Desktop companion bot workshop

25th, Nov ‘25

What You'll Build Today

In this workshop, you'll assemble and program a desktop companion robot that brings a touch of personality to your workspace. This project will guide you through connecting hardware and writing code to make your robot interactive.

ESP8266 Microcontroller

The brain of your robot, handling all processing and enabling interactive responses to its environment.

MPU6050 Motion Sensor

Detects movement and tilt, allowing your robot to perceive its surroundings and react accordingly.

2.4" TFT Display

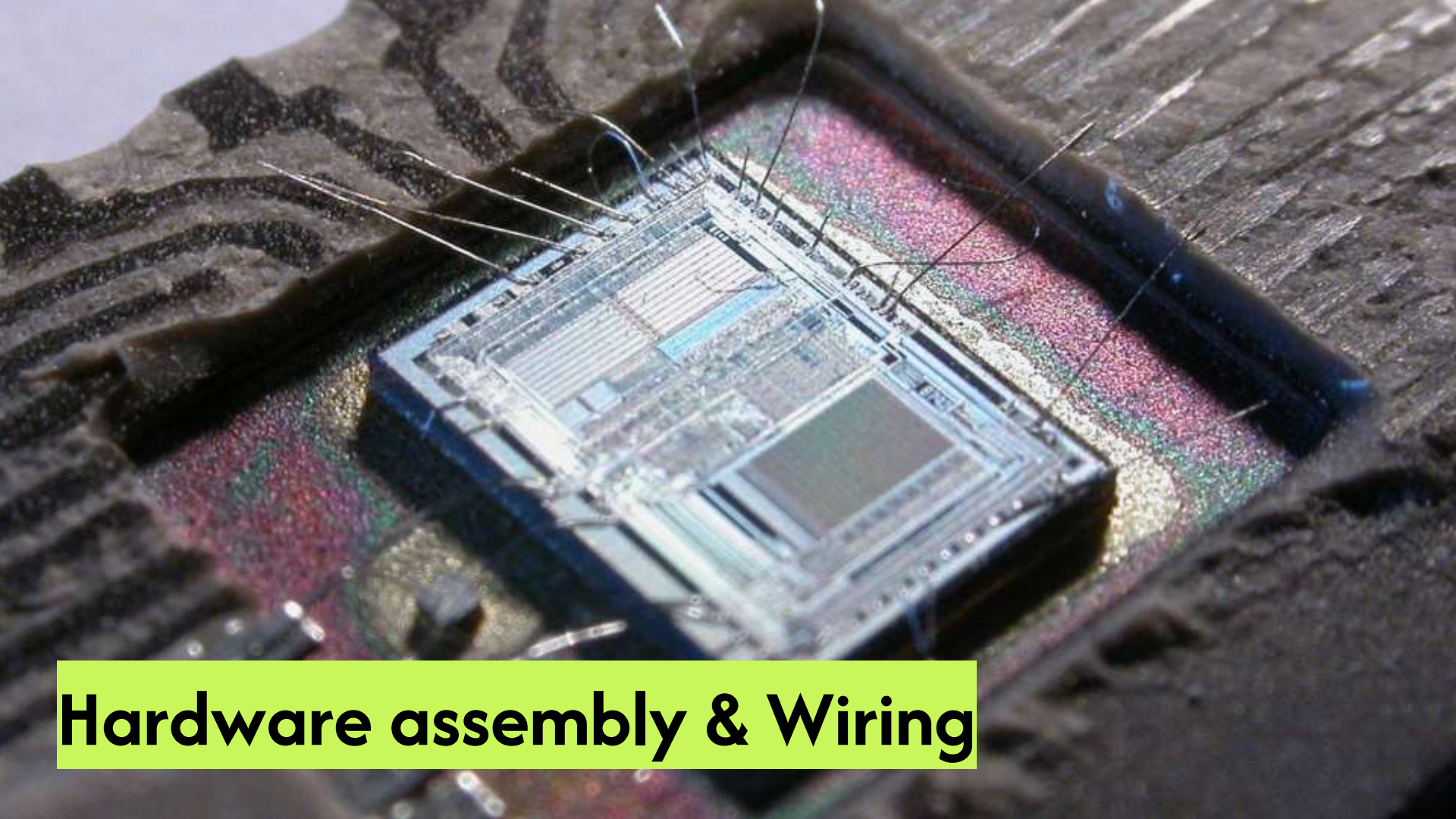
Your robot's animated face, bringing expressions and personality to life based on sensor input.

Components Overview

ESP8266	Microcontroller & WiFi	160MHz, 4MB Flash
MPU6050	Motion Sensor	6-axis (accel + gyro)
TFT Display	Visual Output	2.4" SPI, 320x240px
Bluetooth Speakers Module	Audio Output	Bluetooth, 3.7V
Jumper Wires	Connections	Male-to-male
USB Cable	Power & Programming	Micro-USB
Remote	BLT module	On/off, play/pause and other functionality

SECTION 1: Hardware Assembly & Wiring

Let's build the foundation



Hardware assembly & Wiring

Safety & Power Considerations

- **USB Power:**

  5V input, regulated to 3.3V for components

- **Current Limits:**

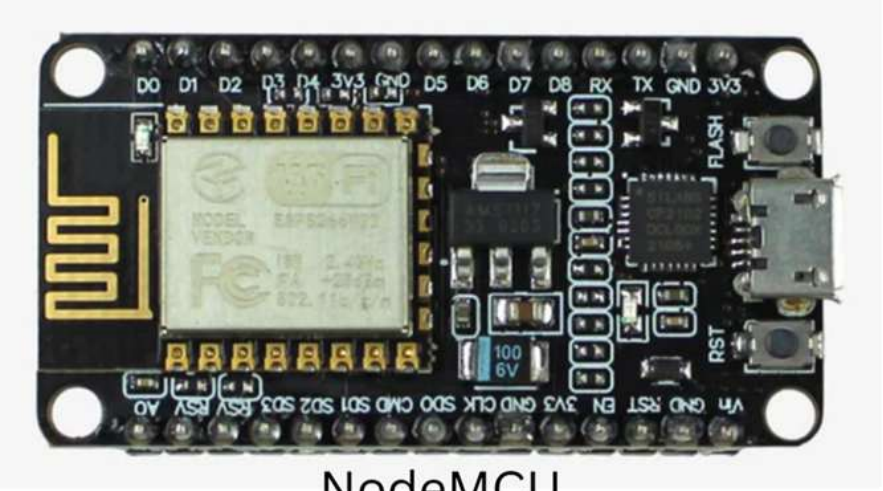
  ESP8266 GPIO max ~12mA per pin

- **Avoid Short Circuits:**

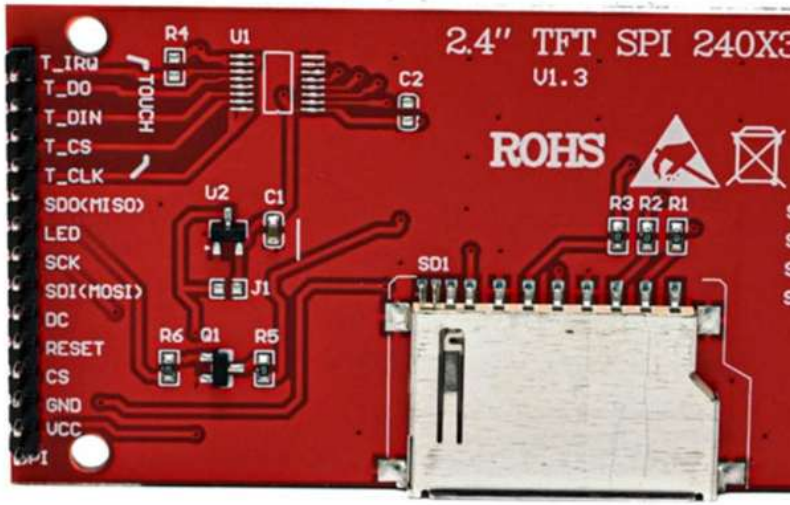
  Don't connect power directly to ground

- **Component Ratings:**

  Check voltage specs (all our components are 3.3V)

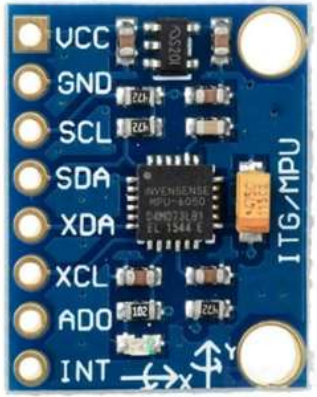


NodeMCU



Display

// Display SDO/MISO to NodeMCU pin D6 (or leave disconnected if not reading TFT)
// Display LED to NodeMCU pin VIN (or 5V, see below)
// Display SCK to NodeMCU pin D5
// Display SDI/MOSI to NodeMCU pin D7
// Display DC (RS/AO)to NodeMCU pin D3
// Display RESET to NodeMCU pin D4 (or RST, see below)
// Display CS to NodeMCU pin D8 (or GND, see below)
// Display GND to NodeMCU pin GND (0V)
// Display VCC to NodeMCU 5V or 3.3V
// MPU VCC to NodeMCU 3.3V
// MPU GND to NodeMCU GND
// MPU SCL to NodeMCU D1
// MPU SDA to NodeMCU D2



MPU

Communication protocols

I2C

i2c (Inter-Integrated Circuit)is a synchronous, multi-master/multi-slave single-ended, serial communication bus.

SPI

Wiring the TFT Display (SPI)

TFT Pin	ESP8266 Pin	Purpose
VCC	3V3	Power
GND	GND	Ground
CS	D8	Chip Select
DC	D3	Data/Command
MOSI	D7	Data In
SCK	D5	Clock
MISO	D6	Data Out
LED	Vin	Backlight
RESET	D4	Reset

SPI is a fast communication protocol. CS selects the device, DC tells it if we're sending data or commands.

Wiring the MPU6050 (I2C)

MPU6050	ESP8266 Pin	Purpose
VCC	3V3	Power
GND	GND	Ground
SCL	D1	Chip Select
SDA	D2	Data/Command

Common Wiring Pitfalls & Debugging

1. Reversed Power/Ground → Check polarity with multimeter
2. Loose Jumper Wires → Wiggle connections, reset wires
3. Wrong Pin Numbers → Double-check against pinout diagram
4. Missing Ground Connection → All components must share GND
5. Breadboard Continuity Issues → Test with multimeter

SECTION 2: Arduino IDE Setup

Getting the software ready

Installing Arduino IDE

01

Download from arduino.cc (free, open-source)

02

Install on your computer (Windows/Mac/Linux)

03

Launch Arduino IDE

04

Connect ESP8266 via USB cable

Arduino IDE is the standard tool for programming microcontrollers. It's free and works on all platforms.

Adding ESP8266 Board Manager

01

Go to File → Preferences

02

In "Additional Boards Manager URLs", paste:

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

03

Click OK

04

Go to Tools → Board → Boards Manager

05

Search for "ESP8266" and install

06

Select Tools → Board → NodeMCU 1.0 (ESP-12E Module)

This tells Arduino IDE how to compile code for the ESP8266 instead of regular Arduino boards.

Installing Libraries

Code Walkthrough

Installing Required Libraries

1. Adafruit_GFX - Graphics library for drawing shapes and text
2. TFT_eSPI - Fast SPI display driver for TFT screens
3. MPU6050 - Motion sensor library (by InvenSense)
4. Wire.h - I2C communication (built-in, no install needed)

Instructions: Search each library name, click Install

These libraries provide pre-written functions so we don't have to code everything from scratch.

Testing Your Setup

```
void setup() {  Serial.begin(115200);  Serial.println("ESP8266 is alive!");}void loop() {  delay(1000);}
```

1. Copy the code above into Arduino IDE
2. Select Tools → Port → (your COM port)
3. Click Upload (arrow button)
4. Open Serial Monitor (Tools → Serial Monitor)
5. You should see "ESP8266 is alive!" printed

Extension Ideas

Easy (Code Only)

- Add more facial expressions (surprised, angry, sleepy)
- Change colors based on time of day
- Add text display (show time, temperature, messages)
- Create animation sequences

Medium (Code + Simple Hardware)

- Add a buzzer for sound effects
- Add an LED for mood indication
- Add a button for interaction
- Connect to WiFi and fetch weather data

Advanced (More Complex)

- Add a servo motor for head movement
- Connect to the internet and respond to voice commands
- Create a web interface to control the robot
- Add machine learning for emotion recognition

Resources & Community

Key Resources

- Arduino Official Documentation
- ESP8266 Community
- Adafruit Learning Guides

Stack Overflow (search your error message)

- GitHub: Find open-source robot projects and code
- YouTube: Tutorials and project walkthroughs

Community

- Join local maker spaces and robotics clubs
- Participate in online forums and Discord communities
- Share your projects on GitHub and social media
- Contribute to open-source projects

Key Takeaways

Hardware

Wired components using breadboards and understood communication protocols (SPI, I2C)

Software

Wrote and uploaded code to a microcontroller using Arduino IDE

Sensors

Read real-world data from a motion sensor and processed it

Graphics

Drew animated graphics on a display

Integration

Combined all components into a working system

Debugging

Learned systematic troubleshooting techniques

Creativity

Built something with personality and character

Wrapping up

What we learnt today?

You learnt :

1. Hardware and Wiring skills
2. Sensors fundamentals
3. Arduino IDE and Logic
4. Code Architecture and Logic
5. Building a functional Desktop companion

How to expand this project?

- Adding WiFi features with ESP8266
- Adding sound, touch sensor, or new animations
- Turning this into a larger robotics/IOT project

Thank you!