

KOCAELİ ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ

Sıcaklığı Ayarlanabilir Su Isıtıcısı

Mühendislik Tasarımı 1

Burak Yeniaydın

Bölümü: Elektronik ve Haberleşme Mühendisliği

Danışman: Prof. Dr. Oğuzhan Urhan

KOCAELİ, 2021

ÖNSÖZ ve TEŞEKKÜR

Son zamanlarda yaşanan gelişmelere birlikte kullanıcı kontrollü akıllı ev cihazlarının kullanımı yaygınlaştı. Su ısıtıcıları da bu cihazlardan biridir.

Su ısıtıcıları genellikle belirli bir sıcaklık değerine kadar çalışan, o değerden sonra çalışmayı durduran bir mekanizmaya sahiptir. Bu projede ise suyun sıcaklık değeri kullanıcı tarafından ayarlanabilmektedir. Su, sistem kapatılana kadar aynı sıcaklık seviyesinde kalmaya devam edecektir.

Bu çalışmada kullanıcı için bir mobil uygulama sunuldu. Bu uygulama üzerinden istenilen sıcaklık değeri girilebilmektedir. Kullanıcı programı durdurana kadar ısıtıcı sıcaklık seviyesini, seçilen sıcaklık değerinde sabitlemek üzere çalışmayı değişen performansla sürdürmektedir.

Bu çalışma kapsamında yardımlarını esirgemeyen Prof. Dr. Oğuzhan Urhan Hocamıza teşekkürlerimi sunuyorum.

Aralık 2021, KOCAELİ

Burak Yeniaydın

İÇİNDEKİLER

ÖNSÖZ ve TEŞEKKÜR.....	i
İÇİNDEKİLER	ii
ŞEKİLLER DİZİNİ.....	ii
1. SICAKLIĞI AYARLANABİLİR SU ISITICISI	iv
2. ADJUSTABLE TEMPERATURE KETTLE	v
3. SICAKLIK DEĞERİNİN OKUNMASI.....	6
3.1 GİRİŞ.....	6
3.2 1-WIRE PROTOKOLÜ.....	6
3.3 DS18B20 SICAKLIK SENSÖRÜ HAFIZA YAPISI.....	7
3.4 DS18B20 SICAKLIK SENSÖRÜNÜN KULLANIMI	8
3.4.1 RESET ve PRESENCE DARBELERİ	8
3.4.2 ROM KOMUTLARI	9
3.4.3 İŞLEVSEL KOMUTLAR	10
3.4.4 OKUMA VE YAZMA ZAMANLARI	12
4. ISITICININ ÇALIŞMA PERFORMANSININ AYARLANMASI	13
5. PID İLE SICAKLIK KONTROLÜ	16
6. BASKI DEVRE VE KUTULAMA	18
7. ANDROİD UYGULAMA	22
SONUÇLAR ve ÖNERİLER.....	25
KAYNAKLAR	26
ÖZGEÇMİŞ	27

ŞEKİLLER DİZİNİ

Şekil 1 - DS18B20 su geçirmez sıcaklık sensörü	6
Şekil 2 - DS18B20 sensörü hafıza yapısı [1]	7
Şekil 3 - Sıcaklık Verisi Genel Yapısı	8
Şekil 4 - Başlatma İşlemi [1].....	9
Şekil 5 - ROM Kodu yapısı [1].....	9
Şekil 6 - Okuma ve Yazma Zamanları [1]	12
Şekil 8 - EL817 Optokuplör [2].....	13
Şekil 7 - Blok Diyagram	14
Şekil 9 - Triyak ile AC dalganın tetiklenmesi [3]	15
Şekil 10 - Referans değeri 60 derece	16
Şekil 11 - Değişkenlerin zamana göre değişimi.....	17
Şekil 12 - Sıcaklık 55 derece seviyesinde 70 derece seviyesine oturuyor	18
Şekil 13 - Şematik Çizimi	18
Şekil 14 - Şematik kütüphanem.....	19
Şekil 15 - PCB tasarımı.....	20
Şekil 16 - Baskı devre aşamaları - 1	20

Şekil 17 - Baskı devre aşamaları – 2.....	21
Şekil 18 - Kutulama	21
Şekil 19 - Son ürün	22
Şekil 20 - Uygulamanın ismi ve simgesi	23
Şekil 21 - Kullanıcı arayüzü	24

1. SICAKLIĐI AYARLANABİLİR SU ISITICISI

Burak Yeniaydın

Anahtar Kelimeler: Msp430, PID kontrol, Android uygulama.

Özet:

Bu projede, içerisindeki suyun kullanıcının ayarladığı değerde kalmasını sağlayabilen bir su ısıtıcısı tasarlanmıştır. Kullanıcıya bir arayüz sunulmuştur. Kullanıcı Android cihazından dilediği sıcaklık değerini girebilecektir. Su sıcaklığını istenilen seviyeye oturabilmesi için PID kontrol algoritması kullanılmıştır. Triyak, optokuplörler vb. elektronik elemanlardan oluşan donanım ile ısıtıcının çalışma performansı ayarlanmıştır. Tüm donanım bileşenleri bir araya getirilerek bir kutuda birleştirilmiştir.

2. ADJUSTABLE TEMPERATURE KETTLE

Burak Yeniaydin

Keywords: msp430, PID control, mobile application

Abstract: In this project, a kettle has been designed that can keep the temperature value set by the user. User can enter the temperature value from the Android device. PID control algorithm is used to set the temperature at the desired level. The operating performance of the heater is adjusted with the hardware consisting of electronic elements such as triac, optocouplers, etc. All hardware components are combined in box.

3. SICAKLIK DEĞERİNİN OKUNMASI

3.1 GİRİŞ

Bu projenin gerçekleştirilebilmesi için suyun sıcaklık değerini anlık olarak ölçülmesi gerekmektedir. Bu ölçüm su geçirmez sıcaklık sensörü olan DS18B20 ile gerçekleştirilmiştir.



Şekil 1 - DS18B20 su geçirmez sıcaklık sensörü

Bu sensör tek bir kablo üzerinden haberleşmektedir (one wire). Bir sonraki bölümde bu protokol detaylandırılmıştır.

3.2 1-WIRE PROTOKOLÜ

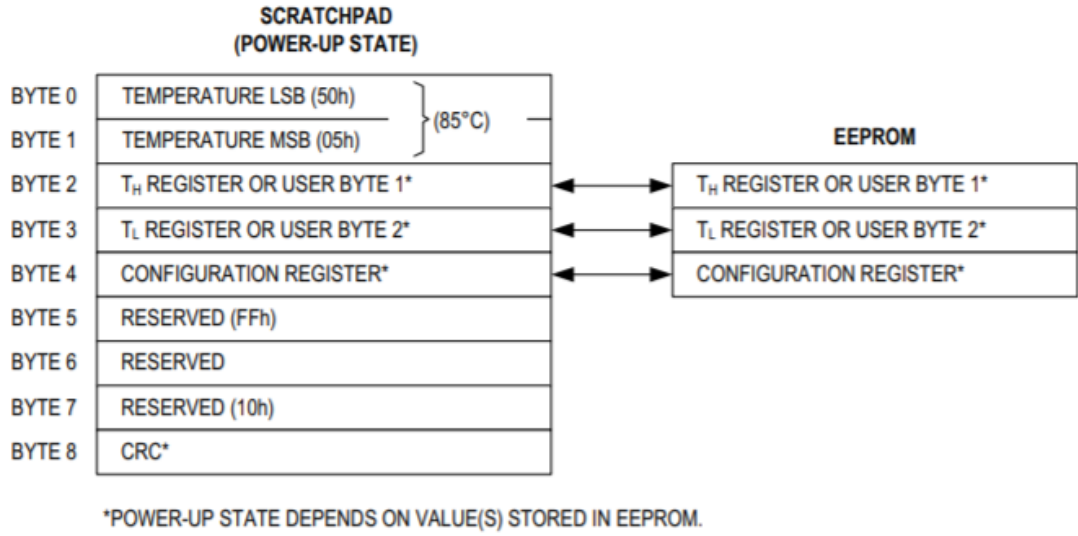
1-wire protokolü hem veri gönderme hem de veri alma işini tek bir kablo üzerinden gerçekleştirilerek kablo sayısından kazanç sağlamaktadır. Buna karşılık işlem yükü ve karmaşıklık artmaktadır.

Bu hatta bir master cihaz ve birden fazla slave cihaz bağlanabilmektedir. Ayrıca sensör harici bir besleme kaynağına ihtiyaç duyulmaksızın data hattından beslenebilmektedir. Buna “parasite power” denilmektedir. Data hattı logic 1 seviyesinde olduğu zaman devrede bu hatta bağlı olan kapasitör dolmaktadır. Hat logic 0 seviyesine düştüğünde sensöre güç bu kapasitör ile sağlanmaktadır. Proje kapsamında bir master cihaz ve ona bağlı olan tek bir slave cihaz kullanılmıştır. Besleme data hattından değil harici olarak sağlanmıştır. Data hattını kullanılmayan

durumlarda logic 1 seviyesine çekmek için bu hat bir direnç ile Vcc hattına bağlanmıştır.

3.3 DS18B20 SICAKLIK SENSÖRÜ HAFIZA YAPISI

DS18B20 sensörü bir hafıza birimine sahiptir. Bu hafıza biriminin yapısı Şekil 2’de gösterilmiştir.



Şekil 2 - DS18B20 sensörü hafıza yapısı [1]

Sıcaklık sensörü ortamın sıcaklık değerini ölçmek için dışarıdan bir komut beklemektedir. Bu komut geldikten sonra çevrimi başlatır. Çevrim tamamlandığında ise elde edilen sıcaklık değerinin hafızaya kaydedilmesi gerekir. Byte 0 ve byte 1 çevrim sonucu elde edilen bu sıcaklık değerini tutmaktadır. İkinci ve üçüncü byte alarm için belirlenen değeri tutmakla görevlidirler. Dördüncü byte ise konfigürasyon registerlarını içermektedir. DS18B20 sensörü, güç kesildiğinde de bu üç değer korunmak istenileceğinden dolayı bu bilgileri kalıcı olarak saklayacağı bir EEPROM’a sahiptir. Sıcaklık verisi ise EEPROM’da değil geçici hafızada tutulmaktadır. Sensöre verilen güç kesildikten sonra sıcaklık değeri otomatik olarak 85 dereceye ayarlanır.

Hafızanın ilk iki byte’ında depolanan sıcaklık verisinin genel yapısı Şekil 3’te gösterilmiştir. Sıfırıncı byte sıcaklık verisinin ondalık kısmını ve tam kısmın bir

bölümünü içerirken birinci byte sıcaklık değerinin tam kısmının bir bölümünü ve işaret bitlerini içermektedir. İşaret bitleri ‘0’ ise sıcaklık verisi pozitif bir değer demektir. Bu bitlerin ‘1’ olması durumunda ise negatif sıcaklık değeri ölçülmüş demektir. İşaret bitlerinden sonra sıcaklık verisinin tam kısmı ve ondalık kısmı gelmektedir. Ondalık kısım için dört bit ayrılarak yüksek hassasiyet sağlanmıştır. Bu 1 derecenin 16 farklı değere bölünmesi anlamına gelmektedir. Yani her adım 0.0625 derecelik hassasiyete denk gelmektedir. Bu bilgiler ışığında Şekil 3’te gösterilen sıcaklık bilgisinin 25.0625 derece olduğu söylenebilir.

MSB ←		→ LSB
İşaret Bitleri – 5 bit	Sıcaklık Değerinin Tam Kısım – 7 bit	Ondalık Kısım – 4 bit
0 0 0 0 0	0 0 1 1 0 0 1	0 0 0 1

Şekil 3 - Sıcaklık Verisi Genel Yapısı

3.4 DS18B20 SICAKLIK SENSÖRÜNÜN KULLANIMI

Bu projede sıcaklık sensörü ortamın sıcaklık verisini elde etmek için kullanılmıştır. Bu değeri elde etmek için protokole uygun olarak işlemler sırasıyla gerçekleştirilmelidir. Master cihaz sıcaklık verisini elde etmek için aşağıdaki üç işlemi sırasıyla izlemek durumundadır.

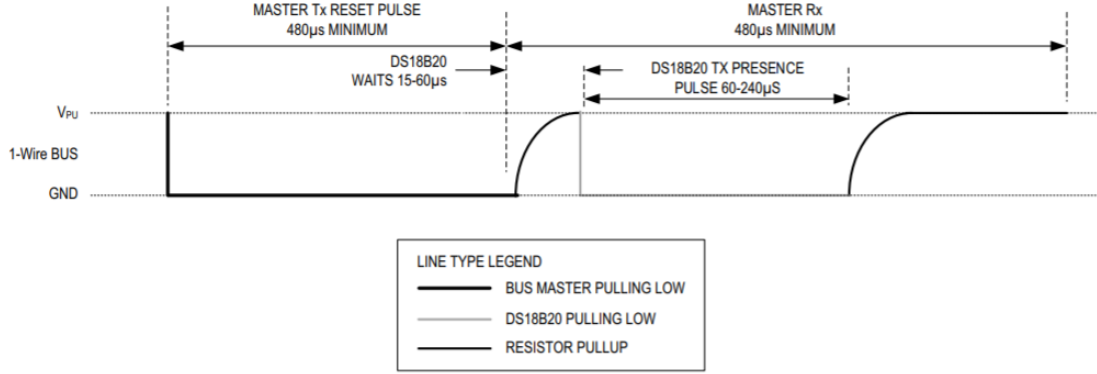
- Reset darbesi göndermek ve ardından sensör tarafından gönderilen “presence” darbesini beklemek.
- ROM komutu göndermek.
- İşlevsel komutu göndermek.

Bu işlemler sonraki başlıklarda adım adım anlatılmıştır.

3.4.1 RESET ve PRESENCE DARBELERİ

Sensöre veri göndermek veya sensörden veri okumak için ilk olarak master cihaz reset darbesi göndermelidir. One-Wire hattı boşta iken pull-up direnci sayesinde hat logic 1 seviyesinde beklemektedir. Reset darbesi hattın, yaklaşık 500 mikro saniye kadar logic 0’a çekilmesi ile gönderilir. Bu darbeyi yakalayan slave cihaz bu darbeye

karşılık olarak hattı logic 0'a çekmek suretiyle yaklaşık 220 mikro saniye kadar bir “presence” darbesi gönderir. Bu işlem Şekil 4’te gösterilmiştir.

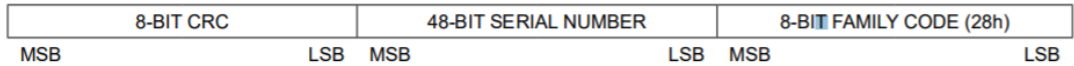


Şekil 4 - Başlatma İşlemi [1]

Master cihaz “presence” darbesini aldıktan sonra artık ROM komutlarını gönderebilir hale gelmektedir. Bir sonraki başlıkta ROM komutları incelenmiştir.

3.4.2 ROM KOMUTLARI

İşlevsel komutların gönderimine geçebilmek için ROM komutlarından en az bir tanesinin gönderilmesi zorunludur. Hatta birden fazla slave cihaz olması durumunda master cihazın hangi slave cihaz ile haberleşeceği bu aşamada seçilir. Her bir slave cihaz eşsiz bir seri numarasına sahiptir. Bu seri numaraları 64 bitlik ROM Kodunun bir bölümünde tutulmaktadır. ROM Kodunun yapısı Şekil 5’te gösterilmiştir.



Şekil 5 - ROM Kodu yapısı [1]

Toplamda 5 adet ROM komutu mevcuttur. Bu komutlar ve görevleri Tablo 1’de gösterilmiştir. Bu projede hatta sadece tek bir slave cihaz bulunması ve işlemlerin hızlandırılması açısından herhangi bir ROM kodu sorgulamasına gidilmemiş olup “ROM komutlarını atla” komutu gönderilmiştir. Bu komut hattaki tüm slave

cihazlara aynı komutları gönderir. Bu projede tek bir slave cihaz olduğundan okuma komutundan sonra herhangi bir örtüşme gözlenmez.

Tablo 1 - ROM Komutları

ROM KOMUTU	KOMUT TANIMI
0xF0	Hatta bağlı olan tüm slave cihazların ROM kodlarını öğrenir.
0x33	Eğer hatta sadece bir slave cihaz varsa ROM kodunu öğrenmek için kullanılır.
0x55	Gönderilen ROM kodu ile aynı koda sahip olan slave cihazı eşleştirir. Sadece eşleşen bu slave cihaz işlem komutuna yanıt verecektir. Diğer cihazlar reset darbesi bekleyecektir.
0xCC	Bir ROM kodu göndermeksizin tüm slave cihazlara komut göndermek için kullanılır. Örneğin tüm slave cihazlar için sıcaklık çevrimini başlat komutu verilebilir. Ancak bir okuma komutu gönderilirse gelen verilerde örtüşme oluşur.
0xEC	En son gerçekleşen sıcaklık çevrimi sırasında herhangi bir alarm durumunun yaşanıp yaşanmadığını anlamak için kullanılır.

ROM komutunun gönderilmesinden sonra master işlevsel komutları gönderme aşamasına geçebilir. İşlevsel komutlar bir sonraki bölümde detaylandırılmıştır.

3.4.3 İŞLEVSEL KOMUTLAR

Toplam altı tane işlevsel komut mevcuttur. Bunların bir tanesi sıcaklık ölçmek için çevrimi başlatma komutu iken diğer beş komut hafızaya veri yazma ve hafızadan veri okumakla ilişkilidir. Bu komutlar ve işlevleri Tablo 2’de gösterilmiştir.

Tablo 2 - İşlevsel Komutlar

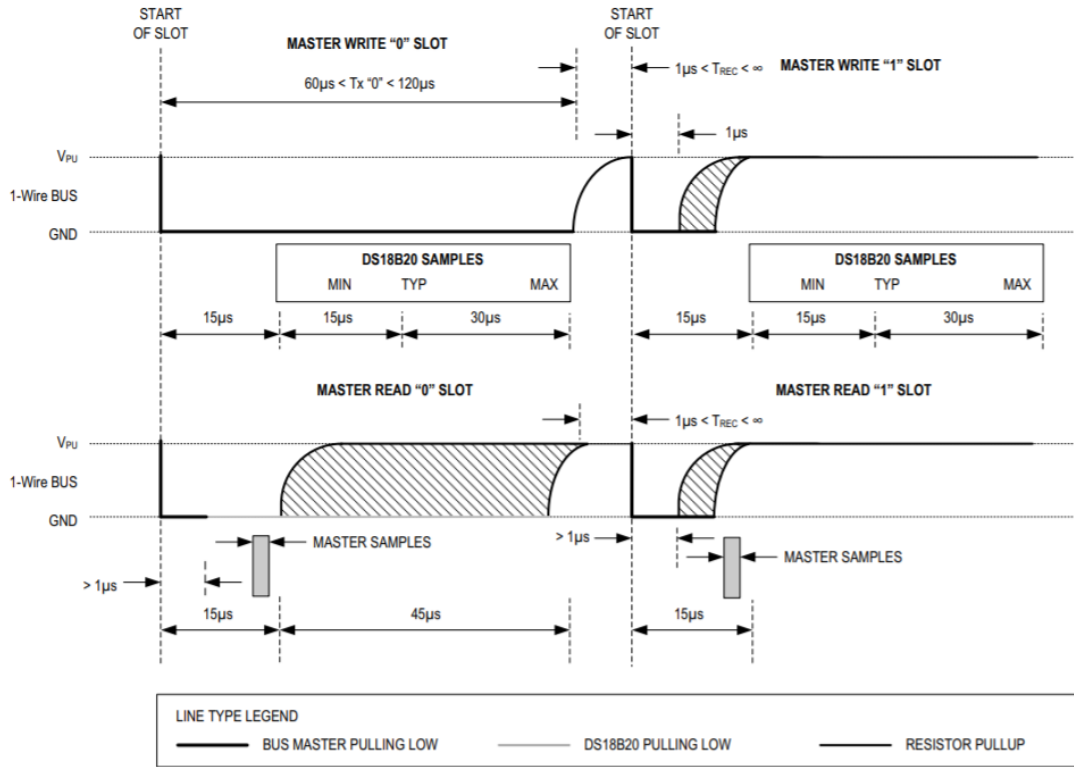
İŞLEVSEL KOMUT	KOMUT TANIMI
0x44	Sıcaklık ölçmek için çevrimi başlatır. Elde edilen sıcaklık verisi sensörün hafızasına yazılır.
0x4E	Hafızaya veri yazmak için kullanılır. Sadece alarm ve konfigürasyonlar için ayrılmış hafıza alanına yazma işlemi yapılabilir.
0xBE	Sensörün hafızasından veri okumak için kullanılır. Okuma işlemi sıfırıncı byte'ın en değersiz bitinden başlayarak başlar. Sekizinci nyte'ın sonuna kadar devam eder. Master ihtiyacı kadarını okuduktan sonra reset darbesi göndererek okuma işlemini sonlandırabilir.
0x48	Geçici hafıza üzerinde yazılı olan alarm ve konfigürasyon bilgilerini EEPROM'a aktarma işlemi yapar.
0xB8	EEPROM'da kalıcı olarak kayıtlı olan alarm ve konfigürasyon bilgilerini geçici hafızaya aktarır.
0xB4	Slave cihazın "parasite power" ile beslenip beslenmediğini öğrenir.

Bu projede işlevsel komutlardan iki tanesini kullanılmıştır. İlk olarak 3.4'te anlatılan adımlardan ilk ikisi sırayla gerçekleştirilerek son adımda 0x44 komutu gönderilir. Bu komut slave cihazın çevrimi başlatmasını sağlayacaktır. Çevrimin gerçekleşmesi ve sıcaklık bilgisinin hafızaya yazılması için belirli bir süre beklenmelidir. Ardından bölüm 3.4'te anlatılan ilk iki adım gerçekleştirip son adımda bu kez işlevsel komutlardan 0xBE komutu gönderilir. Bu komut ile sensörün hafızasında kayıtlı olan 16 bitlik sıcaklık verisi okunmuştur.

Buraya kadar olan bölümde slave cihaza komutlar gönderilmesi, slave cihazın hafızasından bilgi okunması ve slave cihazın hafızasına veri yazılması anlatılmıştır. logic 0 ya da logic 1 bitinin hatta nasıl yazıldığı veya hattan nasıl okunduğu bir sonraki bölümde anlatılmıştır.

3.4.4 OKUMA VE YAZMA ZAMANLARI

Hatta veri yazarken sıcaklık sensörünün örnekleme zaman aralığı esas alınmalıdır. Hattan gelen veri okunurken ise master cihaz doğru bir örnekleme aralığı bulmalıdır. Şekil 6’da yazma ve okuma süreleri tüm ayrıntılarıyla gösterilmiştir.



Şekil 6 - Okuma ve Yazma Zamanları [1]

Yazma veya okuma fark etmeksizin tüm durumlarda başlangıçta hat master cihaz tarafından logic 0 değerine düşürülmelidir. Hatta veri yazma durumunda gönderilecek bitin değeri, master cihazın davranışını değiştirecektir. Eğer logic 1 değeri yazılacaksa yaklaşık 6 mikro saniye beklenip hat logic 1 seviyesine çıkarılmalıdır. Eğer logic 0 değeri yazılacaksa hat yaklaşık 60 mikro saniye logic 0 değerinde bekletilmelidir. Hattan veri okuma sırasında hat logic 0 değerine

düşürüldükten yaklaşık 6 mikro saniye sonra portun yönü değiştirilmelidir. Toplamda 13-15 mikro saniye aralığında hattaki değer master cihaz tarafından örneklenmelidir. Sadece bu küçük aralıktaki değer gelen verinin ne olduğunu belirtir. Aksi durumda gelen veri yanlış algılanabilmektedir.

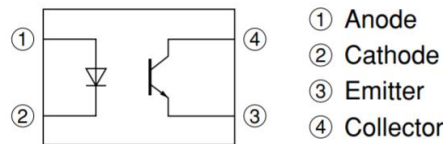
Bir sonraki bölümde ısıtıcı performansının ayarlanması için kurulan donanım anlatılacaktır.

4. ISITICININ ÇALIŞMA PERFORMANSININ AYARLANMASI

Projede kullanılan su ısıtıcısı açma kapama mantığıyla çalışmakta ve açık kaldığı süreç boyunca tam performansta çalışmaktadır. Akıllı ve daha verimli bir su ısıtıcısı sürekli olarak tam performansta çalışmak yerine değişen performansla çalışmalıdır.

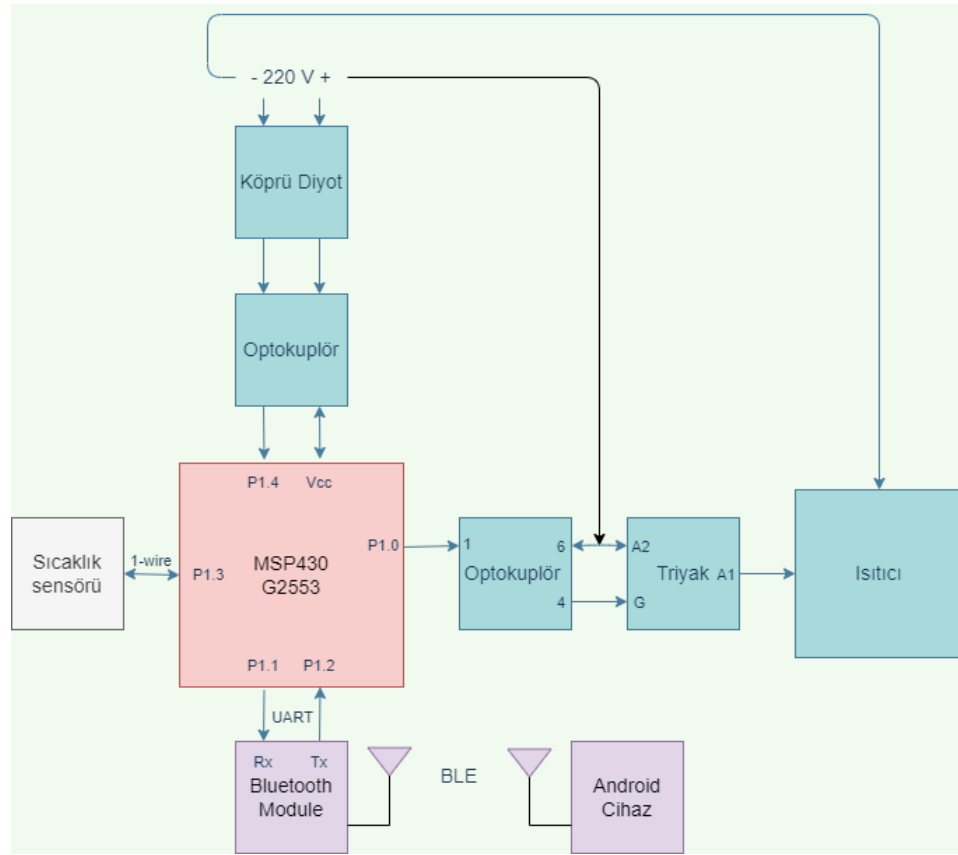
Kullandığım ısıtıcı 220 V AC gerilim ile çalışmaktaydı. Şekil 8’de tüm sistemin blok diyagramı gösterilmiştir. Performans ayarı için kullanılan devre ise mavi kutular ile gösterilmiştir.

Performans ayarı için kullanılan donanımda köprü diyot 220 V’luk AC sinyali doğrultulmuştur. Optokuplör kullanılarak 220 V gerilim ile mikrodenetleyici birbirinden izole edilmiştir. Ayrıca optokuplörün girişine köprü diyotun çıkışındaki tam doğrultulmuş sinyal verilmektedir. Optokuplör bu sinyaldeki sıfır geçişleri yakalamak için kullanılmıştır. Optokuplörün bu işi nasıl yaptığını anlamak için genel yapısını incelemek gerekmektedir. Bu genel yapı Şekil 7’de gösterilmiştir.



Şekil 7 - EL817 Optokuplör [2]

Optokuplör Şekil 7’de de gözüktüğü üzere, içerisinde bir adet kırmızı ışın yayan led ve bir adet bu ışığı algılayınca ilettime geçen fototransistörden oluşmaktadır. Bu yapıyla birlikte optokuplörün girişindeki sinyal diyotun eşik geriliminden aşağıda olursa fototransistör kesime gider. Böylelikle 3 numaralı bacak logic 1 değerinden logic 0 değerine düşer. Sıfır geçişleri yakalamak için her düşen kenar algılanmalıdır. Bunun için mikrodenetleyicinin P1.4 pini düşen kenarda kesme alacak şekilde ayarlanmıştır. Şehir şebekesinin frekansının 50 Hz olduğundan yola çıkılırsa bu sinyalin her yarım periyodunun 10 milisaniye olduğu anlaşılır. Yani her 10 milisaniyede bir düşen kenar kesmesi oluşmaktadır.



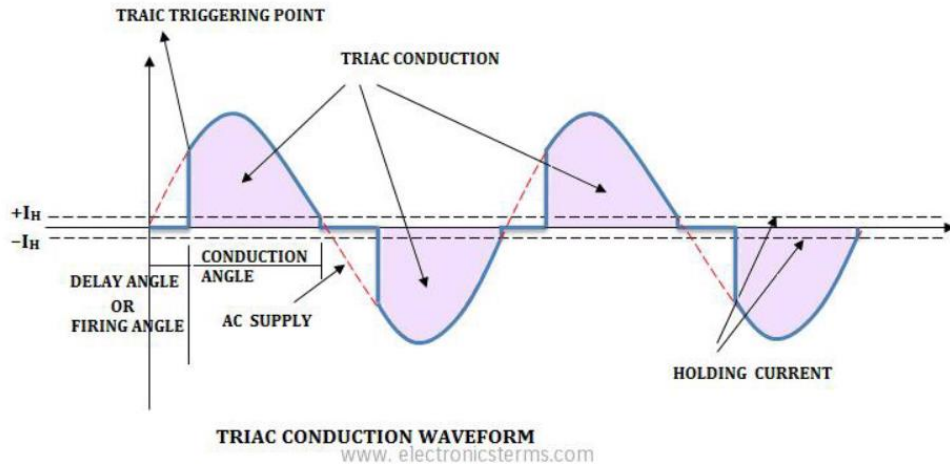
Şekil 8 - Blok Diyagram

Isıtıcının performansını belirleyen faktör düşen kenar algılandıktan sonra triyaka gönderilecek olan tetik sinyalinin ne kadar sürede gönderildiğidir. Burada Timer kullanılmıştır. Düşen kenar algılandıktan hemen sonra Timer başlatılmıştır.

TACCRO'ya yazılan değere göre ısıtıcının performansı değişmektedir. Her 10 milisaniyede AC dalganın yarım periyodu tamamlanacağından ötürü TACCRO'ya yazılan değer Timer sayma işleminin 10 milisaniyeyi geçmeyecek şekilde ayarlanması gerekmektedir.

Timer sayma işlemini bitirdikten sonra mikrodnetleyicinin P1.0 pininden optokuplöre tetik sinyali gönderir. Bu optokuplör ise mikrodnetleyiciden gelen tetik sinyaliyle 220 V ile çalışan triyakı tetiklemek için kullanılmıştır.

Triyak tetik sinyali geldiği andan itibaren A2 ucundaki 220V değerindeki sinyali çıkışına verir. Sıfır geçişe geldiğinde yeniden tetik sinyali gönderilmesi gerekir. Çünkü AC gerilimin sıfır seviyesinde olduğu yerde triyakın içinde bulunan iki diyotta kesimde olmuş olur.

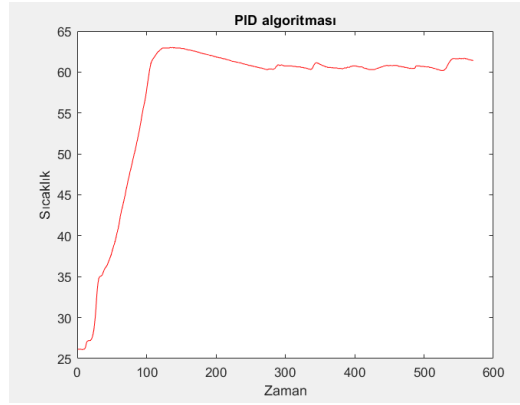


Şekil 9 - Triyak ile AC dalganın tetiklenmesi [3]

Kullanılan bu donanım sayesinde ısıtıcı değişen performansla çalışabilir hale gelmiştir. PID kontrol algoritması ile bu çalışma performansı ayarlanarak sıcaklık kontrol edilebilir. Önümüzdeki başlıkta PID algoritmasının bu devreye uygulanması anlatılacaktır.

5. PID İLE SICAKLIK KONTROLÜ

PID bir geri beslemeli kontrol yapısıdır. Sürekli olarak bir önceki çevrim ile aradaki hata değerlerini hesaplayarak, hatayı en aza indirmeyi amaçlar. Bir önceki başlıkta TACCR0'a yazılan değere göre ısıtıcının performansının değiştiği söylenmişti. PID kontrol algoritmasının sonucunda oluşan çıktı değeri ise map edilerek TACCR0'a aktarılarak PID kontrol algoritması sisteme dahil edilmiştir.



Şekil 10 - Referans değeri 60 derece

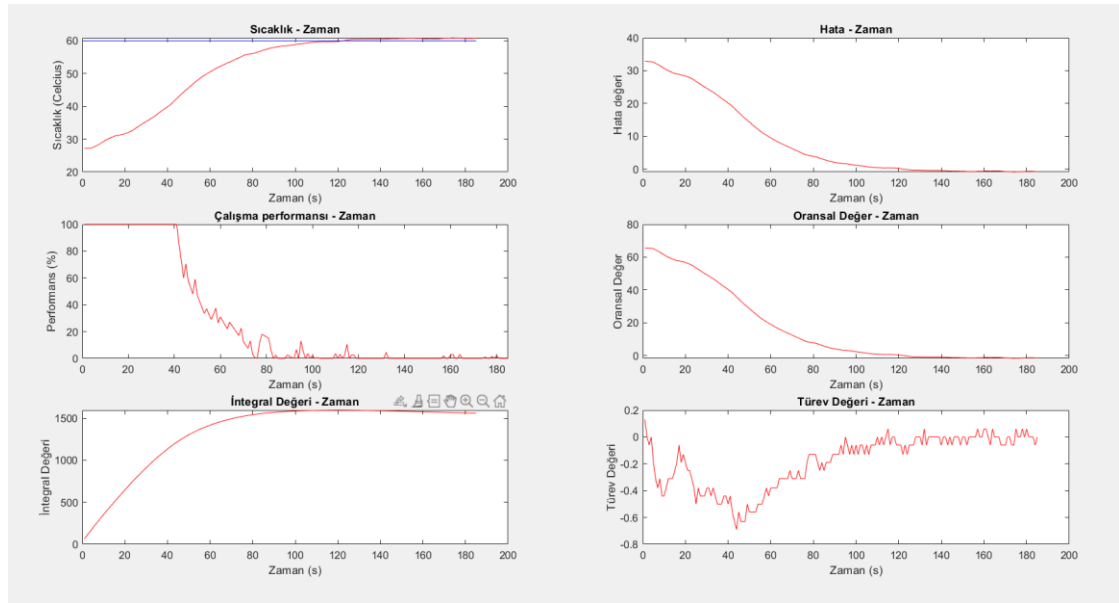
Şekil 10'da 60 derece olarak belirlenen referans değeri civarında ölçülen sıcaklık değerinin salınımı gözükmemektedir

Oransal (Proportional) kontrol yapısında elde edilen hata değerini belirli bir katsayı ile çarpılır. Bu projede oransal katsayı (K_p değeri) 2 olarak belirlenmiştir. Hata değeri ne kadar büyükse oransal kontrolün performansa katkısı o kadar fazla olacaktır. Referans değeri aşılsa hata değeri negatif çıkacağı için oransal kontrol bu kez ters yönde bir etki gösterecektir.

İntegral geçmişte yapılan hatalar için bir hafıza gibidir. Geçmiş hata değerleri ile anlık hata değerini sistem kapanana kadar toplar. İlgili değer referans değerini geçse bile integral bir süre daha aynı yönde etki yapmaya devam eder. Isıtıcı hızlı ısınıp

yavaş soğuyan bir elemandır. Bundan dolayı aşım değerinin yüksek olması bu proje için hiç istenmeyen bir durumdur. Bu durumda integral katsayısı küçük seçilerek etkisi çok azaltılmıştır. İntegral katsayısı (K_i) 0.0001 olarak seçildi.

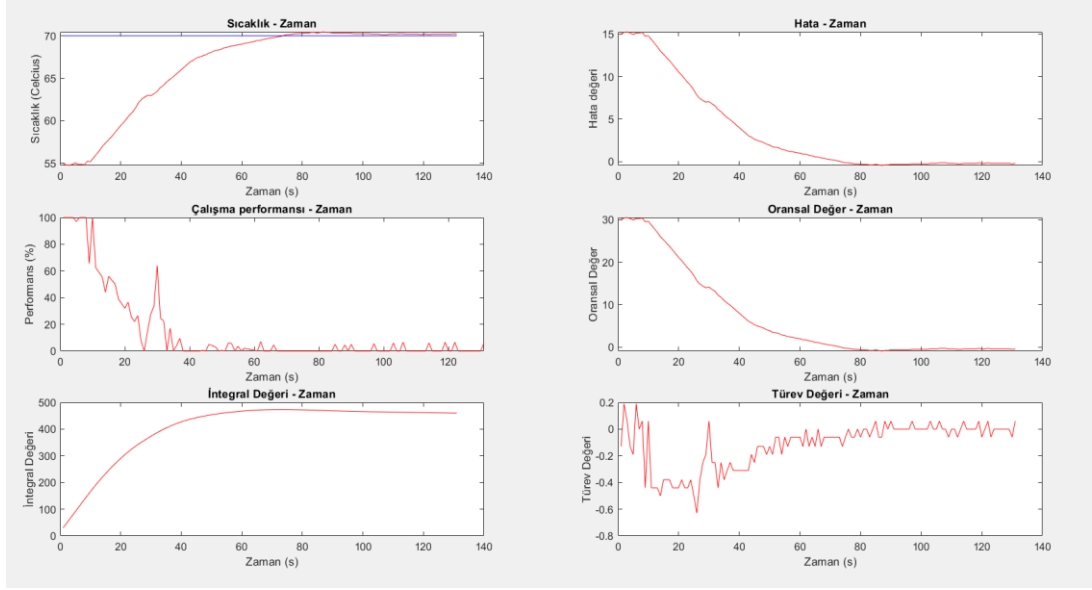
Türev ise genelde frenleyici bir etkiye sahiptir. Anlık hata değeri ile bir önceki hata değerinin arasındaki farkı bir katsayıyla çarpar. Eğer hata azalma eğilimindeyse performansı düşürücü yönde etki yapar. Eğer hata artıyorsa bu kez performansı düşürücü etki yapar. Bu projede türev katsayısı (K_d) 30 olarak seçildi.



Şekil 11 - Değişkenlerin zamana göre değişimi

Şekil 11'de sıcaklığın ve hatanın zamana göre değişimleri gösterilmiştir. Sıcaklık-zaman grafiği üzerindeki mavi çizgi referans değerini göstermektedir. Sıcaklığın zaman içinde referans değeri civarına oturduğu gözlemlenmiştir. 3. görselde ısıtıcının çalışma performansının zamana göre değişimi gösterilmiştir. Diğer 3 görselde ise oransal değer, türev ve integral değerleri gösterilmiştir. Oransal değer hata değerinin 2 katı kadar çıkışa etki yapmıştır.

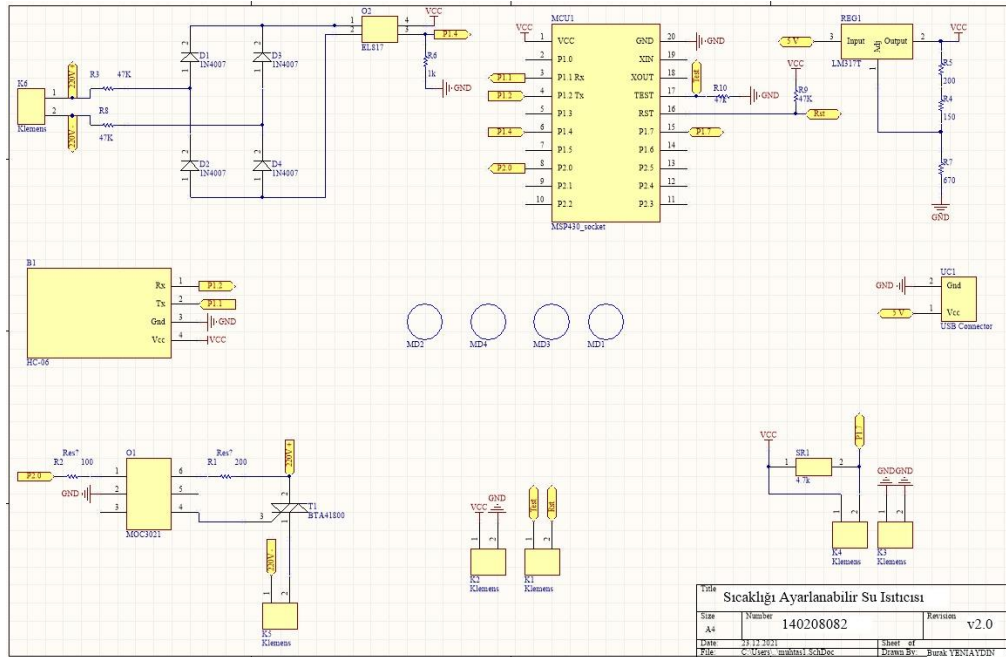
Şekil 12'de de benzer bir şekilde sıcaklığın istenen seviyeye yakın bir değere oturduğunu gözlemliyoruz.



Şekil 12 - Sıcaklık 55 derece seviyesinde 70 derece seviyesine oturuyor

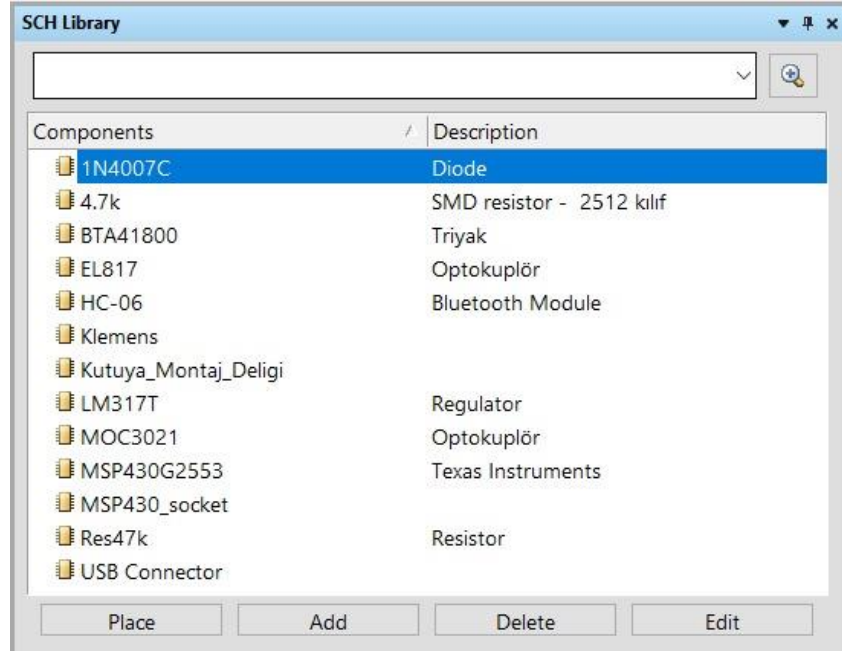
6. BASKI DEVRE VE KUTULAMA

Çizimler “Altium” programı kullanılarak oluşturulmuştur.



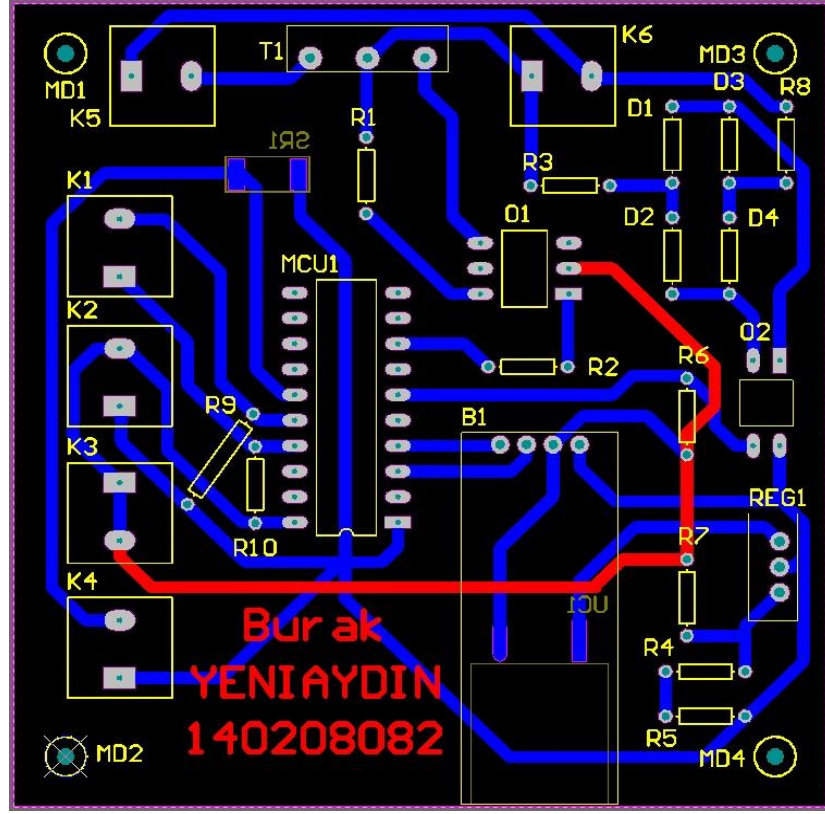
Şekil 13 - Şematik Çizimi

Şekil 13'te devrenin şematik çizimi gösterilmiştir. Baskı devrede dışarı ile bağlantıyı sağlayacak 6 tane klemens kullanılmıştır. Mikrodenetleyiciyi dışarıdan programlayabilmek için TEST ve RST pinleri de klemensler aracılığıyla dışarı çıkarılmıştır. Baskı devreyi kutuda sabitleyebilmek için M3 vida delikleri çizime dahil edilmiştir. Isıtıcı 220 V ile K6 klemensinden dolayı olarak beslenmiştir.



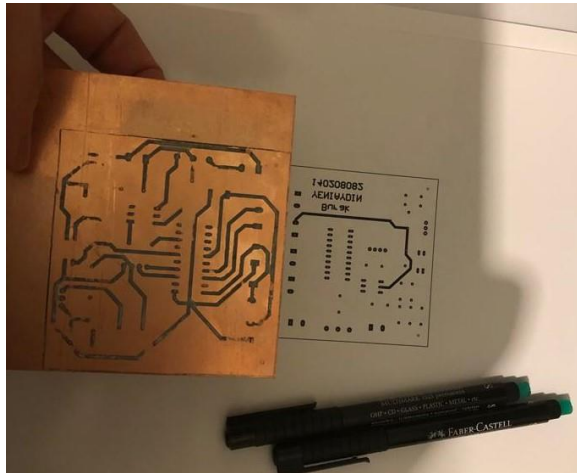
Şekil 14 – Şematik kütüphanem

Şematik çiziminde ve PCB tasarımında tüm devre elemanlarını tarafımca çizilmiştir. Şekil 15'te iki katlı PCB tasarımı yer almaktadır. Alt katmanda yer alan SR1 direnci yüzey montalı devre elamanı olarak tasarıma eklenmiştir. Alt katmanda bir SMD direncin dışında USB soketi bulunmaktadır. Devreye 5V buradan verilmektedir. Ardından LM317T regülatörü kullanılarak 5V, 3.6V seviyesine düşürülmüştür. Tasarımdaki bluetooth modülü, sıcaklık sensörü ve mikrodenetleyici 3.6 V ile tek bir regülatörden beslenmektedir.

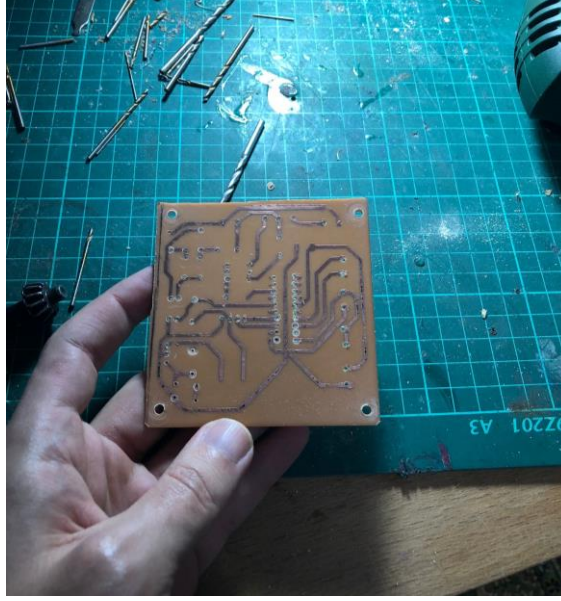


Şekil 15 - PCB tasarımı

Şekil 16 ve Şekil 17’de baskı devre aşamaları gösterilmiştir.

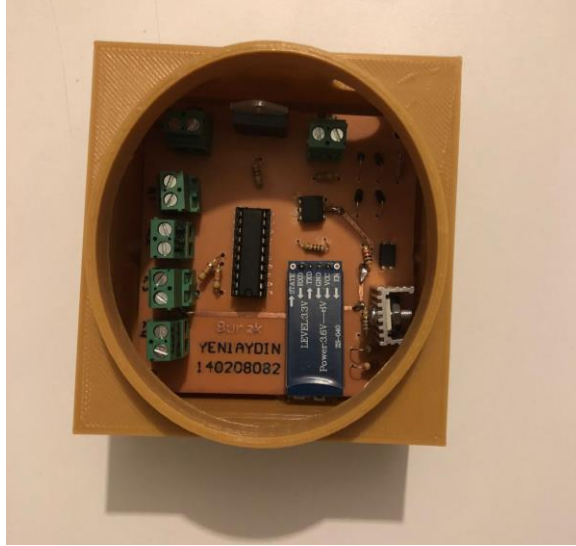


Şekil 16 - Baskı devre aşamaları - 1



Şekil 17 - Baskı devre aşamaları – 2

Şekil 18’de gösterilen kutu 3 boyutlu yazıcı ile özel olarak basılmıştır. Temas ettiği yüzeylerin sıcaklık seviyesi yüksek olduğundan kutu için ısıya dayanıklı olan ABS filament malzemesi kullanılmıştır.



Şekil 18 - Kutulama



Şekil 19 - Son ürün

Şekil 19’da tasarımın son hali gösterilmektedir.

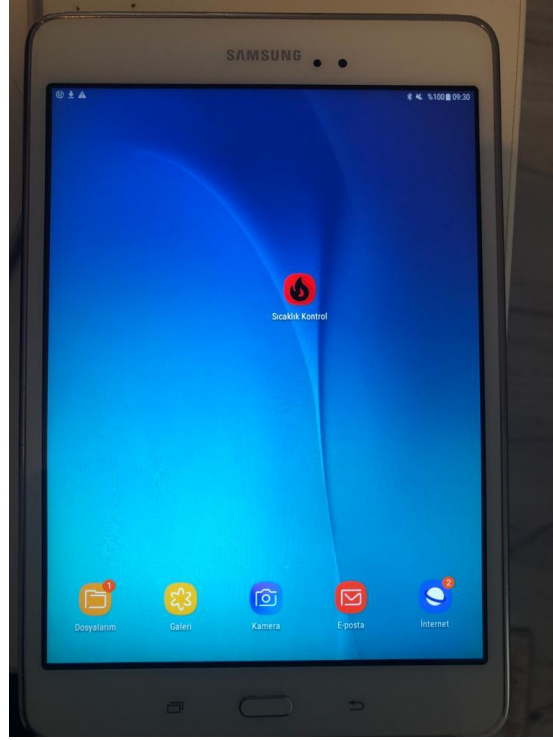
7. ANDROİD UYGULAMA

Sıcaklık değerini dışarıdan ayarlayabilmek için kullanıcıya bir arayüz sunulmuştur. Kullanıcı sıcaklık değerini bu arayüzden girerek cihazı başlatabilir. Dilediği zaman ısıtma işlemini kapatabilir. Ayrıca kullanıcı anlık sıcaklık değerine arayüzde görebilmektedir.

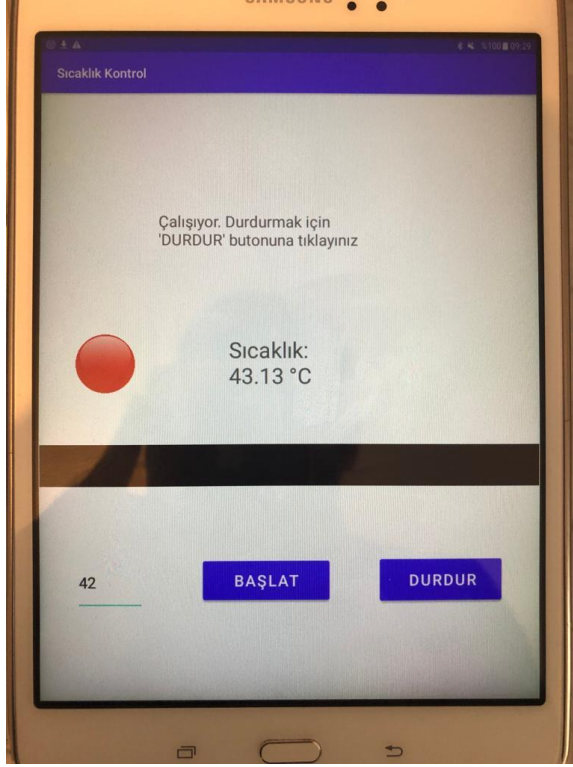
Android uygulama “Android Studio” ortamında “Java” programlama dili kullanılarak oluşturulmuştur. İlk olarak bazı sınıflar projeye dahil edilmiştir. BluetoothAdapter sınıfı mevcut cihazdaki Bluetooth’u gösterir. Dışarıdaki cihazları tarama, gelen istekleri dinleme gibi işlemleri yapan birçok metodu içerisinde barındırır. BluetoothDevice sınıfı ise uzaktaki diğer Bluetooth cihazlarını gösterir. Bu sınıf sayesinde ilgili cihaz ile bağlantı oluşturulabilir. Ayrıca cihazla ilgili adres,

isim gibi birçok bilgiye ulaşabiliriz. İki cihaz arası bağlantının sağlamak ve veri alışverişi yapmak için BluetoothSocket kullanılır.

Uygulama arayüzünde Button, EditText, TextView, ImageView, ToolBar



Şekil 20 - Uygulamanın ismi ve simgesi



Şekil 21 - Kullanıcı arayüzü

SONUÇLAR VE ÖNERİLER

Son zamanlarda yaşanan gelişmelere birlikte kullanıcı kontrollü akıllı ev cihazlarının kullanımı yaygınlaştı. Su ısıtıcıları da bu cihazlardan biridir.

Su ısıtıcıları genellikle belirli bir sıcaklık değerine kadar çalışan, o değerden sonra çalışmayı durduran bir mekanizmaya sahiptir. Bu projede ise suyun sıcaklık değeri kullanıcı tarafından ayarlanabilmektedir. Su, sistem kapatılana kadar aynı sıcaklık seviyesinde kalmaya devam edecektir.

Bu çalışmada kullanıcı için bir mobil uygulama sunuldu. Bu uygulama üzerinden istenilen sıcaklık değeri girilebilmektedir. Kullanıcı programı durdurana kadar ısıtıcı sıcaklık seviyesini, seçilen sıcaklık değerinde sabitlemek üzere çalışmayı değişen performansla sürdürmektedir.

KAYNAKLAR

- [1] <https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf>
- [2] <https://www.tme.eu/Document/371e110151e1c8afec5198356913d01a/EL817.pdf>
- [3] <https://appliantology.org/gallery/image/1292-triac-output-and-gating-waveform/>

ÖZGEÇMİŞ