

Sistemas embarcados

Exemplos de projeto de sistemas embarcados com máquinas de estados finitos (FSM)

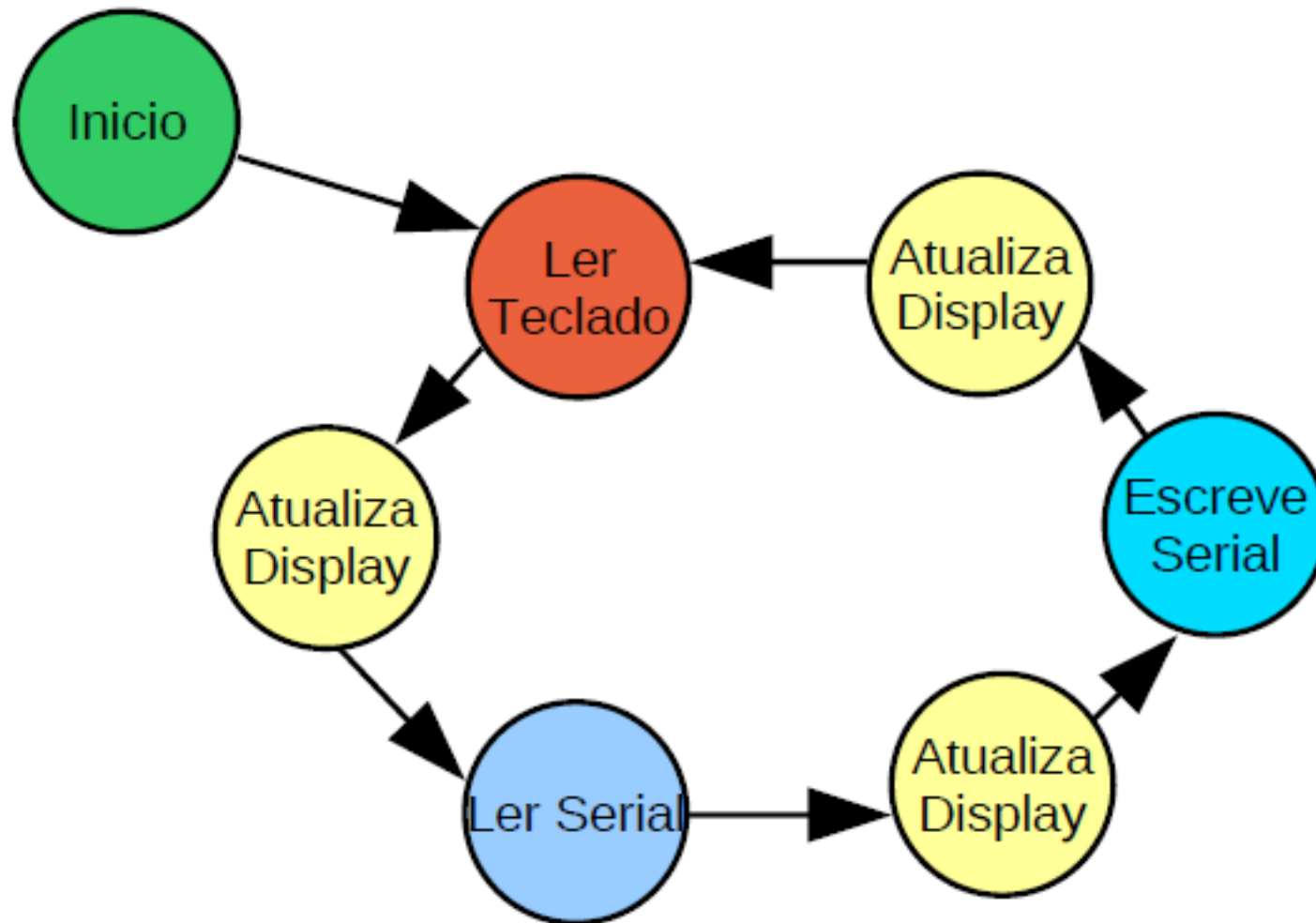
Projeto com máquinas de estados finitos (FSM)

Uma técnica mais organizada de se projetar o software de um sistema embarcado é a utilização do modelo de uma máquina de estados finitos (FSM).

Nesta técnica, o sistema é separado em estados em que ele pode estar e cada estado executa uma das tarefas apenas.

Projeto com máquinas de estados finitos (FSM)

Exemplo de máquina de estados



Exemplo

Definição: A entrada de um típico **sistema de automação de estacionamento** é composta pelos seguintes elementos:

- um emissor de tickets,
- uma cancela e
- um ou mais loops de detecção de veículos.

Exemplo extraído de <http://sergioprado.org/maquina-de-estados-em-c>.

Exemplo

Definição: O loop é um sensor de chapa metálica localizado no chão e utilizado para identificar o veículo. Existem dois loops:

- um em frente ao emissor de tickets (loop A)
- e outro abaixo da cancela (loop B)

Exemplo

As seguintes ações são realizadas na entrada de um veículo:

1. O usuário posiciona o veículo em frente ao emissor de tickets, **o loop A será acionado** e uma mensagem **“Pressione o botão”** é emitida.
2. O usuário **pressiona o botão** para emitir o ticket, o sistema **realiza a emissão e levanta a cancela**.
3. O usuário **passa pela cancela (loop B)** e entra no estacionamento. O sistema **aguarda e abaixa a cancela**.

Exemplo

Pelas ações listadas, podemos identificar basicamente **os estados** abaixo, na ordem em que são executados:

1. Aguardando veículo no loop A.
2. Aguardando usuário pressionar o botão para emitir ticket.
3. Aguardando veículo sair do loop A.
4. Aguardando veículo passar pelo loop B.

Exemplo

Pelas ações listadas, podemos identificar basicamente **os estados** abaixo, na ordem em que são executados:

1. Aguardando veículo no loop A.
2. Aguardando usuário pressionar o botão para emitir ticket.
3. Aguardando veículo sair do loop A.
4. Aguardando veículo passar pelo loop B.

Exemplo

Percebe-se que, para qualquer um dos estados acima, dependendo da entrada, a saída será diferente.

Por exemplo, se o sistema está no estado 2 e recebe como entrada o botão pressionado, ele vai para o estado 3.

Porém, se ao invés do botão pressionado, ele receber como entrada a saída de veículo do loop A (usuário deu ré e desistiu de estacionar), ele deve voltar para o estado 1.

Exemplo

Desenhe um diagrama de funcionamento do sistema.

Exemplo 2

Um problema que pode ser resolvido com máquina de estados é o **tratamento de protocolos de comunicação**. Neste exemplo, o protocolo de comunicação tem o seguinte formato:

`| STX | QTD_DADOS | DADOS | CHK | ETX |`

STX	(1 Byte)	-> Início da transmissão (0x02)
QTD_DADOS	(1 Byte)	-> Quantidade de dados
DADOS	(N Bytes)	-> Dados
CHK	(1 Byte)	-> Checksum da transmissão
ETX	(1 Byte)	-> Fim da transmissão (0x03)

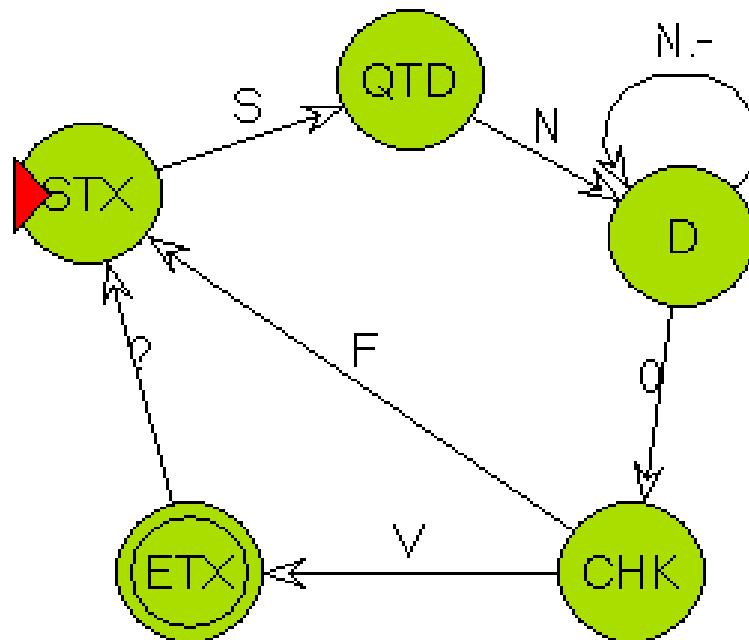
Exemplo extraído de <http://sergioprado.org/maquina-de-estados-em-c>.

Exemplo

Desenhe um diagrama de funcionamento do sistema e implemente o tratamento do protocolo usando máquina de estados.

Exemplo

Desenhe um diagrama de funcionamento do sistema.



Exemplo

Solução com switch-case.

```
/* possíveis estados da máquina de estados de comunicacao */  
typedef enum {  
    ST_STX = 0, ST_QTD, ST_DATA, ST_CHK, ST_ETX  
} States;
```

```
/* Implementação da máquina de estados */  
void handleRx(unsigned char *data, int qtd)  
{  
    static States state = ST_STX;  
    static unsigned char buffer[MAX_BUFFER];  
    static int indBuffer = 0, qtdBuffer = 0;  
    static unsigned char chkBuffer = 0;  
    int i;  
    .  
    .  
    .
```

```

for (i = 0; i < qtd; i++) {
    switch (state) {
        case ST_STX:
            if (data[i] == STX) {
                indBuffer = qtdBuffer = chkBuffer = 0;           // Início dos dados
                state = ST_QTD;
            }
            break;

        case ST_QTD:
            qtdBuffer = data[i];                                   // Quantidade de dados
            state = ST_DATA;
            break;

        case ST_DATA:
            buffer[indBuffer++] = data[i];
            chkBuffer ^= data[i];
            if (--qtdBuffer == 0) {                                // Calcula soma de verificação
                state = ST_CHK;                                    (checksum)
            }
            break;

        case ST_CHK:
            if (data[i] == chkBuffer) {                            // Testa checksum
                state = ST_ETX;
            }
            else {
                state = ST_STX;
            }
            break;

        case ST_ETX:
            if (data[i] == ETX) {
                handlePackage(buffer, indBuffer);                // Imprime dados recebidos
            }
            state = ST_STX;
            break;
    }
}

```

Exemplo

Solução com ponteiros de função.

```
/* possíveis estados da maquina de estados de comunicacao */
typedef enum {
    ST_STX = 0, ST_QTD, ST_DATA, ST_CHK, ST_ETX
} states;

typedef void (*Action)(unsigned char data);    // Ponteiro de função

struct StateMachine {
    States state;
    unsigned char buffer[MAX_BUFFER];
    unsigned char chkBuffer;                  // Estrutura da FSM
    int indBuffer;
    int qtdBuffer;
    Action action[5];
} sm;
```



```

void stSTX(unsigned char data)
{
    if (data == STX) {
        sm.indBuffer = sm.qtdBuffer = 0;
        sm.chkBuffer = 0;
        sm.state = ST_QTD;
    }
}

// Início dos dados

void stQtd(unsigned char data)
{
    sm.qtdBuffer = data;
    sm.state = ST_DATA;
}

// Quantidade de dados

void stData(unsigned char data)
{
    sm.buffer[sm.indBuffer++] = data;
    sm.chkBuffer ^= data;
    if (--sm.qtdBuffer == 0) {
        sm.state = ST_CHK;
    }
}

// Calcula soma de verificação (checksum)

void stChk(unsigned char data)
{
    if (data == sm.chkBuffer) sm.state = ST_ETX;
    else sm.state = ST_STX;
}

// Testa checksum

void stETX(unsigned char data)
{
    if (data == ETX) {
        handlePackage(sm.buffer, sm.indBuffer);
    }
    sm.state = ST_STX;
}

// Imprime dados recebidos

```

```

void handlerX(unsigned char *data, int qtd)
{
    int i;

    for (i = 0; i < qtd; i++) {
        sm.action[sm.state](data[i]);
    }
}

```

// Execução da FSM

```

void initSM()
{
    sm.state = ST_STX;
    sm.buffer[0] = 0;
    sm.chkBuffer = 0;
    sm.indBuffer = 0;
    sm.qtdBuffer = 0;
    sm.action[ST_STX] = stSTX;
    sm.action[ST_QTD] = stQtd;
    sm.action[ST_DATA] = stData;
    sm.action[ST_CHK] = stChk;
    sm.action[ST_ETX] = stETX;
}

```

// Inicialização da FSM

```

/* main para simular o uso da maquina de estados */
int main()
{
    unsigned char data1[] = { STX, 5, 11, 22, 33, 44 }; // exemplos de entrada
    unsigned char data2[] = { 55, 39, ETX };

    initSM();

    handlerX(data1, sizeof(data1));
    handlerX(data2, sizeof(data2));

    return 0;
}

```