

DOMOR

Contents

DOMOR Driver	3
Driver specifications.....	4
Configuring the device's communication settings	Error! Bookmark not defined.
Adding a communication driver to your project	5
About driver worksheets	5
<i>Adding and configuring a Standard Driver Sheet</i>	5
<i>Configuring the Main Driver Sheet</i>	9
Checking the Driver Runtime task.....	10
Troubleshooting	11
Revision history.....	16

DOMOR Driver

DOMOR Driver for Automation Direct Do-More PLC (version 1.4, last revised 29 Aug 2017).

The DOMOR driver enables communication between the Studio system and remote devices using the Do-more 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 XP/Vista/7 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 DOMOR driver in Studio and remote devices using the Do-more protocol.

Driver files

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

- `DOMOR.DLL`: Compiled driver.
- `DOMOR.INI`: Internal driver file. *You must not modify this file.*
- `DOMOR.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.*
- `DOMOR.PDF`: This document, which provides complete information about using the driver.



Note: You must use a compatible PDF reader to view the `DOMOR.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 DOMOR driver on the following operating systems:

- Windows 7/8/2008/2012

Device specifications

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

- Manufacturer: Automation Direct
- Compatible Equipment: Do-more H2 Series PLCs
- Programmer Software:

Network specifications

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

- Device Communication Port: UDP (28784)
- Physical Protocol: Ethernet UDP/IP
- Logic Protocol: Do-more
- Device Runtime Software: None
- Specific PC Board: None
- Cable Wiring Scheme: Regular Ethernet cable

Conformance testing

The following hardware/software was used for conformance testing:

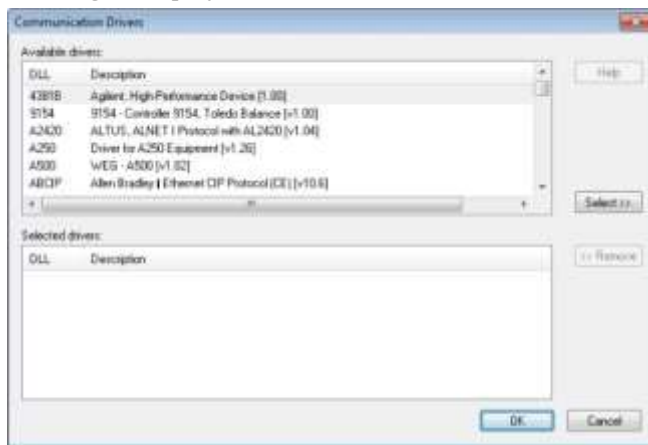
- Driver Version: 1.4
- Studio Version: 8.0+SP2+P1
- Operating System (development): Windows 7/8
- Operating System (target): Windows 7/8/2008/2012
- Equipment: H2-DM1E
- Communication Settings:
- UDP/IP Port: 28784

- Cable: Regular Ethernet cable

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.



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 DOMOR 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.

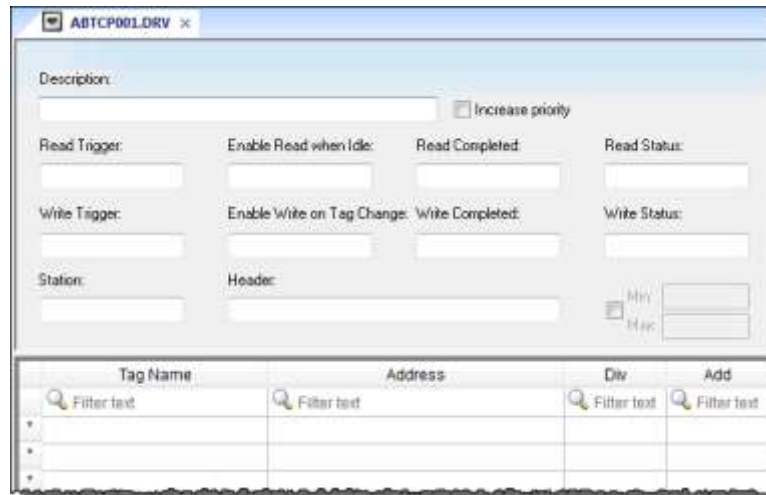
The DOMOR driver must be added to the project before you can configure any of its worksheets. For more information, see [Adding a communication driver to your project](#) on page 6.

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 DOMOR driver.

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

A new DOMOR driver worksheet is inserted into the **DOMOR** 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 `DOMOR001.drv`.

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

Station

Specify the station in the driver sheet using the following syntax. Note that this field cannot be left empty. Examples are given in the table below. **[CSV Filename]:<IP**

Address>[:PortNumber][:Password] Where:

[CSV Filename] This is the name of the file exported by the Do-more designer. This is required only when accessing the user defined blocks, heap items, or nicknames. If you only specify the file name or a folder/filename, the driver will consider the path relative to the Studio project path. You can export the file on the Do-More designer by clicking on **File** and choosing **Export**

and then **Element Documentation**. On the new window enable the check-box of **C-More Do More Driver Format** (without STRUCT fields) and click on **Save** .(This is an optional parameter).

<IP Address> is the IP Address of the device on the Ethernet network.

[PortNumber] This is the port number. If not specified, the driver will use the default UDP port 28784. (This is an optional parameter)

<Password> is the password of the user logged on the device. This parameter is required if you configure a password for the PLC (This is an optional parameter)

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
[CSV Filename:]<IP Address>[:PortNumber][:Password]	192.168.110.101 192.168.110.101:28785 192.168.110.101:28785:123456 192.168.110.101:28785:abcdef TagFile.csv:192.168.110.101:28785:abcdef

Header

The header field is not in use by this driver.

- 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 variable you want to communicate with. The element can be a memory block, a heap item or a nickname. Please notice that user defined memory blocks or heap items and nicknames will require the CSV file in the station field.

- Accessing Atomic Memory Blocks (non-structure):**

They are accessed by specifying the name of the block followed by an index variable. The index should be within the valid range of values configured for that memory block. User defined blocks require the CSV file in Station field. The syntax is as follows:

<BlockName><Index>

Where:

<BlockName> is the name of the memory block which can be either a System memory block or a User defined block.

<Index> index of the element to be accessed.

- Accessing Structure Memory Blocks :**

They are accessed by specifying the name of the block followed by an index and an element. The index should be within the valid range of values configured for that memory block. User defined blocks require the CSV file in Station field. The syntax is as follows:

<BlockName><Index>.<ElementName>

Where:

<BlockName> is the name of the memory block which can be either a System memory block or a User defined block.

<Index> index of the element to be accessed.

<ElementName> is the name of the element in the structure that is being accessed.

- **Accessing Heap Items:**

They are accessed by giving the name of the heap item and the name of the element of the structure to be accessed, without giving any index. User defined heap items require the CSV file in Station field. The syntax is as follows:

<HeapItemName>.<ElementName>

Where:

<HeapItemName> is the name of the Heap item being accessed which can be either a Built-in or User defined item.

<ElementName> is the name of the element in the structure that is being accessed.

- **Accessing by Nicknames:**

Nicknames can be created on the programming software for any element that can be accessed. Nicknames require the CSV file in Station field. The syntax is as follows:

<NickName>

Where:

<NickName> is the nickname assigned to the element being accessed.

- **String Structure:**

The String structure can be created by the programming software and it can be accessed by its name.


<StringName><Index>.<Optional ElementName>

Where:

<StringName> is the name assigned to the element being accessed.

<Index> index of the element to be accessed.

<Optional ElementName> is the name of the element in the structure that is being accessed. For example for String we have the MaxLen and the Length elements.

 **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 Header and Address fields in Standard Driver Sheet

Controller Address	Address on Studio
D10	D10
V90	V90
\$Main.RunCounter	\$Main.RunCounter
T1.Acc	T1.Acc
TestHeap.Acc	TestHeap.Acc
NickName1	NickName1

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

The DOMOR 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 DOMOR 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 **DOMOR** from the list.
- In the **Comm** tab of the Project Explorer, expand the **DOMOR** folder and then double-click **MAIN DRIVER SHEET**.

The Main Driver Sheet is displayed.

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

- 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

Please see address field for the standard driver sheet.

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.

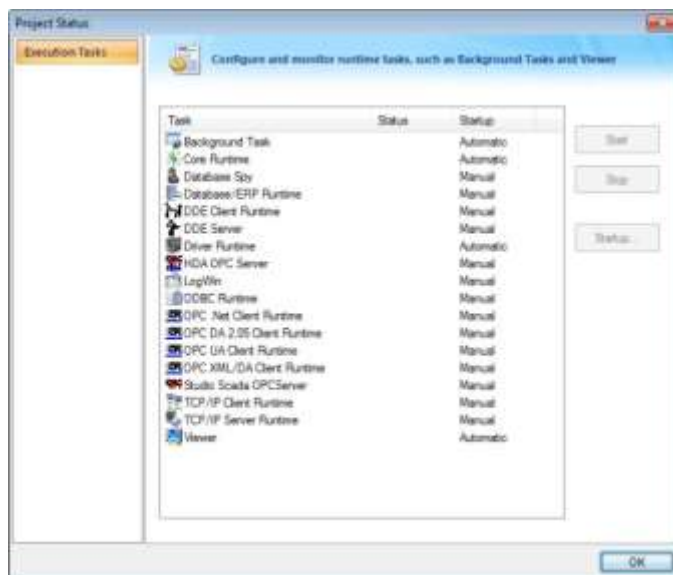
- Save and close the worksheet.

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.

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



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 DOMOR 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
1	Unknown Command	The command sent is not correct or protocol version is mismatched or the message is corrupted.	Check the versions and update. If they are correct, check the cabling, routing and switches for bad packets.
2	Out of Sessions	Too many devices connected.	Reduce the number of devices or restart the driver.
3	Illegal Operation	Not sufficient permission levels for the operation performed.	Increase the permission level for the user on the device.
4	Invalid Session	The session number does not match the sending device.	Re-establish connection or restart the driver and try again.
5	Out of Range	Invalid address accessed	Ensure that the correct address is given and is within range specified. Or expand the address range on PLC and load configuration.

6	Invalid Argument	Message corrupted or not parsed properly. Could occur from noise or faulty wiring.	Check the parameters sent.
7	Program Update Active	Program Update is running	Wait until Program Update is complete.
8	No Token	Attempted to update the program without first acquiring the program update token.	Wait until Program Update is complete.
9	Program Update Inhibited	Client attempted to update the program when ST21 is TRUE. This allows the customer to use the program to prevent the project from being updated.	-
10	System Configuration Update Active	System Configuration Update Active is running	Wait till it is complete to continue communications.
11	Invalid Mode	The mode is not valid.	Ensure that the switch on the CPU is in Term mode
12	Mode Change Active	Occurs when a PLC mode change is attempted while a mode change is in progress.	In some cases it takes several scans for a mode change. So, wait till the mode change is complete.
13	Mode Locked	Occurs when mode change is attempted and keyswitch is not in Term.	Ensure that the switch on the CPU is in Term mode
14	Invalid Password	The password sent does not exist on the PLC.	Check the password and send the correct one.
15	Resource Locked	Trying to update a tag that is forced.	Force must be removed in order to update the tag.
16	Doc Update Active	Attempted to access the documentation file while it is being written back to ROM.	Wait till it is done.
17	Invalid Driver	Error while reading	-
18	Invalid Driver Data	Error while reading	-
30	Invalid Transaction ID. The ID received does not match the ID sent.	Invalid Station	Check the station used.
31	NAK received. The Request sent is not acknowledged by the PLC	Invalid Station	Check the station used.
32	Invalid MX App Protocol Function. The code on response received does not match the code on Request sent.	Possible problem on the CSV file or on the variables on the PLC.	Check if the CSV variables matches with the PLC variables.
33	Invalid Group for Reading	The read group is not valid	Check the manual for the valid ranges and addresses.
34	Invalid Address	Invalid Address	Check the manual for valid address.
5	UDP Timeout Error. Unable to connect probably due to an Invalid Station.	PLC connections, invalid station or invalid port number.	Check the PLC connections and verify the station format. Also verify the port number used.

36	Unable to Parse CSV File. Check if the file exists in the correct path and is valid.	Check if the file exists in the correct path and is valid	Check if the file exists in the correct path and is valid
37	Out-of-Sync. The version of the CSV file does not match with the PLC	The version of the CSV file does not match with the PLC	Update the PLC program and the CSV file.
38	PLC is being uploaded	PLC is being uploaded	Wait until the communication is reestablished.

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	<ul style="list-style-type: none"> • 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). 	<ul style="list-style-type: none"> • 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).
-33	Invalid driver configuration file	The driver configuration file (<i>drivername</i> .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.

Status Code	Description	Possible Causes	Procedure To Solve
			 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.	<ul style="list-style-type: none"> 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 inaccessible.
-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.
-56	Invalid connection handle	The connection is no longer valid.	Please contact Technical Support.
-57	Message could not be sent	The socket was unable to send the TCP or UDP message.	<ul style="list-style-type: none"> Check the station IP address and port number. Confirm that the device is active and accessible. Try to ping the address.
-58	TCP/IP could not send all bytes	The TCP/IP stack was not able to send all bytes to destination.	<ul style="list-style-type: none"> Check the station IP address, port number and/or ID number. Confirm that the device is active and accessible. Try to ping the address.
-60	Error to establish TCP/IP connection	Error while establishing a TCP/IP connection with the slave device. Possibly incorrect IP address or port number in the specified station.	<ul style="list-style-type: none"> Check the station IP address, port number and/or ID number. Confirm that the device is active and accessible. Try to ping the address.
-61	TCP/IP socket error	The TCP/IP connection has been closed by the device.	Confirm that the device is active and accessible. Try to ping the address.
-62	UDP/IP receive call returned error	The UDP socket is in error.	<ul style="list-style-type: none"> Check the station IP address, port number and/or ID number. Confirm that the device is active and accessible. Try to ping the address.
-63	UDP/IP error initializing	The UDP socket initialization failed.	Confirm that the operating system supports UDP sockets.

-64	UDP/IP receive call returned error	The UDP socket is in error.	<ul style="list-style-type: none"> • Check the station IP address, port number and/or ID number. • Confirm that the device is active and accessible. • Try to ping the address.
-65	UDP/IP bind error, port number may already be in use	The driver was not able to bind the UDP port.	<ul style="list-style-type: none"> • Check the port number used by the driver. • Check for other programs that might be bound to the UDP port.

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	Aug 20, 2014	<ul style="list-style-type: none">• First driver revision	Paulo Balbino
1.1	Sep 11, 2014	<ul style="list-style-type: none">• Documentation revised.	Eduardo Castro
1.2	Oct 20, 2014	<ul style="list-style-type: none">• Fixed the Invalid Trasaction ID issue.• Fixed issue where CSV file and PLC versions were not always compared.• Fixed issue where a BAD tag makes the entire sheet tags go BAD on the Standard Driver Sheet.	Felipe Andrade
1.3	May 9, 2017	<ul style="list-style-type: none">• Fixed issue of driver creating incorrect blocks while communicating causing the invalid block size error.	Anushree Phanse
1.4	Aug 29, 2017	<ul style="list-style-type: none">• Fixed validation of groups created to avoid communicating with an invalid block size error.	Anushree Phanse