

Modbus Client

Modbus Client Lite



August 20, 2026
v1.5.0

Modbus Client

SetupAboutInputsOutputs

Setup

OK

TCP Port

502

Server Address

192.168.1.236

Reset

Bus Address

0

Connected To Device

Polling Interval in milliseconds

33

Polling time: 33ms

Cycle Monitor

Read Cycle

Write Cycle

Read Start

13431745632.714

Write Start

13431745421.779

Read End

13431745632.714

Write End

13431745421.779

Modbus Client : Modbus Client

SetupAboutInputsOutputs

Inputs (Read Only)

Bits	Address	Description	Value
1	E 0 0000	Button 1	
2	E 1 0001	Button 2	
3	E 2 0002	Button 3	
4	E 3 0003	Button 4	
5	E 4 0004	Button 5	

Registers

Address	Description	Value
E 10 000A	Analog 1	32768 8000
E 11 000B	Analog 2	39321 9999
E 12 000C	Analog 3	3932 0F5C
E 13 000D	Analog 4	19661 4CCD
E 14 000E	Analog 5	0 0000

Modbus Client : Modbus Client

SetupAboutInputsOutputs

Outputs (Read Write)

Clear All Overrides

Bits	Address	Description	Value	Override	Out	Sync
1	E 20 0014	Relay 1		0		S RO
2	E 21 0015	Relay 2		0		S RO
3	E 22 0016	Relay 3		0		S RO
4	E 23 0017	Relay 4		0		S RO
5	E 24 0018	Relay 5		0		S RO

Registers

Address	Description	Value	Override	Output	Sync
E 30 001E	Value 1	1234 04D2	0 0000 0	1234 04D2	S RO
E 31 001F	Value 2	5678 162E	0 0000 0	5678 162E	S RO
E 32 0020	Value 3	65535 FFFF	17039 428F 0	17039 428F	S RO
E 33 0021	Value 4	41287 A147	0 0000 0	41287 A147	S RO
E 34 0022	Value 5	0	0 0000 0	0	S RO



Contents

Introduction - Modbus Client	3
Introduction - Modbus Client Lite	4
Warnings and Disclaimers	5
Properties.....	6
Controls	8
Notes	13
Changelog	16



Introduction - Modbus Client

Modbus Client allows Q-SYS to interface with external Modbus Server devices over standard TCP/IP communication. This allows Q-SYS to receive inputs and control outputs of automation equipment that supports the Modbus TCP protocol, greatly expanding the I/O capabilities of a Q-SYS system by allowing both digital and analog signals to be easily integrated into a design.

The Pro version of the plugin offers up to 999 Input Bits, Input Registers, Output Bits, and Output Registers, providing a large number of control I/O points - hundreds more than a typical control system. Inputs are polled from the Modbus Server Device up to 30 times per second, configurable in 1 ms increments, and outputs are updated up to 30 times per second. Both Input and Output Bits are represented as toggle buttons for easy control pin wiring. Registers are represented as integer fields with possible values from 0 to 65,535, and as hexadecimal string fields with values from 0x0000 to 0xFFFF.



Each Input and Output in the plugin may be configured for any I/O address and does not have to be sequential. Unused I/O can be disabled to reduce clutter and confusion, and I/O may also be temporarily disabled while testing or commissioning is ongoing.

Outputs offer an "Override" feature, which allows output values to be specified within the plugin UI, ignoring data from the control pin. When an Override is active, the provided value will be applied to the output, allowing for easier system testing, commissioning, and temporary bypass. Outputs also offer a "Sync" feature, which allows them to be updated in Q-SYS whenever they are changed on the server device by another actor, such as PLC logic or another control system.

Introduction - Modbus Client Lite

Modbus Client Lite is a lighter, more limited version of Modbus Client intended for smaller installations.

Functionality is the same as Modbus Client, with the following limitations:

- Maximum number of inputs is 16 (bits + registers), and 16 outputs (bits + registers)
- Shortest polling frequency is 8 times per second
- Output Sync is not available
- Overrides are not available
- Only one instance of Modbus Client Lite may be present in a Q-SYS design



Warnings and Disclaimers

Warning

The Modbus Client plugins are not intended to be used in manufacturing, motion, or safety applications, or any other applications where a software failure could result in damage to equipment or property, injury to persons or animals, or loss of life. The plugins are only intended to expand the I/O capabilities of a Q-SYS system for control usage within an entertainment-related permanent installation in which General Purpose I/O is not sufficient.

While ProIO attempts to mitigate software errors, we do not guarantee that our software is free from errors as outlined in the Software License Agreement. By using these plugins, you assume all risk associated with any consequences that may arise if the plugins should malfunction, including but not limited to, malfunctions of the Q-SYS system, the scripting environment, or with the intellectual property of ProIO.

Important Disclaimer

Due to data processing limitations within the Q-SYS control engine, information may only be processed up to 30 times per second, or about every 33 ms. These refresh rates are far too slow for use in any motion or safety applications. These refresh rates may also be considered too slow for certain control applications. Careful consideration should be given when designing and implementing a system that relies on this plugin or third-party devices. These performance rates are approximately equal to the refresh rates of native Q-SYS GPIO which is sufficient for most general-purpose control applications.

Programmers and users are advised to ensure response times to any remote systems are within limits that may be set by local laws or regulations.



About Modbus

Modbus is an open protocol used by various industrial and automation devices. Data can be exchanged between a source referred to as a Server or Slave, and an observer referred to as a Client or Master. Modbus is a polling-based protocol where all data transactions are initiated by the Client and the Server responds to the request. Inputs must be requested by the client in order to receive those values. A Server device does not send or initiate data on its own.

Data is organized into Inputs and Outputs. Inputs represent sensory or stateful information of a device and Outputs represent controllable parameters or actions of a device. In the Modbus Client Plugins, Inputs are read-only. Both Inputs and Outputs can be represented by two different types of data; Bits and Registers. A Bit is a single boolean value; on or off/true or false. They are represented in the plugin as a Toggle control. A Register is a 16-bit word with 65,536 unique possible values. Registers are represented in the plugin as both numerical values in base 10 (decimal) and as text boxes showing their hexadecimal equivalent. It is important to note that the data encoding within a register is specific to the Server device. Consult the manufacturer's technical reference to understand how data may be encoded.

All data in the device is listed according to its memory address. According to the Modbus specification, a device can determine the difference between address 0x000 of input bits and address 0x000 of input registers. However, the specific memory addresses used and whether the overlap or are separate ranges is determined by the device's manufacturer.



Device Connection

The Modbus protocol can exist in a number of different styles communicating over different wiring methodologies. The original form is on a serial bus that allows many devices to be connected on the same cable. Each device is configured with a unique bus address to allow a Client to address each device individually. In this topology, Modbus may appear as "Modbus RTU". This plugin is only directly compatible with Modbus TCP, using the TCP communication method over Ethernet networks. However, there are Ethernet to serial gateway devices that can translate Modbus TCP to Modbus RTU. In this setup, the bus address is still relevant and it is provided in this plugin.

The default for most devices is a Bus Address of zero.

Typically, an IP addresses is all that is needed for the plugin to connect via TCP with a device over Ethernet. The default TCP port number is 502 although this is changeable as necessary.

The Status control of the plugin will indicate when the plugin is connected or disconnected from the device, however this only indicates whether a TCP session has been achieved. A dedicate LED for Server Connection specifically indicates when Modbus application data is being received. It is possible for some devices to form a valid TCP session but fail to answer Modbus poll requests which is why this control is useful.

Traditional Modbus is a serial line protocol that supported connecting multiple devices to the same line. Therefore, the protocol includes a "Bus Address" field which allows requests to target a specific device on the communication bus. Modbus TCP offers the same functionality for cases where multiple servers share the same IP address.

Multiple instances of the Modbus Client plugin can run in the same Q-SYS design simultaneously provided that they are connecting to different Server devices. It is not recommended for two Client plugins to attempt to connect to the same Server device.



Output Features

Both Output Bit and Output Registers offer the ability to set Overrides on each output. Typically, the "Value" control pin would be wired to something in the Q-SYS Design to send data to the Modbus device. However, if this normal logic should need to be interrupted or overridden to some other state, the plugin allows for this by using the Override section without needing to implement this functionality outside of the plugin. Overrides is the same concept as using "forces" in traditional PLC platforms.

To use an Override, first specify the intended value that should be held, then engage the Override by click the orange "O" button. The Status will update to provide a reminder that an override is in effect. The Output fields will reflect the Override state and the Value fields will be ignored. The Output field is the actual state that is transmitted to the device.

The Pro version of the plugin allows for Outputs to be read back from a device in addition to inputs. This synchronization can be two-way or one-way. To enable this feature, set the pluing property "Sync Outputs" to "Enable". The plugin interface will change to provide additional controls on each output. Each output will present a yellow "S" button in the Output section as well as a pink "RO" button. The "S" button stands for Sync and "RO" stands for Read-Only. When Sync is enabled on the output, the value of the output will be read from the device allowing for two-way synchronization between Q-SYS and any internal logic of the device. Overrides are not possible when Sync is enabled. Enabling Read-Only on the output will prevent the plugin from modifying that particular output.



Cycle Monitor

The Pro version of the plugin features a Cycle Monitor. The Cycle Monitor will raise triggers and provide timestamps at the beginning and end of each set of read and write requests. This is useful when troubleshooting communication issues with slower response devices to help fine tune the ideal polling interval time.

For advanced applications such as data logging and time averaging, the plugin will retain the last values received by a device. However if the data is stale or the connection is dropped, an averaging function would not be able to determine this without monitoring the plugin's connection status. Instead, the Cycle Monitor provides "Read End" and "Write End" triggers that a script can use to know when valid data has been received by the device.

Read Start is triggered and time-stamped at the beginning of a new poll of data. Read End is triggered and time-stamped when all read requests for the interval have been answered. Read End is reliable for processing the data.

Similarly, Write Start is triggered and time-stamped when write requests are enqueued to be sent to the device, and Write End is triggered and time-stamped when all write requests have been acknowledged by the device.



Properties

Input Bits

an integer that defines the number of input bits available for configuration. In Modbus Client, acceptable values are between 0 and 999. In Modbus Client Lite, acceptable values are between 0 and 16.

Input Registers

an integer that defines the number of input registers available for configuration. In Modbus Client, acceptable values are between 0 and 999. In Modbus Client Lite, acceptable values are between 0 and 16.

Pro Feature: The total number of Input Bits + number of Input Registers cannot exceed 16 in Modbus Client Lite.

Output Bits

an integer that defines the number of output bits available for configuration. In Modbus Client, acceptable values are between 0 and 999. In Modbus Client Lite, acceptable values are between 0 and 16.

Output Registers

an integer between 0 and 999 that defines the number of output registers available for configuration. In Modbus Client, acceptable values are between 0 and 999. In Modbus Client Lite, acceptable values are between 0 and 16.

Pro Feature: The total number of Output Bits + number of Output Registers cannot exceed 16 in Modbus Client Lite.



Properties

Is Required

a boolean that sets whether the target device is required for system functionality

Restate Outputs

an enumeration with choices of "Never", "On Connect", and "Continuously" that determines how the plugin should retransmit outputs to the remote device. "Never" will send the output value only when it is changed in the plugin and never retransmit it. "On Connect" will transmit all outputs, regardless of change, whenever the plugin connects or reconnects to the device. "Continuously" will cause all enabled outputs to be transmitted on every poll interval, similar to how a PLC operates. This property allows unnecessary network traffic to be minimized.

Sync Outputs

an integer between 0 and 999 that defines the number of output registers available for configuration. In Modbus Client, acceptable values are between 0 and 999. In Modbus Client Lite, acceptable values are between 0 and 16

Pro Feature: Sync Outputs is not available in Modbus Client Lite

Registered Owner

a string that lists the license holder's name of the plugin registration for this installation

Registered Project

a string that lists the licensed project name of the plugin registration for this installation

Registered Key

a string that lists the license key code of the plugin registration for this installation



Controls

Status

displays various status messages to aid in monitoring the connection with the Server Device. Communication errors will show as a fault. When a force is present on an output, the status will show as Compromised as a reminder that the system has been overridden

Server Port

sets the TCP communication port number to use when connecting to the Server Device

Server Address

sets the IP address to use when connecting to the Server Device

Server Connected

an LED that indicates specifically that Modbus data is actively being received by the Client. This is different than the TCP connection state.

Reset

triggers the plugin to close the current TCP connection and re-establish a new connection. This is useful for refreshing an unresponsive connection.

Bus Address

sets the Modbus Bus Address (or Device ID) of the Server Device. This is useful if multiple servers share the same IP address. Refer to the Notes section for more details

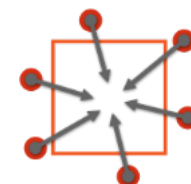
Polling Time

sets the polling frequency of inputs from the Server Device in milliseconds

Pro Feature: Minimum value in Modbus Client is 33. Minimum value in Modbus Client Lite is 150

Polling Interval

displays the time of each input poll



Cycle Monitor Controls

Read Start Trigger

a Trigger button that is activated when the Client begins sending read requests

Read Start Time

the epoch time when the Client begins sending read requests

Read End Trigger

a Trigger button that is activated when all read requests have been answered by the Server

Read End Time

the epoch time when the Client has received all answers by the Server

Write Start Trigger

a Trigger button that is activated when the Client begins sending write requests

Write Start Time

the epoch time when the Client begins sending write requests

Write End Trigger

a Trigger button that is activated when all write requests have been acknowledged by the Server

Write End Time

the epoch time when the Client has received all acknowledgements by the Server



Cycle Monitor is only available in the Pro version of the Modbus Client



Controls

Input Enable

a toggle button that sets whether this Input is used and should be polled

Input Dec Address

an integer field that ranges from 0 to 65,535 that sets the memory address of the input in decimal

Input Hex Address

a string field that ranges from 0x0000 to 0xFFFF that sets the memory address of the input in hexadecimal

Input Description

a text field that provides human-readable labeling of each input

Input Bit Value

a toggle button that represents the state of the input on the Server Device

Input Register Dec Value

an integer field that ranges from 0 to 65,535 that represents the value of the register on the Server Device in decimal

Input Register Hex Value

a string field that represents the register value in hexadecimal, ranging from 0x0000 to 0xFFFF

Output Enable

a toggle button that sets whether this Output is used and should send updates to the Server Device



Controls

Output Dec Address

an integer field that ranges from 0 to 65,536 that sets the memory address of the output of the Server Device in decimal

Output Hex Address

a string field that ranges from 0x0000 to 0xFFFF that sets the memory address of the output in hexadecimal

Output Description

a text field that provides human-readable labeling of each output

Output Bit Value

a toggle button that sets the desired state of the output on the Server Device

Output Bit Override Value

a toggle button that sets the value of the output in the overridden state

Output Bit Override Enable

a toggle button that overrides the output bit's value with the value specified by the Bit Override Value toggle button. When active, the output will only follow the overridden value

Pro Feature: Output Bit Override is not available in Modbus Client Lite

Output Bit Output

a toggle button that represents the actual output state, after considering the overridden state



Controls

Output Bit Sync

a toggle button that enables the bit to be read back from the remote device to keep the current state synchronized with the plugin

Pro Feature: Output Bit Sync is not available in Modbus Client Lite

Output Bit Read Only

a toggle that sets whether the output bit should be read-only within Q-SYS, causing the plugin to only show the current state of the output on the remote device

Output Register Dec Value

an integer field ranging from 0 to 65,535 that sets the desired decimal value of the output of the Server Device

Output Register Hex Value

a string field ranging from 0x0000 to 0xFFFF that indicates the hexadecimal value of the decimal value

Output Register Dec Override Value

an integer field ranging from 0 to 65,535 that sets the decimal value of the output in the overridden state

Output Register Hex Override Value

a string field ranging from 0x0000 to 0xFFFF that sets the hexadecimal value of the output in a overridden state

Output Register Override Enable

a toggle button that overrides the output register's value with the value specified by the Register Override Dec/Hex Value fields. When active, the output will only follow the overridden values.

Pro Feature: Output Register Override is not available in Modbus Client Lite.



Controls

Output Register Dec Output

an integer field ranging from 0 to 65,535 that represents the actual output value in decimal, after considering the overridden state

Output Register Hex Output

a string field ranging from 0x0000 to 0xFFFF that represents the actual output value in hexadecimal, after considering the overridden state

Output Register Sync

a toggle button that enables the register to be read back from the remote device to keep the current state synchronized with the plugin

Pro Feature: Output Register Sync is not available in Modbus Client Lite.

Output Register Read Only

a toggle that sets whether the output register should be read-only within Q-SYS, causing the plugin to only show the current state of the output on the remote device



Implemented Modbus Functions

The Modbus Client plugin is compliant with the official Modbus TCP protocol and uses methods:

- 0x01 (Read Output Coils)
- 0x02 (Read Discrete Inputs)
- 0x03 (Read Holding/Output Registers)
- 0x04 (Read Input Registers)
- 0x05 (Write Single Coil)
- 0x15 (Write Multiple Coils)
- 0x16 (Write Multiple Registers)

I/O Addressing - Supporting both Modbus and JBus

Modbus TCP is a somewhat modified version of the original Modbus Serial communication protocol. Modbus TCP has certain limitations to the number of addresses available and the quantity of coils and registers that can be communicated in a single request. This is to keep a TCP message under the default MTU size. JBus, on the other hand, is more relaxed with these requirements. The most noticeable difference between the two protocols for system programmers is that Modbus TCP is 0-based in its memory addressing while JBus is 1-based.

The Modbus Client plugin is designed to support both 0-based and 1-based addressing. It is up to the Q-SYS programmer to define the memory addresses for their use case appropriately. The plugin does not apply any offset to the address numbers listed. If no I/O points are addressed as 0, and a request is issued for memory address 0, then the request will fail.

Consequently, the plugin allows I/O to be addressed as high as 65,536 or 0x10000. This is to maintain compliance with JBus, however, it is an illegal data address for the Modbus protocol. Since all hexadecimal values are limited to 4 digits (2 bytes), the plugin will show a decimal address of 65,536 but a hexadecimal address of 0xFFFF.



Hexadecimal readouts of address and value fields are just for the programmer's convenience and do not have any practical effect on data transmission. The plugin only considers decimal values for requests and transmissions. When a hexadecimal value is entered or is changed, however, the corresponding decimal field will also be updated to the correct value.

While the plugin will attempt to receive either Modbus TCP or JBus-style messages without additional user configuration, it is recommended that messages stay under the default MTU size of 1,500 bytes, otherwise, the message may be split into multiple packets, and the request may fail to be processed correctly.

Non-Sequential and Non-Contiguous Addresses

As the Modbus Client plugin allows any Input or Output to have any address, addressing may be non-sequential and noncontiguous. This allows flexibility in only creating I/O that are needed at any address. Many installations typically prefer to use logical numbering conventions for various groups of I/O that often skips over blocks of addresses. The plugin supports this without having to have a large number of unused controls.

When a poll request is made for inputs, all inputs are arranged first by bits, then by registers, in ascending order based on their Modbus Address, regardless of line number in the plugin. Contiguous regions of address numbers are then requested in a single message. Each non-contiguous break will result in a separate message being transmitted. A large number of requests may result if many non-contiguous input addresses are requested. Performance may degrade if the requests cannot be fulfilled faster than the specified polling time. For this reason, the polling time should be set to a slower interval when using a large number of I/O, depending on the performance of the Server Device.

I/O Priority

All enabled inputs are polled on every polling interval. Outputs are transmitted according to the property "Restate Outputs". As Q-SYS only updates controls at a rate of 30 Hz, this becomes the fastest polling and updating interval. Output write messages and input read requests are generated asynchronously within the plugin. To prevent overwhelming the server device with requests, only one message may be in flight at a time. Messages are queued within the plugin and are only transmitted when a response is received. The queue honors output write requests first before sending poll requests from the queue.



Notes

This allows outputs to be more time-accurate at the minor expense of delaying input triggers.

When the Sync Outputs property is Enabled, and any outputs have their Sync button on, the outputs will be added to the polling queue on each polling interval. These will be updated after the input bits and registers. If a remote device has a force enabled on an output, that is, it is set to ignore change requests, this will cause the plugin to appear as if the outputs are not updating or changing.



Changelog

v1.5.0 – August 20, 2026

- New feature: Cycle Monitor
- New feature: Server Connectivity indicator
- Added the Status control to the top of all pages (except About)
- Minor UI tweaks
- Improved the response and useability of the hexadecimal readouts

v1.4.0 – August 19, 2026

- Implemented support for function code 0x05 Single Coil Write. Single updates now default to using this code.
- Correction for handling error codes 0x81 and 0x83
- Improvement in handling devices that stop responding
- Corrections in how polling is handled when the polling interval is faster than a device can respond
- Hardened data management when requesting a large number of registers
- Resolved odd behaviors in Override, Sync, and Read-Only buttons for outputs
- Addressed critical issues that could affect other plugin instances

v1.3.2 – February 11, 2026

- Resolved a bug which caused Sync Outputs buttons not to display under certain circumstances.

Changelog

v1.3.1 – December 10, 2025

- Minor updates improving stability.

v1.3.0 – October 10, 2025

- Minor updates improving stability.
- Added About tab to describe I/O configuration & license state.
- Introduced Modbus Client Lite, using same logic with feature limits.

v1.2.0 – January 21, 2025

- Polling time is no longer expressed as steps of 50ms and is now expressed a direct number of milliseconds ranging from 33ms to 10,000ms.
- Various stability fixes when dealing with broken TCP connections and buffer management
- Improved the user experience when changing output values when the output is not Enabled.
- Resolved an issue where the plugin would not reconnect automatically if a device became unresponsive but still believed to be connected.
- Added the Restate Outputs property to allow more robust handling of output synchronization at user discretion.
- Added the Reset button to allow the connection to be manually refreshed without needing to change the IP address.

v1.1.2 – July 11, 2024

- Added demo functionality for a time-limited trial of the plugin.

Changelog

v1.1.1 – June 28, 2024

- Changed the input controls from read-only to be editable to allow for easier system commissioning without needing to rely on the connected device to change input values. Now a Q-SYS design can be fully tested from the plugin without hardware connection.
- Modified the behavior of the plugin to not throw a fault when a device does not support the Vendor or Product Code Identification function used by the plugin as a standby heartbeat mechanism.

v1.1.0 – March 21, 2023

- Changed the “Force” terminology to “Override” to minimize confusion as to the influence over the end device
- Added new property “Sync Outputs” to enable reading output states from the remote server device
- Added new controls “Sync” and “Read Only” on outputs, when the “Sync Outputs” property is Enabled
- Output controls in the plugin UI are now read-only and cannot be directly manipulated
- Improved the change detection method to reduce unnecessary network traffic
- Added better connection handling, timeout detection, and reconnection with the remote device
- Added a safeguard to prevent polling faster than what the network or remote server can handle
- Added a safeguard to reconnect if the remote device stops communicating for 5 times the polling interval
- Added a default message heartbeat to keep connections alive when no input or output controls are enabled

v1.0.4 – May 25, 2022

- Improved messaging with the remote device upon TCP connection establishment allowing for immediate controls updates upon connection. This is most noticeable when using a slower polling interval.

v1.0.3 – April 13, 2022

- Modified the connection timeout to be equal to the polling interval to better detect and manage dead TCP sessions
- Added a limit to the number of messages to be queued to prevent increasing memory consumption leading to a core crash. An error message will appear when this occurs and the connection to the remote device will be reset.

v1.0.2 – April 12, 2022

- Fixed an issue where messages would not be sent after the connection with a device was dropped and reconnected which could lead to high memory consumption causing a core crash.
- Added a safeguard to ensure the maximum amount of memory used by the plugin would not impact system stability. A status fault message will appear when this occurs.
- Lengthened the logging interval of repetitive verbose log messages from 5 seconds to 30 seconds, cutting down on log entries.
- Improved connection monitoring to report connection interruption faster (within 1 second).
- Improved status reporting when multiple read or write errors were received which caused the status control to flicker violently.
- Bug fix where changing the polling interval caused a script error and failed to update the displayed polling time in milliseconds.

v1.0.0 – December 2, 2021

Original release