pbsSoftLogic IEC104 Slave Driver Configuration

Dec 2024

Ver. 1.0

Kamjoo Bayat

Kb@pbscontrol.com

1 – Introduction

This document describes how to configure the pbsSoftLogic IEC 870-5-104 slave driver.

The IEC104 Slave driver supports the IEC 62351 and TLS layers and this document explains all the configuration details for both layers.

IEC62351 is about authentication and TLS is about encryption of IEC104 frames.

The IEC62351 layer is named SA layer. SA stands for Security and Authentication of frames.

2 – Adding New IEC104 Slave Driver

Create a new project in pbsSoftLogic and name it IEC104Slave. Click on Project setting and select Controller type. Let's say we want to use BPI-6202 RTU from Banana PI Company. Set RTU IP and save the configuration.(Figure 1)

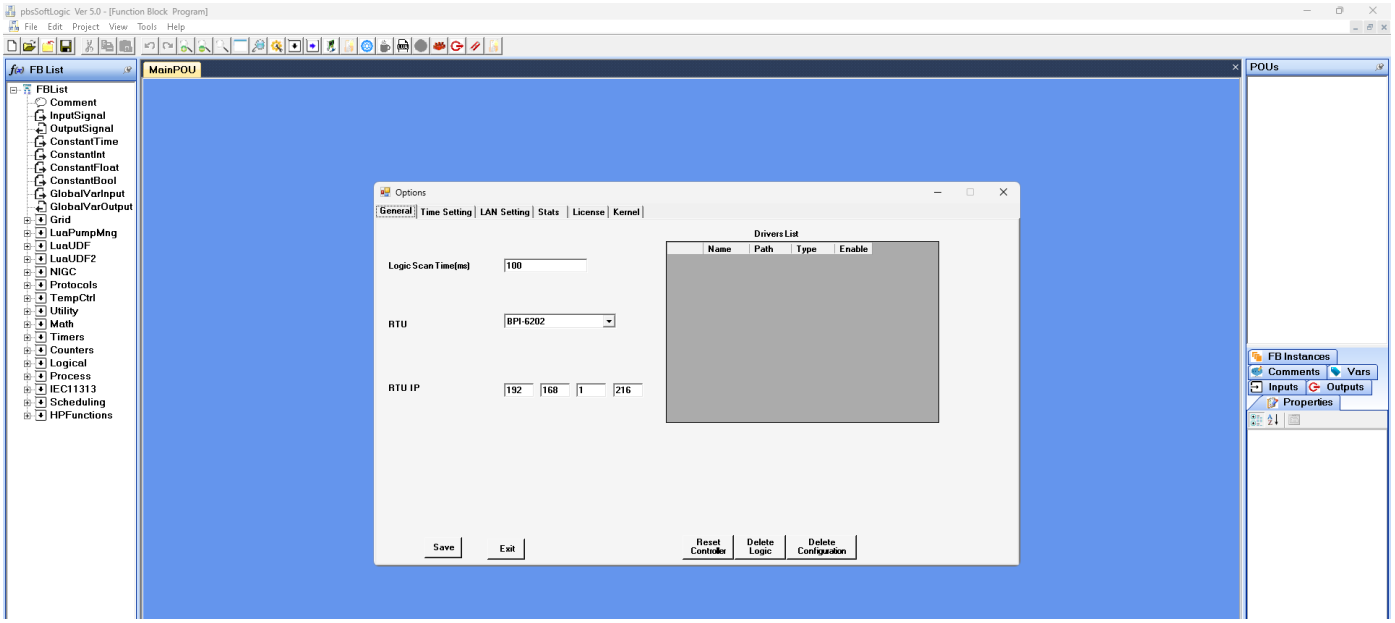


Figure 1

Right click on the list of drivers and add a new driver of type "IEC8705Slave" to the project. You can use any name for the driver but it must be unique in the project. Let's say we use "IEC104S" as the driver name. Keep the instance number at 1. The concept of Instance has been removed in pbsSoftLogic 2025. (Figure 2)

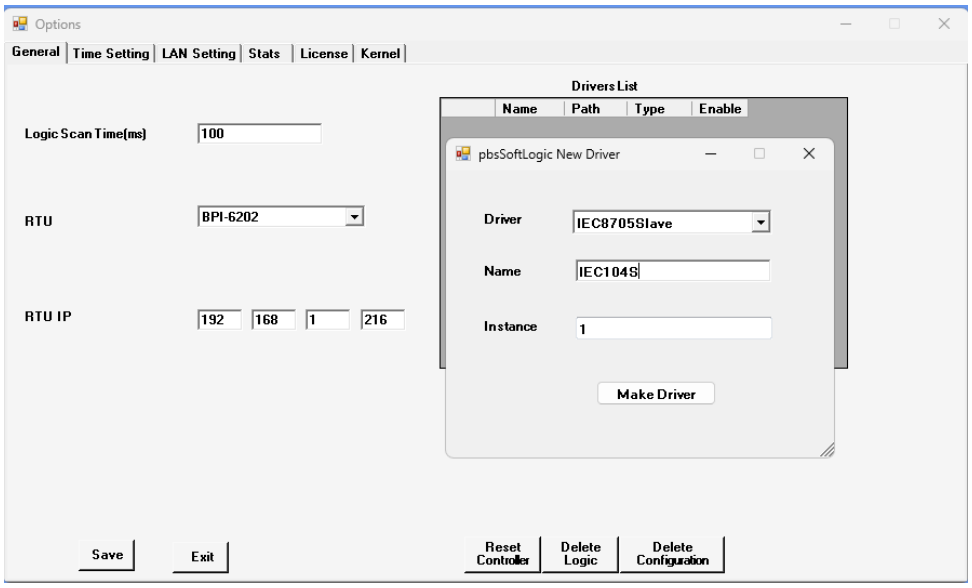


Figure 2

Click the “Make Driver” button, the initial settings and tags for the IEC104 Slave driver will be added to the project. (Figure 3)

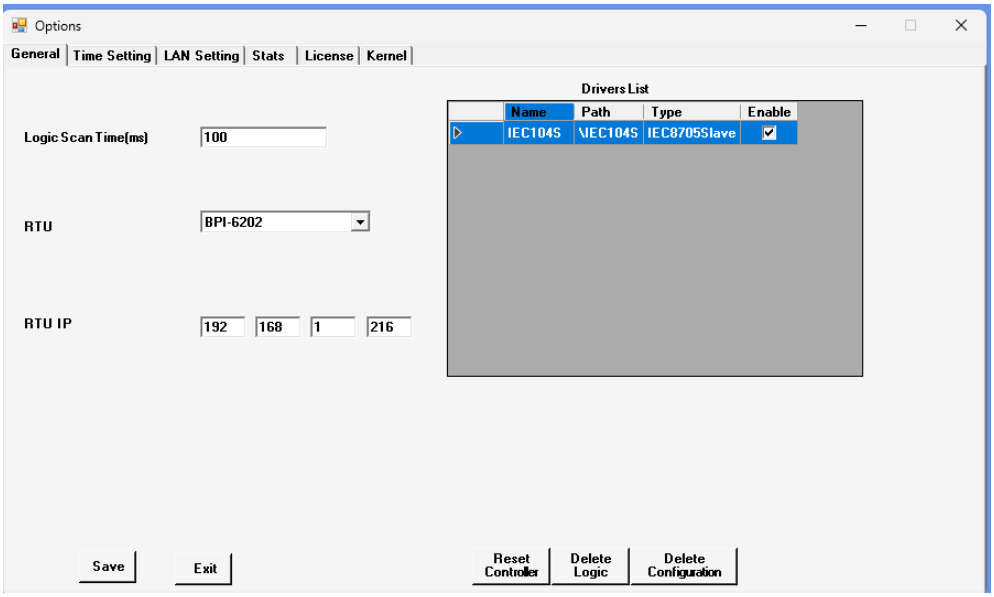


Figure 3

In the project folder you can see a folder for each driver. After adding "IEC104S" to the project, the project folder is as shown in Figure 4.

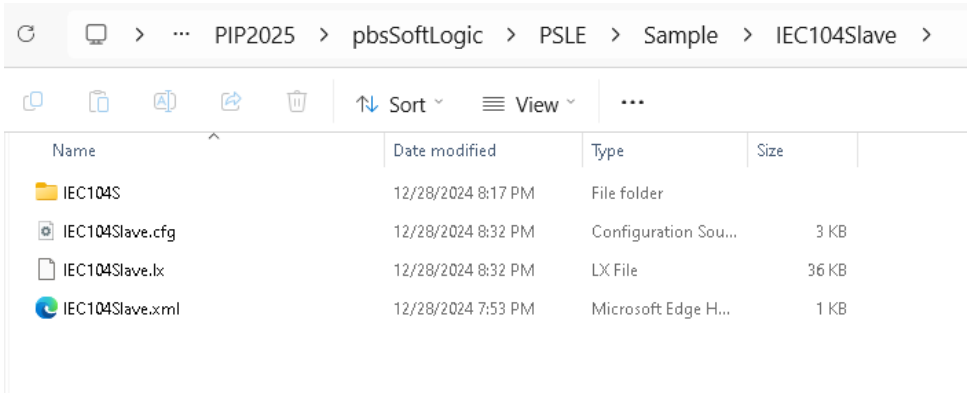


Figure 4

Two files have been created in the "IEC104S" folder. (Figure 5)

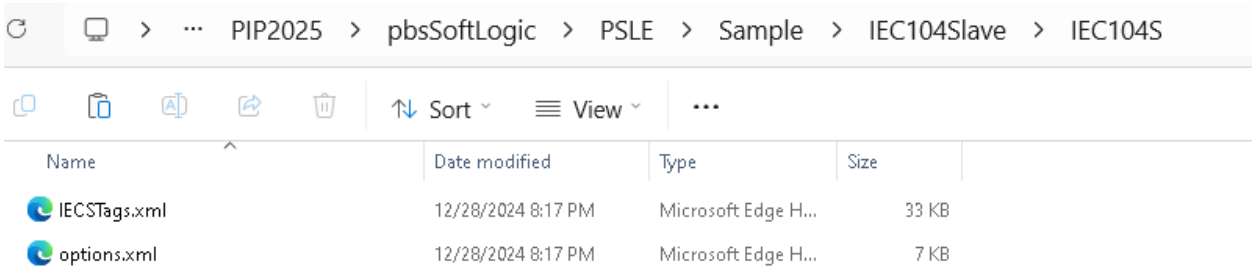


Figure 5

The default IEC104 tags are placed in the IECSTags.xml file. You can change the list of tags in this file. All other driver parameters are placed in the options.xml file.

By double-clicking on the "IEC104S" driver or by right-clicking on it and using the "Edit" menu, the IEC104 Slave configuration tool will be launched. (Figure 6)

The IEC 8705 Slave Editor has several tabs:

- Physical Layer: Setup the physical layer (Serial for IEC101 and TCP for IEC104)
- IEC101: Set up the IEC101 parameters. Not included in this document
- IEC: IEC104 parameters
- SA Layer: IEC62351 parameters
- TLS: Configure the TLS layer
- Others: Other options such as debug parameters.
- Tags: IEC8705 tags definition.

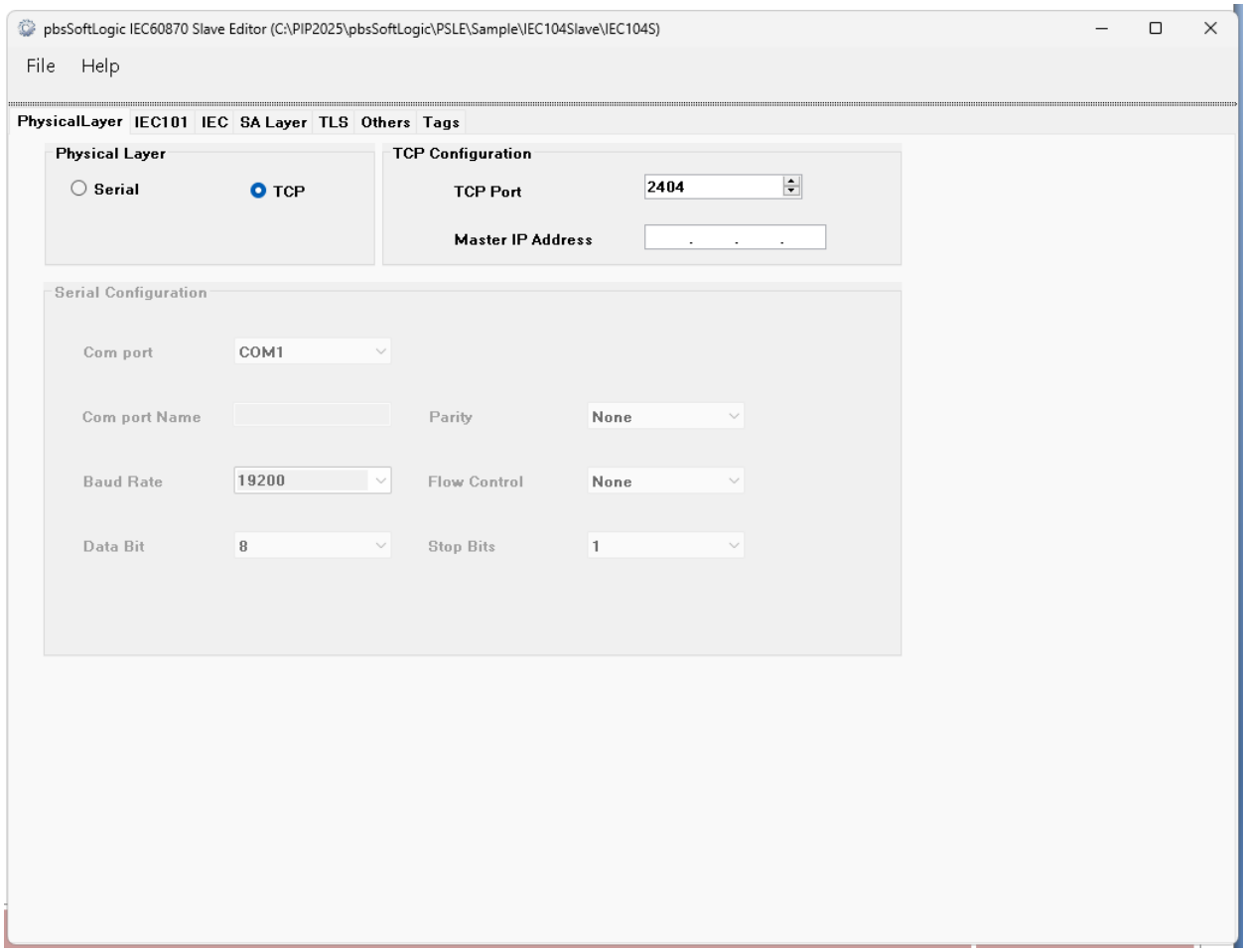


Figure 6

TCP Port: By default the port for IEC104 is 2404. If you add more IEC104 Slave drivers to the project, you must use a different port.

Master IP Address: If you want the RTU to connect only to a specific Master with a specific IP address, you need to set the Master IP here. If the Master IP address is blank, the RTU will connect to any Master and will not check the Master IP.

IEC104 Parameters (Figure 7)

Slave Address (CASDU): This is the common ASDU address. In IEC104 standard, two bytes are reserved for CASDU which is the RTU ID. 0xFFFF is considered as broadcast address.

Originator address: In the IEC104 standard, one byte is reserved for the Originator address. This address must be the same for both the master and slave.

K (T) Parameter: The transmitter stops the transmission at k unacknowledged I format APDUs.

W (R) Parameter: The receiver acknowledges at the latest after receiving w I format APDU

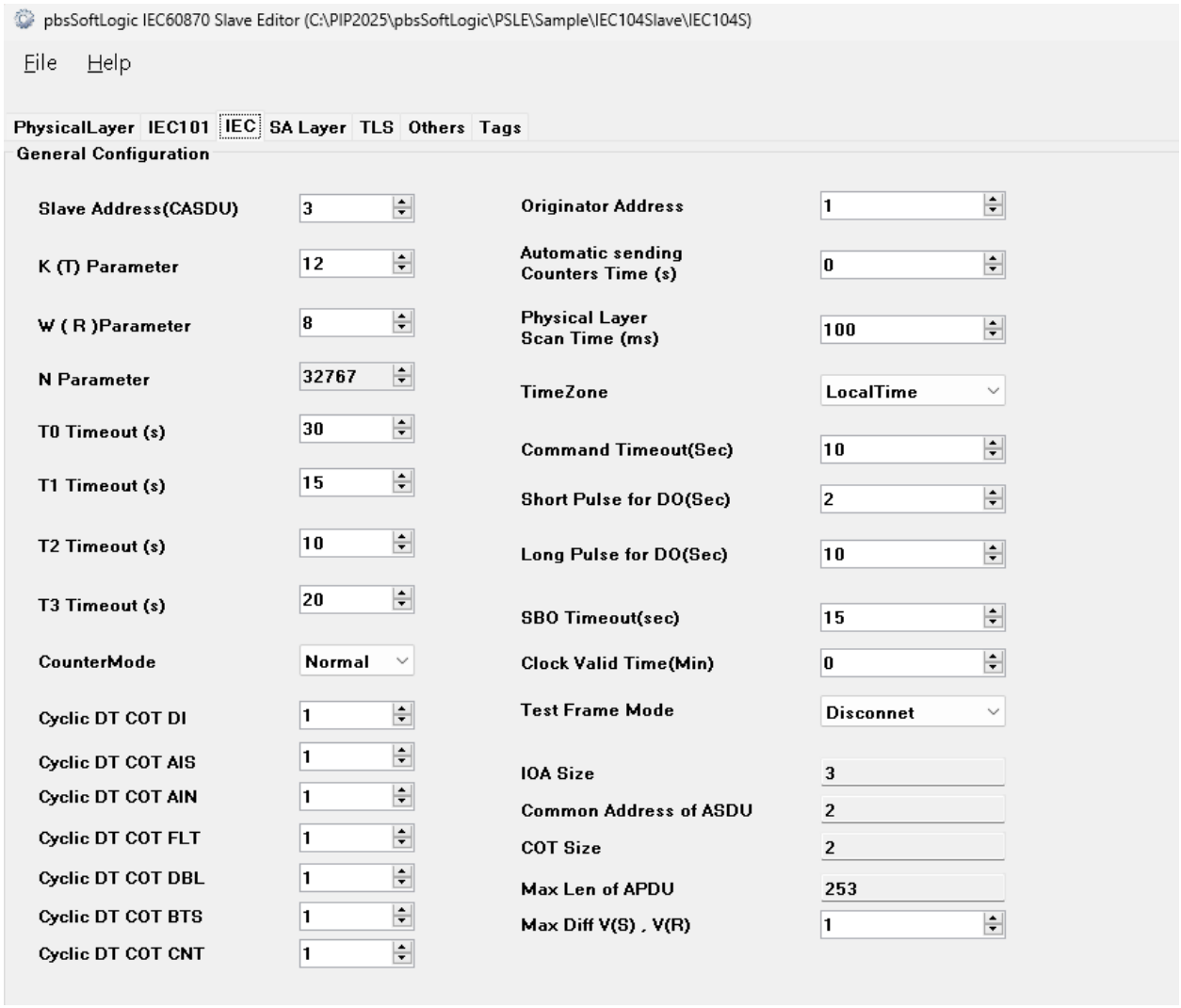


Figure 7

T0 , T1 , T2 , T3 Timeout (Sec)

There are following timeouts in the IEC104 standard that you can set them with T0, T1, T2 and T3 parameters.

Definition of time outs

Parameter	Default value	Remarks	Selected value
t_0	30 s	Time-out of connection establishment	
t_1	15 s	Time-out of send or test APDUs	
t_2	10 s	Time-out for acknowledges in case of no data messages $t_2 < t_1$	
t_3	20 s	Time-out for sending test frames in case of a long idle state	

Maximum range of values for all time-outs: 1 to 255 s, accuracy 1 s.

Cyclic DT COT DI/AIS/AIN/FLT/DBL/BTS/CNT: You can set COT for cyclic data transfer for each tag type. You can select 1 or 2.

In the IEC104 standard one byte is considered for Cause of Transmission (COT).

COT=1, Periodic, Cyclic

COT=2, Background scan

COT=3, Spontaneous, by changes from Slave to master without master request

When you define IEC104 tags in the driver, you can set the periodic transfer time from the Slave to the Master. In the following tag definition for the IEC104 slave driver, AITag33 is transferred to the Master every 5 seconds and AITag34 is transferred every 10 seconds. Other tags are not transferred to the Master periodically.

Name	Type	Class	Init	Address	Period
DITag32	DI-Digital Input (IEC Tag Type 1,30)	1	0	32	0
AITag33	AI-Analog Input (IEC Tag Type 9,34)	1	0	33	5
AITag34	AI-Analog Input (IEC Tag Type 9,34)	1	0	34	10
AITag35	AI-Analog Input (IEC Tag Type 9,34)	1	0	35	0
AITag36	AI-Analog Input (IEC Tag Type 9,34)	1	0	36	0

Command Timeout (Sec): The following ASDU types are timed commands. pbsSoftLogic IEC104 Slave driver does not support type 60.

<58> :=	Single command with time tag CP56Time2a	C_SC_TA_1
<59> :=	Double command with time tag CP56Time2a	C_DC_TA_1
<60> :=	Regulating step command with time tag CP56Time2a	C_RC_TA_1
<61> :=	Set point command, normalized value with time tag CP56Time2a	C_SE_TA_1
<62> :=	Set point command, scaled value with time tag CP56Time2a	C_SE_TB_1
<63> :=	Set point command, short floating point value with time tag CP56Time2a	C_SE_TC_1
<64> :=	Bitstring of 32 bit with time tag CP56Time2a	C_BO_TA_1

When the master sends a command with time, the slave compares the current time with the time in the command. The difference should not be greater than the Command Timeout parameter. Otherwise, the RTU will not execute the command.

Short Pulse for DO (Sec)

Long Pulse for DO (Sec)

For Double/Digital Output commands, the Master can send a short pulse, a long pulse, or set the output to a fixed value.

These are the qualified commands in the IEC104 standard:

<1> := short pulse duration (circuit-breaker), duration determined by a system parameter in the outstation
<2> := long pulse duration, duration determined by a system parameter in the outstation
<3> := persistent output

By using Short Pulse for DO and Long Pulse for DO you can set these pulse times in the RTU.

SBO Timeout (Sec): In SCADA systems, the master can send commands with SBO mode (Selection Before Operate). In SBO mode, the master first selects an output for the RTU; the RTU arms the output and waits for the final command from the master. If the master sends the final command to the RTU in less than the SBO timeout, the RTU executes the command; otherwise the RTU forgets the command.

Test Frame Mode : When T3 is passed in the RTU and the Master does not send a Test Frame to the RTU, then the RTU can send a Test Frame to the Master and wait for an acknowledgement (Send Test Frame Mode) or disconnect (Disconnect Mode) TCP or TLS connection.

IEC62351 Parameters

With the SA layer tab, Figure 8, you can set authentication parameters for the RTU.

SA_Enable : You can enable or disable the IEC62351 layer.

Use a static update key. Dynamic update key is supported but not yet finalized.

The Update key is the same between master and slave. You can generate a new update key by pbsFIT – IEC104 Tester software. Or generate a new key using other IEC104 testers. The key length should be 16 bytes or 32 bytes. You need to include a space between every two bytes and the bytes are in hex format like in Figure 8.

Aggressive Mode : Aggressive mode: If enabled, the master must send aggressive mode for commands. In this mode, the commands and authentication data are placed in a single frame. If false, the master must use challenge replay mode for authentication.

Aggressive Mode Answer: According to IEC62351 standard, when RTU receives aggressive mode commands, RTU should respond to master in IEC104 normal mode. IEC104 testers like DNV tester expect response from RTU in aggressive mode format not IEC104 format. But TMW and pbsFIT expect normal IEC104 response from RTU.

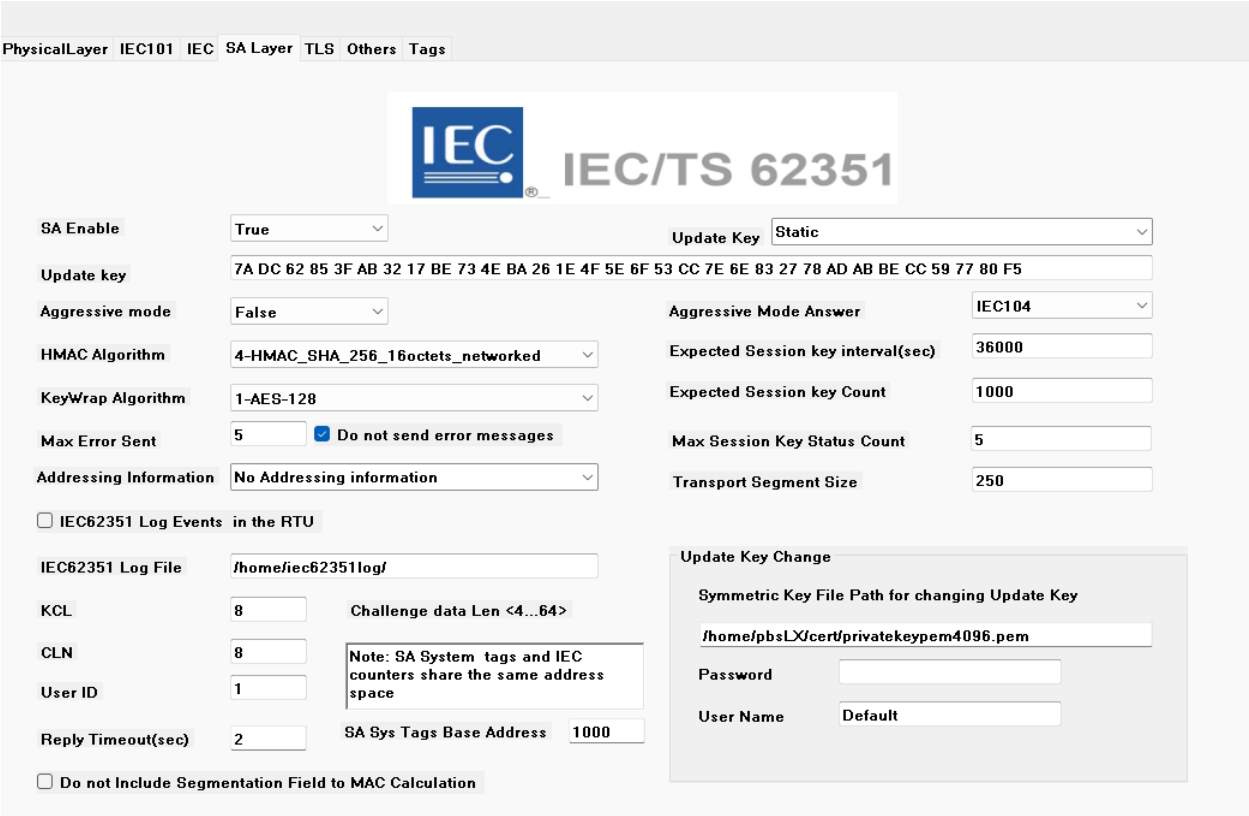


Figure 8

HMAC Algorithm: HMAC is used to authenticate frames. The pbsSoftLogic IEC104 slave driver supports SHA-256_ 8_OCTS and SHA-256_ 16_OCTS. When the master is initializing SA with the RTU, the RTU informs the master about the HMAC supported by the RTU.

KeyWrap Algorithm : Key wrap is used to send the session key from the master to the RTU. The IEC104 slave driver supports AES-128 and AES-256 format for Key Wrap algorithm. When the Master is initializing the SA with the RTU, the RTU informs the Master about the Key wrap supported by the RTU.

Max Error Sent : The number of error frames sent by the RTU to the master. After this limit, the RTU will not send an error message.

Addressing Information: always use No Addressing information

IEC62351 log Events in the RTU

IEC62351 Log File:

To get a certificate for your RTU from IEC labs, you need to save all frames between RTU and Master with all details in log files inside RTU. Use this option only for certificate or debugging communication. Otherwise your RTU flash memory will be damaged.

KCL: Key Challenge data Length

The number of random data sent between the Master and the RTU to change the session key between them. This random data is used by the HMAC algorithm to authenticate both parties.

CLN: Authentication Challenge data Length

When the master sends a critical command to the RTU, the RTU sends an authentication challenge to the master (ID type 81). In the authentication challenge, the RTU sends random data to the master and the CLN is the length of this random data. A maximum of 64 random bytes is allowed.

User ID : In the IEC104 Slave driver, one user is supported. This parameter shows the active user ID.

Replay Timeout (Sec): When RTU sends SA frames like ID 81 which is challenge authentication, and then RTU waits for response from master side. This parameter shows the timeout value in seconds.

Do not include Segmentation Field to HMAC calculation: For tester software like DNV, it omits the Segmentation field when calculating HMAC. But other tools like TMW and pbsFIT include the Segmentation field for calculating HMAC. So when you use DNV tester, enable this parameter.

SA Sys tags base address: When you add a new IEC 104 Slave driver to the project, pbsSoftLogic adds the following SA counters to the project:

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

All system tags that start with "SA_" are IEC62351 counters. The final address for these counters is the sum of the SA Sys base address and the SA counter address. For example, if the base address is 1000, the final counter address for SA_TotalMesagesSentNum is 1008.

Expected Session Key Interval (Sec): When the master is initializing SA with the RTU, it sends a session key to the RTU. After this time elapses, the RTU waits for the new session key and changes the key state to Not Initialized.

Expected session Key count : Like the expected session key interval, but if the number of critical messages exceeds this parameter, the RTU waits for a new session key.

Max Session key Status Count If the number of session key status or session key change commands exceeds this parameter; the RTU sends an error message with a value of 3 to the master.

Transport Segment Size: The original IEC104 standard does not support the transport layer. But in IEC62351, the transport layer is included between the data link and application layers. This parameter indicates the length of the transport layer segmentation.

TLS Parameters

The TLS layer is used to encrypt IEC104 frames. The TLS layer runs over the TCP connection. In Figure 9 you can see the TLS parameters for the IEC104 slave driver.

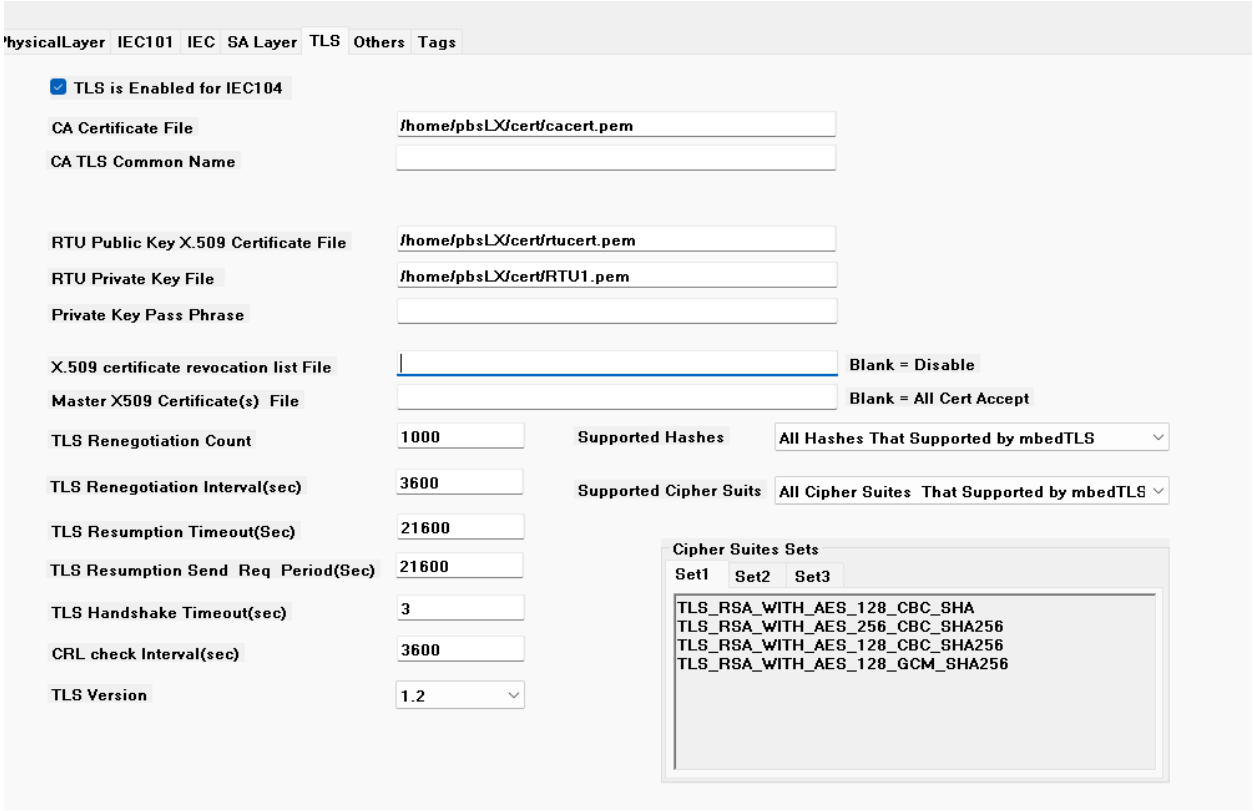


Figure 9

To establish a TLS connection, you must use certificate files for the Master and RTU. The easiest way to create certificate files is to use the free XCA tool. Please refer to
https://www.hohnstaedt.de/xca/index.php/download

With the XCA tool, you must first create a root CA certificate, and then generate the RTU and tester certificates and primary key files.

CA Certificate file : Set the CA certificate file path on the RTU. Assume you have moved the certificate files to the /home/pbsLX/cert folder.

CA Common Name: When you create the CA certificate file, you can set the common name for the certificate. If you set the common name here, the driver compares the certificate common name with the configured name. If they are not the same, the connection is not allowed. If it is empty, the common name is not checked.

RTU certificate file

RTU private key file

Private key pass phrase

Set the full path of the RTU certificate file and private key file here, and if you have encrypted the private key file with a passkey, you must set the passkey here.

X509 revocation list file

You can create a revocation list with the XCA tool. The revocation certificate is not allowed to connect to the RTU. If the revocation list is empty, the revocation list is disabled.

Master certificate file: If empty, the RTU does not check the master certificate, otherwise the RTU compares the master certificate received in the handshake process with the configured certificate path.

The definition and function of other parameters are quite clear.

Others parameters

In Figure 10 you can see other parameters of the IEC104 slave driver.

Diagnostic mode

If you enable diagnostic mode, you can view frames in the RTU console if you manually run the pbsSoftLogic execution core.

To run the pbsSoftLogic runtime on the RTU, connect to the RTU using an SSH client tool and run the following commands:

cd /home/pbsLX

pkill pbsSLKLX

./pbsSLKLX

BufferAtFlash

Buffer Path

If you want to save IEC104 offline events to the RTU flash, you can enable and set the file path with these parameters. It is usually set for RTUs that have an external SD card for data logging.

Enable Buffering

Buffering is a very important and practical concept in pbsSoftLogic. Please refer to the specific manual dedicated to buffering concepts in pbsSoftLogic.

Other parameters are internal to the pbsSoftLogic operation.

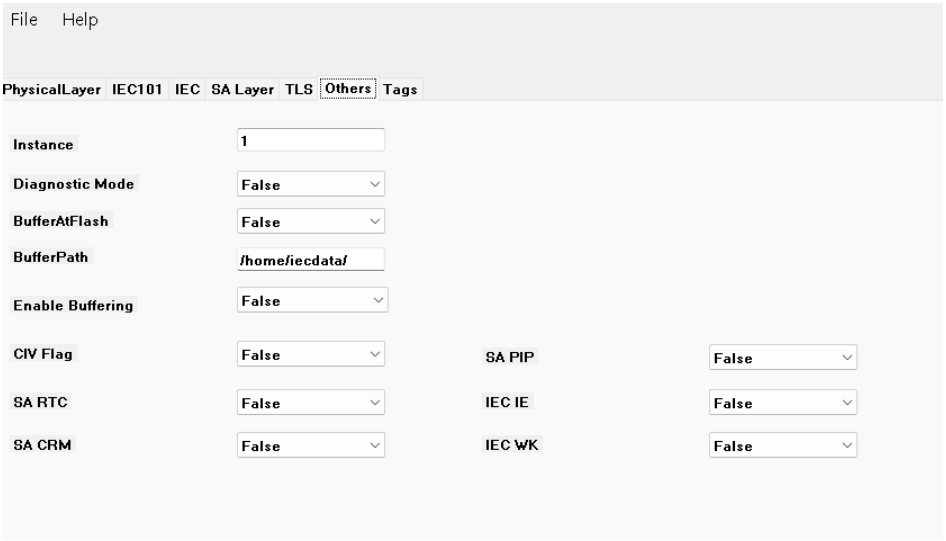


Figure 10

IEC104 Tags

When you add a new IEC104 Slave driver to the project, default parameters and tags are added to the project. The following IEC data types are supported in the slave driver.

TYPE IDENT 1: M_SP_NA_1
Single-point information without time tag

TYPE IDENT 3: M_DP_NA_1
Double-point information without time tag

TYPE IDENT 7: M_BO_NA_1
Bitstring of 32 bit

TYPE IDENT 9: M_ME_NA_1
Measured value, normalized value

TYPE IDENT 11: M_ME_NB_1
Measured value, scaled value

TYPE IDENT 13: M_ME_NC_1
Measured value, short floating point number

TYPE IDENT 15: M_IT_NA_1
Integrated totals

TYPE IDENT 30: M_SP_TB_1
Single-point information with time tag CP56Time2a

TYPE IDENT 31: M_DP_TB_1
Double-point information with time tag CP56Time2a

TYPE IDENT 33: M_BO_TB_1
Bitstring of 32 bits with time tag CP56Time2a

TYPE IDENT 34: M_ME_TD_1
Measured value, normalized value with time tag CP56Time2a

TYPE IDENT 35: M_ME_TE_1
Measured value, scaled value with time tag CP56Time2a

TYPE IDENT 36: M_ME_TF_1
Measured value, short floating point number with time tag CP56Time2a

TYPE IDENT 37: M_IT_TB_1
Integrated totals with time tag CP56Time2a

TYPE IDENT 45: C_SC_NA_1
Single command

TYPE IDENT 46: C_DC_NA_1
Double command

TYPE IDENT 48: C_SE_NA_1
Set-point command, normalized value

TYPE IDENT 49: C_SE_NB_1
Set-point command, scaled value

TYPE IDENT 50: C_SE_NC_1
Set-point command, short floating point number

TYPE IDENT 51: C_BO_NA_1
Bitstring of 32 bit

TYPE IDENT 110: P_ME_NA_1
Parameter of measured values, normalized value

TYPE IDENT 111: P_ME_NB_1
Parameter of measured values, scaled value

TYPE IDENT 112: P_ME_NC_1
Parameter of measured values, short floating point number

TYPE IDENT 113: P_AC_NA_1
Parameter activation

TYPE IDENT 58: C_SC_TA_1
Single command with time tag CP56Time2a

TYPE IDENT 59: C_DC_TA_1
Double command with time tag CP56Time2a

TYPE IDENT 61: C_SE_TA_1
Set-point command with time tag CP56Time2a, normalized value

TYPE IDENT 62: C_SE_TB_1
Set-point command with time tag CP56Time2a, scaled value

TYPE IDENT 63: C_SE_TC_1
Set-point command with time tag CP56Time2a, short floating point number

TYPE IDENT 64: C_BO_TA_1
Bitstring of 32 bit with time tag CP56Time2a

Other types supported in the driver but classified as functions are as follows:

TYPE IDENT 70: M_EI_NA_1
End of initialization

TYPE IDENT 100: C_IC_NA_1
Interrogation command

TYPE IDENT 101: C_CI_NA_1
Counter interrogation command

TYPE IDENT 102: C_RD_NA_1
Read command

TYPE IDENT 103: C_CS_NA_1
Clock synchronization command

TYPE IDENT 104: C_TS_NA_1
Test command

TYPE IDENT 106: C_CD_NA_1
Delay acquisition command

TYPE IDENT 107: C_TS_TA_1
Test command with time tag CP56Time2a

TYPE IDENT 81: S_CH_NA_1
Authentication challenge

TYPE IDENT 82: S_RP_NA_1
Authentication Replay

TYPE IDENT 83: S_AR_NA_1
Aggressive mode Authentication request

TYPE IDENT 84: S_KR_NA_1
Session key status request

TYPE IDENT 85: S_KS_NA_1
Session key status

TYPE IDENT 86: S_KC_NA_1
Session key change

TYPE IDENT 87: S_ER_NA_1
Authentication Error

Following properties can be defined for each IEC104 tag:

- Tag Name
- Tag Type
- Init Value
- Address
- Period
- Log
- Description

pbsSoftLogic Tag Types are as following :

DI : 1 – 30

DO: 45 – 58

AI : 9 - 34

AO :48 - 61

CNT : 15 - 37

FI : 13 - 36

FO : 50 - 63

DPI : 3 - 31

DPO : 46 - 59

AI11 : 11 - 35

AO49 : 49 – 62

BSI : 7 – 33

BSO : 51 – 64

Address : Since the Read function - Type ID 102 - in IEC104 standard only accept the tag address from the RTU for reading, so we assume a unique address for all tags. You can assume any address but it must be unique for all tags.

7.3.4.3 TYPE IDENT 102: C_RD_NA_1
Read command

Single information object (SQ = 0)

0 1 1 0 0 1 1 0	TYPE IDENTIFICATION	
0 0 0 0 0 0 0 1	VARIABLE STRUCTURE QUALIFIER	DATA UNIT IDENTIFIER Defined in 7.1
Defined in 7.2.3	CAUSE OF TRANSMISSION	
Defined in 7.2.4	COMMON ADDRESS OF ASDU	
Defined in 7.2.5	INFORMATION OBJECT ADDRESS	INFORMATION OBJECT

IEC 158/03

Period: If set to a non-zero value, the RTU will send the tag with a cyclic transmission with COT 1 or 2.

The value is in seconds. If you set Period to 5 seconds, the RTU will send the signal to the master with COT 1 or 2, which is defined as a parameter for each data type.

Log: If set for a signal, especially set point signals sent by the master to the RTU, the last value of the signal is stored in the RTU flash and when the RTU is restarted, the last value for the set point is loaded.

Sample Tag definition in IECtags.xml

```
<Tag Name="DITag1" Type="DI" Class="1" Init="0" Address="1" Period="0" Log="0" Desc="" />
<Tag Name="AITag33" Type="AI" Class="1" Init="0" Address="33" Period="5" Log="0" Desc="" />
 <Tag Name="AITag34" Type="AI" Class="1" Init="0" Address="34" Period="10" Log="0" Desc="" />
<Tag Name="AISCTag65" Type="AI11" Class="1" Init="0" Address="65" Period="0" Log="0" Desc="" />
<Tag Name="AISCTag66" Type="AI11" Class="1" Init="0" Address="66" Period="0" Log="0" Desc="" />
<Tag Name="FITag97" Type="FI" Class="1" Init="0" Address="97" Period="0" Log="0" Desc="" />
  <Tag Name="FITag98" Type="FI" Class="1" Init="0" Address="98" Period="0" Log="0" Desc="" />
<Tag Name="CNTTag130" Type="CNT" Class="1" Init="0" Address="130" Period="0" Log="0" Desc="" />
<Tag Name="CNTTag131" Type="CNT" Class="1" Init="0" Address="131" Period="0" Log="0" Desc="" />
<Tag Name="DPITag161" Type="DPI" Class="1" Init="0" Address="161" Period="0" Log="0" Desc="" />
<Tag Name="DOTag193" Type="DO" Class="1" Init="0" Address="193" Period="0" Log="0" Desc="" />
<Tag Name="AOTag209" Type="AO" Class="1" Init="0" Address="209" Period="0" Log="1" Desc="" />
```

<Tag Name="AOSCTag225" Type="AO49" Class="1" Init="0" Address="225" Period="0" Log="0" Desc=""/>

<Tag Name="FOTag241" Type="FO" Class="1" Init="0" Address="241" Period="0" Log="1" Desc="" />

<Tag Name="DPOTag257" Type="DPO" Class="1" Init="0" Address="257" Period="0" Log="0" Desc="" />

<Tag Name="BSITag273" Type="BSI" Class="1" Init="0" Address="273" Period="0" Log="0" Desc="" />

<Tag Name="BSITag287" Type="BSI" Class="1" Init="0" Address="287" Period="0" Log="0" Desc="" />

<Tag Name="BSOTag294" Type="BSO" Class="1" Init="0" Address="294" Period="0" Log="0" Desc="" />

You need to modify the default tag list based on your project needs. You can use Notepad++ editor and create new tag list based on your project requirement.

FB Programming

After defining IEC104 tags and setting parameters, you can use the tags in your Function Block program.

In Figure 11, a simple program that writes the output of a pulse generator to a DI tag is shown.

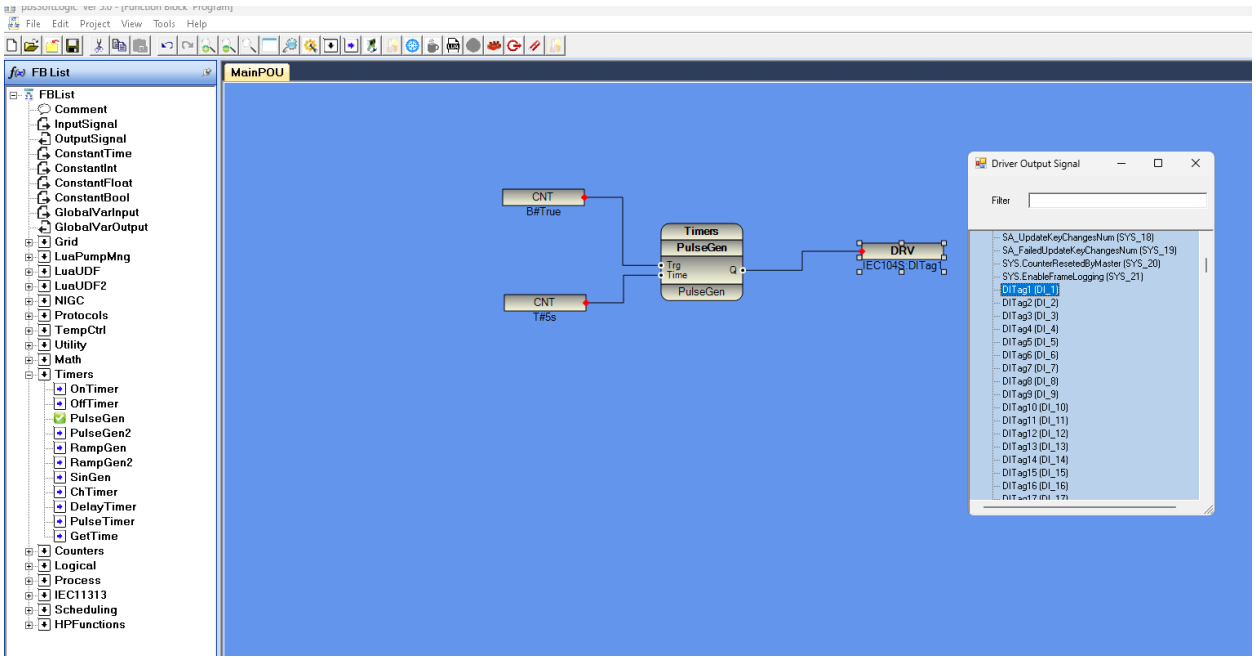


Figure 11

All input signals (DI, AI, AI11, FI, DPI, CNT, BSI) must be placed on the right side of a Function Block to write a value to the driver.

In Figure 12, the pulse generator output is counted by a CTU function block and written to an IEC104 tag with ID type 11 and address 65.

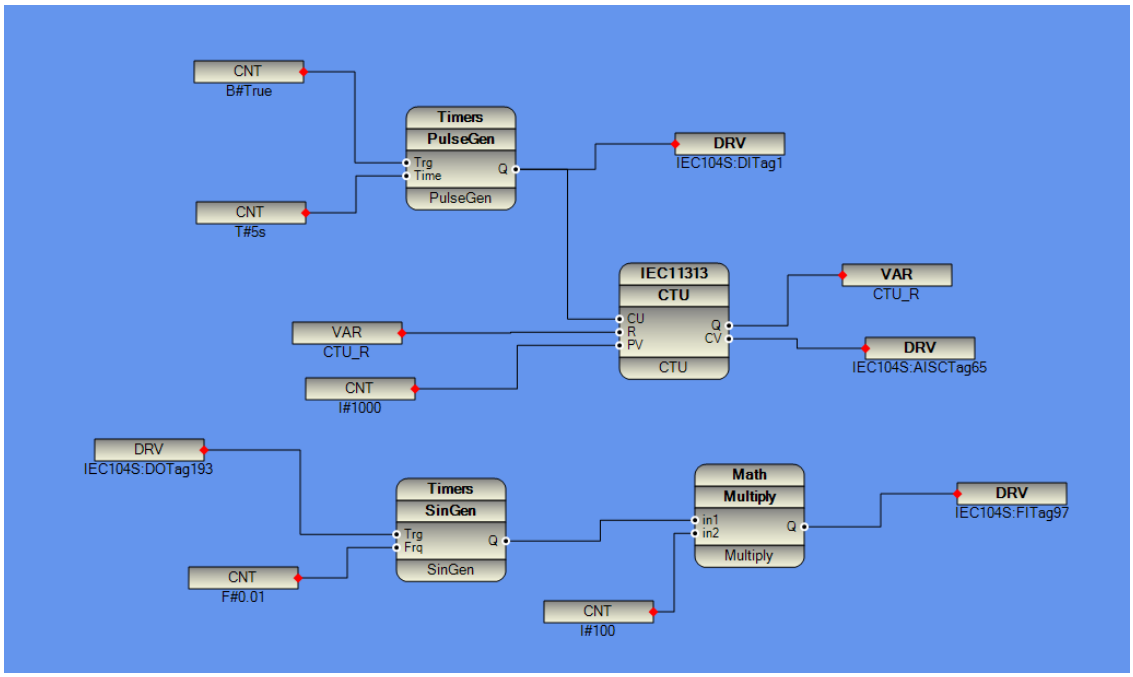


Figure 12

In Figure 12, you can see that DOTag193 is connected to the trigger input of a sine generator function block, and the output is written to FITag97 after being multiplied by 100.

Transfer the logic and configuration to the RTU and reset the RTU. Then connect to the RTU with pbsSoftLogic and monitor the logic. Figure 13

You can use pbsFIT or any other IEC104 tester and connect to the RTU. We use pbsFIT and send a DO command to the DOTag193 to the RTU and start the sine generator function block. Figure 13 , 14

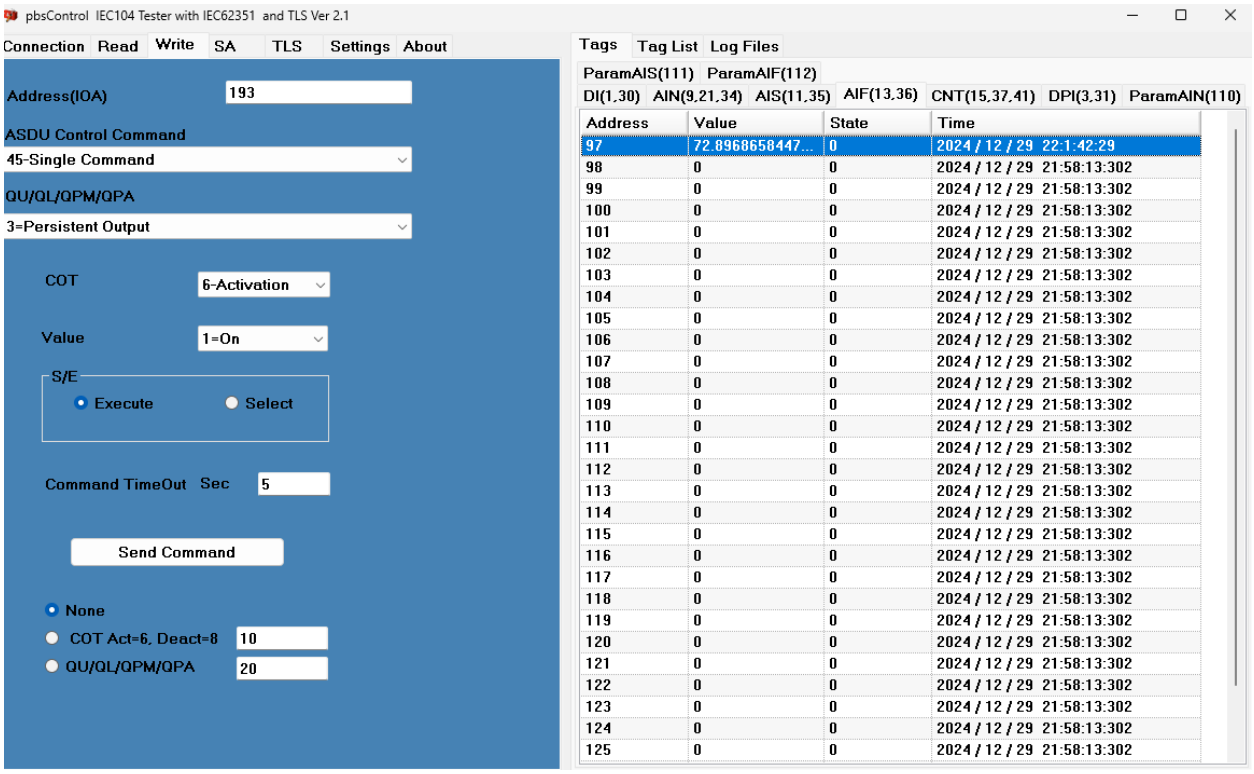


Figure 13

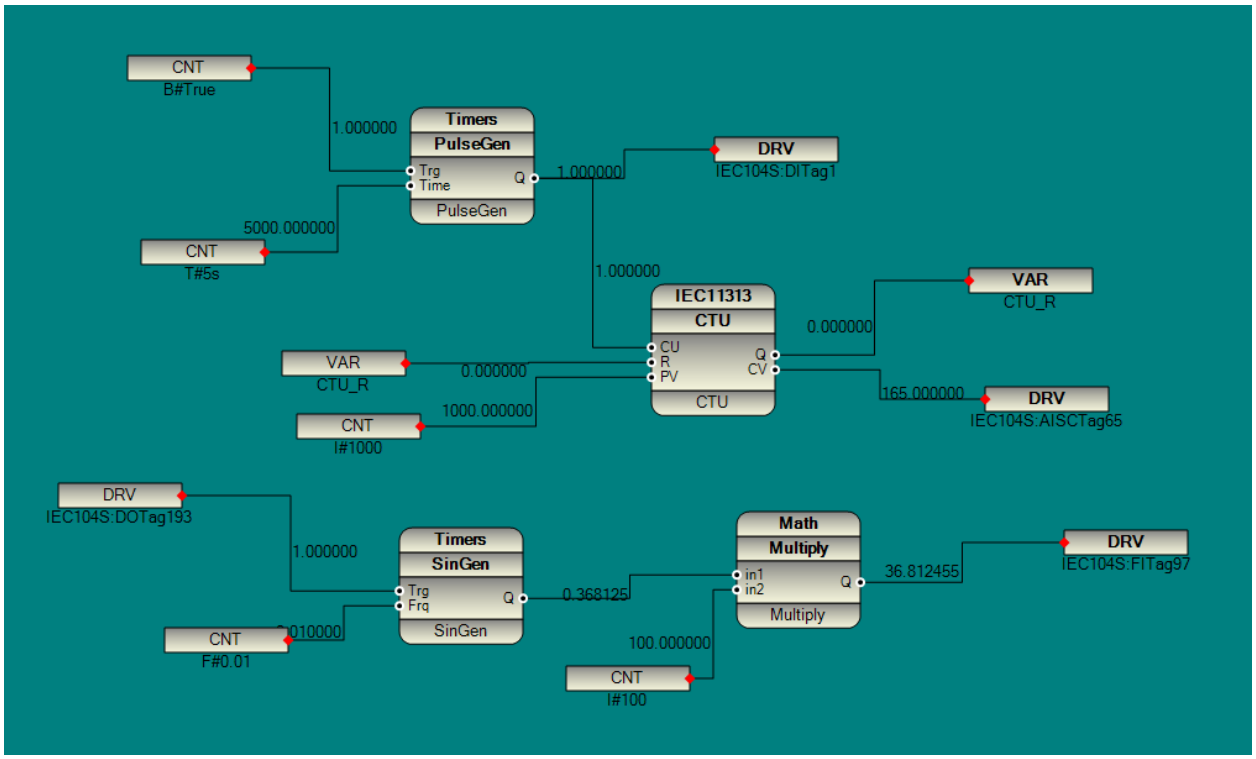


Figure 14

Using system tags, you can monitor the main connection status, GI status, and CI Reset status, as in the following program.. Figure 15

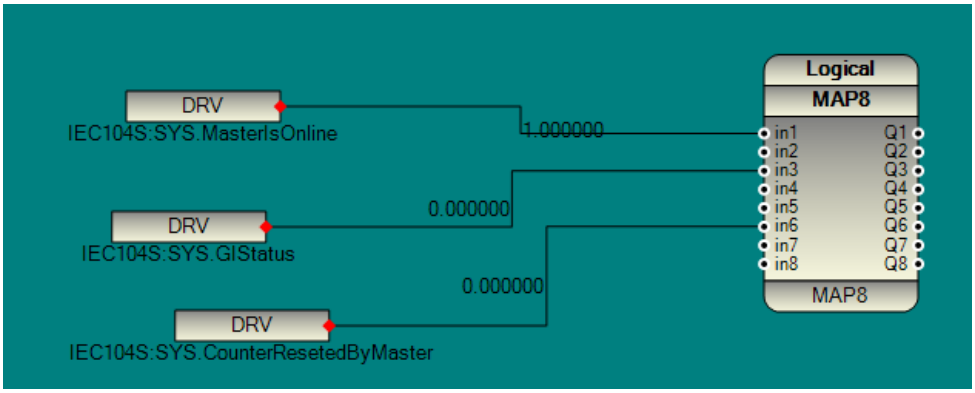


Figure 15

SYS.MasterIsOnline : When the Master connects to the RTU, this signal changes to 1. If the Master is disconnected, the signal value changes to 0.

SYS.GIStatus : When the Master sends a GI to the RTU, this signal value changes to 1 for 5 seconds. Figure 16

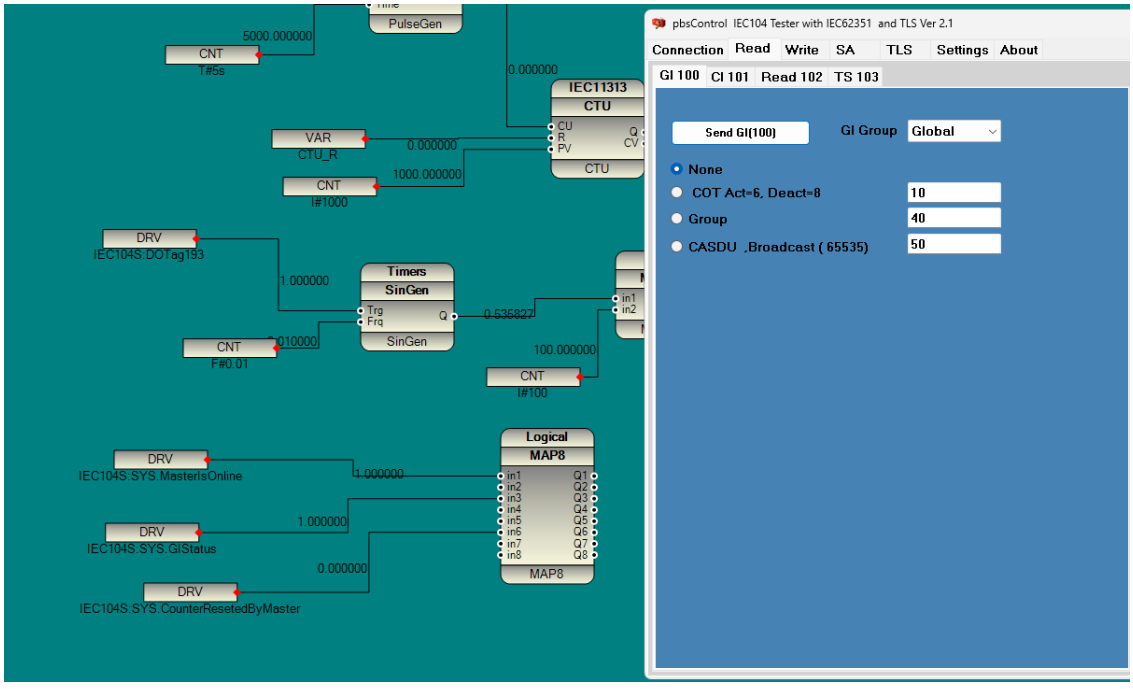


Figure 16

SYS.CounterResetedByMaster : When the Master resets the counters, this signal changes to 1 for five seconds. Figure 17

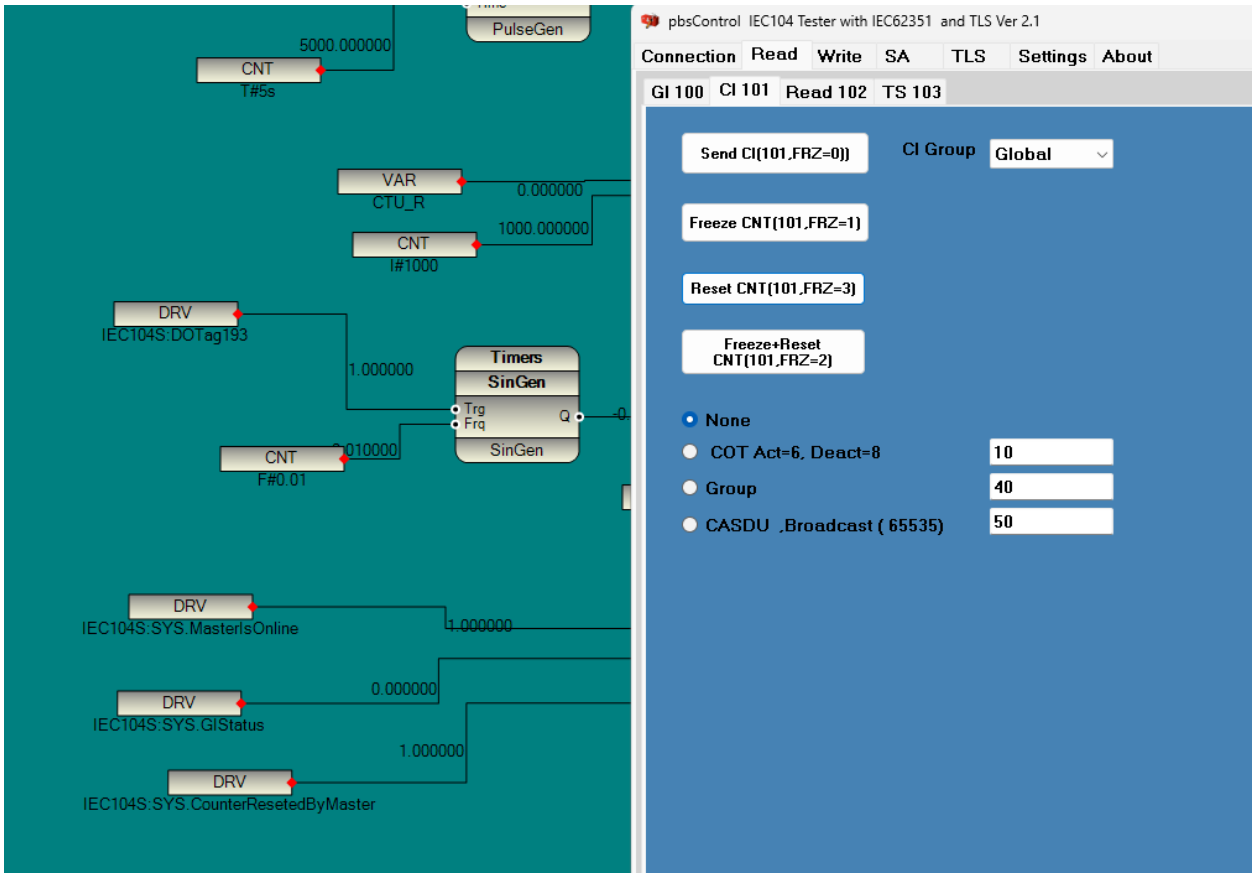


Figure 17

SYS.EnableFrameLogging : When changed to 1, IEC62351 frame recording is enabled in the RTU, when changed to 0, frame recording is disabled and the current file in the RTU is closed.

Other system signals are SA counters. These counters only work when IEC62351 is enabled. The initial value for SA counters is the maximum value defined in the IEC62351 standard. SA counters are only transferred when their value exceeds the maximum value.

IEC Tags state (Quality Descriptor)

You can set input tag state in the logic. In the Tags tab , you can check quality tag for input signals . Figure 18

PhysicalLayer IEC101 IEC SA Layer TLS Others Tags

Name	Type	Class	Init	Address	Period	Log	Quality Tag	Desc
SYS.MasterIsOnline	SYS-System Diagnostic	0	0	1	0	☐	☐	
SYS.GIStatus	SYS-System Diagnostic	0	0	2	0	☐	☐	
SA_UnexpectedMessagesNum	SYS-System Diagnostic	0	3	3	0	☐	☐	
SA_AuthorizationFailuresNum	SYS-System Diagnostic	0	5	4	0	☐	☐	
SA_AuthenticationFailuresNum	SYS-System Diagnostic	0	5	5	0	☐	☐	
SA_ReplyTimeoutsNum	SYS-System Diagnostic	0	3	6	0	☐	☐	
SA_RekeysDueToAuthenticationFailureNum	SYS-System Diagnostic	0	3	7	0	☐	☐	
SA_TotalMessagesSentNum	SYS-System Diagnostic	0	100	8	0	☐	☐	
SA_TotalMessagesReceivedNum	SYS-System Diagnostic	0	100	9	0	☐	☐	
SA_CriticalMessagesSentNum	SYS-System Diagnostic	0	100	10	0	☐	☐	
SA_CriticalMessagesReceivedNum	SYS-System Diagnostic	0	100	11	0	☐	☐	
SA_DiscardedMessagesNum	SYS-System Diagnostic	0	10	12	0	☐	☐	
SA_ErrorMessagesSentNum	SYS-System Diagnostic	0	10	13	0	☐	☐	
SA_ErrorMessagesReceivedNum	SYS-System Diagnostic	0	10	14	0	☐	☐	
SA_SuccessfulAuthenticationsNum	SYS-System Diagnostic	0	100	15	0	☐	☐	
SA_SessionKeyChangesNum	SYS-System Diagnostic	0	10	16	0	☐	☐	
SA_FailedSessionKeyChangesNum	SYS-System Diagnostic	0	5	17	0	☐	☐	
SA_UpdateKeyChangesNum	SYS-System Diagnostic	0	1	18	0	☐	☐	
SA_FailedUpdateKeyChangesNum	SYS-System Diagnostic	0	1	19	0	☐	☐	
SYS.CounterResetedByMaster	SYS-System Diagnostic	0	0	20	0	☐	☐	
SYS.EnableFrameLogging	SYS-System Diagnostic	0	0	21	0	☐	☐	
DITag1	DI-Digital Input (IEC Tag Type 1,30)	1	0	1	0	☐	☑	
DITag2	DI-Digital Input (IEC Tag Type 1,30)	1	0	2	0	☐	☑	
DITag3	DI-Digital Input (IEC Tag Type 1,30)	1	0	3	0	☐	☐	
DITag4	DI-Digital Input (IEC Tag Type 1,30)	1	0	4	0	☐	☐	
DITag5	DI-Digital Input (IEC Tag Type 1,30)	1	0	5	0	☐	☐	

Figure 18

When you save this configuration, the IEC104 Slave Editor adds a new tag to the tag list with the same name but with ".s" added to the name, the same address and the type changed to state type which is the same input type but with an added .S at the end. For example if the tag type is DI, the state tag type is DIS. Figure 19

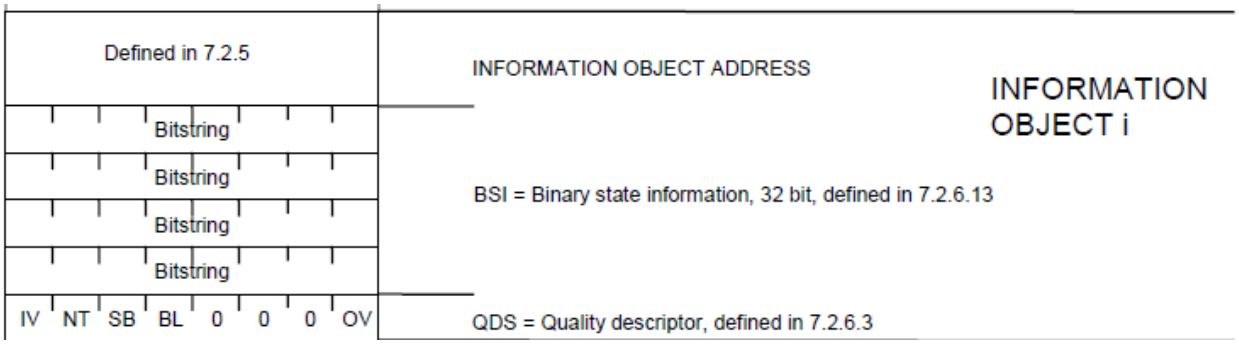
```
<Tag Name="DITag1" Type="DI" Class="1" Init="0" Address="1" Period="0" Log="0" Desc="" />
<Tag Name="DITag1.s" Type="DIS" Class="1" Init="0" Address="1" Period="0" Log="0" Desc="_Quality" />
<Tag Name="DITag2" Type="DI" Class="1" Init="0" Address="2" Period="0" Log="0" Desc="" />
<Tag Name="DITag2.s" Type="DIS" Class="1" Init="0" Address="2" Period="0" Log="0" Desc="_Quality" />
```

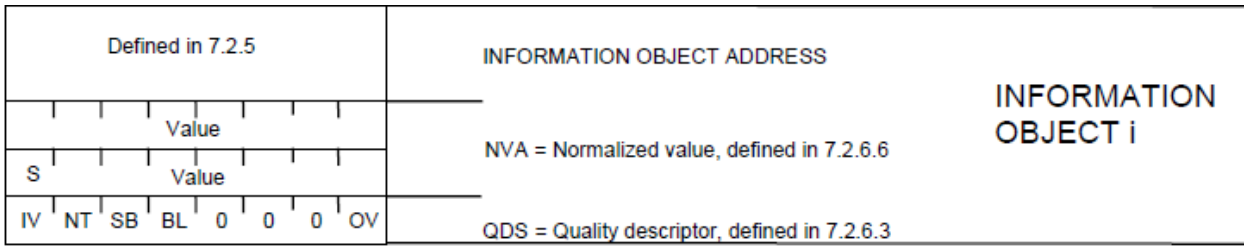
Figure 19

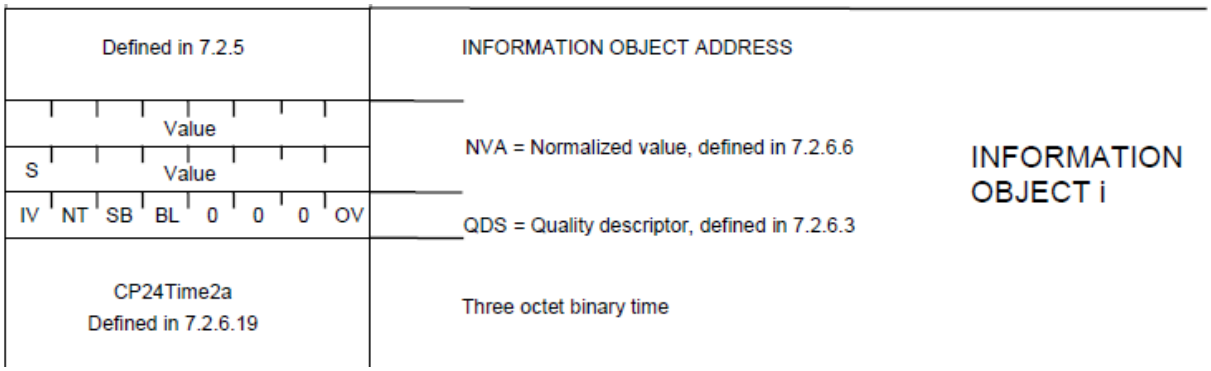
In the IEC104 standard for input data types we have following quality descriptor:

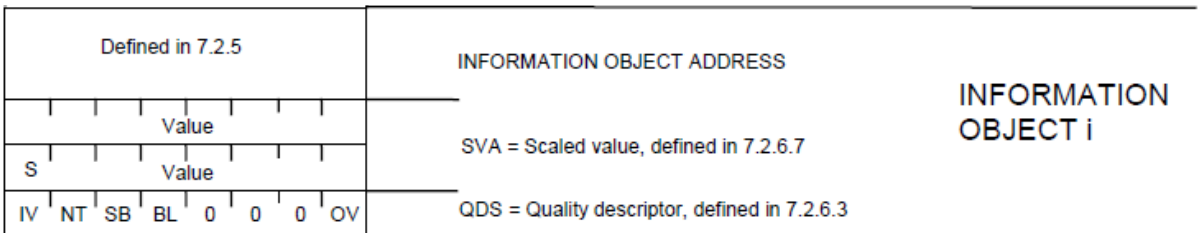
Defined in 7.2.5								INFORMATION OBJECT ADDRESS	INFORMATION OBJECT i
IV	NT	SB	BL	0	0	0	SPI	SIQ = Single-point information with quality descriptor, defined in 7.2.6.1	

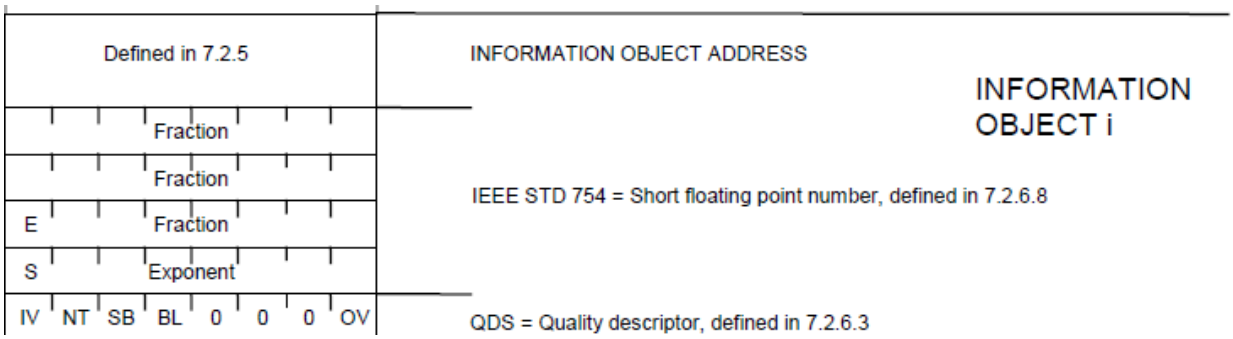
Defined in 7.2.5							INFORMATION OBJECT ADDRESS	INFORMATION OBJECT i
IV	NT	SB	BL	0	0	DPI	DIQ = Double-point information with quality descriptor, defined in 7.2.6.2	

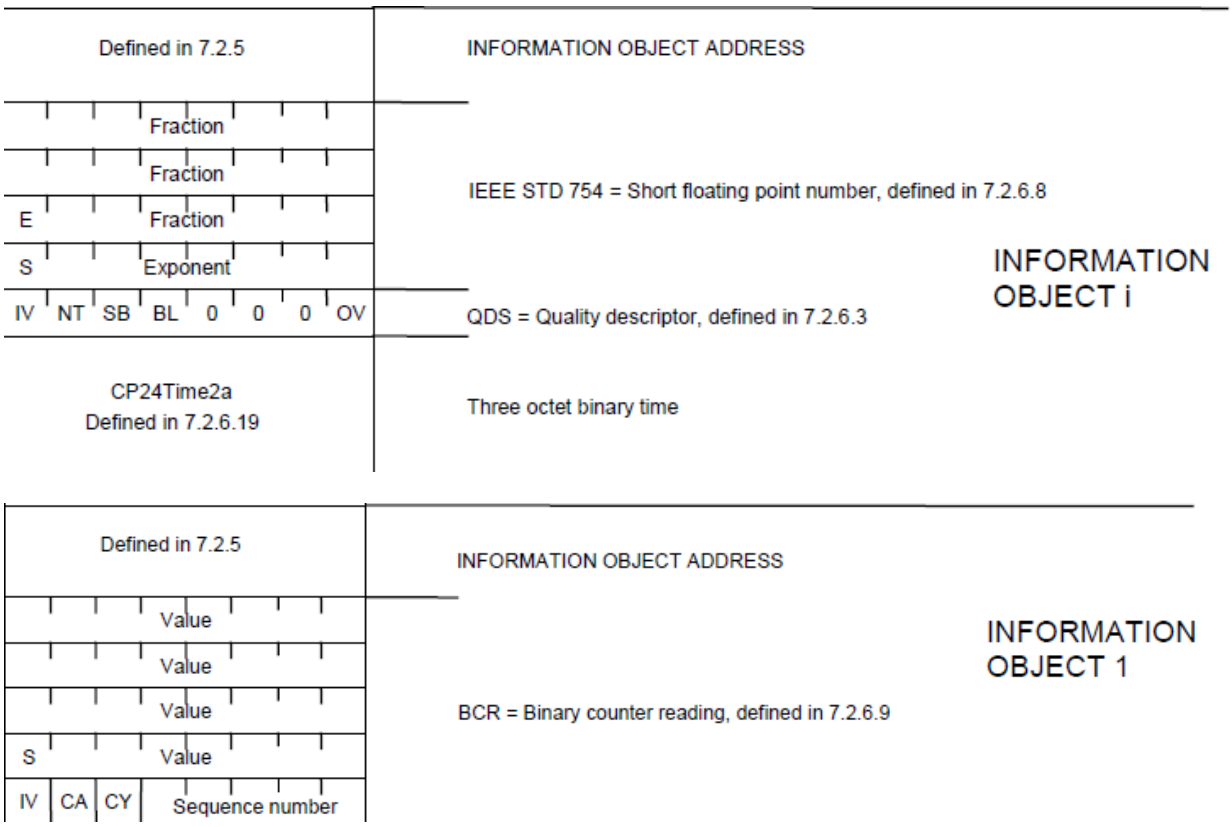

Defined in 7.2.5
INFORMATION OBJECT ADDRESS
INFORMATION OBJECT i
Bitstring
Bitstring
Bitstring
Bitstring
BSI = Binary state information, 32 bit, defined in 7.2.6.13
IV NT SB BL 0 0 0 OV
QDS = Quality descriptor, defined in 7.2.6.3


Defined in 7.2.5
INFORMATION OBJECT ADDRESS
INFORMATION OBJECT i
Value
S Value
NVA = Normalized value, defined in 7.2.6.6
IV NT SB BL 0 0 0 OV
QDS = Quality descriptor, defined in 7.2.6.3


Defined in 7.2.5
INFORMATION OBJECT ADDRESS
INFORMATION OBJECT i
Value
S Value
NVA = Normalized value, defined in 7.2.6.6
IV NT SB BL 0 0 0 OV
QDS = Quality descriptor, defined in 7.2.6.3
CP24Time2a
Defined in 7.2.6.19
Three octet binary time


Defined in 7.2.5
INFORMATION OBJECT ADDRESS
INFORMATION OBJECT i
Value
S Value
SVA = Scaled value, defined in 7.2.6.7
IV NT SB BL 0 0 0 OV
QDS = Quality descriptor, defined in 7.2.6.3


Defined in 7.2.5
INFORMATION OBJECT ADDRESS
INFORMATION OBJECT i
Fraction
Fraction
E Fraction
S Exponent
IEEE STD 754 = Short floating point number, defined in 7.2.6.8
IV NT SB BL 0 0 0 OV
QDS = Quality descriptor, defined in 7.2.6.3


Defined in 7.2.5
INFORMATION OBJECT ADDRESS
Fraction
Fraction
E
Fraction
S
Exponent
IEEE STD 754 = Short floating point number, defined in 7.2.6.8
INFORMATION OBJECT i
IV NT SB BL 0 0 0 OV
QDS = Quality descriptor, defined in 7.2.6.3
CP24Time2a
Defined in 7.2.6.19
Three octet binary time
Defined in 7.2.5
INFORMATION OBJECT ADDRESS
Value
Value
Value
S
Value
INFORMATION OBJECT 1
BCR = Binary counter reading, defined in 7.2.6.9
IV CA CY
Sequence number

7.2.6.3 Quality descriptor (separate octet)

The quality descriptor consists of five defined quality bits which may be set independently from each other. The quality descriptor provides the controlling station with additional information on the quality of an information object.

QDS		:=	CP8{OV,RES,BL,SB,NT,IV}	
OV		:=	BS1[1]<0..1>	(Type 6)
	<0>	:=	no overflow	
	<1>	:=	overflow	
RES = RESERVE		:=	BS3[2..4]<0>	(Type 6)
BL		:=	BS1[5]<0..1>	(Type 6)
	<0>	:=	not blocked	
	<1>	:=	blocked	
SB		:=	BS1[6]<0..1>	(Type 6)
	<0>	:=	not substituted	
	<1>	:=	substituted	
NT		:=	BS1[7]<0..1>	(Type 6)
	<0>	:=	topical	
	<1>	:=	not topical	
IV		:=	BS1[8]<0..1>	(Type 6)
	<0>	:=	valid	
	<1>	:=	invalid	

OV = OVERFLOW/NO OVERFLOW

The value of the INFORMATION OBJECT is beyond a predefined range of value (mainly applicable to analog values).

BL = BLOCKED/NOT BLOCKED

The value of the INFORMATION OBJECT is blocked for transmission; the value remains in the state that was acquired before it was blocked. Blocking and deblocking may be initiated for example by a local lock or a local automatic cause.

SB = SUBSTITUTED/NOT SUBSTITUTED

The value of the INFORMATION OBJECT is provided by the input of an operator (dispatcher) or by an automatic source.

NT = NOT TOPICAL/TOPICAL

A value is topical if the most recent update was successful. It is not topical if it was not updated successfully during a specified time interval or if it is unavailable.

IV = INVALID/VALID

A value is valid if it was correctly acquired. After the acquisition function recognizes abnormal conditions of the information source (missing or non-operating updating devices) the value is then marked invalid. The value of the INFORMATION OBJECT is not defined under this condition. The mark INVALID is used to indicate to the destination that the value may be incorrect and cannot be used.

7.2.6.9 Binary counter reading

BCR	:=	CP40{Counter reading, Sequence notation}	
Counter reading	:=	I32[1..32]<$-2^{31}..+2^{31}-1$>	(Type 2.1)
Sequence notation	:=	CP8{SQ,CY,CA,IV}	
SQ	:=	UI5[33..37]<0..31>	(Type 1.1)
CY	:=	BS1[38]<0..1>	(Type 6)
<0>	:=	no counter overflow occurred in the corresponding integration period	
<1>	:=	counter overflow occurred in the corresponding integration period	
CA	:=	BS1[39]<0..1>	(Type 6)
<0>	:=	counter was not adjusted since last reading	
<1>	:=	counter was adjusted since last reading	
IV	:=	BS1[40]<0..1>	(Type 6)
<0>	:=	counter reading is valid	
<1>	:=	counter reading is invalid	

SQ = sequence number
CY = carry
(Counter overflow occurs when the value increments from $+2^{31}-1$ to zero or from -2^{31} to zero)
CA = counter was adjusted
(The counter is considered to have been adjusted if a counter is initialized to some value, for example set to zero or another value at startup).
IV = invalid

Note that CA, CY and IV are only modified when the value is determined. This may be in response to a counter interrogation command or in response to an automatic internal function that performs the counter freeze or freeze and reset command.

You can set the value of quality descriptor tags with the following function blocks: Figure 20

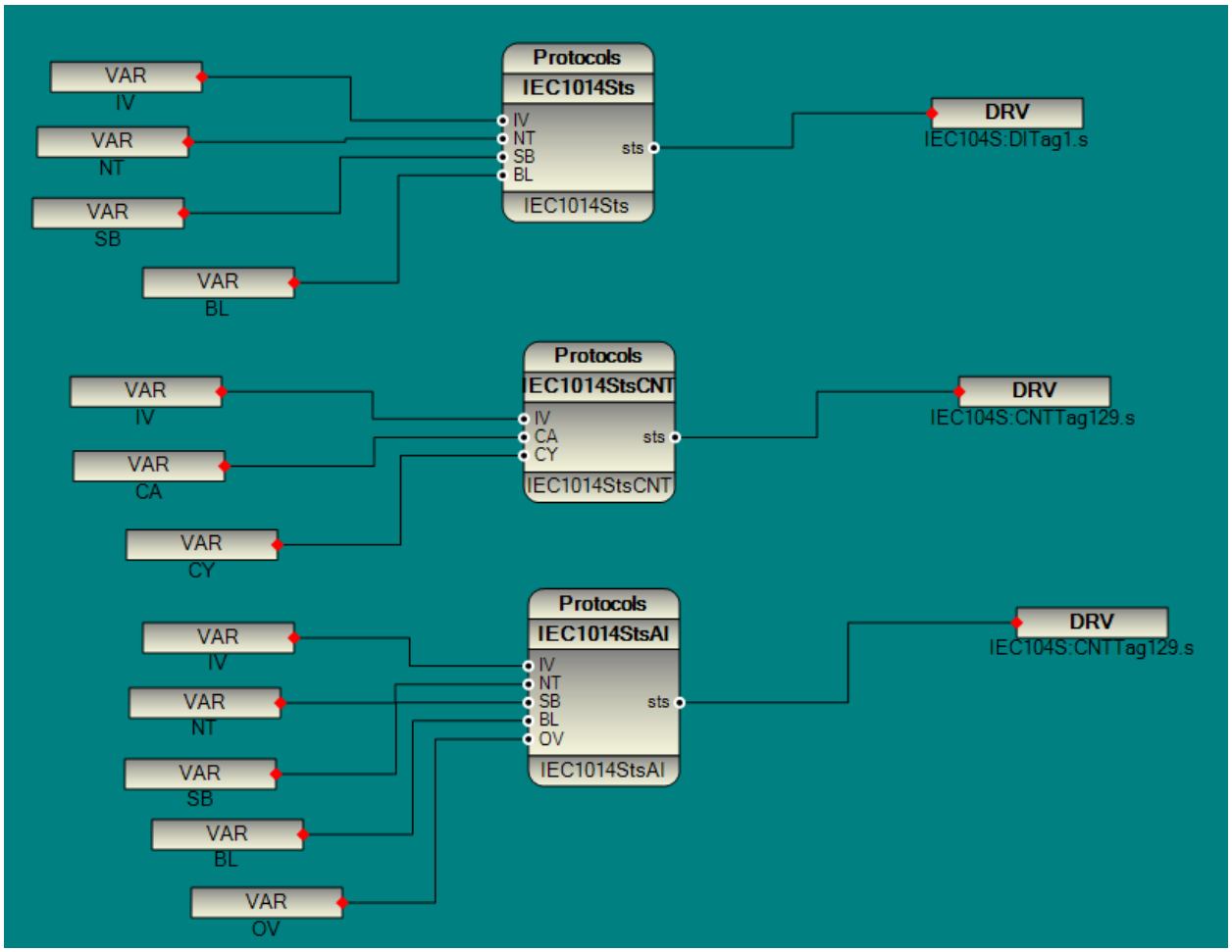
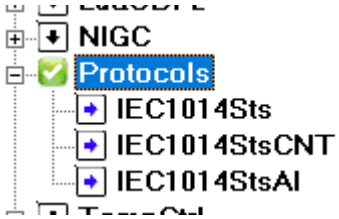


Figure 20

The value of IV, NT, SB, BL, ... is set from external devices or protocols. For example, the IV value of a counter is read from a gas meter, you can link the IV signal to the communication status of the gas meter.

These function blocks are placed at Protocols group:



IEC104 Counters mode

You can define 2 mode for IEC104 counters : Figure 21

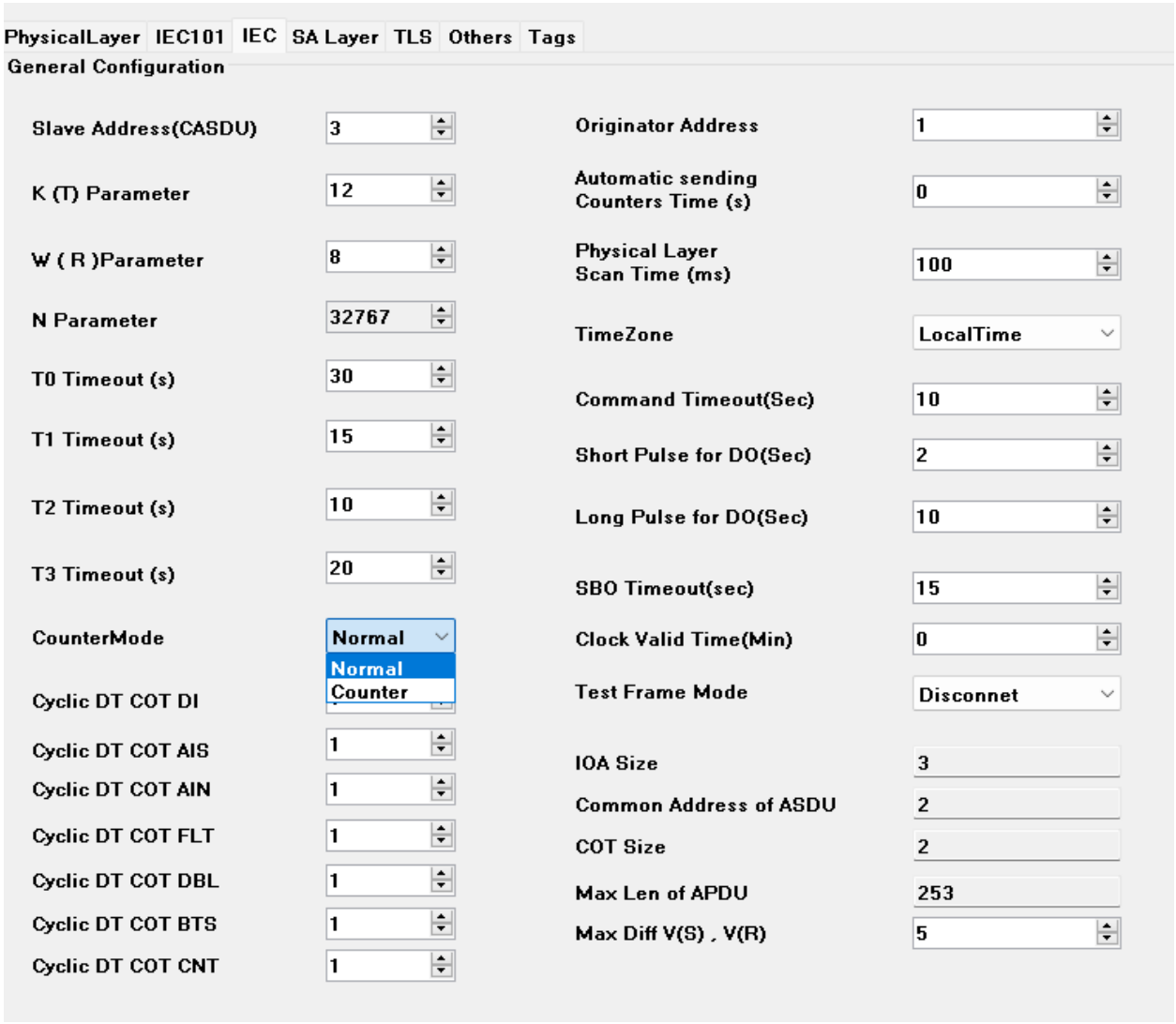


Figure 21

Normal mode: In normal mode, the IEC104 driver does not count the signal and you have to write the counter value to the signal. Suppose you read a counter from a power meter by modbus and write the final value to an IEC104 counter. In this mode, the IEC104 driver will transfer the counter value to the Master as per the standard. If you reset this counter, the counter value will not be reset because the driver does not have direct access to the counter. In this mode, you have to use the SYS.CounterResetedByMaster tag and reset all the counters in your logic.

Counter mode: In counter mode, you connect the digital input signal to the counter and the driver counts the signal changes. In this mode, if the master resets the counters, the driver can reset the counters value. **End of Document**

9 Interoperability

This companion standard presents sets of parameters and alternatives from which subsets must be selected to implement particular telecontrol systems. Certain parameter values, such as the choice of "structured" or "unstructured" fields of the INFORMATION OBJECT ADDRESS of ASDUs represent mutually exclusive alternatives. This means that only one value of the defined parameters is admitted per system. Other parameters, such as the listed set of different process information in command and in monitor direction allow the specification of the complete set or subsets, as appropriate for given applications. This clause summarizes the parameters of the previous clauses to facilitate a suitable selection for a specific application. If a system is composed of equipment stemming from different manufacturers, it is necessary that all partners agree on the selected parameters.

The interoperability list is defined as in IEC 60870-5-101 and extended with parameters used in this standard. The text descriptions of parameters which are not applicable to this companion standard are strike-through (corresponding check box is marked black).

NOTE In addition, the full specification of a system may require individual selection of certain parameters for certain parts of the system, such as the individual selection of scaling factors for individually addressable measured values.

The selected parameters should be marked in the white boxes as follows:

☐ Function or ASDU is not used

☒ Function or ASDU is used as standardized (default)

☒ Function or ASDU is used in reverse mode

☒ Function or ASDU is used in standard and reverse mode

The possible selection (blank, X, R, or B) is specified for each specific clause or parameter.

A black check box indicates that the option cannot be selected in this companion standard.

9.1 System or device

(system-specific parameter, indicate definition of a system or a device by marking one of the following with "**X**")

☐ System definition

☐ Controlling station definition (Master)

☒ Controlled station definition (Slave)

9.2 Network configuration

(network-specific parameter, all configurations that are used are to be marked "**X**")

■ ~~Point-to-point~~ ■ ~~Multipoint-~~

■ ~~Multiple point-to-point~~ ■ ~~Multipoint-star~~

9.3 Physical layer

(network-specific parameter, all interfaces and data rates that are used are to be marked "**X**")

Transmission speed (control direction)

Unbalanced interchange Circuit V.24/V.28 Standard	Unbalanced interchange Circuit V.24/V.28 Recommended if >1 200 bit/s	Balanced interchange Circuit X.24/X.27	
~~100 bit/s~~	~~2 400 bit/s~~	~~2 400 bit/s~~	~~56 000 bit/s~~
~~200 bit/s~~	~~4 800 bit/s~~	~~4 800 bit/s~~	~~64 000 bit/s~~
~~300 bit/s~~	~~9 600 bit/s~~	~~9 600 bit/s~~	
~~600 bit/s~~		~~19 200 bit/s~~	
~~1 200 bit/s~~		~~38 400 bit/s~~	

Transmission speed (monitor direction)

Unbalanced interchange Circuit V.24/V.28 Standard	Unbalanced interchange Circuit V.24/V.28 Recommended if >1 200 bit/s	Balanced interchange Circuit X.24/X.27	
~~100 bit/s~~	~~2 400 bit/s~~	~~2 400 bit/s~~	~~56 000 bit/s~~
~~200 bit/s~~	~~4 800 bit/s~~	~~4 800 bit/s~~	~~64 000 bit/s~~
~~300 bit/s~~	~~9 600 bit/s~~	~~9 600 bit/s~~	
~~600 bit/s~~		~~19 200 bit/s~~	
~~1 200 bit/s~~		~~38 400 bit/s~~	

9.4 Link layer

(network-specific parameter, all options that are used are to be marked "X". Specify the maximum frame length. If a non-standard assignment of class 2 messages is implemented for unbalanced transmission, indicate the Type ID and COT of all messages assigned to class 2.)

~~Frame format FT 1.2, single character 1 and the fixed time out interval are used exclusively in this companion standard.~~

Link transmission

- ~~Balanced transmission~~
- ~~Unbalanced transmission~~

Frame length

- ~~Maximum length L (number of octets)~~

Address field of the link

- ~~not present (balanced transmission only)~~
- ~~One octet~~
- ~~Two octets~~
- ~~Structured~~
- ~~Unstructured~~

When using an unbalanced link layer, the following ASDU types are returned in class 2 messages (low priority) with the indicated causes of transmission:

■ ~~The standard assignment of ASDUs to class 2 messages is used as follows:~~

Type identification	Cause of transmission
9, 11, 13, 21	<1>

■ ~~A special assignment of ASDUs to class 2 messages is used as follows:~~

Type identification	Cause of transmission

~~Note: (In response to a class 2 poll, a controlled station may respond with class 1 data when there is no class 2 data available).~~

9.5 Application layer

Transmission mode for application data

Mode 1 (Least significant octet first), as defined in 4.10 of IEC 60870-5-4, is used exclusively in this companion standard.

Common address of ASDU
(system-specific parameter, all configurations that are used are to be marked "**X**")

■ ~~One octet~~ [X] Two octets

Information object address
(system-specific parameter, all configurations that are used are to be marked "**X**")

■ ~~One octet~~
■ ~~Two octets~~
[X] Three octets

[ ] Structured
[ ] Unstructured

Cause of transmission
(system-specific parameter, all configurations that are used are to be marked "**X**")

■ ~~One octet~~ [X] Two octets (with originator address). Originator address is set to zero if not used

Length of APDU
(system-specific parameter, specify the maximum length of the APDU per system)

The maximum length of the APDU is 253 (default). The maximum length may be reduced by the system.

[253] Maximum length of APDU per system

Selection of standard ASDUs

Process information in monitor direction
(station-specific parameter, mark each Type ID "**X**" if it is only used in the standard direction, "**R**" if only used in the reverse direction, and "**B**" if used in both directions).

[X]	<1> :=	Single-point information	M_SP_NA_1
[ ]	~~<2> :=~~	~~Single-point information with time tag~~	~~M_SP_TA_1~~
[X]	<3> :=	Double-point information	M_DP_NA_1
[ ]	~~<4> :=~~	~~Double-point information with time tag~~	~~M_DP_TA_1~~
[ ]	<5> :=	Step position information	M_ST_NA_1
[ ]	~~<6> :=~~	~~Step position information with time tag~~	~~M_ST_TA_1~~
[ ]	<7> :=	Bitstring of 32 bit	M_BO_NA_1
[ ]	~~<8> :=~~	~~Bitstring of 32 bit with time tag~~	~~M_BO_TA_1~~
[X]	<9> :=	Measured value, normalized value	M_ME_NA_1
[ ]	~~<10> :=~~	~~Measured value, normalized value with time tag~~	~~M_ME_TA_1~~
[X]	<11> :=	Measured value, scaled value	M_ME_NB_1
[ ]	~~<12> :=~~	~~Measured value, scaled value with time tag~~	~~M_ME_TB_1~~
[X]	<13> :=	Measured value, short floating point value	M_ME_NC_1
[ ]	~~<14> :=~~	~~Measured value, short floating point value with time tag~~	~~M_ME_TC_1~~
[X]	<15> :=	Integrated totals	M_IT_NA_1
[ ]	~~<16> :=~~	~~Integrated totals with time tag~~	~~M_IT_TA_1~~
[ ]	~~<17> :=~~	~~Event of protection equipment with time tag~~	~~M_EP_TA_1~~
[ ]	~~<18> :=~~	~~Packed start events of protection equipment with time tag~~	~~M_EP_TB_1~~
[ ]	~~<19> :=~~	~~Packed output circuit information of protection equipment with time tag~~	~~M_EP_TC_1~~
[ ]	<20> :=	Packed single-point information with status change detection	M_SP_NA_1
[ ]	<21> :=	Measured value, normalized value without quality descriptor	M_ME_ND_1

[X]	<30> :=	Single-point information with time tag CP56Time2a	M_SP_TB_1
[X]	<31> :=	Double-point information with time tag CP56Time2a	M_DP_TB_1
[ ]	<32> :=	Step position information with time tag CP56Time2a	M_ST_TB_1
[ ]	<33> :=	Bitstring of 32 bit with time tag CP56Time2a	M_BO_TB_1
[X]	<34> :=	Measured value, normalized value with time tag CP56Time2a	M_ME_TD_1
[X]	<35> :=	Measured value, scaled value with time tag CP56Time2a	M_ME_TE_1
[X]	<36> :=	Measured value, short floating point value with time tag CP56Time2a	M_ME_TF_1
[X]	<37> :=	Integrated totals with time tag CP56Time2a	M_IT_TB_1
[ ]	<38> :=	Event of protection equipment with time tag CP56Time2a	M_EP_TD_1
[ ]	<39> :=	Packed start events of protection equipment with time tag CP56Time2a	M_EP_TE_1
[ ]	<40> :=	Packed output circuit information of protection equipment with time tag CP56Time2a	M_EP_TF_1

Either the ASDUs of the set <2>, <4>, <6>, <8>, <10>, <12>, <14>, <16>, <17>, <18>, <19> or of the set <30> – <40> are used.

Process information in control direction
(station-specific parameter, mark each Type ID "**X**" if it is only used in the standard direction, "**R**" if only used in the reverse direction, and "**B**" if used in both directions).

[X]	<45> := Single command	C_SC_NA_1
[X]	<46> := Double command	C_DC_NA_1
[ ]	<47> := Regulating step command	C_RC_NA_1
[X]	<48> := Set point command, normalized value	C_SE_NA_1
[X]	<49> := Set point command, scaled value	C_SE_NB_1
[X]	<50> := Set point command, short floating point value	C_SE_NC_1
[ ]	<51> := Bitstring of 32 bit	C_BO_NA_1
[X]	<58> := Single command with time tag CP56Time2a	C_SC_TA_1
[X]	<59> := Double command with time tag CP56Time2a	C_DC_TA_1
[ ]	<60> := Regulating step command with time tag CP56Time2a	C_RC_TA_1
[X]	<61> := Set point command, normalized value with time tag CP56Time2a	C_SE_TA_1
[X]	<62> := Set point command, scaled value with time tag CP56Time2a	C_SE_TB_1
[X]	<63> := Set point command, short floating point value with time tag CP56Time2a	C_SE_TC_1
[ ]	<64> := Bitstring of 32 bit with time tag CP56Time2a	C_BO_TA_1

Either the ASDUs of the set <45> – <51> or of the set <58> – <64> are used.

System information in monitor direction
(station-specific parameter, mark "**X**" if used)

[X]	<70> := End of initialization	M_EI_NA_1

System information in control direction
(station-specific parameter, mark each Type ID "**X**" if it is only used in the standard direction, "**R**" if only used in the reverse direction, and "**B**" if used in both directions).

[X]	<100>:= Interrogation command	C_IC_NA_1
[X]	<101>:= Counter interrogation command	C_CI_NA_1
[X]	<102>:= Read command	C_RD_NA_1
[X]	<103>:= Clock synchronization command (option see 7.6)	C_CS_NA_1
[■]	~~<104>:= Test command~~	~~C_TS_NA_1~~
[ ]	<105>:= Reset process command	C_RP_NA_1
[■]	~~<106>:= Delay acquisition command~~	~~C_CD_NA_1~~
[X]	<107>:= Test command with time tag CP56Time2a	C_TS_TA_1

Parameter in control direction
(station-specific parameter, mark each Type ID "**X**" if it is only used in the standard direction, "**R**" if only used in the reverse direction, and "**B**" if used in both directions).

[X]	<110>:= Parameter of measured value, normalized value	P_ME_NA_1
[X]	<111>:= Parameter of measured value, scaled value	P_ME_NB_1
[X]	<112>:= Parameter of measured value, short floating point value	P_ME_NC_1
[X]	<113>:= Parameter activation	P_AC_NA_1

File transfer
(station-specific parameter, mark each Type ID "**X**" if it is only used in the standard direction, "**R**" if only used in the reverse direction, and "**B**" if used in both directions).

[ ]	<120>:= File ready	F_FR_NA_1
[ ]	<121>:= Section ready	F_SR_NA_1
[ ]	<122>:= Call directory, select file, call file, call section	F_SC_NA_1
[ ]	<123>:= Last section, last segment	F_LS_NA_1
[ ]	<124>:= Ack file, ack section	F_AF_NA_1
[ ]	<125>:= Segment	F_SG_NA_1
[ ]	<126>:= Directory {blank or X, only available in monitor (standard) direction}	F_DR_TA_1

Type identifier and cause of transmission assignments
(station-specific parameters)

Shaded boxes: option not required.

Black boxes: option not permitted in this companion standard

Blank: functions or ASDU not used.

Mark Type Identification/Cause of transmission combinations:

"**X**" if only used in the standard direction;

"**R**" if only used in the reverse direction;

"**B**" if used in both directions.

Type identification		**Cause of transmission**																	
		1	2	3	4	5	6	7	8	9	10	11	12	13	20 to 36	37 to 41	44	45	46	47
<1>	M_SP_NA_1		X	X		X									X					
~~<2>~~	~~M_SP_TA_1~~																			
<3>	M_DP_NA_1		X	X		X									X					
~~<4>~~	~~M_DP_TA_1~~																			
<5>	M_ST_NA_1																			
~~<6>~~	~~M_ST_TA_1~~																			
<7>	M_BO_NA_1																			
~~<8>~~	~~M_BO_TA_1~~																			
<9>	M_ME_NA_1	X	X	X		X									X					
~~<10>~~	~~M_ME_TA_1~~																			
<11>	M_ME_NB_1	X	X	X		X									X					
~~<12>~~	~~M_ME_TB_1~~																			
<13>	M_ME_NC_1	X	X	X		X									X					
~~<14>~~	~~M_ME_TC_1~~																			
<15>	M_IT_NA_1			X												X				
~~<16>~~	~~M_IT_TA_1~~																			
~~<17>~~	~~M_EP_TA_1~~																			
~~<18>~~	~~M_EP_TB_1~~																			
~~<19>~~	~~M_EP_TC_1~~																			
<20>	M_PS_NA_1																			
<21>	M_ME_ND_1																			
<30>	M_SP_TB_1			X		X														
<31>	M_DP_TB_1			X		X														
<32>	M_ST_TB_1																			
<33>	M_BO_TB_1																			
<34>	M_ME_TD_1			X		X														
<35>	M_ME_TE_1			X		X														
<36>	M_ME_TF_1			X		X														
<37>	M_IT_TB_1			X																
<38>	M_EP_TD_1																			
<39>	M_EP_TE_1																			
<40>	M_EP_TF_1																			
<45>	C_SC_NA_1						X	X	X	X	X						X	X	X	X
<46>	C_DC_NA_1						X	X	X	X	X						X	X	X	X
<47>	C_RC_NA_1																			
<48>	C_SE_NA_1						X	X	X	X	X						X	X	X	X
<49>	C_SE_NB_1						X	X	X	X	X						X	X	X	X

Type identification		Cause of transmission																		
		1	2	3	4	5	6	7	8	9	10	11	12	13	20 to 36	37 to 41	44	45	46	47
<50>	C_SE_NC_1						X	X	X	X	X						X	X	X	X
<51>	C_BO_NA_1																			
<58>	C_SC_TA_1						X	X	X	X	X						X	X	X	X
<59>	C_DC_TA_1						X	X	X	X	X						X	X	X	X
<60>	C_RC_TA_1																			
<61>	C_SE_TA_1						X	X	X	X	X						X	X	X	X
<62>	C_SE_TB_1						X	X	X	X	X						X	X	X	X
<63>	C_SE_TC_1						X	X	X	X	X						X	X	X	X
<64>	C_BO_TA_1																			
<70>	M_EI_NA_1*																			
<100>	C_IC_NA_1						X	X	X	X	X						X	X	X	X
<101>	C_CI_NA_1						X	X			X						X	X	X	X
<102>	C_RD_NA_1					X														
<103>	C_CS_NA_1						X	X									X	X	X	X
~~<104>~~	~~C_TS_NA_1~~																			
<105>	C_RP_NA_1																			
~~<106>~~	~~C_CD_NA_1~~																			
<107>	C_TS_TA_1						X	X									X	X	X	X
<110>	P_ME_NA_1						X	X									X	X	X	X
<111>	P_ME_NB_1						X	X									X	X	X	X
<112>	P_ME_NC_1						X	X									X	X	X	X
<113>	P_AC_NA_1						X	X	X	X							X	X	X	X
<120>	F_FR_NA_1																			
<121>	F_SR_NA_1																			
<122>	F_SC_NA_1																			
<123>	F_LS_NA_1																			
<124>	F_AF_NA_1																			
<125>	F_SG_NA_1																			
<126>	F_DR_TA_1*																			

* Blank or X only

9.6 Basic application functions

Station initialization
(station-specific parameter, mark "**X**" if function is used)

[X] Remote initialization

Cyclic data transmission
(station-specific parameter, mark "**X**" if function is only used in the standard direction, "**R**" if only used in the reverse direction, and "**B**" if used in both directions)

[X] Cyclic data transmission

Read procedure
(station-specific parameter, mark "**X**" if function is only used in the standard direction, "**R**" if only used in the reverse direction, and "**B**" if used in both directions)

[X] Read procedure

Spontaneous transmission
(station-specific parameter, mark "**X**" if function is only used in the standard direction, "**R**" if only used in the reverse direction, and "**B**" if used in both directions)

[X] Spontaneous transmission

Double transmission of information objects with cause of transmission spontaneous
(station-specific parameter, mark each information type "**X**" where both a Type ID without time and corresponding Type ID with time are issued in response to a single spontaneous change of a monitored object)

The following type identifications may be transmitted in succession caused by a single status change of an information object. The particular information object addresses for which double transmission is enabled are defined in a project-specific list.

- [X] Single-point information M_SP_NA_1, M_SP_TA_1, M_SP_TB_1 and M_PS_NA_1
- [X] Double-point information M_DP_NA_1, M_DP_TA_1 and M_DP_TB_1
- [ ] Step position information M_ST_NA_1, M_ST_TA_1 and M_ST_TB_1
- [ ] Bitstring of 32 bit M_BO_NA_1, M_BO_TA_1 and M_BO_TB_1 (if defined for a specific project)
- [X] Measured value, normalized value M_ME_NA_1, M_ME_TA_1, M_ME_ND_1 and M_ME_TD_1
- [X] Measured value, scaled value M_ME_NB_1, M_ME_TB_1 and M_ME_TE_1
- [X] Measured value, short floating point number M_ME_NC_1, M_ME_TC_1 and M_ME_TF_1

Station interrogation
(station-specific parameter, mark "**X**" if function is only used in the standard direction, "**R**" if only used in the reverse direction, and "**B**" if used in both directions).

[X] global
[X] group 1
[X] group 2
[X] group 3
[X] group 4
[X] group 5
[X] group 6
[X] group 7
[X] group 8
[X] group 9
[X] group 10
[X] group 11
[X] group 12
[X] group 13
[X] group 14
[X] group 15
[X] group 16

Information object addresses assigned to each group must be shown in a separate table.

Clock synchronization
(station-specific parameter, mark "**X**" if function is only used in the standard direction, "**R**" if only used in the reverse direction, and "**B**" if used in both directions).

[X] Clock synchronization

optional, see 7.6

Command transmission
(object-specific parameter, mark "**X**" if function is only used in the standard direction, "**R**" if only used in the reverse direction, and "**B**" if used in both directions).

[X] Direct command transmission
[X] Direct set point command transmission
[X] Select and execute command
[X] Select and execute set point command
[X] C_SE ACTTERM used

[X] No additional definition
[X] Short-pulse duration (duration determined by a system parameter in the outstation)
[X] Long-pulse duration (duration determined by a system parameter in the outstation)
[X] Persistent output

[X] Supervision of maximum delay in command direction of commands and set point commands

[Parameter] Maximum allowable delay of commands and set point commands

Transmission of integrated totals
(station- or object-specific parameter, mark "**X**" if function is only used in the standard direction, "**R**" if only used in the reverse direction, and "**B**" if used in both directions).

- [ ] Mode A: Local freeze with spontaneous transmission
- [ ] Mode B: Local freeze with counter interrogation
- [X] Mode C: Freeze and transmit by counter-interrogation commands
- [ ] Mode D: Freeze by counter-interrogation command, frozen values reported spontaneously

- [X] Counter read
- [X] Counter freeze without reset
- [X] Counter freeze with reset
- [X] Counter reset

- [X] General request counter
- [X] Request counter group 1
- [X] Request counter group 2
- [X] Request counter group 3
- [X] Request counter group 4

Parameter loading
(object-specific parameter, mark "**X**" if function is only used in the standard direction, "**R**" if only used in the reverse direction, and "**B**" if used in both directions).

- [X] Threshold value
- [X] Smoothing factor
- [X] Low limit for transmission of measured values
- [X] High limit for transmission of measured values

Parameter activation
(object-specific parameter, mark "**X**" if function is only used in the standard direction, "**R**" if only used in the reverse direction, and "**B**" if used in both directions).

- [X] Act/deact of persistent cyclic or periodic transmission of the addressed object

Test procedure
(station-specific parameter, mark "**X**" if function is only used in the standard direction, "**R**" if only used in the reverse direction, and "**B**" if used in both directions).

- [X] Test procedure

File transfer
(station-specific parameter, mark "**X**" if function is used).

File transfer in monitor direction

☐ Transparent file

☐ Transmission of disturbance data of protection equipment

☐ Transmission of sequences of events

☐ Transmission of sequences of recorded analogue values

File transfer in control direction

☐ Transparent file

Background scan
(station-specific parameter, mark "**X**" if function is only used in the standard direction, "**R**" if only used in the reverse direction, and "**B**" if used in both directions).

☒ Background scan

Acquisition of transmission delay
(station-specific parameter, mark "**X**" if function is only used in the standard direction, "**R**" if only used in the reverse direction, and "**B**" if used in both directions).

■ ~~Acquisition of transmission delay~~

Definition of time outs

Parameter	Default value	Remarks	Selected value
t_0	30 s	Time-out of connection establishment	parameter
t_1	15 s	Time-out of send or test APDUs	parameter
t_2	10 s	Time-out for acknowledges in case of no data messages $t_2 < t_1$	parameter
t_3	20 s	Time-out for sending test frames in case of a long idle state	parameter

Maximum range of values for all time-outs: 1 to 255 s, accuracy 1 s.

Maximum number of outstanding I format APDUs *k* and latest acknowledge APDUs (*w*)

Parameter	Default value	Remarks	Selected value
k	12 APDUs	Maximum difference receive sequence number to send state variable	parameter
w	8 APDUs	Latest acknowledge after receiving *w* I format APDUs	parameter

Maximum range of values *k*: 1 to 32767 (2^{15}–1) APDUs, accuracy 1 APDU

Maximum range of values *w*: 1 to 32767 APDUs, accuracy 1 APDU (Recommendation: *w* should not exceed two-thirds of *k*).

Portnumber

Parameter	Value	Remarks
Portnumber	2404	In all cases

RFC 2200 suite

RFC 2200 is an official Internet Standard which describes the state of standardization of protocols used in the Internet as determined by the Internet Architecture Board (IAB). It offers a broad spectrum of actual standards used in the Internet. The suitable selection of documents from RFC 2200 defined in this standard for given projects has to be chosen by the user of this standard.

[X] Ethernet 802.3

[ ] Serial X.21 interface

[X] Other selection from RFC 2200:

List of valid documents from RFC 2200

1. TLS

2. .....

3. .....

4. .....

5. .....

6. .....

7. etc.

_____

11 Protocol implementation conformance statement

11.1 Overview of clause

Implementors of this specification shall supply the information in Clause 11 on request. An “X” in a box means that the implementation supports the listed feature.

11.2 Required algorithms

If the implementor does not declare support for an algorithm marked “(required)”, interoperability cannot be guaranteed.

If an algorithm is not supported due to export restrictions, the implementor shall provide a copy of the export restriction that prohibits its export. This algorithm shall not be supported if and only if export restrictions do not allow any mechanism of exportation. If this algorithm is not supported, the implementation shall be clearly documented as adhering to the export restrictions, as supplied. The documentation shall also specify that the interoperable/base specification requirements are not supported. Samples of the documentation shall be provided.

11.3 MAC algorithms

[X] HMAC-SHA-256 (required)

[ ] Other _____

11.4 Key wrap algorithms

[X] AES-256 Key Wrap (required)

[ ] Other AES-128 _____

11.5 Maximum Error messages sent

[ ] Fixed at 2

[X] Configurable

11.6 Use of Error messages

[X] Transmits Error messages Configurable

11.7 Update Key Change Methods

[ ] None permitted

[x] <4> Symmetric AES-256 / HMAC-SHA-256 (required)

[ ] <5> Symmetric AES-256 / AES-GMAC

[ ] <68> Asymmetric
RSA-2048 / DSA SHA-256 (L=2048 N=256) / HMAC-SHA-256

[ ] <69> Asymmetric
RSA-3072 / DSA SHA-256 (L=3072 N=256) / AES-SHA-256

[ ] <70> Asymmetric
RSA-2048 / DSA SHA-256 (L=2048 N=256) / HMAC-GMAC

[ ] <71> Asymmetric
RSA-3072 / DSA SHA-256 (L=3072 N=256) / AES-GMAC

[ ] Other _____

11.8 User Status Change

[x] Non-certificate method (required)

[ ] Use IEC/TS 62351-8 Certificates

11.9

x Challenge response Supported

x Aggressive Mode Supported

11.10

x Only one user with Configurable ID is supported