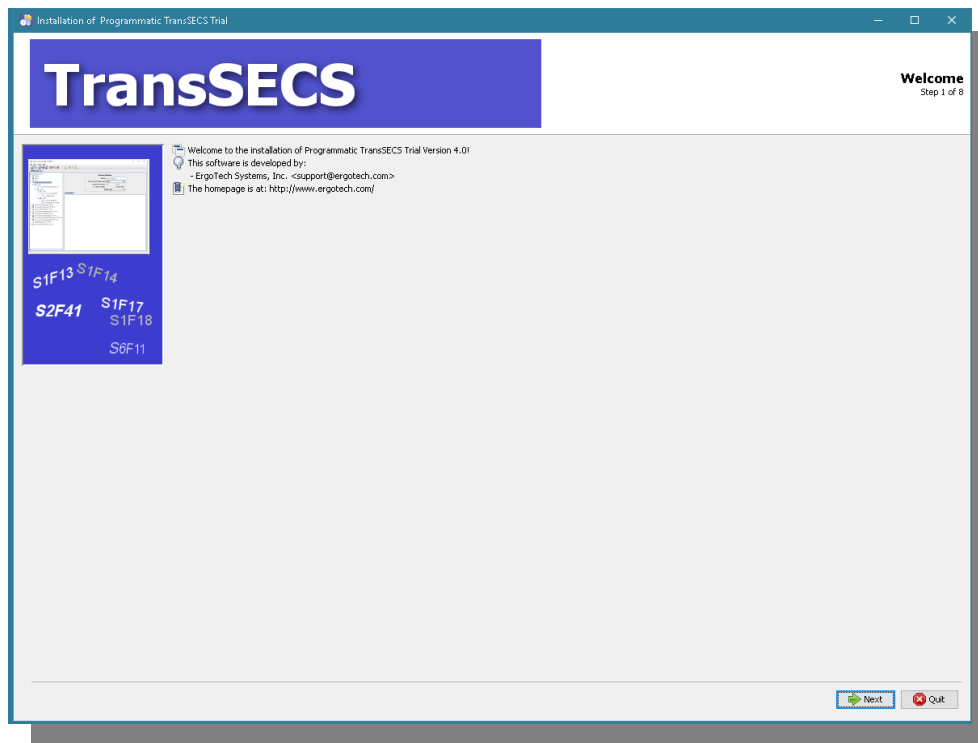




Using Devices TransSECS for a Modbus PLC based SECS/GEM Interface

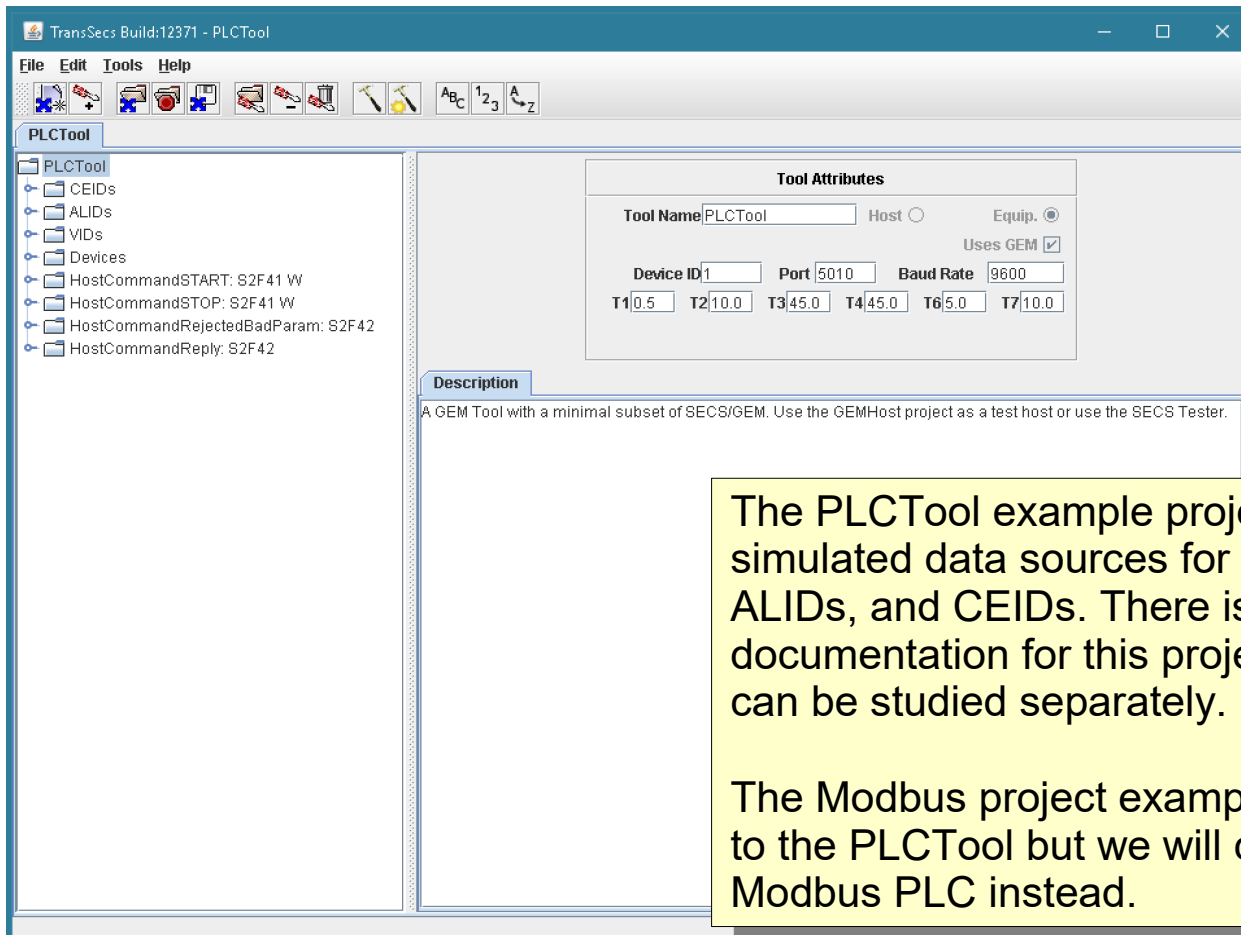
Using Devices TransSECS to create a SECS/GEM tool interface with data connections to a Modbus PLC

Installing TransSECS



Double click on the installer and click **Next** when prompted. Once installed, start the TransSECS Devices Builder application
(MIStudioSuite/TransSECS/Builder/TransSECSDevices.exe)

The example PLCTool will be loaded when you start the TransSECS Devices Builder. Next we will load the ModbusPLCTool example.



The PLCTool example project uses simulated data sources for the VIDs, ALIDs, and CEIDs. There is specific documentation for this project which can be studied separately.

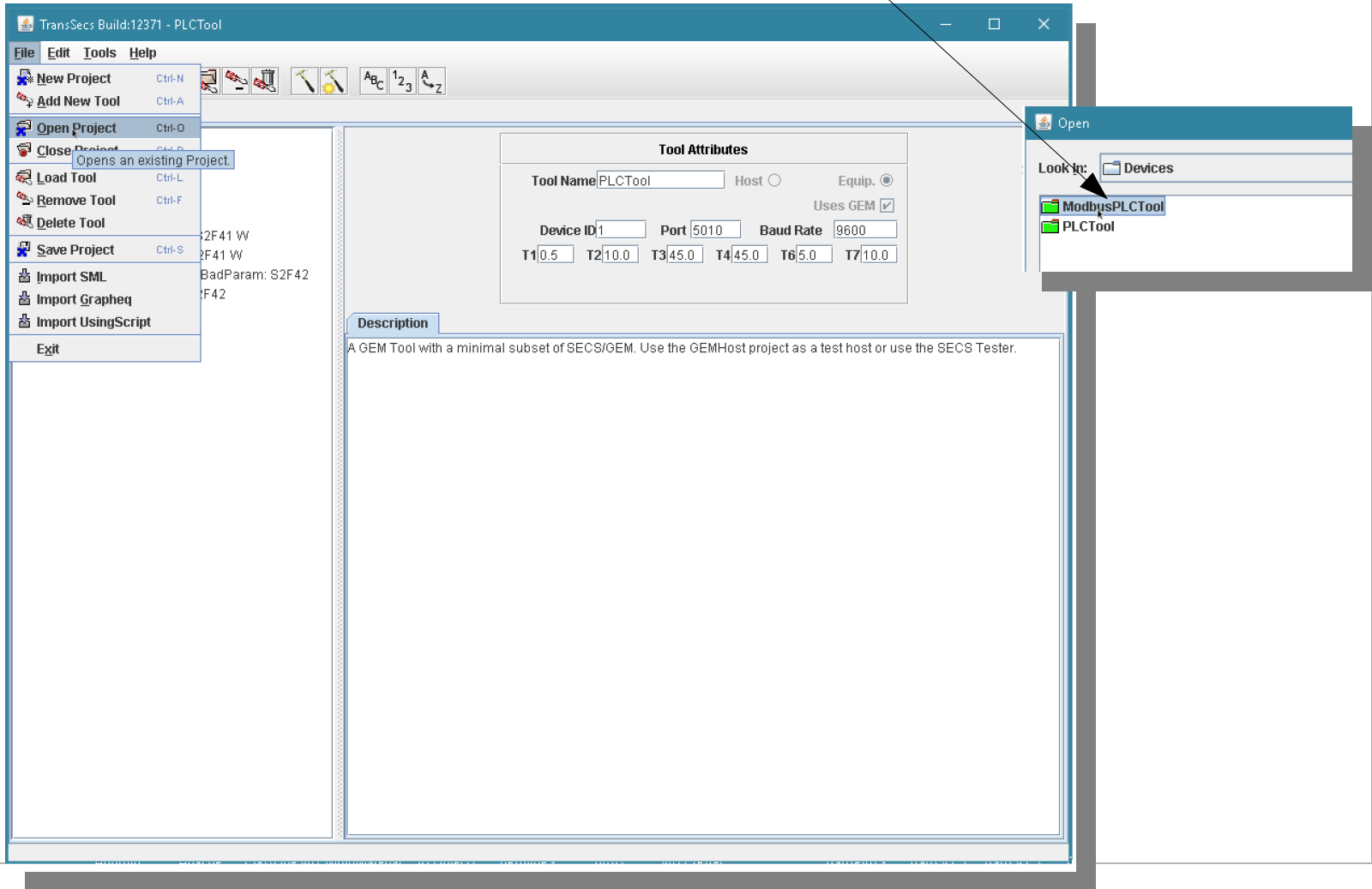
The Modbus project example is similar to the PLCTool but we will connect to a Modbus PLC instead.



33.5
35.7
32.2
30.9
28.2

ErgoTech Systems, Inc.

Load the example ModbusPLCTool project using Open Project from the File Menu. Browse for the **green project folder** with the name "ModbusPLCTool" and double click on it.





33.5
35.7
32.2
30.9
28.8

This is the ModbusPLCTool Example loaded in TransSECS

The tool interface defaults to running on port 5010 (HSMS)

and Device ID 1

Tool Attributes

Tool Name: ModbusPLCTool Host: Equip.:

Uses GEM:

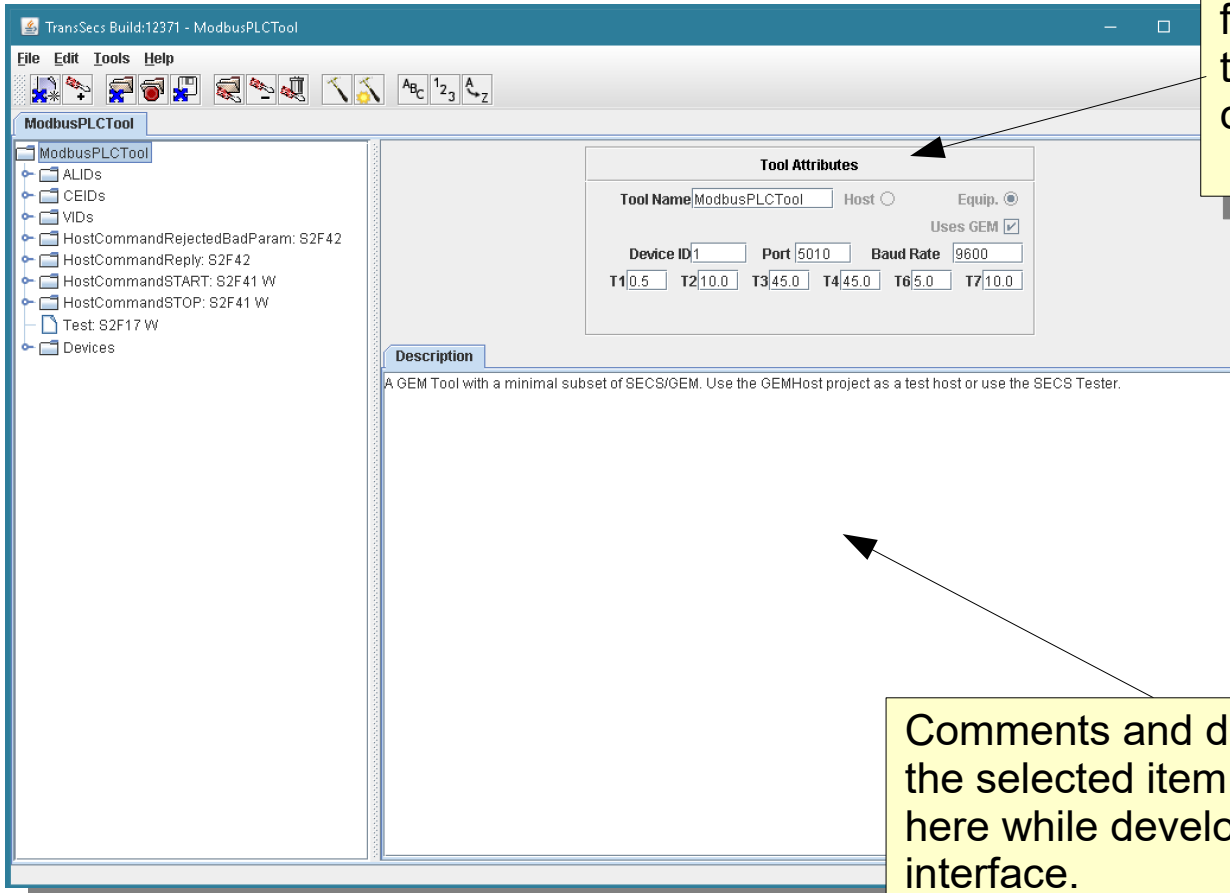
Device ID: 1 Port: 5010 Baud Rate: 9600

T1: 0.5 T2: 1.0 T3: 45.0 T4: 45.0 T6: 5.0 T7: 10.0

Description

A GEM Tool with a minimal subset of SECS/GEM. Use the GEMHost project as a test host or use the SECS Tester.

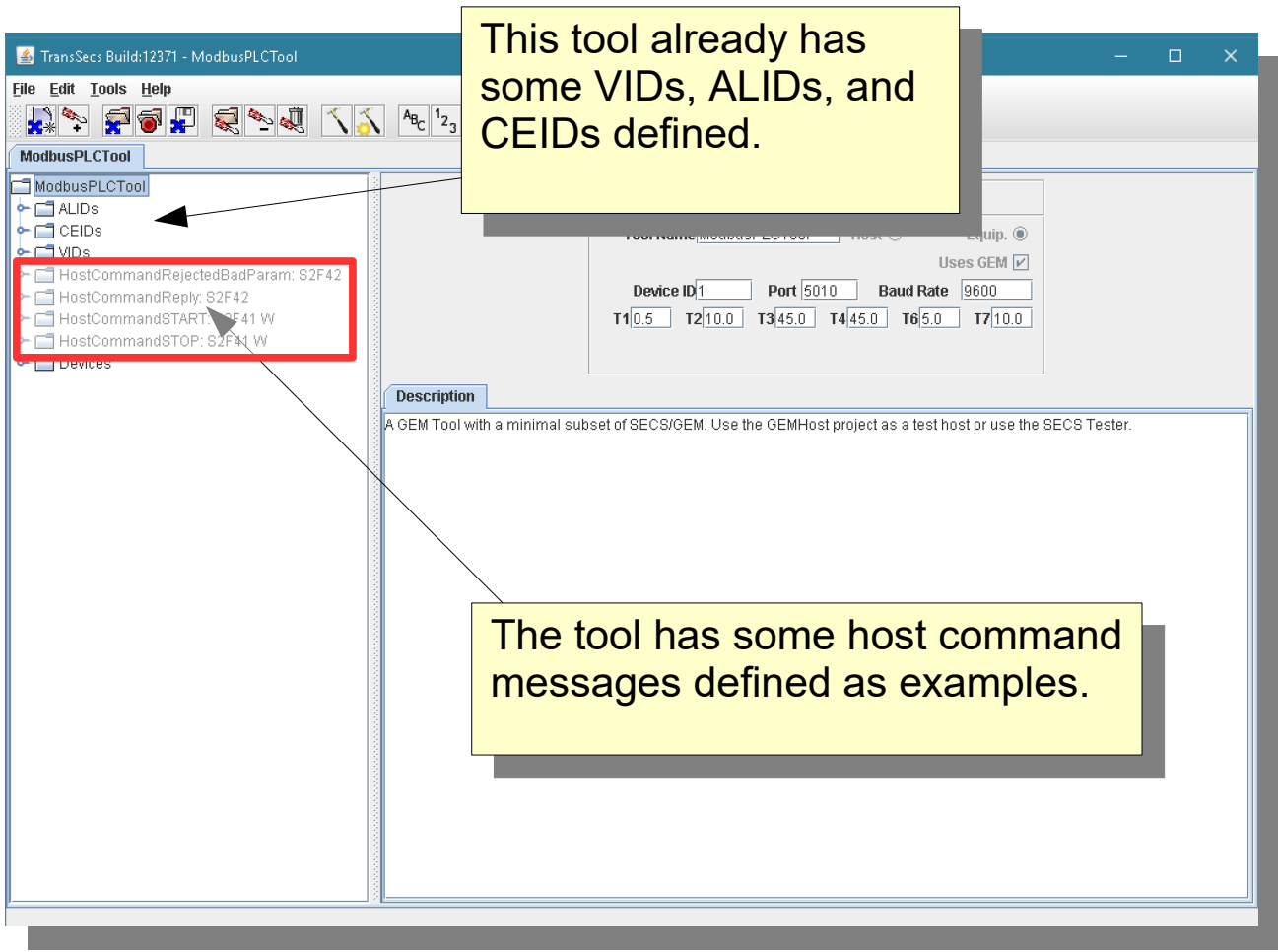
This is the ModbusPLCTool Example loaded in TransSECS



Data and other details for the selected item in the left hand panel are configured here

Comments and descriptions of the selected item can be entered here while developing the interface.

This is the ModbusPLCTool Example loaded in TransSECS



The screenshot shows the ModbusPLCTool interface within the TransSECS environment. The left sidebar contains a tree view with folders for ALIDs, CEIDs, VIDs, and Devices. A red box highlights a list of host command messages: HostCommandRejectedBadParam: S2F42, HostCommandReply: S2F42, HostCommandSTART: S2F41 W, and HostCommandSTOP: S2F41 W. The main window displays configuration fields for Device ID, Port, Baud Rate, and timing parameters (T1-T7). A description at the bottom states: "A GEM Tool with a minimal subset of SECS/GEM. Use the GEMHost project as a test host or use the SECS Tester."

This tool already has some VIDs, ALIDs, and CEIDs defined.

The tool has some host command messages defined as examples.

Scripting can be added to any message or data item. For example, in this project a script has been added to the Host Command START message to analyze the data and send a correct reply message to the host.

The screenshot displays the ModbusPLCTool application window. The title bar reads "TransSecs Build:12371 - ModbusPLCTool". The interface includes a menu bar (File, Edit, Tools, Help) and a toolbar with various icons. On the left, a tree view shows the project structure under "ModbusPLCTool", with "HostCommandSTART: S2F41 W" selected. The main area is divided into two panes. The top pane, titled "Message Attributes", shows the following configuration for the selected message:

- Name: HostCommandSTART
- Auto Response Message: <none>
- Stream & Function: S2 F41
- Expects Reply
- Snoop Only
- Remote Command
- Send Err Msg
- Message Direction: In Message

The bottom pane, titled "Script", contains the following code:

```

4 print("Host Command START");
5 //1. START has been requested. Parse parameter(s) in the message to get the PPID.
6 //2. Store PPID in RecipeName device register
7 //3. Set STARTFlag in PLC to 99
8 //4. Wait up to 2000 ms for PLC to signal OK or error in register STARTFlag
9 // 0 or 4(OK), 2 (cannot perform now), 5 (already in that state)
10 //5. Send reply to host
11
12
13 var TransSecsController = Java.type("com.ergotech.transsecs.secs.TransSecsController");
14 try {
15
16 secsInterface=TransSecsController.findController("ModbusPLCTool");
17 hostcommand=secsInterface.getMessageBean("HostCommandSTART");
18
19 //recipe name is the second element of the first parameter array
20 //Note:if CPName Validation is not used, you need to manually check that there is a
21 //single CPName which is "PPID". This example assumes this host message matched HostComm
22 recipeName = hostcommand.getCommandParams().elementAt(0).elementAt(1);
23
24 /Devices/DemoServers_Servers/RecipeName->setStringValue(recipeName);
25

```




33.5
35.7
32.2
30.9
28.2

Select the Modbus TCP Devices Node

The screenshot shows the ModbusPLCTool application window. The left sidebar displays a tree view of the project structure, with the 'ModbusTCP' node under 'Devices' selected. The main window is divided into two panes: 'Basic' and 'Expert'. The 'Basic' pane shows the configuration for the 'Modbus TCP Device Server' with the following parameters:

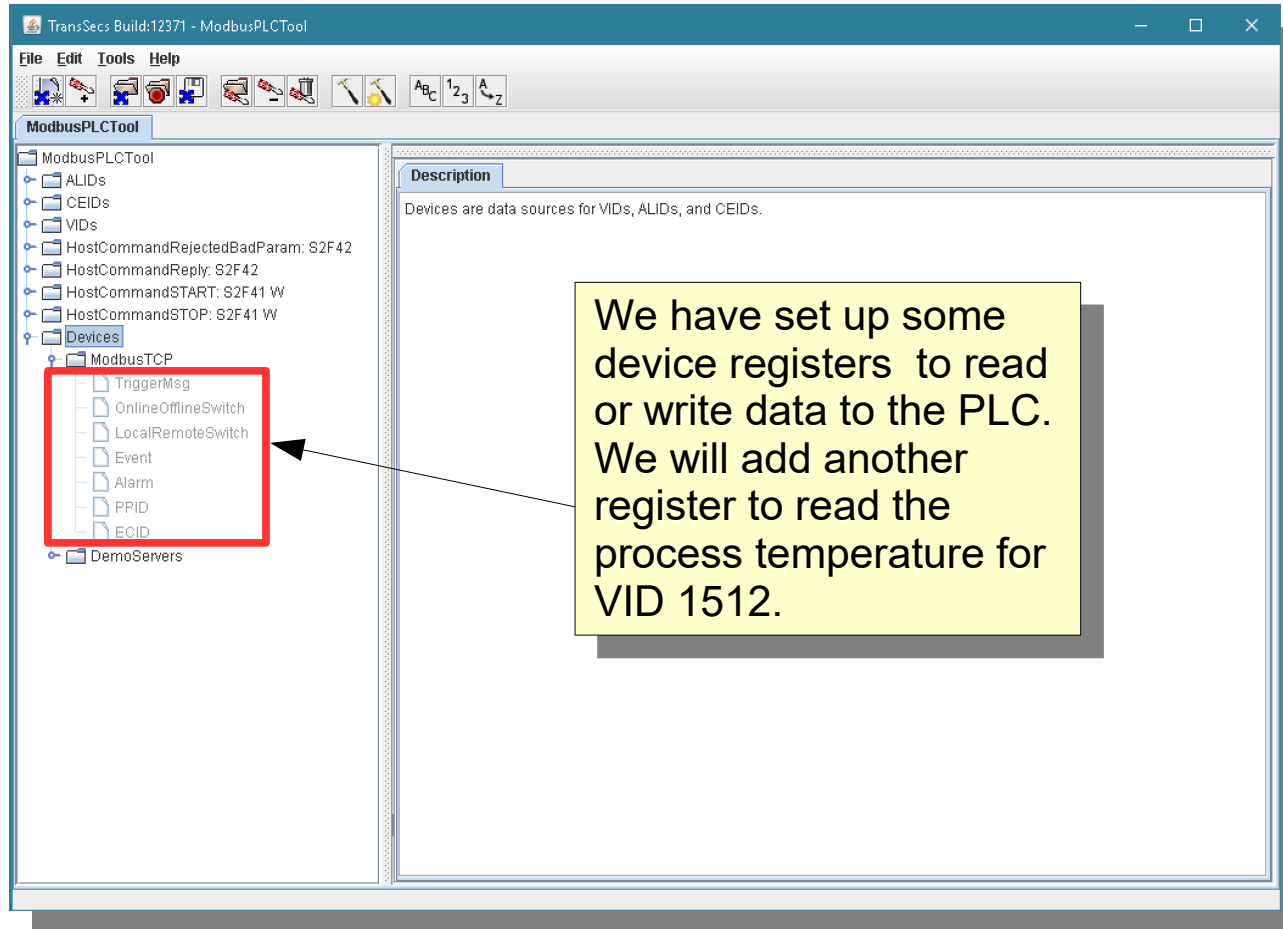
Property	Value
Name	ModbusTCP
Response Timeout	250
Simulating ?	False
Hostname	192.168.5.98
Maximum Socket Connections	5

The 'Expert' pane is currently empty. Below the configuration panes is a 'Description' section with the following text:

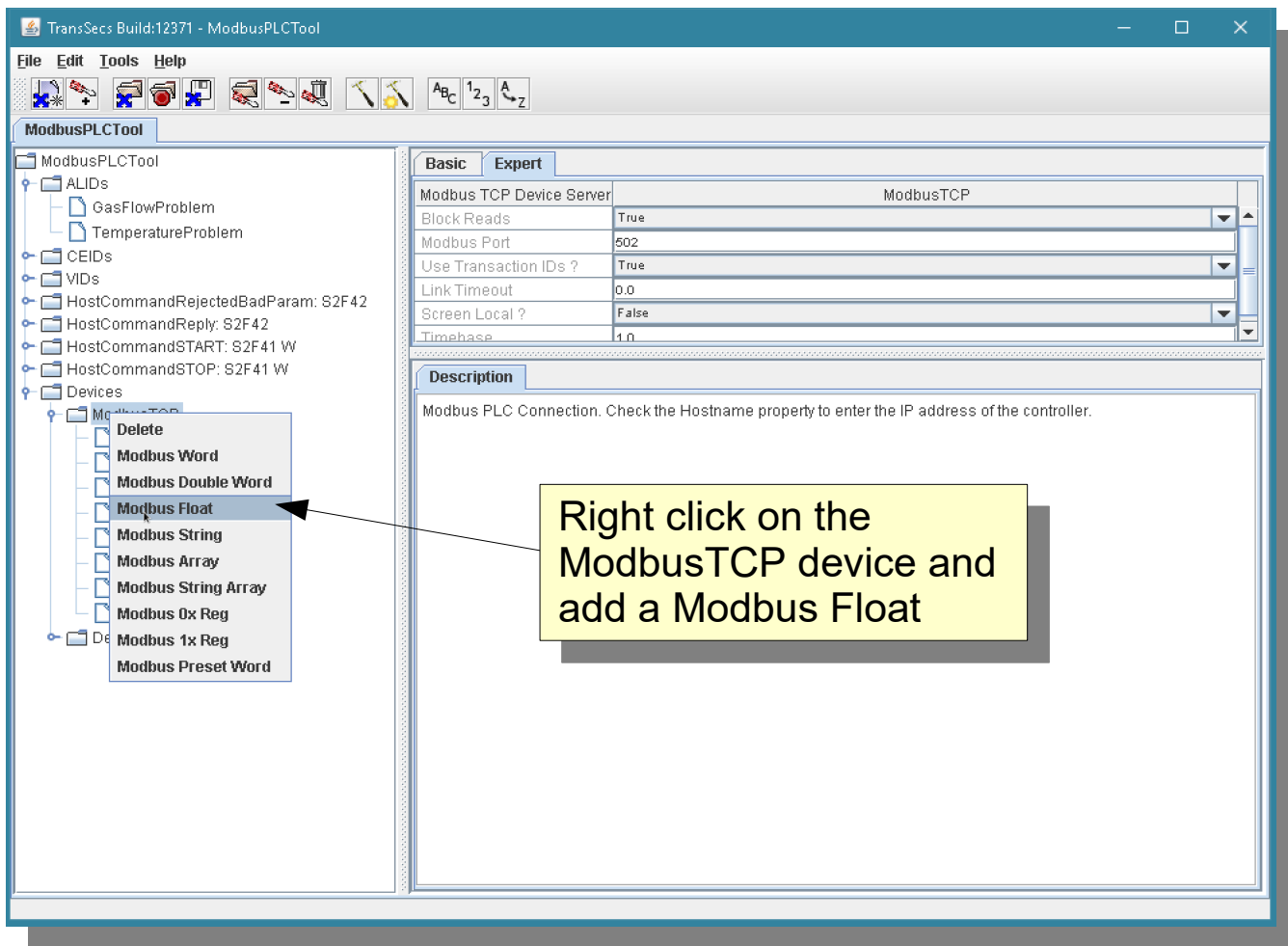
Modbus PLC Connection. Check the Hostname property to enter the IP address of the controller.

The connection parameters and configuration for the Modbus TCP Device Server is set here.

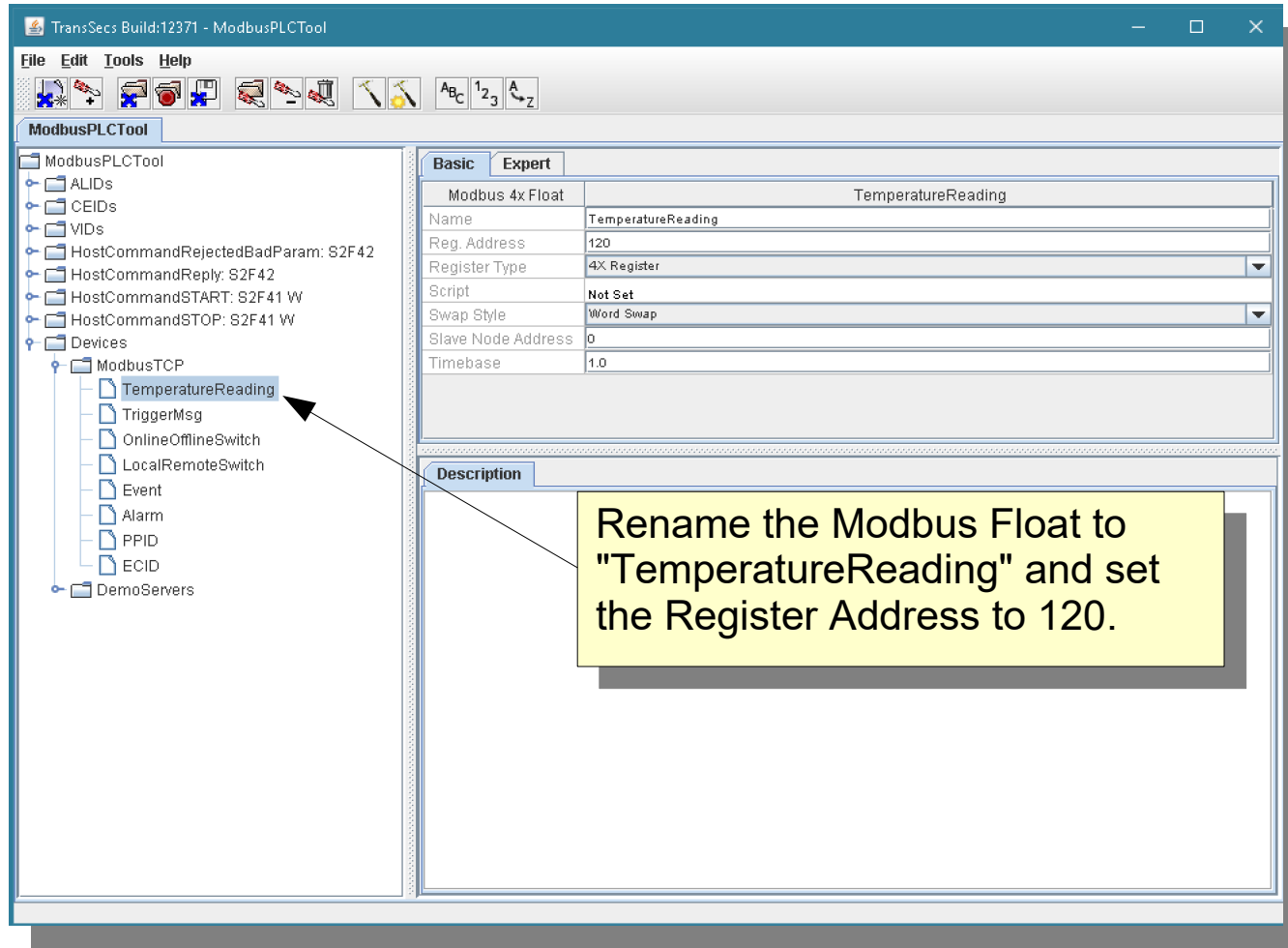
Expand the Modbus TCP Devices Node



Add a Modbus Float for the Process Temperature Reading



Add a Modbus Float for the Process Temperature Reading



The screenshot shows the ModbusPLCTool software interface. On the left is a tree view with a folder named 'ModbusTCP' containing a 'TemperatureReading' entry. On the right is a configuration panel for a 'Modbus 4x Float' with the following settings:

Modbus 4x Float	
Name	TemperatureReading
Reg. Address	120
Register Type	4X Register
Script	Not Set
Swap Style	Word Swap
Slave Node Address	0
Timebase	1.0

Below the configuration panel is a 'Description' section with a yellow callout box containing the text: "Rename the Modbus Float to 'TemperatureReading' and set the Register Address to 120." An arrow points from this callout to the 'TemperatureReading' entry in the tree view.



33.5
35.7
32.2
30.9
28.2

Assign the Temperature Reading Modbus Float to the Process Temperature VID (VID 1512)

The screenshot shows the ModbusPLCTool software interface. On the left, a tree view under 'VIDs' has 'ProcessTemperature' selected. The main window displays the configuration for this VID. The 'Name' is 'ProcessTemperature', 'VID' is '1512', 'Data Type' is 'Float-4', and 'VID Type' is 'SVID'. The 'Device Name' is set to 'ModbusTCP' and the 'Tag Name' is 'TemperatureReading'. A description at the bottom reads: 'An SVID with the process temperature value'. Three yellow callout boxes provide instructions: one points to the 'ProcessTemperature' node in the tree, another points to the 'Device Name' dropdown, and a third points to the 'TemperatureReading' tag in the list.

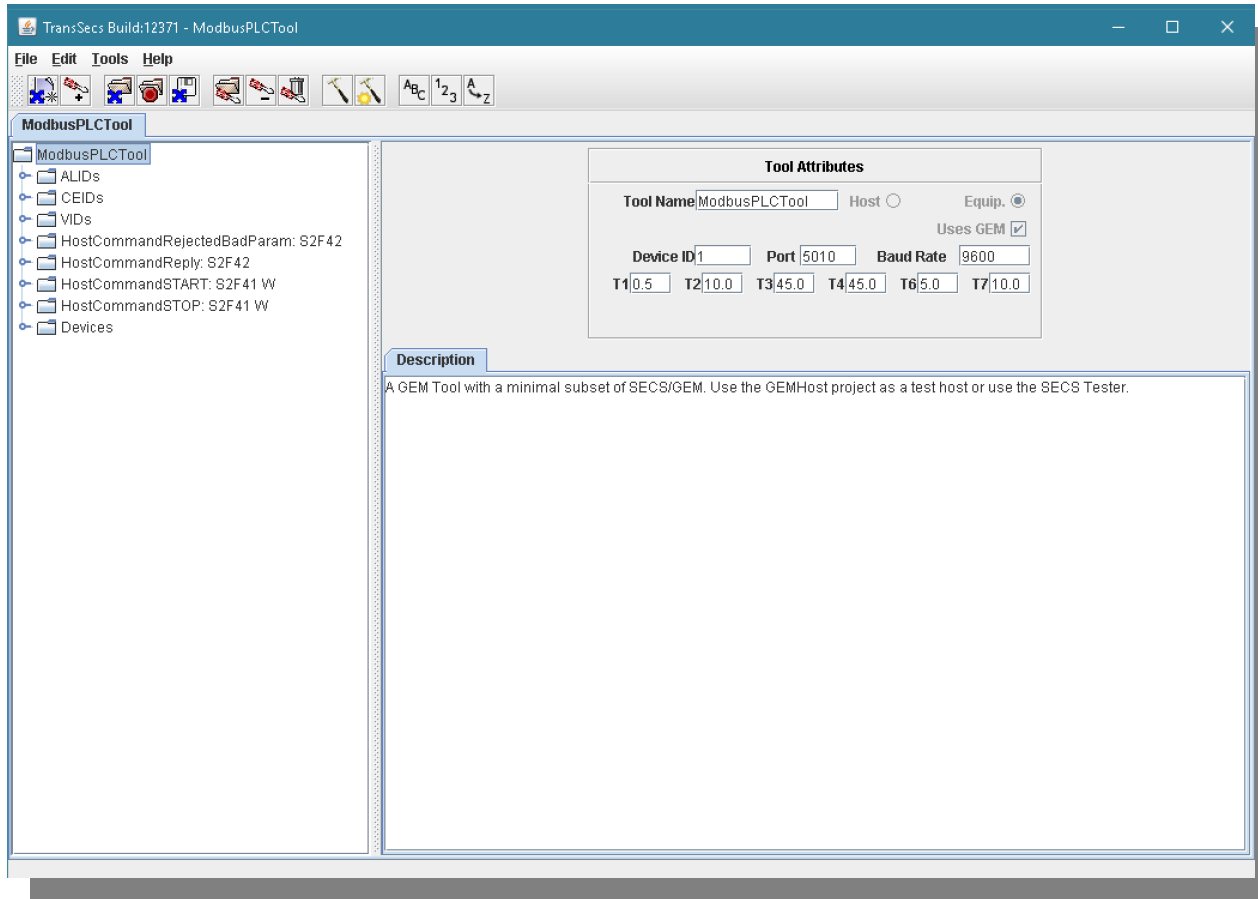
Select the VID node and find the ProcessTemperature VID

Change the Device Name to ModbusTCP

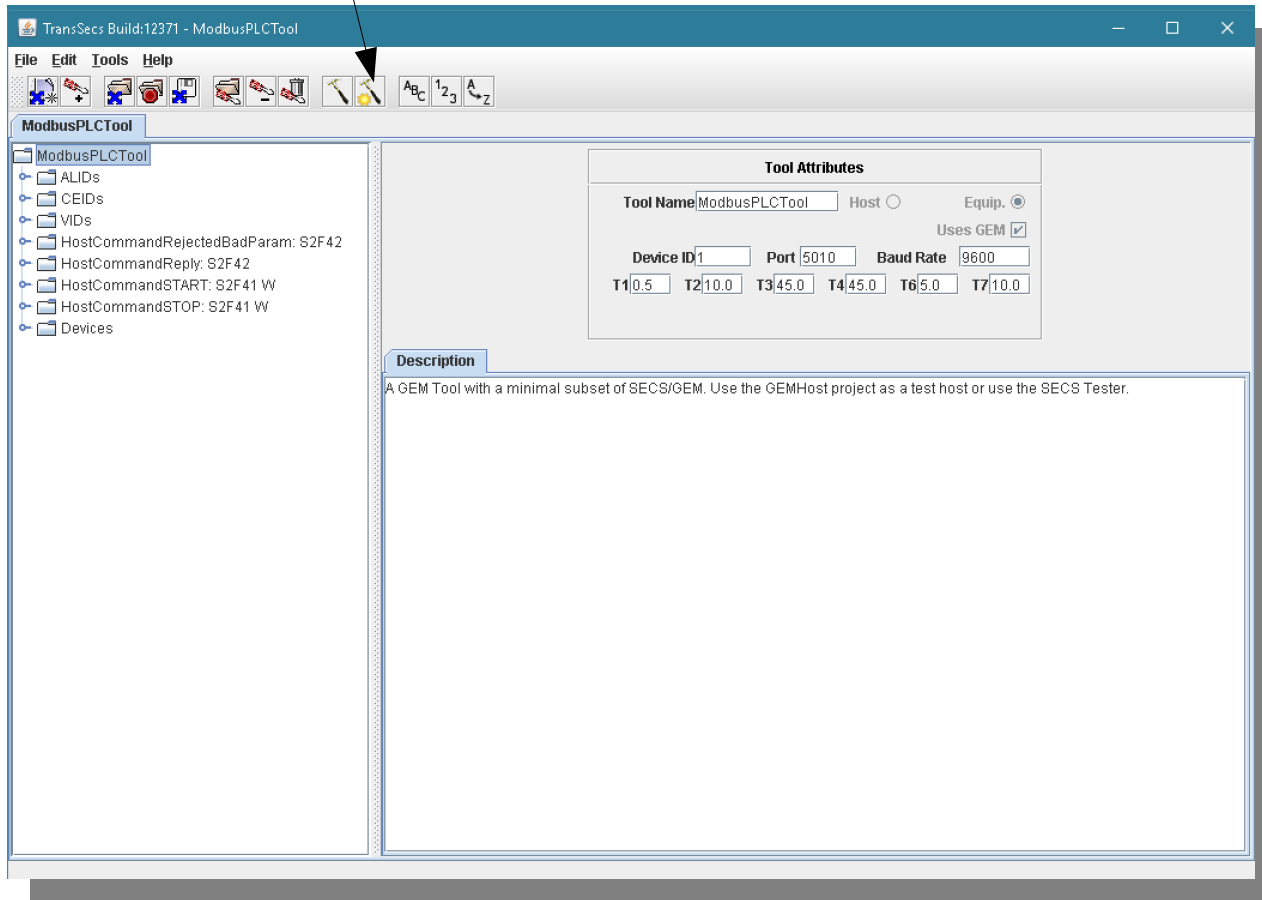
Select "TemperatureReading" from the Tag Name list



Now it is time to test the changes to the project. We will build the project then run it and test it with a Host application



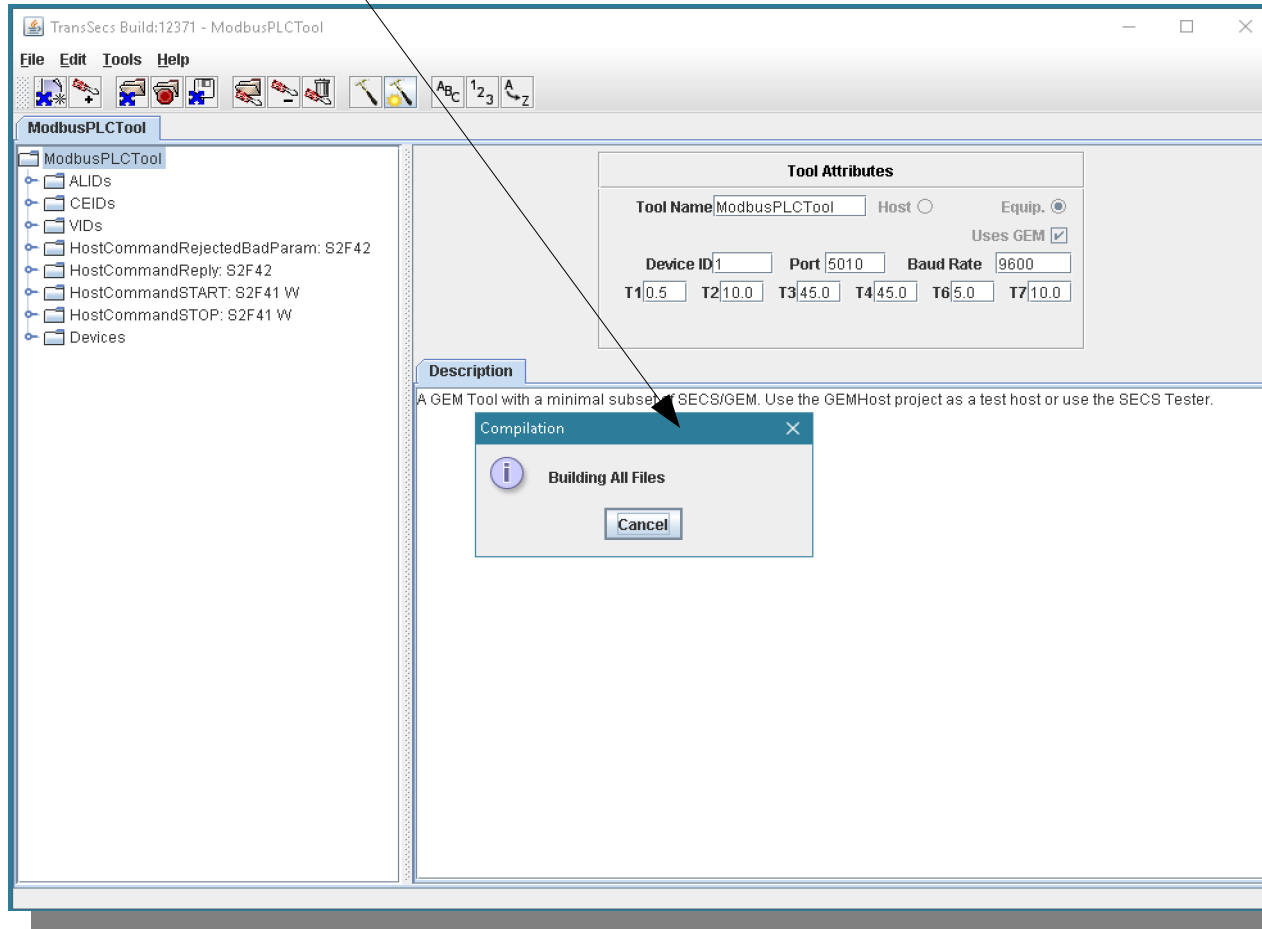
Press the Hammer/Star button to build the project



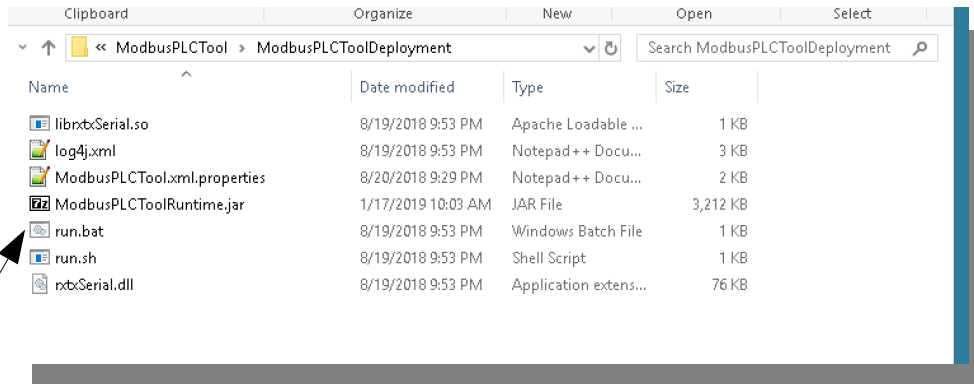


33.5
35.7
32.2
30.9
28.8

When the "Compilation" popup closes the build is complete.
This may take a minute or so.



After the code is generated the code for the tool will be in the project's ModbusPLCTool/ModbusPLCToolDeployment directory.



Double click on run.bat for Windows to run the tool.

Everything you need to run on Windows is in this directory. For Linux systems you will need install rxtxSerial on your Linux system and use run.sh

Run the SECS/GEM Interface

Use the run.bat file in the deployment directory. You may need to edit the path to the jre if you move the file location.

```
F:\TransSECSProjects\Devices\ModbusPLCTool\ModbusPLCToolDeployment>rem Ensure there is a jre on the path. Copy the
jre provided or install a jvm

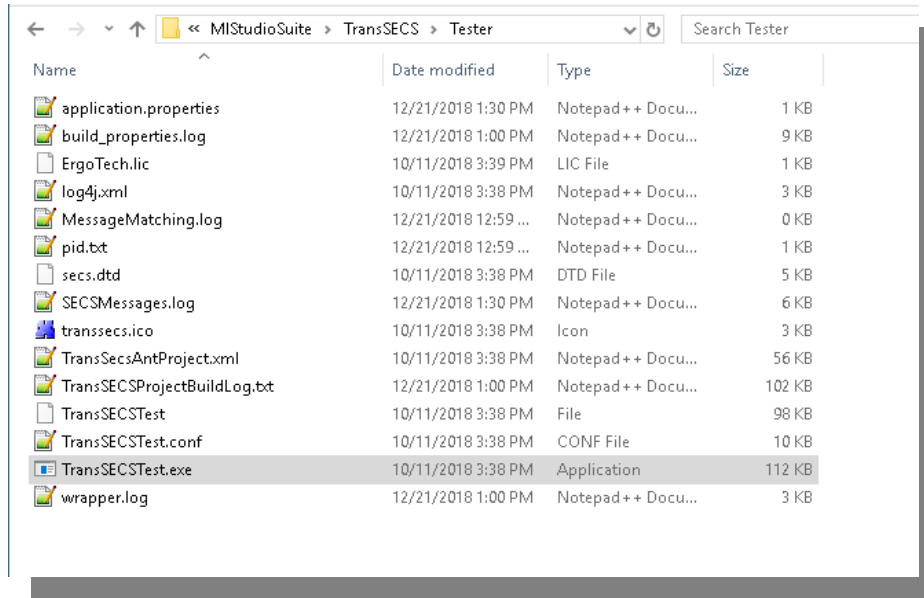
F:\TransSECSProjects\Devices\ModbusPLCTool\ModbusPLCToolDeployment>set classpath=.;./ModbusPLCToolRuntime.jar

F:\TransSECSProjects\Devices\ModbusPLCTool\ModbusPLCToolDeployment>java deploy.ModbusPLCTool.EquipmentController
user.dir F:\TransSECSProjects\Devices\ModbusPLCTool\ModbusPLCToolDeployment
ECID SetPoint (2000) value changed
Started tool ModbusPLCTool on port 5010 device id 1
```

When you run the generated run.bat, the tool's SECS/GEM will be running on Port 5010 and Device ID 1



Test the SECS/GEM Interface with TransSECS GEMHost using TransSECSTest.exe



This starts TransSECS with the GEMHost project which runs as a Host to test the tool interface.



33.5
35.7
32.2
30.9
28.8

ErgoTech Systems, Inc.

Run TransSECS as a Test Host

Set to LIVE mode before building the GEMHost project

TransSECS Build:12376 - GEMHost

File Edit Tools Help

GEMHost

Tool Attributes

Tool Name: GEMHost Host Equip.

Tool IP Addr...: localhost Uses GEM

Device ID1: Port: 5010 Baud Rate: 19200

T1|0.5 T2|10.0 T3|45.0 T4|45.0 T6|5.0 T7|10.0

Deployment Type: Test Host Only

Description

Sample Host for TransSECS. Use with the GEMTool project.

Build with the Hammer/
Star button

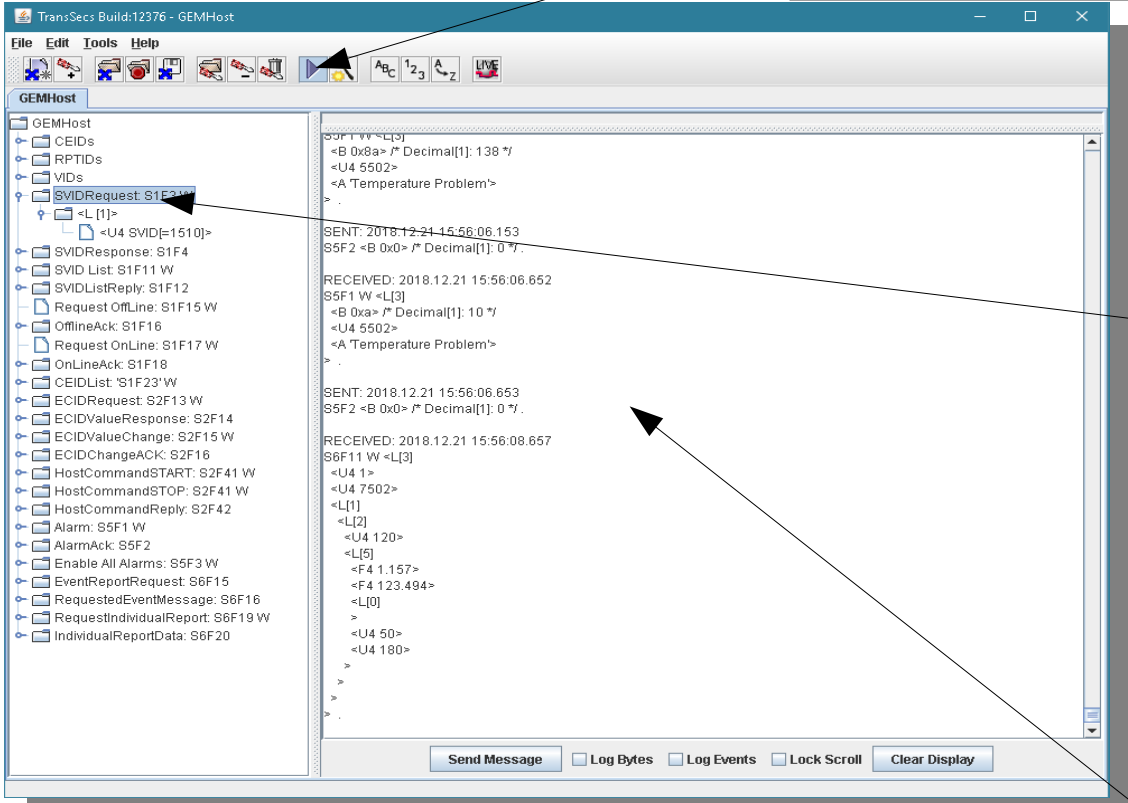
After the Host is built it will run and automatically connect to the GEMTool and set up event reports and enable alarms



33.5
35.7
32.2
30.9
28.2

Run TransSECS Test as a host

This arrow shows that the host is running



Select a primary (outgoing) message and Press "Send Message" button to send it

Messages sent and received will be shown in the Messages panel (when a message is selected)

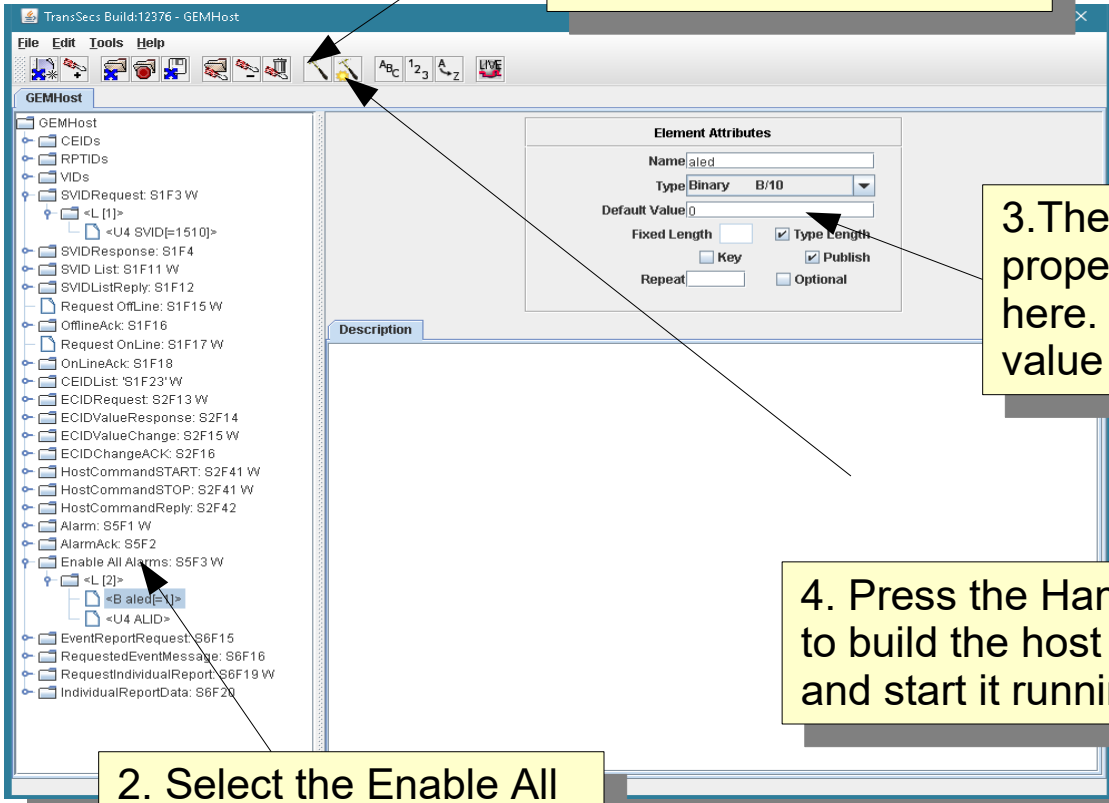


33.5
35.7
32.2
30.9

Temporarily Disable Alarms

1. Press the run button to take TransSECS out of run mode

You can make changes to the messages in the TransSECS Host and send the changed message



2. Select the Enable All Alarms message, expand it, and select the aled field

3. The message field properties are shown here. Change the aled value from 1 to 0.

4. Press the Hammer/Star button to build the host with this change and start it running again



33.5
35.7
32.2
30.9
28.8

ErgoTech Systems, Inc.

Temporarily Disable Alarms

Select Enable All Alarms and send the message with the Send Message button

The screenshot shows the 'GEMHost' interface. The left sidebar contains a tree view of system components, including 'CEIDs', 'RPTIDs', 'VIDs', and various 'SVIDRequest' and 'SVIDResponse' items. The 'Enable All Alarms: S5F3 W' item is selected. The main window displays a log of received and sent messages. The log shows several 'RECEIVED' and 'SENT' messages with timestamps and hexadecimal data. A 'Send Message' button is visible at the bottom of the window.

The message and the acknowledgment reply from the tool will be shown here



For testing it is easier to see the event and host messages if the alarm messages are disabled



33.5
35.7
32.2
30.9
28.8

Send a Host Message

Select Enable All Alarms and send the message with the Send Message button

The screenshot shows the 'TransSecs Build:12376 - GEMHost' application. The left-hand tree view contains various message categories such as CEIDs, RPTIDs, VIDs, and SVIDRequest. The 'Enable All Alarms: S5F3 W' option is highlighted. The main display area shows a series of log entries, including 'RECEIVED' and 'SENT' messages with their respective timestamps and hexadecimal data. At the bottom of the window, there are several control buttons: 'Send Message', 'Log Bytes', 'Log Events', 'Lock Scroll', and 'Clear Display'.

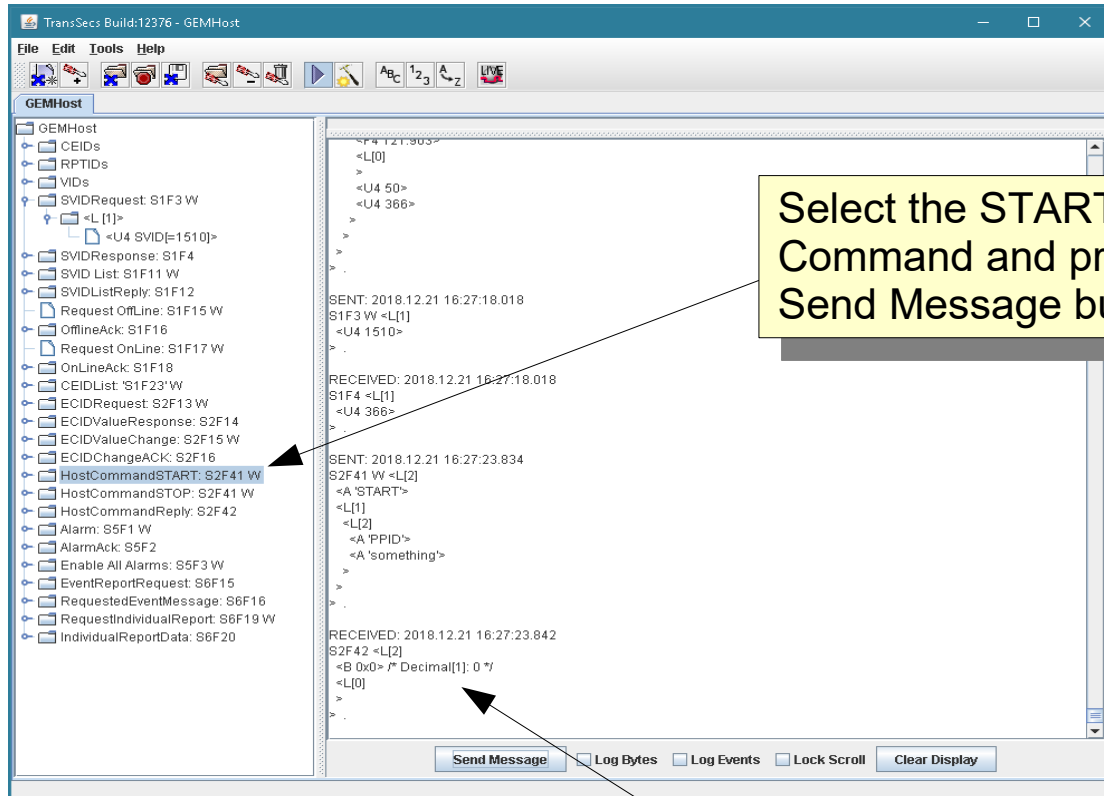
The message and the acknowledgment reply from the tool will be shown here



For testing it is easier to see the event and host messages if the alarm messages are disabled

Send a Host Message to the tool

The GEMTool is set up to handle the Host Command "START"



Select the START Host Command and press the Send Message button

The message was correctly handled by the tool and a reply was sent to the host



33.5
35.7
32.2
30.9
28.2

Check the Process Temperature VID

Send an S1F3 using VID 1512 to the tool to check the current temperature read from the PLC

The screenshot shows the 'ExtendedHost' window in TransSecs. On the left is a tree view of messages, and on the right is a detailed view of the selected 'SVID' (1510). The detailed view shows the following sequence of events:

- SENT: 2019.01.17 10:29:38.447
S1F3 W <L[1]
<U4 1512>
- RECEIVED: 2019.01.17 10:29:38.449
S1F4 <L[1]
<F4 233.2>
- SENT: 2019.01.17 10:29:52.618
S1F3 W <L[1]
<U4 1512>
- RECEIVED: 2019.01.17 10:29:52.619
S1F4 <L[1]
<F4 230.7>

At the bottom of the window, there are buttons for 'Send Message', 'Log Bytes', 'Log Events', 'Lock Scroll', and 'Clear Display'.

Here are two current temperature values from the PLC



That's it.

The code for the ModbusPLCTool interface is not complete but this simple example should give you some ideas and get you going. More VIDs, ALIDs, and CEIDs can be connected to live data in the PLC. Further host command message handling can be added, and also there are options for manually handling recipe messages, or you can set up simple automatic recipe handling.