

**2020-2021**

**MICROCONTROLLER BASED SYSTEM DESIGN**

**LABORATUVAR PROJESİ**

**Programlanabilir Çalar Saat ve Takvim**

**GRUP NO: 1**

**05170000022, Mehmet Anıl TAYSI**

**05180000087, Emel KAYACI**

**TARİH**

**08.07.2021**

## 1. İhtiyaç ve Proje Analizi

### a. Projenin amacı ve hedefi

Projemizin amacı, 8051 mikroişlemcisini kullanarak saat ve tarihin belirlenebildiği, bu saat ve tarih üzerinde isteğe göre ayarlamalar yapılabilen bir uygulama geliştirmektir. Bunun yanında, kullanıcıya bir alarm ayarlama imkanı da verilmek istenmektedir.

### b. Kullanıcıya yönelik faydaları

Geliştirilen bu uygulama, herhangi bir otomasyon sisteminde (otomat, mikrodalga, çeşitli tabelalar vb.) kullanılabilir, kullanıcı istediği takdirde alarm oluşturup, bunu amacına uygun şekilde kullanabilecektir.

### c. Kullanılan yazılım dili, derleyici veya diğer araç gereksinimleri

**Kullanılan Yazılım Dili:** C

**Derleyici:** Keil mikroVision5, devre tasarımı için Proteus Professional 8.12

Yazılım C dilinde Keil ortamında yazılmış olup, devre tasarımı ve test işlemleri Proteus 8.12 yazılımı üzerinde yapılmıştır.

## 2. Tasarım

### a. Proje tasarımında kullanılan devre elemanları

*AT89C51 – 8051 Mikroişlemci*

*LM044L LCD*

*2 adet 33 pikoFarad Kapasitör*

*Crystal*

*Push Button*

*Buzzer*

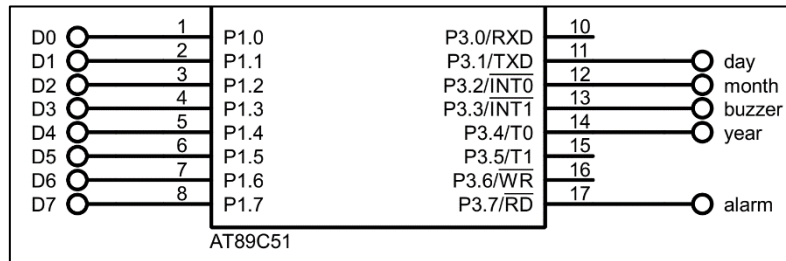
*330R Resistor*

*BC547 Transistor*

*Ground*

### b. Projenin devre şeması

Projenin devre şemasını paylaşmadan önce önemli bir noktaya değinmek istiyoruz. Devreyi oluştururken örneğin LCD ile 8051'in pinlerini birleştirirken görünen kablo arayüzlerinin devreyi çok karmaşık gösterdiğini fark ettik ve 'label atama' yöntemi ile devremizi gerçekleştirdik.



**Şekil 1.** Kabloların label atama yöntemi ile ayarlanması

The diagram illustrates a digital clock circuit using an AT89C51 microcontroller. The microcontroller is connected to an LCD144L display, a buzzer, and various input/output pins. The circuit includes a crystal oscillator (X1) with 33pF capacitors (C1, C2) and a power supply section with a buzzer and a BC547 transistor (Q1).

**Microcontroller Pin Connections:**

- XTAL1:** Connected to pin 19.
- XTAL2:** Connected to pin 18.
- RST:** Connected to pin 9.
- PSEN:** Connected to pin 29.
- ALE:** Connected to pin 30.
- EA:** Connected to pin 31.
- P0.0/A0:** Connected to pin 39.
- P0.1/A1:** Connected to pin 38.
- P0.2/A2:** Connected to pin 37.
- P0.3/A3:** Connected to pin 36.
- P0.4/A4:** Connected to pin 35.
- P0.5/A5:** Connected to pin 34.
- P0.6/A6:** Connected to pin 33.
- P0.7/A7:** Connected to pin 32.
- P2.0/A8:** Connected to pin 21.
- P2.1/A9:** Connected to pin 22.
- P2.2/A10:** Connected to pin 23.
- P2.3/A11:** Connected to pin 24.
- P2.4/A12:** Connected to pin 25.
- P2.5/A13:** Connected to pin 26.
- P2.6/A14:** Connected to pin 27.
- P2.7/A15:** Connected to pin 28.
- P3.0/RXD:** Connected to pin 10.
- P3.1/TXD:** Connected to pin 11.
- P3.2/INT0:** Connected to pin 12.
- P3.3/INT1:** Connected to pin 13.
- P3.4/T0:** Connected to pin 14.
- P3.5/T1:** Connected to pin 15.
- P3.6/WR:** Connected to pin 16.
- P3.7/RD:** Connected to pin 17.

**Other Components and Connections:**

- LCD144L:** Connected to pins 1 (VSS), 2 (VDD), 3 (VEE), 4 (RS), 5 (RW), 6 (E), 7 (D0), 8 (D1), 9 (D2), 10 (D3), 11 (D4), 12 (D5), 13 (D6), 14 (D7).
- Buzzer:** Connected to pin 26 (DEC) and pin 17 (alarm).
- BC547 Transistor (Q1):** Connected to pin 26 (DEC) and pin 17 (alarm).
- Resistor R2:** Connected to pin 26 (DEC) and pin 17 (alarm).
- Capacitors:** C1 and C2 are 33pF capacitors connected to the crystal oscillator.

Tasarımın detayları D seçeneğinde açıklanmıştır.

```

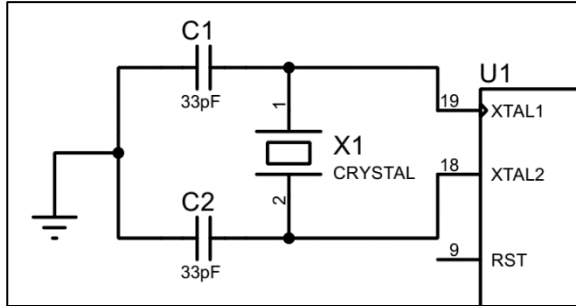
graph TD
    Start([Start]) --> LCD_Activated[LCD Activated]
    LCD_Activated --> Is_Alarm_Gate_Active{Is Alarm Gate Active?}
    Is_Alarm_Gate_Active -- NO --> Alarm_Inactive[Alarm Inactive]
    Alarm_Inactive --> Time_Values_Selected[Time Values Selected]
    Is_Alarm_Gate_Active -- YES --> Alarm_Active[Alarm Active]
    Alarm_Active --> Alarm_Values_Selected[Alarm Values Selected]
    Alarm_Values_Selected --> Is_Decrement_Gate_Active{Is Decrement Gate Active?}
    Time_Values_Selected --> Is_Decrement_Gate_Active
    Is_Decrement_Gate_Active -- NO --> Increase_Value[Increase Value]
    Increase_Value --> Is_Alarm_Gate_Active
    Is_Decrement_Gate_Active -- YES --> Decrease_Value[Decrease Value]
    Decrease_Value --> Is_Alarm_Gate_Active
    Alarm_Active --> Time_Equals_Alarm{TIME == ALARM}
    Time_Equals_Alarm --> Buzzer_Activated[Buzzer Activated]
    Buzzer_Activated --> Is_Alarm_Gate_Active

```

3

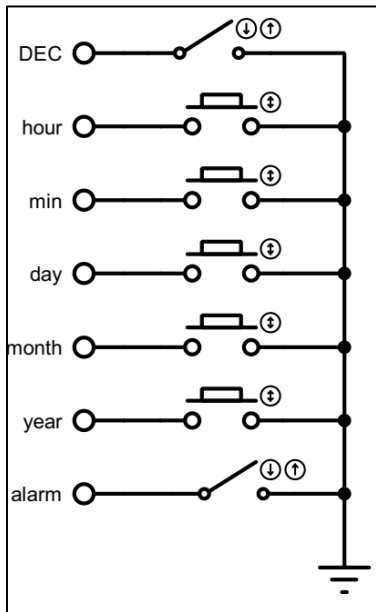
Programımız başladığında LCD aktif hale geliyor ve Tarih ve Saat değerleri default değerlerinde başlayarak çalışıyor. Eğer Alarm için oluşturulan kapı aktif değilse, kullanıcı saat ve tarih üzerinde değişiklikler yapabiliyor (artırma ve azaltma), eğer DEC kapısı aktifse her tuşa tıklandığında ilgili değer bir azalıyor, aktif değilse bir artıyor (increase işlemi oluyor). Eğer alarm kapısı aktifse, saat butonları alarm değerleri için çalışmaya başlıyor. Ve nihayetinde saatin değerleri (dakika ve saat) alarm verilerine (dakika, saat) eşit olursa, buzzer aktif oluyor ve alarm çalmaya başlıyor.

d. Tasarım ekran görüntüleri



İki 33pF'lik kapasitörün kuvars kristali (12 MHz) ile bu paralel kombinasyonu, 8051'e, 11.0592 MHz'lik tam saat frekansını sağlar.

Şekil 4. Saat frekansının ayarlanması



Saat, tarih ve alarm ayarları kısmında 5 buton ve 2 tane anahtar kullanılmıştır.

DEC anahtarı kapalı olduğunda (DEC == 0) butonlara basıldıkça istenilen veriler birer birer azalmaya başlar, açık iken ise artar.

Alarm anahtarı kapalı durumdayken saat ve dakika butonları alarm zamanı ayarını yapmaktadır. Saat ayarına dönebilmek için anahtarı açmak gerekir.

Ayrıca alarm anahtarı açık durumda ise alarm LCD ekranında görünmez ve saat geldiğinde bile buzzer çalışmaz. Böylece kullanıcıya istediği zamanlarda alarm kurabilme seçeneği de sunulmuştur.

Şekil 5. Ayarların yapılması

Character No.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1st line	80	81	82	83	84	85	86	87	88	89	8A	8B	8C	8D	8E	8F	90	91	92	93
2nd line	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF	D0	D1	D2	D3
3rd line	94	95	96	97	98	99	9A	9B	9C	9D	9E	9F	A0	A1	A2	A3	A4	A5	A6	A7
4th line	D4	D5	D6	D7	D8	D9	DA	DB	DC	DD	DE	DF	E0	E1	E2	E3	E4	E5	E6	E7

Şekil 6. Kullanılan LCD'nin Karakter adres haritası

Kullandığımız LCDnin datasheet sayfası üzerinden çeşitli bilgilere ulaştık ve bu bilgileri kullanarak Tarih saat ve Alarm kısımlarını LCD'ye yerleştirdik.

### 3. Kodlama

- a. Kullandığınız giriş ve çıkış değişkenleri (portlar, vs.)

LCD'de verileri görüntülemek için P1 portunun tamamı, kontrol pinlerini ayarlamak için de P2 portunun 1, 2 ve 3. Pinleri kullanılmıştır.

**P2.1** ile LCD'nin RS pini bağlanmıştır. RS (Register Select) pini LCD'nin veri (RS=1) ya da komut (RS=0) modunu ayarlamak için kullanılır.

**P2.2** ile LCD'nin RW (Read/Write) pini bağlanmıştır. LCD'den herhangi bir okuma yapmadığımız için write (RW=0) moduna göre ayarlanmıştır.

**P2.3** ile LCD'nin E (Enable) pini bağlanmıştır. Yüksek pulse değerinden küçük pulse değerine geçişte LCD'yi etkinleştirir.

**P1** portunun tamamı LCD'ye verilerin aktarılmasından sorumludur.

Tarih ayarlarının yapılabilmesi için **P3.1**, **P3.2** ve **P3.4** pinleri kullanılmıştır. Bu pinler sırasıyla gün, ay ve yıl ayarlarını yapan butonları kontrol eder.

Saat ayarlarının yapılabilmesi için **P2.6**, **P2.7** kullanılmıştır. Bu pinler sırasıyla saat ve dakika ayarlarını yapan butonları kontrol eder.

DEC anahtarı, **P2.5** pinini kullanmaktadır. Bu pinden aldığı değere göre butonların sorumlu oldukları değerleri artırma (DEC=1) veya azaltma (DEC=0) modunda kullanılacağını belirler.

Alarm anahtarı, **P3.7** pinini kullanmaktadır. Bu pinden aldığı değere göre (alarm=0) alarmın saat ve dakika ayarları yapabilmemesini sağlar. Ayrıca ekranda alarmın görüntülenmesini ve alarmın kurulup kurulmayacağını (zilin çalıp çalmaması) kontrol eder.

**P3.3** pinine bağlı olan buzzer alarm ziline çalmasını kontrol eder. Bu işlem interrupt servis rutini ile gerçekleştirilir.

**XTAL1** ve **XTAL2** pinleri harici saat frekansını sağlamak için kullanılır. Bizim devremizde bu frekans 11.0594 MHz olarak belirlenmiştir.

- b. Kullandığınız Timer ve Interrupt verileri, ne amaçla kullanıldığı, Kullandığınız fonksiyonlar veya alt program çağrılar, ne amaçla yazıldığı

**Timer** fonksiyonu, timer ayarlarını yapmak için kullanılır. TMOD, 0x01 (16 bitlik mod) olarak belirlenmiştir. 1 milisaniyelik gecikmeler yaratmak için TL ve TH registerlarına uygun değerleri atamamız gerekir.

1 milisaniyelik gecikmeyi şu şekilde yaratılmıştır: 16 bitlik timer'ın maksimum 65536 ( $2^{16}$ ) mikrosaniye sayabilmektedir. Fakat bizim yalnızca 1000 mikrosaniyeye ihtiyacımız var. Bu nedenle timer'ın başlangıç değerini buna uygun ayarlamamız gerekir.  $65536 - 1000 = 64536$  olup bizim timer'ımızın başlangıç değerini ifade eder.

64536'in hexadecimal formu FC18'dir. Böylece high bitleri (FC) TH registerına, low bitleri (18) TL registerına yükleyip 1 milisaniye ayarını gerçekleştiririz.

ET0 ve EA pinleri IE (Interrupt register) içinde bulunur. EA pini 1 olarak ayarlanmış olup her interrupt kaynağının kendi bitini ayarlayarak çalışmasını sağlar. Bizim projemizde bu kaynak zili etkinleştiren interrupt olup işi bitince TF0 = 0 ile interrupt flag değerini temizler. ET0 ise Timer 0 için interrupt etkinleştirme bitidir.

Milisaniye sayma işlemi timer fonksiyonunun sonundaki döngüde overflow (taşma) kontrolü ile gerçekleşir.

**Interrupt servis rutini** içerisinde gereken koşullar (alarmın ekranda yer alması ve alarm zamanının gelmesi) sağlandığı takdirde zilin çalması gerçekleştirilir. Programımızda alt program çağırımını da bu şekilde gerçekleştirdik.

**LCD komut fonksiyonu**, LCD'ye özel komutların çalıştırılmasını sağlar. Örnek komutlar şunlardır: 0x80 imleci birinci satırın birinci pozisyonuna getirir. 0x06 gibi imleci sağa kaydırır. 0x38 ise 2 satır ve 5x7'lik matris yazma alanı oluşturur.

**LCD görüntüleme fonksiyonu**, LCD'ye verilerimizi yazdırmak için kullanılır. RS biti yukarıda da bahsediliği üzere 1'e ayarlanmış olup veri modunda olduğunu belirtir.

**Main fonksiyonumuz** asıl programımızı içerir. Timer, 1000 kez çağırılıp 1 saniyelik delay oluşturulur böylece saniye ilerleyişi gerçekleşir. Fonksiyon içerisinde bulunan 4 adet for döngüsü ile saniye, dakika, saat ve gün ilerleyişi gerçekleşir. Ayarlar kısmı da burada yer alır böylece kullanıcı uygun butonları kullanarak tarih, saat ve alarmı düzenler.

#### 4. Doğrulama ve Test

##### a. Projenizin kısıtları

###### i. Projede neler yapıldı?

‘Programlanabilir Çalar Saat ve Takvim’ uygulaması amacı olan Tarih, zaman ve alarm özellikleri 8051 mikroişlemci kullanılarak gerçekleştirildi.

###### ii. Projenizin özgünlüğü ve pratik yaşama katkısı nedir?

Pratik yaşama katkı olarak herhangi bir cihazın eklentisi olarak böyle bir programın kullanılabileceğini düşünüyoruz, alarm özelliği de o cihaz için faydalı olabilir.

###### iii. Projenin eksiklikleri neler?

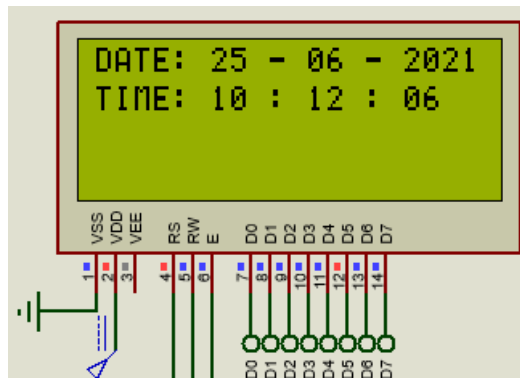
Proje istenilen tüm gereklilikleri sağlamaktadır. Detaylı kontrollerle alarm çalması, aylara göre gün sayılarının doğruluğu, dakika, saat işleyişi incelenmiştir.

###### iv. Projede olması gereken diğer durumlar

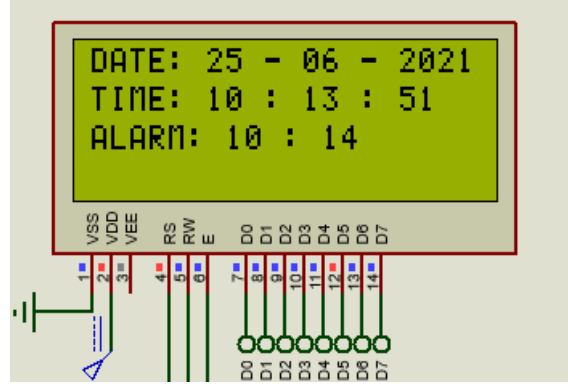
Proje istenilen tüm fonksiyonellik sağlanmaktadır. Proje şu şekilde geliştirilebilir: kullanıcının isteğine bağlı olarak buton yerine keypad eklenebilir, alarmın yanı sıra kronometre vs. eklenebilir. RTC modülü kullanılıp gerçek zamanlı saat verisi projeye eklenebilirdi. Fakat daha önce hiç RTC kullanmadığımızdan ve verilerine ulaşmak kodu karmaşık hale getirdiğinden bunu eklemekten vazgeçtik.

##### b. Projeden beklenen sonuçlar

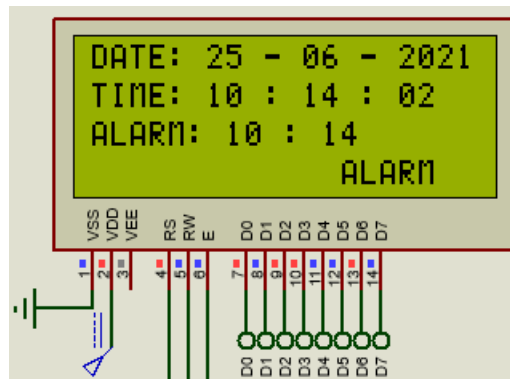
Projemizden beklenen yegane sonuç, tarih ve saat gösteriminin doğru ve anlaşılır bir şekilde olması, ayrıca alarm özelliğini de doğru bir şekilde yerine getirebilmesidir. Örnek bir çıktı (LCD) ekran görüntüsü,



Bu görüntüde alarm aktif değil, tarih ve zaman şekildeki gibi görünmekte.

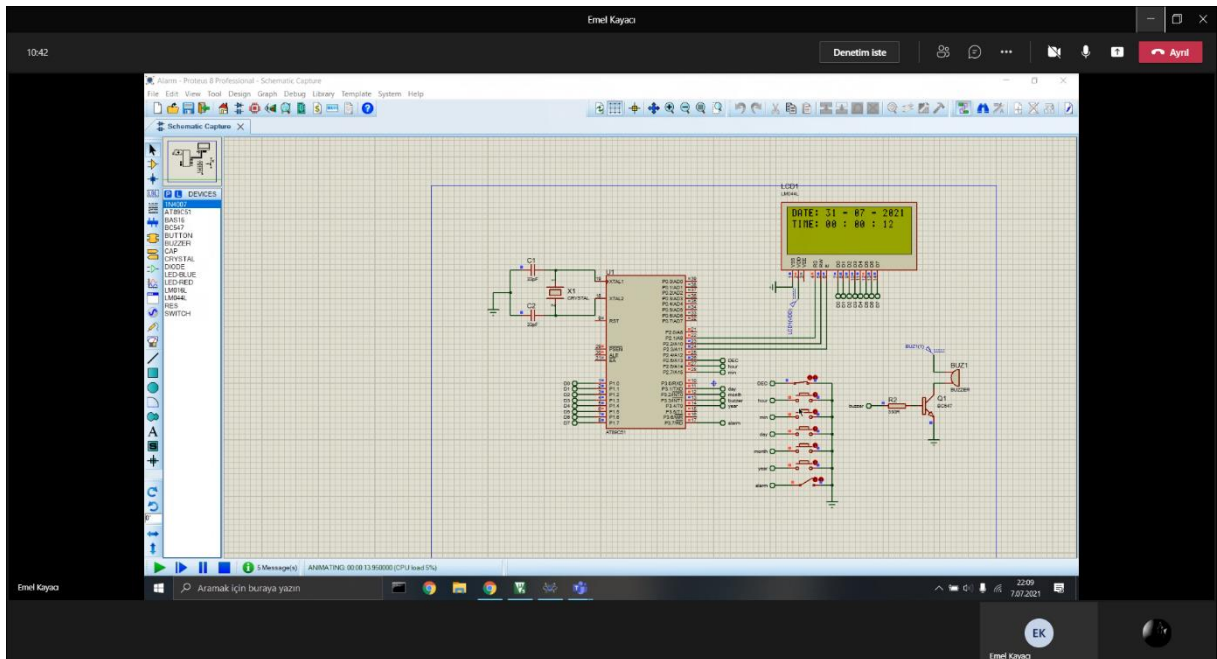


Alarm switch'i aktif hale getirilmiş ve alarm kurulmuş vaziyette. Alarm switch'i inaktif hale getirilene kadar alarm kurulu vaziyette kalır.



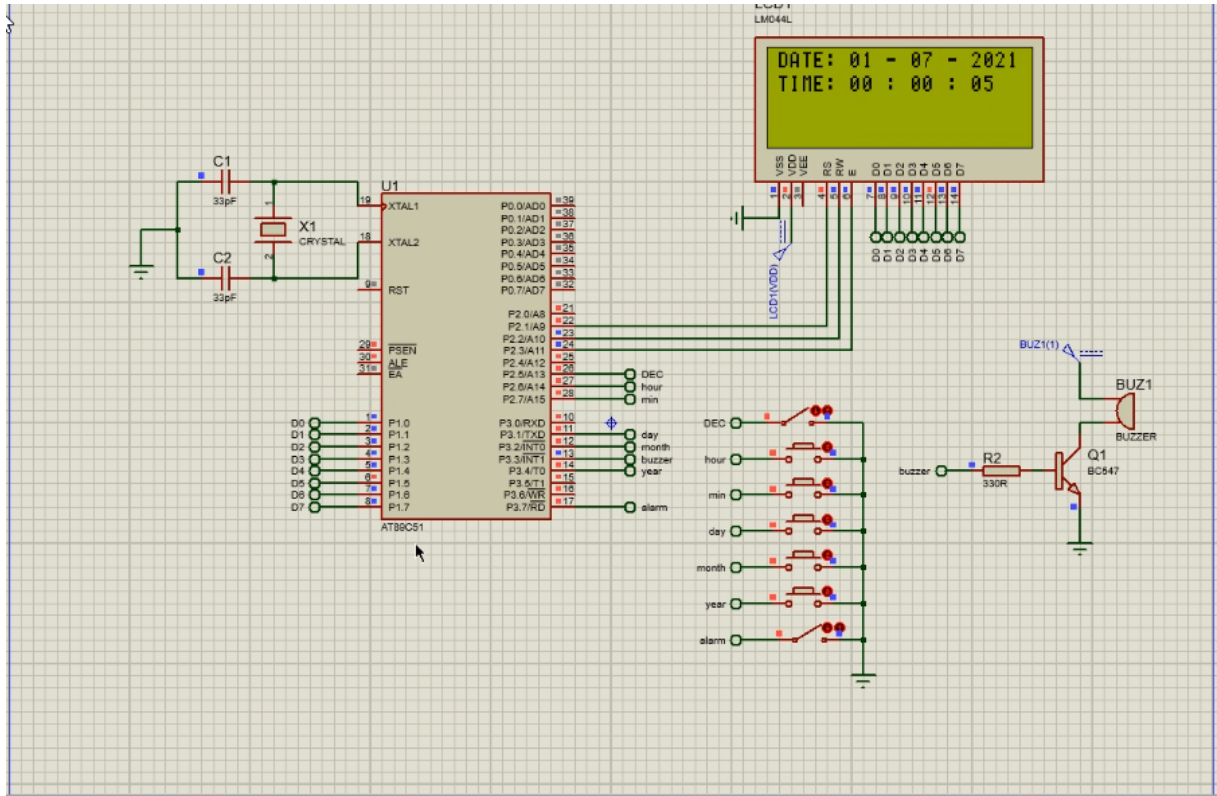
Alarm vakti geldiğinde hem LCD'de Alarmı niteleyen bir yazı belirir, hem de buzzer aktif hale gelir.

- c. Ortak çalışma adına gerçekleştirilen görev paylaşımları ve toplantılardan görüntüler



Teams ortamında Projenin Tasarımı üzerine denemeler yapıyoruz.





Zaman sayacının gün ay ve yıl kısımlarına yaptığı etkileri gözlemek adına denemeler yapıyoruz.

```

89 //
90 h=0;
91 m=0;
92 s=0;
93 y=2021; //başlangıç tarihi
94 m=7;
95 d=1;
96 con=0;
97 for(mo=1; mo=12; mo++)
98 for(l=1; l=d; l++)
99 for(h=0; h=23; h++)
100 for(m=0; m=59; m++)
101 for(s=0; s=59; s++)
102 if(mo==1 || mo==3 || mo==5 || mo==7 || mo==8 || mo==10 || mo==12) {
103 d=31;
104 }
105 else if(mo==2) {
106 if((y%4)==0) {
107 d=29;
108 }
109 else {
110 d=28;
111 }
112 }
113 else {
114 d=30;
115 }
116 if(alarm==0) {
117 h3 = h/10; //alarmda saatı göstermek için
118 h4 = h%10;
119 m3 = m/10; //dakikayı göstermek için
120 m4 = m%10;
121 s3 = s/10;
122 s4 = s%10;
123 }

```

Proteusta işleyişi gözleyerek, proje kodu üzerinde değişiklikler yapıyoruz.

## Görev Paylaşımı

Projenin kodlama ve tasarım kısımları birbirlerine kenetli durumdadır, hal böyle olunca her iki aşamada da eşit süreleri ayırdık diyebiliriz.