

# Kütüphaneler için Telsiz Duyarga Ağı Tabanlı Yangın Tespit Sistemi

## Wireless Sensor Network based Fire Detection System for Libraries

Sadık Arslan

Kent Kart Ar-Ge Merkezi  
Kent Kart Ege Elektronik A.Ş.  
İzmir, Türkiye  
sadik.arslan@kentkart.com.tr

Moharram Challenger

Uluslararası Bilgisayar Enstitüsü  
Ege Üniversitesi  
İzmir, Türkiye  
moharram.challenger@ege.edu.tr

Orhan Dagdeviren

Uluslararası Bilgisayar Enstitüsü  
Ege Üniversitesi  
İzmir, Türkiye  
orhan.dagdeviren@ege.edu.tr

**Özetçe**—Kütüphane ortamı içerdiği kağıt kaynaklı yayımlardan dolayı yangın tehlikesi altındadır. Kamuya açık bir alan olduğundan ve el yazması tarihi dokümanlar gibi çok değerli kaynaklar içerdiğinden dolayı kütüphanelerde can ve mal güvenliğinin sağlanması çok önemlidir. Bu sebeple sıcaklık, gaz ve duman algılayıcı tabanlı yangın algılama sistemleri kullanılmaktadır. Bu çalışmada, telsiz duyarga ağları kullanılarak, TinyOS geliştirme platformunda, bir kütüphane yangın uyarı sistemi tasarlanmıştır. Sistemde IRIS düğümler ve sıcaklık duyargaları kullanılmıştır. Tasarlanan sistem kitaplık ortamlarında performans testlerine sokulmuştur. Bilgilerimiz dahilinde, geleneksel duman duyargaları içeren sistemlerde 30 sn civarında tepki verdiği gözlenirken, yeni sistemlerin 3 sn civarında tepki verdiği rapor edilmiştir. Elde ettiğimiz sonuçlara göre bizim sistemimiz 3 sn'den önce tepki vermiştir. Önerilen sistem kolay kurulum, hızlı uyarı oluşturma, basit kullanım gibi alanlarda oldukça başarı elde etmiştir.

**Anahtar Kelimeler**—telsiz duyarga ağları; TinyOS; kütüphane yangın tespit sistemi

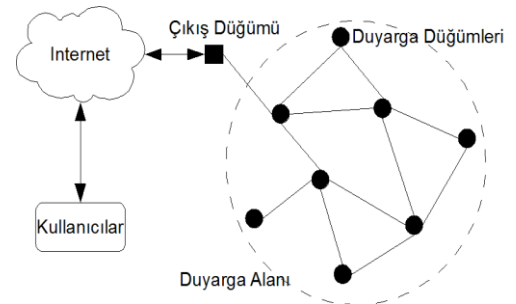
**Abstract**—The library environment is vulnerable to fire threat due to the paper based publications it contains. Since it is a public open environment and it contains valuable resources like handwritten historical documents, it is very important to ensure the safety of life and property in libraries. Therefore, gas and smoke detection based fire detection systems are used. In this study, using wireless sensor networks, a library fire detection system has been designed and implemented on the TinyOS development platform. IRIS nodes and temperature sensors are used in the system. The designed system is tested in library environments. To the best of our knowledge, it has been reported that detection systems generate alarm in approximately 30 s, while new systems are reported to react in about 3 s. According to the results obtained, our system has reacted before 3 s. The proposed system is successful with features such as easy installation, quick alert creation and simple usage.

**Keywords**—wireless sensor networks; TinyOS; library fire detection system

### I. GİRİŞ

Günümüzde telsiz duyarga ağları (TDA) kullanımı sistemlerdeki fiyat düşüşleri boyutlardaki küçülmeler ile oldukça yaygın hale gelmiştir. İsteğe göre oldukça yüksek sayıda düğümün kullanılması ile kendi kendini yöneten tasarsız sistemler kurulabilmektedir [1]. TDA kullanımı, yangın yönetimi, askeri uygulamalar, tarım ve hayvancılık gibi birçok farklı alanda oldukça yaygındır.

TDA sisteminde bulunan düğümlerde uygulamaya özel duyargalar bulunmaktadır. Duyargalar toplamış oldukları verileri sistemde bulunan diğer düğümler üzerinden çıkış düğümüne iletmektedir. Çıkış düğümü, genel olarak diğer düğümlerden gelen verileri toplar ve sistem kullanıcıların erişebileceği farklı ağlara ağ geçidi görevini yapar. Örnek bir TDA yapısı Şekil 1'de verilmiştir. Şekilde çıkış düğümünde internet bağlantısı olacak şekilde gösterilmiştir ancak burada farklı ağlar da olabilmektedir. Sistemlerde düğümler sadece tek yönlü çalışmamaktadır. Düğümlerden çıkışa veri gidebildiği gibi çıkış düğümünden de ters yönde diğer düğümlere veri gidebilmektedir [2]. Örneğin bir yangın tespit sisteminde sıcaklık duyargasından alınan bilgiler düğümlerden çıkış düğümüne doğru gitmektedir. Alınan sıcaklık bilgisine göre sistemde bir sorun olduğu sonucu üretilirse, çıkış düğümünden söndürme sisteminin bağlı olduğu düğümlere doğru da bir veri akışı olabilmektedir.



Şekil 1. Telsiz duyarga ağlarının genel yapısı [2].

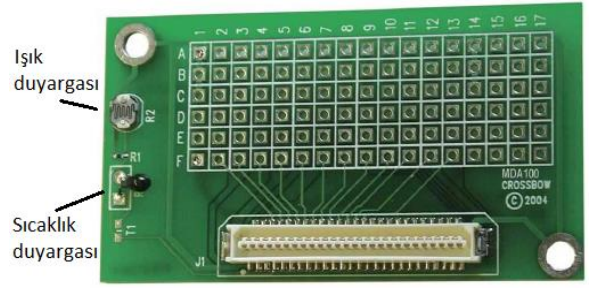
TDA'nın yangın, duman, gaz kaçağı gibi felaket ortamlarında kullanımını içeren oldukça fazla sayıda çalışma mevcuttur. Yangın durumu binalarda, ormanlarda, madenlerde, yeraltında bulunan metro istasyonları ve tüneller gibi birçok farklı alanlarda ortaya çıkabilmektedir. Kapalı alanlarda TDA kullanımını içeren çalışma sayısı oldukça yüksek iken kütüphane ortamına özelleşmiş TDA kullanımını içeren çalışmalar daha kısıtlıdır [3]. Bina içlerinde yangın ve gaz kaçağı durumunda uyarı üreten çalışmalar kablosuz ağlar ile gerçekleştirilmiştir [4]. Bina içinde yangın esnasında itfaiye bölümünü uyarın, söndürme ve kurtarma konusunda yardımcı olma hedefiyle tasarlanmış sistemler bulunmaktadır [5]. Kapalı alan yer belirleme teknikleri kullanılarak yangının yerini bulma işlemini sağlayan çalışmalar da mevcuttur [6]. Bazı geliştiriciler önemli tarihi yapıların yangından korunması için telsiz duyurğa ağları üzerinde çalışmalar yapmışlardır [7]. Kapalı alanlardan bir diğeri olan madenlerde de yangın riskleri oldukça yüksektir. Özellikle kömür madeni gibi alanlarda gaz ve yangın konusunda güvenlik konularını içeren çalışmalar mevcuttur [8]. Yeraltı tünellerinde ve yeraltı treni istasyonu gibi tesislerde yangın tahliye ve tespit konularını içeren çalışmalar literatürde bulunmaktadır [9]. Ormanlık alanlar yangına en açık alanlardır. Bu alanlarda yangın ve duman algılama hayati önem taşımaktadır. Birçok bilim insanı ormanlardaki yangın uyarı sistemlerinin TDA uygulamaları üzerinde çalışmaktadırlar [10-16].

Bu çalışmada, kütüphane ve kitaplık ortamı için özelleşmiş bir sistem tasarımına ağırlık verilmiştir. Sistemin çalışma parametreleri tasarım esnasından itibaren bu çalışma ortamına uygun olacak şekilde optimize edilmiştir. Kütüphane ortamının hızlı yanıt alma ihtiyacını karşılanabilmesi için geliştirme ortamının sınırları zorlanmıştır. Bu çerçevede tüm geliştirme ve test işlemleri yerine getirilmiştir.

Bildirinin 2. bölümünde sistemde kullanılan birimler için konfigürasyon ve kurulum anlatılmaktadır. Bölüm 3'te sistemin mimarisi ve tasarımı yazılım ayrıntılarına girilmeden verilmiştir. Bildirinin 4. bölümünde sistem için geliştirilen TinyOS ve bilgisayar uygulamaları ayrıntılı olarak verilmiştir. Bölüm 5'te sistemin kitaplık ortamında yapılan test çalışmaları ele alınmaktadır. Son bölümde, yapılan çalışmanın tartışması ve elde edilen sonuçları özetlenmiştir.

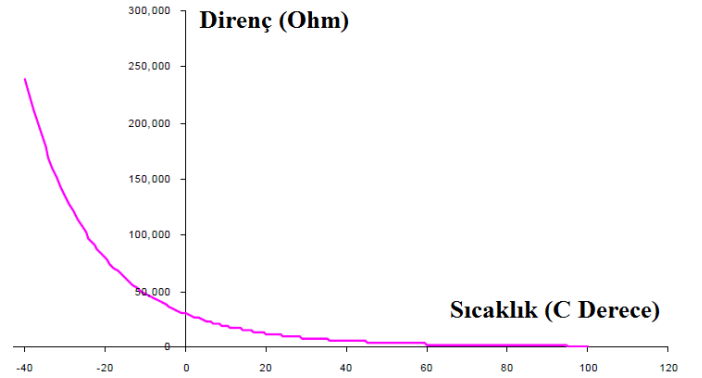
## II. SİSTEM KONFIGÜRASYON VE KURULUMU

Üzerinde çalışma yapılan sistemde düğüm olarak IRIS XM2110 [17] ürünü kullanmıştır. Bu düğüm 2.4 GHz frekansında çalışan, IEEE 802.15.4 [18] standardı uyumludur. Ayrıca çok farklı duyurğa kartlarının desteği de bulunmaktadır. Sistemde sıcaklık duyurğasında bilgi alınabilmesi için Şekil 3'teki MDA100CB [19] ürünü kullanılmıştır. Bu kart üzerinde ışık ve sıcaklık bilgilerinin alınabildiği duyurğalar ve IRIS düğümünün kullanabileceği diğeri ara yüzlerin dışarıya çıkarılmış olduğu lehimlenebilir delikler bulunmaktadır.



Şekil 3. MDA100CB duyurğa kartı.

Sıcaklık ölçümü duyurğası MDA100CB kartında bulunmaktadır. Burada sıcaklık ölçümü için analog-sayısal dönüştürücü (analog-to-digital converter, ADC) vardır. Bir düğümde toplam 8 adet ADC girişi bulunmaktadır. Sıcaklık duyurğası olarak bir yarı iletken malzeme olan termistör, YSI 44006 [20], kart üzerinde bulunmaktadır. Bu malzeme negatif ısıl katsayısına (Negative Temperature Coefficient, NTC) sahiptir. Bunun anlamı sıcaklık arttıkça malzemenin direnci azalmaktadır. Malzemenin bu özelliğinden faydalanılarak sıcaklık ölçümü yapılmaktadır. Bir gerilim bölücü devresine bağlanan termistör, düğümlerin ADC1 girişine bağlanmaktadır. Şekil 4'te termistörün sıcaklık direnç grafiği görülebilmektedir.



Şekil 4. Termistörün sıcaklık direnç grafiği.

Şekil 4'teki grafikten de anlaşılacağı üzere termistörün direnç sıcaklık karakteristiği doğrusal değildir. Ancak 0-50 derece aralığında doğrusal kabul edilerek işlemlerin yapılması da mümkündür. ADC1'den elde edilen değerin santigrat karşılığı hesaplanabilmektedir. 1023 alındığında 50 C olacağı kabul edilmektedir. Örneğin geliştirilen program aracılığı ile duyurğadan 544 değeri okunmuş ise 544 değeri 26.5 C olmaktadır. Bu ölçüm de oda sıcaklığına yakındır.

IRIS duyurğa düğümleri üzerinde TinyOS işletim sistemi kullanılmıştır. Uygulamalar C tabanlı nesC programlama dili ile geliştirilmiştir. TinyOS, olay tabanlıdır ve TDA düğümleri için düşük güç kullanarak TDA üzerinde uygulamaları daha kolay geliştirmeyi amaçlayan bir işletim sistemidir [2]. İşlem gücü ve belleği kısıtlı cihazlarda kullanılan TinyOS, klasik işletim sistemlerinden farklı olarak daha hafif bir yapıya sahiptir. TinyOS farklı bileşenlerden oluşmaktadır. Bu bileşenler uygulamalarda farklı amaçlar için kullanılmaktadır. Her uygulama kendi ihtiyaçlarına göre gereken bileşenleri programa ekler ve az kaynak kullanarak amaçlarını

gerçekleştirebilir. Genel olarak TinyOS bir telsiz duyurga düğümünün bileşenlerine çeşitli servisler sunmaktadır.

### III. SİSTEM MİMARİSİ VE TASARIMI

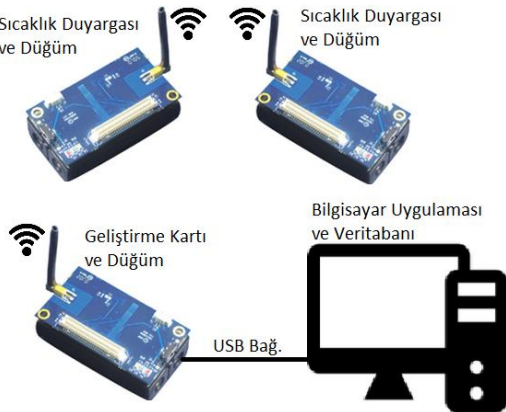
Tasarlanan sistemde temel olarak 4 bölüm bulunmaktadır. Bunlar, ölçüm yapılan düğümler, bilgilerin toplandığı çıkış düğümü, bilgilerin değerlendirildiği bilgisayar uygulaması ve ölçüm bilgilerinin kaydedildiği veri tabanıdır.

İlk bölüm sıcaklık ölçümlerinin yapıldığı düğümlerdir. Bu düğümdeki uygulama sıcaklık ölçümlerini almaktadır ve aldığı duyurga bilgilerini çıkış düğümüne göndermektedir. Ölçüm uygulaması toplam 10 adet ölçüm almaktadır. Aldığı ölçüm bilgilerini biriktirmekte ve ardından göndermektedir. Sistemdeki düğümler için yıldız bağlantı topolojisi kullanılmıştır. Burada ölçüm yapan düğümler sadece çıkış düğümüne bilgi göndermektedir. Çoklu atlama yoktur.

İkinci bölüm olan çıkış düğümü 2,5 sn periyotlarla bilgileri ölçüm yapılan düğümden almaktadır. Alınan bilgiler bilgisayara gönderilmektedir. IRIS düğümleri bilgisayara bağlandığında sanal seri haberleşme ara yüzü kullanılabilir hale gelmektedir. Düğümlerden seri veri gönderildiğinde bilgi haberleşme ara yüzünden iletilmektedir. Bilgisayarda bulunan uygulama bu ara yüzü kullanmakta ve işlemlerini yapmaktadır.

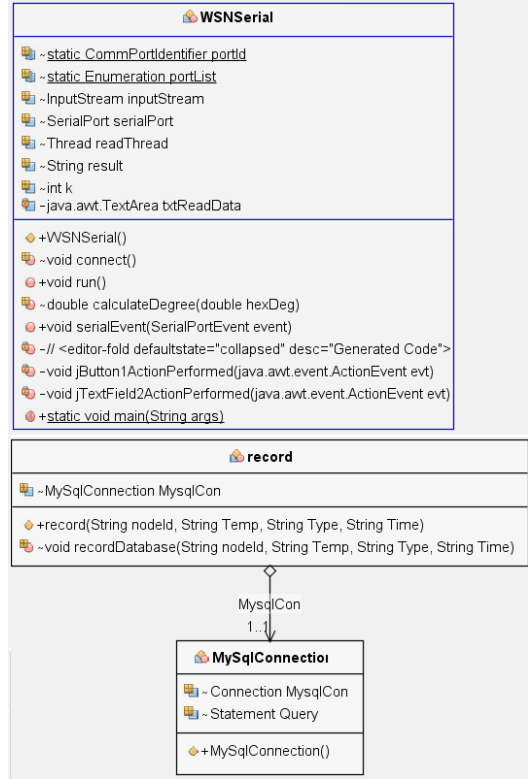
Üçüncü temel bölüm bilgisayar uygulamasıdır ve NetBeans ortamı kullanılarak Java dilinde geliştirilmiştir. Bilgisayar uygulaması Java Swing kütüphanelerinin kullanılması ile bir grafik kullanıcı ara yüzü oluşturmaktadır. Bu uygulama istenilen haberleşme ara yüzü, haberleşme hızı, düğüm kimlik bilgisi ve alarm eşik değerinin ekrandan girilmesinden sonra çalışmaktadır. Çıkış düğümü ile bağlantı kurulduktan sonra haberleşme portundan gelen bilgiler ayrıştırılmakta ve sıcaklık bilgisini ekrana yazılmaktadır. Bu uygulamada alınan onaltılık tabanda (hex) sıcaklık bilgilerinin santigrat derece birimine dönüşüm işlemi de yapılmaktadır. Ekranda düğüm kimlik numarasına göre alınan hex ölçüm değeri ve bu değerın santigrat derece bilgileri görülebilmektedir. Ölçüm bilgileri düğüm kimlik numarasına göre zaman etiketi de eklenerek bir veri tabanına kaydedilmektedir.

Son bölüm olan veri tabanı, MySQL WAMP ile oluşturulmuştur. Veri tabanındaki tabloda düğüm numarası, sıcaklık, normal veya uyarılı durum ve zaman bilgileri bulunmaktadır. Şekil 5'te sistemin yapısı görülebilmektedir.



Şekil 5. Sistemin genel yapısı.

Bilgisayar uygulamasında veri tabanına kayıt işleminin yapılabilmesi için *record* sınıfı kullanılmaktadır. *MySQLConnection* sınıfı ile de veri tabanı bağlantısı işlemleri yapılmaktadır. *SerialEvent* metodunda *record* sınıfından nesne oluşturulmakta ve burada veri tabanına yazılacak olan bilgiler nesneye girilmektedir. *Record* sınıfının değişken tanımlamalarında *MySQLConnection* sınıfından bir nesne oluşturularak veri tabanı bağlantısı yapılmaktadır. Ayrıca *record* sınıfı yapılandırıcısında alınan bilgiler ile veri tabanına kayıt işlemi için *recordDatabase* metodu çağrılmaktadır. Böylece veri tabanı kaydı yapılmaktadır. Şekil 6'da uygulamanın UML diyagramı bulunmaktadır.



Şekil 6. Sıcaklık okuma uygulamasının UML diyagramı.

### IV. SİSTEM YAZILIM GELİŞTİRME

#### A. Duyurga Düğümü Yazılımı

Sistem IRIS düğümleri üzerinde, sıcaklık ölçümü yapılabildiği diğer düğümlere kablosuz olarak gönderilebilmesi için TinyOS işletim sisteminde, nesC programlama dili kullanılıp *Oscilloscope* isminde uygulama geliştirilmiştir. Uygulama geliştirmeden önce sıcaklık ölçümünün yapılabilmesi için MDA100CB duyurga kartının kütüphanelerinde güncelleme yapılmıştır. Temel olarak *DemoSensorC* kütüphanesindeki *Read* ara yüzünde *TempC()* fonksiyonu kullanılmıştır. Bu kullanım ile aşağıdaki kodda görülebileceği gibi duyurganın sıcaklık biriminden veri alınması sağlanmıştır.

```

generic configuration DemoSensorC() {
    provides interface Read<uint16_t>;
}
  
```

```
implementation {
    components new TempC() as Sensor;
    Read = Sensor;
}
```

Sıcaklık okuma ve okunan değerlerin çıkış düğümüne gönderiminde bir ortak mesaj paket yapısı kullanılmıştır. Mesaj paket yapısında eş zamanlılık için *version* ve *interval* değişkenleri kullanılmaktadır. Düğümün kimlik bilgisi için *id*, okuma sayısını tutan *count* ve okunan sıcaklık bilgilerini tutan *readings* değişkenleri bulunmaktadır. Okuma değişkeni 10 adet okuma tutmaktadır. Tüm değişkenler *nx\_uint16\_t* tipindedir, tanımlanması aşağıdaki gibi *Oscilloscope.h*'ta yapılmıştır.

```
typedef nx_struct oscilloscope {
    nx_uint16_t version; // zaman versiyonu
    nx_uint16_t interval; // örnekleme periyodu
    nx_uint16_t id; // dugun kimlik numarası
    nx_uint16_t count; // mesaj gönderim sayısı/
    nx_uint16_t readings[NREADINGS]; //sıcaklık bilgileri
} oscilloscope_t;
```

Bu yapıda bulunan *NREADINGS* her mesajda kaç adet okuma bulunacağını belirlemektedir ve *enum* olarak değeri 10'a eşitlenmiştir. Her duyarga okuması 2 byte iken 10 kez okununca her mesajda 20 byte duyarga verisi bulunmaktadır.

Okuma yapacak düğüm açılırken *Boot.booted* metodunda *Oscilloscope.h* dosyasında tanımlanan ilk zaman aralığı bilgisi kullanılır. Burada bu bilgi 256 olarak girilmiştir. Bu durumda her duyarga okuması aralığı 256 ms olmaktadır ve 10 adet okuma işleminin tümü 2,5 sn civarındadır. Bu okumalardan sonra diğer mesaj bilgileri ile birlikte paket çıkış düğümüne atılır. Sonuç olarak her mesaj gönderim işlemi 2,5 sn periyotlarla gerçekleştirilmektedir. Zamanlayıcı ayarlanırken de alınan değerler ile periyodik kurulum yapılmaktadır. *Boot* ve *timer* işlemlerini içeren kodlar aşağıdaki gibidir.

```
event void Boot.booted() {
    local.interval = DEFAULT_INTERVAL;
    local.id = TOS_NODE_ID;
    if (call RadioControl.start() != SUCCESS)
        report_problem();
}
void startTimer() {
    call Timer.startPeriodic(local.interval);
    reading = 0;
}
event void RadioControl.startDone(error_t error) {
    startTimer();
}
```

Zamanlayıcının dolması koşulunda, mesaj gönderimi durumu ve paketin gönderim büyüklüğünde olup olmadığı bir *if* karşılaştırması ile yapılır. Eğer okuma sayısı yüksek olmuşsa ve paket büyüklüğü yeterli seviye ise *broadcast* işlemi yapılır. Mesaj gönderilememiş ise *Read.read* çağrılarak tekrar okuma işlemi yapılmaktadır. Aşağıda bu işlemlerin yapılmış olduğu *Timer.fired()* metodu inceleme amacıyla verilmiştir.

```
event void Timer.fired() {
    if (reading == NREADINGS){
        if (!sendBusy && sizeof local <= call
            AMSend.maxPayloadLength()){
            memcpy(call AMSend.getPayload(&sendBuf, sizeof(local)),
                &local, sizeof local);
```

```
        if (call AMSend.send(AM_BROADCAST_ADDR, &sendBuf,
            sizeof local) == SUCCESS)
            sendBusy = TRUE;
        }
        if (!sendBusy)
            report_problem();
        reading = 0;
        if (!suppressCountChange)
            local.count++;
        suppressCountChange = FALSE;
    }
    if (call Read.read() != SUCCESS)
        report_problem();
}
```

Duyargadan okuma işleminde atamalar *Read.readDone* metodunun içerisinde yapılmaktadır. Sıcaklık bilgisi *data* değişkeninde bulunmaktadır. Veri okunamaması durumunda ise *0xffff* bilgisi *data* değişkenine aşağıdaki gibi yazılmaktadır.

```
event void Read.readDone(error_t result, uint16_t data) {
    if (result != SUCCESS){
        data = 0xffff; //veri okunamıyor
        report_problem();
    }
    local.readings[reading++] = data; //okuma ataması
}
```

## B. Çıkış Düğümü Yazılımı

Duyarga okumalarından gelen paketler çıkış düğümünde yakalanmaktadır. Çıkış düğümünde TinyOS uygulamalarından olan *BaseStation* uygulaması referans alınmıştır. Bu uygulama temelde, sıcaklık okuyan düğümlerden aldığı paketleri sanal seri ara yüz aracılığı ile bilgisayara göndermektedir. *Boot.booted* metodunda sistemin radyo ve seri ara yüz başlangıçları yapılmaktadır. *RadioReceive.receive* metodu ile paket alındıktan sonra *receive* fonksiyonu çağrılır. Fonksiyonun içerisinde alınan paketler *uartSendTask()* kullanılarak seri ara yüz ile gönderilir. Burada mesaj alma ve gönderme işlemleri *atomic* olarak yapılmıştır. *uartSendTask()* içerisinde de *atomic* işlemler mevcuttur ve ayarlamalardan sonra bilgisayara bilgi gönderimi yapılmaktadır. Aşağıda *uartSendTask()* fonksiyonunun kodu görülebilmektedir.

```
task void uartSendTask() {
    uint8_t len;
    am_id_t id;
    am_addr_t addr, src;
    message_t* msg;
    atomic //işlemler atomic
    if (uartIn == uartOut && !uartFull){
        uartBusy = FALSE;
        return;
    }
    msg = uartQueue[uartOut];
    tmpLen = len = call RadioPacket.payloadLength(msg);
    id = call RadioAMPacket.type(msg);
    addr = call RadioAMPacket.destination(msg);
    src = call RadioAMPacket.source(msg);
    call UartPacket.clear(msg);
    call UartAMPacket.setSource(msg, src);
    if (call UartSend.send[id](addr, uartQueue[uartOut], len) ==
        SUCCESS)
        call Leds.ledIToggle();
    else{
        failBlink();
        post uartSendTask(); //veri gönderiliyor
```



```

}
}

```

### C. Bilgisayar Uygulaması ve Veritabanı

Seri ara yüzden okuma yapan, Java dili ile geliştirilen, uygulama çıkış düğümünü dinlemektedir. Alınan bilgileri ekrana ham hex bilgisi olarak ve santigrat derece şeklinde yazmaktadır. Program açıldığında ilk olarak seri haberleşme ara yüzü, haberleşme hızı, bilgilerin alınacağı düğüm kimlik numaraları ve sıcaklık uyarı eşik değeri bilgileri alınmaktadır. Çalışma yapılan sistemde haberleşme ara yüzü COM6, hız ise IRIS haberleşme hızı olan 57200 buad'dur. Connect butonuna basıldığında sistem çalışmaya başlamakta ve çıkış düğümünden gelen bilgiler alınmaya başlamaktadır. Uygulama ayrıca sıcaklık bilgisinin veri tabanına kaydedilmesi işlemi de yapmaktadır. Düğümlerden gelen ve uyarı seviyesinin altında bulunan, 2,5 sn'de periyodik olarak gelen bilgi veri tabanına her seferinde kaydedilmez. 10 ölçümden bir tanesi veri tabanına kaydedilir. Böylelikle dakikada 2 adet kayıt yapılmaktadır. 24 saatte ise 2280 adet kayıt yapılmaktadır. Aşırı ve gereksiz kayıtların önlenmesi için geçmişe yönelik veriler silinebilir. Burada dakikada 2 adet kayıt özellikle alarm durumlarında bilgilerin kaybı anlamına gelmektedir. Bu nedenle alarm koşulunda bilgi doğrudan kaydedilmektedir.

Geliştirilen uygulama ana sınıfı *WSNSerial*'dir. Bu sınıf *javax.swing.JFrame* kütüphanesini *extend* etmektedir. Ayrıca *Runnable* ve seri arayüz için *SerialPortEventListener* kütüphanelerini *implement* etmektedir. Connect butonuna basılmasından sonra çalışan *JButton1ActionPerformed* metodunun içerisinde çağrılan *connect()* metodu ile seri ara yüz bağlantısı yapılmaktadır. Bu metotta butonlardan bilgiler alınmaktadır. *serialPort.setSerialPortParams* metodu çağrılarak seri ara yüz bağlantısı yapılır. Ayarlamalardan sonra sürekli okuma yapılabilmesi *thread* çalıştırılır ve *threadler* aracılığı ile seri ara yüzden gelen bilgiler okunabilmektedir.

Her seri ara yüzden veri geldiğinde, aşağıda görülebilecek, *serialEvent* çalışmaktadır ve uygulamada *override* edilmektedir. Metot içerisindeki *InputStream.read(readBuffer)* çalıştığında *readBuffer* dizisi içerisinde 64 byte uzunluğundaki dizide okuma bilgisi bulunmaktadır. Bu dizi ayrıştırılıp duyarga okumaları elde edilmektedir. Burada duyarga okumalarını alabilmek için *readBuffer* dizisi üzerinde *for* döngüsü ile gezinme yapılmaktadır. Gezinme işleminde duyarga okuması yapan düğümün kimlik numarası aranır. Bulunan indise göre referans alınarak duyarga okuma bilgisi elde edilir. Elde edilen bilgi 2 byte olmaktadır. Daha sonra hex değer onluk (decimal) tabana dönüştürülür. Burada *calculateDegree* metodu kullanılarak sıcaklık santigrat derece olarak alınır. Alınan değer hex bilgi ve decimal bilgi olarak ekrana basılır. Bununla birlikte eğer sıcaklık eşik değerinin üzerinde ise ekranda uyarı mesajı belirmektedir.

```

while (inputStream.available() > 0) { //paket aliniyor
numBytes = inputStream.read(readBuffer); //seriden okuma
for (int v = 0; v < nodes.length; v++) {
for (int z = 0; z < 64; z++) { //paket üzerinde gezinme
if (readBuffer[z] == Integer.parseInt(nodes[v])) {
if (readBuffer[z + 12] == 0x01 || readBuffer[z + 12]){
String result1 = String.format("Node ID=%s,
Hex=%02X%02X,", nodes[v], readBuffer[z + 12], readBuffer[z +
13]);

```

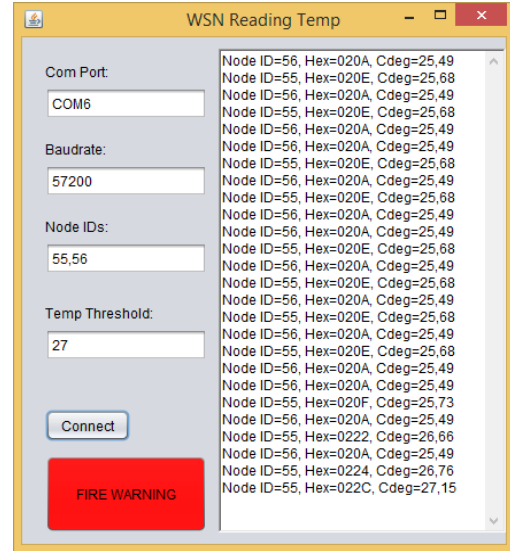
```

//hex decimale donusuyor
hexdeg = 256 * readBuffer[z + 12] + readBuffer[z + 13];
//santigrat derece elde etme
cdeg = calculateDegree(hexdeg);
result = result + result1 + result2;
txtReadData.setText(result); //ekrana bilgileri yazma
if (k == 10) { //10 okumadan sonra veri tabanına kayit
record currentRec = new record(nodes[v],
String.format("%02f", cdeg), "Normal", now.toString());
}
k++;
if (cdeg > threshold) { //sicaklik esik uzerinde
jButton2.setText("FIRE WARNING");
jButton2.setBackground(Color.red);
//uyari durumu veri tabani kaydi
record warningRec = new record(nodes[v],
String.format("%02f", cdeg), "Warning", now.toString());
} else {
jButton2.setText("");
jButton2.setBackground(Color.white);
} } } } } }

```

### V. TEST ÇALIŞMASI

Sistemin test çalışmaları esnasında, 55 ve 56 kimlik numaralı duyarga düğümlerinden alınan bilgileri ayrıştırarak bilgisayar uygulamasının örnek ekran görüntüsü Şekil 7'de verilmiştir. Şekilden de görülebileceği gibi, COM6 seri haberleşme ara yüzü kullanılmış, 57200 baud haberleşme hızı seçilmiş, uyarı eşik değeri olarak 27 C seçilmiş, ekranda elde sıcaklık bilgileri de verilmektedir. Ayrıca uyarı koşulu da oluşturulmuş ve yangın uyarısı da şekilden görülebilmektedir.



Şekil 7. Bilgisayar uygulaması ekran görüntüsü.

Geliştirilen sistem, Kent Kart Ege Elektronik A.Ş. ve Ege Üniversitesi Uluslararası Bilgisayar Enstitüsü kütüphanelerinde sınanmıştır. Düğümler her kitaplığa 1 adet gelecek şekilde yerleştirilmiştir. Çıkış düğümü ise kütüphane bilgisayarında çalışacak şekilde kurulmuştur. Kapalı ortamlarda cam kenarında bulunacak düğümlerde sıcaklık diğer düğümlere göre daha yüksek olma ihtimali yüksektir. Ayrıca klima ve havalandırma önlerine yerleştirilecek olan düğümlerde de daha düşük sıcaklık okumaları gerçekleştirilmektedir. Bu nedenlerle sıcaklık uyarı eşik değerinin ayarlanması kritiktir. Testlerde bu

durumlar göz önünde bulundurularak düğümler yerleştirilmiş ve ölçüm sonuçları toplanmıştır. Yangın koşulunun yaratılması için düğüm yakınında ateş yakılmış ve 3 sn'nin altında bir gecikme (2,5 sn-3 sn arası) ile uyarı durumuna geçildiği gözlenmiştir. Bu gecikme de zaten sistemin tasarımından kaynaklı olan en fazla gecikmedir. Testler esnasında bilgi gönderme zaman aralıkları, veri tabanına kaydetme zaman aralıkları, sıcaklık eşik değeri gibi birçok parametre optimize edilmiştir.

## VI. SONUÇLAR VE TARTIŞMA

Edwards United Technologies firmasının ürünü olan EST3 sisteminin [21] ağ cevap hızının en iyi koşulda 3 sn civarında olduğu belirtilmiştir. Elde edilen sonuçlarda geliştirilen sistemin de 3 sn'nin altında alarm oluşturduğu ölçülmüştü. Bununla birlikte geleneksel duman duyarları içeren sistemlerde [22] en iyi sonuç veren koşullarda bile 30sn civarında alarm üretimi olduğu görülmektedir. Böylelikle yüksek maliyetli, yabancı kaynaklı, kablolu sistemlerin yerine benzer performansta bir TDA önerilmektedir.

Bu çalışmada, IRIS XM2110 düğümü ve MDA100CB duyar kartı kullanılarak sıcaklık okuması uygulaması geliştirilmiştir. Tasarlanan tüm sistemin donanım ve yazılım yapısı ayrıntılı olarak anlatılmıştır. Düğümler üzerinde çalışan uygulamalar TinyOS geliştirme ortamı ve NesC dili kullanılarak geliştirilmiştir. Sıcaklık okuma yapan düğümler yıldız ağ yapısında, tek atlamalı olacak şekilde, okuma bilgilerini çıkış düğümüne iletmektedir. Çıkış düğümü aldığı bilgileri sanal seri ara yüzü aracılığı ile bilgisayara göndermektedir. Geliştirilen bir masa üstü uygulaması ile seri port paketleri alınmakta ve ekrana alınan bilgiler yazdırılmaktadır. Uygulama aracılığı ile alınan sıcaklık eşik değerine göre kullanıcıya uyarı mesajı verilmektedir. Sıcaklık ölçümleri düğüm bilgilerine göre veri tabanına kaydedilmektedir. Bir kitaplık ortamında sistem denenmiş ve tasarım iyileştirme çalışmaları yapılmıştır.

Gelecek çalışmalarda, çok atlamalı ağ ihtiyacı duyan, büyük ölçekli düğüm sistemi incelemesi amaçlanmaktadır. Bununla birlikte, geliştirme prosedürlerini iyileştirmek ve karmaşıklığı azaltmak için sistemin geliştirilmesinde model tabanlı geliştirme teknikleri incelenecektir [23].

## TEŞEKKÜR

Yazarlar, bu çalışmaya verdiği destekler için Ege Üniversitesi 17-UBE-002 nolu bilimsel araştırma projesine ve Kent Kart Ege Elektronik A.Ş.'ye teşekkür eder.

## KAYNAKÇA

- [1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, "A Survey on Sensor Networks," *IEEE Commun. Mag.*, vol. 40(8), pp. 102-114, 2002.
- [2] O. Dağdeviren ve V.K. Akram, "TinyOS Tabanlı Telsiz Duyar Ağları için Bir Konumlandırma ve k-Bağlılık Denetleme Sistemi," *Bilişim Teknolojileri Dergisi*, vol. 10-2, pp. 139-152, 2017.
- [3] J. Liu, Y. Tian and H. Qi, "Wireless Sensor Networks for Fire Detection in the Library Environment," *Applied Mechanics and Materials*, vol. 166-169, pp. 2684-2687, 2012
- [4] S. Rao, G. K. Nithya and K. Rakesh, "Development of a wireless sensor network for detecting fire and Gas leaks in a collapsing building," *Fifth*

- International Conference on Computing, Communications and Networking Technologies*, 2014, pp. 1-7.
- [5] S. Duan, K. Cheng, Q. Ge and C. Wen, "Wireless Intelligent Fire Fighting Systems Software Platform R&D," *2011 International Conference on Internet of Things and 4th International Conference on Cyber, Physical and Social Computing*, 2011, pp. 94-99.
- [6] T. Islam, H.A. Rahman and M.A. Syrus, "Fire detection system with indoor localization using ZigBee based wireless sensor network," *International Conference on Informatics, Electronics & Vision*, 2015, pp. 1-6.
- [7] Y. Hongyan, "Research and Implementation of Historical Building Wireless Fire Monitoring System," *Ninth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, 2013, pp. 575-577.
- [8] Z.A. Khan, E.P. Freitas, T. Larsson and H. Abbas, "A Multi-agent Model for Fire Detection in Coal Mines Using Wireless Sensor Networks," *12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications*, 2013, pp. 1754-1761.
- [9] Y. Chen, D. Zhang, J. Chen, K. Liu and F. Yang, "A new fire protection system of power cable tunnel based on wireless sensor network monitoring system," *IEEE 11th Conference on Industrial Electronics and Applications*, 2016, pp. 1103-1106.
- [10] G. Demin, L. Haifeng, J. Anna and W. Guoxin, "A forest fire prediction system based on rechargeable wireless sensor networks," *4th IEEE International Conference on Network Infrastructure and Digital Content*, 2014, pp. 405-408.
- [11] A. Herutomo, M. Abdurrohman, N.A. Suwastika, S. Prabowo and C.W. Wijiutomo, "Forest fire detection system reliability test using wireless sensor network and OpenMTC communication platform," *3rd International Conference on Information and Communication Technology*, 2015, pp. 87-91.
- [12] L. Guo, W. Wang, G. Wang and J. Cui, "Research and implementation of forest fire early warning system based on UWB wireless sensor networks," *Second International Conference on Communication Systems, Networks and Applications*, 2010, vol. 1, pp. 176-179.
- [13] P. Bolourchi and S. Uysal, "Forest Fire Detection in Wireless Sensor Network Using Fuzzy Logic," *Fifth International Conference on Computational Intelligence, Communication Systems and Networks*, 2013, pp. 83-87.
- [14] H. Soliman, K. Sudan and A. Mishra, "A smart forest-fire early detection sensory system: Another approach of utilizing wireless sensor and neural networks," *SENSORS IEEE*, 2010, pp. 1900-1904.
- [15] V. Devadevan and S. Suresh, "Energy Efficient Routing Protocol in Forest Fire Detection System," *6th International Conference on Advanced Computing*, 2016, pp. 618-622.
- [16] G. Lu and W. Xue, "Adaptive Weighted Fusion Algorithm for Monitoring System of Forest Fire Based on Wireless Sensor Networks," *Second International Conference on Computer Modeling and Simulation*, 2010, vol. 4, pp. 414-417.
- [17] IRIS XM2110 Wireless Measurement System, datasheet, Crossbow Technology, Inc, 6020-0124-01 Rev A, 2006.
- [18] J.A. Gutierrez, E.H. Callaway and R.L. Barrett, *Low-Rate Wireless Personal Area Networks: Enabling Wireless Sensors with IEEE 802.15.4*. Standards Information Network IEEE Press, NY: New York, 2007.
- [19] MTS/MDA Sensor Board Users Manual, Crossbow Technology, Inc, 7430-0020-04 Rev B, 2006.
- [20] YSI 44006 datasheet, YSI Precision Temperature Group, 1998.
- [21] EST3 Fire and Life Safety System datasheet, Edwards United Technologies, 2012.
- [22] J.D. Averill, N.P. Bryner, R.W. Bukowski, T.G. Cleary, E.D. Kuligowski, R.D. Peacock, P.A. Reneke and W.D. Walton, "Performance of home smoke alarms: Analysis of the response of several available technologies in residential fire settings", *NIST Technical Note 1455-1*, 2010.
- [23] Ç. Durmaz, M. Challenger, O. Dagdeviren and G. Kardas, "Modeling Contiki-based IoT Systems," *17th Symposium on Languages, Applications and Technologies*, 26-27 June 2017 (Accepted).