KOCAELİ ÜNİVERSİTESİ * FEN BİLİMLERİ ENSTİTÜSÜ

KABLOSUZ ALGILAYICI AĞLAR İÇİN MATLAB İLE KULLANICI ARAYÜZ TASARIMI

YÜKSEK LİSANS TEZİ

ELİF TURHAN

Anabilim Dalı: Elektronik ve Bilgisayar Eğitimi

Danışman: Prof. Dr. Kadir ERKAN

KOCAELİ, 2011

KOCAELİ ÜNİVERSİTESİ * FEN BİLİMLERİ ENSTİTÜSÜ

KABLOSUZ ALGILAYICI AĞLAR İÇİN MATLAB İLE KULLANICI ARAYÜZ TASARIMI

YÜKSEK LİSANS TEZİ

Elif TURHAN

Tezin Enstitüye Verildiği Tarih: 02 HAZİRAN 2011

Tezin Savunulduğu Tarih: 05 TEMMUZ 2011

Tez Danışmanı
Prof.Dr. Kadir ERKAN
(.....)

Üye
Prof.Dr. Fevzi BABA
(.....)

Üye
Doç.Dr. Celal ÇEKEN
(.....)

KOCAELİ, 2011

ÖNSÖZ VE TEŞEKKÜR

Son yıllarda bilginin kaynaklarının giderek artmasıyla verilerin toplanması, analiz edilmesi, saklanması ve görüntülenmesi büyük önem kazanmıştır. Bu uygulamada sensörlerden gelen verilerin MATLAB ortamında görüntülenmesi sağlanmıştır. Uygulamanın MATLAB ortamında yapılması yazılımın ileriye yönelik olarak geliştirilmesini amaçlamaktadır.

Yüksek lisans eğitimim süresince değerli birikimlerini benimle paylaşan, tezimin her aşamasında sorunlarımı dinleyerek, çalışmalarıma yön veren ve yoğun akademik yaşamında değerli zamanını her türlü problemimi çözmeye ayıran tez danışmanım saygıdeğer hocam Prof. Dr. Kadir ERKAN' a; NCS (Network Control System) gurubunun rutin toplantılarına katılarak değerli fikirlerini paylaşan saygıdeğer hocalarım Doç. Dr. Celal ÇEKEN ve Yrd. Doç. Dr. Mehmet YILDIRIM' a; tez çalışmalarına başladığımdan bu yana tüm çalışmalarımda ve tezimin yazımında bilgilerini ve zamanını paylaşan araştırma görevlisi arkadaşlarım Faruk AKTAŞ ve Adem TUNCER' e; maddi ve manevi desteklerini esirgemeyen, hayatımda özel bir yeri olan değerli arkadaşım Veysi DUDUNALI' ya ve yine her konuda desteğini hissettiğim değerli meslektaşım Abdullah KOCATAŞ' a teşekkürlerimi sunarım.

Bugünlere gelmemi sağlayan, maddi manevi desteklerini esirgemeyen çok kıymetli aileme saygı, sevgi ve sonsuz teşekkürler.

İÇİNDEKİLER

ÖNSÖZ i
İÇİNDEKİLER ii
ŞEKİLLER DİZİNİ v
KISALTMALAR vii
ÖZET viii
İNGİLİZCE ÖZET ix
1. GİRİŞ 1
1.1. Tez Çalışmasının Amacı ve Başlatılma Sebepleri 1
1.2. Tez Düzeni 1
2. KABLOSUZ ALGILAYICI AĞLAR 2
2.1. Genel Bilgiler 2
2.2. Algılayıcı Ağda Bulunan Düğümler ve Bileşenleri 10
2.2.1. Düğümdeki bileşenler 11
2.2.1.1 İşlemci (Mikro denetleyici): 11
2.2.1.2. Alıcı-Verici 12
2.2.1.3. Dışsal bellek 13
2.2.1.4. Güç kaynağı 13
2.2.1.5. Algılayıcılar 14
2.2.1.5.1. Pasif, her yöne acık (yönsüz) algılayıcılar 14
2.2.1.5.2. Pasif, dar ışınlı algılayıcılar 14
2.2.1.5.2. Aktif algılayıcılar 14
2.2.1.6. Radyo 15
2.3. Kablosuz Algılayıcı Ağ Mimarisi 17
2.3.1. Kablosuz algılayıcı ağların mobil tasarsız (ad-hoc) ağlara göre avantajları 18
2.3.2. Tasarsız algılayıcı ağları nasıl çalışır? 19
2.3.3. Algılayıcı ağlarda veri birleştirme ve yayma 22
2.3.4. Sıradüzensel algılayıcı ağlar 24
2.4. Kablosuz Algılayıcı Ağlarda Güvenlik 25
2.4.1. Kablosuz algılayıcı ağların güvenliğini tehlikeye atan durumlar 26
2.4.1.1. Düşman saha 26
2.4.1.2. Kaynakların sınırlılığı 26
2.4.1.3. Ağ içinde işlem yapma 27
2.4.1.4. Uygulamaya özel mimari yapı 27
2.4.2. Kablosuz algılayıcı ağların güvenliği için gereksinimler 27
2.4.2.1. Dışarıdan gelen saldırılara karşı dayanıklılık 27
2.4.2.2. İç krizlere karşı direnç 28
2.4.2.3. Güvenliğin gerçekçi seviyesi 28
2.4.2.4. Veri gizliliği 28
2.4.2.5. Veri doğrulama / kimlik denetimi 29
2.4.2.6. Veri bütünlüğü 30
2.4.2.7. Verinin tazeliği 30
2.4.2.8. Kullanılabilirlik 30

2.4.2.9. Hizmet bütünlüğü 31
2.4.3. Saldırılar ve karşı tedbirler 31
2.4.3.1. Gizlilik ve kimlik doğrulama 32
2.4.3.2. Anahtar tespiti ve yönetimi 32
2.4.3.3. Broadcast / Multicast kimlik doğrulama 33
2.5. Örnek Uygulama 33
2.5.1. Kablosuz toprak nemi ölçüm ve kontrol sistemi 33
3. MATLAB GRAFİKSEL KULLANICI ARAYÜZÜ 35
3.1. Giriş 35
3.2. MATLAB GUİ Çalışma Sistemi 35
3.2.1. MATLAB GUİ nasıl çalışır? 35
3.3. Matlab'ta GUI Oluşturma Yöntemleri 36
3.4. MATLAB GUIDE Aracı ile GUI Tasarımı Oluşturma 36
3.5. GUI Çalışma Yüzeyi 39
3.5.1. Bileşenler çalışma alanına ekleme 41
3.5.2. Çalışma alanının boyutunu değiştirmek 41
3.5.3. Nesneleri hizalamak 42
3.5.4. Nesnelere yazı ekleme ve özelliklerini değiştirme 43
3.5.5. GUI tasarımını kaydetme ve çalıştırma 45
3.5.6. GUI ara yüzünün programlanması 46
3.6. Programlama Yoluyla Nesnelerin Eklenmesi 49
3.7. GUIDE Aracının İncelenmesi 53
3.7.1. Layout editor 55
3.7.2. Figure resize tab 55
3.7.3. Menu editor 55
3.7.4. Align objects 55
3.7.5. Tab order editor 55
3.7.6. Property inspector 56
3.7.7. Object browser 56
3.7.8. Run 56
3.7.9. M-File editor 56
3.7.10. GUIDE tercihleri 56
3.8. GUI Nesnelerinin Açıklanması 56
3.8.1. Push button 57
3.8.2. Slider 57
3.8.3. Radio button 57
3.8.4. Check box 57
3.8.5. Edit text 57
3.8.6. Static text 57
3.8.7. Listbox 58
3.8.8. Pop-up menu 58
3.8.9. Toggle button 58
3.8.10. Table 58
3.8.11. Axes 58
3.8.12. Panel 58
3.8.13. Button group 59
3.8.14. ActiveX control 59
3.9. GUI Nesnelerinin Programlanması 59
3.9.1. Push button ve static text uygulaması 59

3.9.2. Toggle button 60
3.9.3. Radio button 61
3.9.4. Checkbox 62
3.9.5. Listbox 63
3.9.6. Pop-up menu 64
3.9.7. Slider 64
3.9.8. Axes 65
3.10. GUI Nesne Özellikleri Penceresi 66
3.11. Nesne Hiyerarşisinin Gösterilmesi 67
3.12. GUI Uygulamalarında Callback' ler Arasında Ortak Veri Geçişini Sağlayan Yollar 68
3.12.1. Handles yapı değişkeni kullanılarak global kullanımı 68
3.12.2. Global değişken tanımlama deyimi 69
3.13. GUI Uygulamalarında Temizleme Komutları 69
3.14. GUI Uygulamalarında Kullanılan Standart Handle Değişkenleri 70
4. UYGULAMADA KULLANILAN KABLOSUZ ALGILAYICI AĞLAR 71
4.1. Giriş 71
4.2. Algılayıcı Düğümler 71
4.3. Ortamdaki Verilerin Toplanması ve Bilgisayara Aktarılması 72
4.3.1. MIB520 Erişim noktası yüklenmesi 73
4.3.2. Xserve 74
4.3.3. XserveTerm 76
4.3.4. XserveTerm' in çalıştırması 77
4.3.4.1. Uzaktan sistem kontrol uygulaması 78
5. KABLOSUZ ALGILAYICI AĞLAR İÇİN MATLAB İLE KULLANICI ARAYÜZ TASARIMI UYGULAMASI 79
5.1. Giriş 79
5.2. Uygulamaya Genel Bakış 79
5.2.1. Nodes paneli 80
5.2.2. Nodes panelinde kullanılan bileşenler 80
5.3. Uygulamanın çalışması 81
5.3.1. Data paneli 81
5.3.2. Command paneli 82
5.3.3. Charts paneli 84
5.3.4. Histogram paneli 86
5.3.5. Scatterplot paneli 87
6. SONUÇLAR VE ÖNERİLER 89
KAYNAKLAR 90
ÖZGEÇMİŞ 92

ŞEKİLLER DİZİNİ

Şekil 2.1: Kablosuz algılayıcı ağ 3
Şekil 2.2: Kablosuz algılayıcı ağların kullanım alanları 5
Şekil 2.3: Kablosuz algılayıcı ağlar ile çevre ve doğa takibi uygulamaları 6
Şekil 2.4: Askeri uygulamalar 6
Şekil 2.5: Tıbbi uygulamalar 7
Şekil 2.6: Akıllı ev ve mekan uygulamaları 8
Şekil 2.7: Çeşitli firmalara ait sensör düğümleri 11
Şekil 2.8: Bir mikro algılayıcı düğümüne ait sistem mimarisi 11
Şekil 2.9: Genel özellikler 16
Şekil 2.10: İşlemci 16
Şekil 2.11: Giriş/Çıkış Birimleri 17
Şekil 2.12: Veri saklama 17
Şekil 2.13: Kablosuz Algılayıcı Ağların İletişim Mimarisi 18
Şekil 2.14: Algılayıcı ağ mimarisi 20
Şekil 2.15: Veri toplayan bir algılayıcı ağ 20
Şekil 2.16: Algılayıcı sıradüzensel ağ 24
Şekil 2.17: Sistemin Grafiksel Gösterimi 33
Şekil 2.18: Sistemle İlgili Diğer Gösterimler 34
Şekil 3.1: Guide penceresi 37
Şekil 3.2: GUI with UIcontrols 37
Şekil 3.3: GUI with axes and menu 38
Şekil 3.4: Modal question dialog 38
Şekil 3.5: GUIDE Quick Start 39
Şekil 3.6: GUI çalışma yüzeyi 40
Şekil 3.7: Figür penceresi 40
Şekil 3.8: Bileşenlerin çalışma alanına eklenmesi 41
Şekil 3.9: Çalışma alanının boyutunu değiştirme 42
Şekil 3.10: Nesneleri hizalama penceresi 43
Şekil 3.11: Nesnelere Yazı Ekleme 43
Şekil 3.12: Nesne özellikleri penceresi 44
Şekil 3.13: Nesne özellikleri değiştirme 44
Şekil 3.14: GUI tasarım kaydetme penceresi 45
Şekil 3.15: GUI Uygulamasının Çalıştırılması 46
Şekil 3.16: GUI Arayüz Programlama Penceresi 46
Şekil 3.17: Figur yüzeyindeki nesnelerin callback' lerine ulaşılması 47
Şekil 3.18: "İlk uygulama" nın çalışması 48
Şekil 3.19: Programlama Yoluyla Nesnelerin Eklenmesi 49
Şekil 3.20: Guide penceresinin yapısı 54
Şekil 3.21: Push button ve static text uygulaması 59
Şekil 3.22: Radiobutton uygulaması 61
Şekil 3.23: Checkbox uygulaması 62
Şekil 3.24: Listbox uygulaması 63

Şekil 3.25: Pop-up menü uygulaması 64
Şekil 3.26: Slider uygulaması.....65
Şekil 3.27: Axes nesnesiyle yapılan uygulama 66
Şekil 3.28: Nesne özellikleri penceresi.....67
Şekil 3.29: Object browser penceresi.....68
Şekil 4.1: MicaZ algılayıcı düğüm ve MIB520 programlama bordu (erişim noktası) 71
Şekil 4.2: Sistemin çalışması.....72
Şekil 4.3: Aygıt yöneticisinde portların listelenmesi.....73
Şekil 4.4: Verilerin raw formatında görüntülenmesi 74
Şekil 4.5: Verilerin parsed formatında görüntülenmesi.....75
Şekil 4.6: Converted formatlı verilerin görünümü 75
Şekil 4.7: Xcommand ile yapılabilecek işlemler.....77
Şekil 4.8: xmlport ve format ayarlarının yapılması.....77
Şekil 4.9: Portun seçilmesi.....77
Şekil 4.10: Cygwin açıldığında çalıştırılacak kodlar.....78

KISALTMALAR

MATLAB	: Matrix Laboratory (Matris Laboratuarı)
WSN	: Wireless Sensor Network (Kablosuz Algılayıcı Ağlar – KAA)
GUI	: Grafics User Interface (Grafik Kullanıcı Ara Yüzü)
GUIDE	: Grafics User Interface Design Environment (Grafiksel Kullanıcı Ara Yüzü Tasarım Ortamı)
XML	: Extensible Markup Language (Genişleyebilir İşaretleme Dili)

KABLOSUZ ALGILAYICI AĞLAR İÇİN MATLAB İLE KULLANICI ARAYÜZ TASARIMI

Elif TURHAN

Anahtar Kelimeler: Kablosuz Algılayıcı Ağlar, MATLAB GUI, Ortam İzleme

Özet: Mikro elektronik ve kablosuz haberleşme teknolojilerindeki süregelen gelişmeler küçük boyutlu, hareketli, düşük maliyetli, düşük enerji gereksinimli ve çok fonksiyonu algılayıcı düğümlerin yaygın olarak kullanılmasına olanak sağlamıştır. Kablosuz algılayıcı ağlar genel olarak çevresel, askeri, sağlık, endüstriyel gibi birçok alanda özellikle ortam izleme amaçlı kullanılmaktadır.

Çevreden alınan verilerin görüntülenmesi de önemli bir ihtiyaçtır. Bunu sağlayan farklı uygulamalar mevcuttur. Bu tez çalışmasında alınan verilerin görüntülenmesi işlemi MATLAB GUI tabanlı olarak gerçekleştirilmiştir.

MATLAB gelişmiş özellikleri olan, birçok çevre tarafından kullanılan, farklı alanlarda farklı eklentiler yapılmaya elverişli, yazılım geliştirmeye açık bir dildir. Bu tez çalışmasının MATLAB GUI tabanlı yapılması ortamdan alınan verilerin görüntülenmesinin yanı sıra, gelen verilerin değerlendirilip kontrolünün yapılarak farklı uygulamalar geliştirilmesine de olanak sağlamaktadır.

USER INTERFACE DESIGN WITH MATLAB FOR WIRELESS SENSOR NETWORKS

Elif TURHAN

Keywords: Wireless Sensor Networks, MATLAB-GUI, Environment Monitoring

Abstract: The continuing developments, in micro-electronics and wireless communication technologies enable widely using of sensor nodes which are small-sized, portable, low-cost, low energy needs and multi-function. Wireless Sensor Networks are generally used for especially environment monitoring in many areas such as environmental, military, healt, industrial.

The display of the data received from the environment is also an critical requirement. There are different applications that do this. In this thesis, process of displaying the data received was carried out as MATLAB-GUI based.

MATLAB is a language that allows software development, which has advanced features, has been used in different areas and is suitable for different plug-in in different areas. Implementing of MATLAB-GUI based in this thesis also allows the development of different applications by controling and evaluating the data received as well as monitoring of data from the environment.

1. GİRİŞ

1.1. Tez Çalışmasının Amacı ve Başlatılma Sebepleri

Bu tez çalışması, KAA' ların kullanıldığı uygulama ortamlarında sensörlerden gelen verileri görüntüleyen C tabanlı MoteView programından esinlenerek yapılmıştır.

Tez çalışmasının ana hedefi, KAA' larda gelen verileri görüntüleme işleminin MATLAB GUI tabanlı gerçekleştirilerek geliştirilmeye/uyarlanmaya açık bir yapıya kavuşmasını, çok çeşitli uygulama alanlarına hitap etmesini ve farklı amaçlar için yeni bir yaklaşım geliştirilmesine katkıda bulunmasını sağlamaktır.

1.2. Tez Düzeni

Tez çalışmaları, beş ana bölümde sunulmaktadır;

Bölüm 2'de KAA'ların gelişimi hakkında genel bilgiler verilmekte, olumlu/olumsuz yönleri, uygulama alanları sunulmaktadır.

Bölüm 3'te MATLAB GUI genel olarak anlatılarak çeşitli uygulama örnekleri açıklanmaktadır.

Bölüm 4'te KAA' lar ve sensörler kullanılarak alınan verilerin MATLAB ortamına gelene kadar geçtiği aşamalar anlatılmaktadır.

Yapılan tez çalışmasında uygulamaların sonuçları ve katkıları Bölüm 5'te anlatılmaktadır.

2. KABLOSUZ ALGILAYICI AĞLAR

2.1. Genel Bilgiler

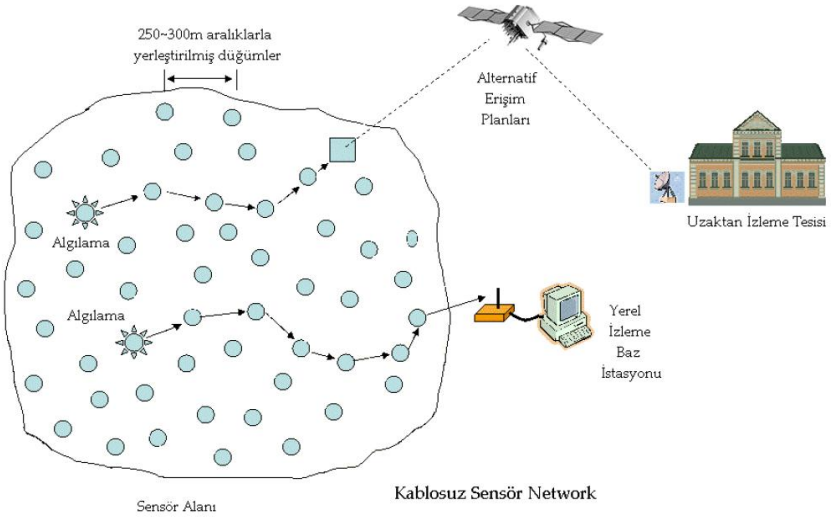
Donanım ve kablosuz sistemlerdeki gelişmeler düşük maliyetli, düşük güç tüketimli, çok işlevli minyatür algılama aygıtlarının üretilmesine olanak sağlamıştır. Bu aygıtlardan yüzlercesi, binlercesi yardımıyla ad-hoc ağlar oluşturulabilmektedir. Örneğin; bu aygıtlar geniş bir coğrafyaya dağıtılarak kablosuz, ad-hoc bir ağ oluşturulmaktadır. Bu dağıtılan ve ağı oluşturan sensörler işbirliği yaparak bir algılama ağ sistemini oluşturmaktadır. Bir sensör ağı bilgiye her an, her yerden kolayca erişilmesini sağlar. Bu işlevi veriyi toplayarak, işleyerek, çözümleyerek ve yayarak yerine getirir. Böylece ağ, etkin bir şekilde zeki bir ortam oluşmasında rol oynamış olur.

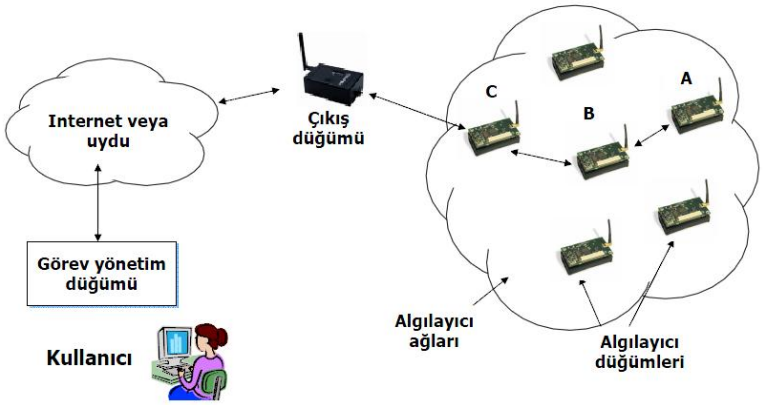
Kablosuz algılayıcı ağlar; geniş bir yelpazede, değişik uygulama alanları için devrimsel algılama özelliği yetenekleri sunmaktadır. Bunun nedeni algılayıcı ağlarının;

- Güvenilirlik
- Doğruluk
- Esneklik
- Maliyet verimliliği
- Kurulum kolaylığı

Özelliklerine sahip olmasıdır.

Tipik bir Kablosuz Algılayıcı Ağ (Wireless Sensör Network WSN) kablosuz bir ortam aracılığı ile birbirine bağlanmış ve birbirleriyle bilgi alışverişi yapan yüzlerce hatta binlerce algılayıcı (sensör) düğümünden oluşur. Şekil 2.1' de kablosuz algılayıcı ağ sistemi görülmektedir. Bu düğümler kendi ağlarını kendileri organize ederler, önceden programlanmış bir ağ topolojisi söz konusu değildir. Pil ömrüne bağlı olan kısıtlamalar yüzünden, sensör düğümleri çok büyük bir zamanı düşük güç tüketimi ile "uyku" modunda geçirirler ya da düğüm verisini işlerler.





Algılayıcı Ağları

Şekil 2.1: Kablosuz algılayıcı ağ

Kablosuz Algılayıcı Ağların (WSN) sağladığı yararlardan ya da artı özelliklerden bazıları aşağıda kısaca açıklanmıştır.

Her zaman, her yerde: Mevcut makrosensör düğümlerinin kapsamı, maliyet kıstasları ve kurulum (plana göre yerleşim) sebepleriyle belirli fiziksel alanlarda dar olarak sınırlıdır. Buna zıt bir şekilde WSN' ler insan bakımına gereksinim duymayan fiziksel olarak ayrılmış pek çok düğüm içerebilir. Düğüm bazında bakıldığında tek bir düğümün kapsamı küçük de olsa, yoğun olarak dağıtılmış düğümler eş zamanlı ve iş birliği prensipleriyle çalışabilir, böylece tüm ağın kapsamı genişletilmiş olur.

Ayrıca algılayıcı düğümler yaşam tehlikesinin olduğu alanlara bırakılabilir ve dört mevsim işlem yapabilir, bu yüzden bu düğümler algılama görevlerini her an yürütebilirler.

Hataya karşı daha fazla tolerans: Bu kazanım WS düğümlerinin yoğun biçimde yerleştirilmesi sonucu sağlanmıştır. Aynı alan içerisinde komşu düğümlerden birbiriyle ilişkili veri alınması sonucunda sistemin hatayı tolere etme şansı, tek başına bulunan bir makrosensöre kıyasla çok daha büyüktür. Eğer bir makrosensör düğümü hata verir ya da işlemi durur ise; sistem, fonksiyonunu algılayıcı düğümün bulunduğu alanda tamamen yitirir.

Bu durumun tam tersi olarak WSN' lerde eğer mikrosensör düğümlerinin küçük bir kısmı hata verirse, WSN kabul edilebilir derecede bilgi üretmeye devam edebilir, çünkü çıkarılan veri gereğinden fazladır. Bundan başka alternatif haberleşme yolları (route), herhangi bir yönlendirme hatası olduğu takdirde kullanılabilir.
Geliştirilmiş doğruluk oranı: Tek başına bir makrosensör düğümü tek bir mikrosensör düğümünden daha doğru bir ölçüm yapsa bile, çok sayıda mikro düğümün topladığı veririn tek parça haline getirilmesi ile oluşan veri gerçekten dünyanın gerçekliğinden daha fazlasını yansıtabilir. Buna ek olarak bu veri, uygun algoritmalar eşliğinde işlenir ve ilişkilendirilir ve/veya kümelenirse genel sinyal geliştirilebilir ve ilişkisiz parazitin bir kısmı temizlenebilir.

Düşük Maliyet: WSN' lerin makrosensörlü sistemdeki eşlerinden (karşıtlarından) daha düşük maliyetli olması beklenmektedir. Bu beklentinin sebepleri; küçültülmüş boyutları, düşük fiyatları ve bunlarla birlikte yerleşim/ kurulum aşamasının kolaylığı olarak gösterilebilir.

WSN' ler ;
• Sıcaklık
• Nem
• Işık
• Basınç
• Nesne hareketleri

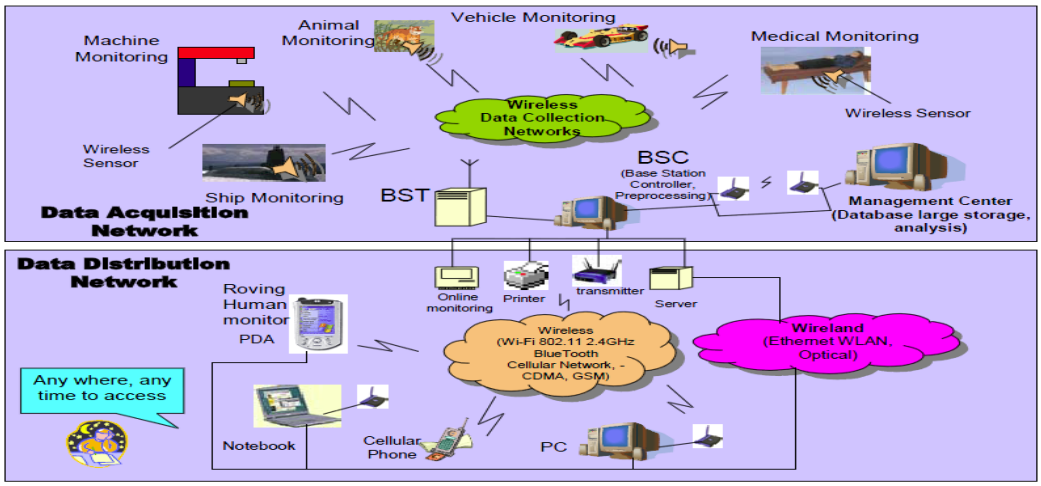
• Toprak bileşimi

• Gürültü seviyesi

• Bir nesnenin mevcudiyeti,

• Belirli bir nesnenin; ağırlık, boyut, hareket hızı, yönü, son konumu gibi fiziksel durumları izleyebilirler (monitoring).

WSN' lerin çok farklı alanlarda uygulamaları bulunmaktadır. Şekil 2.2' de bu uygulamalarla ilgili gösterim bulunmaktadır.



Şekil 2.2: Kablosuz algılayıcı ağların kullanım alanları

• Çevre ve Doğa Takibi

• Envanter İzleme

• Tıbbi Takip

• Askeri Uygulamalar

• Endüstriyel Takip

• Sismik Algılama

• Akıllı Mekânlar

• Trafik Kontrol

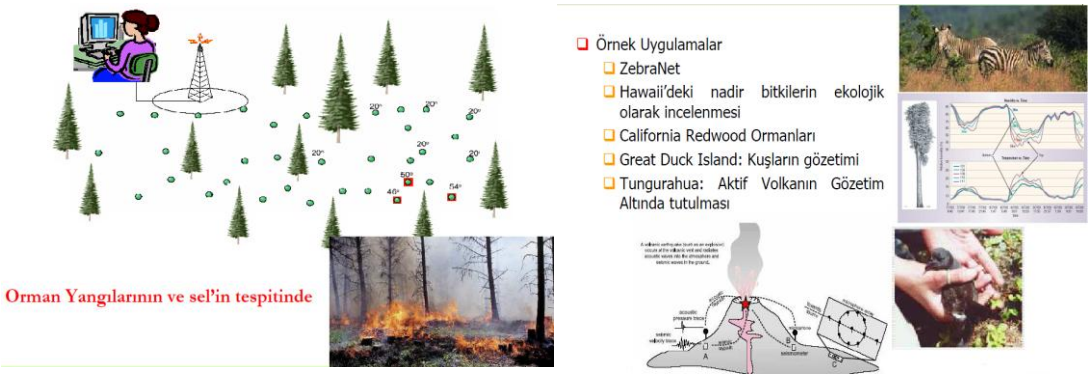
• Akustik Algılama

Bu uygulamaların bir kısmının detayları aşağıda verilmiştir.

Çevre ve Doğa Takibi: İzlenmesi gereken niceliklerin geniş alanlara yayılması nedeniyle bu alanda algılayıcı (sensör) ağlara ihtiyaç duyulmaktadır. Bitki

örtülerinin, iklimsel değişikliklere ve hastalıklara tepkilerinin ölçülmesinde, ayrıca ses ve görüntü sensörleri de, kuş ve diğer türlerin nüfus takibinde, tanımlanmasında ve ölçülmesinde kullanılmıştır. Görüntü sensörleri uzay tabanlı iken, radarlar uçaklara yerleştirilmiş ve çevresel algılayıcılar da genelde zemine yerleştirilmiştir.

Algılayıcılar arasındaki iletişim ağının hızı farklılık göstermektedir. Şekil 2.3' te Çevre ve Doğa Takibi ile ilgili çeşitli uygulamalar görülmektedir.



Şekil 2.3: Kablosuz algılayıcı ağlar ile çevre ve doğa takibi uygulamaları

Askeri Uygulamalar: Algılayıcı ağlar güvenlik altyapısı nedeniyle terörizme karşı uygulamalarda da oldukça kullanışlıdır. Nükleer santraller ve iletişim merkezleri gibi önemli yerlerin olası risklere karşı korunması gerekmektedir. Bu tür birimlere video, ses ve diğer algılayıcı ağlarının yerleştirilmesi sonucu olası tehditlerin erken tespiti mümkün olacaktır. Farklı algılayıcılardan gelen verilerin birleştirilmesi ile geliştirilmiş kapsama alanı ve tespit sağlanabilmektedir. Ayrıca bu sayede hatalı alarm oranı düşürülebilmektedir. Algılayıcı ağlar biyolojik, kimyasal ve nükleer saldırıların tespiti amacı için de kullanılabilmektedir. Şekil 2.4' te askeri alandaki ile ilgili çeşitli uygulamalar görülmektedir.

Şekil 2.4: Askeri uygulamalar

Tıbbi Takip: Algılayıcı ağlar zamanında ve etkin sağlık hizmetlerinin sağlanması ile insanlık için daha sağlıklı bir çevrenin oluşturulmasında oldukça yardımcıdır. Şekil 2.5' te detaylı bir gösterim bulunmaktadır.

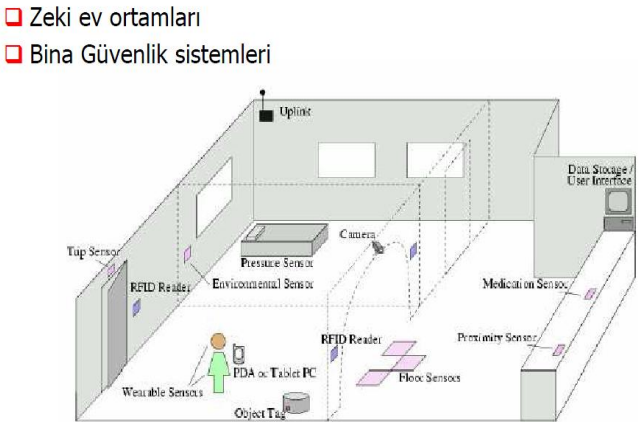
Şekil 2.5: Tıbbi uygulamalar

Endüstride Takip: Performansı arttırmak ve maliyeti düşürmek amacı ile üreticiler algı kavramı ile her zaman ilgilenmişlerdir. WSN' lerin uzaktan kontrol edilebilmeleri, kurulum ücretlerini düşük tutarken fabrikalarda güvenliği kontrole yardımcı olmaktadır. Endüstriyel izlemenin amacı çok noktalı veya matris algılamayı mümkün kılmaktır. Bu uygulamalarda yüzlerce veya binlerce algılayıcı düğümden gelen veriler veri tabanına sürekli bilgi yollarlar. Böylece gerçek zamanlı bilgiler geniş veya dar ölçekli olarak farklı şekillerde sorgulanabilmektedir.

Trafik Kontrolü: Algılayıcı ağlar trafik kontrolü ve takibi için uzun yıllardır kullanılmaktadır. WSN' ler; kaza, araba arızası, yol durumu ve sinyalizasyon onarımını takip etmek için oldukça fonksiyoneldir. Benzer şekilde trafik sıkışıklıklarını tespit edip kullanıcıları uyarmak açısından da verimlidir. Bunların yanı sıra, trafiğin akışını değiştirip taşımacılık kapasitesini arttırabilmek için de uygundur. Ayrıca park alanlarının yönetimi ve yasal olmayan sürüş ve park davranışlarını tespit için de kullanışlıdır. Bu algılayıcılar ve bunları birbirine bağlayan iletişim ağlarının pahalı olması nedeniyle trafik takibi ender kritik noktalar için bu şekilde yapılmaktadır. Ucuz WSN' lerin trafik yönetimi, takibi ve kontrolü için kullanımı bu alanda değişiklikler yaratacaktır. Daha ekonomik algılayıcı düğümlerin yollara entegrasyonu sonucu yollardaki araçlar sayılabilecek veya hızları

tespit edilebilecektir. Ya da araçların kendi algılayıcılarına sahip olmaları durumunda çok daha ileri bir sistem kullanılabilecek; yan yana geçen araçlar birbirlerinden yol, trafik, hız bilgilerini alabileceklerdir. Hatta zemin algılayıcıları ile irtibata geçebilecek ve böylece trafik sıkışıklığı, ya da yol durumu bilgileri önceden elde edilebilecektir.

Akıllı Evler ve Mekanlar: WSN' ler tüm insanlık için daha rahat ve akıllı yaşam alanlarının oluşturulmasında rol alabilir. Bu tür uygulamalara örnek verirsek; uzaktan ölçüm: WSN' ler gaz, elektrik, oda sıcaklığı gibi verileri kablosuz ağ aracılığı ile istenen noktaya iletebilir. Ya da parkmetrenin süresinin dolmak üzere olduğunu araç sahibine iletebilir. Son zamanlarda teknolojideki gelişmeler sonrasında, çeşitli kablosuz algılayıcı düğümlerin kişisel mobilya ya da araçlara iliştirilmesi mümkün kılınmıştır, bu sayede otonom bir ağ oluşturulabilir. Örnek olarak, akıllı bir buzdolabı ailenin doktordan alınan diyet programına göre buzdolabının envanterini tutup, alışveriş listesini tutan kişisel dijital asistana alınacaklar listesini gönderebilir. Şekil 2.6' da akıllı ev ortamlarıyla ilgili örnek bir uygulama görülmektedir.



Şekil 2.6: Akıllı ev ve mekan uygulamaları

Algılayıcı ağ teknolojilerini gerçekleştirmede; donanım tasarımı, iletişim protokolleri ve uygulama tasarımlamada zorluklar çıkmaktadır. Algılayıcı ağın yaşam ömrünü uzatmak ve zeki veri toplama sistemleri kurmak bu zorluklardan ikisidir. Diğer zorluklar şu şekilde listelenebilir:

- Algılayıcı ağlarının topolojisi çok sık değişir.
- Algılayıcı düğümler noktadan noktaya iletişime dayanan ağlarda yayım iletişim paradigmasını kullanır.
- Çok kısıtlı güç, hesaplama yeteneği ve hafızaya sahiptir.
- Bozulmaya yatkındır.
- Çok fazla yükten dolayı genel kimlik (ID) sahibi olmayabilir.
- Çok fazla sayılarda kurulur, bu nedenle kalabalıktan kaynaklanan tıkanma ve çarpışmalar olabilir. Önlemek için birbirine yakın algılayıcı düğümler eşzamanlı iletişim yapmamalıdır.
- Ad-hoc yerleştirilmiş sistemin, sonuç dağıtım ve düğümlerin bağlantılılığını (connectivity) tanımlaması ve sağlaması gerekir.
- Devingen ortam durumları, sistemin zamanla bağlantılılık ve sistem uyarımını uyarlamasını gerekli kılar.

Algılayıcı ağları oluştururken de aşağıdaki gereksinimleri göz önünde bulundurmak gerekir:

Fazla sayıda algılayıcı düğüm: Ucuz, küçük boyutlu algılayıcı düğümler kullanılarak algılayıcı ağlar binlerce algılayıcı düğümü içerebilir. Ölçeklenebilirlik ve bu yüksek sayıdaki düğümü yönetmek önemli bir sorundur.

Düşük enerji kullanımı: Düğümün ömrü, üzerindeki pilin ömrüyle belirleniyor, böylece minimal düzeyde enerji tüketilerek pilin en verimli şekilde kullanılması gerekiyor.

Düşük belleğin verimli kullanımı: Algılayıcı ağları kurulurken yönlendirme tablosu veri yineleme (data replication), güvenlik ve benzeri konular algılayıcı düğümdeki düşük belleğe sığacak şekilde değerlendiriliyor.

Veri toplama: Çok sayıda algılama düğümü ağı bilgiyle şişirebilir. Bu problemi çözmek için, bazı düğümler (küme başları gibi) veriyi toparlayarak, bazı hesaplamalar yaparak (ortalama, toplam, en yüksek, vb.) elde ettiği özetleri yayınlayabilir.

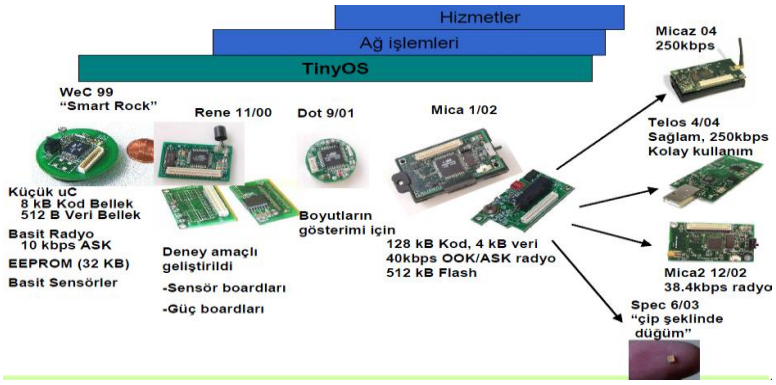
Ağ öz örgütlenmesi: Çok sayıda düğüm ve bu düğümlerin erişimi zor (vahşi-hostile) ortamlarda yerleştirilmesi gibi durumlarda, ağın kendini örgütleyebilmesi olmazsa olmazdır. Ağın yaşamı süresince düğümler çökebilir, yeni düğümler ağa katılabilir. Bu yüzden, ağ belirli aralıklarla kendini yeniden yapılandırabilmelidir.

İşbirlikçi sinyal işleme: Bu ağları mobil ad-hoc ağlardan ayıran önemli bir etken, ağların amacının sadece iletişim değil, ilgi duyulan bir olayın belirlenmesi/tahmininin yapılmasıdır.

Sorgulama yeteneği: Sensör ağında sorgulamanın ne şekilde yapılabildiği önemlidir.
Düşük Maliyet: Ağlarda binlerce düğüm kullanılacağı için sensör düğümlerinin maliyetinin düşük olması gereklidir.

2.2. Algılayıcı Ağda Bulunan Düğümler ve Bileşenleri

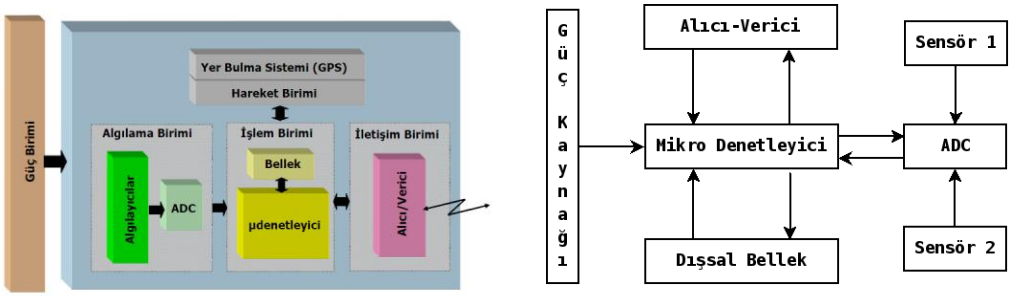
Algılayıcı düğüm, kablosuz algılayıcı ağlarda kullanılan ve hesaplama, algısal bilgi toplama ve ağdaki diğer bağlantılı düğümlerle haberleşme yeteneklerine sahip düğümlerdir. Tipik bir algılayıcı düğüm mimarisi görülebilir. Algılayıcı düğümlerinin geliştirilmesinin başlangıcı 1998 yılındaki Smartdust projesine dayanır. Bu projenin amaçlarından biri kübik milimetre içerisinde otonom algılama ve iletişim yaratmaktır. Bu proje erken bitmesine rağmen, bir kaç araştırma projesinin doğmasına neden olmuştur. Bu projeler Berkeley NEST1 ve CENS2 projeleridir. Bu projelerde yer alan araştırmacılar algılayıcı düğüm için mote terimini kullanmaktadır. Şekil 2.7' de değişik firmaların algılayıcıı düğümleri ve özellikleri görülmektedir.



Şekil 2.7: Çeşitli firmalara ait sensör düğümleri

2.2.1. Düğümdeki bileşenler

Bu düğümler genelde 6 tip bileşenden oluşur. Bunlar; İşlem Birimi, Bellek Ünitesi, Güç Kaynağı, Sensör ve/ veya erişim düzeneği (Algılama Birimi) ve haberleşme alt sistemi (radyo). Standart işlemcilerin DSP (Sayısal İşaret İşleme) ile takviye edildiği, yardımcı işlemciler ve ASIC üniteleri ile düşük enerji seviyelerinde çalışabildiği bu sayede yeterli yeteneklere sahip olduğu görünmektedir. Erişim düzenekleri (actuators) çağdaşlık bakımından henüz SN düğümlerinde kullanılabilecek seviyede değildir. Bu sebeple, dikkatler diğer beş bileşen üzerindedir. Şekil 2.8' de bir Mikro Sensör Düğümünün Sistem Mimarisinin iki farklı gösterimi karakterize edilmiştir.



Şekil 2.8: Bir mikro algılayıcı düğümüne ait sistem mimarisi

2.2.1.1 İşlemci (Mikro denetleyici):

Mikro denetleyici görevleri yapar, veriyi işler ve algılayıcı düğüm içerisindeki diğer bileşenlerin işlevselliğini denetler. Denetleyici olarak kullanılabilecek diğer alternatifler arasında şunlar sayılabilir: Genel amaçlı masaüstü mikroişlemci, sayısal sinyal işlemciler (SSİ), alanı programlanabilir geçit dizileri (FPGA) ve uygulamaya

özgü tümleşik devreler. Mikro denetleyiciler algılayıcı düğüm için en uygun seçimdir. Her seçeneğin kendine özgü avantaj ve dezavantajları vardır. Diğer aygıtlara bağlanmadaki esneklikleri, programlanabilir olması, bu aygıtlar uyuma moduna girebildiği ve sadece denetleyicinin bir kısmının etkin olması nedeniyle düşük enerji tüketimi nedeniyle mikro denetleyiciler gömülü sistemler için en uygun secimdir. Genel amaçlı mikroişlemciler mikro denetleyicilerden daha fazla enerji harcamaktadır. Sayısal sinyal işlemciler (SSİ) geniş bant kablosuz iletişim için uygundur. Kablosuz algılayıcı ağlarda, kablosuz iletişim yalın olmalıdır. Modülasyonu işlemek daha kolay ve asli olan veri algılanması sinyal işleme görevleri daha az karmaşık olmalıdır. Bu yüzden SSİ' lerin avantajlarının kablosuz sensör ağları açısından fazla bir önemi kalmamaktadır. FPGA' lar gereksinimlere göre tekrar programlanabilir ve yapılandırılabilirler. Ancak bu zaman ve enerji tüketimine yol acar, bu nedenle FPGA' lar tavsiye edilmemektedir. Uygulamaya özgü tümleşik devreler belirli bir uygulama için tasarlanmış, uzmanlaşmış işlemcilerdir. ASIC' ler işlevselliği donanım olarak sunarken, mikro denetleyiciler yazılımsal olarak sağlarlar.

2.2.1.2. Alıcı-Verici

Algılayıcı düğümleri ISM bandını kullanır. Bu bant sayesinde geniş dalga kuşağında ve global elverişlilikte özgür radyo yayını sağlanmış olur. Kablosuz iletim ortamlarında tercihler radyo frekansı,optik iletişim (lazer) ve kızılötesidir. Lazer daha az enerji gerektirir, ancak iletişim için görüş alanı gerektirir ve atmosferik koşullara duyarlıdır. Kızılötesi lazer gibidir, anten gerektirmez ancak yayım kapasitesi olarak sınırlıdır. Radyo frekansı (RF) tabanlı iletişim çoğu WSN uygulaması için uygun olan iletişim şeklidir. WSN' ler 433 MHz ve 2.4 GHz arasındaki iletişim frekanslarını kullanırlar. Alıcı ve vericinin işlevselliği alıcı-verici adı verilen tek bir aygıt içerisinde birleştirilmiştir. Alıcı-vericiler tekil belirteçten yoksundur. İşlemsel durumlar İletme(Transmit), Alma (Receive), Boş (Idle) ve Uyku(Sleep)'dur. Bugünkü nesil radyolar bu işlemi otomatik olarak gerçekleştiren gömülü durum makinelerine sahiptir. Alıcı-vericideki radyolar yukarıda belirtilen dört farklı modda çalışmaktadır. Boş modda çalışan radyoların güç tüketimi neredeyse Alma modundaki enerji tüketimine eşittir. Bu yüzden alma veya iletme işlemi yapmayan

radyoları boş moda almak yerine kapatmak en iyi çözümdür. Ayrıca paket iletimi için Uyku modundan İletme moduna geçerken önemli miktarda enerji tüketimi olmaktadır.

2.2.1.3. Dışsal bellek

Enerji bakış açısından yaklaşıldığında, en uygun bellek çeşitleri mikro denetleyici çipi üzerindeki bellek ve FLASH belleklerdir. Çip dışı RAM' ler seyrek veya hiç kullanılmamaktadır. FLASH bellekler maliyeti ve depolama kapasitesi nedeniyle kullanılmaktadır. Bellek gereksinimleri yüksek oranda uygulama bağımlıdır. Depolamanın türüne göre iki farklı bellek kategorisinden bahsedilebilir: Uygulamayla ilgili veya kişisel bilgileri saklamak için kullanılan kullanıcı belleği, aygıtın programlanması için kullanılan program belleği. Bu bellek ayrıca eğer varsa aygıtın tanımlayıcı verisini içerebilir.

2.2.1.4. Güç kaynağı

Algılayıcı düğümdeki enerji tüketimi algılama, iletişim ve veri işleme nedeniyle olmaktadır. Algılayıcı düğümde veri iletişimi için daha fazla enerji gerekmektedir. Algılama ve veri işleme için enerji tüketimi daha azdır. 1 Kb veriyi 100 metrelik bir uzaklığa iletmek için gereken enerji, yaklaşık olarak saniyede 100 milyon komut işleyen bir işlemcide 3 milyon komut işlemek için gereken enerjiye eşittir. Enerji pil veya kapasitörler içerisinde saklanmaktadır. Piller algılayıcı düğümlerinin enerji ihtiyaçlarının temel kaynağıdır. Şarj edilebilir ve şarj edilemez olmak üzere iki tip pil kullanılmaktadır.

Ayrıca piller içerisinde kullanılan elektromekanik malzemeye göre de sınıflandırılabilir (NiCd – Nikel Kadmiyum, NiZn - Nikel Çinko, Nimh - Nikel Metal hidrid, Lityum-İyon). Günümüzdeki sensörler yenilenebilir enerji kaynaklarını da (güneş enerjisi, ısı enerjisi, titreşim enerjisi vb.) kullanabilecek şekilde geliştirilmektedir. Kullanılan en önemli iki güç koruma politikası Devingen Güç Yönetimi (Dynamic Power Management DPM) ve Devingen Voltaj Ölçeklendirme (Dynamic Voltage Scaling - DVS)'dir. DPM kullanılmayan veya etkin olmayan

parçaları kapatma görevini gerçekleştirir, DVS yaklaşımı determinist olmayan iş yüküne bağlı olarak güç seviyeleri arasında geçişler yaparak çalışır. Voltajı frekans ile birlikte değiştirerek güç tüketiminde kuadratik azalmalar sağlamak mümkündür.

2.2.1.5. Algılayıcılar

Algılayıcılar sıcaklık, basınç gibi fiziksel durumlardaki değişimlere ölçülebilir tepkiler üretebilen donanım aygıtlarıdır. Algılayıcılar gözlemlenecek alanın fiziksel verisini ölçer veya algılarlar. Algılayıcılar tarafından algılanan sürekli analog sinyaller "Analog-to-Digital" çeviriciler yardımıyla sayısallaştırılarak denetleyicilere daha fazla işlem için gönderilir. Algılayıcı düğümler küçük boyutlarda, düşük enerji tüketimli, yüksek hacimsel yoğunluklarda çalışabilen, otonom ve gözetimsiz çalışan, ortama uyum sağlayabilen özelliklere sahip olmalıdır. Kablosuz algılayıcı düğümleri sadece sınırlı güç kaynağına sahip (0.5 Ah ve 1.2 V gibi) mikro elektronik algılayıcı aygıtlarını kullanabilir. Algılayıcılar üç kategori şeklinde sınıflandırılmaktadır.

2.2.1.5.1. Pasif, her yöne acık (yönsüz) algılayıcılar

Pasif algılayıcılar ortamı aktif araştırma ile değiştirmeden verileri toplayan algılayıcılardır. Kendi enerjilerine sahiptir, enerji analog sinyali yükseltmek için gereklidir. Bu ölçümlerde "yön" şeklinde bir kavram yoktur.

2.2.1.5.2. Pasif, dar ışınlı algılayıcılar

Bu algılayıcılar pasiftir ancak iyi tanımlanmış ölçüm yönü kavramına sahiptir. Tipik bir örnek olarak kamera verilebilir.

2.2.1.5.2. Aktif algılayıcılar

Bu gruptaki algılayıcılar ortamı aktif olarak araştırırlar, örnek olarak sonar veya radar algılayıcıları veya küçük patlamalarla şok dalgaları üreterek çalışan bazı sismik algılayıcı tipleri verilebilir.

Kablosuz Algılayıcı Ağlar' daki kapsayıcı teorik çalışmalar Pasif, yönsüz algılaycıları kastetmektedir. Her algılayıcı düğüm belirli bir kapsama alanına sahiptir. Bu kapsama alanındaki gözlemlerini güvenilir ve doğru bir şekilde raporlayabilir. Kapsama alanını arttırmaya ve algılayıcıların dizilimini iyileştirmeye yönelik çalışmalar yapılmaktadır.

Algılayıclardaki güç tüketim kaynakları olarak a) Sinyal örnekleme ve fiziksel sinyalleri elektrik sinyallerine çevirme, b) Sinyal iyileştirme ve c) Analog'tan sayısala çevirme sayılabilir.

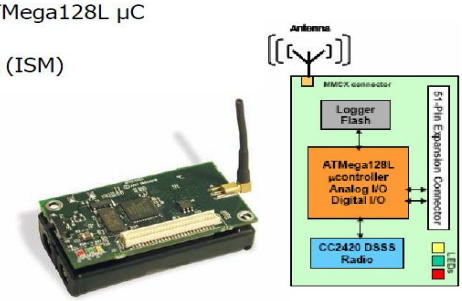
Algılayıcı düğümlerinin birbirleriyle haberleşebilmesi için kullanılabilecek farklı iletişim yöntemleri vardır. Bu yöntemler optik iletişim (lazer), kızıl berisi (IR) ve radyo frekansıdır (RF). Lazer iletişim görüş alanı gereksinimi, atmosfer koşullarından etkilenme ve tek yönlü olması nedeniyle tercih edilen bir yöntem değildir. Kızıl berisi iletişim ise yine tek yönlü olması ve kısa erimi dolayısıyla tercih edilmez. Elektromanyetik dalgalar şeklinde yapılan iletişim yöntemidir. En önemli problemi anten gereksinimidir. İletimin ve alımın eniyilenmesi için minimum bir anten uzunluğuna ihtiyaç vardır. Bu uzunluk en az λ/4 (λ taşıma frekansının dalga boyudur) olmalıdır. RF iletişimin avantajları kullanım kolaylığı, bütünlük, ticari olarak yaygın kullanımıdır. Dikkat edilmesi gereken bir başka unsur, güç tüketimini azaltmak için modülasyon, filtreleme, demodülasyon, vb. işlemlerin yapılması gerekliliğidir.

2.2.1.6. Radyo

Kısa mesafe radyolarının iletişim bileşeni olarak kullanımı son derece önemlidir. Çünkü, enerji sarfiyatında mesaj alma verme – alıcı/verici işlemleri toplam sarfiyat üstünde en etkin kalemlerin başında gelir. Radyonun dizayn ve seçim aşamasında en az 3 farklı katman dikkate alınmalıdır; Fiziki, MAC, ve Network. Fiziki katman diğer alıcı/verici yada alıcılarla fiziki bağlantıyı kurmakla yükümlüdür. Bu seviyedeki ana görevler; sinyal kipleme (modülasyon) ve verinin şifrelenerek iletişimin, kanal gürültüsü ve sinyal karışmasından korunmasıdır. Band genişliğini etkin bir biçimde kullanmak ve geliştirme maliyetini azaltmak için yapılması gereken standart

uygulama; birden çok radyonun aynı ortamı (birbirine bağlı) paylaşmasıdır. Ortamın paylaşımı (zaman veya frekans) MAC katmanı tarafından kolaylaştırılmıştır. Son olarak Network katmanı bir mesajın kaynaktan hedefe transfer edilebilmesi için izlemesi gereken yolun tespitinden sorumludur. Şekil 9-10-11-12' de özel olarak Micaz olarak adlandırılan düğümün özellikleri görülmektedir.

- CPU:
  - 8-bit, 16 MHz, 16 MIPS ATMega128L µC
- Radyo:
  - Frekans: 2400-2483.5MHz (ISM)
  - İletim Hızı: 250Kbps
  - Mesafe: 75-100m
- Bellek:
  - 128 KB kod bellek (Flash)
  - 4 KB SRAM
  - 4 KB EEPROM
- Sensörler ve ADC
  - 8-kanal, 10-bit ADC
  - ışık, ses, sıcaklık, basınç,n
- Kullanıcı tarafından ayarlanabilen 3 adet LED
- 2xAA Pil



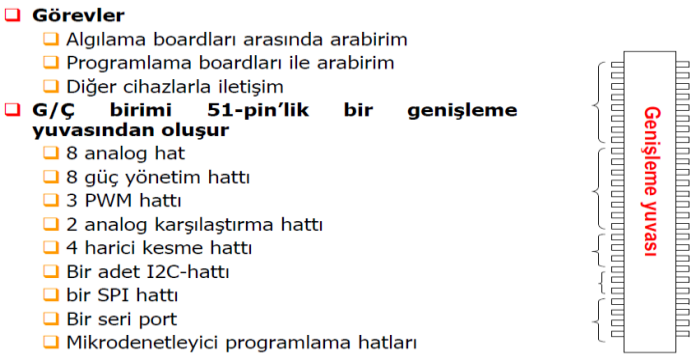
MicaZ

Şekil 2.9: Genel özellikler

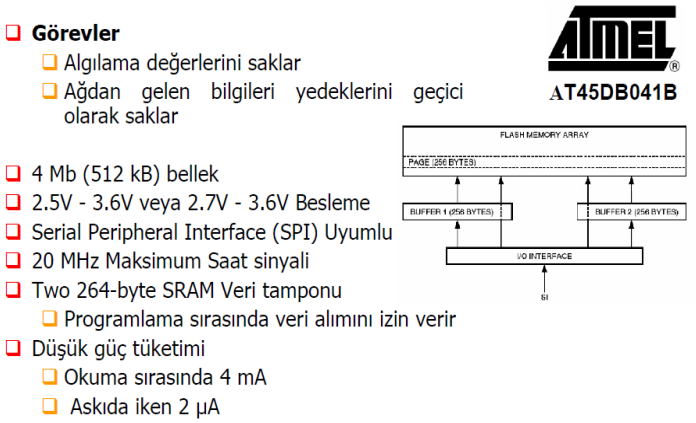
- **Görevler**
  - Uygulamaların çalıştırılması
  - Kaynakların Yönetimi
  - Çevre Birimlerin Kontrolü
- **Atmel AVR ATMEGA128L**
  - 16 Mhz'de 16 MIPS Çalışabilme
  - RISC Mimari
  - 133 Komut – Çoğu tek saykıllık
  - 8 bit ALU/veri yolu
  - 128 Kb Kod Bellek
  - 4 Kb SRAM – Veri Bellek
  - 4 Kb EEPROM
  - 53 Programlanabilir G/Ç hattı
  - 3 zamanlayıcı, 2 UART, 1 SPI port
  - JTAG hata ayıklama desteği


ATMEGA128

Şekil 2.10: İşlemci



Şekil 2.11: Giriş/Çıkış Birimleri



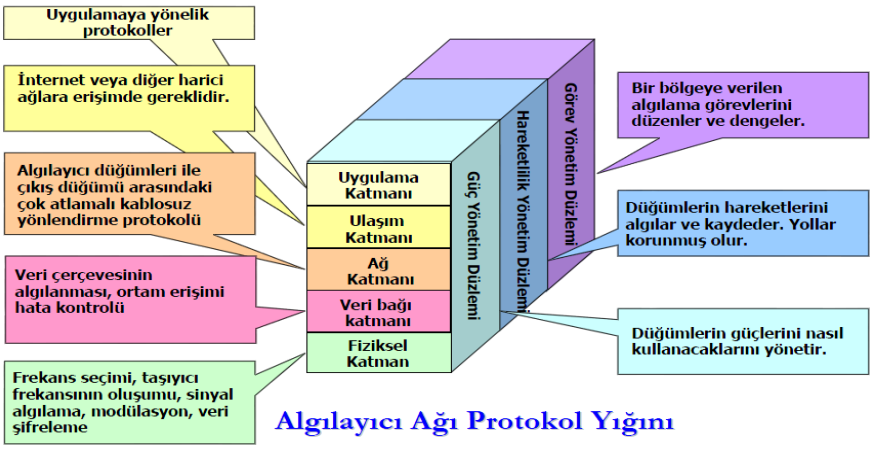
Şekil 2.12: Veri saklama

2.3. Kablosuz Algılayıcı Ağ Mimarisi

Kablosuz Algılayıcı Ağların temel elemanları algılama, veri işleme ve haberleşme özelliğine sahip algılayıcı düğümlerdir. Bilindiği gibi algılayıcı düğümler, herhangi bir kablo olmaksızın, izleyecekleri ortama rastgele saçılmış halde bulunurlar. İzlemenin yapıldığı ortamda toplanan veri genelde 3 seviyede işlenilir.

1. İzlenilecek ortamdaki olaylar, algılayıcı düğümler tarafından algılanır. Her bir algılayıcı düğüm elde ettiği veriyi ayrı ayrı işlemektedir.
2. İkinci seviye de her düğüm algılayıp, işledikleri veriyi komşularına yollamaktadır.
3. Algılayıcı ağ haberleşmesinde ki en üst katman, işlenmiş verinin baz (base) olarak adlandırılan merkeze yollanılmasıdır.

Baza gönderilen veri eğer başka kıstaslar eşliğinde tekrar analiz edilecekse ya da başka amaçlar için kullanılacaksa bu işlemlerin yapılacağı sistemlere ya da merkezlere iletimi sağlanır. Şekil 2.13' te Kablosuz Algılayıcı Ağların iletişim mimarisi görülmektedir.



Şekil 2.13: Kablosuz Algılayıcı Ağların İletişim Mimarisi

2.3.1. Kablosuz algılayıcı ağların mobil tasarsız (ad-hoc) ağlara göre avantajları

Geleneksel kablosuz tasarsız ağlar için birçok algoritma ve protokol önerilmiş olsa da, bu algoritma ve protokoller algılayıcı ağlarının eşsiz özellik ve uygulama gereksinimlerine uymamaktadır. Algılayıcı ağlar hatalara eğilimli ve genel kimliğe sahip olmayabilir ancak yine de geleneksel kablosuz ad-hoc ağlara göre bazı avantajlara sahiptir:

- Binlerce algılayıcının dağıtılmasıyla çok geniş alanların kapsanmasına olanak sağlarlar.
- Ağ oluşturmuş olan algılayıcılar, bir algılayıcının hatası durumunda da doğru bir şekilde çalışmaya devam ederler. Böylece, yüksek seviyeli artıklık ("redundancy") geniş ölçüde hata toleransı sağlamış olurlar.
- Kablosuz algılayıcı ağlar ayrıca sink düğümlerinin başka ağlara (İnternet, Geniş Alan Ağları, vb.) bağlantı sağlamasıyla uzaktan erişim olanağını arttırırlar.
- Ayrık fenomenini ("discrete phenomenon") yerelleştirerek güç tüketimini azaltabilirler.
- İnsan müdahalesini ve yönetimini azaltabilirler.

- Gözetimsiz, erişimi zor bölgelere ortamlarda çalışabilirler.

Değişen ağ durumlarına devingen olarak tepki gösterebilirler.

2.3.2. Tasarsız algılayıcı ağları nasıl çalışır?

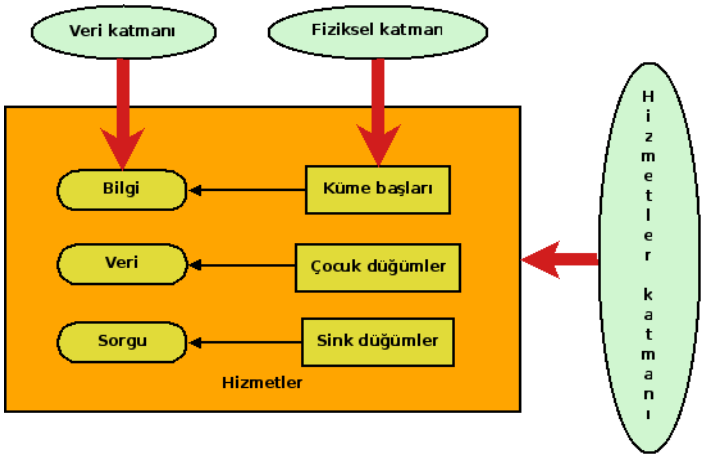
Ad hoc algılayıcı ağı, merkezi bir yönetim veya destek hizmetlerinin yardımı olmadan geçici bir ağ oluşturan algılayıcı düğümler kümesidir. Başka bir söyleşiyle ana istasyonlar gibi sabit bir altyapının olmadığı ağlardır.

Genel olarak, algılayıcı düğümler kablosuz radyo frekans (RF) alıcı-vericilerini ağ arabirimi olarak kullanarak, birbirleriyle iletişimi multi hop kablosuz bağlantılar şeklinde gerçekleştirirler. Ağdaki her algılayıcı düğüm ayrıca yönlendirici ("router") şeklinde davranarak veri paketlerinin komşu düğümler arasında iletilmesini sağlar.

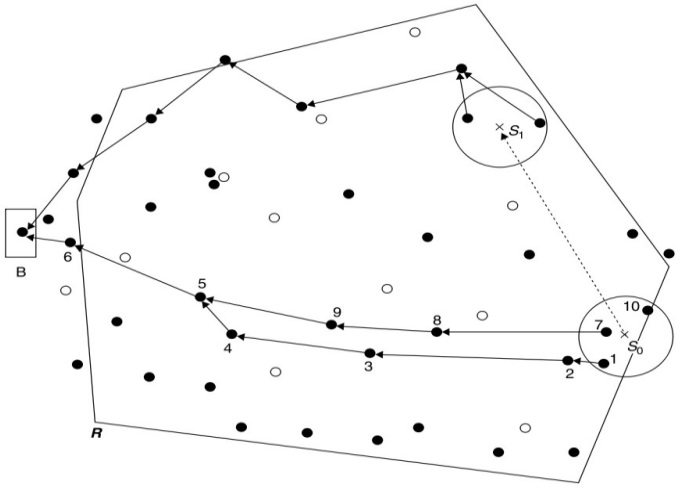
Ad hoc ağlar topolojideki sık değişimlerle ilgilenmek zorundadır. Bu algılayıcıların hataya eğilimli olmasından ve çöken düğümlerin yerini tutmak veya ilgilenen alanı genişletmek için yeni algılayıcı düğümlerin ağa katılmasından dolayı gereklidir. Bu özelliklerden dolayı ad hoc algılayıcı ağının tasarımındaki temel zorluk öz örgütlenebilen algılayıcı ağlarının ve haberleşen iki düğüm arasındaki yolu verimli bir şekilde belirleyen devingen yönlendirme("routing") iletişim kurallarının (protokoller) geliştirilmesidir.

Ufak sensörler düşük enerji tüketimiyle daha kapsamlı bir algılama işini sağlamak için aralarındaki koordinasyonu gerçekleştirmeleri, kümeler ("cluster") halinde çalışmalarıyla mümkündür. Her bir küme algılayıcıların yönetimi için kendisine bir küme başı ("cluster head") atar. Küme başlarının avantajları:
Kümeleme sensörlerin daha global hedeflere erişmek için kendi yerel etkileşimlerini verimli bir şekilde düzenlemelerine olanak sağlaması
Ölçeklenebilirlik
İyileştirilmiş sağlamlık ("improved robustness")
Daha verimli kaynak kullanımı
Düşük enerji tüketimi

Sağlam bağlantı veya düğüm çökmeleri ve ağ bölümleri:



Şekil 2.14: Algılayıcı ağ mimarisi



Şekil 2.15: Veri toplayan bir algılayıcı ağ

Şekil 2.14' te bir Kablosuz Algılayıcı Ağ mimarisi, Şekil 2.15' te ise veri toplama işlemi gösterilmiştir. Şekil 11'dende görüldüğü gibi, üç katman vardır: hizmet katmanı, veri katmanı ve fiziksel katman. Hizmetler yönlendirme iletişim kuralları, veri toplama ve veri yayma hizmetlerini (bunlarla sınırlı değildir) içerir.

Fiziksel katman fiziksel düğümlerden oluşur. Bu düğümler sinkler, çocuk düğümler, küme başları ve ebeveyn düğümlerdir. Ebeveyn düğümler iki veya daha fazla küme

başına bağlanan düğümlerdir. Tüm mesajlar veri katmanında neredeyse modellenmiştir.

Sink düğümleri ya tüm sensör ağına veya kullanılan sorgunun tipine bağlı olarak sadece belli bir bölgeye sorgu yayınlarlar("broadcast"). Sensör düğümleri bir algılamada bulunduğunda (nesne algılama, sıcaklık titreşim konum değişimleri, vb.) bu algılama sonucu elde ettikleri veriyi komşu sensör düğümlerine yayınlarlar.

Her bir sensör (çocuk) en az bir küme başına bağlandığı için, küme başları bu veriyi alırlar. Küme başlarının görevi bu veriyi işlemek ve birleştirmek, sonra komşu düğümlere yayınlama yoluyla sink düğüme aktarmaktır. Küme başları çocuk düğümlerden birçok veri paketi aldığı için verileri süzmeli, işlemeli ve bilgi haline getirmelidir.

Sensör uygulamalarında sensör düğümlerindeki bellek, pil ve işlem gücü gibi donanımsal sınırlamalar, hedeflenen alana oldukça fazla sayıda sensör düğümünün konuşlandırılmasıyla karşılanır. Bu sensör düğümleri bir büyük kablosuz ad hoc ağ şeklinde işleri işbirliği içerisinde gerçekleştirirler. Düğümler arasındaki mesafelerin kısa olması, her düğümün iletim çapını düşürerek güç korunmasına da yardımcı olur.

Şekil 12'de gösterilen ağın amacı, x ile gösterilmekte olan ve R alanı içerisinde kalan nesneden veri alabilmektir. Ana istasyon B ile gösterilmektedir. Dolu daireler yaşayan düğümleri, boş daireler ölü düğümleri göstermektedir. Bu örnekte nesnenin algılanabilmesi için en azından iki sensör gerekmektedir. Nesne S0 konumunda iken 1 ve 7 nolu düğümler algılama işini yaparlar. 2, 3, 4, 5 ve 6 nolu düğümler 1. düğümden; 8, 9, 5 ve 6 nolu düğümler 7 düğümden verinin iletişim yolunu oluşturur. Veri 5. düğümde birleştirilebilir. Bu tek mümkün algılama şekli değildir, 7 nolu düğüm yerine 10 nolu düğümde nesnenin algılamasını yapabilir. Böylece verinin iletişim yolu da değişecektir. Nesne S1'e doğru ilerledikçe algılama, aktarma ve birleştirme görevleri değişecektir.

2.3.3. Algılayıcı ağlarda veri birleştirme ve yayma

Algılayıcı ağların adres merkezli olması yerine veri merkezli olması gerekir. Algılayıcı ağların temel fikri çok ucuz ve basit algılayıcı düğümlerinin tasarlanmasıdır. Bu şekilde algılayıcı uygulamaları binlerce atılabilir düğümler herhangi bir yük oluşturmadan kullanılabilir. Her bir düğüme tekil bir adres vermek, özellikle algılayıcı ağ uygulamasında binlerce düğüm kullanıldığında oldukça masraflı bir iştir. Tek bir algılayıcı düğümünün sınırlı bellek ve işlem gücünden ziyade bizi ilgilendiren algılayıcı düğüm gruplarıdır.

Veri merkezli uygulamalar, algılayıcılar tarafından üretilen verilere odaklanmıştır. Bu yüzden algılayıcı #46'ya bir sorgu göndermek yerine, sorgu üzerine GPS (Küresel konumlandırma sistemi) yerleştirilmiş algılayıcı yardımıyla konumu bilinen #6 nolu bölgeye gönderilmektedir. GPS kullanımının arkasındaki ana fikir, veri yayılımı acısından önemli olan algılayıcıların konumunu kolayca belirlemektir. Böylece GPS gömülmüş algılayıcılar yardımıyla konumları bilinen belli bir alana gönderilebilmektedir. (Maalesef gömülü GPS algılayıcı düğümler görüş acıları engellediğinde yanıltıcı olabilmektedir, ayrıca GPS tam konumu değil belli bir konum aralığını verir. Bu yüzden birbirine yakın düğümler aynı GPS değerini üretecektir).

Birleştirme (“Aggregation”): Bazı sensör düğümleri komşularından aldığı verileri birleştirmekle yükümlüdür. Birleştirici düğümler sink düğümlerine bilgi halinde gönderebilmek için veriyi süzebilir, işleyebilir ve saklayabilir. Birleştirme işlemi aşağıdaki nedenlerden dolayı faydalıdır:

- Bilgi döngüsünü arttırmak
- Doğruluk düzeyini arttırmak
- Çöken sensör düğümlerini karşılamak için veri artıklığı (“redundancy”)

Yayma (“Dissemination”): Sensörler tarafından üretilen veri, hedefine ulaşmak için birçok ara düğüm üzerinden geçmelidir. Ara düğümler çöktüğü zaman gelen mesajların iletilmesinde sorunlar oluşmaktadır. Diğer sorunlar aşağıda listelenmiştir:

- Yönlendirme iletişim kuralları en kısa yolu bulmalıdır.
- Artıklık: Bir algılayıcı aynı veri paketini birden fazla alabilir.

Algılayıcı ağlarda veri yayma için iki senaryo vardır: Sorgu güdümlü ve sürekli güncelleme. Her senaryo belirli tipteki algılayıcı uygulamalarına uygulanabilmektedir. Birinci yöntem bire-bir ilişki olarak kullanılmaktadır, sink bir sorgu yayınlayarak algılayıcı düğümlerinden bu sorgusuna yönelik raporlanan yanıtları alır. Örneğin sink bir nesnenin (düşman tankı veya bir hayvan) ilk defa görünüp görünmediğine dair sorguda bulunabilir.

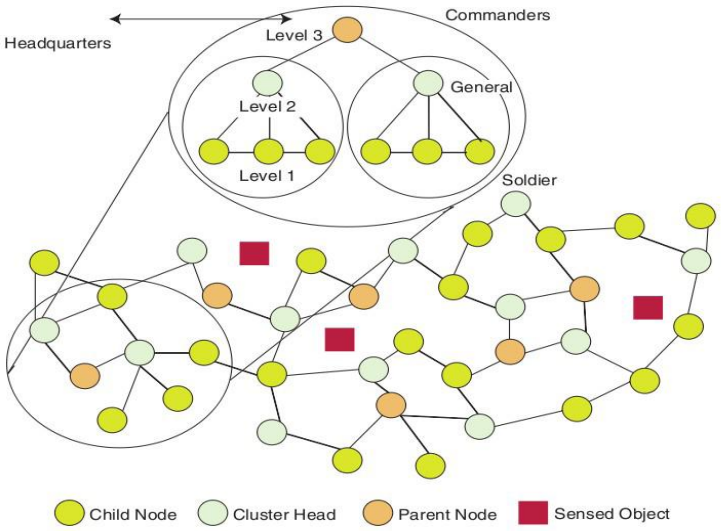
İkinci senaryo bire-çok ilişki örneğidir. Sink düğümü bir sorgu gönderir ve sorgusuna yönelik sürekli güncellemeler alır. Örneğin sink hareketli bir nesnenin doğrultusunu sorgulayabilir. Sensör düğümleri bu sorgu sonucunda hareket eden nesnenin yeni konumunu sürekli olarak raporlarlar. Sürekli güncellemeye dayanan veri yayma senaryosu yüksek oranda enerji tüketimine neden olmaktadır, ancak sorgu güdümlü yönteme Gore daha güvenilir ve doğrudur. Bunun nedeni daha fazla sensörün sorgu raporlamada yer almasıdır. Örnek olarak sensör düğümlerinin bir park alanı ağında tek adreslenebildiğini düşünelim. Bu şekilde tüm boş park alanlarının belirlenmesi kolaylaşacaktır.

Başka bir örnek ise bir uçakta her yolcunun koltuğunun üstüne herhangi bir yolcunun beklenmeyen hareketini belirleyen sensörler yerleştirmektir. Herhangi bir tehlike durumunda sensör ağı uçağın kontrolünü almak üzere işbirliğine gidebilir (Örneğin ışıkları kapatmak, pilotun kokpit kapısını kapatmak gibi).

Bu sensörlerin kullanılmasındaki en önemli avantaj herhangi bir hareket durumunda bağlantılılığı korumasıdır. Bu sensörler oldukça küçük olabildiği için yanlışlıkla yerlerinin değiştirilmesi oldukça olasıdır. Bu yüzden sensör ağları, bazı sensörler yerlerinden oynasa bile bağlantılılığı sağlamak zorundadır. Örneğin bu sensörler bir ormanda bulunabilir ve her turlu hareket ettirilmeye karşı savunmasız olabilir (Örneğin insan, hayvan, böcek, yağmur, rüzgar vb.).

2.3.4. Sıradüzensel algılayıcı ağlar

Sıradüzensel sensör ağları, askeri sıradüzenini model alan ve bu modele göre çalışmasının biçimlendiren bir yapıdır. Bir taktiksel askeri ağ yapısı incelenerek bu sıradüzensel sensör ağlarının benzer yapısı anlaşılabilir. Şekil 2. 16' da sıradüzensel ağların yapısı görülmektedir.



Şekil 2.16: Algılayıcı sıradüzensel ağ

Bir taktiksel askeri ağ, komutanlar tarafından (parent nodes) yönetilen birim gruplarından (clusters) oluşur. Bu komutanlar emirleri ana karargahtan (sink node) alarak, birimdeki gözlemleri ve verileri geriye gönderirler.

Komutanlar gelen emirleri generallere (cluster heads) gönderirler. Her general bir grup askerden (children) sorumludur. Askerler diğer askerler ve generalle yerel iletişim kurarlar.

Askerlerden mesajları alan generaller bu bilgiyi komutanlarına iletir.

Savaş alanında bir gözlemde bulunan askerler bu bilgiyi generallere aktarırlar. General askerlere emir vererek eyleme geçmelerini sağlayabilir veya komutanına

danışabilir. Karar eylemlerinde (saldırı gibi) sadece ana karargah bilgiye dayanarak bir karar verebilir.

2.4. Kablosuz Algılayıcı Ağlarda Güvenlik

Güvenlik ve Gizlilik birçok WSN (Wireless Sensor Network) uygulamasında aşırı derecede öneme sahiptir. Bu uygulamalardan bazıları ; savaş alanlarında kullanılan hedef izleme ve takip sistemleri, kanun yaptırımı uygulamaları, otomotiv telemetrik uygulamaları, işyerlerinde odaların izlenmesi, benzin istasyonlarında sıcaklık ve basınç ölçümleri ve orman yangın tespit sistemleridir.Tüm bu uygulamalar çok sayıda yarara sahiptir ve geliştirilme potansiyelleri yüksektir. Ancak, sensör bilgisi düzgün bir şekilde korunmaz ise, bilginin yanlış sonuçlara yol açacak şekilde tahrip edilmesi olasıdır.

Algılayıcı Ağ çalışmaları en hızlı biçimde askeri uygulamalarda kendini göstermektedir, bu alandaki güvenliğin önemi herkesçe bilinmektedir. Savaş alanı hakkında bilgiyi, kimsenin hayatını riske atmadan toplayabilmesine karşın, tatmin edici bir şekilde korunmayan WSN' ler düşmanın eline geçtiğinde güçlü bir silah olarak kullanılabilir. Bu tip uygulamalar için sağlam güvenlik önlemleri alınmalıdır.

WSN' lerin ticari uygulamalarında ise "Gizliliğin Korunumu" meselesi, ağın güvenli ve stabil halde çalışır olması kadar önemle ele alınmalıdır. Kişiler hakkındaki fizyolojik ya da psikolojik bilginin güvenliği her kullanıcı tarafından korunması gereken bilgiler içerisindedir. WSN uygulamaları ne kadar yaygınlaşırsa ve karmaşıklaşırsa, bu sistemlerin yetkisiz kullanıcılara karşı korunmasının önemi artacaktır. Algılayıcı Ağ uygulamaları çok çeşitli fiziksel ortamlarda ve kısıtlamalar altında çalışmaktadır. Algılayıcı ağ düğümlerinin etkin bir şekilde kullanılması için her uygulama için farklı uyarlamalar ve tasarımlar gerekecektir. Çünkü güvenlik ve gizliliğin sağlanması önemli ölçüde hesaplama ve depolama kaynağının kullanılmasını gerektirir. Güvenliği sağlamak için gerekli mekanizmalar, hedef uygulamanın mimari yapısına ve içinde bulunduğu fiziksel çevreye uygun hale getirilmelidir.

2.4.1. Kablosuz algılayıcı ağların güvenliğini tehlikeye atan durumlar

Aşağıda açıklanan dört durum; Düşman Saha, Kaynakların Sınırlılığı, Ağ İçinde İşlem Yapma ve Uygulamaya Özel Mimari Yapı, WSN' lerin güvenliği konusunda dikkat edilmesi gereken durumlardır.

2.4.1.1. Düşman saha

WSN' ler savaş alanları gibi düşman bölgelere yerleştirilebilir. Bu durumlarda düğümler fiziksel saldırıya karşı korunmasızdır. Güvenlik bilgisi, genelde kaybedilmesi (düşman tarafından tahrip edilmesi) muhtemel düğümlerden alınabilir.

Kurcalanamayacak şekilde tasarlanan düğümler düşman sahalarda güvenliğin sağlanması için yapılması gereken işlemlerden biridir. Fakat yapılması gereken bu işlem basitlikten çok uzaktadır, bellek ve hesaplama gereksinimleri açısından kesinlikle pahalı bir işlemdir. Sensör düğümlerinin fiziksel olarak erişiminin mümkün olmasından dolayı, WSN' ler için güvenlik mekanizmaları bir ya da daha çok düğümün tehlikeye atıldığı durumlarla ilgilidir.

2.4.1.2. Kaynakların sınırlılığı

Algılayıcı ağ düğümleri kompakt bir yapıda tasarlanmıştır. Bu yüzden boyut, enerji, hesaplama gücü ve depolama noktası yönünden sınırlı yapıya sahiptirler. Sınırlı kaynaklar gerçekleştirilmek istenen güvenlik algoritmalarını ve protokollerini sınırlandırırlar. WSN' ler için güvenlik çözümleri; güvenliğe harcanan kaynaklar ve elde edilen korunma arasından yapılan tercih tarafından tanımlı çözüm alanında işler. Sınırlı kaynaklar, düğümlerin yeni saldırı tiplerine karşı açık hale gelmesine sebebiyet verir.

2.4.1.3. Ağ içinde işlem yapma

WSN' in kullanılabilir enerjisinin büyük çoğunluğunu düğümler arasındaki haberleşme tüketir, enerjinin küçük bir kısmı algılama ve hesaplama için kullanılır. Bu sebepten dolayı WSN' ler sınırlandırılmış işleme ve veri toplama gerçekleştirirler. Bu tip iletişim tarzı için; uygun güvenlik mimarisi anlık komşuluk durumlarında bir grup anahtarının düğümler arasında paylaşılması ile oluşturulur. Ancak, düğümlerin yakalanmasının olası olduğu ortamlarda, gizli olarak atanmış paylaşılmış simetrik anahtar tehlike altında kalır.

2.4.1.4. Uygulamaya özel mimari yapı

Yukarıda anlatılan durumlardan dolayı WSN' ler uygulamaya göre değişen mimari yapılara sahiptirler. Genel amaçlı mimari yapının esnekliği kaynakların etkin kullanımını gerektirir. WSN' ler neredeyse her yönden kaynakların tüketimini optimize etme ve performansı yükseltmek için uygulamanın özelliklerine göre ayarlanabilirler. Bu, ağı dizayn eden kişiye çeşitli güvenlik açıklarını tespit etme ve bu açıklara göre güvenlik mekanizmalarını düzenleme izni verir.

2.4.2. Kablosuz algılayıcı ağların güvenliği için gereksinimler

Bu bölümde kablosuz algılayıcı ağlar tarafından gereksinim duyulan güvenlik özellikleri incelenecektir.

2.4.2.1. Dışarıdan gelen saldırılara karşı dayanıklılık

Birçok uygulama dışarıdan gelen saldırılara karşı güvenlik gerektirir. Gizlice dinleme (eavesdropping) ya da Paket Enjeksiyonu (packet injection) gibi bilinen saldırılara karşı standart güvenlik tekniklerinin seviyesini yükseltmemiz gerekebilir. Örnek olarak; şifrelenmiş primitifler kullanarak orijinalliği ve iletişimin gizliliğini ağ içerisindeki düğümler arasında sağlayabiliriz. Buna ek olarak, düğümlerde meydana gelebilecek hatalara karşı dayanıklı mekanizmalar dizaynetmemiz gereklidir. Bu dayanıklılığa erişmek için büyük miktarlarda düğüm kullanmak ve gerekenden fazla

sayıda düğüm bulundurmak (fazlalık) gereklidir böylece birkaç düğümde oluşabilecek hata sonrası sistemin bütünü fazlaca etkilenmez.

Ayrıca işlevini kaybeden düğümlerin yerine geçen düğümler dolayısıyla ağın topolojisinde değişim meydana gelecektir, bunu anında fark edip yeni topolojiye göre iletişimi sağlayacak protokollere ihtiyaç vardır.

2.4.2.2. İç krizlere karşı direnç

Güvenlik-Kritik Algılayıcı Ağlar, tehlike altındaki düğümleri göz önüne alan mekanizmaların üretilmesini gerektirir. İdeal olarak tehlike altındaki düğümleri saptayıp sahip oldukları kriptografik anahtarları geri alabilmeliyiz. Fakat pratikte bu her zaman mümkün değildir. Bu duruma alternatif tasarım yaklaşımı; düğüm kaybına ya da tehlike altında bulunmasına dayanıklı mekanizmalar tasarlamaktır, böylece azar azar sistemin düğüm kaybetmesi sistemin tümden kaybına değil de performansında küçük çaplı düşüşlere neden olur.

2.4.2.3. Güvenliğin gerçekçi seviyesi

Genel olarak güvenliğin gereksinimleri tartışılırken, sensör ağların uygulamadan uygulamaya güvenlik gereksinimlerinin değişim göstereceği unutulmamalıdır. Örnek olarak tıbbi gözlem cihazlarında insanın vücuduna yerleştirilmiş sensör düğümlerinden hastanın sağlık durumu izlenir, bu durumda güvenliğin amacı hastanın mahremiyetini gizlemektir. Fakat okyanustaki balığın durumunun izlendiği bir uygulamada balığın mahremiyetini gizlemek çok önemli değildir.

2.4.2.4. Veri gizliliği

Bir sensör ağ kesinlikle sensör bilgisini komşu ağlara sızdırmamalıdır. Bir çok uygulamada (örn. anahtar dağıtımı) düğümler çok önemli veri iletirler. Hassas bilginin gizlenmesindeki standart yaklaşım, veriyi sadece planlanan alıcının sahip olduğu gizli bir anahtarla şifreleyip yollamaktır, böylece gizliliğe ulaşılmış olunur.

Gözlenen iletişim modellerinde, baz ve düğümler arasında güvenli kanallar kurulur ve gerekli olduğu durumlarda diğer güvenli kanallar sonradan (geç önyükleme) devreye sokulur. Algılanan verinin gizliliğinin garanti altına alınması veriyi, eavesdropper (kulak misafiri) tipi saldırılardan korumak için önemlidir. Bunu sağlamak için standart şifreleme fonksiyonları kullanılabilir (örn: AES blok şifreleme) ya da gizli bir anahtar iletişim halindeki bölümler arasında kullanılabilir. Ancak, şifreleme tek başına yeterli bir çözüm değildir, bir eavesdropper alıcıya gönderilen şifreli anahtar üzerinde analiz yaparak, önemli veriye ulaşabilir. Şifrelemeye ek olarak algılanan verinin gizliliği, baz istasyonlarında yanlış kullanımının engellenmesi için erişim kontrol kurallarına ihtiyaç duyar.

Örnek vermek gerekirse, kişisel yer tespit uygulaması verilebilir. Kişinin yerini tespit eden sensörlerin, algıladıkları veriyi bir Web Server'a yolladığını düşünelim. İzlenen kişi, yerinin sadece kısıtlı bir grup tarafından bilinmesini isteyebilir, bu yüzden Web Server da erişim hakları kısıtlandırılmalıdır.

2.4.2.5. Veri doğrulama / kimlik denetimi

Sensör ağlarda mesaj doğrulama birçok uygulama için önemlidir. Sensör ağın tasarım kısmında, doğrulama birçok yönetici görevleri (örn. ağın yeniden programlanması ya da sensör düğümünün iş çevriminin kontrolü) için gereklidir. Aynı zamanda, muhalif ya da rakip kişiler kendi mesajlarını kolayca araya sokabilirler. Alıcıların, gelen mesajın yollandığı kaynağı/göndereni doğrulaması gerekmektedir. Veri doğrulama, alıcının mesajın gerçekten belirtilen gönderenden gelip gelmediğini kontrol etmesine olanak verir.

İki taraflı iletişim durumunda, veri doğrulama /kimlik denetimi sadece simetrik bir mekanizmayla sağlanabilir: Gönderen ve alıcı gizli bir anahtarı paylaşır bu anahtar sayesinde tüm haberleşmede kullanılan verinin MAC (Message Authentication Code)' ı hesaplanır. Doğru MAC değerine sahip bir mesaj geldiğinde , "Alıcı" bu mesajın mesajda belirtilen "Gönderen" tarafından gönderildiğini anlar. Bu tarzda bir doğrulama sistemi çok daha kuvvetli güvenlik kriterleri ağ düğümlerine yerleştirilmediği müddetçe yayın ortamı(broadcast) tipindeki ağlara uygulanamaz.

Eğer güvenilir veri, karşılıklı olarak güvenin sağlanmadığı alıcılara yollanmak isteniyorsa, simetrik MAC kullanımı güvenli değildir. Alıcılardan MAC anahtarını bilen biri gerçek gönderen kimliğine bürünerek diğer alıcılara sahte mesajlar yollayabilir. Bu yüzden asimetrik doğrulama, yayın tipi ağlarda güvenliğin sağlanması için gerekli olan mekanizmadır.

2.4.2.6. Veri bütünlüğü

Haberleşmede veri bütünlüğü, alıcının aldığı verinin art niyetli kişilerce aktarım sırasında değiştirilmediğine karşı garanti verir. SPINS (Security Protocols for Sensor Networks) ile veri bütünlüğünü, veri doğrulama ile sağlayabiliriz. Veri doğrulama daha güçlü bir özelliktir.

2.4.2.7. Verinin tazeliği

Sensör ağlar anlık değişen verileri/ölçümleri algılayıp işlediği için, sadece gizlilik ve güvenliğin sağlanması yeterli değildir aynı zamanda her mesajın tazeliğinin de garanti edilmesi gerekir. Verinin tazeliği verinin yeni olduğunu belirtir ve bu sayede art niyetli kişilerin eski mesajları tekrar göndermediğini garanti eder.

İki tip tazelik tanımlanabilir: Zayıf tazelik, kısmi mesaj sırası sağlar. Fakat gecikme zamanı bilgisini taşımaz. Güçlü tazelik, istek-cevap çifti sırasının tamamını sağlar ve gecikme tahminine izin verir. Zayıf tazelik sensör ölçümlerinde gereklidir, güçlü tazelik ise ağ içindeki zaman senkronizasyonu için kullanışlıdır.

2.4.2.8. Kullanılabilirlik

Kullanılabilirliği sağlamak, sensör ağın ömrü boyunca fonksiyonelliğini yitirmeden çalışması demektir. DoS (Denial-of-Service) saldırıları, sık sık sistemin kullanılabilirliğinde kayıplara yol açar.

Pratikte kullanılabilirlikteki kayıp ciddi sonuçlar doğurabilir. Üretim gözleme uygulamasında meydana gelebilecek kullanılabilirlik kaybı potansiyel bir kazanın önüne geçilmesini engelleyebilir bu da finansal kayıplara yol açar. Savaş alanındaki kayıplarda ise sonuç düşmanın bir arka kapı açmasıyla sonuçlanabilir. Çeşitli saldırılar sensör ağın kullanılabilirliğini tehlikeye atabilir. Kullanılabilirliğin sağlanması düşünülürken, düğüm kayıpları ya da hataları ile sistemin tümden çökmesi engellenmeye çalışılmalıdır.

2.4.2.9. Hizmet bütünlüğü

Ağ katmanının üzerinde, sensör ağ genelde çeşitli uygulama- seviyesinde hizmet verir. Veri toplama/kümeleme sensör ağlardaki en yaygın hizmetlerden biridir. Veri toplama işleminde düğümler komşu düğümlerden veriyi alır, veriyi topladıktan sonra ya baz istasyonuna ya da veri üzerinde işlem yapacak olan düğümler varsa o düğümlere iletir.

Güvenli veri toplama göreceli olarak gerçek-dünya verilerinin ölçümünün doğru hesaplanmasını ve bozulmuş düğümlerden gelen verinin tespit edilip hesaplamalara katılmadan atılmasını sağlar. Hizmet örneği olarak zaman senkronlama hizmeti de verilebilir. Sensör ağlar için geçerli zaman senkronizasyon protokolleri, güvenilir bir ortamın oluşturulmasını sağlar. Mevcut araştırma alanlarından birisi de kaybedilen düğümlerin varlığında zaman senkronizasyonu sağlayacak protokollerin geliştirilmesidir.

2.4.3. Saldırılar ve karşı tedbirler

Bu bölümde bilindik saldırılara karşı kablosuz algılayıcı ağlarda alınabilecek tedbirleri incelenecektir.

2.4.3.1. Gizlilik ve kimlik doğrulama

Standart kriptografik teknikler, eavesdropping (kulak misafiri), paket tekrarlama, sahte paket yollama gibi dış kaynaklı saldırılara karşı iletişim bağlantılarının güvenilirliğini ve gizliliğini koruyabilir.

2.4.3.2. Anahtar tespiti ve yönetimi

İki sensör düğümünün güvenli ve doğrulanmış bir bağlantı kurması için, gizli bir anahtarın paylaşımının sağlanması gerekmektedir. Anahtar tespit problemi, ağ üzerindeki bir düğüm çifti arasında gizli anahtarın nasıl tespit edilip kurulması gerektiği konusunu irdeler. Saf bir fikir olarak kurulumdan önce global bir anahtarın her düğüme yerleştirilmesi ve kullanılması düşünülebilir, bu düğümlerin kendi aralarında kolayca iletişimine imkan verirken aynı zamanda muhalif kişilerin sadece bir düğümün anahtarını ele geçirdikten sonra istediği mesajları istediği düğümlere göndermesini ve veri transferini istediği anda takip edebilmesini sağlar.

Ortak Anahtar şifreleme, anahtar tespiti için popüler bir metod olarak karşımıza çıkmaktadır, fakat hesaplama için harcanan kaynaklar göz önüne alındığında, düğümlerin sadece kurulum aşamasında bu değer ile ilklenmesine rağmen, birçok uygulama için fazla masraflı bir seçim olur. Ortak anahtar şifreleme tekniğinin eksiklilerinden birisi DoS (Denial-of-Service- Hizmet Reddi) saldırılarına karşı ağda açık meydana getirmesidir, saldırgan sahte bir mesajı düğüme gönderebilir, böylece düğüm sadece mesajın sahte olduğunu tespit etmek için imza doğrulama gerçekleştirir bu bile sistemi saldırganın istediği gibi yorar.

Son zamanlarda, araştırmacılar, rastgele anahtar ön-dağıtım tekniklerinin anahtar tespit problemine çözüm üreteceği yönünde önerilerde bulunmuşlardır. Fakat, mevcut algoritmaların ölçeklenebilirlik, düğüm uyuşmasının esnekliği, bellek gereksinimleri ve haberleşme genel giderleri açısından geliştirilmesi için daha fazla araştırma gereklidir.

2.4.3.3. Broadcast / Multicast kimlik doğrulama

Broadcast ve Multicast birçok sensör network protokolü için zorunludur. Broadcast ve Multicast'de kaynak doğrulama, yeni bir araştırma konusunu ortaya atar. Olası kazanımlardan birisi sayısal imza kullanmaktır. Kaynak her mesajı özel anahtar (private key) ile imzalar ve tüm alıcılar mesajın doğruluğunu ortak anahtar kullanarak kontrol ederler. Ne yazık ki ortak anahtar şifreleme sensör ağlar için çok pahalı bir tekniktir. Bu problemi çözmek için Perrig (ve başka araştırmacılar)güvenli broadcast doğrulama sağlamak için µTesla protokolünü önermiştir bu protokol sensör düğümler arasında gevşek zaman senkronizasyonunu varsaymaktadır. µTesla'nın arkasındaki temel fikir; Simetrik anahtar şifrelemeye, gecikmiş anahtar açımı ve tek yön fonksiyon anahtar zinciri ile asimetriyi getirmektir.

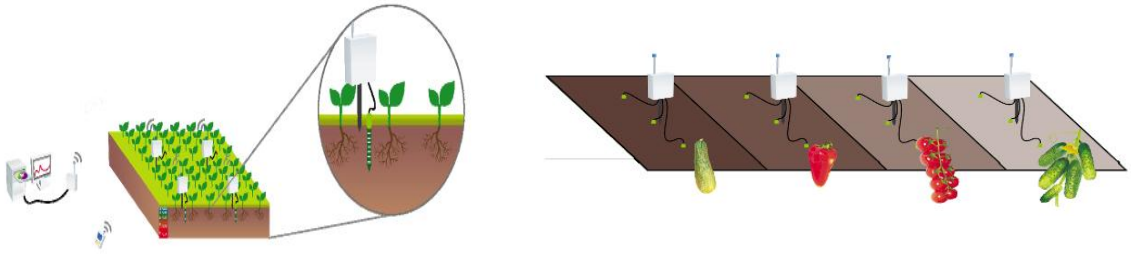
2.5. Örnek Uygulama

2.5.1. Kablosuz toprak nemi ölçüm ve kontrol sistemi

Kablosuz toprak nemi ölçüm ve kontrol sistemi;

* Bitkiye ihtiyacı kadar su verilmesini sağlar.
* Su israfını yok eder.
* Bitkide sık sulama yüzünden hastalık oluşumunu engeller.
* Verim ve kalite artışı sağlar.
* Maliyetleri azaltır.
* Aşırı sulamayla oluşan toprağın tuzlanmasını azaltır.

Şekil 2.17: Sistemin Grafiksel Gösterimi



Şekil 2.18: Sistemle İlgili Diğer Gösterimler

Sistemde toprağın nemi, toprak nemi sensörü ile ölçülür. Toprak nemi ölçer etrafındaki ortamın dielektrik permitivitesini (elektriksel geçirgenliği) ölçen kapasitans algılayıcıdır. Suyun elektriksel geçirgenliği havadan daha fazladır. Suyun elektriksel geçirgenlik katsayısı 80, toprağın 4, havanın 1 dir. Cihaz topraktaki su miktarını hacimsel yüzde olarak ölçer. Ölçümlenen değer elektronik olarak değerlendirilir ve eğer toprağın nem değeri değişmiş ise merkez alıcıya bilgi aktarılır. Merkez alıcı gerekli kontrolleri ve hesaplamaları yapar ve sulama komutu verip vermeyeceğine karar verir. Böylece toprak, yazılımla önceden belirlenmiş nem aralığında (örn %25- %85) sabit kalması sağlanır. Su; yalnızca toprağın nem oranı düştüğünde, yani bitkinin suya ihtiyacı olduğunda verilir.

Aynı sahada ekili farklı bitkilerin nem ihtiyacı da farklıdır. Örneğin bir bitki çok nemli toprağa ve sık sulamaya ihtiyaç duyarken, hemen yanında daha az nem isteyen bir bitki yetiştirilebilir. Sistem bitkiye ve sahaya özel sulama zamanı planlaması oluşturduğundan dolayı bitkilerin sulama ihtiyacı en iyi şekilde karşılanır ve en yüksek verime ulaşılması sağlanır.

Sistem gücünü üzerinde bulunan aküden alır. Saha koşullarına göre değişmekle birlikte uzun süreler şarja ihtiyaç duymadan kullanılabilir (2-3 yıl) Sisteme ihtiyaç duyulan farklı sensörlerde eklenebilir. Solenoid valfler kontrol edilerek açma – kapama yaptırılabilir. Böylece damlama sistemde tam otomatik kontrollü sulama yapılabilir.

3. MATLAB GRAFİKSEL KULLANICI ARAYÜZÜ

3.1. Giriş

MATLAB Grafiksel Kullanıcı Arayüzü, diğer bir söylemi ile MATLAB GUİ, matlab programcısı tarafından hazırlanan grafik tabanlı uygulamaların, son kullanıcıya fare ve klavye arabirimi ile interaktif olarak hitap etmesini sağlayan bir platformdur.

Günümüzde hazırlanan uygulamaların grafik tabanlı oluşu ve bu uygulamaların son kullanıcı tarafından kullanım kolaylığına sahip olması, MATLAB GUİ uygulamalarının gerekliliğinin temel sebeplerinin başında gelmektedir.

Bunun yanı sıra MATLAB GUİ, M-File veya M-Fonksiyon hazırlayabilen herkes tarafından oluşturulacak kadar kolay bir esnekliğe sahiptir. GUİ altında iş yapan komut yapısı ile daha önceden bilinen MATLAB komut yapıları arasında bir fark yoktur.

3.2. MATLAB GUİ Çalışma Sistemi

3.2.1. MATLAB GUİ nasıl çalışır?

Her bir nesne (veya komponent) GUI için tanımlanan programlama dosyasında callback diye adlandırılan ayrı alt rutin programlama parçalarına sahiptir. Bu şekilde, her bir nesnede oluşan olaylara (örnek olarak bir buton nesnesinin tıklanması ile click event oluşması gibi) GUI o olaya ait callback rutinlerini icra ettirir. Yani, GUI hem bir arayüz hem de program çağrılarını icra ettirme mekanizması olarak çalışır.

3.3. Matlab'ta GUI Oluşturma Yöntemleri

MATLAB GUI tasarımları iki ayrı yöntem kullanılarak yapılabilir. Bunlar,

- MATLAB GUIDE aracı kullanılarak,
- M-File programlama yöntemi kullanılarak

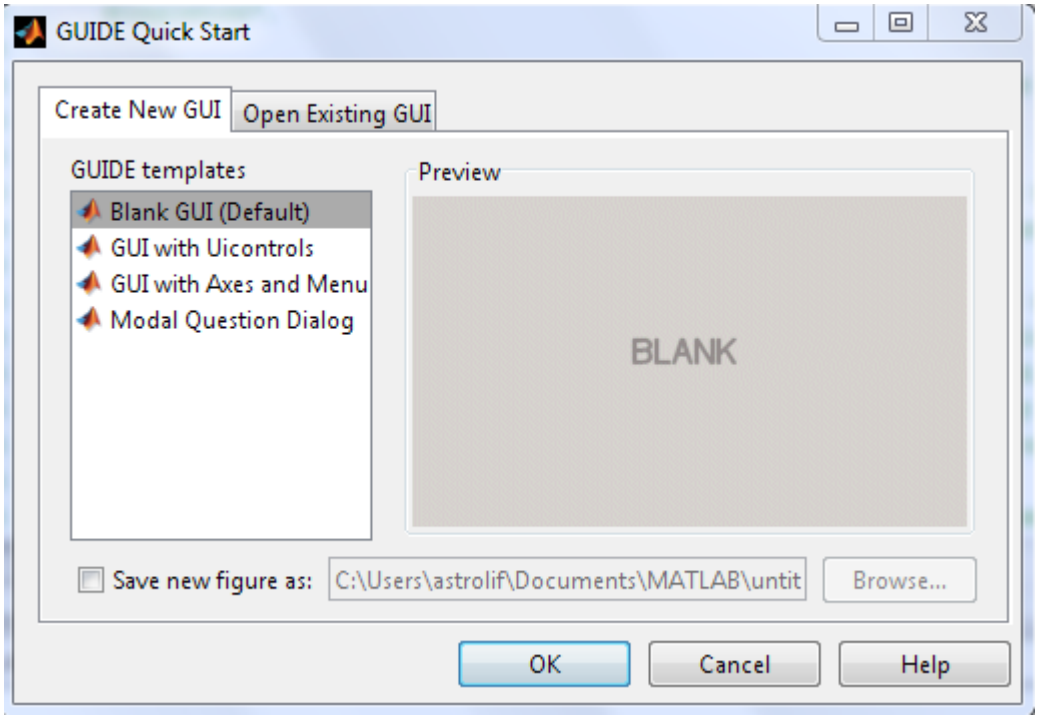
MATLAB GUIDE aracının kullanılması, GUI tasarımında hızlı arayüzler dizayn etmek için özellikle bu işe ilk başlayan programcılar için büyük bir kolaylık sağlar. Bu yöntem, GUI objelerinin figür yüzeyine sürükle-bırak şeklinde yerleştirilmesi ve açılan pencerelerde objelere ait özelliklerin değiştirilmesine dayanır.

M-File programlama yönteminde tüm GUI tasarımları ve callback program parçalarının yazılması tamami ile programlama kodları kullanılarak yapılır. Burada tasarımcı herşeye hakimdir ve bu teknik uzman bir programlama bilgisi gerektirir. Bu yöntem ile tasarım zamanı uzamasına rağmen programcıya her türlü manipülasyonu yapabilme imkanı sunduğu için çok yararlıdır.

3.4. MATLAB GUIDE Aracı ile GUI Tasarımı Oluşturma

GUIDE, matlab'ın GUI tasarımcılarına sunduğu, içerisinde çeşitli araçlar içeren ve kolaylık sağlayan bir grafiksel GUI geliştirme ortamıdır. GUIDE kullanılarak tıkla ve sürükle-bırak tekniği ile GUI arayüzüne nesneler (örneğin butonlar, text kutuları, liste kutuları, grafikler v.s.) kolaylıkla eklenebilir. Ayrıca, eklenen nesnelerin hizalanması, tab sırasının değiştirilmesi de bu ortamın tasarımcılara sunduğu imkânlardan bazılarıdır.

Guide aracına ulaşmak için, Başlat düğmesi tıklanarak MATLAB / GUİDE seçeneği ile veya Dosya menüsü altında yer alan, yeni sekmesi altındaki Gui seçeneği ile Guide penceresi açılabilir. Bir başka yol da komut penceresine GUİDE yazmaktır. Bu adımdan sonra karşımıza Şekil 3.1' teki gibi bir pencere gelir.



Şekil 3.1: Guide penceresi

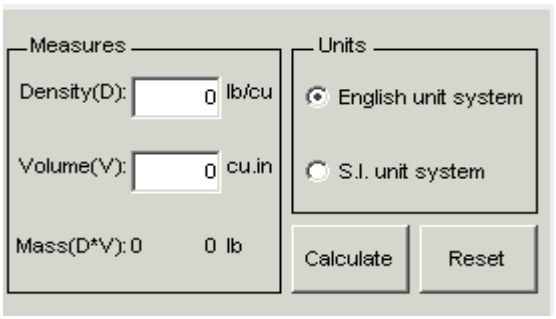
Açılan pencerede GUIDE seçenekleri (GUIDE templates) yer almaktadır. Bu seçenekler ile grafiksel kullanıcı arayüzlü uygulamalar hazırlanır.

Guide Seçenekleri:

- Blank GUI (Default):

Boş GUI çalışma penceresi açar. Bu pencereye istenen GUI objeleri yerleştirilip, bu objelere gerekli işlevler kazandırılarak GUI uygulaması hazırlanır.

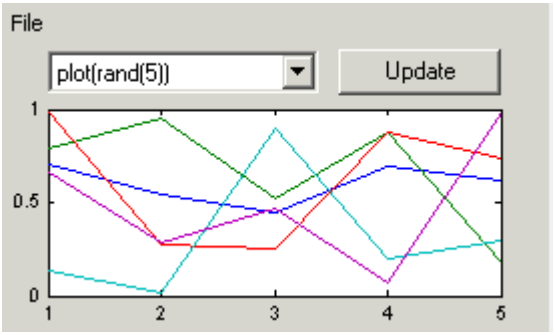
- GUI with Uicontrols:



Şekil 3.2: GUI with UIcontrols

GUI figür yüzeyine hazır olarak yerleştirilmiş üstteki uygulamayı hazırlar. Şekil 3.2' de gösterilen bu uygulamada; static text, edit text, frame, radio buton ve pushbutton objelerinden yararlanılmıştır. Bu objeler hem GUI figür yüzeyine yerleştirilmiş, hem de M-Fonksiyon dosyalarında bu objelere işlev kazandırılmıştır. Böylelikle GUI programcılarına, objelere nasıl işlev eklendiği hakkında fikir verilmektedir.

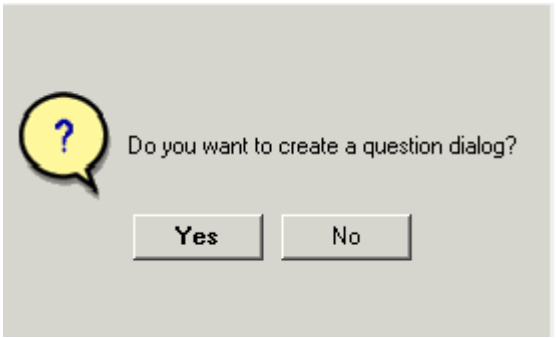
- GUI with Axes and Menu



Şekil 3.3: GUI with axes and menu

Bu seçenek ile GUI figür yüzeyine yerleştirilmiş Axes nesnesi ve bu nesne üzerinde değişiklik yapan popup nesnesine dayalı uygulamanın olduğu bir örnek açılır. Şekil 3.3' te görülen bu uygulamada, Axes nesnesine ve popup nesnesine ait işlev kodları verilmiştir. Bu sayede, nesnelerin Axes nesnesinde değişiklikleri nasıl yaptığı ve çizim işlemlerini nasıl gerçekleştirdiği örneklenmiş olmaktadır.

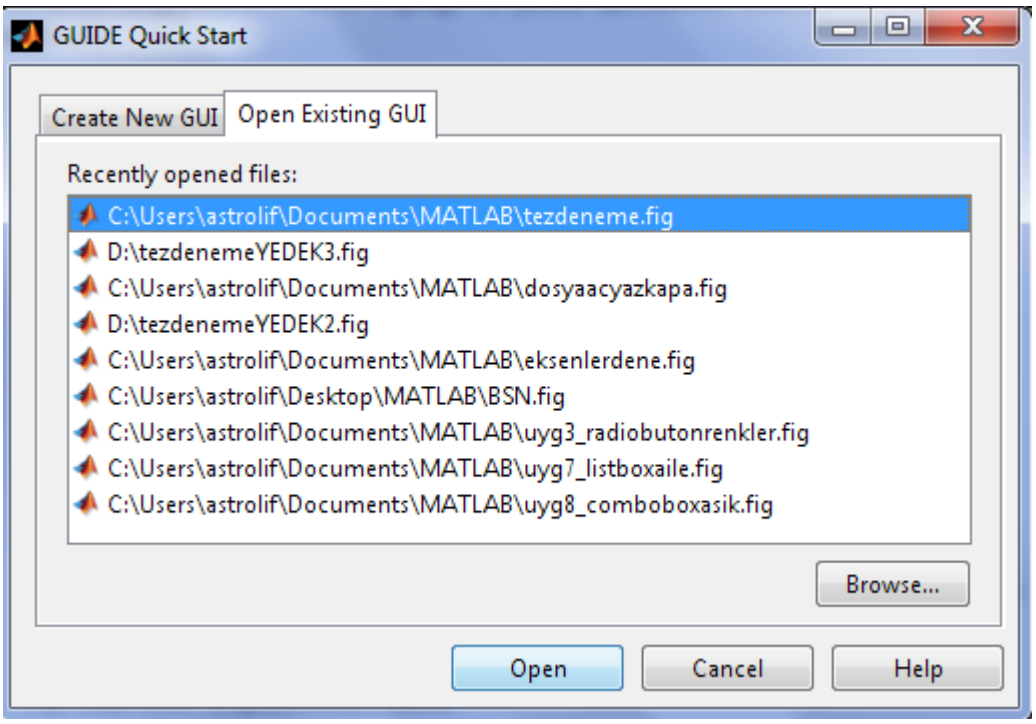
- Modal Question Dialog:



Şekil 3.4: Modal question dialog

GUI' de kullanıcılarla bilgi alış verişi yapan özel etkileşim kutuları tasarlamak mümkündür. Şekil 3.4' te modal question dialog örneği gösterilmiştir.

- Var olan GUI uygulamasını açma:

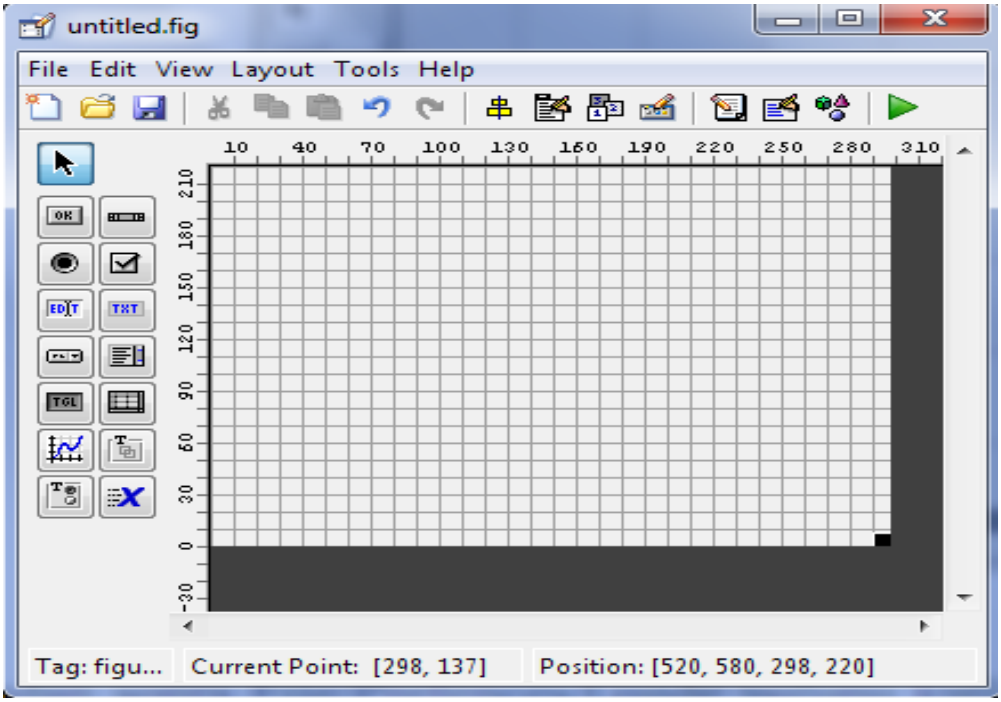


Şekil 3.5: GUIDE Quick Start

Şekil 3.5' te görülen GUIDE quick start penceresinden "Open Existing GUI" sekmesine tıklanarak, çalışma dizininde yer alan GUI uygulamalarının bir listesi görüntülenmektedir. Açılmak istenen uygulama tıklanarak "Open" düğmesine basılır.

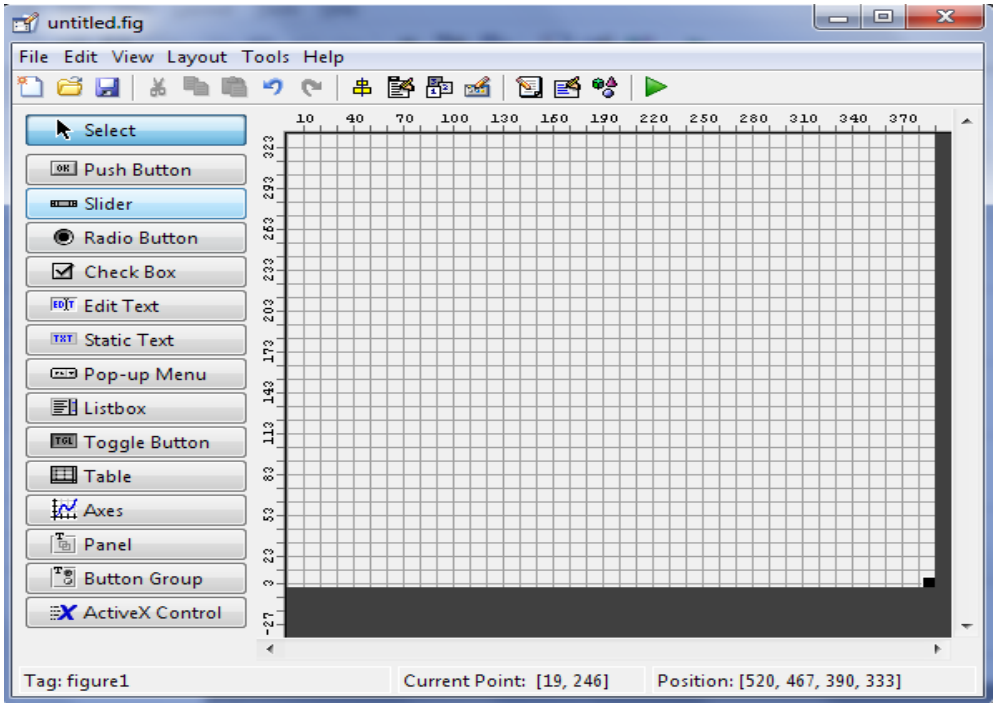
3.5. GUI Çalışma Yüzeyi

Yeni bir GUI tasarımı yapacak isek, Şekil 3.1' deki pencereden "Blank Gui" seçeneğini tıklayarak Şekil 3.6' daki GUIDE layout editor (GUIDE çalışma alanı) penceresine ulaşılır.



Şekil 3.6: GUI çalışma yüzeyi

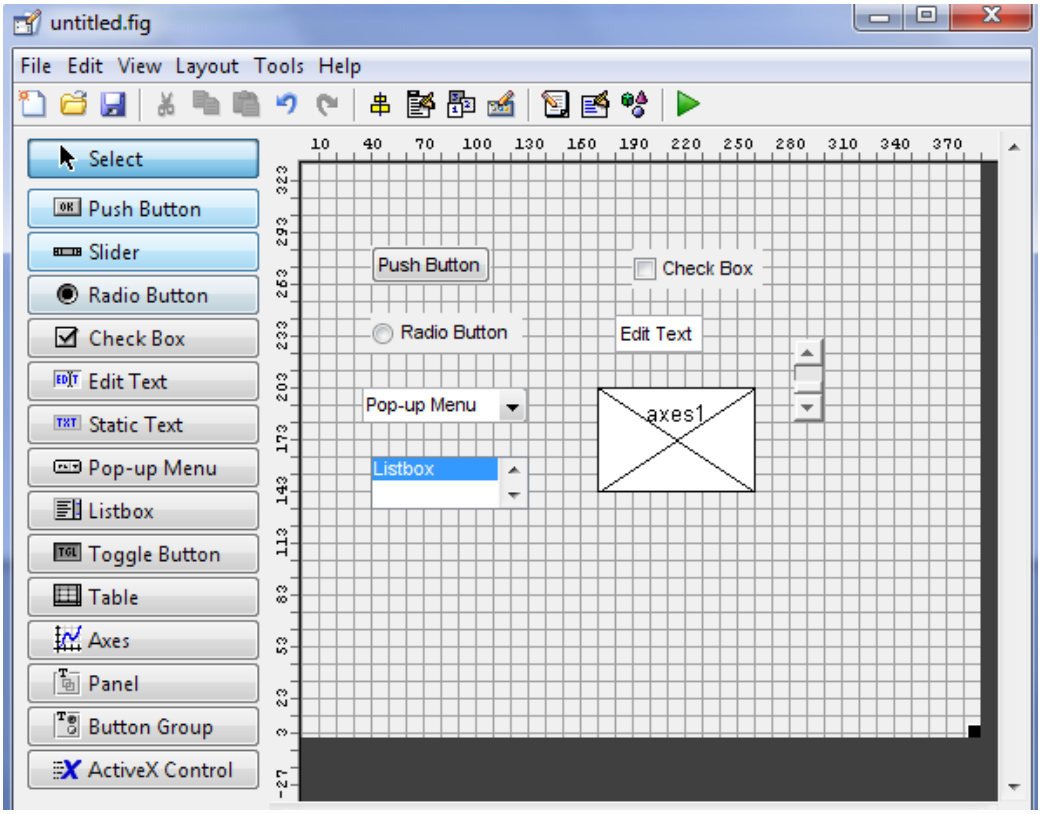
Bu adımdan sonra File/Prefences/Guide yolunu kullanılarak gelen pencereden "Show names in component palette" seçeneğini tıklayıp OK düğmesine basılarak, Şekil 3.6' daki gibi çalışma alanındaki kontrollerin adları ayrıntılı şekilde görülür.



Şekil 3.7: Figür penceresi

3.5.1. Bileşenler çalışma alanına ekleme

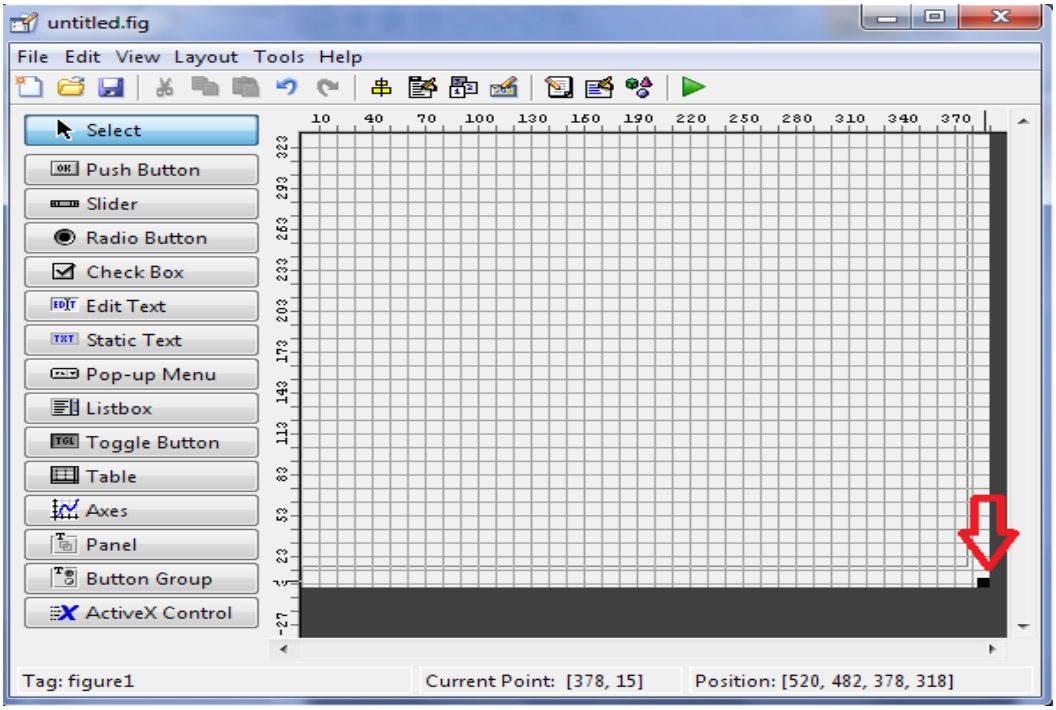
Bunun için sol tarafta bulunan nesne butonlarından istenilen nesneye ait buton tıklanır ve daha sonra çalışma alanında uygun görülen bir noktaya tıklandığında o noktaya ilgili nesne eklenmiş olacaktır. İstenirse çalışma alanındaki bir nesne farenin sol tuşu ile tıklanıp bırakılmadan çalışma alanının herhangi bir yerine sürüklenebilir. Bu durum Şekil 3.8' de de görülmektedir.



Şekil 3.8: Bileşenlerin çalışma alanına eklenmesi

3.5.2. Çalışma alanının boyutunu değiştirmek

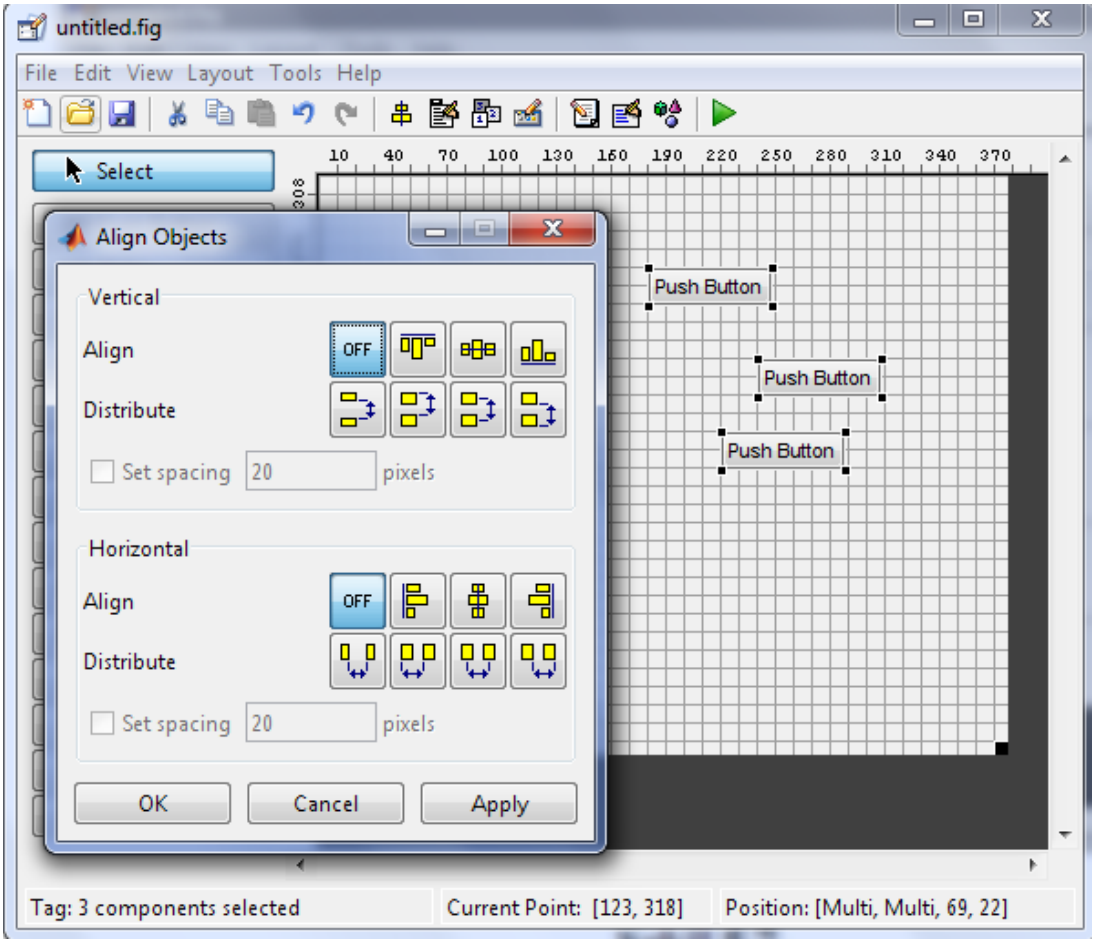
Burada da çalışma alanının sağ alt tarafında bulunan siyah karenin üzerine fare işaretçisi getirilir ve fare işaretçisi durum değiştirdiğinde farenin sol tuşu basılı tutularak çalışma alanı istenilen boyutlarda olacak şekilde düzenleme yapılabilir. Bu durum Şekil 3.9' da gösterilmiştir.



Şekil 3.9: Çalışma alanının boyutunu değiştirme

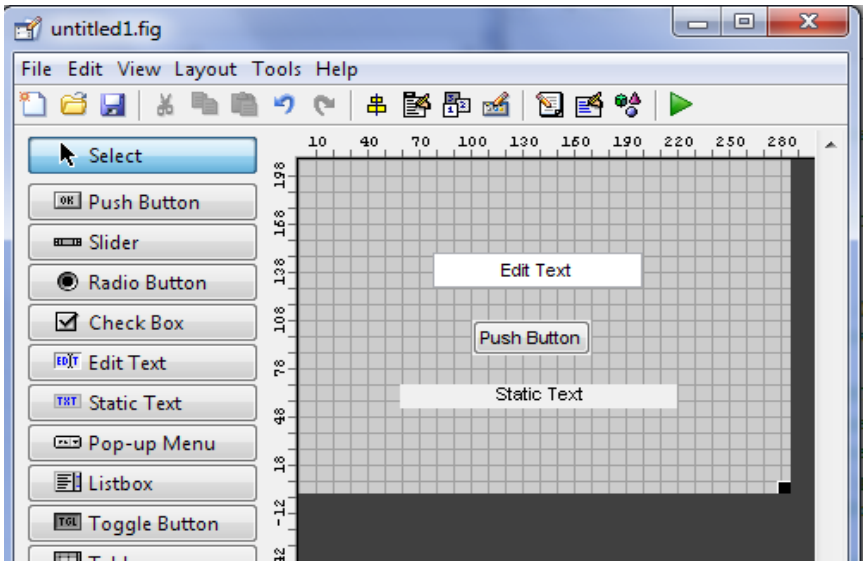
3.5.3. Nesneleri hizalamak

Bu işlemi yapmak için öncelikle hizalanacak nesneler seçilir. Topluca seçmek için çalışma alanında fare işaretçisini herhangi bir yere tıklayıp sürükleyerek açılan kesik kenarlı pencerenin içinde nesneler kalacak şekilde hareket ettirip, hizalanacak nesneler bu çerçeve içinde kalınca farenin sol tuşunu bırakın. Bu şekilde sadece o çerçeve içinde kalan nesneler seçilmiş olacaktır. Ayrıca, nesneleri Ctrl tuşunu basılı tutarak farenin sol tuşu ile teker teker de seçme imkânı bulunmaktadır. Hizalanacak nesneler seçildikten sonra Tools/Align Object yolunu kullanarak Aligment Tool (Hizalama Aracı) penceresini açınız. Şekil 3.10' daki gibi bir ekran ile karşılaşırız. Burada yatay ve dikey hizalamaları kendimize göre butonlardan seçip OK butonuna bastığımız zaman nesnelerimiz hizalanmış olacaktır. Eğer ki hizalama istenilen gibi olmadı ise Ctrl + Z kısayolu ile yapılan işlemler geri alınabilir.



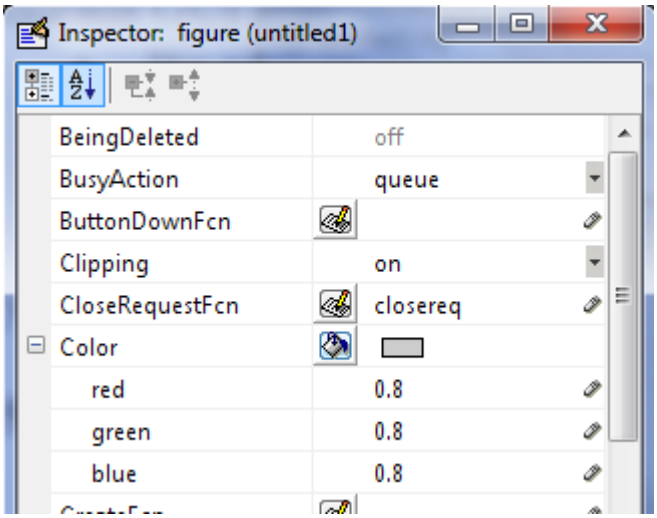
Şekil 3.10: Nesneleri hizalama penceresi

3.5.4. Nesnelere yazı ekleme ve özelliklerini değiştirme



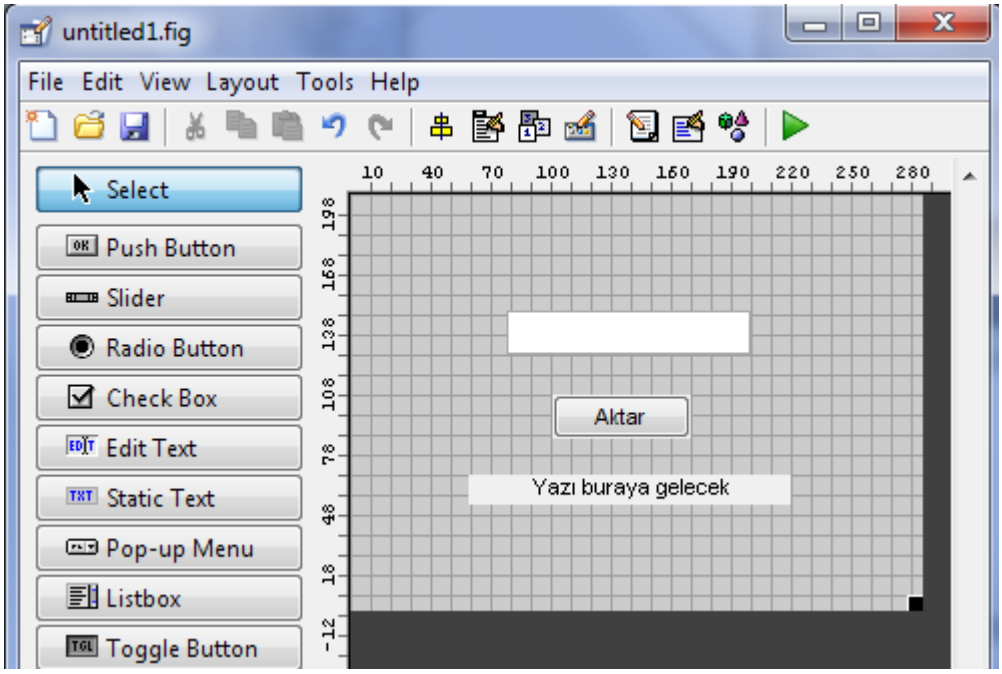
Şekil 3.11: Nesnelere Yazı Ekleme

Nesnelerin özelliklerini değiştirmek istersek ya ilgili nesne farenin sol tuşu ile çift tıklanır ya da ilgili önce seçilip daha sonra View/Property Inspector komutu ile özellikler penceresi (Şekil 3.12) açılır.



Şekil 3.12: Nesne özellikleri penceresi

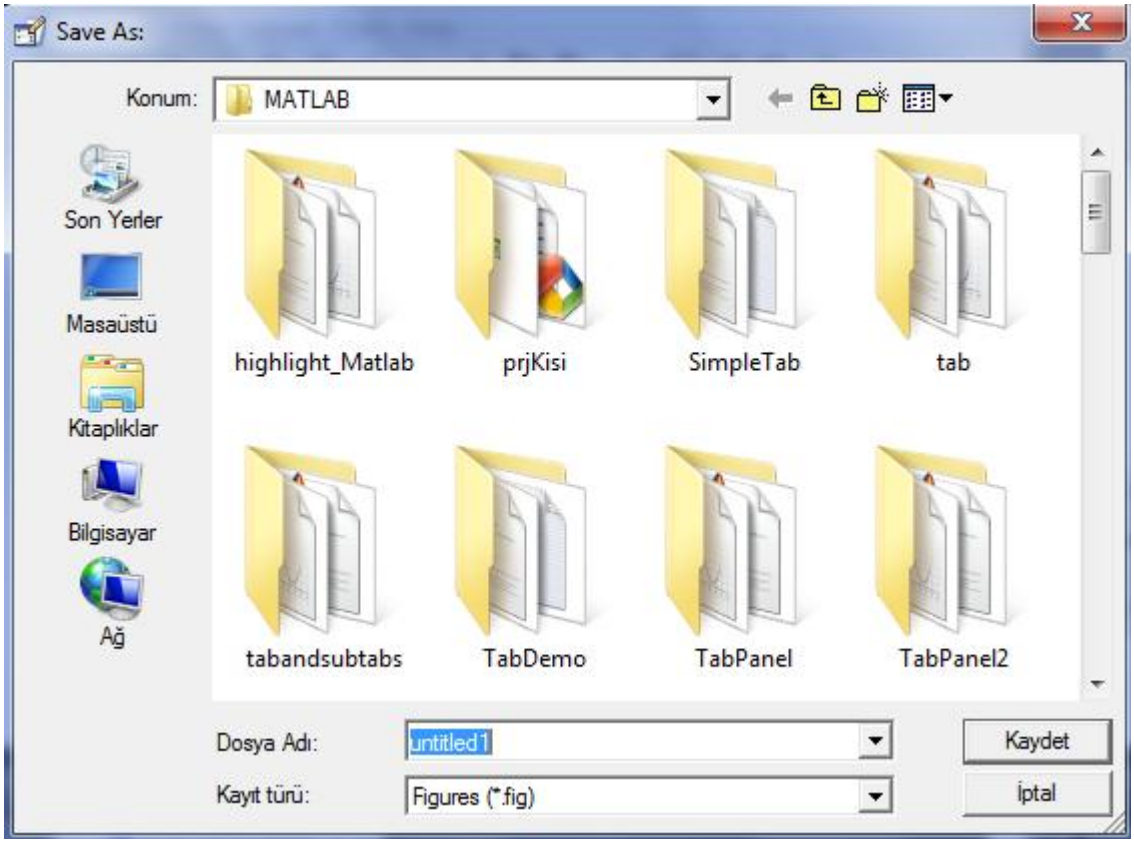
Buradan ilgili nesneleri seçerek (Şekil 3.13'teki örneğe eklenen pushbutton, statictext, vs nesnelerin String özelliği gibi) çeşitli özellikleri değiştirilebilir.



Şekil 3.13: Nesne özellikleri değiştirme

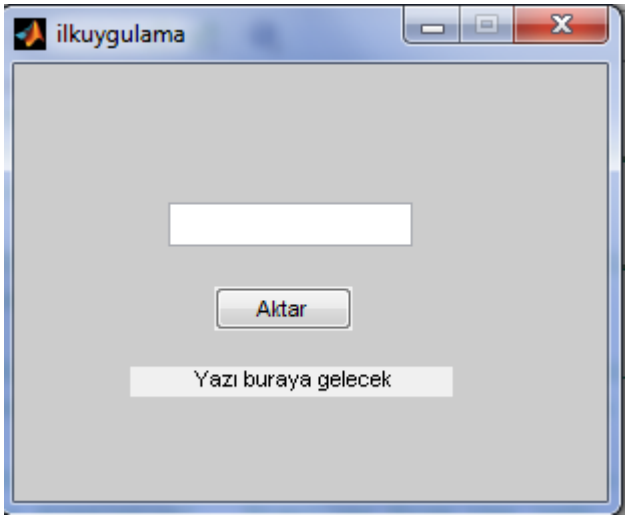
3.5.5. GUI tasarımını kaydetme ve çalıştırma

Bundan sonra bitmiş olan bu GUI arayüzü çalıştırarak görmek için öncelikle Tools/Run yolundan Run (Çalıştır) komutu verilir. Daha sonra gelen pencereden çalışmamnın Run edilebilmesi için kaydedilmesi gerektiğini bildiren bir pencere çıkar. Burada Yes butonuna tıklandıktan sonra MATLAB GUIDE bize tasarımın kaydedileceği dosya ismini soran Şekil 3.14' te görülen bir pencere getirir.



Şekil 3.14: GUI tasarım kaydetme penceresi

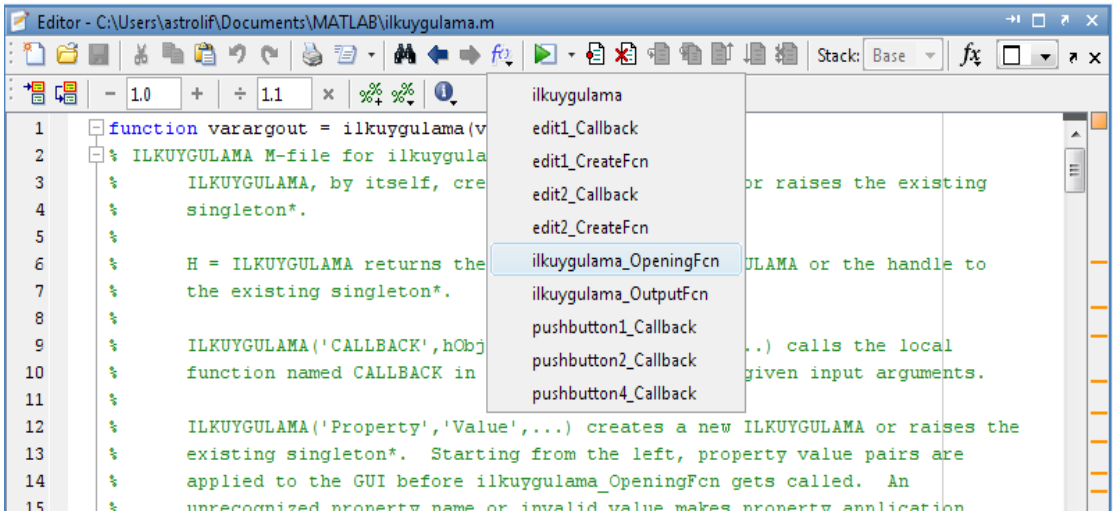
Bu pencereden çalışmamıza bir isim vererek tasarımızı kaydetmiş oluruz. Ardından karşımıza Change the MATLAB Directory gibi bir ekran gelirse burada bu ekranı OK tuşuna basarak kapatabilirsiniz. Bu ekran kaydedilen dosya MATLAB tanımlıyor. Dizinler dışında bir yere kaydedilme sözkonusu olduğunda bizi uyarmaktadır. Sonra da GUI tasarımımızın çalışmasının sonucu Şekil 3.15' te gösterilmiştir.



Şekil 3.15: GUI Uygulamasının Çalıştırılması

3.5.6. GUI ara yüzünün programlanması

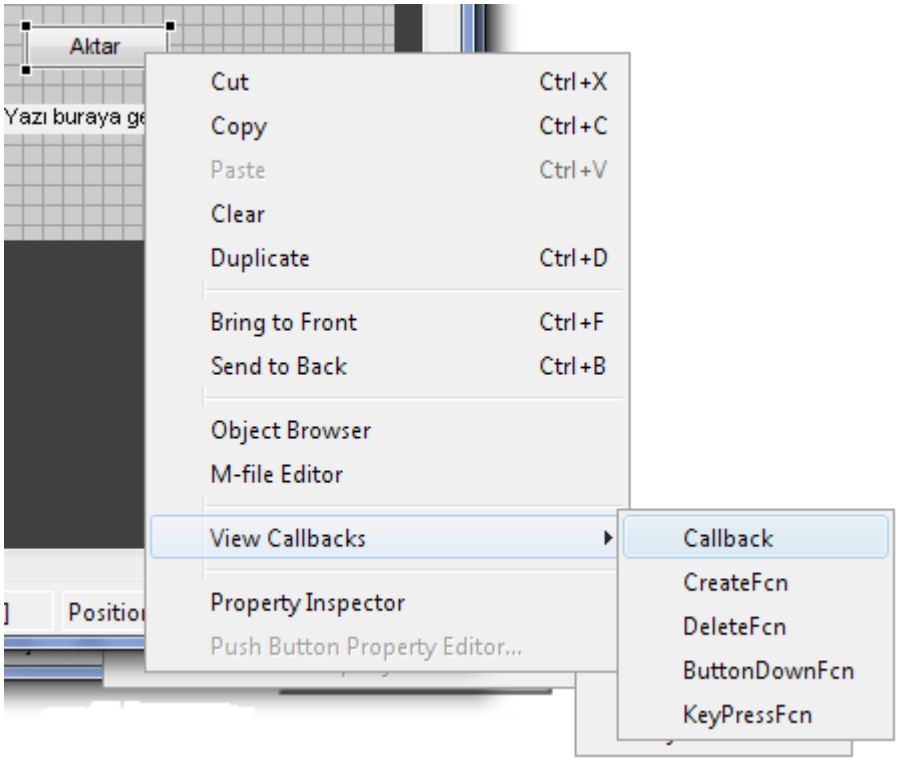
Bir GUI ara yüzünün programlanması demek o çalışmanın kaydedildiği isimle aynı zamanla oluşturulan .m uzantılı dosya içerisine kodlama satırlarının eklenmesi demektir. Bu dosyanın içine görebilmek, değişiklik yapabilmek için GUIDE çalışma ekranı penceresinden View/M-File Editor komutu işletilebilir. Ardından karşımıza Şekil 3.16'daki pencere gelecektir.



Şekil 3.16: GUI Arayüz Programlama Penceresi

Şekil 5.16' daki pencerede hazırlamış olduğumuz GUI tasarımına ait kodlar görünmektedir. Burada pek çok kodun hazır eklenmiş olduğu görülecektir. Bu kodlar otomatik olarak MATLAB GUIDE tarafından eklenmiştir. Biz burada ilgili butona ya da istenilen bir nesneye ait callback isimli alt program parçalarına ilgili kodları

yazacağız. Bir nesneye ait callback in bulunduğu satıra gitmek için araç çubuğunda yer alan fx simgeli butona tıklanır ve açılan listeden ilgili nesneye ait callback in ismi seçilir. Bu durum yukarıdaki pencerede de görülmektedir. Ayrıca, GUIDE çalışma ekranından da direk istenilen bir callback satırına gidilebilir. Bunun için ilgili nesne üzerinde sağ tıklanır ve açılan pencereden View Callbacks menüsünden ilgili callback tıklanması ya da ilgili nesne seçilip View/View Callbacks yolu üzerinden açılan listeden gidilmek istenilen callback tıklanması yeterlidir.



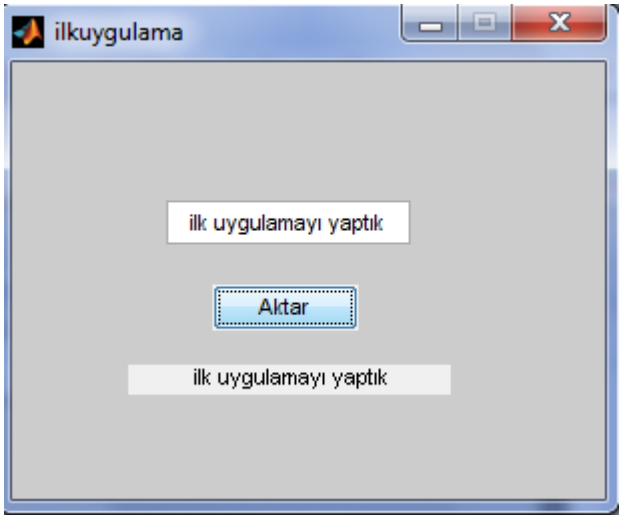
Şekil 3.17: Figur yüzeyindeki nesnelerin callback' lerine ulaşılması

Aşağıda butona basıldığında çalışan kod gösterilmiştir;

```
function varargout = ilkuygulama(varargin)
% butona basıldığında çalışacak olan kodlar
function pushbutton4_Callback(hObject, eventdata, handles)
str1=findobj(gcbf,'Tag','text2');
str2=get(handles.edit2,'String');
set(str1,'String',str2);
```

Callback' lerdeki kodlar, her nesnenin üzerine sağ tıklayıp View Callback ve ilgili Callback satırına gidilerek ayrı ayrı da yazılabilirdi. Kodlama satırlarında % işareti

ile başlayan satırlar açıklama satırları olup, bu satırlar herhangi bir komut olarak görülmezler sadece açıklama amacı taşırlar. Tüm işlemler tamamladığına göre Tools/Run komutu ile GUI uygulamamızı çalıştırdığımızda Şekil 3.18' teki sonuç görülür.



Şekil 3.18: "İlk uygulama" nın çalışması

Burada yazılan kod parçalarını (callback rutinlerini) kısaca açıklayalım.

function varargout = ilkuygulama(varargin)

Yukarıdaki function bloğu GUIDE tarafından otomatik olarak oluşturulur. Burada GUI uygulamasına komut satırından gönderilen parametrelerin alınması ve GUI uygulaması çalıştıktan sonra bir fonksiyon olarak dışarıya gönderilecek parametrelerin tanımlanması ile ilgili kod satırları mevcuttur.

function ilkuygulama_OpeningFcn(hObject, eventdata, handles, varargin)

Bu fonsiyon GUI arayüzü ekrana gelmeden (visible olmadan) hemen önce çalıştırılacak kodları içerir. Örneğin böyle bir callback bir GUI uygulaması çalışmadan önce initialization işlemlerinin yapılması ya da bazı GUI nesne özelliklerinin değiştirilmesi istendiğinde kullanılabilir.
Ayrıca, varargin giriş parametresi kullanılarak da MATLAB komut satırından girilen parametre değerleri GUI uygulaması içinde kullanılmak üzere bu blokta alınır.

function varargout = ilkuygulama_OutputFcn(hObject, eventdata, handles)

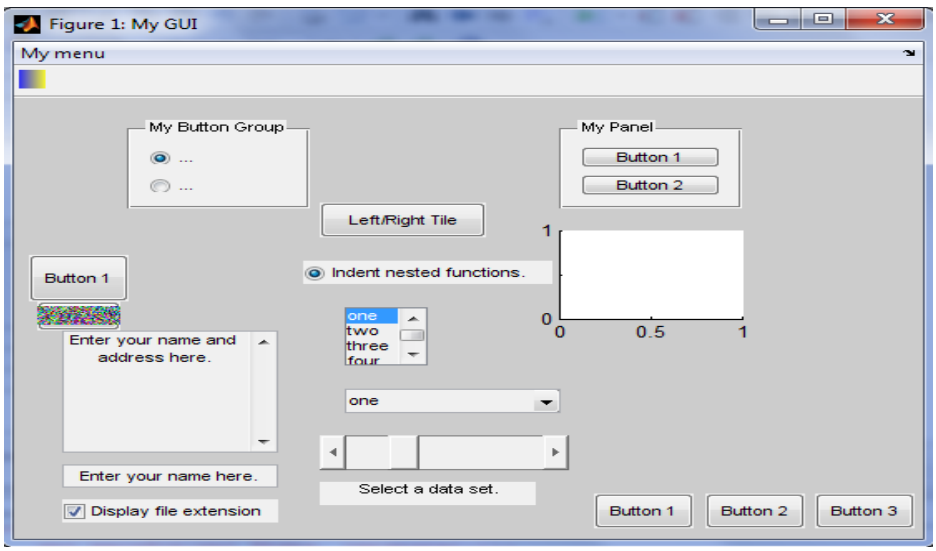
Bu fonksiyon bloğu bir GUI uygulaması hafızadan silinip programı sonlandırılmadan hemen önce (destroy edilmeden önce) çalıştırılacak komutlar içerir. Ayrıca, komut satırına gönderilecek çıkış parametre değerleri de bu blok tarafından varargout değişkeni kullanılarak işleme konulur.

function pushbutton4_Callback(hObject, eventdata, handles)

Bu callback bloğu pushbutton4 isimli buton (istenilse bu buton ismi butonun Tag özelliğine özellikler penceresinden yeni bir isim verilerek de değiştirilebilir.) ki burada Aktar stringine sahip olan buton tıklandığı zaman edittext'e girilen verileri statictext'e aktaracak olan komutları içerir.

3.6. Programlama Yoluyla Nesnelerin Eklenmesi

M dosyasının içeriğine aşağıda belirtilen kodları eklediğimizde GUI arayüz Şekil 3.19' daki gibi görülecektir.



Şekil 3.19: Programlama Yoluyla Nesnelerin Eklenmesi

function varargout = mygui(varargin) mygui icin fonksiyon tanımı

```
fh = figure('Visible','on','Name','My GUI',...      bu satırlar ekrana belirtilen boyut
'Position',[360,550,550,300]);                      ve konumda figure (GUI yüzeyi)  getirme

cbh = uicontrol(fh,'Style','checkbox',...   GUI yüzeyine checkbox nesnesi
'String','Display file extension',...       ekleme
'Value',1,'Position',[30 15 130 20]);

eth = uicontrol(fh,'Style','edit',...   GUI yüzeyine edit kutusu ekleme
'String','Enter your name here.',...
'Position',[30 45 130 20]);

eth = uicontrol(fh,'Style','edit',...                GUI yüzeyine çok satırlı edit kutusu
'String','Enter your name and address here.',... ekleme  (çünkü max-min>1 durumu)
'Max',2,'Min',0,...
'Position',[30 75 130 105]);

lbh = uicontrol(fh,'Style','listbox',...             GUI yüzeyine liste kutusu ekleme
'String',{'one','two','three','four'},...            (elemanlari 'one','two','three','four' )
'Value',1,'Position',[200 150 50 50]);

pmh = uicontrol(fh,'Style','popupmenu',...     GUI yüzeyine popup menü ekleme
'String',{'one','two','three','four'},...            (elemanlari 'one','two','three','four' )
'Value',1,'Position',[200 110 130 20]);

pbh1 = uicontrol(fh,'Style','pushbutton','String','Button 1',.. GUI yüzeyine push buton
'Position',[10 205 60 40]);                                     ekleme

img(:,:,1) = rand(16,64);            rasgele sayılardan oluşan dizi tanımı
img(:,:,2) = rand(16,64);
img(:,:,3) = rand(16,64);

pbh2 = uicontrol(fh,'Style','pushbutton',...       GUI yüzeyine push buton ekleme
'Position',[15 180 50 25],'CData',img);
```

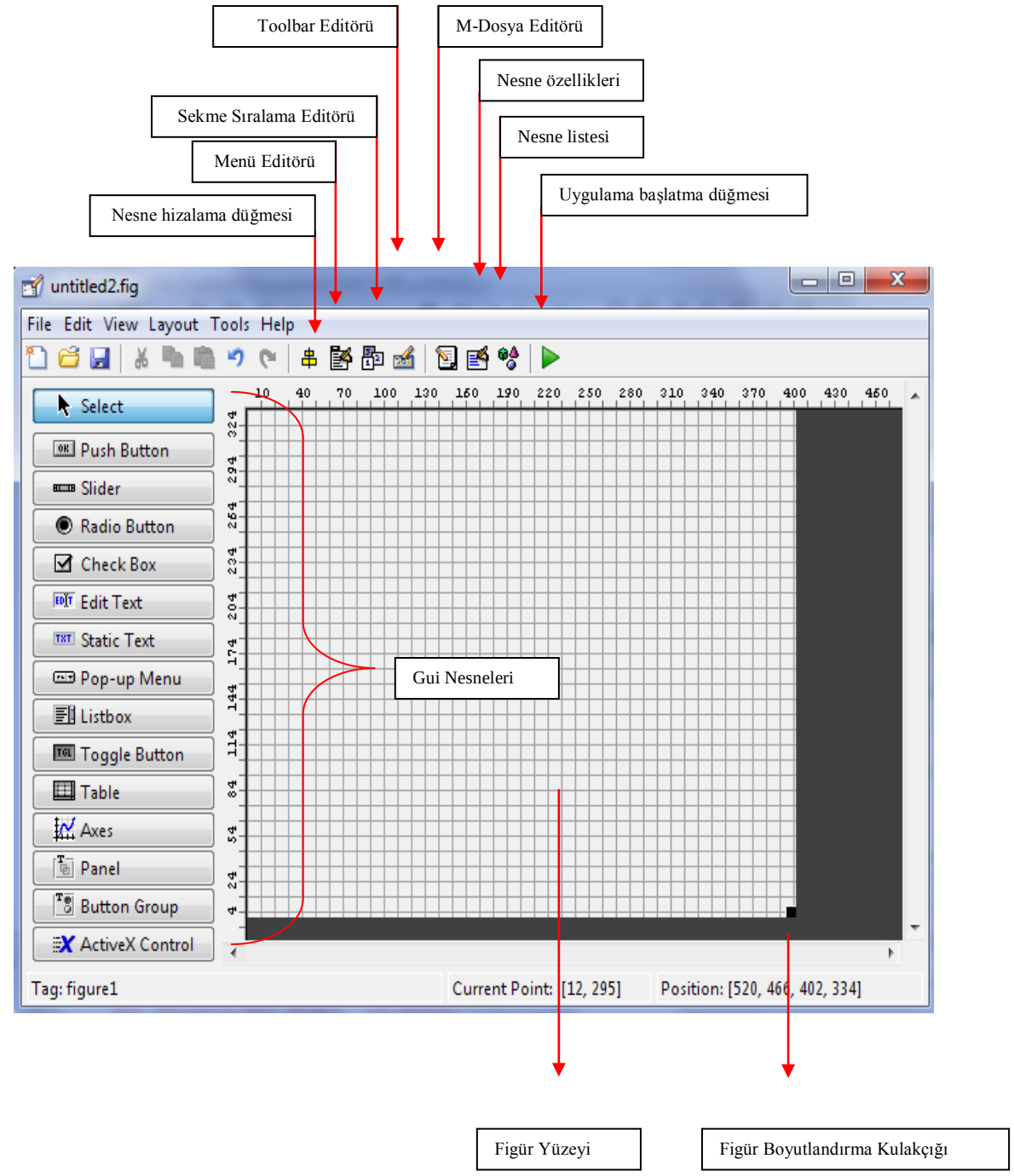
```
rbh = uicontrol(fh,'Style','radiobutton',...          GUI yüzeyine radio buton ekleme
'String','Indent nested functions.',...
'Value',1,'Position',[175 220 150 20]);
sh = uicontrol(fh,'Style','slider',...                GUI yüzeyine kaydırıcı ekleme
'Max',100,'Min',0,'Value',25,...
'SliderStep',[0.05 0.2],...
'Position',[185 60 150 30]);

sth = uicontrol(fh,'Style','text',...                 GUI yüzeyine static text kutusu
'String','Select a data set.',...                     ekleme
'Position',[185 30 130 20]);

tbh = uicontrol(fh,'Style','togglebutton',...         GUI yüzeyine toggle (çift durumlu)
'String','Left/Right Tile',...    buton ekleme
'Value',0,'Position',[185 260 100 30]);

ph = uipanel('Parent',fh,'Title','My Panel',...       GUI yüzeyine panel ekleme
'Position',[.60 .75 .2 .2]);

pbh3 = uicontrol(ph,'Style','pushbutton','String','Button 1',... ph paneline push buton
'Units','normalized',...                                          ekleme

'Position',[.1 .55 .8 .3]);
pbh4 = uicontrol(ph,'Style','pushbutton','String','Button 2',... ph paneline 2.push buton
'Units','normalized',...                                          ekleme
'Position',[.1 .15 .8 .3]);
bgh=uibuttongroup('Parent',fh,'Title','MyButtonGroup',...     GUI yüzeyine
butongrubu ekleme
'Position',[.125 .75 .2 .2]);

rbh1 = uicontrol(bgh,'Style','radiobutton','String','Red',... bgh grubuna radio buton
'Units','normalized',...                                      ekleme
'Position',[.1 .6 .3 .2]);
```

```
rbh2 = uicontrol(bgh,'Style','radiobutton','String','Blue',...  bgh grubuna 2. radio buto
'Units','normalized',...                                        ekleme
'Position',[.1 .2 .3 .2]);

ah = axes('Parent',fh,'Position',[.60 .50 .2 .2]);   GUI yüzeyine grafik çizim
                                                     alanı ekleme

b1 = uicontrol(fh,'Posit',[330 80 60 30],'String','Button 1');  GUI yüzeyine buton 1
b2 = uicontrol(fh,'Posit',[350 50 60 30],'String','Button 2');  GUI yüzeyine buton 2
b3 = uicontrol(fh,'Posit',[310 10 60 30],'String','Button 3');  GUI yüzeyine buton 3

align([b1 b2 b3],'Right','None');           b1, b2 ve b3 nesnelerini sağa hizala
%align([b1 b2 b3],'Left','Distribute');     b1, b2 ve b3 nesnelerini sola dağınık hizala
align([b1 b2 b3],'Center','Fixed',7);       b1, b2 ve b3 nesnelerini ortala
align([b1 b2 b3],'Fixed',5,'Bottom');       b1, b2 ve b3 nesnelerini aşağıyı doğru hizala

set(fh,'MenuBar','figure');     standart araç çubuğunun gösterilmesi
set(fh,'MenuBar','none');       standart araç çubuğunun gizlenmesi

mh = uimenu(fh,'Label','My menu');    GUI yüzeyi menüsüne My menu eklenm
eh1 = uimenu(mh,'Label','Item 1');    mh menüsüne alt menü tanımlanması
eh2 = uimenu(mh,'Label','Item 2','Checked','on');       mh menüsüne alt menü
tanımlanması

set(eh2,'Separator','on');     mh2 menu seceneğinin üzerine ayıraç kon.

seh1 = uimenu(eh1,'Label','Choice 1','Accelerator','C',...  eh1'e kisayol (Ctrl+C)
tanımı
'Enable','off');       pasif yapılması
seh2 = uimenu(eh1,'Label','Choice2','Accelerator','H');  eh2'ye kisayol (Ctrl+H)
tanımı
th = uitoolbar(fh);     GUI yüzeyine araç çubuğu ekleme
a = [.20:.05:0.95];     bu ve alt satırlarla rasgele renkleri
```

```
img1(:,:,1) = repmat(a,16,1);     temsil etmek üzere dizilerin tanımlanması
img1(:,:,2) = repmat(a,16,1);
img1(:,:,3) = repmat(flipdim(a,2),16,1);

pth = uipushtool(th,'CData',img1,...   th araç çubuğuna push buton ekleme
'TooltipString','My push tool',...
'HandleVisibility','off');    handlevisibility komut satırından erişimi ayarlar

img2 = rand(16,16,3);      rastgele renk dizisi tanımlama
tth = uitoggletool(th,'CData',img2,'Separator','on',...  araç çubuğuna ayırıcı ekleme
'TooltipString','Your toggle tool',...
'HandleVisibility','off');

oldOrder = allchild(th);              bu ve aşağıdaki satırlar ile nesnelerin tab tuşu
newOrder = flipud(oldOrder);      ile geçiş sırası ayarlanmakta
set(th,'Children',newOrder);
delete(tth);                  tth handleini tutan nesnenin (burada araç çubuğu) silinmesi
end                           fonksiyon sonu
```

3.7. GUIDE Aracının İncelenmesi

Bir önceki konularımızda da bu aracı kısaca incelemye çalıştık. Burada GUIDE aracı detaylı olarak incelenecektir. MATLAB komut satırından "guide" komutunu yazdığımızda ve gelen pencereden boş (blank) bir GUI tasarımını seçtiğimizde Şekil 3.20' deki pencere ile karşılaşılır.



Şekil 3.20: Guide penceresinin yapısı

Bu ekrandaki araçlar ile ilgili açıklama aşağıda verilmiştir.

3.7.1. Layout editor

GUIDE çalışma alanı ve penceresidir. Bu ekran ile GUI yüzeyine paletten seçilen ilgili nesneler eklenebilir ya da diğer araçlar ile program kodlarının yazılması, nesnelerin GUI yüzeyi üzerinde hizalanması, tab tuşu geçiş sırasının değiştirilmesi gibi pek çok işlem gerçekleştirilebilir.

3.7.2. Figure resize tab

Bu araç GUI çalışma alanının boyutlandırılmasını sağlar. Fare işaretçisi bu alan üzerine getirildiğinde konum değiştirecektir. Bu anda farenin sol tuşu tıklanıp ileri geri hareket ettirilerek GUI yüzey alanının boyutları değiştirilebilir.

3.7.3. Menu editor

GUI uygulamasına istenilirse File, edit… vb. gibi menü içeren programlarda olduğu gibi bir menü eklenmesi ve eklenen menü ile ilgili işlemlerin yapılması bu araç vasıtasıyla sağlanır.

3.7.4. Align objects

Bu araç sayesinde GUI çalışma alanına eklenilen nesnelerin yatay ya da dikey olarak hizalanması işlemleri gerçekleştirilir.

3.7.5. Tab order editor

Tab Order Editor kullanılarak GUI yüzeyindeki nesnelerin birinden diğerine tab tuşu ile geçiş sırası (örneğin bir buton seçili ve aktif iken bir başka butona ya da bir liste kutusuna tab tuşu kullanılarak geçilmesi gibi.) değiştirilebilir.

3.7.6. Property inspector

Bu pencere sayesinde de GUI uygulamasına eklenen nesnelerin özellikleri değiştirilebilir ya da var olan özelliklerinin ve değerlerinin neler olduğu gözlemlenebilir.

3.7.7. Object browser

Bu araç ile tasarımcı GUI uygulamasına eklemiş olduğu nesnelerin ve isimlerinin neler olduğu genel hali ile bakabilir.

3.7.8. Run

Bu buton yardımı ile de hazırlanmış olan bir GUI uygulaması çalıştırılabilir. Bu şekilde tasarımcı hazırlamış olduğu GUI' yi test etme imkânına sahiptir.

3.7.9. M-File editor

Hazırlanmış olan GUI uygulaması ile ilgili komutları görebilmek ve üzerinde değişiklik yapabilmek için bu araç kullanılır.

3.7.10. GUIDE tercihleri

Bu tercihleri görebilmek için GUIDE ekranında File menüsünden Preferences komutu çalıştırılır.

3.8. GUI Nesnelerinin Açıklanması

Şimdi bu nesnelerin sırasıyla özellikleri ile ilgili bilgiler verilecek ve nasıl programlanacağı gösterilecektir.

3.8.1. Push button

Butonun tıklanması ile icra ettirilmek istenen komutlar bu nesnenin callback' i altına yazılır. Buton tıklandığında komutlar işletilir.

3.8.2. Slider

Kullanıcıdan bir giriş değerini kaydırılmak suretiyle kolaylıkla alınmasına imkân veren bir nesnedir. Başında 0 ve sonunda 1 değerine sahip olan ve arası değerlerin de bulunduğu sayılar kümesidir. Sürükle bırak yöntemi ile yüzeye yatay ve dikey olarak yerleştirilebilir, yüzeye tek tıklandığında ise dikey şekliyle eklenir.

3.8.3. Radio button

Kullanıcıya birden fazla seçenek sunulup, kullanıcıdan bu seçeneklerden sadece birinin seçilmesinin istendiği durumlar için kullanılan nesnedir.

3.8.4. Check box

Kullanıcıya birden fazla seçenek sunulup, birden fazla seçeneğin işaretlenebileceği durumlar için kullanılan nesnedir.

3.8.5. Edit text

Kullanıcıya herhangi bir bilgi verme ya da bulunan bir sonuç veya değeri gösterme amacıyla sıklıkla kullanılan bir nesnedir. Edit text ile alınan tüm veriler karakter tipindedir. Sayısal veri olarak kullanılmak istendiğinde Matlab fonksiyonları ile sayıya dönüştürülebilir.

3.8.6. Static text

Uygulamalarda açıklama kutucuğu olarak kullanılabildiği gibi sonuçların görüntülendiği ekranlar olarak da kullanılması mümkün olan nesnedir.

3.8.7. Listbox

Kullanıcıya bilgi verme amacıyla kullanılabileceği gibi bir değeri listeden seçmek amacıyla da kullanılan sabit bir liste kutusu niteliğinde kullanılan bir nesnedir.

3.8.8. Pop-up menu

Listbox ile aynı çalışma ilkesine sahiptir. Fakat eldeki seçenek sayısı çok sayıda ise, seçenekler uzama menüsü halinde bu nesne içinde belirtilir.

3.8.9. Toggle button

Bu nesnenin özelliği tıklandığında basılı kalmasıdır. Tekrar tıklandığında ise normal haline döner. Bu özelliği sayesinde toggle buton basılı konumdayken farklı işlem grubu, basılı değilken farklı işlem grubu uygulanabilir.

3.8.10. Table

Verilerin satır ve sütunlarla tablo şeklinde alınmasını veya gösterilmesini sağlayan nesnedir.

3.8.11. Axes

Figür yüzeyinde grafik, resim veya hareketli görüntü gösterilmesini sağlayan nesnedir.

3.8.12. Panel

GUI yüzeyi nesnelerinin kullanıcıya daha anlamlı ve güzel gözükmesini sağlayan, ayrıca tasarımcıya GUI dizaynında kolaylık sunan bir nense olup, GUI yüzeyi nesnelerinin gruplanması ve bir arada gösterilmesi amacıyla kullanılır.

3.8.13. Button group

Radio veya toggle tipteki buton nesnelerinin bir arada kullanılarak kullanıcının birden fazla seçenekten sadece bir tanesini seçmesini sağlamak amacıyla kullanılan bir nesnedir.

3.8.14. ActiveX control

MATLAB GUI tasarımları sadece yukarıda belirtilen nesneler ile sınırlı değildir. Tasarımcı ve programcı ayrıca, ActiveX adı verilen ve değişik alternatifi olan nesnelerin kullanılmasına da imkân verir. Böylece hem tasarımcı hem tasarlanılacak GUI ara yüzünün kullanımı bakımından kullanıcıya esneklik sağlanmış olunur.

3.9. GUI Nesnelerinin Programlanması

3.9.1. Push button ve static text uygulaması

Şekil 3.21' de "saydır" butonuna basınca statictext' de tıklanma sayısını gösteren bir uygulama gerçekleştirilmektedir.

Şekil 3.21: Push button ve static text uygulaması

Figür yüzeyinde bulunan pushbutton nesnesine işlev eklemek için, bu nesne üzerine sağ tıklanıp açılan menüden "View Callbacks/ Callback" komutu seçilir.
Bu seçimden sonra MATLAB; M-Fonksiyon'da programcıyı, callback ifadesinin olduğu satıra yönlendirir.

```
function pushbutton1_Callback(hObject, eventdata, handles)

global sayac;
%sayac adlı değişken tanımlanır.

if isempty(sayac)
  sayac = 0;
end
%sayac değişkeninin başlangıç değeri 0 olarak atanır.
sayac=sayac+1;
%değer buton her tıklandığında sayac 1 arttırılır.

str1=sprintf('sayaç:%d',sayac);
%sayaç: kelimesiyle,sayac değişkeninin degeri birleştirilerek string bir dizi oluştulur.str1 değişkeninde saklanır.

str2=findobj(gcbf,'Tag','text1');
%üzerinde çalışılan figürde,Tag'i text1 olan nesne bulunarak str2 değişkenine atanır.

set(str2,'String',str1);
%str2 değişkenindeki nesnenin(yani Tag'i text1 olan static text nesnesi), string özelliği str1 değişkeninde tutulan sayaç değeriyle değiştirilir.
```

3.9.2. Toggle button

Bir toggle buton çift durumlu çalıştığından, nesnede geçerli buton konumunu öğrenmek ve kullanabilmek için aşağıdaki komut satırları kullanılmalıdır.

```
function togglebutton1_Callback(hObject, eventdata, handles)
buton_durum = get(hObject,'Value');
if button_state == get(hObject,'Max')
% Toggle buton basıldığında yapılacak işlemler
...
```

```
elseif button_state == get(hObject,'Min')
% Toggle buton basılmadığı durumda yapılacak işlemler
...
End
```

3.9.3. Radio button

Şekil 3.22' deki uygulamada görüldüğü gibi, çoklu seçimlerden sadece bir tanesini seçmeyi sağlar.

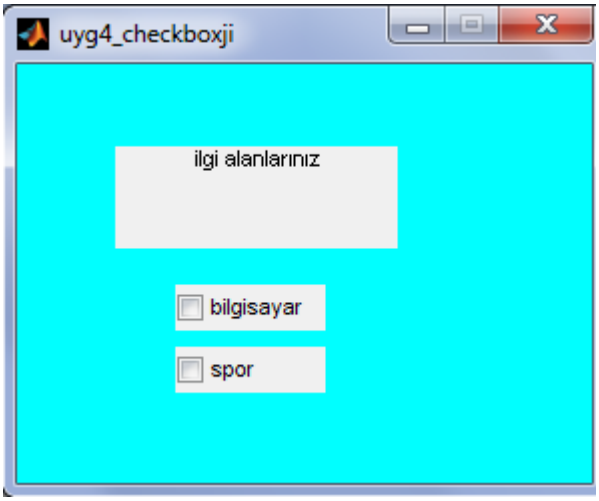
Şekil 3.22: Radiobutton uygulaması

```
function radiobutton1_Callback(hObject, eventdata, handles)
% Hint: get(hObject,'Value') returns toggle state of radiobutton1
 durum1=get(gcbo,'Value');
 str1=findobj(gcbf,'Tag','text1');
renk=get(str1,'BackgroundColor');

rdb2=findobj(gcbf,'Tag','radiobutton2');
set(rdb2,'Value',0);

rdb3=findobj(gcbf,'Tag','radiobutton3');
set(rdb3,'Value',0);
```

```
if durum1==1
   set(str1,'BackgroundColor','red');
   set(str1,'String','KIRMIZI RENGİ SEÇTİNİZ');
else
   set(str1,'String','seçim yapmadınız');
end
```

3.9.4. Checkbox

Checkbox nesnesinin konum kontrolü de radio butonlarınkine benzer şekildedir. Farklı olarak checkbox' lar kullanıcıya birden fazla seçim imkanı vermektedir. Şekil 3.23' te olduğu gibi seçeneklerden birden fazlası seçilebilir.



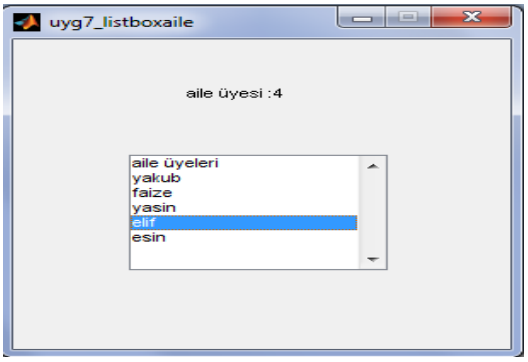
Şekil 3.23: Checkbox uygulaması

```
global durum1
global durum2
str1=findobj(gcbf,'Tag','text1');

if isempty(durum2)
   set(durum2,'Value',0);
end

durum1=get(gcbo,'Value');
if durum1==1 & durum2==1
```

```
    set(str1,'String','ikisini de seçtiniz');
elseif durum1==0 & durum2==0
    set(str1,'String','en az bir seçim yapınız');
elseif durum1==1 & durum2==0
    set(str1,'String','vj yi seçtiniz');
else
    set(str1,'String','ei yi seçtiniz');
end
```

3.9.5. Listbox

Listbox nesnelerinin liste tipindeki string içeriğinin kullanılabilmesi için bu nesnelerin Value ve String özellikleri birlikte kullanılır. Şekil 3.24' teki uygulamanın kodları aşağıda görülmektedir.

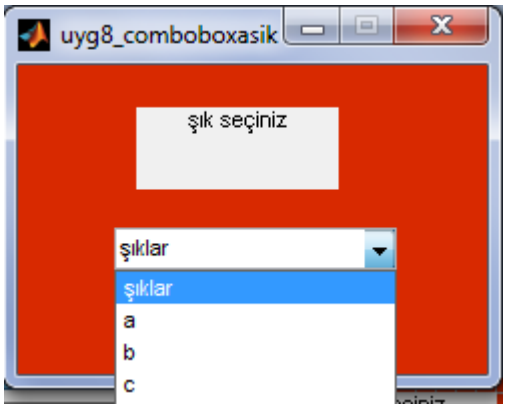


Şekil 3.24: Listbox uygulaması

```
function listbox1_Callback(hObject, eventdata, handles)
% Hints: contents = get(hObject,'String') returns listbox1 contents as cell array
%        contents{get(hObject,'Value')} returns selected item from listbox1

str1=findobj(gcbf,'Tag','text1');

deger=get(gcbo,'Value');

str2=['aile üyesi :' num2str(deger-1)]; %karakter dizisi oluşturuyoruz
set(str1,'String',str2);
```

3.9.6. Pop-up menu

Pop-up menü nesnelerinde seçilen bir öğenin hangisi olduğu anlamak için bu nesnelerin Value özelliğinden yararlanılır. Mantığı listbox ile aynıdır. Fakat bu bileşen daha uzun listeleri saklamak için kullanılır. Şekil 3. 25' te örnek bir uygulama görülmektedir.

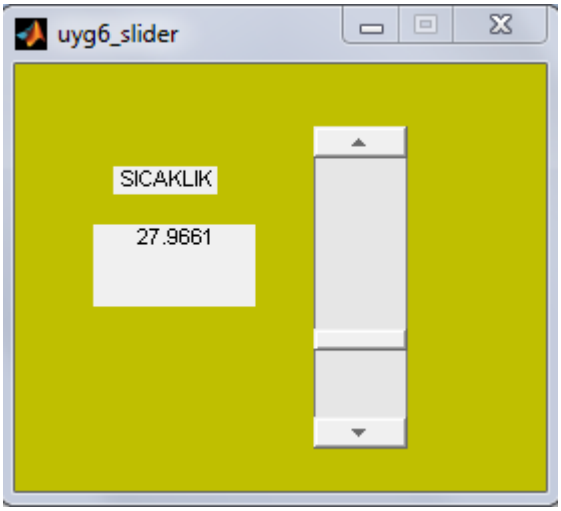


Şekil 3.25: Pop-up menü uygulaması

```
function popupmenu1_Callback(hObject, eventdata, handles)
% Hints: contents = get(hObject,'String') returns popupmenu1 contents as cell array
%        contents{get(hObject,'Value')} returns selected item from popupmenu1
 str1=findobj(gcbf,'Tag','text1');
durum=get(gcbo,'Value');
str2=['seçtigin şık:' num2str(durum-1)];
set(str1,'String',str2);
```

3.9.7. Slider

Bir kaydırıcı (slider) nesnesinin geçerli değerini program yoluyla okumak için gerekli komut satırları şöyle olmalıdır. Şekil 3.26' da slider' ın aldığı değer statictext içerisinde gösterilmektedir.

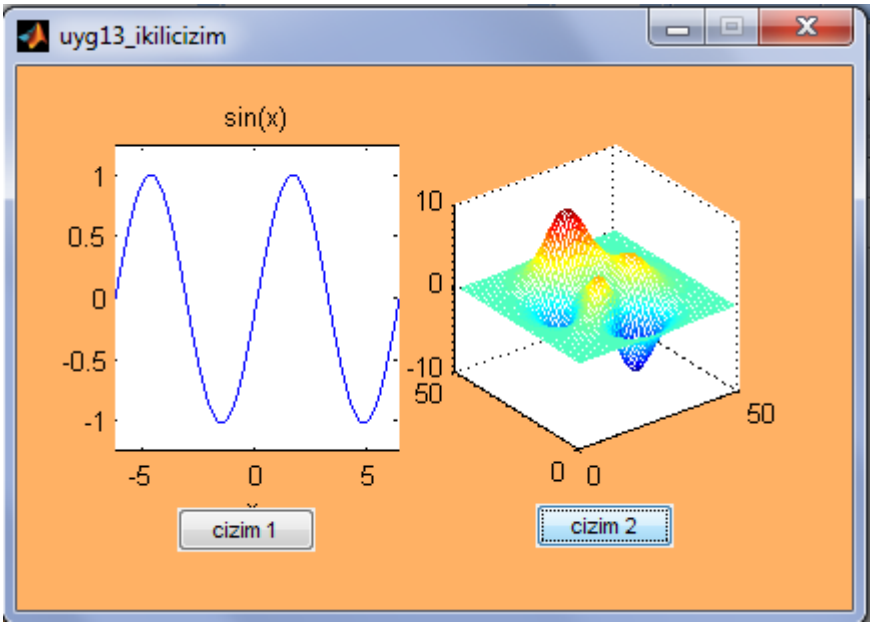


Şekil 3.26: Slider uygulaması

```
function slider1_Callback(hObject, eventdata, handles)
% Hints: get(hObject,'Value') returns position of slider
%        get(hObject,'Min') and get(hObject,'Max') to determine range of slider

str1=findobj(gcbf,'Tag','text1');

deger=get(gcbo,'Value');

set(str1,'String',deger*100);
```

3.9.8. Axes

Grafik çizimlerinin kullanıcıya sunulmasında sıklıkla kullanılan bir nesnedir. Şekil 3.27' de iki farklı axes' te iki farklı çizim gösterilmektedir.

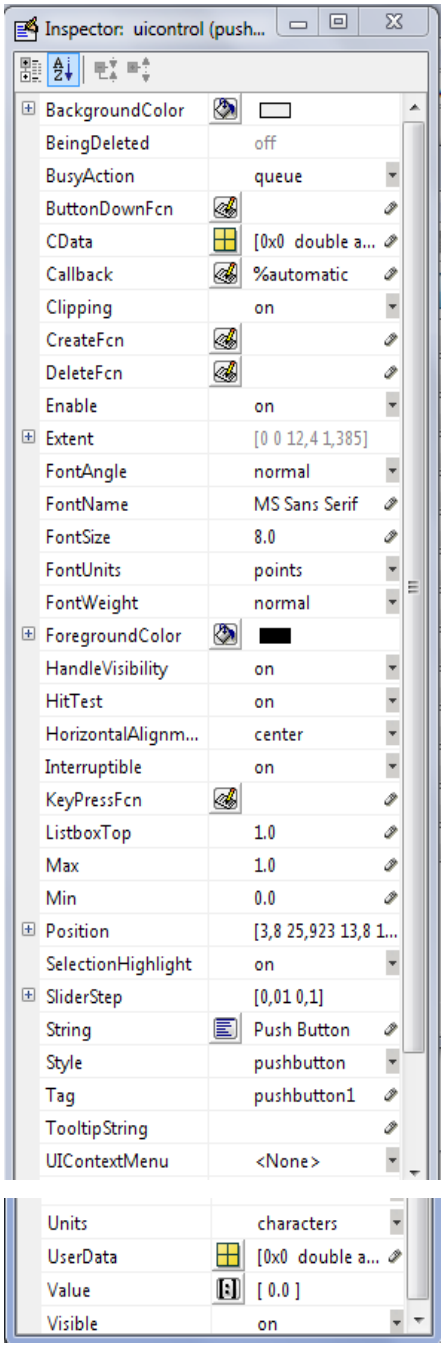


Şekil 3.27: Axes nesnesiyle yapılan uygulama

```
function pushbutton1_Callback(hObject, eventdata, handles)
axes(handles.axes1);
ezplot('sin(x)');

function pushbutton2_Callback(hObject, eventdata, handles)

axes(handles.axes2);
mesh(peaks);
```

3.10. GUI Nesne Özellikleri Penceresi

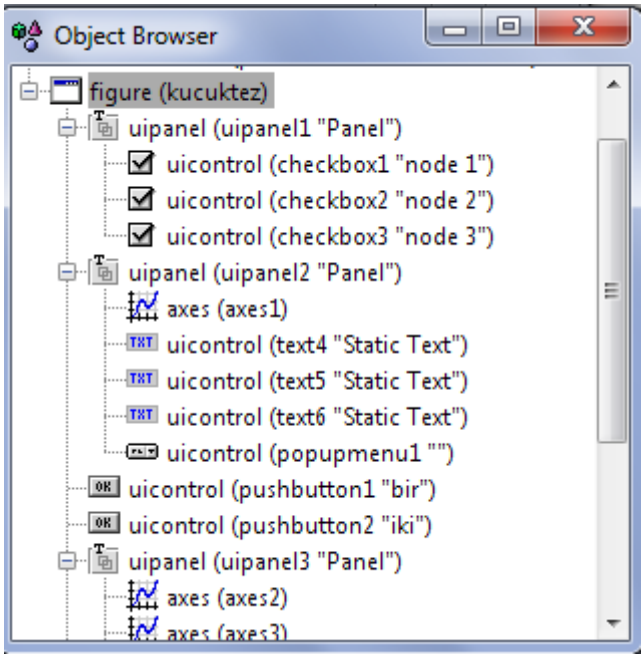
Bir gui nesnesinin boyutunun, renginin, etiketinin, yazı karakterinin vb. gibi birçok özelliğinin ayarlandığı bir penceredir. Nesne üzerine çift tıklanarak bu pencere açılır. Şekil 3.28' de her nesne için farklılık gösteren bu penceredeki bazı özellikler tanıtılmıştır.



Şekil 3.28: Nesne özellikleri penceresi

3.11. Nesne Hiyerarşisinin Gösterilmesi

Tasarım esnasında programcı hangi nesneleri ve hangi isimlerle kullandığı genel hataları ile görmek isteyebilir. Bu gibi durumlar için GUIDE tasarımcıya Object Browser aracını sunar. Bu araç View menüsünden çalıştırılabilir. Bu araç Şekil 3.29' da görülmektedir.



Şekil 3.29: Object browser penceresi

Object Browser aracı ile tasarımcı hangi nesneyi hangi Panel içinde veya hangi isim ile kullandığını kolaylıkla takip edebilir. Örneğin buradaki örnekte GUI figure içinde popup menu (açılır liste kutusu), axes (grafik çizimi), panel ile menü nesnelerinin eklendiği gözükmekte olup, panel içinde de üçü radio olmak üzere toplam dört adet buton kullanıldığı söylenilebilir.

3.12. GUI Uygulamalarında Callback' ler Arasında Ortak Veri Geçişini Sağlayan Yollar

GUI uygulamalarında bir değişken içeriği birden fazla callback içerisinde kullanılmak istenebilir. Ya bir GUI callback function' ın ürettiği değer başka bir function için giriş verisi olabilir. Konuyu daha geniş anlamıyla anlatılmak istenirse GUI uygulamalarında global değişken kullanım yolları öğretilmeye çalışılmaktadır. Bu durma benzer örnekler çoğaltılabilir. Bu durumu gerçekleştirmek üzere GUI uygulamalarında 6 farklı yöntem vardır.

3.12.1. Handles yapı değişkeni kullanılarak global kullanımı

GUI uygulamalarında en sık kullanılan yöntemdir. Bu yöntemde her callback function' a giriş parametre olarak gönderilen ve GUI uygulamalarında kullanıcı

verilerinden ziyade GUI nesneleri ile ilgili handle değerlerini tutmaya yarayan "handles" yapı değişkeninden yararlanılır. Bu yapıyı çok kolaydır. Örnek olarak sistem_cikis_sayisi isminde bir değişkeni her callback içerisinden ortak olarak kullanmak isteğimiz varsayılsın ve içeriği 4 yapılsın. Daha sonra da bu değişkeni, handles yapısı içerisine koymak istediğimizi düşünelim. Bu işlem için aşağıdaki komut satırları yazılmalıdır.

3.12.2. Global değişken tanımlama deyimi

Herhangi bir callback başında global deyimi ile bir değişken ismi girildiği takdirde eğer daha önce böyle bir değişken yok ise oluşturulur ve başlangıç değeri otomatik olarak 0 (sıfır) değeri atanır. Ancak, eğer daha önceden bu callback icra edilmiş ve global deyimi ile tanımlanan değişken bellekte bir yere sahip ve değeri var ise bu değerinin bir sonraki callback çağrısında da devam ettirecektir. Bu durumun kullanıma örnek komut satırları aşağıda gösterilmiştir.

```
Function edit1_callback( … )
global sayac;
sayac=sayac+1;
% diğer icra edilecek komutlar
if isequal(sayac,5)
sayac=0;
end
```

Yukarıdaki örnekte sayac değişkeni 0 dahil toplam 6 farklı değer almakta ve 5 olduğunda değeri tekrar sıfırlanmaktadır.

3.13. GUI Uygulamalarında Temizleme Komutları

Herhangi bir GUI uygulamasında figure penceresi veya komut satırı alanı ya da axes (grafik çizim) nesnesinin içeriğinin temizlenmesi gerekebilir. Ayrıca herhangi bir değişkenin workspace den atılarak bellekten temizlenmesi de gerekebilir. Bu durumlar için şu komutlar kullanılmalıdır.

• Figure alanını temizlemek için clf komutu
• Axes nesnesindeki çizimin temizlenmesi için cla komutu,
• Komut satır ekranının temizlenmesi için clc komutu,
• Worspace alanındaki tüm değişikenleri silmek ve bellekten temizlemek için clear all veya kısaca clear komutu
• Workspace alanından örneğin adet_no isimli değişken kaldırılmak istenirse clear adet_no komutu kullanılmalıdır.

3.14. GUI Uygulamalarında Kullanılan Standart Handle Değişkenleri

GUI uyugulamalarında birden fazla nesne veya figure alanı ile çalışıldığı düşünülürse bazen yanlış veya istenmeyen nesnelere kontrollerin kaydığı görülebilir. Bu gibi durumlardan sakınmak için aktif axes veya grafik nesnesi gibi bazı belirli nesneler için özel olarak handle numarasını tutmak amacıyla tanımlanmış değişkenler Matlab tarafından GUI tasarımcılarına sunulmuştur. Ayrıca ince bir bilgi olmasına karşılık bilmekte yarar var. Bir GUI uygulamasına ait figure nesnesinin handle numarası varsayılan olarak GUI handles yapısının içinde yer alan "output" isimli değişkende de tutulmaktadır.

GUI uygulamalarında handle numaralarını öğrenmek amacıyla sıklıkla kullanılan standart değişkenler şunlardır:

gcf : Geçerli figure nesnesinin handle numarasını verir.
gca : Geçerli axes (grafik çizim) nesnesinin handle numarasını verir.
gco : nesne_handle= gco(fig_handle) kullanımı ile GUI alanında en sok tıklanmış ya da en son aktif olan nesnenin handle numarasını verir.
gcbf : figure_bo = gcbf; kullanımı ile hangi figure nesnesine ait bir callback (veya figürün içerdiği bir nesne callback in çalışıyor olabilir) bu figure nesnesine ait handle numarası döner.
gcbo : Aktif olarak hangi nesnenin callback' i çalışıyor ise o nesneye ait handle numarası döner. Nesne_handle = gcbo olarak kullanılabileceği gibi [nesne_handle, figure_handle] = gcbo şeklinde kullanım ile aktif nesnenin bulunduğu figure handle numrası da elde edilebilir.

4. UYGULAMADA KULLANILAN KABLOSUZ ALGILAYICI AĞLAR

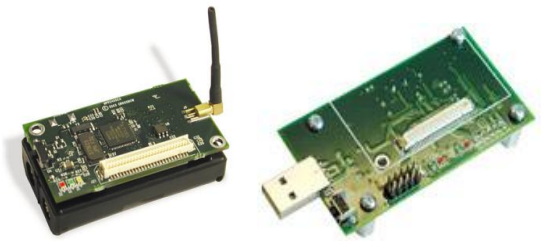
4.1. Giriş

Gerçekleştirilen sistemde Micaz algılayıcı düğümler kullanılmaktadır. Micaz algılayıcı düğümleri üzerinde MTS400 sensör kartları bulunmaktadır. Bu sensör kartı üzerinde ortamdaki çeşitli verileri (ışık, nem, sıcaklık, vb.) algılayan sensörler bulunur. Düğümlerde tinyOS işletim sistemi yüklüdür. tinyOS açık kaynak kodlu bir işletim sistemidir. Algılayıcı düğümler nesC dili ile programlanmaktadır.

Düğümlerden erişim noktasına gelen bilgiler –xmlport= 9002 portundan XML olarak yayınlanmaktadır. Perl yazılımı aracılığıyla gelen bilgiler 9002 portu dinlenerek alınmaktadır. Alınan veriler metin dosyalarına uygun biçimde yazılmaktadır.

4.2. Algılayıcı Düğümler

MTS400CA algılayıcı kartı ile kullanılarak ortamdaki sıcaklık, nem, basınç, ışık, voltaj bilgilerini ölçebilir. MicaZ algılayıcı düğümleri, gömülü algılayıcı ağ sistemleri için tasarlamış TinyOS açık kaynak kodlu işletim sistemi ile nesC programlama dili kullanılarak programlanabilmektedir. MicaZ düğümler içerisine yazılan kodu gömmek için MIB520 programlama bordu kullanılmaktadır. USB kablo ile bilgisayara bağlanan MIB520, aynı zamanda, algılayıcı ve eyleyicilerin bilgisayar ile bağlantısını sağlayan erişim noktası olarak da kullanılmaktadır. Şekil 4.1’ de MicaZ algılayıcı düğüm ve MIB520 programlama bordu (erişim noktası) görülmektedir.



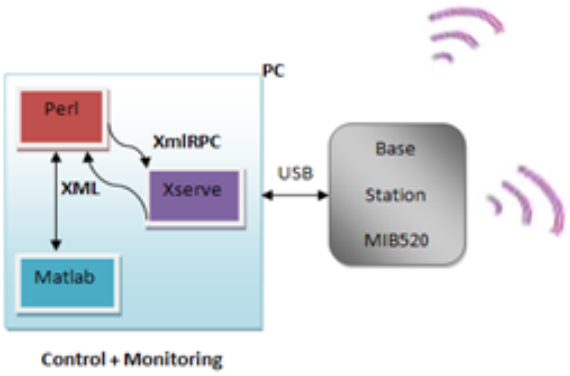
Şekil 4.1: MicaZ algılayıcı düğüm ve MIB520 programlama bordu (erişim noktası)

4.3. Ortamdaki Verilerin Toplanması ve Bilgisayara Aktarılması

Çalışmanın ilk kısmı olan verilerin ortamdan toplanması işlemini MicaZ algılayıcı düğümler gerçekleştirmektedir. MPR2600 devresi ile MTS400CA algılama kartı birlikte kullanılarak algılayıcı MicaZ düğüm oluşturulur. MP2600 içerisine yüklenecek uygun kod ile düğüm ortamdaki fiziksel büyüklükleri algılayabilir. MicaZ düğümler, TinyOS açık kodlu işletim sistemi ve nesC derleyicisi kullanılarak programlanabilirler.

TinyOS işletim sistemi bileşen tabanlıdır ve bu bileşenlerin iki tipi vardır; bunlar modül ve konfigürasyon dosyalarıdır. Konfigürasyon dosyası bileşenleri ve ara yüzleri birbirine bağlamayı sağlar. Bu sayede bileşenler birbirlerinin görevlerini, komutlarını ve olaylarını kullanabilmektedir. Bileşenler ara yüzler ile birbirlerine bağlanmaktadırlar. Ara yüz ve bileşenler paket haberleşmesi, yönlendirme, algılama, harekete geçirme ve depolama gibi işlemleri gerçekleştirebilmektedir.

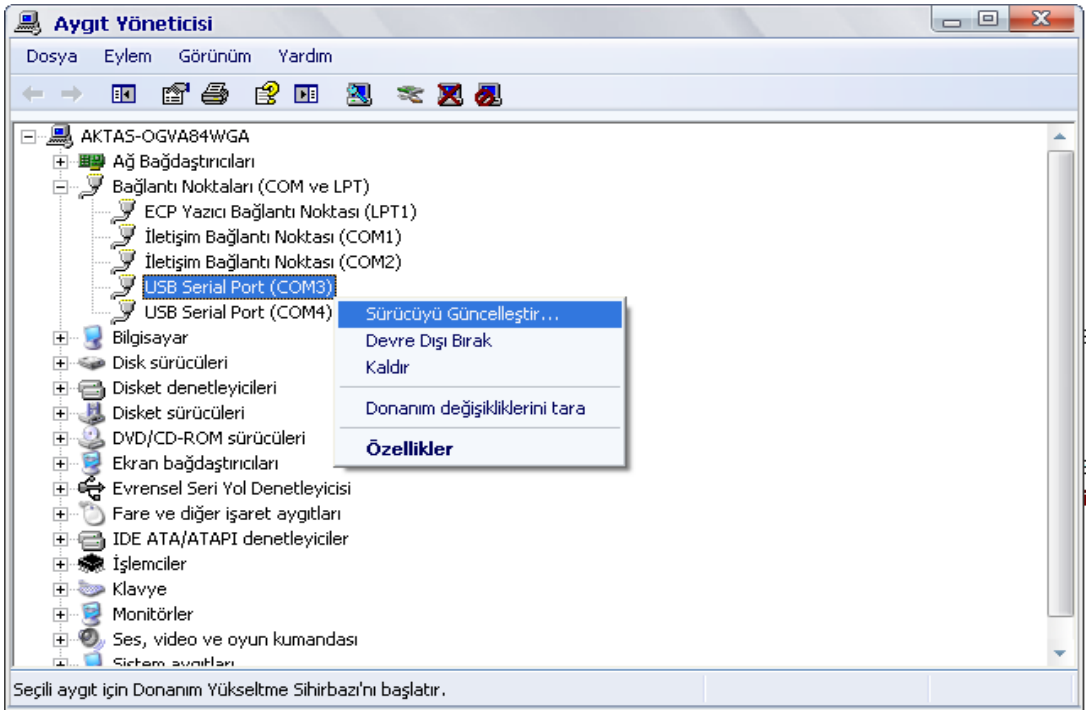
Algılayıcı düğümler ortamdaki farklı fiziksel büyüklükleri algılamaktadır. Algılanan veriler baz istasyonuna gönderilmektedir ve baz istasyonuna gelen veriler Xserve ile bilgisayar ortamına aktarılmaktadır. Xserve KAA ile uygulama yazılımı arasında geliştirilmiş bir geçit olarak kabul edilebilir. Algılanan bu veriler XML formatında yayınlanmaktadır. Bu veriler Perl yazılımı tarafından XML paket formatında alınır ve Matlab yazılımında kullanılmak üzere metin dosyalarına kaydedilir. (Şekil 4.2)



Şekil 4.2: Sistemin çalışması

4.3.1. MIB520 Erişim noktası yüklenmesi

Sistemin ilk kurulacak elemanı baz istasyonu yani erişim noktasıdır. Baz istasyonu USB ile bilgisayara bağlanmaktadır. Baz istasyonu bilgisayara ilk takıldığında yeni donanım bulundu sihirbazı çalışmaktadır. Yükleme Cd'si taklıdıktan sonra bilgisayar baz istasyonunu direkt tanıyarak yüklemeleri gerçekleştirmektedir. Yalnız Xserve programı baz istasyonunu seri porttan dinlemektedir. Bunun için erişim noktası yüklendiğinde iki tane sanal port açılmaktadır. Bu işlem "USB Serial Converter" olarak adlandırılabilir. Sanal portları aktif etmek için "Bilgisayarım/Sistem Bilgisini Görüntüle/Donanım/Aygıt Yöneticisi /Bağlantı Noktaları (COM ve LPT)" kısmına basmak gereklidir. Açılan portlarda sorun varsa (soru işareti gözüküyorsa), ilgili port üzerinde sağ tuşa basıp "Sürücüyü Güncelleştir" demek yeterlidir. Dosyalar Cd üzerinden bulunup yüklenecektir. Aşağıdaki şekilde görüldüğü gibi iki tane " USB Serial Port" oluşmuştur. Bunlar COM3 ve COM4 portlarıdır. Bu portlardan büyük numaralı olan ile haberleşme yapılır. Numarası küçük olan porttan ise sensör düğümüne komut gönderilir. Bu sistem için COM4 portu ile Xserve programı ile haberleşilir. Portlar Şekil 4.3' te aygıt yöneticisi penceresinde görülmektedir.



Şekil 4.3: Aygıt yöneticisinde portların listelenmesi

COM3 portuyla ise Xserveterm ile komut gönderilebilir.

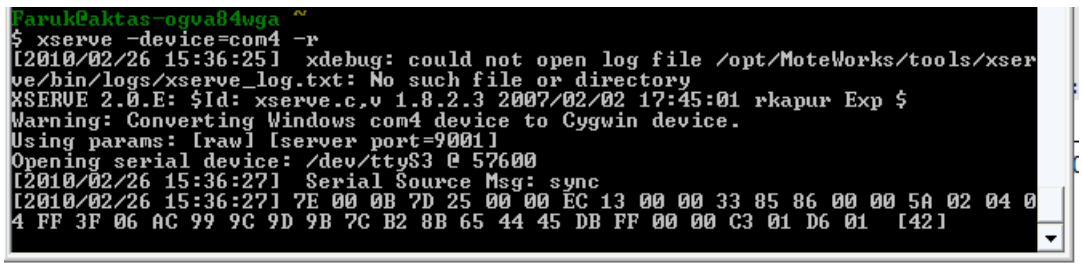
4.3.2. Xserve

Gerçek ortamdan bilgi almak için Xserve yazılımı çalıştırılmaktadır. Böylece o anda aktif olan tüm düğümlerden bilgi alınabilmektedir.

XSERVE mesh kablosuz ağları ve kurumsal uygulamalar ağ ile etkileşim arasında birincil ağ geçidi olarak hizmet vermektedir. Yüksek düzeyde hizmet XML tabanlı konfigürasyon dosyaları ve yüklenebilir eklenti modülleri kullanılarak özelleştirilebilir.

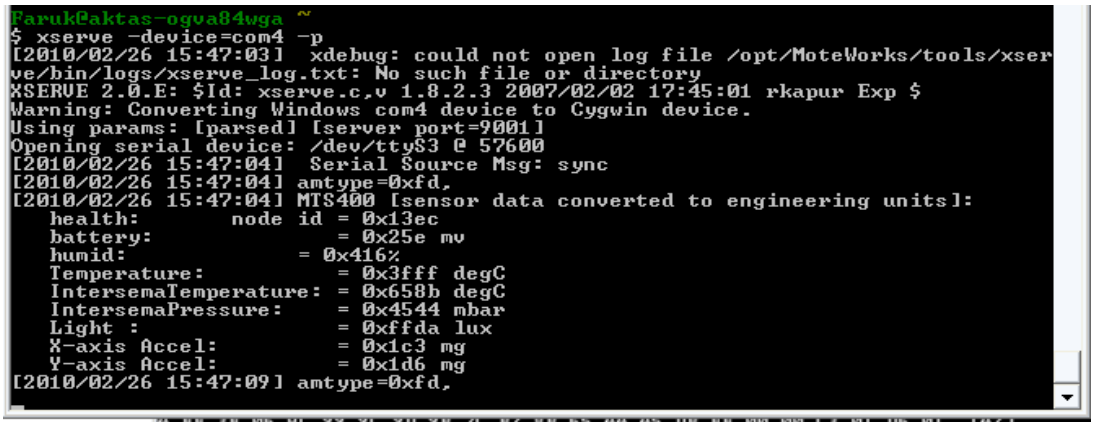
XSERVE' de veriler üç biçimde gösterilir. Bunlar; a-) Raw b-) Parsed c-) Converted formatlarıdır.

Raw Format: Raw formatında görüntüler. Bu format Şekil 4.4' te görünmektedir. 16lık sayı sistemine göre 1 byte' lık veri şeklinde görüntülenir. Raw formatlı verilerin görünümü aşağıdaki gibidir.



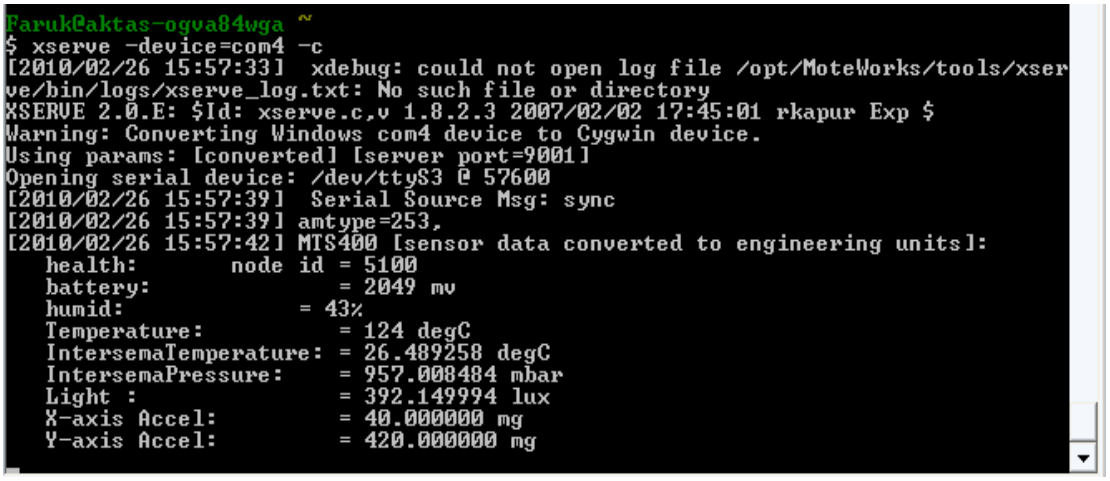
Şekil 4.4: Verilerin raw formatında görüntülenmesi

Parsed Format: Parsed formatında veriler ham olarak 16'lık sayi sistemine göre gösterilir. Yalnız verilerin özel değerleri yanlarına yazılır. Parsed formatlı verilerin görünümü Şekil 4.5' teki gibidir.



Şekil 4.5: Verilerin parsed formatında görüntülenmesi

Converted Format: Her değer ham değerinden ölçü birimine uygun şekilde dönüştürülmüş halde gösterilir. Converted formatlı verilerin görünümü Şekil 4.6' daki gibidir.



Şekil 4.6: Converted formatlı verilerin görünümü

Xserve ile bilgiler XML formatında alınabiliyor. XML formatlı bilgi almak için XML portuna bağlanmak gereklidir. Default port 9002'dir. XML bilgi çeşitli formatlarda dışarıya alınabilmektedir.

-xmlr= Raw formatlı XML paket çıkışı olur.

-xmlp=Parsed formatlı XML paket çıkışı olur.

-xmlc= Converted formatlı XML paket çıkış olur.

CYGWIN programı çalıştırıldıktan sonra aşağıdaki ifade yazılır. Cygwin, Windows içinde Linux tabanlı kod yazabilmek için kullanılan programdır.
Xserve -device=COM4 -h –c -xmlc –xmlport=9002
Yukarıdaki yazımın ayrıntıları şunlardır.

-device=COM4 ➔ Baz istasyonu ile hangi port üzerinden haberleşileceğini belirtir. Çalışılan bilgisayar için haberleşme portu COM4 portudur. (Açılan sanal portlardan büyük olanı.)

-h ➔ Xserve'in Web ara yüzüne erişim için yazılır. Bilgileri Web formatında da görmek için yazılır.
-c➔ Ortamdan gelen bilgileri anlamlı (Converted Format) halde almak için yazılır.
-xmlc➔ Anlamlı XML formatlı bilgi çıkışı için yazılır.
-xmlport=9002➔ XML formatlı bilgi göndermek için bağlanılacak port numarası yazılır.

4.3.3. XserveTerm

Xcommand ile uzaktan düğümlerin çalışma parametreleri değiştirilebilir, durum değişkenleri sorgulanabilir, düğümler üzerinde bulunan led veya buzzer gibi devreler çalıştırılabilir veya durdurulabilir yada bir düğümün örnekleme hızı değiştirilebilir. Bu işlemleri Xcommand' in ara yüzü Xserveterm ile gerçekleştirilmektedir. Şekil 4.7' de Xcommand ile yapılabilecek işlemler ve tanımlamaları bulunmaktadır.

Command Category	Command	Arguments	Description
Power Management	RESET SLEEP WAKEUP		To reset the sleep and wakeup time.
Basic Update Rate	SET_RATE		To get or set the update rate. The set_rate command changes the data acquisition duty cycle of the mote. The first argument is the new timer interval in milliseconds.
Mote Configuration Parameter Settings	• GET_CONFIG • SET_NODEID • SET_GROUP • SET_RF_POWER • SET_RF_CHANNEL		This set of commands allows you to get and set radio frequency and power, including channel.
Actuation	ACTUATE • SET_LED • SET_SOUND • SET_RELAY	0=OFF, 1=ON, 2=TOGGLE 0=OFF, 1=ON 0=OFF, 1=ON	This set of commands actuates each individual LED with the option to operate on all three LEDs. This command turns the sounder off or on. This command turns the relays on or off.

Şekil 4.7: Xcommand ile yapılabilecek işlemler

4.3.4. XserveTerm' in çalıştırması

Düğüm veya düğümler çalışır duruma getirildikten sonra ilk önce masaüstündeki Cygwin uygulaması çalıştırılır. Düğümlerden gelen bilgileri, düğüme ait ID bilgilerini ve xmlport ayarlarını yapmak için Şekil 4.8' deki kod satırı yazılmalıdır.

```
Elo@atlantis ~
$ xserve -device=com6 -c -h -xmlc
```

Şekil 4.8: xmlport ve format ayarlarının yapılması

Çalışan bilgisayarda com5 ve com6 sanal portları açılmıştır. Bu portlardan büyük numaralı olanı ile haberleşme yapılmaktadır. Burada com6 yazılmıştır. Xmlport ile ilgili bir bilgi yazılmadığında default olarak 9003 portuna yazılmaktadır. 9002 portuda kullanılabilir. Onun için gerekli olan kod satırı Şekil 4.9' daki gibidir.

```
Elo@atlantis ~
$ xserve -device=com6 -c -h -xmlport=9002 -xmlc
```

Şekil 4.9: Portun seçilmesi

Daha sonra masaüstündeki Cygwin uygulaması tekrar çalıştırılır.

4.3.4.1. Uzaktan sistem kontrol uygulaması

Masaüstündeki cygwin programı çalıştırılır. Şekil 4.10' daki kodlar yazılır.

Şekil 4.10: Cygwin açıldığında çalıştırılacak kodlar

Xserve programı xml desteğiyle çalıştırımış olur.

H:\WNCS\WSN\Perl\XCommandReadXML1.pl perl programıyla xserve programındaki bilgiler (ışık şiddediti) xml formatında alınıp Matlab' da kullanılmak üzere dosyaya (data.txt) yazılır.

H:\WNCS\WSN\Perl\GrafikCiz.m programıyla ekrana dosyaya yazılan ışık şiddeti bilgisinin grafiği çizdirilir.

H:\WNCS\WSN\Perl\Kontrol.m programıyla ekrana dosyaya yazılan ışık şiddeti bilgisinin grafiği çizdirilir.

H:\WNCS\WSN\Perl\XCommandCustomAction.pl perl programı kullanılarak xmlrpc yöntemiyle mda320 kartının digital çıkışlarına istenen değer gönderilerek kontrol yapılması sağlanır.

5. KABLOSUZ ALGILAYICI AĞLAR İÇİN MATLAB İLE KULLANICI ARAYÜZ TASARIMI UYGULAMASI

5.1. Giriş

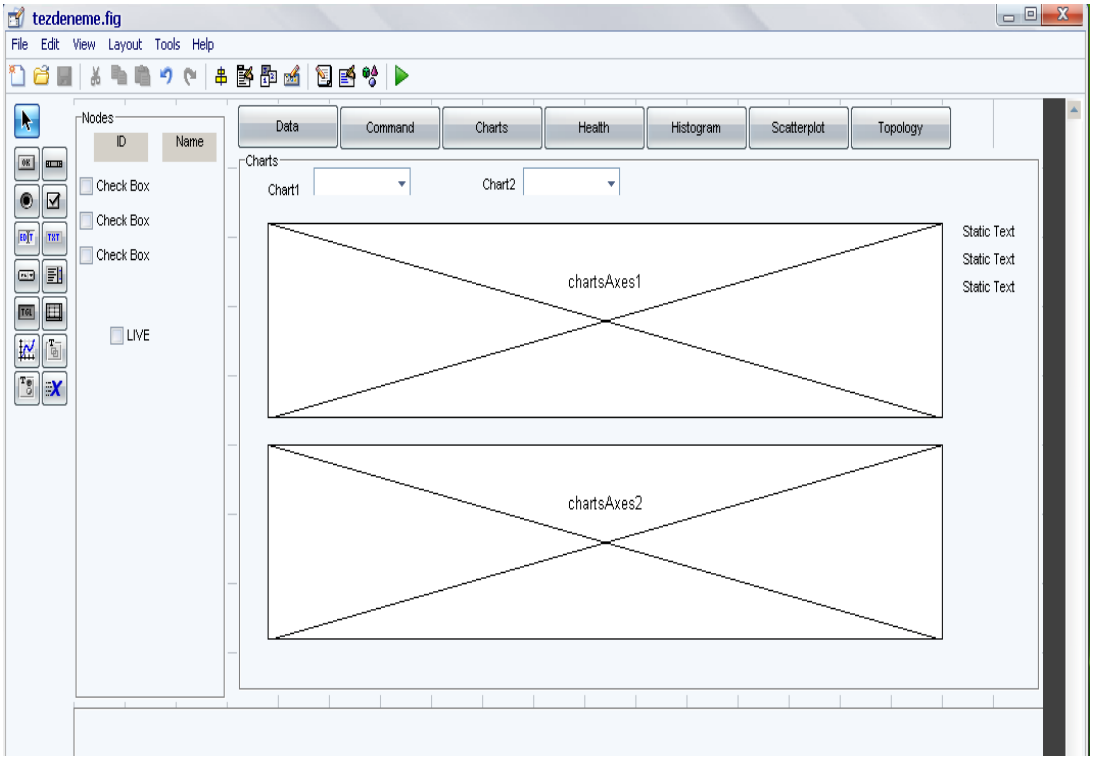
Bu bölümde, KAA' lar kullanılarak hazırlanmış olan sistemde istenilen sayıdaki algılayıcı düğümden gelen bilgilerin metin dosyalarına uygun formatta yazıldıktan sonraki görüntüleme işlemi ve sonuçları anlatılmıştır. Metin belgelerine düğümlerden gelen node id, düğümün pil gerilimi (voltage), sıcaklık (temp), basınç (press), nem (humid), ışık (light) verileri kaydedilir. Her algılayıcı düğüm için ayrı bir metin dosyası oluşturulur. Gelen veriler ilgili dosyalarda aralarında birer boşluk olacak şekilde tek satırda saklanır. Yeni değer geldiğinde kaydedilen veriler güncellenir.

5.2. Uygulamaya Genel Bakış

Tez çalışmasında bilgilerin görüntülenmesi amaçlı yapılan kullanıcı ara yüzünün tasarımı MATLAB GUI kullanılarak yapılmıştır. Şekil 5.1' de görülen kısım uygulamanın .fig dosyasıdır. Uygulamada bileşen olarak üst üste paneller, butonlar, açılır menüler, axes grafik nesneleri, seçme kutuları (checkbox), metin kutuları (statictext) kullanılmıştır.

Sol taraftaki "Nodes" panelinde uygulama çalıştığı anda sisteme bağlı olan algılayıcı düğüm sayısı kadar eklenen seçme kutucukları görülmektedir. Verilerin dosyalardan yeniden okunmasını ve güncellenmesini sağlayan "Live" bileşeni de yine bu panelde yer almaktadır.

"Data", "Command", "Charts", "Histogram", "Scatterplot" butonlarının tümü birer sekme olarak kullanılmıştır. Uygulamanın orta bölümünde seçilen sekmeye göre görüntülenen her sekmeye ait paneller bulunmaktadır.



Şekil 5.1: Uygulamanın .fig dosyası

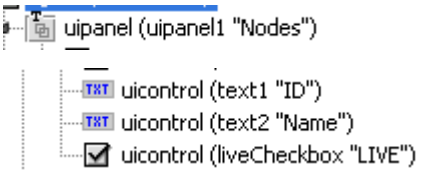
5.2.1. Nodes paneli

Sol taraftaki "Nodes" panelinde uygulama çalıştığı anda sisteme kaç adet algılayıcı düğüm bağlıysa o kadar seçme kutusu otomatik olarak oluşmaktadır. Seçme kutularına isim olarak "node id" verilmektedir. Bu işlem, uygulama çalıştığında tüm dosyaların açılarak "node id" bilgilerinin okunması ve seçme kutularının "string" özelliğine atanmasıyla gerçekleştirilir. "Live" seçme kutusu tıklanarak dosyaların belli rutin aralıklarla açılıp yeniden okunarak verilerin güncellenmesi sağlanır.

5.2.2. Nodes panelinde kullanılan bileşenler

Şekil 5.2' de görülen bileşenler kullanılmaktadır. Düğümler için kullanılan seçme kutuları "figure" dosyasına eklenmemiştir. Çünkü düğüm sayısı değişkendir, sabit bir sayı kullanılmamaktadır. Düğüm sayısına göre seçme kutusu ekleme işlemi, uygulama çalıştığı anda otomatik olarak gerçekleştirilecektir.

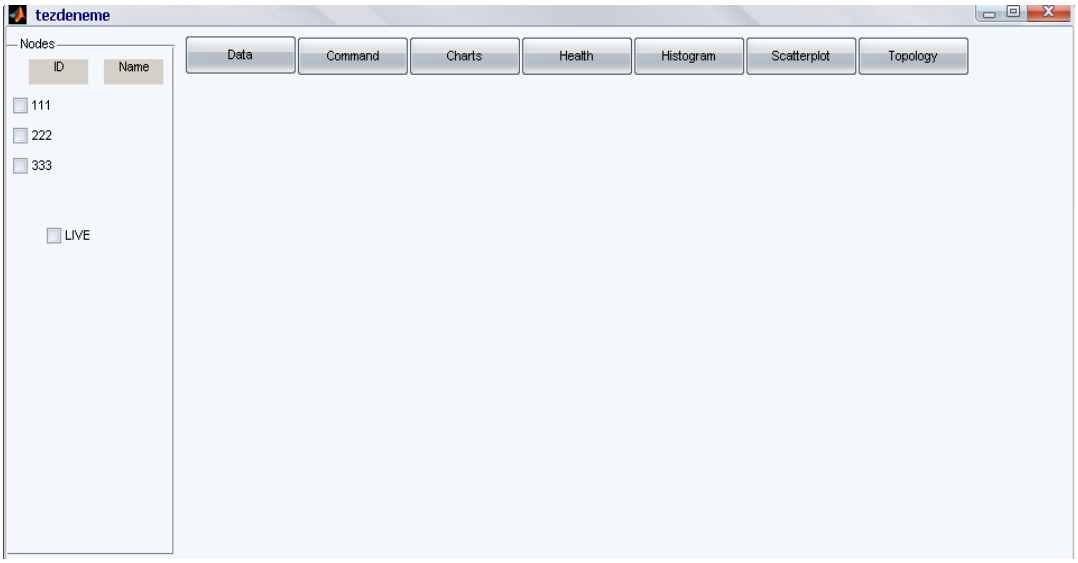
"Live" butonuyla, seçilen düğümlerin dosyaları rutin aralıklarla açılıp okunarak verilerin güncellenmesi sağlanmaktadır.



Şekil 5.2: Nodes paneli object browser

5.3. Uygulamanın çalışması

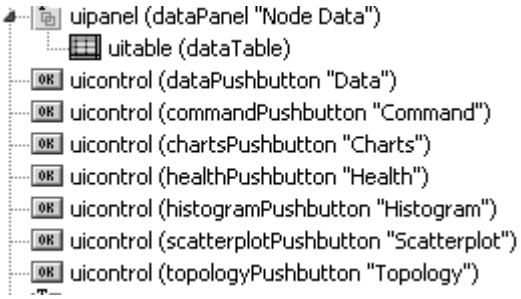
Uygulama çalıştırıldığında Şekil 5.3' teki pencereyle karşılaşılır. "Nodes" panelinde o anda sisteme bağlı olan düğümler seçme kutusu bileşeni olarak listelenir. Arayüzün orta kısmında uygulama ilk çalıştığı anda hiçbir panel görüntülenmez. Seçilen sekmeye göre görüntülenir.



Şekil 5.3: Uygulamanın çalışmış hali

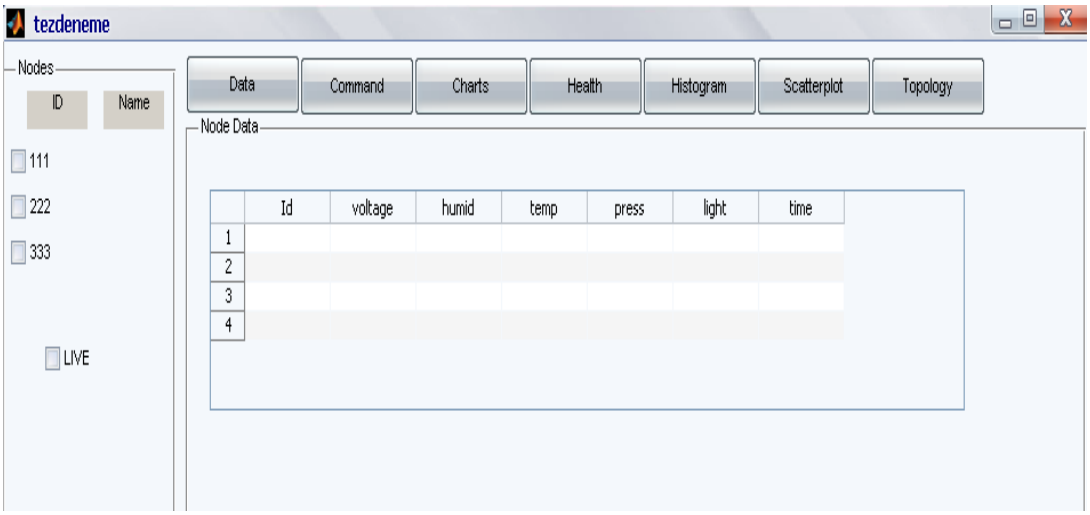
5.3.1. Data paneli

"Data" panelinde Şekil 5.4' teki bileşenler kullanılmaktadır. "Uitable" nesnesi sistemdeki düğümlerden gelen verileri görüntülemek için kullanılır.



Şekil 5.4: Data paneli object browser

Şekil 5.5' teki tabloda verilerin görüntülenmesi için verilerin kaydedildiği dosyalar açılarak veriler okunarak "uitable" bileşenine yansıtılır.

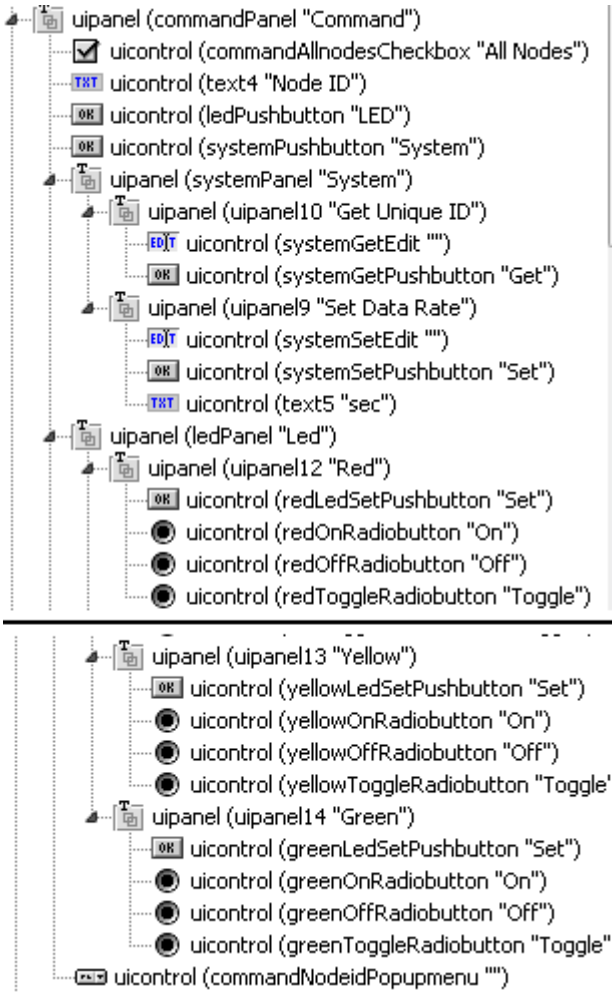


Şekil 5.5: Data sekmesi

5.3.2. Command paneli

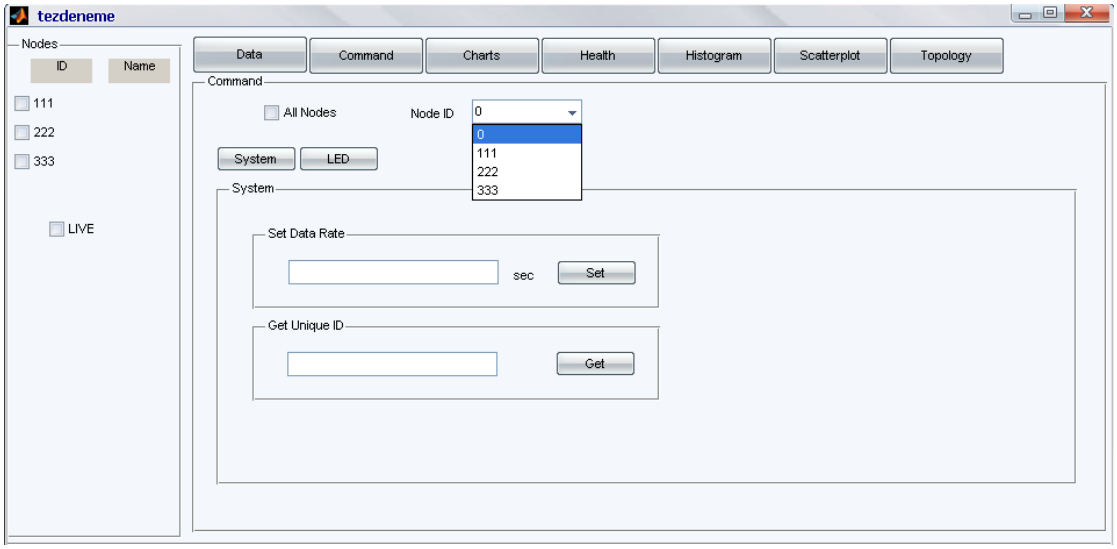
"Command" panelinde düğümlere komut gönderme işlemi gerçekleştirilir. Bu panelde içinde iki panel daha kullanılmıştır. Bu iki panel ve diğer bileşenler Şekil 5.6' da görülmektedir.

Bu panelde genel olarak tüm düğümlere aynı komutu göndermemizi sağlayacak olan "All nodes" seçme kutusu ve ayrı ayrı düğüm seçmeyi sağlayacak olan açılır menü kullanılmıştır. "System" ve "Led" butonları ise yukarıda bahsedilen iki panel arasında geçiş yapmayı sağlamaktadır.



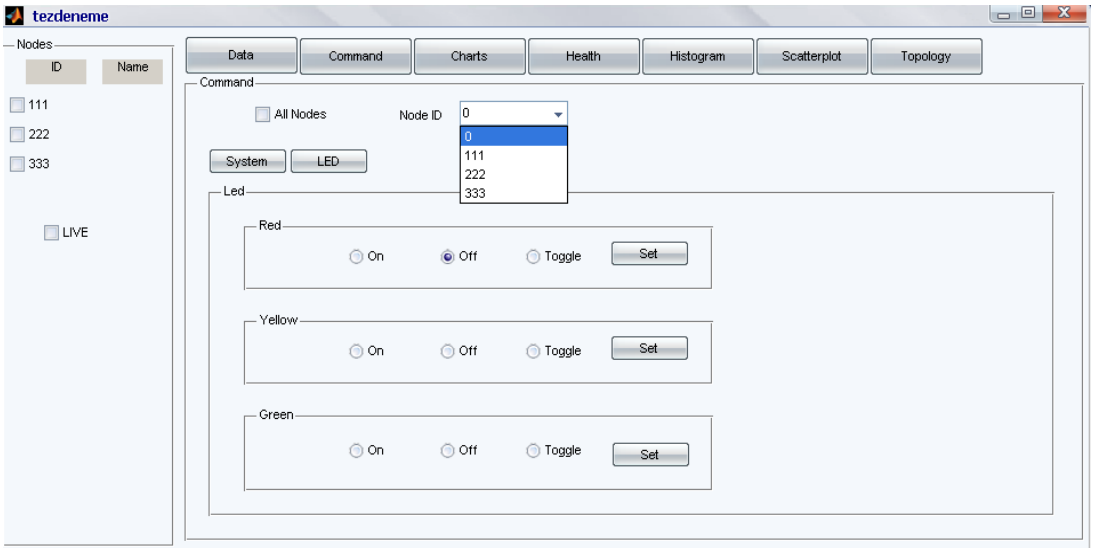
Şekil 5.6: Command paneli object browser

Şekil 5.7' de "System" paneli görülmektedir. Bu panelin "Set Data Rate" bölümündeki "Set" butonu, düğümlerin saniyede alınan veri oranının değiştirilmesini sağlar.



Şekil 5.7: Command sekmesindeki system paneli

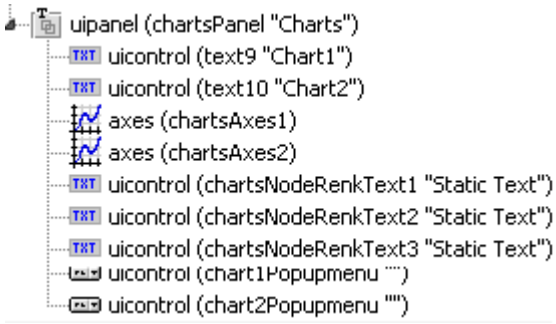
Şekil 5.8' de "Led" sekmesi görülmektedir. Bu sekmede düğümlerin ledlerine müdahale edilebilir. Led ışıklarının yanıp sönmesine yönelik komut gönderme işlemleri gerçekleştirilir.



Şekil 5.8: Command sekmesindeki led paneli

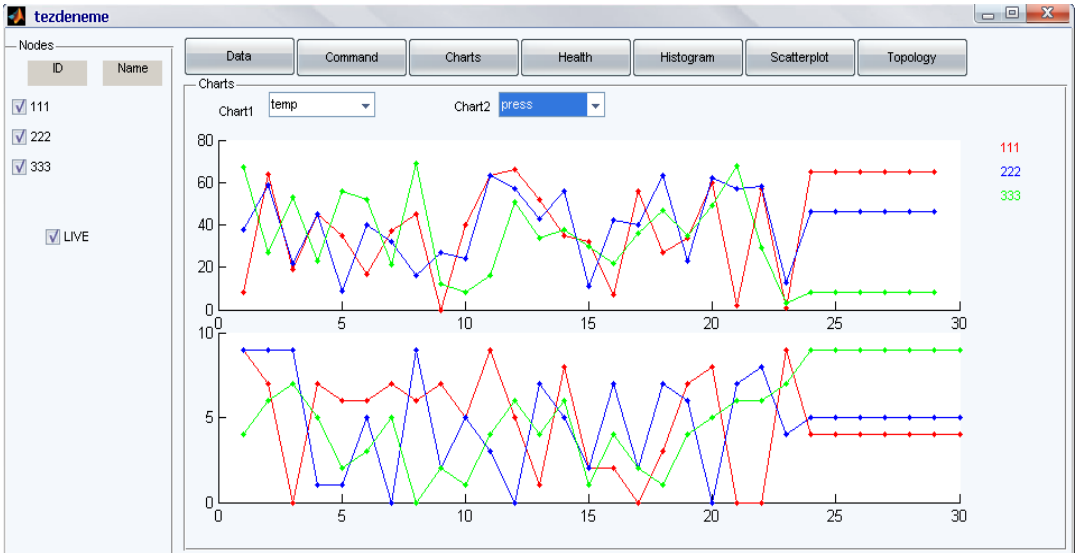
5.3.3. Charts paneli

"Charts" panelinde kullanılan bileşenler Şekil 5.9' da görülmektedir. İki adet "axes" grafik bileşeni ve iki adet açılır menü kullanılmıştır. Açılır menülerde görüntülenmek istenen veri çeşitleri tutulur.



Şekil 5.9: Charts paneli object browser

"Chart" panelinin amacı, seçilen düğümlere ait farklı 2 veriyi aynı anda görüntülemektir. Şekil 5.10' da seçilen tüm düğümlerin birinci grafik bileşeninde sıcaklık verilerinin grafikleri, ikinci grafik bileşeninde basınç verilerinin grafikleri görülmektedir.



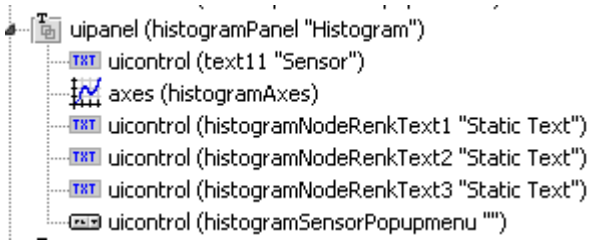
Şekil 5.10: Charts sekmesi

Görüntülenme işlemine "Live" seçme kutusundan seçim yapılmışsa başlanır. "Live" butonuna tıkladıktan sonra rutin zaman aralıklarıyla verilerin kaydedildiği dosyalar açılır ve güncellenen veriler grafik olarak grafik ekranlarına yansıtılır.

Her düğümden gelen veriler için grafik çiziminde farklı renkler kullanılarak, grafikteki verilerin hangi düğüme ait oldukları daha net bir şekilde görüntülenmektedir. Grafik çizdirilirken, eski veriler de saklanarak sürekli bir grafik oluşturulması sağlanmaktadır.

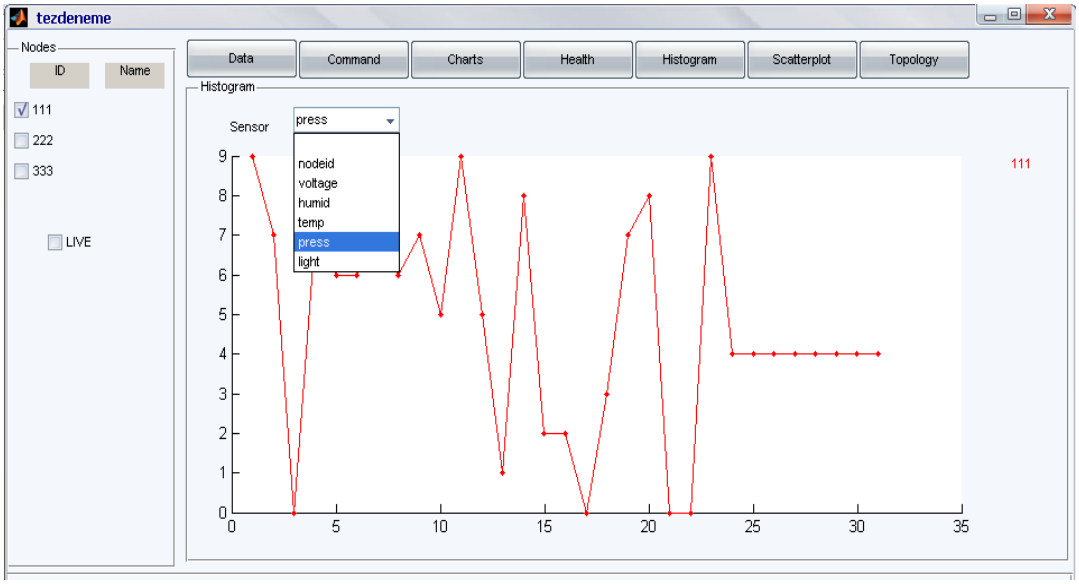
5.3.4. Histogram paneli

"Histogram" panelinde Şekil 5.11' deki bileşenler kullanılmaktadır. Verileri görüntülemek istenilen sensör listesi açılır menüde tutulmaktadır. Buradan seçilen verilerin grafiğini görüntülemek için de bir adet "axes" nesnesi kullanılmıştır.



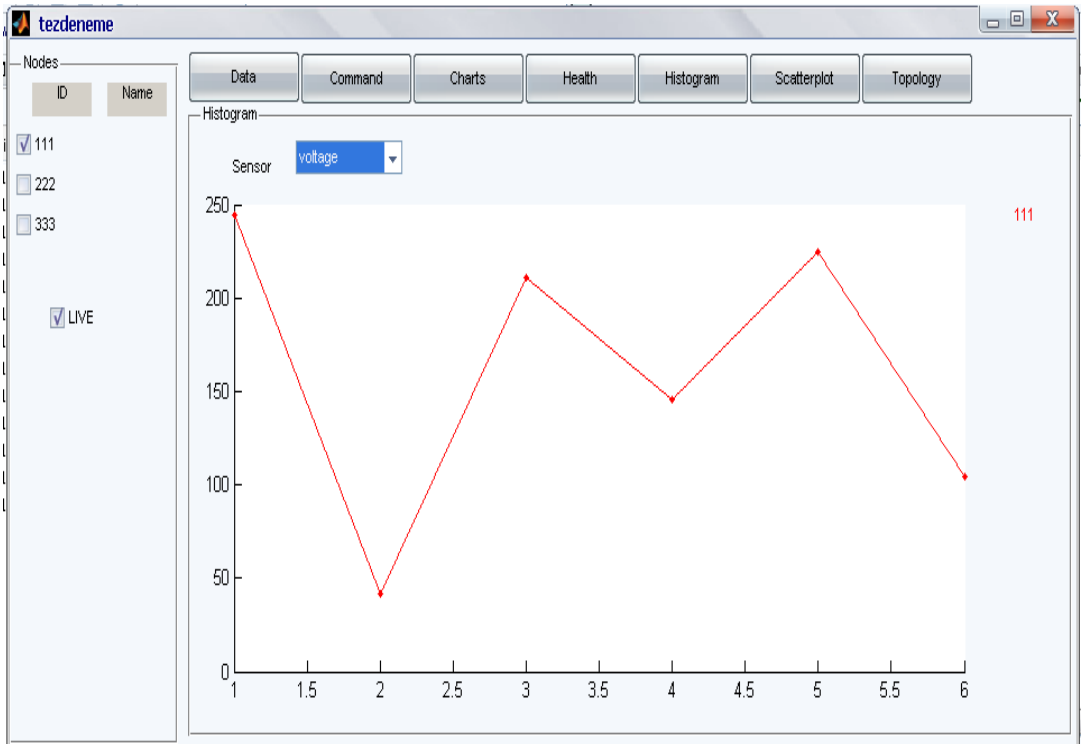
Şekil 5.11: Histogram panel object browser

Şekil 5.12' de öncelikle "Nodes" panelinden düğüm seçilmiş, "live" seçme kutusuna tıklanmış ve sensör listesinden "press" seçilerek basınç verilerinin görüntülenmesi ve grafiğinin çizdirilmesi işlemi gerçekleştirilmiştir.



Şekil 5.12: Histogram sekmesinde düğüm seçilmesi

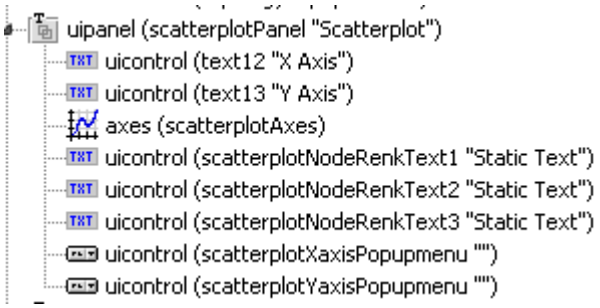
Şekil 5.13' te ise farklı bir sensör verisi görüntülenmektedir.



Şekil 5.13: Histogram sekmesinde farklı verinin görüntülenmesi

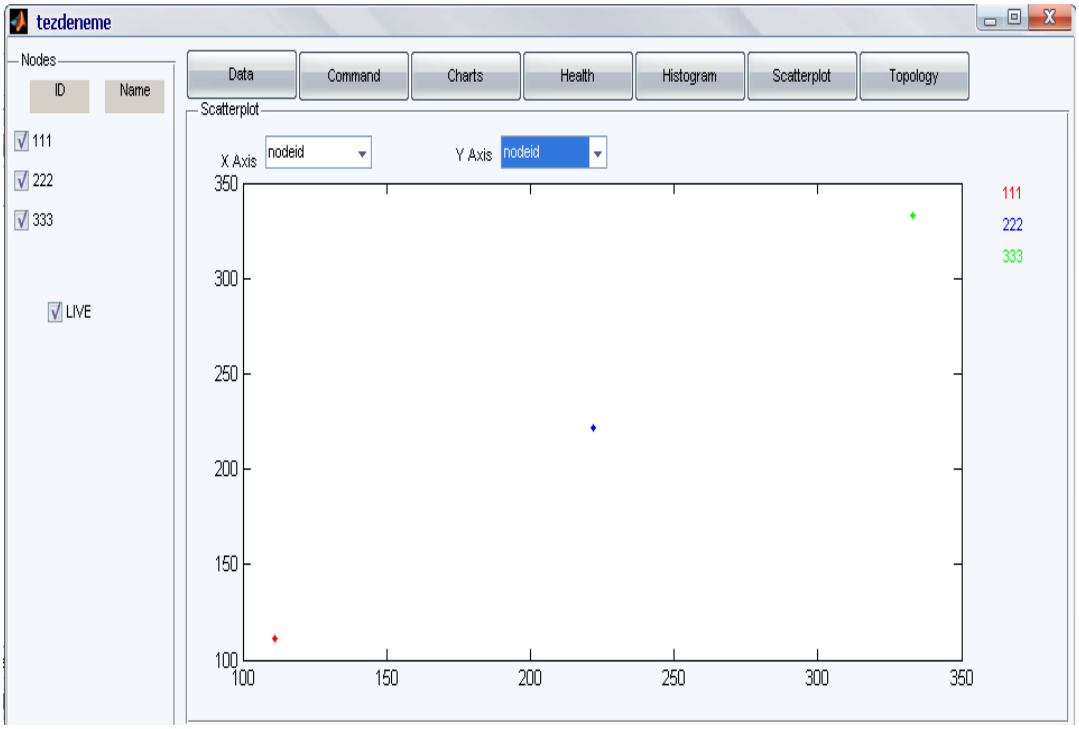
5.3.5. Scatterplot paneli

"Scatterplot" panelinde Şekil 5.14' te görüldüğü gibi "axes" ve iki adet açılır menü bileşeni kullanılmaktadır. Açılır menüler x ve y eksenlerinde hangi sensör bilgisinin görüntüleneceğini belirlemeyi sağlamaktadır.



Şekil 5.14: Scatterplot paneli object browser

“Scatterplot” sekmesinin amacı, seçilen düğümlere ait farklı sensörlerin değerlerinin eksenlere yansıtılarak, değerlerin birbiri ile ilişkilerinin gözlemlenmesidir. Şekil 5.15’ te görüldüğü gibi, x ve y eksenlerinden “node id” bilgisi seçilerek iki büyüklüğün aynı grafik ekranından ilişkilerinin gözlemlenmesi sağlanmaktadır. Eksenlerde görüntülenmek istenen sensörler aynı olmak zorunda değildir.



Şekil 5.15: Scatterplot sekmesi

6. SONUÇLAR VE ÖNERİLER

Bu tez çalışmasında, KAA' larda düğümlerden gelen verileri görüntüleme işleminin MATLAB GUI tabanlı gerçekleştirilerek geliştirilmeye açık bir yapıya kavuşması, çok çeşitli uygulama alanlarına hitap etmesi ve farklı amaçlar için yeni bir yaklaşım geliştirilmesine katkıda bulunması sağlanmaktır.

Uygulamanın MATLAB tabanlı olması sayesinde, ileriki çalışmalarda MATLAB' da gerçekleştilebilen veri görüntüleme, görüntü işleme ve kontrol ile ilgili uygulamalar eklenebilmesi mümkün görünmektedir.

KAYNAKLAR

Aktaş, F., Çeken C., Erkan, K., Yıldırım, M., "Kablosuz Algılayıcı Ağlar Kullanılarak Birinci Dereceden Ölü Zamanlı Bir Sistemin Denetimi", ***6th International Advanced Technologies Symposium (IATS' 11)***, 2011, Elazığ, Türkiye

Aydın, H., "Matlab ile Kontrol Sistemlerinin İncelenmesi", Bitirme Tezi, ***Marmara Üniversitesi Teknik Eğitim Fakültesi***, İstanbul, 2003

A. Mainwaring, J. Polastre, R. Szewcyk, D. Culler, J. Anderson, "Wireless Sensor Networks for Habitat Monitoring", ***International Workshop on Wireless Sensor Networks and Applications***, Atlanta, Georgia, ABD (2002).

Doğan, U., "Temel Bilgisayar Bilimleri Ders Notları", Lisans Ders Notları, **Yıldız Teknik Üniversitesi**, İstanbul

Erdal, H., Savaş, K., "Kontrol Sistemleri için MATLAB' da GUI Uygulamaları Tasarımı", Lisans Bitirme Tezi, ***Marmara Üniversitesi Elektronik Bilgisayar Bölümü***, İstanbul, 2007

Kalaycı, T. E., "Kablosuz Sensör Ağlar ve Uygulamaları", ***Ege Üniversitesi Bilgisayar Mühendisliği Bölümü***

MathWorks, "***MATLAB, The Language of Technical Computing***", September 2006

Okçuoğlu Z., Ertürk İ., Karahan A., "Kablosuz Algılayıcı Ağ Uygulaması: İdeal İzleme", ***Elektrik−Elektronik ve Bilgisayar Mühendisliği Sempozyumu, ELECO'08***, Bursa, Türkiye, 26–30, Kasım (2008).

Raghavendra C. S., Sivalingam K. M., Znati T., "Wireless Sensor Networks**"**, ***Springer***, (2004).

Ruken, Z., "Kablosuz Sensör Ağlarda Servis Kalitesi ve Enerji Tüketimi Denetiminde Enformasyon Teorisi Kullanımı", Yüksek Lisans Tezi, ***Atılım Üniversitesi Fen Bilimleri Enstitüsü***, Ankara

Şefkat, G., Yüksel, İ., "Matlab Gui Tabanlı Elektromıknatıs Devre Tasarımı ve Analizi", ***Pamukkale Üniversitesi Mühendislik Fakültesi Mühendislik Bilimleri Dergisi***, 2003

Uzunoğlu, M., Kızıl, A., Onar, Ö.Ç., "Heryönü ile MATLAB", ***Türkmen Kitapevi***, İstanbul, 2003

Wahlin, A. K., "Matlab Course", ***Department of Geophysics***, University of Oslo, January 2003

Ye Akyıldız, I.F., Su, W., Sankasubramaniam, Y., Çayırcı, E., "Wireless Sensor Networks: A Survey", ***Computer Networks***, 393 – 422, (2002).

"Crossbow MICAz Datasheet" [Online], http://www.xbow.com/Products/ Product_pdf_files/Wireless_pdf/MICAz_Datasheet.pdf, (**Ziyaret Tarihi: 02 Nisan 2011**).

"How to use tables in Matlab" [Online], http://stackoverflow.com/ questions/ 5452944/how-to-use-tables-in-matlab, (**Ziyaret Tarihi: 12 Kasım 2010**).

"Kablosuz Toprak Nemi Ölçüm ve Kontrol Sistemi" [Online], http://www.cvm.com.tr, (**Ziyaret Tarihi: 05 Ocak 2011**).

"Tab Panel Constructory", [Online] http://www.mathworks.com/ matlabcentral/ fileexchange/6996-tabpanel-constructor-v2-8-2010, (**Ziyaret Tarihi: 30 Ekim 2010**).

"Tab panels – uitab and relatives" [Online], http://undocumentedmatlab.com/ blog/tab-panels-uitab-and-relatives, (**Ziyaret Tarihi: 17 Mart 2011**).

"Updating graph or plot in realtime" [Online], http://www.mathworks.com/ matlabcentral/newsreader/view_thread/155564, (**Ziyaret Tarihi: 10 Şubat 2011**).

ÖZGEÇMİŞ

1986 yılında İzmit' te doğdu. İlk, orta ve lise öğrenimini Kocaeli' de tamamladı. 2004 yılında girdiği Kocaeli Üniversitesi Teknik Eğitim Fakültesi Bilgisayar Öğretmenliği Bölümü' nden 2008 yılında Bilgisayar Öğretmeni olarak mezun oldu. 2008 yılında Kocaeli Üniversitesi Fen Bilimleri Enstitüsü Elektronik ve Bilgisayar Eğitimi Anabilim Dalı' nda Yüksek Lisans öğrenimine başladı. 2009 yılından beri Körfez Hereke Nuh Çimento Teknik ve Endüstri Meslek Lisesi' nde Bilgisayar Öğretmeni olarak görev yapmaktadır.