

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ BİLİŞİM ENSTİTÜSÜ

**DÜŞÜK MALİYETLİ VE KAYNAKLARI VERİMLİ KULLANABİLEN
SÜREKLİ ÖĞRENEBİLEN AKILLI CİHAZ ÇEKİRDEĞİ**

YÜKSEK LİSANS TEZİ

Onur AKDEMİR

Bilgisayar Bilimleri Anabilim Dalı

Bilgisayar Bilimleri Programı

MAYIS 2014

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ BİLİŞİM ENSTİTÜSÜ

**DÜŞÜK MALİYETLİ VE KAYNAKLARI VERİMLİ KULLANABİLEN
SÜREKLİ ÖĞRENEBİLEN AKILLI CİHAZ ÇEKİRDEĞİ**

YÜKSEK LİSANS TEZİ

**Onur AKDEMİR
(704101009)**

Bilgisayar Bilimleri Anabilim Dalı

Bilgisayar Bilimleri Programı

Tez Danışmanı: Doç. Dr. Turgay ALTILAR

MAYIS 2014

İTÜ, Bilişim Enstitüsü'nün 704101009 numaralı Yüksek Lisans Öğrencisi **Onur AKDEMİR**, ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı “**DÜŞÜK MALİYETLİ VE KAYNAKLARI VERİMLİ KULLANABİLEN SÜREKLİ ÖĞRENEBİLEN AKILLI CİHAZ ÇEKİRDEĞİ**” başlıklı tezini aşağıda imzaları olan jüri önünde başarı ile sunmuştur.

Tez Danışmanı : **Doç. Dr. Turgay ALTILAR**

İstanbul Teknik Üniversitesi

Jüri Üyeleri : **Prof. Dr. Coşkun SÖNMEZ**

İstanbul Teknik Üniversitesi

Doç. Dr. Ali Gökhan YAVUZ

Yıldız Teknik Üniversitesi

Teslim Tarihi : **5 Mayıs 2014**
Savunma Tarihi : **26 Mayıs 2014**

Eşime,

ÖNSÖZ

Yüksek lisans eğitimim boyunca bana destek olan eşime ve arkadaşlarıma, tez sürecimde bana yol gösteren danışmanım Doç. Dr. Turgay ALTILAR'a teşekkür ederim.

Mayıs 2014

Onur AKDEMİR
(Bilgisayar Mühendisi)

İÇİNDEKİLER

Sayfa

ÖNSÖZ.....	vii
İÇİNDEKİLER	ix
KISALTMALAR	xi
ÇİZELGE LİSTESİ.....	xiii
ŞEKİL LİSTESİ.....	xv
ÖZET.....	xvii
SUMMARY	xix
1. GİRİŞ	1
1.1 Tezin Amacı	1
1.2 Literatür Araştırması	2
1.3 Hipotez	3
2. AMAÇ VE MOTİVASYON.....	5
2.1 Amaç	5
2.2 Kaynak Verimli Mikroişlemciler ve Uygulamalar.....	5
2.2.1 Mikroişlemci çeşitleri	5
2.2.2 TI stellaris LM4F120XL geliştirme kartı.....	6
2.2.3 TI-Stellaris bellek teknolojisi.....	7
2.2.4 Uygulamalar	8
2.3 Otonomi ve Öğrenme	8
2.3.1 K-means algoritması ve devamlı öğrenme.....	9
2.3.2 Otonom sistemler	9
2.4 İşletim Sistemleri.....	10
2.4.1 FreeRTOS	11
2.5 Motivasyon.....	12
3. GERÇEKLEŞTİRME	15
3.1 Geliştirme Kartı ve Çevre Birimleri.....	15
3.2 FreeRTOS Özellikleri	17
3.3 İşlemler ve Görevleri.....	17
3.3.1 Öğrenme işlemi	18
3.3.1.1 Veri optimizasyonu	19
3.3.1.2 Flash bellek üzerinde verilerin tutulması	20
3.3.1.3 Veri tekrarının kod içerisinde en aza indirilmesi	20
3.3.1.4 Bellek kullanımının minimum seviyeye düşürülmesi.....	21
3.3.1.5 K-means algoritması	21
3.3.2 Sistem istatistik işlemi	22
3.3.3 USB işlemi	22
3.3.4 Diğer işlemler.....	22
3.4 Merkez İstasyon Yazılımı	22
4. SONUÇLAR	25
5. GELECEK ÇALIŞMALAR VE ÖNERİLER.....	27
KAYNAKLAR	29

ÖZGEÇMİŞ.....	31
----------------------	-----------

KISALTMALAR

RTOS	: Gerçek Zamanlı İşletim Sistemi
SOC	: System on Chip
USB	: Universal Serial Bus
UART	: Universal Async
IoT	: Internet of Things
CSS	: Code Composer Studio
ÇD	: Çevresel Denetleyici

ÇİZELGE LİSTESİ

	<u>Sayfa</u>
Çizelge 1.1 : Geliştirme kartı arayüzü	17
Çizelge 2.1 : Platform karşılaştırma çizelgesi	25

ŞEKİL LİSTESİ

Sayfa

Şekil 1.1 : TI stelarris LM4F120XL.....	7
Şekil 1.2 : K-means algoritması örnek adımları.....	9
Şekil 1.3 : İşlem akış şeması.....	12
Şekil 1.4 : Geliştirme kartı.....	16
Şekil 1.5 : Bağlantı şeması.....	16
Şekil 1.6 : Cihaz alarm oluşturma akışı.....	19
Şekil 1.7 : Cihaz öğrenme durum geçiş diyagramı.....	19

DÜŞÜK MALİYETLİ ve KAYNAKLARI VERİMLİ KULLANABİLEN SÜREKLİ ÖĞRENEBİLEN BİR AKILLI CİHAZ ÇEKİRDEĞİ

ÖZET

Günümüzdeki teknolojik gelişmeler, yüksek işlem yapma gücüne sahip ve ucuz bilgisayarların geliştirilmesine olanak sağlamaktadır. Çok düşük enerji ihtiyacı olan bu ucuz bilgisayarlar gelecekte insanların yerini alabilecek düzeye gelecektir. “İnsan-İnsan” etkileşimli sistemler yerine “İnsan-Cihaz” ve hatta “Cihaz-Cihaz” etkileşimli sistemler geliştirilmesine olanak sağlanabilecektir. Bu çalışma, orman yangınlarının oluşmasını önlemek için, orman yangınları oluşmadan otonom bir şekilde karar alabilen bir cihazın merkez bir istasyondaki görevlileri uyarması ve merkez istasyondaki görevlilerin karar hakkındaki olumlu-olumsuz sonuçları cihaza göndermesi ile sürekli öğrenebilen örnek bir cihazın geliştirilmesi üzerine planlanmıştır. Ayrıca bu tez kapsamında cihazlardan bilgi bekleyen bir merkez istasyon yazılımı geliştirilmiştir. Bu tez kapsamında çok düşük enerji ihtiyacı olan TI LM4F120H5QR mikroişlemcisi, Texas Instruments firmasının geliştirdiği LM4F120XL geliştirme kartı yardımıyla kullanılmıştır. Bu geliştirme kartı üzerinde bulunan USB ve UART arayüzleri ile dış ortamla iletişim sağlanmıştır. Orman yangınları ile ilgili veri seti (<http://archive.ics.uci.edu/ml/datasets/Forest+Fires>) ve dış ortamdan alınan veriler k-means algoritması kullanılarak yangın tehlikesi olup olmadığı kararını alır, bu karar yangın tehlikesi olması durumunda Merkez istasyona bildirilir. K-means algoritmasının kullandığı veri cihazdaki düşük bellek kapasitesi sebebiyle, Flash bellek üzerinde saklanmıştır. Yeni eklenecek tüm verilerde yine Flash bellek üzerine yazılacaktır. Deneyimler sonucunda Flash bellek hızının bu algoritmaların çalışması için yeterli olduğu görülmüştür.

Bu cihazda FREERTOS işletim sistemi, işlemlerin doğru ve zamanında çalışmasını sağlamıştır. FREERTOS işletim sisteminin sunduğu ancak bu cihazda çalışmayan özellikler yine bu tez kapsamında gerçekleştirilmiştir. Üzerinde bir işletim sistemi bulunan bu cihaz gelecekte geliştirilebilecek sistemler içinde bir altyapı oluşturmaktadır. Yeni bir görevin eklenmesi ile cihaz daha farklı görevleri yerine getirebilecektir. K-means algoritması için kullanılan yöntemler göz önünde bulundurularak geliştirilecek yeni algoritmalar sayesinde, sadece pil veya güneş enerji panelleri yardımıyla uzun yıllar çalıştırılabilir ve insanların ulaşmasının zor ve gereksiz olduğu durumlarda, insanların yerini alabilecek cihazlar geliştirilebilecektir.

Tez kapsamında geliştirilen sistemin var olan fiziksel belleğin sadece %30’unu ve flash belleğin sadece %10 luk kısmını kullandığı görülmüştür. Sadece orman yangınları değil, vahşi doğa takibi, nükleer enerji tehlikelerinin önceden sezilmesi gibi birçok uygulama için kapasitesi olan bu cihaz, gelecekte bir çok alanda kullanılabilir.

LOW-COST AND RESOURCE-AWARE CONTINUOUS LEARNING DEVICE CORE OF THING

SUMMARY

Today's technological advances allows the development of inexpensive and high power computers. These cheap computers which have very low energy consumption will make decisions for people in the future. Instead of "Human - Human " interactive systems "Human - device " and even " device - device " interactive systems will be available. This study shows how to prevent forest fires from occurring in an autonomous way that device gives decisions without the need for a central station. Device alerts the staff of the central station who gives feedback as positive or negative after taking some actions on alert. These feedbacks help the device to improve decision making capability. Also within the scope of this thesis, a central station waiting for information from the device software has been developed. In this thesis, TI Stellaris LM4F120H5Q microprocessor which developed by Texas Instruments with very low energy requirements is used. Development of a hardware platform was out of scope, so EK-LM4F120XL development board was chosen for platform. USB and UART interfaces, which are located on the development board allow communication with external environment. Forest fires related data set (<http://archive.ics.uci.edu/ml/datasets/forest+fires>) and the data received from the external environment processed with a modified k-means algorithm. After processing decision whether or not a fire hazard given, if a fire hazard exists, just this hazardous decision is sent to central station. The data which is processed by K-means algorithm is stored on flash memory due to low memory capacity. All data which is a new observation will be added again in Flash memory. The speed of Flash memory has been found to be sufficient to run K-means algorithm. FreeRTOS operating system on this device has provided accurate and timely work. This core, which is based on an operating system constitutes an infrastructure which can be used to develop devices in the future. New task addition is very simple with the help of RTOS which makes device able to fulfill different tasks . A new k-means algorithm developed which use efficient RAM and Flash memory, is able to run for many years using battery or solar panels.

Thesis scope of the system developed in only 30% of the available physical memory and Flash memory uses just 10% of this was seen. Not only forest fires, wildlife tracking, foreseeable hazards such as nuclear energy capacity, which for many applications , this device would be used in many areas in the future. The IoT is the concept of networking real-world objects and is regarded as the next logical generation of the Internet. It is predicted that hundred billion devices to be connected in near future, so research on both hardware and connection methods are so popular. These connected devices need some behaviour to act as useful ends. Many challenges must be faced to make this devices intelligent. Low resources and limited energy that is supplied by small capacity batteries contradict with off-the-shelf learning algorithms.

Algorithms for IoT must be implemented without wasting of any resource. IoT cover all digital devices that can connect to internet and interact with humans. This communication can be between a human and IoT or IoT-IoT. IoT enables collective intelligence which eliminates human factor for decision making on multiple sensors. IoT devices give final decision and just result is sent to humans. Collective intelligence is not possible without intelligent IoT. It is obvious that management of this devices could be impossible for humans in near future when IPv6 enabled. Some intelligence can help both for management effort and automating the system.

Artificial Intelligence software is an active research for several decades. Very useful libraries that implemented several AI and pattern recognition algorithms have widespread usage. Weka, matlab toolbox and other small scale libraries exist. Main disadvantage of these libraries is that they are too big for a device which has low-resources. Besides this, resource-aware intelligent software development must be done carefully to allow space for other tasks. Intelligent algorithms are based on the prior knowledge on specific topics. Starting with a known data on a special topic and further create new knowledge or extract knowledge from this data can be implemented with pattern recognition techniques. It is obvious that making machines as smart as humans not possible. So simulating human behaviour is the only way to think like a human. Alan Turing's proposal that the question " Can machine think?" can be replaced with the question "Can machines do what we (as thinking entities) can do?". This idea lead us to learning algorithms which starts with a prior knowledge and predicts some rules from new observations with the help of this prior knowledge. Learning from scratch for machines need special efforts. An IoT without prior knowledge has to built some base knowledge with the help of observations that are collected from sensors. Some trust level must be determined until that IoT could make trusted decisions. It maybe impossible for some knowledge to find a suitable trust level , some decisions must be done on this subject. Because of open issues exist about learning from scratch, in this paper this method ignored. Proposed solution uses k-means algorithm which is a machine learning technique. Machine learning, a branch of artificial intelligence, concerns the construction and study of systems that can learn from data. More formal definition: "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E". Our model applies for situations where decisions limited to discrete number of status. Yes/No like situations with appropriate data provided prior to run, easily predicted by FILID. Learning evolves with the help of Center which could be run by a Human. Therefore, human intuition factor somehow added to FILID. FILID runs on predefined intervals and executes decision task. This task normally just checks data which come from sensors on that FILID. Decision task works without any intervention from another human or any other computer. Fully automated decision's results send to Center only if there exists an alarm situation. Communication needs lowered by this design so energy consumption. Self-living FILID core decisions based on prior data. Prior data and sensors identify the new FILID thing and task. Alarm status that is sent to Center needs some feedback for continuous learning. A human whose job is to monitor Center, feedback to alarms so learning evolves. Same alarm situation may sent to Center more than one which is not an error but feedback for the alarm must be provided at least one by Center. With the help of feedback result and decision that created the alarm situation, new observation added to knowledge base. After that learning phase followed by normal run which listens environment and checks environment to give new decisions.

Future devices seem to have limited cpu power, low capacity storage and expected to be long-life, so new designs must be proposed. In this paper , it is shown that machine can learn new observations from experiments on low power devices. Human interaction is needed for this devices, but in near future machine teaching to machine concept seems possible. Some improvements on communication technology, and embedded programming techniques also required. Smart watches, intelligent home systems, intelligent shopping machines are some examples of our future.

1. GİRİŞ

Bu tezde gelecekte yaygın bir şekilde kullanılacağı düşünülen “Internet of Things” cihazlarının tanıtımı ardından, tam otonom bir cihazın tasarımı ve gerçekleşmesi açıklanacaktır. Bu tasarım farklı cihazların geliştirilmesi içinde kullanılabilir.

1.1 Tezin Amacı

Günümüzde insanların yapabildiği birçok işlemi cihazlar gerçekleştirmektedir. Bu cihazların karar alabilmeleri için çoğu zaman insan etkileşimine ihtiyaç duyulmaktadır. Bu cihazların tam otonom olması ile ortaya çıkabilecek yeni cihazlar hayatın her alanında insanların ihtiyaç duyduğu birçok konuda yardımcı olabilecektir. Geleceğin küçük cihazları her yerde ve her zaman bağlantı kurulabilir durumda çalışacaktır, bu cihazların tam otonom olarak çalışması gelecekte hayatımızı daha çok kolaylaştıracaktır. Thing cihazlar kendi kimlik bilgileri ve sanal kişilikleriyle daha akıllı ortamlarda sosyal, çevresel ve insan ortamlarında bulunacaklardır [1]. “Geleceğin dünyasının thing’lerin otonom bir şekilde bilgisayarlar ve birbirleriyle rahatlıkla iletişime geçebilen insan yararına servisler sunabilen ürünlerden oluşacağı “ düşünülmemektedir [2].

[2] IoT olarak tanımlanan geleceğin küçük boyutlu, her zaman bağlantı kurulabilir akıllı cihazlarının ortak noktası çok düşük kaynaklara ve enerji kullanımına sahip olmalarıdır. Bu sebeple bu cihazlara yönelik geliştirilecek yazılımlarında kaynakları en verimli şekilde kullanması hedeflenmelidir.

Geçmişte çok büyük projelerin çok düşük kapasiteli bilgisayarlar ile başarılılabildiği görülmüştür [3]. Bundan 10-15 yıl öncesine kadar uzay araçları, uçak yazılımları,vb. birçok yazılım bugün için çok düşük seviyede kapasiteli bilgisayarlarla gerçekleştirilebilmiştir. Günümüzdeki araçların kapasiteleri yükselse, geleceğin teknolojisi “thing” lerin düşük kapasitesi, bu düşük kapasitelerle ne tür cihazlar geliştireceğimizi düşünmemizi gerektirmektedir. Kullanmakta olduğumuz Embedded Linux, Windows CE gibi işletim sistemleri gerçekten “thing” ler için uygun mu? Çok

düşük kapasiteli bu cihazlar hangi yazılımları çalıştırabilirler? Bu düşük kapasiteli cihazlar tam otonom olarak çalışabilir mi? [4] Az kapasiteli bu cihazlar yazılım geliştirme yöntemlerinin değiştirilmesi ile daha verimli kullanılabilir mi? Bu tez kapsamında bu soruların cevabı araştırılacak ve gelecekte genel bir altyapı oluşturabilecek örnek bir uygulama geliştirilecektir.

1.2 Literatür Araştırması

Kaynak yetersiz ve tam otonom sistemler üzerinde çalışmalar var olsada, küçük boyutlu ve tam otonom cihazlar önemli bir çalışma alanı değildir. Bu alanda yapılan bazı çalışmalar olsada, kaynakların verimli kullanılmasından çok, hızlı ürün çıkarmaya odaklanılmıştır. Hem tam otonomi hem de kaynak yetersizliği ön plana alan çalışmalar yaygın değildir. [5] ile yapılan çalışmada, görev bilgisayarı olarak ARM9 468 MHz gömülü mikroişlemci ile 128 M bellek üzerinde Linux İşletim sistemi seçilmiştir. Bu sistem otonom görevleri yerine getirebilse de kullanılan işlemci enerjisi sistemi sürekli bir enerji kaynağına bağımlı yapmaktadır. [4] Adriana Wilde et. al. çalışmasında çok düşük kapasiteli LM3S6965 cihazı üzerinde bir sistem geliştirmiş ve çok düşük boyutlarda yüksek işlem gücüne ulaşabilmiştir. Ortaya çıkarılan çalışma sayesinde gelecekte hayatımıza girmesi beklenen “thing” lerin düşük kapasitede yüksek verim ihtiyaçlarına bir çözüm önerilmiştir. [2] Luigi Atzori et. al. geleceğin “thing” cihazlarının hayatımızın her alanında kullanılacağını, artık cihazların insanlara yol gösterebilecek kapasitelere ulaşabileceğini belirtmektedir. [5] Rahul Shah et. al. mobil cihazları göz önünde bulundurarak k-means algoritmasını bu cihazlar üzerinde çalışmasının yöntemlerini araştırmıştır. Veri akışının sürekli olduğu bu cihazlarda kaynağın yetersiz olduğu durumlarda dahi algoritmanın çalıştığı görülmüştür. [6] Xu Zhen et. al. Cortex M3 üzerinde akıllı bir ev sistemi geliştirmiştir. Yapılan bu çalışma daha yüksek kapasiteli ARM7 mikroşlemcileri ile karşılaştırılmıştır. Bu çalışma ile Cortex M3 mikroşlemcisinin gelecekteki akıllı cihazlar için kullanılabilirliği bulunmuştur. [7] Vivek Tiwari et. al. sadece derleme şekillerini ve derleme sırasında oluşacak optimizasyonları doğru yöneterek daha düşük bellek ihtiyacı olan yazılımların geliştirilebileceğini göstermiştir. Intel bilgisayarlar üzerinde %40 a varan enerji verimi sağlanmıştır.

1.3 Hipotez

Bu tezde çok düşük kapasiteli ve enerji ihtiyacı çok az olan LM4F1205HQR işlemcisi üzerinde, k-means algoritmasını kullanarak orman yangınları sezebilen ve bir tehlike anında bu bilgiyi merkez bir istasyona iletebilen, merkez istasyonun cevabı doğrultusunda sürekli öğrenmeye devam edebilen bir sistem tasarlanmıştır. Bu cihaz üzerinde FreeRTOS işletim sistemi kullanılarak, kaynakların verimli kullanılması hedeflenmiş ve bu önerilen altyapıyı kullanacak gelecek projelerin yeni işlemleri rahatlıkla ekleyebilmeleri amaçlanmıştır. Ortaya çıkacak bu sistem en fazla 300 mA lik enerji ihtiyacı ile düşük boyuttaki güneş panelleri ile uzun yıllar çalışabileceği varsayılmaktadır [7]. K-means algoritması gibi çok yüksek bellek kullanımı olan bir algoritmayı bu cihaz üzerinde çalıştırabilmek için flash bellek ve fiziksel bellek ortak kullanılmıştır.

Cihazın aldığı kararları merkez istasyona iletmesinin ardından merkez istasyondaki verilen kararın cihaza iletilmesi ile aldığı kararı flash bellekte olumlu veya olumsuz olarak saklayan cihaz sürekli öğrenebilecek ve o ortamı en iyi tanıyan bir duruma gelecektir. Bu sayede insanların yapmasına olanak olmayan görevleri cihazların rahatlıkla yapabileceği gösterilecektir. Düşük enerji ve yüksek verim ile bu cihazlar geleceğimizi oluşturacaktır.

2. AMAÇ VE MOTİVASYON

2.1 Amaç

Bu tezin amacı, günümüzde var olan teknolojilerin geleceğin küçük ve akıllı cihazları için hangi noktada olduğunu bulmak, bulunan yazılım ve donanım ürünlerini kullanarak otonom bir cihaz ve alt yapı geliştirerek bu cihazların kullanılabilir durumda olduğunu göstermektir.

2.2 Kaynak Verimli Mikroişlemciler ve Uygulamalar

Günümüzde teknolojinin hızla ilerlemesi ile çok hızlı işlemciler çok uygun fiyatlarla alınabilmektedir. İşlemcilerin fiyatları yanı sıra kullandıkları enerji, boyutları ve yazılım çalıştırma yöntemleri de değişmiştir.

2.2.1 Mikroişlemci çeşitleri

Yaygın olarak kullanılmakta olan mikroişlemciler, düşük enerji ihtiyaçları sebebiyle birçok taşınabilir üründe kullanılmaya başlanmıştır. Özellikle soğutma ihtiyacı olmayan işlemciler birçok cihazda yaygın olarak kullanılmaktadır.

Günümüzde teknolojinin hızla ilerlemesi ile çok hızlı işlemciler çok uygun fiyatlarla alınabilmektedir. İşlemcilerin fiyatları yanı sıra kullandıkları enerji, boyutları ve yazılım çalıştırma yöntemleri de değişmiştir. Düşük enerji ihtiyacı ve yüksek verimli ARM mikroişlemciler gelecek teknolojisinin gideceği yönü belirleyecektir. ARM mikroişlemciler tasarımlarının ilk yıllarından itibaren son kullanıcıya SOC olarak ulaşmış ve birtek cihazla birçok yeni ürünün üretilebilmesine olanak sağlamışlardır.

ARM mikroişlemciler 1 Watt altında enerji tüketimleri, buna bağlı olarak soğutma ihtiyaçlarının çok az olması ve yüksek performansları ile günümüzde birçok taşınabilir cihazda tercih edilmektedir. Cortex-A işlemciler en güçlü ARM mikroişlemcileri olarak, gelişmiş işletim sistemlerini çalıştırabilmektedirler. Linux, Windows gibi bellek koruma gerektiren işletim sistemleri bu cihazlar üzerinde performanslı bir şekilde çalışmaktadır. Akıllı telefonlar, tablet bilgisayarlar, digital

televizyonlar, sunucular ve hatta kurumsal şirketlerde dahi kullanılabilirler. 32-bit ve 64-bit olan işlemciler tekli ve çoklu işlemci yongası sayesinde birçok alanda yaygın bir şekilde kullanılmaktadır. Bu işlemciler diğer şirketlerin ürettiği işlemcilere göre verimli olsada, küçük boyutlu “thing” ler için çok enerji ihtiyacına sahiptirler.

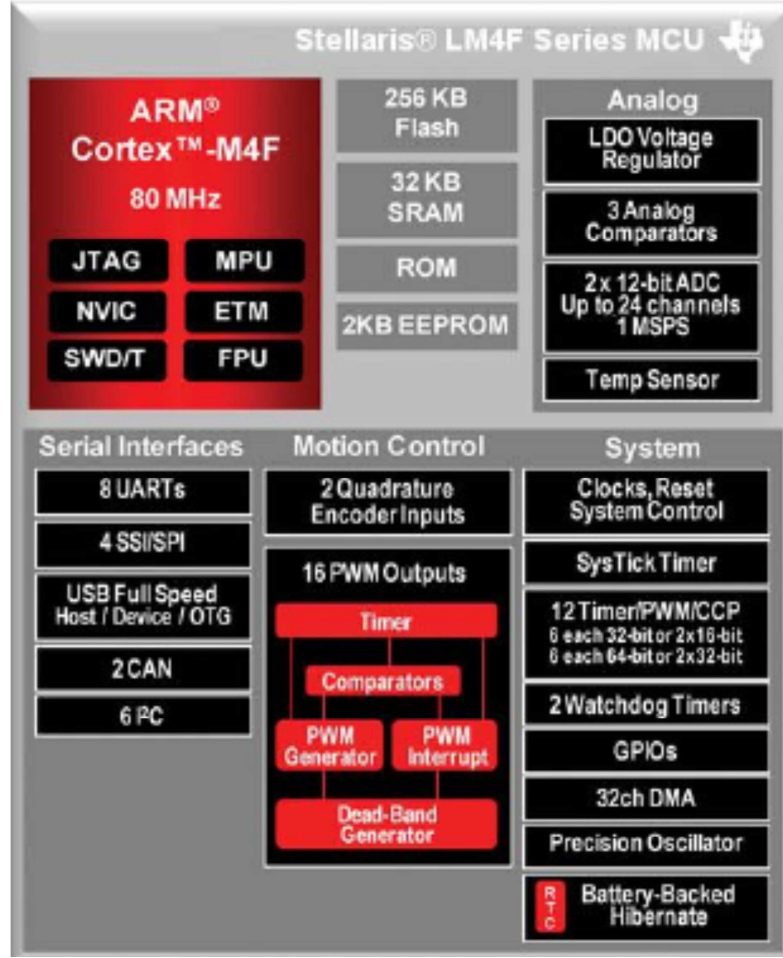
Cortex-R mikroişlemciler gömülü gerçek zamanlı işlemler için tasarlanmışlardır. Üzerlerinde bir gerçek zamanlı işletim sistemi çalıştırabilecek kaynaklara sahip olan bu işlemciler, bellek koruma teknolojisi yardımıyla bir işletim sisteminde bulunması gereken tüm özellikleri çalıştırabilmektedir. Otomobil kontrol sistemleri, kablolu ve kablosuz sensor ağları, büyük veri depolama birimleri, ağ ürünleri gibi birçok alanda kullanılmaktadırlar. Bu mikroişlemciler performans, kararlılık, düşük boyut ve verimli enerji ile tasarımcılara doğru çözümler sunmaktadır.

Cortex-M mikroişlemciler, çok düşük enerji ihtiyacı olan cihazlar için mükemmel bir sonuç çıkarmaktadır. Önceden atanmış görevleri yerine getirmek için doğru bir seçim olan bu işlemci, yeni donanım bağlanması için çok uygun değildir. Bu özellikleri ile “Thing” ler için çok iyi bir çözüm sunan ve kaynakları gereksiz yere harcamayan bu işlemci eklenecek donanım özellikleri ile birçok alanda kullanılabilir. Ancak Cortex-M işlemcilerinin daha çok kontrol amaçlı yazılımlarda kullanılması ve bu cihazların kaynaklarını gereksiz harcamaktadır. Bu cihazlar üzerinde çalıştırılacak doğru işletim sistemleri ile performans artırımı sağlanabilir ve düşük güç avantajı sayesinde bu cihazlar geleceğin ürünlerini oluşturabilir.

2.2.2 TI stellaris LM4F120XL geliştirme kartı

Bu tez çalışmasında TI serisi LM4F120XL geliştirme kartının seçilmesinin en önemli nedeni, benzer güçteki diğer mikroişlemcilere göre daha iyi geliştirme olanakları sunması, çok yetenekli kütüphanelere sahip olması; bunların yanında çok düşük bir fiyata satın alınabilmesidir. Bu işlemcinin göze çarpan bazı özellikleri şu şekildedir;

- 80 Mhz işlemci Hızı / 100 DMIPS performans
- uDMA, GPTM, WDT, Hibernation Modül, JTAG hata ayıklama
- 256 KB Flash bellek
- 32 KB SRam bellek



Şekil 1.1 : TI stelarris LM4F120XL.

2.2.3 TI-Stellaris bellek teknolojisi

LM4F120XL işlemcilerin üzerinde bulunan 32 KB SRAM, oku-değiştir-yaz işlemlerinin uzun sürmesi sebebiyle, ARM işlemcilerinde ortaya atılan “bit-banding” teknolojisini kullanmaktadır. Bu teknoloji sayesinde bellek bölgelerine bit seviyesinde adres tanımları yapabilmek ve atomic operasyonlar mümkün olmaktadır.

LM4F120XL işlemcisinde bulunan ROM bellek birçok işlem için gerekli olan yardımcı kütüphaneleri üzerinde barındırmaktadır. Bu sayede Flash bellek ve SRAM bölgesinden gereksiz yer kullanılmamaktadır. ROM üzerinde bulunan yazılımlar şu şekildedir.

- Stellaris cihaz sürücü kütüphanesi
- Stellaris cihaz ilklendirici
- Kontrol ve kriptoyardımcı fonksiyonları

Bu yazılımlar tamamen ücretsiz olup, TI Stellaris serisi her cihazla birlikte sunulmaktadır. Cihaz sürücü kütüphanesi, mikroişlemcinin üzerinde bulunan çevre birimlerinin iklendirilmesini ve kontrolünü sağlar.

2.2.4 Uygulamalar

Mikroişlemci fiyatlarının çok düşük olması uygulamalarda kullanılacak mikroişlemci seçiminide etkilemektedir. Mikroişlemcilerin bir cihaz için uygun olup olmadığından çok uygulamaların daha kolay geliştirilmesi ön planda tutularak mikroişlemci seçimi yapılmaktadır. Özellikle birçok yardımcı kütüphaneyi çalıştırabilen işletim sistemlerinin seçilmesiyle birlikte, tasarımın kalan kısımlarında bu işletim sisteminin harcadığı kaynakları da göz önünde bulundurarak, daha yüksek güçte ve enerji sarfiyatı çok olan mikroişlemcilerin seçilmesi kaçınılmazdır.

Günümüzde ARM mikroişlemcisi kullanan bir cihazın Android veya Linux türevi bir işletim sistemi kullanıyor olması sıkça rastlanmaktadır. Bilgisayarla görü işlemleri için OpenCV kütüphanesi, makina öğrenmesi işlemleri için WEKA kütüphaneleri birçok tasarımcının hayatını kolaylaştırmaktadır. Sadece bir öğrenme algoritmasını çalıştıracak bir sistem için Linux,WEKA kütüphanesini kullanmak çok verimsizdir. Böyle bir sistem en az ARM9 bir çekirdek kullanmak zorundadır ve bu cihazın enerji ihtiyacı diğer ARM serilerine göre çok yüksektir. Bu sebeble geliştirilecek bu ürün enerjinin zor ulaşılabildiği yerlerde kullanılamıyacaktır. Daha birçok alanda sadece kolay yazılım geliştirme amacıyla birçok kaynak verimsiz kullanılmaktadır.

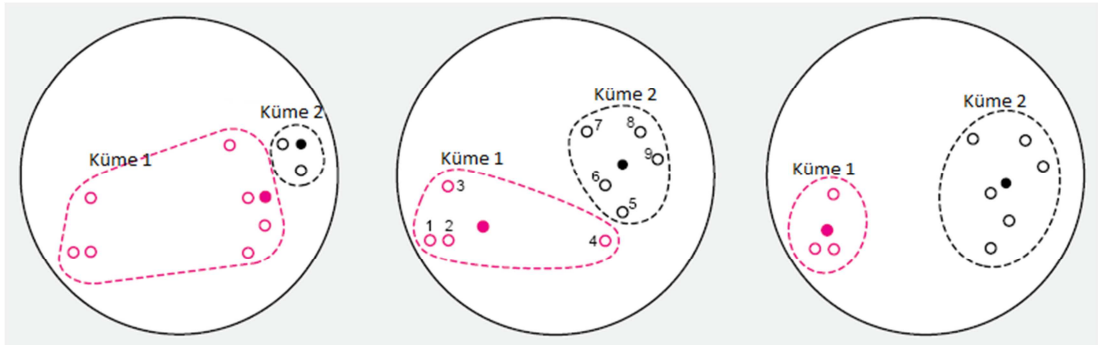
2.3 Otonomi ve Öğrenme

Bir veri seti üzerinde basit veri toplama yöntemleri ile ulaşamayacağımız bilgilere, örüntü tanıma yöntemleri ile ulaşılabilir. Karar verme gibi karmaşık bir işlemde basit veri tabanı sorguları ile ulaşılacak sonuçlar, bu kararın önceden belirlenmiş kalıplara göre yapılmasından öteye geçemez. Bu tez kapsamında kullanmış olduğumuz orman yangınları ile ilgili veri seti başlangıç için düşük boyutlu bir veri kümesi olduğu için sadece veri karşılaştırması yapmak doğru sonuçlar vermez. Kullanılması gereken SQL cümlelerinin önceden tanıtılması zorunluluğu bu cihazın otonom olmasını engeller.

2.3.1 K-means algoritması ve devamlı öğrenme

K-means algoritması bir verinin analizinde ve veri üzerindeki örüntüleri bulmakta kullanılan yaygın bir algoritmadır. Bir kümeleme algoritması olan bu yöntemle, veri setindeki noktalar birbiriyle örtüşen özelliklerdeki benzer veri noktaları ile bir araya toplanırlar. Bu kümeler sadece bir merkez nokta ile tanımlandıkları için, işlem yapılması ve veri depolama ihtiyacı çok düşük seviyede olmaktadır. Vance Faber. [8] Bu algoritmanın koşturulmadan önce kaç adet küme istendiği önceden belirtilmelidir, belirtilen küme adeti kadar küme oluşsada bu kümelerin anlamlı olup olmayacağı algoritmayı koşturan kişinin yorumu ile ortaya çıkacaktır.

Standart k-means algoritması önceden belirlenmiş bir hata değerine kadar çalıştığı için hızlı bir şekilde sonuç üretmektedir. Bu algoritmada her bir veri işleme sonucunda belirtilen sayıda küme merkezi ortaya çıkar. Bu merkezler o noktanın etrafındaki tüm veri noktalarını tanımlarlar.



Şekil 1.2 : K-means algoritması örnek adımları.

Öğrenmenin sürekli olarak ilerlemesini sağlamak için iki öneri sunmuştur. Bu önerilerden ilki her bir adımda, tüm verinin üzerinde çalışmanın gerekli olmadığıdır. Sadece merkez noktaların veya bu merkez noktaya yakın noktaların değerlendirilmesi ile bir noktanın hangi merkez noktaya ait olduğu bulunabilir ve bu işlem sonucunda merkez noktalar güncellenebilir. Bir diğer geliştirme ise alınan bir örnek veri kümesinden başlangıç merkez noktalarının seçilmesi gerekliliğidir. Bu sayede tüm verinin küçük bir parçası verinin tümü gibi davranır ve olası merkez noktalara daha hızlı ulaşılmasını sağlar.

2.3.2 Otonom sistemler

Tam otonom sistemlerin ana hedefi karmaşık optimizasyon problemlerini belirsiz ve neredeyse gerçek bir şekilde, insan yardımı olmaksızın çözmektir. Otonom sözcüğü

hiç bir dış müdahale olmadan çalışabilen, kendi kendine var olabilen sistemlere denilmektedir [9]. Uçaklarda bulunan otopilot uygulamaları gibi uygulamalar önceden tanımlanmış birçok işlemi, yine önceden belirlenmiş olan durumlar için kullanıcı yerine çalıştıran sistemlerdir. Ancak tam otonom sistemler bu işlemlerden daha zor olan önceden belirlenmemiş ortamlarda veya durumlarda da rahatlıkla çalışabilmelidirler.

Bilgisayarla görü, ses tanıma,vb. uygulamalar insanlar için çok basit görevler olsada, bir bilgisayar sistemine bu işlemleri yaptırmak oldukça güçtür. Veri analizi ve örüntü tanıma yöntemleri sayesinde bu konuda birçok ilerleme kaydedilmiştir. Bu görevler genellikle yüksek işlem ihtiyacı olan durumlardır ve bilgisayar sistemlerinin bu görevleri yapması uzun zaman almakta veya çok yüksek bilgi işleme gücü gerektirmektedir. Örüntü tanıma yöntemlerinin kullanılması ile daha az karmaşık şekilde ve daha düşük işlem güçleriyle bu problemler çözülebilmektedir.

Gelecekte insan-makina karar verme sistemlerinin yaygınlaşacağı ve birçok standart algılama işleminin makinalar tarafından yapılacağı, bu sayede insanların daha karmaşık işlemler için kararlar alabileceği düşünülmektedir [10]. Kullanılan yöntemlerin verimliliğin artırılması ve kullanılacakları alanlara uygun olarak tasarımlarının gözden geçirilmesi ile otonom cihazların hayatımızın her alanında yaygınlaşacaktır.

2.4 İşletim Sistemleri

Günümüzde kullanılan işletim sistemleri artık her alanda ortak özellikler barındırmaktadırlar. Bir masaüstü işletim sisteminin sadece kullandığı sürücü yazılımları değiştirilmiş yeni bir sürümünün bir taşınabilir cihazda kullanılması sıkça rastlanmaktadır. Bu sebeple son kullanıcıların kullandıkları işletim sistemleri ölçek olarak kullanılan üründen ürüne çok fazla bir değişiklik göstermemektedirler.

Windows , Linux gibi çok yaygın bir şekilde kullanılmakta olan işletim sistemlerinin daha az özellik barındıran türevleri birçok taşınabilir cihazda kullanılmaktadır. Mikroişlemci çeşitlerinde de görüldüğü üzere bu işletim sistemlerini çalıştıran cihazlar yüksek hesaplama gücüne sahiptirler. Bu işletim sistemlerinin sunduğu avantajlar ve bu işletim sistemleri ile kullanılabilinecek kütüphane çokluğu, işletim sistemi seçiminde bu iki ürün arasında seçim yapmayı sağlamaktadır. ARM9 bir

cihaz üzerinde gömülü bir linux işletim sistemi ile bir OpenCV uygulamasını çalıştırmak zahmetsiz ve daha az hatalı olabilir. Ancak bu işlem sonucunda yapmak istediğimiz görevin kat ve kat fazlası kaynağı boşa harcamış olacağımız açıktır.

Daha düşük güçteki işlemcileri çalıştırabilmek için kullanılabilinecek, düşük boyutlu ve enerji verimli işletim sistemleri, sadece kontrol yazılımları çalıştırmak amaçlı olarak kullanılan bu cihazları daha karmaşık görevler için kullanmayı sağlayabilir. TI-RTOS,FreeRTOS,vb. birçok işletim sistemi hem zamanı hemde kaynaklarını çok yüksek bir seviyede verimli kullanmaktadırlar [11]. Bu işletim sistemleri ile geliştirilen uygulamalar çok az kaynağı sadece cihazı ilklendirme ve cihaz özelliklerini kullanmak için harcarlar. Bu sayede uygulamalar için daha fazla kaynak kalmış olmaktadır. Bunun yanı sıra bu işletim sistemlerin çok küçük boyutlarda cihazlarda da çalışması ile çok kaynak gerektirdiği düşünülen örüntü tanıma gibi uygulamalar birçok alanda kullanılmaya başlanabilecektir.

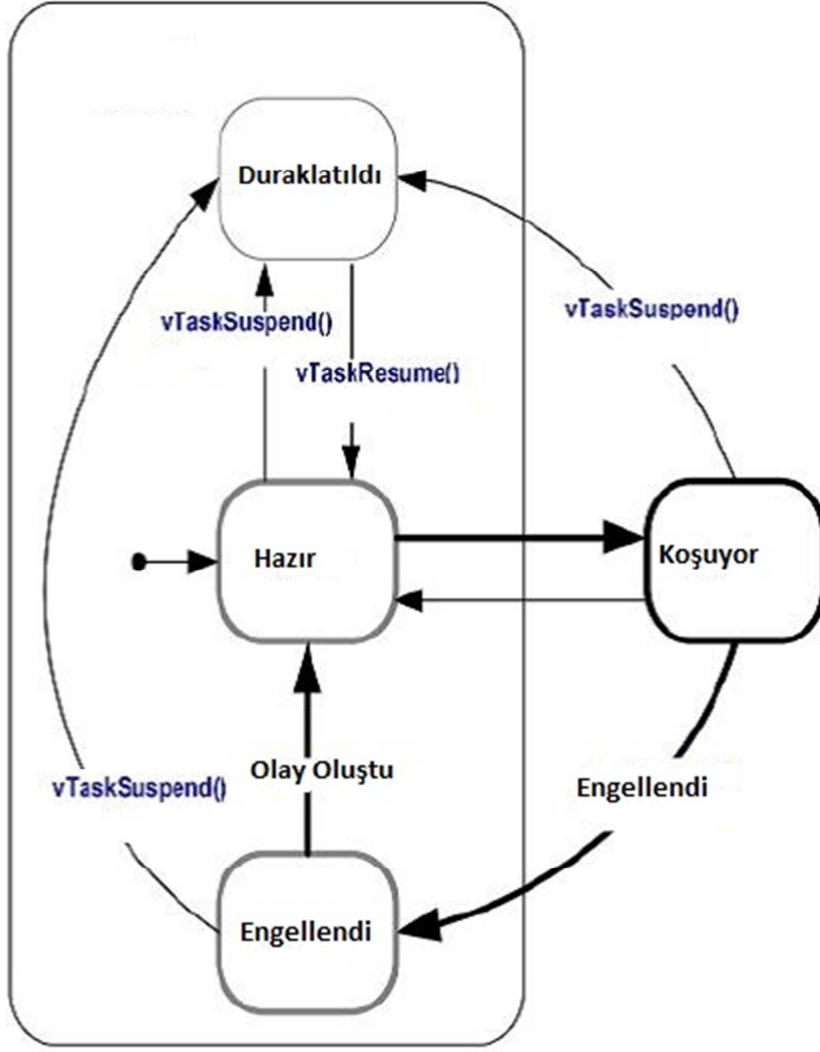
2.4.1 FreeRTOS

FreeRTOS küçük boyutlu gömülü cihazlar için ücretsiz bir gerçek zamanlı işletim sistemidir. Kaynak kodunun büyük bir kısmının C programlama dili ile yazılmış olması, bu işletim sisteminin birçok platforma kolaylıkla uyarlanmasını sağlamıştır.

FreeRTOS kesmeli, işbirlikli ve ikisi bir arada iş dizilim algoritmaları sunmaktadır. Bu işletim sistemi çeşitli seviyelerde uyku modlarını desteklemektedirler. Daha gelişmiş işletim sistemlerinde görmeye alışkın olduğumuz, kuyruklar, semaforlar, muteks yapıları ile işlemler arasında haberleşme sağlanır. FreeRTOS ile sınırsız sayıda işlem oluşturabilir ve kaynakları istediğimiz şekilde bu işlemlere atayabiliriz.

Gerçek zamanlı uygulamalarda görevler birçok işleme atanmış durumdadır. Fakat işlemci bir anda sadece bir işlemlle ilgilenebileceği için, her sistem bu işlemleri hangi sırada çalıştıracığıyla ilgili bir yöntem belirlemelidir. Bu sebeble her işlemi tanımlayan bir veri yapısı ve bu veri yapılarını yöneten bir iş sıralama algoritmasına ihtiyaç vardır.

İşlemciler her an bir işlem yapmak zorundadırlar, bu sebeble her zaman üzerinde işlem yapılabilen bir işlem sistemde tanımlı olmalıdır. Idle Task işlem sıralamacısı tarafından otomatik olarak tanımlanır ve işlemcinin boşta olduğu her an çalıştırılır. Bu işlem en düşük seviyede önceliğe sahiptir ve işlemci zamanını doldurmak dışında kullanılmayan kaynakların serbest bırakılması görevinde üstlenir.



Şekil 1.3 : İşlem akış şeması.

İşletim sisteminin zaman yönetimi Cortex-M4 işlemcinin SysTick özelliği kullanılarak gerçekleştirilmektedir. İşletim sisteminin frekansına göre her döngüde bir kesme oluşturularak işletim sisteminin zamanlaması ayarlanmaktadır. Bu her zaman kesmesinde işletim sistemi iş sıralama algoritması yardımıyla çalıştırılması gereken işlemi belirler ve çalıştırır.

2.5 Motivasyon

Bu tez kapsamında araştırılan işletim sistemleri, düşük kapasiteli işlemciler ve öğrenme algoritmaları aslında birçok görevin bu bilgisayarlar tarafından yapılabileceğini göstermektedir. Bir işletim sistemi çalıştırabileceğimiz mikroişlemci ile uygulamanın genişletilebilir olmasını sağlayabileceğiz. Ayrıca çok düşük enerji

ihtiyacı olan bu mikroişlemciler her türlü enerji ulaşımı zor olan ortamlarda da kullanılabilir.

FreeRTOS işletim sisteminin her türlü yazılım tasarımcısı için geçerli olan işletim sistemi özelliklerini barındırması sayesinde, birçok yazılımın geliştirilmesi kolaylıkla sağlanabilmektedir. K-means algoritmasının bu işletim sistemine bir işlem olarak çalıştırılmasının yanında birçok farklı özelliğinde bu sisteme eklenebilecek olması büyük avantaj sağlayacaktır. Merkez bir istasyon ile iletişimi sağlayabilecek birçok özelliğin kullanılmasına olanak sağlayan Cortex-M4 işlemci çıkarılan platformun farklı alanlarda da kullanılmasına olanak sağlayacaktır.

Düşük boyutlu, kendi kararlarını alabilen tam otonom bir cihaz ortaya çıkarıldığında her alanda rahatlıkla kullanılabilir yüksek verimli işlemler yapabilen geleceğin cihazlarını geliştirebileceğiz.

3. GERÇEKLEŐTİRME

Geliştirilen uygulamanın iki temel parçası herhangi bir alanda kullanılabilir bir "Thing" ve merkez bir istasyondur. Bu iki yazılım parçası birbirleri ile etkileşim kurarak öğrenmenin sürekli olmasını sağlarlar. Ayrıca bu uygulama bir altyapı olarak kullanılabilir. Bu uygulama üzerine yeni işlemler ekleme, var olan işlemleri çıkarma gibi işlemlere rahatlıkla yapılabilmektedir.

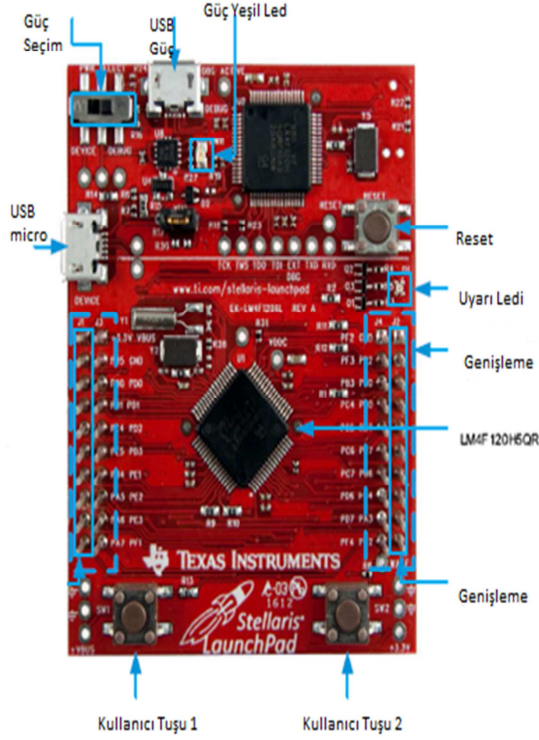
3.1 Geliştirme Kartı ve Çevre Birimleri

Bu projede TI Stellaris LM4F120XL geliştirme kartı kullanılmıştır. Bu cihazın seçimindeki önemli etkenler çok düşük fiyatı(5\$) ve kolay uygulama geliştirmeye olanak vermesidir.

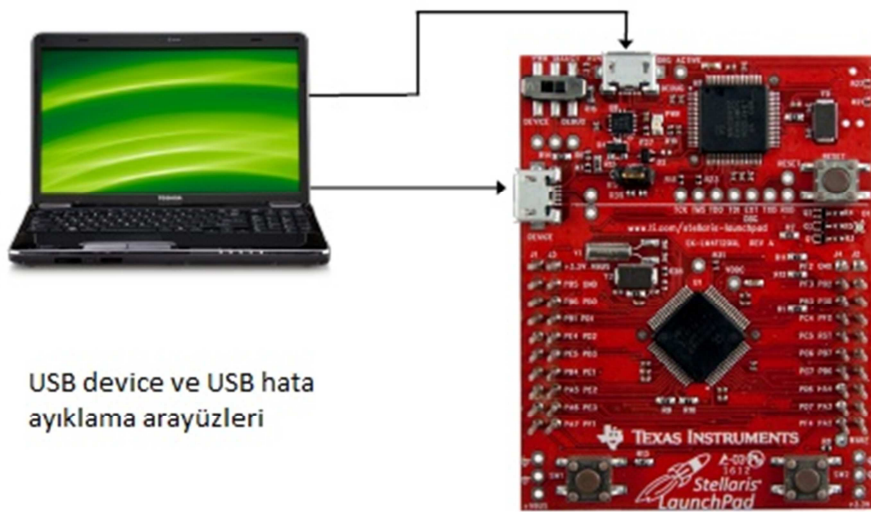
Hata ayıklama arayüzü geliştirme kartı ile gönderilen USB kablosu yardımıyla geliştirme bilgisayarına bağlanarak kullanılmaktadır. Geliştirme ortamı olarak seçilen Code Composer Studio bu kart ile uyumlu bir şekilde kod geliştirme, derleme, kodu hedef karta yükleme ve hata ayıklama işlemlerinin tümünü gerçekleştirmeye olanak sağlar. Özellikle düşük bellek kapasitesi sebebiyle bellek üzerindeki hata ayıklama işlemlerinde CSS sunduğu "disassembly" ve "Memory" görüntüleri çok kolaylık sağlamıştır [12]. Bu geliştirme kartı üzerinde birçok giriş-çıkış birimi yeni birimlerin bağlanması için kart üzerinde kolay ulaşılabilir şekilde sunulmuştur. Bu giriş-çıkış birimleri dışında hazır halde kullanıcıya sunulan 5 adet birim vardır. İki adet kullanıcı butonu kullanıcının belirleyeceği şekilde konfigure edilerek kartın özellikleri kullanılabilir [13]. Bu tez kapsamında bu butonlar ve ledler kullanılarak sürekli öğrenme ve bu bilgiyi iletme görevini üstlenmiş olan "Thing" cihazın, yeni eklenebilecek uygulamalarda tepki verebileceğini göstermek için basit kesme uygulamaları tasarlanmıştır.

Geliştirme Kartı üzerinde bulunan "USB device" girişi sayesinde her türlü USB cihazı bu geliştirme kartına bağlanabilir. USB özelliği bu projede "Thing" ile

Merkez istasyon arasındaki bağlantıyı sağlamak için kullanılmıştır. Bu USB girişi kullanılarak TI Stellaris cihazına Wi-Fi, Zigbee, vb iletişim yöntemleri eklenebilir ve kartın kullanım alanı yaygınlaşabilir.



Şekil 1.4 : Geliştirme kartı.



Şekil 1.5 : Bağlantı şeması.

Çizelge 1.1 : Geliştirme Kartı Arayüzü

G/Ç Bacak No	Bacak Fonksiyonu	Cihaz
F4	ÇD	Buton 1
F0	ÇD	Buton 2
F1	ÇD	Kırmızı Led
F2	ÇD	Mavi Led
F3	ÇD	Yeşil Led

3.2 FreeRTOS Özellikleri

FreeRTOS konfigüre edilebilir özellikleriyle her bir geliştirme kartı için farklı özellikler gösterebilecek şekilde kullanılabilir. FreeRTOS işletim sisteminin LM4F120XL geliştirme kartı üzerinde çalıştırılabilmesi için FreeRTOS işletim sisteminin temel özelliklerini barındıran FreeRTOSConfig.h dosyasının düzenlenmesi gerekmektedir.

Bu Cihaz için kesikli işlem sıralama algoritması seçilmiştir. Sadece belirli bir görevi gerçekleştirmesi beklenen bu cihazın, aynı anda birçok işlemi bir anda yapması ihtiyacı bu işlem sıralama algoritmasının seçilmesini gerektirmiştir.

FreeRTOS işletim sisteminin her bir işlemin çalışmasına karar vereceği zamanlar mikroişlemcinin SysTick özelliği ile sağlanmıştır. Bir kesme olarak çalışan bu özellik belirtilen işlemci vuruş sayısına ulaşıldığında işlemciyi uyarır ve işlemcinin önceden atanmış olan servis yazılımını çalıştırmasını sağlar. Bu uygulama için bu süre 1 ms olarak ayarlanmıştır. 32 KB fiziksel bellek alanına sahip olan geliştirme kartının belleğin büyük çoğunluğu uygulamaların kullanacağı şekilde, büyük bellek bölgesine atanmıştır.

Cihazın kaynaklarının takip edilebilmesi için FreeRTOS işletim sisteminin genel olarak sunduğu özelliklerin bu işlemci kartına uyarlanması gerekmiştir. FreeRTOS işletim sistemine atadığımız tik sıklığını ve işlemcinin üzerindeki zamanlayıcılara bağımlı olarak çalışan bu mekanizma için TIMER2 çevre birimi kullanılmıştır.

3.3 İşlemler ve Görevleri

Bu tez kapsamında gerçekleştirilen örnek sistem 5 izlektan oluşmaktadır. Bu izlekler çok düşük bellek ihtiyacı ve kaynak kullanımı sebebiyle FreeRTOS işletim sistemi üzerinde çalıştırılmıştır. İzleklerin öncelikleri eşit şekilde verilerek, genel amaçlı

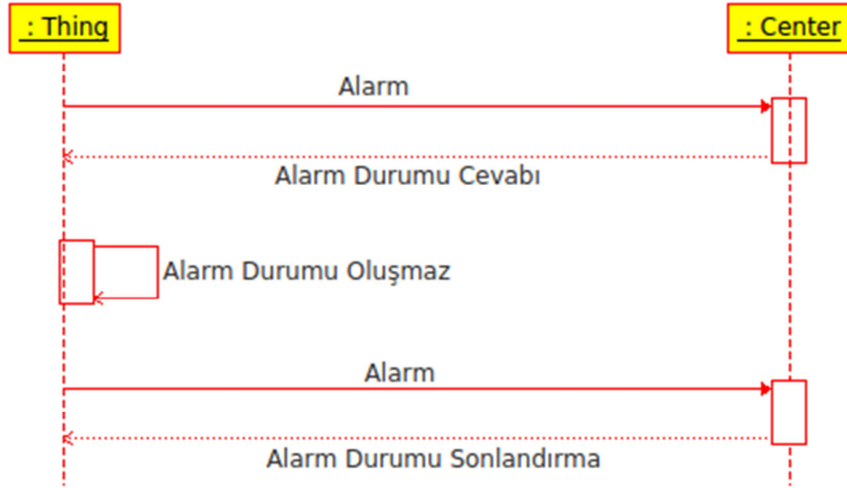
işletim sistemlerindeki(Linux,windows gibi) çalışma prensibine yakın bir sonuç elde edilmiştir.

3.3.1 Öğrenme işlemi

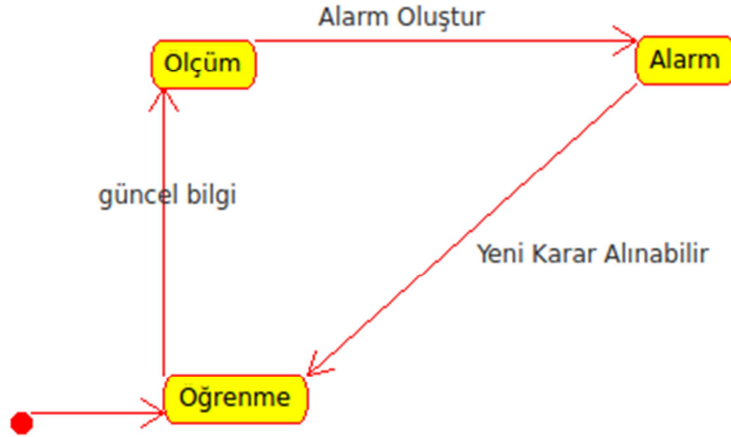
Makina öğrenmesi ve bilgisayarla görü algoritmalarının çok yüksek kapasiteli işlem gücüne ihtiyaç duyduğu için LM4F120XL gibi bir cihazda çalışmasının uygun olmadığı tespit edilmiştir. Hazır bir şekilde bulunabilen kütüphanelerin bellek ihtiyaçlarının yüksek olması ve kullandıkları yöntemlerin kaynakların sınırsız olduğunu kabul etmeye dayanması , bu hazır kütüphanlerin kullanılamamasına neden olmuştur [14]. Bu kütüphanlerin kullanılabilceği işletim sistemlerinin dahi ihtiyaç duyduğu bellek ve işlemci gücü ihtiyacı göz önüne alındığında , seçilen platform için bu algoritmaların gözden geçirilmesi ve verimli bir halde çalıştırılması önem kazanmıştır.

Cihazın çalışmasına örnek oluşturacak bir orman yangını erken uyarı sistemi gerçekleştirilmiştir. Bu sistemin çalışması için herhangi bir dış bilgisayara ihtiyaç duyulmamaktadır. Cihaz tüm kararları kendi verebilmekte ve bu kararı belli aralıklarla merkez bir istasyona iletmektedirler [15]. Akış şu şekildedir;

- Cihaz 60 sn aralıklarla ortamdaki bilgileri (Nem, Sıcaklık, Rüzgar , Mevsim) alıp kendi belleğindeki eski bilgiler ile işler ve sonuçlar oluşturur.
- Eğer bir sonuç yangın oluştuğunu gösterirse bu cihazdan daha önce tanımlanmış olan merkez istasyona alarm bilgisi gönderilir.
- Merkez istasyondaki görevliler bir yangın durumunda yapılması gereken işlemleri gerçekleştirirler.
- Eğer cihaz doğru bir yangın durumunu bildirdiyse bu yeni veri veri setine eklenir ve pozitif olarak işaretlenir. Aksi durumda veri yine eklenir ve negatif olarak öğrenilmiş olur.
- Cihaz bu şekilde bir süre sonra doğru karar oranı artmış olarak çalışmasına devam eder.



Şekil 1.6 : Cihaz alarm oluşturma akışı.



Şekil 1.7 : Cihaz öğrenme durum geçiş diyagramı

3.3.1.1 Veri optimizasyonu

Bu algoritmaların en önemli sorunu kullanılan verilerin büyük boyutlarda olabilmesi ve verilerin tanımlarında verimsiz bellek kullanılmasıdır. Bellek ihtiyacı azaltmak için öncelikle verilerin sembolik bir şekilde ifade edilmesi zorunludur. Yüksek kapasitede belleklerin varlığı sebebiyle veriler insanlar tarafından kolay anlaşılabilir şekilde bırakılmaktadır.

Bu şekilde bir gösterim LM4F120XL gibi bir cihazın bu veri ile çalışmasını olanaksız kılmaktadır. Veriler, daha az bellek ihtiyacı duyacak şekilde yeniden organize edilirse düşük boyutlara sığan veri sayesinde, LM4F120XL sistemimiz kullanılabilir olabilmektedir. Basit görünen bu aşama, aslında yeni tasarımların önünü açan büyük bir engel durumundadır.

Bunun yanında bir verideki her bir özelliğin kullanılmayacağı açıktır. PCA , özellik seçimi gibi yöntemler sayesinde asıl gerekli olan özellikler çok az sayıya indirilebilmektedir. Düşük boyuttaki bu verilerin işlenmesi de kolaylaşmakta, yüksek kapasiteli bilgisayar ihtiyacını azaltmaktadır.

3.3.1.2 Flash bellek üzerinde verilerin tutulması

Yazılan her program ve işletim sistemi parçası bellekte yer kaplamaktadır. Genel amaçlı işletim sistemleri üzerinde (Linux, Windows, vb) çalışan öğrenme algoritmalarının verileri bellek üzerinde tutulmaktadır. Bu yaklaşım LM4F120XL gibi 32KB ana belleğe sahip olan bir cihazın bu algoritmaları çalıştırmasını olanaksız kılmaktadır. Aslında LM4F120XL üzerinde 256 KB flash bellek bölgesinin bu amaçla kullanılması ile bu kısıtı ortadan kaldırmak mümkündür [16]. Algoritmanın çalışma hayatı boyunca değişmeyecek olan bu veriler, sabit bir flash bellek alanında tutulabilirler. Yeni gelen verilerinde flash bellek alanına yazılması ile yeni verilerin eklenmeside sağlanmış olacaktır. Flash bellek okuma-yazma hızının azımsanmayacak kadar yüksek olması, düşük bellek kapasitesiyle dezavantajlı gibi düşünülen LM4F120XL gibi cihazların daha avantajlı olarak kullanılabilmesine olanak sağlar [17].

3.3.1.3 Veri tekrarının kod içerisinde en aza indirilmesi

Veri işlenmesi için tasarlanan algoritmalar, bu verileri işleme sırasında belleği verimli bir şekilde kullanmalıdır. Özellikle verilerin tutulduğu veri yapılarının doğru seçilmesi, düşük kapasiteli bilgisayarlar için büyük önemdedir. Bir veri kümesinin bellekte tutulması için bağlı liste veri yapısının kullanılması, ileri ve geri göstericiler sebebiyle her bir veri için en az 2 işlemci kelime boyu kadar alanın harcanmasına sebep olur. 32 Kbyte boyuttunda bir veri kümesi için oluşturulacak bir kmeans algoritması 32*3 Kbyte bellek bölgesi kullanılması sebebiyle LM4F120XL cihazında çalıştırılmayacaktır. Bunun yanında veri kümesinin parçalara ayrılmasını gerektiren

algoritmelerde(Karar ağaçları gibi) bu küçük veri parçalarının kolay ulaşım için ayrı bir veri yapısında tutulması, yüksek bellek ihtiyacı sebebiyle düşük kapasiteli cihazlarda kullanımı olanaksız kılmaktadır.

3.3.1.4 Bellek kullanımının minimum seviyeye düşürülmesi

Bellek kullanımının verimsiz olmasını engelleyecek bir diğer önemli nokta olan doğru veri için doğru veri tipi seçimi, algoritmaların verimliliğini de arttırmaktadır. Bir "long" veri tipinin yerine en az 8 karakter sığabileceği düşünülürse, dikkatsizce kullanılan birçok verinin aslında kaynaklarımızı ne denli boşa harcadığı ortaya çıkmaktadır. Bir diğer önemli konuda, global olabilecek veri yapılarının doğru belirlenerek her izlekte tekrar tekrar yaratılmasını engellemek, bu sayede izleklerin yığın belleklerini verimli kullanmaktır [18].

3.3.1.5 K-means algoritması

K-means algoritması için küme sayısı iki olarak belirlenmiştir. Bir bölgede yangın çıkma olasılığının olması veya olmaması durumunun göz önünde bulundurulduğu bu uygulamada iki durum için karar verme beklenmektedir. Bu sebeple örnek veri kümesinin sınıf atamaları yapılmış ve sadece iki adet küme oluşturulması için algoritma koşturulmuştur. Bulunan merkez noktalar ile tanımlanan bu kümelere yeni gelen bir elemanın ait olup olmadığı hangi merkeze yakın ise bu kümenin elemanı olması yönünde karar alınması ile sonuçlanmaktadır. Bu karar sonun merkez noktalar tekrar hesaplanır ve güncellenir. Bu şekilde cihazın yangın çıkma tehlikesinin olup olmamasına yönelik kararlar alma ve bu kararlar sonucunda beklenen görevleri yerine getirmesi beklenmektedir.

Orman yangınlarına ait olan bu veri kümesinde 517 adet çeşitli zamanlarda ve hava koşullarında ortaya çıkmış orman yangınları bulunmaktadır. Bu orman yangınlarının nedenleri her bölge için farklılık gösterebileceği için basit sorgulama yöntemleri ile bu sonuca ulaşamaz. Bir insanın karar alma durumu gibi kesinlik olmayan konularda birçok özellik, önceki bilgilerle harmanlanarak sonuç bilgisi oluşturulur.

Sonuçlar o an için cihazın kararını belirtip bu bilginin doğru olup olmadığına karar vermek için başka cihazlar veya insanların etkileşimide gerekli olmaktadır. Alınan karar doğru sonuca ulaşıp ulaşılmadığını alınacak bir doğrulama ile ilerleyebilecektir.

3.3.2 Sistem istatistik işlemi

Sistemin çalışmasını ve gelecekte eklenebilecek uygulamalar için istatistik bilgiler sunan bu işlem ile;

- CPU kullanım oranları ve süreleri
- Yığın Bellek kullanım oranları ve ana bellek kapasitesi
- CPU toplam kullanım bilgisi

takip edilebilmektedir.

3.3.3 USB işlemi

Kullanıcı ile haberleşebilen, dış dünya ile veri alışverişi yapmak için kullanılan bu izlek, usb arayüzü üzerinden LM4F120XL ile bağlantı kurar.

3.3.4 Diğer işlemler

Bu tez kapsamında birçok izleğin düşük kapasiteli bir cihazda çalıştırılması ve aynı zamanda yeni izlekler içinde bir altyapı oluşturması hedeflenmiştir. Yeni bir izleğin hem işlevsel olduğunu hem de tepki sürelerinin doğru bir şekilde kullanıldığını göstermek için 2 adet yeni izlek eklenmiştir. LED izleğinin temel görevi kullanıcı LED ışıklarının rengini değiştirmektir. Kullanıcı butonu 1 ile kontrol edilebilen bu izlek LED rengini kırmızı-yeşil-mavi olarak değiştirir. Frekans değişimini sağlayan diğer izlek ise, LED yanıp sönme frekansını ayarlamaktadır. Kullanıcı butonu 2 ile frekans değiştirilebilmektedir.

3.4 Merkez İstasyon Yazılımı

İnsan öğrenmesinin temel işlemlerinden biriside çevreden ve diğer canlılardan alınan bilgiler doğrultusunda doğru ve yanlış bilgi ile ilgili kendine göre bir kararının olmasıdır. Her türlü ortamda çalışması beklenen bir "Thing" o ortama göre ayrıca özelleşmek zorundadır [19]. Bu tez ile beklenen sonuç bir ortam hakkında en doğru karar verme yetkisinin o ortama konulacak "Thing" olmasıdır.

İnsanların öğrenme şekilleri göz önüne alındığında "Thing" cihazlarında öğrenmek için kararlarının doğru veya yanlış olduğunu söyleyecek bir mekanizmaya ihtiyaç

vardır. Merkez istasyon yazılımı bu ihtiyaç doğrultusunda geliştirilmiştir. Merkez istasyonun çalışma aşamaları şu şekildedir.

- 1) "Thing" ortamında durumu ile ilgili belli aralıklarla kararlar alır.
- 2) "Thing" bu kararları kullanılan iletişim yöntemi ile Merkez istasyona gönderir. Bu proje kapsamında USB haberleşmesi kullanılmıştır. Kullanılacak ortamın özelliklerine göre diğer yöntemlerde seçilebilir.
- 3) Merkez İstasyon yazılımı yangın ile ilgili acil durum işlemlerini gerçekleştirmesi için bir görevliye görsel olarak uyarı verir.
- 4) Bu uyarının gereğini yerine getiren görevli , kararın doğru olup olmamasını göz önüne alarak kararını "Thing" cihazına iletir.
- 5) "Thing" cihazı bu kararı kalıcı belleğinde saklayarak öğrenme işlemlerinde bu verininde kullanılmasını sağlayacaktır.

4. SONUÇLAR

Bu tez çalışması sonucunda genel amaçlı olarak kullanılabilinecek bir altyapı oluşturulmuş ve bu altyapı kullanılarak örnek bir uygulamaya başarı ile çalıştırılmıştır. Düşük enerji tüketimi ve yüksek verim hedefi ile oluşturulan bu altyapı ile yüksek kapasiteli işlemcilerle gerçekleştirilebileceği düşünülen ve bu şekilde geliştirilmekte olan diğer uygulamalara göre avantaj sağlanmıştır.

Çok düşük seviyede enerjiye ihtiyaç duyan bu altyapı ile güneş enerji panelleri yardımıyla uzun yıllar bağımsız olarak çalışabilecek cihazların geliştirilmesi sağlanacaktır. Soğutma ihtiyacının olmaması , pil değiştirme maliyetinin olmaması veya diğer sistemlere göre uzun yıllar gerektirmesi, sessiz olması gibi avantajlar bu altyapıyı gelecekte bir çok cihazda kullanımını cazip hale getirmektedir.

Bu altyapının rakipleri olabilecek sistemlerin fiyat ve enerji maliyetleri tablo 3 te verilmiştir. Görüldüğü üzere hem fiyat hem enerji verimliliği açısından bu altyapı mükemmel sonuçlar vermektedir. Ayrıca gerçek zamanlı bir işletim sistemi kullanan bu altyapıya herhangi bir geliştirici kolaylıkla yeni özellikler ekleyebilecektir. Kullanılan işlemcinin sunduğu kütüphaneler ve ileri çalışmalarda eklenebilecek kütüphaneler yardımıyla, geliştiriciler alt seviye karmaşıklıktan uzak olarak rahatlıkla uygulamalar geliştirebileceklerdir.

Çizelge 1.2 : Platform karşılaştırma çizelgesi.

İşlemci	Enerji İhtiyacı	Fiyat
Raspberry pi B(ARM11)	500 mA	35\$
Arndale Exynos (Cortex A15)	750 mA	259\$
EK-LM4F120XL (Cortex-M4)	100 mA	5\$

Ayrıca gerçek zamanlı bir işletim sistemi kullanan bu altyapıya herhangi bir geliştirici kolaylıkla yeni özellikler ekleyebilecektir. Kullanılan işlemcinin sunduğu kütüphaneler ve ileri çalışmalarda eklenebilecek kütüphaneler yardımıyla,

geliştiriciler alt seviye karmaşıklıktan uzak olarak rahatlıkla uygulamalar geliştirebileceklerdir.

5. GELECEK ÇALIŞMALAR VE ÖNERİLER

Bu tez çalışması ile gerçekleştirilen altyapının iletişim katmanı 6LoWPAN uygulamalarının yaygınlaşmaması sebebiyle eklenememiştir. Hızla ilerlemekte olan bu teknoloji için gerekli iletişim araçlarının bu altyapıya eklenmesi ile çok daha yaygın bir altyapı oluşturulmuş olacaktır.

KAYNAKLAR

- [1] **Patrick Kurp** (2008). Green Computing Communications of the ACM October 2008 Vol. 51 No.10.
- [2] **R. Barry** (2009). Using the FreeRTOS Real Time Kernel - A Practical Guide, 2009.
- [3] **Harnad,Stevan** (2008), "The Annotation Game: On Turing (1950) on Computing, Machinery, and Intelligence",in Epstein, Robert; Peters, Grace, The Turing Test Sourcebook: Philophysical and Methodological Issues in the Quest for the Thinking Computer, Kluwer
- [4] **Luigi Atzori, Antonio Iera, Giacomo Morabito** (2010). The Internet of Things: A survey, Computer Networks 54 (2010) 2787-2805.
- [5] **Vivek Tiwari, Sharad Malik, Andrew Wolfe** (1994). IEEE Symposium on Low Power Electronics 1994.
- [6] **M. Pachter and P.R. Chandler** (1998). Challenges of Autonomous Control IEE Control Systems 1998.
- [7] **Mirela Simonovic, Lazar Saranovac** (2013). Power Management Implementation in FreeRTOS on LM3S3748 Serbian Journal of Electrical Engineering Vol. 10, No. 1, Feburary 2013, 199-208.
- [8] **Mitchell, T.** (1997). Machine Learning, McGraw Hill. ISBN 0-07-042307-7, p.2
- [9] **Mohamed Medhat Gaber, Shonali Krishnaswamy, Arkady Zaslavsky** (2005). Resource-aware Mining of Data Streams Journal of Universal Computer Science, vol 11, no.8 (2005) 1440-1453.
- [10] **ARM.** "Cortex-M4 devices Generic User Guide", ARM Limited, 2010
- [11] **Adirana Wilde, Richard Oliver ve Ed Zaluska** (2013). Developing a Low-cost General-purpose Device for The Internet of Things. In, The 7th International Conference on Sensing Technology (ICST 2013), Wellington, NZ, 03-05 Dec 2013.
- [12] **Rahul Shah, Shonali Krishnaswamy and Mohamed Medhat Gaber** (2005). Resource-Aware Very Fast K-Means for Ubiquitous Data Stream Mining. In: Proceedings of the Second International Workshop on Knowledge Discovery in Data Streams, held in conjunction with ECML PKDD 2005, 3-5 October 2005, Porto, Portugal.
- [13] **Xu Zhen, Tao Yaxiong** (2013). The Design of 3G Embedded Smart Home System Based on ARM Cortex MCU. Applied Mechanics and Materials Vols 373-375 (2013) pp 1618-1621.

- [14] **Lu Tan, Neng Wang** (2010). Future Internet: The Internet of Things. 3rd International Conference on Advanced Computer Theory And Engineering(ICACTE),2010.
- [15] **Vance Faber** (1994). Clustering and the Continous k-Means Algorithm Los Alamos Science Number 22 1994.
- [16] **Texas Instruments**: Stellaris LM4F120XL Evaluation Kit User's Manual, Jan. 2010.
- [17] **Texas Instruments**: Stellaris LM4F120H5QR Microcontroller Data Sheet, Jan. 2011.
- [18] **Yayuan Wen, Wenming Huang, Jie WU, Yue CHEN ve Jiqing SONG** (2013). Water Consumption Analysis System Based on Data Mining. In Applied Mechanics and Materials Vols. 241-244 (2013) pp 1093-1097.
- [19] <<http://www.freertos.org>>, alındığı tarih: 10 Mart 2014.

ÖZGEÇMİŞ

Ad Soyad: Onur Akdemir

Doğum Yeri ve Tarihi: Sivas - 1983

Adres: Ayçil Sokak No:26 D:11 Kadıköy/Istanbul

E-Posta: onurakdemir@gmail.com

Lisans: Ege Üniversitesi

Mesleki Deneyim ve Ödüller: Tübitak Bilgem BTE - Uzman Araştırmacı

TEZDEN TÜRETİLEN YAYINLAR/SUNUMLAR

- **Onur Akdemir**, Turgay Altılar , 2014: International Conference on Computer Science and Engineering (ICCSE 2014), March 17-18, 2014 Chennai, India.