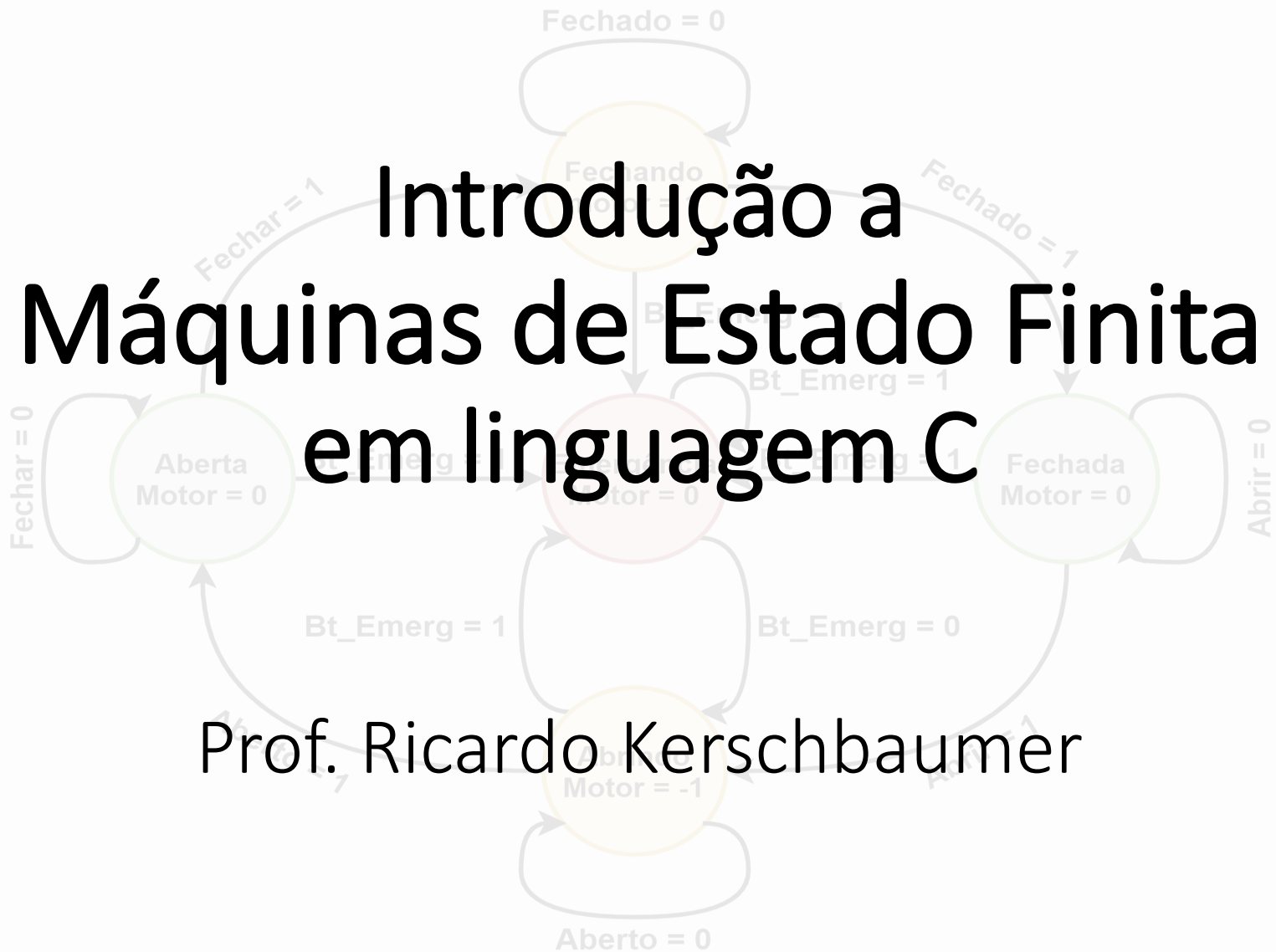




Introdução a Máquinas de Estado Finita em linguagem C



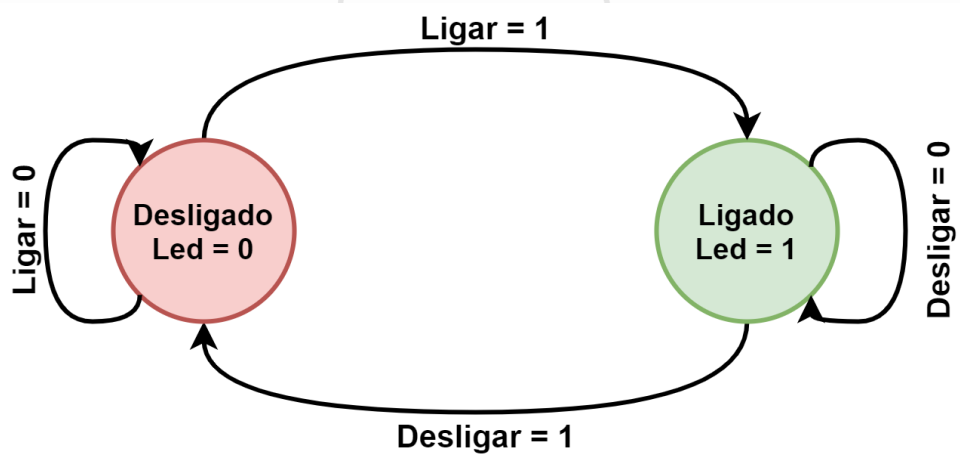
Prof. Ricardo Kerschbaumer



Máquinas de Estado Finita

- Uma máquina de estados finita (FSM - do inglês **F**inite **S**tate **M**achine) ou autômato finito é usado para representar programas de computadores ou circuitos lógicos.
- O conceito é concebido como uma máquina abstrata que deve estar em um de um número finito de estados.
- A máquina está em apenas um estado por vez, chamado de estado atual.
- Um estado armazena informações de entradas e saídas.
- Uma transição indica a condição necessária para uma mudança de estado.
- Uma ação é a atividade realizada num determinado estado.

Exemplo de máquina de estados para o controle de um LED





Uma máquinas de estado finitas podem modelar um grande número de problemas, entre os quais a automação de design eletrônico, projeto de protocolo de comunicação, análise e outras aplicações de engenharia.

Maquinas de estado podem ser representadas de várias forma, diagramas tabelas etc.

Existem basicamente dois tipos de máquinas de estado:

- **Máquina de Moore** onde a saída depende somente do estado atual;
- **Máquina de Mealy** onde a saída depende da entrada e do estado atual.

Maquinas de estado finitas podem ser implementadas de diversas formas, as mais comuns são em software, seja em computadores, em microcontroladores ou ainda em controladores lógicos programáveis (CLP) ou em hardware, em circuitos digitais discretos, em lógica reconfigurável ou em circuitos integrados dedicados.



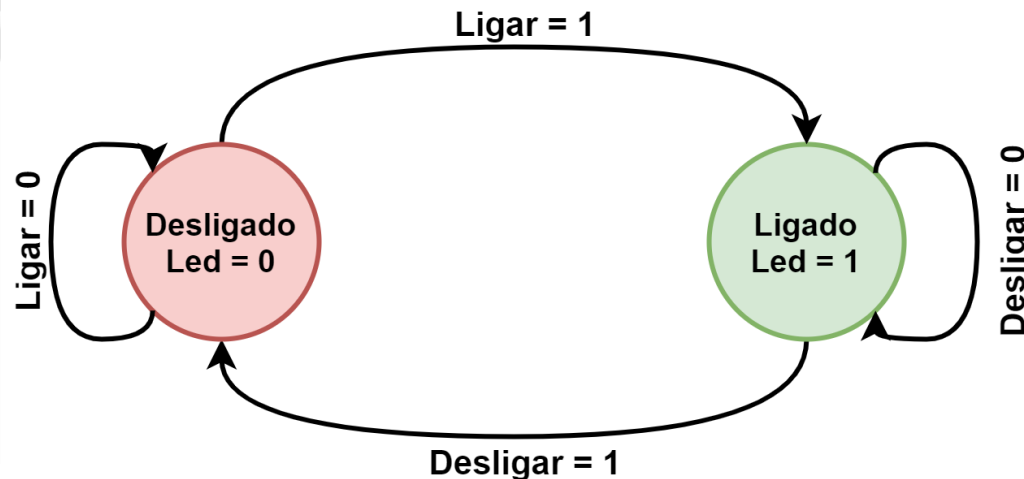
Partes de uma Máquina de Estado

De forma simplificada uma máquina de estados é composta por:

- Estados: possíveis estados de operação do sistema ex: Desligado, Ligado.
- Transições: passagem de um estado a outro ex de Desligado para Ligado.
- Entradas: entradas do sistema, que disparam as transições ex: Ligar, Desligar.
- Saídas: são as saídas da maquina de estados que interagem com o resto do sistema.

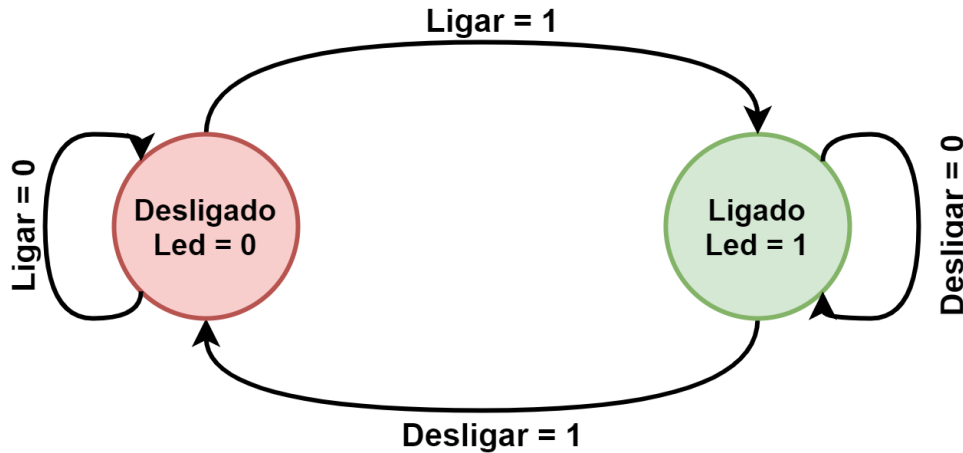
Cada estado monitora um conjunto de entradas para determinar se deve passar a outro estado ou permanecer onde está.

Cada estado fornece uma determinada saída ao sistema.





Um exemplo em linguagem C

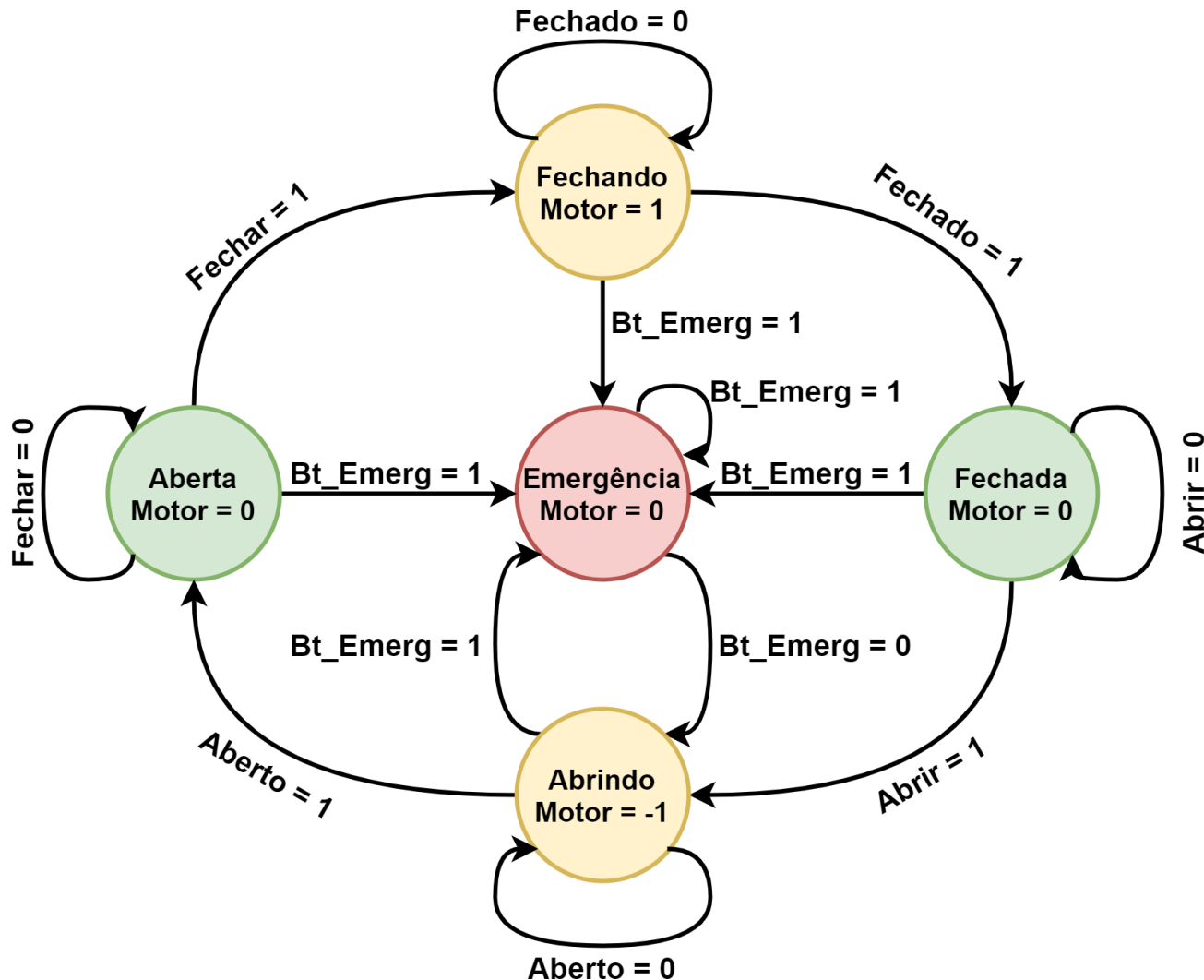


```
int main(void)
{
    inicializa();
    int EstadoAtual = StDesligado;
    int ProximoEstado = StDesligado;
    while (1)
    {
        if(EstadoAtual == StDesligado)
        {
            desligaLed();
            if(entradaLigar()==1)
            {
                ProximoEstado = StLigado;
            }
        }
        if(EstadoAtual == StLigado)
        {
            ligaLed();
            if(entradaDesligar()==1)
            {
                ProximoEstado = StDesligado;
            }
        }
        EstadoAtual = ProximoEstado;
    }
}
```



Um exemplo mais complexo

Porta Automática



Máquina de estados para controle de LED

```
#include <avr/io.h>

#define StDesligado 0
#define StLigado 1

int entradaLigar()
{
    if((PINC&(1<<PC1)) != 0)
        return 1;
    else
        return 0;
}

int entradaDesligar()
{
    if((PINC&(1<<PC0)) != 0)
        return 1;
    else
        return 0;
}

void ligaLed()
{
    PORTB|=(1<<PB0); // liga o LED em PB0
}

void desligaLed()
{
    PORTB&=~(1<<PB0); //desliga o LED em PB0
}

void inicializa()
{
    DDRB|=(1<<PB0); //define PB0 como saída
}

int main(void)
{
    inicializa();
    int EstadoAtual = StDesligado;
    int ProximoEstado = StDesligado;

    while (1)
    {
        if(EstadoAtual == StDesligado)
        {
            desligaLed();
            if(entradaLigar()==1)
            {
                ProximoEstado = StLigado;
            }
        }
        if(EstadoAtual == StLigado)
        {
            ligaLed();
            if(entradaDesligar()==1)
            {
                ProximoEstado = StDesligado;
            }
        }
        EstadoAtual = ProximoEstado;
    }
}
```

Máquina de estados para controle de porta automática

```
#include <avr/io.h>

#define StAberta 0
#define StAbrindo 1
#define StFechada 2
#define StFechando 3
#define StEmergencia 4

int entradaFechar()
{
    if((PINC&(1<<PC0)) != 0)
        return 1;
    else
        return 0;
}

int entradaAbrir()
{
    if((PINC&(1<<PC1)) != 0)
        return 1;
    else
        return 0;
}

int sensorFechado()
{
    if((PINC&(1<<PC2)) != 0)
        return 1;
    else
        return 0;
}

int sensorAberto()
{
    if((PINC&(1<<PC3)) != 0)
        return 1;
    else
        return 0;
}

int emergencia()
{
    if((PINC&(1<<PC4)) != 0)
        return 1;
    else
        return 0;
}

void motorAbre()
{
    PORTB|=(1<<PB0);
    PORTB&=~(1<<PB1);
}

void motorFecha()
{
    PORTB&=~(1<<PB0);
    PORTB|=(1<<PB1);
}

void motorDesliga()
{
    PORTB&=~(1<<PB0);
    PORTB&=~(1<<PB1);
}

void inicializa()
```



```

{
    DDRB|=(1<<PB0); //define PB0 como saída
    DDRB|=(1<<PB1); //define PB1 como saída
}

int main(void)
{
    inicializa();
    int EstadoAtual = StEmergencia;
    int ProximoEstado = StEmergencia;

    while (1)
    {
        if(EstadoAtual == StEmergencia) // estado Emergência
        {
            motorDesliga();
            if(emergencia()==0)
            {
                ProximoEstado = StAbrindo;
            }
        }
        if(EstadoAtual == StAberta) // estado Porta Aberta
        {
            motorDesliga();
            if(emergencia()==1)
            {
                ProximoEstado = StEmergencia;
            }
            else
            {
                if(entradaFechar()==1)
                {
                    ProximoEstado = StFechando;
                }
            }
        }
        if(EstadoAtual == StAbrindo) // estado Porta Abrindo
        {
            motorAbre();
            if(emergencia()==1)
            {
                ProximoEstado = StEmergencia;
            }
            else
            {
                if(sensorAberto()==1)
                {
                    ProximoEstado = StAberta;
                }
            }
        }
        if(EstadoAtual == StFechada) // estado Porta Fechada
        {
            motorDesliga();
            if(emergencia()==1)
            {
                ProximoEstado = StEmergencia;
            }
            else
            {
                if(entradaAbrir()==1)
                {
                    ProximoEstado = StAbrindo;
                }
            }
        }
        if(EstadoAtual == StFechando) // estado Porta fechando
        {
            motorFecha();
        }
    }
}

```

```
        if(emergencia()==1)
        {
            ProximoEstado = StEmergencia;
        }
        else
        {
            if(sensorFechado()==1)
            {
                ProximoEstado = StFechada;
            }
        }
    }
    EstadoAtual = ProximoEstado;
}
}
```