

```

#include "DHT.h" //Biblioteca para o sensor de temperatura
#include <SoftwareSerial.h> //Para a comunicação BlueTooth
SoftwareSerial bluetooth(10, 11);
//rx = 10 do Arduino
//tx = 11 do Arduino

#define DHTPIN A1 // pino que estamos conectado -DHT
#define DHTTYPE DHT11 // DHT 11 biblioteca do sensor de temperatura
// #define SETPOINT 23
const int SETPOINT = 23; //Setpoint para o controle de temperatura

DHT dht(DHTPIN, DHTTYPE); //Função do biblioteca do DHT

//*****Pinos*****
//Saídas

const int led_fora = 2; //Luz externa
const int led_dentro = 5; //Pode ser dois
const int m_portao_abre = 12; // Vai no IN A da ponte
const int m_portao_fecha = 13; //Vai no IN B da ponte
const int ventilador = 4; //ventilador
const int buzzer = 6; //Segurança

//Entradas
const int habilita_blue = 1;
const int b_portao_abre = 7; //Portao
const int b_portao_para = 8; //Portao--Sempre do VCC 5V
const int b_portao_fecha = 9; //Portao
const int s_janela = 3; //Segurança
const int ldr_ext = 0; //Entrada analogica A0
//*****Variaveis do programa*****
//bluetooth
char c;
//sensor temperatura
float errotemp = 0;
int state_vent = 0;
float errohum = 0;
//Acionamento do portao
boolean abre = false;
boolean fecha = false;
boolean para = false;
//Sistema de segurança
boolean valor_s_janela = false;
//Acionamento dos LEDs-Imluminação

```

```

int val_ldr_ext = 0;
boolean habilita_ilum = false;

void setup()
{
  Serial.begin(9600);
  bluetooth.begin(9600);
  bluetooth.flush();
  pinMode(habilita_blue, INPUT);
  //Portao
  pinMode(m_portao_abre, OUTPUT);
  pinMode(m_portao_fecha, OUTPUT);
  pinMode (b_portao_abre, INPUT);
  pinMode (b_portao_fecha, INPUT);
  //Temperatura
  //Serial.println("DHTxx test!");
  pinMode(ventilador, OUTPUT);
  dht.begin();
  //segurança
  pinMode(buzzer, OUTPUT);
  pinMode(s_janela, INPUT);
  //iluminação
  pinMode(led_fora, OUTPUT);
  pinMode(led_dentro, OUTPUT);
}

void loop() {
  while (digitalRead(habilita_blue) == HIGH) {          //Se for habilitado entra o
modo Bluetooth
    mod_bluetooth();
    seguranca();
  }

  temp_umid();          //DHT
  portao();             //Abre e fecha o portao
  seguranca();          //Alarme com buzzer e Led piscante
  //iluminacao();       //Iluminação baseada na luz externa
}

//*****Função de cada sistema*****

//_____Sensor de Umidade_____
void temp_umid()
{
  // A leitura da temperatura e umidade pode levar 250ms!

```

```

// O atraso do sensor pode chegar a 2 segundos.
float h = dht.readHumidity(); //Lelo-Função da biblioteca
criada
float t = dht.readTemperature();
// testa se retorno é valido, caso contrário algo está errado.
if (isnan(t) || isnan(h))
{
    Serial.println("Failed to read from DHT"); //Pode interferir no
Bluetooth
}
else
{
    //Condição para acionamento do ventilador
    errotemp = t - SETPOINT;
    if (errotemp >= 1) { //Tolernacia de 1°C
        digitalWrite(ventilador, HIGH);
        //state_vent = 1;
    }
    else {
        digitalWrite(ventilador, LOW);
        //state_vent = 0;
    }
    Serial.print( "SETPOINT:  ");
    Serial.print (SETPOINT);
    Serial.print( "  Temperatura:  ");
    Serial.println(t);
}

} //Fim da função temp_umid

//_____Portão_____
void portao()
{
    //int count = 0;
    abre = digitalRead(b_portao_abre);
    para = digitalRead(b_portao_para);
    fecha = digitalRead(b_portao_fecha);

    //Abrir Portão
    if (abre == HIGH) {
        digitalWrite(m_portao_fecha, LOW); //Intertravamento
        digitalWrite(m_portao_abre, HIGH); //Abre com velocidade
intermediária
        delay(2000);
        digitalWrite(m_portao_abre, LOW); //parada

```

```

        digitalWrite(m_portao_fecha, LOW);                //parada
    }

    if (fecha == HIGH) {
        digitalWrite(m_portao_abre, LOW);                //Intertravamento
        digitalWrite(m_portao_fecha, HIGH);                //Abre com velocidade
intermediária
        delay(2000);
        digitalWrite(m_portao_abre, LOW);                //parada
        digitalWrite(m_portao_fecha, LOW);                //parada
    }

} //Fim da função do portão

//_____Sistema de segurança_____
//Quando o sensor de abertura da janela for ativado, sorár um alarme e um led piscará
void seguranca() {
    valor_s_janela = digitalRead(s_janela);
    if (valor_s_janela == LOW) {
        int i;
        for (i = 0; i < 3; i++) {
            tone(buzzer, 1000);
            digitalWrite(led_fora, HIGH);
            delay(500);
            noTone(buzzer);
            digitalWrite(led_fora, LOW);
            delay(500);
        }
    }
    else {
        digitalWrite(buzzer, LOW);
    }
} //Fim da função segurança

//_____iluminacao_____
//void iluminacao(){
// // acionamento por LDR
//
// val_ldr_ext = analogRead(ldr_ext);
// Serial.println(val_ldr_ext);
// delay(10);
// if ( val_ldr_ext > 500) {                //Quanto mais escuro maior resistencia e maior
tensão
//     digitalWrite(led_fora, HIGH);
// }
// else {

```

```

//    digitalWrite(led_fora, LOW);
//    }
//
//}

//_____Modobluetooth_____

void mod_bluetooth() {

    if (bluetooth.available() > 0) {

        c = bluetooth.read();

        if (c == 'A') {
            digitalWrite(m_portao_fecha, LOW);           //Intertravamento
            digitalWrite(m_portao_abre, HIGH);           //Abre com
velocidade intermediária
            delay(2000);
            digitalWrite(m_portao_abre, LOW);           //parada
            digitalWrite(m_portao_fecha, LOW);           //parada
        }
        else if (c == 'a') {
            digitalWrite(m_portao_abre, LOW);           //Intertravamento
            digitalWrite(m_portao_fecha, HIGH);          //Abre com velocidade
intermediária
            delay(2000);
            digitalWrite(m_portao_abre, LOW);           //parada
            digitalWrite(m_portao_fecha, LOW);           //parada
        }
        else if (c == 'B') {
            digitalWrite(ventilador, HIGH);
        }
        else if (c == 'b') {
            digitalWrite(ventilador, LOW);
        }
        else if (c == 'C') {
            digitalWrite(led_fora, HIGH);
        }
        else if (c == 'c') {
            digitalWrite(led_fora, LOW);
        }

    }

} //Fim função Bluetooth

```