

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-nasil-gerceklesir/> <https://devreyakan.com/spi-nedir/> linkinden ayrıntılı bilgilere ulaşabiliriz.
- SPI (Serial Peripheral Interface), mikrodnetleyiciler, sensörler, dijital IC'ler ve diğer entegre devreler arasında seri veri iletişimi 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 mikrodnetleyicilerle veri transferi sağlayan yazılım/donanım tabanlı seri iletişim protokolüdür. Bu haberleşme şekli karşılıklı iki tarafın **clocklarının senkronize çalışmasıyla** data iletişimi sağlamaktadır.
- SPI, genellikle düşük maliyetli, düşük güç tüketimi gerektiren uygulamalarda kullanılır. Özellikle mikrodnetleyiciler, sensörler, veri dönüştürücüler, hafıza kartları ve diğer entegre devreler arasında veri iletişimi için 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 (neredeyse iki kat daha hızlı).
- Ayrı MISO ve MOSI hatları, böylece veri aynı anda gönderilebilir ve alınabilir.

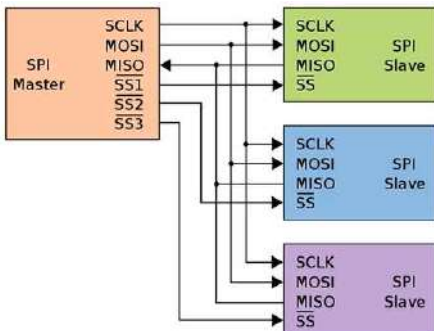
Dezavantajlar;

- 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 biçimi yok.
- Yalnızca tek bir master'a izin verir.

Bağlantılar

- SPI, genellikle dört telli bir bağlantıyla gerçekleştirilir:
MOSI (Master Out Slave In), Master cihazdan genellikle mikrodnetleyici 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.

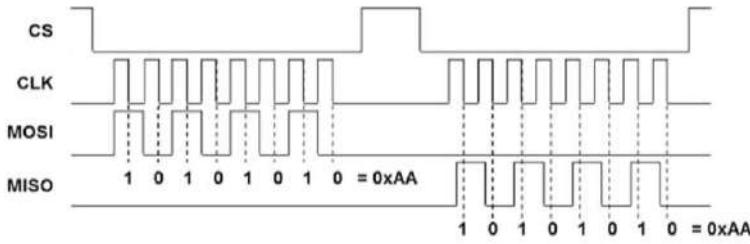
Tüm pinlerin kullanılmasına ihtiyaç yoktur.



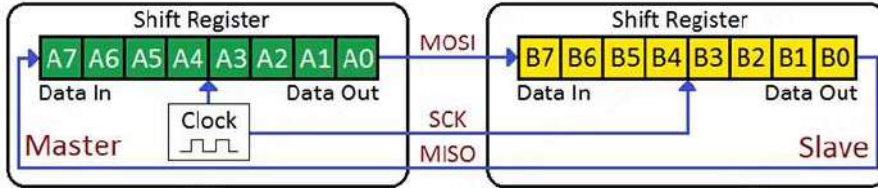
Veri İletimi

- 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'e gönderir. Slave, bitleri aldıkça okur.
- Bir yanıt gerekiyorsa, slave MISO hattı boyunca her seferinde bir bit veriyi master'a döndürür. Master, bitleri aldıkça okur.



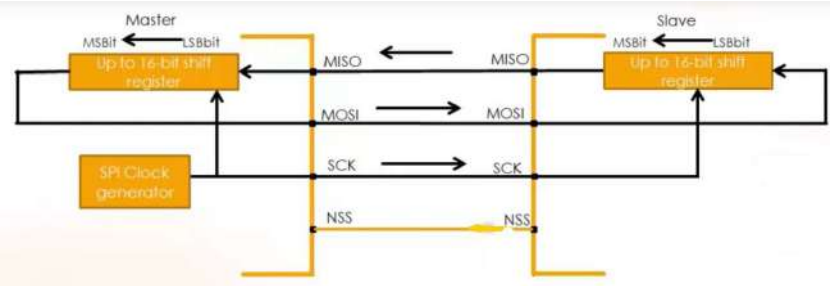
- SPI'da veri iletim sırası genellikle **8 bitlik** veri paketleri halinde olur, ancak bu uzunluğu değiştirilebilir. Veri iletim sırası, verilerin en yüksek veya en düşük anlamlı bit ile başlayıp bitebileceği şekilde yapılandırılabilir.



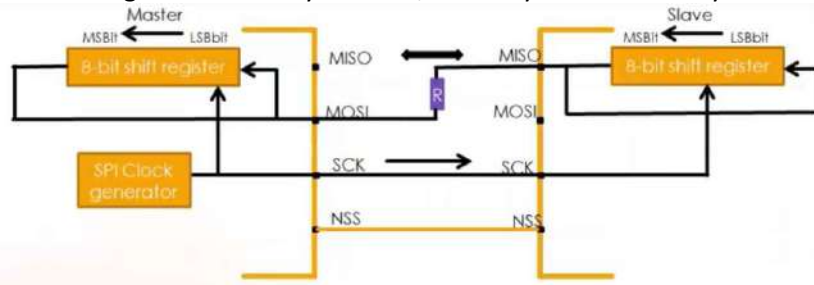
- Ana cihaz, iletişimi **başlatır** ve **sonlandırır**. Slave cihazlar ise ana cihazın taleplerine **yanıt** verir.

Veri Yolu

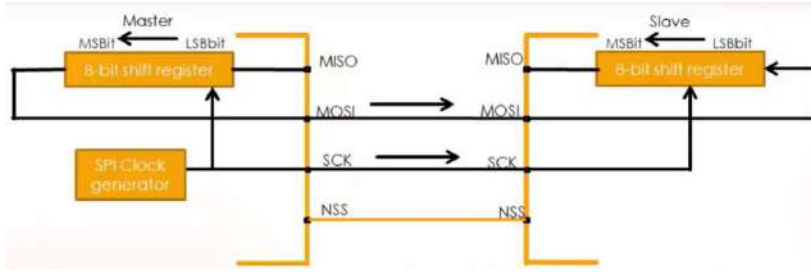
- SPI haberleşmesi için üç farklı mod vardır. Bunlar Full Duplex, Half duplex ve Simplex'dir.
- İki çift yönlü iken diğeri tek yönlü haberleşmedir. Bu modlar hakkında detaylı bilgi almak için <https://fastbitlab.com/spi-bus-configuration-discussion-full-duplex-half-duplex-simplex/> linkteki yazıyı okuyabiliriz.
- Tek yönlü olan Simplex iletişimde 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 **gönderme** 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.

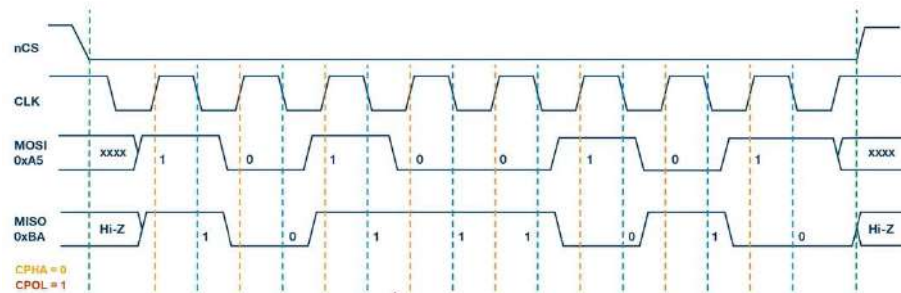
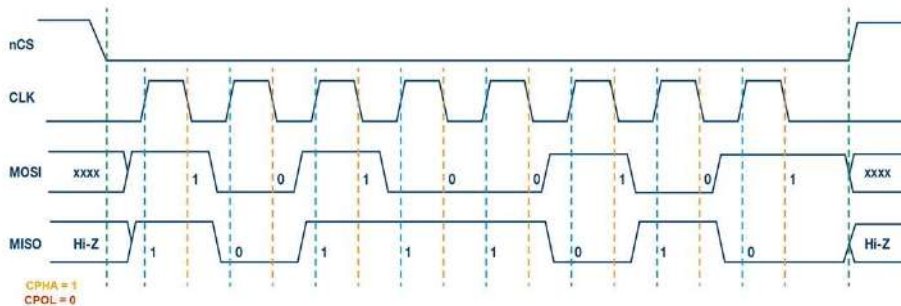
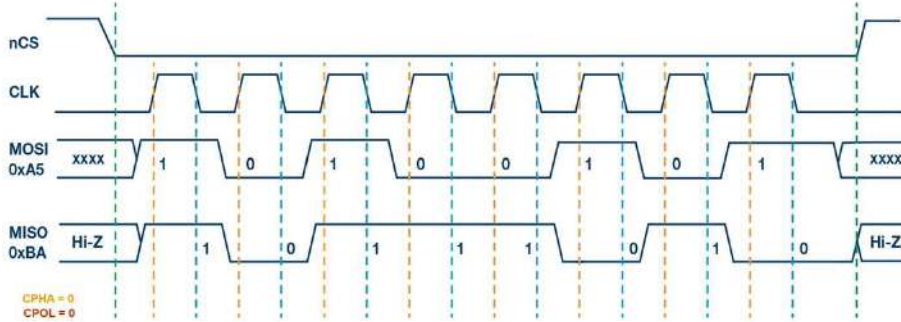


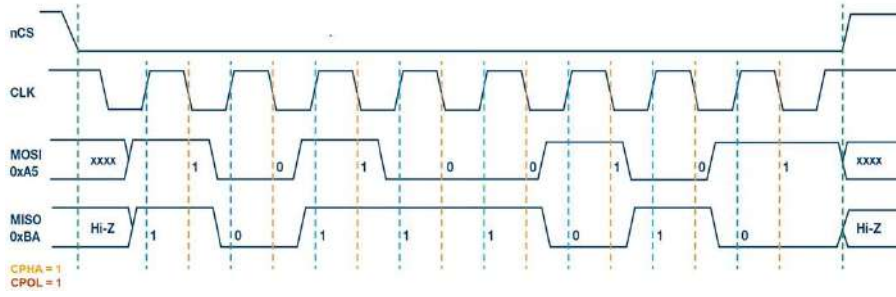
- **Simplex**, sadece **bir yönde** veri iletimine izin veren bir çalışma modudur. Genellikle tek bir iletişim hattı kullanılır ve veri gönderme veya veri alma işlemi yapılır. Her iki tarafta aynı anda veri gönderip alamaz.
- Eğer master sadece veri gönderip slave sadece veri alacaksa, bu durumda **MOSI** hattı kullanılır. Diğer bir durumda, master sadece veri alıp slave sadece verecekse, 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.



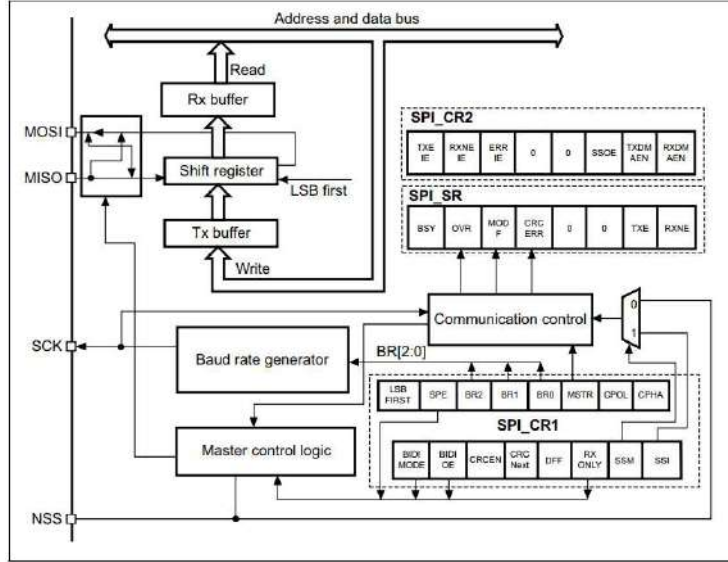
Çalışma Modları

- SPI iletişimde saat sinyalinin nasıl üretileceğini belirler. Clock polarity (CPOL), saat sinyalinin **yüksek** veya **düşük** seviyede başlayacağını belirtir, clock phase (CPHA) ise veri okuma/yazma işleminin saat sinyalinin hangi **kenarında** gerçekleşeceğini belirler.
- Saat sinyali**, CPOL değeri 0 olduğunda düşük seviyeden, 1 olduğunda ise yüksek seviyeden başlar.
Veri, CPHA değeri 0 iken saat sinyalinin 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 sinyalinin yükselen kenarında olur.
- SPI Modu 1 (CPOL=0, CPHA=1)**
CPOL = 0: Saat sinyali, düşük seviyede başlar.
CPHA = 1: Veri değişikliği saat sinyalinin 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	SPI_CR1	Reserved																BIDMODE	BIDIOE	CRCEN	CRCNEXT	DFR	RXONLY	SSM	SSI	LSBFIRST	SPE	BR [2:0]			MSTR	CPOL	CPHA			
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x04	SPI_CR2	Reserved																									TXEIE			RXNEIE	ERRIE	FRF	Reserved	SSOE	TXDMAEN	RXDMAEN
	Reset value																										0	0	0	0	0	0	0	0	0	0
0x08	SPI_SR	Reserved																									FRE		BSY	OVR	MODF	CRCERR	UDR	CHSIDE	TXE	RXNE
	Reset value																										0	0	0	0	0	0	0	0	1	0
0x0C	SPI_DR	Reserved																DR[15:0]																		
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x10	SPI_CRCPR	Reserved																CRCPOLY[15:0]																		
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1			
0x14	SPI_RXCR	Reserved																RxCRC[15:0]																		
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x18	SPI_TXCR	Reserved																TxCRC[15:0]																		
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		

- **SPI_CR1** ve **SPI_CR2** (Control Register 1 ve 2), iletişimin modunu (Master veya Slave) ve saat hızını, Data frame format, Half duplex iletişim ve diğer özellikleri yapılandırmak için kullanılır.
 - **CPHA (Clock Phase)**: SPI saat fazını belirler (ilk veya ikinci kenarda veri örneklenir).
 - **CPOL (Clock Polarity)**: SPI saatinin boşa olduğu seviyeyi belirler (yüksek veya düşük).
 - **MSTR (Master Selection)**: SPI'nin Master veya Slave modunda çalışacağını belirler.
 - **BR (Baud Rate Control)**: SPI iletişim hızını ayarlar.
 - **SPE (SPI Enable)**: SPI çevre birimini etkinleştirir veya devre dışı bırakır.
 - **LSBFIRST (Frame Format)**: Verinin MSB veya LSB ilk gönderileceğini belirler.

- **SSM (Software Slave Management)**: Slave yönetiminin yazılımla mı yapılacağını belirler.
- **SSI (Internal Slave Select)**: Dahili Slave seçim sinyalini kontrol eder.
- **DFF (Data Frame Format)**: Veri çerçevesinin 8-bit veya 16-bit olacağını ayarlar.
- **RXONLY (Receive Only Mode)**: SPI'nin sadece alım modunda çalışmasını sağlar.
- **SPI_CR2 (Control Register 2)**, SPI'nin ek kontrol ayarlarını içerir.
 - **TXEIE (TX Buffer Empty Interrupt Enable)**: Gönderme tamponu boşsa kesmeyi etkinleştirir.
 - **RXNEIE (RX Buffer Not Empty Interrupt Enable)**: Alım tamponu dolduğunda kesmeyi etkinleştirir.
 - **ERRIE (Error Interrupt Enable)**: Hata kesmelerini etkinleştirir.
 - **SSOE (SS Output Enable)**: Slave Select (NSS) çıkışını kontrol eder.
 - **FRF (Frame Format)**: SPI çerçeve formatını belirler (Motorola veya TI).
 - **TXDMAEN (Tx Buffer DMA Enable)** ve **RXDMAEN (Rx Buffer DMA Enable)**, DMA kullanarak veri gönderme ve alma işlemlerini etkinleştirir.
- **SPI_SR (Status Register)**, SPI haberleşme durumunu izlemek için kullanılır. İletimin tamamlanıp tamamlanmadığı, veri alımı durumu gibi bilgileri içerir.
 - **RXNE (Receive Buffer Not Empty)**: Veri alım tamponunun dolu olduğunu gösterir.
 - **TXE (Transmit Buffer Empty)**: Veri gönderme tamponunun boş olduğunu gösterir.
 - **CHSIDE (Channel Side)**: I2S modunda aktif olan kanalı gösterir.
 - **UDR (Underrun Flag)**: Gönderme sırasında beklenen veri gelmediğinde ortaya çıkar.
 - **CRCERR (CRC Error Flag)**: CRC (Döngüsel Artıklık Kontrolü) hata bayrağı.
 - **MODF (Mode Fault)**: Master ve Slave çakışma hatası.
 - **OVR (Overrun Flag)**: Alım sırasında veri kaybı olduğunu gösterir.
 - **BSY (Busy Flag)**: SPI'nin hala meşgul olduğunu gösterir.
- **SPI_DR (Data Register)**, Veri gönderip almak için kullanılır. Gönderilen veya alınan veriyi bu kayıt aracılığıyla işleyebilirsiniz.
 - **DR (Data Register)**: SPI üzerinden gönderilecek veya alınacak veriyi tutar.
- **SPI_CRCPR (CRC Polynomial Register)** ve **SPI_RXCRCR/SPI_TXCRCR (CRC Receive/Transmit Register)**, SPI verilerinin döngüsel hata denetimi (CRC) için kullanılır.
 - **CRCPOLY (CRC Polynomial Value)**: CRC kontrolü için kullanılan polinom değerini belirler.
 - **RX CRC (Receive CRC Value)**: SPI üzerinden alınan verinin CRC kontrol değerini saklar.
 - **TX CRC (Transmit CRC Value)**: SPI üzerinden gönderilen verinin CRC kontrol değerini saklar.

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.

Konfigürasyon Ayarları

- SPI Modülünü Etkinleştir
 - SPI çevre biriminin clock'unu etkinleştirmen gerekiyor.
- SPI için Gerekli GPIO Pinlerini Ayarla
 - GPIO pinler Alternatif Fonksiyon olarak ayarlanmalıdır.
- SPI'nin Temel Çalışma Modunu Belirle
 - SPI'nin Master veya Slave olarak çalışacağını ayarla.
 - SPI clock hızı (Baud Rate) belirlenir.
 - Clock Polarity (CPOL) ve Clock Phase (CPHA) belirlenir.
 - Veri çerçevesinin boyutunu belirle.
 - SPI'nin etkinleştirilip etkinleştirilmeyeceğini belirle
- SPI Ek Ayarlarını ve Kesme ile DMA Durumlarını Belirle
- SPI Veri Gönderme ve Alma İşlemlerini Ayarla