

**T.C.  
SAKARYA ÜNİVERSİTESİ  
BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ**

**BSM 498 BİTİRME ÇALIŞMASI**

**KABLOSUZ ALGILAYICI AĞLAR İLE WEB TABANLI  
GÖZETLEME UYGULAMASI**

**B121210025 – İsmet GÜZELGÜN  
G111210070 – Ali COŞKUN**

**Fakülte Anabilim Dalı : BİLGİSAYAR MÜHENDİSLİĞİ  
Tez Danışmanı : Prof. Dr. Celal ÇEKEN**

**2015-2016 Bahar Dönemi**

**T.C.  
SAKARYA ÜNİVERSİTESİ  
BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ**

**KABLOSUZ ALGILAYICI AĞLAR İLE WEB TABANLI  
GÖZETLEME UYGULAMASI**

**BSM 498 - BİTİRME ÇALIŞMASI**

**İsmet GÜZELGÜN**

**Ali COŞKUN**

**Fakülte Anabilim Dalı : BİLGİSAYAR MÜHENDİSLİĞİ**

**Bu tez .. / .. / ... tarihinde aşağıdaki jüri tarafından oybirliği / oyçokluğu ile kabul edilmiştir.**

.....  
**Jüri Başkanı**

.....  
**Üye**

.....  
**Üye**

## ÖNSÖZ

İnsanoğlu tarih boyunca bir çok keşifte bulunmuştur. Tüm bunların birikimi ve bir sonucu olarak bugün geçmiş nesillerin bıraktıklarını miras alan bizler bir teknoloji çağına tanıklık ediyoruz. Önceleri yarı iletkenlerin keşfiyle evlerimize girebilecek kadar küçülen bilgisayarlar artık cebimizde taşıyabileceğimiz boyutlara indirilmiş durumda. Tüm bu süreçte bizler köklerimize ve keşfetme isteğimize hakim olmaksızın gelişmeye devam etmekteyiz. Bilgisayarların bu teknoloji çağının birer mihenk taşı olduğunu düşündüğümüzde geleceğin şimdiden daha ilginç buluşlara gebe olduğunu tahmin edebilmekteyiz.

Günümüzde bilinen en büyük ağ olarak Internet haliyle birlikte ağlar, artık hayatımızın bir parçası halini almıştır. Internet ise ağların ağı olarak yalnızca bilgisayar dünyasını değil milyonlarca insanın da yaşam şeklini değiştirmiştir. Bu haliyle bile henüz genç denebilecek bu sistem içerisinde IoT olarak bilinen yeni bir kavram doğmuştur. Bu başlık altında toplanan donanımsal ürünler ise son yıllarda oldukça revaçta olup bir çok akademik çalışmanın konusu halini almıştır. Dilimize kablosuz algılayıcı ve eyleyici ağlar olarak kazandırılmış kablosuz iletişim yeteneğine sahip sistemler yine bu çatı altında değerlendirilip çok farklı amaçlarla kullanılabilinmektedir. Genel olarak bir takım ölçüm yeteneklerine sahip özelleştirilmiş sensörlerin yaptıkları bu ölçümlere mana yüklenebilmesi adına kullanılan ağ oluşturma yeteneğine sahip bu donanımlardan yardım alınarak, şehirler için akıllı sistemler geliştirilmesi bu projenin amacı olmakla beraber çalışmamız boyunca sadece tek bir alana değil bir çok konuda çok farklı şekillerde bu donanımların kullanılabileceğine ilişkin başlıklara değinilecektir.

## İÇİNDEKİLER

ÖNSÖZ .....	iii
İÇİNDEKİLER .....	iv
SİMGELER VE KISALTMALAR LİSTESİ.....	vi
ŞEKİLLER LİSTESİ .....	vii
TABLolar LİSTESİ .....	viii
ÖZET.....	ix
BÖLÜM 1. GİRİŞ.....	10
BÖLÜM 2. NESNELERİN İNTERNETİ VE UYGULAMALARI.....	11
2.1. Projenin Konusu ve Nesnelerin İnterneti .....	11
2.1.1. Projenin Konusu .....	11
2.1.2. Nesnelerin İnterneti Kavramı ve Tarihçesi.....	11
2.1.3. Nesnelerin İnterneti Kavramına Dair İstatistikler .....	13
BÖLÜM 3. PROJENİN GERÇEKLENMESİ.....	17
3.1. Projenin Amacı .....	17
3.2. Kullanım Senaryosu .....	17
3.3. Neden Waspote Entegresi Tercih Edildi ? .....	18
3.4. Donanım İçin Gerekli Enerji Analizi.....	20
3.4.1. ON Modu.....	20
3.4.2. Uyku (Sleep) Modu .....	20
3.4.3. Derin Uyku (Deep Sleep) Modu .....	21
3.4.4. Kış Uykusu (Hibernate) Modu .....	21
3.5. Waspote Blok Diyagramı .....	23
3.6. Waspote Sensörleri .....	24
3.6.1. Toz Sensörü ( GP2Y1010AU0F).....	25
3.6.2. Çatlak Denetleme Sensörü .....	25
3.6.3. LD Sensörü.....	26
3.6.4. Gürültü Sensörü (Microphone POM-2735P-R) .....	26
3.6.5. Nem Sensörü .....	26
3.6.6. Işık Sensörü (LDR) .....	27
3.6.7. Sıcaklık Sensörü (MCP9700A) .....	27

3.7.	Wasmote Bağlantı Kurulum Modülleri .....	27
3.7.1.	XBee Kablosuz Ağ Protokolü .....	28
3.7.2.	Bluetooth Protokolü .....	28
3.8.	Wasmote ile Arduino Karşılaştırması .....	29
3.8.1.	Geliştirme Ortamı Bazında Karşılaştırma .....	29
3.8.2.	Kullanım Amaçları Bazında Karşılaştırma.....	30
3.8.3.	Giriş/Çıkış Üniteleri Bazında Karşılaştırma.....	30
3.8.4.	Bellek ve Mikrokontrolcü Bazında Karşılaştırma.....	31
3.8.5.	Enerji Tüketimi Bazında Karşılaştırma .....	31
BÖLÜM 4.	KULLANILAN YAZILIMLAR.....	32
4.1.	Node.js Çalıştırma Ortamı.....	32
4.1.1.	Node Paket Yöneticisi .....	32
4.1.2.	Socket.io .....	33
4.1.3.	Serialport .....	33
4.1.4.	Express Framework .....	33
4.1.5.	D3.JS Kütüphanesi .....	34
4.1.6.	Web Servis Geliştirme Süreci.....	34
BÖLÜM 5.	VERİTABANI YAPISI.....	36
5.1.	MongoDB Veritabanı Yapısı ve NoSQL .....	36
5.1.1.	NoSQL Veritabanı .....	36
5.1.2.	MongoDB Veritabanı.....	37
5.2.	MongoDB Veritabanı Yapısı.....	38
5.3.	Veritabanı Diyagramı .....	39
5.4.	Use Case Diyagramı .....	39
BÖLÜM 6.	SONUÇLAR VE ÖNERİLER .....	41
KAYNAKLAR.....		42
ÖZGEÇMİŞ .....		43
BSM 498 BİTİRME ÇALIŞMASI DEĞERLENDİRME VE SÖZLÜ SINAV TUTANAĞI.....		44

## SİMGELER VE KISALTMALAR LİSTESİ

IoT	: Internet Of Things
M2M	: Machine To Machine
POS	: Point Of Sale
WPAN	: Wireless Personal Area Network
ISM	: Industrial Scientific Medical Band
OTAP	: Over the Air Programming
NPM	: Node Packet Manager
NoSQL	: Not Only SQL
SQL	: Structured Query Language
CPP	: C Plus Plus Programlama Dili
İVTYS	: İlişkisel Veri Tabanı Yönetim Sistemleri
ACID	: Atomicity, Consistency, Isolation ve Durability
CSS	: Cascading Style Sheet
HTML	: Hyper Text Markup Language
DOM	: Document Object Model
SVG	: Scalable Vector Graphics

## ŞEKİLLER LİSTESİ

Şekil 2.1. İlk Kimlik Doğrulama Alıcısı. ....	13
Şekil 2.2. Intel Firmasının Sitesinden Alınan ve Kullanım Alanlarına Göre Kategorilendirilmiş İnfografik. ....	15
Şekil 2.3. Dünya Üzerinde İnternete Çıkabilen Cihazların Çeşitleri ve IoT.....	16
Şekil 2.4. Gelecek 10 Yıl İçin IoT Tahmini.....	16
Şekil 2.5. Nesnelerin İnterneti Kavramına Dahil Olan Cihazlar ve Kullanım Alanlarına İlişkin Yıl Bazında Değerlendirmeler İçeren Çizgi Grafik. ....	17
Şekil 3.1. Waspote Düğümleri. ....	18
Şekil 3.2. Waspote Üstten Görünüş. ....	20
Şekil 3.3. Waspote Altan Görünüş.....	21
Şekil 3.4. On Modundan Uyku Moduna Geçiş Şeması. ....	22
Şekil 3.5. On Modundan Derin Uyku Moduna Geçiş Şeması. ....	22
Şekil 3.6. On Modundan Kış Uykusu Moduna Geçiş Şeması. ....	23
Şekil 3.7. Veri Sinyalleri Blok Diyagramı. ....	24
Şekil 3.8. Enerji Sinyalleri Blok Diyagramı.. ....	25
Şekil 3.9. Akıllı Şehir Sensörleri.....	26
Şekil 3.10. Toz Sensörü Grafik Örneği. ....	26
Şekil 3.11. LD Sensörü. ....	27
Şekil 3.12. LD Sensör Kullanım Örneği. ....	27
Şekil 3.13. Gürültü Sensörleri Ölçüm Değer Aralıkları.....	28
Şekil 3.14. Xbee Modülü. ....	29
Şekil 3.15. Bluetooth Modülü. ....	30
Şekil 4.1. RESTful Web Servis Çalışma Diyagramı .....	35
Şekil 5.1. Oluşturduğumuz Veritabanı Kayıtlarından Bir Örnek.....	38
Şekil 5.2. Veritabanı Diyagramı.....	39
Şekil 5.3. Use Case Diyagramı. ....	40
Şekil 6.1. Nesnelerin İnterneti Ağ Topolojisi .....	41

## TABLÖLAR LİSTESİ

Tablo 3.1. Waspote Kablosuz Bağlantı Standartları. ....	29
Tablo 3.2. Maliyet ve Karşılaştırma Tablosu. ....	32
Tablo 3.3. Giriş/Çıkış Üniteleri Bazında Karşılaştırma. ....	32
Tablo 3.4. Bellek ve Mikrokontrolcü Bazında Karşılaştırma. ....	32
Tablo 3.5. Enerji Tüketimi Bazında Karşılaştırma. ....	32



## ÖZET

Anahtar kelimeler: Kablosuz algılayıcı eyleyici ağlar, IoT, Wasmote, Sensör, Node.js, RESTful

İnsanların her gün kullandığı fakat dikkat etmediği; sokağının başında, okulunun bahçesinde, arabasının yanında duran birçok nesne mevcuttur. Çöp konteynerleri, sokak lambaları, park alanları, trafik ışıkları, yol çizgileri ilk etapta örnek olarak verilebilir. Tüm bu buluşlar insan hayatını kolaylaştırmak ve şehir yaşamını belirli kurallar çerçevesinde sürdürebilmek için yapılmıştır. Peki bu nesnelerin kullanım amaçları değiştirilmeden hem insanlar için çok daha kullanışlı hale getirilip hem de hizmet ettikleri amaç ve görevleriyle ilgili bir takım geliştirmeler ve değişiklikler yapılabilir mi ?

Kablosuz ağ uygulamaları için kullanılan Wasmote ve beraberinde gelen sensör kitiyle bu sorulara cevap verilmiştir. Örneğin karbondioksit gibi belli oranın üzerine çıkıldığında zararlı hale gelen gazların ölçümünü sağlayıp yine Wasmote ile haberleştirerek belirli bir oranın üzerindeki ani değer artışları saptanarak, gerekli önlemlerin alınması sağlanmıştır.

Sensörlerden toplanan veriler node.js ve RESTful mimarisi yardımıyla geliştirdiğimiz web servis ile internet üzerinden akıllı cihazların hizmetine sunulmuş, ölçülen bu değerler analiz edilmiştir. Projemiz dahilinde sensörlerden faydalanarak tam teşekküllü bir gözetleme uygulaması geliştirilmiştir.

## BÖLÜM 1. GİRİŞ

Aynı insanların çevrelerinde olup bitenleri duyu organlarıyla algıladığı gibi sensörler de makineler için duyu organı görecektir şekilde tasarlanmış olup etraftaki değişiklikleri ölçmemize yararmaktadır. Fakat bu sensörler ölçtükleri değerler ile ne gibi değerlendirmeler yapılması gerektiği gibi bir hususta kendilerine pay düşmediğinin farkında değillerdir. Ölçümlerine anlam kazandırabilmek adına bu ölçümlerin işlenmesi gerekir. Hangi sensörü kullanırsak ve ne çeşit veriler elde edersek edelim bu durum baki kalmaktadır.

Mikroişlemciler bu iş için yaygın kullanımdadırlar. Ölçülen değerlerin işlenmesi ve buna göre aksiyon alınabilmesi için çeşitli araçlar geliştirilmiştir. Biz projemizde Wasmote adlı cihaz yardımıyla uyumlu sensörlerden ölçülen değerleri işleyip bu değerleri bir nosql veritabanı çeşidi olan mongodb veritabanına kaydederek, nodejs çatısının yardımıyla internet ortamına aktarmaya çalıştık.

Bunun için öncelikle Wasmote cihazına entegre olarak gelen akselerometre yardımıyla x,y ve z eksenlerinde yer-yön değişikliklerini ölçümledik ve elde ettiğimiz değerleri dinamik olarak değişiklik gösteren bir web ortamına dahil ettik. Bu süreçte kullandığımız yapılara dahil ileri açıklamalar çalışmamızın devamında yapılacaktır.

## **BÖLÜM 2. NESNELERİN İNTERNETİ VE UYGULAMALARI**

Bu bölümde projenin konusu detaylandırılacak ve projenin gerçekleştirilmesi için kullanılan yazılım ve donanımlara ilişkin bilgiler verilecektir.

### **2.1. Projenin Konusu ve Nesnelerin İnterneti**

#### **2.1.1. Projenin Konusu**

Waspnote üzerinde entegre olarak gelen akselerometre ile yapılan ölçümlerin web ortamına taşınması olan konumuz için şu şekilde bir kullanım senaryosu düşündük. Bir büyükşehir belediyesi tahsisine sunduğu araçların kullanıcılarının yerini anlık olarak görmek istiyor olabilir. Bu amaçla istenildiği taktirde yazılan uygulama sayesinde baz alınan bir noktaya göre mevcut uzaklığın tespiti sağlanabilir.

Yine bir belediye için çalışanların iş güvenliğinin sağlanması adına mesai saatleri içerisinde beline takacağı bir kutu ile örneğin aksislerdeki değişiklik neticesinde yere düştüğünü anlayabilmemizi sağlayacak bir sistem geliştirebilmemiz mümkün.

#### **2.1.2. Nesnelerin İnterneti Kavramı ve Tarihçesi**

Kablolu iletişim araçları yirminci yüzyılın başından beri bilgi alışverişi için sinyal kullanabilmektedir. Hücresel iletişim yöntemlerinin gelişmesi ve bilgisayar tabanlı otomasyonun endüstri de öne çıkmasıyla M2M kavramı ağ tarafında büyük önem kazandı.[1]

Telefon haberleşmesini ve bilgisayarları birlikte kullanan ilk M2M uygulama Theodore G. Paraskevakos tarafından kullanıcı kanalı tranıma sistemi geliştirilirken 1968 yılında kullanılıp daha sonra 1973 yılında Amerika’da lisanslanmıştır. Bu sistem 1920’lerde kullanılan panel bazlı haberleşmeye benzer fakat her kullanıcıya atanan ve caller\_id adı verilen bir metotla farklılaşmıştır.

İlk denemelerden sonra Theodore telefonların kullanıcılarını anlayabilmeleri adına bir çeşit zekaya sahip olmaları gerektiğini anlamıştır. Arayan kişiyle numarasını eşleştiren bu sistem ilk olarak 1971 de kullanılmıştır.



Şekil 2.1. İlk kimlik doğrulama alıcısı

Bir yandan hücresel haberleşme daha yaygınlaşırken bir çok makina IP ağına bağlanabilmek için hala kablolu sistemler kullanıyordu.

1995 yılında cihazların kablosuz haberleşebilmeleri için ilk adımı Siemens firması mobil departmanı altına M1 adlı GSM modülünün geliştirilmesi için yine kendi üretimi S6 mobil telefonlarını baz alan M2M uygulamaları için özelleştirilmiş birimlerin kullanımını sağladı. 2000 yılında modül departmanı mobil departmanından ayrılarak “Kablosuz Modül” departmanı başlığı altında tekrar ele alındı ve bu platform 2008 yılında Cinterion Wireless Modules adı altında nihai halini aldı.[2]

İlk M1 modülü uzaktan araç takibi yapılabilmesi için POS terminallerinde kullanıldı.

Nesnelerin internet kavramının bugün ki ivmesini kazanmasında sim kartlar gibi makine optimize hale getirilmiş teknolojiler, gömülü JAVA, GPS gibi teknolojiler bir çok firmanın öngörüsü ve katkısı sayesinde yardımcı olmuştur.

Nesnelerin internet tam bu noktada ve günümüzde fiziksel obje ya da nesne olarak adlandırılabilir elektronik cihazlar, sensörler ve ağ bağlanılabilirlik bazında veri alış verişinde bulunabilen her donanımı kapsamakta ve M2M kavramını da içine alarak bir çatı oluşturmaktadır. Bu çatı altında geliştirilen yeni yazılım teknolojileri her geçen gün bu sektörün daha da büyümesine ön ayak olmaktadır.

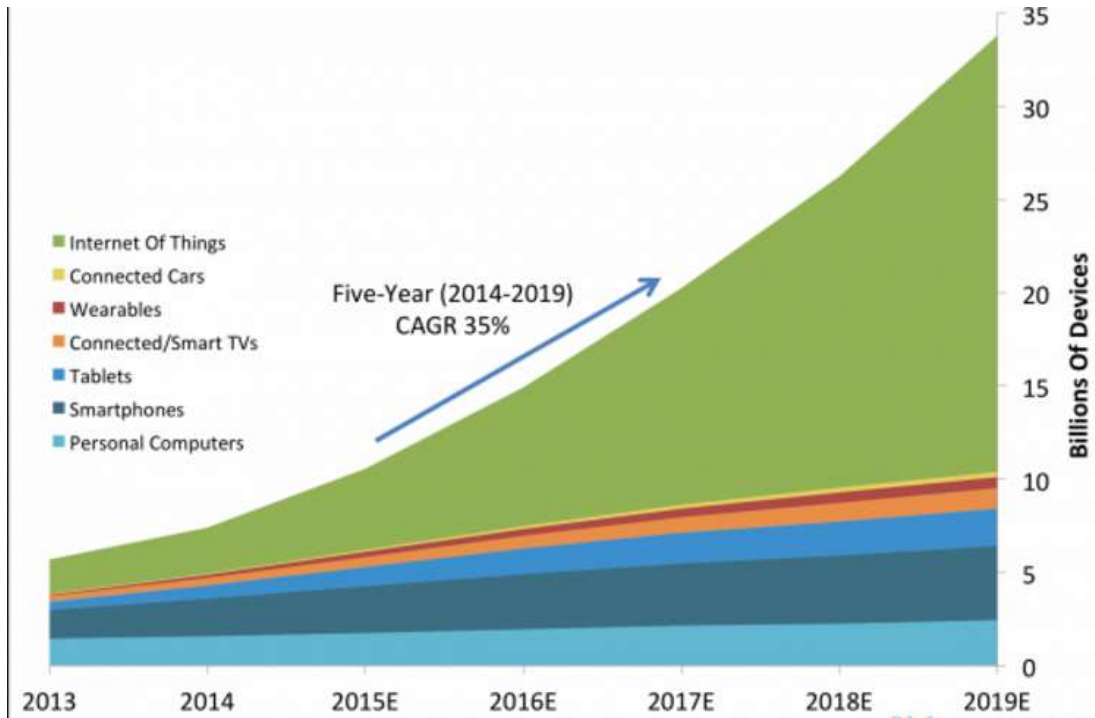
Nesnelerin uzaktan mevcut ağı kullanarak sensörler tarafından bilgi alınabilmesi ve kontrol edilebilmesi için geliştirilen bu sistemler, fiziksel dünya ile bilgisayarlar arasında direk bir iletişim sağlanmasının önünü açmakla kalmayıp daha etkili, tutarlı ve ekonomik kazançları ön planda tutarak akıllı evler, akıllı ulaşım, akıllı şehirler ve akıllı tarım gibi çok daha kompleks ama gerekli alanları da içine alır ve uygulama tarafında güzel işler çıkartılmasını sağlar.

### **2.1.3. Nesnelerin İnterneti Kavramına Dair İstatistikler**

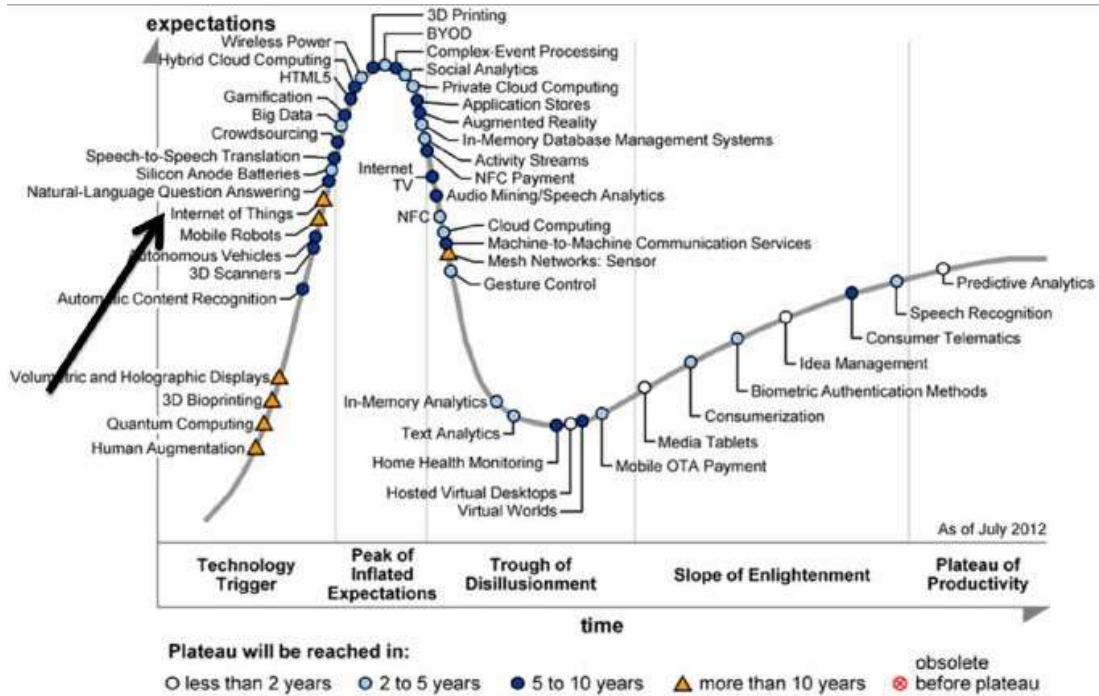
Aşağıdaki bölümde hala büyümekte ve üzerine tahminler, yatırımlar yapılmakta olan nesnelerin internet kavramına ilişkin kullanım alanı, diğer teknolojiler arasındaki mevcut yeri ve ne gibi nesnelerin internet ortamına bağlandığına ilişkin istatistikler yer almaktadır.[3][4][5][6][7]



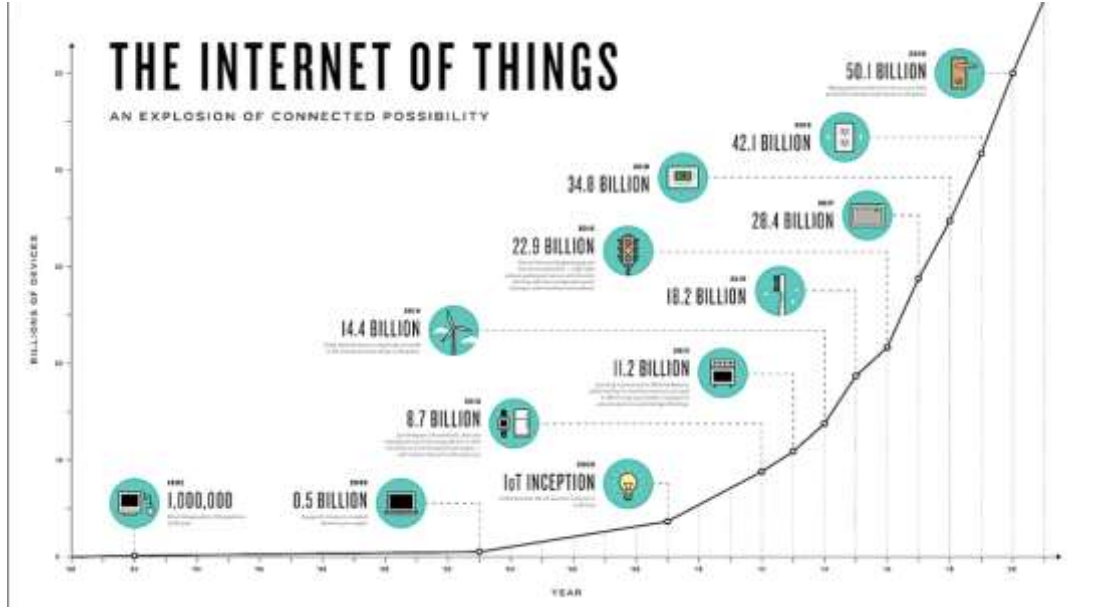
Şekil 2.2. Intel firmasının sitesinden alınan ve kullanım alanlarına göre kategorilendirilmiş bir infografik



Şekil 2.3. Dünya üzerinde İnternete çıkabilen cihazların çeşitleri ve 2019 yılına değin nasıl bir yol izleyeceklerine dair yapılmış tahminleri gösteren bir grafik (Kaynak var)



Şekil 2.4. Gelecek 10 Yıl İçin IoT Tahmini



Şekil 2.5. Nesnelerin İnterneti Kavramına Dahil Olan Cihazlar ve Kullanım Alanlarına İlişkin Yıl Bazında Değerlendirmeler İçeren Çizgi Grafik



## BÖLÜM 3. PROJENİN GERÇEKLENMESİ

### 3.1. Projenin Amacı

Projemizde Wasmote entegresi yardımıyla bu entegrenin üzerine hali hazırda dahil edilmiş 3-yönlü akselerometre ile x,y ve z bazında ölçümler alınarak bu ölçümleri mongodb veritabanına kaydetmeyi ve web ortamında bu ölçümleri sonuçlarını dinamik bir şekilde göstermek amacıyla hareket ettik.

Farklı entegrelerle yapılmış benzer çalışmalar olduğu ve bu çalışmamızın üzerine konulabilecek farklı diğer çalışmalar olduğunun bilincinde hareket ettik.

### 3.2. Kullanım Senaryosu

Kullanım senaryosu için ise akıllı şehir uygulamalarında iş güvenliği tarafında bir çalışma yapıldığını düşünerek zorlu şartlar altında çalışan bir işçinin güvenliğini riske sokabilecek bir durumda örneğin yere düşmesi halinde iş veren tarafından farkedilebilmesini ve hemen müdahale edilebilmesini sağlamak adına geliştirilebilecek bir cihaz için Wasmote entegresini kullanabileceğimizi düşündük.



Figure : Typical QTAG process

Şekil 3.1. Wasmote Düğümleri

### 3.3. Neden Wasmote Entegresi Tercih Edildi ?

Wasmote açık kaynak kodlu bir sensor platformu olarak öncelikle iki noktada rakiplerinden sıyrılmayı başarıyor. Düşük güç tüketim modları ile düğümler ya da terminallerle haberleşirken sadece bataryasından güç çekecek kadar otonom olması sağlanıyor ve özelleştirilmiş bir platform olduğu için tak-çalıştır metodunu benimsemiş olması kurulum ya da güncelleme durumlarında kullanıcıya oldukça yardımcı oluyor. Açık kaynak kodlu olması ve Arduino gibi bir platformun geliştiricilerini de benzer kütüphaneler kullanması sebebiyle kendisine çekiyor olması, tasarım aşamasında ya da teknik anlamda bir sorun yaşandığında çözüm bulunabilmesini kolaylaştırıyor.[8]

İlk olarak 2007 yılında Libelium firmasının Arduino için bir XBee Shield' ı tasarlamasıyla başlayan süreç Arduino ile kablosuz sensor ağları kurmak fikrine dayanıyor. İki sebeple Libelium firması Arduino ile olan ortaklığını bırakıp kendi yoluna gitme kararı almıştır. İlk Arduino nun 5V-3.3V enerji tüketim değerleri ki Libelium firması düşük güç tüketimi için Arduino entegresine uyku modu eklemek istiyordu fakat bu voltaj değerleri 50mA bir bataryayı bir kaç günde tüketmekten öteye gidemiyordu. Diğerleri ise akıllı şehir, akıllı ev uygulamaları için bu platformun bazı sertifikalara ihtiyaç duyuyor olmasıydı. Bu iki noktada tıkanan Libelium ekibi tamamiyle modüler ve düşük güç tüketimi odaklı fakat Arduino nun geliştirdiği bir takım kütüphane ve geliştirme ortamından ödün vermeden Wasmote platformunu kurdular.[9]

Hali hazırda Wasmote platformu ile kullanılabilen modüller aşağıdaki gibidir.

ZigBee/802.15.4 (2.4GHz, 868MHz, 900MHz)

LoRaWAN (868, 900-915 and 433 MHz bands)

LoRa (868 and 900-915 MHz bands)

Sigfox (868 MHz band)

GSM - 3G/GPRS (Quadband: 850MHz/900MHz/1800MHz/1900MHz)

Sensor (Sensor boards)

Storage : SD Memory Card

GPS Module

Donanımsal özellikleri ise şu şekilde sıralanabilir.

Mikrokontrolcü: ATmega1281

Frekans: 14MHz

SRAM: 8KB

EEPROM: 4KB (1KB rezerve)

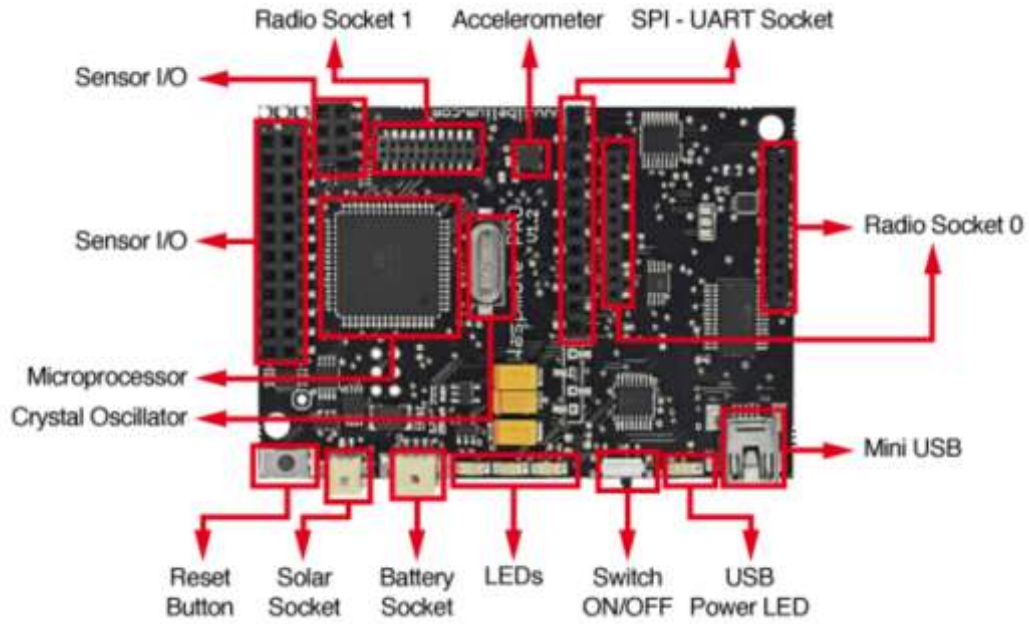
FLASH: 128KB

SD Card: 2GB

Ağırlık: 20gr

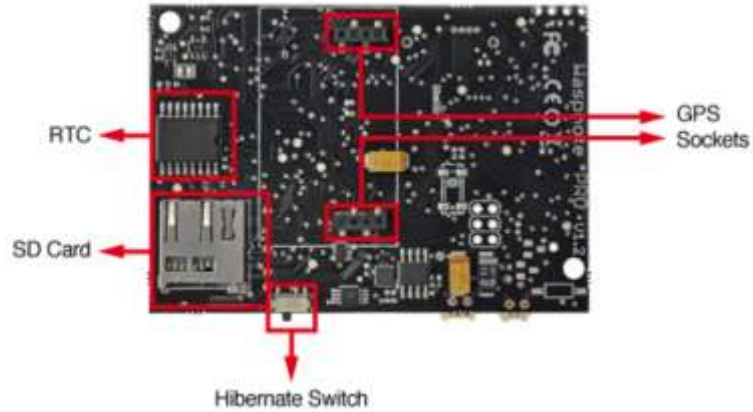
Boyutlar: 73.5 x 51 x 13 mm

Sıcaklık Aralığı: [-10°C, +65°C]



Şekil 3.2. Waspote Üstten Görünüş

Şekil 3.3. Wasmote üstten görüş



Şekil 3.3. Wasmote Altan Görünüş

### 3.4. Donanım İçin Gerekli Enerji Analizi

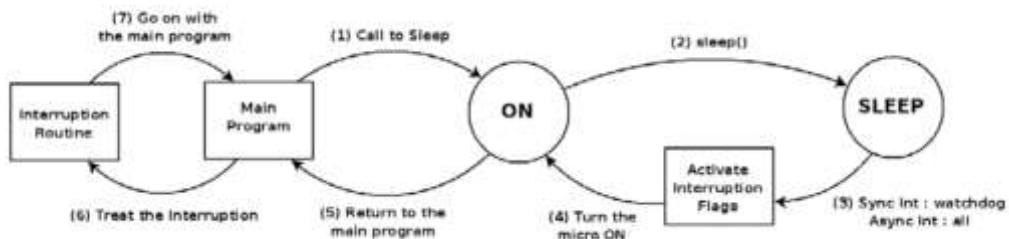
Wasmote entegresinin dört tane operasyonel moda sahiplik yapar. Bu uyku modlarını kullanarak sahadaki ömrü 1 ila 5 yıl arasında olan Wasmote güneş enerjisi paneli bağlanırsa bu süreyi aşabilmektedir.[10]

#### 3.4.1. ON Modu

Normal operasyon modudur. Enerji tüketimi 9mA.

#### 3.4.2. Uyku (Sleep) Modu

Bu durumdayken ana program duraklatılmıştır, mikrokontrolcü saklı duruma geçer ki bu durum tüm asenkron kesmelerde ya da yalnızca watchdog tarafından üretilen senkron kesmelerde tekrar eski halini alır. Bu durumun süresi 32ms ile 8s arasında değişir. Enerji Tüketimi 62µA.



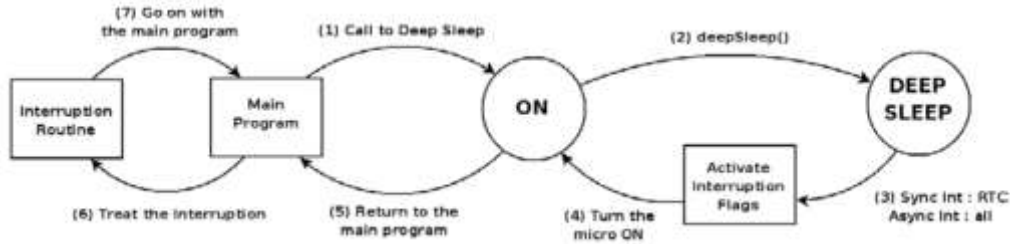
Şekil 3.4. On Modundan Uyku Moduna Geçiş Şeması.

Aşağıdaki kod örneği geliştirme ortamı üzerinde kullanılarak Waspote cihazı 32ms kadar uyku moduna sokulabilir.

```
{
    PWR.sleep(WTD_32MS, ALL_OFF);
}
```

### 3.4.3. Derin Uyku (Deep Sleep) Modu

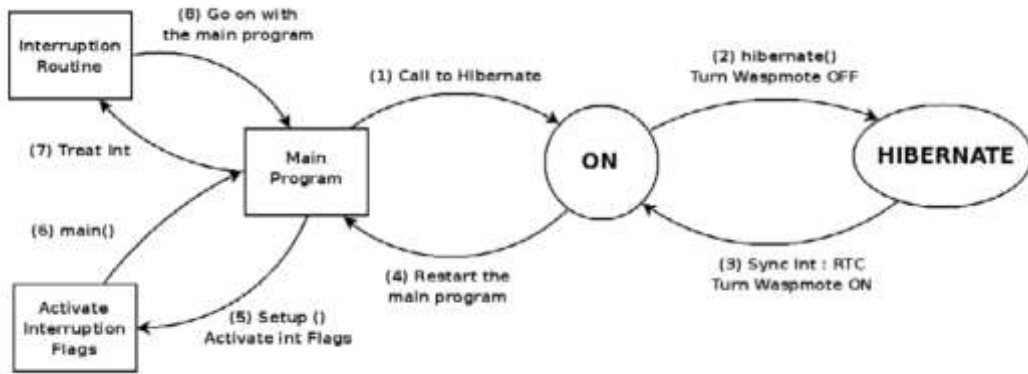
Bu durumdayken ana program duraklatılmıştır, mikrokontrolcü saklı duruma geçer ki bu durum tüm asenkron kesmelerde ya da yalnızca RTC tarafından üretilen senkron kesmelerde tekrar eski halini alır. 8 saniyeden dakikalar, saatler, günlere kadar sürebilir. Enerji tüketimi 62µA.



Şekil 3.5. On Modundan Derin Uyku Moduna Geçiş Şeması

### 3.4.4. Kış Uykusu (Hibernate) Modu

Bu durumdayken ana program durdurulmuştur. Waspote ve diğer tüm modüller arası bağlantılar tamamiyle kopar. Tek yöntem RTC içerisinde daha önceden ayarlanmış senkron kesmedir. 8 saniyeden dakikalar, saatler, günlere kadar sürebilir. Ana batarya ile arasındaki bağlantı tamamen kopmuş olduğu için RTC ek bataryadan beslenir ve sadece 0.7µA güç tüketir.



Şekil 3.6. On Modundan Kış Uykusu Moduna Geçiş Şeması

Bu moddan doğru bir şekilde çıkabilmek için hibernate kablosuna ihtiyaç vardır. Aşağıdaki metotlar izlenir.

Bataryayı bağlarız.

Buton bataryasını bağlarız

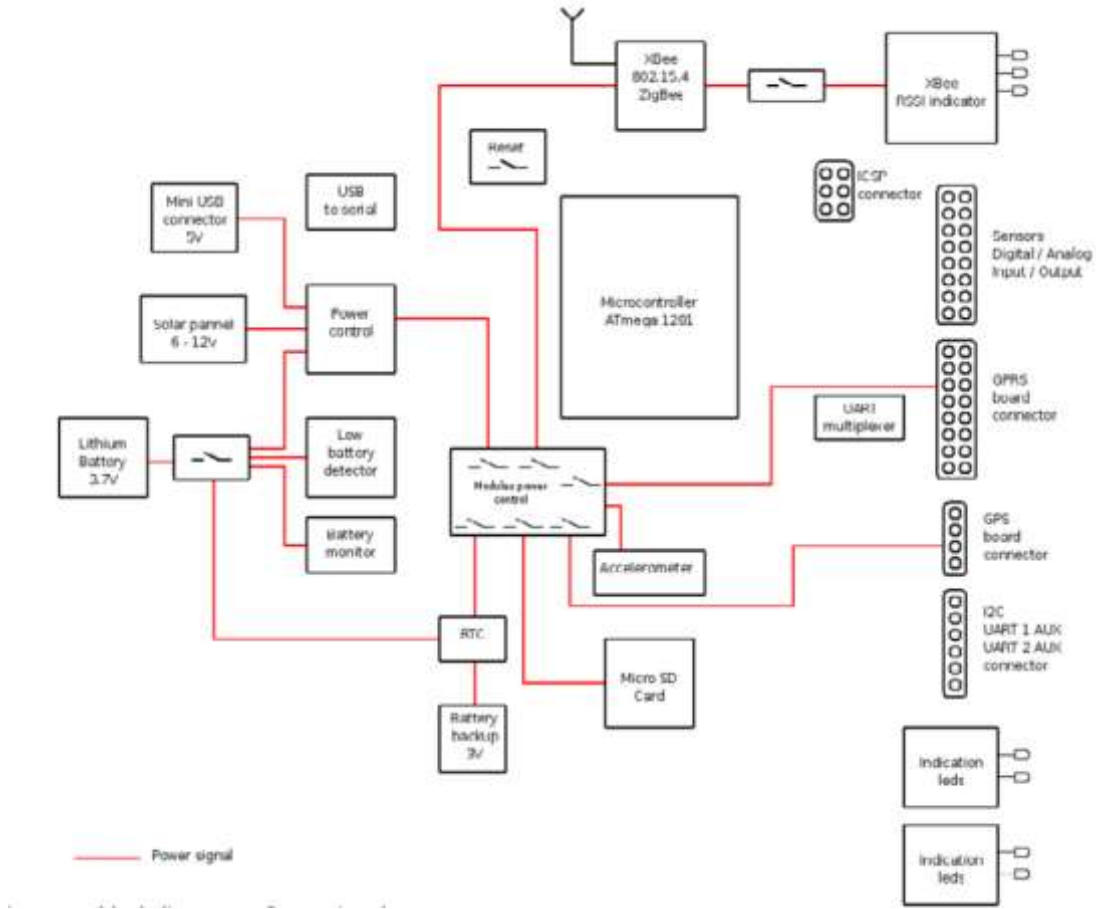
Wasp mote u ON moduna getiririz.

Kırmızı ışığın yanıp sönmesini bekleriz.

Kırmızı ışık söndüğünde hibernate kablosunu çıkartabiliriz.

```
{ PWR.hibernate("02:01:30:00", RTC_OFFSET, RTC_ALM1_MODE2); }
```



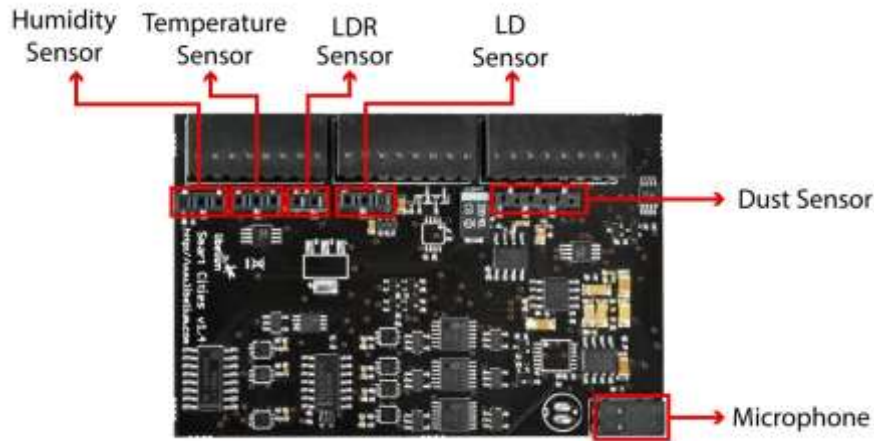


Şekil 3.8. Enerji Sinyalleri Blok Diyagramı.

### 3.6. Waspnote Sensörleri

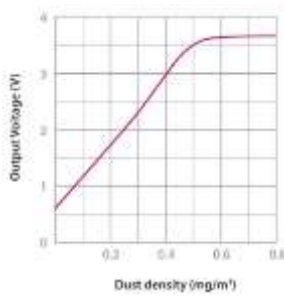
Waspnote yaklaşık 80 adet sensor ile uyumlu olarak çalışabilmektedir. Bu bölümde akıllı şehir uygulamaları için kullanılabilecek olan bazı sensörlere değinilecektir. Bizim kullandığımız akselerometre Waspnote un üzerine entegre olduğu için sensor bulmakta ya da kullanmakta herhangi bir güçlük yaşamadık. Fakat başka diğer sensörler bağlanmak istendiğinde modülerlik bir kenara bırakılmasın da isteniyorsa farklı bazı yapılar devreye entegre edilmek durumunda kalınabilir. Firma tarafından kablosuz olarak haberleşmek üzere şehrin belirli noktalarına dağıtılmış algılayıcı cihazlardan oluşturulan bu ağdan veri toplamak için ayrıca meshlium adında bir terminal geliştirilmiştir. [11][12]





Şekil 3.9. Akıllı Şehir Sensörleri

### 3.6.1. Toz Sensörü ( GP2Y1010AU0F)



İçerdiği ILED diyot tarafından emilen kızılötesi ışığın emilimine göre çalışan optic bir sensördür. Bu kod bloğu değer okumasını sağlayabilir.

Şekil 3.10. Toz Sensörü Grafik Örneği

```
{
  SensorCities.ON();
  SensorCities.setSensorMode(SENS_ON, SENS_CITIES_DUST);
  delay(2000);
  float dust_value;
  dust_value = SensorCities.readValue(SENS_CITIES_DUST);
}
```

### 3.6.2. Çatlak Denetleme Sensörü

Üzerinde fiberglass bir film kaplı olan bu sensör çok düşük dirence sahip küçük kondüktiv bir tel içerir. Çatlak oluştuğunda bu mekanizma bozulur ve logic sıfır Waspnote'a döndürülür. Böylece sistem bir çatlak oluştuğunu anlar. (Analog5 pinine oturtulur.)

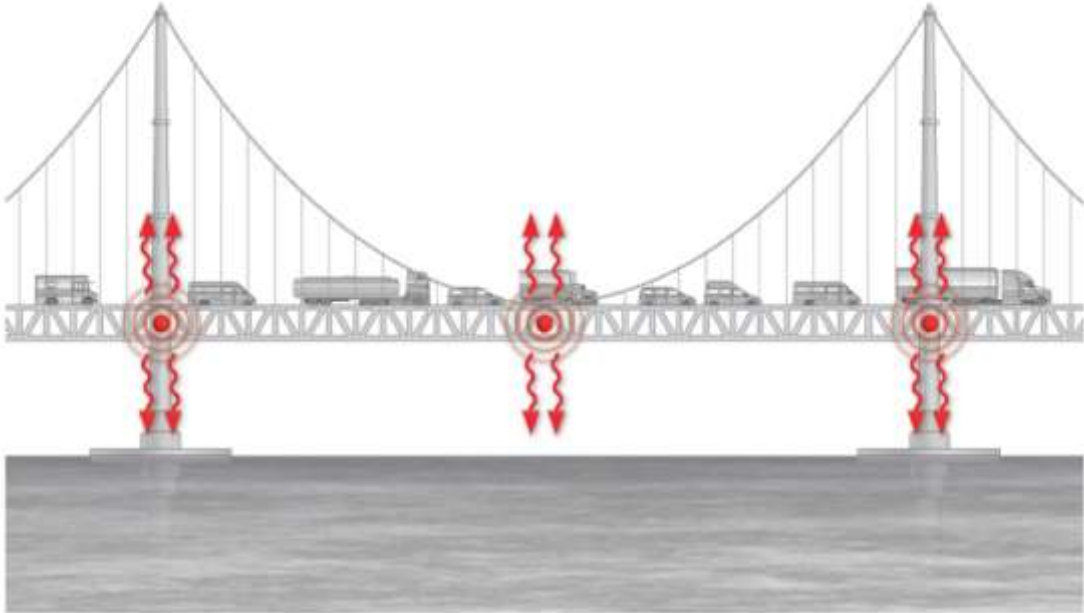
### 3.6.3. LD Sensörü



Yine çatlakların ölçülmesi hususunda Waspote ile kullanılabilecek olan bu sensor üzerindeki oynar yapıyla birlikte bir çatlak oluştuğunda genişlemek suretiyle

Şekil 3.11. LD Sensörü

topladığı verileri Waspote cihazına iletir. Analog67 pini üzerinden bağlanır. Bir destek voltajına ihtiyaç duyduğunda ise bunu Dijital1 pininden alabilmektedir.



Şekil 3.12. LD Sensör Kullanım Örneği

### 3.6.4. Gürültü Sensörü (Microphone POM-2735P-R)

20Hz ile 20kHz arasında çalışan bu sensor neredeyse insanların işitme düzeyiyle aynı frekansta düzgün ölçüm yapabilmektedir. Çevre gürültüsünü ölçmek için dahili mikrofona ile kullanılagelmiştir. Üzerinde EEPROM bulunmaktadır. Bunun üzerine veri yazarken dikkat edilmelidir. Analog6 pininden çıkış alınabilmektedir.

### 3.6.5. Nem Sensörü

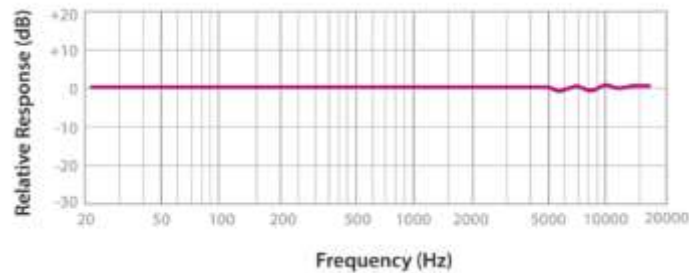
Havadaki nem deęerlerini ölçebilmek için kullanılır. Çıkış voltajı 0,48 ile 2,34V arasına ayarlanmıştır. Bu deęerlerin üstüne çıkıldığında bir terslik olduğuna dair Waspote u uyarır.

### 3.6.6. Işık Sensörü (LDR)

400nm – 700nm arasında ölçüm yapabilmektedir. Ölçümler Waspote'un analog-dijital çeviricisinden geçerek mikrokontrolcü tarafından işlenir. Beslemesi Dijital2 pini üzerinden yapılabilmektedir.

### 3.6.7. Sıcaklık Sensörü (MCP9700A)

Çıkış voltajı 100mV(-40 santigrat derece) 1.75V(125 santigrat derece) seviyesinde olup Analog4 pinini kullanmaktadır. Dijital2 beslemesi kullanılır.



Şekil 3.13. Gürültü Sensörleri Ölçüm Deęer Aralıkları.

## 3.7. Waspote Bağlantı Kurulum Modülleri

Wasmote'un kullanabildiği yaklaşık 17 adet kablosuz iletişim modülü bulunmaktadır. Bunlardan XBee ve Bluetooth modüllerine değineceğiz. Tüm modüller aynı soket üzerinden bağlantı kurabilmektedirler.

Model	Protokol	Frekans	txPower	Mesafe *
XBee-802.15.4-Pro	802.15.4	2.4GHz	100mW	7000m
XBee-S55	RF	868MHz	50mW	12km
XBee-900	RF	900MHz	315mW	1000m
LoRaWAN	LoRaWAN	868 and 433MHz	55mW	- 1m - Typical base station range
LoRa	RF	868 and 900 MHz	14dBm	22km
Sigfox	Sigfox	868MHz	14dBm	- 1m - Typical base station range
WiFi	802.11b/g	2.4GHz	0dBm - 12dBm	50m-500m
GPRS-Pro and GPRS+GPS	-	850MHz/900MHz/ 1800MHz/1900MHz	2W(Class4) 850MHz/900MHz 1W(Class1) 1800MHz/1900MHz	- Km - Typical carrier range
3G/GPRS	-	Tri-Band UMTS 2100/1900/900MHz Quad-Band GSM-EDGE 850/900/1800/1900 MHz	UMTS 900/1900/2100 0.25W/GSM 850MHz/900MHz 2W/DCS1800MHz/PCS1900MHz 1W	- Km - Typical carrier range
Bluetooth Low Energy	Bluetooth	2.4GHz	3dBm	100m

Tablo 3.1. Wasmote Kablosuz Bağlantı Standartları

### 3.7.1. XBee Kablosuz Ağ Protokolü

IEEE 802.15.4 protokolünü esas alan bir kablosuz ağ protokolüdür. Wasmote uygulamaları için kullanılan ve IP bilgisi gerektirmeden direk Ethernet üzerinden iletişim kurulmasını sağlayan önemli bir standarttır.



Protocol	Frequency	TX power	Sensitivity	Channels	Distance
802.15.4	2,405 – 2,485GHz	63.1mW	-100dBm	12	7000m

Antenna: 2dBi/5dBi  
Encryption: AES 128b  
Signal Control: RSSI  
Standards: IEEE -802.15.4 - Compliant

Şekil 3.14. Xbee Modülü

### 3.7.2. Bluetooth Protokolü

WPAN çatısı altında konumlandırılabilir bu standart RF teknolojisi kullanır. 1994 yılında Ericsson firması tarafından cep telefonları ve diğer mobil cihazlar arasında iletişim kurulabilmesi amacıyla geliştirilmiştir.

2.4 GHz ISM serbest frekans bandını kullanır. Master-slave yapısını kullanmaktadır. Seçilen cihazlardan birisi master konumunda bulunur ve diğer cihazlar bu cihaza hizmet eder. 4. Versiyonun çıkmasıyla birlikte enerji tüketimi kısmında da önemli geliştirmeler yapılmıştır.



Şekil 3.15. Bluetooth Modülü

### 3.8. Wasmote ile Arduino Karşılaştırması

Kablosuz ağ uygulamalarında kullanılan çok daha ulaşılabilir ve popüler olan Arduino platformu ile kullandığımız Wasmote entegresini karşılaştırma gereği duyduk. Böylece bu cihazı kullanmayı neden tercih ettiğimize dair daha ikna edici sonuçlar elde edeceğimizi düşünüyoruz.

#### 3.8.1. Geliştirme Ortamı Bazında Karşılaştırma

Wasmote, Arduino ile aynı geliştirme ortamını kullanmaktadır. Yine aynı şekilde kodlar ve kütüphanelerde birbiriyle uyumludur. Arduino platformunun hali hazırda kullandığı geliştirme ortamının Wasmote'un gereklerini de yeterince karşıladığı düşünüldüğü için bu yola gidilmiştir.

### 3.8.2. Kullanım Amaçları Bazında Karşılaştırma

Waspote ile Arduino platformlarını bu alanda rakip olarak düşünmek pek doğru olmaz. Paket içeriği dışında ayrıca alınabilecek entegrelerle tek başına çok uygun fiyatlı satılan Arduino platformu da Waspote kadar kablosuz ağ uygulamaları için kullanılabilir hale getirilebilir. Her iki platformun odaklandıkları alanlar farklıdır.

Waspote özellikle kablosuz iletişim tarafına odaklandığı için board üzerinde dahili olarak kesme üretebilmek için RTC saati, sensörlerden elde edilen verileri depolayabilmek için uSD yuvası, 3-yönlü akselerometre (düğümlerin yerlerinden oynaması durumunda farkedilebilmelerini sağlamak adına çok kullanışlıdır) ve batarya, güneş enerjisi paneli gibi girişler hali hazırda bulunmaktadır.

	Arduino UNO	Arduino Mega 2560	Waspote
Board	22,00 €	41,00 €	155,00 €
Arduino Xbee 802.15.4 + 2dBi antenna	45,00 €	45,00 €	
Triple axis accelerometer	7,75 €	7,75 €	
On Board Programmable LED + ON/OFF Switch	1,00 €	1,00 €	
RTC DS3234 + Button Battery	16,00 €	16,00 €	
uSD Adaptor	20,00 €	20,00 €	
Solar Panel Socket	47,00 €	47,00 €	
6600mAh Battery			30,00 €
Total	158,75 €	177,75 €	185,00 €

Tablo 3.2. Maliyet ve Karşılaştırma Tablosu

### 3.8.3. Giriş/Çıkış Üniteleri Bazında Karşılaştırma

Model	Analog In	Digital I/O	UART's	SPI	I2C	PWM	USB
Arduino	6	8	1	Yes	Yes	6	Yes
Arduino Mega	16	54	4	Yes	Yes	15	Yes
Waspnote	7	8	2	Yes	Yes	1	Yes

Tablo 3.3. Giriş/Çıkış Üniteleri Bazında Karşılaştırma

### 3.8.4. Bellek ve Mikrokontrolcü Bazında Karşılaştırma

Model	Microcontroller	Frequency	RAM	EEPROM	FLASH	External Storage (SD card)
Arduino	ATMega328	16MHz	2KB	1KB	32KB	-
Arduino Mega	ATMega2560	16MHz	8KB	4KB	256KB	-
Waspnote	ATMega1281	14MHz	8KB	4KB	128KB	2GB

Tablo 3.4. Bellek ve Mikrokontrolcü Bazında Karşılaştırma

### 3.8.5. Enerji Tüketimi Bazında Karşılaştırma

Model	Consumption ON	Sleep mode	Consumption Sleep mode	Hibernate mode	Consumption Hibernate mode
Arduino	50mA	No	-	No	-
Arduino Mega	50mA	No	-	No	-
Waspnote	15mA	Yes	55µA	Yes	0.7µA

Tablo 3.5. Enerji Tüketimi Bazında Karşılaştırma

## BÖLÜM 4. KULLANILAN YAZILIMLAR

### 4.1. Node.js Çalıştırma Ortamı

Açık kaynaklı, sunucu tarafında çalışan ve ağ ilintili uygulamalar için geliştirilmiş bir çalıştırma ortamıdır. Node.js uygulamaları genelde istemci tarafı betik dili olan JavaScript kullanılarak geliştirilir. Cross-platform bir çalıştırma ortamıdır. Bunun anlamı OS X, Microsoft Windows, Linux, FreeBSD, NonStop, IBM AIX, ve IBM i gibi bir çok işletim sisteminde çalıştırılabilir olmasıdır.

En önemli avantajı JavaScript'in sağladığı non-blocking I/O olanağı sayesinde yüksek ölçeklenebilirlik (ing. scalability) ve yüksek veri aktarabilme yetenekleridir. Bu teknolojiler sıklıkla gerçek zamanlı web uygulamalarında tercih edilmekle beraber, kullanım alanı popülaritesiyle orantılı olarak genişlemiştir.

Node.js Google V8 JavaScript motorunu kullanarak betik dilini yorumlar. Node.js, içerisinde standart olarak dağıtılan kütüphaneler sayesinde ek bir sunucu yazılımına (Apache HTTP Sunucusu, Nginx, IIS vb.) gerek kalmadan uygulamanın web sunucusu görevini görür.

Node.js IBM, Microsoft, Yahoo!, Walmart, Groupon, SAP, LinkedIn, Rakuten, PayPal, Voxel ve GoDaddy gibi şirketler tarafından aktif olarak kullanılmaktadır.[13]

Context switching (İçerik Değiştirme) yapmadan eş zamanlı bir şekilde bir çok isteği cevaplayabilmek adına bloklanmamış giriş/çıkış çağrılarını kullanarak çalışmaktadır.

#### 4.1.1. Node Paket Yöneticisi

CPP programlama diliyle yazılmış ve JavaScript çalıştırma zamanını kullanan node.js bu sayede oldukça hızlıdır. Başlarda yalnızca server tarafında işleri kontrol etmek amacıyla kullanılan node.js geliştiricilerin yerel görev otomasyonlarında kullanmak üzere bir takım araçlar geliştirilmesiyle beraber farklı bir yönde de



gelişmeye başlamıştır. Bundan sonra front-end geliştiriciler için oldukça kullanışlı bir hale gelen node.js için bir paket yöneticisi kullanmak kaçınılmaz olmuştur.

NPM kullanmak istediğimiz paketlerin kurulmasında ve kurulan bu paketlerin kullanılmasında kullanıcıya kolaylık sağlar. Yerel modundayken paketleri node\_modules içerisine kurar npm, global modunda ise kurulan bu paketlere istediğimiz yerden erişebiliriz.

#### **4.1.2. Socket.io**

Socket.io'nun amacı, gerçek zamanlı olarak hemen hemen her tarayıcı ile (mobil cihazlar dahil) farklı iletişim protokollerine rağmen iletişim kurabilmelerini sağlamaktır.

Sunucu ve istemci tarafında anlık iletişimi sağlamak ve bunun için mevcut tarayıcılar ve farklı protokollerle çalışmayı hedeflemiş olan bu yapı oldukça kullanışlıdır.

Bizde projemizde sensörlerden okunan verilerin kullanılabilir hale getirilmesi noktasında bu yapıdan faydalandık.

#### **4.1.3. Serialport**

Blenderlardan robotlara kadar çok çeşitli alanlarda JavaScript kodu yazıp bunlarla nesneleri kontrol edebilmemiz noktasında Wasmote, Arduino gibi chipsetler ve Zigbee gibi standartlarla uyumlu çalışabilen bir sistemde en büyük yardımcılarımızdan biri seriportlar aracılığıyla aktarılan verileri işleme noktasında kullanabileceğimiz serialport modülüdür.[14]

Bu modülü yine projemizde alınan değerlerin USB'den okunması ve node.js ile beraber kullanılabilmesi adına ekledik.

#### **4.1.4. Express Framework**

Express web ve mobil geliřtirmeler için imkan sunan minimal ve kullanışlı bir node.js uygulama geliřtirme çatısıdır. Diğeri node.js modülleriyle uyumlu ve hızlı çalışan bir ortam sağlamasıyla bir çok web geliřtiricisinin tercih ettiğı bir çatıdır.

Uygulamamızda bu yapıyı yine okuduğumuz değeri web ortamında gösterme adına kullandık.

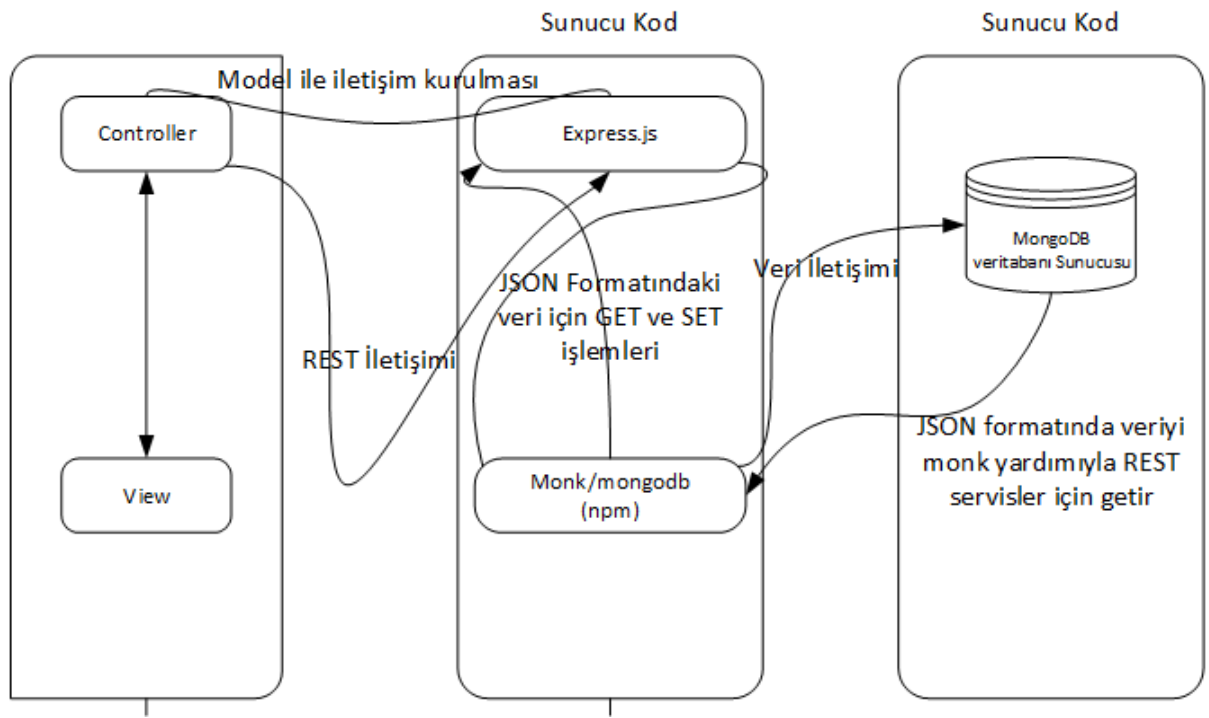
#### **4.1.5. D3.JS Kütüphanesi**

Bir javascript kütüphanesi olan D3.js döküman bazı verileri manipüle etmek için kullanılmaktadır. HTML, SVG, CSS kullanarak kullandığımız verilerin web tarafında hayat bulmasını sağlamaktadır. DOM manipölasyonu sağlayan bu kütüphane verilerimize web tarafında anlam kazandırmak adına karmaşık çatı yapıları ya da farklı diğeri başka çözümler kullanmak yerine işimizi hızlı ve kolayca halledebilmemizi sağlamaktadır.

Uygulamamızda bu yapıyı web tarafında göstermek istediğimiz verileri d3.js kütüphaneleri yardımıyla gerçeklenmiş bazı görsellerle desteklemek için kullandık.

#### **4.1.6. Web Servis Geliřtirme Süreci**

Geliřtirdiğimiz web uygulamasında verileri tutmak için NoSQL veritabanı olan MongoDB den yararlanıyoruz. Bu tarafa kaydedilen verileri daha sonar başka uygulamaların ya da servislerin hizmetine sunabilmek için RESTful API yardımıyla bir web servis geliřtirdik. Bu servisi geliřtirirken node.js paketlerinden olan express oldukça yardımcı oldu. App olarak oluşturduğumuz uygulama değışkeninin route işlemlerini yapan kısmıyla express.js çatısını kullanarak get ve post işlemleri gerçekleřtirdik.



Şekil 4.1 RESTful Web Servis Çalışma Diyagramı

## BÖLÜM 5. VERİTABANI YAPISI

Projemizde okunan veriler dinamik olarak alındığı ve alınan veriler aynı tip veriler olduğu için bir nosql veritabanı yapısı olan MongoDB yi kullanmayı tercih ettik. Detaylara aşağıda değinilmiştir.

### 5.1. MongoDB Veritabanı Yapısı ve NoSQL

#### 5.1.1.NoSQL Veritabanı

Bilgisayar bilimleri alanında NoSQL terimi klasik yani ilişkisel veritabanı sistemlerinden bir şekilde farklı olan veritabanı yönetim sistemleri için kullanılan bir kavramdır.

Bu veri depolama sistemlerinin sabit table düzenlerine ihtiyaçları olmayabilir, alışlagelmiş join işlemleri kullanılmaz, tipik olarak yatay ölçeklemeye gidilir. Makalelerde ve araştırma ekipleri tarafından bu tip veritabanı örneklerine yapılanmış bellek modeli denir. Bu kavram ilişkisel veritabanı sistemlerinin altkümesi olarak görülür.

1998’de çıkarılan SQL arayüzü olmayan hafif bir açık kaynak ilişkisel veritabanıdır. Tasarımcısı Carlo Strozzi tasarımın yapısı gereği bu sisteme NoREL yani ilişkisel olmayan anlamına gelebilecek bir ad vermek gerektiğini düşünür. “select fun, profit from real\_world where relational=false” sloganıyla 2009 yılında Atlanta’da yapılan bir konferansta bu veritabanı sisteminin geleceği üzerine çok büyük etkiler olmuştur.

Tipik modern ilişkisel veritabanları çok sayıda belgeyi indeksleme, yoğun trafiği olan sunan Web sayfalarında ve streaming medya sağlama gibi kimi yoğun veri uygulamalarında yetersiz edim göstermişlerdir. Tipik İVTYS uygulamaları ya küçük fakat sıkça oku/yaz işlemleri ya da ender yazaz büyük komut listesi komut liste (İng. batch) hareketleri (işlemleri) ayarlanır. Diğer taraftan NoSQL, yoğun oku/yaz hizmeti verir. Gerçek Dünya NoSQL yerleşimleri green badges (sosyal ağlarda başkalarınca değerlendirilen hikâyeleri gösteren belirleyiciler) için Digg’in

3 TB'lı çözümü, Facebook'un gelen postaları arama için 50 TB ve eBay'in bütün verileri için 2 PB çözümü gibi.

NoSQL mimarileri çoğu zaman sonunda tutarlı veya tek veri maddesiyle sınırlı işlemlerde zayıf tutarlılık garantisi verir. Fakat kimi sistemler, yardımcı özel yazılım tabakası ekleyerek bâzı oluşumlarda tam ACID garantisi verirler (mesela CloudTPS). Sütun depoları için şipşak yalıtım sağlayan iki sistem geliştirilmiştir. Bunlar Google'un BigTable'e dayanan Percolator sistemi ve Waterloo Üniversitesi'nde HBase için geliştirilmiş hareketli sistem (İng. transactional system). Aynı ayrı geliştirilmiş bu sistemler, benzer kavramlar şipşak yalıtım kullanarak veri yönetimi için fazladan işlemlere, ara yazılım yerleştirmesi (İng. middleware system deployment) veya ara yazılım tabakasından kaynaklanan bakıma gerek duymadan altındaki sütun deposu için garantili çok satırlı dağıtık ACID işlemleri sağlar.

Birçok NoSQL sistemi, verilerin farklı sunucularda yedeklemesini yapan dağıtık mimariyi kullanır. Bu saklamalar genelde dağıtık hash çizelgeleri ile yapılır. Böylece sistem kolayca yeni sunucular eklenerek büyütülebilir ve bir sunucunun arızalanmasına katlanılabilir.

Biz uygulamamızda döküman tabanlı bir NoSQL veritabanı olan MongoDB'yi kullandık.

### 5.1.2. MongoDB Veritabanı

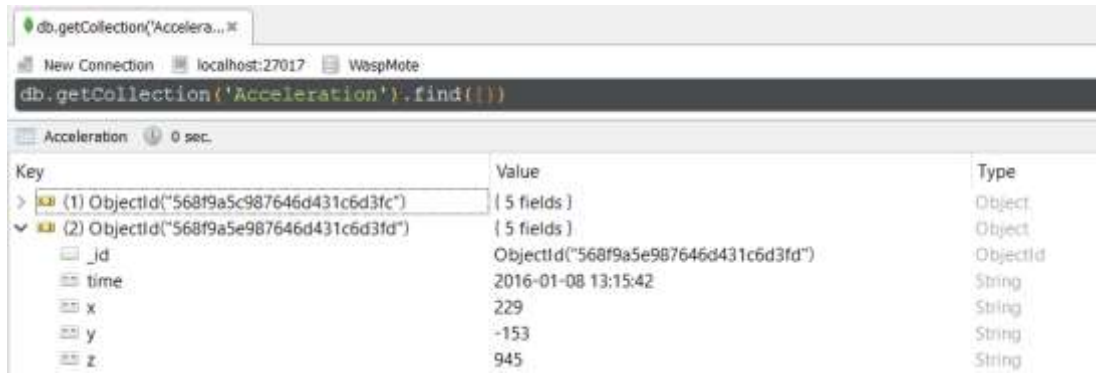
İlk sürümü 2009 yılında 10gen firması tarafından geliştirilmiş olan MongoDB (“humongous”) ölçeklenebilir, döküman tabanlı, C++ ile geliştirilmiş açık kaynak, NoSQL veritabanı uygulamasıdır. Linux, Mac Os X, Solaris ve Windows işletim sistemleri üzerinde çalışabilmektedir.

MongoDB, özellikle hız gerektiren ve geleneksel ilişkisel veritabanlarının (rdbms) hantal ve yavaş kaldığı yapılarda kullanılmaktadır. Örnek kullanım alanları arasında: yüksek hacim/içerikli problemler, analiz için veri saklanması,

MMORPG uygulamaları, caching sistemleri, web içerik yönetim sistemleri, web yorum/etiket saklama ve yönetme gibi alanlar mevcuttur.[15]

## 5.2. MongoDB Veritabanı Yapısı

İlişkisel veritabanlarından farklı olarak NoSQL veritabanları kendilerine ve kullanım amaçlarına has olarak yapılar kullanmaktadırlar. MongoDB bu aşama da kullandığı collection, schema yapılarıyla diğer sistemlerden ayrılmaktadır. Collection'lar altında aynı schema yapısı bulunmalıdır. Verileri kaydederken x, y, z axisleri için atanan değişkenler üzerine her biri o kayıta özgü ve JSON formatını kullanarak kayıt oluşturmaktadır.



Şekil 5.1. Oluşturduğumuz veritabanı kayıtlarından bir örnek

JSON formatı bir kere oluşturulduktan sonra ardından tabloya eklenen tüm veriler aynı formatta tabloya eklenecektir. Hali hazırda MongoDB kullanıyor olma sebebimizde budur. Aynı formatta ve aynı tipte verilerin hızlıca yazılıp okunmasının sağlanması amaçlanmıştır. Yukarıda kod parçası olarakta görülen ve 27017 portunun dinlenmesiyle eklenen kayıtları Acceleration collection'ının altında find() komutu ile çağırarak tek tek kayıtları görebilmekteyiz.

```
{
  "_id" : ObjectId("568f9a5e987646d431c6d3fd"),
  "time" : "2016-01-08 13:15:42",
  "x" : "229",
  "y" : "-153",
  "z" : "945"
```

}

Görüldüğü gibi her bir kayıt için atanan tek bir id vardır ve kayıtlar sistemden çekilirken bu id ler kullanılır. Bu sayede verilere çok hızlı bir şekilde erişim sağlanabilmektedir.

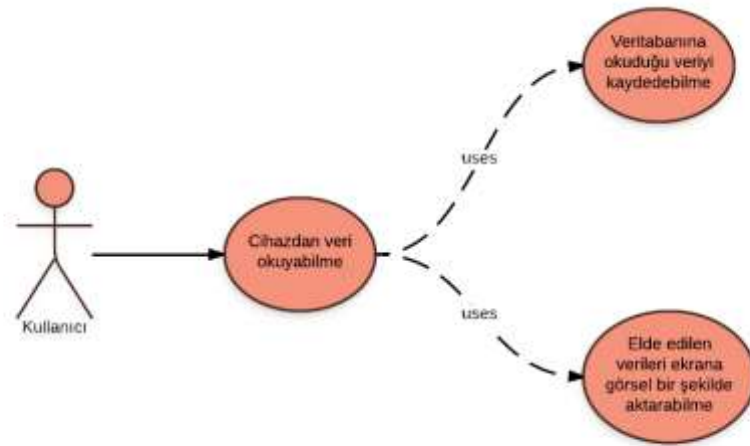
### 5.3. Veritabanı Diyagramı

## WaspMote Database Model

Acceleration		
Unique Id	Object	id
	Date	time
	Text	xValue
	Text	yValue
	Text	zValue

Şekil 5.2. Veritabanı Diyagramı

### 5.4. Use Case Diyagramı



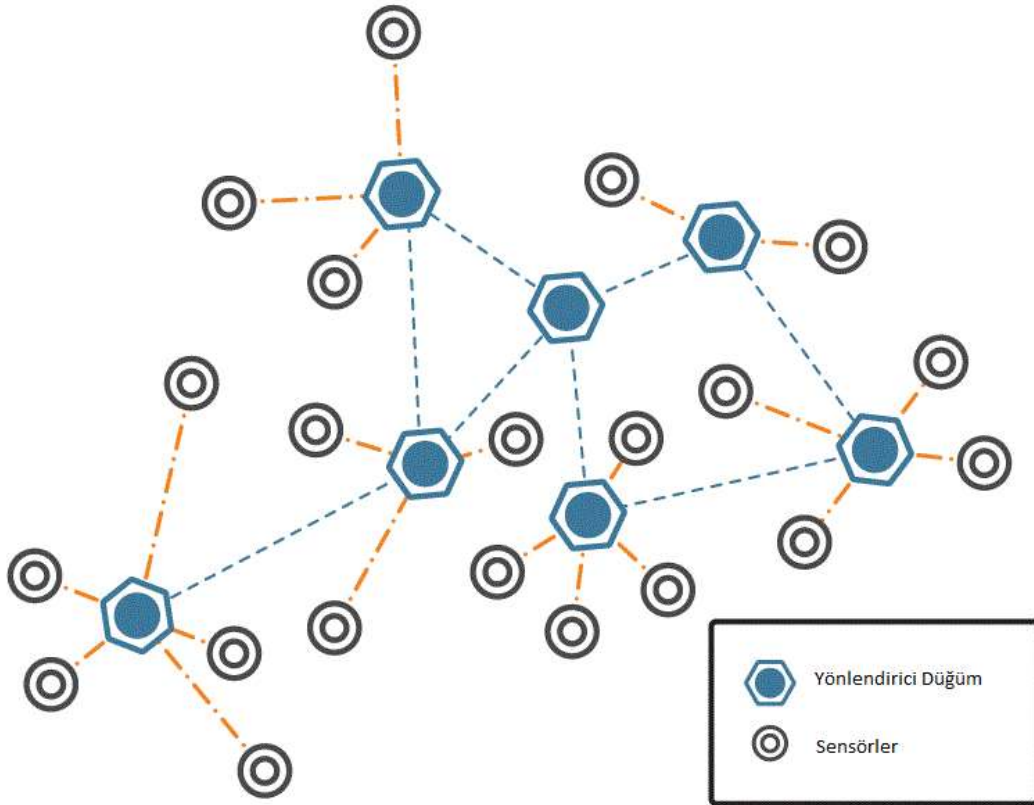
Şekil 5.3. Use Case Diyagramı

Kullanıcı cihazdan verileri okuyabilir, verileri ekranda ayrıntılı bir şekilde görebilir ve bu verileri saklayabilir. Kullanıcının, verileri ekranda göstermesi ve bu verileri saklayabilmesi için cihazdan verilerin okunabilmesi şartı vardır bundan dolayı bu işlemler arasında “uses” etiketi kullanılır.



## BÖLÜM 6. SONUÇLAR VE ÖNERİLER

Gelinen noktada internet artık yalnızca haber ya da bilgi almak, paylaşmak amacıyla kullandığımız bir ağ olarak varlığını sürdürmekten çıkmış, çok daha kritik amaçlara hizmet etme yolunda içinde yaşadığımız çağ ile birlikte eksponansiyel olarak gelişimini sürdürmektedir. Artık nesnelerinde varlık gösterdiği, birbirleriyle bilgi alış verişinde bulunduğu, ürettikleri çıktılarla bizleri ve diğer nesneleri etkileyebildikleri bir ağ yapısına sahibiz. Bu dönemin merkezinde yer alan biz bilgisayar mühendisleri geliştirdiğimiz bu ve benzeri uygulamalarla gündelik ya da endüstriyel sorunlara çözüm yolları üretmek konusunda becerilerimizi sergiledik. Web servis kullanarak uygulamamızın çıktılarına farklı servisler aracılığıyla erişilebilmesini sağladık. Benzer çözümlerle karşılaştırıldığında geliştirdiğimiz uygulamanın aktif olarak kullanılabileceğini gördük.



Şekil 6.1 Nesnelerin İnterneti Ağ Topolojisi

## KAYNAKLAR

- [1] [https://en.wikipedia.org/wiki/Internet\\_of\\_Things](https://en.wikipedia.org/wiki/Internet_of_Things)
- [2] [https://en.wikipedia.org/wiki/Machine\\_to\\_machine](https://en.wikipedia.org/wiki/Machine_to_machine)
- [3] <http://www.cmo.com/articles/2015/4/13/mind-blowing-stats-internet-of-things-iot.html>
- [4] <http://www.statista.com/topics/2637/internet-of-things/>
- [5] <http://tech.co/internet-of-things-shaping-future-2014-11>
- [6] <https://www.accenture.com/interactive-index.aspx?sfvrsn=0>
- [1] <http://www.intel.com/content/www/us/en/internet-of-things/infographics/guide-to-iot.html>
- [8] [http://www.libelium.com/v11-files/documentation/waspmote/waspmote-technical\\_guide\\_eng.pdf](http://www.libelium.com/v11-files/documentation/waspmote/waspmote-technical_guide_eng.pdf)
- [9] <https://www.cooking-hacks.com/blog/waspmote-starter-kit-beginners-guide/>
- [10] <https://www.cooking-hacks.com/documentation/tutorials/waspmote>
- [21] [https://www.libelium.com/v11-files/documentation/waspmote/waspmote-accelerometer-programming\\_guide.pdf](https://www.libelium.com/v11-files/documentation/waspmote/waspmote-accelerometer-programming_guide.pdf)
- [12] <https://en.wikipedia.org/wiki/Sensor>
- [13] <https://nodejs.org/en/>
- [14] <https://www.npmjs.com/package/serialport>
- [15] <https://docs.mongodb.org/getting-started/node/insert/>

## ÖZGEÇMİŞ

İsmet GÜZELGÜN, 10.02.1993 tarihinde Ankara’da doğdu. Üniversite öğrenimini Ankara’da tamamladı. 2012 Yılında Sakarya Üniversitesi Bilgisayar Mühendisliği bölümünü kazanarak mevcut öğrenimini halen bu bölümde sürdürmektedir. Üniversite öğrenimi sırasında çift anadal yapma hakkı kazanıp yine Sakarya Üniversitesi’nin Endüstri Mühendisliği bölümünde halen öğrencisidir. Öğrenimi sırasında ilgi duymaya başladığı gömülü sistemler, ağ programlama, yapay sinir ağları konularında çalışmalar göstermektedir.

Ali Coşkun, 4 Eylül 1992 tarihinde Eminönü, İstanbul’da doğdum. Orta öğretimimi İstanbul Bayrampaşa Tuna Lisesi’nde, Sayısal bölümünde gerçekleştirim. Lise ve öncesinde bilgisayar’a, özellikle yazılım kısmına ilgi duydum. Üniversitede de bölüm tercihim bu ilgimden yana kullanıp Bilgisayar Mühendisliği’ni seçtim ve halihazırda Sakarya Üniversitesi’nde 4. sınıfta okumaktayım. Bunun yanı sıra 6 ay gibi bir süredir Ford Otosan’da Yazılım Uzmanı olarak çalışmaya devam etmekteyim.

## BSM 498 BİTİRME ÇALIŞMASI DEĞERLENDİRME VE SÖZLÜ SINAV TUTANAĞI

KONU : KABLOSUZ ALGILAYICI AĞLAR İLE WEB TABANLI GÖZETLEME  
UYGULAMASI

ÖĞRENCİLER (Öğrenci No/AD/SOYAD): B121210025 İsmet GÜZELGÜN  
G111210070 Ali COŞKUN

Değerlendirme Konusu	İstenenler	Not Aralığı	Not
<b>Yazılı Çalışma</b>			
<b>Çalışma klavuza uygun olarak hazırlanmış mı?</b>	x	0-5	
<b>Teknik Yönden</b>			
<b>Problem tanımı yapılmış mı?</b>	x	0-5	
Geliştirilecek yazılımın/donanımın mimarisini içeren blok şeması (yazılımlar için veri akış şeması (dfd) da olabilir) çizilerek açıklanmış mı?			
Blok şemadaki birimler arasındaki bilgi akışına ait model/gösterim var mı?			
Yazılımın gereksinim listesi oluşturulmuş mu?			
Kullanılan/kullanılması düşünülen araçlar/teknolojiler anlatılmış mı?			
Donanımların programlanması/konfigürasyonu için yazılım gereksinimleri belirtilmiş mi?			
UML ile modelleme yapılmış mı?			
Veritabanları kullanılmış ise kavramsal model çıkarılmış mı? (Varlık ilişki modeli, noSQL kavramsal modelleri v.b.)			
Projeye yönelik iş-zaman çizelgesi çıkarılarak maliyet analizi yapılmış mı?			
Donanım bileşenlerinin maliyet analizi (prototip-adetli seri üretim vb.) çıkarılmış mı?			
Donanım için gerekli enerji analizi (minimum-uyku-aktif-maksimum) yapılmış mı?			
Grup çalışmalarında grup üyelerinin görev tanımları verilmiş mi (iş-zaman çizelgesinde belirtilebilir)?			
Sürüm denetim sistemi (Version Control System; Git, Subversion v.s.) kullanılmış mı?			
Sistemin genel testi için uygulanan metotlar ve iyileştirme süreçlerinin dökümü verilmiş mi?			
Yazılımın sızma testi yapılmış mı?			
Performans testi yapılmış mı?			
Tasarımın uygulamasında ortaya çıkan uyumsuzluklar ve aksaklıklar belirtilerek çözüm yöntemleri tartışılmış mı?			
<b>Yapılan işlerin zorluk derecesi?</b>	x	0-25	
<b>Sözlü Sınav</b>			
<b>Yapılan sunum başarılı mı?</b>	x	0-5	
<b>Soruları yanıtlama yetkinliği?</b>	x	0-20	
<b>Devam Durumu</b>			
<b>Öğrenci dönem içerisindeki raporlarını düzenli olarak hazırladı mı?</b>	x	0-5	
<b>Diğer Maddeler</b>			
<b>Toplam</b>			

DANIŞMAN (JÜRİ ADINA):

DANIŞMAN İMZASI: