MİKRODENETLEYİCİ KONTROLLÜ BİR ISIL SİSTEM TASARIMI VE GERÇEKLEŞTİRİLMESİ

Murat ALTIPARMAK

YÜKSEK LİSANS TEZİ

ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ

GAZİ ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

KASIM 2005

ANKARA

Murat ALTIPARMAK tarafından hazırlanan MİKRODENETLEYİCİ KONTROLLÜ BİR ISIL SİSTEM TASARIMI VE GERÇEKLEŞTİRİLMESİ adlı bu tezin Yüksek Lisans tezi olarak uygun olduğunu onaylarım.

Prof. Dr. Veysel SİLİNDİR
Tez Yöneticisi

Bu çalışma, jürimiz tarafından Elektrik Elektronik Mühendisliği Anabilim Dalında Yüksek Lisans tezi olarak kabul edilmiştir.

Başkan: : Doç. Dr. İrfan KARAGÖZ

Üye : Prof. Dr. Veysel SİLİNDİR

Üye : Yrd. Doç. Dr. Fadıl ÇELİKKOL

Üye :

Üye :

Bu tez, Gazi Üniversitesi Fen Bilimleri Enstitüsü tez yazım kurallarına uygundur.

MİKRODENETLEYİCİ KONTROLLÜ BİR ISIL SİSTEM TASARIMI VE GERÇEKLEŞTİRİLMESİ

(Yüksek Lisans Tezi)

Murat ALTIPARMAK

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Kasım 2005

ÖZET

Bu çalışmada; On/Off, Oransal (P), Oransal+Türevsel (PD), Oransal+İntegral (PI) ve Oransal+İntegral+Türevsel (PID) denetleyici algoritmalarının, gömülü bir mikrodenetleyici yazılımı aracılığıyla, tasarlanan model bir ısıl sistem üzerinde uygulaması gerçekleştirilmiştir. Sistem yazılımı; kontrol sisteminin sıcaklık kontrolünü yapmasının yanı sıra, bir "süreç-kontrol" deney seti olarak kullanılabilmesini, çeşitli kontrol türlerinin model üzerindeki etkilerinin gözlenebilmesini sağlayacak şekilde düzenlenmiştir. Kontrol sisteminde; mikro-denetleyici ile dış dünya arasındaki iletişim, sayısal büyüklüklerin analoğa dönüşümleri ayrıca elde edilen verilerin işlenerek istenen çıktıların elde edilebilmesi ve mikrodenetleyicinin, tasarımlanarak çevre birimleriyle bir bütün olarak "model bir sistem" üzerinde uygulanması amaç edinilmiştir.

Bilim Kodu : **905.1.084**
Anahtar Kelimeler : **Gömülü Yazılım, Sıcaklık Kontrolü, Mikrodenetleyici, On/Off, Oransal (P), Oransal + Türevsel (PD), Oransal + İntegral (PI) ve Oransal + İntegral + Türevsel (PID) Denetleyiciler.**
Sayfa Adedi : **152**
Tez Yöneticisi : **Prof. Dr. Veysel SİLİNDİR**

DESIGNING AND IMPLEMENTING A MICROCONTROLLER BASED HEATING CONTROL SYSTEM

(M.Sc. Thesis)

Murat ALTIPARMAK

GAZI UNIVERSITY

INSTITUTE OF SCIENCE AND TECHNOLOGY

November 2005

ABSTRACT

In this study; On/Off, Proportional (P), Proportional+Derivative (PD), Proportional+Integral (PI) and Proportional+Integral+Derivative (PID) controller algorithms have been implemented based on an embedded microcontroller software and applied to a model heating system that was designed. System software has been performed in a way that not only to control the temperature of the system but also to use it as a "process-control" experimental kit and for observing the effects of various control types on the model. The communication between the microcontroller and the outside world, conversion of the digital quantities to their analog counterparts, processing the obtained data in order to acquire the desired output, designing the microcontroller and applying it to "a model system" with suitable peripherals as a whole one are the aims of the control system in this thesis.

Science Code : 905.1.084
Key Words : Embedded Software, Temperature Control, Microcontroller, On/Off, Proportional (P), Proportional + Derivative (PD), Proportional + Integral (PI) and Proportional + Integral + Derivative (PID) Controllers.
Page Number: 152
Adviser : Prof. Dr. Veysel SİLİNDİR

TEŞEKKÜR

Çalışmalarım boyunca değerli yardım ve katkılarıyla beni yönlendiren, değerli hocam Sayın Prof. Dr. Veysel SİLİNDİR'e; pratik sistem tasarımımda Elektronik Devreler Laboratuvarını kullanımıma açan hocam Sayın Yrd. Doç. Dr. M. Timur AYDEMİR'e, odalarını ve bilgisayar donanımını sürekli kullandığım GÜMMF Elektrik-Elektronik Müh. Böl. Araştırma Görevlileri Muhammet ÜNAL ve Ahmet Devrim ERDOĞAN'a, sıcaklık kontrol ve ısıl sistem mimarisi tasarımımda yardımcı olan GÜMMF Makine Müh. Böl. Araştırma Görevlisi Tuncay KARAÇAY'a, çalışmalarımda sıklıkla görüş alışverişinde bulunduğum Gazi Üniv. Fen Bilimleri Enstitüsü Elektrik-Elektronik Müh. Böl. Yüksek Lisans Öğrencileri İsa GÖK, Mustafa Ergin ŞAHİN, Murat SALTAN ve Mehmet TÜREN'e, PIC mikrodenetleyici yazılımını gerçekleştirmemde yardımcı olan GÜMMF Elektrik-Elektronik Müh. Böl. Lisans Öğrencisi Bahadır BÜLBÜL'e, Türk Telekom AŞ İnternet Veri Merkezi (IDC) Müdürü Dirsehan TUNÇEL ve TTNet E-posta sunucu grubu çalışanlarına, TAI A.Ş. çalışanı Yük. Elektrik-Elektronik Müh. Levent YILDIRIM'a, TEDAŞ çalışanları Elektrik-Elektronik Müh. Harun ULUCANLAR ve Uğur Can YOLDAŞ'a, zor anlarımın yardımcısı Yük. Elektrik-Elektronik Müh. Serdar FİRENGİZ'e ve beni her zaman destekleyen aileme yardımlarından ötürü teşekkürü bir borç bilirim.

İÇİNDEKİLER

Sayfa

Sayfa

Sayfa

ŞEKİLLERİN LİSTESİ

Şekil **Sayfa**

Şekil **Sayfa**

Şekil **Sayfa**

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış bazı simgeler ve kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Simgeler	**Açıklama**
2δ	Diferansiyel Aralık

Kısaltmalar	**Açıklama**
A/D	Analog/ Sayısal
D/A	Sayısal/Analog
INT	Kesme
LCD	Sıvı Kristal Ekran
LED	Işık Yayan Diyot
P	Oransal
PD	Oransal+Türevsel
PI	Oransal+İntegral
PIC	Çevresel Arabirim Denetleyicisi (PIC 16F877 Mikrodenetleyici)
PID	Oransal+İntegral+Türevsel
PWM	Darbe Genişlikli Modülasyon
RISC	Azaltılmış Komut Setli Bilgisayar
RTD	Direnç Sıcaklık Algılayıcı

1. GİRİŞ

Sıcaklık kontrolü, endüstride ve günlük yaşamda sıkça karşılaşılan bir parametre kontrolüdür. Termostatlarda yer alan, sıcaklığın belli değerinde sistemin ısısının artırılması ya da durdurulması işlemi gibi basit uygulamaların yanında çeşitli kimyasal süreçlerde, ortam sıcaklığının oldukça hassas biçimde kontrol edilmesi gereken endüstri uygulamalarında ve medikal odalarda küvöz vb. ortamların izlenmesinde bu kontrolün verimli biçimde gerçekleştirilebilmesi esastır.

Sıcaklık kontrolünde amaçlanan, temelde ortam sıcaklığını belli bir değerde sabit tutmak olabileceği gibi tasarlanan sistemin ya da tesisin konumuna ve kullanım amacına göre sıcaklığı bir değer aralığında sabit tutma hedefi de taşıyabilir. Her durumda istenilen bir değer, ortamdan elde edilen veri sonucunda hesaplanan bir fark değer ve bu fark değeri işleyecek verimli bir denetleyici mevcut olacaktır. Denetleyicinin karşılaştırma algoritması ve tekniği, elektronik bileşenlerin gelişmesiyle birlikte çeştili alternatifler değerlendirilebilecek noktaya ulaşmıştır. Bu alandaki ilk çalışmalarda karşılaştırıcı olarak işlemsel yükselteçler kullanılsa da (Savaş, 1998; Kalender, 1991) günümüzde mikrodenetleyicilerin gelişmesi, fiyatlarının ucuzlaması ve diğer çevre birimleriyle etkileşimlerinin genişlemesi nedeniyle (Başar, 2002) bu entegre devrelerin dahili yeteneği olan darbe genlikli modülasyon (PWM) tekniğinin, karşılaştırıcı amaçlı kullanılması avantaj olacaktır. Bu yetenek, DC motor hız kontrolü (Duran, 2001) ve güç elektroniğinde DC-DC dönüştürücü (Gök, 2005) gibi pek çok uygulamada da kullanım alanı bulmaktadır.

Önceki çalışmalarda hep sıcaklık algılayıcısı olarak analog bir bileşen kullanılmış (Savaş, 1998; Kalender, 1991) kullanılan mikrodenetleyicinin marka ve modeline göre onun dahili A/D dönüştürücüsü kullanılmış ya da harici bir A/D dönüştürücü eleman kullanılarak sıcaklık değeri mikrodenetleyicinin sayısal dünyasına taşınmış ve kontrol algoritmaları gerçekleştirilmiştir. Ancak sıcaklığı kendi içerisinde doğrudan sayısala çeviren ve hatta bu değeri ayarlanabilir çözünürlükte mikrodenetleyiciye aktaran entegre devre tipinde sayısal bir sıcaklık algılayıcının kullanılması hem harici bir dönüştürücü entegre kullanmaktan ve onu mikrodenetleyiciyle ilişkilendirmek-

ten tasarruf ettirecek hem de dahili A/D dönüştürme işlemini kullanmaya gerek bıraktırmayacaktır. Bu çalışmada bu amaçla, sıcaklık algılayıcı olarak mikrodenetleyiciye 12 bit çözünürlükte veri gönderebilen sayısal bir bileşen kullanılmıştır. Bununla birlikte, sistem tepkisinin elde edilmesinde ve uygulanan kontrol algoritmalarının çıkışlarının izlenmesinde değişen ortam sıcaklığının gerilimle orantılı olarak zaman ekseninde çizdirilmesi osiloskop üzerinde gerçekleştirilmiştir ve bunu sağlamak için harici R2R merdiven biçiminde bir D/A çevirici devresi hazırlanmıştır. Bu devrenin harici bir A/D entegresi ile mikrodenetleyici iletişimini sağlamaktan çok daha rahat biçimde yapılabileceği kolaylıkla gözlenebilir. Bu çalışmada kullanılan ısıtıcı eleman, ısıtma paspası olarak adlandırılan ve ısıl sistemin tabanına yayılan silikon tabakalar içerisine sarılan tellerden oluşmuş esnek tipte özel bir ısıtıcıdır.

Bu tezde; Microchip firmasının 16X ürün ailesine ait RISC temelli bir mikrodenetleyici olan PIC 16F877 entegresi kullanılarak model bir ısıl sistemin sıcaklık kontrol işlemi, On/Off, Oransal, Oransal + İntegral, Oransal + Türevsel ve Oransal + İntegral + Türevsel denetleyici algoritmalarının mikrodenetleyici yazılımı üzerinde gerçeklenmesi ile sağlanmış ve diğer çevre birimlerle veri aktarımı sonucunda laboratuvar ortamında kullanılabilecek bir kontrol sistemi ortaya çıkarılmıştır.

İkinci bölümde, mikrodenetleyici tabanlı sıcaklık algılama, ölçüm ve denetleyici algoritmalarının gerçeklenebildiği reel bir sistem mimarisi anlatılmış, projede kullanılan aygıtlar hakkında genel bilgi verilmiş, mimari bileşenlerinin matematiksel modellemeleri yapılmış, sistem ölçüm sonuçları ve transfer fonksiyonu belirlenmiştir.

Üçüncü bölümde, sistemin matematiksel modeli Matlab Simulink benzetim yazılımı aracılığıyla hazırlanmıştır. Benzetim sonuçları kullanılarak denetleyici yazılımlarının parametre değer aralıkları saptanmıştır.

Dördüncü bölümde denetleyici tiplerinin genel şekli blok diyagramlar ile birlikte açıklanmakta, mikrodenetleyicinin kullanacağı yazılım algoritmaları listelenmekte ve ilgili yazılım geliştirme araçları, geliştirilen yazılım ve yazılımın programlayıcı aracılığıyla mikrodenetleyiciye aktarılması yer almaktadır.

Beşinci bölümde, model sistem üzerinde gerçeklenen yazılım sonucunda osiloskop aracılığıyla elde edilen deneysel sonuçlar gösterilmektedir.

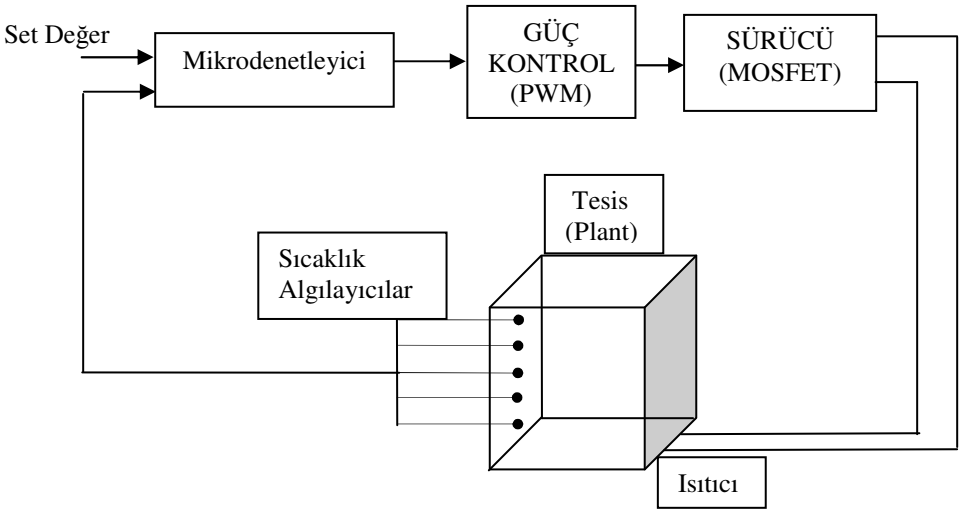
Tezde kullanılan tüm elektronik bileşenlere ait veri kitapçıkları (datasheet), baskı devre çizimleri, fotoğraflar ve diğer dokümanlar da “Ek” bölümünde yer almaktadır.

2. MİKRODENETLEYİCİ KONTROLLÜ ISIL SİSTEM TASARIMI

2.1. Mikrodenetleyici ile Sıcaklık Kontrolü

Sıcaklık kontrol süreçleri genellikle sürece ısı çıkışı sağlayan bir tür aygıttan oluşmaktadır (rezistans bir ısıtıcı gibi). Böyle bir sistemde bir sıcaklık algılayıcısı sıcaklığı algılamak ve mikrodenetleyiciye geri bildirim sağlamak için kullanılır. Bu geri bildirim üzerine, mikrodenetleyici ısıtıcıyı kontrol eder (bazı sistemlerde sistem fanını da) (İbrahim, 2002).

Bu çalışmada tasarlanan ısıl sistemin basit bir diyagramı Şekil 2.1'de gösterilmiştir.



Şekil 2.1. Mikrodenetleyici kontrollü ısıl sistem genel diyagramı

2.2. Sistem Mimarisi

Şekil 2.1'de genel diyagramı verilen ısıl sistemin mimarisi, blok diyagramlar içerisinde gösterilen bileşenler ile oluşturulur. Bu bileşenlerin donanımsal nitelikleri ve işlevleri mikrodenetleyici yazılımı ile yönetilmektedir.

Çoğu sıcaklık kontrol sistemi; spesifik ısı katsayısı, sıcaklık katsayısı, hacim vb. parametrelere bağlı olan bir zaman sabiti ve ölü zaman gecikmesi içeren birinci merte-

beden bir sisteme yaklaştırılabilir. Sayısal sıcaklık kontrolünün en kolay biçimi on-off tipi bir kontrol kullanmaktır. Bu kontrol formu hemen hemen tüm bina içi termostatlarda kullanılmaktadır (İbrahim, 2002).

On-off tipi kontrol; ucuz, basit, olmasına karşın bu çalışmada tek kontrol algoritması olarak kullanılmamış, diğer kontrol algoritmalarının mikrodenetleyici aracılığıyla aynı tesis üzerinde gerçeklenmesi sağlanmaya çalışılmıştır. Bu işlem, sürekli değişken denetleyicilerin kullanımı ile mümkün olmaktadır.

Sürekli değişken denetleyiciler, on-off tipi denetleyicilerdeki çevrimi (cycling) elimine etmek için tasarlanmıştır. Bu tip denetleyiciler, istenilen sıcaklık değeri ile gerçek sıcaklık arasındaki farkı ölçer ve bu değeri kullanarak ısıtıcıya ne kadar güç aktarılacağını belirler. Ölçülen değer istenen değere yakınsa ısıtıcıya daha az güç aktarılır.

Oransal denetleyici, yükselme süresini azaltıcı bir etkiye sahiptir ancak kalıcı durum hatasını hiçbir zaman elimine etmeyecektir. Oransal kazancın artırılması yükselme zamanını azaltacaktır bununla birlikte aşım miktarını (overshoot) da artıracaktır. İntegral denetleyicinin kalıcı durum hatasını elimine etme etkisi mevcuttur fakat bu durumda geçici durum tepkisi daha kötüye gidebilir. Gereğinden fazla integral işlemi büyük aşımlara neden olacak ve bir osilasyon davranışı sergileyecektir. Yine büyük çapta integral işlemi yükselme zamanını azaltma ve sistem yerleşme zamanını artırma eğilimindedir. Türevsel bir denetleyici ise sistem kararlılığını artırıcı, aşımı azaltıcı ve geçici durum tepkisini düzeltici bir etkiye sahiptir. Türevsel işlemin artırılması hem aşımı hem de sistem yerleşme zamanını azaltacaktır (İbrahim, 2002).

2.3. Mimari Bileşenleri

2.3.1. Sıcaklık algılayıcıları

Bir sıcaklık kontrol uygulamasında doğru algılayıcının seçimi her zaman kolay değildir. Bu işlem; sıcaklık aralığı, gerekli doğruluk, ortam, tepki hızı, kullanım kolaylığı, maliyet ve değiştirilebilirlik gibi parametrelere bağlıdır.

Sıcaklık ölçümü için çeşitli tipte algılayıcılar mevcuttur. Isılçift (thermocouple), termistör ve RTD ler gibi klasik algılayıcılar geniş biçimde kullanım alanı bulmaktadır. Entegre devre biçimindeki algılayıcılar ve uzaktan sıcaklık ölçüm (radiation thermometry) aygıtları gibi yeni nesil algılayıcılar sadece sınırlı uygulamalar için kullanım alanı bulsalar da popüler olmaktadır (İbrahim, 2002).

Bu projede Dallas Semiconductor firmasının DS18B20 ürün kodlu sayısal sıcaklık algılayıcısı kullanılmıştır. -10°C ila +85 °C arasında ±0,5°C doğrulukla ölçüm yapan bu algılayıcı bu firmanın ticari standardı olan 1-wire arabirimiyle sadece bir port kullanarak mikrodenetleyici ile haberleşmektedir. Sıcaklığı 12 bit sayısal sözcük (word) olarak mikrodenetleyiciye göndermektedir. Bu algılayıcının kullanılma sebebi, sıcaklığın doğrudan sayısal olarak mikrodenetleyiciye aktarılabilmesine olanak sağlaması böylelikle mikrodenetleyici içerisinde bir A/D işlemi yapılmasına gerek bırakmamasıdır. Ancak sistemin birim basamak tepkisinin osiloskop ile çizdirilmesinde ve yine denetleyici algoritmalarına karşın sistem tepkisinin gözlenmesinde harici bir D/A devresi kullanılmıştır.

2.3.2. Isıtıcı

Bu çalışmada kullanılan ısıtıcı, preslenmiş silikon plastik iki kalıp arasına sarılmış nikel krom tellerden oluşan özel tipte bir malzemedir ve yurt dışından temin edilmiştir. Isıtıcı paspas (mat) olarak da adlandırılan bu yapı, sistemin laboratuvar ortamında modüler olarak kullanılmasına olanak tanıması açısından tellerin ve ısıtma tertibatının açıkta olmaması gereği göz önünde bulundurularak seçilmiştir. Esnek yapısı,

ısıtacağı hacme göre kesilebilmesi önemli olmakla birlikte elektriksel anlamda 230 V AC gerilim, 100 W güç ve 530 Ω direnç karakteristiğine sahiptir ve boyutları 15x18 cm'dir. Depolama tankları, konteynerler, koltuk ısıtıcılar ve taban ısıtma gibi çeşitli uygulamalarda kullanım alanı bulmaktadır. Kullanım sırasında yüzey üzerine doğrudan montajı esastır böylelikle yüzey boyunca herhangi bir noktadaki hava aralıklarından sakınılmış olur. Bu amaçla arka yüzeyinde yapışkan yer almaktadır. Maksimum 150 °C'lik bir ortam sıcaklığına çalışma dayanımı mevcuttur.

2.3.3. Tesis

Sistem tepkisinin gözlenmesindeki belirleyici parametrelerden biri olan hacim için, ısıtıcının gücü ile orantılı olarak dikdörtgen prizması biçiminde cam bir yapı seçilmiştir. Isı yalıtımı sağlama açısından, ısıtıcının yerleştirileceği dip yüzeye alüminyum folyoya sarılmış taş yünü yerleştirilmiştir. Aynı biçimde prizmanın iç yüzeylerine de foam board levha döşenmiştir. Prizmanın kapağı içerisine sistemin soğutmasında kullanılacak olan fan monte edilmiş ve sıcaklık algılayıcılar ve ısıtıcı ve fan kabloları ayrı kablo kanalları içerisinden döşenmiştir. Bunun nedeni deneysel çalışma sırasında sıkça karşılaşılan sıcaklık verisinde bozulma etkisinin en aza indirilmesinin amaçlanmasıdır. Ortam sıcaklığı 85 °C'ye ulaşabildiğinden prizma içerisinde tutturucu malzeme olarak kırmızı renkli özel RTV silikon kullanılmıştır. Tesisin açık döngü birim basamak tepkisi (unit step response) ısıtıcıya tam güç verilmişken osiloskop aracılığıyla gözlemlenmiştir.

2.3.4. PIC 16F877 mikrodenetleyici

Bu çalışmada kontrol bileşeni olarak Microchip firmasına ait PIC 16F877 ürünü kullanılmıştır. PIC 16F87X serisi öncelikle, PIC 16CXX ailesinin özelliklerini taşır (11, 12). PIC 16CXX'de Harvard mimarisi kullanılmıştır. Von Neuman mimarisinde, veri ve program belleğine aynı yoldan erişilebilirken, bu mimaride program belleği ve veri belleğine erişim farklı boylarda yapılır. Veri yolu (databus) 8 bit genişliğindedir. Aynı anda, veri belleğine 8 bit genişliğindeki bu yolla erişilirken; program belleğine program yolu ya da adres yolu (program bus / adress bus) denilen 14 bit genişliğin-

deki diğer bir yolla erişilir. Bunun için PIC 16F87X ve PIC 16F84'de komut kodları (opcode), 14 bittir. 14 bitlik program belleğinin her bir adresi, bir komut koduna (Instruction Code / Instruction Word) karşılık gelir. Dolayısıyla her komuta bir çevrim süresinde (cycle) erişilir ve komut kaydedicisine yüklenir. Komut kaydedicisi, CPU tarafından kullanılan bir kaydedicidir ve dallanma komutları dışındaki bütün komutlar, aynı çevrim süresinde çalıştırılırlar. Bu sırada program sayacı, PC (Program Counter) bir artar. Dallanma ya da sapma komutları ise, iki ardışık periyotta çalıştırılır ve program sayacı PC, iki arttırılır.

16F87X Mikrodenetleyici ailesi aşağıdaki temel özellikleri taşır:

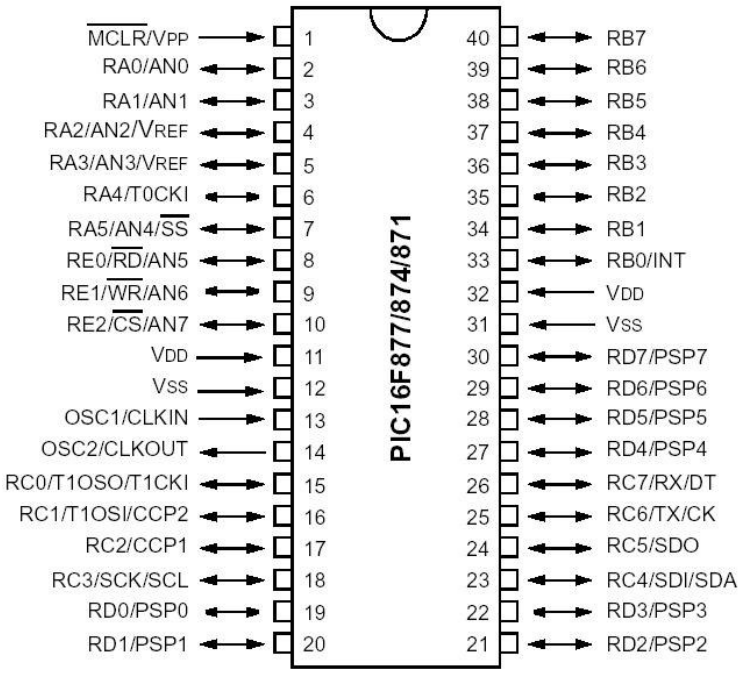
- CPU azaltılmış komut seti RISC temeline dayanır.
- Öğrenilecek 35 komut vardır ve her biri 14 bit uzunluktadır.
- Dallanma komutları iki çevrim (cycle) sürede, diğerleri ise bir çevrimlik sürede uygulanır.
- İşlem hızı 16F877'de DC-20 MHz'dir. (16F877'de bir komut DC-200 ns hızında çalışır.)
- Veri yolu (databus) 8 bittir.
- 32 adet SFR (Special Function Register) olarak adlandırılan özel işlem kaydedicisi vardır ve bunlar statik RAM üzerindedir.
- 8 Kword'e kadar artan flash belleği 1 milyon kez programlanabilir.
- 368 Byte'a kadar artan veri belleği (RAM), ve 256 Byte'a kadar artan EEPROM veri belleği vardır.
- Pin çıkışları PIC 16C73B/74B/76 ve 77 ile uyumludur.
- 14 kaynaktan kesme yapabilir.
- Yığıt (stack) derinliği 8'dir.

- Doğrudan, dolaylı ve göreceli adresleme yapabilir.
- Power-on Reset (Enerji verildiğinde sistemi resetleme) özelliği vardır.
- Power-up Timer (Power-up zamanlayıcı) mevcuttur.
- Osilatör Start-up Timer (Osilatör başlatma zamanlayıcısı) mevcuttur.
- Watch-dog Timer (Gözleyici), devre içi RC osilatör mevcuttur.
- Programla kod güvenliğinin sağlanabilmesi özelliği mevcuttur.
- Devre içi Debugger (Hata ayıklamakta kullanılabilecek modül) mevcuttur.
- Düşük gerilimli programlama özelliği vardır.
- Flash ROM program belleği (EEPROM özellikli program belleği) mevcuttur.
- Enerji tasarrufu sağlayan, uyku –Sleep Modu özelliği vardır.
- Seçimli osilatör özellikleri mevcuttur.
- Düşük güçle, yüksek hızla erişilebilen, CMOS-Flash EEPROM teknolojisi mevcuttur.
- Tümüyle statik tasarıma sahiptir.
- 2 pinle programlanabilme ve yalnız 5V girişle, devre içi seri programlanabilme özelliği mevcuttur.
- İşlemcinin program belleğine, okuma/yazma özelliği ile erişimi mevcuttur.
- 2,0 V – 5,0 V arasında değişen geniş işletim aralığı mevcuttur.
- 25 mA'lik kaynak akımına sahiptir.
- Devre içi, iki pin ile hata ayıklama özelliği mevcuttur.

- Geniş sıcaklık aralığında çalışabilme özelliği mevcuttur.
- Düşük güçle çalışabilme özelliği mevcuttur.

Çevresel özellikleri ise şöyle sıralanabilir:

- TMR0: 8 bitlik zamanlayıcı, 8 bit önbölücülüdür.
- TMR1: Önbölücülü, 16 bit zamanlayıcı, uyuma modundayken dış kristal zamanlayıcıdan kontrolü arttırılabilir.
- TMR2: 8 bitlik zamanlayıcı, hem önbölücü hem de sonbölücü sabiti mevcuttur.
- İki Capture / Compare / PWM modülü mevcuttur.
- 10 bit çok kanallı A/D çevirici mevcuttur.
- Senkron seri port (SSP), SPI (Master mod) ve I2C (Master Slave) ile birliktedir.
- Paralel Slave (bağımlı) Port, 8 bit genişlikte ve dış RD, WR, CS kontrolleri mevcuttur.
- USART/SCI, 9 bit adres yakalamalıdır.
- BOR Reset (Brown Out Reset) özelliği mevcuttur.



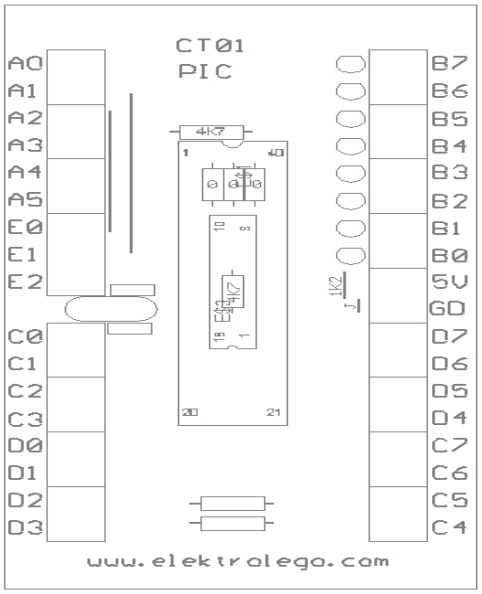
Şekil 2.2. PIC 16F877 bacak (port) bağlantıları

2.3.5. Kontrol kutusu

PIC mikrodenetleyicinin ve çevre birimlerinin içerisine yerleştirildiği kontrol kutusu, sıcaklık algılayıcılardan gelen veri kabloları ile bağlantı yapılan ve mikrodenetleyici ile tesis arasındaki tüm veri haberleşmesinin yapıldığı ve kontrol edildiği bir kutudur. Laboratuvar ortamında tesis ile veri kablolarıyla üzerinde gerçekleştirilen sıcaklık kontrolü için kullanıcıya set değerleri girmeye yarayan butonlar, kontrol sonuçlarını gösteren LCD ekran kutunun ön panelinde yer almaktadır. Kutu içerisindeki baskı devre kartları tek bir kart olarak tasarlanmamış; mikrodenetleyici kartı, röle ve D/A dönüştürücünün yer aldığı kart, güç kartı ve PWM kartı ayrı ayrı monte edilmiştir.

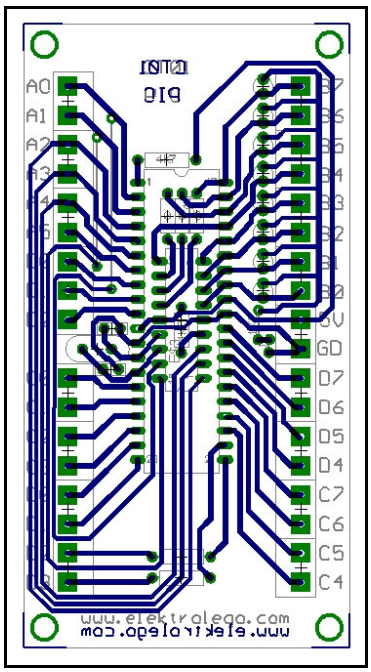
Mikrodenetleyici Kartı

PIC mikrodenetleyicinin üzerinde yer aldığı bu kart için mikromodül olarak adlandırılan (14) ve osilatör devresinin ve mikrodenetleyici giriş/çıkış portlarının klemenslerinin hazır olarak yer aldığı kart kullanılmıştır. 4 MHz'lik bir kristal osilasyon frekansını oluşturmaktadır.



Şekil 2.3. Mikrodenetleyici kartı yerleşim şeması

Bu kartın baskı devre şeması aşağıda verilmektedir:

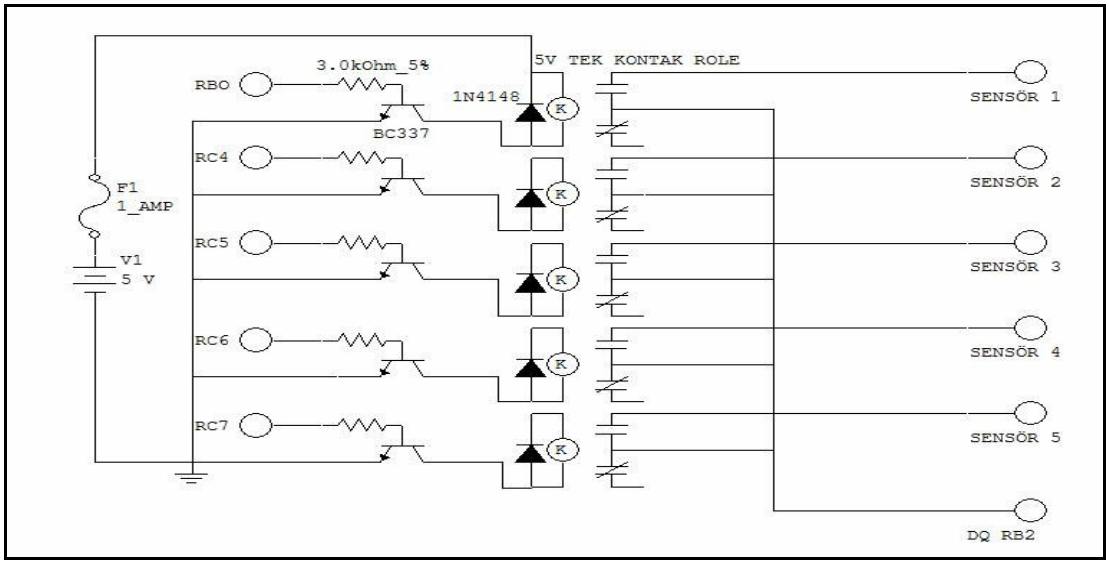


Şekil 2.4. Mikrodenetleyici kartı baskı devre şeması

Röle ve D/A Dönüştürücü Kartı

Röle Devresi

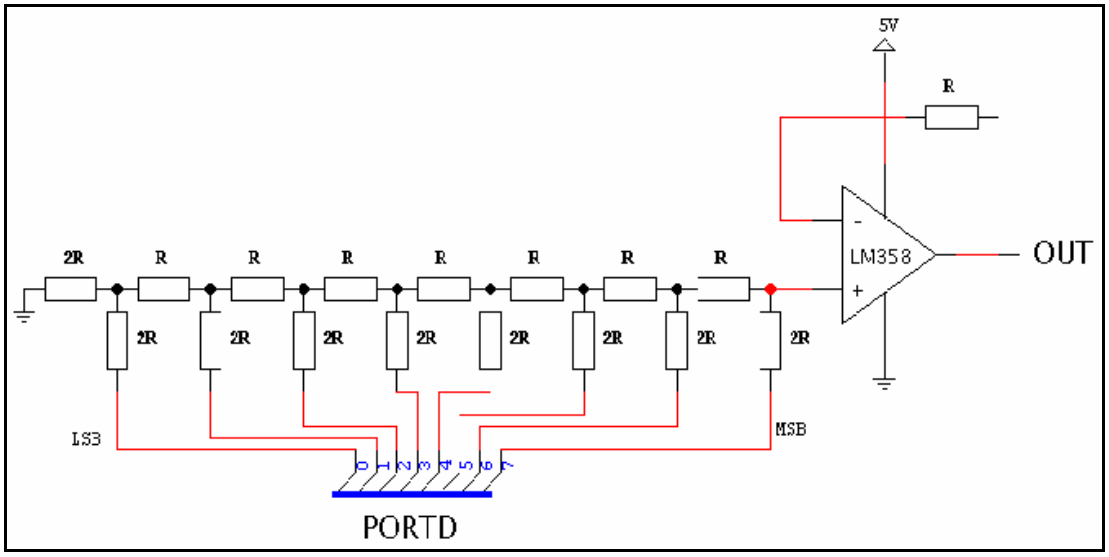
Tesis içerisinde beş farklı bölgede sıcaklık algılayıcısı mevcuttur. Isınan havanın alttan üste doğru yükseleceği dolayısıyla ısıtıcıya yakın yerdeki algılayıcılar ile üst bölümdeki algılayıcılar arasında değer farkı olacağı açıktır. Kullanılan sıcaklık algılayıcıları sayısal çıkış vermekte ve üretici firmaya ait "1-wire" teknolojisiyle mikrodenetleyici ile haberleşmektedir. Yazılımda kullanılan derleyicideki kısıtlama nedeniyle bu protokolde, mikrodenetleyicinin yalnızca bir çıkış portu aynı anda algılayıcı ile veri iletişiminde bulunabilmektedir. Dolayısıyla her bir saniyede tek bir algılayıcıdan gelen veri mikrodenetleyicinin RB2 portundan alınmaktadır. Tesiste beş adet sıcaklık algılayıcı bulunduğundan her birinden gelen veriyi mikrodenetleyiciye aktarmak için beş adet röle içeren devre tasarlanmıştır. Röleler her saniyede anahtarlama yaparak algılayıcıyı RB2 portu ile ilişkilendirir ve algılayıcıdan gelen veri yazılımda bir değişkene atılır ve beş saniye sonunda değişkendeki bu değerin aritmektik ortalaması alınarak ortam sıcaklığı hesaplanmış olur. Sistemin geneli değerlendirildiğinde toplam örnekleme zamanı beş saniye olmaktadır.



Şekil 2.5. Röle devre şeması

D/A Dönüştürücü Devresi

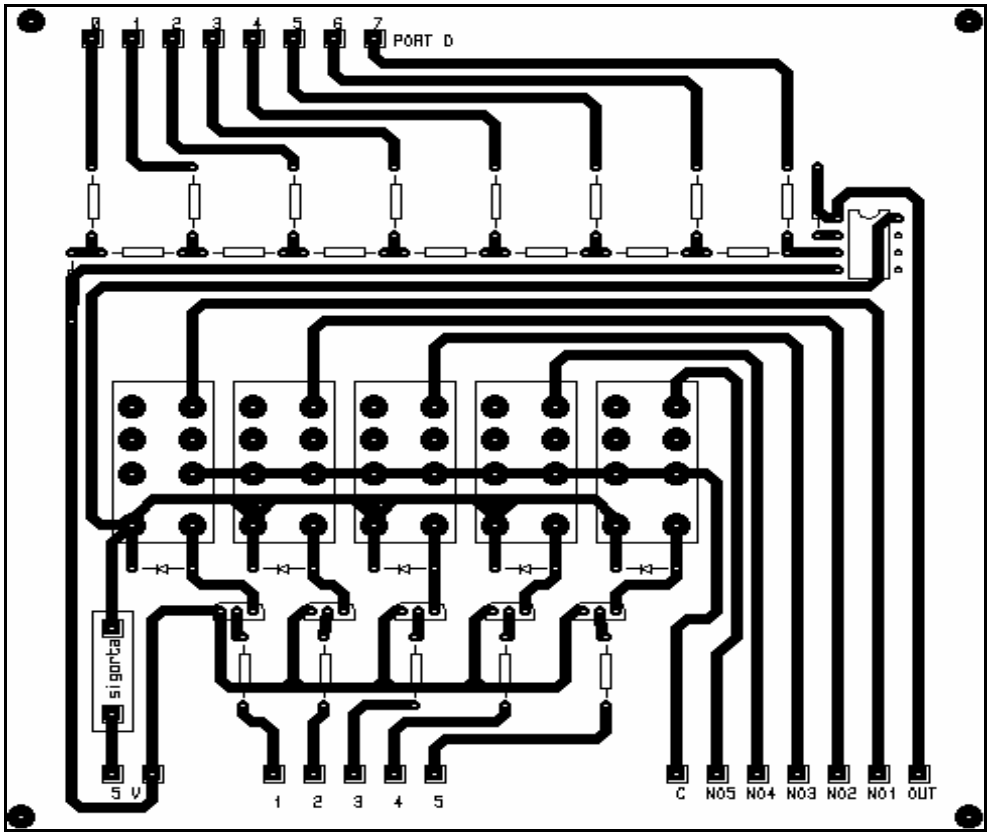
Sayısal/Analog dönüştürme işlemi için tasarlanan devre R2R merdiven biçiminde adlandırılan bir işlemsel yükselteç ve sıra dirençlerden oluşan aşağıdaki devredir:



Şekil 2.6. D/A dönüştürücü devre şeması

Burada R direnci olarak 10 kΩ'luk bir direnç kullanılmıştır. PIC mikrodenetleyicinin giriş/çıkış portlarından olan D portu bütünüyle D/A dönüştürme işlemi için ayrılmıştır. Devrede OUT olarak adlandırılan bölümden osiloskoba çıkış verilmekte böylelikle sistemin sıcaklık-gerilim ilişkisi gözlemlenebilmektedir.

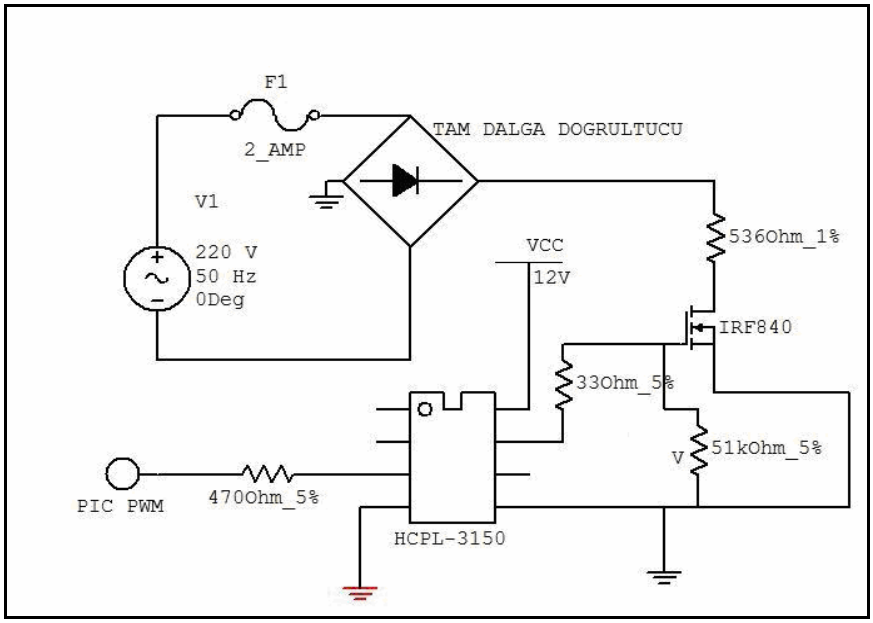
Rölelerin ve D/A dönüştürücünün yer aldığı kartın baskı devre şeması Şekil 2.7'deki gibi tasarlanmıştır:



Şekil 2.7. Röle&D/A dönüştürücü kartı baskı devre şeması

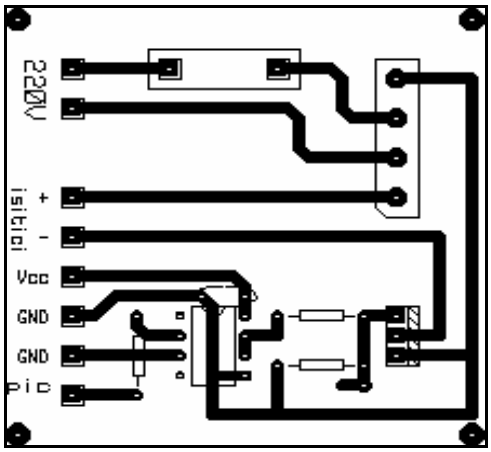
PWM Kartı

Isıtıcının sürüldüğü ısıl enerjinin belirlenmesinde mikrodenetleyicinin PWM (darbe genişlikli modülasyon) tekniğinden faydalanılmaktadır. Bu amaçla üzerinde bir MOSFET ve Optocoupler bulunan ve mikrodenetleyicinin PWM çıkışı olan RC2 portunun bağlandığı PWM kartı tasarlanmıştır. Kartın devre şeması Şekil 2.8'deki gibidir:



Şekil 2.8. PWM kartı devre şeması

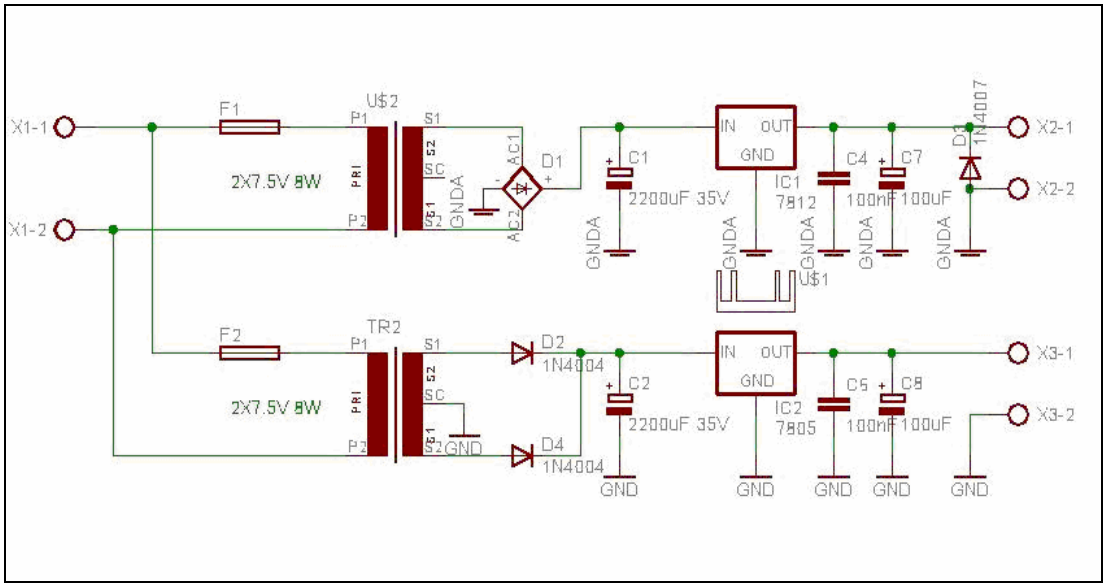
Mikrodenetleyici ile anahtarlama devreleri arasındaki referans bölümlerinin (toprak) elektriksel izolasyonunu gerçekleştirmek için izolasyonlu güç kaynağı kullanılmıştır. Hem 12 V hem de 5 V DC güç verebilen bu kaynağın topraklarından biri kırmızı renkli olarak işaretlenmiştir. Kullanılan optocoupler anahtarlama devresinde oluşabilecek herhangi bir kaçak akım, parazit vb. zararlı etkinin mikrodenetleyici kartına ulaşmaması için konulmuştur. Kartın baskı devre şeması Şekil 2.9'da yer almaktadır:



Şekil 2.9. PWM kartı baskı devre şeması

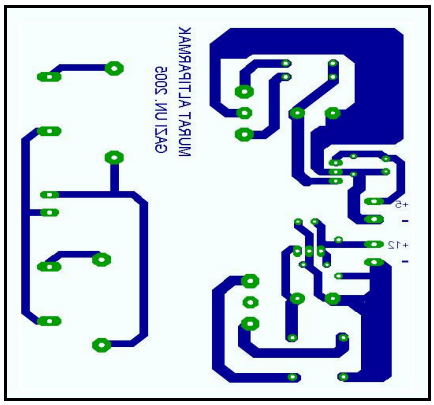
Güç Kartı

Kontrol kutusunun tüm güç ihtiyacını karşılamak üzere tasarlanmıştır. PIC mikrodenetleyici 5 V DC gerilimle çalışmakta ancak PWM kartında yer alan optocoupler 12 V DC gerektirmektedir. Ayrıca sistem fanı da yine 12 V DC gerilimle çalışmaktadır. Elektriksel izolasyonu sağlamak için bu kartta iki adet trafo kullanılmış ve iki farklı toprak elde edilmiştir.



Şekil 2.10. Güç kartı devre şeması

Kartın baskı devre şeması ise Şekil 2.11'de gösterilmiştir:

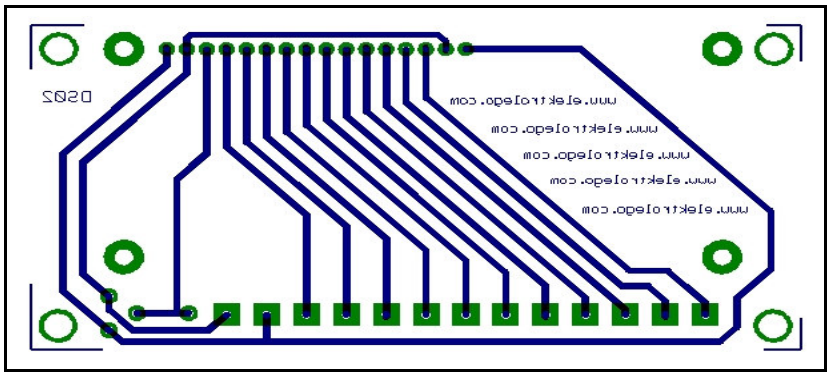


Şekil 2.11. Güç kartı baskı devre şeması

LCD Kartı

Mikrodenetleyici ile çevre birimleri arasındaki tüm veri alışverişi kontrol kutusu üzerindeki LCD ekran aracılığıyla görüntülenebilmektedir. Sistem başlatıldığında ortam sıcaklığı, istenilen set değer, uygulanacak kontrol tipi vb. bilgiler kullanıcıya sunulmakta ve kullanıcı ekranın yanındaki butonlar aracılığıyla bütün bu adımları gerçekleştirebilmektedir. LCD ekran olarak yaygın biçimde kullanılan Hitachi 44780 denetleyicili 2*16 satırlık LM016 tipi ekran kullanılmıştır. Mikrodenetleyici kartında olduğu gibi bu kart ta mikromodül olarak entegre edilmiştir. Giriş-çıkış bağlantıları ve besleme klemensleri kart üzerinde adreslenmiş ayrıca arka aydınlatması (backlight) için de bir pot yer almaktadır.

Şekil 2.12. LCD kartı yerleşim şeması



Şekil 2.13. LCD kartı baskı devre şeması

3. SİSTEMİN MATEMATİKSEL MODELİ

Sistemin matematiksel modelinin çıkarılmasında MATLAB Simulink programından faydalanılmıştır.

3.1. Matlab Simulink Hakkında Genel Bilgi

Simulink, karmaşık sistemleri tasarlama ve benzetim yapma olanağı vermektedir (15). Mühendislik sistemlerinde benzetimin önemi gün geçtikçe artmaktadır. Sistemlerin tasarımında büyük oranda bilgisayar benzetimlerinden faydalanmakta, mümkün olduğunda tasarımın test aşamaları da bilgisayarlar yardımıyla yapılmaktadır. Bu da prototiplere olan ihtiyacı azaltarak maliyetlerin büyük oranda düşmesini sağlamaktadır. Günümüzde mühendislik alanında en çok kullanılan programlardan birisi MATLAB'dir. Simulink, MATLAB ile birlikte bütünleşik olarak çalışan bir benzetim ortamıdır. Sürekli zamanlı ve kesikli zamanlı sistemleri veya her ikisini de içeren hibrit sistemleri desteklemektedir. Zengin blok kütüphanesi sayesinde içinde birçok altsistemi blok olarak barındırdığından sürükle-bırak yöntemiyle pek çok sistem birkaç dakikada kurularak benzetilebilir ve değişik durumlardaki tepkileri test edilebilir.

3.1.1. Simulink kütüphanesi

Simulink çalıştırıldığında ekrana Simulink kütüphanesi gelecektir. Benzetim yaparken kullanılacak bloklar kategorilere ayrılmış biçimde burada bulunmaktadır.

3.1.2. Blok diyagramlar

Her bir blok sürekli zamanda ya da kesikli zamanda çıkış veren temel bir dinamik sistemi ifade eder. Hatlar, blokların giriş ve çıkışları arasındaki bağlantıları gösterir. Blok diyagramdaki her bir blok belirli bir tip bloğun örneğidir. Bloğun tipi, bloğun giriş ve çıkışları, durumları ve zaman arasındaki bağıntıyı belirler.

3.1.3. Bloklar

Bloklar, Simulink'in nasıl simule edileceğini bildiği temel dinamik sistemleri temsil eder. Blokları programlama dillerinde kullanılan fonksiyonlara benzetebiliriz. Her biri dinamik sistem için yazılmış hazır fonksiyonlardır. Bize kalan sadece bu hazır fonksiyonları kullanarak sistemimizi kurmaktır. Bir blok şu bileşenlerin birinden veya birkaçından oluşur: Giriş kümesi, durum kümesi ve çıkış kümesi.

3.1.4. Durumlar

Bloklar durumlara sahip olabilirler. Durum, bloğun çıkışını belirleyen ve şimdiki değeri, bloğun önceki durumları ve/veya girişlerinin fonksiyonu olan bir değişkendir. Duruma sahip olan bir blok şimdiki durumunu belirlemek için geçmiş değerlerini kaydetmek zorundadır. Duruma sahip olan bloklara, hafızalı blok denilir. Çünkü bu bloklar o anki değerlerini belirlemek için geçmiş değerlerini kaydetmek zorundadırlar.

Simulink İntegral alıcı (integrator) bloğu duruma sahip bloklara bir örnektir. İntegrator bloğu benzetimin başlangıcından o anki zamana kadar giriş sinyalinin integralini çıkış olarak verir. O anki zaman adımındaki integral değeri, integrator bloğunun geçmişteki giriş değerlerine bağlıdır. Dolayısıyla integral, integrator bloğunun durumudur. Durumlu bloklara bir başka örnek de Simulink Hafıza (memory) bloğudur. Hafıza bloğu girişlerindeki değerleri o anda kaydedip ileriki bir zamanda çıkışına verir. Hafıza bloğunun durumları önceki giriş değerleridir.

Simulink Kazanç bloğu (Gain) durumsuz bloklara bir örnektir. Kazanç bloğu girişindeki değeri kazanç adı verilen bir sabitle çarparak çıkışına verir. Kazanç bloğunun çıkışı tamamiyle o anki giriş değeri ve sabit olan kazanç ile belirlenir. Dolayısıla Kazanç bloğunun durumu yoktur. Diğer bazı durumsuz bloklar Toplam (Sum) ve Çarpım (product) bloklarıdır. Bu blokların çıkışları tamamiyle girişlerinin bir fonksiyonudur (ilki için toplam, ikincisi için çarpımdır). Dolayısıyla bu blokların durumları yoktur.

3.1.5. Blok parametreleri

Birçok standart bloğun anahtar özellikleri parametrik hale getirilmiştir. Örneğin, Kazanç Bloğunun (Gain) kazancı bir parametredir. Her bir parametrik blok, blok parametrelerini belirleyebileceğimiz bir diyalog kutusu sunar. Blok parametrelerini belirlemek için MATLAB ifadeleri kullanılabilir. Simulink bu ifadeleri benzetimi çalıştırmadan önce hesaplar. Parametrelerin değerlerini benzetim esnasında değiştirebiliriz. Bu, parametrenin en uygun değerini interaktif bir şekilde belirlememize olanak sunar. Parametrik bloklar etkili bir biçimde benzer blok ailelerini temsil ederler. Örneğin, bir model oluştururken modeldeki her bir Kazanç (Gain) bloğunun kazanç değerlerini ayrı ayrı belirleyerek her bir Kazanç Bloğunun farklı davranmasını sağlayabiliriz. Blokların parametrik hale getirilmesi, her bir standart bloğun bir blok ailesini temsil etmesini sağlayarak, Simulink'in modelleme gücünü artırmaktadır.

3.1.6. Değiştirilebilir parametreler

Birçok blok parametresi değiştirilebilirdir. Benzetim yapılırken değeri değiştirilebilen parametreler değiştirilebilir parametrelerdir. Örneğin Kazanç bloğunun kazanç parametresi değiştirilebilir parametredir. Benzetim çalışırken bloğun kazancı değiştirilebilir. Bir parametre değiştirilebilir değilse ve benzetim çalışıyorsa Simulink parametreyi ayarlayan diyalog kutusunu engeller. Simulink belirlediklerimiz dışında bütün parametreleri değiştirilemez olarak belirlememize izin verir. Bu büyük modellerin çalıştırılmasını hızlandırır ve modelimizden daha hızlı bir şekilde kod üretilmesini sağlar.

3.1.7. Altsistemler

Simulink bize, kompleks sistemleri, blok diyagramları ile temsil edilen birbirine bağlı altsistemler şeklinde modellememize izin verir. Altsistemler, Simulink altsistem (subsystem) bloğuyla ve model editörüyle oluşturulabilir. Alt sistemleri, ana sistemlere istediğimiz derinliğe kadar gömerek hiyerarşik modeller oluşturabiliriz. Bir geçiş durumu olduğunda, bir tetikleme veya yetkilendirme girişi geldiğinde çalıştırılan

şarta bağlı çalışan alt sistemler oluşturulabilir.

3.1.8. Sinyaller

Simulink sinyal terimini blokların çıkış değerlerini belirtmekte kullanır. Simulink sinyal ismi, veri tipi (örn: 8-bit,16-bit veya 32-bit tamsayı), nümerik tip (Reel veya kompleks), ve boyutluluk (tek boyutlu veya 2-boyutlu dizi) gibi sinyal özelliklerini belirlememize olanak verir. Birçok blok herhangi bir veri veya nümerik tipte ve boyutta çıkışı kabul edebilir. Diğerleri de kabul edebildikleri sinyal özellikleri ile ilgili kısıtlamalar taşırlar.

3.1.9. Veri tipleri

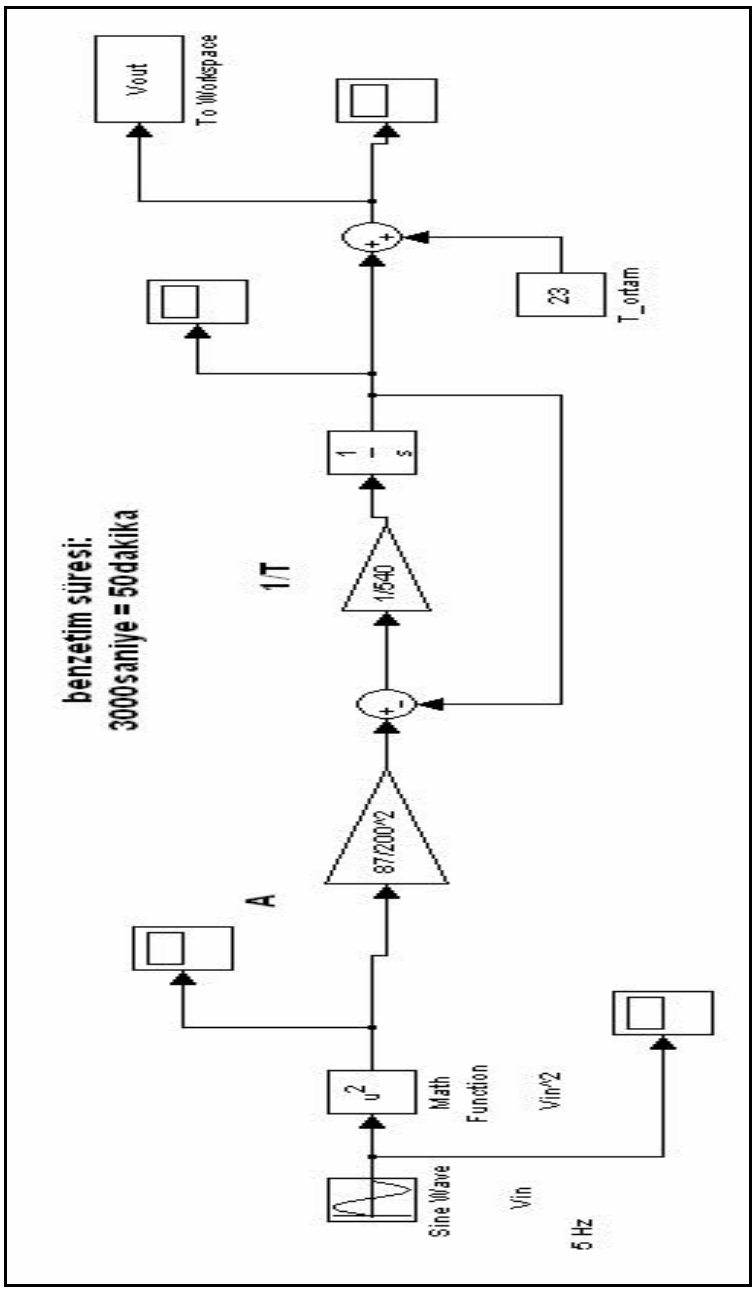
Veri tipi, verinin bilgisayardaki temsiline verilen addır. Simulink MATLAB'te desteklenen int8, double ve boolean gibi herhangi bir dahili veri tipini kullanabilir. Bunlara ek olarak Simulink kendine özgü iki veri tipi daha tanımlar:

- Simulink.Parameter
- Simulink.Signal

Bu Simulink'e özgü veri tipleri diğer genel veri tipleri ile tutulamayan Simulink'e özgü bazı bilgilerin tutulmasında kullanılırlar. Simulink, veri objeleri denilen Simulink veri tiplerini kullanarak parametre değerleri ve sinyal olarak kullanılmak üzere yeni veri tipleri oluşturulmasına olanak tanır. Her iki Simulink veri tipi de kullanılarak istenilen modellere özgü bilgileri saklayabilen yeni veri tipleri gerçekleştirilebilir.

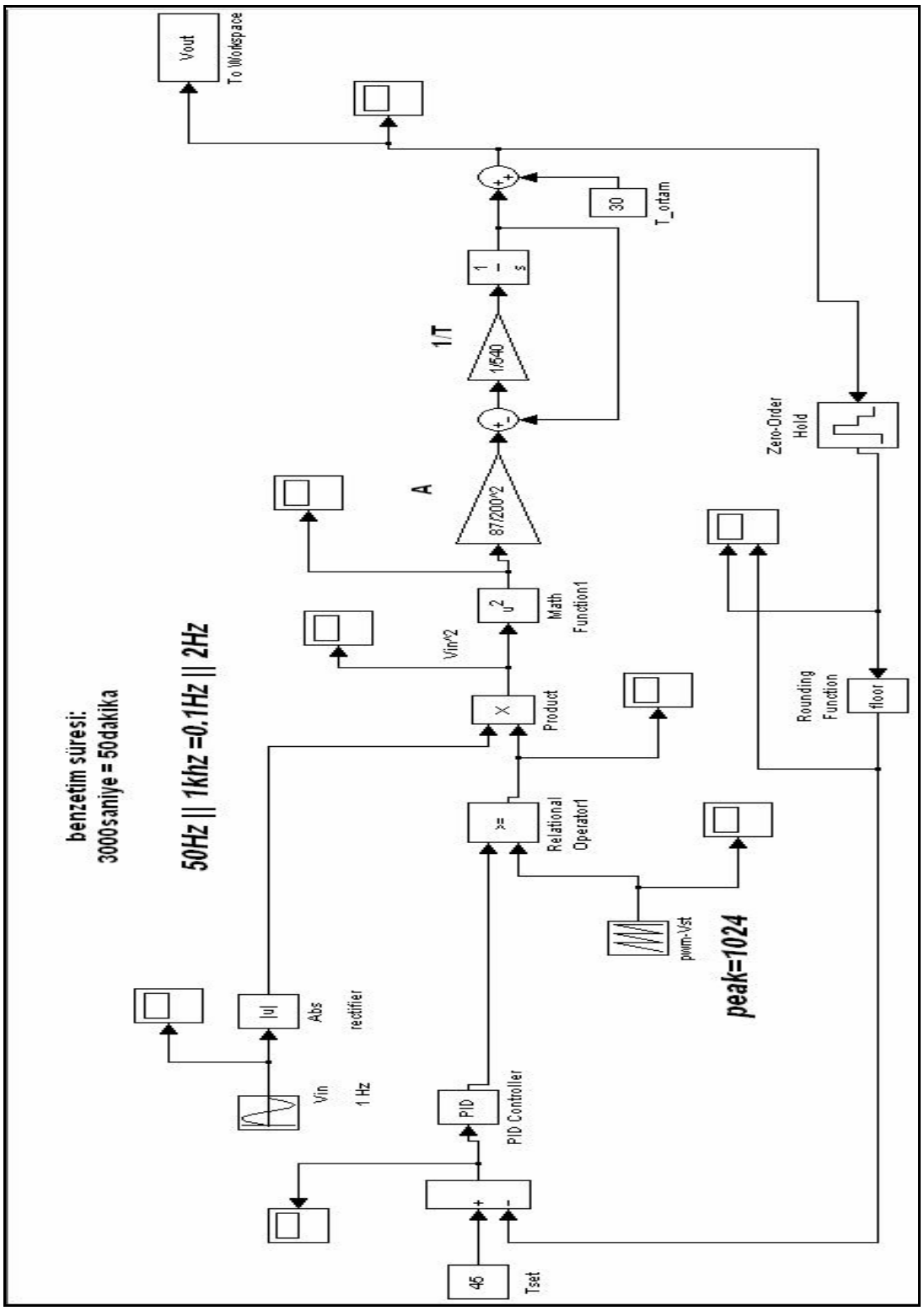
3.2. Isıl Sistemin Simulink ile Modellenmesi

Sıcaklık algılayıcıların yaklaşık 20 mV/°C karakteristiğinde olduğu gözlenmiştir. Isıl sistemin zaman sabitinin yaklaşık 540 s olduğu da ölçüm ile saptanmıştır. Bu doğrultuda sistemin açık döngü transfer fonksiyonu Şekil 3.1'deki gibi çizilebilir:



Şekil 3.1. Isıl sistemin açık döngü transfer fonksiyonu benzetimi

Sistemin mikrodenetleyici içeren bölümünün modellenmiş biçimi de Şekil 3.2'dedir.



Şekil 3.2. Isıl sistem simulink modeli

Zero-Order Hold ve Rounding Function blokları ile sıcaklık algılayıcılardan alınan bilgi mikrodenetleyicide değerlendirilmiş olur. Benzetim, gerçeğe yakın yapılmıştır. Floor (yuvarlatma) ile sıcaklık "50 51 49" biçiminde mikrodenetleyiciye atılır. Zero-Order Hold bloğu 5 saniyede bir değer değiştirir. Şekil 3.1 ve Şekil 3.2'deki Simulink bloklarındaki değerler, 5. Bölümde yer alan deneysel bulgulara göre belirlenmiştir.

4. ON/OFF, P, PI, PD, PID DENETLEYİCİLER ve SİSTEM YAZILIM ALGORİTMALARI

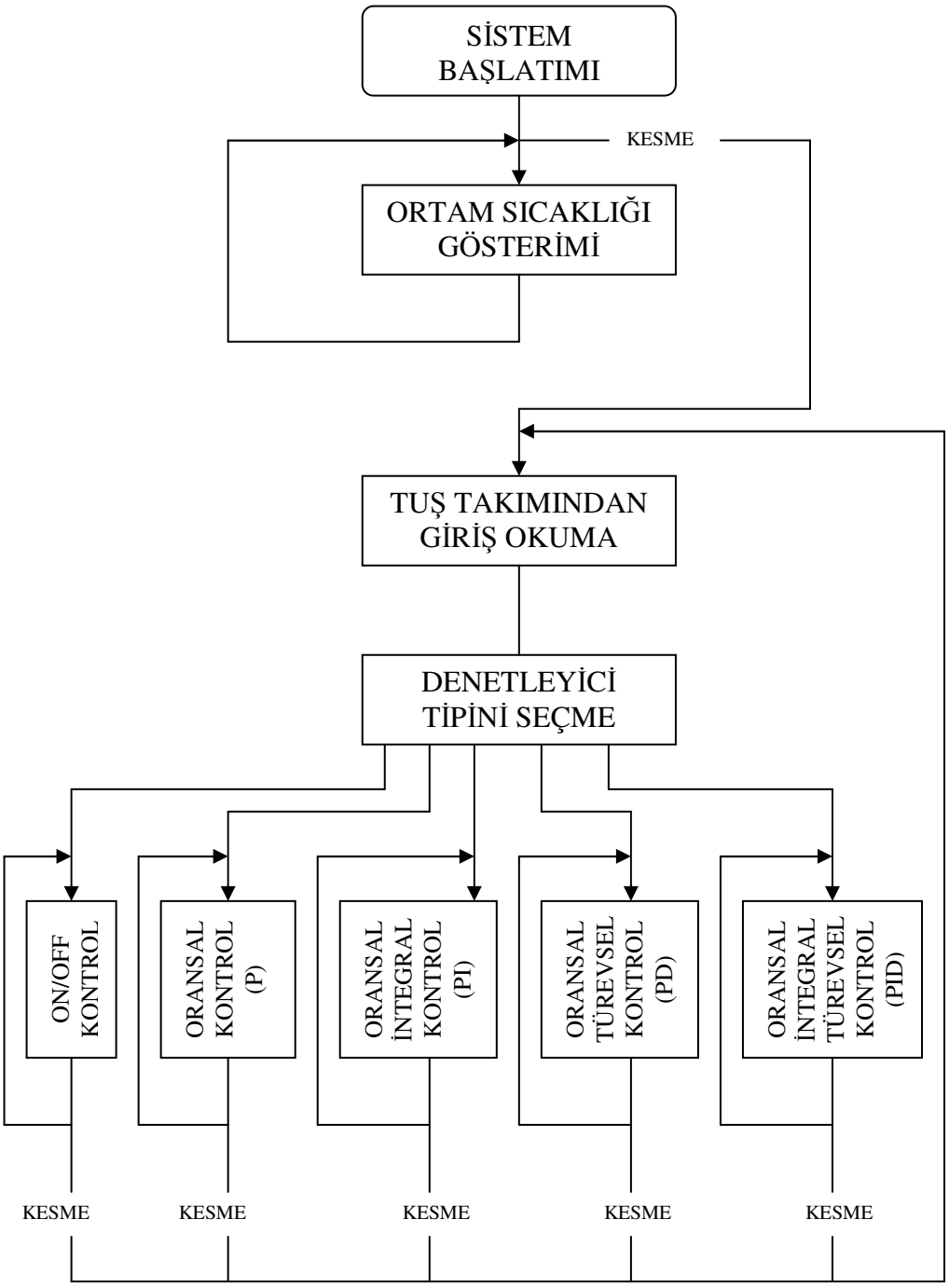
Bu bölümde denetleyici tipleri ile birlikte sistem yazılımı algoritmaları tanıtılmaktadır. Sistem yazılımı genel olarak iki bölümde incelenebilir: Birinci bölümde ısıl sistemde çeşitli kontrolleri (on/off, oransal vb.) yerine getiren programlar, ikinci bölümde ise bu kontrol programlarında kullanılan alt programlar bulunmaktadır.

4.1. Genel Akış Şeması

Şekil 4.1'de sistem yazılımının genel akış şeması görülmektedir. Sistem ilk açıldığında otomatik olarak reset olur, mikrodenetleyicinin giriş-çıkış portları, tuş takımı ve lcd ekran programın devam edecek kısmında istenilen işlevleri görebilmeleri için düzenlenir. Bu kısım "Sistem Başlatımı" olarak adlandırılabilir. Mikrodenetleyicinin giriş-çıkış portlarının düzenlenmesinden sonra yazılım, ortam sıcaklığının gösterildiği program parçasını uygulamaya başlar. Programın bu bölümü bir döngüden ibarettir. Saniyenin belli bir kesrinde ortam sıcaklığını gösteren veri elde edilir, işlenir, görüntülenir ve tekrar bu programın başına dönülerek bu işleme devam edilir.

Tuş takımında denetleyici tipini gösteren fonksiyon tuşlarından herhangi birine basılmışsa mikrodenetleyiciye kesme (INT) isteği gönderilir ve PIC bu kesmenin sonucu tuş takımından bilgi okuma programına sapar. Sapılan bu programda seçilen denetleyici tipinin gerektirdiği bilgilerin tuş takımından girilmesi gerekir. Bu durumda program, bu bilgilerin girilmesini bekleyecektir. Tuş takımından bu değerlerin girilmesinden sonra seçilen denetleyici tipinin uygulanmasına başlanır. Döngü ile bu program sürekli yürütülür.

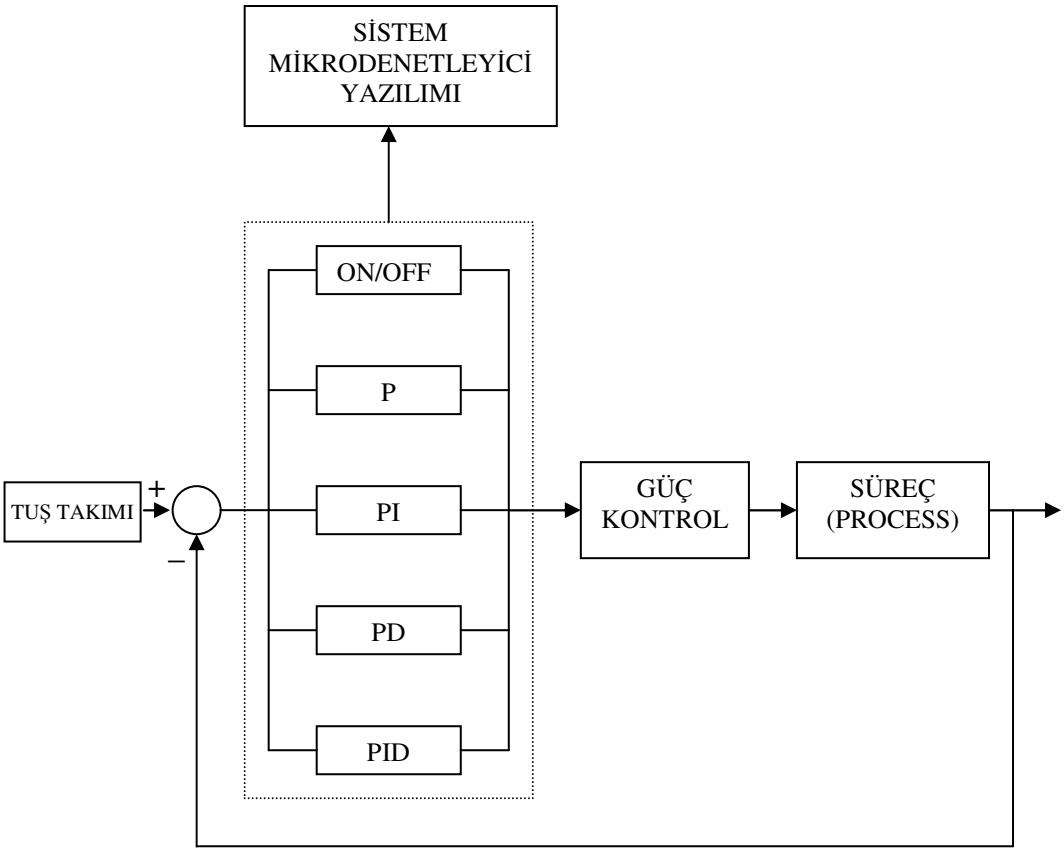
Döngüler reset işlemi (kurma) ya da set değere ulaşılması ve yeni değer girişi için basılmasıyla birlikte kesilir, tuş takımından giriş okuma programına sapılır ve önceki işlemler yinelenir.



Şekil 4.1. Sistem yazılımı genel akış şeması

4.2. Yazılım İle Denetleyici Tipinin Gerçekleştirilmesi

Yazılım ile gerçekleştirilecek, kapalı döngü sistemin kontrol bileşenleri Şekil 4.2'de görülmektedir.

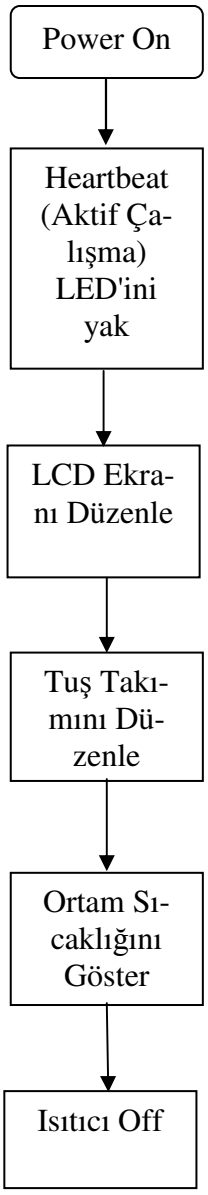


Şekil 4.2. Sistem kapalı döngü blok diyagramı

Şekil 4.2'de kesikli çizgi ile sınırlandırılmış bölüm, sistem yazılımının gerçekleştirmesi gereken bölümdür. Yazılımla bu bölümdeki işlemler gerçekleştirilmeden önce çıkış verisinin, tuş takımından ise giriş verilerinin elde edilmesi gerekir. Bu verilerin elde edilmesinden sonra sistemi sürecek hata sinyali, giriş ile ölçülen değerin aritmetiksel olarak çıkarılmasıyla elde edilir (Ogata, 1990). Sonraki işlem bu hata sinyalinin seçilen kontrol bileşeni tipine göre işlenip çıktı oluşturulmasıdır. Bu çıktı daha sonra güç kontrol ünitesi (PWM kartı) ile ısıl sistemi sürecek olan ısıya dönüştürülür.

Sistem yazılımının tümü Şekil 4.1 ve yukarıda anlatılan sıraya uygun olarak verilmektedir.

4.2.1. Sistem başlatımı

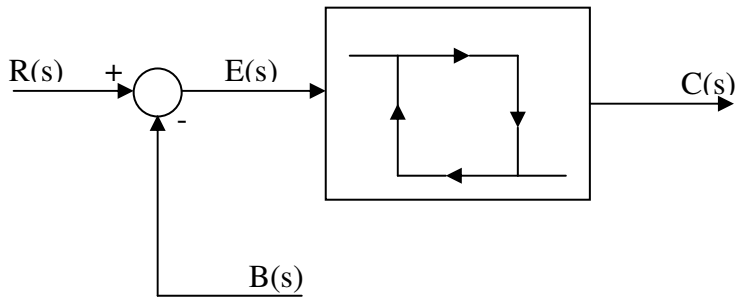


Şekil 4.3. Sistem başlatımı akış şeması

Sistem yazılımının doğru biçimde çalıştığını operatörün gözlemlemesi için yazılımda bir “heartbeat” (aktif çalışma) döngüsü tanımlanmış ve bu döngünün çalıştırdığı bir led kontrol kutusuna eklenmiştir.

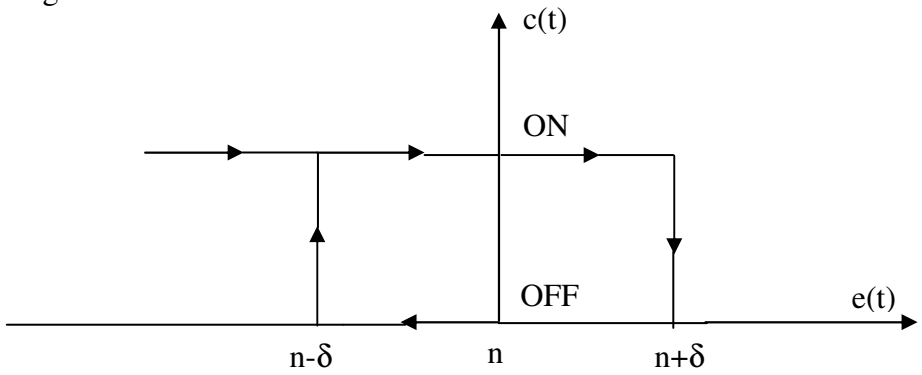
Mikrodenetleyici ve çevre birimlerinin, istenen fonksiyonları yazılımın sonraki bölümlerinde yerine getirebilmeleri için programın başında ilklendirme (initialize) gerçekleştirilmelidir. Sistem başlatımı olarak adlandırılan bu bölüm ayrıca tuş takımı ve LCD ekran için düzenlemeler içermektedir. Belli bir zaman aralığında sıcaklık algılayıcısından alınan bilgi, mikrodenetleyiciye aktarıldıktan sonra LCD ekranda görüntülenir ve tekrar başa dönülür, yukarıdaki işlemler tekrar edilir. Bu adıma ait akış şeması Şekil 4.3 'te yer almaktadır.

4.3. On/Off Denetleyici



Şekil 4.4. On/Off kontrol blok diyagramı

Şekil 4.5'te On/Off kontrol bileşeninin çıkış sinyali c(t)'nin, hata sinyaline göre değişimi görülmektedir.



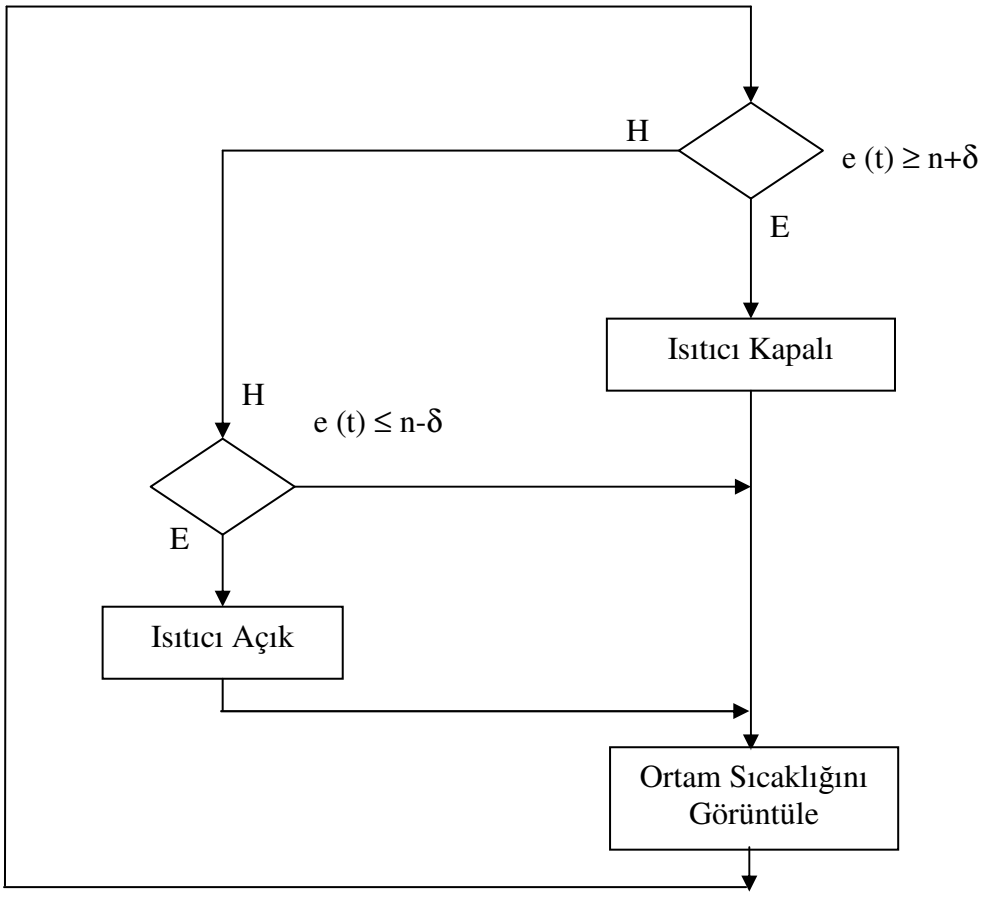
Şekil 4.5. Hata sinyaline göre değişim

Hata sinyali verilen bir "n" değerinin n+δ kadar üstünde iken çıkış kapanmakta, n-δ değerine düştüğünde ise c(t) açılmakta ve maksimum çıkış vermektedir. 2δ değeri diferansiyel aralık olarak adlandırılır.

$e(t) \geq n+\delta \rightarrow c(t)$ kapalı

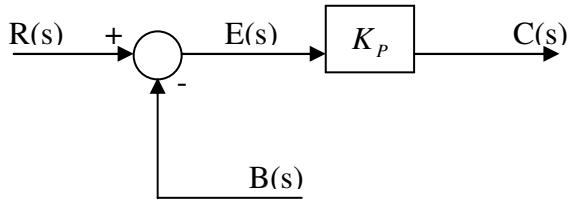
$e(t) \leq n-\delta \rightarrow c(t)$ açık

Diferansiyel aralık minimum 0, maksimum 65 °C olabilmektedir. Bu fonksiyonun akış şeması Şekil 4.6'da yer almaktadır. Programın çalıştırılması esnasında sıcaklık değeri girilir, ON/OFF kontrol seçilir ve diferansiyel aralık girilir.



Şekil 4.6. On/Off kontrol akış şeması

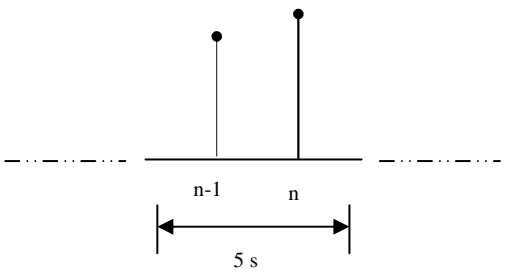
4.4. Oransal (P) Denetleyici



Şekil 4.7. Oransal kontrol blok diyagramı

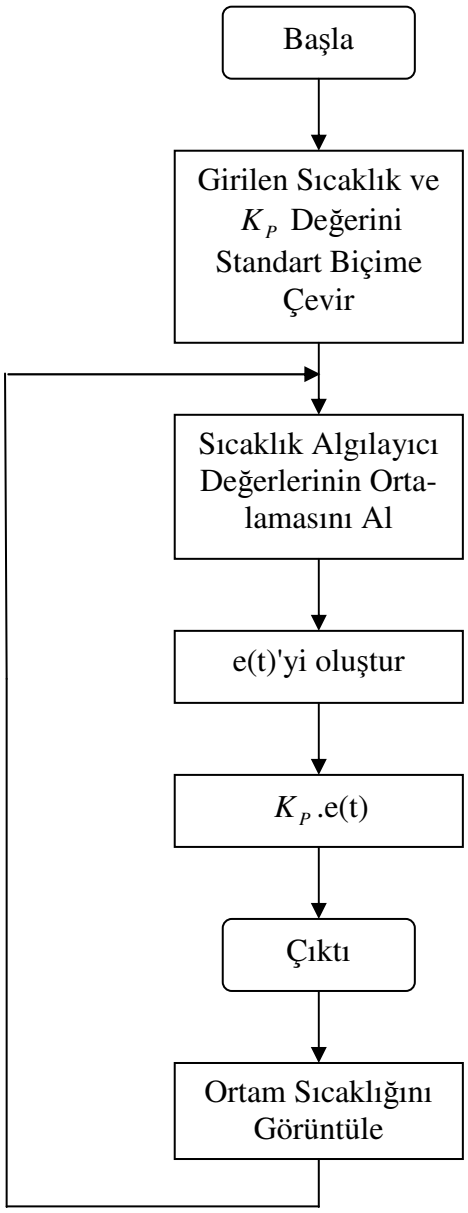
Çıkış c(t) ve hata sinyali arasındaki bağıntı c(t)= K_P. e(t)'dir. Burada K_P kazancı göstermektedir. Şekil 4.8'de oransal denetleyici tipini gerçekleştiren yazılımın akış şeması görülmektedir. Tuş takımından girilen K_P ve sıcaklık algılayıcılardan alınan ortam sıcaklık verileri b(t), aritmetik işlemlerde kullanılacak standart biçime dönüştürülür.

Sistemde, sayısal sıcaklık algılayıcıların konumu gereği yaklaşık 1 s aralıklarla alınan, ortam sıcaklığını gösteren veriler toplanır ve 5. s sonunda aritmetik ortalaması alınır. Böylece b(t) oluşturulur. e(t) =r(t)-b(t) olduğundan, bu işlem yalnızca ikili sayı sisteminde iki sayının çıkarılması işlemidir. Sonuçta e(t) elde edilir. K_P. e(t) oluşturulur ve ısıl sistemi sürecek olan sıcaklık değerine dönüştürülür. Bu dönüştürme işlemi akış şemasında "çıktı" olarak görülmektedir. Akış şemasında ki son blokta eldeki b(t) verisi LCD ekranda görüntülenir. Döngü ile yukarıdaki işlemler tekrar edilir. Tüm işlemler yaklaşık olarak 5 s sürmektedir. Şekil 4.8'de bu süre gösterilmektedir.



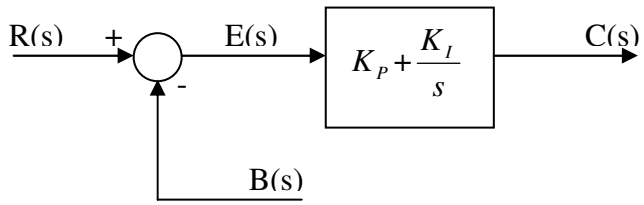
Şekil 4.8. Oransal kontrol örnekleme yaklaşımı

Oransal kontrol programının çalıştırılması esnasında, istenen ortam sıcaklığı (set değer) girilir. Örneğin 33 °C. Kontrol edilebilen sıcaklık aralığı 20 – 85 °C arasındadır. Aralık herbiri 0,25 °C olan toplam 256 parçaya bölünmüştür. "P" ile oransal denetleyicinin seçildiği bildirilir ve istenen K_P değeri girilir. Bundan sonra program işlemeye başlar.



Şekil 4.9. Oransal kontrol akış şeması

4.5. Oransal+İntegral (PI) Denetleyici



Şekil 4.10. Oransal+İntegral kontrol blok diyagramı

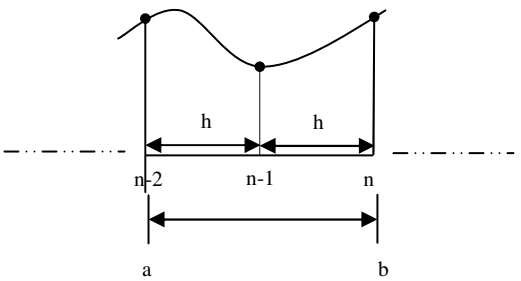
Çıkış c(t) ile hata sinyali e(t) arasındaki eşitlik:

$$m(t) = K_P.e(t) + K_I \int_0^t e(t).dt \qquad [2.1]$$

olarak yer almaktadır. Mikrodenetleyici yazılımı aracılığıyla bu eşitlikteki integrali hesaplamak için sayısal integrasyon yöntemlerinden biri olan yamuk yöntemi kullanılmıştır. Bu yöntemde, 0-t aralığındaki bir fonksiyonun integrali

$$\int_a^b f(t).dt \cong \frac{h}{2} \sum_{n=1}^{N} \left[f(a+(n-1)h) + f(a+n.h) \right] \qquad [2.2]$$

eşitliği ile elde edilir (Vatansever, 2002). Burada "n" n. örneklemeyi, h iki örnekleme arası zamanı, N a ile b arasındaki toplam parça sayısını göstermektedir. Şekil 4.11'de formüldeki değerler gösterilmektedir.



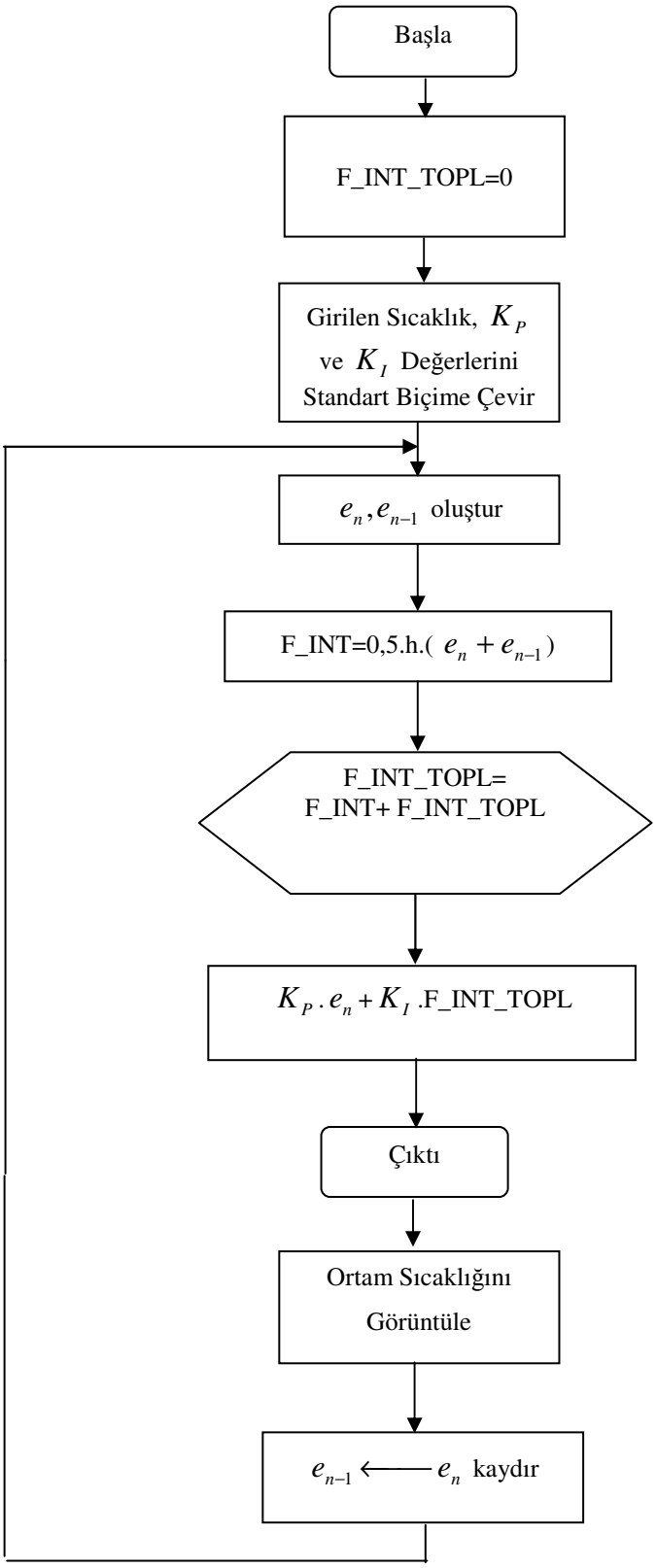
Şekil 4.11. Oransal+İntegral kontrol örnekleme yaklaşımı

Şekil 4.10'daki akış şemasındaki ilk işlemler tuş takımından girilen K_P ve K_I katsayılarının işlenebilmeleri için standart şekle getirilmeleridir. İlk üç örneklemeden sonra her bir örnekleme için hata sinyali oluşturulur (e_{n-2}, e_{n-1}, e_n). Bu değerler daha sonra Eş. 2.2'de yerine konularak:

$$K_I \int_0^t e(t).dt \cong K_I . \frac{h}{2} . \sum_{n=1}^{N} \left[e(0+(n-1)h) + e(0+n).h) \right] \qquad [2.3]$$

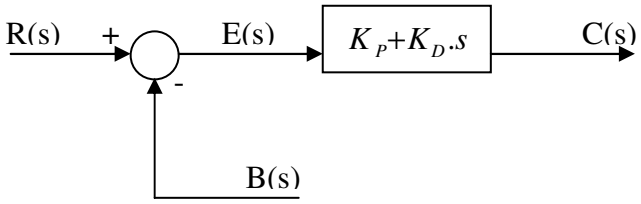
eşitliği elde edilir. En son örnekleme e_n, K_P ile çarpılarak oransal terim oluşturulur, Eş. 2.3'te bulunan integral terimi ile toplanır ve sonuç ifade olan m(t) bulunur. "Çıktı" olarak gösterilen blokta m(t) ile sistemi süren ısı arasındaki bağıntı sağlanır. Sistemdeki ölü zaman gecikmesinden ve yavaş tepkiden dolayı h 20 s olarak alınmakta ve integral alınmaya set değer ile ortam sıcaklığı arasında 10 °C kala başlanmaktadır. Bu değerler, Simulink benzetim sonuçları ve deneysel çalışma sonuçları karşılaştırmalı değerlendirilerek saptanmıştır. Oransal+İntegral Denetleyicinin çalıştırılması:

- İstenen ortam sıcaklığı 30-60 °C arası girilir.
- PI ile bu kontrol tipinin seçildiği belirtilir.
- K_P ve K_I girilir.



Şekil 4.12. Oransal+İntegral kontrol akış şeması

4.6. Oransal+Türevsel (PD) Denetleyici



Şekil 4.13. Oransal+Türevsel kontrol blok diyagramı

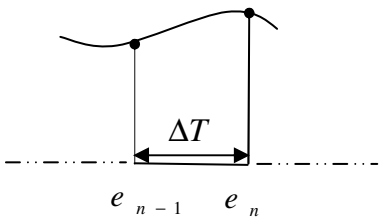
Çıkış c(t) ile hata sinyali e(t) arasındaki eşitlik:

$$m(t) = K_P.e(t) + K_D.\frac{de(t)}{dt} \qquad [2.4]$$

de(t)/dt türevi yeterince küçük ΔT zaman aralığı için:

$$\frac{de(t)}{dt} \cong \frac{e_n - e_{n-1}}{\Delta T} \qquad [2.5]$$

olarak ifade edilebilir. Sistem zaman sabitine, sıcaklık algılayıcı örnekleme düzeneğine ve deneysel sonuçlara göre $\Delta T = 1$ s olarak alınmıştır. Şekil 4.14'te örnekleme işlemi görülmektedir.

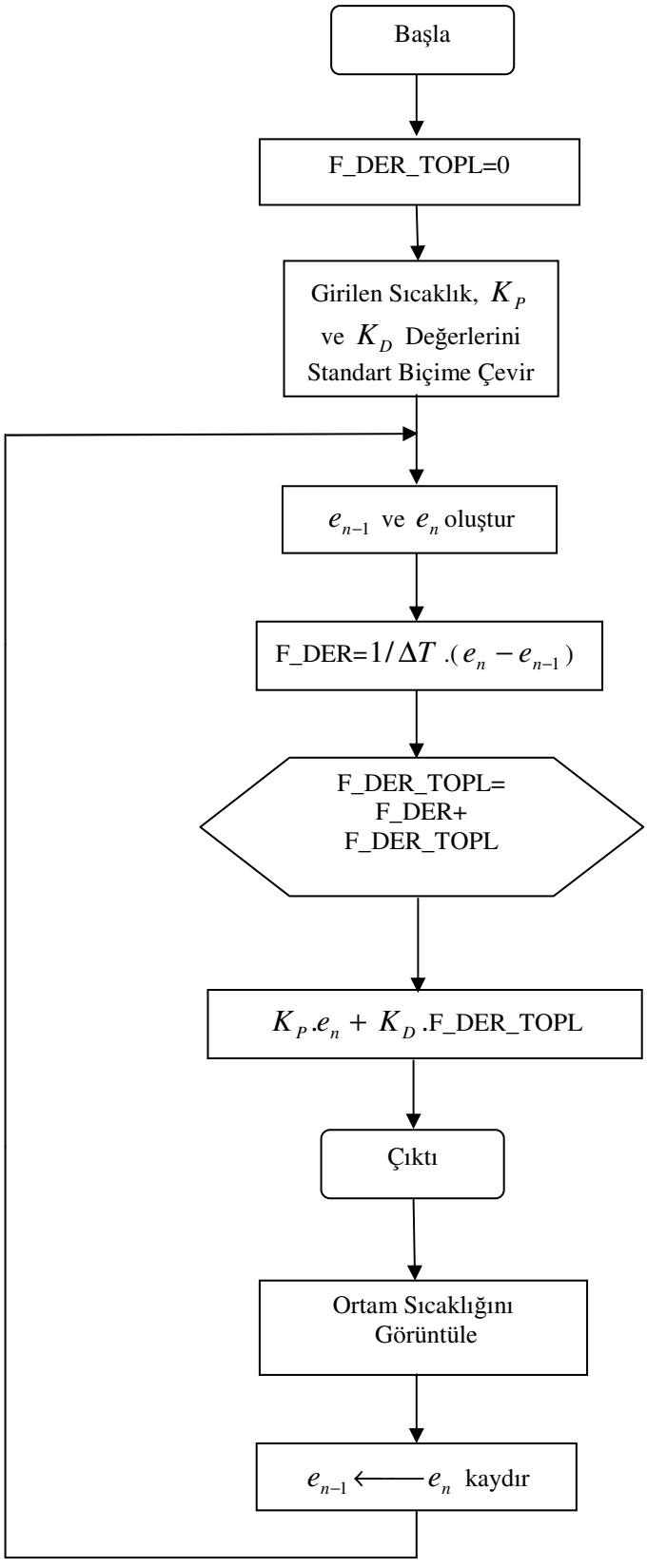


Şekil 4.14. Oransal+Türevsel kontrol örnekleme yaklaşımı

Şekil 4.15'de görülen akış şemasında önceki denetleyici tiplerinde olduğu gibi girilen K_P ve K_D katsayıları standart biçime dönüştürülür.

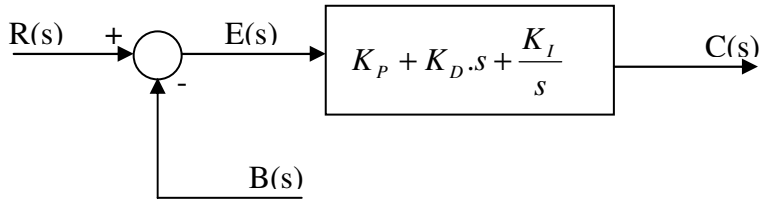
5 sn aralıklarla iki örnekleme alınarak ana döngüye girilir. e_n, e_{n-1} hata sinyali değerleri oluşturulur. $e_n - e_{n-1}$ oluşturularak türevin ilk aşaması gerçekleştirilir, l/ ΔT ile türevin gerçek değeri bulunur. $K_D.1/\Delta T.(e_n - e_{n-1})$ ile türevsel terim oluşturulur. En son örneklemenin hata sinyali değeri e_n, K_P ile çarpılarak oransal terim oluşturulur. Bu iki terim toplanarak oransal+türevsel denetleyicinin çıkış c(t) değeri oluşturulur. Akış semasındaki "çıktı" olarak gösterilen blokta c(t) ile sistemi süren ısı arasındaki eşitlik sağlanır. Oransal+Türevsel kontrolün çalıştırılması:

- İstenen sıcaklık değeri girilir
- PD ile bu denetleyici tipi seçilir
- K_P ve K_D değerleri girilir.



Şekil 4.15. Oransal+Türevsel kontrol akış şeması

4.7. Oransal+İntegral+Türevsel (PID) Denetleyici



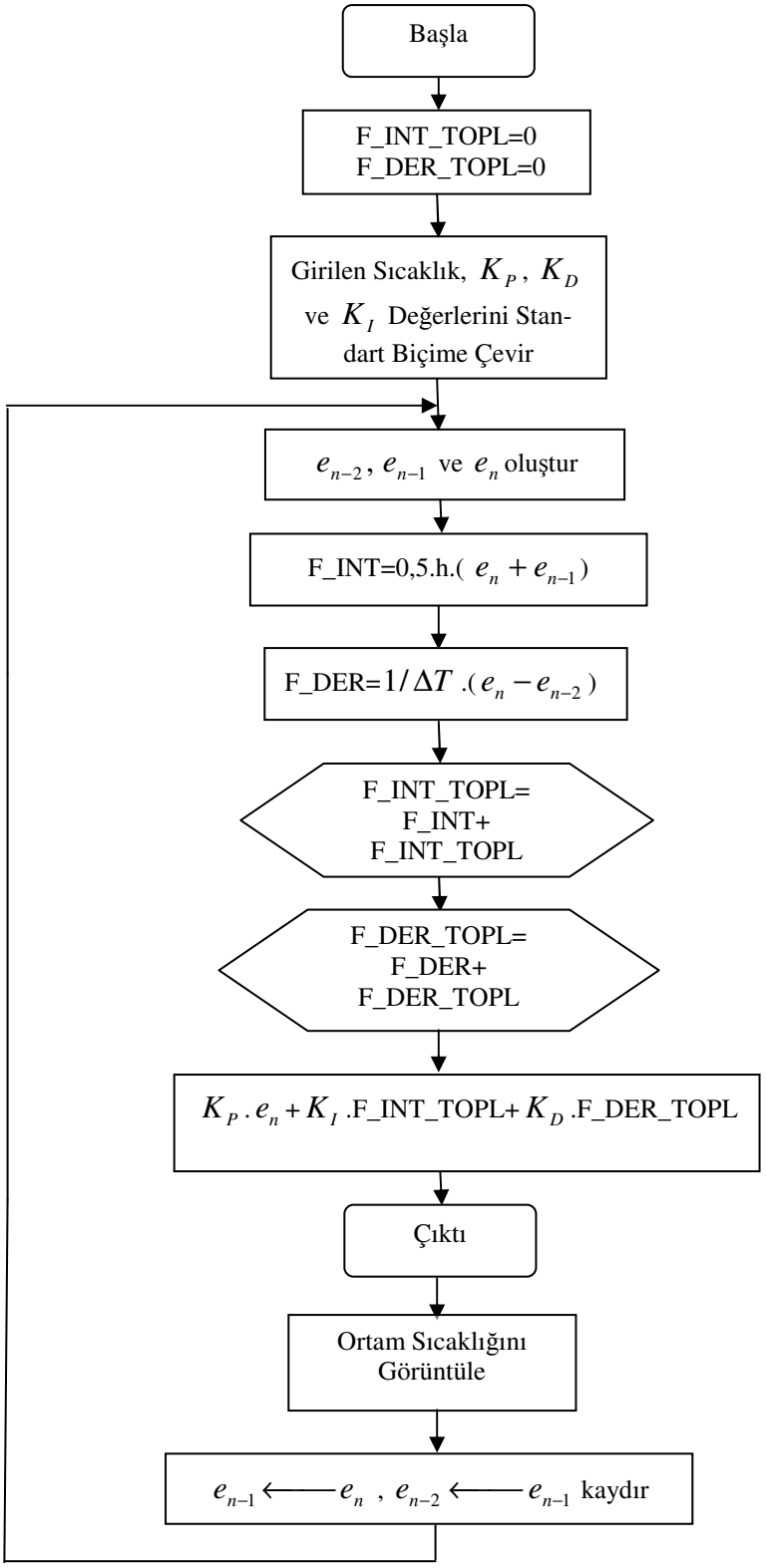
Şekil 4.16. Oransal+İntegral+Türevsel kontrol blok diyagramı

Çıkış c(t) ile hata sinyali e(t) arasındaki eşitlik:

$$c(t) = K_P.e(t) + K_D.\frac{de(t)}{dt} + K_I\int_0^t e(t).dt \qquad [2.6]$$

Eşitlikteki integral, türev ve oransal bölümler önceki bölümlerde anlatıldığı şekilde gerçekleştirilmektedir. Sadece türev alınırken e_{n-1} ve e_{n-2} kullanılır. Şekil 4.14'teki akış şemasından da görüleceği üzere program daha önce anlatılan denetleyici tiplerinin bir birleşimidir. Programın çalıştırılması:

- 30-60°C arasında sıcaklık değeri girilir.
- PID ile bu denetleyici tipi seçilir.
- K_P, K_D ve K_I değerleri girilir.



Şekil 4.17. Oransal+İntegral+Türevsel kontrol akış şeması

4.8. Mikrodenetleyici Yazılımı

PIC 16F877 mikrodenetleyici için gömülü yazılım geliştirmede çeşitli araçlar mevcuttur. Üst seviye programlama dillerinden PASCAL, BASIC gibi dillerin standart söz dizimi ve komutlarını PIC mikrodenetleyici için derleyen pek çok ticari ve ücretsiz yazılım geliştirme aracı ile birlikte aynı biçimde orta seviye C (İbrahim, 2003; Gardner, 1998) programlama dili üzerinde geliştirilen kodun mikrodenetleyiciye aktarılmasında kullanılabilecek derleyiciler de bulunabilmektedir. Bütün bu derleyiciler, .c ya da .bas (13) gibi standart programlama dosyalarını çapraz çevirerek (cross compile) mikrodenetleyici- ye aktarılabilecek onaltılı sayı sisteminde karşılıklarına dönüştürürler. .hex uzantılı bu veri, kişisel bilgisayar üzerinden seri ya da usb port aracılığıyla mikrodenetle-yicinin bağlı olduğu programlayıcıya iletilir (16) ve PIC programlanır. FLASH program bellekleri sayesinde PIC mikrodenetleyiciler defalarca programlanıp silinebilir.

Bu çalışmada yazılım geliştirme aracı olarak Crownhill Associates firmasının Proton PICBASIC derleyicisi ve yazılım geliştirme ortamı kullanılmıştır.

4.8.1. Program ve açıklaması

```
'------------------PIC16F877, 4 MHz Kristal, Gözleyici Devre Kapalı--------------------


DEVICE = 16F877
XTAL = 4
CONFIG XT_OSC , WDT_OFF , PWRTE_OFF , BODEN_OFF , LVP_OFF ,
WRTE_ON

'---------------------------------------I/O Tanımlamaları-------------------------------------

Symbol butondown    = PORTC.0    'Buton kontrol port birimleri
Symbol butonset     = PORTC.1
Symbol butonup      = PORTC.3

Symbol role1 = PORTB.0           'Röle kontrol port birimleri
Symbol role2 = PORTC.5
Symbol role3 = PORTC.6
```

```
Symbol role4 = PORTC.4
Symbol role5 = PORTC.7

Symbol DAC = PORTD              'DAC port birimi

Symbol led_4 = PORTA.0          'Heartbeat (Aktif Çalışma) Ledi port birimi

Symbol DQ = PORTB.2             ' Sıcaklık Algılayıcılar ile iletişimin yapılacağı
                                port birimi


TRISA.0=0
TRISB.0=0
TRISC = %00001011
TRISD = %00000000

'---------------------------------GENEL DEĞİŞKENLER--------------------------------------

Dim var1 as byte, durum as byte, DERECE_SET as byte, ekran as byte, kontroltipi
as byte, parametresec as byte, i as byte, KONTROLSTART as byte, C as byte

Dim var2 as word, KP_VALUE as word, KI_VALUE as word, KD_VALUE as
word, DA_VALUE as word, FARK as word, DUTY as word

DIM PIK AS WORD                 'Sayaç amaçlı değişken
DIM DEGISIM AS WORD             'flag

Dim ONOFFSAYAC as byte
Dim DERECE_UST as byte
Dim DERECE_ALT as byte


'--------------------------- SICAKLIK OKUMA DEĞİŞKENLERİ -------------------------


DIM KUSUR AS WORD               'Sıcaklığın virgülden sonraki kısmı
DIM TEMP AS WORD                'Sıcaklığı tutan değişken
DIM TEMP_TAM AS WORD
DIM TEMPTOPLA AS WORD
DIM TEMPORT AS WORD
DIM FSAYAC AS BYTE
DIM F1 AS WORD
DIM F2 AS WORD
DIM F3 AS WORD
DIM F_0 AS WORD
DIM EKSI AS BYTE
DIM FARK_E AS WORD
```

```
DIM FARK_D AS WORD
DIM F_TRV AS WORD , F_INT AS WORD, F_DER_TOPL AS WORD,
F_INT_TOPL AS WORD
DIM ED1 AS BYTE, ED2 AS BYTE, F_DER AS WORD

'----------------------------DEĞİŞKENLERİN İLK DEĞERLERİ----------------------------

initial:

durum=255
DERECE_SET=36
kontroltipi=1
var1=0
var2=0
KONTROLSTART=0
FSAYAC=0
FARK_E=0
F_INT=0
F1=20
KP_VALUE=50
KI_VALUE=3
KD_VALUE=3
DA_VALUE=0
ONOFFSAYAC=0
EKSI=0
F_DER=0
F_0=0
ED1=25

'-------------------------------------PWM BAŞLATMA-------------------------------------

Ccp1_pin = Portc.2
TRISC.2 = 0                    'PORTC.2 (CCP1) çıkış yapıldı
CCP1CON = %00001100            'CCP1 yazmacı PWM'e ayarlandı
T2CON = %00000100              'Timer2 aktif edildi, Prescale=4
PR2 = 255                      'PR2 1KHz çıkışa ayarlandı

Duty = 0 : Ccp1con.4 = Duty.0 : Ccp1con.5 = Duty.1 : Ccpr1l = Duty >> 2


'-------------------------------LCD EKRAN BAŞLATMA-------------------------------

DECLARE LCD_TYPE 0
DECLARE LCD_DTPIN PORTB.4          '4 bit iletişim arabirimi
DECLARE LCD_ENPIN PORTB.3
DECLARE LCD_RSPIN PORTB.1
DECLARE LCD_INTERFACE 4
```

```
DECLARE LCD_LINES 2
        PRINT $FE , 1 : DELAYMS 30
        PRINT $FE , 2
   PRINT $FE , $0C

'----------------------------------TIMER 0 BAŞLATMA--------------------------------------

OPTION_REG=$55                  'Prescaler (Bölme Değeri) =1/64
        INTCON=$a0
        ON INTERRUPT GOTO PIKINT
        ekran=1
        gosub lcdcikis
        delayms 1000



                '-------------------Timer0 alt programı : 16384us ---------------------------

        DISABLE
        PIKINT:
                        PIK=PIK+1
                        IF PIK<61 THEN goto PIEXIT
                        PIK=0
                        DEGISIM=1
        PIEXIT:
        INTCON.2=0
        RESUME

'----------------------AKTİF ÇALIŞMA LEDİ (Heartbeat) Döngüsü----------------------

heartbeat:
        led_4=1
        delayms 500
        led_4=0
        delayms 500
return


'-----------------ALGILAYICIDAN SICAKLIK OKUMA DÖNGÜSÜ------------------

sicaklikoku:

DELAYMS 500
OWRITE DQ, 1, [$CC, $44]        ' Sıcaklığı hesaplaması için algılayıcıya komut
                                gönderme
REPEAT
DELAYMS 25                      ' İşlem bitene kadar bekleme
```

```
OREAD DQ, 4, [C]                ' İşlem süresince portb.2=0 olur
UNTIL C <> 0                    ' Okuma işlemi tamam
OWRITE DQ, 1, [$CC, $BE]        ' Sıcaklık değerini RAM dan okuması için ko-
                                  mut gönderiyor

OREAD DQ, 2,[TEMP.LOWBYTE,TEMP.HIGHBYTE]
                                'Sıcaklık değeri algılayıcıdan okunuyor

   durum=255
        TEMP_TAM = TEMP/16
return

'--------------------RÖLE İLE ALGILAYICI BELİRLEME DÖNGÜSÜ-----------------


sicaklikdongusu:

if degisim=1 then
    degisim=0
 TEMPTOPLA = 0
 role2=0 : role3=0 : role4=0 : role5=0 : delayms 5 : role1=1        '1.algılayıcı seçildi
                    gosub sicaklikoku
                    TEMPTOPLA =TEMP_TAM
       if butonset=0 then goto cks101
 role1=0 : role3=0 : role4=0 : role5=0 : delayms 5 : role2=1        '2.algılayıcı seçildi
                    gosub sicaklikoku
                    TEMPTOPLA = TEMPTOPLA + TEMP_TAM
       if butonset=0 then goto cks101
 role2=0 : role1=0 : role4=0 : role5=0 : delayms 5 : role3=1        '3.algılayıcı seçildi
                    gosub sicaklikoku
                    TEMPTOPLA = TEMPTOPLA + TEMP_TAM
       if butonset=0 then goto cks101
 role2=0 : role3=0 : role1=0 : role5=0 : delayms 5 : role4=1        '4.algılayıcı seçildi
                    gosub sicaklikoku
                    TEMPTOPLA = TEMPTOPLA + TEMP_TAM
       if butonset=0 then goto cks101
 role2=0 : role3=0 : role4=0 : role1=0 : delayms 5 : role5=1        '5.algılayıcı seçildi
                    gosub sicaklikoku
                    TEMPTOPLA = TEMPTOPLA + TEMP_TAM
                    if butonset=0 then  goto cks101
                    TEMPORT=TEMPTOPLA/5

                    PRINT $FE , 1 : DELAYMS 30
               PRINT $FE , 2 : PRINT "TEMP:+"
               PRINT DEC TEMPORT            ' SICAKLIĞI EKRANA YAZDIR
                    PRINT " C"
                    DAC = TEMPORT
```

```
cks101:
            end if
return

'-----------------------------BUTON KONTROL DÖNGÜSÜ------------------------------

butonkontrol:

                    '------------------- buton down döngüsü-----------------------
  if butondown=0 then
BEKLE1:
  if butondown=0 then GOTO BEKLE1
    if durum=1 then
                    DERECE_SET=DERECE_SET-1
                    if DERECE_SET<30 then DERECE_SET= 30
       ekran=3
                    gosub lcdcikis
          elseif durum=2 then
                    kontroltipi=kontroltipi-1
                    if kontroltipi<1 then kontroltipi= 1
                    ekran=5
                    gosub lcdcikis
          elseif durum=3 then
            parametresec=parametresec+1
                    if parametresec>4 then parametresec=1
                    ekran=7
                    gosub lcdcikis
          elseif durum=4 then
            durum=3
                    ekran=8
                    gosub lcdcikis
          end if
  end if

                    '-------------------------buton set döngüsü-----------------------

  if butonset=0 then
BEKLE3:
  if butonSET=0 then GOTO BEKLE3
          if durum=255 then
                    durum=1
                    ekran=2
                    gosub lcdcikis
                    goto cikis1
          end if
          if durum=1 then
            durum=2
```

```
            ekran=4
                gosub lcdcikis
                goto cikis1
        end if
        if durum=2 then
                durum=3
                ekran=6
                gosub lcdcikis
                goto cikis1
        end if
        if durum=3 then
                durum=4
                ekran=8
                gosub lcdcikis
                KONTROLSTART=1
                goto cikis1
        end if
cikis1:
  end if

                '--------------------buton up döngüsü-------------------------

  if butonup=0 then
BEKLE2:
  if butonUP=0 then GOTO BEKLE2
          if durum=1 then
                DERECE_SET=DERECE_SET+1
                if DERECE_SET>75 then DERECE_SET = 75
      ekran=3
                gosub lcdcikis
          elseif durum=2 then
                kontroltipi=kontroltipi+1
                if kontroltipi>5 then kontroltipi= 5
                ekran=5
                gosub lcdcikis
          elseif durum=3 then
            if parametresec=1 then
                  KP_VALUE=KP_VALUE+10
                  IF KP_VALUE>100 THEN KP_VALUE=10
            elseif parametresec=2 then
                  KI_VALUE=KI_VALUE+1
                  IF KI_VALUE>10 THEN KI_VALUE=1
            elseif parametresec=3 then
                  KD_VALUE=KD_VALUE+1
                  IF KD_VALUE>10 THEN KD_VALUE=1
                elseif parametresec=4 then
                  DA_VALUE=DA_VALUE+1
```

```
                    IF DA_VALUE>5 THEN DA_VALUE=0
              end if
                    ekran=7
                    gosub lcdcikis
            end if
  end if
return


'-----------SONUÇLARI LCD EKRANDA GÖRÜNTÜLEME DÖNGÜSÜ------------

lcdcikis:

if ekran=1 then
            PRINT $FE , 1 : DELAYMS 30
            PRINT $FE , 2 : PRINT "SICAKLIK KONTROL"
            PRINT $FE , $C0  : PRINT " MURAT 6PARMAK"
elseif ekran=2 then
            PRINT $FE , 1 : DELAYMS 30
            PRINT $FE , 2 : PRINT "  SET DEGERINI"
            PRINT $FE , $C0  : PRINT "-<  GIRINIZ   >+"
elseif ekran=3 then
            PRINT $FE , 1 : DELAYMS 30
            PRINT $FE , 2 : PRINT "    SICAKLIK"
            PRINT $FE , $C0  : PRINT "       ",DEC DERECE_SET
elseif ekran=4 then
            PRINT $FE , 1 : DELAYMS 30
            PRINT $FE , 2 : PRINT "    SICAKLIK"
            PRINT $FE , $C0  : PRINT " SET EDILMISTIR"
            delayms 200
            PRINT $FE , 1 : DELAYMS 30
            PRINT $FE , 2 : PRINT " KONTROL TIPINI"
            PRINT $FE , $C0  : PRINT "    GIRINIZ"
elseif ekran=5 then
     PRINT $FE , 1 : DELAYMS 30
     if kontroltipi=1 then
                    PRINT $FE , 2 : PRINT "   ON/OFF"
            elseif kontroltipi=2 then
                    PRINT $FE , 2 : PRINT "   P"
            elseif kontroltipi=3 then
                    PRINT $FE , 2 : PRINT "   PI"
            elseif kontroltipi=4 then
                    PRINT $FE , 2 : PRINT "   PD"
            elseif kontroltipi=5 then
                    PRINT $FE , 2 : PRINT "   PID"
            end if
elseif ekran=6 then
```

```
        PRINT $FE , 1 : DELAYMS 30
        PRINT $FE , 2 : PRINT " KONTROL TIPI"
        PRINT $FE , $C0  : PRINT " SET EDILMISTIR"
        delayms 200
        PRINT $FE , 1 : DELAYMS 30
        PRINT $FE , 2 : PRINT " PARAMETRELERI"
        PRINT $FE , $C0  : PRINT "S<  GIRINIZ   >+"
elseif ekran=7 then
        if parametresec=1 then
                PRINT $FE , 1 : DELAYMS 30
                PRINT $FE , 2 : PRINT "  KP : "
                PRINT DEC KP_VALUE
        elseif parametresec=2 then
                PRINT $FE , 1 : DELAYMS 30
                PRINT $FE , 2 : PRINT "  KI : "
                IF KI_VALUE = 1 THEN
                  PRINT "0,1"
                ELSEIF KI_VALUE = 2 THEN
                  PRINT "0,2"
                ELSEIF KI_VALUE = 3 THEN
                  PRINT "0,3"
                ELSEIF KI_VALUE = 4 THEN
                  PRINT "0,4"
                ELSEIF KI_VALUE = 5 THEN
                  PRINT "0,5"
                ELSEIF KI_VALUE = 6 THEN
                  PRINT "0,6"
                ELSEIF KI_VALUE = 7 THEN
                  PRINT "0,7"
                ELSEIF KI_VALUE = 8 THEN
                  PRINT "0,8"
                ELSEIF KI_VALUE = 9 THEN
                  PRINT "0,9"
                ELSEIF KI_VALUE = 10 THEN
                  PRINT "1"
                END IF
        elseif parametresec=3 then
                PRINT $FE , 1 : DELAYMS 30
                PRINT $FE , 2 : PRINT "  KD : "
                PRINT DEC KD_VALUE
        elseif parametresec=4 then
                PRINT $FE , 1 : DELAYMS 30
                PRINT $FE , 2 : PRINT "  DA : "
                PRINT DEC DA_VALUE
        end if
elseif ekran=8 then
        PRINT $FE , 1 : DELAYMS 30
```

```
            PRINT $FE , 2 : PRINT "  PARAMETRELER"
            PRINT $FE , $C0  : PRINT " SET EDILMISTIR"
            DELAYMS 1000
            PRINT $FE , 1 : DELAYMS 30
            PRINT $FE , 2 : PRINT "    KONTROL"
            PRINT $FE , $C0  : PRINT "  BASLAMISTIR"
end if
return

'---------------------------------ANA PROGRAM DÖNGÜSÜ-------------------------------

DONGU:

   IF KONTROLSTART = 1 THEN
            GOSUB KONTROL
                  IF KONTROLSTART = 0 THEN
                        GOTO CKS100
                  END IF
                  gosub sicaklikdongusu
       PRINT $FE , $C0 ' :PRINT "KONTROL BASLADI"
                  GOTO DONGU
         END IF
CKS100:
         IF DURUM=255 THEN
           gosub sicaklikdongusu
         END IF

   gosub butonkontrol

GOTO DONGU



'-------------------------------DENETLEYİCİ DÖNGÜLERİ-----------------------------


KONTROL:

'---------------------------------On/Off Kontrol-----------------------------------------------

if kontroltipi=1 then

  DERECE_UST=DERECE_SET+DA_VALUE
  DERECE_UST = DERECE_UST - 1
  DERECE_ALT=DERECE_SET-DA_VALUE
  IF TEMPORT > DERECE_UST THEN
     ONOFFSAYAC=ONOFFSAYAC+1
     Duty = 0 : Ccp1con.4 = Duty.0 : Ccp1con.5 = Duty.1 : Ccpr1l = Duty >> 2
```

```
            GOTO KONTROL_CIK
   ELSEIF TEMPORT < DERECE_ALT THEN
      Duty = 950 : Ccp1con.4 = Duty.0 : Ccp1con.5 = Duty.1 : Ccpr1l = Duty >> 2
            GOTO KONTROL_CIK
   END IF

'----------------------------------------Oransal (P) Kontrol-------------------------------

elseif kontroltipi=2 then

      IF DERECE_SET >= TEMPORT THEN
      FARK=DERECE_SET-TEMPORT
            DUTY = KP_VALUE*FARK
            ELSEIF  DERECE_SET < TEMPORT THEN
            DUTY = 0
            ENDIF

            IF DUTY > 950 THEN
                  DUTY = 950
            END IF
            Ccp1con.4 = Duty.0 : Ccp1con.5 = Duty.1 : Ccpr1l = Duty >> 2
            GOTO KONTROL_CIK

'---------------------------------------Oransal+İntegral(PI) Kontrol--------------------------


elseif kontroltipi=3 then

KONTROL_PI:
    IF DERECE_SET >= TEMPORT THEN
                  FARK_E=DERECE_SET-TEMPORT
       IF FARK_E<10 THEN
          F_INT=FARK_E+F_INT
                  END IF
          END IF
    IF DERECE_SET < TEMPORT THEN
           FARK_E=TEMPORT-DERECE_SET
                  IF F_INT>FARK_E THEN
          F_INT=F_INT-FARK_E
                  ELSE
                    F_INT=0
                  END IF
          END IF
          F_INT_TOPL=0
          FOR i = 0 TO KI_VALUE
                   F_INT_TOPL= F_INT + F_INT_TOPL
          NEXT
```

```
                PRINT DEC F_INT_TOPL
                PRINT " "
          IF DERECE_SET >= TEMPORT THEN
      FARK = DERECE_SET-TEMPORT
           DUTY = KP_VALUE*FARK
          ELSEIF  DERECE_SET < TEMPORT THEN
           DUTY = 0
          ENDIF

          DUTY = DUTY + F_INT_TOPL

           IF DUTY > 950 THEN
                DUTY = 950
           END IF

           PRINT DEC DUTY
           Ccp1con.4 = Duty.0 : Ccp1con.5 = Duty.1 : Ccpr1l = Duty >> 2
          GOTO KONTROL_CIK

PI_END:

'-----------------------------------Oransal+Türevsel (PD) Kontrol----------------------------

elseif kontroltipi=4 then

KONTROL_PD:

          IF DERECE_SET >= TEMPORT THEN
      FARK = DERECE_SET-TEMPORT
           DUTY = KP_VALUE*FARK
          ELSEIF  DERECE_SET < TEMPORT THEN
           DUTY = 0
          ENDIF

          ED2=TEMPORT
          IF ED2>=ED1 THEN
            FARK_D=ED2-ED1
                F_DER_TOPL=0
                FOR i = 0 TO KD_VALUE
                 F_DER_TOPL= F_DER + F_DER_TOPL
            NEXT
                DUTY = DUTY + F_DER_TOPL
          ELSEIF ED1>ED2 THEN
            FARK_D=ED1-ED2
                F_DER_TOPL= 0
                FOR i = 0 TO KD_VALUE
                 F_DER_TOPL= F_DER + F_DER_TOPL
```

```
            NEXT
                DUTY = DUTY - F_DER_TOPL
            END IF
            ED1=ED2

            IF DUTY > 950 THEN
                DUTY = 950
            END IF
            PRINT DEC DUTY
            Ccp1con.4 = Duty.0 : Ccp1con.5 = Duty.1 : Ccpr1l = Duty >> 2
            GOTO KONTROL_CIK

'-------------------------Oransal-İntegral+Türevsel (PID) Kontrol-------------------------

elseif kontroltipi=5 then
 PID_KONTROL:

            IF DERECE_SET >= TEMPORT THEN
                FARK_E=DERECE_SET-TEMPORT
        IF FARK_E<10 THEN
          F_INT=FARK_E+F_INT
                END IF
            END IF
     IF DERECE_SET < TEMPORT THEN
            FARK_E=TEMPORT-DERECE_SET
                IF F_INT>FARK_E THEN
          F_INT=F_INT-FARK_E
                ELSE
                  F_INT=0
                END IF
            END IF
            F_INT_TOPL=0
            FOR i = 0 TO KI_VALUE
                F_INT_TOPL= F_INT + F_INT_TOPL
            NEXT


    IF DERECE_SET >= TEMPORT THEN
      FARK = DERECE_SET-TEMPORT
            DUTY = KP_VALUE*FARK
          ELSEIF  DERECE_SET < TEMPORT THEN
            DUTY = 0
          ENDIF

          DUTY = DUTY + F_INT_TOPL
```

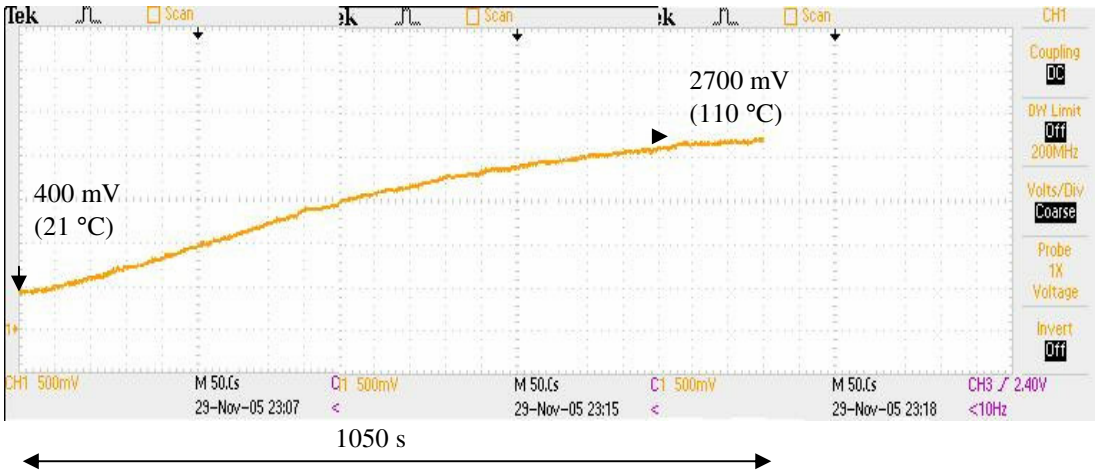
```
        ED2=TEMPORT
        IF ED2>=ED1 THEN
          FARK_D=ED2-ED1
                F_DER_TOPL=0
                FOR i = 0 TO KD_VALUE
                 F_DER_TOPL= F_DER + F_DER_TOPL
          NEXT
                DUTY = DUTY + F_DER_TOPL
        ELSEIF ED1>ED2 THEN
          FARK_D=ED1-ED2
                F_DER_TOPL=0
                FOR i = 0 TO KD_VALUE
                 F_DER_TOPL= F_DER + F_DER_TOPL
          NEXT
                DUTY = DUTY - F_DER_TOPL
        END IF
        ED1=ED2

          IF DUTY > 950 THEN
                 DUTY = 950
          END IF

          PRINT DEC DUTY
          Ccp1con.4 = Duty.0 : Ccp1con.5 = Duty.1 : Ccpr1l = Duty >> 2
        GOTO KONTROL_CIK
END IF
KONTROL_CIK:



RETURN
```

5. SONUÇ ve ÖNERİLER

Model bir süreç üzerinde gerçekleştirilen sıcaklık kontrol sistemi, 20 ºC - 85 ºC sıcaklık aralığını 0,25 ºC hata ile kontrol edebilmektedir. Deneysel çalışmanın sonuçları osiloskop aracılığıyla elde edilmiştir. Osiloskobun Time/Div skalası maksimum 50 s'ye ayarlanabilmekte ve her ekran görüntüsü toplamda 450 s sürmektedir. Tesisin tepkisindeki ölü zaman gecikmesi yaklaşık olarak 100 s civarındadır ve toplamda zaman tepkisinin oldukça yavaş olmasından dolayı ard arda üç kez osiloskop ekran görüntüsü alınmış ve bu veriler bir resim işleme programı yardımıyla birleştirilmiştir.

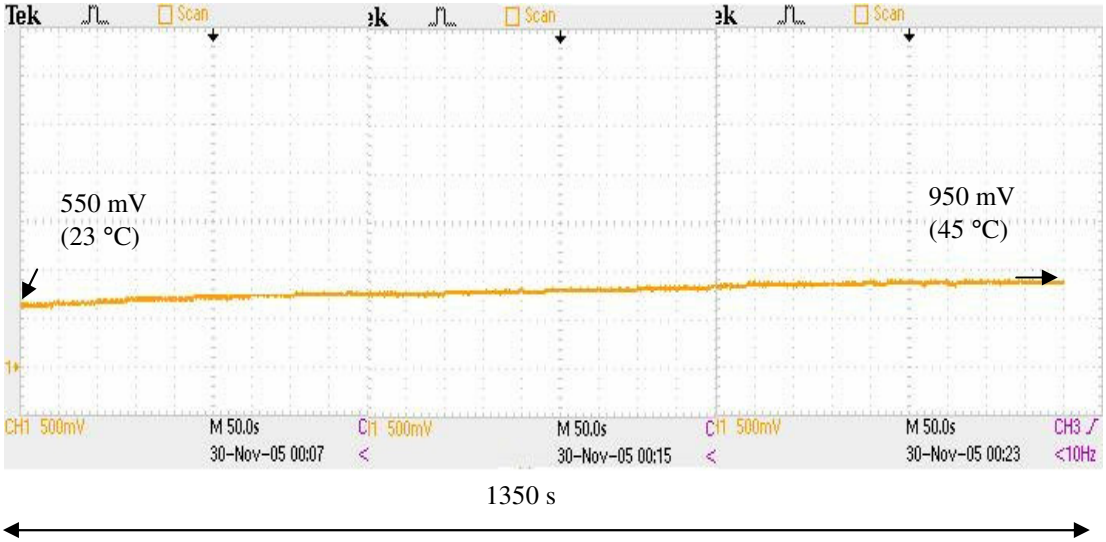
Karar organı olarak mikrodenetleyicinin kullanıldığı sistem On/Off, Oransal, Oransal+İntegral, Oransal+Türevsel ve Oransal+İntegral+Türevsel denetleyici tiplerini yazılım ile gerçekleştirmektedir. Ayrıca her kontrol türünün gerektirdiği ayarlanan parametrelerin tuş takımı ile isteğe göre seçimi sağlanmıştır. Böylece seçilen kontrol türündeki çeşitli parametrelerin (diferansiyel aralık, K_P, K_D ve K_I) değişiminin sistem tepkisi üzerindeki etkileri incelenebilmektedir.

Sistemin açık döngü birim basamak tepkisi (open loop unit step response), ısıtıcıya doğrultucu çıkışı etkin değerde yaklaşık 200 V gerilim verilmişken elde edilen tepkidir.



Şekil 5.1. Sistemin açık döngü birim basamak tepkisi

Sıcaklık algılayıcısının karakteristiği deneysel olarak 20 mV/°C olarak gözlenmiştir. Düşey eksen 500 mV/div, yatay eksen ise 50 sn/div ölçeğinde osiloskoptan çıktı elde edilmiştir. Sistem 23 °C ortam sıcaklığında çalıştırılmaya başlanmış ve 110 °C civarında hacimsel özellikler de göz önünde bulundurularak kararlı duruma ulaştığı gözlenmiştir. Bu süre toplamda 1050 s civarındadır. Sistemin zaman sabiti, bu sürenin % 60'ına karşılık geldiğinden 540 s olarak kabul edilmiştir. Benzer biçimde sıcaklığın gerilimle orantısının deneysel olarak gözlenmesinde sisteme varyak aracılığıyla yaklaşık olarak 100 V gerilim uygulanmış ve kararlı durumdaki tepkisi Şekil 5.2'de çizdirilmiştir.



Şekil 5.2. Isıtıcıya 100 V giriş gerilimi uygulanmasıyla elde edilen çıkış karakteristiği

Şekil 5.2'den görüldüğü üzere ısıtıcının giriş gerilimi yarıya indirildiği halde kararlı durumda yaklaşık 400 mV civarında (20 °C) bir değişim gözlenmiştir. Tam giriş gerilimi uygulandığında 87 °C lik bir artış gözlenmişken, gerilim değeri yarıya indirildiğinde sadece 20 °C lik bir artış kaydedilmiştir yani sıcaklık gerilimin kendisiyle değil karesiyle doğru orantılı olarak değişmektedir. Dolayısıyla sistemin matematiksel modeli çıkarılırken giriş geriliminin karesi alınmıştır.

Sistemin matematiksel modelinin çıkarılmasında Matlab Simulink simulasyon programı kullanılmış ve buradaki sonuçlar operatör tarafından butonlardan girilen denet-

leyici parametrelerin değer aralıklarının belirlenmesinde kullanılmıştır. Sistemin iç yüzeylerine de bu ısıtıcıdan yerleştirilmesi durumunda ölü zaman gecikmesi azaltılabilir.

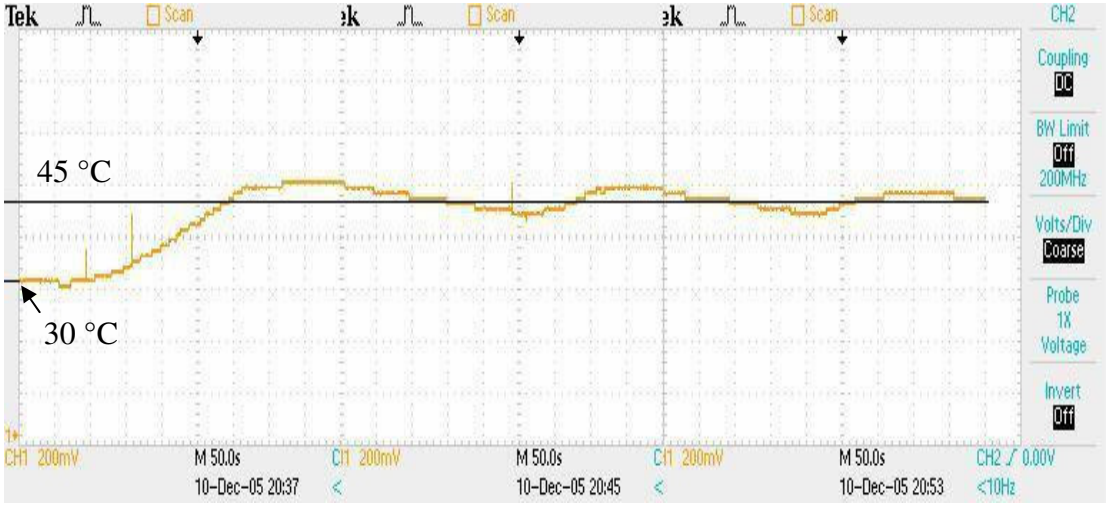
Anahtarlama anında PWM kartından mikrodenetleyiciye yansıyabilecek parazitik etki osiloskopta sonuç sırasında nadiren de olsa görülmekle beraber, bu etkinin azaltılması için ısıtıcının ve sistem fanının güç kabloları tesis içerisinde ayrı bir kablo kanalından geçirilmiş ve doluluk oranı değişkeninin maksimum değeri yazılım içerisinde 950 olarak sınırlandırılmıştır.

5.1. Deneysel Çalışma ve Simulink Benzetim Sonuçları

Deneysel çalışma gerçekleştirilirken osiloskobun Time/Div ayarı 50 s/Div, Volts/Div ayarı ise 200 mV/Div olarak ayarlanmıştır. Ortam sıcaklığı 30 °C iken set değer 45 °C olarak girilmiştir.

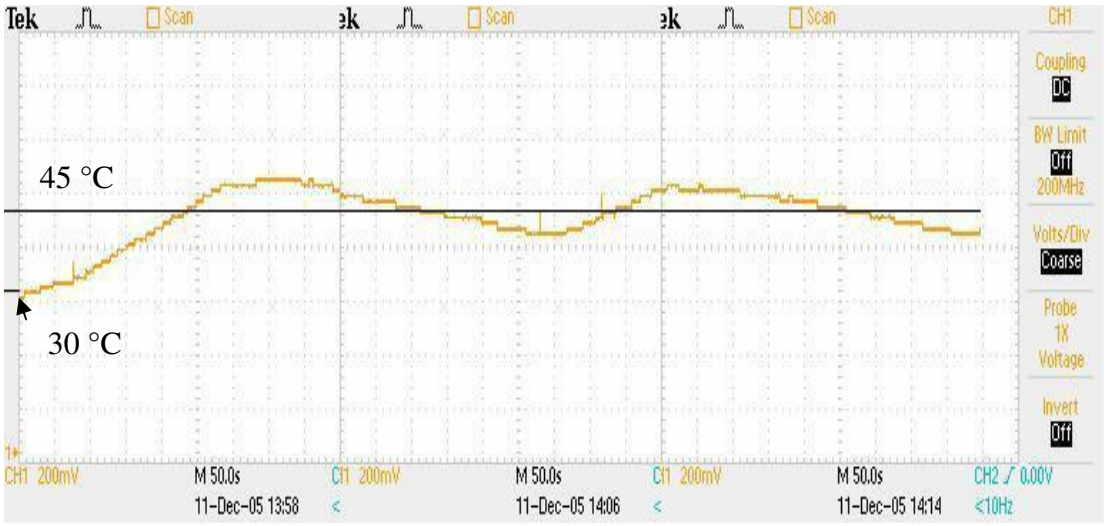
5.1.1. On/Off denetleyici

1. Ölçüm: Ortam Sıcaklığı:30 °C, Set Değer: 45 °C, Diferansiyel Aralık: 1



Şekil 5.3. On/Off 30-45 °C, diferansiyel aralık=1

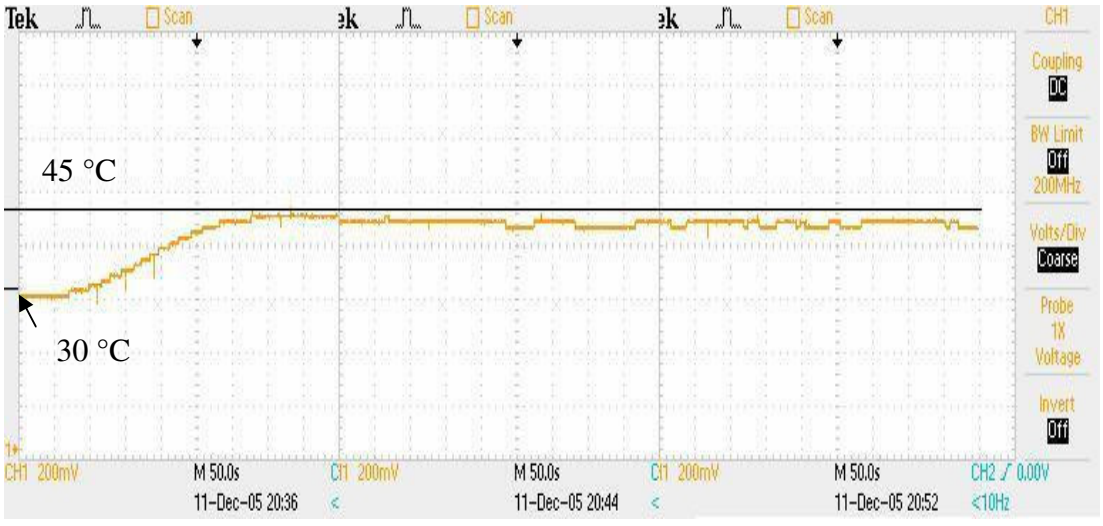
2. Ölçüm: Ortam Sıcaklığı:30 °C, Set Değer: 45 °C, Diferansiyel Aralık: 3



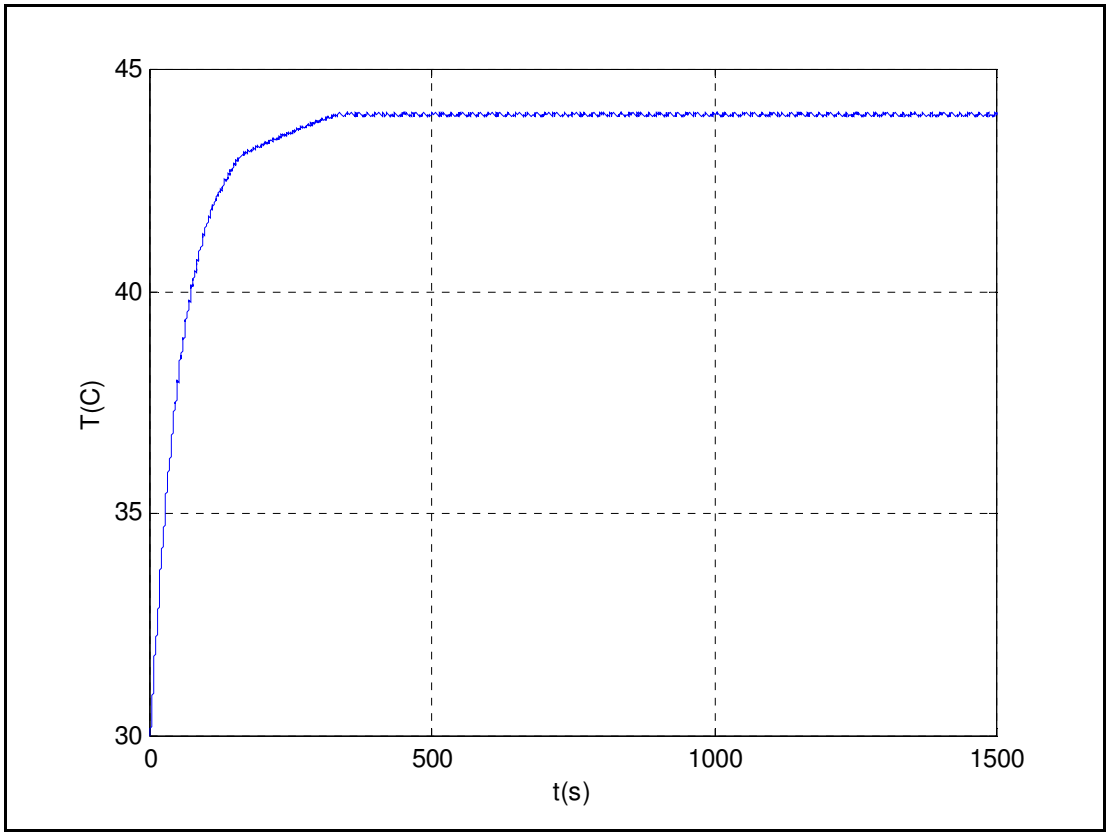
Şekil 5.4. On/Off 30-45 °C, diferansiyel aralık=3

5.1.2. Oransal (P) denetleyici

1. Ölçüm: Ortam Sıcaklığı:30 °C, Set Değer: 45 °C, K_P=80



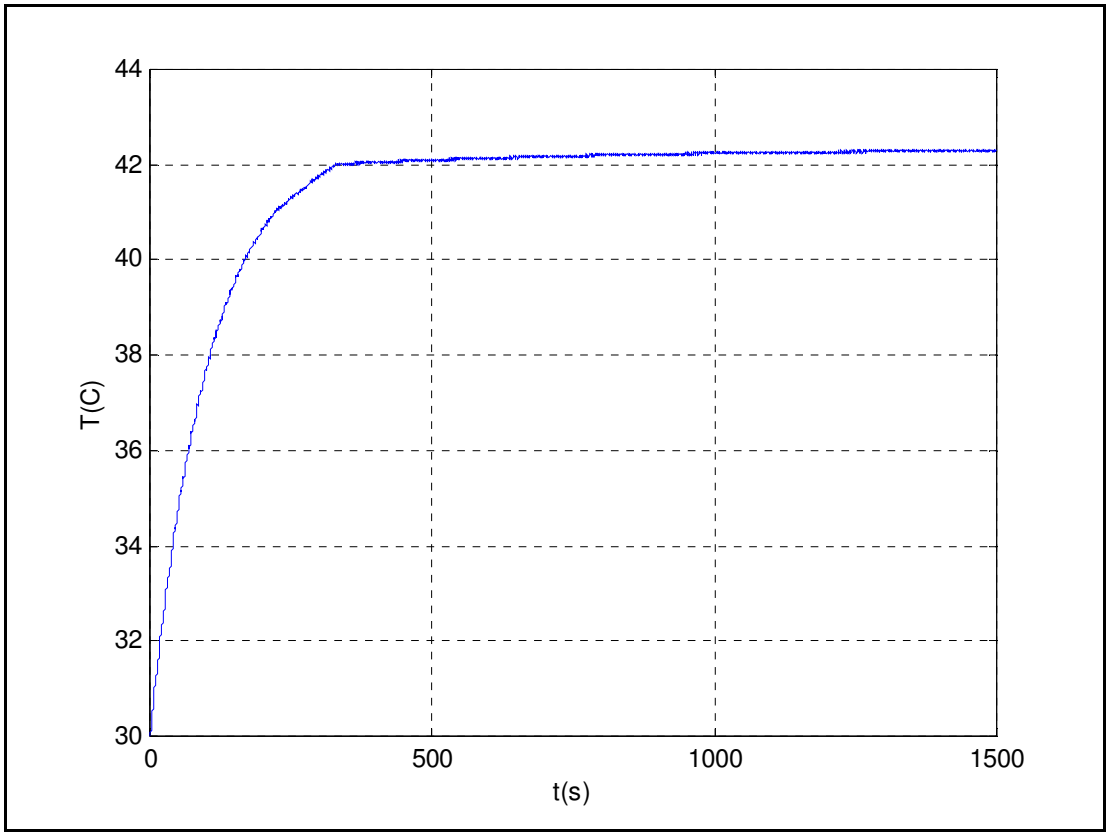
Şekil 5.5. Oransal 30-45 °C, K_P=80



Şekil 5.6. Simulink- P 30-45 °C, K_P=80

2. Ölçüm: Ortam Sıcaklığı: 30 °C, Set Değer: 45 °C, K_P=40

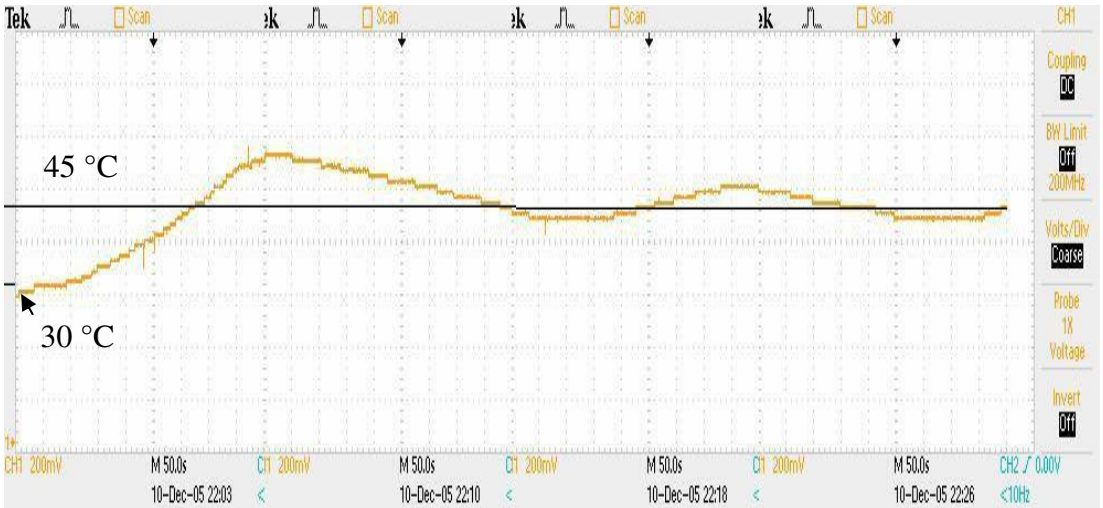
Şekil 5.7. Oransal 30-45 °C, K_P=40



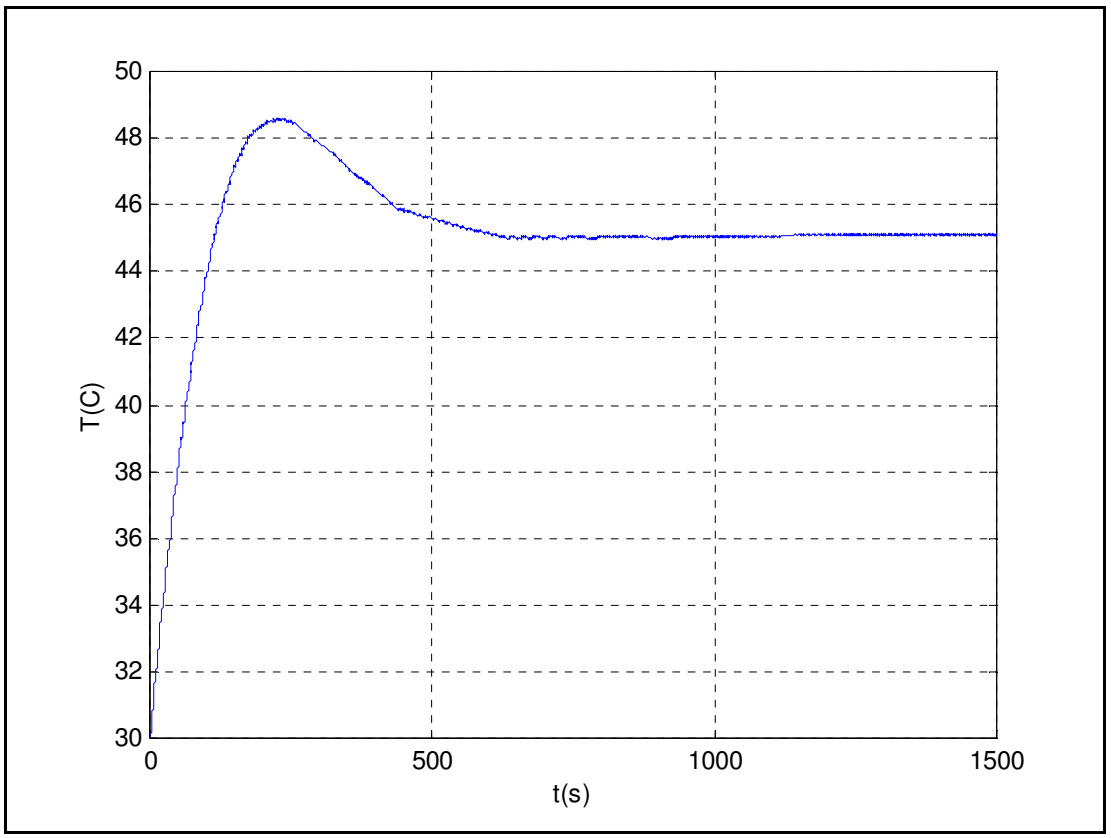
Şekil 5.8. Simulink- P 30-45 °C, K_P=40

5.1.3. Oransal+İntegral (PI) denetleyici

1. Ölçüm: Ortam Sıcaklığı: 30 °C, Set Değer: 45 °C, K_P=60, K_I=0,6

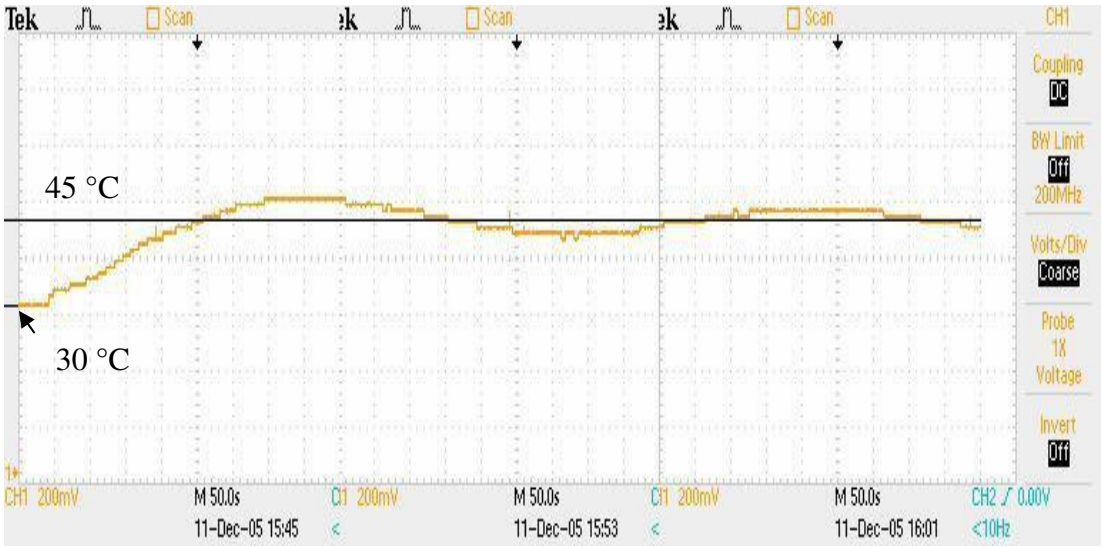


Şekil 5.9. Oransal+İntegral 30-45 °C, K_P=60, K_I=0,6

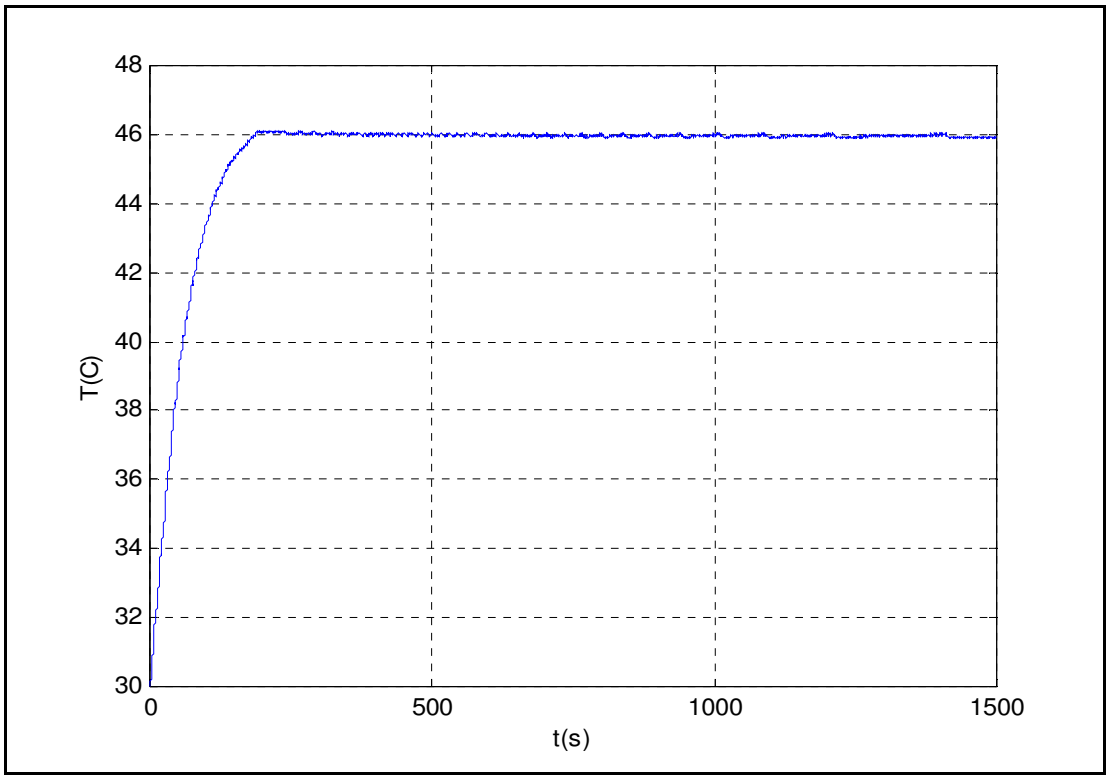


Şekil 5.10. Simulink- PI 30-45 °C, K_P=60, K_I=0,6

2. Ölçüm: Ortam Sıcaklığı:30 °C, Set Değer: 45 °C, K_P=90, K_I=0,3



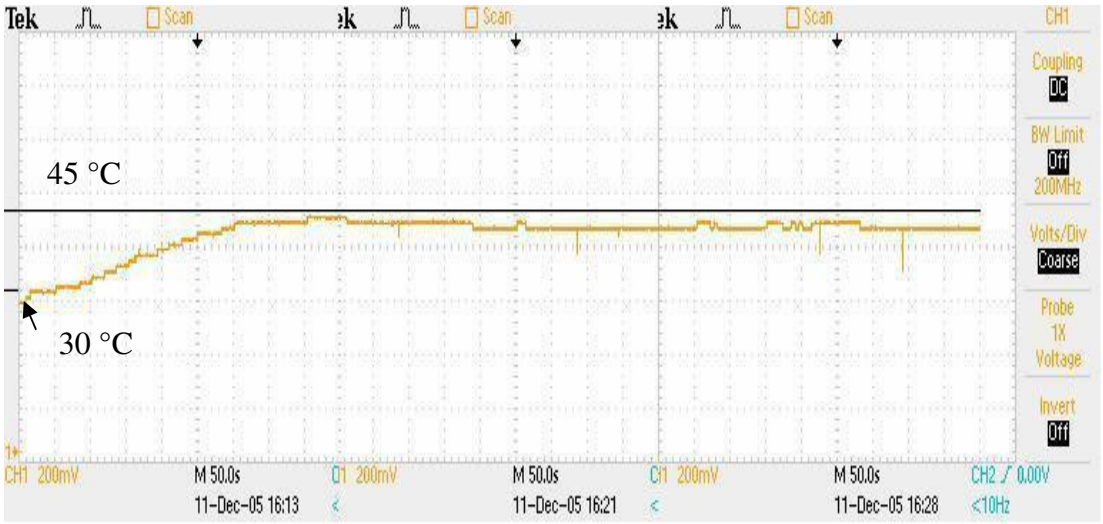
Şekil 5.11. Oransal+İntegral 30-45 °C, K_P=90, K_I=0,3



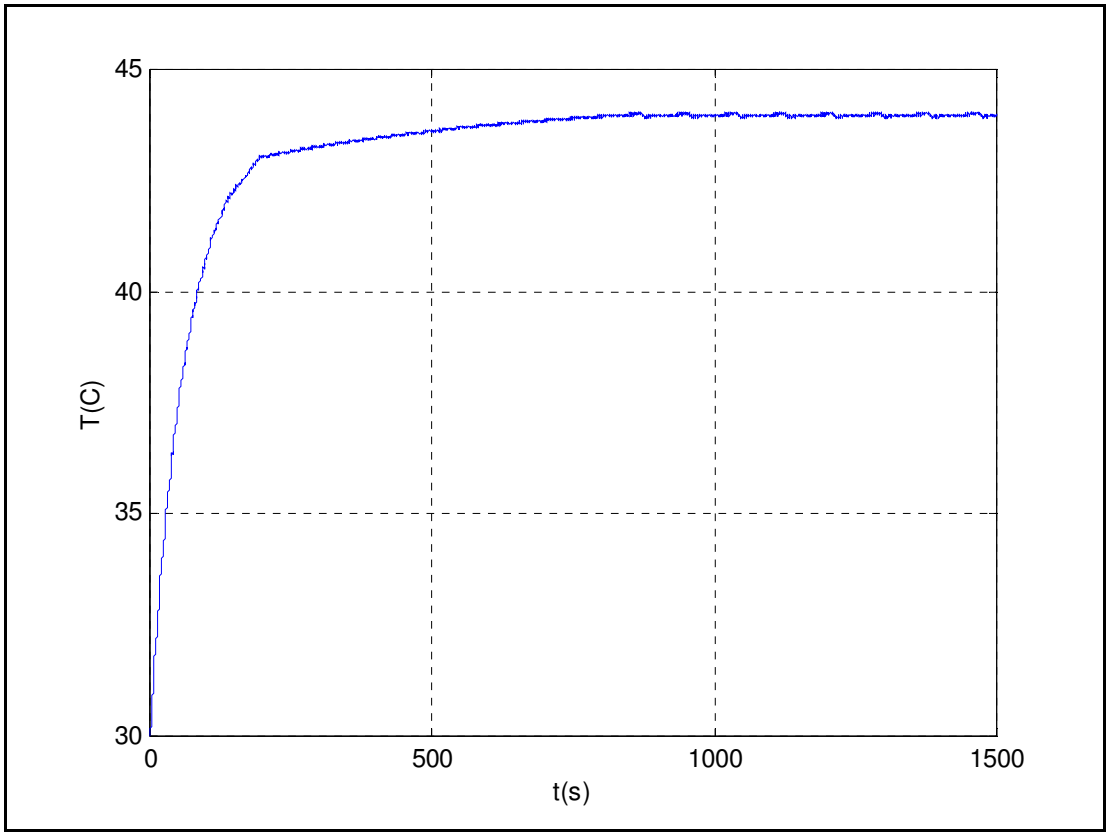
Şekil 5.12. Simulink- PI 30-45 °C, K_P=90, K_I=0,3

5.1.4. Oransal+Türevsel (PD) denetleyici

1. Ölçüm: Ortam Sıcaklığı: 30 °C, Set Değer: 45 °C, K_P=70, K_D=6

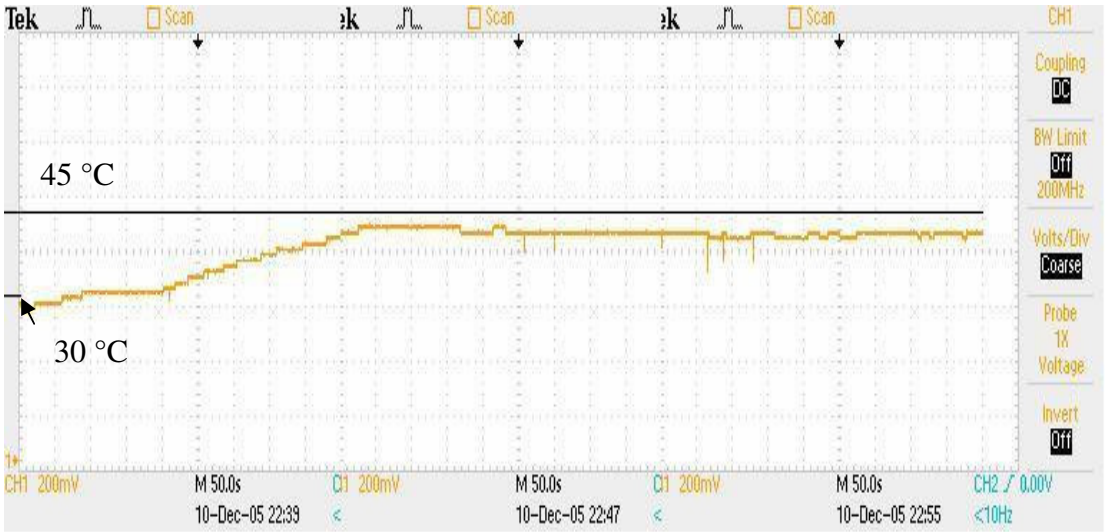


Şekil 5.13. Oransal+Türevsel 30-45 °C, K_P=70, K_D=6

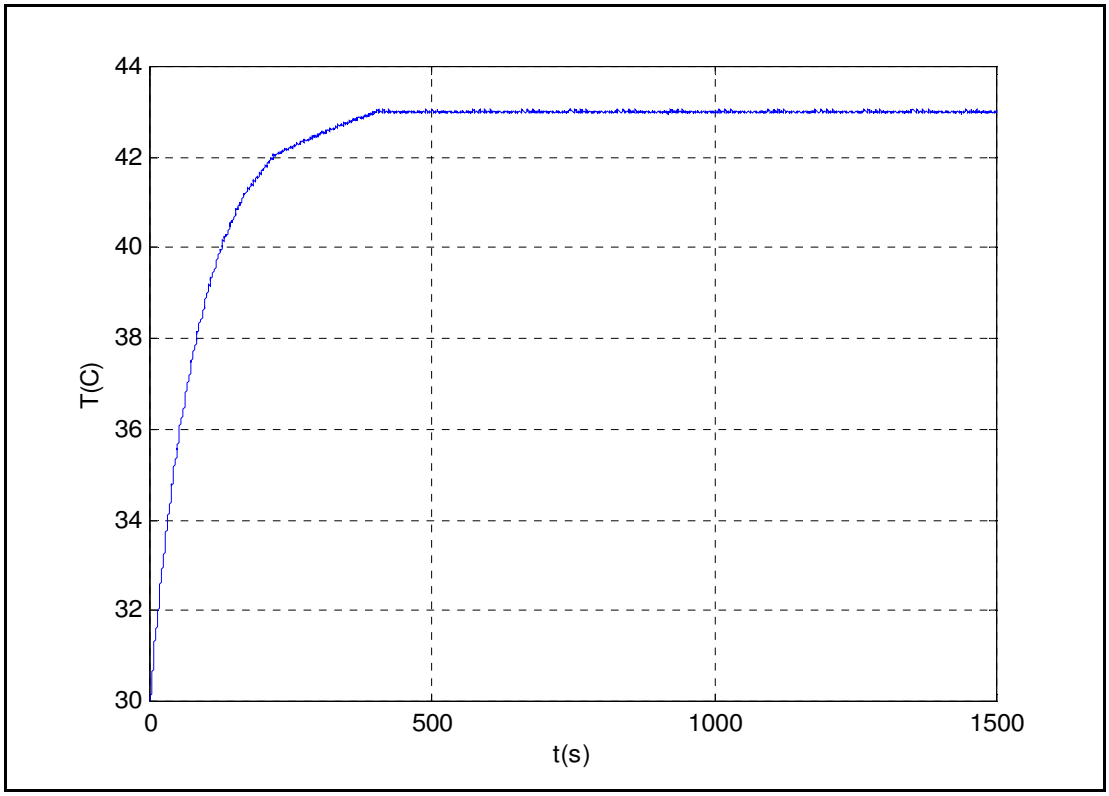


Şekil 5.14. Simulink- PD 30-45 °C, K_P=70, K_D=6

2. Ölçüm: Ortam Sıcaklığı: 30 °C, Set Değer: 45 °C, $K_P = 50$, $K_D = 4$



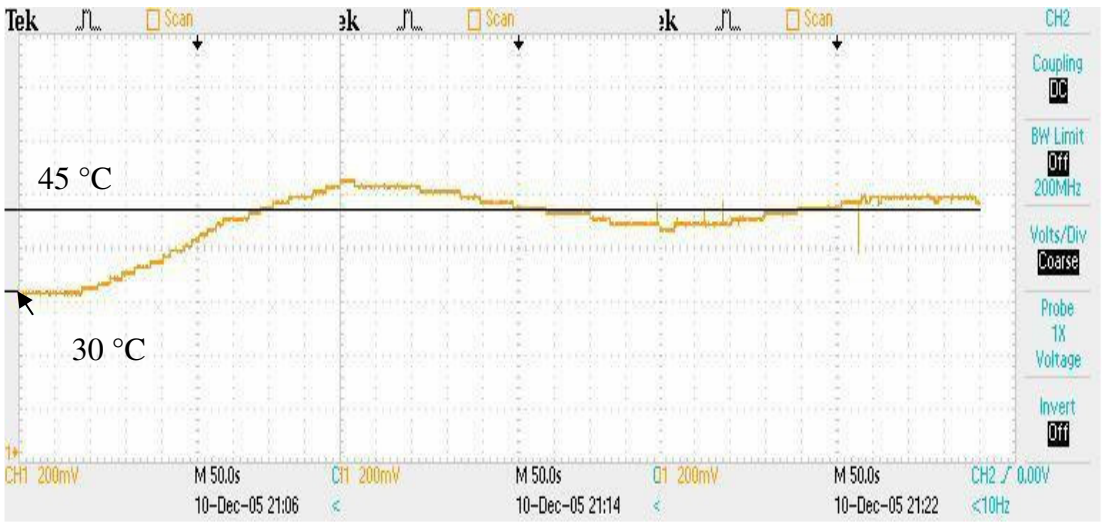
Şekil 5.15. Oransal+Türevsel 30-45 °C, K_P=50, K_D=4



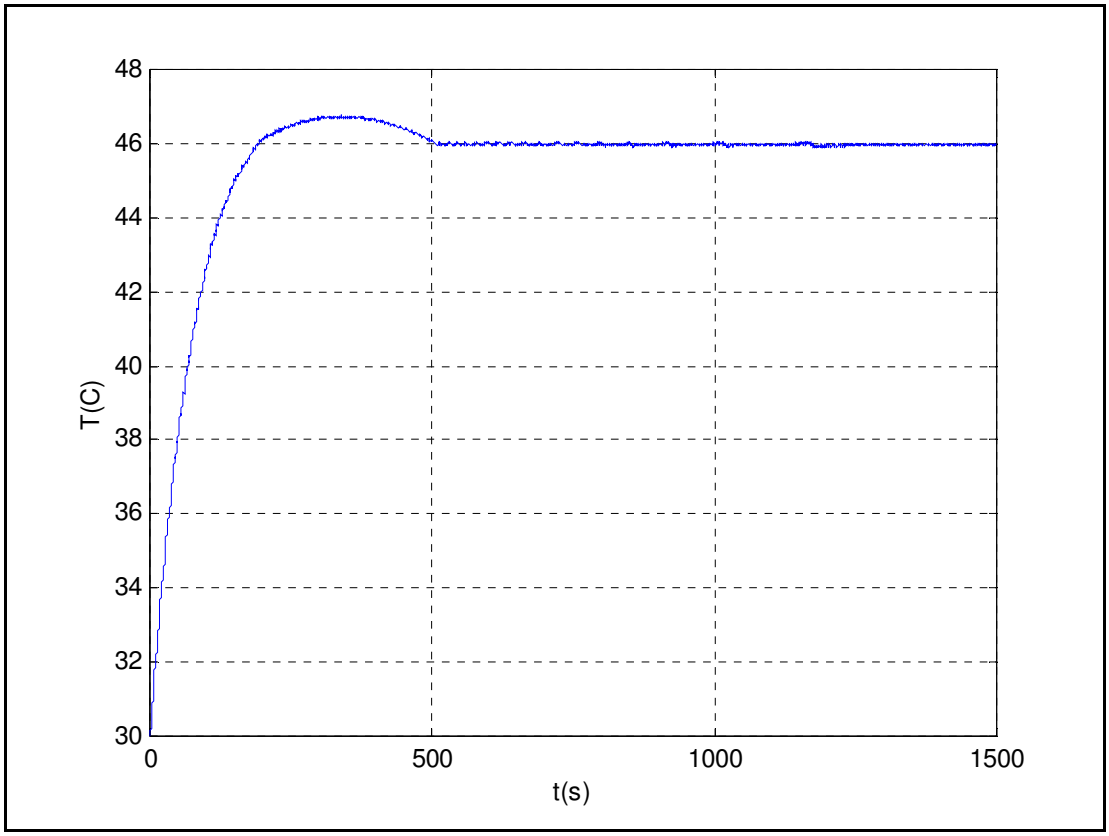
Şekil 5.16. Simulink- PD 30-45 °C, K_P=50, K_D=4

5.1.5. Oransal+İntegral+Türevsel (PID) denetleyici

1. Ölçüm: Ortam Sıcaklığı:30 °C, Set Değer: 45 °C, K_P=70, K_D=5, K_I=0,3

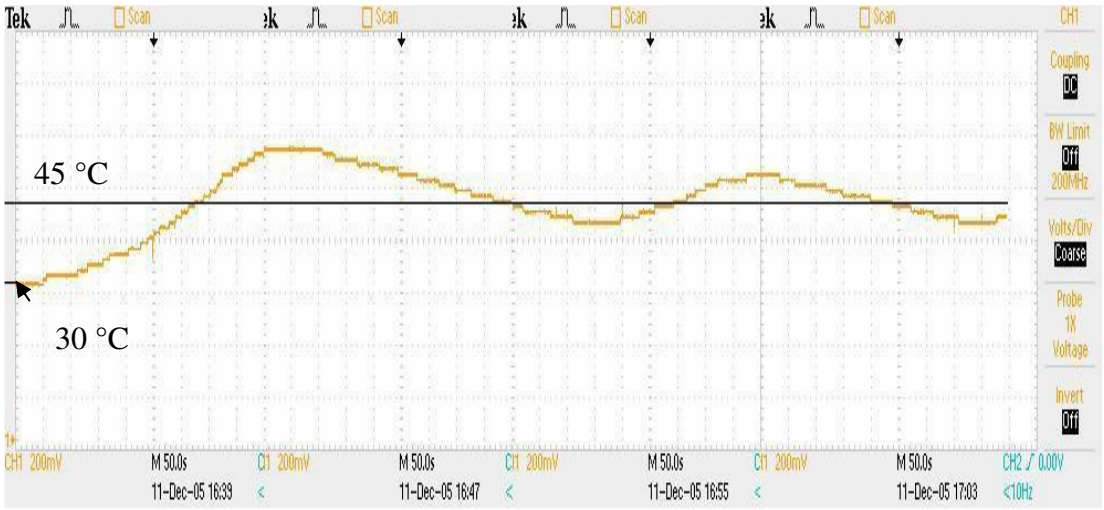


Şekil 5.17. Oransal+İntegral+Türevsel 30-45 °C, K_P=70, K_D=5, K_I=0,3

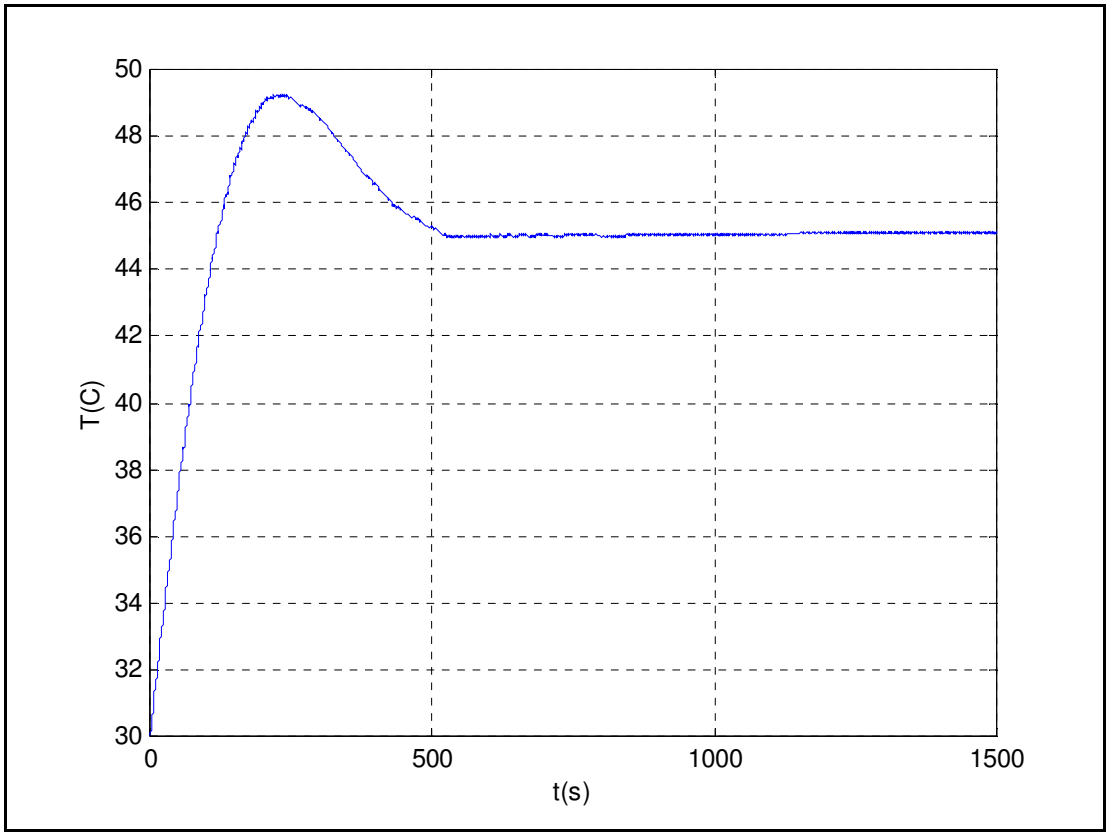


Şekil 5.18. Simulink-PID 30-45 °C, K_P=70, K_D=5, K_I=0,3

2. Ölçüm: Ortam Sıcaklığı:30 °C, Set Değer: 45 °C, K_P=50, K_D=7, K_I=0,6



Şekil 5.19. Oransal+İntegral+Türevsel 30-45 °C, K_P=50, K_D=7, K_I=0,6



Şekil 5.20. Simulink-PID 30-45 °C, K_P=50, K_D=7, K_I=0,6

KAYNAKLAR

1. Ogata, K., "Modern Control Engineering", ***Prentice Hall,*** US, 182-190 (1990).

2. İbrahim, D., "Microcontroller-Based Temperature Monitoring And Control", ***Newnes,*** England, 171-219 (2002).

3. Başar, E., "PIC16C74B Mikrodenetleyici Tabanlı 5 Bölgeli Isıtma Kontrol Sistemi", Yüksek Lisans Tezi, ***G. Ü. Fen Bil. Enstitüsü*****,** Ankara, 1-2 (2002).

4. Savaş, Y., "Sayısal Göstergeli Sıcaklık Ölçme ve Kontrol Cihazının Tasarım ve Yapımı", Yüksek Lisans Tezi, ***G. Ü. Fen Bil. Enstitüsü*****,** Ankara, 1-44 (1988).

5. Kalender, O., "MC 6802 Tabanlı Oransal Sıcaklık Kontrol Devresinin Tasarımı ve Yapımı", Yüksek Lisans Tezi, ***G. Ü. Fen Bil. Enstitüsü*****,** Ankara, 1-77 (1991).

6. Duran, F., "PIC Mikrodenetleyicisi ile DC Motor Hız Kontrolü", Yüksek Lisans Tezi, ***G. Ü. Fen Bil. Enstitüsü*****,** Ankara, 23-47 (2001).

7. Gök, İ., "Faz Kaymalı Darbe Genişlik Modülasyonlu DC-DC Dönüştürücü Kullanan Akü Şarj Cihazı", Yüksek Lisans Tezi, ***G. Ü. Fen Bil. Enstitüsü*****,** Ankara, 1-3 (2005).

8. İbrahim, D., "PIC C ile Sıcaklık Projeleri", ***Bileşim***, 2-72 (2003).

9. Vatansever, F., "Algoritma Geliştirme ve Programlamaya Giriş", ***Seçkin***, 49-59, 435-436 (2002).

10. Gardner, N., "An Introduction to Programming the Microchip PIC in C", ***Bluebird Electronics***, 1-121 (1998).

11. İnternet: "PIC 16F877 Yapısı" http://www.robolab.gazi.edu.tr (2005).

12. İnternet: "PIC 16F877 Datasheet and Application Notes" http://www.microchip.com (2005).

13. İnternet: "PIC Basic Programming Forums" http://www.picbasic.org/forums (2005).

14. İnternet: "PIC modüller" http://www.mikromodul.com (2005).

15. İnternet: "MATLAB Simulink'e Giriş" http://www.bilimonline.8k.com/matlab/simulink/giris.htm (2005).

16. İnternet: "ICProg PIC Programlayıcı" http://www.ic-prog.com (2005).

Not: İnternet kaynakları EK'te CD'de sunulmuştur.

EKLER

EK-1. Denetleyici yazılımlarının assembly çıktıları

LIST
VAR1 Equ 49
DURUM Equ 50
DERECE_SET Equ 51
EKRAN Equ 52
KONTROLTIPI Equ 53
PARAMETRESEC Equ 54
_I Equ 55
KONTROLSTART Equ 56
_C Equ 57
VAR2 Equ 58
VAR2H Equ 59
KP_VALUE Equ 60
KP_VALUEH Equ 61
KI_VALUE Equ 62
KI_VALUEH Equ 63
KD_VALUE Equ 64
KD_VALUEH Equ 65
DA_VALUE Equ 66
DA_VALUEH Equ 67
FARK Equ 68
FARKH Equ 69
DUTY Equ 70
DUTYH Equ 71
PIK Equ 72
PIKH Equ 73
DEGISIM Equ 74
DEGISIMH Equ 75
ONOFFSAYAC Equ 76
DERECE_UST Equ 77
DERECE_ALT Equ 78
KUSUR Equ 79
KUSURH Equ 80
TEMP Equ 81
TEMPH Equ 82
TEMP_TAM Equ 83
TEMP_TAMH Equ 84
TEMPTOPLA Equ 85
TEMPTOPLAH Equ 86
TEMPORT Equ 87
TEMPORTH Equ 88
FSAYAC Equ 89
F1 Equ 90
F1H Equ 91
F2 Equ 92
F2H Equ 93
F3 Equ 94
F3H Equ 95
F_0 Equ 96
F_0H Equ 97
EKSI Equ 98
FARK_E Equ 99
FARK_EH Equ 100

```
FARK_D Equ 101
FARK_DH Equ 102
F_TRV Equ 103
F_TRVH Equ 104
F_INT Equ 105
F_INTH Equ 106
F_DER_TOPL Equ 107
F_DER_TOPLH Equ 108
F_INT_TOPL Equ 109
F_INT_TOPLH Equ 110
ED1 Equ 111
ED2 Equ 112
F_DER Equ 113
F_DERH Equ 114
        #Define BUTONDOWN PORTC,0
        #Define BUTONSET PORTC,1
        #Define BUTONUP PORTC,3
        #Define ROLE1 PORTB,0
        #Define ROLE2 PORTC,5
        #Define ROLE3 PORTC,6
        #Define ROLE4 PORTC,4
        #Define ROLE5 PORTC,7
        #Define DAC PORTD
        #Define LED_4 PORTA,0
        #Define DQ PORTB,2
F1_SOF equ $ ; SONKOD.BAS
F1_000005 equ $ ; in [SONKOD.BAS] CONFIG XT_OSC , WDT_OFF , PWRTE_OFF ,
BODEN_OFF , LVP_OFF , WRTE_ON
F1_000026 equ $ ; in [SONKOD.BAS] TRISA.0=0
        Bsf STATUS,5
ram_bank = 1
        Bcf TRISA,0
F1_000027 equ $ ; in [SONKOD.BAS] TRISB.0=0
        Bcf TRISB,0
F1_000028 equ $ ; in [SONKOD.BAS] TRISC = %00001011
        Movlw 11
        Movwf TRISC
F1_000029 equ $ ; in [SONKOD.BAS] TRISD = %00000000
        Clrf TRISD
INITIAL
        Bcf STATUS,5
ram_bank = 0
F1_000068 equ $ ; in [SONKOD.BAS] durum=255
        Movlw 255
        Movwf DURUM
F1_000069 equ $ ; in [SONKOD.BAS] DERECE_SET=36
        Movlw 36
        Movwf DERECE_SET
F1_000070 equ $ ; in [SONKOD.BAS] kontroltipi=1
        Movlw 1
        Movwf KONTROLTIPI
F1_000071 equ $ ; in [SONKOD.BAS] var1=0
        Clrf VAR1
F1_000072 equ $ ; in [SONKOD.BAS] var2=0
        Clrf VAR2H
        Clrf VAR2
F1_000073 equ $ ; in [SONKOD.BAS] KONTROLSTART=0
```

```
        Clrf KONTROLSTART
F1_000074 equ $ ; in [SONKOD.BAS] FSAYAC=0
        Clrf FSAYAC
F1_000075 equ $ ; in [SONKOD.BAS] FARK_E=0
        Clrf FARK_EH
        Clrf FARK_E
F1_000076 equ $ ; in [SONKOD.BAS] F_INT=0
        Clrf F_INTH
        Clrf F_INT
F1_000077 equ $ ; in [SONKOD.BAS] F1=20
        Clrf F1H
        Movlw 20
        Movwf F1
F1_000078 equ $ ; in [SONKOD.BAS] KP_VALUE=50
        Clrf KP_VALUEH
        Movlw 50
        Movwf KP_VALUE
F1_000079 equ $ ; in [SONKOD.BAS] KI_VALUE=3
        Clrf KI_VALUEH
        Movlw 3
        Movwf KI_VALUE
F1_000080 equ $ ; in [SONKOD.BAS] KD_VALUE=3
        Clrf KD_VALUEH
        Movlw 3
        Movwf KD_VALUE
F1_000081 equ $ ; in [SONKOD.BAS] DA_VALUE=0
        Clrf DA_VALUEH
        Clrf DA_VALUE
F1_000082 equ $ ; in [SONKOD.BAS] ONOFFSAYAC=0
        Clrf ONOFFSAYAC
F1_000083 equ $ ; in [SONKOD.BAS] EKSI=0
        Clrf EKSI
F1_000084 equ $ ; in [SONKOD.BAS] F_DER=0
        Clrf F_DERH
        Clrf F_DER
F1_000085 equ $ ; in [SONKOD.BAS] F_0=0
        Clrf F_0H
        Clrf F_0
F1_000086 equ $ ; in [SONKOD.BAS] ED1=25
        Movlw 25
        Movwf ED1
F1_000091 equ $ ; in [SONKOD.BAS] TRISC.2 = 0    'PORTC.2 (CCP1) çıkış yapıldı
        Bsf STATUS,5
ram_bank = 1
        Bcf TRISC,2
F1_000092 equ $ ; in [SONKOD.BAS] CCP1CON = %00001100  'CCP1 yazmacı PWM'e ayarlandı
        Movlw 12
        Bcf STATUS,5
ram_bank = 0
        Movwf CCP1CON
F1_000093 equ $ ; in [SONKOD.BAS] T2CON = %00000100  'Timer2 aktif edildi, Prescale=4
        Movlw 4
        Movwf T2CON
F1_000094 equ $ ; in [SONKOD.BAS] PR2 = 255    'PR2 1KHz çıkışa ayarlandı
        Bsf STATUS,5
ram_bank = 1
        Movlw 255
```

```
        Movwf PR2
F1_000096 equ $ ; in [SONKOD.BAS] Duty = 0 : Ccp1con.4 = Duty.0 : Ccp1con.5 = Duty.1 : Ccpr1l
= Duty >> 2
        Bcf STATUS,5
ram_bank = 0
        Clrf DUTYH
        Clrf DUTY
; Bit_Bit DUTY,0,CCP1CON,4
        Bsf CCP1CON,4
        Btfss DUTY,0
        Bcf CCP1CON,4
; Bit_Bit DUTY,1,CCP1CON,5
        Bsf CCP1CON,5
        Btfss DUTY,1
        Bcf CCP1CON,5
        Rrf DUTYH,W
        Movwf PP4
        Rrf DUTY,W
        Movwf CCPR1L
        Rrf PP4,F
        Rrf CCPR1L,F
        Movfw DUTYH
        Movwf PP0H
        Movfw DUTY
        Movwf PP0
        Movlw 2
        F@Call r@sh
        Movwf CCPR1L
F1_000107 equ $ ; in [SONKOD.BAS] PRINT $FE , 1 : DELAYMS 30
        Movlw 254
        F@Call Print
        Movlw 1
        F@Call Print
        Movlw 30
        F@Call dl@ms
F1_000108 equ $ ; in [SONKOD.BAS] PRINT $FE , 2
        Movlw 254
        F@Call Print
        Movlw 2
        F@Call Print
F1_000109 equ $ ; in [SONKOD.BAS] PRINT $FE , $0C
        Movlw 254
        F@Call Print
        Movlw 12
        F@Call Print
F1_000113 equ $ ; in [SONKOD.BAS] OPTION_REG=$55      'Prescaler (Bölme Değeri) =1/64
        Movlw 85
        Bsf STATUS,5
ram_bank = 1
        Movwf OPTION_REG
F1_000114 equ $ ; in [SONKOD.BAS] INTCON=$a0
        Movlw 160
        Movwf INTCON
        Bsf INTCON,7
        Bcf STATUS,5
ram_bank = 0
        F@Jump INT@LBL2
```

```
INT@LBL1
        Btfsc INTCON,7
        Return
        F@Jump PIKINT
INT@LBL2
        F@Call INT@LBL1
F1_000116 equ $ ; in [SONKOD.BAS] ekran=1
        Movlw 1
        Movwf EKRAN
        F@Call INT@LBL1
F1_000117 equ $ ; in [SONKOD.BAS] gosub lcdcikis
        F@Call LCDCIKIS
        F@Call INT@LBL1
F1_000118 equ $ ; in [SONKOD.BAS] delayms 1000
        Movlw 3
        Movwf PP1H
        Movlw 232
        F@Call dly@w
F1_000124 equ $ ; in [SONKOD.BAS] DISABLE
PIKINT
F1_000126 equ $ ; in [SONKOD.BAS] PIK=PIK+1
        Incf PIK,F
        Skpnz
        Incf PIKH,F
F1_000127 equ $ ; in [SONKOD.BAS] IF PIK<61 THEN goto PIEXIT
        Movfw PIKH
        set@page bc@ll2
        Btfss STATUS,2
        Goto bc@ll2
        Movlw 61
        Subwf PIK,W
        set@page bc@ll2
        Btfsc STATUS,0
        Goto bc@ll2
        F@Jump PIEXIT
bc@ll2
F1_000128 equ $ ; in [SONKOD.BAS] PIK=0
        Clrf PIKH
        Clrf PIK
F1_000129 equ $ ; in [SONKOD.BAS] DEGISIM=1
        Clrf DEGISIMH
        Movlw 1
        Movwf DEGISIM
PIEXIT
F1_000131 equ $ ; in [SONKOD.BAS] INTCON.2=0
        Bcf INTCON,2
F1_000132 equ $ ; in [SONKOD.BAS] RESUME
        Retfie
HEARTBEAT
F1_000137 equ $ ; in [SONKOD.BAS] led_4=1
        Bsf PORTA,0
F1_000138 equ $ ; in [SONKOD.BAS] delayms 500
        Movlw 1
        Movwf PP1H
        Movlw 244
        F@Call dly@w
F1_000139 equ $ ; in [SONKOD.BAS] led_4=0
```

```
        Bcf PORTA,0
F1_000140 equ $ ; in [SONKOD.BAS] delayms 500
        Movlw 1
        Movwf PP1H
        Movlw 244
        F@Call dly@w
F1_000141 equ $ ; in [SONKOD.BAS] return
        Return
SICAKLIKOKU
F1_000148 equ $ ; in [SONKOD.BAS] DELAYMS 500
        Movlw 1
        Movwf PP1H
        Movlw 244
        F@Call dly@w
F1_000149 equ $ ; in [SONKOD.BAS] OWRITE DQ, 1, [$CC, $44]  ' Sıcaklığı hesaplaması için
algılayıcıya komut gönderme
        Movlw PORTB
        Movwf GEN
        Movlw 4
        Movwf GENH
        Movlw 1
        Movwf GPR
        F@Call Ow@rst2
        Movlw 204
        F@Call ow@out
        Movlw 68
        F@Call ow@out
F1_000150 equ $ ; in [SONKOD.BAS] REPEAT
bc@ll3
F1_000151 equ $ ; in [SONKOD.BAS] DELAYMS 25    ' İşlem bitene kadar bekleme
        Movlw 25
        F@Call dl@ms
F1_000152 equ $ ; in [SONKOD.BAS] OREAD DQ, 4, [C]    ' İşlem süresince portb.2=0 olur
        Movlw PORTB
        Movwf GEN
        Movlw 4
        Movwf GENH
        Movlw 4
        Movwf GPR
        F@Call Ow@in
        Movwf _C
F1_000153 equ $ ; in [SONKOD.BAS] UNTIL C <> 0   ' Okuma işlemi tamam
        Movfw _C
        set@page bc@ll3
        Btfsc STATUS,2
        Goto bc@ll3
bc@ll4
F1_000154 equ $ ; in [SONKOD.BAS] OWRITE DQ, 1, [$CC, $BE]  ' Sıcaklık değerini RAM dan
okuması için komut gönderiyor
        Movlw PORTB
        Movwf GEN
        Movlw 4
        Movwf GENH
        Movlw 1
        Movwf GPR
        F@Call Ow@rst2
        Movlw 204
```

```
            F@Call ow@out
            Movlw 190
            F@Call ow@out
F1_000156 equ $ ; in [SONKOD.BAS] OREAD DQ, 2,[TEMP.LOWBYTE,TEMP.HIGHBYTE]
            Movlw PORTB
            Movwf GEN
            Movlw 4
            Movwf GENH
            Movlw 2
            Movwf GPR
            F@Call Ow@in
            Movwf TEMP
            F@Call Ow@in
            Movwf TEMPH
            F@Call Ow@rst1
F1_000159 equ $ ; in [SONKOD.BAS] durum=255
            Movlw 255
            Movwf DURUM
F1_000160 equ $ ; in [SONKOD.BAS] TEMP_TAM = TEMP/16
            Movfw TEMP
            Movwf TEMP_TAM
            Movfw TEMPH
            Movwf TEMP_TAMH
            Movlw 4
            Movwf PP0
D@LB1
            Clrc
            Rrf TEMP_TAMH,F
            Rrf TEMP_TAM,F
            Decf PP0,F
            set@page D@LB1
            Btfss STATUS,2
            Goto D@LB1
F1_000161 equ $ ; in [SONKOD.BAS] return
            Return
SICAKLIKDONGUSU
F1_000168 equ $ ; in [SONKOD.BAS] if degisim=1 then
            Decf DEGISIM,W
            Iorwf DEGISIMH,W
            set@page bc@ll6
            Btfss STATUS,2
            Goto bc@ll6
F1_000169 equ $ ; in [SONKOD.BAS] degisim=0
            Clrf DEGISIMH
            Clrf DEGISIM
F1_000170 equ $ ; in [SONKOD.BAS] TEMPTOPLA = 0
            Clrf TEMPTOPLAH
            Clrf TEMPTOPLA
F1_000171 equ $ ; in [SONKOD.BAS] role2=0 : role3=0 : role4=0 : role5=0 : delayms 5 : role1=1
'1.algılayıcı seçildi
            Bcf PORTC,5
            Bcf PORTC,6
            Bcf PORTC,4
            Bcf PORTC,7
            Movlw 5
            F@Call dl@ms
            Bsf PORTB,0
```

```
F1_000172 equ $ ; in [SONKOD.BAS] gosub sicaklikoku
        F@Call SICAKLIKOKU
F1_000173 equ $ ; in [SONKOD.BAS] TEMPTOPLA =TEMP_TAM
        Movfw TEMP_TAMH
        Movwf TEMPTOPLAH
        Movfw TEMP_TAM
        Movwf TEMPTOPLA
F1_000174 equ $ ; in [SONKOD.BAS] if butonset=0 then goto cks101
        set@page bc@ll8
        Btfsc PORTC,1
        Goto bc@ll8
        F@Jump CKS101
bc@ll8
F1_000175 equ $ ; in [SONKOD.BAS] role1=0 : role3=0 : role4=0 : role5=0 : delayms 5 : role2=1
'2.algılayıcı seçildi
        Bcf PORTB,0
        Bcf PORTC,6
        Bcf PORTC,4
        Bcf PORTC,7
        Movlw 5
        F@Call dl@ms
        Bsf PORTC,5
F1_000176 equ $ ; in [SONKOD.BAS] gosub sicaklikoku
        F@Call SICAKLIKOKU
F1_000177 equ $ ; in [SONKOD.BAS] TEMPTOPLA = TEMPTOPLA + TEMP_TAM
        Movfw TEMP_TAM
        Addwf TEMPTOPLA,F
        Movfw TEMP_TAMH
        Skpnc
        Addlw 1
        Addwf TEMPTOPLAH,F
F1_000178 equ $ ; in [SONKOD.BAS] if butonset=0 then goto cks101
        set@page bc@ll10
        Btfsc PORTC,1
        Goto bc@ll10
        F@Jump CKS101
bc@ll10
F1_000179 equ $ ; in [SONKOD.BAS] role2=0 : role1=0 : role4=0 : role5=0 : delayms 5 : role3=1
'3.algılayıcı seçildi
        Bcf PORTC,5
        Bcf PORTB,0
        Bcf PORTC,4
        Bcf PORTC,7
        Movlw 5
        F@Call dl@ms
        Bsf PORTC,6
F1_000180 equ $ ; in [SONKOD.BAS] gosub sicaklikoku
        F@Call SICAKLIKOKU
F1_000181 equ $ ; in [SONKOD.BAS] TEMPTOPLA = TEMPTOPLA + TEMP_TAM
        Movfw TEMP_TAM
        Addwf TEMPTOPLA,F
        Movfw TEMP_TAMH
        Skpnc
        Addlw 1
        Addwf TEMPTOPLAH,F
F1_000182 equ $ ; in [SONKOD.BAS] if butonset=0 then goto cks101
        set@page bc@ll12
```

```
            Btfsc PORTC,1
            Goto bc@ll12
            F@Jump CKS101
bc@ll12
F1_000183 equ $ ; in [SONKOD.BAS] role2=0 : role3=0 : role1=0 : role5=0 : delayms 5 : role4=1
'4.algılayıcı seçildi
            Bcf PORTC,5
            Bcf PORTC,6
            Bcf PORTB,0
            Bcf PORTC,7
            Movlw 5
            F@Call dl@ms
            Bsf PORTC,4
F1_000184 equ $ ; in [SONKOD.BAS] gosub sicaklikoku
            F@Call SICAKLIKOKU
F1_000185 equ $ ; in [SONKOD.BAS] TEMPTOPLA = TEMPTOPLA + TEMP_TAM
            Movfw TEMP_TAM
            Addwf TEMPTOPLA,F
            Movfw TEMP_TAMH
            Skpnc
            Addlw 1
            Addwf TEMPTOPLAH,F
F1_000186 equ $ ; in [SONKOD.BAS] if butonset=0 then goto cks101
            set@page bc@ll14
            Btfsc PORTC,1
            Goto bc@ll14
            F@Jump CKS101
bc@ll14
F1_000187 equ $ ; in [SONKOD.BAS] role2=0 : role3=0 : role4=0 : role1=0 : delayms 5 : role5=1
'5.algılayıcı seçildi
            Bcf PORTC,5
            Bcf PORTC,6
            Bcf PORTC,4
            Bcf PORTB,0
            Movlw 5
            F@Call dl@ms
            Bsf PORTC,7
F1_000188 equ $ ; in [SONKOD.BAS] gosub sicaklikoku
            F@Call SICAKLIKOKU
F1_000189 equ $ ; in [SONKOD.BAS] TEMPTOPLA = TEMPTOPLA + TEMP_TAM
            Movfw TEMP_TAM
            Addwf TEMPTOPLA,F
            Movfw TEMP_TAMH
            Skpnc
            Addlw 1
            Addwf TEMPTOPLAH,F
F1_000190 equ $ ; in [SONKOD.BAS] if butonset=0 then  goto cks101
            set@page bc@ll16
            Btfsc PORTC,1
            Goto bc@ll16
            F@Jump CKS101
bc@ll16
F1_000191 equ $ ; in [SONKOD.BAS] TEMPORT=TEMPTOPLA/5
            Movfw TEMPTOPLAH
            Movwf PP0H
            Movfw TEMPTOPLA
            Movwf PP0
```

```
            Clrf PP1H
            Movlw 5
            Movwf PP1
            F@Call d@vd
            Movwf TEMPORT
            Movfw PP0H
            Movwf TEMPORTH
F1_000193 equ $ ; in [SONKOD.BAS] PRINT $FE , 1 : DELAYMS 30
            Movlw 254
            F@Call Print
            Movlw 1
            F@Call Print
            Movlw 30
            F@Call dl@ms
F1_000194 equ $ ; in [SONKOD.BAS] PRINT $FE , 2 : PRINT "TEMP:+"
            Movlw 254
            F@Call Print
            Movlw 2
            F@Call Print
            Movlw 'T'
            F@Call Print
            Movlw 'E'
            F@Call Print
            Movlw 'M'
            F@Call Print
            Movlw 'P'
            F@Call Print
            Movlw ':'
            F@Call Print
            Movlw '+'
            F@Call Print
F1_000195 equ $ ; in [SONKOD.BAS] PRINT DEC TEMPORT       ' SICAKLIĞI EKRANA
YAZDIR
            Movlw 128
            Movwf BPFH
            Clrf GEN4H
            Movfw TEMPORTH
            Movwf PP2H
            Movfw TEMPORT
            Movwf PP2
            F@Call out@dec
F1_000196 equ $ ; in [SONKOD.BAS] PRINT " C"
            Movlw ' '
            F@Call Print
            Movlw 'C'
            F@Call Print
F1_000197 equ $ ; in [SONKOD.BAS] DAC = TEMPORT
            Movfw TEMPORT
            Movwf PORTD
CKS101
F1_000199 equ $ ; in [SONKOD.BAS] end if
bc@ll6
F1_000200 equ $ ; in [SONKOD.BAS] return
            Return
BUTONKONTROL
F1_000207 equ $ ; in [SONKOD.BAS] if butondown=0 then
            set@page bc@ll18
```

```
            Btfsc PORTC,0
            Goto bc@ll18
BEKLE1
F1_000209 equ $ ; in [SONKOD.BAS] if butondown=0 then GOTO BEKLE1
            set@page bc@ll20
            Btfsc PORTC,0
            Goto bc@ll20
            F@Jump BEKLE1
bc@ll20
F1_000210 equ $ ; in [SONKOD.BAS] if durum=1 then
            Movlw 1
            Subwf DURUM,W
            set@page bc@ll22
            Btfss STATUS,2
            Goto bc@ll22
F1_000211 equ $ ; in [SONKOD.BAS] DERECE_SET=DERECE_SET-1
            Decf DERECE_SET,F
F1_000212 equ $ ; in [SONKOD.BAS] if DERECE_SET<30 then DERECE_SET= 30
            Movlw 30
            Subwf DERECE_SET,W
            set@page bc@ll24
            Btfsc STATUS,0
            Goto bc@ll24
            Movlw 30
            Movwf DERECE_SET
bc@ll24
F1_000213 equ $ ; in [SONKOD.BAS] ekran=3
            Movlw 3
            Movwf EKRAN
F1_000214 equ $ ; in [SONKOD.BAS] gosub lcdcikis
            F@Call LCDCIKIS
            F@Jump bc@ll21
F1_000215 equ $ ; in [SONKOD.BAS] elseif durum=2 then
bc@ll22
            Movlw 2
            Subwf DURUM,W
            set@page bc@ll25
            Btfss STATUS,2
            Goto bc@ll25
F1_000216 equ $ ; in [SONKOD.BAS] kontroltipi=kontroltipi-1
            Decf KONTROLTIPI,F
F1_000217 equ $ ; in [SONKOD.BAS] if kontroltipi<1 then kontroltipi= 1
            Movlw 1
            Subwf KONTROLTIPI,W
            set@page bc@ll27
            Btfsc STATUS,0
            Goto bc@ll27
            Movlw 1
            Movwf KONTROLTIPI
bc@ll27
F1_000218 equ $ ; in [SONKOD.BAS] ekran=5
            Movlw 5
            Movwf EKRAN
F1_000219 equ $ ; in [SONKOD.BAS] gosub lcdcikis
            F@Call LCDCIKIS
            F@Jump bc@ll21
F1_000220 equ $ ; in [SONKOD.BAS] elseif durum=3 then
```

```
bc@ll25
        Movlw 3
        Subwf DURUM,W
        set@page bc@ll28
        Btfss STATUS,2
        Goto bc@ll28
F1_000221 equ $ ; in [SONKOD.BAS] parametresec=parametresec+1
        Incf PARAMETRESEC,F
F1_000222 equ $ ; in [SONKOD.BAS] if parametresec>4 then parametresec=1
        Movlw 5
        Subwf PARAMETRESEC,W
        set@page bc@ll30
        Btfss STATUS,0
        Goto bc@ll30
        Movlw 1
        Movwf PARAMETRESEC
bc@ll30
F1_000223 equ $ ; in [SONKOD.BAS] ekran=7
        Movlw 7
        Movwf EKRAN
F1_000224 equ $ ; in [SONKOD.BAS] gosub lcdcikis
        F@Call LCDCIKIS
        F@Jump bc@ll21
F1_000225 equ $ ; in [SONKOD.BAS] elseif durum=4 then
bc@ll28
        Movlw 4
        Subwf DURUM,W
        set@page bc@ll31
        Btfss STATUS,2
        Goto bc@ll31
F1_000226 equ $ ; in [SONKOD.BAS] durum=3
        Movlw 3
        Movwf DURUM
F1_000227 equ $ ; in [SONKOD.BAS] ekran=8
        Movlw 8
        Movwf EKRAN
F1_000228 equ $ ; in [SONKOD.BAS] gosub lcdcikis
        F@Call LCDCIKIS
F1_000229 equ $ ; in [SONKOD.BAS] end if
bc@ll31
bc@ll21
F1_000230 equ $ ; in [SONKOD.BAS] end if
bc@ll18
F1_000234 equ $ ; in [SONKOD.BAS] if butonset=0 then
        set@page bc@ll33
        Btfsc PORTC,1
        Goto bc@ll33
BEKLE3
F1_000236 equ $ ; in [SONKOD.BAS] if butonSET=0 then GOTO BEKLE3
        set@page bc@ll35
        Btfsc PORTC,1
        Goto bc@ll35
        F@Jump BEKLE3
bc@ll35
F1_000237 equ $ ; in [SONKOD.BAS] if durum=255 then
        Incf DURUM,W
        set@page bc@ll37
```

```
            Btfss STATUS,2
            Goto bc@ll37
F1_000238 equ $ ; in [SONKOD.BAS] durum=1
            Movlw 1
            Movwf DURUM
F1_000239 equ $ ; in [SONKOD.BAS] ekran=2
            Movlw 2
            Movwf EKRAN
F1_000240 equ $ ; in [SONKOD.BAS] gosub lcdcikis
            F@Call LCDCIKIS
F1_000241 equ $ ; in [SONKOD.BAS] goto cikis1
            F@Jump CIKIS1
F1_000242 equ $ ; in [SONKOD.BAS] end if
bc@ll37
F1_000243 equ $ ; in [SONKOD.BAS] if durum=1 then
            Movlw 1
            Subwf DURUM,W
            set@page bc@ll39
            Btfss STATUS,2
            Goto bc@ll39
F1_000244 equ $ ; in [SONKOD.BAS] durum=2
            Movlw 2
            Movwf DURUM
F1_000245 equ $ ; in [SONKOD.BAS] ekran=4
            Movlw 4
            Movwf EKRAN
F1_000246 equ $ ; in [SONKOD.BAS] gosub lcdcikis
            F@Call LCDCIKIS
F1_000247 equ $ ; in [SONKOD.BAS] goto cikis1
            F@Jump CIKIS1
F1_000248 equ $ ; in [SONKOD.BAS] end if
bc@ll39
F1_000249 equ $ ; in [SONKOD.BAS] if durum=2 then
            Movlw 2
            Subwf DURUM,W
            set@page bc@ll41
            Btfss STATUS,2
            Goto bc@ll41
F1_000250 equ $ ; in [SONKOD.BAS] durum=3
            Movlw 3
            Movwf DURUM
F1_000251 equ $ ; in [SONKOD.BAS] ekran=6
            Movlw 6
            Movwf EKRAN
F1_000252 equ $ ; in [SONKOD.BAS] gosub lcdcikis
            F@Call LCDCIKIS
F1_000253 equ $ ; in [SONKOD.BAS] goto cikis1
            F@Jump CIKIS1
F1_000254 equ $ ; in [SONKOD.BAS] end if
bc@ll41
F1_000255 equ $ ; in [SONKOD.BAS] if durum=3 then
            Movlw 3
            Subwf DURUM,W
            set@page bc@ll43
            Btfss STATUS,2
            Goto bc@ll43
F1_000256 equ $ ; in [SONKOD.BAS] durum=4
```

```
        Movlw 4
        Movwf DURUM
F1_000257 equ $ ; in [SONKOD.BAS] ekran=8
        Movlw 8
        Movwf EKRAN
F1_000258 equ $ ; in [SONKOD.BAS] gosub lcdcikis
        F@Call LCDCIKIS
F1_000259 equ $ ; in [SONKOD.BAS] KONTROLSTART=1
        Movlw 1
        Movwf KONTROLSTART
F1_000260 equ $ ; in [SONKOD.BAS] goto cikis1
        F@Jump CIKIS1
F1_000261 equ $ ; in [SONKOD.BAS] end if
bc@ll43
CIKIS1
F1_000263 equ $ ; in [SONKOD.BAS] end if
bc@ll33
F1_000267 equ $ ; in [SONKOD.BAS] if butonup=0 then
        set@page bc@ll45
        Btfsc PORTC,3
        Goto bc@ll45
BEKLE2
F1_000269 equ $ ; in [SONKOD.BAS] if butonUP=0 then GOTO BEKLE2
        set@page bc@ll47
        Btfsc PORTC,3
        Goto bc@ll47
        F@Jump BEKLE2
bc@ll47
F1_000270 equ $ ; in [SONKOD.BAS] if durum=1 then
        Movlw 1
        Subwf DURUM,W
        set@page bc@ll49
        Btfss STATUS,2
        Goto bc@ll49
F1_000271 equ $ ; in [SONKOD.BAS] DERECE_SET=DERECE_SET+1
        Incf DERECE_SET,F
F1_000272 equ $ ; in [SONKOD.BAS] if DERECE_SET>75 then DERECE_SET = 75
        Movlw 76
        Subwf DERECE_SET,W
        set@page bc@ll51
        Btfss STATUS,0
        Goto bc@ll51
        Movlw 75
        Movwf DERECE_SET
bc@ll51
F1_000273 equ $ ; in [SONKOD.BAS] ekran=3
        Movlw 3
        Movwf EKRAN
F1_000274 equ $ ; in [SONKOD.BAS] gosub lcdcikis
        F@Call LCDCIKIS
        F@Jump bc@ll48
F1_000275 equ $ ; in [SONKOD.BAS] elseif durum=2 then
bc@ll49
        Movlw 2
        Subwf DURUM,W
        set@page bc@ll52
        Btfss STATUS,2
```

```
        Goto bc@ll52
F1_000276 equ $ ; in [SONKOD.BAS] kontroltipi=kontroltipi+1
        Incf KONTROLTIPI,F
F1_000277 equ $ ; in [SONKOD.BAS] if kontroltipi>5 then kontroltipi= 5
        Movlw 6
        Subwf KONTROLTIPI,W
        set@page bc@ll54
        Btfss STATUS,0
        Goto bc@ll54
        Movlw 5
        Movwf KONTROLTIPI
bc@ll54
F1_000278 equ $ ; in [SONKOD.BAS] ekran=5
        Movlw 5
        Movwf EKRAN
F1_000279 equ $ ; in [SONKOD.BAS] gosub lcdcikis
        F@Call LCDCIKIS
        F@Jump bc@ll48
F1_000280 equ $ ; in [SONKOD.BAS] elseif durum=3 then
bc@ll52
        Movlw 3
        Subwf DURUM,W
        set@page bc@ll55
        Btfss STATUS,2
        Goto bc@ll55
F1_000281 equ $ ; in [SONKOD.BAS] if parametresec=1 then
        Movlw 1
        Subwf PARAMETRESEC,W
        set@page bc@ll57
        Btfss STATUS,2
        Goto bc@ll57
F1_000282 equ $ ; in [SONKOD.BAS] KP_VALUE=KP_VALUE+10
        Movlw 10
        Addwf KP_VALUE,F
        Skpnc
        Incf KP_VALUEH,F
F1_000283 equ $ ; in [SONKOD.BAS] IF KP_VALUE>100 THEN KP_VALUE=10
        Movfw KP_VALUEH
        set@page cp@lb4
        Btfss STATUS,2
        Goto cp@lb4
        Movlw 101
        Subwf KP_VALUE,W
        set@page bc@ll59
        Btfss STATUS,0
        Goto bc@ll59
cp@lb4
        Clrf KP_VALUEH
        Movlw 10
        Movwf KP_VALUE
bc@ll59
        F@Jump bc@ll56
F1_000284 equ $ ; in [SONKOD.BAS] elseif parametresec=2 then
bc@ll57
        Movlw 2
        Subwf PARAMETRESEC,W
        set@page bc@ll60
```

```
        Btfss STATUS,2
        Goto bc@ll60
F1_000285 equ $ ; in [SONKOD.BAS] KI_VALUE=KI_VALUE+1
        Incf KI_VALUE,F
        Skpnz
        Incf KI_VALUEH,F
F1_000286 equ $ ; in [SONKOD.BAS] IF KI_VALUE>10 THEN KI_VALUE=1
        Movfw KI_VALUEH
        set@page cp@lb5
        Btfss STATUS,2
        Goto cp@lb5
        Movlw 11
        Subwf KI_VALUE,W
        set@page bc@ll62
        Btfss STATUS,0
        Goto bc@ll62
cp@lb5
        Clrf KI_VALUEH
        Movlw 1
        Movwf KI_VALUE
bc@ll62
        F@Jump bc@ll56
F1_000287 equ $ ; in [SONKOD.BAS] elseif parametresec=3 then
bc@ll60
        Movlw 3
        Subwf PARAMETRESEC,W
        set@page bc@ll63
        Btfss STATUS,2
        Goto bc@ll63
F1_000288 equ $ ; in [SONKOD.BAS] KD_VALUE=KD_VALUE+1
        Incf KD_VALUE,F
        Skpnz
        Incf KD_VALUEH,F
F1_000289 equ $ ; in [SONKOD.BAS] IF KD_VALUE>10 THEN KD_VALUE=1
        Movfw KD_VALUEH
        set@page cp@lb6
        Btfss STATUS,2
        Goto cp@lb6
        Movlw 11
        Subwf KD_VALUE,W
        set@page bc@ll65
        Btfss STATUS,0
        Goto bc@ll65
cp@lb6
        Clrf KD_VALUEH
        Movlw 1
        Movwf KD_VALUE
bc@ll65
        F@Jump bc@ll56
F1_000290 equ $ ; in [SONKOD.BAS] elseif parametresec=4 then
bc@ll63
        Movlw 4
        Subwf PARAMETRESEC,W
        set@page bc@ll66
        Btfss STATUS,2
        Goto bc@ll66
F1_000291 equ $ ; in [SONKOD.BAS] DA_VALUE=DA_VALUE+1
```

```
            Incf DA_VALUE,F
            Skpnz
            Incf DA_VALUEH,F
F1_000292 equ $ ; in [SONKOD.BAS] IF DA_VALUE>5 THEN DA_VALUE=0
            Movfw DA_VALUEH
            set@page cp@lb7
            Btfss STATUS,2
            Goto cp@lb7
            Movlw 6
            Subwf DA_VALUE,W
            set@page bc@ll68
            Btfss STATUS,0
            Goto bc@ll68
cp@lb7
            Clrf DA_VALUEH
            Clrf DA_VALUE
bc@ll68
F1_000293 equ $ ; in [SONKOD.BAS] end if
bc@ll66
bc@ll56
F1_000294 equ $ ; in [SONKOD.BAS] ekran=7
            Movlw 7
            Movwf EKRAN
F1_000295 equ $ ; in [SONKOD.BAS] gosub lcdcikis
            F@Call LCDCIKIS
F1_000296 equ $ ; in [SONKOD.BAS] end if
bc@ll55
bc@ll48
F1_000297 equ $ ; in [SONKOD.BAS] end if
bc@ll45
F1_000298 equ $ ; in [SONKOD.BAS] return
            Return
LCDCIKIS
F1_000305 equ $ ; in [SONKOD.BAS] if ekran=1 then
            Movlw 1
            Subwf EKRAN,W
            set@page bc@ll70
            Btfss STATUS,2
            Goto bc@ll70
F1_000306 equ $ ; in [SONKOD.BAS] PRINT $FE , 1 : DELAYMS 30
            Movlw 254
            F@Call Print
            Movlw 1
            F@Call Print
            Movlw 30
            F@Call dl@ms
F1_000307 equ $ ; in [SONKOD.BAS] PRINT $FE , 2 : PRINT "SICAKLIK KONTROL"
            Movlw 254
            F@Call Print
            Movlw 2
            F@Call Print
            Movlw 'S'
            F@Call Print
            Movlw 'I'
            F@Call Print
            Movlw 'C'
            F@Call Print
```

```
        Movlw 'A'
        F@Call Print
        Movlw 'K'
        F@Call Print
        Movlw 'L'
        F@Call Print
        Movlw 'I'
        F@Call Print
        Movlw 'K'
        F@Call Print
        Movlw ' '
        F@Call Print
        Movlw 'K'
        F@Call Print
        Movlw 'O'
        F@Call Print
        Movlw 'N'
        F@Call Print
        Movlw 'T'
        F@Call Print
        Movlw 'R'
        F@Call Print
        Movlw 'O'
        F@Call Print
        Movlw 'L'
        F@Call Print
F1_000308 equ $ ; in [SONKOD.BAS] PRINT $FE , $C0  : PRINT " MURAT 6PARMAK"
        Movlw 254
        F@Call Print
        Movlw 192
        F@Call Print
        Movlw ' '
        F@Call Print
        Movlw 'M'
        F@Call Print
        Movlw 'U'
        F@Call Print
        Movlw 'R'
        F@Call Print
        Movlw 'A'
        F@Call Print
        Movlw 'T'
        F@Call Print
        Movlw ' '
        F@Call Print
        Movlw '6'
        F@Call Print
        Movlw 'P'
        F@Call Print
        Movlw 'A'
        F@Call Print
        Movlw 'R'
        F@Call Print
        Movlw 'M'
        F@Call Print
        Movlw 'A'
        F@Call Print
```

```
            Movlw 'K'
            F@Call Print
            F@Jump bc@ll69
F1_000309 equ $ ; in [SONKOD.BAS] elseif ekran=2 then
bc@ll70
            Movlw 2
            Subwf EKRAN,W
            set@page bc@ll71
            Btfss STATUS,2
            Goto bc@ll71
F1_000310 equ $ ; in [SONKOD.BAS] PRINT $FE , 1 : DELAYMS 30
            Movlw 254
            F@Call Print
            Movlw 1
            F@Call Print
            Movlw 30
            F@Call dl@ms
F1_000311 equ $ ; in [SONKOD.BAS] PRINT $FE , 2 : PRINT "  SET DEGERINI"
            Movlw 254
            F@Call Print
            Movlw 2
            F@Call Print
            Movlw ' '
            F@Call Print
            F@Call Print
            Movlw 'S'
            F@Call Print
            Movlw 'E'
            F@Call Print
            Movlw 'T'
            F@Call Print
            Movlw ' '
            F@Call Print
            Movlw 'D'
            F@Call Print
            Movlw 'E'
            F@Call Print
            Movlw 'G'
            F@Call Print
            Movlw 'E'
            F@Call Print
            Movlw 'R'
            F@Call Print
            Movlw 'I'
            F@Call Print
            Movlw 'N'
            F@Call Print
            Movlw 'I'
            F@Call Print
F1_000312 equ $ ; in [SONKOD.BAS] PRINT $FE , $C0  : PRINT "-<  GIRINIZ   >+"
            Movlw 254
            F@Call Print
            Movlw 192
            F@Call Print
            Movlw '-'
            F@Call Print
            Movlw '<'
```

```
            F@Call Print
            Movlw ' '
            F@Call Print
            F@Call Print
            Movlw 'G'
            F@Call Print
            Movlw 'I'
            F@Call Print
            Movlw 'R'
            F@Call Print
            Movlw 'I'
            F@Call Print
            Movlw 'N'
            F@Call Print
            Movlw 'I'
            F@Call Print
            Movlw 'Z'
            F@Call Print
            Movlw ' '
            F@Call Print
            F@Call Print
            F@Call Print
            Movlw '>'
            F@Call Print
            Movlw '+'
            F@Call Print
            F@Jump bc@ll69
F1_000313 equ $ ; in [SONKOD.BAS] elseif ekran=3 then
bc@ll71
            Movlw 3
            Subwf EKRAN,W
            set@page bc@ll72
            Btfss STATUS,2
            Goto bc@ll72
F1_000314 equ $ ; in [SONKOD.BAS] PRINT $FE , 1 : DELAYMS 30
            Movlw 254
            F@Call Print
            Movlw 1
            F@Call Print
            Movlw 30
            F@Call dl@ms
F1_000315 equ $ ; in [SONKOD.BAS] PRINT $FE , 2 : PRINT "    SICAKLIK"
            Movlw 254
            F@Call Print
            Movlw 2
            F@Call Print
            Movlw ' '
            F@Call Print
            F@Call Print
            F@Call Print
            F@Call Print
            Movlw 'S'
            F@Call Print
            Movlw 'I'
            F@Call Print
            Movlw 'C'
            F@Call Print
```

```
        Movlw 'A'
        F@Call Print
        Movlw 'K'
        F@Call Print
        Movlw 'L'
        F@Call Print
        Movlw 'I'
        F@Call Print
        Movlw 'K'
        F@Call Print
F1_000316 equ $ ; in [SONKOD.BAS] PRINT $FE , $C0  : PRINT "       ",DEC DERECE_SET
        Movlw 254
        F@Call Print
        Movlw 192
        F@Call Print
        Movlw ' '
        F@Call Print
        F@Call Print
        F@Call Print
        F@Call Print
        F@Call Print
        F@Call Print
        F@Call Print
        Movlw 128
        Movwf BPFH
        Movfw DERECE_SET
        F@Call out@decb
        F@Jump bc@ll69
F1_000317 equ $ ; in [SONKOD.BAS] elseif ekran=4 then
bc@ll72
        Movlw 4
        Subwf EKRAN,W
        set@page bc@ll73
        Btfss STATUS,2
        Goto bc@ll73
F1_000318 equ $ ; in [SONKOD.BAS] PRINT $FE , 1 : DELAYMS 30
        Movlw 254
        F@Call Print
        Movlw 1
        F@Call Print
        Movlw 30
        F@Call dl@ms
F1_000319 equ $ ; in [SONKOD.BAS] PRINT $FE , 2 : PRINT "    SICAKLIK"
        Movlw 254
        F@Call Print
        Movlw 2
        F@Call Print
        Movlw ' '
        F@Call Print
        F@Call Print
        F@Call Print
        F@Call Print
        Movlw 'S'
        F@Call Print
        Movlw 'I'
        F@Call Print
        Movlw 'C'
```

```
        F@Call Print
        Movlw 'A'
        F@Call Print
        Movlw 'K'
        F@Call Print
        Movlw 'L'
        F@Call Print
        Movlw 'I'
        F@Call Print
        Movlw 'K'
        F@Call Print
F1_000320 equ $ ; in [SONKOD.BAS] PRINT $FE , $C0  : PRINT " SET EDILMISTIR"
        Movlw 254
        F@Call Print
        Movlw 192
        F@Call Print
        Movlw ' '
        F@Call Print
        Movlw 'S'
        F@Call Print
        Movlw 'E'
        F@Call Print
        Movlw 'T'
        F@Call Print
        Movlw ' '
        F@Call Print
        Movlw 'E'
        F@Call Print
        Movlw 'D'
        F@Call Print
        Movlw 'I'
        F@Call Print
        Movlw 'L'
        F@Call Print
        Movlw 'M'
        F@Call Print
        Movlw 'I'
        F@Call Print
        Movlw 'S'
        F@Call Print
        Movlw 'T'
        F@Call Print
        Movlw 'I'
        F@Call Print
        Movlw 'R'
        F@Call Print
F1_000321 equ $ ; in [SONKOD.BAS] delayms 200
        Movlw 200
        F@Call dl@ms
F1_000322 equ $ ; in [SONKOD.BAS] PRINT $FE , 1 : DELAYMS 30
        Movlw 254
        F@Call Print
        Movlw 1
        F@Call Print
        Movlw 30
        F@Call dl@ms
F1_000323 equ $ ; in [SONKOD.BAS] PRINT $FE , 2 : PRINT " KONTROL TIPINI"
```

```
        Movlw 254
        F@Call Print
        Movlw 2
        F@Call Print
        Movlw ' '
        F@Call Print
        Movlw 'K'
        F@Call Print
        Movlw 'O'
        F@Call Print
        Movlw 'N'
        F@Call Print
        Movlw 'T'
        F@Call Print
        Movlw 'R'
        F@Call Print
        Movlw 'O'
        F@Call Print
        Movlw 'L'
        F@Call Print
        Movlw ' '
        F@Call Print
        Movlw 'T'
        F@Call Print
        Movlw 'I'
        F@Call Print
        Movlw 'P'
        F@Call Print
        Movlw 'I'
        F@Call Print
        Movlw 'N'
        F@Call Print
        Movlw 'I'
        F@Call Print
F1_000324 equ $ ; in [SONKOD.BAS] PRINT $FE , $C0  : PRINT "    GIRINIZ"
        Movlw 254
        F@Call Print
        Movlw 192
        F@Call Print
        Movlw ' '
        F@Call Print
        F@Call Print
        F@Call Print
        F@Call Print
        Movlw 'G'
        F@Call Print
        Movlw 'I'
        F@Call Print
        Movlw 'R'
        F@Call Print
        Movlw 'I'
        F@Call Print
        Movlw 'N'
        F@Call Print
        Movlw 'I'
        F@Call Print
        Movlw 'Z'
```

```
            F@Call Print
            F@Jump bc@ll69
F1_000325 equ $ ; in [SONKOD.BAS] elseif ekran=5 then
bc@ll73
            Movlw 5
            Subwf EKRAN,W
            set@page bc@ll74
            Btfss STATUS,2
            Goto bc@ll74
F1_000326 equ $ ; in [SONKOD.BAS] PRINT $FE , 1 : DELAYMS 30
            Movlw 254
            F@Call Print
            Movlw 1
            F@Call Print
            Movlw 30
            F@Call dl@ms
F1_000327 equ $ ; in [SONKOD.BAS] if kontroltipi=1 then
            Movlw 1
            Subwf KONTROLTIPI,W
            set@page bc@ll76
            Btfss STATUS,2
            Goto bc@ll76
F1_000328 equ $ ; in [SONKOD.BAS] PRINT $FE , 2 : PRINT "   ON/OFF"
            Movlw 254
            F@Call Print
            Movlw 2
            F@Call Print
            Movlw ' '
            F@Call Print
            F@Call Print
            F@Call Print
            Movlw 'O'
            F@Call Print
            Movlw 'N'
            F@Call Print
            Movlw '/'
            F@Call Print
            Movlw 'O'
            F@Call Print
            Movlw 'F'
            F@Call Print
            F@Call Print
            F@Jump bc@ll75
F1_000329 equ $ ; in [SONKOD.BAS] elseif kontroltipi=2 then
bc@ll76
            Movlw 2
            Subwf KONTROLTIPI,W
            set@page bc@ll77
            Btfss STATUS,2
            Goto bc@ll77
F1_000330 equ $ ; in [SONKOD.BAS] PRINT $FE , 2 : PRINT "   P"
            Movlw 254
            F@Call Print
            Movlw 2
            F@Call Print
            Movlw ' '
            F@Call Print
```

```
            F@Call Print
            F@Call Print
            Movlw 'P'
            F@Call Print
            F@Jump bc@ll75
F1_000331 equ $ ; in [SONKOD.BAS] elseif kontroltipi=3 then
bc@ll77
            Movlw 3
            Subwf KONTROLTIPI,W
            set@page bc@ll78
            Btfss STATUS,2
            Goto bc@ll78
F1_000332 equ $ ; in [SONKOD.BAS] PRINT $FE , 2 : PRINT "   PI"
            Movlw 254
            F@Call Print
            Movlw 2
            F@Call Print
            Movlw ' '
            F@Call Print
            F@Call Print
            F@Call Print
            Movlw 'P'
            F@Call Print
            Movlw 'I'
            F@Call Print
            F@Jump bc@ll75
F1_000333 equ $ ; in [SONKOD.BAS] elseif kontroltipi=4 then
bc@ll78
            Movlw 4
            Subwf KONTROLTIPI,W
            set@page bc@ll79
            Btfss STATUS,2
            Goto bc@ll79
F1_000334 equ $ ; in [SONKOD.BAS] PRINT $FE , 2 : PRINT "   PD"
            Movlw 254
            F@Call Print
            Movlw 2
            F@Call Print
            Movlw ' '
            F@Call Print
            F@Call Print
            F@Call Print
            Movlw 'P'
            F@Call Print
            Movlw 'D'
            F@Call Print
            F@Jump bc@ll75
F1_000335 equ $ ; in [SONKOD.BAS] elseif kontroltipi=5 then
bc@ll79
            Movlw 5
            Subwf KONTROLTIPI,W
            set@page bc@ll80
            Btfss STATUS,2
            Goto bc@ll80
F1_000336 equ $ ; in [SONKOD.BAS] PRINT $FE , 2 : PRINT "   PID"
            Movlw 254
            F@Call Print
```

```
            Movlw 2
            F@Call Print
            Movlw ' '
            F@Call Print
            F@Call Print
            F@Call Print
            Movlw 'P'
            F@Call Print
            Movlw 'I'
            F@Call Print
            Movlw 'D'
            F@Call Print
F1_000337 equ $ ; in [SONKOD.BAS] end if
bc@ll80
bc@ll75
            F@Jump bc@ll69
F1_000338 equ $ ; in [SONKOD.BAS] elseif ekran=6 then
bc@ll74
            Movlw 6
            Subwf EKRAN,W
            set@page bc@ll81
            Btfss STATUS,2
            Goto bc@ll81
F1_000339 equ $ ; in [SONKOD.BAS] PRINT $FE , 1 : DELAYMS 30
            Movlw 254
            F@Call Print
            Movlw 1
            F@Call Print
            Movlw 30
            F@Call dl@ms
F1_000340 equ $ ; in [SONKOD.BAS] PRINT $FE , 2 : PRINT " KONTROL TIPI"
            Movlw 254
            F@Call Print
            Movlw 2
            F@Call Print
            Movlw ' '
            F@Call Print
            Movlw 'K'
            F@Call Print
            Movlw 'O'
            F@Call Print
            Movlw 'N'
            F@Call Print
            Movlw 'T'
            F@Call Print
            Movlw 'R'
            F@Call Print
            Movlw 'O'
            F@Call Print
            Movlw 'L'
            F@Call Print
            Movlw ' '
            F@Call Print
            Movlw 'T'
            F@Call Print
            Movlw 'I'
            F@Call Print
```

```
        Movlw 'P'
        F@Call Print
        Movlw 'I'
        F@Call Print
F1_000341 equ $ ; in [SONKOD.BAS] PRINT $FE , $C0  : PRINT " SET EDILMISTIR"
        Movlw 254
        F@Call Print
        Movlw 192
        F@Call Print
        Movlw ' '
        F@Call Print
        Movlw 'S'
        F@Call Print
        Movlw 'E'
        F@Call Print
        Movlw 'T'
        F@Call Print
        Movlw ' '
        F@Call Print
        Movlw 'E'
        F@Call Print
        Movlw 'D'
        F@Call Print
        Movlw 'I'
        F@Call Print
        Movlw 'L'
        F@Call Print
        Movlw 'M'
        F@Call Print
        Movlw 'I'
        F@Call Print
        Movlw 'S'
        F@Call Print
        Movlw 'T'
        F@Call Print
        Movlw 'I'
        F@Call Print
        Movlw 'R'
        F@Call Print
F1_000342 equ $ ; in [SONKOD.BAS] delayms 200
        Movlw 200
        F@Call dl@ms
F1_000343 equ $ ; in [SONKOD.BAS] PRINT $FE , 1 : DELAYMS 30
        Movlw 254
        F@Call Print
        Movlw 1
        F@Call Print
        Movlw 30
        F@Call dl@ms
F1_000344 equ $ ; in [SONKOD.BAS] PRINT $FE , 2 : PRINT " PARAMETRELERI"
        Movlw 254
        F@Call Print
        Movlw 2
        F@Call Print
        Movlw ' '
        F@Call Print
        Movlw 'P'
```

```
        F@Call Print
        Movlw 'A'
        F@Call Print
        Movlw 'R'
        F@Call Print
        Movlw 'A'
        F@Call Print
        Movlw 'M'
        F@Call Print
        Movlw 'E'
        F@Call Print
        Movlw 'T'
        F@Call Print
        Movlw 'R'
        F@Call Print
        Movlw 'E'
        F@Call Print
        Movlw 'L'
        F@Call Print
        Movlw 'E'
        F@Call Print
        Movlw 'R'
        F@Call Print
        Movlw 'I'
        F@Call Print
F1_000345 equ $ ; in [SONKOD.BAS] PRINT $FE , $C0  : PRINT "S<  GIRINIZ  >+"
        Movlw 254
        F@Call Print
        Movlw 192
        F@Call Print
        Movlw 'S'
        F@Call Print
        Movlw '<'
        F@Call Print
        Movlw ' '
        F@Call Print
        F@Call Print
        Movlw 'G'
        F@Call Print
        Movlw 'I'
        F@Call Print
        Movlw 'R'
        F@Call Print
        Movlw 'I'
        F@Call Print
        Movlw 'N'
        F@Call Print
        Movlw 'I'
        F@Call Print
        Movlw 'Z'
        F@Call Print
        Movlw ' '
        F@Call Print
        F@Call Print
        F@Call Print
        Movlw '>'
        F@Call Print
```

```
            Movlw '+'
            F@Call Print
            F@Jump bc@ll69
F1_000346 equ $ ; in [SONKOD.BAS] elseif ekran=7 then
bc@ll81
            Movlw 7
            Subwf EKRAN,W
            set@page bc@ll82
            Btfss STATUS,2
            Goto bc@ll82
F1_000347 equ $ ; in [SONKOD.BAS] if parametresec=1 then
            Movlw 1
            Subwf PARAMETRESEC,W
            set@page bc@ll84
            Btfss STATUS,2
            Goto bc@ll84
F1_000348 equ $ ; in [SONKOD.BAS] PRINT $FE , 1 : DELAYMS 30
            Movlw 254
            F@Call Print
            Movlw 1
            F@Call Print
            Movlw 30
            F@Call dl@ms
F1_000349 equ $ ; in [SONKOD.BAS] PRINT $FE , 2 : PRINT "  KP : "
            Movlw 254
            F@Call Print
            Movlw 2
            F@Call Print
            Movlw ' '
            F@Call Print
            F@Call Print
            Movlw 'K'
            F@Call Print
            Movlw 'P'
            F@Call Print
            Movlw ' '
            F@Call Print
            Movlw ':'
            F@Call Print
            Movlw ' '
            F@Call Print
F1_000350 equ $ ; in [SONKOD.BAS] PRINT DEC KP_VALUE
            Movlw 128
            Movwf BPFH
            Clrf GEN4H
            Movfw KP_VALUEH
            Movwf PP2H
            Movfw KP_VALUE
            Movwf PP2
            F@Call out@dec
            F@Jump bc@ll83
F1_000351 equ $ ; in [SONKOD.BAS] elseif parametresec=2 then
bc@ll84
            Movlw 2
            Subwf PARAMETRESEC,W
            set@page bc@ll85
            Btfss STATUS,2
```

```
        Goto bc@ll85
F1_000352 equ $ ; in [SONKOD.BAS] PRINT $FE , 1 : DELAYMS 30
        Movlw 254
        F@Call Print
        Movlw 1
        F@Call Print
        Movlw 30
        F@Call dl@ms
F1_000353 equ $ ; in [SONKOD.BAS] PRINT $FE , 2 : PRINT "  KI : "
        Movlw 254
        F@Call Print
        Movlw 2
        F@Call Print
        Movlw ' '
        F@Call Print
        F@Call Print
        Movlw 'K'
        F@Call Print
        Movlw 'I'
        F@Call Print
        Movlw ' '
        F@Call Print
        Movlw ':'
        F@Call Print
        Movlw ' '
        F@Call Print
F1_000354 equ $ ; in [SONKOD.BAS] IF KI_VALUE = 1 THEN
        Decf KI_VALUE,W
        Iorwf KI_VALUEH,W
        set@page bc@ll87
        Btfss STATUS,2
        Goto bc@ll87
F1_000355 equ $ ; in [SONKOD.BAS] PRINT "0,1"
        Movlw '0'
        F@Call Print
        Movlw ','
        F@Call Print
        Movlw '1'
        F@Call Print
        F@Jump bc@ll86
F1_000356 equ $ ; in [SONKOD.BAS] ELSEIF KI_VALUE = 2 THEN
bc@ll87
        Movfw KI_VALUE
        Xorlw 2
        Iorwf KI_VALUEH,W
        set@page bc@ll88
        Btfss STATUS,2
        Goto bc@ll88
F1_000357 equ $ ; in [SONKOD.BAS] PRINT "0,2"
        Movlw '0'
        F@Call Print
        Movlw ','
        F@Call Print
        Movlw '2'
        F@Call Print
        F@Jump bc@ll86
F1_000358 equ $ ; in [SONKOD.BAS] ELSEIF KI_VALUE = 3 THEN
```

```
bc@ll88
        Movfw KI_VALUE
        Xorlw 3
        Iorwf KI_VALUEH,W
        set@page bc@ll89
        Btfss STATUS,2
        Goto bc@ll89
F1_000359 equ $ ; in [SONKOD.BAS] PRINT "0,3"
        Movlw '0'
        F@Call Print
        Movlw ','
        F@Call Print
        Movlw '3'
        F@Call Print
        F@Jump bc@ll86
F1_000360 equ $ ; in [SONKOD.BAS] ELSEIF KI_VALUE = 4 THEN
bc@ll89
        Movfw KI_VALUE
        Xorlw 4
        Iorwf KI_VALUEH,W
        set@page bc@ll90
        Btfss STATUS,2
        Goto bc@ll90
F1_000361 equ $ ; in [SONKOD.BAS] PRINT "0,4"
        Movlw '0'
        F@Call Print
        Movlw ','
        F@Call Print
        Movlw '4'
        F@Call Print
        F@Jump bc@ll86
F1_000362 equ $ ; in [SONKOD.BAS] ELSEIF KI_VALUE = 5 THEN
bc@ll90
        Movfw KI_VALUE
        Xorlw 5
        Iorwf KI_VALUEH,W
        set@page bc@ll91
        Btfss STATUS,2
        Goto bc@ll91
F1_000363 equ $ ; in [SONKOD.BAS] PRINT "0,5"
        Movlw '0'
        F@Call Print
        Movlw ','
        F@Call Print
        Movlw '5'
        F@Call Print
        F@Jump bc@ll86
F1_000364 equ $ ; in [SONKOD.BAS] ELSEIF KI_VALUE = 6 THEN
bc@ll91
        Movfw KI_VALUE
        Xorlw 6
        Iorwf KI_VALUEH,W
        set@page bc@ll92
        Btfss STATUS,2
        Goto bc@ll92
F1_000365 equ $ ; in [SONKOD.BAS] PRINT "0,6"
        Movlw '0'
```

```
            F@Call Print
            Movlw ','
            F@Call Print
            Movlw '6'
            F@Call Print
            F@Jump bc@ll86
F1_000366 equ $ ; in [SONKOD.BAS] ELSEIF KI_VALUE = 7 THEN
bc@ll92
            Movfw KI_VALUE
            Xorlw 7
            Iorwf KI_VALUEH,W
            set@page bc@ll93
            Btfss STATUS,2
            Goto bc@ll93
F1_000367 equ $ ; in [SONKOD.BAS] PRINT "0,7"
            Movlw '0'
            F@Call Print
            Movlw ','
            F@Call Print
            Movlw '7'
            F@Call Print
            F@Jump bc@ll86
F1_000368 equ $ ; in [SONKOD.BAS] ELSEIF KI_VALUE = 8 THEN
bc@ll93
            Movfw KI_VALUE
            Xorlw 8
            Iorwf KI_VALUEH,W
            set@page bc@ll94
            Btfss STATUS,2
            Goto bc@ll94
F1_000369 equ $ ; in [SONKOD.BAS] PRINT "0,8"
            Movlw '0'
            F@Call Print
            Movlw ','
            F@Call Print
            Movlw '8'
            F@Call Print
            F@Jump bc@ll86
F1_000370 equ $ ; in [SONKOD.BAS] ELSEIF KI_VALUE = 9 THEN
bc@ll94
            Movfw KI_VALUE
            Xorlw 9
            Iorwf KI_VALUEH,W
            set@page bc@ll95
            Btfss STATUS,2
            Goto bc@ll95
F1_000371 equ $ ; in [SONKOD.BAS] PRINT "0,9"
            Movlw '0'
            F@Call Print
            Movlw ','
            F@Call Print
            Movlw '9'
            F@Call Print
            F@Jump bc@ll86
F1_000372 equ $ ; in [SONKOD.BAS] ELSEIF KI_VALUE = 10 THEN
bc@ll95
            Movfw KI_VALUE
```

```
        Xorlw 10
        Iorwf KI_VALUEH,W
        set@page bc@ll96
        Btfss STATUS,2
        Goto bc@ll96
F1_000373 equ $ ; in [SONKOD.BAS] PRINT "1"
        Movlw '1'
        F@Call Print
F1_000374 equ $ ; in [SONKOD.BAS] END IF
bc@ll96
bc@ll86
        F@Jump bc@ll83
F1_000375 equ $ ; in [SONKOD.BAS] elseif parametresec=3 then
bc@ll85
        Movlw 3
        Subwf PARAMETRESEC,W
        set@page bc@ll97
        Btfss STATUS,2
        Goto bc@ll97
F1_000376 equ $ ; in [SONKOD.BAS] PRINT $FE , 1 : DELAYMS 30
        Movlw 254
        F@Call Print
        Movlw 1
        F@Call Print
        Movlw 30
        F@Call dl@ms
F1_000377 equ $ ; in [SONKOD.BAS] PRINT $FE , 2 : PRINT "  KD : "
        Movlw 254
        F@Call Print
        Movlw 2
        F@Call Print
        Movlw ' '
        F@Call Print
        F@Call Print
        Movlw 'K'
        F@Call Print
        Movlw 'D'
        F@Call Print
        Movlw ' '
        F@Call Print
        Movlw ':'
        F@Call Print
        Movlw ' '
        F@Call Print
F1_000378 equ $ ; in [SONKOD.BAS] PRINT DEC KD_VALUE
        Movlw 128
        Movwf BPFH
        Clrf GEN4H
        Movfw KD_VALUEH
        Movwf PP2H
        Movfw KD_VALUE
        Movwf PP2
        F@Call out@dec
        F@Jump bc@ll83
F1_000379 equ $ ; in [SONKOD.BAS] elseif parametresec=4 then
bc@ll97
        Movlw 4
```

```
        Subwf PARAMETRESEC,W
        set@page bc@ll98
        Btfss STATUS,2
        Goto bc@ll98
F1_000380 equ $ ; in [SONKOD.BAS] PRINT $FE , 1 : DELAYMS 30
        Movlw 254
        F@Call Print
        Movlw 1
        F@Call Print
        Movlw 30
        F@Call dl@ms
F1_000381 equ $ ; in [SONKOD.BAS] PRINT $FE , 2 : PRINT "  DA : "
        Movlw 254
        F@Call Print
        Movlw 2
        F@Call Print
        Movlw ' '
        F@Call Print
        F@Call Print
        Movlw 'D'
        F@Call Print
        Movlw 'A'
        F@Call Print
        Movlw ' '
        F@Call Print
        Movlw ':'
        F@Call Print
        Movlw ' '
        F@Call Print
F1_000382 equ $ ; in [SONKOD.BAS] PRINT DEC DA_VALUE
        Movlw 128
        Movwf BPFH
        Clrf GEN4H
        Movfw DA_VALUEH
        Movwf PP2H
        Movfw DA_VALUE
        Movwf PP2
        F@Call out@dec
F1_000383 equ $ ; in [SONKOD.BAS] end if
bc@ll98
bc@ll83
        F@Jump bc@ll69
F1_000384 equ $ ; in [SONKOD.BAS] elseif ekran=8 then
bc@ll82
        Movlw 8
        Subwf EKRAN,W
        set@page bc@ll99
        Btfss STATUS,2
        Goto bc@ll99
F1_000385 equ $ ; in [SONKOD.BAS] PRINT $FE , 1 : DELAYMS 30
        Movlw 254
        F@Call Print
        Movlw 1
        F@Call Print
        Movlw 30
        F@Call dl@ms
F1_000386 equ $ ; in [SONKOD.BAS] PRINT $FE , 2 : PRINT "  PARAMETRELER"
```

```
        Movlw 254
        F@Call Print
        Movlw 2
        F@Call Print
        Movlw ' '
        F@Call Print
        F@Call Print
        Movlw 'P'
        F@Call Print
        Movlw 'A'
        F@Call Print
        Movlw 'R'
        F@Call Print
        Movlw 'A'
        F@Call Print
        Movlw 'M'
        F@Call Print
        Movlw 'E'
        F@Call Print
        Movlw 'T'
        F@Call Print
        Movlw 'R'
        F@Call Print
        Movlw 'E'
        F@Call Print
        Movlw 'L'
        F@Call Print
        Movlw 'E'
        F@Call Print
        Movlw 'R'
        F@Call Print
F1_000387 equ $ ; in [SONKOD.BAS] PRINT $FE , $C0  : PRINT " SET EDILMISTIR"
        Movlw 254
        F@Call Print
        Movlw 192
        F@Call Print
        Movlw ' '
        F@Call Print
        Movlw 'S'
        F@Call Print
        Movlw 'E'
        F@Call Print
        Movlw 'T'
        F@Call Print
        Movlw ' '
        F@Call Print
        Movlw 'E'
        F@Call Print
        Movlw 'D'
        F@Call Print
        Movlw 'I'
        F@Call Print
        Movlw 'L'
        F@Call Print
        Movlw 'M'
        F@Call Print
        Movlw 'I'
```

```
            F@Call Print
            Movlw 'S'
            F@Call Print
            Movlw 'T'
            F@Call Print
            Movlw 'I'
            F@Call Print
            Movlw 'R'
            F@Call Print
F1_000388 equ $ ; in [SONKOD.BAS] DELAYMS 1000
            Movlw 3
            Movwf PP1H
            Movlw 232
            F@Call dly@w
F1_000389 equ $ ; in [SONKOD.BAS] PRINT $FE , 1 : DELAYMS 30
            Movlw 254
            F@Call Print
            Movlw 1
            F@Call Print
            Movlw 30
            F@Call dl@ms
F1_000390 equ $ ; in [SONKOD.BAS] PRINT $FE , 2 : PRINT "    KONTROL"
            Movlw 254
            F@Call Print
            Movlw 2
            F@Call Print
            Movlw ' '
            F@Call Print
            F@Call Print
            F@Call Print
            F@Call Print
            Movlw 'K'
            F@Call Print
            Movlw 'O'
            F@Call Print
            Movlw 'N'
            F@Call Print
            Movlw 'T'
            F@Call Print
            Movlw 'R'
            F@Call Print
            Movlw 'O'
            F@Call Print
            Movlw 'L'
            F@Call Print
F1_000391 equ $ ; in [SONKOD.BAS] PRINT $FE , $C0  : PRINT "  BASLAMISTIR"
            Movlw 254
            F@Call Print
            Movlw 192
            F@Call Print
            Movlw ' '
            F@Call Print
            F@Call Print
            Movlw 'B'
            F@Call Print
            Movlw 'A'
            F@Call Print
```

```
            Movlw 'S'
            F@Call Print
            Movlw 'L'
            F@Call Print
            Movlw 'A'
            F@Call Print
            Movlw 'M'
            F@Call Print
            Movlw 'I'
            F@Call Print
            Movlw 'S'
            F@Call Print
            Movlw 'T'
            F@Call Print
            Movlw 'I'
            F@Call Print
            Movlw 'R'
            F@Call Print
F1_000392 equ $ ; in [SONKOD.BAS] end if
bc@ll99
bc@ll69
F1_000393 equ $ ; in [SONKOD.BAS] return
            Return
DONGU
F1_000399 equ $ ; in [SONKOD.BAS] IF KONTROLSTART = 1 THEN
            Movlw 1
            Subwf KONTROLSTART,W
            set@page bc@ll101
            Btfss STATUS,2
            Goto bc@ll101
F1_000400 equ $ ; in [SONKOD.BAS] GOSUB KONTROL
            F@Call KONTROL
F1_000401 equ $ ; in [SONKOD.BAS] IF KONTROLSTART = 0 THEN
            Movfw KONTROLSTART
            set@page bc@ll103
            Btfss STATUS,2
            Goto bc@ll103
F1_000402 equ $ ; in [SONKOD.BAS] GOTO CKS100
            F@Jump CKS100
F1_000403 equ $ ; in [SONKOD.BAS] END IF
bc@ll103
F1_000404 equ $ ; in [SONKOD.BAS] gosub sicaklikdongusu
            F@Call SICAKLIKDONGUSU
F1_000405 equ $ ; in [SONKOD.BAS] PRINT $FE , $C0 ' :PRINT "KONTROL BASLADI"
            Movlw 254
            F@Call Print
            Movlw 192
            F@Call Print
F1_000406 equ $ ; in [SONKOD.BAS] GOTO DONGU
            F@Jump DONGU
F1_000407 equ $ ; in [SONKOD.BAS] END IF
bc@ll101
CKS100
F1_000409 equ $ ; in [SONKOD.BAS] IF DURUM=255 THEN
            Incf DURUM,W
            set@page bc@ll105
            Btfss STATUS,2
```

```
        Goto bc@ll105
F1_000410 equ $ ; in [SONKOD.BAS] gosub sicaklikdongusu
        F@Call SICAKLIKDONGUSU
F1_000411 equ $ ; in [SONKOD.BAS] END IF
bc@ll105
F1_000413 equ $ ; in [SONKOD.BAS] gosub butonkontrol
        F@Call BUTONKONTROL
F1_000415 equ $ ; in [SONKOD.BAS] GOTO DONGU
        F@Jump DONGU
KONTROL
F1_000422 equ $ ; in [SONKOD.BAS] if kontroltipi=1 then
        Movlw 1
        Subwf KONTROLTIPI,W
        set@page bc@ll107
        Btfss STATUS,2
        Goto bc@ll107
F1_000424 equ $ ; in [SONKOD.BAS] DERECE_UST=DERECE_SET+DA_VALUE
        Movfw DERECE_SET
        Addwf DA_VALUE,W
        Movwf DERECE_UST
F1_000425 equ $ ; in [SONKOD.BAS] DERECE_UST = DERECE_UST - 1
        Decf DERECE_UST,F
F1_000426 equ $ ; in [SONKOD.BAS] DERECE_ALT=DERECE_SET-DA_VALUE
        Movfw DA_VALUE
        Subwf DERECE_SET,W
        Movwf DERECE_ALT
F1_000427 equ $ ; in [SONKOD.BAS] IF TEMPORT > DERECE_UST THEN
        Movfw TEMPORTH
        set@page cp@lb18
        Btfss STATUS,2
        Goto cp@lb18
        Movfw TEMPORT
        Subwf DERECE_UST,W
        set@page bc@ll109
        Btfsc STATUS,0
        Goto bc@ll109
cp@lb18
F1_000428 equ $ ; in [SONKOD.BAS] ONOFFSAYAC=ONOFFSAYAC+1
        Incf ONOFFSAYAC,F
F1_000429 equ $ ; in [SONKOD.BAS] Duty = 0 : Ccp1con.4 = Duty.0 : Ccp1con.5 = Duty.1 : Ccpr1l
= Duty >> 2
        Clrf DUTYH
        Clrf DUTY
; Bit_Bit DUTY,0,CCP1CON,4
        Bsf CCP1CON,4
        Btfss DUTY,0
        Bcf CCP1CON,4
; Bit_Bit DUTY,1,CCP1CON,5
        Bsf CCP1CON,5
        Btfss DUTY,1
        Bcf CCP1CON,5
        Rrf DUTYH,W
        Movwf PP4
        Rrf DUTY,W
        Movwf CCPR1L
        Rrf PP4,F
        Rrf CCPR1L,F
```

```
        Movfw DUTYH
        Movwf PP0H
        Movfw DUTY
        Movwf PP0
        Movlw 2
        F@Call r@sh
        Movwf CCPR1L
F1_000430 equ $ ; in [SONKOD.BAS] GOTO KONTROL_CIK
        F@Jump KONTROL_CIK
        F@Jump bc@ll108
F1_000431 equ $ ; in [SONKOD.BAS] ELSEIF TEMPORT < DERECE_ALT THEN
bc@ll109
        Movfw TEMPORTH
        set@page bc@ll110
        Btfss STATUS,2
        Goto bc@ll110
        Movfw DERECE_ALT
        Subwf TEMPORT,W
        set@page bc@ll110
        Btfsc STATUS,0
        Goto bc@ll110
F1_000432 equ $ ; in [SONKOD.BAS] Duty = 950 : Ccp1con.4 = Duty.0 : Ccp1con.5 = Duty.1 :
Ccpr1l = Duty >> 2
        Movlw 3
        Movwf DUTYH
        Movlw 182
        Movwf DUTY
; Bit_Bit DUTY,0,CCP1CON,4
        Bsf CCP1CON,4
        Btfss DUTY,0
        Bcf CCP1CON,4
; Bit_Bit DUTY,1,CCP1CON,5
        Bsf CCP1CON,5
        Btfss DUTY,1
        Bcf CCP1CON,5
        Rrf DUTYH,W
        Movwf PP4
        Rrf DUTY,W
        Movwf CCPR1L
        Rrf PP4,F
        Rrf CCPR1L,F
        Movfw DUTYH
        Movwf PP0H
        Movfw DUTY
        Movwf PP0
        Movlw 2
        F@Call r@sh
        Movwf CCPR1L
F1_000433 equ $ ; in [SONKOD.BAS] GOTO KONTROL_CIK
        F@Jump KONTROL_CIK
F1_000434 equ $ ; in [SONKOD.BAS] END IF
bc@ll110
bc@ll108
        F@Jump bc@ll106
F1_000438 equ $ ; in [SONKOD.BAS] elseif kontroltipi=2 then
bc@ll107
        Movlw 2
```

```
        Subwf KONTROLTIPI,W
        set@page bc@ll111
        Btfss STATUS,2
        Goto bc@ll111
F1_000440 equ $ ; in [SONKOD.BAS] IF DERECE_SET >= TEMPORT THEN
        Movfw TEMPORTH
        set@page bc@ll113
        Btfss STATUS,2
        Goto bc@ll113
        Movfw TEMPORT
        Subwf DERECE_SET,W
        set@page bc@ll113
        Btfss STATUS,0
        Goto bc@ll113
F1_000441 equ $ ; in [SONKOD.BAS] FARK=DERECE_SET-TEMPORT
        Movfw TEMPORT
        Subwf DERECE_SET,W
        Movwf FARK
        Comf TEMPORTH,W
        Skpnc
        Addlw 1
        Movwf FARKH
F1_000442 equ $ ; in [SONKOD.BAS] DUTY = KP_VALUE*FARK
        Movfw KP_VALUEH
        Movwf PP3H
        Movfw KP_VALUE
        Movwf PP3
        Movfw FARKH
        Movwf PP1H
        Movfw FARK
        Movwf PP1
        F@Call m@py
        Movwf DUTY
        Movfw PP2H
        Movwf DUTYH
        F@Jump bc@ll112
F1_000443 equ $ ; in [SONKOD.BAS] ELSEIF  DERECE_SET < TEMPORT THEN
bc@ll113
        Movfw TEMPORTH
        set@page cp@lb21
        Btfss STATUS,2
        Goto cp@lb21
        Movfw TEMPORT
        Subwf DERECE_SET,W
        set@page bc@ll114
        Btfsc STATUS,0
        Goto bc@ll114
cp@lb21
F1_000444 equ $ ; in [SONKOD.BAS] DUTY = 0
        Clrf DUTYH
        Clrf DUTY
F1_000445 equ $ ; in [SONKOD.BAS] ENDIF
bc@ll114
bc@ll112
F1_000447 equ $ ; in [SONKOD.BAS] IF DUTY > 950 THEN
        Movlw 3
        Subwf DUTYH,W
```

```
            set@page bc@ll116
            Btfss STATUS,0
            Goto bc@ll116
            set@page cp@lb22
            Btfss STATUS,2
            Goto cp@lb22
            Movlw 183
            Subwf DUTY,W
            set@page bc@ll116
            Btfss STATUS,0
            Goto bc@ll116
cp@lb22
F1_000448 equ $ ; in [SONKOD.BAS] DUTY = 950
            Movlw 3
            Movwf DUTYH
            Movlw 182
            Movwf DUTY
F1_000449 equ $ ; in [SONKOD.BAS] END IF
bc@ll116
F1_000450 equ $ ; in [SONKOD.BAS] Ccp1con.4 = Duty.0 : Ccp1con.5 = Duty.1 : Ccpr1l = Duty >>
2
; Bit_Bit DUTY,0,CCP1CON,4
            Bsf CCP1CON,4
            Btfss DUTY,0
            Bcf CCP1CON,4
; Bit_Bit DUTY,1,CCP1CON,5
            Bsf CCP1CON,5
            Btfss DUTY,1
            Bcf CCP1CON,5
            Rrf DUTYH,W
            Movwf PP4
            Rrf DUTY,W
            Movwf CCPR1L
            Rrf PP4,F
            Rrf CCPR1L,F
            Movfw DUTYH
            Movwf PP0H
            Movfw DUTY
            Movwf PP0
            Movlw 2
            F@Call r@sh
            Movwf CCPR1L
F1_000451 equ $ ; in [SONKOD.BAS] GOTO KONTROL_CIK
            F@Jump KONTROL_CIK
            F@Jump bc@ll106
F1_000456 equ $ ; in [SONKOD.BAS] elseif kontroltipi=3 then
bc@ll111
            Movlw 3
            Subwf KONTROLTIPI,W
            set@page bc@ll117
            Btfss STATUS,2
            Goto bc@ll117
KONTROL_PI
F1_000459 equ $ ; in [SONKOD.BAS] IF DERECE_SET >= TEMPORT THEN
            Movfw TEMPORTH
            set@page bc@ll119
            Btfss STATUS,2
```

```
            Goto bc@ll119
            Movfw TEMPORT
            Subwf DERECE_SET,W
            set@page bc@ll119
            Btfss STATUS,0
            Goto bc@ll119
F1_000460 equ $ ; in [SONKOD.BAS] FARK_E=DERECE_SET-TEMPORT
            Movfw TEMPORT
            Subwf DERECE_SET,W
            Movwf FARK_E
            Comf TEMPORTH,W
            Skpnc
            Addlw 1
            Movwf FARK_EH
F1_000461 equ $ ; in [SONKOD.BAS] IF FARK_E<10 THEN
            Movfw FARK_EH
            set@page bc@ll121
            Btfss STATUS,2
            Goto bc@ll121
            Movlw 10
            Subwf FARK_E,W
            set@page bc@ll121
            Btfsc STATUS,0
            Goto bc@ll121
F1_000462 equ $ ; in [SONKOD.BAS] F_INT=FARK_E+F_INT
            Movfw FARK_E
            Addwf F_INT,F
            Movfw FARK_EH
            Skpnc
            Addlw 1
            Addwf F_INTH,F
F1_000463 equ $ ; in [SONKOD.BAS] END IF
bc@ll121
F1_000464 equ $ ; in [SONKOD.BAS] END IF
bc@ll119
F1_000465 equ $ ; in [SONKOD.BAS] IF DERECE_SET < TEMPORT THEN
            Movfw TEMPORTH
            set@page cp@lb25
            Btfss STATUS,2
            Goto cp@lb25
            Movfw TEMPORT
            Subwf DERECE_SET,W
            set@page bc@ll123
            Btfsc STATUS,0
            Goto bc@ll123
cp@lb25
F1_000466 equ $ ; in [SONKOD.BAS] FARK_E=TEMPORT-DERECE_SET
            Movfw DERECE_SET
            Subwf TEMPORT,W
            Movwf FARK_E
            Movfw TEMPORTH
            Skpc
            Addlw 255
            Movwf FARK_EH
F1_000467 equ $ ; in [SONKOD.BAS] IF F_INT>FARK_E THEN
            Movfw F_INTH
            Subwf FARK_EH,W
```

```
            set@page cp@lb26
            Btfss STATUS,0
            Goto cp@lb26
            set@page bc@ll125
            Btfss STATUS,2
            Goto bc@ll125
            Movfw F_INT
            Subwf FARK_E,W
            set@page bc@ll125
            Btfsc STATUS,0
            Goto bc@ll125
cp@lb26
F1_000468 equ $ ; in [SONKOD.BAS] F_INT=F_INT-FARK_E
            Movfw FARK_E
            Subwf F_INT,F
            Movfw FARK_EH
            Skpc
            Addlw 1
            Subwf F_INTH,F
            F@Jump bc@ll126
bc@ll125
F1_000469 equ $ ; in [SONKOD.BAS] ELSE
F1_000470 equ $ ; in [SONKOD.BAS] F_INT=0
            Clrf F_INTH
            Clrf F_INT
F1_000471 equ $ ; in [SONKOD.BAS] END IF
bc@ll126
F1_000472 equ $ ; in [SONKOD.BAS] END IF
bc@ll123
F1_000473 equ $ ; in [SONKOD.BAS] F_INT_TOPL=0
            Clrf F_INT_TOPLH
            Clrf F_INT_TOPL
F1_000474 equ $ ; in [SONKOD.BAS] FOR i = 0 TO KI_VALUE
            Clrf _I
fr@lb128
            Movfw KI_VALUEH
            set@page cp@lb27
            Btfss STATUS,2
            Goto cp@lb27
            Movfw _I
            Subwf KI_VALUE,W
            set@page nx@lb129
            Btfss STATUS,0
            Goto nx@lb129
cp@lb27
F1_000475 equ $ ; in [SONKOD.BAS] F_INT_TOPL= F_INT + F_INT_TOPL
            Movfw F_INT
            Addwf F_INT_TOPL,F
            Movfw F_INTH
            Skpnc
            Addlw 1
            Addwf F_INT_TOPLH,F
F1_000476 equ $ ; in [SONKOD.BAS] NEXT
            Incf _I,F
            set@page fr@lb128
            Btfss STATUS,2
            Goto fr@lb128
```

```
nx@lb129
F1_000477 equ $ ; in [SONKOD.BAS] PRINT DEC F_INT_TOPL
        Movlw 128
        Movwf BPFH
        Clrf GEN4H
        Movfw F_INT_TOPLH
        Movwf PP2H
        Movfw F_INT_TOPL
        Movwf PP2
        F@Call out@dec
F1_000478 equ $ ; in [SONKOD.BAS] PRINT " "
        Movlw ' '
        F@Call Print
F1_000479 equ $ ; in [SONKOD.BAS] IF DERECE_SET >= TEMPORT THEN
        Movfw TEMPORTH
        set@page bc@ll130
        Btfss STATUS,2
        Goto bc@ll130
        Movfw TEMPORT
        Subwf DERECE_SET,W
        set@page bc@ll130
        Btfss STATUS,0
        Goto bc@ll130
F1_000480 equ $ ; in [SONKOD.BAS] FARK = DERECE_SET-TEMPORT
        Movfw TEMPORT
        Subwf DERECE_SET,W
        Movwf FARK
        Comf TEMPORTH,W
        Skpnc
        Addlw 1
        Movwf FARKH
F1_000481 equ $ ; in [SONKOD.BAS] DUTY = KP_VALUE*FARK
        Movfw KP_VALUEH
        Movwf PP3H
        Movfw KP_VALUE
        Movwf PP3
        Movfw FARKH
        Movwf PP1H
        Movfw FARK
        Movwf PP1
        F@Call m@py
        Movwf DUTY
        Movfw PP2H
        Movwf DUTYH
        F@Jump bc@ll129
F1_000482 equ $ ; in [SONKOD.BAS] ELSEIF  DERECE_SET < TEMPORT THEN
bc@ll130
        Movfw TEMPORTH
        set@page cp@lb29
        Btfss STATUS,2
        Goto cp@lb29
        Movfw TEMPORT
        Subwf DERECE_SET,W
        set@page bc@ll131
        Btfsc STATUS,0
        Goto bc@ll131
cp@lb29
```

```
F1_000483 equ $ ; in [SONKOD.BAS] DUTY = 0
        Clrf DUTYH
        Clrf DUTY
F1_000484 equ $ ; in [SONKOD.BAS] ENDIF
bc@ll131
bc@ll129
F1_000486 equ $ ; in [SONKOD.BAS] DUTY = DUTY + F_INT_TOPL
        Movfw F_INT_TOPL
        Addwf DUTY,F
        Movfw F_INT_TOPLH
        Skpnc
        Addlw 1
        Addwf DUTYH,F
F1_000488 equ $ ; in [SONKOD.BAS] IF DUTY > 950 THEN
        Movlw 3
        Subwf DUTYH,W
        set@page bc@ll133
        Btfss STATUS,0
        Goto bc@ll133
        set@page cp@lb30
        Btfss STATUS,2
        Goto cp@lb30
        Movlw 183
        Subwf DUTY,W
        set@page bc@ll133
        Btfss STATUS,0
        Goto bc@ll133
cp@lb30
F1_000489 equ $ ; in [SONKOD.BAS] DUTY = 950
        Movlw 3
        Movwf DUTYH
        Movlw 182
        Movwf DUTY
F1_000490 equ $ ; in [SONKOD.BAS] END IF
bc@ll133
F1_000492 equ $ ; in [SONKOD.BAS] PRINT DEC DUTY
        Movlw 128
        Movwf BPFH
        Clrf GEN4H
        Movfw DUTYH
        Movwf PP2H
        Movfw DUTY
        Movwf PP2
        F@Call out@dec
F1_000493 equ $ ; in [SONKOD.BAS] Ccp1con.4 = Duty.0 : Ccp1con.5 = Duty.1 : Ccpr1l = Duty >>
2
; Bit_Bit DUTY,0,CCP1CON,4
        Bsf CCP1CON,4
        Btfss DUTY,0
        Bcf CCP1CON,4
; Bit_Bit DUTY,1,CCP1CON,5
        Bsf CCP1CON,5
        Btfss DUTY,1
        Bcf CCP1CON,5
        Rrf DUTYH,W
        Movwf PP4
        Rrf DUTY,W
```

```
        Movwf CCPR1L
        Rrf PP4,F
        Rrf CCPR1L,F
        Movfw DUTYH
        Movwf PP0H
        Movfw DUTY
        Movwf PP0
        Movlw 2
        F@Call r@sh
        Movwf CCPR1L
F1_000494 equ $ ; in [SONKOD.BAS] GOTO KONTROL_CIK
        F@Jump KONTROL_CIK
PI_END
        F@Jump bc@ll106
F1_000500 equ $ ; in [SONKOD.BAS] elseif kontroltipi=4 then
bc@ll117
        Movlw 4
        Subwf KONTROLTIPI,W
        set@page bc@ll134
        Btfss STATUS,2
        Goto bc@ll134
KONTROL_PD
F1_000504 equ $ ; in [SONKOD.BAS] IF DERECE_SET >= TEMPORT THEN
        Movfw TEMPORTH
        set@page bc@ll136
        Btfss STATUS,2
        Goto bc@ll136
        Movfw TEMPORT
        Subwf DERECE_SET,W
        set@page bc@ll136
        Btfss STATUS,0
        Goto bc@ll136
F1_000505 equ $ ; in [SONKOD.BAS] FARK = DERECE_SET-TEMPORT
        Movfw TEMPORT
        Subwf DERECE_SET,W
        Movwf FARK
        Comf TEMPORTH,W
        Skpnc
        Addlw 1
        Movwf FARKH
F1_000506 equ $ ; in [SONKOD.BAS] DUTY = KP_VALUE*FARK
        Movfw KP_VALUEH
        Movwf PP3H
        Movfw KP_VALUE
        Movwf PP3
        Movfw FARKH
        Movwf PP1H
        Movfw FARK
        Movwf PP1
        F@Call m@py
        Movwf DUTY
        Movfw PP2H
        Movwf DUTYH
        F@Jump bc@ll135
F1_000507 equ $ ; in [SONKOD.BAS] ELSEIF  DERECE_SET < TEMPORT THEN
bc@ll136
        Movfw TEMPORTH
```

```
            set@page cp@lb32
            Btfss STATUS,2
            Goto cp@lb32
            Movfw TEMPORT
            Subwf DERECE_SET,W
            set@page bc@ll137
            Btfsc STATUS,0
            Goto bc@ll137
cp@lb32
F1_000508 equ $ ; in [SONKOD.BAS] DUTY = 0
            Clrf DUTYH
            Clrf DUTY
F1_000509 equ $ ; in [SONKOD.BAS] ENDIF
bc@ll137
bc@ll135
F1_000511 equ $ ; in [SONKOD.BAS] ED2=TEMPORT
            Movfw TEMPORT
            Movwf ED2
F1_000512 equ $ ; in [SONKOD.BAS] IF ED2>=ED1 THEN
            Movfw ED1
            Subwf ED2,W
            set@page bc@ll139
            Btfss STATUS,0
            Goto bc@ll139
F1_000513 equ $ ; in [SONKOD.BAS] FARK_D=ED2-ED1
            Movfw ED1
            Subwf ED2,W
            Movwf FARK_D
            Clrf FARK_DH
            Skpc
            Decf FARK_DH,F
F1_000514 equ $ ; in [SONKOD.BAS] F_DER_TOPL=0
            Clrf F_DER_TOPLH
            Clrf F_DER_TOPL
F1_000515 equ $ ; in [SONKOD.BAS] FOR i = 0 TO KD_VALUE
            Clrf _I
fr@lb141
            Movfw KD_VALUEH
            set@page cp@lb33
            Btfss STATUS,2
            Goto cp@lb33
            Movfw _I
            Subwf KD_VALUE,W
            set@page nx@lb142
            Btfss STATUS,0
            Goto nx@lb142
cp@lb33
F1_000516 equ $ ; in [SONKOD.BAS] F_DER_TOPL= F_DER + F_DER_TOPL
            Movfw F_DER
            Addwf F_DER_TOPL,F
            Movfw F_DERH
            Skpnc
            Addlw 1
            Addwf F_DER_TOPLH,F
F1_000517 equ $ ; in [SONKOD.BAS] NEXT
            Incf _I,F
            set@page fr@lb141
```

```
            Btfss STATUS,2
            Goto fr@lb141
nx@lb142
F1_000518 equ $ ; in [SONKOD.BAS] DUTY = DUTY + F_DER_TOPL
            Movfw F_DER_TOPL
            Addwf DUTY,F
            Movfw F_DER_TOPLH
            Skpnc
            Addlw 1
            Addwf DUTYH,F
            F@Jump bc@ll138
F1_000519 equ $ ; in [SONKOD.BAS] ELSEIF ED1>ED2 THEN
bc@ll139
            Movfw ED1
            Subwf ED2,W
            set@page bc@ll142
            Btfsc STATUS,0
            Goto bc@ll142
F1_000520 equ $ ; in [SONKOD.BAS] FARK_D=ED1-ED2
            Movfw ED2
            Subwf ED1,W
            Movwf FARK_D
            Clrf FARK_DH
            Skpc
            Decf FARK_DH,F
F1_000521 equ $ ; in [SONKOD.BAS] F_DER_TOPL= 0
            Clrf F_DER_TOPLH
            Clrf F_DER_TOPL
F1_000522 equ $ ; in [SONKOD.BAS] FOR i = 0 TO KD_VALUE
            Clrf _I
fr@lb144
            Movfw KD_VALUEH
            set@page cp@lb34
            Btfss STATUS,2
            Goto cp@lb34
            Movfw _I
            Subwf KD_VALUE,W
            set@page nx@lb145
            Btfss STATUS,0
            Goto nx@lb145
cp@lb34
F1_000523 equ $ ; in [SONKOD.BAS] F_DER_TOPL= F_DER + F_DER_TOPL
            Movfw F_DER
            Addwf F_DER_TOPL,F
            Movfw F_DERH
            Skpnc
            Addlw 1
            Addwf F_DER_TOPLH,F
F1_000524 equ $ ; in [SONKOD.BAS] NEXT
            Incf _I,F
            set@page fr@lb144
            Btfss STATUS,2
            Goto fr@lb144
nx@lb145
F1_000525 equ $ ; in [SONKOD.BAS] DUTY = DUTY - F_DER_TOPL
            Movfw F_DER_TOPL
            Subwf DUTY,F
```

```
            Movfw F_DER_TOPLH
            Skpc
            Addlw 1
            Subwf DUTYH,F
F1_000526 equ $ ; in [SONKOD.BAS] END IF
bc@ll142
bc@ll138
F1_000527 equ $ ; in [SONKOD.BAS] ED1=ED2
            Movfw ED2
            Movwf ED1
F1_000529 equ $ ; in [SONKOD.BAS] IF DUTY > 950 THEN
            Movlw 3
            Subwf DUTYH,W
            set@page bc@ll146
            Btfss STATUS,0
            Goto bc@ll146
            set@page cp@lb35
            Btfss STATUS,2
            Goto cp@lb35
            Movlw 183
            Subwf DUTY,W
            set@page bc@ll146
            Btfss STATUS,0
            Goto bc@ll146
cp@lb35
F1_000530 equ $ ; in [SONKOD.BAS] DUTY = 950
            Movlw 3
            Movwf DUTYH
            Movlw 182
            Movwf DUTY
F1_000531 equ $ ; in [SONKOD.BAS] END IF
bc@ll146
F1_000532 equ $ ; in [SONKOD.BAS] PRINT DEC DUTY
            Movlw 128
            Movwf BPFH
            Clrf GEN4H
            Movfw DUTYH
            Movwf PP2H
            Movfw DUTY
            Movwf PP2
            F@Call out@dec
F1_000533 equ $ ; in [SONKOD.BAS] Ccp1con.4 = Duty.0 : Ccp1con.5 = Duty.1 : Ccpr1l = Duty >>
2
; Bit_Bit DUTY,0,CCP1CON,4
            Bsf CCP1CON,4
            Btfss DUTY,0
            Bcf CCP1CON,4
; Bit_Bit DUTY,1,CCP1CON,5
            Bsf CCP1CON,5
            Btfss DUTY,1
            Bcf CCP1CON,5
            Rrf DUTYH,W
            Movwf PP4
            Rrf DUTY,W
            Movwf CCPR1L
            Rrf PP4,F
            Rrf CCPR1L,F
```

```
            Movfw DUTYH
            Movwf PP0H
            Movfw DUTY
            Movwf PP0
            Movlw 2
            F@Call r@sh
            Movwf CCPR1L
F1_000534 equ $ ; in [SONKOD.BAS] GOTO KONTROL_CIK
            F@Jump KONTROL_CIK
            F@Jump bc@ll106
F1_000538 equ $ ; in [SONKOD.BAS] elseif kontroltipi=5 then
bc@ll134
            Movlw 5
            Subwf KONTROLTIPI,W
            set@page bc@ll147
            Btfss STATUS,2
            Goto bc@ll147
PID_KONTROL
F1_000541 equ $ ; in [SONKOD.BAS] IF DERECE_SET >= TEMPORT THEN
            Movfw TEMPORTH
            set@page bc@ll149
            Btfss STATUS,2
            Goto bc@ll149
            Movfw TEMPORT
            Subwf DERECE_SET,W
            set@page bc@ll149
            Btfss STATUS,0
            Goto bc@ll149
F1_000542 equ $ ; in [SONKOD.BAS] FARK_E=DERECE_SET-TEMPORT
            Movfw TEMPORT
            Subwf DERECE_SET,W
            Movwf FARK_E
            Comf TEMPORTH,W
            Skpnc
            Addlw 1
            Movwf FARK_EH
F1_000543 equ $ ; in [SONKOD.BAS] IF FARK_E<10 THEN
            Movfw FARK_EH
            set@page bc@ll151
            Btfss STATUS,2
            Goto bc@ll151
            Movlw 10
            Subwf FARK_E,W
            set@page bc@ll151
            Btfsc STATUS,0
            Goto bc@ll151
F1_000544 equ $ ; in [SONKOD.BAS] F_INT=FARK_E+F_INT
            Movfw FARK_E
            Addwf F_INT,F
            Movfw FARK_EH
            Skpnc
            Addlw 1
            Addwf F_INTH,F
F1_000545 equ $ ; in [SONKOD.BAS] END IF
bc@ll151
F1_000546 equ $ ; in [SONKOD.BAS] END IF
bc@ll149
```

```
F1_000547 equ $ ; in [SONKOD.BAS] IF DERECE_SET < TEMPORT THEN
        Movfw TEMPORTH
        set@page cp@lb38
        Btfss STATUS,2
        Goto cp@lb38
        Movfw TEMPORT
        Subwf DERECE_SET,W
        set@page bc@ll153
        Btfsc STATUS,0
        Goto bc@ll153
cp@lb38
F1_000548 equ $ ; in [SONKOD.BAS] FARK_E=TEMPORT-DERECE_SET
        Movfw DERECE_SET
        Subwf TEMPORT,W
        Movwf FARK_E
        Movfw TEMPORTH
        Skpc
        Addlw 255
        Movwf FARK_EH
F1_000549 equ $ ; in [SONKOD.BAS] IF F_INT>FARK_E THEN
        Movfw F_INTH
        Subwf FARK_EH,W
        set@page cp@lb39
        Btfss STATUS,0
        Goto cp@lb39
        set@page bc@ll155
        Btfss STATUS,2
        Goto bc@ll155
        Movfw F_INT
        Subwf FARK_E,W
        set@page bc@ll155
        Btfsc STATUS,0
        Goto bc@ll155
cp@lb39
F1_000550 equ $ ; in [SONKOD.BAS] F_INT=F_INT-FARK_E
        Movfw FARK_E
        Subwf F_INT,F
        Movfw FARK_EH
        Skpc
        Addlw 1
        Subwf F_INTH,F
        F@Jump bc@ll156
bc@ll155
F1_000551 equ $ ; in [SONKOD.BAS] ELSE
F1_000552 equ $ ; in [SONKOD.BAS] F_INT=0
        Clrf F_INTH
        Clrf F_INT
F1_000553 equ $ ; in [SONKOD.BAS] END IF
bc@ll156
F1_000554 equ $ ; in [SONKOD.BAS] END IF
bc@ll153
F1_000555 equ $ ; in [SONKOD.BAS] F_INT_TOPL=0
        Clrf F_INT_TOPLH
        Clrf F_INT_TOPL
F1_000556 equ $ ; in [SONKOD.BAS] FOR i = 0 TO KI_VALUE
        Clrf _I
fr@lb158
```

```
            Movfw KI_VALUEH
            set@page cp@lb40
            Btfss STATUS,2
            Goto cp@lb40
            Movfw _I
            Subwf KI_VALUE,W
            set@page nx@lb159
            Btfss STATUS,0
            Goto nx@lb159
cp@lb40
F1_000557 equ $ ; in [SONKOD.BAS] F_INT_TOPL= F_INT + F_INT_TOPL
            Movfw F_INT
            Addwf F_INT_TOPL,F
            Movfw F_INTH
            Skpnc
            Addlw 1
            Addwf F_INT_TOPLH,F
F1_000558 equ $ ; in [SONKOD.BAS] NEXT
            Incf _I,F
            set@page fr@lb158
            Btfss STATUS,2
            Goto fr@lb158
nx@lb159
F1_000561 equ $ ; in [SONKOD.BAS] IF DERECE_SET >= TEMPORT THEN
            Movfw TEMPORTH
            set@page bc@ll160
            Btfss STATUS,2
            Goto bc@ll160
            Movfw TEMPORT
            Subwf DERECE_SET,W
            set@page bc@ll160
            Btfss STATUS,0
            Goto bc@ll160
F1_000562 equ $ ; in [SONKOD.BAS] FARK = DERECE_SET-TEMPORT
            Movfw TEMPORT
            Subwf DERECE_SET,W
            Movwf FARK
            Comf TEMPORTH,W
            Skpnc
            Addlw 1
            Movwf FARKH
F1_000563 equ $ ; in [SONKOD.BAS] DUTY = KP_VALUE*FARK
            Movfw KP_VALUEH
            Movwf PP3H
            Movfw KP_VALUE
            Movwf PP3
            Movfw FARKH
            Movwf PP1H
            Movfw FARK
            Movwf PP1
            F@Call m@py
            Movwf DUTY
            Movfw PP2H
            Movwf DUTYH
            F@Jump bc@ll159
F1_000564 equ $ ; in [SONKOD.BAS] ELSEIF  DERECE_SET < TEMPORT THEN
bc@ll160
```

```
            Movfw TEMPORTH
            set@page cp@lb42
            Btfss STATUS,2
            Goto cp@lb42
            Movfw TEMPORT
            Subwf DERECE_SET,W
            set@page bc@ll161
            Btfsc STATUS,0
            Goto bc@ll161
cp@lb42
F1_000565 equ $ ; in [SONKOD.BAS] DUTY = 0
            Clrf DUTYH
            Clrf DUTY
F1_000566 equ $ ; in [SONKOD.BAS] ENDIF
bc@ll161
bc@ll159
F1_000568 equ $ ; in [SONKOD.BAS] DUTY = DUTY + F_INT_TOPL
            Movfw F_INT_TOPL
            Addwf DUTY,F
            Movfw F_INT_TOPLH
            Skpnc
            Addlw 1
            Addwf DUTYH,F
F1_000571 equ $ ; in [SONKOD.BAS] ED2=TEMPORT
            Movfw TEMPORT
            Movwf ED2
F1_000572 equ $ ; in [SONKOD.BAS] IF ED2>=ED1 THEN
            Movfw ED1
            Subwf ED2,W
            set@page bc@ll163
            Btfss STATUS,0
            Goto bc@ll163
F1_000573 equ $ ; in [SONKOD.BAS] FARK_D=ED2-ED1
            Movfw ED1
            Subwf ED2,W
            Movwf FARK_D
            Clrf FARK_DH
            Skpc
            Decf FARK_DH,F
F1_000574 equ $ ; in [SONKOD.BAS] F_DER_TOPL=0
            Clrf F_DER_TOPLH
            Clrf F_DER_TOPL
F1_000575 equ $ ; in [SONKOD.BAS] FOR i = 0 TO KD_VALUE
            Clrf _I
fr@lb165
            Movfw KD_VALUEH
            set@page cp@lb43
            Btfss STATUS,2
            Goto cp@lb43
            Movfw _I
            Subwf KD_VALUE,W
            set@page nx@lb166
            Btfss STATUS,0
            Goto nx@lb166
cp@lb43
F1_000576 equ $ ; in [SONKOD.BAS] F_DER_TOPL= F_DER + F_DER_TOPL
            Movfw F_DER
```

```
            Addwf F_DER_TOPL,F
            Movfw F_DERH
            Skpnc
            Addlw 1
            Addwf F_DER_TOPLH,F
F1_000577 equ $ ; in [SONKOD.BAS] NEXT
            Incf _I,F
            set@page fr@lb165
            Btfss STATUS,2
            Goto fr@lb165
nx@lb166
F1_000578 equ $ ; in [SONKOD.BAS] DUTY = DUTY + F_DER_TOPL
            Movfw F_DER_TOPL
            Addwf DUTY,F
            Movfw F_DER_TOPLH
            Skpnc
            Addlw 1
            Addwf DUTYH,F
            F@Jump bc@ll162
F1_000579 equ $ ; in [SONKOD.BAS] ELSEIF ED1>ED2 THEN
bc@ll163
            Movfw ED1
            Subwf ED2,W
            set@page bc@ll166
            Btfsc STATUS,0
            Goto bc@ll166
F1_000580 equ $ ; in [SONKOD.BAS] FARK_D=ED1-ED2
            Movfw ED2
            Subwf ED1,W
            Movwf FARK_D
            Clrf FARK_DH
            Skpc
            Decf FARK_DH,F
F1_000581 equ $ ; in [SONKOD.BAS] F_DER_TOPL=0
            Clrf F_DER_TOPLH
            Clrf F_DER_TOPL
F1_000582 equ $ ; in [SONKOD.BAS] FOR i = 0 TO KD_VALUE
            Clrf _I
fr@lb168
            Movfw KD_VALUEH
            set@page cp@lb44
            Btfss STATUS,2
            Goto cp@lb44
            Movfw _I
            Subwf KD_VALUE,W
            set@page nx@lb169
            Btfss STATUS,0
            Goto nx@lb169
cp@lb44
F1_000583 equ $ ; in [SONKOD.BAS] F_DER_TOPL= F_DER + F_DER_TOPL
            Movfw F_DER
            Addwf F_DER_TOPL,F
            Movfw F_DERH
            Skpnc
            Addlw 1
            Addwf F_DER_TOPLH,F
F1_000584 equ $ ; in [SONKOD.BAS] NEXT
```

```
            Incf _I,F
            set@page fr@lb168
            Btfss STATUS,2
            Goto fr@lb168
nx@lb169
F1_000585 equ $ ; in [SONKOD.BAS] DUTY = DUTY - F_DER_TOPL
            Movfw F_DER_TOPL
            Subwf DUTY,F
            Movfw F_DER_TOPLH
            Skpc
            Addlw 1
            Subwf DUTYH,F
F1_000586 equ $ ; in [SONKOD.BAS] END IF
bc@ll166
bc@ll162
F1_000587 equ $ ; in [SONKOD.BAS] ED1=ED2
            Movfw ED2
            Movwf ED1
F1_000589 equ $ ; in [SONKOD.BAS] IF DUTY > 950 THEN
            Movlw 3
            Subwf DUTYH,W
            set@page bc@ll170
            Btfss STATUS,0
            Goto bc@ll170
            set@page cp@lb45
            Btfss STATUS,2
            Goto cp@lb45
            Movlw 183
            Subwf DUTY,W
            set@page bc@ll170
            Btfss STATUS,0
            Goto bc@ll170
cp@lb45
F1_000590 equ $ ; in [SONKOD.BAS] DUTY = 950
            Movlw 3
            Movwf DUTYH
            Movlw 182
            Movwf DUTY
F1_000591 equ $ ; in [SONKOD.BAS] END IF
bc@ll170
F1_000593 equ $ ; in [SONKOD.BAS] PRINT DEC DUTY
            Movlw 128
            Movwf BPFH
            Clrf GEN4H
            Movfw DUTYH
            Movwf PP2H
            Movfw DUTY
            Movwf PP2
            F@Call out@dec
F1_000594 equ $ ; in [SONKOD.BAS] Ccp1con.4 = Duty.0 : Ccp1con.5 = Duty.1 : Ccpr1l = Duty >>
2
; Bit_Bit DUTY,0,CCP1CON,4
            Bsf CCP1CON,4
            Btfss DUTY,0
            Bcf CCP1CON,4
; Bit_Bit DUTY,1,CCP1CON,5
            Bsf CCP1CON,5
```

```
        Btfss DUTY,1
        Bcf CCP1CON,5
        Rrf DUTYH,W
        Movwf PP4
        Rrf DUTY,W
        Movwf CCPR1L
        Rrf PP4,F
        Rrf CCPR1L,F
        Movfw DUTYH
        Movwf PP0H
        Movfw DUTY
        Movwf PP0
        Movlw 2
        F@Call r@sh
        Movwf CCPR1L
F1_000595 equ $ ; in [SONKOD.BAS] GOTO KONTROL_CIK
        F@Jump KONTROL_CIK
F1_000596 equ $ ; in [SONKOD.BAS] END IF
bc@ll147
bc@ll106
KONTROL_CIK
F1_000599 equ $ ; in [SONKOD.BAS] RETURN
        Return
        END
```

EK-2. Devrelerde kullanılan elemanlara ait teknik bilgiler

723-1477 to 723-1696

JIMI~HEAT

Trace Heating Systems For Industry

Technical Data Sheet

JMS SILICONE RUBBER HEATER MATS

CE

Introduction

Jimi-Heat silicone rubber heater mats are available in a number of different formats for temperature maintenance, heat-up duties or defrost applications in refrigeration. They have a withstand temperature range of -50°C to +180°C, and a maximum heated area load of 5Kw/m^2 (allowances made for edge perimeter unheated).Can be higher for specific applications above 5 Kw/m^2, but will not be covered under Jimi-Heat's Warranty. One of the major benefits of this type of mat is that it can be cut to shape during the manufacturing process and is ideally suited for small and intricate applications. The mats consist of a wire resistance element embedded and then cured into a silicone rubber impregnated glass cloth carrier.

Application

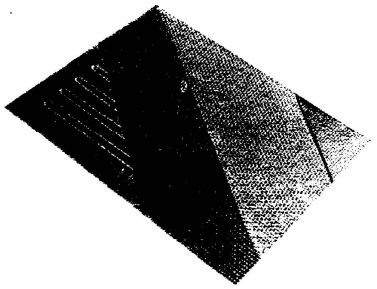
JMS heater mats can be designed for a variety of applications.
These include:-

- **Storage Tanks**
- **Tank Containers**
- **Hopper heating**
- **Platens**
- **Food Counters**
- **Rail carriage floor heating**

Electrical Connection & Installation

Each JMS heater mat would have a designed power rating depending on the application. The maximum power rating would be 5Kw/m^2. Depending on the option selected, the mat will have either a sheathed cold lead or fly lead termination for connection to the power source. The heater mat should be installed directly on to a surface. It is essential that it has direct contact with the surface so that air gaps are avoided at any position along the surface of the mat.

Heater Mat Type JMS/EA



Heater mats are available with adhesive backing sheet, but as an alternative can be installed either with a silicone RTV adhesive between the mat and the surface concerned or with suitable strapping. Can also be supplied ready-fixed to metal sheets. Thermal insulation may be fitted behind the mat to reduce heat losses. Under no circumstances should the insulation enter between the mat and the surface to be heated.

Means of Control

If maintaining the temperature of a vessel or surface, then it is essential that suitable surface control is provided ie. via thermostat bulb or sensor device from an electronic controller.

Jimi-Heat is well regarded as a leading name in the manufacture and supply of high quality electric surface-heating products for a wide variety of applications. Products are manufactured within the scope of our ISO 9002 Quality Assurance System - Certificate No. FM 24706. The product portfolio spans cut-to-length and fixed-length heater tapes, heater cables, heater mats, drum heaters and temperature controls.

Jimi~Heat Ltd, Jimi~Heat House, 200 Rickmansworth Road, Watford, Herts WD1 7JS
Tel: (01923) 234477 Fax: (01923) 240264

FM24706

BEAMA

JMS TECHNICAL SPECIFICATION	
Loading (Standard)	Up to 5Kw/m² (within heated perimeter)
Construction	Straight or spiralled nichrome wire element. Silicone rubber impregnated glass cloth carrier
Maximum Size	2m x 0.8m.
Minimum Size	0.05m x 0.05m
Operating Voltage	230/240 Volts standard, variation from 12-415 Volts
Maximum Operating Temp.	150°C (Must not be exceeded)
Thickness	2.00mm/2.50mm
Min. Withstand Temp.	Minus 50°C (minus 30°C with adhesive backing).
Max Withstand Temp	180°C (150°C with adhesive backing).
Standard	Routine tested in line with BS 6351 & BS EN 60519 Parts 1,2, & 10
Dielectric Test	1.5 KV
Terminations	Depending on mat variation.

SILICONE RUBBER HEATER MATS CODING							
Basic Type	Adhesive Backed B	Override Thermostat T	Waterproof W	Earthed Metal Protection E	Mould M	Cold Lead (Sheathed) ie. 2 Core)	Single Core Flyleads FL
JMS/E	X	X	X	X	X	X	X
JMS/F	X	X	X	X	X	X	X
JMS/EA	X	X	-	X	X	X	X
JMS/AA	X	-	-	-	-	-	X

The above table provides the many different variations of silicone rubber heater mats available

Type E represents heavy durable - both sides
Type F represents flexible thick - both sides
Type EA represents heavy durable one side/thin other side
Type AA represents thin material both sides.

Note:

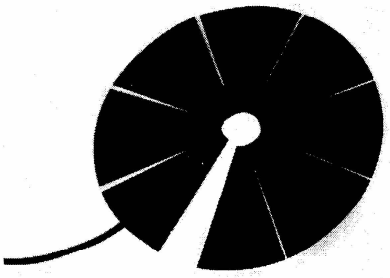
(a) If adhesive backing required, this will be 100% cover (maximum withstand temp 150°C).

(b) 1m. cold lead via mould/flylead without mould provided as standard with each mat (minimum 0.50mm²)

(c) JMS/F - Flexible applications eg. small curvatures.
JMS/E - General applications eg. flat surfaces/curvatures.
JMS/EA - Flat permanently fixed applications/slight curvatures.
JMS/AA - Standard applications eg. flat surfaces (indoor use).

(d) Over-temperature protection (of JMS heater mat) thermostat cut-out available as an option (standard range 55°C-150°C.)

Heater Mat Type JMS/E



Health and Safety

All Jimi-Heat JMS products are manufactured to conform with the requirements of the Low Voltage Directive and are CE marked to indicate conformance. Products are deemed to be safe and non-hazardous providing they are used within the manufacturer's specifications and instructions.

DALLAS SEMICONDUCTOR MAXIM

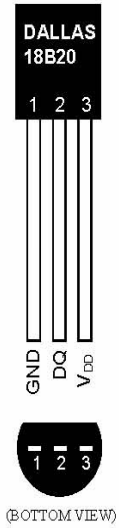
DS18B20
Programmable Resolution 1-Wire Digital Thermometer

www.maxim-ic.com

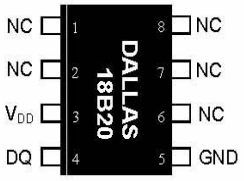
FEATURES

- Unique 1-Wire® interface requires only one port pin for communication
- Each device has a unique 64-bit serial code stored in an onboard ROM
- Multidrop capability simplifies distributed temperature sensing applications
- Requires no external components
- Can be powered from data line. Power supply range is 3.0V to 5.5V
- Measures temperatures from –55°C to +125°C (–67°F to +257°F)
- ±0.5°C accuracy from –10°C to +85°C
- Thermometer resolution is user-selectable from 9 to 12 bits
- Converts temperature to 12-bit digital word in 750ms (max.)
- User-definable nonvolatile (NV) alarm settings
- Alarm search command identifies and addresses devices whose temperature is outside of programmed limits (temperature alarm condition)
- Available in 8-pin SO (150mil), 8-pin µSOP, and 3-pin TO-92 packages
- Software compatible with the DS1822
- Applications include thermostatic controls, industrial systems, consumer products, thermometers, or any thermally sensitive system

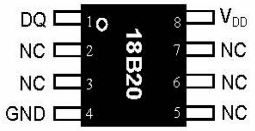
PIN ASSIGNMENT



TO-92
(DS18B20)



8-Pin 150mil SO
(DS18B20Z)



8-Pin µSOP
(DS18B20U)

PIN DESCRIPTION

GND - Ground
DQ - Data In/Out
V_{DD} - Power Supply Voltage
NC - No Connect

DESCRIPTION

The DS18B20 Digital Thermometer provides 9 to 12–bit centigrade temperature measurements and has an alarm function with nonvolatile user-programmable upper and lower trigger points. The DS18B20 communicates over a 1-Wire bus that by definition requires only one data line (and ground) for communication with a central microprocessor. It has an operating temperature range of –55°C to +125°C and is accurate to ±0.5°C over the range of –10°C to +85°C. In addition, the DS18B20 can derive power directly from the data line ("parasite power"), eliminating the need for an external power supply.

Each DS18B20 has a unique 64-bit serial code, which allows multiple DS18B20s to function on the same 1–wire bus; thus, it is simple to use one microprocessor to control many DS18B20s distributed over a large area. Applications that can benefit from this feature include HVAC environmental controls, temperature monitoring systems inside buildings, equipment or machinery, and process monitoring and control systems.

ORDER INFORMATION

ORDERING NUMBER	PACKAGE MARKING	DESCRIPTION
DS18B20	18B20	DS18B20 in 3-pin TO92
DS18B20/T&R	18B20	DS18B20 in 3-pin TO92, 2000 Piece Tape-and-Reel
DS18B20+	18B20 (See Note)	DS18B20 in Lead-Free 3-pin TO92
DS18B20+T&R	18B20 (See Note)	DS18B20 in Lead-Free 3-pin TO92, 2000 Piece Tape-and-Reel
DS18B20U	18B20	DS18B20 in 8-pin uSOP
DS18B20U/T&R	18B20	DS18B20 in 8-pin uSOP, 3000 Piece Tape-and-Reel
DS18B20U+	18B20 (See Note)	DS18B20 in Lead-Free 8-pin uSOP
DS18B20U+T&R	18B20 (See Note)	DS18B20 in Lead-Free 8-pin uSOP, 3000 Piece Tape-and-Reel
DS18B20Z	DS18B20	DS18B20 in 150 mil 8-pin SO
DS18B20Z/T&R	DS18B20	DS18B20 in 150 mil 8-pin SO, 2500 Piece Tape-and-Reel
DS18B20Z+	DS18B20 (See Note)	DS18B20 in Lead-Free 150 mil 8-pin SO
DS18B20Z+T&R	DS18B20 (See Note)	DS18B20 in Lead-Free 150 mil 8-pin SO, 2500 Piece Tape-and-Reel
DS18B20X	28	DS18B20 in Flip Chip, 10000 Piece Tape-and-Reel

Note: A "+" symbol will also be marked on the package.

DETAILED PIN DESCRIPTIONS Table 1

SO*	µSOP*	TO-92	SYMBOL	DESCRIPTION
5	4	1	GND	**Ground.**
4	1	2	DQ	**Data Input/Output pin.** Open-drain 1-Wire interface pin. Also provides power to the device when used in parasite power mode (see "Parasite Power" section.)
3	8	3	V_{DD}	**Optional V_{DD} pin.** V_{DD} must be grounded for operation in parasite power mode.

*All pins not specified in this table are "No Connect" pins.

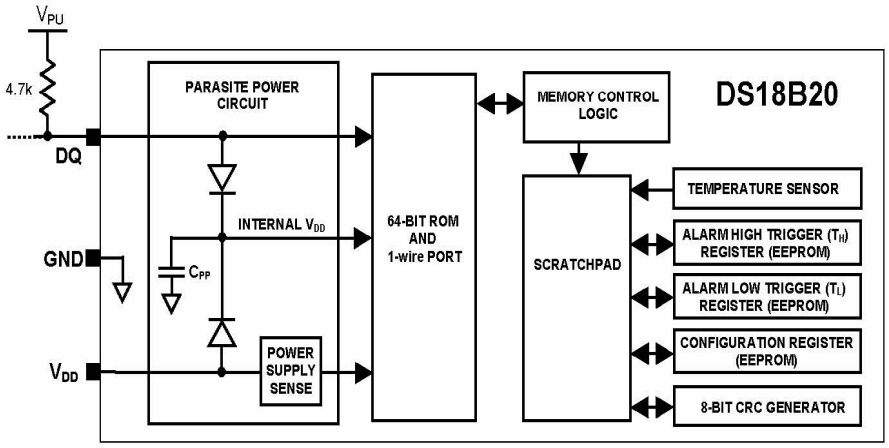
OVERVIEW

Figure 1 shows a block diagram of the DS18B20, and pin descriptions are given in Table 1. The 64-bit ROM stores the device's unique serial code. The scratchpad memory contains the 2-byte temperature register that stores the digital output from the temperature sensor. In addition, the scratchpad provides access to the 1-byte upper and lower alarm trigger registers (T_H and T_L), and the 1-byte configuration register. The configuration register allows the user to set the resolution of the temperature-to-digital conversion to 9, 10, 11, or 12 bits. The T_H, T_L and configuration registers are nonvolatile (EEPROM), so they will retain data when the device is powered down.

The DS18B20 uses Dallas' exclusive 1-Wire bus protocol that implements bus communication using one control signal. The control line requires a weak pullup resistor since all devices are linked to the bus via a 3-state or open-drain port (the DQ pin in the case of the DS18B20). In this bus system, the microprocessor (the master device) identifies and addresses devices on the bus using each device's unique 64-bit code. Because each device has a unique code, the number of devices that can be addressed on one bus is virtually unlimited. The 1-Wire bus protocol, including detailed explanations of the commands and "time slots," is covered in the *1-WIRE BUS SYSTEM* section of this datasheet.

Another feature of the DS18B20 is the ability to operate without an external power supply. Power is instead supplied through the 1-Wire pullup resistor via the DQ pin when the bus is high. The high bus signal also charges an internal capacitor (C_{PP}), which then supplies power to the device when the bus is low. This method of deriving power from the 1-Wire bus is referred to as "parasite power." As an alternative, the DS18B20 may also be powered by an external supply on V_{DD}.

DS18B20 BLOCK DIAGRAM Figure 1



OPERATION — MEASURING TEMPERATURE

The core functionality of the DS18B20 is its direct-to-digital temperature sensor. The resolution of the temperature sensor is user-configurable to 9, 10, 11, or 12 bits, corresponding to increments of 0.5°C, 0.25°C, 0.125°C, and 0.0625°C, respectively. The default resolution at power-up is 12-bit. The DS18B20 powers-up in a low-power idle state; to initiate a temperature measurement and A-to-D conversion, the master must issue a Convert T [44h] command. Following the conversion, the resulting thermal data is stored in the 2-byte temperature register in the scratchpad memory and the DS18B20 returns to its idle state. If the DS18B20 is powered by an external supply, the master can issue "read time slots" (see the *1-WIRE BUS SYSTEM* section) after the Convert T command and the DS18B20 will respond by transmitting 0 while the temperature conversion is in progress and 1 when the conversion is done. If the DS18B20 is powered with parasite power, this notification technique cannot be used since the bus must be pulled high by a strong pullup during the entire temperature conversion. The bus requirements for parasite power are explained in detail in the *POWERING THE DS18B20* section of this datasheet.

The DS18B20 output temperature data is calibrated in degrees centigrade; for Fahrenheit applications, a lookup table or conversion routine must be used. The temperature data is stored as a 16-bit sign-extended two's complement number in the temperature register (see Figure 2). The sign bits (S) indicate if the temperature is positive or negative: for positive numbers S = 0 and for negative numbers S = 1. If the DS18B20 is configured for 12-bit resolution, all bits in the temperature register will contain valid data. For 11-bit resolution, bit 0 is undefined. For 10-bit resolution, bits 1 and 0 are undefined, and for 9-bit resolution bits 2, 1 and 0 are undefined. Table 2 gives examples of digital output data and the corresponding temperature reading for 12-bit resolution conversions.

TEMPERATURE REGISTER FORMAT Figure 2

	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
LS Byte	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}
	bit 15	**bit 14**	**bit 13**	**bit 12**	**bit 11**	**bit 10**	**bit 9**	**bit 8**
MS Byte	S	S	S	S	S	2^6	2^5	2^4

TEMPERATURE/DATA RELATIONSHIP Table 2

TEMPERATURE	DIGITAL OUTPUT (Binary)	DIGITAL OUTPUT (Hex)
+125°C	0000 0111 1101 0000	07D0h
+85°C*	0000 0101 0101 0000	0550h
+25.0625°C	0000 0001 1001 0001	0191h
+10.125°C	0000 0000 1010 0010	00A2h
+0.5°C	0000 0000 0000 1000	0008h
0°C	0000 0000 0000 0000	0000h
-0.5°C	1111 1111 1111 1000	FFF8h
-10.125°C	1111 1111 0101 1110	FF5Eh
-25.0625°C	1111 1110 0110 1111	FE6Fh
-55°C	1111 1100 1001 0000	FC90h

*The power-on reset value of the temperature register is +85°C

OPERATION — ALARM SIGNALING

After the DS18B20 performs a temperature conversion, the temperature value is compared to the user-defined two's complement alarm trigger values stored in the 1-byte T_H and T_L registers (see Figure 3). The sign bit (S) indicates if the value is positive or negative: for positive numbers S = 0 and for negative numbers S = 1. The T_H and T_L registers are nonvolatile (EEPROM) so they will retain data when the device is powered down. T_H and T_L can be accessed through bytes 2 and 3 of the scratchpad as explained in the *MEMORY* section of this datasheet.

T_H AND T_L REGISTER FORMAT Figure 3

bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
S	2^6	2^5	2^5	2^5	2^2	2^1	2^0

Only bits 11 through 4 of the temperature register are used in the T_H and T_L comparison since T_H and T_L are 8-bit registers. If the measured temperature is lower than or equal to T_L or higher than T_H, an alarm condition exists and an alarm flag is set inside the DS18B20. This flag is updated after every temperature measurement; therefore, if the alarm condition goes away, the flag will be turned off after the next temperature conversion.

The master device can check the alarm flag status of all DS18B20s on the bus by issuing an Alarm Search [ECh] command. Any DS18B20s with a set alarm flag will respond to the command, so the master can determine exactly which DS18B20s have experienced an alarm condition. If an alarm condition exists and the T_H or T_L settings have changed, another temperature conversion should be done to validate the alarm condition.

POWERING THE DS18B20

The DS18B20 can be powered by an external supply on the V_{DD} pin, or it can operate in "parasite power" mode, which allows the DS18B20 to function without a local external supply. Parasite power is very useful for applications that require remote temperature sensing or that are very space constrained. Figure 1 shows the DS18B20's parasite-power control circuitry, which "steals" power from the 1-Wire bus via the DQ pin when the bus is high. The stolen charge powers the DS18B20 while the bus is high, and some of the charge is stored on the parasite power capacitor (C_{PP}) to provide power when the bus is low. When the DS18B20 is used in parasite power mode, the V_{DD} pin must be connected to ground.

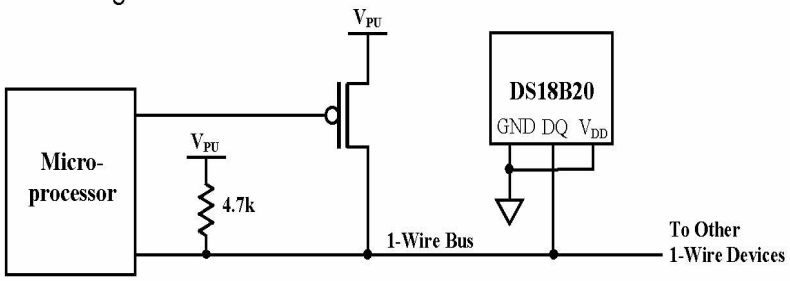
In parasite power mode, the 1-Wire bus and C_{PP} can provide sufficient current to the DS18B20 for most operations as long as the specified timing and voltage requirements are met (refer to *the DC ELECTRICAL CHARACTERISTICS* and the *AC ELECTRICAL CHARACTERISTICS* sections of this data sheet). However, when the DS18B20 is performing temperature conversions or copying data from the scratchpad memory to EEPROM, the operating current can be as high as 1.5mA. This current can cause an unacceptable voltage drop across the weak 1-Wire pullup resistor and is more current than can be supplied by C_{PP}. To assure that the DS18B20 has sufficient supply current, it is necessary to provide a strong pullup on the 1-Wire bus whenever temperature conversions are taking place or data is being copied from the scratchpad to EEPROM. This can be accomplished by using a MOSFET to pull the bus directly to the rail as shown in Figure 4. The 1-Wire bus must be switched to the strong pullup within 10µs (max) after a Convert T [44h] or Copy Scratchpad [48h] command is issued, and the bus must be held high by the pullup for the duration of the conversion (t_{conv}) or data transfer (t_{WR} = 10ms). No other activity can take place on the 1-Wire bus while the pullup is enabled.

The DS18B20 can also be powered by the conventional method of connecting an external power supply to the V_{DD} pin, as shown in Figure 5. The advantage of this method is that the MOSFET pullup is not required, and the 1-Wire bus is free to carry other traffic during the temperature conversion time.

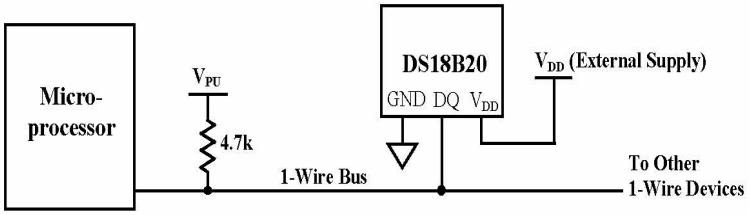
The use of parasite power is not recommended for temperatures above +100°C since the DS18B20 may not be able to sustain communications due to the higher leakage currents that can exist at these temperatures. For applications in which such temperatures are likely, it is strongly recommended that the DS18B20 be powered by an external power supply.

In some situations the bus master may not know whether the DS18B20s on the bus are parasite powered or powered by external supplies. The master needs this information to determine if the strong bus pullup should be used during temperature conversions. To get this information, the master can issue a Skip ROM [CCh] command followed by a Read Power Supply [B4h] command followed by a "read time slot". During the read time slot, parasite powered DS18B20s will pull the bus low, and externally powered DS18B20s will let the bus remain high. If the bus is pulled low, the master knows that it must supply the strong pullup on the 1-Wire bus during temperature conversions.

SUPPLYING THE PARASITE-POWERED DS18B20 DURING TEMPERATURE CONVERSIONS Figure 4



POWERING THE DS18B20 WITH AN EXTERNAL SUPPLY Figure 5



64-BIT LASERED ROM CODE

Each DS18B20 contains a unique 64–bit code (see Figure 6) stored in ROM. The least significant 8 bits of the ROM code contain the DS18B20's 1-Wire family code: 28h. The next 48 bits contain a unique serial number. The most significant 8 bits contain a cyclic redundancy check (CRC) byte that is calculated from the first 56 bits of the ROM code. A detailed explanation of the CRC bits is provided in the *CRC GENERATION* section. The 64-bit ROM code and associated ROM function control logic allow the DS18B20 to operate as a 1-Wire device using the protocol detailed in the *1-WIRE BUS SYSTEM* section of this datasheet.

64-BIT LASERED ROM CODE Figure 6

8-BIT CRC		48-BIT SERIAL NUMBER		8-BIT FAMILY CODE (28h)	
MSB	LSB	MSB	LSB	MSB	LSB

MEMORY

The DS18B20's memory is organized as shown in Figure 7. The memory consists of an SRAM scratchpad with nonvolatile EEPROM storage for the high and low alarm trigger registers (T_H and T_L) and configuration register. Note that if the DS18B20 alarm function is not used, the T_H and T_L registers can serve as general-purpose memory. All memory commands are described in detail in the *DS18B20 FUNCTION COMMANDS* section.

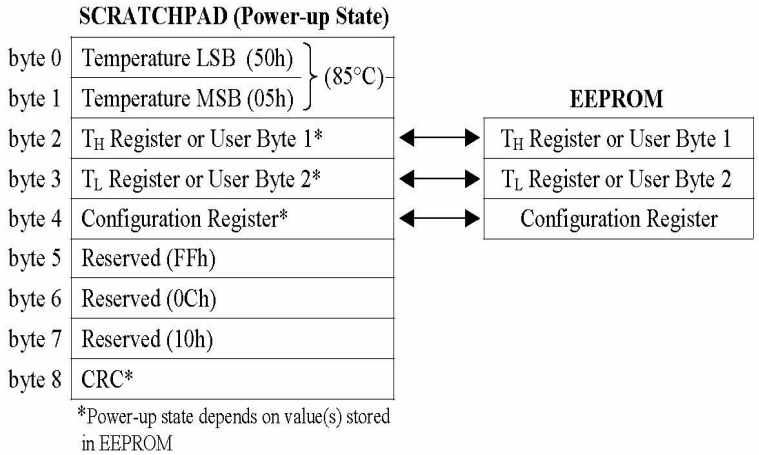
Byte 0 and byte 1 of the scratchpad contain the LSB and the MSB of the temperature register, respectively. These bytes are read-only. Bytes 2 and 3 provide access to T_H and T_L registers. Byte 4 contains the configuration register data, which is explained in detail in the CONFIGURATION REGISTER section of this datasheet. Bytes 5, 6, and 7 are reserved for internal use by the device and cannot be overwritten; these bytes will return all 1s when read.

Byte 8 of the scratchpad is read-only and contains the cyclic redundancy check (CRC) code for bytes 0 through 7 of the scratchpad. The DS18B20 generates this CRC using the method described in *the CRC GENERATION* section.

Data is written to bytes 2, 3, and 4 of the scratchpad using the Write Scratchpad [4Eh] command; the data must be transmitted to the DS18B20 starting with the least significant bit of byte 2. To verify data integrity, the scratchpad can be read (using the Read Scratchpad [BEh] command) after the data is written. When reading the scratchpad, data is transferred over the 1-Wire bus starting with the least significant bit of byte 0. To transfer the T_H, T_L and configuration data from the scratchpad to EEPROM, the master must issue the Copy Scratchpad [48h] command.

Data in the EEPROM registers is retained when the device is powered down; at power-up the EEPROM data is reloaded into the corresponding scratchpad locations. Data can also be reloaded from EEPROM to the scratchpad at any time using the Recall E^2 [B8h] command. The master can issue read time slots following the Recall E^2 command and the DS18B20 will indicate the status of the recall by transmitting 0 while the recall is in progress and 1 when the recall is done.

DS18B20 MEMORY MAP Figure 7



CONFIGURATION REGISTER

Byte 4 of the scratchpad memory contains the configuration register, which is organized as illustrated in Figure 8. The user can set the conversion resolution of the DS18B20 using the R0 and R1 bits in this register as shown in Table 3. The power-up default of these bits is R0 = 1 and R1 = 1 (12-bit resolution). Note that there is a direct tradeoff between resolution and conversion time. Bit 7 and bits 0 to 4 in the configuration register are reserved for internal use by the device and cannot be overwritten; these bits will return 1s when read.

CONFIGURATION REGISTER Figure 8

bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0	R1	R0	1	1	1	1	1

THERMOMETER RESOLUTION CONFIGURATION Table 3

R1	R0	Resolution	Max Conversion Time	
0	0	9-bit	93.75 ms	($t_{CONV}/8$)
0	1	10-bit	187.5 ms	($t_{CONV}/4$)
1	0	11-bit	375 ms	($t_{CONV}/2$)
1	1	12-bit	750 ms	(t_{CONV})

CRC GENERATION

CRC bytes are provided as part of the DS18B20's 64-bit ROM code and in the 9th byte of the scratchpad memory. The ROM code CRC is calculated from the first 56 bits of the ROM code and is contained in the most significant byte of the ROM. The scratchpad CRC is calculated from the data stored in the scratchpad, and therefore it changes when the data in the scratchpad changes. The CRCs provide the bus master with a method of data validation when data is read from the DS18B20. To verify that data has been read correctly, the bus master must re-calculate the CRC from the received data and then compare this value to either the ROM code CRC (for ROM reads) or to the scratchpad CRC (for scratchpad reads). If the calculated CRC matches the read CRC, the data has been received error free. The comparison of CRC values and the decision to continue with an operation are determined entirely by the bus master. There is no circuitry inside the DS18B20 that prevents a command sequence from proceeding if the DS18B20 CRC (ROM or scratchpad) does not match the value generated by the bus master.

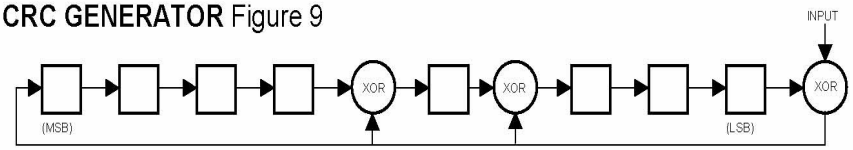
The equivalent polynomial function of the CRC (ROM or scratchpad) is:

$$CRC = X^8 + X^5 + X^4 + 1$$

The bus master can re-calculate the CRC and compare it to the CRC values from the DS18B20 using the polynomial generator shown in Figure 9. This circuit consists of a shift register and XOR gates, and the shift register bits are initialized to 0. Starting with the least significant bit of the ROM code or the least significant bit of byte 0 in the scratchpad, one bit at a time should shifted into the shift register. After shifting in the 56th bit from the ROM or the most significant bit of byte 7 from the scratchpad, the polynomial generator will contain the re-calculated CRC. Next, the 8-bit ROM code or scratchpad CRC from the DS18B20 must be shifted into the circuit. At this point, if the re-calculated CRC was correct, the shift register will contain all 0s. Additional information about the Dallas 1-Wire cyclic redundancy check

is available in *Application Note 27: Understanding and Using Cyclic Redundancy Checks with Dallas Semiconductor Touch Memory Products.*

CRC GENERATOR Figure 9



1-WIRE BUS SYSTEM

The 1-Wire bus system uses a single bus master to control one or more slave devices. The DS18B20 is always a slave. When there is only one slave on the bus, the system is referred to as a "single-drop" system; the system is "multidrop" if there are multiple slaves on the bus.

All data and commands are transmitted least significant bit first over the 1-Wire bus.

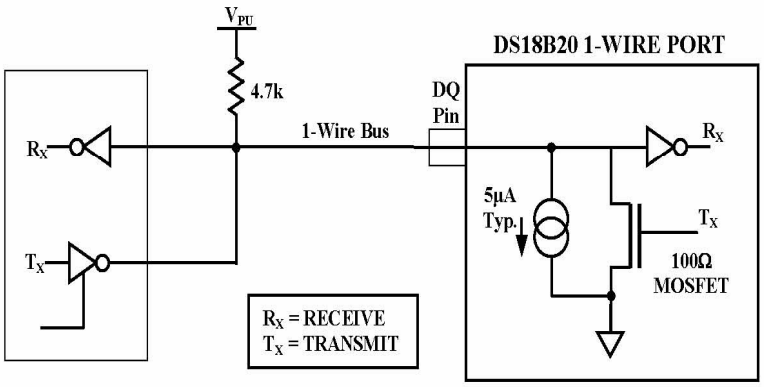
The following discussion of the 1-Wire bus system is broken down into three topics: hardware configuration, transaction sequence, and 1-Wire signaling (signal types and timing).

HARDWARE CONFIGURATION

The 1-Wire bus has by definition only a single data line. Each device (master or slave) interfaces to the data line via an open-drain or 3-state port. This allows each device to "release" the data line when the device is not transmitting data so the bus is available for use by another device. The 1-Wire port of the DS18B20 (the DQ pin) is open drain with an internal circuit equivalent to that shown in Figure 10.

The 1-Wire bus requires an external pullup resistor of approximately 5kΩ; thus, the idle state for the 1-Wire bus is high. If for any reason a transaction needs to be suspended, the bus MUST be left in the idle state if the transaction is to resume. Infinite recovery time can occur between bits so long as the 1-Wire bus is in the inactive (high) state during the recovery period. If the bus is held low for more than 480µs, all components on the bus will be reset.

HARDWARE CONFIGURATION Figure 10



TRANSACTION SEQUENCE

The transaction sequence for accessing the DS18B20 is as follows:

Step 1. Initialization

Step 2. ROM Command (followed by any required data exchange)

Step 3. DS18B20 Function Command (followed by any required data exchange)

It is very important to follow this sequence every time the DS18B20 is accessed, as the DS18B20 will not respond if any steps in the sequence are missing or out of order. Exceptions to this rule are the Search ROM [F0h] and Alarm Search [ECh] commands. After issuing either of these ROM commands, the master must return to Step 1 in the sequence.

INITIALIZATION

All transactions on the 1-Wire bus begin with an initialization sequence. The initialization sequence consists of a reset pulse transmitted by the bus master followed by presence pulse(s) transmitted by the slave(s). The presence pulse lets the bus master know that slave devices (such as the DS18B20) are on the bus and are ready to operate. Timing for the reset and presence pulses is detailed in the *1-WIRE SIGNALING* section.

ROM COMMANDS

After the bus master has detected a presence pulse, it can issue a ROM command. These commands operate on the unique 64-bit ROM codes of each slave device and allow the master to single out a specific device if many are present on the 1-Wire bus. These commands also allow the master to determine how many and what types of devices are present on the bus or if any device has experienced an alarm condition. There are five ROM commands, and each command is 8 bits long. The master device must issue an appropriate ROM command before issuing a DS18B20 function command. A flowchart for operation of the ROM commands is shown in Figure 11.

SEARCH ROM [F0h]

When a system is initially powered up, the master must identify the ROM codes of all slave devices on the bus, which allows the master to determine the number of slaves and their device types. The master learns the ROM codes through a process of elimination that requires the master to perform a Search ROM cycle (i.e., Search ROM command followed by data exchange) as many times as necessary to identify all of the slave devices. If there is only one slave on the bus, the simpler Read ROM command (see below) can be used in place of the Search ROM process. For a detailed explanation of the Search ROM procedure, refer to the *iButton® Book of Standards* at www.ibutton.com/ibuttons/standard.pdf. After every Search ROM cycle, the bus master must return to Step 1 (Initialization) in the transaction sequence.

READ ROM [33h]

This command can only be used when there is one slave on the bus. It allows the bus master to read the slave's 64-bit ROM code without using the Search ROM procedure. If this command is used when there is more than one slave present on the bus, a data collision will occur when all the slaves attempt to respond at the same time.

MATCH ROM [55h]

The match ROM command followed by a 64-bit ROM code sequence allows the bus master to address a specific slave device on a multidrop or single-drop bus. Only the slave that exactly matches the 64-bit ROM code sequence will respond to the function command issued by the master; all other slaves on the bus will wait for a reset pulse.

iButton is a registered trademark of Dallas Semiconductor.

SKIP ROM [CCh]

The master can use this command to address all devices on the bus simultaneously without sending out any ROM code information. For example, the master can make all DS18B20s on the bus perform simultaneous temperature conversions by issuing a Skip ROM command followed by a Convert T [44h] command.

Note that the Read Scratchpad [BEh] command can follow the Skip ROM command only if there is a single slave device on the bus. In this case time is saved by allowing the master to read from the slave without sending the device's 64-bit ROM code. A Skip ROM command followed by a Read Scratchpad command will cause a data collision on the bus if there is more than one slave since multiple devices will attempt to transmit data simultaneously.

ALARM SEARCH [ECh]

The operation of this command is identical to the operation of the Search ROM command except that only slaves with a set alarm flag will respond. This command allows the master device to determine if any DS18B20s experienced an alarm condition during the most recent temperature conversion. After every Alarm Search cycle (i.e., Alarm Search command followed by data exchange), the bus master must return to Step 1 (Initialization) in the transaction sequence. Refer to the *OPERATION — ALARM SIGNALING* section for an explanation of alarm flag operation.

DS18B20 FUNCTION COMMANDS

After the bus master has used a ROM command to address the DS18B20 with which it wishes to communicate, the master can issue one of the DS18B20 function commands. These commands allow the master to write to and read from the DS18B20's scratchpad memory, initiate temperature conversions and determine the power supply mode. The DS18B20 function commands, which are described below, are summarized in Table 4 and illustrated by the flowchart in Figure 12.

CONVERT T [44h]

This command initiates a single temperature conversion. Following the conversion, the resulting thermal data is stored in the 2-byte temperature register in the scratchpad memory and the DS18B20 returns to its low-power idle state. If the device is being used in parasite power mode, within 10μs (max) after this command is issued the master must enable a strong pullup on the 1-Wire bus for the duration of the conversion (t_{conv}) as described in the *POWERING THE DS18B20* section. If the DS18B20 is powered by an external supply, the master can issue read time slots after the Convert T command and the DS18B20 will respond by transmitting a 0 while the temperature conversion is in progress and a 1 when the conversion is done. In parasite power mode this notification technique cannot be used since the bus is pulled high by the strong pullup during the conversion.

WRITE SCRATCHPAD [4Eh]

This command allows the master to write 3 bytes of data to the DS18B20's scratchpad. The first data byte is written into the T_H register (byte 2 of the scratchpad), the second byte is written into the T_L register (byte 3), and the third byte is written into the configuration register (byte 4). Data must be transmitted least significant bit first. All three bytes MUST be written before the master issues a reset, or the data may be corrupted.

READ SCRATCHPAD [BEh]

This command allows the master to read the contents of the scratchpad. The data transfer starts with the least significant bit of byte 0 and continues through the scratchpad until the 9th byte (byte 8 – CRC) is read. The master may issue a reset to terminate reading at any time if only part of the scratchpad data is needed.

COPY SCRATCHPAD [48h]

This command copies the contents of the scratchpad T_H, T_L and configuration registers (bytes 2, 3 and 4) to EEPROM. If the device is being used in parasite power mode, within 10µs (max) after this command is issued the master must enable a strong pullup on the 1-Wire bus for at least 10ms as described in the *POWERING THE DS18B20* section.

RECALL E^2 [B8h]

This command recalls the alarm trigger values (T_H and T_L) and configuration data from EEPROM and places the data in bytes 2, 3, and 4, respectively, in the scratchpad memory. The master device can issue read time slots following the Recall E^2 command and the DS18B20 will indicate the status of the recall by transmitting 0 while the recall is in progress and 1 when the recall is done. The recall operation happens automatically at power-up, so valid data is available in the scratchpad as soon as power is applied to the device.

READ POWER SUPPLY [B4h]

The master device issues this command followed by a read time slot to determine if any DS18B20s on the bus are using parasite power. During the read time slot, parasite powered DS18B20s will pull the bus low, and externally powered DS18B20s will let the bus remain high. Refer to the *POWERING THE DS18B20* section for usage information for this command.

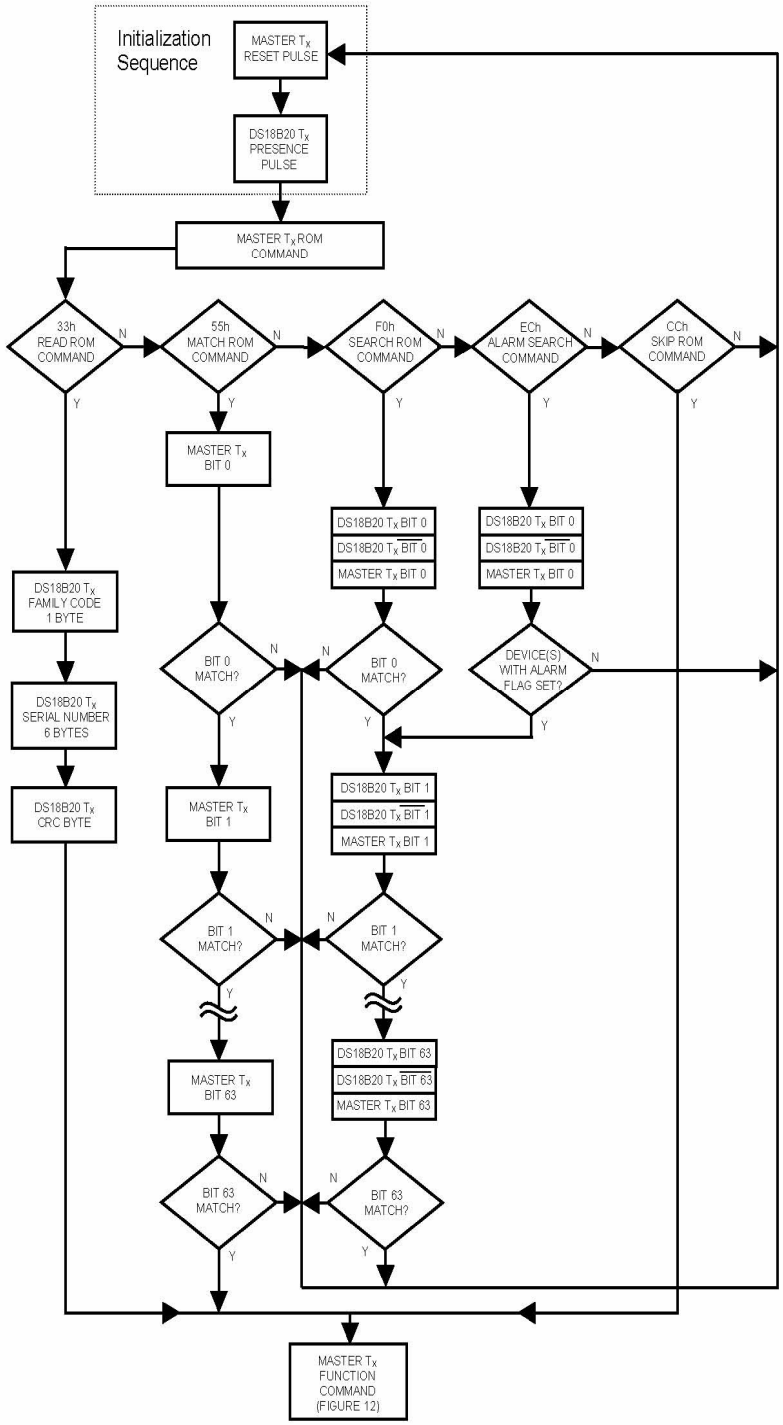
DS18B20 FUNCTION COMMAND SET Table 4

Command	Description	Protocol	1-Wire Bus Activity After Command is Issued	Notes
TEMPERATURE CONVERSION COMMANDS				
Convert T	Initiates temperature conversion.	44h	DS18B20 transmits conversion status to master (not applicable for parasite-powered DS18B20s).	1
MEMORY COMMANDS				
Read Scratchpad	Reads the entire scratchpad including the CRC byte.	BEh	DS18B20 transmits up to 9 data bytes to master.	2
Write Scratchpad	Writes data into scratchpad bytes 2, 3, and 4 (T_H, T_L, and configuration registers).	4Eh	Master transmits 3 data bytes to DS18B20.	3
Copy Scratchpad	Copies T_H, T_L, and configuration register data from the scratchpad to EEPROM.	48h	None	1
Recall E^2	Recalls T_H, T_L, and configuration register data from EEPROM to the scratchpad.	B8h	DS18B20 transmits recall status to master.	
Read Power Supply	Signals DS18B20 power supply mode to the master.	B4h	DS18B20 transmits supply status to master.	

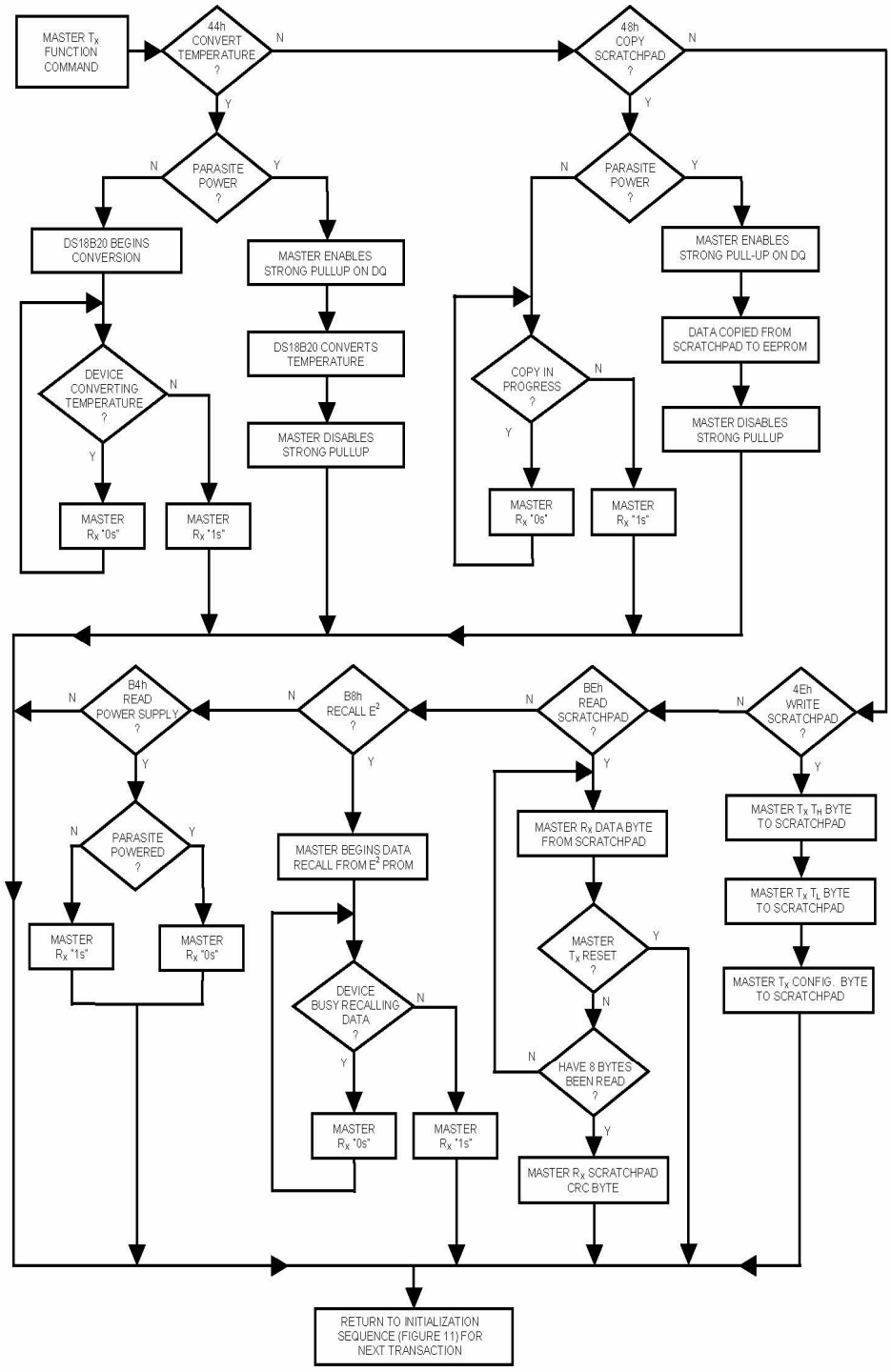
NOTES:

1) For parasite-powered DS18B20s, the master must enable a strong pullup on the 1-Wire bus during temperature conversions and copies from the scratchpad to EEPROM. No other bus activity may take place during this time.
2) The master can interrupt the transmission of data at any time by issuing a reset.
3) All three bytes must be written before a reset is issued.

ROM COMMANDS FLOW CHART Figure 11



DS18B20 FUNCTION COMMANDS FLOW CHART Figure 12



1-WIRE SIGNALING

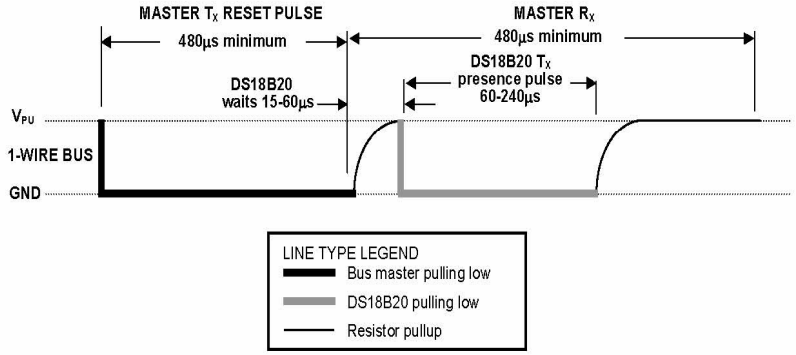
The DS18B20 uses a strict 1-Wire communication protocol to insure data integrity. Several signal types are defined by this protocol: reset pulse, presence pulse, write 0, write 1, read 0, and read 1. The bus master initiates all of these signals, with the exception of the presence pulse.

INITIALIZATION PROCEDURE: RESET AND PRESENCE PULSES

All communication with the DS18B20 begins with an initialization sequence that consists of a reset pulse from the master followed by a presence pulse from the DS18B20. This is illustrated in Figure 13. When the DS18B20 sends the presence pulse in response to the reset, it is indicating to the master that it is on the bus and ready to operate.

During the initialization sequence the bus master transmits (T_X) the reset pulse by pulling the 1-Wire bus low for a minimum of 480µs. The bus master then releases the bus and goes into receive mode (R_X). When the bus is released, the 5k pullup resistor pulls the 1-Wire bus high. When the DS18B20 detects this rising edge, it waits 15µs to 60µs and then transmits a presence pulse by pulling the 1-Wire bus low for 60µs to 240µs.

INITIALIZATION TIMING Figure 13



READ/WRITE TIME SLOTS

The bus master writes data to the DS18B20 during write time slots and reads data from the DS18B20 during read time slots. One bit of data is transmitted over the 1-Wire bus per time slot.

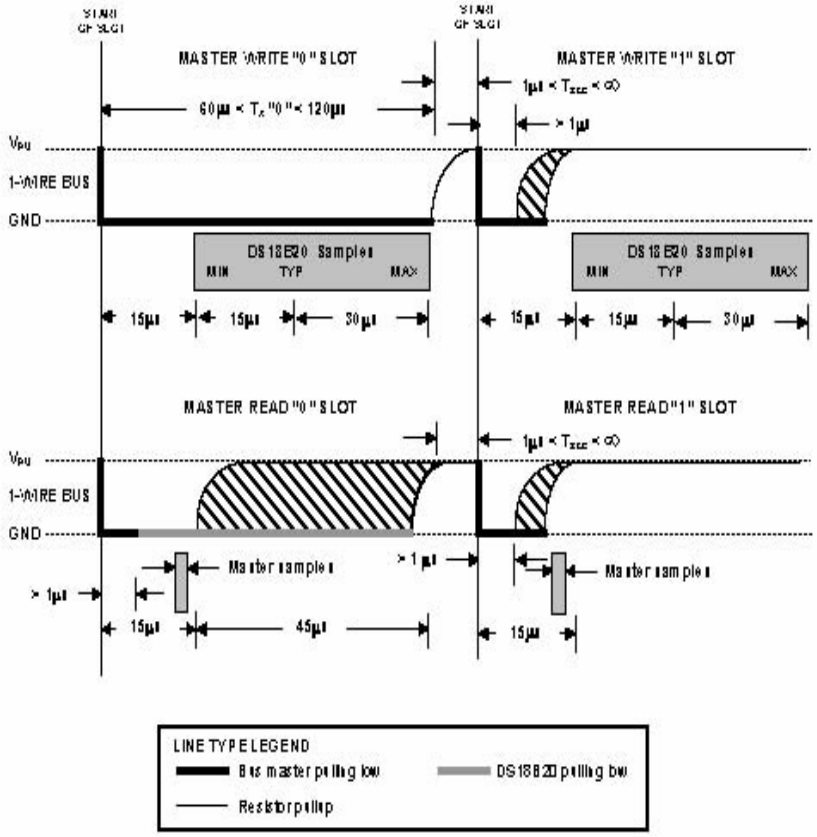
WRITE TIME SLOTS

There are two types of write time slots: "Write 1" time slots and "Write 0" time slots. The bus master uses a Write 1 time slot to write a logic 1 to the DS18B20 and a Write 0 time slot to write a logic 0 to the DS18B20. All write time slots must be a minimum of 60µs in duration with a minimum of a 1µs recovery time between individual write slots. Both types of write time slots are initiated by the master pulling the 1-Wire bus low (see Figure 14).

To generate a Write 1 time slot, after pulling the 1-Wire bus low, the bus master must release the 1-Wire bus within 15µs. When the bus is released, the 5k pullup resistor will pull the bus high. To generate a Write 0 time slot, after pulling the 1-Wire bus low, the bus master must continue to hold the bus low for the duration of the time slot (at least 60µs).

The DS18B20 samples the 1-Wire bus during a window that lasts from 15µs to 60µs after the master initiates the write time slot. If the bus is high during the sampling window, a 1 is written to the DS18B20. If the line is low, a 0 is written to the DS18B20.

READ/WRITE TIME SLOT TIMING DIAGRAM Figure 14



READ TIME SLOTS

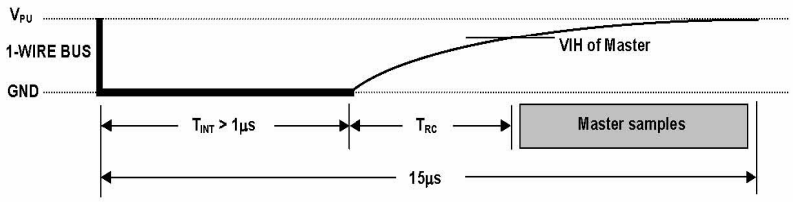
The DS18B20 can only transmit data to the master when the master issues read time slots. Therefore, the master must generate read time slots immediately after issuing a Read Scratchpad [BEh] or Read Power Supply [B4h] command, so that the DS18B20 can provide the requested data. In addition, the master can generate read time slots after issuing Convert T [44h] or Recall E^2 [B8h] commands to find out the status of the operation as explained in the *DS18B20 FUNCTION COMMAND* section.

All read time slots must be a minimum of 60µs in duration with a minimum of a 1µs recovery time between slots. A read time slot is initiated by the master device pulling the 1-Wire bus low for a minimum of 1µs and then releasing the bus (see Figure 14). After the master initiates the read time slot, the DS18B20 will begin transmitting a 1 or 0 on bus. The DS18B20 transmits a 1 by leaving the bus high and transmits a 0 by pulling the bus low. When transmitting a 0, the DS18B20 will release the bus by the end of the time slot, and the bus will be pulled back to its high idle state by the pullup resister. Output

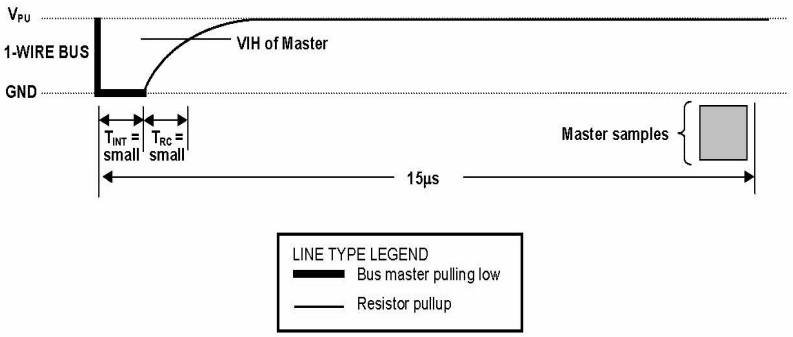
data from the DS18B20 is valid for 15µs after the falling edge that initiated the read time slot. Therefore, the master must release the bus and then sample the bus state within 15µs from the start of the slot.

Figure 15 illustrates that the sum of T_{INIT}, T_{RC}, and T_{SAMPLE} must be less than 15µs for a read time slot. Figure 16 shows that system timing margin is maximized by keeping T_{INIT} and T_{RC} as short as possible and by locating the master sample time during read time slots towards the end of the 15µs period.

DETAILED MASTER READ 1 TIMING Figure 15



RECOMMENDED MASTER READ 1 TIMING Figure 16



RELATED APPLICATION NOTES

The following Application Notes can be applied to the DS18B20. These notes can be obtained from the Dallas Semiconductor "Application Note Book," via the Dallas website at http://www.dalsemi.com/, or through our faxback service at (214) 450–0441.

Application Note 27: Understanding and Using Cyclic Redundancy Checks with Dallas Semiconductor Touch Memory Product
Application Note 55: Extending the Contact Range of Touch Memories
Application Note 74: Reading and Writing Touch Memories via Serial Interfaces
Application Note 104: Minimalist Temperature Control Demo
Application Note 106: Complex MicroLANs
Application Note 108: MicroLAN — In the Long Run
Application Note 162: Interfacing the DS18X20/DS1822 1-Wire Temperature Sensor in a Microcontroller Environment

Sample 1-Wire subroutines that can be used in conjunction with AN74 can be downloaded from the Dallas website or anonymous FTP Site.

DS18B20 OPERATION EXAMPLE 1

In this example there are multiple DS18B20s on the bus and they are using parasite power. The bus master initiates a temperature conversion in a specific DS18B20 and then reads its scratchpad and recalculates the CRC to verify the data.

MASTER MODE	DATA (LSB FIRST)	COMMENTS
TX	Reset	Master issues reset pulse.
RX	Presence	DS18B20s respond with presence pulse.
TX	55h	Master issues Match ROM command.
TX	64-bit ROM code	Master sends DS18B20 ROM code.
TX TX	44h DQ line held high by strong pullup	Master issues Convert T command. Master applies strong pullup to DQ for the duration of the conversion (t_{conv}).
TX	Reset	Master issues reset pulse.
RX	Presence	DS18B20s respond with presence pulse.
TX	55h	Master issues Match ROM command.
TX	64-bit ROM code	Master sends DS18B20 ROM code.
TX	BEh	Master issues Read Scratchpad command.
RX	9 data bytes	Master reads entire scratchpad including CRC. The master then recalculates the CRC of the first eight data bytes from the scratchpad and compares the calculated CRC with the read CRC (byte 9). If they match, the master continues; if not, the read operation is repeated.

DS18B20 OPERATION EXAMPLE 2

In this example there is only one DS18B20 on the bus and it is using parasite power. The master writes to the T_H, T_L, and configuration registers in the DS18B20 scratchpad and then reads the scratchpad and recalculates the CRC to verify the data. The master then copies the scratchpad contents to EEPROM.

MASTER MODE	DATA (LSB FIRST)	COMMENTS
TX	Reset	Master issues reset pulse.
RX	Presence	DS18B20 responds with presence pulse.
TX	CCh	Master issues Skip ROM command.
TX	4Eh	Master issues Write Scratchpad command.
TX	3 data bytes	Master sends three data bytes to scratchpad (T_H, T_L, and config).
TX	Reset	Master issues reset pulse.
RX	Presence	DS18B20 responds with presence pulse.
TX	CCh	Master issues Skip ROM command.
TX	BEh	Master issues Read Scratchpad command.
RX	9 data bytes	Master reads entire scratchpad including CRC. The master then recalculates the CRC of the first eight data bytes from the scratchpad and compares the calculated CRC with the read CRC (byte 9). If they match, the master continues; if not, the read operation is repeated.
TX	Reset	Master issues reset pulse.
RX	Presence	DS18B20 responds with presence pulse.
TX	CCh	Master issues Skip ROM command.
TX	48h	Master issues Copy Scratchpad command.
TX	DQ line held high by strong pullup	Master applies strong pullup to DQ for at least 10ms while copy operation is in progress.

ABSOLUTE MAXIMUM RATINGS*

Voltage on Any Pin Relative to Ground	-0.5V to +6.0V
Operating Temperature Range	-55°C to +125°C
Storage Temperature Range	-55°C to +125°C
Solder Temperature	See IPC/JEDEC J-STD-020A
Reflow Oven Temperature	+220°C

**These are stress ratings only and functional operation of the device at these or any other conditions above those indicated in the operation sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods of time may affect reliability.*

DC ELECTRICAL CHARACTERISTICS (-55°C to +125°C; V_{DD}=3.0V to 5.5V)

PARAMETER	SYMBOL	CONDITION	MIN	TYP	MAX	UNITS	NOTES
Supply Voltage	V_{DD}	Local Power	+3.0		+5.5	V	1
Pullup Supply Voltage	V_{PU}	Parasite Power	+3.0		+5.5	V	1,2
		Local Power	+3.0		V_{DD}		
Thermometer Error	t_{ERR}	-10°C to +85°C			±0.5	°C	3
		-55°C to +125°C			±2		
Input Logic Low	V_{IL}		-0.3		+0.8	V	1,4,5
Input Logic High	V_{IH}	Local Power	+2.2		The lower of 5.5 or V_{DD} + 0.3	V	1, 6
		Parasite Power	+3.0				
Sink Current	I_L	$V_{I/O}$=0.4V	4.0			mA	1
Standby Current	I_{DDS}			750	1000	nA	7,8
Active Current	I_{DD}	V_{DD}=5V		1	1.5	mA	9
DQ Input Current	I_{DQ}			5		µA	10
Drift				±0.2		°C	11

NOTES:

1) All voltages are referenced to ground.
2) The Pullup Supply Voltage specification assumes that the pullup device is ideal, and therefore the high level of the pullup is equal to V_{PU}. In order to meet the V_{IH} spec of the DS18B20, the actual supply rail for the strong pullup transistor must include margin for the voltage drop across the transistor when it is turned on; thus: $V_{PU\_ACTUAL} = V_{PU\_IDEAL} + V_{TRANSISTOR}$.
3) See typical performance curve in Figure 17
4) Logic low voltages are specified at a sink current of 4mA.
5) To guarantee a presence pulse under low voltage parasite power conditions, V_{ILMAX} may have to be reduced to as low as 0.5V.
6) Logic high voltages are specified at a source current of 1mA.
7) Standby current specified up to 70°C. Standby current typically is 3µA at 125°C.
8) To minimize I_{DDS}, DQ should be within the following ranges: GND ≤ DQ ≤ GND + 0.3V or V_{DD} – 0.3V ≤ DQ ≤ V_{DD}.
9) Active current refers to supply current during active temperature conversions or EEPROM writes.
10) DQ line is high ("hi-Z" state).
11) Drift data is based on a 1000 hour stress test at 125°C with V_{DD} = 5.5V.

AC ELECTRICAL CHARACTERISTICS: NV MEMORY

(-55°C to +100°C; V_{DD} = 3.0V to 5.5V)

PARAMETER	SYMBOL	CONDITION	MIN	TYP	MAX	UNITS
NV Write Cycle Time	t_{wr}			2	10	ms
EEPROM Writes	N_{EEWR}	-55°C to +55°C	50k			writes
EEPROM Data Retention	t_{EEDR}	-55°C to +55°C	10			years

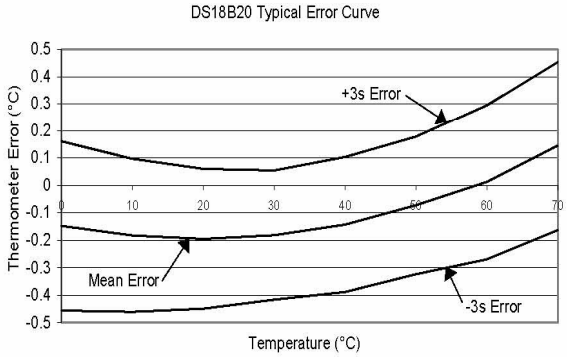
AC ELECTRICAL CHARACTERISTICS (-55°C to +125°C; V_{DD} = 3.0V to 5.5V)

PARAMETER	SYMBOL	CONDITION	MIN	TYP	MAX	UNITS	NOTES
Temperature Conversion Time	t_{CONV}	9-bit resolution			93.75	ms	1
		10-bit resolution			187.5	ms	1
		11-bit resolution			375	ms	1
		12-bit resolution			750	ms	1
Time to Strong Pullup On	t_{SPON}	Start Convert T Command Issued			10	µs	
Time Slot	t_{SLOT}		60		120	µs	1
Recovery Time	t_{REC}		1			µs	1
Write 0 Low Time	r_{LOW0}		60		120	µs	1
Write 1 Low Time	t_{LOW1}		1		15	µs	1
Read Data Valid	t_{RDV}				15	µs	1
Reset Time High	t_{RSTH}		480			µs	1
Reset Time Low	t_{RSTL}		480			µs	1,2
Presence Detect High	t_{PDHIGH}		15		60	µs	1
Presence Detect Low	t_{PDLOW}		60		240	µs	1
Capacitance	$C_{IN/OUT}$				25	pF	

NOTES:

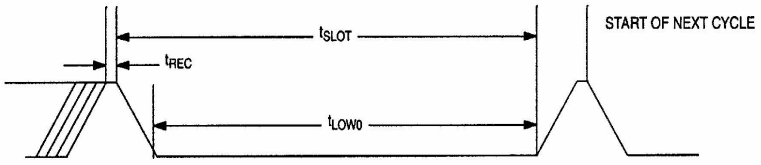
1) Refer to timing diagrams in Figure 18.
2) Under parasite power, if t_{RSTL} > 960µs, a power on reset may occur.

TYPICAL PERFORMANCE CURVE Figure 17

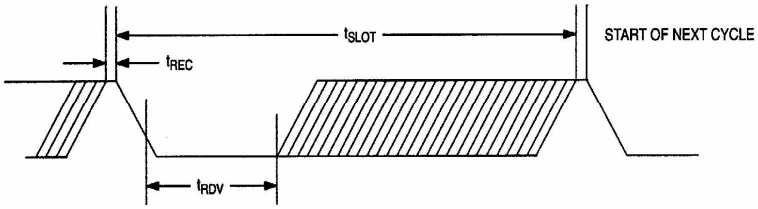


TIMING DIAGRAMS Figure 18

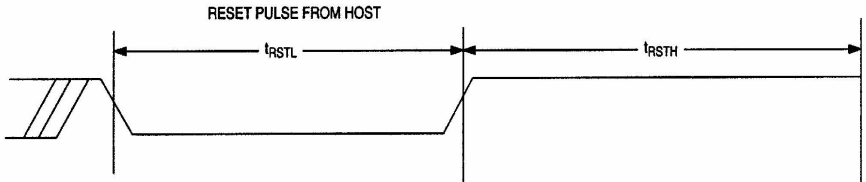
1–WIRE WRITE ZERO TIME SLOT



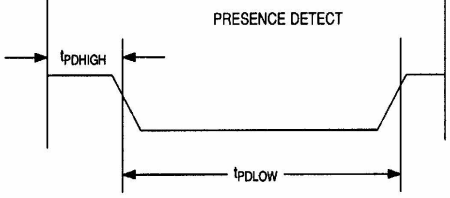
1–WIRE READ ZERO TIME SLOT



1–WIRE RESET PULSE



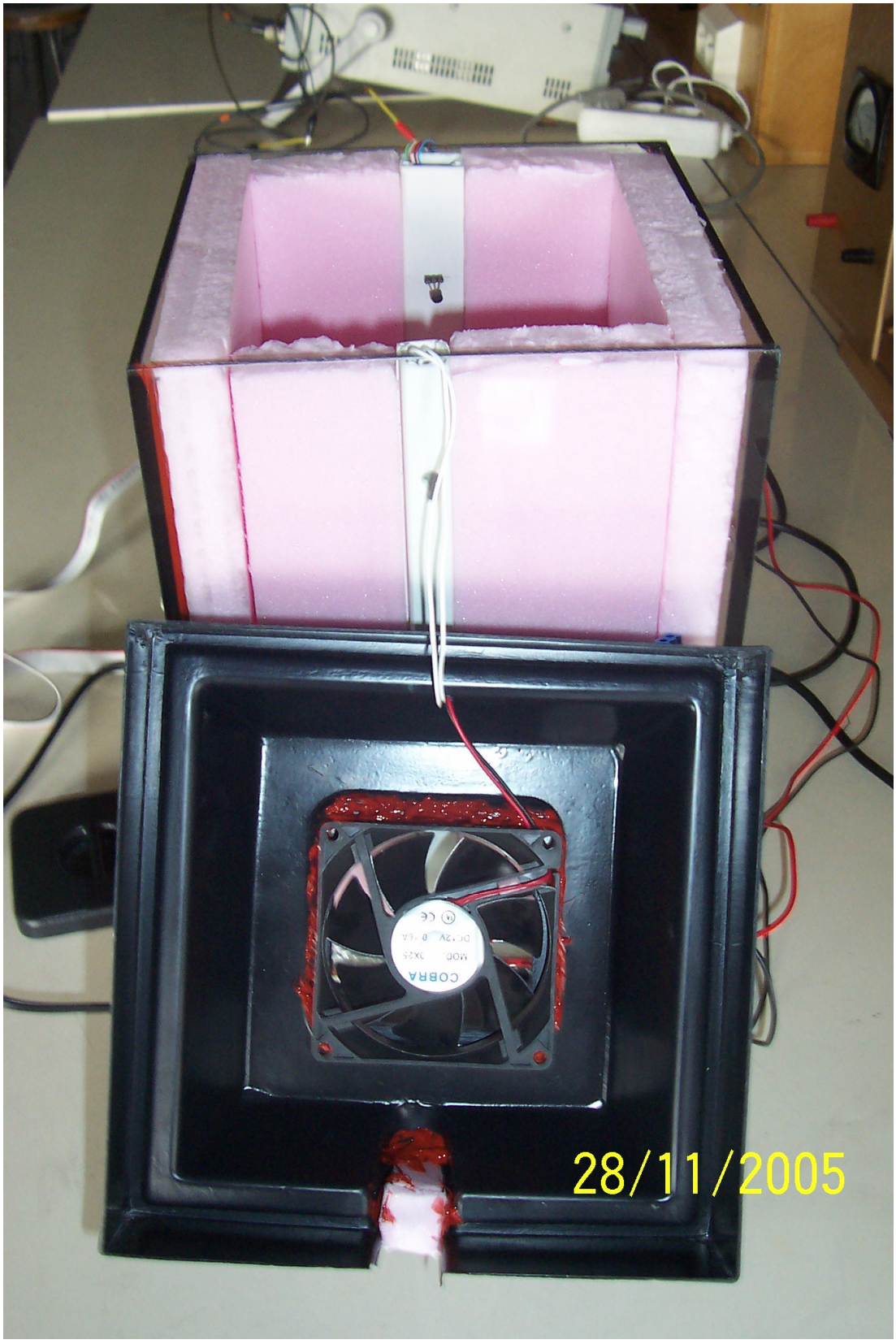
1–WIRE PRESENCE DETECT



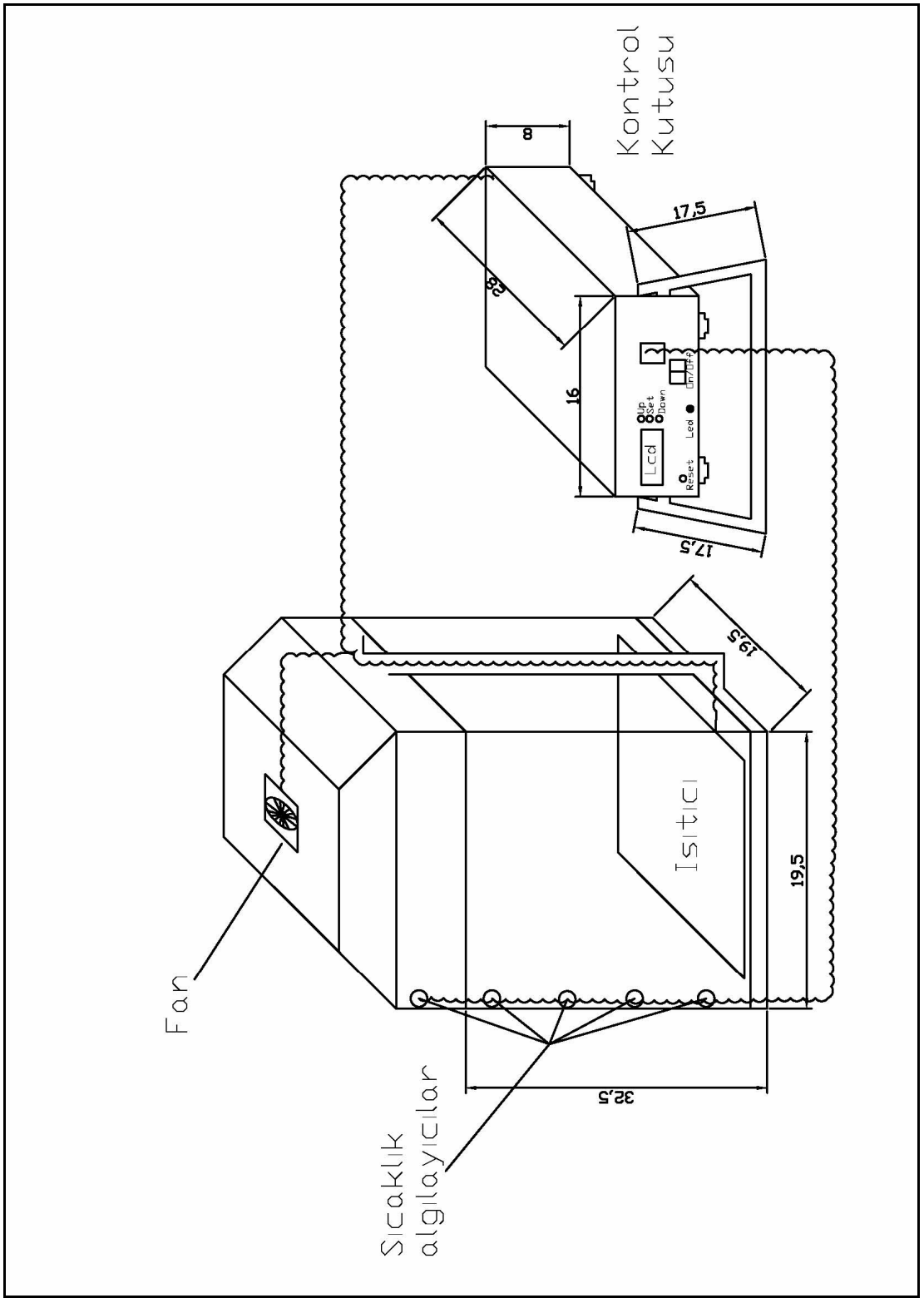
EK-3. Sistemin teknik çizimi ve deneysel çalışma fotoğrafları

Resim Ek-3.1. Isıl sistem genel görünümü

Resim Ek-3.2. Tesis İç Görünümü



Resim Ek-3.3. Tesis üst görünümü



Resim Ek-3.4. Isıl sistem teknik çizimi

ÖZGEÇMİŞ

Murat ALTIPARMAK, 07/09/1976 tarihinde Ankara'da doğdu. İlk, orta ve lise öğretimini Ankara'da tamamladı. 1993 yılında girdiği Gazi Üniversitesi Elektrik-Elektronik Mühendisliği Bölümü'nden Temmuz 1999'da mezun oldu. 1999-2000 yılları arasında GSM altyapı ekipmanları montaj mühendisi olarak çalıştı. 2000-2001 yılları arasında 275. Dönem Piyade Asteğmen olarak askerlik görevini tamamladı. Askerlik sonrası 2002-2003 Eğitim Öğretim yılında Gazi Üniversitesi Fen Bilimleri Enstitüsü'nde yüksek lisans öğrenimine başladı. Bu dönemde, sektörde Bilgi İşlem bölümü Sistem Yöneticisi olarak başladığı görevine halen Türk Telekom A.Ş., İnternet Veri Merkezi (IDC) TTNet E-Posta Grubu Sistem Yöneticisi olarak devam etmektedir. Bilgi teknolojileri ve İnternet, temel ilgi alanını oluşturmaktadır. Amacı internet tabanlı teknolojiler ile elektronik uygulamaları birleştiren model yazılımlar geliştirmektir.