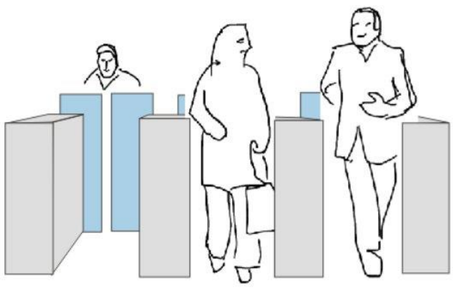
Industrial Processes Automation

MSc in Electrical and Computer Engineering
Spring Semester 2019/2020
1st Exam, 16th June 2020

Read all questions of the exam carefully before starting to answer.

- *Provide detailed justifications to all answers.*
- *The use of bibliographic material, either in paper or in digital format is allowed.*
- *Exchange of information is forbidden (e.g. voice, WiFi, Bluetooth, GPRS, WAP,…).*
- *Exam duration: 2 hours and 30min.*

Q1. *[PLC IO]* Consider an automatic door that opens upon the presentation of a valid ticket (see figure). The control of the system is based on one main PLC and one auxiliary PLC. The main PLC has connected two input switches for system power ON/OFF (inputs on **%i0.2.0** and **%i0.2.1**) and commands the motor to open / close the door (outputs **%q0.4.0** and **%q0.4.1**). The auxiliary PLC receives an enabling signal from the main PLC and outputs to the main PLC the ticket detected and ticket valid signals.



(i) Draw an electric diagram that details the automation solution. Indicate all sensors, actuators, interconnections of the PLCs and power supply. Use a normally open switch to turn the system ON and a normally closed to turn the system OFF.

(ii) Propose one ladder diagram for the main PLC, considering the ON/OFF switches (connected to the inputs **%i0.2.0** and **%i0.2.1**), to actuate a memory variable **%M0** indicating the ON/OFF state of the system.

(iii) Write the truth table where the main PLC inputs **%i0.2.0** and **%i0.2.1** actuate the memory variable **%M0**.

Q2. *[Counters]* Consider the problem of counting of the number of cars entering / leaving a parking lot. The park has one sensor that signals every car entering the park (sensor wired as **%i0.2.1**) and has another sensor that signals every car leaving the park (**%i0.2.2**). There are two semaphore lights, red (**%q0.4.1**) and green (**%q0.4.2**), in the entry gate of the park. The number of parking places is **200**.

(i) Write a *Ladder* program, based in an Up/Down counter block, to turn on the red light when the park is full. Include code also to command the green light.

(ii) Write a Structured Text section to replace the counter using a variable **numberOfCars** of type UINT, and any other variables necessary. Consider a sensor is continuously ON while the car is passing over it. Comment whether your code handles sensors staying ON for longer than one scan cycle.

(iii) Given the code written in (ii) consider the parking lot is **empty** and a large metallic object is used to **simulate a car leaves** the parking lot. Does this incorrect use of the system pose a problem to your code? More in detail, consider that, after simulating one car left the (empty) parking lot, **210 cars** try to enter the parking lot. Indicate the **state, ON/OFF, of the red light** that each car sees when it arrives to the parking lot entrance. Assume that a car that sees the red light ON does not wait the red light going OFF, it goes to another parking lot.

Q3. *[Square waves]* The code provided in this question allows generating square waves based on a sequence of pulses. Various wave forms are available depending on **your student number** written in variable **%MD0**. Consider that the code shown bellow is the single code run by a PLC in a MAST section configured to cyclic.

```
(* sequence of 1sec pulses *)
TP_0(IN:= NOT(%m0),
     PT:= t#1s,
     Q => %m0 );

(* output signal on %m10 *)
if RE(%m0) then

  %MW2:= MOD(%MW2+1,3);

  case MOD(%MD0,10) of
    0,1,8,9: %m10:= (%MW2=0);
    2      : %m10:= (%MW2=2);
    3      : %m10:= (%MW2<>1);
    4      : %m10:= (%MW2=1);
    5      : %m10:= (%MW2<>2);
    6,7    : %m10:= (%MW2<>0);
  end_case;

end_if;
```

(i) Assume that the assignments denoted by the symbol "**:=**" are the single instructions consuming time, and that each one takes about **0.1msec**. The PLC input and output take **1msec + 1msec**. Note some scan cycles are faster than others. Compute the minimum scan cycle period and the maximum scan cycle period.

(ii) Write additional Structured Text code so that at the first scan cycle the memory variable **%MD0** is set with **your student number** (5 digits).

(iii) Considering that **%MD0** contains your student number, which value results from the calculation **MOD(%MD0, 10)** ? Indicate also the possible values of **%MW2**.

(iv) Assume **%MW2** is zero at t=0 (as usual in PLCs after boot). Plot the values of **%MW2** and of the memory **%m10** from **t=0** to **t=3** seconds, according to the program running having **%MD0** set with your student number. Indicate the approximate frequency and duty cycle of the square wave in **%m10**.

Q4. *[String write]* Consider the objective of writing a string in the memory of a PLC. In the following the PLC is referred as ***PLC1***. The string is 10 characters long, is terminated by two zero bytes (00), and has the format:

*A*B*C*D*E00

where * indicates the ASCII character $42_{(10)}$, and (10) denotes base-ten. ABCDE indicate the five digits that form **your student number**. Note: the ASCII representation for the character 0 (zero) is $48_{(10)}$, for character 1 (one) is $49_{(10)}$, and so on.

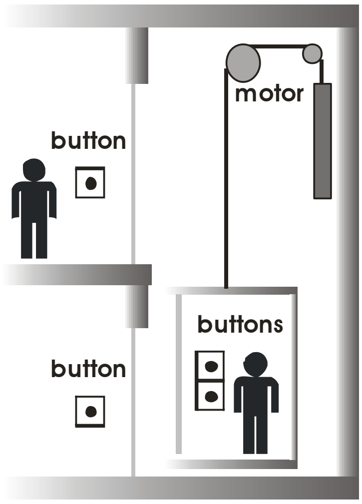
(i) Make a Structured Text program writing the string on the PLC memory words, starting at word **%MW100** and ending at word **%MW105**. Condition the writing of the string to happen only if **%m0** is true. Use **%m0** to ensure the string is written just once. Note: the six words provide a total of 12 bytes.

(ii) Rewrite your Structure Text using double words, starting at **%MD100**.

(iii) Consider a second PLC, named ***terminal PLC***, connected to ***PLC1*** by an Ethernet connection. Do you need to write code in ***PLC1*** for the ***terminal PLC*** to access (read) the string written in the memory of ***PLC1***? Justify by giving examples of functionalities provided by default, i.e. not explicitly written by you, in the PLCs used in this course.

(iv) READ_VAR is a PLC function, available in the PLCs used in this course, that reads remote data. Like other PLC functions, as timers or counters, READ_VAR in general requires multiple scan cycles to complete its work. Can READ_VAR be called at every scan cycle as other PLC functions? Discuss memory usage by the PLC functions to justify your answer.

Q5. *[Petri net model]* In this problem is modelled an elevator working in a two floors building (see figure). The elevator is commanded by one PLC given the input of buttons. Two buttons within the cabin allow selecting floor0 or floor1. Two external buttons allow calling the elevator to floor0 or floor1. The motor moving the elevator cabin has two commanding signals, one for moving up and the other for moving down.



a) Model the elevator as a Petri net (PN). Modelling starts with separated PNs with zero initial markings. The separated PNs will be combined later. Two incidence matrices are to be used as building blocks:

$$D_a = [+1 \quad -1] \text{ and } D_b = [+1 \quad -1 \quad -1]$$

for the following model specifications:

[Requests-buffer] The requests generated by the buttons are accumulated in a queue. Requests sorting is not considered in this problem. Just consider a simplified PN with incidence matrix $D_1 = D_a$ where the place marking indicates the number of queued requests (pushed down buttons).

[Requests-server] Consider the incidence matrix $D_2 = D_b$ to represent receiving a motion request stored in the buffer and dispatching to move up/down the elevator.

[Elevator-moving] Consider the block diagonal incidence matrix $D_3 = diag(D_a,\ D_a)$ indicating the elevator can be moving up or down. Transitions indicate start / stop events for the motor to go up / down.

a1) Draw a graphical representation of the PN.
a2) Denote transitions as t_{ij} and places as p_{ij}, where $i \in \{1, 2, 3\}$ indicates the part of the complete Petri net and j is used to number transitions or places. Detail the meanings of places and transitions involved in the model.

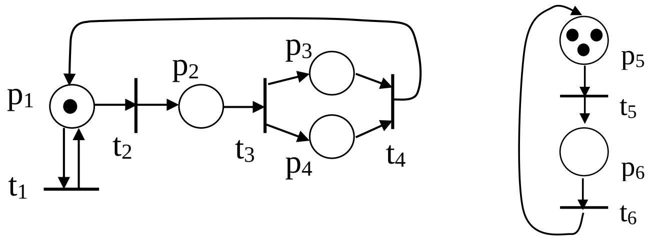
b) Design a supervisor based on place invariants to impose (i) the requests-server handles at most one request at a time, (ii) the elevator cannot have two opposing motions occurring simultaneously.

c) Use generalized linear constraints to design a supervisor that (i) makes the requests-server accept requests stored in the requests-buffer, and (ii) sends the move up/down request generated by the requests-server to the elevator.

d) Use generalized linear constraints to design a supervisor that conditions the requests-server to accept one novel request just upon completion of an up/down motion.

e) Indicate the boundedness and liveness of your complete PN. Comment the sentence: "The model can be unbounded despite not needing to handle an arbitrary large number of requests".

Q6. *Petri net properties:* This problem focus on Discrete Event Systems analysis tools studied on the course, for the Petri Net graph depicted in the figure in the right.



a) Discuss the boundedness and safeness of the Petri net, resorting to a reachability tree. Discuss the liveness of each transition.

b) Discuss the conservativeness of the Petri net and provide the weight vector.

c) Find cycles of operation encompassing two or three transitions in this Petri Net.

d) Discuss liveness, boundedness and existence of states covering others, in two cases (i) after setting the arc weight $w(t_1,p_1)=0$ and (ii) after setting $w(p_1,t_1)=0$ while keeping $w(t_1,p_1)=1$. In the next questions return to the case where all weights are 1.

e) Design a supervisor based on place invariants, using generalized linear constraints, such that firing t_5 cannot consume more that 2 tokens from p_5. Draw the supervisor just determined on the Petri net. Is it possible to reach a state where the marking of p_5 is null, i.e. $\mu_5 = 0$?

f) Consider transitions t_3 and t_6. Design a supervisor based on place invariants, using generalized linear constraints, such that if the two transitions are to fire at the same time, then transition t_3 has the priority to fire first.

Q7. *[Safeness]* Consider a Petri net $C = (P, T, D^+, D^-, \mu_0)$ where D^+andD^- denote pre and post incidence matrices, μ_0 is the initial state and

$$P = \{p_1, p_2, \ \dots \ , p_N\} \qquad D^+ = \begin{bmatrix} \vec{0} & 1 \\ I_{N-1} & \vec{0} \end{bmatrix} \qquad D^- = I_N$$
$$T = \{t_1, t_2, \ \dots \ , t_N\}$$

N is a natural number, $N \geq 3$
I_N is a $N \times N$ identity matrix

a) Let $\mu_0 = [1 \ 1 \ 0 \ \cdots \ 0]^T$. Compute a supervisor,$(D_c, \mu_{c0})$ based in place invariants, that makes the net safe.

b) Ignore the supervisor designed in (a). Consider $N = 5$, $\mu_0 = [1 \ 1 \ 0 \ 0 \ 0]^T$, $L = [0 \ 0 \ 1 \ 0 \ 0]$, $b = [1]$, $R_1 = [0 \ 0 \ 0 \ 1 \ 0]$, $R_2 = [1]$. The linear constraint $L\mu_p \leq b$ implies safeness of any place(s)? Let $L_a = R_1 + R_2 L$ and $b_a = R_2(b+1) - 1$, prove that the solutions $\mu_p \in \mathbb{N}_0^5$ of $L_a \mu_p \leq b_a$ are also valid solutions for $L\mu_p \leq b$.

c) Continuing (b), propose a supervisor respecting $L_a \mu_p \leq b_a$. Check the transitions involved in the supervisor. Suggest a supervisor respecting $L\mu_p \leq b$ and that t_3 is unobservable.

PS: Do not forget to identify all sheets of paper.

Good Luck,
José Gaspar