

Smart Water Tank IoT Project

Yazar: Berk Barış Kara

Üniversite: Pamukkale Üniversitesi

Şehir: Denizli - Türkiye

Mail: berkk.baris48@gmail.com

I. INTRODUCTION

Akıllı su tankı projesi için gerekli devre elemanları:

-Nodemcu esp8266

-HC-SR04 (Ultrasonik sensör)

-Breadboard

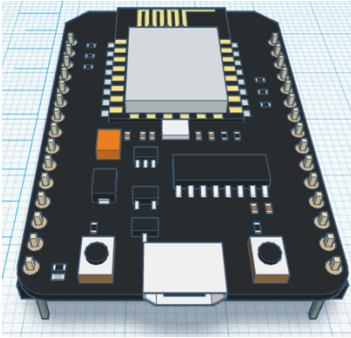
-Jumper wires

Projenin amacı Nodemcu esp8266'nın serverımıza anlık olarak tankta bulunan su seviyesini 7/24 her an 1 saniye aralıklarla aktarmasıdır. Bu su seviyesine göre de kişi Su tankındaki su miktarı azaldığı zaman bundan haberdar olabilecektir.

II. DEVRE ELEMANLARINI TANIYALIM

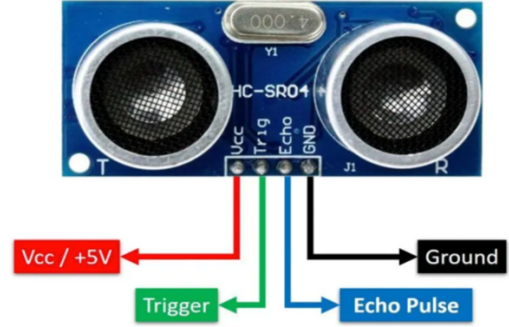
A. Nodemcu esp8266

Bu cihaz bir mikrokontrolcüdür. Üzerindeki WiFi modülü sayesinde herhangi bir ağı bağlanabilmektedir. Aynı zamanda devredeki diğer elemanlara da güç sağlar. Yeri geldiğinde devredeki diğer elemanları (sensörlerleri) uyarıcı bir güç 10 μ s uygulayabilmektedir. (trigger).



B. HC-SR04 (Ultrasonik sensör)

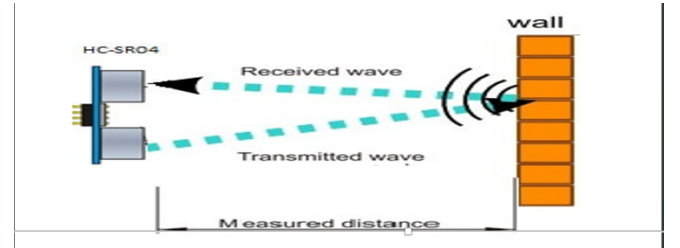
HC-SR04 bir ultrasonik ses sensörüdür. 40Khz frekasında dalgalar üretir ; bu dalgalar bir engelle karşılaşırca çarpar ve sensöre geri yansır. Böylelikle aradaki zamanı ve ses dalgasının hızını bildiğimiz için aradaki mesafeyi hesaplamamızı sağlar. (Distance = duration * 0.0034/2) HC-SR04 üzerinde 4 adet Pin'i bulunmaktadır. VCC-GROUND pinleri güç pinleridir ve 5 V güç uygulanmak zorundadır. Trigger pini nodemcu-esp8266 tarafından 10 μ s güç uygulanarak sensörün harekete geçmesini gerektiğini söyleyen pindir.Echo pini çıkış pinidir. Bu pin dalgaların çıkması sağlar



III. HC-SR04 NASIL ÇALIŞIR?

HC-SR04 ultrasonik sensör, 40KHz frekansında duyulmayan ultrasonik ses dalgalarını kullanarak mesafeyi ölçer. Ses dalgaları gibi ultrasonik dalgalar da havada ilerler ve eğer önlerinde herhangi bir engel varsa geliş açılırlarına göre yansır.

A. Nodemcu esp8266 trigger pinine verdiği güç ile 40Khz ses dalgaları yaymaya başlar etrafına daha sonrasında ise ultrasonik sensörü sayesinde objeden seken dalgaları yakalar ; sesin dalgasını bildiğimiz için basit bir matematiksel formülasyon ile arduino kodu mesafeyi hesaplayabiliyoruz. (distance = duration * 0.034 / 2;)



B. Nodemcu ile HC-SR04 arasındaki bağlantılar:

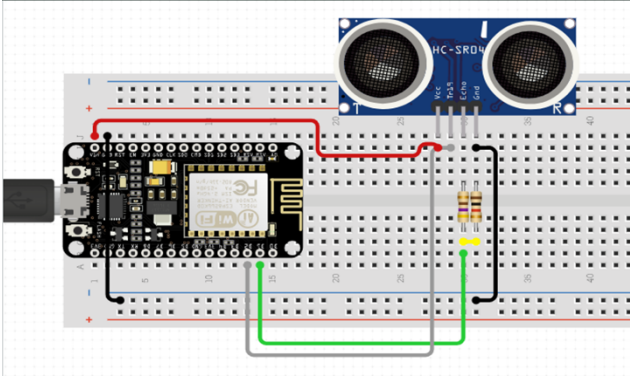
(HC-SR04)-VCC -> VU (Nodemcu)

GND -> GND

TRIG -> D2

ECHO -> D1

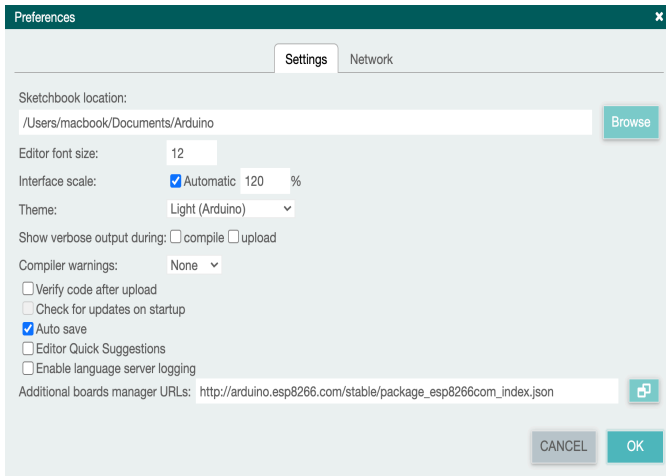
- Devrenin Bağlantılarının Gösterimi:



C. Nodemcu esp8266 konfigüre etmek

Nodemcu esp8266 açık kaynak kodlu bir modüldür. Bu sebepten dolayı dünyanın her tarafındaki geliştiriciler tarafından birçok özellik programlanmıştır bu özellikleri çok kolay cihazına entegre edip kullanılabilir. Bizim projemizde ise aşağıdaki linki Arduino IDE'ye eklememiz gerekmektedir.

https://arduino.esp8266.com/stable/package_esp8266com_index.json



Preferences kısmında add URL kısmına bu linki yapıştırıyoruz daha sonra ise bu kütüphaneyi indiriyoruz.

D. Nodemcu esp8266'yi ağa bağlamak

- Char formatında 2 değişken tanımlıyoruz isimleri ssid ve pass olmak üzere ; daha sonra ise bu değişkenlere ağıımızın ssid ve şifresini giriyoruz.
- Yukardaki link üzerinden import ettiğimiz kütüphanemiz aracılığı ile aşağıdaki kod satırı ile nodemcu'yu ağa bağlıyoruz.

```
WiFi.begin(ssid, pass);
while(Wifi.status() != WL_CONNECTED) {
  delay(1000);
  Serial.println('.');
}
Serial.println(Wifi.localIP());
```

IV. NODEMCU'NUN ALDIĞI VERİLERİ İLETME

A. Socket.io ile kendi server'ımıza bağlanma

Öncelikle Node-JS üzerinden kendi web serverımızı kodluyoruz:

```
const express = require('express');
const app = express();
const socket = require('socket.io');
const PORT = process.env.PORT || 3000;

const server = app.listen(PORT, () =>{
  console.log('server running..');
})

const io = socket(server);
```

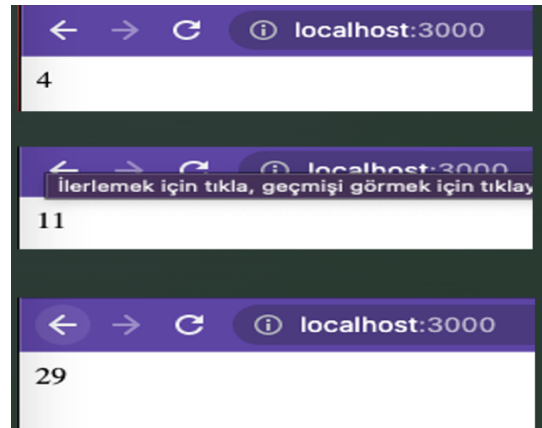
Socket değişkenimizi server'a bağlıyoruz.

Daha sonrasında ise socket.io üzerinden bir bağlantı kuruyoruz ve artık nodemcu serverımıza sensörden aldığı veriyi aktarıyor.

```
io.on('connection', (socket) => {
  console.log('Nodemcu Bağlandı' + socket.id);
  socket.on('message', function(message){
    app.get('/', (req,res,message) => {
      res.send(message);
    });
    console.log(message);
  })
})
```

B. Çıktılar

Node-JS üzerinden kurduğumuz server; socket.io ile nodemcu'ya bağlanıp bu veriyi localhost:3000 portuna göndermektedir.



C. Blynk ile nodemcu'dan gelen verileri alma

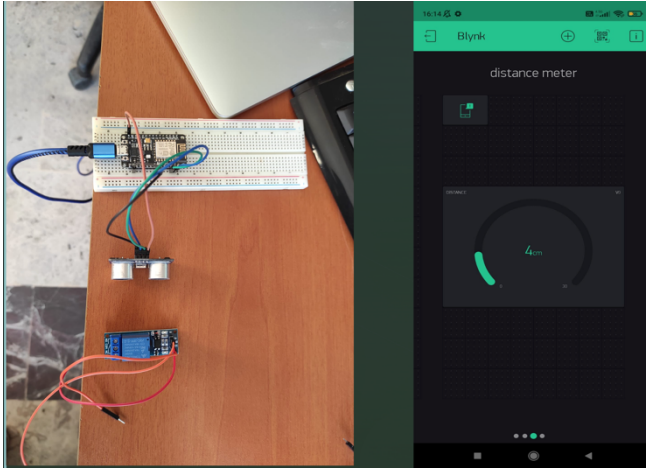
Blynk IoT uygulamaları mobil ve web arayüzü ile. Gelen verileri göstermemizi sağlayan özel bir yazılımdır. Size verilen özel bir key ile bağlantı kurulmaktadır.

```
char auth[] = "LX9WqbGF6BhVYo4AMMYD2ebiGeXquopA"; // Blynk Auth token
char ssid[] = "baris"; // WiFi veya Network adı
char pass[] = "12345678"; // WiFi password

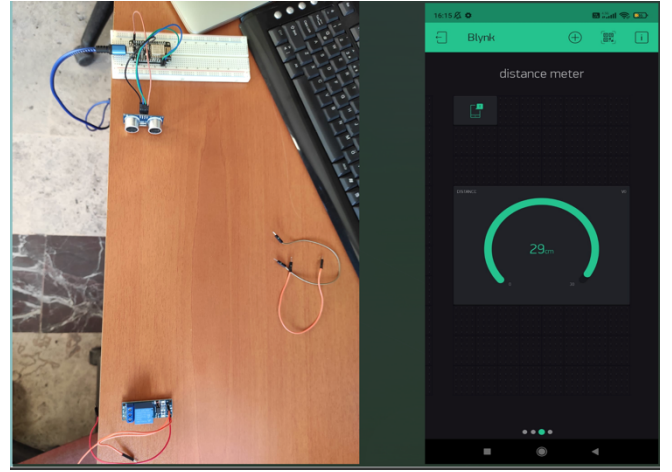
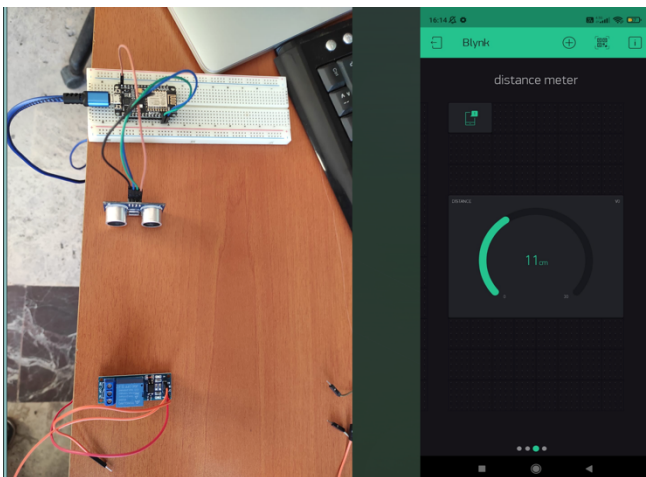
Blynk.begin(auth, ssid, pass); // blynk wifi bağlama
```

Blynk ile bağlantıyı kurduktan sonra mobil uygulamasını indiriyoruz;bu şekilde veriler anlık olarak telefonumuza geliyor.

D. Blynk üzerinden çıktıları görüntüleme



(4 cm üst görsel)
(11 cm alt görsel)



(29 cm)



Deneysel mesafe üzerinden yapılmıştır lakin su da bir engel olduğu için Arduino kodunu şu şekilde güncellediğimiz zaman bize su tankının içinde su yüzdesi gösterecek:

```
distance = duration * 0.034 / 2;
su_yuzdesi = length - distance;
```

Bu arada length su kabının uzunluğudur.

ACKNOWLEDGMENT

Projenin içinde tüm kodların bulunduğu ve sunumun da bulunduğu github linki:

- https://github.com/bariss48/Smart_Water_Tank

REFERENCES

- [1] <https://www.arduino.cc>
- [2] <https://github.com/esp8266/Arduino>
- [3] https://arduino.esp8266.com/stable/package_esp8266com_index.json
- [4] <https://blynk.io/>
- [5] <https://nodejs.org/en/>
- [6] <https://socket.io/>
- [7] <https://expressjs.com/>
- [8] <https://www.npmjs.com/>
- [9] <https://www.autodesk.com/solutions/circuit-design-software>
- [10] <https://www.tinkercad.com/>
- [11] <https://www.circuitlab.com/>
- [12] <https://www.circuito.io/>

Pamukale Üniversitesi IoT dersi Final sonu ödevi:

a) **Öğretim görevlisi: İbrahim KÖK**

b-) Öğrenci: Berk Barış Kara

