Introduction to Programmable Logic Controllers Ex7_conveyor

DTU 31343

Eduard Maximilian Fiedler s210134

Task 1

The control system for this purpose should use a shift register, that updates every time, the conveyor moves one item at a time. Then, the left most bit would be determined based on the reading from the detector. If the detector reads an item, the bit would read 1, and if no item is present, the bit would read 0. For the shift register, a byte would be used or at least a 6-bit register. Each shift would move the bits to the right, where the paint sprayer would activate based on the 6^{th} bit from the left. The proposed shift register is shown in Figure 1.

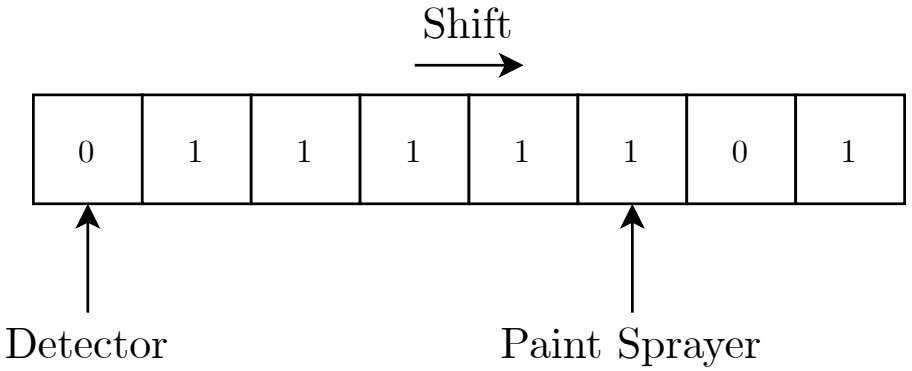

Figure 1: 8-bit shift register for the conveyor bet control system.

If a ladder diagram were used to implement this, checking if the 6^{th} bit from the left is 1 would be based on whether or not the coil in the respective rung has been energised.

A diagram showing the process for one box on the conveyor is shown in Figure 2.

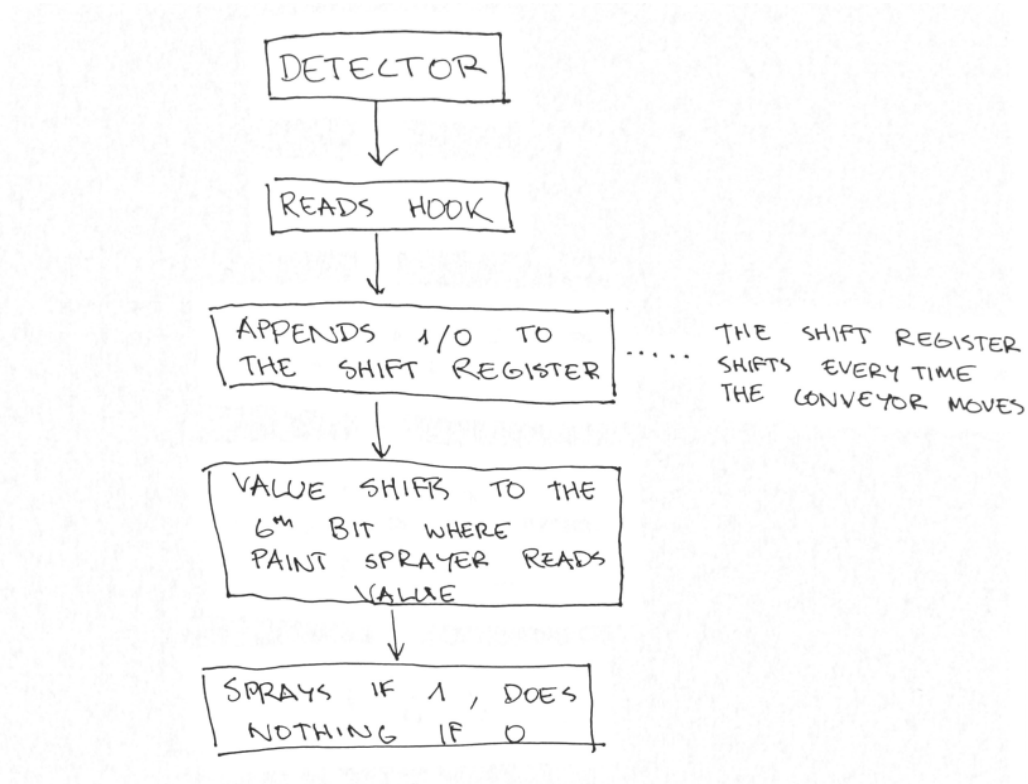

Figure 2: Diagram showing the process for a box running through the conveyor system.

Task 2

Before describing the logic based on what the inputs and outputs have to do, it was decided that given that there is a 180 cm distance between the detector and gripper, the conveyor belt should ideally move at $60\,\mathrm{cm\,s^{-1}}$. Therefore, a 4-bit shift register could be used, similar to what was described in Task 1, that would shift right with a 0 every second or a 1, in the left most bit, if a failure was detected. Then, the right most bit would be read by the gripper such that it would lower when there is a 1.

The logic of the input and outputs is described by describing their functions:

- di0: must read if a failure (1) or not (0) every second to update the 4-bit shift register as described previously
- di1: returns 1 if the box has become full (i.e. incoming boxes interrupt the light beam), and must cut the motor in correspondence with the second intervals used for the shift register
- do0: must release boxes in sync with shifts, and prevent a release if di1 reads 1
- do1: must grip when di0 has read a 0, 3 shifts prior
- ai0: must read 0 when 7 V and 1 when less than 4 V and counting up, and resetting this counter when di1 reads 1
- ao0: given that it was decided for the conveyor belt to move at $60\,\mathrm{cm\,s^{-1}}$, the signal strength should be 6 V as long as di1 reads 0.

As mentioned, it is very important to keep the flow of boxes in sync with the shift register. This is important in regards to how boxes are dropped onto the conveyor belt and when the conveyor belt stops, as they must remain in sync with the shifts in the register. Therefore, there must be an internal timer that only updates every second, that creates intervals that allow the affected inputs and outputs to update.

A flowchart describing the process is shown in Figure 3.

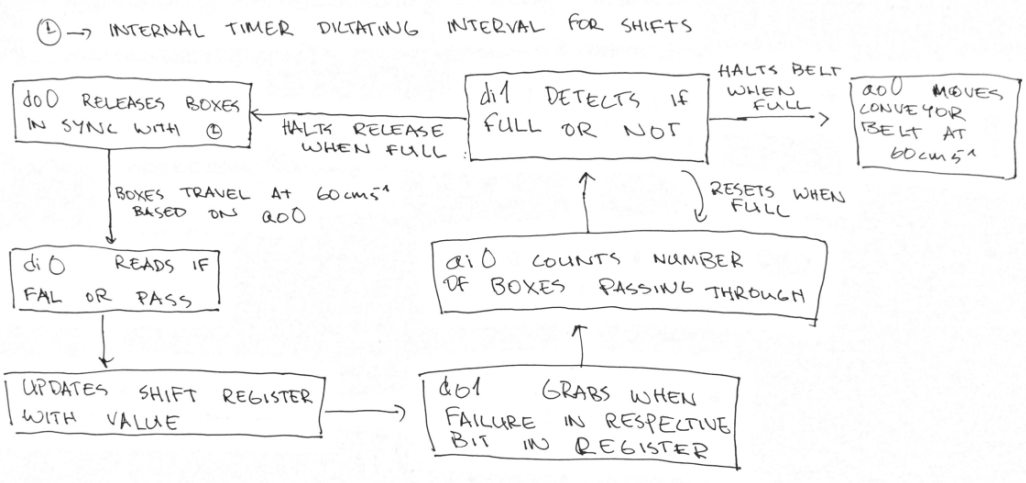

Figure 3: Diagram showing how the sensors work with each other in the conveyor belt setup.