

Chapter 22

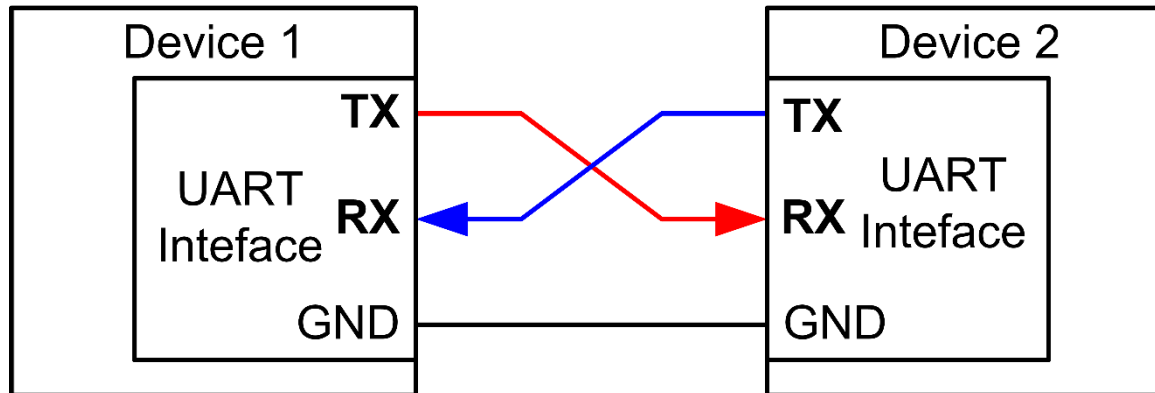
Serial Communication

Dr. Yifeng Zhu
Electrical and Computer Engineering
University of Maine

Spring 2018

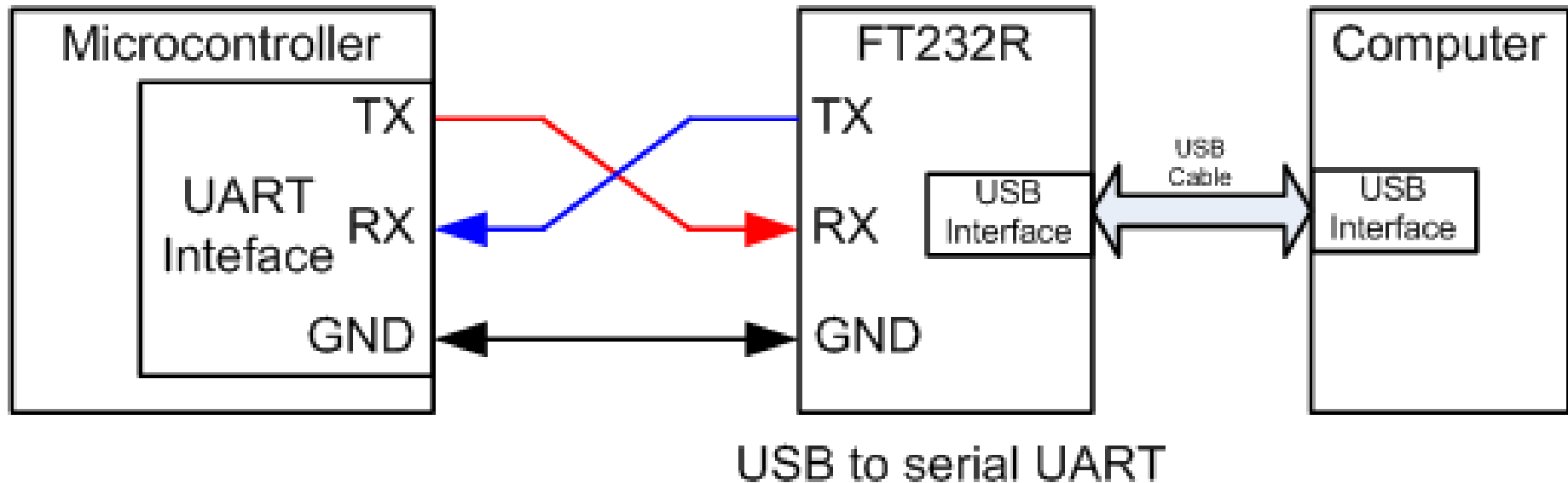
Universal Asynchronous Receiver and Transmitter (UART)

- ▶ **Universal**
 - ▶ UART is programmable.
- ▶ **Asynchronous**
 - ▶ Sender provides no clock signal to receivers

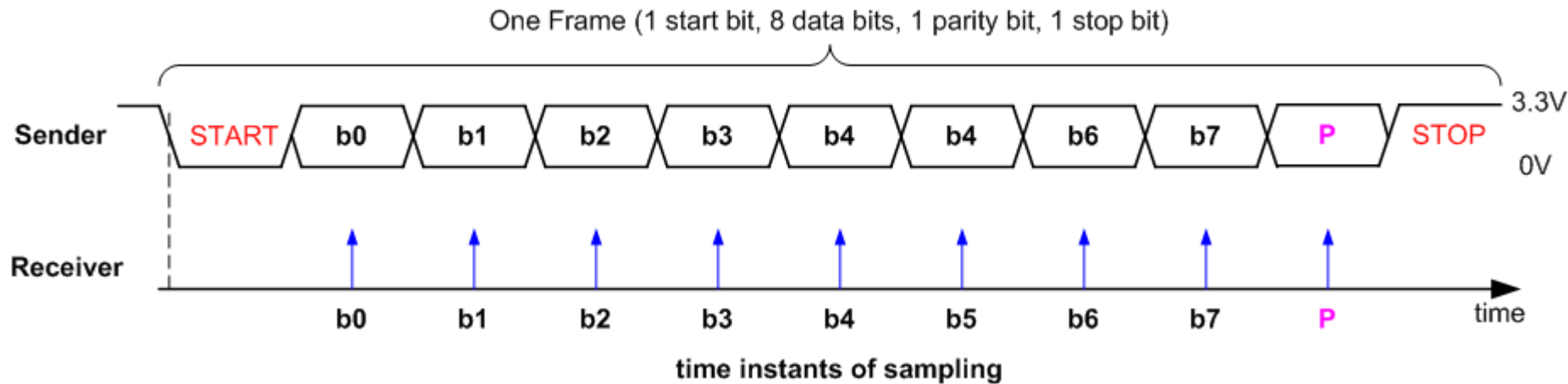


Connecting to PC

- ▶ FT232R converts the UART port to a standard USB interface



Data Frame



Tolerate 10% clock shift during transmission

- ▶ Sender and receiver uses the same transmission speed
- ▶ Data frame
 - ▶ One start bit
 - ▶ Data (LSB first or MSB, and size of 7, 8, 9 bits)
 - ▶ Optional parity bit
 - ▶ One or two stop bit

Baud Rate

- ▶ Historically used in telecommunication to represent the number of pulses physically transferred per second
- ▶ In digital communication, baud rate is the number of bits physically transferred per second
- ▶ Example:
 - ▶ Baud rate is 9600
 - ▶ each frame: a start bit, 8 data bits, a stop bit, and no parity bit.
 - ▶ Transmission rate of actual data
 - ~~$9600/8 = 1200 \text{ bytes/second}$~~
 - $9600/(1 + 8 + 1) = 960 \text{ bytes/second}$
 - ▶ The start and stop bits are the protocol overhead

Baud Rate

$$\text{Baud Rate} = \frac{f_{PCLK}}{8 \times (2 - OVER8) \times USARTDIV}$$

- ▶ If *OVER8* is 0, then the signal is oversampled by 16, and 4 bits are used for the fractional part.
- ▶ If *OVER8* is 1, then the signal is oversampled by 8, and 3 bits are used.
- ▶ If *BRR* is **0x1BC** and *OVER8* is 0, then **0x1B** is the integer part and **0xC** is the fractional part.
- ▶ $USARTDV = 0x1B + \frac{0xC}{0x10} = 27 + \frac{12}{16} = 27.75$

Baud Rate

- ▶ Suppose the processor clock f_{PCLK} is 16MHz, and the system is oversampled by 16 ($OVER8 = 0$),

- ▶ $USARTDIV = \frac{f_{PCLK}}{8 \times (2 - OVER8) \times Baud\ Rate}$

$$= \frac{16 \times 10^6}{8 \times (2 - 0) \times 9600} = 104.1667$$

- ▶ Thus $USARTDIV$ is 104.1875, which is encoded as 0x683.
- ▶ desired baud rate 9600

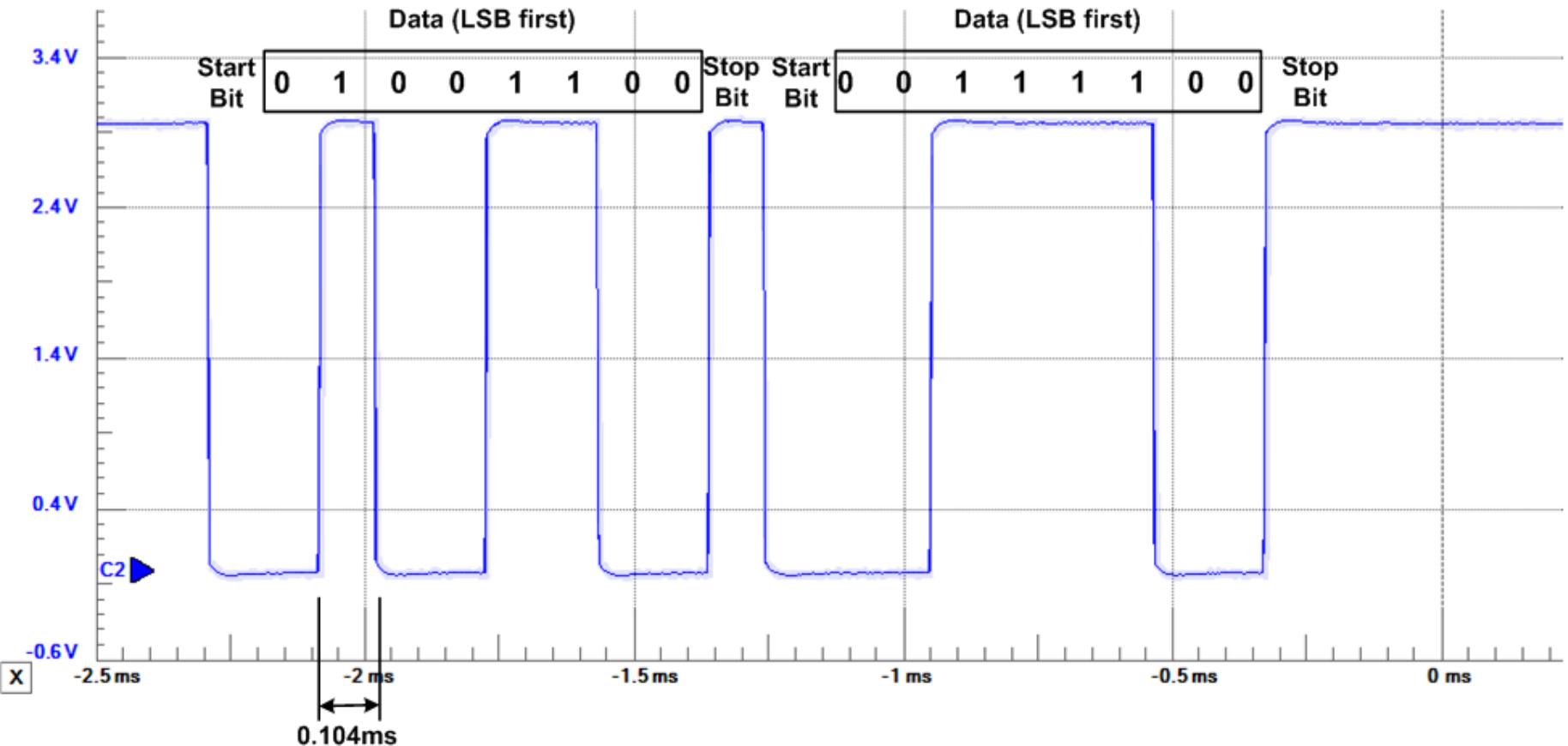
$$Baud\ Rate = \frac{16 \times 10^6}{8 \times (2 - 0) \times 104.1875} = 9598$$

Error Detection

- ▶ **Even Parity**: total number of “1” bits in data and parity is even
- ▶ **Odd Parity**: total number of “1” bits in data and parity is odd
- ▶ Example: Data = 10101011 (five “1” bits)
 - ▶ The parity bit should be 0 for odd parity and 1 for even parity
- ▶ This can detect single-bit data corruption

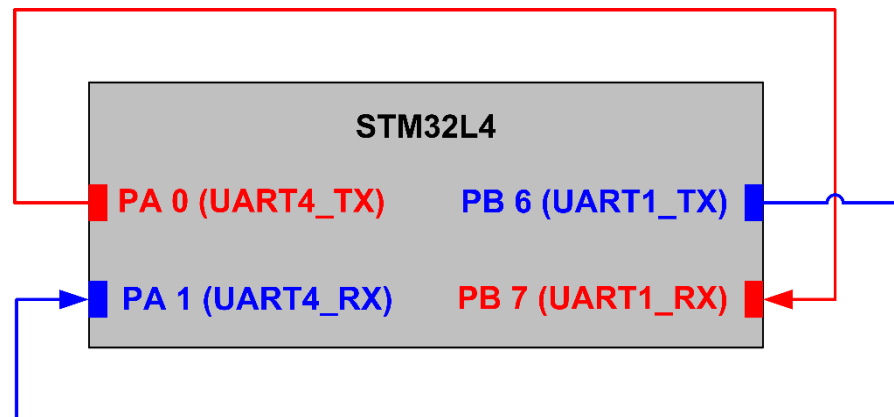
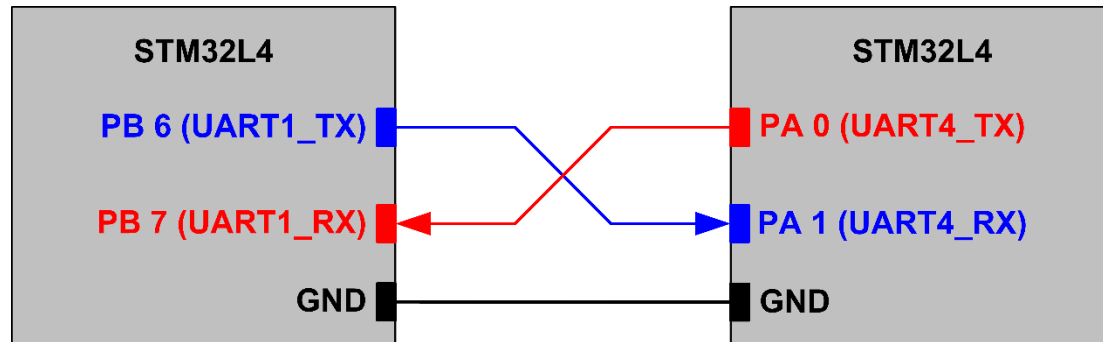


Transmitting 0x32 and 0x3C



1 start bit, 1 stop bit, 8 data bits, no parity, baud rate = 9600

UART Connection



Sending Data

```
void USART_Write(USART_TypeDef * USARTx, uint8_t * buffer, int nBytes) {
    int i;

    // TXE is cleared by a write to the USART_DR register.
    // TXE is set by hardware when the content of the TDR
    // register has been transferred into the shift register.

    for (i = 0; i < nBytes; i++) {
        // wait until TXE (TX empty) is set
        // Writing USART_DR automatically clears the TXE flag
        while (!(USARTx->SR & USART_SR_TXE));
        USARTx->DR = (buffer[i] & 0xFF);
    }

    while (!(USARTx->SR & USART_SR_TC));    // wait until TC bit is set
    USARTx->SR &= ~USART_SR_TC;
}
```

Receiving Data

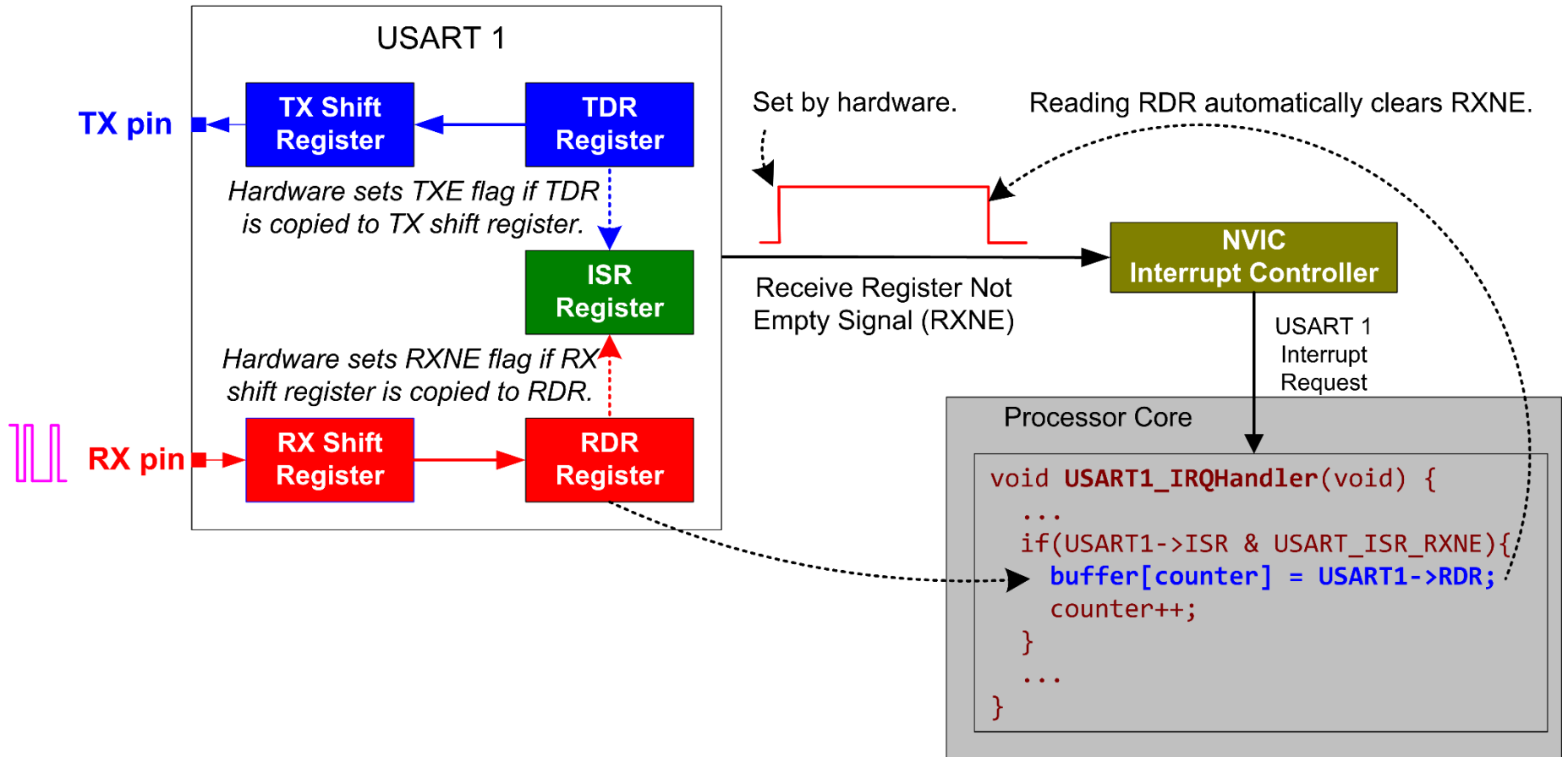
```
void USART_IRQHandler(USART_TypeDef * USARTx, uint8_t * buffer,  uint8_t * pRx_counter){
    if(USARTx->SR & USART_SR_RXNE) { // Received data
        buffer[*pRx_counter] = USARTx->DR;
        // Reading USART_DR automatically clears the RXNE flag
        (*pRx_counter)++;
        if((*pRx_counter) >= BufferSize )
            (*pRx_counter) = 0;
    }
}

void USART1_IRQHandler(void) {
    USART_IRQHandler(USART1, USART1_Buffer_Rx, &Rx1_Counter);
}

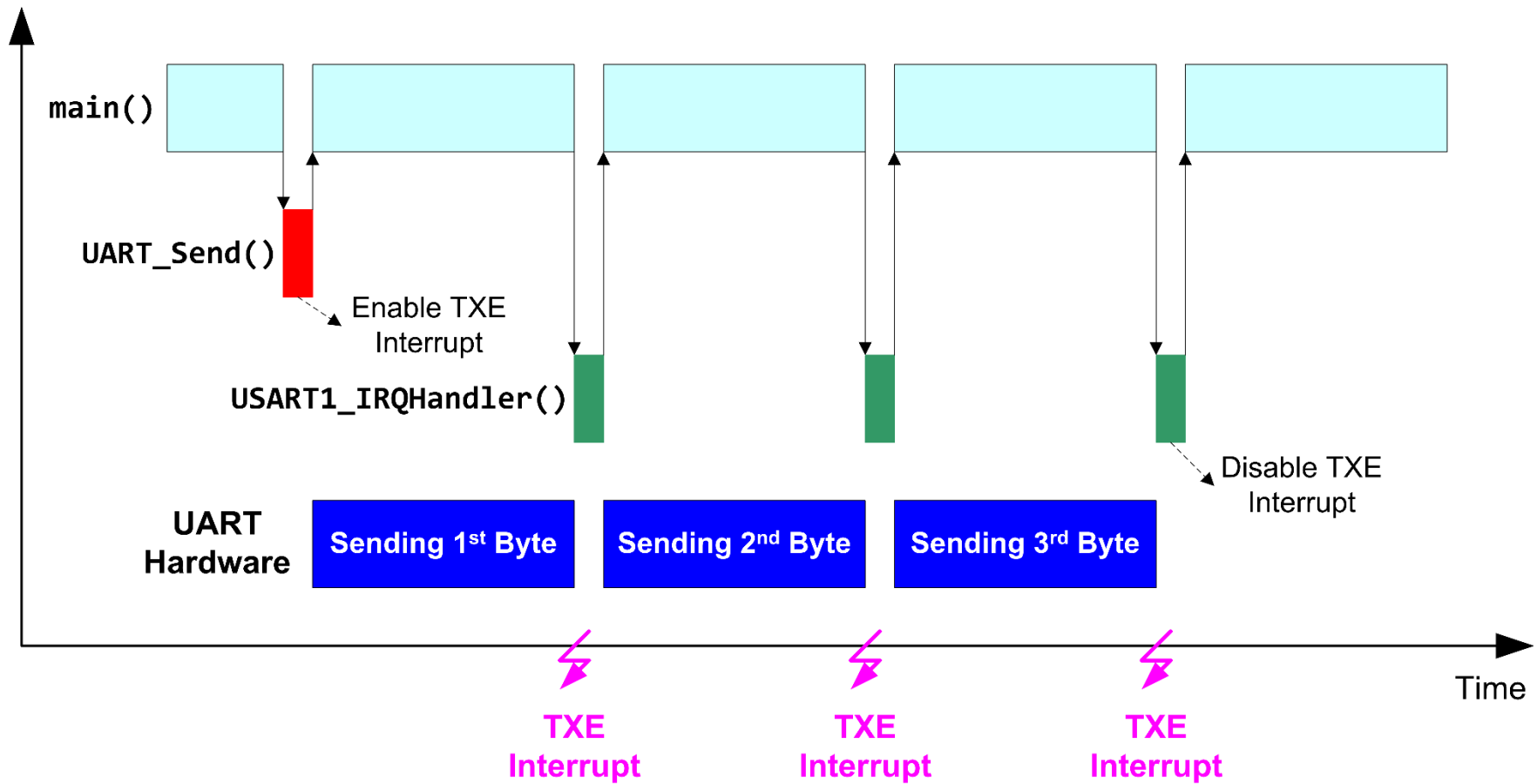
void USART2_IRQHandler(void) {
    USART_IRQHandler(USART2, USART2_Buffer_Rx, &Rx2_Counter);
}
```



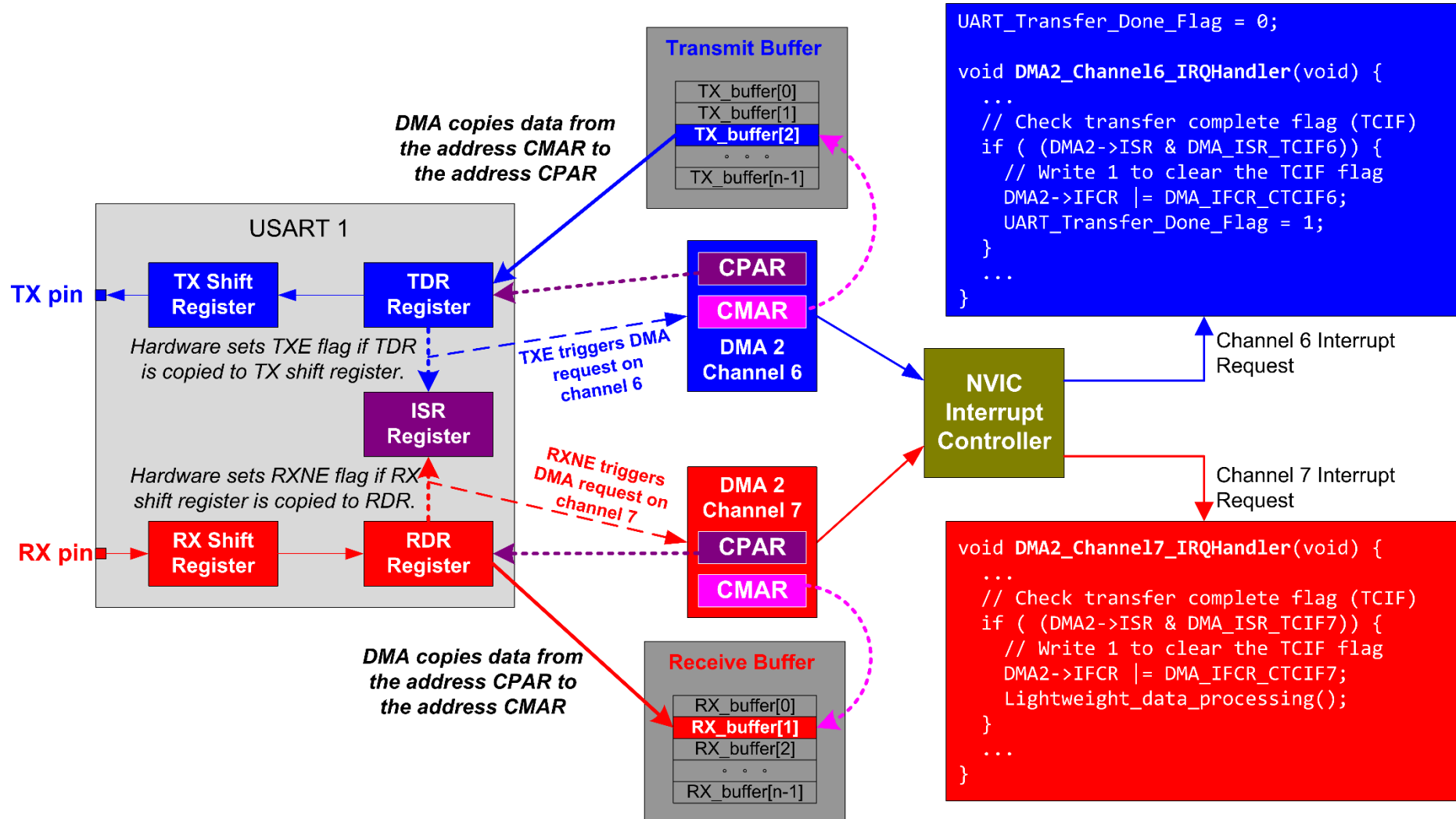
UART Interrupt: Receiving Data



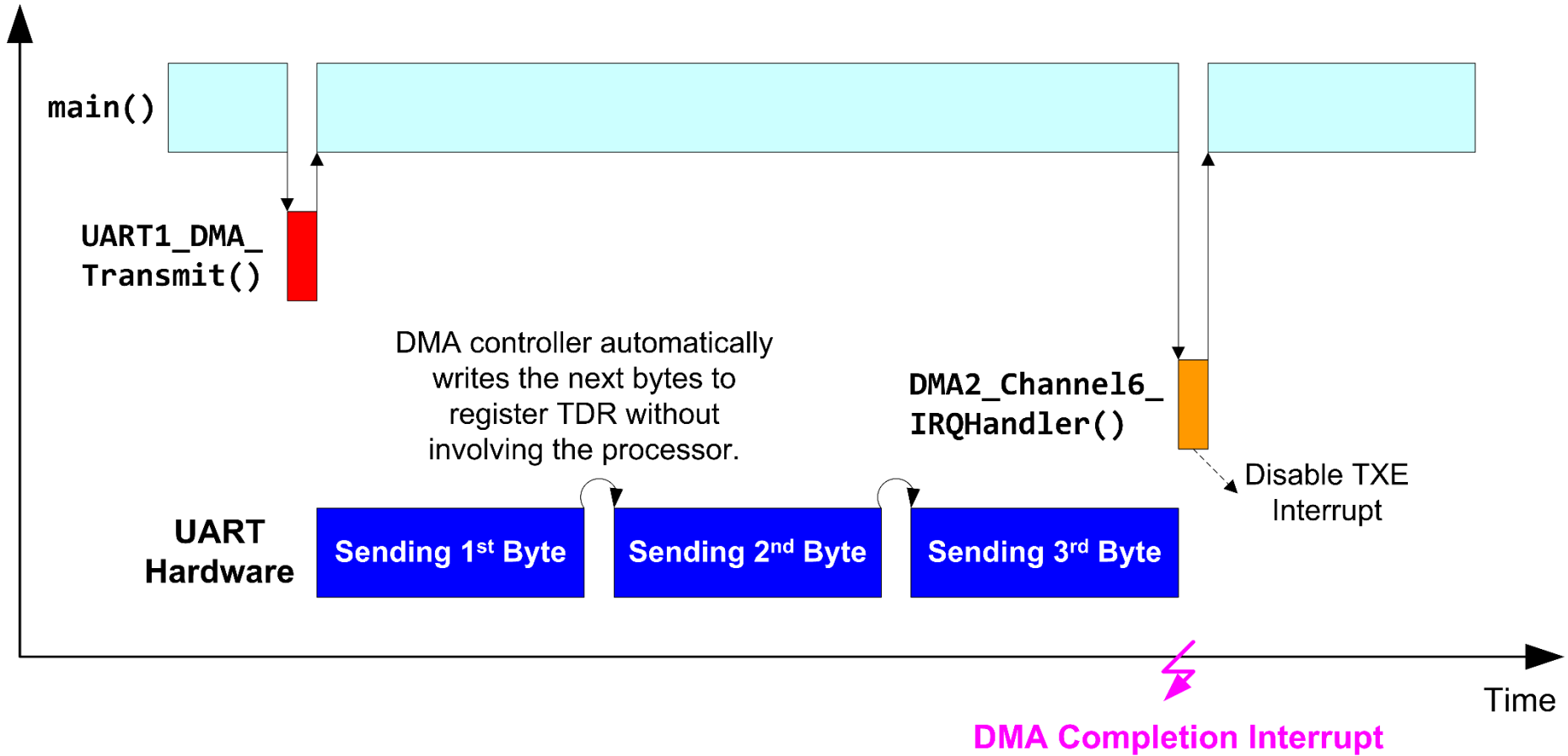
UART Interrupt: Receiving Data



UART DMA: Receiving & Sending



UART DMA: Receiving & Sending



Voltage Levels

Standard	Voltage signal	Max distance	Max speed	Number of devices supported per port
RS-232	Single end (logic 1: +5 to +15V, logic 0: -5 to -15 V)	100 feet	115Kbit/s	1 master, 1 receiver
RS-422	Differential (-6V to +6V)	4000 feet	10Mbit/s	1 master, 10 receivers
RS-485	Differential (-7V to +12V)	4000 feet	10Mbit/s	32 masters, 32 receivers

Bluetooth

