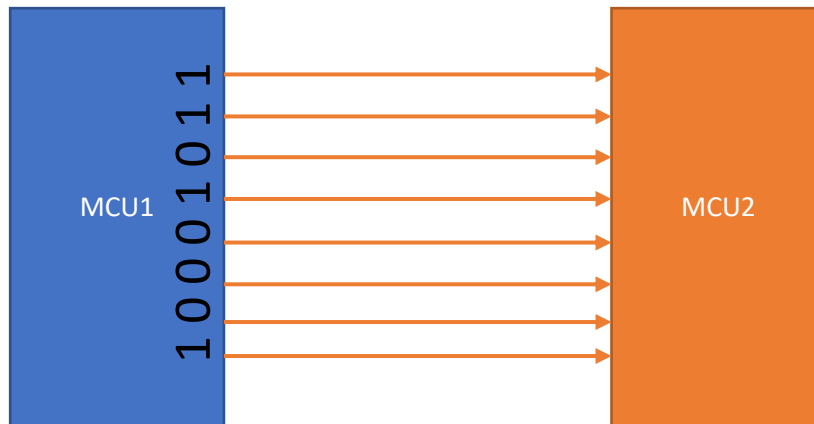


Trong một dự án lớn, một board mạch STM32 không thể đảm nhiệm xử lý hết được các yêu cầu. Cho nên chúng ta cần chia các tác vụ riêng biệt để mỗi board mạch xử lý một tác vụ. Như vậy cần có sự thống nhất và giao tiếp giữa các board mạch. Trong quá trình làm việc thì các board mạch cần trao đổi dữ liệu cho nhau. Ví dụ: Khối điều khiển trung tâm truyền lệnh đến các vi điều khiển và nhận dữ liệu từ các vi điều khiển đó về xử lý. Cách thức truyền nhận dữ liệu gồm có 2 loại là truyền thông nối tiếp và truyền thông song song.

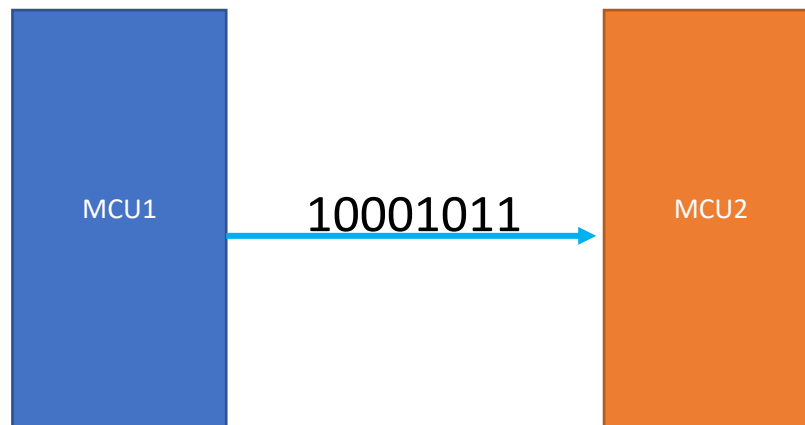
Truyền thông song song:



Trong truyền thông song song, mỗi line sẽ chịu trách nhiệm truyền/nhận 1 bit dữ liệu.

- Ưu điểm: Truyền nhiều bit cùng 1 lúc. Tốc độ truyền nhanh.
- Nhược điểm: Kết nối cồng kềnh do phải kết nối nhiều dây.

Truyền thông nối tiếp:



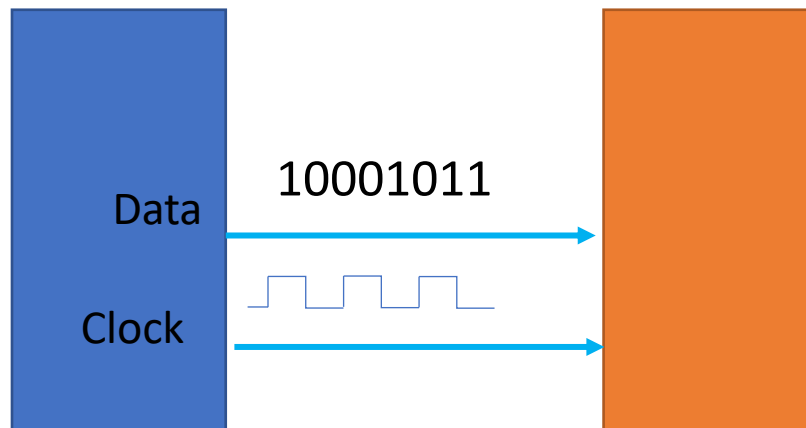
Truyền thông nối tiếp giải quyết vấn đề của truyền thông song song, dữ liệu sẽ được truyền từng bit trên 1 (hoặc một ít) đường truyền.

- Ưu điểm: Kết nối đơn giản.

- Nhược điểm: Tốc độ truyền chậm hơn so với truyền thông song song.

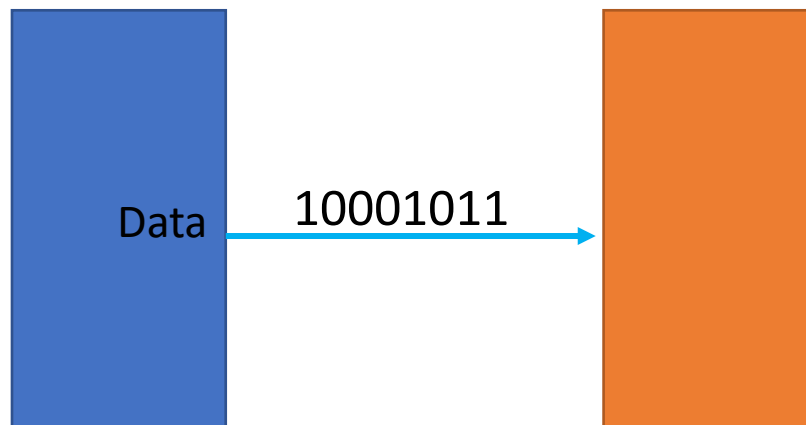
Trong thực tế thường dùng truyền thông nối tiếp.

Truyền thông đồng bộ:



Truyền thông nối tiếp đồng bộ để chỉ sự báo trước trong quá trình truyền. MCU1 kết nối với MCU2 bởi 2 đường, một đường dữ liệu và một đường xung nhịp. Mỗi lần MCU1 gửi đi 1 bit dữ liệu thì MCU1 điều khiển đường xung nhịp chuyển từ mức LOW lên mức HIGH báo cho MCU2 sẵn sàng nhận 1 bit dữ liệu. Bằng cách này tất cả các bit dữ liệu có thể truyền/nhận dễ dàng với ít rủi ro trong quá trình truyền.

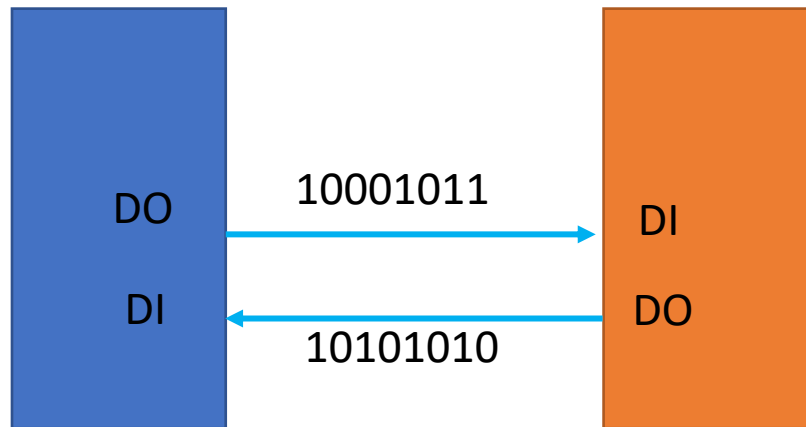
Truyền thông nối tiếp không đồng bộ:



Truyền thông nối tiếp không đồng bộ chỉ cần một đường truyền cho một quá trình. Khung dữ liệu đã được chuẩn hóa bởi các thiết bị nên không cần đường xung nhịp báo trước dữ liệu đến. Ví dụ 2 MCU đang giao tiếp với nhau theo phương pháp này, chúng đã được thỏa thuận với nhau rằng cứ 1ms thì sẽ

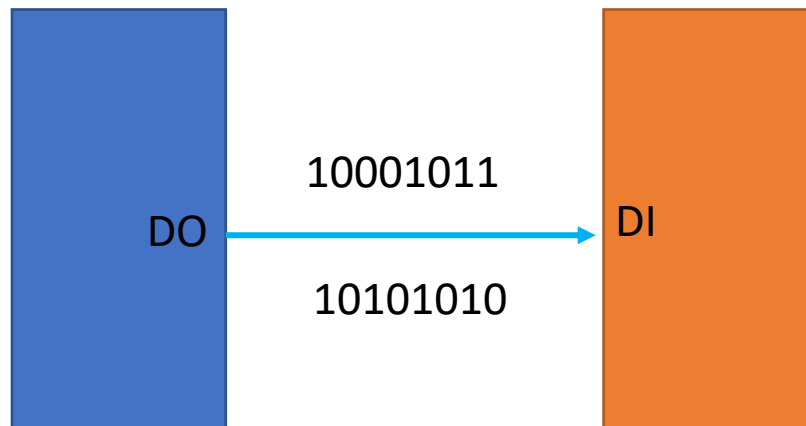
có 1 bit dữ liệu truyền đến. Tuy nhiên, để quá trình truyền thành công thì việc tuân thủ các tiêu chuẩn truyền là hết sức quan trọng.

Giao tiếp song công:



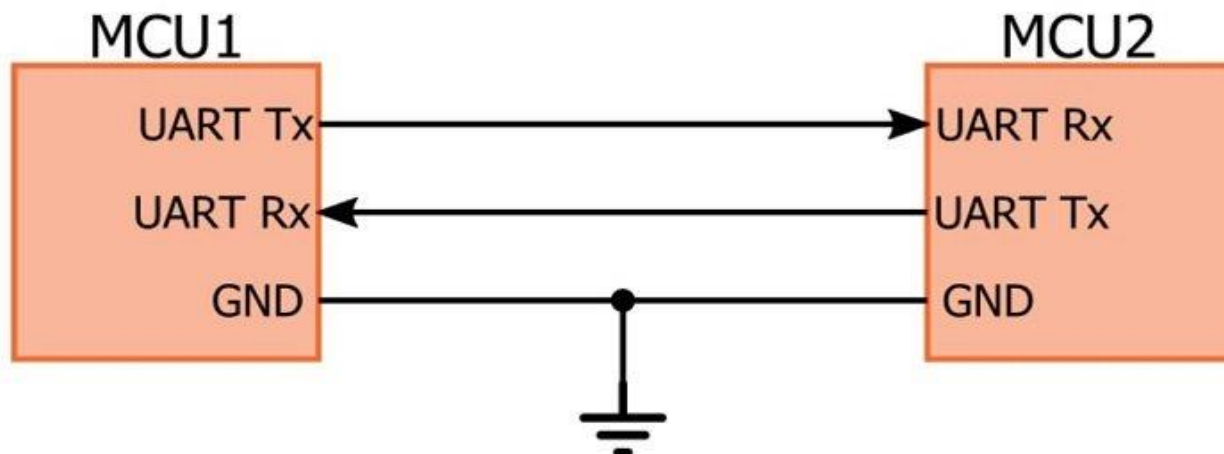
Giao tiếp song công có thể truyền/nhận dữ liệu cùng 1 thời điểm. Có 2 đường dây truyền/nhận dữ liệu riêng biệt.

Giao tiếp bán song công:



Với giao tiếp bán song công, trong một thời điểm chỉ có thể truyền hoặc nhận dữ liệu. Chỉ có 1 đường dây dùng để truyền/nhận dữ liệu.

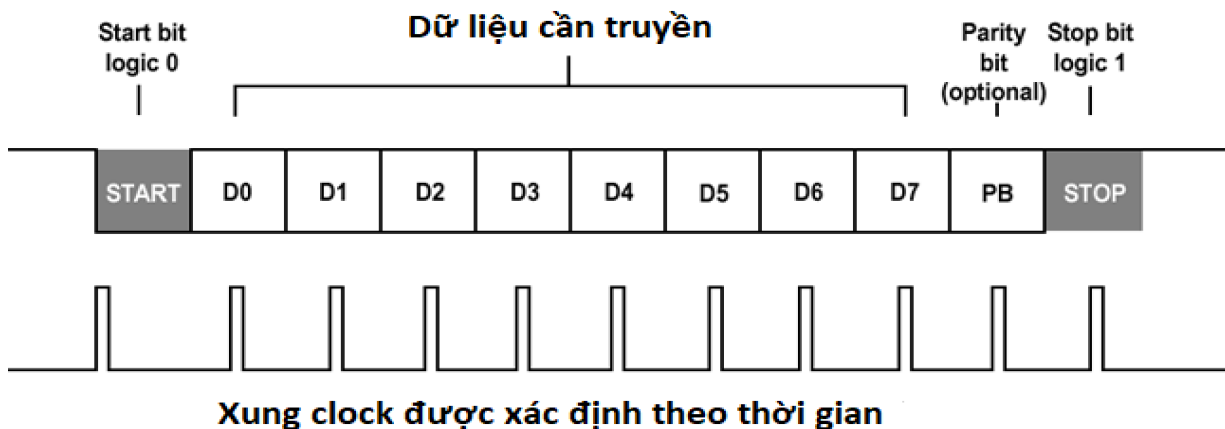
UART: Universal Asynchronous serial Receiver and Transmitter là chuẩn giao tiếp truyền nhận nối tiếp không đồng bộ.



Các khái niệm trong UART:

- Baud rate (tốc độ baud): số bit truyền nhận trong 1s, tốc độ hay sử dụng phải là bội của 8 thường là 2400, 4800, 9600, 19200, 38400, 56000, 115200... cả 2 thiết bị truyền nhận cho nhau phải cùng 1 tốc độ baud.
- Frame (khung truyền): Khung truyền quy định về số bit trong mỗi lần truyền. Thông thường chúng ta sẽ chọn khung truyền 8 bit.
- Start bit: là bit đầu tiên được truyền trong 1 Frame. Báo hiệu cho thiết bị nhận có một gói dữ liệu sắp được truyền đến. **Đây là bit bắt buộc.**
- Data: là dữ liệu cần truyền. Bit có trọng số nhỏ nhất LSB được truyền trước sau đó đến bit MSB.
- Parity bit: bit kiểm tra chẵn lẻ.
- Stop bit: là 1 hoặc các bit báo cho thiết bị rằng các bit đã được gửi xong. Thiết bị nhận sẽ tiến hành kiểm tra khung truyền nhằm đảm bảo tính đúng đắn của dữ liệu. **Đây là bit bắt buộc.**

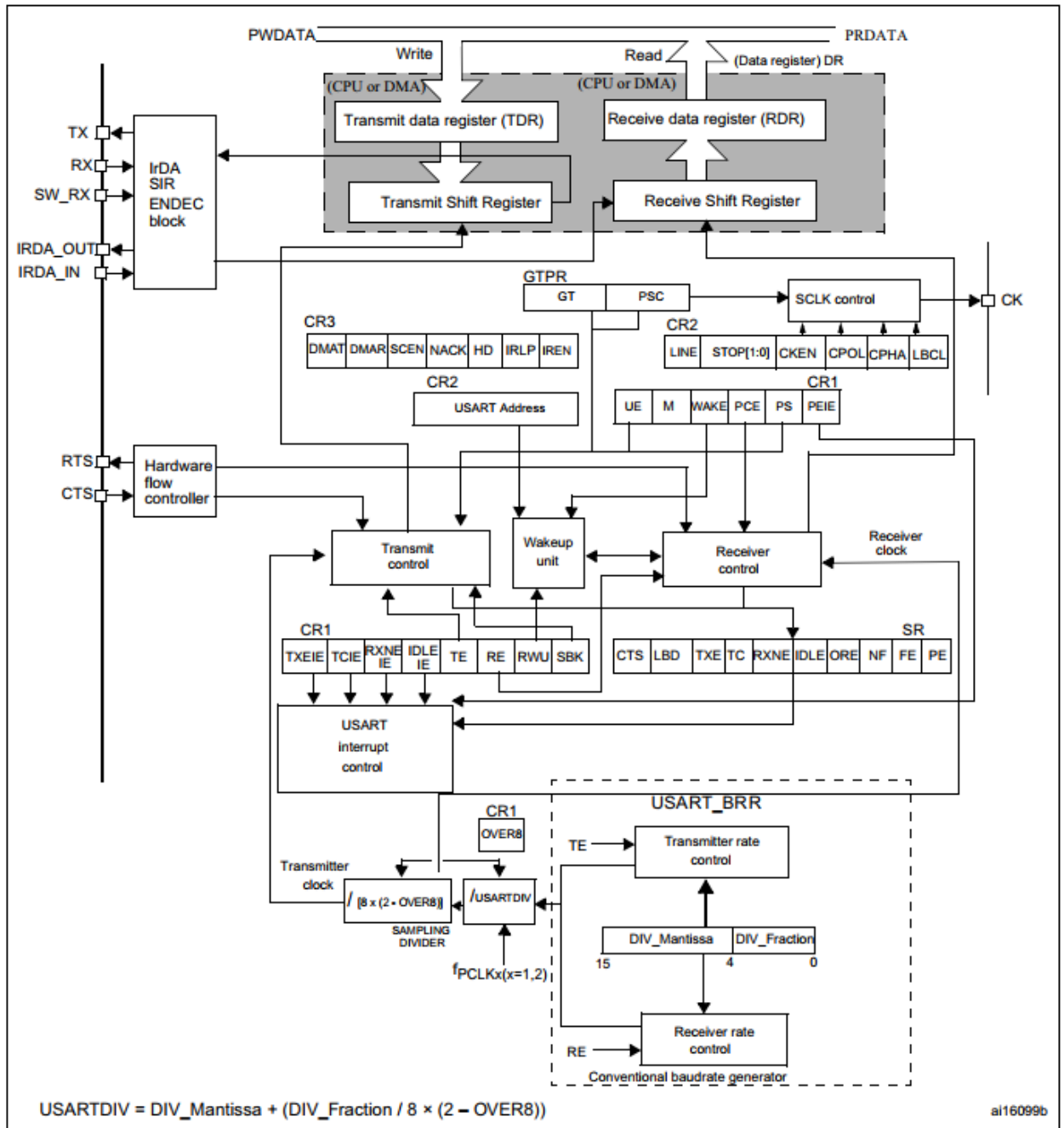
Khung truyền phổ biến nhất là: Start bit + 8 bit data + 1 stop bit.



UART/USART trên STM32

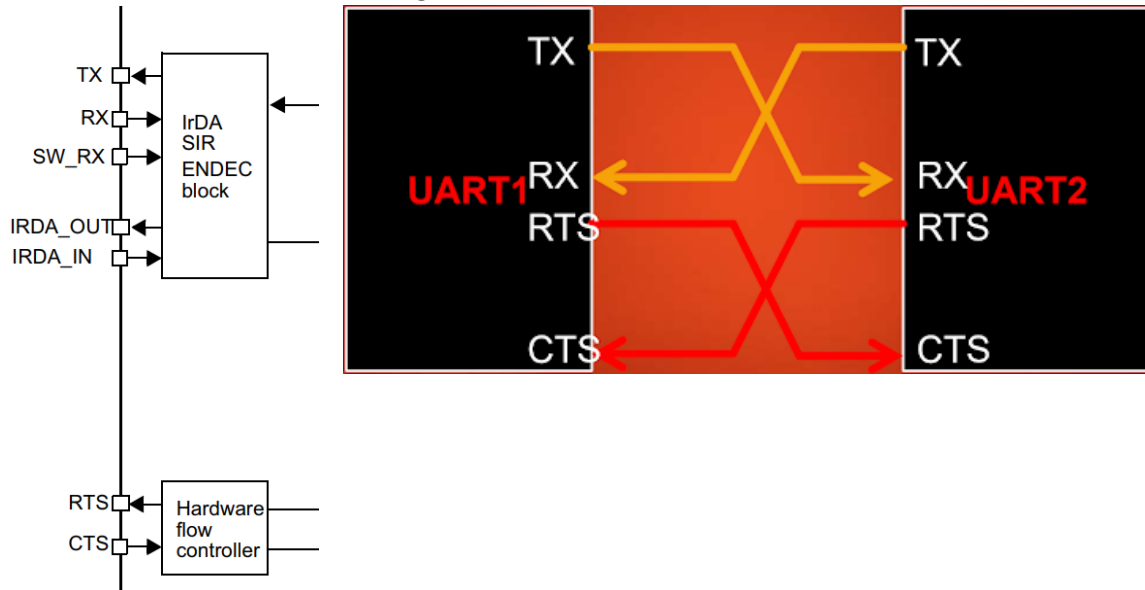
- STM32 hỗ trợ 4 USART và 4 UART trong đó USART2, USART3, UART4, UART5, UART7, UART8 nằm trên bus APB1. Còn USART1 và USART6 nằm trên bus APB2.
- Các cờ truyền/nhận, thanh ghi truyền/nhận riêng biệt.
- USART có thể hỗ trợ truyền cả chế độ đồng bộ và chế độ bất đồng bộ (Ở chế độ đồng bộ, sẽ có thêm một đường clock độc lập để đồng bộ dữ liệu, các bit start/stop sẽ không được sử dụng).
- Hỗ trợ thêm giải mã cho bộ mã hóa IrDA SIR cho giao tiếp hồng ngoại.
- Hỗ trợ giao tiếp SmartCard.
- Hỗ trợ DMA để truyền nhận dữ liệu thông qua SRAM.
- Tổng cộng có 4 cờ: Overrun Flag (Việc truyền nhận diễn ra quá nhanh), Noise Detection (phát hiện nhiễu), Frame Error (Lỗi khung truyền sai), Parity Error (lỗi chẵn lẻ của data).

Sơ đồ khối của USART trên STM32:

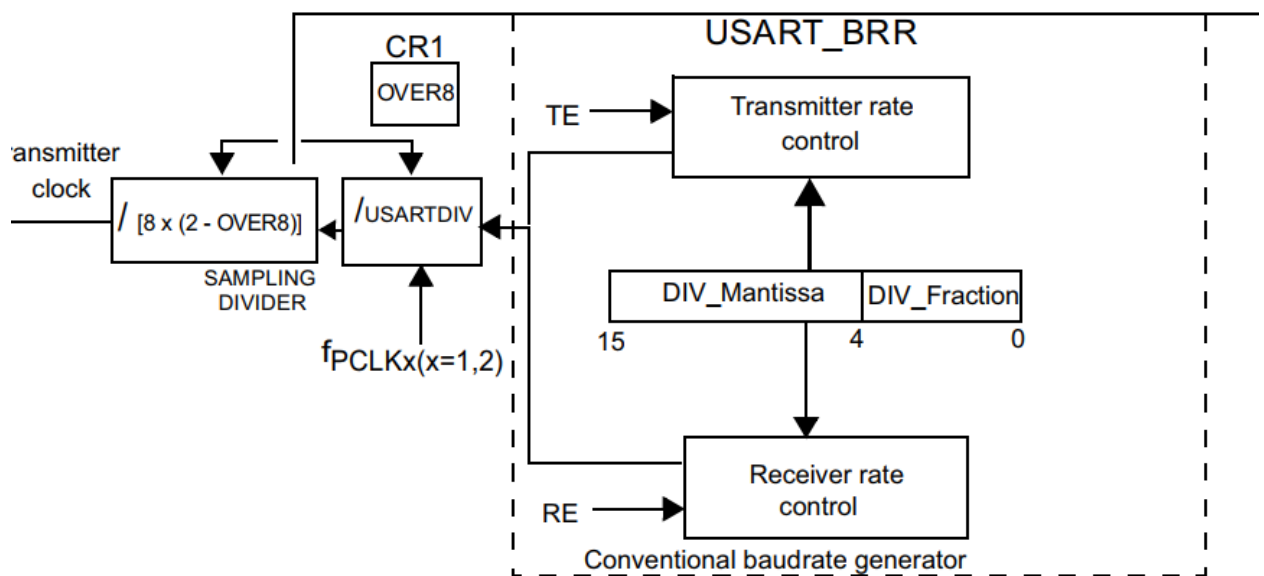


2 chân TX và RX được dùng để truyền và nhận dữ liệu. Nếu MCU không truyền dữ liệu gì thì chân TX được giữ ở mức HIGH, đó là chế độ nghỉ của TX. Để có thể nhận được dữ liệu ở chân RX thì chân RX sẽ liên tục sample để có thể bắt được start bit của frame.

Còn 2 chân RTS(Request to Send) và CTS(Clear to send) được dùng cho hardware flow control. Khi hardware flow control được dùng, dữ liệu cần được truyền ở TX chỉ được truyền đi khi chân CTS được kéo xuống mức LOW, còn chân RTS là chân được sử dụng để báo cho MCU khác là nó cần dữ liệu được từ MCU đó bằng cách ở mức LOW.

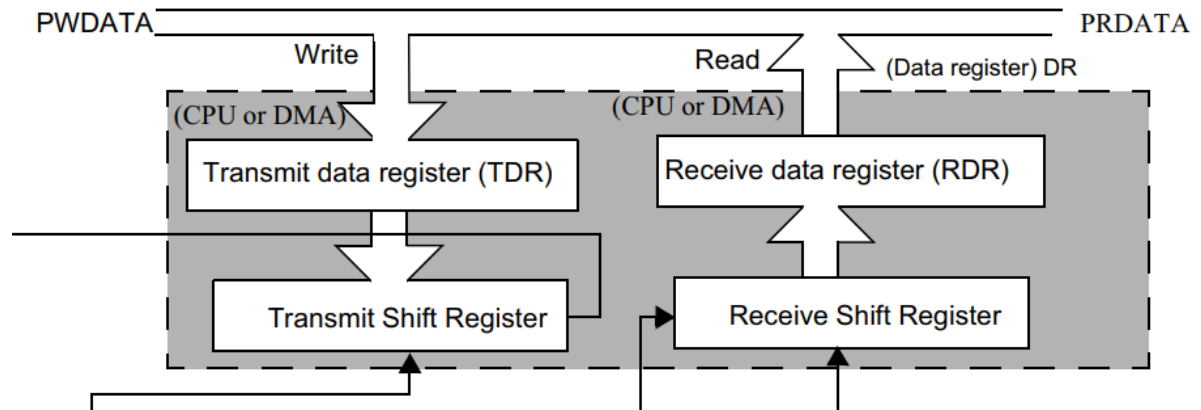


Bộ Baudrate Generator:



Đây là thứ tạo ra Baudrate cho việc truyền dữ liệu. USART_BRR phải được cấu hình phải DIV_Mantissa và DIV_Fraction cùng với bộ chia USARTDIV và tham số OVER8 để tạo ra baudrate mong muốn.

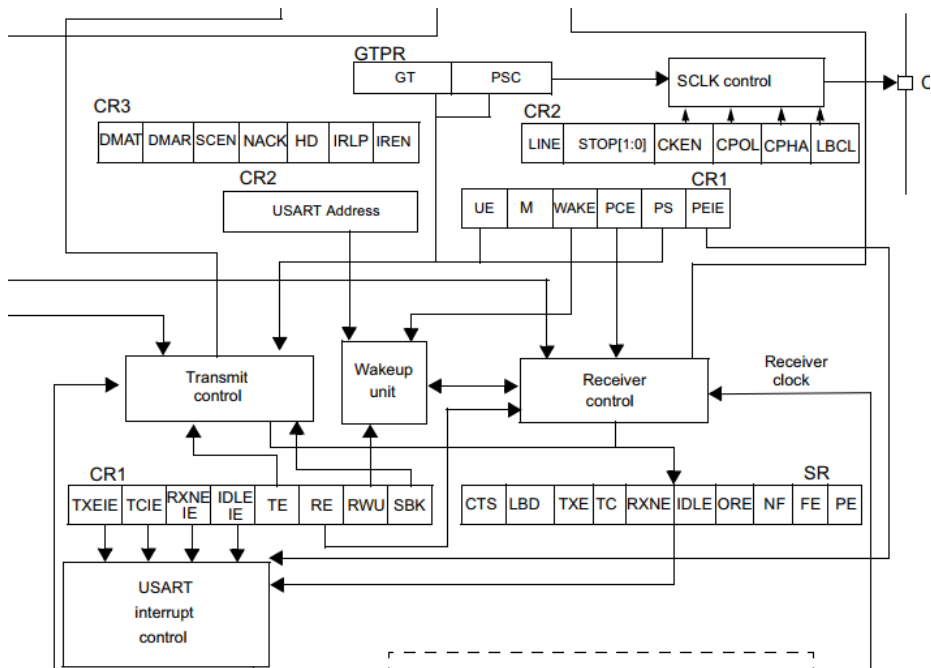
Thanh ghi TX/RX Shift và thanh ghi dữ liệu (data register):



Các thanh ghi dịch sinh ra với mục đích chuyển đổi dữ liệu.

- Khi truyền đi, nó sẽ chuyển dữ liệu song song (từ bus dữ liệu đặt lên thanh ghi data) thành tín hiệu nối tiếp trên đầu ra chân TX. Dữ liệu sẽ không được đưa tới thanh ghi dịch cho đến khi stop bit được chuyển tới thanh ghi truyền dữ liệu(TDR). Ngay sau khi stop bit được chuyển tới thanh ghi dữ liệu thì thanh ghi dịch sẽ nhận được dữ liệu mới từ thanh ghi dữ liệu.
- Ngược lại, khi nhận dữ liệu, nó sẽ chuyển tín hiệu nối tiếp từ chân RX thành tín hiệu song song để đặt vào thanh ghi data. Sau khi stop bit đến chân RX, dữ liệu nhận được ở thanh ghi dịch được chuyển đến thanh ghi nhận dữ liệu.

Khối điều khiển việc truyền nhận (Transmit/Receive control blocks):



USART registers

Thanh ghi USART_SR – Status Register

30.6.1 Status register (USART_SR)

Address offset: 0x00

Reset value: 0x0000 00C0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						CTS	LBD	TXE	TC	RXNE	IDLE	ORE	NF	FE	PE
						rc_w0	rc_w0	r	rc_w0	rc_w0	r	r	r	r	r

Thanh ghi này chứa các bit (cờ) trạng thái:

Bit[7] - TXE (Transmit data register empty): Dùng để báo hiệu data đã truyền đi hay chưa.

Khi các bit data truyền hết từ thanh ghi data sang thanh ghi dịch (thanh ghi data rỗng) thì bit này sẽ tự động được set lên 1.

Khi ta ghi dữ liệu vào thanh ghi data DR, bit này tự động set về 0.

Bit[6] - TC (Transmission Complete): Bit này tự động set lên 1 khi thanh ghi data truyền hết dữ liệu sang thanh ghi dịch, và bit TXE được set lên 1. Khác với bit TXE là bit này có thể được xóa bằng phần mềm, thường ứng dụng trong việc giao tiếp đa bộ đệm.

Bit[5] - RXNE (Read data register not empty): Dùng để báo hiệu xem data đã được nhận hay chưa?

RXNE = 1, tức là data đã được truyền xong và sẵn sàng được đọc.

RXNE = 0, tức là data chưa nhận được hoặc chưa nhận xong.

Thanh ghi USART_DR – Data register

Address offset: 0x04

Reset value: 0xFFFF XXXX

Bits 31:9 Reserved, must be kept at reset value

Bits 8:0 DR[8:0]: Data value

Thanh ghi này chứa dữ liệu truyền và nhận, nó làm cả 2 nhiệm vụ này, và tùy thuộc vào trạng thái của ngoại vi là đang truyền hay đang nhận, nó sẽ giao tiếp với các bộ đệm dữ liệu truyền/nhận.

Thanh ghi này chỉ sử dụng 9 bit LSB để chứa dữ liệu (8 bit hoặc 9 bit).

Thanh ghi USART_BRR - USART Baudrate

Note: The baud counters stop counting if the TE or RE bits are disabled respectively.

Address offset: 0x08

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIV_Mantissa[11:0]												DIV_Fraction[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Thanh ghi này dùng để tính toán tốc độ Baudrate cho UART với:

Các bit[11:0] - DIV_Mantissa: là thành phần đứng trước dấu phẩy.

Các bit[3:0] - DIV_Fraction: là thành phần đứng sau dấu phẩy.

Thanh ghi USART_CR1 - Control Register 1

Address offset: 0x0C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OVER8	Reserved	UE	M	WAKE	PCE	PS	PEIE	TXEIE	TCIE	RXNEIE	IDLEIE	TE	RE	RWU	SBK
rw	Res.	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Thanh ghi này và 2 thanh ghi sau đều chứa các bit dùng để cấu hình cho USART:

Bit[15] - OVER8 (Oversampling Over): Dùng để cấu hình trong việc tính Baudrate.

OVER8 = 0, oversampling by 8.

OVER8 = 1, oversampling by 16.

Bit[14] - UE (USART Enable): Set lên 1 để cho phép ngoại vi USART hoạt động.

Bit[13] - M (Word length): Xác định độ dài data truyền đi.

M = 0, 1 Start bit, 8 Data bits, n Stop bit.

M = 1, 1 Start bit, 9 Data bits, n Stop bit.

Bit[9] - PS (Parity Selection): Chọn bit chẵn/lẻ khi bit chẵn/lẻ được cho phép (bằng bit PCE).

PS = 0, even parity (chẵn).

PS = 1, odd parity (lẻ).

Bit[7:5] - 3 bit này là các cờ báo ngắt, tương ứng với 3 bit TXE, TC, RXNE ở trên.

Bit[3] - TE (Transmitter Enable): Set bằng 1 để cho phép truyền dữ liệu.

Bit[2] - RE (Receiver Enable): Set bằng 1 để cho phép nhận dữ liệu.

Thanh ghi USART_CR2 - Control Register 2

Address offset: 0x10

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	LINEN	STOP[1:0]		CLKEN	CPOL	CPHA	LBCL	Res.	LBDIE	LBDL	Res.	ADD[3:0]			
	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw

Bit[13:12] - STOP: Cấu hình số bit Stop.

Thanh ghi USART_CR3 - Control Register 3

Address offset: 0x14

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				ONEBIT	CTSIE	CTSE	RTSE	DMAT	DMAR	SCEN	NACK	HDSEL	IRLP	IREN	EIE
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit[7] - DMAT (DMA Enable Transmitter): Set lên 1 cho phép truyền dữ liệu bằng DMA.

Bit[6] - DMAR (DMA Enable Receiver): Set lên 1 cho phép nhận dữ liệu bằng DMA.

Thanh ghi USART_GTPR - Guard Time and Prescaler Register

Address offset: 0x18

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GT[7:0]								PSC[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Thanh ghi này chủ yếu sử dụng các bit từ 0 đến 4 để tính toán tốc độ Baudrate.

Các bước sử dụng USART để nhận dữ liệu:

1. Cho phép USART hoạt động bằng cách set bit UE ở thanh ghi USART_CR1.
2. Cấu hình bit M ở thanh ghi USART_CR1 để khai báo số bit dữ liệu.
3. Cấu hình số lượng stop bit bằng thanh ghi USART_CR2 (bit 13, 12).
4. Chọn baudrate mong muốn bằng thanh ghi USART_BRR.
5. Set bit RE ở thanh ghi USART_CR1 để cho phép nhận dữ liệu.
6. Khi dữ liệu từ thanh ghi dịch đã được chuyển đến thanh ghi nhận dữ liệu(RDR) thì cờ RXNE được set và dữ liệu sẽ được đọc từ thanh ghi dữ liệu USART_DR. (Cờ RXNE phải được clear trước khi dữ liệu tiếp theo truyền đến thanh ghi nhận dữ liệu kết thúc)

Các bước sử dụng USART để truyền dữ liệu:

1. Cho phép USART hoạt động bằng cách set bit UE ở thanh ghi USART_CR1.
2. Cấu hình bit M ở thanh ghi USART_CR1 để khai báo số bit dữ liệu.
3. Cấu hình số lượng stop bit bằng thanh ghi USART_CR2 (bit 13, 12).
4. Chọn baudrate mong muốn bằng thanh ghi USART_BRR.
5. Set bit TE ở thanh ghi USART_CR1 để cho phép truyền dữ liệu.
6. Lúc này cờ TXE được set. Sau đó viết dữ liệu để gửi vào thanh ghi dữ liệu USART_DR, lúc này cờ TXE sẽ được clear.
7. Sau khi viết xong, đợi cho đến khi bit TC = 1. Nó cho biết quá trình truyền đã xong.

Cấu hình Baudrate.

Tốc độ baudrate được cấu hình bởi thanh ghi USART_BRR.

Có 2 cách để chọn baudrate:

1. Sử dụng các bảng tính toán baudrate có sẵn ở trong Reference Manual.

Table 135. Error calculation for programmed baud rates at $f_{PCLK} = 8 \text{ MHz}$ or $f_{PCLK} = 12 \text{ MHz}$, oversampling by 8⁽¹⁾

Oversampling by 8 (OVER8 = 1)							
Baud rate		$f_{PCLK} = 8 \text{ MHz}$			$f_{PCLK} = 12 \text{ MHz}$		
S.No	Desired	Actual	Value programmed in the baud rate register	% Error = (Calculated - Desired) B.rate / Desired B.rate	Actual	Value programmed in the baud rate register	% Error
1	1.2 Kbps	1.2 Kbps	833.375	0	1.2 Kbps	1250	0
2	2.4 Kbps	2.4 Kbps	416.625	0.01	2.4 Kbps	625	0
3	9.6 Kbps	9.604 Kbps	104.125	0.04	9.6 Kbps	156.25	0
4	19.2 Kbps	19.185 Kbps	52.125	0.08	19.2 Kbps	78.125	0
5	38.4 Kbps	38.462 Kbps	26	0.16	38.339 Kbps	39.125	0.16
6	57.6 Kbps	57.554 Kbps	17.375	0.08	57.692 Kbps	26	0.16
7	115.2 Kbps	115.942 Kbps	8.625	0.64	115.385 Kbps	13	0.16
8	230.4 Kbps	228.571 Kbps	4.375	0.79	230.769 Kbps	6.5	0.16
9	460.8 Kbps	470.588 Kbps	2.125	2.12	461.538 Kbps	3.25	0.16
10	921.6 Kbps	888.889 Kbps	1.125	3.55	923.077 Kbps	1.625	0.16
11	2 MBps	NA	NA	NA	NA	NA	NA
12	3 MBps	NA	NA	NA	NA	NA	NA

1. The lower the CPU clock the lower the accuracy for a particular baud rate. The upper limit of the achievable baud rate can be fixed with these data.

2. Sử dụng các công thức được viết trong Reference Manual.

Equation 1: Baud rate for standard USART (SPI mode included)

$$\text{Tx/Rx baud} = \frac{f_{\text{CK}}}{8 \times (2 - \text{OVER8}) \times \text{USARTDIV}}$$

Equation 2: Baud rate in Smartcard, LIN and IrDA modes

$$\text{Tx/Rx baud} = \frac{f_{\text{CK}}}{16 \times \text{USARTDIV}}$$

Ví dụ minh họa: Cấu hình baudrate là 19200, tần số PCLK = 16MHz.

Chọn OVER8 = 0 ta có USARTDIV xấp xỉ 52,0833, để tính toán DIV_Fraction ra số nguyên thì chọn USARTDIV = 52,0625. Nên DIV_Mantissa = 52 = 0x34, DIV_Fraction = 0,0625 * 8 * (2 - OVER8) = 1 = 0x01. Vậy USART_BRR = 0x00000341.