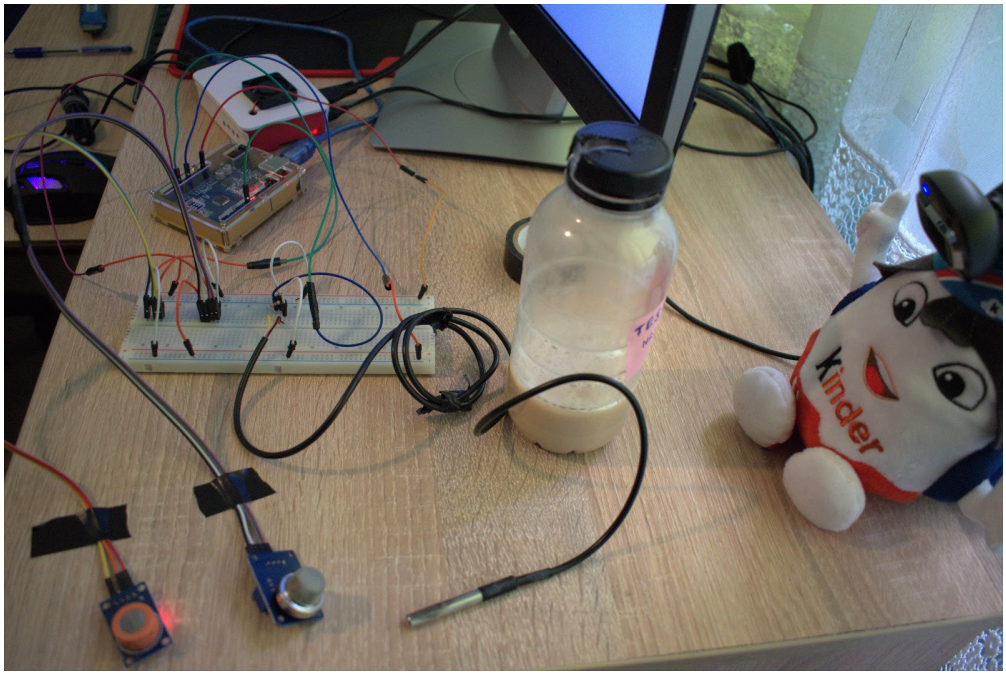
Stefan Andrei

Alcohol Monitoring Device with Pi

Cover Image



Description:

Realizarea unui smart device care ne permite măsurarea exactă a indicilor de fermentație a amestecurile pe bază de fructe ,în vederea producerii de băuturi alcoolice cu ajutorul unei configurații Master-Slave dintre un Raspberry Pi 3 Model B+ și un Arduino Uno R3 prin comunicație serială .

Fermentația alcoolică este procesul prin care se formează alcool etilic și dioxid de carbon, sub acțiunea diferitelor microorganisme,în acest sens arduino va măsura cantitatea de ethanol cu un senzor MQ-3,cantitatea de Co2 cu un senzor MQ-135 și temperatura medie a amestecului cu ajutorul unei sonde de temperatura DS18B20. Datele măsurate vor fi trimise prin serială și recepționate de RPI ,urmând ca acestea să fie procesate pt vizualizare pe o pagină web .În cadrul acestui proiect eu mă voi ocupa de partea de

senzoristică si testare a aplicației.
Proiect Hackster

Povestea din spatele proiectului:
Ideea proiectului a venit din dorința de a automatiza și a îmbunătăți procesul de fermentație atât de des întâlnit în orice gospodărie din România și din pasiunea membrilor echipei pt fabricarea băuturilor alcoolice ,încercând astfel să îmbine utilul cu plăcutul.

În faza de început, device ul creat de noi poate monitoriza un singur recipient destinat fermentării, însă pot fi adăugate mai multe plăci arduino care să monitorizeze în mod separat recipiente de diferite mărimi,cantități și materie primă .

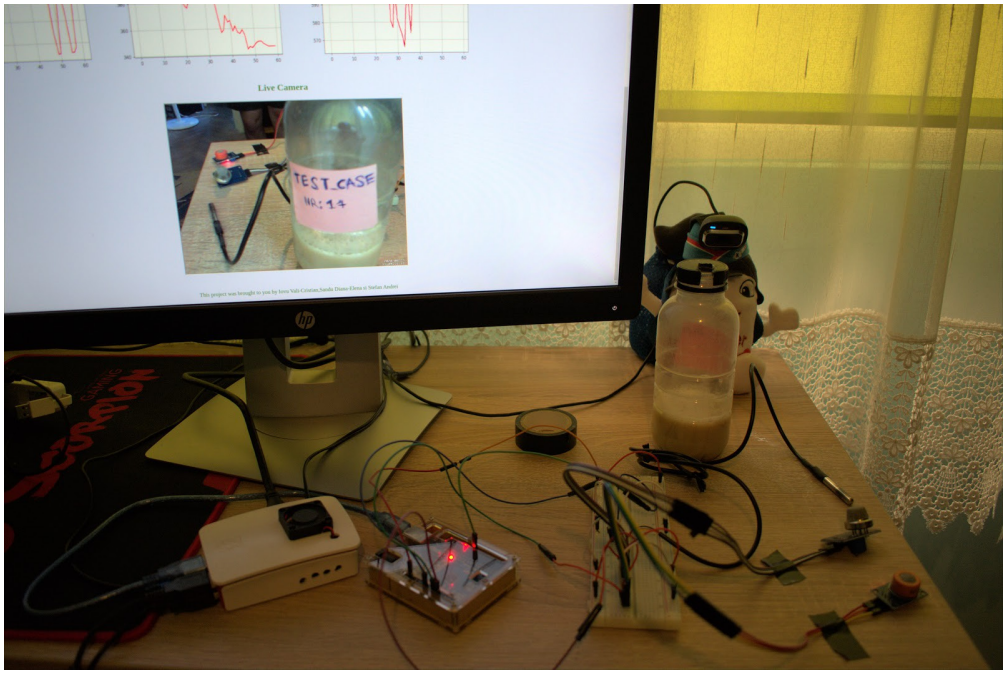
Descrierea procesului:
Fermentarea alcoolică este un proces biochimic complex în timpul căruia drojdiile transformă zaharurile din amestecul de fructe în etanol, dioxid de carbon și alte produse secundare metabolice care contribuie la compoziția chimică și proprietățile senzoriale ale produsului final.

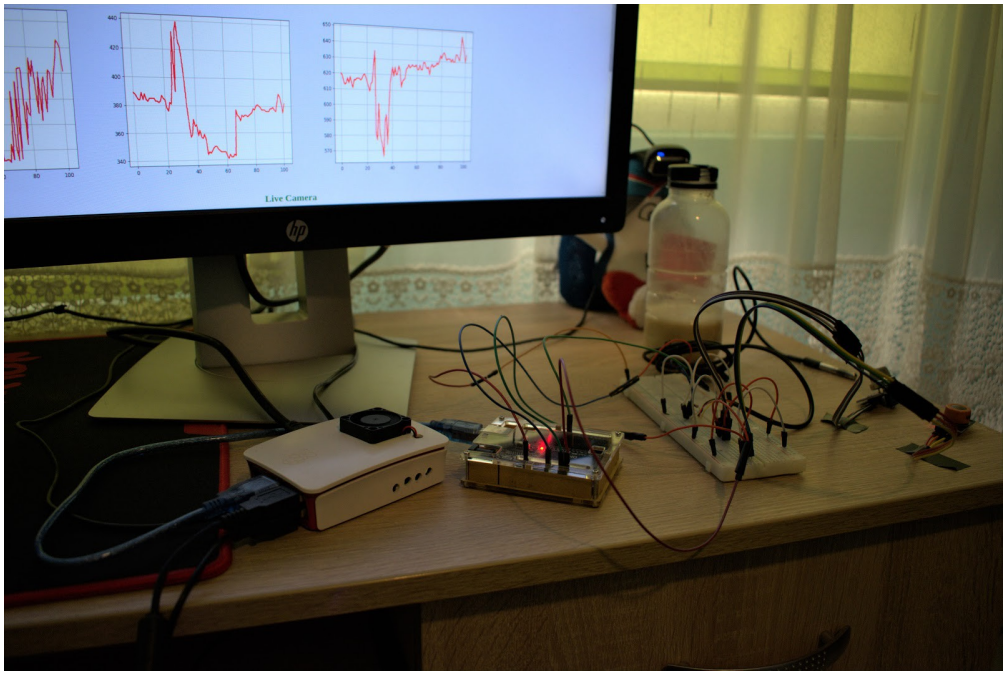
$$C_6H_{12}O_6 \longrightarrow 2C_2H_5OH + 2CO_2 + 22Kcal$$

Mai sus este descrisă formula chimică a fermentatiei unde C6H12O6 (glucoza) este descompusă de bacterii în 2C2H5OH (etanol) și 2CO2 (dioxid de carbon) plus o cantitate de căldură degajată.

În prima instanță ne dorim să urmărim creșterea cantității de dioxid de carbon în amestec,într-o fermentare uniformă aceasta ar trebui să aibă loc sub formă unei evoluții exponențiale,urmând ca după ce toată glucoza este consumată de drojdie,nivelul de C02 să stagneze.De asemenea este foarte importantă măsurarea nivelul de C02 deoarece un nivel prea mare ar putea duce la inhibarea fermentatiei, alterarea ph-ului amestecului și acidifierea acestuia ,probleme care pot distruge gustul,aroma și proprietățile produsului finit.
Urmărim de asemenea și măsurarea nivelului de ethanol produs , în funcție de tipul amestecului,fructele folosite sau zahărul adăugat acesta poate oferi date importante privind cantitatea de alcool obtinuta.Astfel după monitorizarea mai multor tipuri de amestecuri utilizatorul își poate face o idee despre rata de conversie a amestecului brut în alcool.
Un alt element care joacă un rol critic în fermentație și pe care ne dorim sa il masuram este temperatura. Drojdia trebuie să fie suficient de caldă pentru a fi sănătoasa, dar prea multă căldură va stresa și va distruge bacteriile care se ocupă cu metabolizarea glucozei in alcool. O temperatura prea rece și drojdia va fi lentă și somnolentă. Pe măsură ce temperatura crește, viteza de fermentare accelerează. Odată cu creșterea vitezei de fermentare, mai mulți compuși aromatici sunt produși.

Photos:





Components and apps:

Componente hardware:

1. Raspberry Pi 3 Model B+
2. Card MicroSD Original de 16 GB cu RASPBIAN

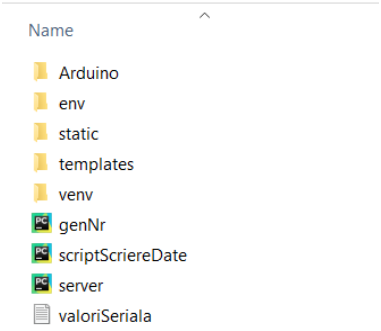
3. Alimentator de 2.5 A, 5.1 V
4. 3 Radiatoare pentru Raspberry Pi 3
5. Cooler Raspberry Pi
6. Placă de dezvoltare Arduino UNO (ATmega328p)
7. Senzor de Temperatura Rezistent la Apa DS18B20
8. Modul Senzor de Gaz MQ-135
9. Modul Senzor de Alcool Gazos MQ-3
10. Fire Colorate Mamă-Tată
11. Breadboard
12. Cablu USB CH340
13. Rezistor 0.25W 4.7KΩ

Componente software:

1. Raspbian OS
2. Arduino IDE

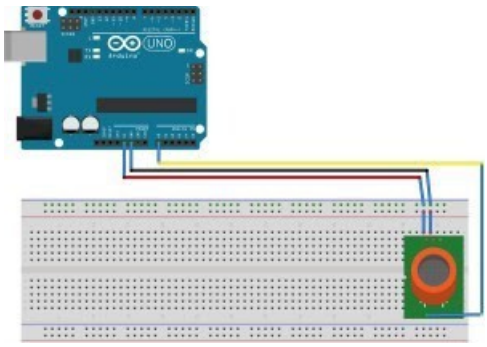
Schematics:

Project Structure



Modul Senzor de Gaz MQ-135

A0 Analog output -connected to A0 on Arduino
D0 Digital output - not used
GND Ground
Vcc Supply(5V)



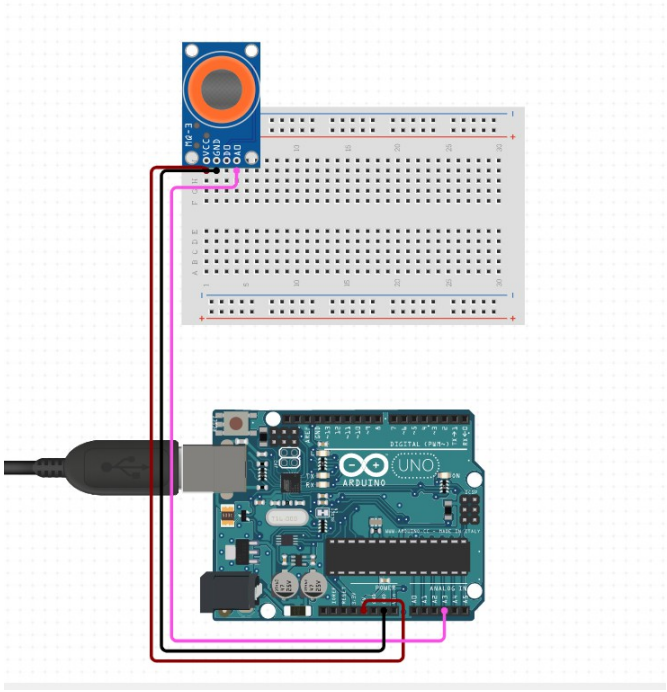
Modul Senzor Alcool MQ-3

A0 Analog output -connected to A3 on Arduino

D0 Digital output - not used

GND Ground

Vcc Supply(5V)

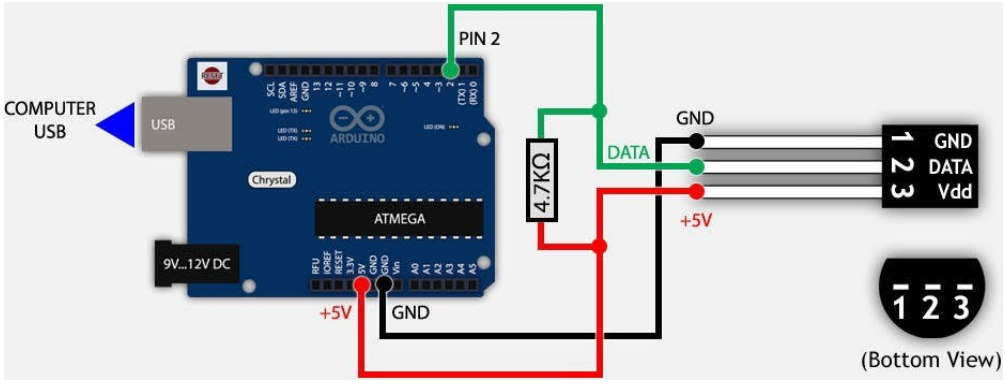


Senzor de Temperatura Rezistent la Apa DS18B20

D0 Digital output - connected to port 2

GND Ground

Vcc Supply(5V)



Code:

Arduino_Code.ino

```
//-- Proiect Sisteme cu Microprocesoare
//-- Titlu: Alcohol Monitoring Device
//-- Autor: Iovu Vali Cristian
//-- Grupa: 1309B
//-- Conținut fișier: Codul C pentru Senzorii conectati la Arduino

//--------Nota : Acest cod trebuie urcat pe placa Arduino si rulat cel putin 24h pentru autocalibrarea senzorilor
#include <LiquidCrystal.h>

#include <OneWire.h>
#include <DallasTemperature.h>
#define ONE_WIRE_BUS 8
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);

float Celsius = 0;
float Fahrenheit = 0;
int sensorValueMQ_135;
float sensorValueMQ_3;
float sensorValueMQ_135_Ethanol;
float sensor_volt;
float RS_gas;
float R0;
int R2 = 2000;
float voltage_ethanol;

void setup()
 {
   pinMode(LED_BUILTIN, OUTPUT);
   //sensors.begin();
   Serial.begin(9600);                    // setam comunicatia seriala cu un baudrate de 9600
 }
void loop()
```

```
{
 //Cod Folosit pentru Calibrarea Senzorului MQ_135
 /*
  float sensor_volt;
 float RS_gas; // Get value of RS in a GAS
 float ratio; // Get ratio RS_GAS/RS_air
 int sensorValue = analogRead(A0);
 sensor_volt=(float)sensorValue/1024*5.0;
 RS_gas = (5.0-sensor_volt)/sensor_volt; // omit *RL


 ratio = RS_gas/0.03;  // ratio = RS/R0


 Serial.print("sensor_volt = ");
 Serial.println(sensor_volt);
 Serial.print("RS_ratio = ");
 Serial.println(RS_gas);
 Serial.print("Rs/R0 = ");
 Serial.println(ratio);

 Serial.print("\n\n");
 delay(1000);
 */

//Citire Cantitate C02 Senzor
sensorValueMQ_3 = analogRead(A0);
// sensorValueMQ_3_Voltage=sensorValueMQ_3/1024*5.0;

//Citire Cantitate Ethanol Senzor
//Se face o medie aritmetica pentru 10 citiri pentru a oferi o medie cat mai precisa
 for(int i=0;i<10;i++)
 {
   sensorValueMQ_135_Ethanol+= analogRead(A1);
   delay(10);
 }
 sensorValueMQ_135= (sensorValueMQ_135_Ethanol/10);
 delay(600);
```

```
    //Citire Temperatura
    sensors.requestTemperatures();
    Celsius = sensors.getTempCByIndex(0);
    delay(600);

    //Trimitere date pe seriala
    String Send=String(Celsius)+String("-")+String(sensorValueMQ_3)+String("-")
+String(sensorValueMQ_135);
    Serial.println(Send);



    delay(3000);

}
```

scriptScriereDate.py

```
#-- Proiect Sisteme cu Microprocesoare
#-- Titlu: Alcohol Monitoring Device
#-- Autor: Iovu Vali Cristian Stefan-Andrei Sandu Diana-Elena
#-- Grupa: 1309B
#-- Conținut fișier: Codul Python pentru comunicatia seriala

import serial
import os
import time
import shutil
if __name__ == '__main__':
    ser = serial.Serial('/dev/ttyUSB0', 9600, timeout=1)
    f = open("valoriSeriala_copy.txt","a")

    ser.flush()
    while True:
        if ser.in_waiting > 0:
            line = ser.readline().decode('utf-8').rstrip()
            line = line + "\r\n"
            print(line)
            f.write(line)
            original =
```

```
r'/home/pi/Documents/SM_RPI_ARDUINO_Project/rpiWebServer/valoriSeriala_copy.txt'
                target =
r'/home/pi/Documents/SM_RPI_ARDUINO_Project/rpiWebServer/valoriSeriala.txt'
                shutil.copyfile(original, target)
                time.sleep(1)

                f.flush()
```

Server.py

```
#!/usr/bin/env python
#-- Proiect Sisteme cu Microprocesoare
#-- Titlu: Alcohol Monitoring Device
#-- Autor: Stefan-Andrei Sandu Diana-Elena
#-- Grupa: 1309B
#-- Conținut fișier: Codul Python pentru realizarea serverului FLASK

from flask import Flask, render_template
import datetime
import io
import os
import random
from flask import Response
import matplotlib
matplotlib.use('agg')
from matplotlib.backends.backend_agg import FigureCanvasAgg as FigureCanvas
from matplotlib.figure import Figure
from enum import Enum

class Data(Enum):
  temperature = 0
  ethanol = 1
  carbon = 2

app = Flask(__name__)
@app.route("/")
def index():

  last_line = "0-0-0"
```

```
  with open("valoriSeriala.txt", "r", os.O_NONBLOCK) as file:
    first_line = file.readline()
    for last_line in file:
      pass
  last_line = last_line.split('-')
  temperatureValue = last_line[0]
  ethanolValue = last_line[1]
  carbonValue = last_line[2]
  now = datetime.datetime.now()
  timeString = now.strftime("%Y-%m-%d %H:%M")
  templateData = {
    'title' : 'HELLO!',
    'time': timeString,
    'temperature': temperatureValue,
    'ethanol': ethanolValue,
    'carbon': carbonValue
    }
  return render_template('index.html', **templateData)

def create_figure(nrPlot):
  fig = Figure()
  fig.set_facecolor("#e6e6ff")
  axis = fig.add_subplot(1, 1, 1)
  v = []
  f=open("valoriSeriala.txt", "r+", os.O_NONBLOCK)
  lines = f.readlines()
  for line in lines:
    if line!='\n':
      dataRecived = line
      values = dataRecived.split("-")
      value = values[nrPlot]
      v.append(float(value))
      axis.plot(v,'r-')
      axis.grid(True)
  return fig

def plot_png(nrPlot):
  fig = create_figure(nrPlot)
```

```
    output = io.BytesIO()
    FigureCanvas(fig).print_png(output)
    return Response(output.getvalue(), mimetype='image/png')

@app.route('/temperaturePlot.png')
def plot_temperature():
    return plot_png(Data.temperature.value)

@app.route('/ethanolPlot.png')
def plot_ethanol():
    return plot_png(Data.ethanol.value)

@app.route('/carbonPlot.png')
def plot_carbon():
    return plot_png(Data.carbon.value)

if __name__ == "__main__":
    app.run(host='0.0.0.0', port=8082, debug=True)
```

Index.html

```
<!-- Proiect Sisteme cu Microprocesoare -->
<!-- Titlu: Alcohol Monitoring Device -->
<!-- Autor: Iovu Vali Cristian Stefan-Andrei Sandu Diana-Elena -->
<!-- Grupa: 1309B -->
<!-- Conținut fișier:  Codul HMTL+JS pentru pagina de index care va prelua datele de la server -->

<!DOCTYPE html>
  <head>


    <title>{{ title }}</title>
    <link rel="stylesheet" href="../static/stil.css">
    <style>

   h1,h2{ color:green; }
    h3 { color: #800080; }
   #container {
```

```
        margin: 0px auto;
        width: 500px;
        height: 375px;
        border: 10px #333 solid;
              }
        #videoElement {
        width: 500px;
        height: 375px;
        background-color: #666;
              }
        .center{
      margin: auto;
         }

         iframe{
      height:480px;
      width:640px;
      }
         </style>
         <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

      <script>
        setInterval(function(){ $( "#data" ).load(window.location.href + " #data" ); },1000);
      </script>

        </head>

          <body style="background-color:#e6e6ff">
         <h1 style="text-align:center;">Alcohol Monitoring Device</h1>
          <br>
          <br>
          <h2>Current data:</h2>

          <div id="data">
         <h3><i>Date & Time: {{ time }}</i></h3>
        <h3><i>Temperature: {{ temperature }} &#8451;</i></h3>
      <h3><i>Ethanol concentration: {{ ethanol }} mg/l</i></h3>
```

```
        <h3><i>Carbon Dioxide: {{ carbon }} ppm(parts per milion)</i></h3>
    <br>
    <br>
    <br>
    </div>

        <h3 style="padding-left:6em;color:green"><i>Fig1.Monitorizare temperatura
<span style="padding-left:14em"> Fig2.Monitorizare ethanol </span> <span
style="padding-left:15em"> Fig3.Monitorizare carbon </span></i></h3>
        <img id="temperature" src="/temperaturePlot.png" alt="temperature plot"
width="500" height="450" style="border=0px;margin:0px;float:left;">
        <img id="ethanol" src="/ethanolPlot.png" alt="ethanol plot" width="500"
height="450" style="border=0px;margin:0px;float:left;">
        <img id="carbon" src="/carbonPlot.png" alt="carbon plot" width="500"
height="450">


    <h2 align="center">Live Camera</h2>
        <p align="center"><iframe src="http://192.168.1.6:8081"></iframe></p>
    <br>

        <footer>

    <p  style="text-align:center; color:green">This project was brought to you by Iovu
Vali-Cristian,Sandu Diana-Elena si Stefan Andrei</p>

        </footer>

        <script>



        function refresh(node)
        {
          var times = 9000; // gap in Milli Seconds;

          (function startRefresh()
          {
```

```
            var address;
            if(node.src.indexOf('?')>-1)
             address = node.src.split('?')[0];
            else
             address = node.src;
            node.src = address+"?time="+new Date().getTime();

            setTimeout(startRefresh,times);
          })();

      }

      setTimeout( function()
      {

        var temp_node = document.getElementById('temperature');
var ethanol_node = document.getElementById('ethanol');
var carbon_node = document.getElementById('carbon');
        refresh(temp_node);
refresh(ethanol_node);
refresh(carbon_node);
console.log("est");
      },3000);
    </script>
    </body>
</html>
```