

## แนะนำ ESP32

ESP32 คือไมโครคอนโทรลเลอร์ที่พัฒนาโดย Espressif ซึ่งมีราคาถูกและใช้พลังงานต่ำ มาพร้อมกับความสามารถในการเชื่อมต่อ Wi-Fi และ Bluetooth รวมถึงโปรเซสเซอร์แบบบิตอัลคอร์ ทำให้เหมาะสำหรับการใช้งานในโปรเจกต์ต่าง ๆ ไม่ว่าจะเป็น IoT หรือบ้านอัจฉริยะ

### ทำไม ESP32 ถึงได้รับความนิยม?

ESP32 เป็นที่นิยมจากคุณสมบัติหลายประการ ได้แก่ :



**ราคาถูก:** ESP32 เริ่มต้นที่หลักร้อยกว่าบาท ซึ่งทำให้ผู้ใช้ทั่วไปสามารถเข้าถึงได้ง่าย

**ประหยัดพลังงาน:** ใช้พลังงานต่ำและรองรับโหมดประหยัดพลังงาน เช่น โหมดหลับลึก

**Wi-Fi Connectivity:** รองรับการเชื่อมต่อ Wi-Fi ทั้งในโหมดสถานี (เชื่อมต่อกับอินเทอร์เน็ต) และโหมดจุดเชื่อมต่อ (สร้างเครือข่าย Wi-Fi ของตัวเอง)

**Bluetooth:** รองรับ Bluetooth แบบคลาสสิกและ Bluetooth Low Energy (BLE) สำหรับแอปพลิเคชัน IoT

**Dual-core Processor:** มีโปรเซสเซอร์แบบ dual-core Xtensa 32-bit LX6 ทำให้สามารถทำงานได้รวดเร็ว

หลากหลายอินเตอร์เฟซ: รองรับการทำงานเชื่อมต่อกับอุปกรณ์ต่าง ๆ เช่น ADC, DAC, UART, SPI, I2C, PWM เป็นต้น

## การเขียนโปรแกรมกับ ESP32

**Arduino IDE:** หากคุ้นเคยกับการเขียนโปรแกรมใน Arduino, ESP32 รองรับการทำงานเขียนโปรแกรมในรูปแบบเดียวกัน

**MicroPython:** คุณสามารถใช้ MicroPython ซึ่งเป็น Python ที่ดัดแปลงให้ทำงานกับไมโครคอนโทรลเลอร์

## ข้อมูลทางเทคนิคของ ESP32

สำหรับข้อมูลเชิงลึกทางเทคนิค, นี่คือลักษณะบางประการของ ESP-WROOM-32 :

### การเชื่อมต่อไร้สาย:

**Wi-Fi:** รองรับการถ่ายโอนข้อมูลที่ 150 Mbps

**Bluetooth:** รองรับทั้ง BLE และ Bluetooth Classic

### โปรเซสเซอร์:

Dual-Core 32-bit LX6 ความเร็ว 160 MHz หรือ 240 MHz

## หน่วยความจำ:

**ROM:** 448 KB สำหรับการบูตและฟังก์ชันหลัก

**SRAM:** 520 KB สำหรับข้อมูลและคำสั่ง

**RTC SRAM:** 8 KB สำหรับการเก็บข้อมูลในโหมดหลับลึก

## พลังงานต่ำ:

รองรับการทำงานในโหมดหลับลึก เช่น การแปลง ADC โดยไม่ต้องใช้พลังงานมาก

## อินเตอร์เฟซ:

รองรับ DMA (Direct Memory Access) และอินเตอร์เฟซต่าง ๆ เช่น Capacitive Touch, I2C, UART, SPI, I2S, PWM

## ความปลอดภัย:

มีการเข้ารหัส AES และ SSL/TLS สำหรับความปลอดภัยในการส่งข้อมูล

## ความแตกต่างหลักระหว่าง ESP32 และ ESP8266

ESP32 เป็นรุ่นที่พัฒนาต่อยอดจาก ESP8266 โดยมีการปรับปรุงและเพิ่มคุณสมบัติใหม่ๆ ดังนี้

### ความแตกต่างหลัก:

#### 1. จำนวนคอร์ของโปรเซสเซอร์:

- **ESP32:** ใช้โปรเซสเซอร์แบบดual-core (Dual-core) ทำให้สามารถทำงานได้รวดเร็วและรองรับการทำงานหลายอย่างพร้อมกัน

- **ESP8266**: ใช้โปรเซสเซอร์แบบซิงเกิลคอร์ (Single-core)

## 2. Wi-Fi:

- **ESP32**: รองรับ Wi-Fi ที่เร็วกว่าและได้มาตรฐานที่ใหม่กว่า
- **ESP8266**: Wi-Fi ค่อนข้างช้ากว่าเมื่อเทียบกับ ESP32

## 3. จำนวนและฟังก์ชันของพอร์ต GPIO:

- **ESP32**: มีพอร์ต GPIO มากกว่า และรองรับฟังก์ชันที่หลากหลาย
- **ESP8266**: มีพอร์ต GPIO น้อยกว่าและฟังก์ชันจำกัดกว่า

## 4. Bluetooth:

- **ESP32**: รองรับ Bluetooth 4.2 และ Bluetooth Low Energy (BLE) เหมาะสำหรับแอปพลิเคชันที่ต้องการเชื่อมต่อ Bluetooth
- **ESP8266**: ไม่รองรับ Bluetooth

## 5. การวัดค่าแอนะล็อก:

- **ESP32**: รองรับการวัดค่าแอนะล็อกบน 18 ช่อง (พอร์ตที่รองรับแอนะล็อก)
- **ESP8266**: มีพอร์ต ADC แบบ 10 บิต เพียงช่องเดียว

## 6. โหมดหลับลึก:

- **ESP32**: รองรับการใช้งานพอร์ตที่สามารถสัมผัส (Touch-sensitive pins) เพื่อปลุกบอร์ดจากโหมดหลับลึก
- **ESP8266**: ไม่มีฟังก์ชันสัมผัสเพื่อปลุกบอร์ดจากโหมดหลับลึก

## 7. ราคา:

- **ESP32**: ราคาประมาณ 200-400 บาท (ขึ้นอยู่กับแหล่งที่ซื้อและรุ่นที่เลือก)
- **ESP8266**: ราคาประมาณ 120-200 บาท (ขึ้นอยู่กับแหล่งที่ซื้อและรุ่นที่เลือก)

สรุป:

- **ESP32**: มีความเร็วที่ดีกว่า รองรับ GPIO ที่มากขึ้น มี Bluetooth และการประมวลผลแบบดวลคอร์
- **ESP8266**: เป็นบอร์ดที่พื้นฐานมากขึ้น ใช้โปรเซสเซอร์ซิงเกิลคอร์ ราคาถูกกว่า และมีฟีเจอร์ที่จำกัด

## บอร์ดพัฒนา **ESP32**

ESP32 หมายถึง ชิป ESP32 ที่เป็นเวอร์ชันเปล่า แต่คำว่า "ESP32" มักจะใช้เรียกบอร์ดพัฒนา ESP32 ด้วยเช่นกัน การใช้ชิป ESP32 แบบเปล่าอาจไม่สะดวกหรือเหมาะสม โดยเฉพาะเมื่อคุณกำลังเรียนรู้ ทดสอบ หรือทำโปรโตไทป์ (prototyping) ดังนั้นบอร์ดพัฒนา ESP32 จึงเป็นตัวเลือกที่ดีที่มีวงจรทั้งหมดที่จำเป็น เช่น การจ่ายพลังงาน การโปรแกรม การเชื่อมต่อกับคอมพิวเตอร์ ขาเชื่อมต่ออุปกรณ์เสริม ไฟ LED ที่ใช้ควบคุมและแสดงสถานะ เส้าอากาศสำหรับสัญญาณ Wi-Fi และฟีเจอร์อื่นๆ บางบอร์ดยังมาพร้อมกับฮาร์ดแวร์เสริม เช่น เซนเซอร์ โมดูลต่างๆ หรือหน้าจอ (เช่น บอร์ด ESP32 CYD) หรือกล้อง (เช่น ESP32-CAM)



## วิธีเลือกบอร์ดพัฒนา ESP32

เมื่อคุณเริ่มค้นหาบอร์ด ESP32 ออนไลน์ คุณจะพบว่ามียี่ห้อบอร์ดหลายรุ่นจากผู้ผลิตต่างๆ ซึ่งถึงแม้ว่าบอร์ดทั้งหมดจะทำงานในลักษณะเดียวกัน แต่บางรุ่นอาจเหมาะสมกับโปรเจกต์บางประเภทมากกว่ารุ่นอื่นๆ เมื่อต้องการเลือกบอร์ดพัฒนา ESP32 คุณควรพิจารณาประเด็นเหล่านี้:

### 1. USB-to-UART Interface และ วงจรควบคุมแรงดัน

บอร์ดพัฒนาเต็มรูปแบบส่วนใหญ่จะมีพีเจิร์สองตัวนี้ ซึ่งจำเป็นสำหรับการเชื่อมต่อ ESP32 กับคอมพิวเตอร์เพื่ออัปโหลดโค้ดและจ่ายพลังงานให้กับบอร์ด

### 2. ปุ่ม BOOT และ RESET/EN

บอร์ดบางรุ่นอาจไม่มีปุ่ม BOOT แต่จะสามารถเข้าสู่โหมดการแฟลช (flash) โดยอัตโนมัติ ส่วนปุ่ม RESET/EN ใช้สำหรับรีเซ็ต (รีสตาร์ท) บอร์ด

### 3. การตั้งค่าของ GPIO และจำนวนขา

เพื่อใช้งาน ESP32 ได้อย่างถูกต้องในโปรเจกต์ของคุณ คุณต้องเข้าถึงการตั้งค่าขา (pinout) ที่แสดงว่าขาไหนเป็น GPIO และพีเจิร์ของมัน คุณจึงควรตรวจสอบว่าได้เข้าถึงการตั้งค่าขาของบอร์ดที่คุณเลือก

### 4. ช่องเชื่อมต่อเสาอากาศ

บอร์ดส่วนใหญ่จะมีเสาอากาศในตัวสำหรับสัญญาณ Wi-Fi แต่บางรุ่นอาจมีช่องเชื่อมต่อเสาอากาศเพื่อให้สามารถเชื่อมต่อเสาอากาศภายนอกได้ ซึ่งจะช่วยให้เพิ่มระยะการรับส่งสัญญาณ Wi-Fi

### 5. ช่องเชื่อมต่อแบตเตอรี่

หากคุณต้องการจ่ายพลังงานให้กับ ESP32 ด้วยแบตเตอรี่ บางบอร์ดพัฒนาอาจมีช่องเชื่อมต่อสำหรับแบตเตอรี่ LiPo คุณสามารถจ่ายพลังงานให้กับ ESP32 ปกติผ่านขาพลังงานได้เช่นกัน

### 6. พีเจิร์ฮาร์ดแวร์เสริม

บางบอร์ดพัฒนา ESP32 มาพร้อมกับพีเจิร์ฮาร์ดแวร์เสริม เช่น หน้าจอ OLED ใน

ตัว โมดูล LoRa โมดูล SIM800 (สำหรับ GSM และ GPRS) ที่ใส่แบตเตอรี่  
กล้อง หน้าจอ TFT Touchscreen หรือพีเจอาร์อื่นๆ

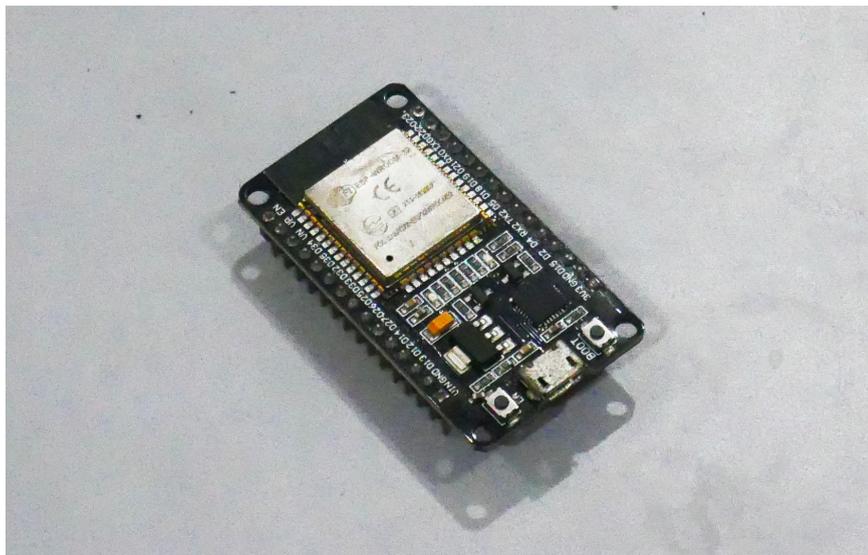
บอร์ดพัฒนา **ESP32** ที่ดีที่สุดสำหรับผู้เริ่มต้น

สำหรับผู้เริ่มต้น เราขอแนะนำบอร์ด ESP32 ที่มีขา GPIO หลากหลายและไม่มีพีเจอาร์ฮาร์ดแวร์เสริมใดๆ ซึ่งจะทำให้คุณเรียนรู้และทดลองใช้งานได้ง่ายขึ้น นอกจากนี้ยังควรมีวงจรรักษาแรงดัน (voltage regulator) และช่อง USB สำหรับการจ่ายพลังงานและอัปโหลดโค้ด

ในโปรเจกต์ ESP32 ส่วนใหญ่ของเรา เราใช้บอร์ด **ESP32 DEVKIT DOIT** และแนะนำให้ผู้เริ่มต้นเลือกบอร์ดนี้ เนื่องจากมีความเหมาะสมสำหรับการเรียนรู้และทดลองโปรเจกต์ต่างๆ โดยบอร์ดนี้มีหลายเวอร์ชันที่มีจำนวนขา GPIO แตกต่างกัน (เช่น 30, 36, และ 38 ขา) ซึ่งทุกเวอร์ชันจะทำงานในลักษณะเดียวกัน

## ESP32 DEVKIT DOIT

ในคอร์สนี้ เราจะใช้บอร์ด **ESP32 DEVKIT DOIT** เป็นตัวอย่างอ้างอิง หากคุณใช้บอร์ดอื่นๆ ก็ไม่ต้องกังวล ข้อมูลใน eBook นี้สามารถใช้ได้กับบอร์ดพัฒนา ESP32 ส่วนใหญ่ ภาพด้านล่างแสดงบอร์ด **ESP32 DEVKIT DOIT V1** ซึ่งเป็นเวอร์ชันที่มี 30 ขา GPIO



## Specifications – ESP32 DEVKIT V1 DOIT

ตารางด้านล่างแสดงคุณสมบัติและสเปคของบอร์ด **ESP32 DEVKIT V1 DOIT** :

คุณสมบัติ	รายละเอียด
จำนวนคอร์	2 คอร์ (dual-core)
<b>Wi-Fi</b>	2.4 GHz รองรับความเร็วสูงสุด 150 Mbit/s
<b>Bluetooth</b>	BLE (Bluetooth Low Energy) และ Bluetooth รุ่นเก่า
สถาปัตยกรรม	32 บิต
ความถี่นาฬิกา	สูงสุด 240 MHz
<b>RAM</b>	512 KB
ขา <b>GPIO</b>	30, 36, หรือ 38 ขึ้นอยู่กับรุ่นของบอร์ด
อุปกรณ์ต่อพ่วง	Capacitive touch, ADC (ตัวแปลงสัญญาณอนาล็อกเป็นดิจิทัล), DAC (ตัวแปลงสัญญาณดิจิทัลเป็นอนาล็อก), I2C (วงจรรวมระหว่างอุปกรณ์), UART (ตัวรับ/ส่งข้อมูลแบบอะซิงโครนัส), CAN 2.0 (คอนโทรลเลอร์พื้นที่เครือข่าย), SPI (อินเตอร์เฟซอุปกรณ์ต่อพ่วงแบบอนุกรม), I2S (อินเตอร์เฟซเสียง)

	ระหว่างชิป), RMI (อินเทอร์เฟซมีเดียอิสระแบบย่อ), PWM (การปรับความกว้างของพัลส์)
--	---

บอร์ด **ESP32 DEVKIT V1 DOIT** นี้มาพร้อมกับขา GPIO จำนวน 30 ขา โดยมี 15 ขาที่แต่ละด้าน จำนวนของขา GPIO ที่สามารถใช้งานได้ขึ้นอยู่กับรุ่นของบอร์ดที่เลือกใช้

บอร์ด **ESP32 DEVKIT V1 DOIT** มาพร้อมกับพอร์ต **microUSB** ที่สามารถใช้เชื่อมต่อบอร์ดกับคอมพิวเตอร์ของคุณเพื่ออัปโหลดโค้ดหรือจ่ายไฟให้กับบอร์ดได้

บอร์ดนี้ใช้ชิป **CP2102** (USB-to-UART) เพื่อสร้างการเชื่อมต่อกับคอมพิวเตอร์ผ่านพอร์ต **COM** โดยใช้การเชื่อมต่อแบบอนุกรม อีกชิปที่นิยมใช้คือ **CH340** ซึ่งคุณต้องตรวจสอบว่าบอร์ดของคุณใช้ชิป **USB-to-UART** แบบใด เพื่อที่จะติดตั้งไดรเวอร์ที่เหมาะสมให้คอมพิวเตอร์สามารถสื่อสารกับบอร์ดได้ โดยข้อมูลเพิ่มเติมเกี่ยวกับกระบวนการนี้จะถูกอธิบายในภายหลัง

บอร์ดยังมาพร้อมกับปุ่ม **RESET** (บางรุ่นอาจมีสัญลักษณ์ว่า EN) ที่ใช้สำหรับรีเซ็ตบอร์ด และปุ่ม **BOOT** เพื่อทำให้บอร์ดเข้าสู่โหมดการแฟลช (พร้อมรับโค้ด) ซึ่งบางรุ่นอาจไม่มีปุ่ม **BOOT**

นอกจากนี้ยังมี **LED** สีน้ำเงิน ในตัวที่เชื่อมต่อกับ **GPIO 2** (ในบางรุ่นอาจเชื่อมต่อกับ GPIO อื่น) LED นี้มีประโยชน์ในการดีบักเพื่อแสดงผลทางกายภาพให้เห็น และยังมี **LED** สีแดง ที่จะสว่างเมื่อคุณจ่ายไฟให้กับบอร์ด

### คู่มือการใช้งาน Pinout ของ ESP32 GPIO

ชิป **ESP32** มาพร้อมกับ 48 ขา (pins) ที่มีฟังก์ชันหลากหลาย แต่ไม่ทุกขาจะถูกใช้งานในทุกบอร์ดพัฒนา ESP32 และบางขาไม่ควรนำมาใช้ในบางกรณี บอร์ด **ESP32 DEVKIT V1 DOIT** จะมาพร้อมกับขา **GPIO** ที่เปิดใช้งาน 36 ขา ซึ่งสามารถใช้เชื่อมต่อกับอุปกรณ์เสริมได้ ขึ้นอยู่กับรุ่นของบอร์ดที่คุณใช้ จำนวนขาที่เปิดใช้งานอาจแตกต่างกันไป

### ขาลังงาน (Power Pins)

โดยทั่วไปแล้ว บอร์ดจะมีขาพลังงานหลัก เช่น **3V3**, **GND**, และ **VIN** ขาเหล่านี้สามารถใช้เพื่อจ่ายพลังงานให้กับบอร์ด (หากคุณไม่ได้จ่ายไฟผ่านพอร์ต USB) หรือใช้จ่ายพลังงานให้กับอุปกรณ์เสริมอื่น ๆ (หากคุณจ่ายไฟให้กับบอร์ดผ่านพอร์ต USB)

## ขา **GPIO** (General Purpose Input Output)

เกือบทุก **GPIO** จะมีหมายเลขที่กำหนด และคุณควรอ้างอิงถึงขาเหล่านั้นโดยใช้หมายเลขของมัน

ด้วย **ESP32** คุณสามารถกำหนดได้ว่าขาไหนจะใช้เป็น **UART**, **I2C**, หรือ **SPI** – คุณเพียงแค่ตั้งค่าในโค้ด (คุณสามารถเรียนรู้เพิ่มเติมเกี่ยวกับเรื่องนี้ในภายหลัง) สิ่งนี้เป็นไปได้เพราะคุณสมบัติการทำงานหลายฟังก์ชัน (Multiplexing) ของชิป ESP32 ที่ช่วยให้สามารถกำหนดฟังก์ชันหลายอย่างให้กับขาเดียวกันได้

หากคุณไม่ได้ตั้งค่าในโค้ด ขาเหล่านั้นจะถูกกำหนดค่าเป็นค่าเริ่มต้นตามที่แสดงในแผนผัง Pinout (ตำแหน่งขาอาจแตกต่างกันไปขึ้นอยู่กับผู้ผลิต) นอกจากนี้ยังมีขาบางขาที่มีคุณสมบัติเฉพาะ ซึ่งทำให้มันเหมาะสมหรือไม่เหมาะสมกับการใช้งานในโครงการบางประเภท

# Deva DIY