The Kamstrup (KMP protocol) plugin
PRINTED MANUAL

Kamstrup (KMP protocol) plugin

Printed: 11/2/2024

Publisher

AGG Software

Production

© 1999-2024 AGG Software
http://www.aggsoft.com

Table of Contents

1 Introduction

The Kamstrup [KMP protocol] parser plugin facilitates communication with Kamstrup Multical 601, 801, and other heat and cooling meters. It serves as a straightforward alternative to M-Bus or MODBUS protocols, offering a simpler communication protocol option. By utilizing user-defined register numbers, this plugin enables effortless retrieval of essential values for both existing and upcoming models.

This protocol adopts a half-duplex communication mode with a master-slave structure. A computer or server acts as the master station that polls multiple meters, which function as slave stations.

This module has the following features:

- The software is capable of extracting real-time data from KMP-compatible devices in a user-friendly format.
- This plugin is compatible with both serial and network interfaces, allowing for seamless integration.
- Users can customize the interval at which meter data is retrieved.
- Simultaneous polling of data from multiple devices is supported by this plugin.
- Measurement units and precision are automatically detected by the plugin, eliminating the need for manual configuration.

2 System requirements

The following requirements must be met for "Kamstrup (KMP protocol)" 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 Kamstrup (KMP protocol)

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.

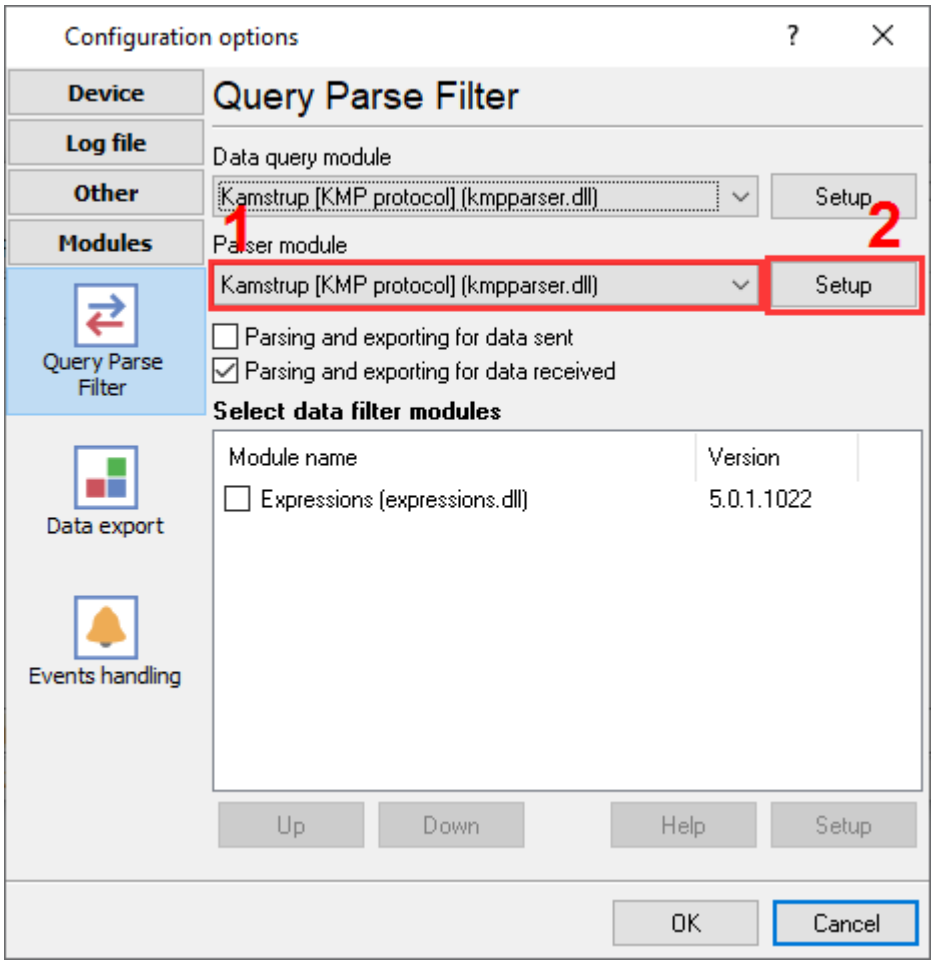

Fig. 1. Example of installed module

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

To add a new item, click "Actions - Add new request." The dialog window will be shown (fig.1). Enter a request description that can contain any characters and click the "OK" button.

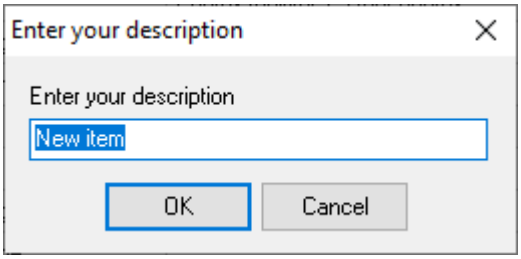

Fig.1. Name dialog

The new data request will appear in the requests tree (fig.2). Each request has a few important options:

Device address - the device address refers to the communication address assigned to a KMP-compatible meter for point-to-point communications. You should program the unique device address in the meter. To specify the device address in the plugin, use a decimal number ranging from 1 to 255. If there are multiple similar meters on one bus, you can enter several device addresses separated by a semicolon. The plugin will then poll them sequentially.

Request timeout - A request timeout refers to the maximum amount of time a program will wait for a response before canceling the current request. When the timeout limit is reached, the program automatically moves on to the next request in the queue. This ensures that the program doesn't get stuck waiting indefinitely for a response. The timeout value is influenced by two main factors: media speed and device performance. If the data transfer rate over the media is slow, it is advisable to set a larger timeout value. This allows sufficient time for the response to be received, preventing premature cancellation of requests.

Response items - by selecting the "Action - Add value" command, you can add up to 8 registers per one request.

Register - the plugin contains the pre-defined list of commonly used registers in the KMP protocol.

Custom register - if you select "Custom" in the register field, you can specify any register number in the range from 1 to 65535 (decimal). Of course, this register should exist in the device. Usually, the device user manual contains the list of supported registers and their meaning.

Export name - the pre-defined registers already come with a variable designated for exportable variables, such as E1, V1, and T1. However, you can assign any name you desire (only Latin letters and digits).

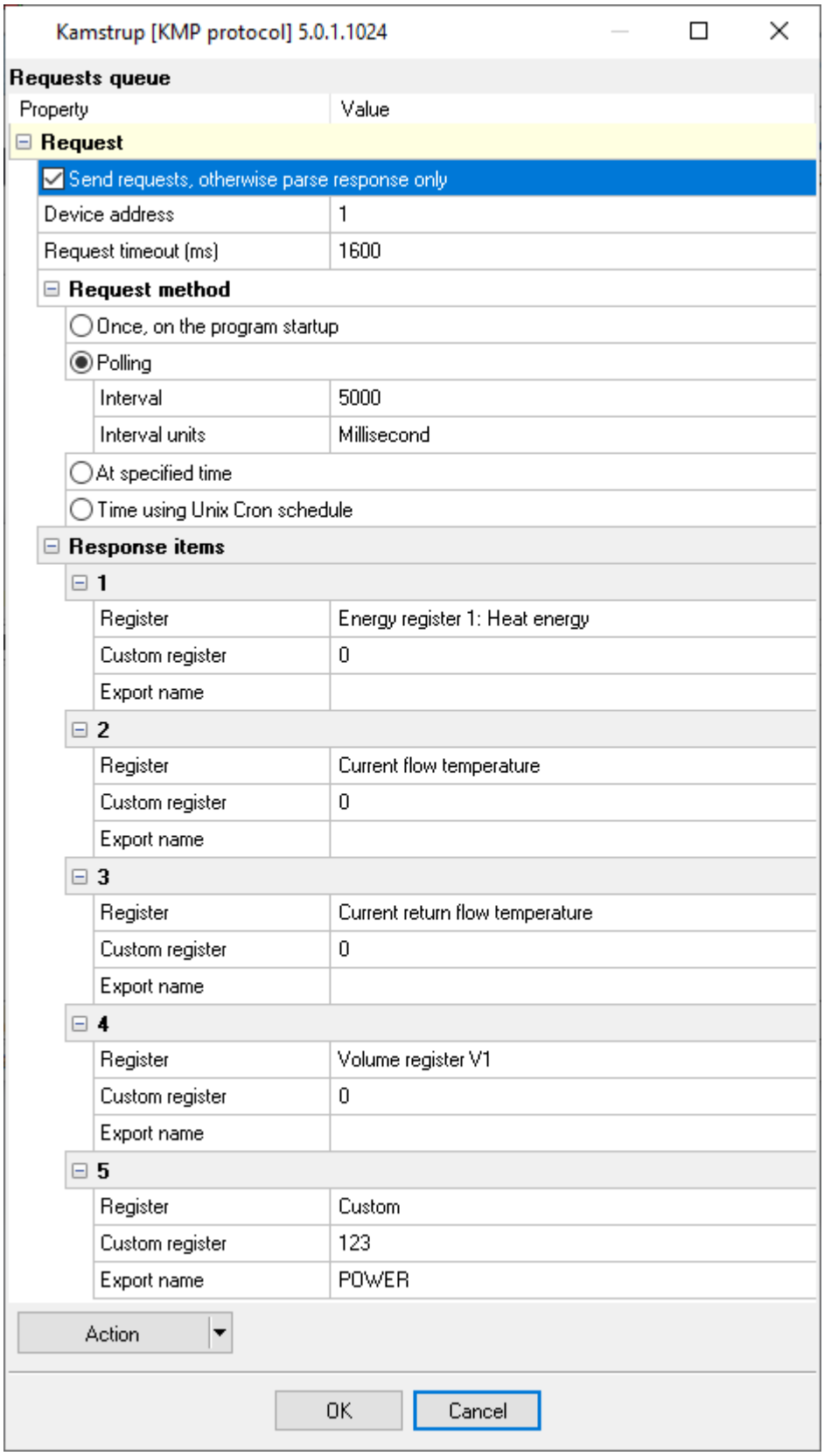

Fig.2. KMP request

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,"

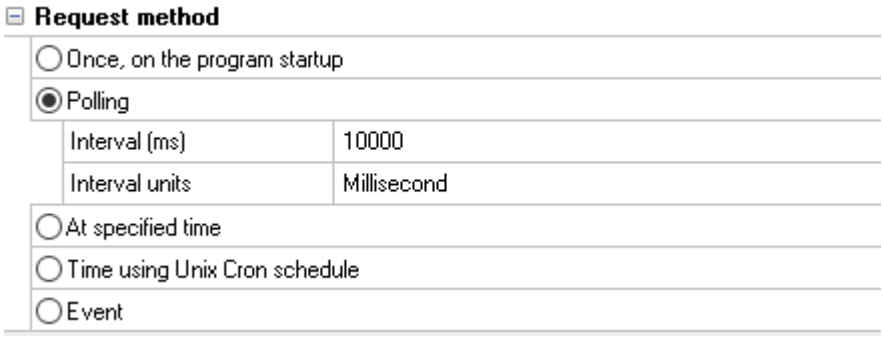

Fig. 2. Request methods

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.