# 임베디드시스템 기초(#514115 )

# #11. Serial Communications

한림대학교
전자공학과 이선우

한림대학교
HALLYM UNIVERSITY

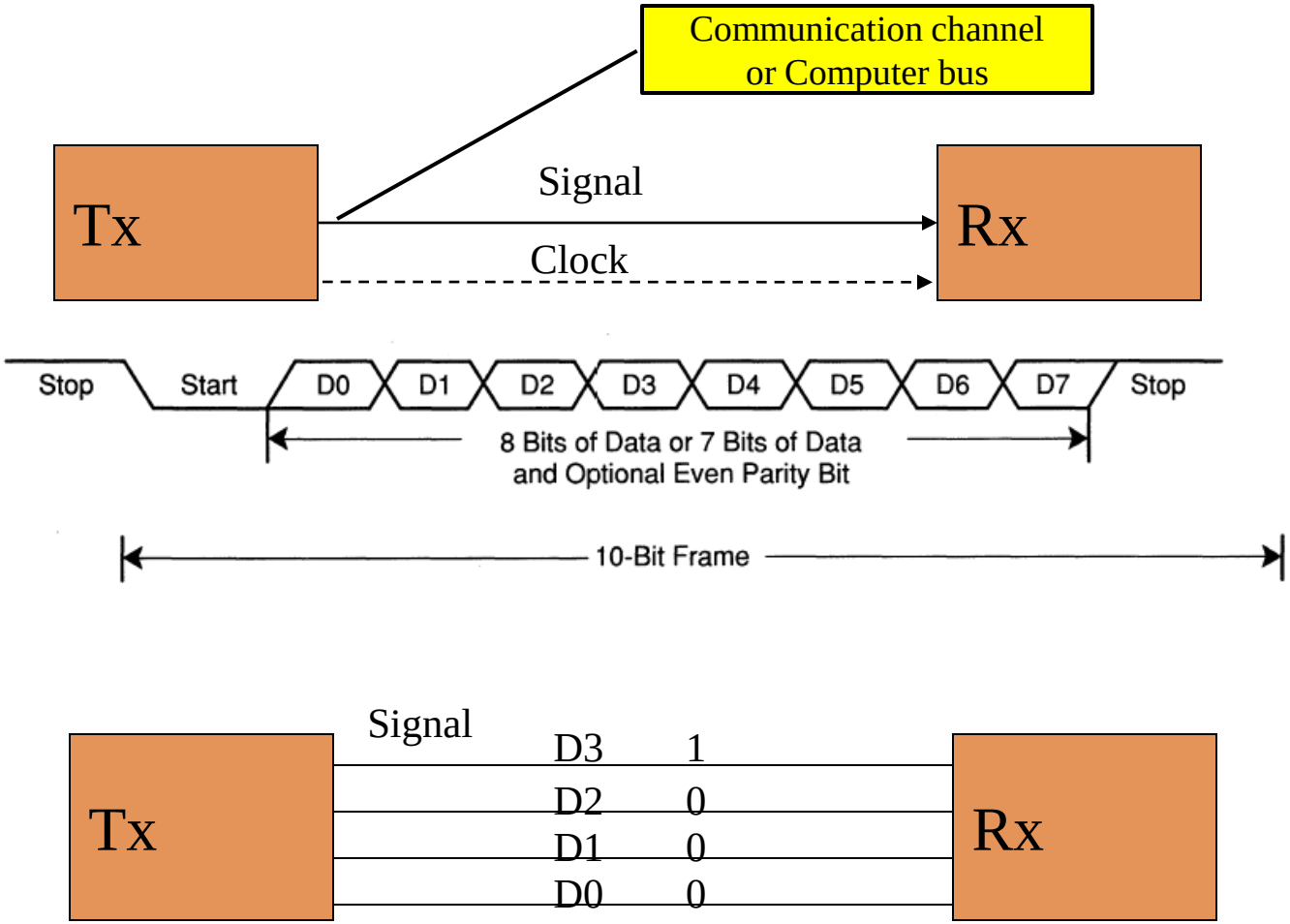
# Contents

- General Serial communications
- Asynchronous serial communications (UART)

# Parallel vs. Serial

- 패러럴 (병렬) 데이터 통신
  - 복수의 신호선을 이용 (대개 8/16/32bit)
  - 올바른 데이터 교환을 위해 일정한 순서-이를 프로토콜, 대개 Handshake방식-로 진행
  - Ex) uP와 외부메모리와의 데이터 통신 방식, PC의 패러랠 포트(8bit)
- 시리얼(직렬) 데이터 통신 (Serial communication)
  - 하나의 신호선을 이용
  - 따라서 일정한 시간간격(이를 baudrate)으로 데이터를 교환
  - Ex) PC 시리얼 포트(COM port), 대개의 MCU의 주 외부와의 통신 방법으로 이용
  - Ref.: http://en.wikipedia.org/wiki/Serial_communications

# Parallel vs. Serial



# Examples of serial communications

- Morse code telegraphy
- **RS-232 (low-speed, implemented by serial ports)** *
- RS-423
- RS-485
- **I²C** *
- **Serial Peripheral Interface Bus (SPI)** *
- Universal Serial Bus (moderate-speed, for connecting peripherals to computers)
- FireWire
- Ethernet
- Fibre Channel (high-speed, for connecting computers to mass storage devices)
- InfiniBand (very high speed, broadly comparable in scope to PCI)
- MIDI control of electronic musical instruments
- DMX512 control of theatrical lighting
- SDI-12 industrial sensor protocol
- Serial Attached SCSI
- Serial ATA

# Asynchronous vs. Synchronous

- Asynchronous (비동기) 통신
  - 송신자와 수신자가 전송 이전에 동기화 필요 없음. 동기를 위한 별도 신호 필요 없음.
  - 대신 송/수신자는 반드시 동일한 클럭 주파수를 알아야 하고, 내장된 동기 신호 (start/stop bit 등) 존재해야 함
  - 항상 전송할 정보를 갖지 않는 장치, 예로 키보드, 마우스 등에 적합
  - 대표적인 표준안: RS-232C, USB 등
- Synchronous (동기) 통신
  - 비동기와 같이 데이터 전송선 이 외에 별도의 동기를 위한 신호선(클럭 signal) 필요
  - 주요 장점: 수신자가 여러 클럭 속도에 대응 가능
  - 표준안: $I^2C$ (필립스), SPI(Motorola) 등

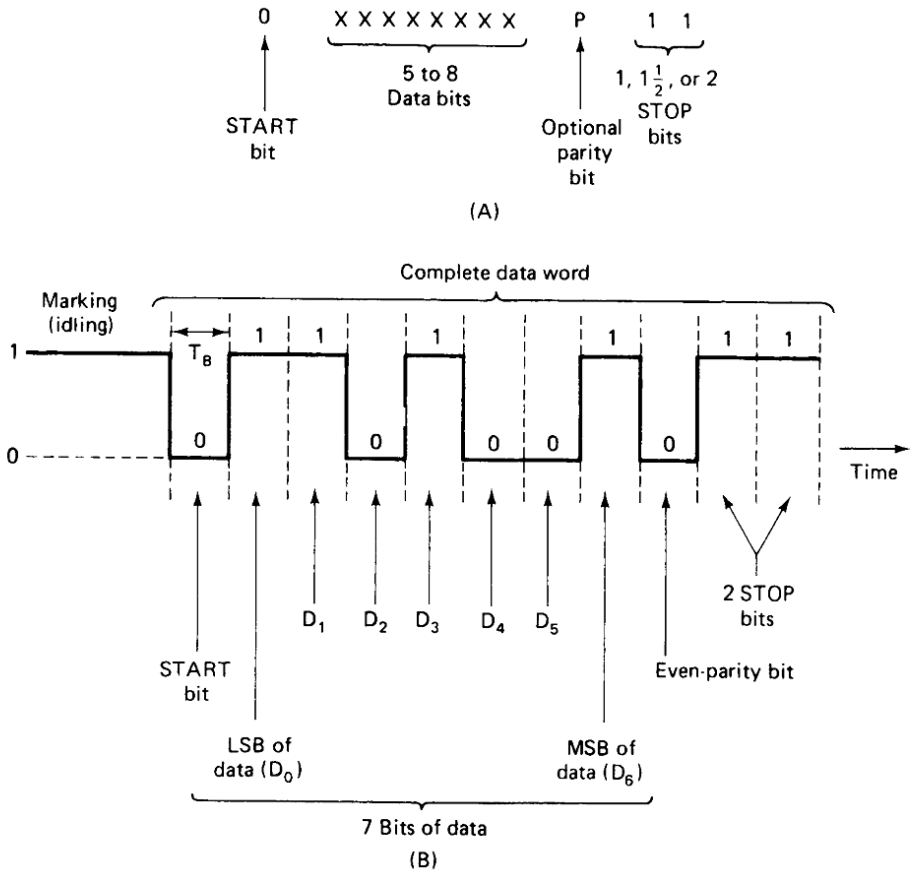
# UART: PC side standard (RS-232C)

- RS-232C 란?
  - EIA (Electronics Industry Association) 표준안 단체에서 시리얼 통신을 위해 정한  전기적 표준안 이름
  - A "Space" (logic 0) will be between +3 and +25 Volts.
  - A "Mark" (Logic 1) will be between -3 and -25 Volts.
  - The region between +3 and -3 volts is undefined.
  - An open circuit voltage should never exceed 25 volts. (In Reference to GND)
  - A short circuit current should not exceed 500mA.
- 표준안에 의해 8m 정도까지는 보장해야 함 (실제 RS-232C를 사용해서 100m 정도까지 가능)
- 18개의 다른 신호선들 정의: GNC, RXD, TXD, DCD, DTR, CTS, RTS, etc
- EIA-232D (1987), EIA-232E(1991) 등으로 발전됨

- * http://en.wikipedia.org/wiki/RS-232

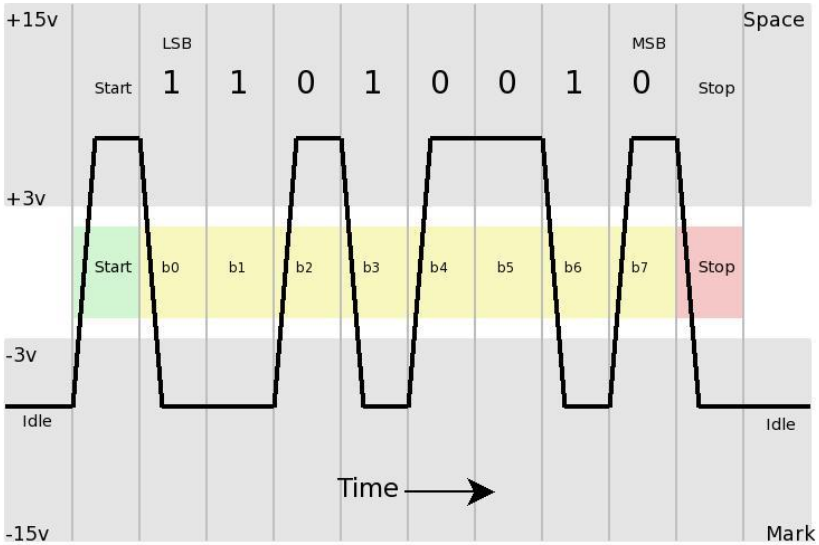
# Waveform

Logical



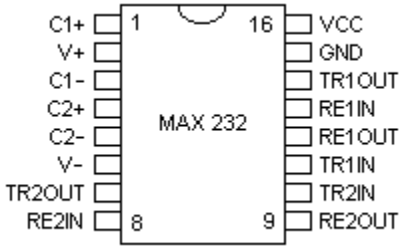

Physical voltage level

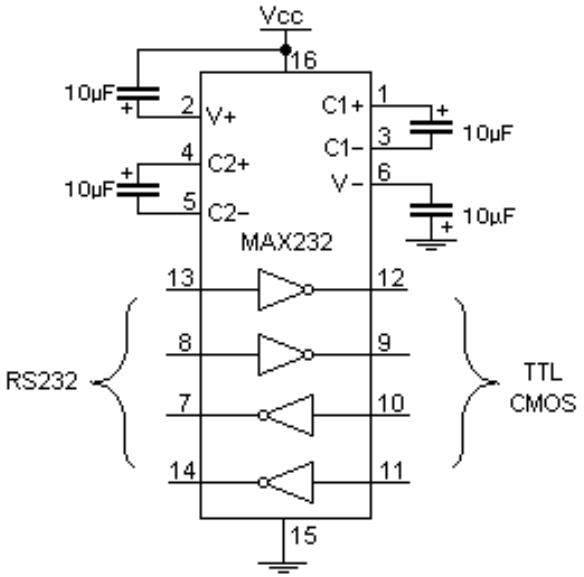
ASCII 'K' (0x4b)



# Level converter

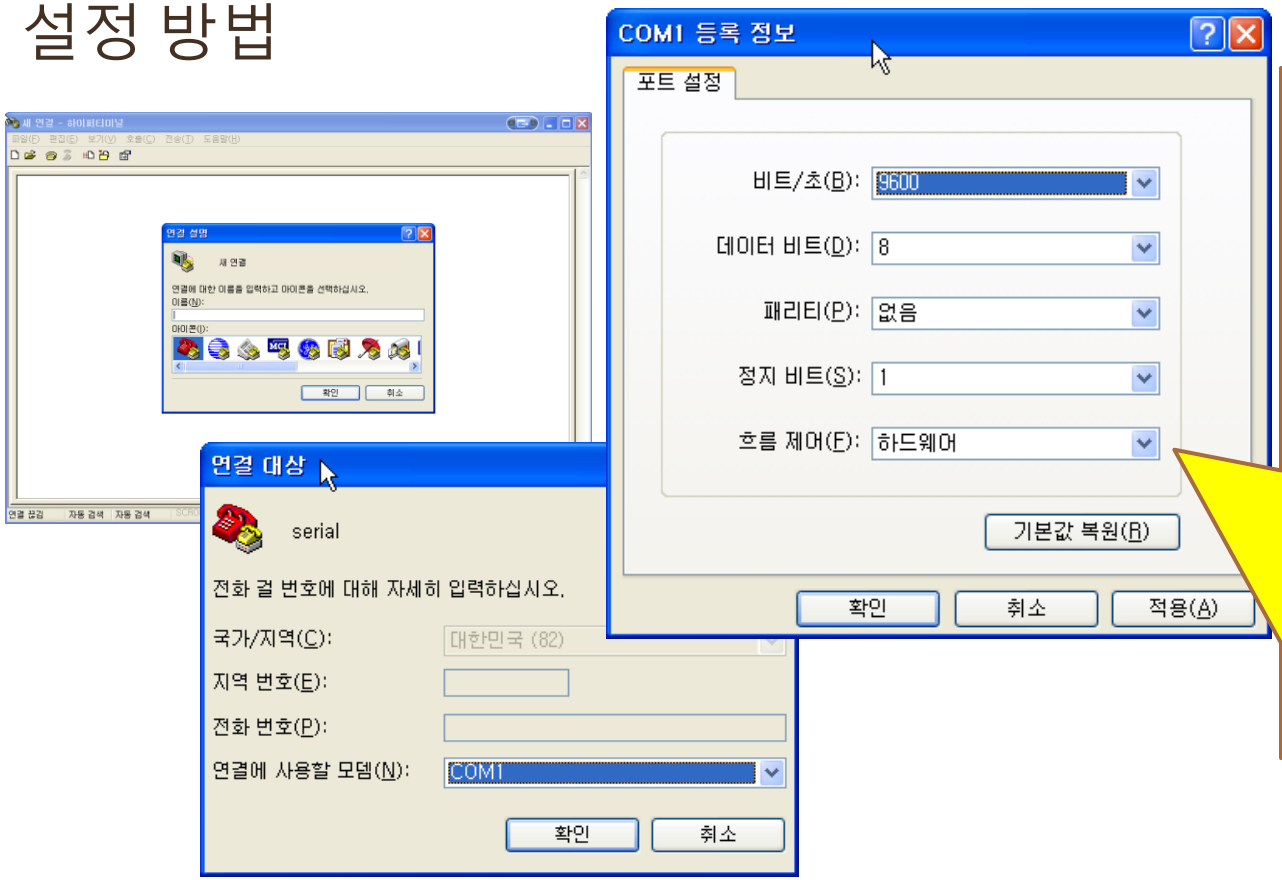
- 대개의 MCU 시스템이나 디지털 시스템은 TTL/CMOS logic level(0,5/3.3V)로 구동되나, RS-232 신호는 ±12V 범위에서 구동되므로 전압의 변환이 필요.
- 이러한 일을 담당해주는 chip들-일종의 dc-dc converter-이 개발, 판매되고 있음.
- 가장 대표적인 칩이 MAXIM의 MAX-232/233임.



# 대표적인 시리얼 통신용 프로그램

- MS Windows OS의 하이퍼터미널
  - 프로그램→보조 프로그램→통신→하이퍼터미널
  - 설정 방법




Parity bit
•전송상의 에러를 찾기 위한 방법
• 2 종류: even(짝수), odd(홀) parity
• 패리티 비트까지 포함하여 1의 개수가 짝/홀수가 되도록 결정함.
• ex. Data 11001010이라면
- Even parity bit=0
- Odd parity bit=1

# Flow control (Handshaking)

- Software flow control:
  - ASCII "XON" (17d,11h) & "XOFF" (19d,13h) code를 이용하여 제어
  - 대개 버퍼의 수가 아주 적은 모뎀과 통신시 이용
    - 예: 모뎀 버퍼가 다 차면 모뎀은 컴퓨터로 XOFF code 전송하여 잠시 중단시키고, 이 후 여유가 생기면 XON code 전송하여 재전송 이루어짐.
    - 장점: 별도의 신호선 필요없으나 속도 저하됨
- Hardware flow control:
  - 별도의 신호선 (RTS/CTS) 이용
  - 컴퓨터가 신호를 보내기 전에 RTS 활성화시키고, 모뎀이 수신 가능하면 CTS를 활성화 시킴.

# MSP430에서의 통신 모듈

- Universal serial communication interface (USCI) module
  - The USCI_Ax modules support:
    - UART mode
    - Pulse shaping for IrDA communications
    - Automatic baud rate detection for LIN communications
    - SPI mode
  - The USCI_Bx modules support:
    - I²C mode
    - SPI mode
- The universal synchronous/asynchronous receive/transmit (USART) module
  - SPI mode
  - UART mode

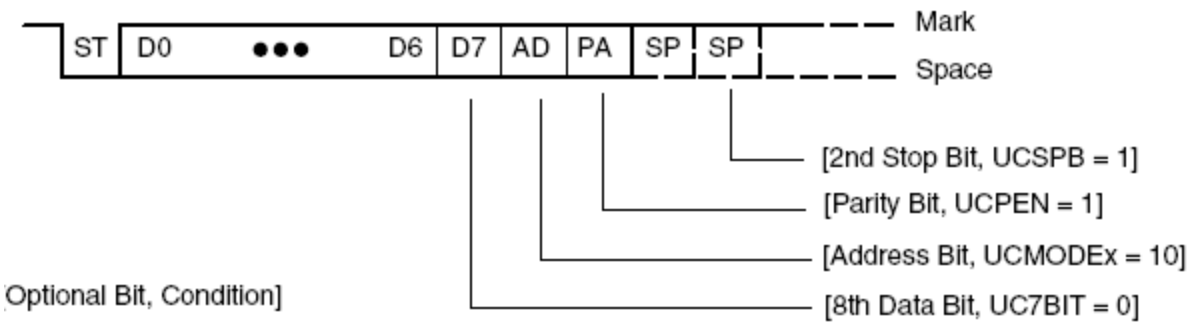
FG461X series has
- USCI_A0 module: SPI (3 or 4 pin), UART, enhanced UART and IrDA.
-USCI_B0 : SPI (3 or 4 pin) and I2C.
- USART1 : UART (double buffered), SPI

# USCI: UART mode (ch.19)

- USCI_Ax modules
  - Use 2 pins: UCAxRXD, UCAxTXD
  - Features
    - 7- or 8-bit data with odd, even, or non-parity
    - Independent transmit and receive shift registers
    - Separate transmit and receive buffer registers
    - LSB-first or MSB-first data transmit and receive
    - Built-in idle-line and address-bit communication protocols for multiprocessor systems
    - Receiver start-edge detection for auto-wake up from LPMx modes
    - Programmable baud rate with modulation for fractional baud rate support
    - Status flags for error detection and suppression
    - Status flags for address detection
    - Independent interrupt capability for receive and transmit

# USCI initialize

- USCI 모듈의 초기화/재설정 순서
  - UCSWRST( SW reset flag in UCAxCTL1)=1
  - 관련 reg.들 설정
  - 포트 설정
  - UCSWRST=0 → 수신/송신 enable (즉 module ON)
  - Enable interrupts (UCAxRXIE, UCAxTXIE)
- Character format



# Automatic Error detection

- Receive error condition

| Error Condition | Error Flag | Description |
|---|---|---|
| Framing error | UCFE | A framing error occurs when a low stop bit is detected. When two stop bits are used, both stop bits are checked for framing error. When a framing error is detected, the UCFE bit is set. |
| Parity error | UCPE | A parity error is a mismatch between the number of 1s in a character and the value of the parity bit. When an address bit is included in the character, it is included in the parity calculation. When a parity error is detected, the UCPE bit is set. |
| Receive overrun | UCOE | An overrun error occurs when a character is loaded into UCAxRXBUF before the prior character has been read. When an overrun occurs, the UCOE bit is set. |
| Break condition | UCBRK | When not using automatic baud rate detection, a break is detected when all data, parity, and stop bits are low. When a break condition is detected, the UCBRK bit is set. A break condition can also set the interrupt flag UCAxRXIFG if the break interrupt enable UCBRKIE bit is set. |

**UCAxSTAT, USCI_Ax Status Register**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| UCLISTEN | UCFE | UCOE | UCPE | UCBRK | UCRXERR | UCADDR UCIDLE | UCBUSY |
| rw–0 | rw–0 | rw–0 | rw–0 | rw–0 | rw–0 | rw–0 | r–0 |

| | | |
|---|---|---|
| UCLISTEN | Bit 7 | Listen enable. The UCLISTEN bit selects loopback mode.<br>0 Disabled<br>1 Enabled. UCAxTXD is internally fed back to the receiver. |
| UCFE | Bit 6 | Framing error flag<br>0 No error<br>1 Character received with low stop bit |
| UCOE | Bit 5 | Overrun error flag. This bit is set when a character is transferred into UCAxRXBUF before the previous character was read. UCOE is cleared automatically when UCxRXBUF is read, and must not be cleared by software. Otherwise, it will not function correctly.<br>0 No error<br>1 Overrun error occurred |
| UCPE | Bit 4 | Parity error flag. When UCPEN = 0, UCPE is read as 0.<br>0 No error<br>1 Character received with parity error |
| UCBRK | Bit 3 | Break detect flag<br>0 No break condition<br>1 Break condition occurred |
| UCRXERR | Bit 2 | Receive error flag. This bit indicates a character was received with error(s). When UCRXERR = 1, on or more error flags (UCFE, UCPE, UCOE) is also set. UCRXERR is cleared when UCAxRXBUF is read.<br>0 No receive errors detected<br>1 Receive error detected |
| UCADDR | Bit 1 | Address received in address-bit multiprocessor mode.<br>0 Received character is data<br>1 Received character is an address |
| UCIDLE | | Idle line detected in idle-line multiprocessor mode.<br>0 No idle line detected<br>1 Idle line detected |
| UCBUSY | Bit 0 | USCI busy. This bit indicates if a transmit or receive operation is in progress.<br>0 USCI inactive<br>1 USCI transmitting or receiving |

# Baud rate generator

- UART 통신을 위해선 정해진 통신 속도(이를 bit rate or baud rate이라 함)를 준수해야 함. 이를 위한 클럭 발생기를 BR generator라 함.
- 크게 2개 모드로 동작
  - Low-frequency mode: UCOS16=0, low-freq. clock source 이용, 저전력 가능. 높은 bps에서 오차 큼.
  - Oversampling mode: UCOS16=1
  - 관련 reg.:
    - Baud rate control reg.: UCAxBR0, UCAxBR1
    - Modulation control reg.: UCAxMCTL
  - Table 19-4, 19-5의 설정값 이용

# USCI Interrupts

- TX/RX 각각 하나씩의 인터럽트가 존재: UCAxTXIFG, UCAxRXIFG
- Transmit
  - 송신 완료 후 새로운 데이터를 받을 준비가 되면 (즉 UCAxTXBUF empty) TXIFG=1이 됨.
  - 새로운 데이터를 TXBUF에 쓰면 자동적으로 reset
- Receive
  - 1 frame data를 수신하고 이를 RXBUF에 넣은 후 발생
  - RXBUF를 읽으면 자동으로 RXIFG reset
- USCI_Ax/USCI_Bx가 인터럽트 벡터 공유
  - SW에서 어떤 모듈이 IRQ 발생했는지 구별해야 함.

| | | | | |
|---|---|---|---|---|
| USCI_A0/USCI_B0 Receive | UCA0RXIFG, UCB0RXIFG (see Note 1) | Maskable | 0FFF2h | 25 |
| USCI_A0/USCI_B0 Transmit | UCA0TXIFG, UCB0TXIFG (see Note 1) | Maskable | 0FFF0h | 24 |
| ADC12 | ADC12IFG (see Notes 1 and 2) | Maskable | 0FFEEh | 23 |

# Example code for USCI-UART operation

```
void main(void)
{
  volatile unsigned int i;

  WDTCTL = WDTPW+WDTHOLD;                    // Stop WDT

  P4SEL |= 0x0C0;                          // P4.7,6 = USCI_A0 RXD/TXD
  UCA0CTL1 |= UCSSEL_2;                      // SMCLK
  UCA0BR0 = 0x09;                            // 1MHz 115200
  UCA0BR1 = 0x00;                            // 1MHz 115200
  UCA0MCTL = 0x02;                           // Modulation
  UCA0CTL1 &= ~UCSWRST;                     // Initialize USCI state
  IE2 |= UCA0RXIE;                         // Enable USCI_A0 RX
interrupt

  __low_power_mode_0();
}

//  Echo back RXed character, confirm TX buffer is ready first
#pragma vector=USCIAB0RX_VECTOR
__interrupt void USCIA0RX_ISR (void)
{
  while(!(IFG2&UCA0TXIFG));
  UCA0TXBUF = UCA0RXBUF; // TX -> RXed character
}
```