	STM32L1	**STM32L4**
USART	3 USART, 2 UART	3 USART, 2 UART, 1 LPUART
	up to 4 Mbit/s (when the clock frequency is 32 MHz and oversampling is by 8)	up to 10 Mbit/s (when the clock frequency is 80 MHz and oversampling is by 8)
	Programmable word length (8 or 9 bits)	Programmable word length (7, 8 or 9 bits), programmable data order with MSB-first or LSB-first shifting
	10 interrupt sources with flags	14 interrupt sources with flags
	U(S)ART clock is APB1 or APB2 clock	U(S)ART clock is derived from one of the four following sources: system clock (SYSCLK), HSI16, LSE, APB1 or APB2 clock

	Data structure
	`typedef struct {`
	`__IO uint16_t SR;`
	`uint16_t RESERVED0;`
	`__IO uint16_t DR;`
	`uint16_t RESERVED1;`
	`__IO uint16_t BRR;`
	`uint16_t RESERVED2;`
	`__IO uint16_t CR1;`
	`uint16_t RESERVED3;`
	`__IO uint16_t CR2;`
	`uint16_t RESERVED4;`
	`__IO uint16_t CR3;`
	`uint16_t RESERVED5;`
	`__IO uint16_t GTPR;`
	`uint16_t RESERVED6;`
	`} USART_TypeDef;`
		`typedef struct {`
		`__IO uint32_t CR1;`
		`__IO uint32_t CR2;`
		`__IO uint32_t CR3;`
		`__IO uint32_t BRR;`
		`__IO uint16_t GTPR;`
		`uint16_t RESERVED2;`
		`__IO uint32_t RTOR;`
		`__IO uint16_t RQR;`
		`uint16_t RESERVED3;`
		`__IO uint32_t ISR;`
		`__IO uint32_t ICR;`
		`__IO uint16_t RDR;`
		`uint16_t RESERVED4;`
		`__IO uint16_t TDR;`
		`uint16_t RESERVED5;`
		`} USART_TypeDef;`

	`USARTx->DR`	`USARTx->TDR or USARTx->RDR`
	`USARTx->SR`	`USARTx->ISR`
	`Clear status flags via USARTx->SR`	`Clear status flags via USARTx->ICR`