COMPSYS 725 2019

Computer Networks and Distributed Applications
Assignment 2 (15%)

Due Dates

- Code/Document Submission Due 19 October (23:59)
- Demo/Interview in Friday Week **12** Lab (12:00-14:00), approx. 10 minutes per group

Overview

The company IdeaCorp has asked your **team of 3** to improve the implementation of a rudimentary baggage handling system (BMS). In this assignment you will document and implement the conveyor control system for a small part of the BHS. Your main aim is to prevent the collision of bags using appropriate mutual exclusion algorithms.

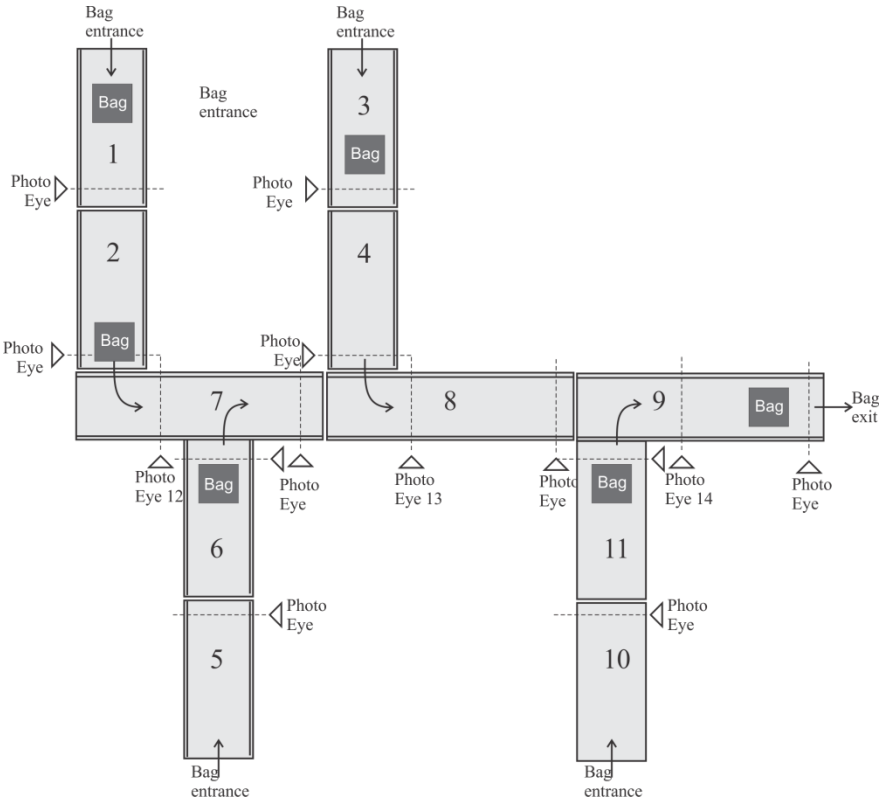


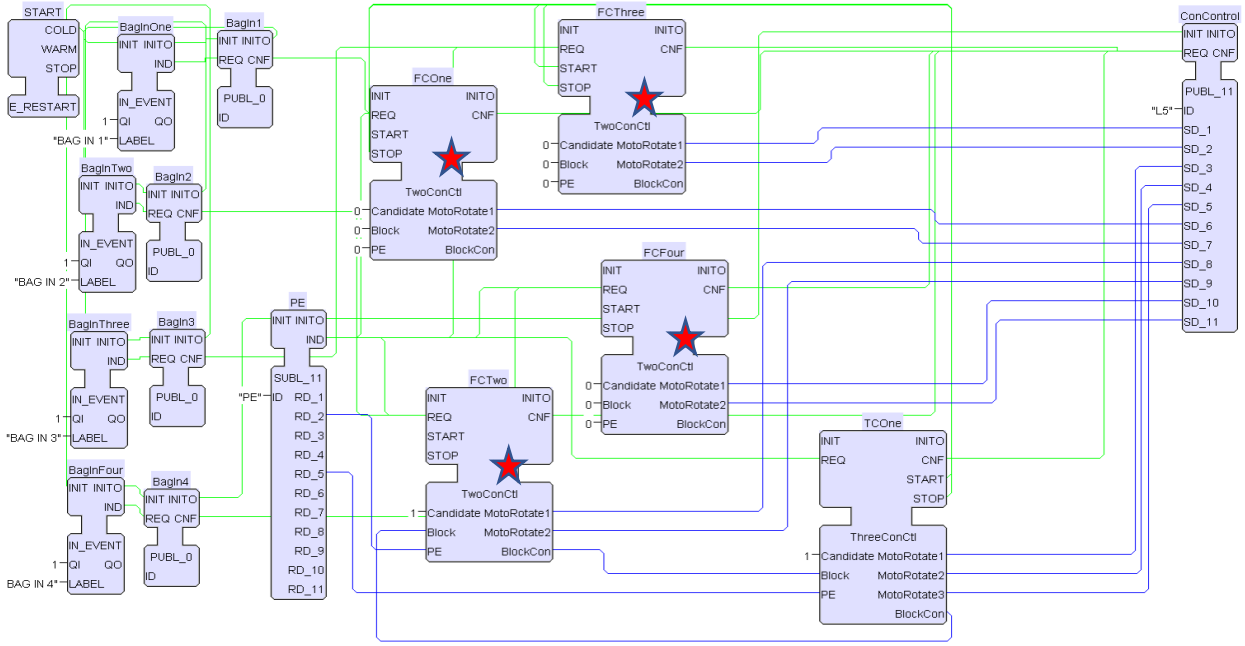
Figure 1: A small baggage handling system

The system consists of 11 conveyor sections. Bags can appear at one of four entrances and should be moved to the bag exit. Conveyors are equipped with several Photo Eye (PE) sensors that can be used as inputs to the control logic. You will use a simulated environment in this assignment running on a desktop PC; in the real system, each conveyor section can be controlled using an embedded microcontroller.

Baggage control is already implemented in a fully distributed fashion (with several re-used function blocks reading sensors and communicating with other controllers). Note that controllers are grouped as composite function blocks according to the grouping in the physical layout.

System Requirements

1. There are three critical sections in the BHS layout, so the system must safely handle bag merging
2. Conveyors should implement cascading starts and stops. For example:
    a. When conveyor 7 stops, it should issue stop to conveyor 2 and 6, and conveyor 2 will issue stop to conveyor 1, and similarly conveyor 6 issues stop to conveyor 5
    b. When conveyor 4 stops, it should issue stop to conveyor 3
    c. Vice versa for starting



Assignment Tasks

1. **Group Task: Your group will need to implement mutual exclusion for the 3 critical sections and demonstrate this in the laboratory (5%):**

   Using the following mutual exclusion algorithms: central server, ring token, and multicast, implement a mutual exclusion algorithm for each critical section. As a group, choose which algorithm is used for which critical section. Also implement cascading start/stop. *Note: These algorithms will be discussed in lectures, but you can also read about these directly in 15.2 of "Distributed Systems" (available in the Reading List on Canvas) if you want a head start*

   In your group code submission, zip the entire FBDK directory (with your solution). You **must** include a README file that explains what you have implemented, which algorithm for which section, and what steps are needed to get your assignment running. *Hint: Use BaggageSystemCTL in the cs725 folder/package. You may also find it helpful to modify control logic of a single conveyor in ConveyorCTL function block to add the implementation of your designated mutual exclusion algorithm. The current code is basic and reads only one PE.*

2. **Individual Task: Documentation (10%): Each student must produce a short report containing 2 parts (5% each):**
   a. Explain each of the algorithms (central server, ring token, and multicast) in your own words, and present a table comparing the algorithms based on various metrics based on lecture material AND your reading of the textbook chapter **(hint: exam revision)**
   b. Explain, in detail, one of the critical section implementations (each student must choose a unique section)
      i. Include a screenshot of the implemented ECC (and network of FBs if necessary)
      ii. Explain the implementation in your own words, including the ECC and how it works, referring to your screenshots as required
      iii. Add a picture of your favourite pokemon at the end of the report
      iv. It may be helpful to draw an additional flow diagram (if your ECC is particularly hard to follow).

3. **Group Task Bonus Marks (TA will not provide help for these)**

   Note: Each of these bonus marks can be used to make up lost marks in other parts of this assignment, but you cannot score more than 15% total.

   - Basic: Cascade start and stops with a real time delay (1 Mark)
   - Intermediate: Add more entry/exist points to the layout. You will need to understand how the HMI can be extended (2 Marks)
   - Advanced: Colour bags differently, depending on exit points (3 marks)