

RS-232 to RS-485 Protocol Conversion

with the 2504 and 2510 Controllers

Introduction

Both the 2504 and 2510 playback controllers may be used as protocol converters, converting standard RS-232 into the RS-485 protocol that Martin uses to control lighting fixtures. The RS-485 has several advantages over the RS-232 protocol: not only does it work at higher transmission rates (187.5 Kbaud), it also specifies a balanced link, which means that several units may be connected in a daisy chain.

Connections

To connect a 2510 controller to a computer RS-232 port, use the 9-pin to 25-pin cable supplied with the controller. This cable is not supplied with the 2504 controller but may be ordered from your Martin dealer (P/N 309129), or alternatively you may make one yourself.

25-pin PC output to 9-pin input

- Connect pin 2 on PC to pin 3 on controller.
- Connect pin 7 on PC to pin 5 on controller.

9-pin PC output to 9-pin 2510 input

- Connect pin 3 on PC to pin 3 on controller.
- Connect pin 5 on PC to pin 5 on controller.

Enabling protocol conversion mode

Playback mode is the default on both controllers. The procedure for enabling protocol conversion is a little different on the 2504 and the 2510.

2504 Controller: Do the following each time the 2504 is switched on.

1. Press the [↑] key until PC is shown in the display.
2. Press [Enter].

2510 Controller: Since this controller has programmable memory, it is possible to set PC mode as the default start-up mode. To do this:

1. Press the [↑] key while switching on the 2510. The display will show PC to indicate that protocol conversion mode is active. This will be the default mode every time the 2510 is switched on.
2. To disable protocol conversion mode, just press the [↑] when switching on, and the 2510 returns to playback mode.

Using protocol conversion

As mentioned above, the conversion translates RS-232 signals into the RS-485 protocol that Martin uses. RS-232 communication is strictly one-way and there is no handshake from the 2504/2510 controller back to the source.

The RS-232 protocol used is 8 data bits, 1 stop bit, no parity. The controller is initiated to a baud rate of 9600. Once in conversion mode it is possible to switch the baud rate up to a maximum of 57600 baud.

The RS-232 data sent to the controller must **ALWAYS** be in the following 3 byte format: ADDRESS, MSB, LSB.

As there is no feedback, it is up to the RS-232 source to maintain the 3 byte synchronization. To ensure this, there must be a period of at least 250 microseconds from the end of one 3-byte packet to the start of the next 3-byte packet. This period must be from the **stop bit** of the last byte in a packet to the **start bit** of the first byte in the next packet. Most constructions including the PC have a UART to clock out the individual bits of a byte, leaving the CPU to perform other operations, so remember to take the time used to clock out the byte into account (10 bit at the relevant baud rate).

The Address, MSB, LSB format primarily refers to the protocol used to control the fixtures, but the addresses 248-255 are set aside for communicating with the controller:

Address	MSB	LSB	Description
255	255	001	Enable protocol conversion
255	255	002	Disable protocol conversion
255	255	003	Set receive baud rate to 9600 baud
255	255	004	Set receive baud rate to 19200 baud
255	255	005	Set receive baud rate to 38400 baud
255	255	006	Set receive baud rate to 57600 baud

When the receive rate is changed, **all** subsequent data must be sent at the new baud rate.

Controlling fixtures

Controlling fixtures is done by transmitting a 2 byte data value to a certain address. The base address for each fixture is calculated as $\text{Address} = 8 * (\text{dipswitch value} - 1)$. A fixture at dipswitch address 11 has a base address of $8 * (11 - 1) = 80$.

Each fixture is then subdivided in to 8 addresses if it uses 1 Martin channel, or 16 addresses if it uses 2. A RoboScan 812 (1 Martin channel) at dipswitch address 1 has base address 0 and covers addresses 0 to 7. Note that dipswitch address 32 is not usable as addresses 248-255 are used to send instructions to the controller.

Effects are controlled by sending a MSB, LSB data value to each of these addresses. For example, to set the gobo wheel of a RoboScan 812 to the third gobo, the command is

“ADR,30,0” where ADR is equal to the base address + the offset for the gobo wheel (refer to the RS-485 protocol for the fixture).

The MSB, LSB data bytes usually control different effects or speeds, so both the MSB and LSB values must be set to the desired values.

The order in which you update fixtures or effects within the fixture is not important. In addition, the fixtures do not require data refresh, so you only need to transmit new data values.

Some fixtures may have unassigned offsets, in which case you do not need to send to these addresses. Some older fixtures may not need LSB data, but due to the 3 byte format of the conversion, you must send both MSB and LSB to the 2504/2510. Just send a 0 as LSB.

Note that when most fixtures are resetting, either after power up or after a reset command, they will not listen to serial transmissions. With most fixtures you may need to re-establish the effect values after the reset.

Martin fixtures that are capable of running directly off a DMX controller have a built in auto-detection routine to determine the transmission protocol used. They need to sample the transmission for about 1.5 seconds before they respond to the Martin protocol. To make sure that the fixture has detected the Martin protocol, send at least one transmission and then wait 1.5 seconds before transmitting to the fixture.