



UFAM

Laboratório de Eletrônica Digital

Cafeteira

Manaus/AM

08/11/2018



UFAM

Laboratório de Eletrônica Digital

Alunos: Jean Cleison Braga Guimarães - 21601227

Maurício Assayag Aguiar- 21554919

Professor: Thiago Brito

Manaus-AM
2018

Resumo

Utilizamos uma máquina de estados finitos como base para 3 módulos, denominados "expresso", "medio" e "cheio", onde dentro de cada módulo é verificado o cumprimento de cada etapa necessária para a produção do café especificado.

Cumpridas todas as etapas necessárias, o café é produzido.

Como módulo principal, denominado "Cafeteira", utilizamos um sistema de escolha de decisões básico, baseado nas entradas do usuário ao pressionar o botão do café desejado na máquina.

Sensores utilizados

Sensor de temperatura: Para medir se a água está na temperatura ideal para a produção do café, utilizamos um termopar tipo K.

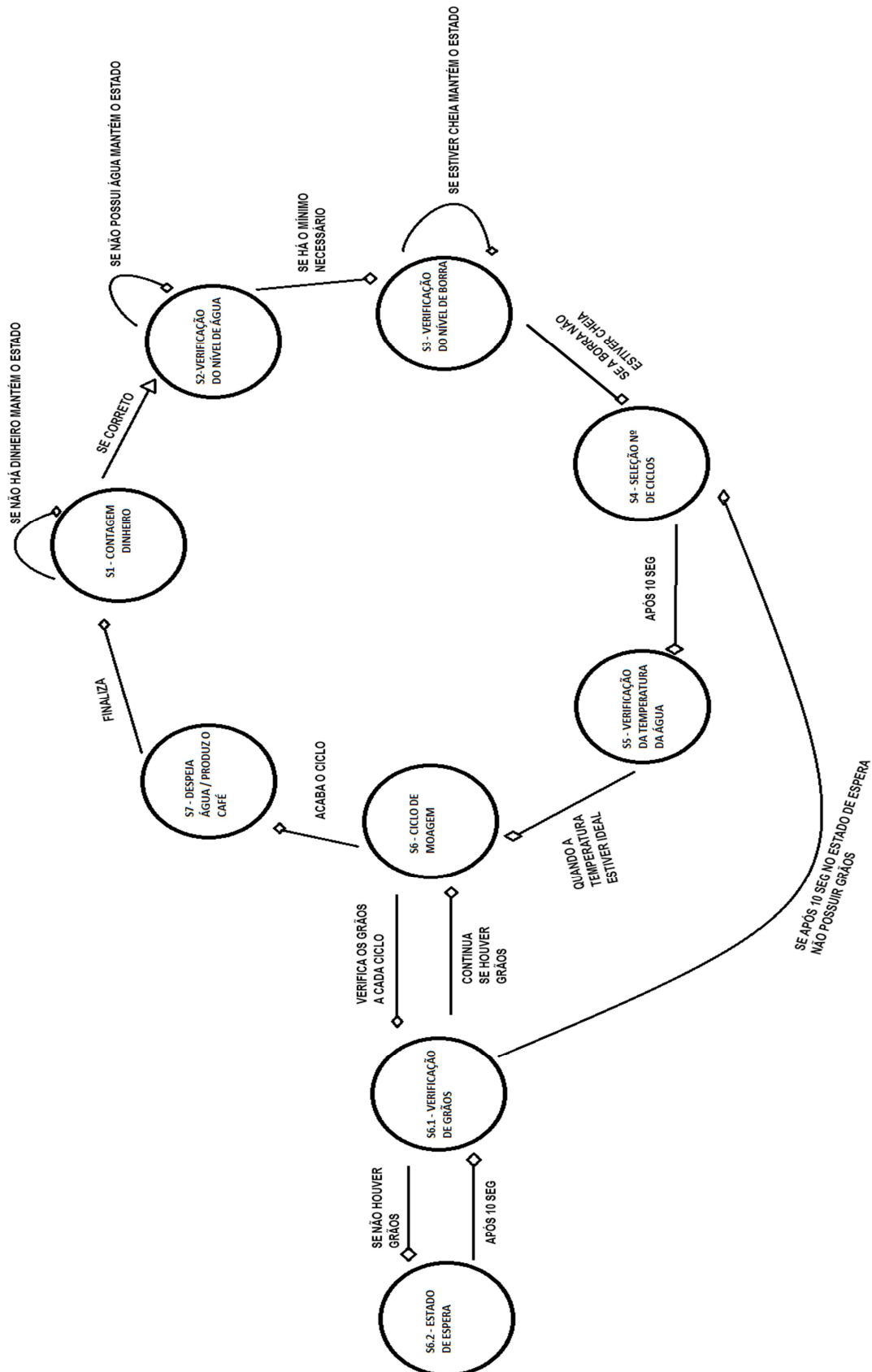
A escolha foi baseada em sua alta versatilidade, baixo custo, alta precisão para o que desejamos e boa durabilidade devido à sua resistência à oxidação em altas temperaturas e à corrosão em baixas temperaturas.

Sensor de borra : Para medir se o compartimento de borra ainda tinha ou não espaço, utilizamos uma célula de carga com Strain Gauge. Determinamos o peso médio de borra necessário para encher o compartimento e quando a célula medisse um peso maior ou igual ao determinado, o sistema entende que o compartimento está cheio, até que seja esvaziado.

Sensor de grãos: De forma similar ao sensor de borra, utilizamos uma célula de carga com Strain Gauge para medir o peso médio de grãos no compartimento de grãos. Determinamos então o peso médio necessário para se fazer um bom café de cada tipo especificado. Quando a célula medir um peso menor que o determinado, o sistema recebe o sinal como uma mensagem de que não há grãos suficientes para produzir o café.

Sensor de nível de água: Para medir o se o nível da água é suficiente para pelo menos um café, foi utilizado um sensor de nível por bóia e definida a tensão proporcional emitida pela posição da bóia necessária para produzir pelo menos um copo de café de cada tipo.

Desenho da máquina de estados



Estrutura do código

Entradas e parâmetros:

```
1 //MÓDULO EXPRESSO
2 module expresso(clk, money, sen_borra, sen_grao, sen_temp, nvl_agua, sel_ciclos, rst, coffee, balance, clock1hz, busy);
3 input clk, money, sen_borra, sen_grao, sen_temp, nvl_agua, rst, clock1hz;
4 input [2:0] sel_ciclos;
5 output reg coffee, balance, busy;
6
7 //parametros utilizando codificação gray
8
9 parameter [3:0] S1contagem_money=4'b0001; //contagem do dinheiro
10 parameter [3:0] S2verifica_agua=4'b0011; //verificação do nível de água
11 parameter [3:0] S3verifica_borra=4'b0010; //verifica do nível de borra
12 parameter [3:0] S4sel_ciclos=4'b0110; // espera de 10 seg para a seleção do numero de ciclos de moagem
13 parameter [3:0] S5verifica_temp=4'b0111; //verifica a temperatura da agua
14 parameter [3:0] S6ciclo_moagem=4'b0101; //ciclo de moagem
15 parameter [3:0] S7verifica_graos=4'b0100; //verifica a qtd de graos a cada ciclo de moagem
16 parameter [3:0] S8espera=4'b1100; //espera de 10seg caso não haja grãos suficientes
17 parameter [3:0] S9cafe=4'b1101; //despeja a água para a produção de cafe
18
```

Variáveis e contador:

```
reg [3:0] next_state;
reg [3:0] current_state;
reg [3:0] contador=0;
reg [2:0] n_ciclos;
reg enable=1'b0;
reg aux;

//contador 10 segundos
always@(posedge clock1hz)
begin
    enable=aux;
    if(enable)
        contador=contador+1;
    if(contador==10)begin
        enable=0;
        contador=0;
    end
end
```

Estrutura do estado atual:

```
always @(posedge clk)
begin
    if(rst) begin
        current_state<=S1contagem_money;
        busy= 0;
    end
    else begin
        current_state<=next_state;
        busy= 1;
    end
end
```

Estrutura do próximo estado:

```
always @(current_state or sen_borra or sen_grao or sen_temp or nvl_agua or sel_ciclos)begin
  case(current_state)
    S1contagem_money: begin
      if(money== 50)
        next_state<=S2verifica_agua; //se o dinheiro estiver correto passa para o próximo estado
      else if(money< 50)
        next_state<=S1contagem_money; //se o dinheiro for menos que o esperado mantém no estado até que haja o valor correto
      else begin
        next_state<=S2verifica_agua; //se o dinheiro for maior que o esperado passa para o próximo estado e devolve o troco
        balance = money - 50;
      end
    end
    S2verifica_agua: begin
      if(nvl_agua)
        next_state<=S3verifica_borra; //se o nível de água for o mínimo necessário passa para o próximo estado
      else
        next_state<=S2verifica_agua; //se não mantém o estado e aguarda a adição de água
      end
    end
    S3verifica_borra: begin
      if(!sen_borra)
        next_state<=S4sel_ciclos; //se a borra não estiver cheia passa para o próximo estado
      else
        next_state<=S3verifica_borra; //se estiver mantém o estado
      end
    end

    S4sel_ciclos: begin
      aux=1'b1;
      case(sel_ciclos)
        3'b000: n_ciclos<=1;
        3'b001: n_ciclos<=2;
        3'b010: n_ciclos<=3;
        3'b011: n_ciclos<=4;
        3'b100: n_ciclos<=5;
        3'b101: n_ciclos<=6;
        3'b110: n_ciclos<=7;
        3'b111: n_ciclos<=8;
      endcase
      if(contador==10)
        next_state<=S5verifica_temp; //Espera 10 segundos para a escolha da quantidade de ciclos
        aux=1'b0;
      end
    end
    S5verifica_temp: begin
      if(sen_temp)
        next_state<=S6ciclo_moagem; //Verifica se a temperatura esta ideal e passa para o proximo estado
      else
        next_state<=S5verifica_temp; //Continua esperando a temperatura ideal
      end
    end
  endcase
end
```

```

S6ciclo_moagem: begin
    case(n_ciclos)
        0: begin
            end
        1: begin
            next_state <= S7verifica_graos;
            end
        2: begin
            n_ciclos <= (n_ciclos-1);
            next_state <= S7verifica_graos;
            end
        3: begin
            n_ciclos <= (n_ciclos-1);
            next_state <= S7verifica_graos;
            end
        4: begin
            n_ciclos <= (n_ciclos-1);
            next_state <= S7verifica_graos;
            end
        5: begin
            n_ciclos <= (n_ciclos-1);
            next_state <= S7verifica_graos;
            end
        6: begin
            n_ciclos <= (n_ciclos-1);
            next_state <= S7verifica_graos;
            end
        7: begin
            n_ciclos <= (n_ciclos-1);
            next_state <= S7verifica_graos;
            end
        8: begin
            n_ciclos <= (n_ciclos-1);
            next_state <= S7verifica_graos;
            end
        default: next_state<=S7verifica_graos;
    endcase
end

S7verifica_graos: begin
    if(sen_grao)
        next_state<=S6ciclo_moagem;
    else
        next_state<=S8espera;
    end
end

S8espera: begin
    aux=1'b1;
    if(contador==10)begin
        aux=1'b0;
        if(sen_grao)
            next_state<=S6ciclo_moagem;
        else
            next_state<=S4sel_ciclos;
        end
    end
endcase
end

```


Estrutura de saída:

```
| always@ (current_state)begin
|   case(current_state)
|     S1contagem_money: coffee=0;
|     S2verifica_agua: coffee=0;
|     S3verifica_borra: coffee=0;
|     S4sel_ciclos: coffee=0;
|     S5verifica_temp: coffee=0;
|     S6ciclo_moagem: coffee=0;
|     S7verifica_graos: coffee= 0;
|     S8espera: coffee=0;
|     S9cafe: coffee=1;
|   endcase
|   end
endmodule
```