Template:Networking rutos manual rs232

The information in this page is updated in accordance with firmware version .

[[{{{name}}} RS232|RS232]] service was moved to [[{{{name}}} Serial Utilities|Serial Utilities]] page from FW version *{{{series}}}_R_00.07.03*.

☐

Contents

Summary

The **RS232** page is used to configure the operating parameter of RS232 serial connector.

This manual page provides an overview of the RS232 page in {{{name}}} devices.

RS232 Configuration

The **RS232 Configuration** section is used to set up the main operating parameters and the serial type of the RS232 connector.

[[File:Networking_rutos_manual_rs232_rs232_configuration_data_bits_{{{data_bit}}}.png|border|class=tlt-border]]

Field	Value	Description
Enabled	off | on; default: **off**	Turns the RS232 service on or off.
Baud rate	{{{baud_rates}}}; default: **{{{baud_rate}}}**	Data rate for serial data transmission (in bits per second (bps)).
Data bits	{{{data_bits}}}; default: **{{{data_bit}}}**	Number of data bits for each character.

Parity	None | Odd | Even; default: **None**	In serial transmission, parity is a method of detecting errors. An extra data bit is sent with each data character, arranged so that the number of 1 bits in each character, including the parity bit, is always odd or always even. If a byte is received with the wrong number of 1s, then it must have been corrupted. However, an even number of errors can pass the parity check. • **None** (**N**) - no parity method is used. • **Odd** (**O**) - the parity bit is set so that the number of "logical ones (1s)" has to be odd. • **Even** (**E**) - the parity bit is set so that the number of "logical ones (1s)" has to be even.
Stop bits	1 | 2; default: **1**	Stop bits sent at the end of every character allow the receiving signal hardware to detect the end of a character and to resynchronise with the character stream. Electronic devices usually use one stop bit. Two stop bits are required if slow electromechanical devices are used.
Flow control	None | RTS/CTS | Xon/Xoff; default: **None**	In many circumstances a transmitter might be able to send data faster than the receiver is able to process it. To cope with this, serial lines often incorporate a "handshaking" method, usually distinguished between hardware and software handshaking. • **RTS/CTS** - hardware handshaking. RTS and CTS are turned OFF and ON from alternate ends to control data flow, for instance when a buffer is almost full. • **Xon/Xoff** - software handshaking. The Xon and Xoff characters are sent by the receiver to the sender to control when the sender will send data, i.e., these characters go in the opposite direction to the data being sent. The circuit starts in the "sending allowed" state. When the receiver's buffers approach capacity, the receiver sends the Xoff character to tell the sender to stop sending data. Later, after the receiver has emptied its buffers, it sends an Xon character to tell the sender to resume transmission.
Serial type	Console | Over IP | Modem | Modbus gateway; default: **Console**	Specifies the serial connection type. **Look to the sections below for information on different RS232 serial type options.**

Console

Console mode requires no further configuration than the settings above and is used as a direct-access method to the device's shell interface. For this purpose you may want use such applications as PuTTY on Windows and microcom, minicom, picocom or similar applications on Linux.

Over IP

The **Over IP** serial type is used to manage serial connections over a TCP/IP network.

[[File:Networking_rutos_manual_usb_tools_printer_server_usb_to_serial_over_ip_{{{series}}}_v1.png|border|class=tlt-border]]

Field	Value	Description
Protocol	TCP | UDP; default: **TCP**	Protocol used in the communication process.
Raw mode	off | on; default; default: **off**	When enabled, all data will be transmitted transparently.
Mode	Server | Client | Bidirect; default: **Server**	This device's role in the connection: • **Server** - the device waits for incoming connections. • **Client** - the device initiates the connection. • **Bidirect** - acts as client by default but waits for incoming connections at the same time.
No leading zeros	off | on; default: **off**	When checked, indicates that the first hex zeros should be skipped.
Server settings: Port	integer [0..65535]; default: **none**	Internal port number used to listen for incoming connections.
Server settings | TCP: Timeout (s)	integer [0..32767]; default: **300**	Specifies an inactivity time limit (in seconds) after which an inactive clients is disconnected.
Bidirect: Output	Specifies which output to manage.	
Bidirect: Output state	0 | 1; default: **0**	Default output state value, when the application is started.
Server settings | UDP: Number of clients	1-10; default: **1**	Specifies how many UDP clients will be supported simultaneously (predefined clients does not count towards this limit).
Server settings | UDP: Predefined client 1 address	ip4; default: **none**	Specifies IP address for predefined connection 1.
Server settings | UDP: Predefined port 1	port; default: **none**	Specifies port number for predefined connection 1.
Server settings | UDP: Predefined client 2 address	ip4; default: **none**	Specifies IP address for predefined connection 2.
Server settings | UDP: Predefined port 2	port; default: **none**	Specifies port number for predefined connection 2.
Client settings: Server Address	ip | host; default: **none**	IP address or hostname of the server that this client will connect to.

	integer [0..65535]; default: **none**	Server's listening port number.
Client settings: Port		
Client settings: Reconnect interval (s)	integer; default: **none**	Time period (in seconds) between reconnection attempts in case a connection fails.
Serial device read time	integer [0..1000]; default: **none**	Specifies custom read time for the serial device.
Echo	off | on; default: **off**	Turns RS232 echo on or off. RS232 echo is a loopback test usually used to check whether the RS232 cable is working properly.
Server settings | TCP: Max clients	integer [1..32]; default: **32**	Specifies how many clients are allowed to connect simultaneously.
TCP: Always reconnect	off | on; default: **off**	When enabled, a new TCP connection will be made after sending every data package.

Modbus gateway

The **Modbus gateway** serial type allows redirecting TCP data coming to a specified port to an RTU specified by the Slave ID. The Slave ID can be specified by the user or be obtained directly from the Modbus header.

Field	Value	Description
Listening IP	ip; default: **0.0.0.0**	IP address to listen for incoming connections. The default value (*0.0.0.0*) means that this device will listen for incoming connections on any interface or IP address.
Port	integer [0..65535]; default: **502**	Port number to listen for incoming connections.
Slave ID configuration type	User defined | Obtained from TCP; default: **User defined**	Specifies whether slave IDs are user defined or automatically obtained from TCP.

Slave ID | Permitted slave IDs	integer | range of integers; default: **1 or 1-247**	Specifies the slave ID of range of permitted slave IDs. The way this field is named and its function depends on the value of the *Slave ID configuration* field. A range of IDs can be specified by placing a hyphen (-) between two integer numbers. For example, if you permit slave IDs in the range of 10 to 20, you would specify it as: *10-20* You can also specify multiple values that are not connected in a range using commas (,). For example, to specify 6, 50 and 100 as permitted slave IDs, you would have to use: *6,50,100*
Open port automatically	off | on; default: **on**	Automatically adds a traffic rule in the firewall configuration to open the required port for serial communication. **Caution:** use with care if listening IP is left as the default value (*0.0.0.0*). Leaving it as such will leave the device open for remote connections on the specified port.
Echo	off | on; default: **off**	Turns RS232 echo on or off. RS232 echo is a loopback test usually used to check whether the RS232 cable is working properly.

IP Filter

The **IP Filter** section is used for configuring which network is allowed to communicate with the device. You may add a new instance by selecting the Interface and pressing Add.

Then enter the IP address and save.

[[Category:{{{name}}} Services section]]