



Industrial Processes Automation

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

Winter Semester 2018/2019

Group: C2

70547 – João

Ferreira

75268 – Rúben

Tadeia

75987 – João

Ribafeita

80978 –

Gonçalo Pedro

2nd Laboratory Assignment ¹

Handling Faults in Keyboard Reading

Part C - DESs Synthesis, Supervised Control



Q1. Design a new Petri net that, instead of reading keyboard keys as the net proposed in part

A, simply *detects there are multiple keys pressed at the same time*. Draw the Petri using the editor (see part A guide). Please indicate clearly the specific place of the net chosen to flag multiple keys detected. Simulate a sequence of events demonstrating the functionality of the net.

Resposta: Por forma a evitar conflitos de teclas pressionadas simultaneamente, começou-se por desenhar uma rede que detetava simplesmente se múltiplas teclas eram pressionadas. A rede proposta encontra-se na *Figura 1*, sendo utilizada para levar o sistema a um estado de erro quando são pressionadas múltiplas, sendo o lugar 17 utilizado como *flag (sinalizador)* indicativa de múltiplas teclas pressionadas. Para disparar a transição T28 é necessário pressionar mais que uma tecla. O passo seguinte será a integração na rede de Petri original.

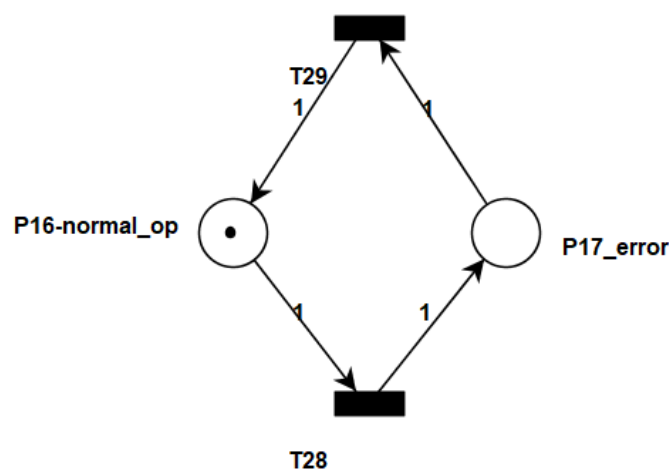


Figura 1 – Rede de Petri do Sinalizador

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



Q2. Combine the Petri net designed in Q1 with the Petri net design in part A. Design a supervisor controller to handle the *multiple keys pressed error*. Specify the linear constraint(s) that must be used. Solve the problem resorting to the place invariant methodologies studied in this course.

Resposta: As restrições lineares impostas referem-se aos lugares de validação e ao lugar de erro não poderem estar ativos simultaneamente. Assim:

$$\mu_k + \mu_{17} \leq 1, k \in \{4,5,6,7,8,9,10,11,12,13,14,15\}$$

Representa a restrição imposta onde μ_k é o número de marcas no lugar k.

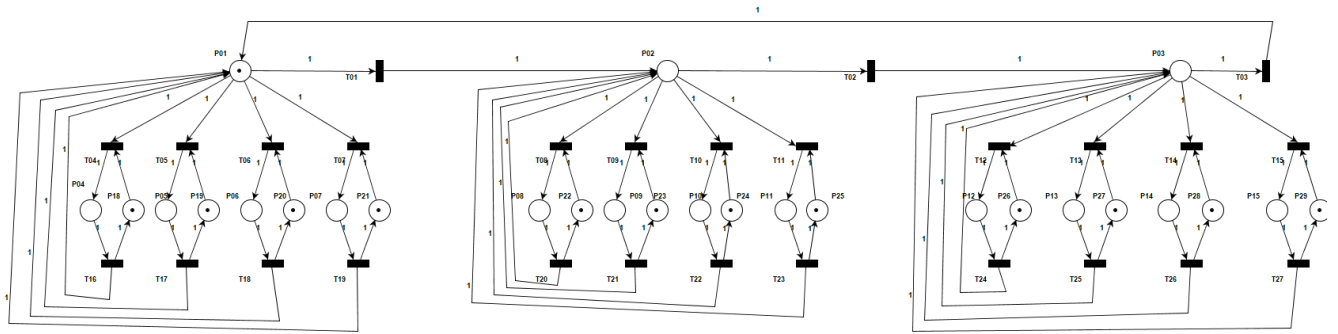


Figura 2 – Rede de Petri combinada

O supervisor irá impedir o sistema de atingir marcas que não satisfaçam:

$$L\mu_p \leq b, b = 1$$

Em que L tem tantas linhas como restrições e tantas colunas como lugares, sendo μ_p o vetor das marcações da rede. Adicionalmente, para que o supervisor funcione, tem de se verificar:

$$L\mu_{p_0} - b \leq 0$$

Em que μ_{p_0} é o vetor de marcações iniciais da rede combinada (17x1). Se esta condição não se verificar não é possível impor as restrições. Uma vez que μ_{p_0} apenas difere de zero (sendo igual a 1) na 1ª e 16ª posições, correspondentes à primeira coluna ativa e teclas não pressionadas simultaneamente.

$$L\mu_p = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \\ \mu_4 \\ \mu_5 \\ \mu_6 \\ \mu_7 \\ \mu_8 \\ \mu_9 \\ \mu_{10} \\ \mu_{11} \\ \mu_{12} \\ \mu_{13} \\ \mu_{14} \\ \mu_{15} \\ \mu_{16} \\ \mu_{17} \end{bmatrix}$$

Observa-se então que para o vetor μ_{p_0} descrito a condição inicial de funcionamento do supervisor se verifica.



Q3. Design the complete resulting Petri net (part A and part C, Q1 and Q2). Identify clearly the supervisor.

Resposta: A rede de petri completa, contendo o supervisor, pode ser observado abaixo na *Figura 3*. Como dito na questão 1, o supervisor encontra-se na parte central desta imagem, ou seja, são os 12 lugares que possuem atualmente uma marca (estados da direita para cada par de estados).

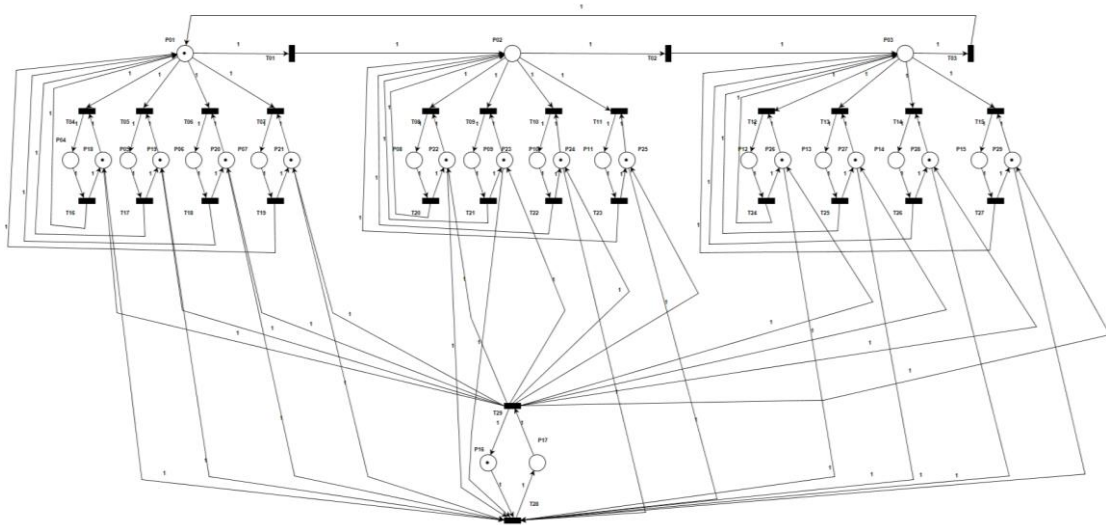


Figura 3 - Rede de Petri desenvolvida para controlo de teclado, com supervisor



Q4. Discuss briefly the properties relevant for the resulting DES, combining the original Petri net (part A), the multiple keys indicators (part C) and the supervisor.

Resposta: Vamos apresentar agora a análise das propriedades de petri nets pela sua ordem de importância, ou seja, da seguinte ordem:

1. Reachability
2. Liveness
3. Temporal Invariance
4. Boundedness
5. Conservation
6. Safeness
7. Coverability

Quanto à **Reachability** (#1) podemos ver pela *Figura 4* que é possível chegar a um dado estado do reachable set a partir de um qualquer que esteja no reachable set, ou seja, nunca ficamos presos num estado.

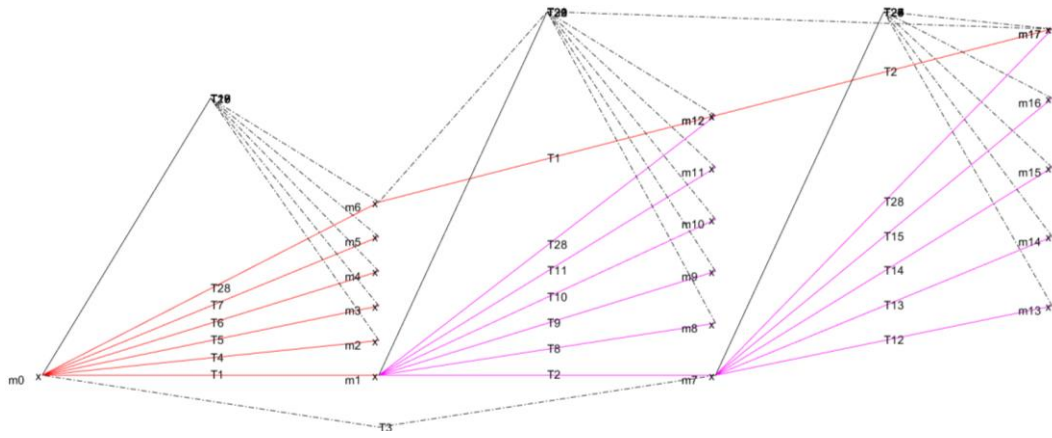


Figura 4 - Reachability Tree

Sobre a **Liveness** (#2) podemos facilmente por observação da *Figura 3*, afirmar que todas as transições são de nível 4, pois não só se conseguem disparar todas as transições infinitas vezes, como nenhuma das transições impede que outra transição seja disparada. Em suma, se todas as transições são de nível 4, a petri net também o será.

Relativamente à **Temporal Invariance** (#3), esta implica um estudo mais em detalhe.

É necessário resolver a seguinte equação, e obter um vetor de **q**, que resolva a seguinte equação.

$$Dq = 0$$

Em que:

$$q = [q_1 \ q_2 \ q_3 \ q_4 \ q_5 \ q_6 \ q_7 \ q_8 \ q_9 \ q_{10} \ q_{11} \ q_{12} \ q_{13} \ q_{14} \ q_{15} \ q_{16} \ q_{17} \ q_{18} \ q_{19} \ \dots \ q_{29}]^T$$

A matriz de incidências D é apresentada nos anexos dada a sua dimensão. ([Clique aqui para ir para os anexos](#))

Utilizando um método automático para determinação das soluções da equação obtiveram-se **14 soluções** para o vetor q. O código que realiza esta simples tarefa pode ser encontrado no zip na pasta “*código-auxiliar*”. Uma das soluções obtidas delas:

$$q = [1 \ 1 \ 1 \ 0 \ 0 \ 0 \ \dots \ 0]^T$$

Com este resultado, podemos então concluir que a nossa petri net é temporalmente invariante.

Da **Boundedness** (#4), podemos notar da *Figura 3* que esta rede de Petri tem no máximo 14 marcas em qualquer que seja o instante. No entanto, o número máximo de marcas por lugar é 1, o que faz com que a rede seja segura, concluindo-se assim, que a rede é **Bounded**, com $K=1$.

Sobre a **Conservation** (#5) dado que o número de marcas em toda a rede varia com o tempo a rede não é conservative. Por exemplo, se não houver flag de erro, então temos marcas em todos os places mutualmente exclusivos, ou seja, no caso do supervisor temos no máximo 12, mais 1 marca da flag e outra da petri net original. No caso de haver erro, existem no máximo 2 marcas, uma da rede original normal e outra da sub-rede da flag.

Relativamente à **Safeness** (#6), e após termos analisado a Boundedness e de se saber que a rede é **Bounded**, com $K=1$. Podemos então afirmar que esta rede é segura, dado que em nenhum instante de tempo existe um place com mais do que um marking.

A respeito da **Coverability** (#7) temos que perceber em primeiro lugar que, esta propriedade é uma generalização do conceito do estado de Reachability e está relacionado com o conceito de eventualmente se disparar uma transição. Como para a esmagadora maioria dos casos em que a rede é segura e de liveness correspondente a nível 4, não existe cobertura de estados. Neste caso, a nosso Reachability Tree é idêntica à Coverability Tree, para tal podemos observar a *Figura 4*. No entanto, considere-se esta explicação, para existir cobertura de estados, é necessário que haja um valor ω , tal que um dado estado μ' tenha em alguma situação um múltiplo de ω noutro estado μ'' . Não obstante, tal situação não se verifica, pois não existe a possibilidade de qualquer estado ter mais que 1 marca.



Q5. [Aggregation of Petri nets] Considering μ_p and D_p the state and incidence matrix of the Petri net designed in part A, and considering μ_d and D_d the state and incident matrix of the Petri net designed in part C (Q1), show that an enlarged net with state $\mu=[\mu_p^T \mu_d^T]^T$, running the two nets simultaneously, has an incidence matrix which is a function of the other two matrices, i.e. $D=f(D_p, D_d)$. Make a Matlab program that illustrates the equivalence of using D versus using D_p and D_d .

Resposta: Podemos observar a matriz D subdividida nas matrizes D_p e D_d ([Clique aqui para ir para os anexos](#)).

Relativamente à figura anterior, esta encontra-se dividida por vários retângulos coloridos. O retângulo amarelo corresponde à matriz de incidência relativamente à petri net utilizada na parte A. O retângulo roxo corresponde à matriz de incidência usada pela petri net de detecção de teclas múltiplas, o retângulo

verde corresponde à matriz de incidência D_p e por fim o retângulo vermelho corresponde à matriz de incidência do controlador do supervisor D_C .

Para demonstrarmos a equivalência das matrizes D relativamente a D_p e D_d , foi executado o programa “PN_Sim” com a matriz D e de seguida com as matrizes D_p e D_d , de onde se comprovou que os resultados eram os mesmos. O que seria de esperar, dado que em termos teóricos as duas matrizes juntas (D_p e D_d) foram a matriz D .



Q6. [Matlab simulation] Simulate in Matlab the complete Petri net that reads keys (developed in part A), while detecting and reacting to multiple keys pressed simultaneously (developed in part C). As a starting point, consider using the code distributed in a zip file “kb_sim_lab2.zip” (made available in the SVN of the course). Note that the Petri net is created by the group and therefore are empty the functions implementing the conversion of the mapping of the state to the actuation/outputs (PN_s2act.m, PN_s2yout.m), and the conversion of keyboard line signals to transitions (PN_tfir.m).

Resposta: Simulou-se no Matlab o código responsável pela leitura de teclas simultaneamente com o código que deteta múltiplas teclas.

Os resultados da simulação estão ilustrados na *Figura 5*, de notar que foram testadas teclas diferentes dos dados como exemplo pelo professor, portanto existem outras teclas a serem pressionadas ao mesmo tempo.

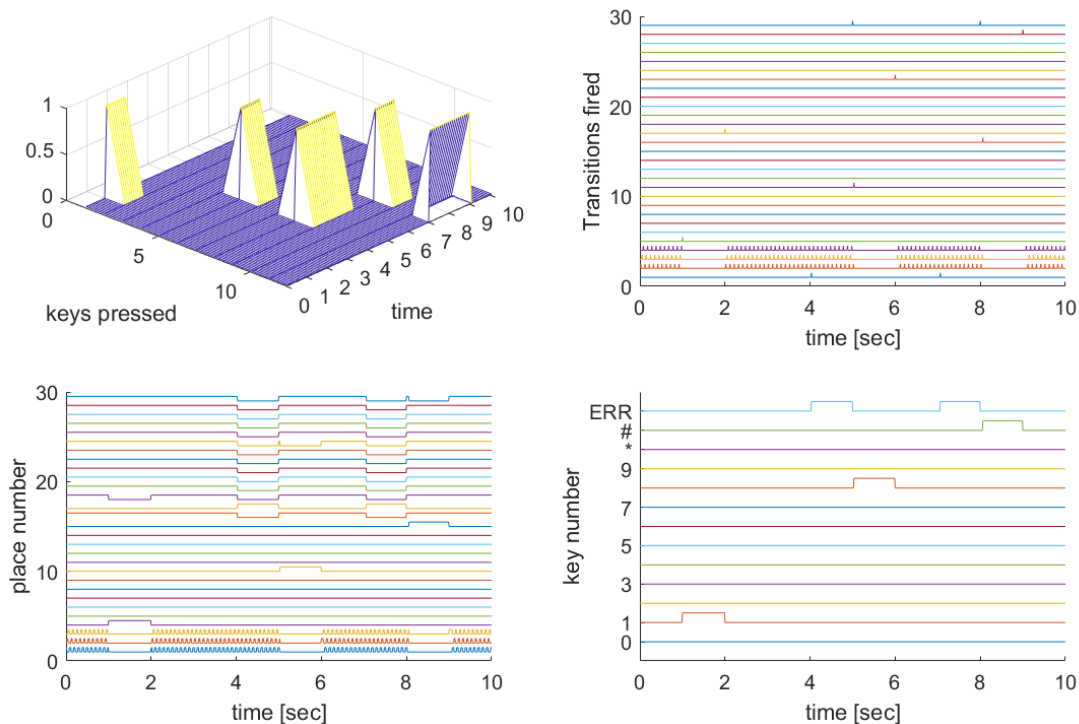


Figura 5 - Resultados da Simulação

De modo a comparar facilmente os resultados esperados dos lugares com as transições ativadas, foi gerada a *Figura 6*.

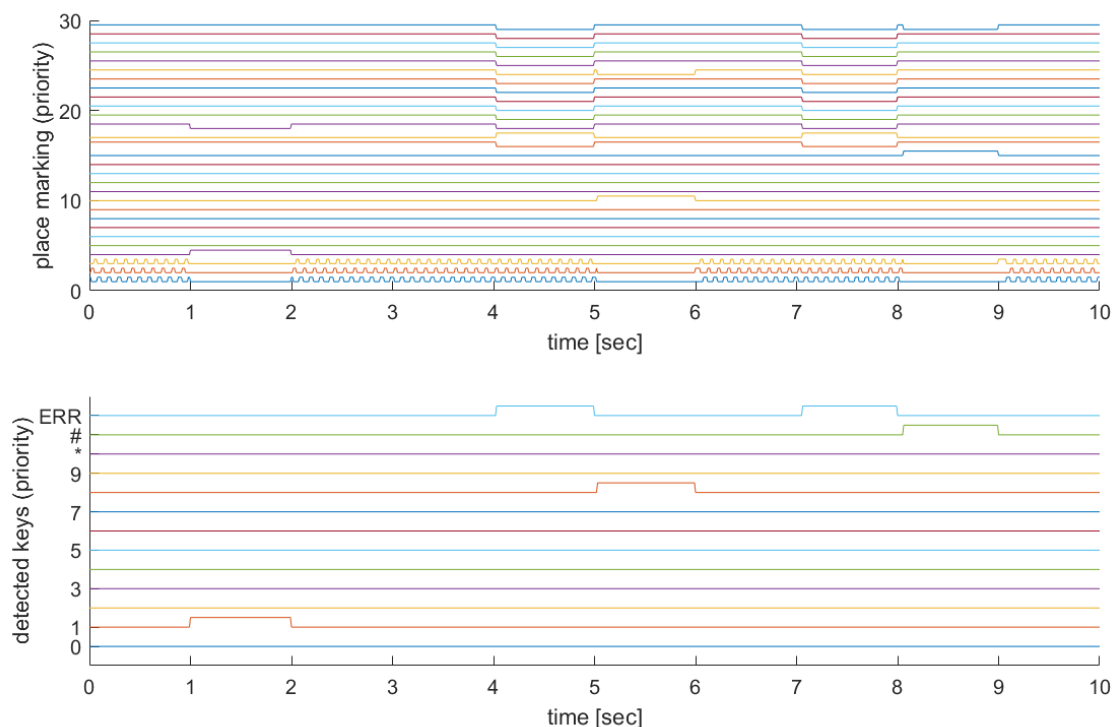


Figura 6 - Lugares e Transições

Em suma, e por observação da *Figura 5* e da *Figura 6*, podemos verificar o funcionamento do supervisor. Veja-se por exemplo, quando a marca transita para o estado do erro, todas as marcas do supervisor são consumidas, inviabilizando a detecção de teclas.



Q7. [Discussion of the PLC implementation] Discuss the implementation on the PLC of the supervisory control synthesized. Indicate eventual modifications needed in your key detection/validation program. Solve this question taking into account the Petri net to PLC code conversion tools introduced in the course classes.

De modo a poder implementar a rede de Petri com o respectivo supervisor na PLC, usando o código fornecido, seria recomendável separar o sistema em duas partes: a da rede de detecção do teclado e do seu supervisor, separada da rede que implementa a *flag*. No entanto, esta não foi a abordagem seguida, de modo a simplificar a implementação do código.

Além disto, seria necessário mapear as entradas (e transições disparadas) e saídas activadas em cada lugar da rede.

Enquanto para as saídas não existe um problema na implementação do código, o mesmo existe para o mapeamento das entradas. De modo a eliminar este problema, foi introduzida uma entrada “fictícia”

no código matlab, tendo sido depois modificada a rotina em ST para incluir a entrada “fictícia”, correspondente à activação de várias teclas (um OR de ANDs de todas as combinações de duas entradas possíveis, de modo a simplificar a implementação).

Foram ainda definidas como prioritárias as transições da rede que implementa a flag de erro, de modo a impôr o correcto funcionamento do sistema, bem como a activação das colunas do teclado quando a flag indica a existência de erro, evitando que as teclas deixem de ser detectadas e o sistema volte ao estado normal.

De modo a simplificar o *datalogging*, o mesmo foi efectuado apenas para os eventos que correspondem à activação de teclas da coluna 1.

De seguida apresentam-se os resultados *datalogging* efectuado com o código introduzido na parte A:

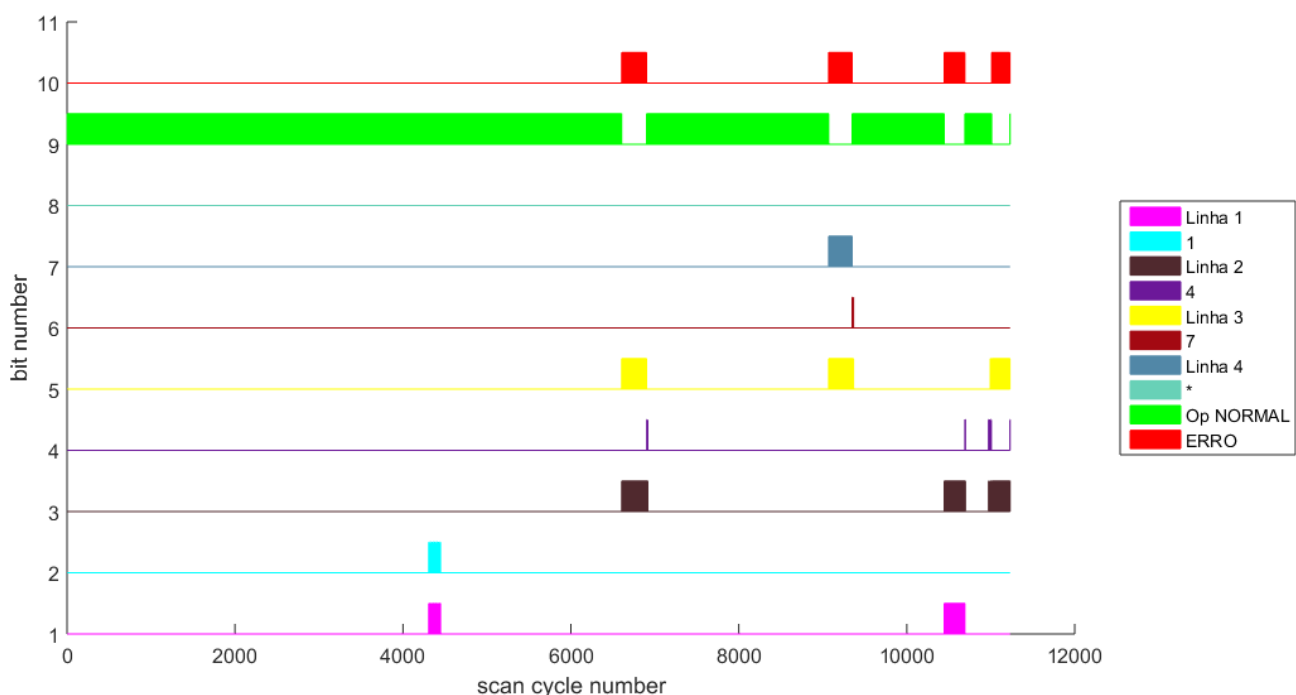


Figura 7 – Datalogging da operação da rede de Petri na PLC

Analisando a Figura 7, verifica-se que o sistema funciona de acordo com o pretendido:

- Quando apenas existe uma linha premida, a tecla correcta é detectada e o sistema transita para o lugar correspondente a essa tecla, e a rede da flag mantém-se no lugar correspondente ao estado normal (indicado por **Op NORMAL**);
- Quando existem duas teclas premidas, o sistema ignora-as, e a rede da flag transita para o estado de erro, saindo do mesmo quando as teclas deixam de ser premidas;
- Os “picos” que se verificam no final dos períodos de erro devem-se ao facto de as duas teclas não serem largadas ao mesmo tempo, pelo que existe uma fracção de segundo em que o sistema apenas “vê” uma tecla; uma implementação mais segura deveria prever um temporizador entre a saída do modo de erro e a pesquisa activa de teclas, de modo a evitar estes casos.

Anexos

Anexo 1 - Incidence Matrix

	T01	T02	T03	T04	T05	T06	T07	T08	T09	T10	T11	T12	T13	T14	T15	T16	T17	T18	T19	T20	T21	T22	T23	T24	T25	T26	T27	T28	T29
P01	-1	0	1	-1	-1	-1	-1	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0
P02	1	-1	0	0	0	0	0	-1	-1	-1	-1	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0
P03	0	1	-1	0	0	0	0	0	0	0	0	-1	-1	-1	-1	0	0	0	0	0	0	0	0	1	1	1	1	0	0
P04	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0
P05	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0
P06	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0
P07	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0
P08	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0
P09	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0
P10	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0
P11	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0
P12	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0
P13	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0
P14	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0
P15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	-1	0	0
P16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	1
P17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	-1
P18	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	-1	1
P19	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	-1	1
P20	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	-1	1
P21	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	-1	1
P22	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	-1	1

P23	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	-1	1
P24	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	-1	1
P25	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	-1	1
P26	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	-1	1
P27	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	-1	1
P28	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	1	0	-1	1
P29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	1	-1	1

Clique para Voltar

Anexo 2 - Incidence Matrix dividida por blocos

	T01	T02	T03	T04	T05	T06	T07	T08	T09	T10	T11	T12	T13	T14	T15	T16	T17	T18	T19	T20	T21	T22	T23	T24	T25	T26	T27	T28	T29	
P01	-1	0	1	-1	-1	-1	-1	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	
P02	1	-1	0	0	0	0	0	-1	-1	-1	-1	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	
P03	0	1	-1	0	0	0	0	0	0	0	0	-1	-1	-1	-1	0	0	0	0	0	0	0	0	1	1	1	1	0	0	
P04	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	
P05	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	
P06	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	
P07	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	
P08	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	
P09	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	
P10	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	
P11	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	
P12	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	
P13	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	
P14	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	
P15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	
P16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	1	0	
P17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	-1	0	
P18	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	-1	1	0
P19	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	-1	1	0
P20	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	-1	1	0
P21	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	-1	1	0
P22	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	-1	1	0
P23	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	-1	1	0
P24	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	-1	1	0
P25	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	-1	1	0
P26	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	-1	1	0

P27	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	-1	1
P28	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	-1	1
P29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	1	-1	1

Clique para Voltar