

**Bitirme Ödevi**

**Emre Kurnaz**

**İlker Furkan Güner**

**AKILLI SERA TASARIM RAPORU**

**GENEL BAKIŞ**

**TANIM**

Sera yetiştiriciliği tarım sektöründe ülkemizde çok önem teşkil etmektedir. Özellikle de yurdumuzun batı ve güney kesimlerinde ovada binlerce sera bulunmakta, burada üretilen gıda maddeleri hem yurtiçi besin ihtiyacının karşılanmasında hem de ihracatta çok önemli bir paya sahiptir. Ancak seraların çok olması önemli olsa da beraberinde bazı problemleri de meydana getirmektedir, örneğin ürünün yetişme aşamasında fazla sıcağa veya soğuğa maruz kalması, seranın içindeki nem oranı, aşırı veya yetersiz şekilde sulanması ürünün verimine ciddi şekilde zarar vermekte olup, hem üreticiyi hem de tüketiciyi mağdur etmektedir. Günümüzde artan maliyetleri de göz önünde bulundurduğumuzda çiftçinin emeğinin tam karşılığını alması önem arz etmektedir. Bu açıdan geliştireceğimiz AKILLI SERA ile bu soruna çözüm bulmayı hedefliyoruz.

**SORUN ÇÖZÜMÜ**

Sera içindeki mahsulün maksimuma yakın verim vermesi için kritik olan; sera içi nem, sera içi sıcaklık, sera içindeki toprağın nemi, sera dışındaki hava durumu gibi bazı durumların sistemimizde anlık kontrol edileceği, son kullanıcının -yani çiftçinin- serasındaki bu değişkenleri anlık olarak görüntüleyebileceği bir sistemle bu sorunu çözmeyi hedeflemekteyiz.

- Sera içindeki sıcaklığı ve nemi ölçmek için DHT11 sıcaklık ve nem sensörünü kullanacağız. Bu ölçülen değerleri ARDUINO ATMEGA2560 adlı akıllı karta göndereceğiz. Gönderdiğimiz bu değer veya değerleri akıllı kartımızda bulunan ESP8266 Wi-Fi Modülü ile uzak sunucuya göndereceğiz. Uzak sunucuya bağlanacak olan son kullanıcı ise bu değerleri akıllı telefonunda görüntüleyebilecektir.

Eğer sera içi sıcaklık belirlenen eşik değerini aşarsa, sera dışındaki hava koşulları sera üstünde bulunan hareketli çatıyı açmaya uygunsa çatı açılacak ve fan aynı anda çalışarak serayı soğutmaya başlayacaktır. Eğer hava koşulları uygun değil ise sadece fan çalışacaktır.

- Sera içindeki toprağın nemini ölçmek için toprak nem sensöründen gelen değerleri ARDUINO ATMEGA2560 adlı akıllı karta göndereceğiz. Gönderdiğimiz bu değer veya değerleri akıllı kartımızda bulunan ESP8266 Wi-Fi Modülü ile uzak sunucuya göndereceğiz. Uzak sunucuya bağlanacak olan son kullanıcı ise bu değerleri akıllı telefonunda görüntüleyebilecektir. Toprağın nemini ölçen sensörlerin sayısı seranın ebatlarına göre değişebilir. Eğer toprak gereğinden fazla susuz ise su pompası devreye girerek seranın içine su pompalayacaktır.

- Sera dışındaki hava durumunu ölçümlemek için ise ıslaklık sensörünü kullanacağız.

Bu sensörü kullanma amacımız ise olası bir sıcaklık artması ile yağmurun aynı anda yağması durumunda çatının açılıp yağan yağmurun sera içindeki mahsule zarar vermesini önlemek adına yapılacaktır.

**HEDEF KİTLESİ**

Genellikle çiftçiler baz alınarak yapmayı planladığımız seradan tarıma merakı olup, meyve sebze ekmeyi hobi edinen herkes faydalanabilir.

**Projede Kullanılacak Yazılımsal Araçlar**

Arduino IDE (Geliştirme Ortamı)

DHT11 Isı ve Sıcaklık Sensörü Kütüphaneleri,

Yağmur/Islaklık Sensörü Kütüphaneleri,

Toprak Nem Sensörü Kütüphaneleri,

TB6612 Dual Çift Motor Sürücü Kartı Kütüphaneleri,

NodeMCU ESP8266 ile ilgili kütüphaneler

**Arduino IDE**

Arduino Mega2560 ‘ı programlamak için bu geliştirme ortamı kullanılacaktır.

Arduino IDE ‘de C++ programlama dili ile kodlama yapılacaktır.

**Projede Kullanılacak Olan Donanımsal Araçlar**

-Sera Maketi

-ARDUINO ATMEGA 2560

-DHT11 Sıcaklık Nem Sensörü

-Yağmur/Islaklık Sensörü

-Toprak Nem Sensörü

-Sınır Switch’i

- TB6612 Dual Çift Motor Sürücü Kartı

-Su borusu

-Su pompası

-Su deposu

-Fan

-Çatı Motoru

**ARDUINO ATMEGA 2560**

Arduino Mega Atmega2560 tabanlı bir mikrodenetleyici kartıdır. Üzerinde 54 dijital giriş / çıkış pini bulunmaktadır ve bunlardan 15’i PWM çıkışı olarak kullanılanilir. 16 adalog giriş, bir 16 MHz kristal osilatör, bir USB bağlantısı, bir güç girişi, bir ICSP bağlantısı ve bir reset butonu bulunmaktadır.

**SENSÖRLER HAKKINDA BİLGİ VE SİSTEME ENTEGRE EDİLİŞİ**

**Toprak ve Nem Sensörü**

Toprak nem sensörü toprağın içindeki nem miktarını ölçmek için kullanabileceğimiz bir sensördür. Nem ölçer problar ölçüm yapılacak toprağa batırılarak ölçüm yapılır.

Toprak nem sensörü analog okuma ile 0 ile 1023 arasında değer okuyabilir. Bu değerler ne kadar fazlaysa topraktaki nem oranının o kadar fazla olduğunu göstermektedir.

Çalışma prensibi ise toprağın veya batırılan sıvının meydana getirdiği dirençten dolayı, prob uçları arasında bir gerilim farkı oluşur. Bu gerilim farkının büyüklüğüne göre de nem miktarı ölçülür. Topraktaki nem (ıslaklık) miktarı artıkça iletkenliği de artmaktadır. Kart üzerinde yer alan trimpot sayesinde hassasiyet ayarı yapılabilmektedir.



A0 pini sensöre gelen veriyi analog olarak okumak için, D0 pini ise potansiyometre ile ayarlananan değere göre digital çıkış almak için kullanılabilmektedir. 1.pini A0 pinine, 3.pini GND pinine ve VCC yi ise 5V pinine bağlamamız gerekiyor.

**DHT11 Sıcaklık Sensörü**

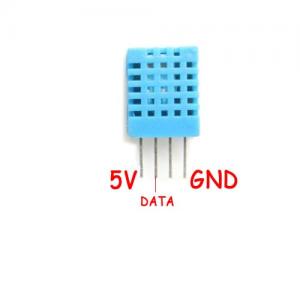
DHT11 sadece 0-50°C aralığındaki sıcaklığı ±2°C hassasiyetle ölçebilen sıcaklık sonucunu Celcius formatına veren hassas bir sıcaklık sensörüdür.Bu nedenle bu sıcaklığın altında ölçüm yapmanız gerekiyor.

DHT11, Arduino’ya veri iletmek için sadece bir sinyal kablosu kullanır.

Güç 5V ve ground kablolarından gelir. Sinyal seviyesinin varsayılan olarak yüksek kalmasını sağlamak için sinyal hattı ile 5V hattı arasında 10K Ohm çekme direnci gereklidir.

Bu sensör için DHT11 kütüphanesi kullanılacaktır.

DHT11 sıcaklık ve nem sensörünün arduino ile bağlantısı.



Sensörü görseldeki gibi tuttuğunuzda; en soldaki pin arduino 5V pinine, en sağdaki pin arduino GND pinine, data pini ise arduino dijital pinlerinden birine bağlanmalıdır.

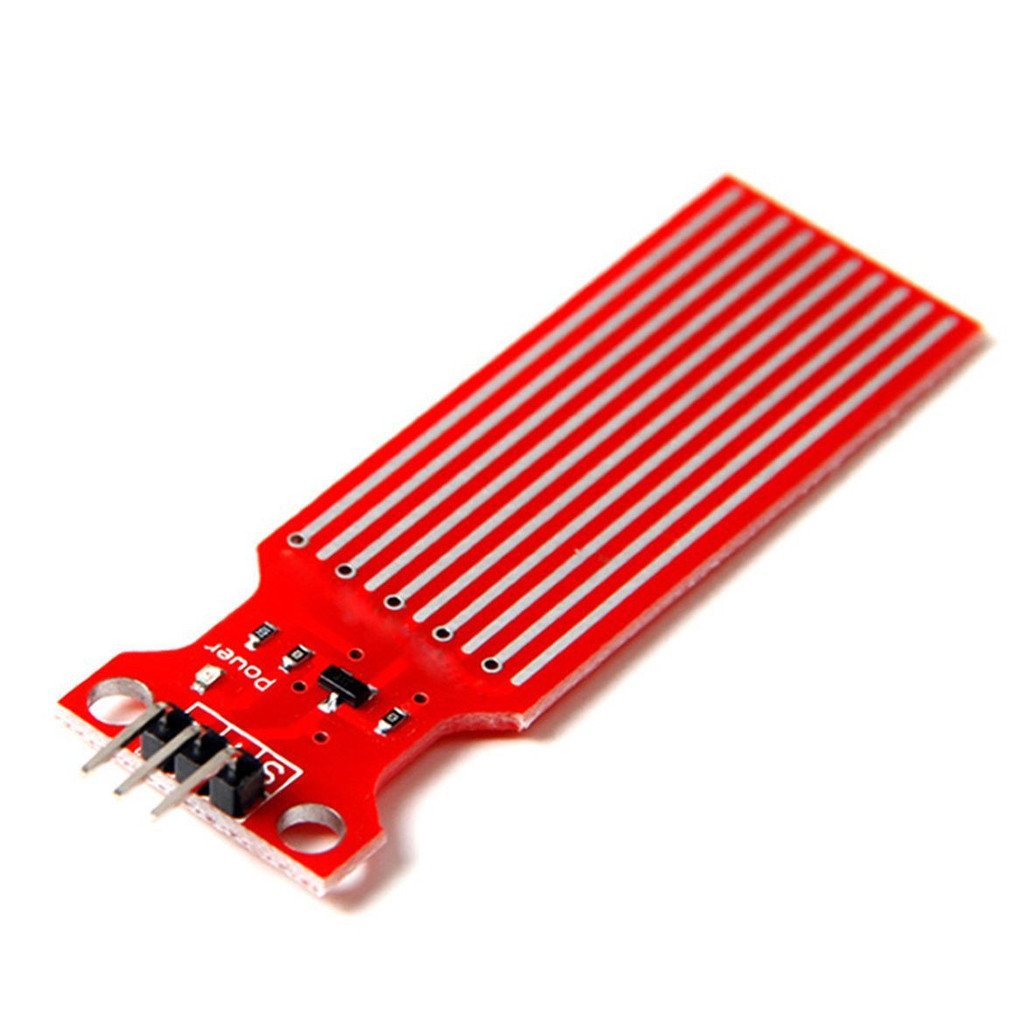
**Yağmur / Islaklık Sensörü**

Yağmur sensörünün çalışma mantığı, üzerinde bulunan iletken hatların su ile teması sonucu sensör çıkış değeri olarak dijital 1 veya 0 üretmesidir.

+ pini güç hattına,

- pini ise toprak hattına bağlanır.

S pini dijital çıkış pinidir.



**Sınır Sivici**

Daha çok vinçler ve halatlı kaldırma sistemlerinde halata gelen yükün limitlemesi için kullanılan sınır sivicini seramızda çatımızın açılıp kapanması sırasında eğer bir sıkıntı çıkarsa, çatı açılır veya kapanırken durmazsa sınır sivici ile bu sorunu çözmeyi amaçlamaktayız.

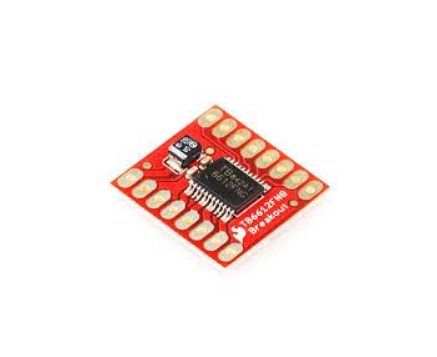


**TB6612 Dual Çift Motor Sürücü Kartı**

Mikrodenetleyicilerin çıkışları DC motorları veya step motorları direkt olarak kontrol etmek için yetersiz olduğundan motor sürücü devreler kullanılır. Motor sürücü devreler ile mikrodenetleyicilerin çıkışlarından alınan sinyaller yükseltilerek motorların kontrolü sağlanır.

Toshiba tarafından üretlien TB6612FNG bir entegre ile 13.5V ve her bir motoru 1.2A daimi akım ile iki motoru ayrı ayrı sürme imkanı verir. IN1 ve IN2 pinleri ile Sağa dönüş, Sola dönüş, Frenleme ve Boşta olamak üzere 4 modu kontrol eder. Out1A, Out1B ve Out2A, Out2B çıkışlarına bağlanan 2 motor hızlarını 100kHz'e kadar PWM sinyali ile hız kontrolü yapılabilir. Lojik beslemesi 2.7-5.5VDC arası olabildiğinden sürücü de bu seviyedeki sinyaller ile kontrol edilir.

Yapısal olarak modül, bir tarafta motor bağlantıları diğer tarafta 0.1" aralıklı kontrol pinleri ile donatılmıştır.



**Su Borusu**

Su deposunda bulunan suyu sera içerisindeki toprağa iletmek amacıyla –temsili olarak- serum lastiği kullanılacaktır.

**Su Pompası**

Su deposunda bulunan suyu seraya iletmek için kullanılacaktır.

**Su Deposu**

Suyun depolanması adına kullanılacaktır**.**

**Fan**

Seradaki optimum sıcaklığı sağlamak adına sistem tarafından gerekli görülürse veya kullanıcıdan direktif gelirse çalışmaya başlayacaktır.

**Çatı Motoru**

Seradaki optimum sıcaklığı sağlamak adına sistem tarafından gerekli görülürse veya kullanıcıdan direktif gelirse çalışmaya başlayacaktır.

**Yazılım Tasarımı Açıklamaları İçin Kavramsal Model**

Kullanacağımız kavramsal model IEEE 1016 standart modelidir.

Bu standart, yazılım tasarımlarını açıklar ve bir yazılım tasarım tanımlamasının (SDD) bilgi içeriğini ve organizasyonunu oluşturur.

SDD, tasarım bilgilerini kaydetmek ve bu tasarım bilgilerini ana tasarım paydaşlarına iletmek için kullanılacak bir yazılım tasarımının temsilidir.

Bu standart, açık bir yazılım tasarım tanımının hazırlanacağı tasarım koşullarında kullanılmak üzere tasarlanmıştır. Bu standart, dijital bilgisayarlarda çalışan ticari, bilimsel veya askeri yazılımlara üst düzey ve detaylı tasarımların tanımlarına uygulanabilir. Uygulanabilirlik, yazılımın boyutu, karmaşıklığı veya kritikliği ile sınırlı değildir.

Bu standart, herhangi bir tasarım dilinin kullanılmasını gerektirmez, ancak bir SDD'de kullanılacak tasarım dillerinin seçimine ilişkin gereksinimleri belirler. Bu standart, kâğıt dokümanlar, otomatik veritabanları, yazılım geliştirme araçları veya başka ortamlar olarak yakalanan SDD'lerin hazırlanmasında uygulanabilir.

**Sistemin Öğelerini Tanımlayan Yazılım/Donanım Tasarımları**

Kullanılan her tasarım görünümü, bir tasarım bakış açısına göre yönetilir ve her tasarım bakış açısı, bir tasarım bağlantısını belirtir ve tasarım görünümünü oluşturmak ve yorumlamak için kullanılan tasarım öğelerini sunar.

Kullanacağımız tasarım görünümleri sistemin tasarımını temsil eden uml diyagramları use case diyagramları vb. ve genel sistem mimarisinin anlaşılmasını sağlamak için tasarım görünümleriyle gösterilmiştir.

**Tasarım Açıklaması Bilgi İçeriği**

**SDD**

Bir yazılım tasarım belgesidir. SDD yazılım ürünün yazılı bir tanımıdır.

Yazılım tasarımcısı tarafından yazılım geliştirici ekibe yazılım projesinin mimarisi üzerinden bir rehberlik etmesi amacıyla yazılıp verilen bir belgedir.

SDD tasarımının ayrıntılı bir şekilde tüm bölümlerini ve nasıl çalışacaklarını ana hatlarıyla belirtir.

**Tasarım Görünümleri ve Bakış Açıları**

Sistemin tasarım mimarisini göstermek için tasarım görünümü ve ilgili bakış açıları kullanılmıştır. Kullanılan bakış açıları bağlam bakış açısı, kompozisyon bakış açısı, mantıksal bakış açısı, bilgi bakış açısı, etkileşim bakış açısı, durum bakış açısıdır.

**Tasarım Ögeleri**

Sistem ile ilişkili olan özellikler ve diğer bazı modüller arasındaki ilişkiler ana tasarım öğeleridir.

**Tasarım Bindirmeleri**

Tasarım bindirmeleri genellikle mevcut görünümlere bilgi eklemek için kullanılır.

**Tasarım Dili**

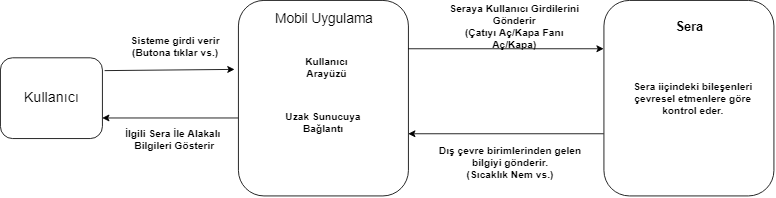
Tasarım dili olarak UML diyagramları kullanılacaktır, bu sayede sistemin statik yapısını ve dinamik davranışları öngörülmüş olur.

**TASARIM BAKIŞ AÇILARI**

**Yazılım ve Sistem Şemaları / Diyagramları**

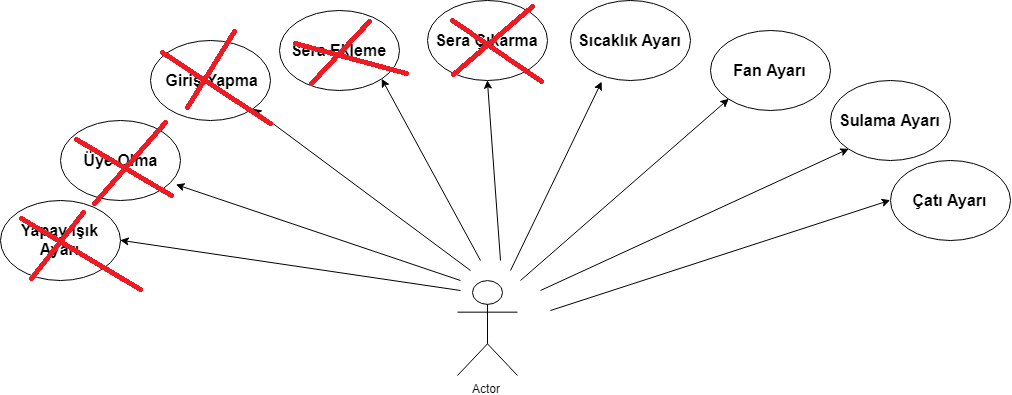
1. **Context (Durum) Diyagramı**

Context (Durum) diyagramları, kullanıcının sistemle kurduğu etkileşimi, sistemi oluşturan bileşenlerin birbirleriyle olan iletişimi gösteren diyagram çeşididir. Aşağıda Akıllı Sera sistemimizde bulunan bir şema gösterilmiştir.

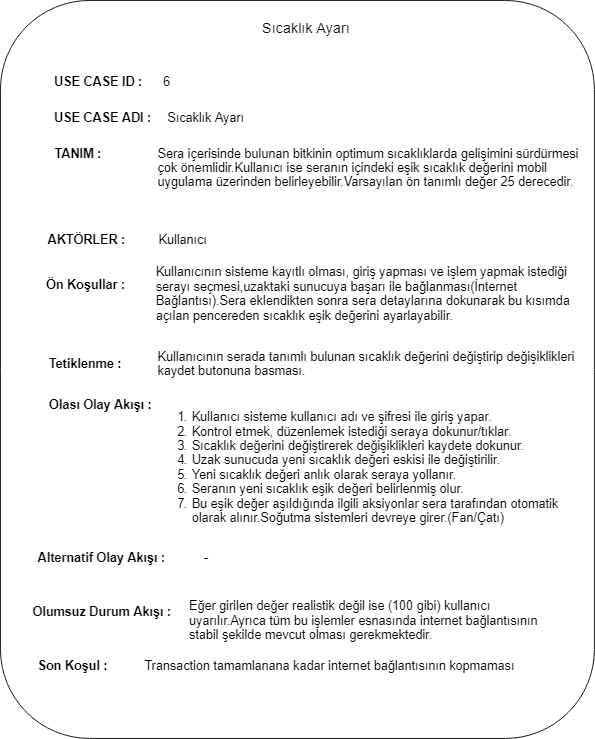


**2. Use Case Diyagramı**

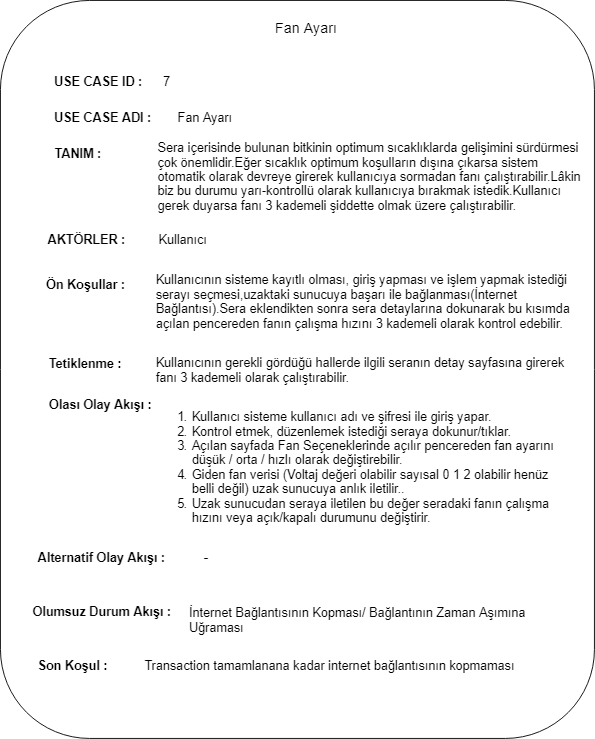
Use case diyagramları sistem içerisindeki durumların, sistemi kullanıcak olan son kullanıcı ile ilişkisini, sistem içindeki durumların birbirleri ile olan ilişkisini (Association), bu durumların ön koşullarını (pre-conditions) ve son koşullarını (post-conditions) , durumların işleyiş biçimlerini (Case Flow) açıklayan diyagram çeşididir. Sistemin kullanıcısının bakış açısıyla sistemin davranışlarının diyagramlarla modellenmesidir. Sistemin kabaca ne yaptığı ile ilgilenir. Use case modelini oluşturan diğer önemli bir yapı ise senaryolardır. Senaryolar kullanıcı tarafından başlatılan çeşitli olaylar dizisidir. Sistemi tasarlamamız için use case’lerden faydalanacağız.



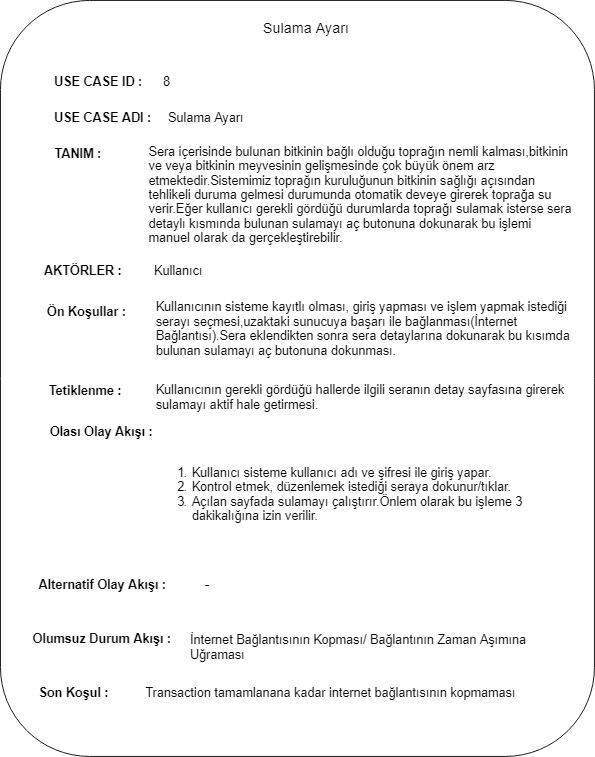
**Use Case’ lerin Detaylı Açıklanması**

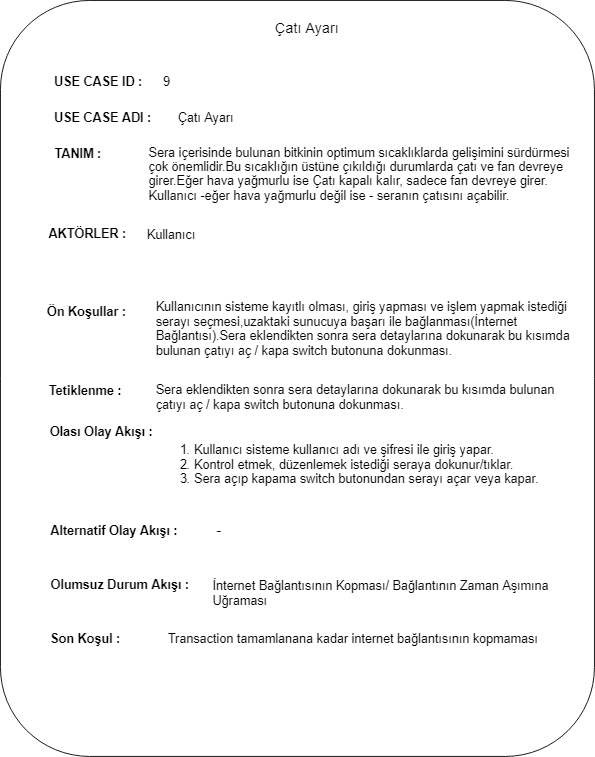
**1-Sıcaklık ve Nem Ayarı**

**2-Fan Ayarı**

****

**3-Sulama Ayarı**

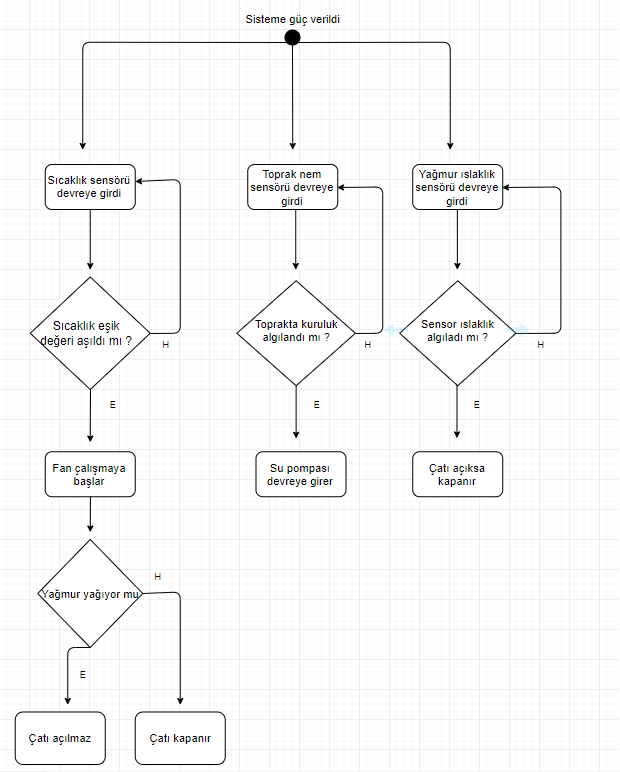
****

**4-Çatı Ayarı**

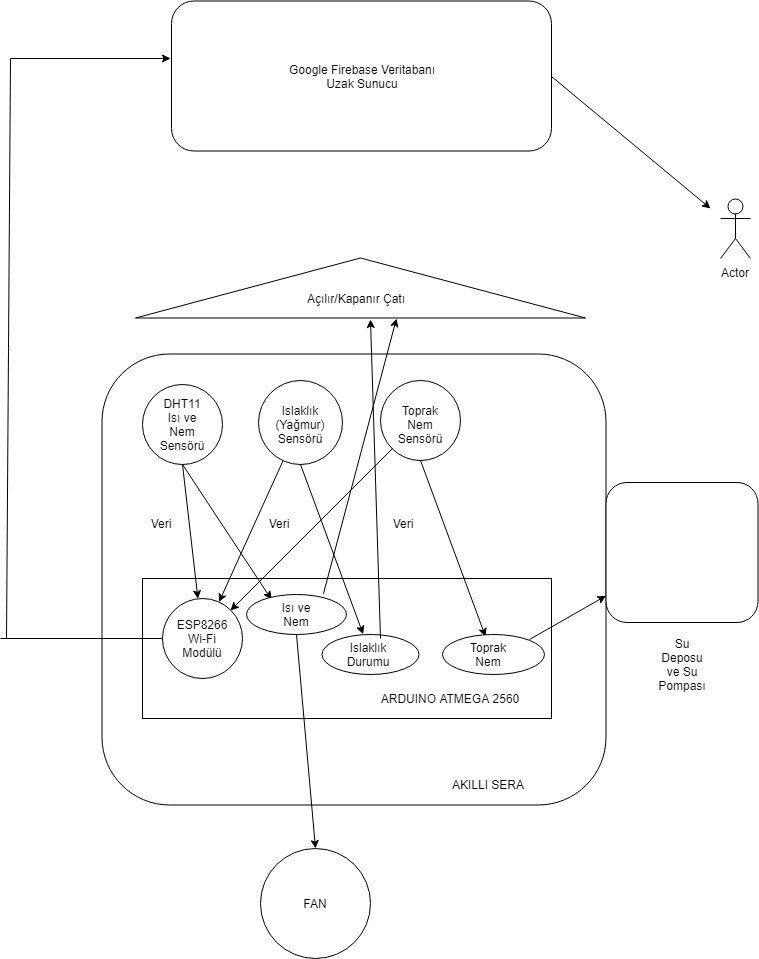
**BLOK DİYAGRAM**

Bir sistemdeki temel bileşenlerin bloklar halinde gösterildiği şema türüdür.

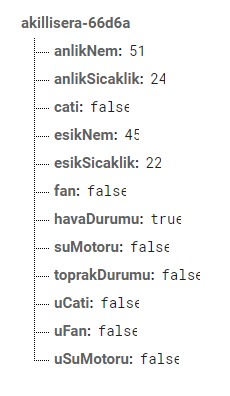
Kutular veya bloklar bileşenleri temsil eder, bloklar arasında bağlantı hatları arayüzleri temsil eder. Sistemimizi tasarlarken blok diyagramından faydalanacağız.



**KOMPONENT (BİLEŞEN) Diyagramı**



**VERİTABANI DİYAGRAMI**

****

-Veritabanında seralar JSON veri formatına uygun şekilde yukarıda örneklendiği gibi tutulacaktır. Veriler Google Firebase’te bulunan realtime database içerisinde saklanacak, sensörlerden gelen veriler anlık olarak işlenecektir.

**Class Diyagramları**

-Projemizde NodeMCU için ayrı, Arduino MEGA 2560 için ayrı ve “Akıllı Sera” mobil uygulaması için ayrı ayrı kodlar bulunmaktadır. NodeMCU ve Arduino MEGA 2560 için olan kodlarda class yapısı yoktur yani nesnel yapı yoktur. Bunun için fonksiyon fonksiyon hangi fonksiyonunun ne iş yaptığını kod parçasının ekran görüntüsünün altına kısaca açıklayarak yaptık.

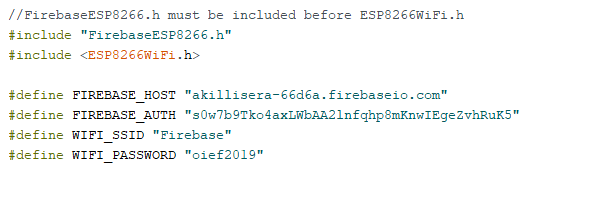
NodeMCU Kaynak Kodları(softwareserial.ino)

\*\*ÖNEMLİ:

-NodeMCU ESP8266 Wi-Fi modülünü Google Firebase’te oluşturduğumuz projeye bağlamak, veri almak, veri göndermek amacıyla hazır kütüphane kullandık. Kütüphanenin linki : <https://github.com/mobizt/Firebase-ESP8266>

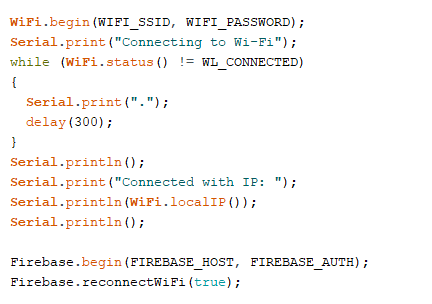
-Kütüphanede sadece veri getir, veri götür, veri tabanına bağlan fonksiyonları hazır fonksiyonlardır. Programın çalışmasındaki algoritma mantığı tamamen tarafımızdan yapılmıştır.

1-



Firebase veritabanına bağlanmada kullanacağımız kütüphaneler ve gerekli değişkenler burada tanımlanmıştır.

2-

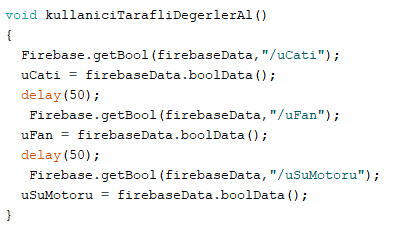


-Wi-Fi ile ilgili cihaza bağlanma fonksiyonu WiFi.begin(…)

-Wi-Fi cihaza bağlanana kadar while döngüsü çalışır.

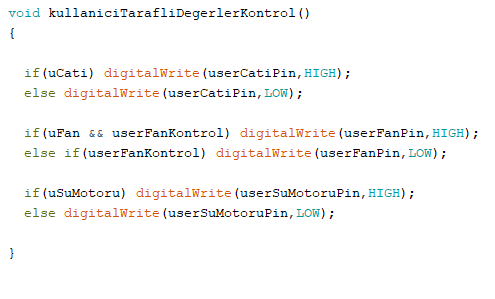
-Firebase.begin(…) fonksiyonu ile Firebase veri tabanına bağlantı gerçekleştirilir.

3-



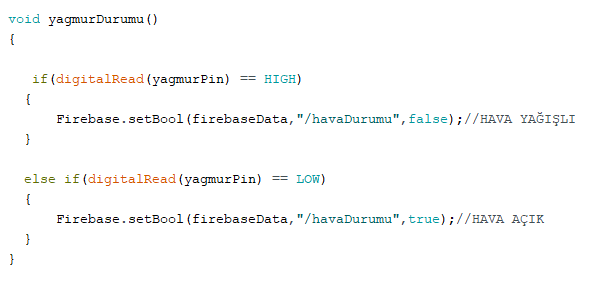
-Bu fonksiyon mobil uygulamadan gelecek olan kullanıcı taraflı; fan açma-kapama, çatı açma-kapama, su motoru açma-kapama değerlerini okur, u ile başlayan değişkenlere atar. Bu ‘u’ ile başlayan –boolean- değişkenleri bazı fonksiyonlarda kontrol amaçlı kullanacağız.

4-



-Yukarda belirlediğimiz değişkenleri burada userCatiPin, userFanPin, userSuMotoruPin gibi pin çıkışlarına HIGH veya LOW vermeye karar verirken kullanıyoruz.

5-



-Burada ise Arduino’dan gelen yağmur verisini okuyup, Firebase’teki ilgili yere yolluyoruz.

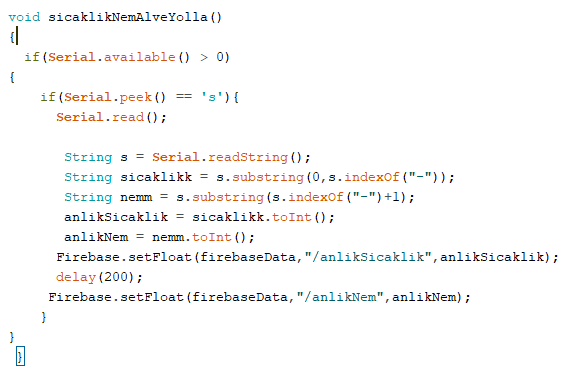
6-



-Arduino’dan gelen su motorunun çalışıp çalışmadığını öğrenip.Firebase’teki suMotoru ve toprakDurumu değişkenlerini true veya false yapıyoruz.

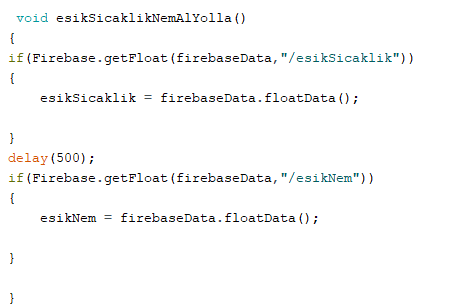
-if(uSuMotoru) kodundaki mantık, toprak kuru olduğunda ve kullanıcı sulamaya izin vermişse toprağı sula, yoksa sulama şeklindedir.

7-



-Sıcaklık ve nem sensöründen ölçülen değerler Arduino’ya gider. Arduino üzerindeki TX portu NodeMCU’nun RX portuna bağlanır. Bu port üzerinden Serial ile iletişim sağlanır. Burada Serial portundan gelen veriyi kontorl ediyoruz.Bu kontrol şu şekilde: Eğer gelen veri ‘s’ karakteri ile başlıyorsa önce Serial.read() ile bir karakter çeker.Yani veri “s25-50” ise geriye “25-50” kalır.25 sıcaklık, 50 nem olur. Sonra bu veri Firebase’e yollanır.

8-



-Eşik sıcaklık ve nem değerleri Firebase’ten alınır. Sonra bu değerler anlık sıcaklık ve anlık nem değerleri ile karşılaştırılacaktır.

9-



-Bu kısım eşik sıcaklık değerinin anlık sıcaklık değerinden düşük olduğu veya eşik nem değerinin anlık nem değerinden düşük olduğu durumlarda veya bu her iki durumun sağlandığı koşullarda yapılacak işlemlerin gösterimidir.

-userFanKontrol false yapılır, çünkü bu durumda fan koşulsuz şartsız çalışmalı, çatı ise kullanıcının isteğine bağlı çalışır.

-D2 pininden gelen yağmur bilgisi kontrol edilir.Yağmur yağıyorsa sadece FAN açılır.

-Diğer durumda ise fan çalışır, çatı ise kullanıcı açılmasına izin vermişse açılır.

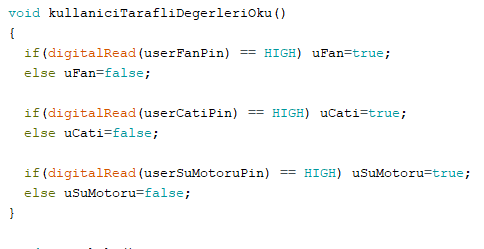
10-



-Üstteki koşul sağlanmadığı takdirde fan kullanıcı taraflı açık ise çalışmaya devam eder, değilse durur. Aynı kural çatı için de geçerlidir. Fakat yağmur yağıyor ise çatı kullanıcı tarafla izne bakmaksızın kapanır.

**Arduino MEGA 2560 Kodları(arduinomega.ino)**

1-



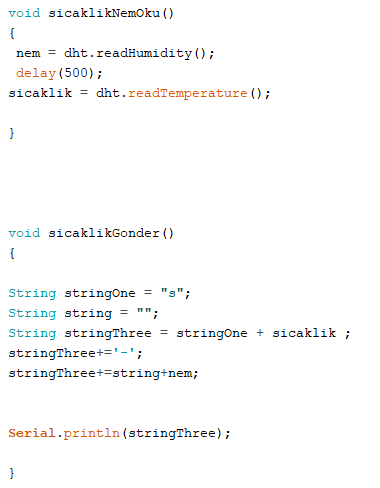
-Firebase’ten NodeMCU’ya gelen kullanıcı taraflı değerler Arduino ile NodeMCU arasında kurulan pin bağlantılarından bu değerler okunur, ilgili değişkenlere true ya da false atanır, sonra bu değişkenler ilgili yerlerde kullanılır.

2-



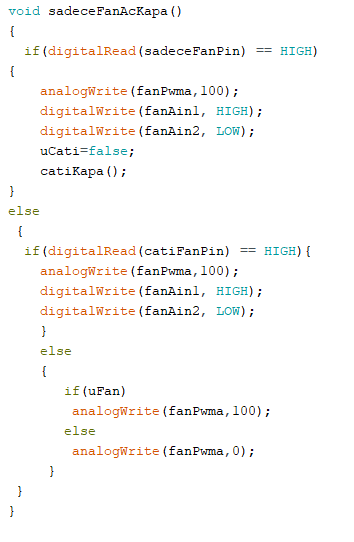
-Toprak sensöründen gelen dijital veriye göre su motoru –eğer kullanıcı taraflı açılmasına izin verilmiş ise- açılır ( digitalWrite(suMotoruPwma,40) )

3-



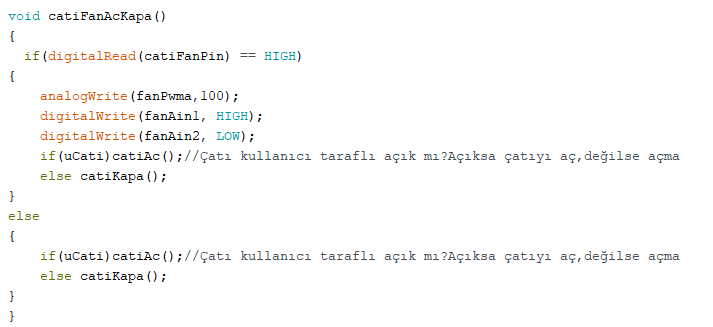
-Sıcaklık ve nem DHT11 sensöründen okunur,sicaklikGonder() metodu ile de serial port üzerinden NodeMCU’ya gönderilir.

4-



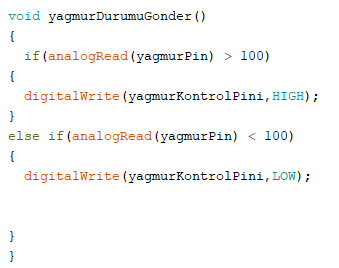
-Bu fonksiyon yağmur yağdığı ama seranın çok sıcak veya çok nemli olduğu durumlarda çalışan kod parçasıdır.Sadece fan çalışır,çatı açıksa kapanır.

5-



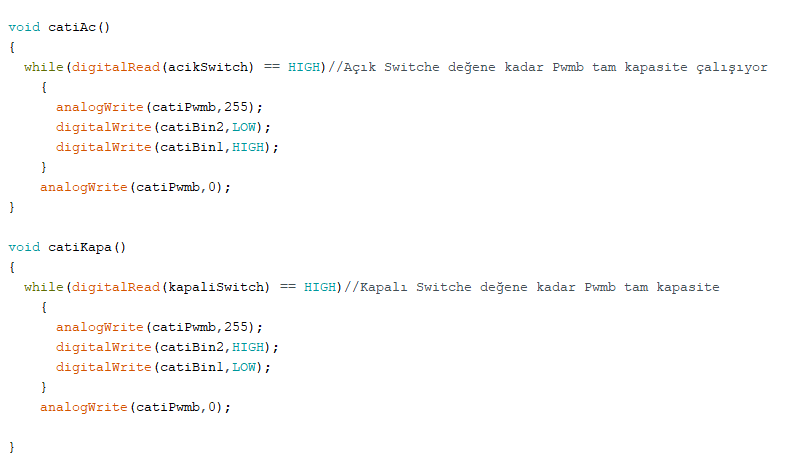
- Bu fonksiyon havanın açık olduğu, seranın çok sıcak veya çok nemli olduğu durumlarda çalışan kod parçasıdır.Fan kesinlikle çalışır,çatı kullanıcının tercihine bağlı açılır.

6-



-Yağmur sensöründen gelen veriye göre eğer yağmur yağıyorsa Arduino’dan NodeMCU’ya giden pinlerden yagmurKontrol pinini HIGH yaparak havanın yağmurlu olduğu bilgisine NodeMCU’ya ulaştırıyoruz.

7-



-Switchlerden gelen veri kontrol edilir while döngüsü koşul sağlanmayana kadar döner döngüden çıktıktan sonra güç kesilir.

**MOBİL ARAYÜZ**



**Mobil Arayüz Hakkında**

-Seradaki anlık sıcaklık değeri, “Anlık Sıcaklık” etiket altında gösterilmiştir. Bu veriler Google Firebase anlık değişken verili veritabanından gelen verilerdir. (Üstteki Görselde 24)

-Seradaki anlık nem değeri “Anlık Nem” etiketi altında gösterilmiştir. Bu veriler Google Firebase anlık değişken verili veritabanından gelen verilerdir.(Üstteki Görselde 51)

-Toprak Durumu etiketinin altında ise toprağın o anki durumu ile alakalı bilgiyi içerir.(Kuru veya Nemli olabilir)

-Hava Durumu etiketinin altında ise sera dışındaki hava durumunun o anki anlık verisi vardır.(Açık-Yağışlı)

-Kullanıcının belirlediği, sıcaklık eşik değeri sol taraftaki “+”,”-“ etiketli butonların arasında gösterilmiştir. Kullanıcı bu eşik değeri arttırıp azaltabilir. Bu “sıcaklık eşik değeri”, “anlık sıcaklıktan“ düşük olursa kullanıcı talimatına bakılmaksızın sera fanı açar, çatının açılması içinse kullanıcı taraflı izin verildikten sonra açılır.

-Kullanıcının belirlediği, nem eşik değeri sol taraftaki “+”,”-“ etiketli butonların arasında gösterilmiştir. Kullanıcı bu eşik değeri arttırıp azaltabilir. Bu “nem eşik değeri”, “anlık nemden“ düşük olursa kullanıcı talimatına bakılmaksızın sera fanı açar, çatının açılması içinse kullanıcı taraflı izin verildikten sonra açılır.

-Su Motoru etiketi altında su motorunun açık olup olmadığı ile alakalı bir bilgi vardır.

-Soğutma(FAN) etiketi altında fanın açık olup olmadığı ile alakalı bir bilgi vardır.

-Çatı etiketi altında çatının açık olup olmadığı ile alakalı bir bilgi vardır.

-Su Motoru “açık-kapalı” butonları su motorunu kullanıcı taraflı açıp kapamaya yarar.

-Soğutma(FAN) “açık-kapalı” butonları fanı kullanıcı taraflı açıp kapamaya yarar.

-Çatı “açık-kapalı” butonları çatıyı kullanıcı taraflı açıp kapamaya yarar.

**Arayüzdeki İstisnai Durumlar**

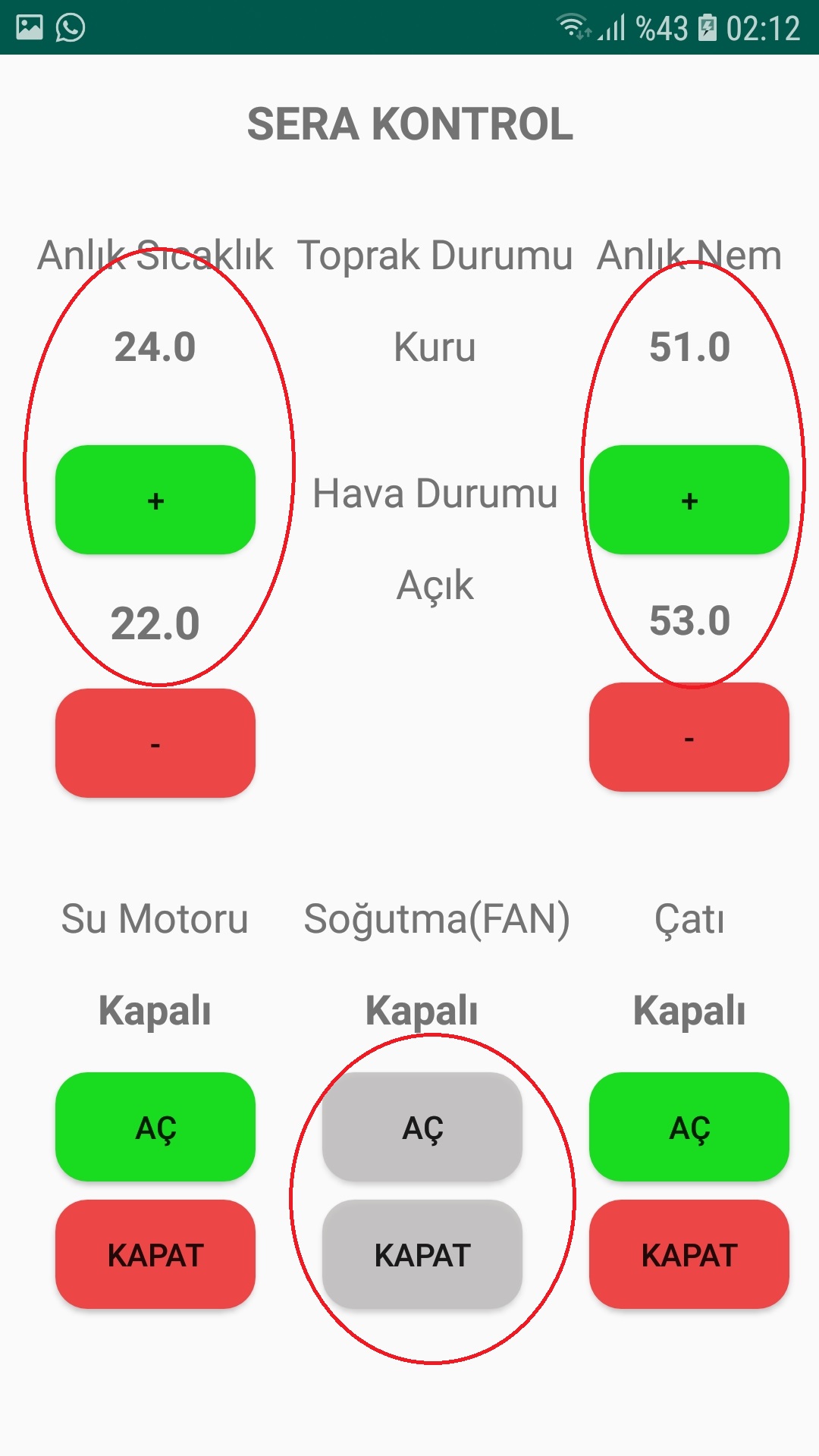
1-Eğer yağmur yağıyor ise çatı kapatılır ve kullanıcı taraflı çatıyı açma kapama butonları pasifleştirilir.



2-Eğer toprak durumu nemli ise su motoru açma kapama butonları pasifleştirilir.



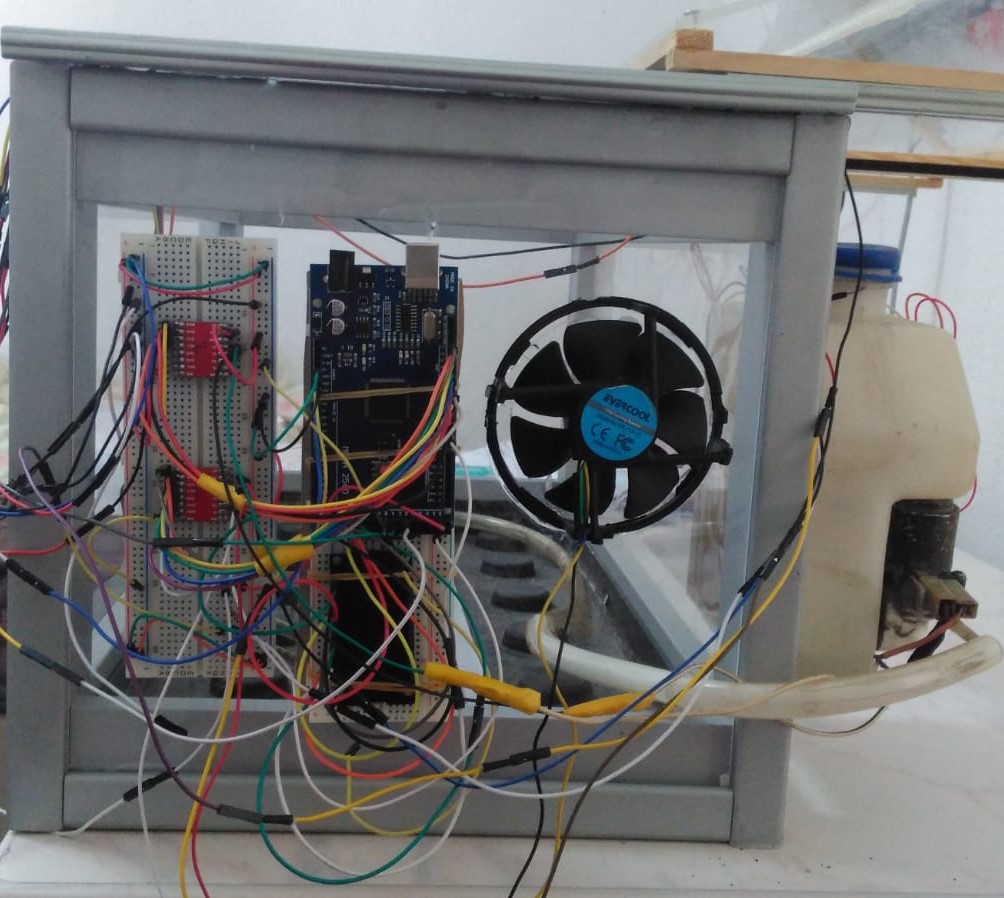
3-Eğer eşik sıcaklık değeri anlık sıcaklık değerinden düşükse veya eşik nem değeri anlık nem değerinden düşükse fan açılır, kullanıcı taraflı fana müdahale engellenir. Fakat bu iki durum söz konusu değilse kullanıcı fanı dilediği gibi açıp kapatabilir.



**Mobil Uygulamanın Kaynak Kodları**

<https://github.com/gunerfurkanilker/AkilliSeraa>

**Seranın Dış Görünümleri**

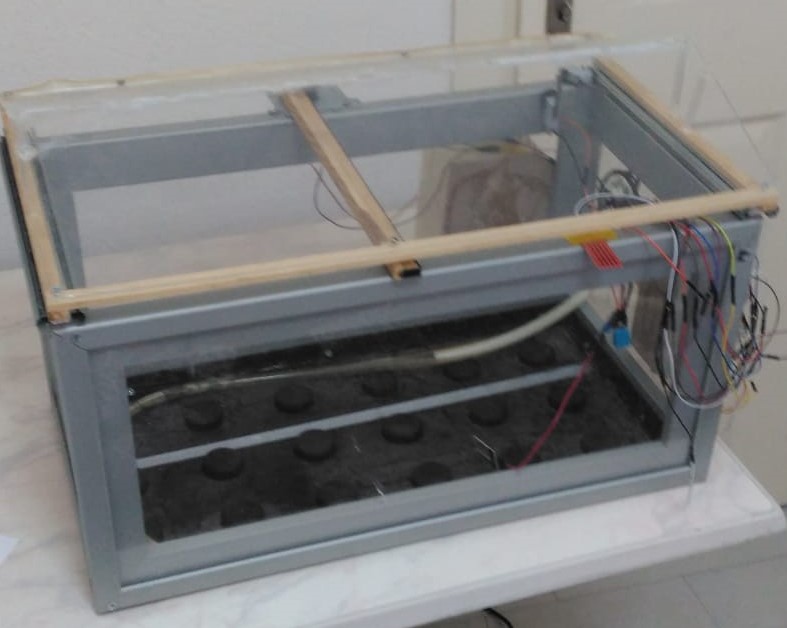


**Yan kısımdan genel görünüm: Soldaki board’un üzerinde 2 adet motor sürücü var Sağdaki board da ise üstte Arduino MEGA 2560 altta NodeMCU ESP8266 bulunuyor.Hemen onun sağında fan,onun da sağında su pompalama amaçlı su motoru bulunuyor.**



**Sınır switchleri : Dış yüzeydeki switch’e çatının tahta kısmı değince-basılınca çatının kapalı olduğunu ,iç yüzeydeki switch e basıldığında ise açıldığını anlıyoruz.**

**Genel Görünüm**



**Üstten görünüm: Yağmur sensörü(Kırmızı olan) ve üstte görsellerde de gösterilen sıcaklık ve nem ve toprak nemi sensörleri.**

**GANNT CHART**

Aktivitelerin planlanması ve sıralamasında kullanılan grafiksel bir araçtır. Bir zaman ölçeğine göre işin planlanmasında ve yürütülmesinde kullanılabilir. Projedeki işlemlerin başlangıç ve bitiş işlemlerini alt alta zaman skalasına yerleştirerek tabloya aktaran diyagram, bize işin nerede başlayıp nerede bittiğini anlamamız konusunda yardımcı olacaktır. Kullanıcı bu çizelge ile projedeki işlerin takibini be kritik yolun değerlendirmesini yapar.

