**2022-2023**

**GÖMÜLÜ SİSTEM**

**DERSİ LABORATUVAR PROJESİ**

**Grup No:11**

**200601038**

**Onur KUDRET**

**200601039**

**Ümit YILMAZ**

**200601019**

**Utku AYDIN**

**180601041**

**Zehra UZUN**

**13.06.2023**

1. İhtiyaç ve Proje Analizi
   1. Projenin amacı ve hedefi
   2. Kullanıcıya yönelik faydaları
   3. Kullanılan yazılım dili, derleyici veya diğer araç gereksinimleri
2. Tasarım
   1. Proje tasarımında kullandığınız platformlar (web tabanlı, mobil veya masaüstü uygulaması, vs.), cihazlar veya kartlar
      1. Tasarımınızda kullandığınız malzemeler (devre elemanları, lojik kapılar, denetleyici, led, göstergeler, vs.) ve ne amaçla kullanıldığı ile ilgili kısa bir açıklama
   2. Projenin devre şeması
   3. Projenin akış diyagramı
   4. Tasarım ekran görüntüleri
3. Kodlama
   1. Kullandığınız giriş ve çıkış değişkenleri (portlar, vs.)
   2. Kullandığınız Timer ve Interrupt verileri, ne amaçla kullanıldığı
   3. Kullandığınız fonksiyonlar veya alt program çağrıları, ne amaçla yazıldığı
4. Doğrulama ve Test
   1. Projenizin kısıtları
      1. Projede neler yapıldı?
      2. Projenizin özgünlüğü ve pratik yaşama katkısı nedir?
      3. Projenin eksiklikleri neler?
      4. Projede olması gereken diğer durumlar
   2. Projeden beklenen sonuçlar
   3. Ortak çalışma adına gerçekleştirilen görev paylaşımları ve toplantılardan görüntüler

1. İhtiyaç ve Proje Analizi

**a.** Projenin Amacı ve Hedefi:

Yapmış olduğumuz “Ayarlanabilir Saat ve Takvim” projemizdeki asıl amaç bu teknoloji ile entegre edilmiş ürünleri kullanırken günlük hayatta aktif olarak saat ve takvim bilgilerine erişmek isteyen kullanıcıların bu ihtiyaçlarını karşılamaktır. Ayrıca sahip olduğu alarm kurma özelliği de istenen herhangi bir zaman için hatırlatıcı olmasını sağlar. Kısacası bu projenin asıl yapılma amacı kullanıcıların günlük hayattaki en temel ve basit ihtiyaçlarına birleşik bir çözüm sunmaktır.

**b.** Kullanıcıya Yönelik Faydaları:

Bu ürün kullanıcılara zamanlarını yönetme ve önemli anlarında birer hatırlatıcı olma özelliği taşımaktadır. Bu basit ama vazgeçilemeyen ihtiyaç, ürünü kullanacak olan kişilere sahip olduğu özellikleriyle büyük ölçüde konfor ve kolaylık sağlamaktadır. Günlük hayatta karşılaşılan durumlarda kullanıcı dostu, kolay anlaşılır ara yüzü ve kullanım şekliyle zahmetsiz bir şekilde zaman planlaması sağlar.

**c.**

Kullanılan Yazılım Dili: “C”

Kullanılan IDE: “Arduino IDE”

Kullanılan Derleyici: “GCC (GNU Compiler)”

2. Tasarım

**a.** Kullanılan Tasarım Aracı: “Proteus Professional”

**a. i.** Tasarımda Kullanılan Devre Elemanları:

RTC(DS1307) Entegre Devre: DS1037 Real Time Clock sahip olduğu gerçek zamanlı saati gösterme özelliğiyle proje içerisinde ARDUINO UNO ile veri senkronizasyonu sağlayarak devre kartının RTC’nin gönderdiği sinyaller ile eş zamanlı olarak zaman verisini almasını sağlar.

ARDUINO UNO Devre Kartı: Proje içerisinde kullanılan tüm devre elemanlarının senkronizasyonunu, veri paylaşımını, I/O değerleri ve bu veriler sonucunda yapılması gereken işlemleri tanımlayan ve düzenleyen devre elemanıdır. Kısacası devrenin beynidir.

LED Lamba: Kurulan alarm sonucunda gerçek zamanlı saat ile kurulan alarm eşleşince durmaksızın yanacak olan led lamba uyarıcı görevi görür.

Buzzer: LED Lamba ile entegre bir şekilde eş zamanlı olarak ses sinyali verir. Bir uyarıcı görevi görür.

Buton: Projede 4 adet buton bulunmaktadır. Bu butonlar “ayarla, arttır, sonraki, azalt” şeklinde belirlenmiştir. Ayarla butonu alarm ayarlama menüsünü açar. Arttır butonu saati ve dakikayı arttırır. Sonraki butonu saatten dakikaya, dakikadan da alarmı onaylama işlevleri arasında geçişi sağlar. Azalt butonu saat ve dakikayı azaltma işlevini yerine getirir.

LCD Panel(LM016L): LCD panel saat, takvim, alarm ayarlama menüsü ve geri bildirim mesajlarını gösterir. Kısacası tüm çıktıları analog sinyallerden dijital sinyale dönüştürerek ekranda görüntü oluşturur.

diyagram, metin, teknik çizim, plan içeren bir resim

Açıklama otomatik olarak oluşturuldu**b.** Devre Şeması:

**c.** Akış Diyagramı

metin, diyagram, plan, teknik çizim içeren bir resim

Açıklama otomatik olarak oluşturuldu

**d.** Tasarıma Ait Ekran Görüntüleri:

LED:

diyagram, ekran görüntüsü, yazı tipi, çizgi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Butonlar:

diyagram, metin, yazı tipi, çizgi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Buzzer:

metin, diyagram, ekran görüntüsü, çizgi içeren bir resim

Açıklama otomatik olarak oluşturuldu

LCD Panel:

metin, yazı tipi, sayı, numara, diyagram içeren bir resim

Açıklama otomatik olarak oluşturuldu

RTC:

metin, ekran görüntüsü, diyagram, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Arduino UNO:

metin, ekran görüntüsü, diyagram, baskı, basma işlemi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Devre:

metin, ekran görüntüsü, ekran, görüntüleme, yazılım içeren bir resim

Açıklama otomatik olarak oluşturuldu

3.Kodlama

**a.** Kullanılan Giriş ve Çıkış Değişkenleri:

lcd: 3,2,4,5,6,7 numaralı portlar bu değişken içerisinde veri aktarımını sağlamak ve LCD panel tarafından dijital bir görüntüye dönüştürülmek üzere tanımlanmıştır.”4,5,6,7”numaralı portlar veri aktarımı için, “3” numaralı port LCD panelin komut ve veri yazma işlemlerini ayırmak için, “2” numaralı port ise LCD panel üzerindeki bir işlemi başlatmak veya sonlandırmak için kullanılır.

RTC: Bu değişken RTClib kütüphanesinde yer alan bir tipte oluşturulmuştur veri aktarımını sağlayan SDA ve saat hızıyla senkronizasyonu sağlayan SCL portları Arduino üzerindeki gerekli “A4,A5” portlarına bağlanmıştır.

buzzer: Bu sembolik sabit, 13 numaralı pin ile ilişkilendirilir. Yani, bu pin, Arduino'da bir hoparlör veya buzzer gibi bir ses çıkış cihazını kontrol etmek için kullanılmaktadır.

INC: 11 numaralı porta bağlı olan buton aracılığıyla saat ve dakika üzerindeki artış işlemlerinin kontrolünü sağlar.

DECC: 9 numaralı porta bağlı olan buton aracılığıyla saat ve dakika üzerindeki azalış işlemlerinin kontrolünü sağlar.

next: 10 numaralı porta bağlı olan buton aracılığıyla saat, dakika ve alarmı onaylama işlemleri arasında geçiş yapmayı sağlar.

set\_mad: 12 numaralı porta bağlı olan buton aracılığıyla default olarak gelen saat ve tarih gösterim ekranından LCD panelin alarm ayarlama ekranına güncellenmesini sağlar.

**b.** Kullanılan Timer ve İnterrupt Verileri:

Bu kodda herhangi bir timer veya interrupt verisine yer verilmemiştir. Yalnızca zaman kontrolü için “delay()” kullanılmıştır. Ek olarak alarmın çalışıp program akışını sekteye uğratarak işlev görmesi sistem için bir interrupt niteliği taşımaktadır.

**c.** Kullanılan Fonksiyonlar ve Alt Program Çağrıları:

setup(): Programdaki arayüz, kontrol öğeleri ve devre elemanlarının başlangıç ve default değerlerinin tanımlandığı temel koddur.

loop(): Yapılan işlemlere göre ekrandaki gösterilen ögelerin sürekli güncel kalmasını ve kullanıcıya geri bildirim verilmesini sağlamak için kullanılan fonksiyondur.

defualt(): RTC entegresiden gelen saat, dakika ve saniye verilerinin tutulduğu değişkenler aracılığıyla bu verilerin ekrana yazılmasını sağlar.

time(): Alarm kurma işlemindeki saat ve dakika değerlerinin artırılıp azaltılmasını ve güncel değerlerin EEPROM belleğine atılmasını sağlayan fonksiyondur.

beep(): LED ve Buzzer devre elemanlarının 0.5 saniyede bir kez yanıp sönmesini tetikleyen fonksiyondur.

match(): Gerçek zamanlı göstergedeki saat ve dakika değerleri ile alarm olarak ayarlanan saat ve dakika değerleri eşleştiğinde yapılacak olan işlemlerin tanımlandığı fonksiyondur.

4.Doğrulama ve Test

**a.** Projenin Kısıtları

Alarm Ayarı: Kod örneği, sadece belirli bir saat ve dakika için alarm ayarı yapmaktadır. Dakika düzeyinde daha hassas bir alarm ayarı veya tekrarlayan alarm özellikleri eklemek istenirse, kodun değiştirilmesi gerekebilir.

Kullanıcı Arayüzü: Kod, sadece düğmelere basılarak saat ve alarm ayarı yapılabilen basit bir kullanıcı arayüzü sunmaktadır. Gelişmiş bir kullanıcı arayüzü veya daha interaktif bir kontrol gerektiğinde, kodun genişletilmesi veya değiştirilmesi gerekebilir.

Alarm İşlevselliği: Kod, belirli bir saat ve dakikaya ayarlanan alarmı kontrol etmektedir. Ancak, alarm tetiklendiğinde yapılacak ekstra eylemler (örneğin, ses düzeyi değişimi vb.) kodda belirtilmemiştir. Bu nedenle, alarm tetiklendiğinde yapılacak özel eylemler için ek kod eklenmesi gerekebilir.

Bellek Sınırlamaları: EEPROM kullanımı, belirli bir hücre aralığında alarm saat ve dakikalarını saklamaktadır. Ancak, EEPROM'un sınırlı bir bellek kapasitesi vardır ve daha fazla veri saklamak veya daha fazla alarm ayarı eklemek istenirse, bellek sınırlamaları göz önünde bulundurulmalıdır.

Donanım Bağımlılığı: Kod, özel bileşenler olan RTC DS1307 ve LiquidCrystal kütüphanelerini kullanmaktadır. Bu nedenle, bu bileşenlere fiziksel olarak erişilebilir olmalı ve doğru şekilde bağlanmış olmalıdır. Ayrıca, kullanılan Arduino modeline ve pin bağlantılarına da bağımlıdır.

**a. i.** Projede Neler Yapıldı:

Arduino ile bir alarm saati uygulaması yaparken aşağıdaki adımları takip ettik:

İhtiyaç Analizi:

Alarm saati uygulaması için temel gereksinimleri belirledik. Bu gereksinimler, kullanıcıya kolay bir kullanıcı arayüzü sunma, alarm zamanını ayarlama, alarm çalarken uygun aralıklarla led ışığı yakma ve alarm bildirimini görüntüleme gibi işlevleri içeriyor.

Donanım Seçimi:

Proteus platformunu kullanarak projemizi gerçekleştirmek için bir Arduino kartı seçtik. Ayrıca bir RTC entegresi, dört düğme, bir LED ve bir LCD ekran gibi ek donanımları da projemize dahil ettik.

Devre Tasarımı:

Alarm saati devresini tasarladık. Arduino kartına RTC’yi, düğmeleri, LED’i ve LCD ekranı bağlamak için gerekli bağlantıları yaptık.

Arduino Kodlama:

Arduino IDE isimli bir yazılım ortamında Arduino kartı için uygun bir program yazdık.

Alarm zamanını ayarlamak için düğmeleri kullanarak kullanıcı arayüzünü oluşturduk. LCD ekranda saat, dakika, saniye, tarih ve alarm ayarları görüntülendi.

Alarm zamanı geldiğinde LED’de uygun uyarı ışığının gösterilmesi için gerekli kodları yazdık.

Test ve Hata Giderme:

Alarm saati uygulamasını Arduino kartına yükledik ve donanımı test ettik.

Alarm zamanını ayarlamayı ve alarmın çalışmasını sağlayan işlevlerin doğru çalıştığından emin olduk.

Karşılaştığımız hataları tespit edip gidermek için adımlar attık.

Raporlama:

Uygulama sürecimizi adım adım belgeledik.

Donanım bileşenlerinin seçimini ve bağlantılarını açıkladık.

Arduino kodunu açıklayıcı bir şekilde yazdık ve kullanılan fonksiyonları belirttik.

Test sonuçlarını ve karşılaşılan hataları raporladık.

Uygulamanın çalışması için gereken kaynakları listeledik.

Bu adımları takip ederek Arduino ile alarm saati uygulamasını başarıyla gerçekleştirdik.

**a. ii.** Projenin Özgünlüğü ve Pratik Yaşama Katkısı:

Bu proje sayesinde kullanıcılar ayarlanabilir saatlerini kendi ihtiyaçlarına ve zevklerine göre özelleştirebilirler. Saat ayarındaki her değer tamamen kullanıcıya bağlıdır. Ayrıca bu sayede insanlara zaman yönetimi konusunda yardımcı ve yoğun iş planlarında hatırlatıcı görevi görmüş olacaktır. Günümüz ayarlanabilir saatlerine kıyasla sadece değeri arttırma değil aynı zamanda azaltma özelliği de vardır. Bu sayede kullanıcı yanlışlıkla istenen değeri geçtiği zaman daha kısa sürede tekrardan amacına ulaşabilecektir.

**a. iii.** Eksiklikler:

Projenin eksiklikleri olarak, belirli bir takvim için hatırlatma sağlanması, ya da kurulan alarmın hangi amaçla kurulduğunu anımsatması amacıyla isimlendirilmenin değiştirilebilmesi ya da kullanıcıya özel kişiselleştirme sağlanması eklenebilirdi. Daha renkli görüntüler kullanıcı memnuniyeti için iyi bir fikir olabilirdi.

**a. iv.** Olması Gereken Diğer Durumlar:

Projede ayarlanabilir saat formatı, isteğe göre 12 saatlik ya da 24 saatlik saat formatları, alarm çaldıktan sonra alarmı durdurma ya da kapatma, saniye bilgisinin gösterilmesi, açma kapama tuşlarının olması diğer beklenen durumlardır.

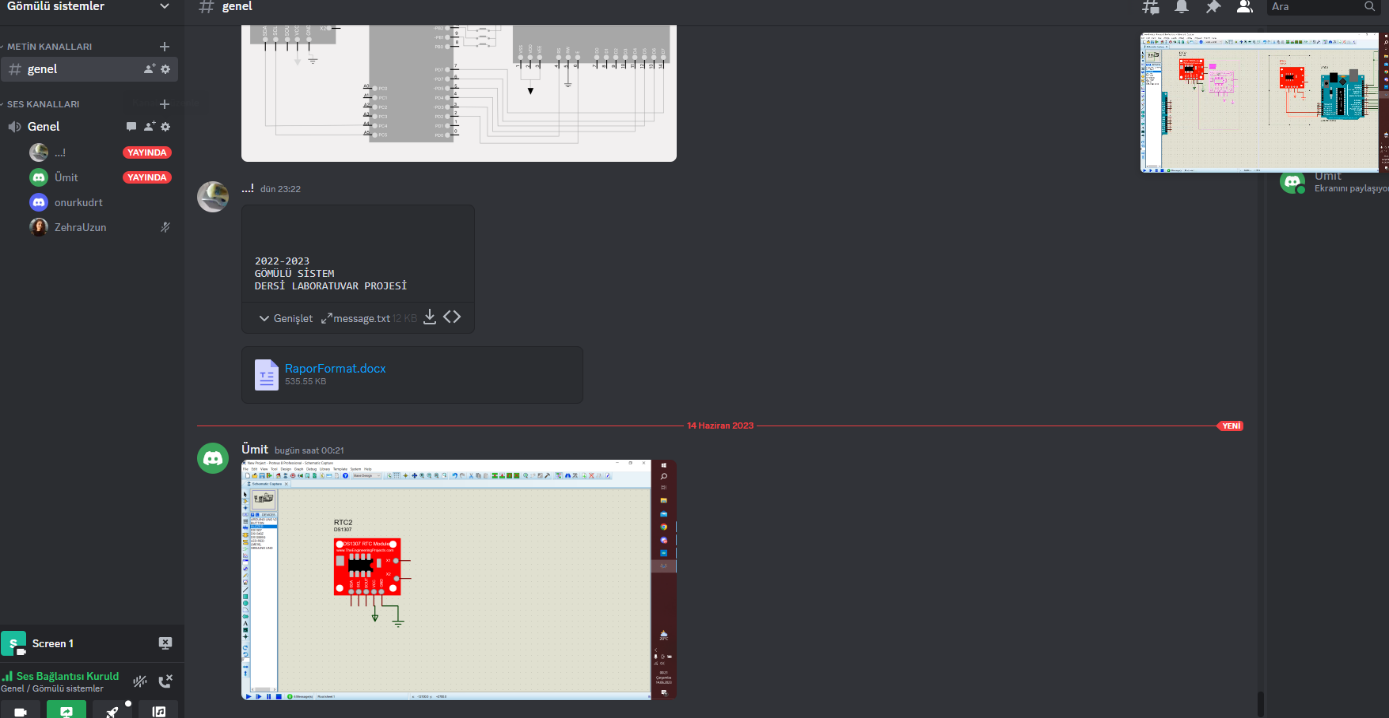
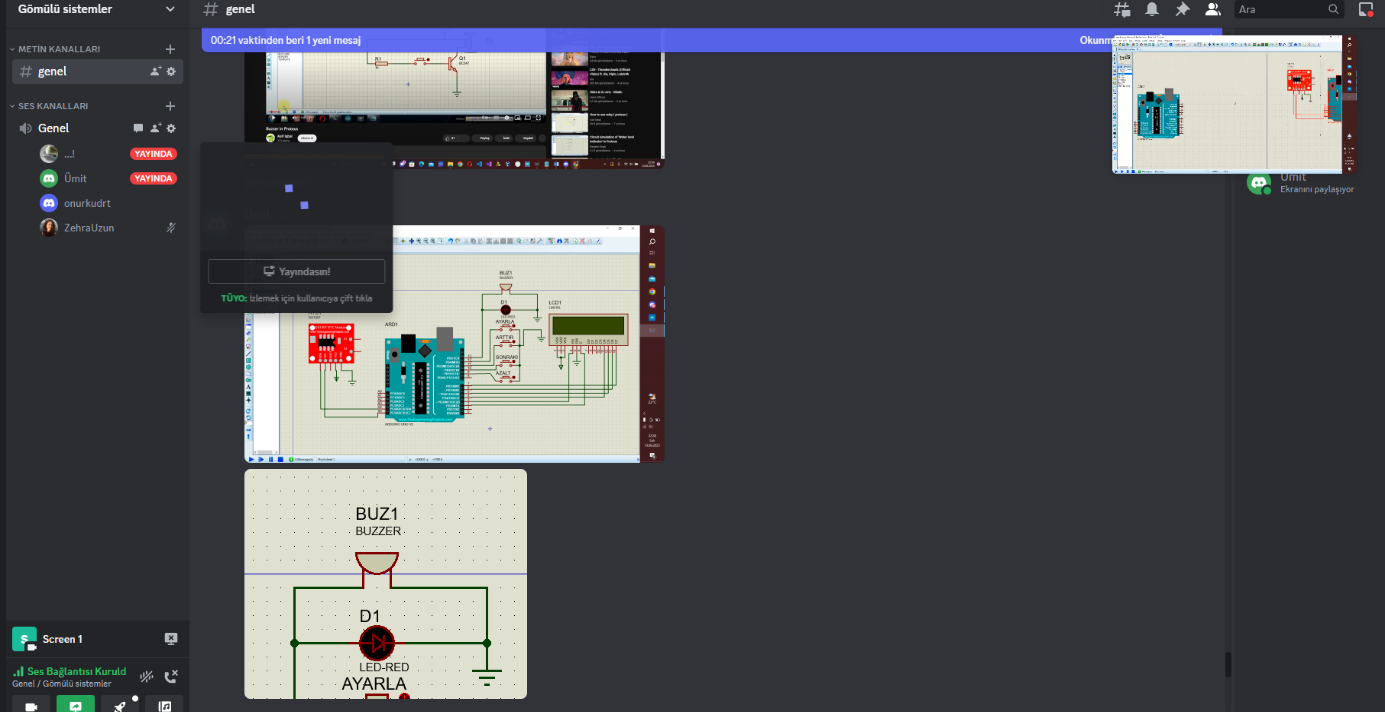
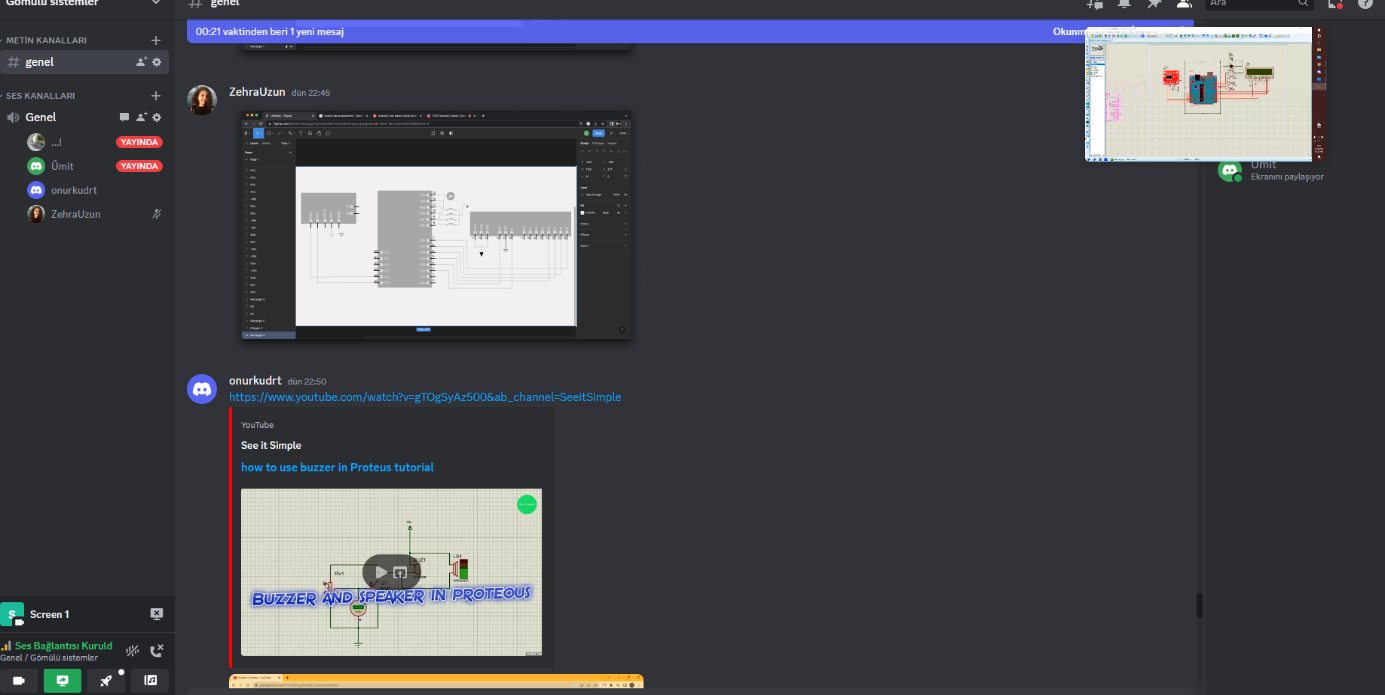
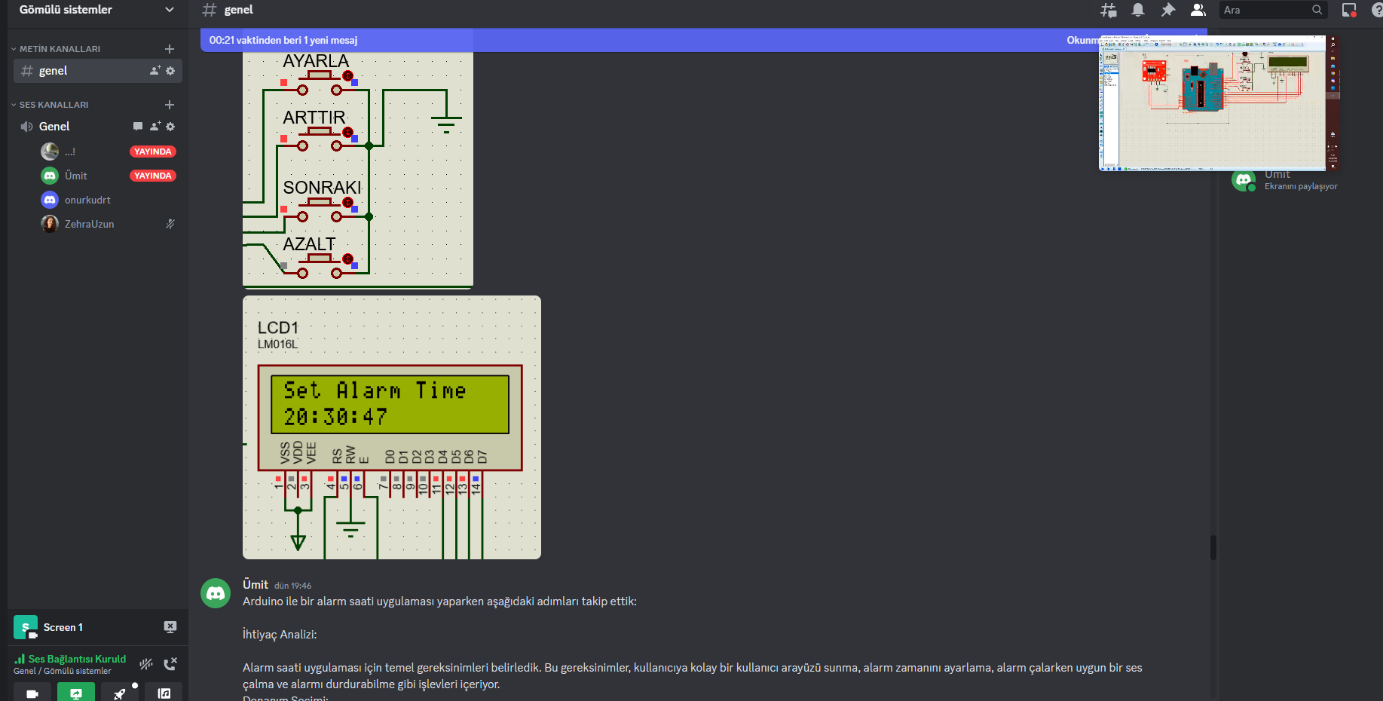
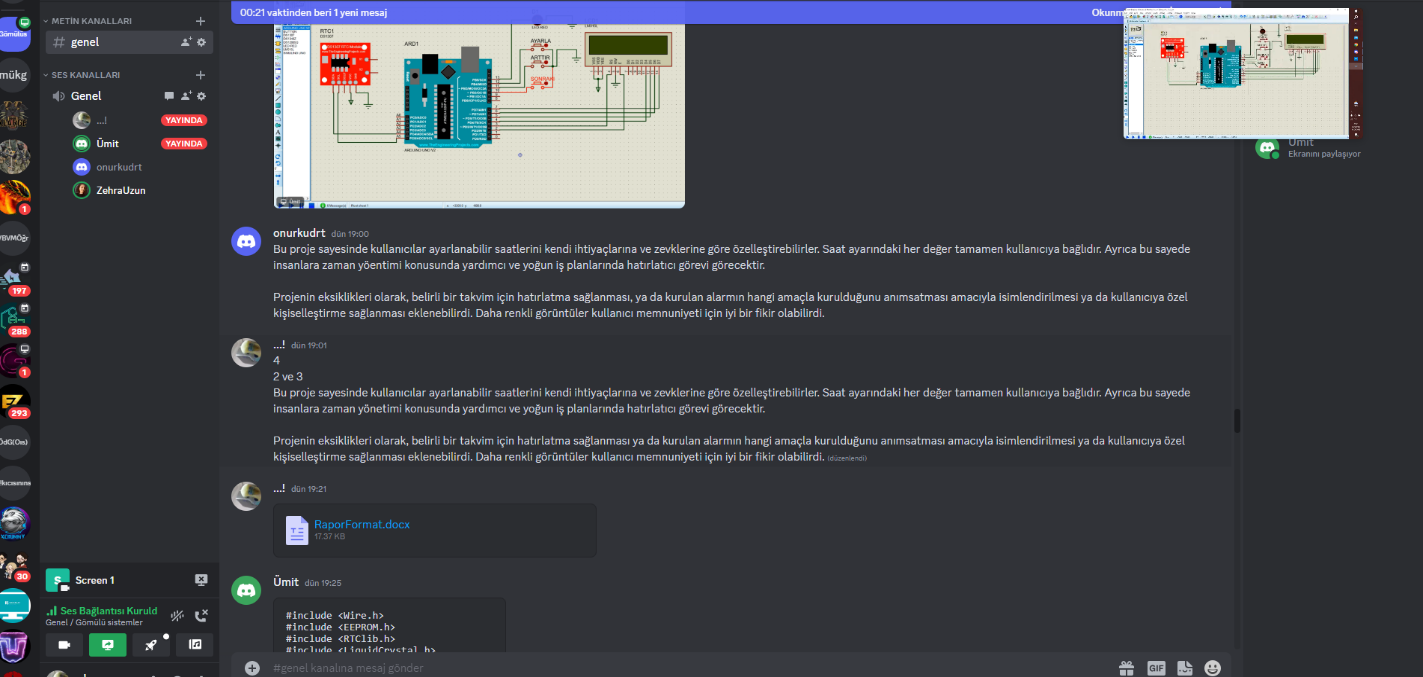
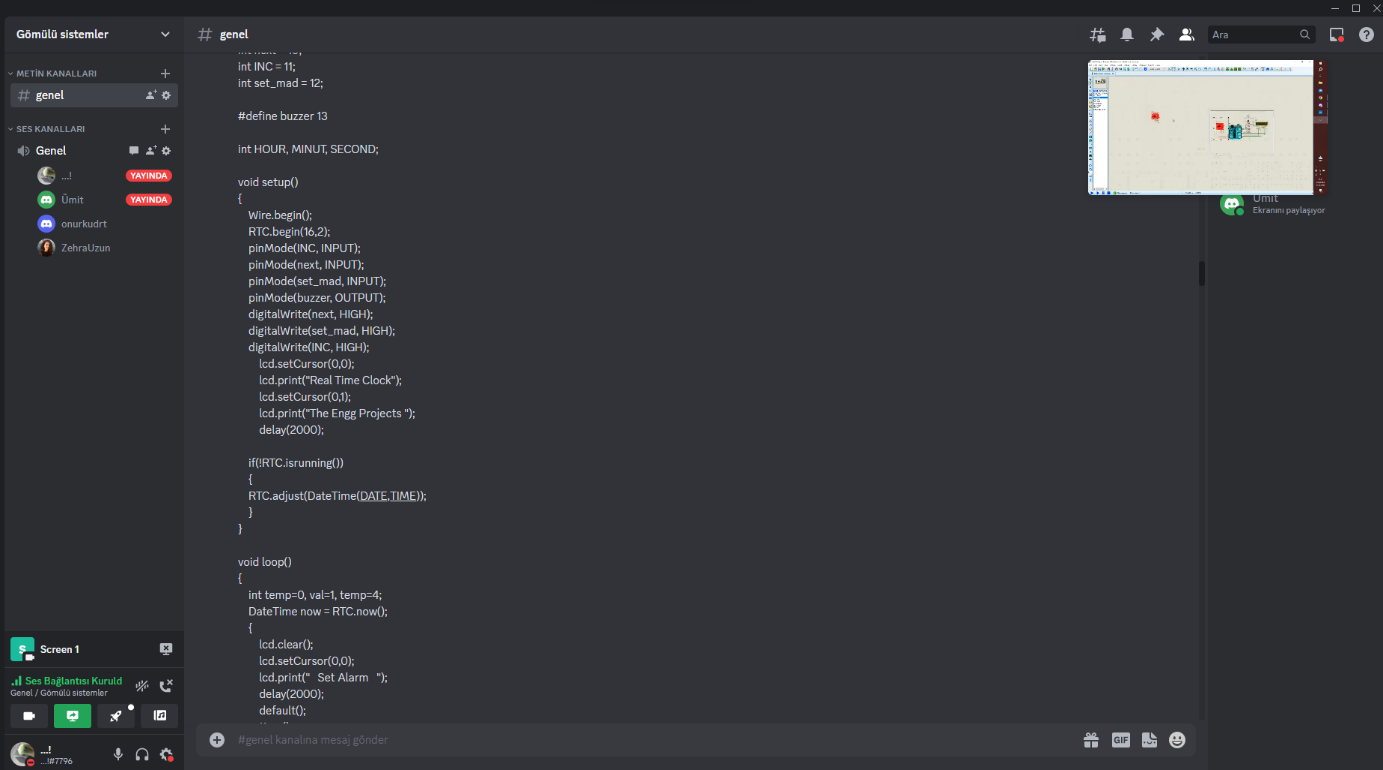
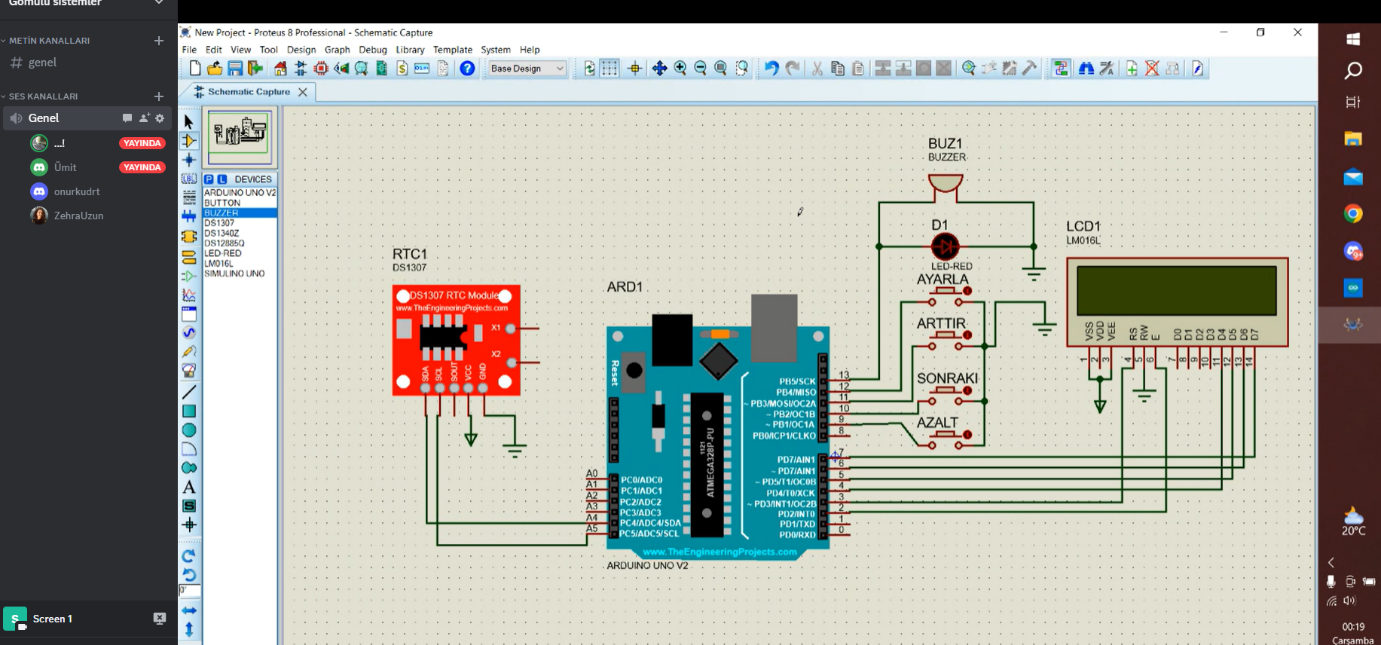
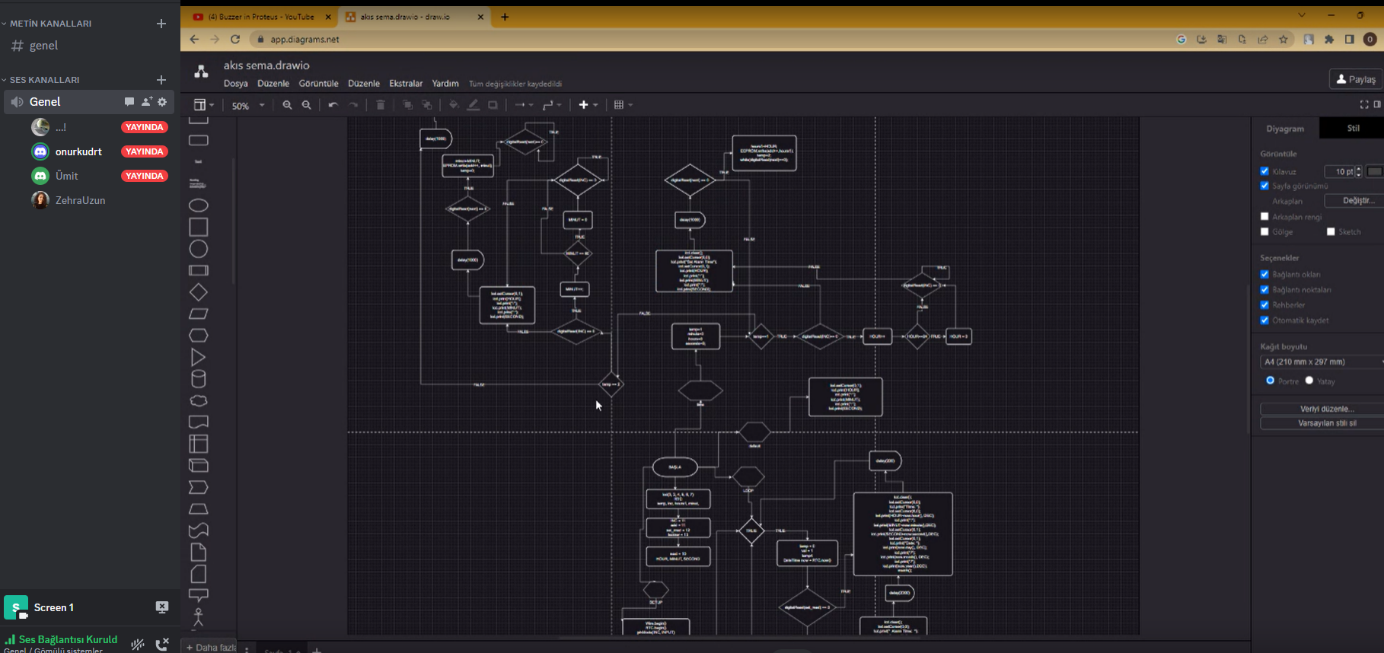
**b.** Beklenen Sonuçlar:

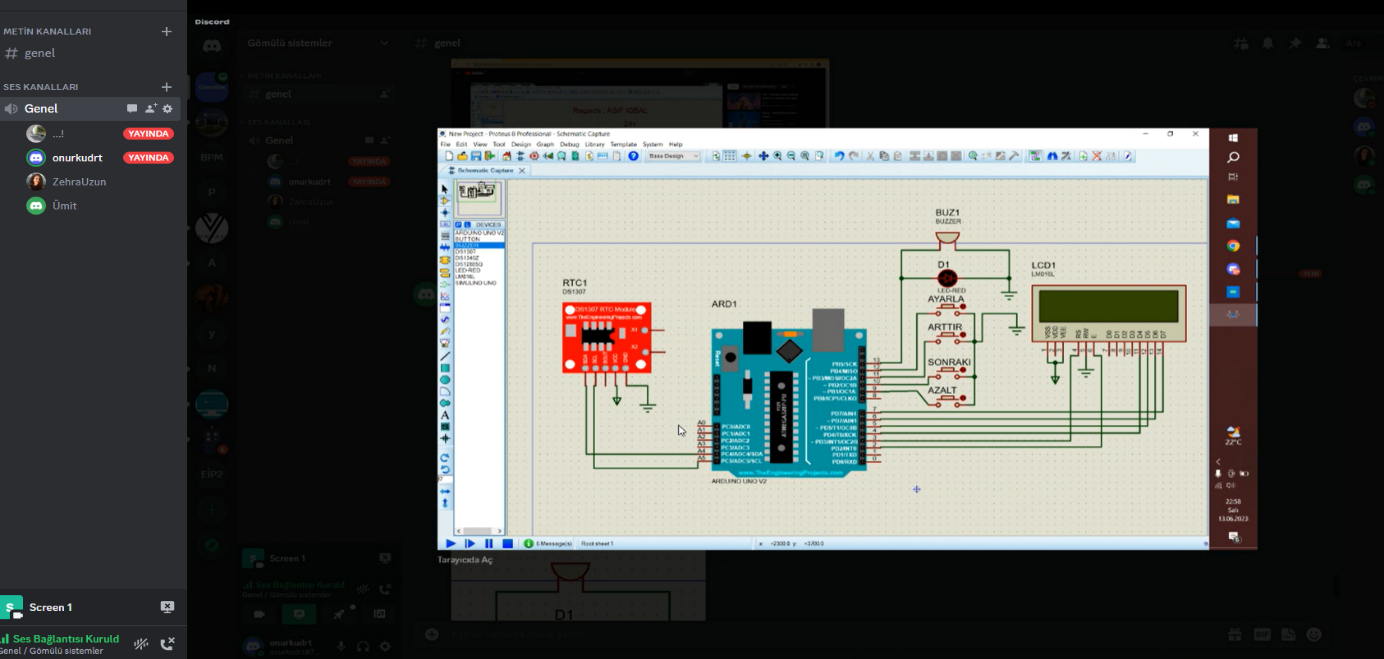
Kullanıcı tarafından başlatılan uygulamada öncelikle gerçek zamanlı saatin ve tarihin RTC entegresi ile eşzamanlı olarak doğru bir şekilde LCD ekran üzerinde gösterilmesi gerekmektedir. Daha sonrasında alarm ayarlamak için gerekli butona basılarak saat ve dakika istenen şekilde artırıp azaltılarak ayarlanıp ana ekrana geri dönülür. Bir sonraki adımda güncel saat ile alarmın kesişim anında LED üzerindeki kırmızı ışık yanıp sönerek “Wake Up….” şeklinde bir uyarıyla bizi karşılamalıdır. Tüm bunlar programın doğru bir şekilde çalışması için beklenen sonuçlardır.

**c.** Görev Paylaşımı ve Toplantılardan Görüntüler:

Yapmış olduğumuz projede tüm süreçleri bir arada gerçekleştirdik. Görev paylaşımı, yapılan her unsur için eşit ve ölçülü bir biçimde yapılmış oldu. İş akışında kimse tek başına çalışmadı, Discord üzerinden anlık olarak iletişimde bulunup projenin tamamını paylaşımlı bir şekilde yürütmüş olduk.

Projenin Toplantı Görüntüleri:





Projenin Yapım Aşaması:

metin, ekran görüntüsü, yazılım, bilgisayar simgesi içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, ekran, görüntüleme, ekran görüntüsü, yazılım içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, ekran görüntüsü, yazılım, multimedya yazılımı içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, ekran görüntüsü, yazılım, multimedya yazılımı içeren bir resim

Açıklama otomatik olarak oluşturuldu metin, ekran görüntüsü, yazılım, bilgisayar simgesi içeren bir resim

Açıklama otomatik olarak oluşturuldu