INSTA-SEQUENCE 1.1

It is no surprise to plant engineers that a substantial cost of installation of a control system is labor for start up and trouble shooting during the initial installation followed by additions or changes to the system as needs change. We control system engineers have widely touted that now with software we can make those changes much easier through software. To a large extent this has been true but there are some generalized applications where much improvement is needed and have yet to be realized. In the old hardwired systems change was avoided like the plaque because of the problems involved. If anything the system was over engineered or over designed to start with so when those changes did take place at least some expansion was possible. Now with software users are demanding more complexity from the system and in return are finding it harder to get the system up and running. Then when a change is required the programmer has to make the change, test and go through start up again. In a plant that is on line down time cannot be tolerated thus changes can take weeks in small steps.

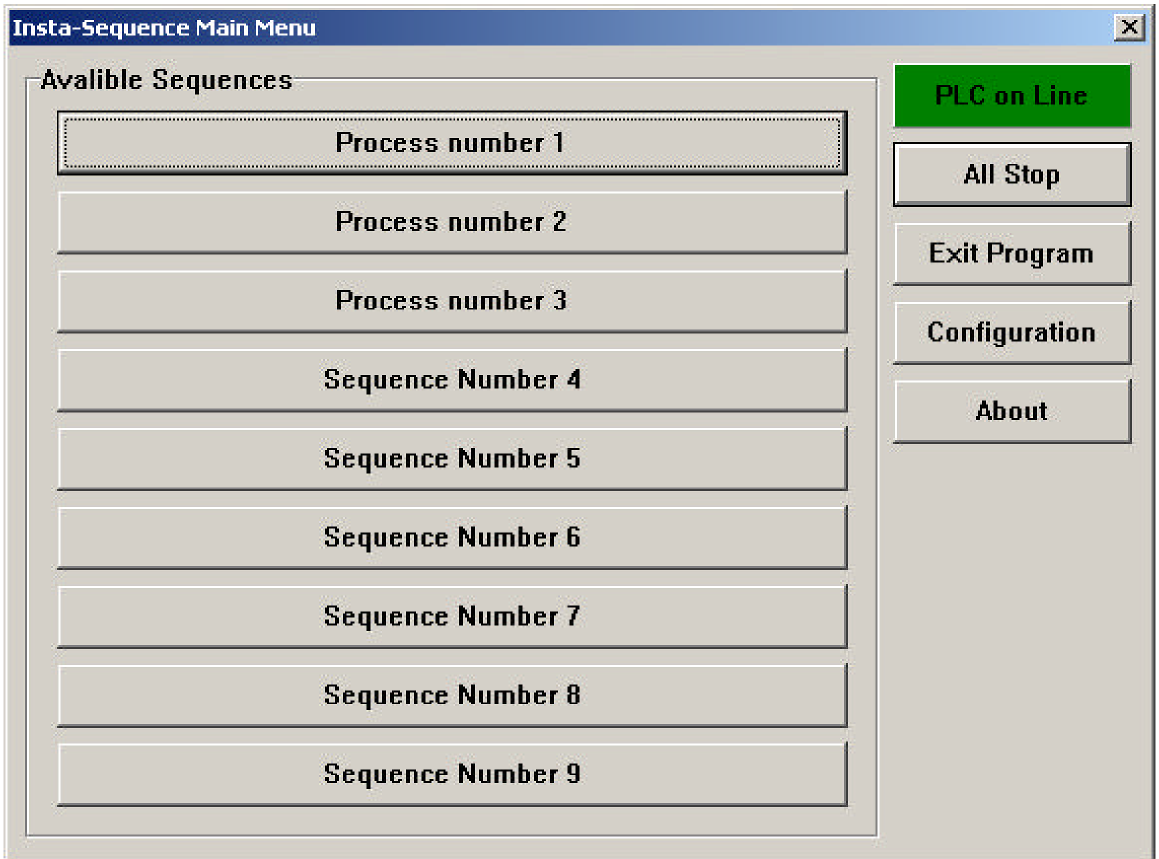
This paper describes recent developments to over come this problem for a generalized situation, which should have wide applicability for those users, which need a system that has dynamic program capability. Most systems installed to day are static systems in that the system was custom designed for a fixed application as to form and function. The system is linear in that an input produces a known output. The inputs and outputs are fixed addresses not easily changed accept through direct programming by a professional. Via dynamic programming this problem has been eliminated for many users. This system should be of interest especially for users in remote locations where access to a programmer is not easy.

Our firm recently was involved in a project in which the customer had a series of motors in sequence, which was made up of pumps and conveyors for a new plant. The system consisted of about 4 separate sequences of about 75 sequenced steps. The system was FILO or first in last out. The last item in the sequence started first until all were started and then during shut down the first conveyor was stop and allowed to empty and so forth down the line. The problem began as soon as the system was conceived. The owner wanted a PC based PLC system integrated into the system. The owner had very limited time operating the new facility and installed all the motors and starters and began to run everything in hand while the system was under development. My advice was to get some running experience so that the final system could be installed with as little change as possible. A year went by and still no consensus on how the system was to work. Our system was programmed and ready to be loaded but the owner was reluctant to install. In the meantime a new engineering staff came on board and naturally had a completely different approach. The software for the operator interface was replaced and new contracts for the programming were initiated. Our scope was reduced to that of the PLC only. A new sequence was presented to us and we started over again. We reprogrammed the system. This was great except that on a weekly basis a new sequence was given to us and we could only test on weekends. The timetables for start up and testing came and went as the seasons changed. The process changed and priorities in the plant changed. Not long after that the plant was sold and the plug was pulled on the project.

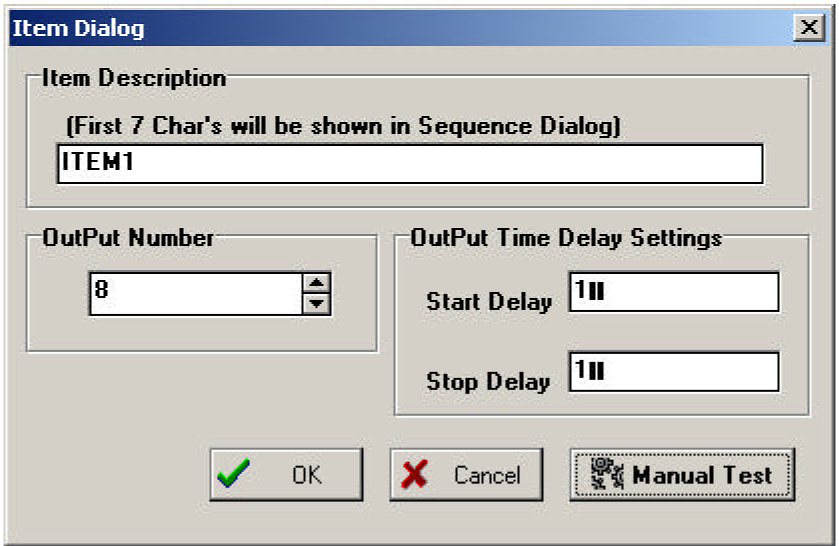
During the course of programming the system I commented to the owners engineer that I believed it was possible to provide a dynamic program that was field configurable. The static system had redundant characteristics in that each sequence item had a start timer, stop timer and was enabled by the previous equipment running confirmed by motor starter closure or similar feedback. The challenge was to develop a system that had all these characteristics plus any device in the sequence could be reassigned a position in the sequence without a programmer.

The initial system had design parameters of 250 inputs and outputs with 100 items in each sequence.

An arbitrary number of sequences were set at 9 per system.



This would allow the user to reuse the output in any sequence. For example each sequence could be a formula or one of 7 specific sequence tasks. Each item in each sequence would have individually set start timer value stop timer value and an assigned output on the PLC. The descriptive label was free form; the user could label it for the name or even in another language if required. When the user pops open the window to assign the start/stop and output value a HOA switch also is available for test the output to make sure it works or any other time hand operation is called for.



When complete the user puts it in auto and closes the window and therefore fixes the parameters for the item. During the hand testing the indicator changes color to show it is running. If duplicates of the output are found all the indicators that are enabled by the device also change color thereby helping to find duplicates of the output. Again the output is free form and can be any output from1 to 250. The user then opens the next item in the sequence on down the line until the last item in the sequence is programmed. If the last item is not item 100 the user identifies it as the last item in the sequence so the program knows where to stop.

The system has other features one of which is break stop. By pushing this on screen button the user can instantly stop the sequenced devices and then by toggling the button instantly start them up again without the timed sequence. The other feature is an emergency stop which when pushed will kill everything instantly and if released will stay off until the user starts the plant up again in sequence.

Thus after the system is configured the user pushes start and each device changes color to describe its state. Green is off, yellow blinking is enabled by previous motor start but the start timer is timing. When the start timer has timed out and the motor is confirmed to be running either with a hard-wired input or software the item in question turns red.

The intent of the system is to provide simplicity while allowing the user to customize their system as much as possible. The system is designed to recognize a motor running by providing input for contact closure on input cards. The system is designed that as soon as the output is selected a corresponding input is provided. For example if output two is selected its confirming input or feedback that the motor is on will be input two.

Thus the system can be used to control large complex systems. The benefit to the user is it is plug and play. The user can purchase and install a system knowing it has already been debugged and configuring the system can be accomplished in a matter of hours not days. Change can be accomplished without need of a programmer and can be done online in a matter of minutes. Although the system should be in a stop mode or unpredictable results will happen.

How many processing plants can benefit from such a system especially in areas where programmers are not abundant? Would not standardization in large facilities be beneficial? The benefits of mass production of such a complex control systems offers the end user reduced down time and low incremental costs associated with change. The initial cost is also lowered because very little customizing of the standard system is required. The system problem becomes one of application rather then implementation.

The only improvement to this system will come when we can read the mind of the end user to configure the system. And how many times have we wished for that?