

# 09 SPI

5 Mayıs 2021 Çarşamba 08:03

## 09 SPI

### Giriş

- <https://ozdenercin.com/2019/02/01/spi-seri-haberlesme-protokolu/> , <https://arduinodestek.com/spi-haberlesme-protokolu-nedir-ve-nasıl-gerçeklesir/> <https://devreyakan.com/spi-nedir/> linkinden ayrıntılı bilgilere ulaşabiliriz.
- SPI (Serial Peripheral Interface), mikrodenetleyiciler, sensörler, dijital IC'ler ve diğer entegre devreler arasında seri veri iletişimini için kullanılan bir seri senkron iletişim protokolüdür
- Özellik ve kullanım olarak I2C'ye benzer. I2C'de olduğu gibi bir adet Master cihaz bulunur. Bu cihaz hatta bağlı çevresel cihazları kontrol eder.
- Çevresel cihazlarla veya diğer mikrodenetleyicilerle veri transferi sağlayan yazılım/donanım tabanlı seri iletişim protokolüdür. Bu haberleşme şekli karşılıklı iki tarafın **clocklarının senkronize çalışmasıyla** data iletişimini sağlamaktadır.
- SPI, genellikle düşük maliyetli, düşük güç tüketimi gerektiren uygulamalarda kullanılır. Özellikle mikrodenetleyiciler, sensörler, veri dönüştürücüler, hafıza kartları ve diğer entegre devreler arasında veri iletişimini tercih edilir.
- SPI'nin hızlı ve basit bir iletişim protokolü olması, çeşitli uygulama alanlarında yaygın olarak kullanılmasını sağlar.

### Avantajları ve Dezavantajları

Avantajlar;

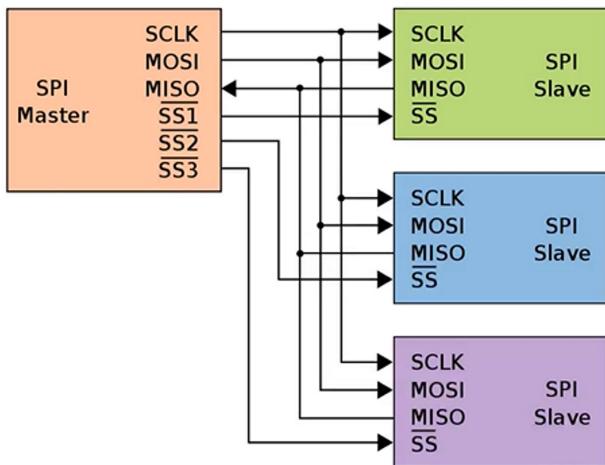
- Başlatma ve durdurma biti yok, böylece veriler kesintisiz olarak aktarılabilir.
- I2C gibi karmaşık bağımlı adresleme sistemi yok.
- I2C'den daha yüksek veri aktarım hızı (neredeysse iki kat daha hızlı).
- Ayri MISO ve MOSI hatları, böylece veri aynı anda gönderilebilir ve alınabilir.

Dezavantajlar;

- Dört kablo kullanır (I2C ve UART'lar iki kablo kullanır).
- Verilerin başarıyla alındığına dair bir onay yok (I2C de vardır).
- UART'taki eşlik biti gibi hata denetimi biçimini yok.
- Yalnızca tek bir master'a izin verir.

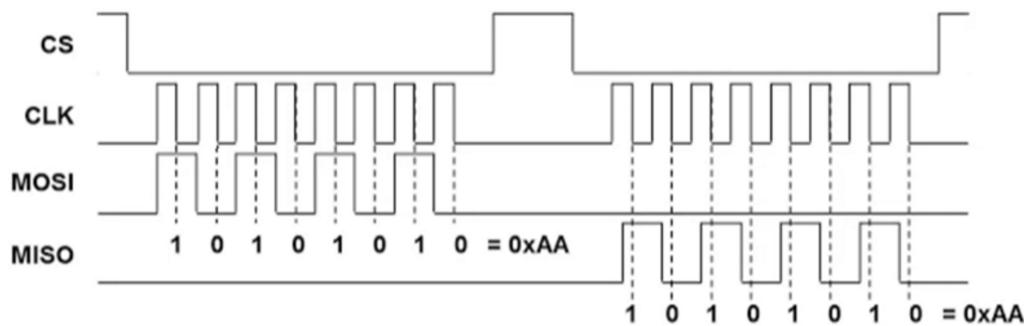
### Bağlantılar

- SPI, genellikle dört telli bir bağlantıyla gerçekleştiriliyor:  
**MOSI** (Master Out Slave In), Master cihazdan (genellikle mikrodenetleyici) slave cihaza veri gönderir.  
**MISO** (Master In Slave Out), Slave cihazdan master cihaza veriyi gönderir.  
**SCLK** (Serial Clock): Saat sinyali, veri iletim hızını senkronize eder.  
Bu sinyal sadece master cihaz tarafından üretilir.  
**SS/CS** (Slave Select/Chip Select), İletişim kurulacak slave cihazı seçer.
- Slave cihaz donanımsal olarak seçildiği için I2C iletişimindeki gibi adres gönderilmez. Fakat birden fazla slave cihazın SPI veri yoluna bağlanması için birden fazla SS/CS pini kullanır.  
Tüm pinlerin kullanılmasına ihtiyaç yoktur.

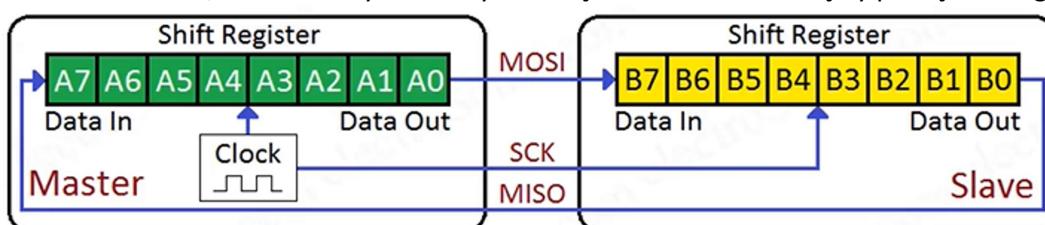


## Veri İletimi

- Master, saat sinyalini verir.
- Master, SS/CS pinini, slave etkinleştiren bir **LOW** voltaj durumuna geçirir.
- Master, verileri MOSI hattı boyunca her seferinde bir bit olarak slave'ye gönderir. Slave, bitleri alındıkça okur.
- Bir yanıt gerekiyorsa, bağımlı, MISO hattı boyunca her seferinde bir bit veriyi master'a döndürür. Master, bitleri alındıkça okur.



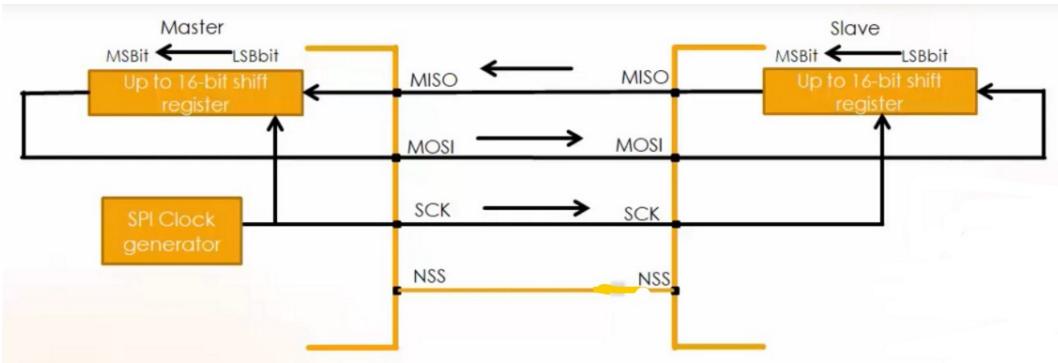
- SPI'da veri iletim sırası genellikle **8 bitlik** veri paketleri halinde olur, ancak bu uzunluğu değiştirilebilir. Veri iletim sırası, verilerin en yüksek veya en düşük anlamlı bit ile başlayıp bitişebileceğinin şekilde yapılandırılabilir.



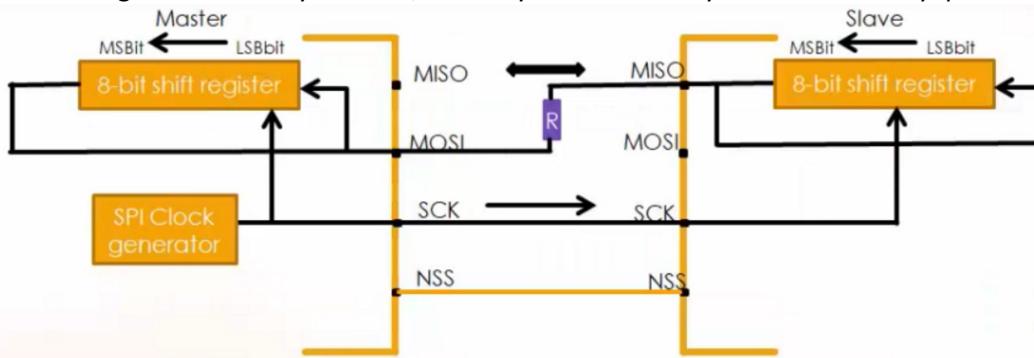
- Ana cihaz, iletişimi **başlatır** ve **sonlandırır**. Slave cihazlar ise ana cihazın taleplerine **yanıt** verir.

## Veri Yolu

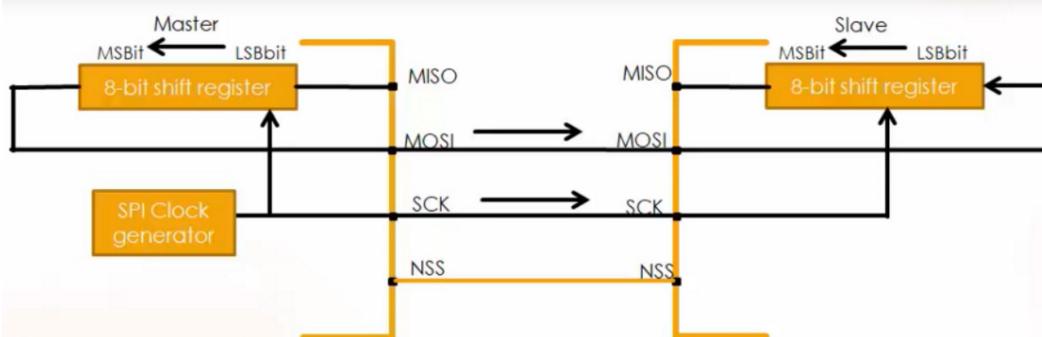
- SPI haberleşmesi için üç farklı mod vardır. Bunlar Full Duplex, Half duplex ve Simplex'dır.
- İkişi çift yönlü iken diğeri tek yönlü haberleşmedir. Bu modlar hakkında detaylı bilgi almak için <https://fastbitlab.com/spi-bus-configuration-discussion-full-duplex-half-duplex-simplex/> linkteki yazıyı okuyabiliriz.
- Tek yönlü olan Simplex iletişiminde SS/CS pini kullanılmayabilir, ancak çift yönlü olan Full Duplex ve Half duplex iletişimde ve birden fazla slave cihazı varsa SS/CS pini kullanışlı olabilir.
- **Full Duplex**, hem veri **gonderme** hem de veri **alma** işlemlerinin **aynı anda** gerçekleştirir. Veri iletimi için iki ayrı iletişim hattı kullanılır. Her iki tarafta bağımsız olarak veri gönderebilir ve alabilir, bu nedenle iletişim hızı genellikle yüksektir.



- **Half Duplex**, sadece bir veri gönderme ya da bir veri alma işleminin aynı anda gerçekleştirildiği bir çalışma modudur.  
Bu modda, genellikle **tek bir çift yönlü** iletişim hattı kullanılır. Her iki taraf da **aynı iletişim hattını** kullanarak veri gönderebilir veya alabilir, ancak aynı anda her iki yönde veri iletimi yapılamaz.



- **Simplex**, sadece **bir yönde** veri iletimine izin veren bir çalışma modudur.  
Genellikle tek bir iletişim hattı kullanılır ve veri gönderme veya veri alma işlemi yapılır. Her iki tarafta aynı anda veri gönderip alamaz.
- Eğer master sadece veri gönderip slave sadece veri alacaksa, bu durumda **MOSI** hattı kullanılır. Diğer bir durumda, master sadece veri alıp slave sadece veri gönderecekse, bu durumda **MISO** hattı tercih edilir.
- Bu iletişimde SS/CS pini kullanılmayabilir. Çünkü Simplex modda genellikle tek yönlü iletişim olduğu için sadece master cihazın veri gönderme veya alım işlemlerini kontrol etmesi yeterlidir.



## Çalışma Modları

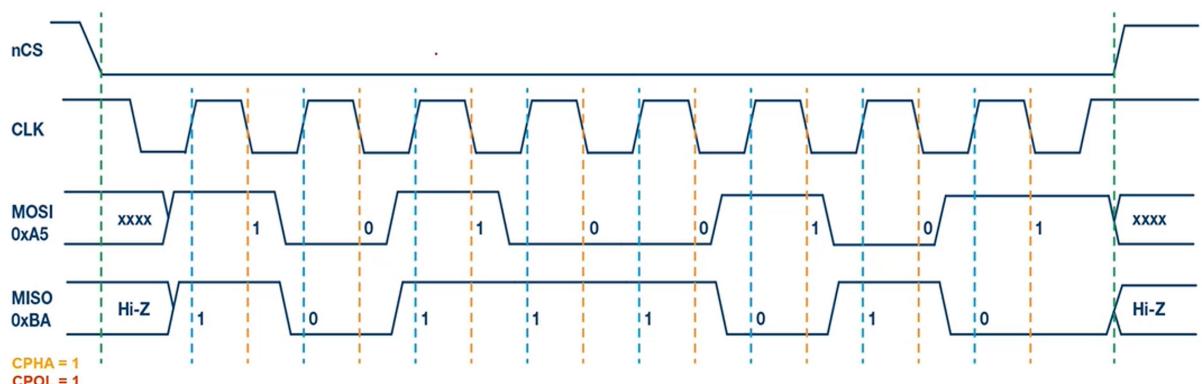
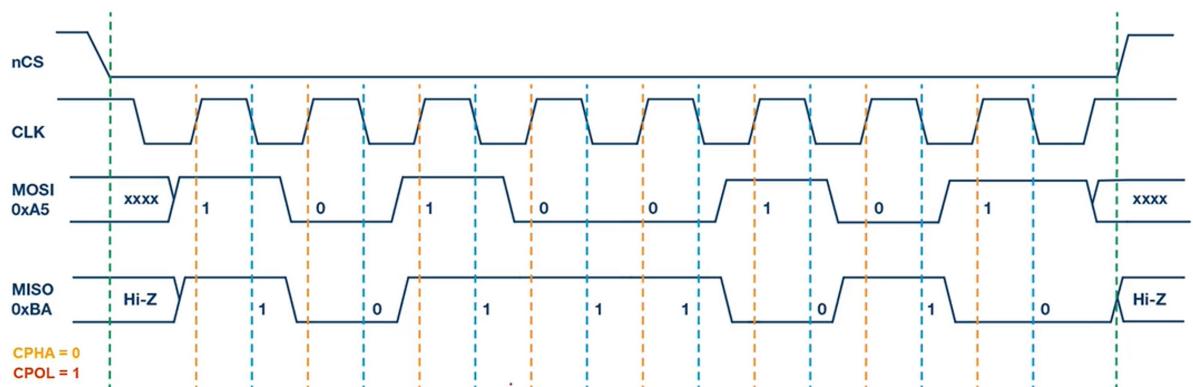
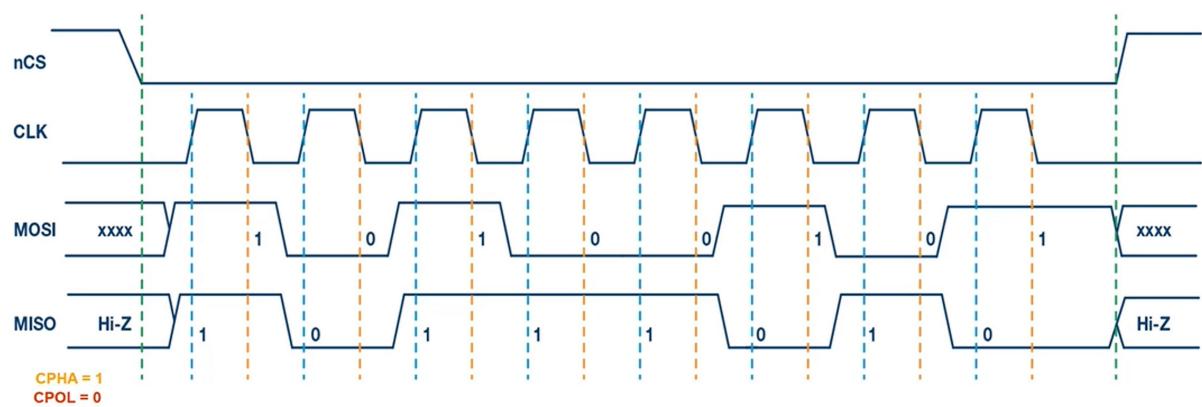
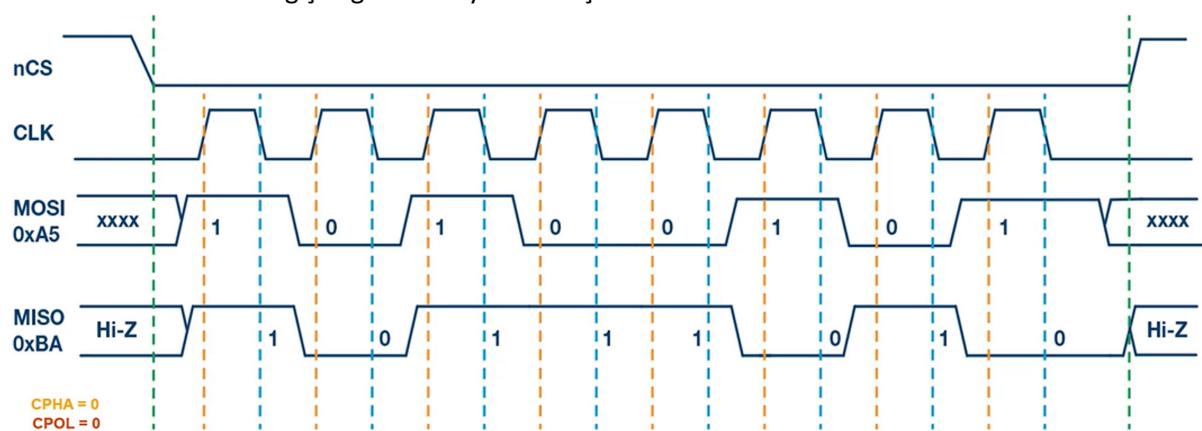
- SPI iletişiminde saat sinyalının nasıl üretileceğini belirler. Clock polarity (CPOL), saat sinyalının **yüksek** veya **düşük** seviyede başlayacağını belirtir, clock phase (CPHA) ise veri okuma/yazma işleminin saat sinyalının hangi **kenarında** gerçekleşeceğini belirler.
- **Saat sinyali**, CPOL değeri 0 olduğunda düşük seviyeden, 1 olduğunda ise yüksek seviyeden başlar.  
**Veri**, CPHA değeri 0 iken saat sinyalının yükselen kenarında yazılır sonraki düşen kenarında okunur, CPHA değeri 1 iken ise düşen kenarında yazılır, sonraki yükselen kenarında yazılır.
- **SPI Modu 0 (CPOL=0, CPHA=0)**  
CPOL = 0: Saat sinyali, düşük seviyede başlar.  
CPHA = 0: Veri değişikliği saat sinyalının yükselen kenarında olur.
- **SPI Modu 1 (CPOL=0, CPHA=1)**  
CPOL = 0: Saat sinyali, düşük seviyede başlar.  
CPHA = 1: Veri değişikliği saat sinyalının düşen kenarında olur.
- **SPI Modu 2 (CPOL=1, CPHA=0)**  
CPOL = 1: Saat sinyali, yüksek seviyede başlar.

CPHA = 0: Veri değişikliği saat sinyalinin yükselen kenarında olur.

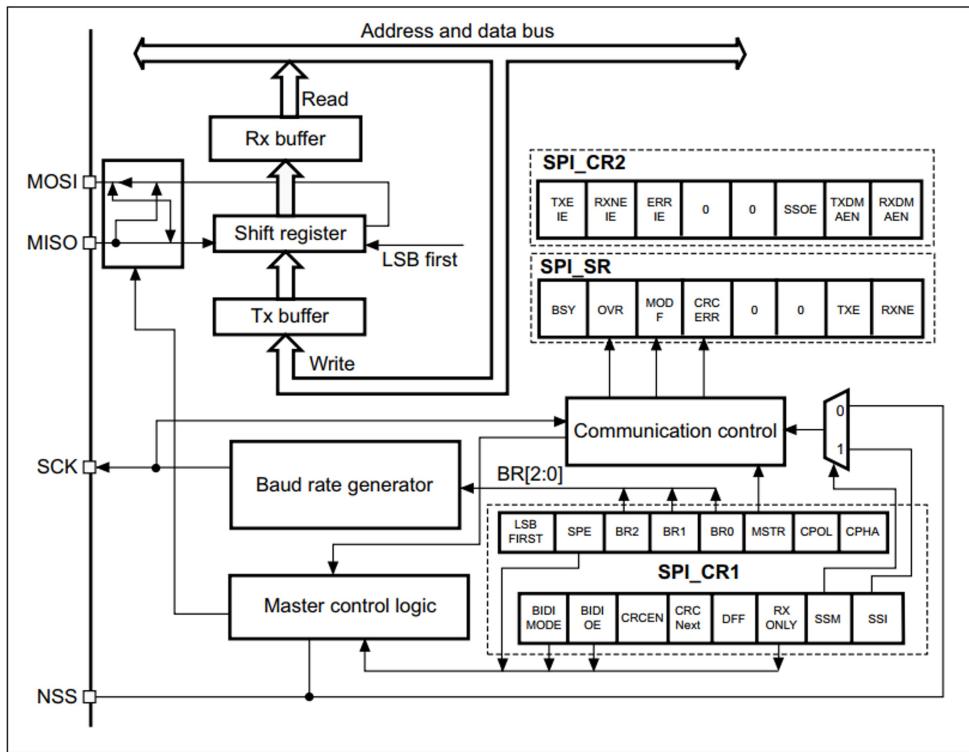
- **SPI Modu 3 (CPOL=1, CPHA=1)**

CPOL = 1: Saat sinyali, yüksek seviyede başlar.

CPHA = 1: Veri değişikliği saat sinyalinin düşen kenarında olur.



## Birim Yapısı



## Register

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	<b>SPI_CR1</b>	Reserved												BIDIMODE	15	BIDIOE	14	CRCEN	13	CRCNEXT	12	DFF	11	RXONLY	10	SSM	9	SSI	8	LSBFIRST			
		Reset value												0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x04	<b>SPI_CR2</b>	Reserved												TXEIE	7	RXNEIE	0	ERRIE	0	FRF	0	Reserved	0	SSOE	0	MSTR	2	CPOL	1	CPHA	0		
		Reset value												0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x08	<b>SPI_SR</b>	Reserved												FRE	7	BSY	6	OVR	5	MODF	4	CRCERR	3	UDR	2	CHSIDE	1	TXE	0	RXNE	0		
		Reset value												0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x0C	<b>SPI_DR</b>	Reserved												DR[15:0]															0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0				
		Reset value												CRCPOLY[15:0]															0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0				
0x10	<b>SPI_CRCPR</b>	Reserved												RxCRC[15:0]															0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0				
		Reset value												TxCRC[15:0]															0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0				
0x14	<b>SPI_RXCRCR</b>	Reserved												0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0															0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0				
		Reset value												0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0															0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0				
0x18	<b>SPI_TXCRCR</b>	Reserved												0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0															0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0				
		Reset value												0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0															0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0				

- **SPI\_CR1 (Control Register 1)** ve **SPI\_CR2 (Control Register 2)**, iletişim modunu (Master veya Slave) ve saat hızını, Data frame format, Half duplex iletişim ve diğer özelliklerini yapılandırmak için kullanılır.
- **SPI\_SR (Status Register)**, SPI haberleşme durumunu izlemek için kullanılır. İletimin tamamlanıp tamamlanmadığı, veri alımı durumu gibi bilgileri içerir.
- **SPI\_DR (Data Register)**, Veri gönderip almak için kullanılır. Gönderilen veya alınan veriyi bu kayıt aracılığıyla işleyebilirsiniz.
- **SPI\_CRCPR (CRC Polynomial Register)** ve **SPI\_RXCRCR/SPI\_TXCRCR (CRC Receive/Transmit Register)**, SPI verilerinin döngüsel hata denetimi (CRC) için kullanılır.

## Haberleşme Metotları

- SPI üzerinden Polling, Interrupt ve DMA olmak üzere üç farklı haberleşme yapılabilir.
- <https://deepbluembedded.com/stm32-spi-tutorial/> linkinden konu hakkındaki bilgileri inceleyebiliriz.