**Proje Hakkında Açıklama Özeti:**

Teknolojinin gelişmesiyle evlerde kullanılan akıllı sistemlerin sayısı git gide artmakta. Bu sistemler günlük hayatta yaşadığımız sorunlara akılcı çözümler getirmek için geliştiriliyor ve geliştirilmeye devam ediyor. [Arduino](http://www.elektrikport.com/teknik-kutuphane/elektronik-ve-mikrodenetleyici-merakllarnn-yeni-gozdesi-arduino!/7917" \l "ad-image-0" \t "_blank) gibi ürünler bizlere yaşadığımız sıkıntılara karşı kendi çözümlerimizi yaratma şansı veriyor.

Bu projede uzaktan telefon kontrolü ile Arduino ile balıklarınızı nasıl besleyebileceğiniz işlenmekte. Balık beslenen evlerde sıkça yaşanılan yem verme , akvaryum su seviyesini ölçme ve akvaryum su sıcaklığını ölçme problemine getirebilecek bir çözüm ele alındı.

Balık yemleme makinesi balıklara telefon ile istediğiniz zaman yem verecek ve yem miktarını ayarlayacak. Dolu bir huni içerisine koyduğumuz tüpün kapağını açıp kapatacak. Kapak kısmını yönetmek için bir servo motor kullanılacak. Arduino servo motora bağlı olan kapağın ne kadar açık kalacağını ayarlayacak. Bu sistem sürekli açık kalması gerektiği için arduinoyu bir adaptör ile beslenmesi tavsiye edilir. Ayrıca kullanılacak olan sıcaklık sensörü ve su seviyesini ölçen sensör ile balıklar için sıcaklık seviyesi ve su derinlik seviyesi ölçülebilecek.

**Projenin Amacı:**

Projede telefon kontrolü ile; balık yemleme, su seviyesini ölçüp telefona yansıtma ve su seviyesinin yeterli olup olmadığını belirleyerek kullanıcıya uyarı vermesi amaçlanmaktadır. Bu kontroller bluetooth modülü ile sağlayarak kullanıcının telefonunun bluetooth ile bağlantısı sonucu dile getirilen işlemlerin yapılması amaçlanmaktadır. Bu sayede kullanıcı telefon kontrolü ile uzaktan belirli işlemleri gerçekleştirebilecek.

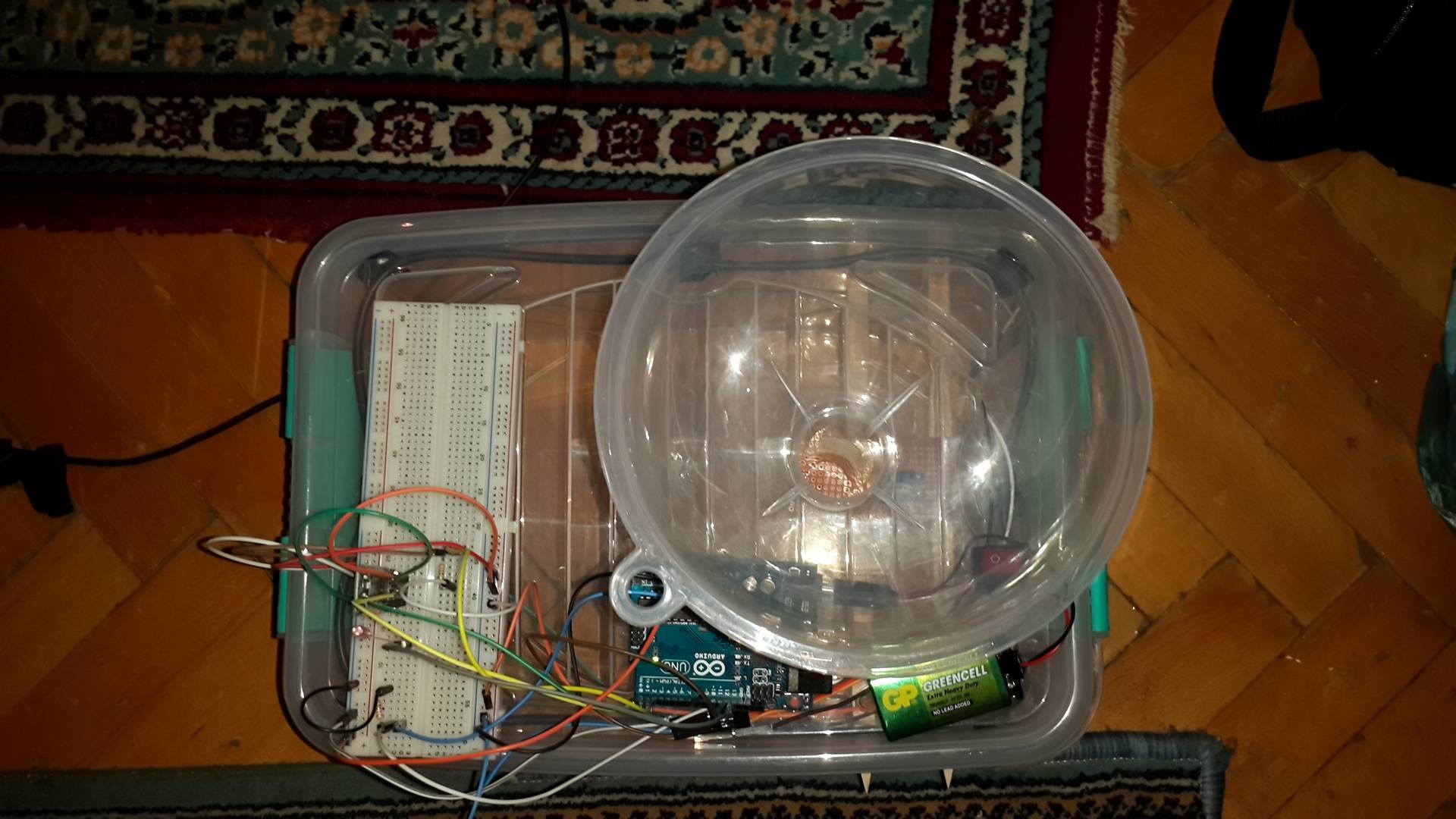
**Projenin Genel Çalışma Şekli:**

Proje kontrolü telefon ile sağlanmaktadır. Telefonumuz bluetooth modülüne bağlandıktan sonra menüde yemleme butonu, sıcaklık değeri ve su seviyesi alanları vardır. Kullanıcı su sıcaklık değerini görebilir ve suyun sıcaklığından haberdar olabilir. Aynı şekilde kullanıcı suyun seviyesinin kurulan devre yardımıyla yeterli olup olmadığını uygulamada görebilir. Yemleme işlemini ise yemle butonuna basarak bu işlemi gerçekleştirebilir.

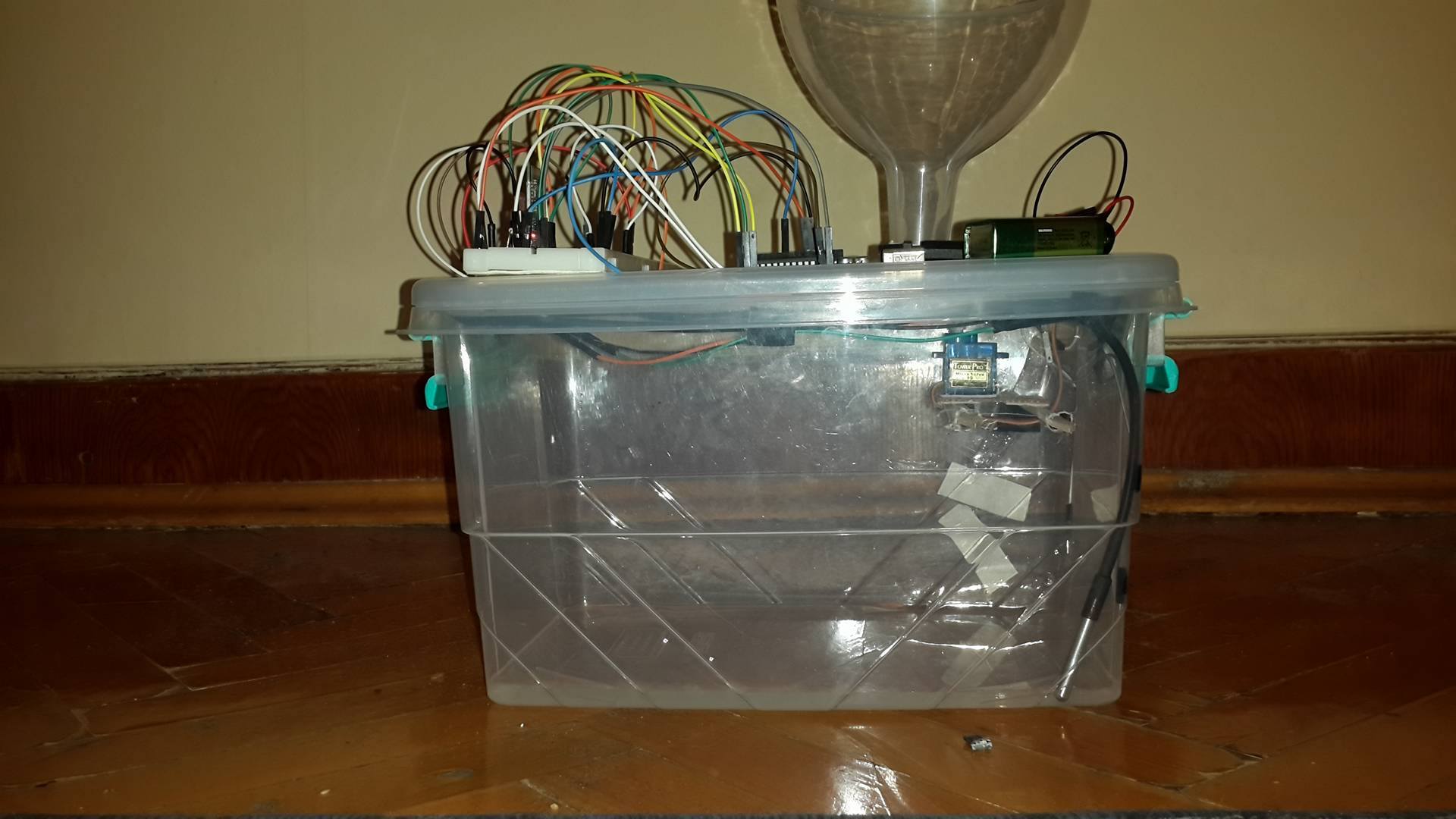
**Bluetooth Kontrollü Akvaryum Sisteminin** **Donanım ve Yazılım Bilgileri**

**Bluetooth Kontrolü Akvaryum Sistemi**

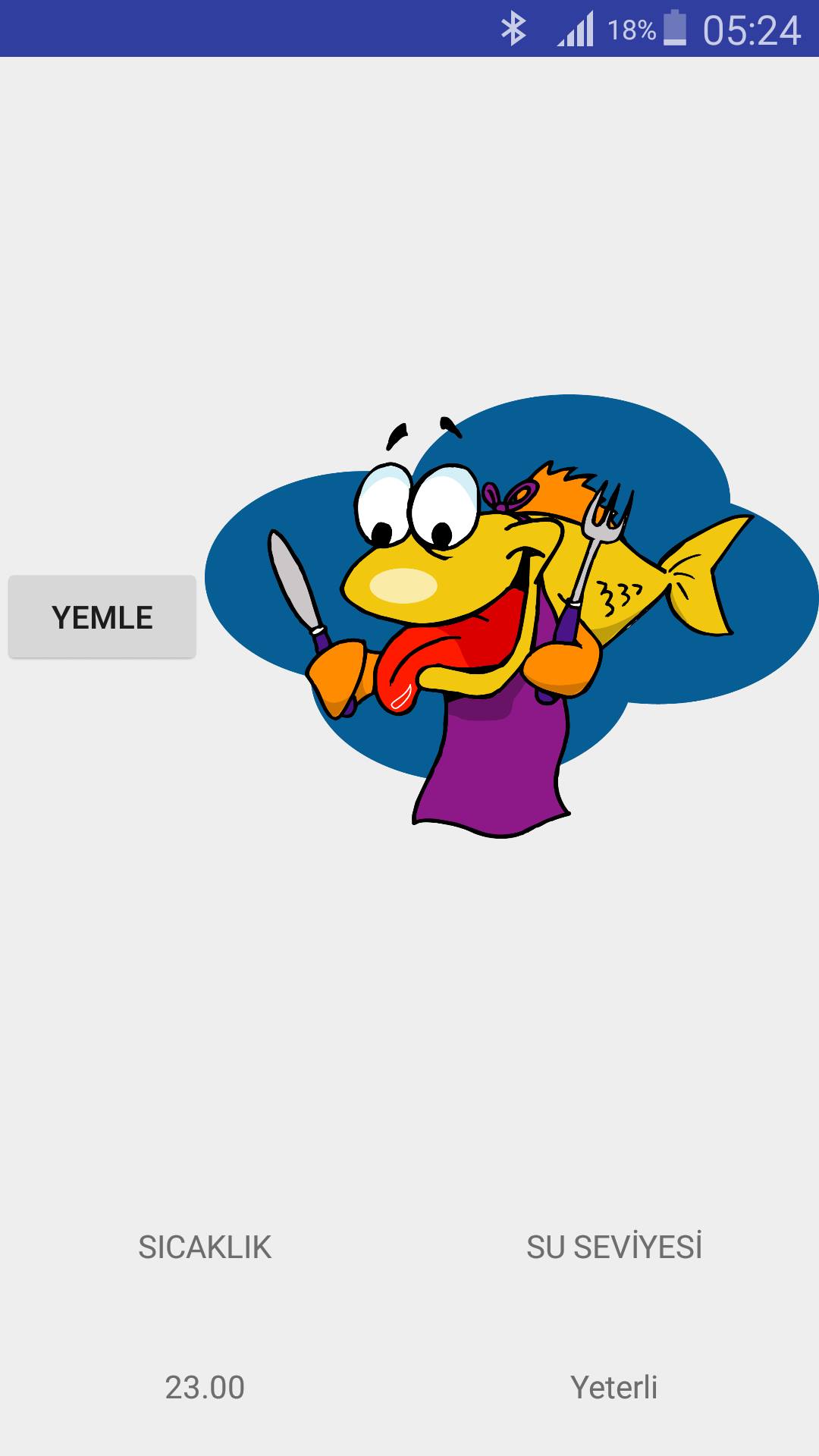
Projenin Üst Profilinden Görünümü:



Projenin Yan Profilinden Görünümü:



Projenin Telefon Uygulaması Görünümü:



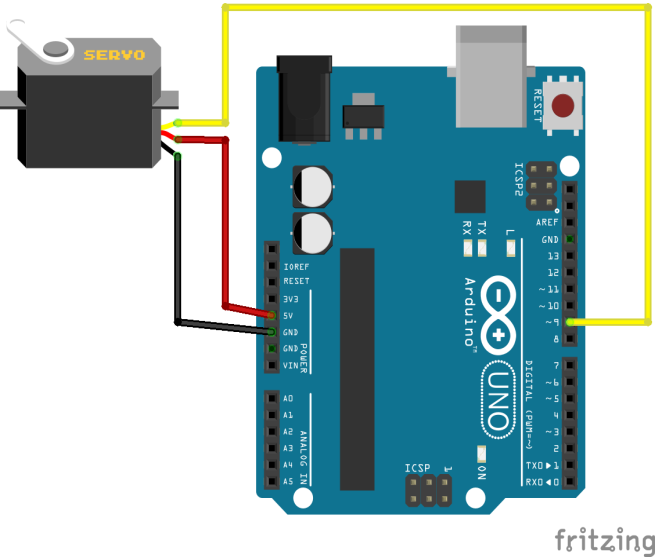
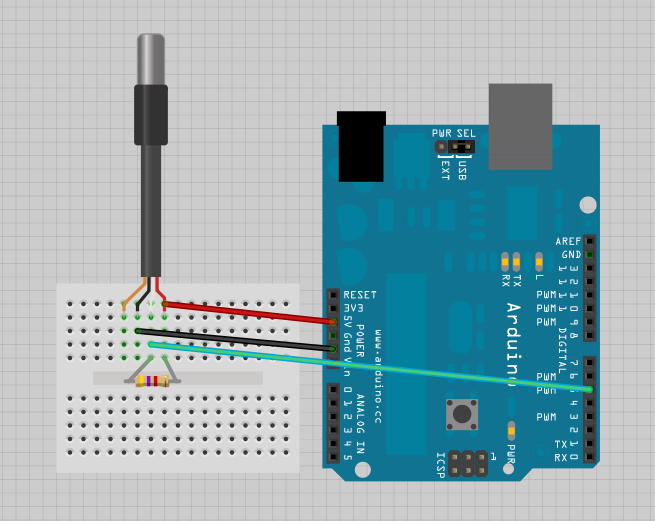
**Projede Kullanılan Parçalar ve Özellikleri**

Bluetooth Kontrollü Akvaryum Sitemi uygulamasını gerçekleştirmek için, aşağıdaki parçalar gerekecektir.

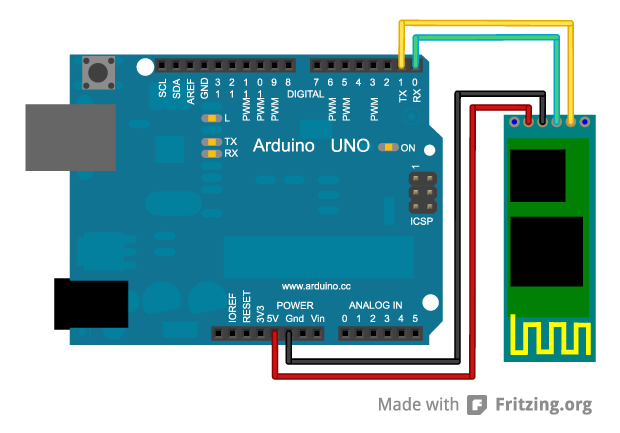
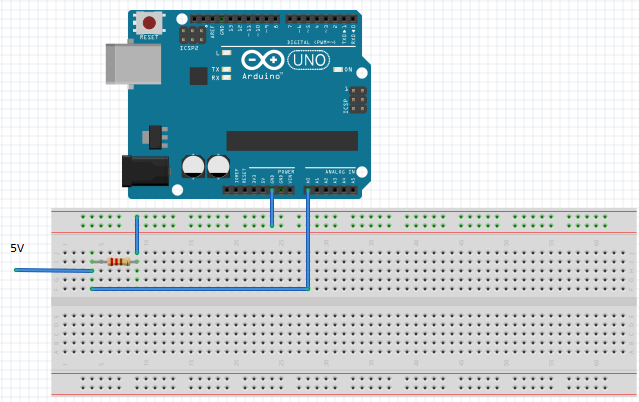
|  |  |  |  |
| --- | --- | --- | --- |
| Şekil | Adı | Sayı | Açıklama |
| ArduinoUno_R3_Front_450px | Arduino Uno | 1 | Bluetooth modülü ile telefondan gelen değeri algılayıp motoru çalıştırmak, sensör ölçümünden gelen değeri telefona yazdırmak için kullanıldı. |
| breadboard ile ilgili görsel sonucu | Breadboard | 1 | Bluetooth modül devresi, servo motor devresi, su seviyesi ölçüm devresi, buzzer devresi, sıcaklık sensörü devresi için kullanıldı. |
| C:\Users\Oem\Desktop\apcnews2012masterclass_arduino_enhanced_rolly_thumb_mainImage3.jpg3.jpg | Servo Motor | 1 | Yemleme yapılabilmesi için kullanıldı. |
| SH-HC-08-CC2541-font-b-Bluetooth-b-font-4-0-BLE-to-UART-Transceiver-font | HC-05 Bluetooth Modülü | 1 | Kullanıcının sistemi telefon ile kontrol etmesi için kullanıldı. |
| 9V-300mAh-primatic-Ni-MH-battery-300x227 | 9V Pil | 1 | Devreyi dışarıdan beslemek için kullanıldı. |
| DreamClub-Rocker-font-b-Switch-b-font-2-font-b-Circuits-b-font-15A-250V-off | Anahtar | 1 | Devreyi çalıştırmak için kullanıldı. |
| 40-pin-ayrilabilen-disi-disi-f-f-jumper-kablo-200-mm-1977-77-B | Jumper Kablo | 25 | Gerekli bağlantılar için kullanıldı. |
| C:\Users\Oem\Desktop\6470-big.jpg | DS18B20  Isı Sıcaklık Sensörü | 1 | Suyun sıcaklığını ölçebilmek için kullanıldı. |
| C:\Users\Oem\Desktop\direnc2.jpg | Direnç | 1 | Seviye ölçmede yardımcı eleman olarak kullanıldı. |

**Bluetooth Kontrollü Akvaryum Sistemi Fritzing Çizimi**

Her devre elemanı için çizimler:

Servo motor devresi DS18B20 sıcaklık sensörü devresi

HC-05 bluetooth modülü devresi Su seviye ölçüm devresi

**Bluetooth Kontrollü Akvaryum Sistemi Arduino Kodu**

#include <OneWire.h>

#include <SoftwareSerial.h>

#include <Servo.h>

Servo servoNesnesi;//pin için ve servo işlemleri için nesne tanımlaması.

SoftwareSerial Genotronex(0, 1); // RX, TX

int data,deger=0;

int DS18S20\_Pin = 2;// Sıcaklık sensörü bağlanan dijital pin 2 olarak belirlendi.

// Sıcaklık Sensörü Giriş-Çıkışı

OneWire ds(DS18S20\_Pin); // 2. Dijital pinde.

int BluetoothData;

int seviye = 0;

float sicaklik = 0.0;

void setup()

{

OneWire ds(DS18S20\_Pin); // 2. Dijital pinde.

servoNesnesi.attach(12);

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

Serial.begin(9600); // Seri iletişimi başlatır.

}

void loop()

{

float temperature = getTemp();// temperature değişkeni sıcaklık değerini alma fonksiyonuna bağlandı.

sicaklik = temperature;

Serial.print("#"+(String)sicaklik+"#"+(String)seviye+"\r\n");//Telefonda yazdırma işlemi.

delay(1000);

deger=analogRead(A0); //Seviye için tanımlanan analog pin.

if(deger>=10)//analog pinden okunan değere göre seviye işlemleri.

{

seviye = 1;

digitalWrite(13,HIGH);

Serial.print("#"+(String)sicaklik+"#"+(String)seviye+"\r\n");

}

else

{

seviye = 0;

digitalWrite(13,LOW);

Serial.print("#"+(String)sicaklik+"#"+(String)seviye+"\r\n");

}

if (Genotronex.available())

{

BluetoothData=Genotronex.read();//bluetooth degerini okuma.

if (BluetoothData=='1') //Butona basıldıysa gelen değer 1 ise yemleme işlemi gerçekleştirme.

{

servoNesnesi.write(180);

delay(150);

servoNesnesi.write(90);

}

}

}

float getTemp()// Bu fonksiyon DS18B20 sıcaklık sensöründen gelen verileri Celcius cinsinden sıcaklık değerlerine çevirmek için kullanılıyor.

{

byte data[12];

byte addr[8];

if ( !ds.search(addr))

{

//Sensörü sıfırlama.

ds.reset\_search();

return -1000;

}

if ( OneWire::crc8( addr, 7) != addr[7])

{

Serial.println("Geçerli değil!");//Hata mesajı.

return -1000;

}

if ( addr[0] != 0x10 && addr[0] != 0x28)

{

Serial.print("Cihaz tanınmadı.");//Cihaz tanınmadı ise.

return -1000;

}

ds.reset();

ds.select(addr);

ds.write(0x44,1); // Dönüşüm başlaması.

byte present = ds.reset();

ds.select(addr);

ds.write(0xBE);

for (int i = 0; i < 9; i++)

{

data[i] = ds.read();

}

ds.reset\_search();

byte MSB = data[1];

byte LSB = data[0];

float tempRead = ((MSB << 8) | LSB);

float TemperatureSum = tempRead / 16;

return TemperatureSum;

}

**Android Kodu**

**Manifest :**

Manifest dosyasına bu kodları yazdığımız taktirde izin almış oluyoruz. Manifeste bu kodları eklememiz yeterlidir.

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

**Main Activity:**

package com.example.oem.akvaryumsistemi;  
import android.app.Activity;  
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.Build;  
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.ArrayAdapter;  
import android.widget.Button;  
import android.widget.ListView;  
import android.widget.TextView;  
import android.widget.Toast;  
import java.io.IOException;  
import java.io.InputStream;  
import java.io.OutputStream;  
import java.lang.reflect.Method;  
import java.util.ArrayList;  
import java.util.Set;  
import java.util.UUID;  
public class MainActivity extends Activity

{  
 private static final String *TAG* = "bluetooth2";  
 Button btnOn;  
 TextView txtSıcaklık,txtSuSeviyesi;  
 Handler h;  
 final int RECIEVE\_MESSAGE = 1; private BluetoothAdapter btAdapter = null;  
 private BluetoothSocket btSocket = null;  
 private StringBuilder sb = new StringBuilder();  
 private ConnectedThread mConnectedThread;  
   
 private static final UUID *MY\_UUID* = UUID.*fromString*("00001101-0000-1000-8000-00805F9B34FB");  
private static String *address* = "98:D3:31:40:9B:3E";@Override  
 public void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 btnOn = (Button) findViewById(R.id.*ac*);  
 txtSıcaklık = (TextView) findViewById(R.id.*suSicaklik*);  
 txtSuSeviyesi = (TextView) findViewById(R.id.*suSeviye*);  
  
 h = new Handler() {  
 public void handleMessage(android.os.Message msg) {  
 switch (msg.what) {  
 case RECIEVE\_MESSAGE:  
  
 byte[] readBuf = (byte[]) msg.obj;  
 String strIncom = new String(readBuf, 0, msg.arg1);sb.append(strIncom); int endOfLineIndex = sb.indexOf("\r\n"); if (endOfLineIndex > 0) { String dataInPrint = sb.substring(0, endOfLineIndex); if (dataInPrint.charAt(0) == '#'){  
 String[] dizi = dataInPrint.substring(1,endOfLineIndex).split("#");  
 txtSıcaklık.setText(dizi[0]);  
 if(dizi[1].equals("0"))  
 {  
 txtSuSeviyesi.setText("Yeterli");  
 }  
 else  
 {  
 txtSuSeviyesi.setText("Cok fazla");  
 }  
  
}  
sb.delete(0, sb.length());  
 dataInPrint = " ";  
 }  
 break;  
 }  
 };  
 };  
  
 btAdapter = BluetoothAdapter.*getDefaultAdapter*();   
 checkBTState();  
  
 btnOn.setOnClickListener(new View.OnClickListener() {  
 public void onClick(View v) {  
   
 mConnectedThread.write("1");   
   
 }  
 });  
  
  
 }  
 private BluetoothSocket createBluetoothSocket(BluetoothDevice device) throws IOException {  
 if(Build.VERSION.*SDK\_INT* >= 10){  
 try {  
 final Method m = device.getClass().getMethod("createInsecureRfcommSocketToServiceRecord", new Class[] { UUID.class });  
 return (BluetoothSocket) m.invoke(device, *MY\_UUID*);  
 } catch (Exception e) {  
 Log.*e*(*TAG*, "Could not create Insecure RFComm Connection",e);  
 }  
 }  
 return device.createRfcommSocketToServiceRecord(*MY\_UUID*);  
 }  
  
 @Override  
 public void onResume() {  
 super.onResume();  
  
 Log.*d*(*TAG*, "...onResume - try connect...");  
 BluetoothDevice device = btAdapter.getRemoteDevice(*address*);  
 try {  
 btSocket = createBluetoothSocket(device);  
 } catch (IOException e) {  
 errorExit("Fatal Error", "In onResume() and socket create failed: " + e.getMessage() + ".");  
 }  
 btAdapter.cancelDiscovery();  
 Log.*d*(*TAG*, "...Connecting...");  
 try {  
 btSocket.connect();  
 Log.*d*(*TAG*, "....Connection ok...");  
 } catch (IOException e) {  
 try {  
 btSocket.close();  
 } catch (IOException e2) {  
 errorExit("Fatal Error", "In onResume() and unable to close socket during connection failure" + e2.getMessage() + ".");  
 }  
 }  
 Log.*d*(*TAG*, "...Create Socket...");  
  
 mConnectedThread = new ConnectedThread(btSocket);  
 mConnectedThread.start();  
 }  
 @Override  
 public void onPause() {  
 super.onPause();  
 Log.*d*(*TAG*, "...In onPause()...");  
 try {  
 btSocket.close();  
 } catch (IOException e2) {  
 errorExit("Fatal Error", "In onPause() and failed to close socket." + e2.getMessage() + ".");  
 }  
 }  
 private void checkBTState() {  
 if(btAdapter==null) {  
 errorExit("Fatal Error", "Bluetooth not support");  
 } else {  
 if (btAdapter.isEnabled()) {  
 Log.*d*(*TAG*, "...Bluetooth ON...");  
 } else {  
 Intent enableBtIntent = new Intent(BluetoothAdapter.*ACTION\_REQUEST\_ENABLE*);  
 startActivityForResult(enableBtIntent, 1);  
 }  
 }  
 }  
 private void errorExit(String title, String message){  
 Toast.*makeText*(getBaseContext(), title + " - " + message, Toast.*LENGTH\_LONG*).show();  
 finish();  
 }  
 private class ConnectedThread extends Thread {  
 private final InputStream mmInStream;  
 private final OutputStream mmOutStream;  
 public ConnectedThread(BluetoothSocket socket) {  
 InputStream tmpIn = null;  
 OutputStream tmpOut = null;  
 try {  
 tmpIn = socket.getInputStream();  
 tmpOut = socket.getOutputStream();  
 } catch (IOException e) { }  
 mmInStream = tmpIn;  
 mmOutStream = tmpOut;  
 }  
  
 public void run() {  
 byte[] buffer = new byte[256];   
 int bytes;  
 while (true) {  
 try {  
 bytes = mmInStream.read(buffer);   
 h.obtainMessage(RECIEVE\_MESSAGE, bytes, -1, buffer).sendToTarget();   
 } catch (IOException e) {  
 break;  
 }  
 }  
 }  
 public void write(String message) {  
 Log.*d*(*TAG*, "...Data to send: " + message + "...");  
 byte[] msgBuffer = message.getBytes();  
 try {  
 mmOutStream.write(msgBuffer);  
 } catch (IOException e) {  
 Log.*d*(*TAG*, "...Error data send: " + e.getMessage() + "...");  
 }  
 }  
 }  
}

**Sonuç:**

Projede telefon kontrolü ile; balık yemleme, su seviyesini ölçüp telefona yansıtma ve su seviyesinin yeterli olup olmadığını belirlendi. Kullanıcıya gerekli uyarılar verildi.

Bu kontroller bluetooth modülü ile sağlayarak kullanıcının telefonunun bluetooth ile bağlantısı sonucu dile getirilen işlemlerin yapılması sağlandı.Devre ve kodlama işlemleri yapıldı. Proje kullanıcının kullanabileceği hale getirildi.

**Bluetooth Kontrollü Akvaryum Sisteminde Projesinde Kullanılan Kaynaklar**

\*http://arduinoturkiye.com/ds18b20-dijital-sicaklik-sensorunun-arduino-ile-kullanimi/

\*http://www.arduinobeyinleri.com/balik-yemleme-makinasi/

\*http://www.elektrikport.com/teknik-kutuphane/tatilde-baliklar-ac-kalmasin-arduino-bayram-projeleri/9020

\*https://gelecegiyazanlar.turkcell.com.tr/konu/arduino/egitim/arduino-301/servo-motor

\*http://arduinoturkiye.com/ds18b20-dijital-sicaklik-sensorunun-arduino-ile-kullanimi/

\*http://maker.robotistan.com/arduino-dersleri-17-hc-05-bluetooth-modulu-kullanimi/

\*http://maker.robotistan.com/arduino-dersleri-9-buzzer-ile-ses-cikisi-alma-2/