

# LẬP TRÌNH ARM - STM32

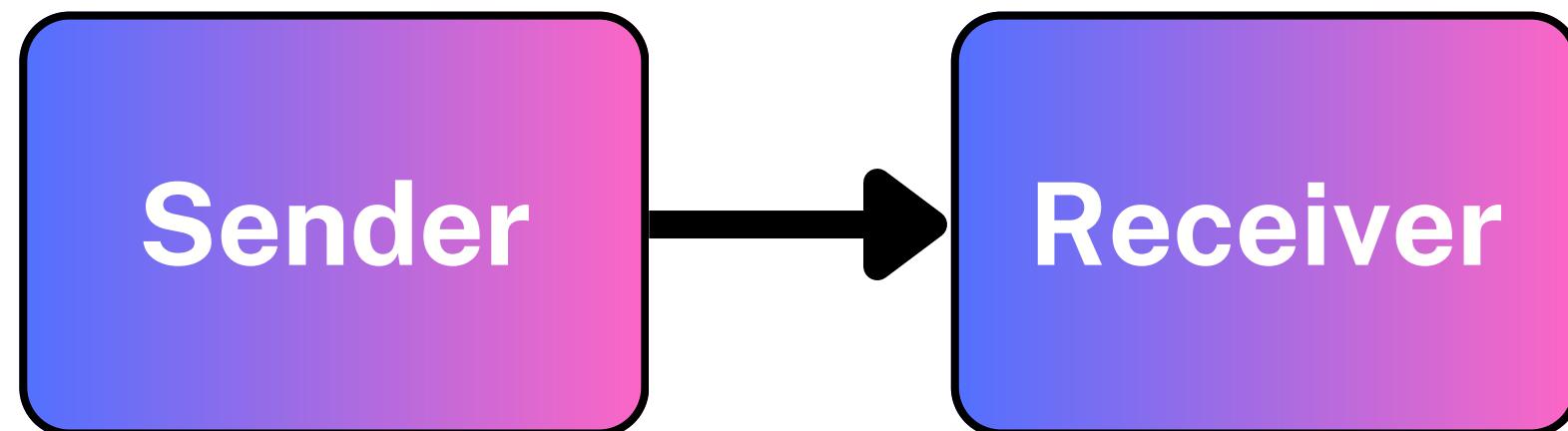
STM32 - UART

Date: Mar 09, 2025

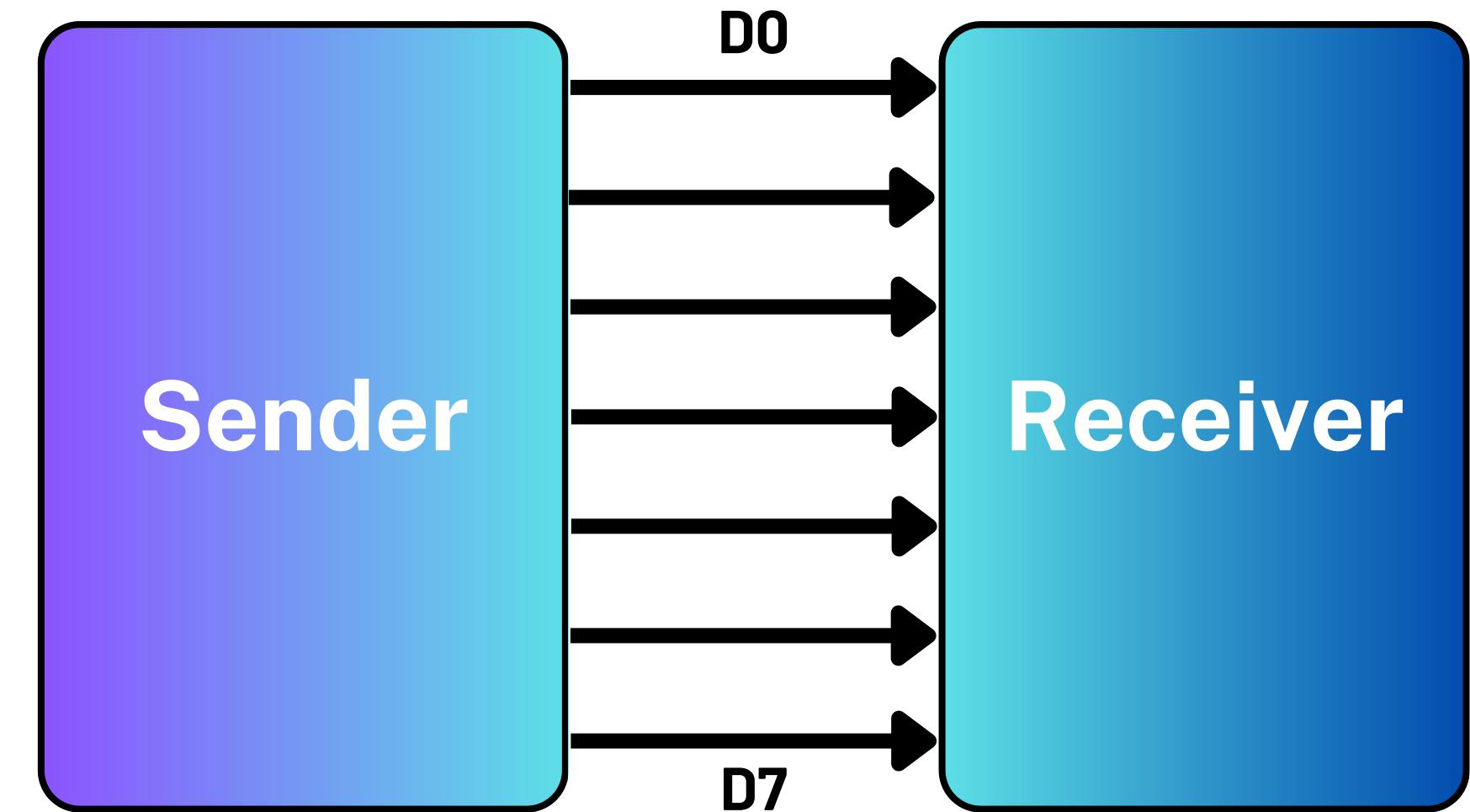


# Serial & Parallel

Serial Transfer



Parallel Transfer

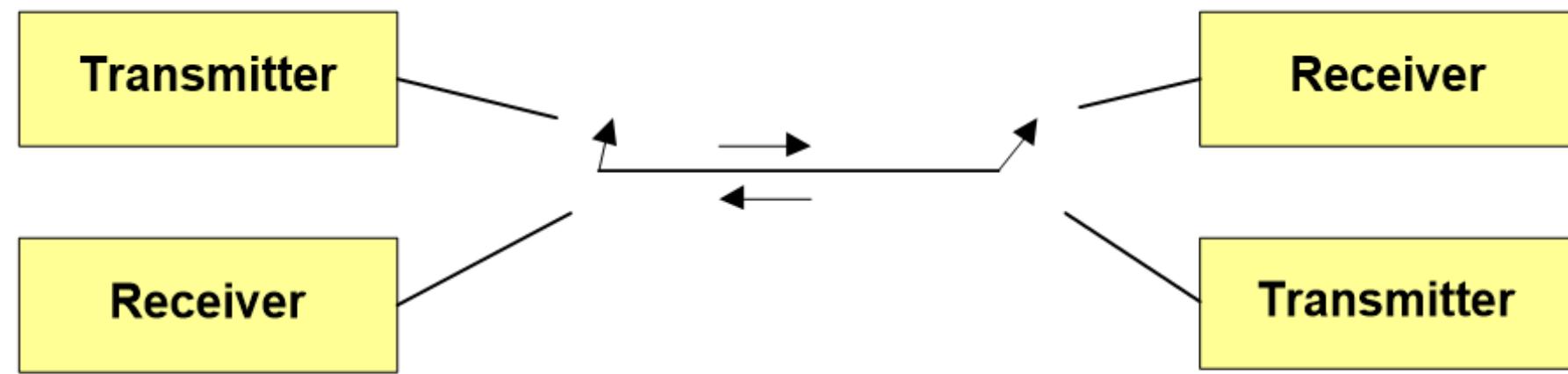


# Direction

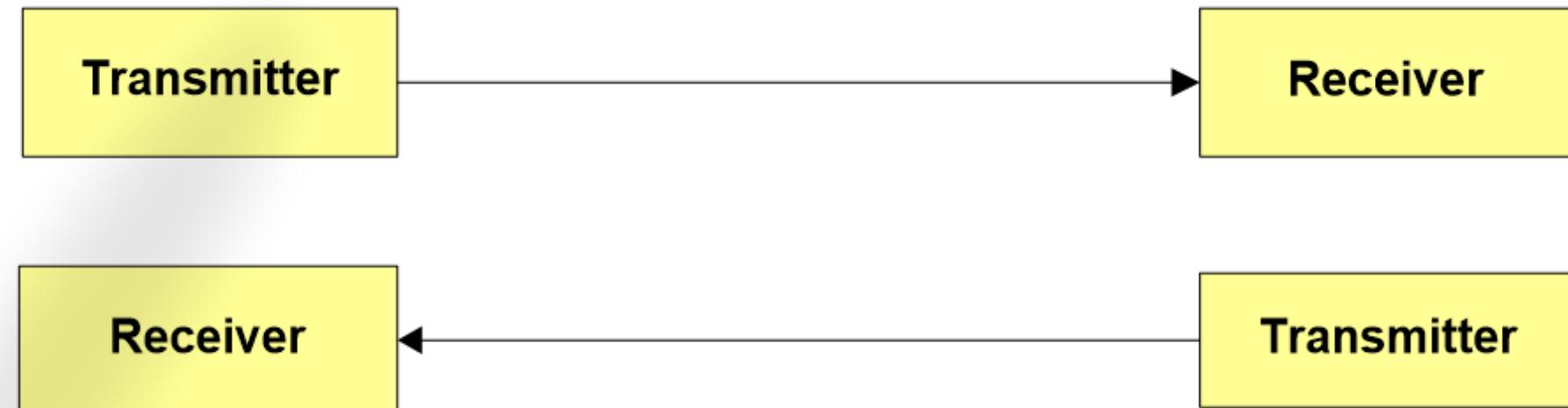
**Simplex**



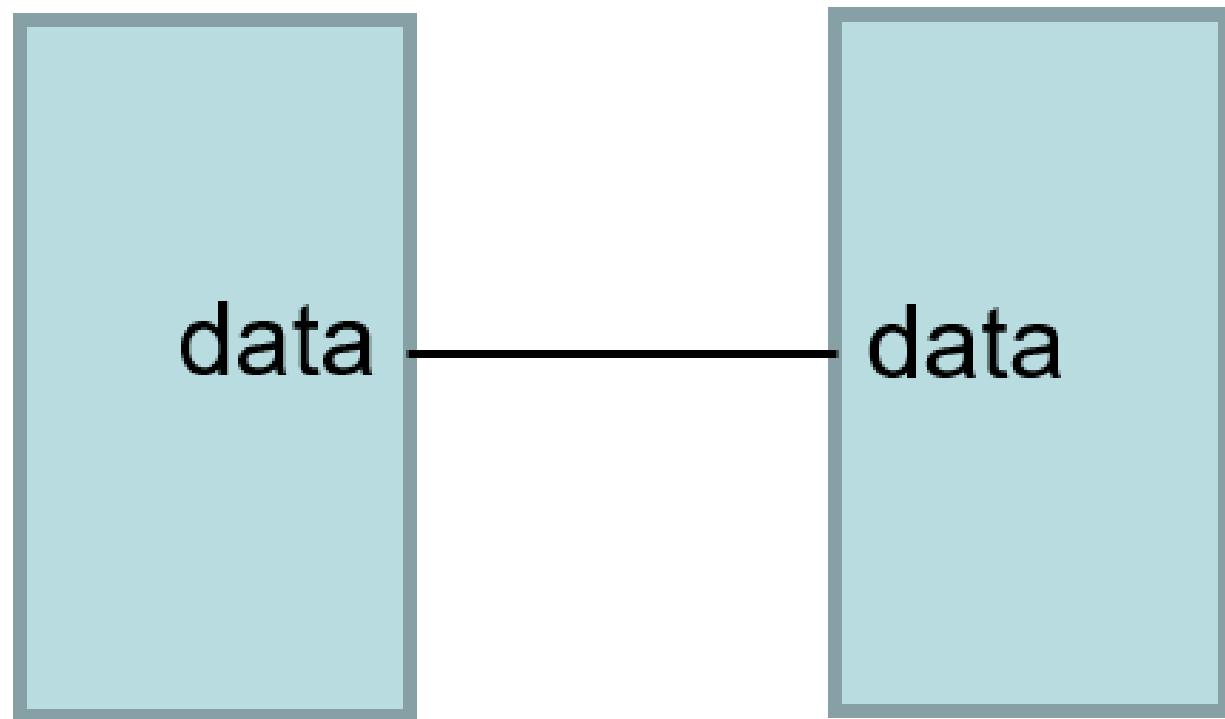
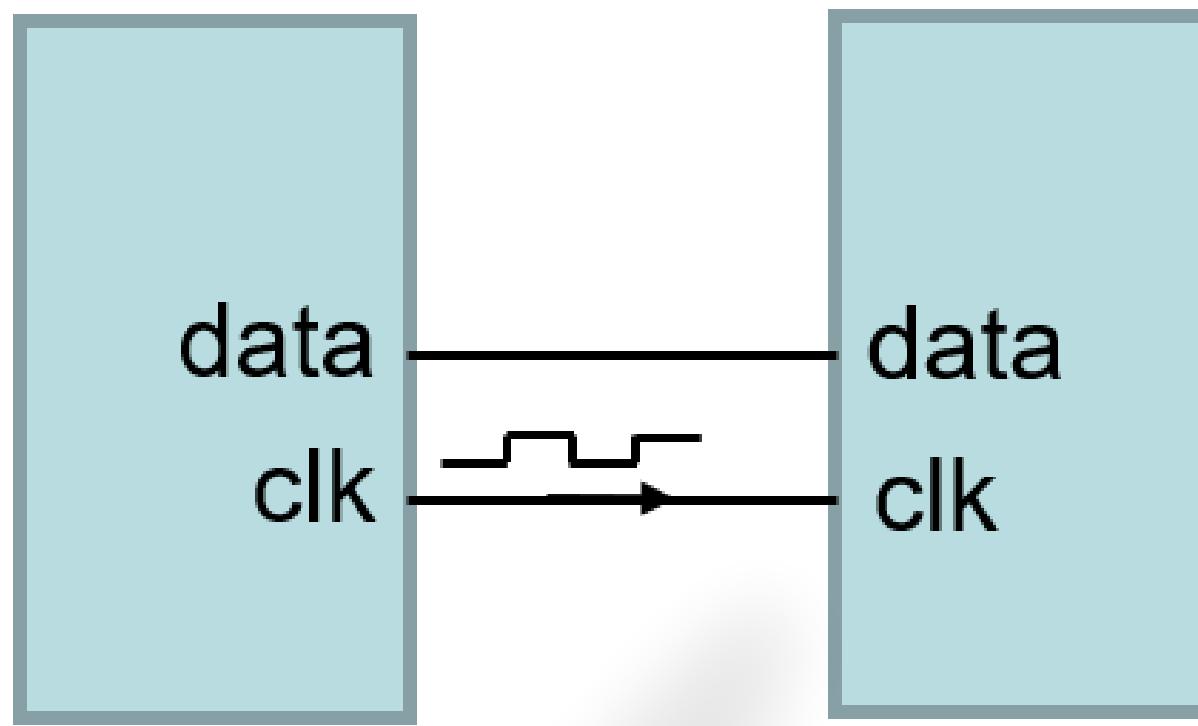
**Half Duplex**



**Full Duplex**

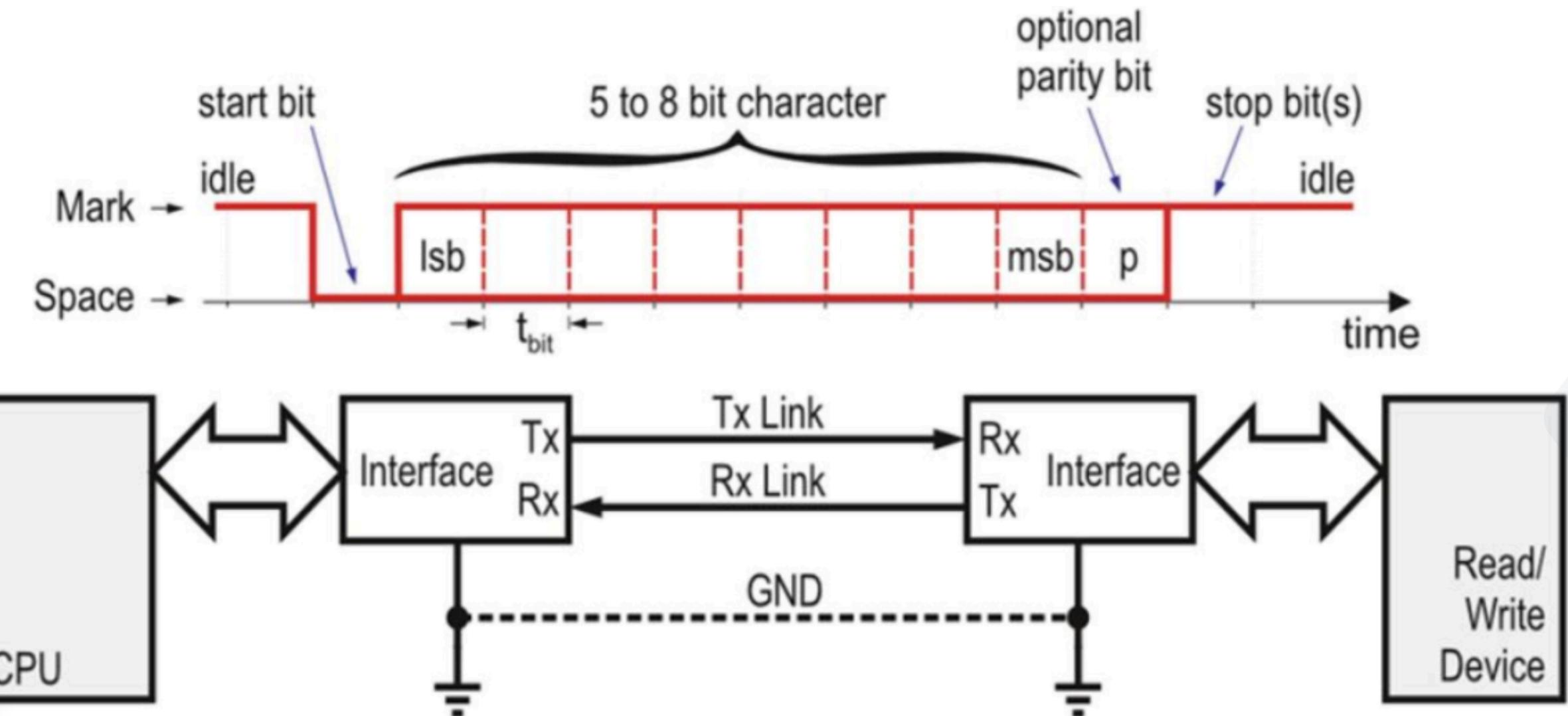


# Synchronous & Asynchronous



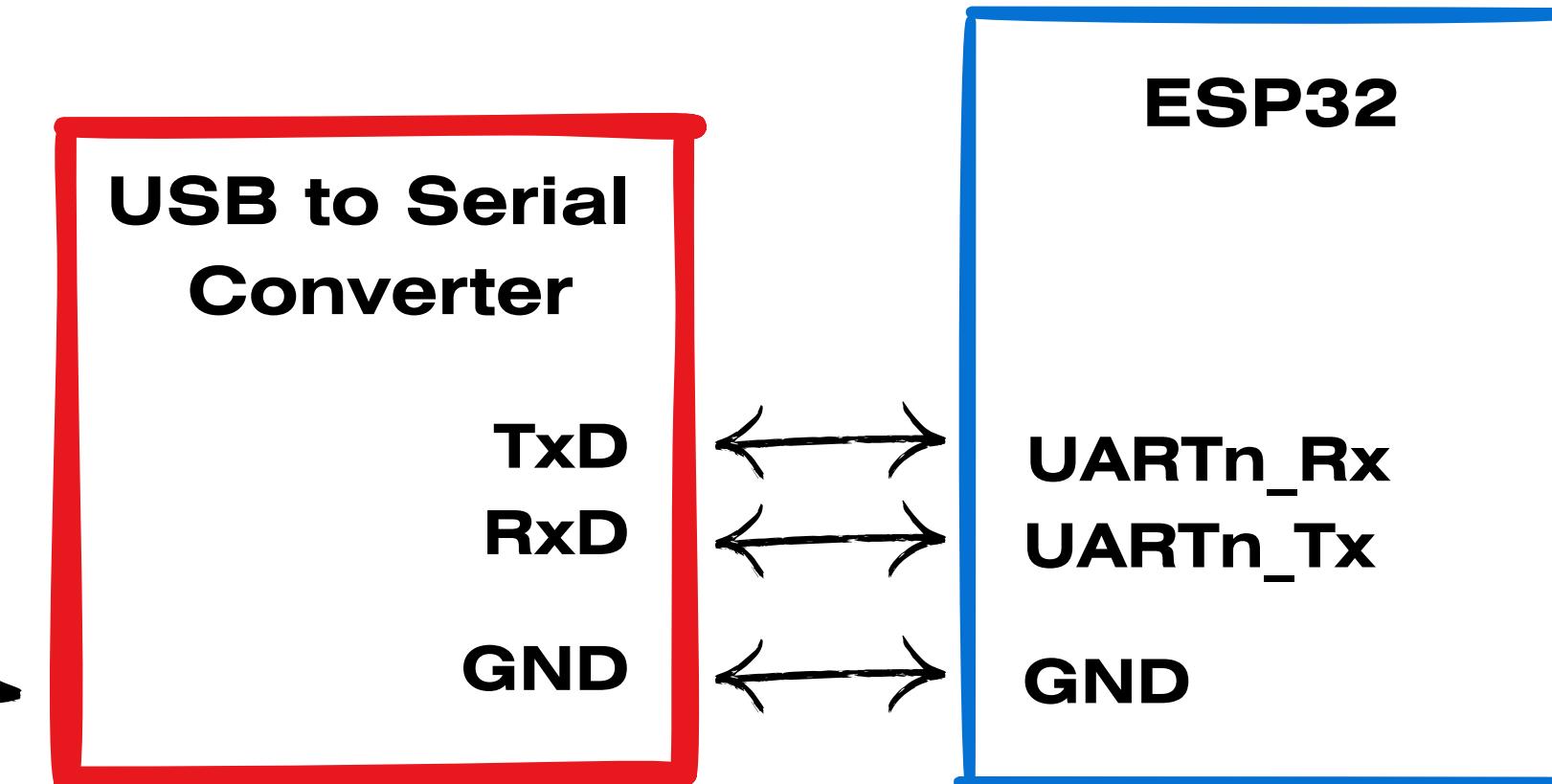
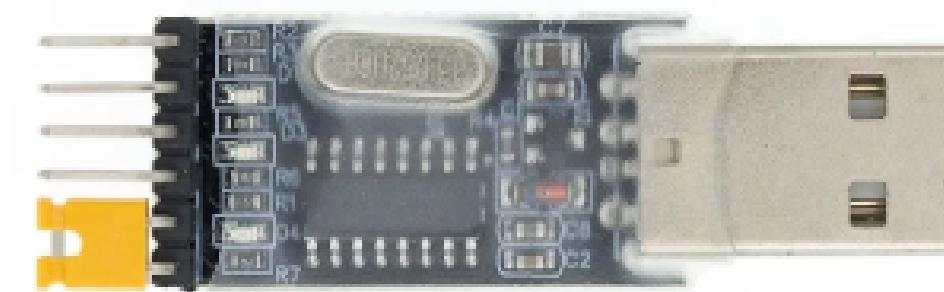
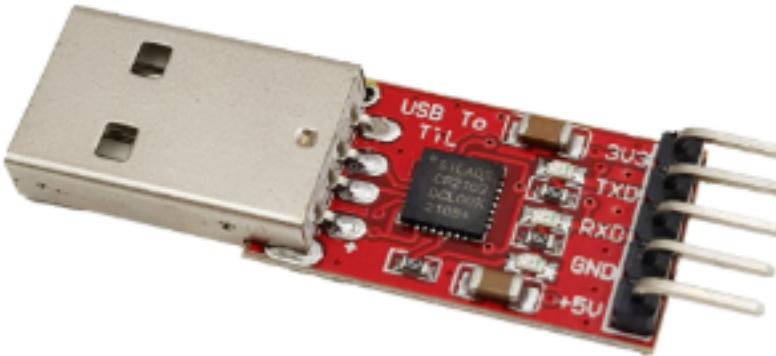
# Data Frame

- ▶ Start bit
- ▶ Data
- ▶ Parity bit
- ▶ Stop bit
- ▶ Baudrate

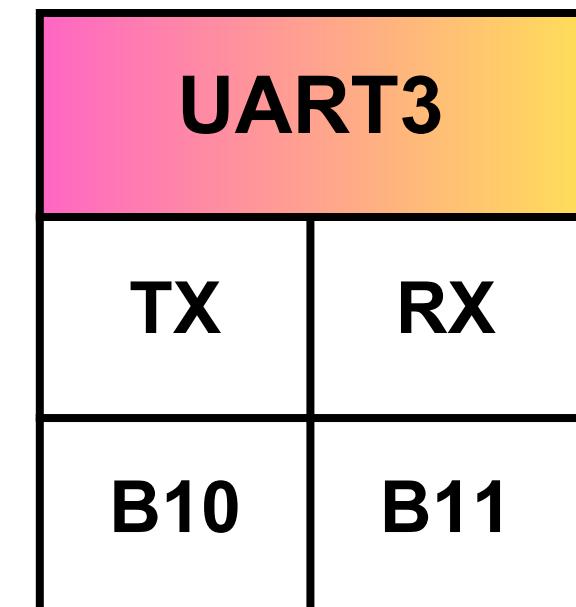
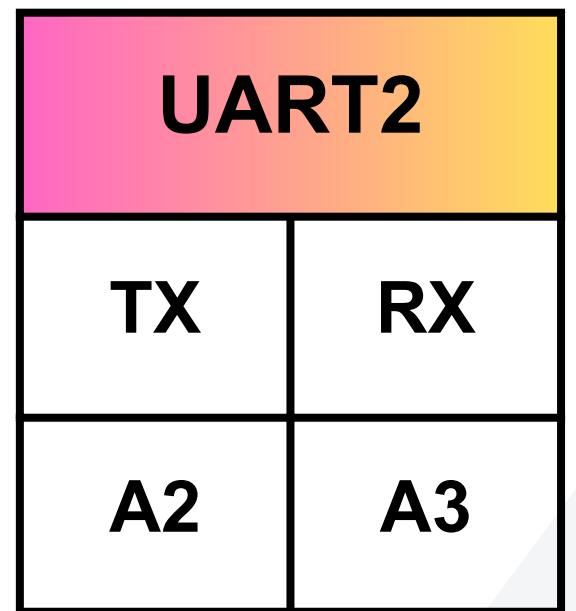
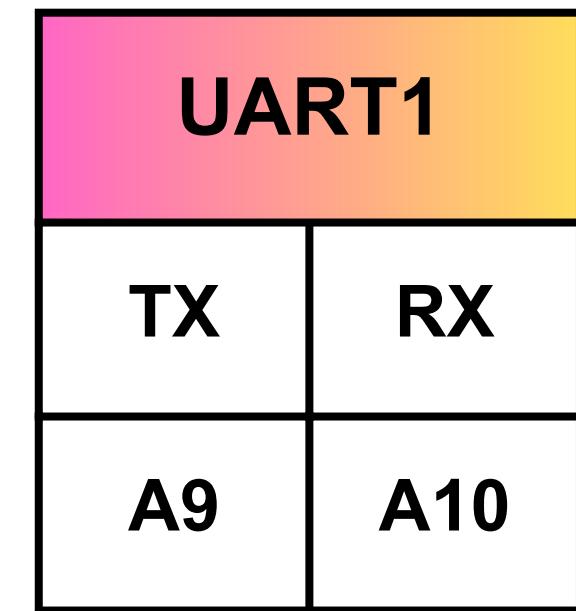
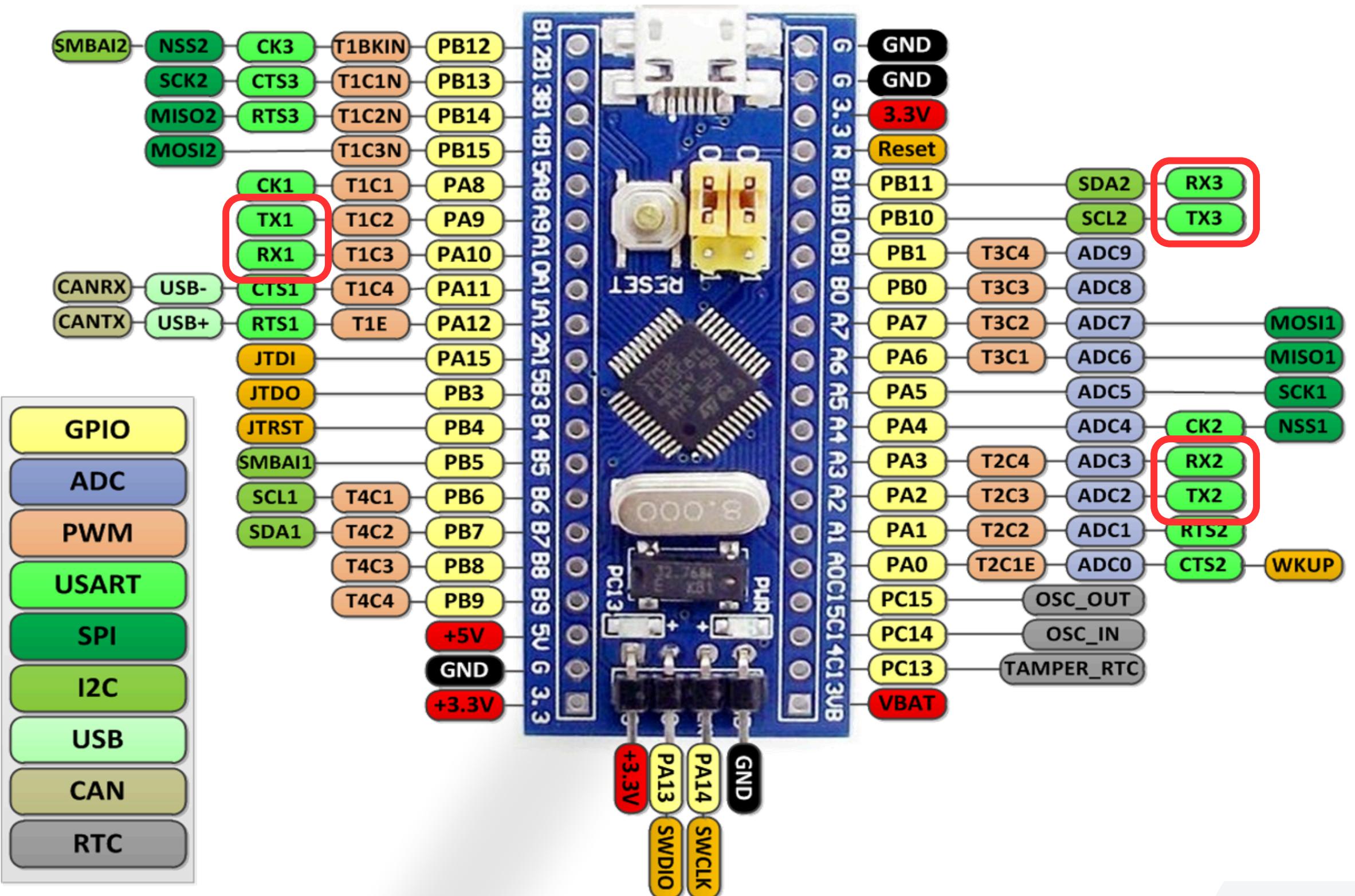


Sơ đồ kết nối và Khung dữ liệu

# Data Frame



# UART - PINOUT



# STM32F1 - UART Library

**USART\_InitTypeDef A;**

(Khai báo biến A thuộc kiểu dữ liệu  
USART\_InitTypeDef)

**A. USART\_BaudRate = X;**

(Cấu hình tốc độ truyền dữ liệu)  
X = 9600, 115200...

**A. USART\_Mode = B;**

(Cấu hình chế độ truyền hoặc nhận dữ liệu)

B:

USART\_Mode\_Rx  
USART\_Mode\_Tx

B:

Chế độ nhận dữ liệu  
Chế độ truyền dữ liệu

**A. USART\_Parity = B;**

(Cấu hình bit kiểm tra)

B:

USART\_Parity\_No  
USART\_Parity\_Even  
USART\_Parity\_Odd

B:

Không sử dụng bit kiểm tra  
Sử dụng bit kiểm tra chẵn  
Sử dụng bit kiểm tra lẻ

# STM32F1 - UART Library

**USART\_Cmd(B, C);**

(Cấu hình cho phép hoặc không cho phép USARTx hoạt động)

B: USARTx

USART1

USART2

...

C:

ENABLE

DISABLE

B:

Sử dụng bộ USART1

Sử dụng bộ USART2

...

C:

Cho phép

Không cho phép

**A. USART\_WordLength = B:**

(Cấu hình số bit dữ liệu)

B:

USART\_WordLength\_8b

USART\_WordLength\_9b

B:

Số bit dữ liệu là 8

Số bit dữ liệu là 9

**USART\_Init(B, &A);**

(Lệnh cấu hình cho USARTx theo các thông số được lưu trong biến A)

B: USARTx

USART1

USART2

...

B: USARTx

USART1

USART2

...

# STM32F1 - UART Library

<b>USART_ITConfig (A, B, C) ;</b> (Hàm cho phép hoặc không cho phép ngắt USART)	
<b>A: USARTx</b> USART1 USART2 USART3	<b>A: Bộ USARTx</b> Sử dụng bộ USART1 Sử dụng bộ USART2 Sử dụng bộ USART3
<b>B:</b> USART_IT_TXE USART_IT_RXN  ...	<b>B:</b> Ngắt truyền Ngắt nhận  ...
<b>C:</b> ENABLE DISABLE	<b>C:</b> Cho phép Không cho phép

<b>USART_ReceiveData (A) ;</b> (Hàm nhận dữ liệu)	
<b>A: USARTx</b> USART1 USART2 ...	<b>A:</b> Sử dụng bộ USART1 Sử dụng bộ USART2 ...
<b>USART_SendData (A, B) ;</b> (Hàm truyền dữ liệu)	
<b>A: USARTx</b> USART1 USART2 ...	<b>A:</b> Sử dụng bộ USART1 Sử dụng bộ USART2 ...
<b>B: Data</b>	<b>B:</b> Dữ liệu cần truyền đi.

# STM32F1 - UART Library

**NVIC\_EnableIRQ(A) ;**  
(Cho phép ngắt tổng quát)

**A:**  
USART1\_IRQHandler  
USART2\_IRQHandler  
USART3\_IRQHandler

**A:**  
Ngắt tổng quát USART1  
Ngắt tổng quát USART2  
Ngắt tổng quát USART3

**USART\_ClearITPendingBit(A, B) ;**  
(Xóa các bit báo ngắt đang chờ xử lý)

**A: USARTx**

USART1

...

**B:**

USART\_IT\_TC

USART\_IT\_RXNE

...

**A: USARTx**

USART1

...

**B:**

Ngắt khi quá trình  
truyền hoàn tất

Ngắt khi thanh ghi dữ  
liệu nhận không trống

**USART\_GetITStatus(A, B) ;**

(Kiểm tra xem ngắt USART được chỉ định có xảy ra hay không)

**A:**  
USART\_IT\_TXE  
USART\_IT\_TC  
USART\_IT\_RXNE

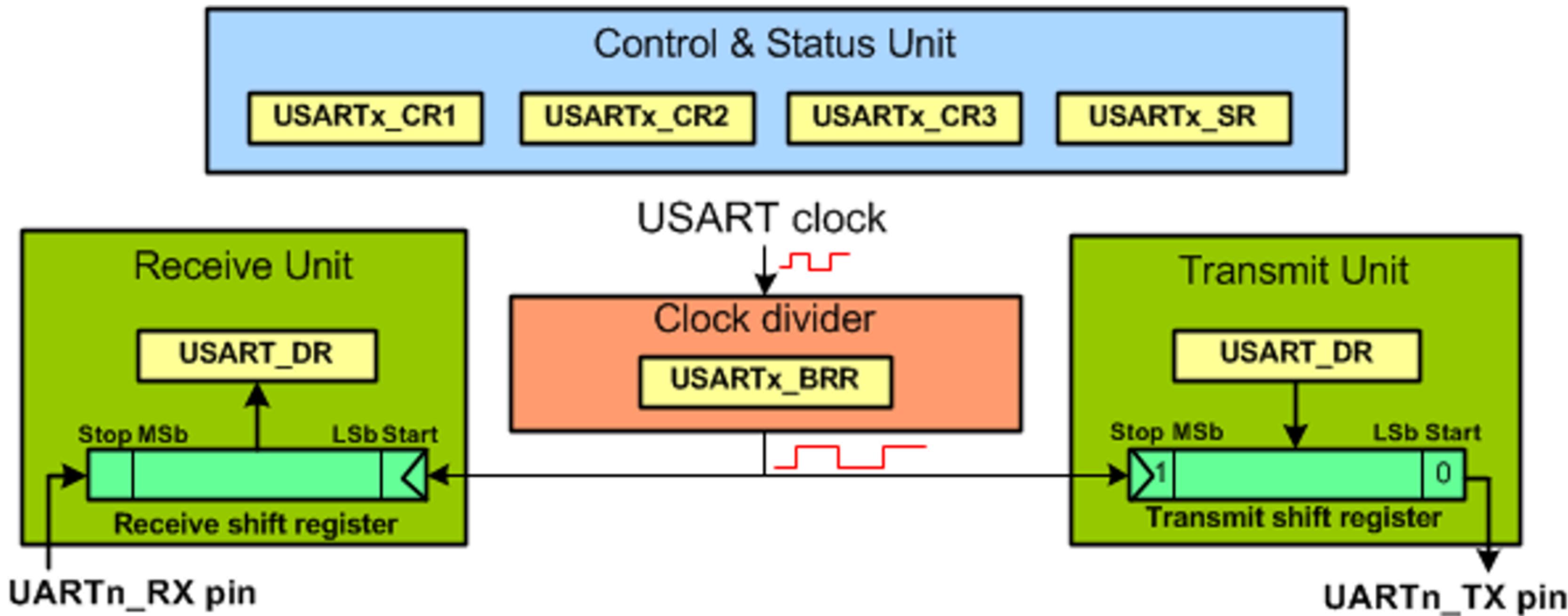
...

**B:**  
SET  
RESET

**A:**  
Ngắt khi thanh ghi dữ liệu truyền trống  
Ngắt khi quá trình truyền hoàn tất  
Ngắt khi thanh ghi dữ liệu nhận không trống

...

# STM32F1 - UART Registers



# STM32F1 - UART Registers

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	DAC EN	PWR EN	BKP EN	CAN2 EN	CAN1 EN	Reserved	I2C2 EN	I2C1 EN	UART5E N	UART4E N	USART3 EN	USART2 EN	Res.		
	RW	RW	RW	RW	RW		RW	RW	RW	RW	RW	RW			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPI3 EN	SPI2 EN	Reserved	WWD GEN	Reserved				TIM7 EN	TIM6 EN	TIM5 EN	TIM4 EN	TIM3 EN	TIM2 EN		
RW	RW		RW					RW	RW	RW	RW	RW	RW		

RCC\_APB1ENR

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
Res.	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	USART 1EN	Res.	SPI1 EN	TIM1 EN	ADC2 EN	ADC1 EN	Reserved	IOPE EN	IOPD EN	IOPC EN	IOPB EN	IOPA EN	Res.	AFIO EN	
	RW		RW	RW	RW	RW		RW	RW	RW	RW	RW		RW	

RCC\_APB2ENR

# STM32F1 - UART Registers

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

**USARTx\_BRR**

$$A = 72\text{MHz/Baud}$$

**USARTx** → BRR    | = A;

# STM32F1 - UART Registers

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		UE	M	WAKE	PCE	PS	PEIE	TXEIE	TCIE	RXNEIE	IDLEIE	TE	RE	RWU	SBK
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

USARTx\_CR1

## ► Bit 2 RE: Receiver enable

- 0: Receiver is disabled
- 1: Receiver is enabled and begins searching for a start bit

## ► Bit 4 IDLEIE: IDLE interrupt enable

- 0: Interrupt is inhibited
- 1: A USART interrupt is generated whenever IDLE=1 in the USART\_SR register

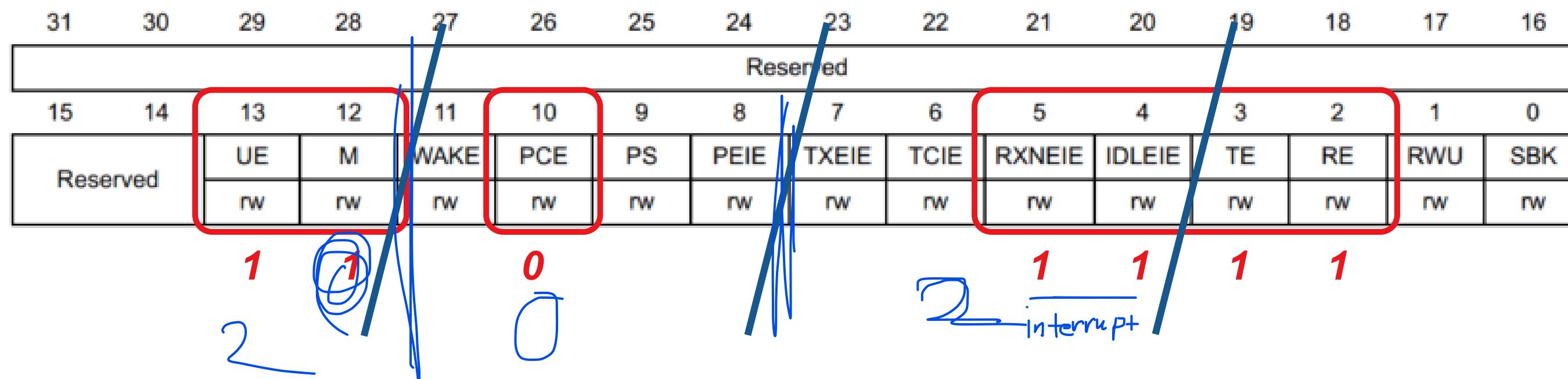
## ► Bit 3 TE: Transmitter enable

- 0: Transmitter is disabled
- 1: Transmitter is enabled

## ► Bit 5 RXNEIE: RXNE interrupt enable

- 0: Interrupt is inhibited
- 1: A USART interrupt is generated whenever ORE=1 or RXNE=1 in the USART\_SR register

# STM32F1 - UART Registers



# USARTx\_CR1

## Bit 12 M: Word length

- 0: 1 Start bit, 8 Data bits, n Stop bit
  - 1: 1 Start bit, 9 Data bits, n Stop bit

## Bit 10 PCE: Parity control enable

- 0: Parity control disabled**  
**1: Parity control enabled**

**USARTx -> CR1 |= 0x~~3~~03C;**

# STM32F1 - UART Registers

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]		CLK EN	CPOL	CPHA	LBCL	Res.	LBDIE	LBDL	Res.	ADD[3:0]			
	RW	RW	RW	RW	RW	RW	RW		RW	RW	RW	RW	RW	RW	RW
	0	0													

USARTx\_CR2

► Bits 13:12 STOP: STOP bits

00: 1 Stop bit

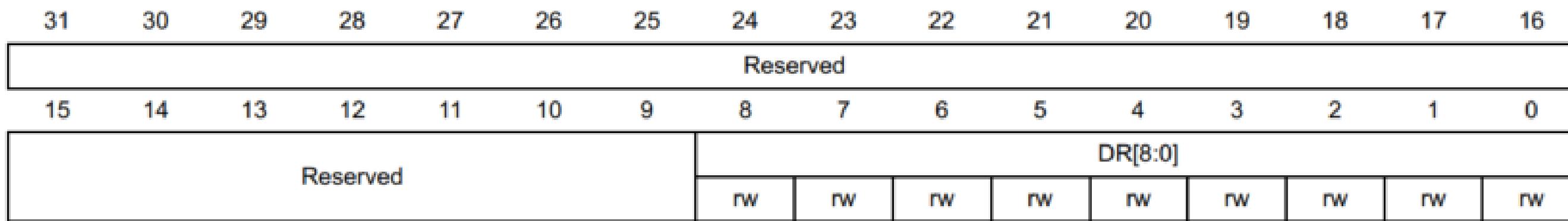
01: 0.5 Stop bit

10: 2 Stop bits

11: 1.5 Stop bit

USARTx → CR2 |= 0x00;

# STM32F1 - UART Registers



**USARTx\_DR**

# STM32F1 - UART Registers

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	NE	FE	PE	
					rc_w0	rc_w0	r	rc_w0	rc_w0	r	r	r	r	r	r

**USARTx\_SR**