



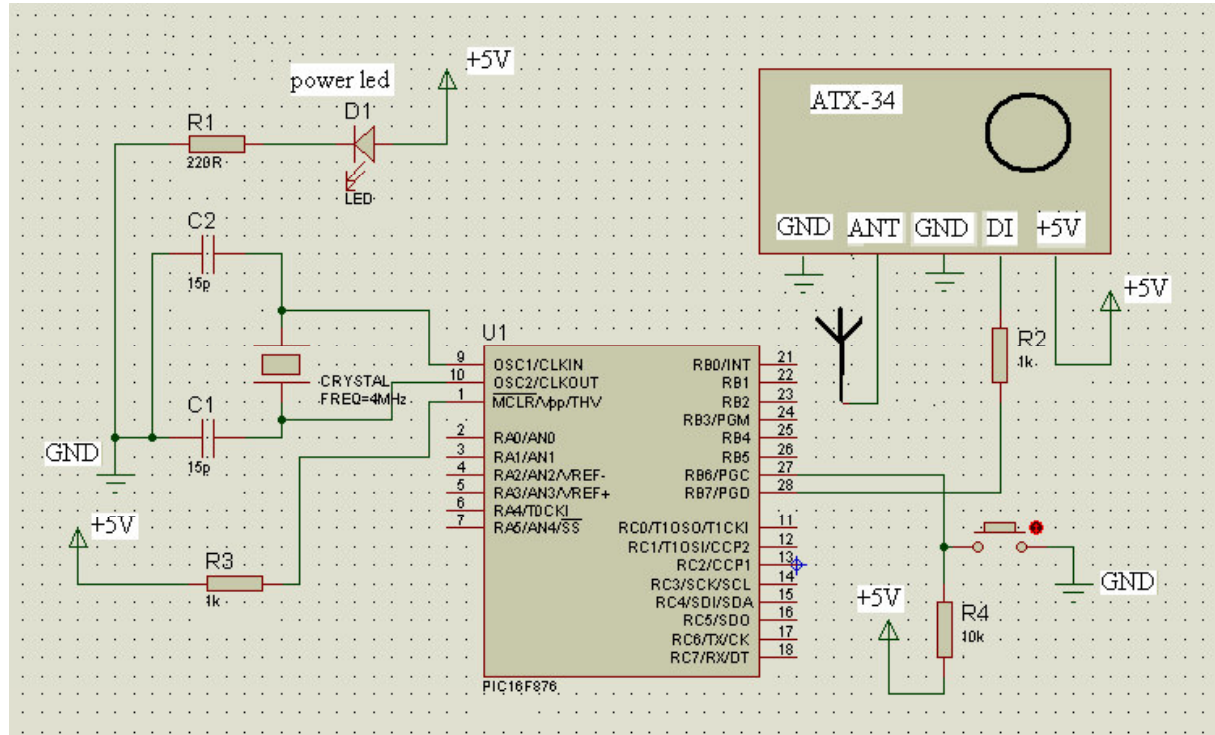
WIRELESS TECHNOLOGIES

UN-0805v01_ATX-ARX

ARX-34 ve ATX-34 KULLANARAK DATA
GÖNDERMEK VE ALMAK İÇİN GEREKLİ YAZILIM ve
DONANIM

WIRELESS COMMUNICATION PRODUCTS

VERİCİ DEVRESİ



- Yazılım aşağıda verilmiştir. Yazılım PIC16F876 veya PIC16F876A için yazılmıştır.
- Baud rate 2400 dür.
- Yazılımda zamanlama çok önemlidir. Bu yüzden 4 Mhz kristal kullanılmalı veya yazılım kullanılacak kristal frekansına uyarlanmalıdır.
- Devre şemasının çizildiği Proteus6 ISIS programında PIC16F876 nın Vss ve Vdd bağlantıları belirtilmemiştir. Devre kurulurken bu bağlantılar unutulmamalıdır. Vss ye GND Vdd ye +5V DC bağlanmalıdır.

```

/*****/
/* Uygulama Notu UN-0805v01_ATX-ARX
/* SOFTWARE
/*
/* File: verici.c
/*
/* Microcontroller: Microchip PIC16F876
/*
/* Yazılımın içinde bir çok bekleme rutini kullanılmıştır.
/* Crystal frekansı ve compiler farklılığından doğacak
/* değişiklikler kullanıcı tarafından dikkate alınmalıdır.
/* Compiler olarak HI-TECH Software kullanılmıştır.
/* XT=4MHz
/*
/*
/* Yazan: Mehmet Yağcı, UDEA
/*
/* Contact: UDEA Elektronik 0.312.395 68 75
/* info@udea.com.tr
/*****/
#include <pic.h>
#include <delay.c>

#define DI RB7 // RB7 pini kullanılarak data gönderiliyor
#define button RB6 // gönderme yapmak için kullanılan tuş

int count=0; // 1 byte=8 bit
unsigned char data=0x00; // gönderilecek verilerin atandığı değişken
unsigned char data_set=0x00; // bit_reset ve bit_set denetim değişkeni
unsigned char mask=0x80;

void p_width()
{
DelayUs(200); // baud rate değerimiz 2400, her bir pluse 416 Us olmalı
DelayUs(175); // send(data) programındaki gecikmeler dikkate alındığında
} // data gönderme sırasında pulse width 375 Us olur

void send(data) // data gönderen alt program
{
while(count<=7)
{
data_set=data & mask;
if(data_set==0x80)
{
DI=1;
}
else
{
DI=0;
}
p_width();
if(count==7)
{

```

```

        count++;
        break;
    }
    data=data<<1;
    count++;
}
count=0;
return;
}

void main() // ana program
{
    TRISB=0X40; // portb nin tüm pinleri çıkış yapıldı
    for(;;)
    {
        if(button==0) // tuş basılı tutulduğu sürece data gönderir
        {
            send(0x55); // 5 byte 0x55 (preamble)
            send(0x55);
            send(0x55);
            send(0x55);
            send(0x55);

            send(0x00); // 5 byte 0x00 ve 5 byte 0xff (senkron)
            send(0x00);
            send(0x00);
            send(0x00);
            send(0x00);

            send(0xff);
            send(0xff);
            send(0xff);
            send(0xff);
            send(0xff);

            DI=0; // stop biti

            DI=1; // start biti
            p_width();

            send(0x42); // gönderilen data

            DelayMs(250); // 500 ms gecikme
            DelayMs(250);
        }
    }
}
//-----

```


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

Aşağıdaki yazılımla bilgisayardan yollanan datanın sencon 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 mikroişlemci sencon aramaya devam eder.

```

/*****/
/* Uygulama Notu UN-0805v01_ATX-ARX
/* SOFTWARE
/*
/* File: alıcı.c
/*
/* Microcontroller: Microchip PIC16F876
/*
/* Yazılımın içinde bir çok bekleme rutini kullanılmıştır.
/* Crystal frekansı ve compiler farklılığından doğacak
/* değişiklikler kullanıcı tarafından dikkate alınmalıdır.
/* Compiler olarak HI-TECH Software kullanılmıştır.
/* XT=4MHz
/*
/*
/* Yazan: Mehmet Yağcı, UDEA
/*
/* Contact: UDEA Elektronik 0.312.395 68 75
/* info@udea.com.tr
/*****/
#include <pic.h>
#include <delay.c>

#define Dout RB6 // ARX den datanın alındığı pin
#define LED RC0 // datanın yanlış okunduğunu gösteren led
#define DLED RC3 // datanın doğru okunduğunu gösteren led
#define bit_set(var,bitno) ((var) |= (1<<(bitno))) // istenilen biti 1 yapan macro
#define bit_reset(var,bitno) ((var) &= ~(1<<(bitno))) // istenilen biti 0 yapan macro

int plus0,plus1,plus2; // sayıcılar
int count,say; // data bit sayıcısı
unsigned char data=0x00; // datanın atanacağı değişken

void led(); // data yanlış okundu ledini yakan alt program
void Dled(); // data doğru okundu ledini yakan alt program
void fivebit0(); // senkron 5 bit 0x00 ı okuyan alt program
void fivebit1(); // senkron 5 bit 0xff i okuyan alt program
void data_oku(); // alınan datayı okuyan program

```

```

//-----ana program-----
void main()
{
  TRISB=0XFF;           //PORTB input tanımlandı
  TRISC=0X00;          //PORTC output tanımlandı
  PORTB=0;
  PORTC=0;
  for(;;)
  {
    fivebit0();
  }
}
//-----
void led()
{
  LED=1;
  DelayMs(250);
  DelayMs(250);
  DelayMs(250);
  DelayMs(250);
  LED=0;
  DelayMs(250);
  DelayMs(250);

  plus0=0;
  plus1=0;
  plus2=0;
  say=0;
  data=0;
  return;
}
//-----
void Dled()
{
  DLED=1;
  DelayMs(250);
  DelayMs(250);
  DelayMs(250);
  DelayMs(250);
  DLED=0;
  DelayMs(250);
  DelayMs(250);

  plus0=0;
  plus1=0;
  plus2=0;
  say=0;
  data=0;
  return;
}
//-----5 bit 0x00 arayan program-----

```

```

void fivebit0()
{
    for(;;)
    {
        if(Dout==0) //senkron 5 bit 0x00 araniyor
        {
            for(;;)
            {
                DelayUs(1);
                plus0++;
                if(plus0>=600) // 5 byte 0x00 bulundu
                {
                    while(Dout==0);
                    fivebit1(); // 5 bte 0xff arayan programa git
                    return;
                }
                if(Dout==1) // 5 byte 0x00 bulunamadı
                break;
            }
            plus0=0; // 0x00 yeterli uzunluğu sağlamiyor başa dön ara
        }
    }
}
//-----5 bit 0xff arayan program-----
void fivebit1()
{
    for(;;)
    {
        if(Dout==1) //senkron 5 byte 0xff araniyor
        {
            for(;;)
            {
                DelayUs(10);
                plus1++;
                if(plus1>=600) // 5 bit 0xff bulundu
                {
                    while(Dout==1);
                    data_oku(); // senkron yakalandı data_oku programına git
                    return;
                }
                if(Dout==0) // 5 byte ff bulunamadı
                {
                    plus0=0; // 0xff yeterli uzunluğu sağlamıyor başa dön ara
                    plus1=0;
                    return;
                }
            }
        }
    }
}

```



```
//-----  
void data_oku()  
{  
    while(say<=7) // 8 bit data oku  
    {  
        if(Dout==1)  
            bit_set(data,0); // biti 1 yap  
        else  
            bit_reset(data,0); // biti 0 yap  
        if(say==7)  
            break;  
        say++;  
        data=data<<1; // datayı bir bit sola kaydır  
        DelayUs(200);  
        DelayUs(210);  
    }  
    if(data==0x42) // 8 bit okundu alınan datayı kontrol et  
    {  
        Dled(); // data doğru  
        return;  
    }  
    else  
    {  
        led(); // data yanlış  
        data=0;  
        say=0;  
        return; // ana programa dön  
    }  
}  
//-----
```