

E1 and E2 Brain Boards for Ethernet-to-Serial Routing

Introduction

Opto 22 E1 and E2 brain boards can be used to route Optomux commands between Ethernet and serial (RS-422/485) networks. Routing from a serial host to Ethernet Optomux units (serial-to-Ethernet routing) is covered in form #1563, the *E1 and E2 User's Guide*. This tech note covers routing from an Ethernet host to Optomux units on a serial network (Ethernet-to-serial routing).

In Ethernet-to-serial routing, one E1 or E2 brain board is wired as a master and set up for Ethernet-to-serial routing. A host on an Ethernet network sends Optomux commands through the master E1 or E2 to other devices on the serial network. The host can communicate with up to 255 Optomux devices.

When the master E1 or E2 brain board receives an Optomux command, it compares the address contained in the command to its own Optomux address. If the addresses are the same, the brain board executes the command on its own I/O and generates a response.

If the addresses are different, the brain board forwards the command out the serial port. No additional received commands are routed until either the response is received via the serial port, or the master brain board times out waiting for the response.

When the master E1 or E2 receives the response, it is sent to the IP address and UDP port from which the command came. Unlike serial-to-Ethernet routing, Ethernet-to-serial routing does not require a routing table, because the brain board simply routes all commands received from the Ethernet port (except for commands intended for the E1 or E2 itself) out the serial port, and it routes the Optomux response packet received from the serial port back via Ethernet/UDP.

If Ethernet-to-serial routing is *not* enabled, the E1 or E2 brain board assumes that all commands received through the Ethernet port are for it and responds locally. In this case, the E1 or E2 checks the destination address in the Optomux command only if the command is received from the serial port.

CAUTION: *Do not enable Ethernet-to-serial routing if there is another host on the RS-422/485 link, because the brain board acts as a host when Ethernet-to-serial routing is enabled. Also, do not enable both Ethernet-to-serial routing and serial-to-Ethernet routing on the same E1 or E2 at the same time.*

Setting Up Ethernet-to-Serial Routing

To use Ethernet-to-serial routing, follow the steps below to change the OmuxSettings file, set the jumpers, and wire the brain boards.

Changing the OmuxSettings File

1. Follow the steps in Appendix B of form #1563, the *E1 and E2 User's Guide*, to add or change the following items in the OmuxSettings file:

To do this	Add/change/delete this text	Details
Enable listening on the Ethernet link	Add: <code>EthernetServer</code>	In by default
Enable listening on the serial link	Add: <code>SerialServer</code>	In by default
Enable Ethernet-to-serial routing	Add: <code>EthernetToSerialRouting</code> Remove: <code>SerialServer</code>	This text must exist in the OmuxSettings file. Otherwise Ethernet-to-serial routing is disabled.
(optional) Change how long the E1 or E2 waits for a response before timing out*	Change: <code>EthernetToSerialTimeout=#</code>	Used with Ethernet-to-serial routing only. Value is in tenths of a second. Default is 5 (0.5 sec).
(optional) Add or change how many times the E1 or E2 retries communication on the serial link**	Add or change: <code>EthernetToSerialRetries=#</code>	Used with Ethernet-to-serial routing only. Value is number of retries. Default is zero, meaning communication is tried only once and not retried.

* The timeout value used by the host application should be longer than the combined timeout/retries period of the E1 or E2 doing the Ethernet to serial routing.

** Normally the host application handles retries. To do this, set the retries on the host application and leave retries on the E1 or E2 set to zero (the default).

2. As described in Appendix B of form #1563, save the OmuxSettings file and upload it to the master E1 or E2 using PAC Manager.

Setting the Jumpers

On the master E1 or E2, set jumpers 0 and 6 in Group A as shown at right.

The master brain board can be placed at either end of the serial link; setting these jumpers terminates the serial link. The slave brain boards may be set to either



repeat or multidrop mode. For more information on setting jumpers, see the installation instructions in form #1563, the *E1 and E2 User's Guide*.

Wiring the Brain Boards

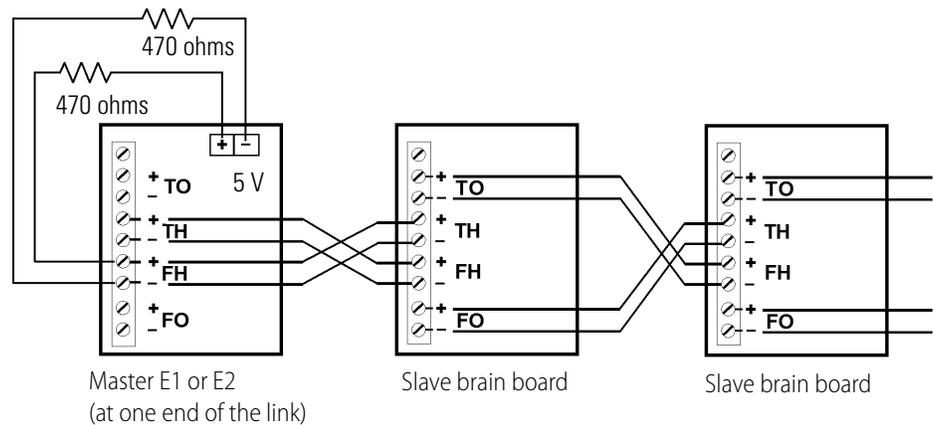
The following graphic shows a master E1 or E2 wired to two slave brain boards. When wiring a series of slave brain boards, always think of the previous brain board as the host.

NOTE: The master and slave brain boards must be wired this way for Ethernet-to-serial routing. It is different from all other Optomux wiring.

To properly bias the RS-422/485 lines, you must add two external 470 ohm resistors:

1. Connect one resistor between the FH+ terminal and the 5V+ terminal.
2. Connect the other resistor between the FH- terminal and the 5V- terminal.

These can be standard 1/4 watt resistors. It would be best to solder braided wire to each end of the resistors and insulate the exposed leads. Then screw the braided wire into the terminals (as opposed to inserting the solid resistor leads into the terminals).



Disabling Ethernet-to-Serial Routing

To stop Ethernet-to-serial routing:

1. Open the OmuxSettings file.
2. Remove the text: `EthernetToSerialRouting`
Add the text: `SerialServer`
3. Save the file.
4. Use PAC Manager to upload the revised OmuxSettings file to the E1 or E2. For more information, see Appendix B of form #1563, the *E1 and E2 User's Guide*.

Using Multiple Hosts

IMPORTANT: *The E1 and E2 Ethernet capability makes it technically possible for multiple host programs to access the Optomux network. In a multiple-host network using Optomux. However, there should be only one host program controlling the I/O. You can use other host programs to monitor the I/O.*

Optomux is a command/response protocol; in order to avoid a collision on a serial line, the master E1 or E2 places all commands from the hosts in a queue and makes sure that a response is returned or the command times out before sending the next command to I/O. If a command addresses a brain board that is offline or does not exist, the command will time out. Meanwhile, all other commands wait in the queue until the timeout occurs.

If you have multiple host applications and a high frequency of commands, system throughput may not be acceptable.

On the other hand, in an all-Ethernet network, the host can send commands to multiple destinations simultaneously and does not require a response or timeout before it sends the next command. For optimum performance in a multiple-host environment, we recommend not using Ethernet-to-serial routing, but replacing all B1s and B2s with E1s and E2s and connecting the brain boards directly to the Ethernet network.