pbsFIT

IEC104 Tester Software with IEC62351 and TLS layer

Ver 2.1

1. Introduction

pbsFIT is the IEC870-5-104 master tester software. With pbsFIT, you can easily connect to the IEC104 server and communicate with the RTU.

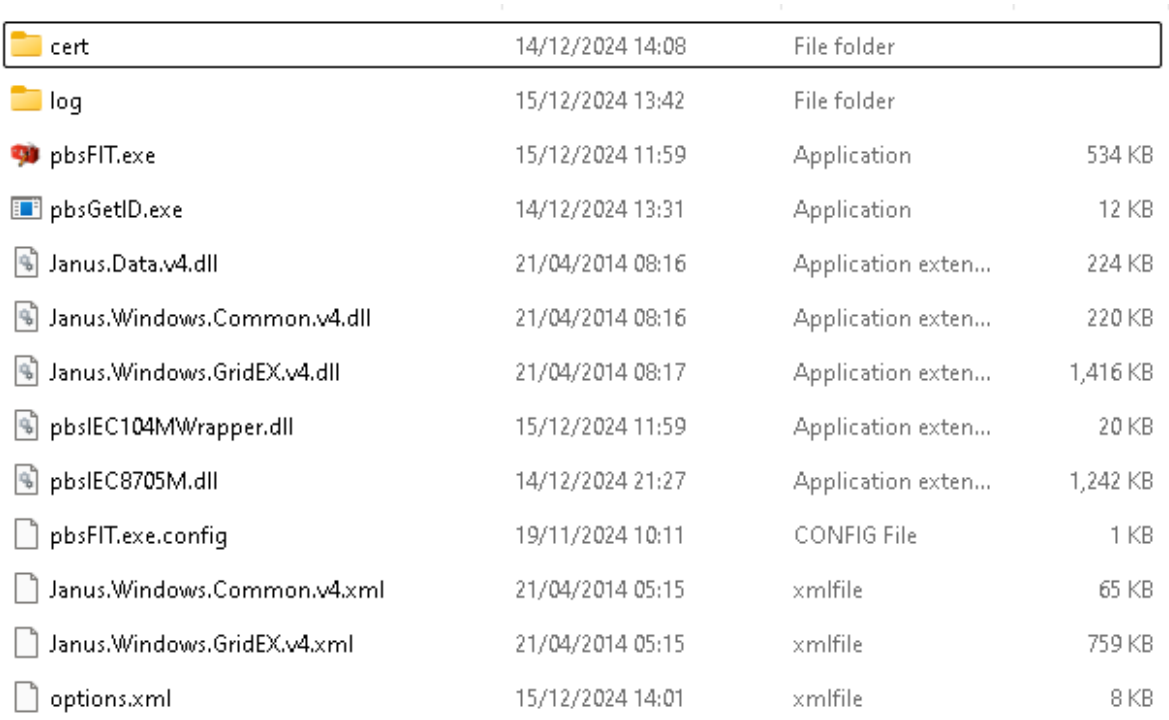
pbsFIT supports following functions :

1. Supported IEC104 data types :1,30,9,21,34,11,35,100,101,103,107,110,111,112,113 ,81,82,83,84,85,86
2. GI and CI with different grouping and test conditions
3. Read/Write function with test conditions
4. TCP and TLS for physical layer
5. IEC62351 SA layer
6. Frame logging and filtering for different layers

2. Installation

To run properly, pbsFIT requires Dot net Framework 4.81 and Visual C runtime for VS2022.

When you unzip the pbsFIT.zip file, you can see the following programs and folders:



cert	14/12/2024 14:08	File folder	
log	15/12/2024 13:42	File folder	
pbsFIT.exe	15/12/2024 11:59	Application	534 KB
pbsGetID.exe	14/12/2024 13:31	Application	12 KB
Janus.Data.v4.dll	21/04/2014 08:16	Application exten...	224 KB
Janus.Windows.Common.v4.dll	21/04/2014 08:16	Application exten...	220 KB
Janus.Windows.GridEX.v4.dll	21/04/2014 08:17	Application exten...	1,416 KB
pbsIEC104MWrapper.dll	15/12/2024 11:59	Application exten...	20 KB
pbsIEC8705M.dll	14/12/2024 21:27	Application exten...	1,242 KB
pbsFIT.exe.config	19/11/2024 10:11	CONFIG File	1 KB
Janus.Windows.Common.v4.xml	21/04/2014 05:15	xmlfile	65 KB
Janus.Windows.GridEX.v4.xml	21/04/2014 05:15	xmlfile	759 KB
options.xml	15/12/2024 14:01	xmlfile	8 KB

pbsFIT.exe is the main program to run.

options.xml is used to store the tester parameters.

pbsIEC870M.dll is the main library to manage IEC104 master, IEC62351 and TLS layers.

pbsgetID.exe is a simple tool to read the serial number of hdd/ssd. To use the licensed software, we need the serial number of hdd/ssd of your computer.

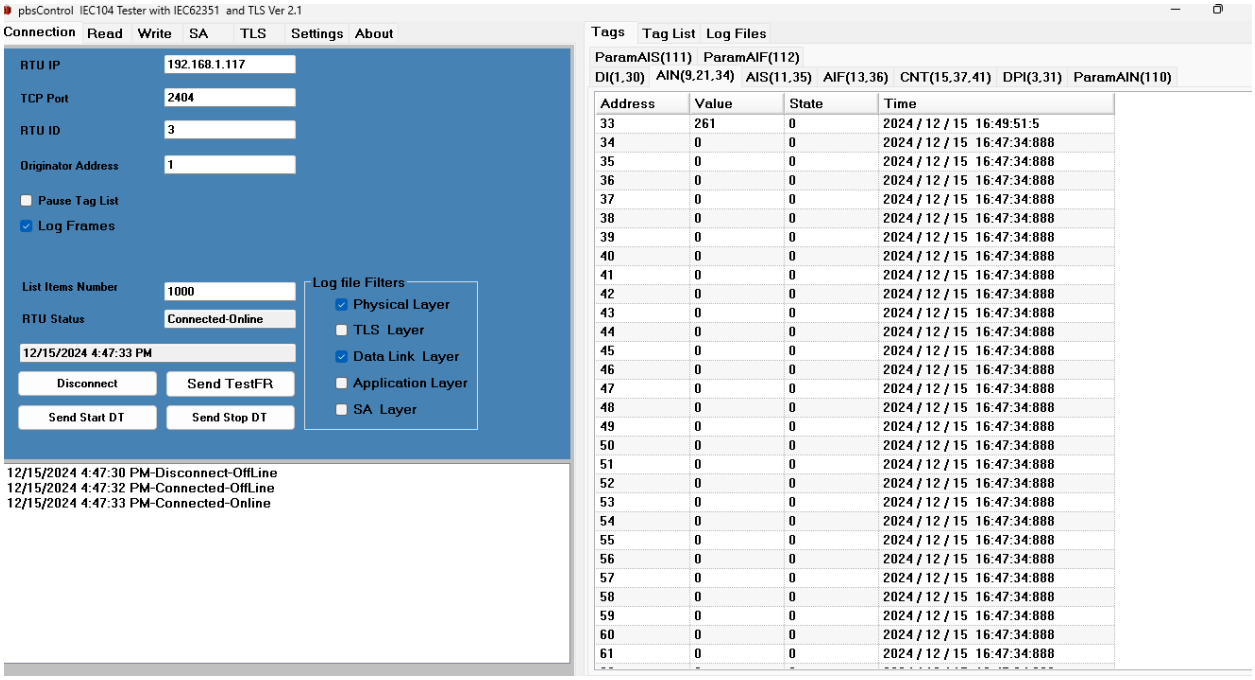
The other files are systematic files.

The Log folder is used to store IEC104 frames in text format.

The Cert folder is used to store TLS certificates.

3. Operation

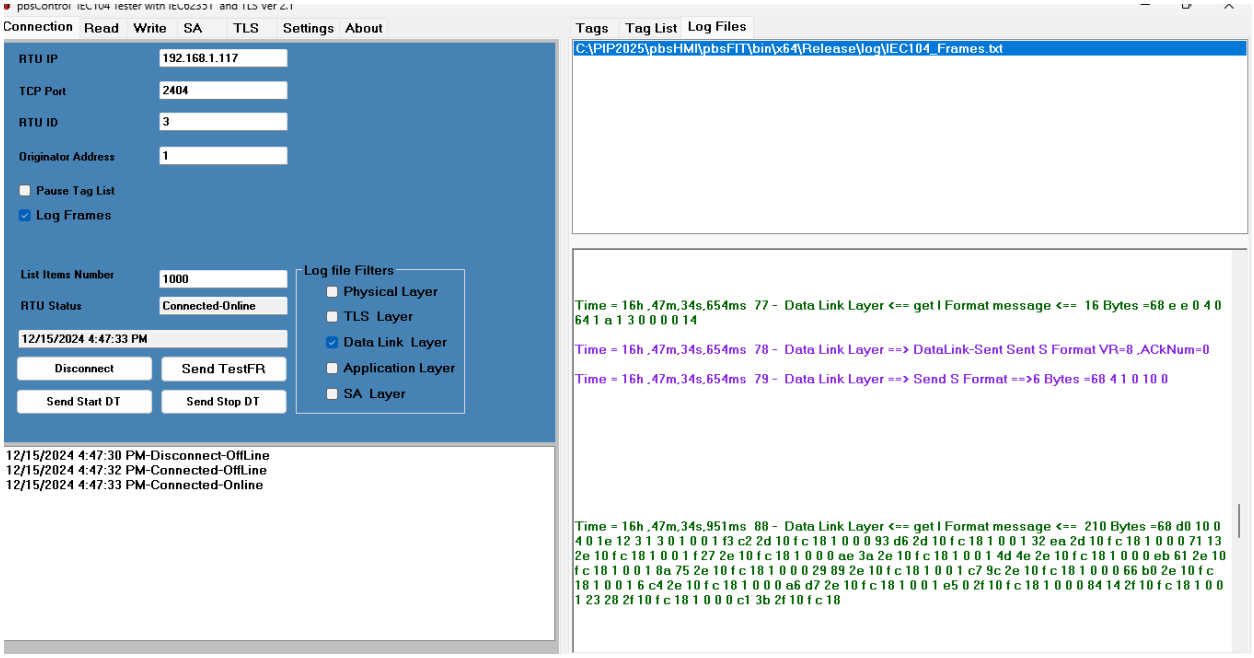
When you run pbsFIT.exe, you can see the following image:



In the Connection tab, you can set the RTU IP, TCP port, RTU ID, and originator address.

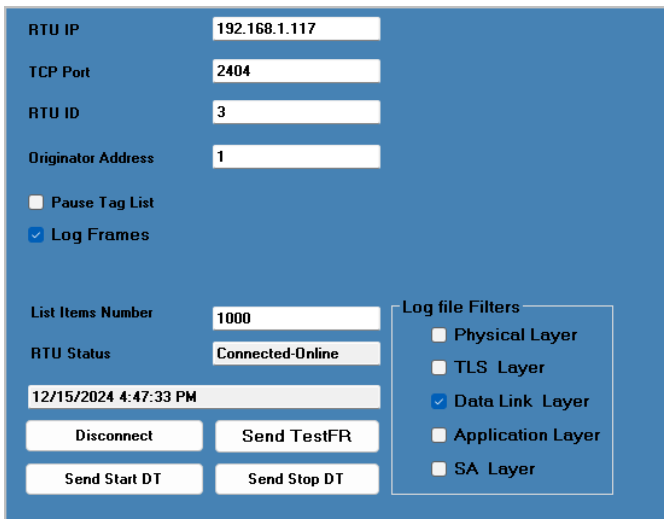
By clicking the Connect/Disconnect button, the tester connects/disconnects to/from the RTU.

To check the frames, you can enable the frame logging function from the Connection tab, or you can enable or disable frame capture by right-clicking on each tab.

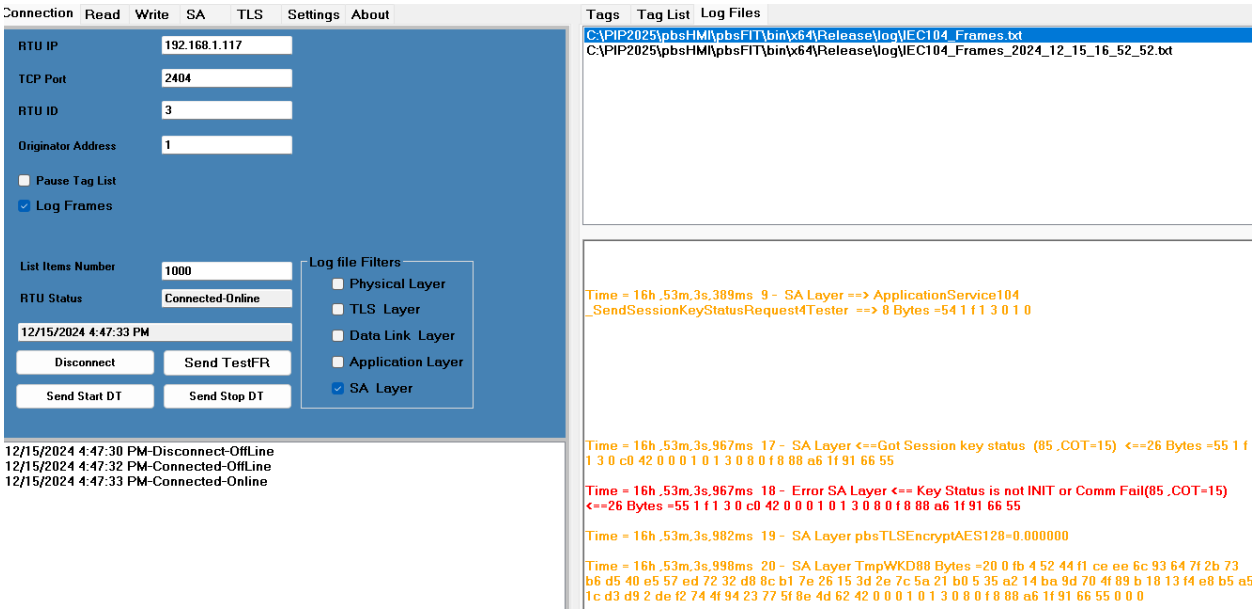


When the RTU is connected to the tester and the log frames are enabled, in the Log files tab, you can see that the logged text files are sorted by time. Every 200 frames are saved in one file. You can view the frames by clicking on each file.

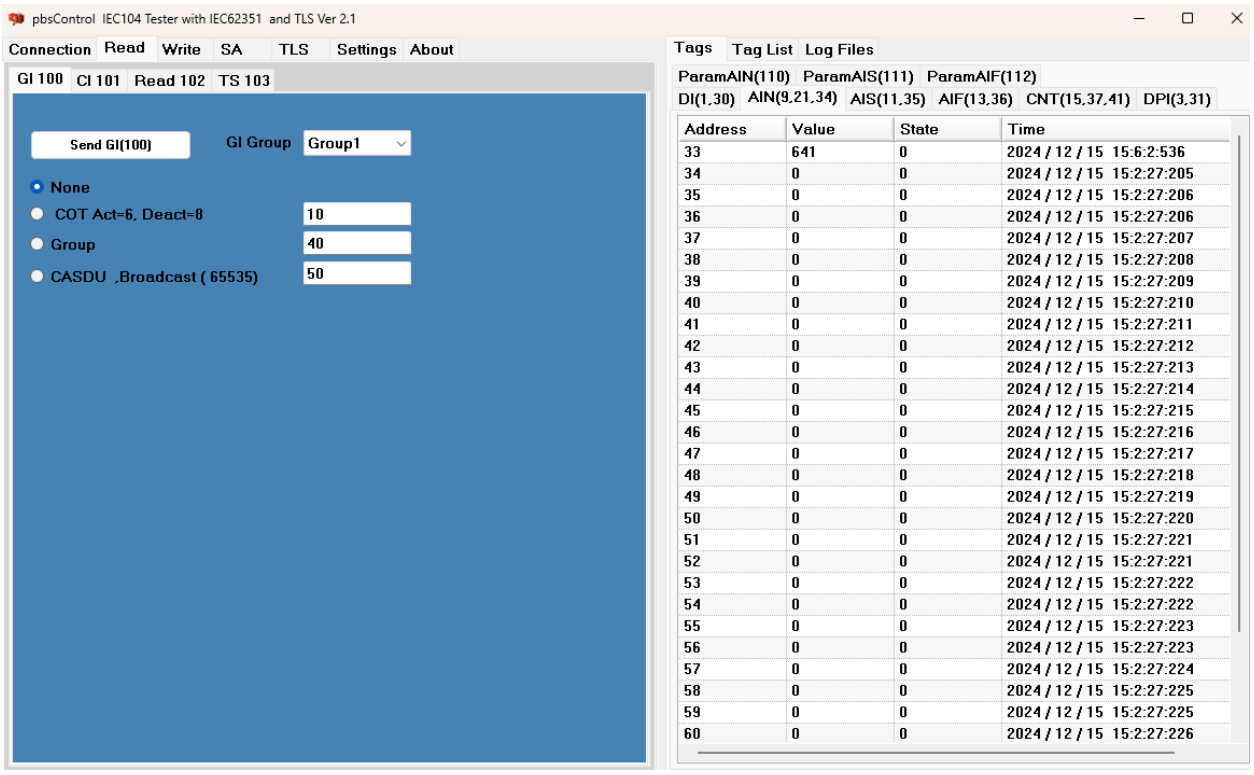
You can view frames from different layers by using the Physical, TLS ,Data Link, Application, and SA layers check boxes .



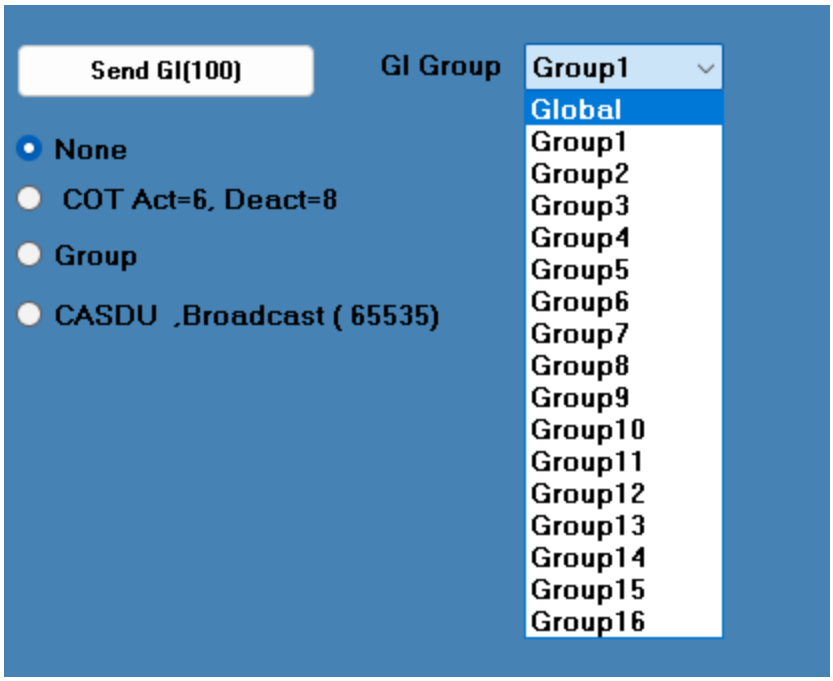
If you want to see only the SA layer frame, uncheck the others and check only the SA layer. And click on log file to update the display of frames.



In the Read tab you can see the following functions:



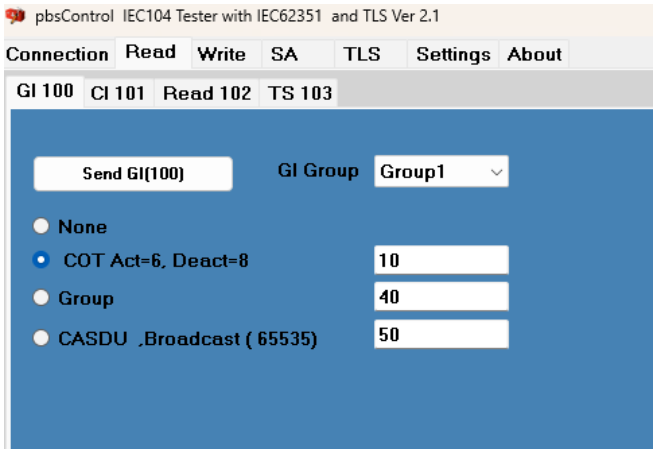
For GI , type ID 100 , you can select different Groups .



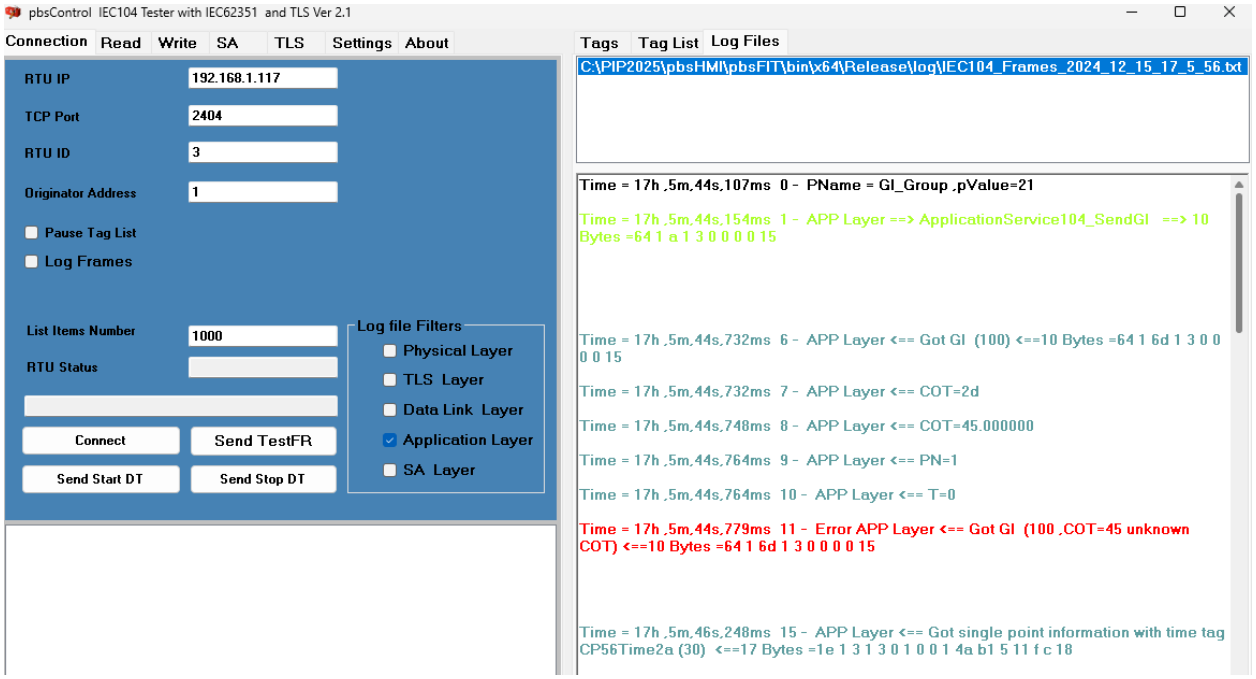
When the tester connects to the RTU, it will automatically send the GI to the RTU. But you can send the GI at any time.

To send the wrong command to the RTU, you can use sending with the wrong COT, group number, and RTU ID.

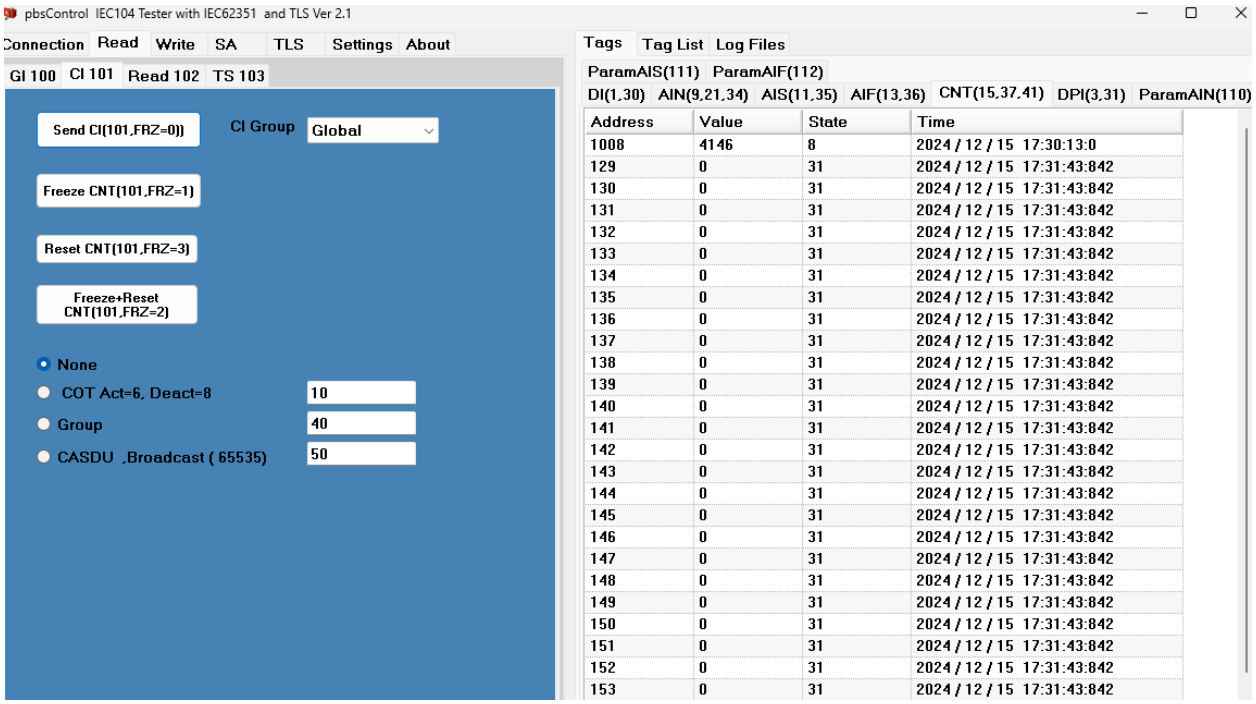
Suppose you want to send GI to RTU with COT 10, normal operation is 6 for activation and 8 for deactivation.



After clicking the SendGI button, the GI command with COT=10 is sent to the RTU.

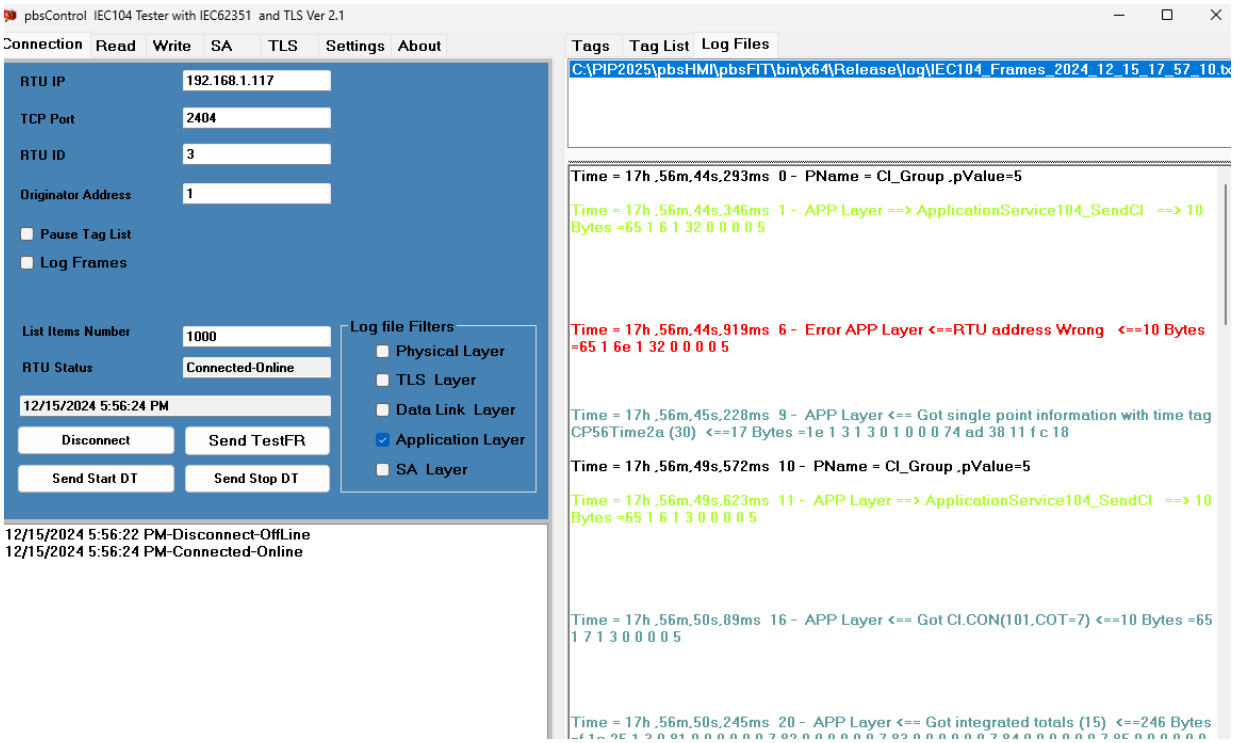


You can use CI Tab to read the counters:

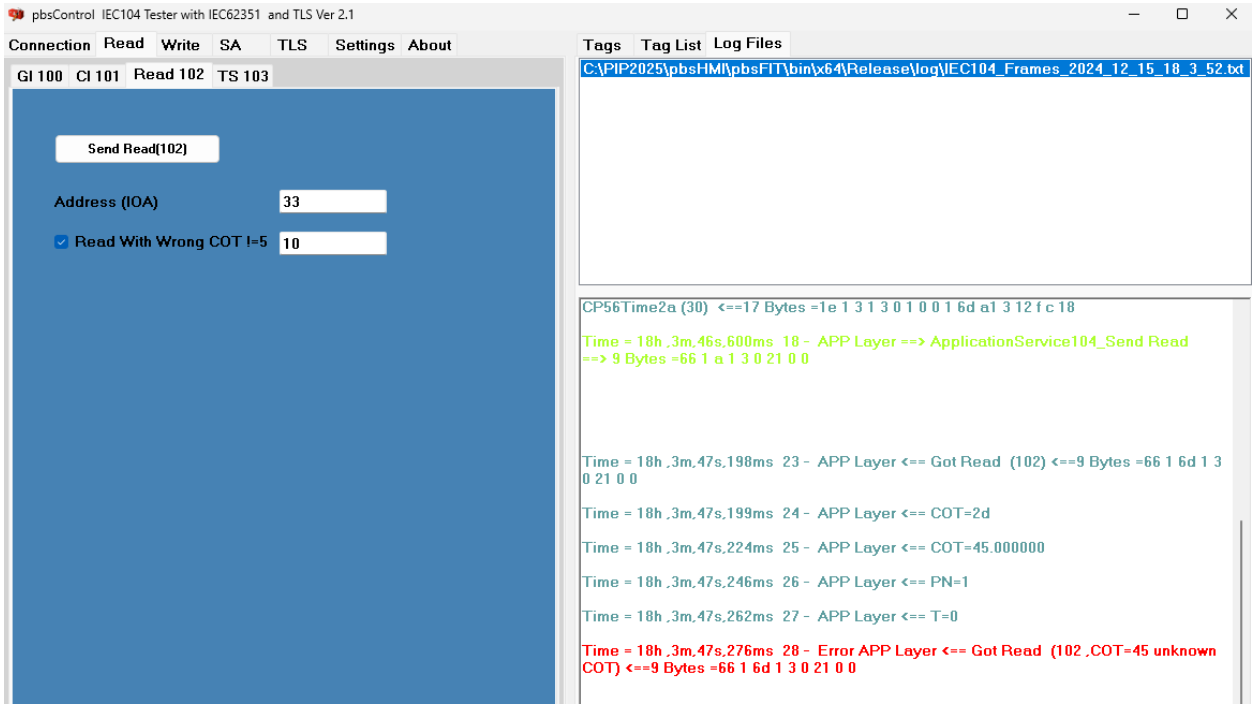


You can freeze the counters by clicking the Freeze CNT button or reset the counters by using Reset CNT button.

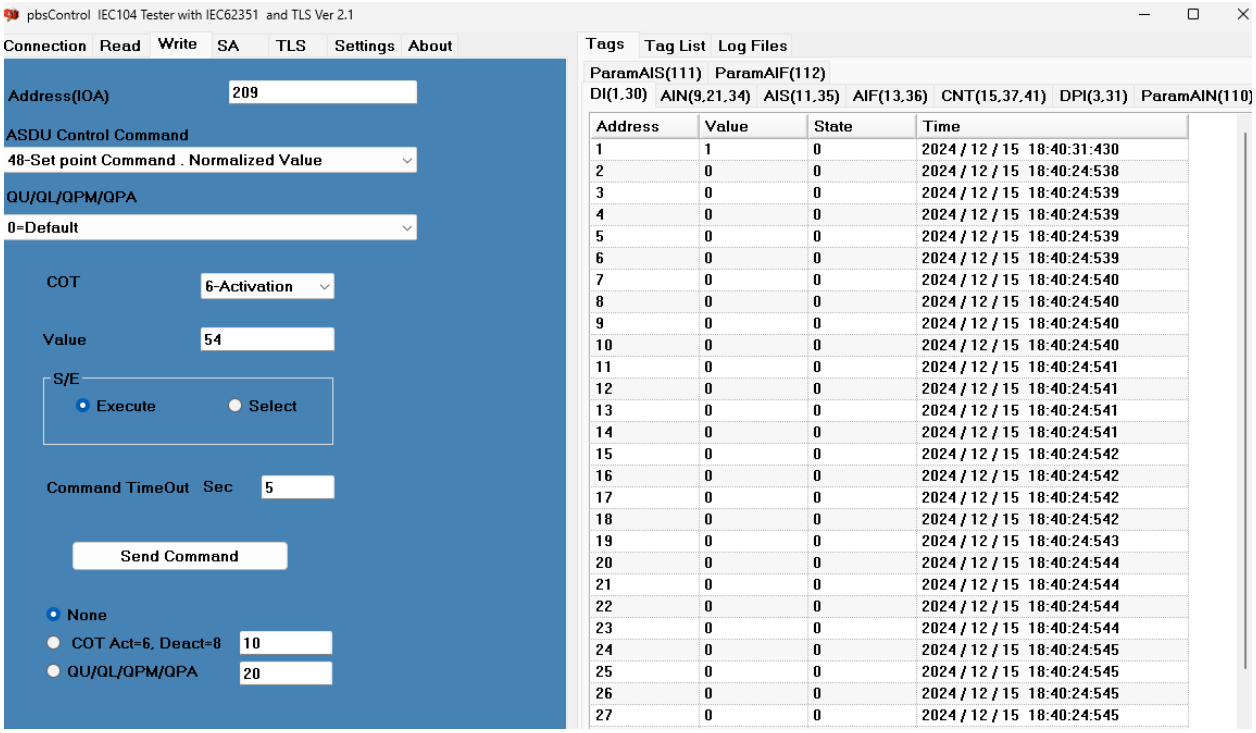
Suppose we want to check the RTU when it receives a CI with an incorrect RTU ID.



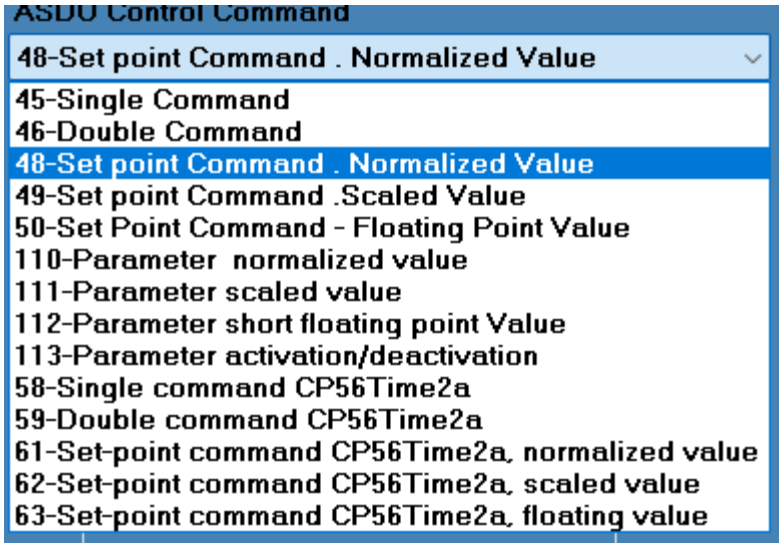
To read IEC104 tags by function 102, you can use the Read 102 tab. You can send function 102 with wrong COT and check the frames as shown below:



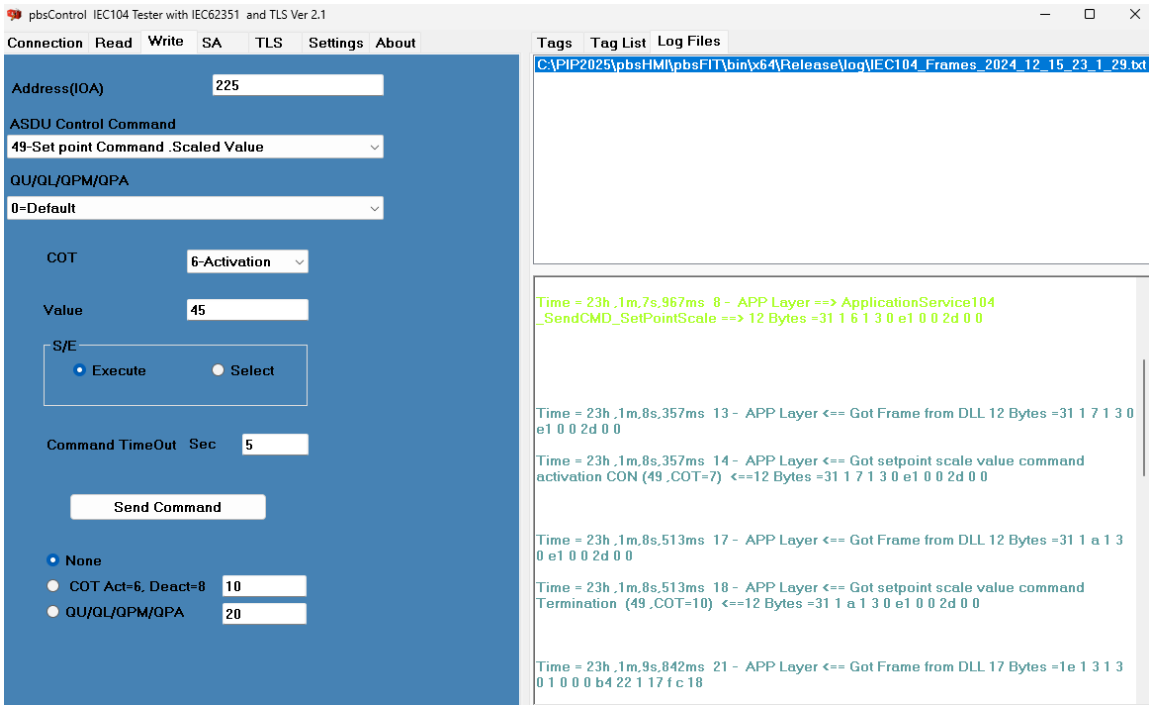
You can use the Write tab to write IEC104 tags:



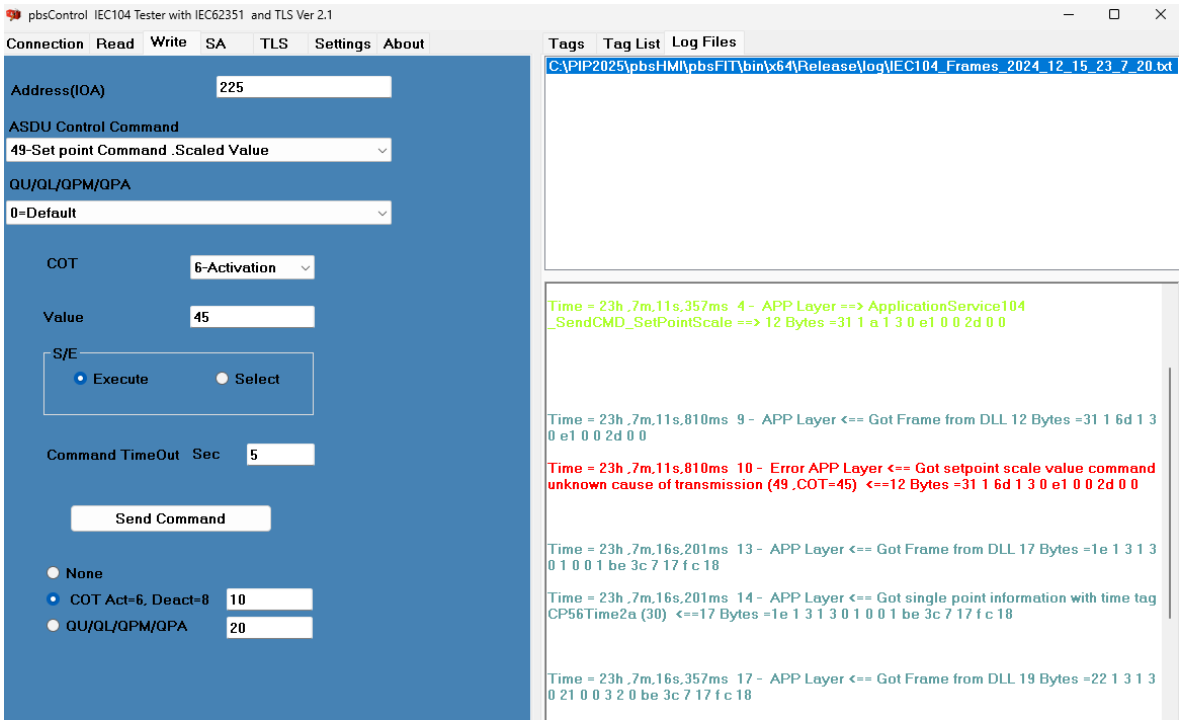
Following tags are supported:



For writing (Execute) value 45 to Set point command 49 (Scaled) with IOA 225, you can send following command:

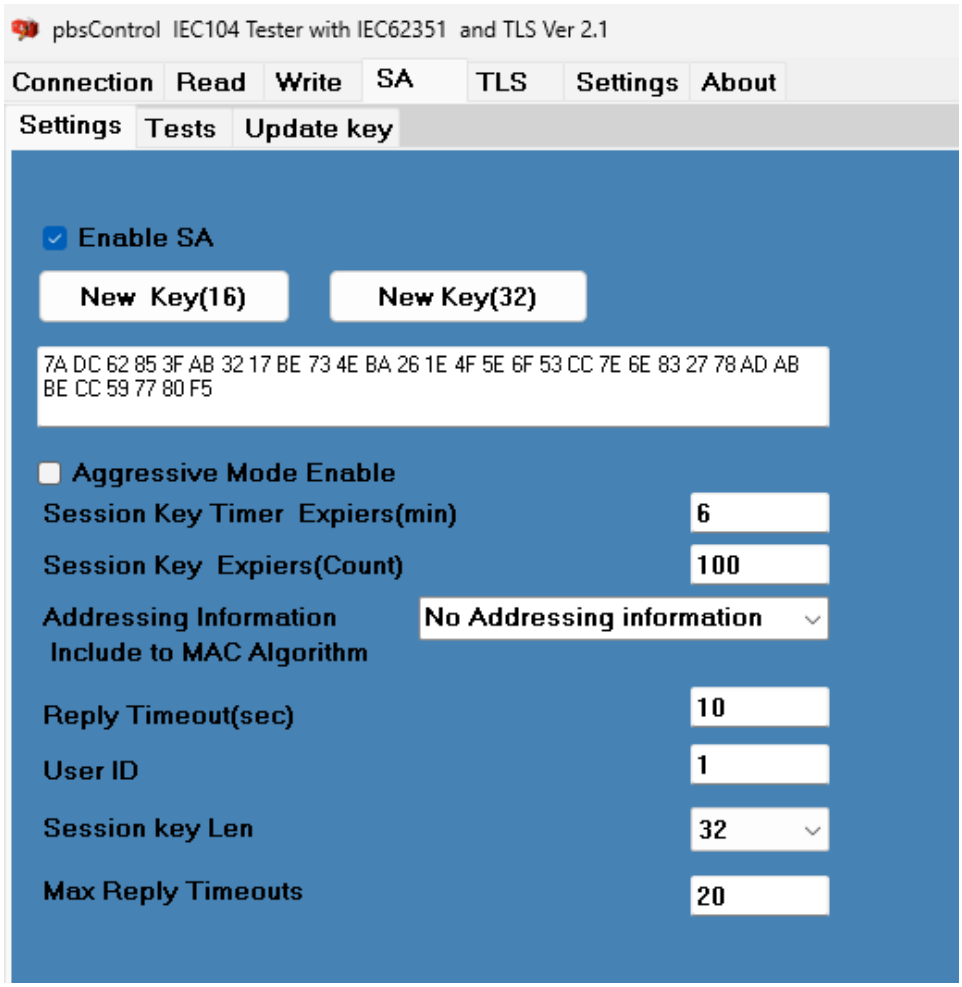


For sending the wrong command, for example sending the wrong COT, the tester will display error frames.



4. Enabling IEC62351

For enabling IEC62351, use SA Tab.



New key (16): generating a new 16 bytes Update key

New key (32): generating a new 32 bytes Update key

Use same update key in Tester and RTU.

Aggressive Mode Enable: If checked, SA uses aggressive mode to send critical commands from the tester to the RTU. In aggressive mode, the tester is sending Critical command and Authentication fields in the same frame. (Type ID 83)

When the tester connects to the RTU, the tester sends a new session key to the RTU. All critical commands must be authenticated with the session key.

Session Key expires (min):

Session Key expires (Count):

The session key between the tester and the RTU expires by number of critical commands or by time. When the session key expires, the tester sends a new session key to the RTU and resets the parameters.

Addressing information: always set to "no Addressing information" between tester and RTU.

Replay Timeout (seconds): There is a timeout period when the tester has sent a critical command to the RTU and is waiting for a response. For example, the tester is sending a key status request (84) and then waiting for a key status response (85).

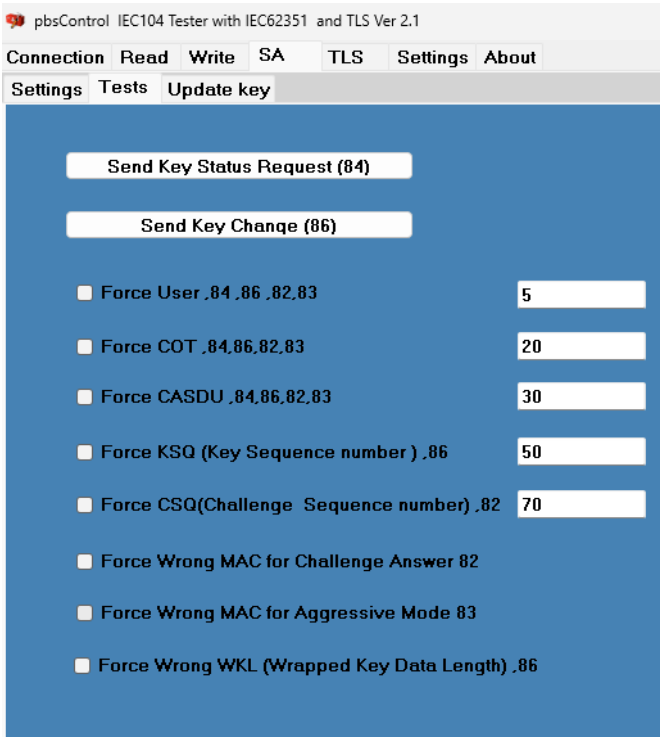
User ID: User ID defined in the RTU. pbsFIT does not support multiple users simultaneously for one RTU.

Session key Len: Length of dynamic session key (16 or 32 bytes)

Maximum Replay Time: If the number of timeouts between the tester and the RTU exceeds this value, the tester reinitializes the SA layer for the RTU and sends a new session key to the RTU.

IEC62351 tests

You can use the Test tab to test the RTU with some error conditions.



All Write commands are affected by this test condition. Suppose if you write a scaled value and Force User is checked, the tester sends a Write command (type 49) with the user on the test page, for example User ID = 5.

Note: For the tester to function properly, you must uncheck all test conditions and send a command to the RTU to disable the test conditions in the tester.

Force User: Suppose in the RTU, user with ID=1 is setup and you want to send key Status with other user ID. This test is working for SA type 84(Session key Request), 86(Change Session key), 82(Authentication reply), 83 (Aggressive Mode).

Force COT: Suppose you want to send key Status with wrong COT. This test is working for SA type 84(Session key Request), 86(Change Session key), 82(Authentication reply), 83 (Aggressive Mode).

Force CASDU: Suppose you want to send key Status with wrong RTU ID. This test is working for SA type 84(Session key Request), 86(Change Session key), 82(Authentication reply), 83 (Aggressive Mode).

Force KSQ: Suppose you want to send key Change with wrong Key Sequence Number. This test is working for SA type 86(Change Session key).

Force CSQ: Suppose you want to send a command with wrong Challenge Sequence Number. This test is working for SA type 82(Authentication reply).

Force Wrong MAC: Suppose you want to send a command with wrong MAC to RTU. This test is working for SA type 82(Authentication reply).

Force Wrong MAC for Aggressive Mode: Suppose you want to send a command with wrong MAC to RTU. This test is working for SA type 83 (Aggressive Mode).

Force WKL: Suppose you want to send key Change with wrong Wrapped Data Length. This test is working for SA type 86(Change Session key).

The resilience test case for session key management in the RTU device (EC104 Slave) is as follows:

7.5.3.3 Resiliency test cases

Table 28 – Session key maintenance: Controlled station resiliency tests

No.	Test	Action	Reference	Required
7.5.3.3.1	Reception of a S_KR_NA_1 for a NOT valid USR (not created in the Controlled Station)	Discard the ASDU received. Increment the Discarded Message Statistic.	IEC TS 62351-5:2013, 7.3.3.5 Table 30	M
7.5.3.3.2	Reception of a S_KC_NA_1 with a NOT valid WKD.	Discard the ASDU received. Increment the Discarded Message Statistic. Increment the Authentication Failures Statistic. Set the Session Key status to AUTH_FAIL Send S_KS_NA_1 with the new key status.	IEC TS 62351-5:2013, 7.3.3.5 Table 30	M
7.5.3.3.3	Reception of a S_KC_NA_1 with USR not matching to that in the last S_KS_NA_1 sent.	Discard the ASDU received. Increment Discarded Message Statistic.		M

S_KR_NA_1 is the key status request command sent by the tester to the RTU (type identifier 84).

S_KC_NA_1 is the key change command sent by the tester to the RTU (type identifier 86).

S_KS_NA_1 is the key status response sent by the RTU to the tester (type identifier 85).

So when the tester sends a key status request or Key change Command to the RTU and the user is not defined in the RTU, the RTU should reject the request and only increment the Discarded message statistics in the RTU.

If you are using pbsSoftLogic for RTU programming (as IEC04 Slave), you have the following SA tags:

PhysicalLayer | IEC101 | IEC | SA Layer | TLS | Others | Tags

Name	Type	Class	Init	Address
SYS.MasterIsOnline	SYS-System Diagnostic	0	0	1
SYS.GIStatus	SYS-System Diagnostic	0	0	2
SA_UnexpectedMessagesNum	SYS-System Diagnostic	0	3	3
SA_AuthorizationFailuresNum	SYS-System Diagnostic	0	5	4
SA_AuthenticationFailuresNum	SYS-System Diagnostic	0	5	5
SA_ReplyTimeoutsNum	SYS-System Diagnostic	0	3	6
SA_RekeysDueToAuthenticationFailureNum	SYS-System Diagnostic	0	3	7
SA_TotalMessagesSentNum	SYS-System Diagnostic	0	100	8
SA_TotalMessagesReceivedNum	SYS-System Diagnostic	0	100	9
SA_CriticalMessagesSentNum	SYS-System Diagnostic	0	100	10
SA_CriticalMessagesReceivedNum	SYS-System Diagnostic	0	100	11
SA_DiscardedMessagesNum	SYS-System Diagnostic	0	10	12
SA_ErrorMessagesSentNum	SYS-System Diagnostic	0	10	13
SA_ErrorMessagesReceivedNum	SYS-System Diagnostic	0	10	14
SA_SuccessfulAuthenticationsNum	SYS-System Diagnostic	0	100	15
SA_SessionKeyChangesNum	SYS-System Diagnostic	0	10	16
SA_FailedSessionKeyChangesNum	SYS-System Diagnostic	0	5	17
SA_UpdateKeyChangesNum	SYS-System Diagnostic	0	1	18
SA_FailedUpdateKeyChangesNum	SYS-System Diagnostic	0	1	19
SYS.CounterResetedByMaster	SYS-System Diagnostic	0	0	20
SYS.EnableFrameLogging	SYS-System Diagnostic	0	0	21

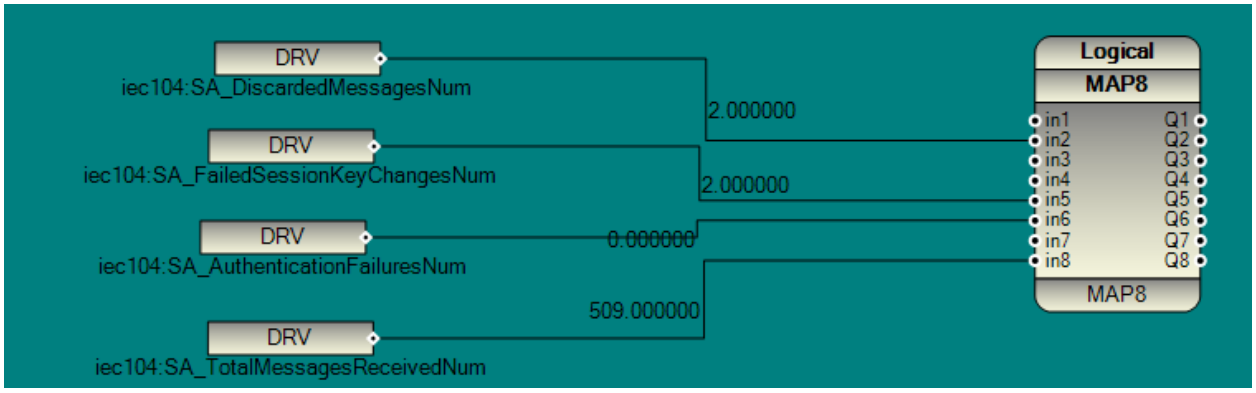
The above variables starting with "SA_" are transmitted to the SCADA centre as SA Counter, type 41. The Init value is the maximum value for spontaneous reporting to the SCADA centre.

For example, for SA_FailedSessionKeyChangeNum, the initial value is 5, so after 5 failed session key changes, its value is reported to the SCADA centre.

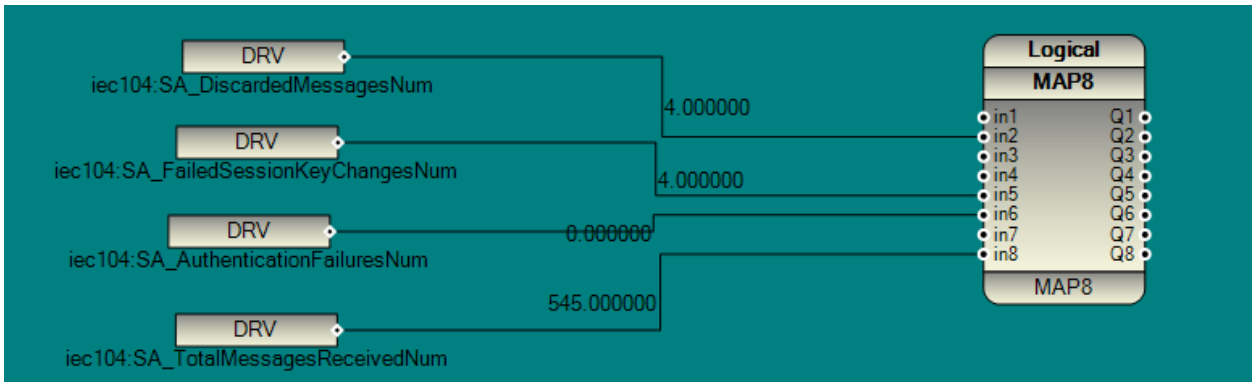
The address field is the same as the IEC104 tag address, but you can add a base value to all SA counters, for example if you consider the base address to be 1000, the IEC104 tag address for FailedSessionKeyChangeNum is 1017.

In the following scenario, we send the key change command 5 times by the tester to the RTU programmed by pbsSoftLogic as Slave RTU and check the value of FailedSessionKeyChangeNum in the RTU and the messages sent between the tester and the RTU.

1. Values of some SA counters before starting the test in the RTU:



2. We enable frame Logging for Tester and send a key change with the correct WKL and a command with the wrong WKL. We repeat the test twice. The SA counter values change as shown in the following image:

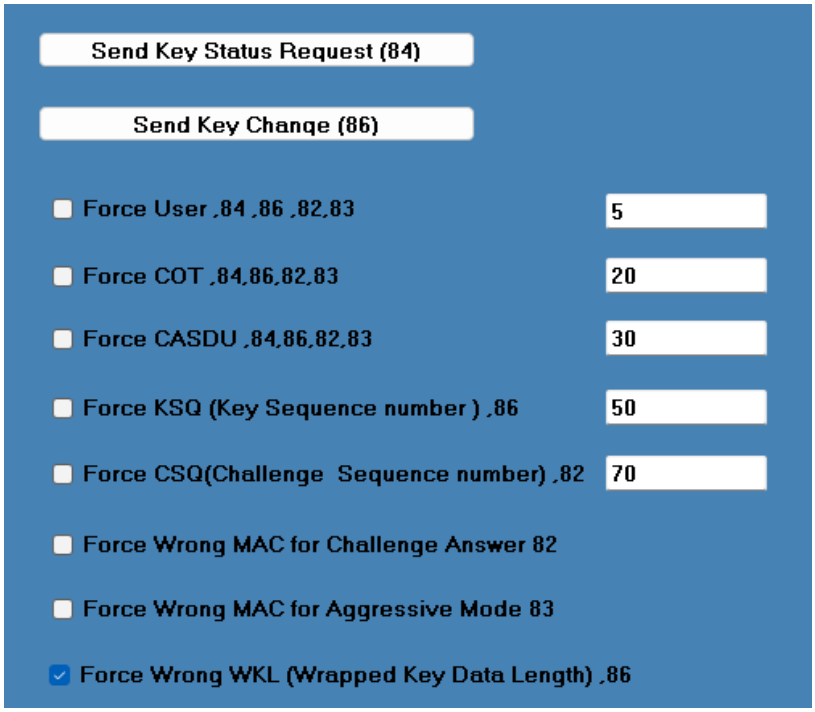


The number of DiscardedMessageNum and FailedSessionKeyChangeNum

increased by two units because we ran the test twice.

The frames between the tester and the RTU are as follows:

```
Time = 11h ,34m,11s,433ms  13 -  SA Layer ==> ApplicationService104_SendSessionKeyChange4Tester  ==> 127 Bytes =56 1 f 1 3 0 c0 58 0 0 0 1 0
70 0 a0 ed c7 f4 1f c2 1b d8 9b 7d 32 ee 60 e8 b3 90 4a 86 76 bf c2 fa a3 50 76 d4 4 f0 88 fd ea c1 52 a0 ca b9 f8 d0 8a 9d 7c f1 d6 89 23 a0 40 b3 13
b5 19 6d 7c 11 1b 44 f3 68 75 ee 23 e2 a6 e8 e9 d1 b3 11 8a b8 5a ac b9 f6 f1 d5 7b b0 b4 97 28 ca f 7c ac 8b ac 66 ee 6e fd fc 15 6f 2d 48 39 cf 8a bb
83 91 38 d0 a9 61 51 9e f2 4d 14 92



Time = 11h ,34m,11s,911ms  19 -  SA Layer <==Got Session key status  (85 ,COT=15)  <==42 Bytes =55 1 f 1 3 0 c0 59 0 0 0 1 0 1 1 4 8 0 dd 93 29 47 52
a2 13 7 29 73 2 b0 b4 68 dd 90 72 64 ea 9f 2 af cd d1

Time = 11h ,34m,11s,922ms  20 -  SA Layer <==key Status Answer is OK  (85 ,COT=15)  <==42 Bytes =55 1 f 1 3 0 c0 59 0 0 0 1 0 1 1 4 8 0 dd 93 29 47
52 a2 13 7 29 73 2 b0 b4 68 dd 90 72 64 ea 9f 2 af cd d1



Time = 11h ,34m,17s,312ms  25 -  SA Layer pbsTLSEncryptAES128=0.000000

Time = 11h ,34m,17s,312ms  26 -  SA Layer ==> ApplicationService104_SendSessionKeyChange4Tester  ==> 127 Bytes =56 1 f 1 3 0 c0 59 0 0 0 1 0
75 0 0 91 34 99 9f 52 c3 2c 82 90 32 e 8d 6e cb e9 94 e 1c e7 a6 ce 37 1f ac dd 57 43 ac 9a d2 d9 fb d7 6f bc 3a b5 a 54 ea 12 d 3 2f a8 2e cc bd ab d2
bd 31 c3 6c b8 a0 22 17 40 21 2f 4c 7d b1 16 27 e5 e6 b1 b8 44 6f 38 c1 97 5d b0 89 8c 1b 85 19 36 d9 33 36 d2 91 ff 8b 42 13 4a 78 1d f3 32 d5 62 b1
d6 ac 8d c7 14 1 86 fb ee 1a fe



Time = 11h ,34m,17s,847ms  32 -  SA Layer <==Got Session key status  (85 ,COT=15)  <==26 Bytes =55 1 f 1 3 0 c0 5a 0 0 0 1 0 1 4 0 8 0 be 63 2c f8 fd
26 2c dd

Time = 11h ,34m,17s,861ms  33 -  Error SA Layer <==Authentication Failed at the Slave  (85 ,COT=15)  <==26 Bytes =55 1 f 1 3 0 c0 5a 0 0 0 1 0 1 4 0 8
0 be 63 2c f8 fd 26 2c dd
```

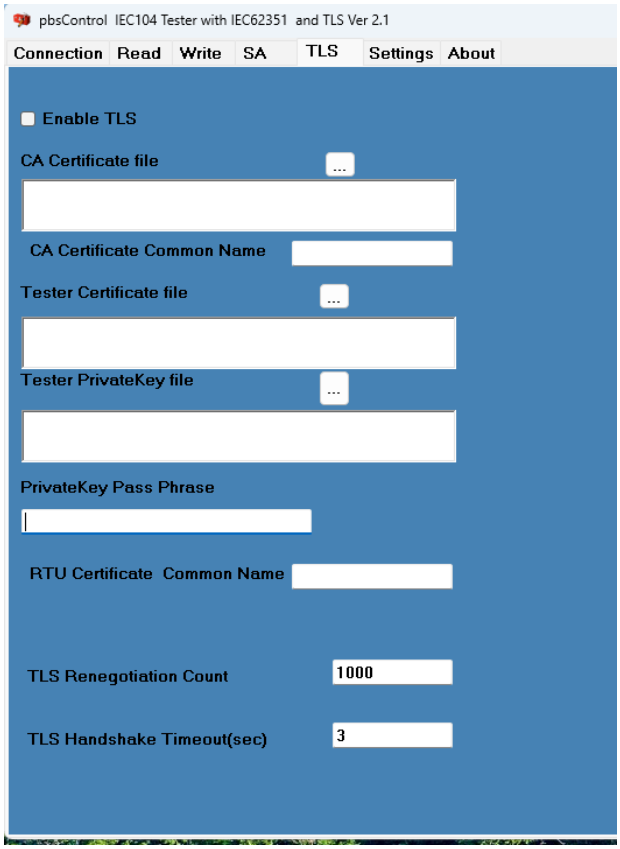


Every time we send the Wrong Session Key change command to the RTU, the RTU increments the AuthenticationFailuresNum counter. When it reaches the maximum value, the SA_RekeysDueToAuthenticationFailureNum counter is incremented.

If RekeysDueToAuthenticationFailureNum exceeds its maximum value, the RTU will disconnect and wait for a new connection from the Master and a New Session Key.

5. Enabling TLS

IEC62351 is responsible for authentication and TLS handles frame encryption. From the TLS tab you can set up TLS communication with the RTU.



The easiest way to create certificate files for testing purposes is to use the XCA tool.

Create certificates for the CA, RTU, and tester with the help of the XCA tool.

If the CA common name is empty, the tester will not check the CA common name during the handshake.

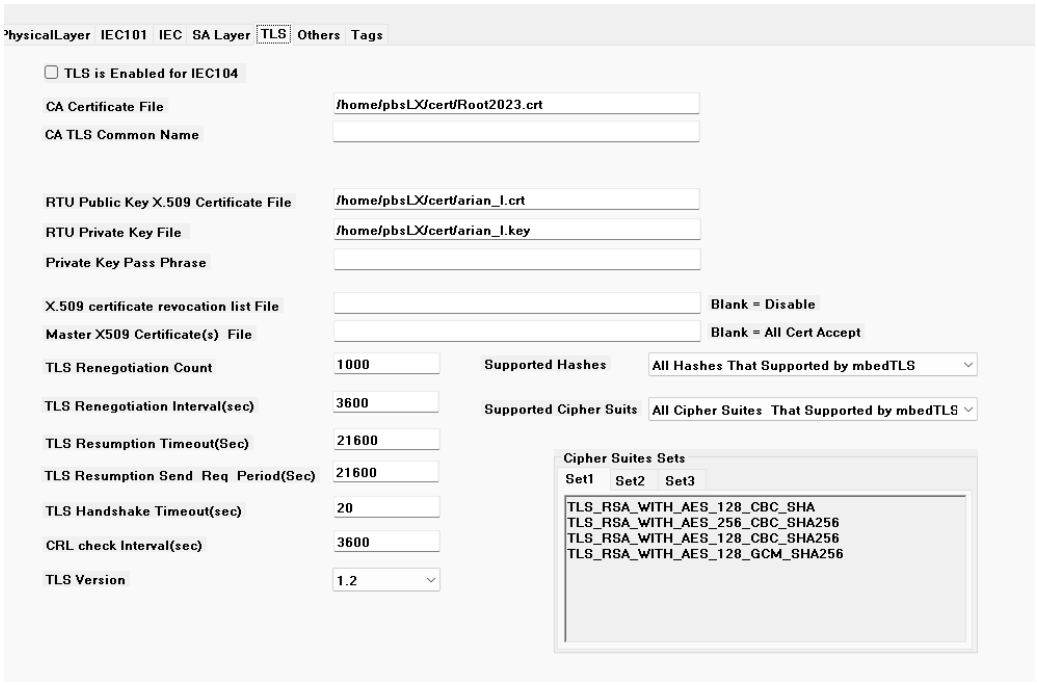
For the tester, you must have the certificate file and the private key file with the passphrase.

If the RTU certificate common name is empty, the tester will not check it during the handshake.

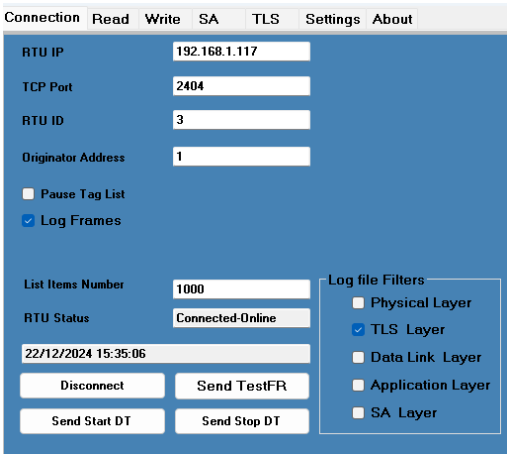
You can set the number of TLS renegotiations and handshake timeout on this page.

For TLS connection, appropriate settings must be made in the RTU.

If you are using pbsSoftLogic for RTU programming, you can set up TLS for the RTU with the following page.

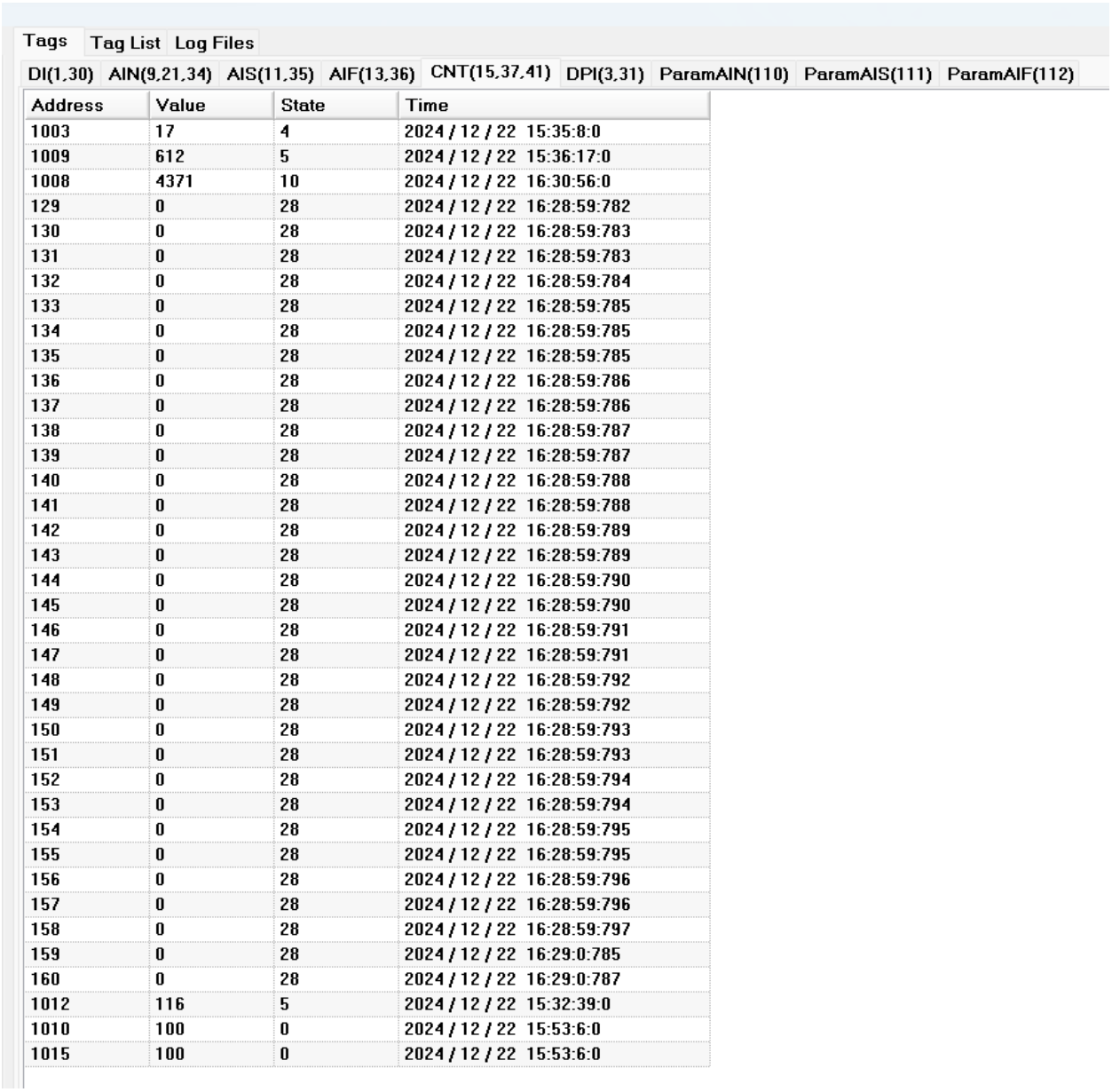


For a detailed description of how to set up TLS for the IEC 104 slave driver, please refer to the pbsSoftLogic user guide. To filter TLS frames, you can check TLS Layer on the Connection page.



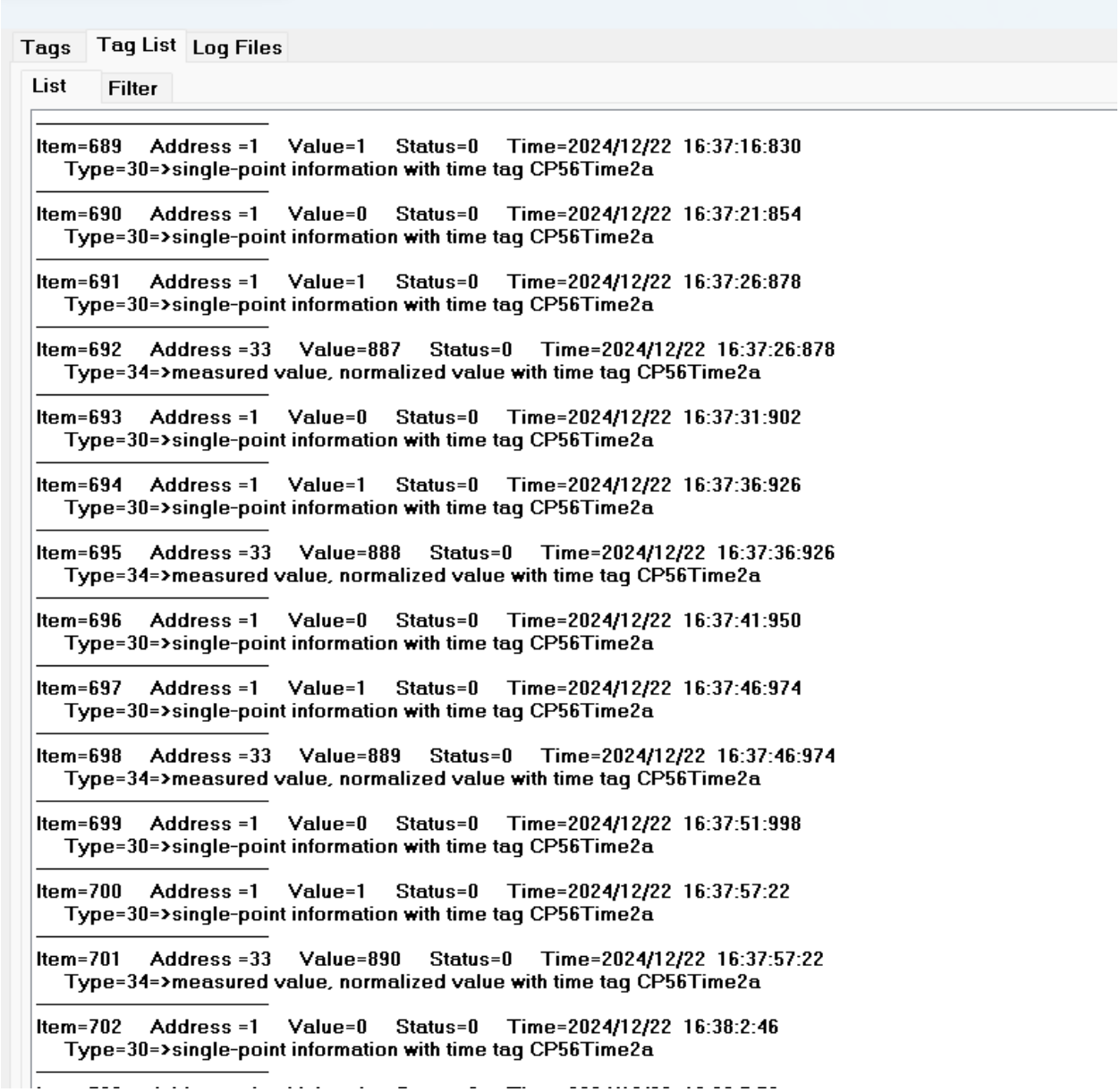
5. Tags monitoring

You can see the tag value, status, address, and time from the Tags tab.



Address	Value	State	Time
1003	17	4	2024 / 12 / 22 15:35:8:0
1009	612	5	2024 / 12 / 22 15:36:17:0
1008	4371	10	2024 / 12 / 22 16:30:56:0
129	0	28	2024 / 12 / 22 16:28:59:782
130	0	28	2024 / 12 / 22 16:28:59:783
131	0	28	2024 / 12 / 22 16:28:59:783
132	0	28	2024 / 12 / 22 16:28:59:784
133	0	28	2024 / 12 / 22 16:28:59:785
134	0	28	2024 / 12 / 22 16:28:59:785
135	0	28	2024 / 12 / 22 16:28:59:785
136	0	28	2024 / 12 / 22 16:28:59:786
137	0	28	2024 / 12 / 22 16:28:59:786
138	0	28	2024 / 12 / 22 16:28:59:787
139	0	28	2024 / 12 / 22 16:28:59:787
140	0	28	2024 / 12 / 22 16:28:59:788
141	0	28	2024 / 12 / 22 16:28:59:788
142	0	28	2024 / 12 / 22 16:28:59:789
143	0	28	2024 / 12 / 22 16:28:59:789
144	0	28	2024 / 12 / 22 16:28:59:790
145	0	28	2024 / 12 / 22 16:28:59:790
146	0	28	2024 / 12 / 22 16:28:59:791
147	0	28	2024 / 12 / 22 16:28:59:791
148	0	28	2024 / 12 / 22 16:28:59:792
149	0	28	2024 / 12 / 22 16:28:59:792
150	0	28	2024 / 12 / 22 16:28:59:793
151	0	28	2024 / 12 / 22 16:28:59:793
152	0	28	2024 / 12 / 22 16:28:59:794
153	0	28	2024 / 12 / 22 16:28:59:794
154	0	28	2024 / 12 / 22 16:28:59:795
155	0	28	2024 / 12 / 22 16:28:59:795
156	0	28	2024 / 12 / 22 16:28:59:796
157	0	28	2024 / 12 / 22 16:28:59:796
158	0	28	2024 / 12 / 22 16:28:59:797
159	0	28	2024 / 12 / 22 16:29:0:785
160	0	28	2024 / 12 / 22 16:29:0:787
1012	116	5	2024 / 12 / 22 15:32:39:0
1010	100	0	2024 / 12 / 22 15:53:6:0
1015	100	0	2024 / 12 / 22 15:53:6:0

In the tag list, you can see the sequence of tags that arrive at the tester in chronological order.



If the tag that reaches the tester has a time stamp, the tester uses that time, otherwise it sticks the current time for the tag.