

# Mikroişlemcili Sistemler Ders Sunumları

## SPI (Serial Peripheral Interface)

Hazırlayan: Arş. Gör. Hakan ÜÇGÜN

## Seri Haberleşme Teknolojileri

- Dijital sistemlerde kablolu seri haberleşme ile ilgili birçok standart vardır. SPI, I<sup>2</sup>C, USART, USB bunlara örnek olarak verilebilir.
- Bu standartların **kullandıkları uç sayısı, ulaşabilecekleri maksimum hızlar** birbirinden farklı olmakla beraber I<sup>2</sup>C protokolü oldukça hızlı veri aktarımına sahiptir.
- Bir arada çalışan, belirli aralıklarla birbiriyle haberleşen, yavaş çeşitli çevresel cihazların minimum harici donanım gereksinimiyle haberleşmelerini sağlar.

Mikroişlemcili Sistemler Dersi 6.12.2018

{ 2 }

## SPI Nedir?

- **SPI (Serial Peripheral Interface)** *Motorola* firması tarafından kendi mikro denetleyicileri için geliştirmiş, sonrasında ise bir standart olarak benimsenerek diğer mikro denetleyici üreten firmalar tarafından da kullanılmaya başlanmıştır.
- *Seri çevresel arayüz* anlamına gelen SPI, isminde de anlaşılacağı üzere bir **seri iletişim protokolüdür**.
- SPI, full duplex(eş zamanlı çift yönlü çalışabilen), senkron(datanın saat darbeleriyle birlikte eşzamanlı olarak aktarıldığı) bir seri haberleşme standardıdır ve pek çok tümleşik devre(IC) tarafından donanımsal olarak desteklenmektedir.

Mikroişlemcili Sistemler Dersi 6.12.2018

{ 3 }

## SPI İletişimi

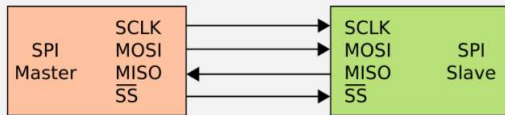
- SPI haberleşmesinde veri(data, bitler) transferi master - slave ilişkisi ile gerçekleşir.
- Master cihaz Slave cihazları yönetmek ve onlara işlemler yaptırmak ile yükümlüdür.
- Master veri haberleşmesini başlatan cihazdır.
- Master tarafından veri transferi başlatıldıktan sonra veri her iki yönde de eşzamanlı olarak aktarılabilir. Alınan baytın anlamlı olup olmadığı bilgiyi alan cihaza kalmıştır.
- Slave cihaz ise Master cihazdan gelen bilgilere göre işlem yapmak ile yükümlüdür.

Mikroişlemcili Sistemler Dersi 6.12.2018

{ 4 }

## SPI İletişimi

- Bir master cihaza birden çok slave cihaz bağlanabilir. Ama aynı anda Slave cihazlardan sadece bir tanesi aktif olarak Master cihaz ile bilgi alışverişi gerçekleştirir.
- Aynı anda veri alıp gönderebilen ( senkron haberleşme ) sistemlere full duplex sistemler denir.
- Spi protokolü de full duplex bir haberleşme protokolüdür. Yani Master mikro denetleyici spi haberleşmesini başlattığı anda hem bilgi gönderir, hem de bilgi alır. Aynı durum Slave mikro denetleyici için de geçerlidir.



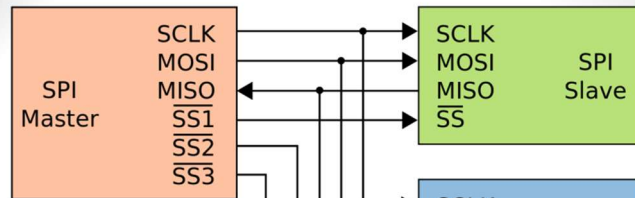
Mikroİşlemci Sistemler Dersi | 6.12.2018

5

- **SCLK: Serial Clock:** Saat darbesi verilerin lojik olarak iletilmesini sağlar. **Slave** cihaz seçildikten sonra haberleşmenin başlayabilmesi için saat darbesine ihtiyaç vardır. Saat darbesi **Master** mikrodenetleyici tarafından sağlanır ve SCLK üzerinden **Slave** mikrodenetleyiciye aktarılır.
- **MOSI: Master Output Slave Input:** **Master** cihazdan **Slave** cihaza, MOSI pini üzerinden bilgiler aktarılır. Bu bilgilerin aktarılması için **Master** mikrodenetleyici SS pinini lojik sıfır seviyesine çekmeli ve SCLK üzerinden saat darbesi üretmelidir.
- **MISO: Master Input Slave Output:** **Slave** cihazdan **Master** cihaza MISO pini üzerinden bilgiler gönderilir. Bu bilgilerin aktarılması için **Master** mikrodenetleyici SS pinini lojik sıfır seviyesine çekmeli ve SCLK üzerinden saat darbesi üretmelidir.
- **SS/CS: Slave Slect/Chip Slect:** Spi'da **Master** cihazdan **Slave** cihaza bilgi aktarabilmek için **Slave** cihazın yetkilendirilmesi gerekmektedir. **Slave** cihazın yetkilendirilmesi için SS pininin lojik sıfır seviyesine çekilmesi gerekir. **Master** mikro denetleyicideki **Slave** Select pinin sayısı **Slave** cihaz sayısına eşittir. Birden fazla **Slave** cihaz varsa, bilgi gönderilmek istenen **Slave** cihaz kendine bağlı olan SS pini ile seçilir ve haberleşme yapılır. Böylelikle gönderilen bilgiler sadece haberleşmek istenen cihaza iletilir.

Mikroİşlemci Sistemler Dersi | 6.12.2018

6



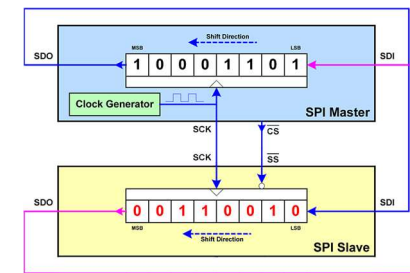
- Resimde Çoklu Spi Haberleşmesi gösterilmektedir.
- Hangi mikro denetleyici ile haberleşileceği master cihaz tarafından belirlenir.

Mikroİşlemci Sistemler Dersi | 6.12.2018

7

## SPI Çalışması

- SPI iletişimde önce çalışacak olan slave cihaz SS pini ile Seçilir. Master tarafından verinin en değerlikli kısmından(MSB) itibaren SDO (MISO) hattı üzerinden slave tarafına her clock palsinde bir bit olarak gönderilir.

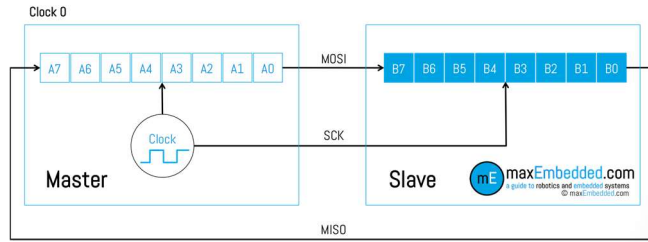


SPI Master to Slave interaction

Mikroİşlemci Sistemler Dersi | 6.12.2018

8

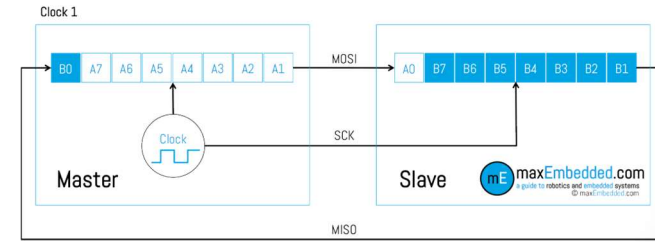
## SPI Çalışması



Mikroişlemci Sistemler Dersi 6.12.2018

9

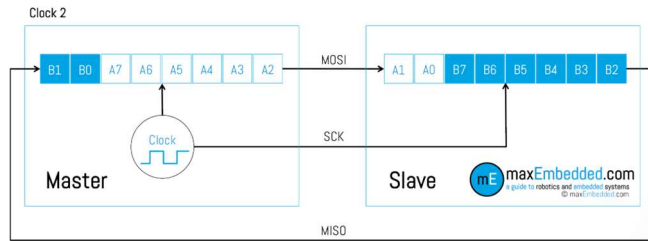
## SPI Çalışması



Mikroişlemci Sistemler Dersi 6.12.2018

10

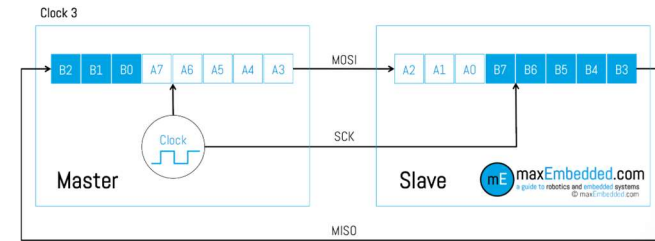
## SPI Çalışması



Mikroişlemci Sistemler Dersi 6.12.2018

11

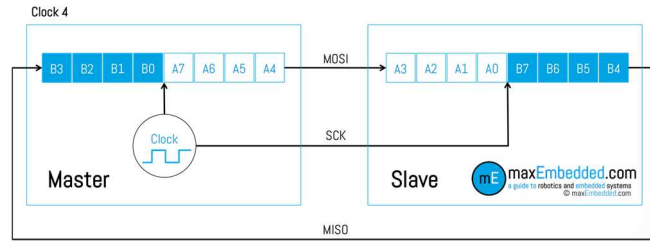
## SPI Çalışması



Mikroişlemci Sistemler Dersi 6.12.2018

12

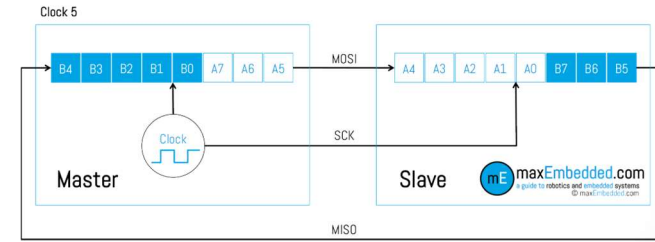
## SPI Çalışması



13

Mikroİşlemcili Sistemler Dersi | 6.12.2018

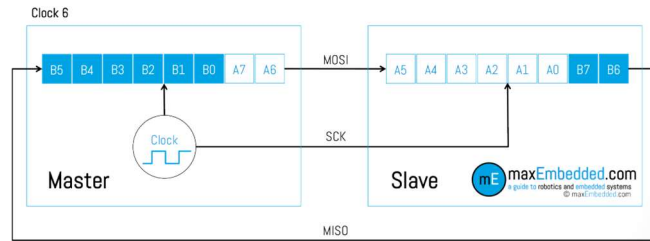
## SPI Çalışması



14

Mikroİşlemcili Sistemler Dersi | 6.12.2018

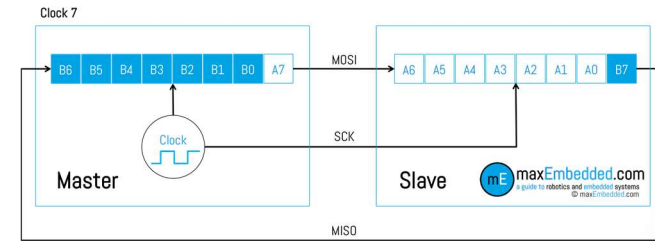
## SPI Çalışması



15

Mikroİşlemcili Sistemler Dersi | 6.12.2018

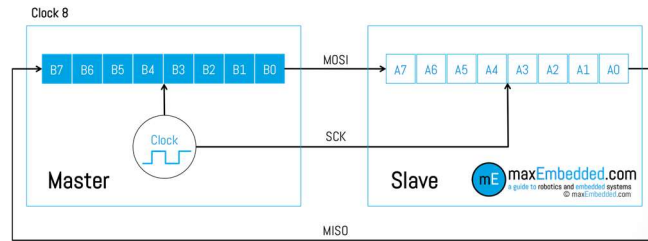
## SPI Çalışması



16

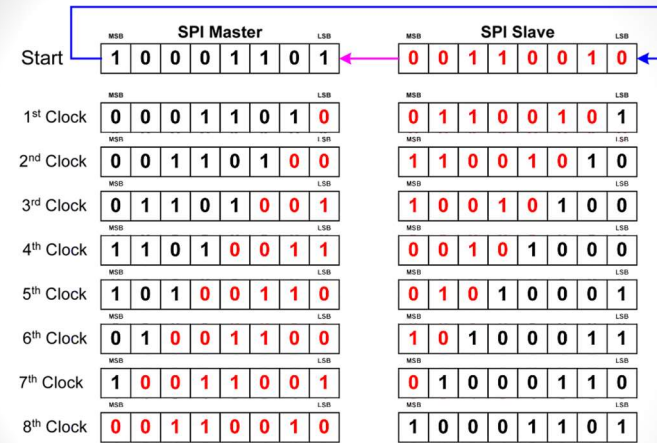
Mikroİşlemcili Sistemler Dersi | 6.12.2018

## SPI Çalışması



Mikroİşlemci Sistemler Dersi 6.12.2018

17



SPI Master and Slave data transfer

Mikroİşlemci Sistemler Dersi 6.12.2018

18

## MikroC SPI Fonksiyonları

### SPIx\_Init

- SPI1\_Init();

### SPIx\_Init\_Advanced

- SPIx\_Init\_Advanced(unsigned short master\_slav, unsigned short data\_sample, unsigned short clock\_idle, unsigned short transmit\_edge);

### SPIx\_Read

- short take, buffer;
- ...
- take = SPI1\_Read(buffer);

### SPIx\_Write

- char buffer;
- ...
- SPI1\_Write(buffer);

Mikroİşlemci Sistemler Dersi 6.12.2018

19

## MikroC SPI Fonksiyonları

### SPI\_Set\_Active

- SPI\_Set\_Active(&SPI2\_Read, &SPI2\_Write);

### SPI\_Read

- short take, buffer;
- ...
- take = SPI\_Read(buffer);

### SPI\_Write

- char buffer;
- ...
- SPI\_Write(buffer);

Mikroİşlemci Sistemler Dersi 6.12.2018

20

## MikroC SPI Örnekleri

```
// DAC module connections
sbit Chip_Select at RCO_bit;
sbit Chip_Select_Direction at TRISCO_bit;
// End DAC module connections
```

```
unsigned int value;
```

```
void InitMain() {
    TRISA0_bit = 1;          // Set RA0 pin as input
    TRISA1_bit = 1;          // Set RA1 pin as input
    Chip_Select = 1;         // Deselect DAC
    Chip_Select_Direction = 0; // Set CS# pin as Output
    SPI1_Init();             // Initialize SPI module
}
```

21

Mikroİlemlenli Sistemler Dersi 6.12.2018

```
// DAC increments (0..4095) -> output voltage (0..Vref)
void DAC_Output(unsigned int valueDAC) {
    char temp;
    Chip_Select = 0;          // Select DAC chip
    // Send High Byte
    temp = (valueDAC >> 8) & 0x0F; // Store valueDAC[11..8] to temp[3..0]
    temp |= 0x30;             // Define DAC setting, see MCP4921 datasheet
    SPI1_Write(temp);         // Send high byte via SPI
    // Send Low Byte
    temp = valueDAC;          // Store valueDAC[7..0] to temp[7..0]
    SPI1_Write(temp);         // Send low byte via SPI
    Chip_Select = 1;         // Deselect DAC chip
}

void main() {
    ANSEL = 0;
    ANSELH = 0;
    InitMain();              // Perform main initialization

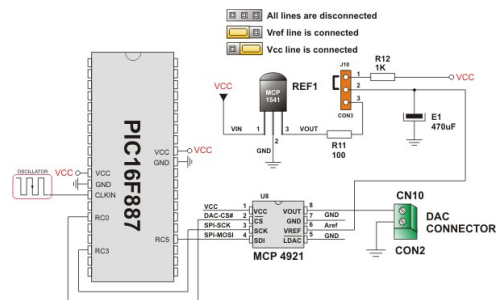
    value = 2048;            // When program starts, DAC gives

    while (1) {              // Endless loop
        if ((RA0_bit) && (value < 4095)) { // If RA0 button is pressed
            value++;          // increment value
        }
        else {
            if ((RA1_bit) && (value > 0)) { // If RA1 button is pressed
                value--;      // decrement value
            }
        }
        DAC_Output(value);    // Send value to DAC chip
        Delay_ms(1);          // Slow down key repeat pace
    }
}
```

22

Mikroİlemlenli Sistemler Dersi 6.12.2018

## Donanımsal Bağlantı Şeması



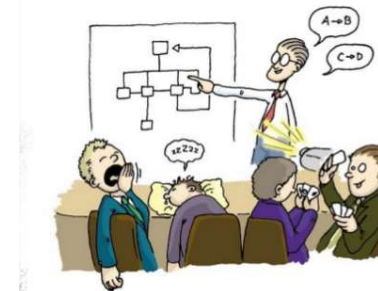
Kaynak: MicroC Pro for PIC

23

Mikroİlemlenli Sistemler Dersi 6.12.2018

## SORULARINIZ ?

### Geri Bildirim ve Tavsiyeleriniz...



24

Mikroİlemlenli Sistemler Dersi 6.12.2018