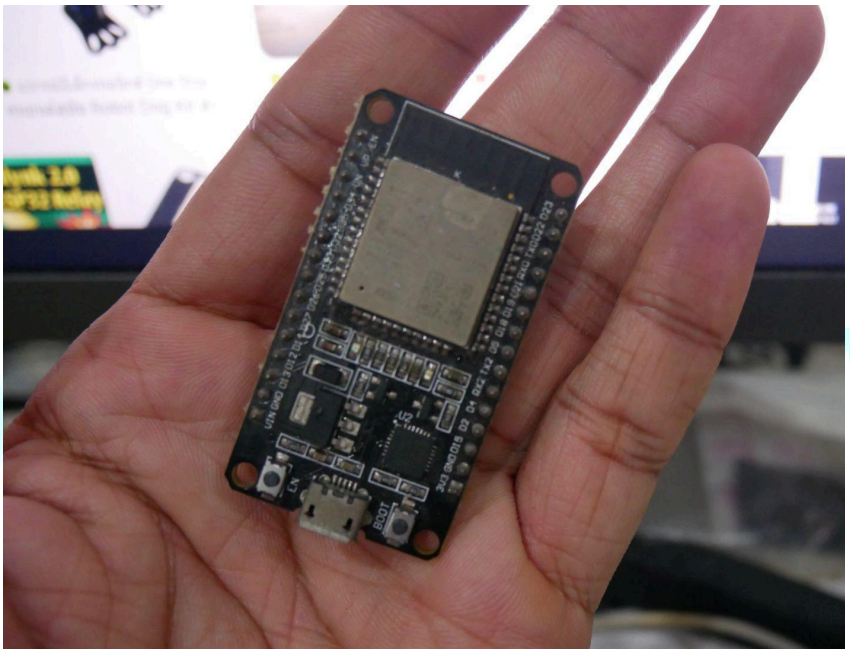
บอร์ด **ESP32** สำหรับบทเรียนนี้

สำหรับการเรียนการสอนในหลักสูตรนี้ คุณจะได้ใช้งาน บอร์ด ESP32 ซึ่งเหมาะสำหรับโปรเจกต์ IoT และการเขียนโปรแกรมแบบเรียลไทม์ บอร์ด ESP32 ที่แนะนำ เช่น DOIT ESP32 DEVKIT V1 หรือรุ่นที่ใกล้เคียง โดยบอร์ดเหล่านี้รองรับฟังก์ชันต่าง ๆ ที่ช่วยให้การพัฒนาโปรเจกต์เป็นเรื่องง่าย

ตรวจสอบให้แน่ใจว่าคุณได้ติดตั้ง Arduino IDE และไดรเวอร์ที่จำเป็นสำหรับการใช้งาน ESP32 เรียบร้อยแล้ว.



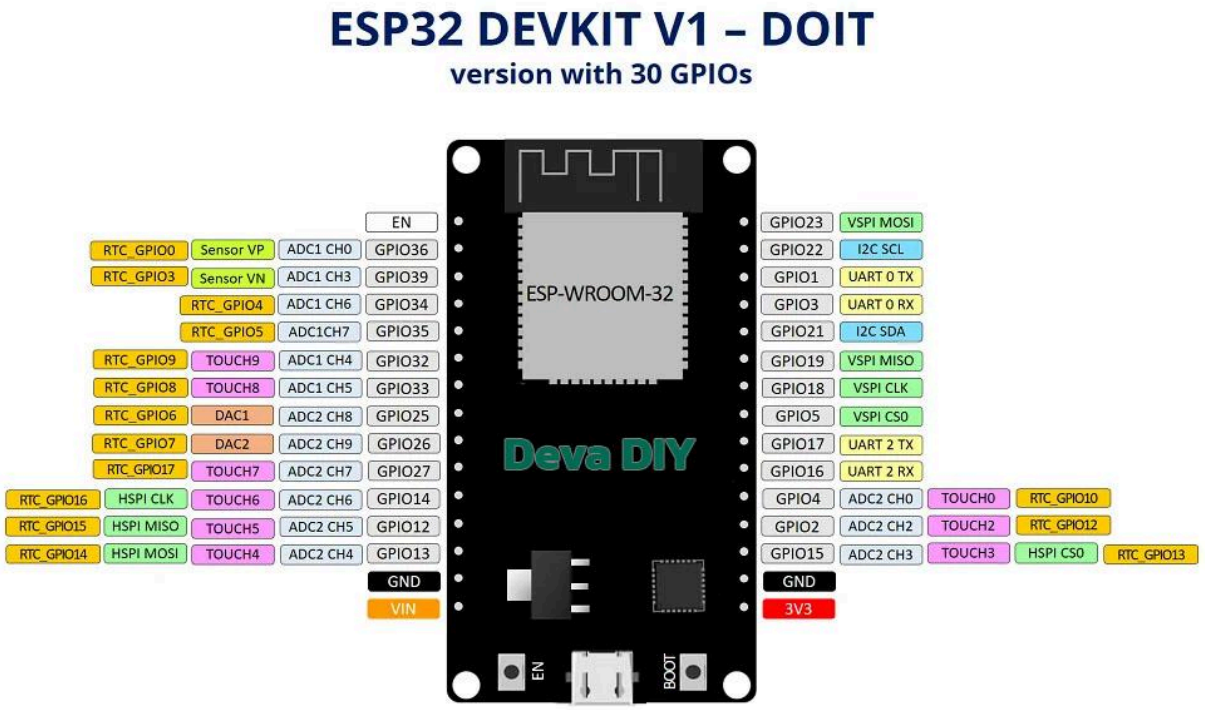
แผนผังพิน **(Pinout)** ของ **ESP32**

การรู้ชื่อรุ่นของบอร์ด ESP32 ที่คุณใช้งานเป็นสิ่งสำคัญ เพื่อที่คุณจะสามารถค้นหาแผนผังพิน (Pinout) ของบอร์ดได้อย่างถูกต้อง

วิธีค้นหาแผนผังพิน

1. ค้นหาชื่อบอร์ด ESP32 ของคุณ เช่น **"ESP32 DEVKIT V1"**
2. เพิ่มคำว่า **"pinout"** ต่อท้ายชื่อบอร์ดในช่องค้นหา (ตัวอย่าง: "ESP32 DEVKIT V1 pinout")
3. ดูข้อมูลแผนผังพินจากเว็บไซต์หรือคู่มือที่เชื่อถือได้

แผนผังพินนี้จะช่วยให้คุณทราบตำแหน่งพอร์ตต่าง ๆ เช่น GPIO, VCC, GND, และพอร์ตอื่น ๆ ที่สำคัญสำหรับการเชื่อมต่ออุปกรณ์.



อ้างอิงแผนผังพิน: การใช้งาน **GPIO** ของ **ESP32**

ESP32 มีพอร์ต GPIO (General Purpose Input/Output) ที่หลากหลายและสามารถตั้งค่าได้สำหรับงานต่าง ๆ เช่น การอ่านค่าจากเซ็นเซอร์ หรือส่งสัญญาณไปยังอุปกรณ์ต่าง ๆ

การใช้งาน **GPIO** ของ **ESP32 (ESP-WROOM-32)**

ชิป ESP32 (ESP-WROOM-32) มี **48** พิน ที่สามารถทำงานได้หลากหลาย แต่ไม่ได้เปิดใช้งานทุกพินในบอร์ดพัฒนาทุกรุ่น และบางพินไม่สามารถใช้งานสำหรับฟังก์ชันบางอย่างได้

คำถามสำคัญเกี่ยวกับ **GPIO:**

1. พินไหนที่ควรใช้**?**
2. พินไหนที่ควรหลีกเลี่ยง**?**

ในส่วนนี้ เราได้เตรียมคำแนะนำที่ง่ายและเข้าใจได้เกี่ยวกับการใช้งาน **GPIO** ของ **ESP32** เพื่อช่วยให้คุณเลือกพินที่เหมาะสมและหลีกเลี่ยงปัญหาที่อาจเกิดขึ้นในการพัฒนาโปรเจกต์.

อุปกรณ์ต่อพ่วง **(Peripherals)** ของ **ESP32**

ESP32 มาพร้อมกับอุปกรณ์ต่อพ่วงที่หลากหลาย ซึ่งช่วยให้เหมาะสำหรับการพัฒนาโปรเจกต์

รายการอุปกรณ์ต่อพ่วง**:**

- 18x ช่องแปลงสัญญาณอนาล็อกเป็นดิจิทัล (ADC)
- 3x อินเทอร์เฟซ SPI
- 3x อินเทอร์เฟซ UART
- 2x อินเทอร์เฟซ I2C
- 16x ช่องสัญญาณ PWM Output
- 2x ตัวแปลงสัญญาณดิจิทัลเป็นอนาล็อก (DAC)
- 2x อินเทอร์เฟซ I2S
- 10x GPIO ที่รองรับ Capacitive Sensing

หมายเหตุสำคัญ**:**

- ไม่ได้สามารถเข้าถึงทุก GPIO ได้ในบอร์ดพัฒนา ESP32 ทุกรุ่น
- อย่างไรก็ตาม พอร์ต GPIO ที่มีอยู่จะทำงานเหมือนกันหากใช้ชิป **ESP-WROOM-32** เดียวกัน

โปรดตรวจสอบข้อมูลพอร์ต GPIO จากแผนผังพินของบอร์ดพัฒนาของคุณเพื่อการใช้งานที่เหมาะสม!

รายละเอียดของ **GPIO** ที่ใช้บ่อยใน **ESP32 DEVKIT V1 DOIT:**

พอร์ต GPIO บางพอร์ตมีคุณสมบัติเฉพาะที่ทำให้เหมาะสมหรือไม่เหมาะสมกับโปรเจกต์ของคุณ

ตารางการใช้งาน **GPIO**

ตารางด้านล่างแสดงพอร์ตที่เหมาะสำหรับใช้งานเป็น Input, Output และพอร์ตที่ควรระวัง:

> คำแนะนำ: ตรวจสอบคุณสมบัติของพอร์ตที่คุณจะใช้ให้เหมาะสมกับการออกแบบโปรเจกต์ของคุณ เพื่อหลีกเลี่ยงปัญหาที่อาจเกิดขึ้น.

GPIO	Input	Output	หมายเหตุ
GPIO 0	pulled up	ใช้งานได้	ส่งสัญญาณ PWM ขณะบูต
GPIO 1	TX pin	ใช้งานได้	ดีบักเอาต์พุตขณะบูต
GPIO 2	ใช้งานได้	ใช้งานได้	เชื่อมต่อกับ LED บนบอร์ด, ต้องลอยขา หรือ LOW ในระหว่างการบูต
GPIO 3	ใช้งานได้	RX pin	HIGH ในระหว่างการบูต
GPIO 4	ใช้งานได้	ใช้งานได้	
GPIO 5	ใช้งานได้	ใช้งานได้	ส่งสัญญาณ PWM ขณะบูต
GPIO 6	X	X	เชื่อมต่อกับ SPI flash ภายใน

GPIO 7	X	X	เชื่อมต่อกับ SPI flash ภายใน
GPIO 8	X	X	เชื่อมต่อกับ SPI flash ภายใน
GPIO 9	X	X	เชื่อมต่อกับ SPI flash ภายใน
GPIO 10	X	X	เชื่อมต่อกับ SPI flash ภายใน
GPIO 11	X	X	เชื่อมต่อกับ SPI flash ภายใน
GPIO 12	ใช้งานได้	ใช้งานได้	หาก pulled high ขณะบูตล้มเหลว, ขา strapping
GPIO 13	ใช้งานได้	ใช้งานได้	
GPIO 14	ใช้งานได้	ใช้งานได้	ส่งสัญญาณ PWM ขณะบูต
GPIO 15	ใช้งานได้	ใช้งานได้	ส่งสัญญาณ PWM ที่บูต, ขา strapping
GPIO 16	ใช้งานได้	ใช้งานได้	
GPIO 17	ใช้งานได้	ใช้งานได้	
GPIO 18	ใช้งานได้	ใช้งานได้	
GPIO 19	ใช้งานได้	ใช้งานได้	
GPIO 21	ใช้งานได้	ใช้งานได้	
GPIO 22	ใช้งานได้	ใช้งานได้	
GPIO 23	ใช้งานได้	ใช้งานได้	
GPIO 25	ใช้งานได้	ใช้งานได้	
GPIO 26	ใช้งานได้	ใช้งานได้	
GPIO 27	ใช้งานได้	ใช้งานได้	
32	ใช้งานได้	ใช้งานได้	
33	ใช้งานได้	ใช้งานได้	

34	ใช้งานได้		ขาอินพุตเท่านั้น
35	ใช้งานได้		ขาอินพุตเท่านั้น
36	ใช้งานได้		ขาอินพุตเท่านั้น
39	ใช้งานได้		ขาอินพุตเท่านั้น

อธิบาย:

- สีเขียว: ขาสามารถใช้งานปกติ
- สีเหลือง: ขานั้นใช้งานได้แต่ควรระวัง
- สีแดง: ไม่แนะนำให้ใช้

หมายเหตุ:

- ขา **GPIO 6-11** เชื่อมต่อกับ **SPI flash** ภายในและไม่ควรใช้เป็น GPIO ทั่วไป
- ขา **GPIO 34-39** เป็น ขาอินพุตเท่านั้น ซึ่งมักใช้สำหรับอินพุตอนาล็อก (ADC)
- ขา **GPIO 0, 2, 12, 15** เป็น ขา **strapping** ซึ่งสถานะของขาเหล่านี้ในตอนบูตสามารถมีผลต่อการทำงานของ ESP32 (เช่น เข้าสู่โหมดแฟลชหรือโหมดบูต)

Deva DIY

SPI Flash ที่ติดตั้งใน ESP-WROOM-32

ขา GPIO ตั้งแต่ GPIO 6 ถึง GPIO 11 บนบอร์ด ESP32 บางรุ่นอาจจะสามารถใช้งานได้ แต่ขาเหล่านี้เชื่อมต่อกับ **SPI flash** ที่ติดตั้งอยู่ภายใน ชิป ESP-WROOM-32 และไม่แนะนำให้ใช้ขาเหล่านี้ในโปรเจคอื่นๆ เพราะมันถูกใช้สำหรับการติดต่อกับหน่วยความจำแฟลชภายในของชิป

ขา **GPIO** ที่ไม่ควรใช้ในโปรเจค:

- **GPIO 6** (SCK/CLK) - ใช้สำหรับสัญญาณนาฬิกาของ SPI
- **GPIO 7** (SDO/SD0) - ใช้สำหรับส่งข้อมูล SPI
- **GPIO 8** (SDI/SD1) - ใช้สำหรับรับข้อมูล SPI
- **GPIO 9** (SHD/SD2) - ใช้สำหรับเลือกชิป SPI
- **GPIO 10** (SWP/SD3) - ใช้สำหรับสัญญาณข้อมูล SPI

- **GPIO 11** (CSC/CMD) - ใช้สำหรับคำสั่ง SPI

หมายเหตุสำคัญ:

- ห้ามใช้ ขาเหล่านี้สำหรับการเชื่อมต่อกับ GPIO ทั่วไปในโปรเจคของคุณ
- ขาเหล่านี้มีความสำคัญสำหรับการทำงานของ SPI flash ภายในชิป ESP32 ซึ่งเป็นหน่วยความจำหลักของบอร์ด
- การใช้ขาเหล่านี้ในโปรเจคของคุณอาจทำให้เกิดปัญหากับการบูตหรือการแฟลชของ ESP32 ได้

ดังนั้นควรหลีกเลี่ยงการใช้ขาเหล่านี้เพื่อให้โปรเจคของคุณทำงานได้อย่างราบรื่นและไม่มีปัญหากับการใช้งานหน่วยความจำแฟลชภายใน.

เซนเซอร์สัมผัสแบบสัมผัสไฟฟ้า **(Capacitive Touch GPIOs)**

ESP32 มีเซนเซอร์สัมผัสแบบไฟฟ้าภายในจำนวน 10 ตัว ซึ่งสามารถตรวจจับการเปลี่ยนแปลงในสิ่งที่มีประจุไฟฟ้า เช่น ผิวหนังมนุษย์ ดังนั้นมันจึงสามารถตรวจจับการเปลี่ยนแปลงที่เกิดขึ้นเมื่อสัมผัส GPIO ด้วยนิ้วมือ เซนเซอร์เหล่านี้สามารถนำมาใช้งานในแผ่นสัมผัสแบบไฟฟ้าและแทนที่ปุ่มกดทางกายภาพได้อย่างง่ายดาย นอกจากนี้ ขาสัมผัสไฟฟ้าเหล่านี้ยังสามารถใช้เพื่อปลุก ESP32 จากโหมด Deep Sleep ได้อีกด้วย

เซนเซอร์สัมผัสภายในนี้เชื่อมต่อกับ GPIO ต่อไปนี้:

- **T0** (GPIO 4)
- **T1** (GPIO 0)
- **T2** (GPIO 2)
- **T3** (GPIO 15)
- **T4** (GPIO 13)
- **T5** (GPIO 12)
- **T6** (GPIO 14)
- **T7** (GPIO 27)
- **T8** (GPIO 33)
- **T9** (GPIO 32)

เซนเซอร์สัมผัสเหล่านี้สามารถใช้ประโยชน์ในโปรเจคที่ต้องการสัมผัสที่ไวต่อการเปลี่ยนแปลงไฟฟ้า เช่น การควบคุมหรือการตรวจจับการสัมผัสในหลายๆ สถานการณ์ได้อย่างสะดวกและมีประสิทธิภาพ.

ตัวแปลงสัญญาณอนาล็อกเป็นดิจิทัล **(ADC)**

ESP32 มาพร้อมกับ ADC ขาอินพุต 12 บิต จำนวน 18 ขา ซึ่งสามารถใช้งานเป็น ADC ได้ ขาและช่องต่าง ๆ ที่สามารถใช้เป็น ADC ได้แก่:

- **ADC1_CH0** (GPIO 36)
- **ADC1_CH1** (GPIO 37)
- **ADC1_CH2** (GPIO 38)
- **ADC1_CH3** (GPIO 39)
- **ADC1_CH4** (GPIO 32)
- **ADC1_CH5** (GPIO 33)
- **ADC1_CH6** (GPIO 34)
- **ADC1_CH7** (GPIO 35)
- **ADC2_CH0** (GPIO 4)
- **ADC2_CH1** (GPIO 0)
- **ADC2_CH2** (GPIO 2)
- **ADC2_CH3** (GPIO 15)
- **ADC2_CH4** (GPIO 13)
- **ADC2_CH5** (GPIO 12)
- **ADC2_CH6** (GPIO 14)
- **ADC2_CH7** (GPIO 27)
- **ADC2_CH8** (GPIO 25)
- **ADC2_CH9** (GPIO 26)

ช่อง ADC มีความละเอียด 12 บิต ซึ่งหมายความว่าค่าที่อ่านได้จะมีช่วงจาก 0 ถึง 4095 โดยที่ 0 หมายถึง 0V และ 4095 หมายถึง 3.3V คุณยังสามารถตั้งค่าความละเอียดของช่อง ADC และช่วงค่าได้ในโค้ด

Deva DIY

ขา ADC จะถูกแบ่งออกเป็นสองประเภท: **ADC1** และ **ADC2**
หมายเหตุ: ขา ADC2 ไม่สามารถใช้งานได้เมื่อเปิด Wi-Fi ดังนั้นหากคุณใช้งาน Wi-Fi และพบปัญหาในการอ่านค่าจากขา ADC2 คุณอาจลองใช้ขา ADC1 แทน ซึ่งจะช่วยแก้ปัญหานี้ได้

ขา ADC ของ ESP32 ไม่มีพฤติกรรมเชิงเส้น **(non-linear behavior)** ซึ่งหมายความว่าอาจไม่สามารถแยกแยะค่าระหว่าง 0V และ 0.1V หรือระหว่าง 3.2V และ 3.3V ได้ คุณควรคำนึงถึงข้อจำกัดนี้เมื่อใช้งานขา ADC

ตัวแปลงสัญญาณดิจิทัลเป็นอนาล็อก **(DAC)**

บน ESP32 มีช่อง DAC 8 บิต จำนวน 2 ช่อง ที่ใช้ในการแปลงสัญญาณดิจิทัลเป็นสัญญาณแรงดันอนาล็อก ช่อง DAC มีดังนี้:

- **DAC1** (GPIO 25)
- **DAC2** (GPIO 26)

ช่อง DAC ใช้สำหรับการแปลงสัญญาณดิจิทัลให้เป็นสัญญาณแรงดันอนาล็อกที่สามารถใช้งานในโปรเจกต์ที่ต้องการการควบคุมสัญญาณอนาล็อก

Deva DIY

GPIO ของ **RTC (Real-Time Clock)**

ESP32 รองรับการใช้งาน GPIO ของ RTC ซึ่งเชื่อมต่อกับระบบย่อยพลังงานต่ำ (Low-Power Subsystem) ของ RTC ที่สามารถใช้เมื่อ ESP32 อยู่ในโหมด Deep Sleep. GPIO ของ RTC สามารถใช้เป็นแหล่งปลุก (Wake-Up Source) เพื่อปลุก ESP32 ออกจากโหมด Deep Sleep เมื่อคอร์โพรเซสเซอร์ Ultra Low Power (ULP) กำลังทำงาน.

GPIO ที่สามารถใช้เป็นแหล่งปลุกมีดังนี้:

- **RTC_GPIO0** (GPIO 36)
- **RTC_GPIO3** (GPIO 39)

- **RTC_GPIO4** (GPIO 34)
- **RTC_GPIO5** (GPIO 35)
- **RTC_GPIO6** (GPIO 25)
- **RTC_GPIO7** (GPIO 26)
- **RTC_GPIO8** (GPIO 33)
- **RTC_GPIO9** (GPIO 32)
- **RTC_GPIO10** (GPIO 4)
- **RTC_GPIO11** (GPIO 0)
- **RTC_GPIO12** (GPIO 2)
- **RTC_GPIO13** (GPIO 15)
- **RTC_GPIO14** (GPIO 13)
- **RTC_GPIO15** (GPIO 12)
- **RTC_GPIO16** (GPIO 14)
- **RTC_GPIO17** (GPIO 27)

GPIO เหล่านี้สามารถใช้งานในโปรเจกต์ที่ต้องการประหยัดพลังงาน โดยการปลุก ESP32 จากโหมด Deep Sleep ด้วยการสัมผัสหรือกระตุ้นจากภายนอก

Deva DIY

PWM (Pulse Width Modulation) บน ESP32

ESP32 มีตัวควบคุม PWM สำหรับ LED ซึ่งรองรับช่องทาง PWM ได้ตั้งแต่ 6 ถึง 16 ช่อง (ขึ้นอยู่กับรุ่นของ ESP32) ที่สามารถตั้งค่าให้สร้างสัญญาณ PWM ที่มีคุณสมบัติต่างๆ ได้. ขา GPIO ที่สามารถทำงานเป็นเอาท์พุต (ยกเว้น GPIO 34 ถึง 39 ที่ไม่สามารถสร้าง PWM) สามารถใช้งานในการสร้างสัญญาณ PWM.

ในการตั้งค่าสัญญาณ PWM, คุณจะต้องกำหนดพารามิเตอร์ต่างๆ ในโค้ดดังนี้:

- ความถี่ของสัญญาณ (Signal’s frequency)
- Duty cycle (ระยะเวลาของการเปิดสัญญาณ)
- ช่อง PWM (PWM channel) (ตัวเลือก)
- GPIO ที่ต้องการส่งสัญญาณออกไป (GPIO where you want to output the signal)

ตัวอย่างการใช้งาน PWM บน ESP32 จะช่วยให้คุณสามารถควบคุมอุปกรณ์ต่างๆ เช่น ปรับความสว่างของ LED หรือควบคุมมอเตอร์ โดยการกำหนดสัญญาณ PWM ที่เหมาะสมกับการใช้งานของคุณ.

I2C บน ESP32

เมื่อใช้ ESP32 กับ Arduino IDE, ขา I2C เริ่มต้นสำหรับบอร์ด ESP32 ส่วนใหญ่ (ที่รองรับโดยไลบรารี Wire) จะเป็นดังนี้:

- **GPIO 21 (SDA)**: ขา Data (ข้อมูล)
- **GPIO 22 (SCL)**: ขา Clock (นาฬิกา)

SPI บน ESP32

ESP32 มี SPI (Serial Peripheral Interface) 4 ตัว คือ SPI0, SPI1, SPI2 (มักเรียกว่า HSPI) และ SPI3 (มักเรียกว่า VSPI).

- **SPI0** และ **SPI1** ใช้งานภายในเพื่อสื่อสารกับหน่วยความจำแฟลชในตัว และไม่ควรใช้สำหรับงานอื่น ๆ
- คุณสามารถใช้ **HSPI** และ **VSPI** ในการสื่อสารกับอุปกรณ์ภายนอก เพราะแต่ละบัสมีสัญญาณบัสที่แยกจากกัน และสามารถเชื่อมต่อกับ SPI slave ได้สูงสุด 3 ตัว

บอร์ด ESP32 หลายรุ่นมาพร้อมกับการตั้งค่าขา SPI เริ่มต้นไว้ล่วงหน้า ซึ่งการแมปพินสำหรับบอร์ดส่วนใหญ่มีดังนี้:

SPI	MOSI	MISO	CLK	CS
VSPI	GPIO 23	GPIO 19	GPIO 18	GPIO 5

HSPI	GPIO 13	GPIO 12	GPIO 14	GPIO 15

หมายเหตุ: ขา SPI เริ่มต้นอาจแตกต่างกันไปตามบอร์ดที่คุณใช้

Interrupts

ขา GPIO ของ ESP32 ทุกขาสามารถตั้งค่าให้ทำงานเป็นการตอบสนองต่อ Interrupts (การเกิดเหตุการณ์ที่ต้องการให้โปรแกรมตอบสนองทันที)

UART Pins – การสื่อสารแบบอนุกรม

ESP32 รองรับการสื่อสารแบบอนุกรมด้วย UART (Universal Asynchronous Receiver-Transmitter) ที่สามารถเชื่อมต่อกับอุปกรณ์ต่าง ๆ ได้ โดยรองรับถึง 3 อินเตอร์เฟส UART ได้แก่ UART0, UART1, และ UART2 ขึ้นอยู่กับบอร์ด ESP32 ที่คุณใช้งาน

Deva DIY

เช่นเดียวกับ I2C และ SPI, UART สามารถแมปไปยังขา GPIO ใด ๆ ก็ได้ แต่จะมีการกำหนดขาเริ่มต้นไว้ในบอร์ดส่วนใหญ่ โดยมีการตั้งค่าขา RX และ TX ของ UART0, UART1, และ UART2 สำหรับ ESP32 ดังนี้:

UART Port	**TX** (ส่งข้อมูล)	**RX** (รับข้อมูล)
UART0	GPIO 1	GPIO 3
UART1	GPIO 10*	GPIO 9*

UART2	GPIO 17	GPIO 16

หมายเหตุ: UART1 ใช้ขาที่เชื่อมต่อกับหน่วยความจำ SPI Flash ดังนั้นหากคุณต้mmองการใช้งาน UART1 คุณจะต้องกำหนดขาใหม่ (reassign) สำหรับการใช้งาน UART1

Strapping Pins

ชิป ESP32 มีขา **strapping pins** ที่ใช้สำหรับการเลือกโหมดการบูตหรือโหมดการแฟลช ดังนี้:

- **GPIO 0** (ต้องเป็น LOW เพื่อเข้าสู่โหมดบูต)
- **GPIO 2** (ต้องลอยขา หรือ LOW ในระหว่างการบูต)
- **GPIO 4**
- **GPIO 5** (ต้องเป็น HIGH ในระหว่างการบูต)
- **GPIO 12** (ต้องเป็น LOW ในระหว่างการบูต)
- **GPIO 15** (ต้องเป็น HIGH ในระหว่างการบูต)

ขาเหล่านี้ถูกใช้เพื่อให้ ESP32 เข้าสู่โหมดบูตหรือโหมดการแฟลช ในบอร์ดพัฒนาส่วนใหญ่ที่มี USB/Serial ในตัว คุณไม่จำเป็นต้องกังวลเกี่ยวกับสถานะของขาเหล่านี้ เพราะบอร์ดจะตั้งขาเหล่านี้ให้อยู่ในสถานะที่ถูกต้องสำหรับโหมดการแฟลชหรือโหมดบูตโดยอัตโนมัติ

อย่างไรก็ตาม หากคุณเชื่อมต่ออุปกรณ์เสริมกับขาเหล่านี้ อาจเกิดปัญหาในการอัปโหลดโค้ดใหม่หรือการแฟลช ESP32 ใหม่ หรือการรีเซ็ตบอร์ด

หากคุณเชื่อมต่ออุปกรณ์เสริมกับขา strapping pins และประสบปัญหาในการอัปโหลดโค้ดหรือแฟลช ESP32 อาจเป็นเพราะอุปกรณ์เหล่านั้นขัดขวางไม่ให้ ESP32 เข้าสู่โหมดที่ถูกต้อง คุณสามารถศึกษาข้อมูลเพิ่มเติมเกี่ยวกับ การเลือกโหมดบูต ในเอกสารการตั้งค่าโหมดบูตของ ESP32 เพื่อให้เข้าใจการตั้งค่าที่ถูกต้องได้

Pins HIGH at Boot

บางขาของ GPIO จะเปลี่ยนสถานะเป็น HIGH หรือส่งสัญญาณ PWM ในระหว่างการบูตหรือรีเซ็ต ซึ่งหมายความว่า หากคุณเชื่อมต่อขาเหล่านี้กับอุปกรณ์เอาต์พุต คุณอาจได้รับผลลัพธ์ที่ไม่คาดคิดเมื่อ ESP32 รีเซ็ตหรือบูตใหม่

ขา GPIO ที่เปลี่ยนสถานะเป็น HIGH หรือส่งสัญญาณ PWM ในระหว่างการบูต:

- **GPIO 1**
- **GPIO 3**
- **GPIO 5**
- **GPIO 6** ถึง **GPIO 11** (เชื่อมต่อกับหน่วยความจำ SPI flash ที่ติดตั้งในตัว ESP32 – ไม่แนะนำให้ใช้งาน)
- **GPIO 14**
- **GPIO 15**

Enable (EN)

ขา **Enable (EN)** เป็นขาเปิดใช้งานของตัวเรกูเลเตอร์ 3.3V ซึ่งจะดึงขึ้น (pull-up) ดังนั้น หากคุณเชื่อมต่อขานี้กับ GND จะทำให้ตัวเรกูเลเตอร์ 3.3V ถูกปิด

คุณสามารถใช้ขานี้เชื่อมต่อกับปุ่มกดเพื่อรีสตาร์ท ESP32 ได้ ตัวอย่างเช่น เมื่อกดปุ่มที่เชื่อมต่อกับขา EN จะทำให้ ESP32 รีสตาร์ทใหม่.

Deva DIY