

INSTITUTO FEDERAL  
SANTA CATARINA  
Campus São José

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E  
TECNOLOGIA  
DE SANTA CATARINA - CAMPUS SÃO JOSÉ  
ENGENHARIA DE TELECOMUNICAÇÕES  
DISCIPLINA: SISTEMAS EMBARCADOS**

# Controle de Temperatura

Renan Rodolfo da Silva

São José, Abril de 2020

# 1.Introdução

Esse relatório tem objetivo demonstrar o comportamento do projeto desenvolvido para simulação de um controlador de temperatura que foi usado para a produção de uma cerveja artesanal. O projeto foi simulado no Software Proteus, onde foi possível utilizar um arduino para controlar um sensor de temperatura, resistência elétrica, Buzzer de um bomba de água.

## 2.Projeto

A base para o desenvolvido do código de controle foi o fluxograma abaixo

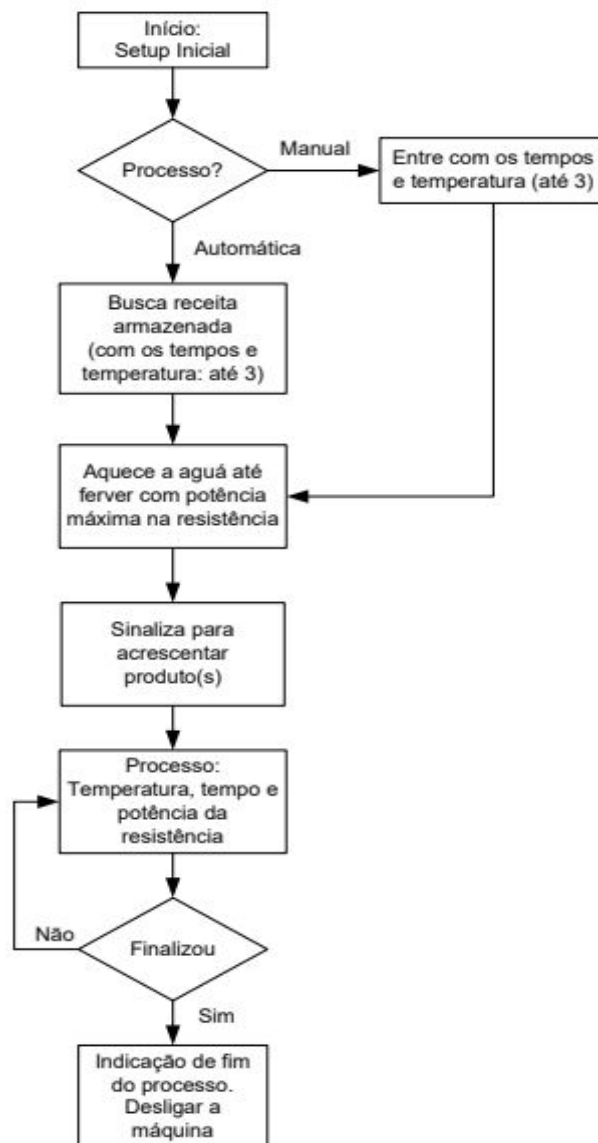
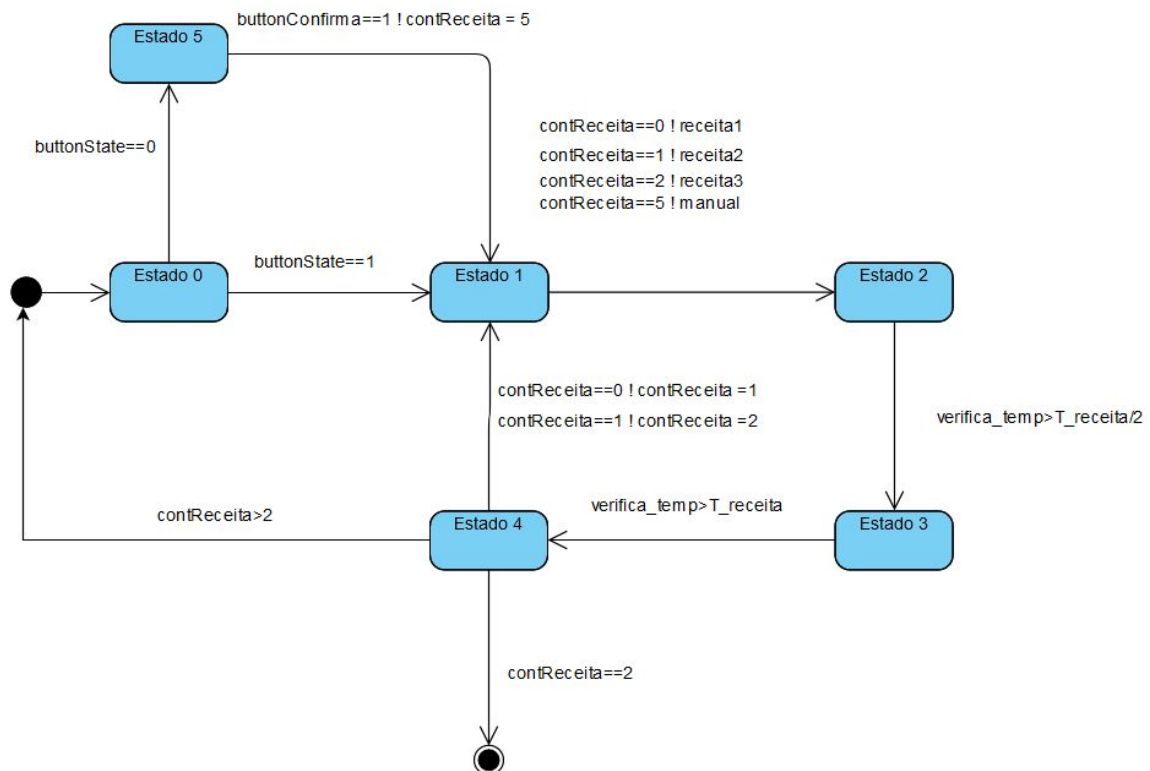


Figura 1: Fluxograma do controle

Com base nessas informações, realizei o desenvolvimento do código com uso de uma máquina virtual, que pode ser observada na figura 2.



### 3. Comportamento

O funcionamento está descrito na máquina de estado na figura 2. Como pode ser observado, a máquina possui 6 estados: estado 1, estado 2, estado 3, estado 4, estado 5 e estado 6.

O estado 0 é responsável por fazer a escolha do modo de operação, quando aparecer a mensagem no display “Selecione Modo” e o botão Selecionar for pressionado, ele entrará em modo Automático, caso contrário ficará como Manual.

Estado 1 serve para selecionar qual das 3 receitas do modo automático será processada. Por padrão, quando o contador de receita estiver em 0, a receita 1 será processada. Quando o contador de receita estiver em 1, a receita 2 será processada e por fim, quando o contador de receita estiver em 2, a receita 3 será processada. Quando estiver no modo manual, o contador de receita estará com o valor 5, nesse caso, a receita em progresso recebeu o valor de temperatura do usuário no estado 5.

Estado 2 é um laço que liga a resistência para iniciar o processo de aquecimento da água até chegar na metade do valor da temperatura da receita. Nesse estado ele deveria aquecer a água com relação ao tempo, mas não consegui desenvolver. A temperatura tem que ser ajustada manualmente com o uso do componente LM35.

Estado 3 verifica se o usuário deseja adicionar algum produto na receita da cerveja. O usuário irá ver um alarme visual(led) e ouvir um alarme sonoro(Buzzer). Caso queira

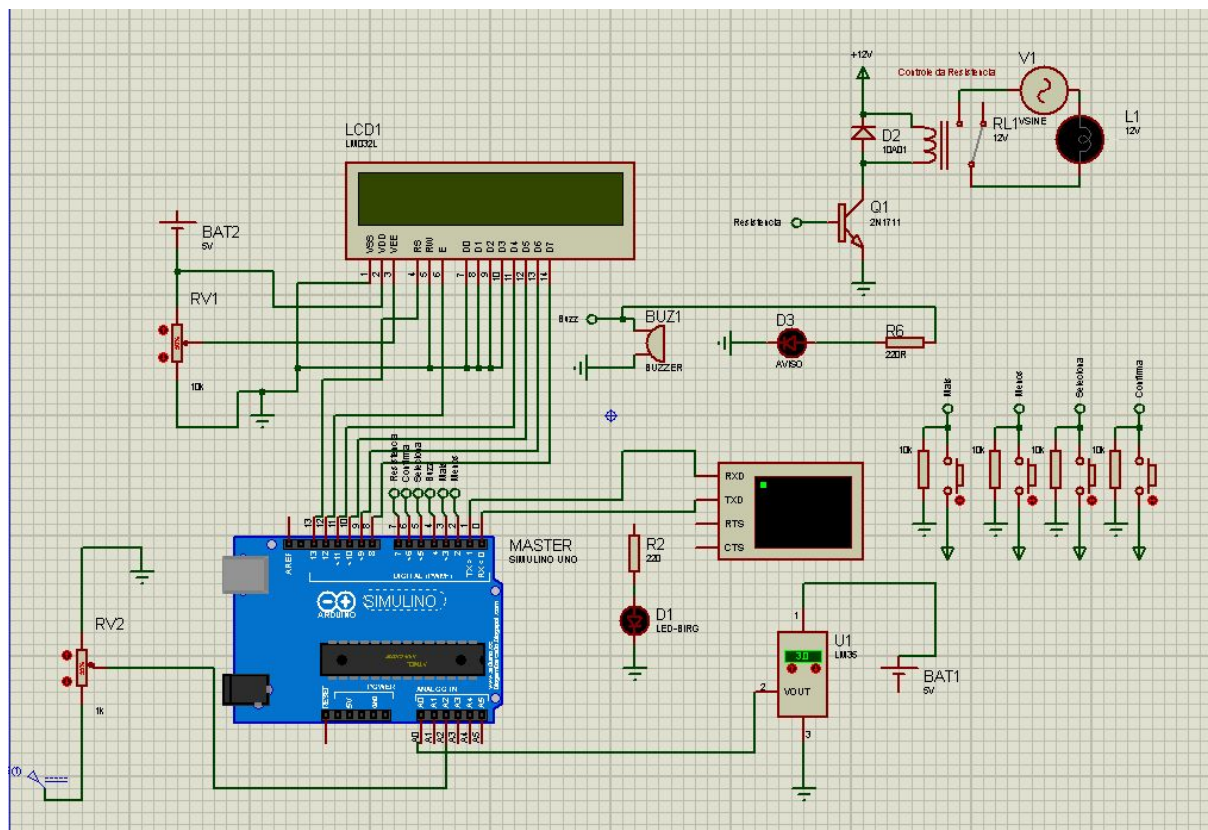
adicionar, precisa pressionar o botão confirmar. Nesse tempo em que o usuário decide se vai adicionar ou não um produto, a resistência fica desligada, para não prejudicar a produção. Adicionando ou não um produto, o estado vai aguardar a temperatura chegar até a temperatura da receita.

Estado 4 é o estado que finaliza a receita, aqui a receita já atingiu a sua temperatura e teoricamente o tempo de produção. Assim como no momento de adicionar um produto, o usuário irá ver um alarme visual e ouvir um alarme sonoro. Após finalizar, caso esteja no modo automático, irá selecionar a próxima receita no estado 1.

Estado 5 é responsável por receber a temperatura do usuário através dos botões Mais e Menos. Assim que escolher a temperatura da receita, deve pressionar o botão confirmar para dar início na produção.

## 4. Esquema elétrico

Para essa simulação foi utilizado o Arduino UNO. Os componentes utilizados para o desenvolvimento foram: LM35 para o sensor de temperatura, Buzzer para aviso sonoro, Led para aviso luminoso, LM032L como display, 4 botões para interface com o usuário, um relay de 12V para simular a resistência que varia de 0% à 100%(no caso ou é 0% ou é 100%).



## 5. Código

Tive algumas dificuldades para conseguir gerar módulos pra deixar o código mais limpo, mas com não consegui, tive que fazer tudo através da IDE do arduino.

```
/*
  Projeto - Termômetro LCD com sensor LM35 e Arduino
  By - Zedequias Fonseca - 08/2014
  http://vfeletronica.blogspot.com.br
*/
#include <stdint.h>
#include <LiquidCrystal.h> // Biblioteca para o LCD
#define TEMPORECEITA A
#define TEMPORECEITA B
#define TEMPORECEITA C
#define TEMPERATURAREICEITA A
#define TEMPERATURAREICEITA B
#define TEMPERATURAREICEITA C

// Variáveis

// Variáveis
int tempPin = A0;           // Declaração do pino de entrada analógica 0
                             (A0) no Arduino
int potPin = A2;           // Declaração do pino de entrada analógica 2
                             (A2) no Arduino
int resistenciaPin = 7;     // Controle da resistência
int sensorbomba = 9;       // sensor do tanque
float mAmostra;            // matriz para armazenar oito amostras para o
                             cálculo da temperatura média
float mLeitura = 0.00;     // variavel para armazenar leitura da
                             temperatura
int pLeitura = 0;          //
volatile int estado = 0;   // Variável que controla os estados
int contReceita = 0;       // Auxiliar para buscar as receitas.
int buzzer = 4;            // Pino onde está o Buzzer
```

```

double tempAtual;           // armazena temperatura atual do sensor
int botomMais = 3;          // Botão para incrementar temperatura
int botomMenos = 2;         // Botão para decrementar a temperatura
int botomConfirma = 6;      // Botão que confirma a escolha
int botomSelecionar = 5;    // Botão que seleciona o modo de operação [0 é
// manual e 1 é automático]
int buttonState = 0;         // Ler as portas seriais
int leituraup = 0;           // Verifica se o botão incrementar está
// pressionado
int leituradown = 0;         // Verifica se o botão decrementar está
// pressionado
int contador = 0;           // Aumenta a temperatura no modo manual

// Struct padrão para armazenar tempos e temperaturas
struct receitas{
    int t_receita; // tempo da receita
    int T_receita; // Temperatura da receita
};

// Inicialização do display e pinos correspondentes
LiquidCrystal lcd(13, 12, 11, 10, 9, 8);

// ligar a bomba
void bomba_on(){
    digitalWrite(potPin, HIGH);
}

// desligar a bomba
void bomba_off(){
    digitalWrite(potPin, LOW);
}

// verificar nível da água
bool check_water(){
    return digitalRead(sensorbomba) == HIGH;
}

// Verifica a temperatura atual do sensor
double verifica_temp(){
    mAmostra = (5 * analogRead(tempPin) * 100.0) / 1024.0;
    mLeitura = analogRead(tempPin); // ler porta do sensor
    delay(500);
}

```

```

        return mAmostra;                // retorna valor lido na porta
analogica
    }

// Seta novas receitas para produzir
receitas receitasPadrao(int t, int T){
    receitas r;
    r.t_receita = t;
    r.T_receita = T;
    return r;
}

// Setando receitas para modo automático
receitas receita1 = receitasPadrao(20,20);
receitas receita2 = receitasPadrao(30,50);
receitas receita3 = receitasPadrao(40,110);
receitas receitaEmProcesso;

void setup()
{
    //Definindo a função dos pinos
    pinMode(resistenciaPin, OUTPUT);
    pinMode(botomMais, INPUT);
    pinMode(botomMenos, INPUT);
    pinMode(botomConfirma, INPUT);
    pinMode(botomSelecionar, INPUT);

    Serial.begin(9600);                // inicia comunicação serial
    lcd.begin(16, 3);                  // inicia e configura o número de
linhas e colunas do LCD
    lcd.setCursor(1, 0);                // seta posição do cursor
(coluna, linha) do LCD
    lcd.print("Controlador de cerveja"); // imprime texto no LCD
    delay(3000);                        // aguarda 3s
    bomba_on();                          // Ativa a bomba
}

```

```

void loop()
{

    switch (estado)
    {
        // Define o modo: Automático ou Manual
        case 0 :

            lcd.clear();
            lcd.setCursor(1, 0);           // seta
posição do cursor LCD
            lcd.print("Selecione Modo");
            delay(2000);
            buttonState = digitalRead(botomSelecionar);           // Recebe
valor do botão selecionar

            if(buttonState==0){           // Se deixa em
0, vai para o estado 5 que processa o modo Manual
                lcd.clear();
                lcd.setCursor(1, 0);
                lcd.print("Manual");
                delay(1500);
                estado = 5;
                break;
            }else{
                lcd.clear();
                lcd.setCursor(1, 0);           // seta
posição do cursor LCD
                lcd.print("Automatico");           // imprime no
LCD[
                estado = 1;           // Para o
estado 1 e processa o modo automático
                delay(1500);
                break;
            }
            // Busca a receita no automático
        case 1:
            analogWrite(buzzer, 0);           // Deixa
buzzer desligado

```



```

        digitalWrite(resistenciaPin, LOW); // Deixa
resistência desativada

        if(contReceita==0){ //Seleciona a
receita 1
            receitaEmProcesso = receita1; // Define a
receita que vai ser processada
            lcd.clear();
            lcd.setCursor(1, 0); // seta
posição do cursor LCD
            lcd.print("Receita1 selected"); // imprime
amostra da temperatura atual no LCD
            delay(2000);
            Serial.println("Tempo da receita");
            Serial.println(receitaEmProcesso.T_receita); // Apenas
imprimindo na serial
            estado = 2;
            delay(1000);
            break;
        }else if(contReceita==1){ //Seleciona a
receita 2
            receitaEmProcesso = receita2;
            lcd.clear();
            lcd.setCursor(1, 0); // seta
posição do cursor LCD
            lcd.print("Receita2 selected"); // imprime
amostra da temperatura atual no LCD
            delay(2000);
            Serial.println("Tempo da receita");
            Serial.println(receitaEmProcesso.T_receita);
            estado = 2; // Passa para
o estado 2
            break;
        }else if(contReceita==2){ //Seleciona a
receita 3
            receitaEmProcesso = receita3;
            lcd.clear();
            lcd.setCursor(1, 0); // seta
posição do cursor LCD

```

```

        lcd.print("Receita3 selected"); // imprime
amostra da temperatura atual no LCD
        Serial.println("Tempo da receita");
        Serial.println(receitaEmProcesso.T_receita);
        delay(2000);
        estado = 2;
        break;
    }else if(contReceita ==5){ // Caso onde a
receita foi definida no manual
        lcd.clear();
        lcd.setCursor(1, 0); // seta
posição do cursor LCD
        lcd.print("Receita Manual"); // imprime
amostra da temperatura atual no LCD
        Serial.println(receitaEmProcesso.T_receita);
        estado = 2;
        break;
    }else{
        break;
    }

    //Aquece a agua até chegar na metade da temperatura da receita
    case 2:
        analogWrite(buzzer, 0);
        digitalWrite(resistenciaPin, HIGH); // Liga a
resistência para começar a esquentar a água.
        lcd.clear();
        lcd.setCursor(1, 0); // seta
posição do cursor LCD
        lcd.print("Aquecendo a Agua"); // imprime
amostra da temperatura atual no LCD
        while(verifica_temp() < (receitaEmProcesso.T_receita/2)){ // Enquanto
não chegar na metade da temperatura, aguarda.
            delay(500);
        }

        estado = 3;
        break;

    //Estado responsável por verificar se quer adicionar algum ingrediente e
aquece até a temperatura final

```

```

    case 3:
        lcd.clear();
        lcd.setCursor(1, 0); // seta
posição do cursor LCD
        digitalWrite(resistenciaPin, LOW);
        lcd.print("Adicionar produto?"); // imprime
amostra da temperatura atual no LCD
        analogWrite(buzzer, 500); // Emite sinal
sonoro pra sinalizar se quer adicionar um produto
        delay(2000);
        buttonState = digitalRead(botomConfirma); // Aguarda
confirmação, caso esteja em 1, vai adicionar um produto.
        analogWrite(buzzer, 0); // Desliga o
buzzer
        if(buttonState==0){
            lcd.clear();
            lcd.setCursor(1, 0); // seta
posição do cursor LCD
            lcd.print("Nao adicionou"); // imprime
amostra da temperatura atual no LCD
            delay(1500);
        }else{
            lcd.clear();
            lcd.setCursor(1, 0); // seta
posição do cursor LCD
            lcd.print("Adicionado"); // imprime
amostra da temperatura atual no LCD[
            delay(1500);
        }
        digitalWrite(resistenciaPin, HIGH);
        lcd.clear();
        lcd.setCursor(1, 0); // seta
posição do cursor LCD
        lcd.println("Aquecendo a Agua"); // imprime
amostra da temperatura atual no LCD

        while(verifica_temp() < (receitaEmProcesso.T_receita)){ // Agurda no
loop até a temperatura chegar no valor da receita
            delay(500);
        }

```

```

        estado = 4;
        break;

        // Estado responsável por sinalizar o fim da receita e mudar para
receita seguinte(no automático)
        case 4:
            lcd.clear();
            lcd.setCursor(1, 0); // seta
posição do cursor LCD
            lcd.print("Finalizou receita"); // imprime
amostra da temperatura atual no LCD
            digitalWrite(resistenciaPin, LOW);
            analogWrite(buzzer, 700); // Liga o
Buzzer
            delay(2000);
            Serial.println(contReceita);
            if(contReceita==0){
                contReceita = 1;
                estado = 1;
                analogWrite(buzzer, 0); // Desliga o
Buzzer
            }else if(contReceita==1){
                contReceita = 2;
                estado = 1;
                analogWrite(buzzer, 0); // Desliga o
Buzzer
            }else if(contReceita){
                lcd.clear();
                lcd.setCursor(1, 0); // seta
posição do cursor LCD
                analogWrite(buzzer, 0);
                lcd.print("Programa encerrado"); // imprime
amostra da temperatura atual no LCD
            }else{
                contReceita = 0;
                lcd.clear();
                lcd.setCursor(1, 0); // seta
posição do cursor LCD
                analogWrite(buzzer, 0);

```

```

        lcd.print("Programa encerrado");           // imprime
amostra da temperatura atual no LCD
    }
    break;

    // Estado responsável por definir as tempereturas das receitas no modo
Manual
    case 5:
        lcd.clear();
        lcd.setCursor(1, 0);                       // seta
posição do cursor LCD
        contador = 0;
        lcd.clear();
        lcd.setCursor(1,0);
        lcd.print("Selecione Temp");
        while(true){
            leituraup = digitalRead(botomMais);      // Le o botão
de Mais
            leituradown = digitalRead(botomMenos);   // Le o botão
de Menos

            if (leituraup == HIGH)                  // Caso botão
Mais seja pressionado, incrementa a temperatura em 5
            {
                delay(300);
                // Incrementa o valor do contador
                contador = contador + 5;             // Incrementa a
temperatura
                // Apaga o valor anterior
                lcd.clear();
                lcd.setCursor(1,0);
                lcd.print("Temperatura");
                // Mostra o valor do contador
                lcd.setCursor(13,0);
                lcd.print(contador);
            }
            if (leituradown == HIGH)                 // Caso botão
Menos seja pressionado, decrementa a temperatura em 5
            {
                delay(300);

```

```

        // Incrementa o valor do contador
        if(contador > 0){
            // Condição
            para não deixar a temperatura ser negativa
            contador = contador - 5;
            // Apaga o valor anterior
            lcd.setCursor(1,0);
            lcd.print("Temperatura");

            // Mostra o
            valor do contador
            lcd.setCursor(13,0);
            lcd.print(contador);

        }else{
            lcd.setCursor(11,0);
            lcd.print("Temperatura");

            // Mostra o
            valor do contador
            lcd.setCursor(11,0);
            lcd.print(contador);
        }
    }
    buttonState = digitalRead(botomConfirma);
    // Caso botão
    confirma ficar em 1, salva a temperatura escolhida.
    if(buttonState==1){
        receitaEmProcesso.T_receita = contador;
        // Salva o
        tempo na receita que sera processada
        delay(1500);
        contReceita = 5;
        // Auxiliar
        para saber qual receita fazer, 0,1,2 são para automático
        estado = 1;
        // Volta ao
        estado 1 para iniciar a produção
        Serial.println(contador);
        break;
    }
}
}
}
}

```

