

ideaspark NodeMCU ESP8266

ideaspark NodeMCU ESP8266 Development Board with 0.96 Inch OLED Display User Manual

Model: NodeMCU ESP8266

1. INTRODUCTION

This manual provides comprehensive instructions for the setup, operation, and maintenance of your ideaspark NodeMCU ESP8266 Development Board with a 0.96 Inch OLED Display. This board integrates the ESP8266 WiFi module with a 128x64 SSD1306 OLED display, offering a compact solution for Internet of Things (IoT) projects, network monitoring, and data visualization.

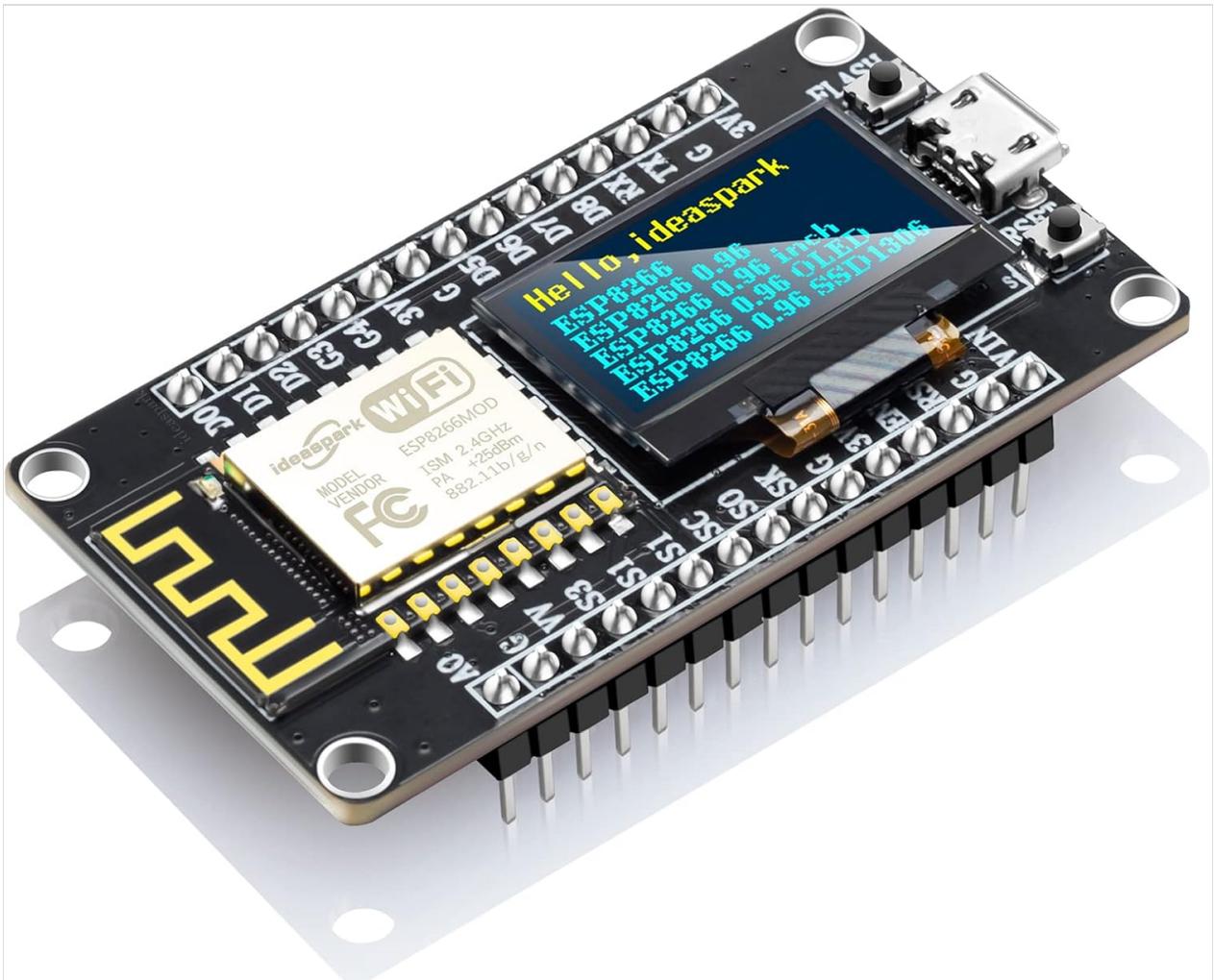


Figure 1: The ideaspark NodeMCU ESP8266 Development Board with integrated 0.96 inch OLED display.

2. PRODUCT OVERVIEW

The ideaspark NodeMCU ESP8266 Development Board combines the powerful ESP-12E WiFi module with a convenient 0.96-inch OLED display. This integration simplifies development by eliminating the need for external wiring for the display. Key features include:

- **ESP-12E Module:** Features a Tensilica Xtensa 32-bit LX106 RISC microprocessor, 128 KB RAM, and 4MB Flash memory.
- **Integrated 0.96-inch OLED Display:** 128x64 resolution with SSD1306 driver, compatible with I2C and SPI interfaces.
- **CH340 Driver:** For USB-to-serial communication.
- **Micro USB Port:** For power and data communication.
- **Self-Recovery Fuse:** Protects against overcurrent and short circuits.

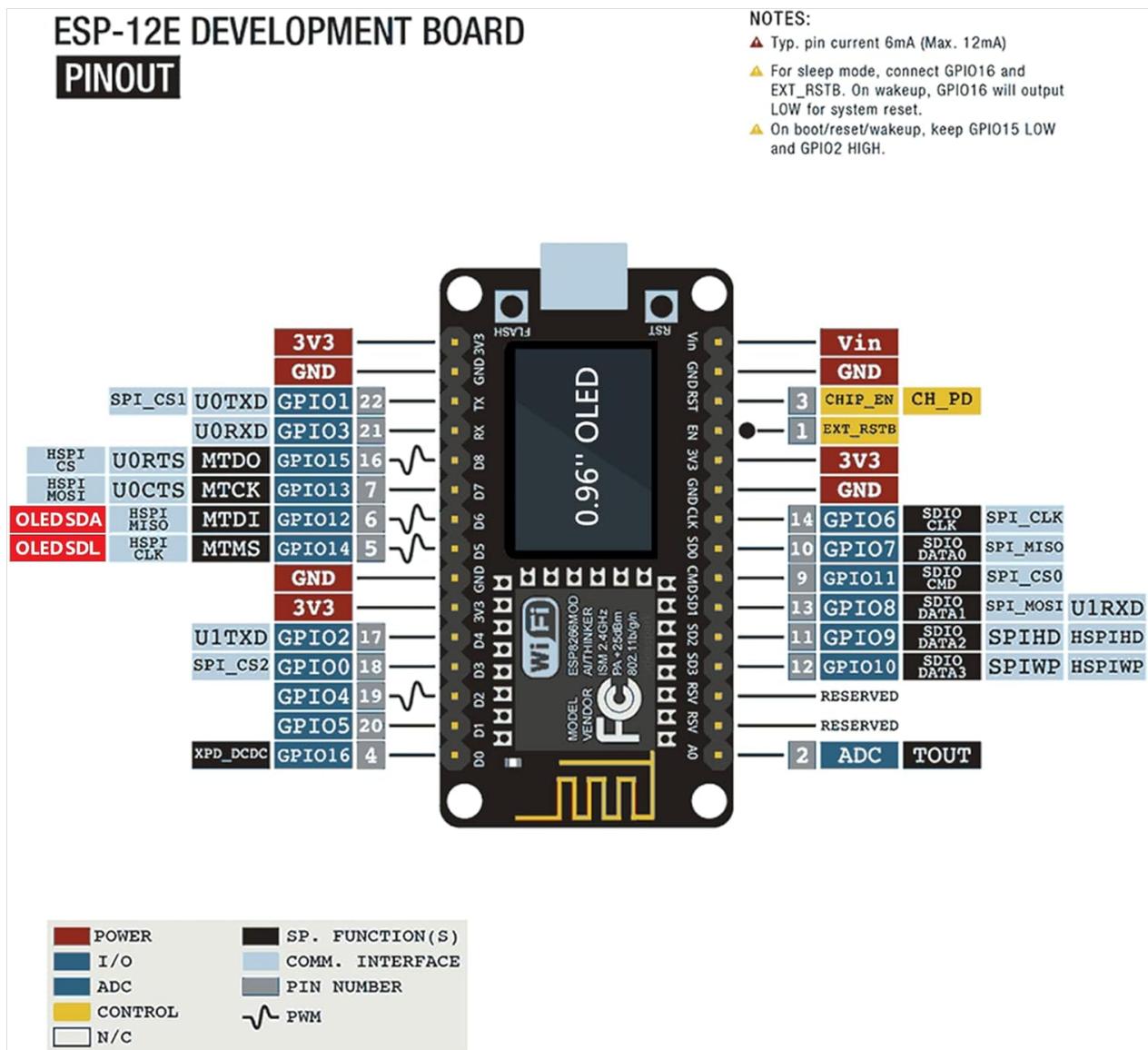


Figure 2: Pinout diagram for the ESP-12E Development Board, detailing GPIO assignments and power connections.

3. SETUP GUIDE

Follow these steps to prepare your NodeMCU ESP8266 board for programming using the Arduino IDE.

3.1. CH340 Driver Installation

The board uses a CH340 chip for USB-to-serial communication. You may need to install the appropriate driver for your operating system.

1. Search for "CH340 Driver" on a search engine (e.g., Google.com or Bing.com).
2. Download the latest driver package compatible with your operating system (Windows 10 recommended).
3. Follow the provided installation instructions.

3.2. Arduino IDE Setup for ESP8266

To program the ESP8266 board using the Arduino IDE, you need to add the ESP8266 board package.

1. Open the Arduino IDE.
2. Go to **File > Preferences**.
3. In the "Additional Boards Manager URLs" field, add the following URL:
`http://arduino.esp8266.com/stable/package_esp8266com_index.json`
4. Click **OK**.
5. Go to **Tools > Board > Boards Manager...**
6. Search for "esp8266" and install the "ESP8266 by ESP8266 Community" package.
7. After installation, go to **Tools > Board** and select **NodeMCU 1.0 (ESP-12E Module)**.



Figure 3: Steps for configuring the Arduino IDE to support the ESP8266 board, including adding the board manager URL and selecting the board type.

3.3. OLED Display Library Installation (U8g2)

To control the integrated OLED display, the U8g2 library is recommended.

1. In the Arduino IDE, go to **Sketch > Include Library > Manage Libraries...**
2. Search for "U8g2" and install the "U8g2 by oliver" library (version 2.19.8 or newer).



Figure 4: Instructions for installing the U8g2 library and setting board parameters in the Arduino IDE.

3.4. Initial Code Upload

After setting up the IDE and libraries, you can upload example code to test the board and OLED display. A common initialization for the OLED using U8g2 is:

```
#include <Arduino.h>
#include <U8g2lib.h>
#include <Wire.h>

// For this board, the OLED uses I2C on D6 (SDA) and D5 (SCL)
// U8G2_SSD1306_128X64_NONAME_F_SW_I2C u8g2(U8G2_R0, /* clock=*/ 14, /* data=*/ 12, /* reset=*/ U8X8_PIN_NONE)
// Or for hardware I2C (GPIO14=SCL, GPIO12=SDA):
U8G2_SSD1306_128X64_NONAME_F_HW_I2C u8g2(U8G2_R0, /* reset=*/ U8X8_PIN_NONE, /* clock=*/ 14, /* data=*/ 12);

void setup() {
  Serial.begin(115200);
  u8g2.begin();
  u8g2.setFont(u8g2_font_7x13_tf);
  u8g2.clearBuffer();
  u8g2.drawStr(0, 10, "Hello, ideaspark");
  u8g2.sendBuffer();
}

void loop() {
  // Your main code here
}
```

Ensure the correct COM port is selected under **Tools > Port** before uploading.

4. OPERATING INSTRUCTIONS

The NodeMCU ESP8266 board is designed for flexible operation in various IoT applications.

4.1. Powering the Board

Connect the board to a computer or a 5V USB power adapter using a Micro USB cable. The board will power on automatically.

4.2. Wi-Fi Connectivity

The ESP8266 module provides 802.11 b/g/n Wi-Fi connectivity. You can program it to connect to existing Wi-

Fi networks or create its own access point. Refer to ESP8266 Wi-Fi examples in the Arduino IDE for detailed programming.

4.3. OLED Display Usage

The 0.96-inch OLED display is connected via I2C. The SDA (Data) pin is D6 (GPIO12) and the SCL (Clock) pin is D5 (GPIO14). Use the U8g2 library functions to draw text, graphics, and display data on the screen.

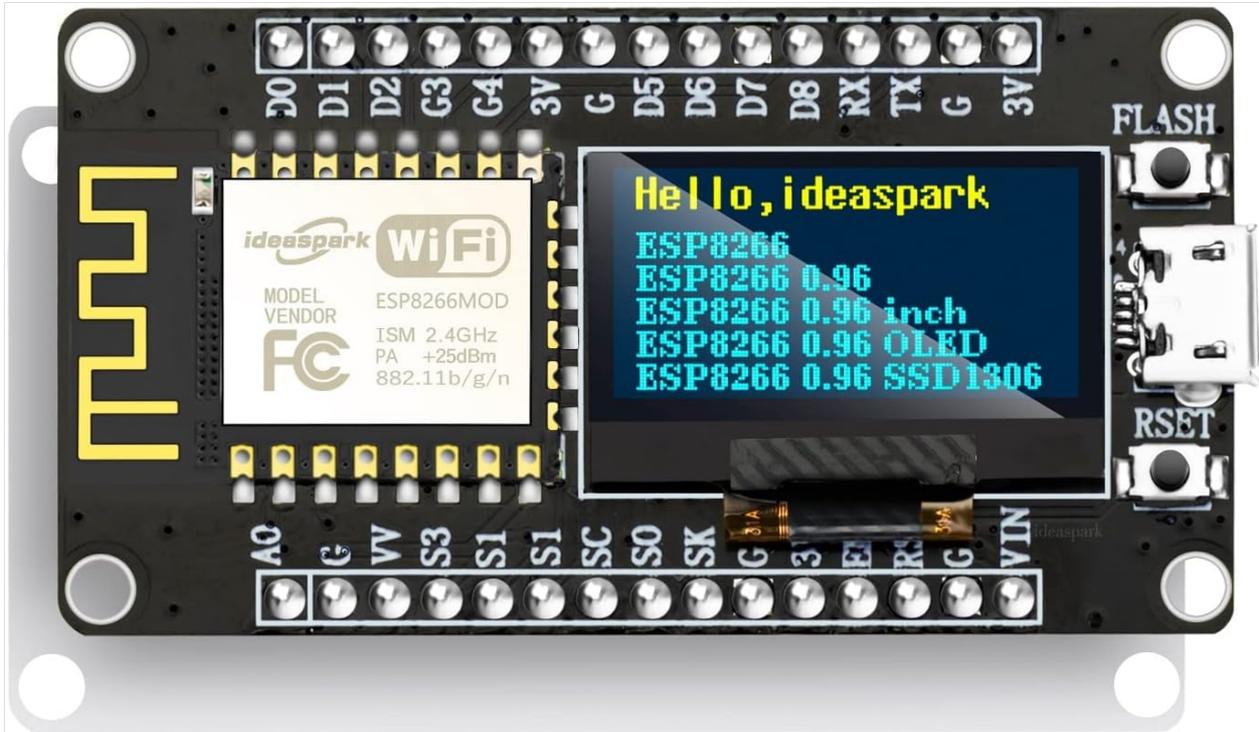


Figure 5: The integrated OLED display showing a sample message, demonstrating its functionality.

5. SPECIFICATIONS

Feature	Detail
Model Name	NodeMCU ESP8266
Processor	Tensilica Xtensa 32-bit LX106 RISC microprocessor
Operating Frequency	80MHz - 160MHz
RAM	128 KB
Flash Memory	4 MB
Wireless Type	Wi-Fi (802.11 b/g/n)
OLED Display	0.96 inch, 128x64 resolution, SSD1306 driver
OLED Interface	I2C (SDA: D6/GPIO12, SCL: D5/GPIO14)
USB-to-Serial Chip	CH340
Power Input	Micro USB, 5V
Dimensions (LxWxH)	2.24 x 1.22 x 0.51 inches

Feature	Detail
Item Weight	0.634 ounces

6. TROUBLESHOOTING

- **Board not recognized by computer:**

Ensure the CH340 driver is correctly installed. Try a different USB cable or port. Restart your computer.

- **Failed to upload code:**

Verify that "NodeMCU 1.0 (ESP-12E Module)" is selected under **Tools > Board** and the correct COM port is chosen under **Tools > Port**. Ensure the board is properly connected. Sometimes, pressing the "Flash" button on the board while uploading can help.

- **OLED display not working:**

Confirm the U8g2 library is installed. Check the I2C pin assignments in your code (SDA: D6/GPIO12, SCL: D5/GPIO14). Ensure the U8g2 constructor matches the hardware I2C configuration for this board. An example constructor is: `U8G2_SSD1306_128X64_NONAME_F_HW_I2C u8g2(U8G2_R0, /* reset= */ U8X8_PIN_NONE, /* clock= */ 14, /* data= */ 12);`

- **Wi-Fi connection issues:**

Double-check your Wi-Fi SSID and password in the code. Ensure the board is within range of the Wi-Fi access point. Verify that the ESP8266 Wi-Fi library is correctly included and used.

7. MAINTENANCE

- Keep the board in a dry, dust-free environment.
- Avoid exposing the board to extreme temperatures or direct sunlight.
- Handle the board by its edges to prevent damage to components or static discharge.
- Regularly check for firmware updates for the ESP8266 core in the Arduino IDE to benefit from bug fixes and new features.

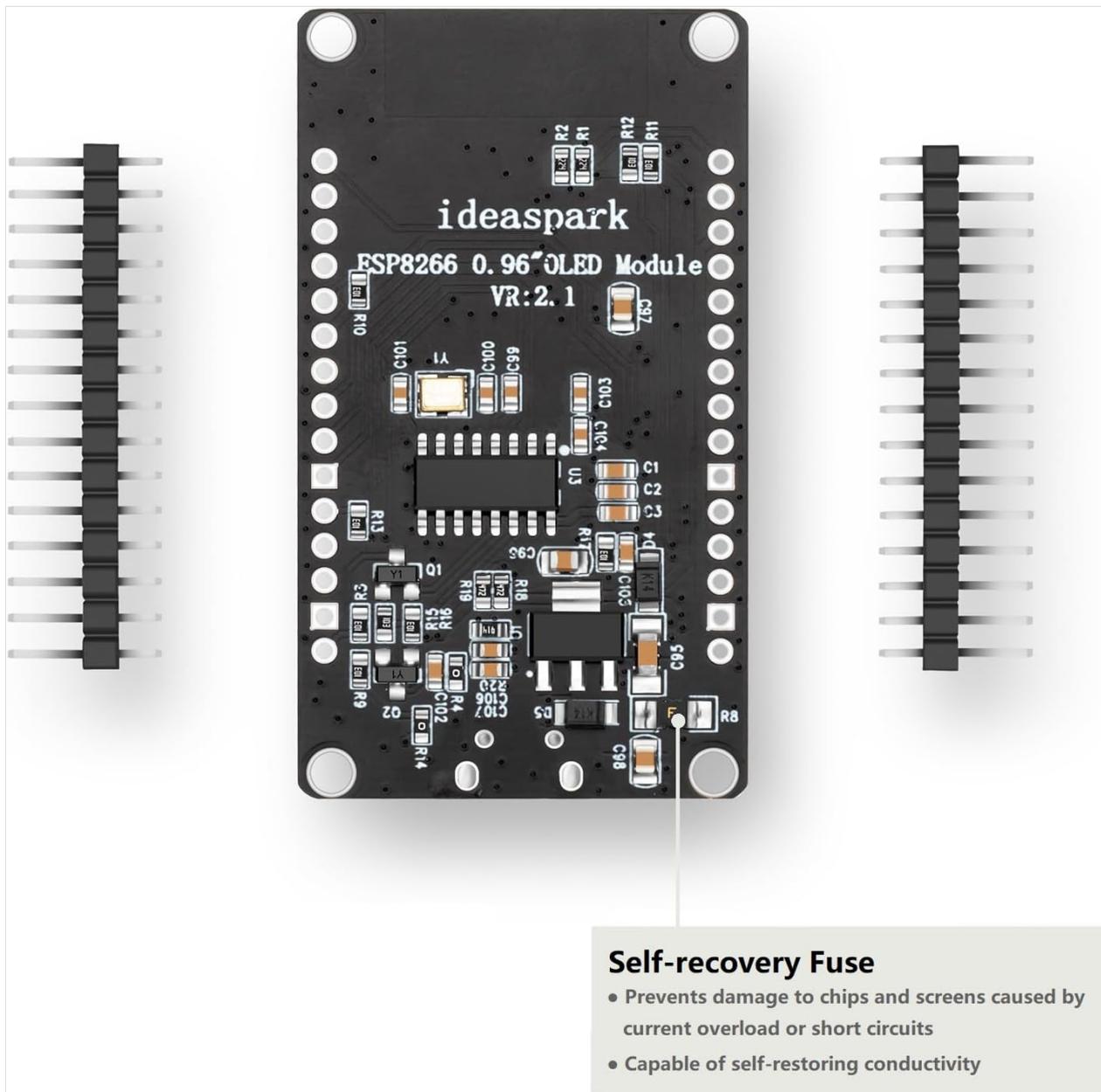


Figure 6: The self-recovery fuse on the board, designed to protect against electrical overloads.

8. WARRANTY AND SUPPORT

For specific warranty information, please refer to the product's purchase documentation or contact the retailer. For technical support, consult online communities and forums dedicated to ESP8266 and NodeMCU development, as they offer extensive resources and peer assistance.



[ESP8266 0.96 Inch OLED Board: Features, Specifications, and Setup Guide](#)

Explore the ESP8266 0.96 Inch OLED Board, a compact development board for IoT projects. This guide details its features, technical specifications, port mapping, and provides step-by-step instructions for setup using Arduino IDE.



[ESP8266 NodeMCU: Features and Arduino IDE Integration](#)

A comprehensive guide to the ESP8266 NodeMCU, detailing its features, specifications, pinouts, and how to use it with the Arduino IDE for IoT projects.



[ESP Prog v1.0 Usage Guide with Arduino IDE](#)

A comprehensive guide on using the ESP Prog v1.0 with the Arduino IDE, covering driver installation, board package setup, and basic programming examples for ESP8266 microcontrollers.



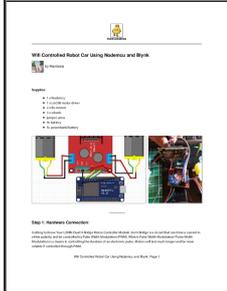
[ESP8266 and Raspberry Pi IoT Ecosystem Guide](#)

Explore the ESP8266/Raspberry Pi IoT ecosystem with practical examples. Learn to set up Arduino IDE, create web clients/servers, use MQTT with Home Assistant, and build RESTful APIs with Node.js for connected devices.

Introduction	1
Chapter 1: Getting Started	1
Chapter 2: Hardware	1
Chapter 3: Software	1
Chapter 4: Projects	1
Chapter 5: Advanced Topics	1
Chapter 6: Troubleshooting	1
Chapter 7: Conclusion	1
Appendix A: Pin List	1
Appendix B: Component List	1
Appendix C: Glossary	1
Appendix D: Index	1
Appendix E: License	1
Appendix F: Acknowledgments	1
Appendix G: About the Author	1
Appendix H: Contact Information	1
Appendix I: Revision History	1
Appendix J: Errata	1
Appendix K: Bibliography	1
Appendix L: References	1
Appendix M: Further Reading	1
Appendix N: Additional Resources	1
Appendix O: Community Support	1
Appendix P: Feedback	1
Appendix Q: Updates	1
Appendix R: Copyright	1
Appendix S: Disclaimer	1
Appendix T: Warranty	1
Appendix U: Terms and Conditions	1
Appendix V: Privacy Policy	1
Appendix W: Cookies Policy	1
Appendix X: Security Policy	1
Appendix Y: Sustainability Policy	1
Appendix Z: Other Policies	1

[ESP8266 Practical Guide: From Basics to Advanced Projects](#)

A comprehensive guide to the ESP8266 microcontroller, covering its introduction, modules, development boards, programming with Lua and Arduino IDE, and various practical projects including sensors, web servers, and robotics.



[Build a WiFi Controlled Robot Car with NodeMCU and Blynk](#)

A comprehensive guide to building a WiFi-controlled robot car using NodeMCU and Blynk. Learn about hardware connections, software setup with Arduino IDE, and control the robot remotely.