



T.C.

KIRIKKALE ÜNİVERSİTESİ

MÜHENDİSLİK ve MİMARLIK FAKÜLTESİ

Veri İletişimi Ders Projesi

Doç. Dr. BÜLENT GÜRSEL EMİROĞLU

Öğrenci Adı Soyadı: Rami KOCO

Öğrenci Numarası: 170255063

Giriş:

1. Giriş	1
2. Projenin Adı.....	2
3. Projenin Amacı.....	2
4. Proje İletişimi.....	2
5. Çoklu Köle (Slave) ile SPI Master	3
6. Arduino Nano'daki SPI Pinleri.....	3
7. Arduino'da SPI Kullanımı.....	4
8. SPI İletişiminin Avantajları	4
9. SPI'nin Dezavantajları.....	5
10. Uzun Mesafe Kablolı İletişim için Devre Şeması.....	5
11. MAX485 UART-RS485 Dönüştürücü Modül.....	7
12. Ethernet CAT-6E Kablosu	7
13. Projede Kullanılan Kodlar	9

Projenin Adı: Tam Çift Yönlü İletişim

Projenin Amacı: İnternetsiz Haberleşme Sistemi.

Proje İletişimi: Full Duplex İletişim SPI (Serial Peripheral Interface) Seri Çevre Birimi Kullanılarak yapıldı.

SPI (Serial Peripheral Interface) bir seri iletişim protokolüdür.

SPI, tam çift yönlü bağlantıya sahiptir, bu da verilerin aynı anda gönderilip alınması anlamına gelir.

Yani bir master, bir slave'e veri gönderebilir ve bir slave, aynı anda master'a veri gönderebilir.

SPI, senkron seri iletişimdir, iletişim amacıyla saatin gerekli olduğu anlamına gelir.

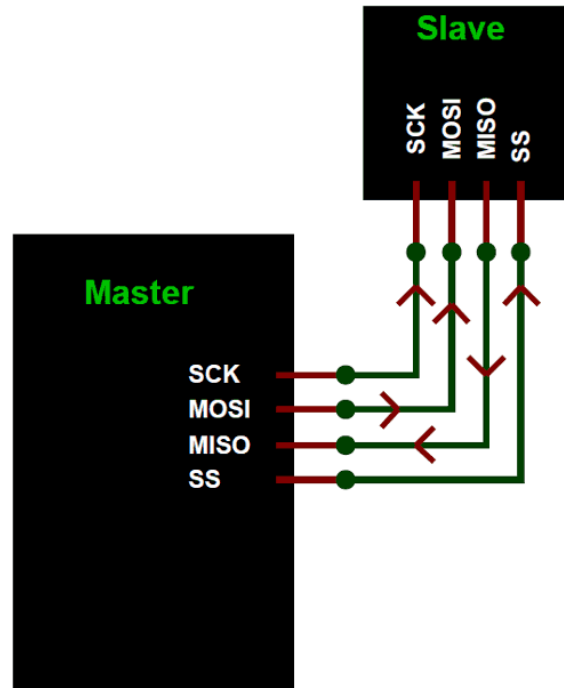
SPI'nin çalışması;

Bir SPI, dört hat kullanarak bir master/Slave iletişimine sahiptir.

Bir SPI'nin yalnızca bir master'ı olabilir ve birden fazla slave'i olabilir.

Master genellikle bir mikrodenetleyicidir ve köleler bir mikrodenetleyici, sensörler, ADC, DAC, LCD vb. olabilir.

Aşağıda Single Slave ile SPI Master'ın blok diyagram gösterimi bulunmaktadır.



SPI, aşağıdaki dört satıra sahiptir: MISO, MOSI, SS ve CLK

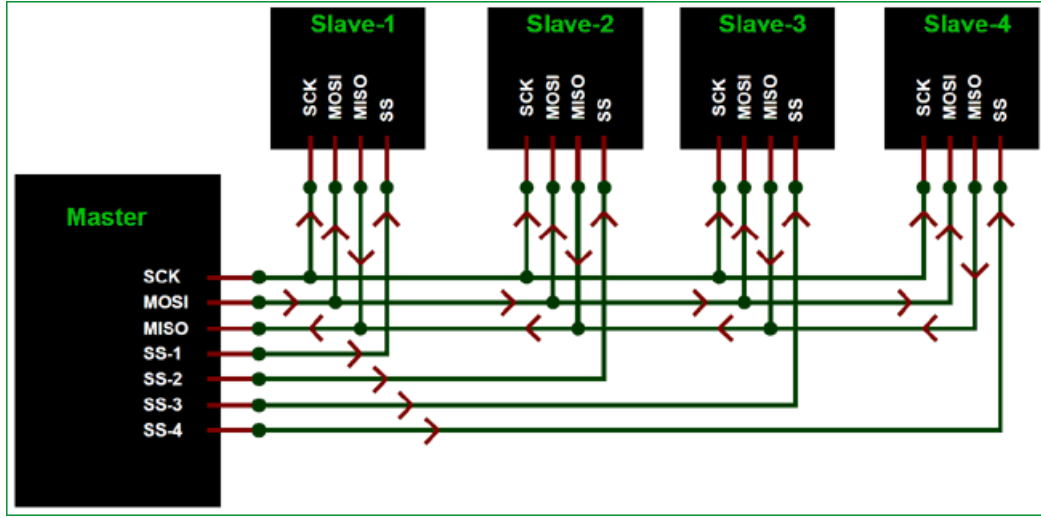
MISO (Master in Slave Out)- Master'a veri göndermek için Slave hattı.

MOSI (Master Out Slave In)- Çevre birimlerine veri göndermek için Ana hat.

SCK (Seri Saat)- Master tarafından üretilen veri iletimini senkronize eden saat darbeleri.

SS (Slave Select) –Master, belirli cihazları etkinleştirmek ve devre dışı bırakmak için bu pimi kullanabilir.

Çoklu Köle (Slave) ile SPI Master



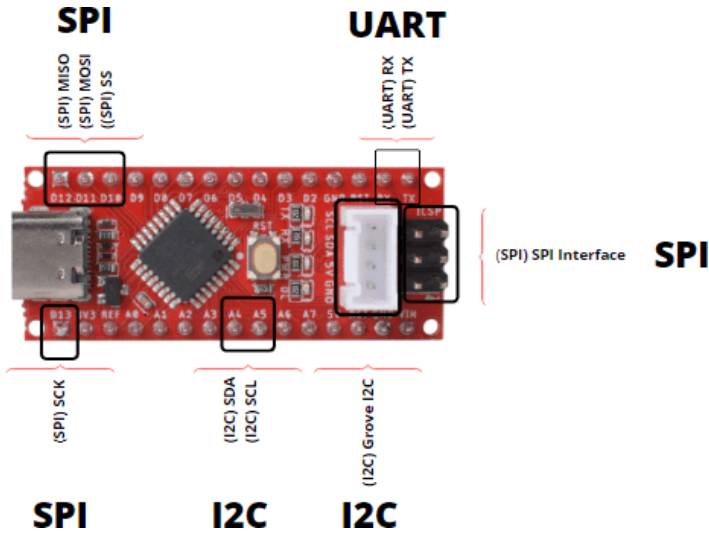
Master ve slave arasındaki iletişimi başlatmak için gerekli cihazın Slave Select (SS) pinini LOW olarak ayarlamamız gerekir, böylece master ile iletişim kurabilir.

Yüksek olduğunda, ustayı görmezden gelir. Bu, aynı MISO, MOSI ve CLK ana hatlarını paylaşan birden fazla SPI cihazına sahip olmanızı sağlar.

Yukarıdaki resimde görebileceğiniz gibi, SCLK, MISO, MOSI'nin master'a ortak olduğu ve her bir slave'in SS'sinin ayrı ayrı SS pinlerine (SS1, SS2, SS3) bağlı olduğu dört slave vardır.

Gerekli SS pinini DÜŞÜK olarak ayarlayarak bir ana, bu bağımlı birim ile iletişim kurabilir.

Arduino Nano'daki SPI Pinleri



SPI Hattı	Arduino'da Pin
MOSI	11
MISO	12
SCK	13
SS	10

Arduino'da SPI Kullanımı

İki Arduino arasındaki SPI iletişimi için programlamaya başlamadan önce. Arduino IDE'de kullanılan Arduino SPI kütüphanesini öğrenmemiz gerekiyor.

<SPI.h> kitaplığı, SPI iletişimi için aşağıdaki işlevleri kullanmak için programa dahil edilmiştir.

1. SPI.başlangıç()

KULLANIM: SCK, MOSI ve SS'yi çıkışlara ayarlayarak, SCK ve MOSI'yi düşük ve SS'yi yüksek çekerek SPI veri yolunu başlatmak için.

2. SPI.setClockDivider(bölücü)

KULLANIM: SPI saat bölücüyü sistem saatine göre ayarlamak için. Kullanılabilir bölücüler 2, 4, 8, 16, 32, 64 veya 128'dir.

Bölücüler:

SPI_CLOCK_DIV2

SPI_CLOCK_DIV4

SPI_CLOCK_DIV8

SPI_CLOCK_DIV16

SPI_CLOCK_DIV32

SPI_CLOCK_DIV64

SPI_CLOCK_DIV128

3. SPI.attachInterrupt(işleyici)

KULLANIM: Bu işlev, bir bağımlı cihaz ana cihazdan veri aldığı anda çağrılır.

4. SPI.transfer(val)

KULLANIM: Bu fonksiyon, master ve slave arasında aynı anda veri göndermek ve almak için kullanılır.

SPI İletişiminin Avantajları

SPI'yi seri iletişim arabirimi olarak öne çıkaran birçok neden vardır.

1. Tam Çift Yönlü seri iletişim.
2. SPI ile potansiyel olarak yüksek hızlı veri aktarım hızları elde edilebilir.
3. SPI Bus (tipik olarak 4 hat) için birkaç kablo gereklidir.
4. Kolay yazılım yapılandırması ve kurulumu.
5. Son derece esnek veri aktarımı. 8-Bit ile sınırlı değildir, herhangi bir keyfi boyutta kelime olabilir.
6. Çok basit donanım yapısı. Kölelerin(slave) benzersiz adreslere ihtiyacı yoktur.

SPI'nin Dezavantajları

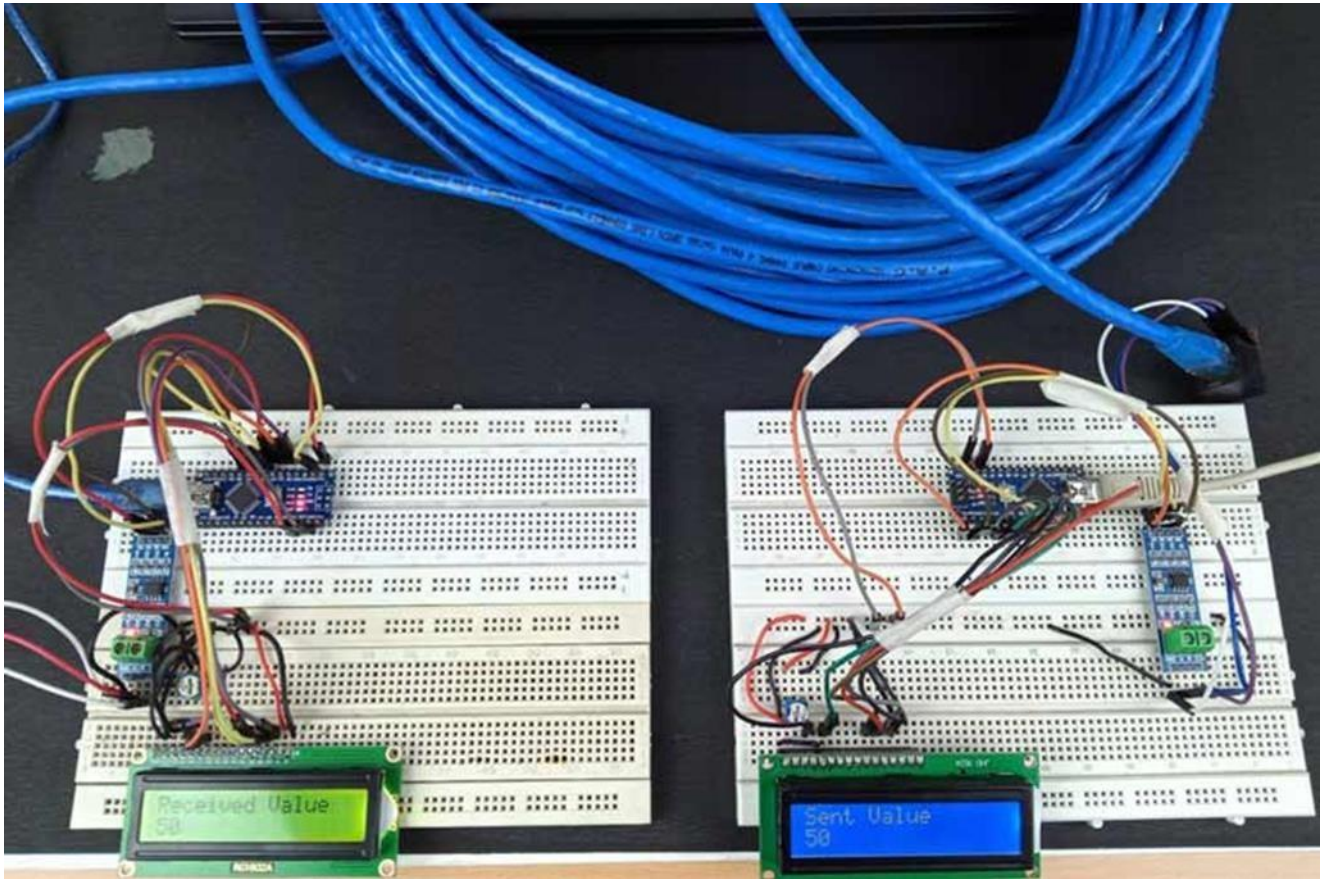
1. Donanım bağımlı onayı yok (master bilmeden hiçbir yere iletmiyor olabilir).
2. Genellikle yalnızca bir ana cihazı destekler.
3. Daha fazla pin gerektirir.
4. Donanım düzeyinde hata denetimi protokolü tanımlanmamıştır.

Uzun Mesafe Kablolu İletişim için Devre Şeması

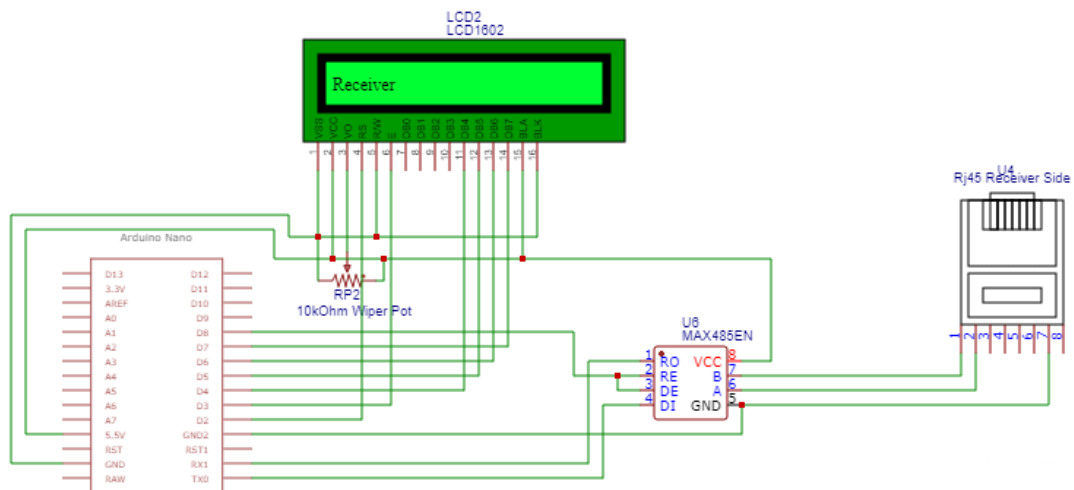
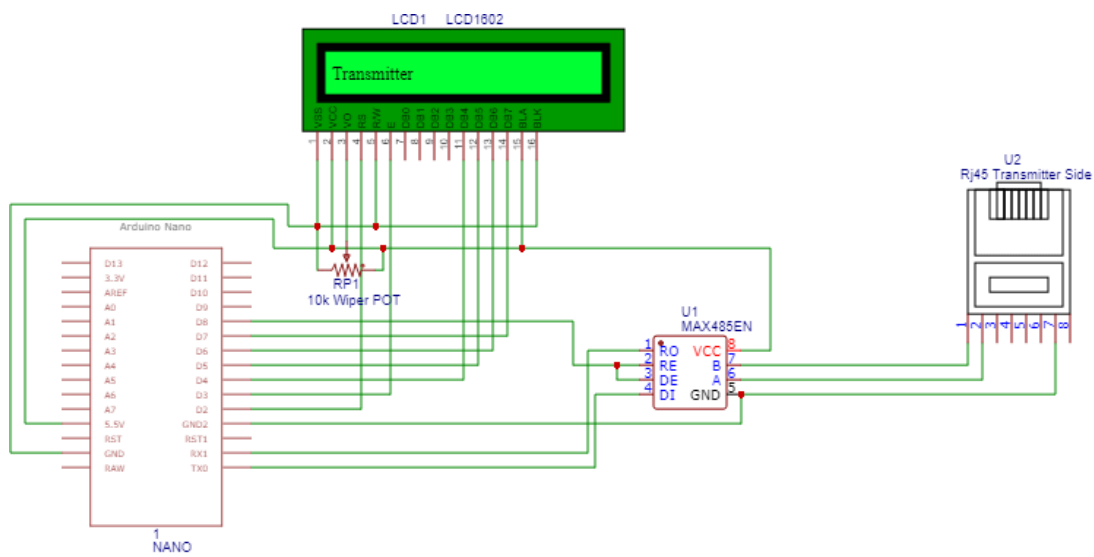
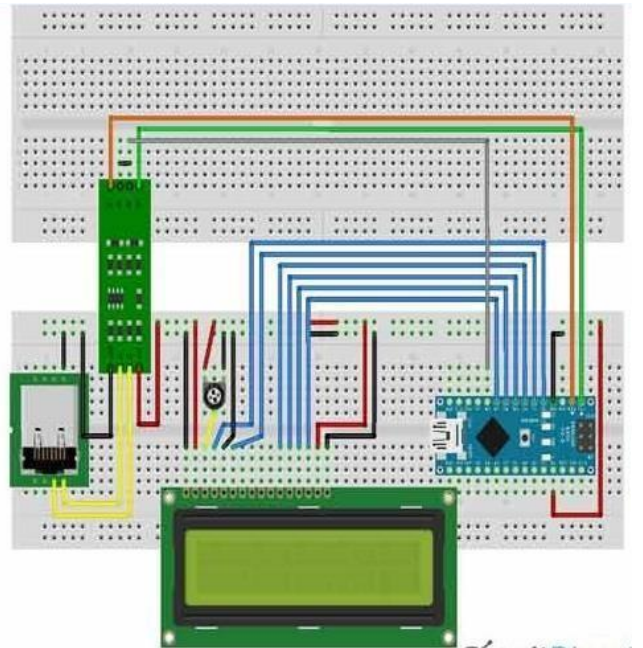
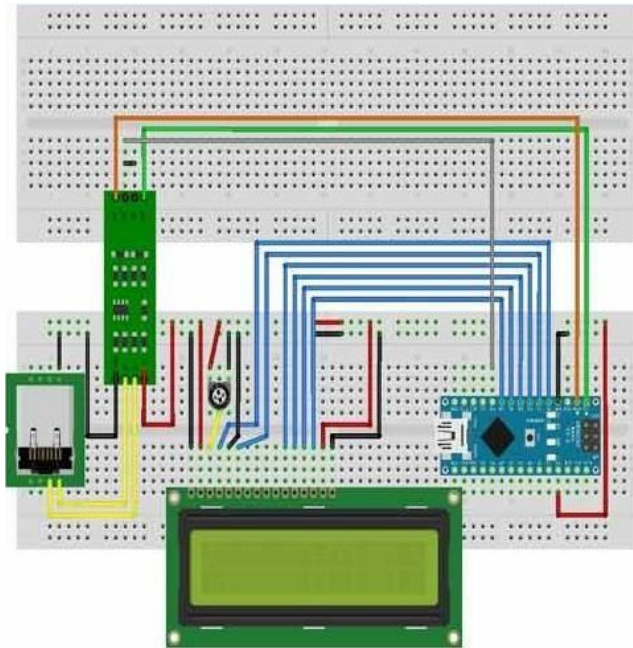
Aşağıdaki görüntü Arduino'nun uzun mesafe kablolu iletişimi için verici ve alıcı devre şemasını göstermektedir .

Hem verici hem de alıcı, farklı olan tek şey, içine yazılan koddur.

Ayrıca gösteri için, bir panoyu verici ve bir panoyu alıcı olarak kullanıyoruz, ancak panoları aynı kurulumla hem verici hem de alıcı olarak çalışacak şekilde kolayca programlayabiliriz.



Yukarıdaki devrenin bağlantı şeması da aşağıda verilmiştir.



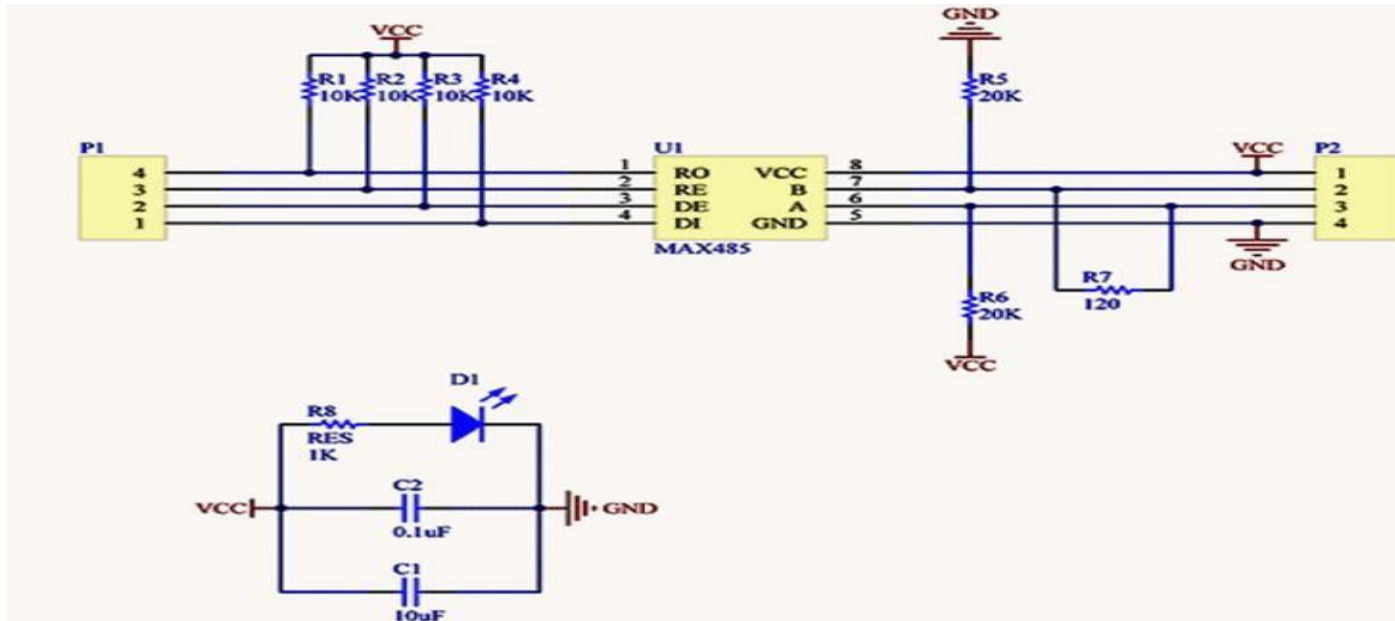
Alıcı uta, MAX485 dnřtrc modl Slave olarak alıřır ve Master'dan iletimi dinlerken aldıđı RS485 Verisini tekrar alıcı Nano tarafından okunacak ve 16*'da grntlenecek standart 5V TTL UART sinyallerine dnřtrr* 2 Alfannmerik LCD ona bađlı.

MAX485 UART-RS485 Dnřtrc Modl

Bu UART-RS485 Dnřtrc modl, RS-485 iletiřimi iin kullanılan dřk gl ve dnř hızı sınırlı bir alıcı-verici olan yerleřik bir MAX485 ipine sahiptir.

Tek bir +5V g kaynađında alıřır ve anma akımı 300 μ A'dır.

MAX485 alıcı-verici, src devre dıřı bırakıldıđında yksz veya tam ykl kořullarda 120 μ A ile 500 μ A arasında bir besleme akımı eker. Src, kısa devre akımı iin sınırlandırılmıřtır ve src ıkıřları, termal kapatma devresi aracılıđıyla yksek empedans durumuna yerleřtirilebilir. Alıcı giriři, giriřin aık devre olması durumunda mantıksal yksek ıkıřı garanti eden arızaya karřı gvenli bir zelliđe sahiptir. Ayrıca, gl parazit nleme performansına sahiptir. Ayrıca ipin mevcut durumunugrntlemek iin yerleřik LED'lere sahiptir, yani



Yukarıda verilen devre řeması, yerleřik MAX485 IC'nin eřitli bileřenlere nasıl bađlandıđını aıklar ve isterseniz breadboard ile kullanılmak zere 0,1 inlik standart aralık bařlıkları sađlar.

Ethernet CAT-6E Kablosu

Uzun mesafeli veri aktarımı denilince aklımıza hemen Ethernet kabloları ile internete bađlanmak geliyor.

Gnmzde internet bađlantısı iin ođunlukla Wi-Fi kullanıyoruz, ancak daha nce internete bađlamak iin her kiřisel bilgisayara giden Ethernet kablolarını kullanıyorduk.

Bu Ethernet Kablolarını normal kablolar üzerinden kullanmanın ana nedeni, yüksek mesafelerde gürültünün içeri girmesine ve sinyalin bozulmasına karşı çok daha iyi koruma sağlamalarıdır.

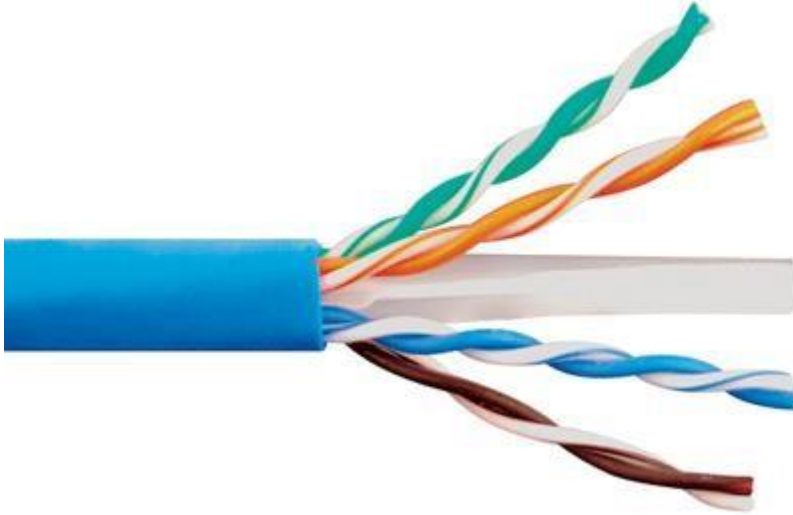
Elektromanyetik Girişime karşı koruma sağlamak için yalıtım katmanının üzerinde Koruyucu Ceket bulunur.

Ve ayrıca her bir tel çifti, herhangi bir akım döngüsü oluşumunu önlemek ve böylece gürültüye karşı daha iyi koruma sağlamak için birlikte bükülür.

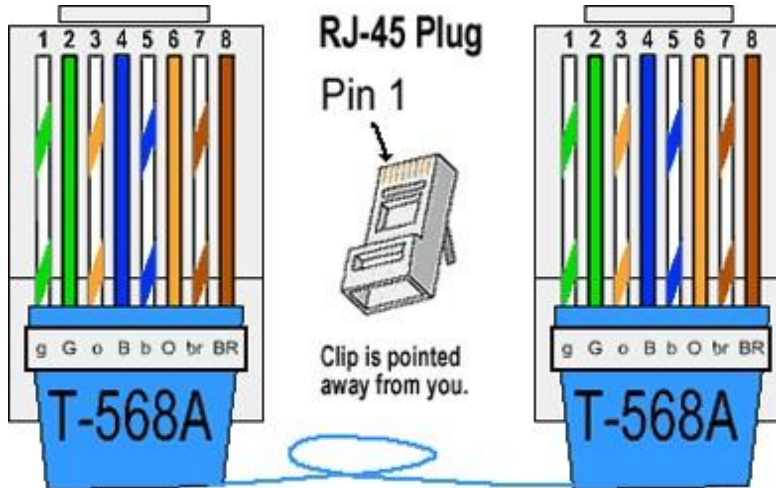
Genellikle her iki uçta da 8 pinli RJ45 konektörlerle sonlandırılırlar. CAT-4, CAT-5, CAT-5E, CAT-6, CAT-6A gibi kullanabileceğimiz birçok Ethernet kablosu kategorisi vardır.

Eğitimimizde 4 bükümlü CAT-6E kablosu kullanacağız. 24AWG kablo çiftleri ve 600MHz'e kadar destekleyebilir.

CAT-6E Kablosunun Yalıtım Katmanı içinde bir çift telin nasıl büküldüğünü gösteren resim



CAT-6E Ethernet Kablosu için RJ-45 Konektörü



Projede Kullanılan Kodlar

Bu projede, sonuçları görüntülemek için her biri 16*2 Alfanümerik LCD kullanan biri verici ve biri alıcı olarak iki Arduino Nano kullanıyoruz.

Bu yüzden Arduino kodunda, verileri göndermeye ve gönderilen veya alınan verileri LCD ekranda görüntülemeye odaklanacağız.

//SPI MASTER CODE

```
#include<SPI.h>                //Library for SPI

#include <LiquidCrystal.h>

#define LED 9                  //Led pin

#define ipbutton 2             //Buttin pin

int buttonvalue;

int x;

String ogrenciisimleri[60] = { "Rami   ", "Ilknur   ", "Kagan   ", "Ali     ", "Veli    ", "Mehmet  ",
    "Ahmet   ", "Murat   ", "Harun   ", "Yunus   ", "Meltem   ",
    "Selin   ", "Cihan   ", "Mert    ", "Berk    ", "Samet   ", "Halil   ", "Hasan   ", "Efe     ", "Can     ",
    "Hulya   ", "Ruya    ", "Aziz    ", "Sevim   ", "Ezgi    ",
    "Tarik   ", "Merve   ", "Zeynep  ", "Gamze   ", "Buse    ", "Bilal   ", "Recep   ", "Metem   ", "Burak   ",
    "Kamil   ", "Cemal   ", "Kemal   ", "Hayri   ", "Berkay  ",
    "Mesut   ", "Alper   ", "Erol    ", "Fikri   ", "Zeki    ", "Candar  ", "Hilal   ", "Hayati  ", "Gulay   ",
    "Esmem   ", "Esra    ", "Aytekin ", "Ramiz   ", "Rasim   ",
    "Derya   ", "Yasemin ", "Mehtap  ", "Bayram  ", "Fatma   ", "Ahmet   ", "Sude    "};

LiquidCrystal lcd(8, 7, 6, 5, 4, 3);

int sayi;

void setup (void)

{

    Serial.begin(115200);        //Starts Serial Communication at Baud Rate 115200

    lcd.begin(16, 2);
```

```
pinMode(ipbutton,INPUT);      //Sets pin 2 as input
pinMode(LED,OUTPUT);         //Sets pin 7 as Output
```

```
SPI.begin();                 //Begins the SPI commnuication

SPI.setClockDivider(SPI_CLOCK_DIV8); //Sets clock for SPI communication at 8 that's mean  $16/8 = 2$ Mhz for the frequency

digitalWrite(SS,HIGH);       //Setting SlaveSelect "SS" as HIGH
//randomSeed(analogRead(A0));
}
```

```
void loop(void)
```

```
{
```

```
byte Mastersend,Mastereceive;
```

```
buttonvalue = digitalRead(ipbutton); //Reads the status of the pin 2
```

```
if(buttonvalue == HIGH)      //Logic for Setting x value depending upon input from pin 2
```

```
{
```

```
  x = 1;
```

```
}
```

```
else
```

```
{
```

```
  x = 0;
```

```
}
```

```
digitalWrite(SS, LOW);      //Starts communication with Slave connected to master
```

```
Mastersend = x;
```

```
Mastereceive=SPI.transfer(Mastersend); //Send the mastersend value to slave also receives value from slave
```

```
Serial.println(Mastereceive);
```

```

    if(Mastereceive == 1)          //Logic for setting the LED output depending upon value received
    from slave
    {
        digitalWrite(LED,HIGH);    //Sets pin 7 HIGH
        Serial.println("Master LED ON");
        lcd.setCursor(0, 0);
        //lcd.print("Master LED ON ");
        // randomSeed(analogRead(A0));
        sayi = random(0,60);
        lcd.print(ogrenciisimleri[sayi]);
    }
    else
    {
        digitalWrite(LED,LOW);      //Sets pin 7 LOW
        Serial.println("Master LED OFF");
        lcd.setCursor(0, 0);
        lcd.print("Master OFF      ");
    }
    delay(1000);
}

```

//SPI SLAVE (ARDUINO)

//SPI COMMUNICATION BETWEEN TWO ARDUINO

//CIRCUIT DIGEST

//Pramoth.T

#include<SPI.h>

#define LEDpin 7

#define buttonpin 2

volatile boolean received;

volatile byte Slavereceived,Slavesend;

```

int buttonvalue;

int x;

void setup()

{
    Serial.begin(115200);

    pinMode(buttonpin,INPUT);        // Setting pin 2 as INPUT
    pinMode(LEDpin,OUTPUT);          // Setting pin 7 as OUTPUT
    pinMode(MISO,OUTPUT);            //Sets MISO as OUTPUT (Have to Send data to Master IN

    SPCR |= _BV(SPE);                //Turn on SPI in Slave Mode
    received = false;

    SPI.attachInterrupt();            //Interuupt ON is set for SPI commnucation

}

ISR (SPI_STC_vect)                  //Inerrrrput routine function
{
    Slaverceived = SPDR;              // Value received from master if store in variable slaverceived
    received = true;                  //Sets received as True
}

void loop()

{ if(received)                       //Logic to SET LED ON OR OFF depending upon the value recerived from
  master
  {
    if (Slaverceived==1)
    {
        digitalWrite(LEDpin,HIGH);    //Sets pin 7 as HIGH LED ON
    }
  }
}

```

```
    Serial.println("Slave LED ON");
}
else
{
    digitalWrite(LEDpin,LOW);    //Sets pin 7 as LOW LED OFF
    Serial.println("Slave LED OFF");
}

buttonvalue = digitalRead(buttonpin); // Reads the status of the pin 2

if (buttonvalue == HIGH)    //Logic to set the value of x to send to master
{
    x=1;

}
else
{
    x=0;
}

Slavesend=x;
SPDR = Slavesend;    //Sends the x value to master via SPDR
delay(1000);
}
}
```
