Embark on IoT

A Hands-On Workshop with NodeMCU ESP8266

Learn to build and program your own Internet of Things devices

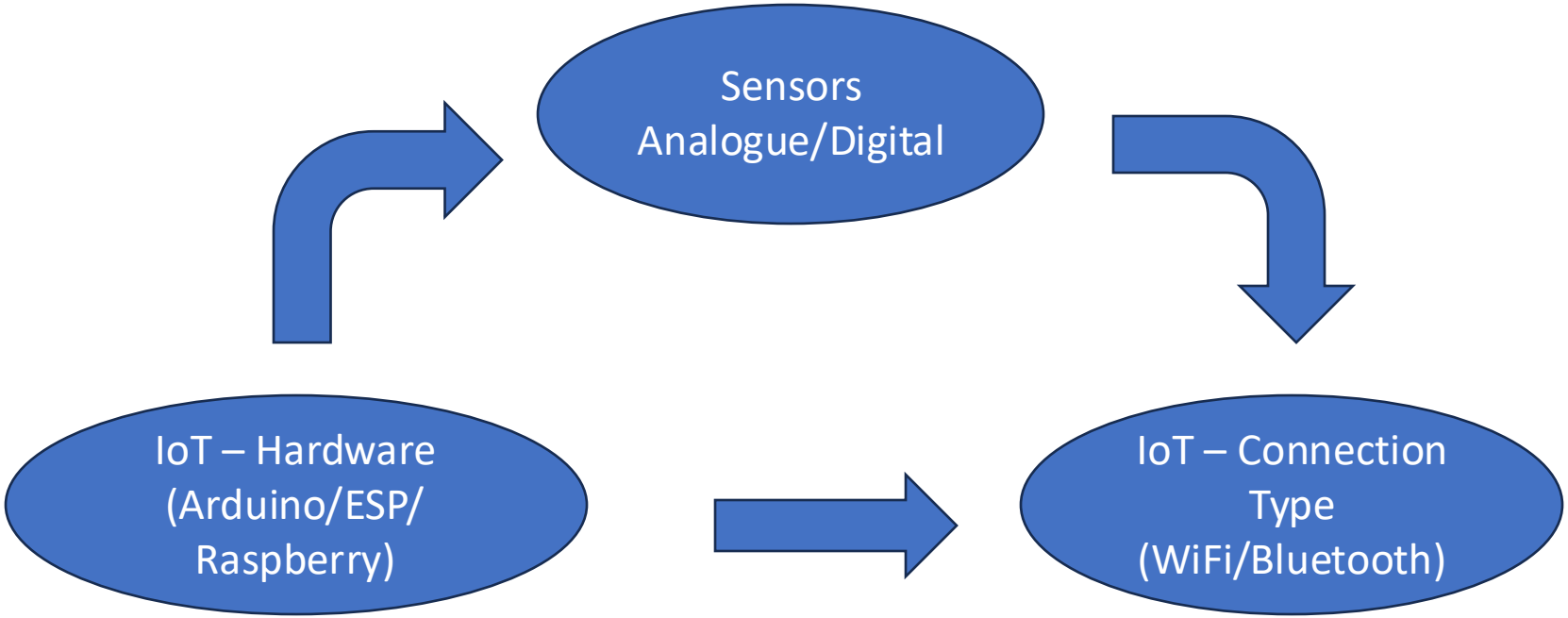
From basic circuits to cloud-connected projects

Beginner-friendly, hands-on approach

By

Dr. R. S. Rimal Isaac

rimalisaac@gmail.com


Sensors
Analogue/Digital
IoT – Hardware
(Arduino/ESP/
Raspberry)
IoT – Connection
Type
(WiFi/Bluetooth)

Introduction to ESP8266

What is ESP8266?

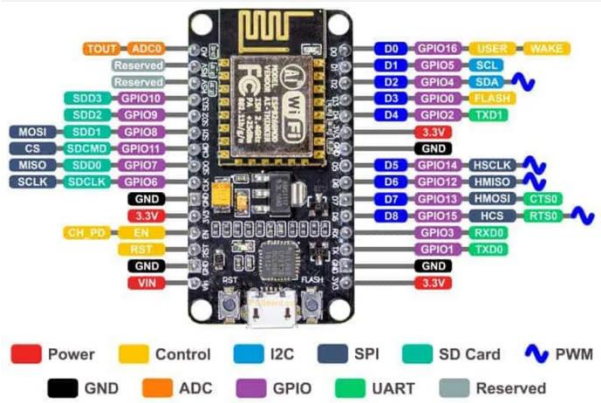
Low-cost Wi-Fi microchip with integrated TCP/IP
Built-in microcontroller capabilities
3.3V operating voltage, not 5V tolerant
17 GPIO pins for various functions
Programmable with Arduino IDE

Comparison with Arduino Uno

Built-in Wi-Fi connectivity
Higher processing power (80-160 MHz vs 16 MHz)
More memory (4MB Flash, 80KB RAM)
Lower power consumption
Smaller form factor

Applications in Physics

Remote data acquisition and logging
Wireless sensor networks for experiments
Real-time data visualization
Automated experiment control
Environmental monitoring for labs



Workshop Requirements

Knowledge Prerequisites

Basic Computer Literacy

Familiarity with using a personal computer, managing files, and browsing the internet

Programming Fundamentals (Recommended)

Basic understanding of programming concepts like variables, if-else conditions, and for loops is helpful but not mandatory

Hardware Requirements

A Laptop

With Windows, macOS, or Linux operating system

Administrative Rights

Ability to install software and drivers on your laptop

USB-A Port

To connect the NodeMCU board (USB-C adapter required if your laptop only has USB-C ports)

Software Requirements

Arduino IDE

Download and install the latest version from the official Arduino website

CH340 Driver

Required for Windows to detect the NodeMCU board's USB-to-Serial chip

ESP8266 Board Support

Add board support for ESP8266 in Arduino IDE

Required Libraries

Various libraries for sensors and displays will be installed during the workshop

Important Note:
All software requirements should be installed **BEFORE** attending the workshop to ensure a smooth experience. Detailed installation instructions will be covered in the following slides.

Arduino IDE Installation

Installation Steps

1. **Download Arduino IDE**
   Visit the official Arduino website at arduino.cc/en/software

2. **Choose Your Operating System**

   Select the appropriate version for your OS (Windows, macOS, or Linux)

3. **Download and Run the Installer**
   For Windows: Run the .exe file
   For macOS: Drag Arduino to Applications folder
   For Linux: Extract the archive and run install.sh

4. **Complete the Installation**

   Follow the on-screen instructions to complete the installation process

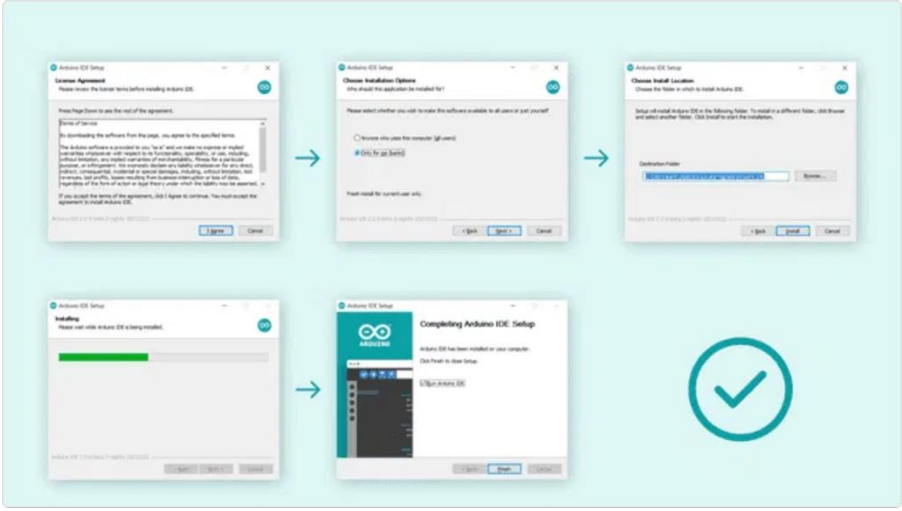
5. **Launch Arduino IDE**

   Open the Arduino IDE from your applications menu or desktop shortcut

Note:

Arduino IDE 2.x is recommended for this workshop as it offers improved features and performance compared to the 1.x version.

Arduino IDE Interface



Arduino IDE 2.0 Interface

Key Features:

- **Code Editor:** Write and edit your Arduino sketches
- **Verify Button:** Check your code for errors
- **Upload Button:** Send your code to the Arduino board
- **Serial Monitor:** View output from your Arduino
- **Board Selection:** Choose the correct board type
- **Port Selection:** Select the USB port your board is connected to

CH340G Driver Installation

What is CH340G?

The CH340G is a USB-to-Serial converter chip used in many NodeMCU boards and Arduino clones. Your computer needs the proper driver to communicate with this chip.

Installation Steps

1. **Download the Driver**
   Get the CH340G driver from
   https://cdn.sparkfun.com/assets/learn_tutorials/5/9/7/Windows-CH340-Driver.zip

2. **Install the Driver**
   Windows: Run the installer (.exe file)
   macOS: Open the .dmg file and follow instructions
   Linux: Usually no installation needed (built-in kernel support)

3. **Verify Installation**
   Windows: Check Device Manager for "USB-SERIAL CH340"
   macOS: Check System Report > USB for "wch.cn"
   Linux: Run `dmesg | grep CH34x` in terminal

Troubleshooting Tips:

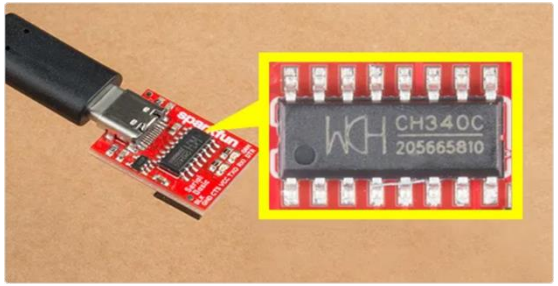
Try a different USB port

Restart your computer after installation

Try a different USB cable (data cable, not charge-only)

On macOS, you may need to allow the driver in Security & Privacy

CH340G Chip Identification



CH340G Chip on a USB-to-Serial Adapter

Why is this important?

Without the CH340G driver, your computer won't recognize the NodeMCU board when connected via USB. This means you won't be able to:

Upload code to the board

Communicate with the board via Serial Monitor

Power the board through USB (in some cases)

Common Issues:

- **"Port not found" in Arduino IDE:** Driver not installed or recognized
- **Board connects/disconnects repeatedly:** Faulty USB cable or insufficient power
- **Driver installation fails:** Incompatible OS version or security restrictions

ESP8266 Board Support Setup

Adding ESP8266 Board Support

1. **Open Arduino IDE Preferences**
   Go to **File > Preferences** in the Arduino IDE menu

2. **Add Board Manager URL**
   In the "Additional Boards Manager URLs" field, paste:

   > http://arduino.esp8266.com/stable/package_esp8266com_index.json

3. **Open Boards Manager**
   Go to **Tools > Board > Boards Manager...**

4. **Install ESP8266 Package**
   Search for "esp8266" and install the package by "ESP8266 Community"

5. **Select NodeMCU Board**
   After installation, go to **Tools > Board** and select **"NodeMCU 1.0 (ESP-12E Module)"**

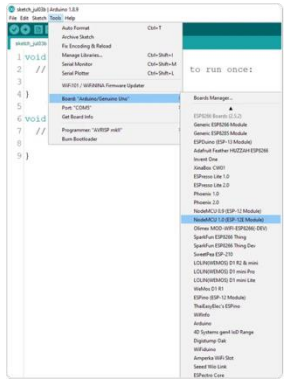
To install libraries:

Go to **Tools > Manage Libraries...**

Search for the library name

Click "Install" (install dependencies if prompted)

Required Libraries Installation



ESP8266 Board Manager Installation

Libraries for Workshop:

- **DHT Sensor Library** by Adafruit
- **Adafruit GFX Library** (for OLED display)
- **Adafruit SSD1306** (for OLED display)
- **MQ135 by GeorgK** (for Gas Sensor)
- **Adafruit Unified Sensor** (dependency)
- **Mpu6050 by Ewan Leng** (For gyroscope)

1.

NodeMCU ESP8266 Introduction

What is NodeMCU ESP8266?

NodeMCU is an open-source IoT platform that includes firmware and development board based on the ESP8266 WiFi SoC from Espressif Systems.

Key Features:

Built-in WiFi

802.11 b/g/n WiFi connectivity with TCP/IP stack

Powerful Processor

Tensilica L106 32-bit processor running at 80MHz

Memory

4MB Flash memory and ~50KB usable RAM

GPIO Pins

11 digital I/O pins with PWM, I2C, SPI, and ADC capabilities

Power

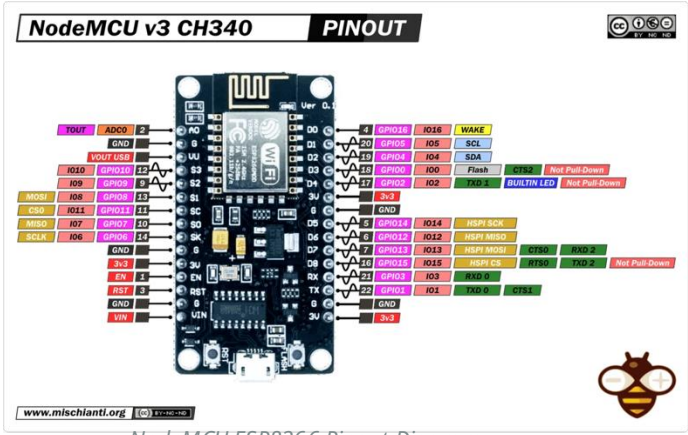
USB powered (5V) with 3.3V operating voltage for I/O

Cost-Effective

Low-cost development board perfect for IoT projects

> **Note:**
>
> For this workshop, we'll be using the NodeMCU 12E version, which is one of the most common variants available.

NodeMCU Pinout



NodeMCU ESP8266 Pinout Diagram

Pin Types:

D0-D8: Digital I/O pins (GPIO pins)

A0: Analog input (0-3.3V, 10-bit resolution)

3V3: 3.3V power output

GND: Ground pins

- **Vin:** External power input (5V)
- **RST:** Reset pin

Physics-Relevant Sensors for ESP8266

Temperature & Humidity

DHT11/DHT22: Temperature and humidity

Applications: Thermodynamics, heat transfer, phase changes

BME680: Temperature, humidity, pressure, gas

Applications: Gas laws, atmospheric physics

Motion & Acceleration

MPU6050: 3-axis accelerometer and gyroscope

Applications: Kinematics, dynamics, rotational motion

HC-SR501: PIR motion sensor

Applications: Event triggering, motion detection

Light & Optics

LDR: Light dependent resistor

Applications: Light intensity, photodetectors

BH1750: Digital ambient light sensor

Applications: Photometry, inverse square law

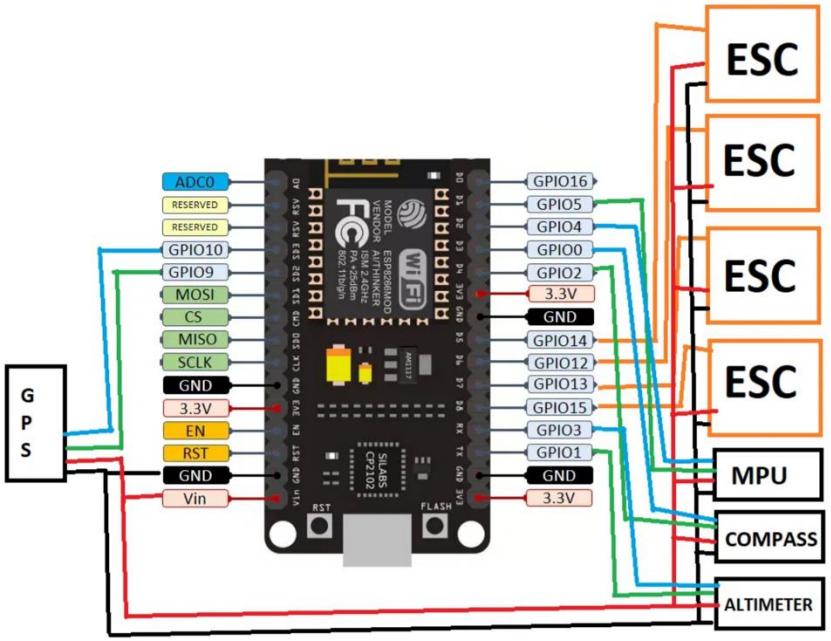
Sound & Force

KY-038: Sound sensor

Applications: Acoustics, sound wave detection

Force Sensor: Pressure/force detection

Applications: Hooke's Law, pressure measurement



ESP8266 Connected to Multiple Sensor Modules

Sensor Connection Types

Digital Connections

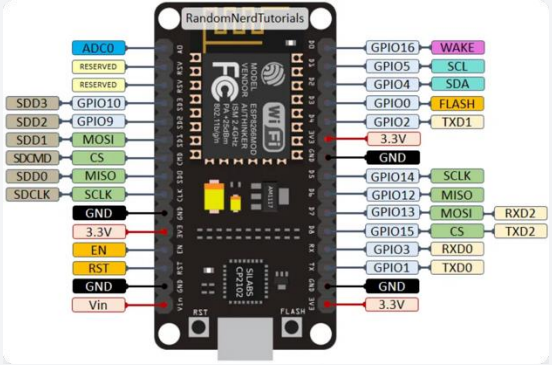
- Uses GPIO pins (D0-D8 on NodeMCU)
- For ON/OFF or HIGH/LOW signals
- Examples: Push buttons, LEDs, relay modules
- Use **pinMode()** and **digitalWrite()/digitalRead()**

Analog Connection

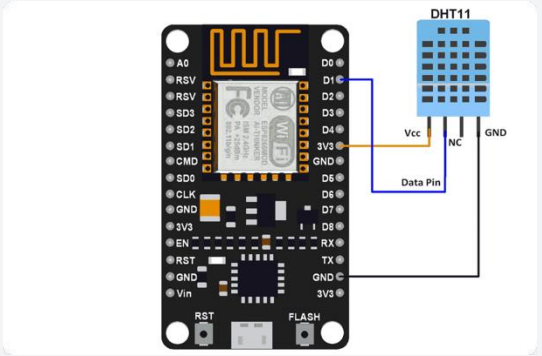
- ESP8266 has one analog input (A0)
- 10-bit resolution (0-1023 values)
- Examples: Potentiometers, light sensors
- Use **analogRead()** function

Communication Protocols

- **I2C** : For multiple sensors (SDA/SCL pins)
- **SPI**: Faster data transfer (MOSI/MISO/SCK)
- **UART** : Serial communication (TX/RX)
- **OneWire** : For temperature sensors like DS18B20



ESP8266 NodeMCU Pinout Reference



Example: DHT11 Temperature Sensor Connection

Basic Programming Concepts

Arduino Code Structure

- **setup()** function: Runs once at the start
- **loop()** function: Runs repeatedly
- Variable declarations at the top
- Custom functions as needed

Key Functions

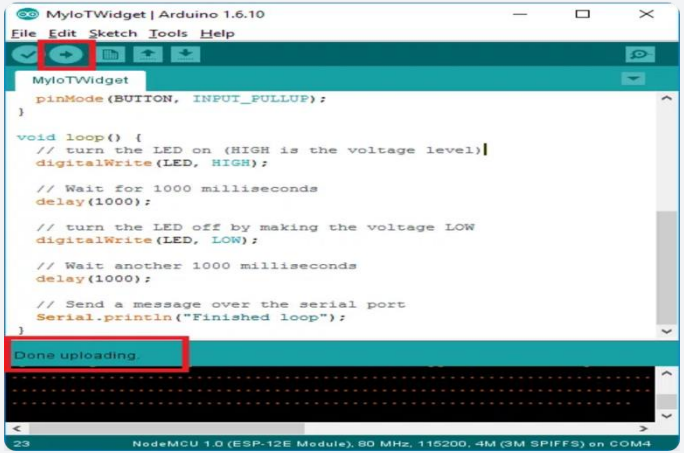
- **pinMode(pin, mode)** : Configures a pin as INPUT or OUTPUT
- **digitalWrite(pin, value)** : Sets the pin HIGH or LOW
- **digitalRead(pin)** : Reads the value from a pin
- **delay(ms)** : Pauses the program for specified time

Uploading and Testing

- Click the upload button (right arrow icon)
- Wait for code to compile and upload
- Check the Serial Monitor for debugging
- Observe the board and connected components

Basic Code Example

```
// LED Blink Example  void setup() { pinMode(LED_BUILTIN, OUTPUT);  // Set LED pin as output  } void loop() { digitalWrite(LED_BUILTIN, HIGH);  // Turn LED on  delay(1000);  // Wait for 1 second  digitalWrite(LED_BUILTIN, LOW);  // Turn LED off  delay(1000);  // Wait for 1 second  }
```



Arduino IDE Interface with ESP8266 Code

Experiments 1-2: LED Traffic Light & Light Sensor

Experiment 1: The "Hello, World!" of Hardware

Objective: Learn the basics of Arduino IDE, upload your first program, and control a digital output.

Components:

- NodeMcu ESP8266
- LED Traffic Lights Signal Module
- Male to Female Jumper Wires (3x)
- Micro USB Cable

LED Traffic Light Module

Connections:

Traffic Light Module Pin	NodeMCU Pin
G (Green LED)	D1 (GPIO5)
Y (Yellow LED)	D2 (GPIO4)
R (Red LED)	D3 (GPIO0)
GND	GND

Experiment 2: Sensing the Light

Objective: Read an analog signal from the LDR (Light Dependent Resistor) module to measure ambient light levels.

Components:

- NodeMcu ESP8266
- LDR Light Sensor Module
- Male to Female Jumper Wires (3x)

LDR Light Sensor Module

Connections:

LDR Module Pin	NodeMCU Pin
A0 (Analog)	A0
VCC / +	3V3
GND	GND

Expected Outcome:

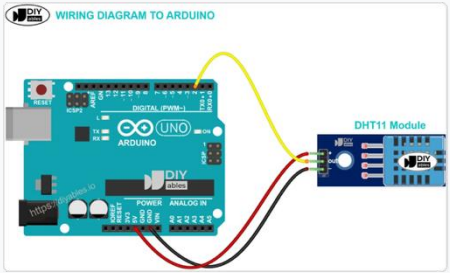
The Serial Monitor will display light level readings. Cover the LDR with your hand to see the value increase (indicating darkness). Shine a light on it to see the value decrease.

Experiments 3-4: DHT11 & OLED Display

Experiment 3: The Digital Weather Station

Objective:

Read temperature and humidity data from the DHT11 sensor



DHT11 Temperature & Humidity Sensor Module

Required Library:

- **DHT sensor library** by Adafruit

Connections:

DHT11 Pin	NodeMCU Pin
DATA / OUT	D4 (GPIO2)
VCC	3V3
GND	GND

Experiment 4: Information at a Glance

Objective:

Display sensor data on an OLED screen to create a standalone device

0.91 inch I2C OLED Display

Required Libraries:

- **Adafruit GFX Library**
- **Adafruit SSD1306**

Connections (I2C):

Component	NodeMCU Pin
OLED SCL/SCK	D1 (GPIO5)
OLED SDA	D2 (GPIO4)
OLED VCC	3V3
OLED GND	GND

Experiments 5-6: Buzzer & MPU-6050

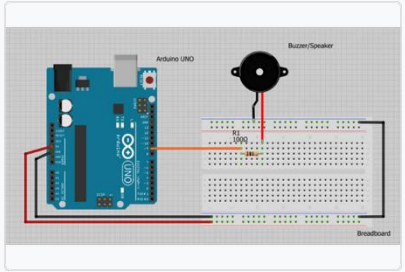
Experiment 5: Creating Alerts

Objective:

Use the Piezo Buzzer to create an audible alert when temperature exceeds a threshold

Connections:

Component	NodeMCU Pin
DHT11 DATA	D4 (GPIO2)
Buzzer +	D5 (GPIO14)
Buzzer -	GND



Piezo Buzzer Module

Experiment 6: Shake, Rattle, and Roll

Objective:

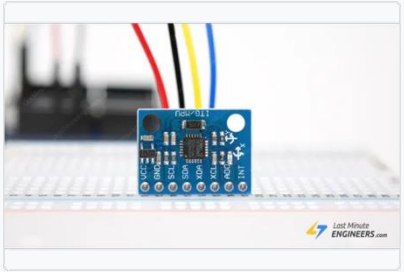
Interface with the MPU-6050, a 3-axis accelerometer and 3-axis gyroscope sensor

Required Libraries:

- **Adafruit MPU6050**
- **Adafruit Unified Sensor**

Connections:

MPU-6050	NodeMCU
VCC	3V3
GND	GND
SCL	D1 (GPIO5)
SDA	D2 (GPIO4)



MPU-6050 Gyroscope/Accelerometer

Expected Outcome:

The Serial Monitor will display acceleration (m/s²) and gyroscope (rad/s) values for all three axes (X, Y, Z). When you tilt or move the sensor, you'll see the values change dramatically.

Experiments 7: Air Quality Monitoring

Experiment 7: Air Quality

Objective:

To detect a wide range of gases, making it ideal for air quality monitoring.

Connections:

Component	NodeMCU Pin
A0	A0
VCC	3V3
GND	GND

Gas Sensor Module

Experiment 8: Your First IoT Project

Connecting to the Internet

Objective:

Connect your device to the Internet and send temperature and humidity data to ThingSpeak.

ThingSpeak Setup:

1. **Create a ThingSpeak Account**
   Sign up at ThingSpeak.com
2. **Create a New Channel**
   Click on "Channels" → "New Channel"
3. **Configure Fields**
   Enable "Field 1" (Temperature) and "Field 2" (Humidity)
4. **Get API Key**
   Go to "API Keys" tab and note your "Write API Key"

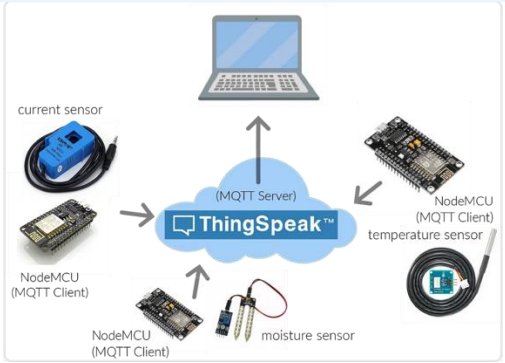
Code Implementation

Required Libraries:

- ESP8266WiFi (included with ESP8266 board support)
- DHT sensor library (from previous experiments)

Expected Outcome:

The NodeMCU will connect to WiFi and upload temperature and humidity data to ThingSpeak every 20 seconds. You can view real-time graphs on the ThingSpeak website.



ThingSpeak IoT Analytics Platform

Experiment 9: Save the Data to Google Sheet

Go to sheets.new to create a new spreadsheet.
Set up your columns in the first row. For example:
•A1: Timestamp
•B1: Temperature (C)
•C1: Humidity (%)

Create the Google Apps Script
•In your new sheet, go to **Extensions** → **Apps Script**.
•Delete any placeholder code in the Code.gs window and **paste the following code**:

Click the **Save project** icon (💾). Give your project a name when prompted (e.g., "Sensor Data Logger").

```
// This function runs when the web app URL receives an HTTP GET request.
function doGet(e) {
  // Get the active spreadsheet and the specific sheet named "Sheet1".
  var ss = SpreadsheetApp.getActiveSpreadsheet();
  var sheet = ss.getSheetByName("Sheet1");

  // Get the temperature and humidity values from the URL parameters.
  // Example URL: .../exec?temp=25.5&hum=45.2
  var temperature = e.parameter.temp;
  var humidity = e.parameter.hum;

  // Create a new Date object for the timestamp.
  var timestamp = new Date();

  // Add a new row to the sheet with the collected data.
  sheet.appendRow([timestamp, temperature, humidity]);

  // Return a success message.
  return ContentService.createTextOutput("Data received successfully.");
}
```

Deploy the Script as a Web App

This is the most important step.

- Click the blue **Deploy** button in the top-right corner and select **New deployment**.
- Click the gear icon ( ) next to "Select type" and choose **Web app**.
- Under "Configuration", fill in the details:
    - **Description:** "NodeMCU Sensor Data" (optional).
    - **Execute as:** Me (your Google account).
    - **Who has access: Anyone**. This is critical for the NodeMCU to be able to send data without needing to log in.
- Click **Deploy**.
- Google will ask you to **Authorize access**. Click the button, choose your Google account, click "Advanced", and then "Go to [Your Project Name] (unsafe)". Allow the permissions.
- After authorizing, a new window will appear with your **Web app URL**. **Copy this URL.** It will look like https://script.google.com/macros/s/LONG_RANDOM_ID/exec

Experiment 10: Create a Website to Display the Data

Publish Your Google Sheet to the Web

1.Open your Google Sheet.

2.Go to the menu and click **File** → **Share** → **Publish to web**.

3.In the new window, under the "Link" tab:

1. In the first dropdown, select the specific sheet that has your data (e.g., Sheet1).
2. In the second dropdown, select **Comma-separated values (.csv)**.

4.Click the green **Publish** button. Confirm that you want to publish.

5.A box will appear with a URL. **Copy this URL.** This is the link your HTML page will use to read the data.

Experiment 10: Create a Website to Display the Data

this is a link to a google csv file. there may be date alone and values missing. In this the data will be updated automatically from an iot device. you need to create a html page to parse the csv data using a script to display the live data last 30 values and a dashboard of current data using a script. create a single html. https://docs.google.com/spreadsheets/d/e/2PACX-1vSCgq-zgskunwfCvm9ABxto7a2lja8iSkd-N4Ah0Ukif6jbTUVlCCusS/pub?gid=0&single=true&output=csv

https://onecompiler.com/

Advanced Concepts and Project Ideas

Advanced ESP8266 Techniques

Power Management

Deep sleep mode can reduce power consumption from ~70mA to ~20µA, extending battery life from days to months

OTA (Over-The-Air) Updates

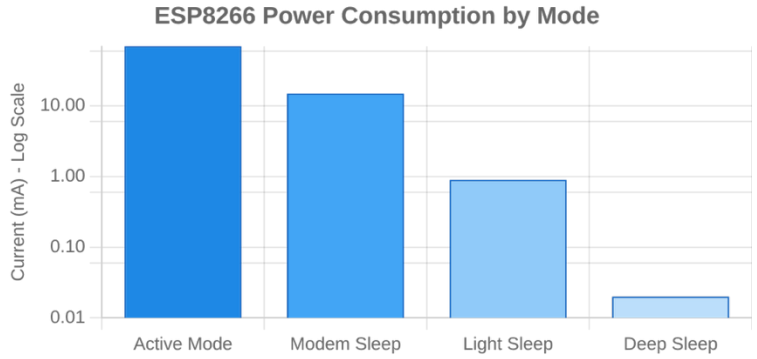
Update firmware wirelessly without physical connection to the device

Web Server Implementation

Create a local web interface for controlling experiments and viewing data

Data Logging to SD Card

Store large datasets locally for later analysis



Implementation Tips

Start with simple projects and gradually add complexity
Use libraries when available to simplify development
Document your code thoroughly for future reference

Consider creating a custom PCB for permanent installations

Implement error handling for robust operation

Physics Project Ideas

Pendulum Period Measurement : Use MPU6050 to precisely measure oscillation period and analyze damping effects

Magnetic Field Mapper : Create a 3D visualization of magnetic field strength using Hall effect sensors

Environmental Physics Monitor : Track temperature, pressure, humidity changes and correlate with physical phenomena

AI-Assisted Coding for ESP8266

How AI Can Help You Code

Generate Code Snippets

Provide natural language descriptions to get working code examples

Debug and Troubleshoot

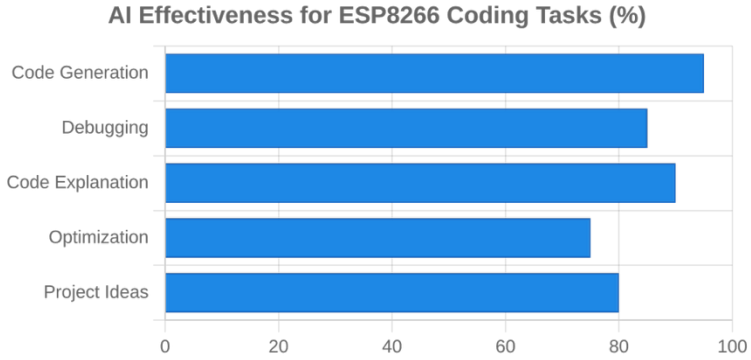
Paste error messages to get solutions and explanations

Explain Code

Understand unfamiliar code or complex functions

Optimize and Refactor

Improve efficiency and readability of your code



Example Prompts

> "Generate Arduino code for ESP8266 to read temperature from DHT11 sensor and print to serial monitor"

> "How can I calculate tilt angle from accelerometer readings (a.acceleration.x, y, z)?"

Best Practices

- Be specific in your prompts
- Iterate and refine your requests
- Understand the code, don't just copy
- Verify information with official documentation
- Provide context when debugging


Thank You
Questions / Feed back