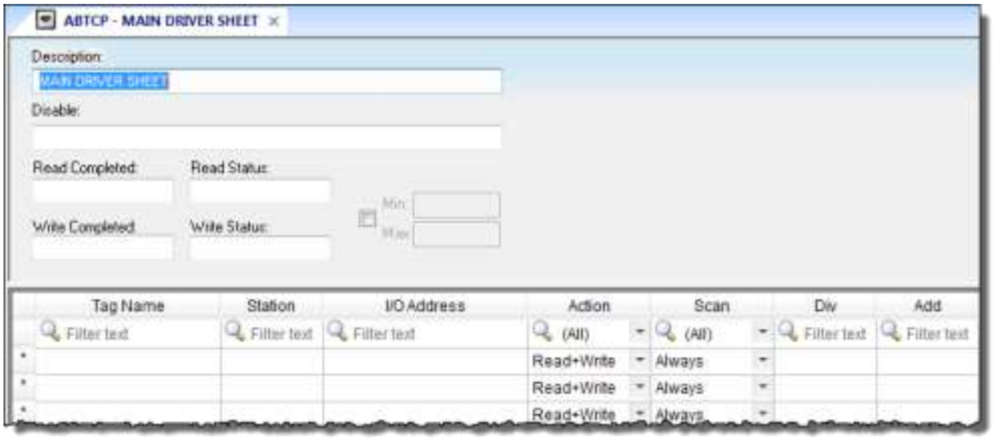
The SITIA driver must be added to the project before you can configure any of its worksheets. For more information, see Adding a communication driver to your project on page 6.

The Main Driver Sheet is used to define tag/register associations and driver parameters that are in effect at all times, regardless of project behavior. The worksheet is continuously processed during project runtime.

> **Note:** Most of the settings on this worksheet are standard for all drivers; for more information about configuring these settings, see the “Communication” chapter of the *Technical Reference Manual*. The **Station** and **I/O Address** fields, however, use syntax that is specific to the SITIA driver.

1. Do one of the following.
   - On the **Insert** tab of the ribbon, in the **Communication** group, click **Main Driver Sheet** and then select **SITIA** from the list.
   - In the **Comm** tab of the Project Explorer, expand the **SITIA** folder and then double-click **MAIN DRIVER SHEET**.

   The Main Driver Sheet is displayed.



Description: MAIN DRIVER SHEET

Tag Name	Station	I/O Address	Action	Scan	Div	Add
Filter text	Filter text	Filter text	(All)	(All)	Filter text	Filter text
			Read+Write	Always		
			Read+Write	Always		
			Read+Write	Always		

Main Driver Sheet

2. For each tag/register association that you want to create, insert a row in the worksheet body and then configure the row's fields as described below.

   Tag Name

   Type the name of the project tag.

   Station

   Please see station for standard driver sheet.

   I/O Address

   There is no supported header for the Main driver sheet and the address is *tag name*

> **Note:** The Main Driver Sheet can have up to 32767 rows. If you need to configure more than 32767 communication addresses, then either configure additional Standard Driver Sheets or create additional instances of the driver.

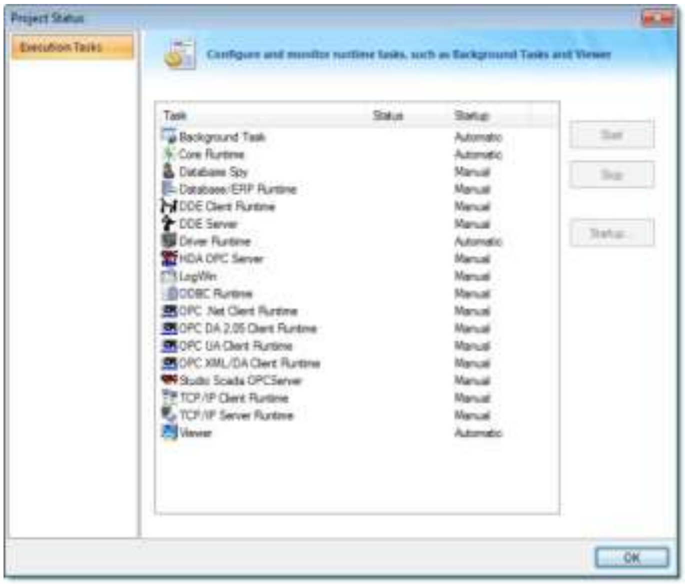
3. Save and close the worksheet.

Checking the Driver Runtime task

This section describes how to check the status of the Driver Runtime task in the list of execution tasks.

The Driver Runtime task handles communication with remote devices and the processing of the driver worksheets. By default, the task is configured to start up automatically when the project is run, but you can check it for yourself.

1. On the **Home** tab of the ribbon, in either the **Local Management** or the **Remote Management** group (depending on where you project server will be running), click **Tasks**.
   The *Project Status* dialog is displayed.



Configure and monitor runtime tasks, such as Background Tasks and Viewer

Task	Status	Startup
Background Task		Automatic
Core Runtime		Automatic
Database Spy		Manual
Database/ERP Runtime		Manual
DDE Client Runtime		Manual
DDE Server		Manual
Driver Runtime		Automatic
HDA OPC Server		Manual
LogWin		Manual
ODBC Runtime		Manual
OPC .Net Client Runtime		Manual
OPC DA 2.05 Client Runtime		Manual
OPC UA Client Runtime		Manual
OPC XML/DA Client Runtime		Manual
Studio Scada OPCServer		Manual
TCP/IP Client Runtime		Manual
TCP/IP Server Runtime		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 the next step.

- If the **Driver Runtime** task is set to **Manual**, select the task and then click **Startup** to change the task to **Automatic**.

3. Click **OK** to close the *Project Status* dialog.

Troubleshooting

This section lists the most common errors for this driver, their probable causes, and basic procedures to resolve them.

Checking status codes

If the SITIA driver fails to communicate with the target device, then the database tag(s) that you configured for the **Read Status** and **Write Status** fields of the driver sheets will receive a status code. Use this status code and the following tables to identify what kind of failure occurred and how it might be resolved. **Status codes for the driver**

Error	Description	Possible Causes	Procedure To Solve
0	OK	Communication without problems.	None required
1	Invalid Station	Invalid Station specified on the driver sheet.	Check the correct format on this manual.
2	Error communicating with device	PLC offline, problem with connections	Check connections, check if PLC is set to Run mode.
3	Failed resolving tags, check the tags configured (Check Output Window for more information)	Some of specified tags are not available on the PLC.	Check the output window for detailed information on the invalid tags.
4	Error reading	PLC offline, problem with connections	Check connections, check if PLC is set to Run mode.
5	Internal Error	-	Contact Support
6	Error writing	Tag is read only	Check the tag specified on the TIA Portal

Common status codes

Status Code	Description	Possible Causes	Procedure To Solve
0	OK	Communicating without error.	None required.
-15	Timeout waiting for message to start	• Disconnected cables. • PLC is turned off, in stop mode, or in error mode. • Wrong station number. • Wrong parity (for serial communication). • Wrong RTS/CTS configuration (for serial communication).	• Check cable wiring. • Check the PLC mode — it must be RUN. • Check the station number. • Increase the timeout in the driver's advanced settings. • Check the RTS/CTS configuration (for serial communication).

Status Code	Description	Possible Causes	Procedure To Solve
-33	Invalid driver configuration file	The driver configuration file (*drivername*.INI) is missing or corrupt.	Reinstall the driver.
-34	Invalid address	The specified address is invalid or out of range.	Check the supported range of addresses described in this document, and then correct the address.
-35	Driver API not initialized	The driver library was not initialized by the driver.	Contact technical support.
-36	Invalid data type	The specified data type is invalid or out of range.	Check the supported data types described in this document, and then correct the data type.
-37	Invalid header	The specified header in the driver worksheet is invalid or out of range.	Check the supported range of headers described in this document, and then correct the header.
-38	Invalid station	The specified station in the driver worksheet is invalid or out of range.	Check the supported station formats and parameters described in this document, and then correct the station.
-39	Invalid block size	Worksheet is configured with a range of addresses greater than the maximum block size.	Check the maximum block size number of registers described in this document, and then configure your driver worksheet to stay within that limit. Keep in mind that you can create additional worksheets. ☞ **Note:** If you receive this error from a Main Driver Sheet or Tag Integration configuration, please contact Technical Support.
-40	Invalid bit write	Writing to a bit using the attempted action is not supported.	• Writing to a bit using Write Trigger is not supported in some drivers. Modify the driver worksheet to use Write On Tag Change. • The bit is read-only.
-42	Invalid bit number	The bit number specified in the address is invalid. The limit for the bit number depends on the registry type.	Check the addresses to see if there are bit numbers configured outside the valid range for the registry.
-43	Invalid byte number	The byte number specified in the address is invalid. The limit for the byte number depends on the registry type.	Check the addresses to see if there are byte numbers configured outside the valid range for the registry.
-44	Invalid byte write	Writing to a byte using the attempted action is not supported.	The byte is read-only or inacessible.
-45	Invalid string size	The string is more than 1024 characters.	Modify the addresses that have string data type to be less than 1024 characters.

Monitoring device communications

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 to establish an event log on a remote unit that runs Windows Embedded. 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.

Contacting Technical Support

If you must contact Technical Support, please have the following information ready:

- **Operating System** and **Project Information**: To find this information, click **Support** in the **Help** tab of the ribbon.
- **Driver Version** and **Communication Log**: Displays in the *Output* window (LogWin module) when the driver is enabled and the project is running is running.
- **Device Model** and **Boards**: Consult the hardware manufacturer's documentation for this information.

Revision history

This section provides a log of all changes made to the driver.

Revision history

Driver Version	Revision Date	Description of Changes	Author
1.0	Jan, 25, 2016	• First driver revision	Paulo Balbino
1.0	Apr, 11, 2016	•Improved driver documentation: Added Additional Specifications, modified station configuration and examples. No changes in driver.	Anushree Phanse
1.1	July, 05, 2016	• Fixed the bug of tags on read showing an incorrect timestamp.	Anushree Phanse
1.2	August, 08, 2016	• Fixed the issue of driver crashing when trying to communicate with an invalid station IP address.	Anushree Phanse
1.3	February, 09, 2017	• Added support to communicating with S7-1500 firmware 2.0	Anushree Phanse
1.4	August, 22, 2017	• Fixed issue of reading LREALS with poor precision and now they read as 64 bit reals	Anushree Phanse
1.5	March 27, 2018	• Added support for S7-1200 PLCs which have firmware v4.0 and higher. • Added support for authentication when communicating with PLCs which use different password protected access levels.	Anushree Phanse

1.6	June 21, 2018	• Added support for simultaneous requests using Advanced Settings	Anushree Phanse
1.7	Dec 05, 2018	Fixed SITIADBExporter.dll (present in the installer) to correctly find tag names with spaces present in them when using S7-1200/S7-1500 tag Integration	Anushree Phanse