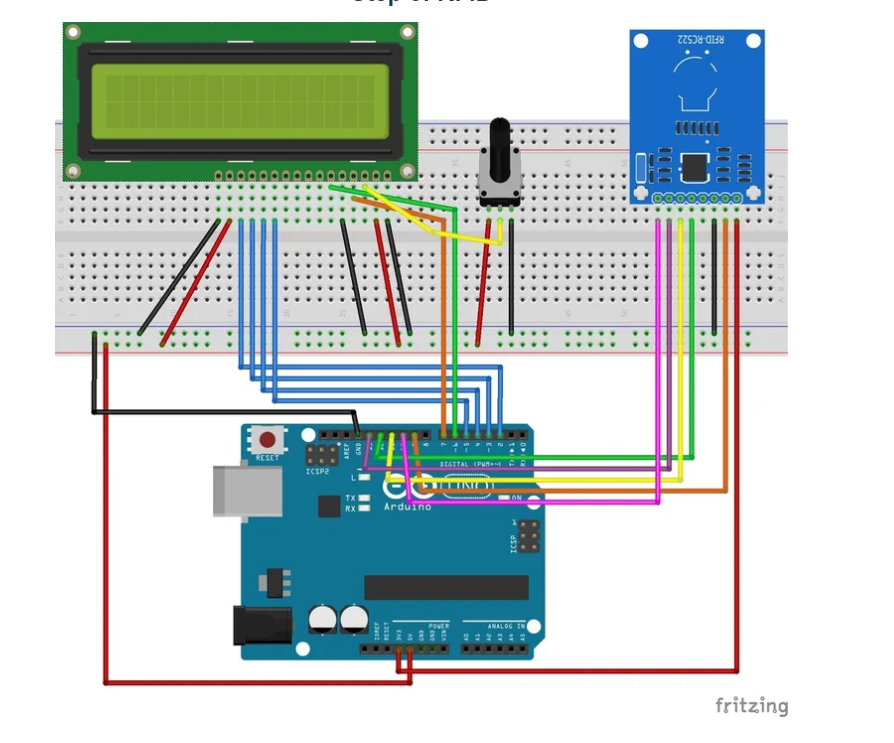
EK OLARRR

EK OLARAK : SERVO MOTORUN SARI UCU A0 A KAHVERENGİ UCU – YE

KIRMIZI UCU + YA TAKILIYOR



ARDUİNO KODLARI

#include <Servo.h>

#include <LiquidCrystal.h>

#include <MFRC522.h>

#include <SPI.h>

int RST\_PIN=9;

int SS\_PIN=10;

int servopin=A0;

Servo motor;

MFRC522 rfid(SS\_PIN,RST\_PIN);

String b="";

String kapi="";

LiquidCrystal lcd(7,6,5,4,3,2);

void setup()

{

motor.attach(servopin);

Serial.begin(9600);

SPI.begin();

rfid.PCD\_Init();

lcd.begin(16,2);

lcd.print("Hosgeldiniz!");

motor.write(0);

}

void loop() {

if (Serial.available()>0){

kapi = Serial.readString();

int YERi = kapi.indexOf("ac");

if (YERi == 0000)

{

motor.write(180);

}

else{

motor.write(0);

}

}

if(!rfid.PICC\_IsNewCardPresent())

return;

if(!rfid.PICC\_ReadCardSerial())

return;

if (!Serial.available()){

ekranaYazdir();

b = Serial.readString();

int YER = b.indexOf("1");

if (YER == 0)

{

lcd.clear();

lcd.print("Giris Basarili");

}

else{

lcd.clear();

lcd.print("Hatali Giris");

}

}

rfid.PICC\_HaltA();

}

void ekranaYazdir(){

for(int sayac=0; sayac<4; sayac++){

Serial.print(rfid.uid.uidByte[sayac]);

}

Serial.println("");

}

C# KODLARI

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.IO.Ports;

using System.Threading;

using System.Data.SqlClient;

using Microsoft.VisualBasic;

namespace atm

{

public partial class Form1 : Form

{

string[] ports = SerialPort.GetPortNames();//bilgisayara bağlı portları getiriyorum bir dizi içine atıyorum.

public Form1()

{

{

InitializeComponent();

serialPort1.BaudRate = 9600;

}

}

Thread saniye;

SqlConnection baglanti = new SqlConnection("Data Source=.;Initial Catalog=atm;Integrated Security=True");

void say()

{

int i = 5;

while (true)

{

label7.Text = i.ToString();

Thread.Sleep(1000);

i--;

if (i < 0)

{

if (serialPort1.IsOpen == true)

{

if(label10.Text=="Giriş yapabilirsiniz")

{

button1.Visible = true;

saniye.Abort();

}

else

{

serialPort1.Write("kapa");

label7.Text = string.Empty;

label8.Text = string.Empty;

label9.Text = string.Empty;

label10.Text = string.Empty;

kartno\_txt.Text = string.Empty;

kturubox.Text = string.Empty;

sifre\_txt.Text = string.Empty;

saniye.Abort();

}

}

else

{

serialPort1.Write("kapa");

label7.Text = string.Empty;

label8.Text = string.Empty;

label9.Text = string.Empty;

label10.Text = string.Empty;

kartno\_txt.Text = string.Empty;

kturubox.Text = string.Empty;

sifre\_txt.Text = string.Empty;

saniye.Abort();

}

}

}

}

void islemformu()

{

atmislem fm2 = new atmislem();

fm2.Show();

this.Hide();

}

void saniyebaslat()

{

Form1.CheckForIllegalCrossThreadCalls = false;

saniye = new Thread(new ThreadStart(say));

saniye.Start();

}

void portbaglanti()//COM4 portuna bağlantı sağlamak için kullanılan fonksyion

{

if (ports.Length > 1)

{

serialPort1.PortName = ports[1];

serialPort1.Open();

label4.ForeColor = Color.Green;

label4.Text = "(Aktif) Bağlantı COM4 var";

}

else

{

label4.ForeColor = Color.IndianRed;

label4.Text = "(Pasif) Bağlantı Yok";

}

}

void databasekontrol()

{

baglanti.Open();

SqlCommand cm = new SqlCommand("select count(\*) from kart where k\_no='"+kartno\_txt.Text+"' and k\_sifre='"+sifre\_txt.Text+"'and k\_turu='"+kturubox.Text+"' ", baglanti);

int ks = Convert.ToInt32(cm.ExecuteScalar());

cm.Dispose();

baglanti.Close();

if (ks > 0)

{

serialPort1.Write("1");

label10.ForeColor = Color.Green;

label10.Text = "Giriş yapabilirsiniz";

}

else

{

serialPort1.Write("0");

label10.ForeColor = Color.Red;

label10.Text = "Kart Türünüzü ve şifrenizi kontrol ediniz";

}

}

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

{

kturubox.Items.Add("Banka Kartı");

kturubox.Items.Add("Kredi Kartı");

portbaglanti();

kartno\_txt.Enabled = false;

button1.Visible = false;

}

private void okumabaslat\_btn\_Click(object sender, EventArgs e)

{

if (kturubox.Text == string.Empty || sifre\_txt.Text == string.Empty)

{

MessageBox.Show("Lütfen Diğer alanları doldurunuz");

}

else

{

if (serialPort1.IsOpen == true)

{

DialogResult mesaj = MessageBox.Show("Kart okutmak istediğinizden Emin misiniz?", "Bilgilendirme Kutucuğu", MessageBoxButtons.YesNo);

if (mesaj == DialogResult.Yes)

{

kartno\_txt.Enabled = true;

kartno\_txt.ReadOnly = true;

serialPort1.Write("ac");

label8.ForeColor = Color.YellowGreen;

label8.Text = "Kart okutulması bekleniyor....";

string kartno = serialPort1.ReadLine();

kartno\_txt.Text = kartno;

label8.ForeColor = Color.Green;

label8.Text = "Kart Okutuldu :)";

databasekontrol();

saniyebaslat();

}

else

{

MessageBox.Show("kart okutma işlemi yapamıyorum yapmam için lütfen evet seçeneğini seçiniz..");

}

}

else

{

MessageBox.Show("Lütfen Bağlantınızı Kontrol Ediniz");

}

}

}

private void kturubox\_TextChanged(object sender, EventArgs e)

{

if (kturubox.Text == "Banka Kartı" || kturubox.Text == "Kredi Kartı")

{

label9.Text = "";

}

else if(kturubox.Text == string.Empty){

label9.Text = "";

}

else

{

label9.ForeColor = Color.Red;

label9.Text = "Lütfen varsayılan kart türleri dışında başka kart türü eklemeyiniz";

kturubox.Text = string.Empty;

}

}

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

{

islemformu();

}

}

}

FORM GÖRÜNÜMÜ

