# Annex 1 - Transition Priorities Setting

Locate in the course SVN the lab2\_sim\_1\_priority simulation folder. Edit in Matlab the test file tst\_set\_priority.m. See in the test file the Petri net defined with three places and four transitions (Figure A1). The simulation is run setting transition 3 to have the highest priority. In this example only one transition is prioritized but one can set a sequence of priorities as e.g. [3 4 1 2].

% incidence matrix

D**=** **[-**1 **+**1 **-**1 **+**1

**+**1 **-**1 0 0

**+**0 0 **+**1 **-**1**];**

Pre **=** **-**D**.\*(**D**<**0**);**

Post**=** D**.\*(**D**>**0**);**

M0 **=** **[**1 0 0**]';** % initial marking

ti\_tf**=** **[**0 3 1e-2**];** % time [ini, end, delta]

% simulation

**[**t2**,** M2**,** **~]=** PN\_sim**(**Pre**,** Post**,** M0**,** ti\_tf**,** struct**(**'tprio'**,**3**));**

Figure A1: Petri net definition and simulation call.

The test file runs the simulation of the Petri net by calling the simulation function PN\_sim.m. The simulation function contains some code to be implemented in the context of the laboratory.   
Figure A2 shows in red the code to implement in order to assign priorities to specific transitions.

**if** **~**isfield**(**options**,** 'tprio'**)** **||** isempty**(**options**.**tprio**)**

% use transition numbers as default priorities

**[**PN**.**MP**,** qk2**]=** PN\_state\_step**(** PN**.**MP**,** Post**,** Pre**,** qk **);**

**else**

% use specific priorities

**error('Yet to implement: define Post2, Pre2 and qk2')**

**[**PN**.**MP**,** qk2**]=** PN\_state\_step**(** PN**.**MP**,** **Post2,** **Pre2,** **qk2** **);**

**end**

Figure A2: Detail of PN\_sim.m showing a part of the code to implement to assign transition priorities.

Recall, in this guide, the suggestion for implementing in Matlab a matrix that permutes the entries of a vector. That matrix allows to permute the entries of the firing vector.

This laboratory assignment aims at studying Discrete Event Systems (DESs) in the aspects of modeling, analysis of properties and synthesis. Synthesis will be based on a recent methodology in the framework of supervised control. This assignment further develops the previous assignment in the keyboard reading component by introducing fault handling mechanisms.

The tools to be used in this work are MATLAB and a Petri Net editor. In the last part of the work the Schneider PLCs will be used once more to validate the proposed methodologies.

**DES modeling:**

In the part A of this second laboratory assignment a Petri net was designed to model the process of reading one keyboard key as a DES. The aim of the current partial assignment, part B, is to analyze that Petri net, resorting to the properties studied in this course. Moreover, its simulation resorting to an available software package is required.

Note: See in the course SVN tools helping this part of the assignment, e.g. the graphical freeware editor "PIPE2" which allows creating models to import with the Matlab toolbox "tpn5" (use the most recent "tpn5" version made available in the API course). These tools are installed in the laboratory computers and can be used at home with SVN:

<svn://svn.isr.tecnico.ulisboa.pt/dccal/projects/lsdc4_api>

See more details in the course webpage.