



UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA MECÂNICA
Curso de Graduação em Engenharia Aeronáutica
Curso de Graduação em Engenharia Mecatrônica



Trabalho 2

Sistemas Digitais para Mecatrônica

Alan Nascimento de Paula 11721EMT010
Caio Augusto de Medeiros 11421EMT023
Murilo Antônio da Silva Rocioli 11811EMT015
Eduardo Marques da Silva 11721EMT018
Eduardo Apolinario Lopes 11621EMT005

Uberlândia, agosto de 2022.

1. Introdução	3
2. Objetivo.....	3
4. Montagem	4
5. Resultados	4
6. Conclusão.....	5
7. Anexo I.....	5

1. Introdução

Diante de um novo e importante período no contexto das revoluções industriais. Os impactos da Indústria 4.0 e da transformação digital sobre produtividade, redução de custos e controle sobre processos produtivos estão modificando profundamente a estrutura das operações industriais. A necessidade de eficiência na indústria manufatureira passou a ser mais exigida, e para assegurar o sucesso, as corporações devem garantir a eficácia de investimentos associados a tempo, equipamentos e outras variáveis.

2. Objetivo

Esse trabalho tem como objetivo a implementação de um sistema de controle a partir de uma comunicação em nuvem e um microcontrolador de escolha do grupo.

3. Flask

Para a implementação do trabalho foi utilizado o FLASK, um micro framework que utiliza a linguagem Python para criar aplicativos Web. Caracterizado pela rapidez no desenvolvimento no qual o desenvolvedor se preocupa em apenas desenvolver o necessário para um projeto, sem a necessidade de realizar configurações que muitas vezes não são utilizadas.



Por se tratar de um sistema simples no qual vamos montar o FLASK se encaixa perfeitamente na nossa aplicação, pois por possuir apenas o necessário para o desenvolvimento de uma aplicação, um projeto escrito com FLASK é mais simples se comparado aos frameworks maiores, já que a quantidade de arquivos é muito menor e sua arquitetura é muito mais simples.

Para a sua utilização no Linux, utilizamos dos comandos pip para a instalação de pacotes para o Python.

```
sudo apt-get install python-pip
```

4. Montagem

Na montagem de nosso sistema foi utilizado um Arduino (Figura 1) uma plataforma de prototipagem eletrônica muito versátil que tem como objetivo principal tornar o acesso à prototipagem eletrônica mais fácil, mais barata e flexível. As versões mais simples da placa utilizam um microcontrolador da família Atmel AVR e uma linguagem de programação baseada em C/C++. Com ele é possível criar projetos variados em eletrônica, desde os mais simples até aplicações intermediárias como Internet das Coisas (IoT), Robôs, Sistemas de Automação Residencial ou Industrial, Alarmes e outros.



Figura 1: Representação Arduino UNO.

Para a representação de uma luz de algum cômodo utilizamos um LED (Figura 2), um componente eletrônico semiconductor, ou seja, um diodo emissor de luz, componente do tipo bipolar, ou seja, tem um terminal chamado anodo e outro, chamado catodo. Dependendo de como for polarizado, permite ou não a passagem de corrente elétrica e, conseqüentemente, a geração ou não de luz.



Figura 2: Representação de um LED.

5. Resultados

A construção da interface Front-End foi realizada com HTML para a interação do usuário com todo o sistema, no caso possibilita o mesmo de ligar e desligar o LED apenas com o clique dos respectivos botões como pode ser visto na Figura 3.



Figura 3: Interface FRONT-END do sistema.

Para a interação do sistema do Arduino com a página web foi utilizado um código em Python junto a biblioteca Flask para permitir a comunicação entre ambos. No controle do sistema de LED foi utilizado Arduino com a linguagem C/C++ pelo próprio software de escrita do microcontrolador, sendo toda a comunicação realizado pelo serial do mesmo. Os códigos de cada sessão se encontram em ANEXOS.

6. Conclusão

Portanto, o grupo encontrou dificuldades inicialmente na comunicação da biblioteca Flask do Python, após conseguir realizar a comunicação entre os sistemas do Arduino e da página Web conseguimos realizar com sucesso o objetivo do sistema de ligar e desligar um LED a partir de uma página Web.

Sendo um trabalho de conclusão do semestre podemos concluir que o mesmo foi de suma importância para a aplicação e teste de todo o conhecimento adquirido durante o período letivo e consideramos como uma experiencia e porta de entrada para a criação de futuros projetos e experiencia na vida acadêmica e profissional dos alunos.

7. Anexo I

Código Arduino:

```
##include <SoftwareSerial.h>

String inByt;
int Led = 4;

void setup() {

  Serial.begin(9600);
```

```

// Inicializando a saída
pinMode(Led, OUTPUT);
// Configurando a saída como desligado
digitalWrite(Led, LOW);
//Serial.begin(9600);
Serial.setTimeout(10);
}

void loop() {
  //

}

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

  if (inByt == "on") {
    digitalWrite(Led, HIGH);
  } else if (inByt == "off") {
    digitalWrite(Led, LOW);
  } //else {
    //continue;
  //}
}

```

-Código Python:

```

from flask import Flask, render_template, Response, request # módulos necessários
#from serial import Serial
import serial # conectar com arduino
import time

serialcom = serial.Serial('/dev/ttyACM0') # porta do arduino
serialcom.timeout = 1

app = Flask(__name__)

def ledOn():
    serialcom.write(str('on').encode())

def ledOff():
    serialcom.write(str('off').encode())

def disconnect():
    serialcom.close()

```

```

@app.route("/", methods=['GET', 'POST'])
def index():
    if request.method == 'POST':
        if 'on' in request.form.to_dict():
            ledOn()
        if 'off' in request.form.to_dict():
            ledOff()
        if 'dis' in request.form.to_dict():
            disconnect()

    return render_template('index.html') # página HTML

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

```

-Código HTML:

```

<!DOCTYPE html>
<html>
<head>
<title>Acendendo Led</title>
<style>
.interItems {
text-align: center;
}
.interactiveButtons {
padding: 2em;
display: flex;
    justify-content: center;
    align-items: center;
    height: 200px;
}
.onButton {
width: 10em;
height: 10em;
}
.offButton {
width: 10em;
height: 10em;
}
</style>
</head>
<body>
<div class='interItems'>
<h1>Sistemas Digitais Para Mecatrônica: Trabalho 2: IOT</h1>

```

```
<div class='interactiveButtons'>
<form action='#' method='POST'>
<button class='onButton' onclick="ledOn()" name="on" value="on">Ligar</button>
<button class='offButton' onclick="ledOff()" name="off" value="off">Desligar</button>
</form>
</div>
<button class='disconnectBtton' onclick="disconnect()" name="dis"
value="dis">Desconectar</button>
</div>
</body>
</html>
```