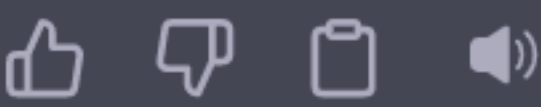




Write a self-contained 61131-3 structured text program for packaging bottles. After filling, the bottles are moved with a conveyor to packaging. A proximity sensor detects any bottle, another one only empty bottles. A cylinder removes empty bottles.



Here is an example program in 61131-3 structured text for packaging bottles using a conveyor, proximity sensors, and a cylinder to remove empty bottles:



objectivec Copy code

```
PROGRAM PackagingBottles
VAR
    conveyorRunning : BOOL := FALSE; // variable to track conveyor status
    bottleDetected : BOOL := FALSE; // variable to track if a bottle is detected
    emptyDetected : BOOL := FALSE; // variable to track if an empty bottle is detected
    cylinderActive : BOOL := FALSE; // variable to track cylinder status
END_VAR

// Main program loop
WHILE TRUE DO
    // Check if the conveyor is running
    IF conveyorRunning THEN
        // Check if a bottle is detected
        IF BottleSensor() THEN
            bottleDetected := TRUE; // set the bottleDetected variable to true
            emptyDetected := FALSE; // set the emptyDetected variable to false
        END_IF

        // Check if an empty bottle is detected
        IF EmptySensor() THEN
            emptyDetected := TRUE; // set the emptyDetected variable to true
            bottleDetected := FALSE; // set the bottleDetected variable to false
        END_IF

        // Check if an empty bottle is detected and the cylinder is not already active
        IF emptyDetected AND NOT cylinderActive THEN
            RemoveEmptyBottle(); // activate the cylinder to remove the empty bottle
            cylinderActive := TRUE; // set the cylinderActive variable to true
        END_IF

        // Check if a bottle is detected and the cylinder is active
        IF bottleDetected AND cylinderActive THEN
            cylinderActive := FALSE; // deactivate the cylinder
        END_IF
    END_IF
END_WHILE

// Function to simulate the bottle proximity sensor
FUNCTION BottleSensor : BOOL
VAR_INPUT
    // add input variables here if needed
END_VAR
VAR_OUTPUT
    result : BOOL; // output variable indicating if a bottle is detected
END_VAR
// replace this with your actual sensor logic
result := RANDOM(0, 2) = 1; // simulate a 50% chance of a bottle being detected
END_FUNCTION

// Function to simulate the empty bottle proximity sensor
FUNCTION EmptySensor : BOOL
VAR_INPUT
    // add input variables here if needed
END_VAR
VAR_OUTPUT
    result : BOOL; // output variable indicating if an empty bottle is detected
END_VAR
// replace this with your actual sensor logic
result := RANDOM(0, 2) = 1; // simulate a 50% chance of an empty bottle being detected
END_FUNCTION

// Function to simulate the cylinder removing an empty bottle
FUNCTION RemoveEmptyBottle
VAR_INPUT
    // add input variables here if needed
END_VAR
VAR_OUTPUT
    // add output variables here if needed
END_VAR
// replace this with your actual cylinder logic
WAIT_TIME(1); // simulate the time it takes for the cylinder to remove the empty bottle
END_FUNCTION
```

This is just an example program, and you will need to replace the simulated sensor and cylinder logic with the actual logic for your specific hardware setup. Also, make sure to add any necessary input and output variables to the functions to interface with your hardware.