IE104 Communication Driver

Driver for Ethernet Communication
With IEC 60870-5-104 protocol

Contents

Introduction

This document will help you to select, configure and execute the IE104 driver, and it is organized as follows:

- **Introduction**: This section, which provides an overview of the document.
- **General Information**: Identifies all of the hardware and software components required to implement communication between the Studio system and the target device.
- **Selecting the Driver**: Explains how to select the IE104 driver in the Studio system.
- **Configuring the Driver**: Explains how to configure the IE104 driver in the Studio system, including how to associate database tags with device registers.
- **Executing the Driver**: Explains how to execute the IE104 driver during application runtime.
- **Troubleshooting**: Lists the most common errors for this driver, their probable causes, and basic procedures to resolve them.
- **Sample Application**: Explains how to use a sample application to test the IE104 driver configuration
- **Revision History**: Provides a log of all changes made to the driver and this documentation.

✍ **Notes:**

- This document assumes that you have read the “Development Environment” chapter in Studio’s *Technical Reference Manual*.
- This document also assumes that you are familiar with the Microsoft Windows 7/XP/Vista environment. If you are not familiar with Windows, then we suggest using the **Help** feature (available from the Windows desktop **Start** menu) as you work through this guide.
- This document uses concepts of the IEC 60870-5-104 protocol, and assumes familiarity with its usage.

General Information

This chapter identifies all of the hardware and software components required to implement communication between the IE104 driver in Studio and remote devices.

The information is organized into the following sections:

- Device Specifications
- Network Specifications
- Driver Characteristics
- Conformance Testing

Device Specifications

You can use this driver to communicate with any device using the IEC 60870-5-104 protocol. (The devices used for conformance testing are listed on the next page.)

Network Specifications

To establish communication, your device network must meet the following specifications:

- **Device Communication Port**: Ethernet
- **Physical Protocol**: TCP/IP
- **Logic Protocol**: IEC 60870-5-104
- **Device Runtime Software**: None
- **Specific PC Board**: None
- **Adapters/Converters**: None
- **Cable Wiring Scheme**: None

Driver Characteristics

The IE104 driver package consists of the following files, which are automatically installed in the **`\DRV`** subdirectory of Studio:

- **`IE104.INI:`** Internal driver file. *You must not modify this file*.
- **`IE104.MSG:`** Internal driver file containing error messages for each error code. *You must not modify this file*.
- **`IE104.PDF:`** This document, which provides detailed information about the IE104 driver.
- **`IE104.DLL:`** Compiled driver.

You can use the IE104 driver on the following operating systems:

- Windows 7/XP/Vista
- Windows CE 4.x, 5.x, 6.x

For a description of the operating systems used to test driver conformance, see "Conformance Testing" below.

Conformance Testing

The following hardware/software was used for conformance testing:

For Ethernet Tests

- **TCP/IP Port**: 2404
- **Protocol**: IEC 60870-5-104
- **Cable**: Ethernet Cable

Driver Version	Studio Version	Operating System	Equipment
1.5	8.1.1	Windows 7/8	Traingle Microwork's Protocol Test Harness – IEC 104 Slave simulator

The IE104 driver supports the following services:

- **Station Initialization**
- **Cyclic data transmission**
- **Acquisition of Events**
- **Transmission of Integrated Totals**
- **General Interrogation**
- **Command transmission**

These services are supported through 3 basic operations:

1. **Interrogation of Groups and Counters**
2. **Monitoring of points (unsolicited messages received)**
3. **Select Before Operate (SBO): Writing to individual points (objects) on the device**

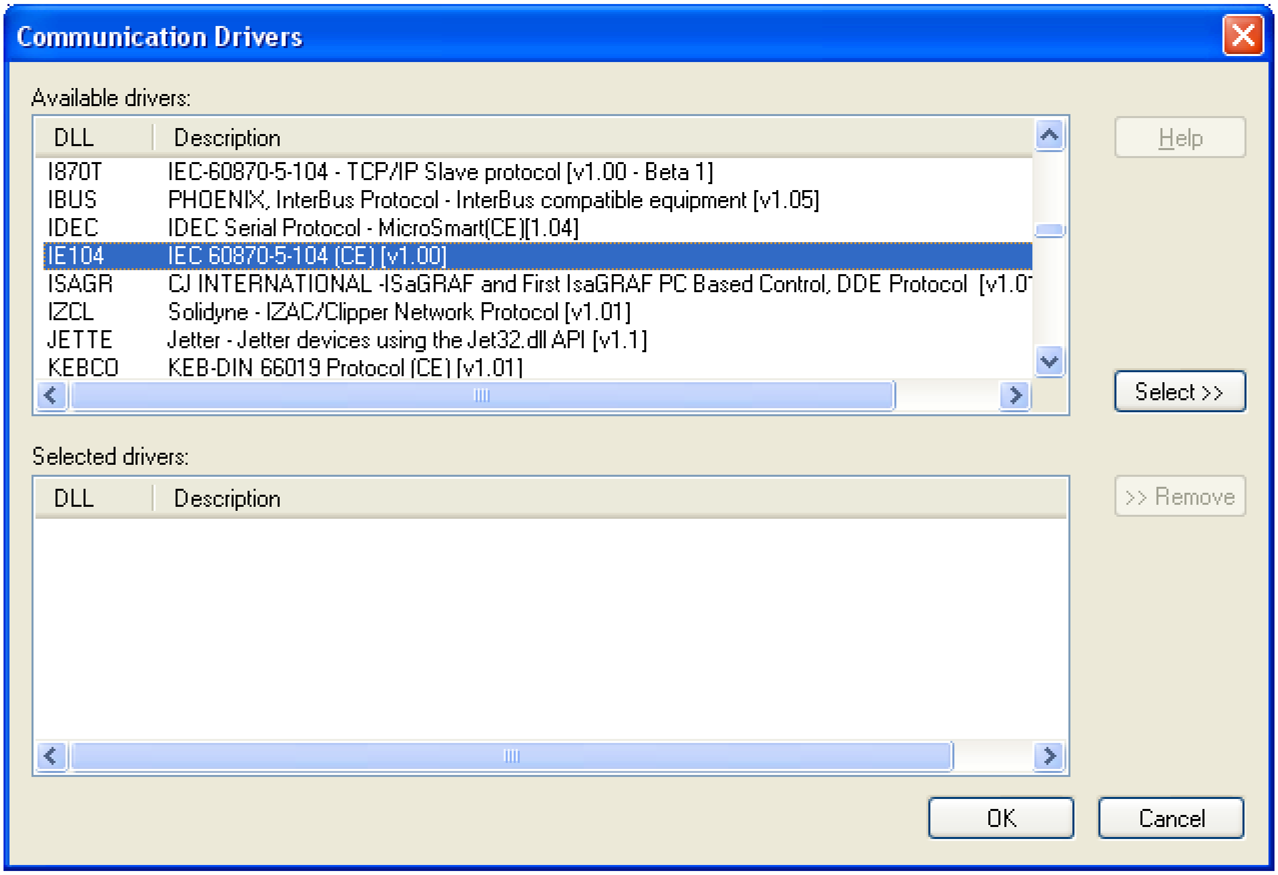
The remainder of this document will help you to configure the driver sheets to access these services.

On the end of the document, the Interoperability Form shows the supported types.

Selecting the Driver

When you install Studio, all of the communication drivers are automatically installed in the **`\DRV`** subdirectory but they remain dormant until manually selected for specific applications. To select the IE104 driver for your Studio application:

1. From the main menu bar, select **Insert** → **Driver** to open the *Communication Drivers* dialog.
2. Select the **IE104** driver from the *Available Drivers* list, and then click the **Select** button.



DLL	Description
I870T	IEC-60870-5-104 - TCP/IP Slave protocol [v1.00 - Beta 1]
IBUS	PHOENIX, InterBus Protocol - InterBus compatible equipment [v1.05]
IDEC	IDEC Serial Protocol - MicroSmart(CE)[1.04]
IE104	IEC 60870-5-104 (CE) [v1.00]
ISAGR	CJ INTERNATIONAL -ISaGRAF and First IsaGRAF PC Based Control, DDE Protocol [v1.0
IZCL	Solidyne - IZAC/Clipper Network Protocol [v1.01]
JETTE	Jetter - Jetter devices using the Jet32.dll API [v1.1]
KEBCO	KEB-DIN 66019 Protocol (CE) [v1.01]

DLL	Description

Communication Drivers Dialog

3. When the **IE104** driver is displayed in the **Selected Drivers** list, click the **OK** button to close the dialog. The driver is added to the *Drivers* folder, in the *Comm* tab of the Workspace.

➲ **Attention:**
For safety reasons, you must take special precautions when installing any physical hardware. Please consult the manufacturer's documentation for specific instructions.

Configuring the Driver

Once you have selected the IE104 driver in Studio, you must properly configure it to communicate with your target device.

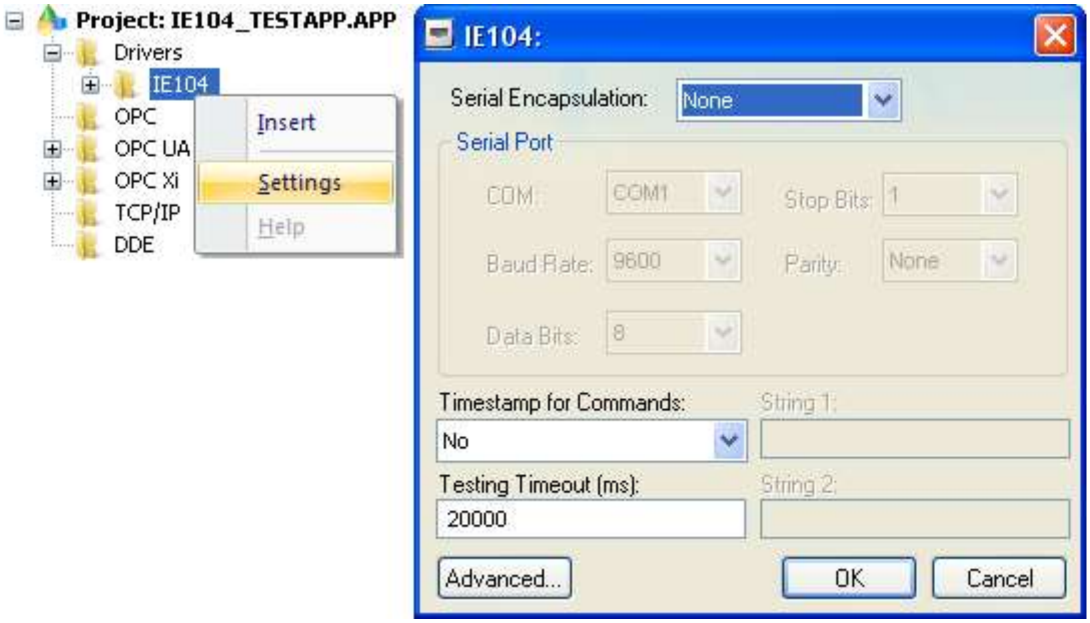
Configuring the Communication Settings

The communication settings are described in detail in the "Communication" chapter of the Studio *Technical Reference Manual*, and the same general procedures are used for all drivers. Please review those procedures before continuing.

For the purposes of this document, only IE104 driver-specific settings and procedures will be discussed here. To configure the communication settings for the IE104 driver:

1. In the *Workspace* pane, select the *Comm* tab and then expand the *Drivers* folder. The IE104 driver is listed here as a subfolder.

2. Right-click on the *IE104* subfolder and then select the **Settings** option from the pop-up menu:



Serial Encapsulation: None
COM: COM1
Stop Bits: 1
Baud Rate: 9600
Parity: None
Data Bits: 8
Timestamp for Commands: No
String 1:
Testing Timeout (ms): 20000
String 2:

IE104: Communication Settings Dialog

3. In the *Communication Settings* dialog, configure the driver settings to enable communication with your target device. To ensure error-free communication, the driver settings must `exactly match` the corresponding settings on the device. Please consult the manufacturer's documentation for instructions how to configure the device and for complete descriptions of the settings.

   Depending on your circumstances, you may need to configure the driver `before` you have configured your target device. If this is the case, then take note of the driver settings and have them ready when you later configure the device.

   > ➲ **Attention:**
   > For safety reasons, you **must** take special precautions when connecting and configuring new

equipment. Please consult the manufacturer's documentation for specific instructions.

The communication settings and their possible values are described in the following table:

Parameters	Default Values	Valid Values	Description
Timestamp for Commands	**`No`**	**`Yes` or `No`**	This setting affect the commands used for commands sent to change values on the device: • **No**: the driver will not send the current timestamp on the message. This corresponds to the type identification 45 – 51 of the IEC 60870-5-104 protocol. • **Yes**: the driver will send the current timestamp on the message. This corresponds to the type identification 58 – 64 of the IEC 60870-5-104 protocol.
Testing Timeout (ms)	**`20000`**	**`Integer`**	This setting affect the time that must elapse before the driver sends a test frame, to check if connection is alive. This is referred on IEC 60870-5-104 as T_3.

Configuring the Driver Worksheets

This driver currently does not support Main Driver Sheet. Standard Driver Worksheets must be inserted to define tag/register associations to be monitored, that are triggered by specific application behaviors.

The configuration of these worksheets is described in detail in the "Communication" chapter of the Studio *Technical Reference Manual*, and the same general procedures are used for all drivers. Please review those procedures before continuing.

For the purposes of this document, only IE104 driver-specific parameters and procedures will be discussed here.

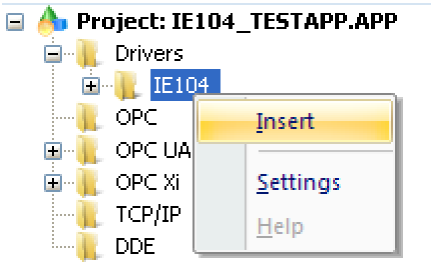
STANDARD DRIVER WORKSHEET

When you select the IE104 driver and add it to your application, it does not have any Driver Sheet added. To start communicating, you must insert Standard Driver Worksheets to define the tags/registers to be monitored and commands to be written. These services are specified by the header used on the driver sheet and the addresses.

The configuration of these worksheets is described in detail in the "Communication" chapter of the Studio *Technical Reference Manual*, and the same general procedures are used for all drivers. Please review those procedures before continuing.

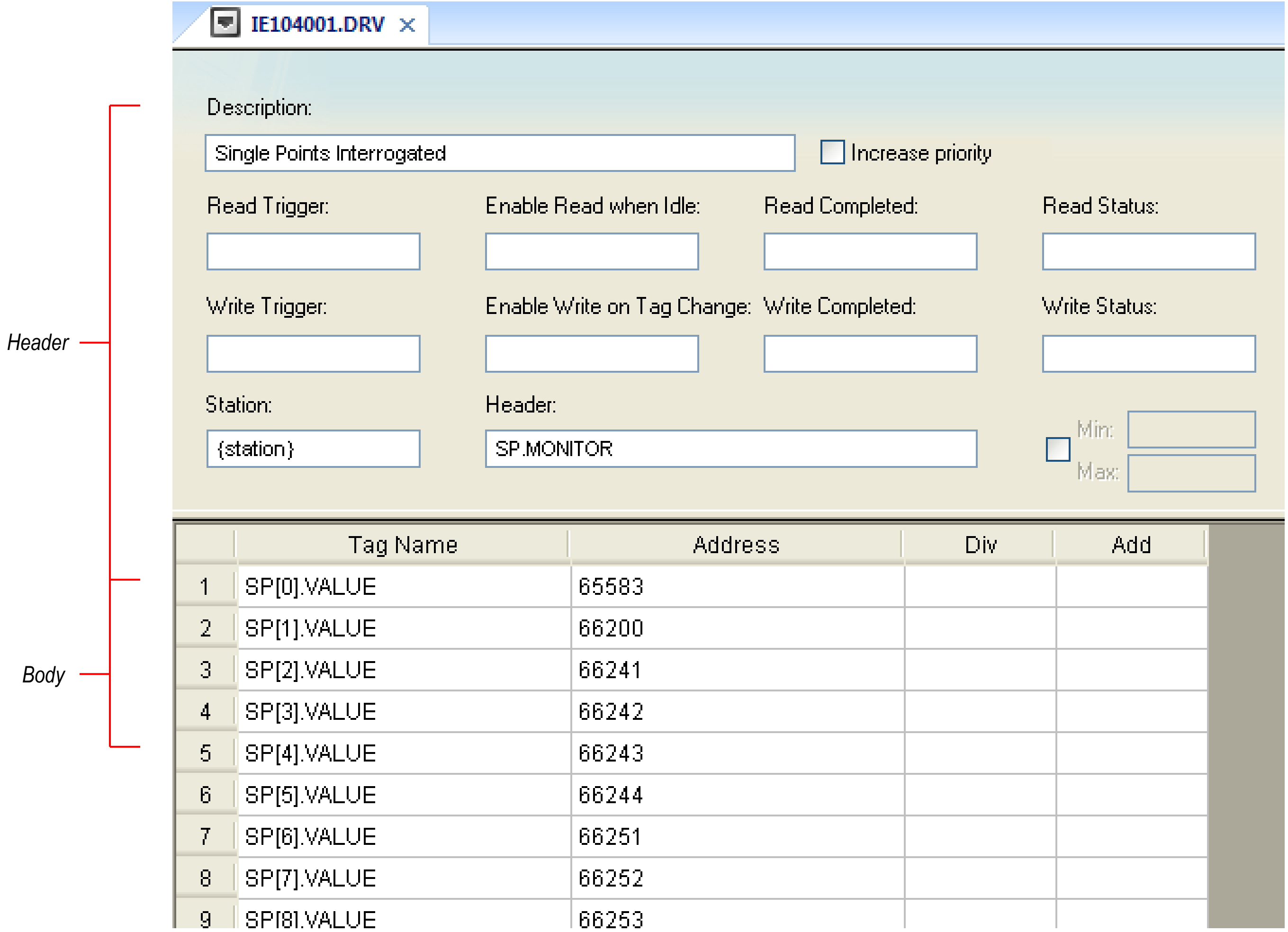
To insert a new driver worksheet:

1. In the *Comm* tab, open the *Drivers* folder and locate the *IE104* subfolder.
2. Right-click on the `IE104` subfolder, and then select **Insert** from the pop-up menu:



Inserting a New Worksheet

A new *IE104* driver worksheet is inserted into the *IE104* subfolder, and the worksheet is opened for configuration:



Description: Single Points Interrogated

□ Increase priority

Read Trigger:

Enable Read when Idle:

Read Completed:

Read Status:

Write Trigger:

Enable Write on Tag Change:

Write Completed:

Write Status:

Station: {station}

Header: SP.MONITOR

□ Min: Max:

	Tag Name	Address	Div	Add
1	SP[0].VALUE	65583		
2	SP[1].VALUE	66200		
3	SP[2].VALUE	66241		
4	SP[3].VALUE	66242		
5	SP[4].VALUE	66243		
6	SP[5].VALUE	66244		
7	SP[6].VALUE	66251		
8	SP[7].VALUE	66252		
9	SP[8].VALUE	66253		

IE104 Driver Worksheet

> **Note:**
> Worksheets are numbered in order of creation, so the first worksheet is **`IE104001.drv`**.

Most of the fields on this worksheet are standard for all drivers; see the "Communication" chapter of the *Technical Reference Manual* for more information on configuring these fields. However **Header** and **Body** (as noted on the above picture) fields use syntax that is specific to the IE104 driver.

3. Configure the **Header** fields as follows:

▪ **Station** field

- Use the following syntax:

  `<Common Address>:<IP Address>:<Opt Port Number>`

Example — **`1:192.168.0.52`** or **`1:192.168.0.52:2404`**

Where:

– `<Common Address>` is the address configured for the outstation in the range 0 to 65535.

– `<IP Address>` is the IP address of the device on the Ethernet network.

– `<Opt Port Number>` is the optional TCP/IP Port number. If you do not configure this parameter, the default value of **2404** will be used

You can also specify an indirect tag (e.g. **`{station}`**), but the tag that is referenced must follow the same syntax and contain a valid value.

➲ **Attention:**

- You cannot leave the Station field **blank**
- Even though the station field might be specified by a tag (e.g. {station}), the tag value must be specified before starting the driver task, for example on a startup script.

✍ **Notes**:

▪ The station field is used to establish communication with various outstations on an Ethernet network. These connections are opened on driver initialization, and errors on it are reported on driver status tags (i.e. configured on header STATUS) and on messages printed on Logwin. The driver will attempt to keep the connections open at all moments during driver execution.

▪ **Header** field: Provides access to the services supported on the driver. The header specification is explained depending on the tasks to be performed:

1. Interrogation of groups or counters groups

Syntax: `CMD:<ConfirmationTimeout>`

Where:

- **Optional:** `<ConfirmationTimeout>` is the time to wait for an activation confirmation command. Set this field to 0 to ignore or not expect a confirmation. If this field is not set, the default value is the T_1 timeout.

ADDRESS SPECIFICATION:

Syntax: `<Choice>:<Number>`

Where:

- `<Choice>` identifies the type of interrogation:

Choice	Meaning
G	Group
C	Counter
CLKS	Clock Syncronization

- `<Number>` identifies the number of the group. If set to `0`, or not present, means a general interrogation. The accepted values for this parameter are 0 to 16. However, for counter interrogation, values above 4 raise an error when used.

USAGE:

On a **Write Trigger** to a group with this header the driver will send the messages for interrogation for each row of the driver sheet. This header does not support **Read Triggers**. The expected usage is writing to each tag configured on the driver sheet separately, enabling **Write on Tag Change**.

For a Clock Syncronization (e.g. CLKS) the parameter <Number> and the tag value is ignored. The Clock Syncronization command is used to synchronize the device time with the client time.

For a Group Interrogation (e.g. `G:10`, for interrogating group number 10), the tag value is ignored. However for Counter Interrogation, the tag value defines the behavior set on the command:

Tag Value	Meaning
Other	Nothing, ignored
0	Read
1	Freeze
2	Freeze and Reset
3	Reset

EXAMPLES:

- Header: CMD:0 → Will not wait for confirmations
- Addresses:
  - G or G:0 → General interrogation
  - G:1 → Interrogation of group number 1
  - C or C:0 → General Counter interrogation
  - C:2 → Interrogation of counter group number 2
  - CLKS -> Clock Syncronication Command

Notes:

- The user must pay attention to the fact that the interrogations are triggered ONLY by its request, using a *Write Trigger* or changing the value of the associated tag and enabling *Write on Tag Change*.
- Also, the use of the tag value for Counter Interrogation requires special attention to obtain the correct information. The usage of tag values other than those on range 0 to 3 are ignored by the driver, and return

a specific error code when used.

- The messages sent by the outstation to the driver are considered monitored values, and are handled by the headers designed for monitoring.

2. Monitoring/Operating

Syntax: `<Type>.<Operation>:<Offset>.<Qualifier>:<ConfirmationTimeout>`

Where:

- `<Type>` specifies which type will be monitored or operated. The following table indicates available types. Blank cells indicate unavailability and refer to invalid operations.

Type	Meaning	Monitor	Select/Execute/Break	Qualifier	Value Range
SP	**Single Point**	•	•	•	**[0,1]**
DP	**Double Point**	•	•	•	**[0,3]**
ST	**Step Position**	•	•	•	**[-64,+63]**
BS	**Bitstring 32-bit**	•	**Not Meaningful**	**Not Meaningful**	**$[0,2^{32}-1]$**
NV	**Normalized Value**	•	•	**Not Meaningful**	**$[-1,1-2^{-15}]$**
SV	**Scaled Value**	•	•	**Not Meaningful**	**$[-2^{15},2^{15}-1]$**
FV	**Floating-Point Value**	•	•	**Not Meaningful**	**IEEE-754 32-bit floats**
IT	**Integrated Totals**	•			**$[-2^{31},2^{31}-1]$**
EI	**End of Initialization**	•			**For address 0: [0,127]** **For address 1: [0,1]**

- `<Operation>` determines to which operation this header is associated:

  1. `MONITOR`: Monitoring purposes.
  2. `SELECT`: Send a selection command to device
  3. `EXECUTE`: Send the command to execute the operation
  4. `BREAK`: Send a command of deactivation of selection

- **Optional:** `<Offset>` determines the initial offset for the addresses
- **Optional:** `<Qualifier>` is useful only for `SELECT/EXECUTE/BREAK` commands of types `SP`, `DP` and `ST`, and indicate a special code to send along the command, on its qualifier. Leaving it out indicates that no additional definition is required (code 0):

  1. `DEFAULT`: default option, no additional definition is sent
  2. `SPULSE`: short pulse
  3. `LPULSE`: long pulse
  4. `PERSISTENT`: persistent

  For further information on how the qualifier is handled, consult your device's documentation.

- **Optional:** `<ConfirmationTimeout>` is the time to wait for an activation confirmation command. Set this field to 0 to ignore or not expect a confirmation. If this field is not set, the default value is the T1 timeout.

MONITORING

For monitoring, the parameter `<Operation>` of the header must be set to `MONITOR`.

All messages received by the driver (Spontaneously or Interrogated) will be directed to the tag associated with the header/address, setting the timestamp properly. Each time a message is received, the data that it contains is directed to the proper tag. Tags for monitoring purposes have its values changed if and only if a message is received with its matching address and type. The quality of a tag set for monitoring is changed to bad when the connection to the outstation is lost.

Groups configured for monitoring support neither Read nor Write Triggers, as values are automatically set on the associated tags as soon as messages are received.

SELECT BEFORE OPERATE (SBO)

SBO operations are performed on two separate commands, a select followed by execute, with a third optional command to cancel a selection.

The `<Operation>` parameter identifies whether the driver will send a `SELECT`, `EXECUTE` or `BREAK` (cancel) command.

The `SELECT`, `BREAK` and `EXECUTE` operations require a confirmation from the outstation. The lack of such confirmation is identified as an error for the user. The amount of time that the driver waits for this confirmation is configured on the header, and allows the confirmation to be ignored.

Once the `EXECUTE` might be sent directly, and it is responsibility of both the user and device to validate that the value sent on `EXECUTE` command is the same as sent on `SELECT`, and if it accepts direct executions, or multiple executions simultaneously.

Groups configured for `SELECT/EXECUTE/BREAK` do not support **Read Triggers**. **Write Triggers** are handled per item, row by row of a driver sheet, or for a single row if **Write on Tag Change** is enabled, once the protocol does not support multiple addresses on the same command.

➲ **Attention:**

All types that support commands will read and write the same values types and range (e.g.: 0 or 1 for single points), except for the regulating step commands. The step information is read as integer values, indicating the current position on the device. However, the command indicates a step up (+1) or down (-1).

For that matter, the value on the tag is interpreted as:

Value	Meaning
Other	Nothing, ignored
-1	Previous Step
+1	Next Step

For example, if a user wants to regulate a step from position 12 to 14, the tag associated with the address must write two commands with value +1 on the tag. Using a **Write On Tag Change** the user can toggle between +1 and any other value two times to achieve the same effect.

As shown on a previous table, not all types support the `SELECT/EXECUTE/BREAK` operations. For the `BS` type, even though not meaningful as is for the other types, for a write operation to succeed, either `SELECT` or `EXECUTE` must be used.

ADDRESS SPECIFICATION:

The address specifies the Information Object Address to be associated with the tag.

Syntax: `<Information Object Address>.<Quality Field>`

Where:

- `<Information Object Address>` is the address of the type on the device on the range 0-16777215
- `<Quality Field>` indicates special fields, available on some of the supported types, for quality purposes. Its meaning and availability are indicated on the following table:

Name	Meaning	SP	DP	ST	BS	NV	SV	FV	IT	EI
OV	Overflow			•	•	•	•	•		
BL	Blocked	•	•	•	•	•	•	•		
SB	Substituted	•	•	•	•	•	•	•		
NT	Not Topical	•	•	•	•	•	•	•		
IV	Invalid	•	•	•	•	•	•	•		
TS	Transient State			•						
CY	Carry (Counter Overflow)								•	
CA	Counter was Adjusted								•	
SQ	Sequence number								•	

Notes:

- If an address uses a quality field, it does not support writing.
- The `EI` key, for receiving values of an End of Initialization command has special values for the addresses. This command send two codes, where the first one indicates the cause of initialization, and will be stored on address 0, and the second one, that is a bit, to indicate change of parameters, will be stored on address 1.
- Notice that quality fields not supported by some headers will not be invalidated, as they simply will not receive any value.

EXAMPLES:

The following examples use the header on first indentation level and addresses on second.

- `SP.MONITOR:2000` → Monitor of single points with address starting on 2000
  - `10` → Monitor of value of Single Point at address 2010 (2000 from offset plus 10 of the address)
  - `10.IV` → Monitor of quality field “Invalid” of Single Point at address 2010
- `DP.SELECT:0.SPULSE:1000` → Selection of double points with address starting on 0, with a qualifier of short pulse, and will wait 1 second for the command activation confirmation
  - `10` → Selection of the Double Point at address 10, for a short pulse operation
  - `10.IV` → This is an invalid address for selection, once quality fields are not writeable

3. Station Status

The header for acquiring the station status is used to get the last error code that happened on the driver for the station specified on the address. The timestamp and the value of the tag are updated when an error happens.

Header Syntax: `STATUS`

Address Syntax: `<Common Address>:<IP Address>:<Opt Port Number>`

Notice that the address syntax follows the same syntax rules as for the station field, and must match exactly the values set on the station field of the driver sheets. *For this header, the station field is meaningless, and should be left blank, as an exception.*

The status of the driver is set on the associated tag on the following situations:

Error Code(s)	Situation
0	Connection reestablished
1, 2, 6, 44, 45, 46, 47	Message arrival
3, 5, -57, -58	Link status check
14	Connection lost

✍ **Notes**:

- The occurrence of error code 14, meaning the connection was lost, also reflects on the quality of the tags associated with the station. All the tags have the quality set to BAD and the timestamp updated to the current time of the operating system.
- The tags associated with stations’ status may change of value very quickly. This will happen, for example, when an error happens that demands the connection to be closed. The status tag will receive the error code that caused the disconnection, the “Connection Lost” code, and, if the driver is allowed to reconnect to the outstation and it happens successfully, the code of “Connection reestablished”.

EXAMPLES:

- `1:127.0.0.1` → Get the last error code for the station of address 1 and IP 127.0.0.1
- `10:192.168.11.125:9600` → Get the last error code for the station of address 1 and IP 192.168.11.125 and port 9600

✍ **Notes**:

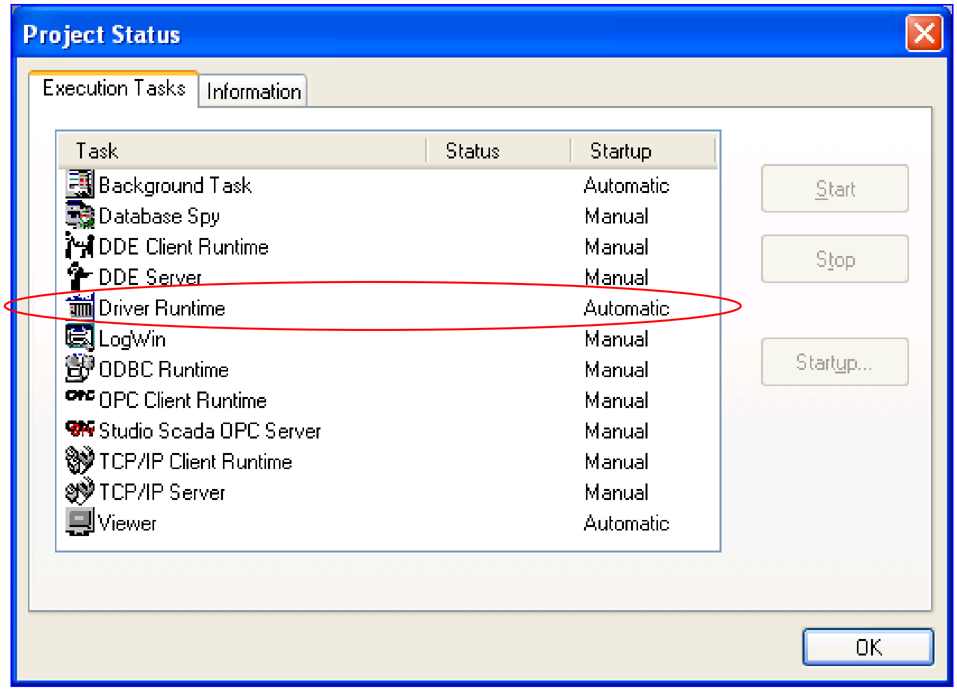
- The tags used by the driver for communication will show the timestamp of the PLC's local time. This must be noted if the driver is running on a machine with a different time zone as the PLC/device/simulator the tags used in communication will show the timestamp of the PLC/device/simulator and not that of the machine the driver is running on.

Executing the Driver

By default, Studio will automatically execute your selected communication driver(s) during application runtime. However, you may verify your application's runtime execution settings by checking the *Project Status* dialog.

To verify that the communication driver(s) will execute correctly:

1. From the main menu bar, select **Project** → **Status**. The *Project Status* dialog displays:



Task	Status	Startup
Background Task		Automatic
Database Spy		Manual
DDE Client Runtime		Manual
DDE Server		Manual
Driver Runtime		Automatic
LogWin		Manual
ODBC Runtime		Manual
OPC Client Runtime		Manual
Studio Scada OPC Server		Manual
TCP/IP Client Runtime		Manual
TCP/IP Server		Manual
Viewer		Automatic

Project Status Dialog

2. Verify that the *Driver Runtime* task is set to **Automatic**.

   - If the setting is correct, then proceed to step 3 below.
   - If the **Driver Runtime** task is set to **Manual**, then select the task and click the **Startup** button to toggle the task's *Startup* mode to **Automatic**.

3. Click **OK** to close the *Project Status* dialog.

4. Start the application to run the driver.

Troubleshooting

If the IE104 driver fails to communicate with the target device, then the database tag(s) that you configured for the **Read Status** or **Write Status** fields of the Standard Driver Sheet will receive an error code. Use this error code and the following table to identify what kind of failure occurred.

Error Code	Description	Possible Causes	Procedure to Solve
0	OK	N/A	N/A
1	Malformed Packet	The data on the packet was not recognized to be a correct IEC 60870-5-104 valid packet	- Check for noise on the communication channel - Check communication settings - Check outstation for correct protocol configuration according to interoperability form
2	Wrong Sequence Number	The received sequence number is wrong	- Check for noise on the communication channel - Check communication settings
3	Timeout (T_1)	Timeout waiting for Acknowledge or confirmation of control functions	- Check for timeout parameters on both sides of communication - Check for noise on the communication channel - Check communication settings
4	Bad ASDU Length	Invalid ASDU length (too many elements)	- Check for noise on the communication channel - Check communication settings
5	Transmission Stopped	The connection is either closed or the outstation has not authorized transmission	- Check for communication settings on the outstation regarding flow control of messages
6	Wrong Common Address	The message received has a different Common Address than configured for the connection	- Check the station of driver sheets - Check the common address sent by the outstations
7	Bad Station	Station address was not found on connection list	- Check the syntax of the station field
8	Invalid Counter Group	The group number for counter interrogation is outside range [0,4]	- Check the number of the group of the counter interrogation command issued
9	Command Ignored	The requested command was ignored due to the value specified on the tag	- If the command was not intended to be ignored, check the value specified on the tag
10	Invalid Operation	The requested operation is not supported (write for a MONITOR data sheet for example, or any read trigger)	- Check triggers - Check headers' syntax
11	Invalid Value	The value for writing is outside valid range	- Check the value specified on the tag to be inside the range of values specified previously on this document

12	Confirmation not received	The expected confirmation for a SELECT/BREAK command was not received before timeout (T_1).	- Check for noise on the communication channel - Check communication settings - Check settings on the outstation
13	Negative Confirmation received	The Activation/Deactivation Confirmation was received; however it had the negative bit set.	- Check if addresses are valid - Check settings on the outstation
14	Connection Lost	The connection to the outstation was lost	- Check for noise on the communication channel - Check communication settings - Check settings on the outstation
44	Unknown type identification	The outstation have not recognized the command	- Check if the requested command is supported by the device
45	Unknown cause of transmission	The outstation have not recognized the cause of transmission of the command	- Please contact support to report and solve the issue
46	Unknown common address of ASDU	The outstation have not accepted the common address configured on the station field	- Check the station fields for the common address - Check the device's configuration for the common address
47	Unknown information object address	The outstation have not accepted the address of the type requested	- Check if the address requested matches the desired object on the device, and if it exists
101	Bad Buffer Length	Internal error.	- Please contact support to report and solve the issue
-37	Invalid Header	Header was not recognized	- Check your driver sheet to ensure the requested operation is valid for the header, and that its syntax is correct
-38	Invalid Station	The syntax for the station field is incorrect. This error is shown on the log windows when attempting to create a connection.	- Check for the station syntax: - Check if the IP address is valid - Check if common address is in range 0-65534 - Check if the station contains at least the common address and IP address separated by colon

⇨ **Tip:**

You can monitor communication status by establishing an event log in Studio's `Output` window (`LogWin` module). To establish a log for **Field Read Commands**, **Field Write Commands** and **Protocol Analyzer,** right-click in the `Output` window and select the desired options from the pop-up menu.

You can also use the *Remote LogWin* module (**Tools** → **LogWin**) to establish an event log on a remote unit that runs Windows CE

If you are unable to establish communication between Studio and the target device, then try instead to establish communication using the device's own programming software. Quite often, communication is interrupted by a hardware or cable problem or by a device configuration error. If you can successfully communicate using the programming software, then recheck the driver's communication settings in Studio.

If you must contact us for technical support, please have the following information available:

- **Operating System and Project Information** (type and version): To find this information, select **Help → Support Information.**
- **Driver Version** and **Communication Log**: Displays in the Studio *Output* window when the driver is running.
- **Device Model** and **Boards**: Consult the hardware manufacturer's documentation for this information.

Interoperability Form

This section was extracted from IEC-60870-104 interoperability document. It should be used to verify the interoperability between the Studio driver and other IEC implementations.

Extracted from 60870-5-104 © IEC:2000, page 83

The interoperability form is used to inform what is supported or not by the driver. The standard of boxes and strike through is taken from the IEC documentation and is expected to be used in the devices' documentation as well. Use the following legend for this section :

■ ~~Filled and strike through~~: Not applicable for this protocol, but applicable in other parts of the same family (such as IEC 101).

□(Clear) : Applicable but not supported by the driver

☒ (Marked with an X): Applicable and supported by the driver

System or device

(system-specific parameter)

□ System definition

☒ Controlling station definition (Master)

□ Controlling station definition (Slave)

Network configuration

(network-specific parameter)

■ ~~Point-to-point~~	■ ~~Multipoint-party line~~
■ ~~Multiple point-to-point~~	■ ~~Multipoint-star~~

Physical Layer

(network-specific parameter)

<u>Transmission speed (control direction)</u>

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~~	■ ~~2400 bit/s~~	■ ~~2400 bit/s~~	■ ~~56000 bit/s~~
■ ~~200 bit/s~~	■ ~~4800 bit/s~~	■ ~~4800 bit/s~~	■ ~~64000 bit/s~~

■ ~~300 bit/s~~
■ ~~600 bit/s~~
■ ~~1200 bit/s~~

■ ~~9600 bit/s~~

■ ~~9600 bit/s~~
■ ~~19200 bit/s~~
■ ~~38400 bit/s~~

Transmission speed (monitor direction)

Unbalanced interchange
circuit V.24/V.28
Standard

■ ~~100 bit/s~~
■ ~~200 bit/s~~
■ ~~300 bit/s~~
■ ~~600 bit/s~~
■ ~~1200 bit/s~~

Unbalanced interchange
circuit V.24/V.28
Recommended if >1 200 bit/s

■ ~~2400 bit/s~~
■ ~~4800 bit/s~~
■ ~~9600 bit/s~~

Balanced interchange
circuit X.24/X.27

■ ~~2400 bit/s~~
■ ~~4800 bit/s~~
■ ~~9600 bit/s~~
■ ~~19200 bit/s~~
■ ~~38400 bit/s~~

■ ~~56000 bit/s~~
■ ~~64000 bit/s~~

Link Layer

(network-specific parameter)

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

Link transmission procedure

■ ~~Balanced transmission~~

■ ~~Unbalanced transmission~~

Frame length

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

Address field of link

■ ~~Not present (balanced transmission only)~~

■ ~~One octet~~

■ ~~Two octets~~

■ ~~Structured~~

■ ~~Unstructured~~

Application Layer

Transmission mode for application data

Mode 1 (Least significant octet first), as defined in clause 4.10 of IEC 870-5-4, is used exclusively in this companion standard.

Common address of ASDU
(system-specific parameter)

■ One octet ☒ Two octets

Information object address
(system-specific parameter)

■ ~~One octet~~
■ ~~Two octets~~
☒ Three octets

□ structured
☒ unstructured

Cause of transmission
(system-specific parameter)

■ One octet ☒ Two octets (with originator address)

Length of APDU
(system-specific parameter)

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)

☒	<1> := Single-point information	M_SP_NA_1
■	~~<2> := Single-point information with time tag~~	~~M_SP_TA_1~~
☒	<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~~
☒	<9> := Measured value, normalized value	M_ME_NA_1
■	~~<10> := Measured value, normalized value with time tag~~	~~M_ME_TA_1~~
☒	<11> := Measured value, scaled value	M_ME_NB_1
■	~~<12> := Measured value, scaled value with time tag~~	~~M_ME_TB_1~~
☒	<13> := Measured value, short floating point value	M_ME_NC_I
■	~~<14> := Measured value, short floating point value with time tag~~	~~M_ME_TC_1~~
☒	<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_PS_NA_1

□<21> := Measured value, normalized value without quality descriptor M_ME_ND_1

☒<30> := Single-point information with time tag CP56Time2A M_SP_TB_1

☒<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

☒<34> := Measured value, normalized value with time tag CP56Time2A M_ME_TD_1

☒<35> := Measured value, scaled value with time tag CP56Time2A M_ME_TE_1

☒<36> := Measured value, short floating point value with time tag CP56Time2A M_ME_TF_1

☒<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

Process information in control direction

(station-specific parameter)

☒<45> := Single command C_SC_NA_1

☒<46> := Double command C_DC_NA_1

☒<47> := Regulating step command C_RC_NA_1

☒<48> := Set point command, normalized value C_SE_NA_1

☒<49> := Set point command, scaled value C_SE_NB_1

☒<50> := Set point command, short floating point value C_SE_NC_1

☒<51> := Bitstring of 32 bit C_BO_NA_1

☒<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

System information in monitor direction

(station-specific parameter)

☒<70> := End of initialization	M_EI_NA_1

System information in control direction
(station-specific parameter)

☒ <100> := Interrogation command	C_IC_NA_1
☒ <101> := Counter interrogation command	C_CI_NA_1
☐ <102> := Read command	C_RD_NA_1
☒ <103> := Clock synchronization command	C_CS_NA_1
■ ~~<104> := Test command~~	~~C_TS_NB_1~~
☐ <105> := Reset process command	C_RP_NC_1
■ ~~<106> := Delay acquisition command~~	~~C_CD_NA_1~~
☐ <107> := Test command with time tag CP56Time2a	C_TS_TA_1

Parameter in control direction
(station-specific parameter)

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

File transfer
(station-specific parameter)

☐ <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	F_DR_TA_1

Basic application functions

Station initialization
(station-specific parameter)

☒ Remote initialization

Cyclic data transmission
(station-specific parameter)

☒ Cyclic data transmission

Read procedure

(station-specific parameter)

□ Read procedure

Spontaneous transmission

(station-specific parameter)

☒ Spontaneous transmission

General Interrogation

(system- or station-specific parameter)

☒ global

☒ group 1

☒ group 2

☒ group 3

☒ group 4

☒ group 5

☒ group 6

☒ group 7

☒ group 8

☒ group 9

☒ group 10

☒ group 11

☒ group 12

☒ group 13

☒ group l4

☒ group 15

☒ group 16

Addresses per group have to be defined

Clock synchronization

(station-specific parameter)

□ Clock synchronization

Command transmission

(object-specific parameter)

☒ Direct command transmission

☒ Direct set point command transmission

☒ Select and execute command

☒ Select and execute set point command

□ C_SE_ACTTERM used

☒ No additional definition

☒ Short pulse duration (duration determined by a system parameter in the outstation)

☒ Long pulse duration (duration determined by a system parameter in the outstation)

☒ Persistent output

Transmission of Integrated totals

(station- or object-specific parameter)

☒ Mode A: Local freeze with spontaneous transmission

☒ Mode B: Local freeze with counter interrogation

☒ Mode C: Freeze and transmit by counter-interrogation commands

☒ Mode D: Freeze by counter-interrogation command, frozen values reported spontaneously

☒ Counter request

☒ Counter freeze without reset

☒ Counter freeze with reset

☒ Counter reset

☒ General request counter
☒ Request counter group 1
☒ Request counter group 2
☒ Request counter group 3
☒ Request counter group 4

Addresses per group have to be defined

Parameter loading
(object-specific parameter)

□ Threshold value
□ Smoothing factor
□ Low limit for transmission of measured value
□ High limit for transmission of measured value

Parameter activation
(object-specific parameter)

□ Act/deact of persistent cyclic or periodic transmission of the addressed object

Test procedure
(station-specific parameter)
□ Test Procedure

File transfer
(station-specific parameter)

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)
☒ Background scan

Acquisition of transmission delay
(station-specific parameter)
■ ~~Acquisition of transmission delay~~

Definition of time outs

Parameter	Default value	Remarks	Selected value
t_0	30 s	Time-out of connection establishment	60 s
t_1	15 s	Time-out of send or test APDUs	Configurable in advanced settings. Default is 1 s
t_2	10 s	Time-out for acknowledges in case of no data messages $t_2 < t_1$	The value is calculated as $t_2 = 2/3 * t_1$
t_3	20 s	Time-out for sending test frames in case of a long idle state	Configurable. Default is 20 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	12 APDUs
w	8 APDUs	Latest acknowledge after receiving w I format APDUs	8 APDUs

Portnumber

Parameter	Value	Remarks	Selected value
Portnumber	2404	In all cases.	Configurable. Defaults to 2404

Revision History

Doc. Revision	Driver Version	Author	Date	Description of Changes
A	1.0	André Körbes	Aug. 1st, 2011	Initial version
B	1.1	André Körbes	Oct. 18th, 2011	Changed a parameter on driver INI
C	1.2	André Körbes	Apr. 16th, 2013	Updated the accepted Common Address and Information Object Address ranges.
D	1.3	Anushree Phanse	Aug 07th 2015	Fixed issue on WinCE version where driver is unable to communicate
E	1.3	Anushree Phanse	Sept 7th 2016	Improved documentation: Legend added to explain the interoperability section. No changes to the driver.
F	1.4	Anushree Phanse	Dec 22nd 2016	Fixed bad timestamp issue on tags used in communication.
G	1.5	Paulo Balbino	Apr 25th 2018	Added Clock Sync feature