

Universidade Tecnológica Federal do Paraná – Toledo
Engenharia da Computação – COENC

Sistemas Embarcados

Padrões de projeto Máquinas de estados

Tiago Piovesan Vendruscolo



Esta licença permite que outros remixem, adaptem e criem a partir do trabalho para fins não comerciais, desde que atribuam o devido crédito aos autores originais. [4.0 international](https://creativecommons.org/licenses/by-nc-nd/4.0/)

- O que é?
 - Máquinas de estados é uma estrutura muito utilizada em sistemas embarcados
 - O softwares desenvolvidos, em muitos casos não possuem fim, eles apenas ficam alternando entre os estados e por último entram em um estado de espera, geralmente de baixo consumo de energia.
- Muitos dispositivos utilizados em sistemas embarcados são dispositivos passivos
 - Monitoram e respondem a determinadas ações do ambiente.
 - A resposta depende do seu estado atual
 - Possuem um conjunto finito de estados e de entradas

■ Exemplo

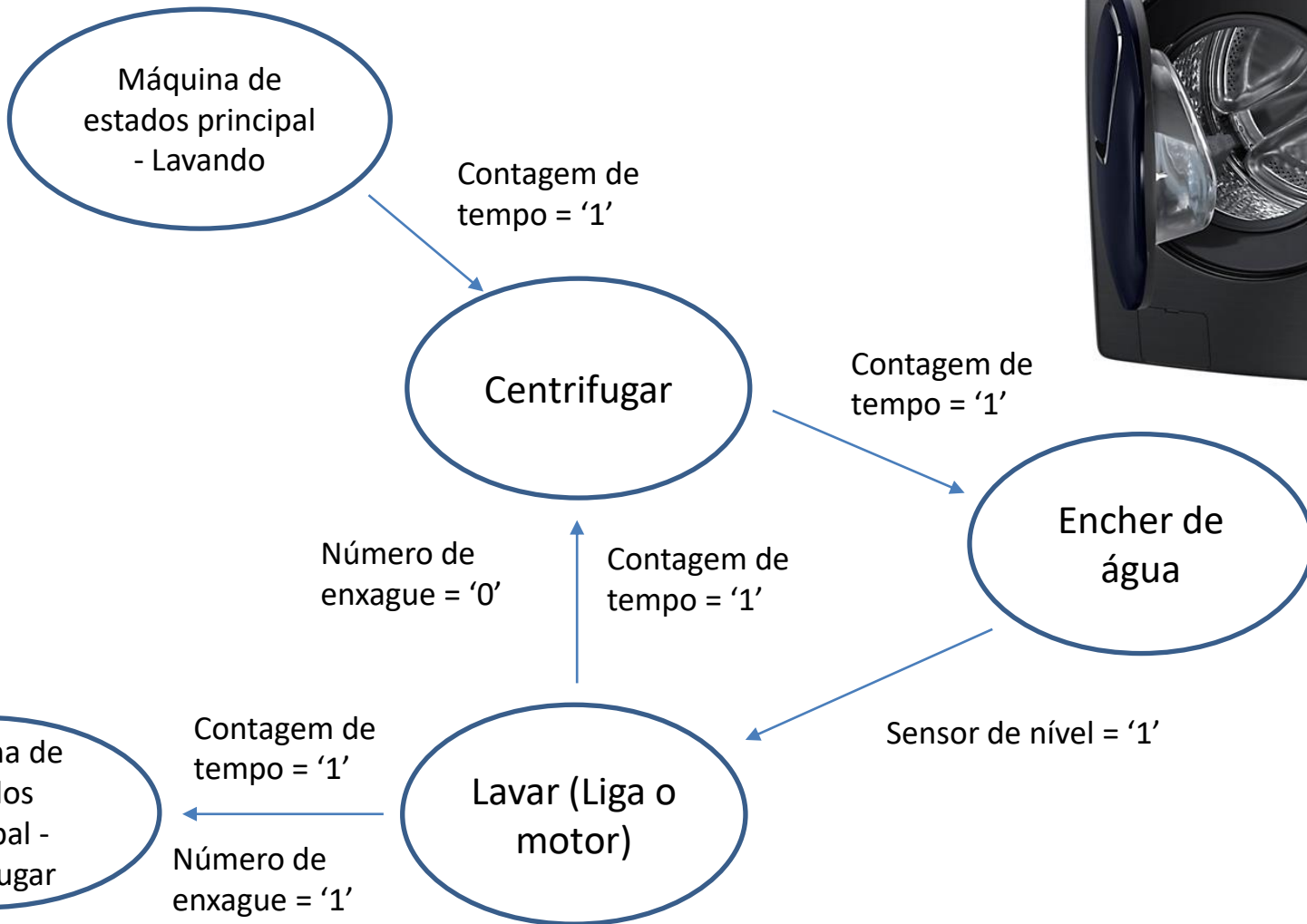
Máquina de lavar roupas

1. Início do ciclo (sem água);
2. Enchendo (até atingir o sensor de nível);
3. Molho (tempo estipulado);
4. Lavando (tempo estipulado);
5. Enxague - outra máquina de estados;
6. Centrifugar;
7. Fim do ciclo;



Máquinas de Estados

■ Enxague:



- Outros exemplos:
 - Circuitos sequenciais complexos;
 - Micro-ondas;
 - Semáforos;
 - Cafeteiras;
 - Etc.

Máquinas de Estados

- Termos e Definições:

- Diagrama de Estados

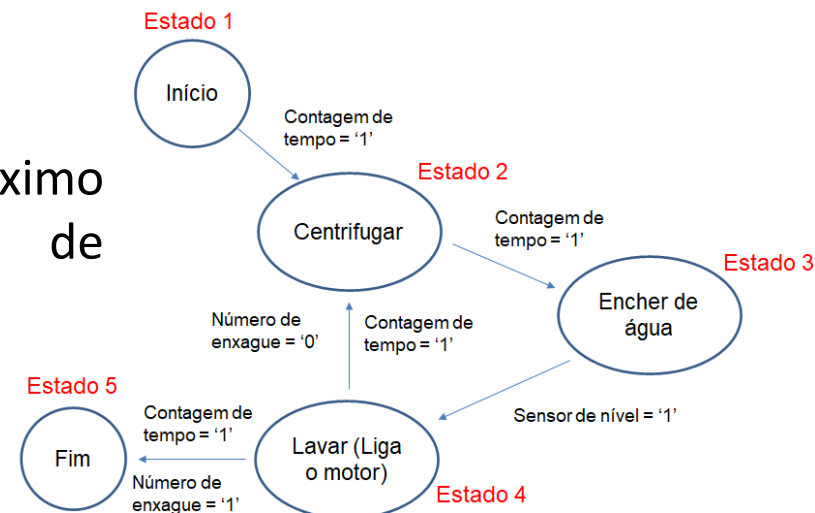
- Mostra a forma e a funcionalidade de uma máquina de estados. É composto pelos estados, transições e ações.

- Estados

- Conjunto de valores unicamente identificáveis. Armazenam informações sobre o passado. Cada estado realiza suas próprias ações

- Transição de Estados

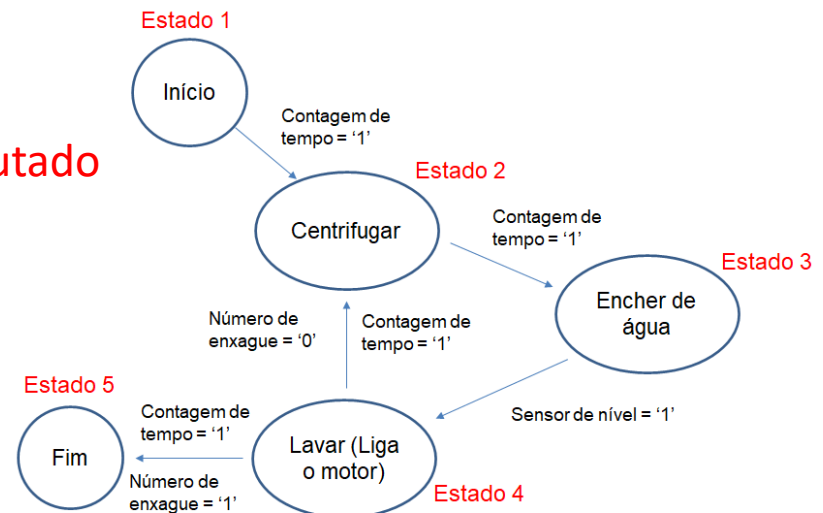
- A troca do estado atual para o próximo estado é baseado nas variáveis de entrada.



Máquinas de Estados

- Os estados podem fazer a seguintes ações:
 - Aguardar pela ocorrência de um evento
 - Reagir a um estímulo
 - Iniciar a execução de uma ação
 - Encerrar alguma ação

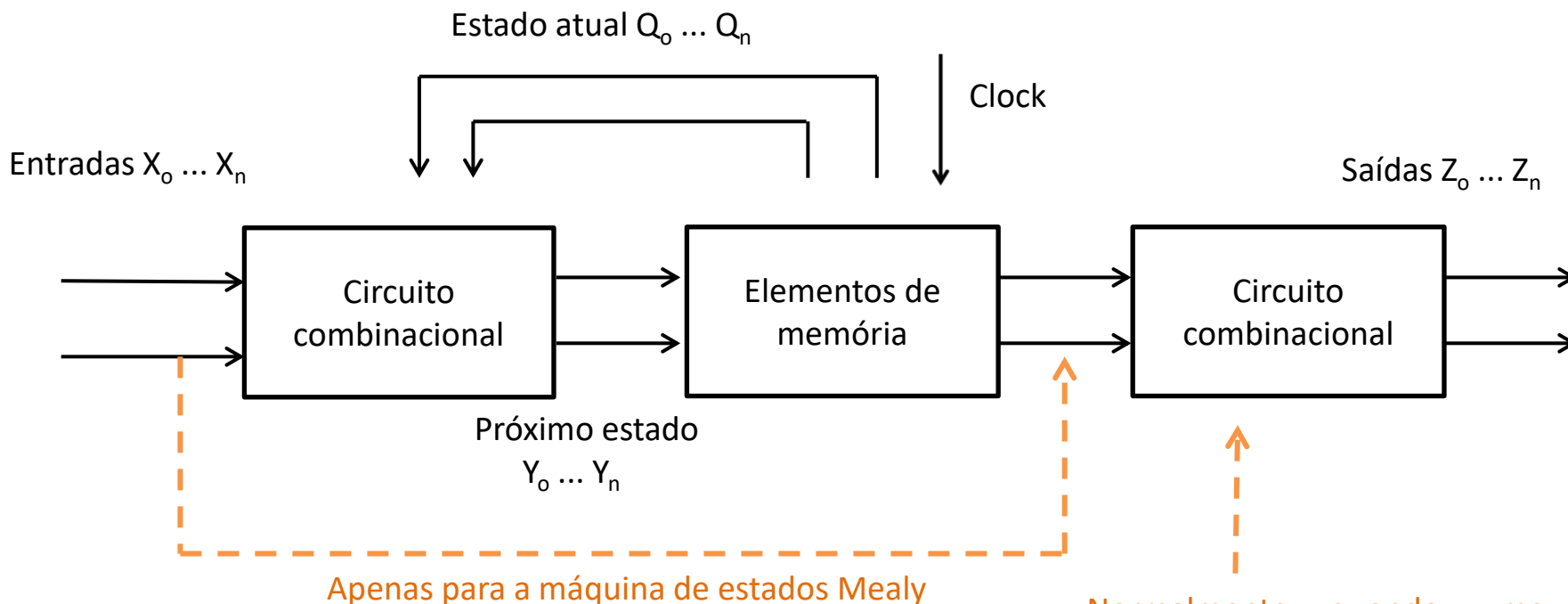
Apenas um estado pode ser executado por vez!



- Termos e Definições:
 - Estado Atual
 - O estado onde a máquina de estados está em determinado momento.
 - Próximo Estado
 - O estado para o qual a máquina de estados faz a próxima transição determinada pelas entradas (ou estado atual) no momento em que ocorre um sinal de clock ou mudança de variável.

- Tipos de máquinas de estados
 - Existem dois tipos de máquinas de estados que seguem as características básicas de uma máquina de estados, mas diferem em como as saídas são determinadas.
- Máquina Moore
 - As saídas são independentes das entradas, isto é, as saídas são determinadas somente a partir do estado atual da máquina de estados.
- Máquina Mealy
 - As saídas podem ser determinadas somente pelo estado atual ou pelo estado atual e as entradas atuais, isto é, as saídas são determinadas enquanto a máquina faz a transição de um estado para outro estado.

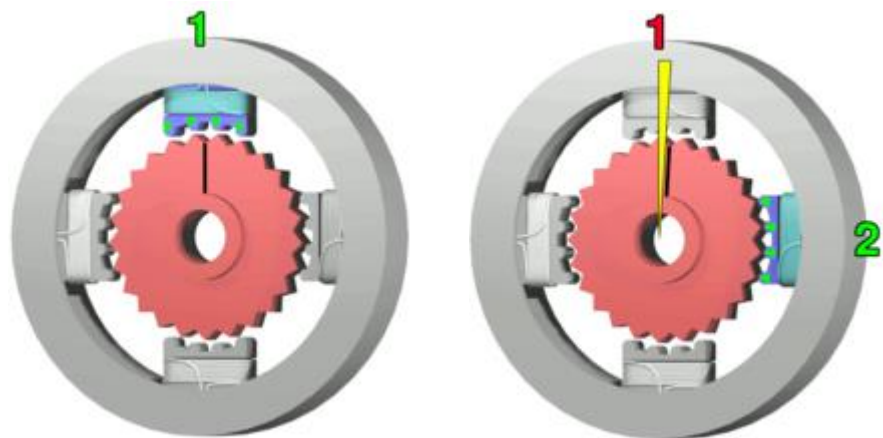
Tipos de máquinas de estados



- Existem duas formas tradicionais para implementar máquinas de estados
 - Switch / Case: Mais simples e normalmente mais usado, porém, conforme a máquina de estados vai ganhando complexidade, fica cada vez mais difícil de manter a organização e realizar mudanças.
 - Ponteiros de funções: É uma forma de dividir a máquina de estados em módulos, o que torna a manutenção mais fácil mesmo em máquina complexas. Também mantém o código mais limpo.

Máquinas de Estados - Motores de passo

- Aplicação típica: Motor de passo

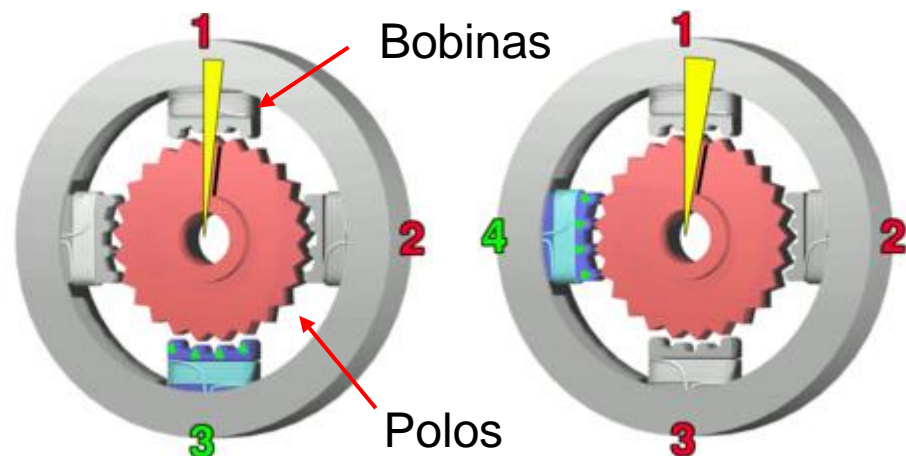


• Passo completo 1 (Full Step)

Nº do passo	B3	B2	B1	B0	Decimal
1	1	0	0	0	8
2	0	1	0	0	4
3	0	0	1	0	2
4	0	0	0	1	1

• Passo Completo 2 (Full Step)

Nº do passo	B3	B2	B1	B0	Decimal
1	1	1	0	0	12
2	0	1	1	0	6
3	0	0	1	1	3
4	1	0	0	1	9

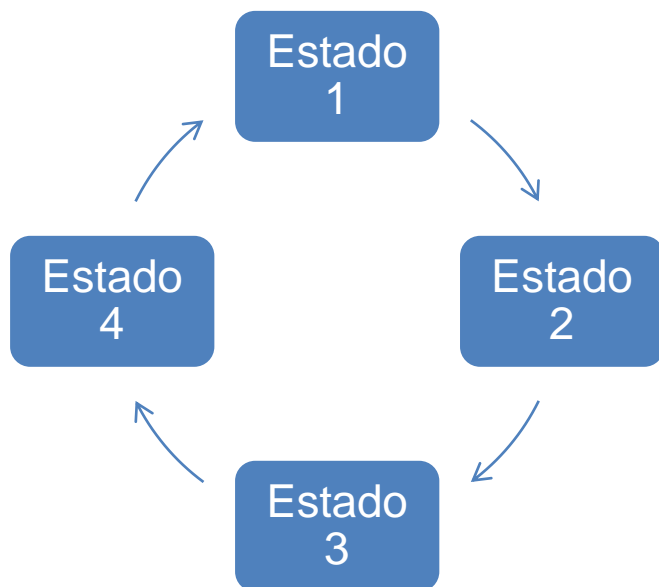


• Meio Passo (Half Step)

Nº do passo	B3	B2	B1	B0	Decimal
1	1	0	0	0	8
2	1	1	0	0	12
3	0	1	0	0	4
4	0	1	1	0	6
5	0	0	1	0	2
6	0	0	1	1	3
7	0	0	0	1	1
8	1	0	0	1	9

Máquinas de estados - Switch / Case

- Estrutura (exemplo):



- Passo completo 1 (Full Step)

Nº do passo	B3	B2	B1	B0	Decimal
1	1	0	0	0	8
2	0	1	0	0	4
3	0	0	1	0	2
4	0	0	0	1	1

```
typedef enum
{
    ESTADO_1 = 0,
    ESTADO_2,
    ...
} maquina_ESTADO;
maquina_ESTADO ESTADO = ESTADO_1;

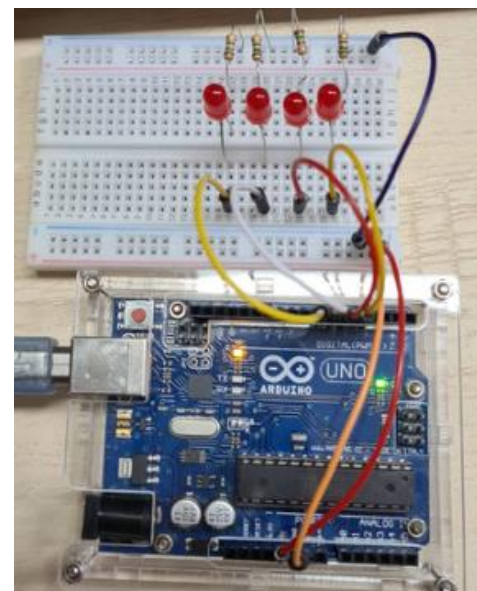
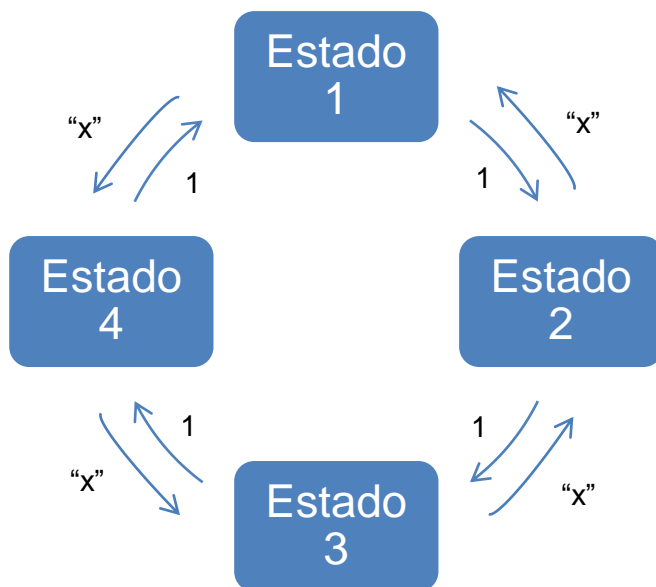
void maquina ()
{
    ...
    switch(ESTADO)
    {
        case ESTADO_1:
            if(input)
                state = ESTADO_3;
            else
                state = ESTADO_1;
            break;
        case ESTADO_2:
            ...
    }
    ...
}
```

Máquinas de estados - Switch / Case

- Exemplo 1: Faça o controle de um motor de passo utilizando passo completo de acordo com a tabela abaixo, utilizando os LED1 a 4. Quando enviar o comando '1' pela serial, o motor deve girar no sentido LED1 – LED4, caso contrário, no sentido inverso.

• Passo completo 1 (Full Step)

Nº do passo	B3	B2	B1	B0	Decimal
1	1	0	0	0	8
2	0	1	0	0	4
3	0	0	1	0	2
4	0	0	0	1	1



GPIO Pin	Dispositivo
4	LED1
5	LED2
6	LED3
7	LED4

Utilizando o Arduino – Serial

- Funções para a leitura da serial:
 - *Utilizado para verificar se possui uma nova informação na serial:*

```
Serial.available()
```

```
if(Serial.available()>0)
```

- *Função para ler a serial:*

```
Serial.read()
```

- *Função para ler um integer:*

```
Serial.parseInt();
```

```
x = Serial.parseInt();
```

- *Função para ler uma String*

```
x = Serial.readString();
```

Cuidado ao utilizar esses tipos de funções, pois ele ocupa o processador até ser concluído ou estourar o timeout.

Por padrão, tem o timeout de 1 segundo para empacotar a String (isso gera um atraso de 1 segundo no código). Pode ser alterado com `Serial.setTimeout(ms)` no setup. Na velocidade padrão de 9600, sendo 1 caractere = 8bit, pode receber até 1200 caracteres por segundo. Para receber mais, precisa aumentar o timeout, ou aumenta a velocidade de transmissão.

Máquinas de estados - Switch / Case

Código genérico

Se rodar esse software no motor de passo, tem algum problema?

```
#define LED1 4
#define LED2 5
#define LED3 6
#define LED4 7
```

```
void setup() {
    Serial.begin(9600);
    Serial.setTimeout(50);
    pinMode(LED1, OUTPUT);
    pinMode(LED2, OUTPUT);
    pinMode(LED3, OUTPUT);
    pinMode(LED4, OUTPUT);
}

int x;
void loop()
{
    typedef enum
    {
        ESTADO_1,
        ESTADO_2,
        ESTADO_3,
        ESTADO_4
    } maquina_ESTADO;

    maquina_ESTADO ESTADO = ESTADO_1;
```

Relembrando: a função delay está sendo utilizada apenas para fins didáticos, não é recomendado utilizar em situações reais.

```
while(1)
{
    if (Serial.available() > 0) {
        x = Serial.parseInt();
        Serial.println(x);
        delay(300);
        switch(ESTADO)
        {
            case ESTADO_1:
                digitalWrite(LED1, HIGH);
                digitalWrite(LED2, LOW);
                digitalWrite(LED3, LOW);
                digitalWrite(LED4, LOW);
                if(x==1){
                    ESTADO = ESTADO_2;
                }
                else{
                    ESTADO = ESTADO_4;
                }
                break;

            case ESTADO_2:
```

Código no moodle - complete o código

Máquinas de estados - Switch / Case

- Uma forma mais fácil é incrementar o estado (estado++), porém não é possível incrementar (typedef enum) diretamente em C++, apenas em C.

Código genérico

módulo de controle

módulo de saída

```
typedef enum
{
    ESTADO_1,
    ESTADO_2,
    ESTADO_3,
    ESTADO_4
} maquina_ESTADO;
maquina_ESTADO ESTADO = ESTADO_1;

while(1)
{
    if(leitura de uma flag, botao...)
    {ESTADO++;}
    else
    {ESTADO--;}

    switch(ESTADO)
    {
        case ESTADO_1:
            LED0=1;
            LED1=0;
            LED2=0;
            LED3=0;
            break;

        case ESTADO_2:
            LED0=0;
            LED1=1;
            LED2=0;
            LED3=0;
            break;

        (outros estados...)
    }
}
```

Máquinas de estados - Switch / Case

- Exemplo 2: Uma forma mais fácil é incrementar o estado (estado++), porém não é possível incrementar (typedef enum) diretamente em C++, apenas em C.

```
void loop()
{
    typedef enum
    {
        ESTADO_1,
        ESTADO_2,
        ESTADO_3,
        ESTADO_4
    } maquina_ESTADO;

    maquina_ESTADO ESTADO = ESTADO_1;

    while(1)
    {
        if (Serial.available() > 0) {
            x = Serial.parseInt();
            Serial.println(x);
            delay(300);
            if(x==1){
                estado_anterior++;
                if(estado_anterior >= 4) {estado_anterior=0;};
                ESTADO = (maquina_ESTADO) estado_anterior;
                estado_anterior = (uint8_t) ESTADO;
            }
            else{
                estado_anterior--;
                if(estado_anterior < 0) {estado_anterior=3;};
                ESTADO = (maquina_ESTADO) estado_anterior;
                estado_anterior = (uint8_t) ESTADO;
            }
        }
    }
}
```

Código no moodle - complete o código

Máquinas de estados - Switch / Case

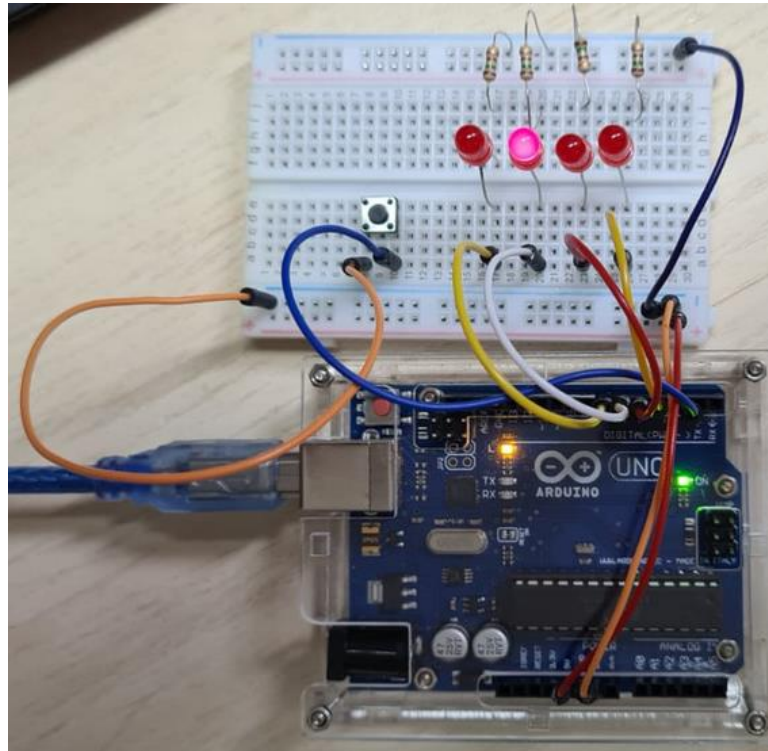
```
if(x==1){
    estado_anterior++;
    if(estado_anterior>=4){estado_anterior=0;};
    ESTADO = (maquina_ESTADO) estado_anterior;
    estado_anterior = (uint8_t) ESTADO;
}
else{
    estado_anterior--;
    if(estado_anterior<0){estado_anterior=3;};
    ESTADO = (maquina_ESTADO) estado_anterior;
    estado_anterior = (uint8_t) ESTADO;
}

switch(ESTADO)
{
    case ESTADO_1:
        digitalWrite(LED1, HIGH);
        digitalWrite(LED2, LOW);
        digitalWrite(LED3, LOW);
        digitalWrite(LED4, LOW);
        break;

    case ESTADO_2:
        digitalWrite(LED1, LOW);
        digitalWrite(LED2, HIGH);
        digitalWrite(LED3, LOW);
        digitalWrite(LED4, LOW);
        break;
}
```

Máquinas de estados - Switch / Case

- Exercício 1: Faça o controle de um motor de passo utilizando passo completo utilizando os LED1 a 4. Quando a entrada serial tiver o valor “1”, o motor deve girar no sentido LED1 – LED4, caso contrário, no sentido inverso. O controle também deve ter um botão de emergência (pino 2 – PULL-UP com interrupção) de forma que quando o botão for pressionado o motor deve parar imediatamente (todas as saídas em 0). Quando o botão for solto, o motor deve continuar de onde parou (analisar pelos LEDs).



Máquinas de estados - Switch / Case

```
void setup() {
    Serial.begin(9600);
    Serial.setTimeout(50);
    pinMode(LED1, OUTPUT);
    pinMode(LED2, OUTPUT);
    pinMode(LED3, OUTPUT);
    pinMode(LED4, OUTPUT);
    pinMode(BOTAO, INPUT_PULLUP);
    attachInterrupt(digitalPinToInterrupt(BOTAO), Botao_ISR, CHANGE);
}

int x;
int y = 1;
int temp;
int estado_anterior;

void Botao_ISR() {
    if(digitalRead(BOTAO)) {
        y=1;}
    else{y=0;}
}

while(1)
{
    if (Serial.available()>0) {
        x = Serial.parseInt();
        Serial.println(x);}

    delay(300);
    if(y==0) {
        ESTADO = ESTADO_5;}

    else if(x==1){
        estado_anterior++;
        if(estado_anterior>=4){estado_anterior=0;};
        ESTADO = (maquina_ESTADO) estado_anterior;
        estado_anterior = (uint8_t) ESTADO;}

    else {
        estado_anterior--;
        if(estado_anterior<0){estado_anterior=3;};
        ESTADO = (maquina_ESTADO) estado_anterior;
        estado_anterior = (uint8_t) ESTADO;}
```

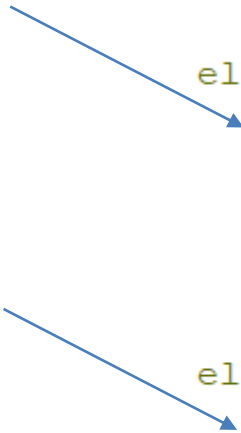
Máquinas de estados - Switch / Case

- Exercício 2: Continuando o exercício 1, altere o código de forma que, caso o botão de emergência fique pressionado por mais de 1 segundo, o Arduino seja resetado (WDT).

```
if(y==0){
    ESTADO = ESTADO_5;}

else if(x==1){
    wdt_reset();
    estado_anterior++;
    if(estado_anterior>=4){estado_anterior=0;};
    ESTADO = (maquina_ESTADO) estado_anterior;
    estado_anterior = (uint8_t) ESTADO;}

else {
    wdt_reset();
    estado_anterior--;
    if(estado_anterior<0){estado_anterior=3;};
    ESTADO = (maquina_ESTADO) estado_anterior;
    estado_anterior = (uint8_t) ESTADO;}
```



Máquinas de estados – Ponteiros de funções

Referências

- <https://sergioprado.org/maquina-de-estados-em-c/>
- <https://www.embarcados.com.br/maquina-de-estado/>