

## RCI Performance Analysis

Performance: Overall system performance is a function of several variables:

- Baud rate – a system running at 38400 baud will perform about twice as fast as a system running at 19200 baud.
- Number of units/blocks polled – Each poll reads 2 bytes of data (16 input points) from a single card, or the 3 bytes of data from a handheld. The more inputs that are defined and polled, the more time will be spent polling those inputs.
- Number of output messages sent – Generally, changing the state of an output device results in one message being sent. A few types of devices may require two or three messages. The maximum number of output devices that can be supported depends on how often their state has to change. For most model RR devices, this is not a major performance consideration unless the PC application is trying to do direct control itself instead of having the RCI card do it, i.e. using the card as a simple on/off device controller instead of taking advantage of the built-in functions.
- Operating System – All else being equal, DOS-based PCs will run faster than Windows PCs. Windows systems running multiple applications will be slower than dedicated systems.

The following example will illustrate the performance that can be expected of a typical system. Assume the following conditions:

- A total of ten IO-48 cards, with all inputs and outputs used (160 inputs, 320 output points)
- Each output device uses two points (e.g. 2 light signals, searchlight signals, turnouts, crossing flashers, etc.) This is generally the worst-case scenario. This means there will be 160 devices attached to the 320 outputs.
- Operation at 38400 baud
- The system is configured to send ten output messages between each polling cycle
- All polling is at maximum priority – i.e. all inputs are polled every cycle.
- A reasonably fast computer is used and sufficient time is available to devote to the communication tasks. The times listed below are based on the communication times and don't include overhead due to other processes running on the PC which could slow down the communications task.

Under these conditions, the following will be found to be true:

- Each polling read takes about 0.005 seconds. Reading the 160 inputs will take 10 reads, or 0.05 seconds.
- Each output message to a device takes about 0.003 seconds. Sending 10 messages takes 0.03 seconds
- The typical cycle will be to poll all 160 inputs, send 10 messages, and repeat. This cycle will take 0.08 seconds (0.05 + 0.03).
- 0.08 seconds per cycle is equivalent to 12.5 cycles per second. Therefore, each input is polled about 12.5 times per second, and 125 output messages per second can be transmitted.
- With 160 output devices, each device could change state 125/160 times each second, or about one change of state every 1.28 seconds. Obviously, none of the devices on a model railroad will change state this frequently. Remember that devices like crossing flashers, blinkers, pulsed turnouts etc. are timed and controlled by the card, not the PC. There is no need to send messages at a high rate to control these functions.
- If we make the more reasonable assumption that every output device on the railroad changes state every 20 seconds, there would be a total of  $160 / 20 = 8$  state changes per second and therefore 8 output messages per second to send. This means only 0.024 seconds of each second is used to send messages – the rest of the time is utilized polling inputs. A much larger number of output devices could be supported with almost no impact on the system's performance.
- The above means 0.976 seconds of each second is spent polling. With each poll taking .005 seconds, this is equivalent to 195 polls per second, which means each of the 10 devices is polled  $195 / 10 = 19.5$  times per second. It is difficult to imagine a situation where this would not be sufficient.
- If we were to relax the requirements and require that each device is polled only 10 times per second, a system could support about twice the numbers specified above, i.e. 20 IO-48 cards or a total of about 320 input devices and 320 output devices. This number could be further extended by having less frequent polling of

devices that did not demand rapid response time. This assumes the low priority devices can be grouped into specific IO-48 cards. Even with 320 output devices, we could still change the state of every device on the layout at least once every 10 seconds, while polling all 320 inputs 10 times per second.

- If additional performance is required, a second serial port could be added to double the above numbers to 640 inputs and 1280 outputs (640 devices). A third or fourth port could also be added if necessary.

It should be evident from this analysis that very few situations would require a capacity that exceeds the capabilities of the RCI interface. Even large club layouts should not exceed 640 inputs and 640 output devices. Most of the arguments about networked systems, vs. polled systems, vs. other architectures (such as communication loops or parallel interfaces) are meaningless when you look at the actual performance of a REAL configuration.

If handheld controllers are used, special consideration should be given for these reasons:

- Handhelds currently only operate at 19200 baud
- Frequent polling is required to give good performance (recommended minimum is 10 per second)
- Errors (timeouts) are generated each time a handheld is unplugged, which tends to slow down the overall operation somewhat. To minimize the effect of unplugged handhelds, the polling priority of a unit will be automatically reduced if it is found disconnected. When it is connected again, the priority is set back to normal.
- Defining a large number of handhelds which don't exist or which are rarely plugged in will have a detrimental effect on the performance of the system. In this case the priority of the unplugged handhelds should be set very low so they are polled only once every two or three seconds. This will result in a short "connect delay" when a handheld is first plugged in, but will improve overall system performance.

Heavy windows activity on the PC (updating complex graphical screens with layout information, for example), or running other tasks at the same time, may have a negative impact on communications performance unless a sufficiently fast CPU is utilized. Also, remember that, although the communication system can send data to and gather data from the cards at the rates computed above, it is up to the application program to utilize this information to make control decisions, operate devices, or display results. How the application is designed will have a major impact on the actual performance as the user sees it. For example, even if the communication system is polling the inputs 20 times per second, if the application program only examines the data once per second, the response time to detectors, etc. will always average about one second. The application should be designed to perform these functions as often as possible without creating excessive overhead that slows down the overall system. An operational rate of about 5 times per second is the minimum suggested for realistic operation, since most people will not notice a time lag of a couple tenths of a second between an event and the system's response to that event.