# UART

## [CẤU TRÚC VÀ CÁC THÀNH PHẦN TRONG](https://nguyenquanicd.blogspot.com/2017/06/cancontroller-area-netwwork-cau-truc-va.html) UART

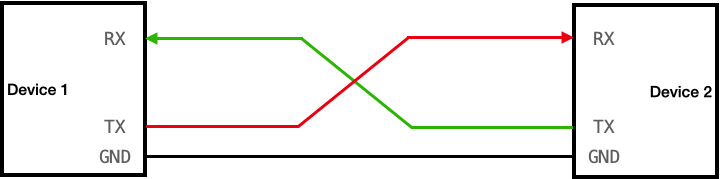
### UART là gì ?

UART (Universal Asynchronous Receiver-Transmitter – Bộ truyền nhận dữ liệu không đồng bộ) là một giao thức truyền thông phần cứng dùng giao tiếp nối tiếp không đồng bộ và có thể cấu hình được tốc độ truyền. Mục đích chính của UART là truyền và nhận dữ liệu nối tiếp.

Chuẩn giao tiếp UART sử dụng 2 dây để truyền và nhận dữ liệu giữa các thiết bị:

1. TX (Transmiter) – Dây truyền dữ liệu
2. RX (Receiver) – Dây nhận dữ liệu

### Cấu trúc phần cứng



**Hình 1.** Cấu trúc phần cứng UART

Chuẩn giao tiếp UART hoạt động theo cơ chế truyền dữ liệu bất đồng bộ, tức là không có một tín hiệu xung clock chung giữa hai thiết bị để đồng bộ việc truyền nhận dữ liệu. Cả hai UART cũng phải được cấu hình để truyền và nhận cùng một cấu trúc gói dữ liệu. Nghĩa là khi thiết lập ở vi điều khiển, các cấu hình tốc độ baud phải giống nhau.

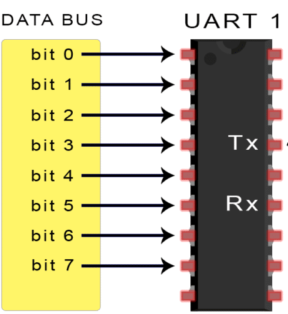
|  |
| --- |
| **Lưu ý**:  GND của hai thiết bị phải được kết nối với nhau. |

|  |  |
| --- | --- |
| Số lượng dây sử dụng | 2 |
| Tốc độ | Từ 9600 bps -> 115200 bps |
| Phương thức truyền dữ liệu | Không đồng bộ |
| Kiểu truyền dữ liệu | Nối tiếp |
| Số lượng Master (thiết bị chủ) | 1 |
| Số lượng Slave (thiết bị tớ) | 1 |

|  |
| --- |
| **Lưu ý**:  Bps: là tốc độ truyền dữ liệu được định nghĩa là khối lượng truyền dữ liệu trên mỗi đơn vị thời gian. Ký hiệu: bit / giây (bit/s) hoặc byte/ giây (B/s) |

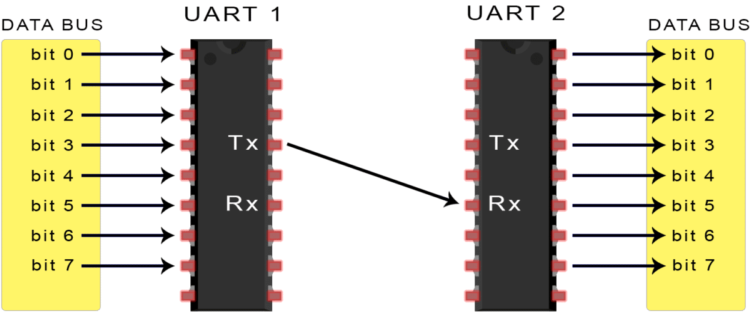
## CẤU TRÚC KHUNG TRUYỀN DỮ LIỆU VÀ KHUNG NHẬN DỮ LIỆU

### Nguyên lý hoạt động



**Hình 1.2:** Truyền dữ liệu từ DATA BUS đến UART

Dữ liệu sẽ được chuyển từ Bus dữ liệu đến UART1 truyền ở dạng song song. Sau khi nhận được dữ liệu, lúc này nó sẽ thêm một bit start, một bit chẵn lẻ và một bit stop để tạo thành gói dữ liệu (packet data).



Gói dữ liệu được truyền ra từng bit nối tiếp tại chân Tx của UART 1 và UART 2 sẽ nhận gói dữ liệu từng bit tại chân Rx của nó. Sau khi nhận xong dữ liệu, UART 2 chuyển đổi dữ liệu trở lại dạng song song và loại bỏ bit start, bit chẵn lẻ và bit stop.

Uart truyền dữ liệu nối tiếp, theo 1 trong 3 chế độ:

* Simplex: Chỉ tiến hành giao tiếp một chiều.
* Half duplex: Dữ liệu sẽ đi theo một hướng tại 1 thời điểm.
* Full duplex: Thực hiện truyền và nhận dữ liệu từ master và slave tại 1 thời điểm.

#### Full duplex

Full duplex hay còn gọi là Song Công: Là quá trình truyền ***và*** nhận dữ liệu từ master và slave tại 1 thời điểm. Giao thức UART có 2 đường truyền dữ liệu riêng biệt, cho phép dữ liệu truyền nhận theo 2 hướng cùng một thời điểm.

A diagram of a link between two lines

Description automatically generated

**Hình 1.4:** Quá trình Full duplex

#### Half duplex

Half duplex hay còn gọi là Bán Song Công: Là quá trình truyền ***hoặc*** nhận dữ liệu từ master và slave tại 1 thời điểm.

A diagram of a link between two lines

Description automatically generated

**Hình 1.4:** Quá trình Half duplex

#### Simplex:

Simplex hay còn gọi là Đơn Công: Là quá trình truyền dữ liệu theo một hướng duy nhất trên một đường truyền đã xác định.

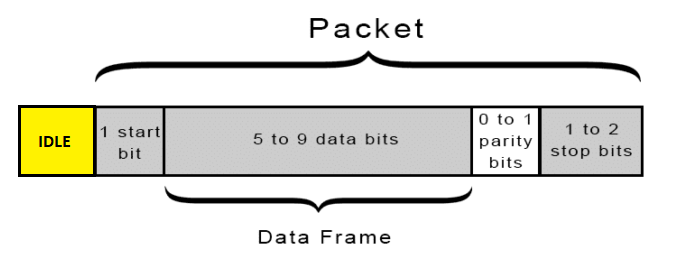
A close-up of a line

Description automatically generated

**Hình 1.5:** Quá trình Simplex

### Khung truyền dữ liệu

Dữ liệu được truyền trong giao tiếp UART được tổ chức thành các gói (Packets). Mỗi Packets chứa 1 **bit** **Start**, 5 đến 9 bit dữ liệu (tùy thuộc vào UART), 1 **bit** **Parity**và 1 hoặc 2 **bit Stop.**



**Hình 1.6:** Khung truyền của giao tiếp UART

* **IDLE** (Trạng thái bus rảnh)
* Là không có sự truyền/nhận dữ liệu trên bus.
* Giá trị của IDLE sẽ luôn luôn ở mức cao để phát hiện điều kiện bắt đầu của **Start bit**
* **Start bit** (Bit khởi đầu):
* Đường truyền dữ liệu trong giao tiếp UART thường được giữ ở mức điện áp cao(IDLE) khi nó không truyền dữ liệu.
* Để bắt đầu truyền dữ liệu, UART truyền sẽ kéo đường truyền từ mức cao xuống mức thấp trong một chu kỳ đồng hồ.
* Khi UART 2 phát hiện sự chuyển đổi điện áp cao xuống thấp, nó bắt đầu đọc các bit trong khung dữ liệu ở tần số của tốc độ truyền (Baud rate).
* **Data bits** (Bit dữ liệu) [0-8]:
* Khung dữ liệu chứa dữ liệu thực tế đang được truyền. Nó có thể dài từ 5 bit đến 8 bit nếu sử dụng **bit Parity**(bit chẵn lẻ).
* Nếu không sử dụng **bit** **Parity**, khung dữ liệu có thể dài 9 bit. Trong hầu hết các trường hợp, dữ liệu được gửi với [**bit LSB**](https://vi.wikipedia.org/wiki/Byte)(bit có trọng số thấp nhất) trước tiên.
* Nếu muốn gửi dữ liệu thì phải gửi 1 chuỗi bit. Tại 1 thời điểm chỉ gửi được 1 bit data. Nếu gửi 1 byte thì gửi 8 lần.
* **Parity bit** (Bit chẵn lẻ)
  + Dùng để kiểm tra dữ liệu truyền có đúng không (một cách tương đối). Có 2 loại parity là parity chẵn (even parity) và parity lẻ (odd parity).
  + Nếu **bit Parity** là 0 (chẵn), thì tổng các bit 1 trong khung dữ liệu phải là một số chẵn.
  + Nếu **bit Parity** là 1 (lẻ), thì tổng các bit 1 trong khung dữ liệu sẽ là một số lẻ.
  + Parity bit không phải là bit bắt buộc và vì thế nó có thể có hoặc không.

|  |
| --- |
| **Thí dụ:**  Nếu dữ liệu của bạn là 10111011 nhị phân, có tất cả 6 bit có giá trị “1” trong dữ liệu này, nếu quy định parity chẵn được dùng, bit parity sẽ mang giá trị 0 để đảm bảo tổng các số “1” là số chẵn (6 số 1). Nếu parity lẻ được yêu cầu thì giá trị của parity bit là 1. Sau khi truyền chuỗi dữ liệu kèm theo cả bit parity trên, bên nhận thu được và kiểm tra lại tổng số số “1” (bao gồm cả bit parity), nếu vi phạm quy định parity đã đặt trước thì ta khẳng định là dữ liệu nhận được là sai, còn nếu không vi phạm thì cũng không khẳng định được điều gì (mang tính tương đối). |

* (Stop bit) Bit dừng
* Để báo hiệu sự kết thúc của gói dữ liệu, UART gửi sẽ điều khiển đường truyền dữ liệu từ điện áp thấp đến điện áp cao trong ít nhất hai khoảng thời gian bit.

### Khung nhận dữ liệu của giao thức UART

A diagram of a data analysis

Description automatically generated

Bên nhận sẽ đọc liên tục đường **IDLE** của RX. Khi phát hiện sự thay đổi logic từ 1 xuống 0 thì **Start bit** được đọc và bắt đầu nhận dữ liệu. **LSB** nhận trước.

A diagram of a diagram

Description automatically generated

Delay ½ chu kì và tiếp tục delay 1 chu kì và bắt đầu lấy mẫu. Bởi vì dữ liệu ở giữa ổn định nhất. Cứ thế sau 1 chu kì lại đọc tiếp.

**Chú ý:** Truyền nhận phải cùng tốc độ Baus bởi vì truyền bất đồng bộ.

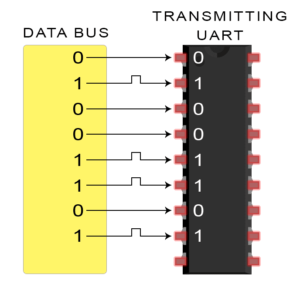
A table with numbers and a few lines

Description automatically generated with medium confidence

### ****Các bước truyền dữ liệu trong chuẩn giao tiếp UART****

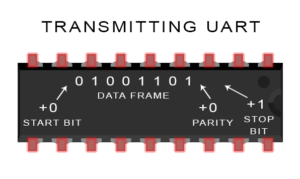
* ***Bước 1:*** UART truyền nhận dữ liệu song song từ bus dữ liệu

Dữ liệu sẽ là song song ở đầu vào. Sau đó sẽ được chuyển thành nối tiếp để truyền trên dây TX. Tốc độ baud thiết lập từ trước ở phần cấu hình.



**Hình 1.7:** Đầu vào của UART (truyền song song)

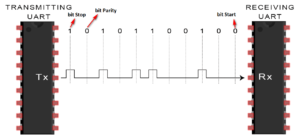
* **Bước 2:** UART truyền thêm bit thành phần như: Start bit, Parity bit và Stop bit vào khung truyền để chuẩn bị cho việc truyền dữ liệu.



**Hình 1.7:** Đưa thêm các bit điều kiện vào khung truyền

* **Bước 3:** Toàn bộ gói tin được gửi nối tiếp từ UART truyền đến UART nhận. UART nhận lấy mẫu đường dữ liệu ở tốc độ truyền được định cấu hình trước:

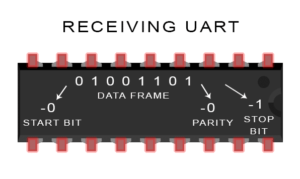
Trong quá trình này các bit trong khung truyền lần lượt được đẩy lên đường dây TX của UART1 (truyền) và đưa tới dây RX của UART2 (nhận).



**Hình 1.7:** Quá trình truyền dữ liệu trong giao tiếp UART

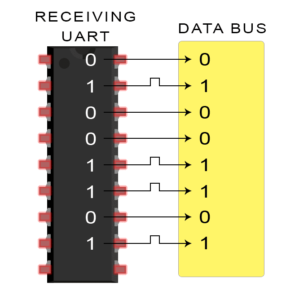
* **Bước 4**: UART nhận loại bỏ bit bắt đầu, bit chẵn lẻ và bit dừng khỏi khung dữ liệu:

Quá trình này sẽ bỏ các bit điều kiện đi, sau đó chỉ còn Data Frame (khung dữ liệu). Cuối cùng dữ liệu cần truyền sẽ tiếp tục được xử lý trong Slave.



**Hình 1.7:** Loại bỏ các bit điều kiện trong khung truyền

* **Bước 5:** UART nhận chuyển đổi dữ liệu nối tiếp trở lại thành song song và chuyển nó đến data Bus



**Hình 1.7:** Chuyển dữ liệu nối tiếp trở lại song song

Lúc này việc truyền dữ liệu qua giao tiếp UART đã hoàn thành.