

T.C.
İNÖNÜ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

LABVIEW VE RASPBERRY Pi 3 KULLANARAK SICAKLIK KONTROLÜ

 **ÖMER FARUK ÖZCAN**

YÜKSEK LİSANS TEZİ
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

HAZİRAN 2019

T.C.
İNÖNÜ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

LABVIEW VE RASPBERRY Pi 3 KULLANARAK SICAKLIK KONTROLÜ

ÖMER FARUK ÖZCAN

YÜKSEK LİSANS TEZİ
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

HAZİRAN 2019

Tezin Başlığı: LabVIEW ve Raspberry Pi 3 Kullanarak Sıcaklık Kontrolü

Tezi Hazırlayan: Ömer Faruk Özcan

Sınav Tarihi: 17. 06. 2019

Yukarıdaki adı geçen tez jürimizce değerlendirilerek Elektrik Elektronik
Mühendisliği Ana Bilim Dalında Yüksek Lisans Tezi olarak kabul edilmiştir.

Sınav Jüri Üyeleri

Tez Danışmanı: Prof. Dr. Nusret TAN

İnönü Üniversitesi

Prof. Dr. İbrahim KAYA

Dicle Üniversitesi

Dr. Öğr. Üyesi Furkan Nur DENİZ

İnönü Üniversitesi

Prof. Dr. Halil İbrahim ADIGÜZEL

Enstitü Müdürü

Aileme...



ONUR SÖZÜ

Yüksek Lisans Tezi olarak sunduğum “**LabVIEW ve Raspberry Pi 3 Kullanarak Sıcaklık Kontrolü**” başlıklı bu çalışmanın bilimsel ahlak ve geleneklere aykırı düşecek bir yardıma başvurmaksızın tarafımdan yazıldığını ve yararlandığım bütün kaynakların, hem metin içinde hem de kaynakçada yöntemine uygun biçimde gösterilenlerden oluştuğunu belirtir, bunu onurumla doğrularım.

Ömer Faruk ÖZCAN

İmza



ÖZET

Yüksek Lisans Tezi

LABVIEW VE RASPBERRY Pi 3 KULLANARAK SICAKLIK KONTROLÜ

Ömer Faruk ÖZCAN

İnönü Üniversitesi
Fen Bilimleri Enstitüsü
Elektrik Elektronik Mühendisliği Anabilim Dalı

71 + x sayfa

2019

Danışman: Prof. Dr. Nusret TAN

Günümüzde gelişen teknoloji ile birlikte kullanılan tüm cihazlarda kontrol sistemleri oldukça önem arz etmektedir. Endüstride ise özellikle sıcaklık kontrolü bir cihazın kararlı çalışması bakımından gereklidir. Sıcaklığın istenen seviyede hassas bir şekilde kontrolü ve uzaktan takip edilmesine ihtiyaç duyulmaktadır. Bu tip sistemlerin kontrolü için pek çok yöntem mevcut olup en çok kullanılan yöntem ise PID kontrol yöntemidir.

Bu tez çalışmasında LabVIEW ve Raspberry Pi 3 kullanarak PID kontrolör ile gerçek zamanlı sıcaklık ölçüm ve kontrol uygulaması gerçekleştirilmiştir. Bu uygulamanın sıcaklık ölçümü DS18B20 sensörü ile yapılmıştır. Uygulamada kullanılan su ısıtıcısının modeli LabVIEW ve Raspberry Pi 3 kullanarak gerçek zamanlı sıcaklık verileri kullanılarak hesaplanmıştır. Elde edilen model için Ziegler-Nichols metodundan yararlanılarak PID katsayıları hesaplanmıştır. Tasarlanan kontrolörün performansı Matlab/Simulink ortamında incelenmiştir. Sistemin gerçek zamanlı kontrolü için Raspberry Pi 3 üzerinde PID kontrolör algoritması yazılımı gerçekleştirilmiştir.

Tez çalışmasının devamında son yıllarda popülaritesi artan kesir dereceli $PI^{\lambda}D^{\mu}$ kontrol yöntemi bu sıcaklık kontrol sistemi için uygulanmıştır. Matlab ortamında geliştirilen IFTM (Inverse Fourier Transform Method) metodu kullanarak sistemin zaman cevabı elde edilmiştir. Tamsayı dereceli PID yöntemi ile elde edilen simülasyon sonuçları ile karşılaştırıldığında kesir dereceli $PI^{\lambda}D^{\mu}$ kontrol yönteminin yerleşme süresine göre daha başarılı sonuç verdiği görülmüştür.

ANAHTAR KELİMELELER: Labview, Raspberry Pi 3, Sıcaklık kontrolü, PID.

ABSTRACT

Master Thesis

TEMPERATURE CONTROL USING LABVIEW AND RASPBERRY Pi 3

Ömer Faruk ÖZCAN

Inonu University
Graduate School of Natural and Applied Science
Department of Electrical-Electronics Engineering

71 + x pages

2019

Supervisor: Prof. Dr. Nusret TAN

Control systems are very important in all devices used in today's developing technology. In the industry, temperature control is necessary in terms of stable operation of a device. It is needed to control the temperature precisely at the desired level and to monitor it remotely. There are many methods for controlling these types of systems and the most common method is PID control.

In this thesis, real-time temperature measurement and control application was performed with PID controller using LabVIEW and Raspberry Pi 3. The temperature measurement of this application was made with the DS18B20 sensor. The model of the water heater used in the application was calculated using real-time temperature data using LabVIEW and Raspberry Pi 3. PID coefficients were calculated using the Ziegler-Nichols method for the model obtained. The performance of the designed controller was examined in Matlab/Simulink environment. PID controller algorithm software was performed on Raspberry Pi 3 for real-time control of the system.

In the continuation of the thesis study, fractional order $PI^\lambda D^\mu$ controller which has become popular in recent years, has been applied for this temperature control system. The time response of the system was obtained using the IFTM (Inverse Fourier Transform Method) method developed in Matlab environment. Compared to the simulation results obtained by the integer PID method, the fractional PID control method was found to be more successful according to settling time.

KEYWORDS: Labview, Raspberry Pi 3, Temperature Control, PID.

TEŞEKKÜR

Tez çalışmam boyunca bilgi birikiminden her zaman faydalandığım, yüksek lisans süresince yaptığım çalışmalarımın her safhasında yardımlarını ve önerilerini esirgemeyen, beni her zaman cesaretlendiren ve yol gösteren değerli danışman hocam Sayın Prof. Dr. Nusret TAN'a en içten teşekkürlerimi sunarım. Tez çalışmam boyunca değerli bilgileriyle beni sabırla yönlendiren Sayın Dr. Öğr. Üyesi Ali YÜCE'ye teşekkürü bir borç bilirim.

Ayrıca bugünlere gelmemde büyük pay sahibi olan, hayatımın her aşamasında beni hep destekleyen anne ve babama, bu süreçte hep yanımda olan eşim Sena ÖZCAN'a en içten teşekkürlerimi bir borç bilirim.

İÇİNDEKİLER

ÖZET	i
ABSTRACT	ii
TEŞEKKÜR	iii
İÇİNDEKİLER	iv
SİMGELER VE KISALTMALAR	vii
ŞEKİLLER DİZİNİ	viii
ÇİZELGELER DİZİNİ	x
1. GİRİŞ	1
2. SICAKLIK ÖLÇÜMÜ VE KONTROLÜ	4
2.1. Sıcaklık Ölçümü.....	4
2.1.1. Direnç Termometreleri (RTD)	5
2.1.2. Isıl Çiftler (Termokupl).....	6
2.1.3. Termistörler.....	7
2.1.4. Entegre Devre Sıcaklık Sensörleri	8
2.2. Kontrol Sistemleri.....	10
2.3. Kontrol Sistemlerinin Türleri.....	10
2.3.1. Açık Çevrim Kontrol Sistemleri	11
2.3.2. Kapalı Çevrim Kontrol Sistemleri	11
2.4. Sıcaklık Kontrol Yöntemleri.....	12
2.4.1. Aç-kapa (On-Off) Kontrol	12
2.4.2. Oransal (P) Kontrol.....	13
2.4.3. Oransal İntegral (PI) Kontrol	14
2.4.5. Oransal İntegral Türevsel (PID) Kontrol	15
3. GERÇEKLEŞTİRİLEN KONTROL SİSTEMİNİN DONANIMI	20
3.1. Giriş.....	20
3.2. Raspberry Pi 3.....	21
3.2.1. Raspberry Pi 3 Dilleri	24
3.3. LabVIEW	25
3.4. LabVIEW ve Raspberry Pi 3 Haberleşmesi.....	26
3.4.1. LabVIEW 2014 SP1 Kurulumu	27
3.4.2. VI Package Manager Kurulumu	27
3.4.3. QwaveSys RPI Package Kurulumu.....	28
3.4.4. MakerHub ve LINX Toolbox Kurulumu	29
3.4.4.1. LabVIEW ve RPI 3 Haberleşmesi Örnek Uygulama.....	36

3.5.	DS18B20 Sıcaklık Sensörü	38
3.6.	Isıtıcı	39
3.6.1.	Isıtıcının Güç Sürücü Devresi	39
4.	LABVIEW VE RASPBERRY Pİ 3 KULLANARAK GERÇEK ZAMANLI BİR SICAKLIK KONTROL UYGULAMASI	41
4.1.	Giriş.....	41
4.1.1.	Raspberry Pi 3 ile Sıcaklık Ölçümü	41
4.1.2.	Sıcaklık Verilerinin LabVIEW'e Kablosuz Aktarılması	45
4.1.3.	Isıtıcının Modellenmesi ve PID Katsayılarının Bulunması.....	46
4.1.4.	Sıcaklık Kontrol Sisteminin Matlab/Simulink ile Modellenmesi	49
4.1.5.	Raspberry Pi 3 ile Sisteminin Gerçek Zamanlı Kontrolü.....	50
5.	SİSTEMİN KESİRLİ DERECELİ PID ($PI^{\lambda}D^{\mu}$) İLE KONTROLÜ	55
5.1.	Giriş.....	55
5.2.	Tamsayı Dereceli PID ve Kesir Dereceli PID ($PI^{\lambda}D^{\mu}$) Kontrolün Karşılaştırılması	55
5.3.	Sistemin Kesir Dereceli PID ($PI^{\lambda}D^{\mu}$) Yöntemi ile Kontrolü.....	57
6.	SONUÇLAR	61
7.	KAYNAKLAR	62
8.	EKLER.....	66
	ÖZGEÇMİŞ	70

SİMGELER VE KISALTMALAR

RPI	Raspberry Pi
PIC	Peripheral Interface Controller
RTD	Resistance Temperature Detector
PTC	Pozitif Sıcaklık Katsayısı (Positive Temperature Coefficient)
NTC	Negatif Sıcaklık Katsayısı (Negative Temperature Coefficient)
LabVIEW	Laboratory Virtual Instrument Engineering Workbench
NI	National Instruments
VI PM	VI Package Manager
I/O	Giriş/Çıkış (Input/Output)
LM35	Precision Centigrade Temperature Sensors
PWM	Darbe Genişlik Modulasyonu (Pulse Width Modulation)
ADC	Analog Dijital Dönüştürücü (Analog Digital Converter)
P	Oransal (Proportional) kontrolör
PI	Oransal-İntegral (Proportional-Integral) kontrolör
PD	Oransal- Türev (Proportional- Derivative) kontrolör
PID	Oransal-İntegral-Türev (Proportional-Integral-Derivative)kontrolör
$PI^{\lambda}D^{\mu}$	Kesir Dereceli PID kontrolör (Fractional Order PID Controller)
IFTM	Ters Fourier Dönüşüm Metodu(Inverse Fourier Transform Method)
K_p	Oransal Kazanç (Proportional Gain)
K_i	İntegral Kazanç (Integral Gain)
K_d	Türevsel Kazanç (Derivative Gain)
T_i	İntegral Zaman Sabiti (Integral Time Coefficient)
T_d	Türev Zamanı Sabiti (Derivative Time Coefficient)
λ	Kesirli İntegralin Derecesi
μ	Kesirli Türevin Derecesi

ŞEKİLLER DİZİNİ

Şekil 2. 1	RTD	5
Şekil 2. 2.	Termokupl	6
Şekil 2. 3.	Termokupl Çalışma mantığı	7
Şekil 2. 4.	NTC-PTC	7
Şekil 2. 5.	NTC ve PTC çalışma mantığı.....	8
Şekil 2. 6.	LM35 entegre devresi.....	9
Şekil 2. 7.	DS18B20 entegre devre ve kablolu modeli.....	10
Şekil 2. 8.	Basit bir kontrol sistemi blok diyagramı	10
Şekil 2. 9.	Açık çevrim kontrol sisteminin blok diyagramı	11
Şekil 2. 10.	Kapalı çevrim kontrol sisteminin blok diyagramı	12
Şekil 2. 11.	Histerisizli aç-kapa kontrol eğrisi	13
Şekil 2. 12.	Oransal Kontrol Eğrisi	14
Şekil 2. 13.	Oransal integral kontrol eğrisi	14
Şekil 2. 14.	Oransal türevsel kontrol eğrisi	15
Şekil 2. 15.	Oransal integral türevsel kontrol eğrisi	15
Şekil 2. 16.	PID katsayılarının etkisinin grafiksel gösterimi	16
Şekil 2. 17.	PID kontrolörün blok diyagramı	17
Şekil 2. 18.	Açık çevrim kontrol sisteminin ideal basamak cevabı	18
Şekil 3. 1.	Sistemin Genel Görünümü	20
Şekil 3. 2.	Gerçekleştirilen kontrol uygulamasının blok diyagramı	21
Şekil 3. 3.	a) RPI 3 ön yüzü b) RPI 3 arka yüzü	22
Şekil 3. 4.	Python Dilleri	24
Şekil 3. 5.	LabVIEW ve geleneksel programlama kodu karşılaştırması	26
Şekil 3. 6.	LabVIEW 2014	27
Şekil 3. 7.	VI Package Manager	27
Şekil 3. 8.	Qwavesys RPI package kurulumu.....	28
Şekil 3. 9.	Qwavesys RPI package LabVIEW'e eklenmesi	28
Şekil 3. 10.	DS18B20 sensörün kütüphanesinin LabVIEW'e eklenmesi.....	29
Şekil 3. 11.	VI PM'da Makerhub Toolbox'ı Yükleme	29
Şekil 3. 12.	Makerhub Toolbox'ın LabVIEW'e eklenmesi.....	30
Şekil 3. 13.	VI PM'den Digilent LINX'i Yükleme	30
Şekil 3. 14.	RPI de IP adresi öğrenme	31

Şekil 3. 15.	LabVIEW’de RPI eşleştirme	31
Şekil 3. 16.	RPI’nin IP, username ve password girme	32
Şekil 3. 17.	a) RPI IP adresinin eşleşmesi b) RPI yazılımının kurulması	32
Şekil 3. 18.	a) LabVIEW programının açılması b) Yeni boş proje oluşturma	33
Şekil 3. 19.	a) Hedef aygıt arama b) Hedeflenen aygıtın seçilmesi.....	34
Şekil 3. 20.	a) RPI’ye bağlanma b) RPI’de bağlantının olduğunu gösteren ikon.....	35
Şekil 3. 21.	Örnek uygulama front panel	36
Şekil 3. 22.	Örnek uygulamanın block diyagramı	36
Şekil 3. 23.	Open fonksiyonu	37
Şekil 3. 24.	Digital Write fonksiyonu	37
Şekil 3. 25.	Close Fonksiyonu	37
Şekil 3. 26.	RPI 3’e bağlı led’i yakma.....	38
Şekil 3. 27.	DS18B20 sıcaklık sensörü.....	38
Şekil 3. 28.	12V Isıtıcı	39
Şekil 3. 29.	Mosfet Sürücü kartı	40
Şekil 4. 1.	RPI 3 ile sıcaklık ölçümü	42
Şekil 4. 2.	RPI’ye DS18B20 haberleşme modul kurulumu	43
Şekil 4. 3.	RPI’ye DS18B20 haberleşme modul kurulumunun tamamlanması.....	44
Şekil 4. 4.	RPI 3 ile sıcaklık ölçümü	44
Şekil 4. 5.	Sistemin LabVIEW’de programlanması	45
Şekil 4. 6.	LabVIEW’de elde edilen sıcaklık-zaman grafiği	45
Şekil 4. 7.	Isıtıcının deney sonucu elde edilen açık çevrim basamak cevabı	46
Şekil 4. 8.	Hesaplanan kapalı çevrim kontrol sisteminin blok diyagramı	48
Şekil 4. 9.	Sıcaklık kontrol sisteminin Matlab/Simulink’te modellenmesi	49
Şekil 4. 10.	Sıcaklık kontrol sisteminin basamak cevabı.....	49
Şekil 4. 11.	Yeni PID kontrolörle sistemin modeli.....	50
Şekil 4. 12.	Yeni PID kontrolörle elde edilen basamak cevabı	50
Şekil 4. 13.	Sıcaklık kontrol sistemin genel görünüşü.....	51
Şekil 4. 14.	Gerçekleştirilen sıcaklık kontrol sisteminin bağlantı şeması	52
Şekil 4. 15.	Raspberry Pi 3 üzerinde yapılan yazılımın akış diyagramı	52
Şekil 4. 16.	Sistemin 30 ve 40 derece set değeri için sonuçları.....	53
Şekil 4. 17.	30 derece set değerinde kontrol edilen sistemin basamak cevabı	54
Şekil 4. 18.	40 derece set değerinde kontrol edilen sistemin basamak cevabı	54
Şekil 5. 1.	PI ^λ D ^μ kontrolörün blok diyagramı.....	56

Şekil 5. 2.	λ ve μ parametrelerinin değişimine göre denetleyici tipleri	56
Şekil 5. 3.	Kesir dereceli PID kontrolör kapalı çevrim diyagramı	58
Şekil 5. 4.	Hesaplanan kesir dereceli $PI^\lambda D^\mu$ kontrolörün kapalı çevrim sisteminin blok diyagramı.....	59
Şekil 5. 5.	Kesir dereceli $PI^\lambda D^\mu$ kontrol yöntemi elde edilen sonuçların tamsayı dereceli PID kontrolör yöntem ile karşılaştırılması.....	60



ÇİZELGELER DİZİNİ

Çizelge 2.1.	PID katsayılarının etkisi	16
Çizelge 2.2.	Ziegler-Nichols PID parametre tablosu	19
Çizelge 4.1.	Sıcaklık-zaman verileri.....	47



1. GİRİŞ

Endüstriyel kontrol sistemlerinin geçmişi çok eski değildir. İlk organize kontrol sistemi sanayi devriminin sonrasında İngiltere’de 18. yüz yılın sonlarında olduğu varsayılır. Son yıllarda kontrol sistemleri, her alanda insanlığın ve uygarlığın gelişmesinde çok önemli rol oynamıştır. Otomatik kontrol sistemleri kullanılan ekipmanların dışarıdan bir insan müdahalesi olmadan sistemi kendi başına denetleyerek belirlenen değer aralıklarında kararlı olarak çalışmasını sağlayan sistemlerdir. Bu sistemler hataları en az seviyeye indirerek sistemin etkin ve kararlı bir şekilde çalışmasını sağlarlar [1].

Bugün otomatik kontrol sistemlerinin yardımı ile ev, ofis ve birçok kapalı mekânın ısıtma ve havalandırma sistemleri, ortamın ısını ve nemini ayarlar. Günümüzde endüstriyel sistemlerde, kullanılan modern ekipmanlarda, otomatik kontrol sistemlerinin her alanda fazla sayıda uygulamaları mevcuttur. Örnek olarak üretimi yapılan ürünlerin kalite kontrolleri, ilaç sanayisinde ilaçların kontrolü, arabaların otonom kontrolü, uçakların oto-pilot ile kontrolü, trenlerin kontrolü, trafik lambalarının kontrolü, kimya endüstrisinde üretilen ürünlerin kontrolü, tank içerisindeki sıvının sıcaklık kontrolü verilebilir.

Endüstride en çok kullanılan kontrol sistemlerden birisi de sıcaklık kontrol sistemleridir. Sıcaklık kontrolünde ortamın sıcaklık değerinin istenilen aralıklarda tutulması son derece önemlidir. Sıcaklığın istenen seviyelerin dışına çıkması cihazlarda karakteristik özelliklerin bozulmasına neden olabilmektedir. Bunun sonucunda ise sistemler kararsız davranışlar gösterebilmektedir. Bu kararsız davranışlar işleyişin bozulmasına ve maddi zararlara sebebiyet vermektedir [2].

Sıcaklık kontrolünde uygulama alanına göre uygulanan teknikler mevcuttur. Bu teknikler sıcaklık seviyesi, ortamın fiziksel ve kimyasal yapısı, hassasiyet, okuma doğruluğu ve hızı, güvenlik ve kontrol çıkışının tipine göre değişmektedir. Sıcaklık bilgisinin elektriksel işaretlere dönüştürülmesinde günümüzde sıklıkla termokupl, RTD, termistör, yarı iletken tabanlı sıcaklık algılayıcıları ve pirometre elemanlarından yararlanılmaktadır. Elektriksel bilgi elde edildikten sonra sayısal veya analog kontrol devreleri ile istenilen kontrol yöntemi uygulanır.

Sıcaklık ve kontrol uygulamalarında genellikle PIC mikrodenetleyicileri kullanılmıştır. Sıcaklık kontrolünde çoğunlukla kapalı çevrim kontrol modeli ile on-off, P, PD, PI ve PID kontrol yöntemleri kullanılmaktadır. Bu kontrol yöntemlerinin

birbirlerine karşı avantajları ve dezavantajları olup kontrol yöntemlerin tercihi kontrol edilecek sistemlerin hassasiyetine ve kullanım alanlarına göre yapılmaktadır.

Yapılan literatür taramalarında, çok sayıda endüstride kullanılan sıcaklık kontrol uygulamaları mevcuttur.

Bu çalışmalardan birinde Metin KAPIDERE, Raşit AHISKA, İnan GÜLER dört sıcaklık algılayıcı ve mikrodenetleyicili termohipoterm sistemi ile kanser tedavisinde kullanılan sıcaklık kontrollü kask cihazı tasarımı ele alınmıştır. Kontrol katmanında PIC16F877 mikrodenetleyicisi ile darbe genişlik modülasyonu (PWM), sıcaklık algılayıcısı olarak ise T tipi termokupllar kullanılmıştır [3].

Başka bir çalışmada Saadettin AKSOY ve Aydın MÜHÜRÜ, mikrodenetleyici tabanlı süt soğutma tankının soğutma kontrolü çalışmasını yapmışlardır. PTC sıcaklık algılayıcı ve On-Off kontrol metodu kullanarak süt sıcaklığının istenilen sıcaklıklarda tutulmasını sağlamışlardır [4].

2007 yılında Savaş ŞAHİN, Yalçın İŞLER ve Mustafa Berkant SELEK, sanal aygıtlarda sıvı seviyesi ve sıcaklık kontrolü deneyinde bir tankın içindeki suyun sıcaklığını termokupl ve PIC mikro denetleyicisi kullanarak on-off kontrol yapmışlardır. [5].

Murat ÇELİK yüksek lisans tez çalışmasında, fırın içerisinde 5 ayrı bölgeye yerleştirilen S tipi termokupl sıcaklık algılayıcılarından elde ettiği ortalama sıcaklık değerini bilgisayar ortamında gözlemlemiş ve PID kontrol metodu ile üfleme fanını bilgisayar ortamında kontrol etmiştir [6].

Erdal BERBER 2008 yılında, mikrodenetleyicili endüstriyel otomatik sıcaklık ölçme ve kontrol sistemini yüksek lisans tezi olarak yapmıştır. Sıcaklık ölçme işlemi DS18B20 ve J tipi termokupl sıcaklık algılayıcıları kullanılmıştır. Sistem on-off kontrol metodu kullanarak PIC16F877A mikrodenetleyici üzerinden programlanmış ve LabVIEW programı ile integrasyonu gerçekleştirilmiştir. LabVIEW programı üzerinde kullanıcı tarafından ilgili butonlar kullanılarak ayarlanan sıcaklık alt ve üst limit aralığının dışında olması durumunda sistemin alarm vermesi sağlanmıştır [7].

2011 yılında Ali YÜCE tarafından gerçekleştirilen yüksek lisans tezinde sanal aygıtlar (LabVIEW) kullanılarak bir tank içerisindeki suyun sıcaklığı PID kontrol metodu ile kontrol edilmiştir [8].

Bu tez çalışmasında LabVIEW ve Raspberry Pi 3 kullanarak PID kontrolör ile gerçek zamanlı bir sıcaklık ölçüm ve kontrol sistemi uygulaması yapılmıştır. Isıtıcıda bulunan suyun sıcaklık ölçümü, DS18B20 dijital sıcaklık sensörü kullanılarak Raspberry

Pi 3 üzerinden yapılmıştır. Toplanan sıcaklık verileri Raspberry Pi 3' ten LabVIEW'e Wi-Fi yolu ile kablosuz olarak aktarılmış ve sıcaklık-zaman grafiğinin LabVIEW ortamında çizimi sağlanmıştır. Isıtıcının açık çevrim basamak cevabı olan sıcaklık-zaman grafiğinin üzerinden ısıtıcının transfer fonksiyonu hesaplanmıştır. Isıtıcı transfer fonksiyonundan Ziegler-Nichols PID parametre tablosu yardımı ile PID katsayıları hesaplanmış ve PID kontrolörün transfer fonksiyonu elde edilmiştir. PID kontrolörlü kapalı çevrim kontrol sisteminin blok diyagramı Matlab/Simulinkte modellenmiş ve hesaplanan PID parametreleri civarında seçilen başka değerler için sistemin birim basamak cevabı performansı incelenmiştir. Başarılı performans gösteren PID katsayıları kullanılarak Raspberry Pi 3 ile gerçek zamanlı bir sıcaklık kontrol uygulaması gerçekleştirilmiştir. Tez çalışmasının devamında son yıllarda popüleritesi artan kesir dereceli kontrol yöntemi bu sıcaklık kontrol sistemi için uygulanmıştır. Matlab ortamında geliştirilen IFTM (Inverse Fourier Transform Method) metodu kullanarak sistemin zaman cevabı elde edilmiştir. Tam sayı dereceli PID yöntemi ile elde edilen simülasyon sonuçları ile karşılaştırıldığında kesir dereceli PID kontrol yönteminin yerleşme zamanına göre daha başarılı sonuç verdiği görülmüştür.

Literatür özeti ve yapılan çalışmaların özetlendiği giriş bölümünden sonra ikinci bölümde yaygınca kullanılan sıcaklık algılayıcılarından ve sıcaklık kontrol sistemlerinin özellikleri ve kontrol metotlarından bahsedilmiştir.

Üçüncü bölümde LabVIEW ve Raspberry Pi 3 kullanılarak yapılan sıcaklık ölçüm ve kontrol uygulamasının donanımı ve tasarlanan devre üzerinde kullanılan komponentlerin temel özelliklerinden ve çalışma prensiplerinden bahsedilmiştir.

Dördüncü bölümde LabVIEW ve Raspberry Pi 3 kullanarak PID kontrolör ile gerçek zamanlı bir sıcaklık ölçümü yapılarak sistem modeli hesaplanmıştır ve Matlab/Simulink ortamında gerçek zamanlı ısıtıcı kontrol sisteminin simülasyonu yapılmıştır. Simülasyon sonucunda elde edilen PID katsayıları kullanılarak Raspberry Pi 3 ile hazırlanan PID kontrolör algoritması yazılımı ile sıcaklığın istenen değerde hassas bir şekilde kontrolü gerçekleştirilmiştir.

Beşinci bölümde ise sistemin kesirli dereceli PID ile kontrolü anlatılmıştır.

Altıncı ve son bölümde tez çalışması boyunca elde edilen sonuçlar ve yorumlara yer verilmiştir.

2. SICAKLIK ÖLÇÜMÜ VE KONTROLÜ

2.1. Sıcaklık Ölçümü

Bu bölümde sıcaklık ölçüm teknikleri ve sıcaklık ölçümlerinde kullanılan sıcaklık algılayıcıları ele alınacaktır. Ortamdaki ısıнын değişimini algılayan cihazlara ısı ya da sıcaklık sensörü adı verilir. Fiziksel, kimyasal, biyolojik, mekanik ve elektronik gibi tüm sistemler sıcaklıktan etkilendiğinden dolayı sıcaklık en çok ölçülen parametrelerin başında gelmektedir.

Teknolojinin endüstriyele olan katkısının sürekli olarak geliştiği günümüzde sıcaklık, hemen her üretim ortamı için en önemli değişken olup endüstride kontrol edilen sistemlerin verimli olarak çalışması için sıcaklığın ölçülmesi ve belli değerlerde tutulması önemlidir.

Bir diğer önemli konu ise kullanılacak olan sıcaklık sensörlerinin seçimidir. Sıcaklık sensörü seçiminde birçok parametre vardır. Bu parametreler; sensörün nerede kullanılacağına karar verilmesi, sensör hassasiyeti, ölçümün aralığı, sensör kararlılığı, tepki (cevap verme) süresi ve maliyeti olarak sınıflandırılır [9, 10].

- **Sensörün nerede kullanılacağına karar verilmesi:** Belirlememiz gereken ilk şey sensörden doğru bilgileri alabilmek için sıcaklık sensörünün nerede kullanılacağıdır.
- **Sıcaklık okuma hassasiyeti:** Tüm sensörler farklı hassasiyetlerde üretilmiş olup ölçülen sıcaklık değerinin istenen gerçek sıcaklık değerine yakınlığı olarak tanımlanır. Örneğin bir ortamda sıcaklık sensörünün sıcaklık okuma hassasiyeti $\pm 3^{\circ}\text{C}$ ise ve sensörden okunan sıcaklık değeri 27°C olarak okunuyorsa bu termometrede okunan değeri bu aralıkta değerlendirmek zorundayız. Yani ortamın sıcaklığı 24°C de olabilir 30°C de olabilir diyebiliriz.
- **Ölçüm yapılacak sıcaklık aralığı:** Sensörler üretilirken belirli bir ölçüm aralığını algılayabilecek şekilde imal edilirler. Sensörü seçmeden önce kullanılacak yere göre uygun olup olmadığının iyi tespit edilmesi gerekmektedir. Çünkü uygun ölçüm aralığına sahip sensör seçimi ölçümün doğruluğu ve maliyeti düşürmek açısından önemlidir. Örneğin bir kaynak atölyesinde proses aralığı $500-900^{\circ}\text{C}$ olması gereken yere $700-1100^{\circ}\text{C}$ arasında ölçüm değeri veren bir sıcaklık sensörü kullanmak akıllıca değildir.
- **Sensörün kararlılığı:** Sensörlerin kararlılığı sensörün kalitesini gösteren bir olgudur. Kararlılık belirli bir zaman periyodu boyunca giriş değerleri ile çıkış değerlerinin

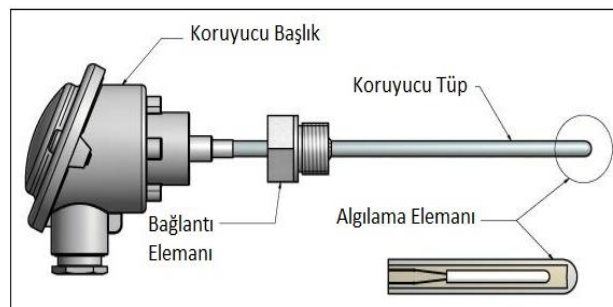
aynı sonuçları vermesi veya bu değerler arasındaki farkın birbirine çok yakın değerde olmasıdır. Bir başka deyişle sensör ile sabit bir ölçüm yapılıyorken hep aynı değeri ölçebiliyor olması sensörün tam kararlı olduğunun göstergesidir.

- **Tepki (cevap verme) Süresi:** Cevap verme süresi, ölçüm yapılarak elde edilen verinin algılanıp bağlı olduğu ilgili yere bu verileri aktarma işlemi boyunca geçen süredir. Her sensörün cevap verme süresi farklı olup milisaniyeler cinsinden ifade edilmektedir.
- **Uygulamanın doğruluk derecesine ve cihazın monte ediliş şekline göre farklılık gösteren maliyet:** Sensör seçimi yaparken fiyatlarına da dikkat etmemiz gerekir. Bu maliyet yapılan işe göre değişmekte olup sensör maliyetlerini minimuma indirmek için amaca uygun en doğru sensörü seçmemiz gerekmektedir. Çünkü kullanım amacı dışında seçilen bir sensör yetersiz kalabilir. Bu durum sistemin düzgün çalışmasına engel olabilir ve maliyeti de artırır.

En çok kullanılan sıcaklık sensörleri: dirençsel sıcaklık sensörleri (RTD-Resistance Temperature Detector), ısıl çiftler (termokupl), termistörler (NTC, PTC) ve entegre devre sıcaklık sensörleridir (LM35, DS18B20, vb).

2.1.1. Direnç Termometreleri (RTD)

Şekil 2.1’de yapısı gösterilen RTD, hassasiyeti yüksek olup ölçüm aralığı -200°C ile $+850^{\circ}\text{C}$ arasında değişim gösterir. İletken telinin direnç değerinin sıcaklıkla değişiminden meydana gelen sıcaklık sensörleridir. Çalışma prensibi bir sarımlı direnç, sıcaklık değeri ölçülecek ortamın içine bırakılır ve direncin üzerinden sabit bir akım geçirilir. Sıcaklık değerinin değişmesi ile birlikte sarımlı direncin direnç değeri değişerek ve direnç üzerinden geçen sabit akımla değişmekte olan bir gerilim değeri elde edilir. RTD sıcaklık algılayıcılarının direnç değeri ne kadar yüksek ise sistemdeki hata payı da o kadar düşüktür.



Şekil 2.1. RTD [11]

RTD imalatında en çok kullanılan maddeler; bakır, demir, nikel ve platindir. Bu maddelerin içerisinde platin, en ideal olanı ve doğrusal sonuçları veren maddedir.

Platin RTD'lerin direnç değerleri, 10 Ohm ile 1000 Ohm arası tel sarımlı veya ince tabakalı RTD olma durumuna göre değişmektedir. RTD sıcaklık algılayıcıları adlandırılırken 0°C'daki direnç değerlerine ve kullanıldığı elementin cinsine göre adlandırılmıştır. (PT100, PT1000...). En çok bilinen (PT100) değer ise 0°C' de 100 Ohm'dur [11].

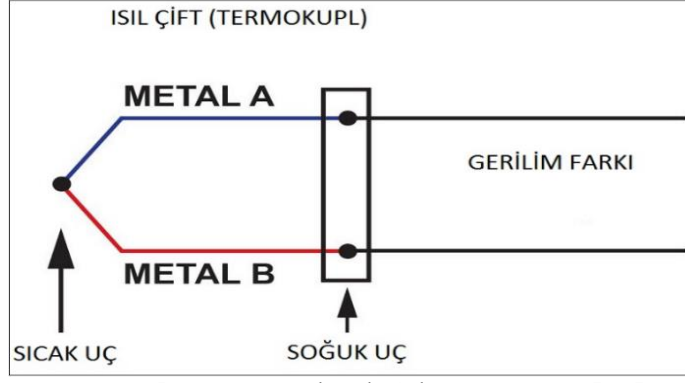
2.1.2. Isıl Çiftler (Termokupl)

Termometreler yüksek sıcaklık ölçümlerinde kullanılamadıkları için bunun yerine geniş ölçüm aralığı olan termokupllar kullanılır. Şekil 2.2'de gösterilen termokupllar endüstride en çok kullanılan ısı kontrol elemanlarından biri olup -200°C ile +2500°C aralığında çalışabilmektedirler. Örneğin endüstri tesislerinde yüksek sıcaklıklarda ölçüm yapılmak istenilen kömür kazanlarında ısı kontrolünde kullanılırlar.



Şekil 2.2. Termokupl [11]

1821 yılında Thomas Seebeck tarafından icat edilen termokuplların çalışma prensibi Şekil 2.3'te gösterildiği gibi demir ve bakır alaşımı gibi farklı iki metalin birleştiği nokta ısıtıldığında zaman iki uç arasında bir gerilim farkı oluşur. Oluşan bu gerilim farkının değeri ise iki ayrı metalin ısınması sonucundaki sıcaklık ve soğukluk değerinin farkına bağlıdır ve gerilim farkı kullanılarak istenilen sıcaklık değerleri ölçülür [11, 12].



Şekil 2.3. Termokupl Çalışma mantığı[12]

2.1.3. Termistörler

Sıcaklığı değiştikçe direnci de değişen sensörlere termistör denir. Yarı iletken malzemelerden üretilen termistörlerin yapımında çoğunlukla oksitlenmiş manganez, nikel, bakır veya kobaltın karışımı kullanılır. Termistörler küçük sıcaklık değişikliklerine karşı duyarlı olup ısınınca direnci değişen algılayıcılardır. Sınırlı sıcaklık aralıklarında ve özellikle düşük sıcaklık uygulamaları için uygundurlar. Şekil 2.4'te gösterildiği gibi fiziksel boyutları küçüktür ve iğne ucu kadar olabilir. Termistörler PTC ve NTC olarak ikiye ayrılırlar.

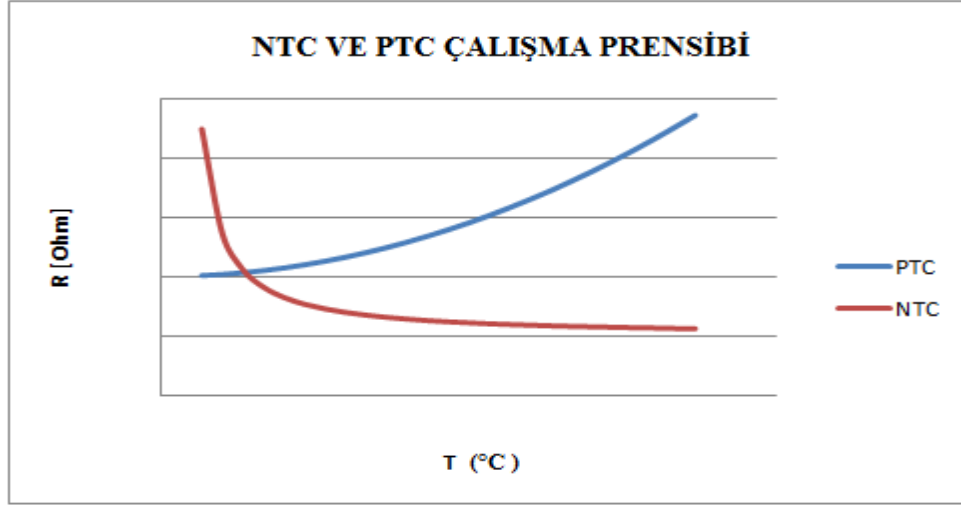


Şekil 2.4. NTC-PTC [12]

- **PTC (Positive Temperature Coefficient):** Sıcaklık değeri arttığında direnci de artan elektronik devre elemanlardır. PTC sensörleri -60°C ile $+150^{\circ}\text{C}$ aralığında $0,1^{\circ}\text{C}$ duyarlılıkta kararlı bir şekilde çalışabilirler. Genellikle elektrik motorlarının fazla ısınmasını engellemek ve sıcaklık seviyesini belirli noktada tutmak amacıyla kullanılırlar.
- **NTC (Negative Temperature Coefficient):** NTC'ler PTC termistörlerin tam tersi çalışma mantığı ile çalışırlar. Sıcaklık değeri arttığında direnci azalan devre

elemanlarıdır. NTC sensörleri -300°C ile $+50^{\circ}\text{C}$ aralığında $0,1^{\circ}\text{C}$ duyarlılıkta kararlı bir şekilde çalışabilirler. Kullanım alanları PTC' lere göre daha fazla olup elektronik termometrelerde, arabaların radyatörlerinde, amplifikatörlerin çıkış güç kartlarında kullanılırlar.

NTC ve PTC'nin çalışma mantığı Şekil 2.5'te gösterildiği gibi tek grafikte verilmiştir.



Şekil 2.5. NTC ve PTC çalışma mantığı [13]

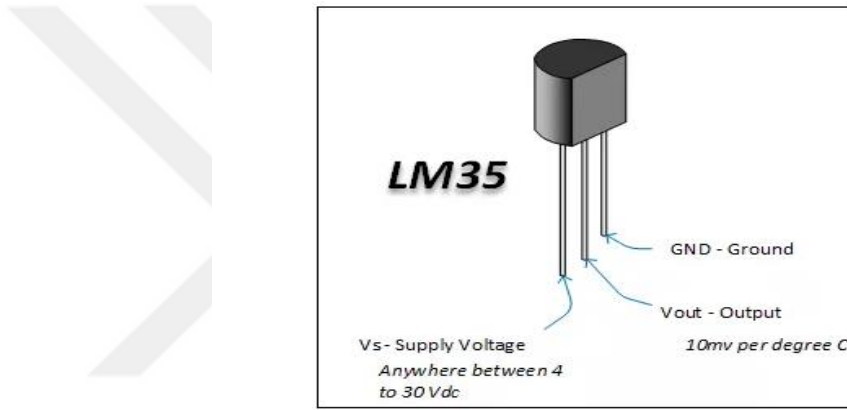
2.1.4. Entegre Devre Sıcaklık Sensörleri

Yarı iletken entegre devrelerin gelişmesiyle birlikte entegre devre sensörleri ortaya çıkmıştır. Germanyum ve silisyumun içerisine kristal karıştırılmasıyla meydana gelen sıcaklık algılayıcıları kullanılmaktadır. Oldukça küçük boyutlara sahiptir. Çalışma sıcaklık aralığı -50°C ile $+150^{\circ}\text{C}$ arasında olup, doğrusal çıkış değerlerine sahiptir. Yapısı gereği yüksek sıcaklıklarda ölçüm yapamayan yarı iletken sıcaklık sensörleri bulundukları devrede uygun olarak termal korumaya alınırlarsa, oldukça iyi ölçüm performansı sergilerler [13].

Çalışma prensipleri; transistörlerin akım karakteristiği üzerine kurulmuştur. Farklı kolektör akımlarına sahip iki transistörün sıcaklık değişimine bağlı olarak 'base-emitter' gerilimlerinin farkları da değişir ve bu fark, akım veya tek bir gerilime çevrilerek ölçüm alınmış olur [14].

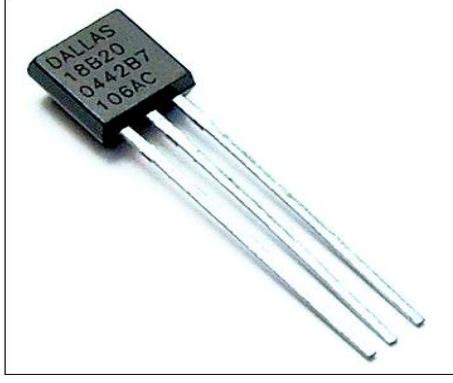
Ortam sıcaklığına göre, çıkış olarak akım veya gerilim değeri veren sensörlere analog sıcaklık sensörleri adı verilir. Genellikle, ikaz ile çalışması gerektiğinden güç kaynağına ihtiyacı vardır.

Bu kurala göre çalışan entegre devre sıcaklık algılayıcıları sıcaklık değerine göre gerilim ve akım üreten sensörler olarak ikiye ayrılır. Sıcaklık değerine bağlı olarak gerilim üreten algılayıcılar LM135, LM235, LM335 (Kelvin), LM35 - LM45 (Celsius), LM34 (Fahrenheit) olarak sınıflandırılır. Şekil 2.6'da yapısı gösterilen entegre devre LM35 -55°C ile +150°C arasındaki sıcaklıkların algılanmasında kullanılır. Sıcaklık değerinin her 1°C artışında yaklaşık 10 mV üretir. Sıcaklık değerine bağlı olarak akım üreten algılayıcılar ise LM134, LM234 ve LM334'dür. Bu algılayıcıların akım çıkışları sıcaklık ile orantılıdır. Bu algılayıcıların sıcaklık hassaslıkları bir dış direnç kullanarak ayarlanabilir. Sıcaklık hassasiyetleri 1 $\mu\text{A} / ^\circ\text{C}$ ile 3 $\mu\text{A} / ^\circ\text{C}$ aralığında değişmektedir [14].



Şekil 2.6. LM35 entegre devresi [14]

Dijital sıcaklık sensörleri ise analog sensörlerin tersine, dijital sinyal çıkışı verirler (1 ve 0 şeklinde). Yapısında bulunan sıcaklık sensörü, analog ölçüm sonucunu analog/dijital çeviriciye (ADC) gönderir. ADC'den çıkan dijital sinyaller seri haberleşme ara yüzleri ile mikro işlemcilere ulaşır. Mikro işlemci ise sinyalleri değerlendirerek gerekli işlemleri gerçekleştirir. Bu çalışmada kullanılan Maxim integrated firması tarafından geliştirilen DS18B20 sıcaklık sensörü sıcaklık bilgisini sayısal olarak vermekte olup analog/sayısal dönüştürücüsüne gerek duymamaktadır. DS18B20 sensörü -55°C ile +125°C arasında ölçüm yapabilmektedir. Şekil 2.7'de gösterildiği gibi DS18B20 sıcaklık sensörünün hem entegre devre hem de kablolu çeşitleri mevcuttur.



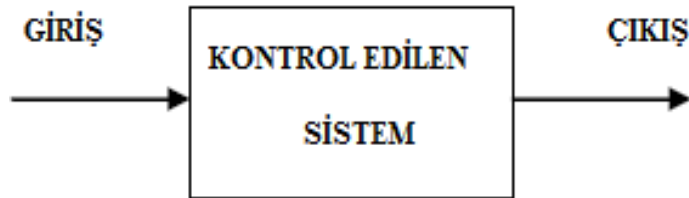
Şekil 2. 7. DS18B20 sıcaklık sensörünün entegre devre ve kablolu modeli

2.2. Kontrol Sistemleri

Kontrol: Bir sistemin önceden belirlenen ve istenen değer aralığında tutulması olarak tanımlanabilir.

Sistem: Belirli bir işlem için bir araya getirilerek karşılıklı etkileşim ve bağlantı içinde olup en az bir işlevi veya amacı bulunan elemanlar topluluğudur.

Kontrol Sistemi: İstenilen değer aralıklarında bir düzeneğin kontrolünün sağlanabilmesi amacı ile kendi içinde stratejiler üreten sistemlere kontrol sistemi denir. Bir giriş ve çıkışlı basit bir kontrol sisteminin blok diyagramı Şekil 2.8'deki gibidir [15].



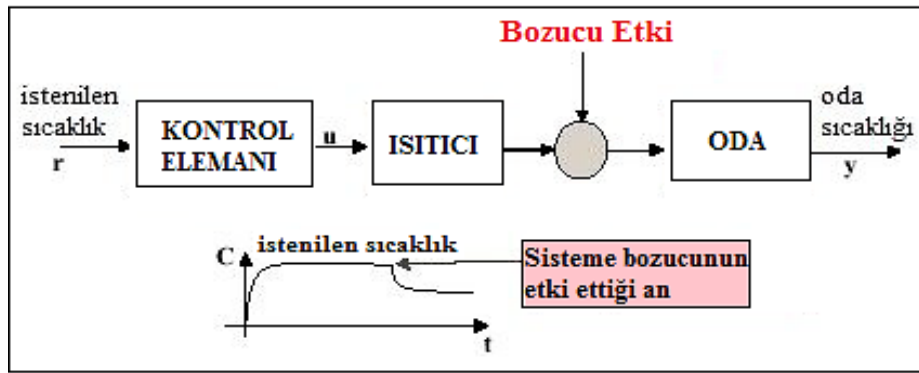
Şekil 2.8. Basit bir kontrol sistemi blok diyagramı [15]

2.3. Kontrol Sistemlerinin Türleri

Kontrol sistemleri çıkış işaretinin veya kontrol edilen sistemin kumanda edilmesi bakımından açık çevrim kontrol sistemleri ve kapalı çevrim kontrol sistemleri olarak ikiye ayrılır.

2.3.1. Açık Çevrim Kontrol Sistemleri

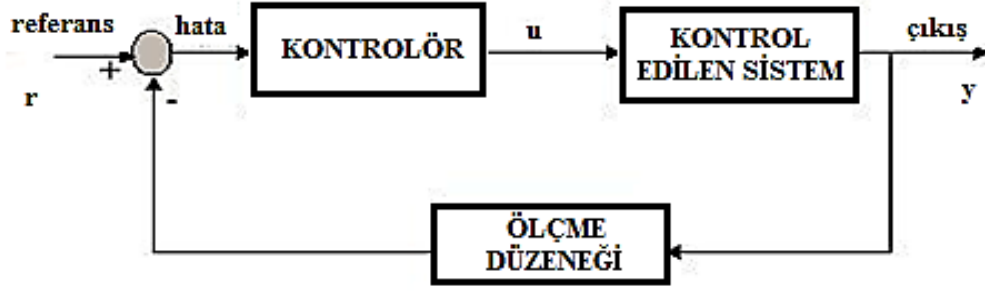
Sistemi kontrol eden mekanizmanın sistemin çıkışından bağımsız olduğu hassasiyet gerektirmeyen sistemlerde kullanılan sistemlerdir. Kontrol işlemi verilen referans değerine göre yapılır. Dışarıdan sisteme bir bozucu etki olduğunda bu etkinin algılanması insan faktörü ile olabilmektedir. Şekil 2.9’da blok diyagramı verilen açık çevrim kontrol sisteminde bir ısıtıcı ile bir odanın sıcaklığını istenen seviyede ayarlanmak istenmektedir fakat bir pencerenin açılması ile sisteme bozucu bir etki geldiğinde sıcaklık değeri istenildiği gibi kontrol edilemeyecektir [15].



Şekil 2.9. Açık çevrim kontrol sisteminin blok diyagramı [8]

2.3.2. Kapalı Çevrim Kontrol Sistemleri

Açık çevrim kontrol sisteminden farklı olarak sistemde bir geri besleme etkisi olup kapalı çevrim denetim sistemine bu sebepten geri beslemeli kontrol sistemi de denmektedir. Şekil 2.10’da gösterildiği gibi ölçme düzeneği kullanılarak bir çıkış değeri elde edilir. Elde edilen çıkış değerinin büyüklüğü girişe geri besleme sayesinde tekrar gönderilir. Bu değer referans değeriyle karşılaştırılır ve bunun sonucunda bir hata sinyali elde edilir. Elde edilen hata sinyalinin yapısına ve çıkış değerine göre uygun bir denetim sinyali üretilir [15]. Örnek olarak bir trafik sinyalizasyon sistemini ele alabiliriz. Sensörler yardımı ile yoldaki trafik yoğunluğun sürekli olarak ölçüldüğü bir sistemde elde edilen veriler doğrultusunda trafiğin yoğun olduğu tarafta yeşil ışık uygulaması otomatik olarak yapılarak trafik yoğunluğu önlenir. Bu sistemde araçların durumu çıkış büyüklüğünü, sinyalizasyonun zamanlaması da girişi temsil etmektedir. İyi bir çıkış için yolun boş ve dolu tarafına göre zamanlama ayarlanabilmelidir.



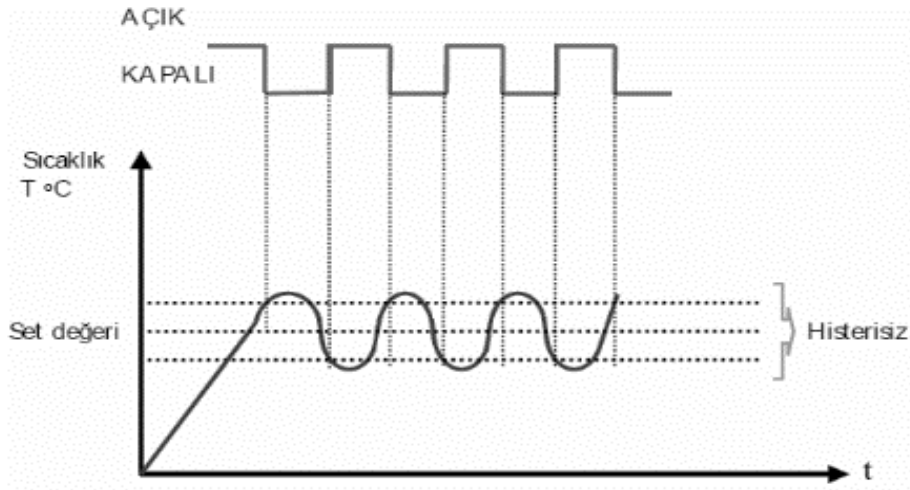
Şekil 2.10. Kapalı çevrim kontrol sisteminin blok diyagramı

2.4. Sıcaklık Kontrol Yöntemleri

Sıcaklık kontrol uygulamalarında sıklıkla kapalı çevrim modeli ile on-off, P, PD, PI, PID metotları kullanılmaktadır. Bu metotlar kontrol hassasiyetleri ve tasarım kolaylıkları bakımından birbirlerine karşı avantaj ve dezavantajlara sahiptirler.

2.4.1. Aç-kapa (On-Off) Kontrol

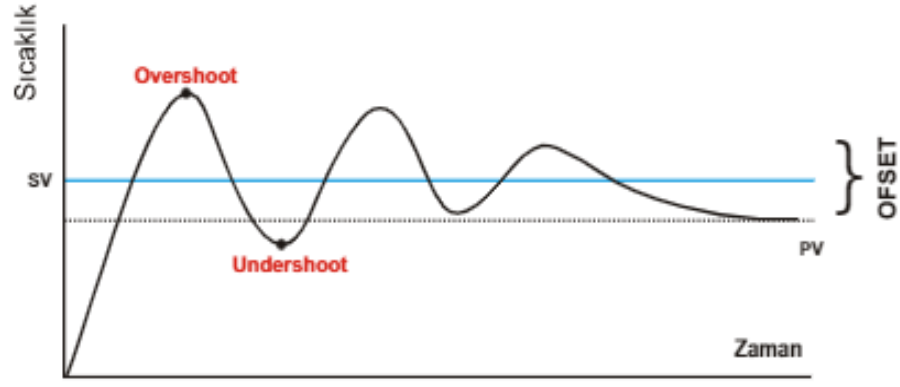
En basit kontrol tekniği olan on-off kontrol yönteminin iki konumu vardır. Sistem ya tamamen açık konumdadır ya da tamamen kapalı konumdadır. Ölçülen sıcaklık değeri istenilen sıcaklık değeri ile aynı değerde olana kadar çıkış sinyali açık (on) konumunda olup referans değerinin üstüne çıkması ile çıkış sinyali kapalı (off) konuma geçer. Fakat kontrol gücü kesildiği halde ölçülen değer set değerine ulaşmış orada kalmak yerine bu set edilen değer etrafında salınım yapacaktır. Sıcaklık belirlenen değerin üstüne çıktığı zaman ısıtıcı çalışmaz, üstüne çıkmazsa çalışır mantığından dolayı sistem sürekli aç-kapa yaptığından dolayı bir süre sonra sık sık mekanik arızalar verebilmektedir. Bu arıza durumlarının etkisini en az seviyeye indirmek için ölü band (histerezis) adı verilen bir referans değer aralığı kullanılmaktadır. Şekil 2.11'deki histerisizli aç-kapa kontrol eğrisinde ölçülen değer istenilen değeri aştığı anda değil referans değerin üstünde önceden ayarlanan bir sıcaklık değerini aştığı anda sistem kapalı (off) değere geçer. Bu yöntem sayesinde aç-kapa değişimleri için bir geciktirme işlemi yaparak mekanik sistemin çalışma ve durma sıklığı azaltılmış olur. Sonuç olarak çok sağlıklı bir kontrol yöntemi değildir. Sistem aşırı enerji tüketimine neden olur ve istenen set değerine sabitlenemez, sürekli salınım yapar. Kritik ve hızlı proseslerde tümüyle yetersizdir [16].



Şekil 2.11. Histerisizli aç-kapa kontrol eğrisi [7]

2.4.2. Oransal (P) Kontrol

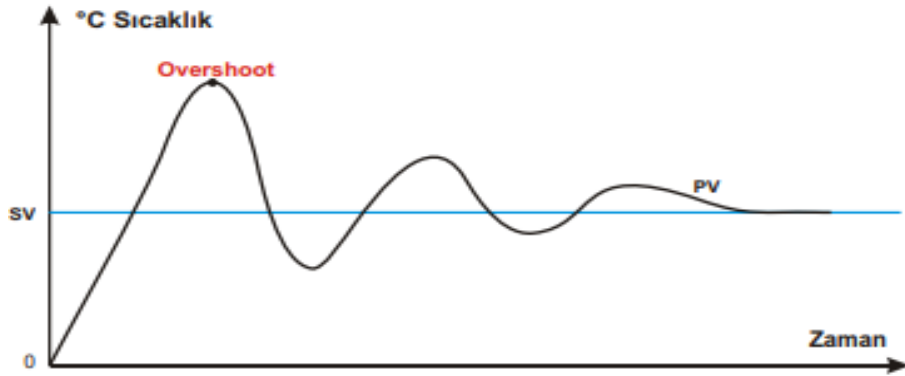
Oransal kontrol işlemi, değişen sıcaklık değerinin set değeri ile arasında oluşan sapmayı kontrol eden işlemdir. Oransal kontrol işleminde kullanılan cihaz, çıkıştan alınacak sinyali ihtiyaç olan enerji doğrultusunda otomatik ayarlar. Bu kontrol sisteminde ihtiyaç duyulan enerji ile sisteme yollanan enerji arasında bir oran vardır. Şekil 2.12’de gösterilen oransal kontrol eğrisinde gerçek değer ile set değeri arasında ölü bantı (offset) adı verilen bir fark değeri oluşmakta olup bu sapma değerini (offset) azaltmak amacıyla oransal band değeri küçültülebilir. Fakat oransal band değerini küçülttükçe sistem on-off kontrol işlemine yaklaştığı için set edilen değerin etrafında salınım miktarı artabilir. Geniş oransal bant işleminde ise ofset değeri büyük olacağından dolayı prosese en uygun oransal bandın seçilmesi gerekir. Örneğin bir oransal kontrol cihazı için ölçüm aralığı 1000°C , oransal bant % 5 ve set değeri 300°C olarak ayarlanmış olsun. Oransal bandın sıcaklık aralığı $0.05 \times 1000 = 50^{\circ}\text{C}$ olarak hesaplanır ve bu değerin 25°C ’si set edilen değerin üstünde 25°C ‘si ise set edilen değerin altında olacak şekilde kontrol işlemi yapar. Bandın uç noktaları 275°C ve 325°C olur. Kontrol cihazı sistemin 275°C ’ye ulaşana kadar ısıtıcılara enerjinin %100’ünü verir, bir başka deyişle enerji tamamen açık konumdadır. 275°C ’ den itibaren 325°C olan set değerine kadar sıcaklık artarken ısıtıcıya verilen enerji yavaş yavaş oransal olarak kısılır. Eğer sıcaklık değeri set edilen değerin üstüne çıkıp yükselmeye devam ederse 325°C ’ye kadar enerji giderek kısılır ve 325°C ’nin üzerine çıktığı zaman sistem enerjiyi tamamen kapatır. Bu işlemler sıcaklık değeri düşerken ise tam tersi olarak gerçekleşir [17].



Şekil 2.12. Oransal Kontrol Eğrisi [7]

2.4.3. Oransal İntegral (PI) Kontrol

Oransal kontrol işleminde gerçekleşen sapma (offset) durumunu azaltmak veya tamamen ortadan kaldırmak için sisteme integral alıcı devre eklenir. Bu devre ölçüm sonucu elde edilen sıcaklık değeri ile ayarlanan (set değeri) sıcaklık değeri arasında oluşan bir fark sinyalinin zamana göre integralini alır. Oluşan bu integral değeri fark değeri ile toplanır ve Şekil 2.13'te gösterildiği gibi oransal kontrol uygulamasında oluşan sürekli hal hatası (offset etkisi) integral sabitinin etkisiyle belirli bir sürede sıfırlanır [18].

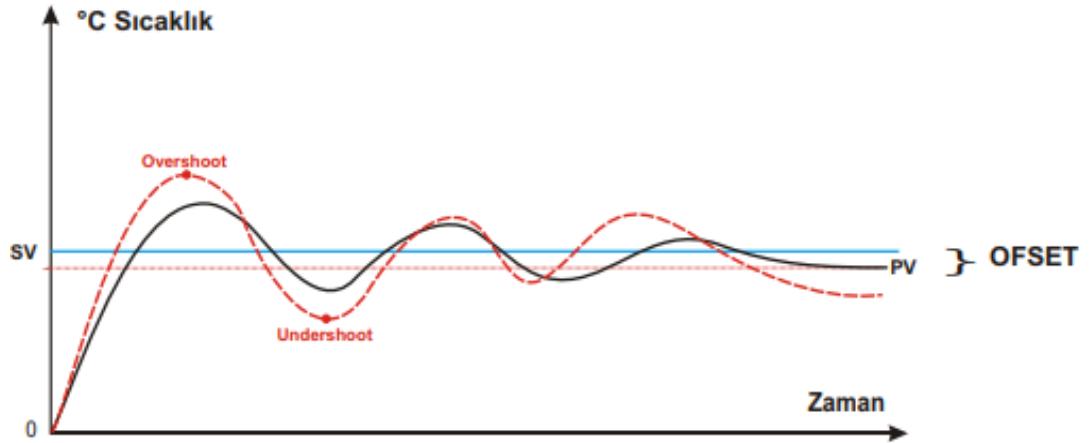


Şekil 2.13. Oransal integral kontrol eğrisi [18]

2.4.4. Oransal Türevsel (PD) Kontrol

Türevsel etkinin asıl amacı yükselme (overshoot) ve alçalma (undershoot) miktarlarını azaltmakla birlikte offset değerini de azaltabilir. Türevsel devre ölçüm sonucu elde edilen değer ile set edilen değer arasındaki fark sinyalinin türevini alır ve oransal devreden geçen fark sinyali ile toplanır. Şekil 2.14'te gösterildiği gibi türevsel etki ile bir düzeltme işlemi yapılmış olur. Türevsel etkinin en büyük özelliği hatayı

önceden sezmesi ve hata büyümeden hızlı bir şekilde müdahale edebilmesidir. Bu nedenle değişimlerin hızlı olduğu kısa süreli proses işlemlerinde kullanılması uygundur. Sürekli tip, uzun süren proses işlemlerinde ve sapma işleminin istenmediği durumda ise sisteme uygun olan PI ve PID kontrol yöntemi seçilmelidir [18].



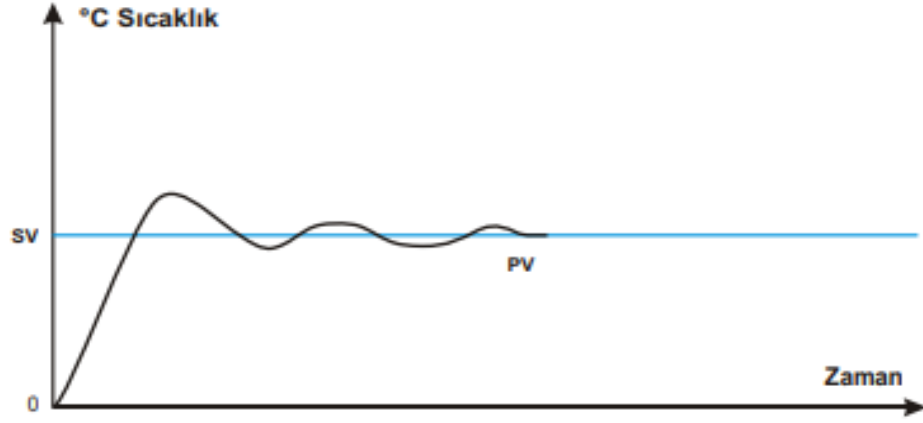
Şekil 2.14. Oransal türevsel kontrol eğrisi [18]

2.4.5. Oransal İntegral Türevsel (PID) Kontrol

Kontrol sistemlerinde en çok uygulanan kontrol metotlarından biri PID kontrol yöntemidir. Hassasiyetin yüksek ve kontrol işlemlerinin zor olduğu süreçlerin oransal (P), integral (I) ve türevsel (D) kontrol gerektiren cihazlarla kontrol edilmesi gerekmektedir [19].

PID denetleyici şu anda var olan hata miktarına bağlı (P) oransal kazanç, hata genliğini tahmin edebilen (I) integral alıcı devre ve geçmiş hata genliğinin toplamını oluşturan (D) türevsel devre parametrelerinden oluşur [20]. Oransal (P) tip kontrol cihazları ile kurulan sistemlerde set değeri ile kontrol edilen değer arasında ofset değeri tam olarak sıfıra indirilememektedir. Sadece oransal (P) olarak kontrol işlemi yapılan bir sistemde integral devresinin (I) ilave edilmesi offset değerinin yok edilmesine yönelik olup sistem dengeye geldikten sonra offset durumunun oluşması beklenmez. Bununla birlikte integral zamanının çok kısa olması sistemde osilasyon oluşturmaya neden olabilir. PI kontrol sistemi mekanizmasına türevsel devrenin (D) eklenmesi ise sistemin set edilen değere ulaşması için geçen zamanı kısaltma amacına yöneliktir [21]. Sonuç olarak PID kontrolörler yukarıda belirtilen bu üç kontrol yönteminin üstünlüklerini tek bir birimde birleştirir [18]. PID kontrolörü oluşturan kısımlar birer katsayı ile yönetilirler. Bu katsayılar oransal kazanç (P) için K_p , integral kazanç (I) için K_i ve Türevsel

kazanç (D) için K_d 'dir. Kontrol edilecek sistemi optimum performansla kontrol edebilecek katsayı değerlerinin ayarlanması ile sistemin başarılı bir kontrolü sağlanmış olur [22].



Şekil 2.15. Oransal integral türevsel kontrol eğrisi [18]

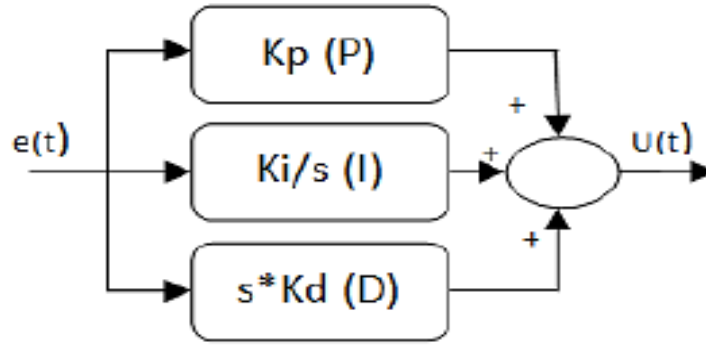
PID katsayılarının sistem performansına olan etkileri Çizelge 2.1'de verilen katsayıların etkisinin grafiksel olarak gösterimi Şekil 2.16'de verilmiştir.

Çizelge2.1. PID katsayılarının etkisi [23]

<i>Artan parametre</i>	<i>Yükselme zamanı</i>	<i>Aşma</i>	<i>Oturma zamanı</i>	<i>Kararlı Durum Hatası(offset)</i>
K_p	Azalı	Artar	Etkilenmez	Azalı
K_i	Azalı	Artar	Artar	Yok edilir
K_d	Etkilenmez	Azalı	Azalı	Etkilenmez



Şekil 2.16. PID katsayılarının etkisinin grafiksel gösterimi



Şekil 2.17. PID kontrolörün blok diyagramı

Şekil 2.17’de blok diyagramı verilen PID kontrolörün çıkışı $u(t)$ denklem (3.1) ile ifade edilir.

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt} \quad (3.1)$$

PID kontrolörün transfer fonksiyonu $u(s)/e(s)$ denklem (3.2)’de gösterilmiştir. PID kontrolörün integral ve türev zaman sabitleri biçiminde gösterimi ise denklem (3.3)’te verilmiştir.

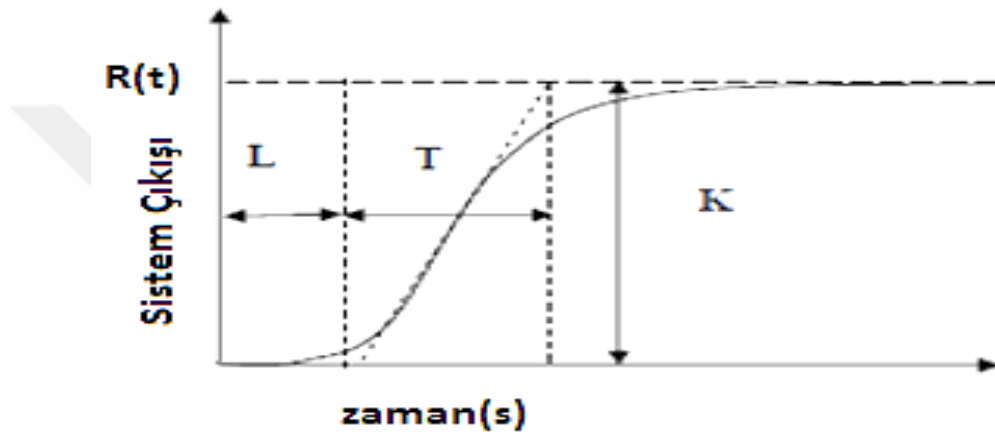
$$\frac{U(s)}{E(s)} = K_p + \frac{K_i}{s} + K_d s \quad (3.2)$$

$$\frac{U(s)}{E(s)} = K_p \left[1 + \frac{1}{T_i s} + T_d s \right] \quad (3.3)$$

K_i ve K_d ’nin zaman sabitleri cinsinden formülleri ise denklem (3.4) kullanılarak hesaplanabilir.

$$K_i = K_p / T_i, \quad K_d = K_p T_d \quad (3.4)$$

Denklem (3.2)'de gösterilen PID parametrelerinin uygun değerlerde seçilmesi, başarılı bir PID kontrol için önemlidir. Fakat bu parametrelerin seçimleri biraz zorluklar içerir ve tek bir parametrenin değerini değiştirmek ile başarılı bir sonuç alınamamaktadır. Bu parametrelerin uygun şekilde elde edilebilmesi amacıyla ortaya konulan Ziegler-Nichols metodu ile bu değerler yaklaşık olarak tespit edilebilmektedir [8, 23, 24]. PID katsayılarını bulabilmek için kontrol etmek istediğimiz ısıtıcının transfer fonksiyonunu Şekil 2.18'de gösterilen açık-çevrim basamak cevabı üzerinden (3.5) numaralı denklem biçiminde modeli oluşturulur.



Şekil 2.18. Açık çevrim kontrol sisteminin ideal basamak cevabı [22]

$$G(s) = \frac{Ke^{-sL}}{(1+sT)} \quad (3.5)$$

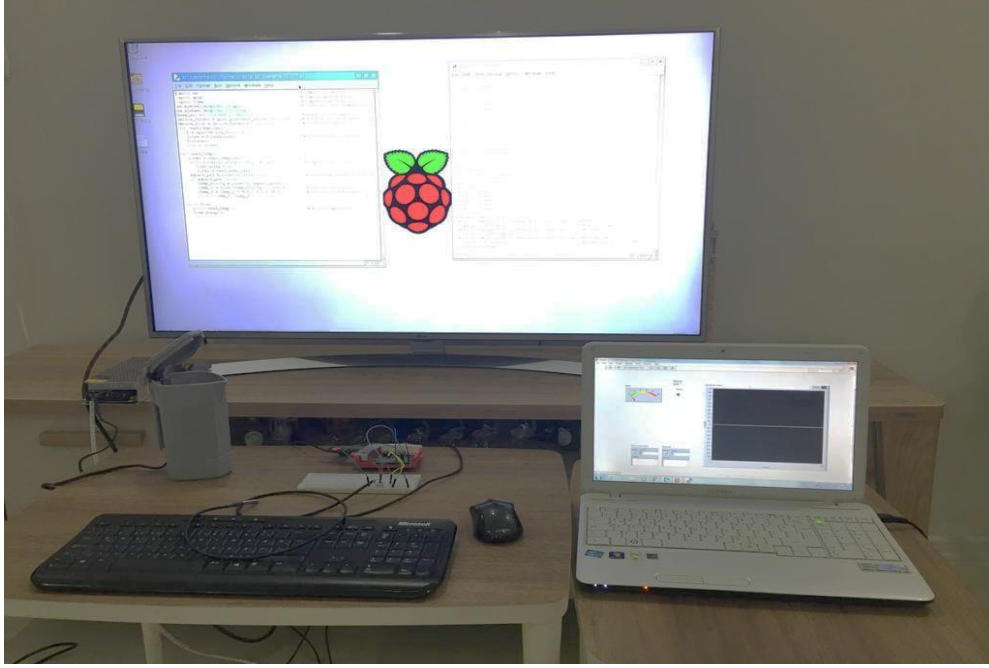
Açık çevrim birim basamak cevabı üzerindeki K , L ve T sayısal verileri kullanılarak Ziegler-Nichols metodu yaklaşımı ile K_p , T_i ve T_d parametreleri Çizelge 2.2'ye göre hesaplanır [25].

Çizelge 2.2. Ziegler- Nichols PID parametre tablosu [25]

<i>Kontrolör</i>	K_p	T_i	T_d
P	$\frac{T}{KL}$	∞	0
PI	$0.9 \frac{T}{KL}$	$3.3L$	0
PID	$1.2 \frac{T}{KL}$	$2L$	$0.5L$

3. GERÇEKLEŐTİRLEN KONTROL SİSTEMİNİN DONANIMI

Bu bölümde gerçekleştirilen sistemde kullanılan devre elemanlarından ve birbirleriyle olan etkileşimlerinden bahsedilecektir. Devrenin tüm donanım elemanları Şekil 3.1’de gösterildiğı gibi bağlanmıştır.

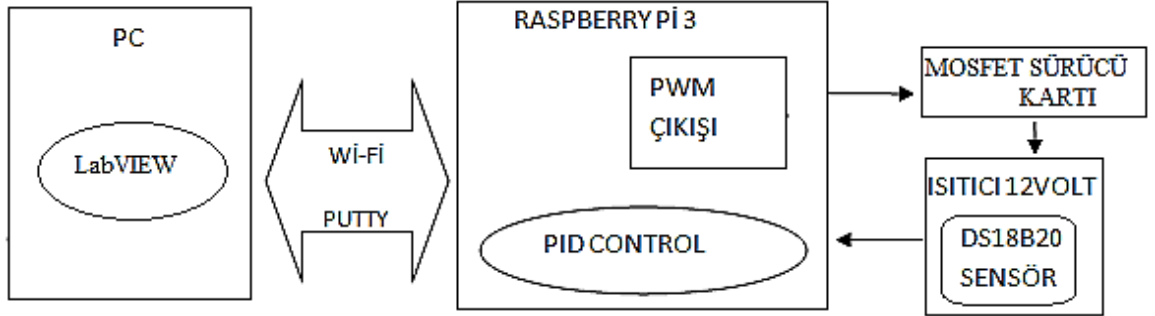


Şekil 3.1. Sistemin Genel Görünümü

3.1. Giriş

Gerçekleştirilen uygulamada bir tank içindeki suyun sıcaklığını kontrol eden uygulamanın tasarımında kullanılan devre elemanlarının blok diyagramı Şekil 3.2’de gösterilmiştir. Sistemde son zamanlarda en çok kullanılan mini bir bilgisayar olan Raspberry Pi 3 kullanılmıştır. Sıcaklık ölçümü için Raspberry Pi 3 üzerinden kodlanan yazılım ile DS18B20 dijital sıcaklık sensörü kullanılarak yapılmıştır. Elde edilen sıcaklık verilerinin uzaktan takip edilebilmesi için LabVIEW programlama dili kullanılarak bir kullanıcı ara yüzü geliştirilmiştir. Raspberry Pi 3’ün Wi-Fi ile kablosuz haberleşme özelliğı kullanarak sıcaklık verilerinin kablosuz olarak izlenebilmesi Raspberry Pi 3’ün sisteme sağladığı en büyük avantajlarından biridir. Ayrıca Raspberry Pi 3’ün mini bilgisayar mimarisine sahip olması birçok kontrol işleminin aynı anda sürdürülebilmesine olanak sağlamaktadır. Raspberry Pi 3 teknolojisi ve LabVIEW programının birlikte kullanılması ile sistemin takibinde ve kontrolünde kolaylıklar

sağlamaktadır. LabVIEW'in grafiksel kullanıcı ara yüzü özelliği sunması geliştirilen uygulamalar için kullanım kolaylığı sağlamaktadır.

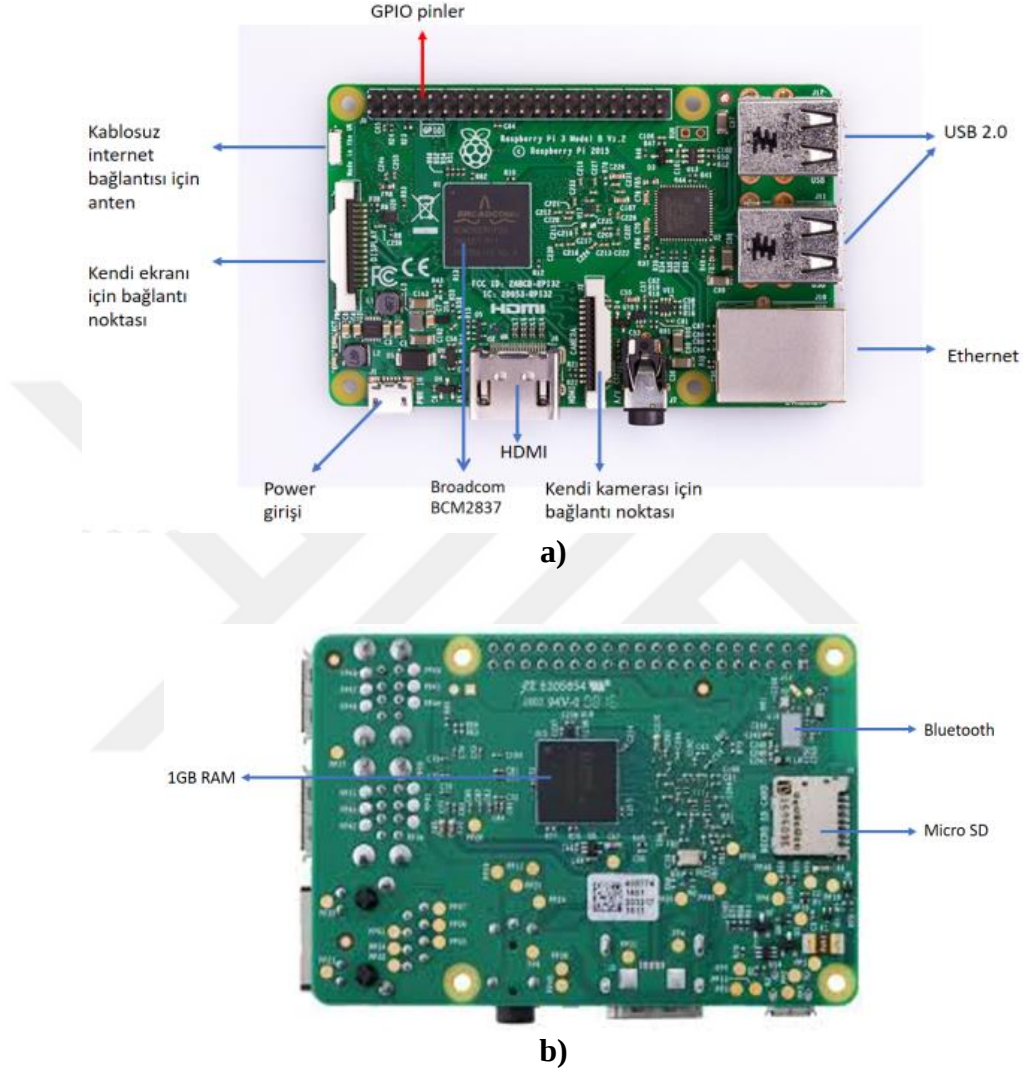


Şekil 3.2. Gerçekleştirilen kontrol uygulamasının blok diyagramı

3.2. Raspberry Pi 3

Raspberry Pi boyut olarak kredi kartı büyüklüğünde olup eğitim, hobi ve birçok amaç için tasarlanmış, maliyeti düşük ve genellikle Linux işletim sistemi ile kullanılan The Raspberry Foundaiton (Raspberry Pi Vakfı) tarafından üretilmiş mini bir bilgisayardır [26]. İlk olarak 2012 yılında satışa sunulan Raspberry mini bilgisayar, düşük maliyeti, taşınabilirliği ve desteklediği özellikler ile yüksek satış rakamlarına ulaşmıştır. Raspberry Pi bir bilgisayar olmasının yanında aynı zamanda fiziksel dünyayı algılayan ve kontrol edebilmek için kullanabileceğimiz basit bir mikroişlemcidir [27]. Raspberry Pi'nin üzerindeki 40 pin sayesinde birçok kontrol uygulaması aynı anda gerçekleştirilebilir. Bluetooth ve wifi özelliği ile haberleşmesi kolay, hızlı ve etkilidir. Dokunmatik ekran ve kamera modülü portları sayesinde kodlaması ve kontrol edilecek sistemlere müdahalesi rahatlıkla yapılabilir [28]. Raspberry Pi ile sıcaklık ve nem kontrolü, motor kontrolü, ev otomasyonu gibi birçok kontrol uygulaması gerçekleştirilebilir. Raspberry Pi kartı ilk üretildiği yıldan itibaren sürekli olarak geliştirilmiş ve yeni çıkan modeline hep bir önceki modele ek olarak yeni donanımlar ve özellikler eklemiştir. RPI kartlarının bugüne kadar çıkarttığı birçok model olup bu modeller Raspberry Pi Zero, Raspberry Pi Zero W, Raspberry Pi 1 Model A+, Raspberry Pi 1 Model B+, Raspberry Pi 2 Model B, Raspberry Pi 3 Model B'dir.

En son çıkarmış olduğu ve tez çalışmasında kullanılan Raspberry Pi 3 Model B+ kartıdır. RPI 3'ün yapısı Şekil 3.3'te gösterilmiştir [29].



Şekil 3.3. a) RPI 3 ön yüzü b) RPI 3 arka yüzü [30]

Raspberry ARM tabanlı bir işlemci (CPU) kullanır. Raspberry Pi 3 kartında Broadcom firmasının üretmiş olduğu BCM2837, 1.2 GHz 64-bit 4 çekirdekli ARM Cortex-A53 işlemciye sahip olup 1200MHZ hızında yüksek işlem gücüne sahiptir. Raspberry Pi 3 üzerinde herhangi bir depolama alanı yoktur bunun yerine SD kart yuvası olup, işletim sisteminin kurulumu bu SD karta indirilerek yapılmaktadır. RPi 3 üzerinde dört tane usb portu vardır. Bu sayede klavye, mouse, kamera, Tv ve daha birçok şey bağlayabilme olanağına sahiptir. Raspberry Pi verimli olarak çalışabilmesi gerekli olan

gücünü 5 volt 2.1 amper gücünde olan adaptörden almaktadır. Raspberry'nin aç-kapa tuşu olmayıp adaptör takıldığı zaman direk çalışmaktadır.

RPI 3'ün bağlantı portları çok sayıda olup bu portlar 40 adet GPIO pinleri, kamera ve ekran portu, 4 tane USB 2.0 portu, 100 Mbit hızında olan ethernet portu, HDMI çıkışları ve güç bağlantısı için mikro USB çıkışı bulunur. Ayrıca RPI 3 modeline eklenen Wi-Fi ve Bluetooth özelliğiyle birlikte kablosuz olarak cihazlarla haberleşebilir, internete bağlanıp dilediğiniz gibi gezinebilirsiniz. Ayrıca HD videolar oynatabilirsiniz [31].

Raspberry Pi'nin birçok avantajları bulunmaktadır. Bu avantajlar:

- Kendine ait bir işletim sistemi bulunması
- Düşük maliyetli olması
- Arttırılabilir hafızanın olması
- USB çıkışının olması
- HDMI ve VGA çıkışının olması
- İşlemcisinin olması
- Başka harici elektronik kartlar ile birlikte çalışabilmeyi desteklemesi
- Taşınabilmesinin kolay olması
- Kablosuz bağlantı yoluna sahip olması (Bluetooth, Wi-fi)
- Açık kaynaklı yazılımları desteklemesi
- Dokunmatik ekran bağlanabilmesi
- Kurulumun ve kullanımının basit olması
- Düşük voltajla çalışabilmesi
- Elektronik çalışmalar için çıkışlarının olması
- Mouse, klavye bağlanması
- Boyutunun küçük olması gibi birçok özellik sayılabilir

Çok avantajının olmasının yanı sıra dezavantajlı durumları da bulunmaktadır. Bunlar:

- Raspberry Pi’de fan mevcut değildir ve ağır programlarda ısınma problemi vardır.
- RPI’nin ısınmasını önlemek için ayrıca bir soğutucu fan kullanılması gerekmektedir.
- Ağır programlarda yavaş çalışması
- Alternatiflerinin çok sayıda olması
- Elektronik çalışmalar için geliştirilmiş daha iyi özellikli cihazların bulunması
- Gücün kesilmesi durumunda RPI’nin kapanması ve baştan başlaması sayılabilir.

3.2.1 Raspberry Pi 3 Dilleri

RPI ile kullanılan programlama dilleri çok sayıda olup bir bölümü Şekil 3.4’teki gibi gösterilmiştir.

- ✓ Python
- ✓ C, C++, C#
- ✓ Perl
- ✓ Java vs.
- ✓ PHP



Şekil 3.4. Python Dilleri

Python, Hollandalı bir Bilgisayar Programcısı olan Guido Van Rossum tarafından 1991 yılında tasarlanmış olup, nesneye yönelik, yorumlanabilir, etkileşimli yüksek seviyeli bir programlama dilidir [32]. Python script dilinin popülerliği 2000 yılından

itibaren giderek artmaya başlamıştır. Son zamanlarda ise Google, Autodesk vb. birçok üründe kullanımı artarak önemli rol oynamaktadır. 2007’de bir yıllık süre zarfında %2,4 artış gösteren Python programlama dili, 2008 yılının ocak ayında TIOBE tarafından 2007’nin programlama dili olarak ilan edilmiştir [33, 34]. Bunun birçok nedenleri vardır.

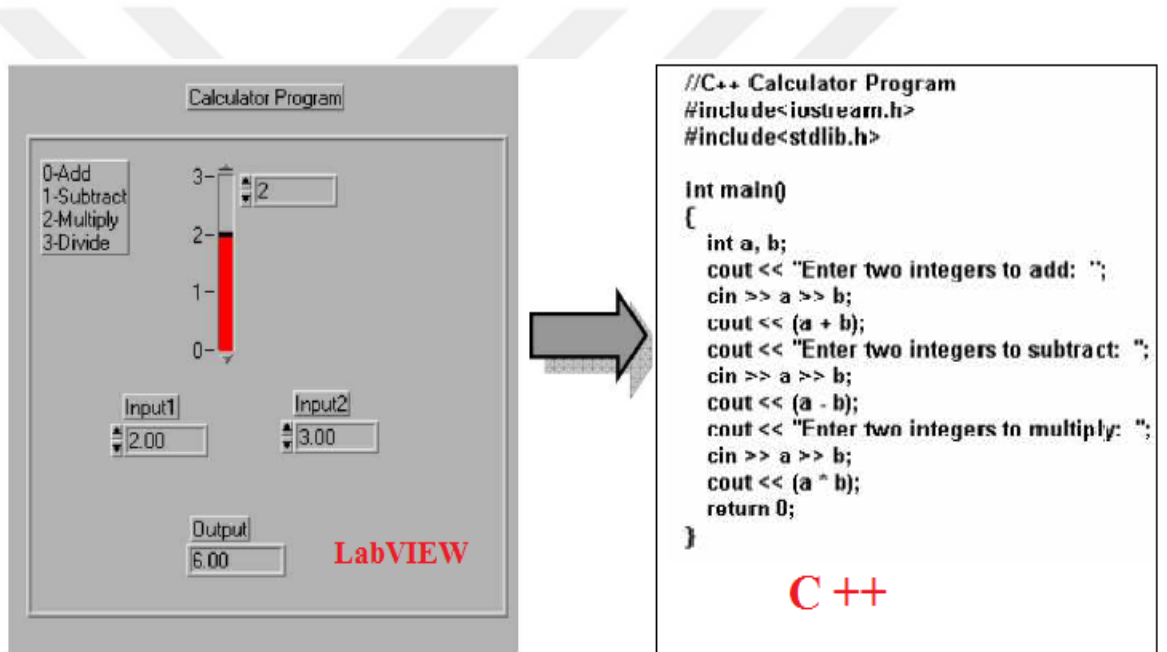
Python dilinin tercih edilme sebepleri;

- **Açık kaynaklıdır.** Öğretici kaynak bulmak kolay olup herhangi bir firmaya bağlı olmadan özgürce çalışılabilmektedir. Ayrıca sadece kişisel kullanımlarda değil aynı zamanda ticari uygulamalarda da ücretsizdir [35].
- **Yapısı sadedir.** Okuması ve yazması kolay olup kullanıcıların hızlı bir şekilde kavramaları için özlü ve basit bir sözdizimine sahiptir bu da Python’ un diğer programlama dillerine kıyasla daha yüksek bir geliştirme verimine sahip olmasını sağlamaktadır [33].
- **Dinamik bir dildir.** Yorumlayıcıyla çalışmakta olup çok karmaşık işlemler basit komutlarla yapılabilmektedir.
- **Etkileşimli olan bir dildir.** Yorumlama penceresinde alt alta komutlar verilerek işlemler yapılabilir.
- **Genel kullanım alanı geniştir.** Birçok işletim sistemi üzerinde çalışmakta olup yazılım sektöründe çok tercih edilmektedir. Bu sebepten dolayı sürekli geliştirilmektedir [34].
- **Bilimsel araştırmalarda çok yaygın olarak kullanılmaktadır.** Çeşitli bilimsel alanlarda kullanılabilir özel olarak hazırlanan kütüphaneler mevcuttur. Bu kütüphanelerin hazırlanması profesyonel yazılımcılar tarafından olup sürekli olarak test edilmektedir [36].

3.3. LabVIEW

LabVIEW, National Instruments tarafından geliştirilen bir görsel programlama dili olup LabVIEW programı “Laboratuary Virtual Instruments For Engineering Workbench” cümlesinin kısaltılmasıdır. Grafıksel ara yüz oluşturmada, veri elde etmede kullanılan LabVIEW ölçme, enstrümantasyon odaklı olarak geliştirilmiş olup bir yazılım üretme platformu olarak birçok yazılımın özelliklerini tek bir pakette birleştirmiştir. Diğer yapısal ve nesne tabanlı programlama dillerinin yapabildiği neredeyse tüm işlemleri yapabilen her yeteneğe sahiptir.

LabVIEW, C tabanlı olarak ölçme amaçlı geliştirilmiş ve C dilinin esnekliğini ve kapsamlı işlevselliğini sunmasına rağmen C tabanlı programlara göre daha fazla verimlilik de sunar. LabVIEW programlamada yeni bir dönem başlatarak grafiksel programlama dili (GPL) teknolojisini sunmuştur. Kullanılan çoğu programlama dilleri kod satırlarını oluşturmak için metin tabanlı bir dil kullanırken LabVIEW programlama dili sunulan bu (GPL) teknoloji ile birlikte görsel ve grafiksel bir programlama dili olan G'yi kullanmaktadır. G sayesinde kodlama tamamen sembolleştirilmiş olup komut ezberleme zorluğu ortadan kaldırılmıştır. Örneğin Şekil 3.5'te LabVIEW programıyla herhangi bir geleneksel programlama kodu karşılaştırılmıştır. C++ programlama dilinde kod satırları oluşturularak yapılan yazılım, Labview ara yüzünde görsel ve grafiksel olarak hızlı ve basit bir şekilde oluşturulabilmektedir [37].



Şekil 3.5. LabVIEW ve geleneksel programlama kodu karşılaştırması [37]

3.4. LabVIEW ve Raspberry Pi 3 Haberleşmesi

Raspberry Pi 3 diğer çoğu mikrodenetleyicilerden farklı olarak kablosuz olarak LabVIEW programı ile haberleşme özelliğine sahip olup Raspberry Pi 3'ün bu özelliği gerçekleştirilen uygulamaya büyük kolaylık sağlamaktadır. Bu bölümde LabVIEW ile Raspberry Pi 3'ün haberleşebilmesi için gereken adımlar anlatılacaktır [38].

Bu adımlar;

- 1- LabVIEW 2014 SP1 Kurulumu
- 2- VI Package Manager Kurulumu
- 3- QwaveSys RPI Package Kurulumu
- 4- MakerHub ve LINX Toolbox Kurulumu

3.4.1. LabVIEW 2014 SP1 Kurulumu

Şekil 3.6’da gösterilen LabVIEW 2014 32-bit, RPI için desteklenen tek LabVIEW sürümü olup diğer versiyonları ile haberleşme imkânı şu an için bulunmamaktadır [39].



Şekil 3.6. LabVIEW 2014

3.4.2. VI Package Manager Kurulumu

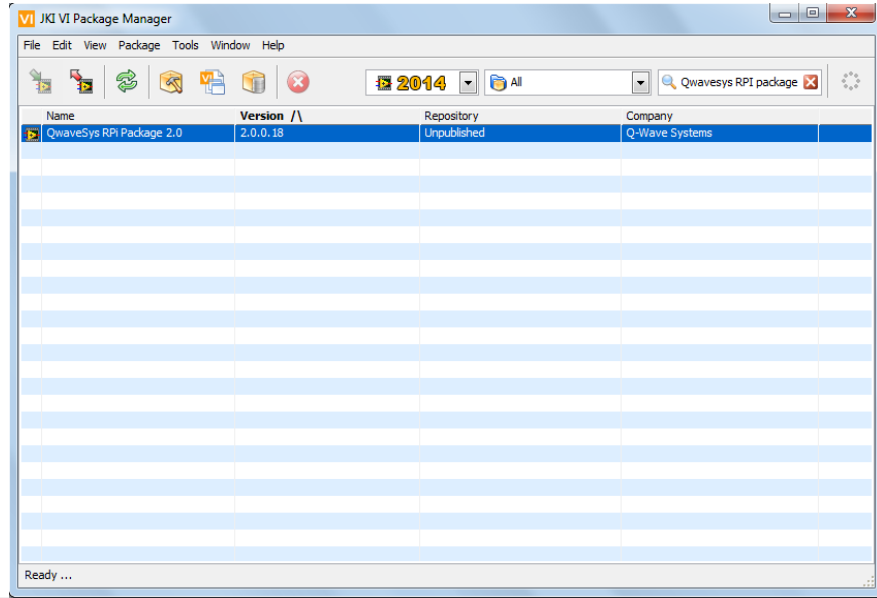
Şekil 3.7’deki VI Package Manager 2017 programı LabVIEW programlama dilinin kurulması ile birlikte gelen paketlerden biridir. Raspberry Pi’yi programlamak için VI Package Manager programını kullanarak gerekli olan fonksiyon menüsü yüklenecektir [40].



Şekil 3.7. VI Package Manager

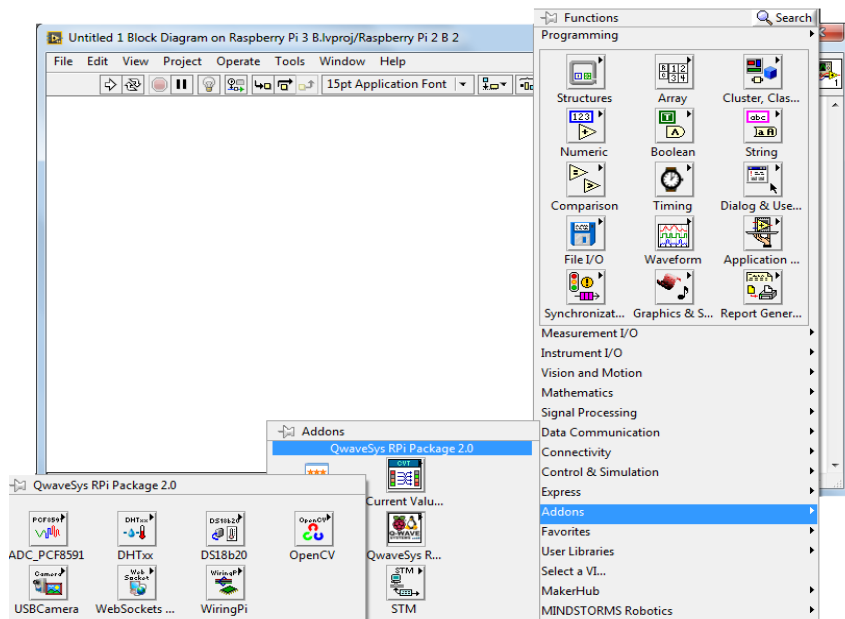
3.4.3.QwaveSys RPI Package Kurulumu

VI PM kurulumu tamamlandıktan sonra LabVIEW’de Raspberry Pi için gerekli olan QwaveSys RPI package eklentisi yüklenecektir. VI Package Manager programının arama kısmına Şekil 3.8’deki gibi “QwaveSys RPI package” yazılarak arama yapıldığında çıkan eklentiye sol üst köşeden yüklenir [41].



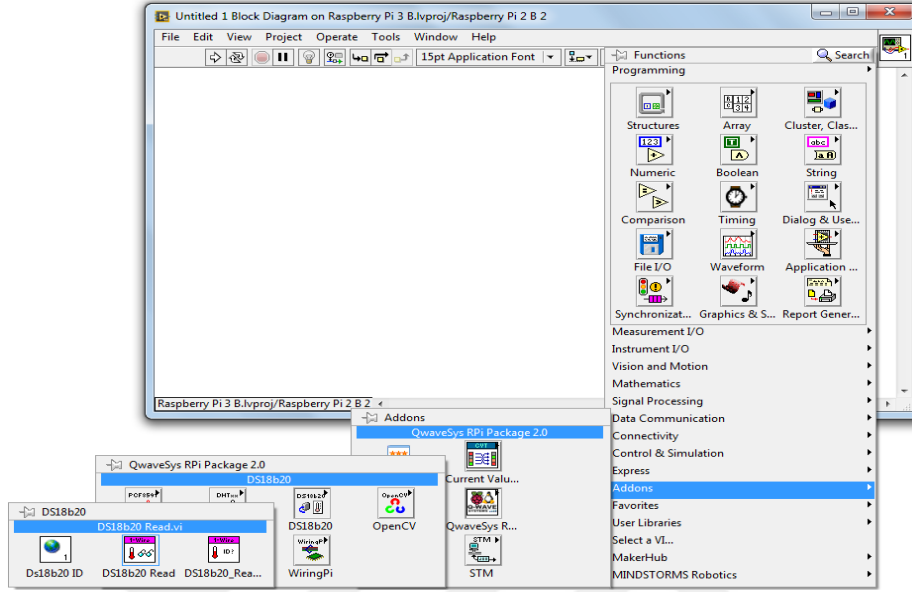
Şekil 3.8. Qwavesys RPI package kurulumu

Yükleme işlemi tamamlandığında Şekil 3.9’deki gibi QwaveSys RPI package eklentisi LabVIEW de addons kısmına eklenecektir.



Şekil 3.9. Qwavesys RPI package LabVIEW’e eklenmesi

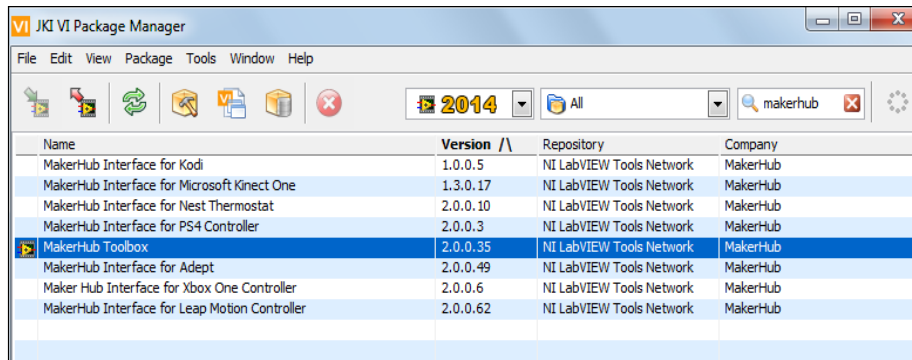
Şekil 3.10’da ise yüklenen eklentinin içinde DS18B20 sıcaklık sensörünün kütüphanesi hazır olarak bulunmaktadır.



Şekil 3.10. DS18B20 sensörün kütüphanesinin LabVIEW’e eklenmesi

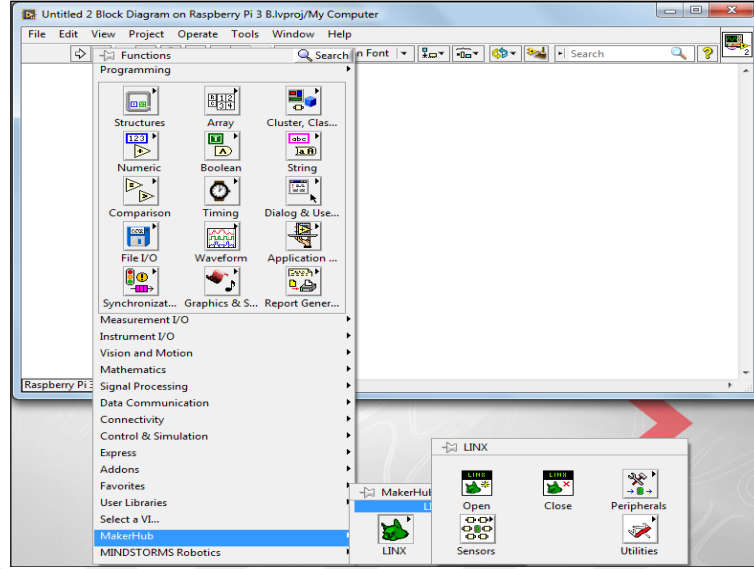
3.4.4. MakerHub ve LINX Toolbox Kurulumu

VI PM kurulumu tamamlandıktan sonra RPI ve LabVIEW haberleşmesi için kurulması gereken eklentiler yüklenecektir [38,42]. Açılan VI Package Manager programında arama kısmına Şekil 3.11’deki gösterdiği gibi ‘makerhub’ yazılarak çıkan sonuçlardan **MakerHub Toolbox** seçilir sol ve üst köşe bulunan paketi yükle seçeneği tıklanır.



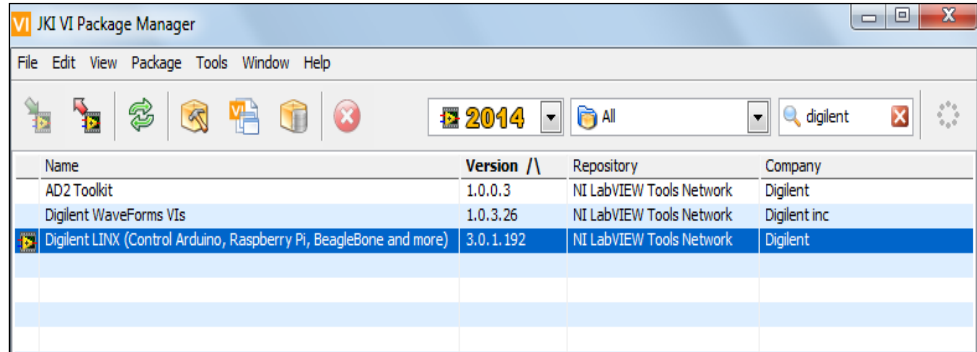
Şekil 3.11. VI PM’da Makerhub Toolbox’ı Yükleme

Bu yükleme işlemi başarı ile yüklenmişse MakerHub fonksiyonu Şekil 3.12’de gösterildiği gibi yüklenmiştir.



Şekil 3.12. Makerhub Toolbox’ın LabVIEW’e eklenmesi

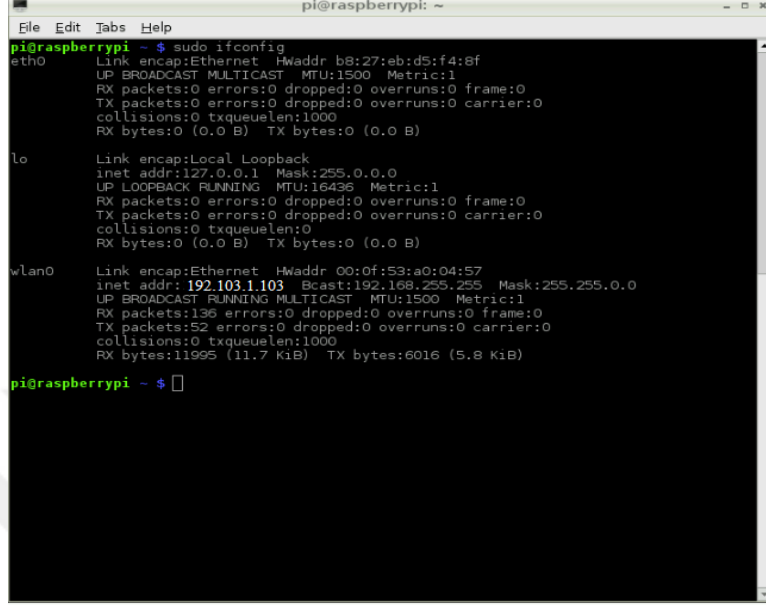
Daha sonraki adımda ise VI Package Manager programında arama kısmına Şekil 3.13’deki gösterildiği gibi ‘Digilent’ yazılarak çıkan sonuçlardan **Digilent LINX** seçilip sol üst köşeden paketi yükle seçeneğine tıklanır.



Şekil 3.13. VI PM’den Digilent LINX’i Yükleme

Tüm bu işlemlerin ardından LabVIEW ile Raspberry Pi’yi programlamak için yapılacak işlemlerden sonuncusu olan Raspberry Pi’yi tanıtırma işlemi gerçekleştirilecektir.

Öncelikle Raspberry Pi’yi açılır ve wifi’ye bağlantısı yapılır. Bilgisayardaki LabVIEW programı ile haberleşmesi için Raspberry Pi’nin terminaline Şekil 3.14’te gösterildiği gibi **if config** komutunu yazılır ve IP adresi öğrenilir.



```
pi@raspberrypi ~ $ sudo ifconfig
eth0      Link encap:Ethernet  HWaddr b8:27:eb:d5:f4:8f
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

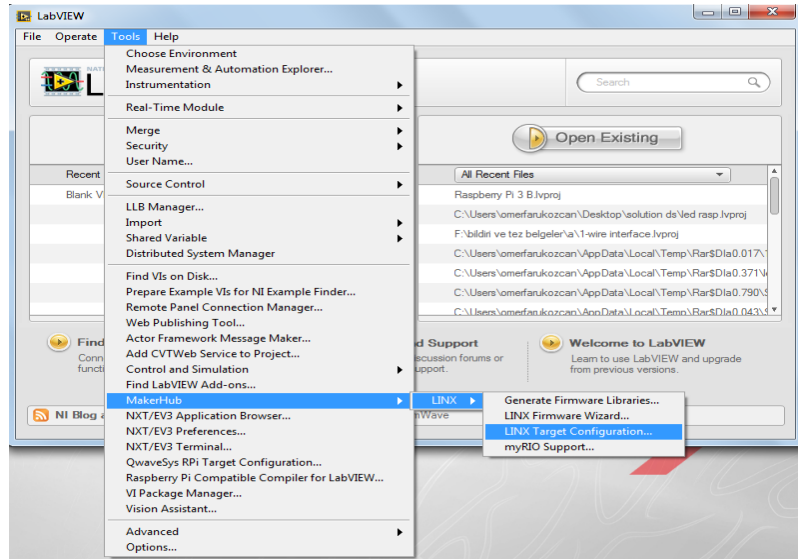
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

wlan0     Link encap:Ethernet  HWaddr 00:0f:53:a0:04:57
          inet addr:192.103.1.103  Bcast:192.168.255.255  Mask:255.255.0.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:186 errors:0 dropped:0 overruns:0 frame:0
          TX packets:52 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:11995 (11.7 KiB)  TX bytes:6016 (5.8 KiB)

pi@raspberrypi ~ $
```

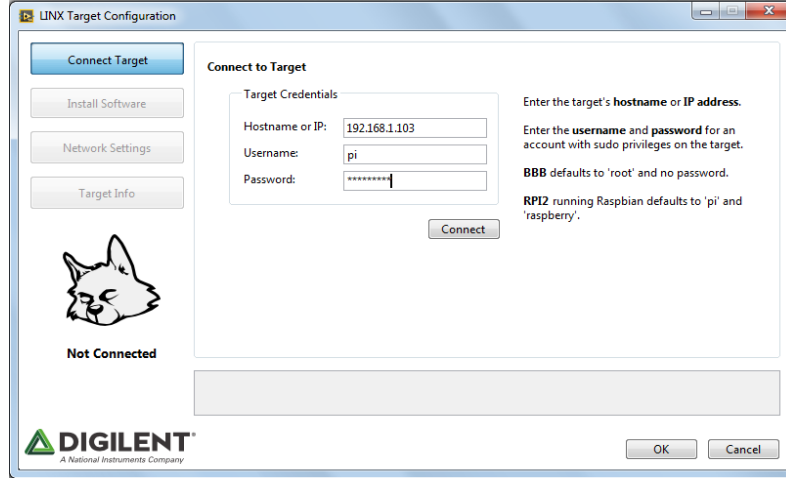
Şekil 3.14. RPI de IP adresi öğrenme

Daha sonra Şekil 3.15’te gösterildiği gibi **Tools > MakerHub > LINUX > LINUX Target Configuration** adımlarını takip ederek pencereyi açılır.



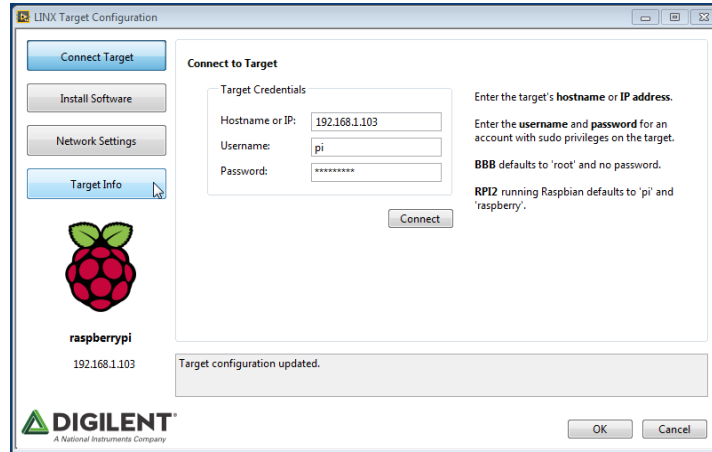
Şekil 3.15. LabVIEW’de RPI eşleştirme

Şekil 3.16'da açılan pencerede Raspberry Pi'nin IP adresini, username ve password bilgileri girilerek Connect target tuşuna basılır.

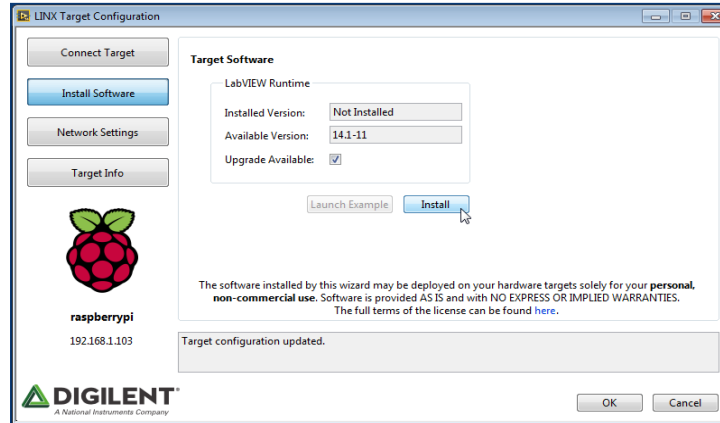


Şekil 3.16. RPI'nin IP, username ve password'unu girme

Şekil 3.17'de açılan pencerede Raspberry Pi'nin IP adresi eşleştirilir install software tıklanır ve hedeflenen RPI kurulumu başarıyla yüklenir.



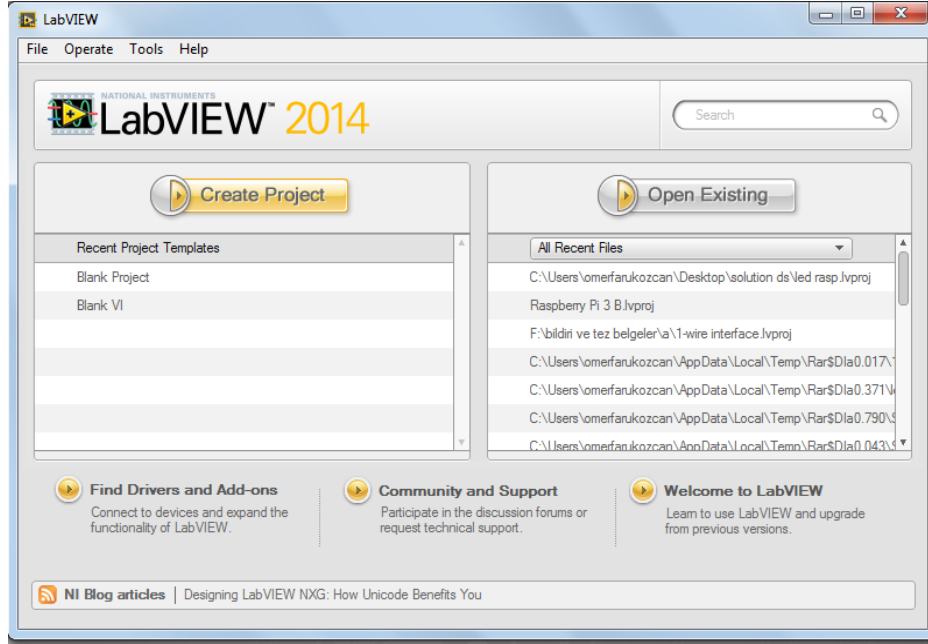
a)



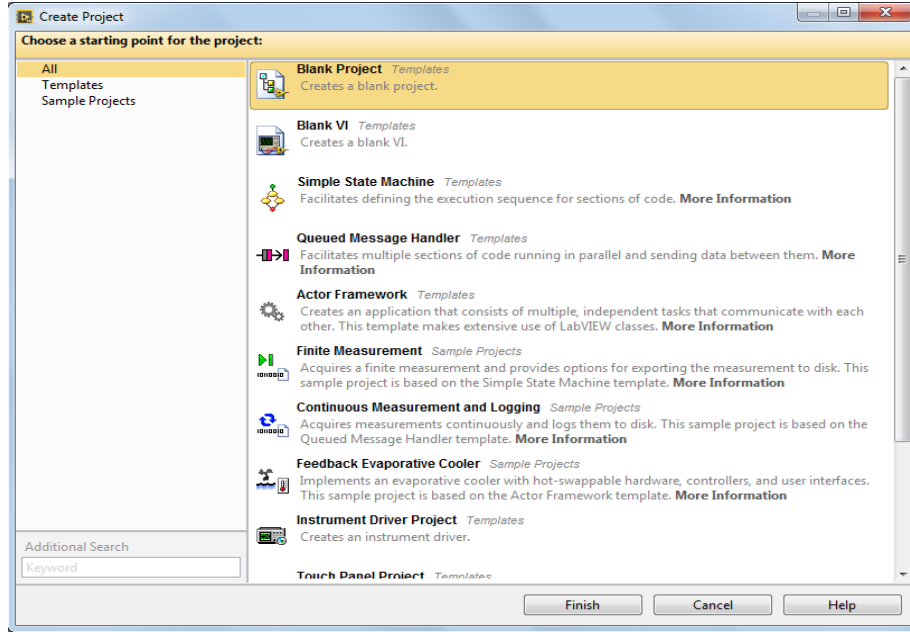
b)

Şekil 3.17. a) RPI IP adresinin eşleşmesi b) RPI yazılımının kurulması

RPI yazılımı kurulduktan sonra Şekil 3.18'deki gibi LabVIEW 2014 programını açıp Create Project tuşuna basarak yeni proje oluşturulur.



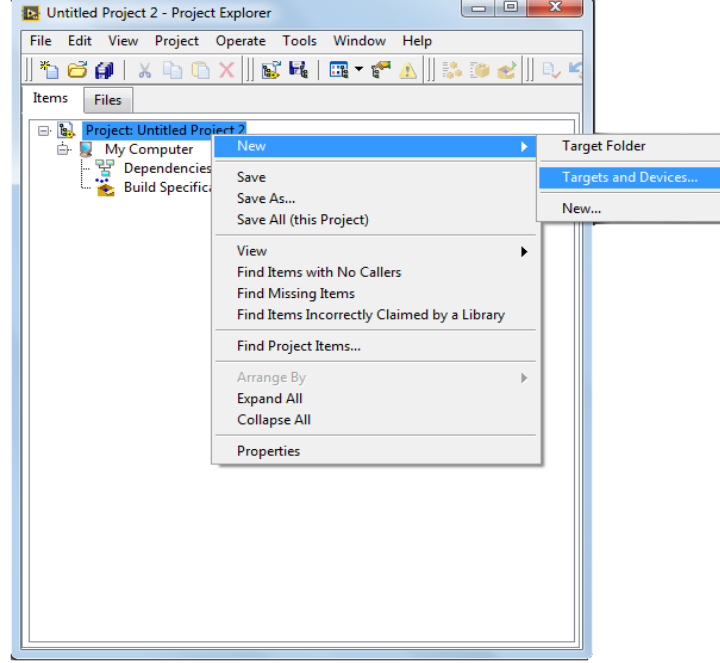
a)



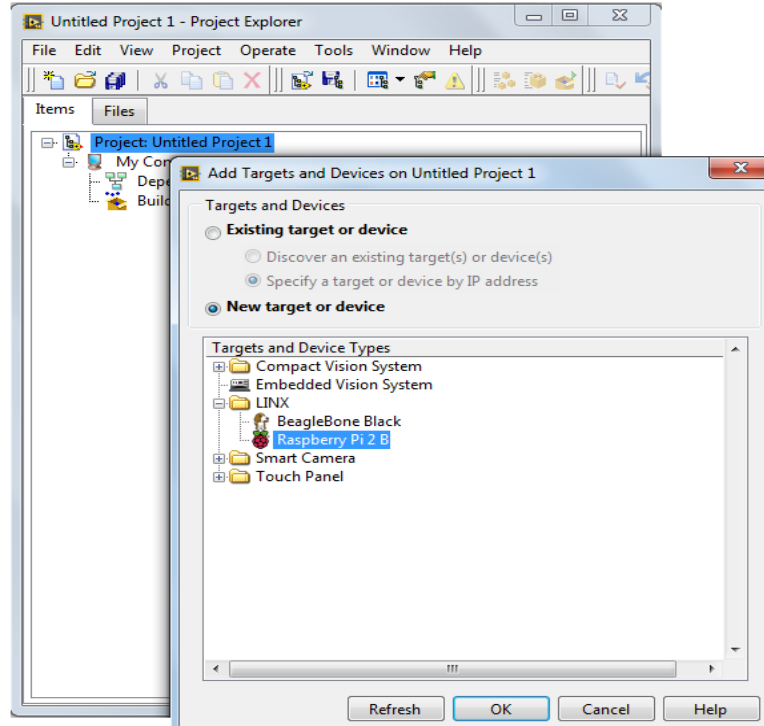
b)

Şekil 3.18. a) LabVIEW programının açılması b) Yeni boş proje oluşturma

Açılan yeni projede Şekil 3.19'da gösterildiği gibi **Project > New > Targets and Devices** tıklanarak hedeflenen aygıtlar arasından LINX dosyasının içinde Raspberry Pi seçilir.



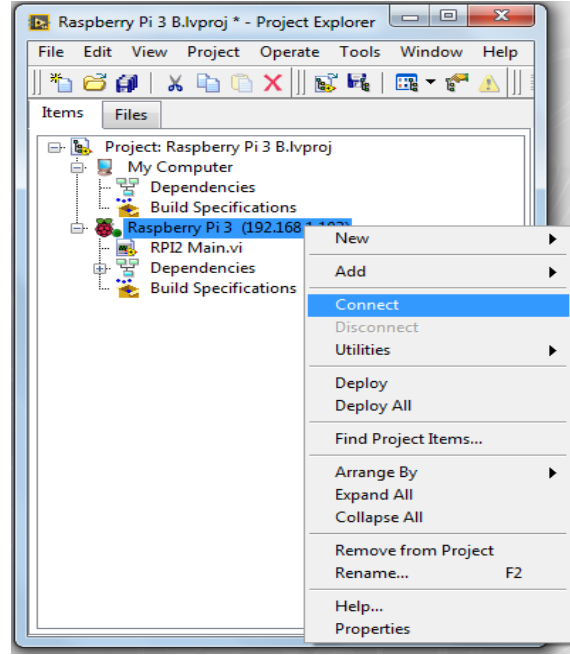
a)



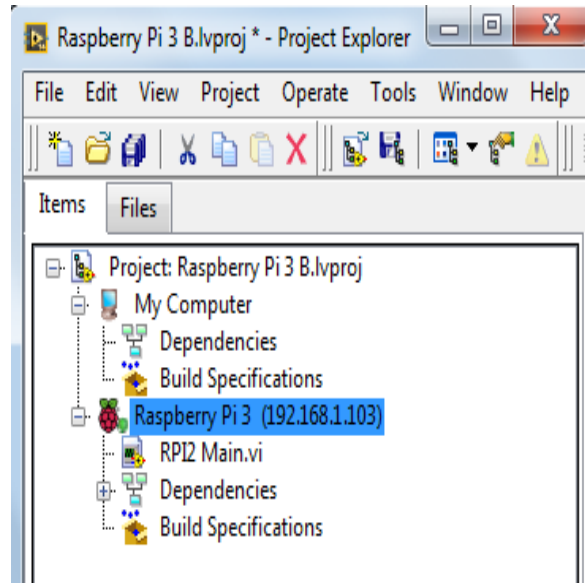
b)

Şekil 3.19. a) Hedef aygıt arama b) Hedeflenen aygıtın seçilmesi

Hedef aygıt olan Raspberry Pi seçildikten sonra proje dosyasının içinde Şekil 3.20'de gösterildiği gibi yer alan RPI 3 ikonuna bağlantının sağlanması için sağ tıklayarak connect tuşuna basılır. Böylece LabVIEW ile RPI 3 arasındaki bağlantı sağlanmış olur. Bağlantının olup olmadığını ise RPI 3 ikonu üzerindeki yeşil ışıktan anlaşılmaktadır.



a)

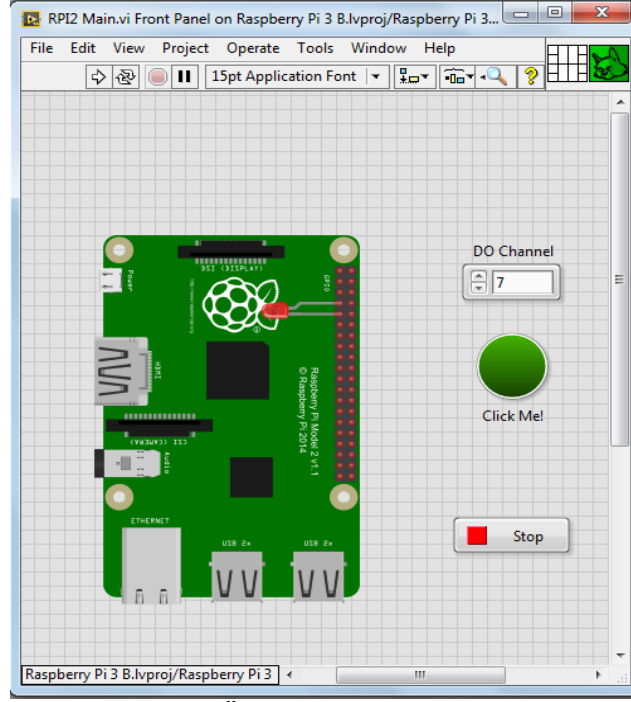


b)

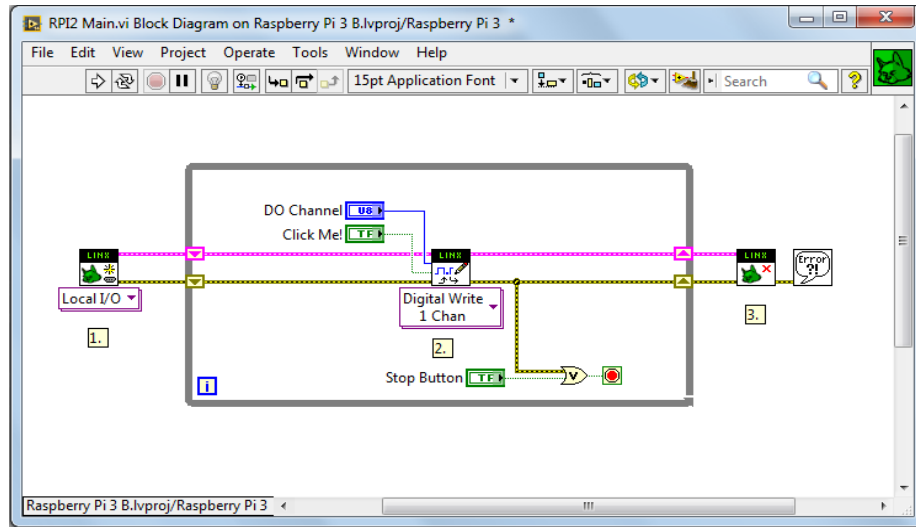
Şekil 3.20. a) RPI'ye bağlanma b) RPI'de bağlantının olduğunu gösteren ikon

3.4.4.1. LabVIEW ve RPI 3 Haberleşmesi Örnek Uygulama

LabVIEW ve RPI 3'ün haberleşmesine örnek olarak yapılan bir uygulama aşağıda gösterilmiştir. LabVIEW'in front panelinde RPI 3'e Şekil 3.21'de gösterildiği gibi bir led bağlanmıştır. Led'in bir bacağı 7 numaralı pine diğer bacağı toprağa bağlanmıştır.



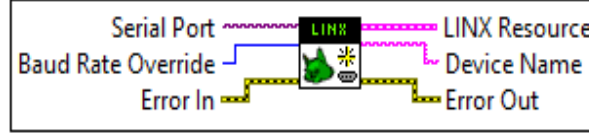
Şekil 3.21. Örnek uygulama front panel



Şekil 3.22. Örnek uygulamanın blok diyagramı

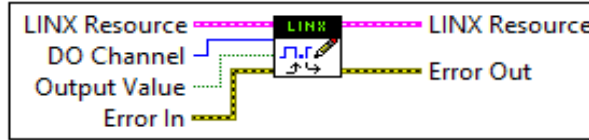
Şekil 3.22'de yapılan uygulamanın blok diyagramı gösterilmiştir. Programda bulunan fonksiyonları açıklarsak;

Şekil 3.23'te gösterilen ilk fonksiyon **open fonksiyonu** olup bu fonksiyonun görevi iletişimin nasıl sağlanacağını belirlemesidir. RPI ile bağlantının kurulması için gerekli IP adresi bu fonksiyonda görülmektedir.



Şekil 3.23. Open fonksiyonu

Şekil 3.24'te gösterilen ikinci fonksiyon **digital write fonksiyonu** olup bu fonksiyonun görevi RPI'de kullanılacak dijital giriş adreslerinin belirlenmesini sağlamaktadır.



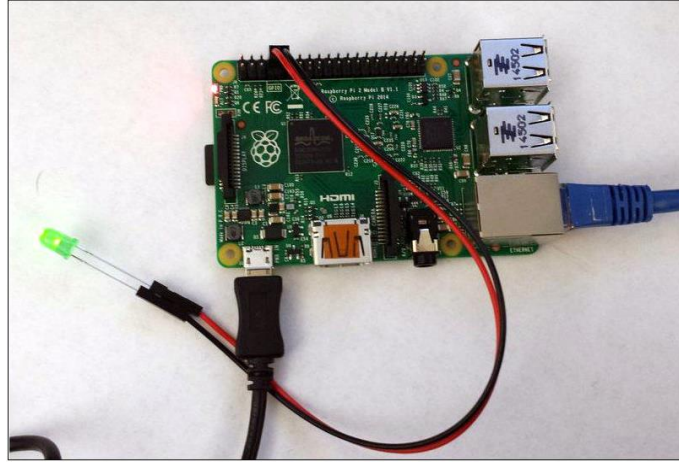
Şekil 3.24. Digital Write fonksiyonu

Şekil 3.25'te gösterilen üçüncü fonksiyon ise **close fonksiyonu** olup bu fonksiyonun görevi yapılan programlamanın bitişi kontrol etmektir. Her programlama close fonksiyonu ile sonlanmaktadır.



Şekil 3.25. Close Fonksiyonu

Uygulama çalıştırılıp Şekil 3.21'de gösterilen front paneldeki LED'in üzerine tıklandığında, Şekil 3.26'da RPI'ye bağlanan LED'in kablosuz olarak uzaktan yakılıp söndürüldüğü görülmektedir.



Şekil 3.26. RPI 3'e bağlı led'i yakma

3.5. DS18B20 Sıcaklık Sensörü

Şekil 3.27'de gösterilen Maxim Dallas firması tarafından üretilen DS18B20 sıcaklık sensörü daha önceki bölümde anlatıldığı gibi NTC, PTC, LM35 gibi sıcaklık sensörlerinden farklıdır. Bu sıcaklık algılayıcıları analog sensörler olduğundan dolayı sayısal sistemler ile kullanılabilmesi için Analog/Sayısal dönüştürücü kullanılması gerekirken DS18B20 sensörü ise sayısal çıkış vermektedir. DS18B20 sıcaklık sensörü tek kablo üzerinden 1-hat (1-Wire) haberleşme protokolünü kullanarak sayısal iletim yapmaktadır [7].



Şekil 3.27. DS18B20 sıcaklık sensörü

Teknik özellikleri;

- 3.0V - 5.5V güç / veri ile kullanılabilir
- $\pm 0.5^{\circ}\text{C}$ -10 $^{\circ}\text{C}$ ile + 85 $^{\circ}\text{C}$ arası doğruluk
- Kullanılabilir sıcaklık aralığı: -55 $^{\circ}\text{C}$ - 125 $^{\circ}\text{C}$ (-67 $^{\circ}\text{F}$ - + 257 $^{\circ}\text{F}$)
- 9 ila 12 bit seçilebilir çözünürlük.
- 64 bitlik

- 1-Wire ara yüzü kullanır- iletişim için sadece bir dijital pin gerektirir
- Maksimum 750ms içerisinde sıcaklık ölçümü yapabilmektedir.
- Sıcaklık sınırı alarm sistemi
- 3 telli ara yüz:
 - Kırmızı tel - VCC
 - Siyah tel - GND
 - Sarı tel - DATA
- Paslanmaz çelik boru 6mm çapında 35mm uzunluğunda
- Kablo çapı: 4mm
- Uzunluk: 90cm

3.6. Isıtıcı

Devrede 12V 120W gücünde led göstergeli, plastik dış kılıflı Şekil 3.28’de gösterilen su ısıtıcısı kullanılmıştır. 0,5 litre su haznesi bulunmakta olup, rezistans direnci 1,2 Ohm’dur.



Şekil 3.28. 12V Isıtıcı

3.6.1 Isıtıcının Güç Sürücü Devresi

Şekil 3.29’da gösterilen ve uygulamada kullanılan MOSFET’li anahtarlama kartı DC 5-36V, maksimum 15A’e kadar yükleri anahtarlama için kullanılabilir. Tıpkı röle kartı gibi çalışır. Röle kartından farklı olarak mekanik parça içermediğinden dolayı PWM sinyali ile ölçekli olarak kontrol imkânı da sunar. Örneğin DC motor hız kontrolü gibi [43].



Şekil 3.29. MOSFET sürücü kartı

Teknik özellikler:

- Çalışma gerilimi: DC 5-36V
- Tetikleme sinyal seviyesi: DC 3.3-20V (mikrokontrolcü G/Ç pinlerine bağlanabilir)
- PWM maksimum 20kHz sinyal destekler
- Çıkış akımı: 15A sürekli, 30A anlık
- Çıkış gücü: 400W maksimum
- Çalışma sıcaklık aralığı: -40°C ~85°C
- Boyutlar: 34 × 17 × 12mm

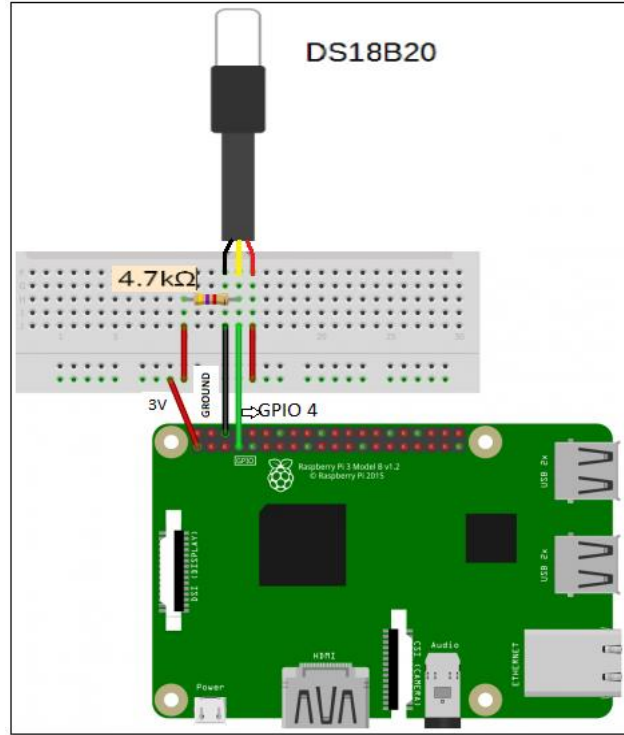
4. LABVIEW VE RASPBERRY Pi 3 KULLANARAK GERÇEK ZAMANLI BİR SICAKLIK KONTROL UYGULAMASI

4.1. Giriş

Bu çalışmada gerçekleştirilen kontrol işlemi için PID kontrolör tercih edilmiştir. Gerçeklenen sistem dört adımdan oluşmaktadır. İlk adımda Raspberry Pi 3 üzerinden DS18B20 sensörü ile ısıtıcıdaki suyun sıcaklık verileri toplanmıştır. İkinci adımda toplanan sıcaklık verileri Raspberry Pi 3'ten LabVIEW'e Wi-Fi yolu ile kablosuz olarak aktarılmış ve sıcaklık-zaman grafiğinin LabVIEW ortamında çizimi sağlanmıştır. Üçüncü adımda deneysel olarak Raspberry ile ölçülen ve LabVIEW'e aktarılan sıcaklık verileri kullanılarak PID katsayılarını hesaplamak için ısıtıcının modeli belirlenmiştir. Isıtıcının açık çevrim birim basamak cevabı olan sıcaklık-zaman grafiğinin üzerinden ısıtıcının transfer fonksiyonu hesaplanmıştır. Isıtıcı transfer fonksiyonundan Ziegler-Nichols PID parametre tablosu yardımı ile PID katsayıları hesaplanmış ve PID kontrolörün transfer fonksiyonu elde edilmiştir. Dördüncü adımda PID kontrolörlü kapalı çevrim kontrol sisteminin blok diyagramı Matlab/Simulink'te modellenmiş ve hesaplanan PID parametreleri değiştirilerek elde edilen yeni değerler için sistemin basamak cevabı performansı incelenmiştir. Beşinci adımda ise başarılı performans gösteren PID katsayıları kullanılarak Raspberry Pi 3 ile gerçek zamanlı bir sıcaklık kontrol uygulaması gerçekleştirilmiştir.

4.1.1. Raspberry Pi 3 ile Sıcaklık Ölçümü

Raspberry Pi 3 ile sıcaklık ölçümü için gerçekleştirilen devre Şekil 4.1'deki gibi gösterilmiştir. RPI 3'ün GPIO 4 pinine sıcaklık sensörünün sarı renkli haberleşme (data) ucu bağlanmıştır. Data ucundan 4.7 Ohm'luk dirençle sensörün kırmızı kablosu ile bağlantı yapılmış ve RPI 3'ün 3V güç kaynağına bağlanmıştır. Son olarak sensörün siyah kablosu RPI 3'ün toprak ucuna bağlanmıştır [44, 45].



Şekil 4.1. RPI 3 ile sıcaklık ölçümü

DS18B20 sensörü One-Wire yöntemi ile iletişim kurduğundan, aşağıdaki adımları izleyerek RPI 3'deki tek tel iletişimini etkinleştirilmiştir. Bunun için gereken adımlar ise aşağıdaki gibi olup Şekil 4.2 ve Şekil 4.3'te gösterildiği gibi uygulanmıştır [46, 47].

Adım 1: Komut yazma terminali açılır ve config dosyasını açmak için aşağıdaki komutu kullanılır.

```
sudo nano /boot/config.txt
```

Adım 2: Bu dosyanın altına aşağıdaki komut girilir.

```
dtoverlay=w1-gpio
```

Adım 3: Bir kez *ctrl x* ve sonra *y tuşlarına* basılarak kayıt işlemi gerçekleştirilir ve çıkılır. Sonrasında aşağıdaki komut çalıştırılır ve Raspberry Pi yeniden başlatılır.

```
sudo reboot
```

Adım 4: Raspberry Pi yeniden başlatıldığında, doğru modülleri yükleyebilmek için modprobe komutunun çalıştırılması gerekmektedir.

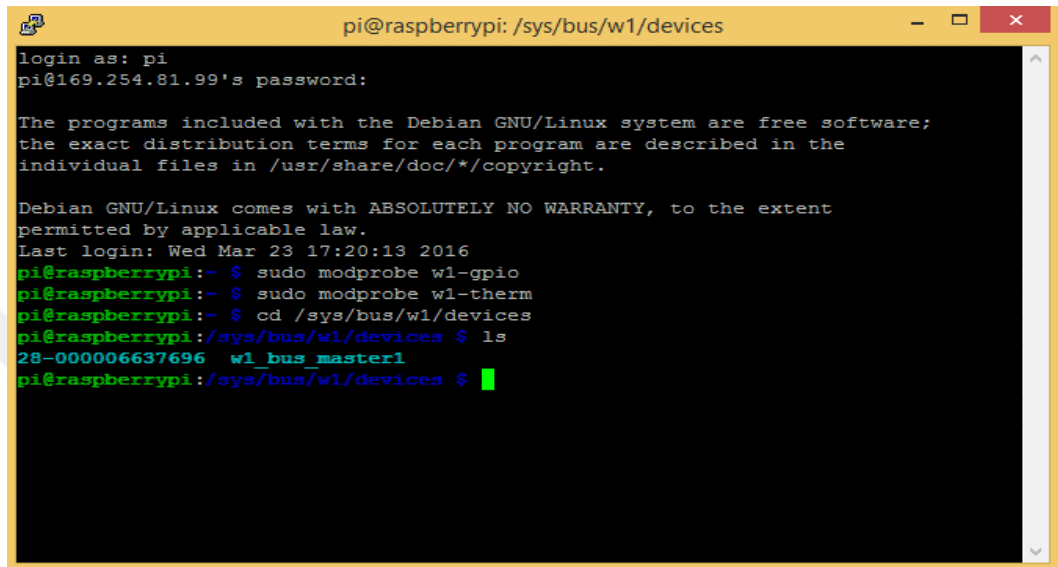
```
sudo modprobe w1-gpio
```

```
sudo modprobe w1-therm
```

Adım 5: Cihazlar dizinine girmek için ls komutu yazılır ve bu dizindeki klasörü / dosyaları görmek için kullanılır.

```
cd /sys/bus/w1/devices
```

```
ls
```



```
pi@raspberrypi: /sys/bus/w1/devices
login as: pi
pi@169.254.81.99's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Mar 23 17:20:13 2016
pi@raspberrypi:~$ sudo modprobe w1-gpio
pi@raspberrypi:~$ sudo modprobe w1-therm
pi@raspberrypi:~$ cd /sys/bus/w1/devices
pi@raspberrypi:/sys/bus/w1/devices$ ls
28-000006637696 w1_bus_master1
pi@raspberrypi:/sys/bus/w1/devices$
```

Şekil 4.2. RPI' ye DS18B20 haberleşme modül kurulumu [46]

Adım 6: Aşağıdaki komut çalıştırılır, cd komutunu kullanarak dizinde görünen numara değiştirilir. (Birden fazla sensör varsa, birden fazla dizin olacaktır)

```
cd 28-000007602ffa
```

Adım 7: Aşağıdaki komutu çalıştırılır.

```
cat w1_slave
```

```
pi@raspberrypi: /sys/bus/w1/devices/28-000006637696
login as: pi
pi@169.254.81.99's password:

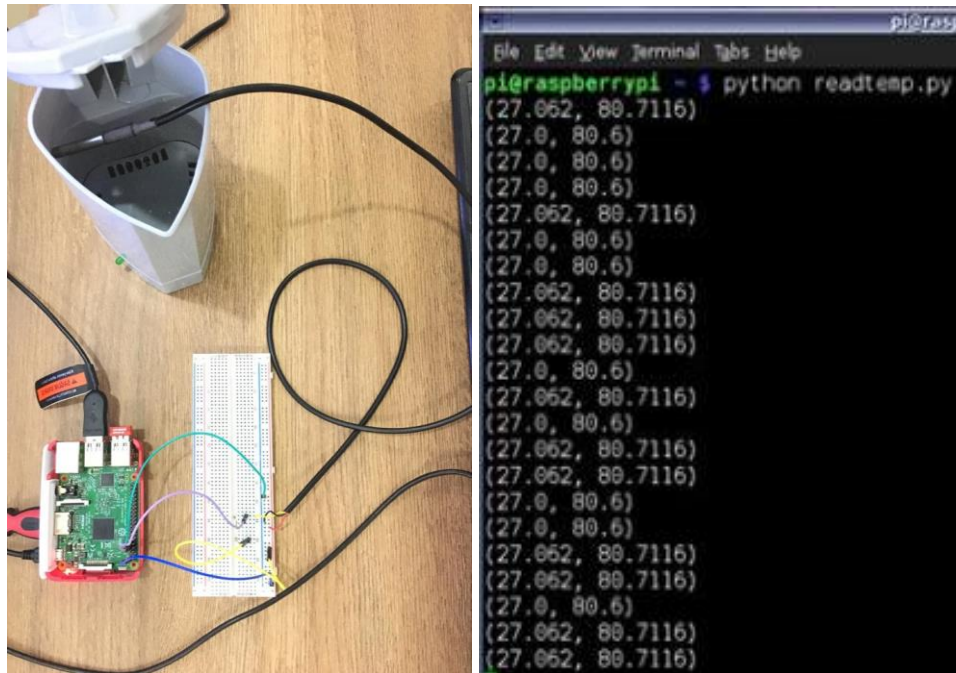
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Mar 23 17:20:13 2016
pi@raspberrypi:~$ sudo modprobe w1-gpio
pi@raspberrypi:~$ sudo modprobe w1-therm
pi@raspberrypi:~$ cd /sys/bus/w1/devices
pi@raspberrypi:/sys/bus/w1/devices$ ls
28-000006637696  w1_bus_master1
pi@raspberrypi:/sys/bus/w1/devices$ cd 28-000006637696
pi@raspberrypi:/sys/bus/w1/devices/28-000006637696$ cat w1_slave
ca 01 4b 46 7f ff 06 10 65 : crc=65 YES
ca 01 4b 46 7f ff 06 10 65 t=28625
pi@raspberrypi:/sys/bus/w1/devices/28-000006637696$
```

Şekil 4.3. RPI'ye DS18B20 haberleşme modül kurulumunun tamamlanması [48]

Adım 8: İlk satırın sonunda cevap EVET veya HAYIR olmalıdır. EVET ise, sıcaklık değerinin yanında ikinci bir satır görünmelidir Burada sıcaklık değeri t = 28625'tir, yani 28.625 santigrat derece sıcaklık anlamına gelmektedir [48].

Yukarıdaki adımlar tamamlandıktan sonra bir ısıtıcı içindeki suyun sıcaklığı Şekil 4.4'te gösterildiği gibi RPI 3 üzerinden yapılan yazılım ile her saniye ölçülmüştür.

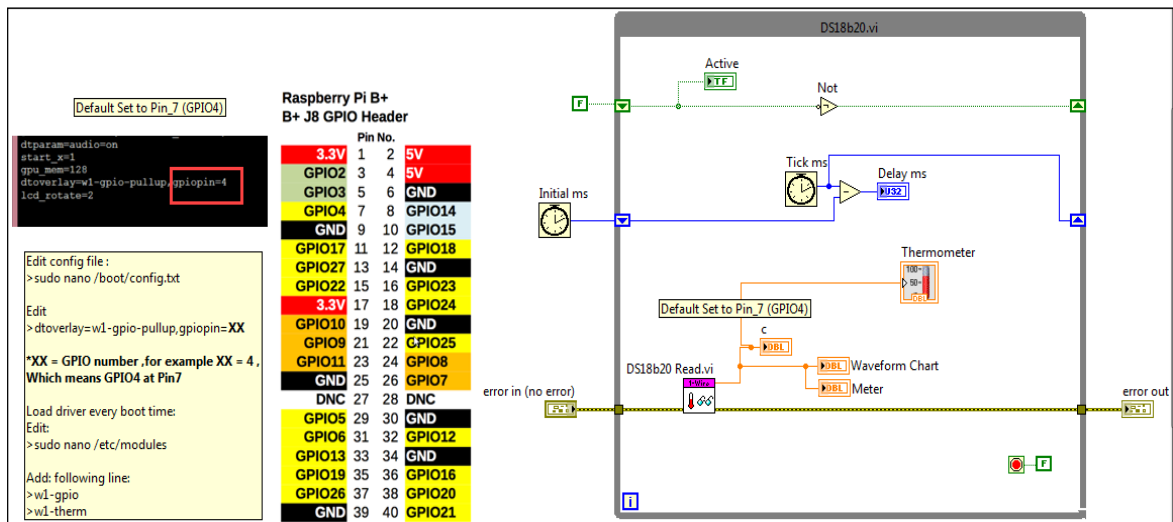


Şekil 4.4. RPI 3 ile sıcaklık ölçümü

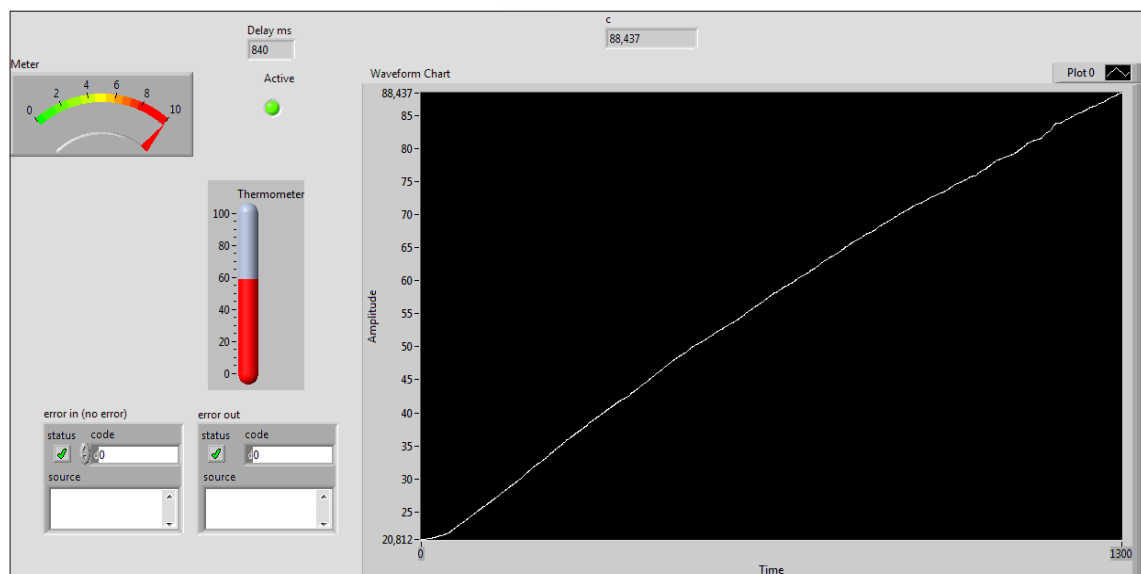
4.1.2. Sıcaklık Verilerinin LabVIEW'e Kablosuz Aktarılması

Bu bölümde RPI 3 ile yapılan sıcaklık ölçümünde elde edilen verilerin kablolu olarak LabVIEW'e aktarılması anlatılacaktır. LabVIEW ile Raspberry Pi 3'ün haberleşmesi başlıklı bölümünde bahsedilen ve kurulan eklentiler sonucunda RPI ile LabVIEW'in haberleşmesi tamamlanmış ve sıcaklık verilerinin aktarımı ise aşağıda gösterildiği gibi yapılacaktır.

Şekil 3.8’de daha önce yüklenen QwaveSys RPI package içinde bulunan DS18B20 kütüphanesi kullanılarak RPI 3 ile elde edilen sıcaklık verilerinin LabVIEW’e aktarımını gösteren programlama Şekil 4.5’de gösterilmiştir. Elde edilen sıcaklık verilerinin grafiği ise Şekil 4.6’da gösterilmiştir.



Şekil 4.5. Sistemin LabVIEW’de programlanması

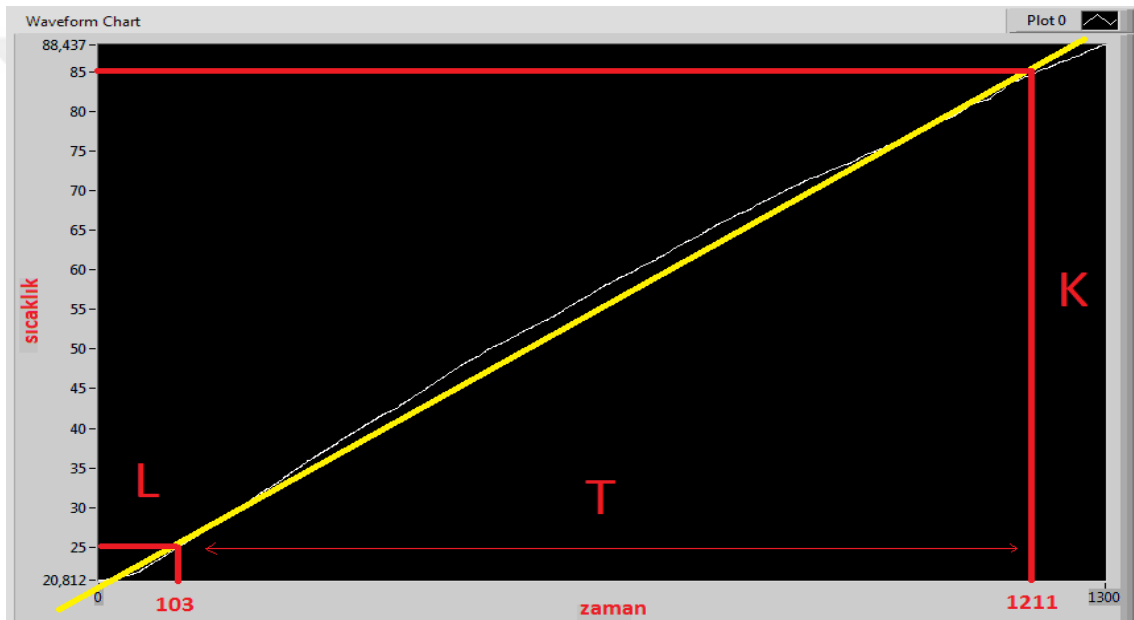


Şekil 4.6. LabVIEW’de elde edilen sıcaklık-zaman grafiği

4.1.3. Isıtıcının Modellenmesi ve PID Katsayılarının Bulunması

Isıtıcı modellenmesi Ali Yüce tarafından yazılan [8] numaralı tezdeki yöntem kullanılarak gerçekleştirilmiş ve ısıtıcının açık çevrim basamak cevabı için gerçekleştirilen deneyde ısıtıcıdaki suyun sıcaklığı Raspberry Pi 3 ve DS18B20 sıcaklık sensörü ile her saniye ölçülmüştür. Ölçülen sıcaklık verilerinin LabVIEW ortamında çizdirilen sıcaklık-zaman grafiği Şekil 4.6’te verilmiştir.

Şekil 2.17’de gösterilen açık çevrim birim basamak cevabı modeli temel alınarak Şekil 4.7 üzerinden denklem (3.5)’in parametreleri denklem 4.1’deki gibi hesaplanmıştır.



Şekil 4.7. Isıtıcının deney sonucu elde edilen açık çevrim basamak cevabı

$L=103s$ (eğimin başladığı nokta) (25 derece)

$T = 1211s$ (ikinci nokta) (85 derece)

$$K = \frac{(85 - 25)}{(88.437 - 20.812)} = 0.888 \quad (4.1)$$

Yapılan deneyde her saniye için alınan sıcaklık verileri Çizelge 4.1’de gösterilmiştir. İki nokta arasında alınan referans değerleri 103. saniyede 25 derece ve 1211. saniyede 85 derecedir.

Çizelge 4.1. Sıcaklık-zaman verileri

Time - Plot	Amplitude - Plot
0	20,812
1	20,812
2	20,812
3	20,812
4	20,875
.	
.	
.	
101	24,875
102	24,937
103	25
104	25,062
105	25,125
106	25,187
.	
.	
.	
1208	84,812
1209	84,875
1210	84,937
1211	85
1212	85,062
1213	85,125

Isıtıcı transfer fonksiyonu, (4.1) numaralı denklemlerinde elde edilen değerlerin 3.5 numaralı denklemde yerine yerleştirilmesiyle denklem (4.2)’deki gibi elde edilir.

$$G(s) = \frac{0.888e^{-103s}}{(1+1211s)} \quad (4.2)$$

Denklem (4.2)’de verilen ısıtıcı transfer fonksiyonundan yola çıkarak Çizelge 2.2’de verilen Ziegler-Nichols PID parametre tablosu yardımı ile PID kontrolörün parametreleri denklem (4.3)’deki gibi hesaplanır.

$$K_p = \frac{1.2 \times 1211}{0.88 \times 103} = 15.88 \quad T_i = 2 \times 103 = 206 \quad T_d = 0.5 \times 103 = 51.5 \quad (4.3)$$

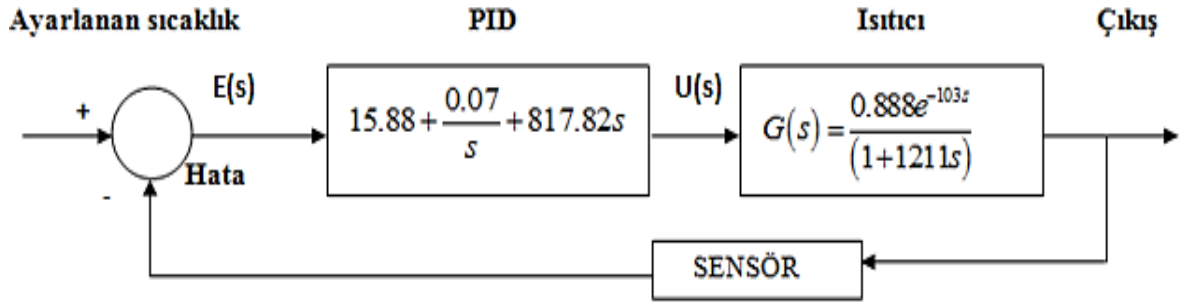
Denklem (4.3)’te hesaplanan parametrelere göre PID kontrolörün transfer fonksiyonu 3.3 numaralı denkleme yerleştirilerek denklem (4.4) gibi elde edilir.

$$\frac{U(s)}{E(s)} = 15.88 \left[1 + \frac{1}{206s} + 51.5s \right] \quad (4.4)$$

Denklem (4.3)'de hesaplanan K_p , T_i ve T_d parametrelerinin denklem (3.4)'de yerine koyulması ile PID kontrolörün K_p , K_i ve K_d parametreleri cinsinden hesabı denklem (4.5) gibi yapılır.

$$K_p = 15.88 \quad , \quad K_i = \frac{K_p}{T_i} = \frac{15.88}{206} = 0.07 \quad , \quad K_d = K_p \times T_d = 15.88 \times 51.5 = 817.82 \quad (4.5)$$

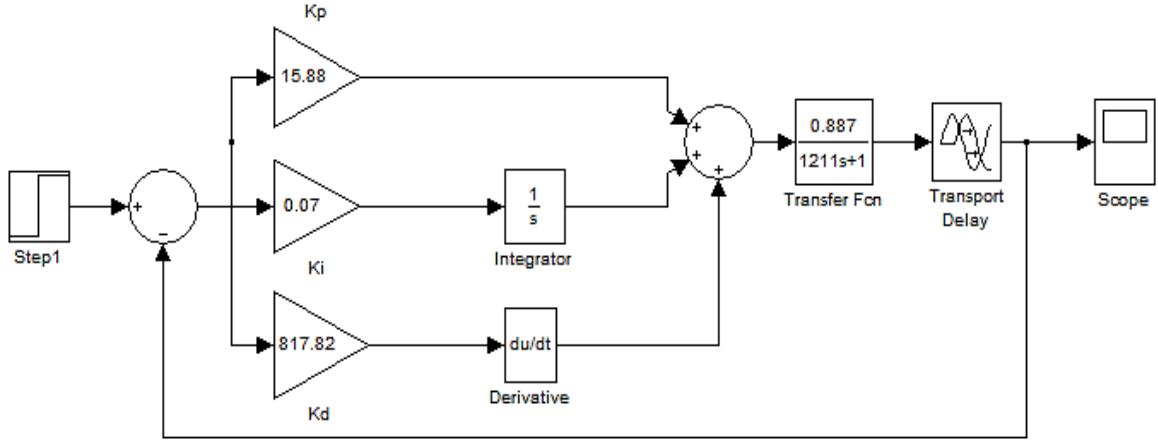
Yapılan hesaplamalar sonucunda kapalı çevrim kontrol sisteminin blok diyagramı Şekil 4.8'deki gibi olur.



Şekil 4.8. Hesaplanan kapalı çevrim kontrol sisteminin blok diyagramı

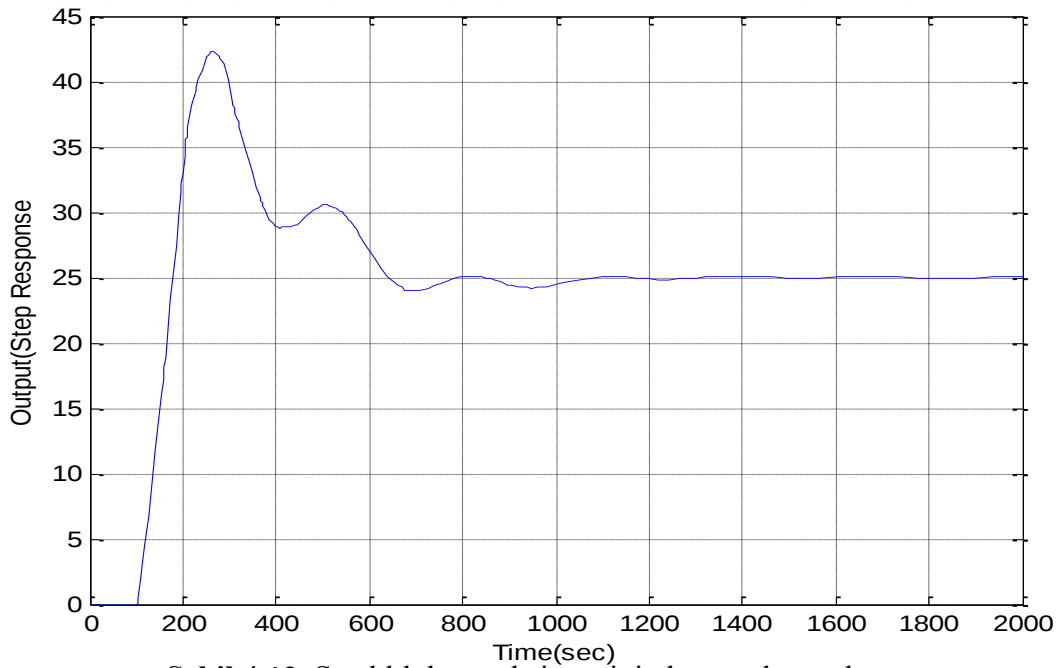
4.1.4. Sıcaklık Kontrol Sisteminin Matlab/Simulink ile Modellenmesi

Bu kısımda Şekil 4.8'deki kapalı çevrim kontrol sisteminin blok diyagramı Matlab/Simulink' te modellenmiştir. Denklem 4.5'te verilen PID kontrolörün performans analizi Şekil 4.9'da gösterilen simulink model üzerinde yapılmıştır.



Şekil 4.9. Sıcaklık kontrol sisteminin Matlab/Simulink'te modellemesi

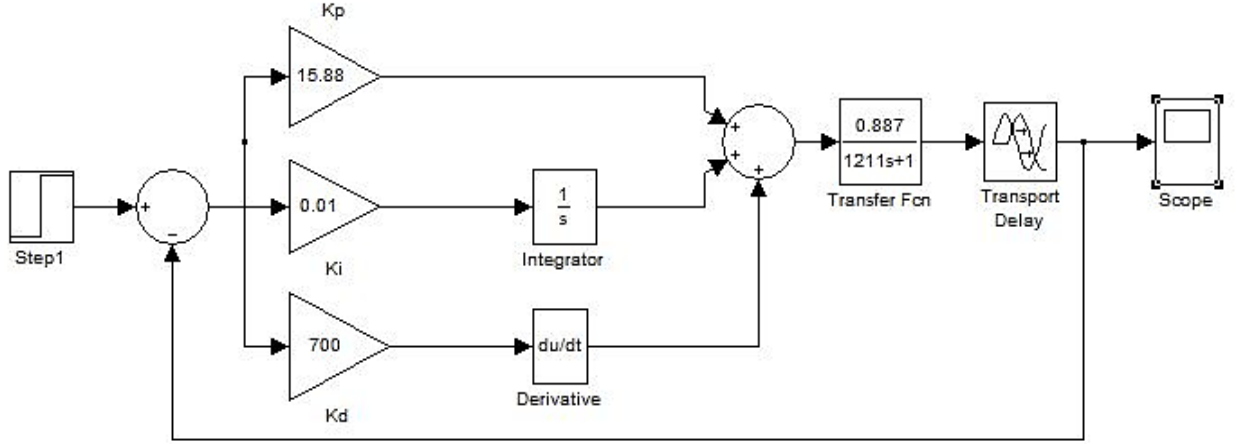
Sistem çalıştırılmış ve sistemin basamak cevabının simülasyon sonucu Şekil 4.10'da gösterilmiştir.



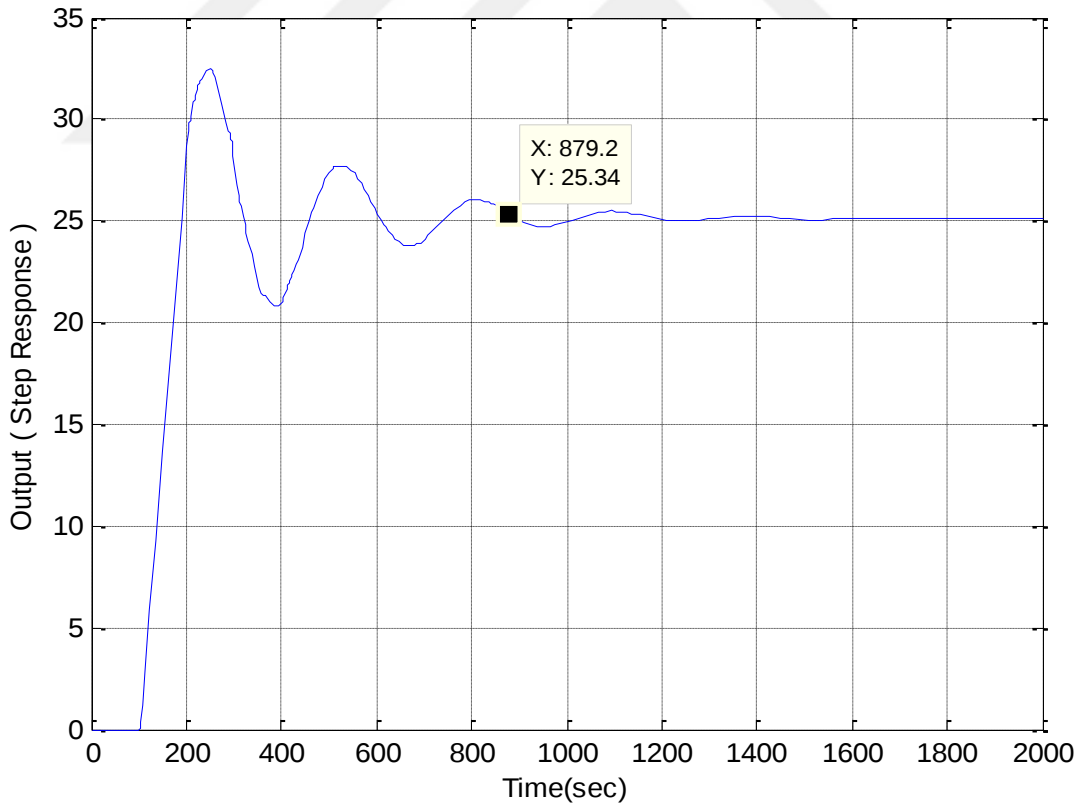
Şekil 4.10. Sıcaklık kontrol sisteminin basamak cevabı

Şekil 4.10'daki sonuç incelendiğinde yüzde aşım oranının yüzde 68 olduğu görülmüştür. Bu nedenle daha uygun yüzde aşım performansı için farklı K_i ve K_d değerleri taranmıştır.

Tarama sonucunda tespit edilen yeni PID kontrolör kullanılarak elde edilen sistem modeli Şekil 4.11’de gösterilmiştir. Değişen katsayılar sonucunda sistemin basamak cevabı Şekil 4.12’de gösterilmiştir. Burada sistemin yüzde aşım performansının yüzde 32’ye düşürüldüğü görülmektedir.



Şekil 4.11. Yeni PID kontrolörle sistemin modeli

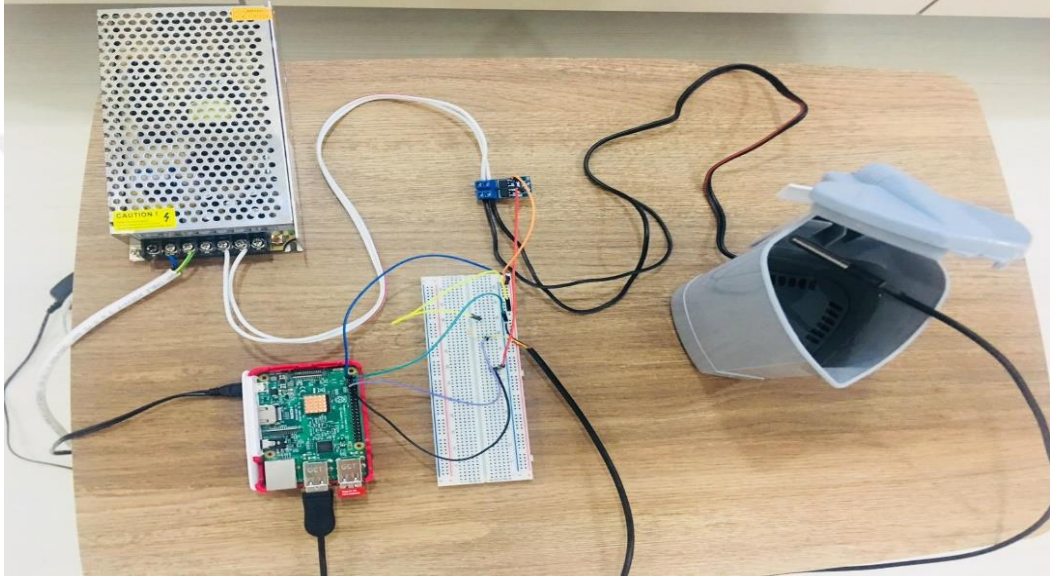


Şekil 4.12. Yeni PID kontrolörle elde edilen basamak cevabı

4.1.5. Raspberry Pi 3 ile Sisteminin Gerçek Zamanlı Kontrolü

Bu kısımda Raspberry Pi 3 ve DS18B20 sıcaklık sensörü kullanarak gerçek zamanlı bir sıcaklık kontrol uygulamasının nasıl gerçekleştirildiği anlatılmıştır. PID kontrolör algoritması Raspberry Pi 3 üzerinde çalıştırılmıştır.

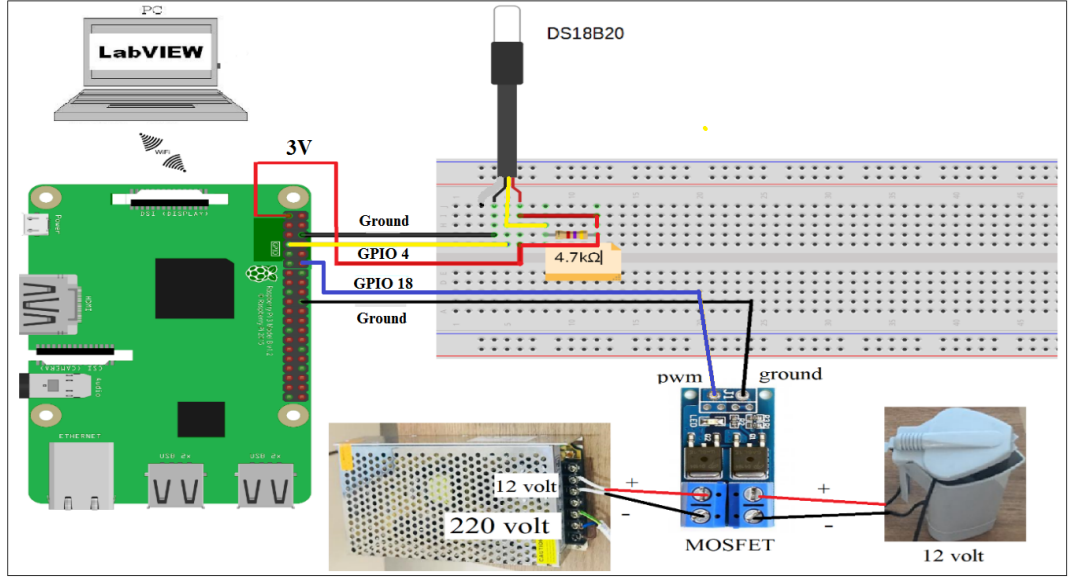
Sıcaklık kontrol sisteminin deney anına ait genel bir görünüşü Şekil 4.13'te gösterilmiştir.



Şekil 4.13. Sıcaklık kontrol sistemin genel görünüşü

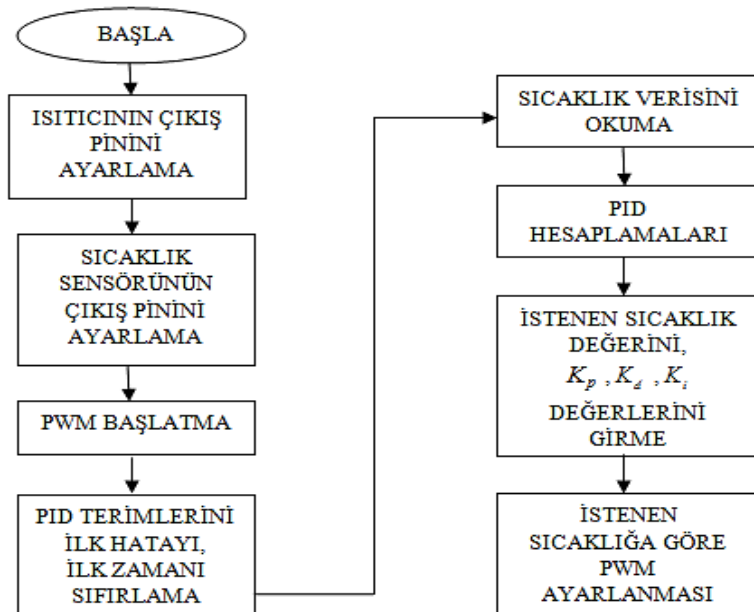
Gerçekleştirilen kontrol sisteminin şematik gösterimi ise Şekil 4.14'te verilmiştir. Bu gerçek zamanlı uygulama için kullanılan malzemeler aşağıda listelenmiştir.

- Raspberry Pi 3,
- DS18B20 Dijital Sıcaklık Sensörü,
- 220V/12V Güç Kaynağı,
- 15A 400W Mosfet Sürücü Kartı
- 12V Su Isıtıcısı.
- 4.7 K Direnç
- Mini USB Çıkışlı Adaptör
- USB Çıkışlı Fare ve Klavye
- HDMI Kablosu



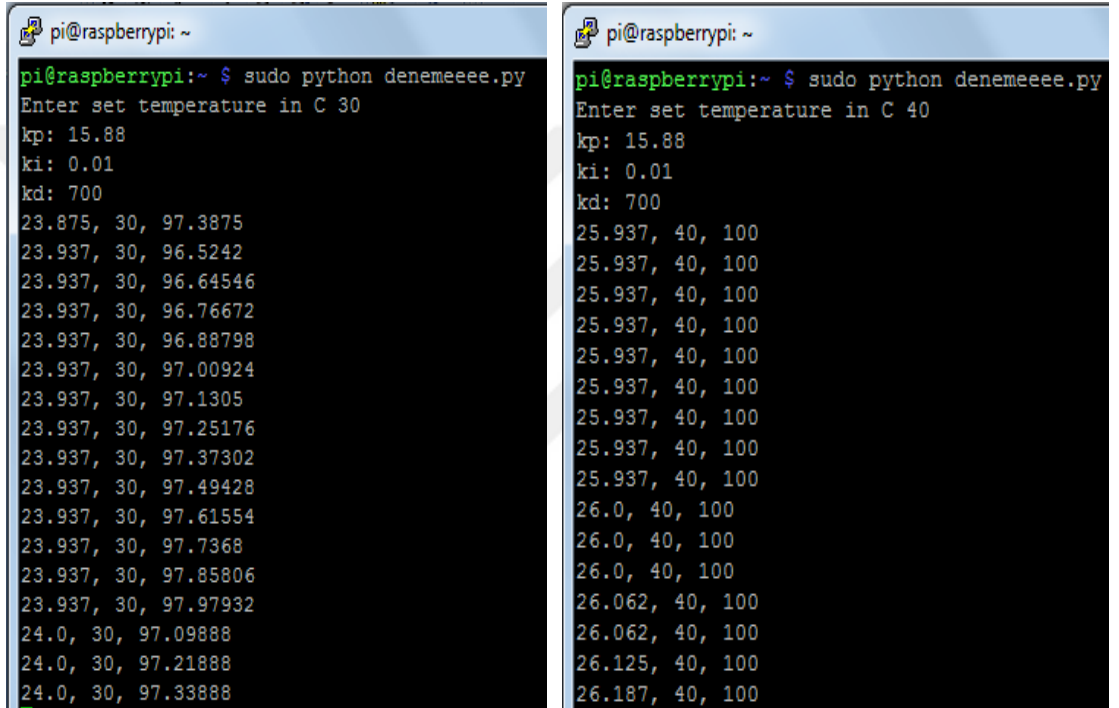
Şekil 4.14. Gerçekleştirilen sıcaklık kontrol sisteminin bağlantı şeması

Şekil 4.14'te gösterilen bağlantı şemasında Raspberry Pi 3'ün giriş olarak tanımlanan GPIO 4 numaralı ayağına DS18B20 sıcaklık sensörü bağlanmıştır. 12 volt gerilim ile çalışabilen su ısıtıcımızı kontrol edebilmek için Raspberry Pi 3 ile uyumlu olan 15A 400W Mosfet sürücü kartı kullanılmıştır. Sürücü kartımızın PWM çıkışı Raspberry Pi 3'ün PWM çıkışı olan GPIO 18 numaralı ayağına bağlanmıştır. Böylece Raspberry Pi 3 ile algoritması yazılan PID kontrolörün çıkışına uygun olarak ayarlanan PWM değeri ile ısıtıcının kontrolü sağlanmış olur. Raspberry Pi 3 üzerinde çalışan programın akış diyagramı Şekil 4.15'te verilmiştir.



Şekil 4.15. Raspberry Pi 3 üzerinde yapılan yazılımın akış diyagramı

Raspberry Pi 3'te gerçekleştirilen yazılımda ilk deney için sıcaklık set değeri 30, ikincisinde ise 40 derece olarak alınmıştır. Gerçek zamanlı sistemin kontrolünde iyileştirilmiş PID kontrolör katsayıları olan $K_p = 15.88$, $K_i = 0.01$ ve $K_d = 700$ değerleri girilmiştir. 30 derece için çalıştırılan program Şekil 4.16'da gösterilmiş olup elde edilen veriler ekranda 3 sütun halinde gösterilmiştir. İlk sütunda ölçülen sıcaklık değeri, ikincisinde set edilen sıcaklık değeri ve üçüncüsünde ise PWM' in 0 ile 100 arasına normalize edilmiş kontrol sinyali değeri gösterilmiştir.

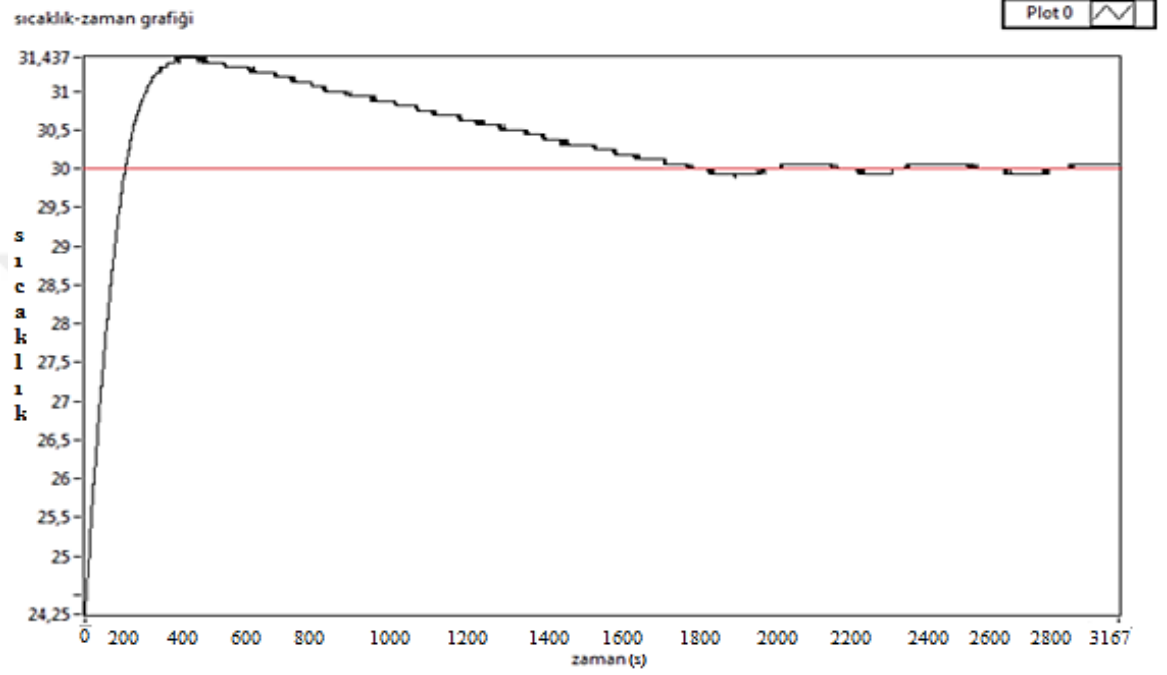


```
pi@raspberrypi: ~  
pi@raspberrypi:~ $ sudo python denemeeee.py  
Enter set temperature in C 30  
kp: 15.88  
ki: 0.01  
kd: 700  
23.875, 30, 97.3875  
23.937, 30, 96.5242  
23.937, 30, 96.64546  
23.937, 30, 96.76672  
23.937, 30, 96.88798  
23.937, 30, 97.00924  
23.937, 30, 97.1305  
23.937, 30, 97.25176  
23.937, 30, 97.37302  
23.937, 30, 97.49428  
23.937, 30, 97.61554  
23.937, 30, 97.7368  
23.937, 30, 97.85806  
23.937, 30, 97.97932  
24.0, 30, 97.09888  
24.0, 30, 97.21888  
24.0, 30, 97.33888  
pi@raspberrypi:~ $ sudo python denemeeee.py  
Enter set temperature in C 40  
kp: 15.88  
ki: 0.01  
kd: 700  
25.937, 40, 100  
25.937, 40, 100  
25.937, 40, 100  
25.937, 40, 100  
25.937, 40, 100  
25.937, 40, 100  
25.937, 40, 100  
25.937, 40, 100  
25.937, 40, 100  
25.937, 40, 100  
26.0, 40, 100  
26.0, 40, 100  
26.0, 40, 100  
26.062, 40, 100  
26.062, 40, 100  
26.125, 40, 100  
26.187, 40, 100
```

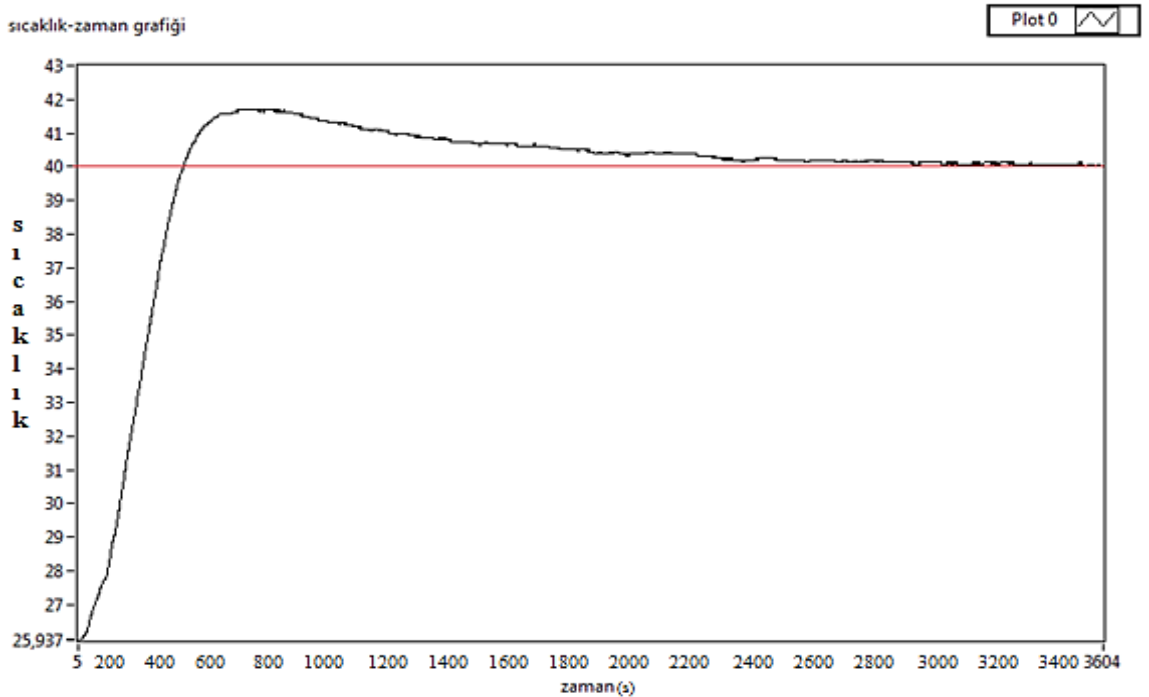
Şekil 4.16. Sistemin 30 ve 40 derece set değeri için sonuçları

30 derecede ve 40 derecede elde edilen sıcaklık kontrolü deneyine ait veriler Şekil 4.17 ve 4.18'de gösterilmiştir.

Şekil 4.17’de görüldüğü gibi 30 dereceye ayarlanan ısıtıcıdaki suyun sıcaklığı yüzde 4.79 aşım ile 31.437 dereceye kadar yükselmiştir ve daha sonra 30 dereceye yaklaşarak o seviyede kalmıştır. Benzer biçimde Şekil 4.18’de 40 derece set değeri için elde edilen basamak cevabında da yaklaşık olarak yüzde 4.75 aşım vardır. Sonuçlar sıcaklık kontrolü gibi yüksek ataletle sahip sistemlerin kontrolü için oldukça tatmin edicidir.



Şekil 4.17. 30 derece set değerinde kontrol edilen sistemin basamak cevabı



Şekil 4.18. 40 derece set değerinde kontrol edilen sistemin basamak cevabı

5. SİSTEMİN KESİRLİ DERECELİ PID ($PI^\lambda D^\mu$) İLE KONTROLÜ

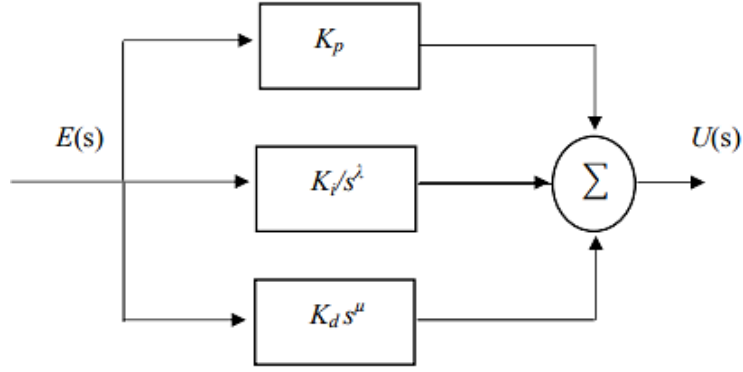
5.1.Giriş

Kesirli hesaplama yönteminin tarihi eskiye dayanmasına rağmen o yıllarda o kadar fazla ilgi görmemiştir. Fakat son yıllarda özellikle bilgisayar teknolojisinin gelişimine bağlı olarak çok popüler bir alan olmuştur [49].

Kesir dereceli matematik sistemleri ilk kez 1695 yılında L'Hospital tarafından Leibniz'e yazılan mektupta geçen bir soru ile teori olarak ortaya atılmıştır. Mektupta bir fonksiyonun n . dereceden türevi olan $D^n f(x)/Dx^n$ için $n=1/2$ olursa nasıl bir sonuç çıkacağı sorusu üzerine kesir dereceli matematik sistemlerinin temelleri atılmıştır. Ancak bu sorunun cevabı uzun yıllar boyunca çelişki olarak kalmıştır. Çünkü bu fonksiyondaki sistemin fiziksel anlamını kavramak ve tamsayı dereceli türev veya integratörlere göre tanımlamalarını ve hesaplamalarını yapmak o dönemde oldukça zordu. Bu durum kesir dereceli sistemlerin mühendislik alanlarında kullanımını zorlaştırmıştır [50]. Sonraki yıllarda kesir dereceli sistemlere ünlü matematikçiler katkı yapmak için çalışmalar yapmışlar, farklı yöntemler ve tanımlamalar geliştirmişlerdir. Kesir dereceli türev ve integral için geliştirilen yöntemler ve tanımlamalar çok fazla ilgi çekmiş, bu yöntemleri esas alan çok sayıda çalışma gerçekleşmiştir. Matematikçiler tarafından en çok kullanılan yöntemler çalışmayı gerçekleştiren yazarların isimlerini taşıyan Riemann-Liouville, Grünwald-Letnikov ve Caputo yöntemleridir [51].

5. 2. Tamsayı Dereceli PID ve Kesir Dereceli PID ($PI^\lambda D^\mu$) Kontrolün Karşılaştırılması

Kesir dereceli PID kontrolör, tamsayı dereceli PID' den farklı olarak kesir dereceli bir integratör ve kesir dereceli türevden oluşmaktadır. Ayrıca hem tamsayı dereceli hem de kesir dereceli sistem modelleri üzerinde uygulanabilir. Tamsayı dereceli PID kontrolör K_p , K_i ve K_d parametrelerinden oluşurken, kesir dereceli $PI^\lambda D^\mu$ 'de parametre sayısı (K_p , K_i , K_d , λ , μ) beş tanedir. $PI^\lambda D^\mu$ kontrolör beş farklı ayarlama parametresine sahip olduğundan dolayı kapalı çevrim kontrol sistemlerinin kontrolünde daha esnek bir çalışma aralığı sağlamaktadır [52].



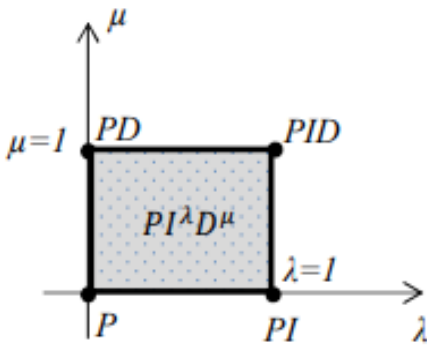
Şekil 5.1. $PI^\lambda D^\mu$ kontrolörün blok diyagramı

Şekil 5.1’de blok diyagramı verilen kesir dereceli PID kontrolörün zaman domenindeki karşılığı denklem (5.1), Laplace dönüşümünün karşılığı ise denklem (5.2) ile verilmektedir [53].

$$U(t) = K_p e(t) + K_i D^{-\lambda} e(t) + K_d D^\mu e(t) \quad (5.1)$$

$$C_f(s) = \frac{U(s)}{E(s)} = K_p + \frac{K_i}{s^\lambda} + K_d s^\mu \quad (5.2)$$

$PI^\lambda D^\mu$ kontrolörde λ ve μ parametrelerinin değerine göre denetleme tipi Şekil 5.2’de gösterilmiştir. $\lambda = 0$, $\mu = 0$ olursa P kontrolör, $\lambda = 1$, $\mu = 0$ için PI kontrolör, $\lambda = 0$, $\mu = 1$ için PD kontrolör $\lambda = 1$, $\mu = 1$ için PID kontrolör elde edilir [52, 53].



Parametre değeri	Denetleyici tipi
$\lambda = 0$ ve $\mu = 0$	P
$\lambda = 1$ ve $\mu = 0$	PI
$\lambda = 0$ ve $\mu = 1$	PD
$\lambda = 1$ ve $\mu = 1$	PID
$0 \leq \lambda \leq 1$ ve $0 \leq \mu \leq 1$	$PI^\lambda D^\mu$

Şekil 5.2. λ ve μ parametrelerinin değişimine göre denetleyici tipleri [52, 53]

5.3. Sistemin Kesir Dereceli PID ($PI^\lambda D^\mu$) Yöntemi ile Kontrolü

Tez çalışmasının önceki bölümünde gerçekleştirilen çalışmalarda sistemin PID kontrol yöntemi kullanarak Matlab/Simulink'te modellenmesi ve gerçek sıcaklık kontrol uygulaması başarı ile gerçekleştirilmişti. Bu bölümde ise sistemin kesir dereceli PID kontrol yöntemi ile simulasyonu gerçekleştirilecektir. Ancak sistemde zaman gecikmesinin büyük olması ve kesir dereceli katsayılar içermesinden dolayı Matlab/Simulink ile sistemi modellemek oldukça zordur ve doğru sonuçlar vermemektedir. Bu sebepten kesirli dereceli sistemler için Atherton, Tan ve Yüce tarafından önerilen IFTM metodu ile bütün kesir dereceli sistemler için kontrolör performanslarını analiz edebilme, zaman cevaplarını hesaplamada ve transfer fonksiyon formları için analitik sonuçlara çok yakın zaman cevapları hesaplamada çok başarılı sonuçlar elde edilebilmektedir [54]. IFTM yönteminin geliştirilmiş olması kesir dereceli sistemlerde kontrolör tasarımı kolaylaştırmaktadır.

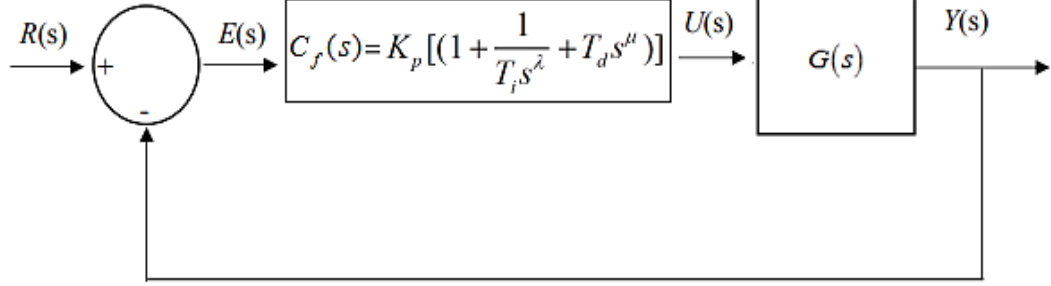
IFTM yöntemi bilinen frekans verilerinden yararlanarak zaman bilgisini hesaplamada kullanılan bir yöntemdir. IFTM metodu ters Fourier Dönüşümünü temel alan bir metottur. Kesir dereceli sistemin transfer fonksiyonu $P(s)$ 'ye göre $p(t)$ birim darbe cevabı $p(t) = L^{-1}(P(s))$ göre elde edilir ve $p(t)$ denklem (5.3) ve (5.4)'te verilmiştir. Ayrıca IFTM metodu, açık çevrim transfer fonksiyonu verilerini kullanarak, kapalı çevrim kontrol sisteminin birim basamak cevabını hesaplayacak şekilde uyarlanmıştır.

$$p(t) = \frac{2}{\pi} \int_0^{\infty} \text{Re}[P(j\omega)] \cos(\omega t) d\omega \quad (5.3)$$

$$p(t) = -\frac{2}{\pi} \int_0^{\infty} \text{Im}[P(j\omega)] \sin(\omega t) d\omega \quad (5.4)$$

Matlab ortamında geliştirilen IFTM programı, denklem (5.3) ve (5.4) için nümerik integral alma yöntemi kullanılarak $p(t)$ fonksiyonunu (birim darbe cevabı) hesaplama özelliğine sahiptir.

Sistemin kesir dereceli PID kontrolörlü kapalı çevrim kontrol sistemi Şekil 5.3'te gösterildiği gibidir. Sistemde $R(s)$ giriş, $Y(s)$ çıkış, $C_f(s)$ kesirli PID kontrolörü ve $G(s)$ ısıtıcı modelini temsil etmektedir.



Şekil 5.3. Kesir dereceli PID kontrolör kapalı çevrim diyagramı

Bölüm 4'te daha önce PID kontrolörün transfer fonksiyonunun kontrolör parametreleri (4.4) ve (4.5) numaralı denklemleri ile elde etmişti.

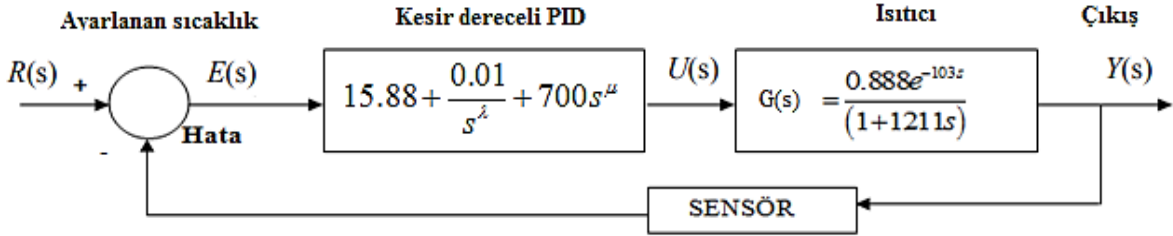
Bu elde edilen verilerden ve denklem (5.2)'den yola çıkarak sistemin kesir dereceli denklemleri elde edilmiş ve denklem (5.7)'te gösterilmiştir.

$$\frac{U(s)}{E(s)} = C_f(s) = K_p \left[1 + \frac{1}{T_i s^\lambda} + T_d s^\mu \right] \quad (5.5)$$

$$\frac{U(s)}{E(s)} = C_f(s) = 15.88 \left(1 + \frac{1}{206 s^\lambda} + 51.5 s^\mu \right) \quad (5.6)$$

$$\frac{U(s)}{E(s)} = C_f(s) = 15.88 + \frac{0.01}{s^\lambda} + 700 s^\mu \quad (5.7)$$

Elde edilen denklemler Şekil 5.3'te gösterilen diyagrama göre kesir dereceli $PI^\lambda D^\mu$ kontrolörün kapalı çevrim sisteminin blok diyagramı Şekil 5.4'teki gibi elde edilir.



Şekil 5.4. Kesir dereceli $PI^\lambda D^\mu$ kontrolör kapalı çevrim sisteminin blok diyagramı

Sistemin açık çevrim transfer fonksiyonu denklem (5.10)'da gösterildiği gibi elde edilir.

$$L(s) = C_f(s) \times G(s) \quad (5.8)$$

$$L(s) = \left(15.88 + \frac{0.01}{s^\lambda} + 700s^\mu\right) \times \frac{0.888}{(1+1211s)} e^{-103s} \quad (5.9)$$

$$L(s) = \frac{621.6s^{(\lambda+\mu)} + 14.101s^\lambda + 0.00888}{1211s^{(\lambda+1)} + s^\lambda} e^{-103s} \quad (5.10)$$

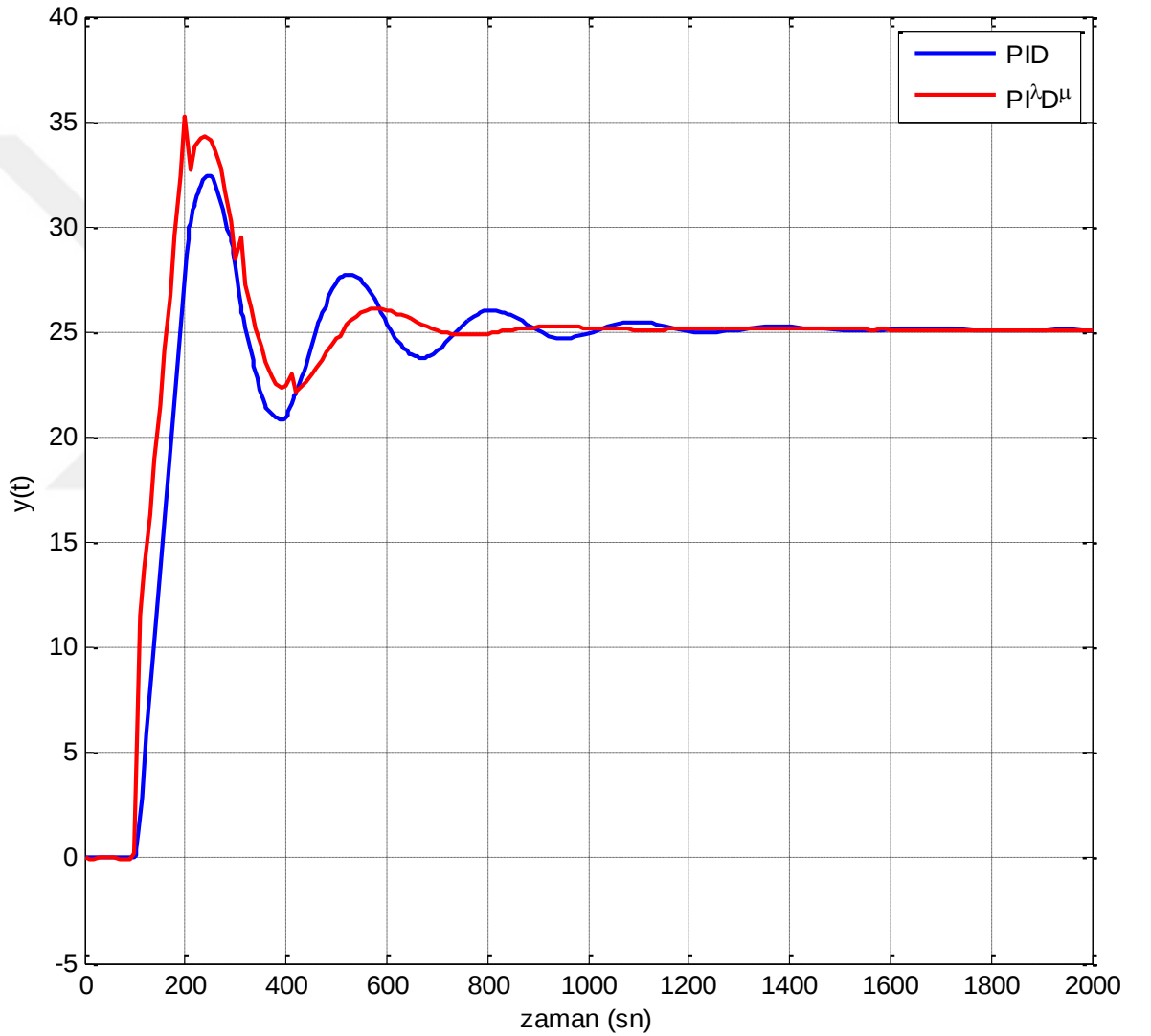
λ ve μ 'nün değiştirilmesi ile farklı kesirli dereceler için sistem performansları zaman cevapları çizdirilerek incelenecektir. $\lambda = 1.05$ ve $\mu = 1.1$ değerleri seçilerek elde edilen açık çevrim cevabı denklem (5.11)'de gösterilmiştir.

$$L(s) = \frac{621.6s^{2.15} + 14.101s^{1.05} + 0.00888}{1211s^{2.05} + s^{1.05}} e^{-103s} \quad (5.11)$$

Denklem (5.11)'de elde edilen değerler Matlab ortamında geliştirilen IFTM programında yerine yazılarak sistemin performansı elde edilmiştir [50]. IFTM metodu kullanarak elde edilen kesir dereceli $PI^\lambda D^\mu$ kontrolörün performans sonuçları ile önceki bölümde elde edilen tamsayı dereceli PID kontrolörün Matlab/Simulink programındaki simulasyon sonuçları Şekil 5.5'te gösterilmiştir.

Şekil 5.5'te iki yöntem karşılaştırıldığında 25 santigrad dereceye ayarlanan sıcaklık değerine, tamsayı dereceli PID kontrol sisteminde 880. saniyede yüzde 2'lik aşım ile yerleşmeye başlarken kesir dereceli $PI^\lambda D^\mu$ kontrol sisteminde bu yerleşme süresi 660. saniye olarak elde edilmiştir.

Bu sonuçlara göre kesir dereceli $PI^\lambda D^\mu$ kontrol yönteminin tamsayı dereceli PID kontrol yöntemine göre yerleşme zamanı açısından daha başarılı sonuçlar verdiği görülmektedir.



Şekil 5.5. Kesir dereceli $PI^\lambda D^\mu$ kontrol yöntemi elde edilen sonuçların tamsayı dereceli PID kontrolör yöntem ile karşılaştırılması

6. SONUÇLAR

Bu tez çalışmasında LabVIEW ve Raspberry Pi 3 kullanarak PID kontrolör ile gerçek zamanlı bir sıcaklık ölçüm ve kontrol uygulaması üzerine bir çalışma yapılmıştır.

Gerçekleştirilen sistem Matlab/Simulink ile modellenmiş ve elde edilen sonuçlara göre gerçek zamanlı bir sıcaklık kontrol uygulaması da başarıyla gerçekleştirilmiştir.

Çalışmanın simülasyon aşamasında Ziegler-Nichols metoduyla tespit edilen PID katsayılarının yakın çevresinde yapılan taramalar ile daha iyi performans verebilen PID kontrolörler elde edebilmek mümkün olmuştur.

Gerçek zamanlı kontrol sistemi üzerinde Raspberry Pi 3 ile PID algoritması başarılı bir şekilde uygulanmıştır. Gerçek zamanlı elde edilen sonuçlar simülasyon sonuçlarının paralelinde ve kontrol performansı kabul edilebilir düzeydedir.

Yapılan çalışmanın öne çıkan avantajlarından biri Raspberry Pi 3 ve LabVIEW programının Wi-Fi bağlantısı ile kablosuz olarak haberleşebilmesidir. Bu özellik sayesinde sıcaklık ve kontrol bilgilerinin uzaktan izlenebilir olmasıdır.

Bir diğer önemli avantajı ise LabVIEW programının grafiksel kullanıcı arayüzüne sahip olması ve interaktif özelliği sayesinde uygulamaya geliştirilen kullanıcı arayüzü üzerinden kablosuz olarak müdahale edilebilmektedir.

Çalışmanın devamında daha başarılı kontrolör tasarımı yapabilmek için son zamanlarda çok popüler olan kesir dereceli $PI^\lambda D^\mu$ kontrol yöntemi bu sisteme uygulanmıştır. Matlab üzerinde geliştirilmiş olan IFTM programı kullanarak sistemin zaman cevabı elde edilmiştir.

Tamsayı dereceli PID kontrol yöntemi ile elde edilen simülasyon sonuçları ile kesir dereceli $PI^\lambda D^\mu$ kontrol yöntemi karşılaştırılmıştır. Karşılaştırma sonucunda kesir dereceli $PI^\lambda D^\mu$ kontrolörün tamsayı dereceli PID kontrolöre göre yerleşme zamanı açısından daha başarılı sonuçlar verdiği elde edilen sonuçlara göre belirlenmiştir.

Bütün bu sonuçlar dikkate alındığında geliştirilen sıcaklık kontrol sisteminin endüstride hassas sıcaklık kontrolü gerektiren alanlarda başarılı olacağı düşünülmektedir.

Ayrıca geliştirilen bu uygulama eğitim materyali olarak laboratuvar uygulamalarında da kullanılabilir.

Gelecek çalışmalarda kesir dereceli PID kontrol yöntemi ile sistemin gerçek zamanlı uygulamasının yapılması planlanmaktadır.

7. KAYNAKLAR

- [1] N. T. Artuğ. *LabVIEW ile mikrodnetleyicili bir endüstriyel otomatik sıcaklık kontrol sistemi*, Yüksek Lisans Tezi, Yıldız Teknik Üniversitesi Türkiye, 2010.
- [2] T. Uzun. *Endüstriyel Otomatik Kontrol Sistemleri Ders Notu*, Yıldız Teknik Üniversitesi Türkiye, 2009.
- [3] M. Kapıdere, R. Ahıska ve Ş. Güler, Dört Sıcaklık Algılayıcı ve Mikrodnetleyicili Termohipoterm Sistemi, 3. International Advanced Technologies Symposium, (2003), Ankara, pp. 90-96.
- [4] S. Aksoy, S. ve A. Mühürücü, Mikrodnetleyici tabanlı bir süt soğutma tankının soğutma kontrolü, Sakarya Üniversitesi yayını, 2006.
- [5] S. Şahin, Y. İşler and M.B. Selek, *Sanal Aygıtlarla Sıvı Seviyesi ve Sıcaklık Kontrolü Deneyine Örnek Bir Uygulama*, **Celal Bayar Üniversitesi Soma Meslek Yüksekokulu Teknik Bilimler Dergisi**, 8:1, (2007), 1-10.
- [6] M. Çelik. *Sıcaklık rejimli seramik pişirme fırınının bilgisayarlı kontrolü*, Yüksek Lisans Tezi Türkiye, 2006.
- [7] E. Berber. *Mikrodnetleyicili endüstriyel otomatik sıcaklık ölçüm ve kontrol sistemi*, Yüksek Lisans Tezi, Yıldız Teknik Üniversitesi Türkiye, 2008.
- [8] A. Yüce. *LabVIEW ile endüstriyel sıcaklık ölçme ve kontrol sistemi*, Yüksek Lisans Tezi, Yıldız Teknik Üniversitesi Türkiye, 2011.
- [9] M.P. Groover, *Automation, Production Systems, and Computer Integrated Manufacturing*, 5th edition, Pearson, 2018, 107-120.
- [10] Anonim.(2012).<https://teknokoliker.com/2012/10/elektronik-sensorlerin-dogru-secilmesi.html> (Online ulaşma tarihi 10 Şubat 2019)
- [11] E.A.PAR, *Endüstriyel Kontrol El Kitabı Cilt 1 Transdüserler*, MEB, İstanbul, 2002.
- [12] B.E.NOLTINGK, *Sıcaklık ve Kimyasal Bileşenlerin Ölçümü*, MEB, Eskişehir, 1994.
- [13] Anonim.(b.t) <https://www.electronics-tutorials.ws/> (Online ulaşma tarihi 19 Şubat 2019)
- [14] T. Kuglestadt, *Semiconductor Temperature Sensors Challenge Precision RTDs and Thermistors in Building Automation, Application Report*, SNAA267–April 2015
- [15] C.B. Kuo, *Automatic Control Systems*, Third edition, Prentice Hall, New Jersey, 1975.
- [16] K. Warwick, *An Introduction To Control Systems*, 2nd Edition, World Scientific, Singapore, 1996.
- [17] Anonim. (b.t.). <http://www.yildiz.edu.tr/~okincay/dersnotu/OtoKontIBol.pdf> (Online ulaşma tarihi 20 Şubat 2019)

- [18] M. Poyraz, *PID Kontrol Proje 1*, Kocaeli Üniversitesi Türkiye, 2006.
- [19] E. Özkan. *Kontrol sistemlerinin modellenmesi ve PID kontrolörü*, Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi Türkiye, 2006.
- [20] A. Güllü. *Kontrolör ara yüzü tasarımı*, Yüksek Lisans Tezi, Marmara Üniversitesi Türkiye, 2010.
- [21] A. Aldemir ve H. Hapoğlu, *Kablosuz Sıcaklık Kontrolü İçin PID Ayarlama Yöntemlerinin Karşılaştırılması*, **Politeknik Dergisi**, 19:1, (2016), 9-19.
- [22] İ. Coşkun ve H. Terzioğlu, “Hız Performans Eğrisi kullanarak Kazanç(PID) Parametrelerinin belirlenmesi, **Teknik-online Dergi**, 6:3, (2007),180-206.
- [23] G. Gidemen ve M. Furat, *PID Parametrelerinin Ayarlama Yöntemleri:2. Derece Sistem Modeline Uygulanması ve Karşılaştırmalı Olarak Değerlendirilmesi*, **Çukurova Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi**, 30:2, (2015), 355- 362.
- [24] K. Ogata, *Modern Control Engineering*, 3rd Edition, Prentice Hall, New Jersey, 1997, 669-677.
- [25] İ. Doğan, *Microcontroller Based Temperature Monitoring and Control*, Elsevier Science & Technology Books, 2002, 201-219.
- [26] Anonim. (2018). <https://learn.adafruit.com/category/learn-raspberry-pi> (Online ulaşma tarihi 25 Şubat 2019)
- [27] D. Kaya, *Raspberry Pi ile Sera İklimlendirme Proje çalışması*, Bilecik Şeyh Edebali Üniversitesi Türkiye, 2017.
- [28] T. Charleston, *Raspberry PI 3: Enchanted Guide for Starters*, Kindle Edition, 2017
- [29] F. Şişik. *Raspberry Pi üzerinde Çalışan ELM algoritması ile beyin tümör segmentasyonu*, Yüksek Lisans Tezi, Kahramanmaraş Sütçü İmam Üniversitesi Türkiye, 2018.
- [30] E. Upton and G. Halfacree, *Raspberry Pi User Guide*, 4th Edition, 2016.
- [31] M. Demir ve B. Güneş. *Zagi tipi bir insansız hava aracının (İHA) tasarımı ve bilgisayar görmesine dayalı olarak kontrol yazılımının geliştirilmesi*, Lisans Bitirme Tezi, Sakarya Üniversitesi Türkiye, 2017.
- [32] F. Özgül, *Her Yönüyle Python*, 6. Baskı, 2013, 4-7.
- [33] Li, J., Li, L., “Comparative Research on Python Speed Optimization Strategies”, School of Electronic Engineering, Guilin University of Electronic and Technology, Guangxi (2009).

- [34] H. Parmaksız. *Raspberry Pi ile fotovoltaik panellerin elektriksel verilerinin gerçek zamanlı izlenmesi*, Yüksek Lisans Tezi, Bilecik Şeyh Edebali Üniversitesi Türkiye, 2017.
- [35] B. Malkoç, *Temel Bilimler ve Mühendislik Eğitiminde Programlama Dili Olarak Python*, 12. Akademik Bilişim Konferansı, (2012), Uşak Üniversitesi.
- [36] F. Georgatos. *How applicable is Python as first computer language for teaching programming in a pre-university educational environment, from a teacher's point of view?*, Masters Thesis, AMSTEL Institute Faculty of Science Universiteit van Amsterdam Holland, 2002.
- [37] Z. Şentürk ve K.R. Cebeci. *LabVIEW denetimli peltierli ortam soğutucusu*, Lisans Bitirme Çalışması, Karadeniz Teknik Üniversitesi Türkiye, 2011.
- [38] S.M Schulstock. *Data Cloud Platform for Data Management, Logging, Control and Monitoring*, Master's Thesis, University College of Southeast Norway, 2016.
- [39] Anonim.(2014).<http://www.ni.com/download/labview-development-system-2014/4735/en/> (Online ulaşma tarihi 20 Mart 2019)
- [40] Anonim. (2019). <https://vipm.jki.net/get> (Online ulaşma tarihi 1 Nisan 2019)
- [41] Anonim. (2016). <https://www.labviewmakerhub.com/forums/viewtopic.php?f=12&t=2062> (Online ulaşma tarihi 3 Nisan 2019)
- [42] D. Murray, *Install Guide: RPi, LabVIEW and MakerHub*,2016.
- [43] Anonim.(2019).<https://store.open-electronics.org/%20PWM-MOS-Driver-15A-400W-assembled-3085-MOSDRIVER> (Online ulaşma tarihi 5 Nisan 2019)
- [44] Anonim.(b.t.).<https://tutorials-raspberrypi.com/raspberry-pi-temperature-sensor-1wire-ds18b20/> (Online ulaşma tarihi 10 Nisan2019)
- [45] A. Yaldaie. *Home automation and security system with the Raspberry Pi*, Bachelor Thesis Helsinki Haaga-Helia University of Applied Sciences Finland, 2016.
- [46] Anonim. (2016). <http://www.circuitbasics.com/raspberry-pi-ds18b20-temperature-sensor-tutorial/> (Online ulaşma tarihi 12 Nisan 2019)
- [47] Vesa-Matti Yli-Heikkilä. *Home surveillance with Raspberry Pi*, Bachelor thesis, Seinäjoki University Of Applied Sciences Finland, 2015.
- [48] Anonim.(2013).<https://learn.adafruit.com/adafruits-raspberry-pi-lesson-11-ds18b20-temperature-sensing?view=all> (Online ulaşma tarihi 15 Nisan 2019)
- [49] T. Doğruer, A. Yüce ve N. Tan, *Zaman Gecikmesine Sahip Kesir Dereceli Bir Kontrol Sisteminde Optimizasyon Metodu Kullanılarak PID Kontrolör Tasarımı*, **Gaziosmanpaşa Bilimsel Araştırma Dergisi 6 (ISMSIT2017)**, 30-39, (2017).

- [50] A. Yüce. *Kesir dereceli kontrol sistemlerinde zaman cevaplarının hesaplanması ve kontrol uygulamaları*, Doktora Tezi, Malatya İnönü Üniversitesi Türkiye, 2018.
- [51] N.F. Deniz. *Kesir dereceli sistemlerde modelleme ve kontrol uygulamaları*, Doktora Tezi, Malatya İnönü Üniversitesi Türkiye, 2017.
- [52] M.M. Özyetkin. *Kesirli dereceli kontrol sistemlerinin dayanıklı analiz ve tasarımı*, Doktora Tezi, Malatya İnönü Üniversitesi Türkiye, 2013.
- [53] H. Saribaş. *Dört rotorlu bir insansız hava aracının kesir dereceli denetleyici ile kontrolü*, Yüksek Lisans Tezi, Eskişehir Anadolu Üniversitesi Türkiye, 2015.
- [54] D.P. Atherton, N. Tan and A.Yüce, *Methods for computing the time response of fractional-order systems*, **IET Control Theory and Applications**, 9,(2015), 817-830.



8. EKLER

Ek 1 Raspberry Pi 3 ile Sıcaklık Ölçümü Yazılımı

Ek 2 Raspberry Pi 3 ile Sıcaklık Kontrol Sisteminin Yazılımı



Ek 1 Raspberry Pi 3 ile Sıcaklık Ölçümü Yazılımı

```
import os
import glob
import time

os.system('modprobe w1-gpio')
os.system('modprobe w1-therm')

base_dir = '/sys/bus/w1/devices/'
device_folder = glob.glob(base_dir + '28*')[0]
device_file = device_folder + '/w1_slave'

def read_temp_raw():
    f = open(device_file, 'r')
    lines = f.readlines()
    f.close()
    return lines

def read_temp():
    lines = read_temp_raw()
    while lines[0].strip()[-3:] != 'YES':
        time.sleep(0.2)
        lines = read_temp_raw()
    equals_pos = lines[1].find('t=')
    if equals_pos != -1:
        temp_string = lines[1][equals_pos+2:]
        temp_c = float(temp_string) / 1000.0
        return temp_c

while True:
    print(read_temp())
    time.sleep(1)
```

Ek 2 Raspberry Pi 3 ile Sıcaklık Kontrol Sisteminin Yazılımı

```
import os
import glob
import time
import RPi.GPIO as GPIO

GPIO.setmode(GPIO.BCM)
ısıtıcının_pini = 18
base_dir = '/sys/bus/w1/devices/'
device_folder = glob.glob(base_dir + '28*')[0]
device_file = device_folder + 'w1_slave'

GPIO.setup(ısıtıcının_pini, GPIO.OUT)
ısıtıcının_pwm_degeri = GPIO.PWM(ısıtıcının_pini, 100)
ısıtıcının_pwm_degeri.start(0)

ilk_hata_degeri= 0
ilk_zaman_degeri = 0
olcülen_sıcaklık_degeri = 0
p_degeri = 0
i_degeri = 0
d_degeri = 0

def read_temp_raw():
    f = open(device_file, 'r')
    lines = f.readlines()
    f.close()
    return lines

def read_temp():
    lines = read_temp_raw()
    while lines[0].strip()[-3:] != 'YES':
        time.sleep(0.2)
        lines = read_temp_raw()
    equals_pos = lines[1].find('t=')
    if equals_pos != -1:
        temp_string = lines[1][equals_pos+2:]
        temp_c = float(temp_string) / 1000.0
        return temp_c

def constrain(deger, minimum, maximum):
    if deger < minimum:
        return minimum
    if deger > maximum:
        return maximum
    else:
        return deger

def pid_kurulum():
    global ilk_zaman_degeri, ilk_hata_degeri, olcülen_sıcaklık_degeri, ayarlanan_sıcaklık_degeri
    global p_degeri, i_degeri, d_degeri
    zaman = time.time()
    dt = zaman - ilk_zaman
    hata = ayarlanan_sıcaklık_degeri - olcülen_sıcaklık_degeri
```

```

de = hata - ilk_hata

p_degeri= kp * hata
i_degeri += ki * hata
i_degeri = constrain(i_degeri, 0, 100)
d_degeri = (de / dt) * kd

ilk_hata_degeri = hata
cikis= p_degeri + i_degeri + d_degeri
cikis = constrain(cikis, 0, 100)
return cikis

def pid2_kurulum():
    global ilk_zaman_degeri, ilk_hata_degeri, olculen_sicaklik_degeri, ayarlanan_sicaklik_degeri
    global p_degeri, i_degeri, d_degeri
    zaman = time.time()
    dt = zaman - ilk_zaman
    hata = ayarlanan_sicaklik_degeri - olculen_sicaklik_degeri
    de = hata - ilk_hata

    p_degeri= kp * hata
    i_degeri += ki * hata
    i_degeri = constrain(i_degeri, 0, 100)
    d_degeri = (de / dt) * kd

    ilk_hata_degeri = hata
    cikis= p_degeri + i_degeri + d_degeri
    cikis = constrain(cikis, 0, 100)
    return cikis

ayarlanan_sicaklik_degeri = giris('ayarlanan sicaklik degerini giriniz ')
kp = giris('kp degerini giriniz: ')
ki = giris('ki degerini giriniz: ')
kd = giris('kd degerini giriniz: ')

ilk_zaman_degeri = time.time()
try:
    while True:
        simdiki_zaman_degeri = time.time()
        if simdiki_zaman_degeri > ilk_zaman_degeri + 1 :
            ilk_zaman_degeri = simdiki_zaman_degeri
            olculen_sicaklik_degeri = read_temp()
            gorev = pid_kurma()
            isiticinin_pwm_degeri.ChangeDutyCycle(gorev)

            print(str(olculen_sicaklik_degeri) + ', ' + str(ayarlanan_sicaklik_degeri) + ', ' +
str(gorev2))

finally:
    GPIO.cleanup()

```

ÖZGEÇMİŞ

Ad Soyad: Ömer Faruk ÖZCAN

Doğum Yeri ve Tarihi: Elbistan - 28.08.1990

Adres: Yeşilyurt Mah. Zeybek sok. No:3 Daire 6 Kahramanmaraş/Elbistan

E-Posta: faruk_omer46@yahoo.com

Lisans: İstanbul Arel Üniversitesi- Mühendislik ve Mimarlık Fakültesi –
Elektrik- Elektronik Mühendisliği- 2013

Mesleki Deneyim: 2013-2014 Afşin-Elbistan B Termik Santrali-Elektrik Bakım Müh.
2015- Elbistan Belediyesi- Elektrik-Elektronik Müh.

YAYIN LİSTESİ:

Uluslararası Bilimsel Toplantılarda Sunulan ve Bildiri Kitabında Basılan Bildiriler:

1. **Özcan, Ö.F.**, Yüce, A. ve Tan, N. "LabVIEW ve Raspberry Pi 3 Kullanarak Gerçek Zamanlı Sıcaklık Ölçüm Sistemi", 3rd International Mediterranean Science and Engineering Congress (IMSEC 2018), Adana/Turkey, ss.930-937, 24-26 October 2018.
2. **Özcan, Ö.F.**, Yüce, A. ve Tan, N. "LabVIEW ve Raspberry Pi 3 Kullanarak Gerçek Zamanlı Sıcaklık Kontrol Uygulaması", 3rd International Science and Academic Congress (INSAC 2019), Konya/Turkey, volume 2 ss.87-100, 19-20 April 2019.

