

Sistema automatizado de controle de velocidade de ar e controle de temperatura

Automated air speed control and temperature control system

Gustavo de Oliveira Martins
gumartins2001@gmail.com
INSTITUTO FEDERAL DO PARANÁ, Londrina, Paraná, Brasil

Mateus Vinicius de Paiva
mateusviniciuspaiva@outlook.com
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ, Londrina, Paraná, Brasil

Leonardo Galice Chies
leonardogalices@gmail.com
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ, Londrina, Paraná, Brasil

Janksyn Bertozzi
janksynbertozzi@utfpr.edu.br
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ, Londrina, Paraná, Brasil

Leonardo Carmezini Marques
leonardo.carmezini@ifpr.edu.br
INSTITUTO FEDERAL DO PARANÁ, Londrina, Paraná, Brasil

Romualdo Rubens de Freitas
romualdo.freitas@ifpr.edu.br
INSTITUTO FEDERAL DO PARANÁ, Londrina, Paraná, Brasil

Jefferson Sussumu de Aguiar Hachiya
jefferson.hachiya@ifpr.edu.br
INSTITUTO FEDERAL DO PARANÁ, Londrina, Paraná, Brasil

Lyssa Setsuko Sakanaka
lyssa@utfpr.edu.br
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ, Londrina, Paraná, Brasil.

RESUMO

Vivemos em uma sociedade em que alimentos são industrializados ou recebem a adição de conservantes, que estão presentes em tudo o que se come. A conservação dos alimentos é muito importante para a humanidade desde a antiguidade, quando os alimentos eram deixados no fundo da caverna onde era mais frio, ou expostos ao sol para que secassem e durassem mais tempo, ou até mesmo, após a descoberta do fogo, assá-los. Ao observar os importantes benefícios da desidratação de alimentos para a história da humanidade, percebe-se que esse processo é vantajoso se comparado ao congelamento ou enlatamento, que fazem uso abundante de energia e aditivos, causando impactos ecológicos mais agressivos. A secagem é uma alternativa interessante, no entanto, o seu sucesso depende do controle do processo. Uma maneira de controle é realizando a automatização. Para automatizar esse processo a baixo custo, de forma a disponibilizar este sistema para pequenos produtores, elaborou-se um sistema automatizado por um microcontrolador, que tem como principal função controlar sensores e atuadores em um sistema que regula a temperatura, umidade relativa e a velocidade da corrente de ar. Com isto, espera-se baixar o custo do processo permitindo que produtores rurais tenham disponível um meio barato e eficiente de realizar o processo de secagem e, consequentemente, a conservação de alimentos.

PALAVRAS-CHAVE: Linguagem C. Arduino. Secador de alimentos.

ABSTRACT

We live in a society where foods are industrialized or given the addition of preservatives, which are present in everything we eat. Food conservation has been very important to mankind since antiquity, when food was left in the bottom of the cave where it was cooler, or exposed to the sun to dry out and last longer, or even, after the discovery of fire, bake them. In observing the important benefits of food dehydration for human history, it is perceived that this process is advantageous when compared to freezing or canning, which make abundant use of energy and additives, causing more aggressive ecological impacts. Drying is an interesting alternative, however its success depends on process control. One way of controlling is by performing process automation. To automate this process at low cost, in order to make this system available to small producers, an automated system was developed by a microcontroller, whose main function is to control sensors and actuators in a system that regulates temperature, relative humidity and speed of the air stream. With this, it is expected to lower the cost of the process by allowing rural producers to have an inexpensive and efficient means of carrying out the drying process and, consequently, food preservation.

KEYWORDS: C language. Arduino. Food dryer.

Recebido: 31 ago. 2018.

Aprovado: 04 out. 2018.

Direito autorial:

Este trabalho está licenciado sob os termos da Licença Creative Commons-Atribuição 4.0 Internacional.



INTRODUÇÃO

O controle de parâmetros como temperatura, fluxo de ar e umidade relativa do ar da secagem são fundamentais para realizar um processo que preserve os nutrientes e as características sensoriais dos alimentos (GAVA, 1984). O controle destes parâmetros pode ser realizado empregando-se os fundamentos da automatização de processos. Neto e Teruel (2008) afirmam que o controle desses parâmetros sem interferência humana melhora a qualidade do produto uniformizando a produção e eliminando possíveis perdas.

Durante a secagem, o fluxo de ar é um parâmetro que deve ser controlado, uma vez que a velocidade do ar controla a eliminação dos vapores de água e também serve de meio de transporte de calor ao alimento.

Neste aspecto, o objetivo do presente trabalho foi empregar o disposto na Lei de Faraday de 1845, para associar um motor que permita gerar uma corrente elétrica que possa ser lida por um microcontrolador de baixo custo, e que possibilite controle de temperatura, velocidade do ar e umidade relativa dentro de um secador didático construído por alunos de engenharia.

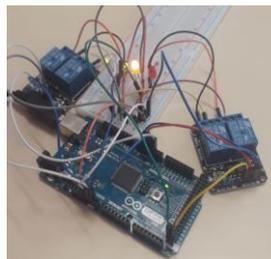
MATERIAIS E MÉTODOS

Para o estudo da automatização do processo de secagem foi utilizado como modelo, um secador didático construído por alunos dos cursos de engenharia. Um microcontrolador Arduino foi adotado, por possuir inúmeras bibliotecas para controlar sensores e atuadores, e por ser uma plataforma de prototipagem eletrônica de hardware livre e de baixo custo. O Arduino recebia as informações de temperatura e umidade em diversos pontos do sistema, e juntamente com a velocidade de fluxo de ar estabelecida por uma ventoinha, criou-se um parâmetro que define o número de resistências ligadas baseado na equação da calorimetria definido pelo balanço de energia.

RESULTADOS E DISCUSSOES

Para aquecer a corrente de ar são necessárias resistências, que serão controladas pelo Arduino por meio de relês, que se assemelham com interruptores apresentado na Figura 1. Para minimizar acidentes devido às altas temperaturas das resistências, realizou-se uma simulação das resistências por meio de LEDs (Light Emitting Diode), validando assim o código desenvolvido na linguagem C, conforme apresentado na Figura 2.

Figura 1. Fotografia do sistema de controle de resistências com LED.



Fonte: Autoria Própria (2018)

Figura 2. Algoritmo desenvolvido para o funcionamento do secador.

```

sketch_aug31a.s
/*
  GUSTAVO DE OLIVEIRA MARTINS - IFPR
  2018
  */
long valAne = 0; // velocidade do ar
#define RELE1 2 //constante para definição de pino
#define RELE2 3 //constante para definição de pino
#define RELE3 4 //constante para definição de pino
#define RELE4 5 //constante para definição de pino
int res = 0; //variavel de controle
void setup() {
  Serial.begin(9600); //inicia o monitor serial
  randomSeed(analogIn(0)); //apresenta random
  pinMode(RELE1, OUTPUT); //define o pino digital
  pinMode(RELE2, OUTPUT); //define o pino digital
  pinMode(RELE3, OUTPUT); //define o pino digital
  pinMode(RELE4, OUTPUT); //define o pino digital
}
int nRes(float valAne){ //resistencia f(velocidade)
  if(valAne > 125 && res == 2){
    res = 3;
    return res;
  }
  else if(res == 1){
    res = 0;
    return res;
  }
}
void loop() {
  valAne = random(50,100); //numeros aleatorios simulacao
  Serial.print(valAne/100.0);
  Serial.print(" = ");
  switch(nRes(valAne)){
    case 0:
      Serial.print(1);
      digitalWrite(RELE1, HIGH); //liga o pino digital
      digitalWrite(RELE2, LOW); //desliga o pino digital
      digitalWrite(RELE3, LOW); //desliga o pino digital
      digitalWrite(RELE4, LOW); //desliga o pino digital
      break;
    case 1:
      Serial.print(2);
      digitalWrite(RELE1, HIGH); //liga o pino digital
      digitalWrite(RELE2, HIGH); //liga o pino digital
      digitalWrite(RELE3, LOW); //desliga o pino digital
      digitalWrite(RELE4, LOW); //desliga o pino digital
      break;
    case 2:
      Serial.print(3);
      digitalWrite(RELE1, HIGH); //liga o pino digital
      digitalWrite(RELE2, HIGH); //liga o pino digital
      digitalWrite(RELE3, HIGH); //liga o pino digital
      digitalWrite(RELE4, LOW); //desliga o pino digital
      break;
    case 3:
      Serial.print(4);
      digitalWrite(RELE1, HIGH); //liga o pino digital
      digitalWrite(RELE2, HIGH); //liga o pino digital
      digitalWrite(RELE3, HIGH); //liga o pino digital
      digitalWrite(RELE4, HIGH); //liga o pino digital
      break;
  }
  Serial.println(" resistências");
  delay(1000);
}
}

```

Fonte: Autoria Própria (2018)

Para se medir a velocidade do ar via comunicação com o microcontrolador, estudou-se a Lei de Faraday (1845) e acoplou-se uma hélice em um motor (Figura 3), que ao receber uma velocidade de ar, gera uma corrente elétrica que pode ser lida pelo microcontrolador. Este sistema apresentou dificuldade de padronização e imprecisão do aparato, de modo que optou-se por apenas validar o fluxo de ar regulando-o pelo microcontrolador eliminando assim a necessidade destes sensores.

Figura 3. Hélice acoplada ao motor / Sensor de fluxo de ar.



Fonte: Autoria Própria (2018)

Para a leitura da corrente gerada pelo motor foram utilizadas portas analógicas do Arduino que trabalham com diferentes voltagens, podendo criar

uma relação de voltagem com a velocidade atual que foi estabelecida pelo algoritmo ilustrado na Figura 4.

Para finalizar o sistema foi levantada a possível criação de uma placa de circuito impresso integrando diretamente os relês, sensores e o microcontrolador, podendo ser implementados também interações com o usuário, registro de dados ou até mesmo controle sem fio, criando uma interface que facilite a utilização do sistema.

Figura 4. Algoritmo desenvolvido para a determinação da voltagem.

```

// Análisis de Voltaje en un Secador
// Autor: [Nombre]
// Fecha: [Fecha]

// Definición de constantes y variables globales
const int PIN_SENSOR = 0; // Pin de conexión del sensor de voltaje
const int PIN_LED = 13; // Pin de conexión del LED
const int PIN_RELAY = 12; // Pin de conexión del relé
const int PIN_BUTTON = 2; // Pin de conexión del botón

// Variables globales
float voltaje = 0; // Valor de voltaje actual
float velocidad = 0; // Velocidad actual
float temperatura = 0; // Temperatura actual
float humedad = 0; // Humedad actual
float tiempo = 0; // Tiempo de ejecución
float max_voltaje = 10; // Valor máximo de voltaje
float min_voltaje = 0; // Valor mínimo de voltaje
float max_velocidad = 100; // Valor máximo de velocidad
float min_velocidad = 0; // Valor mínimo de velocidad
float max_temperatura = 100; // Valor máximo de temperatura
float min_temperatura = 0; // Valor mínimo de temperatura
float max_humedad = 100; // Valor máximo de humedad
float min_humedad = 0; // Valor mínimo de humedad

// Función para inicializar el sistema
void inicializar() {
  pinMode(PIN_SENSOR, INPUT);
  pinMode(PIN_LED, OUTPUT);
  pinMode(PIN_RELAY, OUTPUT);
  pinMode(PIN_BUTTON, INPUT);
  digitalWrite(PIN_LED, LOW);
  digitalWrite(PIN_RELAY, LOW);
  digitalWrite(PIN_BUTTON, LOW);
}

// Función para leer el sensor de voltaje
float leerSensor() {
  return analogRead(PIN_SENSOR);
}

// Función para convertir el valor del sensor a voltaje
float convertirVoltaje(float valor) {
  return valor * 0.00476;
}

// Función para leer el sensor de velocidad
float leerSensorVelocidad() {
  return digitalRead(PIN_SENSOR);
}

// Función para convertir el valor del sensor a velocidad
float convertirVelocidad(float valor) {
  return valor * 100;
}

// Función para leer el sensor de temperatura
float leerSensorTemperatura() {
  return digitalRead(PIN_SENSOR);
}

// Función para convertir el valor del sensor a temperatura
float convertirTemperatura(float valor) {
  return valor * 100;
}

// Función para leer el sensor de humedad
float leerSensorHumedad() {
  return digitalRead(PIN_SENSOR);
}

// Función para convertir el valor del sensor a humedad
float convertirHumedad(float valor) {
  return valor * 100;
}

// Función para controlar el relé
void controlarRelay(float voltaje) {
  if (voltaje > max_voltaje) {
    digitalWrite(PIN_RELAY, HIGH);
  } else {
    digitalWrite(PIN_RELAY, LOW);
  }
}

// Función para controlar el LED
void controlarLED(float voltaje) {
  if (voltaje > max_voltaje) {
    digitalWrite(PIN_LED, HIGH);
  } else {
    digitalWrite(PIN_LED, LOW);
  }
}

// Función para controlar el botón
void controlarBoton() {
  if (digitalRead(PIN_BUTTON) == HIGH) {
    // Acción al presionar el botón
  }
}

// Función para calcular el promedio de los últimos 10 valores
float calcularPromedio(float *datos, int cantidad) {
  float suma = 0;
  for (int i = 0; i < cantidad; i++) {
    suma += datos[i];
  }
  return suma / cantidad;
}

// Función para calcular el promedio de los últimos 10 valores de voltaje
float calcularPromedioVoltaje() {
  float *datos = new float[10];
  for (int i = 0; i < 10; i++) {
    datos[i] = leerSensor();
  }
  return calcularPromedio(datos, 10);
}

// Función para calcular el promedio de los últimos 10 valores de velocidad
float calcularPromedioVelocidad() {
  float *datos = new float[10];
  for (int i = 0; i < 10; i++) {
    datos[i] = leerSensorVelocidad();
  }
  return calcularPromedio(datos, 10);
}

// Función para calcular el promedio de los últimos 10 valores de temperatura
float calcularPromedioTemperatura() {
  float *datos = new float[10];
  for (int i = 0; i < 10; i++) {
    datos[i] = leerSensorTemperatura();
  }
  return calcularPromedio(datos, 10);
}

// Función para calcular el promedio de los últimos 10 valores de humedad
float calcularPromedioHumedad() {
  float *datos = new float[10];
  for (int i = 0; i < 10; i++) {
    datos[i] = leerSensorHumedad();
  }
  return calcularPromedio(datos, 10);
}

// Función para mostrar los datos en el monitor
void mostrarDatos() {
  Serial.print("Voltaje: ");
  Serial.println(voltaje);
  Serial.print("Velocidad: ");
  Serial.println(velocidad);
  Serial.print("Temperatura: ");
  Serial.println(temperatura);
  Serial.print("Humedad: ");
  Serial.println(humedad);
  Serial.print("Tiempo: ");
  Serial.println(tiempo);
}

// Función principal
void setup() {
  inicializar();
  Serial.begin(9600);
}

void loop() {
  leerSensor();
  leerSensorVelocidad();
  leerSensorTemperatura();
  leerSensorHumedad();
  calcularPromedioVoltaje();
  calcularPromedioVelocidad();
  calcularPromedioTemperatura();
  calcularPromedioHumedad();
  controlarRelay(voltaje);
  controlarLED(voltaje);
  controlarBoton();
  mostrarDatos();
  delay(1000);
}

```

Fonte: Autoria Própria (2018)

CONCLUSÃO

Foi possível desenvolver um sistema diretamente integrado com resistências, relês, sensores e o microcontrolador a baixo custo para um secador didático de pequena produção.

A simulação realizada com LEDs serviu para verificar que será possível controlar, por meio de relês, o funcionamento de uma a quatro resistências elétricas para aquecimento do ar dentro do secador. O algoritmo escrito serviu



para que o Arduino conseguisse controlar o acendimento de quatro LEDs por meio de quatro relês.

A tarefa mais difícil está relacionada com a escolha de um conjunto hélice-motor eletrônico capaz de fornecer um *range* de voltagem que possa ser correlacionado, via Arduino, com a velocidade do fluxo de ar dentro do secador. Já foi possível verificar que a hélice de *cooler* de computadores tem excessiva resistência. Por isso, não é possível “quebrar” sua inércia com baixas velocidades de fluxo de ar.

REFERÊNCIAS

GAVA, A. J. Princípios de tecnologia de alimentos. São Paulo: Nobel, 1984.

NETO, A.R.; TERUEL, B. Supervisão e controle automático de sistema de secagem de produtos agrícolas através do software termicont. Bioeng. Campinas, v.2, n.1, p. 71-79, jan/abr., 2008.

AGRADECIMENTOS

Agradeço ao CNPq pela bolsa PIBIC de Ensino Médio, por toda estrutura da UTFPR e seus docentes altamente qualificados, e pelas oportunidades ofertadas pelo Instituto Federal do Paraná.