



BİLECİK ŞEYH EDEBALI  
ÜNİVERSİTESİ

**BİLECİK  
ŞEYH EDEBALI ÜNİVERSİTESİ**

**Fen Bilimleri Enstitüsü  
Bilgisayar Mühendisliği Ana Bilim Dalı**

**RASPBERRY Pİ MİNİ BİLGİSAYARI İLE ORTAM  
İZLEME SİSTEMİ GELİŞTİRİLMESİ**

**Mehmet BALCI  
Yüksek Lisans**

**Tez Danışmanı  
Doç. Dr. Uğur YÜZGEÇ**

**BİLECİK, 2019  
Ref. No: 10275056**



BİLECİK ŞEYH EDEBALI  
ÜNİVERSİTESİ

**BİLECİK  
ŞEYH EDEBALI ÜNİVERSİTESİ**

**Fen Bilimleri Enstitüsü  
Bilgisayar Mühendisliği Ana Bilim Dalı**

**RASPBERRY Pİ MİNİ BİLGİSAYARI İLE ORTAM  
İZLEME SİSTEMİ GELİŞTİRİLMESİ**

**Mehmet BALCI  
Yüksek Lisans**

**Tez Danışmanı  
Doç. Dr. Uğur YÜZGEÇ**

**BİLECİK, 2019**



BİLECİK ŞEYH EDEBALI  
ÜNİVERSİTESİ

**BİLECİK  
SEYH EDEBALI UNIVERSITY**

**Graduate School of Sciences  
Department of Computer Engineering**

**DEVELOPING THE ENVIRONMENT MONITORING SYSTEM  
WITH RASPBERRY PI MINI COMPUTER**

**Mehmet BALCI  
Master's Thesis**

**Thesis Advisor  
Assoc. Dr. Uğur YÜZGEÇ**

**BİLECİK, 2019**



**BİLECİK ŞEYH EDEBALI ÜNİVERSİTESİ**  
**FEN BİLİMLERİ ENSTİTÜSÜ**

**YÜKSEK LİSANS**  
**JÜRİ ONAY FORMU**

Bilecik Şeyh Edebali Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulunun 12/06/2019 tarih ve 30-03 sayılı kararıyla oluşturulan jüri tarafından 11/07/2019 tarihinde tez savunma sınavı yapılan Mehmet BALCI'nın "Raspberry Pi Mini Bilgisayarı ile Ortam İzleme Sistemi Geliştirilmesi" başlıklı tez çalışması Bilgisayar Mühendisliği Ana Bilim Dalında YÜKSEK LİSANS tezi olarak oy birliği/ oy çokluğu ile kabul edilmiştir.

**JÜRİ**

(Başkan olarak belirlenen kişi (JÜRİ BAŞKANI) şeklinde belirtilmelidir.)

ÜYE

(TEZ DANIŞMANI):

*11/07/2019*  
*Dr. Öğr. Üyesi Uğur YÜZGEÇ (Jüri Başkanı)*

ÜYE :

*Dr. Öğr. Üyesi Nesibe YALÇIN*

ÜYE :

*Dr. Öğr. Üyesi Süleyman UZUN*

**ONAY**

Bilecik Şeyh Edebali Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulunun  
.../.../..... tarih ve ...../..... sayılı kararı.

İMZA/ MÜHÜR

## TEŐEKKÜR

Çalıőmalarım sırasında yardımlarını esirgemeyen tez danıőmanım Doç. Dr. Uęur YÜZGEÇ, Bilgi İőlem Daire Başkanım Murat FİDAN ve Őube Müdürüm Hasan KILIÇ, Araő. Gör. Hakan ÜÇGÜN, mesai arkadaşlarım Öğr. Gör. Yusuf MUŐTU ve Bilgisayar Mühendisi Semih KARACA'ya teőekkürlerimi sunuyorum.

## BEYANNAME

Bilecik Şeyh Edebali Üniversitesi Fen Bilimleri Enstitüsü Tez Yazım Kılavuzu'na uygun olarak hazırladığım bu tez çalışmada, tez içindeki tüm verileri akademik kurallar çerçevesinde elde ettiğimi, görsel ve yazılı tüm bilgi ve sonuçların akademik ve etik kurallara uygun olarak sunulduğunu, kullanılan verilerde herhangi bir tahrifat yapılmadığını, başkalarının eserlerinden yararlanılması durumunda ilgili eserlere bilimsel normlara uygun olarak atıfta bulunulduğunu, tezde yer alan verilerin bu Üniversite veya başka bir üniversitede herhangi bir tez çalışmada kullanılmadığını beyan ederim.

11./07/ 2019



Mehmet BALCI

# RASPBERRY Pİ MİNİ BİLGİSAYARI İLE ORTAM İZLEME SİSTEMİ GELİŞTİRİLMESİ

## ÖZET

Nesnelerin interneti kavramının gelişmesiyle birlikte akıllı ev konsepti son yıllarda insanların hayatına girmiştir. Bu sayede insanlar, evlerinden veya iş yerlerinden uzakta olduklarında, bu mekanlarda meydana gelen değişimleri anlık olarak takip edebilmekte ve mekandaki cihazları uzaktan kontrol edebilmektedirler. Ayrıca ortamdaki sıcaklık, nem, gaz, alev, hareket ve su basmaları gibi veriler izlenebilmekte ve bu veriler ile ilgili alarmlar oluşturulup söz konusu alarm durumları meydana geldiğinde kullanıcılara bilgilendirme yapılabilmektedir. Uzaktan verilerin anlık olarak izlenmesi üzerine birçok sistem geliştirilmiştir ve teknolojinin hızlı gelişmesi ile bu tip sistemler de gelişmektedir. Piyasada var olan uzaktan algılama sistemleri izlenecek verilere, kullanıcı sayısına, haberleşme tekniğine bağlı olarak yüksek maliyetli olabilmektedirler. Bu çalışma kapsamında Raspberry Pi mini bilgisayar ile uzaktan bir ortam izleme sistemi geliştirilmiştir. Bu sistemin kurulu olduğu ortamlar internet tarayıcıları aracılığıyla uzaktan takip edilebilmekte ve önceden belirlenen alarm durumları meydana geldiğinde sisteme tanımlanan kişilere e-posta ile bilgilendirme yapılabilmektedir. Ayrıca ortamın fiziksel değerleri için sisteme limitler tanımlanabilmekte; bu limitler aşıldığında uyarılar penceresinde bildirimler yapılmaktadır. Bu çalışmadaki sistem; veri merkezleri, seralar ve evler gibi kapalı ortamlardaki fiziksel koşulları takip edebilecek şekilde tasarlanmıştır.

**Anahtar Kelimeler:** Raspberry Pi Mini Bilgisayarı; Ortam İzleme Sistemi; Nesnelerin İnterneti

## **DEVELOPING THE ENVIRONMENT MONITORING SYSTEM WITH RASPBERRY MINI COMPUTER**

### **ABSTRACT**

With the development of the Internet of Things concept, the concept of smart home has entered the lives of people in recent years. In this way, when people are away from home or workplace, they can instantly monitor the changes occurring in these places and control the devices in the place remotely. In addition, data such as temperature, humidity, gas, flame, motion and floods in the environment can be monitored and alarms related to this data can be generated and users can be informed when such alarm conditions occur. Many systems have been developed on the instant monitoring of remote data and such systems are developing with the rapid development of technology. Remote sensing systems in the market can be costly depending on the data to be monitored, the number of users and the communication technique. Within the scope of this study, a remote environment monitoring system has been developed with Raspberry Pi minicomputer. The environments in which this system is installed can be monitored remotely via internet browsers and e-mails can be sent to the persons identified in the system when pre-determined alarm conditions occur. In addition, limits can be defined for the physical values of the environment; when these limits are exceeded, notifications are made in the warnings window. The system in this study; it is designed to monitor physical conditions in indoor environments such as data centers, greenhouses and homes.

**Keywords:** Raspberry Pi Mini Computer; Environment Monitoring System; Internet of Things

## İÇİNDEKİLER

	Sayfa No
TEŞEKKÜR .....	
BEYANNAME .....	
ÖZET.....	I
ABSTRACT .....	II
İÇİNDEKİLER .....	III
ŞEKİLLER DİZİNİ .....	V
ÇİZELGELER DİZİNİ .....	VII
SİMGELER ve KISALTMALAR DİZİNİ .....	VIII
<b>1. GİRİŞ.....</b>	<b>1</b>
<b>2. KULLANILAN YAZILIMLAR ve DONANIMLAR .....</b>	<b>5</b>
2.1. Kullanılan Yazılımlar .....	5
Python Programlama Dili .....	5
HTML.....	6
PHP.....	6
JavaScript.....	6
JSON (JavaScript Object Notation).....	7
MySQL .....	7
2.2. Kullanılan Donanımlar .....	8
2.2.1. Raspberry Pi Mini Bilgisayarı .....	8
2.2.2. Sensörler .....	9
2.2.3. Analog Sayısal Dönüştürücü (Analog Digital Converter, ADC) .....	15
<b>3. RASPBERRY PI İLE ORTAM VERİLERİNİN İZLENMESİ.....</b>	<b>17</b>
3.1. Sensörlerden Veri Alınması .....	19
3.2. Verilerin Veritabanına Kaydedilmesi.....	26
3.3. Verilerin İnternet Sayfasında Yayınlanması .....	33
3.3.1. Anlık Verilerin Yayınlanması.....	34
3.3.2. Kayıtlı Verilerin Yayınlanması.....	41
3.3. Alarmlar .....	43
<b>3. SONUÇ .....</b>	<b>46</b>
<b>KAYNAKLAR .....</b>	<b>48</b>

<b>EKLER.....</b>	<b>50</b>
<b>ÖZ GEÇMİŞ.....</b>	

## ŞEKİLLER DİZİNİ

### Sayfa No

Şekil 2.1. Kullanılan yazılımların birbiriyle ilişkisi. ....	5
Şekil 2.2. Veritabanına ait ilişkiler diyagramı (ER). ....	8
Şekil 2.3. Raspberry Pi giriş-çıkış pinleri ve diğer bağlantı portları (BMOW, 2019). ....	9
Şekil 2.4. Çalışmada kullanılan sensörler; (a) sıcaklık/nem sensörü, (b) gaz sensörü, (c) su sensörü, (d) mesafe sensörü, (e) alev sensörü, (f) ışık sensörü, (g) hareket sensörü. ....	10
Şekil 2.5. DHT11 sensörünün yapısındaki dirence bağlı sıcaklık grafiği (Last Minute Engineers, 2019). ....	10
Şekil 2.6. DHT11 sensörünün yapısındaki dirence bağlı nem grafiği (Algün, 2018). ...	11
Şekil 2.7. MQ-2 gaz sensörüne ait ölçüm grafiği (Seeed, 2019). ....	12
Şekil 2.8. Su sensörüne su temasının olduğu ve olmadığı durumlara ait grafik. ....	12
Şekil 2.9. Mesafe sensörünün ses dalgası ile uzaklık ölçümü (Faranux Electronics, 2019). ....	13
Şekil 2.10. Gerilimin ışığın yoğunluğuna bağlı değişimi. ....	14
Şekil 2.11. ADC ve pinlerine ait görüntü. ....	16
Şekil 3.1. Sistemin çalışmasına ait genel diyagram. ....	17
Şekil 3.2. Sistemin ana sayfasına ait ekran görüntüsü. ....	18
Şekil 3.3. Çalışma kapsamında önerilen sistemin fotoğrafı. ....	19
Şekil 3.4. Programın genel çalışma diyagramı. ....	21
Şekil 3.5. Sıcaklık ve nem bilgisinin terminalde görünümü. ....	22
Şekil 3.6. Gaz sensörünün kalibrasyonunun yapılması. ....	23
Şekil 3.7. Gaz ölçümü sonuçlarına ait ekran görüntüsü. ....	23
Şekil 3.8. Su seviyesinin ölçümü. ....	25
Şekil 3.9. Ortam parlaklığının terminaldeki ekran görüntüsü. ....	26
Şekil 3.10. Anlik_Olcumler Tablosu. ....	27
Şekil 3.11. Kayitli_Olcumler Tablosu. ....	27
Şekil 3.12. ID'lerin veritabanında kaydedilmesi. ....	28
Şekil 3.13. Nem bilgisinin veritabanında kaydedilmesi. ....	28
Şekil 3.14. Sıcaklık bilgisinin veritabanında kaydedilmesi. ....	29
Şekil 3.15. Hareket bilgisinin veritabanında kaydedilmesi. ....	29
Şekil 3.16. Alev bilgisinin veritabanında kaydedilmesi. ....	29

<b>Şekil 3.17.</b> CO'nun ppm miktarının veritabanında kaydedilmesi.....	30
<b>Şekil 3.18.</b> Ortamdaki dumanın ppm miktarının veritabanında kaydedilmesi. ....	30
<b>Şekil 3.19.</b> Ortamdaki LPG'nin ppm miktarının veritabanına kaydedilmesi. ....	31
<b>Şekil 3.20.</b> Su tespitine ait bilginin veritabanında kaydedilmesi. ....	31
<b>Şekil 3.21.</b> Ortamdaki suyun seviyesinin veritabanında kaydedilmesi.....	32
<b>Şekil 3.22.</b> Zaman bilgisinin veritabanında kaydedilmesi. ....	32
<b>Şekil 3.23.</b> Zaman bilgisinin, Unix Epoch Zamanı şeklinde kaydedilmesi. ....	33
<b>Şekil 3.24.</b> Durum bilgisinin veritabanında kaydedilmesi.....	33
<b>Şekil 3.25.</b> Sistemin giriş sayfası. ....	34
<b>Şekil 3.26.</b> Anlık bilgilerin sistemin ana sayfasında gösterilmesi. ....	35
<b>Şekil 3.27.</b> Anlık sıcaklık ve nem grafiklerinin oluşumu. ....	36
<b>Şekil 3.28.</b> Anlık sıcaklık bilgisi grafiği. ....	36
<b>Şekil 3.29.</b> Anlık nem bilgisi grafiği.....	37
<b>Şekil 3.30.</b> Anlık parlaklık bilgisine ait grafik.....	39
<b>Şekil 3.31.</b> Anlık gaz bilgisi grafiği. ....	40
<b>Şekil 3.32.</b> Uyarılar sayfasına ait ekran görüntü.....	41
<b>Şekil 3.33.</b> Kayıtlı verilerin toplu bir şekilde bulunduğu ana sayfa. ....	42
<b>Şekil 3.34.</b> Sıcaklık bilgisine ait kayıtlı veriler grafiği. ....	42
<b>Şekil 3.35.</b> Ortamın genel durumuna ait son durumlar listesi.....	43
<b>Şekil 3.36.</b> Sistem tarafından gönderilen örnek alarm e-postası.....	45

**ÇİZELGELER DİZİNİ****Sayfa No**

<b>Çizelge 2.1.</b> ADC'nin Raspberry Pi'ye bağlanması .....	15
--	----

**SİMGELER ve KISALTMALAR DİZİNİ****Simgeler**

ADC : Analog Digital Converter

CO : Karbonmonoksit

IoT : Internet of Things

LDR : Light Dependent Resistor

LPG : Liquefied Petroleum Gas

PIR : Passive Infrared Sensor

PPM : Parts Per Million

## 1. GİRİŞ

Akıllı ev otomasyon sistemleri ve nesnelerin interneti ile insanlar, istedikleri bir ortama kurulacak sistemlerle internet erişiminin olduğu yerlerde, ortam hakkında kolayca bilgi sahibi olabilmektedir. Bu sayede insanlar tatil ve iş görüşmesi gibi nedenlerle evlerinden uzaklaştıklarında, sistem ortamı sürekli olarak kontrol etmekte ve anormal bir durum olduğunda kullanıcıyı bilgilendirmektedir. Böylece meydana gelebilecek olumsuz durumların önüne geçilebilmekte veya bu tür durumlar sonucunda oluşabilecek maddi-manevi zararların etkisi azaltılabilmektedir.

Bu tez çalışması kapsamında Raspberry Pi mini bilgisayarını kullanarak uzaktan bir ortam izleme sistemi geliştirilmiştir. Çalışmada ortamdaki sıcaklık, nem, CO (karbonmonoksit), LPG (liquefied petroleum gas) ve duman değerleri kullanılan sensörler ile sürekli olarak ölçülmektedir. Ortamı su basma ihtimaline karşı zemine su sensörleri kullanılarak, su tespit edildiğinde suyun zeminden yüksekliği de kontrol edilebilmektedir. Meydana gelebilecek yangınlar için alev tespiti, ortamdaki yabancı kişilerin tespit edilebilmesi için hareket algılaması ve gece-gündüz kontrolü için parlaklık ölçümü yapılmaktadır. Geliştirilen sistem üzerinden ölçülen değerler için eşik değerleri oluşturulmuştur. Belirlenen normal koşullar dışında bir durum veya ölçüm tespit edildiğinde önceden belirlenen kullanıcılara meydana gelen durumla ilgili e-posta gönderilmektedir.

Bu çalışma kapsamında kurulan sistem ile ortama ait bilgiler anlık olarak internet üzerinden takip edilebilmektedir. Sistem tarafından veritabanına kaydedilen veriler ile internet ara yüzünden seçilecek tarihler arasında bulunan ortamın geçmişe dönük istatistiksel değerleri de görülebilmektedir. Anormal durumlar için alarm e-postaları da gönderilebilmektedir. Sistemde kullanılan sensörlerin ve mini bilgisayarın fiyatlarının ekonomik olması ve sistemde kullanılan yazılımların ücretsiz olması sistemin maliyetini düşürmektedir. Ayrıca kullanılan yazılımların açık kaynak kodlu olması da sistem üzerinde yapılacak değişiklikler için geniş bir yetki vermektedir.

Literatürde Raspberry Pi, nesnelerin interneti (IoT), akıllı ev otomasyonu, uzaktan ortam izleme konuları ile ilgili yapılan bazı çalışmalara ait bilgiler aşağıda verilmektedir.

Jain ve arkadaşlarının yaptıkları çalışma, Raspberry Pi tabanlı olup; Python dili ile geliştirilen sistemde e-posta hesabı ve web sunucusu kullanılmaktadır. Python dili

kullanılarak yazılmış bir algoritmaya e-posta ile gönderilen talimatlar, programda yorumlanarak Raspberry Pi'nin e-postada istenen çıkış veya çıkışlarındaki ledleri yakmaktadır. Çıkışta bulunan ledler ortamda bulunan cihazları temsil etmektedir; bu şekilde sistemin sağlıklı çalışıp çalışmadığı kontrol edilmektedir. Proje benzetim şeklinde yapılmış olup çıkışlarda gerçek cihazlar kullanılmamaktadır; ancak çalışma sayesinde e-posta ile sistem (e-posta ile gönderilen komutlar algoritmaya uygun olmalı) kontrol edilebilmektedir (Jain vd., 2014).

Prasad ve meslektaşları tarafından geliştirilen sistemde Raspberry Pi, PIR sensörü, USB kamera ve 3G modem kullanılmaktadır. Sistemin kurulu olduğu ortamda hareket sensörleri tarafından bir hareket algılandığında alarm verilip kamera kaydına başlamaktadır. Alarm durumuna ait görüntü, 3G modem aracılığıyla web uygulamaları kullanılarak kullanıcının akıllı telefonuna gönderilebilmektedir. Kullanıcıya ortama ait kayıtlı ve canlı kamera kayıtlarını izleme olanağı iki şekilde verilmektedir. Birincisi internet tarayıcısı aracılığıyla; ikincisi Raspberry Pi'deki uygulamalarla HDMI bağlantısına bir ekran bağlanması şeklindedir. Sistemde USB kamera kullanıldığından kameralar Raspberry Pi'ye birkaç metre uzaklıkta olmalı ve Raspberry Pi'deki USB sayısı kadar kamera bağlanabilmektedir (Prasad vd., 2014).

Patchava ve arkadaşları tarafında geliştirilen sistem içerisinde Raspberry Pi, CVT<sup>1</sup> ve M-JPG Streamer<sup>2</sup> kullanılmaktadır. Hareket sensörleri ve kameralara IoT (Internet of Things) cihazlarının entegrasyonunun Raspberry Pi ile kullanılarak dizayn edilen bir akıllı ev otomasyonudur. Sistemin internet sayfasında ortam, kamera ile canlı olarak izlenebilmektedir. Ayrıca internet sayfasında, sisteme entegre olan ortamın ışıkları ve iklimlendirme sistemi açılıp kapatılabilmektedir. Ortamdaki hareket sensörleriyle hareket algılandığında sisteme tanımlanan telefon numarasına mesaj da gönderilebilmektedir (Patchava vd., 2015).

Sandeep ve arkadaşları tarafından yapılan çalışma, web sunucu tabanlı yapılan ev otomasyonu gibi sistemler üzerine yapılan bir araştırmadır. Bu çalışmada, insanların yaşadığı ortamlara Raspberry Pi tabanlı otomasyon sistemi kurulmuştur. Sistem için bir

---

<sup>1</sup> **CVT:** Computer Vision Technique, kamera gibi kaynaklardan alınan resim veya özellik akışları üzerinde çalışmaya olanak sağlayan açık kaynak kodlu programdır.

<sup>2</sup> **M-JPG Streamer:** kamera gibi bir kaynaktan alınan JPEG formatındaki resimlerin IP tabanlı ağ üzerinden tarayıcı veya video oynatıcıya akışını sağlar.

internet sayfası oluşturulmuş ve bu sayfa üzerinde ortamdaki cihazların kontrol edilebilmesi için kontroller (cihazları açıp kapatma) bulunmaktadır. Örneğin uygulamanın kullanıcı ve şifresini bilen biri internet bağlantısının olduğu bir yerden sisteme erişip sistemdeki tüm cihazları kontrol edebilmektedir. Bu tür uygulamalarla insanların günlük yaşamı kolaylaşmakta ve ortamların güvenliği rahatlıkla sağlanabilmektedir (Sandeep vd., 2015).

Agrawal ve Singhal tarafından önerilen sistem Raspberry Pi, Arduino, Xbee modülleri ve röle devresinden oluşan bir akıllı damlama sulama sistemidir. Xbee modülü sistemin kablosuz kontrolü için kullanılmaktadır. Raspberry Pi'nin Xbee modülüyle haberleşebilmesi için TTL dönüştürücü kullanılması gerekmektedir. Röle devresi sayesinde 240 V'ye kadar enerji kontrolü sağlanmaktadır. Sistemde çalışan program Python dili ile yazılmıştır. E-posta ile sulama sisteminin çalışacağı süre Raspberry Pi'ye gönderildikten sonra selenoid valfler açılarak sulama başlamaktadır; süre bittikten sonra valfler kapanıp sulama işlemi bitirilmektedir. Ultrasonik sensörler sayesinde depodaki su seviyesi eşik değerin altına düştüğünde röle ile su pompasının motoruna enerji verilerek depoya su takviyesi yapılmaktadır. Sistem sayesinde tarlalarda ve bahçelerde; ucuz, kolay ve verimli sulama sağlanabilmektedir (Agrawal ve Singhal, 2015).

Akkuş, yaptığı çalışmada IoT teknolojisinin kullanıldığı cihazlardan alınan bilgileri şifreleyerek iletmeyi amaçlamıştır. Bunun için Arduino üzerinde WEP şifreleme yapısını tasarlayarak ağda güvenli iletişim kurmuştur. Arduino ile sistemde kullanılan sensörlerden alınan veriler şifrelenerek yerel ağda depolanmak üzere sistemdeki Raspberry Pi'ye gönderilir. Veriler daha sonra Raspberry Pi ile internet aracılığıyla uzaktaki bir sunucuya gönderilir. Uzaktaki sunucu AES-128 algoritması ve verilerin şifresine ait anahtarı kullanarak şifreli veriyi çözmektedir. Sunucu şifresi çözülen verileri kendi ağındaki cihazlarla paylaşabilmektedir. Bu sayede veriler güvenli bir şekilde iletilmekte ve yetkisiz cihazlar veriye erişememekte; veriye erişebilse bile verinin içeriğine ulaşamamaktadır (Akkuş, 2016).

Şansal ve Temurtaş'ın çalışmasında Raspberry Pi ile akıllı ev otomasyonu dizayn edilmektedir. Ekibin çalışması tasarım olarak iki ana bölümden oluşmaktadır. Birinci bölümde eve yerleştirilen sıcaklık, nem, gaz, yangın, sıvı seviye kontrol ve hareket sensörleri ile evin durumu gözlemlenmektedir. İkinci bölümde ise eve hırsız girmesi

durumunda alarm verilmektedir. Sistemde sunucu olarak Raspberry Pi ve mikrodenetleyici olarak Arduino kullanılmıştır. Raspberry Pi’de kurulan web sunucu ile internetten evin durumu izlenebilmektedir. Ayrıca hırsızlık durumunda Raspberry Pi’ye kamera ile kayıt yapıp ev sahibine de e-posta ile bilgilendirme yapılmaktadır. Arduino Ethernet Shield üzerine kurulan web sunucu sayesinde ise istenilen zamanda sisteme bağlı kameralar izlenebilmekte ve sensörlerin durumu kontrol edilebilmektedir (Şansal ve Temurtaş, 2017).

Básaca-Preciado ve arkadaşları tarafından geliştirilen akıllı ev sistemi, bir Facebook Messenger botu ile entegre çalışmaktadır. Bota arkadaş listesinden messenger aracılığıyla belirli komutlardan oluşan sorular sorulmakta ve bot da cevap verebilmektedir. Bu sorular ortama ait bilgilerden (sıcaklık, nem, ortamdaki kameranın görüntüsü vb.) sorulardan oluşmaktadır. İletişim Facebook Messenger üzerinden kurulmaktadır. Arkadaş listesindeki kişilere sistem için kullanıcı kaydı oluşturulup kullanıcıya yetkiler verilebilmektedir. Kullanıcı, yetkisine göre evdeki kapılardan geçip evde dolaşabilmektedir. Ayrıca sisteme tanımlanan cep telefonu MAC adresi sayesinde, telefon, evdeki ağa dahil olduğunda ortamdaki ışıklar açılmakta ve telefon ağdan çıktığında ise ışıklar kapatılmaktadır (Básaca-Preciado vd., 2017).

Yüzgeç ve Aba tarafından geliştirilen sistem, Raspberry Pi tabanlı olup prototip bir ev üzerinde uygulanmaktadır. Sistemde bir web arayüzü üzerinden evdeki kameralardan görüntü alınabilmekte, ortamın sıcaklığı kontrol altında tutulabilmekte, evdeki ışıklar ve evin kapısı açılıp kapatılabilmektedir. Ayrıca bu sistemde hırsız ve gaz kaçağı alarmı da bulunmaktadır. (Yüzgeç ve Aba, 2017)

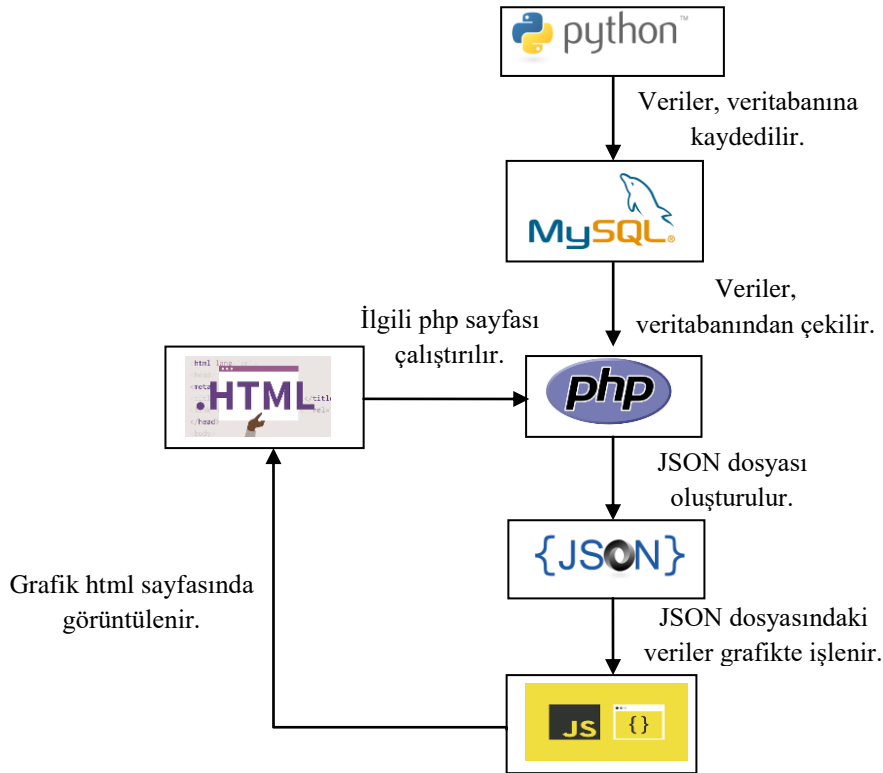
Uçar ve Uludağ tarafından yapılan çalışmada, Nesnelerin interneti (IoT) ile akıllı sınıf ve öğrenci takip sistemi tasarımı adlı Raspberry Pi tabanlı proje ile sınıfın ışıkları ve kapıları sistem veya yetkililer tarafından kontrol edilebilmektedir. Ayrıca sistem sayesinde öğrencilerin ders ve sınav yoklaması da alınabilmektedir. Yoklamalar RFID donanımlı kimliklerle veya parmak izi ile yapılabilmektedir. Sistem Öğrenci Bilgi Sistemi ile entegreli çalışmakta olup öğrencinin devam sınırına yaklaşması durumunda ve sınavdan başarısız olunması durumunda öğrencinin kayıtlı e-posta adresine mesaj gönderilmektedir (Uçar ve Uludağ, 2018).

## 2. KULLANILAN YAZILIMLAR ve DONANIMLAR

Bu bölümde, Raspberry Pi mini bilgisayarı ile uzaktan bir ortam izleme sisteminin geliştirilmesinde kullanılan yazılımlar ve donanımlar hakkında bilgiler verilmektedir.

### 2.1. Kullanılan Yazılımlar

Bu bölümde Python programlama dili, HTML, PHP, JavaScript ve JSON hakkında bilgiler verilmektedir. Yazılımların birbiriyle etkileşimi Şekil 2.1’de gösterilmektedir.



Şekil 2.1. Kullanılan yazılımların birbiriyle ilişkisi.

### Python Programlama Dili

1990 yılından beri geniş bir topluluk tarafından geliştirilen, öğrenilmesi kolay bir programlama dilidir. Bu dille geliştirilen programlar derlenmeden çalıştırılabilmektedir. Söz dizimi basit olduğundan program yazmak C ve C++ gibi dillerde program yazmaktan daha kolaydır. Bu dil, Windows, Linux ve MacOSX gibi platformlarda kolayca çalışmaktadır (Özgül, 2017). Bu özelliklerinden dolayı bu

projede Python programlama dili kullanılmıştır. Bu programlama dili, sensörlerden verileri almak ve alınan verileri veritabanına kaydetmek için kullanılmaktadır.

## **HTML**

İnternet sayfaları hazırlamak için kullanılan bir işaretleme dilidir. HTML bir programlama dili değildir. Google Chrome, İnternet Explorer ve Mozilla Firefox gibi internet tarayıcıları HTML kodlarını işleyerek bunları görsel şekle dönüştürürler. HTML ile internet sayfalarına yazılar ve görsel içerikler kolayca yerleştirilip konumlandırılabilir (w3schools.com, 2018). HTML, bu çalışmada genel olarak veritabanındaki verileri JavaScript tabanlı grafiklerde işlemek amacıyla kullanılmaktadır.

## **PHP**

İnternet tabanlı, nesne yönelimli, öğrenilmesi kolay ve içerisine HTML gömülebilen bir betik dilidir. Bu dil sunucu tabanlıdır. Yani derlemeler sunucu tarafında yapılmakta, kodlar kullanıcı tarafından görülmemekte; kullanıcı tarafında sadece derlemenin sonucu görülmektedir. Bu sayede internet sayfaları daha güvenli olmaktadır. PHP'nin geniş bir veritabanı desteği bulunmaktadır. Bu sayede veritabanı bağlantıları kolayca yapılabilir. PHP, temelde Linux tabanlı olup açık kaynak kodlu bir dildir (phpr.org, 2018). Açık kaynak kodlu olduğundan kitleler tarafından sürekli geliştirilmektedir. Bu sayede sürekli gelişip güncel kalmaktadır. PHP, bu tez çalışmasında internet sayfalarında yayınlanacak verileri veritabanından almak ve JSON dosyası oluşturmak için kullanılmaktadır.

## **JavaScript**

İnternet tarayıcılarında yaygın olarak kullanılan nesneye dayalı bir betik dilidir. İnternet sayfalarında görselliğin artırılması ve dinamik görüntülerin oluşturulması amacıyla HTML kodları içine gömülü olarak kullanılır ve istemci tarafında yorumlanır (Pekgöz, 2005). İnternette çok sayıda ücretsiz taslak JavaScript projesi bulunmaktadır. Bu projeler üzerinde birçok değişiklik yapılabilir ve bu sayede internet sayfalarında görsellik artırılabilir. Bu çalışmada JavaScript, grafiklerin oluşturulmasında kullanılmaktadır.

## **JSON (JavaScript Object Notation)**

Veri deęişim biçimidir. XML JavaScript ile etkin bir şekilde kullanılmadığından, JavaScript uygulamalarında sıklıkla tercih edilmektedir. JSON ile düşük boyutlarda veri alışverişı yapılabilmektedir. Bu türdeki veriler; anahtar (key) ve deęer (value) olmak üzere iki temel bileşenden oluşur. Anahtarda nesnenin özellięi, deęerde nesnenin özellięinin deęeri belirtilir.

```
{  
  "Ad": "Mehmet",  
  "Soyad": "BALCI"  
}
```

Bu dosya türü grafiklerde gösterilecek verilerin oluşturulmasında kullanılmaktadır. Grafiklerin bulunduğu internet sayfaları yenilendiğinde, ilgili grafik için yeni bir JSON dosyası oluşmaktadır.

## **MySQL**

Genel Kamu Lisanslı (GNU) ilişkisel bir veritabanı yönetim sistemidir. Açık kaynak kodlu olduğundan geliştirilmeye açıktır. Windows, Linux ve UNIX işletim sistemlerinde kullanılabilir. Farklı karakter setlerini desteklemekte ve bu dillerde hata mesajları da vermektedir. Çalışma hızı yüksek olup kolay sorgulama alt yapısı bulunmaktadır. Güçlü bir kullanıcı erişim yetkilendirme sistemine sahiptir. Ekstra olarak yüklenen phpMyAdmin arayüzü sayesinde web üzerinden sorgularla veya arayüzdeki araçlarla veritabanında istenilen deęişiklikler yapılabilmektedir. Veritabanına ait ilişkiler diyagramı (ER) Şekil 2.2'de gösterilmektedir. Şekilde görülebildięi gibi tablolar arasında herhangi bir ilişki bulunmamaktadır.



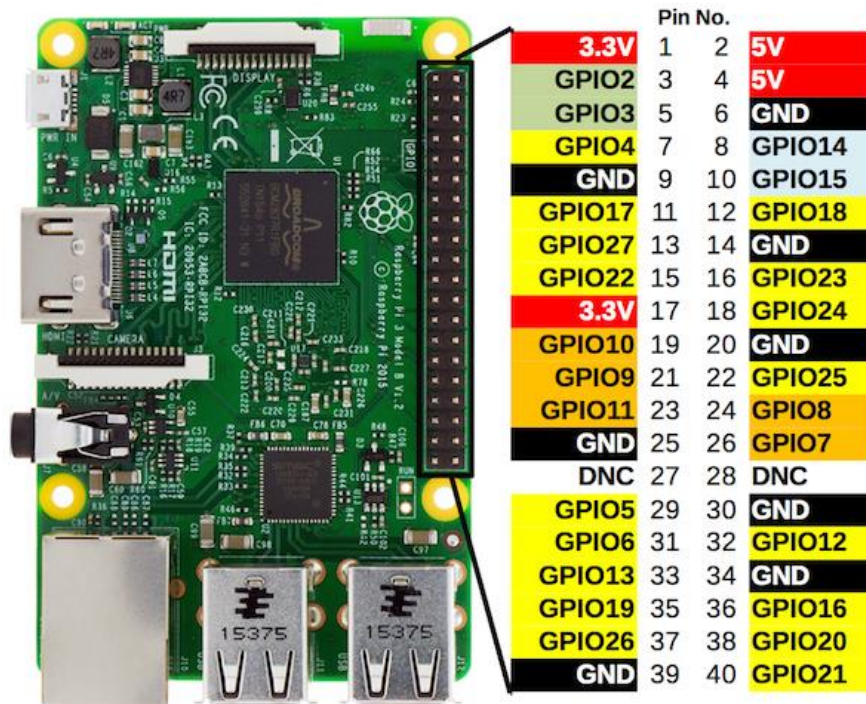
Şekil 2.2. Veritabanına ait ilişkiler diyagramı (ER).

## 2.2. Kullanılan Donanımlar

Bu bölümde Raspberry Pi, sıcaklık-nem, gaz, su, mesafe, alev, ışık, hareket sensörleri ile ADC (Analog Sayısal Dönüştürücü) hakkında bilgi verilecektir.

### 2.2.1. Raspberry Pi Mini Bilgisayarı

Kredi kartı büyüklüğünde ve tek bir karttan oluşan mini bir bilgisayardır. Piyasada A ve B olmak üzere iki modeli bulunmaktadır. Bu çalışmada “Raspberry Pi 3 B Ver1.2” modeli kullanılmıştır. Bu modelde 4 adet USB 2.0, bir adet HDMI ve bir adet ethernet portu bulunmaktadır. Ayrıca ARMv7 1200 MHz CPU ve 1 G RAM donanımına sahiptir. Cihaz Linux tabanlı işletim sistemleriyle çalışmaktadır. İşletim sistemi bir SD karta yüklenip, SD kart üzerinden çalıştırılmaktadır. Cihaz 5V beslemeyle çalışmaktadır ve bu sayede güç tüketimi oldukça düşüktür. Kart üzerinde sensörler gibi donanımları beslemek için 3.3V, 5V güç beslemeleri ve giriş-çıkış pinleri bulunmaktadır. Bu çalışmada 3.3V ve 5V pinleri sensörlere ve ADC’ye enerji vermek, giriş-çıkış pinleri sensörlerden veri almak ve ethernet portu da internet erişimi için kullanılmaktadır. Bir bilgisayarda yapılan hemen hemen tüm işlemler bu cihazda da yapılabilmektedir. Şekil 2.3’de Raspberry Pi B mini bilgisayarı ve portları gösterilmiştir.

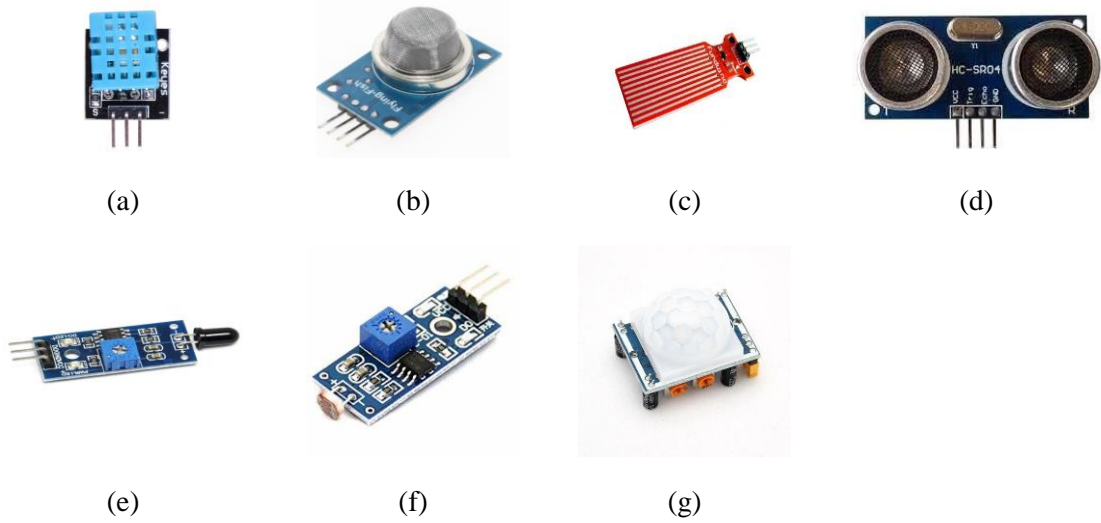


Şekil 2.3. Raspberry Pi giriş-çıkış pinleri ve diğer bağlantı portları (BMOW, 2019).

Raspberry Pi; maliyeti düşük, çok sayıda giriş-çıkış portuna sahip, boyutları düşük ve bir bilgisayarın özelliklerine sahip olduğundan bu çalışmada kullanılmaktadır. Bu özelliklerinden dolayı cihaz, hava durum istasyonu, robot beyni, termal yazıcı, kanatlı mini hava araçları ve çocuk oyuncukları gibi birçok alanda kullanılabilir.

### 2.2.2. Sensörler

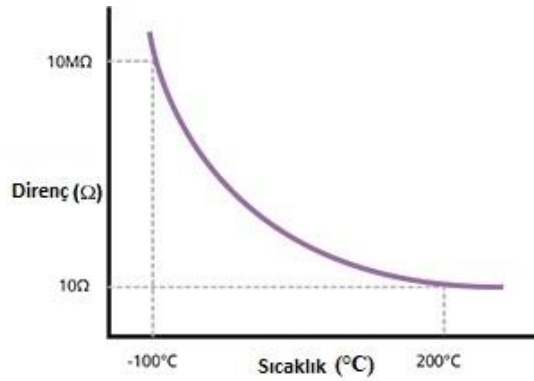
Çalışma kapsamında sıcaklık/nem sensörü, gaz sensörü, su sensörü, mesafe sensörü, alev sensörü, ışık sensörü ve hareket sensörü kullanılmıştır.



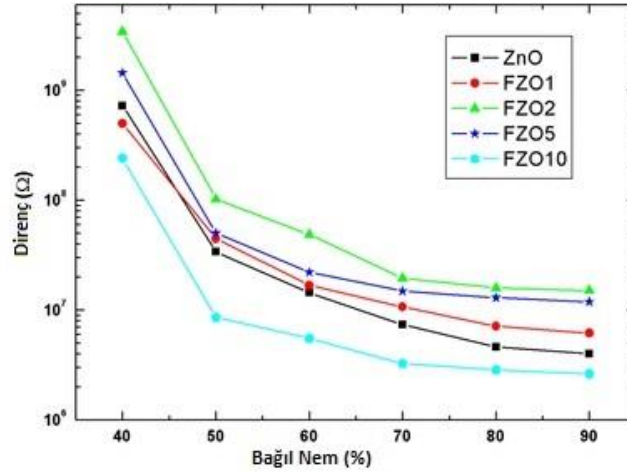
**Şekil 2.4.** Çalışmada kullanılan sensörler; (a) sıcaklık/nem sensörü, (b) gaz sensörü, (c) su sensörü, (d) mesafe sensörü, (e) alev sensörü, (f) ışık sensörü, (g) hareket sensörü.

### ***Sıcaklık / Nem Sensörü***

Ortamdaki sıcaklık ve nemi ölçmek için DHT11 sensörü kullanılmıştır. Sensör Şekil 2.4 (a)'da gösterilmiştir. DHT11 tüm kapalı ortamlardaki sıcaklık ve nem ölçümlerinde kullanılabilir. Sensör içerisinde sıcaklık ölçümü için termistör ve nem ölçümü için ise yapısında ZnO (Çinko Oksit) olan kapasitif nem sensörü bulunur. Sensörlerden alınan veriler dijital çıkış şeklinde alınabilmektedir. %20-80 oranındaki nem ölçümleri için %5 hassasiyete; 0-50 °C arasındaki sıcaklık ölçümleri için  $\pm 2$  °C hassasiyete sahiptir. Şekil 2.5'te çalışmada kullanılan DHT11 sensörünün yapısındaki direncin sıcaklık grafiği, Şekil 2.6'da kapasitif nem sensörünün nem grafiği gösterilmektedir.



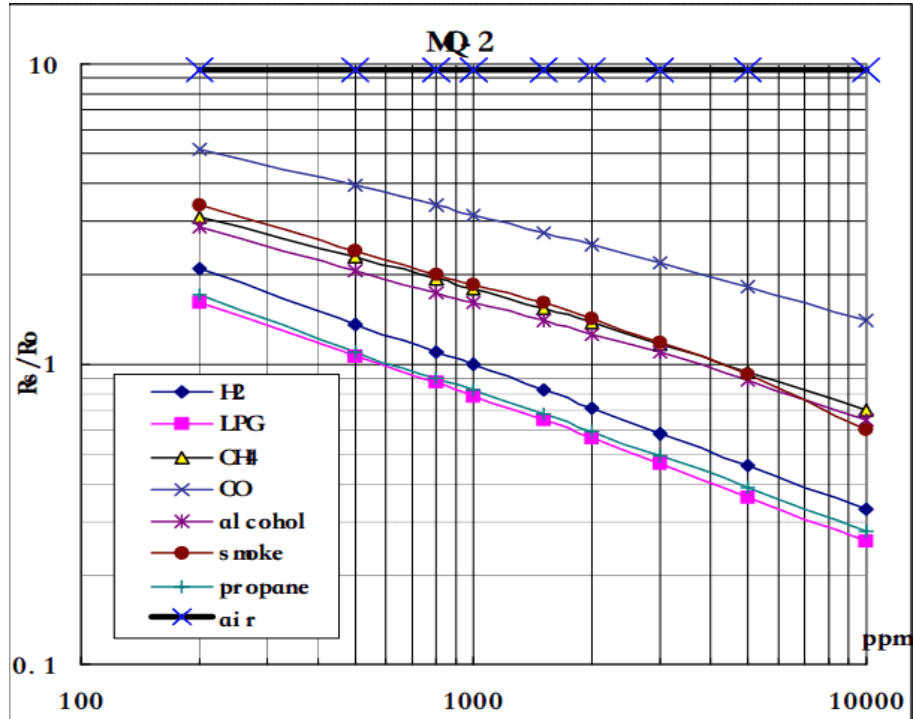
**Şekil 2.5.** DHT11 sensörünün yapısındaki dirence bağlı sıcaklık grafiği (Last Minute Engineers, 2019).



**Şekil 2.6.** DHT11 sensörünün yapısındaki dirence bağlı nem grafiği (Algün, 2018).

### **Gaz Sensörü**

Ortamda bulunan LPG ve karbon monoksit (CO) gibi yanıcı gazlar ile dumanı ölçmek için MQ-2 sensörü kullanılmıştır. Sensör analog çıkış vermektedir. Çıkışlar analog olduğu için ADC (Analog Digital Converter) kullanılmaktadır. Bu sensör, evlerde gaz kaçakları, işyerlerinde temiz hava kontrolünde kullanılabilir. Sensörden sağlıklı veri alınabilmesi için temiz ortamda önce kalibrasyon yapılmalıdır. Kalibrasyon ortamdaki gazlar normal seviyede iken alınan ölçümlerin ortalaması alınarak yapılmaktadır ve elde edilen ortalama değer, ortamın normal değeri olmaktadır. Bu tez çalışmasında kullanılan gaz sensörü, Şekil 2.4. (b)'de gösterilmektedir. Sensörün ölçüm grafiği Şekil 2.7'de verilmektedir.



Şekil 2.7. MQ-2 gaz sensörüne ait ölçüm grafiği (Seeed, 2019).

### *Su Sensörü*

Birbirine paralel olarak çekilmiş iletken tellerin su ile temasıyla çıkış voltajı oluşur. Sensöre su teması olmasıyla normalde açık devre olan paralel teller, kapalı devre oluşturarak sensörün çıkışında gerilim oluşmasını sağlamaktadır. İletkenlik özelliği bulunan ve sensörün yapısına zarar vermeyen tüm sıvılar için kullanılabilir. Sensör, seralarda sulama sistemlerinde, ev ve işyerlerinde su basmalarının tespiti gibi su algılamasının yapılmak istendiği her yerde kullanılabilir. Sensör, Şekil 2.4. (c)'de gösterilmektedir. Sensörün oluşturduğu çıkış sinyalinin sistem tarafından kullanılmasına ait grafik Şekil 2.8'de verilmektedir; şekilde su teması olmaması durumu 0, su teması olması durumu ise 1 olarak ifade edilmektedir.



Şekil 2.8. Su sensörüne su temasının olduğu ve olmadığı durumlara ait grafik.

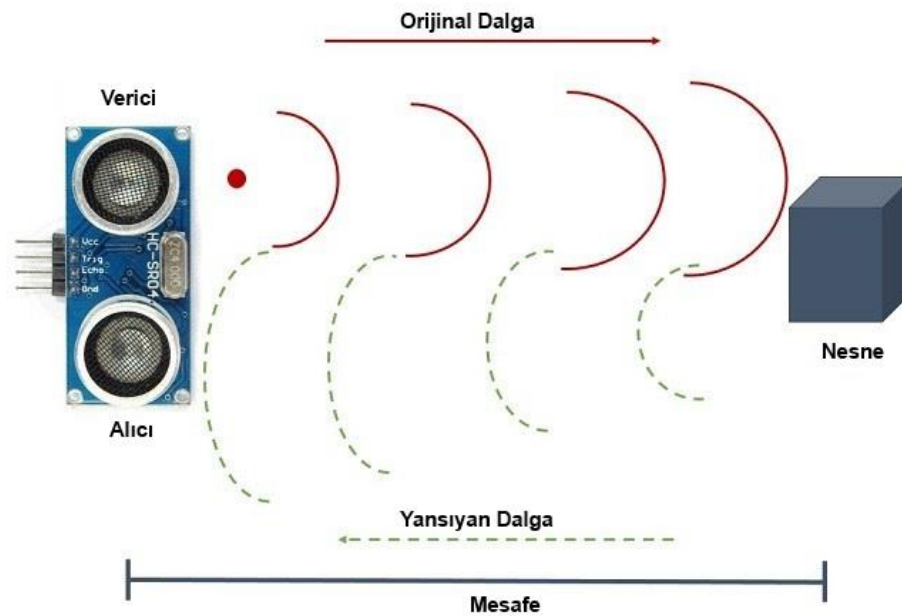
### Mesafe Sensörü

Su basmalarında suyun yüksekliğini ölçmek için mesafe sensörü kullanılmaktadır. Bu çalışmada mesafe sensörü olarak için HC-SR04 sensörü kullanılmaktadır. 2 cm'den 400 cm'ye kadar ölçüm yapabilen ultrasonik sensördür. 15 derecelik görme açısına sahiptir. Sensörde mesafe ölçümü ses dalgaları (sonar) kullanılarak yapılmaktadır. Tetikleme (Trig) pininden gönderilen sinyalin Eko (Echo) pininden okunmasıyla mesafe ölçümü yapılmaktadır. Mesafe sensörleri su depolarında kalan su miktarı, arabaların park sensörü gibi alanlarda mesafe ölçümünde kullanılabilir. Sensör, Şekil 2.4. (d)'de gösterilmektedir. Mesafe ölçümüne ait formül aşağıda verilmektedir;

$$\text{Geçen Süre} = \text{Bitiş Süresi} - \text{Başlangıç Süresi}$$

$$\text{Mesafe} = (\text{Geçen Süre} \times 34300) / 2$$

Formülde kullanılan “Başlangıç Süresi” ses dalgasının ilk gönderilme zamanı, “Bitiş Süresi” ses dalgasının ses geri geldiği zamandır; 34300 sayısı ses dalgasının hızıdır ve birimi cm/s'dir; ses dalgasının gidişi ve geri dönüşü olduğu için 2 sayısına bölünmektedir. Sensörün mesafe ölçümü, Şekil 2.9'da gösterilmektedir.



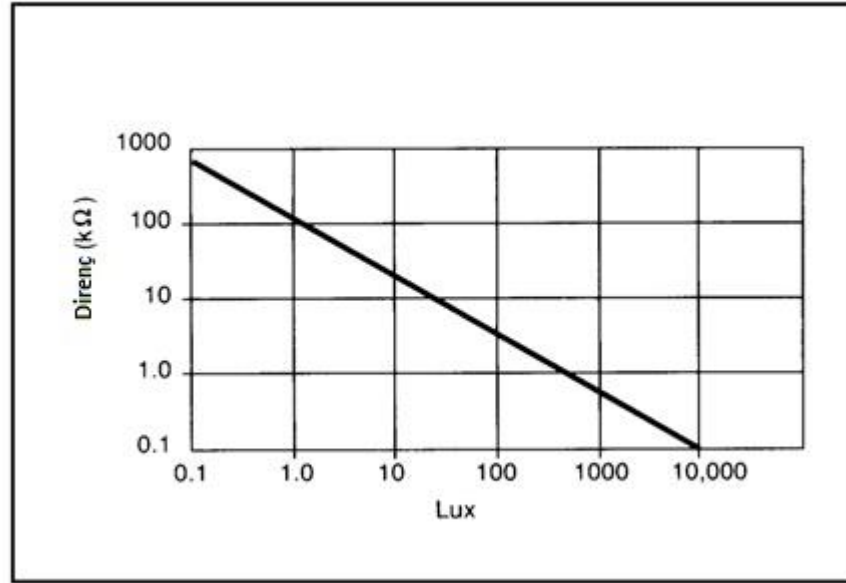
**Şekil 2.9.** Mesafe sensörünün ses dalgası ile uzaklık ölçümü (Faranux Electronics, 2019).

### ***Alev Sensörü***

760 nm – 1100 nm dalga boyları arasındaki ateşi tespit edebilmektedir. Sensörde kızılötesi (IR) alıcı bulunmaktadır. Sensörün üzerindeki trimpot ile hassasiyet ayarı yapılabilmektedir. Bu tez çalışmasında kullanılan sensör, Şekil 2.4. (e)'de gösterilmektedir. Sensör, dijital çıkış (0/1) vermektedir. 0 durumu alev tespit edilmediği, 1 durumu ise alev tespit edildiği durumdur; bu durumlar Şekil 2.8'de verilmiştir. Sensör, 60 derecelik görme açısına sahiptir.

### ***Işık Sensörü***

Ortamdaki ışığın şiddetini ölçmek için LDR (Light Dependent Resistor) kullanılmıştır. Sensör, Şekil 2.4. (f)'de verilmektedir. Sensör analog çıkış vermektedir. Çıkış akımı sensörün üzerine düşen ışığın şiddetine göre değişmektedir. Işık şiddeti arttıkça sensörün direnci azalmakta ve çıkış akımı da yükselmektedir. Sensörün direncinin ışığa bağlı değişimi Şekil 2.10'da gösterilmektedir.



**Şekil 2.10.** Gerilimin ışığın yoğunluğuna bağlı değişimi(CircuitDigest, 2019).

### ***Hareket Sensörü***

Hareket algılama sensörü olarak PIR (Passive Infrared Sensor) kullanılmıştır. Sensör, Şekil 2.4. (g)'de gösterilmektedir. Sensör isminden de anlaşılacağı gibi kızılötesi ile algılama yapmaktadır. Bu sensör 3-5 metreye kadar algılama alanına ve

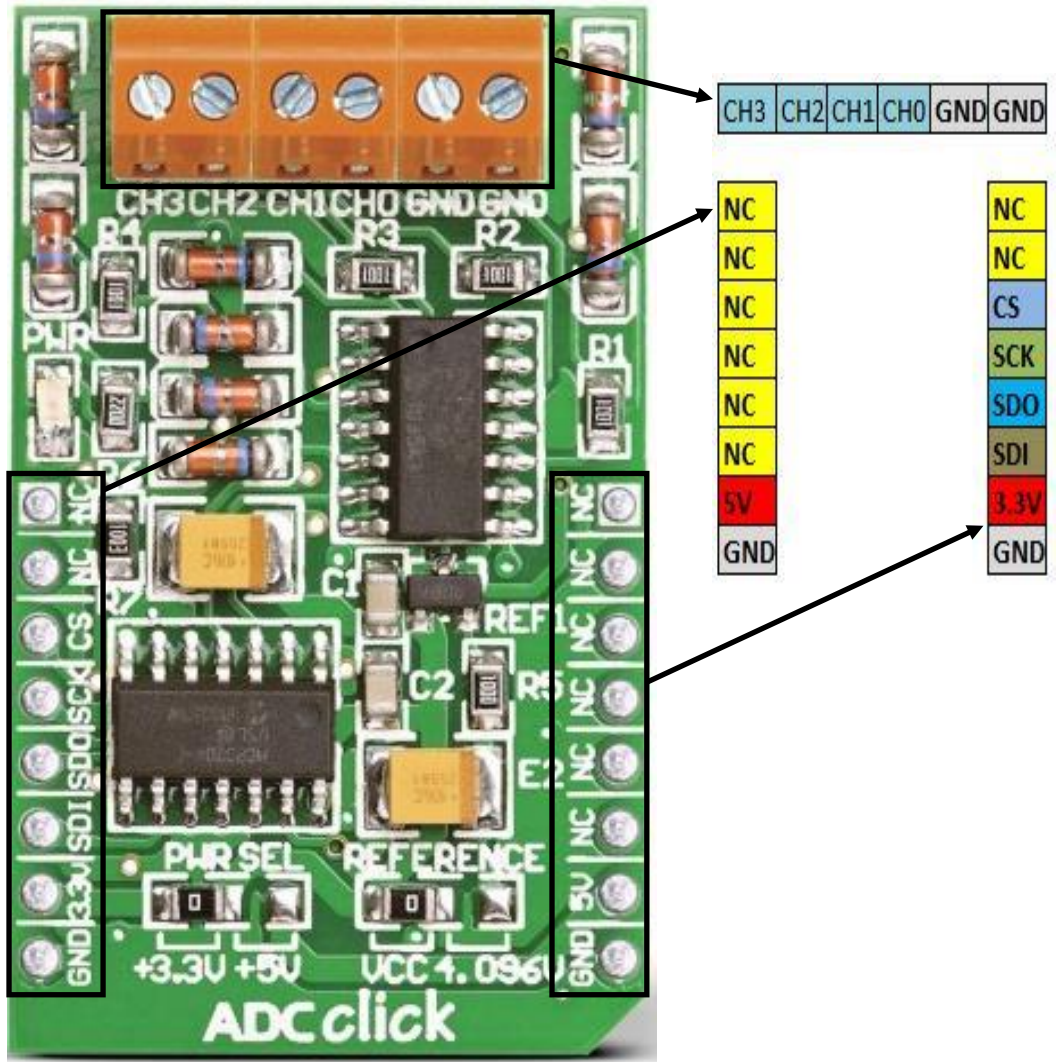
140 derecelik görme açısına sahiptir. Sensör IR'ye duyarlı iki yuvadan oluşmaktadır. Sensör algılama yapmadığı zaman iki yuva da ortamdan aynı miktarda IR almaktadır. Hareket olduğunda önce bir yuvadaki IR miktarı değişeceğinden iki yuva arasında fark meydana oluşmakta ve pozitif diferansiyel değişim meydana gelmektedir. Hareketi meydana getiren cisim sensörün algılama açısından çıktığında ise negatif değişim meydana gelmektedir. Hareket, bu iki değişim sayesinde algılanmaktadır. Sensör, bu tez çalışmasında ortamda hareket olup olmadığını (0 veya 1) tespit etmek için kullanılmaktadır. Ortamda hareket olup olmadığına ait durum Şekil 2.8'de verilmiştir.

### 2.2.3. Analog Sayısal Dönüştürücü (Analog Digital Converter, ADC)

Analog sinyal üreten sensörlerin çıkış sinyallerini işleyebilmek ve ölçümleri yapabilmek için ADC kullanılmıştır. Bunun için MCP3204 tercih edilmiştir. Bu ADC modülü SPI bağlantı arayüzüne ve 4 kanala sahiptir. Cihaz 12 bitlik analog-dijital dönüştürme ve saniyede 50k örnekleme yapabilmektedir. 3V veya 5V ile çalışabilmektedir. Dönüştürücünün Raspberry Pi'ye bağlantı pinleri Çizelge 2.1'de verilmiştir. Dönüştürücünün dış görünüşüne ait görüntü Şekil 2.11'de verilmektedir.

**Çizelge 2.1.** ADC'nin Raspberry Pi'ye bağlanması.

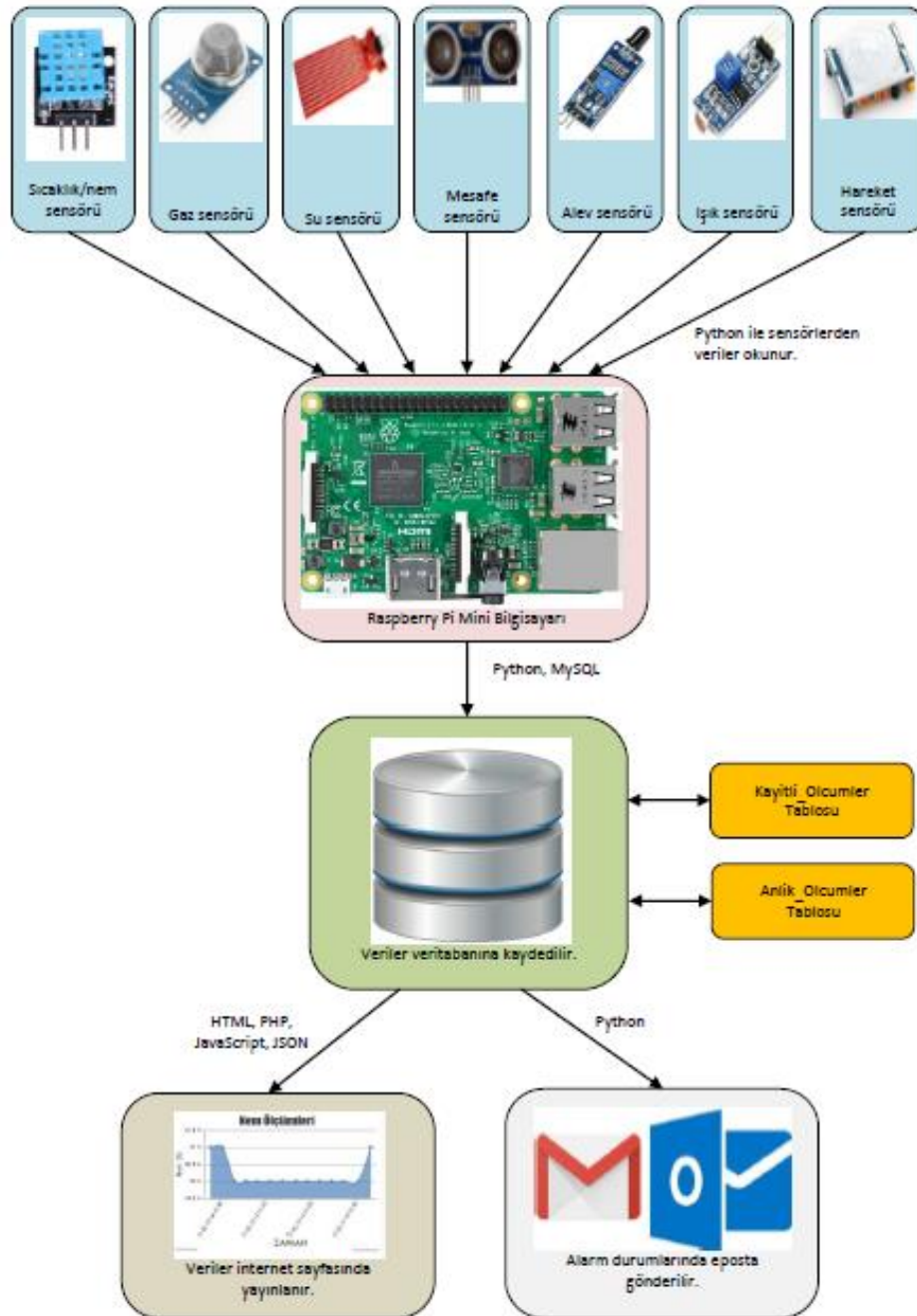
ADC	RASPBERRY Pİ
GND	GND
3.3 V	3.3 V
SDI	SPI0 MOSI
SDO	SPI0 MISO
SCK	SPI0 SCLK
CS	SPI0 CS0



Şekil 2.11. ADC ve pinlerine ait görüntü.

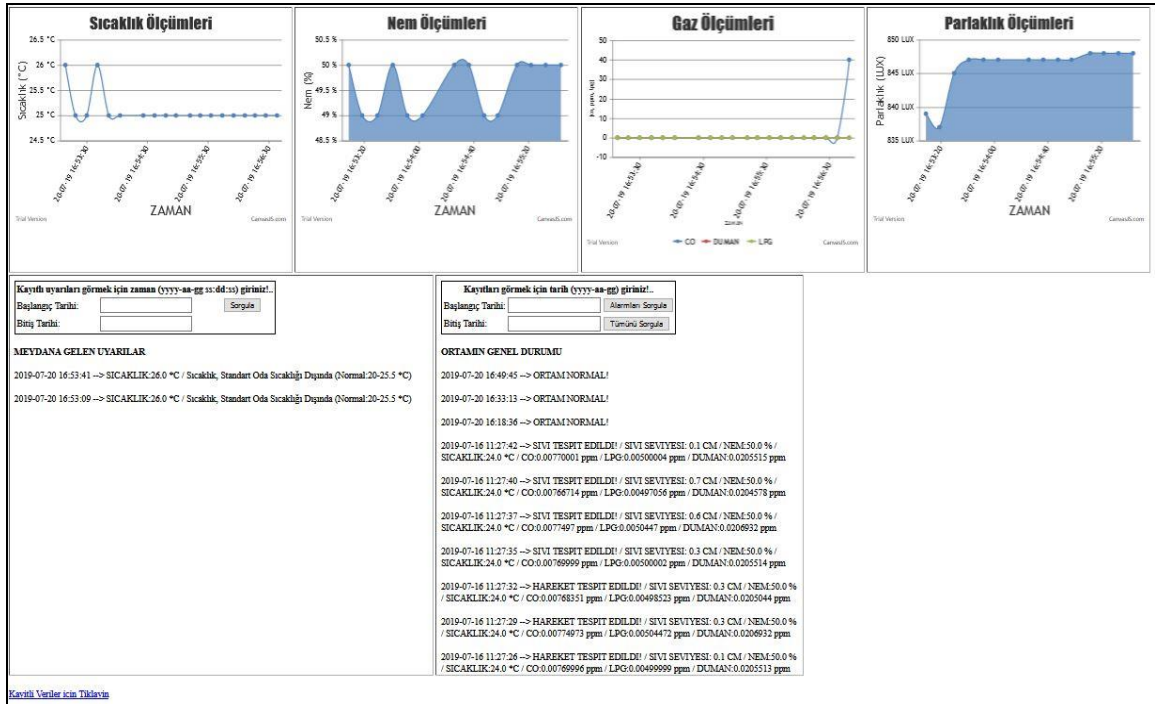
### 3. RASPBERRY PI İLE ORTAM VERİLERİNİN İZLENMESİ

Sistem, Python diliyle yazılan programla ortama ait verileri işleyip veritabanına kaydetmektedir ve bu veriler web uygulamaları ile internet sayfalarında yayınlanmaktadır. Sistemin çalışmasına ait genel diyagram Şekil 3.1’de verilmektedir.



Şekil 3.1. Sistemin çalışmasına ait genel diyagram.

Veritabanında ortama ait sıcaklık, nem, LPG, CO, duman ve parlaklık bilgilerine ait veriler, kalıcı ve geçici olmak üzere iki farklı tabloda tutulmaktadır. Veriler, internet tarayıcılarında grafikler şeklinde görüntülenebilmektedir. PHP ile oluşturulan JSON dosyalarının, JavaScript ile hazırlanmış grafikler üzerinde işlenmesiyle son kullanıcının ortama ait verileri grafiksel olarak görmesi sağlanmaktadır. Ana sayfada geçici verilere ait tüm grafikler, kalıcı verilerin bulunduğu ana sayfaya ait bağlantı, ortamın güncel durumunu (alarm olup olmadığı) ve medyana gelen uyarıları bildiren pencereler bulunmaktadır. Ortama ait her bilgi için ayrı birer grafik bulunmaktadır. Anlık verilere ait sayfalar on saniyelik otomatik yenileme süresine sahip iken, kayıtlı verilerin bulunduğu sayfalar on beş dakikalık otomatik yenileme süresine sahiptir. Sayfaların otomatik yenilenme süreleri, ilgili tablodaki verinin veritabanına kaydedilme süresine eşittir. Kayıtlı verilerin bulunduğu sayfalarda seçilen tarih aralıklarına göre ortama ait ortalama, maksimum ve minimum değerler de görülebilmektedir. Güncel durumun görüntülediği pencerede meydana gelen alarmlara ait bilgi bulunmaktadır. Uyarılar penceresinde ortamın fiziksel değerleri, normal değerlerin dışına çıktığında söz konusu değerle ilgili bildirimler verilmektedir. Sistemin ana sayfasına ait ekran görüntüsü Şekil 3.2’de gösterilmektedir.



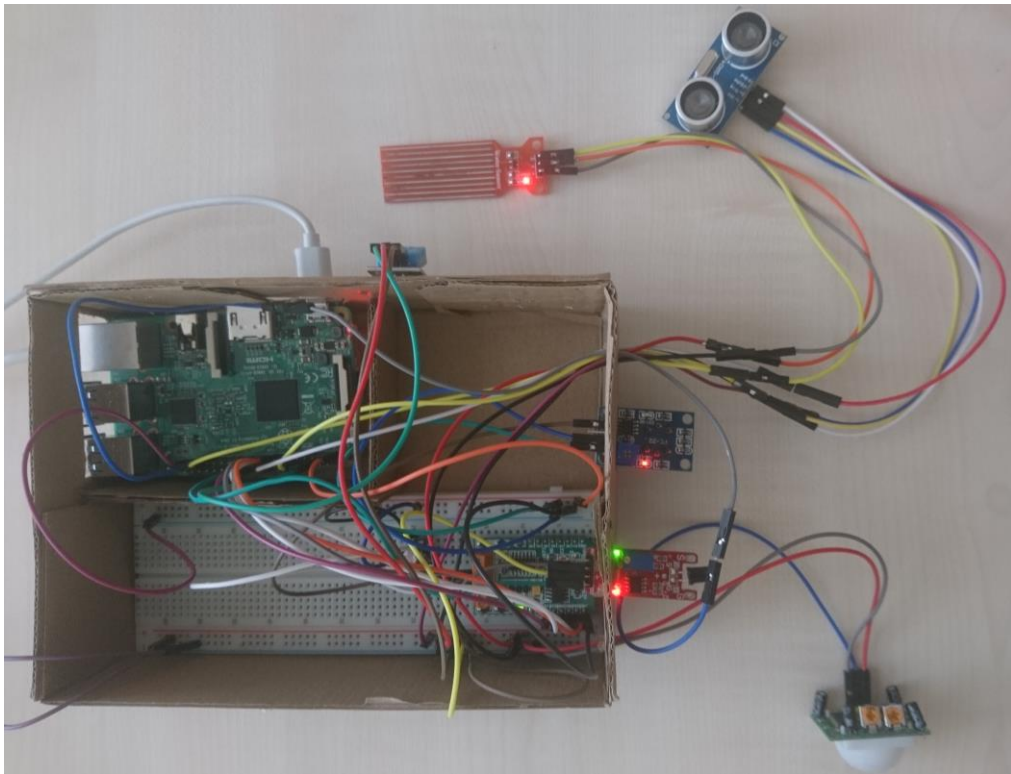
Şekil 3.2. Sistemin ana sayfasına ait ekran görüntüsü.

### 3.1. Sensörlerden Veri Alınması

Python uygulamalarının çalışabilmesi için “python3-pip” paketinin yüklü olması gerekmektedir. Normalde bu paket Raspberry Pi’ye yüklenen tam sürümlü işletim sistemlerinde bulunmaktadır. Paketin yüklü olmaması durumunda aşağıdaki komutlarla paket yüklenebilmektedir;

```
sudo apt-get install python3-pip
```

Bu çalışmada kullanılan sensörler ve Raspberry Pi Şekil 3.3’de görsel olarak gösterilmekte ve programın çalışma diyagramı da Şekil 3.4’te verilmektedir. Sistem, Bilecik Şeyh Edebali Üniversitesi Bilgi İşlem Daire Başkanlığı Teknik Servisinde kurulmuş olup sürekli olarak çalışmaktadır.

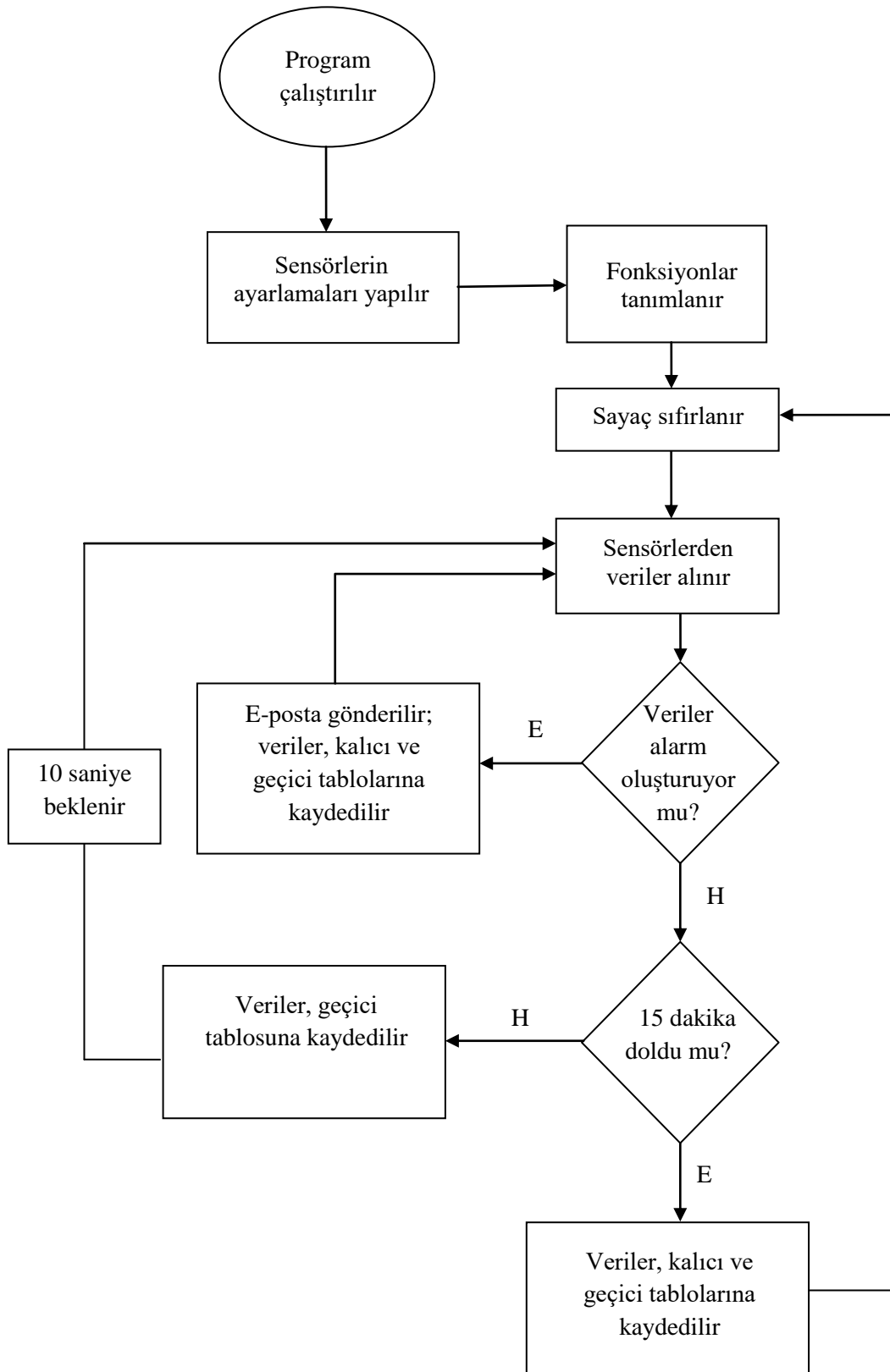


Şekil 3.3. Çalışma kapsamında önerilen sistemin fotoğrafı.

**Sıcaklık ve Nem Sensöründen Veri Alınması:** Ortamdaki nem ve sıcaklık değerlerini DHT11 sensörüyle elde etmek için açık kaynak kodlu “Adafruit\_DHT” kütüphanesinin yüklü olması gerekmektedir. Kütüphaneyi yüklemek Raspberry Pi’nin

terminalinde aşağıdaki kodların sırasıyla çalıştırılması gerekmektedir. (Raspberrypi.org, 2018)

```
sudo apt-get update  
sudo apt-get install build-essential python-dev  
git clone https://github.com/adafruit/Adafruit_Python_DHT.git  
cd Adafruit_Python_DHT  
sudo python setup.py install  
sudo python3 setup.py install
```



Şekil 3.4. Programın genel çalışma diyagramı.

“Adafruit\_DHT” kütüphanesi yüklendikten sonra, kütüphanenin ana programda (ortam\_izleme.py) kullanılabilmesi için “import Adafruit\_DHT” nın yazılması gerekmektedir.

Sensörün bağlanacağı pin aşağıdaki kodlarla tanımlanmaktadır;

```
sensor=Adafruit_DHT.DHT11
gpio=14
```

Programda ölçülen sıcaklık ve nem değerleri aşağıdaki kod yardımıyla, nem değerleri “humidity” ve sıcaklık değerleri ise “temperature” değişkenlerine atanmaktadır.

```
humidity, temperature = Adafruit_DHT.read_retry(sensor, gpio)
```

Ölçülen değerlerin terminalde gösterilmesi aşağıdaki kodla tanımlanmaktadır;

```
print("\nSicaklik={0:0.1f}*C Nem={1:0.1f}%".format(temperature, humidity))
```

Ölçülen değerler terminalde Şekil 3.5'de verildiği gibi gözükmektedir.

```
Sicaklik=24.0*C Nem=49.0%
```

Şekil 3.5. Sıcaklık ve nem bilgisinin terminalde görünümü.

**Gaz Sensöründen Verilerin Alınması:** MQ-2 sensörüyle ortamdaki LPG, karbonmonoksit (CO) ve dumanın yoğunluğunu ölçmek için “ortam\_izleme.py” programında sensöre ait kalibrasyonun ve ölçüme ait hesaplama ayarlarının yapılması gerekmektedir. Bu ayarlar “mq.py” dosyasında bulunmaktadır. Bu ayarların “ortam\_izleme.py” programında yapılmasına ait kodlar aşağıda gösterilmektedir;

```
from mq import *      #koduyla mq.py'deki kodlar cagriliyor
mq = MQ();           #koduyla sensör ölçüm yapmaya başlamaktadır
```

Program çalıştırıldığında ölçümlere başlanmadan önce otomatik kalibrasyon yapılmaktadır. GAS\_LPG, GAS\_CO ve GAS\_SMOKE değerlerinin doğru ölçülebilmesi için kalibrasyon kodlarındaki, sensörün bağlantısında kullanılan RL\_VALUE direncinin elle girilmesi gerekmektedir. Kalibrasyonun yapıldığı kodlar aşağıda gösterilmektedir;

```

val = {}
read = self.MQRead(self.MQ_PIN)
val["GAS_LPG"] = self.MQGetGasPercentage(read/self.Ro, self.GAS_LPG)
val["CO"] = self.MQGetGasPercentage(read/self.Ro, self.GAS_CO)
val["SMOKE"] = self.MQGetGasPercentage(read/self.Ro, self.GAS_SMOKE)

```

Sensörün çıkış bilgisi analog olduğundan Analog-Digital Converter (ADC) kullanılmaktadır. ADC'nin çalışması için ADC'ye ait kütüphanenin "mq.py" programına yüklenmesi gerekmektedir. Bu çalışmada kullanılan ADC dört kanallı olduğundan sensörün bağlı olduğu kanalın tanımlanması gerekmektedir. Kütüphanenin yüklenmesi ait kodlar aşağıda gösterilmektedir.

```

from MCP3008 import MCP3008
#ve ADC'nin pin ayarlamasının yapılması;
def __init__(..., ..., analogPin=0):
... = ...
self.MQ_PIN = analogPin
self.adc = MCP3008()

```

Kalibrasyon ve yapılan ölçümlerin terminaldeki ekran görüntüleri Şekil 3.6 ve 3.7'de gösterilmiştir.

```

Gaz Sensörü Kalibre Ediliyor...
Calibration is done...

Ro=-0.027964 kohm

```

**Şekil 3.6.** Gaz sensörünün kalibrasyonunun yapılması.

```

LPG: 0.00764421 ppm, CO: 0.00495002 ppm, Smoke: 0.0203924 ppm

```

**Şekil 3.7.** Gaz ölçümü sonuçlarına ait ekran görüntüsü.

**Su Sensöründen Veri Alınması:** Su sensöründe paralel iki direnç arasında kısa devre oluşmasıyla sensörün çıkışında gerilim oluşur. Çıkış mantıksal değer (sıfır veya bir) olarak kullanılmaktadır. Sıfır (0) durumu ortamda su bulunmadığı, bir (1) durumu ise ortamda su bulunduğu anlamına gelmektedir. Sensöre ait bilginin alınacağı pin aşağıdaki kodla tanımlanmaktadır.

```
GPIO.setup(13, GPIO.IN)
```

Ortamda su olup olmadığı (k sensörün çıkış bilgisi olmak üzere) kontrol edilmekte ve su tespit edildiğinde terminalde bilgilendirme yapılmasına ait kod aşağıdadır.

```
elif k==1:  
print "\nSivi Temasi Var!..",k
```

**Mesafe Sensöründen Veri Alınması:** Mesafe sensörü ortamda su bulunduğu ortamdaki suyun yüksekliğini ölçmek için kullanılmaktadır. Su seviyesinin bilinmesi sayesinde su sensörünün her su tespitinin tehlike oluşturup oluşturmadığı analiz edilebilmektedir. Sensöre ait pin tanımlamaları aşağıdaki kodlarla yapılmaktadır.

```
GPIO_TRIGGER = 18  
GPIO_ECHO = 24  
GPIO.setup(GPIO_TRIGGER, GPIO.OUT)  
GPIO.setup(GPIO_ECHO, GPIO.IN)
```

Mesafe ölçümü, program içerisinde bir fonksiyonla yapılmaktadır. Fonksiyona ait kodlar aşağıdaki gibidir. Ölçümü yapılan mesafenin terminalde gösterildiği görüntü Şekil 3.8’de gösterilmiştir.

```
def distance():  
GPIO.output(GPIO_TRIGGER, True)  
time.sleep(0.00001)  
GPIO.output(GPIO_TRIGGER, False)  
StartTime = time.time()  
StopTime = time.time()  
  
while GPIO.input(GPIO_ECHO) == 0:  
StartTime = time.time()  
while GPIO.input(GPIO_ECHO) == 1:  
StopTime = time.time()  
  
TimeElapsed = StopTime - StartTime  
distance = (TimeElapsed * 34300) / 2  
return distance
```

```
Olculen Yukseklik = 0.4 cm
```

**Şekil 3.8.** Su seviyesinin ölçümü.

**Alev Sensöründen Veri Alınması:** Sensör tarafından ortamda alev algılandığında çıkış gerilimi meydana gelmektedir. Oluşan değer mantıksal olarak işlenmektedir (sıfır veya bir). Sıfır (0) ortamda alev olmadığı, bir (1) ise ortamda alev bulunduğu anlamında gelmektedir. Sensör ile ilgili kodlar aşağıdaki gibidir.

```
GPIO.setup(7,GPIO.IN) #koduyla pin tanımlanmakta
if (GPIO.input(7)):    #koduyla da alev olup olmadığı kontrol edilmekte
```

**Işık Sensöründen Veri Alınması:** Karanlık ve aydınlık ortamda sensörün çıkışında ölçülen en yüksek ve en düşük gerilime karşılık ölçülen parlaklık değerlerine göre ortamın parlaklığı ölçülmektedir. Ölçülen parlaklık değerine göre ortamın karanlık veya aydınlık olduğu tespit edilebilmektedir. Ortamdaki aydınlatma sistemi, çalışma ile birleştirilip lambaların otomatik açılıp kapanması sağlanabilir (bu çalışmada sadece parlaklık ölçümü yapılmaktadır). Sensör analog çıkış ürettiği için ADC kullanılmaktadır. Verilerin ADC'nin hangi kanalından alınacağını tanımlanmasına ait kodlar aşağıdadır.

```
def ReadChannel(channel):
    adc = spi.xfer2([1,(8+channel)<<4,0])
    data = ((adc[1]&3) << 8) + adc[2]
    return data
```

Sensörün çıkış voltajının ışık şiddetine dönüştürülmesi aşağıdaki kodla yapılmaktadır.

```
def ConvertVolts(data,places):    #LDR için
    volts = (data * 3.3) / float(1023) # ADC'den veri LDR'nin 3.3 besleme
                                         # gerilimi ile çarpılıp 1023 bölüldüğünde
                                         #LDR'nin çıkış gerilimi elde edilir.
    volts = round(volts,places)
    return volts
```

Işık şiddetinin ortamdaki parlaklığa dönüştürülmesi, çıkış voltajı ve ortam parlaklığının terminalde görüntülenmesine ait kod aşağıda gösterilmiştir ve parlaklığın terminaldeki görünümü Şekil 3.9'da gösterilmektedir.

```
print("Parlaklik: {} ({}V)".format(1200-light_level,light_volts))
lux=1200-light_level
```

```
Parlaklik: 792 LUX
```

**Şekil 3.9.** Ortam parlaklığının terminaldeki ekran görüntüsü.

**Hareket Sensöründen Veri Alınması:** Ortamda sensörün görüş açısında bir hareket meydana geldiğinde çıkış gerilimi oluşmaktadır. Oluşan değer mantıksal olarak işlenmektedir (sıfır veya bir). Sıfır ortamda herhangi bir hareketin olmadığı anlamına gelmekte iken, bir ise ortamda hareket algılandığı anlamına gelmektedir. Sensörle ilgili yazılan kodlar aşağıdaki gibidir.

```
GPIO.setup(26, GPIO.IN) #hareket sensörünün çıkış pininin tanımlanması
i=GPIO.input(26) #elif için sensörün çıkış bilgisinin değişkene atanması
elif i==1: #hareket olup olmadığının kontrol edilmesi
```

### 3.2. Verilerin Veritabanına Kaydedilmesi

Veriler iki farklı tabloda tutulmaktadır. İki tablo da aynı yapıda olup, verilerin kaydedilmesine ait zaman periyotları farklıdır. Birinci tablo, on saniyelik aralıklarla verilerin kaydedildiği “Anlık\_Olcumler” adlı tablodur. On saniyelik süre; Raspberry Pi, sürekli meşgul olmayacak ve iki kayıt arasındaki zaman en az olacak şekilde yapılan testlerle tespit edilmiştir. Bu tablodaki veriler iki günden fazla tutulmamaktadır. Yani iki gününü dolduran veriler silinmektedir. İkinci tablo ise on beş dakikalık aralıklarla ölçülen veriler ile meydana gelen alarmların kaydedildiği “Kayitli\_Olcumler” adlı tablodur. Sürenin on beş dakika olarak seçilmesinin nedeni, anlık ölçümlerde önemli zamanın 8-15 dakika olmasıdır (Akal, 2013). Bu tablodaki veriler ise hiçbir zaman silinmemektedir. Bu iki tabloya ait bilgilerin bulunduğu ekran görüntüleri Şekil 3.10’da ve Şekil 3.11’de verilmiştir.

```
MariaDB [Ortam_Izleme]> show columns from Anlik_Olcumler;
```

Field	Type	Null	Key	Default	Extra
ID	int(11)	NO	PRI	NULL	auto_increment
NEM	decimal(4,1)	YES		NULL	
SICAKLIK	decimal(4,1)	YES		NULL	
HAREKET	varchar(255)	YES		NULL	
ALEV	varchar(255)	YES		NULL	
CO	float	YES		NULL	
DUMAN	float	YES		NULL	
LPG	float	YES		NULL	
SIVI	varchar(255)	YES		NULL	
SIVI_SEVIYE	decimal(4,1)	YES		NULL	
PARLAKLIK	decimal(4,1)	YES		NULL	
ZAMAN	datetime	YES		NULL	
ZAMAN2	int(11)	YES		NULL	
DURUM	varchar(255)	YES		NULL	

Şekil 3.10. Anlik\_Olcumler Tablosu.

```
MariaDB [Ortam_Izleme]> show columns from Kayitli_Olcumler;
```

Field	Type	Null	Key	Default	Extra
ID	int(11)	NO	PRI	NULL	auto_increment
NEM	decimal(4,1)	YES		NULL	
SICAKLIK	decimal(4,1)	YES		NULL	
HAREKET	varchar(255)	YES		NULL	
ALEV	varchar(255)	YES		NULL	
CO	float	YES		NULL	
DUMAN	float	YES		NULL	
LPG	float	YES		NULL	
SIVI	varchar(255)	YES		NULL	
SIVI_SEVIYE	decimal(4,1)	YES		NULL	
PARLAKLIK	decimal(4,1)	YES		NULL	
ZAMAN	datetime	YES		NULL	
ZAMAN2	int(11)	YES		NULL	
DURUM	varchar(255)	YES		NULL	

Şekil 3.11. Kayitli\_Olcumler Tablosu.

Ortama ait verilerin kaydedildiği örneğe ait ekran görüntüsü EK-1’de gösterilmektedir.

Tablolardaki alanlar şu şekildedir;

**ID:** Ait olduđu satırdaki verilerin numarasıdır ve benzersizdir. Kayıtlar INTEGER tipinde olup, otomatik olarak artmaktadır. ID'lerin veritabanında kaydedilmesine ait örnek ekran görüntüsü Şekil 3.12'de gösterilmektedir.

ID
43918
43917
43916
43915
43914

Şekil 3.12. ID'lerin veritabanında kaydedilmesi.

**NEM:** Ölçülen bağıl nem değerlerinin kaydedildiği alandır. Bağıl nemin birimi % (yüzde) ile ifade edilmektedir; ancak veritabanına sadece sayısal değer olarak ve DECIMAL tipinde kaydedilmektedir. Ortamdaki nem bilgisinin veritabanında kaydedilmesine ait örnek ekran görüntüsü Şekil 3.13'te gösterilmektedir.

NEM
50.0
50.0
50.0
50.0
50.0

Şekil 3.13. Nem bilgisinin veritabanında kaydedilmesi.

**SICAKLIK:** Ortama ait sıcaklık değerinin kaydedildiği alandır. Sıcaklığın birimi "°C" (santigrat derece) ile belirtilmektedir; fakat sıcaklık, sadece sayısal değer olarak ve veritabanına DECIMAL tipinde kaydedilmektedir. Ortamın sıcaklık bilgisinin veritabanında kaydedilmesine ait örnek ekran görüntüsü Şekil 3.14'te gösterilmektedir.

SICAKLIK
24.0
24.0
24.0
24.0
24.0

**Şekil 3.14.** Sıcaklık bilgisinin veritabanında kaydedilmesi.

**HAREKET:** Ortamda hareket meydana geldiğinde, ortamda hareket olduğuna dair kaydın yapıldığı alandır. Veriler karakter olarak kaydedilmektedir. Veriler VARCHAR tipinde kaydedilmektedir. Hareket bilgisinin veritabanında kaydedilmesine ait örnek ekran görüntüsü Şekil 3.15’te gösterilmektedir.

HAREKET
HAREKET TESPIT EDİLDİ!
HAREKET TESPIT EDİLDİ!
NULL
HAREKET TESPIT EDİLDİ!
HAREKET TESPIT EDİLDİ!

**Şekil 3.15.** Hareket bilgisinin veritabanında kaydedilmesi.

**ALEV:** Ortamda alev tespit edilmesi durumunda, alev tespit edildiği bilgisinin kaydedildiği alandır. Veriler VARCHAR tipinde kaydedilmektedir. Alev bilgisinin veritabanında kaydedilmesine ait örnek ekran görüntüsü Şekil 3.16’da gösterilmektedir.

ALEV
NULL
ALEV TESPIT EDİLDİ!
ALEV TESPIT EDİLDİ!
ALEV TESPIT EDİLDİ!
ALEV TESPIT EDİLDİ!

**Şekil 3.16.** Alev bilgisinin veritabanında kaydedilmesi.

**CO:** Ortamdaki karbon monoksitin (CO) ölçülen ppm (parts per million) miktarının sayısal değerinin kaydedildiği alandır. Veriler FLOAT tipinde kaydedilmektedir. CO miktarının veritabanında kaydedilmesine ait örnek ekran görüntüsü Şekil 3.17’de gösterilmektedir.

CO
0.00770001
0.00766714
0.0077497
0.00769999
0.00768351

**Şekil 3.17.** CO’nun ppm miktarının veritabanında kaydedilmesi.

**DUMAN:** Dumanın ortamdaki ppm miktarına ait sayısal değerinin kaydedildiği alandır. Kayıtlar FLOAT şeklinde kaydedilmektedir. Ortamdaki duman miktarının veritabanında kaydedilmesine ait örnek ekran görüntüsü Şekil 3.18’de gösterilmektedir.

DUMAN
0.0205515
0.0204578
0.0206932
0.0205514
0.0205044

**Şekil 3.18.** Ortamdaki dumanın ppm miktarının veritabanında kaydedilmesi.

**LPG:** Ortamdaki LPG’nin ppm miktarına ait sayısal değerinin kaydedildiği alandır. Ölçümler FLOAT veri tipinde kaydedilmektedir. Ortamdaki LPG miktarının veritabanında kaydedilmesine ait örnek ekran görüntüsü Şekil 3.19’da gösterilmektedir.

LPG
0.00500004
0.00497056
0.0050447
0.00500002
0.00498523

**Şekil 3.19.** Ortamdaki LPG'nin ppm miktarının veritabanına kaydedilmesi.

**SIVI:** Ortamda su tespit edildiğinde, su tespitinin olduğu bilgisinin kaydedildiği alandır. Veriler VARCHAR tipinde kaydedilmektedir. Ortamda su tespitine ait bilginin veritabanında kaydedilmesine ait örnek ekran görüntüsü Şekil 3.20'de gösterilmektedir.

SIVI
SIVI TESPIT EDİLDİ!
SIVI TESPIT EDİLDİ!
SIVI TESPIT EDİLDİ!
SIVI TESPIT EDİLDİ!
NULL

**Şekil 3.20.** Su tespitine ait bilginin veritabanında kaydedilmesi.

**SIVI\_SEVIYE:** Su basması ihtimaline karşılık, ortamdaki su seviyesine ait bilginin kaydedildiği alandır. Ölçüm cm cinsinden yapılmaktadır; ancak ölçüme ait sadece sayısal değerler ve DECIMAL olarak veritabanına kaydedilmektedir. Ortamda bulunan suyun seviyesine ait yüksekliğin veritabanında kaydedilmesine ait örnek ekran görüntüsü Şekil 3.21'de gösterilmektedir.

SIVI_SEVIYE
0.1
0.7
0.6
0.3
0.3

**Şekil 3.21.** Ortamdaki suyun seviyesinin veritabanında kaydedilmesi.

**ZAMAN:** Kaydedilen her satır için “Gün/Ay/Yıl Saat/Dakika/Saniye” bilgisinin kaydedildiği alandır. Bu bilgi DATETIME olarak veritabanına kaydedilmektedir. Zaman bilgisinin veritabanında kaydedilmesine ait örnek ekran görüntüsü Şekil 3.22’de gösterilmektedir.

ZAMAN
2019-07-16 11:27:42
2019-07-16 11:27:40
2019-07-16 11:27:37
2019-07-16 11:27:35
2019-07-16 11:27:32

**Şekil 3.22.** Zaman bilgisinin veritabanında kaydedilmesi.

**ZAMAN2:** Her satır için kaydedilen zaman bilgisinin “Unix Epoch Zamanı” olarak kaydedildiği alandır. Unix Epoch Zamanı, 1 Ocak 1970’ten ölçülen zaman arasındaki farkın saniye olarak ifade edilmesidir. Bu zaman veritabanında INTEGER olarak kaydedilmektedir. Bu türdeki zamanlar, hazır JavaScript grafiklerinde kullanım açısından kolaylık sağladığı için kullanılmıştır. Zaman bilgisinin, Unix Epoch Zamanı şeklinde veritabanında kaydedilmesine ait örnek ekran görüntüsü Şekil 3.23’de gösterilmektedir.

ZAMAN2
1563265662
1563265660
1563265657
1563265655
1563265652

**Şekil 3.23.** Zaman bilgisinin, Unix Epoch Zamanı şeklinde kaydedilmesi.

**DURUM:** Meydana gelen alarmların ve ortamda bir sorun bulunmadığı bilgisinin kaydedildiği alandır. Bilgiler VARCHAR olarak kaydedilmektedir. Durum bilgisinin veritabanında kaydedilmesine ait örnek ekran görüntüsü Şekil 3.24'te gösterilmektedir.

DURUM
SIVI TESPİT EDİLDİ!
SIVI TESPİT EDİLDİ!
SIVI TESPİT EDİLDİ!
SIVI TESPİT EDİLDİ!
HAREKET TESPİT EDİLDİ!

**Şekil 3.24.** Durum bilgisinin veritabanında kaydedilmesi.

### 3.3. Verilerin İnternet Sayfasında Yayınlanması

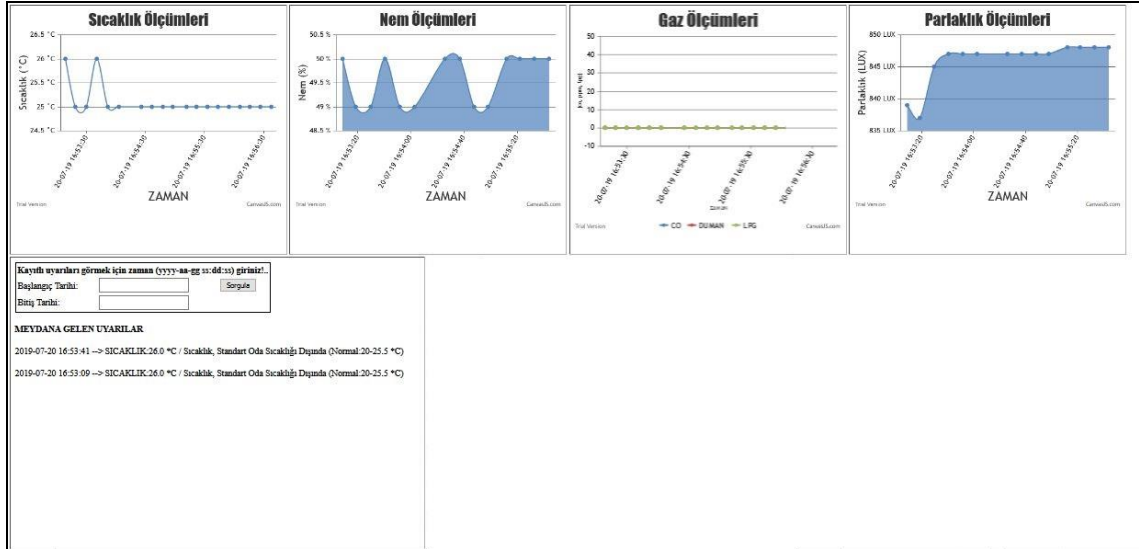
Bu bölümde veritabanındaki verilerin internet sayfalarında yayınlanması anlatılmaktadır. Sıcaklık, nem, LPG, CO, duman ve parlaklık bilgileri grafik şeklinde gösterilmektedir. Sistemin internet sayfalarına erişmek için kullanıcı adı ve şifre ile oturum açılmalıdır. Sisteme girişin yapıldığı giriş sayfasının ekran görüntüsü Şekil 3.25'te gösterilmektedir. Başarılı bir şekilde oturum açıldıktan sonra sisteme ait anasayfa görüntülenmektedir.



Şekil 3.25. Sistemin giriş sayfası.

### 3.3.1. Anlık Verilerin Yayınlanması

Verilere ait grafikler ana sayfada küçük pencerelerde toplu olarak gösterilmektedir. Bu pencerelere tıklandığında grafikler daha detaylı bir şekilde görüntülenebilmektedir. Ayrıca ana sayfada ortamın genel durumunun ve uyarıların bulunduğu iki pencere de bulunmaktadır. Ortamda meydana gelen alarmlar genel durumlar penceresinde ve ortamın fiziksel değerleri normal değerler dışına çıktığında ise duruma ait bildirimler uyarılar penceresinde kullanıcıya gösterilmektedir. “Anlik\_Olcumler” tablosundaki sıcaklık, nem, gaz ve parlaklık bilgilerinin zamana bağlı olarak, toplu bir şekilde sistemin ana sayfasında gösterilmesi Şekil 3.26’da verilmiştir.



Şekil 3.26. Anlık bilgilerin sistemin ana sayfasında gösterilmesi.

**Sıcaklık ve Nem Bilgisinin Yayınlanması:** Ortama ait sıcaklık ve nem bilgileri “olcumler.php” yardımıyla veritabanının “Anlik\_Olcumler” tablosundaki “SICAKLIK” ve “NEM” alanlarından sorgulanıp JSON formatında, “olcumler.json” adında bir dosyaya kaydedilir. JSON dosyasında hangi verilerin bulunacağı PHP kodunda belirtilmektedir. Sıcaklık bilgisinin grafiği “sicaklik.php” ve nem bilgisinin grafiği de “nem.php” dosyalarındaki HTML ve JavaScript kodları ile JSON dosyasından alınan veriler yardımıyla oluşturulmaktadır. “sicaklik.php” ve “nem.php” her yenilendiğinde JSON dosyası tekrar oluşturulmaktadır. Bu sayede grafikteki veriler de güncellenmektedir.

Anlık verilerin toplu bir şekilde gösterildiği sistemin ana sayfasına ait HTML kodu aşağıda gösterilmektedir.

```
<?php
require_once('dogrulama.php');
?>
<html>
<div id="iframeHolder">
<iframe src="sicaklik.php" width="500" height="450" scrolling="no"></iframe>
<iframe src="nem.php" width="500" height="450" scrolling="no"></iframe>
<iframe src="gaz1.php" width="500" height="450" scrolling="no"></iframe>
<iframe src="parlaklik.php" width="500" height="450" scrolling="no"></iframe>
<iframe src="uyarilar.php" width="750" height="680" scrolling="no"></iframe>
<iframe src="durum.php" width="650" height="680" scrolling="no"></iframe>
</div>
<p><a href=" ../kayitli/anasayfa.php" target="_blank">Kayitli Veriler icin Tiklayin</a> </p>
</html>
```

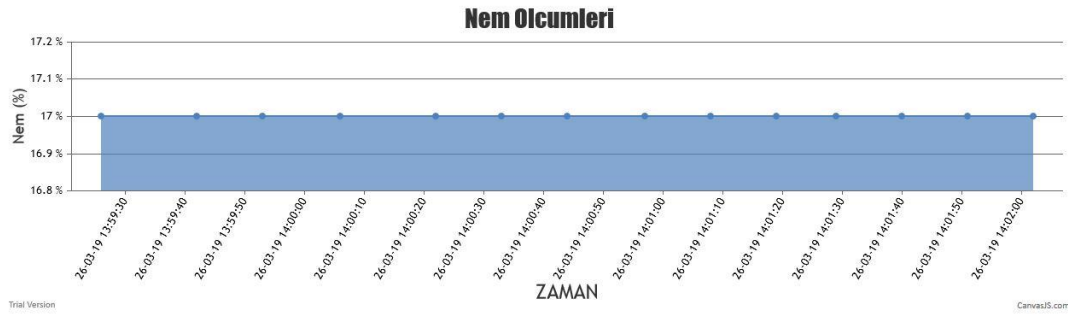
Grafiklerin oluşumuna ait akış diyagramı Şekil 3.27’de verilmiştir. Anlık sıcaklık grafiği Şekil 3.28’de ve anlık nem grafiği de Şekil 3.29’da gösterilmektedir. “olcumler.php” dosyasındaki kodlar ve veritabanı sorgularıyla JSON dosyasına kaydedilecek veriler sorgulanıp veritabanı tablosundaki sütun adlarına göre önce satır satır dizilere atanır, daha sonra bu dizilerle “olcumler.json” dosyası oluşturulur.



Şekil 3.27. Anlık sıcaklık ve nem grafiklerinin oluşumu.



Şekil 3.28. Anlık sıcaklık bilgisi grafiği.



**Şekil 3.29.** Anlık nem bilgisi grafiği.

Bu JSON dosyasındaki veriler sadece sıcaklık ve nem bilgisi bulunmamaktadır. Ayrıca bu JSON dosyasında parlaklık bilgisi ve suyun yüksekliği ait bilgi de bulunmaktadır. JSON dosyası oluşturulduktan sonra internet sayfası on saniye arayla otomatik olarak yenilenmektedir. Söz konusu PHP kodları aşağıda verilmiştir.

```
$sql = "SELECT ZAMAN,ZAMAN2,SICAKLIK,NEM,PARLAKLIK,SIVI_SEVIYE FROM Anlik_Olcumler
ORDER BY ID DESC LIMIT 20";
$result = $conn->query($sql);

if ($result->num_rows > 0) {
    $rows = array();
    while($row = $result->fetch_assoc()) {
        $rows[] = $row;
    }
    $fp = fopen('olcumler.json', 'w');
    fwrite($fp, json_encode($rows));
    fclose($fp);
} else {
    echo "0 results";
}
$conn->close();
include 'sicaklik.html'; //sicaklik.html sayfası cagriliyor
$url=$_SERVER['olcumler.php'];
header("Refresh: 10; URL=$url"); //sayfa 10 saniyede bir yenileniyor*/
```

“sicaklik.php” ve “nem.php” dosyasındaki grafikler çizilirken “olcumler.json” dosyasından veriler alınıp bir fonksiyon yardımıyla grafikteki zaman, sıcaklık ve nem eksenlerine atanmaktadır. Sıcaklık grafiğine ait JavaScript kodları aşağıda verilmektedir.

```
function addData(data) {
    for (var i = 1; i < data.length; i++) {
        dataPoints.push({
            x: new Date(data[i].ZAMAN2*1000),
            y: Number(data[i].SICAKLIK)
        });
    }
}
```

```

    }
    chart.render();
    console.log( JSON.stringify(dataPoints) );
}
$.getJSON("olcumler.json", addData);
function toggleDataSeries(e){
    if (typeof(e.dataSeries.visible) === "undefined" || e.dataSeries.visible) {
        e.dataSeries.visible = false;
    }
    else{
        e.dataSeries.visible = true;
    }
    chart.render();
}
}

```

Nem grafiğine ait JavaScript kodları aşağıda verilmektedir.

```

function addData(data) {
for (var i = 6; i < data.length; i++) {
    dataPoints.push({
        x: new Date(data[i].ZAMAN2*1000),
        y: Number(data[i].NEM)
    });
}
chart.render();
console.log(JSON.stringify(data.Points)); }
$.getJSON("olcumler.json", addData);
function toggleDataSeries(e){
    if (typeof(e.dataSeries.visible) === "undefined" || e.dataSeries.visible) {
        e.dataSeries.visible = false;
    }
    else{
        e.dataSeries.visible = true;
    }
    chart.render();
}
}

```

**Parlaklık Bilgisinin Yayınlanması:** Ortama ait parlaklık bilgisi “olcumler.php” yardımıyla “olcumler.json” dosyasına kaydedilmektedir. “parlaklik.php” dosyasındaki grafik çizilirken JSON dosyasından zaman ve parlaklık eksenine ait veriler, bir fonksiyon ile grafiğe aktarılıp grafiğin çizimi yapılır. Grafiğin internet tarayıcılarındaki görünümüne ait ekran görüntüsü Şekil 3.30’da gösterilmiştir. Bu fonksiyona ait JavaScript kodu da aşağıda verilmektedir.

```

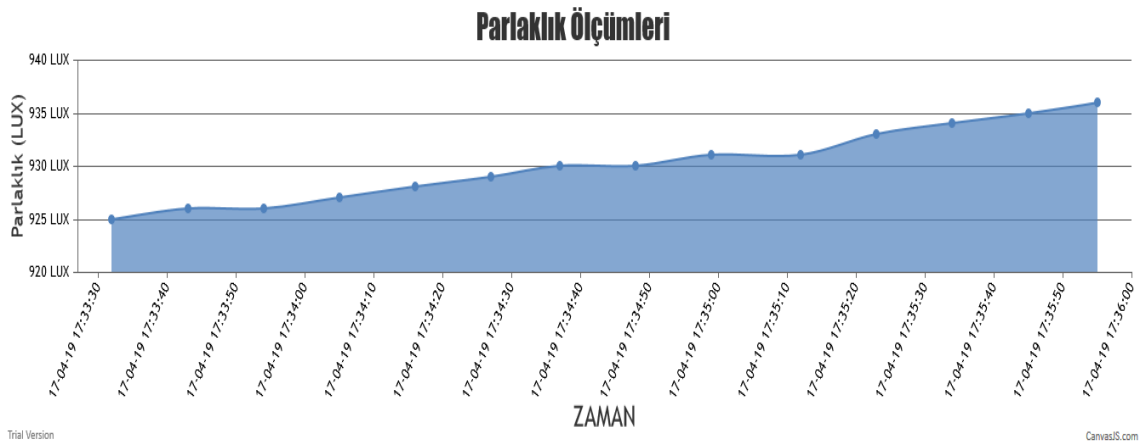
function addData(data) {
for (var i = 6; i < data.length; i++) {
    dataPoints.push({
        x: new Date(data[i].ZAMAN2*1000),
        y: Number(data[i].PARLAKLIK)
    });
}
}

```

```

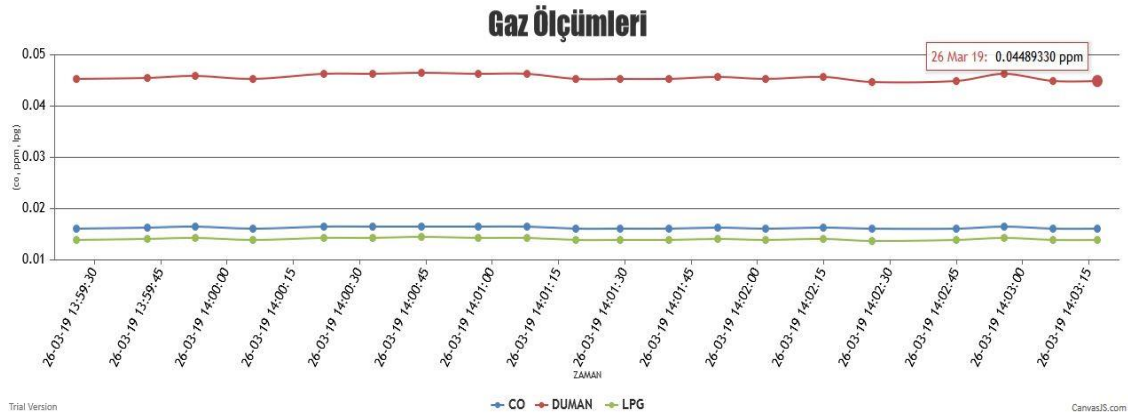
    chart.render();
    console.log( JSON.stringify(dataPoints) );
}
$.getJSON("olcumler.json", addData);
function toggleDataSeries(e){
    if (typeof(e.dataSeries.visible) === "undefined" || e.dataSeries.visible) {
        e.dataSeries.visible = false;
    }
    else{
        e.dataSeries.visible = true;
    }
}
chart.render();}

```



**Şekil 3.30.** Anlık parlaklık bilgisine ait grafik.

**Gaz Bilgisinin Yayınlanması:** Ortamdaki LPG, duman ve CO miktarını “gaz1.php” internet sayfasında yayınlamak için “gaz.php” dosyasındaki PHP kodları yardımıyla “Anlik\_Olcumler” tablosunun “LPG”, “DUMAN” ve “CO” alanlarından veriler çekilip “gaz.json” adlı bir dosyaya kaydedilir. İnternet sayfası yüklenirken JSON dosyasındaki veriler Şekil 3.31’deki gibi grafikte gösterilir.



**Şekil 3.31.** Anlık gaz bilgisi grafiği.

“gaz.php” dosyasında JSON dosyasına kaydedilecek verilere ait veritabanı sorgusu, json dosyasının oluşturulması ve internet sayfasının otomatik yenilemesi ait PHP kodu aşağıda gösterilmektedir.

```
$sql = "SELECT ZAMAN,ZAMAN2,CO,LPG,DUMAN FROM Anlik_Olcumler ORDER BY ID DESC LIMIT 20";
$result = $conn->query($sql);
if ($result->num_rows > 0) {
    // echo "<table><tr><th>zaman2</th><th>SICAKLIK</th><th>NEM</th></tr>";
    // output data of each row
    $rows = array();

    while($row = $result->fetch_assoc()) {
        // echo "<tr><td>" . $row["zaman2"]. "</td><td>" . $row["sicaklik"]. "</td><td>" .
        $row["nem"]. "</td></tr>";
        // $rows['date'] = $row["zaman2"];
        $rows[] = $row;
    }

    $fp = fopen('gaz.json', 'w');
    fwrite($fp, json_encode($rows));
    fclose($fp);
}
```

“gaz1.php” dosyasındaki JavaScript kodlarıyla “gaz.json” dosyasındaki LPG, duman ve CO bilgilerinin okunup grafiğe aktarılmasına ait JavaScript kodu aşağıda verilmiştir.

```
function addData(data) {
    for (var i = 0; i < data.length; i++) {
        dataPoints.push({x: new Date(data[i].ZAMAN2*1000), y: Number(data[i].CO)});
        dataPoints1.push({x: new Date(data[i].ZAMAN2*1000), y: Number(data[i].DUMAN)});
        dataPoints2.push({x: new Date(data[i].ZAMAN2*1000), y: Number(data[i].LPG)});
    }

    chart.render(); }
}
```

```

$.getJSON("gaz.json", addData);
function toggleDataSeries(e){
    if (typeof(e.dataSeries.visible) === "undefined" || e.dataSeries.visible) {
        e.dataSeries.visible = false;
    }
    else{
        e.dataSeries.visible=true;
    }
}
chart.render(); }

```

**Uyarıların Yayınlanması:** Ortamın fiziksel değerlerinin belirlenen limitlerin dışına çıkması durumunda uyarıların listelendiği sayfadır. Uyarılar, Anlık\_Olcumler tablosunda bulunan verilerin PHP kodlarıyla denetlenmesiyle elde edilmektedir. Bu veriler veritabanına kaydedilmemektedir. Sayfa yenilendiğinde uyarılar da güncellenmektedir. Sayfada tarih aralığı girilerek geçmişte uyarı oluşturmuş durumlar da görülebilmektedir. Uyarılar sayfasına ait ekran görüntü Şekil 3.32’de verilmektedir.

Kayıtlı uyarıları görmek için zaman (yyyy-aa-gg ss:dd:ss) giriniz!..

Başlangıç Tarihi:

Bitiş Tarihi:

**MEYDANA GELEN UYARILAR**

2019-07-20 16:53:41 --> SICAKLIK:26.0 \*C / Sıcaklık, Standart Oda Sıcaklığı Dışında (Normal:20-25.5 \*C)

2019-07-20 16:53:09 --> SICAKLIK:26.0 \*C / Sıcaklık, Standart Oda Sıcaklığı Dışında (Normal:20-25.5 \*C)

**Şekil 3.32.** Uyarılar sayfasına ait ekran görüntü

İnsanların çoğunun kendilerini rahat ettiği ortamın sıcaklığının 20-25,5 °C ve bağıl neminin %30-60 şeklinde olduğu tespit edilmiştir (İlten vd., 2017). Bu yüzden bu çalışmada sıcaklık ve nem için limit değerler bu şekilde seçilmiştir.

Sekiz saat çalışılan bir ortamda CO, en fazla 50 ppm olmalıdır (Durşen ve Yasun, 2012). Bu yüzden sisteme CO'nun uyarı limiti 50 ppm olacak şekilde tanımlamalar yapılmıştır.

Normal şartlar altında bir ortamda duman ve LPG bulunmamaktadır. Bu yüzden duman ve LPG için uyarı limiti 1 ppm olarak belirlenmiştir.

### 3.3.2. Kayıtlı Verilerin Yayınlanması

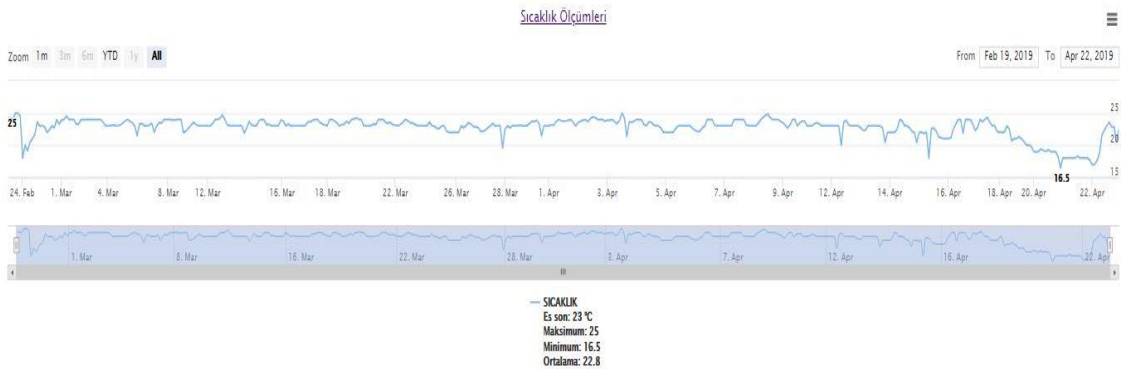
Kayıtlı veriler on beş dakikalık aralıklarla kaydedilen ortam bilgileri ve ortamda meydana gelen alarm durumlarından oluşmaktadır. Kayıtlı verilere ait grafikler toplu bir

şekilde “.../kayitli/anasayfa.php” sayfasında Şekil 3.33’deki gibi gösterilmektedir. Ayrıntılı incelenmek istenen grafiğe tıklandığında grafik yeni bir sekmede açılmaktadır.



Şekil 3.33. Kayıtlı verilerin toplu bir şekilde bulunduğu ana sayfa.

Anlık sıcaklık, nem, parlaklık ve gaz bilgisinin görüntülediği grafiklerden farklı olarak; bu grafiklerde bilgilerin görüntüleneceği tarih aralığı seçilebilmekte ve seçilen tarih aralığındaki maksimum, minimum ve ortalama değerler de görüntülenebilmektedir. Örnek olarak kayıtlı sıcaklık bilgilerine ait grafik Şekil 3.32’de gösterilmektedir.



Şekil 3.34. Sıcaklık bilgisine ait kayıtlı veriler grafiği.

**Durum Bilgisinin Yayınlanması:** Bu alanda “Kayıtlı\_Veriler” tablosundaki “DURUM” sütununda bulunan son on satır liste halinde gösterilmektedir. Bu satırlar meydana gelen alarmlar veya on beş dakika arayla kaydedilmiş “ORTAM NORMAL!” bilgisinden oluşmaktadır. Ayrıca geçmiş alarmlar veya kayıtların hepsi tarih aralığı girilerek görüntülenebilmektedir. Durum bilgisi sistemin ana sayfasında Şekil 3.35’teki gibi gösterilmektedir.

Kayıtları görmek için tarih (yyyy-aa-gg) giriniz!..	
Başlangıç Tarihi:	Alarmları Sorgula
Bitiş Tarihi:	Tümünü Sorgula

**ORTAMIN GENEL DURUMU**

2019-07-20 16:49:45 --> ORTAM NORMAL!

2019-07-20 16:33:13 --> ORTAM NORMAL!

2019-07-20 16:18:36 --> ORTAM NORMAL!

2019-07-16 11:27:42 --> SIVI TESPİT EDİLDİ! / SIVI SEVİYESİ: 0.1 CM / NEM:50.0 % / SICAKLIK:24.0 \*C / CO:0.00770001 ppm / LPG:0.00500004 ppm / DUMAN:0.0205515 ppm

2019-07-16 11:27:40 --> SIVI TESPİT EDİLDİ! / SIVI SEVİYESİ: 0.7 CM / NEM:50.0 % / SICAKLIK:24.0 \*C / CO:0.00766714 ppm / LPG:0.00497056 ppm / DUMAN:0.0204578 ppm

2019-07-16 11:27:37 --> SIVI TESPİT EDİLDİ! / SIVI SEVİYESİ: 0.6 CM / NEM:50.0 % / SICAKLIK:24.0 \*C / CO:0.0077497 ppm / LPG:0.0050447 ppm / DUMAN:0.0206932 ppm

2019-07-16 11:27:35 --> SIVI TESPİT EDİLDİ! / SIVI SEVİYESİ: 0.3 CM / NEM:50.0 % / SICAKLIK:24.0 \*C / CO:0.00769999 ppm / LPG:0.00500002 ppm / DUMAN:0.0205514 ppm

2019-07-16 11:27:32 --> HAREKET TESPİT EDİLDİ! / SIVI SEVİYESİ: 0.3 CM / NEM:50.0 % / SICAKLIK:24.0 \*C / CO:0.00768351 ppm / LPG:0.00498523 ppm / DUMAN:0.0205044 ppm

2019-07-16 11:27:29 --> HAREKET TESPİT EDİLDİ! / SIVI SEVİYESİ: 0.3 CM / NEM:50.0 % / SICAKLIK:24.0 \*C / CO:0.00774973 ppm / LPG:0.00504472 ppm / DUMAN:0.0206932 ppm

2019-07-16 11:27:26 --> HAREKET TESPİT EDİLDİ! / SIVI SEVİYESİ: 0.1 CM / NEM:50.0 % / SICAKLIK:24.0 \*C / CO:0.00769996 ppm / LPG:0.00499999 ppm / DUMAN:0.0205513 ppm

**Şekil 3.35.** Ortamın genel durumuna ait son durumlar listesi.

### 3.3. Alarmlar

Ortamda anormal bir durum meydana gelmesi durumunda daha önceden belirlenen kişilere e-posta ile bilgi verilmektedir. Alarm durumları şunlardır;

- i. Ortamda alev tespit edilmesi,
- ii. Ortamda su tespit edilmesi,
- iii. Ortamdaki gaz değerlerinin normal değerlerin üstüne çıkması,

- iv. Ortamın sıcaklığının 48°C'nin üstüne çıkması,
- v. Ortamda hareket tespit edilmesi şeklindedir.

Sistem tarafından e-posta gönderilebilmesi için "smtplib" kütüphanesinin "ortam\_olcum.py" programına eklenmesi gerekmektedir. Kütüphaneyi Raspberry Pi'ye indirmek için aşağıdaki komut kullanılmalıdır.

```
pip install smtplib
```

Alarm durumları meydana geldiğinde e-postanın gönderilebilmesi için bazı tanımlamaların yapılması gerekmektedir. Bunlar; gönderici e-postanın adı, şifresi, posta sunucunun adresi, port numarası ve alıcı e-posta şeklindedir. Bu bilgiler ve mesaj şablonu bir fonksiyon altında oluşturulup alarm durumlarında fonksiyonun çağrılıp mesajın gönderilebilmesi sağlanmaktadır. Oluşturulan fonksiyona ait kodlar aşağıdaki gibidir.

```
def mailgonder(message):
    fromaddr = 'raspmbalci@gmail.com'
    toaddrs = 'balci_mehmet@windowslive.com'

    username = 'raspmbalci@gmail.com'
    password = '....'

    server = smtplib.SMTP('smtp.gmail.com', 587)
    server.ehlo()
    server.starttls()
    server.login(username, password)
    server.sendmail(fromaddr, toaddrs, message)
    server.quit()
```

Alarm meydana geldiğinde fonksiyonun çağrılmasına ait kodlar aşağıda verilmiştir.

```
elif float(format(temperature))>(48.0): #Sicaklik Kontrolu
mesaj=("ORTAMDAKI SICAKLIK NORMALIN USTUNDE!...\nSivi Yuksekligi={0:0.1f} cm \
\nSicaklik={1:0.1f}*C Nem={2:0.1f}% \nLPG={3:0.5f} ppm CO={4:0.5f} ppm DUMAN={5:0.5f} \
ppm".format(seviye,temperature,humidity,LPG,KARBMNKST,DUMAN))
mailgonder(message = 'Subject: {konu}\n\n{mesaj}'.format(konu=konu,mesaj=mesaj))
x.execute("INSERT INTO Kayitli_Olcumler (ZAMAN2, ZAMAN, SICAKLIK, NEM, SIVI_SEVIYE, \
PARLAKLIK, CO, LPG, DUMAN, DURUM) VALUES (UNIX_TIMESTAMP(NOW()), NOW(),\
%s, %s, %s, %s, %s, %s, 'ORTAMDAKI SICAKLIK NORMALIN USTUNDE!')", \
(format(temperature), format(humidity), seviye, lux, perc["GAS_LPG"], perc["CO"], \
perc["SMOKE"]))
conn.commit()
```

Yukarıdaki kodda “konu” deęişkeni, gönderilecek e-postanın konusunu oluşturmakta ve “if” dışında tanımlanmış olup “konu='ORTAM IZLEME ALARM!’” biçimindedir. “mesaj” deęişkeni ise e-postanın içerięi olup alarmin türünü, ortamın sıcaklık ve nem bilgilerini içermektedir.

Bu çalışmada mail göndermek için gmail hesabı kullanıldı. Gmail hesabından bu tür sistemlerin e-posta gönderilebilmesi için güvenlik ayarının deęiştirilmesi gerekmektedir. Bu ayar, “Düşük Güvenlikli Uygulamalardan Posta Göndermeye İzin Ver” olup, aktif olması gerekmektedir. Sistem tarafından gönderilen bir örnek posta Şekil 3.36’da verilmiştir.



**Şekil 3.36.** Sistem tarafından gönderilen örnek alarm e-postası.

### 3. SONUÇ

Bu çalışmada Raspberry Pi mini bilgisayarına Linux tabanlı işletim sistemi kuruldu. Python yazılım dili aracılığıyla bir ortama ait sıcaklık, nem, parlaklık, LPG, duman ve CO gibi değerler ölçüldü. Bu değerler bir internet sayfasında yayınlanıp, internet erişiminin olduğu her yerde canlı olarak, sistemin kurulu olduğu ortam izlenebilmektedir; ortamın fiziksel durumunda uyarı oluşturacak bir değişim meydana geldiğinde ise internet sayfasından bildirimlerle kullanıcı bilgilendirilmektedir. Ayrıca ortamda meydana gelebilecek bazı tehlikeli durumlar için alarmlar oluşturulabilmektedir. Bunlar sıcaklık, LPG, duman ve CO miktarının belirlenen eşik değerinin üzerine çıkması; ortamda hareket olmaması gerektiği zamanlarda hareket tespit edilmesi, ortamda alev tespit edilmesi ve su sensörünün bulunduğu konumda su tespit edilmesi şeklindedir. Alarmlar meydana geldiğinde belirlenen kişilere e-posta ile bilgilendirme yapılmaktadır.

Ortama ait ölçülen değerler bir veritabanında iki farklı tabloya kaydedilmektedir. Bu tablolardan birindeki veriler geçici olup, diğerindeki veriler ise kalıcıdır. Anlık veriler ortamın durumunu gerçek zamanlı takip etmek; kalıcı veriler ise istatistiksel veri sağlamak amacıyla kaydedilmektedir ve kullanıcının geri dönük bilgi almasına yardımcı olmaktadır. Verilere erişebilmek için kullanıcı adı ve şifreyle oturum açılması gerekmektedir.

IoT'nin yaygınlaşmasıyla birçok profesyonel ortam izleme sistemi geliştirilmektedir. Bu tür sistemlerin hem maliyetleri yüksek olmakta, hem de sistemde değişiklik yapılmasına izin verilmemektedir. Bu çalışma kapsamında geliştirilen sistemde kullanılan yazılımlar, sensörlerden veri alınmasında kullanılan kütüphaneler gibi argümanların hepsi ücretsiz platformlardan sağlanmıştır. Kodların hemen hemen tümü açık kaynak kodlu olup istenilen değişiklikler yapılabilmektedir. Bu sayede düşük maliyetli bir ortam izleme sistemi geliştirilebilmektedir. Sisteme ait python, PHP, HTML ve JavaScript kodlarının başka projelerde kullanılabilmesi için tüm kodlar, "[https://github.com/mbalci02/ortam\\_izleme](https://github.com/mbalci02/ortam_izleme)" sitesinde paylaşılmıştır.

Bu çalışmadaki sistem aşağıdaki gibi geliştirilebilir;

➤ Anlık verilerin iki gün sonunda silinmesi yerine, bir uygulama yazılarak veriler sıkıştırılıp depolanabilir.

- Sisteme SMS modülü eklenerek alarm durumlarında SMS gönderilmesi sağlanabilir.
- Mobil bir yazılım geliştirilerek alarmlar ve ortam bilgileri bu yazılımdan izlenebilir.
- Sisteme kameralar eklenerek alarm durumlarında ortama ait görüntüler alınabilir.
- Elektrik kesintilerinde sisteme enerji sağlamak amacıyla kesintisiz güç kaynağı sisteme entegre edilebilir. Şebekede elektrik kesintisi meydana geldiğinde kullanıcıya bilgi verecek harici bir sistem de geliştirilip, kullanıcı bilgilendirilebilir.

## KAYNAKLAR

- Agrawal, N., & Singhal, S. (2015). Smart Drip Irrigation System using Raspberry pi and Arduino. *International Conference on Computing, Communication and Automation (ICCCA)*, 15-16 Mayıs 2015, Noida, 928-932.
- Akal, D. (2013). İç Ortam Hava Kirliliği ve Çalışanlara Olumsuz Etkileri. *ÇSGB Çalışma Dünyası Dergisi*, 1(1), 112-119.
- Akkuş, S. (2016). Nesnelerin İnterneti Teknolojisinde Güvenli Veri İletişimi - Programlanabilir Fiziksel Platformlar Arasında WEP Algoritması ile Kriptolu Veri Haberleşmesi Uygulaması. *Marmara Fen Bilimleri Dergisi*, 3, 100-111.
- Algün, G. (2018). Humidity Sensing Properties of Fluorine Doped Zinc Oxide Thin Films. *Journal of Materials Science: Materials in Electronics*, 29 (19), 17039-17046.
- Básaca-Preciado, L. C., Moreno-Partida, A. S., Terrazas-Gaynor, J. M., Ponce, M., López, J., Rodríguez-Quinonez, J. C., Fuentes, W. F., & Sergiyenko, O. (2017). Home and Building Automation through Social Networks. *Environment and Electrical Engineering and 2017 IEEE Industrial and Commercial Power Systems Europe (EEEIC / I&CPS Europe)*. 6-9 Haziran 2017, Milan, 3193-3199.
- Durşen, M., & Yasun, B. (2012). *Yeraltı Madenlerin Bulunan Zararlı Gazlar Metan Drenajı*. [http://www.isgum.gov.tr/rsm/file/isgdoc/IG15-yeraltinda\\_bulunan\\_zararli\\_gazlar\\_ve\\_metan\\_drenaji.pdf](http://www.isgum.gov.tr/rsm/file/isgdoc/IG15-yeraltinda_bulunan_zararli_gazlar_ve_metan_drenaji.pdf)
- URL 1: [http://jsfiddle.net/d\\_paul/supgh9c1/](http://jsfiddle.net/d_paul/supgh9c1/), (Erişim Tarihi: 30.12.2018).
- URL 2: [http://wiki.seeedstudio.com/Grove-Gas\\_Sensor-MQ2/](http://wiki.seeedstudio.com/Grove-Gas_Sensor-MQ2/), (Erişim Tarihi: 01.07.2019).
- URL 3: <https://canvasjs.com/javascript-charts/>, (Erişim Tarihi: 18.10.2018).
- URL 4: <https://circuitdigest.com/microcontroller-projects/light-intensity-measurement-using-ldr-and-atmega8/>, (Erişim Tarihi: 01.07.2019).
- URL 5: <https://lastminuteengineers.com/dht11-dht22-arduino-tutorial/>, (Erişim Tarihi: 01.07.2019).
- URL 6: <https://www.bigmessowires.com/2018/05/26/raspberry-pi-gpio-programming-in-c/>, (Erişim Tarihi: 13.03.2019)
- URL 7: <https://www.faranux.com/product/hc-sr04-ultrasonic-module-sensor/>, (Erişim Tarihi: 01.07.2019).
- URL 8: <https://www.highcharts.com/stock/demo/basic-line>, (Erişim Tarihi: 10.12.2018).
- URL 9: <https://www.phpr.org/php-dersleri/>, (Erişim Tarihi: 22.09.2018).
- URL 10: <https://www.raspberrypi.org/>, (Erişim Tarihi: 05.05.2018).
- URL 11: <https://www.w3schools.com/html/>, (Erişim Tarihi: 10.09.2018).
- URL 12: <https://www.w3schools.com/php/>, (Erişim Tarihi: 21.09.2018).

### KAYNAKLAR (Devam Ediyor)

- İlten, N., Tülay, A. T., & Caner, İ. (2017). İç Ortamlarda Sıcaklık ve Bağıl Nem Parametrelerinin Sosyo-Ekonomik Yapı ile İlişkisi. *BAUN Fen Bilimleri Enstitüsü Dergisi*, 19(2), 52-61.
- Jain, S., Vaibhav, A., & Goyal, L. (2014). Raspberry Pi Based Interactive Home Automation System through E-mail. 2014 International Conference on Reliability Optimization and Information Technology (ICROIT), 6-8 Şubat 2014, 277-280.
- Özgül, F. (2017). *Her Yönüyle Python*. İnkılâp Kitabevi, İstanbul, 511.
- Patchava, V., Kandala, H. B., & Babu, P. R. (2015). A Smart Home Automation Technique with Raspberry Pi using IoT. *International Conference on Smart Sensors and Systems (IC-SSS)*, 21-23 Aralık 2015, Bengaluru, 1-4.
- Pekgöz, N. (2005). *Webmaster için Javascript*. Pusula Yayıncılık, İstanbul, 251.
- Prasad, S., Mahalakshmi, P., Sunder, A. J. C., & Swathi, R. (2014). Smart Surveillance Monitoring System Using Raspberry PI and PIR Sensor. *International Journal of Computer Science and Information Technologies (IJCSIT)*, 5(6), 7107-7109.
- Sandeep, V., Gopal, K. L., Naveen, S., Amudhan, A., & Kumar, L. S. (2015). Globally Accessible Machine Automation Using Raspberry Pi Based on Internet of Things. 2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 10-13 Ağustos 2015, Kochi, 1144-1147.
- Şansal, U., & Temurtaş, F. (2017). Raspberry Pi Kullanarak Evlerde Güvenlik Sistemi Oluşturma. 4. *Ulusal Meslek Yüksekokulları Sosyal Ve Teknik Bilimler Kongresi*. 11-13 Mayıs 2017, Burdur, 836-842
- Uçar, A., & Uludağ, M. H. (2018). Nesnelerin İnterneti (IoT) ile Akıllı Sınıf ve Öğrenci Takip Sistemi Tasarımı. *DÜMF Mühendislik Dergisi*, 9(2), 591-600.
- Yüzgeç, U., & Aba, Ö. (2017). Raspberry Pi Kullanılarak bir Akıllı Ev Uygulaması Geliştirilmesi. *Bilecik Şeyh Edebali Üniversitesi Fen Bilimleri Dergisi*, 4 (1), 21-29.

## EKLER

**EK-1:** Veritabanında tutulan örnek kayıtlar.

ID	NEM	SICAKLIK	HAREKET	ALEV	CO	DUMAN	LPG	SIVI	SIVI_SEVIYE	PARLAKLIK	ZAMAN	ZAMAN2	DURUM
43918	50.0	24.0	NULL	NULL	0.00770001	0.0205515	0.00500004	SIVI TESPIT EDILDI!	0.1	796.0	2019-07-16 11:27:42	1563265662	SIVI TESPIT EDILDI!
43917	50.0	24.0	NULL	NULL	0.00766714	0.0204578	0.00497056	SIVI TESPIT EDILDI!	0.7	796.0	2019-07-16 11:27:40	1563265660	SIVI TESPIT EDILDI!
43916	50.0	24.0	NULL	NULL	0.0077497	0.0206932	0.0050447	SIVI TESPIT EDILDI!	0.6	796.0	2019-07-16 11:27:37	1563265657	SIVI TESPIT EDILDI!
43915	50.0	24.0	NULL	NULL	0.00769999	0.0205514	0.00500002	SIVI TESPIT EDILDI!	0.3	796.0	2019-07-16 11:27:35	1563265655	SIVI TESPIT EDILDI!
43914	50.0	24.0	HAREKET TESPIT EDILDI!	NULL	0.00768351	0.0205044	0.00498523	NULL	0.3	796.0	2019-07-16 11:27:32	1563265652	HAREKET TESPIT EDILDI!
43913	50.0	24.0	HAREKET TESPIT EDILDI!	NULL	0.00774973	0.0206932	0.00504472	NULL	0.3	796.0	2019-07-16 11:27:29	1563265649	HAREKET TESPIT EDILDI!
43912	50.0	24.0	HAREKET TESPIT EDILDI!	NULL	0.00769996	0.0205513	0.00499999	NULL	0.1	796.0	2019-07-16 11:27:26	1563265646	HAREKET TESPIT EDILDI!
43911	50.0	24.0	HAREKET TESPIT EDILDI!	NULL	0.00769999	0.0205514	0.00500002	NULL	0.3	796.0	2019-07-16 11:27:24	1563265644	HAREKET TESPIT EDILDI!
43910	51.0	23.0	NULL	NULL	0.00773307	0.0206457	0.00502974	NULL	0.6	800.0	2019-07-16 11:21:34	1563265294	ORTAM NORMAL!
43909	50.0	23.0	NULL	NULL	0.00761811	0.0203181	0.00492667	NULL	0.3	803.0	2019-07-16 11:05:43	1563264343	ORTAM NORMAL!
43908	50.0	24.0	NULL	NULL	0.00753734	0.020088	0.00485461	NULL	2.4	812.0	2019-07-16 10:51:09	1563263469	ORTAM NORMAL!
43907	51.0	23.0	NULL	NULL	0.00763442	0.0203645	0.00494126	NULL	0.2	815.0	2019-07-16 10:35:20	1563262520	ORTAM NORMAL!
43906	51.0	24.0	NULL	NULL	0.0076344	0.0203645	0.00494124	SIVI TESPIT EDILDI!	0.4	811.0	2019-07-16 10:23:23	1563261803	SIVI TESPIT EDILDI!
43905	51.0	24.0	NULL	NULL	0.00761811	0.0203181	0.00492667	SIVI TESPIT EDILDI!	0.3	811.0	2019-07-16 10:23:21	1563261801	SIVI TESPIT EDILDI!
43904	51.0	24.0	NULL	NULL	0.00761809	0.020318	0.00492665	SIVI TESPIT EDILDI!	0.3	812.0	2019-07-16 10:23:18	1563261798	SIVI TESPIT EDILDI!
43903	51.0	24.0	NULL	NULL	0.00756948	0.0201796	0.00488325	SIVI TESPIT EDILDI!	0.3	811.0	2019-07-16 10:23:16	1563261796	SIVI TESPIT EDILDI!
43902	51.0	23.0	HAREKET TESPIT EDILDI!	NULL	0.00760185	0.0202717	0.00491214	NULL	4.3	812.0	2019-07-16 10:21:01	1563261661	HAREKET TESPIT EDILDI!
43901	51.0	23.0	HAREKET TESPIT EDILDI!	NULL	0.00760187	0.0202718	0.00491216	NULL	4.3	812.0	2019-07-16 10:20:54	1563261654	HAREKET TESPIT EDILDI!
43900	45.0	24.0	NULL	NULL	0.0189696	0.0538404	0.0173883	NULL	0.3	820.0	2019-07-13 11:41:46	1563007306	ORTAM NORMAL!
43899	45.0	24.0	NULL	NULL	0.0189023	0.0536364	0.0173031	NULL	0.3	821.0	2019-07-13 11:26:45	1563006405	ORTAM NORMAL!

## ÖZ GEÇMİŞ



### Kişisel Bilgiler

Adı Soyadı : Mehmet BALCI  
Doğum Yeri ve Tarihi : Kahta – 01.04.1989

### Eğitim Durumu

Lisans Öğrenimi : Bilecik Şeyh Edebali Üniversitesi, Mühendislik Fakültesi  
Elektrik-Elektronik Mühendisliği (2012)  
Bildiği Yabancı Diller : İngilizce

### İş Deneyimi

Stajlar : 40 İş Günü Zorunlu Mühendislik Fakültesi Stajı  
Projeler : Masaüstü Bilgisayarların PXE ile Ağ Üzerinden Boot Edilip  
Açılması, Akıllı Kart Sistemi Sistem Yöneticiliği  
Çalıştığı Kurumlar : ANATEKS Anadolu Tekstil Fabrikaları SAN. TİC. AŞ,  
MOPAK Kağıt, Karton ve Sanayi AŞ, Bilecik Şeyh Edebali  
Üniversitesi

### İletişim

Adres : Hürriyet Mah. Atılğan Sok. Atılım Sitesi B Blok Kat:2 No:4  
Merkez/Bilecik  
E-Posta Adresi : mehmet.balci@bilecik.edu.tr

### Akademik Çalışmaları

- Balci, M., & Yüzgeç, U. (2018). Raspberry Pi mini bilgisayarı ile ortam izleme sistemi geliştirilmesi. *International Eurasian Conference on Science, Engineering and Technology (EurasianSciEnTech 2018)*. 22-23 Kasım, Ankara, 74.

**Tarih:11/07/2019**