	STM32L1	STM32L4
SPI	Data size is fixed, configurable to 8 or 16 bits	Data size is programmable, from 4 to 16-bit
	Tx & Rx 16-bit buffers (single data frame)	32-bit Tx & Rx FIFOs (up to 4 data frames) Rx buffer not empty (RXNE): The RXNE flag is set depending on the FRXTH bit value in the SPIx_CR2 register: (1) If FRXTH is set, RXNE goes high and stays high until the RXFIFO level is greater or equal to 1/4 (8-bit). (2) If FRXTH is cleared, RXNE goes high and stays high until the RXFIFO level is greater than or equal to 1/2 (16-bit). **SPIx->CR2 |= SPI_CR2_FRXTH;**
	No data packing (16-bit access only) `SPIx->DR = byte_data;`	Data packing (8-bit, 16-bit or 32-bit data access, programmable FIFOs data thresholds) **`*((volatile uint8_t*)&SPIx->DR) = byte_data;`** **`byte_data = (uint8_t)(SPIx->DR);`**
		The data size and Tx/Rx flow handling are different in STM32L1 and STM32L4 series hence requiring different SW sequences

	`typedef struct{`
	`  __IO uint16_t CR1;`
	`  __IO uint16_t CR2;`
	`  __IO uint16_t SR;`
	`  __IO uint16_t DR;`
	`  __IO uint16_t CRCPR;`
	`  __IO uint16_t RXCRCR;`
	`  __IO uint16_t TXCRCR;`
	`} SPI_TypeDef;`
		`typedef struct{`
		`  __IO uint32_t CR1;`
		`  __IO uint32_t CR2;`
		`  __IO uint32_t SR;`
		`  __IO uint32_t DR;`
		`  __IO uint32_t CRCPR;`
		`  __IO uint32_t RXCRCR;`
		`  __IO uint32_t TXCRCR;`
		`} SPI_TypeDef;`