Template:Networking rut manual rs232 rs485

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

□

Contents

Summary

RS232 and **RS485** serial interfaces provide a possibility for legacy devices to gain access to IP networks.

This chapter of the user manual provides an overview of the RS232/RS485 page for {{{name}}} devices.

RS232

Field	Value	Description
Enabled	yes | no; default: **no**	Turns the RS232 service on or off.

Baud rate	300 | 1200 | 2400 | 4800 | 9600 | 19200 | 38400 | 57600 | 115200; default: **115200**	Data rate for serial data transmission (in bits per second (bps)).
Data bits	5 | 6 | 7 | 8; default: **8**	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 | NTRIP client; default: **Console**	Specifies the serial connection type. More information on serial types can be found in each of their respective sections of this manual page.
Echo	yes | no; default: **no**	Turns RS232 echo on or off. RS232 echo is a loopback test usually used to check whether the RS232 cable is working properly.

RS232 Connector Pinout

RS232 connector type on this device is 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)

* The names and descriptions that indicate signal direction (such as TXD, RXD, RTS, CTS, DTR, and DSR) are named from the point of view of the DTE device.

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.

RS485

RS-485 is a serial data transmission standard for use in long ranges or noisy environments.

Field	Value	Description
Enabled	yes | no; default: **no**	Turns the RS485 service on or off.
Baud rate	300 | 1200 | 2400 | 4800 | 9600 | 19200 | 38400 | 57600 | 115200; default: **115200**	Data rate for serial data transmission (in bits per second (bps)).
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.
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 | NTRIP client; default: **Console**	Specifies the serial connection type. More information on serial types can be found in each of their respective sections of this manual page.

Maximum data rate vs. transmission line length

The RS-485 standard can be used for network lengths up to 1200 meters, but the maximum usable data rate decreases as the transmission length increases. A device operating at the maximum data transfer rate (10 Mbps) is limited to a transmission length of about 12 meters, while the 100 kbps data rate can achieve a distance up to 1200 meters. A rough relation between maximum

transmission length and data rate can be calculated using this approximation:

Where Lmax is the maximum transmission length in meters and DR is maximum data rate in bits per second.

Twisted pair is the preferred cable type for RS-485 networks. Twisted pair cables pick up noise and other electromagnetically induced voltages as common mode signals, which are rejected by the differential receivers.

Cable Type

Recommended cable parameters:

PARAMETER	VALUE
Cable Type	22-24 AWG, 2 – pair (used for full-duplex networks) or 1-pair (used for half duplex networks). One addtitional wire for ground connection is needed
Characteristic cable Impedance	120 Ω @ 1MHz
Capacitance (conductor to conductor)	36 pF/m
Propagation Velocity	78% (1.3 ns/ft)

RS485 connector pin-out

NAME	DESCRIPTION	TYPE
D_P	Driver positive signal	Differential Output
D_N	Driver negative signal	Differential Output
R_P	Receiver positive signal	Differential Input
R_N	Receiver negative signal	Differential Input
Ground	Device ground	Differential Output

2-Wire and 4-Wire Networks

Below is an example of a 4-wire network electrical connection. There are 3 devices shown in the example. One of the devices is the “master” and other two are “slaves”. Termination resistors are placed at each cable end. Four-wire networks consists of one „master“ with its transmitter connected to each of the “slaves‘” receivers on one twisted pair. The “slave” transmitters are all connected to the “master” receiver on a second twisted pair.

Example 2-wire network electrical connection: to enable a 2-wire RS-485 configuration on a

Teltonika router, you need to connect D_P to R_P and D_N to R_N on the device's RS-485 socket. Termination resistors are placed at each cable end.

Termination

When to use (place jumper)

Termination resistor, equal in resistance to cable characteristic impedance, must be connected at each end of the cable to reduce reflection and ringing of the signals when the cable lengths get relatively long. Rise time of the {{{name}}} RS-485 driver is about 5 ns, so the maximum unterminated cable length is about 12 cm. As transmission line cables will always be longer than 12 cm, termination is mandatory all the time if {{{name}}} is located at the end of the cable.

When not to use (remove jumper)

If your RS-485 consists of more than two devices and the {{{name}}} router is located not on the end of the line but, for example, in the middle, {{{name}}} termination resistor needs to be disabled. In this case, place termination at other devices which are situated at the ends of the line.

How to enable termination

120 Ω termination resistor is included on the {{{name}}} PCB and can be enabled by shorting contacts (shown in the picture below), placing 2.54mm pitch jumper:

But on the [[{{{name}}}_Product_Change_Notifications#2019.01.22:_visual_design_changes|{{{name}}} improved housing design]] PCB the terminator connections are missing. Since the housing of the {{{name}}} improved model is not disassembled, the termination resistor must be connected to the outside of the device by connecting a through-hole 100 ohm resistor to the terminal contacts. The connection of the external resistor is shown in the figure:

Number of devices in an RS-485 Network

One {{{name}}} RS-485 driver is capable of driving a maximum of 32 receivers, provided that the receiver input impedance is 12 kΩ. If receiver impedances are higher, the maximum number of receivers in the network increases. Any combination of receiver types can be connected together, provided their parallel impedance does not exceed RLoad > 375 Ω.

Modes of different serial types in RS232 and RS485

Console

In this mode the serial interface set up as a Linux console of the device. It can be used for debugging purposes, to get the status of the device or to control it.

Over IP

In **Over IP Serial** type the router provides a connection to a TCP/IP network for the devices connected via serial interfaces.

Mode: Server

Field	Value	Description
Protocol	TCP; default: **TCP**	Specifies the protocol used in the communication process
Mode	Server | Client | Bidirect; default: **Server**	Specifies the 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	yes | no; default: **no**	Specifies that the first hex zeros should be skipped
TCP port	integer [0..65535]; default: **none**	The port number used to connect to the server
Timeout (s)	integer; default: **none**	Disconnects clients after the amount of inactivity time (in seconds) specified in this field

Mode: Client

Field	Value	Description
Protocol	TCP; default: **TCP**	The protocol used for data transmission
Mode	Server | Client | Bidirect; default: **Server**	**Server** - waits for incoming connection **Client** - initiates the connection **Bidirect** – acts as a client by default, but at the same time waits for incoming connections
No leading zeros	yes | no; default: **no**	Skips first hex zeros
Server address	host | ip; default: **no**	Server address to which the client will connect to
TCP port	integer [0..65535]; default: **none**	The port number used to listen for incoming connections

Reconnect intervals (s)	integer; default: **none**	Indicates the time period between reconnection attempts

Mode: Bidirect

Field	Value	Description
Always reconnect	yes | no; default: **no**	When checked, established a new TCP connection aftereach message.
Mode	Server | Client | Bidirect; default: **Server**	Serial connection operating mode. • **Server** - this device wait for incoming connection. • **Client** - this device initiates the connection • **Bidirect** – acts as a client by default, but at the same time waits for incoming connections
Server address	host | ip; default: **no**	IP address or hostname of the server that this client will connect to.
TCP port	integer [0..65535]; default: **none**	Internal port number used to listen for incoming connections.
Reconnect intervals (s)	integer; default: **none**	Time period (in seconds) between reconnection attempts in case a connection fails.
Port	integer [0..65535]; default: **none**	Server's listening port number.
Timeout (s)	integer; default: **none**	Disconnects client after the specified timeout of inactivity
Output	OC Output | Relay Output; default: **OC Output**	Output to indicate that application switched from client (default) to server state.
Output state	integer [0..1]; default: **0**	Output state value after the application reverts to server mode.

Modem

With Modem Serial type, the router imitates a dial-up modem. Connections to TCP/IP networks can be established using AT commands. The connection can be initiated by the device connected via serial interface with an ATD command: ATD <host>:<port>. If Direct connect settings are specified, the connection to the server is always active. Data mode can be entered by issuing the ATD command. Incoming connections are indicated by sending a RING to the serial interface.

Field	Value	Description
Direct connect	host:port | ip:port; default: **none**	Maintains a constant connection to specified host. Leave empty to use an ATD command to initiate the connection
TCP port	integer [0..65535]; default: **none**	The port number used to listen for incoming connections. Leave it empty to disable incoming connections

Initiation string	string; default: **none**	A command string that will be sent to the modem to initiate it in some special way
No extra CR LF in response	yes | no; default: **yes**	Removes extra CR LF before and LF after response code

This is the AT command set* used in Modem mode of the serial interfaces:

COMMAND	DESCRIPTION	USAGE
A	Answers incoming call	To answer incoming connection: ATA
D	Dial a number	To initiate data connection: ATD <host>:<port> To enter data mode with Direct connect settings: ATD
E	Local echo	Turn local echo on: ATE1 Turn local echo off: ATE0
H	Hang up current call	To end data connection: ATH
O	Return to data mode	To return to data mode from command mode: ATO
Z	Reset to default configuration	To reset the modem to default configuration: ATZ

* Only these commands are supported in Modem mode.

Modbus gateway

The Modbus gateway Serial type allows redirecting TCP data coming to a specified port to 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 on which the Modbus gateway will wait for incoming connections
Port	integer [0..65535]; default: **none**	The port number used 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**	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 permitt 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**

NTRIP client

Networked Transport of RTCM via Internet Protocol (**NTRIP**) is a protocol for streaming differential GPS (DGPS) data over the Internet in accordance with specification published by RTCM.

During the connection initiation between the NTRIP client and NTRIP caster the router sends an initial GPGGA sentence which can be specified in the "Initial NMEA-GGA" field or obtained from the GNSS receiver of {{{name}}}. After the connection is established, further GGA data required by the NTRIP caster to maintain an established connection is acquired from a device connected to the router via RS232/RS485 and immediately forwarded to the NTRIP caster.

Field	Value	Description
IP address	ip; default: **0.0.0.0**	IP address of an NTRIP server.
Port	integer [0..65535]; default: **none**	Port number of the NTRIP server.
Mount point	string; default: **none**	NTRIP mount point.
Data format	NTRIP V2.0 TCP/IP | NTRIP V2.0 RSTP/RTP | NTRIP V1.0 | Automatic detection | NTRIP V2.0 UDP; default: **NTRIP V1.0**	Specifies the version of NTRIP.
User name	string; default: **none**	Username for authentication to NTRIP server.
Password	string; default: **none**	Password for authentication to NTRIP server.
Initial NMEA-GGA	string; default: **none**	Optional NMEA string that will be used as the default value when initiating a connection to an NTRIP server (this value is only sent to the server if there is no NMEA from router's GPS device).
Acquire NMEA-GGA	yes | no; default: **no**	Obtains initial NMEA-GGA string from GPS module of this device. Only works if GPS service is enabled and location fix is obtained at the time of NTRIP service start.

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