

## The Emulator II RS422 Computer Interface

### Introduction

The Emulator II RS-422 computer interface is a powerful low-level interface to the Internal data structures of the Emulator II (EII) microprocessor. While the interface was specifically intended to work with the Apple Macintosh computer, it will work with any computer that has a asynchronous serial port capable of being externally clocked at 500 kBaud.

The Emulator II uses a 500-kBaud asynchronous serial interface to exchange data with external computers. The EII provides a 500 kHz clock.

A MIDI system exclusive command can be used to put the Emulator under control of the external computer, but otherwise MIDI is not used in this interface.

A low-overhead packet oriented communications protocol with block parity and retransmission of corrupted blocks is used.

The protocol allows any block of memory in the EII to be read or written, and a set of subroutines can be remotely executed to perform various EII functions. The EII is the slave in all external computer transactions.

The addresses of EII voices in sound data memory are stored in a Voice Table, which resides at a fixed address. The detailed RS422 interface specification will contain the memory map of the important EII data structures, allowing an experienced programmer to access sound memory and control data structures.

It is assumed that the potential user of this document has a great deal of knowledge of microprocessors, assembly language programming, hexadecimal arithmetic, serial interface programming, and some knowledge of the Emulator II. It is not intended to be used by people new to computers. Unless otherwise stated, this document is provided "as is", with the understanding that E-mu will not provide any support to the user in programming or consultation in using it (you're on your own!).

This document is still under development, and will include more specific details later. This original specification provides information about the physical interface, and about the software communication protocol. Further information about the Emulator II internal data structures will be provided later.

## Level 0 Physical Hardware Specification

Although the Emulator II uses a 25 pin DB25 connector for its computer interface, which does not comply with the physical specification for RS422, signal levels are compatible with RS422. The pin out is given in table 1. Hardware handshaking is not used on the E-II. The EII MIDI port is shared with the RS422 port. Sending or receiving data on the RS422 port at MIDI Baud rate (31.25 k) is equivalent to sending or receiving data on the MIDI port.

This interface requires the Emulator II to have RS-422 capability. If the serial number of the Emulator is below about 900, then it probably has the Revision 0 digital board, requiring a small retrofit board to provide the EII with RS-422. The digital board is the large board on the right inside of the Emulator II. The revision level is on the lower right hand corner of the Digital board. The retrofit board can be obtained from E-mu Systems or from an E-mu service center.

Pin #	Signal Name	Signal Voltage	EII Input/Output
2	-Receive Data	Nominal $\pm 4V$ Absolute maximum $\pm 15V$	EII Input
3	-Transmit Data	0 to 4 V	EII Output
7	Signal Ground	Ground	
9	500 kHz Clock	-1.5 V to +2.5 V	EII Output
10	+Receive Data	Nominal $\pm 4V$ Absolute maximum $\pm 15V$	EII Input
11	+Transmit Data	0 to 4 V	EII Output

**Table 1: RS422 Connector Pin-out**

The Emulator II is compatible with normal RS232. RS232 is defined to use voltage levels of  $\pm 12 V$ , while the E-II outputs are 0 to +4 V. Zero volts is defined to default to a negative (marking) state in RS232, and a signal above 3 Volts at the receiver should produce a high (space) condition. In practice, RS232 works well with the EII. The transmit and receive data are differential, but unipolar, while the clock line is bipolar but not differential. This arrangement is due to the RS422 signals available on the Macintosh. The clock can be converted to a single-ended clock line by adding a jumper across capacitor C162 on the Rev 1 digital board or capacitor C1 the RS-422 retrofit board.

Standard RS232 drivers will not, however, clock at 500kBaud. In order to communicate at 500kBaud, the user's computer must have RS422 or equivalent drivers.

## Level 1 Serial Data Specification

Serial data are transmitted at 500,000 bits per second (Baud), asynchronous, with 8 bits per character, 1 stop bit, 1 start bit, Odd Parity. The 2651 USART used in the EII is clocked at the X1 clock rate.

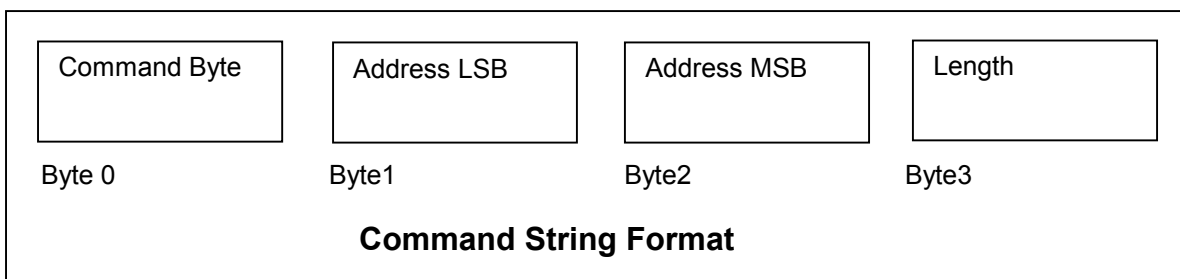
## Level 2 Software Command Protocol

The Emulator II can be put into Mac Mode (external computer control) either by using special function 16 or by a MIDI system exclusive command (MIDI communications are of course at 31.25 kBaud). Once under control of the external computer (now at 500kBaud), the EII executes a simple command loop that insures synchronization between the EII and the external computer. This command loop is shown in block diagram form in Figure 2.

The MIDI System Exclusive command follows the standard format for MIDI system exclusive;

byte 1 (enter system exclusive mode) FOh,  
byte 2 (E-mu Manufacturer's ID. #) 17h,  
byte 3 (enter Mac Mode command) 00h,  
byte 4 (end of system exclusive mode) F7h.

The basic goal of the command loop is to send or receive a command string consisting of four bytes; the format is shown in figure 3. The command string always starts with a single byte command, a two-byte address, and a single byte count or length argument. The command string must always be four bytes long, even if the arguments (2nd and 4th bytes) are dummy arguments. The commands available are listed in table 2. Protocol bytes are listed in table 3.



Command Byte (Hex)	Command	Comments
F9	Call subroutine	Used to call a subroutine; First byte of Command String Address is subroutine number.
FA	Input from I/O port output	First byte of Command String Address is input port address. Command String Length is the actual value.
FB	Output to I/O port	First byte of Command String Address is output port address. Command String Length is the actual value output.
FC	Init	Initialize command sequence
FD	Exit	Exit external computer control
FE	Send Packet	Used to read data from The EII. Command string address is destination in EII for data.
FF	Receive Packet	Used to load data into the EII from the external computer. Command String Address is source of data.

**Table 2: Commands**

Protocol Byte Name	Protocol Byte (ASCII)
ACK	'A'
NAK	'N'
RAK	'R'
Init ACK	'I'
STX	'S'

**Table 3: External Computer Interface Protocol Bytes**

Subroutines can be called from the external computer interface. The subroutine number is passed as the first byte of the command string address. Parameters are passed through a register save area, which is loaded into the actual registers when the subroutine is executed.