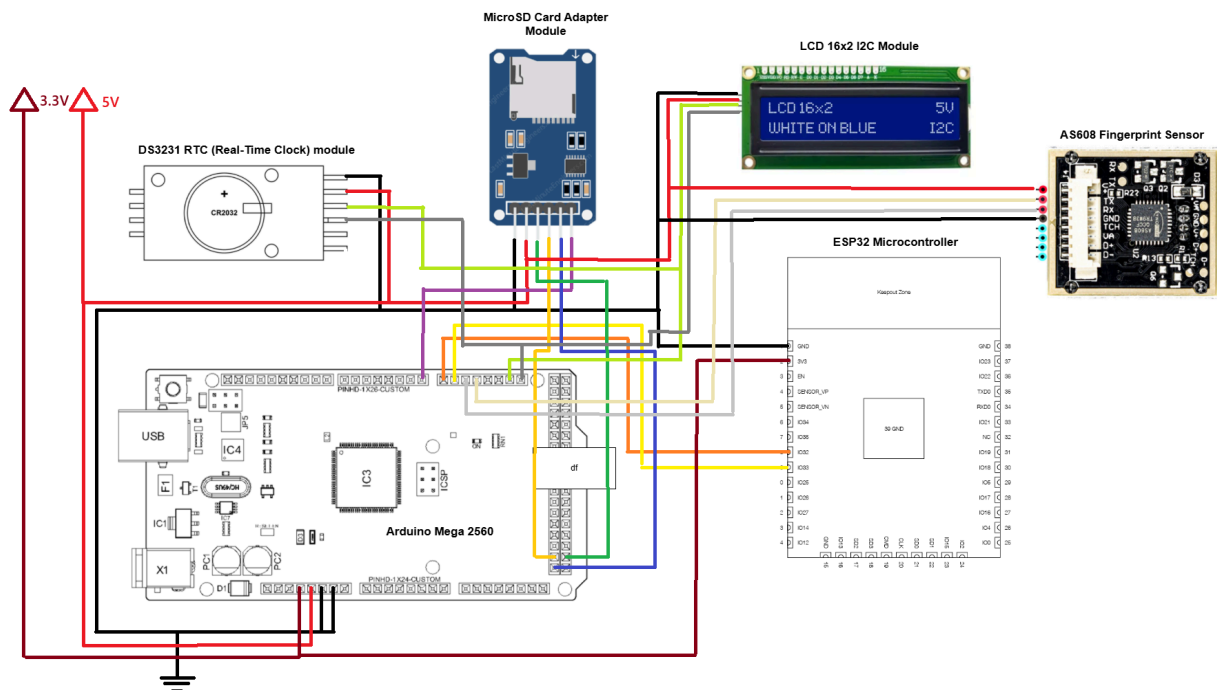


Fingerprint Biometrics System: User Manual



This manual provides step-by-step instructions to set up and operate your Fingerprint Biometrics System on a new computer. This system involves an Arduino Mega (with fingerprint sensor, RTC, LCD), an ESP32, and a PHP web dashboard running on your computer.

1. System Overview

Your system consists of:

- **Arduino Mega:** Connects to the fingerprint sensor, Real-Time Clock (RTC), and LCD. It manages fingerprint enrollment, deletion, and verification. It communicates with the ESP32 via serial.
- **ESP32:** Connects to your Wi-Fi network. It acts as a bridge, forwarding commands from the web dashboard to the Arduino Mega and sending fingerprint events/status from the Mega back to the web dashboard. It also hosts a small web server to receive commands from your dashboard.
- **Computer (PC):** Runs a local web server (XAMPP or Laragon) hosting a PHP web dashboard. This dashboard allows you to enroll/delete fingerprints, view attendance logs, and see the system's status.

2. Setting Up Your Computer (PC)

2.1. Install XAMPP or Laragon

You need a local web server to run the PHP dashboard. XAMPP is widely used, but Laragon is also an excellent alternative.

- **XAMPP:**

1. Download XAMPP from the official website:
<https://www.apachefriends.org/index.html>
2. Run the installer and follow the on-screen prompts. You can usually accept the default settings.
3. Once installed, open the **XAMPP Control Panel**.
4. Start the **Apache** module by clicking the "Start" button next to it. It should turn green.

- **Laragon:**

1. Download Laragon from the official website: <https://laragon.org/download/>
2. Run the installer and follow the on-screen prompts.
3. Once installed, open **Laragon**.
4. Click "Start All" to start the Apache and MySQL services.

2.2. Deploy the Web Dashboard Files

1. Locate your web server's document root:
 - **XAMPP:** C:\xampp\htdocs\
 - **Laragon:** C:\laragon\www\
2. Create a new folder inside this document root named fingerprint_dashboard.
 - For XAMPP: C:\xampp\htdocs\fingerprint_dashboard\
 - For Laragon: C:\laragon\www\fingerprint_dashboard\
3. Place your index.php file, arduino_status.txt, attendance_log.csv, and esp32_inbox.log (if they exist, otherwise they will be created) inside this fingerprint_dashboard folder.

2.3. Configure PC's IP Address (for serverUrl in ESP32)

Your ESP32 needs to know your computer's IP address to send data to the web dashboard.

1. **Find your PC's IP Address:**

- Open **Command Prompt** (search for cmd in Windows Start Menu).
- Type ipconfig and press Enter.
- Look for the section titled "Wireless LAN adapter Wi-Fi" (if you're on Wi-Fi) or "Ethernet adapter Ethernet" (if you're on wired).

- Note down the **IPv4 Address**. It will look something like 192.168.68.112. This is your PC_IP_ADDRESS.

2. Update serverUrl in ESP32 Sketch:

- Open your **ESP32 Arduino sketch** (e.g., FingerBioESP-old.ino).
- Find the line:

```
const char* serverUrl =
"http://192.168.68.112/fingerprint_dashboard/index.php"; // Use your
computer's actual IP
```
- **Replace 192.168.68.112 with your PC_IP_ADDRESS** that you found in ipconfig.
- Save the ESP32 sketch.

3. Update esp32_ip in index.php (for dashboard commands):

- Open your index.php file in the fingerprint_dashboard folder.
- Find the line:

```
$esp32_ip = "192.168.68.110"; // This is the IP of your ESP32 module itself
```
- **You will update this 192.168.68.110 to your ESP32's actual IP address *after* you get the ESP32 connected to Wi-Fi (see Section 3.3).**
For now, leave it as is, or put a placeholder like 0.0.0.0.

2.4. Set File Permissions (Crucial for Data Logging)

Your web server needs permission to write to the attendance_log.csv, arduino_status.txt, and esp32_inbox.log files.

1. Identify Apache's User:

- Ensure Apache is running in your XAMPP/Laragon Control Panel.
- Open **Task Manager** (Ctrl+Shift+Esc).
- Go to the **"Details"** tab.
- Look for httpd.exe. Note the **"User name"** associated with it (e.g., Janno, SYSTEM, NETWORK SERVICE). This is the user that needs write permissions.

2. Grant Permissions to the fingerprint_dashboard folder:

- Navigate to your fingerprint_dashboard folder (e.g., C:\xampp\htdocs\fingerprint_dashboard\).
- **Right-click** on the fingerprint_dashboard folder and select **Properties**.
- Go to the **"Security"** tab.
- Click **"Edit..."**.
- Click **"Add..."**.
- In the "Enter the object names to select" field, type the "User name" you found in Task Manager (e.g., Janno, IUSR, NETWORK SERVICE). Click **"Check**

Names" then **"OK"**.

- With the newly added user/group selected, in the "Permissions for [User/Group]" section, **check the "Allow" box for "Modify" and "Write"** (or "Full control" for simpler testing, but remember to revert for production).
 - Click **"Apply"** then **"OK"** on all windows.
3. **Restart Apache** from your XAMPP/Laragon Control Panel to apply the new permissions.

3. Setting Up Your Arduino IDE and Hardware

3.1. Install Arduino IDE

1. Download the Arduino IDE from <https://www.arduino.cc/en/software>. Choose the installer for your operating system.
2. Run the installer and follow the prompts.

3.2. Install Board Support Packages

You need to add support for ESP32 and Arduino Mega boards.

1. **Open Arduino IDE.**
2. Go to **File > Preferences.**
3. In "Additional Boards Manager URLs", add these URLs (one per line, or separated by commas):
 - For ESP32:
https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json
 - For Arduino AVR Boards (Mega):
https://raw.githubusercontent.com/arduino/arduino-board-index/main/package_arduino_index.json (This is usually included by default, but it's good to ensure).
4. Go to **Tools > Board > Boards Manager...**
5. Search for esp32 and install "esp32 by Espressif Systems".
6. Search for arduino avr and install "Arduino AVR Boards by Arduino".
7. Close the Boards Manager.

3.3. Install Required Libraries

1. Go to **Sketch > Include Library > Manage Libraries...**
2. Install the following libraries:
 - Adafruit Fingerprint Sensor Library by Adafruit
 - RTCLib by Adafruit
 - SD (usually built-in, but check if needed)

- LiquidCrystal I2C by Frank de Brabander (or a similar one if you have issues, e.g., by fdebrabander)
- WiFi (built-in for ESP32)
- HTTPClient (built-in for ESP32)
- WebServer (built-in for ESP32)

3.4. Upload Arduino Mega Sketch

1. Open your **Arduino Mega sketch** (FingerBio-old.ino).
2. **Connect your Arduino Mega to your PC** via USB.
3. Go to **Tools > Board > Arduino AVR Boards > Arduino Mega or Mega 2560**.
4. Go to **Tools > Port** and select the COM port for your Arduino Mega.
5. **Verify the LCD_ADDR:** If your LCD doesn't work, you might need to run an I2C scanner sketch (search online for "Arduino I2C scanner") to find your exact LCD address and update `const int LCD_ADDR = 0x27;` in your sketch.
6. **Set RTC Time (if needed):**
 - In your Mega sketch's `setup()` function, find the `rtc.adjust(...)` line.
 - If you want to set the RTC to the current time, uncomment (remove `//`) this line:

```
// rtc.adjust(DateTime(F(__DATE__), F(__TIME__))); // Sets to compile time
```

Or, to set a specific time (e.g., May 29, 2025, 3:05:00 AM):

```
// rtc.adjust(DateTime(2025, 5, 29, 3, 5, 0));
```

 - **Upload the sketch once with this line uncommented.** Let it run for a few seconds.
 - **Then, comment out the `rtc.adjust()` line again and re-upload.** This prevents the RTC from resetting every time the Mega powers on.
7. Click the **"Upload"** button (right arrow icon).
8. Open the **Serial Monitor** (magnifying glass icon) for the Mega (set baud rate to 57600). You should see initialization messages. Pay attention to "Fingerprint sensor found!" and "Sent MEGA_READY signal to ESP32."

3.5. Upload ESP32 Sketch

1. Open your **ESP32 Arduino sketch** (FingerBioESP-old.ino).
2. **Connect your ESP32 to your PC** via USB.
3. Go to **Tools > Board > ESP32 Arduino > ESP32 Dev Module**.
4. Go to **Tools > Port** and select the COM port for your ESP32.
5. **Verify Wi-Fi Credentials:** Ensure ssid and password match your home Wi-Fi.

6. **Verify serverUrl:** Double-check that serverUrl is set to your PC's IP address (from Section 2.3).
7. Click the **"Upload"** button.
8. Open the **Serial Monitor** for the ESP32 (set baud rate to 115200).
9. **Note the ESP32's IP Address:** Once connected to Wi-Fi, the ESP32 Serial Monitor will print "IP Address: 192.168.68.110" (or similar). **This is your ESP32_IP_ADDRESS.**
10. **Update index.php with ESP32 IP:**
 - Go back to your index.php file on your PC.
 - Update the line \$esp32_ip = "192.168.68.110"; to use your ESP32_IP_ADDRESS that you just found.
 - Save index.php.

3.6. Wire Arduino Mega and ESP32

Connect the two boards using jumper wires for serial communication:

- **ESP32 GPIO16 (RX2) to Arduino Mega Pin 15 (TX3)**
- **ESP32 GPIO17 (TX2) to Arduino Mega Pin 14 (RX3)**
- **Connect GND of ESP32 to GND of Arduino Mega.** (Essential for common reference voltage).

4. How the System Works

4.1. Initial Setup and Handshake

1. When both Arduino Mega and ESP32 power on, they connect to their respective components (fingerprint sensor, LCD, Wi-Fi).
2. The Arduino Mega sends a "MEGA_READY" signal to the ESP32 via serial.
3. The ESP32 waits for this signal to confirm the Mega is ready to receive commands.

4.2. Web Dashboard Operation (index.php)

Open your web browser and go to:

http://your_pc_ip_address/fingerprint_dashboard/index.php (e.g.,
http://192.168.68.112/fingerprint_dashboard/index.php).

- **Arduino Status:** Displays the current status messages from the Arduino Mega (e.g., "Ready to enroll ID X", "Finger not found"). This updates automatically.
- **Enroll Fingerprint:**
 1. Enter an ID (1-254) in the "Enter ID" field.
 2. Click "Enroll Finger".
 3. The dashboard sends this command to the ESP32, which forwards it to the

Arduino Mega.

4. Follow the prompts on the Arduino Mega's LCD (e.g., "Place finger", "Remove finger", "Place same finger again").
5. The Mega will send success/failure messages back to the ESP32, which updates the dashboard status.

- **Delete Fingerprint:**

1. Enter the ID of the fingerprint you wish to delete.
2. Click "Delete Finger".
3. The command is sent to the Mega, which performs the deletion and reports back.

- **Recent Attendance Log:**

- This table displays fingerprint scans (ID and Timestamp) that have been recorded.
- Click "Refresh Attendance" to manually update the list.
- Click "Clear Attendance Log" to erase all entries from the attendance_log.csv file.

- **How Attendance is Logged:**

1. When a recognized finger is scanned on the sensor (while in attendance mode), the Arduino Mega gets the fingerprint ID and the current time from its RTC.
2. The Mega sends this ID,TIMESTAMP data to the ESP32 via serial.
3. The ESP32 receives this data, formats it as id=X×tamp=Y, and sends it as a POST request to your index.php script on your PC.
4. The index.php script receives this POST request, parses the id and timestamp, and appends it as a new line to the attendance_log.csv file.
5. The web dashboard automatically refreshes to display the new entry.

5. Troubleshooting Common Issues

- **"Error on HTTP request: 0" (ESP32 Serial Monitor):**

- **Cause:** ESP32 cannot reach your PC's web server.
- **Fix:**
 - Confirm your PC's IP address (ipconfig) and update serverUrl in the ESP32 sketch if it changed.
 - Ensure Apache (XAMPP/Laragon) is running on your PC.
 - **Crucially, check your PC's firewall.** Temporarily disable it for testing, then add an exception for Apache (httpd.exe) to allow incoming connections on port 80.
 - Ensure both ESP32 and PC are on the same Wi-Fi network.

- **"Failed to write attendance to attendance_log.csv. Check permissions."**

(Apache error.log):

- **Cause:** The web server (Apache) does not have permission to write to the attendance_log.csv file or its directory.
- **Fix:** Follow **Section 2.4: Set File Permissions** carefully, ensuring the correct user (found in Task Manager for httpd.exe) has "Modify" and "Write" permissions on the fingerprint_dashboard folder.
- **Timestamp in CSV/Dashboard shows "%Y-%m-%d %H:%M:%S":**
 - **Cause:** The Arduino Mega's RTC_DS3231 library's toString() method is not formatting the DateTime object correctly into a string.
 - **Fix:** Ensure you have implemented the sprintf method in your Arduino Mega sketch as described in the last communication to correctly format the timestamp into a char array before sending it. Also, ensure your RTC has been set to the correct time (see Section 3.4, step 6).
- **"Error: Did not receive MEGA_READY from Arduino within timeout." (ESP32 Serial Monitor):**
 - **Cause:** Serial communication between Arduino Mega and ESP32 is not working.
 - **Fix:**
 - **Wiring:** Double-check the serial connections (ESP32 RX2-GPIO16 to Mega TX3-Pin 15, ESP32 TX2-GPIO17 to Mega RX3-Pin 14, and common GND).
 - **Baud Rates:** Ensure SerialArduino.begin(38400, SERIAL_8N1, 16, 17); on ESP32 and Serial3.begin(38400); on Mega have matching baud rates (38400).
- **LCD is blank or shows gibberish:**
 - **Cause:** Incorrect I2C address, faulty wiring, or LCD not initializing.
 - **Fix:**
 - Run an I2C scanner sketch on the Mega to find the exact LCD address and update LCD_ADDR in your Mega sketch.
 - Double-check all LCD wiring (SDA, SCL, VCC, GND).
- **Fingerprint sensor not found:**
 - **Cause:** Incorrect wiring, power, or baud rate for the sensor.
 - **Fix:**
 - Check sensor wiring to Mega (RX to Pin 14, TX to Pin 15, VCC, GND).
 - Ensure finger.begin(57600); matches your sensor's baud rate (try 9600 if 57600 doesn't work).