MICROCONTROLADORES ESP8266 e ESP32
(TEORIA E PRÁTICA)

Edição
Julho/2025

15 - ETAPA 3 — Logger Ultrassônico com Buffer Circular (500 Registros Máx) em SPIFFS

Objetivo:

1. Gravar medições de distância (< 400cm) com data/hora no arquivo CSV no SPIFFS.
2. Limitar o arquivo a **500 registros fixos**.
3. Quando o arquivo atingir 500 linhas, o sistema automaticamente:
   - Remove a linha mais antiga (primeira linha).
   - Adiciona a nova medida no final (buffer circular).
4. Exibir o histórico via Web Page e permitir download do CSV atualizado.

Código Arduino IDE (Buffer Circular no Arquivo CSV):

```
#include <WiFi.h>
#include <WebServer.h>
#include <FS.h>
#include <SPIFFS.h>
#include <time.h>
#include <sys/time.h>

const char* ssid = "SEU_SSID";
const char* password = "SUA_SENHA";

WebServer server(80);

const int trigPin = 4;
const int echoPin = 5;

#define CSV_FILE "/medicoes.csv"
#define MAX_REGISTROS 500

// Configurar horário local com base no horário de compilação
void setLocalTimeFromBuild() {
  struct tm tm_build = {0};
  strptime(__DATE__ " " __TIME__, "%b %d %Y %H:%M:%S", &tm_build);
  time_t t = mktime(&tm_build);
  struct timeval now = { .tv_sec = t };
  settimeofday(&now, NULL);
}

// Remover a primeira linha do CSV para manter o limite
void removePrimeiraLinha() {
  File file = SPIFFS.open(CSV_FILE, FILE_READ);
  if (!file) return;
```

```
  File tempFile = SPIFFS.open("/temp.csv", FILE_WRITE);
  if (!tempFile) {
    file.close();
    return;
  }

  bool skipFirstLine = true;
  while (file.available()) {
    String line = file.readStringUntil('\n');
    if (skipFirstLine) {
      skipFirstLine = false;
      continue;
    }
    tempFile.println(line);
  }

  file.close();
  tempFile.close();

  SPIFFS.remove(CSV_FILE);
  SPIFFS.rename("/temp.csv", CSV_FILE);
}

// Contar linhas no arquivo CSV
int contarLinhas() {
  File file = SPIFFS.open(CSV_FILE, FILE_READ);
  if (!file) return 0;

  int linhas = 0;
  while (file.available()) {
    file.readStringUntil('\n');
    linhas++;
  }
  file.close();
  return linhas;
}

// Medir distância com HC-SR04
float measureDistance() {
  long duration;
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH, 30000); // Timeout 30ms
  if (duration == 0) return 9999;
  return (duration * 0.0343) / 2;
}

// Adicionar nova linha ao CSV (com controle de limite)
void appendToCSV(String dataLine) {
  int linhas = contarLinhas();
  if (linhas >= MAX_REGISTROS) {
    Serial.println("Arquivo cheio. Removendo a linha mais antiga...");
    removePrimeiraLinha();
  }

  File file = SPIFFS.open(CSV_FILE, FILE_APPEND);
  if (file) {
    file.println(dataLine);
    file.close();
```

```
    Serial.println("Registro adicionado: " + dataLine);
  } else {
    Serial.println("Erro ao abrir o arquivo para escrita.");
  }
}

// Página HTML com tabela e botão de download
void handleRoot() {
  String html = "<!DOCTYPE html><html><head><meta charset='UTF-8'>";
  html += "<meta http-equiv='refresh' content='5'>";
  html += "<title>Histórico de Medições</title></head><body>";
  html += "<h1>Histórico de Medições (Máx 500)</h1>";
  html += "<a href='/download'>📥 Baixar CSV</a><br><br>";
  html += "<table border='1'><tr><th>Data/Hora</th><th>Distância (cm)</th></tr>";

  File file = SPIFFS.open(CSV_FILE, FILE_READ);
  if (file) {
    while (file.available()) {
      String line = file.readStringUntil('\n');
      int sep = line.indexOf(',');
      if (sep > 0) {
        String dataHora = line.substring(0, sep);
        String distancia = line.substring(sep + 1);
        html += "<tr><td>" + dataHora + "</td><td>" + distancia + "</td></tr>";
      }
    }
    file.close();
  }

  html += "</table></body></html>";
  server.send(200, "text/html", html);
}

// Servir arquivo CSV para download
void handleDownload() {
  File file = SPIFFS.open(CSV_FILE, FILE_READ);
  if (!file) {
    server.send(500, "text/plain", "Arquivo não encontrado");
    return;
  }
  server.sendHeader("Content-Type", "text/csv");
  server.sendHeader("Content-Disposition", "attachment; filename=\"medicoes.csv\"");
  server.streamFile(file, "text/csv");
  file.close();
}

void setup() {
  Serial.begin(115200);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);

  if (!SPIFFS.begin(true)) {
    Serial.println("Erro ao montar SPIFFS");
    return;
  }

  WiFi.begin(ssid, password);
  Serial.print("Conectando à rede");
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("\nConectado! IP: " + WiFi.localIP().toString());
```

```
  // Configurar horário sem NTP
  setLocalTimeFromBuild();

  // Iniciar servidor
  server.on("/", handleRoot);
  server.on("/download", handleDownload);
  server.begin();
}

void loop() {
  float dist = measureDistance();

  if (dist < 400.0) {
    struct tm timeinfo;
    String dataHora = "N/A";

    if (getLocalTime(&timeinfo)) {
      char timeStringBuff[64];
      strftime(timeStringBuff, sizeof(timeStringBuff), "%d/%m/%Y %H:%M:%S", &timeinfo);
      dataHora = String(timeStringBuff);
    }

    String dataLine = dataHora + "," + String(dist, 2);
    appendToCSV(dataLine);

    delay(1000);
  }

  server.handleClient();
}
```

Explicações sobre o Buffer Circular:

Item	Explicação
Contagem de Linhas	Antes de adicionar um novo registro, conta quantas linhas o arquivo tem.
Remoção da Primeira Linha	Se já existirem 500 linhas, apaga a mais antiga (a primeira do arquivo).
Arquivo Temporário (temp.csv)	Cria um arquivo temporário sem a primeira linha e substitui o arquivo original.
Adição de Novos Registros	Após a remoção (se necessário), o novo registro é adicionado no final.

O que aprendemos nesta etapa:

1. Como implementar buffers circulares em arquivos.
2. Manipulação de arquivos SPIFFS (ler, apagar, renomear arquivos).
3. Conceito de *"Fila de registros fixos"*.
4. Sistemas de log de sensores com memória limitada.
5. Aplicação prática de gerenciamento de dados em dispositivos embarcados.

Bibliografia:

https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf
https://docs.espressif.com/projects/esp-idf/en/stable/esp32/index.html
https://randomnerdtutorials.com/
http://www.practical-arduino.com/