



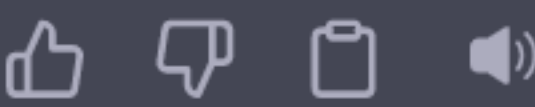
Write a PLC program in ST according to IEC 61131-3, for a pick-and-place application for a robot with two conveyors.

Here is the description of the process:

- 1) Whole machine has two modes: auto and manual
- 2) Manual Mode: When the button Manual is pressed, the robotic arm will begin to execute the manual control process: pressing Clip to clip the product from conveyor A, pressing Transfer to move the product to conveyor B, and pressing Release to release the product and send it away by conveyor B.
- 3) Auto Mode: When the button Auto is pressed, the robotic arm will begin to execute the auto control process once: clip the product (keep holding this product before releasing it), then transfer the product (the action takes 2 seconds), and then lastly release the product. The auto control process can be performed one more time if the button Auto is pressed again.
- 4) Manual control process and auto control process are interlocked.



Here is an example program written in Structured Text (ST) according to IEC 61131-3 for a pick-and-place application for a robot with two conveyors, based on the description provided:



objectivec Copy code

```
PROGRAM PickAndPlace
VAR
    ManualButton : BOOL; // Input signal for manual mode
    AutoButton : BOOL; // Input signal for auto mode
    ClipButton : BOOL; // Input signal for clip action
    TransferButton : BOOL; // Input signal for transfer action
    ReleaseButton : BOOL; // Input signal for release action
    ConveyorA : BOOL; // Input signal for presence of product on conveyor A
    ConveyorB : BOOL; // Output signal to control conveyor B
    RoboticArm : BOOL; // Output signal to control the robotic arm
    Mode : INT := 0; // Internal variable to store the current mode (0 = manual, 1 = auto)
    AutoProcess : BOOL := FALSE; // Internal variable to store whether the auto control process is running
END_VAR

// Manual mode control process
IF ManualButton THEN
    Mode := 0; // Set mode to manual
END_IF

IF Mode = 0 THEN // Manual mode
    IF ClipButton AND ConveyorA THEN
        RoboticArm := TRUE; // Clip the product
    ELSIF TransferButton THEN
        ConveyorB := TRUE; // Transfer the product to conveyor B
    ELSIF ReleaseButton THEN
        ConveyorB := FALSE; // Release the product from conveyor B
    END_IF
END_IF

// Auto mode control process
IF AutoButton THEN
    Mode := 1; // Set mode to auto
END_IF

IF Mode = 1 THEN // Auto mode
    IF NOT AutoProcess AND ConveyorA THEN // Only start the process if not currently running and product is present
        AutoProcess := TRUE; // Set flag to indicate that the auto process is running
        RoboticArm := TRUE; // Clip the product
        WAIT 2; // Wait for 2 seconds to transfer the product
        ConveyorB := TRUE; // Transfer the product to conveyor B
    END_IF
    IF ConveyorB AND NOT ConveyorA THEN // Release the product from conveyor B once it reaches the end
        ConveyorB := FALSE;
        AutoProcess := FALSE; // Clear the flag to indicate that the auto process is finished
    END_IF
END_IF
```

In this program, the input signals for the buttons and conveyors are defined as `BOOL` variables, and the output signals for controlling the conveyor and robotic arm are also defined as `BOOL` variables. The `Mode` variable is used to keep track of the current mode (manual or auto), and the `AutoProcess` variable is used to keep track of whether the auto control process is currently running.

In the control process for manual mode, the program checks the status of the buttons and conveyors to perform the clip, transfer, and release actions as described. In the control process for auto mode, the program checks the status of the buttons and conveyors to perform the auto control process as described. The `WAIT` instruction is used to pause the program for 2 seconds during the transfer action in auto mode.

The interlock between manual mode and auto mode is achieved through the use of the `Mode` variable, which determines which control process to execute based on the current mode.