

# Trabalho Final - 2020/2

Rodrigo Naves Rios, 16/0144094

<sup>1</sup>Dep. Engenharia Elétrica – Universidade de Brasília (UnB)  
ENE0240 - Controle para Automação - Turma A

rodrigoner98@gmail.com

## 1. Modelagem por autômatos e código em C

### 1.1. Modelagem

Nesta seção, é mostrada a modelagem proposta do sistema de detecção de *buffer* por meio de autômatos. Especificamente, o sistema identifica se o *buffer* está cheio. Assim sendo, definem-se os eventos  $E = \{\text{SIN.ON}, \text{SIN.OFF}, \text{BM}, \text{SENSOR.ON}, \text{M.INV}, \text{EM.OFF}, \text{EM.ON}\}$ , os quais se relacionam à leitura de sensores e a operações externas:

1. SIN.ON: Ativação da botoeira geral NA que liga o sistema;
2. SIN.OFF: Ativação da botoeira geral NF que desliga o sistema;
3. SENSOR.ON: Ativação do sensor por meio de leitura;
4. BM: Ativação da botoeira NA que aciona o motor;
5. EM.OFF: Ativação da botoeira de emergência;
6. EM.ON: Ativação da botoeira de retorno da emergência;

Os estados, por sua vez são dados a seguir:  $X = \{\text{OFF}, \text{ON}, \text{M1}, \text{M2}, \text{MOFF}, \text{EM}\}$ . Nesse sentido, algumas escolhas de projeto são feitas: a) o estado ON indica que o sistema está apto a funcionar, bem como com seu respectivo sinalizador de funcionamento está ativo; b) escolhe-se um sentido preferencial de rotação, ou seja, o sentido em que o motor gira uma vez acionado, e que pode ser invertido por meio de acionamento de uma chave comutadora; c) escolhe-se um estado específico (MOFF) para que o sistema possa ser desativado por meio da respectiva botoeira. Uma breve explicação dos estados é dada:

1. OFF: Sistema desligado;
2. ON: Sistema apto a funcionar e com o respectivo sinalizador ligado;
3. M1: motor girando no sentido horário e com respectivo sinalizador ligado. Como o motor está acionado, o sinalizador que indica que o *buffer* não está cheio também está ligado;
4. M2: motor girando no sentido anti-horário e com respectivo sinalizador ligado. Como o motor está acionado, o sinalizador que indica que o *buffer* não está cheio também está ligado;
5. MOFF: motor desligado. Neste caso, o motor só é acionado - e transita para o estado M1 - por meio de botoeira.
6. EM: sistema em estado de emergência com seu sinalizador ligado;

A Figura 1 mostra a modelagem do referido sistema a eventos discretos por meio de autômatos.

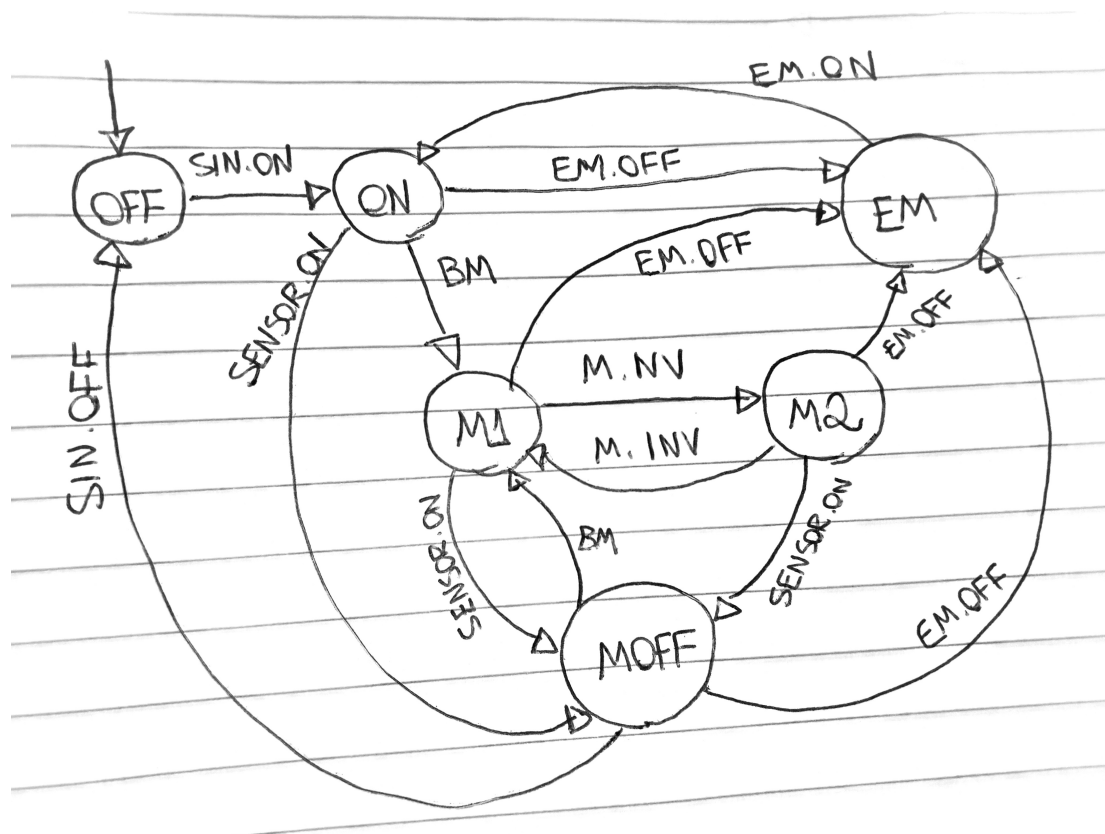


Figure 1. Modelagem por autômatos

## 1.2. Implementação em C

Tendo em vista a verificação da execução do sistema, implementou-se um código em C simulando a execução das tarefas em um microcontrolador. O sistema comporta como entradas as leituras de sensor e intervenções externas do usuário, ambos modelados como eventos na descrição feita acima. A Tabela 1 demonstra o comportamento para uma sequência de eventos pré-definida. Como as operações são sequenciais, o estado de chegada após a ocorrência de um determinado evento é o estado de partida da próxima linha na tabela. O evento "EV\_NONE" se refere a uma não ocorrência de evento. Em termos de modelagem, isto é equivalente a um auto-laço, uma vez que o estado permanece o mesmo. Além disso deve-se notar a correspondência entre os eventos definidos nesta implementação e aqueles definidos na seção 1.1, por exemplo, BM e EV\_BMOTOR. Nos outros casos, a correspondência é direta. É possível observar que a implementação se deu de maneira correta, respeitando as transições modeladas na seção 1.1. Além disso, é possível observar que as mensagens correspondem aos estados de chegada previstos para cada par estado de partida e evento.

## 2. Rede de Petri

### 2.1. Modelagem

Nesta seção, mostra-se a modelagem do SED por meio de redes de Petri. Os lugares da rede representam logicamente (1 ou 0) os estados dos dispositivos. Desse modo, temos os seguintes lugares: OFF, ON, SC, SN, M, SM1, SM2 e EM. O estado genérico é dado

Estado de Partida	Evento	Mensagem
OFF	EV_NONE	Evento ignorado
OFF	EV_SIN_ON	execução de sm_sin_on
ON	EV_SENSOR_ON	execução de sm_motor_off
MOFF	EV_BMOTOR	execução de sm_motor_on1
M1	EV_MOTOR_INV	execução de sm_motor_on2
M2	EV_NONE	Evento ignorado
M2	EV_MOTOR_INV	execução de sm_motor_on1
M1	EV_SENSOR_ON	execução de sm_motor_off
MOFF	EV_SIN_OFF	execução de sm_sin_off
OFF	EV_SIN_ON	execução de sm_sin_on

**Table 1. Resultados da implementação em C**

por:  $x = [x(\text{OFF}) \ x(\text{ON}) \ x(\text{SC}) \ x(\text{SN}) \ x(\text{M}) \ x(\text{SM1}) \ x(\text{SM2}) \ x(\text{EM})]$  e o estado inicial é  $x_0 = [1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]$ . A seguir, uma breve explicação dos lugares:

1. OFF: sistema inativo;
2. ON: sistema ativo;
3. SC: sinalizador ativo quando *buffer* está cheio;
4. SN: sinalizador ativo quando *buffer* não está cheio;
5. M: motor;
6. S1: sinalizador do motor no sentido horário;
7. S2: sinalizador do motor no sentido anti-horário;
8. EM: sinalizador de emergência;

As transições utilizadas são as seguintes:  $B_{on}, B_{off}, S_{on1}, S_{on2}, B_m, C1, C2, E_{m1}, E_{m2}, E_{m3}, E_{m4}, E_{m5}$ . No entanto, algumas delas recebem os mesmos rótulos:  $l(E_{m1}) = l(E_{m2}) = l(E_{m3}) = l(E_{m4}) = l(E_{m5}) = E_{moff}$ , já que todas representam o acionamento de emergência. Além disso,  $l(S_{on1}) = l(S_{on2}) = S_{on}$ . A seguir, uma breve explicação das transições:

1.  $B_{on}$ : acionamento da botoeira de início;
2.  $B_{off}$ : acionamento da botoeira de término;
3.  $S_{on}$ : ativação do sensor;
4.  $B_m$ : acionamento da botoeira que liga o motor;
5. C1: chaveamento do motor para o sentido anti-horário;
6. C2: chaveamento do motor para o sentido horário;
7.  $E_{moff}$ : acionamento do botão de emergência;
8.  $E_{mon}$ : acionamento do botão de retorno da emergência;

A Figura 2 mostra a estrutura da rede de Petri. Tendo em vista que nela a botoeira que aciona o motor pode ser acionada indefinidamente, tem-se que algumas transições indesejadas poderiam ocorrer - como a inversão do sentido do motor imediatamente após a ocorrência de uma emergência. De modo a corrigir isto, cria-se um novo lugar,  $I_m$  (início do motor), para limitar esse acionamento. Para tanto, criam-se os arcos  $(B_{on}, I_m)$  e  $(I_m, B_m)$ . Assim, temos:  $x = [x(I_m) \ x(\text{OFF}) \ x(\text{ON}) \ x(\text{SC}) \ x(\text{SN}) \ x(\text{M}) \ x(\text{SM1}) \ x(\text{SM2}) \ x(\text{EM})]$ . Além disso, considera-se que o *buffer* não inicia cheio e que o estado é desligado somente quando o motor está desligado.

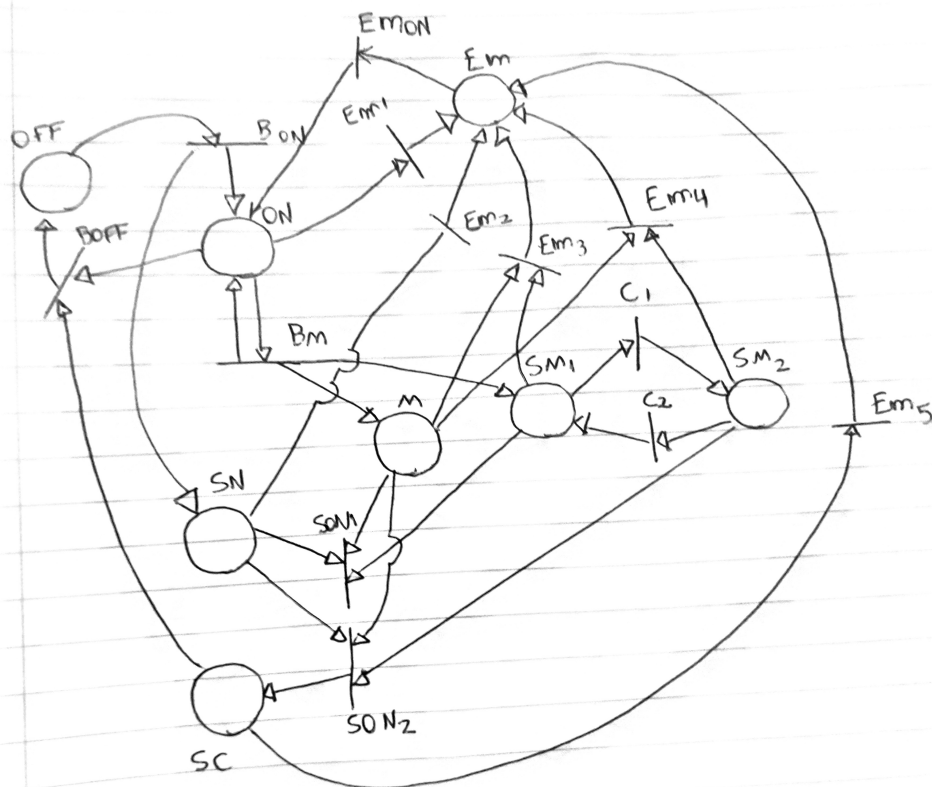


Figure 2. Estrutura da Rede de Petri

## 2.2. Implementação em Matlab

Neste ponto, são mostrados os resultados para a implementação em Matlab. Primeiramente, criou-se uma função que indica se dada transição é válida a partir de certo estado. Esta função é utilizada por outras duas: a que mostra todas as sequências de transições até um número N de disparos e a que indica qual o estado a que se chega a partir de uma dada sequência de disparos, caso esta seja válida. Os resultados para a primeira com  $N = 4$  são dispostos na Tabela 2.

## 3.

A Tabela 3 mostra alguns resultados para a segunda implementação.

Tamanho	Sequências
1	$B_{on}$
2	$B_{on}B_m, B_{on}E_{moff}$
3	$B_{on}B_mC1, B_{on}B_mE_{moff}, B_{on}B_mS_{on}, B_{on}E_{moff}E_{mon}$
4	$B_{on}B_mC1C2, B_{on}B_mC1E_{moff}, B_{on}B_mE_{moff}C1,$ $B_{on}B_mE_{moff}E_{mon}, B_{on}B_mE_{moff}S_{on},$ $B_{on}B_mS_{on}B_{off}, B_{on}B_mS_{on}E_{moff},$ $B_{on}E_{moff}E_{mon}B_m, B_{on}E_{moff}E_{mon}E_{mon}$

Table 2. Todas as sequências de até quatro disparos

Disparos	Estado alcançado
$B_{on}, B_m, S_{on}$	[0 0 1 1 0 0 0 0]
$B_m$	Transição inválida
$B_{on}$	[1 0 1 0 1 0 0 0]
$B_{on}, E_{moff}, E_{mon}$	1 0 1 0 1 0 0 0]
$B_{on}, B_m, C1, C2, S_{on}, B_{off}$	[0 1 0 0 0 0 0 0]

**Table 3. Estados alcançados**

#### 4. Linguagem Ladder

Por fim, implementa-se o mesmo SED em linguagem ladder. Abaixo, resume-se o resultado para algumas sequências de intervenções externas e de leitura do sensor. LED 1 refere-se ao sinalizador de quando *buffer* não está cheio e LED 2 de quando está cheio. A chave indica a comutação do sentido de rotação do motor.

- Start Button: Led Funcionamento ON, LED 1 ON
- Start button, Botoeira Motor: Led Funcionamento ON, LED1 ON, LED Motor horario ON
- Start Button, Botoeira Motor, Sensor ON: Led Funcionamento ON, LED2 ON (piscando)
- Start Button, Botoeira Motor, Chave: Led Funcionamento ON, LED 1 ON, LED Motor anti-horario ON