

09 SPI

5 Mayıs 2021 Çarşamba 08:03

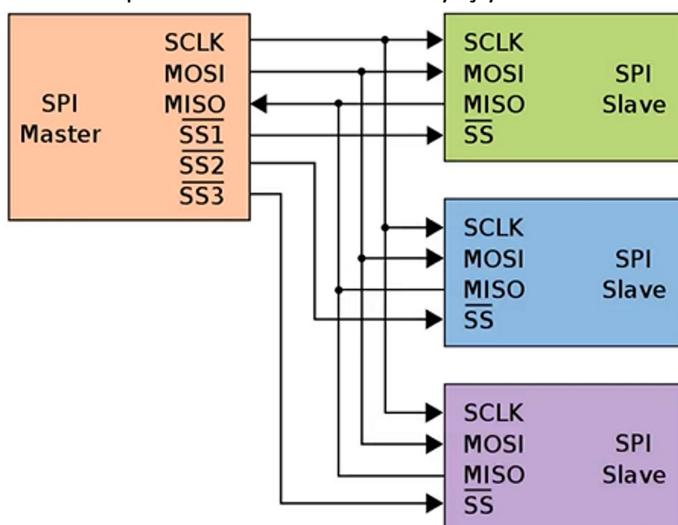
09 SPI

Giriş

- <https://ozdenercin.com/2019/02/01/spi-seri-haberlesme-protokolu/> , <https://arduinodestek.com/spi-haberlesme-protokolu-nedir-ve-nasıl-gerçekleşir/> <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 bir şekilde çalışmasıyla data iletişimini sağlamaktadır.
- SPI'da veri transfer hızı I2C veri yolundan daha hızlıdır.
- SPI, genellikle düşük maliyetli, düşük güç tüketimi gerektiren uygulamalarda kullanılır. Özellikle mikrodenetleyiciler, sensörler, veri dönüştürücüler, hafıza kartları ve diğer entegre devreler arasında veri iletişimini için tercih edilir.
- SPI'nin hızlı ve basit bir iletişim protokolü olması, çeşitli uygulama alanlarında yaygın olarak kullanılmasını sağlar.

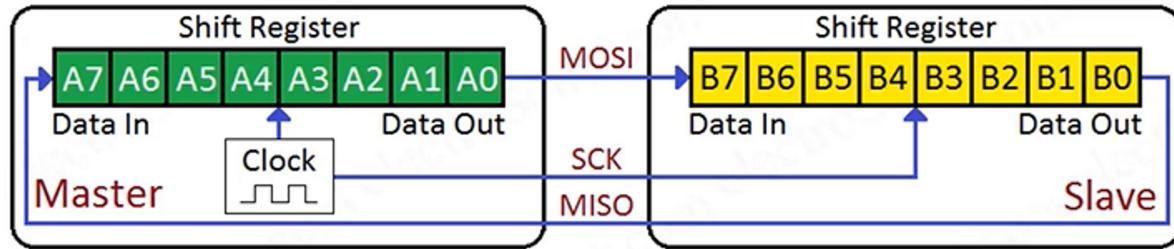
Bağlantılar

- SPI, genellikle dört telli bir bağlantıyla gerçekleştirilir:
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 ve CS pini kullanır.
- Tüm pinlerin kullanılmasına ihtiyaç yoktur.

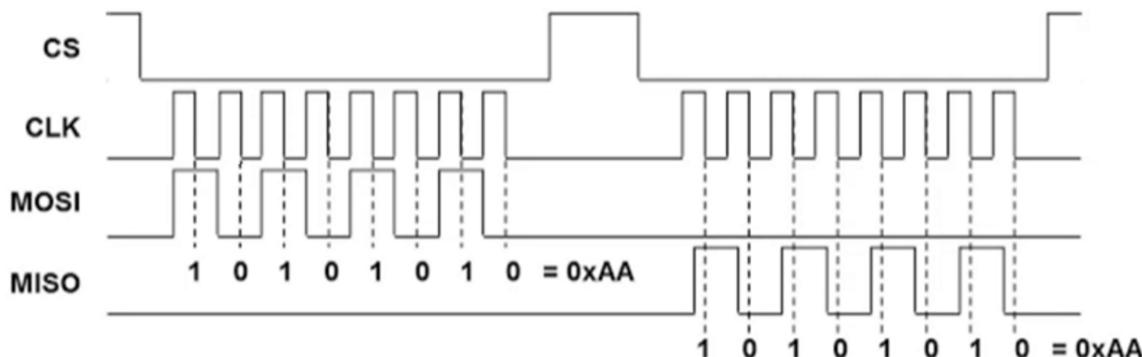


Veri İletimi

- SPI iletişiminde, önce çalışılmak istenen slave cihazın bağlı olduğu SS/CS pini seçilir.
- Master tarafından verinin en öncelikli (MSB) kısmından itibaren MOSI hattı üzerinden slave cihazına her clock pulsusunda bir bit olmak üzere tüm veri gönderilir.



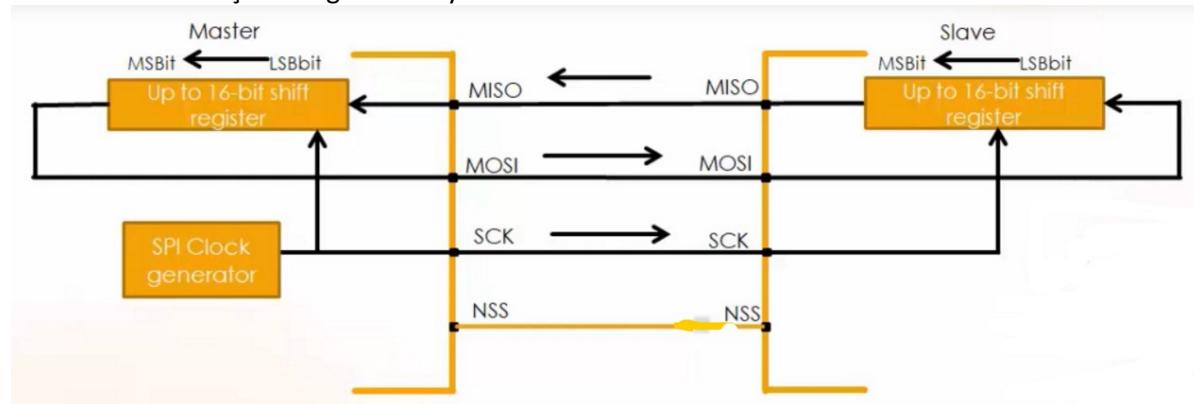
- Master, saat sinyalini verir.
- Master, SS/CS pinini, köleyi etkinleştirilen bir düşük voltaj durumuna geçirir:
- Master, verileri MOSI hattı boyunca her seferinde bir bit olarak slave'ye gönderir. Slave, bitleri alındıkça okur.
- Bir yanıt gerekiyorsa, bağımlı, MISO hattı boyunca her seferinde bir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



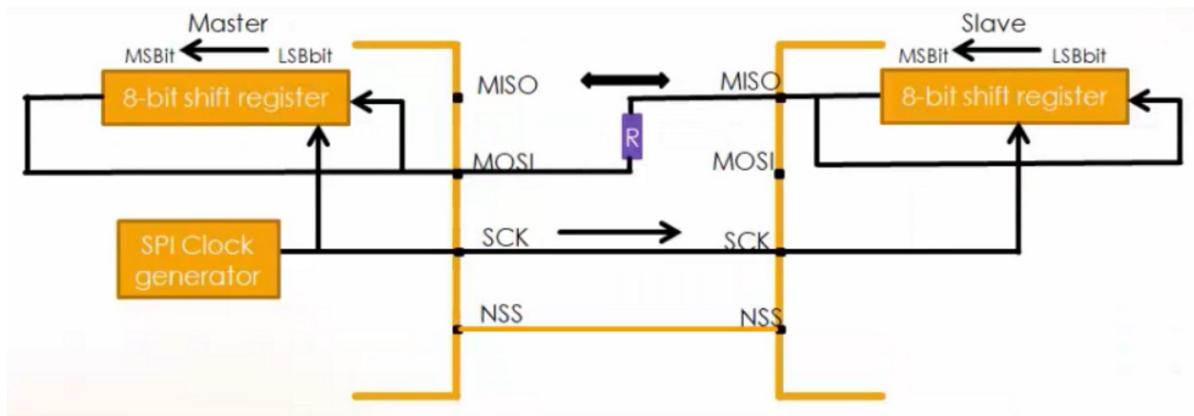
- SPI'da veri iletim sırası genellikle 8 bitlik veri paketleri halinde olur, ancak bu uzunluğu değiştirilebilir. Veri iletim sırası, verilerin en yüksek veya en düşük anlamlı bit ile başlayıp bitişebileceği şekilde yapılandırılabilir.
- Ana cihaz, iletişimi başlatır ve sonlandırır. Slave cihazlar ise ana cihazın taleplerine yanıt verir.

Veri Yolu

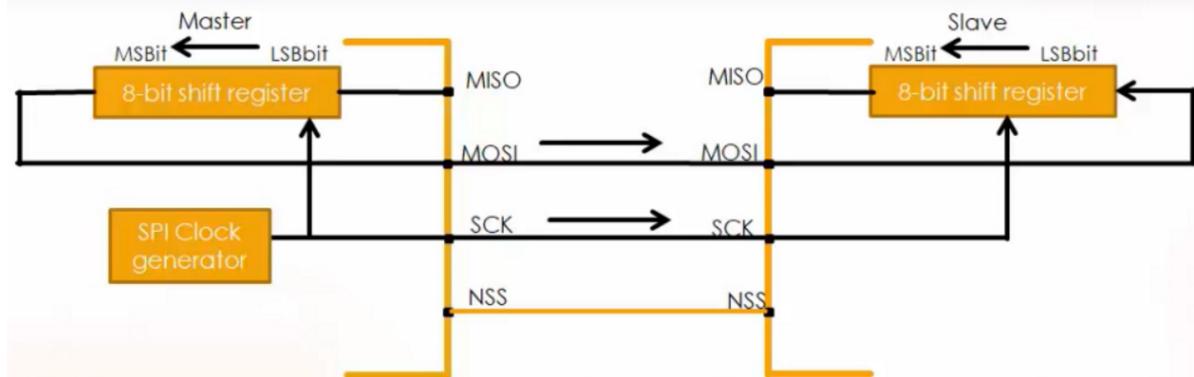
- SPI haberleşmesi için üç farklı mod vardır. Bunlar Full Duplex, Half duplex ve Simplex'dır.
- İki çift yönlü iken diğer tek yönlü haberleşmedir. Bu modlar hakkında detaylı bilgi almak için <https://fastbitlab.com/spi-bus-configuration-discussion-full-duplex-half-duplex-simplex/> linkteki yazımı okuyabiliriz.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

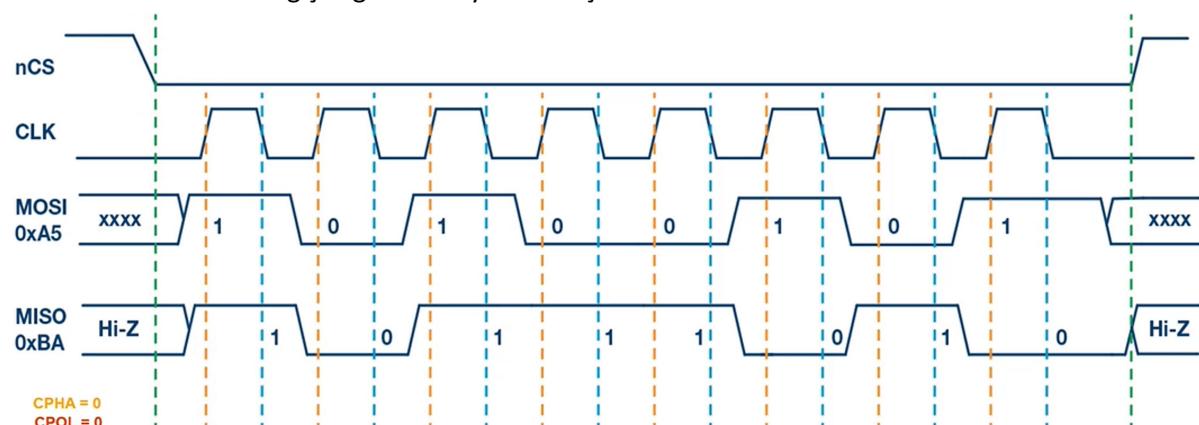


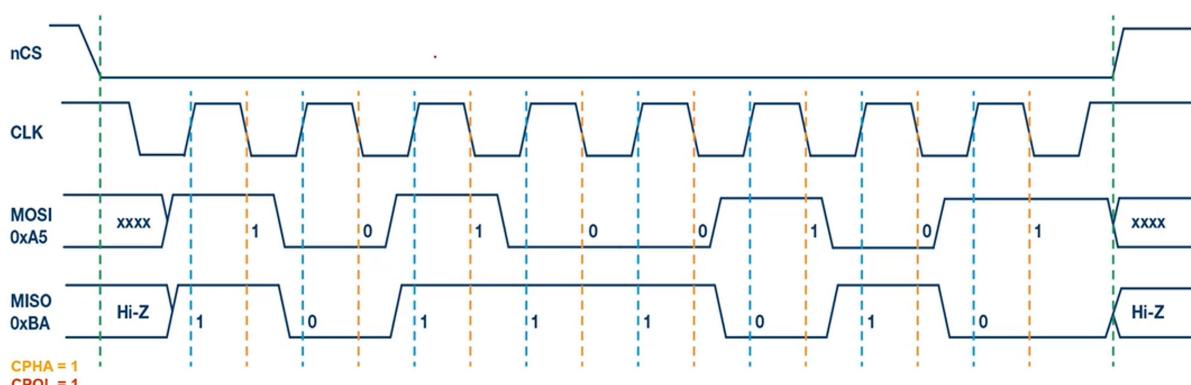
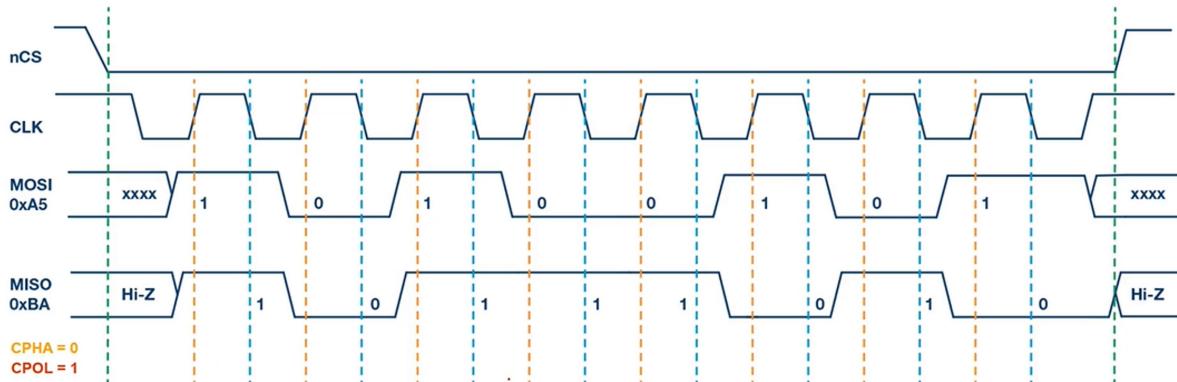
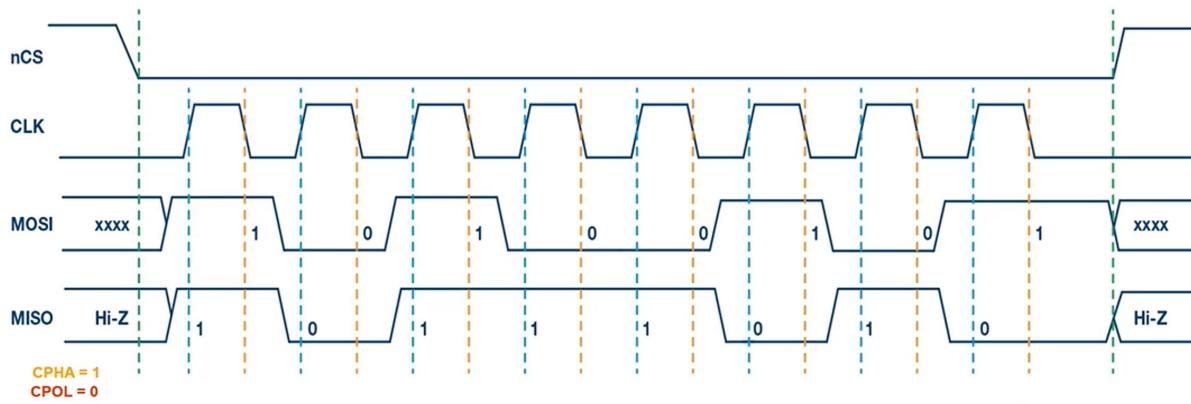
- **Simplex**, sadece bir yönde veri iletimine izin veren bir çalışma modudur. Genellikle tek bir iletişim hattı kullanılır ve veri gönderme veya veri alma işlemi yapılır. Her iki tarafta aynı anda veri gönderip alamaz.



Çalışma Modları

- SPI'nin çalışma modları, saat sinyalinin hangi kenarda değişeceğini (CPHA) ve veri iletiminin hangi kenarda başlayacağını (CPOL) belirler.
- **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.





Avantajları ve Dezavantajları

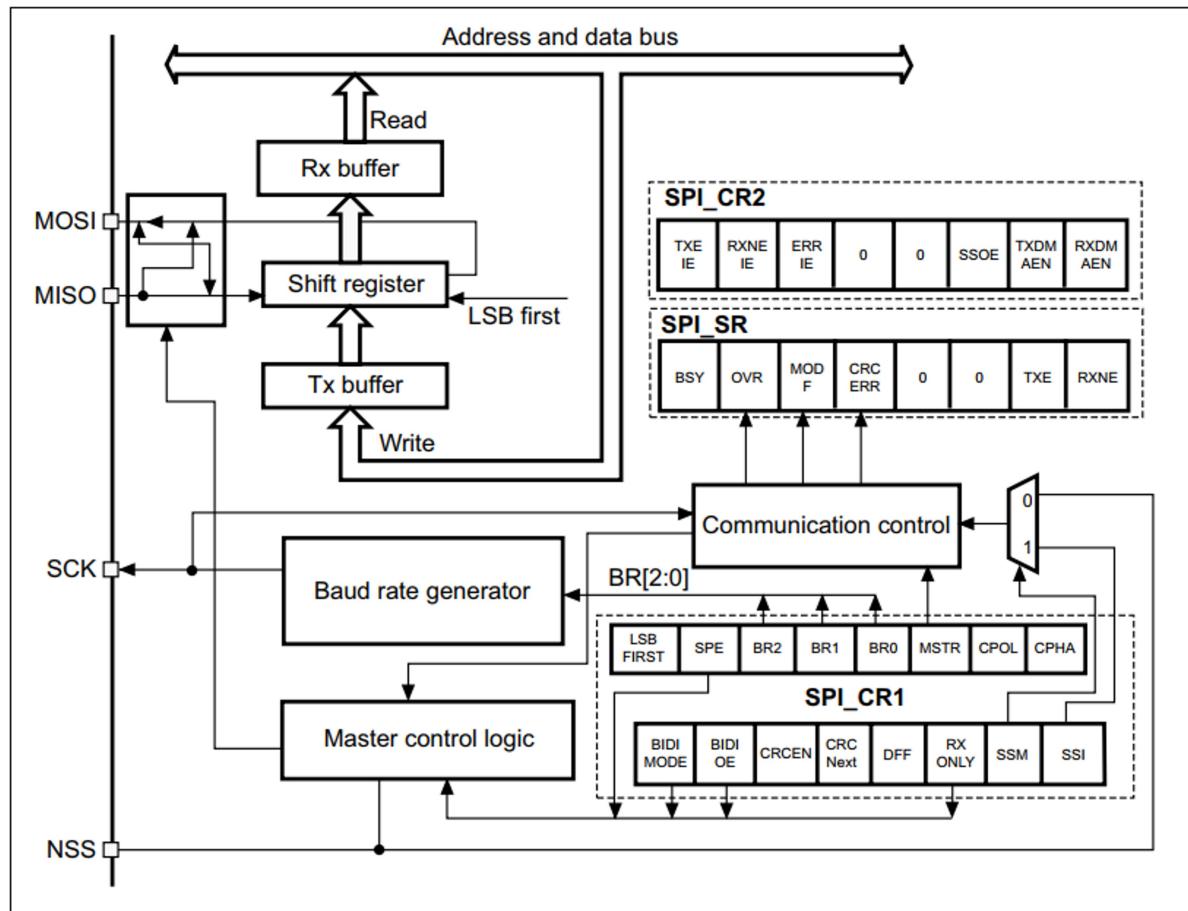
Avantajlar;

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

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 | | | | | | | | | | BIDIMODE | 15 | 14 | CRCEN | CRCNEXT | DFF | RXONLY | SSM | SSI | BR [2:0] | | BIDIMODE | BIDIOE | CRCEN | MSTR | CPOL | CPHA | LSBF/RST | 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 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | |
| 0x04 | SPI_CR2 | Reserved | | | | | | | | | | TXEIE | 15 | RXNEIE | 14 | ERRIE | 13 | FRF | BSY | OVR | MODF | CRCERR | UDR | Reserved | SSOE | TXDMAEN | 5 | 4 | 3 | 2 | 1 | 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 | | | |
| 0x08 | SPI_SR | Reserved | | | | | | | | | | FRE | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 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 | | | |
| 0x0C | SPI_DR | Reserved | | | | | | | | | | DR[15:0] | | | | | | | | | | 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 | | | |
| 0x10 | SPI_CRCPR | Reserved | | | | | | | | | | CRCPOLY[15:0] | | | | | | | | | | 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 | | | |
| 0x14 | SPI_RXCRCR | Reserved | | | | | | | | | | RxCRC[15:0] | | | | | | | | | | 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 | | | |
| 0x18 | SPI_TXCRCR | Reserved | | | | | | | | | | TxCRC[15:0] | | | | | | | | | | 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 | | | |

- **SPI_CR1** (Control Register 1), SPI haberci modunu (Master veya Slave) ve saat hızını belirlemek için kullanılır.
- **SPI_CR2** (Control Register 2), Data frame format, yarı-dubleks iletişim ve diğer özelliklerini yapılandırmak için kullanılır.
- **SPI_SR** (Status Register), SPI haberleşme durumunu izlemek için kullanılır.
- **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 yapılabılır.
- <https://deepbluembedded.com/stm32-spi-tutorial/> linkinden konu hakkındaki bilgileri inceleyebiliriz.