RUT900 Data to Server

The information in this page is updated in accordance with firmware version .
***Note**: **click here** for the old style WebUI (FW version RUT9XX_R_00.06.09.5 and earlier) user manual page.*

□

Contents

- 1 Summary
- 2 Data Senders
  - 2.1 Sender Settings
    - 2.1.1 Data configuration
    - 2.1.2 Collection general settings
    - 2.1.3 Collection advanced settings
    - 2.1.4 Server configuration

Summary

The **Data to Server** feature provides you with the possibility to set up data senders that collect data from various sources and periodically send it to remote servers.

If you're having trouble finding this page or some of the parameters described here on your device's WebUI, you should **turn on "Advanced WebUI" mode**. You can do that by clicking the "Advanced" button, located at the top of the WebUI.

Data Senders

A **Data Sender** is an instance that gathers and periodically sends collected data to a specified server. The Data Senders list is empty by default so, in order to begin configuration you must add a new data sender first. To add a new data sender, click the 'Add' button.

After this you should be redirected to the newly added data sender's configuration page.

Sender Settings

Refer to the table below for descriptions on data sender configuration fields.

Note: these tables have coloring schemes to indicate which fields can be seen with different configuration.

Data configuration

Field	Value	Description
Name	string; default: **none**	Name of the data sender. Used for easier data senders management purposes only (optional).
Type	ModbusModbus alarms | WiFi scanner data* | DNP3* | DNP3 data flash*; default: **Modbus data**	Source of the data to be sent to server.
Format type	Json, custom; default: **Json**	Arranges the format of the sent segment.
JSON format	string; default: **{"TS": "%t", "D": "%d", "data": %a}**	Arranges the format of the sent JSON segment.
Segment count	integer [1..10]; default: **1**	Max segment count in one JSON string sent to server.
Send as object	off | on; default: **off**	When turned on, sends JSON segment as object and not as an array element.
Data filtering	All data | By slave ID | By slave IP; default: **All data**	If Data source: Modbus data. Choose which data this sender will send to server.
By slave ID: Slave ID	integer [1..255]; default: **none**	ID of the Modbus slave whose collected data will be sent to server.
By slave IP: Slave IP	ip; default: **none**	IP address of the Modbus slave whose collected data will be sent to server (for Modbus TCP slaves).
Data filtering	All data | DNP3 Address | DNP3 IP; default: **All data**	If Data source: DNP3. Choose which data this sender will send to server.
DNP3 Address: DNP3 Address	integer [0..65519]; default: **none**	DNP3 address of device whose collected data will be sent to server.
DNP3 IP: DNP3 IP	ip | domain; default: **none**	IP address of DNP3 device whose collected data will be sent to server.
Retry on fail	off | on; default: **off**	When turned on, the data sender retries failed sending attempts until the are successfully delivered.

Collection general settings

Field	Value	Description
Enable	on, off; default: **on**	Enables instance.
Type	Json, custom; default: **Json**	Data input type.
Json	string; default: **{"TS": "%t", "D": "%d", "data": %a}**	Arranges the format of the sent JSON segment.

custom	string; default: **empty**	Type of data formatting.
Empty value	string; default: **N/A**	A string which will be placed if any value cannot be received.

Collection advanced settings

Field	Value	Description
Period	integer [1..86400]; default: **60**	Data sending frequency (in seconds).
Retry	off | on; default: **off**	When turned on, the data sender retries failed sending attempts until the are successfully delivered.
Retry count	integer [1..10]; default: **10**	Retry to send the same data N times.
Timeout	integer [1..60]; default: **1**	Timeout in second between retry attempts.

Server configuration

Field	Value	Description
Protocol	HTTP(S) | MQTT* | Kinesis; default: **HTTP(S)**	Protocol used for sending the data to server.
Server address	string; default: **none**	URL for HTTP(S); Host for MQTT; Connection string for Azure MQTT.
HTTP Header	string; default: **none**	Allows to add custom headers to the HTTP requests.
Enable secure connection	on, off; default: **off**	Enables use of TLS certificates
Certificate files from device	on, off; default: **off**	Choose this option if you want to select certificate files from device.
Certificate based: CA File	.ca file; default: **none**	**Certificate authority** is an entity that issues digital certificates. A digital certificate certifies the ownership of a public key by the named subject of the certificate.
Certificate based: Client Certificate	.crt file; default: **none**	Certificate file is a type of digital certificate that is used by client systems to make authenticated requests to a remote server. If client certificate is not needed, leave both client certificate and client key fields empty.
Certificate based: Private Key	.key file; default: **none**	File containing private key for this client. This file needs to be not encrypted.

* This is additional software that can be installed from the **System → Package Manager** page.