



SISTEMAS EMBARCADOS II

Trabalho Prático 2

BRUNO DAMASCENO REIS	11721EAU008
ÍTALO MARANGONI DE SOUZA	11811EAU014
MATHEUS RODRIGUES CUNHA COELHO	11711ETE002
PEDRO JACOB FAVORETO	11721EAU003
WANDER VICTOR VERÇOSA MARES	11811EAU010

Objetivos

O objetivo do trabalho prático em questão foi compreender os assuntos abordados em aula e aplicá-los na prática.

No sistema desenvolvido, foi feita uma representação de automação residencial IoT, através de um Arduino UNO, programação C++, Python e HTML.

Desenvolvimento

O desenvolvimento constituiu-se na aplicação dos conceitos passados em aula, onde foi realizada uma representação de automação residencial utilizando um Arduino UNO, no qual cada led habilitado nas saídas digitais representa um dos 7 cômodos da casa:

- Garagem;
- Sala;
- Banheiro;
- Cozinha;
- Quarto 1;
- Quarto 2;
- Quarto 3.

No código *webLED.ino* são definidas as portas digitais referentes a cada cômodo. Posteriormente, são definidos os estados dessas portas (no código em questão, cada porta é habilitada como *OUTPUT*). Dessa forma, é definida uma variável do tipo *String* que irá comparar as strings geradas na leitura serial do Arduino.

Realizada a leitura serial, o valor lido é comparado e, em caso de valor verdadeiro, as saídas anteriormente definidas são habilitadas. Por exemplo: em caso de leitura serial “*Gon*”, a porta “*G*”, que define a garagem, é habilitada em “*HIGH*”; mas em caso de leitura serial “*Goff*”, a mesma porta é habilitada como “*LOW*”, e assim sucessivamente para as outras portas digitais.

De forma geral, há três aplicações: em linguagem HTML, Python e C++ (Arduino).

No script em HTML, há um vínculo entre botão e string. É nesta área que ocorre a interação entre usuário e sistema. O acionamento do botão virtual envia uma string ao código em Python.

A aplicação em Python, por sua vez, faz a intermediação dessa string para a aplicação em C++, no próprio Arduino, interpretando as mensagens enviadas pelo HTML e transmitindo-as à aplicação do Arduino via comunicação serial.

No Arduino, essa comunicação serial já está pré-definida, uma vez que estabelece uma conexão entre as informações que chegam com as saídas digitais.

Bibliotecas utilizadas:

- **Flask:** Biblioteca que importa o servidor. Traz uma *request* para chamada de função e *render_template* para armazenar e carregar os dados gerados para o servidor.
- **Serial:** Biblioteca utilizada para encapsular dados e transmitir os mesmo via porta serial.

Resultados

Os resultados foram favoráveis de acordo com o esperado pelo grupo. A aplicação consegue executar os comandos nos LED's através dos botões da página em HTML, que é onde toda a aplicação é interfaceada com o usuário.

Conclusão

A partir do desenvolvimento desse projeto, o grupo contornou as dificuldades relacionadas à implementação das funcionalidades exigidas, fazendo com que o sistema funcionasse da forma desejada. Dessa forma, os integrantes do grupo concordam que foi de suma importância para agregar ao conhecimento a respeito de diferentes formas de programação e comunicação entre as mesmas.

Apêndice

Código HTML:

```
<!DOCTYPE html>
<html>
  <head>
    <title>web lights</title>
    <style>
```

```
table.GeneratedTable {
width: 75%;
background-color: #ffffff;
border-collapse: collapse;
border-width: 2px;
border-color: #000000;
border-style: solid;
color: #000000;
}

table.GeneratedTable td, table.GeneratedTable th {
border-width: 2px;
border-color: #000000;
border-style: solid;
padding: 3px;
}
.interItems {
text-align: center;
}
.interactiveButtons {
padding: 2em;
display: flex;
border-radius: 50%;
justify-content: center;
align-items: center;
height: 200px;
display: inline-block;
}
.onButton {
background-color: #4CAF50;
width: 5em;
height: 5em;
}
.offButton {
background-color: #e7e7e7;
color: black;
width: 5em;
height: 5em;
}
</style>
<link rel="icon" type="image/x-icon" href="pics/fav.png">
</head>
<body>
```

```
<div class='interactiveButtons'>
  <h1>AutoHome!</h1>
  <form action='#' method='POST'>
    <table class="GeneratedTable">

      <tbody>
        <tr>
          <table class="GeneratedTable">
            <tbody>
              <tr>
                <td>Portão</td>
              </tr>
              <tr>
                <td>Garagem
                  <button class='onButton'
onclick="GarageOn()" name="Gon" value="Gon">on</button>
                  <button class='offButton'
onclick="GarageOff()" name="Goff" value="Goff">off</button>
                </td>
              </tr>
            </tbody>
          </table>
        <td>
          <table class="GeneratedTable">
            <tbody>
              <tr>
                <td>
                  <table class="GeneratedTable">
                    <tbody>
                      <tr>
                        <td>Sala
                          <button class='onButton'
onclick="SOn()" name="Son" value="Son">on</button>
                          <button class='offButton'
onclick="SOff()" name="Soff" value="Soff">off</button>
                        </td>
                      <td>
                        <table class="GeneratedTable">
                          <tbody>
                            <tr>
                              <td>corredor</td>
                            </tr>
                          </tbody>
                        </table>
                      </td>
                    </tr>
                  </table>
                </td>
              </tr>
            </tbody>
          </table>
        </td>
      </tr>
    </tbody>
  </table>
</form>
</div>
```

```

        <tr>
            <td>Banheiro
                <button class='onButton'
onclick="BanheiroOn()" name="Bon" value="Bon">on</button>
                <button class='offButton'
onclick="BanheiroOff()" name="Boff" value="Boff">off</button> </td>
        </tr>
    </tbody>
</table>

```

```

        </td>
        <td>Cozinha
            <button class='onButton'
onclick="CozinhaOn()" name="Con" value="Con">on</button>
            <button class='offButton'
onclick="CozinhaOff()" name="Coff" value="Coff">off</button>

        </td>
    </tr>
</tbody>
</table>

```

```

    </td>
</tr>
<tr>
    <td>
        <table class="GeneratedTable">
            <tbody>
                <tr>
                    <td>Quarto 1
                        <button class='onButton'
onclick="Q1On()" name="Q1on" value="Q1on">on</button>
                        <button class='offButton'
onclick="Q1Off()" name="Q1off" value="Q1off">off</button>
                    </td>
                    <td>Quarto 2

```

```

                                <button class='onButton'
onclick="Q2On()" name="Q2on" value="Q2on">on</button>
                                <button class='offButton'
onclick="Q2Off()" name="Q2off" value="Q2off">off</button>
                                </td>
                                <td>Quarto 3
                                <button class='onButton'
onclick="Q3On()" name="Q3on" value="Q3on">on</button>
                                <button class='offButton'
onclick="Q3Off()" name="Q3off" value="Q3off">off</button>

                                </td>
                                </tr>
                                </tbody>
                                </table>

                                </td>
                                </tr>
                                </tbody>
                                </table>

                                </td>
                                </tr>
                                </tbody>
                                </table>

                                </td>
                                </tr>
                                </tbody>
                                </table>

                                </form>
                                </div>
                                </div>
                                <button class='disconnectBtton' onclick="disconnect()" name="dis"
                                value="dis">disconnect</button>
                                </body>
                                </html>

```

Código Python:

```
from flask import Flask, render_template, request
import serial

serialcom = serial.Serial('COM6', 9600)
serialcom.timeout = 1

app = Flask(__name__)

def GarageOn():
    serialcom.write(str('Gon').encode())

def GarageOff():
    serialcom.write(str('Goff').encode())

def SalaOn():
    serialcom.write(str('Son').encode())

def SalaOff():
    serialcom.write(str('Soff').encode())

def BanheiroOn():
    serialcom.write(str('Bon').encode())

def BanheiroOff():
    serialcom.write(str('Boff').encode())

def CozinhaOn():
    serialcom.write(str('Con').encode())

def CozinhaOff():
    serialcom.write(str('Coff').encode())

def Quarto1On():
    serialcom.write(str('Qlon').encode())

def Quarto1Off():
    serialcom.write(str('Qloff').encode())
```



```
def Quarto2On():
    serialcom.write(str('Q2on').encode())

def Quarto2Off():
    serialcom.write(str('Q2off').encode())

def Quarto3On():
    serialcom.write(str('Q3on').encode())

def Quarto3Off():
    serialcom.write(str('Q3off').encode())


def disconnect():
    serialcom.close()

@app.route("/", methods=['GET', 'POST'])
def index():
    if request.method == 'POST':
        if 'Gon' in request.form.to_dict():
            GarageOn()
        if 'Goff' in request.form.to_dict():
            GarageOff()
        if 'Son' in request.form.to_dict():
            SalaOn()
        if 'Soff' in request.form.to_dict():
            SalaOff()
        if 'Bon' in request.form.to_dict():
            BanheiroOn()
        if 'Boff' in request.form.to_dict():
            BanheiroOff()
        if 'Con' in request.form.to_dict():
            CozinhaOn()
        if 'Coff' in request.form.to_dict():
            CozinhaOff()
        if 'Q1on' in request.form.to_dict():
            Quarto1On()
        if 'Q1off' in request.form.to_dict():
            Quarto1Off()
        if 'Q2on' in request.form.to_dict():
            Quarto2On()
        if 'Q2off' in request.form.to_dict():
```

```

        Quarto2Off()

        if 'Q3on' in request.form.to_dict():
            Quarto3On()

        if 'Q3off' in request.form.to_dict():
            Quarto3Off()

        if 'dis' in request.form.to_dict():
            disconnect()

    return render_template('index.html')

if __name__ == "__main__":
    app.run()

```

Código C++ (Arduino):

```

#define G 13
#define S 12
#define B 11
#define C 10
#define Q1 9
#define Q2 8
#define Q3 7

String inByt;

void setup() {
    pinMode(G, OUTPUT);
    pinMode(S, OUTPUT);
    pinMode(C, OUTPUT);
    pinMode(B, OUTPUT);
    pinMode(Q1, OUTPUT);
    pinMode(Q2, OUTPUT);
    pinMode(Q3, OUTPUT);
    Serial.begin(9600);
    Serial.setTimeout(10);
}

void loop() {
    //
}

```

```
void serialEvent() {
    inByt = Serial.readStringUntil('\n');

    if (inByt == "Gon") {
        digitalWrite(G, HIGH);
    } else if (inByt == "Goff") {
        digitalWrite(G, LOW);
    } else if (inByt == "Son") {
        digitalWrite(S, HIGH);
    } else if (inByt == "Soff") {
        digitalWrite(S, LOW);
    } else if (inByt == "Con") {
        digitalWrite(C, HIGH);
    } else if (inByt == "Coff") {
        digitalWrite(C, LOW);
    } else if (inByt == "Bon") {
        digitalWrite(B, HIGH);
    } else if (inByt == "Boff") {
        digitalWrite(B, LOW);
    } else if (inByt == "Q1on") {
        digitalWrite(Q1, HIGH);
    } else if (inByt == "Q1off") {
        digitalWrite(Q1, LOW);
    } else if (inByt == "Q2on") {
        digitalWrite(Q2, HIGH);
    } else if (inByt == "Q2off") {
        digitalWrite(Q2, LOW);
    } else if (inByt == "Q3on") {
        digitalWrite(Q3, HIGH);
    } else if (inByt == "Q3off") {
        digitalWrite(Q3, LOW);
    } else {
        // continue;
    }
}
```

Referências

ARDUINO. **Arduino Docs**. Disponível em: <<https://docs.arduino.cc/>>. Acesso em: 23 mar. 2022.

FLASK. **Flask web development**. Disponível em: <<https://flask.palletsprojects.com/en/2.1.x/>>. Acesso em: 29 mar. 2022.

MOZILLA DOCS HTML. **HTML**. Disponível em: <<https://developer.mozilla.org/en-US/docs/Web/HTML>>. Acesso em: 25 mar. 2022.

PYTHON. **Python 3 Documentation**. Disponível em: <<https://docs.python.org/3/>>. Acesso em: 24 mar. 2022.