



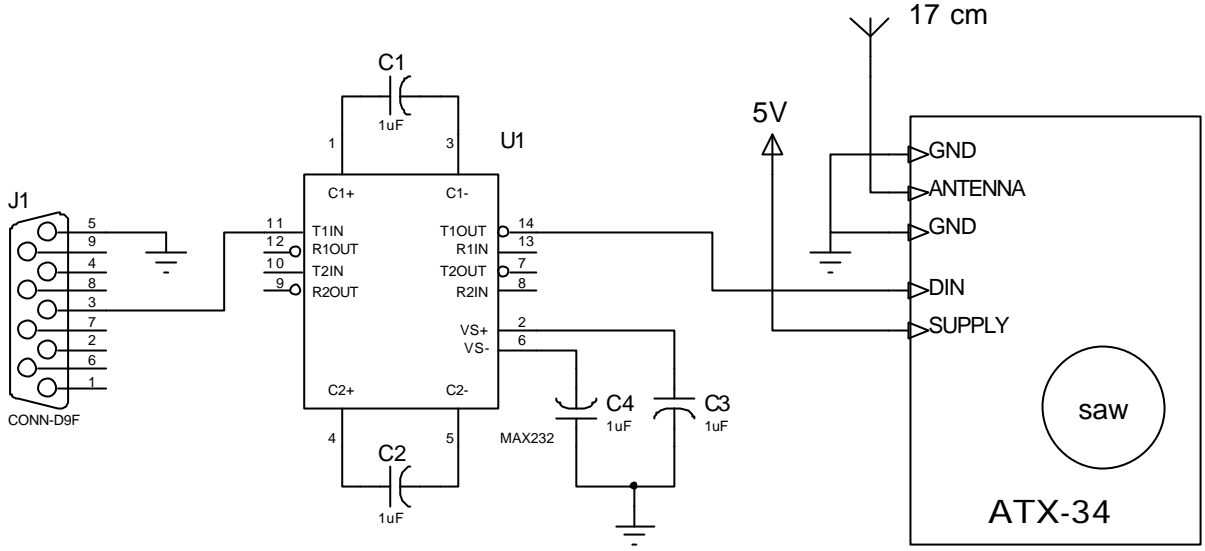
WIRELESS TECHNOLOGIES

UN-0405v01_PC-LCD

**ARX-34 ve ATX-34 KULLANARAK BILGISAYARIN
SERI PORTUNDAN 2*16 ALPHANUMERIC LCD'YE
DATA GÖNDERMEK İÇİN GEREKLİ YAZILIM ve
DONANIM**

WIRELESS COMMUNICATION PRODUCTS

VERICI DEVRESİ



Konnektör bilgisayarın COM portuna takılırken, PIC16F84'lü devrenin enerjili olmamasına dikkat edilmelidir.

Seri porttan göndermek istediğimiz datadan önce preamble ve sencron gönderilmelidir. Bu uygulamada preamble 5 byte 0x55, sencron ise 5 byte 0x00'dir. Örneğin LCD'ye "UDEA Elektronik" stringini göndermek için ;

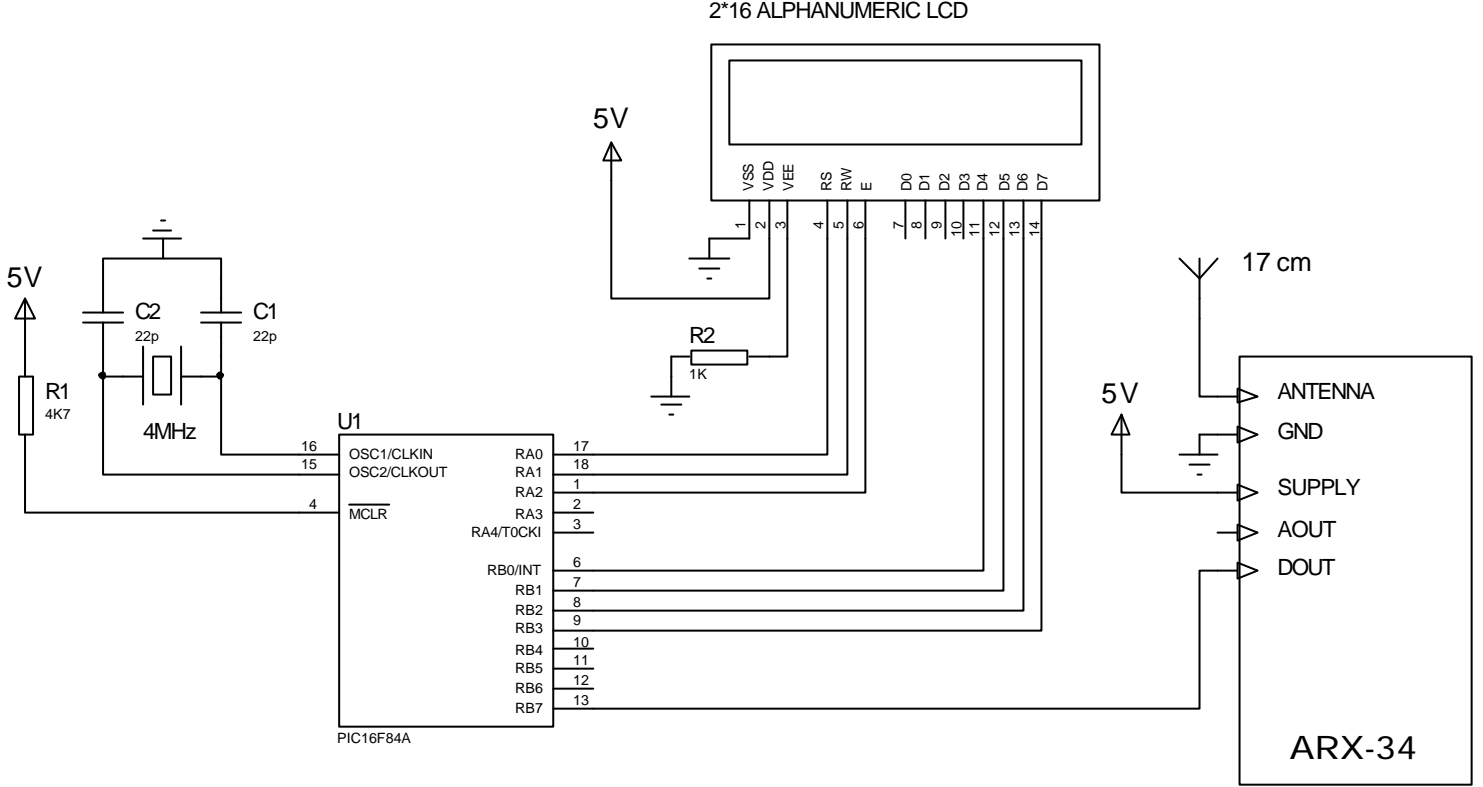
UUUUU\$00\$00\$00\$00\$00UDEA Elektronik.

Burada 0x55'in ASCII karşılığı olarak U, ve mesaj sonu bilgisi için 'nokta' karakteri kullanılmıştır. Mesaj sonu bilgisi, isteğe bağlı olarak herhangi bir karakter seçilebilir (yazılımda gerekli değişikliği yaparak).

Seri portu kullanacağınız programda yapmanız gereken diğer ayarlar da şunlardır:

Baudrate : 300
 Parity : Yok
 Byte size : 8
 Stop bits : 1

ALICI DEVRESI



ARX-34 içerisinde bir voltaj regülatörü bulunmamaktadır. Tasarım pil kullanımı düşünülerek yapılmıştır. Bu nedenle besleme voltajında belirtilen değerlere dikkat edilmelidir.

Modül belirtilen değerlerin altında bir besleme yapıldığında kararsız çalışacaktır. Besleme voltajı +5 VDC ve tapraklama GND bağlantısı belirtilen değerlerin üzerinde veya ters olursa, modülde kalıcı tahribatlara yolaçılabilir.

Modül basit bir anten bağlantı pinine sahiptir. Uygun bir UHF anten doğrudan bu pine bağlanabilir. ARX-34 modülüne bağlanabilecek en basit anten 17.3cm uzunluğundaki bir kablunun anten girişine lehimlenmesidir. Anteni, modülden uzak bir yere bağlamanız gerekiyorsa 50 Ohm Coax anten kablosu kullanmanız gerekmektedir. Anten kablosunun topraklaması, modülün anten girişine yakın bir yerden yapılmalıdır.

Aşağıdaki kurallar size yardımcı olacaktır:

- * Anten 50 ohm empedancli olmalıdır.
- * Lambda/4 whip anten boyu 433MHz. için yaklaşık 17.3 cm dir.
- * Anteni modül dik gelecek şekilde yukarıya veya aşağıya doğru monte ediniz.
- * Anteni metal bir hazne içine koymayınız.
- * İnsan vücudu metal objeler gibi etkiler gösterebilirler. Tasinabilir alıcı veya vericiler

vücuttan uzak bir şekilde elde tutulmalıdır.

- * En iyi iletişim mesafesi her iki taraftaki antenlerin birbirini görmesi ile elde edilebilir. Herhangi bir obje veya metal bir engel iletişim mesafesini düşürecektir.
- * Sinyal göndermeleri, gönderilen sinyallerin metal yüzeylerden, binalardan vb. gelen yansımalarından etkilenirler. Bu yanlış data alımlarına yolaçabilir.

Aşağıdaki yazılımla bilgisayardan yollanan datanın senkron bilgisi yakalanıp sonraki karakterler LCD'ye yazdırılmaktadır. Mesaj sonu bilgisi (bu yazılımda nokta karakteri) gelene kadar her karakter LCD'ye yazdırılır. Mesaj sonu bilgisi geldiğinde gönderilen mesaj 3 saniye kadar ekranda bekletilir. Daha sonra yeni bir mesaj için LCD temizlenir ve mikroislemci senkron aramaya devam eder.

```

/*****/
/* Uygulama Notu UN-0405v01_PC-LCD
/* SOFTWARE
/*
/* File: UN-2608v01_ASKRX.c
/*
/* Microcontroller:
/* Microchip PIC16F84
/*
/* Bu yazılım bilgisayardan alphanumeric LCD'ye
/* RF data göndermek için yazılmış basit bir uygulama notudur.
/*
/* Yazılımın içinde bir çok bekleme rutini kullanılmıstır.
/* Crystal frekansı ve compiler farklılığından doğacak
/* değişiklikler kullanıcı tarafından dikkate alınmalıdır.
/* Compiler olarak HI-TECH Software kullanılmıstır.
/* XT=4MHz
/*
/* Baudrate 300, None parity, Byte size 8, Stop bit 1
/*
/*
/* Yazan: Çagatay BÜYÜKTOPÇU, UDEA
/*
/* Contact: UDEA Elektronik 0.312.395 68 75
/* info@udea.com.tr
/*****/

```

```

#include <pic.h>
#include <delay.c>
#define RS RA0
#define RW RA1
#define EN RA2
#define RX RB7
#define bit_set(var,bitno) ((var) |= (1<<(bitno)))
#define bit_reset(var,bitno) ((var) &= ~(1<<(bitno)))

```

```
__CONFIG( XT&WDTDIS&PWRTDIS);
```

```
void lcd_reset(void);
void clear(void);
void function_set(void);
```

```
void on_off(void);
void entry_mode(void);
void data_gonder(unsigned char lcd);
void baud(void);
```

```
unsigned char karakter, lcd;
unsigned char x, y, say;
```

main()

```
{
    TRISB=0b10000000;           //PORTB<0:6> çıkis,PORTB<7> giris(RX)
    TRISA=0;                   //PORTA çıkis
    PORTA=0;
    PORTB=0;
    RW=0;                      //LCD'ye yaz
    TOCS=0;                    //Timer için clock internal
    PSA=0;                     //Prescaler TMR0 için
    PS2=1;
    PS1=1;
    PS0=1;                     //TMR0, prescale 256
    DelayMs(100);              //LCD'yi bekle
    lcd_reset();               //LCD initialization
    function_set();            //
    on_off();                  //
    entry_mode();              //
    clear();
    RS=1;                      //Data

    for(;;)
    {
        clear();
        while(y<=3)
        {
            TMR0=0;            //Timer'i baslat
            while(RX==0);      //Start bitinin geçmesini bekle
            if(TMR0>95)        //24ms geçtiyse
            {
                y++;
                while(RX==1);  //Stop bitini bekle
            }
            else
            {
                y=0;
            }
        }
        y=0;
        while(karakter!='.')
        {
            karakter=0;
            say=0;
            while(RX==0);      //Start bitini bekle
            baud();
            while(say<=7)
            {
                if(RX==1)
                    bit_set(karakter,7); //Gelen bit 1
                else
                    bit_reset(karakter,7); //Gelen bit 0
                karakter=karakter>>1;    //Bir bit saga kaydir
            }
        }
    }
}
```

```

        say++;
        baud();
    }
    data_gonder(karakter);
    while(RX==1);
}
for(x=0;x<=14;x++)
{
    DelayMs(200);
}

clear();
}
//Yeni bir mesaj için LCD'yi temizle
//Kursoru en basa al

void lcd_reset(void)
{
    RS=0;
    karakter=1;
    for(x=0;x<=2;x++)
    {
        PORTB=0;
        EN=1;
        NOP();
        EN=0;
        DelayMs(1);
    }
}

void clear(void)
{
    RS=0;
    karakter=0b00000001;
    PORTB=(karakter>>4)&(0b00001111);
    EN=1;
    NOP();
    EN=0;
    PORTB=(karakter&0b00001111);
    EN=1;
    NOP();
    EN=0;
    DelayUs(250);
}

void function_set(void)
{
    RS=0;
    karakter=0b00101000;
    PORTB=(karakter>>4)&(0b00001111);
    EN=1;
    NOP();
    EN=0;
    PORTB=(karakter&0b00001111);
    EN=1;
    NOP();
    EN=0;
    DelayMs(1);
}

```

//8 kere tekrar et
//Diğer bite geç
//Gelen datayı LCD'ye yaz
//Stop bitinin geçmesini bekle
//3 saniye kadar ekranda görüntüle
//Komut
//Komut
//Displayi temizle ve kursoru sayfa başına al
//Önce üst 4 biti gönder
//Sonra da alt 4 biti
//Komut
//4-bit veri aktarımı, 2 satır, 5*7 font

}

void on_off(void)

```

{
    RS=0;
    karakter=0b00001111;
    PORTB=(karakter>>4)&(0b00001111);
    EN=1;
    NOP();
    EN=0;

    PORTB=(karakter&0b00001111);
    EN=1;
    NOP();
    EN=0;
    DelayMs(1);
}

```

```

//Komut
//Display on, kursor on

```

void entry_mode(void)

```

{
    RS=0;
    karakter=0b00000110;
    PORTB=(karakter>>4)&(0b00001111);
    EN=1;
    NOP();
    EN=0;
    PORTB=(karakter&0b00001111);
    EN=1;
    NOP();
    EN=0;
    DelayMs(1);
}

```

```

//Komut
//Kursor saga dogru hareket eder

```

void data_gonder(unsigned char lcd)

```

{
    RS=1;
    karakter=lcd;
    PORTB=(karakter>>4)&(0b00001111);
    EN=1;
    NOP();
    EN=0;
    PORTB=(karakter&0b00001111);
    EN=1;
    NOP();
    EN=0;
}

```

void baud(void)**//Baud rate 300**

```

{
    for(x=0;x<16;x++)
    {
        DelayUs(215);
    }
}

```