MT Focus 6000

Open Protocol Test Program

Atlas Copco Industrial Technique AB

9839 0733 01
Specification release 2.0
2017-10
MT Focus 6000 Open Protocol test program documentation

Atlas Copco Industrial
Technique AB
SE-10523 STOCKHOLM
Sweden
Telephone: +46 8 743 95 00
Telefax: +46 8 644 90 45
www.atlascopco.com

Table of Contents

1 Introduction

This document describes the functionality of the MT Open Protocol Test Program which purpose is to test the Open Protocol implementation in the MT Focus 6000 controller. For a description of the protocol itself, see document **9839 0732 00 MT Focus 6000 Open Protocol Specification**.

1.1 Revision history

Version	Date	Author	Change
1	2017-01-26	Martin Persson	First revision
2	2017-10-23	Therese Alm	Second revision

2 Requirements

2.1 Software requirements

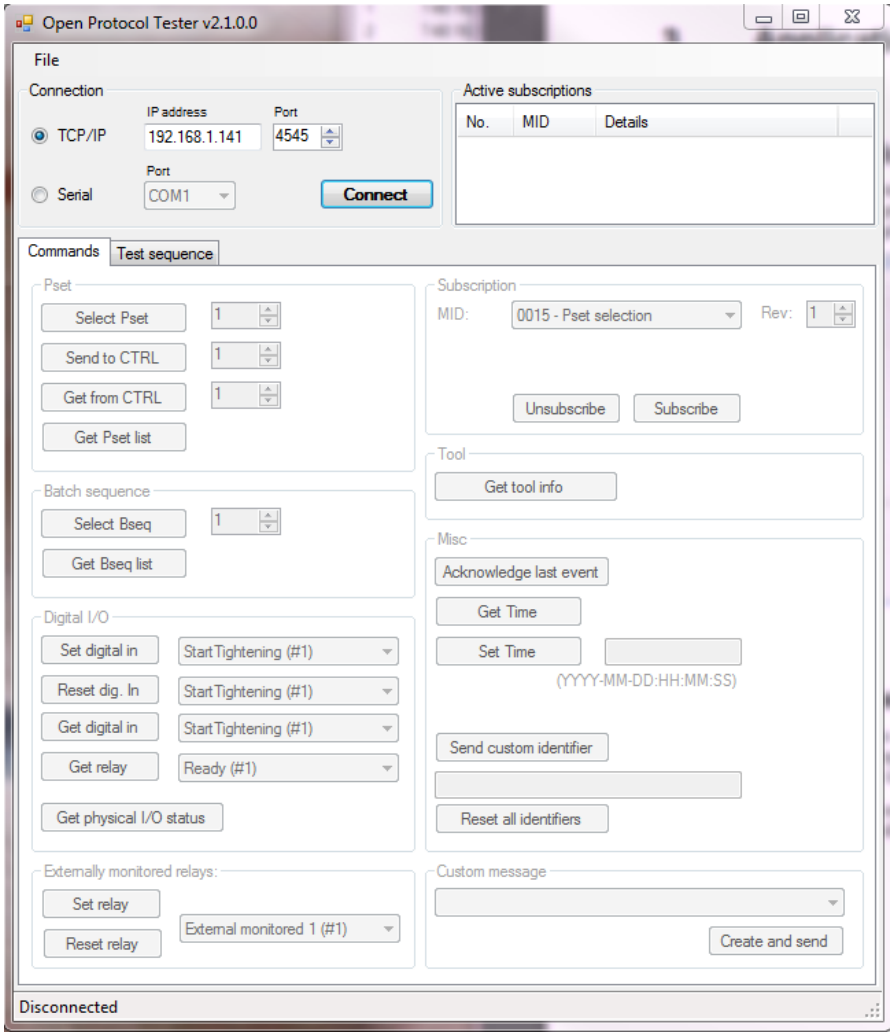
The Open Protocol Test Program is dependent on the following software.

- MTCom, bundled with ToolsTalk MT installation.
- .NET Framework 4.5.1, can be installed during setup if an internet connection is available.

3 Application overview

3.1 Main window

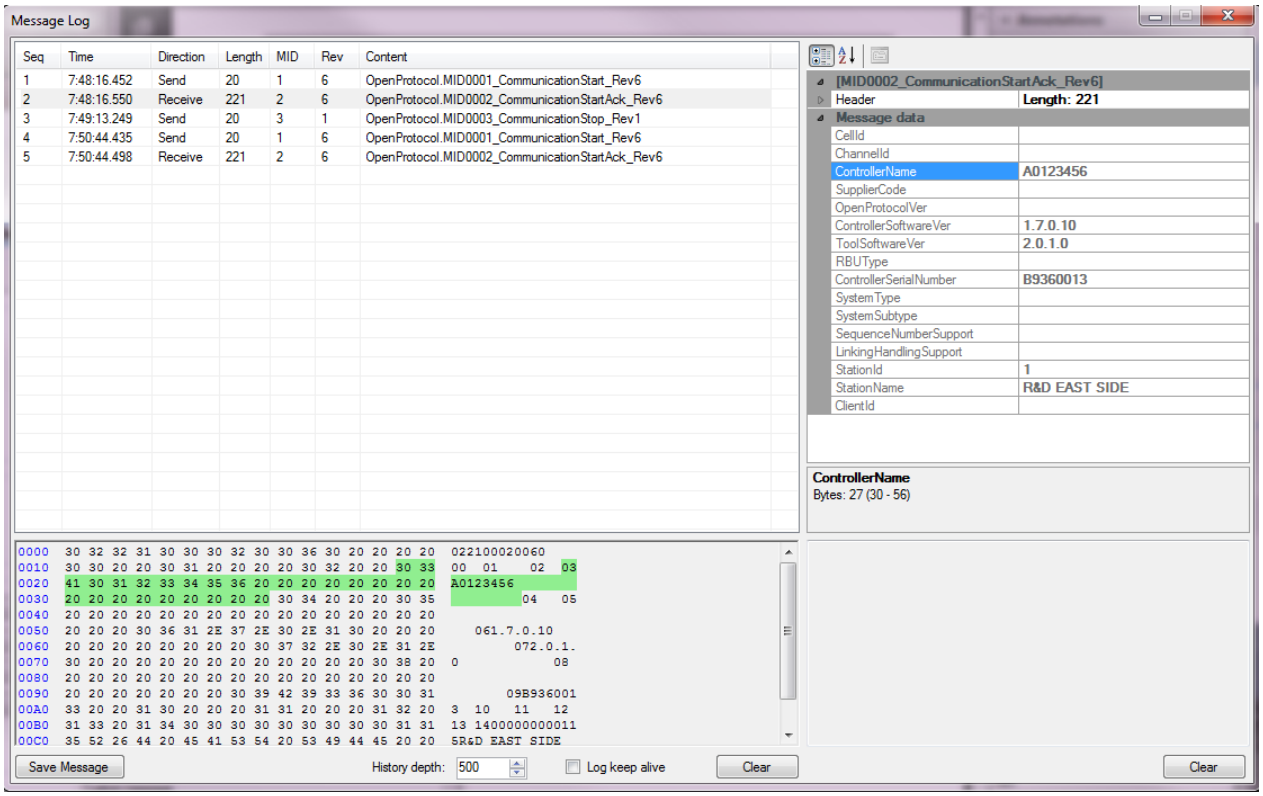
The main window is the first window presented when the application is started. All controls except the Connection group are initially greyed out until a connection has been established. Every protocol request that can be sent through this application are created from this window.



3.2 Message window

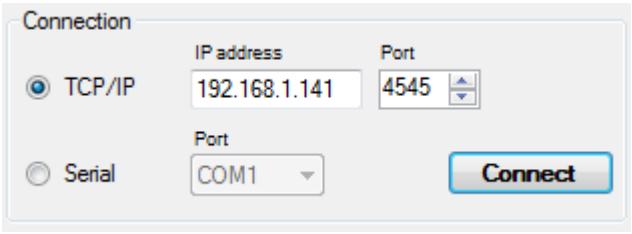
The message window maintains a list of messages sent and received from the client. It is split into four main parts. The message list (top left), a detailed message property view (top right), a binary message view (bottom left) and an error log (bottom right).

Selecting a message in the message list will update both the property view and the binary view. Data fields selected in the property view will also be highlighted in the binary view.


Message Log
Seq Time Direction Length MID Rev Content
1 7:48:16.452 Send 20 1 6 OpenProtocol.MID0001_CommunicationStart_Rev6
2 7:48:16.550 Receive 221 2 6 OpenProtocol.MID0002_CommunicationStartAck_Rev6
3 7:49:13.249 Send 20 3 1 OpenProtocol.MID0003_CommunicationStop_Rev1
4 7:50:44.435 Send 20 1 6 OpenProtocol.MID0001_CommunicationStart_Rev6
5 7:50:44.498 Receive 221 2 6 OpenProtocol.MID0002_CommunicationStartAck_Rev6
[MID0002_CommunicationStartAck_Rev6]
Header Length: 221
Message data
CellId
ChannelId
ControllerName A0123456
SupplierCode
OpenProtocolVer
ControllerSoftwareVer 1.7.0.10
ToolSoftwareVer 2.0.1.0
RBUType
ControllerSerialNumber B9360013
SystemType
SystemSubtype
SequenceNumberSupport
LinkingHandlingSupport
StationId 1
StationName R&D EAST SIDE
ClientId
ControllerName
Bytes: 27 (30 - 56)
0000 30 32 32 31 30 30 30 32 30 30 36 30 20 20 20 20 022100020060
0010 30 30 20 20 30 31 20 20 20 20 30 32 20 20 30 33 00 01 02 03
0020 41 30 31 32 33 34 35 36 20 20 20 20 20 20 20 20 A0123456
0030 20 20 20 20 20 20 20 20 30 34 20 20 20 30 35 04 05
0040 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
0050 20 20 20 30 36 31 2E 37 2E 30 2E 31 30 20 20 20 061.7.0.10
0060 20 20 20 20 20 20 20 20 30 37 32 2E 30 2E 31 2E 072.0.1.
0070 30 20 20 20 20 20 20 20 20 20 20 20 20 30 38 20 0 08
0080 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
0090 20 20 20 20 20 20 20 30 39 42 39 33 36 30 30 31 09B936001
00A0 33 20 20 31 30 20 20 20 31 31 20 20 20 31 32 20 3 10 11 12
00B0 31 33 20 31 34 30 30 30 30 30 30 30 30 30 31 31 13 1400000000011
00C0 35 52 26 44 20 45 41 53 54 20 53 49 44 45 20 20 5R&D EAST SIDE
Save Message
History depth: 500
Log keep alive
Clear
Clear

4 Application functions

4.1 Connection group



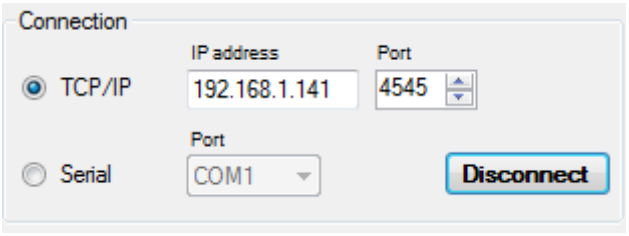
4.1.1 Connect button

Connect to a client via Ethernet by entering its IP address and clicking the connect button. The default port of 4545 should not be changed. It is also possible to connect a client via RS232.

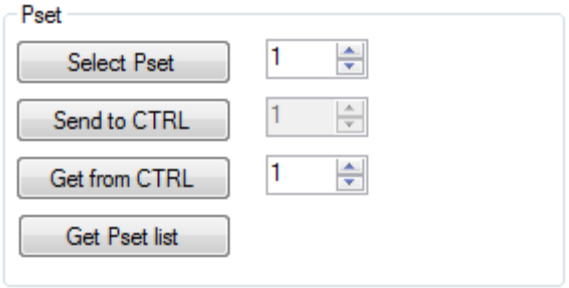
Make sure that "RS232 mode" in controller is set to "Protocol"

4.1.2 Disconnect button

Disconnect from a client by clicking this button.



4.2 Pset group



4.2.1 Select Pset button

Send a select Pset request to the controller by clicking this button. The edit box to the right selects which Pset that will be selected.

4.2.2 Send to CTRL button

Upload a previously downloaded Pset to the controller.

4.2.3 Get from CTRL button

Download a Pset (number indicated by edit box to the right) from the controller by clicking this button. The downloaded file can be uploaded again using “Send to CTRL”.

4.2.4 Get Pset list button

Send a request to download a list of available Psets by clicking this button.

4.3 Batch sequence group

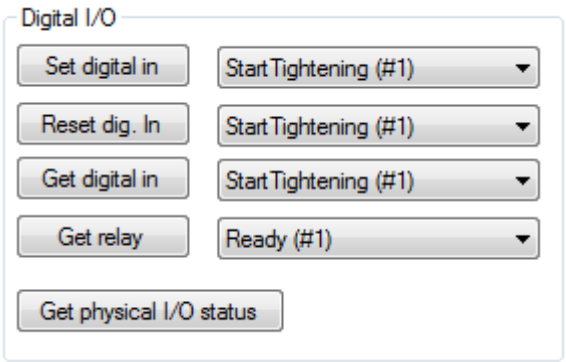
4.3.1 Select Bseq button

Send a select Batch sequence request to the controller by clicking this button. The edit box to the right selects which Batch sequence that will be selected.

4.3.2 Get Bseq list

Send a request to download a list of available Batch Sequences by clicking this button.

4.4 Digital I/O group



4.4.1 Set digital in button

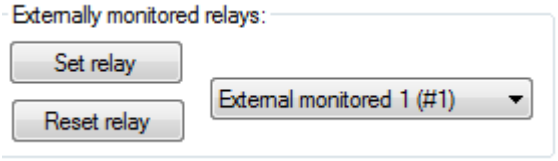
Select function to set in the list to the right and click this button to trigger corresponding function in the controller. This corresponding input will be set as high.

4.4.2 Reset dig. in button

Select function to reset in the list to the right and click this button to reset corresponding function in the controller. Most digital I/O functions are automatically reset (such as “start tightening”) when triggered, but toggles such as “Disable tool” needs to be manually reset when the function is no longer needed.

4.5 Externally monitored relays

Externally monitored relays are used to manipulate the digital outputs of the controller without linking them to a specific function.



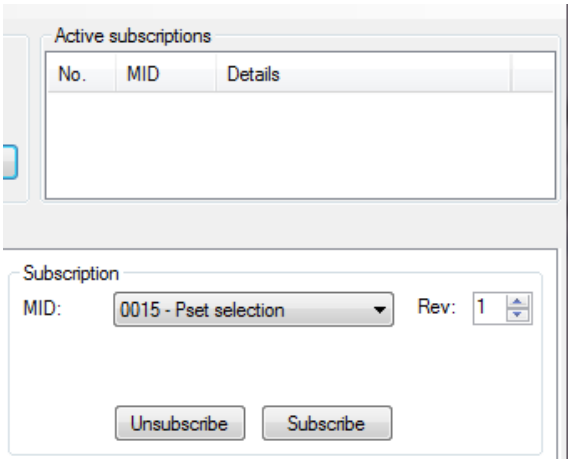
4.5.1 Set relay button

Sets an external monitor signal high. External monitored 1-10 can be selected from the dropdownlist to the right. (Controller must link external monitored 1-10 to physical PINs for this message to have any effect.)

4.5.2 Reset relay button

Sets an external monitor signal low. External monitored 1-10 can be selected from the dropdownlist to the right. (Controller must link external monitored 1-10 to physical PINs for this message to have any effect.)

4.6 Subscriptions



4.6.1 Subscribe button

Select which message to subscribe to by selecting a MID from the dropdown list. Select revision and press Subscribe.

The following messages can be subscribed to:

1) Pset selected (MID 0015)
2) Bseq selection (MID 0035)
3) Alarm (MID 0071)
4) Ext. monitored inputs (MID 0211)
5) Relay Function (MID 0217)
6) Digital Input Function (MID 0221)
7) Tightening graphs (MID 0900)
8) Tightening results (MID 1201)

For subscription of relay function and digital input function one must also select a specific function to subscribe to. I.e. MID 0217 and 0221 will not automatically subscribe to all functions

4.6.2 Unsubscribe button

Select which message to unsubscribe to by selecting a MID from the dropdown list and press Unsubscribe.

4.6.3 Active subscriptions

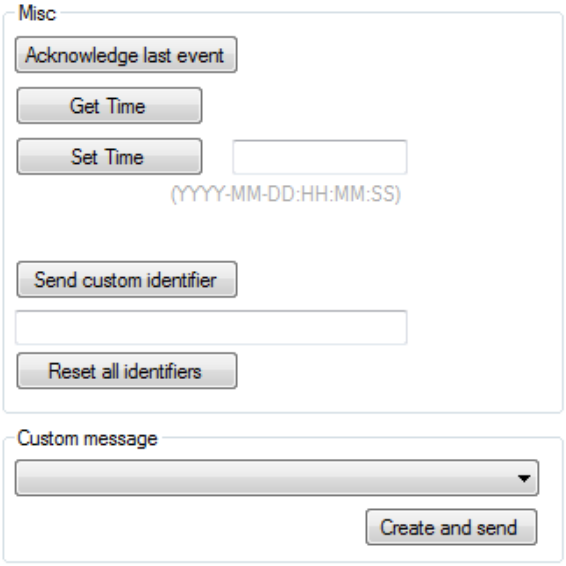
All active subscriptions will be displayed in this list.

4.7 Tool group

4.7.1 Get tool info

Send a request to download tool information by clicking this button.

4.8 Misc group



4.8.1 Acknowledge last event button

Acknowledge an active event by clicking this button.

4.8.2 Get Time button

Get the current controller time by clicking this button.

4.8.3 Set Time button

Enter requested time into the edit box to the right and click this button to set the time in the controller.

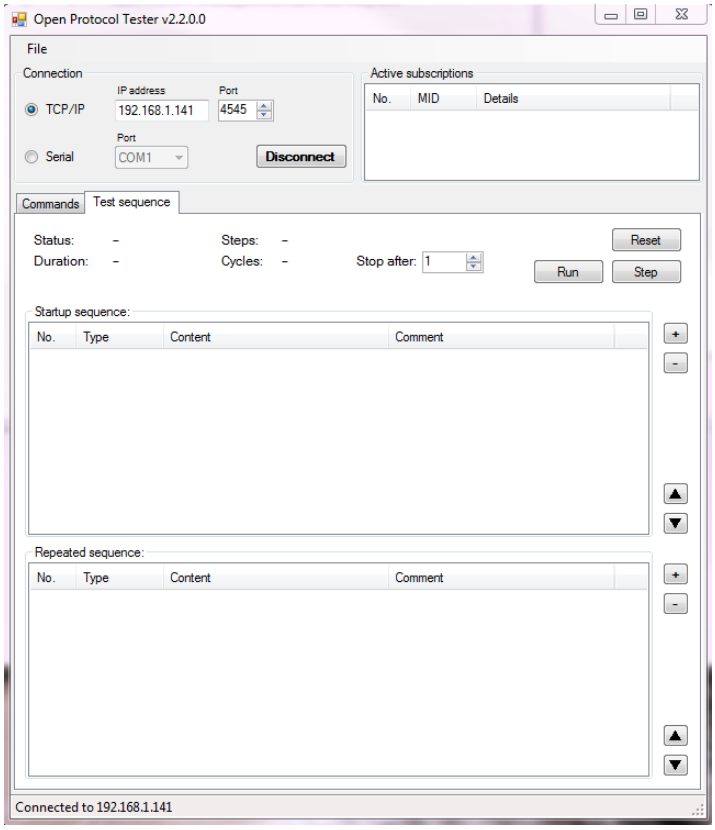
4.8.4 Send Custom Identifier

Send an identifier to the controller by clicking this button.

4.8.5 Reset All Identifiers

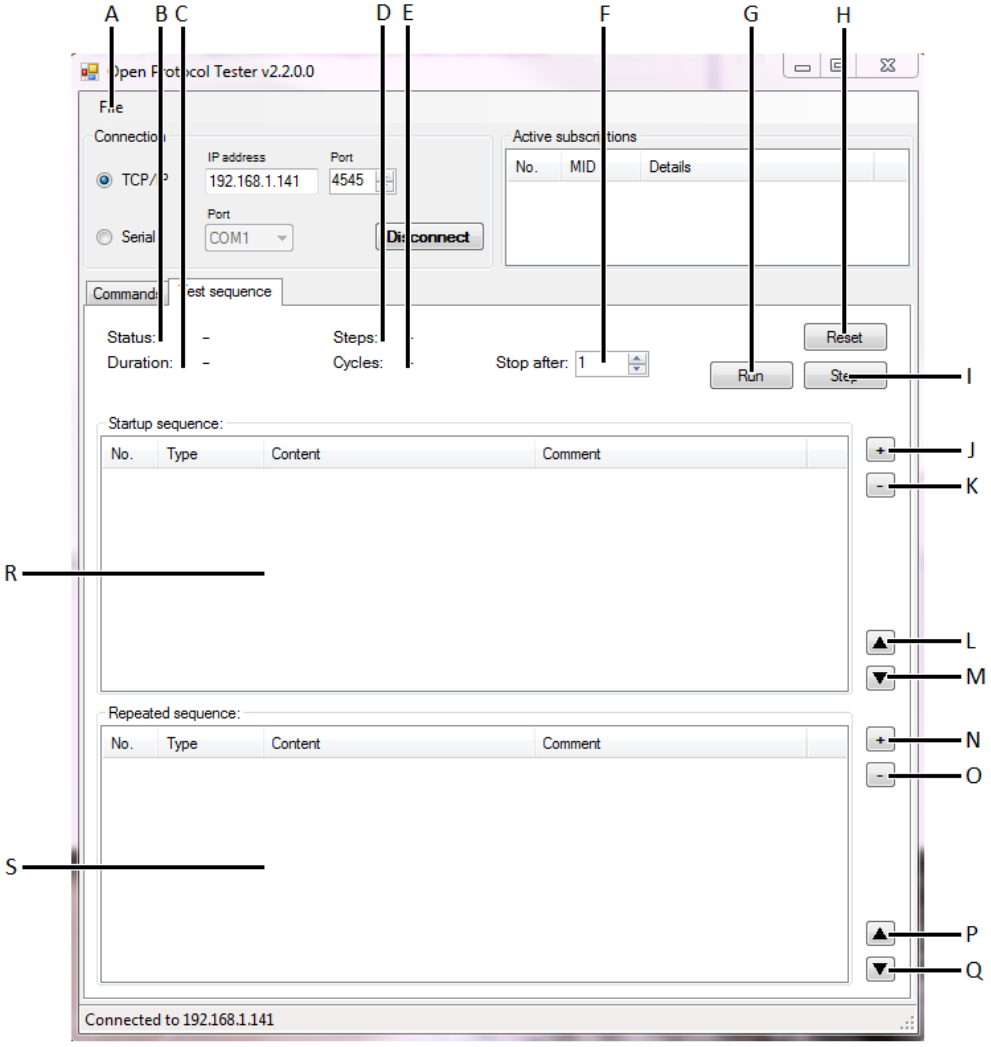
Send a request to clear all previously set identifiers in the controller.
(This will clear all stored Custom IDs)

4.9 Test sequence



The test sequence is used to help the user setup a series of messages that can run repeatedly for a set number of cycles.

4.9.1 Overview



Letter	Function	Description
A	File	Via *File* it is possible to save and load sequences. All sequences are saved in an .xml format.
B	Status	The *Status* displays the current status of the sequence • '-' – Default at startup • Running – Sequence is currently executing • Paused – Sequence has been paused on a specific instruction and can continue from that point without having to restart the sequence. • Stopped – Sequence has been stopped. It is not possible to pick up where the sequence stopped so the sequence must be restarted to be able to run again.
C	Duration	Time sequence has been active
D	Steps	Number of sent instructions
E	Cycles	Number of times *Repeated sequence[S]* has been run
F	Stop after	Number of cycles to run

G	Run	Run all instructions in order automatically
H	Reset	Abort ongoing sequence and jump back to start position
I	Steps	Execute instruction and move on to the next one
J	+	Add instruction for *Startup sequence[R]* *See 4.8.2 Add instructions*
K	-	Remove instruction from *Startup sequence[R]*
L	Up	Move selected instruction up in *Startup sequence[R]* list
M	Down	Move selected instruction down in *Startup sequence[R]* list
N	+	Add instruction for *Repeated sequence[S]* *See 4.8.2 Add instructions*
O	-	Remove instruction from *Repeated sequence[S]*
P	Up	Move selected instruction up in *Repeated sequence[S]* list
Q	Down	Move selected instruction down in *Repeated sequence[S]* list
R	Startup sequence	Group of instructions that will only be executed once
S	Repeated sequence	Group of instructions that will be repeated according to the selected number of cycles [F]

4.9.2 Add instructions

It is possible to add six different types of instructions:

- Send Message
- Expect Acknowledge
- Expect Message
- Wait Input State
- Wait Relay State
- Delay

For each instruction it is possible to enter a comment which will be displayed in the Start/Repeated sequence view.

4.9.2.1 Send message

Parameter	Comment
ClearReceiverQueueOnSend	Read only, always *True.*
Comment	User defined text that will be displayed in sequence view.
EntryType	Read only, always *Instruction.*
Message	Select which MID to send. (All available MIDs are displayed in the drop down list.)
Timeout (ms)	If the instruction doesn't finish executing within this timeframe the sequence will be paused. If that happens a popup will appear with the option of aborting the sequence (yes) or to try again (no).

4.9.2.2 Expect acknowledge

Parameter	Comment
AcknowledgeMID	MID to be acknowledged.
AllowNegativeAck	False = Will not accept receiving a NACK. If that happens a popup will appear with the option of aborting the sequence (yes) or to try again (no). True = Will continue if NACK is received.
AllowPositiveAck	False = Will not accept receiving an ACK. If that happens a popup will appear with the option of aborting the sequence (yes) or to try again (no). True = Will continue if ACK is received.
Comment	User defined text that will be displayed in sequence view.
EntryType	Read only, always *Instruction.*
Timeout (ms)	If the instruction doesn't finish executing within this timeframe the sequence will be paused. If that happens a popup will appear with the option of aborting the sequence (yes) or to try again (no).

4.9.2.3 Expect Message

Parameter	Comment
Comment	User defined text that will be displayed in sequence view.

EntryType	Read only, always *Instruction.*
ExpectMID	MID of expected message.
StopOnWrongMID	True = If the instruction doesn't receive the expected MID the sequence will be paused. If that happens a popup will appear with the option of aborting the sequence (yes) or to try again (no). False = Will wait for correct MID.
Timeout (ms)	If the instruction doesn't finish executing within this timeframe the sequence will be paused. If that happens a popup will appear with the option of aborting the sequence (yes) or to try again (no).

4.9.2.4 Wait Input State

Do not forget to subscribe to selected input.

Parameter	**Comment**
Comment	User defined text that will be displayed in sequence view.
EntryType	Read only, always *Instruction.*
Timeout (ms)	If the instruction doesn't finish executing within this timeframe the sequence will be paused. If that happens a popup will appear with the option of aborting the sequence (yes) or to try again (no).
WaitFunction	Specifies which input function to wait for.
WaitState	Specifies which level the input should take. False = Low True = High

4.9.2.5 Wait Input State

Do not forget to subscribe to selected output (relay).

Parameter	**Comment**
Comment	User defined text that will be displayed in sequence view.
EntryType	Read only, always *Instruction.*
Timeout (ms)	If the instruction doesn't finish executing within this timeframe the sequence will be paused. If that

	happens a popup will appear with the option of aborting the sequence (yes) or to try again (no).
WaitFunction	Specifies which output (relay) function to wait for.
WaitState	Specifies which level the output (relay) should take. False = Low True = High

4.9.2.1 Delay

Do not forget to subscribe to selected input.

Parameter	**Comment**
Comment	User defined text that will be displayed in sequence view.
Duration (ms)	Time to wait before executing the next step.
EntryType	Read only, always *Instruction.*
Timeout (ms)	If the instruction doesn't finish executing within this timeframe the sequence will be paused. If that happens a popup will appear with the option of aborting the sequence (yes) or to try again (no).