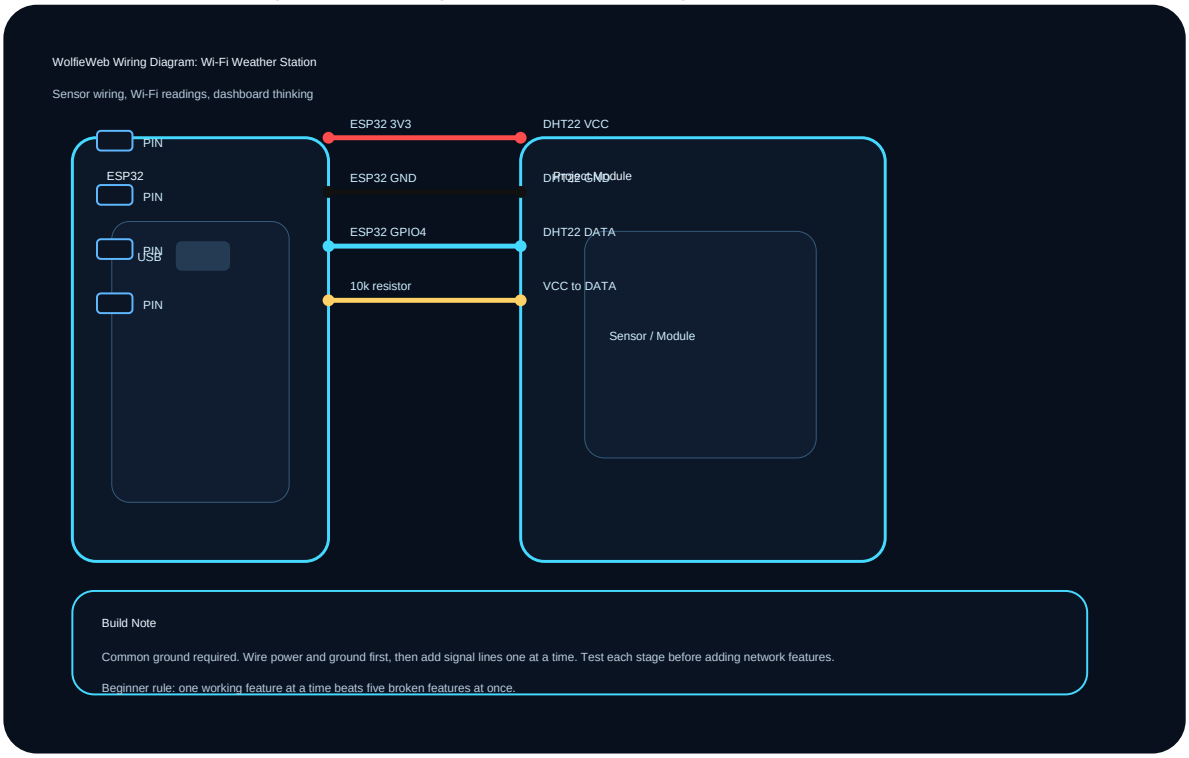
WolfieWeb IoT Lab Manual

Printable wiring and build guide for works02.html. Use this as the bench reference while building ESP32 smart-device projects.

Wi-Fi Weather Station

Skill focus: Sensor wiring, Wi-Fi readings, dashboard thinking



Parts Needed

- ESP32 Dev Board
- DHT22 or DHT11 sensor
- 10k ohm resistor
- Breadboard
- Male-to-male jumper wires
- USB cable

Pin Mapping

Board / Source	Connects To	Why It Matters
ESP32 3V3	DHT22 VCC	Power the sensor from 3.3V, not 5V.
ESP32 GND	DHT22 GND	Common ground is required.
ESP32 GPIO4	DHT22 DATA	Digital data signal.
10k resistor	VCC to DATA	Pull-up keeps data stable.

Step-by-step Build

1. Place the DHT22 sensor across the breadboard gap so each pin has its own row.
2. Connect VCC to ESP32 3V3 and GND to ESP32 GND before adding the data line.
3. Connect DATA to GPIO4, then place the 10k resistor between VCC and DATA.
4. Upload a basic DHT test sketch and confirm serial readings before adding Wi-Fi or dashboard code.
5. Only after the readings are stable, add the network publishing or dashboard portion.

Common Mistakes

- Using 5V on the sensor when your board expects 3.3V logic.
- Forgetting the 10k pull-up resistor on DATA.
- Reading from the wrong GPIO number in code.

Upgrade Ideas

- Send readings to MQTT every 60 seconds.
- Add a small OLED display.
- Log data to a web dashboard.

Smart Relay Controller

Skill focus: Safe switching, GPIO output control, automation basics

Parts Needed

- ESP32 Dev Board
- 1-channel 3.3V-compatible relay module
- Low-voltage LED strip or small DC load
- External low-voltage power supply
- Breadboard/jumpers
- USB cable

Pin Mapping

Board / Source	Connects To	Why It Matters
ESP32 3V3 or VIN	Relay VCC	Match the relay module requirements.
ESP32 GND	Relay GND	Common ground for control side.
ESP32 GPIO5	Relay IN	GPIO toggles relay state.
Load +	Relay COM/NO	Use low voltage while learning.

Step-by-step Build

1. Start with a low-voltage DC load only; do not begin with wall voltage.
2. Wire relay VCC and GND to the control side first.
3. Connect relay IN to GPIO5 and test whether the relay clicks with a simple blink-style sketch.
4. Wire the low-voltage load through COM and NO so it only powers when the relay closes.
5. Test short ON/OFF cycles and confirm the load switches cleanly.

Common Mistakes

- Working with AC mains before understanding relay wiring.
- Buying a 5V relay that does not trigger reliably from 3.3V GPIO.

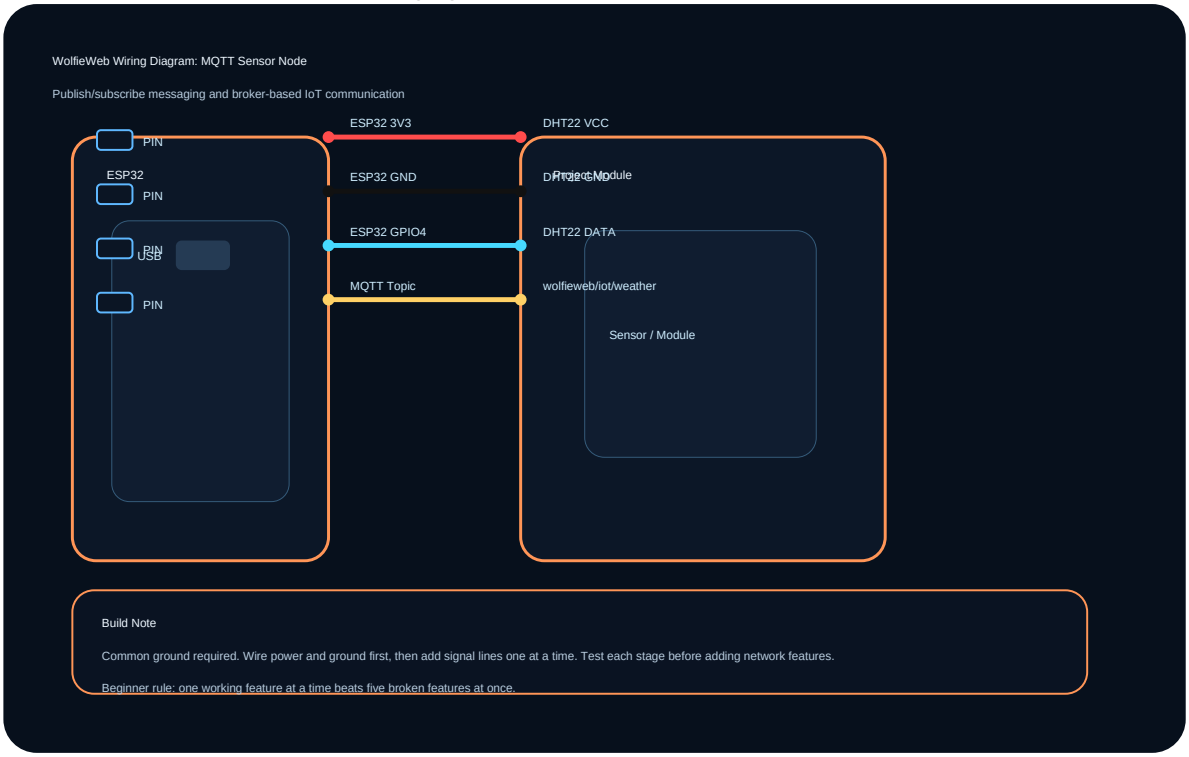
• Mixing load power and ESP32 power incorrectly.

Upgrade Ideas

• Add a web button for ON/OFF control.
• Add a physical override button.
• Add MQTT commands such as home/relay/set.

MQTT Sensor Node

Skill focus: Publish/subscribe messaging and broker-based IoT communication



Parts Needed

- ESP32 Dev Board
- DHT22 sensor
- 10k ohm resistor
- Breadboard/jumpers
- MQTT broker such as Mosquitto
- Wi-Fi network

Pin Mapping

Board / Source	Connects To	Why It Matters
ESP32 3V3	DHT22 VCC	Sensor power.
ESP32 GND	DHT22 GND	Common ground.
ESP32 GPIO4	DHT22 DATA	Sensor input.
MQTT Topic	wolfieweb/iot/weather	Published readings go here.

Step-by-step Build

1. Build and test the sensor wiring first, exactly like the weather station project.
2. Install or choose an MQTT broker and confirm the broker IP address.
3. Flash code that connects to Wi-Fi, then connects to the MQTT broker.
4. Publish test messages before publishing sensor values.
5. Subscribe from a dashboard or MQTT client and verify the data appears.

Common Mistakes

- Wrong broker IP address or port.
- Using the wrong topic name between publisher and subscriber.

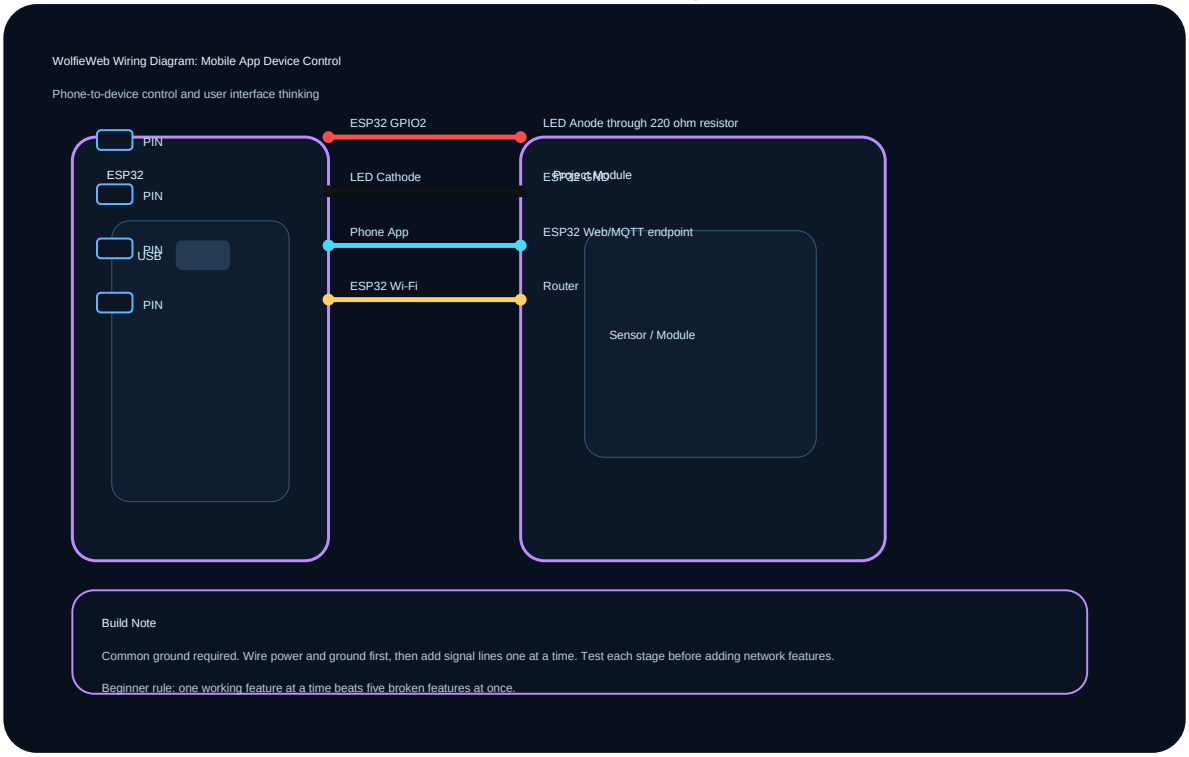
• Publishing too fast and flooding the broker.

Upgrade Ideas

• Add JSON payloads for temperature and humidity.
• Add retained status messages.
• Use Home Assistant or Node-RED as a dashboard.

Mobile App Device Control

Skill focus: Phone-to-device control and user interface thinking



Parts Needed

- ESP32 Dev Board
- LED or relay test output
- 220 ohm resistor for LED
- Breadboard/jumpers
- Mobile control app or simple web dashboard
- Wi-Fi network

Pin Mapping

Board / Source	Connects To	Why It Matters
ESP32 GPIO2	LED Anode through 220 ohm resistor	Test output.
LED Cathode	ESP32 GND	Completes circuit.
Phone App	ESP32 Web/MQTT endpoint	Sends command.
ESP32 Wi-Fi	Router	Network path.

Step-by-step Build

1. Wire a simple LED output first so testing is safe and obvious.
2. Upload code that turns the LED on and off from a local command.
3. Connect the ESP32 to your Wi-Fi network and print the IP address.
4. Use your phone on the same Wi-Fi network to open the control screen or send commands.
5. After the LED works, replace it with a relay or larger controlled output.

Common Mistakes

- Phone and ESP32 on different networks.
- Router blocking local device discovery.

• Trying relay control before proving simple LED control.

Upgrade Ideas

• Add a better mobile UI with large buttons.
• Add state feedback so the app shows ON/OFF correctly.
• Add password protection before exposing anything outside the home network.

Connected Security Camera

Skill focus: Camera streaming, monitoring, and connected alerts

Parts Needed

- ESP32-CAM or Raspberry Pi camera setup
- FTDI programmer for ESP32-CAM
- PIR motion sensor optional
- Stable 5V power supply
- Jumper wires
- MicroSD card optional

Pin Mapping

Board / Source	Connects To	Why It Matters
5V Supply	ESP32-CAM 5V	Camera needs stable power.
GND	ESP32-CAM GND	Common ground.
FTDI TX/RX	ESP32-CAM U0R/U0T	Used for programming.
GPIO0 to GND	Flash mode only	Remove after flashing.

Step-by-step Build

1. Use a stable 5V supply; camera boards brown out easily with weak USB power.
2. Wire FTDI programmer TX/RX crossed to the ESP32-CAM serial pins.
3. Hold GPIO0 to GND only while flashing firmware.
4. After flashing, remove GPIO0 from GND and reboot the board.
5. Open the printed camera URL from a browser on the same network.

Common Mistakes

- Leaving GPIO0 grounded after flashing.
- Weak power supply causing restarts.

• Expecting camera stream to work outside the home network without router/security setup.

Upgrade Ideas

• Add PIR-triggered snapshots.
• Save images to microSD.
• Send motion alerts to a dashboard or phone.