

# Sistemas Embebidos

## *Trabalho Prático 4*

**Carlos Abreu<sup>1</sup> e João Faria<sup>2</sup>**

<sup>1</sup>[cabreu@estg.ipvc.pt](mailto:cabreu@estg.ipvc.pt)

<sup>2</sup>[joao.pedro.faria@estg.ipvc.pt](mailto:joao.pedro.faria@estg.ipvc.pt)

Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Viana do Castelo

Instituto Politécnico de Viana do Castelo  
Escola Superior de Tecnologia e Gestão  
2024

Carlos Abreu  
[www.estg.ipvc.pt/~cabreu](http://www.estg.ipvc.pt/~cabreu)

**Curso:**

Licenciatura em Engenharia de Redes e Sistemas de Computadores

## Objetivo Pedagógico

Implementação de máquinas de estados.

### Sumário:

Duração: 3 horas

1. Compreender de forma genérica a diferença entre máquinas de estados de *Moore* e *Mealy*.
2. Representação de sistemas com recurso a diagramas de estados e diagramas de fluxo.
3. Implementação de máquinas de estados no Atmega328p.

## 1. FSM - Introdução

As máquina de estados em sistema embebidos são uma forma de descrever o modelo comportamental de um sistema. São compostas por um número finito de estados e habitualmente designadas *finite-state machine (FSM)*. Com base no estado atual e nas entradas do sistema, a máquina realiza transições de estado e produz as saídas pretendidas. Tal como representado na figura 1, existem dois tipos de máquinas de estados:

- *Moore* - As saídas são determinadas exclusivamente com base no estado atual.
- *Mealy* - As saídas dependem não só do estado atual mas também das entradas.

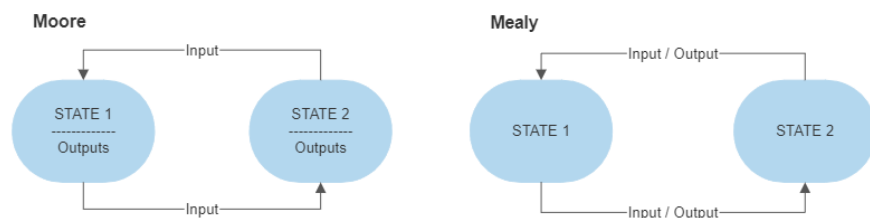
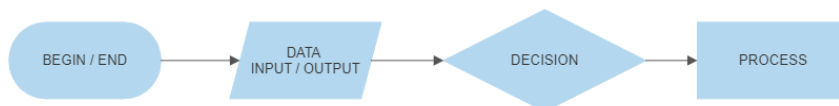


Figura 1

Representação genérica de máquinas de estados de *Moore* e *Mealy*

Depois de definida a máquina de estados é necessário definir a arquitetura do sistemas e detalhar a sua implementação. Um fluxograma permite representar visualmente um fluxo de dados e descrever de forma clara e objetiva um algoritmo. Um dos objetivos da utilização de fluxogramas é ajudar os programadores

a entender os algoritmos **ANTES** de começar a escrever o código. Na figura 2 encontram-se representados os símbolos usados com mais frequência na representação fluxogramas.



**Figura 2**

Símbolos mais comuns na representação de fluxogramas

## 2. FSM - Exercícios

**Exercício 1** Pretende-se implementar uma aplicação composta por um led verde (LedG), um led vermelho (LedR) e um botão de pressão (Btn) que cumpra com os seguintes requisitos:

- O botão é usado para acender e apagar o led verde (acende se estiver apagado e apaga se estiver aceso).
- O led verde não pode estar aceso durante períodos superiores a 6 segundos e é automaticamente desligado ao fim deste tempo.
- O led vermelho deve acender quando o led verde estiver ligado à mais de 4 segundos para indicar que se o utilizador não pressionar o botão para desligar o led verde este será desligado automaticamente quando passarem 6 segundos desde o momento em que foi ligado.

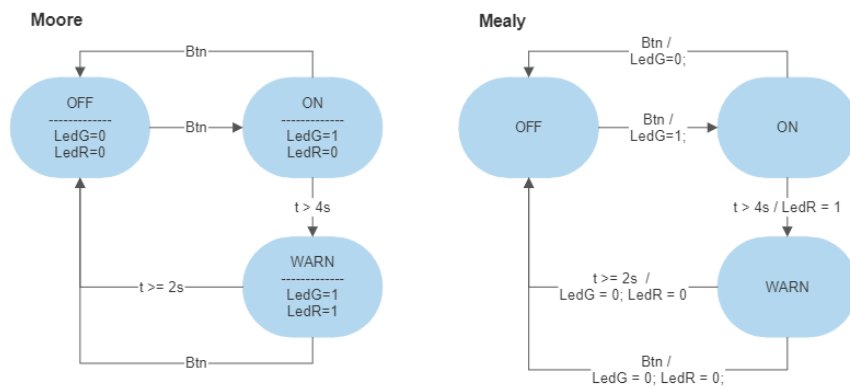
Na figura 3 encontram-se representados 2 possíveis diagramas de estados da aplicação, ambos consideram a existência de 3 estados distintos. Analise as soluções apresentadas e verifique se estão de acordo com os requisitos da aplicação.

De seguida são apresentados os fluxogramas de cada um dos diagramas de estados. Ambas as soluções tiram partido do TIMER 1 para controlar os intervalos de tempo entre as transições de estado e da utilização de interrupções externas para detetar eventos com origem no botão de pressão.

**1.1** Analise as duas soluções apresentadas e identifique se correspondem à implementação de máquinas de estados de Moore ou de Mealy.

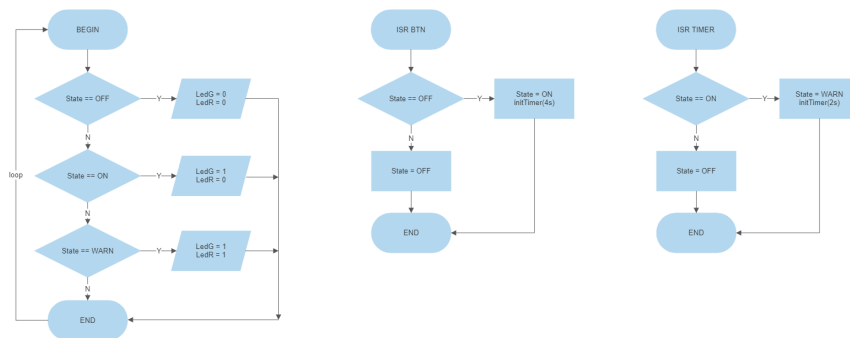
**1.2** Importe para a sua área de trabalho o circuito disponibilizado [neste link](#) e implemente em arduino C a solução 1 apresentada anteriormente.

**1.3** Teste o código implementado e não se esqueça de o comentar convenientemente.



**Figura 3**

Diagrama de estados da aplicação



**Figura 4**

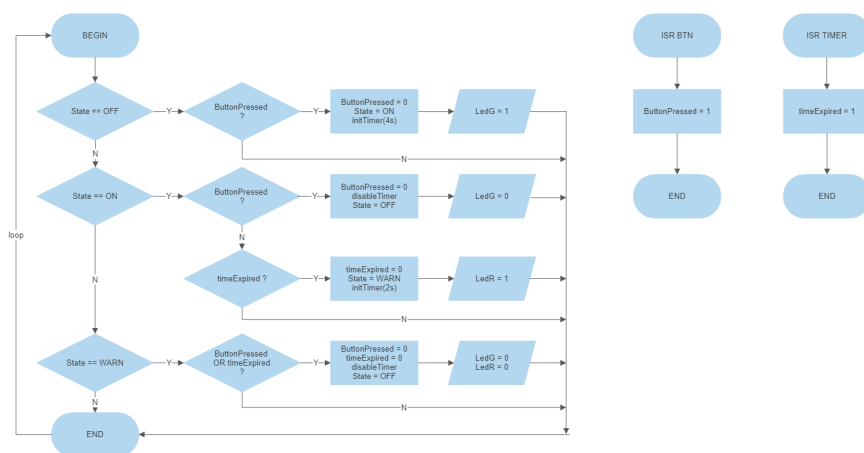
Diagramas de fluxo da aplicação - solução 1

**1.4** Importe para a sua área de trabalho o circuito disponibilizado [neste link](#) e implemente em arduino C a solução 2 apresentada anteriormente.

**1.5** Teste o código implementado e não se esqueça de o comentar convenientemente.

**Exercício 2** Pretende-se implementar uma aplicação que interage com um utilizador através de um LED e um botão de pressão que deverão ser configurados para operar como pinos digitais, saída e entrada, respetivamente, de forma a cumprir com os seguintes requisitos de aplicação:

- Enquanto o utilizador estiver a clicar no botão o LED deverá estar aceso, caso contrário o LED deverá estar apagado.
- Se o utilizador clicar no botão por um longo período ( $> 1$  segundo), o LED deverá piscar indefinidamente com um período de aproximadamente 200 ms.



**Figura 5**

Diagramas de fluxo da aplicação- solução 2

**2.1** Tendo em conta os requisitos da aplicação, quantos estados distintos identifica? Desenhe o respetivo diagrama de estados.

**2.2** Desenhe os fluxogramas das funções que considerar relevantes e que descrevam a operação lógica recorrendo à implementação de uma máquina de estados.

**2.3** Importe para a sua área de trabalho o circuito disponibilizado [neste link](#) e implemente a solução apresentada nos pontos anteriores.

**2.4** Teste o código implementado e não se esqueça de o comentar convenientemente.

**Exercício 3** Pretende-se implementar uma aplicação que simula um semáforo de controlo de velocidade. A interação com o utilizador é feita através de 3 LEDs (Verde, Amarelo e Vermelho) e 2 botões de pressão. Estes encontram-se ligados a pinos digitais configurados para operar como pinos de saída e entrada respetivamente. Os LEDs devem apresentar um comportamento análogo aos dos semáforos colocados na estrada para controlo de velocidade. O botão 1 é usado para ligar/desligar o controlo de velocidade. Quando o controlo de velocidade está desligado o led amarelo deve piscar continuamente com um período de 500 ms. O botão 2 é usado para simular a deteção de um veículo em excesso de velocidade. Quando o controlo de velocidade está ligado (semáforo verde) e o botão é carregado o semáforo deverá fazer a sequência: amarelo, vermelho e voltar ao verde. O semáforo deverá funcionar no seguinte modo:

- Ao inicializar o programa o controlo de velocidade deve estar desligado i.e. o LED amarelo deve ficar a piscar em modo intermitente com período de 500ms (250

*ms ligado e 250 ms desligado). Este processo repete-se indefinidamente até que o controlo de velocidade seja ligado.*

- *Ao clicar no botão 1 (ligar controlo de velocidade) o semáforo deve passar a verde*
- *Ao clicar no botão 2 (veículo em excesso de velocidade) o semáforo deve:*
  - *Ficar amarelo durante 1 segundo e transitar para vermelho*
  - *Ficar vermelho durante 3 segundos e transitar para verde*
  - *Ficar verde até que um dos botões seja clicado repetindo-se todo o processo.*
- *O botão 1 pode ser clicado a qualquer momento para desligar o controlo de velocidade e todo o processo repete-se indefinidamente.*

**3.1** *Tendo em conta os requisitos da aplicação, quantos estados distintos identifica? Desenhe o respetivo diagrama de estados.*

**3.2** *Desenhe os fluxogramas das funções que considerar relevantes e que descrevam a operação lógica recorrendo à implementação de uma máquina de estados.*

**3.3** *Importe para a sua área de trabalho o circuito disponibilizado [neste link](#) e implemente a solução apresentada nos pontos anteriores.*

**3.4** *Teste o código implementado e não se esqueça de o comentar convenientemente.*

**Exercício 4** *Pretende-se implementar uma aplicação que interage com um utilizador através de 6 LEDs vermelhos e um botão de pressão que deverão ser configurados para operar como pinos digitais, saída e entrada, respetivamente, de forma a cumprir com os seguintes requisitos de aplicação:*

- *Ao inicializar o programa, cada LED deverá acender um a um da esquerda para a direita durante aproximadamente 500 ms;*
- *Após o processo de inicialização, o LED mais à direita deverá ficar a piscar em modo intermitente, com um período de 500 ms. Todos os outros LEDs devem ficar apagados;*
- *Sempre que clicar no botão o modo de operação intermitente de operação deverá passar para o LED imediatamente à esquerda do que estava a piscar anteriormente. Todos os outros LEDs devem ficar apagados;*
- *Quando o modo de operação intermitente chegar ao LED mais à esquerda, deverá repetir os passos 2, 3, 4 e 5 indefinidamente;*

**4.1** *Tendo em conta os requisitos da aplicação, quantos estados distintos identifica? Desenhe o respetivo diagrama de estados.*

**4.2** *Desenhe os fluxogramas das funções que considerar relevantes e que descrevam a operação lógica recorrendo à implementação de uma máquina de estados.*

**4.3** *Importe para a sua área de trabalho o circuito disponibilizado [neste link](#) e implemente a solução apresentada nos pontos anteriores.*

**4.4** *Teste o código implementado e não se esqueça de o comentar convenientemente.*