

# OVERVIEW VỀ ESP-WROOM32

Ngày 8 tháng 1 năm 2023

## Mục lục

<b>1</b>	<b>Tổng quan về esp32</b>	<b>1</b>
1.1	ESP32 . . . . .	1
1.2	ESP-WROOM-32 Module . . . . .	1
<b>2</b>	<b>Về Module ESP-WROOM 32</b>	<b>2</b>
2.1	Yêu cầu về năng lượng . . . . .	2
2.2	Công tắc & đèn LED trên bo mạch . . . . .	3
2.3	Serial Communication . . . . .	4
<b>3</b>	<b>Các chân của esp</b>	<b>4</b>
3.1	Sơ đồ chân ESP-WROOM-32 . . . . .	5
3.2	Sơ qua về các chân của module . . . . .	6
<b>4</b>	<b>Cụ thể các chân của ESP</b>	<b>7</b>
4.1	POWER . . . . .	8
4.2	GND (Ground) . . . . .	8
4.3	GPIO (General Purpose Input/Output) . . . . .	9
4.4	PMW (Pulse Width Modulated) . . . . .	9
4.5	RTC GPIO (Real Time Clock ) . . . . .	9
4.6	ADC (Analog Digital Convert) . . . . .	9
4.7	DAC (Digital Analog Convert) . . . . .	10
4.8	TOUCH GPIO - GPIO cảm ứng điện dung . . . . .	10
4.9	Các chức năng giao tiếp . . . . .	10
4.9.1	UART (Universal Asynchronous Receive Transmit) . . . . .	10
4.9.2	SPI (Serial Peripheral Interface) . . . . .	12

4.9.3 I2C . . . . .	14
<b>5 Những GPIO ESP32 nào an toàn để sử dụng?</b>	<b>15</b>
<b>6 Tổng kết</b>	<b>19</b>

*File này Liu tổng hợp sơ qua về esp32 , chữ màu đỏ là cmt và 1 số cái Liu muốn cho vào slide*

# 1 Tổng quan về esp32

Phần này để trong 1 slide thui nhé

## 1.1 ESP32

ESP32 là một series các vi điều khiển trên một vi mạch giá rẻ, năng lượng thấp có tích hợp WiFi và dual-mode Bluetooth . Với hiệu suất và giá cả tuyệt vời , ngoài ra có thể lập trình bằng nhiều ngôn ngữ khác nhau như : C/C++, Python..., nên esp32 được sử dụng rất nhiều trong các ứng dụng khác nhau hiện nay .

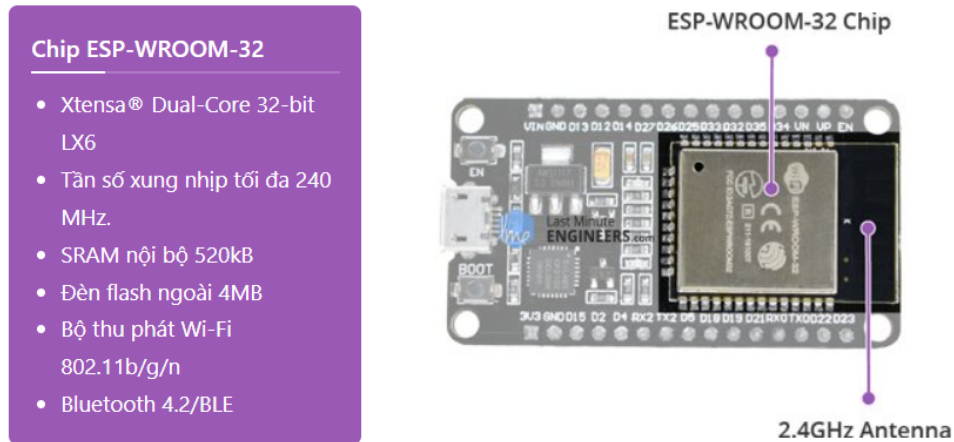


Dòng esp32

## 1.2 ESP-WROOM-32 Module

Bộ mạch phát triển trang bị cho mô-đun ESP-WROOM-32 chứa bộ vi xử lý LX6 lõi kép 32 bit Tensilica Xtensa® . Bộ xử lý này tương tự như ESP8266 nhưng có hai lõi CPU (có thể được điều khiển riêng), hoạt động ở tần số xung nhịp có thể điều chỉnh từ 80 đến 240 MHz và thực hiện ở tốc độ lên tới 600 DMIPS (Dhrystone Million Instructions Per Second).

Con chip này cũng có khả năng Bluetooth chế độ kép, nghĩa là nó hỗ trợ cả Bluetooth 4.0 (BLE/Bluetooth Smart) và Bluetooth Classic (BT) , làm cho nó trở nên linh hoạt hơn.



Module esp-wroom 32

## 2 Về Module ESP-WROOM 32

Bỏ vào slide thì ngắn gọn thôi nhe , bỏ cái hình vào với 1 ít giải thích bên cạnh thôi cũng được

Cả mục 2.1, 2.2 và 2.3 nếu ổn và dễ nhìn thì nên bỏ vào cùng 1 slide

### 2.1 Yêu cầu về năng lượng

Do dải điện áp hoạt động của ESP32 là 2,2V đến 3,6V nên bo mạch đi kèm với bộ ổn áp LDO ( low-dropout ) để giữ điện áp ổn định ở mức 3,3V.

Cấp nguồn cho bo mạch phát triển ESP32 được cung cấp qua đầu nối USB MicroB trên bo mạch .

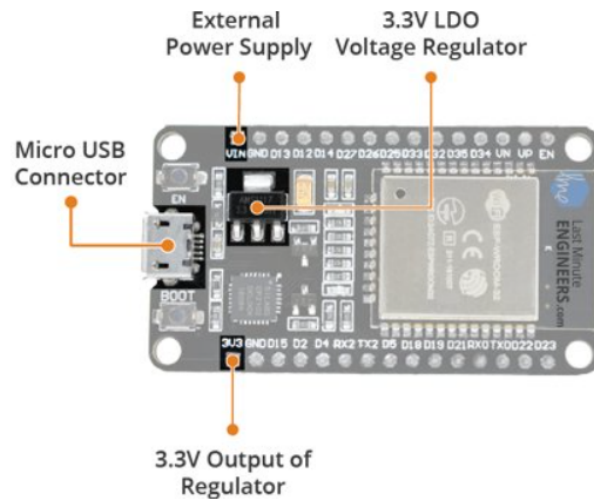
Ngoài ra, dòng sleep của chip ESP32 nhỏ hơn 5  $\mu$ A, khiến nó phù hợp với các ứng dụng điện tử đeo được và chạy bằng pin.

**Lưu ý :** ESP32 yêu cầu nguồn điện 3,3V và mức logic 3,3V để liên lạc. Các chân GPIO không được thiết kế để dùng với điện áp được 5V! Nếu muốn

kết nối bo mạch với điện áp 5V (hoặc cao hơn), cần thực hiện một số hiệu chỉnh (level shifting ).

#### Yêu cầu năng lượng

- Điện áp hoạt động: 2.2V đến 3.6V
- Bộ điều chỉnh 3.3V 600mA trên bo mạch
- 5  $\mu$ A trong Chế độ ngủ
- 250mA trong quá trình truyền RF

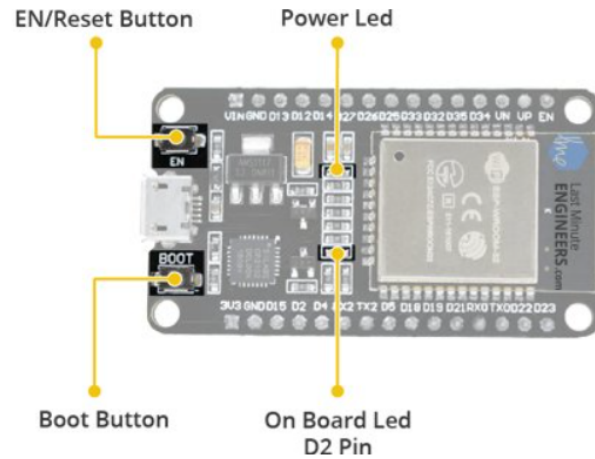


## 2.2 Công tắc & đèn LED trên bo mạch

Board ESP32 có 2 nút : một nút EN để reset và 1 nút BOOT để tải sketch/program. Board này cũng có 2 LED định vị đỏ và xanh. LED đỏ cho biết là board đã dc bật và có 3.3V từ regulator. LED xanh có thể dc lập trình và dc nối với pin D2

#### Công tắc & đèn báo

- VI – Đặt lại chip ESP32
- Khởi động – Tải xuống các chương trình mới
- Đèn LED đỏ – Đèn báo nguồn
- Đèn LED xanh lam – Người dùng có thể lập trình

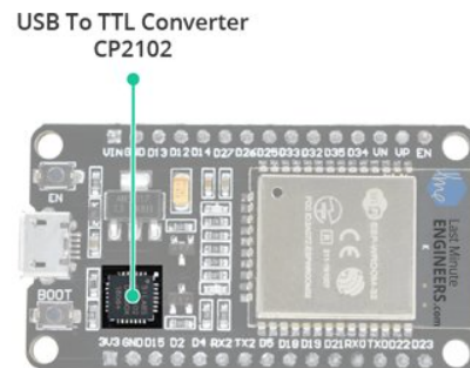


## 2.3 Serial Communication

Board này chứa CP2012 USB-to-UART Bridge Controller, chuyển tín hiệu USB sang serial và cho phép máy tính của bạn lập trình và giao tiếp với ESP 32

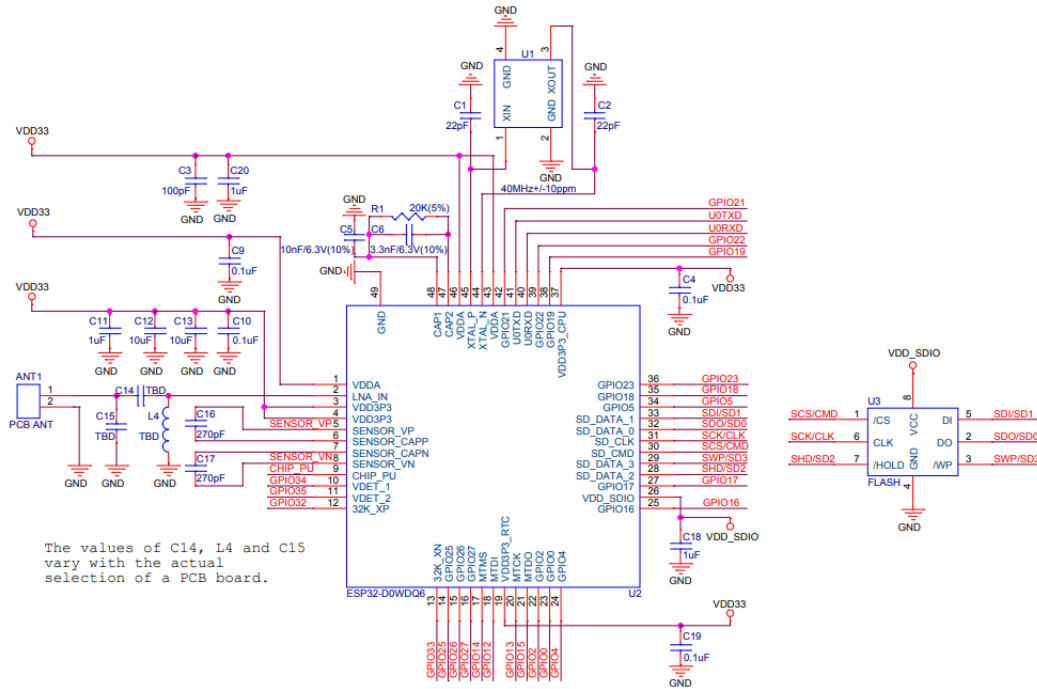
#### Serial Communication

- Bộ chuyển đổi USB sang UART CP2012
- Tốc độ truyền thông 5 Mbps
- hỗ trợ IrDA



## 3 Các chân của esp

Phần này thì chủ yếu là bỏ hình vào slide , nếu phần nội dung có giải thích gì mà hình không có thì hãy bỏ vào , mỗi hình trong 1 slide là ổn  
ESP32 có tổng cộng 48 chân GPIO



**Strapping Pins - Các chân Strapping ESP32** có 6 strapping pins: MTDI/GPIO12, GPIO0, GPIO2, GPIO4, MTDO/GPIO15, GPIO5.

Trong giai đoạn reset chip(power-on-reset, RTC watchdog reset và brownout reset), những chốt của strapping pin sẽ lấy mẫu thông tin của mức điện thế và lưu dưới dạng các bit strapping 0 và 1, và giữ bit này cho đến khi tắt chip. Những bit strapping này sẽ cấu tạo nên chế độ boot của thiết bị, điện áp hoạt động của VDDSDIO và các cài đặt ban đầu khác. Các giá trị của strapping pin này có thể thay đổi được.

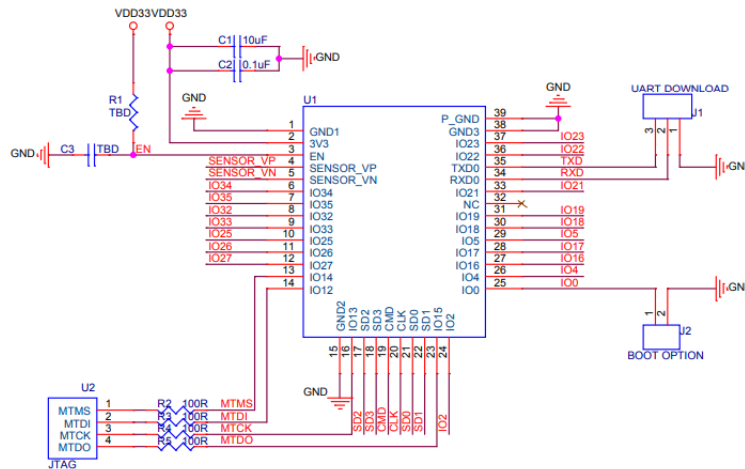
Mỗi strapping pin sẽ được nối với nội trở kéo(pull-up/pull-down) tương đương trong khi reset chip. Do đó, nếu 1 strapping pin không được nối hoặc nối với mạch có trở kháng cao, nội trở kéo sẽ mức đầu vào mặc định của các strapping pin.

### 3.1 Sơ đồ chân ESP-WROOM-32

Mặc dù ESP32 có tổng cộng 48 chân GPIO. Tuy nhiên, bo mạch dựa trên ESP32 có nhiều hình dạng và kích thước khác nhau và sơ đồ chân của mỗi bo

là khác nhau. Ngoài ra, không phải tất cả các chân của vi điều khiển ESP32 đều có sẵn trên bo dev vì một số chân có thể được gắn vĩnh viễn với một chức năng chuyên dụng

Các chân sẽ được nối đến các vị trí khác nhau của module , trong đó có 30 chân ở ngoài , và có thể kết nối với các thiết bị ngoại vi



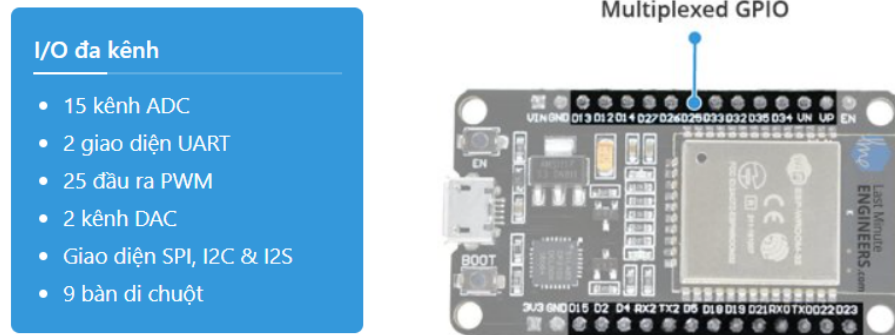
### 3.2 Sơ qua về các chân của module

Các chân được chia ra thành các đầu chân ở cả hai mặt của bảng phát triển. Các chân này có thể được gán cho tất cả các loại nhiệm vụ ngoại vi, bao gồm:

- 15 kênh ADC – 15 kênh 12-bit SAR ADC. Phạm vi ADC có thể được đặt thành 0-1V, 0-1.4V, 0-2V hoặc 0-4V
- 2 giao diện UART – 25 kênh chân PWM để làm mờ đèn LED hoặc điều khiển động cơ.
- 25 đầu ra PWM – 25 kênh chân PWM để làm mờ đèn LED hoặc điều khiển động cơ.
- 2 kênh DAC – DAC 8 bit để tạo ra tín hiệu analog.
- Giao diện SPI, I2C – để kết nối tất cả các loại cảm biến và thiết bị ngoại vi.



- 9 Touch Pad – 9 GPIO có cảm biến cảm ứng điện dung.

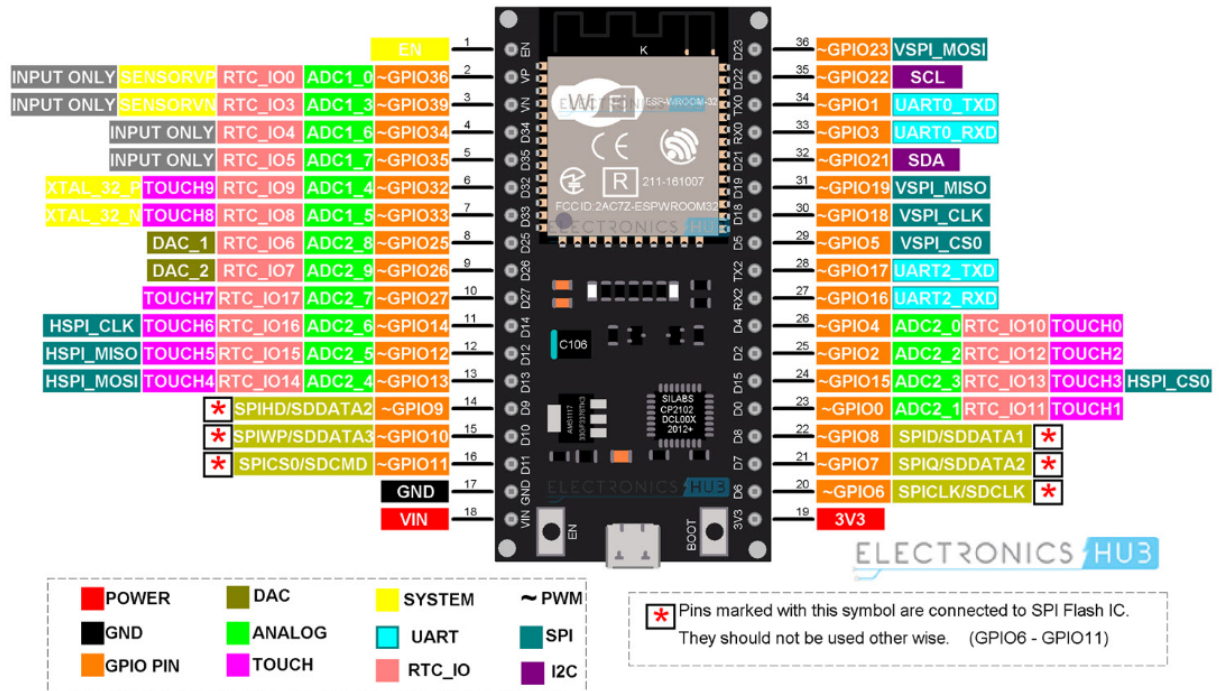


**Các chân Input Only:** D34, D35, VP, VN ko thể dc cài đặt thành output nhưng có thể dùng như analog input, digital input, hoặc 1 vài mục đích khác. Các pin này ko có nội trở kéo như các pin GPIO.

VP, VN là 1 phần quan trọng của "bộ tiền khuếch đại điều chỉnh âm siêu thấp", dùng để chỉnh khoảng thời gian và độ ồn mà bộ tiền khuếch đại nhận.

## 4 Cụ thể các chân của ESP

Phần này là quan trọng nhất, nên cố gắng cho vào slide ngắn gọn, nhưng đủ ý xịu nhé :>> Hình dưới là sơ đồ chân của bo 30 chân ESP32.



### 4.3 GPIO (General Purpose Input/Output)

Thiết bị ngoại vi được sử dụng phổ biến nhất là GPIO. Bạn có thể cấu hình chân dưới dạng GPIO hoặc ADC hoặc UART trong chương trình. Bảng phát triển ESP32 có 25 chân GPIO có thể được gán cho các chức năng khác nhau theo chương trình. Mỗi GPIO được kích hoạt kỹ thuật số có thể được cấu hình để kéo lên hoặc kéo xuống bên trong hoặc đặt thành trở kháng cao. Khi được định cấu hình làm đầu vào, nó cũng có thể được đặt thành kích hoạt cạnh hoặc kích hoạt mức để tạo ngắt CPU.

Các chân ADC và DAC được xác định trước và bạn phải sử dụng các chân do nhà sản xuất chỉ định. Nhưng các chức năng khác như PWM, SPI, UART, I2C, ... có thể được gán cho bất kỳ chân GPIO nào thông qua chương trình.

Tất cả các chân GPIO đều có khả năng ngắt.

Chân có thể sử dụng là các chân từ 2-13 , và 18-30

### 4.4 PMW (Pulse Width Modulated)

Điều chỉnh điện áp ra tải, hay nói cách khác, là phương pháp điều chế dựa trên sự thay đổi độ rộng của chuỗi xung vuông, dẫn đến sự thay đổi điện áp ra.

Có thể điều chỉnh ở tất cả các chân GPIO

### 4.5 RTC GPIO (Real Time Clock )

Đồng hồ thời gian thực ( RTC ) dùng để đo thời gian trôi qua , RTC có mặt trong hầu hết mọi thiết bị điện tử cần giữ thời gian chính xác trong ngày .

RTC được sử dụng để tránh nhầm lẫn với đồng hồ phần cứng thông thường chỉ là tín hiệu điều khiển thiết bị điện tử kỹ thuật số và không tính thời gian theo đơn vị con người.

Chân có thể sử dụng là các chân từ 2-13 , và 18-20

### 4.6 ADC (Analog Digital Convert)

ESP32 có hai module chuyển đổi analog sang kỹ thuật số SAR 12 bit với 8 kênh và 10 kênh mỗi module. Vì vậy, khối ADC1 và ADC2 kết hợp với nhau có 18 kênh ADC 12-bit.

Với độ phân giải 12-bit, các giá trị ở đầu ra sẽ nằm trong khoảng từ 0 - 4093.

Chân có thể sử dụng là các chân từ 2-13 , và 18-20

## 4.7 DAC (Digital Analog Convert)

Bộ vi điều khiển ESP32 có hai kênh chuyển đổi kỹ thuật số sang analog 8 bit độc lập để chuyển đổi các giá trị kỹ thuật số sang tín hiệu điện áp analog.

Hai chân GPIO sau được liên kết với các chức năng của DAC :  
DAC1 (chân 8) và DAC2 (chân 9)

## 4.8 TOUCH GPIO - GPIO cảm ứng điện dung

SoC ESP32 có 10 GPIO cảm ứng điện dung, có thể phát hiện các biến thể về điện dung trên chân cảm do chạm hoặc tiếp cận chân GPIO bằng ngón tay hoặc bút stylus. Chúng ta đều biết con người và các loài động vật đều có dòng điện sinh học trong người(tuy nhiên rất nhỏ), đầu vào Touch pin sẽ được khuếch đại tín hiệu để có thể dò được sự thay đổi (chạm) trên chân đó. Sau đó quy đổi thành giá trị có thể đo lường và lập trình được.

Các chân cảm ứng ESP32 có thể được sử dụng để đánh thức ESP32 khỏi Sleep Mode

## 4.9 Các chức năng giao tiếp

Lưu ý cái này bro chỉ cần cho vào slide hình ảnh , với các đường hoạt động - như nội dung ở dưới đây là ổn , còn so sánh với điểm mạnh điểm yếu thì để lúc thuyết trình t nói cũng được

### 4.9.1 UART (Universal Asynchronous Receive Transmit)

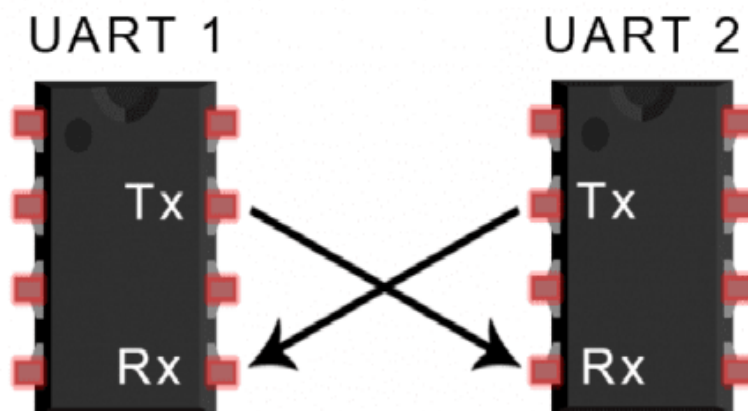
UART (Universal Asynchronous Receiver-Transmitter – Bộ truyền nhận dữ liệu nối tiếp bất đồng bộ) là một trong những giao thức truyền thông giữa thiết bị với thiết bị được sử dụng nhiều nhất. Bạn có thể thấy giao tiếp UART được sử dụng nhiều trong các ứng dụng để giao tiếp với các module như: Wifi, Bluetooth, Xbee, module đầu đọc thẻ RFID với Raspberry Pi, Arduino hoặc vi điều khiển khác. Đây cũng là chuẩn giao tiếp thông dụng và phổ biến trong công nghiệp từ trước đến nay.

Trong giao tiếp UART, hai UART giao tiếp trực tiếp với nhau. UART truyền chuyển đổi dữ liệu song song từ một thiết bị điều khiển như CPU thành dạng nối tiếp, truyền nó nối tiếp đến UART nhận, sau đó chuyển đổi dữ liệu nối tiếp trở lại thành dữ liệu song song cho thiết bị nhận.

UART truyền dữ liệu không đồng bộ, có nghĩa là không có tín hiệu đồng hồ để đồng bộ hóa đầu ra của các bit từ UART truyền đến việc lấy mẫu các bit bởi UART nhận.

Hai đường dây mà mỗi thiết bị UART sử dụng để truyền dữ liệu đó là:

- Transmitter (Tx) - chân 22 và chân 28
- Receiver (Rx) - chân 21 và 27



#### Ưu điểm :

- Chỉ sử dụng hai dây
- Không cần tín hiệu clock
- Có một bit chẵn lẻ để cho phép kiểm tra lỗi
- Cấu trúc của gói dữ liệu có thể được thay đổi miễn là cả hai bên đều được thiết lập cho nó
- Phương pháp có nhiều tài liệu và được sử dụng rộng rãi

#### Nhược điểm :

- Kích thước của khung dữ liệu được giới hạn tối đa là 9 bit
- Không hỗ trợ nhiều hệ thống slave hoặc nhiều hệ thống master
- Tốc độ truyền của mỗi UART phải nằm trong khoảng 10 % của nhau

#### 4.9.2 SPI (Serial Peripheral Interface)

SPI là một giao thức giao tiếp phổ biến được sử dụng bởi nhiều thiết bị khác nhau.

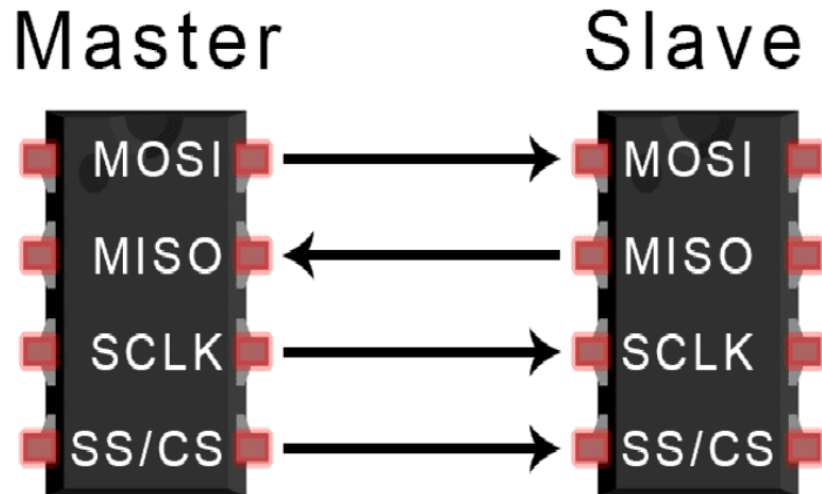
Lợi ích duy nhất của SPI là dữ liệu có thể được truyền mà không bị gián đoạn. Bất kỳ số lượng bit nào cũng có thể được gửi hoặc nhận trong một luồng liên tục. Với I2C và UART, dữ liệu được gửi dưới dạng gói, giới hạn ở một số bit cụ thể. Điều kiện bắt đầu và dừng xác định điểm bắt đầu và kết thúc của mỗi gói, do đó dữ liệu bị gián đoạn trong quá trình truyền.

Các thiết bị giao tiếp qua SPI có quan hệ master - slave. Master là thiết bị điều khiển (thường là vi điều khiển), còn slave (thường là cảm biến, màn hình hoặc chip nhớ) nhận lệnh từ master. Cấu hình đơn giản nhất của SPI là hệ thống một slave, một master duy nhất, nhưng một master có thể điều khiển nhiều hơn một slave.

SPI có thể chia thành 2 loại, tuy nhiên không có sự khác nhau nhiều giữa chúng, có thể sự khác nhau nằm ở tốc độ khi thực hiện.

Hoạt động của SPI gồm :

- MOSI (đầu ra master / đầu vào slave) - đường truyền cho master gửi dữ liệu đến slave. Chân được sử dụng là chân 13 (HSPI) và 30 (VSPI)
- MISO (đầu vào master / đầu ra slave) - đường cho slave gửi dữ liệu đến master. Chân được sử dụng là chân 12 (HSPI) và 25 (VSPI)
- SCLK (clock) - đường cho tín hiệu xung nhịp. Chân được sử dụng là chân 11 (HSPI) và 24 (VSPI)
- SS / CS (Slave Select / Chip Select) - đường cho master chọn slave nào để gửi tín hiệu. Chân được sử dụng để chọn Slave CS là chân 18 (HSPI) và 23 (VSPI)

**Ưu điểm :**

- Không có bit bắt đầu và dừng, vì vậy dữ liệu có thể được truyền liên tục mà không bị gián đoạn
- Không có hệ thống định địa chỉ slave phức tạp như I2C
- Tốc độ truyền dữ liệu cao hơn I2C (nhỏ gấp đôi)
- Các đường MISO và MOSI riêng biệt, vì vậy dữ liệu có thể được gửi và nhận cùng một lúc

**Nhược điểm :**

- Sử dụng bốn dây (I2C và UART sử dụng hai)
- Không xác nhận dữ liệu đã được nhận thành công (I2C có điều này)
- Không có hình thức kiểm tra lỗi như bit chẵn lẻ trong UART
- Chỉ cho phép một master duy nhất

### 4.9.3 I2C

I2C kết hợp các tính năng tốt nhất của SPI và UART. Với I2C, bạn có thể kết nối nhiều slave với một master duy nhất (như SPI) và bạn có thể có nhiều master điều khiển một hoặc nhiều slave. Điều này thực sự hữu ích khi bạn muốn có nhiều hơn một vi điều khiển ghi dữ liệu vào một thẻ nhớ duy nhất hoặc hiển thị văn bản trên một màn hình LCD.

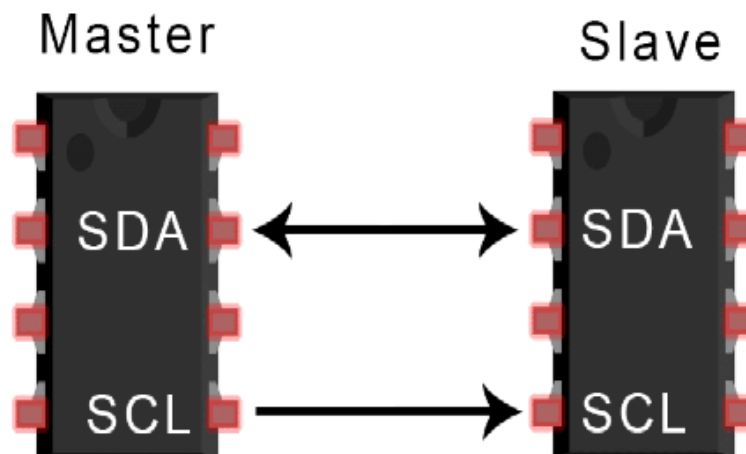
Giống như giao tiếp UART, I2C chỉ sử dụng hai dây để truyền dữ liệu giữa các thiết bị:

- SDA (Serial Data) - đường truyền cho master và slave để gửi và nhận dữ liệu. Chân được sử dụng là chân 32
- SCL (Serial Clock) - đường mang tín hiệu xung nhịp. Chân được sử dụng là chân 35

I2C là một giao thức truyền thông nối tiếp, vì vậy dữ liệu được truyền từng bit dọc theo một đường duy nhất (đường SDA).

Giống như SPI, I2C là đồng bộ, do đó đầu ra của các bit được đồng bộ hóa với việc lấy mẫu các bit bởi một tín hiệu xung nhịp được chia sẻ giữa master và slave. Tín hiệu xung nhịp luôn được điều khiển bởi master.

Tất cả các SPI cũng có thể được sử dụng để kết nối với Flash/SRAM và LCD bên ngoài.





**Ưu điểm :**

- Chỉ sử dụng hai dây
- Hỗ trợ nhiều master và nhiều slave
- Bit ACK / NACK xác nhận mỗi khung được chuyển thành công
- Phần cứng ít phức tạp hơn so với UART
- Giao thức nổi tiếng và được sử dụng rộng rãi

**Nhược điểm**

- Tốc độ truyền dữ liệu chậm hơn SPI
- Kích thước của khung dữ liệu bị giới hạn ở 8 bit
- Cần phần cứng phức tạp hơn để triển khai so với SPI

## 5 Những GPIO ESP32 nào an toàn để sử dụng?

Vì ESP32 có nhiều chân với các chức năng cụ thể nên chúng có thể không phù hợp với các dự án bất kỳ . Bảng sau đây cho biết những chân nào an toàn để sử dụng và những chân nào cần được chú ý nhiều hơn trước khi sử dụng chúng .

✔ – Ghim ưu tiên đầu tiên. Chúng hoàn toàn tốt để sử dụng.

⚠ – Hãy chú ý vì hành vi của chúng có thể không đoán trước được, chủ yếu là trong quá trình khởi động. Đừng sử dụng chúng trừ khi thực sự cần.

✖ – Không nên sử dụng các chân này. Vì vậy, tránh chúng.

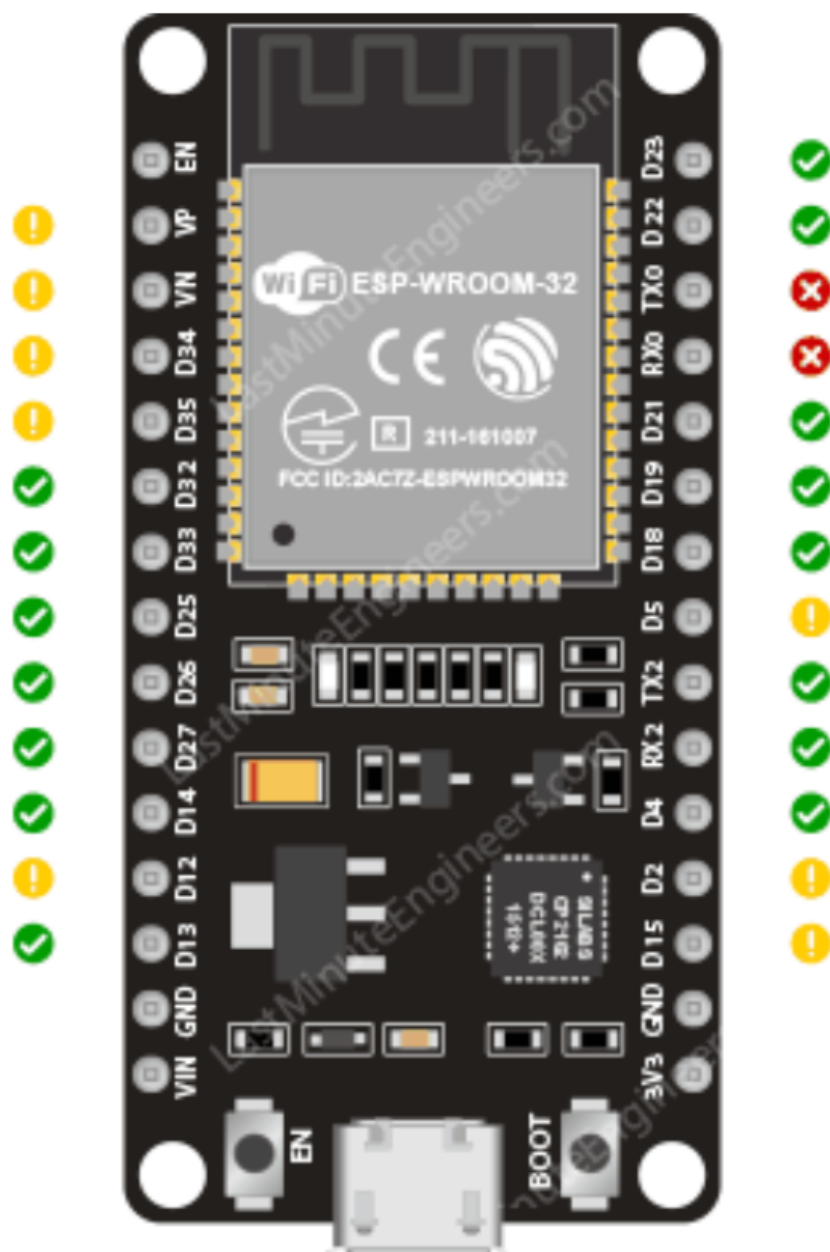
Nếu bro thấy bảng này dài quá , hay khó bỏ slide gì đó thì có thể tự chụp , link Liu chụp bảng này ở đây nhé : <https://lastminuteengineers.com/esp32-pinout-reference/>

Bảng :

Nhãn	GPIO	An toàn để sử dụng?	Lý do
D0	0	⚠	phải CAO trong khi khởi động và THẤP để lập trình
TX0	1	✗	Chân Tx, được sử dụng để flash và gỡ lỗi
D2	2	⚠	phải ở mức THẤP trong khi khởi động và cũng được kết nối với đèn LED trên bo mạch
RX0	3	✗	Chân Rx, được sử dụng để flash và gỡ lỗi
D4	4	✓	
D5	5	⚠	phải ở mức CAO trong khi khởi động
D6	6	✗	Đã kết nối với bộ nhớ Flash
D7	7	✗	Đã kết nối với bộ nhớ Flash
D8	số 8	✗	Đã kết nối với bộ nhớ Flash
D9	9	✗	Đã kết nối với bộ nhớ Flash
D10	10	✗	Đã kết nối với bộ nhớ Flash
D11	11	✗	Đã kết nối với bộ nhớ Flash
D12	12	⚠	phải ở mức THẤP trong khi khởi động
D13	13	✓	
D14	14	✓	
D15	15	⚠	phải ở mức CAO trong khi khởi động, ngăn nhậ ký khởi động nếu kéo xuống THẤP
RX2	16	✓	
TX2	17	✓	
D18	18	✓	
D19	19	✓	
D21	21	✓	
D22	22	✓	
D23	23	✓	
D25	25	✓	

D26	26	✓	
D27	27	✓	
D32	32	✓	
D33	33	✓	
D34	34	!	Chỉ đầu vào GPIO, không thể được định cấu hình làm đầu ra
D35	35	!	Chỉ đầu vào GPIO, không thể được định cấu hình làm đầu ra
VP	36	!	Chỉ đầu vào GPIO, không thể được định cấu hình làm đầu ra
VN	39	!	Chỉ đầu vào GPIO, không thể được định cấu hình làm đầu ra

Để thuận tiện, dưới đây là hình ảnh hiển thị chân GPIO nào an toàn để sử dụng :



## 6 Tổng kết

Sau ctm quá trình chạy dl từ 8h tới cmn 3h đêm của Liu , và còn rất nhiều dl khác đang chờ được thực hiện éc ô éc =)))

Nói chung là nội dung bỏ vào slide hầu như là hình , Liu cop nhiều nội dung để bro đọc , hiểu và chọn ý bỏ vào slide , chứ cố gắng bỏ không cần nhiều chữ đâu . Cố gắng xong nhé :))

**Thân ái và quyết thắng !**