



Industrial Processes Automation

MSc in Electrical and Computer Engineering
Scientific Area of Systems, Decision, and Control

Winter Semester 2018/2019

2nd Laboratory Assignment ¹

Handling Faults in Keyboard Reading

Part A – Petri Net

Group: C2
70547 – João
Ferreira
75268 – Rúben
Tadeia
75987 – João
Ribafeita
80978 –
Gonçalo Pedro



Q1: Write the list of events of the DES you are proposing. (Fill the next table, adding as many lines as needed).

Event Identifier	Description
T1	Temporizador que controla a transição da coluna 1 para a 2 do teclado
T2	Temporizador que controla a transição da coluna 2 para a 3 do teclado
T3	Temporizador que controla a transição da coluna 3 para a 1 do teclado
T4 T8 T12	Tecla premida na 1 ^a linha do teclado
T5 T9 T13	Tecla premida na 2 ^a linha do teclado
T6 T10 T14	Tecla premida na 3 ^a linha do teclado
T7 T11 T15	Tecla premida na 4 ^a linha do teclado
T16 T20 T24	Tecla levantada na 1 ^a linha do teclado
T17 T21 T25	Tecla levantada na 2 ^a linha do teclado
T18 T22 T26	Tecla levantada na 3 ^a linha do teclado
T19 T23 T27	Tecla levantada na 4 ^a linha do teclado

Resposta: De modo a diminuir o número de entradas da tabela, a partir da 3^a linha da tabela, colocaram-se 3 transições que correspondem respectivamente, em todos os casos, às colunas 1, 2 e 3 do teclado.

¹ Original guide by Prof. Paulo J. Oliveira. Revised by Prof. José Gaspar (2019).



Q2: Write the list of conditions of the DES you are proposing. *(Fill the next table, adding as many lines as needed).*

Condition Identifier	Description
P1	Coluna 1 ativa com as 4 linhas desligadas
P2	Coluna 2 ativa com as 4 linhas desligadas
P3	Coluna 3 ativa com as 4 linhas desligadas
P4	Coluna 1 ativa com a 1ª linha ligada [Tecla 1]
P5	Coluna 1 ativa com a 2ª linha ligada [Tecla 4]
P6	Coluna 1 ativa com a 3ª linha ligada [Tecla 7]
P7	Coluna 1 ativa com a 4ª linha ligada [Tecla *]
P8	Coluna 2 ativa com a 1ª linha ligada [Tecla 2]
P9	Coluna 2 ativa com a 2ª linha ligada [Tecla 5]
P10	Coluna 2 ativa com a 3ª linha ligada [Tecla 8]
P11	Coluna 2 ativa com a 4ª linha ligada [Tecla 0]
P12	Coluna 3 ativa com a 1ª linha ligada [Tecla 3]
P13	Coluna 3 ativa com a 2ª linha ligada [Tecla 6]
P14	Coluna 3 ativa com a 3ª linha ligada [Tecla 9]
P15	Coluna 3 ativa com a 4ª linha ligada [Tecla #]



Q3: Write the table of pre and post conditions for each of the events. *(Fill the next table, adding as many lines as needed).*

Event	Pre-Conditions	Post-Conditions
T1	P1	P2
T2	P2	P3
T3	P3	P1
T4	P1	P4
T5	P1	P5
T6	P1	P6
T7	P1	P7
T8	P2	P8
T9	P2	P9
T10	P2	P10

T11	P2	P11
T12	P3	P12
T13	P3	P13e
T14	P3	P14
T15	P3	P15
T16	P4	P1
T17	P5	P1
T18	P6	P1
T19	P7	P1
T20	P8	P2
T21	P9	P2
T22	P10	P2
T23	P11	P2
T24	P12	P3
T25	P13	P3
T26	P14	P3
T27	P15	P3

Resposta: Para auxiliar a compreensão desta tabela sugere-se que a análise da *Figura 1*, presente na resposta seguinte.



Q4: Characterization of the Petri net. (i) Draw the graph of the Petri Net. (ii) Comment about temporizations used, or not used, in the proposed DES. (iii) Obtain the incidence matrix, $D=D^+-D^-$, describing the proposed Petri Net.

Resposta: A abordagem escolhida consiste numa divisão em 3 grandes blocos. Em que cada bloco representa uma das 3 colunas do teclado. Em cada um dos estados espera-se que seja pressionada uma tecla na sua coluna respetiva. A partir deste momento o sistema evolui a partir do primeiro estado, em que espera por uma nova tecla premida. As transições T1, T2 e T3 correspondem a temporizadores usados neste DES. Estes temporizadores são a chave para a leitura correta de uma tecla em qualquer coluna.

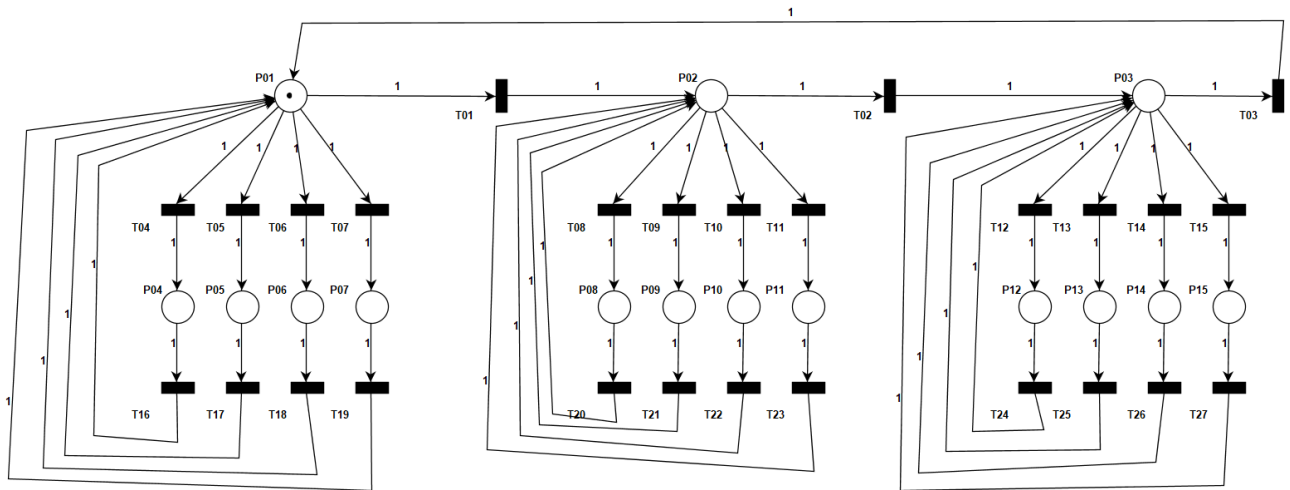


Figura 1 – Gráfico da Petri Net desenvolvida

A matriz de incidência pode ser observada no [Anexo 1](#). Esta matriz já está na forma de $D = D^+ - D^-$ e tem dimensões 15x27, ou seja 15 lugares e 27 transições. O método utilizado para atribuir valores foi, -1 às *pre-conditions* e $+1$ às *post-conditions*.

De modo a evitar ter uma petri net morta em que nenhuma transição era disparada, foi adicionado um marker na primeira coluna. Este marker pode deslocar-se ao longo das outras 2 colunas (horizontalmente) ou ao longo das linhas, neste caso ao longo das teclas respectivas de cada coluna (verticalmente) até regressar novamente ao estado inicial. A análise da liveness desta petri net será feita em detalhe na segunda parte do trabalho de laboratório.



Q5: [PN simulation] Simulate the Petri net you proposed in the previous questions. Comment whether your simulation software imply, or does not imply, privileged places or transitions while receiving random inputs. As helping references see the "5 philosophers dinner" simulation referred in Annex 1 and the template for simulating the keyboard "lab2_sim_kb_v6.zip" (to be completed by the groups).

Resposta: Após ter efectuado as alterações relevantes no código, além de *bugfixes* e de ter acrescentado um gráfico com as linhas premidas a cada instante, foi simulado o sistema para o conjunto de teclas predefinido (*demo*), tendo sido obtidos os resultados presentes na figura seguinte:

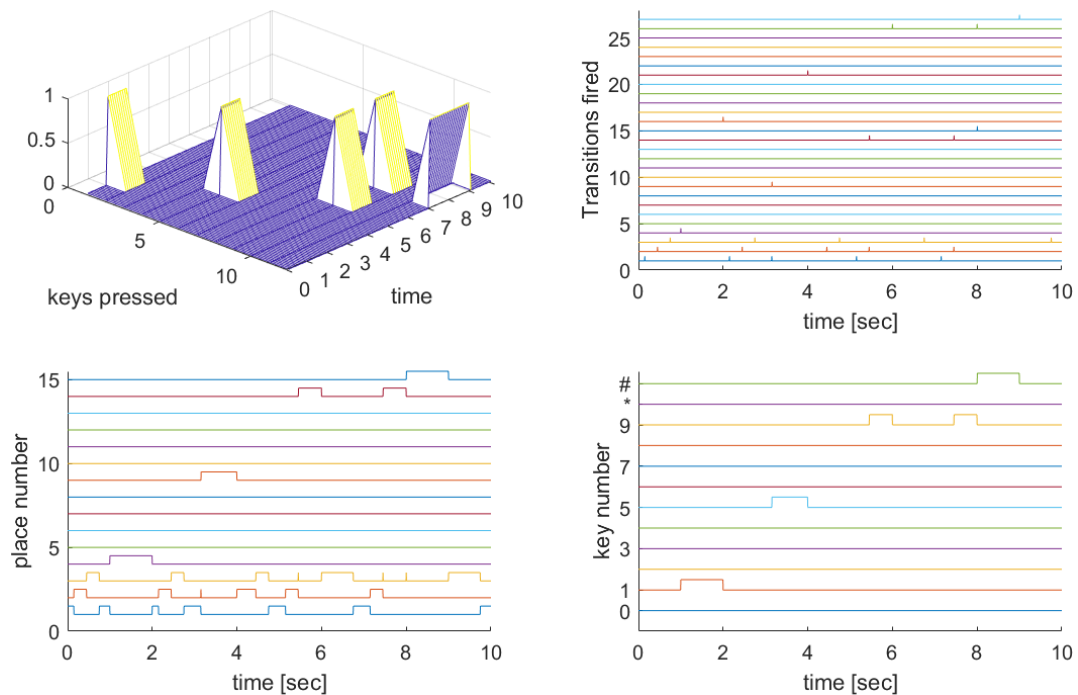


Figura 2 - Resultados da Simulação da Petri Net proposta

Na *Figura 2* observam-se os resultados obtidos na simulação da rede de Petri proposta para a leitura do teclado. A simulação foi alimentada pela lista de eventos (teclas pressionadas) seguinte:

$$x \rightarrow 1 \rightarrow x \rightarrow 5 \rightarrow x \rightarrow 9 \rightarrow x \rightarrow 9,12 \rightarrow x$$

Os x representam nenhuma tecla ser pressionada e 12 representa a tecla #.

É claro, no gráfico do canto inferior direito da *Figura 2*, que esta foi a sequência detetada na saída, com exceção do instante em que são premidas 2 teclas (Só é detetada a que dispara a transição com a numeração mais baixa, dado que o sistema faz disparos sequenciais). O gráfico imediatamente acima suporta esta conclusão uma vez que a transição com a numeração mais baixa é que é disparada.



Q6: [PLC test] See the demonstration of the Petri nets to PLC code converter, named `tst1_blink_on_off.m`, provided in the course SVN (see Annex 2). **(i)** Run the demo on a

PLC. **(ii)** Draw the graph of the Petri net considered in the demonstration. **(iii)** Show that one can detect keyboard keys 1, 2 or 3, ignoring the problem of multiple keys, by adding two places and four transitions, and redefining inputs and outputs. The detections of keys 1, 2 or 3 are suggested to be displayed by activating outputs `%q0.4.11`, `%q0.4.12` or `%q0.4.13` (or `%q0.3.25`, `%q0.3.26` or `%q0.3.27` in case of the 28FK output module), respectively. Suggestion: in `tst1_blink_on_off.m` replace the string `"3, zCode.outpMin+2"`, without quotation marks, by the string (also without quotation marks)

```
"3, zCode.outpMin+2; 4, zCode.outpMin+1; zCode.outpMin+2; zCode.outpMin+3; ..."
```

and observe that when place 4 is active it generates three outputs simultaneously

Resposta: Analisando o código fornecido, e em específico, a matriz de incidência e o estado inicial do sistema, verifica-se que a rede de Petri é a seguinte:

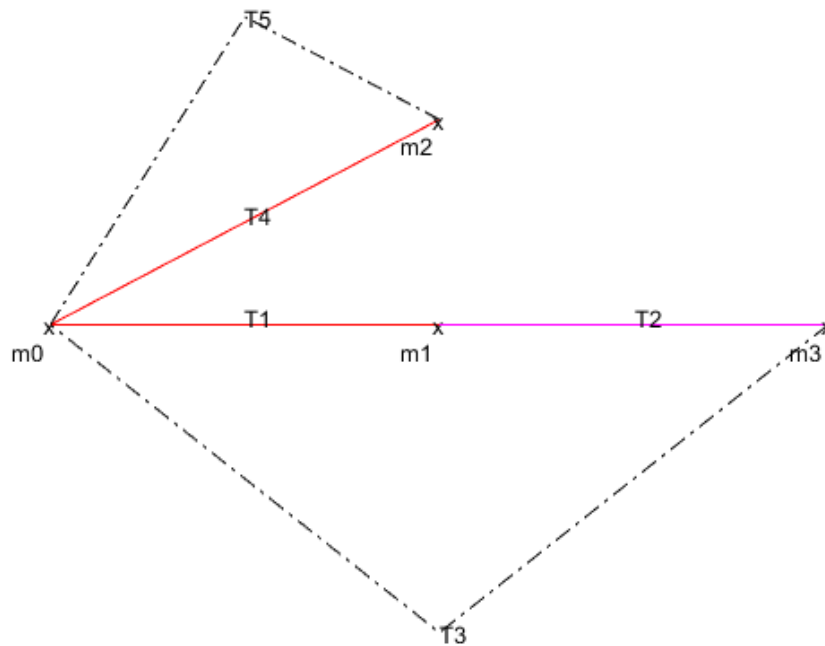


Figura 3 - Petri Net da demonstração

A rede de Petri presente na figura anterior tem 4 lugares e 5 transições e descreve um sistema que funciona como um alarme, que activa o *buzzer* cada vez que o estado lógico da entrada 0 é TRUE e desactiva o *buzzer* quando o mesmo estado é FALSE. Existe um ciclo de operação que passa pelos lugares 0, 1 e 3. No entanto, quando é activada a entrada 0, o sistema vai para o lugar 2, activando o *buzzer*, dado que a transição 4 é prioritária face às restantes.

Modificando a rede, é possível passar a detectar as teclas 1, 2 e 3 do teclado, caso o sistema deixe de analisar a entrada 0. Para tal, basta acrescentar 2 lugares e 4 transições, além de alterar o mapeamento de entradas e saídas do lugar 4.

Modificando a rede para desempenhar as funções referidas, obteve-se a rede de Petri que se apresenta a seguir:

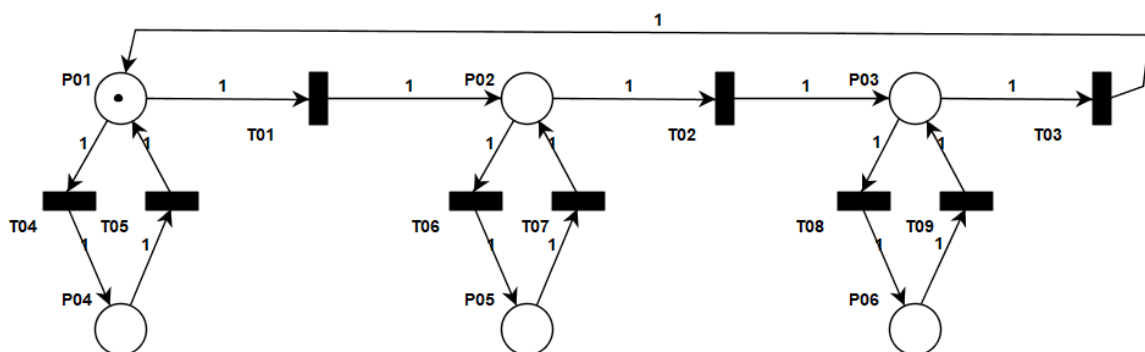


Figura 4 – Rede de Petri Modificada

A rede anterior é descrita pela seguinte matriz de incidências e estado inicial:

$$D = \begin{bmatrix} -1 & 0 & +1 & -1 & +1 & 0 & 0 & 0 & 0 \\ +1 & -1 & 0 & 0 & 0 & -1 & +1 & 0 & 0 \\ 0 & +1 & -1 & 0 & 0 & 0 & 0 & -1 & +1 \\ 0 & 0 & 0 & +1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & +1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & +1 & -1 \end{bmatrix} \quad \mu_0 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Foram ainda efectuadas as seguintes alterações:

- Nos lugares 1, 2 e 3 o sistema activa, respectivamente, as colunas 1, 2 e 3 do teclado (%q0.4.4, %q0.4.5 e %q0.4.6) e verifica a entrada %i0.2.4, que corresponde à primeira linha do teclado;
- Nos lugares 4, 5 e 6 o sistema activa as saídas sugeridas (%q0.4.11, %q0.4.12 e %q0.4.13), que correspondem respectivamente à detecção das teclas 1, 2 e 3;
- A activação da entrada %i0.2.4 despoleta as transições T04, T06 e T08 e a sua desactivação despoleta as transições de regresso aos lugares anteriores, T05, T07 e T09;
- As transições T04, T06 e T08 foram definidas como transições prioritárias.

Verificou-se que a alteração do mapeamento das saídas produzia os efeitos referidos no enunciado.

No entanto, decidimos não implementar tal mapeamento neste caso, por não ser relevante; de modo a melhorar o *datalogging* do funcionamento do sistema, optou-se por activar saídas diferentes para cada lugar.

De modo a efectuar o *datalogging* correctamente, foi ainda necessário efectuar alterações no código das rotinas produzido pelo conversor PN_TO_PLC, nomeadamente no que diz respeito ao mapeamento de memória: visto que a PLC usada tem memória que pode ser endereçada até %MW1023, todas as posições de memória usadas pelo programa foram remapeadas para posições acima de %MW300. Deste modo, é seguro proceder ao *datalogging* de até 100 eventos, sem que existam colisões de memória.

Com o auxílio da rotina (modificada) e código de *datalogger* (também modificado), foi possível obter os resultados da *Figura 5*:

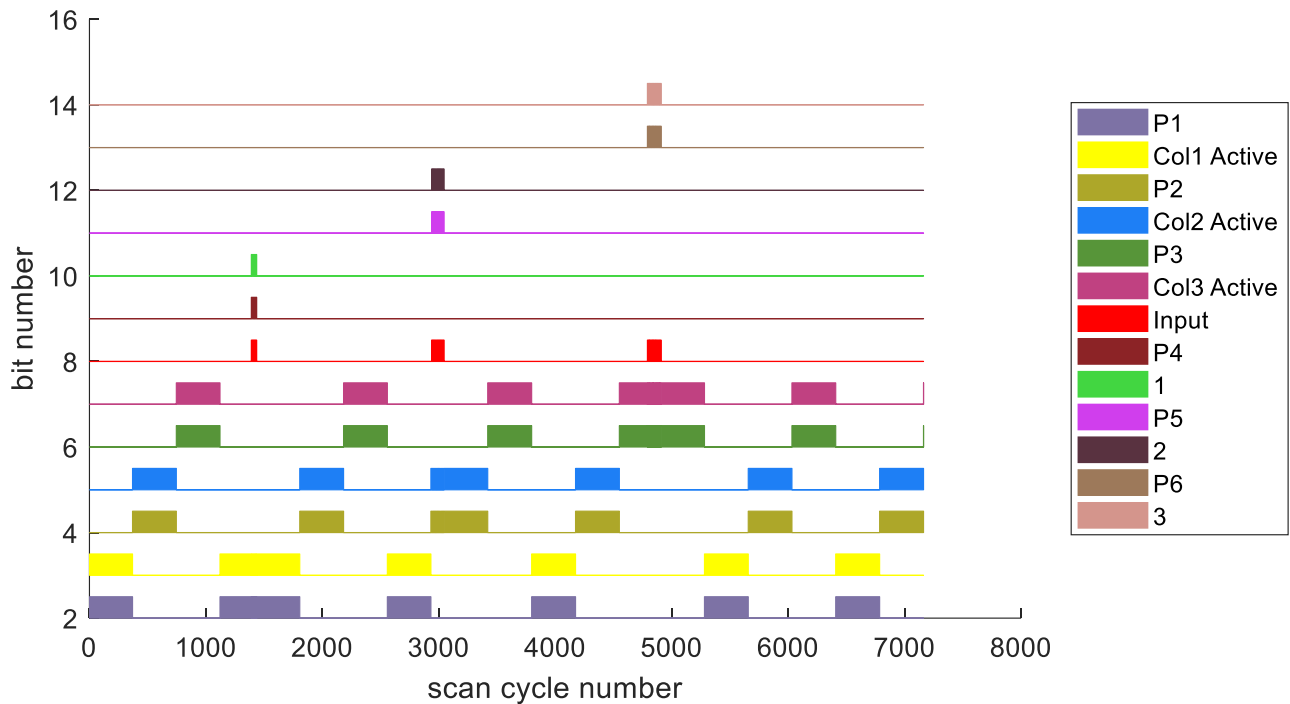


Figura 5 – Datalogging do funcionamento do sistema

No *datalogging* do sistema foram registadas as entradas (Input) e saídas (colunas do teclado e saídas %q0.4.11 a %q0.4.13) relevantes, além de representações booleanas (INT_TO_BOOL) das variáveis de memória correspondentes a cada lugar da rede de Petri (de modo a verificar a correcta activação das saídas do sistema a cada momento).

Da análise da figura anterior, verifica-se que o sistema funciona de acordo com o planeado:

- O sistema transita ciclicamente entre os lugares 1, 2 e 3, caso não seja premida nenhuma tecla;
- Nos lugares 1, 2 e 3 (representados por P1, P2 e P3), o sistema activa correctamente as colunas do teclado correctas;
- Com a activação da entrada %i0.2.4, o sistema transita para os lugares correspondentes a cada uma das teclas premidas (P4, P5 e P6), e activa as saídas correctas, correspondendo às teclas 1, 2 e 3.

Anexos

Anexo 1 - Incidence Matrix

[illegible]

Clique para Voltar