

Mikroişlemcili Sistemler Ders Sunumları

UART (Universal Asynchronous Receiver-Transmitter)

Hazırlayan: Arş. Gör. Hakan ÜÇGÜN

Seri Haberleşme Teknolojileri

- Dijital sistemlerde kablolu seri haberleşme ile ilgili birçok standart vardır. SPI, I²C, USART, USB bunlara örnek olarak verilebilir.
- Bu standartların **kullandıkları uç sayısı, ulaşabilecekleri maksimum hızlar** birbirinden farklı olmakla beraber I²C 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 14.12.2018

{ 2 }

UART Nedir?

- İnsanlar çok eski zamanlardan günümüze kadar birçok iletişim yönleri kullanmışlardır. İlk başlarda dumanla haberleşirken, ateş üzerinde bir örtü örtterek, dumanın belirli sürelerde ve boyutlarda oluşmasını sağlıyorlardı. Böylece gökyüzünde oluşturulan her duman kümesi farklı anlamalar taşıyordu.
- Telgraf haberleşmesi de kullanılan mors alfabesi yöntemi duman ile haberleşmeye benzetilebilir. Örneğin; mors alfabesinde 'A' harfine karşılık gelen sinyal bir nokta ve bir çizgiden meydana gelir (._).

Mikroişlemcili Sistemler Dersi 14.12.2018

{ 3 }

UART Nedir?

- Günümüz kullanılan dijital haberleşme sistemlerinde her harfin, rakamın karşılığı '1' veya '0' sayıları ile temsil edilen voltaj seviyeleridir.
- UART (Universal Asynchronous Receiver - Transmitter) iletişimi 1 ve 0'lardan oluşan verileri iki dijital sistem arasında alıp-verme işleminde kullanılan bir iletişim protokolüdür.
- İki sistem arasında verilerin bit bit sıra ile yani seri olarak iletildiği iletişim protokolüdür.

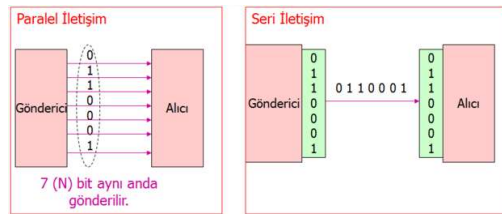
Mikroişlemcili Sistemler Dersi 14.12.2018

{ 4 }

UART Nedir?

- Dijital sistemlerde iletişim paralel ve seri olmak üzere iki şekilde gerçekleştirilir.

Paralel İletişim	Seri İletişim
Aynı anda birden fazla bit,	Aynı anda tek bit,
İletişim hızı yüksek,	İletişim hızı düşük,
Hat sayısı fazladır.	Hat sayısı azdır.



Mikroişlemli Sistemler Dersi 14.12.2018

5

UART İletişimi

- UART protokolü bir seri iletişim protokolüdür.
- UART protokolünde veriler senkron veya asenkron olarak gönderilip alınabilir.
- Senkron iletişimde alıcı ve verici devre eşzamanlı olarak çalışırlar. Bu iletişimde data hattı ve clock sinyali hattı bulunur. Clock hattı verilerin eşzamanlı olarak gönderilip alınmasını sağlar.



Mikroişlemli Sistemler Dersi 14.12.2018

6

UART İletişimi

- Asenkron iletişimde ise sadece data hatları bulunur. Asenkron iletişim, clock sinyali hattı bulundurmaz, bu nedenle alıcı ve verici eşzamanlı olarak çalışmazlar.



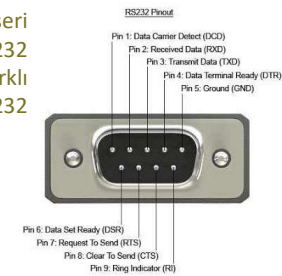
- Asenkron iletişimde clock sinyali yerine start, stop ve parity bitleri bulunur.
 - Start biti, gönderdiğimiz veriden önce gönderilir ve verinin gönderileceğini alıcı tarafa bildirir.
 - Stop biti, gönderdiğimiz verinin bittiğinin işaretidir ve veri sonunda alıcı tarafa gönderilir.
 - Parity biti ise verilerin alıcıya doğru şekilde iletilip iletilmediğini kontrol etmek için kullanılır.

Mikroişlemli Sistemler Dersi 14.12.2018

7

UART İletişimi

- Asenkron iletişimin hızı, baud rate ile belirlenir.
- Baud oranı (baud rate, bps) saniyede gönderilen bit sayısı olarak tanımlanır. İletişimin doğru bir şekilde sağlanabilmesi için alıcı ve vericinin baud oranları aynı olmalıdır.
- Masaüstü bilgisayarlarda RS232 seri iletişim portları bulunmaktadır. RS232 portu aracılığıyla bilgisayarlar farklı cihazlar ile haberleşebilir. RS232 portunun 9 adet pini vardır.



Mikroişlemli Sistemler Dersi 14.12.2018

8

MikroC UART Fonksiyonları

- **UARTx_Init**
- **UARTx_Data_Ready**
- **UARTx_Tx_Idle**
- **UARTx_Read**
- **UARTx_Read_Text**
- **UARTx_Write**
- **UARTx_Write_Text**
- **UART_Set_Active**

Mikroçlemelli Sistemler Dersi 14.12.2018

9

MikroC UART Fonksiyonları

UARTx_Init

- Seri iletişimi istenilen hız için başlangıç ayarlarını yapar. `UART1_Init(9600);`
// 9.6 kbps hızı için modül ayarlanır.

UARTx_Data_Ready

- Okunacak verinin hazır olduğunu gösterir.
- 1: veri hazır, 0: veri hazır değil.

// veri hazır ise okuma yapılır.

```
if (UART1_Data_Ready() == 1)
    receive = UART1_Read();
```

UARTx_Tx_Idle

- Gönderilmiş olan veri gittiye yeni veri gönderimi için transmitter portunun meşgul olup olmadığını test edilir.
- 1: veri gönderildi, 0 : meşgul.

```
if (UART1_Tx_Idle() == 1)
    { UART1_Write(_data); }
```

Mikroçlemelli Sistemler Dersi 14.12.2018

10

MikroC UART Fonksiyonları

UARTx_Read

- **UART üzerinden 1 byte lık veri okur.**
- `if (UART1_Data_Ready() == 1)`
`{ receive = UART1_Read(); }`

UARTx_Read_Text

- **UART üzerinden karakter dizisi okur.**
- `UARTx_Read_Text(çıkış, sınırlayıcı, girişim);`
 - Çıkış: Alınan karakter katarının kaydedileceği adresi belirten işaretçi,
 - Sınırlayıcı: Alınan karakter katarının sonundaki karakterleri belirten karakter dizisi,
 - Girişim: Alınan karakterlerin başlangıcından itibaren kaç tanesinde, «sınırlayıcı» olarak belirtilen karakterlerin aranacağını belirten sayı,

Mikroçlemelli Sistemler Dersi 14.12.2018

11

MikroC UART Fonksiyonları

UARTx_Write

- **UART üzerinden 1 byte lık veri yazar.**
- `unsigned veri=0x1E;`
-
- `UART1_Write(veri);`

UARTx_Write_Text

- **UART üzerinden karakter dizisi göndermek için kullanılır.**
- `Char dizi[25]=«Gomulu Sistemlere Giris»;`
-
- `UART1_Write_Text(dizi);`

Mikroçlemelli Sistemler Dersi 14.12.2018

12

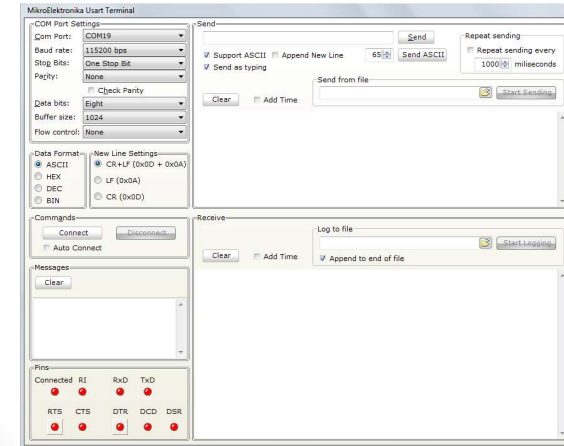
MikroC UART Fonksiyonları

UART Set Active

- **Aktif UART modülünü seçmek için kullanılır.**
- **UART_Set_Active**(Okuma, Yazma, Gelen veri hazır, Gönderme kaydedicisi hazır);
 - **Okuma: UART üzerinden veri okuma,**
 - **Yazma: UART üzerinden veri gönderme,**
 - **Gelen veri hazır: Veri tamponunun durumunu öğrenmek için kullanılır.**
 - **Gönderme kaydedicisi hazır: Gönderilecek veri kaydedicisinin durumunu öğrenmek için kullanılır.**
- **UART_Set_Active(&UART1_Read, &UART1_Write, &UART1_Data_Ready, &UART1_Tx_Idle);**

13

MikroC Usart Terminal Tool



14

MikroC UART Örnekleri

```

char uart_rd;

void main() {
    ANSEL = 0;           // Configure AN pins as digital
    ANSELH = 0;

    UART1_Init(9600);   // Initialize UART module at 9600 bps
    Delay_ms(100);      // Wait for UART module to stabilize

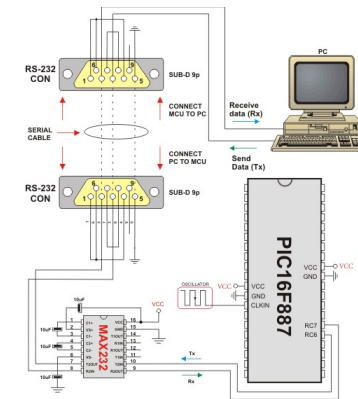
    UART1_Write_Text("Start");
    UART1_Write(10);
    UART1_Write(13);

    while (1) {        // Endless loop
        if (UART1_Data_Ready()) { // If data is received,
            uart_rd = UART1_Read(); // read the received data,
            UART1_Write(uart_rd); // and send data via UART
        }
    }
}

```

15

MikroC UART Örnekleri



Kaynak: MikroC Pro for PIC

16