**PROJE ÖZETİ**

Projede toprak nem sensörü ile nem miktarına göre verilmesi gereken suyu kullanıcının telefonuyla yönlendirilmesi sonucu basacak olan motorun kontrolü Arduino ile yapılması ve kullanıcının telefonuyla nem miktarını kontrol edip çalıştırılması amaçlanmıştır.Aynı zamanda lcd ekran ile nem miktarının son durumu ve su gerekli veya gereksiz ise kullanıcıya bilgi vermektedir.

Sulama işlevini sulama motoru ile, nemi toprak nem sensörü ile yapmaktadır. Motor devresi için motor sürücü kartı kullanılmıştır. Kullanıcının telefon ile verdiği sulama komutu sırasında buzzer ürettiği ses ile ve lcd ekranda yazanlar sayesinde sulama yapıldığını, kullanıcıya bilgi vermektedir.Kullanıcı telefonu ile ardiuno arasındaki bağlantıyı bluetooth modülü sağlamıştır.Devre harici olarak 9V'luk pil ile beslenmiştir ve açma kapama devresi kullanılmıştır.Aynı şekilde reset butonu ile resetleme işlemi sağlanmıştır.

1

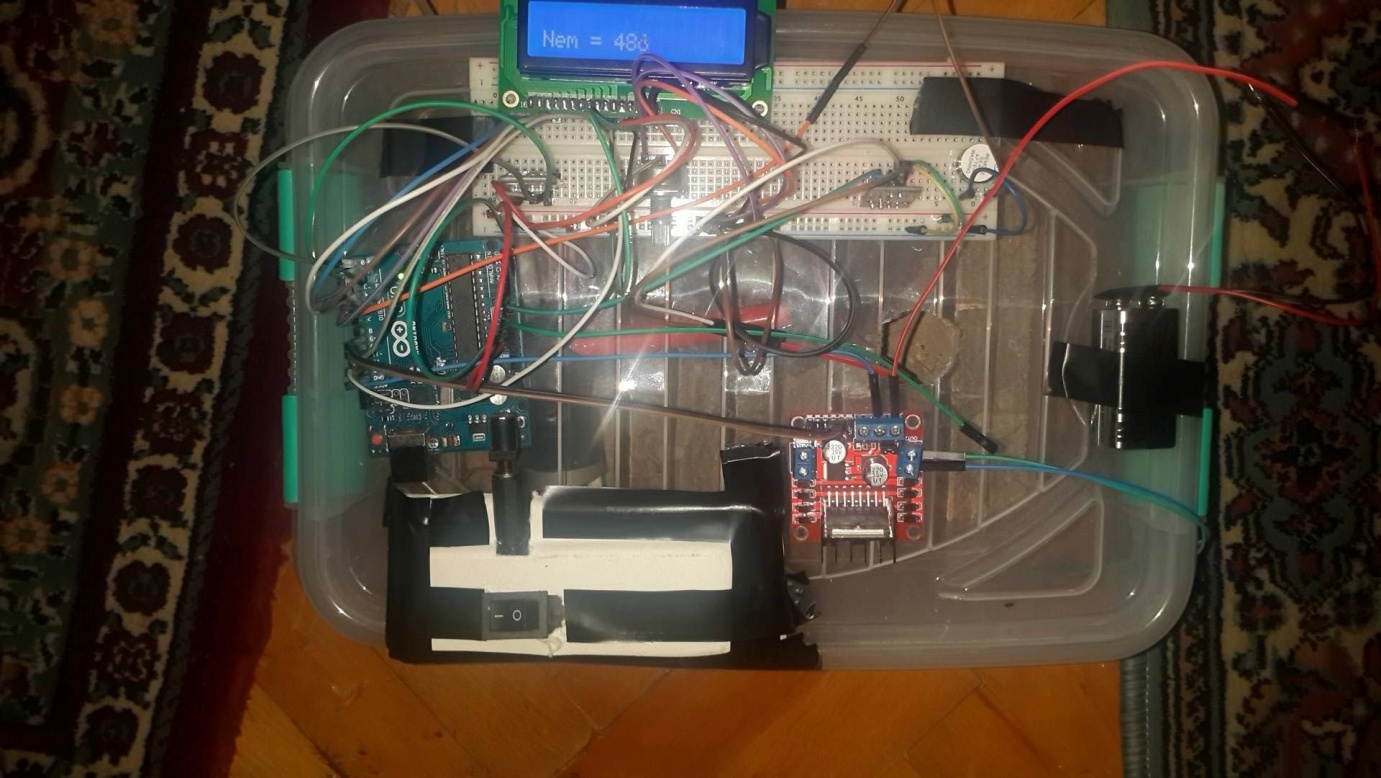
**Bluetooth Kontrollü Sulama Sisteminin**

**Donanım ve Yazılım Bilgileri**

Uygulama Adı :Bluetooth Kontrolü Sulama Sistemi

**Bluetooth Kontrolü Sulama Sistemi**

Projenin Üst Profilinden Görünümü:

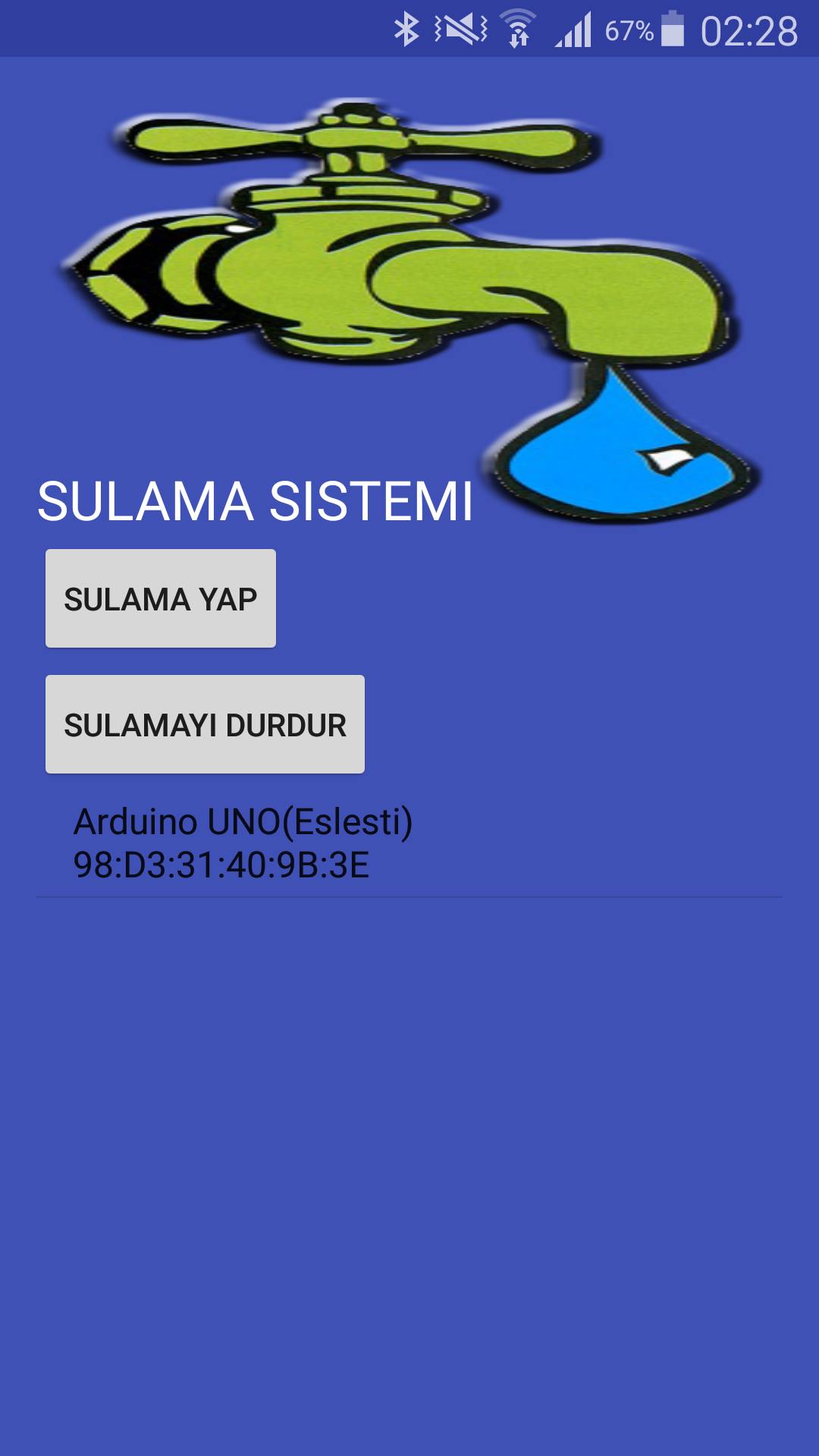


Projenin Yan Profilinden Görünümü:



2

Projenin Telefon Uygulaması Görünümü:



3

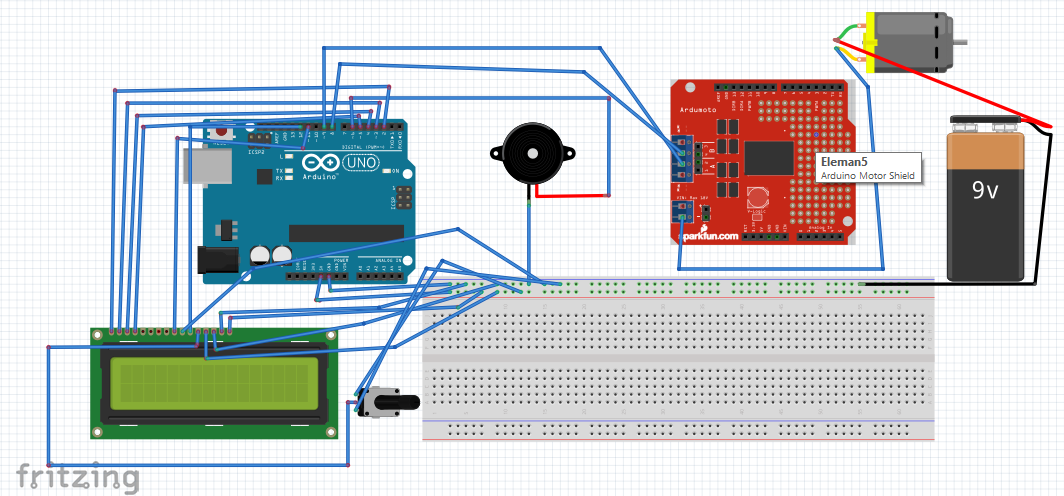
**Projede Kullanılan Parçalar ve Özellikleri**

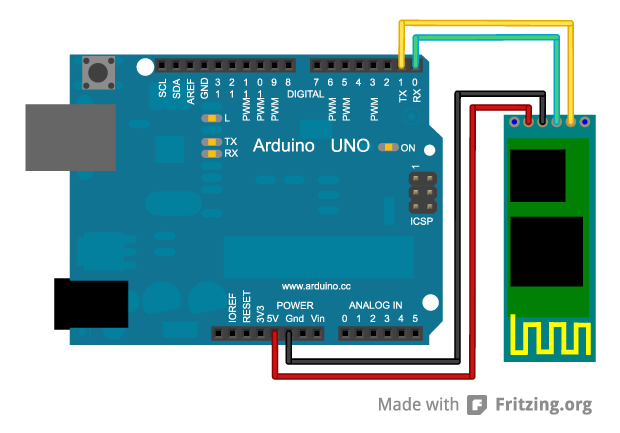
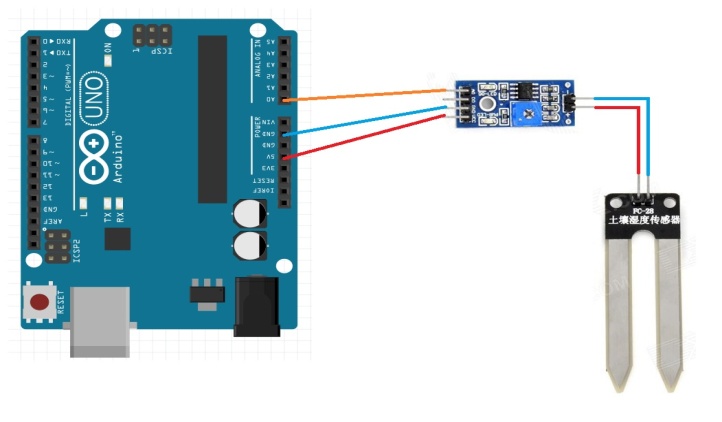
Bluetooth Kontrollü Sulama Sistemi uygulamasını gerçekleştirmek için, aşağıdaki parçalar gerekecektir.

|  |  |  |  |
| --- | --- | --- | --- |
| Şekil | Adı | Sayı | Açıklama |
| ArduinoUno_R3_Front_450px | Arduino Uno | 1 | Bluetooth modülü ile telefondan gelen değeri algılayıp motoru çalıştırmak, sensör ölçümünden gelen değeri lcd ekrana yazdırmak için kullanıldı. |
| breadboard ile ilgili görsel sonucu | Breadboard | 1 | Bluetooth modül devresi, motor sürücü kartı ve devresi, lcd devresi, buzzer devresi, toprak nem sensörü devresi için kullanıldı. |
| C:\Users\AYDENİZ\Desktop\Arduino-ws\University_homework\IşıkSensörü\indir.jpg | 10K Potansiyometre | 1 | Lcd ekran için kullanıldı. |
| C:\Users\Oem\Desktop\Smart-Electronics-5pcs-L298N-Stepper-DC-font-b-Motor-b-font-Driver-font-b-Shield-b.jpg | L293D Motor Sürücü | 1 | DC motoru ile suyu pompalamak için kullanıldı. |
| C:\Users\Oem\Desktop\013.jpg | DC Motor | 1 | Su pompası yapımında kullanıldı. |
| C:\Users\Oem\Desktop\indir.jpg | Toprak Nemi Algılama Sensörü | 1 | Toprak nemini ölçmek için kullanıldı. |
| C:\Users\Oem\Desktop\indir (1).jpg | 16x2 LCD Ekran | 1 | Topraktan gelen nem değerini kullanıcıya sunmak ve su ihtiyacını kullanıcıya göstermek için kullanıldı. |
| C:\Users\Oem\Desktop\SH-HC-08-CC2541-font-b-Bluetooth-b-font-4-0-BLE-to-UART-Transceiver-font.jpg | HC-05 Bluetooth Modülü | 1 | Kullanıcının sistemi telefon ile kontrol etmesi için kullanıldı. |
| C:\Users\Oem\Desktop\buzzer-nedir-ne-i--e-yarar.jpg | Buzzer | 1 | Sulama işlemi gerçekleşmesi sırasında kullanıcı için haber niteliğinde kullanıldı. |
| C:\Users\Oem\Desktop\9V-300mAh-primatic-Ni-MH-battery-300x227.jpg | 9V Pil | 2 | Motorun çalışması için ve devreyi dışarıdan beslemek için kullanıldı. |
| C:\Users\Oem\Desktop\DreamClub-Rocker-font-b-Switch-b-font-2-font-b-Circuits-b-font-15A-250V-off.jpg | Anahtar | 1 | Devreyi çalıştırmak için kullanıldı. |
| C:\Users\Oem\Desktop\40-pin-ayrilabilen-disi-disi-f-f-jumper-kablo-200-mm-1977-77-B.jpg | Jumper Kablo | 25 | Gerekli bağlantılar için kullanıldı. |
| C:\Users\Oem\Desktop\1653.jpg | Buton | 1 | Devreyi resetlemek için kullanıldı. |

4

**Bluetooth Kontrollü Sulama Sistemi Fritzing Çizimi**

****

 HC-05 Bluetooth Modülü Toprak Nem Sensörü 5

**Bluetooth Kontrollü Sulama Sistemi Arduino Kodu**

#include <LiquidCrystal.h> /\* LCD kullanımı için kütüphane dahil edilmelidir. \*/

#include <SoftwareSerial.h>/\*Bluetooth kullanimi için kütüphane dahil edilmelidir.\*/

#include <AFMotor.h>/\*Motor sürücü için kütüphane dahil edilmelidir.\*/

SoftwareSerial Genotronex(0, 1); /\* RX, TX.\*/

int data;/\*Gerekli Tanımlamalar\*/

int in1=9;

int in2=8;

int buzzer=6;

char BluetoothData;

const int toprak\_deger=A0;

int olcum\_sonucu;

int kontrol = 0;

LiquidCrystal lcd(12, 11, 5, 4, 3, 2); /\* LCD'nin bağlandığı Arduino pinleri. \*/

void setup()

{

lcd.begin(16, 2); /\* Kullandığımız LCD'nin sütun ve satır sayısını belirtmeliyiz.\*/

pinMode(6,OUTPUT);/\*Gerekli pin çıkışları\*/

pinMode(in1,OUTPUT);

pinMode(in2,OUTPUT);

Genotronex.begin(9600);/\*Seri iletişimi başlatır.\*/

Serial.begin(9600);

}

void loop()

{

if (Genotronex.available())/\*Bluetooth nesne kontrolü.\*/

{

BluetoothData=Genotronex.read();/\*Bluetooth değerini okuyoruz.\*/

if (BluetoothData=='1') /\*Sula butonuna basılırsa.\*/

6

{

digitalWrite(in1,HIGH);/\*Motorun dönme işlemi.\*/

digitalWrite(in2,LOW);

lcd.clear();/\*LCD temizleme.\*/

lcd.print("Sulama yapiliyor...");

digitalWrite(buzzer,HIGH);/\*Buzzer çalışması.\*/

kontrol = 1;/\*Nem aralığı için komtrol değeri.\*/

}

else if(BluetoothData=='0')/\*Sulamayı durdur butonuna basılırsa\*/

{

digitalWrite(in1,LOW);/\*Motoru durdurma işlemi\*/

digitalWrite(in2,LOW);

lcd.clear();/\*LCD temizleme.\*/

digitalWrite(buzzer,LOW);/\*Buzzer durdurulması.\*/

lcd.print("Sulama tamamlandi...");

kontrol = 0;/\*Nem aralığı için komtrol değeri.\*/

}

}

lcd.setCursor(0, 1); /\*İmlecin yeri 1. satır 0. sütun olarak ayarlandı.\*/

olcum\_sonucu=analogRead(toprak\_deger);/\*Analog pininden değer okunması.\*/

lcd.print("Nem = ");

lcd.print(olcum\_sonucu);/\*Sensörden okunan değer sonucunun yazdırılması.\*/

if(olcum\_sonucu<=500 && kontrol ==1)/\*Sulama için gerekli kontrol.\*/

{

lcd.clear();/\*LCD temizleme.\*/

lcd.print("Sulama yapmayiniz...");

}

else if(kontrol == 0 && olcum\_sonucu >= 700 )/\*Sulama için gerekli kontrol.\*/

{

lcd.clear();/\*LCD temizleme.\*/

lcd.print("Sulama yapiniz..."); }} 7

**Android Kodu**

Manifest kısmı için gerekli kodlar:

<uses-permission android:name="android.permission.BLUETOOTH\_ADMIN"/>  
<uses-permission android:name="android.permission.BLUETOOTH"/>

Main Activity kısmı için gerekli kodlar:

**package** com.example.oem.sulamasistemi;  
  
  
 **import** java.io.IOException;  
 **import** java.io.InputStream;  
 **import** java.io.OutputStream;  
 **import** java.util.ArrayList;  
 **import** java.util.Set;  
 **import** java.util.UUID;  
  
 **import** android.app.Activity;  
 **import** android.app.ProgressDialog;  
 **import** android.bluetooth.BluetoothAdapter;  
 **import** android.bluetooth.BluetoothDevice;  
 **import** android.bluetooth.BluetoothSocket;  
 **import** android.content.BroadcastReceiver;  
 **import** android.content.Context;  
 **import** android.content.Intent;  
 **import** android.content.IntentFilter;  
 **import** android.os.Bundle;  
 **import** android.os.Handler;  
 **import** android.os.Message;  
 **import** android.util.Log;  
 **import** android.view.View;  
 **import** android.widget.AdapterView;  
 **import** android.widget.AdapterView.OnItemClickListener;  
 **import** android.widget.ArrayAdapter;  
 **import** android.widget.Button;  
 **import** android.widget.ListView;  
 **import** android.widget.TextView;  
 **import** android.widget.Toast;  
  
  
  
**public class** MainActivity **extends** Activity **implements** AdapterView.OnItemClickListener {  
 Button **btnOn**;  
 Button **btnOff**;  
 TextView **text**;  
 String **address**=**null**;  
 8

ListView **liste**;  
 **public static final int *baglanti***=0;  
 **public static final int *mesajoku***=1;  
 OutputStream **outStream**;  
 InputStream **instream**;  
 ArrayAdapter<String> **adaptorlist**;  
 ArrayList<String > **eslesen**;  
 ArrayList<BluetoothDevice> **aygitlar**;  
 IntentFilter **filtre**;  
 **private** BroadcastReceiver **receiver**;  
 **private** ProgressDialog **progress**;  
 BluetoothAdapter **myBluetooth** = **null**;  
 BluetoothSocket **btSocket** =**null**;  
 **private boolean isBtConnected**=**false**;  
 Set<BluetoothDevice> **devicearray**;  
 **static final** UUID ***myUUID*** = UUID.*fromString*(**"00001101-0000-1000-8000-00805F9B34FB"**);  
  
 Handler **mHandler** = **new** Handler(){  
 @Override  
 **public void** handleMessage(Message msg) {  
 *//* ***TODO Auto-generated method stub* super**.handleMessage(msg);  
 **switch**(msg.**what**){  
 **case *baglanti***:  
  
 ConnectedThread connectedThread = **new** ConnectedThread((BluetoothSocket)msg.**obj**);  
 Toast.*makeText*(getApplicationContext(), **"Baglandi"**,Toast.***LENGTH\_LONG***).show();  
 String s = **"successfully connected"**;  
 **break**;  
 **case *mesajoku***:  
 **byte**[] readBuf = (**byte**[])msg.**obj**;  
 String string = **new** String(readBuf);  
 Toast.*makeText*(getApplicationContext(), string, Toast.***LENGTH\_LONG***).show();  
 **break**;  
 }  
 }  
 };  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
 **liste**=(ListView)findViewById(R.id.***lv***);  
 **liste**.setOnItemClickListener(**this**);  
 **adaptorlist**=**new** ArrayAdapter<String>(**this**,android.R.layout.***simple\_list\_item\_1***,0);  
 9

**liste**.setAdapter(**adaptorlist**);

**myBluetooth**=BluetoothAdapter.*getDefaultAdapter*();  
 **eslesen**=**new** ArrayList<String>();  
 **filtre**=**new** IntentFilter(BluetoothDevice.***ACTION\_FOUND***);  
 **aygitlar** = **new** ArrayList<BluetoothDevice>();  
  
 **receiver** = **new** BroadcastReceiver() {  
  
 @Override  
 **public void** onReceive(Context context, Intent intent) {  
 String action = intent.getAction();  
 **if** (BluetoothDevice.***ACTION\_FOUND***.equals(action))  
 {  
 BluetoothDevice aygit = intent.getParcelableExtra(BluetoothDevice.***EXTRA\_DEVICE***);  
 **aygitlar**.add(aygit);  
  
 String s=**""**;  
 **for** (**int** i=0;i<**eslesen**.size();i++)  
 {  
 **if** (aygit.getName().equals(**eslesen**.get(i))){  
 s=**"(Eslesti)"**;  
 **break**;  
 }  
 }  
 **adaptorlist**.add(aygit.getName()+**""**+s+**"\n"**+aygit.getAddress());  
 }**else if** (BluetoothAdapter.***ACTION\_STATE\_CHANGED***.equals(action))  
 {  
 **if** (**myBluetooth**.getState()==**myBluetooth**.***STATE\_OFF***){  
 bltac();  
 }  
 }  
 }  
 };  
 registerReceiver(**receiver**,**filtre**);  
 **filtre** = **new** IntentFilter(BluetoothAdapter.***ACTION\_STATE\_CHANGED***);  
 registerReceiver(**receiver**,**filtre**);  
 **if** (**myBluetooth**==**null**)  
 {  
 **text**.setText(**"Bluetooth yok"**);  
 Toast.*makeText*(getApplicationContext(),**"Bluetooth yok"**,Toast.***LENGTH\_SHORT***).show();  
 }**else** {  
 **if** (**myBluetooth**.isEnabled())  
 {  
 bltac();  
 }  
 secim();  
 10

startDiscovery();  
 **btnOn**=(Button)findViewById(R.id.***button1***);  
 **btnOn**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 **try** {  
 sensor();  
 } **catch** (IOException e) {  
  
 }  
 }  
 });  
 **btnOff**=(Button)findViewById(R.id.***button2***);  
 **btnOff**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 **try**{  
 sula();  
 }**catch** (IOException e)  
 {  
  
 }  
 }  
 });  
 }  
  
 }  
  
 **private void** bltac(){  
 Intent acintent = **new** Intent(BluetoothAdapter.***ACTION\_REQUEST\_ENABLE***);  
 startActivityForResult(acintent, 1);  
 }  
 **private void** secim(){  
 **devicearray**=**myBluetooth**.getBondedDevices();  
 **if**(**devicearray**.size()>0)  
 {  
 **for** (BluetoothDevice aygit:**devicearray**){  
 **eslesen**.add(aygit.getName());  
 }  
 }  
 }  
  
 **private void** startDiscovery(){  
 **myBluetooth**.cancelDiscovery();  
 **myBluetooth**.startDiscovery();  
 }  
  
  
 11

@Override  
 **protected void** onActivityResult(**int** requestCode, **int** resultCode, Intent data)   
 {  
 **super**.onActivityResult(requestCode, resultCode, data);  
 **if** (resultCode==***RESULT\_CANCELED***)  
 {  
 Toast.*makeText*(getApplicationContext(),**"Bluetoothu aciniz."**,Toast.***LENGTH\_SHORT***).show();  
 finish();  
 }  
 }  
 **public void** onItemClick(AdapterView<?> arg0,View arg1, **int** arg2, **long** arg3)  
 {  
 **if** (**myBluetooth**.isDiscovering())  
 {  
 **myBluetooth**.cancelDiscovery();  
 }  
 **if** (**adaptorlist**.getItem(arg2).contains(**"Eslesti"**))  
 {  
 BluetoothDevice secilenaygit = **aygitlar**.get(arg2);  
 ConnectThread connect= **new** ConnectThread(secilenaygit);  
 connect.start();  
 }**else**{  
 Toast.*makeText*(getApplicationContext(),**"Bluetootha baglanamadi."**,Toast.***LENGTH\_SHORT***).show();  
 **text**.setText(**"Baglanamadi"**);  
 }  
 }  
  
 **private class** ConnectThread **extends** Thread {  
  
 **private final** BluetoothSocket **mmSocket**;  
 **private final** BluetoothDevice **mmDevice**;  
  
 **public** ConnectThread(BluetoothDevice device) {  
  
 BluetoothSocket tmp = **null**;  
 **mmDevice** = device;  
  
 **try** {  
  
 tmp = device.createRfcommSocketToServiceRecord(***myUUID***);  
 } **catch** (IOException e) {  
  
 }  
 **mmSocket** = tmp;  
 }  
  
 12

**public void** run() {  
 **myBluetooth**.cancelDiscovery();  
  
 **try** {  
  
 **mmSocket**.connect();  
  
 } **catch** (IOException connectException) {  
  
 **try** {  
 **mmSocket**.close();  
 } **catch** (IOException closeException) { }  
 **return**;  
 }  
  
 **mHandler**.obtainMessage(***baglanti***, **mmSocket**).sendToTarget();  
 }  
  
  
 **public void** cancel() {  
 **try** {  
 **mmSocket**.close();  
 } **catch** (IOException e) { }  
 }  
 }  
  
 **private class** ConnectedThread **extends** Thread {  
  
 **private final** BluetoothSocket **mmSocket**;  
 **private final** InputStream **mmInStream**;  
  
 **public** ConnectedThread(BluetoothSocket socket) {  
 **mmSocket** = socket;  
 InputStream tmpIn = **null**;  
 OutputStream tmpOut = **null**;  
  
 **try** {  
 tmpIn = socket.getInputStream();  
 tmpOut = socket.getOutputStream();  
 } **catch** (IOException e) { }  
  
 **mmInStream** = tmpIn;  
 **outStream** = tmpOut;  
 }  
  
 **public void** run() {  
 **byte**[] buffer;

13

**int** bytes;  
 **while** (**true**) {  
 **try** {  
  
 buffer = **new byte**[1024];  
 bytes = **mmInStream**.read(buffer);  
  
 **mHandler**.obtainMessage(***mesajoku***, bytes, -1, buffer)  
 .sendToTarget();  
  
 } **catch** (IOException e) {  
 **break**;  
 }  
 }  
 }  
  
  
 **public void** cancel() {  
 **try** {  
 **mmSocket**.close();  
 } **catch** (IOException e) { }  
 }  
 }  
  
 **public void** sensor() **throws** IOException  
 {  
 **outStream**.write(**"1"**.getBytes());  
 }  
  
  
 **public void** sula() **throws** IOException  
 {  
 **outStream**.write(**"0"**.getBytes());  
 }  
  
}

14

**Bluetooth Kontrollü Sulama Sistemi Projesinde Kullanılan Kütüphaneler**

**LiquidCrystal Kütüphanesi**

LiquidCrystal.h arduinoda LCD ekran kullanabilmemiz için rehber bir kütüphanedir. Bu kütüphanenin içerdiği fonksiyonlar ve özellikler aşağıda açıklanmıştır.

[**LiquidCrystal()**](https://www.arduino.cc/en/Reference/LiquidCrystalConstructor)**:**Lcd’nin ayaklarını tanımlar ve setup içine yazılır.Lcd tipinde bir nesne tanımlayıp onu üzerinde işlem yaparız.Örnek kullanım: LiquidCrystal lcd(12,11,5,4,3,2);

[**begin()**](https://www.arduino.cc/en/Reference/LiquidCrystalBegin)**:** Liquidcrystal sınıfında tanımladığımız bir nesneyi başlatmaya yarar. Örn: lcd.begin(16,2);

[**clear()**](https://www.arduino.cc/en/Reference/LiquidCrystalClear)**:** Lcd ekranı temizlemeye yarar. Kullanımı: lcd.clear();

[**home()**](https://www.arduino.cc/en/Reference/LiquidCrystalHome)**:** Lcd ekrandaki imleci lcd nin en başına götürür. Lcd.home(); şeklinde kullanılır.

[**setCursor()**](https://www.arduino.cc/en/Reference/LiquidCrystalSetCursor)**:**Lcd de imleci istediğimiz yere konumladırır. Lcd.setCursor(0,1);//0. Satır 1. Sütun

[**write()**](https://www.arduino.cc/en/Reference/LiquidCrystalWrite)**:**Lcd ye bir karakter olarak veri yazar.lcd.write(‘a’);

[**print()**](https://www.arduino.cc/en/Reference/LiquidCrystalPrint)**:**Lcd kullanılırken en çok kullanılan fonksiyonlardandır.Ekrana string ifade yazmaya yarar. Lcd.print(‘Aydın Özyılmaz’)

[**cursor()**](https://www.arduino.cc/en/Reference/LiquidCrystalCursor)**:** Görünmeyen imleci görünür hale getirir. Lcd.cursor();

[**noCursor()**](https://www.arduino.cc/en/Reference/LiquidCrystalNoCursor)**:**Görünen imleci tekrar görünmez yapar.

[**blink()**](https://www.arduino.cc/en/Reference/LiquidCrystalBlink)**:**blink fonksiyonu ile yanıp sönen imleç elde ediyoruz. Lcd.blink();

[**noBlink()**](https://www.arduino.cc/en/Reference/LiquidCrystalNoBlink)**:**Yanıp sönen imleci tekrar normal hale getirir. B harfi büyüktür. Lcd.noBlink();

[**display()**](https://www.arduino.cc/en/Reference/LiquidCrystalDisplay)**:**Lcd yi açmaya yarar. Lcd zaten açık geleceği için kapatmadığımız sürece ihtiyacımız yoktur.

[**noDisplay()**](https://www.arduino.cc/en/Reference/LiquidCrystalNoDisplay)**:**Lcd yi kapatır.

[**scrollDisplayLeft()**](https://www.arduino.cc/en/Reference/LiquidCrystalScrollDisplayLeft)**:**Lcd üzerinde bulunan her yazıyı bir birim sola kaydırır. Böylelikle kayan yazı elde edilir.

[**scrollDisplayRight()**](https://www.arduino.cc/en/Reference/LiquidCrystalScrollDisplayRight)**:**Lcd üzerinde bulunan her yazıyı bir birim sağa kaydırır. Böylelikle kayan yazı elde edilir.

15

[**autoscroll()**](https://www.arduino.cc/en/Reference/LiquidCrystalAutoscroll)**:**Otomatik kaydırma işlemi yapar. Yeni yazılar geldikçe soldakiler kaybolur.

[**noAutoscroll()**](https://www.arduino.cc/en/Reference/LiquidCrystalNoAutoscroll)**:**Autoscroll özzelliğini kapatır.

[**leftToRight()**](https://www.arduino.cc/en/Reference/LiquidCrystalLeftToRight)**:**Standart olarak açık olan bu fonksiyon yazılarımızı sola sıfırlar.

[**rightToLeft()**](https://www.arduino.cc/en/Reference/LiquidCrystalRightToLeft)**:**Yazılarımızı sağa hizalar.

[**createChar()**](https://www.arduino.cc/en/Reference/LiquidCrystalCreateChar)**:**Lcd de özel karakter kullanma fonksiyondur.

# SoftwareSerial Kütüphanesi

Bilgisayarla seri haberleşme yaptığımız gibi, diğer elektronik elemanlarla da seri haberleşme yapabiliriz. Bunun için haberleşilecek elemanların Tx ve Rx uçlarını çapraz bir şekilde Arduino'nun Tx ve Rx pinlerine takmalıyız. Arduino UNO'da sadece bir çift Tx ve Rx (1. ve 0. pinler) bulunur. Bu pinler aynı zamanda USB üzerinden bilgisayarla haberleşmemizi sağlayan pinlerdir. Yani bilgisayarla haberleşme halinde bulunan Arduino'nun 0 ve 1. pinler kullanılamaz.

Arduino MEGA gibi gelişmiş kartlarda birden fazla Tx Rx çifti bulunduğu için bu cihazlar, hem harici olarak başka modüllerle seri haberleşebilirken hem de bilgisayara veri yollayabilir. "SoftwareSerial" kütüphanesi Arduino Uno gibi sadece bir çift Tx Rx pini bulunan kartlar için geliştirilmiştir. Bu kütüphane yardımıyla Arduino'nun diğer pinleri de Tx ve Rx olarak kullanılabilmektedir. Bu kütüphanenin fonksiyonları aşağıdadır.

[**SoftwareSerial**](https://www.arduino.cc/en/Reference/SoftwareSerialConstructor)**():**Kullanacağımız softwareserial nesnesinin rx tx pinlerini belirtiriz. Begin print gibi fonksiyonlar ile birlikte kullanılır . Örnek: Softwareserial myserial(0,1)//RX TX

[**available**](https://www.arduino.cc/en/Reference/SoftwareSerialAvailable)**():** Myserial portuna gelen bir mesaj varmı kontrol eder. Kullanımı **while**(myserial.available())

[**begin**](https://www.arduino.cc/en/Reference/SoftwareSerialBegin)**():**Seri Haberleşmeyi başlatır. Myserial.begin(9600);

[**isListening**](https://www.arduino.cc/en/Reference/SoftwareSerialIsListening)**():**Portta dinleme var mı yok mu kontrol eder.   if (myserial.isListening()) {  
   Serial.println("Port One is listening!");

[**overflow**](https://www.arduino.cc/en/Reference/SoftwareSerialOverflow)**():**Taşma değerini temizler.  if (portOne.overflow()) {  
   Serial.println("SoftwareSerial overflow!")

[**peek**](https://www.arduino.cc/en/Reference/SoftwareSerialPeek)**():**Sadece bir karakter alır. Alınan karkteri gönderme yapar.

[**read**](https://www.arduino.cc/en/Reference/SoftwareSerialRead)**():**char c = mySerial.read();

16

[**print**](https://www.arduino.cc/en/Reference/SoftwareSerialPrint)**():**Gelen okunan değeri yazar. Myserial.print();

[**println**](https://www.arduino.cc/en/Reference/SoftwareSerialPrintln)**():**Yazdıktan sonra satır başı yapar. Myserial.println();

[**listen**](https://www.arduino.cc/en/Reference/SoftwareSerialListen)**():**Dinleme var mı yok mu kontrol eder.

[**write**](https://www.arduino.cc/en/Reference/SoftwareSerialWrite)**():**Stringi byte değere dönüştürür**.** int bytesSent = mySerial.write(“hello”);

# AFMotor Kütüphanesi

Kütüphaneyi çağırmak için **#include <AFMotor.h>**yazabilir veya kütüphane ekle menüsünden çağırabilinir.

**Dc Motor Kontrol Fonksiyonları:**

**AF\_DCMotor :**Bu fonksiyon ile dc motor tanımı yapılır.

Dc motor sürmede iki husus vardır. Birincisi kanal numarası, ikincisi de pwm frekansıdır.

1 ve 2 nolu kanal için frekans tanımlamaları;

* MOTOR12\_1KHZ // 1khz frekans
* MOTOR12\_2KHZ // 2khz frekans
* MOTOR12\_8KHZ // 8khz frekans
* MOTOR12\_64KHZ // 64khz frekans

3 ve 4 nolu kanal frekans tanımlamaları;

* MOTOR34\_1KHZ // 1khz frekans
* MOTOR34\_8KHZ // 8khz frekans
* MOTOR34\_64KHZ  // 64khz frekans

Motorların tanımlarını iki şekilde yapabilirsiniz.

**AF\_DCMotor motor1(1):** Burada parantez içindeki sayı kanal numarasını verir.Dışındaki  “motor1” tanımı ise 1.kanala atadığımız isimdir Bu şekilde tanımladığımızda pwm frekansını girmediğimiz için varsayılan frekans değeri 1khz olarak tanımlanmış olacaktır.

**AF\_DCMotor left\_motor(1, MOTOR12\_64KHZ):**Burada yine 1. kanalı tanımladık. Kanal adını “left\_motor” olarak tanımladık. Bu tanımlamada frekans değerimizi de girdik.

Parantezde birinci kısım kanal adı ikinci kısım ise frekans değerimizdir.

17

**setSpeed Fonksiyonu:** Bu fonksiyon ile motorları 0 ile 255 değerler arasında hızını ayarlayabilirsiniz.

**motor1.setSpeed(180):**Bu şekilde motor1 isimli motor kanalına parantez içine girdiğimiz değer ile hızını belirlemiş olduk.

**run fonksiyonu:**Bu fonksiyon ile motorların ileri geri ve durma işlemini gerçekleştiriyoruz.

Fonksiyonda yön işlemleri ingilizce terimlerle sağlanıyor.

FORWARD – ileri  
BACKWARD  – geri  
RELEASE – dur

**motor1.run(FORWARD):**Bu şekilde “motor1” isimli kanalın motorun yönü ileri olarak ayarlandı.

**motor.run(BACKWARDS):**Geri git.  
**motor.run(RELEASE):**Dur.

**Step Motor Kontrol Fonksiyonları:**

**AF\_Stepper:**Bu fonksiyon step motor tanımı yapar.**AF\_Stepper motor\_adı(adım, kanal numarası)**

**motor\_adı**kısmı step motorunuza vereceğiniz isimdir.

**“adım”**kısmına step motorun  bir devirdeki adım sayısı girilir.

**“kanal numarası”**kısmına da step motoru hangi kanala bağlamak istiyorsak o kanalın numarası girilir. Yani sol kısım 1. kanal sağ kısım ise 2. kanal idi. 1 veya 2 girmemiz gerekir.

**“yön”**kısmına motorun dönme yönü girilir. Yön fonksiyonları “FORWARD” ve “BACKWARD” olarak tanımlanır.

**“mod”**kısmında step motorun hareket şekli girilir. Bunlar;

**SINGLE**: Bir bobine enerji  vermesiyle motoru çalıştırır. Güç tasarrufu gerektiren yerlerde kullanışlıdır fakat çok yaygın değildir. Motora az tork verir.

**DOUBLE :**İki bobine enerji vermesiyle motoru çalıştırır. Motora tam tork verir.

18

**INTERLEAVE :** Bobinlere eş zamanlı enerji vererek adım açısını yarıya düşürür. Bu sayede adım sayısı iki katına çıkar ve çözünürlük artar. Örneğin 200 adım 1.8 derece motor bu fonksiyon ile 0.9 derece 400 adım değerini alır.

**MICROSTEP :**Pürüzsüz motor sürüşü sağlar. Yaygın kullanılan fonksiyondur. Adımlararasında yumuşak geçiş sağlar. Fakat torku düşürür.

Örnek fonksiyon :

**Stepper1.step(150, FORWARD, DOUBLE):**Burada “**Stepper1**” isimli motorun 150adım ile “**DOUBLE**” modunda döneceği bilgisi girildi.

**setSpeed(rpm):**Step motorun rpm değerini ayarlamak için kullanılan fonksiyondur. Parantez içine rpm değeri girilir.

**onestep(yön, mod)**: Bu fonksiyon tek adım step motorlar için geçerlidir. Parantez içindeki yön ifadesine FORWARD veya BACKWARD yazılarak yön belirtilir. mod ifadesine de yukarıdaki aynı modları kullanarak tanımlamamızı yapıyoruz.

**release()**: Bu fonksiyon motoru durdurur ve torku tutar yani enerji kesmez ve motorun aynı torkta kalmasını sağlar.

19

**Bluetooth Kontrollü Sulama Sisteminde Projesinde Kullanılan Kaynaklar**

\* https://www.youtube.com/watch?v=kE0Vrtbw4Vw

\* https://www.youtube.com/watch?v=cQEomlhuyR8

\* https://www.youtube.com/watch?v=Nk80L7zp5\_4

\* http://www.robimek.com/

\* https://gelecegiyazanlar.turkcell.com.tr/konu/arduino/egitim/arduino-102/arduino-ile-lcd-ekran-kullanimi

\*http://www.gokhangol.com/tag/android-bluetooth/

20