File Name	SpeakercraftRS232Rev1-2	Document Revision	1.2
Prepared By	Ivan Potter	Prepared For	iLED PTY LTD
Description	Speakercraft RS232 Technical Tips		

Speakercraft RS232

iLED (PTY) Ltd
Tel: +27 11 958 2529
Fax: +27 11 958 2591
Email:info@iled.co.za

41 Willow Brook Office Park
BLOCK C Van Hoof Street
Ruimsig

Roodepoort
P.O. Box 3221
Wilropark
1731

Introduction

This document details my experience with Speakercraft MZC's when implementing RS232 interface to external devices. Speakercraft has a mine of information included in the CD's provided to each Custom Installer.

As additional reading, there are numerous on line resources which have been used as reference material. The resource list has been added for your convenience.

Extracts have been used from some of these resources to compile this document

HDMI Organisaton	http://www.hdmi.org
Wikipedia	http://en.wikipedia.org/wiki/High-bandwidth_Digital_Content_Protection
Lammertbies	http://www.lammertbies.nl/comm/cable/RS-232.html

Glossary

RS232	Recommended Standard 232
DTE	Data Terminal Equipment
DCE	Data Communication Equipment
ASCII	American Standard code for Information Interchange

Contents

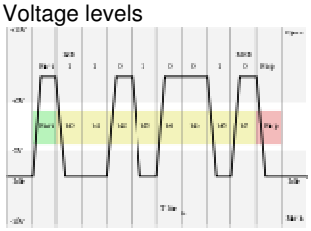
Overview of RS232

Electrical Overview

In telecommunications, **RS-232** is a standard for serial binary data signals connecting between a *DTE* and a *DCE*. It is commonly used in computer serial ports.

In RS-232, user data is sent as a time-series of bits. Both synchronous and asynchronous transmissions are supported by the standard. We only use asynchronous transmissions for this discussion. In addition to the data circuits, the standard defines a number of control circuits used to manage the connection between the DTE and DCE. Each data or control circuit only operates in one direction, that is, signaling from a DTE to the attached DCE or the reverse. Since transmit data and receive data are separate circuits, the interface can operate in a full duplex manner, supporting concurrent data flow in both directions. The standard does not define character framing within the data stream, or character encoding.

Voltage levels



Diagrammatic oscilloscope trace of voltage levels for an uppercase ASCII "K" character (0x4b) with 1 start bit, 8 data bits, 1 stop bit

The RS-232 standard defines the voltage levels that correspond to logical one and logical zero levels. Valid signals are plus or minus 3 to 15 volts. The range near zero volts is not a valid RS-232 level; logic one is defined as a negative voltage, the signal condition is called marking, and has the functional significance of OFF.

Logic zero is positive, the signal condition is spacing, and has the function ON. The standard specifies a maximum open-circuit voltage of 25 volts; signal levels of ±5 V,±10 V,±12 V, and ±15 V are all commonly seen depending on the power supplies available within a device.

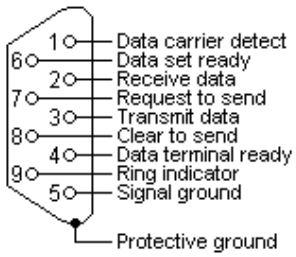
RS-232 drivers and receivers must be able to withstand indefinite short circuit to ground or to any voltage level up to ±25 volts. The slew rate, or how fast the signal changes between levels, is also controlled.

Because the voltage levels are higher than logic levels typically used by integrated circuits, special intervening driver circuits are required to translate logic levels. These also protect the device's internal circuitry from short circuits or transients that may appear on the RS-232 interface, and provide sufficient current to comply with the slew rate requirements for data transmission.
Because both ends of the RS-232 circuit depend on the ground pin being zero volts, problems will occur when connecting machinery and computers where the voltage between the ground pin on one end, and the ground pin on the other is not zero. This may also cause a hazardous ground loop.

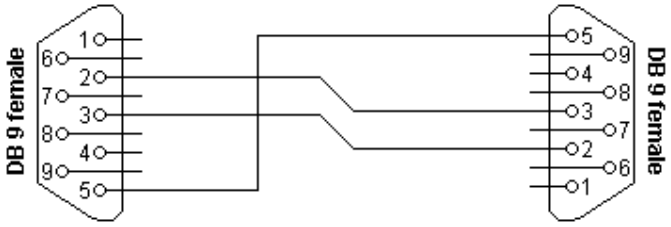
Unused interface signals terminated to ground will have an undefined logic state. Where it is necessary to permanently set a control signal to a defined state, it must be connected to a voltage source that asserts the logic 1 or logic 0 level. Some devices provide test voltages on their interface connectors for this purpose.

Pin Outs

RS232 DB9 Pin Allocation

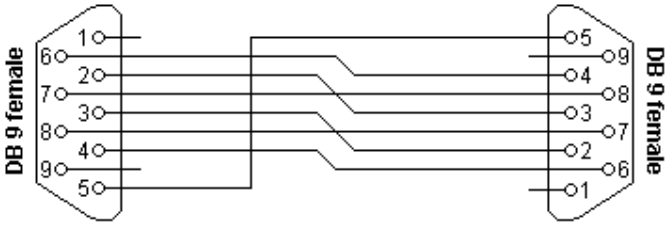


RS232 DB9 Null Modem Wiring (No Handshake)



Connector 1	Connector 2	Function
2	3	Rx ← Tx
3	2	Tx → Rx
5	5	Signal ground

RS232 DB9 Null Modem Wiring (Handshake)



Connector 1	Connector 2	Function
2	3	Rx ← Tx
3	2	Tx → Rx
4	6	DTR → DSR
5	5	Signal ground
6	4	DSR ← DTR
7	8	RTS → CTS
8	7	CTS ← RTS

General Note

Most equipment is configured as either DCE or DTE. DTE is normally a DB9Male. If the equipment you are connecting to also has a male connector, a cross over cable is required (refer Null Modem configuration above). For DTE to DCE connections (Male to Female connector) a pin to pin configuration is normally required. This “rule of thumb” is not always the case because some manufacturers have not followed the rules. Always check the equipment documentation before wiring the connectors.

Overview of ASCII plus a little bit of Hexadecimal

ASCII and HEX

Computers in their basic form represent everything as a number. The interpretation of this number as - for example text or an integer - depends on the context in which the number is being used. The Wikipedia definition for ASCII is given below:
“American Standard Code for Information Interchange (ASCII), is a coding standard that can be used for interchanging information, if the information is expressed mainly by the written form of English words. It is implemented as a character-encoding scheme based on the ordering of the English alphabet.
ASCII codes represent text in computers, communications equipment, and other devices that work with text. Most modern character-encoding schemes—which support many more characters than did the original—have a historical basis in ASCII. “
Some of the ASCII codes refer to Control characters – these Control characters indicate actions such as

- Line Feed (LF). Indicates that a line is to be skipped on the paper. Ther is a better analogy that refers to an old typewriter but the audience this document is targeted will probably not be able to visualise an old typewriter.
- Carriage Return (CR). Indicates to return to the beginning of a line.

When entering RS232 control strings, eZi Tools allows entry of the standard codes in either ASCII (just type the number or letter) or HEX or Hexadecimal. If you look at an ASCII table, you will see how numbers have been used to reference AlphaNumeric characters. Lets take an example – Look at small letter a in the table below – it has a number of 97 Dec (Dec refers to Decimal). Look for capital A – it has a value of 65 Decimal. It should now be clear that ASCII is simply a table that allows conversion between numbers and Alpha Numeric Characters and Control Codes.

Char	Dec	Oct	Hex	Char	Dec	Oct	Hex	Char	Dec	Oct	Hex	Char	Dec	Oct	Hex
(nul)	0	0000	0x00	(sp)	32	0040	0x20	@	64	0100	0x40	`	96	0140	0x60
(soh)	1	0001	0x01	!	33	0041	0x21	A	65	0101	0x41	a	97	0141	0x61
(stx)	2	0002	0x02	"	34	0042	0x22	B	66	0102	0x42	b	98	0142	0x62
(etx)	3	0003	0x03	#	35	0043	0x23	C	67	0103	0x43	c	99	0143	0x63
(eot)	4	0004	0x04	$	36	0044	0x24	D	68	0104	0x44	d	100	0144	0x64
(enq)	5	0005	0x05	%	37	0045	0x25	E	69	0105	0x45	e	101	0145	0x65
(ack)	6	0006	0x06	&	38	0046	0x26	F	70	0106	0x46	f	102	0146	0x66
(bel)	7	0007	0x07	'	39	0047	0x27	G	71	0107	0x47	g	103	0147	0x67
(bs)	8	0010	0x08	(	40	0050	0x28	H	72	0110	0x48	h	104	0150	0x68
(ht)	9	0011	0x09	)	41	0051	0x29	I	73	0111	0x49	i	105	0151	0x69
(nl)	10	0012	0x0a	*	42	0052	0x2a	J	74	0112	0x4a	j	106	0152	0x6a
(vt)	11	0013	0x0b	+	43	0053	0x2b	K	75	0113	0x4b	k	107	0153	0x6b
(np)	12	0014	0x0c	,	44	0054	0x2c	L	76	0114	0x4c	l	108	0154	0x6c
(cr)	13	0015	0x0d	-	45	0055	0x2d	M	77	0115	0x4d	m	109	0155	0x6d
(so)	14	0016	0x0e	.	46	0056	0x2e	N	78	0116	0x4e	n	110	0156	0x6e
(si)	15	0017	0x0f	/	47	0057	0x2f	O	79	0117	0x4f	o	111	0157	0x6f
(dle)	16	0020	0x10	0	48	0060	0x30	P	80	0120	0x50	p	112	0160	0x70
(dc1)	17	0021	0x11	1	49	0061	0x31	Q	81	0121	0x51	q	113	0161	0x71
(dc2)	18	0022	0x12	2	50	0062	0x32	R	82	0122	0x52	r	114	0162	0x72
(dc3)	19	0023	0x13	3	51	0063	0x33	S	83	0123	0x53	s	115	0163	0x73
(dc4)	20	0024	0x14	4	52	0064	0x34	T	84	0124	0x54	t	116	0164	0x74
(nak)	21	0025	0x15	5	53	0065	0x35	U	85	0125	0x55	u	117	0165	0x75
(syn)	22	0026	0x16	6	54	0066	0x36	V	86	0126	0x56	v	118	0166	0x76
(etb)	23	0027	0x17	7	55	0067	0x37	W	87	0127	0x57	w	119	0167	0x77
(can)	24	0030	0x18	8	56	0070	0x38	X	88	0130	0x58	x	120	0170	0x78
(em)	25	0031	0x19	9	57	0071	0x39	Y	89	0131	0x59	y	121	0171	0x79
(sub)	26	0032	0x1a	:	58	0072	0x3a	Z	90	0132	0x5a	z	122	0172	0x7a
(esc)	27	0033	0x1b	;	59	0073	0x3b	[	91	0133	0x5b	{	123	0173	0x7b
(fs)	28	0034	0x1c	<	60	0074	0x3c	\	92	0134	0x5c	|	124	0174	0x7c
(gs)	29	0035	0x1d	=	61	0075	0x3d	]	93	0135	0x5d	}	125	0175	0x7d
(rs)	30	0036	0x1e	>	62	0076	0x3e	^	94	0136	0x5e	~	126	0176	0x7e
(us)	31	0037	0x1f	?	63	0077	0x3f	_	95	0137	0x5f	(del)	127	0177	0x7f

The ASCII table has a column marked Oct (Octal) and Hex (Hexadecimal). These are number systems that work well with Binary which is the basic computer number system. Ignore Oct for this discussion and look at Hex. This is a number system that has 16 unique characters – Decimal has 10 – they are 0,1,2,3,4,5,6,7,8,9. The Hex numbers are 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F. Count them – there are 16.

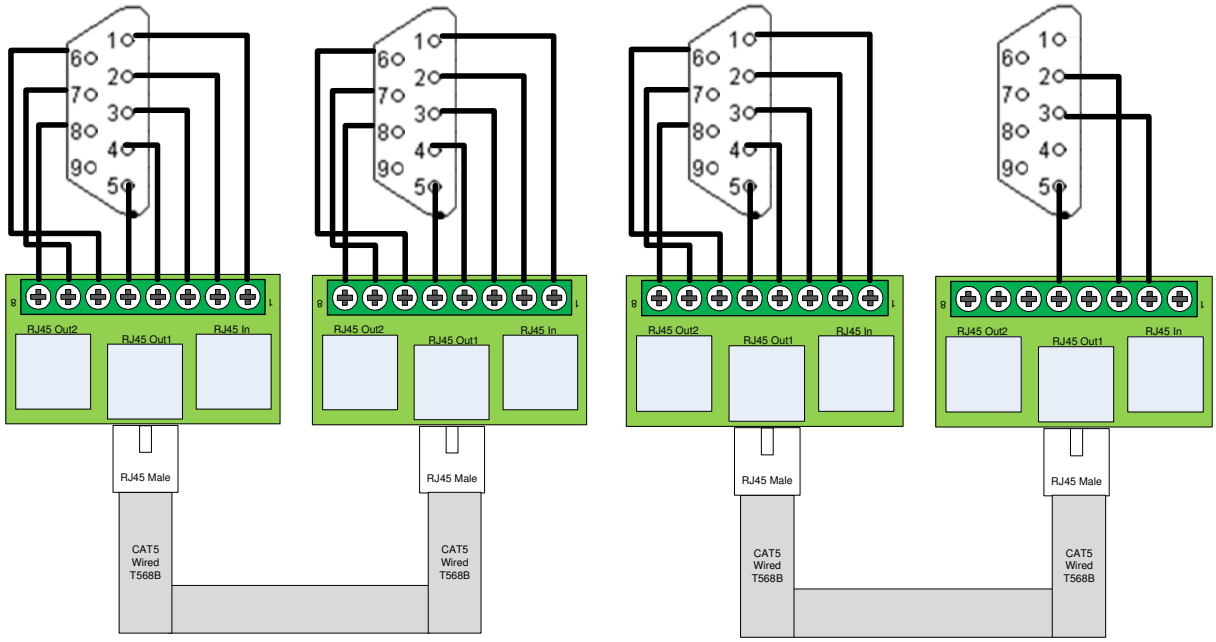
The only Hex you require for RAKO is 0D (Zero D). Look for 0D on the ASCII table above in the Hex column (Hint it is in the left column of the table). The value 0Dhex = 15decimal = CRascii.

The Command Properties in eZi Tools require a CR in ASCII or 0D hex to be entered to indicate to the MZC that you have completed the string the MZC must send. (Note – eZi Tools will not accept you typing CR in the ASCII column – you must move the cursor to the HEX row and type 0D.

Troubleshooting RS232

Make sure the Cables are correct

I have a set of RS232 cables connected to a BOT-X that allow me to connect any combination of 8 wires and make bridges when required to short out handshake lines. The little kit is shown below – this is a bit of a Heath Robinson creation but uses items from a speakercraft technicians toolbox. I have shown a fully configured DTE to DCE configuration and the more common Null Modem with no handshake. For your application make sure TX goes to RX, RX to TX and GND to GND by checking the literature of the devices to be connected



DTE to DCE fully wired cable
(DB9Female to DB9 Male)

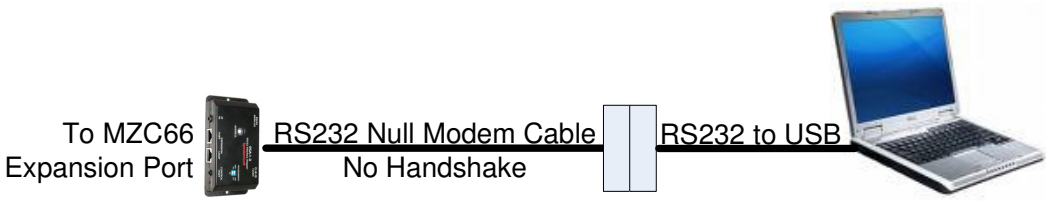
Null Modem – No Handshake
(DB9Female to DB9Female)

If you need to prepare a cable without the BOT-X, wire the DB9's as required by the manufacturers literature, following the rules above.

Test Communications using a Visual Tool - Hyperterminal

One of the most frustrating things about testing is not having the correct tools. RS232 can be tested using a software package on Windows/XP called hyperterminal – for those of you using Windows Vista, a trial version of Hyperterminal is available from http://www.hilgraeve.com/htpe/download.html . Download and install if required. Once Hyperterminal is running, select the serial port and communication parameters in Hyperterminal to suit the device the PC is connected to.

Example of connecting MZC66 to PC serial port



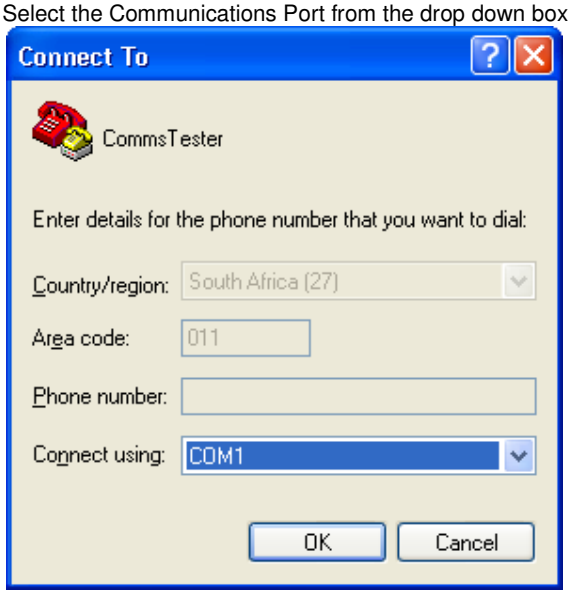
Running HyperTerminal to verify MZC Communications

Run HyperTerminal – From Windows Start Menu. Select All Programs – Accessories – Communications – HyperTerminal

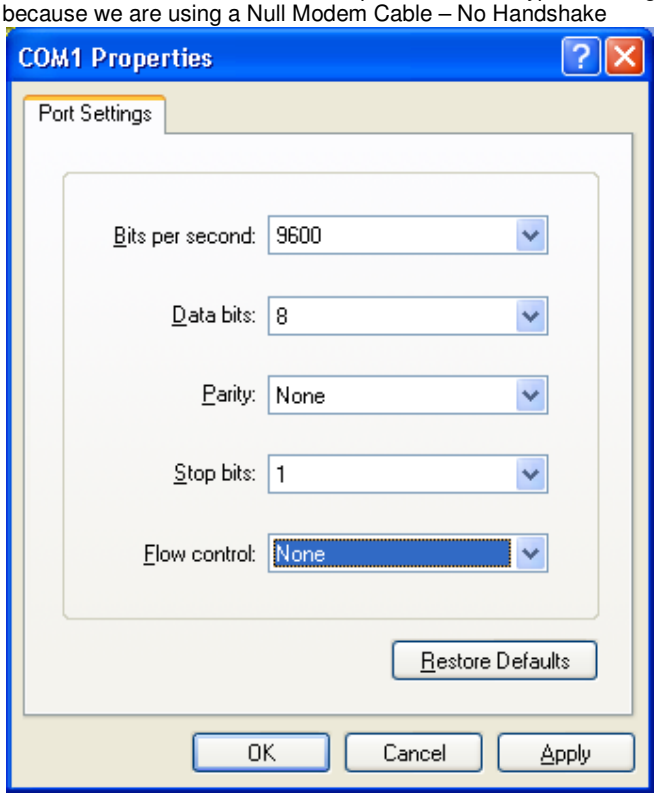
Type in the Name – Example - CommsTester

Select OK

Select the Communications Port from the drop down box

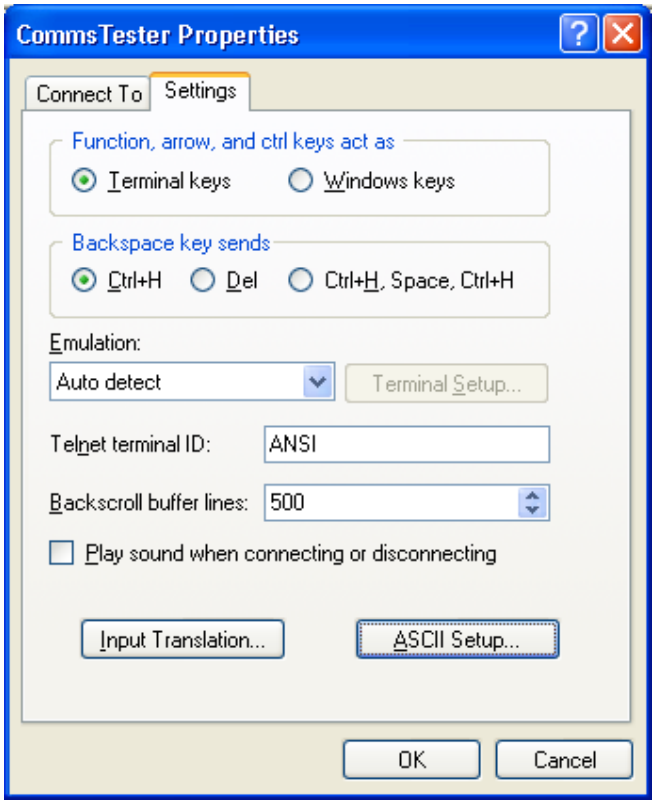


Select the communication interface parameters – A typical setting is shown. Please note the handshake is set to None because we are using a Null Modem Cable – No Handshake



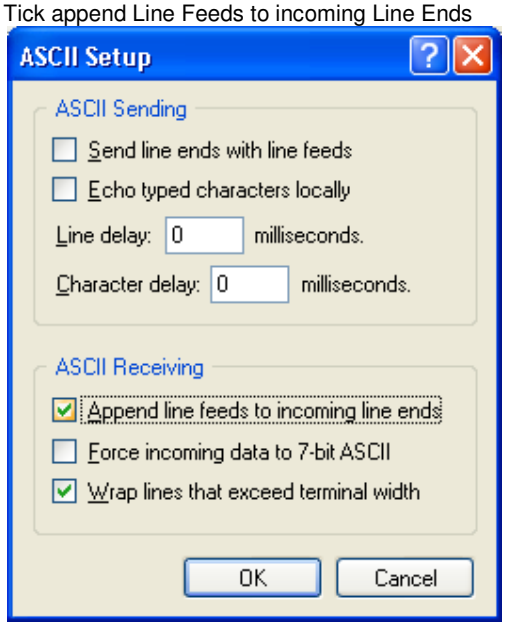
Select OK

To ensure that a line feed is appended to a carriage return, Select File - Properties from the menu. Select the Settings Tab



Select ASCII Setup

Tick append Line Feeds to incoming Line Ends



Select OK and then OK again to return to HyperTerminal main screen

The main screen will display whatever communications are received and as you type, the ASCII value will be transmitted. If the screen has characters that make no sense, verify that the communication parameters (excluding handshake) are correct.

MZC RS232 Connections

MZC88 serial port

The MZC88 has an on board serial port that does not support hardware handshake. A simple Null Modem no handshake cable will suffice.

RSA1-0 serial port

The RSA serial port will operate without handshaking. There is no setup for hardware handshaking in eZi-Tools therefore a simple Null Modem no handshake cable will suffice.

MZC66 serial port

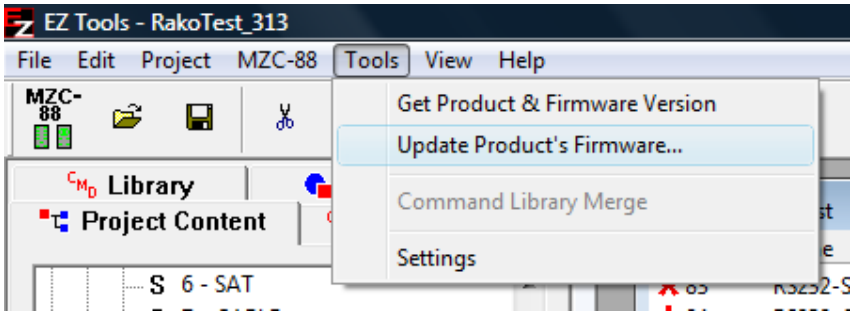
No serial port is supplied with the MZC66. Please add an additional RSA1-0 per additional serial port required.

RSA1-0 Firmware

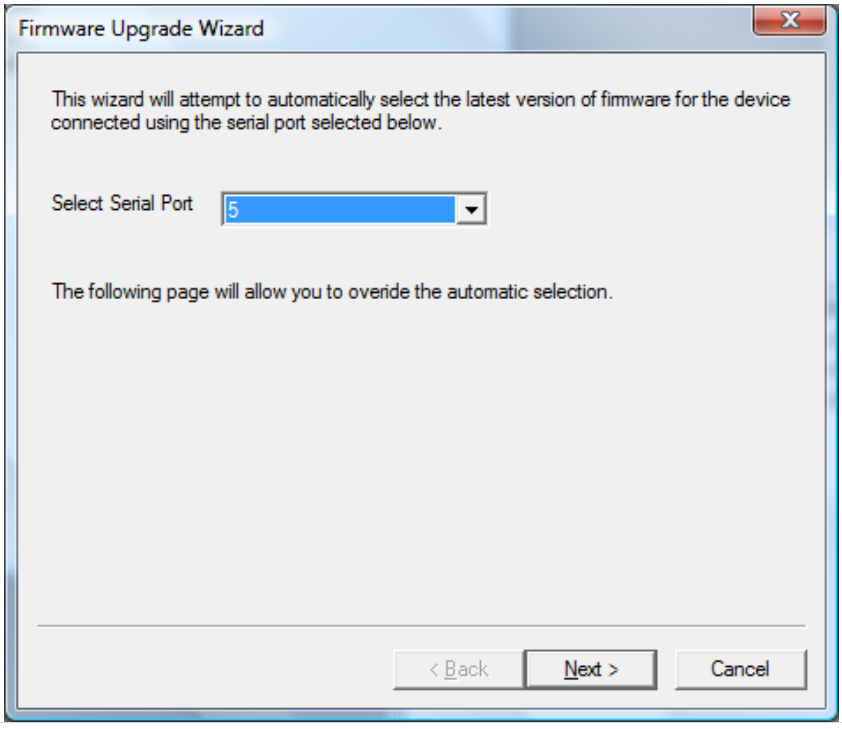
RSA1-0 Upgrade Procedure

- Unplug the RSA1-0 from IR , Expansion Port , Expansion Loop
- Plug PS1-0 into power socket and into RSA1-0
- Plug RS232 cable into the RSA1-0 Control Port and computer. PLEASE refer to #RSA1-0 Upgrade Problems for details on problems with some USB to RS232 Convertors before proceeding.
- Ensure the RSA1-0 Programming Switch is OFF

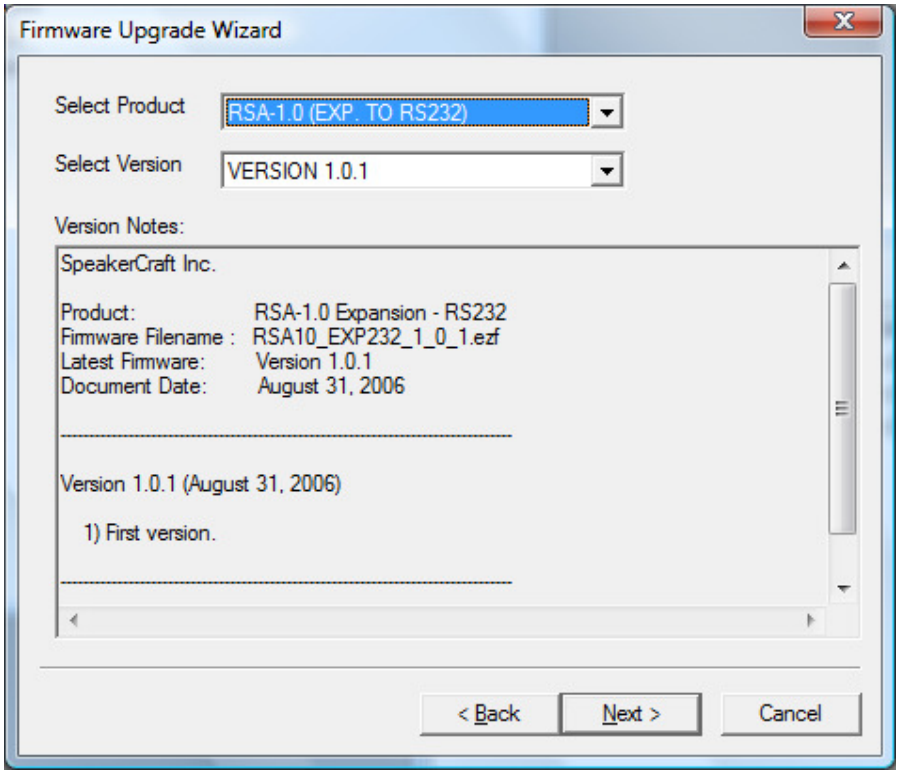
- Open EZ Tools software (Screen shots are from EZ Tools 3_13)
- Upgrade Firmware as shown on screen dumps



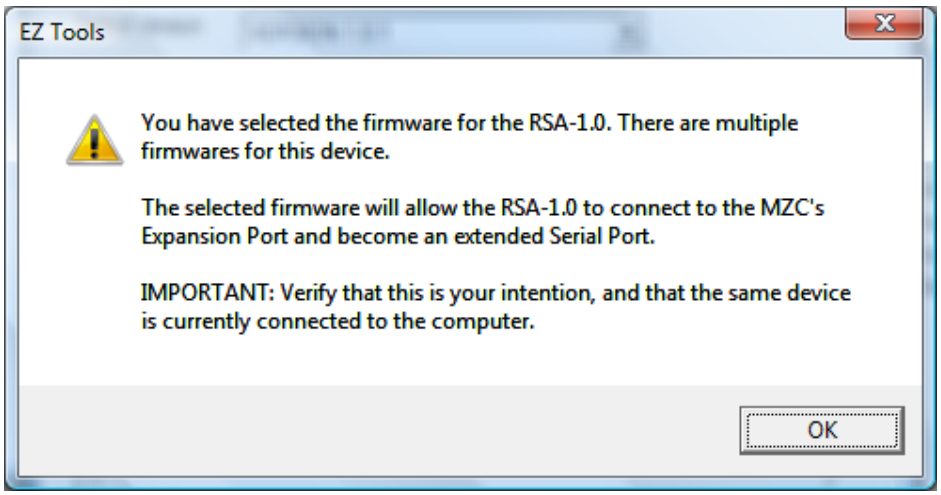
- You will be prompted to confirm the serial port – select Next



- The RSA1-0 will be detected and the current firmware revision displayed
- Select the firmware required from the drop down box. If a newer Version number is available than the one shown below select it. Refer to Firmware Note below
- Select Next



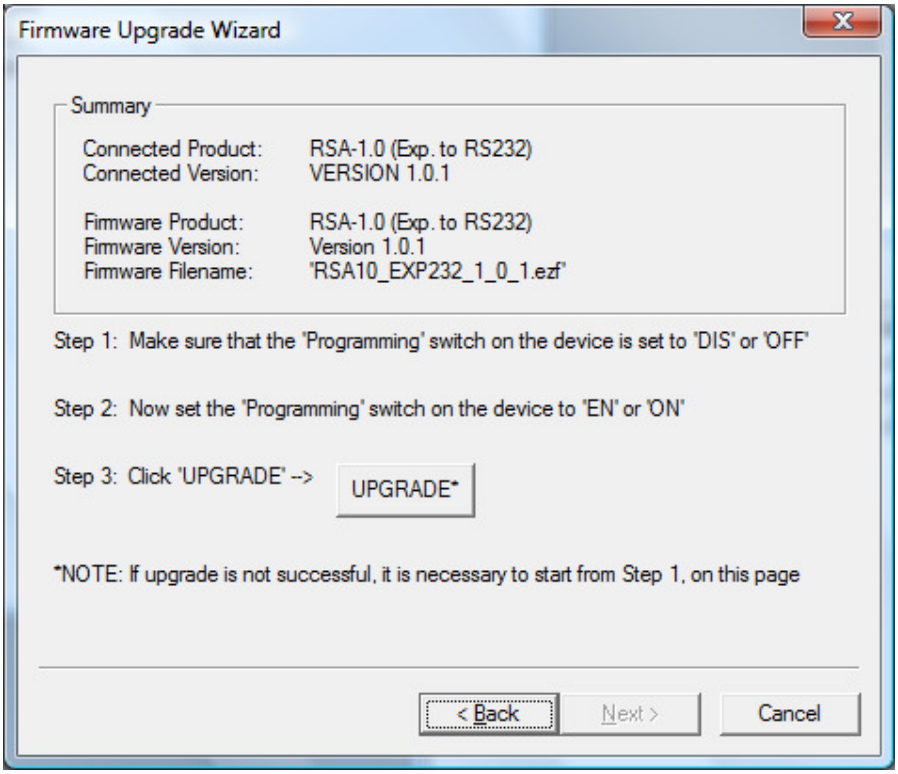
- Select OK



Firmware Note

- Use “EXP to RS232” firmware where the MZC is controlling an external device.
- Use “MZC CONTROL INTERFACE” where the external device is controlling the MZC and providing feedback.

- Put Programming Switch ON at the RSA1-0. Select Upgrade



- On successful completion of the upgrade process, select OK
- Put Programming switch on the RSA1-0 to OFF.
- Remove PSU1 (not required for normal operation)
- Connect external RS232 Device

RSA1-0 Upgrade Problems

The RSA1-0 firmware will fail the upgrade process if an external USB to RS232 converter is supplied. Only on board serial ports can be used for the upgrade process. This presents a problem for Technicians with new generation laptops. iLED will upgrade the RSA1-0 at no charge. Please plan this stage carefully to prevent last minute crisis management. Not all firmware revisions present this problem. The current list of known versions with this problem are

- EXP TO RS232 Ver 1.01

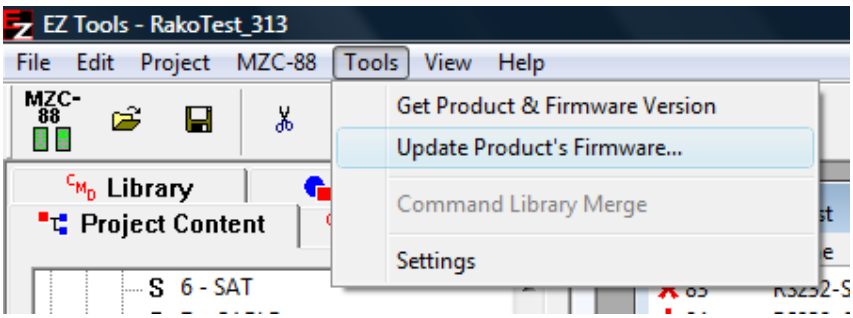
RSA1-0 Firmware Recovery

If for any reason the firmware upgrade fails, the device must be reset as follows. An indication of upgrade failure can be seen as follows

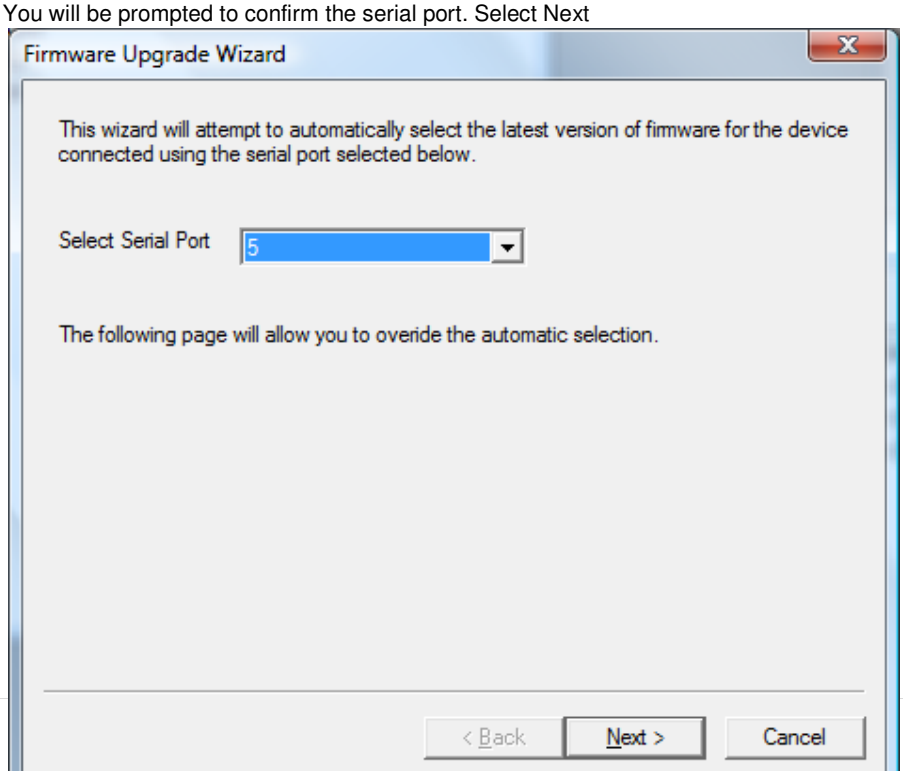
1. Put Programming Switch OFF
2. Ensure unit is powered using PSU1
3. Check Power LED (RED LED)
4. If the LED is OFF, the RSA1-0 needs to be re-initialised to recover the firmware

Re-Initialising the RSA1-0 to recover the firmware

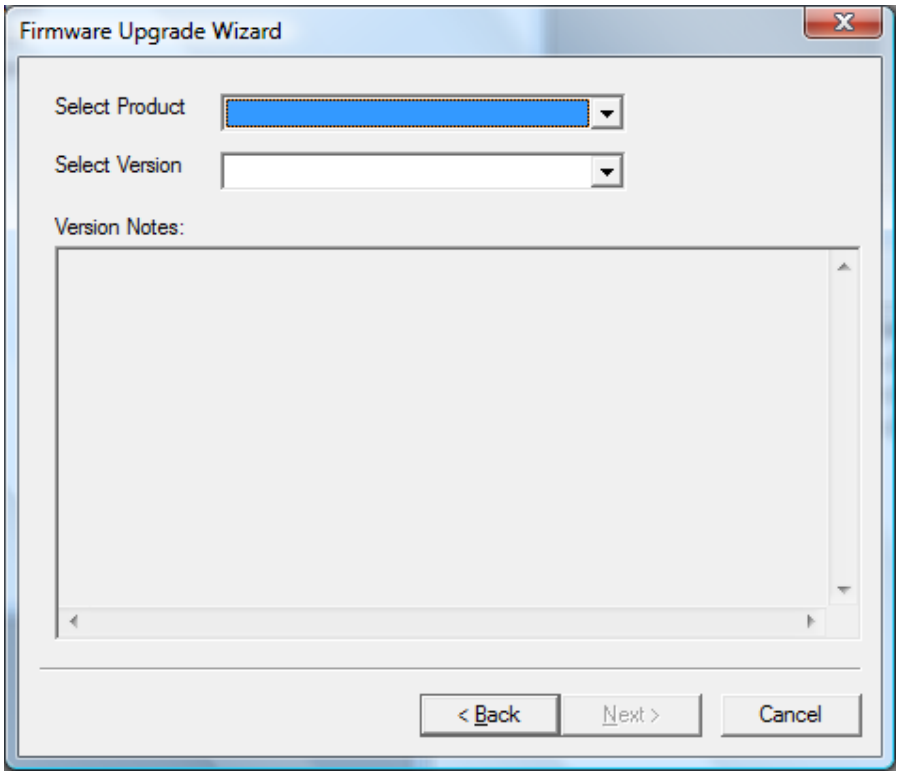
1. Use EZ Tools software



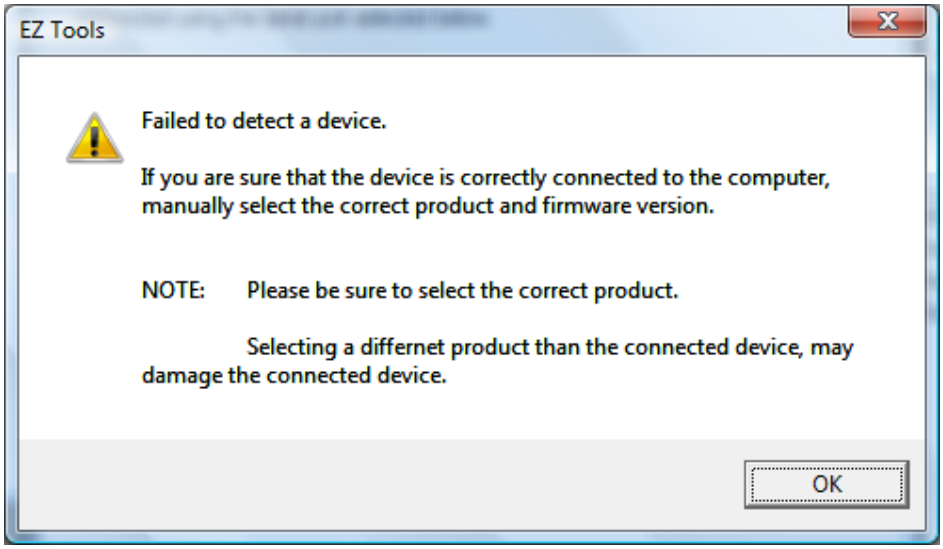
2. You will be prompted to confirm the serial port. Select Next



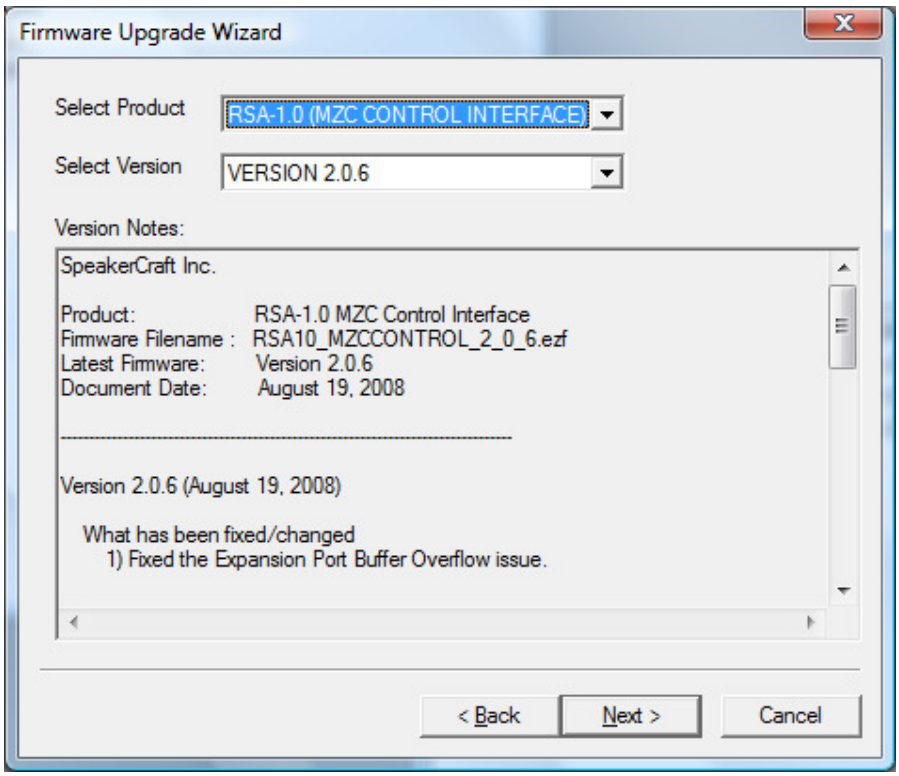
3. No device will be detected. Select OK



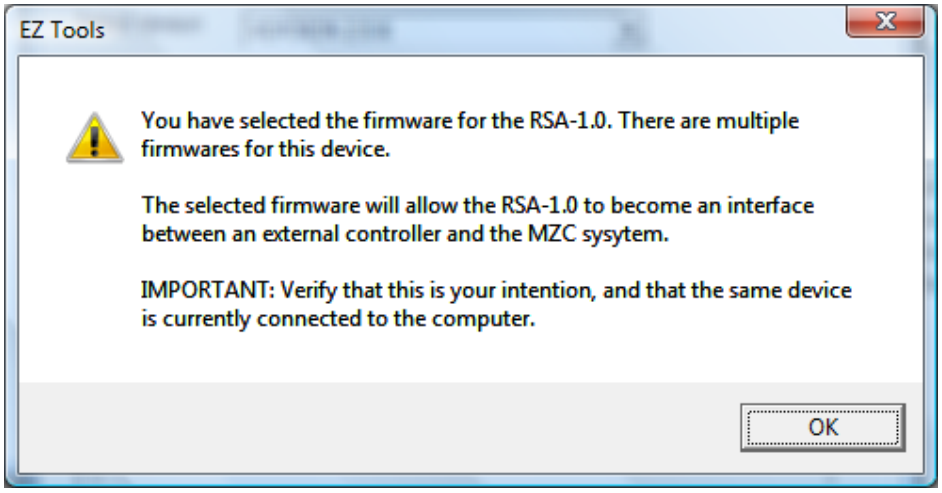
4. A blank device screen will be shown.



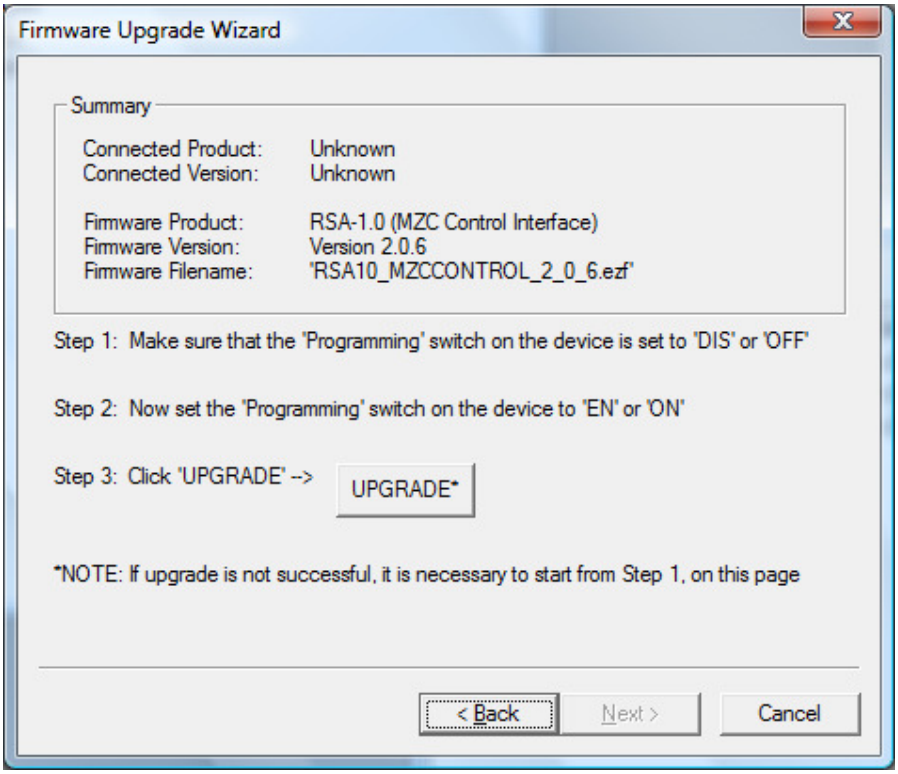
5. Use the drop down boxes to select RSA1-0. Select required MZC Control Interface (for explanation, refer to Firmware Note on pages above). Select Next



6. Select OK



7. Put the RSA1-0 Programming Switch to ON and select Upgrade.



This process will reset the RSA to factory default. Go back to #RSA1-0 Upgrade Procedure to reflash the firmware.

Product specific Tech Tips

Rako

The following has been tested with the RAKO RAV232 and RAV232+. The literature refers to the RAV232 only. This reference includes the RAV232+

RSA1-0 Firmware

RSA-1.0 (EXP to RS232)
Version 1.01

Cable

The cable supplied with the RAV232 plugs directly into the RSA1-0.

RS232 Commands

iLED has a library prepared for RAKO – please request a copy from our technical department. For clarity, I have included a summary of how the commands are structured. For information and training aids Speakercraft eZi Tools please request from iLED. The RS232 commands are covered very well in the Speakercraft Video Tutorials.

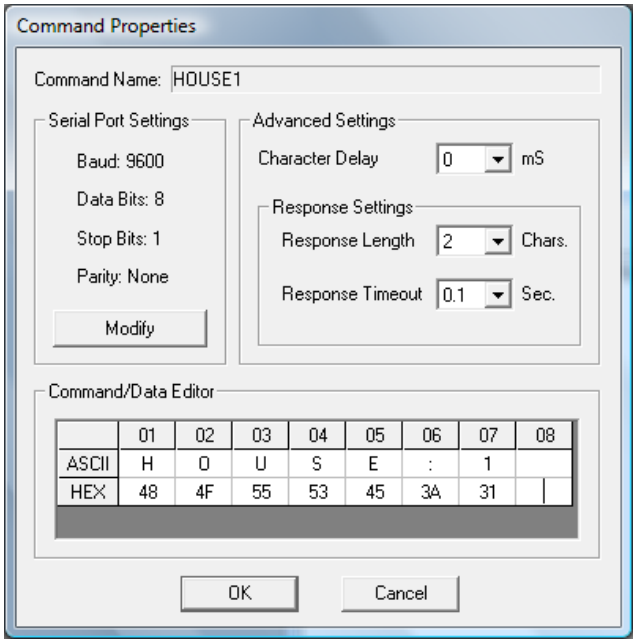
Please note following communication parameters (Refer to Serial Port settings shown below)

- RAV232
  - 1200baud, no parity, 8 data bits, 1 stop bit. No Flow Control – Factory Default
  - 9600baud from Rev1.4.1 – Factory Default
- RAV232+
  - 9600baud, no parity, 8 data bits, 1 stop bit. Select No Flow Control

The EZ Tools library has the communication parameters linked to a command – in other words, the baud rate etc are specified per command and NOT per interface. You will find a RAV232 model and a RAV232+ model with the libraries for the RAV232 and RAV232+ to accommodate this.

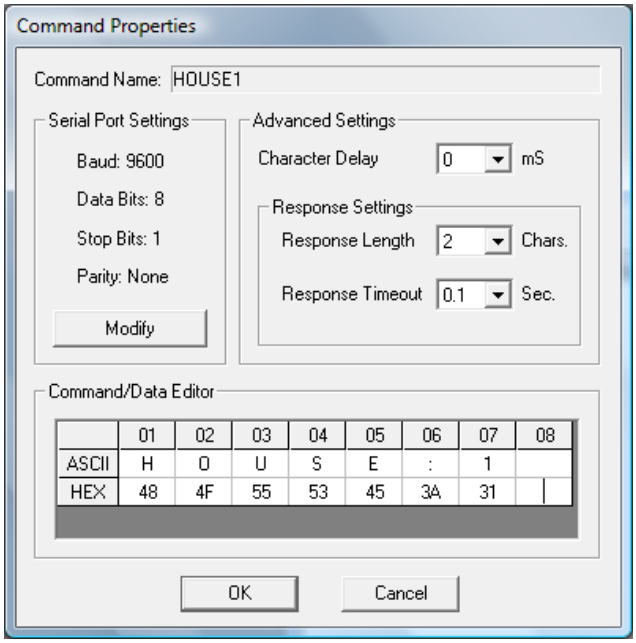
Refer to the RAKO RS232 Command Summary for a full list of commands. (An excerpt indicating the most common commands is included below.)

Please note that the RAV232 replies with OK – this is ignored by the RSA1-0 but needs to be taken into consideration when setting up each command. As shown below in the Response Settings box

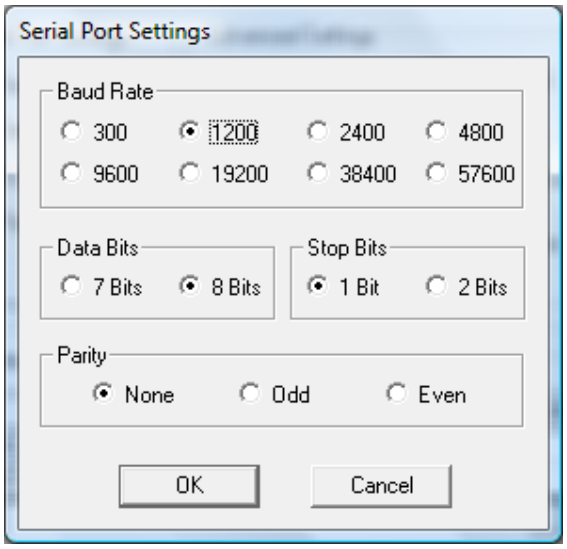


To use a RAKO RS232 command, build the command from the list of available commands in the library. I have given a few examples below.

To build your own commands in eZi Tools, Create a RAKO brand and add 2 models. RAV232 and RAV232+. Select the model (I have chosen RAV232) then select new command – RS232 Serial command. A command Properties window will show

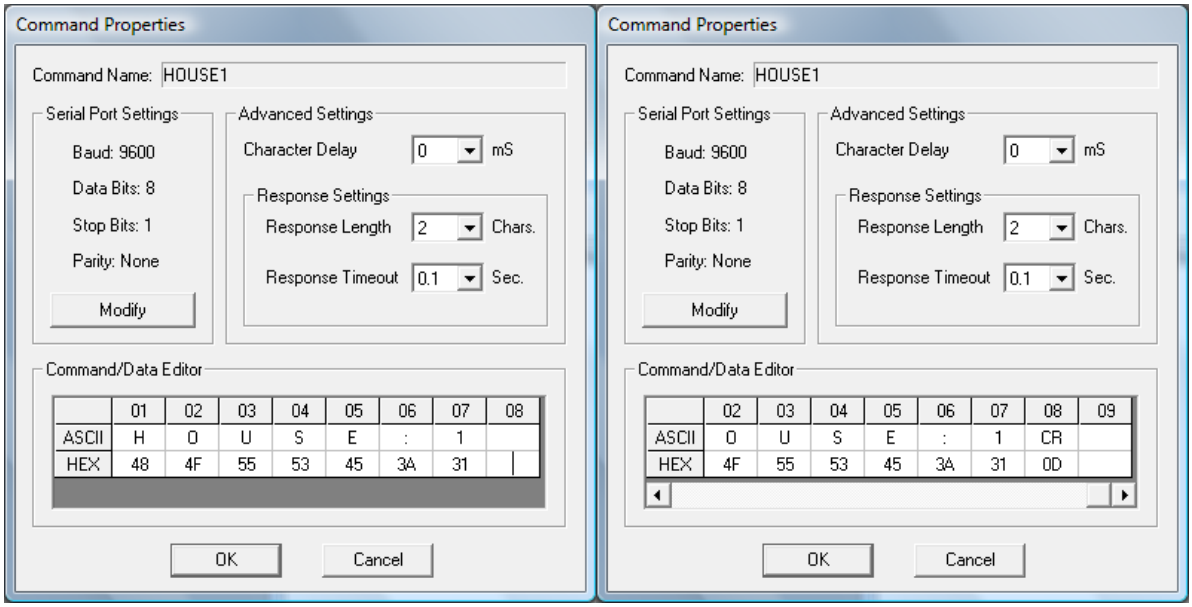


Click on Serial Port Settings to adjust to as shown below



Select OK to return to the command properties window.

Enter the command by selecting the first ASCII block (ASCII 01) – the example shows I have typed in H O U S E : 1. Move the cursor 1 row down using the down arrow key. Type in 0D (Zero D). CR will be displayed on the ASCII line

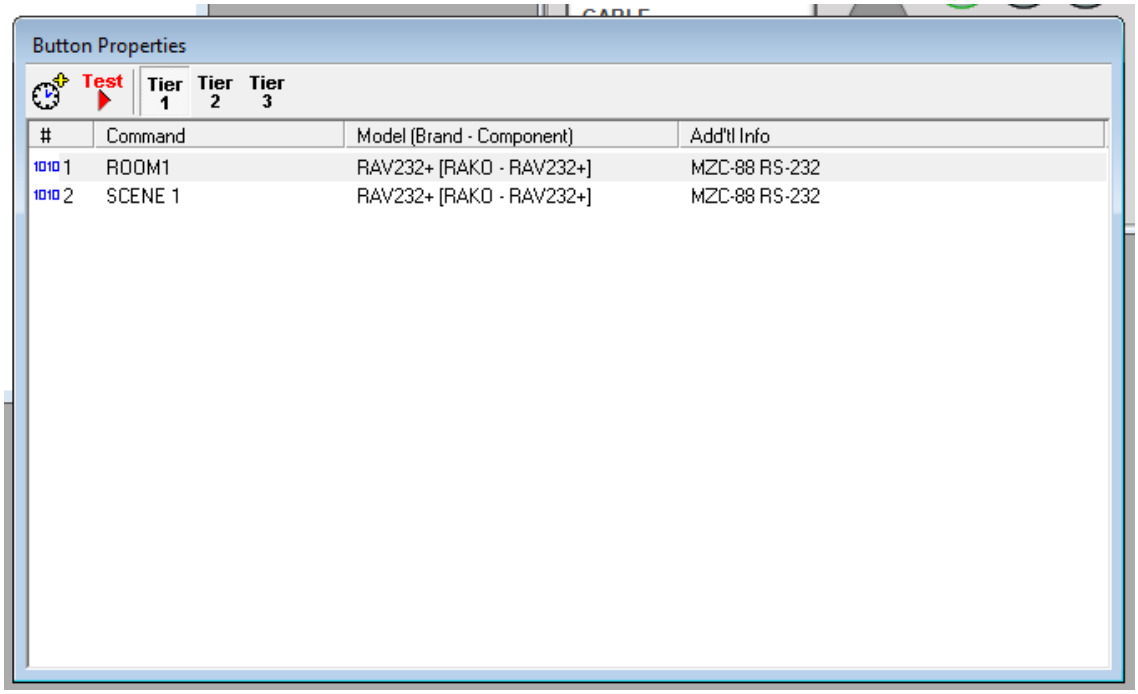


Select OK

The RS232 Serial Command is now complete.

To set the house number (only needs to be done the first time a RAV232 is used in a house) send the command HOUSE:1

To set the scene in a room, select the commands as follows from the library



RAKO Command List (excerpt)

Command	Notes
HOUSE <house_number>	Sets the HOUSE address to <house_number>. The House number must be between 1 and 255. The House number is stored in non-volatile memory
ROOM [<room_number>]	Sets the ROOM address to <room_number>. The Room number must be between 0 and 255. Room 0 controls all units with the same House address. If <room_number> is omitted, the room is set to 0. The Room address is stored in non-volatile memory.
CHANNEL [<Channel_Number>]	Selects the Channel address. The <channel_number> must be between 0 and 15. Channel 0 controls all channels within the current Room. If <Channel_Number> is omitted the channel is set to 0.
SCENE <scene number>	Sets the scene for the current
	House/Room/Channel. The <scene number>
	must be between 1 and 4, which correspond to the
	buttons on a control panel.
OFF	Turns off the lights in the current House/Room/Channel.
LEVEL <Power_level>	Sets the power level for the current House/Room/Channel. The <power_level> must be between 0 and 255, with 255 representing 100% power.
STORE	Stores the current power level to the current Scene. This will only apply to dimmers addressed by the current House/Room/Channe
RESET	Resets the microcontroller.
VER	Displays version information
STATUS	Displays current House, Room and Channel in the form: HO:nnn RO:nnn CH:nnn / nnn is 3 digit decimal number with leading zeros. / nnn is between 0 and 255.
COMMAND [<command_number>]	Issues <command_number> to the lights in the current House/Room/Channel. The <command_number> must be between 0 and 15.
ADDRESS <EEPROM_address>	Sets the EEPROM address within the dimmers. The (EEPROM_address> must be between 0 and 127
DATA <EEPROM_data>	Transmits <EEPROM_data> to the address set using the ADDRESS command. The <EEPROM_data> must be between 0 and 255.
NOECHO	Turns off character echoing. This command is only available on the Bi-directional RS232 interface. The current echo mode is stored in non- volatile memory.
ECHO	Turns on character echoing. This command is only available on the Bi-directional RS232 interface. The current echo mode is stored in non- volatile memory.

Running HyperTerminal to verify RAKO Communications

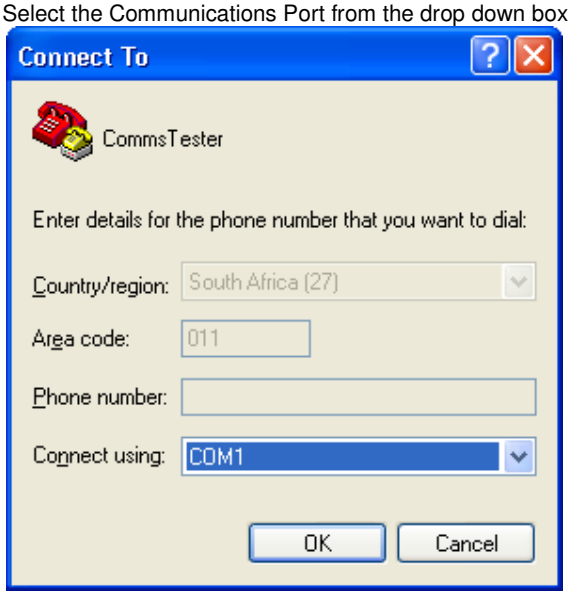
Connect the RS232 Female DB9 supplied with the RAV232 to the RS232 port on the PC or the RS232 to USB convertor.

Run HyperTerminal – From Windows Start Menu. Select All Programs – Accessories – Communications – HyperTerminal

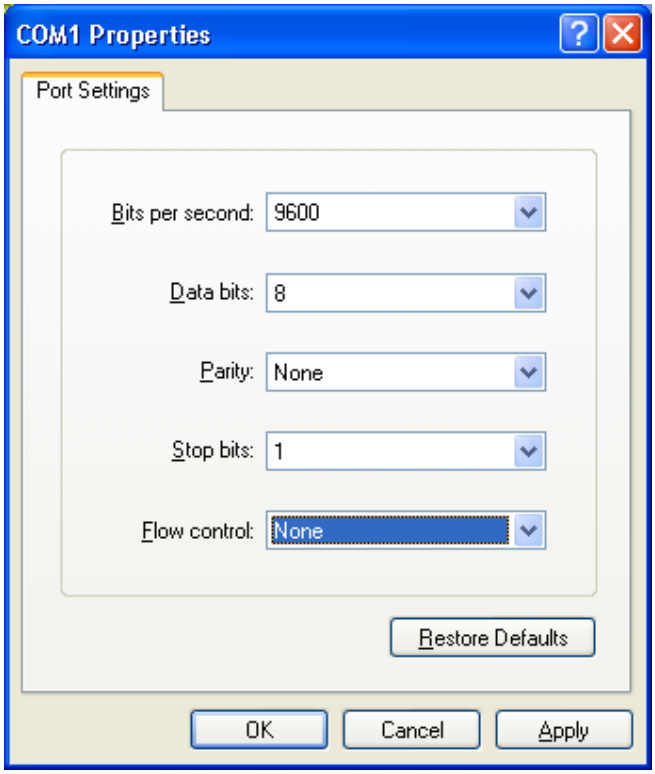
Type in the Name – Example - CommsTester

Select OK

Select the Communications Port from the drop down box

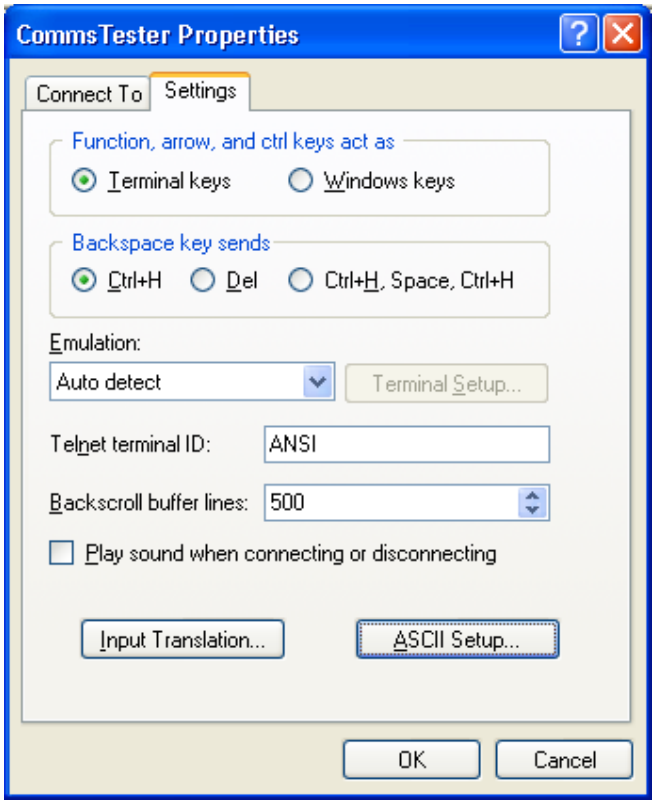


Select the communication interface parameters – A typical setting is shown – this is as per factory setting for the RAV232 .



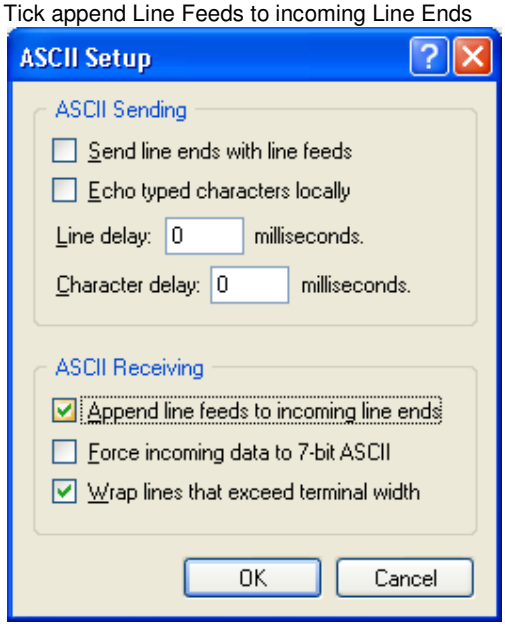
Select OK

To ensure that a line feed is appended to a carriage return, Select File - Properties from the menu. Select the Settings Tab



Select ASCII Setup

Tick append Line Feeds to incoming Line Ends



Select OK and then OK again to return to HyperTerminal main screen

Type in the RAKO commands from the RAKO command List above and verify that the lighting zones respond as required.

After each press of the Enter key following a command, the RAV232 will respond. This is normally in the form of OK. If you do not get this, no communications is taking place. Check the cables and communication parameters.

SmartBus

The following has been tested with SmartBus ASCII Translator.

RSA1-0 Firmware

RSA-1.0 (MZC CONTROL INTERFACE)
Version 1.01

Cable

The cable supplied with the SmartBus ASCII Translator plugs directly into the RSA1-0.

RS232 Commands

To be Finalised