

Page

Read

[View source](#)[View history](#)More 

# RUT955 RS232

[Main Page](#) > [RUT Routers](#) > [RUT955](#) > [RUT955 Manual](#) > [RUT955 WebUI](#) > [RUT955 Services section](#) > **RUT955 RS232**

The information in this page is updated in accordance with firmware version [RUT9\\_R\\_00.07.03.2](#).

**Note:** [click here](#) for the old style WebUI (FW version RUT9XX\_R\_00.06.09.2 and earlier) user manual page.

## Contents [\[hide\]](#)

- 1 [Summary](#)
- > [General information](#)
- [RS232 connector pinout](#)
- 4 [Cables](#)
- 5 [RS232 Configuration](#)
  - 5.1 [Console](#)
  - 5.2 [Over IP](#)
  - 5.3 [Modem](#)
  - 5.4 [Modbus gateway](#)
- 6 [IP Filter](#)

## Summary

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

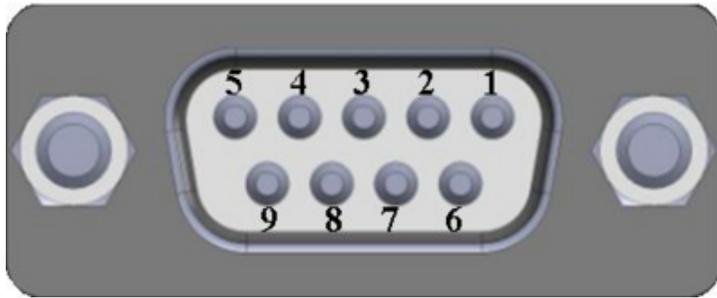
This manual page provides an overview of the RS232 page in RUT955 devices.

## General information

## RS232 connector pinout

The RS232 connector type on this device is a **DCF female**. DCF stands for Data Communication Equipment

The RS232 connector type on this device is a DCE female. DCE stands for Data Communication Equipment.



PIN	NAME*	DESCRIPTION*	DIRECTION ON THIS DEVICE
1	DCD	Data Carrier Detect	Output
2	RXD	Receive Data	Output
3	TXD	Transmit Data	Input
4	DTR	Data Terminal Ready	Input
5	GND	Signal Ground	-
6	DSR	Data Set Ready	Output
7	RTS	Ready To Send	Input
8	CTS	Clear To Send	Output
9	RI	Ring Indicator	Output (connected to +5V permanently via a 4.7k resistor)

## Cables

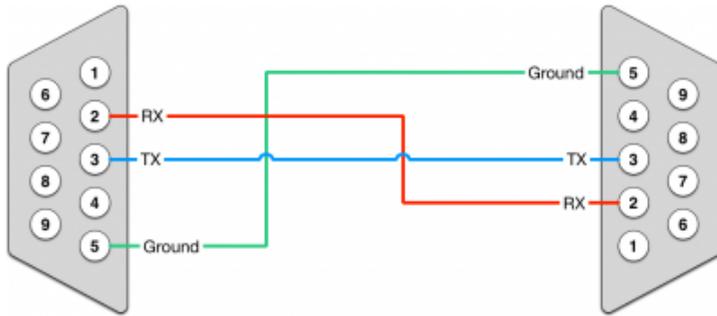
There are two types of RS232 serial devices: **DTE** and **DCE**. DTE typically refers to the serial port on a PC or terminal, while DCE refers to communication devices. Connectors mounted on DTE are likely to be male, and those mounted on DCE are likely to be female.

This device is DCE and has a female connector.

To connect a standard DTE device, use a straight-through Female/Male RS232 cable:



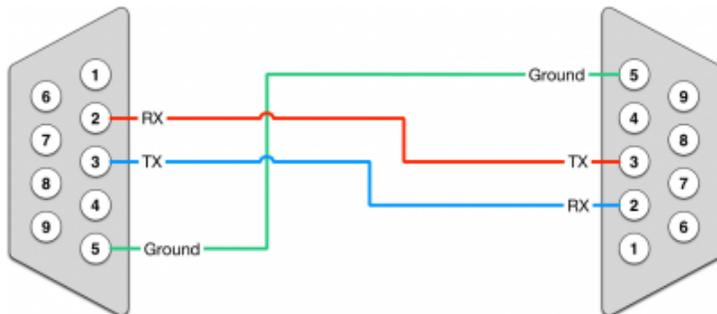
See straight cable pinout below:



To connect another DCE device to RUT955, a Null-modem (crossed) Male/Male cable should be used:



See straight crossed cable pinout below:



Maximum cable length is 15 meters or the cable length equal to a capacitance of 2500 pF (for a 19200 baud rate). Using lower capacitance cables can increase the distance. Reducing communication speed can also increase maximum cable length.

## RS232 Configuration

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

### ^ RS232 CONFIGURATION

---

Enabled  off on

Baud rate

Data bits

Parity

Stop bits

Flow control

Serial type

[SAVE & APPLY](#)

Field	Value	Description
Enabled	off   on; default: <b>off</b>	Turns the RS232 service on or off.
Baud rate	300   600   1200   2400   4800   9600   19200   38400   57600   115200; default: <b>115200</b>	Data rate for serial data transmission (in bits per second (bps)).
Data bits	5   6   7   8; default: <b>8</b>	Number of data bits for each character.
Parity	None   Odd   Even; default: <b>None</b>	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.

		<ul style="list-style-type: none"> <li>• <b>None (N)</b> - no parity method is used.</li> <li>• <b>Odd (O)</b> - the parity bit is set so that the number of "logical ones (1s)" has to be odd.</li> <li>• <b>Even (E)</b> - the parity bit is set so that the number of "logical ones (1s)" has to be even.</li> </ul>
<b>Stop bits</b>	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.
<b>Flow control</b>	None   RTS/CTS   Xon/Xoff; default: <b>None</b>	<p>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.</p> <ul style="list-style-type: none"> <li>• <b>RTS/CTS</b> - 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.</li> <li>• <b>Xon/Xoff</b> - 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.</li> </ul>
<b>Serial type</b>	<a href="#">Console</a>   <a href="#">Over IP</a>   <a href="#">Modem</a>   <a href="#">Modbus gateway</a> ; default: <b>Console</b>	<p>Specifies the serial connection type.</p> <p><b>Look to the sections below for information on different RS232 serial type options.</b></p>

## 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.

Serial type

Console
▼

## Over IP

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

Serial type

Protocol

Raw mode

Mode

No leading zeros

Port

Timeout (s)

Serial device read time

Echo

Max clients

Always reconnect

Field	Value	Description
Protocol	TCP   UDP; default: TCP	Protocol used in the communication process.
Raw mode	off   on; default: off	When enabled, all data will be transmitted transparently.
Mode	Server   Client   Bidirect; default: Server	This device's role in the connection: <ul style="list-style-type: none"> <li><b>Server</b> - the device waits for incoming connections.</li> <li><b>Client</b> - the device initiates the connection.</li> <li><b>Bidirect</b> - acts as client by default but waits for incoming connections at the same time.</li> </ul>
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.

<b>Server settings</b>   <b>TCP: Timeout (s)</b>	integer [0..32767]; default: <b>300</b>	Specifies an inactivity time limit (in seconds) after which an inactive clients is disconnected.
<b>Bidirect: Output</b>	Output (4)   Isolated Output (3,4,8)   Relay (5,10); default: <b>Output (4)</b>	Specifies which output to manage.
<b>Bidirect: Output state</b>	0   1; default: <b>0</b>	Default output state value, when the application is started.
<b>Server settings</b>   <b>UDP: Number of clients</b>	1-10; default: <b>1</b>	Specifies how many UDP clients will be supported simultaneously (predefined clients does not count towards this limit).
<b>Server settings</b>   <b>UDP: Predefined client 1 address</b>	ip4; default: <b>none</b>	Specifies IP address for predefined connection 1.
<b>Server settings</b>   <b>UDP: Predefined port 1</b>	port; default: <b>none</b>	Specifies port number for predefined connection 1.
<b>Server settings</b>   <b>UDP: Predefined client 2 address</b>	ip4; default: <b>none</b>	Specifies IP address for predefined connection 2.
<b>Server settings</b>   <b>UDP: Predefined port 2</b>	port; default: <b>none</b>	Specifies port number for predefined connection 2.
<b>Client settings: Server Address</b>	ip   host; default: <b>none</b>	IP address or hostname of the server that this client will connect to.
<b>Client settings: Port</b>	integer [0..65535]; default: <b>none</b>	Server's listening port number.
<b>Client settings: Reconnect interval (s)</b>	integer; default: <b>none</b>	Time period (in seconds) between reconnection attempts in case a connection fails.
<b>Serial device read time</b>	integer [0..1000]; default: <b>none</b>	Specifies custom read time for the serial device.
<b>Echo</b>	off   on; default: <b>off</b>	Turns RS232 echo on or off. RS232 echo is a loopback test usually used to check whether the RS232 cable is working properly.
<b>Server settings</b>   <b>TCP: Max clients</b>	integer [1..32]; default: <b>32</b>	Specifies how many clients are allowed to connect simultaneously.
<b>TCP: Always reconnect</b>	off   on; default: <b>off</b>	When enabled, a new TCP connection will be made after sending every data package.

## Modem

The **Modem** serial type is used to manage modem functionality which could be accessed using shell interface. For this purpose you may want use such applications with CR/LF (Carriage Return, Line Feed) capable applications like PuTTY on Windows and microcom, minicom, cutecom or similar applications on Linux.

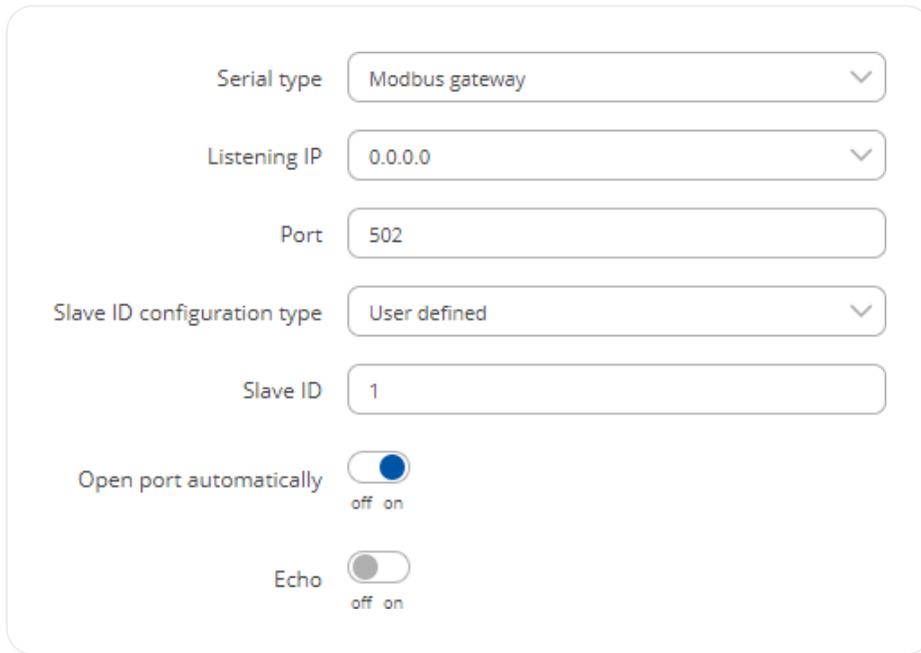
Serial type

Mode

Field	Value	Description
<b>Mode</b>	Partial control   Full control; default: <b>Partial control</b>	Specifies modem control mode. <ul style="list-style-type: none"> <li>• <b>Partial control</b>- enables modem control with AT commands, mobile connection will be controlled by RUTOS.</li> <li>• <b>Full control</b>- enables modem control with AT commands, mobile connection will be controlled by user.</li> </ul>

## 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.



Serial type: Modbus gateway

Listening IP: 0.0.0.0

Port: 502

Slave ID configuration type: User defined

Slave ID: 1

Open port automatically:  on

Echo:  off

Field	Value	Description
Listening IP	ip; default: <b>0.0.0.0</b>	IP address to listen for incoming connections. The default value ( <i>0.0.0.0</i> ) means that this device will listen for incoming connections on any interface or IP address.
Port	integer [0..65535]; default: <b>502</b>	Port number to listen for incoming connections.
Slave ID configuration type	<b>User defined</b>   Obtained from TCP; default: <b>User defined</b>	Specifies whether slave IDs are user defined or automatically obtained from TCP.
<b>Slave ID</b>   Permitted slave IDs	<b>integer</b>   range of integers; default: <b>1 or 1-247</b>	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 <i>Slave ID configuration</i> 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: <i>10-20</i> 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: <i>6,50,100</i>
Open port automatically	off   on; default: <b>on</b>	Automatically adds a traffic rule in the firewall configuration to open the required port for serial communication. <b>Caution:</b> use with care if listening IP is left as the default value ( <i>0.0.0.0</i> ). Leaving it as such will leave the device open for remote connections on the specified port

Echo

off | on; default: **off**

open for remote connections on the specified port.

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.

✓ **IP FILTER**

---

INTERFACE	ALLOW IP
This section contains no values yet	

✓ **ADD NEW INSTANCE**

---

INTERFACE	ALLOW IP
<input type="text" value="LAN"/>	

Then enter the IP address and save.

✓ **IP FILTER**

---

INTERFACE	ALLOW IP
lan	<input type="text" value="0.0.0.0/24"/> <input type="button" value="+"/>

[DISCLAIMERS](#) [MOBILE VIEW](#)

## PRODUCTS

MODEMS  
GATEWAYS  
ROUTERS  
SWITCHES  
IoT PLATFORMS  
ACCESSORIES  
CUSTOM PRODUCT

## INDUSTRIES

INDUSTRIAL & AUTOMATION  
ENERGY & UTILITIES  
SMART CITY  
TRANSPORTATION  
ENTERPRISE  
RETAIL  
ALL USE CASES

## SUPPORT

WIKI KNOWLEDGE BASE  
CROWD SUPPORT FORUM  
REPORT A VULNERABILITY  
WARRANTY & REPAIR  
GPL CODE CENTER  
EOL POLICY

## RESOURCES

WEBINARS  
VIDEO ARCHIVE  
ARTICLES ARCHIVE  
NEWSLETTER

## ABOUT US

NEWS  
COVID-19 RESPONSE  
WHO WE ARE  
NETWORKS IoT ACADEMY  
POLICIES & CERTIFICATES  
CAREER  
CONTACTS

[PRIVACY](#) | [ALL POLICIES](#) | [SITEMAP](#)

COPYRIGHT © 2023, TELNIKA

## NEWSLETTER

SUBSCRIBE

## CONNECT



We use cookies to ensure that we give you the best experience on our website. If you continue to use this site we will assume that you are happy with it.

OK

READ MORE