

## Programming Features

This table highlights the programming features of a PLC-5 programmable controller.

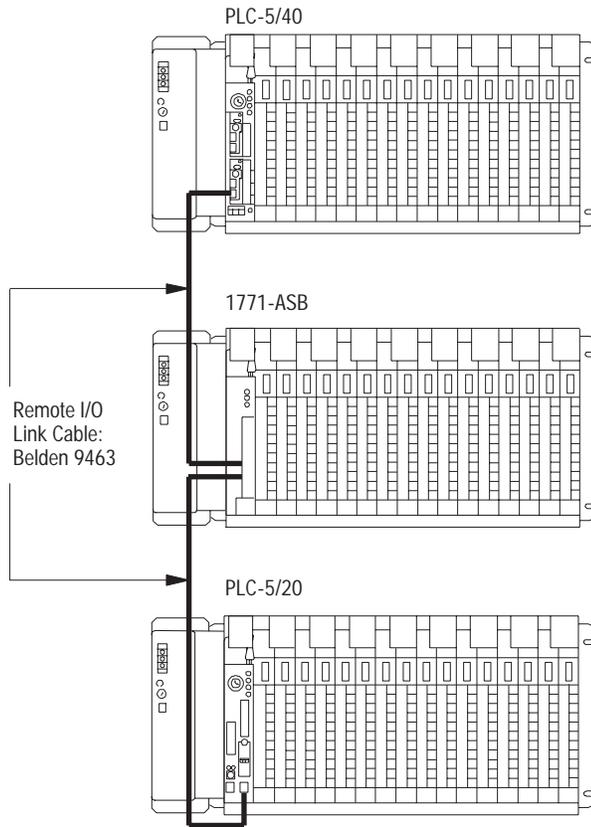
This Capability	Lets You
Ladder logic	<p>program using a language that is representative of relay logic.</p> <p>Choose this language</p> <ul style="list-style-type: none"> <li>• if you are more familiar with ladder logic than with programming languages such as BASIC Your plant personnel may be more familiar with ladder logic; consider their needs as well.</li> <li>• performing diagnostics</li> <li>• programming discrete control</li> </ul>
Subroutines	<p>store recurring sections of program logic that can be accessed from multiple program files.</p> <p>A subroutine saves memory because you program repetitive logic only once. The JSR instruction directs the controller to go to a separate subroutine file within the logic controller, scan that subroutine file once, and return to the point of departure.</p>
Sequential Function Charts (SFCs)	<p>use sequence-control language to control and display the state of a sequential process.</p> <p>Instead of using one long ladder program for your application, divide the logic into steps and transitions. A step corresponds to a control task; a transition corresponds to a condition that must occur before the programmable controller can perform the next control task. The display of these steps and transitions lets you see what state the machine process is in at a given time via a flowchart form.</p> <p>SFCs offer constructs that enable execution of multiple paths of logic, or a single selected path of logic, as well as the ability to jump forwards and backwards.</p> <p>Troubleshooting can be reduced to a small routine of logic instead of an entire ladder file.</p> <p>SFCs are best for defining the order of events in a sequential process.</p>
Structured text	<p>program using a language similar to BASIC.</p> <p>Choose structured text if you are:</p> <ul style="list-style-type: none"> <li>• more familiar with programming languages such as BASIC than with ladder logic</li> <li>• using complex mathematical algorithms</li> <li>• using program constructs that repeat or "loop"</li> <li>• creating custom data-table monitoring screens</li> </ul>
Main Control Programs (MCPs)	<p>separate sequential logic from ladder logic and structured text as a way of modularized your process and making troubleshooting easier.</p> <p>Use several main control programs (MCPs) to define one main control program for each particular machine or function of your process. MCPs accommodate independent or non-sequential activities.</p> <p>A main control program can be an SFC file numbered 1-999 or a ladder-logic file or structured-text program numbered 2-999.</p> <p>One data table is used by all MCPs (i.e., you do not have a separate data table for each MCP).</p>

## Using a Controller Channel as a Remote I/O Scanner

Configure a remote I/O channel for scanner mode to read and write I/O information between a controller and an I/O device remotely located from the controller.

A controller with a channel configured for scanner mode acts as a supervisory controller for other controllers that are in adapter mode as well as remote I/O adapter modules. The scanner-mode PLC-5 controller can:

- gather data from node adapter devices in remote I/O racks
- process I/O data from 8-, 16-, or 32-point I/O modules
- address I/O in 2-, 1-, or 1/2-slot I/O groups
- support a complementary I/O configuration
- support block-transfer in any I/O chassis



The scanner-mode PLC-5 controller:

- transfers discrete data and block-transfer data to/from modules in remote I/O racks as well as to/from controllers in adapter mode.
- scans **remote I/O buffers** asynchronously to the program scan.
- updates the **input/output image data table** from the remote I/O buffer(s) synchronously to the program scan

PLC-5 data table is updated synchronously to program scan (at housekeeping).

Remote I/O buffers are updated asynchronously to the program scan.

