

ABB Modbus TCP driver for JMobile

This document contains the information needed
to connect the panel to ABB controllers using the
Modbus TCP protocol.

Copyright © 2011 Sitek S.p.A. – Verona, Italy

Subject to change without notice

The information contained in this document is provided for informational purposes only. While efforts were made to verify the accuracy of the information contained in this documentation, it is provided “as is” without warranty of any kind.

Third-party brands and names are the property of their respective owners.

Contents

ABB Modbus TCP Driver	4
Protocol editor settings.....	4
Tag import	6
Special Data Types	7
Communication Status	8
Implementation details	8

ABB Modbus TCP Driver

Various Modbus TCP-capable devices can be connected to the operator panel. To set-up your Modbus TCP device, please refer to the documentation you have received with the device.

This specific implementation of the Modbus TCP driver provides easy handling of the connection to the ABB controllers providing specific supports for PLC models and tag import facilities.

The implementation of the protocol operates as a Modbus TCP client only.

Protocol editor settings

Add (+) a driver in the Protocol editor and select the protocol called “ABB Modbus TCP” from the list of available protocols.

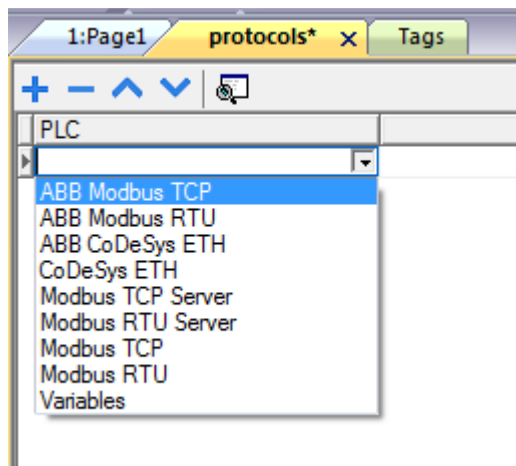


Figure 1

The driver configuration dialog is showed in the following figure.

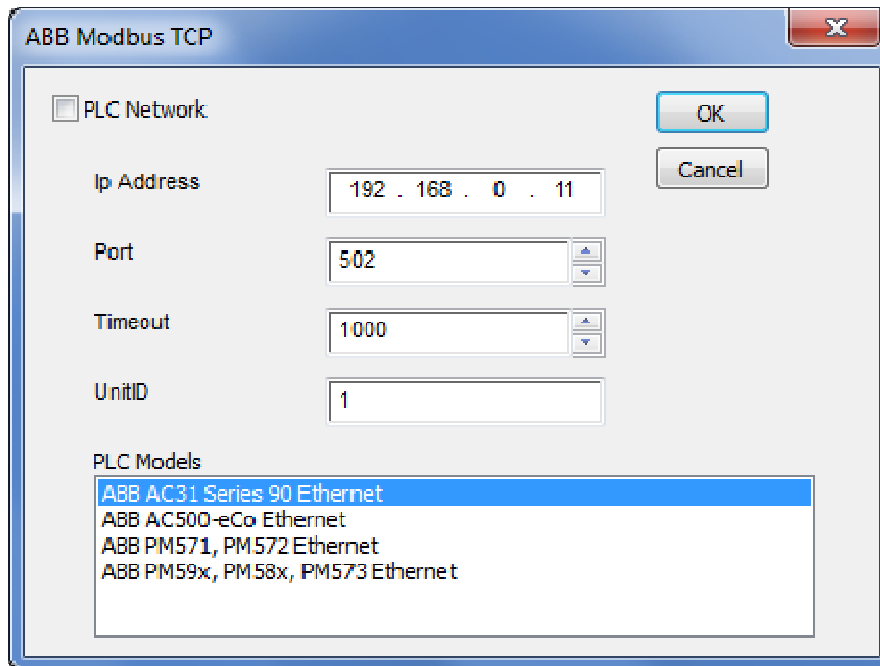


Figure 2

IP address	Ethernet IP address of the controller
Port	Allows to change the default port number used by the Modbus TCP driver; it could be useful whenever the communication passes through Routers or Internet gateways where the default port number is already in use
Timeout (ms)	Defines the time inserted by the protocol between two retries of the same message in case of missing response from the server device. It is expressed in milliseconds.
Unit ID	The Unit ID is rarely used and in most cases can be left zero. The value of the Modbus ID is simply copied into the Unit Identifier field of the Modbus TCP communication frame. Usually it is used when communicating over Ethernet-to-Serial bridges and is then representing the Slave ID.
PLC Models	The list allows to select the PLC model you are going to connect to. The selection will influence the data range offset per each data type according to the specific PLC memory resources.
PLC Network	The protocol allows the connection of multiple controllers to one operator panel. To set-up multiple connections, check "PLC network" checkbox and enter IP Address for all controllers.

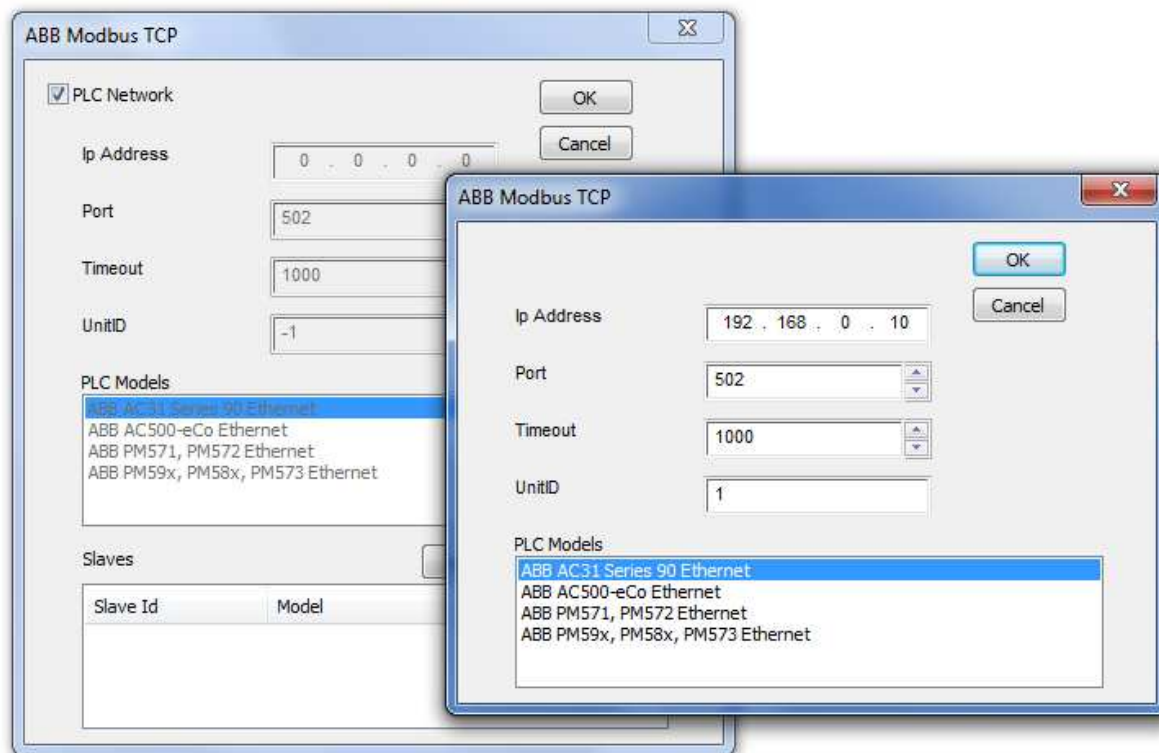


Figure 3

Tag import

This special Modbus driver supports tag import.

The ABB controllers are programmable with a programming tool called ControlBuilder which is based on the CoDeSys V2.3 soft PLC.

The tag importer supports the CoDeSys export file in “.exp” format.

In the CoDeSys programming software the Export command is available under the Project menu as shown in the following figure.

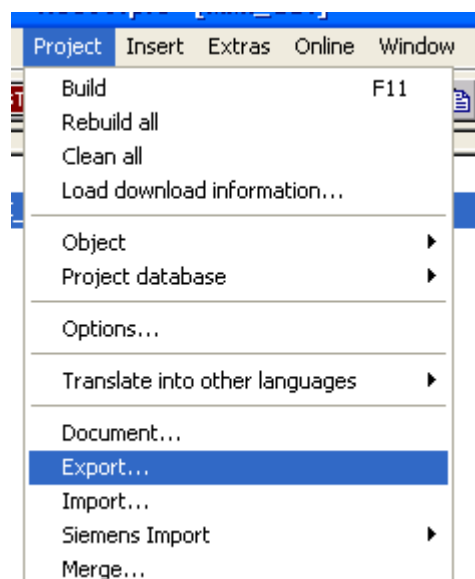


Figure 4

Select the driver in the PB610 Panel Builder 600 Studio tag editor and click on the “Import tag” button to start the importer.

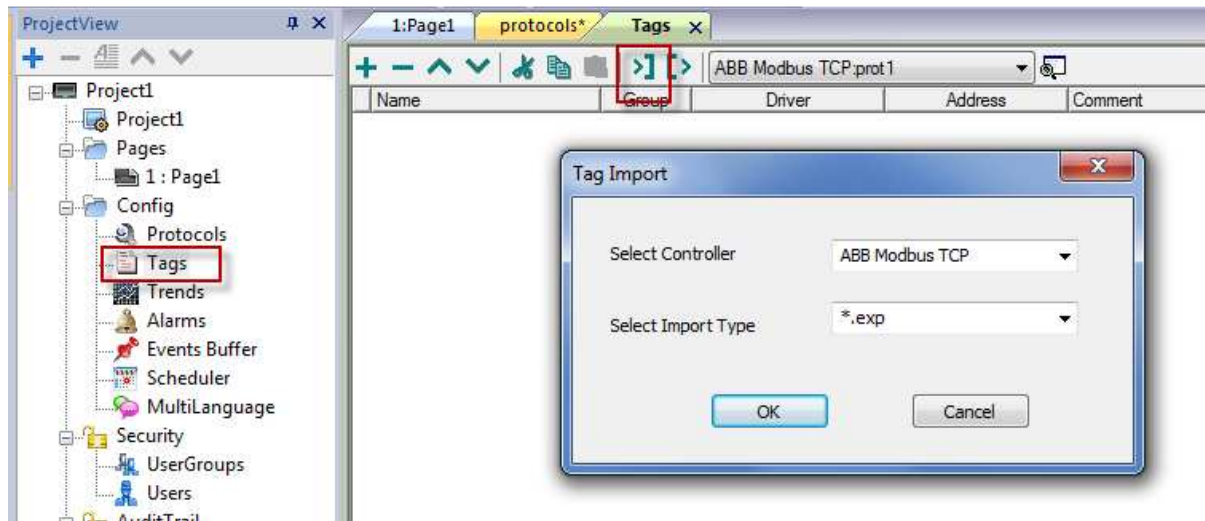


Figure 5

Locate the “.exp” file and confirm.

The tags present in the exported document are listed in the tag dictionary from where they can be directly added to the project using the add tags button as shown in the following figure.

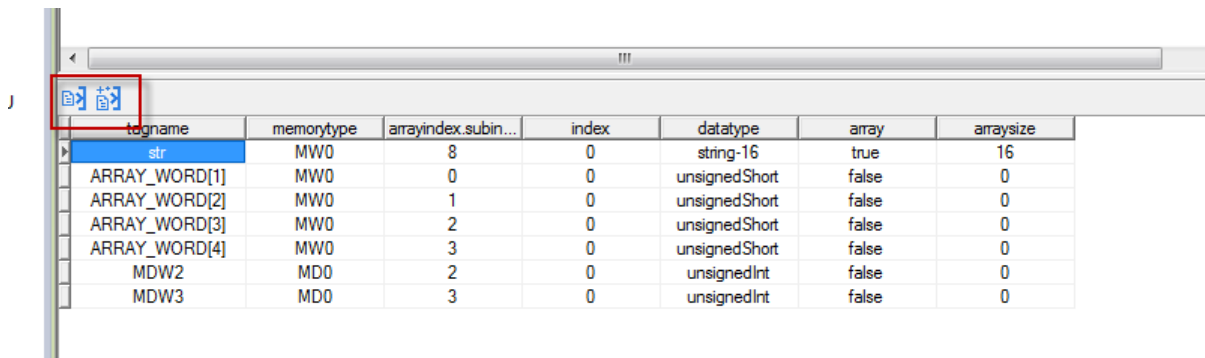


Figure 6

Special Data Types

The ABB Modbus TCP driver provides one special data type called "Node Override IP".

The Node override permits to change at run time the target controller IP. This memory type is an array of 4 unsigned bytes, one per each byte of the IP address.

The node override IP is initialized with the value of the controller IP specified in the project at programming time.

If the node override is set to 0.0.0.0, all the communication with the slave is stopped, no request frames are generated anymore.

If the Node Override has a value different from 0.0.0.0, it is interpreted as node IP override and the target IP address is replaced run-time with the new value.

In case the panel has been configured to access to a network of controllers, each node has its own Override variable.

Note: the Node Override values assigned at run-time are not retained among power cycles

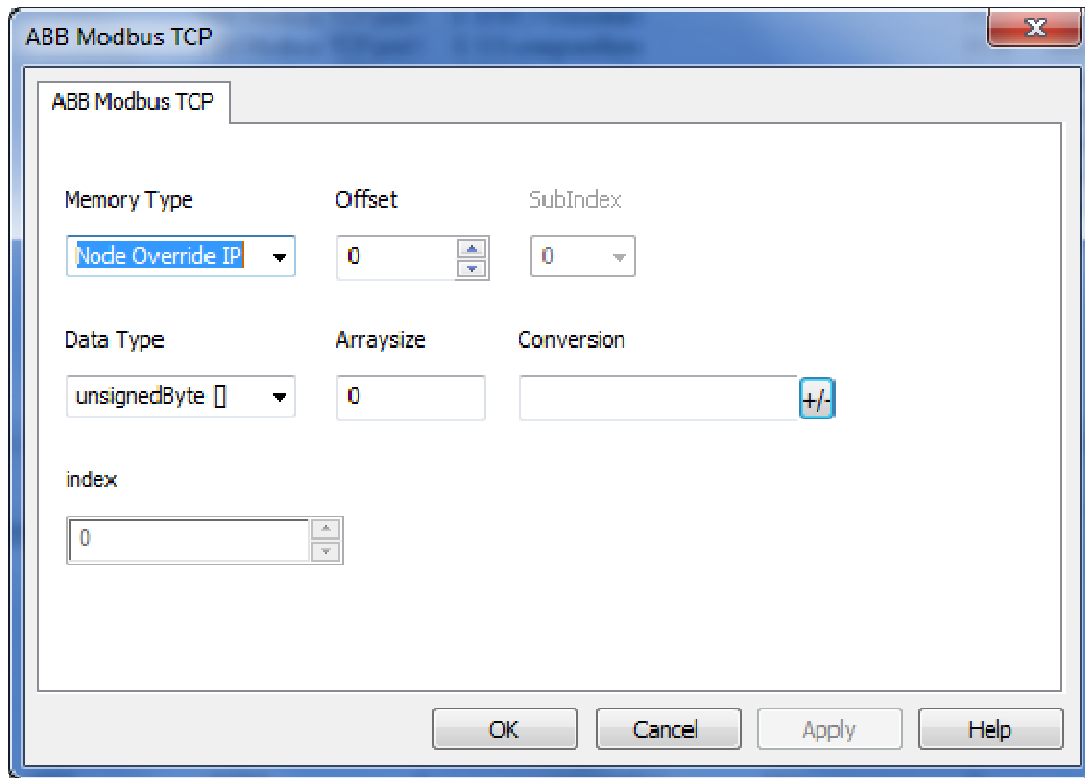


Figure 7

Communication Status

The current communication status can be displayed using the dedicated system variables. Please refer to the chapter “system variables” about available types and their use.

The codes supported for this communication driver are:

Error	Notes
NAK	Returned in case the controller replies with a not acknowledge
Timeout	Returned when a request is not replied within the specified timeout period; ensure the controller is connected and properly configured to get network access
Invalid response	The panel did receive from the controller a response, but its format or its contents is not as expected; ensure the data programmed in the project are consistent with the controller resources
General Error	Error cannot be identified; should never be reported; contact technical support

Implementation details

This Modbus TCP implementation supports only a subset of the standard Modbus TCP Function Codes.

The supported Function Codes are listed in the table below.

Code	Function	Description
01	Read Coil Status	Reads multiple bits in the panel Coil area
02	Read Input Status	Read the ON/OFF status of the discrete inputs (1x reference) in the slave
03	Read Holding Registers	Read multiple Registers
04	Read Input Registers	Reads the binary contents of input registers (3x reference) in the slave
05	Force Single Coil	Forces a single Coil to either ON or OFF
06	Preset Single Register*	Presets a value in a Register
16	Preset Multiple Registers*	Presets value in multiple Registers