



Configuring RS-232 Communications

There are several pieces of information needed to successfully establish an RS-232 connection between the control processor (RP6/ ZRP-6/ MRP-64/ XP-8 or CPB-1) and the device to be control. Configuration is required for the control processor's RS232 settings to exactly match what the equipment being controlled expects. Depending on the control specifications of the equipment being controlled, it may be necessary to use a null modem adapter in conjunction with the CM-232 RS-232 control module.

Checklist of information you will need from the manufacturer

- Baud rate / Parity / Data bits / Stop bits
- Which pin the equipment receives communication on (pin 2 or 3)
- List of RS232 strings for device functions

Communication Parameters

How to read an RS-232 specification

RS-232 parameter specifications will often be spelled out similar to the following format: 9600n81. In this example, 9600 corresponds to the baud rate of the connection, n refers to the parity (n = no parity, other possibilities are e=even parity, o=odd parity, m=mark parity, s=space parity), 8 is the number of data bits, and 1 is the number of stop bits. The specifications might be in a different order, 9600 8n1 for example, but note that the baud rate will always be the largest number, the data bits will always be 7 or 8, the parity is always the single letter, and the stop bit specification is always 1 or 2.

Baud Rate

The baud rate (also called the speed or bps rate) defines how fast that data can be sent to and from the device. The RTI processor serial ports support the following baud rates: 110, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, and 57600. Some devices allow you to select a baud rate, using DIP switches or a setup menu. For control applications, lower baud rates are usually preferable, as they allow for the use of longer cables. If you cannot find the baud rate specified anywhere in the device documentation, you will have to contact the manufacturer of the device or you can try several different settings to see if you can determine the correct one, as choosing an incorrect baud rate will not damage the controlled device.

Data Bits

The data bits setting determines how large each piece of information transmitted over the RS-232 link is. This setting will always be 7 or 8. As a general rule (note: this is not strictly true) if the system is using no parity, then this will be 8 and if the system uses odd, even, mark, or space parity the number of data bits will be 7.

Parity

Parity is a rudimentary form of error detection used with RS-232 connections. There are five possible parity settings: None, Even, Odd, Mark, and Space. In most cases, if the system uses a parity setting of “None” then it will use 8 data bits, otherwise it will use 7.

Stop Bits

The stop bits setting indicates how long a signal that the processor sends to indicate that it has finished sending one byte worth of information to the controlled equipment. This setting will always be 1 or 2, but for almost all equipment, this setting will be 1. If you do not see a stop bit setting specified in the equipment manual, you can usually assume 1 stop bit.

RS-232 Command Strings

The most difficult part of the process is usually configuring the RS-232 command strings. There is no standard for how the command strings are defined, so every manufacturer does it differently. The RS-232 command strings fall into two basic categories: ASCII-based and Binary (or Hex)-based. Note that most strings using letters are case sensitive.

ASCII

ASCII-based communications protocols use simple letters and numbers to send commands to the controlled device. For example, you might send the simple string “OFF” or “INPUT 6” to the device. Another important thing to find out is what type of termination is needed after each string. Most equipment will require a “carriage return,” “linefeed,” or both after each string. This indicates that the command string is complete and the equipment should act on it. When you are creating the command strings in Integration Designer, you place `\r` at the end of the string to indicate a carriage return and `\n` to indicate a linefeed. If the equipment requires both, put `\r\n`. See the table below for a complete list of special characters. Note, that since Integration Designer uses the `\` character to indicate that a “special” character follows, you cannot enter a plain `\` character in your string if you need one. You must write `\` characters as `\\`. This will cause a single backslash to be sent to the equipment you are controlling.

Binary

Binary communication protocols will usually be specified as a sequence of bytes in hexadecimal. A hexadecimal number will usually be specified as two digits, using the number 0-9 and the letters A-F. Each two digit byte is separated by entering the `\x` special character or you can use the “Hex Edit” button in Integration Designer, which will eliminate the need to enter the `\x`, allowing you to simply enter the hexadecimal information. For example, if you have a processor with a “Volume Up” command that is listed as `0x80 0x0A 0x78 0xFF`, you would enter the following string into Integration Designer: `\x80\x0A\x78\xff` (When in “Hex Edit” mode, you would enter 80 0A 78 FF). Unlike ASCII commands, you usually do not need to provide a carriage return or linefeed with a binary command.

<i>Char.</i>	<i>Description</i>
<code>\a</code>	Alert / Bell (Control-G)
<code>\b</code>	Backspace (Control-H)
<code>\e</code>	Escape (Control-I)
<code>\f</code>	Form Feed (Control-L)
<code>\n</code>	Line Feed (Control-J)
<code>\r</code>	Carriage Return (Control-M)
<code>\t</code>	Tab (Control-I)
<code>\v</code>	Vertical Tab (Control-K)
<code>\\</code>	Backslash
<code>\xHH</code>	Hexadecimal constant (where H is a hex digit from 0-9 or A-F)
<code>\dNNN</code>	Decimal constant (where N is a digit from 0-9). All three digits MUST be entered, so you may need to add leading zeros.

Table 1: Integration Designer RS-232 Special Characters

Examples of Different RS232 Command Strings (examples subject to change)

Denon Receiver – Power On = PWON\r (You may need to install a separate ground cable from the RP6 Ground to the Denon chassis ground.)

JVC TV – Power On = \x21\x82\x80\x50\x57\x31\x0A

Marantz Receiver – Power On = @PWR:2\r

Samsung TV - \xAA\x11\xFE\x01\x01\x12 (contains a Checksum calculation at the end - \x12, you must use a Checksum calculator to determine this value. Use the calculator found under Accessories on PC's then change the view to "Scientific", enter each 2 character byte then +).

Fujitsu Plasma (2006 > Newer) – Power On = you must enter in at least 3 commands in a macro with these steps:

Step 1= @G\r (this is the open communication string, should only be sent on Power On)

Step 2= Pause .2 seconds (delay is needed by the Fujitsu to process the previous string)

Step 3= %A0001\r (this is the Power On command string)

Fujitsu Plasma (2006 > Newer) – All other commands = after Power on sequence enter command strings only (No start communication string and time delay needed)

Set Video Input HDMI = %A1016\r (this is the Select Input HDMI command string)

Fujitsu Plasma (2005 > Older) – Power On = you must enter in at least 5 commands in a macro with these steps:

Step 1= @G\r (this is the open communication string)

Step 2= Pause .2 seconds (delay is needed by the Fujitsu to process the previous string)

Step 3= %A0001\r (this is the Power On command string)

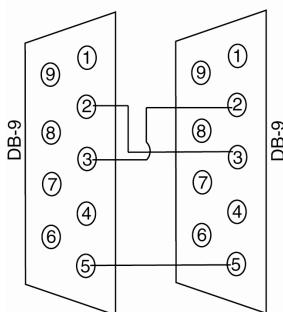
Step 4= Pause .2 seconds (delay is needed by the Fujitsu to process the previous string)

Step 5= @Q\r (this is the close communication string)

Cable Wiring Configuration

RS-232 communication is established by connecting a CM-232 module (cable) between the Control Processor (connected to an available MPIO port, do not use 6" pigtail cable included with processor when using CM-232) and the equipment being controlled. Alternately, the RJ-45 RS-232 connections can be used (CPB-1, ZRP-6, MRP-64, XP-8) for communication. The control processor transmits on pin 3 of the CM-232's DB9 connector and most home theater equipment is wired to receive communications through pin 3, so you can connect the CM-232 directly. If the equipment you are trying to control is wired to receive on pin 2, you will need a null modem adapter or cable to connect between the equipment and the CM-232. A selection of null modem adapters is available at Radio Shack or most computer equipment stores. When using the RJ-45 RS-232 connections, a crossover can be made by simply switching pins 2 and 3 before terminating one end of the cable with the RJ-45 connection. The maximum distance you should expect reliable RS-232 communication is around 50-100 feet depending on the baud rate (longer distances can be achieved with lower baud rates). Note: Do not cut or alter the CM-232 module - if the distance to the controlled equipment needs to be extended, add a cable between the DB9 on the CM-232 and the equipment.

Creating a Null Modem Cable



Use the connection diagram to the left if you wish to make a null modem cable

Troubleshooting

Cabling

- Unplug and plug back in the CM-232 module to verify solid connection
- Verify CM-232 module plugged into RP6 directly (do not use IR extenders that ship with processor)
- Verify which pin the equipment receives communication on.
 - o Pin 3 – connect CM-232 module directly
 - o Pin 2 – install a null modem adapter/cable
- Verify CM-232 cable has not been altered in any way (DB9 contains circuitry that must be in tact)

RS-232 String

- Verify that you have the correct baud rate, parity, data bits and stop bits.
- Verify the format of the RS-232 string
 - o Use correct case (capital/lowercase letters as used in documentation)
 - o If a carriage return is required enter it as: \r
 - o If documentation represents binary bytes as 0xXX, enter them as: \xFF
- If the string requires a Checksum, verify it is correct by using the scientific calculator built in to Windows.

Using HyperTerminal to verify communication

- Plug in a null modem cable into a serial port on your PC (or USB to Serial adapter).
- Connect CM-232 module or RJ-45/DB-9 adapter from processor to null modem cable
- Open HyperTerminal
 - o Go to Start>All Programs>Accessories>Communications>HyperTerminal
 - o Enter a name for connection (any name will do) and click OK
 - o Connect To – set it to the correct Com Port and click OK
 - o Set the Baud Rate (Bits per second), Data Bits, Parity and Stop Bits to the specifications you entered into Integration Designer for your equipment.
 - o Set Flow Control to: None
 - o Click OK

-Press buttons on the remote with RS-232 commands on them and verify that the string shows up in the HyperTerminal window. (Note: the string may look substantially different in HyperTerminal, as your string is being converted into hex characters)