PRODSP TECHNOLOGIES

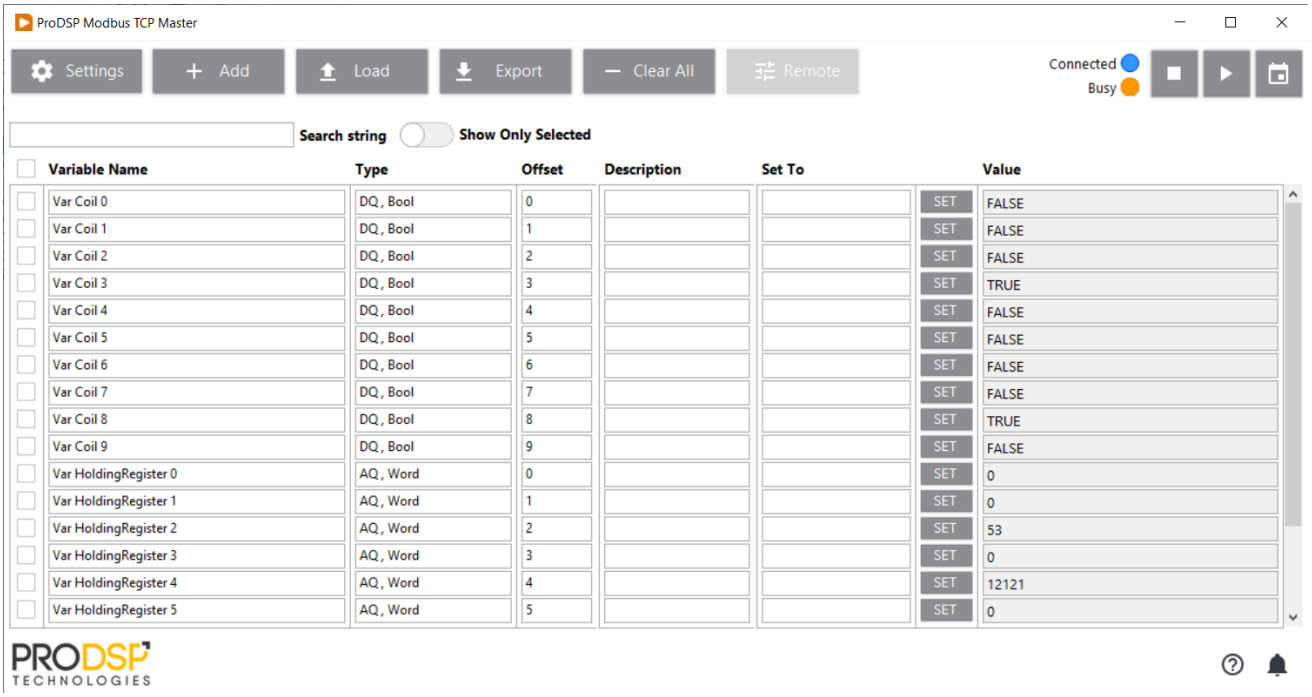
User Manual

ProDSP Modbus TCP Master

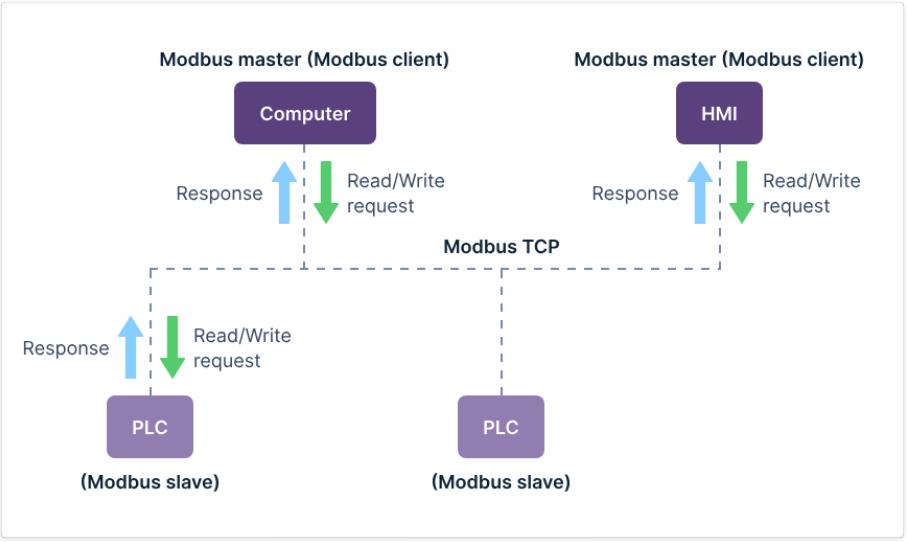
ProDSP Technologies Zrt. | H-1035 Budapest, Raktár u. 25-31. | info@prodsp.hu

Intended use

The ProDSP Modbus TCP Master (client) is developed to help system integrators, automation engineers and hobbyists to test and manually control Modbus TCP ready slave (server) devices (such as smart sensors, PLCs, instruments, automation actuators, etc.). The program can be installed on any Windows 10 (or higher) PCs. The convenient user interface enables to set up any Modbus table configurations according to the slave device description, use labels for read/write parameters, poll and control Modbus table entries. The program is commercially free to use.



The program is periodically polling the configured Blocks (see later), and displays their current values on the screen. The writable variables can be modified asynchronously from the UI. Users can also rename all the variables and add descriptions to help the testing process.



General Modbus TCP network (from kaaiot.com)

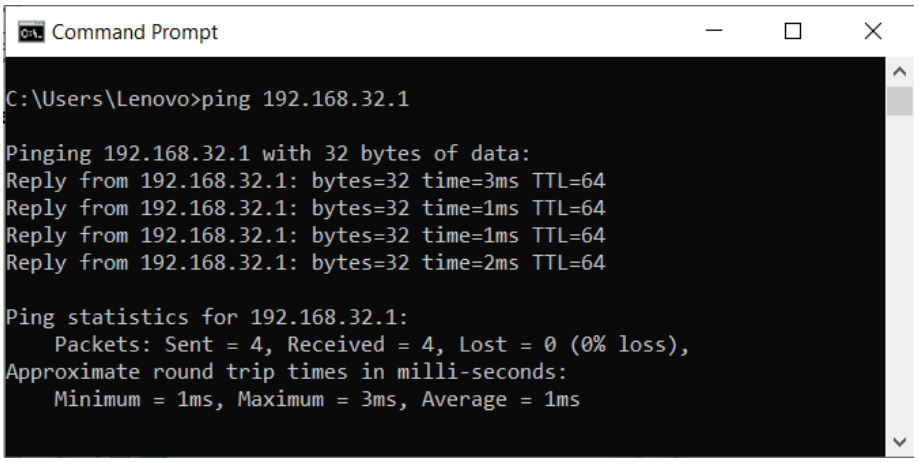
Installation

The installer will guide the user to install all the required packages and extensions. The product is dependent on the National Instruments LabView 32 bit Runtime package (installed by the installer if not present on your system). The default installation folder is “C:\Program Files (x86)\ProDSP\Modbus TCP Master” the program data folder is “C:\Users\Public\Documents\ProDSP Tools\ProDSP Modbus TCP Master”.

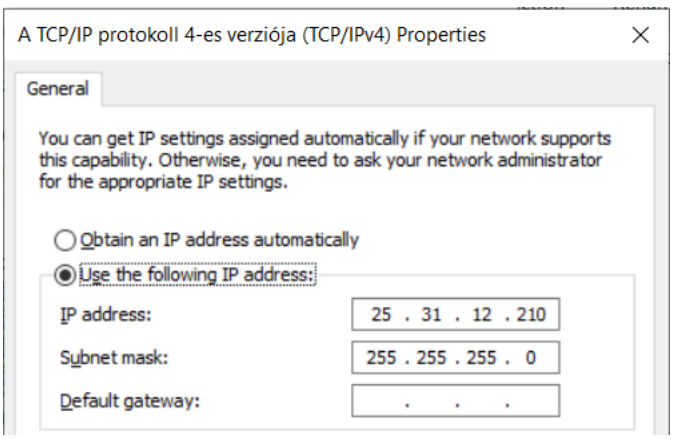
The program can be removed by the Windows “Add or remove programs” tool. This will not affect the data folder, so the configuration will be maintained even after a reinstall of the program.

Before Start

Make sure that the controlled device is connected to your PC and are in the same subnetwork. Verify the network connection with the command prompt ping tool. Open the Windows *Command Prompt* or *Power Shell* and type “ping *SLAVE_IP_ADDRESS*” where *SLAVE_IP_ADDRESS* is the actual IP address of the slave device. You should get positive results:



If your ping is timed out, check the connection and the ethernet settings of your device. Setup your PC network adapter to the same subnet as the slave device (via Windows Ethernet Settings) or check the parameters of your network switch.



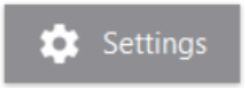
First Start

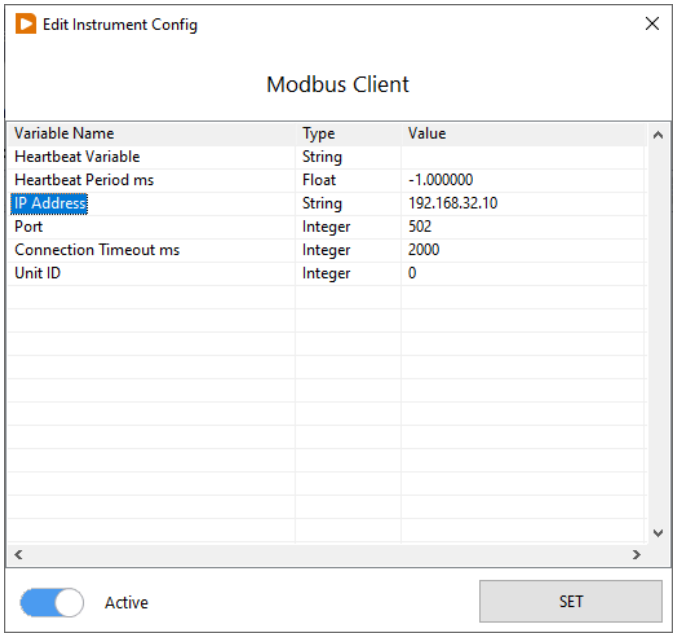
On the first startup the firewall of your operating system may ask for the preferred settings. **Enable network connections to the program in all types of networks on the firewall.**

The program will guide the user with a Getting Started pop-up through the configuration of the program. Follow the instructions to set up the parameters according to your Modbus slave device needs.

Setup Connection Parameters

The network settings have to be configured in the Settings menu. Press the button in the top toolbar:



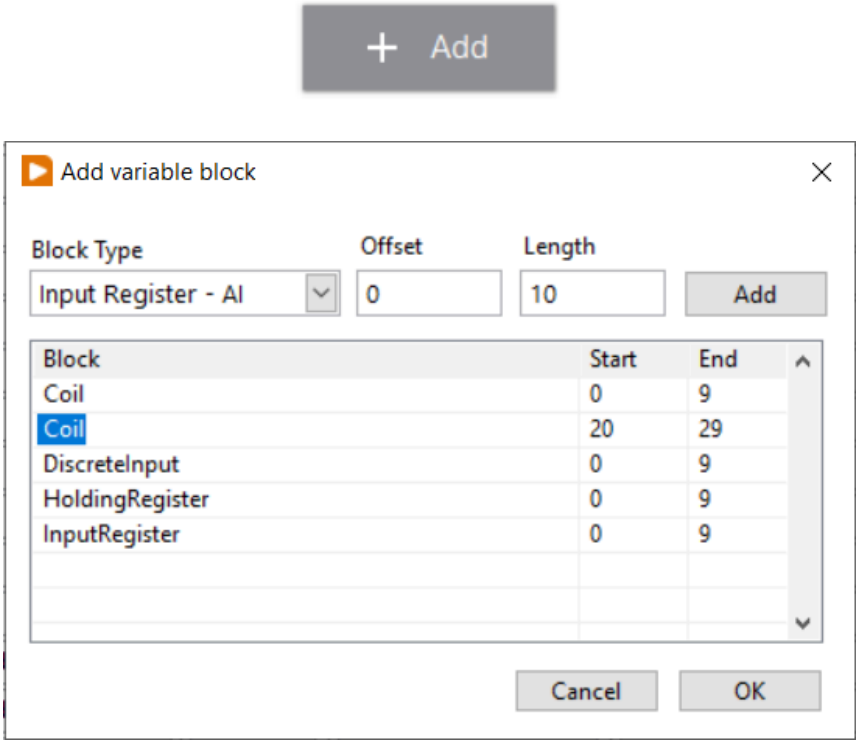


Double click on the parameter values to type the correct values. Press SET to accept changes.

- **Heartbeat Variable**: The variable name of the Modbus parameter used as heartbeat signal. Leave it empty if not needed.
- **Heartbeat Period ms**: The heartbeat value toggle period in milliseconds. Valid values are positive.
- **IP Address:** IP Address of the slave device
- **Port:** TCP Port of the slave device. Typically 502 for Modbus standard.
- **Connection Timeout ms:** Maximum wait time for TCP connection.
- **Active/Simulated Switch:** Switch between Simulated mode and Device (active) mode. Simulated mode does not use the slave device. It is only serving testing demo purposes.
- **Unit ID:** Modbus Unit ID (usually not used by the TCP slaves)

Setup Modbus Blocks

The used Modbus table configuration can be configured in the “Add variable block” window. Press Add in the top toolbar:



Add the usable Modbus variable blocks one-by-one. Select the **Block Type** (Modbus table type), then specify the **Offset** and **Length** in the Modbus table and press **Add**.

The program will poll all the defined blocks each poll cycle, and the users can write the defined write areas on the UI. Thanks to this approach the variable areas can be defined all over the available table in the slave, while only the required areas are polled to minimise the network communication. The blocks cannot overlap. Make sure that you follow the slave device table description and **only add areas which are supported by the slave**. The program follows the Modbus table data modell:

Primary tables	Access	Size	Features
Discrete input	R	1 bit (0–65,535)	Read on/off value
Coil (discrete output)[12]	R/W	1 bit (0–65,535)	Read/Write on/off value
Input register	R	16 bit words (0–65,535)	Read measurements and statuses
Holding register	R/W	16 bit words (0–65,535)	Read/Write configuration values

Modbus tables and variable types. Source and more information: https://en.wikipedia.org/wiki/Modbus

Press **OK** if all the required blocks are specified. Come back to this screen if additional blocks should be added.

Connect

After all the network and table configurations are set the operation can be started. The program tries to connect every 10 seconds to the slave if all the parameters are correct. Alternatively you can initiate the connect and disconnect action from the top toolbar.

- **Connected:** Connection status of the slave device
  - Green: Device successfully connected
  - Gray: Device not connected
  - Blue: Simulated mode, device not used
- **Busy**: Ongoing communication indicator
- **Stop Button:** Disconnect from the device
- **Start Button:** Connect to the device
- **Calendar Button:** Show communication error log.

Troubleshooting of connection issues

In case of unsuccessful connection:

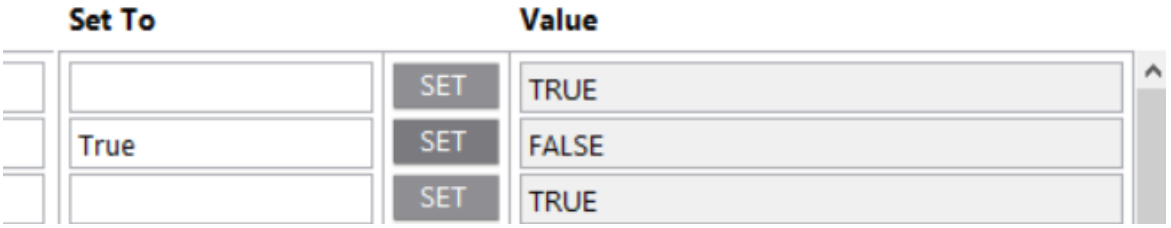
- Check network connection as described above.
- Check firewall settings. Enable all connections for the program.
- Check the communication error log.

Device is connected but disconnected shortly:

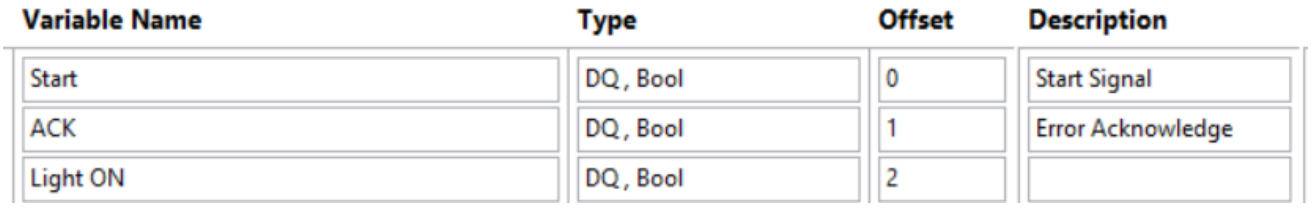
- Check the configured blocks. Most likely it is mismatched to the slave device table.

Operation

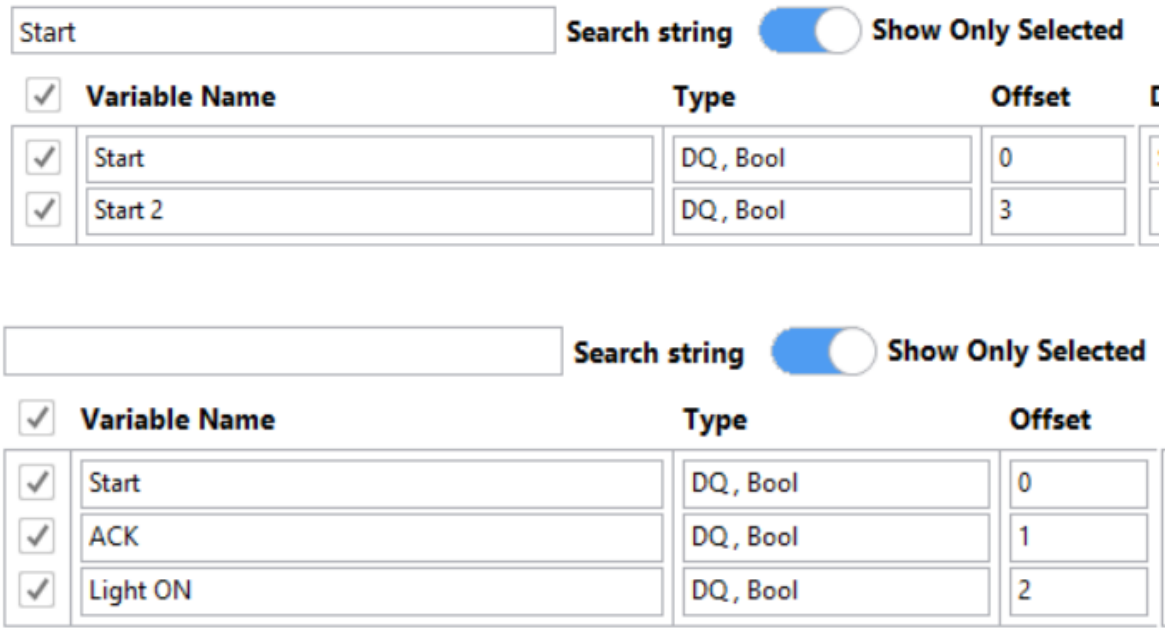
Check and write variables on the screen. The value column displays the actual value of the Modbus table. Type the desired value to the input field and press the “Set” button to overwrite (only for writable variables). The “Set” button toggles the boolean values if the input field is empty.



Rename variables and add description for easier operation. The names have to be unique.



Search for variables and select only a set of variables for display.



Close the program if the testing is over. All configuration and connection settings will be saved and the device will automatically connect on the next startup. Press the “Clear All” button in the top toolbar if a new device configuration is needed. This will erase all block and variable settings.

Advanced Features

Configuration file

The configuration can be saved to a file on the disk and can be loaded back later. Use the “Load” and “Export” buttons for this on the top toolbar:

The configuration file can be also modified in any text editor. Open the file after it is exported (e.g. with Notepad) and change the needed parameters. Do not change the format and make sure that you are following the original structure.

Variable Groups (Blocks) can be added or removed. Type has to be one of the following: DiscreteInput, Coil, HoldingRegister, InputRegister.

```
1   #GROUPS
2   Type,Offset,Length
3   InputRegister,0,10
4   DiscreteInput,0,10
5   Coil,20,10
6   HoldingRegister,0,10
7   Coil,0,10
```

Variable names can be also edited and added. Names have to be unique and VarType must be one of the supported block types. Only variable areas which are specified in the Groups section will be used in the program.

```
8   #VARIABLES
9   Name,Type,Offset,VarType,Description
10  Start,Coil,0,,Start Signal
11  ACK,Coil,1,,Error Acknowledge
12  Light ON,Coil,2,,
13  Start 2,Coil,3,,
14  Var Coil 4,Coil,4,,
```

Load the file into the program after the new configuration is saved.

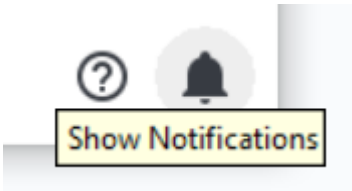
Heartbeat signal

It is often needed to use a heartbeat signal to prove connection between the slave and the master. Usually a boolean value is toggled periodically by the master which is checked by the slave to implement this function. The ProDSP Modbus TCP Master can also support this by specifying the

variable and the period in the Settings window (see above). The periodicity is not strictly followed on the PC, make sure that the threshold is high enough in the slave.

Error handling

In case any software error has occurred it is shown in the notification window. Check the error message and contact ProDSP if it cannot be resolved.



Additional Functionality

Remote Control

The program can be controlled via a standard TCP/IP Interface to read and write variables over the Modbus connection. This can enable automated testing of Modbus slave devices and control Modbus ready PLCs from any external programs and scripts. Python library available on request.

```
23  FW_Interface=CoreFW.FW()
24  conn=FW_Interface.Connect("localhost","XYZ")
25
26  if conn:
27
28      param = {
29          "Variable Name":"Start",
30          "Value":"True"
31      }
32
33      reply1 = FW_Interface.SendModuleCommand("Modbus Client", "Set Variable", param, 5000, True)
```

Please contact ProDSP if your project would require remote control functionality.

Bug reports and feature suggestions

Please contact us via email (info@prodsp.hu) in case of any bug or malfunction observed. Also let us know which features would you like to see in future versions and what do you like in the current one. Use our product freely on your own but it is strictly prohibited to resale in any form.