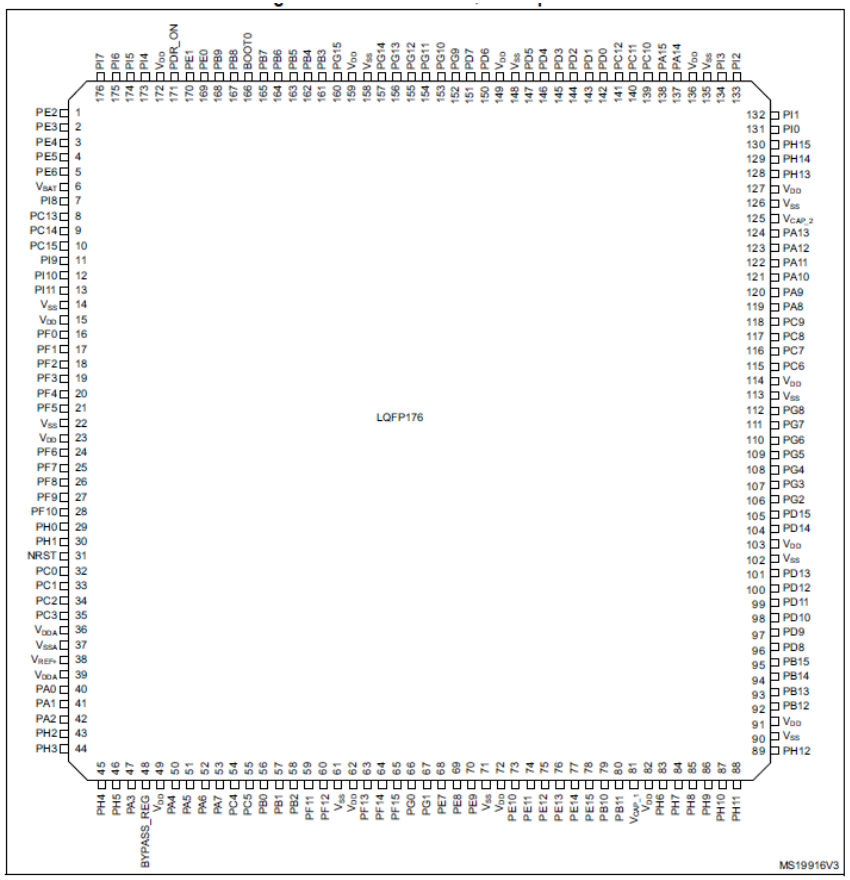
STM32F407 USART/UART



한국산업기술대학교 메카트로닉스공학과
마이크로컴퓨터응용
담당교수: 남윤석

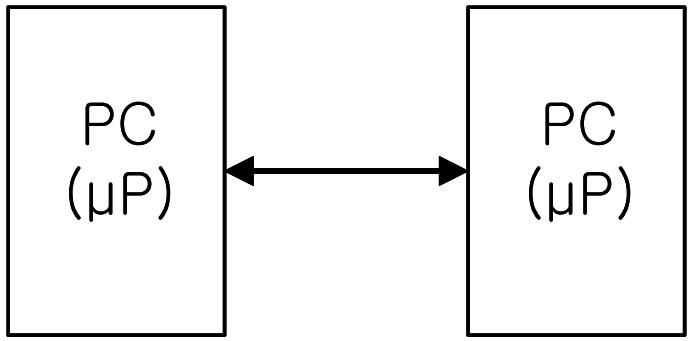
0. 개요

0.1 데이터(정보) 통신 정의

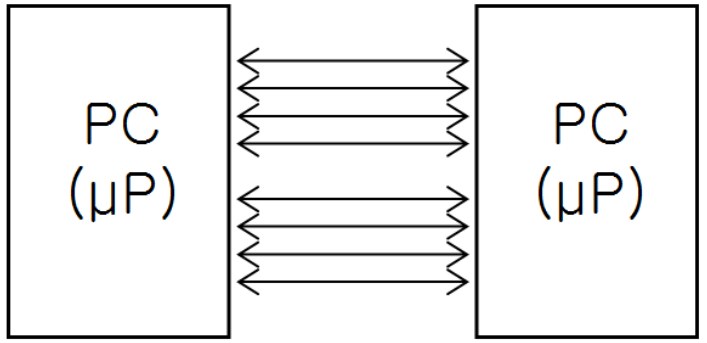
: 컴퓨터와 마이크로프로세서, 그리고 IC 칩들 사이에서 서로 데이터를 주고 받는 것

0.2 데이터(정보) 통신 분류

0.2.1 데이터 선로수에 의한 분류

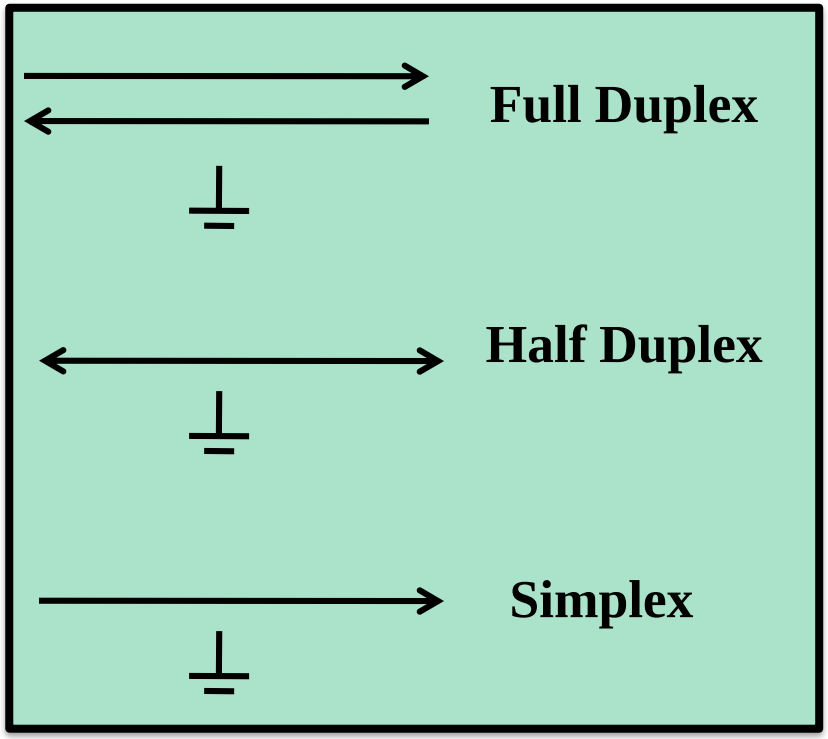


(a) 직렬 통신

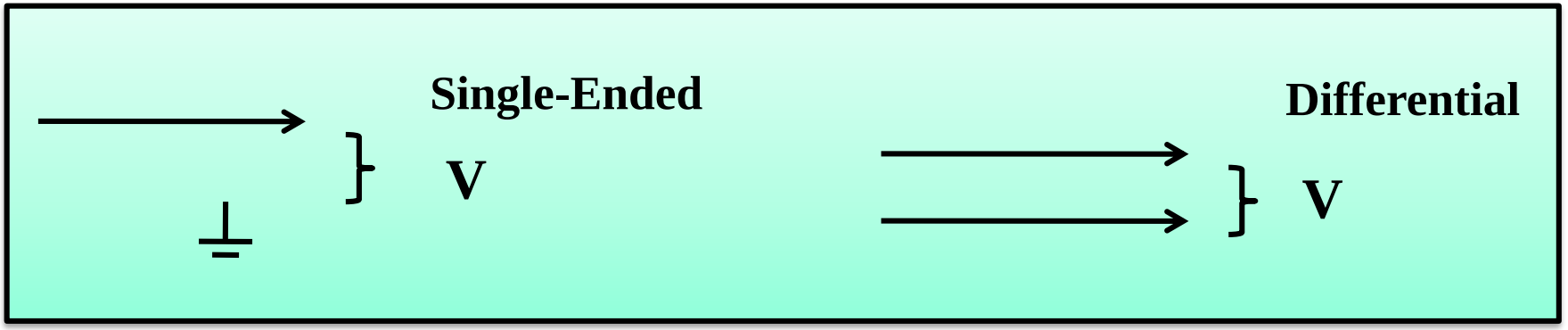


(b) 병렬 통신(8비트 통신의 경우)

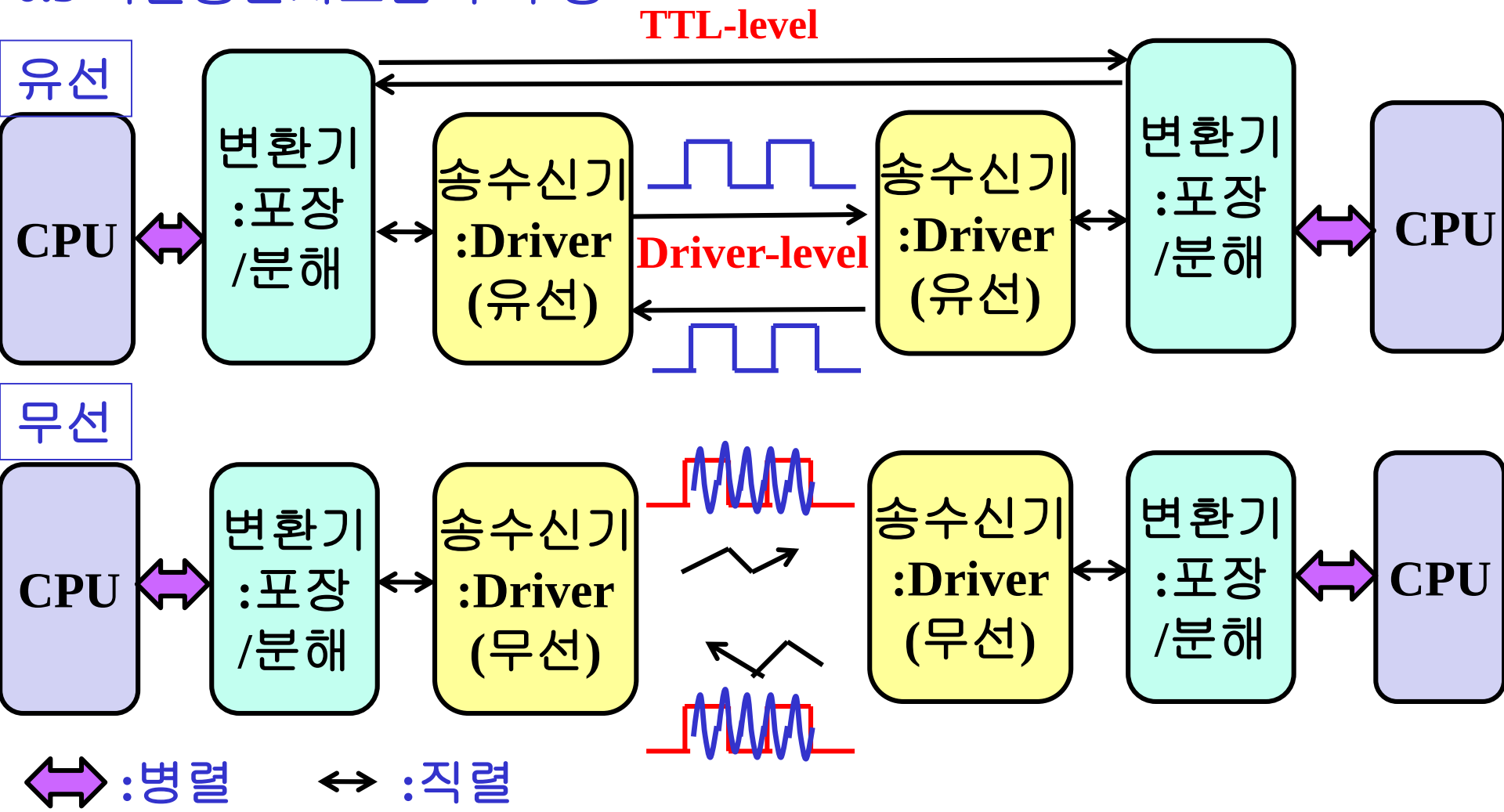
0.2.2 데이터 흐름 방법에 의한 분류



0.2.3 한 비트의 데이터를 전송하기 위한 선 수에 의한 분류



0.3 직렬통신시스템의 구성



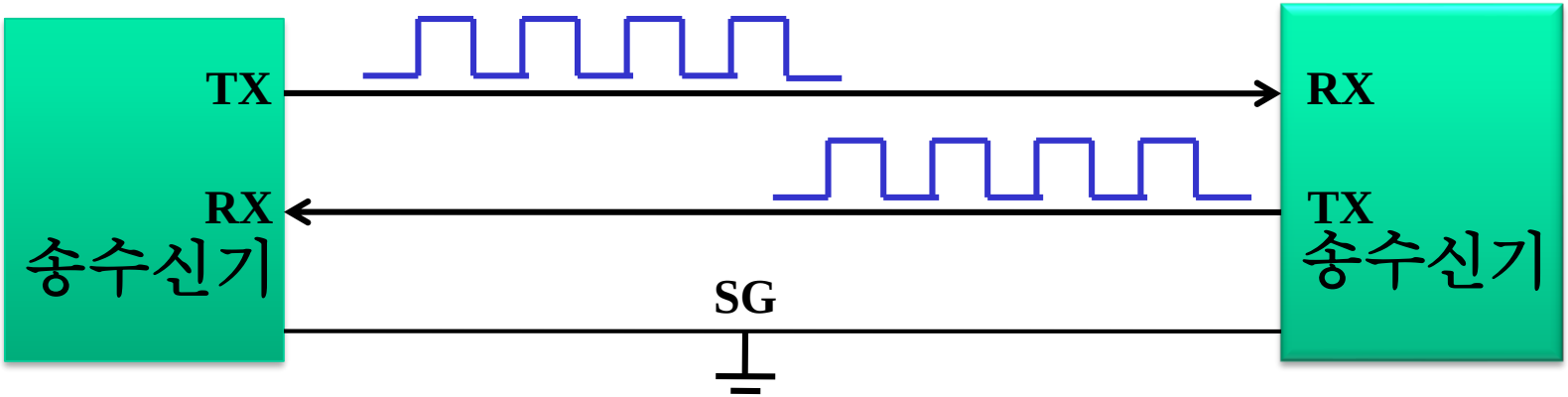
0.4 직렬데이터 통신시스템의 종류

- USART, USB, LAN, BlueTooth, CAN, I²C, SPI …

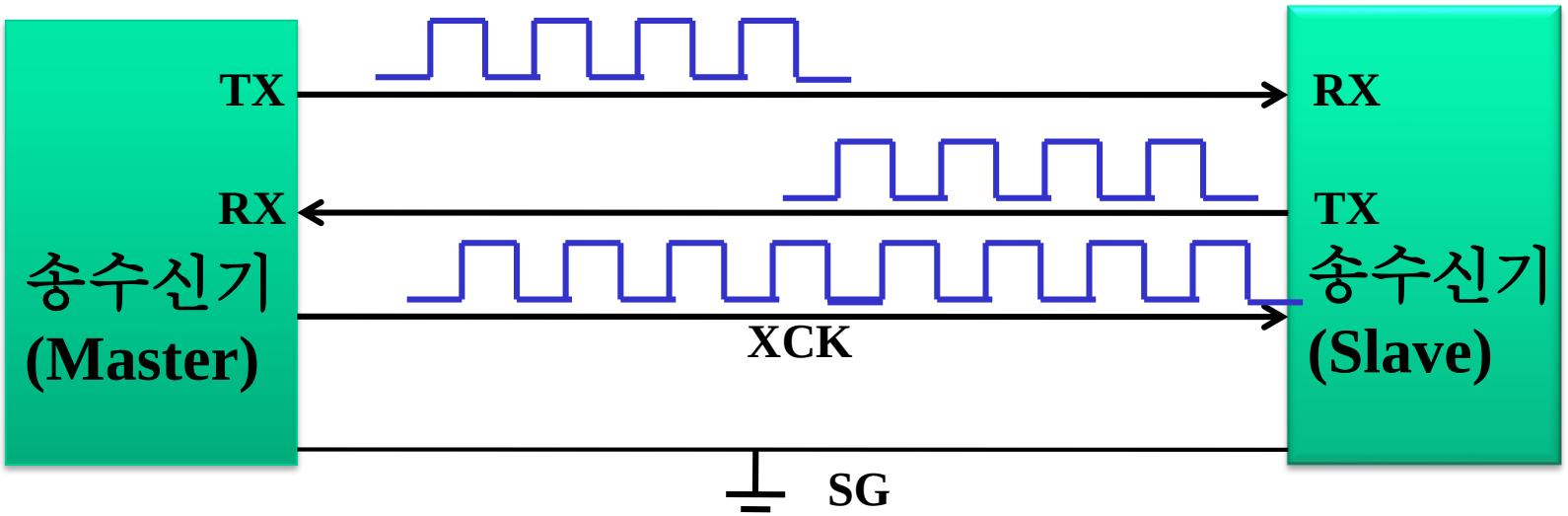
1. USART(Universal Synchronous/Asynchronous Receiver/Transmitter)

: 데이터를 직렬 또는 병렬로 변환시켜주는 물리적 회로중의 한가지로, PC, 마이크로프로세서와 주변장치들 간에 직렬(Serial) 포트를 이용한 통신에 주로 사용

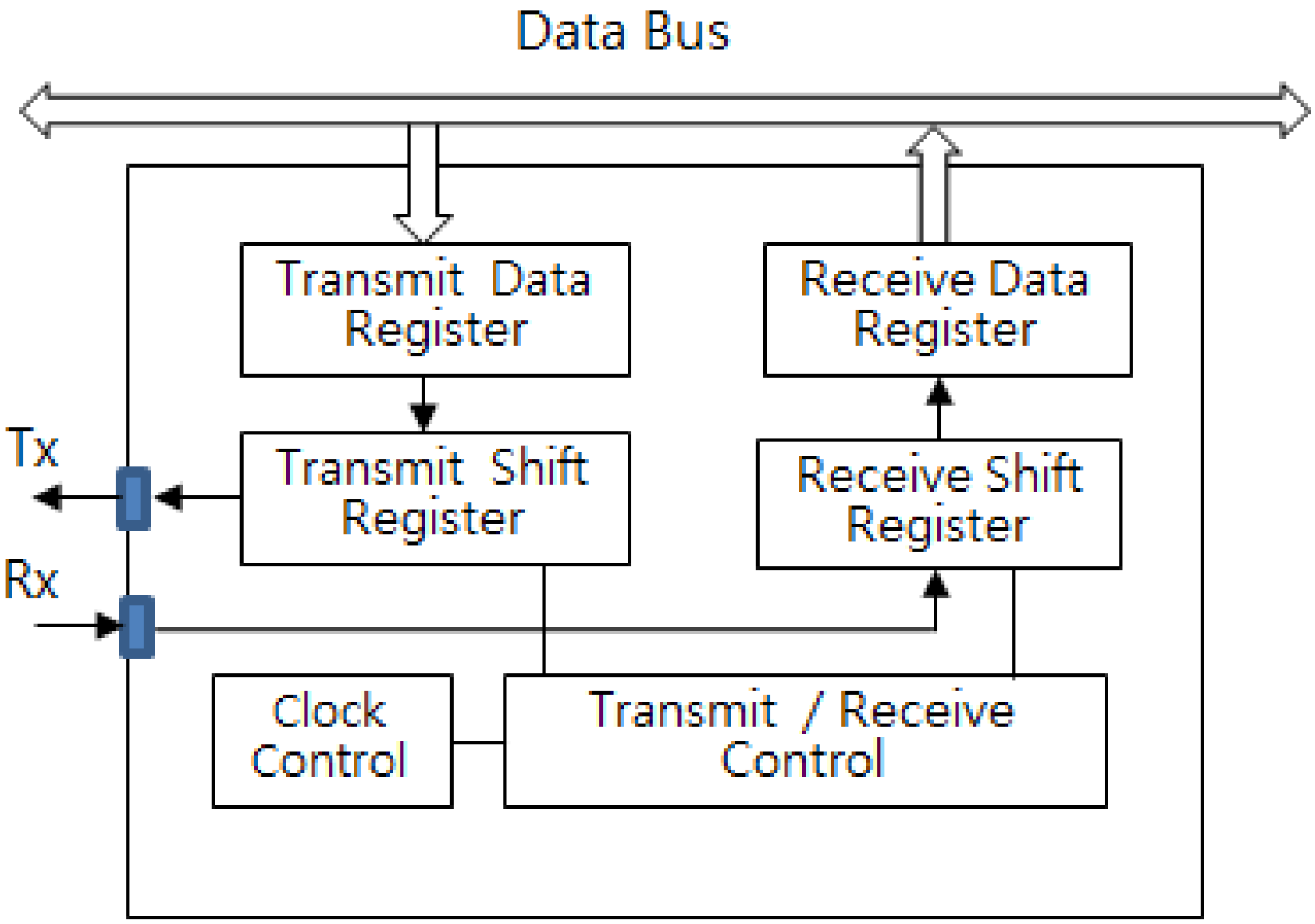
- **Asynchronous mode**



- **Synchronous mode**

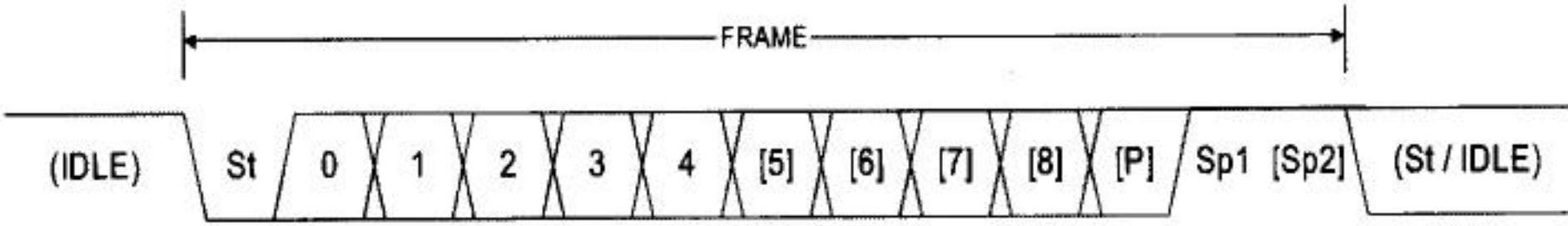


1.1 USART의 일반적 HW 구조



1.2 USART Data Frame Format (Protocol)

데이터 전송 방향



St	Start bit, always low.
(n)	Data bits (0 to 8).
P	Parity bit. Can be odd or even.
Sp	Stop bit, always high.
IDLE	No transfers on the communication line (RxD or TxD). An IDLE line must be high.

1.3 USART 데이터 전송을 위한 클럭의 조건

- 비동기 통신에서는 송신부와 수신부의 클럭이 동기되지 않으므로 송신부와 수신부는 서로 별개의 클럭으로 동작
- 송신부의 클럭과 수신부의 클럭 주파수는 서로 일치해야 함

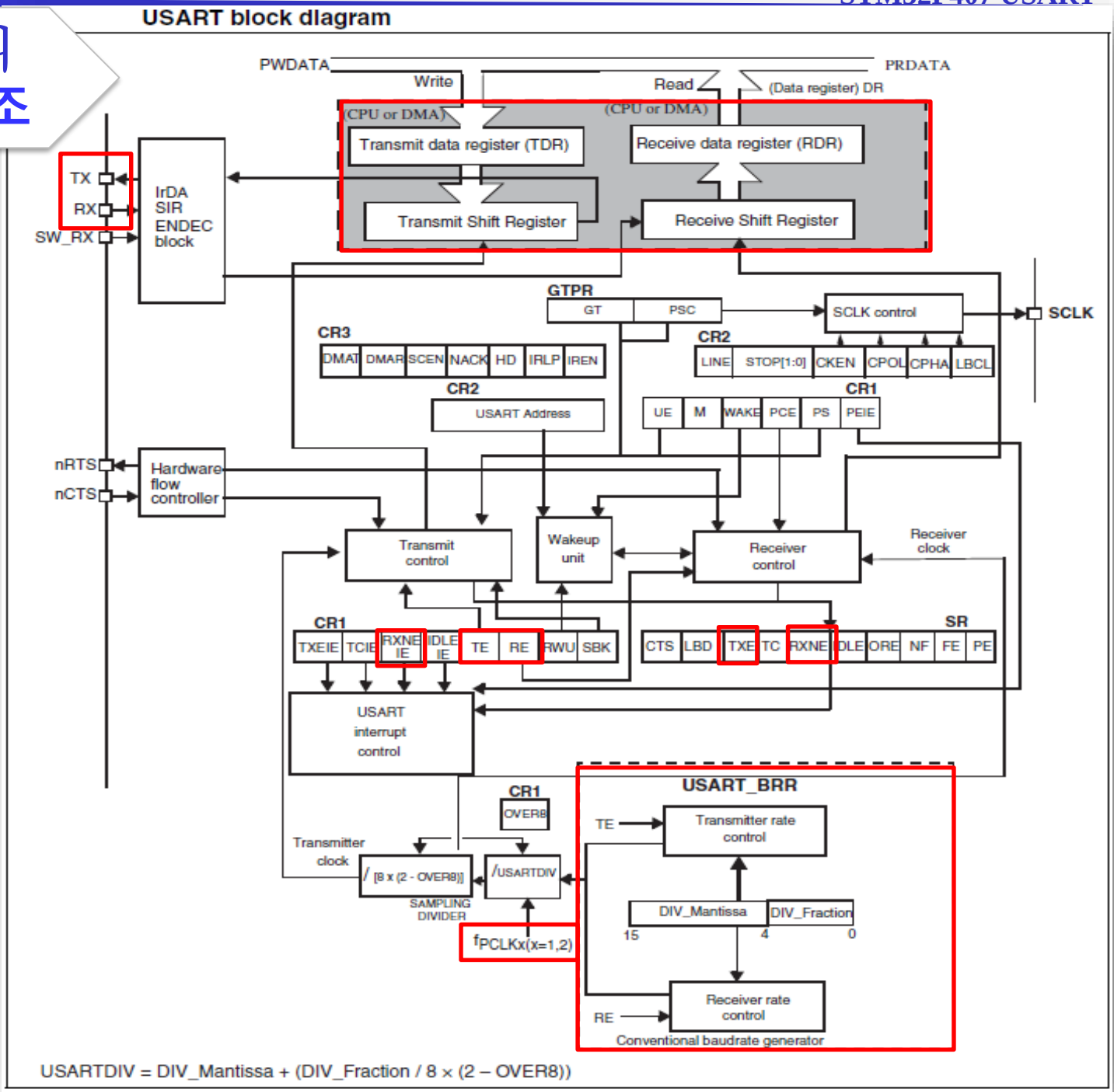
1.4 보 레이트 (Baud Rate)

- 보레이트: 데이터의 전송 속도 단위. 초당 전송되는 비트수
- **bps (bits per second)**와 같음

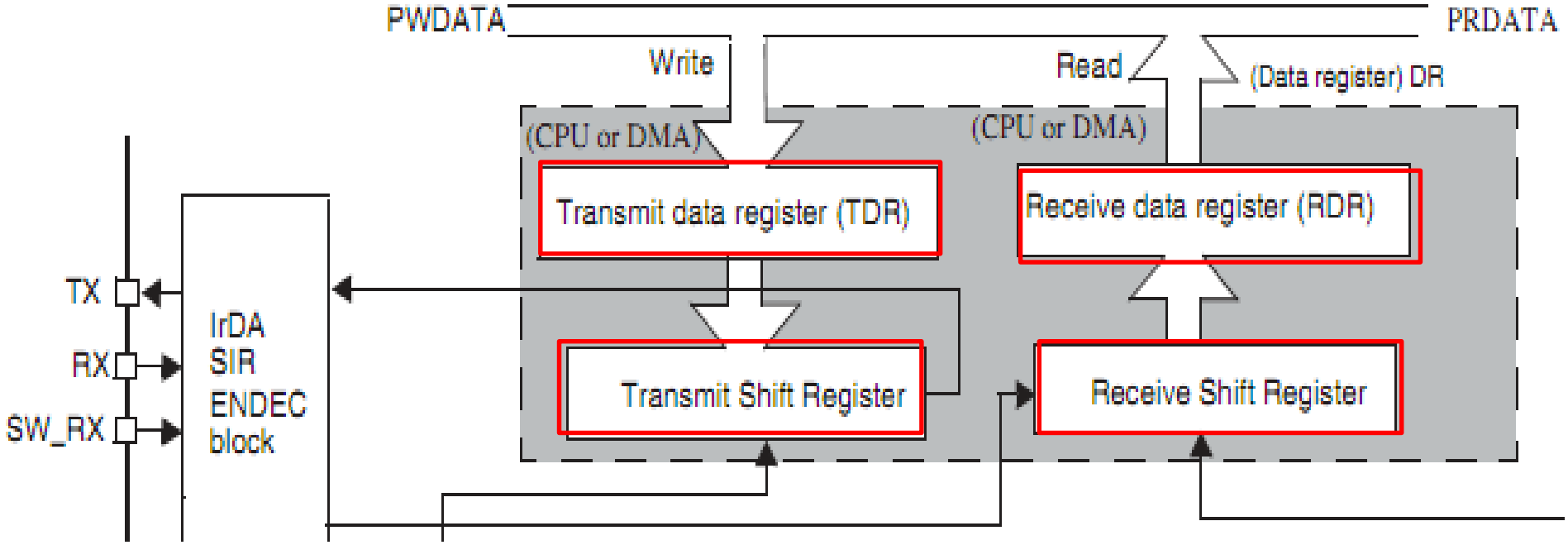
1.5 문자 송신 및 수신

- 데이터 송신시에 하위 비트부터 TX 핀을 통해 송신
- 수신시에도 하위 비트부터 RX 핀을 통해 수신

2. STM32F407의 USART HW 구조

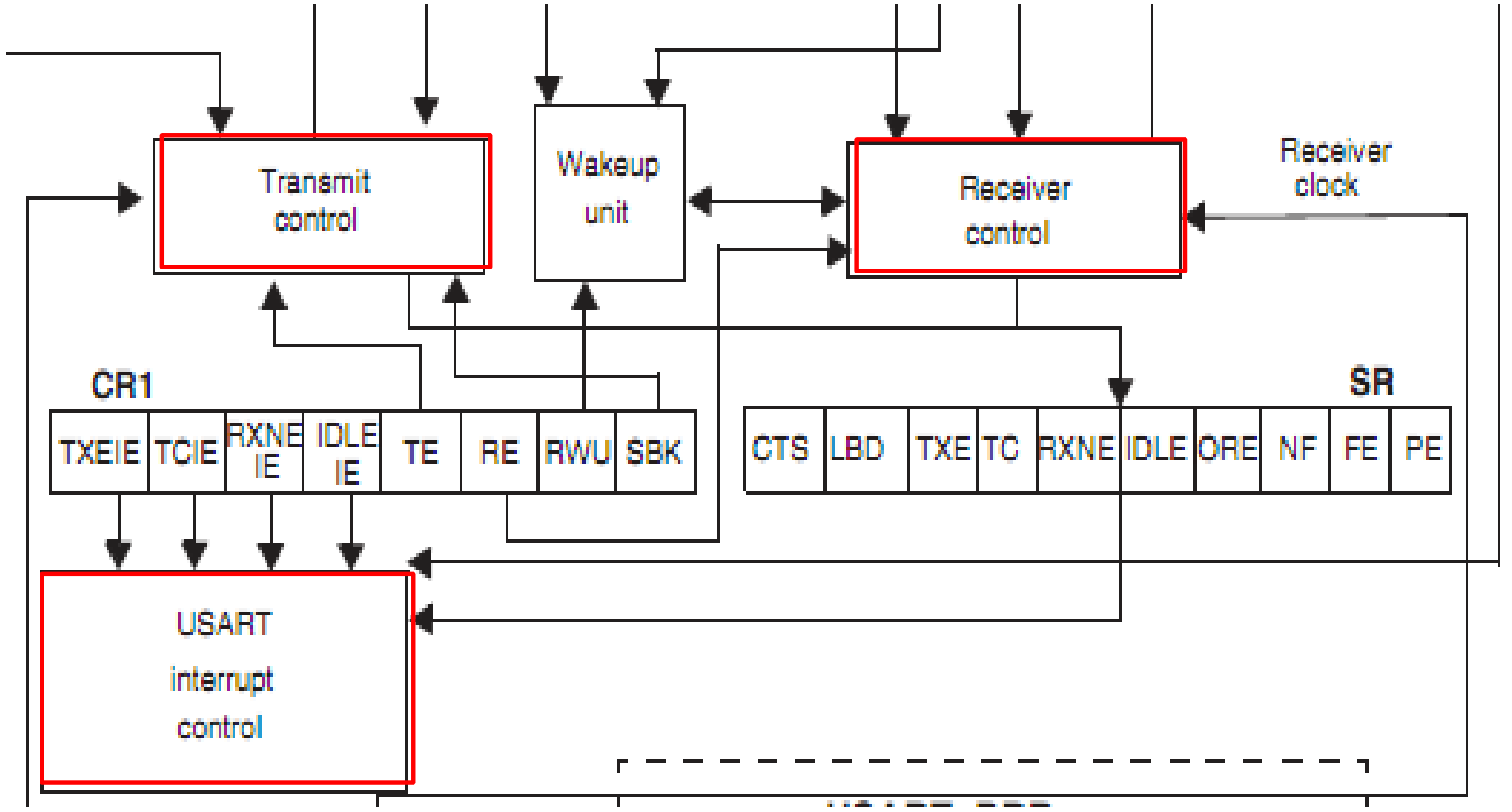


2.1 USART 송수신 영역 HW 구조

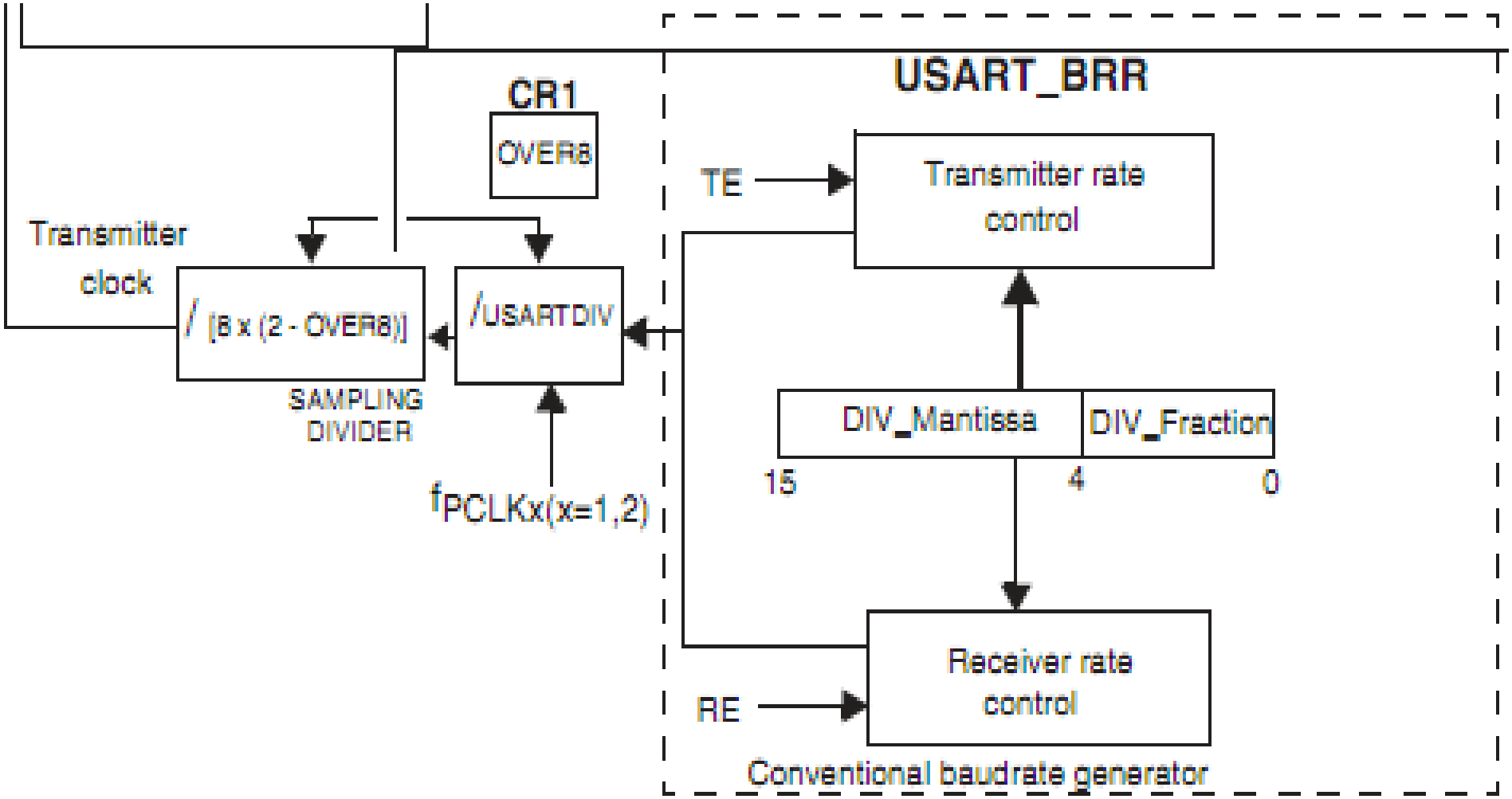


- **TX:** 데이터를 출력하는 핀(데이터출력 없을 시 **'H'** 상태), **single-wire** 나 **smartcard mode**에서 데이터 송수신용**(TX,RX)**으로 사용
- **RX:** 데이터를 입력받기 위한 핀
- **SW_RX: single-wire** 모드나 스마트카드 모드에서 데이터 수신용 핀으로도 사용
- **SCLK:** 동기모드에서 클럭 출력 핀

2.2 USART 제어 영역 HW 구조

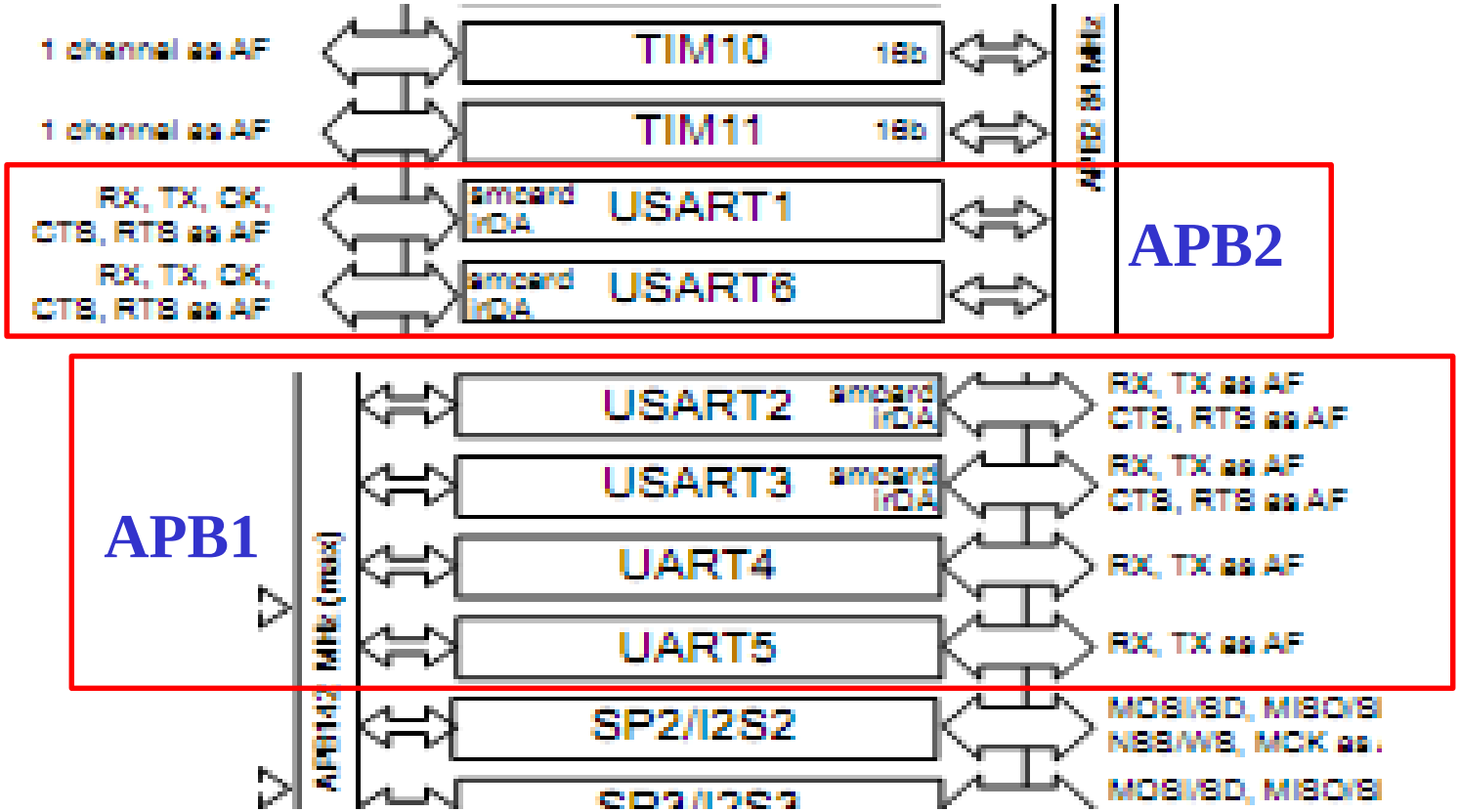


2.3 USART 송수신속도제어 영역 HW 구조



3. USART의 구성 및 특징

(1) 4 x USART / 2 x UART



(2) Full duplex, asynchronous communication 기능

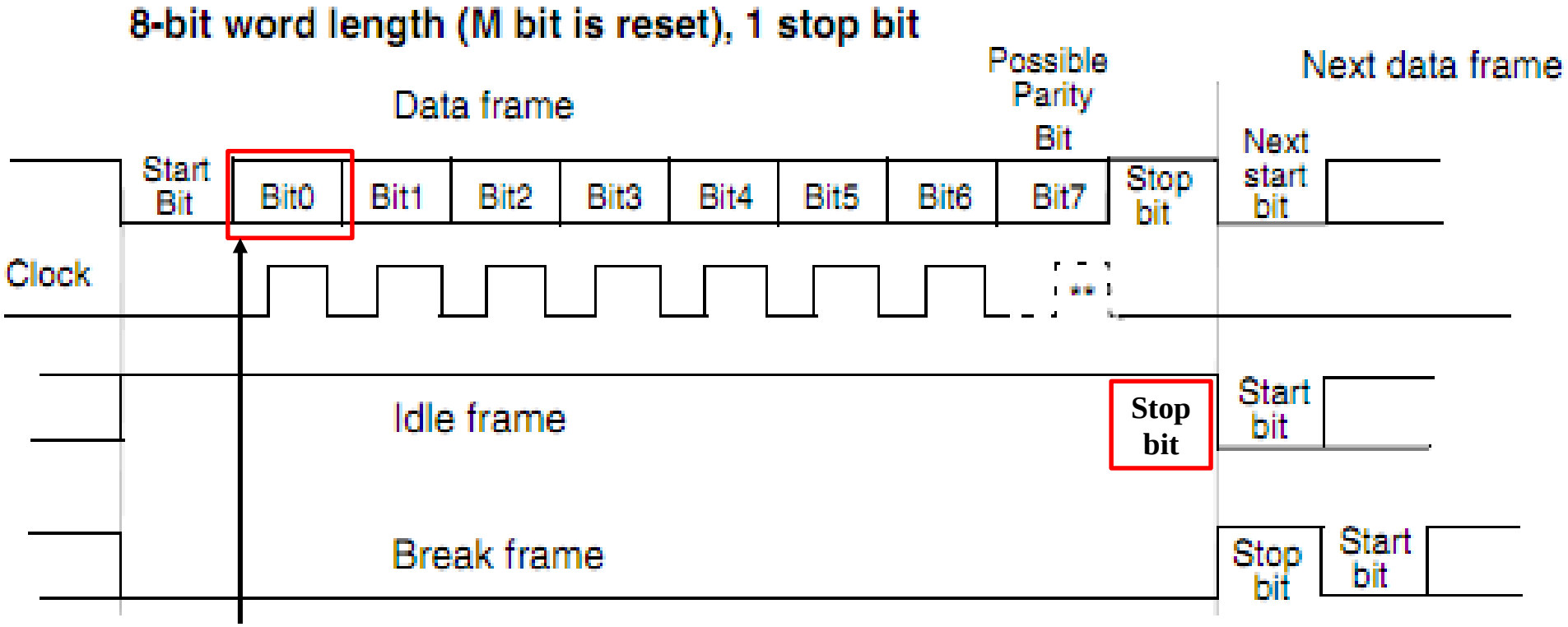
(3) Programmable data word length (8 or 9 bits)

(4) Configurable stop bits - support for 0.5, 1, 1.5 or 2 stop bits

(5) 4.5 Mbit/s 까지의 보레이트 구현 가능

(6) Event(전송 status flag)의 검출 가능
① 수신 버퍼 가득 참 **(Receive Buffer Full)**
② 송신 버퍼 비어 있음 **(Transmit Buffer Empty)**
③ 송신 완료 **(End of Transmission)**
(7) Parity control: TX parity bit, Checks parity of received data byte
(8) 4 Error detection flag
- Overrun error, Noise error, Frame error, Parity error
(9) 10개의 인터럽트 소스와 대응되는 flag 발생
① **Receive Buffer Full** ② **Transmit Buffer Empty**
③ **End of Transmission** ④ **Idle line received**
⑤ **Overrun error** ⑥ **Noise error** ⑦**Frame error** ⑧**Parity error**
⑨ **CTS change**
⑩ **LIN(Local Interconnection Network) break detection**
(10) 스마트 카드 프로토콜 지원
(11) IrDA (Infrared Data Association) 지원
(12) Single-wire half-duplex communication 지원
(13) Configurable multi-buffer communication using DMA
(14) Transmitter clock output for synchronous transmission

3.1 USART Data format



* 한 바이트 데이터의 **LSB** 부터 송신

3.1.1 Parity Bit

- 전송시 에러발생여부를 **check**하기 위한 비트
- 전송시 데이터에 추가로 **1**비트를 붙여 전송

- Even parity 방법: data + parity bit 의 '1'의 총수가 짝수가 되도록 Parity bit를 '0' 또는 '1'로 설정

- Odd parity 방법: data + parity bit 의 '1'의 총수가 홀수가 되도록 Parity bit를 '0' 또는 '1'로 설정

- 수신기에서 수신시 '1'의 개수를 카운트하여 짝수인지 홀수 인지 판단하여 전송에러가 발생했는지 판단
- 예제: **0x31**전송시

Even parity : 0b00110001 + 1(parity)

Odd parity : 0b00110001 + 0(parity)

Table 120. Frame formats

M bit	PCE bit	USART frame(1)
0	0	ǀ SB ǀ 8 bit data ǀ STB ǀ
0	1	ǀ SB ǀ 7-bit data ǀ PB ǀ STB ǀ
1	0	ǀ SB ǀ 9-bit data ǀ STB ǀ
1	1	ǀ SB ǀ 8-bit data PB ǀ STB ǀ

1. Legends: SB: start bit, STB: stop bit, PB: parity bit.

3.2 USART mode

USART mode configuration

Table 122. USART mode configuration(1)

USART modes	USART1	USART2	USART3	UART4	UART5	USART6
Asynchronous mode	X	X	X	X	X	X
Hardware flow control	X	X	X	NA	NA	X
Multibuffer communication (DMA)	X	X	X	X	X	X
Multiprocessor communication	X	X	X	X	X	X
Synchronous	X	X	X	NA	NA	X
Smartcard	X	X	X	NA	NA	X
Half-duplex (single-wire mode)	X	X	X	X	X	X
IrDA	X	X	X	X	X	X
LIN	X	X	X	X	X	X

1. X = supported; NA = not applicable.

3.3 Transmitter(TX)

- **Transmit enable bit (USARTx→CR1.TE)=1 이면, transmit shift register에 있는 데이터가 TX pin을 통해 출력됨**
- **Stop bit 선택**

- 1 stop bit: default value
- 2 Stop bits: single-wire and modem modes
- 0.5, 1.5 stop bit: Smartcard mode

- **TX 과정:**

(1) Enable USART by USART_CR1.UE=1
(2) Program USART_CR1.M to define the word length
(3) Program the number of stop bits in USART_CR2
(4) Select baud rate using USART_BRR
(5) USART_CR1.TE=1 to send an idle frame as first transmission
(6) Write the data to send in the USART_DR (this clears the TXE bit). Repeat this for each data
(7) After writing the last data into the USART_DR, wait until TC=1

• **Single byte communication**

- USART_DR에 데이터를 쓰면 TXE=0 이 됨

- 다음과 같은 작업이 실행되면 TXE =1이 됨

① TDR→shift register 후 TX시작

② TDR empty

-TXE=1 generates an interrupt if TXEIE=1

-하나의 frame이 송신(after the stop bit)되면 TXE=1, TC=1 됨

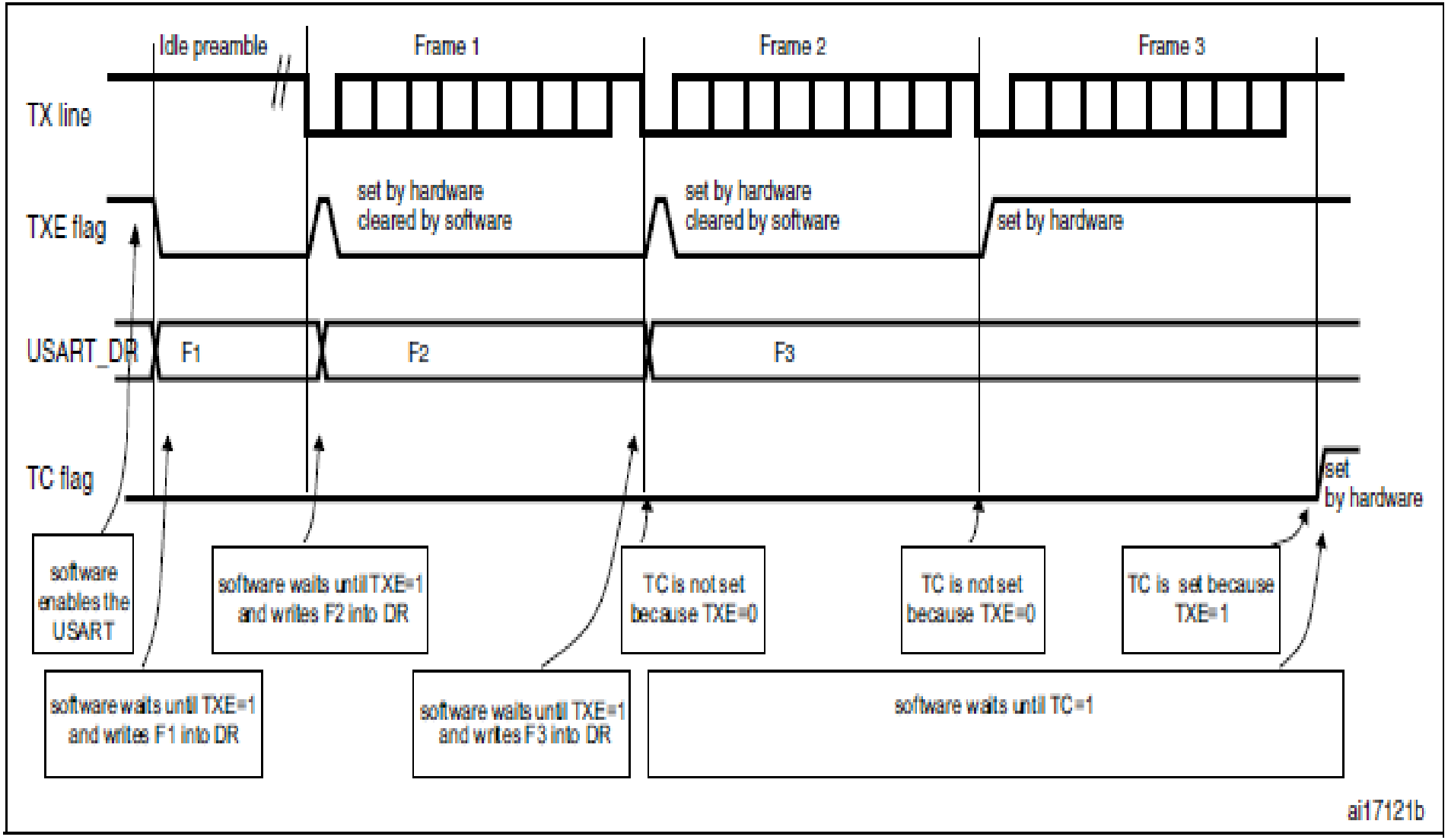
이때 USART_CR1.TCIE=1이면 인터럽트 발생

*** 다음의 경우에 TC=0 이 됨**

- Reading USART_SR

- Writing to USART_DR

Figure 249. TC/TXE behavior when transmitting



3.4 Receiver(RX)

• **RX 과정:**

(1) Enable USART by USART_CR1.UE=1
(2) Program USART_CR1.M to define the word length
(3) Program the number of stop bits in USART_CR2
(4) Select baud rate using USART_BRR
(5) USART_CR1.RE=1 는 수신기가 수신되는 start bit 를 찾도록 함

• 한 문자가 수신되었을 때
-RXNE=1 (이것은 shift register →RDR 을 의미)
- 인터럽트 발생 if RXNEIE=1
- 수신중 frame , noise, overrun error 발생되면 Error flags set
- USART_DR 을 읽으면 RXNE bit=0
***강제로 RXNE=0 도 가능**

3.5 USART Baud rate

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

- **USARTDIV is an unsigned fixed point number that is coded on the USART_BRR**
- **OVER8=0:소수영역 4 bits, DIV_fraction[3:0] in USART_BRR**
  OVER8=1:소수영역 3 bits, DIV_fraction[2:0] in USART_BRR

- **How to derive USARTDIV from USART_BRR when OVER8=0**

Ex1:

If DIV_Mantissa = 0d27 and DIV_Fraction = 0d12 (USART_BRR = 0x1BC), then

Mantissa (USARTDIV) = 0d27

Fraction (USARTDIV) = 12/16 = 0d0.75

Therefore USARTDIV = 0d27.75

Ex2:

To program USARTDIV = 0d25.62
This leads to: DIV_Fraction = 16*0d0.62 = 0d9.92
The nearest real number is 0d10 = 0xA
DIV_Mantissa = mantissa (0d25.620) = 0d25 = 0x19
Then, USART_BRR = 0x19A hence USARTDIV = 0d25.625

Ex3:

To program USARTDIV = 0d50.99
This leads to: DIV_Fraction = 16*0d0.99 = 0d15.84
The nearest real number is 0d16 = 0x10 => overflow of DIV_frac[3:0] => carry must be added up to the mantissa
DIV_Mantissa = mantissa (0d50.990 + carry) = 0d51 = 0x33
Then, USART_BRR = 0x330 hence USARTDIV = 0d51.000

Table 108. Error calculation for programmed baud rates at f_{PCLK} = 8 MHz or f_{PCLK} = 12 MHz, oversampling by 16(1)

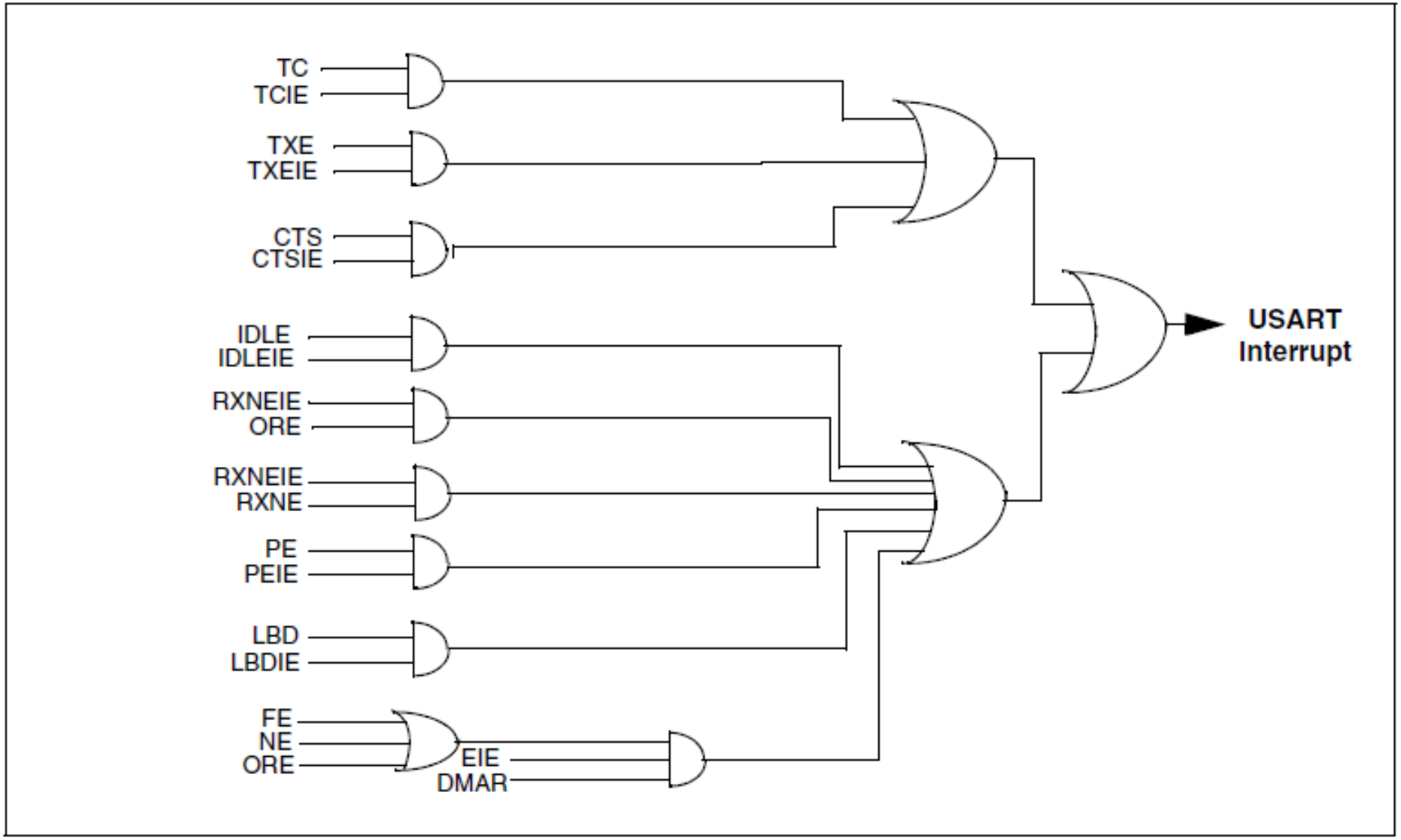
Oversampling by 16 (OVER8=0)							
Baud rate7		f_{PCLK} = 8 MHz			f_{PCLK} = 12 MHz		
S.No	Desired	Actual	Value programmed in the baud rate register	% Error = (Calculated - Desired) B.rate / Desired B.rate	Actual	Value programmed in the baud rate register	% Error
1	1.2 KBps	1.2 KBps	416.6875	0	1.2 KBps	625	0
2	2.4 KBps	2.4 KBps	208.3125	0.01	2.4 KBps	312.5	0
3	9.6 KBps	9.604 KBps	52.0625	0.04	9.6 KBps	78.125	0
4	19.2 KBps	19.185 KBps	26.0625	0.08	19.2 KBps	39.0625	0
5	38.4 KBps	38.462 KBps	13	0.16	38.339 KBps	19.5625	0.16
6	57.6 KBps	57.554 KBps	8.6875	0.08	57.692 KBps	13	0.16
7	115.2 KBps	115.942 KBps	4.3125	0.64	115.385 KBps	6.5	0.16
8	230.4 KBps	228.571 KBps	2.1875	0.79	230.769 KBps	3.25	0.16
9	460.8 KBps	470.588 KBps	1.0625	2.12	461.538 KBps	1.625	0.16
10	921.6 KBps	NA	NA	NA	NA	NA	NA
11	2 MBps	NA	NA	NA	NA	NA	NA
12	3 MBps	NA	NA	NA	NA	NA	NA

3.6 USART Interrupt

Table 121. USART Interrupt requests

Interrupt event	Event flag	Enable control bit
Transmit Data Register Empty	TXE	TXEIE
CTS flag	CTS	CTSIE
Transmission Complete	TC	TCIE
Received Data Ready to be Read	RXNE	RXNEIE
Overrun Error Detected	ORE	
Idle Line Detected	IDLE	IDLEIE
Parity Error	PE	PEIE
Break Flag	LBD	LBDIE
Noise Flag, Overrun error and Framing Error in multibuffer communication	NF or ORE or FE	EIE

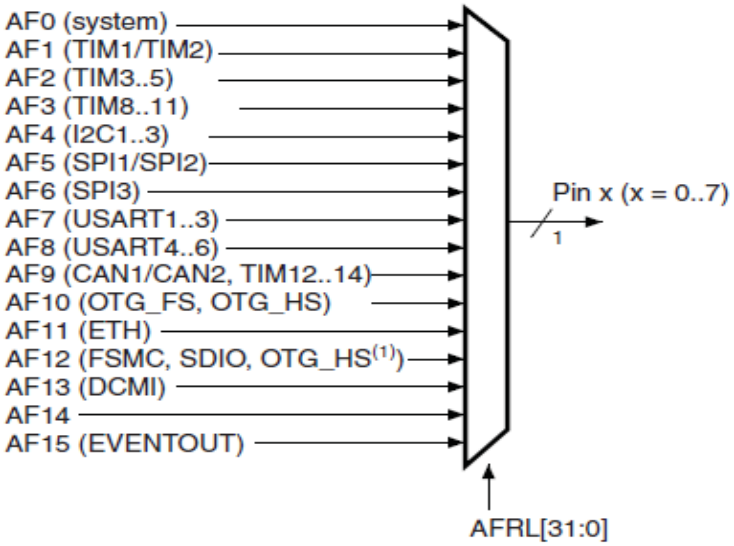
Figure 270. USART interrupt mapping diagram



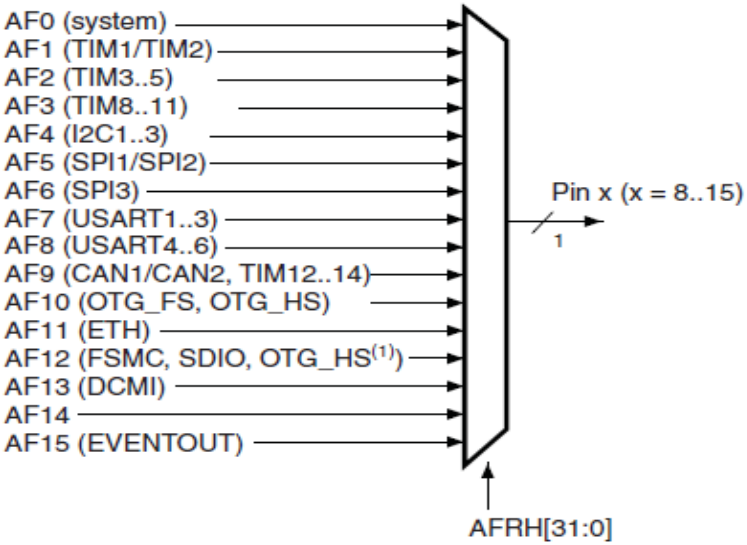
• Selecting an Alternate function

Selecting an alternate function on STM32F405xx/07xx and STM32F415xx/17xx

For pins 0 to 7, the GPIOx_AFRL[31:0] register selects the dedicated alternate function



For pins 8 to 15, the GPIOx_AFRH[31:0] register selects the dedicated alternate function



4. USART 레지스터

4.1 Register Map

Table 123. USART register map and reset values

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	USART_SR	Reserved																						CTS	LBD	TXE	TC	RXNE	IDLE	ORE	NF	FE	PE
	Reset value																							0	0	1	1	0	0	0	0	0	0
0x04	USART_DR	Reserved																						DR[8:0]									
	Reset value																								0	0	0	0	0	0	0	0	0
0x08	USART_BRR	Reserved																DIV_Mantissa[15:4]												DIV_Fraction [3:0]			
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0C	USART_CR1	Reserved																OVER8	Reserved	UE	M	WAKE	PCE	PS	PEIE	TXEIE	TCIE	RXNEIE	IDLEIE	TE	RE	RWU	SBK
	Reset value																	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10	USART_CR2	Reserved																	LINEN	STOP [1:0]		CLKEN	CPOL	CPHA	LBCL	Reserved	LBDIE	LBDL	Reserved	ADD[3:0]			
	Reset value																		0	0	0	0	0	0	0		0	0		0	0	0	0
0x14	USART_CR3	Reserved																				ONEBIT	CTSIE	CTSE	RTSE	DMAT	DMAR	SCEN	NACK	HDSEL	IRLP	IREN	EIE
	Reset value																					0	0	0	0	0	0	0	0	0	0	0	0
0x18	USART_GTPR	Reserved																GT[7:0]								PSC[7:0]							
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

4.2 USART 주요 레지스터

- **Status register (USART_SR)**

Reset value: 0x00C0 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						CTS	LBD	TXE	TC	RXNE	IDLE	ORE	NF	FE	PE
						rc_w0	rc_w0	r	rc_w0	rc_w0	r	r	r	r	r

Bit 7 TXE: Transmit data register empty
This bit is set by hardware when the content of the TDR register has been transferred into the shift register. An interrupt is generated if the TXEIE bit =1 in the USART_CR1 register. It
is cleared by a write to the USART_DR register.
0: Data is not transferred to the shift register
1: Data is transferred to the shift register)
Note: This bit is used during single buffer transmission.

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

Bit 6 TC: Transmission complete

This bit is set by hardware if the transmission of a frame containing data is complete and if TXE is set. An interrupt is generated if TCIE=1 in the USART_CR1 register. It is cleared by a software sequence (a read from the USART_SR register followed by a write to the USART_DR register). The TC bit can also be cleared by writing a '0' to it. This clearing sequence is recommended only for multibuffer communication.

0: Transmission is not complete

1: Transmission is complete

RXNE: Read data register not empty

This bit is set by hardware when the content of the RDR shift register has been transferred to the USART_DR register. An interrupt is generated if RXNEIE=1 in the USART_CR1 register. It is cleared by a read to the USART_DR register. The RXNE flag can also be cleared by writing a zero to it. This clearing sequence is recommended only for multibuffer communication.

0: Data is not received

1: Received data is ready to be read.

- **Control register 1 (USART_CR1)**

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

Bit 13 UE: USART enable
When this bit is cleared the USART prescalers and outputs are stopped and the end of the current byte transfer in order to reduce power consumption. This is set and cleared by S/W.
0: USART prescaler and outputs disabled
1: USART enabled
Bit 12 M: Word length : This bit determines the word length. It is set or cleared by software.
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
This bit selects the hardware parity control (generation and detection). When the parity control is enabled, the computed parity is inserted at the MSB position (9th bit if M=1; 8th bit if M=0) and parity is checked on the received data. This bit is set and cleared by S/W. Once it is set, PCE is active after the current byte (in reception and in transmission).
0: Parity control disabled, 1: Parity control enabled

Bit 9 PS: Parity selection
This bit selects the odd or even parity when the parity generation/detection is enabled (PCE bit set). It is set and cleared by software. The parity will be selected after the current byte.
0: Even parity, 1: Odd parity

Bit 7 TXEIE: TXE interrupt enable : This bit is set and cleared by software.
0: Interrupt is inhibited
1: An USART interrupt is generated whenever TXE=1 in the USART_SR register

Bit 6 TCIE: Transmission complete interrupt enable
This bit is set and cleared by software.
0: Interrupt is inhibited
1: An USART interrupt is generated whenever TC=1 in the USART_SR register.

Bit 5 RXNEIE: RXNE interrupt enable : This bit is set and cleared by software.
0: Interrupt is inhibited
1: An USART interrupt is generated whenever ORE=1 or RXNE=1 in the USART_SR

Bit 3 TE: Transmitter enable : This bit enables the transmitter. It is set and cleared by S/W.
0: Transmitter is disabled, 1: Transmitter is enabled
Bit 2 RE: Receiver enable : This bit enables the receiver. It is set and cleared by software.
0: Receiver is disabled
1: Receiver is enabled and begins searching for a start bit

- **Control register 1 (USART_CR2)**

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

Bits 13:12 STOP: STOP bits
These bits are used for programming the stop bits.
00: 1 Stop bit
01: 0.5 Stop bit
10: 2 Stop bits
11: 1.5 Stop bit
Note: The 0.5 Stop bit and 1.5 Stop bit are not available for UART4 & UART5.

- **Data register (USART_DR)**

Reset value: 0xXXXX XXXX

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

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

Contains the Received or Transmitted data character, depending on whether it is read from or written to.

- **The Data register performs a double function (read and write) since it is composed of two registers, one for transmission (TDR) and one for reception (RDR).**
- **The TDR register provides the parallel interface between the internal bus and the output shift register.**
- **The RDR register provides the parallel interface between the input shift register and the internal bus.**
- **When transmitting with the parity enabled (PCE bit set to 1 in the USART_CR1 register), the value written in the MSB (bit 7 or bit 8 depending on the data length) has no effect because it is replaced by the parity. When receiving with the parity enabled, the value read in the MSB bit is the received parity bit.**

- **Baud rate register (USART_BRR)**

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

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

Bits 31:16 Reserved, must be kept at reset value

Bits 15:4 DIV_Mantissa[11:0]: mantissa of USARTDIV

These 12 bits define the mantissa of the USART Divider (USARTDIV)

Bits 3:0 DIV_Fraction[3:0]: fraction of USARTDIV

These 4 bits define the fraction of the USART Divider (USARTDIV). When OVER8=1, the DIV_Fraction3 bit is not considered and must be kept cleared.

4.3 NVIC Parameter 및 INT Handler Function

(NVIC Parameter 및 INT Handler Function)
USARTx_IRQn
USARTx_IRQHandler (x=1~6)

5. STM32F407의 USART 프로그래밍 실습

5.1 프로그램에서의 USARTx set-up 과정 및 레지스터 설정

RCC 설정

- RCC→AHB1ENR(GPIOy Clock Enable)
- RCC→APBzENR(USARTx Clock Enable)

INT Enable

- NVIC→ISER[ ]

USARTx 초기 설정

- USARTx→BRR
- USARTx→CR1, CR2, CR3

Int Handler 설정

- USARTx_IRQHandler()

5.2 STM32F407의 USART의 Address(Memory map)

Bus	Boundary address	Peripheral
APB2	0x4001 4C00 - 0x4001 57FF	Reserved
	0x4001 4800 - 0x4001 4BFF	TIM11
	0x4001 4400 - 0x4001 47FF	TIM10
	0x4001 4000 - 0x4001 43FF	TIM9
	0x4001 3C00 - 0x4001 3FFF	EXTI
	0x4001 3800 - 0x4001 3BFF	SYSCFG
	0x4001 3400 - 0x4001 37FF	Reserved
	0x4001 3000 - 0x4001 33FF	SPI1
	0x4001 2C00 - 0x4001 2FFF	SDIO
	0x4001 2400 - 0x4001 2BFF	Reserved
	0x4001 2000 - 0x4001 23FF	ADC1 - ADC2 - ADC3
	0x4001 1800 - 0x4001 1FFF	Reserved
	0x4001 1400 - 0x4001 17FF	USART6
	0x4001 1000 - 0x4001 13FF	USART1
	0x4001 0800 - 0x4001 0FFF	Reserved
	0x4001 0400 - 0x4001 07FF	TIM8
	0x4001 0000 - 0x4001 03FF	TIM1
	0x4000 7800- 0x4000 FFFF	Reserved

Bus	Boundary address	Peripheral
APB1	0x4000 7800 - 0x4000 7FFF	Reserved
	0x4000 7400 - 0x4000 77FF	DAC
	0x4000 7000 - 0x4000 73FF	PWR
	0x4000 6C00 - 0x4000 6FFF	Reserved
	0x4000 6800 - 0x4000 6BFF	CAN2
	0x4000 6400 - 0x4000 67FF	CAN1
	0x4000 6000 - 0x4000 63FF	Reserved
	0x4000 5C00 - 0x4000 5FFF	I2C3
	0x4000 5800 - 0x4000 5BFF	I2C2
	0x4000 5400 - 0x4000 57FF	I2C1
	0x4000 5000 - 0x4000 53FF	UART5
	0x4000 4C00 - 0x4000 4FFF	UART4
	0x4000 4800 - 0x4000 4BFF	USART3
	0x4000 4400 - 0x4000 47FF	USART2
	0x4000 4000 - 0x4000 43FF	I2S3ext
	0x4000 3C00 - 0x4000 3FFF	SPI3 / I2S3
	0x4000 3800 - 0x4000 3BFF	SPI2 / I2S2
	0x4000 3400 - 0x4000 37FF	I2S2ext
	0x4000 3000 - 0x4000 33FF	IWDG
	0x4000 2C00 - 0x4000 2FFF	WWDG
	0x4000 2800 - 0x4000 2BFF	RTC & BKP Registers
	0x4000 2400 - 0x4000 27FF	Reserved
	0x4000 2000 - 0x4000 23FF	TIM14
	0x4000 1C00 - 0x4000 1FFF	TIM13
	0x4000 1800 - 0x4000 1BFF	TIM12
	0x4000 1400 - 0x4000 17FF	TIM7
	0x4000 1000 - 0x4000 13FF	TIM6
	0x4000 0C00 - 0x4000 0FFF	TIM5
	0x4000 0800 - 0x4000 0BFF	TIM4
	0x4000 0400 - 0x4000 07FF	TIM3
	0x4000 0000 - 0x4000 03FF	TIM2

7.3 STM32F407의 ADC관련 header file(stm32f4xx.h)주요 부분

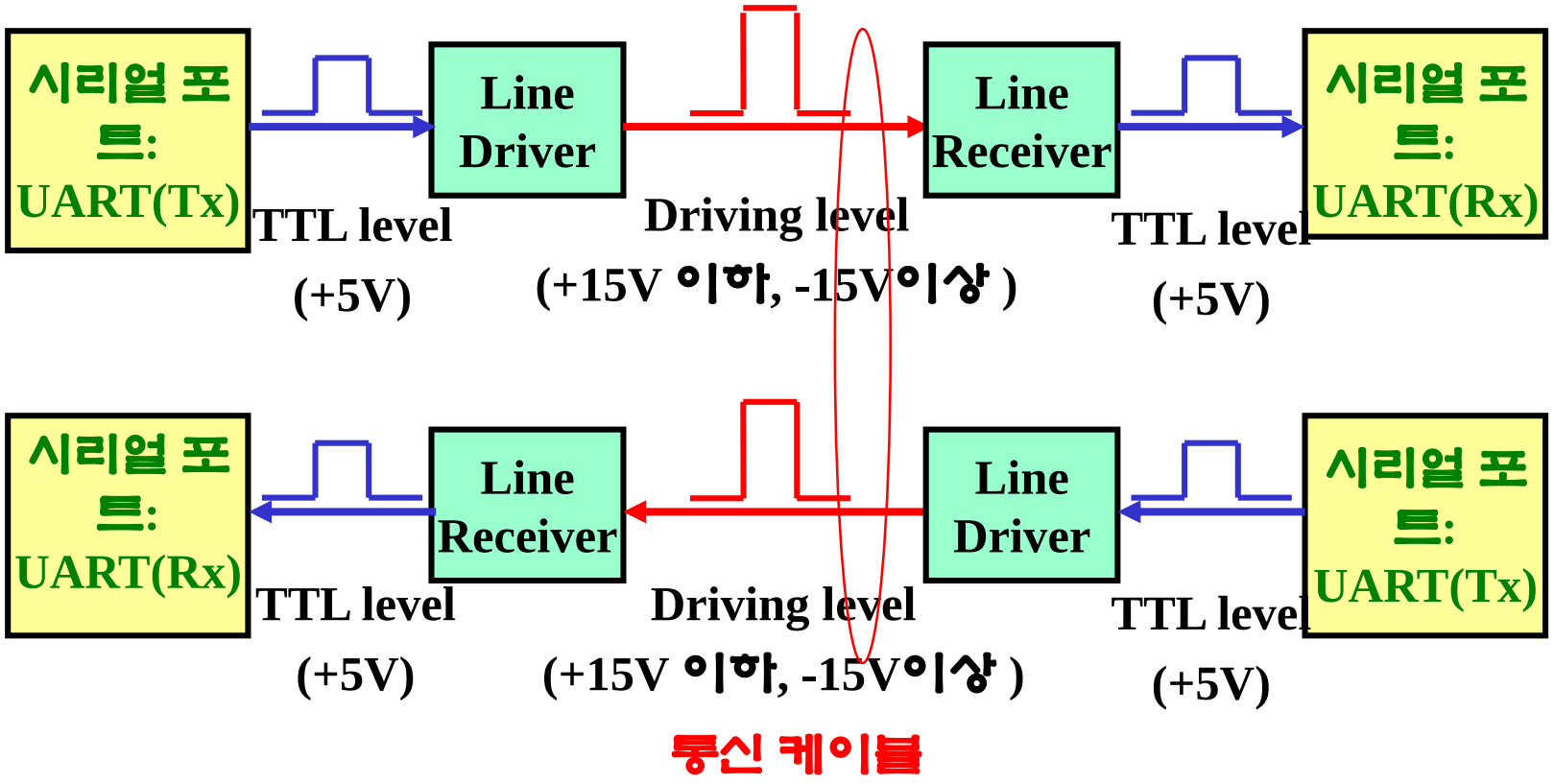
```
/* Peripheral memory map */
#define PERIPH_BASE    ((uint32_t)0x40000000) /* Peripheral base address */
#define APB1PERIPH_BASE      PERIPH_BASE
#define APB2PERIPH_BASE      (PERIPH_BASE + 0x00010000)

 #define USART2_BASE          (APB1PERIPH_BASE + 0x4400)
 #define USART3_BASE          (APB1PERIPH_BASE + 0x4800)
 #define UART4_BASE          (APB1PERIPH_BASE + 0x4C00)
 #define UART5_BASE          (APB1PERIPH_BASE + 0x5000)

 #define USART1_BASE          (APB2PERIPH_BASE + 0x1000)
 #define USART6_BASE          (APB2PERIPH_BASE + 0x1400)

 #define USART2              ((USART_TypeDef *) USART2_BASE)
 #define USART3              ((USART_TypeDef *) USART3_BASE)
 #define UART4              ((USART_TypeDef *) UART4_BASE)
 #define UART5              ((USART_TypeDef *) UART5_BASE)

 #define USART1              ((USART_TypeDef *) USART1_BASE)
 #define USART6              ((USART_TypeDef *) USART6_BASE)
```

```
typedef struct
{
  __IO uint16_t SR;         // USART Status register, offset: 0x00
  __IO uint16_t DR;         // USART Data register offset: 0x04
  __IO uint16_t BRR;        // USART Baud rate register, offset: 0x08
  __IO uint16_t CR1;        // USART Control register 1, offset: 0x0C
  __IO uint16_t CR2;        // USART Control register 2, offset: 0x10
  __IO uint16_t CR3;        // USART Control register 3, offset: 0x14
  __IO uint16_t GTPR;   // USART Guard time and prescaler register, offset: 0x18
} USART_TypeDef;
```

부록 : RS-232C 규격

• RS-232C 직렬통신

• 신호를 먼 거리로 전송하는데 적합한 신호로 변환하기 위한 신호레벨의 인터페이스 회로 필요(TTL 레벨은 원거리전송 부적합)
• 통일된 규격이 필요하므로 국제기구를 통하여 규격화
• 현재까지 일반적으로 널리 사용되고 있는 국제규격의 예:
RS-232C, RS-422A, RS-485 등 (보통 USART통신의 변환기 역할)



• RS-232C의 전기적 사양

- 컴퓨터가 외부와 자료를 주고 받기 위하여 국제적으로 표준화한 데이터 통신규격(모뎀, LAN, RS232 및 X.25 등)의 하나
- 데이터를 직렬 전송 방식으로 전송할 때 통신회선에서 사용하는 전기적인 신호의 특성과 연결장치의 형상 등 물리적인 규격(protocol:주로 H/W)을 규정(참고: RS-422, RS-485)

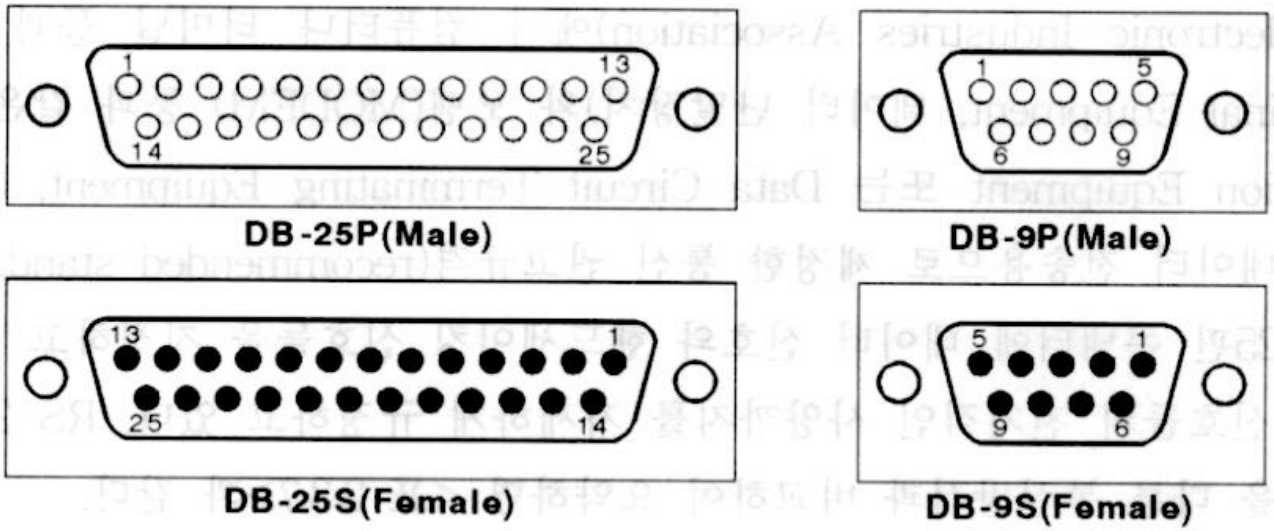
Specification	RS-232C	RS-422	RS-485
동작 모드	Single-Ended	Differential	Differential
최대 Driver/Receiver 수	1 Driver 1 Receiver	1 Driver/ 10 Receivers	32 Drivers 32 Receivers
최대 통달거리	약 15 m	약 1.2 km	약 1.2 km
최고 통신속도	20 Kb/s	10 Mb/s	10 Mb/s
지원 전송방식	Full Duplex	Full Duplex	Half Duplex
최대 출력전압	±25V	-0.25V to +6V	-7V to +12V
최대 입력전압	±15V	-7V to +7V	-7V to +12V

• RS-232C 신호의 기능 및 커넥터 구조

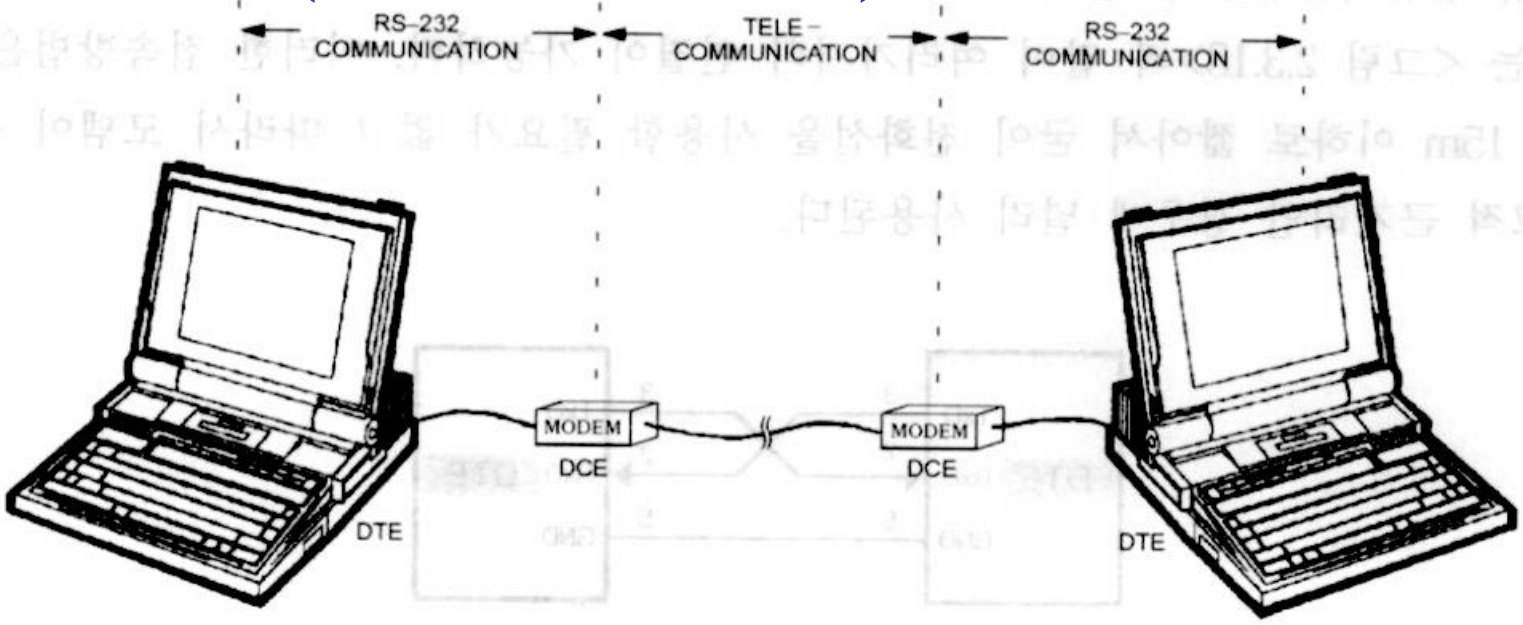
미니DSUB 커넥터	25핀 DSUB	신호 명칭	간단한 사용법
1	8	CD(Data Carrier Detect)	입력, 사용하지 않는다
2	3	RX(Receive Data)	입력, 상대방 TD에 접속
3	2	TX(Transmit Data)	출력, 상대방 RD에 접속
4	20	DTR(Data Terminal Ready)	출력, 사용하지 않는다
5	7	SG(Signal Ground)	그라운드, 상대방 SG에 접속
6	6	DSR(Data Set Ready)	입력, 사용하지 않는다
7	4	RTS(Request to Send)	출력, 상대방 CTS와 접속
8	5	CTS(Clear to Send)	입력, 상대방 RTS와 접속
9	22	RI(Ring Indicate)	입력, 사용하지 않는다

*흐름제어(핸드세이킹 제어)에 필수적인 신호: RTS 및 CTS

- **RS-232C 신호의 기능 및 커넥터 구조**

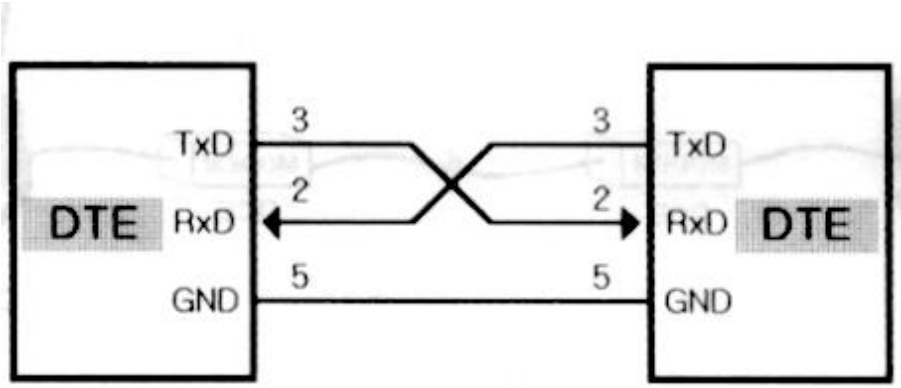


- **시스템 접속도(MODEM 사용한 예)**

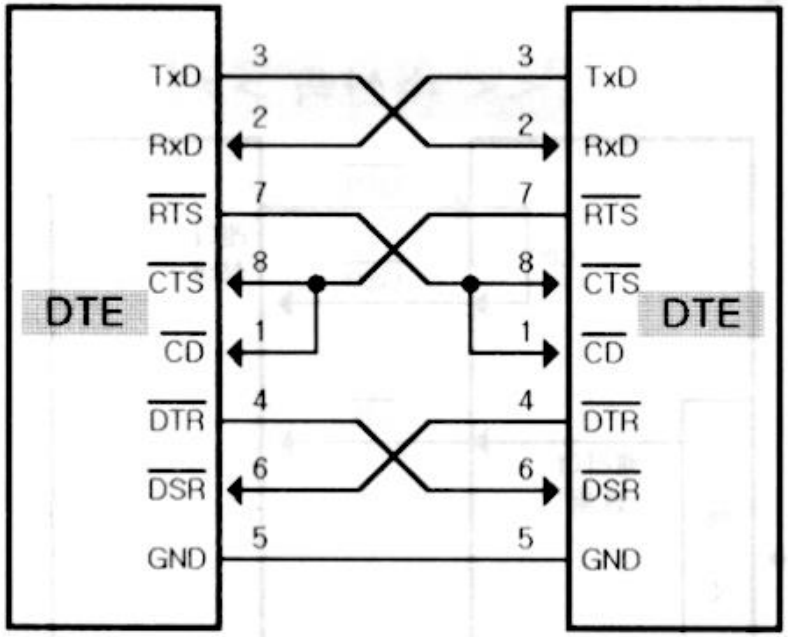


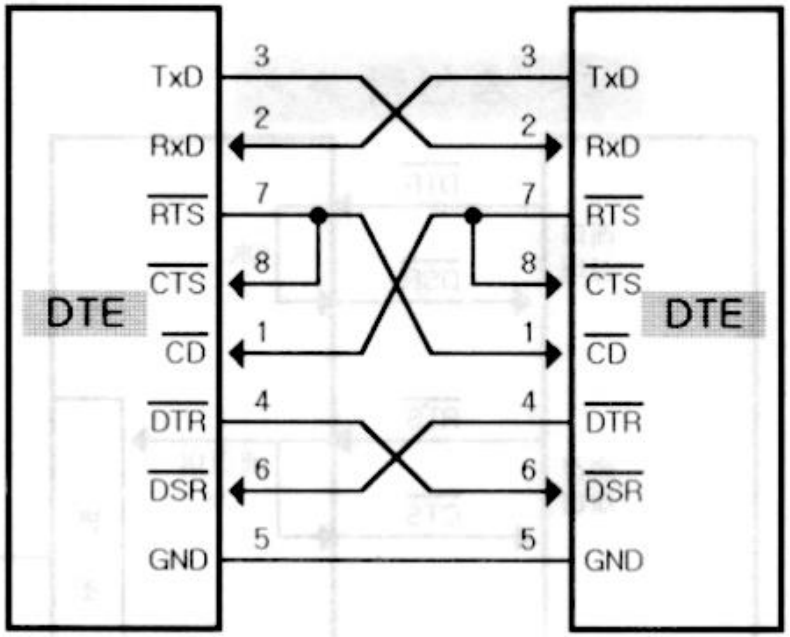
• RS-232C 신호선의 접속: MODEM 미사용(근거리 통신)

(a) No Handshaking 방법

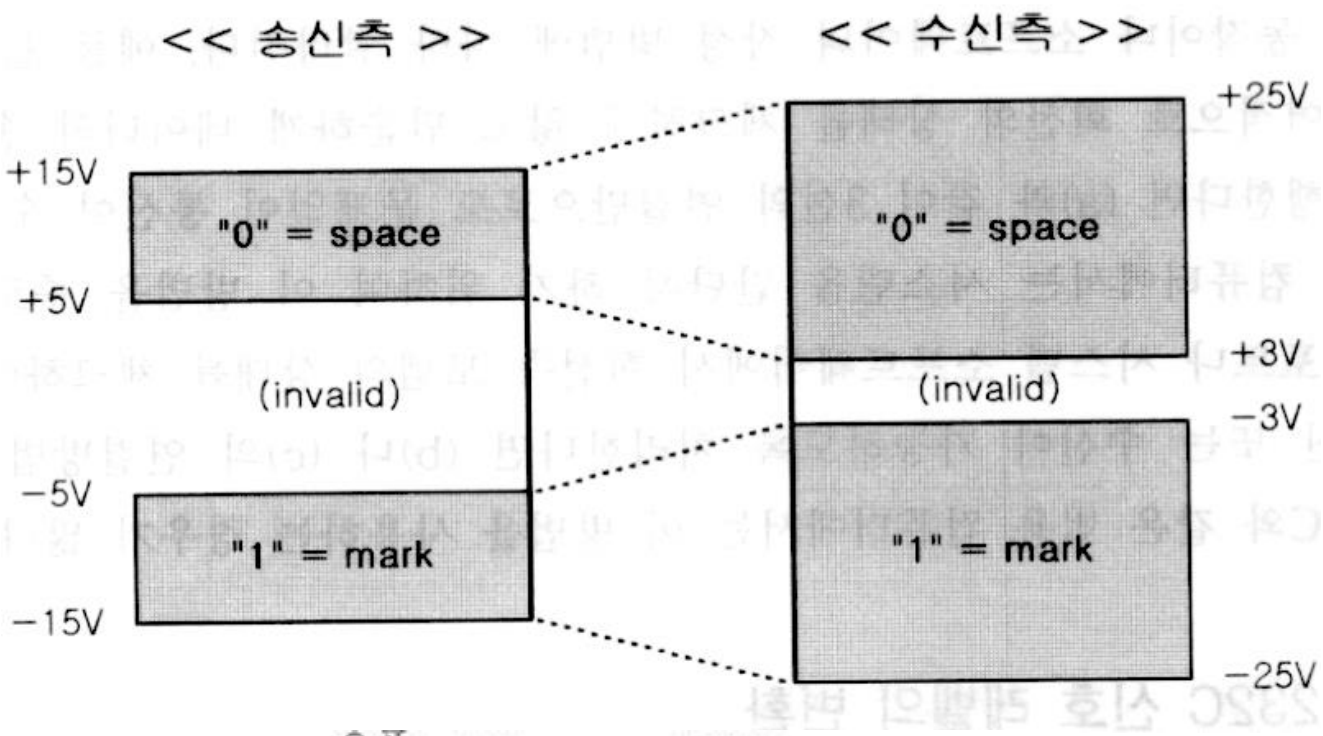


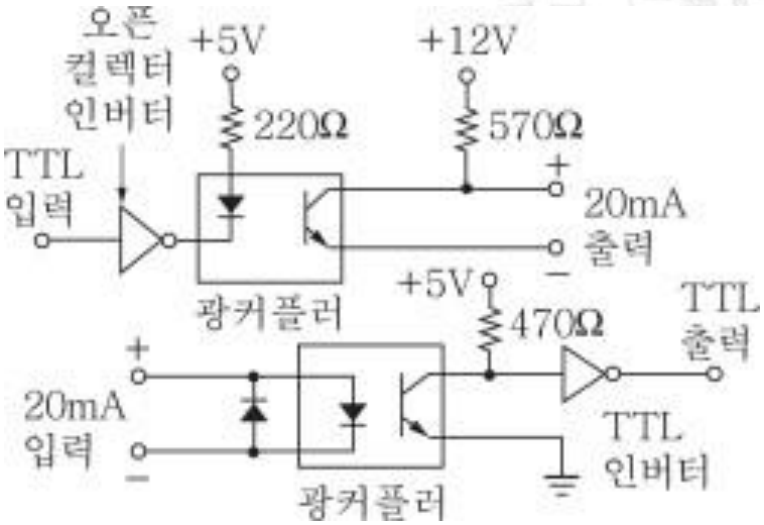
(b) Handshaking 방법





• RS-232C 신호선의 접속: MODEM 미사용(근거리 통신)





• RS-232C 신호선의 접속: MODEM 미사용(근거리 통신)