Mehmet Çınar

ARDUNIO VE BİLGİSAYAR ARASI VERİ İLETİŞİMİ

**Proje Hakkında Kısa Bilgilendirme**

**Proje Adı:** Mesafe Önemli

**Projenin Amacı:** Sensor den ölçülen değerlerin bilgisayara aktarılması ve bilgisayardan ardunıo devreye veri gönderilmesi ile Ardunıo ve bilgisayar arası karşılıklı veri iletişimini proje yardımıyla daha iyi kavramak ve anlatmak.

**Projeyi Yaparken Kullandığımız Uygulamalar:**

* Ardunıo IDE
* Visual Studıo (Windows Form Application)

**Projede Kullanılan Devre ve Elemanları:**

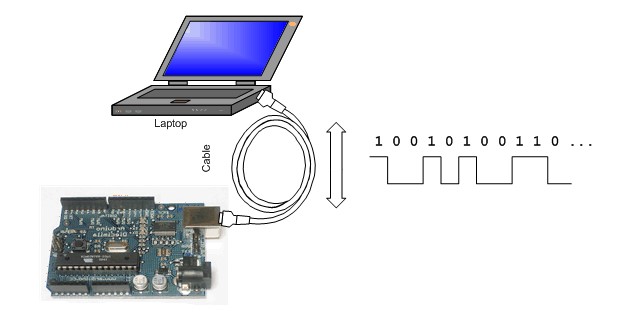
* 2 Adet Ardunıo Uno R3
* 1 Adet HC-SR04 Ultrasonik Mesafe Ölçer
* 1 Adet Breadboard
* 2 Adet Kırmızı Led
* 9 Adet Jumper Kablo

**Proje Öncesi Bilinmesi Gerekenler**

**Ardunıo ve Bilgisayar Arası İletişim**

Projemizde veri iletimi yolu olarak Seri Port kullanıyoruz Seri port bize hem veri alma hem de gönderme esnasında bir yol oluyor. Bilgisayardan devremize veri gönderilirken bu verilen ASCII tablosundaki sayı karşılıkları kullanılarak devremize byte tipinde gönderim sağlıyoruz. Örneğin devremize R harfini göndermek istiyoruz bu R harfinin ASCII tablosundaki sayısal değer karşılığı 82dir.

Bu sayıyı Binary olarak 0 ve 1ler ile karşılıklı olarak alıp gönderebiliyoruz.



Gönderdiğimiz veriler devremizde 128 byte lık (tampon bölge) sınırlı bir hafızada okunana kadar tutulur. Veri aktarımını sağlamak için yaygın ve popüler olan UART protokolünü kullanmaktayız.

**UART İLETİŞİM NEDİR NASIL SAĞLANIR?**

UART – Universal Asynchronous Receiver/Transmitter (Evrensel Asenkron Alıcı/Verici) kelimelerinin baş harflerinden oluşmaktadır. Günümüzde neredeyse birçok mikro denetleyicide UART modülü bulunmaktadır. Evrensel olmasının sebebi iletişim özelliklerinin değiştirilebilir olmasından dolayı bu modül bulunan bütün cihazlarda kullanılabilmesidir. Asenkron olması ise iletişim esnasında harici bir saat sinyaline ihtiyaç duymamasına dayanır. UART haberleşmede genel olarak 8 bit (1byte) veri iletimi kullanılmaktadır.

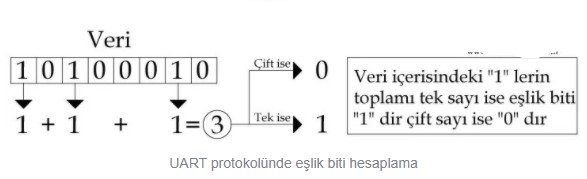


Protokol veri kalıbına baktığımız zaman 4 bölümden oluştuğunu görmekteyiz. Bu bölümleri kısaca açıklamak gerekir ise;

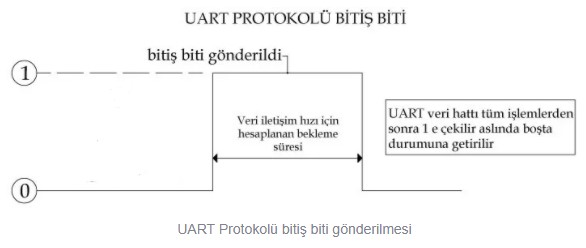
**Başlangıç Biti:** Senkronizasyon bitidir 1 bit başlangıç bitini, iletişimi başlatır hattı 1’den 0’a çeker.

**Veri:** Yukarıda bahsettiğimiz üzere 8 bit lik verimizdir.

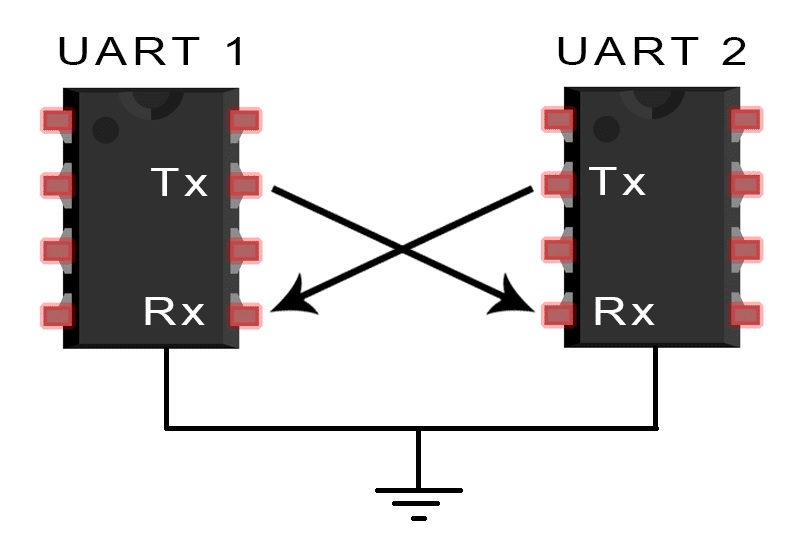
**Eşlik Biti:** 1 bit eşlik biti duruma göre 0 ya da 1 olur, verinin ulaştığı yerde hızlı olarak doğruluk kontrolü yapılmasını sağlar



**Bitiş Biti:** 1 bit bitiş biti başlangıç bitinin yaptığı şeyin tam tersini yapar 0 olan hattı 1’e çeker böylece iletişimin bittiği anlaşılır.



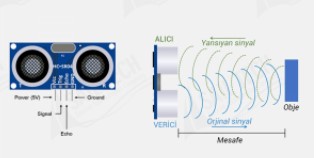
**UART Modülünün Bağlanması:**

Yaygın olarak birçok mikro denetleyicilerde olan UART modülü bağlaması sırasında 3 adet kabloya ihtiyacımızyvardır.

Tx, Rx ve GND olmak üzere 3 pin bulunmaktadır Tx ve Rx pinleri çapraz olarak bağlanır yani görsel dede gözüktüğü üzere Tx pini diğer UART modülünün Rx pinine bağlanırken Rx pini ise Tx pinine bağlanır GND pini ise diğer UART modülünün GND pinine bağlanır bunun neden böyle bağlanması gerektiğini açıklamak gerekirse, Tx ,Rx ve GND pinlerinin ne işe yaradığı anlatınca daha iyi anlayacaksınızdır. Tx pini veri gönderme Rx pini ise veri alma pinleridir. GND pini ise Şase voltaj farkını sıfırlamak amacıyla ortak bağlanır.

**HC-SR04 ÇALIŞMA PRENSİBİ**

HC-SR04 ultrasonik sensör sonar iletişim dalgalarını kullanarak karşısındaki nesneye olan mesafeyi hesaplayan bir devre donanımıdır. Sonar iletişim dalgasının cisme varıp geri yansıması ve kaynağa ulaşması olayları gerçekleşirken geçen süreyle mesafe hesaplama yapar.



HC SR04 üzerindeki pinler ve işlevleri:

VCC Pin: Sensör 5V güç kaynağı için kullanılır.

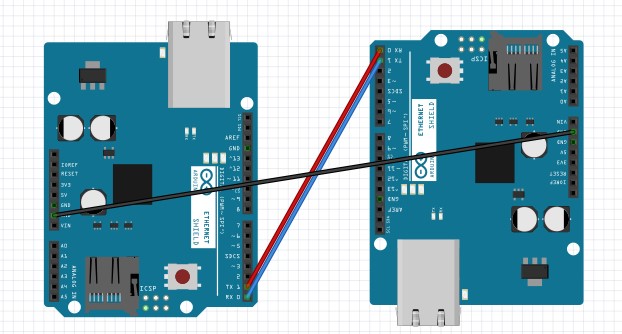
Trig Pin: Sensörün ses dalgalarını gönderen kısmıdır.

Echo Pin: Gönderilen ses dalgalarını alan kısımdır.

GND Pin: Topraklama.

**PROJE KISMI DEVRELERİN VE DONANIMLARIN BAGLANTISI**

İlk olarak iki devre arası bağlantıların yapılmasından başlayalım iki devre arası UART iletişimi sağlamak için bir devrenin Rx pininden diğer devrenın Tx pinileri arası kablo bağlantısı yapıyoruz bu çaprazlama bağlantısını yaptıktan sonra her iki devredede olan power bölümündeki GND pinleri arası bağlantısı da şekilde gözüktüğü gibi yapıyoruz.



1.VE 2. DEVRE

Bu yaptığımız devreye şimdi HC-SR04 sensörümüzü ve ledlerimizi bağlayalım.

**1.DEVRE BAGLANTILARI**

Öncelikle 1. Devremizin led bağlantılarını yapalım bunun için öncelikle bir led alalım bu ledimizi 1. Devremizdeki 13.girişine ledin uzun bacağı 13. Girişin hemen yanındaki GND pinine ise ledimizin kısa bacağına bağlantısını sağlıyoruz. Bu bağladığımız led 2. Devreden gelen mesafe verisi 10cm ve altında olduğu zaman bize uyarı verecek yani yanıp sönecek. 1. Devremize son olarak Bilgisayar da yapacagımız uygulama Form uygulaması ile bağlantı saglandıgını ve veri akışının gerçekleştiğini göstermek için bir led daha bağlıyoruz bu ledimiz veri akışı sağlandıkça devamlı yanıcak bağlantı kesildiği zaman sönecek bu ledimizi ise uzun bacağı 1.devrenın 3. Girişine kısa bacağını ise power bölümündeki GND pinine bağlıyoruz ve bu şekilde 1. Devremize yapmamız gereken bağlantıları tamamlamış oluyoruz.

**2.DEVRE BAGLANTILARI**

Bu devremiz HC-SR04 sensörümüzün bağlı olduğu ve mesafenin ölçüleceği devre olacak.



Sensörümüzde 4 adet pin olduğunu görmekteyiz bu pinlerden

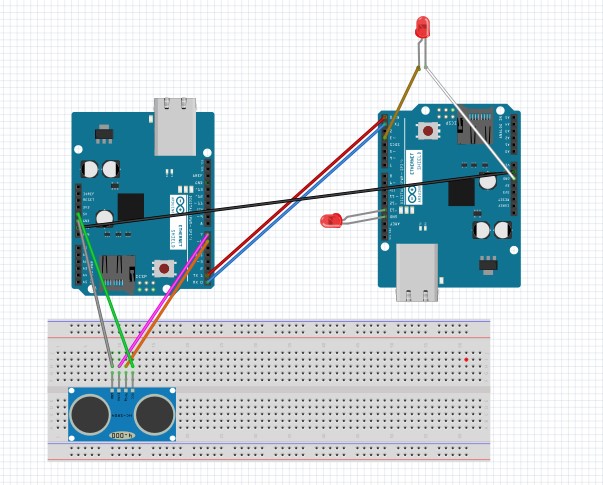
GND pinini 2. Devremizin power bölümünde olan GND pinine bağlıyoruz.

Echo pinini devremizde olan 7 girişe bağlıyoruz.

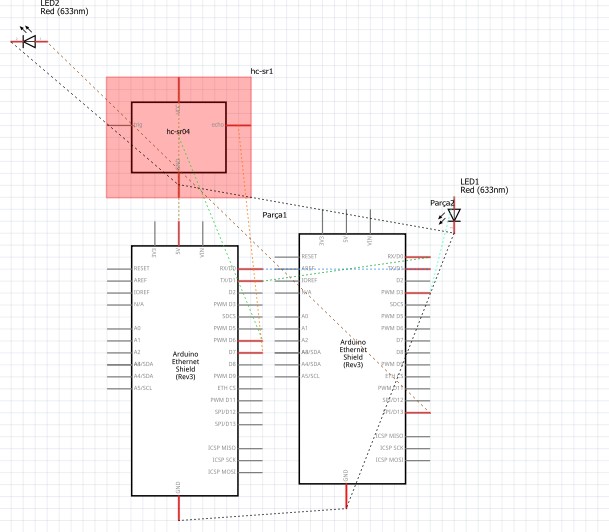
Trig pinini devremizde yer alan 6. Girişe bağlıyoruz.

Vcc pinini ise power bölümündeki 5V girişine bağlantısı yapıyoruz.

Bu şekilde 2. Devremizinde bağlantılarını tamamlamış oluyoruz ve devremizin görüntüsü aşağıdaki şekilde gözüktüğü gibi oluyor.



Devrenin şematik gösterimini merak edenler için aşağıdaki şekil gibi oluyor.



**ARDUNIO UNO DEVRELERİMİZİN KODLARI**

**Not: Devreye kodların yüklenmesi yapılırken devrenın Rx bağlantısını kesiniz.**

**1.DEVRE KODLARI**

int giris=0;

int cikisled=13;

int ata=0;

int led=3;

void setup() {

pinMode(cikisled, OUTPUT);

pinMode(led,OUTPUT);

pinMode(giris,INPUT);

Serial.begin(9600);// Seri haberleşmeyi başlatıyoruz (9600 bps hızında)

}

void loop() {

//Bu fonksiyon sonsuza kadar çalışır

//Arduino'nun yapması gereken işlemler buraya yazılır

digitalWrite(cikisled,LOW);

if(Serial.available()>0)//ardunıo devresıne gelen veri 0 dan büyükmü

{

ata=Serial.read();//gelen verileri oku ata degıskenınde ata

Serial.println(ata);//ata degişkenını ekrana yaz alt satısa geç

if(ata=='1'){

//ata degişkenı 1 ise ledi yak

digitalWrite(led,HIGH);

}else if(ata=='2'){

//ata degişkenı 2 ise ledi söndür

digitalWrite(led,LOW);

}

if(ata<10)//ata degişkeni 10 dan küçük ise led yak

{

digitalWrite(cikisled,HIGH);

delay(3);//3 ms bekleme yapar yanıp söner yani

}else

{ //ata degişkeni 10 dan büyükse led söner ve yanmaz

digitalWrite(cikisled,LOW);

delay(5);

}

}

}

**2.DEVRE KODLARI**

int trigPin=6;

int echoPin=7;

long sure;

long uzaklik;

void setup() {

pinMode(trigPin,OUTPUT);

pinMode(echoPin,INPUT);

Serial.begin(9600);//Seri haberleşmeyi başlatıyoruz (9600 bps hızında)

}

void loop() {

digitalWrite(trigPin,LOW);

delayMicroseconds(5);

digitalWrite(trigPin,HIGH);

delayMicroseconds(10);

digitalWrite(trigPin,LOW);

sure=pulseIn(echoPin,HIGH);//sensörümüzden gönderilen dalganın gidiş geliş süresini sure degişkenine atıyoruz

uzaklik=sure/29.1/2; //süre degişkenini yanı gidiş geliş hızını kullanarak formül yardımı ile cm cinsine çevirip uzaklık degıskenıne atıyoruz

if(uzaklik>200){

//uzaklıgımızın 200 cm den fazla olmasını istemedım eger fazla ise 200 olsun dıye sınırlandırma yaptım

uzaklik=200;

}

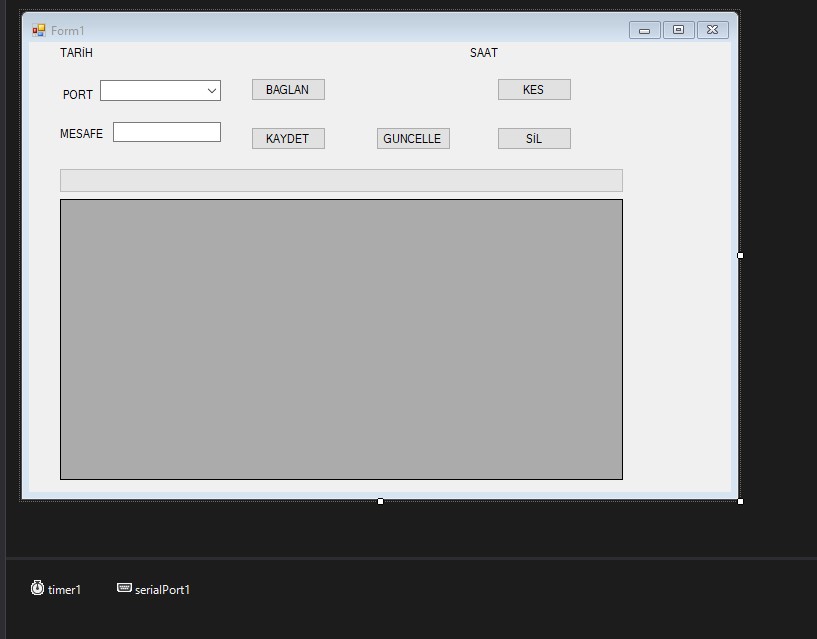
Serial.write(uzaklik);//serial write komutu ile uzaklık degerımızı UART baglantısı ile 1.devremıze gönderdim

delay(500);

}

**Windows Form Uygulaması Tasarımı ve Kodlaması**

**TASARIM KISMI:**



Tasarım kısmımızda gözüktüğü üzere 5 buton 4 label(biri tarih göstermek için diğeri saat diğer ikisi de port ve mesafe yazıları için) 1 adet progres bar bulunmakta görsellik açısından mesafe ne kadar uzaksa progres bar o kadar doluluk gösterecek 1 adet combobox bulunmakta bu comboboxda bilgisayarımıza bağlı olan com girişler olacak farklı seçeneklere bağlanabileceğiz ; bir adet textbox ımız bulunmakta bu textboxda 1.ardunıo devremizden gelen mesafe verileri gözükecektir son olarak Data grıdwiev bu araçla da veri tabanına kayıt ettiğimiz verileri görüp istediğimiz veriyi silebileceğiz. Ek olarak bilgisayar ve ardunıo devresı arası bağlantıyı sağlayabilmek için seriport aracını ve timer aracını formumuzun alt tarafına sürükleyip bırakıyoruz. (Veri tabanı olarak sql kullanacağız sql tablo içeriğinde tarih, saat ve mesafe sütunları olacaktır.)

**Windows Form Kodlama Kısmı**

İlk olarak ek olarak ikitane kütüphane eklememız gerekiyor.

*Bunlar: using System.Data.SqlClient;*

*using System.IO.Ports;*

kütüphaneleridir.

**KODLARIN TAMAMI**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

using System.IO.Ports;

namespace VERI

{

public partial class Form1 : Form

{

SqlConnection baglanti = new SqlConnection(@"Data Source=DESKTOP-RLCKHNM\SQLEXPRESS;Initial Catalog=veri;Integrated Security=True;Pooling=False");

int a = 0;

string gelen = "0";

SerialPort serialport;

public Form1()

{

InitializeComponent();

serialport = new SerialPort();

serialport.BaudRate = 9600;

comboBox1.Text = "COM3";//program başlar başlamaz com3 ile baglanmayı denıycek olmassa farklı port seçebiliriz comboboxdan

serialPort1.PortName = comboBox1.Text;

serialPort1.Open();//seri port baglantımızı açıyoruz

serialPort1.Write("1");

//baglantı saglandıgı zaman serialport1.write komutu ile devremıze 1 göneriyoruz

//bu şekilde 1 degerını alan devre ledımızı yakıcak buda baglantı saglandıgını göstericek

timer1.Start();

}

private void Form1\_Load(object sender, EventArgs e)

{

string[] ports = SerialPort.GetPortNames(); //Seri portları diziye ekleme

foreach (string port in ports)

comboBox1.Items.Add(port); //Seri portları comboBox1'e ekleme

guncelle();

}

private void button1\_Click(object sender, EventArgs e)

{

//baglan butonumuzun kodları

try

{

serialPort1.PortName = comboBox1.Text;

serialPort1.Open();//seri port baglantımızı açıyoruz

serialPort1.Write("1");//baglantı saglandıgı zaman serialport1.write komutu ile devremıze 1 göneriyoruz

//bu şekilde 1 degerını alan devre ledımızı yakıcak buda baglantı saglandıgını göstericek

timer1.Enabled = true;

}

catch

{

MessageBox.Show("Seri Port Hatası!");

}

}

private void button2\_Click(object sender, EventArgs e)

{

serialPort1.Write("2");

//baglantı saglandıgı zaman serialport1.write komutu ile devremıze 2 göneriyoruz

//bu şekilde 1 degerını alan devre ledımızı sönecek buda baglantı kesıldıgını göstericek

timer1.Stop();

serialport.Close();//seri port baglantımızı kesıyoruz

}

private void timer1\_Tick(object sender, EventArgs e)

{

try

{

gelen = serialPort1.ReadLine();

//seri port ıle devremızden gelen mesafe degerını gelen degişkenıne atıyoruz.

a = Convert.ToInt32(gelen);

if (a >= 0 && a < 200)

{

//gelen mesafe verisini textboxa yazıyoruz

textBox1.Text = gelen;

}

progressBar1.Value = a;//gelen verı degerını progres bara atıyoruz bu şekilde görsel olarak doluluk gosterıyor

label3.Text = DateTime.Now.ToLongDateString();//tarihi labele yazdırıyoruz

label4.Text = DateTime.Now.ToLongTimeString();//saati labele yazıyoruz

}

catch (Exception) { }

}

private void guncelle()

{//guncel verı tabanı bılgılerıı grıd view e çagırıyoruz

try

{

baglanti.Open();

string kayit = "SELECT \* from VERİ";

//veri tablosundaki tüm kayıtları çekecek olan sql sorgusu.

SqlCommand komut = new SqlCommand(kayit, baglanti);

//Sorgumuzu ve baglantimizi parametre olarak alan bir SqlCommand nesnesi oluşturuyoruz.

SqlDataAdapter da = new SqlDataAdapter(komut);

//SqlDataAdapter sınıfı verilerin databaseden aktarılması işlemini gerçekleştirir.

DataTable dt = new DataTable();

da.Fill(dt);

//Bir DataTable oluşturarak DataAdapter ile getirilen verileri tablo içerisine dolduruyoruz.

dataGridView1.DataSource = dt;

//Formumuzdaki DataGridViewin veri kaynağını oluşturduğumuz tablo olarak gösteriyoruz.

baglanti.Close();

}catch(Exception hata)

{

MessageBox.Show(Convert.ToString( hata));

}

}

private void ekle()

{

//kaydetme kodlarımız olculen mesafe degerını ve tarih ve saatı verı tabanına kaydedıyoruz

try

{

baglanti.Open();

SqlCommand veriekle = new SqlCommand("insert into VERİ (TARIH,SAAT,MESAFE) values (@TRH,@ST,@MSF)", baglanti);

veriekle.Parameters.AddWithValue("@TRH", label3.Text);

veriekle.Parameters.AddWithValue("@ST", label4.Text);

veriekle.Parameters.AddWithValue("@MSF", textBox1.Text);

veriekle.ExecuteNonQuery();

baglanti.Close();

guncelle();

}catch(Exception hata)

{

MessageBox.Show(hata.ToString());

}

}

private void button3\_Click(object sender, EventArgs e)

{

//kaydet butonumuza ekle fonksiyonunu çagırıyouz.

ekle();

}

private void button5\_Click(object sender, EventArgs e)

{ //kayıt sil butonumuzun kodları

try

{

for (int i=0;i<dataGridView1.SelectedRows.Count;i++)

{

baglanti.Open();

SqlCommand kmt = new SqlCommand("delete from VERİ where SAAT='"+dataGridView1.SelectedRows[i].Cells["SAAT"].Value.ToString()+"'",baglanti);

kmt.ExecuteNonQuery();

baglanti.Close();

}

MessageBox.Show("KAYITLAR SİLİNDİ");

guncelle();

}catch(Exception hata)

{

MessageBox.Show(hata.ToString());

}

}

private void button4\_Click(object sender, EventArgs e)

{//guncel butonu ile verı tabanı bılgılerıı grıd view e çagırıyoruz

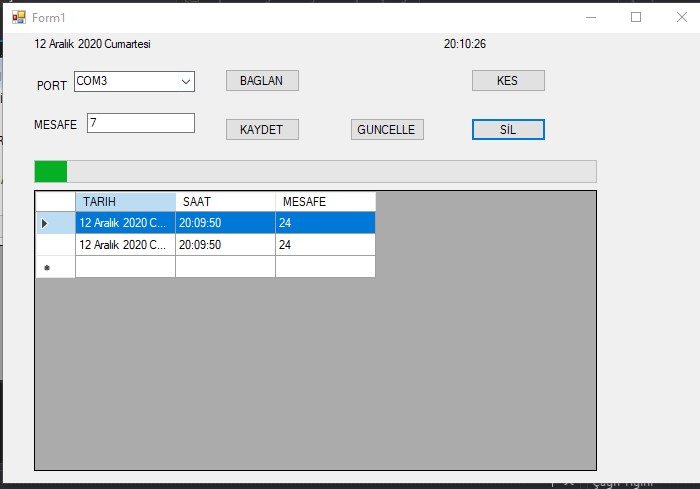
guncelle();

}

}

}

**ÇALIŞMA SIRASINDA EKRAN GÖRÜNTÜLERİ**

****

