

**CHAPTER 15 SOLVED PROBLEMS****General instructions for problems:**

Implement the solution and include the simulation logic for one of the following PLC processors:

ControlLogix/CompactLogix, **or**  
Micro800, **or**  
MicroLogix, **or**  
SLC-500

Simulation logic specifications:

1. Do not simulate START\_PB, STOP\_PB, and RESET\_PB in your tieback logic. However, define these physical inputs as internal tags/symbols so they can be manipulated without having to force them.
2. The simulation logic cannot use any internal logic (step internal coils, etc.) of the program being tested. You are allowed to use any of the defined physical I/O.
3. If the application contains a RESET\_PB, or some other tag/symbol used to reset the process, it may be used to set any initial states of the simulation logic (for example, set LS3 **on**).

When your solution is downloaded to a processor and the processor is placed in Run mode, the operator will force the appropriate inputs to start the process. For most of the problems, the test is started with the following procedure:

1. The START\_PB, STOP\_PB, and RESET\_PB are initially **off**.
2. The RESET\_PB is turned **on** and then **off**.
3. The STOP\_PB is turned **on**.
4. The START\_PB is turned **on** and then **off**.

The application should start. If the process repeats its operation, it should continue until stopped by the operator (for example, by turning **off** the STOP\_PB).

Your solution should include the following:

1. Function chart
2. Listing of program
3. Project files

## 2 Control Projects

### **Additional Specifications for a ControlLogix/CompactLogix Implementation:**

1. Place simulation logic in a **program** in the main task named “Simulation.”
2. Tags for any internal logic (timers, counters, Booleans, integers, etc.) must be program tags defined only for the Simulation program. They **must not** be controller tags.

### **Additional Specifications for a Micro800 Implementation:**

1. Place logic for transferring or not transferring to the duplicate input image in a POU named “Simulation\_Prog.”
2. Simulation logic must be in a User-Defined Function Block named “Simulation”.
3. Names for any timers used in the simulation logic need to start with “SimTmr\_” and must be local variables (scope is “Simulation” UDFB), not global variables.

### **Additional Specifications for a MicroLogix Implementation:**

1. The routine to copy the physical input addresses to the duplicate input image is in program file #99, named “DUPLICATE\_INS”
2. The duplicate input image is in data file B99.
3. Place simulation logic in program file #100, named “SIMULATE.”
4. Any simulation timers should go in data file T100.
5. Any internal bits for the simulation should go in data file B101.

### **Additional Specifications for a SLC-500 Implementation:**

1. Place simulation logic in program file #100, named “TIEBACK.”
2. Any simulation timers should be in data file T100.
3. Any internal bits for the simulation should be in data file B101.

**SP15-1.** Oiler Station Control. Implement the oiler station control of problem SP6-9 and include the simulation logic.

**SP15-2.** Case Erector Control. Implement the case erector control of problem SP6-11 and include the simulation logic.

**SP15-3.** Erbium Can Tipper/Rotator Control. Implement the station of problem SP6-13 and include the simulation logic.

**SP15-4.** Leak Check Station Control. Implement the valve leak check station control of problem SP7-9 and include the simulation logic.