

**ANKARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

YÜKSEK LİSANS TEZİ

**SERALAR İÇİN KABLOSUZ SENSÖR AĞI VE WEB TABANLI SICAKLIK
İZLEME SİSTEMİNİN GELİŞTİRİLMESİ**

Furkan UĞUR

**TARIM MAKİNALARI VE TEKNOLOJİLERİ MÜHENDİSLİĞİ
ANABİLİM DALI**

ANKARA

2016

Her hakkı saklıdır.

TEZ ONAYI

Furkan UĞUR tarafından hazırlanan “**Seralar İçin Kablosuz Sensör Ağı ve Web Tabanlı Sıcaklık İzleme Sisteminin Geliştirilmesi**” adlı tez çalışması 13/01/2016 tarihinde aşağıdaki jüri tarafından oy birliği ile Ankara Üniversitesi Fen Bilimleri Enstitüsü Tarım Makinaları ve Teknolojileri Mühendisliği Anabilim Dalı’nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Danışman : Yrd. Doç. Dr. Mehmet Ali DAYIOĞLU

Jüri Üyeleri :

Başkan : Doç. Dr. Ercan Nurcan YILMAZ
Gazi Üniversitesi Fen Bilimleri Enstitüsü
Elektrik-Elektronik Mühendisliği A.B.D.

Üye : Yrd. Doç. Dr. Mehmet Ali DAYIOĞLU
Ankara Üniversitesi Fen Bilimleri Enstitüsü
Tarım Makinaları ve Teknolojileri Mühendisliği A.B.D.

Üye : Yrd. Doç. Dr. Caner KOÇ
Ankara Üniversitesi Fen Bilimleri Enstitüsü
Tarım Makinaları ve Teknolojileri Mühendisliği A.B.D.

Yukarıdaki sonucu onaylarım.

Prof. Dr. İbrahim DEMİR
Enstitü Müdürü

ETİK

Ankara Üniversitesi Fen Bilimleri Enstitüsü seminer yazım kurallarına uygun olarak hazırladığım bu seminer içindeki bütün bilgilerin doğru ve tam olduğunu, bilgilerin üretilmesi aşamasında bilimsel etiğe uygun davrandığımı, yararlandığım bütün kaynakları atıf yaparak belirttiğimi beyan ederim.

13/01/2016

Furkan UĞUR

ÖZET

Yüksek Lisans Tezi

SERALAR İÇİN KABLOSUZ SENSÖR AĞI VE WEB TABANLI SICAKLIK İZLEME SİSTEMİNİN GELİŞTİRİLMESİ

Furkan UĞUR

Ankara Üniversitesi
Fen Bilimleri Enstitüsü
Tarım Makinaları ve Teknolojileri Mühendisliği Anabilim Dalı

Danışman: Yrd. Doç. Dr. Mehmet Ali DAYIOĞLU

Bu yüksek lisans tez çalışmasında, seralar için ARM-ağ tabanlı kablosuz sıcaklık ölçüm sistemi geliştirilmiştir. Sistem iki kablosuz sensör düğümü, kablosuz ağ geçidi ve istemcilerden oluşmaktadır. Her kablosuz sensör düğümünde Raspberry Pi 2 ARM tabanlı tek kart bilgisayar, WiFi adaptörü ve ikişer adet DS18B20 sıcaklık sensörü bulunmaktadır. Sistemdeki kablosuz düğümler gerçek zamanlı olarak sıcaklıkları ölçme, verileri kaydetme, ana düğüme veri aktarma ve takip etme işlemlerini yapacak şekilde programlanmıştır. Sistem maliyetinin azaltılması ve sera çevresinde esnek çözüm sağlanması amacıyla kablosuz sensör düğümlerinden birisi sunucu olarak kullanılmıştır. Sistemin ağ ortamında çalışabilmesi için ARM tabanlı donanıma özel servis ve ara yüz yazılımları geliştirilmiştir.

Geliştirilen prototip sistem laboratuvar ve sera koşullarında test edilmiştir. Sera içinde farklı noktalardaki hava, toprak ve su sıcaklıkları ölçülmüş, ortalama sıcaklık seviyeleri sırasıyla 7,63 °C, toprak sıcaklığı 2,26 °C ve 1,31 °C olarak hesaplanmıştır.

Ocak 2016, 74 sayfa

Anahtar Kelimeler: ARM, İnternet, Sensör, Sera

ABSTRACT

Master Thesis

DEVELOPMENT OF WIRELESS SENSOR NETWORK AND WEB-BASED TEMPERATURE MONITORING SYSTEM FOR GREENHOUSES

Furkan UĞUR

Ankara University
Graduate School of Natural and Applied Sciences
Agricultural Machinery and Technologies Engineering

Supervisor: Asst. Prof. Dr. Mehmet Ali DAYIOĞLU

In this master thesis, an ARM-Web-based wireless temperature measurement system for greenhouses has been developed. The system contains two wireless sensor nodes, a wireless gateway and clients. There are the components including a Raspberry Pi 2 ARM computer, a Wi-Fi adapter, two DS18B20 temperature sensors at each wireless sensor node. Wireless sensor nodes were programmed to make real-time temperatures measuring, data recording, data transfer to main node and system monitoring. One of wireless sensor nodes was used as server for reducing the system cost and providing flexible at greenhouse environment. The service and interface software were developed to be operate at network environment of system for specific ARM based hardware.

The developed prototype system was experimented under laboratory and greenhouse conditions. The air, soil and water temperatures at different points greenhouse were measured, and average temperature levels were calculated as 7.63 °C, 2.26 °C and 1.31 °C, respectively.

January 2016, 74 pages

Key Words: ARM, Internet, Sensor, Greenhouse

TEŐEKKÜR

Çalıřmalarımı yönlendiren, bilgi ve birikimi ile bana yol gösteren danıřman hocam Sayın Yrd. Doç. Dr. Mehmet Ali Dayıođlu'na, destekleri ve deneyimlerini paylařtıđı için eřim Melek Uđur bařta olmak üzere tüm aileme ve bu çalıřmanın hazırlanması sırasında hořgörülerinden dolayı T.C. Gıda Tarım ve Hayvancılık Bakanlıđı Eđitim Yayım ve Yayınlar Dairesi Bařkanlıđı'nda çalıřan tüm mesai arkadařlarım ve yöneticilerime teőekkürü bir borç bilirim.

Furkan UĐUR
Ankara, Ocak 2016

İÇİNDEKİLER

TEZ ONAY SAYFASI	
ETİK	i
ÖZET	ii
ABSTRACT	iii
TEŞEKKÜR	iv
KISALTMALAR DİZİNİ	vii
ŞEKİLLER DİZİNİ	viii
ÇİZELGELER DİZİNİ	ix
1. GİRİŞ	1
1.1 Sera	1
1.2 Türkiye’de Sera Sektörü	2
1.3 Sera Teknolojisi	5
2. KAYNAK ÖZETLERİ	9
3. KURAMSAL TEMELLER	14
3.1 TCP/IP ve HTTP	14
3.2 JSON	16
3.3 WSN	17
3.4 1-Wire	18
3.5 IoT	18
4.MATERYAL VE YÖNTEM	20
4.1 Sistem Bileşenleri	20
4.1.1 Kablosuz sensör düğümü	20
4.1.2 Kablosuz ağ geçidi	22
4.1.3 Sunucu	22
4.1.4 İstemciler	22
4.1.5 Yazılımlar	23
4.1.6 Deney serası	24
4.2 Sistem Mimarisi	24
4.3 Kablosuz Sensör Düğümünün Tasarımı	26
4.4 DS18B20 - Raspberry Pi bağlantısı	28
4.5 Sıcaklık Bilgisinin Okunması	29
4.6 Linux Servisi	31
4.7 Ağ Servisinin Yazılması	33
4.8 Kullanıcı Arayüzü Tasarımı	35
5. ARAŞTIRMA BULGULARI	37
5.1 Sistem Maliyeti	37
5.2 Prototip Sistemin Test Bulguları	37
5.2.1 Laboratuvar denemesi	38
5.2.2 Sera denemesi	48
6. TARTIŞMA VE SONUÇ	52
KAYNAKLAR	55
EKLER	59
EK 1 ReadSensors Uygulaması Kaynak Kodları	60
EK 2 Linux Servisinin Kaynak Kodları	63
EK 3 config.ini Dosyasının İçeriği	69

EK 4 rth Dosyasının İeriđi	70
EK 5 Ađ Servisinin Kaynak Kodları	72
ÖZGEÇMİŞ.....	74



KISALTMALAR DİZİNİ

ARM	Acorn RISC Machine
CAN	Controller Area Network
CGI	Ortak ağ geçidi arayüzü (Common Gateway Interface)
CSS	Basamaklı stil şablonu (Cascading Style Sheets)
EC	Electrical conductivity
Gnd	Toprak hattı (Ground)
GPIO	Genel amaçlı giriş/çıkış (General-Purpose Input/Output)
GPRS	General Packet Radio System
GPS	Küresel konumlandırma sistemi (Global Positioning System)
GSM	Global System for Mobile
HTML	Hipermetin işaretleme dili (Hypertext Markup Language)
SBC	Tek kartta bilgisayar (Single Board Computer)
SoC	Tek yongada sistem (System of Chip)
USB	Evrensel seri veriyolu (Universal Serial Bus)
Vcc	Entegre güç kaynağı girişi (IC Power-supply Pin)
Wi-Fi	Wireless Fidelity
WSN	Kablosuz sensör ağı (Wireless Sensor Network)

ŞEKİLLER DİZİNİ

Şekil 1.1 Örtü altı tarım alanlarındaki değişim	3
Şekil 1.2 Türkiye’de örtü altı yetiştiricilik sistemleri dağılımı	4
Şekil 1.3 Yıllara göre üretilen toplam örtü altı ürün miktarı	5
Şekil 3.1 TCP/IP soyutlama katmanlarının şematik görüntüsü	15
Şekil 3.2 Örnek JSON verisi	17
Şekil 4.1 Raspberry Pi 2 Model B.....	21
Şekil 4.2.a USB Wi-fi adaptörleri, b. MicroSD kart, c. DS18B20 sayısal sıcaklık sensörü, d. Su geçirmez DS18B20.....	21
Şekil 4.3 Sistemin test edildiği sera	24
Şekil 4.4 Sistem katmanları.....	25
Şekil 4.5 Düğüm cihazının genel mimarisi	26
Şekil 4.6 Tek sensör düğümü kullanım diagramı	27
Şekil 4.7 Birden çok sensör düğümü kullanım diagramı	28
Şekil 4.8 DS18B20 sensörünün bağlanması	29
Şekil 4.9 Örnek w1_slave dosyası içeriği	30
Şekil 4.10 Sensörlerden veri okuma uygulaması akış diyagramı	31
Şekil 4.11 a. Çekirdek modülü yükleme, b. Servis akış diyagramı	32
Şekil 4.12 Ağ servisinin akış diagramı	34
Şekil 4.13 Kullanıcı arayüzü genel bilgi sayfası	35
Şekil 5.1 Master düğümünün ad ve çalışma modu ayarı	39
Şekil 5.2 Master düğümünün api anahtarı ayarı.....	39
Şekil 5.3 Slave düğümünün ad ve çalışma modu ayarı.....	40
Şekil 5.4 Slave düğümü remote ayarları	41
Şekil 5.5 Düğümlerin sensör ayarları.....	41
Şekil 5.6 MasterPi ve SlavePi düğümleri.....	42
Şekil 5.7 Laboratuvar testi sırasında sıcaklık ölçümlerinin izlenmesi.....	42
Şekil 5.8 Master düğümünde rapor sihirbazının kullanılması	43
Şekil 5.9 MasterPi düğümü 3.12.2015 tarihli sıcaklık grafiği	44
Şekil 5.10 SlavePi düğümü 3.12.2015 tarihli sıcaklık grafiği.....	45
Şekil 5.11 MasterPi düğümü 4.12.2015 tarihli sıcaklık grafiği	45
Şekil 5.12 SlavePi düğümü 4.12.2015 tarihli sıcaklık grafiği.....	46
Şekil 5.13 MasterPi düğümü 5.12.2015 tarihli sıcaklık grafiği	46
Şekil 5.14 SlavePi düğümü 5.12.2015 tarihli sıcaklık grafiği.....	47
Şekil 5.15 MasterPi düğümü 6.12.2015 tarihli sıcaklık grafiği	47
Şekil 5.16 SlavePi düğümü 6.12.2015 tarihli sıcaklık grafiği.....	47
Şekil 5.17 SeraPi düğümünün adı ve çalışma modu ayarı	48
Şekil 5.18 SeraPi sensor ayarları.....	49
Şekil 5.19 Denemenin yapıldığı sera	49
Şekil 5.20 Serada kablosuz sensör düğümünün kurulumu	50
Şekil 5.21 SeraPi düğümü sıcaklık grafiği	50

ÇİZELGELER DİZİNİ

Çizelge 1.1 Türkiye’de örtü altı yetiştiriciliğın yoğun olduđu iller	4
Çizelge 4.1 Raspberry Pi 2 Model B'nin teknik özellikleri.....	21
Çizelge 4.2 DS18B20 teknik özellikleri.....	22
Çizelge 5.1 Kablosuz sensör düğümünün maliyet analizi	37
Çizelge 5.2 Testlere ait yapılandırma bilgileri	38
Çizelge 5.3 Laboratuvar ölçüm istatistikleri	44
Çizelge 5.4 Sera ölçüm istatistikleri.....	50



1. GİRİŞ

Dünya nüfusunun artışıyla orantılı olarak, yılın her mevsiminde kaliteli taze sebze ve meyve gereksinimi sürekli artmaktadır. Dünyada sera sektörü bugünün gelişmiş yapısına uzun bir dönemde ulaşılmış ve bu gelişim süreci hâlâ devam etmektedir. Sera teknolojisinin gelişiminde yapısal dayanım, düşük enerji tüketimi, yüksek ışık geçirgenliği, makine-donanım-sistem olarak fonksiyonel özellikler önemli rol oynamaktadır (Dayıoğlu 2014b).

Modern seraların gelişim süreci beş safhada incelenebilir (Dayıoğlu 2015):

- 1950 – 1973 deneme-yanılma metodu ile yapısal gelişmeler sağlanmış,
- 1974 – 1978 araştırma deneyimi kazanılmış,
- 1979 – 1999 modern sera geliştirilmesine yönelik çalışmalar yapılmış,
- 2000 – 2010 seralara fonksiyon kazandırma sağlanmış
- 2010 – ... akıllı sera teknolojisi geliştirmeye yönelik çalışmalar devam etmektedir.

1.1 Sera

Seralar, yetiştiriciliği hedeflenen bitkiler için uygun çevre şartlarının sağlanması amacıyla paslanmaz çelik konstrüksiyon karkas üzerine cam, polietilen ve polikarbonat gibi ışık geçirgen malzemelerle kaplanmış yüksek örtüaltı yetiştiricilik yapılarıdır.

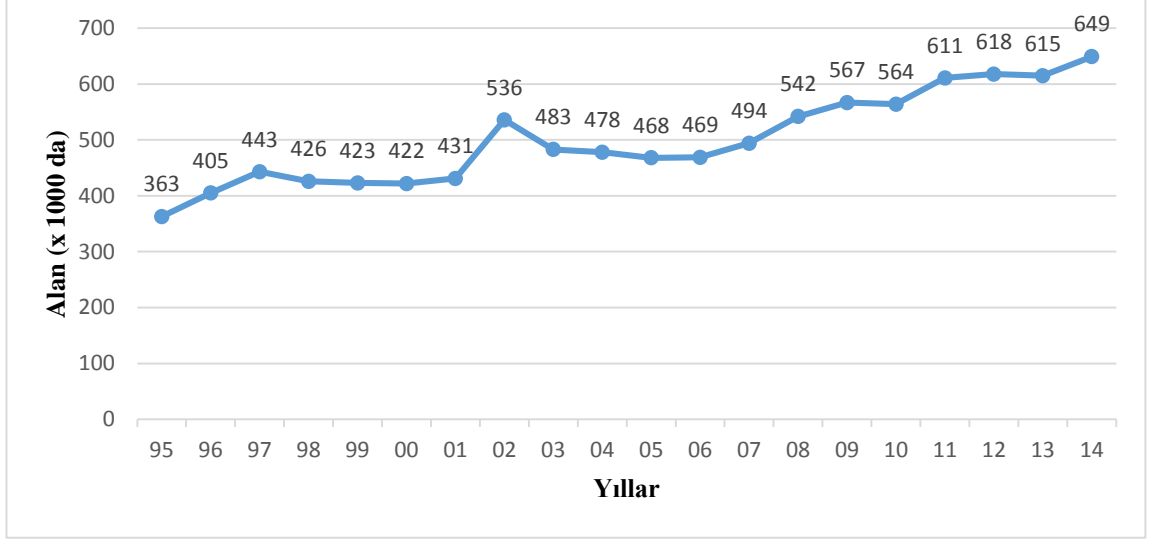
Türk Standartları Enstitüsü'nün 12741 standardına göre ise “İklimle ilgili çevre koşullarına tamamen veya kısmen bağlı kalmadan gerektiğinde sıcaklık, nem, ışık ve havalandırma gibi faktörleri kontrol altında tutularak bütün yıl boyunca çeşitli kültür bitkileri ile bunların tohum, fide ve fidanlarını üretmek, bitkilerini saklamak, sergilemek amacıyla cam, plastik vb. ışık geçirebilen maddelerle kaplanarak değişik şekillerde inşa edilen yüksek sistemde bir örtü altı yetiştiriciliği yapısı” olarak tanımlanmaktadır (Anonim 2001).

1.2 Türkiye’de Sera Sektörü

Ülkemiz nüfusun artmasına rağmen tarım arazisi miktarının sabit olması, varolan tarım arazilerinin de miras yoluyla bölünmesi gıda arzında sorunlara neden olmaktadır. Küçülen tarım arazileri, modern tarım yöntemlerinin uygulanmasına engel teşkil etmektedir. Bu durum uluslararası arenada tarım ürünleri alanındaki rekabet gücümüzü azaltmakla birlikte sosyokültürel açıdan bakıldığında kırsal nüfusun şehirlere göçü ve sonucu olan çarpık yerleşme, işsizliğin artması gibi önemli sorunlara yol açmaktadır. Sera yetiştiriciliği ile birlikte;

- Tüm yıl ürün alınarak birim alandan sağlanan gelir artışı,
- Pazara sürekli ürün arzı,
- Birim alandan alınan ürün miktarı ve kalitesinin artması,
- Sürekli işgücü gereksinimi nedeniyle kırsaldaki işsizliği azaltarak göçü engellemesi,
- Sera malzeme ve ekipman ihtiyacının karşılanması için yeni sanayi kollarının geliştirilmesi sağlanabilir.

Türkiye’de örtü altı yetiştiriciliği 1960’lı yıllarda tarımda plastiğin kullanımıyla birlikte yaygınlaşmaya başlamıştır. Özellikle 1970’li yıllarda ısıtma maliyetlerinin yükselmesine neden olan petrol fiyatlarındaki artışlar sonucunda sera yatırımları duraklama dönemine girmiş ve 1980’li yıllarda sera örtü materyallerindeki gelişmeler sonucunda tekrar ivme kazanmıştır. Sera yatırımlarındaki en hızlı gelişme 1990-1995 yılları arasında uygulanan % 25’lik kaynak kullanımı ve destekleme fonu teşvikleri sonucunda olmuştur. Bu dönemde yapılan seralarda havalandırma açıklıkları başta olmak üzere, bazı teknik iyileştirmeler bitkisel üretimin verim ve kalitesinde de artış sağlamıştır. Yüksek teknolojinin kullanıldığı modern seraların ve topraksız tarımın uygulanmaya başlaması (1990’lı yıllar) ve 2000’li yıllarda da sürdürülebilir üretim tekniklerinin ve danışmanlı ve sertifikalı üretimin yaygınlaşmaya başlaması ile Türkiye seracılığı bugünkü düzeyine ulaşmıştır (Tüzel ve Gül 2008). 1995-2014 örtü altı tarım alanlarındaki değişim şekil 1.1’de verilmiştir (Anonim 2015).

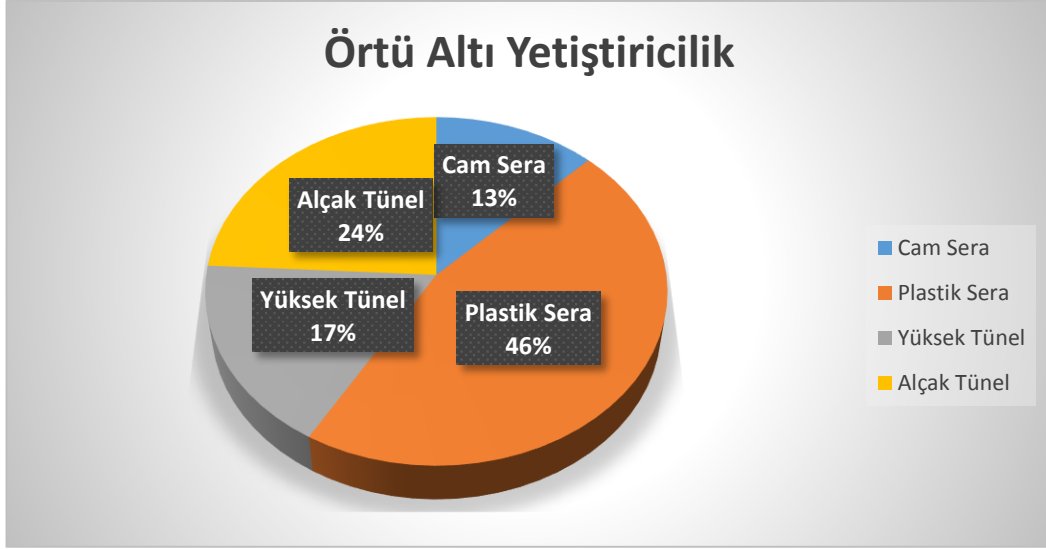


Şekil 1.1 Örtü altı tarım alanlarındaki değişim (Anonim 2015)

Ülkemizde örtü altı tarımının artışıdaki başlıca nedenler aşağıda sıralanmıştır (Tüzel vd 2010):

- Turfanda sebzeye oluşan yüksek iç talep,
- Seracılığın hızla arttığı yılların bir yıl öncesinde yatırımcıyı yeni sera kurmaya yönlendiren cazip ürün fiyatları,
- Aile işletmeciliğinin (ort. 1.000 - 3.000 m²) hâkim olması,
- Ek iş gücü ihtiyacının ortakçı sistemi ile çözülmüş olması ve bu sistemin işveren-üretici konumuna geçişe olanak sağlaması,
- 1990-1995 yılları arasında sağlanan % 25'lik kaynak kullanımı ve destek fonu teşvikleri

Toplamda 38,56 milyon hektar tarım arazisi olan ülkemizde bu alanların 64.912 ha örtü altı yetiştiricilik için kullanılmaktadır. Tüm örtü altı tarım alanlarının 80.976 dekarı cam, 298.561 dekarı plastik sera, geri kalanı ise yüksek ve alçak tünel olarak kullanılmaktadır. 2014 yılına ait örtü altı yetiştiricilik sistemi dağılımı Şekil 1.2'de verilmiştir (Anonim 2015).



Şekil 1.2 Türkiye’de örtü altı yetiştiricilik sistemleri dağılımı (Anonim 2015)

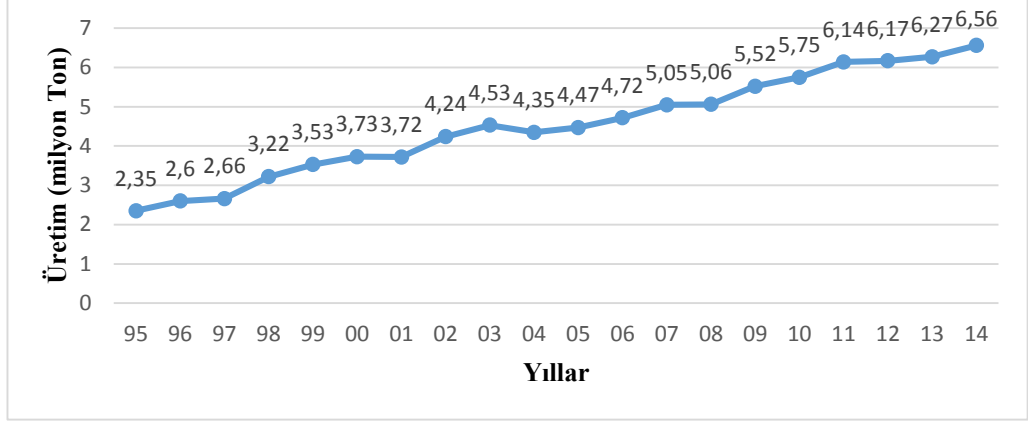
Ülkemizde örtü altı yetiştiriciliği ekolojik koşullara bağlı olarak Akdeniz bölgesinde yoğunlaşmıştır (Sevgican vd. 2000). TÜİK verilerine göre 2014 yılında iller bazındaki örtü altı yetiştiricilik yapılan iller Çizelge 1.1’de verilmiştir.

Arazinin ekonomik kullanımına olanak sağlaması, devlet teşvik ve destekleri, işletmelerin yapısı gibi nedenlerden dolayı ülkemiz genelinde seracılık artış göstermektedir.

Çizelge 1.1 Türkiye’de örtü altı yetiştiriciliğin yoğun olduğu iller (Anonim 2015)

İl	Toplam	Cam Sera	Plastik Sera	Yüksek Tünel	Alçak Tünel
Antalya	248.253	67.025	155.091	13.514	12.623
Mersin	158.846	6.472	75.253	54.103	23.017
Adana	94.479	6	748	3.455	90.270
Muğla	42.651	6.179	31.440	388	4.643
Samsun	21.512	0	348	6.474	14.690
Diğerleri	83.377	1294	35.770	34.838	11.476

2014 yılında örtü altı koşullarında sebze ve meyve üretimi toplamda yaklaşık 6,56 milyon ton olarak gerçekleşmiştir. Üretimin 6,22 milyon tonluk kısmı sebze, 339 bin tonluk kısmı ise meyvedir. En çok yetiştirilen sebze 3,285 milyon ton ile domates olurken meyve ise 180 bin tonluk üretimle muzdur (Anonim 2015). 1995-2014 yılları arasında üretilen toplam örtü altı ürün miktarı Şekil 1.3’te verilmiştir.



Şekil 1.3 Yıllara göre üretilen toplam örtü altı ürün miktarı (Anonim 2015)

1.3 Sera Teknolojisi

Modern sera, uygun iklim koşullarının ve yetiştirme ortamlarının yapay olarak yaratıldığı ve donanım olarak birçok özelliklere sahip olan bitki üretim fabrikalarıdır (Dayıoğlu 2015). Örtü altı yetiştiriciliğinde uygun iklim şartlarının sağlanması yapay olarak sağlanmaktadır. Bu durum, sistemin açık alanda yapılan yetiştiriciliğe göre daha karmaşık olmasına neden olmaktadır. Dış hava nemi ve sıcaklığı, rüzgâr yönü ve hızı, güneş ışınımı, sera içi nemi ve sıcaklığı, buharlaşma, toprak sıcaklığı vb. etkenlerin hepsi birbirleriyle etkileşim halindedir. Mükemmel bir otomasyon sistemi bu etkenleri iyi algılayabilmeli, değerlendirebilmeli ve etkili kontrol özelliğine sahip olmalıdır (Kürklü ve Çağlayan 2005).

Sera içinde iklimin yapay olarak sağlanması için aşağıda sıralanan donanımların bulunması gereklidir (Dayıoğlu 2015):

- Isıtma,
- Havalandırma,
- Soğutma sistemi
- Gölgeleme,
- Sisleme sistemi,
- CO₂ enjeksiyon sistemi.

Otomasyon sistemi kullanılmayan bir sistemde ısıtma maliyetinin bazı durumlarda toplam maliyetin % 65'ini geçtiği görülmektedir. Bu maliyet nedeniyle sera ortamında yeterli ısıtmanın sağlanmaması durumunda ise ürün kalitesi ve miktarında istenilen verim alınamamaktadır (Baytürk vd. 2013).

Sera otomasyon sisteminin aşağıda sıralanan fonksiyonları yerine getirmesi gerekir (Dayıoğlu 2014b):

- Ölçme
- Kumanda
- Denetleme
- Veri depolama
- Veri alışverişi
- Görüntüleme
- Alarm

Sensörler, denetleyiciler ve kumanda cihazlarının oluşturduğu katman birlikte çalışmaktadır. Seradaki iklim kontrolünün başarısı ancak, iklim kontrolünü sağlayan kontrol katmanlarının saniye zaman ölçeğinde doğru veri akışı ve çıktısı ile sağlanabilmektedir.

Sera gibi hacimsel olarak geniş ve karmaşık ortamlarda kablo ile iletişim çoğunlukla sorun yaratmaktadır. Günümüz elektronik, sensör ve kablosuz iletişim teknolojileri uygun maliyette esnek çözümler sunabilmektedir (Dayıoğlu 2014a).

Modern seralar bitkilerin yıl boyu üretimi için uygun iklim koşullarının yapay olarak oluşturulduğu ışık geçirgen yapılardır. Bu yapılarda amaç, değişken dinamik koşullara uygun olarak çok miktarda, yüksek kalitede ve düşük maliyetli ürün yetiştirmektir. Sera içinde bitkiler farklı büyüme evrelerinde gelişirken, dış ortamda iklim koşulları sürekli değişmekte ve bunlar, seranın etkin bir şekilde yönetimini zorlaştırmaktadır. Bu zorlukları endüstriyel seviyede çözmek için, seralar enerji-iklim yönetimi, sulama-

gübreleme yönetimi, bitki kalitesi yönetimi, veri yönetimi gibi birçok katmanda yönetilmelidir. Başarılı yönetim için bu katmanların her birinde verilerin ölçülmesi, izlenmesi ve depolanması gerekmektedir. Bu nedenle seralarda sıcaklık, bağıl nem, ışık, pH, EC ve CO₂ konsantrasyonunu ölçen birçok sensör ile birlikte, iletişim ve otomasyon teknolojileri kullanılmaktadır (Dayıoğlu, 2014 b).

Sensör verilerinin iletimi için RS-485 (ModBus), CAN (Controller Area Network), Bluetooth, Wi-Fi (Wireless Fidelity), GPRS (General Packet Radio System), GSM (Global System for Mobile) ve Internet gibi farklı iletişim protokolleri mevcuttur. Ancak son yıllarda, yaygın kullanımı, hızı, geniş alanlarda yayılma potansiyeli, uzak veri iletişiminde interneti öne çıkarmaktadır. Endüstriyel uygulamalarda prosesin izlenmesi için pahalı birçok çözüm mevcuttur.

Konvansiyonel internet teknolojisinden farklı olarak (Serodio ve ark. 2001; Yang ve Chen 2003), günümüz internet teknolojisi elektronik, mikro bilgisayar, sensör, iletişim teknolojilerindeki gelişmeler sayesinde gömülü sunucu uygulamalarına doğru değişmektedir (Bertolotti ve Hu, 2015). ARM (Advanced RISC Machine) tabanlı gömülü Ağ sunuculu Internet uygulamalarında kablolu ve/veya kablosuz birçok çözüm üzerinde çalışmalar yapılmaktadır (Hariyale ve Gulhane 2012; Montoya ve ark. 2013). Uygulamada gömülü sunucuların düşük maliyeti, az güç tüketimi, iletişim hızı, açık kaynak kod kullanımı, bir yazılıma ihtiyaç duyulmadan istedikleri bilgisayardan bağlanabilmesi ve verilerin kolaylıkla izlenebilmesi, esneklik ve görsellik gibi özellikleriyle ön plana çıkmaktadır. Süreçlerin uzak noktalardan güvenli olarak izlenmesi ve kontrolü mümkün olabilmektedir.

Günümüz teknolojisinin geldiği noktaya paralel olarak nesnelerin interneti (IoT: Internet of Things) kavramı ile birlikte her elektronik cihaz internetin küçük bir parçasına dönüşmektedir. Sensör yada sensörlerin bağlanabildiği bu mikrobilgisayarların Sensör Ağ düğümlerin seralarda kullanımı esnek çözümler sağlayabilecektir.

Bu yüksek lisans tezi kapsamında sera çevresindeki sıcaklık deęişimini izlemek için prototip olarak ARM-ađ tabanlı kablosuz takip sisteminin geliştirilmesi amaçlanmıştır.



2. KAYNAK ÖZETLERİ

Serodio vd. (2001), modern seralardaki süreçler için kontrol bilgisayarlarının önemini vurgulayarak, farklı platformlarla çalışabilen CAN (controller area network), RF bantları (433 ve 458 MHz), Ethernet ve internet teknolojilerini kullanarak veri toplama ve kontrol işlemlerini gerçekleştiren düşük maliyetli ve genişletilebilir bir sistem geliştirmişlerdir. Geliştirdikleri sisteme grafiksel kullanıcı arayüzü ekleyerek kullanımını kolaylaştırmışlardır.

Fukatsu ve Hirafuji (2005), tarım alanları ve çevresel görüntüleme için internet aracılığı ile erişim, ağ sunucu teknolojileri ve kablosuz sensör ağları, kablosuz tarla başı sunucuları hakkında bilgi vermişlerdir.

Zhu ve ark. (2006), serada sıcaklık, nem ve ışık yoğunluğu verilerini periyodik olarak toplamak için kablosuz sensör ağı uygulaması üzerinde çalışmışlardır. Çalışmalarında sera içindeki hava sıcaklığı, bağıl nem ve yaprak sıcaklığını ölçmek için ZigBee sensör düğümlerini kullanmışlar; toplanan verileri ana istasyona göndererek bir veri tabanında depolanmasını sağlamışlardır. İnternet aracılığıyla uzak kontrol sistemine veriler düzenli zaman aralıklarında gönderilmiştir. Seranın havalandırma, sulama ve ısıtma otomasyonu kontrol komutları kullanılarak sağlanmıştır.

Hwang ve Yoe (2011), kablosuz sensör ağı uygulamaları için ara katmanların tarım uygulamalarındaki çalışmaların diğer sektörlere göre yetersiz olduğunu belirterek kablosuz sensör ağı teknolojisi ve birleşik servislerle birlikte seradan veri toplayan yer-bağımsız ara katmanı geliştirerek biyolojik veriyi toplamışlardır.

Junxiang ve Haiqing (2011) yaptıkları çalışmada gömülü ağ teknolojisinin hassas tarımda uygulamalarının arttığından bahsetmişlerdir. Arm işlemcisi ve GNU/Linux işletim sistemi temel alınarak geliştirdikleri gömülü ağ sunucusuyla sera bilgilerini toplayıp aktarmışlardır. Geliştirdikleri cihazın yazılım kısmında gömülü ağ sunucusu Boa ve gömülü veritabanı yönetim sistemi olan SQLite'tan yararlanmışlardır. CGI

(Common Gateway Interface) tabanlı geliştirdikleri programı ise kullanıcı etkileşimi için kullanmışlardır. Araştırmacılar yaptıkları deneyler neticesinde geliştirdikleri sistemin performansının oldukça stabil olduğu, gerçek zamanlı veri toplama ve uzaktan kontrol konusundaki tasarım amacına ulaştığını belirtmişlerdir.

Xia vd. (2011), 7. Uluslararası Mobil Ad-Hoc ve Sensör Ağları Konferansında sundukları çalışmada, geleneksel tarımda yetersiz gerçek zamanlı veri toplama, kısıtlı alan izleme, yüksek bedensel işgücü gereksinimi vb. problemleri çözmek için Çin koca yemişi yetiştirilen serada kullanılmak üzere kablosuz sensör ağı kullanan çevre izleme sistemi geliştirdiklerinden bahsetmişlerdir. Geliştirdikleri sistem kurulduğu alanda sıcaklık, nem, aydınlanma gibi parametreleri GPRS teknolojisi ile gerçek zamanlı aktarmaktadır. Ayrıca sisteme entegre ettikleri alt sistemlerle gerçek zamanlı sesli ve kısa mesaj servisi (SMS) alarm desteğini de kullanmışlardır. Denemelerin yapıldığı alandaki enerji sorununu güneş panelleri ve bataryalar ile çözen araştırmacılar, geliştirdikleri sistemin düşük maliyetli, kararlı ve genişletilebilir olduğunu savunmuşlardır.

Aland ve Banhazi (2014), “Hayvan barınakları” adlı kitaplarında çiftlik hayvanlarının optimum yetiştirme koşullarının sağlanması, modern yönetim süreçleri, gerçek zamanlı internet teknolojilerinin kullanımı konularında detaylı bilgiler vermişlerdir.

Ateş (2013), Control of mobile robots via internet and wireless communication adlı yüksek lisans tezi çalışmasında genel amaçlı bir mobil robot platformu tasarlayıp üretmiştir. Çalışmasında Raspbian dağıtımını çalıştıran Raspberry Pi tek kartta bilgisayarı kullanmış ve Raspberry Pi ile robotun kontrolünü sağlamıştır. Ateş çalışmasında, robotu HTML5 ve PHP dilleri kullanarak tasarladığı ağ arayüzü ile kontrol etmiş; ağ arayüzü ile robot arasındaki veri iletişimini JSON kullanarak sağlamıştır.

Baytürk vd. (2013), makale olarak yayınladıkları çalışmada, yerleşim yerine uzak sera ortamlarının denetlenmesine olanak sağlayan gömülü sistem ve internet tabanlı maliyet avantajlı bir sera otomasyon sistemi yaptıklarından bahsetmişlerdir. Geliştirdikleri sistem ile internet üzerinden uzaktaki seraya müdahale edebilme ve sulama sistemini

kontrol edebilme olanağı sağlamışlardır. Yazarlar ayrıca geliştirdikleri sistemin ufak değişikliklerle diğer tarım alanlarında da kullanılabileceğini belirtmişlerdir.

Dayıoğlu (2013) yaptığı çalışmada çalışmasında, kablosuz iletişim teknolojisinin kablolu teknolojiye göre maliyet, montaj ve yönetim konularında daha uygun seçenek olduğunu belirterek seralar için Bluetooth'lu kablosuz ölçüm sistemi tasarlamıştır. Tasarladığı sistemde PIC mikrodenetleyicisi, bluetooth modülleri ve sayısal sıcaklık ve nem sensörlerine yer vermiştir.

Mahadik ve Chilveri (2013), yayınladıkları çalışmalarında ARM işlemci tabanlı Ethernet arabiriminin tasarımına değinmişler ve geliştirdikleri cihaza sayısal sıcaklık ve nem sensörü bağlayarak LAN'daki farklı bir bilgisayarla sensör değerlerini okumuşlardır.

Birajdar (2014), ARM tabanlı mikroişlemciye sahip Raspberry Pi tek kartta bilgisayar ve GNU/Linux işletim sistemiyle gömülü ağ sunucusu inşa etmiştir. Ayrıca yazar, gömülü ağ sunucularının maliyeti azaltmaya yardımcı olduğunu belirtmiştir.

Dayıoğlu (2014a), sera çevresinin gerçek zamanlı izlenmesi için endüstriyel Bluetooth teknolojisine dayalı olarak çalışan bir sistem geliştirmiştir. Geliştirdiği sistemde yönetici bilgisayar ve ölçüm birimleri Bluetooth tabanlı kablosuz ağ üzerinden haberleşmektedir. Sistemi beş kablosuz ölçüm birimi domates serasında 24 saat süreyle test eden araştırmacı, sera içindeki farklı noktalardan bağıl nem ve sıcaklık verilerini toplamış ve değerlendirmiştir.

Gawali ve Gajbhiye (2014), Arm7 işlemcisi ve ENC28j60 ethernet yongası ile tarımsal uygulamalar için gömülü ağ sunucusu geliştirmişlerdir. Çoğu internet uygulamasının istemci/sunucu bazlı çalıştığını belirten araştırmacılar; kapladığı hacim, maliyet ve enerji kullanımı göz önüne alınarak PC sunucular yerine gömülü ağ sunucuların kullanımının daha yerinde olduğunu belirtmişlerdir.

Patil ve Malviya (2014), yayınladıkları çalışmalarının odak noktasını tarımda sulama tekniklerinin modernleşmesi ve belirli alanların verimli sulanması olarak belirlemişlerdir. ARM tabanlı işlemci ve GSM modem desteği ile geliştirdikleri sistemi kısa mesaj sistemi (SMS) ile kontrol etmişlerdir.

Ersin (2015), yüksek lisans tez çalışmasında güneş paneli ile çalıştırdığı mikrodenetleyici ve nem sensörünü kullanarak, toprak nemine göre otomatik sulama yapan bir cihaz tasarlamıştır. Tasarladığı cihazda Raspberry Pi tek kartta bilgisayarı, sulamadan sonra bitki fotoğrafını çekip, operatöre bilgi e-postası yollayacak yardımcı sistem olarak kullanmıştır.

Schauer ve Debita (2015), yayınladıkları makalelerinde IoT'nin dağıtık sistemler için yenilikçi bir yaklaşım olduğunu ve her geçen gün popülerliğinin arttığından bahsetmişlerdir. IoT uygulamaları tasarlayanın zorlayıcı bir araştırma konusu olduğunu belirten araştırmacılar, servis yönelimli mimari deneyimlerine göre IoT uygulaması tasarım süreci önermişlerdir. Ayrıca araştırmacılar standardizasyonun önemini vurgulayarak, farklı yöntemleri karşılaştırmışlardır.

Srbinaovska ve ark. (2015), serada bilimsel sebze üretimi ve çevresel verilerin takibi sayesinde daha az yönetim maliyetini sağlamak için kablosuz sensör ağı mimarisi önermişlerdir. Sıcaklık, nem ve ışık gibi temel çevresel parametreleri izlemek için pratik ve ucuz maliyetli bir veri izleme sistemini kablosuz sensör ağı teknolojisine dayalı olarak tasarlamışlardır.

Vijayakumar ve Ramya (2015), yayınladıkları makalede içme suyunun güvenli dağıtımını sağlamak için su kalitesinin gerçek zamanlı izlenmesi gerektiğini belirtmişlerdir. İçme suyu kalitesini ölçmek için tasarlayıp geliştirdikleri sistemin düşük maliyetli bir nesnelerin interneti uygulaması olduğunu belirtmişlerdir. Çalışmada sensörlerin bağlandığı ana kontrolcü Raspberry Pi tek kartta bilgisayarın düşük maliyetli, verimli ve toplanan verilerin işlenmesi, analizi, bulut sunucuya gönderilmesi ve izlenmesi için yeterli olduğunu savunmuşlardır.

Vujovic ve Maskimovic (2015), yayınladıkları makalelerinde, ev otomasyonu ile ilgili uygulamalarda Nesnelerin İnterneti (IoT) ile gerçekleştirmeye başladığını vurgulamışlardır. Nesnelerin İnterneti kavramının temel üstünlüğünün her bir cihazın internetin bir parçası olduğuna değinmişlerdir. Raspberry Pi tek kartta bilgisayarın ucuz, tamamen özelleştirilebilir ve programlanabilir nesnelerin interneti düğümü olarak tanımlamışlardır.

Zaceping ve Kviess (2015), yayınladıkları makalelerinde hassas arıcılıkta kullanılabilecek gerçek zamanlı arı kolonisi sıcaklık izleme sistemi mimarisi için altı farklı yaklaşımı değerlendirmiştir. Yaklaşımlar için işletme ve kurulum maliyetlerinin önemine vurgu yapan araştırmacılar, en uygun yöntemin izlenecek arı kovanlarının çevresindeki imkânlar, yetiştiricinin ihtiyaçları ve bütçeye göre arı kolonisi sistem mimarisinin seçilmesi gerektiğini belirtmişlerdir.

3. KURAMSAL TEMELLER

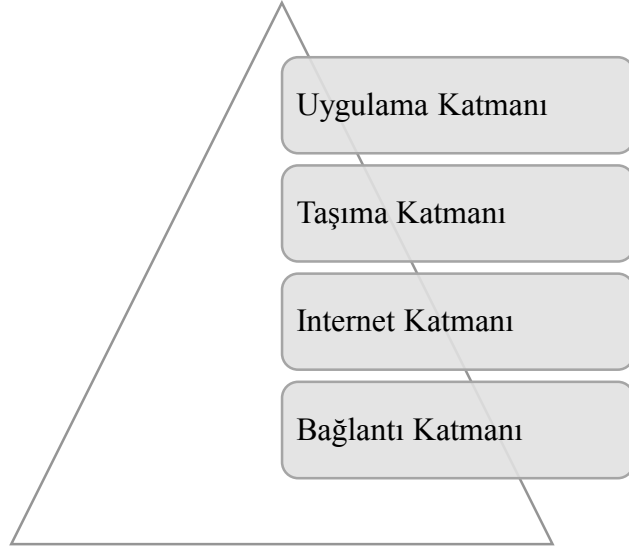
3.1 TCP/IP ve HTTP

TCP/IP, internet ve benzeri bilgisayar ağıları için haberleşme protokolü kümesidir. TCP/IP, internet paketi (internet suite) olarak da adlandırılmaktadır. TCP/IP ilk olarak Amerikan Savunma Bakanlığı (DoD) bünyesinde geliştirilmiş ve internetin atası olan ARPANet'te kullanılmıştır (Ertam 2005, Kılınç 2012).

TCP/IP, noktadan noktaya iletişimde verinin paketlenmesi, adreslenmesi, yönlendirilmesi ve alıcı tarafından alınması ile ilgili işlemleri tanımlamaktadır.

TCP/IP ile Şekil 3.1'de gösterildiği gibi dört soyutlama katmanı tanımlanabilir:

- Uygulama katmanı (application layer): Farklı sunucular arasındaki proses ve uygulama iletişiminden sorumlu katmandır. Uygulama katmanındaki protokollere DNS, FTP, http, SMTP, SSH örnek olarak verilebilir.
- Taşıma katmanı (transport layer): Noktadan noktaya veri akışından sorumlu katmandır. En çok kullanılan taşıma katmanı protokolleri TCP ve UDP'dir. TCP protokolü verinin sıralı ve kayıpsız aktarımını garanti ederken, UDP protokolü verinin başarıyla ve gönderilen sırayla aktarılacağını garanti etmez. UDP protokolü ile yollanan veri paketlerinin hedefe ulaşması halinde ise paketin içeriği hatasız olmaktadır.
- İnternet katmanı (internet layer): Ağ geçitleri ile birbirine bağlanmış ağ üzerinde verinin kaynaktan hedefe yönlendirilmesini sağlayan katmandır. ICMP, IPv4 ve IPv6 internet katmanlarından birkaçıdır.
- Bağlantı katmanı (link layer): İnternet protokolünün donanımdan soyutlanması için tasarlanmış katmandır. TCP/IP veri paketlerinin elektriksel ifadesinden bağımsız olarak farklı tip ağlar arasında iletilmesini sağlar. Ethernet, DSL, ISDN, PPPoE ve PPPoA sık kullanılan bağlantı katmanlarıdır.



Şekil 3.1 TCP/IP soyutlama katmanlarının şematik görüntüsü

TCP/IP katmanlarını kullanmak için yazılım tarafında en az 30 KB veri alanı ve 10 KB boyutunda geçici bellek alanı gerekmektedir. Bu durum çoğu mikrodenetleyicinin TCP/IP protokolünü kullanmasını engellemektedir (Kılınç 2012).

HTTP (hypertext transfer protocol), ağ'ın standart protokolüdür ve TCP/IP'de uygulama katmanı olarak yer alır. HTTP'deki komutlar istek-cevap şeklinde işlenerek istemci/sunucu modeli oluşturulmuş olur. Tipik bir http protokolü kullanımı şöyledir:

- i. İstemci, sunucunun ip adresini içeren bir TCP bağlantısı başlatır. Bu bağlantının hedef portu 80 olarak ayarlanır.
- ii. Sunucu istemcinin isteğini kabul eder.
- iii. Sunucu istemciye gerekli veriyi gönderir.
- iv. TCP bağlantısı kapanır.

HTTP bu yüzden durumsuz olarak nitelendirilir. Başka bir ifadeyle sunucu, istemcinin geçmiş isteklerini hatırlamak zorunda değildir. İstemcinin her bir isteği, sunucu tarafından tekrar değerlendirilerek cevaplanır (Küçük 2015).

TCP/IP katmanlı mimarisi ve HTTP, zamanlama ile ilgili problemlere neden olabilir. İstemci/sunucu arasında gerçekleşen gecikme eş zamanlı izleme ve kontrol uygulamalarında dikkate alınması gereken bir konudur.

$$t_{toplam} = t_1 + t_2 + t_3 + t_4 + t_5$$

t_{toplam} : Toplam gecikme,
 t_1 : İstemcinin istek süresi,
 t_2 : Veri paketinin sunucuya ulaşması,
 t_3 : Sunucunun işlem zamanı,
 t_4 : Veri paketinin istemciye ulaşması,
 t_5 : istemcinin yanıtı çözmesi,

formülü ile toplam gecikme hesaplanabilir.

3.2 JSON

Açılımı JavaScript Object Notation (JavaScript nesne notasyonu) olan JSON, hafif bir veri değişim formatıdır. Programlama dillerinden bağımsız olması, tanımlayıcı yükünün az olması ve insan tarafından rahatlıkla manipüle edilebilmesi sebebiyle asenkron veri haberleşmesinde sıklıkla kullanılmaktadır. JSON modern programlama dillerinin de desteklediği aşağıdaki veri tiplerini saklayabilir:

- i. Sayı,
- ii. Karakter dizisi,
- iii. Mantıksal ifade (doğru – yanlış),
- iv. Dizi,
- v. Nesne,
- vi. Boş değer (null).

JSON olarak tanımlanmış veri örneği şekil 3.2’de verilmiştir.

```
{
  "istasyon" : "istasyon1",
  "olcumler" :
  [
    { "sensor" : "sensor1", "deger": 40.000},
    { "sensor" : "sensor2", "değer": 39.375}
  ]
}
```

Şekil 3.2 Örnek JSON verisi

3.3 WSN

WSN (kablosuz sensör ağı), çevrelerindeki fiziksel değişiklikleri algılayan akıllı cihazların kablosuz olarak oluşturduğu özel kablosuz ağıdır. Her bir cihaz topladığı veriyi işleyebilir veya geçit cihaza aktarabilir. Kablosuz sensör ağını oluşturan cihazlara düğüm (node) adı da verilebilir.

Kablosuz sensör ağındaki düğümler genellikle bir mikrodenetleyici, bir ya da daha fazla sensör, düşük güç tüketen kablosuz arabirim, güç kaynağının birleşmesiyle oluşturulur. Bazı durumlarda düğüme GPS gibi bir konum belirleyici ya da topladığı verileri uzun süre saklaması amacıyla geçici olmayan bellek (non-volatile memory) birimleri de eklenebilir.

Kablosuz sensör ağlarını diğer kablosuz ağlardan ayıran karakteristikler ise şunlardır:

- Düğümlerin taşınabilirliği: Düğümler ve geçit arasındaki bağlantı için fiziksel bir veri hattı gerekmediği için tüm ağ, düğümlerin kablosuz bağlantı arabiriminin desteklediği menzil içinde çalışmaya devam edecektir. Çalışan bir ağ içinde bağlantı arabiriminin etkinlik derecesine bağlı olarak düğümlerin konumca serbesttir.
- Katılımsız çalışma: Birçok kullanım senaryosunda düğümleri bir defa kurulum sırasında yapılandırmak yeterlidir.
- Düşük güç: Sensorlar, yapıları gereği çok fazla güç tüketmezler. Bu bakımdan güneş paneli gibi alternatif güç kaynakları ile beslenebilirler.

- Fazla sayıda düğüm içerme: Düğümler, kablosuz sensör ağının kablosuz bağlantı arabiriminin desteklediği sayıda kullanılabilir.

WSN'nin kablosuz arabirimi olarak sıklıkla kullanılan standartlar ise ZigBee, 6LoWPAN, endüstriyel bluetooth, WiFi'dir. Bunlarında dışında WirelessHART, IEEE 1451 standartları da sıklıkla kullanılabilir.

3.4 1-Wire

1-Wire, Dallas Semiconductor Corp.'un (Maxim Integreted) geliştirdiği düşük hızlı seri veriyolu sistemidir. I²C veriyoluna benzemekle birlikte daha düşük hız ve daha yüksek menzile sahiptir. Bunun yanında 1-Wire seri veriyolunun önemli avantajları da vardır. Bunlar (Anonymous 2008):

- İletişim için referans toprak hattından alınır.
- Cihazların haberleşmesi için toprak ve data olmak üzere iki bağlantı yeterlidir.
- Aynı veri hattına bir tane "master" ve çok sayıda "slave" cihaz bağlanabilir.
- Tüm 1-Wire destekleyen "slave" cihazlar fabrika çıkışı benzersiz ve değiştirilemez 64 bit uzunlukta kimlik bilgisine sahiptir. 64 bit'in 8 biti ise cihazın tipini ve özelliğini barındırır.
- 1-Wire "slave" cihazlar 2,8 – 5,25 V arasındaki geniş aralıkta çalışabilir.
- Çoğu 1-Wire cihaz parazit enerjisiyle çalışabilir.

1-Wire çift yönlü iletişimi desteklememektedir. Piyasada sayısal sıcaklık sensörlerinden mobil ödeme sistemlerine yaklaşık 40 çeşit "slave" cihaz bulunmaktadır.

3.5 IoT

Nesnelerin interneti kavramı ile ilgili literatürde pek çok tanım yer almaktadır. Uluslararası Telekomünikasyon Birliği (ITU) tarafından yapılan tanımıyla IoT, herhangi bir zamanda herhangi bir yerde her nesnenin birbirine bağlanabileceği bir teknolojidir.

Bazı kaynaklarda nesnelerin interneti yerine her şeyin interneti olarak da yer alabilmektedir. Var olan tanımlardan yola çıkılarak Nesnelerin İnterneti, tüm nesnelerin çeşitli haberleşme protokolleri ve algılama yöntemleri aracılığıyla tanımlanarak birbirleri ile iletişime geçebileceği, internet ortamına çıkabilecekleri akıllı ağlardan oluşan bir teknoloji olarak tanımlanabilir (Bozdoğan 2015).

IoT konseptinin tarihi 1982 yılına kadar gitmektedir. İlk IoT kapsamında değerlendirilebilecek çalışma modifiye edilmiş gazlı içecek otomatıdır. Otomat, internet üzerinden soğuk içecek stoğu bilgisini vermektedir (Farooq vd. 2015). 1991 yılında Xerox firmasında araştırmacı olarak çalışan Mark Weiser yayınladığı 21. yy için Bilgisayar adlı makalesinde IoT ile ilgili gelecek vadeden fikirlerini aktarmıştır (Weiser 1991). 2009 yılında ise Kevin Ashton, internete bağlı cihazları açıklamak için IoT terimini kullanmıştır (Ashton 2009).

4. MATERYAL VE YÖNTEM

Tez çalışmasında, seralar için kablosuz sensör ağı ve ağ tabanlı sıcaklık izleme sistemi geliştirilmiştir. Sistem iki kablosuz sensör düğümü, kablosuz ağ geçidi ve istemcilerden oluşturulmuştur. Sistem maliyetini düşürmek ve sahada, sera çevresinde esnek çözüm sağlamak için kablosuz sensör düğümlerinden biri sunucu olarak belirlenmiştir. Sistem donanıma özel servis ve arayüz yazılımları esas alınarak esnek şekilde programlanmıştır.

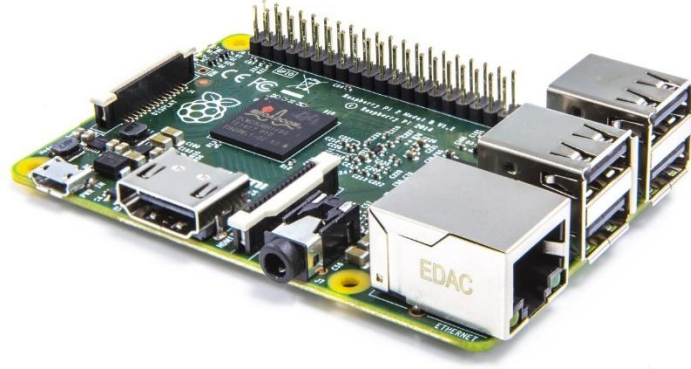
4.1 Sistem Bileşenleri

4.1.1 Kablosuz sensör düğümü

Her kablosuz sensör düğümü ARM tabanlı tek kartta bilgisayar, kablosuz ağ adaptörü, MicroSD hafıza kartı ve iki sayısal sıcaklık sensöründen oluşturulmuştur.

Kablosuz sensör düğümü bileşenleri, kullanılan modeller ve teknik özellikleri aşağıda sıralanmıştır:

- 1) ARM tabanlı tek kart bilgisayar: Raspberry Pi 2 Model B (Şekil 4.1, Çizelge 4.1)
- 2) Kablosuz ağ adaptörü: ZyXEL NWD-270N ve Asus USB-N10 Nano (Şekil 4.2.a)
- 3) Hafıza kartı: HLV-MCSDC10/8G (Şekil 4.2.b)
- 4) Sıcaklık sensörü: DS18B20 (Şekil 4.2.c,d, Çizelge 4.2)
- 5) Güç kaynağı (Belkin, 5 V DC, 2.6 A, 13 W)



Şekil 4.1 Raspberry Pi 2 Model B (www.raspberrypi.org 2015)

Çizelge 4.1 Raspberry Pi 2 Model B'nin teknik özellikleri (www.raspberrypi.org 2015)

SoC	Broadcom BCM2836
İşlemci	900 MHz dört çekirdek ARM Cortex-A7
Grafik İşlemci	Broadcom VideoCore IV @ 250 MHz
Bellek	1 GB
USB 2.0 portları	4
Dâhili Depolama	MicroSD slotu
Ağ Bağlantısı	10/100 Mbit/s Ethernet
Sayısal G/Ç	17× GPIO
Güç Tüketimi	800 mA (4.0 W)
Boyutlar	85.60 mm × 56.5 mm
Ağırlık	45 g

Kullanılan tek kartta bilgisayar dâhili kablosuz ağ bağdaştırıcısına sahip olmadığı için Linux çekirdeği tarafından ek sürücüye gereksinim duymadan kullanılabilen USB Wi-Fi adaptörleri kullanılmıştır (Şekil 4.2.a).



Şekil 4.2.a USB Wi-fi adaptörleri, b. MicroSD kart, c. DS18B20 sayısal sıcaklık sensörü, d. Su geçirmez DS18B20

Sera içinde hava ve toprak sıcaklıklarının ölçümünde paslanmaz çelik su geçirmez kapsül takılmış olan şekil 4.2.d’de gösterilen sensör kullanılmıştır.

Çizelge 4.2 DS18B20 teknik özellikleri (Anonymous 2008)

Ölçüm aralığı	-55/125°C
Çözünürlük	Programlanabilir 9 - 12 bit
Hassasiyet	±0.5 °C
Yinelenebilirlik	±0.1 °C
Güç tüketimi	5 mW
Güç kaynağı	3-5.5 VDC
İletişim	1-Wire
Ölçüm için çevrim süresi	750 ms (12 bit)
Kalibrasyon	Fabrika ROM

4.1.2 Kablosuz ağ geçidi

Ölçüm istasyonunun internete bağlanabilmesi ve diğer ölçüm istasyonları ile kablosuz iletişim kurabilmesi amacıyla ZyXEL VMG3312-B10B kablosuz erişim noktası özellikli vDSL modem kullanılmıştır.

4.1.3 Sunucu

Geliştirilen kablosuz ağ düğümlerinden biri ana bilgisayar olarak kullanılarak sistemde ayrı sunucu bilgisayar kullanımı gereksinimi ortadan kaldırılmıştır.

4.1.4 İstemciler

Sistemde, istemci olarak HTML5 standardını destekleyen ağ tarayıcı yazılımına sahip dizüstü bilgisayar, tablet ve akıllı cep telefonları kullanılmıştır.

4.1.5 Yazılımlar

Raspbian Linux dağıtımı

Tez çalışmasında işletim sistemi olarak Raspbian Linux dağıtımı kullanılmıştır. Raspbian, Debian dağıtımı temelli bir genel kamu lisansı (GNU) Linux işletim sistemi türü olup, Raspberry Pi tek kartta bilgisayarlarda kullanılmak üzere optimize edilmiştir. Raspbian dağıtımının en önemli avantajı, çalışma sırasında ihtiyaç duyulan tüm yazılım paketlerinin derlenmiş halde kuruluma hazır olmasıdır. Bunun yanında Raspbian'ın çekirdeği, tek kablo (1-Wire) protokolünü kullanmak ve DS18B20 sensöründen veri okumak için gereken w1_gpio ve w1_therm çekirdek modüllerini sorunsuz çalıştırma yeteneğine sahiptir. w1_gpio ve w1_therm çekirdek modülleri sayesinde sıcaklık sensöründen okuma yapmak, herhangi bir metin dosyasından okuma yapmakla eşdeğerdir.

MonoDevelop

Ücretsiz olarak sunulan MonoDevelop, Raspbian yazılım deposundan “apt-get install monodevelop” komutuyla kurulmuştur. Tez çalışması için hazırlanan yazılımlar MonoDevelop bütünleşik geliştirme ortamında (IDE) C# ile yazılmış ve mono derleyicisi ile derlenmiştir.

Apache web server

Apache 2.2.22 sürümü, mod_php5 kütüphanesi ile birlikte Raspbian yazılım deposundan yüklenmiştir (Upton ve Halfacree 2013). İstasyonun konfigürasyon ve izleme arayüzü Apache internet sunucusu tarafından ağ ortamına aktarılmaktadır.

MySQL veritabanı

Çalışmada verilerin depolanması amacıyla MySQL veritabanı yönetim yazılımının 5.5.46 sürümü Raspbian yazılım deposundan yüklenerek kullanılmıştır (Upton ve Halfacree 2013).

4.1.6 Deney serası

Denemeler T.C. Gıda Tarım ve Hayvancılık Bakanlığı'na ait kuzey-güney doğrultusunda yerleştirilmiş, 2.000 m² büyüklüğünde çelik konstrüksiyonlu cam örtülü serada gerçekleştirilmiştir (Şekil 4.3). Serada çatı pencereleri, fan-ped soğutma sistemi, sulama sistemi, havalandırma fanları, ısı perdesi ve ısıtma sistemi mevcuttur. Denemenin yapıldığı serada ısıtma ve havalandırma yapılmamaktadır.

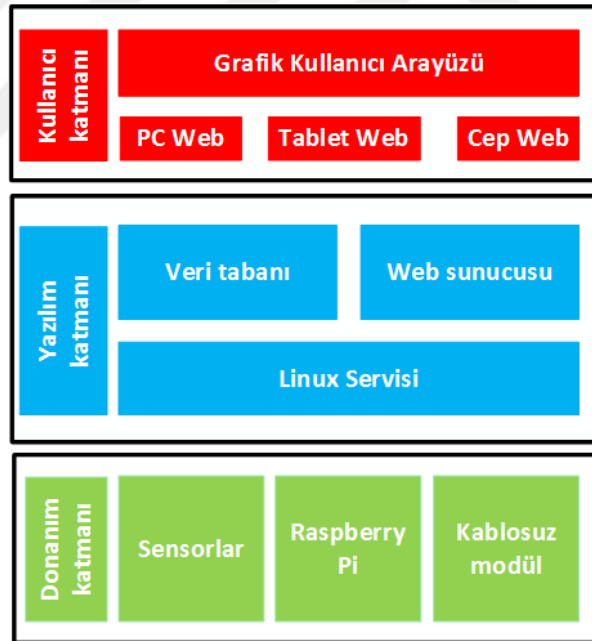


Şekil 4.3 Sistemin test edildiği sera

4.2 Sistem Mimarisi

Sistem şekil 4.4'te gösterildiği gibi, kablosuz sensör düğümü, kablosuz ağ geçidi, sunucu ve istemcilerden oluşacak şekilde üç katmanlı mimariye dayalı olarak tasarlanmıştır:

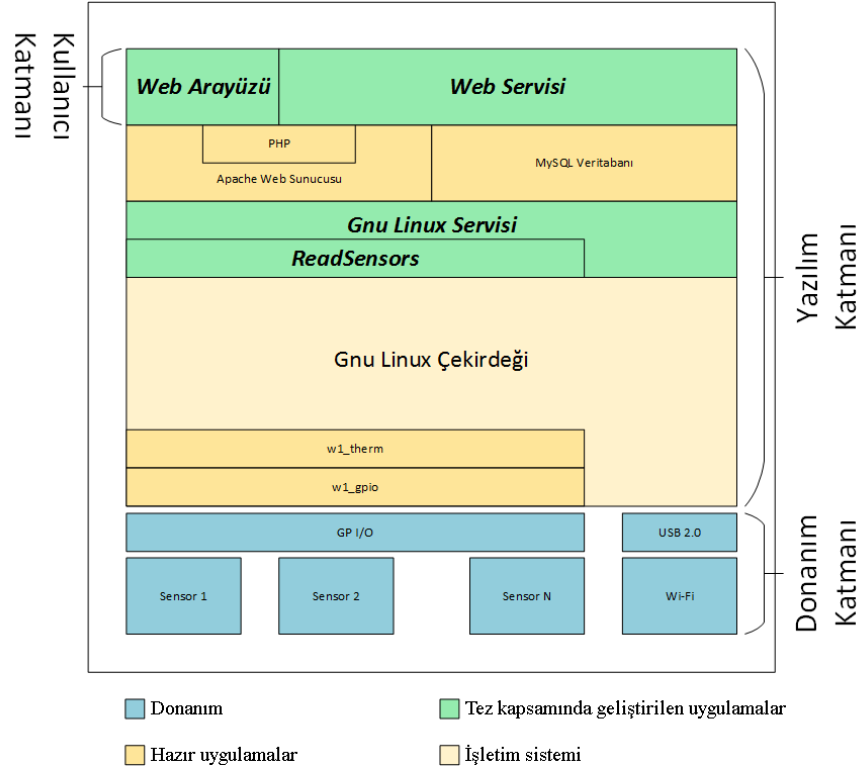
- a. Kullanıcı katmanı:** Ağ işaretleme dilini okuyup yorumlayabilen ve internet erişimine sahip tüm cihazları kapsayan geniş bir tanımdır. Bu cihazlara kişisel bilgisayarlar, akıllı cep telefonları ve diğer akıllı cihazlar örnek olarak gösterilebilir. Sistemin ürettiği verilerin ve genel durumun kullanıcıya sunulması, kullanıcı katmanındaki cihazların sahip oldukları ağ tarayıcıları sayesinde gerçekleştirilmektedir.
- b. Yazılım katmanı:** Fiziksel katmanı donanımdan soyutlayarak veri toplama, depolaması ve işleme görevlerini yerine getirmektedir. Soyutlama işlemi, kullanıcı deneyimi açısından kolaylık sağlamaktadır.
- c. Donanım katmanı:** Ölçümleri yapan sensörler, sensörlerden aldığı ölçüm verilerini kaydeden ve işleyen tek kartta bilgisayar, cihazların birbirleri ile iletişimi sağlayan kablosuz ağ geçidi ve tüm sistemin internete erişiminden sorumlu kablosuz yönlendirici (router) donanım katmanını oluşturur.



Şekil 4.4 Sistem katmanları

4.3 Kablosuz Sensör Düğümünün Tasarımı

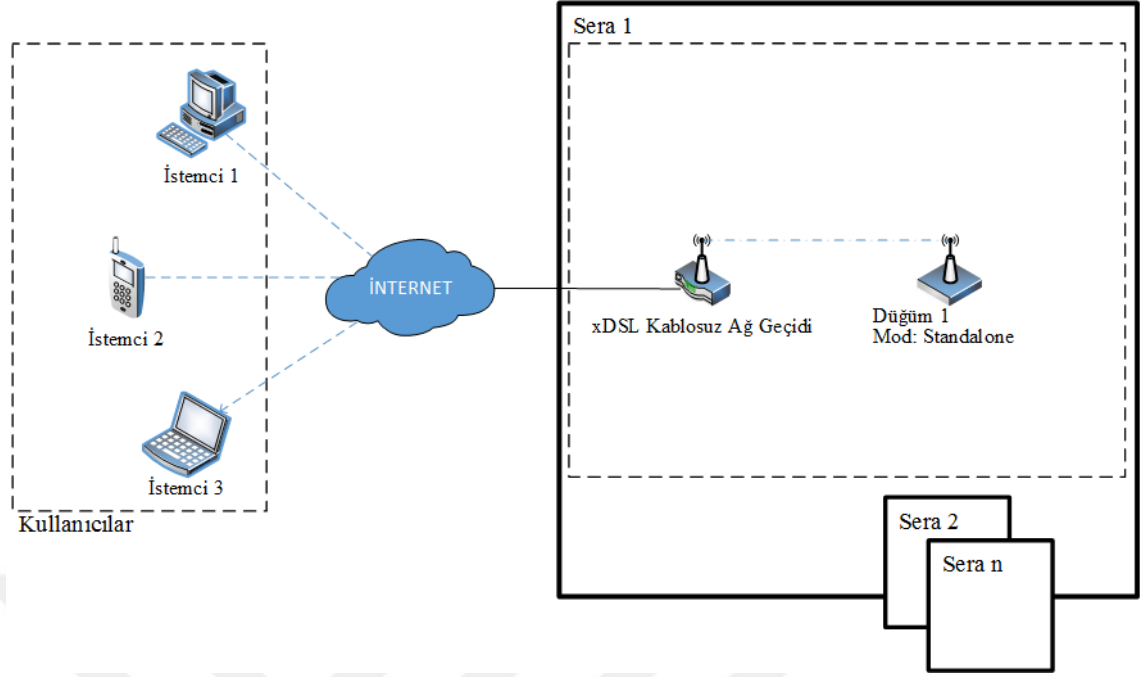
Sistemin donanım ve yazılım katmanını “kablosuz sensör düğümü” olarak adlandırılan cihaz sağlamaktadır. Tez kapsamında prototipi geliştirilen düğüm cihazının genel mimarisi şekil 4.5’te verilmiştir.



Şekil 4.5 Düğüm cihazının genel mimarisi

Sensör düğümü, önceki çalışmalardan farklı olarak iki kullanım senaryosu kullanılabilir şekilde tasarlanmıştır:

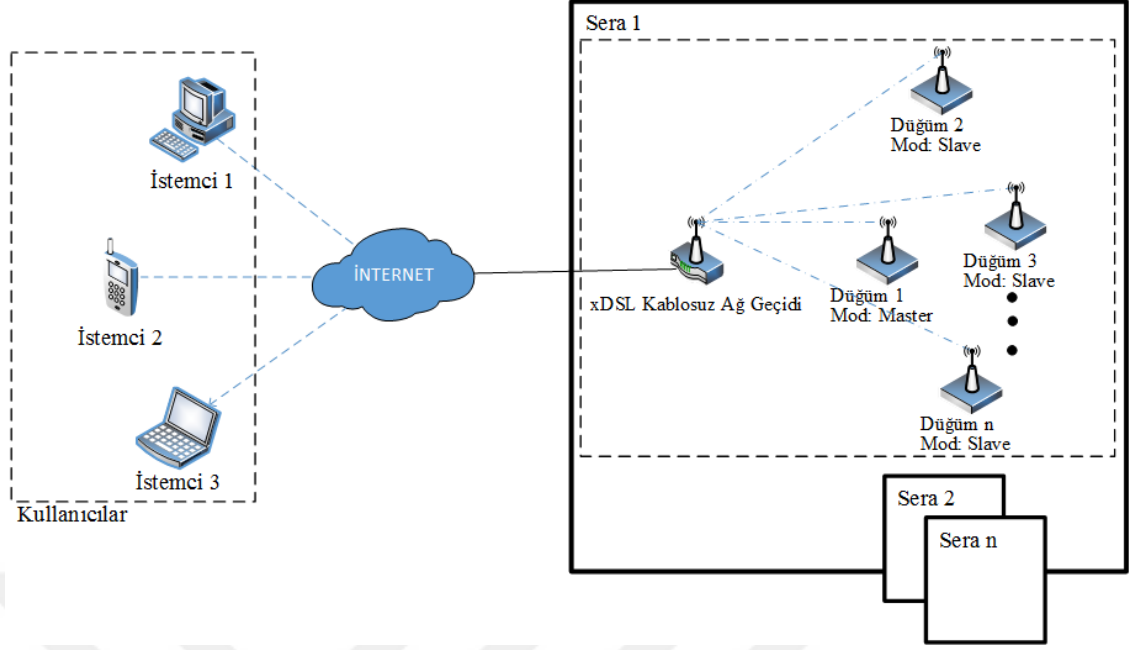
Senaryo 1: Sera içinde tek noktaya ölçüm istasyonu yerleştirilmesi söz konusu olan uygulamaları kapsamaktadır. Bu durumda cihaz wi-fi bağlantı ile doğrudan internete erişmektedir (Şekil 4.6). Bu senaryoda cihaz tek başına (standalone) çalışma şekline ayarlanarak sayısal sensörlerden topladığı veriyi dâhili veri tabanında saklar. Tek başına modundaki cihaza internet üzerinden erişilerek isteğe bağlı anlık ölçüm yapılabilmesinin yanı sıra önceki ölçüm verilerine de ulaşılabilir.



Şekil 4.6 Tek sensör düğümü kullanım diagramı

Senaryo 2: Birden çok en az iki adet sensör düğümüne gerek duyulan uygulamalar için oluşturulmuştur (Şekil 4.7). Sensör düğümlerinin kurulum aşamasında bir istasyon yönetici (master) çalışma şekline ayarlanır. Yönetici istasyon “IP” adresi ile Yönetilen (slave) moda ayarlanan uzak istasyonların “IP” adresi aynı olmalıdır. Kurulan sistemin doğru çalışması için hem yönetici olarak belirlenmiş istasyonun hem de diğer istasyonların ayarlarındaki doğrulama anahtarları aynı olmalıdır.

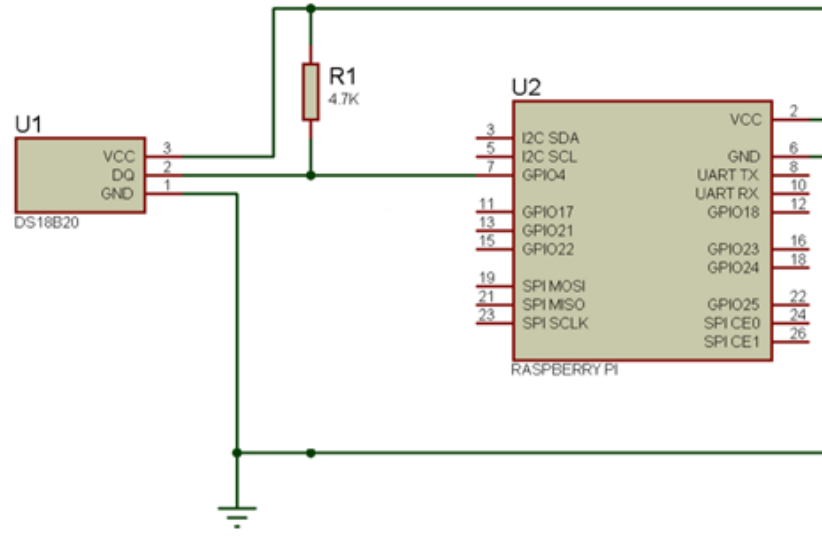
Birden çok sensör düğümünün kullanıldığı durumlarda, sadece yönetici cihaza internet erişimi vermek ve internet üzerinden erişilebilmesini sağlamak yeterlidir. Ölçüm istasyonlarının bulunduğu ağa içten erişimle, sadece ölçüm yapan düğümlere erişmek ve anlık ölçüm yapmak mümkündür.



Şekil 4.7 Birden çok sensör düğümü kullanım diagramı

4.4 DS18B20 - Raspberry Pi bağlantısı

DS18B20 'nin V_{CC} ve DQ bacakları arasında lojik sinyal seviyesi için 4.7 k Ω direnç bağlantısı (pull-up) yapılmıştır. DS18B20 tek kart bilgisayarın 2 numaralı 5V V_{CC} hattından beslenmiştir. Toprak için ise tek kartta ARM bilgisayarın 6 numaralı bacağı kullanılmıştır (Şekil 4.8).



Şekil 4.8 DS18B20 sensörünün bağlanması

4.5 Sıcaklık Bilgisinin Okunması

Raspberry Pi'nin 7 nolu pini (GPIO4) DS18B20'nin data hattına bağlanmıştır. Gerekli çekirdek modüllerinin yüklenmesinden sonra her bir sensörün ölçümleri "/sys/bus/w1/devices/" klasörü altına yazılan sensörlerin seri numaraları ve ilgili klasörler şekil 4.9'da görüntülenmektedir. Her bir seri numarası klasörünün içindeki w1_slave dosyası okuma yapıldığında, sensörden ölçüm yapılır ve sonuç geri döndürülür (Horan 2013).

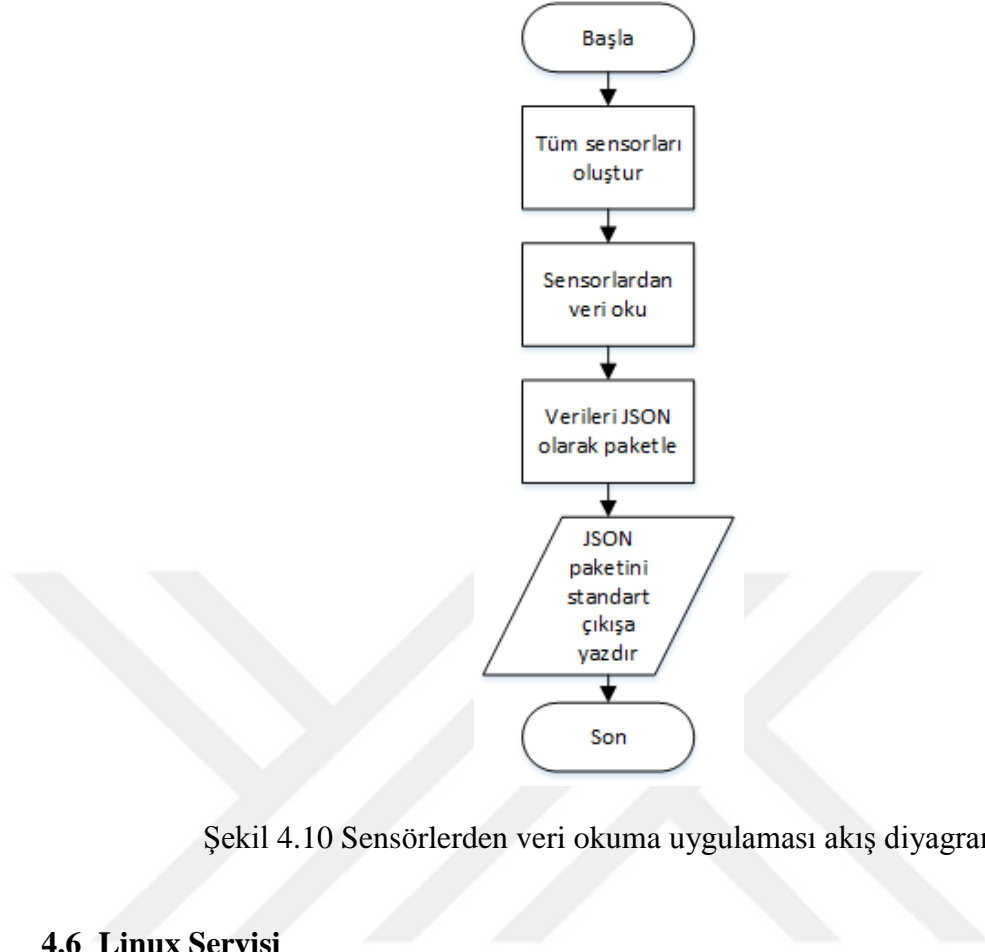
w1_slave dosyası toplam iki satırdan oluşmaktadır. İlk satır 16'lık sayısal sıcaklık bilgisini (byte 0: 4b ve byte 1: 01) içermektedir. Her ölçümde çevrimsel hata denetimi (cyclic redundancy check: crc) yapılarak doğru okuma onayı (YES) alındıktan sonra, t sıcaklık bilgisi ($t/1000 = 20,687 \text{ }^\circ\text{C}$) elde edilir.

Geliştirilen uygulama ile sisteme bağlı sensörlerden ölçümleri okumak için önce /sys/bus/w1/devices altındaki klasörlerin listesini alınır. Klasör listesinde, içinde "master" sözcüğü geçmeyen alt klasörler, 1-wire protokolü ile bağlı "slave" cihazlara aittir. Sensörler, sensör nesnesi olarak tanımlanır. Her bir "slave" cihaz, aynı zamanda "slave" cihazın seri numarası olan klasör adıyla birlikte kaydedilir.


```
pi@raspberrypi: /sys/bus/w1/devices/28-000003cee4ca
File Edit Tabs Help
pi@raspberrypi /sys/bus/w1/devices $ sudo modprobe w1-gpio
pi@raspberrypi /sys/bus/w1/devices $ sudo modprobe w1-therm
pi@raspberrypi /sys/bus/w1/devices $ cd /sys/bus/w1/devices/
pi@raspberrypi /sys/bus/w1/devices $ ls
28-000003cee4ca w1 bus master
pi@raspberrypi /sys/bus/w1/devices $ cd 28-000003cee4ca
pi@raspberrypi /sys/bus/w1/devices/28-000003cee4ca $ cat w1_slave
4b 01 4b 46 7f ff 05 10 e1 : crc=e1 YES
4b 01 4b 46 7f ff 05 10 e1 t=20687
pi@raspberrypi /sys/bus/w1/devices/28-000003cee4ca $ cat w1_slave
a2 01 4b 46 7f ff 0e 10 d8 : crc=d8 YES
a2 01 4b 46 7f ff 0e 10 d8 t=26125
pi@raspberrypi /sys/bus/w1/devices/28-000003cee4ca $
```

Şekil 4.9 Örnek w1_slave dosyası içeriği

Tüm sensörlerden veri okumaya ilişkin akış diyagramını şekil 4.10'da gösterilmiştir.



Şekil 4.10 Sensörlerden veri okuma uygulaması akış diyagramı

4.6 Linux Servisi

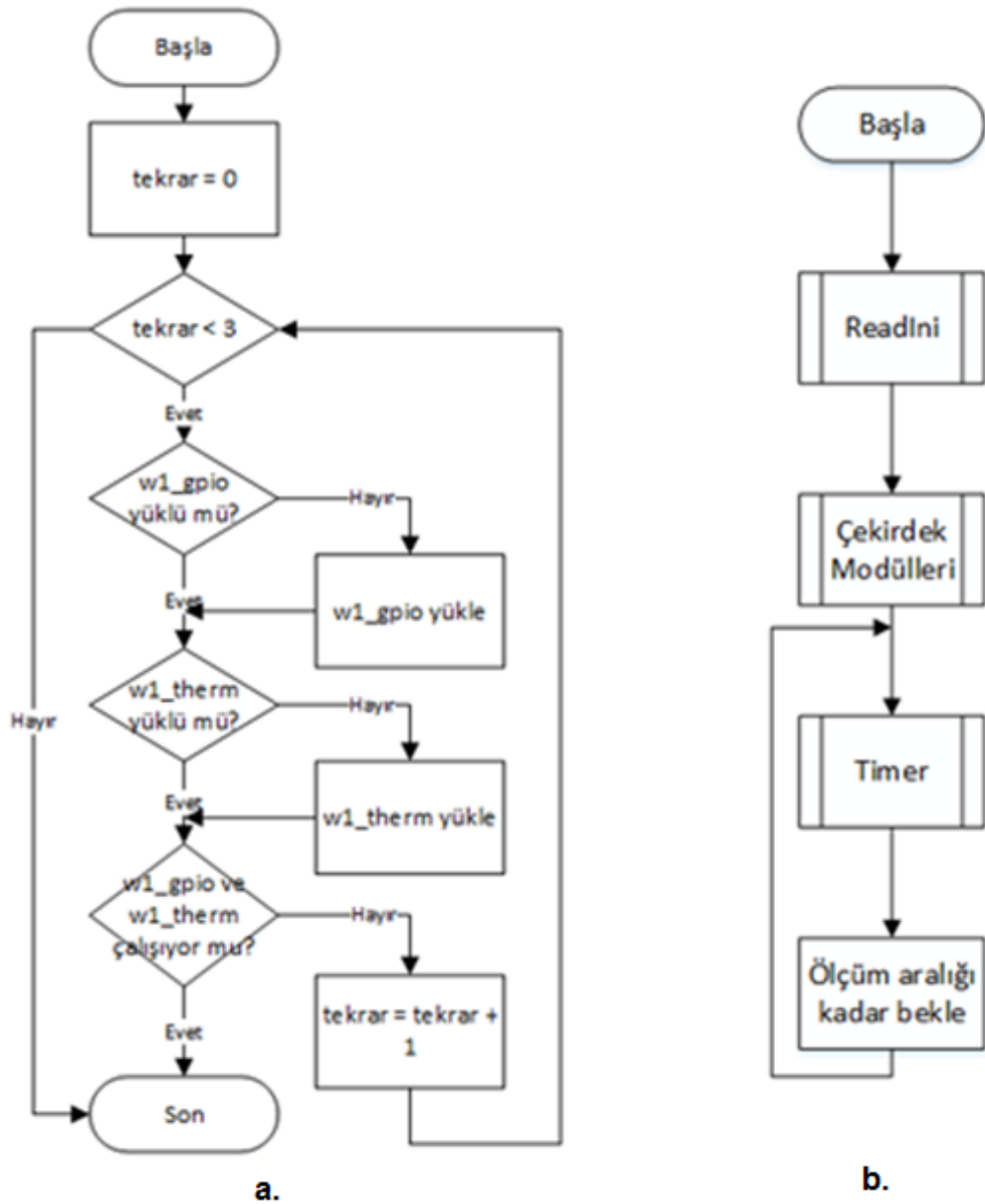
Tez kapsamında hazırlanan Linux servisi, istasyonun en önemli bileşenidir. Servis çekirdek modüllerini hazırlama, belirlenen zaman aralıklarında sensörlerden ölçüm alma ve bu ölçümleri yerel veya uzak sunucuya gönderme görevini üstlenmektedir.

Servis uygulaması, işletim sisteminin açılması ile başlar ve istasyon üzerinde yönetici haklarına sahip olan kişi tarafından kapatılıncaya ya da istasyonun gücü kesilinceye kadar çalışmasını sürdürür.

Linux servisi sırasıyla aşağıdaki işlemleri yapacak şekilde tasarlanmıştır (Şekil 4.11.b):

- 1) etc/rth/config.ini dosyasını okuyarak gerekli mevcut istasyon adı, uzak sunucu adresi, API anahtarı ve ölçümlerin kaç saniyede bir tekrarlanacağı bilgilerini alır.

- 2) w1_gpio ve w1_therm çekirdek modülleri yüklenir; test edilir (Şekil 4.11.a).
- 3) Ölçümler arası süreyi bekleyecek bir zamanlayıcı oluşturulur. Sensör verileri okunur ve JSON tipi çıktı elde edilir.
- 4) JSON tipindeki çıktı, api anahtarı ve servisin çalıştığı istasyonun ismi ile birlikte tekrar JSON tipinde sarmalanarak, config.ini dosyasında belirtilen uzak sunucunun ağ servisine gönderilir.



Şekil 4.11 a. Çekirdek modülü yükleme, b. Servis akış diyagramı

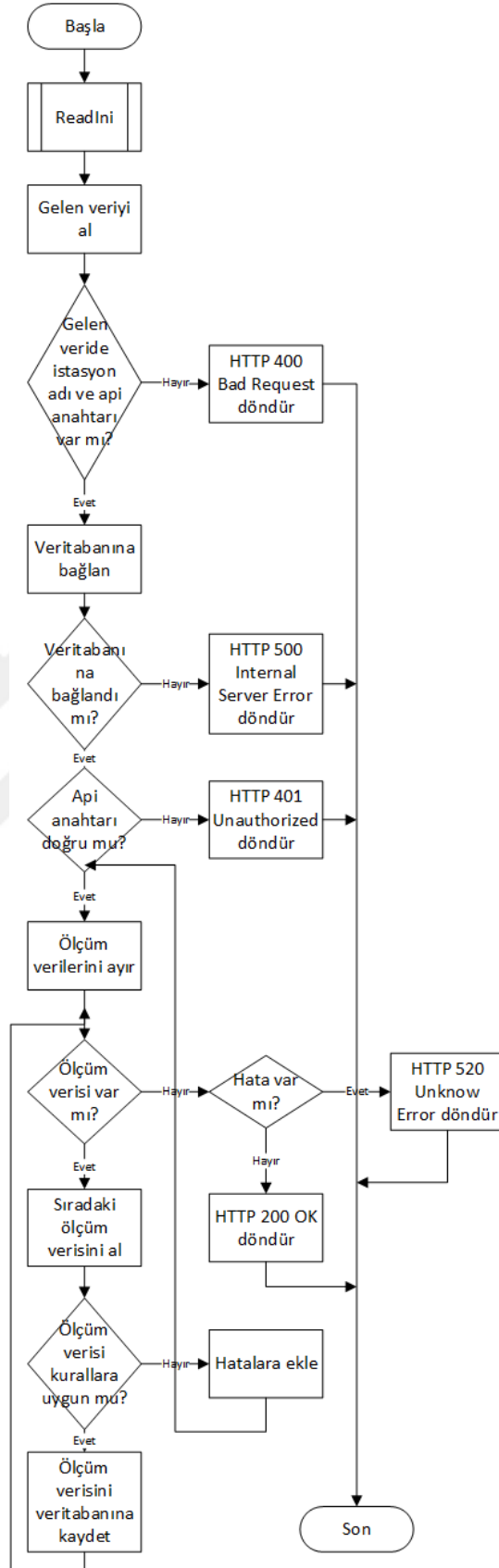
Servis yazılımını Mono C# derleyicisi kullanılarak derlenmiş ve oluşan uygulama dosyası “service.exe” olarak /usr/local/RTHService klasörü altına yerleştirilmiştir.

Servis yazılımının, işletim sistemi çalıştığında başlaması ve arka planda sürekli çalışması için /etc/init.d klasörü içine rth adında bir kabuk betiği hazırlanmıştır. Hazırlanan kabuk betiği aynı zamanda istasyonun konfigürasyon değişikliklerinde yeniden başlatılması için gereken start/restart ve stop komutlarını sağlamaktadır.

4.7 Ağ Servisinin Yazılması

Ağ servisi, Linux servisi-intranet-veritabanı sunucusu arasında köprü kurarak, sensörlerden okunan verinin veri tabanında depolanmasını sağlamaktadır.

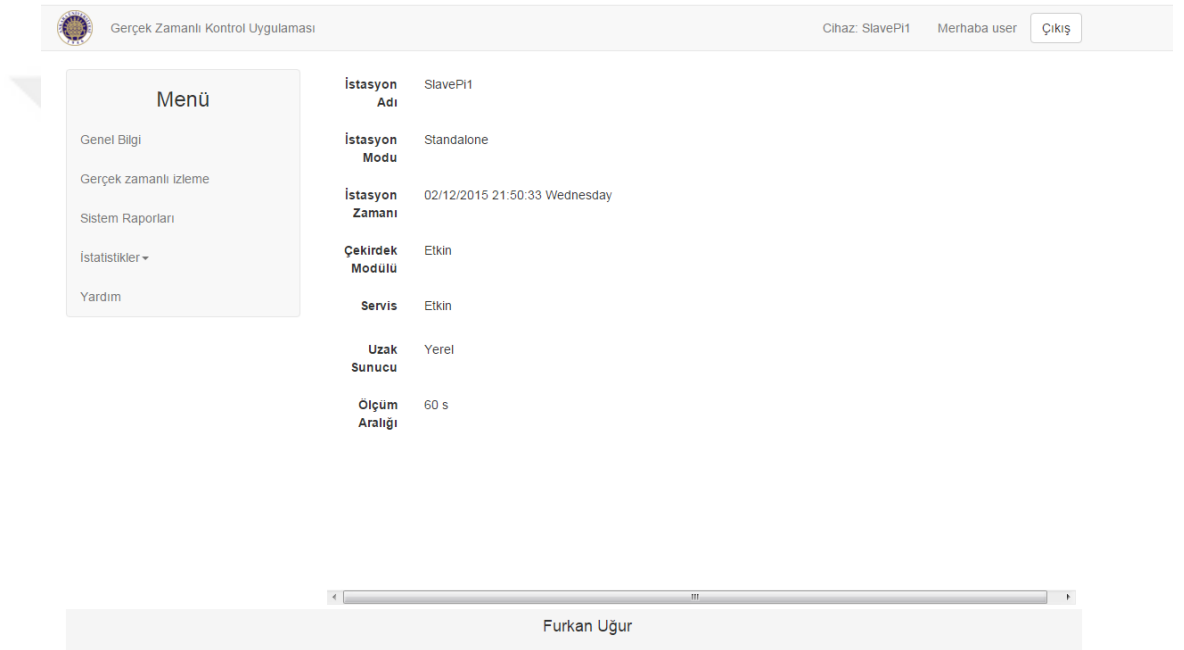
Sistemin yetkisiz erişimden korunması için ağ servisi, kendisine gelen veri paketi ile yüklü olduğu ölçüm istasyonundaki API anahtarını karşılaştırmaktadır. API anahtarının uyuşması durumunda gelen veri paketi içindeki verileri veritabanına kaydetmektedir. Ağ servisinin akış diagramı şekil 4.12’de verilmiştir.



Şekil 4.12 Ağ servisinin akış diagramı

4.8 Kullanıcı Arayüzü Tasarımı

Kullanıcı arayüzü, Html5, PHP, JS ve CSS ile hazırlanmıştır. Kullanıcı arayüzü sayesinde, sensör düğümünün kurulumu ve uzaktan izlenmesi sağlanmaktadır (Şekil 4.13). Arayüz, Bootstrap 3 CSS çatısının yardımıyla geliştirilerek kullanıcı ekranı boyutundan bağımsızlığı gözetilmiştir. Modern bir ağ tarayıcıya ve sensör düğümü giriş bilgilerine sahip herhangi biri, cihaz üzerinden ölçüm sonuçlarına erişebilmekle birlikte cihazın konfigürasyonunu da değiştirebilir.



Şekil 4.13 Kullanıcı arayüzü genel bilgi sayfası

Yetkisiz erişimlerden sistemi korumak için kullanıcı arayüzüne şifre ile oturum açma özelliği eklenmiştir.

Kullanıcı arayüzünde iki farklı rol tanımlanmıştır. İlk rol, kullanıcının sadece ölçüm istasyonunu izlediği “Kullanıcı” rolüdür. Bu rol ile oturum açan kişinin yetkisi, istasyonun genel bilgilerine göz atmak ve ölçüm sonuçlarını izleyebilmek olarak kısıtlanmıştır.

Kullanıcı arayüzündeki ikinci rol ise “Yöneticidir.” Yöneticisi şifresi ile sensör düğümü kullanıcı arayüzünde oturum açan kişi, kullanıcının yapabildiği tüm eylemleri gerçekleştirebilmenin yanı sıra sensör düğümünün ayarlarını da değiştirebilmektedir.



5. ARAŞTIRMA BULGULARI

Çalışmada geliştirilen kablosuz sensör düğümleri laboratuvar ve sera ortamında farklı çalışma senaryoları ve ağ geçitleri ile denenmiştir. Çalışılan ortama ve yapılandırmaya göre kablosuz sensör düğümleri tekrar isimlendirilmiştir.

5.1 Sistem Maliyeti

Dört adet sayısal sıcaklık sensörüne sahip düğümün birim maliyeti yaklaşık olarak KDV hariç 71 \$ olarak hesaplanmıştır. Düğüm için maliyet analizi çizelge 5.1’de verilmiştir.

Çizelge 5.1 Kablosuz sensör düğümünün maliyet analizi

Parça	Birim Fiyatı (\$)	Adet	Toplam (\$)
Tek kartta bilgisayar	35,00	1	35
Kablosuz adaptörü	10,13	1	10,13
Sayısal Sıcaklık Sensörü	1,40	4	5,6
Saklama kutusu	3,50	1	3,5
Güç adaptörü	8,00	1	8
MicroSD depolama birimi	7,53	1	7,53
Pasif soğutucu	1,10	1	1,1
TOPLAM			70,86

5.2 Prototip Sistemin Test Bulguları

Sistem farklı test senaryoları esas alınarak laboratuvar ortamı ve kıvrıcık serasında denenmiştir. Testlere ait yapılandırma bilgileri çizelge 5.2’de verilmiştir.

Çizelge 5.2 Testlere ait yapılandırma bilgileri

Test Ortamı	Laboratuvar		Sera
Düğüm 1	Ad	MasterPi	SeraPi
	Mod	Master	Standalone
	IP	192.168.1.34	192.168.1.55
	Sensör	28-000000c30cb0 (Hava)	28-000000c30cb0 (Hava)
		28-0000010f489 (Toprak)	28-0000010f390b (Hava) 28-00044a3880ff (Toprak)
Düğüm 2	Ad	SlavePi	-
	Mod	Slave	
	IP	192.168.1.55	
	Sensör	28-0000010f390b (Hava)	
		28-00044a3880ff (Su)	
Ölçüm aralığı	60 s		
Çalışma Süresi	4 gün	1+ gün	

5.2.1 Laboratuvar denemesi

Laboratuvar ortamında master ve slave çalışma moduna ayarlanmış iki adet düğüm kullanılarak birden fazla düğümün kullanıldığı senaryo test edilmiştir. Test sırasında master modundaki düğüm “MasterPi”, slave modundaki düğüm ise “SlavePi” olarak isimlendirilmiş ve kablosuz ağ geçidinden IP adresleri sırasıyla 192.168.1.34 ve 192.168.1.55 olarak sabitlenmiştir.

Düğümlerin bulunduğu yerel ağa bağlı bilgisayardan <http://192.168.1.34> adresine ağ tarayıcı ile bağlanılarak master modunda çalışacak düğüm ayarlanmıştır. Düğümün ağ arayüzünde yönetici olarak giriş yapılmıştır. Sol taraftaki menüden sistem ayarlarından genel ayarlar bölümüne girilmiştir. Düğüm adı ve çalışma modu şekil 5.1’de gösterildiği gibi ayarlanmış ve kaydet tuşuna basılmıştır.

Web Tabanlı Sera İzleme Sistemi Cihaz: MasterPi Merhaba Yönetici Çıkış

Menü

- Genel Bilgi
- Gerçek zamanlı izleme
- Sistem Raporları
- Sistem ayarları ▾
 - Genel Ayarlar
 - Sensor Ayarları
 - VT Ayarları
 - Master Ayarları
 - Parola Ayarları

İstasyon Adı MasterPi

Çalışma Modu Master Standalone Slave

Kaydet

Servis

Furkan Uğur

Şekil 5.1 Master düğümünün ad ve çalışma modu ayarı

Daha sonra sistem ayarları bölümünden master ayarlarına girilerek api anahtarı “1234567890” olarak değiştirilip kaydet tuşuna basılmıştır (Şekil 5.2). Bu yapılan ayarlardan sonra düğüm ana sunucu olarak yapılandırılmış, slave modundaki düğümlerden veri almaya hazır hale gelmiştir.

Web Tabanlı Sera İzleme Sistemi Cihaz: MasterPi Merhaba Yönetici Çıkış

Menü

- Genel Bilgi
- Gerçek zamanlı izleme
- Sistem Raporları
- Sistem ayarları ▾
 - Genel Ayarlar
 - Sensor Ayarları
 - VT Ayarları
 - Master Ayarları
 - Parola Ayarları

Master Api Key 1234567890

Kaydet

Furkan Uğur

Şekil 5.2 Master düğümünün api anahtarı ayarı

Slave düğümün ayarlanması için düğümün ip adresi olan http://192.168.1.55 adresine ağ tarayıcıyla bağlanılmış ve ağ arayüzünde yönetici olarak oturum açılmıştır. Master düğümünün ayarlanmasına benzer şekilde sistem ayarları menüsünün genel ayarlar kısmından düğüm adı ayarlanmış ve slave çalışma modu seçilmiştir (Şekil 5.3).

The screenshot displays the 'Web Tabanlı Sera İzleme Sistemi' (Web-based Plant Monitoring System) interface. The top navigation bar includes the system logo, the title 'Web Tabanlı Sera İzleme Sistemi', the device name 'Cihaz: SlavePI', the user name 'Merhaba Yönetici', and a 'Çıkış' (Logout) button. A 'Menü' (Menu) sidebar on the left lists options: Genel Bilgi, Gerçek zamanlı izleme, Sistem Raporları, Sistem ayarları (selected), Genel Ayarlar, Remote Ayarları, Sensor Ayarları, and Parola Ayarları. The main content area shows the 'İstasyon Adı' (Station Name) field set to 'SlavePI'. Below it, the 'Çalışma Modu' (Operating Mode) is set to 'Slave' (indicated by a selected radio button). A 'Kaydet' (Save) button is positioned below the mode selection. The 'Servis' (Service) section contains three buttons: 'Başlat' (Start), 'Durdur' (Stop), and 'Yeniden Başlat' (Restart). The footer of the interface displays the name 'Furkan Uğur'.

Şekil 5.3 Slave düğümünün ad ve çalışma modu ayarı

Düğüm adı ve çalışma modu ayarları yapıldıktan sonra remote ayarları bölümüne girilerek remote address kısmına master düğümün IP adresi, Remote Api Key kısmına ise master'a girilen api anahtarı girilmiştir (Şekil 5.4).

Web Tabanlı Sera İzleme Sistemi Cihaz: SlavePi Merhaba Yönetici ÇIKIŞ

Menü

- Genel Bilgi
- Gerçek zamanlı izleme
- Sistem Raporları
- Sistem ayarları ▾
 - Genel Ayarlar
 - Remote Ayarları
 - Sensor Ayarları
 - Parola Ayarları

Remote Address: 192.168.1.34

Remote Api Key: 1234567890

Kaydet

Furkan Uğur

Şekil 5.4 Slave düğümü remote ayarları

Son olarak master ve slave düğümleri için sensör ayarları, sensör ayarları kısmından şekil 5.5’te gösterildiği gibi 60 s olarak değiştirilmiştir.

Menü

- Genel Bilgi
- Gerçek zamanlı izleme
- Sistem Raporları
- Sistem ayarları ▾
 - Genel Ayarlar
 - Remote Ayarları
 - Sensor Ayarları
 - Parola Ayarları

Sensor Interval: 60 (s)

Kaydet

Furkan Uğur

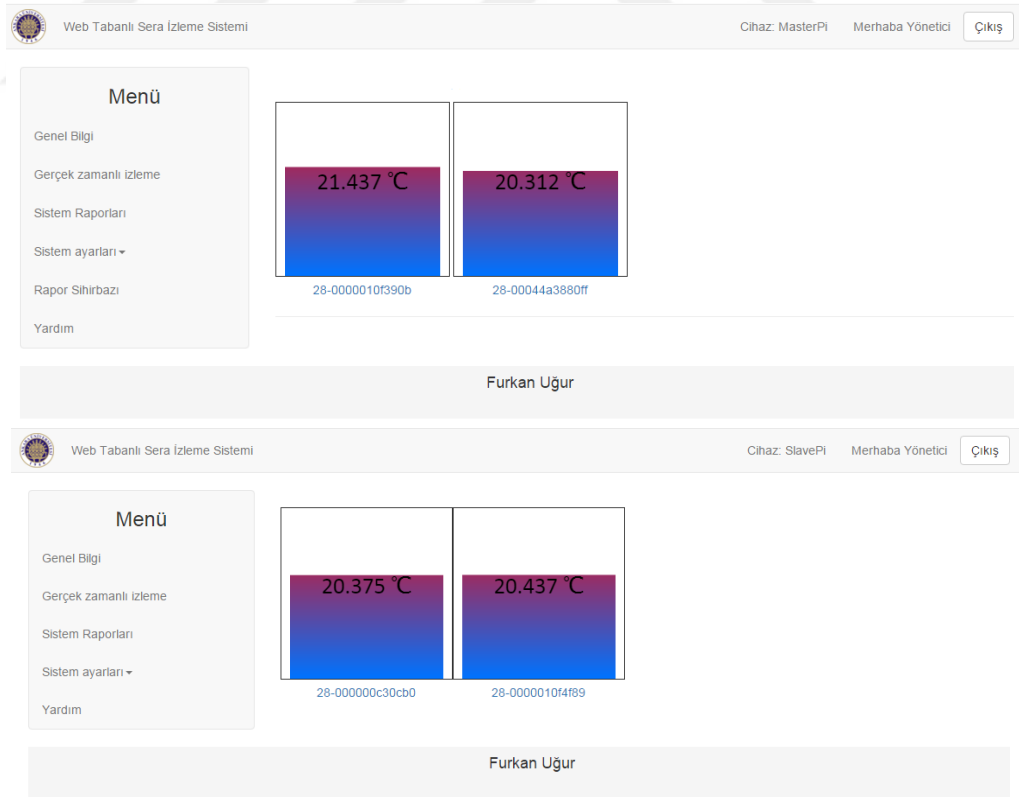
Şekil 5.5 Düğümlerin sensör ayarları

Konfigürasyonu tamamlanan düğümler laboratuvar ortamına yerleştirilerek sistem kullanılmaya hazır hale getirilmiştir (Şekil 5.6).



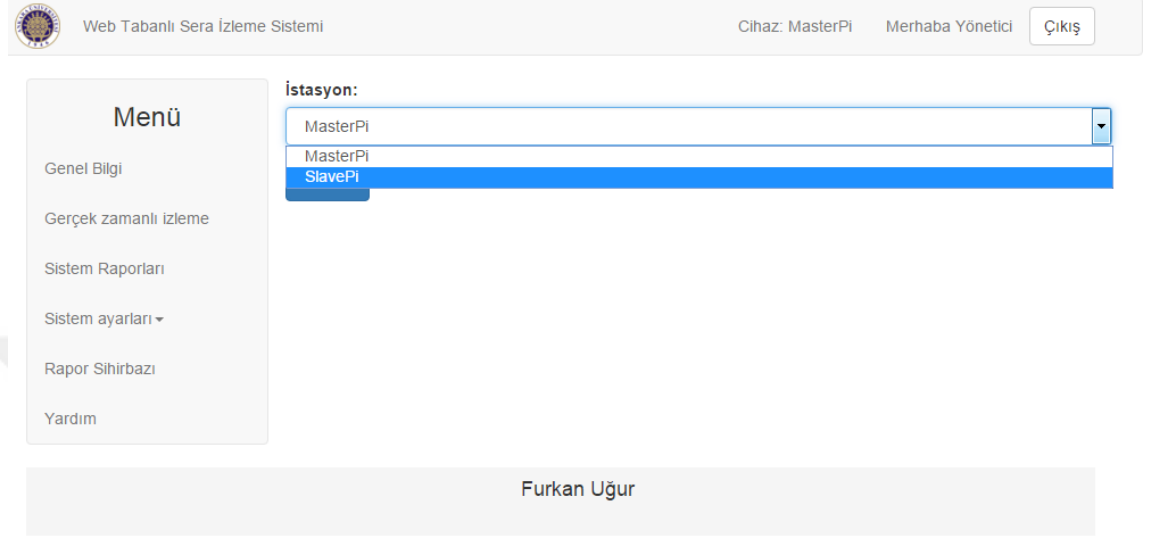
Şekil 5.6 MasterPi ve SlavePi düğümleri

Laboratuvar testi sırasında sıcaklıklar, düğümlerin IP adreslerine ağ tarayıcı ile bağlanılarak izlenmiştir (Şekil 5.7). Sistem dört gün süresince çalıştırılmıştır.



Şekil 5.7 Laboratuvar testi sırasında sıcaklık ölçümlerinin izlenmesi

Sistemin çalıştığı dört günün sonunda master düğümüne ağ tarayıcı ile bağlanılarak ağ arayüzündeki rapor sihirbazıyla sıcaklık ölçüm verileri MS Excel dosyası olarak çıktı alınmıştır (Şekil 5.8).



Şekil 5.8 Master düğümünde rapor sihirbazının kullanılması

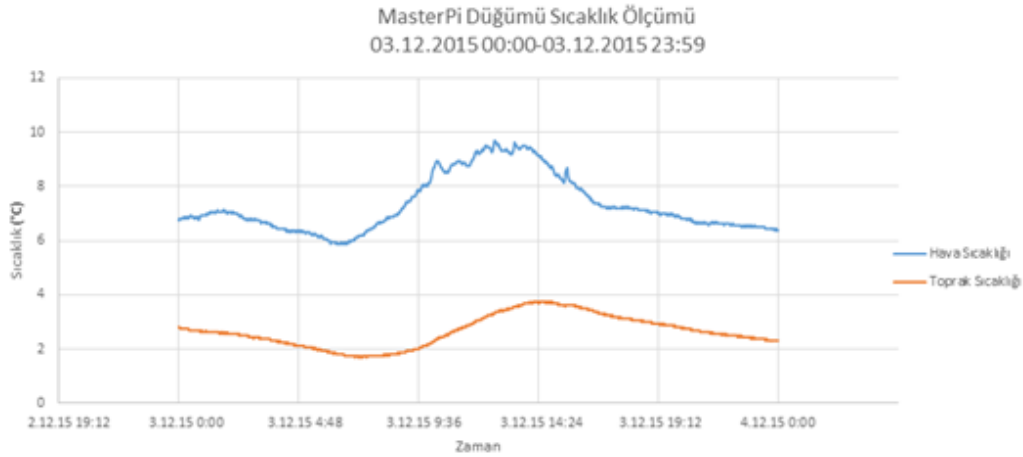
Laboratuvar ölçümleri sırasında en düşük hava sıcaklığı 6.12.2015 06:33 tarihinde 4,062 °C, en yüksek hava sıcaklığı 6.12.2015 14:50 tarihinde 13,5 °C olarak ölçülmüştür.

Toprak, su ve hava sensörleri kullanılarak yapılan laboratuvar denemesi sonucunda elde edilen sıcaklık verilerinin istatistikleri çizelge 5.3'te verilmiştir.

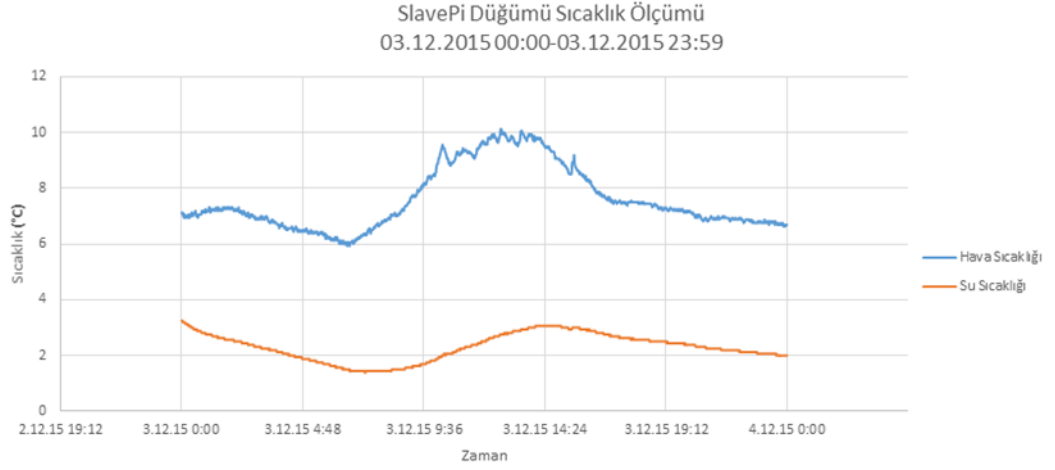
Çizelge 5.3 Laboratuvar ölçüm istatistikleri

Tarih	Cihaz	Ort. Hava – Su/Toprak	Min. Hava (Saat)	Maks. Hava (Saat)	Min. Toprak /Su (Saat)	Maks. Toprak /Su (Saat)
3.12.2015	MasterPi	7,324 °C – 2,660 °C	5,875 °C 06:04	9,687 °C 12:38	1,687 °C 07:03	3,750 °C 14:03
	SlavePi	7,602 °C – 2,302 °C	5,937 °C 06:32	10,125 °C 12:38	1,375 °C 07:16	3,25 °C 00:02
4.12.2015	MasterPi	7,396 °C – 2,645 °C	5,875 °C 23:52	10,937 °C 15:05	1,562 °C 07:32	4,375 °C 15:44
	SlavePi	7,737 °C – 2,242 °C	6,187 °C 23:48	12,062 °C 15:06	1,312 °C 07:07	3,75 °C 15:32
5.12.2015	MasterPi	6,565 °C – 1,908 °C	4,187 °C 06:31	11,562 °C 14:33	0,125 °C 07:39	4,625 °C 16:08
	SlavePi	6,977 °C – 0,629 °C	4,25 °C 07:01	12,687 °C 14:39	0 °C 06:52	3 °C 15:35
6.12.2015	MasterPi	6,645 °C – 1,831 °C	3,875 °C 06:50	12,375 °C 14:50	-0,187 °C 07:58	4,187 °C 15:34
	SlavePi	7,050 °C – 0,071 °C	4,062 °C 06:33	13,5 °C 14:50	-0,312 °C 04:48	0,812 °C 23:03

3.12.2015 tarihinde ortalama hava sıcaklığı 7,463 °C, su sıcaklığı 2,302 °C, toprak sıcaklığı 2,660 °C olarak ölçülmüştür. En yüksek hava sıcaklığı saat 12:38’de 10,125 °C olarak ölçülürken en düşük sıcaklık ölçümü 06:04’te gerçekleşmiştir. 3.12.2015 tarihinde düğümlerin ölçüm grafikleri şekil 5.9 ve şekil 5.10’da verilmiştir. Toprak, su ve hava sıcaklıkları arasında paralel değişimler olduğu bulunmuştur.

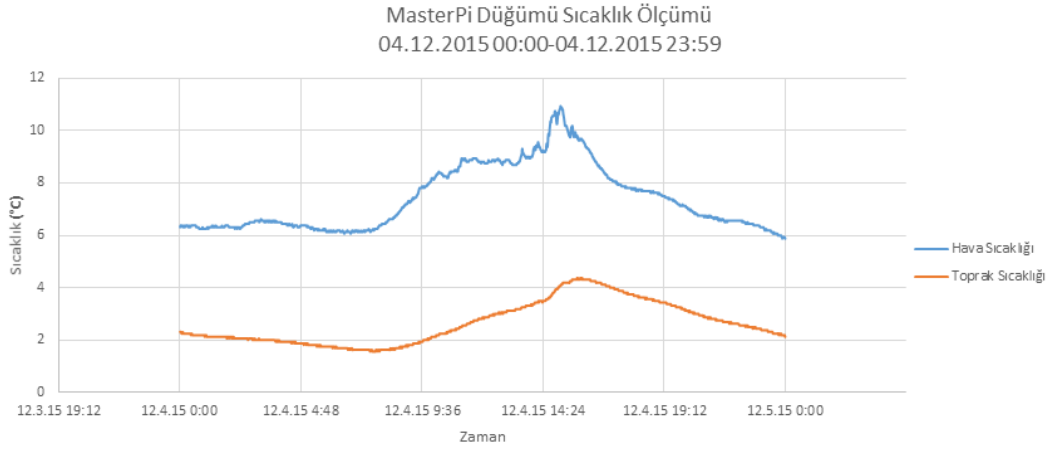


Şekil 5.9 MasterPi düğümü 3.12.2015 tarihli sıcaklık grafiği



Şekil 5.10 SlavePi düğümü 3.12.2015 tarihli sıcaklık grafiği

4.12.2015 tarihinde ortalama hava sıcaklığı 7,567 °C, toprak sıcaklığı 2,645 °C ve su sıcaklığı 2,242 °C olarak hesaplanmıştır. En düşük hava sıcaklığı saat 23:52’de 5,875 °C ile gerçekleşirken en yüksek sıcaklık 12,062 °C olarak saat 15:05’te ölçülmüştür (Şekil 5.11 ve Şekil 5.12). 4.12.2015 tarihinde toprak, su ve hava sıcaklığı arasında paralel değişimler gözlemlenmiştir.

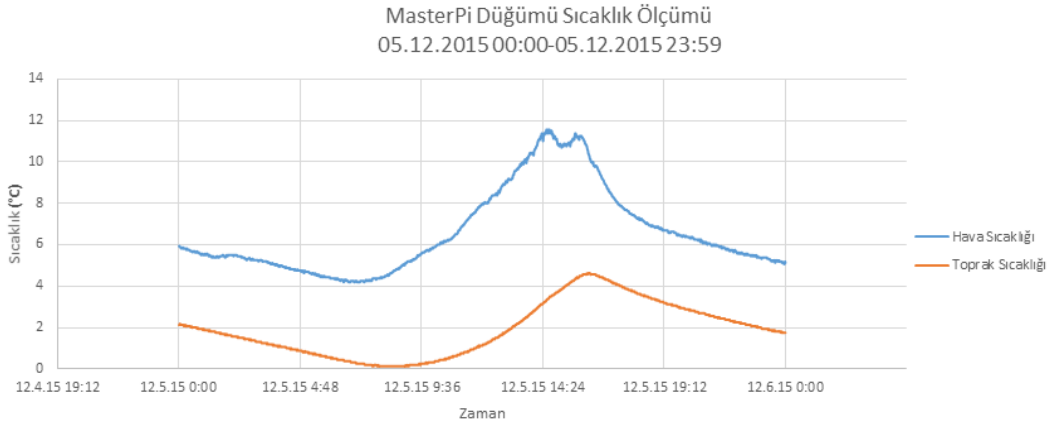


Şekil 5.11 MasterPi düğümü 4.12.2015 tarihli sıcaklık grafiği

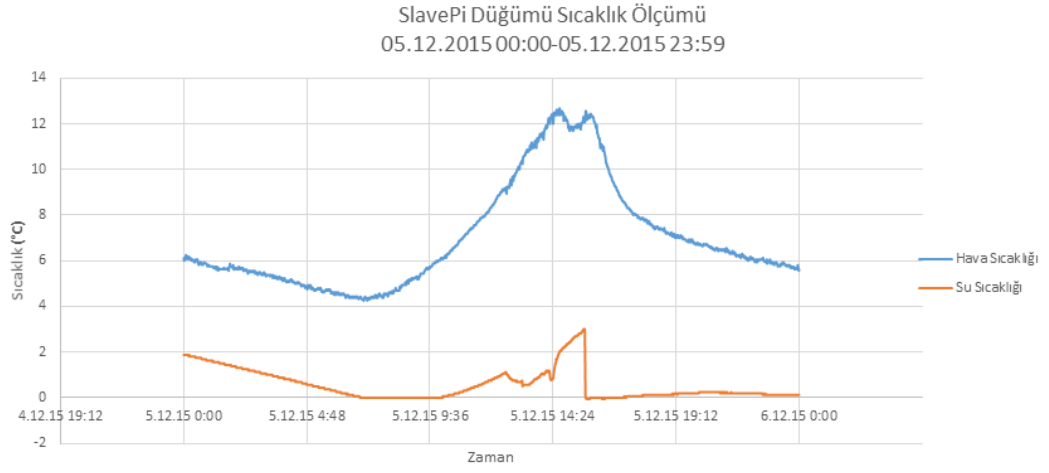


Şekil 5.12 SlavePi düğümü 4.12.2015 tarihli sıcaklık grafiği

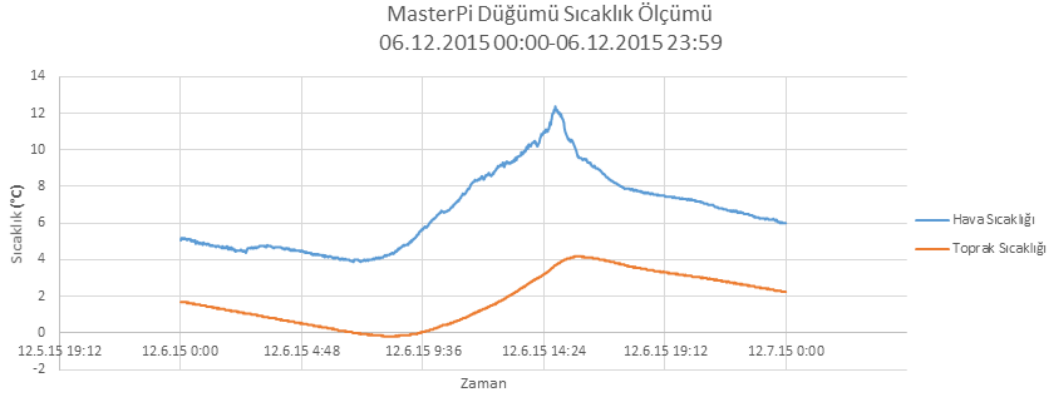
Sistem kararlılığının test edilmesi amacıyla 5.12.2015 tarihinde saat 15:00'da su sıcaklığı 0 °C'nin altına düşürülmüş ve sistem iki gün çalıştırılmaya devam ettirilmiştir (Şekil 5.14). Sistem kararlılığı testi sürecinde ortalama hava sıcaklığı 6,809 °C, toprak sıcaklığı ise 1,870 °C olarak hesaplanmıştır. En düşük hava sıcaklığı 6.12.2015 tarihinde 06:50'de gerçekleşirken en yüksek hava sıcaklığı ise 6.12.2015 tarihinde saat 14:40'de 13,500 °C ile gerçekleşmiştir. Kararlılık testi süresince toplanan sıcaklık verilerinin grafikleri şekil 5.13 – 5.16'da verilmiştir.



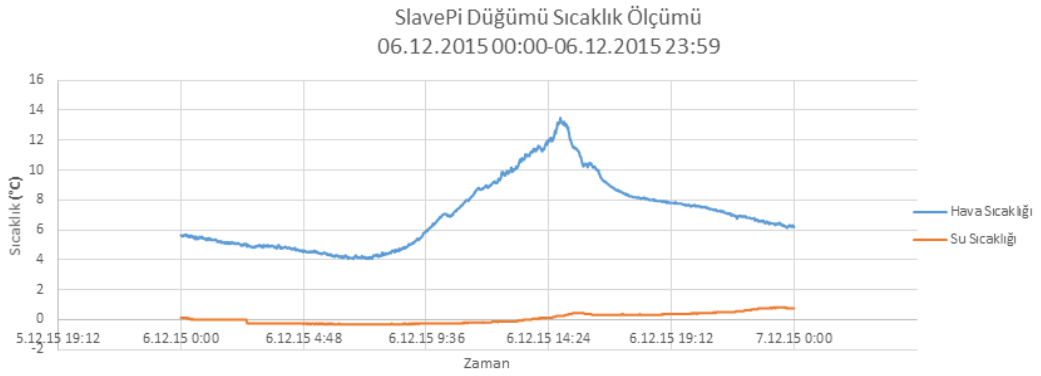
Şekil 5.13 MasterPi düğümü 5.12.2015 tarihli sıcaklık grafiği



Şekil 5.14 SlavePi düğümü 5.12.2015 tarihli sıcaklık grafiği



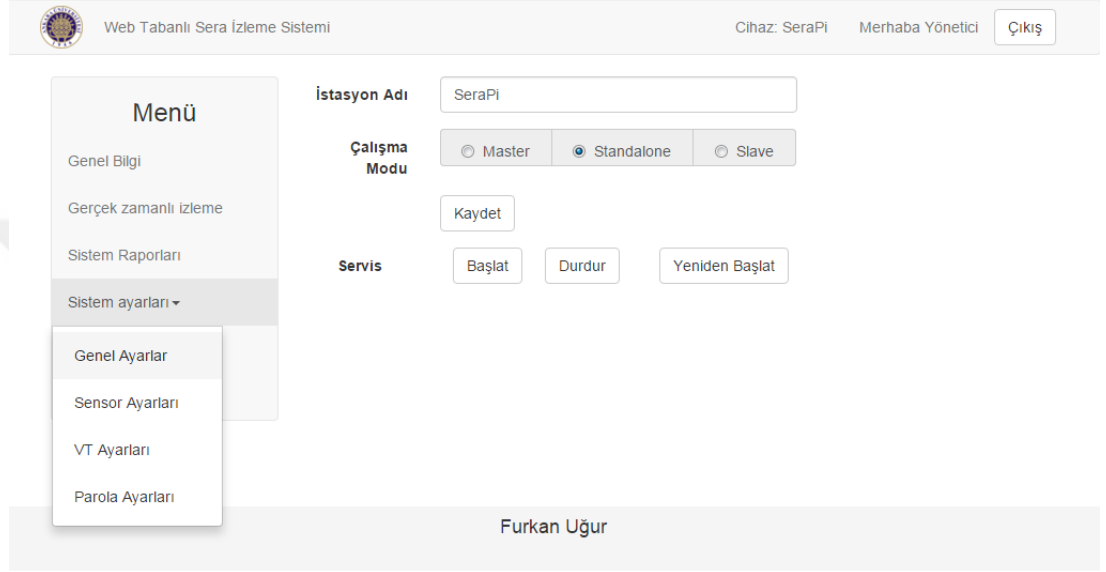
Şekil 5.15 MasterPi düğümü 6.12.2015 tarihli sıcaklık grafiği



Şekil 5.16 SlavePi düğümü 6.12.2015 tarihli sıcaklık grafiği

5.2.2 Sera denemesi

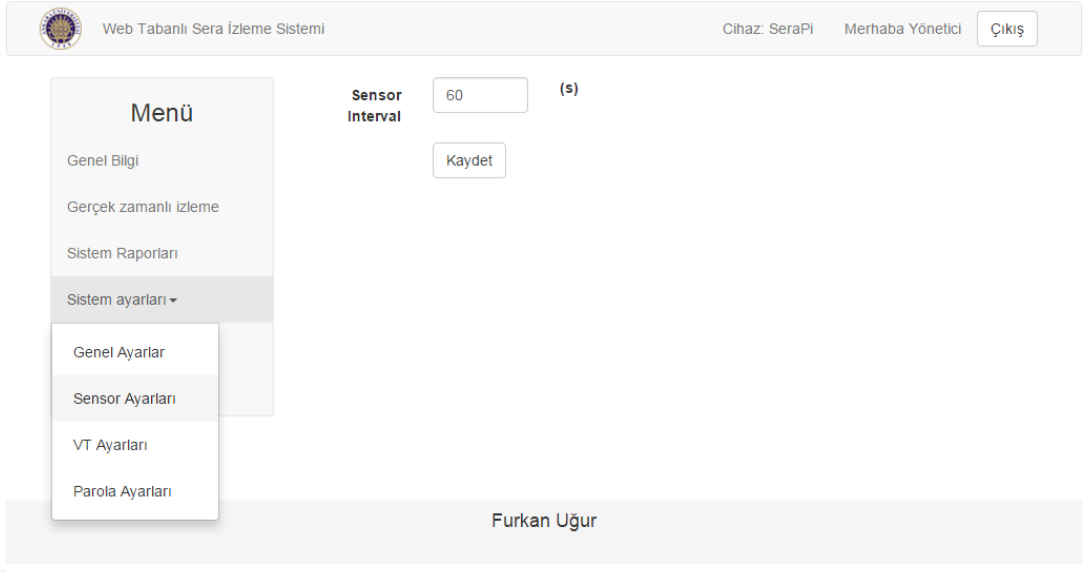
Sera ortamında bir düğüm kullanılarak bir düğümlü çalışma senaryosu test edilmiştir. Düğüm, http://192.168.1.55 adresindeki ağ arayüzünden “SeraPi” olarak adlandırılmış ve çalışma modu “Standalone” olarak ayarlanmıştır (Şekil 5.17).



The screenshot displays the 'Web Tabanlı Sera İzleme Sistemi' interface. At the top, it shows 'Cihaz: SeraPi' and 'Merhaba Yönetici' with a 'Çıkış' button. A 'Menü' sidebar on the left includes options like 'Genel Bilgi', 'Gerçek zamanlı izleme', 'Sistem Raporları', 'Sistem ayarları', 'Genel Ayarlar', 'Sensor Ayarları', 'VT Ayarları', and 'Parola Ayarları'. The main configuration area includes: 'İstasyon Adı' (Station Name) set to 'SeraPi'; 'Çalışma Modu' (Operating Mode) with radio buttons for 'Master', 'Standalone' (selected), and 'Slave'; a 'Kaydet' (Save) button; and 'Servis' (Service) buttons for 'Başlat' (Start), 'Durdur' (Stop), and 'Yeniden Başlat' (Restart). The footer identifies the user as 'Furkan Uğur'.

Şekil 5.17 SeraPi düğümünün adı ve çalışma modu ayarı

Ağ arayüzünden sensör ölçüm değeri 60 s olarak ayarlanmıştır (Şekil 5.18).



Şekil 5.18 SeraPi sensor ayarları

Konfigürasyonu tamamlanan düğüm şekil 5.19’da gösterilen sera içine şekil 5.20’deki şekilde kurularak sistem hazır hale getirilmiştir.

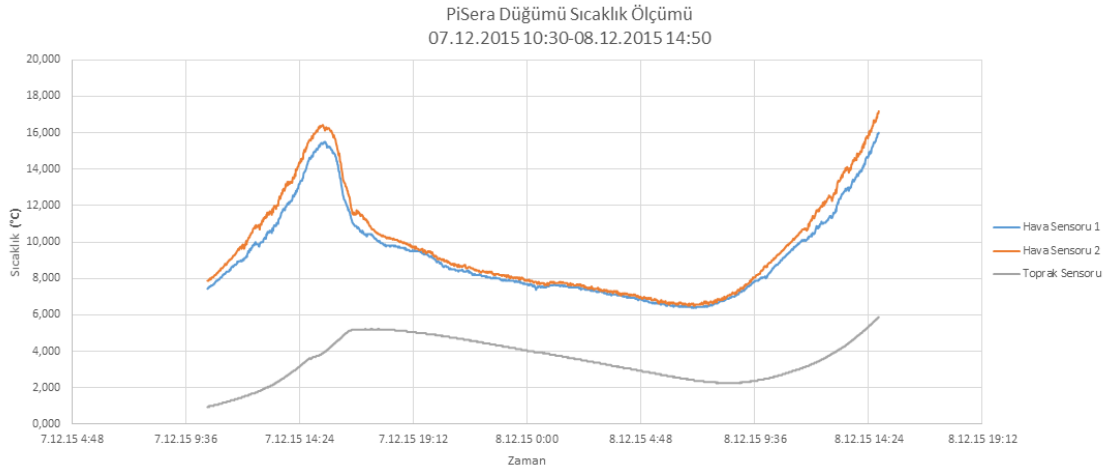


Şekil 5.19 Denemenin yapıldığı sera



Şekil 5.20 Serada kablosuz sensör düğümünün kurulumu

Sera denemesi sonrasında ağ arayüzündeki rapor sihirbazı kullanılarak veriler MS Excel çıktısı olarak alınmıştır. Denemeye ilişkin sıcaklık değişim grafiği şekil 5.21’de verilmiştir. Sera sıcaklık ölçümleri ile ilgili istatistikler çizelge 5.4’te verilmiştir.



Şekil 5.21 SeraPi düğümü sıcaklık grafiği

Çizelge 5.4 Sera ölçüm istatistikleri

No	Sensör Adı	Sensör ID	Ortalama (°C)	En düşük (°C)	En yüksek (°C)
1	Hava Sensörü 1	28-000000c30cb0	+9,244	+6,375	+16
2	Hava Sensörü 2	28-0000010f390b	+9,713	+6,5	+17,187
3	Toprak Sensörü	28-00044a3880ff	+3,506	+0,937	+5,875

Sera içinde en yüksek hava sıcaklıkları 8.12.2015 tarihinde saat 14:49'da 16 ve 17,18 °C olarak, toprak sıcaklığı 5,875°C olarak ölçülmüştür. En düşük hava sıcaklıkları saat 06:56'da (8.12.2015) gerçekleşmiş olup 6,375 ve 6,5 °C toprak sıcaklığı 2,437 °C olarak ölçülmüştür.

Sistemin çalıştığı 1580 dakika boyunca toplamda 4.740 ölçüm yapılmıştır. Denemelerde ortalama hava sıcaklığı 9,479 °C, ortalama toprak sıcaklığı 3,506 °C olarak hesaplanmıştır.

Deneme sonucunda hava sıcaklığı artışı ile toprak sıcaklığı arasındaki değişimin paralel olduğu gözlemlenmiştir.

6. TARTIŞMA VE SONUÇ

Bu çalışmada, sera sıcaklığının ölçülmesi için düşük maliyetli sistem geliştirilmiş olup; test deneyleri yapılmıştır. Geliştirilen sistem düğümlerinde ARM tabanlı tek kartta bilgisayar ve gömülü ağ sunucusu ile ilerideki gelişmelere açık bir yapı oluşturulmuştur.

Sistem, önceki çalışmalardan farklı olarak tek bir düğümün hem kablosuz sensör düğümü hem de sunucu bilgisayarı olarak yapılandırılmasına olanak vermektedir. Bu özelliğe sahip kablosuz sensör düğümleri internetin bir parçası olmakta ve IoT düğümü kapsamına girmektedir.

Sistemin internet üzerinden kullanılabilmesi için internet bağlantısı gerekmekte, bunun sonucunda ise işletme maliyeti artmaktadır. Düğümlerdeki gömülü ağ sunucusu nedeniyle sistem intranet üzerinden, internet bağlantısı olmadan da kullanılabilmesi diğer sistemlere göre avantaj sağlamaktadır.

Bluetooth gibi diğer kablosuz bağlantı standartları ile karşılaştırıldığında Wifi yaygın olması, daha geniş kapsama alanına sahip olması ve kullanılacak düğüm sayısını pratik olarak sınırlamaması dolayısıyla sistemin esnekliğini artmıştır. Wifi ayrıca kişisel bilgisayarların, akıllı mobil telefonlar ve tabletlerin de kablosuz sensör ağına bağlanabilmesini olanak vererek kullanıcıya herhangi bir cihazdan izleme avantajı sağlamıştır.

Düğümlerin tasarımında “kur ve unut” yaklaşımı izlendiğinden yeni kablosuz sensör ağının kurulması ve mevcut ağlarda düğüm ekleme, çıkarma ve değiştirme işlemleri kolaylaşmıştır. Düğümlerin çalışma ayarlarını değiştirmek için özel programlayıcılarla düğümün sistem yazılımını değiştirmeye gerek duyulmamakta, bu iş için her bir düğümün sahip olduğu ağ arayüzünden yararlanılmaktadır. Başka bir deyişle kurulum için düğümün bir sefere mahsus olmak üzere monitör ve klavyeye bağlanarak kablosuz ağa tanıtılması yeterli olmaktadır.

Geliştirilen düğümlerde sensörler ile iletişim 1-wire protokolü ile sağlandığından dolayı aynı düğüme teorik olarak 100 adet sayısal 1-wire sensörü bağlanabilir. 5 sayısal sensörün bağlı olduğu durumda ise düğüm ile en uzak sensörün uzaklığı 164 m'ye kadar çıkabilir.

Sistemin veri iletişim standardı esnek olduğu için wifi ağına bağlanan ve JSON paketi standartlarına uyan başka kablosuz sensör düğümleri de sisteme dâhil edilebilir.

Kablosuz sensör düğümünün çalışması için elektrik enerjisine ihtiyaç vardır. Elektrik enerjisi ihtiyacı sisteme bağlanacak güneş panelleri ve akülerle çözülebilir.

Düğümlerin ortamın nem ve sıcaklık faktörlerinden etkilenmemesi için toza ve suya dayanımlı muhafaza kutusu içine alınmalıdır. Muhafaza kutusu en az IP 65 standardını desteklemelidir.

Raspbian GNU/Linux dağıtımı, daha çok ev kullanıcılarını hedeflemekte bu yüzden sistemin çalışması ile doğrudan ilgili olmayan birçok bileşeni içermektedir. Raspbian dağıtımı yerine GNU/Linux çekirdeği ve sadece gerekli bileşenlerin bulunduğu bir dağıtım oluşturularak düğümlerin daha az ısınması ve güç tüketmesi sağlanacaktır.

Çalışma kapsamında geliştirilen yazılımlar PHP ve Mono'nun kullanımını gerekli kılmaktadır. Sonraki aşamada bu yazılımlar C++ dilinde tekrar yazılarak bağımlılıklar azaltılacaktır.

Apache ağ sunucusu ve MySQL veritabanı yönetim yazılımları oldukça kapsamlı olmasının yanı sıra kablosuz sensör düğümlerinde kullanılmayan birçok ek özellik sunmaktadır. Apache ağ sunucusu yerine daha hafif bir ağ sunucu yazılımı ve MySQL yerine gömülü sistemler için geliştirilmiş SQLite veritabanı yazılımı kullanılarak düğümlerin performansının artırılması planlanmaktadır.

Kullanılan tek kartta bilgisayar olan Raspberry Pi'nin HDMI, ses bağlantısı gibi birçok özelliğine ihtiyaç duyulmamıştır. Sadece gereken özelliklerin olduğu bir tek kartta bilgisayar seçilerek maliyetin azaltılması sağlanacaktır.

Düğümlere e-posta ile sıcaklık düzey alarmı eklenerek, seranın iklimlendirme sistemlerine müdahale edilmesini gerektirecek durumlarda düğümlerin operatörü bilgilendirmesi sağlanacaktır.



KAYNAKLAR

- Aland, A. and Banhazi, T. 2013. Livestock housing. Wageningen Academic Publishers, 497 p., Netherlands.
- Anonim. 2001. Sera- Terimler ve Tarifleri. TSE TS 12741. Türk Standartları Enstitüsü, Ankara.
- Anonim. 2015. Türkiye İstatistik Kurumu, <http://www.tuik.gov.tr>
Erişim Tarihi: 10.12.2015.
- Anonymous. 2008. DS18B20 Programmable Resolution 1-Wire Digital Thermometer Datasheet. Maxim Integrated, 22 p., USA.
- Anonymous. 2015. <http://raspberrypi.org/products/raspberry-pi-2-model-b/>,
Erişim Tarihi: 11.12.2015.
- Ashton, K. 2009. That 'Internet of Things' Thing, RFID Journal, 22 June 2009.
- Ateş, İ. 2013. Control of mobile robots via internet and wireless communication. M. Sc. Thesis, Dokuz Eylül University, Graduate School of Natural and Applied Sciences, İzmir.
- Baytürk, M., Çetin, G. ve Çetin, A. 2013. Gömülü Sunucu ile Tasarlanmış İnternet Tabanlı. Bilişim Teknolojileri Dergisi, Cilt 6, Sayı 2, 53-58.
- Bertolotti, I., C., Hu, T., 2015. Modular design of an open-source, networked embedded system, Computer Standards & Interfaces 37 (2015) 41–52.
- Birajdar, G. 2014. Implementation of Embedded Web Server Based on ARM11 and Linux using Raspberry Pi. IJRTE, V3-3.
- Bozdoğan, Z. 2015. Nesnelerin interneti için mimari tasarım, Yüksek Lisans Tezi. Düzce Üniversitesi, Fen Bilimleri Enstitüsü, Düzce.
- Dayıoğlu, M.A. 2013. Seralar İçin Bluetooth Tabanlı Kablosuz Ölçüm Sisteminin Tasarımı Prototip Geliştirme ve Uygulama. Tarım Makinaları Bilimi Dergisi, 2013, 9(2), 117-125.
- Dayıoğlu, M.A. 2014a. Seralar İçin Gerçek Zamanlı Kablosuz Veri Takip Sisteminin Geliştirilmesi Endüstriyel Bluetooth Uygulaması. IEEE SIU2014.
- Dayıoğlu, M.A. 2014b. Ünite 6: Seralarda Bilişim ve Otomasyon Teknolojisi, 2. Baskı s: 102-134. Örtüaltı Üretim Sistemleri, Anadolu Üniversitesi Yayın No: 2275, Eskişehir.

- Dayiođlu, M.A. 2015. Akıllı Sera Teknolojisi. Tarım Türk Dergisi. Sayı 14, Yıl 3, 108-111.
- Ersin, Ç. 2015. Arduino mikrodenetleyici ve güneş enerjisi ile çalışan otomatik bitki sulama sistemi. Yüksek Lisans Tezi, Süleyman Demirel Üniversitesi, Fen Bilimleri Enstitüsü, Isparta.
- Ertam, F. 2005. Bilgisayar ağları ve internet üzerinden başka bir bilgisayarın kontrolü. Yüksek Lisans Tezi, Fırat Üniversitesi, Fen Bilimleri Enstitüsü, Elazığ.
- Farooq, M. U., Waseem, M., Mazhar, S., Khairi, A. and Kamal, T. 2015. A Review on Internet of Things (IoT). International Journal of Computer Applications (0975 8887), Volume 113 - No. 1, March 2015.
- Fukatsu, T. and Hirafuji, M. 2005. Field Monitoring Using Sensor-Nodes with a Web Server. Journal of Robotics and Mechatronics, Vol. 17, No. 2, pp. 164-172.
- Gawali, S.M. and Gajbhiye, S.M., 2014. Design of ARM based Embedded Web Server for Agricultural Application. IJCSIT, Vol. 5(1), pp. 254-358.
- Hariyale, I., Gulhane V., 2012. Development of an Embedded Web Server System for Controlling and Monitoring of Remote Devices Based on ARM and Win CE. International Journal of Recent Technology and Engineering (IJRTE). ISSN: 2277-3878, Volume-1, Issue-2, p. 70-75.
- Horan, B. 2013. Practical Raspberry Pi. Apress, 261 p., USA.
- Hwang, J. and Yoe, H. 2011. Study on the Context-Aware Middleware for Ubiquitous Greenhouses Using Wireless Sensor Networks. Sensors, 2011, 11, 4539-4561.
- Junxiang, G. and Haiqing, D. 2011. Design of Greehouse Surveillance System Based on Embedded Web Server Technology. Procedia Engineering, 23, pp. 374-379.
- Kılınç, M. 2012. Uzaktan Erişimli Sistem Odası Kontrolü. Yüksek Lisans Tezi, Bozok Üniversitesi, Fen Bilimleri Enstitüsü, Yozgat.
- Küçük, Ü. 2015. Veri Haberleşmesi Ders Notları. Yıldız Teknik Üniversitesi, Elektronik ve Haberleşme Mühendisliği Bölümü, İstanbul.
- Kürklü, A. ve Çağlayan, N. 2005. Sera Otomasyon Sistemlerinin Geliştirilmesine Yönelik Bir Çalışma, Akdeniz Üniversitesi Ziraat Fakültesi Dergisi, 2005, 18(1), 25-34.
- Mahadik, S. and Chilveri, G. 2013. Data Acquisition and Control System for Real Time Applications. IJSRD, Vol 1, Issue 9, pp. 1925-1928.
- Montoya, F. G., Gomez, J., Cama, A., Zapata-Sierra, A., Martinez, F., De La Cruz, J. L, Manzano-Agugliaro, F., 2013. A monitoring system for intensive agriculture

- based on mesh networks and the android system, *Computers and Electronics in Agriculture* 99, 14–20.
- Patil, S. and Malviya, A.V. 2014. Review for ARM Based Agricultural Field Monitoring. *IJSRD*, Vol 4, Issue 2.
- Rbinovska, M., Gavrovski, C., Dimcev, V., Krkoleva, A. And Borozan, V. 2015. Environmental parameters monitoring in precision agriculture using wireless sensor networks. *Journal of Cleaner Production*, 88, 297-307.
- Schauer, P. and Debita, G. 2015. Internet of Things Service Systems Architecture. *New Trends in Intelligent Information and Database Systems*, pp. 239-248, ISBN 978-3-319-16211-9.
- Serodio, C., Cunha, J. B., Morais, R., Couto, C. and Monteiro, J. 2001. A Networked Platform for Agricultural Management Systems, *Computer and Electronics In Agriculture*, 31, 75-90.
- Sevgican, A., Tüzel, Y., Gül, A. ve Eltez, R. Z. 2000. Türkiyede Örtüaltı Yetiştiriciliği, V. Türkiye Ziraat Mühendisliği Teknik Kongresi, 17-21 Ocak 2000, Milli Kütüphane, Ankara.
- Tüzel, Y. ve Gül, A. 2008. Seralarda İyi Tarım Uygulamaları. s 172. İzmir.
- Tüzel, Y., Gül, A., Daşgan, H. Y., Öztekin, G. B., Engindeniz, S., Boyacı, H. F., Ersoy, A., Tepe, A. ve Uğur, A. 2010. Örtüaltı Yetiştiriciliğinin Gelişimi. VII. Türkiye Ziraat Mühendisliği Teknik Kongresi, 11-15 Ocak 2010, Milli Kütüphane, Ankara.
- Upton, E. and Halfacree, G. 2013. *Raspberry Pi User Guide*. Wiley, p. 314, USA.
- Vijayakumar, N. and Ramya, R. 2015. The Real Time Monitoring of Water Quality in IoT, *IJSR* 4-3, 879-891.
- Vujovic V. and Maksimovic, M. 2015. Raspberry Pi as a Sensor Web node for home automation. *Computers and Electrical Engineering* Volume 44, May 2015, Pages 153–171.
- Weiser, M. 1991. The computer for the 21st century, *Sci. Amer.*, 1991, pp.66 -75.
- Xia, J., Tang, Z., Shi, X., Fan, L. and Li, H. 2011. An environment monitoring system for precise agriculture based on wireless sensor networks. 2011 Seventh International Conference on Mobile Ad-hoc and Sensor Networks.
- Yang, S.H. ve Chen, X., 2003. Design Issues and Implementation of Internet-based Process Control Systems, *Control Engineering Practice*, 11, 709-720.
- Zaceping, A. and Kviess, A. 2015. System Architectures for Real-time Bee Colony Temperature, *Procedia Computer Science*, 43, 86-94.

Zhu, Y.W., Zhong, X.X. and Shi, J.F. 2006. The Design of Wireless Sensor Network System Based on ZigBee Technology for Greenhouse. Journal of Physics: Conference Series 48, 115-1199.



EKLER

EK 1 ReadSensors Uygulaması Kaynak Kodları.....	60
EK 2 Linux Servisi Kaynak Kodları.....	63
EK 3 config.ini Dosyasının İeriđi.....	69
EK 4 rth Dosyasının İeriđi.....	70
EK 5 Ađ Servisinin Kaynak Kodları.....	72

EK 1 ReadSensors Uygulaması Kaynak Kodları

a. Main.cs

```
using System;
using System.IO;
using System.Collections.Generic;
using System.Text;

namespace CSReadSensors
{
    class MainClass
    {
        public static void Main (string[] args)
        {
            try
            {
                List<Sensor> allsensors = new List<Sensor>();
                var sensors = Directory.GetDirectories(Assets.GetSensorsPath());
                foreach (var sensor in sensors)
                {
                    var sensorname = new DirectoryInfo(sensor).Name;
                    if(sensorname.IndexOf("master") >= 0)
                    {
                        continue;
                    }
                    allsensors.Add(new Sensor(sensorname, Assets.GetSensorsPath()));
                }
                StringBuilder tmp = new StringBuilder();
                tmp.Append("[");
                foreach (var sensor in allsensors)
                {
                    tmp.Append("{\"sensor_id\":\"" + sensor.SensorName + "\",\"sensor_value\":\"" + sensor.ReadSensor()
+ "\"},");
                }
                var tmp2 = tmp.ToString().TrimEnd(new char[]{' '});
                Console.Write(tmp2 + ']');
            }
            catch (Exception ex)
            {
                Console.WriteLine("Error: " + ex.Message);
            }
        }
    }
}
```

b. Sensor.cs

```
using System;
using System.IO;

namespace CSReadSensors
{
    public class Sensor
    {
        private string _sysdir;
        private string _sensorid;
        private readonly string _file = "w1_slave";

        public Sensor (string SensorName, string Path)
        {
            _sysdir = Path;
            _sensorid = SensorName;
        }

        public string SensorName {
            get {
                return _sensorid;
            }
        }

        public string ReadSensor ()
        {
            using (StreamReader sr = new StreamReader(GetPathName(_sysdir, _sensorid, _file)))
            {
                return GetTemperatureValue(sr.ReadToEnd());
            }
        }

        private string GetPathName (string path, string sensorname, string file)
        {
            return Path.Combine(path,sensorname,file);
        }

        private string GetTemperatureValue (string rawdata)
        {
            var tvalindex = rawdata.IndexOf("t=");

            if (rawdata.IndexOf("YES") > -1 && tvalindex > -1)
            {
                var tval = rawdata.Substring(tvalindex + 2);
                int rtn = 0;
                if (int.TryParse(tval,out rtn))
                {
                    double dbl = rtn / (double)1000;
                    return (dbl.ToString());
                }
                else
                {
                    return "Error";
                }
            }
            else {
                return "Error";
            }
        }
    }
}
```


c. Assets.cs

```
using System;

namespace CSReadSensors
{
    public static class Assets
    {
        private static readonly string SensorsPath =
        ":seperator:sys:seperator:bus:seperator:w1:seperator:devices:seperator:";
        public static string GetSensorsPath ()
        {
            return Assets.SensorsPath.Replace(":seperator:", System.IO.Path.DirectorySeparatorChar.ToString());
        }
    }
}
```



EK 2 Linux Servisinin Kaynak Kodları

a. KernelModules.cs

```
using System;
using System.Diagnostics;
using System.IO;

namespace CSSendData
{
    public static class KernelModules
    {
        private static readonly string _W1ModuleName = "w1_gpio";
        private static readonly string _W1ThermModuleName = "w1_therm";
        private static string _lsmout = "";

        public static bool InitModules ()
        {
            for (int retrycount = 0; retrycount < 3; ++retrycount)
            {
                _lsmout = StartCommand ("/sbin/lsmout", "");
                if (!CheckW1ModuleIsLoaded ())
                {
                    StartCommand ("/sbin/modprobe", _W1ModuleName);
                }
                if (!CheckW1ThermModuleIsLoaded ())
                {
                    StartCommand ("/sbin/modprobe", _W1ThermModuleName);
                }
                if (CheckW1ModuleIsLoaded () && CheckW1ThermModuleIsLoaded ())
                {
                    return true;
                }
            }
            throw (new Exception ("Modules fail initialization!"));
        }

        public static string StartCommand (string CommandName, string Arguments)
        {
            string stderr = "";
            string stdout = "";
            int exitcode = -1;
            ProcessStartInfo psi = new ProcessStartInfo (CommandName);
            psi.Arguments = Arguments;
            psi.CreateNoWindow = true;
            psi.UseShellExecute = false;
            psi.RedirectStandardError = true;
            psi.RedirectStandardOutput = true;

            using (Process p = Process.Start(psi))
            {
                StreamReader srError = p.StandardError;
                StreamReader srStandartOut = p.StandardOutput;
                p.WaitForExit ();
                stderr = srError.ReadToEnd ();
                stdout = srStandartOut.ReadToEnd ();
                exitcode = p.ExitCode;
                p.Close ();
            }
            if (exitcode == 0)

```

```
{
    return stdout;
}
return stderr;

}

private static bool CheckW1ModuleIsLoaded ()
{
    return (_lsmodule.IndexOf (_W1ModuleName) > -1);
}

private static bool CheckW1ThermModuleIsLoaded ()
{
    return (_lsmodule.IndexOf (_W1ThermModuleName) > -1);
}
}
```



b. ReadIni.cs

```
using System;
using System.IO;
using System.Collections.Generic;

namespace CSSendData
{
    public class ReadIni
    {
        public static Dictionary<string, string> Settings {
            get { return _settings; }
        }
        private static Dictionary<string, string> _settings;

        static ReadIni ()
        {
            _settings = new Dictionary<string, string> ();
        }

        try
        {
            using (StreamReader sr = new StreamReader("/etc/rth/config.ini"))
            {
                var tmp = sr.ReadToEnd ();
                var tmp2 = tmp.Split (new string[] { "\r\n" }, StringSplitOptions.RemoveEmptyEntries);
                foreach (var item in tmp2)
                {
                    var tmp3 = item.Split (new char[] { '=' });
                    for (int i = 0; i < tmp3.Length; i+= 2)
                    {
                        _settings.Add (tmp3 [i].Trim (new char[] { ' ', '\'" }),
                                      tmp3 [i + 1].Trim (new char[] { ' ', '\'" }));
                    }
                }
            }
        }
        catch (Exception ex)
        {
            Console.WriteLine (ex);
            throw ex;
        }
    }
}
```

c. Program.cs

```
using System;
using System.Collections.Generic;
using System.Text;
using System.Threading;
using System.Threading.Tasks;
using System.Net;
using System.IO;
using System.Diagnostics;

namespace CSSendData
{
    class Program
    {
        private static ManualResetEvent mre = new ManualResetEvent (false);
        private static System.Timers.Timer _timer2;
        private static string _remotehost;
        private static string _apikey;
        private static string _stationname;
        private static double _sensorinterval;
        private static string _serviceuri;

        private static void InitializeVariables ()
        {
            Dictionary<string,string> configs = ReadIni.Settings;

            _remotehost = configs ["RemoteAddress"];
            _stationname = configs ["StationName"];
            _apikey = configs ["ApiKey"];
            _serviceuri = string.Format ("http://{0}/service/jsonservice.php", _remotehost);

            double tmp = 60 * 1000;
            if (Double.TryParse (configs ["SensorInterval"], out tmp))
            {
                if(tmp < 6)
                {
                    tmp = 6;
                }
                _sensorinterval = tmp * 1000;
            }
            else
            {
                _sensorinterval = tmp;
            }
            _timer2 = new System.Timers.Timer ();
            _timer2.Elapsed += HandleElapsed;
            _timer2.Interval = _sensorinterval;
        }

        static void Main (string[] args)
        {
            InitializeVariables ();
            try
            {
                KernelModules.InitModules ();
            }
            catch (Exception ex)
            {
                var t = Console.Error;
                t.WriteLine (string.Format ("Fail {0} at {1}.", ex.Message,
                    DateTime.Now.ToString ("yyyy-MM-dd HH:mm:ss")));
            }
            mre.Set ();
        }
    }
}
```

```

    }

    _timer2.Start ();
    Console.WriteLine (string.Format ("Start at {0}.", DateTime.Now.ToString ("yyyy-MM-dd
HH:mm:ss")));
    mre.WaitOne ();
    Console.WriteLine (string.Format ("Stop at {0}.", DateTime.Now.ToString ("yyyy-MM-dd
HH:mm:ss")));
    }

private static void HandleElapsed (object sender, System.Timers.ElapsedEventArgs e)
{

    var f = sender as System.Timers.Timer;
    if (f != null)
    {
        f.Stop ();
        var EXECRTN = KernelModules.StartCommand("/usr/bin/mono",
"/usr/local/RTHService/ReadSensors.exe");
        var postdata = "{" +
            "\"station_name\": \"" + _stationname + "\", " +
            "\"api_key\": \"" + _apikey + "\", " +
            "\"data\": " + EXECRTN + " }";
        if (POST (_serviceuri, postdata) == true)
        {
        }
        f.Start ();
    }
}

static bool POST (string url, string jsonContent)
{
    HttpWebRequest request = (HttpWebRequest)WebRequest.Create (url);
    request.Method = "POST";
    request.Timeout = 2000;
    System.Text.UTF8Encoding encoding = new System.Text.UTF8Encoding ();
    Byte[] byteArray = encoding.GetBytes (jsonContent);

    request.ContentLength = byteArray.Length;
    request.ContentType = @"application/json";
    try
    {
        using (Stream dataStream = request.GetRequestStream())
        {
            dataStream.Write (byteArray, 0, byteArray.Length);
        }
        long length = 0;

        using (HttpWebResponse response = (HttpWebResponse)request.GetResponse())
        {
            length = response.ContentLength;
            using (var sr = new StreamReader(response.GetResponseStream()))
            {

                if (response.StatusCode == HttpStatusCode.OK)
                {
                    //Console.WriteLine (sr.ReadToEnd ());
                    return true;
                }
                else
                {
                    Console.WriteLine (string.Format ("Posting to {0} has error {1} at {2}.", url,

```

```
response.StatusCode.ToString (), DateTime.Now.ToString ("yyyy-MM-dd HH:mm:ss"));
    return false;
}
}
}
}
catch (Exception ex)
{
    Console.WriteLine (string.Format ("Posting to {0} failed at {1}. Exception: {2}", url,
DateTime.Now.ToString ("yyyy-MM-dd HH:mm:ss"), ex.Message));
    return false;
}
}
}
```



EK 3 config.ini Dosyasının İeriĐi

```
StationName = "SlavePi1"  
StationMod = 1  
RemoteAddress = "127.0.0.1"  
ApiKey = 1234567890  
SensorInterval = 60  
DBAddress = "localhost"  
DBUserName = "root"  
DBPassword = "dbsifre"  
DBName = "rtsys"  
DBTableName = "measurements"  
UserPassword = "kullanicisifre"  
AdminPassword = "yoneticisifre"
```



EK 4 rth Dosyasının İçeriği

```
#!/bin/bash
#/etc/init.d/rth

### BEGIN INIT INFO
# Provides:      RTHService
# Required-Start: $remote_fs $syslog
# Required-Stop: $remote_fs $syslog
# Should-Start:  $network $time
# Should-Stop:   $network $time
# Default-Start: 2 3 4 5
# Default-Stop:  0 1 6
# Short-Description: Starts RTHService automatically
# Description:
### END INIT INFO

SERVICE_NAME=RTHService
PATH_TO_APP=/usr/local/RTHService/service.exe
PID_PATH=/tmp/RTHService-pid

case $1 in
start)
    echo "Starting $SERVICE_NAME ..."
    if [ ! -f $PID_PATH ]; then
        nohup /usr/bin/mono $PATH_TO_APP > /tmp/$SERVICE_NAME-log.txt 2>
/tmp/$SERVICE_NAME-error.txt &
        echo $! > $PID_PATH
        echo "$SERVICE_NAME started ..."
    else
        echo "$SERVICE_NAME is already running ..."
    fi
    ;;
stop)
    if [ -f $PID_PATH ]; then
        PID=$(cat $PID_PATH);
        echo "$SERVICE_NAME stopping ..."
        kill $PID;
        echo "$SERVICE_NAME stopped ..."
        rm $PID_PATH
    else
        echo "$SERVICE_NAME is not running ..."
    fi
    ;;
restart)
    if [ -f $PID_PATH ]; then
        PID=$(cat $PID_PATH);
        echo "$SERVICE_NAME stopping ..."
        kill $PID;
        echo "$SERVICE_NAME stopped ..."
        rm $PID_PATH
        echo "Starting $SERVICE_NAME ..."
        nohup /usr/bin/mono $PATH_TO_APP >> /tmp/$SERVICE_NAME-log.txt 2>>
/tmp/$SERVICE_NAME-error.txt &
        echo $! > $PID_PATH
        echo "$SERVICE_NAME started ..."
    else
        echo "$SERVICE_NAME is not running ..."
        echo "Starting $SERVICE_NAME ..."
        nohup /usr/bin/mono $PATH_TO_APP >> /tmp/$SERVICE_NAME-log.txt 2>>
/tmp/$SERVICE_NAME-error.txt &
```

```
echo $! > $PID_PATH
echo "$SERVICE_NAME started ..."
fi
;;
cleanlogs)
if [ -f /tmp/"$SERVICE_NAME"-log.txt ]; then
rm /tmp/"$SERVICE_NAME"-log.txt
fi
if [ -f /tmp/"$SERVICE_NAME"-error.txt ]; then
rm /tmp/"$SERVICE_NAME"-error.txt
fi
;;
*)
echo "Usage: rth {start|stop|restart|cleanlogs}"
;;
esac

exit 0
```



EK 5 Ağ Servisinin Kaynak Kodları

```
<?php
ini_set('display_errors', 'off');
require_once('../classes/db.php');
require_once('../classes/iniutils.php');
require_once('../classes/iputils.php');

$str_json = file_get_contents('php://input');
$response = json_decode($str_json);

$configs = get_configs();

if ( isset($response)
    && property_exists($response, 'station_name')
    && property_exists($response, 'api_key') )
{

    $station_name = $response->station_name;
    $api_key = $response->api_key;

    $DBConn = new DBConnection();
    if(!$DBConn)
    {

        header("HTTP/1.0 500 Internal Server Error");
        exit;
    }
    if($configs["ApiKey"] == $api_key)
    {

        $datas = $response->data;
        $errs = array();

        foreach ($datas as $data)
        {

            if( property_exists($data, 'sensor_id') && property_exists($data, 'sensor_value'))
            {

                if($rtn = $DBConn->addMeasurement($station_name, $data->sensor_id,
                    $data->sensor_value, getRealIp())

                )
                {
                    //echo 'OK';
                }
                else
                {
                    $errs[] = $rtn;
                }
            }
            else
            {
                header("HTTP/1.0 400 Bad Request");
                exit();
            }
        }
    }

    if(count($errs) != 0)
    {
        header("HTTP/1.0 520 Unknown Error");
        exit();
    }
}
```

```
else
{
    header("HTTP/1.0 200 OK");
    exit();
}
else
{
    header("HTTP/1.0 401 Unauthorized");
    exit();
}
else
{
    header("HTTP/1.0 400 Bad Request");
    exit();
}
?>
```



ÖZGEÇMİŞ

Adı Soyadı : Furkan Uğur
Doğum Yeri : Ankara
Doğum Tarihi : 23.01.1988
Medeni Hali : Evli
Yabancı Dili : İngilizce

Eğitim Durumu (Kurum ve Yıl)

Lise : Ankara Bahçelievler Deneme Lisesi (2005)
Lisans : Ankara Üniversitesi Ziraat Fakültesi
Tarım Makinaları Bölümü (2012)
Yüksek Lisans : Ankara Üniversitesi Fen Bilimleri Enstitüsü
Tarım Makinaları ve Teknolojileri A.B.D.
(Eylül 2012 - Ocak 2016)

Çalıştığı Kurum/Kurumlar ve Yıl

T.C. Gıda Tarım ve Hayvancılık Bakanlığı Eğitim Yayım ve Yayınlar Dairesi, 2013 - ...