

SD Card Forensics on ESP32

RAS Minor Mini-Project - 1 PC408

Phase 1 : Design



**Dhirubhai Ambani
University**

Mentored by : Dr. Tapas Kumar Maiti

Jhil Patel

31.10.2025
SID : 202301090

INTRODUCTION

With the rapid growth of IoT devices, secure data logging has become critical, especially for applications like evidence collection, monitoring, and industrial automation. Logs stored on removable memory, such as microSD cards, are vulnerable to tampering by unauthorized parties. If manipulated, these logs can mislead forensic investigations and compromise system reliability.

This project focuses on designing a secure logging mechanism using an ESP32 development board and a microSD card. The system will first operate without security to demonstrate how an attacker can modify logs remotely. Later, security enhancements like hashing and tamper detection will be added to prevent and detect manipulation.

HYPOTHESIS

If sensor data stored on a microSD card is not protected, an attacker can alter the recorded information without being detected. Implementing cryptographic file hashing and verification routines will allow the system to detect such tampering, ensuring data authenticity and integrity even after attempted attacks.

MATERIALS

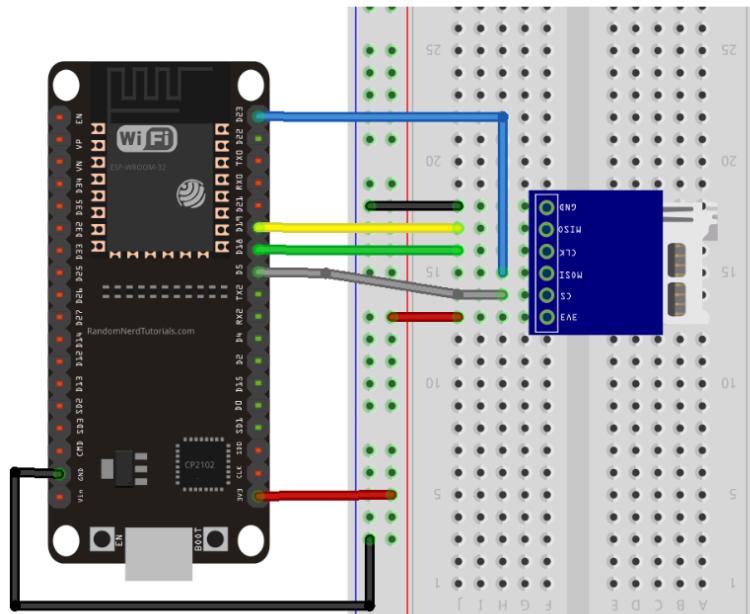
1. ESP32 Development Board
2. microSD Card Module & microSD Card (FAT32 formatted)
3. Jumper Wires & Breadboard
4. Wi-Fi-enabled smartphone (to simulate attacker device)
5. Arduino IDE / Wokwi for simulation
6. SHA-256 integrity checking library (for Phase 3)

PROCEDURE

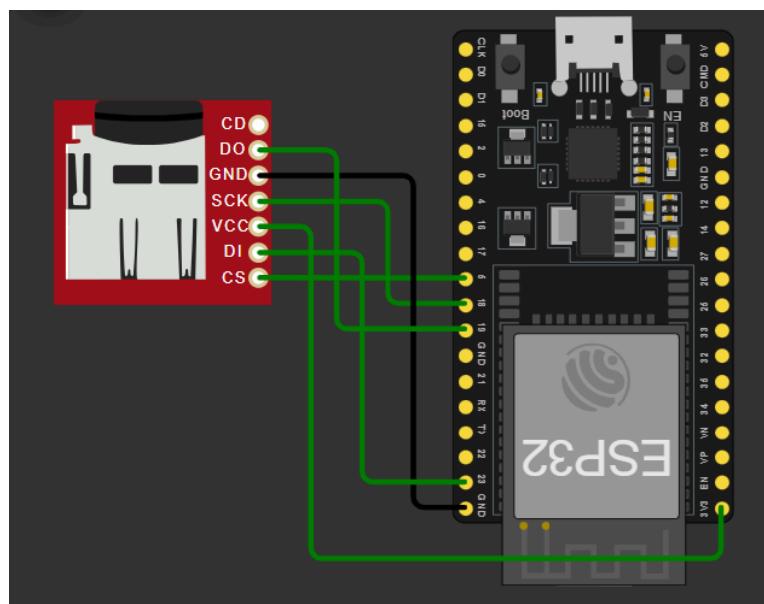
1. Connect ESP32 to microSD card module for log storage
2. Program ESP32 to log dummy sensor data to microSD
3. Enable Wi-Fi connectivity on ESP32
4. Allow external commands via network (HTTP/UDP) to modify logs
5. Simulate an attack using a smartphone to send malicious overwrite commands

6. Record behavior of system without protection (Phase-2 baseline)
 7. Plan implementation of SHA-256 hashing + verification to detect tampering

DESIGN REFERENCE

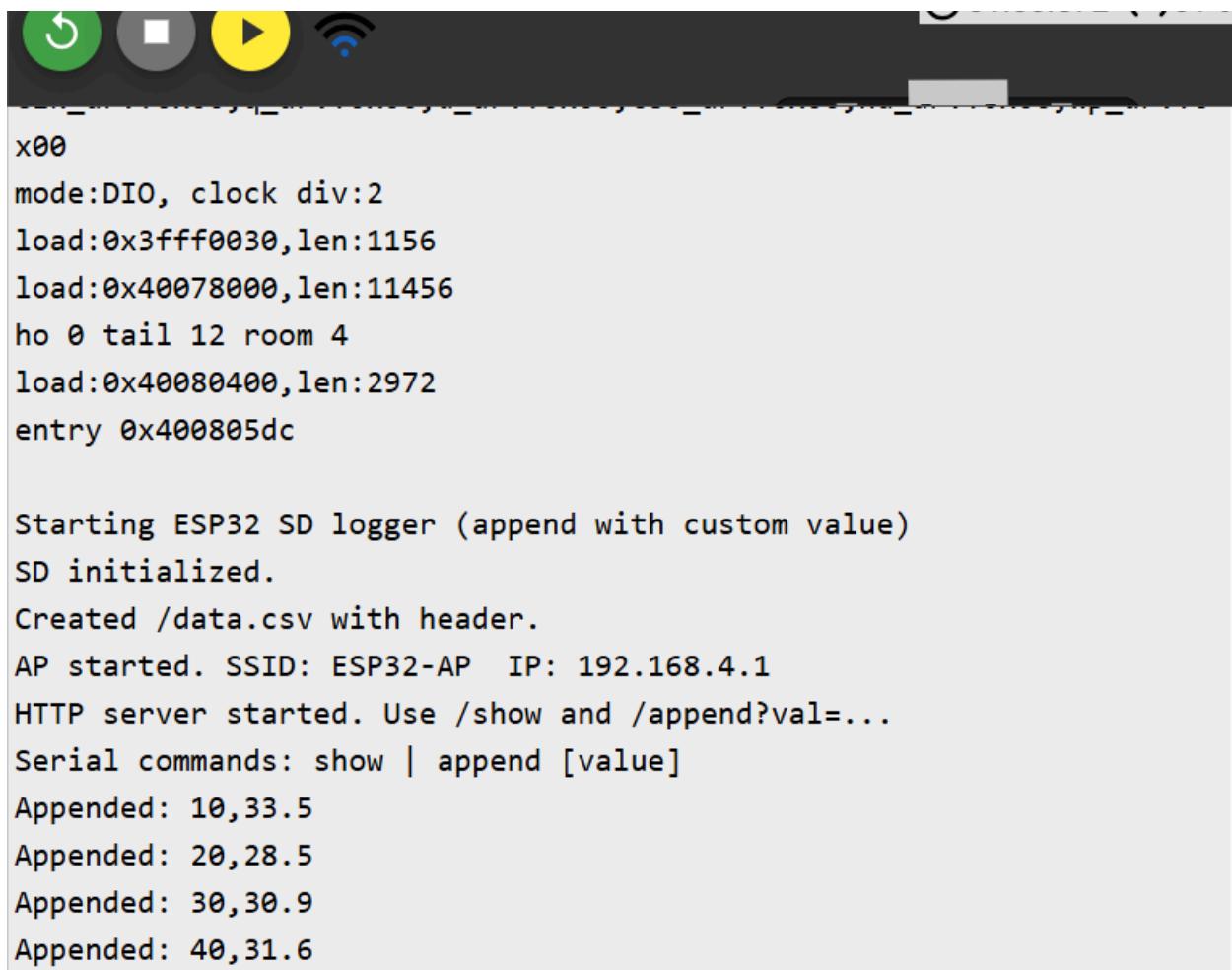


SIMULATION ON WOKWI



SIMULATION DRY RUN RESULTS

In normal flow the temperature sensor data is logged every 10 seconds for dry run



The screenshot shows a terminal window with a dark background and light-colored text. At the top, there are four icons: a green circle with a white arrow, a grey square, a yellow circle with a white play button, and a blue Wi-Fi signal icon. Below these icons, the terminal displays the following text:

```
x00
mode:DIO, clock div:2
load:0x3fff0030,len:1156
load:0x40078000,len:11456
ho 0 tail 12 room 4
load:0x40080400,len:2972
entry 0x400805dc

Starting ESP32 SD logger (append with custom value)
SD initialized.
Created /data.csv with header.
AP started. SSID: ESP32-AP IP: 192.168.4.1
HTTP server started. Use /show and /append?val=...
Serial commands: show | append [value]
Appended: 10,33.5
Appended: 20,28.5
Appended: 30,30.9
Appended: 40,31.6
```

Manually appending a value to simulate an external data injection, for example an out-of-range temperature of -20°C.

append -20 

Simulation

01:06.572 61%

```
x00
mode:DIO, clock div:2
load:0xffff0030,len:1156
load:0x40078000,len:11456
ho 0 tail 12 room 4
load:0x40080400,len:2972
entry 0x400805dc

Starting ESP32 SD logger (append with custom value)
SD initialized.
Created /data.csv with header.
AP started. SSID: ESP32-AP IP: 192.168.4.1
HTTP server started. Use /show and /append?val=...
Serial commands: show | append [value]
Appended: 10,33.5
Appended: 20,28.5
Appended: 30,30.9
Appended: 40,31.6
Appended: 50,21.6
Appended: 53,-20
Appended: 60,29.1
```

Displaying logged SD card data

show | ↻ | ⏸ | ⏹

```
---- /data.csv ----  
time_seconds,value  
10,33.5  
20,28.5  
30,30.9  
40,31.6  
50,21.6  
53,-20  
60,29.1
```

In the absence of a security layer, manual appends and legitimate data are treated identically by the system. As a result, malicious or accidental data injections may remain untracked, since there is no visible way to distinguish authentic records from tampered ones.

SIMULATION VIDEO LINK

https://youtu.be/qBASPebt6uw?si=N-i_R8BNAHHwSp1x

REFERENCES

1. <https://randomnerdtutorials.com/esp32-microsd-card-arduino/>
2. <https://wokwi.com/projects/new/esp32>
3. https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf