



Relatório de Práticas de Plataforma de Hardware para Internet das Coisas

Disciplina: IMD0904 -PLATAFORMAS DE HARDWARE PARA INTERNET DAS COISAS - 2023.2

Orientador: Prof. Leonardo Augusto

Universidade Federal do Rio Grande do Norte

Pedro Galvão do Amaral Neto
Raoni Marti Silvestre Silva



PRÁTICA 4 - SPIFFS

1. Introdução

A prática de uso do SPIFFS tem como objetivo experimentar a utilização da memória interna do ESP32 por meio da biblioteca SPIFFS. Para tanto, o projeto se baseia em duas partes, sendo a primeira o exercício, e a segunda o desafio.

Ao que tange o exercício, é necessário montar um circuito simples que liga e desliga um LED de acordo com o pressionamento de um botão:

1. Caso o botão esteja desligado, ao apertar o botão, ele é ligado;
2. Caso o botão esteja ligado, ao apertar o botão ele é desligado.

Toda vez que o estado do LED muda, é necessário salvar o último estado do LED em um arquivo .txt dentro da memória do ESP32, pois quando o microcontrolador for reiniciado, o LED se manterá no último estado salvo.

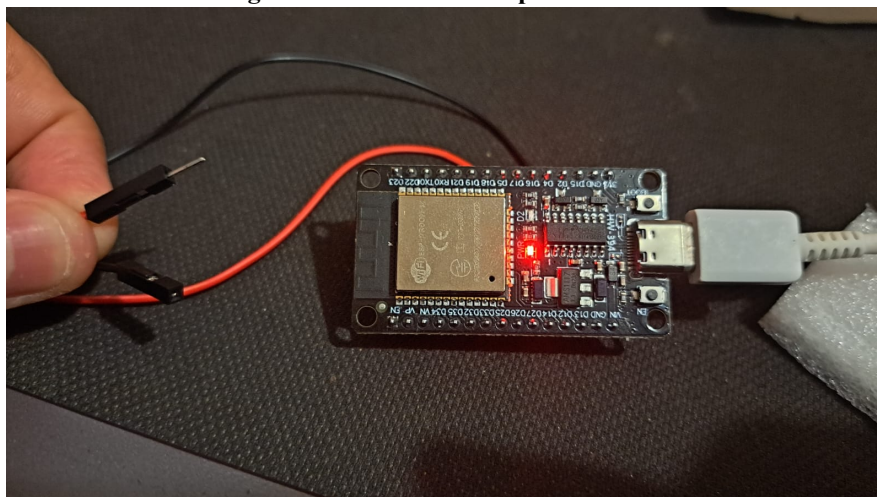
Prosseguindo para o desafio, é pedido que seja criada a lógica para salvar todas as data e hora em que o LED alterou seu estado em um arquivo de logs. Para pegar a data e hora é necessário utilizar um servidor NTP externo.



2. Montagem

Como grande parte desse projeto era a utilização de coisas internas ao microcontrolador, a parte de montagem acabou sendo bem simples, pois além do ESP32, só foram necessários dois fios, que acabam servindo como botões: quando conectam-se, o botão é pressionado, quando se desconectam, o botão é desconectado.

Figura 1 - “Circuito” completo



Fonte : Autoria própria (2023)

3. Códigos

O código a ser apresentado será a versão final, onde já foi implementada a parte do exercício e do desafio. Desse modo, algumas partes que são apresentadas no desafio estarão misturadas ou aparecerão antes da parte do exercício.

Primeiramente, é necessário destacar que para desenvolver o código, foram utilizadas as seguintes bibliotecas:

- FS (File System)
- SPIFFS (Alterar dados na memória do ESP32)
- WIFI (Acessar Wifi)
- NTPClient (Acessar a data e hora por meio da internet)

Além disso, é importante destacar que já foram implementadas as funções associadas ao SPIFFS `writeFile` e `readFile`, que escrevem e lêem dados na memória flash do ESP32.

Passando para a primeira parte importante, teremos:

Figura 2 - Setup

```
83  WiFi.begin(ssid, password);
84  while (WiFi.status() != WL_CONNECTED)
85  {
86      delay(500);          /* Aguarda meio segundo */
87      Serial.println(WiFi.status()); /* Conectando */
88  }
89  Serial.println("Conectado"); /* Conectado */
90  delay(200);
91
92  ntp.begin();             /* Inicia o protocolo */
93  ntp.forceUpdate();       /* Atualização */
94  ntp.setTimeOffset(-10800);
95  // formatFile();
96  You, 2 minutes ago • Uncommitted changes
97  openFS();
98  String s = readFile("/ledState.txt");
99  oldLedVal = s.toInt();
100  digitalWrite(LED, oldLedVal);
101  }
```

Fonte : Autoria própria (2023)

No setup, ocorre a inicialização do acesso a internet, seguido pelo acesso ao servidor NTP, e por fim, a abertura do arquivo `/ledState.txt`. Importante destacar que o código só passa da parte do wifi se ele conseguir se conectar com a rede informada nas variáveis “ssid” e

“password”. Além disso, nota-se que o servidor NTP ao ser contatado, é setado o “Offset” para o horário do Brasil (UTC -3).

Figura 3 - loop

```
103 void loop()
104 {
105     botaoVal = !digitalRead(botao);
106     if (botaoVal == 1 && botaoVal != oldVal)
107     {
108         ledVal = (ledVal == LOW ? HIGH : LOW);
109         digitalWrite(LED, ledVal);
110         delay(50);
111         data = ntp.getFormattedDate();
112         writeFile(data, "/ledLog.txt", "a+");
113         readFile("/ledLog.txt");
114     }
115     oldVal = botaoVal;
116
117     str = String(ledVal);
118     writeFile(str, "/ledState.txt", "w+");
119     delay(100);
120 }
```

Fonte: Autoria própria (2023)

No loop, o que acontece é que o valor do botão é analisado, caso seja um valor diferente do antigo e seja o valor 1 (botão ligado), o código entra na branch de alterar o valor do LED, salvar o último estado e adicionar um log.

Primeiramente, o valor do LED se altera para o oposto do que estava atualmente. Logo após isso, por meio da função `ntp.getFormattedDate()`, é armazenado o valor da data e hora atual por meio do servidor NTP, e então esse valor é escrito no final do arquivo `/ledLog.txt` e então aparece quais são os valores salvos no arquivo por meio do monitor Serial.

Independentemente de entrar ou não nessa branch, o que ocorre é que o valor do LED é armazenado dentro do arquivo `/ledState.txt`, e por fim, o código entra novamente no loop.

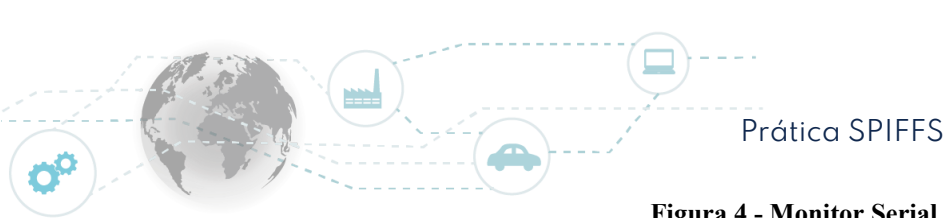
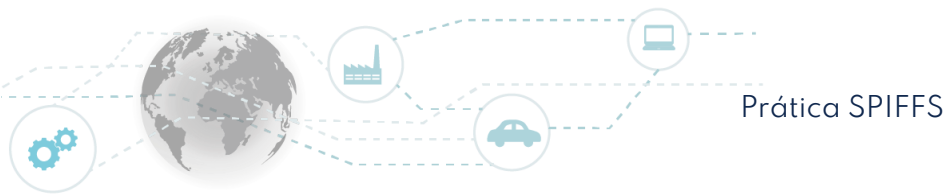


Figura 4 - Monitor Serial

```
Formantando SPIFFS
Formatou SPIFFS

Sistema de arquivos aberto com sucesso!
Read file
Read file
2023-11-09T19:19:43Z
Read file
2023-11-09T19:19:43Z
2023-11-09T19:19:47Z
Read file
```

Fonte: Autoria própria (2023)



4. Resultados e discussões

Em síntese, a prática envolvendo o SPIFFS proporcionou uma compreensão aprofundada e prática do funcionamento e da versatilidade dessa tecnologia embutida no ESP32. Ao explorar a implementação do SPIFFS para o armazenamento de dados em um sistema de arquivos integrado a um microcontrolador, conseguimos criar uma solução compacta e eficiente para o armazenamento persistente de informações em sistemas com recursos limitados.

Além disso, a sincronização do tempo por meio do uso de um servidor NTP fortaleceu a precisão temporal do sistema de forma econômica, sem a necessidade de um módulo RTC adicional, sendo essencial para aplicações que dependem do registro temporal preciso, como o registro de acesso de uma determinada sala ou local em um prédio. A integração desses recursos no ESP32 proporcionou uma base sólida para o desenvolvimento de projetos mais avançados e complexos e também mostrou a importância da gestão eficiente do armazenamento de dados e da sincronização temporal em dispositivos IoT.