

BMS 302 MİKROKONTROLCÜLER

SERİ HABERLEŞME

Prof.Dr. Mutlu AVCI

Çukurova Üniversitesi Biyomedikal Mühendisliği Bölümü

Bahar 2020

- 
- *SPI (Serial Peripheral Interface)* seri iletişimi az sayıda uç ile sağlayan bir seri iletişim metodudur. SPI iletişimi master-slave (usta-köle) iletişim prensibine göre çalışır. Master cihaz gerekli clock (saat) sinyalini sağlar. SPI iletişimde aynı anda sadece bir master cihaz aktif olabilir.

- SPI iletişimde toplam 4 uç kullanılır. Bu uçlar şunlardır :

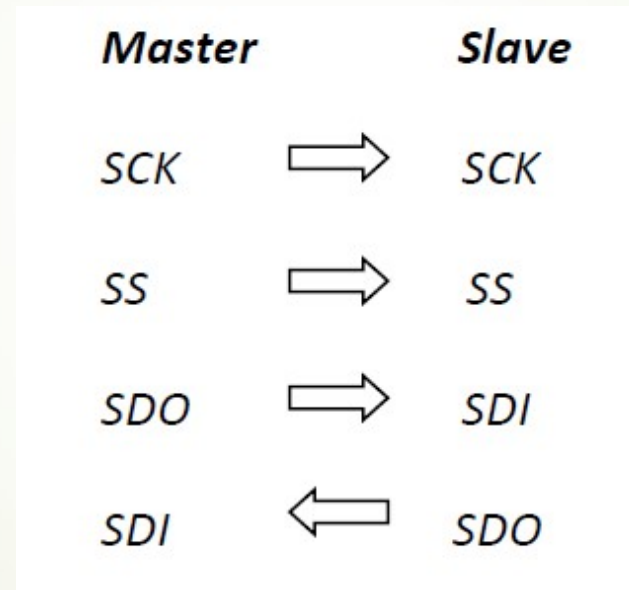
SDO/MISO = Serial Data Output / Master Input Slave Output – Seri Data Çıkış Ucu

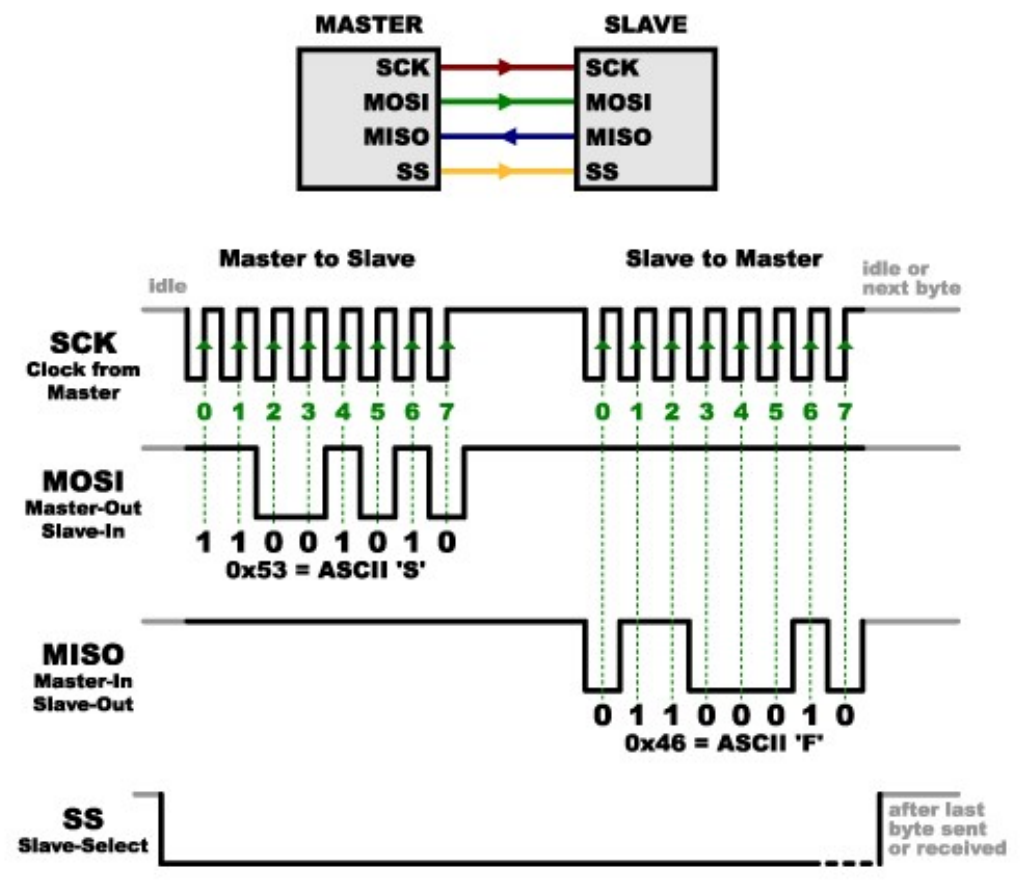
SDI/MOSI = Serial Data In / Master Output Slave Input – Seri Data Giriş Ucu

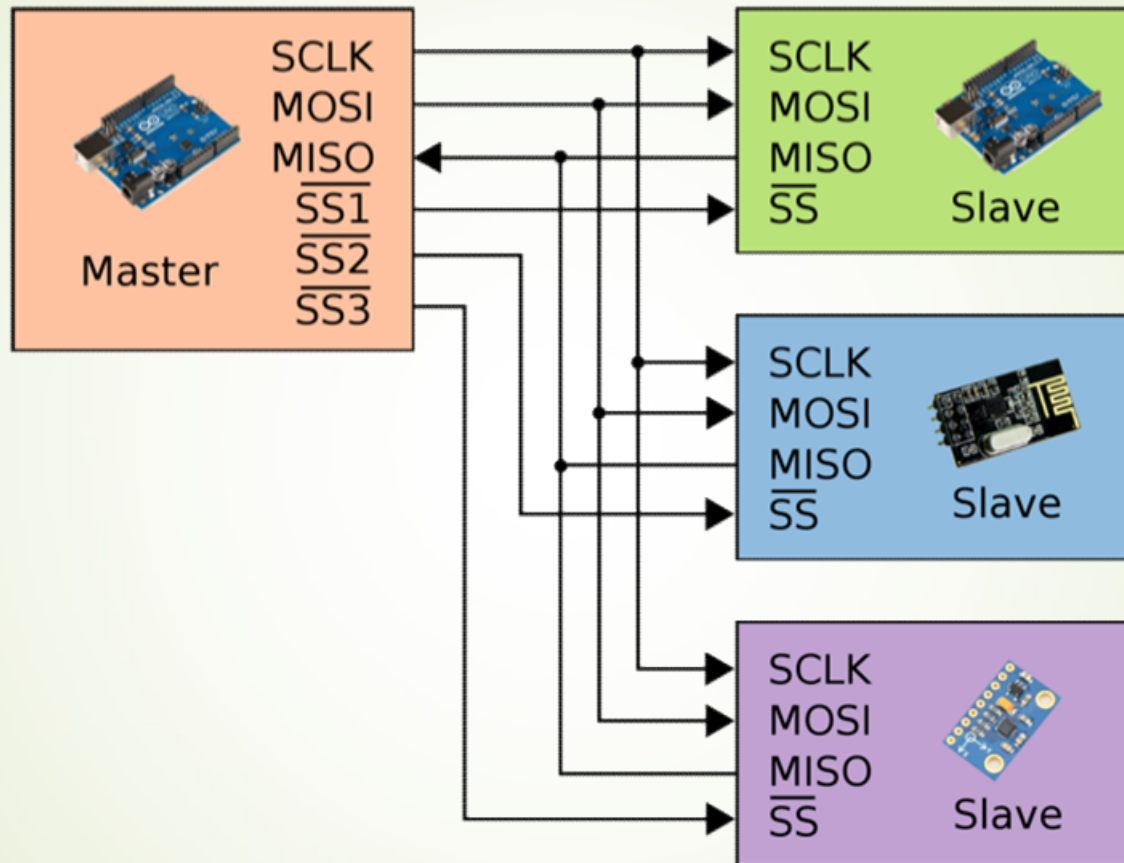
SCK = Serial Clock – Seri Saat Sinyali Ucu

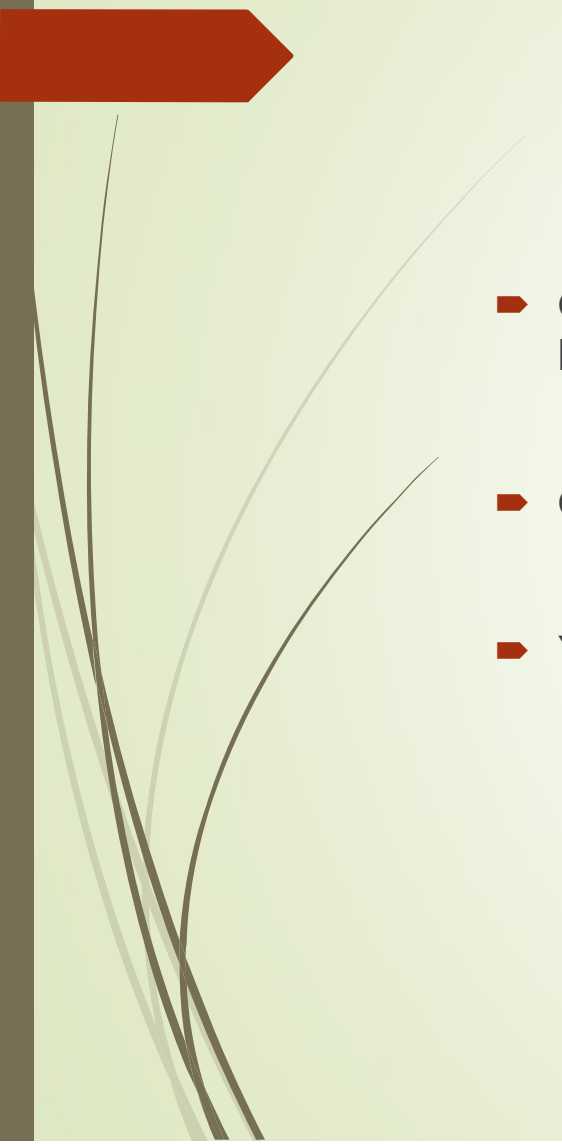
SS/CS = Slave Select Pin / Chip Select – Slave Cihazı Seçme Ucu

- Tek bir master cihaz ile tek bir slave cihaz arasındaki bağlantılar aşağıda verilmiştir.





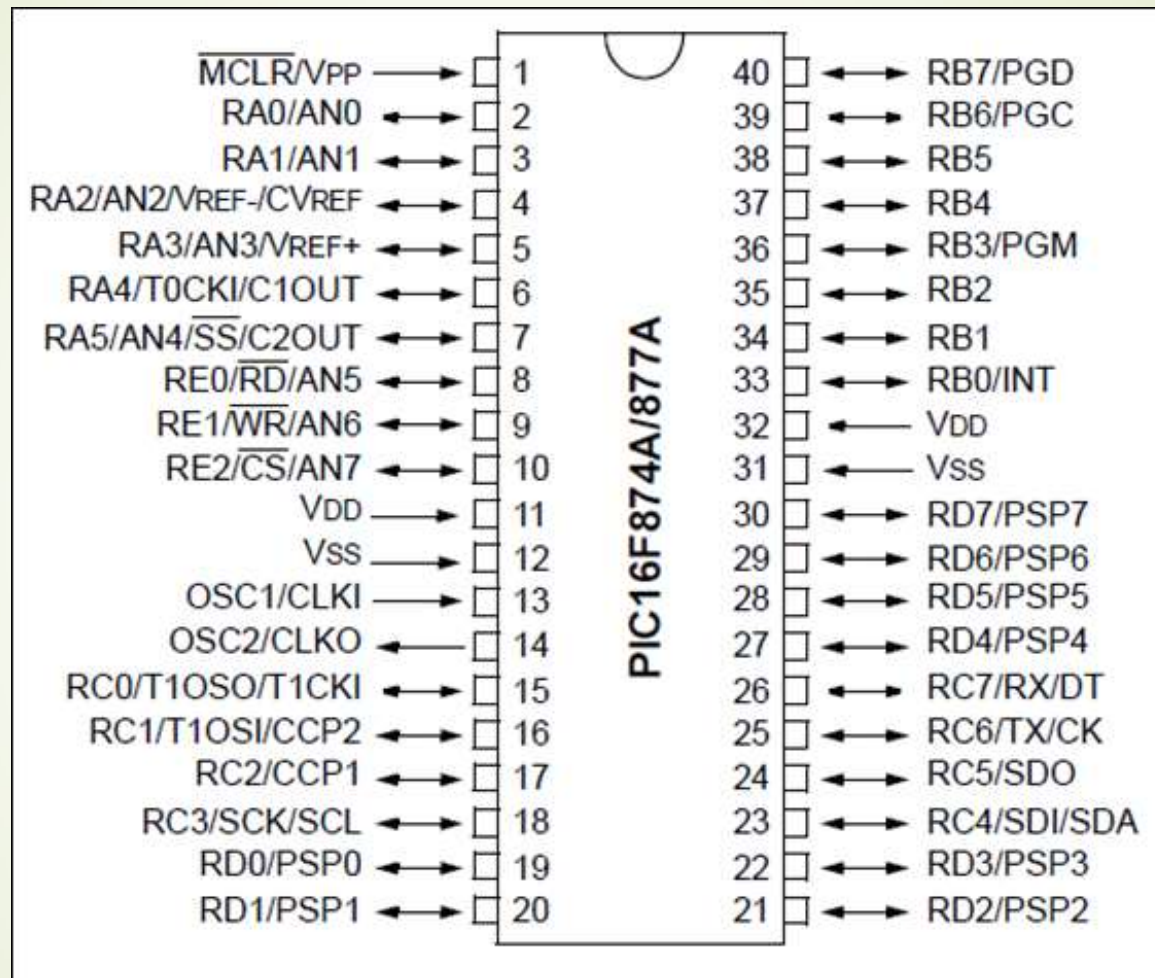


- 
- CCS C derleyici içinde SPI iletişimi kontrol etmemizi sağlayan fonksiyonlar bulunmaktadır.
 - CCS C programı hem donanımsal hem de yazılımsal SPI iletişimini destekler.
 - Yazılımsal SPI donanımsal SPI iletişime göre daha yavaştır.

➤ **SETUP_SPI(), SPI_READ (), SPI_WRITE(), SPI_DATA_IS_IN (), SPI_XFER()**

Bu fonksiyonlar ile CCS C'de SPI iletişimi sağlanabilmektedir. Bu komutların kullanılabilmesi için kontrolcü içinde SPI donanım biriminin bulunması gereklidir(SPI_XFER () komutu hariç).

Bu komutlar kullanılan kontrolcünün SPI donanım biriminin standart uçlarını kullanır. Bu uçlar değiştirilemez. Fakat istenilen uçlar ile SPI iletişimi yapmak için CCS C derleyicisinin yazılımsal SPI iletişimi kullanılabilir.



► **SETUP_SPI()**Fonksiyonu

Bu fonksiyon SPI donanım biriminin iletişim ayarlarını yapmaya yarar. Kullanımı ;

► **setup_spi (mod) ;**

şeklindedir. Fonksiyonun “mod” kısmına aşağıda belirtilen sabit tanımlamalardan biri veya birden fazlası yazılabilir. İki sabit arasına ‘ | ’ operatörü konulur.

SPI_MASTER // SPI Master modunda, SCK Clock sinyali dışarıya verilir

SPI_SLAVE // SPI Slave modunda, SCK Clock sinyali dışarıdan alınır

SPI_L_TO_H // SCK sinyalinin yükselen kenarında veri gönderilir

SPI_H_TO_L // SCK sinyalinin düşen kenarında veri gönderilir

SPI_SS_DISABLED // Slave Modunda SS ucunun kullanılmayacağını belirtir

- 
- Aşağıdaki tanımlamalar master mod seçeneğinde kullanılır

SPI_CLK_DIV_4 // SCK Sinyalini 4'e böler

SPI_CLK_DIV_16 // SCK Sinyalini 16'ya böler

SPI_CLK_DIV_64 // SCK Sinyalini 64'e böler

SPI_CLK_T2 // SCK frekansı olarak Timer2 çıkış frekansı kullanılır

➤ Fonksiyonun kullanımı ;

`setup_spi(spi_master | spi_L_to_H | SPI_CLK_DIV_4) ;`

şeklindedir. Yukarıdaki örnek ile kontrolcünün master modunda çalışacağı, SCK sinyalinin düşen kenarında veri gönderileceği ve SCK sinyalinin 4'e bölüneceği belirtilmiştir.



➤ ***SPI_WRITE () Fonksiyonu***

SPI biriminden bir byte gönderme işlemi yapar. Kullanımı ;

```
spi_write(data);
```

şeklindedir. Fonksiyondaki '*data*' kısmı 8 bitliktir



► ***SPI_READ () Fonksiyonu***

SPI biriminden bilgi okumaya yarayan fonksiyondur. Eğer girişte bilgi (data) varsa o bilgiyi okur ve o değerle fonksiyon geri döner, eğer girişte bilgi yoksa bilgi gelene kadar bekler.

Fonksiyonda ayrıca parantez içine 8 bitlik bir veri değeri yazılabilir. Bu şekilde kullanıldığında fonksiyon ilk başta parantez içine yazılan veriyi gönderir, ardından bir veri gelmesini bekler. Gelen veriyi okur ve o değerle geri döner.

➤ Kullanımı ;

değişken ismi = spi_read () ;

değişken ismi = spi_read (data) ;

data =spi_read() ;

data = spi_read (0x08) ;

şeklindedir.



► ***SPI_DATA_IS_IN() Fonksiyonu***

SPI biriminden herhangi bir bilgi (data) alındığında fonksiyon 1 (TRUE) değeri ile geri döner. SPI biriminden herhangi bir bilgi alınmadığında 0 (FALSE) değeri ile geri döner. Kullanımı ;

Değişken adı = spi_data_is_in() ;

y = spi_data_is_in () ;

şeklindedir.



► ***SPI_XFER()*** Fonksiyonu

Bu fonksiyon SPI iletişim protokolü ile yazma ve/veya okuma işlemleri yapmaya yarar. Yukarıda anlatılan okuma veya yazma fonksiyonlarının tümü sadece donanımsal SPIN iletişimi içindir. *spi_xfer()* fonksiyonu sayesinde yazılımsal SPI iletişimi için okuma ve yazma işlemleri sağlanabilir.

`spi_xfer(veri veya degisken);`

`degisken=spi_xfer();`

`Degisken=spi_xfer(veri);`



► **#USE SPI() Fonksiyonu**

Eğer donanımsal SPI birimi ile iletişim yapmak değil de, yazılımsal olarak SPI iletişimi yapmak istiyorsanız bu fonksiyonu kullanmanız gerekir. Bu fonksiyon ile içinde SPI donanım birimi olan denetleyiciler ve içinde SPI birimi bulunmayan denetleyiciler için yazılımsal olarak SPI iletişimi sağlanabilir.

Sabitler:

MASTER / SLAVE : SPI Master / Slave modu ayarı

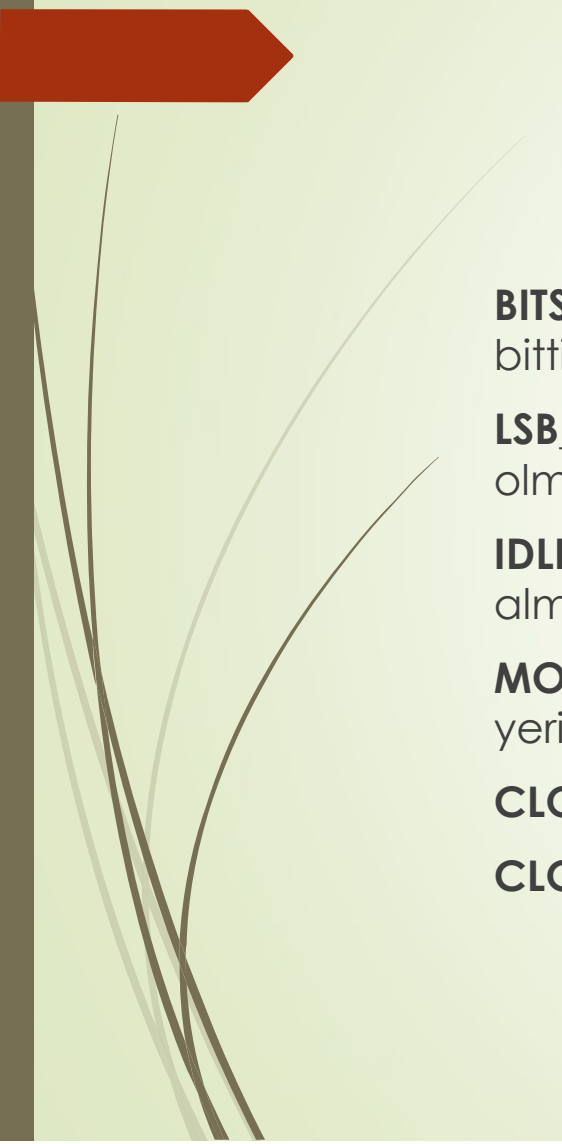
DI = pin : MOSI / SDI ucu belirleme (DI=pin_B0)

DO = pin : MISO / SDO ucu belirleme (DO=pin_B2)

CLK = pin : SCK ucunu belirler (CLK=pin_B3)

BAUD = hız : saniyedeki bit sayısı (bit per second – bps) cinsinden haberleşme hız ayarı için kullanılır.

SAMPLE_RISE / SAMPLE_FALL : Bilgiler clock sinyalinin YÜKSELEN/DÜŞEN kenarında gönderilir.



BITS=n : gönderilecek veya alınacak maksimum bit sayısını belirler (ön tanımlı 16 bittir)

LSB_FIRST / MSB_FIRST : İlk gönderilecek bitin DÜŞÜK_DEĞERLİ / YÜKSEK_DEĞERLİ olması durumunu belirler.

IDLE=n : Clock ucunun boşta bekleme sırasında lojik 0 veya lojik 1 değeri almasını ayarlar. Burada n, 0 veya 1 olur.

MODE=n : SPI iletişimde bulunan 4 moddan birinin kullanılması istenirse n yerine 0,1,2,3 değerlerinden biri yazılarak kullanılır.

CLOCK_HIGH=n : Clock sinyalinin lojik 1 de kalma süresini belirtir.

CLOCK_LOW=n : Clock sinyalinin lojik 0 da kalma süresini belirtir.



FORCE_HW : Donanımsal SPI modülünün kullanılmasını sağlar.

STREAM=isim : haberleşme ayarlarının isim ile kaydedilmesini böylece fazla sayıda farklı haberleşme olaylarının farklı isimlerle çağırılıp kullanılabilmesini sağlar.

ENABLE=pin_ismi : her veri paketi gönderimi ve alımı boyunca istenilen bir pin aktif/pasif olur. İşlem bitince durum değiştirir.

ENABLE_ACTIVE=n : enable pininin aktif durumdaki değerini belirtir. Burada n 0 veya 1 olur.



LOAD=pin_ismi : Veri paketindeki her byte (8-bit) veri gönderimi için uç aktif olur sonra pasife döner.

LOAD_ACTIVE=n : Load ucunun aktif durumdaki değerini belirtir. (0 veya 1)

DIAGNOSTIC=pin_ismi : Her bir bit veri gönderiminde aktif olur.

SPI1 / SPI2 : Fazla sayıda SPI modülü içerilmesi durumunda aralarında seçim yapmayı sağlar.



■ SPI Kesmesi

SPI iletişim için #INT_SSP kesmesi kullanılır. SPI iletişimde okuma veya yazma işlemi yapıldığında kesme meydana getirilebilir. Sadece donanımsal SPI iletişimde bu kesme aktif olur.

