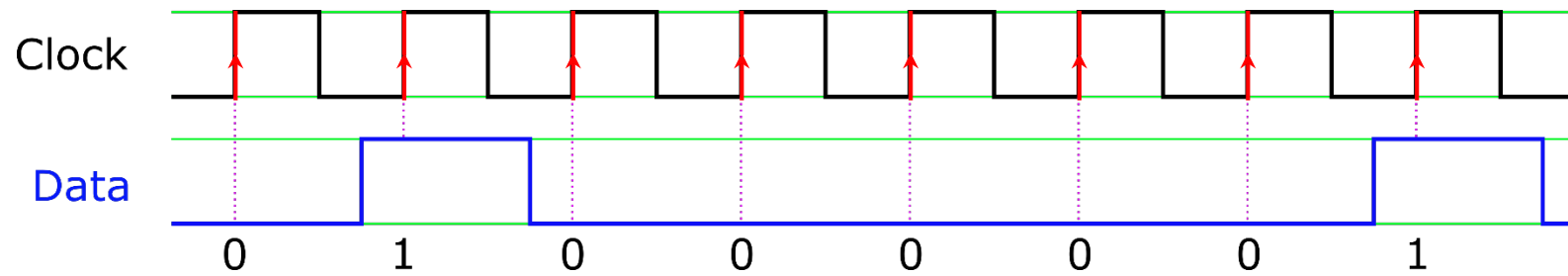
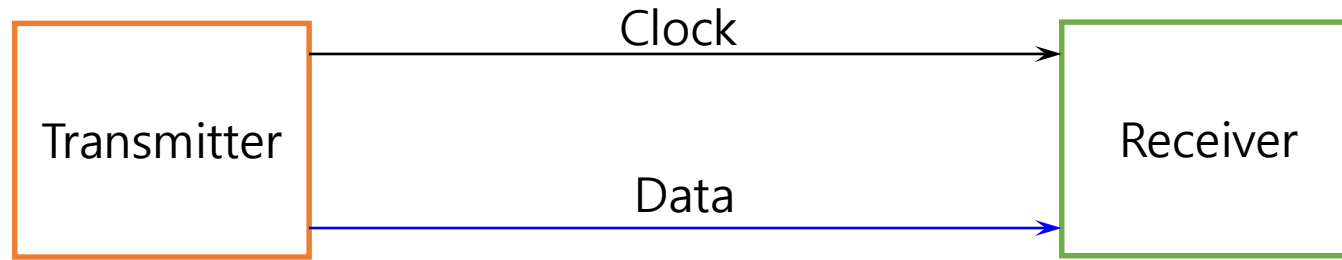


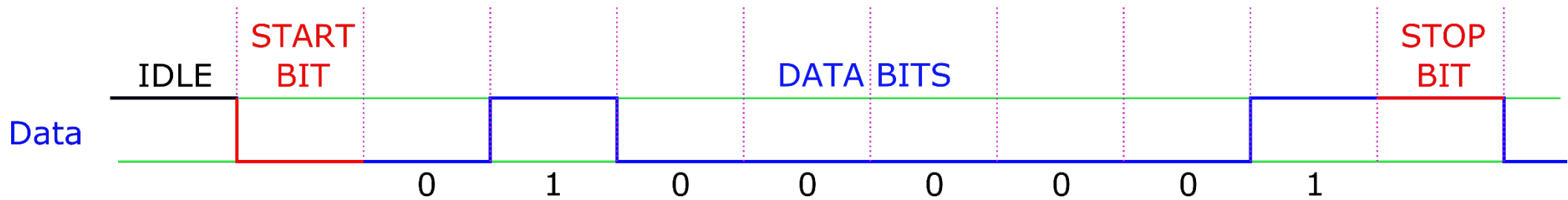
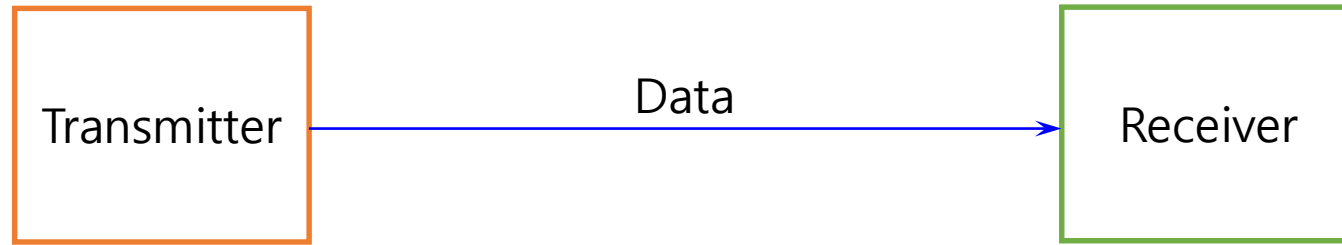
USART

Universal Synchronous/Asynchronous Receiver and Transmitter

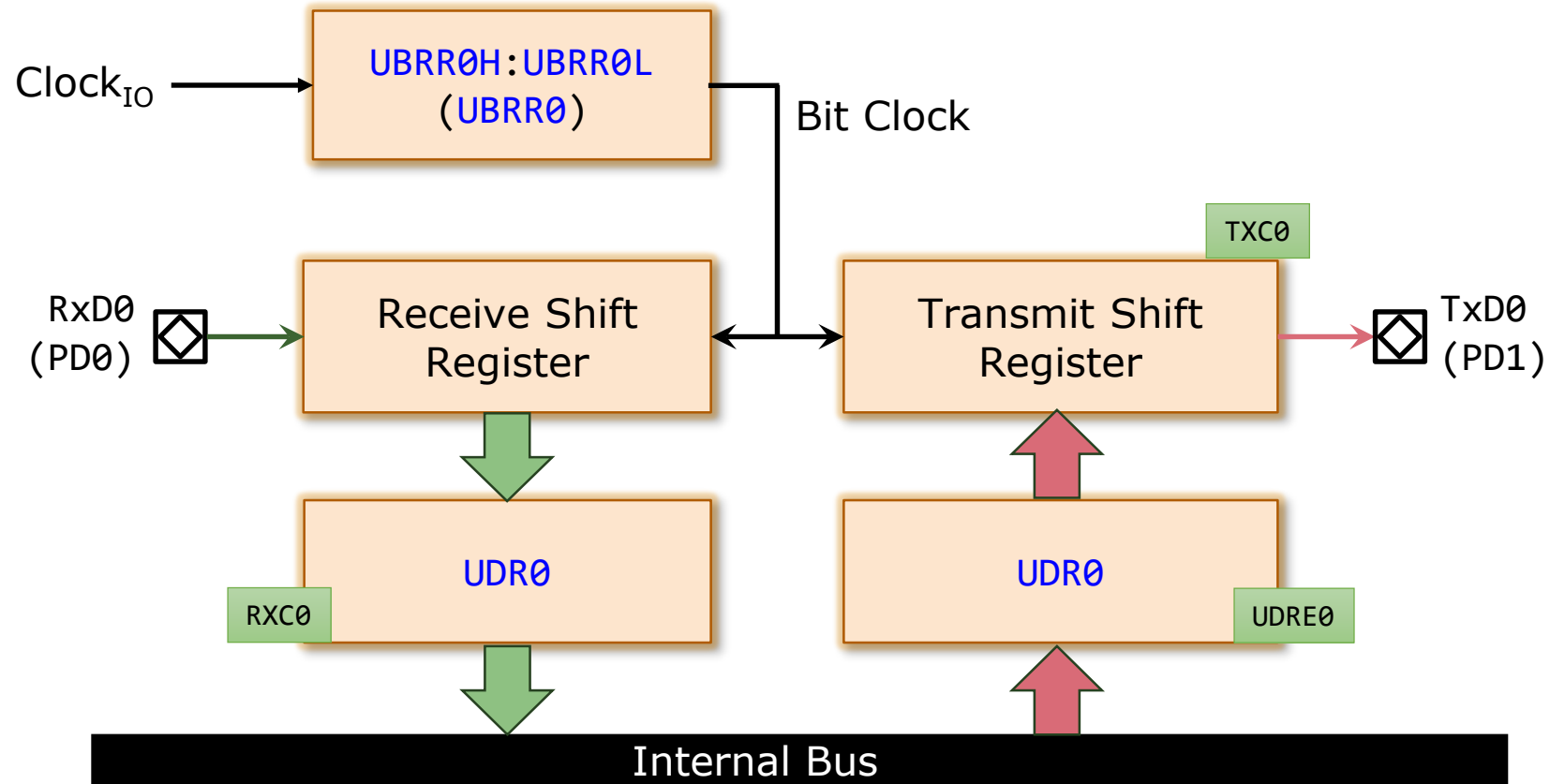
Synchronous Serial Communication



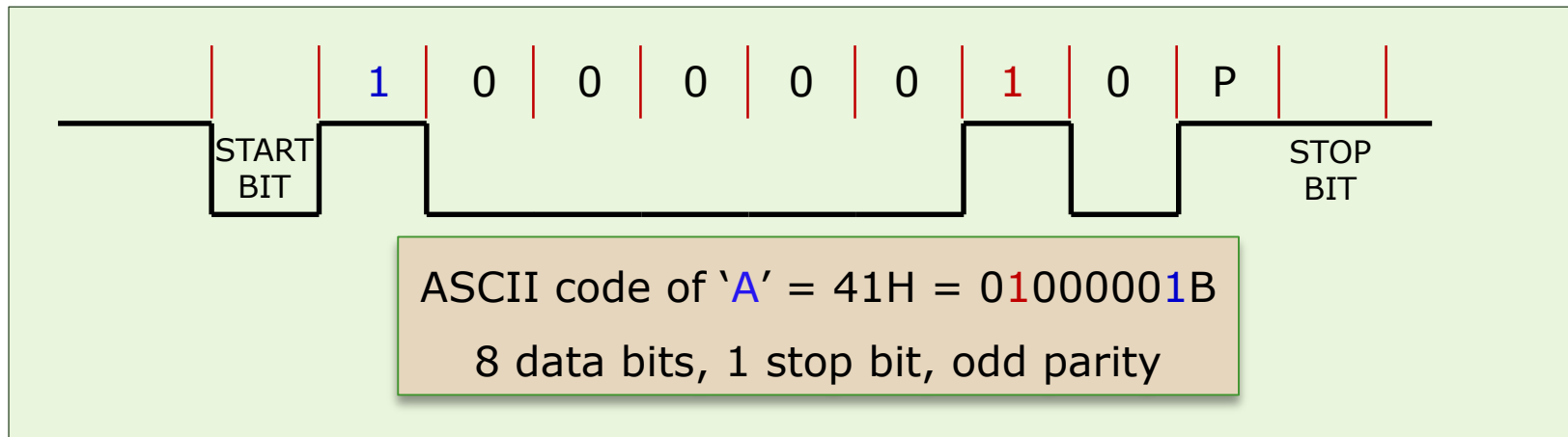
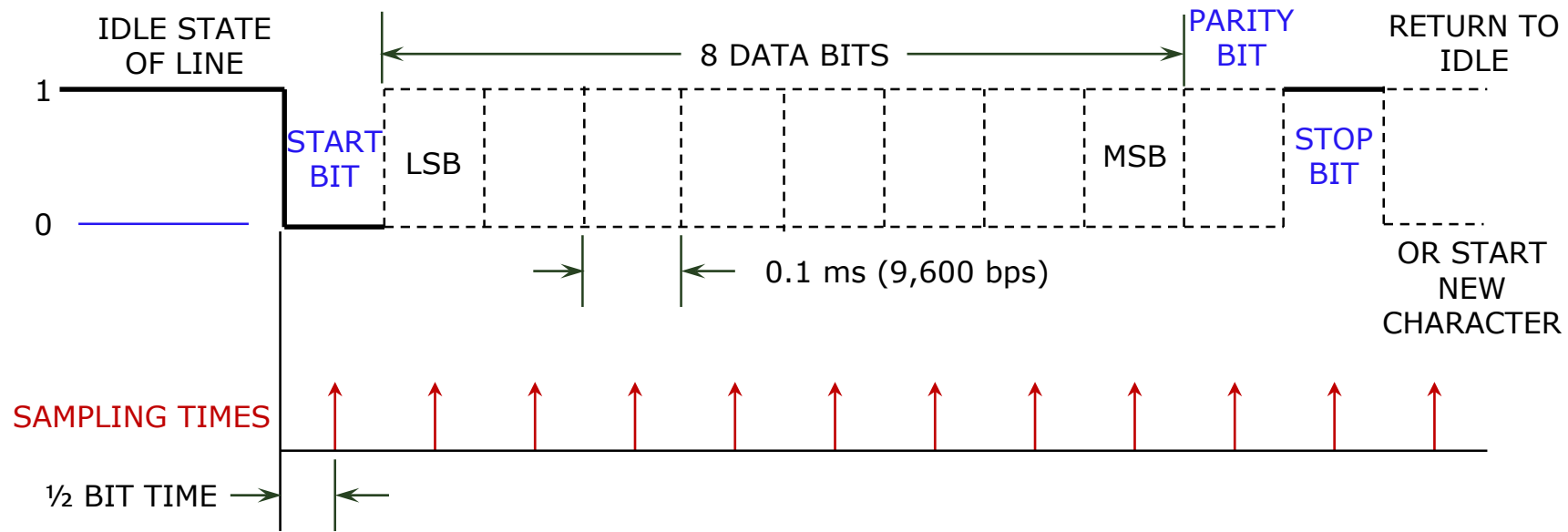
Asynchronous Serial Communication



ATmega328PB UART0 Block Diagram



UART Serial Data Transmission



Baud Rate and Bit Rate

- Bit Rate
 - The number of bits conveyed or processed per unit of time.
 - Unit: bits per second (bit/s or **bps**).
 - can be used interchangeably with "baud" *only* when there are two levels or symbols, representing 0 and 1 respectively.
- Baud Rate
 - The number of distinct symbol changes (signaling events) made to the transmission medium per second.
 - The term baud rate is the same as bit rate when only one bit per symbol is used. (Binary "0" is represented by one symbol, and binary "1" by another symbol)
 - Unit: Bd (/ 'bɔ:d/)

UART Special Receiver Conditions (1)

- Overrun error
 - An "overrun error" occurs when the receiver cannot process the character that just came in before the next one arrives.
 - Various devices have different amounts of buffer space to hold received characters.
 - The CPU must service the UART in order to remove characters from the input buffer.
 - If the CPU does not service the UART quickly enough and the buffer becomes full, an Overrun Error will occur, and incoming characters will be lost.

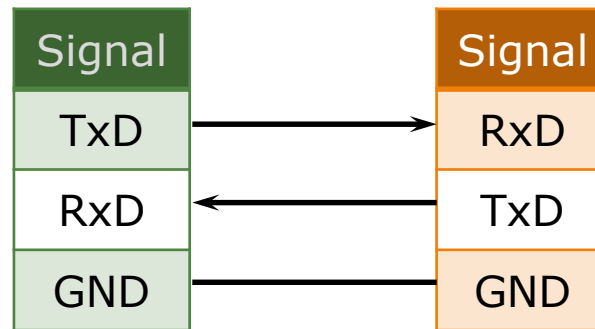
UART Special Receiver Conditions (3)

- Framing error
 - A "framing error" occurs when the designated "start" and "stop" bits are not valid.
 - As the "start" bit is used to identify the beginning of an incoming character, it acts as a reference for the remaining bits.
 - If the data line is not in the expected idle state when the "stop" bit is expected, a Framing Error will occur.

UART Special Receiver Conditions (4)

- Parity error
 - A "parity error" occurs when the number of "active" bits does not agree with the specified parity configuration of the USART, producing a Parity Error.
 - Because the "parity" bit is optional, this error will not occur if parity has been disabled.
 - Parity error is set when the parity of an incoming data character does not match the expected value

UART-UART Wiring



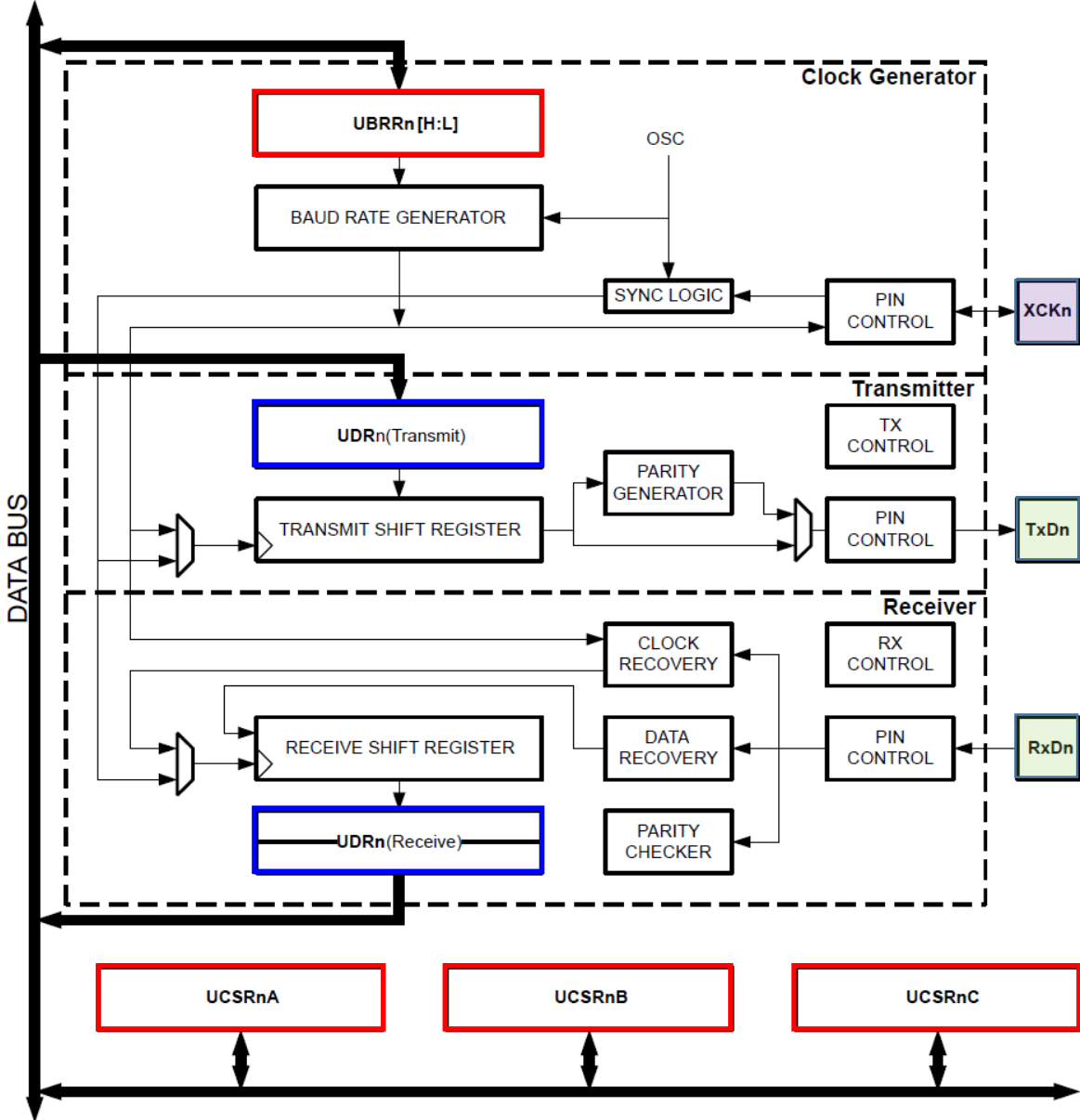
ATmega328PB USART Features (1)

- Two USART instances **USART0**, **USART1**
- **Full Duplex** Operation (Independent Serial Receive and Transmit Registers)
- **Asynchronous** or **Synchronous** Operation
- Master or Slave Clocked Synchronous Operation
- High Resolution Baud Rate Generator
- Supports Serial Frames with **5, 6, 7, 8, or 9 data bits** and **1 or 2 stop bits**
- Odd or Even **Parity** Generation and Parity Check Supported by Hardware
- Data Overrun Detection
- Framing Error Detection

ATmega328PB USART Features (2)

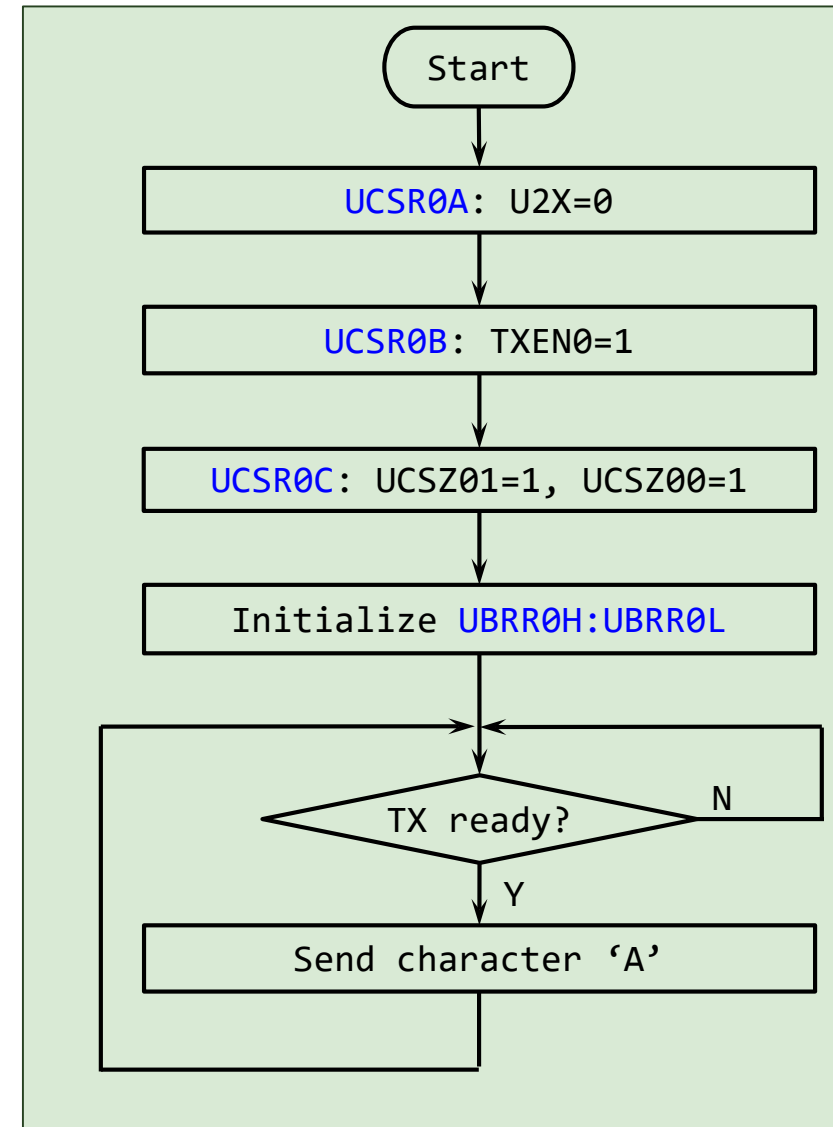
- Noise Filtering Includes False Start Bit Detection and Digital Low Pass Filter
- Three Separate Interrupts on **TX Complete**, **TX Data Register Empty** and **RX Complete**
- Multi-processor Communication Mode
- Double Speed Asynchronous Communication Mode
- Start Frame Detection

ATmega328PB USART Block Diagram



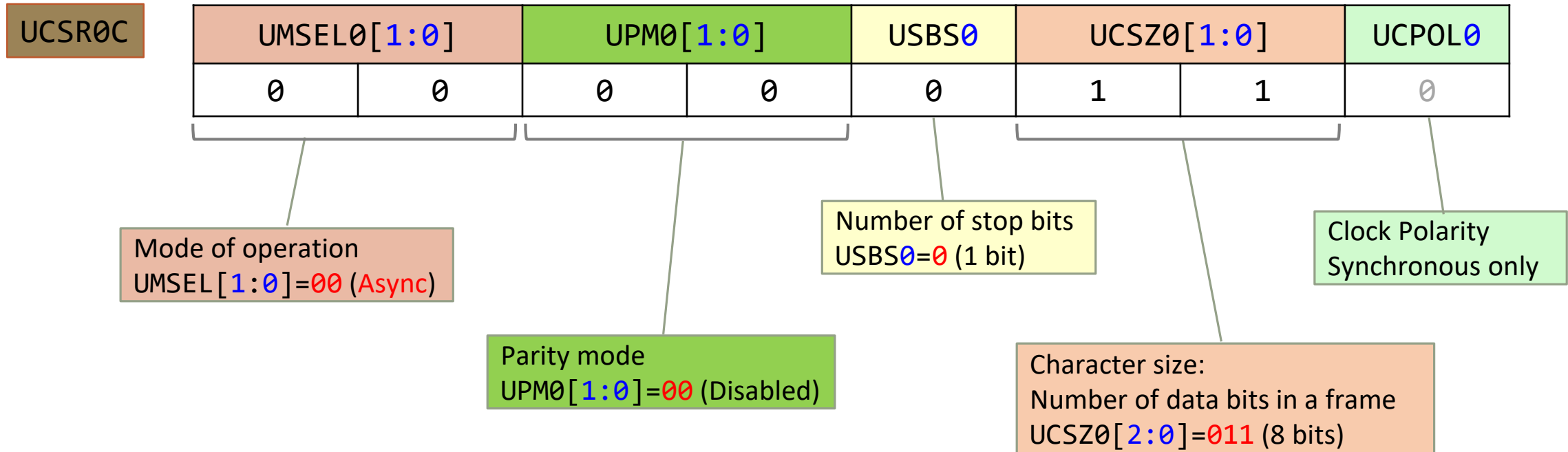
ATmega328PB USART0 Example 1 (Tx, Polling) (1)

- Specifications ($\text{Clock}_{\text{IO}} = 16 \text{ MHz}$):
 - 9,600 baud rate,
 - 8 data bits,
 - no parity,
 - 1 stop bit
 - Do not use double speed
- Transmit character 'A' via USART0 continuously.
- Use Polling method



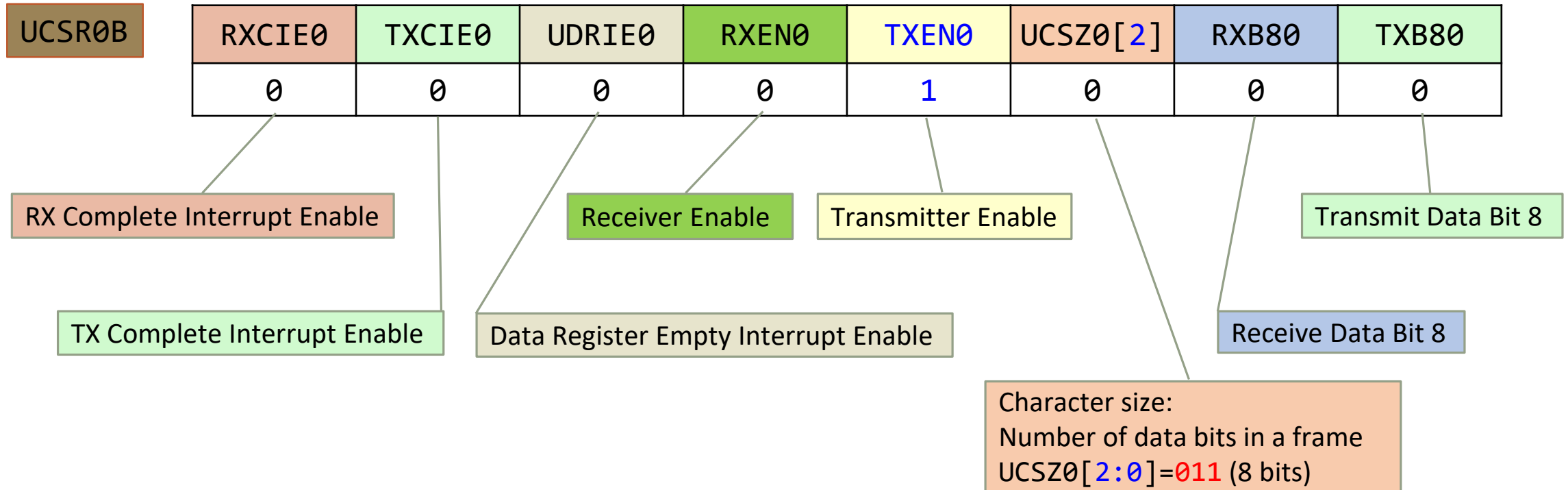
ATmega328PB USART0 Example 1 (Tx, Polling) (2)

Asynchronous, no parity, 1 stop bit, 8 data bits



ATmega328PB USART0 Example 1 (Tx, Polling) (3)

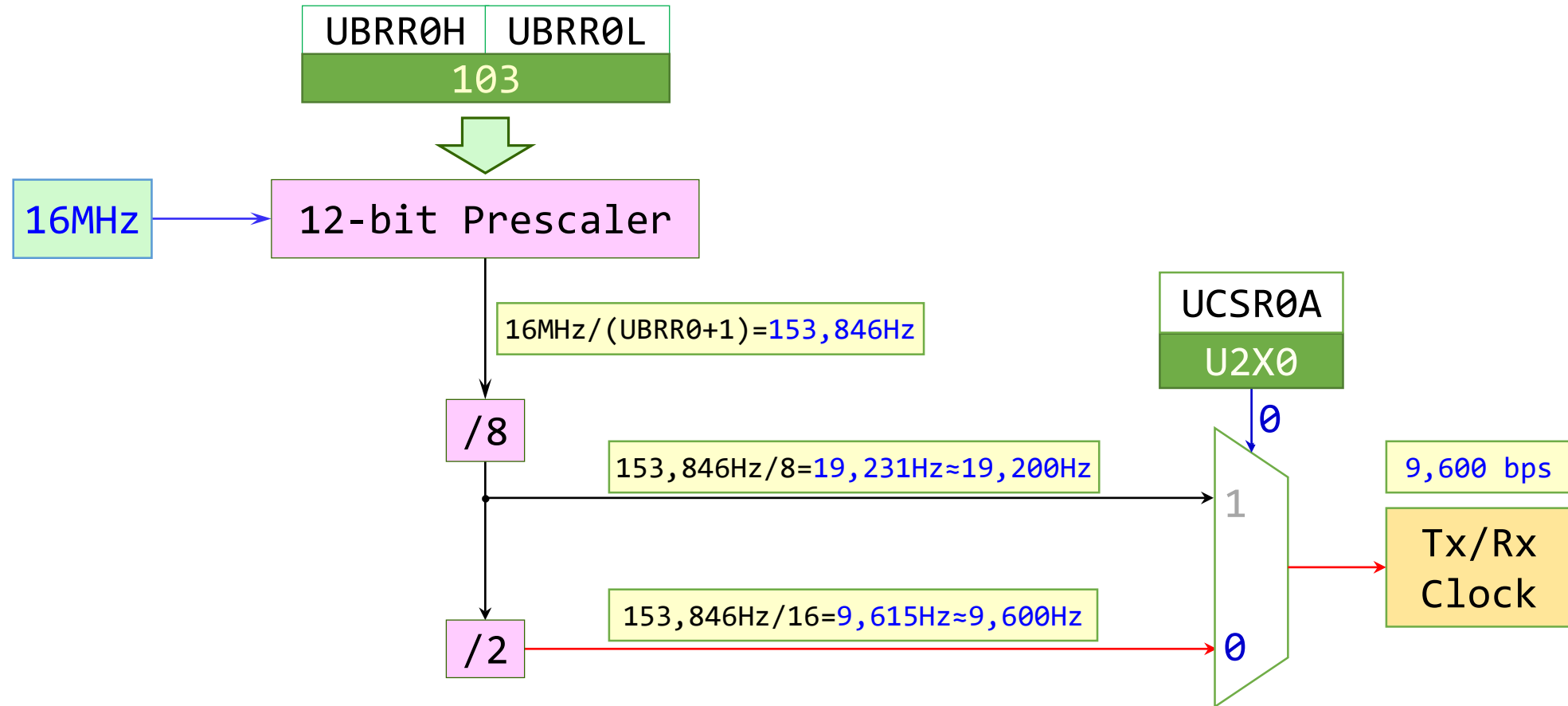
Polling (non-interrupt), enable TX0



ATmega328PB USART0 Tx/Rx Clock Generation

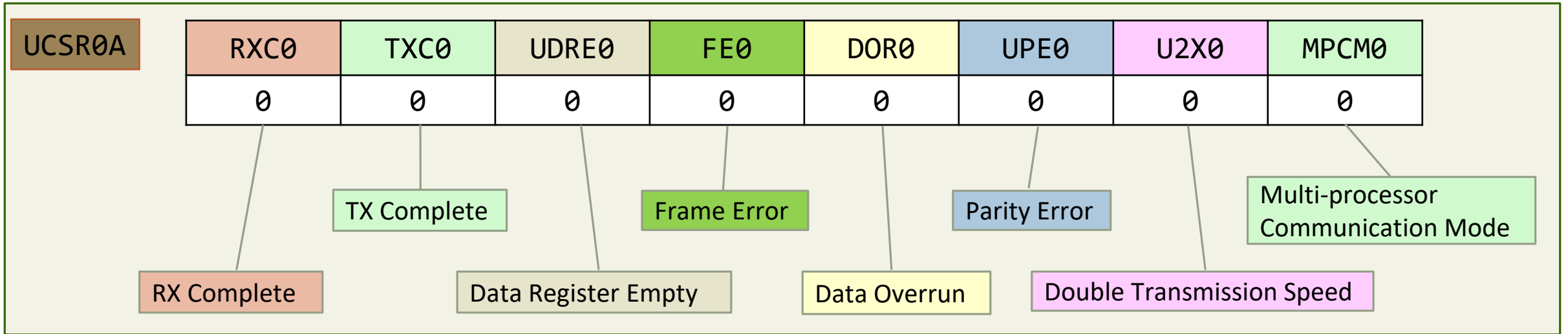
SYSCLK = 16 MHz

Baud Rate = $16,000,000 / (103 + 1) / 8 / 2 = 9,615\text{Hz} \approx 9,600\text{Hz}$



ATmega328PB USART0 Example 1 (Tx, Polling) (4)

Baud Rate = 9,600 bps



UBRR0H	UBRR0L
0	103

$$UBRR0 = \frac{f_{osc}}{16 \times BAUD} - 1 \quad (\text{when } U2X0=0)$$

ATmega328PB USART0 Example 1 (Tx, Polling) (5)

- Specifications($\text{Clock}_{\text{IO}}=16 \text{ MHz}$):
 - baud rate: 9,600 bps,
 - 8 data bits,
 - no parity,
 - 1 stop bit
 - Do not use double speed
- Transmit character 'A' via USART0 continuously.
- Use Polling method

```
#include <avr/io.h>

int main(void)
{
    UCSRA = 0b00000000; // U2X0=0: No double speed
    UCSRB = 0b00001000; // Enable Tx, 8 Data bits
    UCSR0C = 0b00000110; // Async mode, No Parity,
                        // 1 Stop bit, 8 Data bits
    UBRR0 = 103;        // Baud Rate=16MHz/(9600*16)-1

    while (1)
    {
        // Wait until Tx Data Register Empty (UDRE0 bit)
        while ((UCSRA & 0b00100000) == 0);

        UDR0 = 'A';    // Transmit character 'A'
    }
}
```

ATmega328PB USART0 Example 1 (Tx, Polling) (6)

- Specifications(ClockIO=16 MHz):
 - baud rate: 9,600 bps,
 - 8 data bits,
 - no parity,
 - 1 stop bit
 - Do not use double speed
- Transmit character 'A' via USART0 continuously.
- Use Polling method

Same program with different format

```
#define F_CPU 16000000UL
#define UART_BAUD_RATE 9600UL
#define DIVISOR ((F_CPU / (UART_BAUD_RATE * 16UL)) - 1)

#include <avr/io.h>

int main(void)
{
    UCSRA &= ~(1 << U2X0); // U2X0=0: No double speed
    UCSRB = 1 << TXEN0; // Enable Tx, 8 Data bits
    UCSRC = (0b00 << UMSEL00) // Async mode
            | (0b00 << UPM00) // No Parity
            | (0 << USBS0) // 1 Stop bit
            | (0b11 << UCSZ00); // 8 Data bits
    UBRR0 = DIVISOR; // Baud Rate

    while (1)
    {
        // Wait until Tx Data Register Empty
        while ((UCSRA & (1 << UDRE0)) == 0);

        UDR0 = 'A'; // Transmit character 'A'
    }
}
```

ATmega328PB USART0 Example 1 (Tx, Polling) (7)

Same program with functions

```
#define F_CPU 16000000UL
#define UART_BAUD_RATE 9600UL
#define DIVISOR ((F_CPU/(UART_BAUD_RATE*16UL))-1)

#include <avr/io.h>

void uart0_init(void);
void uart0_putchar(char ch);

int main(void)
{
    uart0_init();

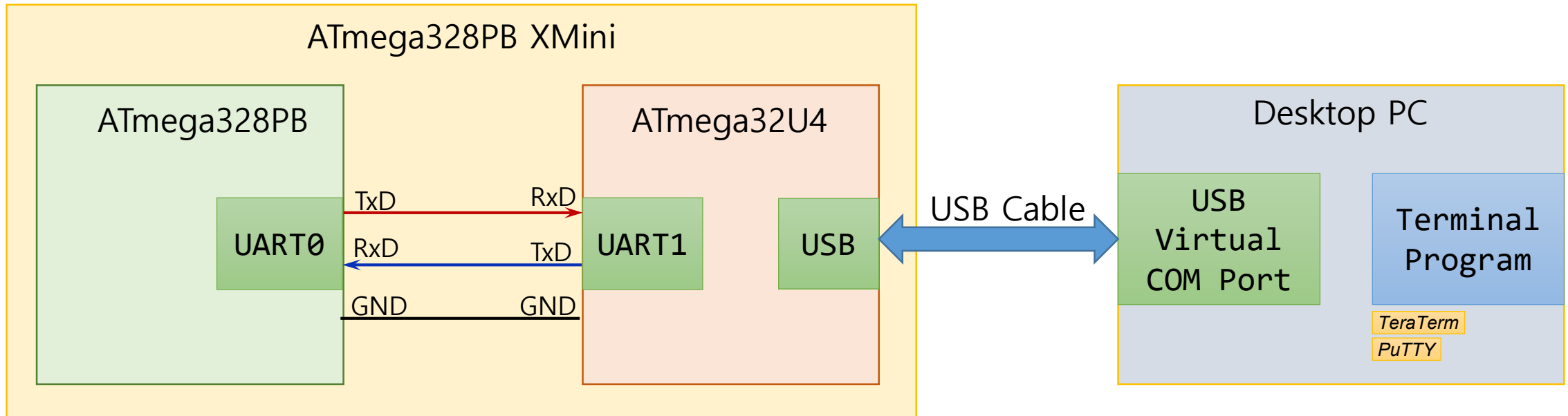
    while (1)
    {
        uart0_putchar('A');
    }
}
```

```
void uart0_init(void)
{
    UCSRA &= ~(1 << U2X0); // U2X0=0: No double speed
    UCSRB = 1 << TXEN0;    // Enable Tx, 8 Data bits
    UCSRC = (0b00 << UMSEL00) // Async mode
           | (0b00 << UPM00)   // No Parity
           | (0 << USBS0)      // 1 Stop bit
           | (0b11 << UCSZ00); // 8 Data bits
    UBRR0 = DIVISOR;         // Baud Rate
}

void uart0_putchar(char ch)
{
    // Wait until Tx Data Register Empty
    while ((UCSRA & (1 << UDRE0)) == 0);

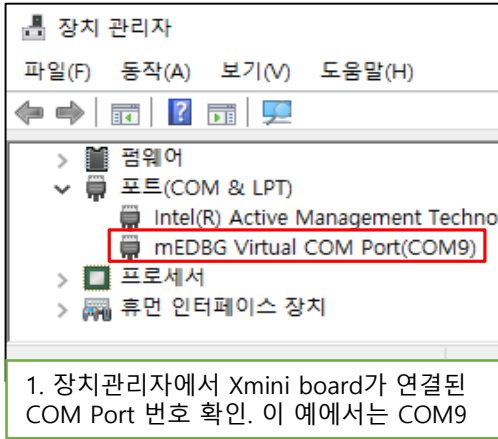
    UDR0 = ch;
}
```

Connect ATmega328PB Xmini to PC Terminal

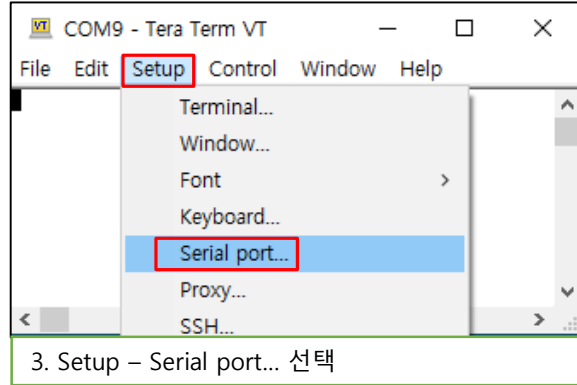


ATmega328PB USART0 Example 1 (Tx, Polling) (8) (Tera Term 4.106)

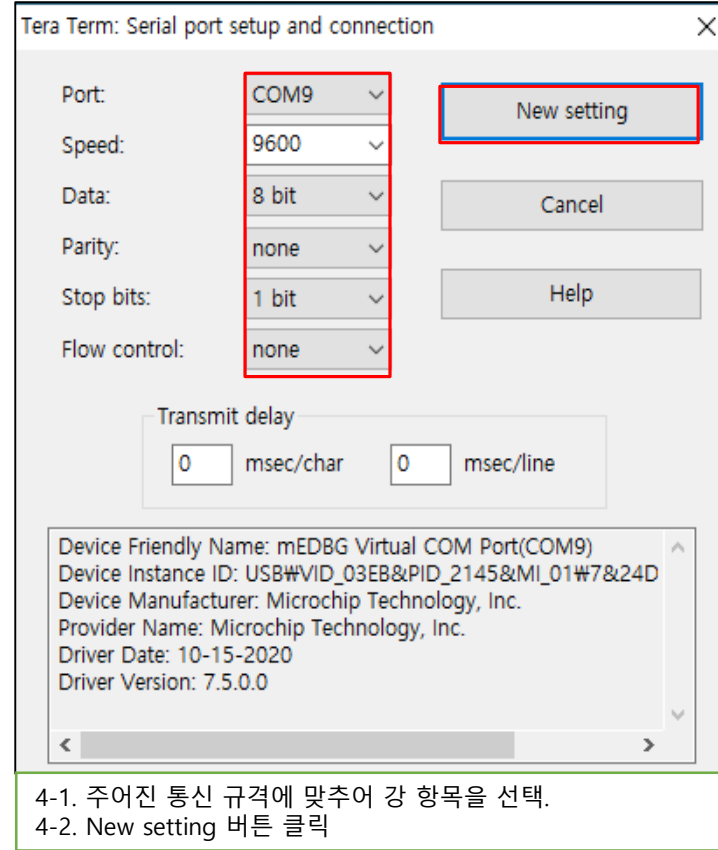
①



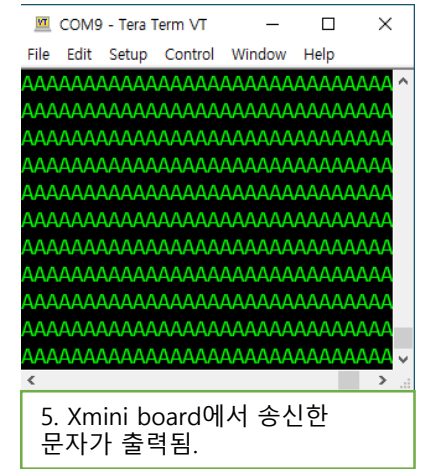
③



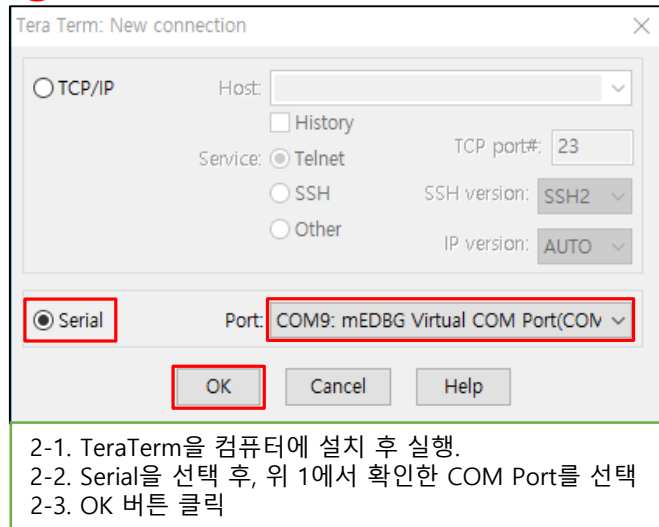
④



⑤

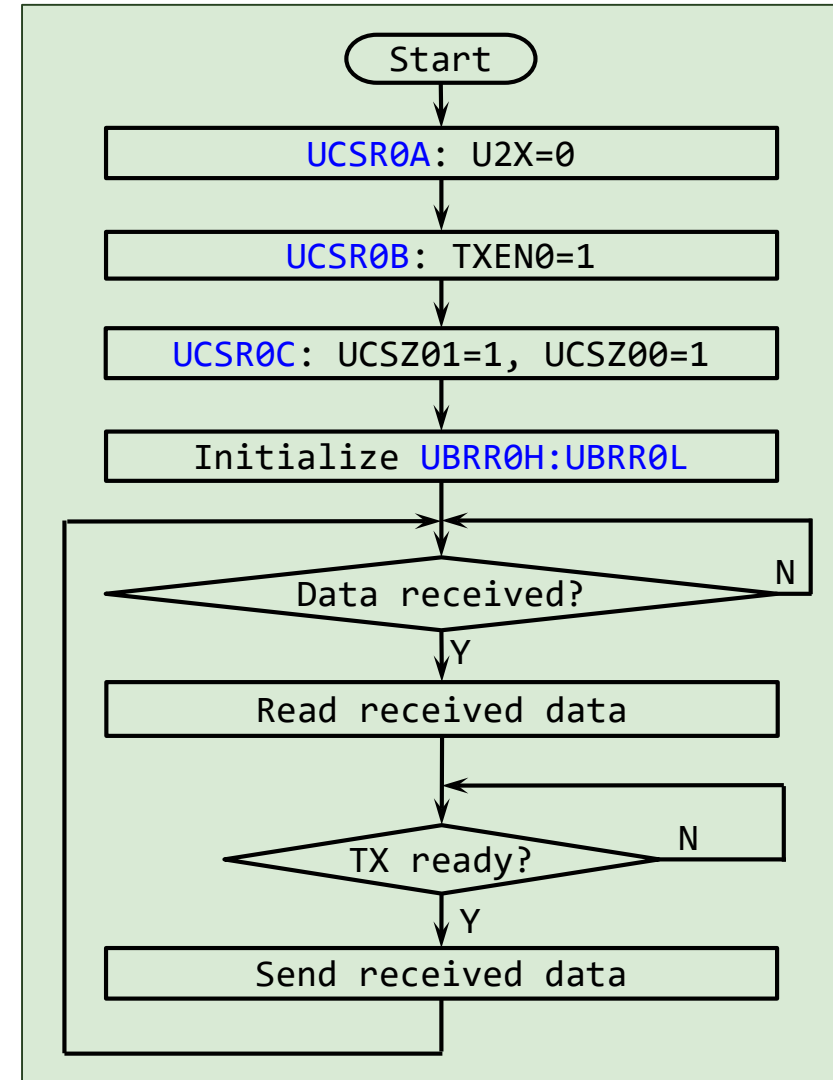


②



ATmega328PB USART0 Example 2 (Rx/Tx, Polling) (1)

- Specifications (Clock_{IO} =16 MHz):
 - 9,600 baud rate,
 - 8 data bits,
 - no parity,
 - 1 stop bit
 - Do not use double speed
- Receive a character and transmit the received character via USART0.
- Use Polling method



ATmega328PB USART0 Example 2 (Rx/Tx, Polling) (2)

- Specifications (Clock_{IO} =16 MHz):
 - 9,600 baud rate,
 - 8 data bits,
 - no parity,
 - 1 stop bit
 - Do not use double speed
- Receive a character and transmit the received character via USART0.
- Use Polling method

```
#include <avr/io.h>

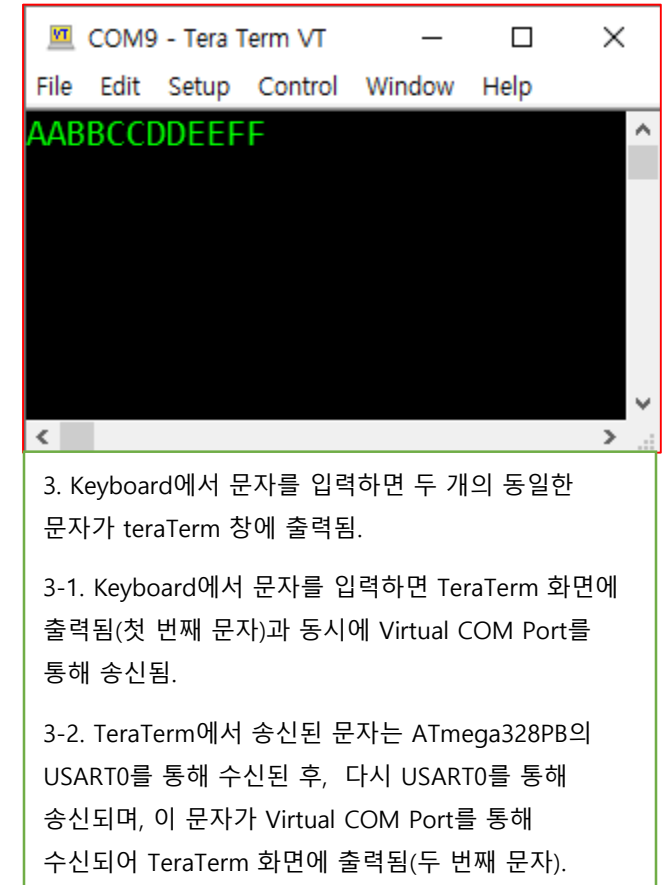
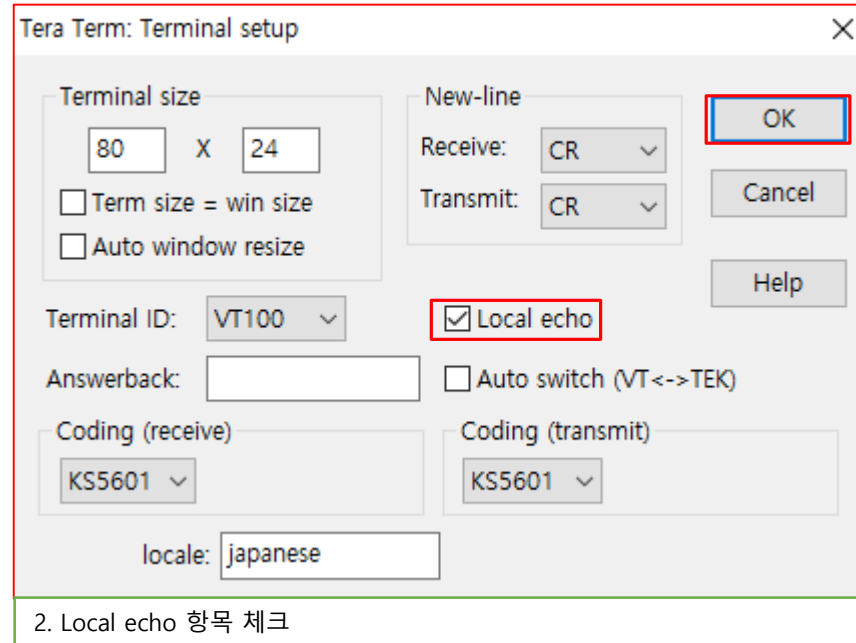
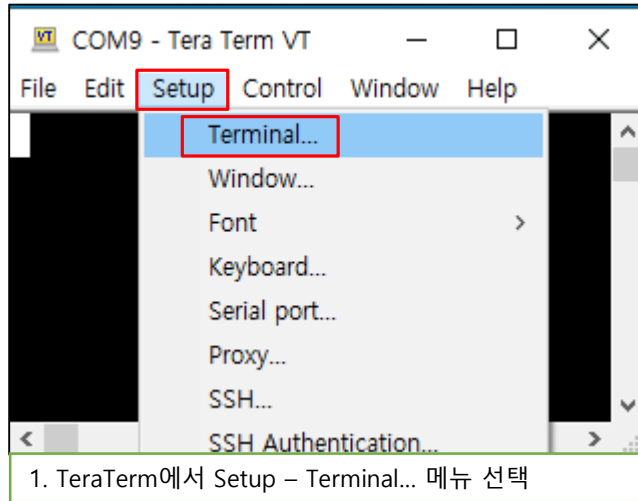
int main(void)
{
    char ch;

    UCSRA = 0b00000000; // U2X0=0: No double speed
    UCSRB = 0b00011000; // Enable Rx and Tx
    UCSRC = 0b00000110; // Async mode, No Parity,
                        // 1 Stop bit, 8 Data bits
    UBRR0 = 103;        // Baud Rate=16MHz/(9600*16)-1

    while (1)
    {
        // Wait until data is received (RXC0 bit)
        while ((UCSRA & (1 << 7)) == 0)
            ;
        ch = UDR0;        // Read received data

        // Wait until Tx Data Register Empty (UDRE0 bit)
        while ((UCSRA & (1 << 5)) == 0)
            ;
        UDR0 = ch;        // Transmit the received data
    }
}
```

ATmega328PB USART0 Example 2 (Rx/Tx, Polling) (3)



ATmega328PB USART0 Example 3 (문자열 송신, Polling) (1)

- Specifications:
 - baud rate: 1 Mbps, 8 data bits, no parity, 1 stop bit
- Transmit 'Switch pressed.' message when switch at PB7 is pressed and 'Switch released.' message when switch at PB7 is released .
- Use Polling method

ATmega328PB USART0 Example 3 (문자열 송신, Polling) (2)

```
#include <avr/io.h>

void uart0_puts(char *str);

int main(void)
{
    UCSRA = 0b00000000; // U2X0=0: No double speed
    UCSRB = 0b00011000; // Enable Rx and Tx
    UCSRC = 0b00000110; // Async mode, No Parity,
                        // 1 Stop bit, 8 Data bits
    UBRR0 = 0;          // Baud Rate=16MHz/(1M*16)-1

    while (1)
    {
        // switch at PB7 is pressed
        if ((PINB & (1 << 7)) == 0)
        {
            uart0_puts("Switch pressed.\n");

            // wait until switch at PB7 is released
            while ((PINB & (1 << 7)) == 0);
            uart0_puts("Switch released.\n");
        }
    }
}
```

```
void uart0_putchar(char ch)
{
    while (!(UCSRA & (1 << UDRE0))); // Wait for Tx ready

    UDR0 = ch; // Sends the data
}

void uart0_puts(char *str)
{
    while (*str != '\0')
    {
        uart0_putchar(*str);
        str++;
    }
}
```

USART0
END