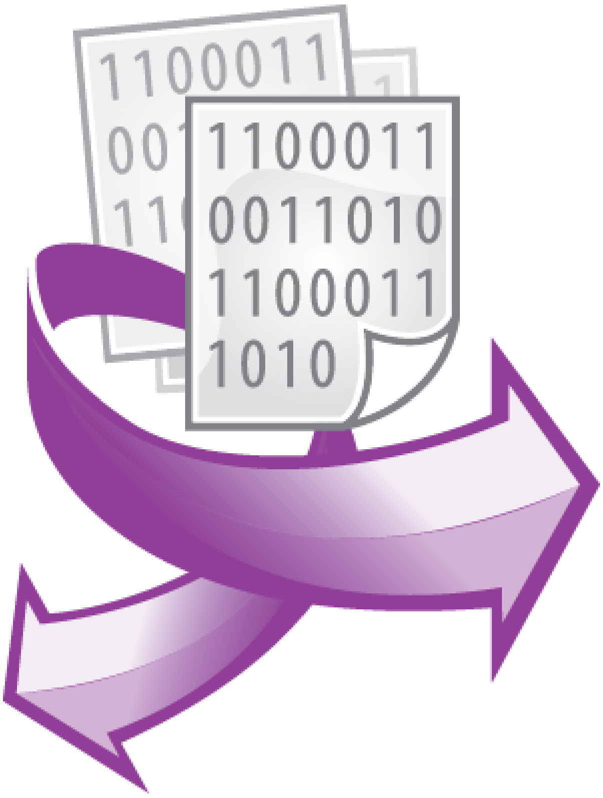


The IEC 62056-21 plugin
PRINTED MANUAL

IEC 62056-21 plugin

While every precaution has been taken in the preparation of this document, the publisher and the author assume no responsibility for errors or omissions, or for damages resulting from the use of information contained in this document or from the use of programs and source code that may accompany it. In no event shall the publisher and the author be liable for any loss of profit or any other commercial damage caused or alleged to have been caused directly or indirectly by this document.

Printed: 11/2/2024

Publisher

AGG Software

Production

© 2018-2024 AGG Software
http://www.aggsoft.com

Table of Contents

1 Introduction

The "IEC 62056-21" module is designed to read data from IEC 62056-21 compatible devices through RS232 or RS485 data interfaces (you need the Opto-to-RS232 converter to read data from optical probes). The module can poll all listed devices and their measured values at regular intervals. The received values are passed to the main program, where they can be, for example, logged or written to a database.

For example, the following devices can send data using this communication protocol:
Satec EM720
Iskra Emeco MT174, MT371, MT372, MT860, MT880
Elgama-Elektronika GAMA 300
DSS1088
Landis+Gyr

The plugin supports modes A, B, C and D. The plugin does not support the mode E. The plugin executes the "Readout" command to get all values at once.

2 System requirements

The following requirements must be met for "IEC 62056-21" to be installed:

Operating system: Windows 2000 SP4 and above, including both x86 and x64 workstations and servers. The latest service pack for the corresponding OS is required.

Free disk space: Not less than 5 MB of free disk space is recommended.

Special access requirements: You should log on as a user with Administrator rights in order to install this module.

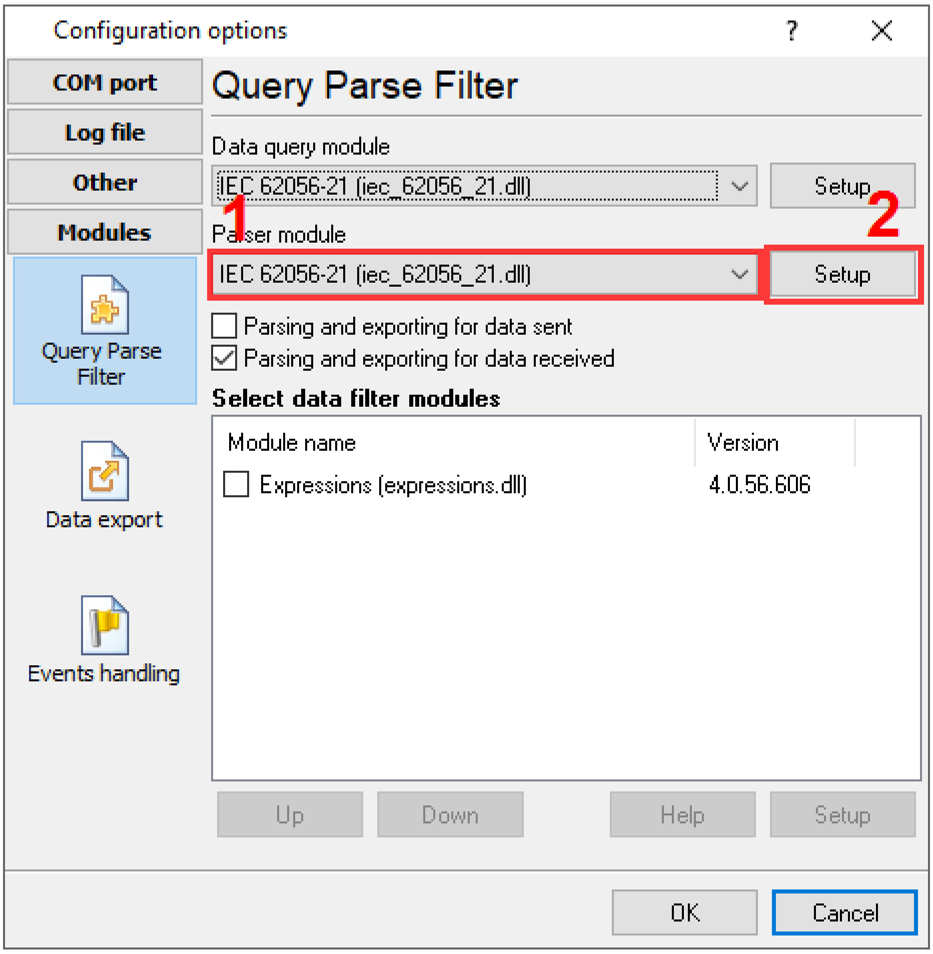
The main application (core) must be installed, for example, Advanced Serial Data Logger.

3 Installing IEC 62056-21

1. Close the main application (for example, Advanced Serial Data Logger) if it is running;
2. Copy the program to your hard drive;
3. Run the module installation file with a double click on the file name in Windows Explorer;
4. Follow the instructions of the installation software. Usually, it is enough just to click the "Next" button several times;
5. Start the main application. The name of the module will appear on the "Modules" tab of the "Settings" window if it is successfully installed.

If the module is compatible with the program, its name and version will be displayed in the module list. You can see examples of installed modules on fig.1-2. Some types of modules require additional configuration. To do it, just select a module from the list and click the "Setup" button next to the list. The configuration of the module is described below.

You can see some types of modules on the "Log file" tab. To configure such a module, you should select it from the "File type" list and click the "Advanced" button.



Query Parse Filter

Data query module: IEC 62056-21 (iec_62056_21.dll)

Parser module: IEC 62056-21 (iec_62056_21.dll)

□ Parsing and exporting for data sent
☑ Parsing and exporting for data received

Select data filter modules

Module name	Version
□ Expressions (expressions.dll)	4.0.56.606

Fig. 1. Example of installed module

4 Glossary

Main program - it is the main executable of the application, for example, Advanced Serial Data Logger and asdlog.exe. It allows you to create several configurations with different settings and use different plugins.

Plugin - it is the additional plugin module for the main program. The plugin module extends the functionality of the main program.

Parser - it is the plugin module that processes the data flow, singling out data packets from it, and then variables from data packets. These variables are used in data export modules after that.

Core - see "Main program."

5 Requests queue

For the module to work, you need to make a request list. Each request may be addressed to one or more devices. All requests in the queue are performed one by one, top to bottom.

To add a new request, click the "Action" button and select "Add a request." A dialog box will open (Figure 1). Enter a request description, which can include any characters, and click the "OK" button.

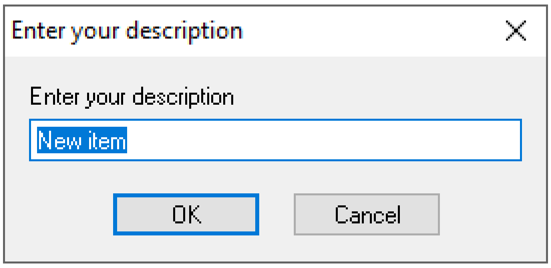
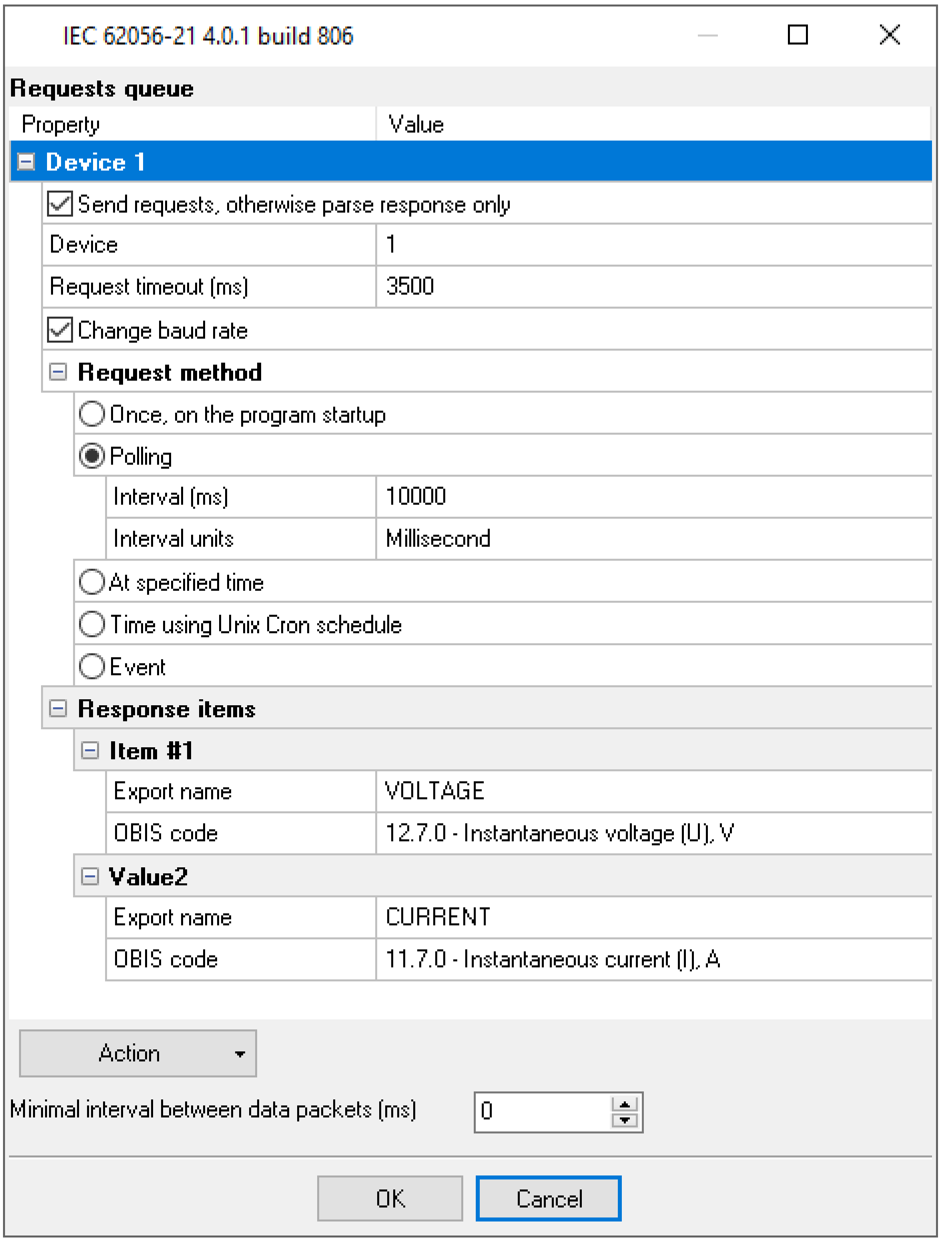


Figure 1: Request description entry dialog box

The new request will appear in the request list (Figure 2). Each request includes a few important options that need to be configured:

- **Send requests, otherwise only process responses**: Turn on this option if you want the module to poll devices at regular intervals. This option should be enabled in most cases.
- **Device**: A unique identifier of a device in the RS485 network. To specify multiple addresses, use the semicolon as a delimiter (for example, 1;2;3).
- **Request timeout (ms)**: The time to wait for a device’s response after sending a request. If this time expires, the module will automatically cancel the current request and proceed to the next request in the queue. If any data are received from that device after the time has expired, they will be ignored. Response waiting time depends on the network and on the performance of the devices.
- **Change baud rate** - the standard states that a device may use two different baud rates while communicating with a host computer. The baud rate "300" is used for establishing the connection and for basic information exchange between the device and the computer. Then the device "proposes" a new baud rate, and the computer have to change to the new baud rate. Despite the fact that the standard defines the first speed as 300, some devices may be configured in order to use another initial baud rate.



IEC 62056-21 4.0.1 build 806

Requests queue

Property	Value
⊟ **Device 1**	
☑ Send requests, otherwise parse response only	
Device	1
Request timeout (ms)	3500
☑ Change baud rate	
⊟ **Request method**	
○ Once, on the program startup	
● Polling	
Interval (ms)	10000
Interval units	Millisecond
○ At specified time	
○ Time using Unix Cron schedule	
○ Event	
⊟ **Response items**	
⊟ **Item #1**	
Export name	VOLTAGE
OBIS code	12.7.0 - Instantaneous voltage (U), V
⊟ **Value2**	
Export name	CURRENT
OBIS code	11.7.0 - Instantaneous current (I), A

Minimal interval between data packets (ms): 0

Figure 2: Request queue

Next chapters explain parameters in the "Request method" and "Response items" groups.

6 Request method

The plugin can send requests in the following mode:

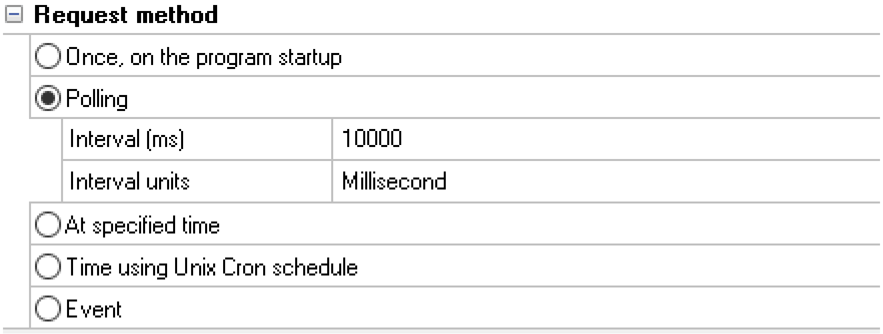
Once, on program startup - the program will send a request once when the program starts.

Polling - the program will send a request periodically based on an interval specified. The interval between requests depends on the network on which master (program) and slave (device) is running. If the network is slow, then the time for each request will be larger and vice versa. Because the program executes all requests in the queue one by one, the time between requests depends on the number of requests in the queue.

At the specified time - the time of the day using the 24hr format (e.g., 18:00:00). You may specify several time points separated by a semicolon (e.g. 11:00:00;11:20:00;11:40:00).

Time, using Unix Cron schedule - a flexible schedule format that allows sending requests periodically or at the specified time. You can find detailed information about this format and see examples in the "Cron time format" section. The default is 0 0 12 * * *, which means "every week, every day at 12:00:00".

Event - the program executes the corresponding request when the plugin receives an external event. These events can be generated by our other plugins, like "Event generator," "Script execute," "Expressions,"



Request method

○ Once, on the program startup

● Polling

Interval (ms)	10000
Interval units	Millisecond

○ At specified time

○ Time using Unix Cron schedule

○ Event

Fig. 2. Request methods

If you have added several requests to the queue, you can move them up or down. To do it, select a request, click the "Action" button, and select an action ("Move up" or "Move down").

You can also click this button to change a request's description or remove a request from the queue.

You can also perform the same actions by using the context menu that pops up when you right-click items in the request tree.

7 Cron time format

The CRON format is a simple yet powerful way to describe time and operation periodicity. The traditional (inherited from the Unix world) CRON format consists of five fields separated with spaces:

<Second> <Minutes> <Hours> <Month days> <Months> <Weekdays>

Any of the five fields can contain the * (asterisk) character as its value. It stands for the entire range of possible values. For example, every minute, every hour and so on. In the first four fields, you can also use the proprietary "?" (w/o quotes) character. See its description below.

Any field can contain a list of comma-separated values (for example, 1,3,7) or an interval (subrange) of values defined by a hyphen (for example, 1-5).

You can use the / character after the asterisk (*) or after an interval to specify the value increment. For example, you can use 0-23/2 in the "Hours" field to specify that the operation should be carried out every two hours (old version analog: 0,2,4,6,8,10,12,14,16,18,20,22). The value */4 in the "Minutes" field means that the operations must be carried out every four minutes. 1-30/3 is the same as 1,4,7,10,13,16,19,22,25,28.

You can use three-word abbreviations in the "Months" (Jan, Feb, ..., Dec) and "Weekdays" (Mon, Tue, ..., Sun) fields instead of numbers.

Examples

Note: the <Second> field equal 0 in all examples

Format	**Description**
* * * * *	every minute
59 23 31 12 5	one minute before the end of the year if the last day in the year is Friday
59 23 31 Dec Fri	one minute before the end of the year if the last day in the year is Friday (one more variant)
45 17 7 6 *	every year on the 7th of June at 17:45
0,15,30,45 0,6,12,18 1,15,31 * 1-5 *	00:00, 00:15, 00:30, 00:45, 06:00, 06:15, 06:30, 06:45, 12:00, 12:15, 12:30, 12:45, 18:00, 18:15, 18:30, 18:45, if it is the 1st, 15th or 31st of any month and only on workdays
*/15 */6 1,15,31 * 1-5	00:00, 00:15, 00:30, 00:45, 06:00, 06:15, 06:30, 06:45, 12:00, 12:15, 12:30, 12:45, 18:00, 18:15, 18:30, 18:45, if it is the 1st, 15th or 31st of any month and only on workdays (one more variant)
0 12 * * 1-5 (0 12 * * Mon-Fri)	at noon on workdays
* * * 1,3,5,7,9,11 *	every minute in January, March, May, July, September, and November
1,2,3,5,20-25,30-35,59 23 31 12 *	on the last day of the year at 23:01, 23:02, 23:03, 23:05, 23:20, 23:21, 23:22, 23:23, 23:24, 23:25, 23:30, 23:31, 23:32, 23:33, 23:34, 23:35, 23:59
0 9 1-7 * 1	on the first Monday of every month at 9 in the morning
0 0 1 * *	at midnight on the 1st of every month
* 0-11 * *	every minute till noon
* * * 1,2,3 *	every minute in January, February, and March
* * * Jan,Feb,Mar *	every minute in January, February, and March
0 0 * * *	every day at midnight
0 0 * * 3	every Wednesday at midnight

You can use the proprietary "?" character in the first four fields of the CRON format. It stands for the start time, i.e., the question mark will be replaced with the start time during the field processing: minute for the minute field, hour for the "Hours" field, month day for the month day field, and month for the month field.

For example, if you specify:

? ? * * *

The task will be run at the moment of startup and will continue being run simultaneously (if the user does not restart the program again, of course) – the question marks are replaced with the time the program was started at. For example, if you start the program at 8:25, the questions marks will be replaced like this:

25 8 * * * *

Here are some more examples:

- ? ? ? ? * - run _only_ at startup;
- ? * * * * - run at startup (for example, at 10:15) and continue being run in exactly one hour: at 11:15, 12:15, 13:15 and so on;
- * ? * * * - run every minute during the startup hour;
- */5 ? * * * - run on the next day (if CRON is not restarted) at the same hour every minute and so on every day, once in five minutes, during the startup hour.

8 Response items

The "Response items" node contains a list of parsed values.

If this list is empty, the plugin will extract and export all possible values returned by the "Readout" command. In this case, the name of an exported variable is an OBIS code.

To add more variables to a request, click the "Action" button, and select "Add a response." Before doing that, you need to select the request from the queue. The new variable will appear in the "Response items" group (Figure 2).

Each response item has the following options:

- **Name**: The name of the variable. The value of the variable is passed under that name to the main program and used in the data export modules. You must give a unique name to each response item in the request. However, variable names in different requests may coincide.
- **OBIS code**: You should select the OBIS code of the necessary value. The list contains all supported codes. Your device may have a limited set of values from this list.

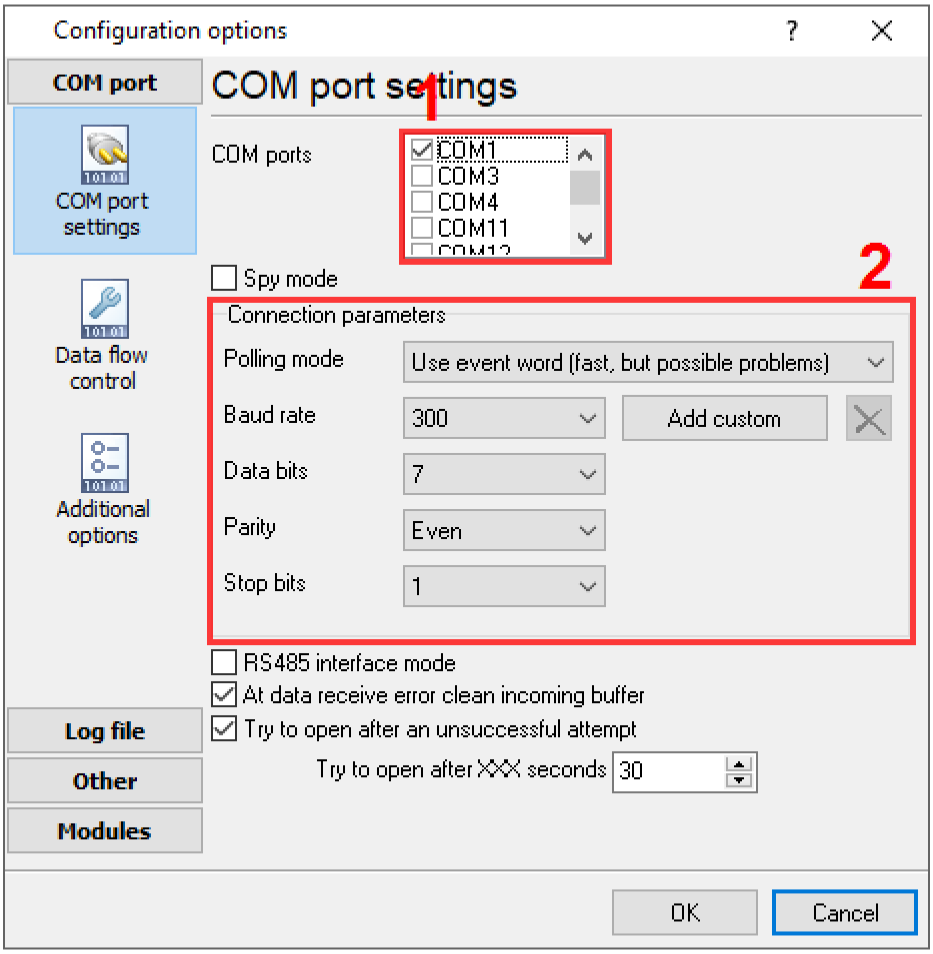
9 How to configure the logger

This module implements data communication based on the IEC 62056-21 protocol through RS232 or RS485 connections. For this reason, the module can be used with the following products only:

- Advanced Serial Data Logger
- Data Logger Suite

To configure a connection, you need to create a new configuration in the program by clicking the "Green Plus" button in the main window. In the COM port settings, select the correct COM port number (Figure 1). The standard COM port parameters are:

Baud rate: 300 (some devices allow configuring the standard baud rate in the device settings)
Data bits: 7
Parity: Even
Stop bits: 1



COM port settings

COM ports
☑ COM1
□ COM3
□ COM4
□ COM11
□ COM12

□ Spy mode

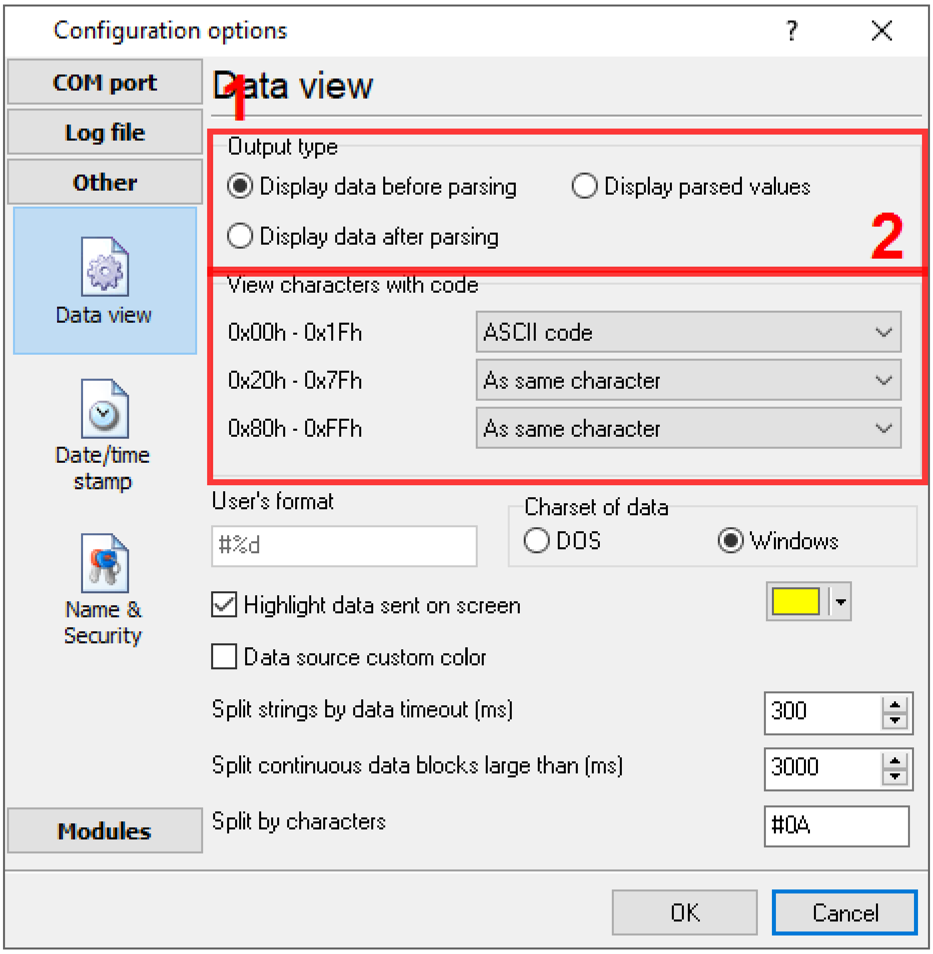
Connection parameters

Polling mode: Use event word (fast, but possible problems)
Baud rate: 300
Data bits: 7
Parity: Even
Stop bits: 1

□ RS485 interface mode
☑ At data receive error clean incoming buffer
☑ Try to open after an unsuccessful attempt
Try to open after XXX seconds: 30

Figure 1: Configuring the COM connection

After that, you can select the recommended data view mode (Figure 2), so that the received values are displayed in the main window.



Data view

Output type

● Display data before parsing ○ Display parsed values

○ Display data after parsing

View characters with code

0x00h - 0x1Fh: ASCII code

0x20h - 0x7Fh: As same character

0x80h - 0xFFh: As same character

User's format: #%d

Charset of data: ○ DOS ● Windows

☑ Highlight data sent on screen 🟨

□ Data source custom color

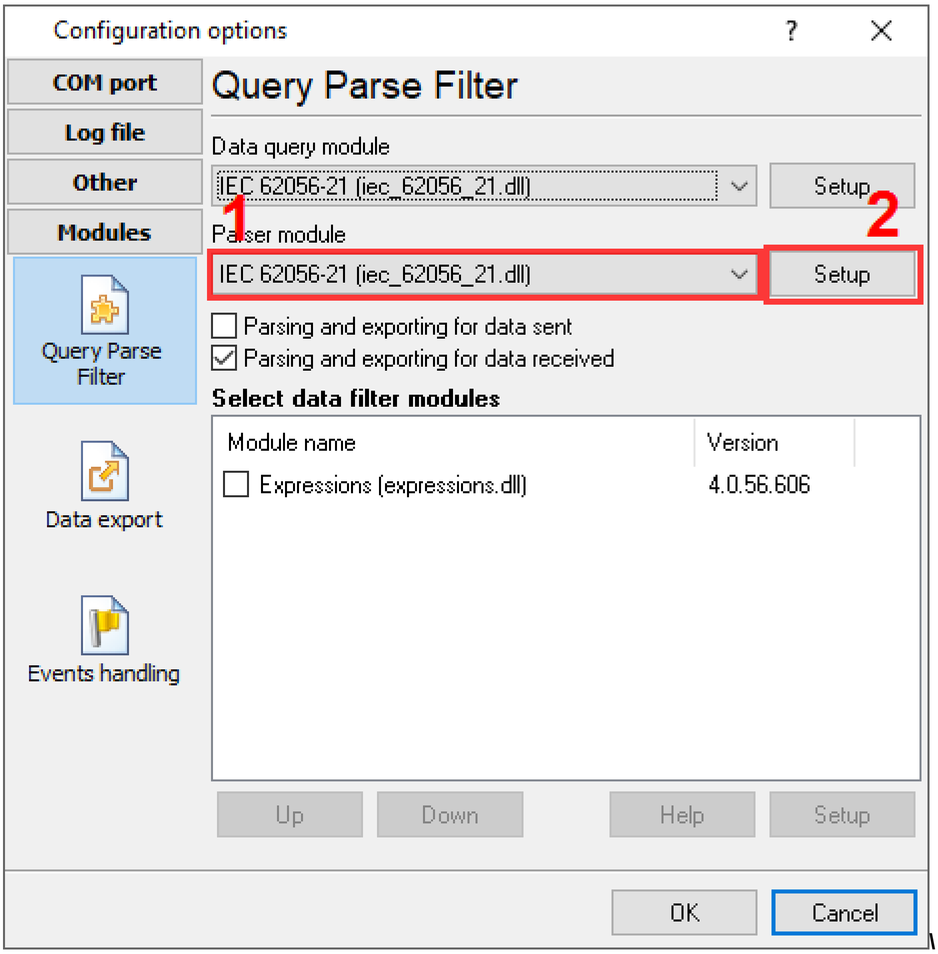
Split strings by data timeout (ms): 300

Split continuous data blocks large than (ms): 3000

Split by characters: #0A

Figure 2: Data display mode

On the "Query Parse Filter" tab, select "IEC 62056-21" in the "Data query" and "Data parser" fields (Figure 3). The list of active filter modules and export modules depends on the specific task. Then click the "Setup" button, and configure the requests queue for your device.



Query Parse Filter

Data query module: IEC 62056-21 (iec_62056_21.dll) Setup

Parser module: IEC 62056-21 (iec_62056_21.dll) Setup

□ Parsing and exporting for data sent

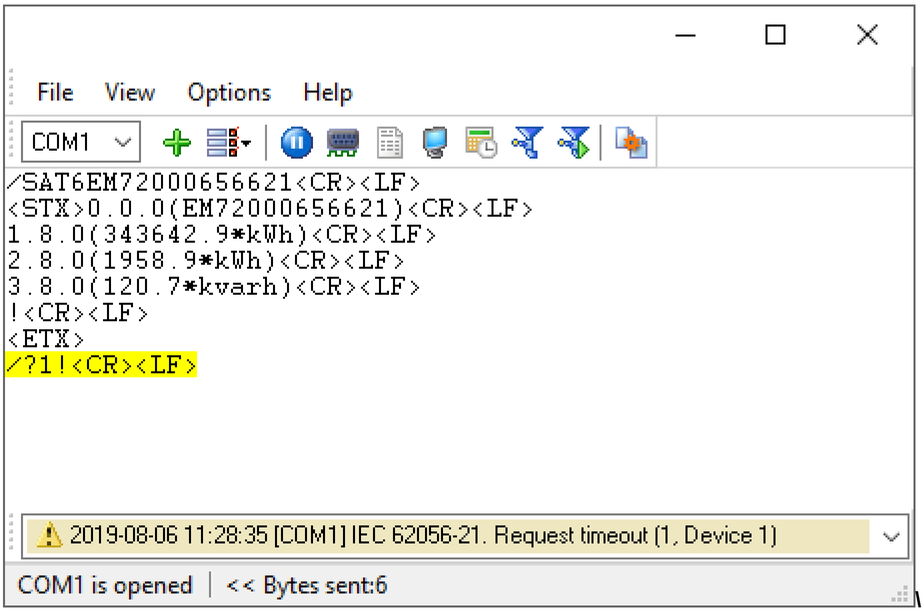
☑ Parsing and exporting for data received

Select data filter modules

Module name	Version
□ Expressions (expressions.dll)	4.0.56.606

Figure 3: Selecting modules

In the result, you should see outgoing requests (yellow) and incoming data (Figure 4). If the device is offline, you'll see the "Timeout" message in the drop-down box with program messages.



```
/SAT6EM72000656621<CR><LF>
<STX>0.0.0(EM72000656621)<CR><LF>
1.8.0(343642.9*kWh)<CR><LF>
2.8.0(1958.9*kWh)<CR><LF>
3.8.0(120.7*kvarh)<CR><LF>
!<CR><LF>
<ETX>
/?1!<CR><LF>
```

2019-08-06 11:28:35 [COM1] IEC 62056-21. Request timeout (1, Device 1)

COM1 is opened | << Bytes sent:6

Figure 4: Outgoing and incoming data

10 Troubles?

10.1 Possible problems

No data for publication/exporting - no data is passed for export. Solution: configure the parser and make sure that one or more variables are defined in the parser.

Error on binding variable with name %s [%s] - the error usually occurs if data does not correspond to the specified format. For example, the date and time format does not match source data.

Unable to disconnect from the database [%s] and **Unable to connect to a database [%s]** - it is impossible to connect or disconnect form the database. You should check the parameters of the database connection. The analysis of the additional information will help you locate the error.

Database access error [%s]. Stop operations with the database? - the message appears if an error occurs during an attempt to execute an SQL query if the second variant of reacting to errors is selected. The message implies the "Yes" or "No" answer. The analysis of the additional information will help you locate the error.

Unable to verify your SQL script [%s] - the message appears when an attempt to analyze your SQL query fails. Check if the syntax of your SQL query is correct.

Tested successfully - the message appears if your database connection is successfully tested. It requires no additional actions.

Database isn't used - the message appears if the module is temporarily disabled (the "Temporarily disabled" check box is selected) or the database name field is empty. Check the connection parameters.

Database isn't selected - the message appears if the database type is not selected. Check the connection parameters.

Database: %s - %s contains the database name. The message appears if the database connection is successful. Usually, you see it when you call the module for the first time. It requires no additional actions.

Invalid data block length (columns=%d,length=%d) - an internal application error. It means that the data sent by the parser is in an invalid format. Perhaps, you are using the module incompatible with the version of the Advanced Serial Data Logger kernel. Update the versions of both the kernel and the module.

The time of connection is not due yet (%d,%d) - the message appears during an attempt to connect to the database after the connection to it has been lost and the "Reconnect after" option is enabled. No additional actions are required.

Invalid procedure call. Bad arguments - an attempt to call the module using invalid parameters. Perhaps, you are using the module incompatible with the version of the Advanced Serial Data Logger kernel. Update the versions of both the kernel and the module.

Writing to the database is complete - the message appears if your queue of SQL queries is successfully executed. It requires no additional actions.

Writing to the database is complete with errors - the message appears if the execution of your queue of SQL queries was interrupted by an error. It requires no additional actions.

Your SQL is empty. Please, specify some SQL text first - the message appears if you do not enter the text for your SQL query. Check if the options on the "SQL queue" tab are configured correctly.

Invalid temporary path - the path to the temporary file specified by you does not exist. Enter a new path in the "Temporary folder" field on the "Errors handling" tab.

%s, %d - will be replaced by additional information.