

STM32 course

Just USART

Going back to DMA for a little while

- Main features
- Set of registers
- Programmable transfer size
- Circular, one-shot modes
- Mem-to-mem mode
- Up to 64K total transfer size

Type of communication service



Simplex

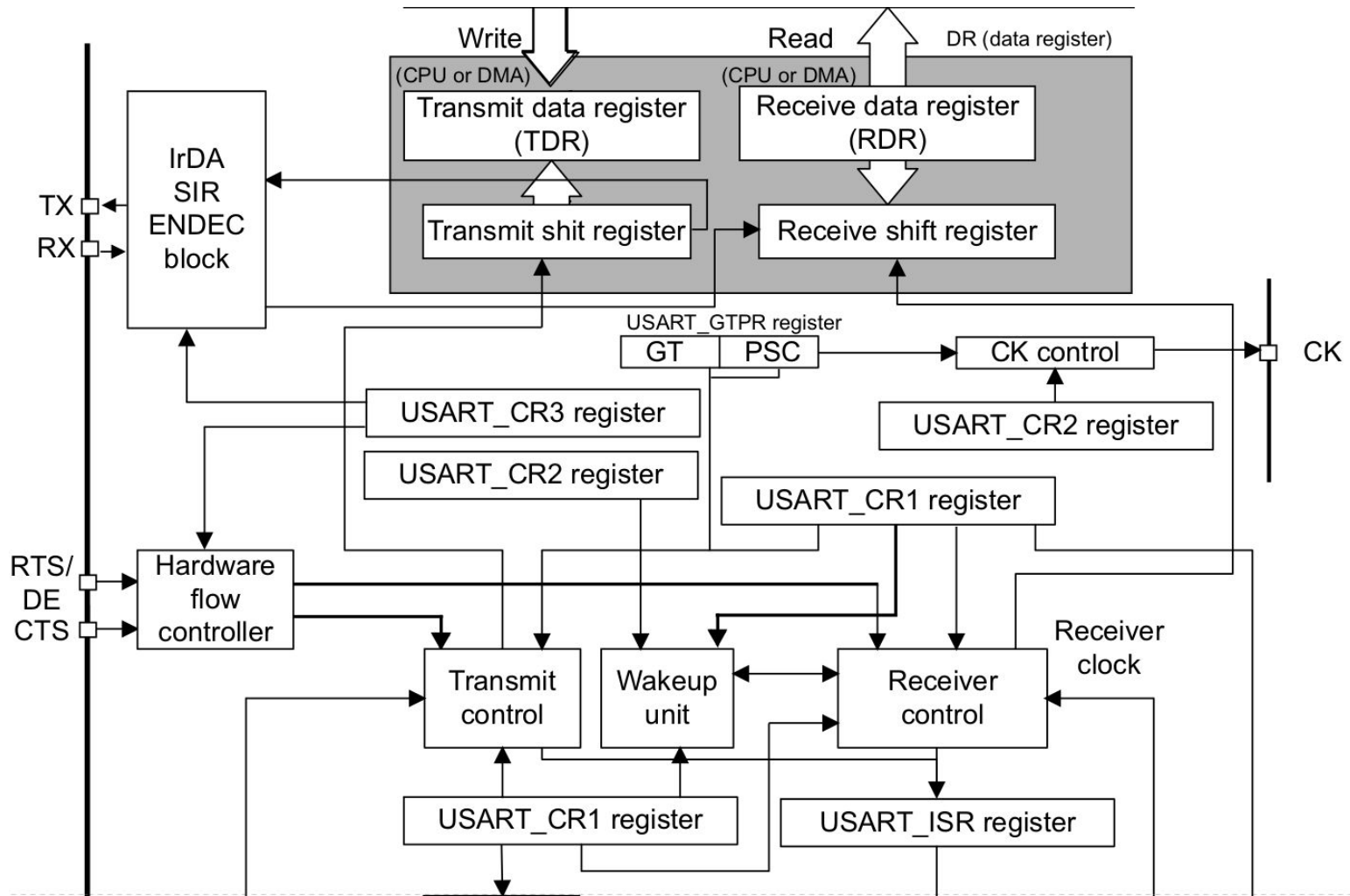
Duplex

Full-duplex

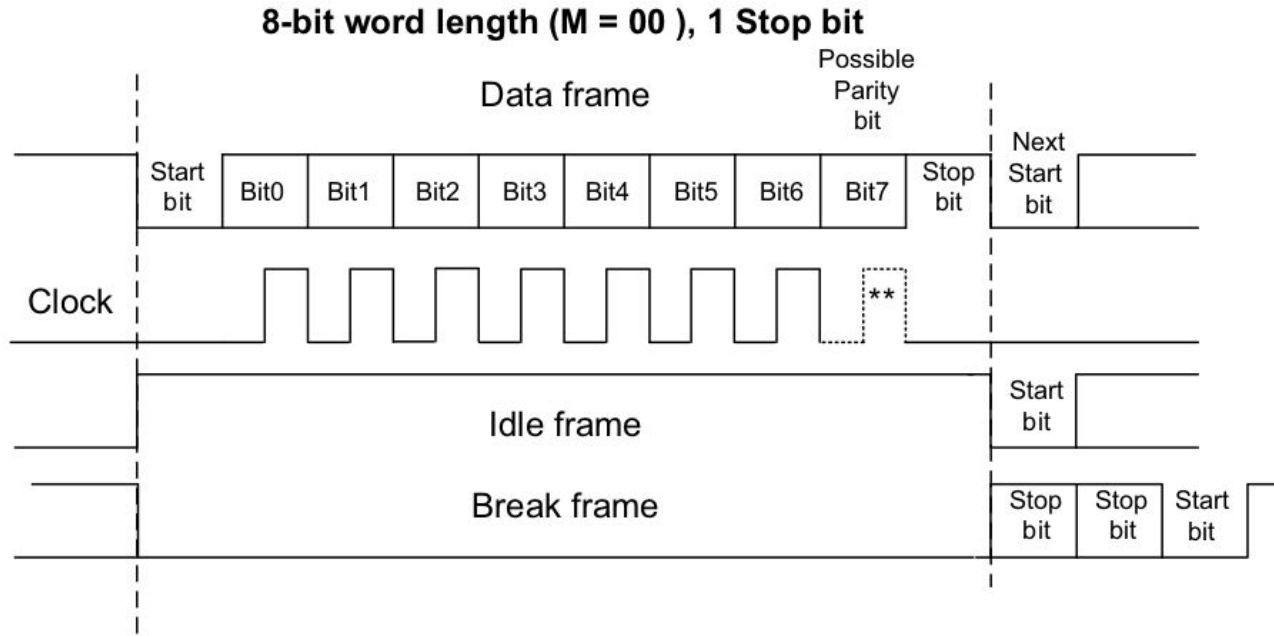
Half-duplex

Universal synchronous asynchronous receiver transmitter (USART). Main features

- Full-duplex asynchronous communications
- Baud rate of up to 6 Mbit/s
- Auto baud rate detection
- Programmable data word length (7, 8 or 9 bits)
- Programmable data order with MSB-first or LSB-first shifting
- Configurable stop bits (1 or 2 stop bits)
- Single-wire Half-duplex communications
- Continuous communications using DMA
- Swappable Tx/Rx pin configuration



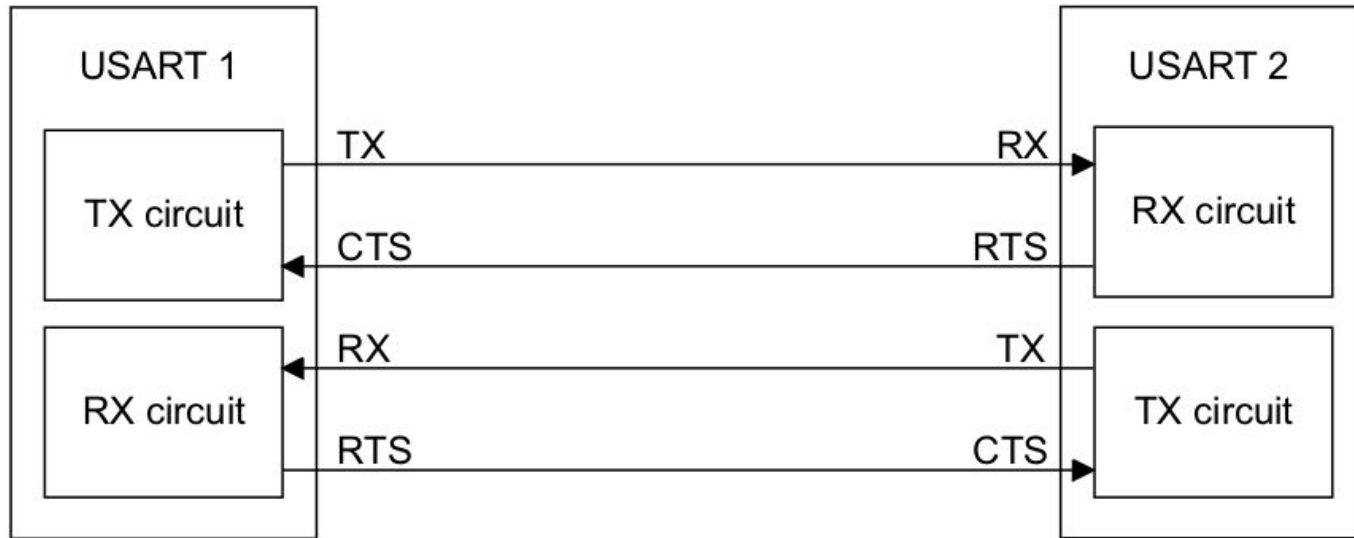
USART. Output waveform



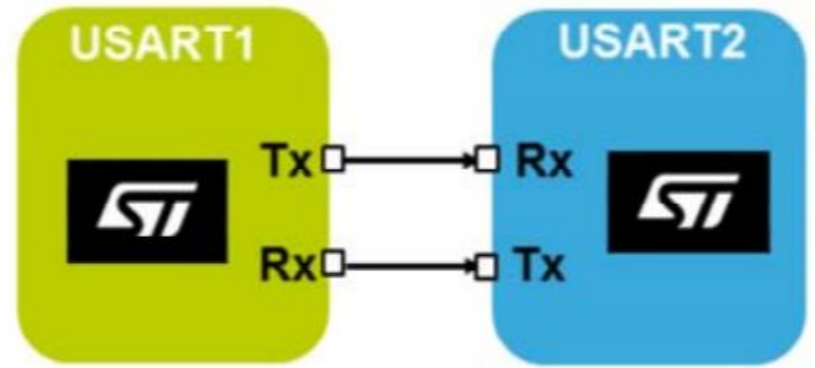
USART. Specific features

- Hardware flow control for modem
- Continuous communication using DMA
- Multiprocessor communication
- Synchronous mode
- Smartcard mode
- Single-wire Half-duplex communication
- IrDA SIR ENDEC block
- LIN mode
- Modbus communication
- Auto baud rate detection
- Receiver timeout interrupt

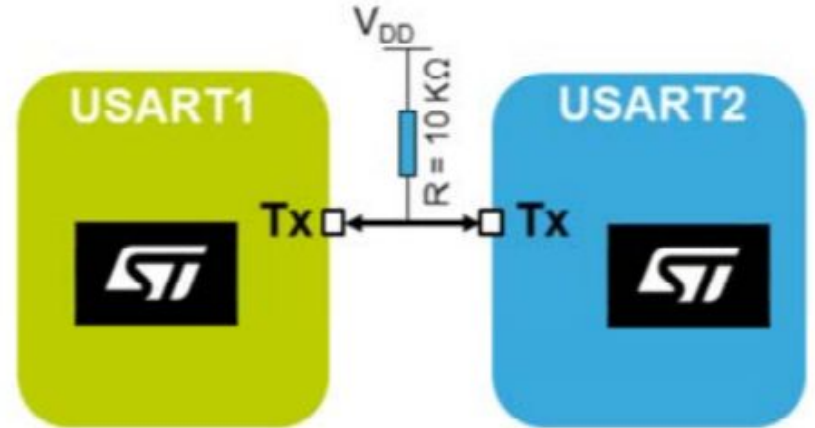
USART. Hardware flow control



USART. Full/half duplex modes

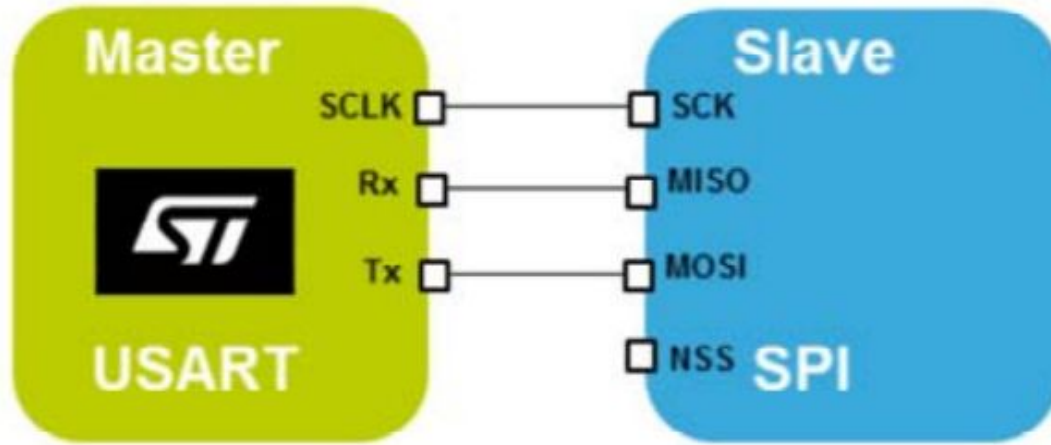


Full Duplex

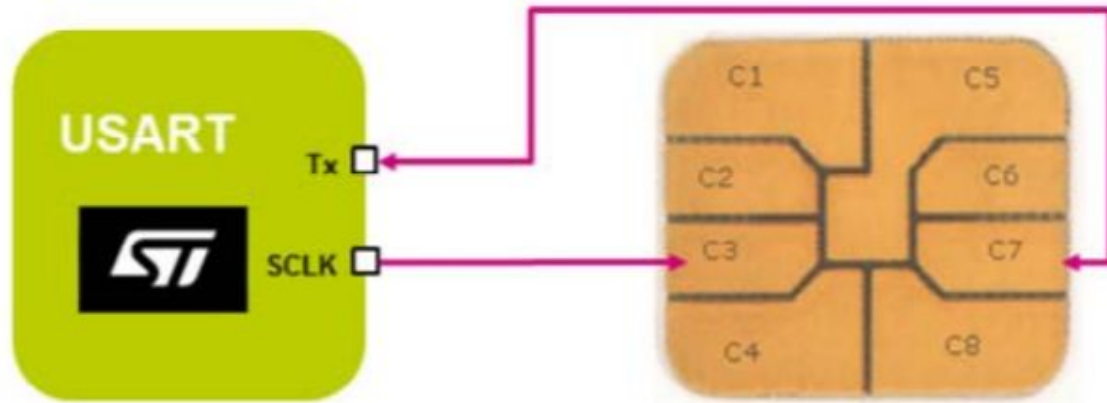


Half Duplex

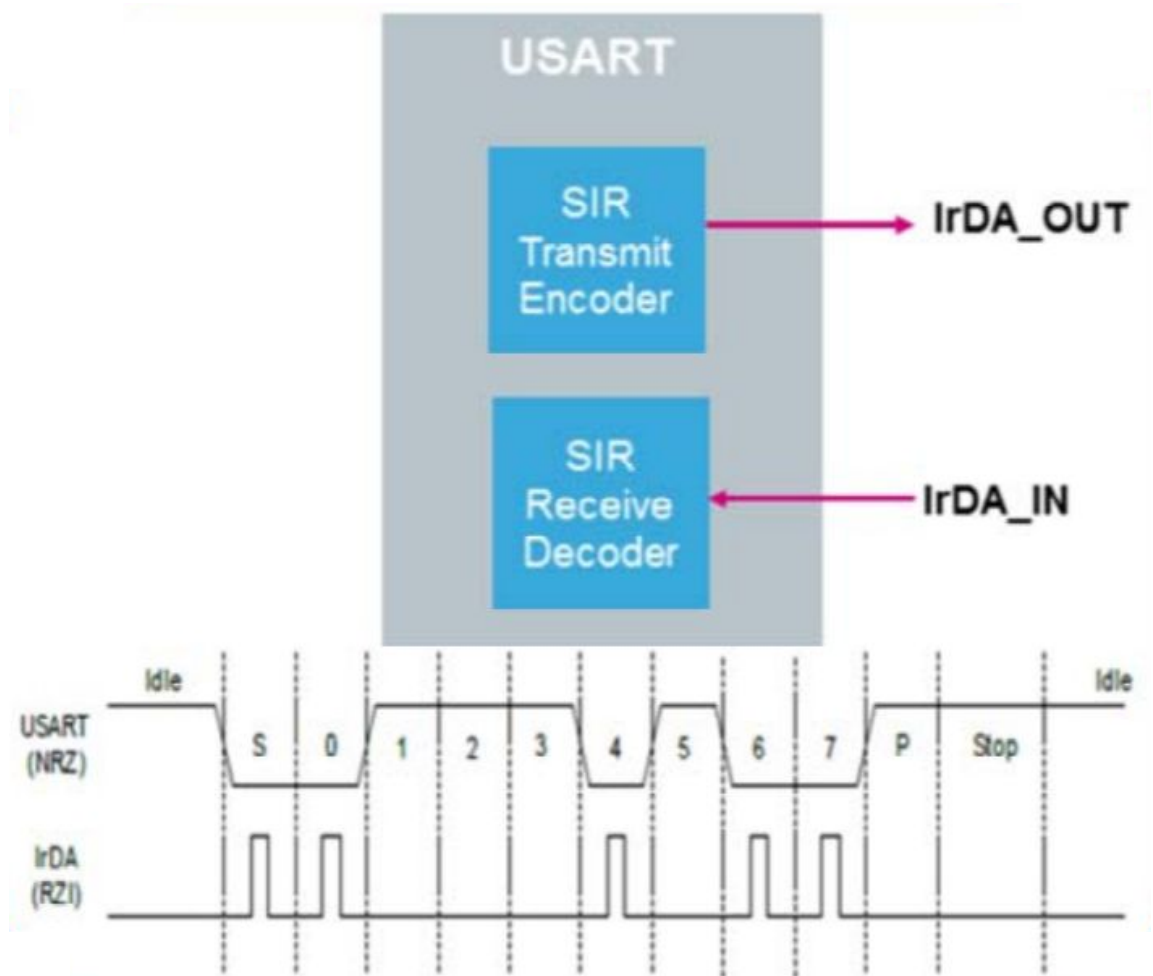
USART. SPI mode



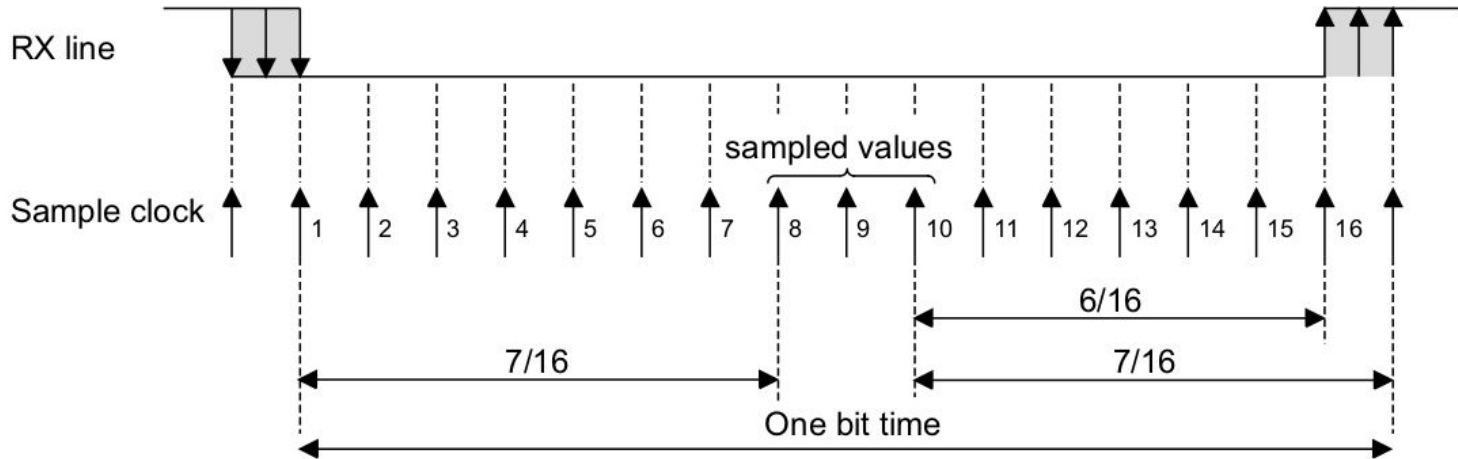
USART. Smartcards



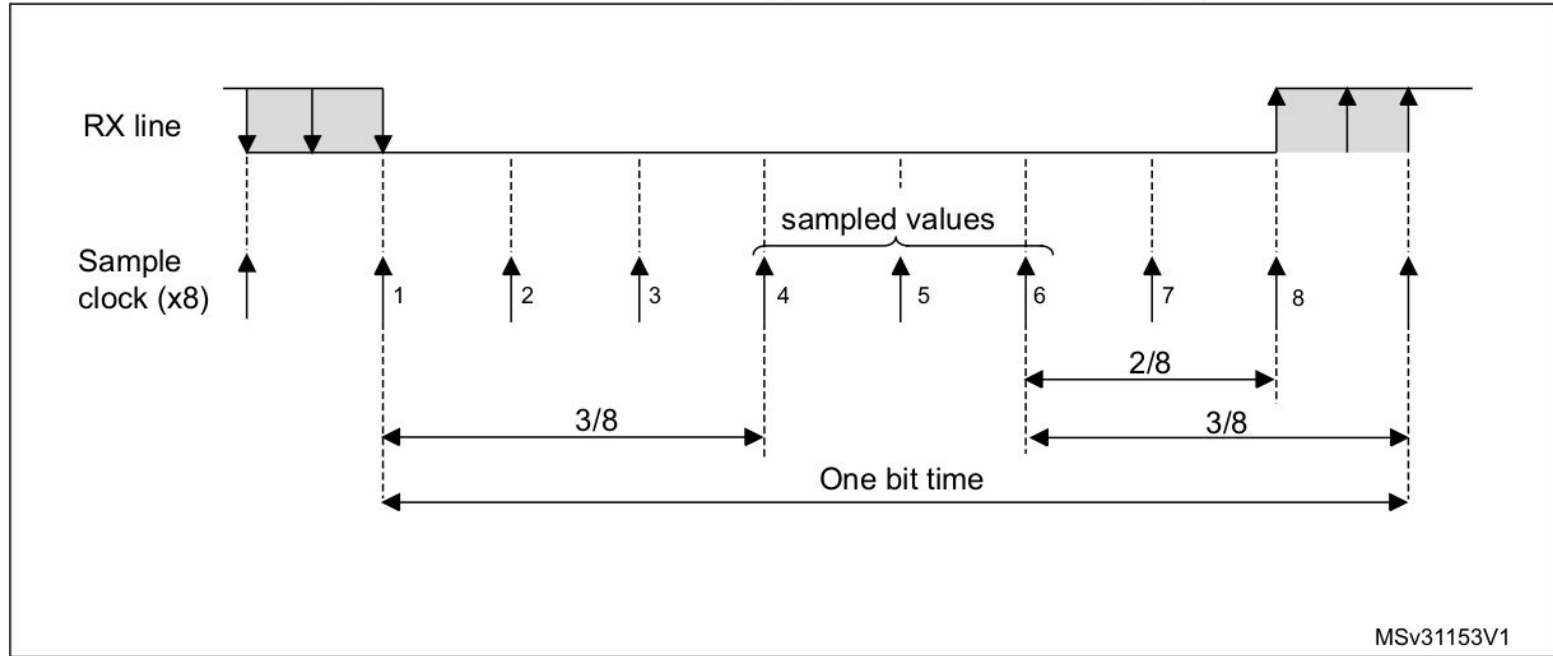
USART. IrDA



USART. Oversampling by 16



USART. Oversampling by 8



UART. Noise detection

Sampled value	NE status	Received bit value
000	0	0
001	1	0
010	1	0
011	1	1
100	1	0
101	1	1
110	1	1
111	0	1

USART. Baud rate generation

In case of oversampling by 16, the equation is:

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

In case of oversampling by 8, the equation is:

$$\text{Tx/Rx baud} = \frac{2 \times f_{CK}}{\text{USARTDIV}}$$

USART. Baud rate generation

To obtain 9600 baud with $f_{CK} = 8 \text{ MHz}$.

In case of oversampling by 16:

$$\text{USARTDIV} = 8,000,000/9,600$$

$$\mathbf{BRR = USARTDIV = 833d = 0x341}$$

In case of oversampling by 8:

$$\text{USARTDIV} = 2 * 8,000,000/9,600$$

$$\text{USARTDIV} = 1,666.66 \text{ (1,667d = 683h)}$$

$$\text{BRR}[3:0] = 3h \gg 1 = 1h$$

$$\mathbf{BRR = 0x681}$$

USART. USART baud rate generation

Baud rate		Oversampling by 16 (OVER8 = 0)			Oversampling by 8 (OVER8 = 1)		
S.No	Desired	Actual	BRR	% Error = (Calculated - Desired)B.Rate/ Desired B.Rate	Actual	BRR	% Error
2	2.4 KBps	2.4 KBps	0x4E20	0	2.4 KBps	0x9C40	0
3	9.6 KBps	9.6 KBps	0x1388	0	9.6 KBps	0x2710	0
4	19.2 KBps	19.2 KBps	0x9C4	0	19.2 KBps	0x1384	0
5	38.4 KBps	38.4 KBps	0x4E2	0	38.4 KBps	0x9C2	0
6	57.6 KBps	57.62 KBps	0x341	0.03	57.59 KBps	0x681	0.02
7	115.2 KBps	115.11 KBps	0x1A1	0.08	115.25 KBps	0x340	0.04
8	230.4 KBps	230.76KBps	0xD0	0.16	230.21 KBps	0x1A0	0.08
9	460.8 KBps	461.54KBps	0x68	0.16	461.54KBps	0xD0	0.16
10	921.6KBps	923.07KBps	0x34	0.16	923.07KBps	0x64	0.16
11	2 MBps	2 MBps	0x18	0	2 MBps	0x30	0
12	3 MBps	3 MBps	0x10	0	3 MBps	0x20	0
13	4MBps	N.A	N.A	N.A	4MBps	0x14	0
14	5MBps	N.A	N.A	N.A	5052.63KBps	0x11	1.05
15	6MBps	N.A	N.A	N.A	6MBps	0x10	0

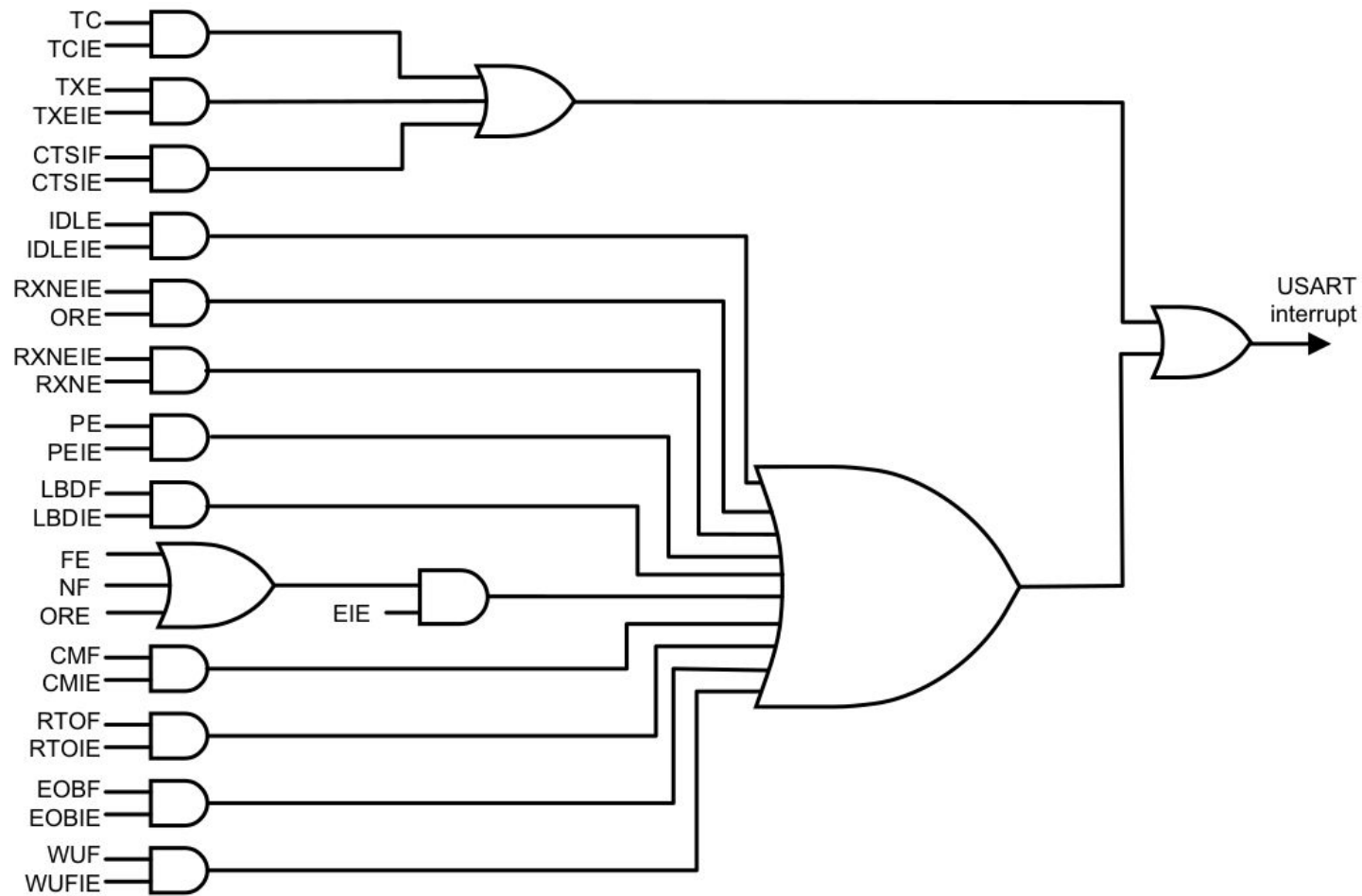
USART. Parity control

M bits	PCE bit	USART frame ⁽¹⁾
00	0	SB 8-bit data STB
00	1	SB 7-bit data PB STB
01	0	SB 9-bit data STB
01	1	SB 8-bit data PB STB
10	0	SB 7-bit data STB
10	1	SB 6-bit data PB STB

1. Legends: SB: start bit, STB: stop bit, PB: parity bit. In the data register, the PB is always taking the MSB position (9th, 8th or 7th, depending on the M bits value).

USART. Interrupts

Interrupt event	Event flag	Enable Control bit
Transmission Complete	TC	TCIE
Receive data register not empty (data ready to be read)	RXNE	RXNEIE
Overrun error detected	ORE	
Idle line detected	IDLE	IDLEIE
Parity error	PE	PEIE
LIN break	LBDF	LBDIE
Noise Flag, Overrun error and Framing Error in multibuffer communication.	NF or ORE or FE	EIE
Character match	CMF	CMIE
Receiver timeout	RTOF	RTOIE
End of Block	EOBF	EOBIE
Wakeup from Stop mode	WUF ⁽¹⁾	WUFIE



USART. Registers

- USART_CR1
- USART_CR2
- USART_CR3
- USART_BRR - Baud rate register
- USART_GTPR - Guard time and prescaler register
- USART_RTOR - Receiver timeout register
- USART_RQR - Request register
- USART_ISR - Interrupt and status register
- USART_ICR - Interrupt flag clear register
- *USART_RDR* - Receive data register
- *USART_TDR* - Transmit data register