

GÖMÜLÜ SİSTEMLER İLE ENERJİ KALİTESİ

İZLEME SİSTEMİ TASARIMI

Mustafa TOPUZ

Yüksek Lisans Tezi

Elektrik-Elektronik Mühendisliği Anabilim Dalı

Ağustos – 2015

GÖMÜLÜ SİSTEMLER İLE ENERJİ KALİTESİ
İZLEME SİSTEMİ TASARIMI

Mustafa TOPUZ

Dumlupınar Üniversitesi
Lisansüstü Eğitim Öğretim ve Sınav Yönetmeliği Uyarınca
Fen Bilimleri Enstitüsü Elektrik- Elektronik Mühendisliği Anabilim Dalında
YÜKSEK LİSANS TEZİ
Olarak Hazırlanmıştır.

Danışman : Yrd. Doç. Dr. Kadir VARDAR

Ağustos – 2015

KABUL VE ONAY SAYFASI

Mustafa TOPUZ 'un YÜKSEK LİSANS tezi olarak hazırladığı “GÖMÜLÜ SİSTEMLER İLE ENERJİ KALİTESİ İZLEME SİSTEMİ TASARIMI” başlıklı bu çalışma, jürimizce Dumlupınar Üniversitesi Lisansüstü Eğitim Öğretim ve Sınav Yönetmeliğinin ilgili maddeleri uyarınca değerlendirilerek kabul edilmiştir.

28/08/2015

Üye : Doç. Dr. Ahmet ÖZMEN

Üye : Doç. Dr. Bekir MUMYAKMAZ

Üye : Yrd. Doç. Dr. Kadir VARDAR

Fen Bilimleri Enstitüsün Yönetim Kurulu'nun/...../..... gün ve sayılı kararıyla onaylanmıştır.

Prof. Dr. Hasan GÖÇMEZ

Fen Bilimleri Enstitüsü Müdürü

ETİK İLKE VE KURALLARA UYGUNLUK BEYANI

Bu tezin hazırlanmasında Akademik kurallara riayet ettiğimizi, özgün bir çalışma olduğunu ve yapılan tez çalışmasının bilimsel etik ilke ve kurallara uygun olduğunu, çalışma kapsamında teze ait olmayan veriler için kaynak gösterildiğini ve kaynaklar dizininde belirtildiğini, Yüksek Öğretim Kurulu tarafından kullanılmak üzere önerilen ve Dumlupınar Üniversitesi tarafından kullanılan İntihal Programı ile tarandığını ve benzerlik oranının %10 çıktığını beyan ederiz. Aykırı bir durum ortaya çıktığı takdirde tüm hukuki sonuçlara razı olduğumuzu taahhüt ederiz.

Yrd. Doç. Dr. Kadir VARDAR

Mustafa TOPUZ

GÖMÜLÜ SİSTEMLER İLE ENERJİ KALİTESİ İZLEME SİSTEMİ TASARIMI

Mustafa TOPUZ

Elektrik – Elektronik Mühendisliği, Yüksek Lisans Tezi, 2015

Tez Danışmanı: Yrd. Doç. Dr. Kadir VARDAR

ÖZET

Elektrik enerjisi, hayatımızda her alanda var olan vazgeçilmez bir enerji çeşididir. Gelişen teknolojiyle beraber üretilen yüksek akım kapasiteli cihazlar ve güç elektroniği elemanları kullanan cihazlar, akım ve gerilim dalga şekillerinde bozulmalara sebep olmaktadır. Bu bozulmalar, enerji kalitesini kötü yönde etkiler ve elektrik iletim sistemine bağlı olan diğer cihazlara da zarar verir. Bu zararlı etkileri kaydederek incelemek, arızayı meydana getiren sebeplerle ilgili faydalı veriler elde edilmesini sağlar.

Gömülü sistem kullanılarak gerçekleştirilen bu çalışmada, anlık akım ve gerilim değerlerinin ölçümünü yapan, harmonik analizlerini ve reaktif güç hesaplamalarını gerçekleştirebilen, gerektiğinde uzun dönemli raporlandıran, düşük maliyetli bir izleme sistemi tasarlanmıştır. Tasarlanan sistem, büyük miktarda verinin gerçek zamanlı işlenmesine ve depolamasına olanak sağlamaktadır. Güç iletim sisteminden alınan veriler gömülü sisteme aktarılır. Elde edilen verilerin FFT, THD ve güç hesapları yapılarak istenildiği zaman rapor halinde sunulmak üzere kaydedilmektedir. Ölçüm ve hesaplamalar için sensör devreleri, sinyal şartlandırıcılar ve TMS320F28069 dijital sinyal işlemcisi kullanılmıştır. TMS320F28069 ile örneklenip yüksek hızlı seri iletişim ile gömülü sisteme aktarılan bilgiler kullanılarak, Fourier dönüşümü, toplam harmonik dönüşümü, anlık aktif-reaktif-görünür güç hesaplamaları yapılmaktadır. Burada doğru bir FFT ölçümü için akım ve gerilim örnekleri anlık gerilimin temel frekansıyla orantılı olarak değişken örnekleme zamanlı bir şekilde alınmıştır.

Anahtar Kelimeler: Gömülü sistemler, Enerji kalitesi izleme sistemi, Fourier dönüşümü.

THE DESIGN OF POWER QUALITY MONITORING SYSTEM WITH EMBEDDED SYSTEMS

Mustafa TOPUZ

Electrical & Electronic Engineering, M.S. Thesis, 2015

Thesis Supervisor: Asst. Prof .Dr Kadir VARDAR

SUMMARY

Electrical energy is an indispensable form of energy that exists in all areas of our lives. With developing technology; devices manufactured with high current capacity and power electronics lead to deteriorations in the current and voltage waveforms. These distortions adversely affect the quality of the energy and cause damage to other devices that are connected to the transmission system. Examine these harmful effects by recording the fault reason, provides useful data.

This study, carried out by using the embedded systems, measurement of the instantaneous current and voltage values, harmonic analysis, and reactive power calculations detailing the long-term as needed, which can perform a cost-effective monitoring system has been designed. The designed system allows large amounts of data in real-time processing and storage. Data received from the power transmission system are transferred to the embedded system. FFT, THD and power calculations of the obtained data are recorded in a report at any time for report request. For measurement and calculation; sensor circuits, signal conditioners and TMS320F28069 digital signal processor is used. Using information transmitted to embedded systems with high-speed serial communication sampled by TMS320F28069, Fourier transform, total harmonic distortion and instantaneous active-reactive-apparent power calculations are performed. For an accurate FFT measurement, current and voltage samples have been taken in such a way that variable sampling time proportional to the fundamental frequency of instantaneous voltage.

Keywords: Embedded systems, Power quality monitoring Fourier transform.

TEŞEKKÜR

Yaptığım çalışmalar süresince büyük yardım ve desteğini gördüğüm değerli danışman hocam Yrd. Doç. Dr. Kadir VARDAR 'a teşekkürlerimi sunarım. Destekleri, gösterdikleri sabır ve fedakârlık için eşim Tuğba TOPUZ'a, kızım Nisa TOPUZ'a, annem Emine TOPUZ'a ve babam Hüseyin TOPUZ'a çok teşekkür ederim.

İÇİNDEKİLER

	<u>Sayfa</u>
ÖZET	v
SUMMARY	vi
ŞEKİLLER DİZİNİ.....	x
ÇİZELGELER DİZİNİ	xii
SİMGELER VE KISALTMALAR DİZİNİ.....	xiii
1. GİRİŞ	1
2. HARMONİKLE İLGİLİ TEMEL KAVRAMLAR	4
2.1. Harmonik Tanımı	4
2.2. Harmonik Kaynakları	4
2.3. Harmoniklerin Etkileri	5
2.4. Harmoniklerle İlgili Standartlar	7
3. HARMONİK ANALİZİ	8
3.1. Fourier Analizi	8
3.2. Fourier Serileri	8
3.3. Fourier Dönüşümü.....	9
3.4. Ayırık Zamanlı Fourier Dönüşümü	10
3.5. Hızlı Fourier Dönüşümü.....	11
3.6. Radix2 Metodu.....	12
3.7. Bit Tersleme	16
3.8. Pencereleme İşlemi	17
3.9. Toplam Harmonik Bozluma.....	18
3.10.Ölçüm Metodu	19
4. GERÇEKLENEN SİSTEMİ TASARIMI.....	21
4.1. Gömülü Sistemler.....	21
4.1.1.Mini-itx anakart	22
4.1.2.Windows embedded standart	22
4.1.3.Gömülü sistem arayüz yazılımı.....	25

İÇİNDEKİLER (devam)

Sayfa

4.2. Yardımcı Donanımlar.....	31
4.2.1.Akım ölçme kartı	31
4.2.2.Gerilim ölçme kartı	34
4.2.3.Piccolo controlstick.....	37
4.2.4.Sinyal şartlandırıcı kartı	38
4.3. Metod	44
4.3.1.Değişken zamanlı örnekleme	45
4.3.2.Deneysel doğrulama.....	48
4.4. Dosya ve raporlama işlemleri.....	51
5. SONUÇLAR	54
KAYNAKLAR DİZİNİ	55

EKLER

- Ek. 1. Sinyal Şartlandırıcı Kartı Tasarımı
- Ek. 2. FFT ve THD için C# Programlama Dilinde Oluşturulan Sınıf Yapısı
- Ek. 3. Fazlardan Her Biri için Hesaplama Kodları
- Ek. 4. DSP Program Kodları

ŞEKİLLER DİZİNİ

<u>Şekil</u>	<u>Sayfa</u>
2.1. Harmonikli Dalga Şekli.....	4
3.1. Bir sinyalin zaman ve frekans domenindeki gösterimi	10
3.2. Sekiz örnek için W durumları	13
3.3. İki örnek için Radix2 gösterimi.....	15
3.4. Sekiz örnek için Radix2 gösterimi	15
3.5. Hanning pencere gösterimi	17
3.6. Hanning pencere yöntemi kullanılan örnek sinyal.....	18
3.7. Anlık reaktif güç hesaplama.....	20
4.1. Çalışmada kullanılan gömülü sistem kartı	22
4.2. Windows embedded standart 7 ilk kurulum ekranı.....	23
4.3. Windows embedded standart 7 şablondan kurulum ekranı.....	24
4.4. Windows embedded standart 7 seçimli kurulum ekranı	25
4.5. Visual C# geliştirme ortamı	26
4.6. Gömülü sistem ara yüz programı	27
4.7. Gömülü sistem yazılımı veri aktarım algoritması	28
4.8. Veri çevrimi kodları	29
4.9. FFT ve THD için işlem sırası.....	29
4.10. FFT işlem algoritması	30
4.11. Hanning pencereleme kodları	30
4.12. Sinyal şartlandırıcı, akım ve gerilim sensör kartları	31
4.13. Akım ölçme kartı	32
4.14. Akım ölçme kartında kullanılan LA100-P sensörü.....	32
4.15. Akım ölçme kartının iletim sistemine bağlantısı.....	33
4.16. Akım ölçme kartı devre şekli	33
4.17. Akım ölçme kartı baskı devre şekli.....	34
4.18. Gerilim ölçme kartı	35
4.19. Gerilim ölçme kartında kullanılan trafo.....	35
4.20. Gerilim ölçme kartı sinyal çıkışı	35
4.21. Gerilim ölçme kartı devre şeması	35
4.22. Gerilim ölçme kartı baskı devre şekli	36
4.23. F28069 Piccolo Controlstick.....	37

ŞEKİLLER DİZİNİ (devam)

<u>Şekil</u>	<u>Sayfa</u>
4.24. TMS320F28069 DSP	37
4.25. Sinyal şartlandırıcı kart	38
4.26. Sistem genel yapısı blok diyagramı	39
4.27. Sinyal şartlandırıcı gerilim çıkış şekli.....	39
4.28. Tek kanal için sinyal şartlandırıcı devre şekli	40
4.29. Sinyal şartlandırıcı baskı devre şekli.....	40
4.30. Code composer studio kod geliştirme ortamı.....	41
4.31. DSP yazılım algoritması	42
4.32. DSP analog-dijital çevrim diyagramı	43
4.33. Seri port – USB çevrim entegresi.....	43
4.34. Değişik frekanslara göre sabit 50 Hz tabanlı Fourier dönüşüm sonuçları.....	44
4.35. Sogi tipi PLL blok diyagramı.....	46
4.36. Çevrim zamanlayıcısı Simulink blok diyagramı	47
4.37. Simulink freq blok diyagramı iç yapısı.....	47
4.38. Tasarlanan metod için simülasyon sonuçları	48
4.39. Gömülü sistem yazılımı ekran görüntüsü	49
4.40. Üç fazlı yük için kurulan deney düzeneği	50
4.41. Raporlama ekranı algoritması	52
4.42. Raporlama ekranı	52
4.43. Raporlama ekranı	53

ÇİZELGELER DİZİNİ

<u>Çizelge</u>	<u>Sayfa</u>
3.1. Ayrık zamanlı ve hızlı Fourier dönüşümü işlem sayıları	16
3.2. Sekiz eleman için bit tersleme işlemi	16
3.3. IEEE 519 standartına göre gerilim THD sınır değerleri	19
3.4. IEEE 519 standartına göre akım THD sınır değerleri	19
4.1. TMS320F28069 DSP özellikleri.....	38
4.2. Sabit yük altında uygulama sonuçları	49
4.3. Fluke43B ve tasarlanan sistemin ölçüm sonuçları	50
4.4. Fluke43B ve tasarlanan sistemin harmonik genlikleri ölçüm sonuçları.....	51

SİMGELER VE KISALTMALAR DİZİNİ

Simgeler

Açıklama

ω	Açısal Frekans
t	Zaman
A	Akım
V	Gerilim
P	Aktif Güç
Q	Reaktif Güç
S	Görünür Güç

Kisaltmalar

Açıklama

ADC	Analog Digital Converter
ARM	Acorn RISC Machine
ATX	Advanced Technology eXtended
CSV	Comma Separated Values
DFT	Discrete Fourier Transform
DSP	Digital Signal Processor
ePWM	Enhanced Pulse Width Modulation
FFT	Fast Fourier Transform
IEC	International Electrotechnical Commission
IEEE	The Institute of Electrical and Electronics Engineers
I2C	Inter-Integrated Circuit
JTAG	Joint Test Action Group
KByte	KiloByte
Mbit	MegaBit
MSPS	Mega Sample Per Second
RMS	Root Mean Square
SPI	Serial Peripheral Interface
THD	Total Harmonics Distorsion
USB	Universal Serial Bus

1. GİRİŞ

Elektrik, hayatımızın hemen her bölümünde kullandığımız enerji türüdür. Kullanılan enerjinin, bu enerjiyi kullanan cihazları bozmayacak sınırlar içerisinde değerlere sahip olması enerjinin kalitesi ile ilgilidir. Kaliteli elektrik enerjisi; genlik ve frekans değerleri daha önceden belirlenmiş sınırlar içinde ve mükemmel sinüsoidal işarete sahiptir. Enerji kalitesi problemi, enerjiyi kullanan cihazda bozulma veya hatalı çalışmaya yol açan; akım, gerilim veya frekansta görülen değişimlerdir (Dugan, vd., 1996).

Enerji kalitesinde oluşan bozulmalar, son kullanıcı cihazlarında bozulmalara veya cihazların hatalı çalışmasına sebep olabilmektedir. Özellikle sanayi ortamında yaşanacak anlık arızalar üretimi süregelen işlemlerde büyük hasarlar oluşturabilir. Kısa süreli kesintiler, ardışık işlemler yapan makinelerde hammaddelerin hatalı işlenerek ürüne dönüşmeden kaybına sebep olabilir. Kalitesiz enerji sebebiyle yarı iletken sanayisi gibi çok hassas işlemler yapılan endüstri kollarında ciddi kayıplar görülmektedir (Chapman, 2001).

Güç elektroniğindeki gelişmeler ve yüksek güçlü cihazların üretilmesiyle beraber sanayi sektöründe elektronik kontrollü yüksek güçlü cihazların kullanımı da yaygınlaşmaktadır. Bu cihazlar, faydalı özelliklerinin yanı sıra elektrik iletim sisteminde de çeşitli bozulmalara sebep olmaktadır. Bu bozulmalardan önemli etkilere sahip olan bir bozulma türü de harmoniklerdir.

Gelişen yarı iletken teknolojisiyle beraber gömülü sistemler olarak isimlendirilen yapılar ortaya çıkmış ve bu sistemlerin kullanımı yaygınlaşmıştır. Gömülü sistemler sadece bir işlevi veya sınırlı sayıda işlevi yerine getirebilecek şekilde tasarlanmış, son kullanıcının sadece veri girişi yaparak kullanabileceği mikroişlemci tabanlı sistemlerdir (Heath, 2003). Sadece belirli işlevleri getirecek şekilde tasarlandıklarından hem maliyetleri daha düşüktür ve hem de bilgisayar tabanlı sistemlere göre çok daha hızlı ve kararlı çalışabilirler.

Gömülü sistemler kullanılarak enerji iletim sistemindeki problemler izlenebilir, kayıt tutulabilir ve tutulan kayıtlar incelenerek problemleri oluşturan sebepler araştırılabilir. Literatürde gömülü sistemlerle gerçekleştirilen, çeşitli yöntemler kullanan enerji kalitesi izleme sistemi çalışmaları bulunmaktadır. Bu çalışmalardan biri, 8051 tabanlı işlemcisiyle harmonik gözlemlemesi yaparak LabView ve Matlab programında işlem yapılmasını sağlamaktadır. Elde edilen sonuçlar internet üzerinden görülebilmektedir (Lin, 2005). ARM tabanlı işlemciye sahip ve C++ dili ile geliştirilen gömülü sistem enerji kalitesi gözlemleme cihazı literatürde rastlanan çalışmalardan bir diğeridir. Enerji iletim sisteminden elde edilen verileri işlemek için FFT ve wavelet dönüşümleri kullanmıştır. Cihaz içerisinde işletim sistemi olarak gömülü Linux

bulunmaktadır (Lu, vd., 2008). Dijital sinyal işleyici tabanlı gerçekleştirilen harmonik gibi enerji kalitesi sorunlarını gözlemleyen bir diğer sistem elde ettiği verilerin FFT dönüşümünü almakta, üzerinde bulunan göstergede göstererek diğer sistemlerde kullanılmak üzere aktarılmasını sağlamaktadır. Cihazda kullanılan işletim sistemi, sistemin dijital sinyal işleyicisinin (DSP) komut kodlarıyla yazılmıştır (Huang, vd., 2011). Gömülü sistem kullanılarak gerçekleştirilen başka bir çalışma Wavelet dönüşümü, yapay sinir sistemleri kullanan harmonik belirleme ve etkilerini azaltma amaçlı tasarlanan sistemidir. Bu sistem, işletim sistemi olarak Linux kullanır. Yazılan kodlar QT yazılım kütüphaneleri kullanılarak hazırlanmıştır. Elde edilen verilerin başka sistemlerde kullanılması amaçlı aktarılacak üzere seri port ve USB bağlantı yöntemlerini kullanır (Li, vd., 2010). Literatürdeki önemli çalışmalardan biri de ülkemizde uygulanmakta olan Güç Kalitesi Milli Projesidir. Proje üzerinde ilk olarak 2006 yılında çalışılmaya başlanmış ve 2009 yılında faaliyete geçirilmiştir. Proje kapsamında güç kalitesi monitörleri tasarlanmış ve sistemde kullanılmıştır. Güç kalitesi monitörü; iletim hattındaki akım, gerilim, frekans, aktarılan güç değerleri, harmonikler, gerilim düşüşleri ve gerilim kesintileri gibi bilgileri gerçek zamanlı olarak izleyip kaydetmektedir. Güç kalitesi monitörlerinin her birinde veri yakalama amaçlı veri akış kartı, mini-itx kart, gps ve adsl modem gibi yardımcı birimler de bulunmaktadır (Gök, vd., 2011).

Gerçekleştirilen bu tez çalışmasında istendiğinde problemlerin gözlemlenip teşhis edilmesinde kullanılabilecek ayrıntılı olarak tutulan olay raporlarının incelenebileceği basit ve düşük maliyetli, gerçek zamanlı çalışabilecek bir sistem tasarımı hedeflenmiştir. Tasarlanan enerji kalitesi izleme sistemi, büyük miktarda verinin gerçek zamanlı işlenmesine ve depolamasına olanak sağlamaktadır. Temel donanım olarak küçük boyutları, düşük fiyat ve yüksek performans özellikleri nedeni ile mini-itx pc ana kartı tercih edilmiştir. Enerji kalitesi izleme yazılımı C# programlama dili kullanılarak gerçek zamanlı çalışabilecek şekilde hazırlanmıştır. Yazılım için tasarlanan sistem, modüler sınıf yapısında kodlandığından istendiğinde çok hızlı bir geliştirme süreciyle ihtiyaç duyulan işlevler de eklenebilir ve çok kolay bir şekilde farklı işletim sistemleriyle de uyumu sağlanabilmektedir. Tasarlanan sistem, .NET sistemi olan her cihazda çalışabilmektedir. Linux ve Linux türevi dağıtımları, Android gibi sistemlerde de .NET kullanımı için oluşturulan yazılım kütüphaneleri yüklenerek platform bağımsız olarak çalışabilir. Yazılımın gerçek zamanlı çalıştırılabilmesi gereksiniminden dolayı Windows 7 Embedded gömülü işletim sistemi kullanılmıştır.

Güç kalitesi problemlerini belirlemek için şebekeden akım ve gerilim değerlerini ölçecek akım ve gerilim sensör kartları tasarlanmıştır. Sensörlerden gelen bilgiler bir sinyal şartlandırıcı devre ile işlenmeye uygun hale getirilerek dijital sinyal işleyicinin analog dijital

çeviricisine aktarılmaktadır. Sinyal şartlandırıcı kartta dijital sinyal işleyici (DSP) kullanılarak verilerin gerçek zamanlı, kayıpsız ve yüksek örnekleme miktarına sahip olacak şekilde örneklenmesi hedeflenmiştir. Gömülü sisteme aktarılan bilgiler, Fourier dönüşümü, toplam harmonik dönüşümü, anlık aktif-reaktif-görünür güç hesapları yapılarak her dakika için ortalamaları zaman bilgisiyle birlikte kaydedilmiştir. Fourier sonuçlarının doğruluğunu artırmak için akım ve gerilim örnekleri, anlık gerilimin temel frekansı ile orantılı olarak yapılan değişken örnekleme ile alınmıştır. Bu işlem sonucu şebeke frekansında değişimler olduğunda FFT sonuçları değişimlerden etkilenmeden değerlendirilecektir. Temel frekans belirlenirken yazılımsal PLL algoritması kullanılmıştır. Elde edilen verilerin sisteme kayıt edilmesi için herhangi bir kota konmamış; sistem, kayıt yapılan disk dolana kadar kayıt edebilmesi için ayarlanmıştır. Bu sayede çok miktarda veri kaydı yapılabilir. Bu sayede çok miktarda veri kaydı yapılabilir.

Daha önce yapılan çalışmalardan farklı olarak bu çalışmada gömülü işletim sistemi haricindeki diğer tüm geliştirmelerde lisans gerektirmeyen yazılımlar kullanılmıştır. Bu şekilde tasarım maliyetleri düşürülmüş, lisanslama gereksinimi olmadığından daha da ucuza mal edilmiştir.

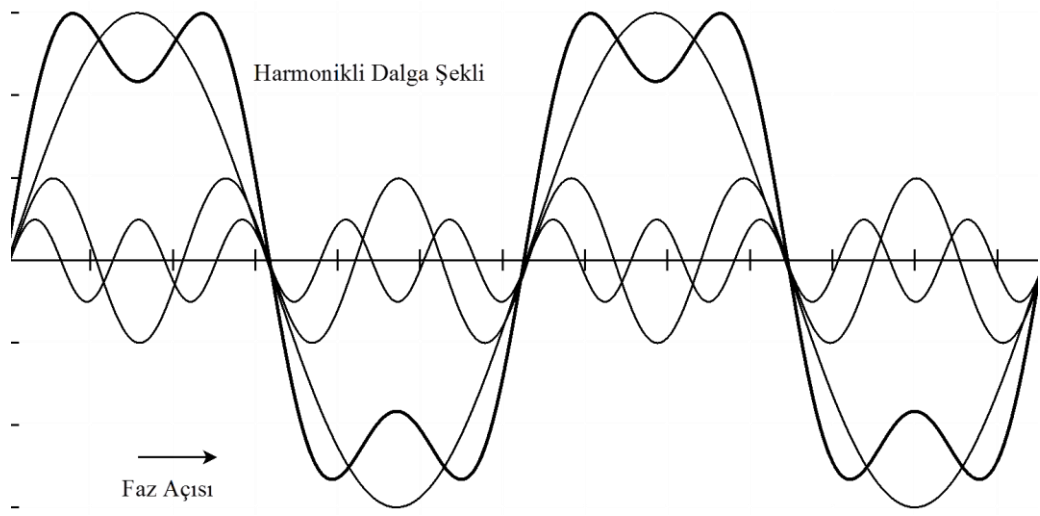
DSP içerisinde çalışması için hazırlanan kod sadece verileri sensör devrelerinden örnekleyecek ve bu değerleri gömülü sisteme aktaracak şekilde tasarlanmıştır. Gömülü sistem yazılımı oldukça esnek ve çok kısa sürede değiştirilerek farklı yöntemler eklenebilir hale getirilmiştir. Gömülü donanım kısmı standartlaşmış bir donanım olarak seçildiğinden hem oldukça ucuz ve hem de arıza yapması durumunda piyasadan herhangi bir yerden elde edilip hemen faaliyete alınacak kadar yaygın kullanımdadır.

Tez çalışmasının ilk bölümünde, temel bilgiler verilmekte ve yapılan çalışma kısaca özetlenmektedir. İkinci bölümde, enerji kalitesi problemlerinden biri olan harmoniklerin yapısı, oluşum sebepleri ve etkileri anlatılmaktadır. Üçüncü bölümde, çalışmada kullanılan yöntemler ayrıntılı olarak incelenmektedir. Dördüncü bölümde, sistemde kullanılan donanım ve yazılım bileşenleri incelenmekte ve yapılan çalışma detaylandırılarak anlatılmaktadır. Sonuç kısmında yapılan çalışmayla ilgili ulaşılan sonuçlar değerlendirilmektedir.

2. HARMONİKLE İLGİLİ TEMEL KAVRAMLAR

2.1. Harmonik Tanımı

Harmonikler; akım veya gerilimde görülen, temel enerji frekansının tamsayı katları toplamı şeklinde oluşan bozulmalardır (Chapman, 2001a; Singh, 2009). Temel enerji frekansının katları olarak görülen akım veya gerilimler, temel akım veya gerilimle birleşerek sinüsoidal dalga şeklinin bozulmasına sebep olurlar (IEEE519:1992, 1992).



Şekil 2.1. Harmonikli Dalga Şekli.

Harmonikler gerilim dalga şeklinde görülebildiği gibi akım dalga şeklinde de görülebilir. Gerilim şeklinde bozulmalar olarak gerçekleşen harmonikler gerilim harmonikleri ve akım şeklinde bozulmalar olarak gerçekleşen harmonikler de akım harmonikleri olarak adlandırılır. Harmoniklerden bahsederken daha çok akım harmonikleri kastedilir çünkü harmoniklerin meydana getirdiği arızaların önemli bir kısmına akım harmonikleri sebep olmaktadır (Chapman, 2001a).

2.2. Harmonik Kaynakları

Harmonikler, genellikle şebekeye bağlı lineer olmayan yükler tarafından oluşturulmaktadır. Bu yükler şebekeden kısa sürelerde yüksek akımlar çekerek bağlı oldukları güç kaynağı gerilimi ve akımında sinüzoidal olmayan değişikliklere yol açarlar (Singh ve Singh, 2010). Akımda oluşan sinüzoidal olmayan bu değişiklikler, gerilimde de bozulmalara yol açmaktadır. Sonuç olarak, şebekede ortaya çıkan harmonikler şebeke üzerinden başka yükleri de etkiler (Bayliss ve Hardy, 2012).

Genel olarak harmonik kaynakları 3 grupta toplanabilir (Wakileh, 2011):

- Kesintisiz güç kaynakları, motor hız sürücüler, doğrultucular, anahtarlama mod güç kaynakları, tristör kontrollü güç elektroniği ekipmanları,
- Floresan lambalar, civa buharlı lambalar ve ark fırınları,
- Transformatör, motor, jeneratör gibi manyetik doyuma sahip cihazlar.

Anahtarlama mod güç kaynakları çalışma şekillerinden dolayı akım harmonikleri oluştururlar. Oluşan bu harmonikler gerilim şeklinde bozulmaya yol açarak bağlı bulundukları şebekeye dağılırlar, başka cihazları da etkilerler (Mok, vd., 2012).

Sanayide sıklıkla kullanılan motor hız sürücüler de harmonik üreten cihazlardır. Bunun başlıca sebebi çekilen akım şeklinin sinüzoidal olmamasından dolayıdır. Motor hız sürücülerinin kullanımı sonucu oluşan harmonikler şebekeye karışır (Jamil, vd., 2007).

Güç elektroniği elemanlarından olan diyotlar ve tristörler çalışma şekilleri sebebiyle harmonik üretir (Bayliss ve Hardy, 2012). Güç elektroniği cihazlarının yüksek frekansla anahtarlama sırasında harmonikler ve elektromanyetik girişim oluşur. Bu etkiler dağıtım hattıyla etkileşime girerek sistemin kararlılığını ve güvenilirliğini etkiler (Jianqu, vd., 2012) .

Günümüzde kullanılan tasarruflu ampul ve led lambaların yapısında bulunan anahtarlama mod güç kaynakları da bir diğer harmonik sebebidir. Tasarruflu ampuller ve ledlerin çalışmaları için gereksinim duydukları elektronik devreler de şebekeye harmonik enjekte etmektedirler (Blanco ve Stiegler, 2013). Küçük akımlar çekmelerine karşın bu cihazlar tarafından oluşturulan gerilim ve akım harmonikleri transformatörlerde ısıya, daha yüksek kayıplara ve transformatör ömrünün azalmasında sebep olur (Shareghi, vd., 2012).

Sanayide kullanılan ark fırınları, metali daha hızlı ve daha ekonomik olarak eritmek için kullanılan ekipmanlardır. Değişken zamanlarda yüksek reaktif güç harcarlar. Çekilen ani akımlar ve yüksek reaktif güç sebebiyle harmonik oluştururlar. Oluşan harmonikler enerji iletim şebekesine dağılır (Zhao, vd., 2010).

Trafolarda ve motorlar gibi dönen manyetik alanlı cihazlarda oluşan manyetik doyumlar harmonik oluşturabilir. Günümüzde gelişen teknoloji ile üretilen katkılı çelik metaller ile bu cihazlar tarafından üretilen harmonikler azaltılmıştır (Bayliss ve Hardy, 2012).

2.3. Harmoniklerin Etkileri

Harmonikler, oluştukları kaynaktan bağlı bulundukları elektrik şebekesine dağılarak şebekeye bağlı başka cihazları da etkileyebilirler (Mok, vd., 2012).

Harmonikler normal gerilim ve akım şeklinden daha yüksek değerlere çıkabileceği için şebekeye bağlı dağıtım hattında gerilim düşümleri ve değişken empedanslar görülebilir. Harmoniklerin iletim sistemindeki bir diğer etkisi de güç katsayısını olumsuz yönde etkilemesidir (Jamil, vd., 2007).

Tek fazlı sistemlerde oluşan harmonikler nötr hattındaki RMS değeri artırarak hattın daha fazla yüklenmesine sebep olur. Nötr hattı normalde olması gerekenden daha fazla yüklenebilir. Bu durum, nötr hattında ısınmaya ve nötr hattı yalıtkanlarının zarar görmesine sebep olabilir. Tek fazlı sistemlerde üçüncü harmonik daha çok görülür. Bu etkilerden dolayı nötr hattı daha kalın bir iletken kullanılır. Ayrıca, harmonikler üç fazlı sistemlerde de nötr hattından akan akımın artmasına sebep olmaktadır (Rosa, 2006).

Motorlar ve dönen makineler üzerinde bakır kayıpları, demir kayıpları, aşırı ısınmaya sebep olur. Aşırı ısınma sonucu makinada iletkenler üzerinde bulunan yalıtkan da zarar görür. İletkenlerdeki ısınma akım taşıma kapasitesini de etkiler (Singh, 2009 ; Wakileh, 2003).

Harmonikler, transformatörler üzerinde demir ve bakır kayıpları oluşturur. Sargılar üzerindeki yalıtkanın zarar görmesine sebep olur. Aşırı ısı ve iletkenler üzerindeki yalıtkanların tahrip olması sebebiyle transformatörlerin ömürlerini kısaltır (Chapman, 2001b).

Harmonikler, sigortalar üzerinde de kötü bir etkiye sahiptir. Harmonikler gerilim ve akım RMS değerlerinde artışa sebep olur. Eğer harmoniklerin eklenmesiyle birlikte oluşan yeni akım değeri, hesaplanan sigorta değerini aşarsa sigortalar atar (Chapman, 2001b).

Güç katsayısını düzeltmek için kullanılan kompanzasyon sistemindeki kondansatör grupları da harmonikler sebebiyle zarar görebilir. Sistemde, harmonikli gerilim değerleri normal gerilim değerlerinin üzerindedir. Bu fazla gerilim, kondansatörlerin ısınmasına ve ömürlerinin kısılmasına sebep olur (Rosa, 2006). Kondansatör gruplarında görülen bu bozulmalar, çekilen reaktif gücün artmasına ve buna bağlı olarak elektrik faturalarında anormal artışlara sebep olabilir. Bazı durumlarda kondansatör gruplarıyla harmonikler rezonansa girerek daha büyük gerilimlerin oluşmasına ve sistem üzerinde daha da büyük yıkıcı etkilere yol açarlar (Chapman, 2001b).

Harmonikler, akım ve gerilimdeki düzgün sinüsoidal şekli bozarak sıfır noktası geçişlerini bularak çalışan ölçüm cihazlarının ve kontrol cihazlarının hatalı çalışmasına sebep olur. Ayrıca ölçüm cihazlarına hatalı sonuçlar üretmesine yol açar (Arrillaga ve Watson, 2003).

Alternatif akım iletken iletkenlerin dış yüzeyinde akar. Buna deri etkisi (skin effect) denir. Deri etkisi düşük frekanslarda çok önemli değerlerde olmadığından göz ardı edilir. Ancak 350 Hz ve daha yüksek frekanslarda kayıp ve ısınmalara sebep olur (Chapman, 2001b).

Harmonikler, haberleşme hatlarında elektromanyetik girişime ve toprak hattından akımlara sebep olabilir (Rosa, 2006).

2.4. Harmoniklerle İlgili Standartlar

Uluslararası kuruluşlar harmonikler hakkında standartlar oluşturmuştur. Bu kuruluşlardan IEC, elektromanyetik uyum başlığı altında standartlar yayınlamıştır. IEC 61000 serisi standartları uluslararası kabul edilmiş güç sistemi harmonikleriyle ilgilidir. Bu standartlardan bazıları; IEC 61000 1-4, IEC 61000 2-1, IEC 61000 2-2, IEC 61000 2-4, IEC 61000 2-12, IEC 61000 3-2, IEC 61000 3-4, IEC 61000 3-6, IEC 61000 3-12, IEC 61000 4-7, IEC 61000 4-13 olarak sayılabilir (Arrillaga ve Watson, 2003).

IEEE kurumu da harmonikler hakkında standart yayınlamıştır. IEEE519 numaralı standart harmoniklerin ana kaynaklarını, etkilerini, kabul edilebilecek harmonik seviyelerini, harmonik ölçümü ile ilgili işlemleri ve harmoniklerin etkilerini azaltmak için gereken bazı yöntemleri tarif eder (IEEE519,1992).

3. HARMONİK ANALİZİ

Güç sistemindeki harmonikler tespit edildiğinde, işletmelerde harmoniklere sebep olan cihazlar ve bu cihazların ürettiği harmoniklere karşı önlem almak kolaylaşabilir. Bir sinyali, temel frekansın katlarında sinüzoidal dalgalarla ifade etmek için Fourier serilerinden faydalanmak mümkündür (Chang, 2008). Bunun için iletim hattındaki akım ve gerilimin Fourier analizini yapmak yeterlidir.

Bu bölümde, Fourier analizinin nasıl yapıldığı ve hızlı Fourier dönüşümü konuları anlatılacaktır.

3.1. Fourier Analizi

Periyodik özellik taşıyan herhangi bir sinyal, çeşitli genlik ve çeşitli frekanslardaki sinüs ve kosinüs sinyallerinin toplamı olarak ifade edilebilir. Bunun için Fourier dönüşümü kullanılabilir (Chang, 2008; Chang, vd., 2010). Bir sinyalin Fourier dönüşümü alındığında sinyali oluşturan frekanslar, frekans genlikleri ve açıları bulunabilir.

Bu özellik kullanılarak, periyodik özellik taşıyan ve sinüzoidal olan elektrik şebekesi sinyalleri de incelenebilir. Güç sistemlerinde akım ve gerilim işaretleri incelenerek sinyallerin frekans içeriği belirlenebilir.

Güç sisteminde görülen harmonikler temel frekans olan 50Hz katları şeklinde olacaktır. Şebekeden alınacak akım ve gerilim üzerinde yapılacak Fourier analizi ile 50Hz ve katları olarak görülen sinyal genlikleri ve faz açıları bulunabilir. Bununla birlikte elde edilen sinyal içeriği, genliği ve açılarıyla ilgili araştırmalar yapılarak bunlara şebekeye bağlı olan elemanlardan hangilerinin sebep olduğu, bunlara karşı ne tür önlemler alınacağı kestirilebilir.

3.2. Fourier Serileri

Fourier serileri Fourier dönüşümünün temelini oluşturur. Fourier serileri ile herhangi bir periyodik sinyal, çeşitli frekanslardaki ve genliklerdeki sinüs ve kosinüs fonksiyonları toplamı şeklinde ifade edilebilir (Chaparro, 2015).

$$f(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} (a_n \cdot \cos n\omega t + b_n \cdot \sin n\omega t) \quad \omega = \frac{2\pi}{T} : \text{Temel frekans} \quad (1)$$

Akım ve gerilim şekilleri de Fourier serisine göre sinüs ve kosinüslerin toplamı şeklinde ifade edilebilir. Akım ve gerilim dalga şekilleri için Fourier serilerinin katsayıları bulunduğunda

harmonik analizi de yapılmış olacaktır (Katznelson, 2004). Fourier serisi katsayılarını bulmak için akım ve gerilim değerlerinin Fourier dönüşümü gerçekleştirilir.

3.3. Fourier Dönüşümü

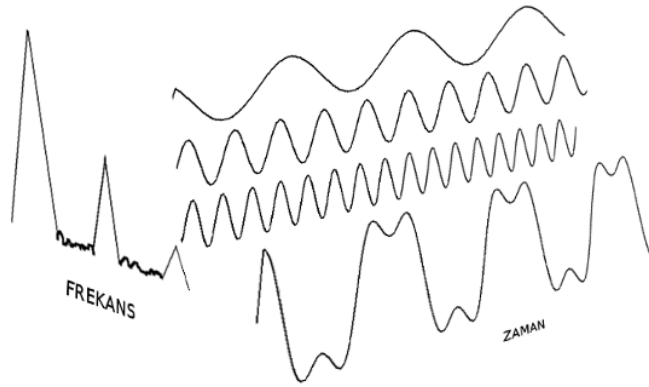
Fourier dönüşümü, zaman domenindeki sinyallerin frekans domeninde ifade edilerek incelenmesini sağlar. Zaman domenindeki veriler Fourier dönüşümüyle frekans domenine; frekans domenindeki veriler de ters Fourier dönüşümüyle zaman domenine aktarılabilir. Fourier dönüşümü sonucunda Fourier serisi katsayılarını elde ederiz. Fourier serisindeki veriler karmaşık sayılardır. Bu veriler kullanılarak sinyallerin genliği, frekansı ve faz farkı elde edilebilir. Fourier dönüşümünden elde edilen sonuçlar kullanılarak ters dönüşüm yapılabilir. Fourier dönüşümü sonucu elde edilen katsayılar kullanılarak orijinal sinyal tekrar elde edilebilir (Katznelson, 2004).

Fourier dönüşümü sonucu elde edilen veriler, dönüşümü yapılan verilerden oldukça azdır. Dolayısıyla bu teknik, sıkıştırma için kullanılabilir. Günümüzde en çok kullanılan müzik dosyası formatı olan mp3, ham ses dosyası olan wav formatının Fourier dönüşümüyle sıkıştırılmış biçimidir. Akıllı telefonlarda kullanılan ses ve müzik tanıma tekniği uygulamalarında Fourier dönüşümü kullanılır. Yine günümüzde sıkça kullanılan sıkıştırılmış dosya formatı olan jpg veya jpeg dosya formatı, ham resim verisi olan bmp dosya formatının Fourier dönüşümünden faydalanarak sıkıştırılmış halidir. Dijital sinyal işlemede de Fourier dönüşümü kullanılır. Bir sinyalin sıkıştırılması, yeniden oluşturulması, frekans içeriğinin belirlenmesi, frekans içeriğinde istenmeyen frekansların bulunarak filtrelmesi gibi işlemlerde Fourier dönüşümü kullanılır (Park, 2010; Smith, 1998; Tan, 2008).

Fourier dönüşümü şu formülle ifade edilir:

$$F(\omega) = \int_{-\infty}^{+\infty} f(t) \cdot e^{-j\omega t} dt \quad (2)$$

Formülde $F(\omega)$ Fourier dönüşüm sonuçlarını içeren fonksiyonu, $f(t)$ ise dönüşümü yapılacak olan fonksiyonu ifade eder. Fourier dönüşümü sonrası, sinyalin içerdiği frekans bileşen bilgileri elde edilir.



Şekil 3.1. Bir sinyalin zaman ve frekans domenindeki gösterimi.

Fourier dönüşümü yapılmış olan bilgilerden sinyalin kendisi tekrar elde edilebilir. Bunun için ters Fourier dönüşümü kullanılır. Ters Fourier formülü şu şekilde tanımlanır:

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} F(\omega) \cdot e^{j\omega t} d\omega \quad (3)$$

3.4. Ayırık Zamanlı Fourier Dönüşümü

Fourier dönüşümü formülüne göre dönüşüm sonsuz uzunlukta bir dizi için tanımlıdır. Bilgisayarlar, gömülü sistemler, dijital sinyal işlemciler gibi elektronik elemanlarda bu işlemi gerçekleştirmek için sinyalin belirli sayıda veri için sonsuz tanımlı olan dizinin sınırlandırılması ve sınırlı sayıda veri ile dönüşümün gerçekleştirilmesi gerekir. Bunun için Fourier dönüşümünün ayırık zamanlı şekli olan ayırık zamanlı Fourier dönüşümü kullanılır.

Ayrık zamanlı Fourier dönüşümü, dijital sinyal işleme için uygundur. Analizi yapılacak sinyal, önce analog-dijital çevirici devrelerle dijital sinyal haline çevrilir. Daha sonra ayırık zamanlı Fourier dönüşümü formülüne uygun olarak dönüşüm gerçekleşir ve dönüşüm sonuçlarına göre işlem yapılır.

Bilgisayarlar, gömülü sistemler, dijital sinyal işlemciler gibi elektronik ekipmanlarda ayırık zamanlı Fourier dönüşümü kullanılır.

$$\overline{X}_k = \sum_{n=0}^{N-1} x(n) \cdot \cos\left(\frac{2\pi kn}{N}\right) - j \cdot \sum_{n=0}^{N-1} x(n) \cdot \sin\left(\frac{2\pi kn}{N}\right) \quad (4)$$

$$\overline{X}_k = X_{kr} + j \cdot X_{ki} \quad (5)$$

$$|\overline{X_k}| = \sqrt{X_{kr} + j \cdot X_{ki}} \quad (6)$$

$$\varphi_h = \arctan\left(\frac{X_{ki}}{X_{kr}}\right) \quad (7)$$

Formüllerde N örnek sayısı, $x(n)$ n. noktadaki giriş değeri, X_k k. harmonik vektör bileşeni, X_{kr} ve X_{ki} bu vektörün gerçekte ve sanal bileşeni ve φ_h faz açısıdır. Euler'in üstel denklemini kullanarak

$$e^{-jx} = \cos x - j \sin x \quad (8)$$

ayrık zamanlı Fourier dönüşümü şu formülle ifade edilir:

$$\overline{X_k} = \sum_{n=0}^{N-1} x(n) \cdot e^{-j\frac{2\pi nk}{N}} \quad k = 0, 1, 2, \dots, N-1 \quad (9)$$

Fourier dönüşümünde olduğu gibi ayrık zamanlı Fourier dönüşümünde de ters dönüşüm uygulanabilir. Fourier dönüşümünden elde edilen sonuçlar; istenmeyen frekansların filtre edilmesi ve daha sonra tekrar oluşturulması, işlenmiş ve Fourier dönüşümüyle sıkıştırılmış ses ve resim dosyalarının tekrar oluşturulması gibi işlemlerin gerçekleştirilmesi için ters ayrık zamanlı Fourier dönüşümü için işlenebilir. Bunun için aşağıdaki formül kullanılır.

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} \overline{X_k} \cdot e^{j\frac{2\pi nk}{N}} \quad k = 0, 1, 2, \dots, N-1 \quad (10)$$

3.5. Hızlı Fourier Dönüşümü

Ayrık zamanlı Fourier dönüşümü Fourier dönüşümünün elektronik cihazlar içinde kullanılabilir halidir. Hızlı Fourier dönüşümü (FFT) ayrık zamanlı sinyallerin ayrık Fourier Dönüşümü (DFT) hesaplanmasında kullanılan etkili bir metottur (Vardar vd., 2009; Rechka vd., 2002).

Ayrık zamanlı Fourier Dönüşümü yapılırken dönüşüm sonucu oluşacak her bir eleman için tüm örnekler boyunca karmaşık sayılarla çarpma ve toplama işlemleri gerekmektedir. Dönüşüm yapılacak veri miktarı arttıkça bu hesap yükü de katlanarak artmaktadır. Tepki süresinin çok önemli olduğu paralel aktif güç filtresi gibi gerçek zamanlı uygulama alanlarında böylesine yavaş hesaplamalar kullanışlı değildir. Ayrık zamanlı Fourier dönüşümünün çok olan hesap yükünü hafifletmek ve dönüşüm sonucunu hızlandırmak için çalışmalar yapılmış ve bu çalışmalar sonucunda Hızlı Fourier Dönüşümü (FFT) ismi verilen yöntem bulunmuştur. FFT

yöntemi Fourier dönüşümünün daha hızlı ve işlem yükü daha az olacak şekilde yeniden düzenlenmiş halidir.

FFT hesaplamasında farklı algoritmalar kullanılır. Bunlardan bazıları Radix2 (Cooley-Tukey) algoritması, Radix3, Radix4, Radix8, Split-Radix yöntemi, modifiye Split-Radix yöntemi, Gao'nun eklemeli FFT algoritması, kaymalı eklemeli FFT, Wang-Zhu-Cantor eklemeli FFT algoritmaları sayılabilir (Mateer, 2008). Günümüzde kullanılan bu sayılan algoritmalarından başka algoritmalar da mevcuttur.

Çoğu enerji kalitesi cihazları güç sistemi harmoniklerini ölçmek için FFT tabanlı yöntemler kullanır (Chang, 2008). Yapılan çalışmada basitliği, kolay anlaşılabilir olması, gömülü sistemlerde çalışabilen hızlı ve etkili bir algoritma kullanmasından dolayı FFT işlemi için Cooley-Tukey algoritması yani diğer ismiyle Radix2 algoritması kullanılacaktır.

3.6.Radix2 Metodu

Bu yöntem, ayrık zamanlı Fourier dönüşümünün daha hızlı gerçekleştirilebilmesi için J.W. Cooley ve John Tukey tarafından bulunmuştur. Bulunan bu yöntem Cooley-Tukey algoritması veya Radix2 FFT algoritması olarak da bilinir (Cooley ve Tukey, 1969).

Radix2 algoritmasının temel mantığı, tüm verilerin bir seferde ayrık zamanlı Fourier dönüşümünü gerçekleştirmek yerine verilerin ikiye bölünmüş gruplar halinde ayrık zamanlı Fourier dönüşümlerini gerçekleştirerek işlemlerin daha da basitleştirilmesini ve hızlandırılmasını sağlamaktır.

Ayrık zamanlı Fourier Dönüşümü

$$X(k) = \sum_{n=0}^{N-1} x(n) \cdot e^{-j\frac{2\pi nk}{N}} \quad k=0,1,2,\dots,N-1 \quad (11)$$

şeklinde tanımlanmıştır. Euler'in üstel denklemini kullanarak:

$$e^{-jx} = \cos x - j \sin x \quad (12)$$

ayrık zamanlı Fourier Dönüşümü

$$X(k) = \sum_{n=0}^{N-1} x(n) \cdot \left[\cos\left(\frac{2\pi nk}{N}\right) - j \sin\left(\frac{2\pi nk}{N}\right) \right] \quad k=0,1,2,\dots,N-1 \quad (13)$$

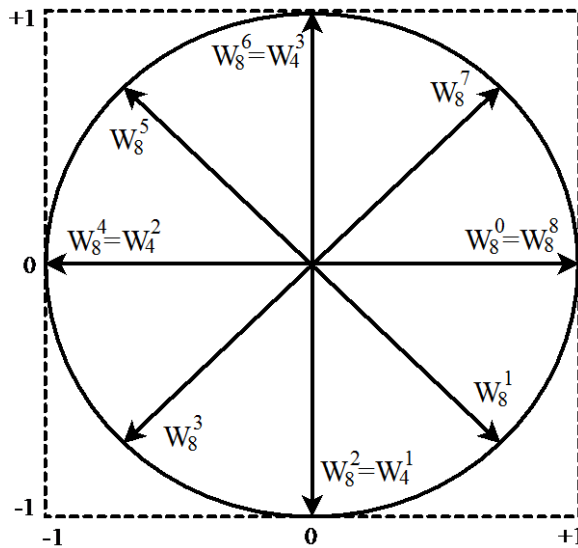
şeklinde tanımlanıp ifade edilir. Burada, dijital sinyal işlemci içerisinde önceden tanımlı olan kosinüs ve sinüs tabloları kullanılarak işlemler hızlandırılabilir. Ancak, yine de bu hesaplama yavaş kalacaktır. Hesaplamaları daha da hızlandırmak için ayrık zamanlı Fourier dönüşümü

$$X(k) = \sum_{n=0}^{N-1} x(n) \cdot W_N^{nk} \quad k=0,1,2,\dots,N-1 \quad (14)$$

şekline getirilir. Burada;

$$W_N^{nk} = e^{-j\frac{2\pi nk}{N}} \quad (15)$$

olarak ifade edilir. Formüldeki W_N^{nk} ifadesi üstel bir ifade olduğundan, veri sayısı olan N 'e göre periyodik ve veri sayısının yarısı olan $N/2$ ye göre simetriktir.



Şekil 3.2. Sekiz örnek için W durumları.

Şekil 3.2’de sekiz örnek için W_N^{nk} üstel fonksiyonunun yukarıdaki özelliklerini gösteren bir örnek çizim gösterilmektedir. Bu veri dizisi için:

- W_N^{nk} ifadesinin periyodiklik özelliği $W_N^{nk} = W_N^{nk+N} = W_N^{nk+2N}$ şeklinde ifade edilir. Şekil 3.2’den de görüleceği gibi $W_8^1 = W_8^9$ olacaktır.
- W_N^{nk} ifadesinin simetriklik özelliği $W_N^{nk+N/2} = -W_N^{nk}$ şeklinde ifade edilir. Şekil 3.2’de görüldüğü gibi $W_8^{1+4} = -W_8^1$ olmaktadır.

- W_N^{nk} ifadesinin özyineleme özelliği $W_{N/2}^{nk} = W_N^{2nk}$ şeklinde ifade edilir. $W_4^1 = W_8^2$ olacaktır.

Bu özelliklerden faydalananak, FFT için ayrık zamanlı Fourier dönüşümünü tek sayıların ve çift sayıların üstel çarpımları toplamı olarak düzenlendiğinde:

$$X(k) = \sum_{\text{Çift}} x(n). W_N^{nk} + \sum_{\text{Tek}} x(n). W_N^{nk} \quad (16)$$

$$X(k) = \sum_{n=0}^{N/2-1} x(2n). W_N^{2nk} + \sum_{n=0}^{N/2-1} x(2n+1). W_N^{(2n+1)k} \quad (17)$$

eşitlikleri elde edilir. Ayrık zamanlı Fourier dönüşümü, tek ve çift sayılarda çarpımların ayrı ayrı işlemler toplamı şeklinde eşitlik 16 ve 17 ile ifade edilmiştir. W_N^{nk} teriminin özelliklerinden faydalananak eşitlikler Radix2 algoritması için düzenlendiğinde:

$$X(k) = \sum_{n=0}^{N/2-1} x(2n). W_N^{2nk} + W_N^k \cdot \sum_{n=0}^{N/2-1} x(2n+1). W_N^{2nk} \quad (18)$$

$W_N^{2nk} = W_{N/2}^{nk}$ özelliğinden faydalananak,

$$X(k) = \sum_{n=0}^{N/2-1} x(2n). W_{N/2}^{nk} + W_N^k \cdot \sum_{n=0}^{N/2-1} x(2n+1). W_{N/2}^{nk} \quad (19)$$

yazılabilir.

$$\sum_{n=0}^{N/2-1} x(2n). W_{N/2}^{nk} = G(k) \text{ ve } \sum_{n=0}^{N/2-1} x(2n+1). W_{N/2}^{nk} = H(k) \quad (20)$$

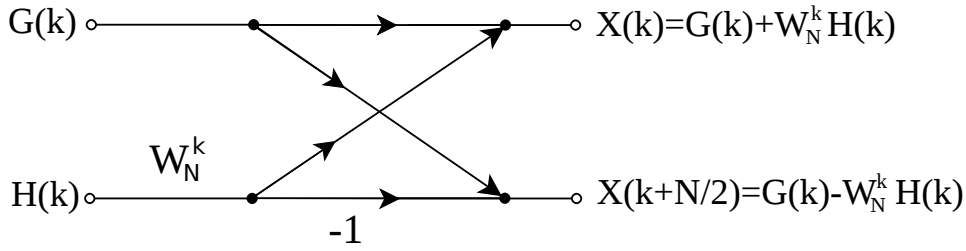
kabul edilirse, sonuç olarak;

$$X(k) = G(k) + W_N^k \cdot H(k) \quad k = 0, 1, 2, \dots, \frac{N}{2}-1 \quad (21)$$

ve

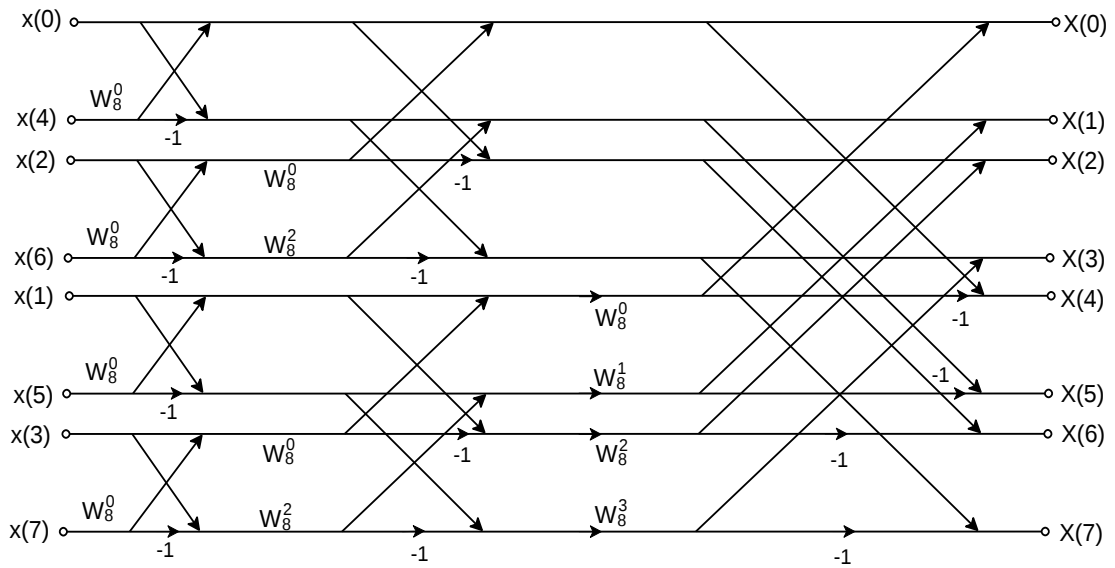
$$X\left(k + \frac{N}{2}\right) = G(k) - W_N^k \cdot H(k) \quad k = 0, 1, 2, \dots, \frac{N}{2}-1 \quad (22)$$

elde edilir. Yukarıdaki eşitlikler Radix2 algoritmasının temelini oluşturmaktadır. Verilerin ifade edildiği çift ve tek indeksler ayrı ayrı işleme alınarak sonuçlar elde edilir.



Şekil 3.3. İki örnek için Radix2 gösterimi.

Şekil 3.3.'de Radix2 temelli iki örnek için FFT dönüşümünün akış şeması gösterilmektedir.



Şekil 3.4. Sekiz örnek için Radix2 gösterimi.

Şekil 3.4'de ise sekiz elemanlı bir veri dizisinin Radix2 yöntemiyle FFT dönüşümü akış şeması gösterilmektedir. Çift ve tek elemanlar kendi aralarında ikili dönüşüm gerçekleşmekte ve bir sonraki dönüşüm aşamasına geçilmektedir. Dönüşüm aşamaları sayısı 2^n tanedir. Sekiz verilik bir dizi için dönüşüm aşamaları sayısı 2^3 'den üç olarak bulunur. Şekilden görüleceği gibi üç aşamalı dönüşüm yapılıncı tüm elemanların FFT işlemi tamamlanmış olur.

Hızlı Fourier dönüşümü ile ayrık zamanlı Fourier dönüşümü arasında işlem sayılarında fark vardır. Ayrık zamanlı Fourier dönüşümü için işlem yükü n^2 ile belirtilirken hızlı Fourier dönüşümü için $n \cdot \log_2 n$ ile hesap edilir (Heath, 2002). Bazı örnek miktarları için Çizelge 3.1.'de DFT ve FFT için işlem sayıları gösterilmektedir.

Çizelge 3.1. Ayrık zamanlı ve hızlı Fourier dönüşümü işlem sayıları.

Eleman Sayısı	Ayrık Zamanlı Fourier Dönüşümü işlem sayısı	Hızlı Fourier Dönüşümü işlem sayısı
64	4096	384
128	16384	896
256	65536	2048
512	262144	4096
1024	1048576	10240

3.7.Bit Tersleme

Hızlı Fourier dönüşümünün yapılabilmesi için veri dizisindeki elemanların Radix2 formülüne uygun olarak işleme alınması gerekmektedir. Bu formülde tek ve çift sayılar ayrılarak işleme alınır. Eğer Fourier dönüşümü alınacak dizi boyutu N olursa; sırasıyla dizi boyutu yarısına kadar olan her çift indisli eleman $N/2$ ilerisindeki çift indisli veri ile işleme alınır. Benzer olarak çift indisli sayılar işlendikten sonra sırasıyla dizi boyutu yarısına kadar olan her tek indisli eleman $N/2$ ilerisindeki tek indisli veri ile işleme alınır. Sekiz elemanlı bir dizi için ilk eleman $x(0)$, $x(0 + N/2)$ ile yani $x(4)$ ile işleme alınır. Benzer olarak çift indisli veriler işlendikten sonra tek indisli ilk veri $x(1)$, $x(1 + N/2)$ ie yani $x(5)$ ile işleme alınır. Sıralama için bit tersleme algoritması kullanılır. Bit tersleme işlemi için veri dizi indisindeki binary bitlerin simetrik olarak yer değiştirmesi yeterlidir. Radix2 algoritmasına göre FFT alınacak veri boyutu ikinin üstel katı olmalıdır. Çizelge 3.2.'de Sekiz eleman için bit tersleme işlemi görülmektedir.

Çizelge 3.2. Sekiz eleman için bit tersleme işlemi.

Desimal Numara	0	1	2	3	4	5	6	7
Binary Karşılığı	000	001	010	011	100	101	110	111
Terslenmiş Binary Karşılığı	000	100	010	110	001	101	011	111
Desimal Karşılığı	0	4	2	6	1	5	3	7

3.8. Pencereleme İşlemi

Sadece bir frekans içeren bir sinyalin frekans spektrumu çizilmek istendiğinde; sinyal sadece bir frekanstan oluştuğu için grafikte sadece o frekansta yükselme ve diğer tüm frekanslarda ise veri olmaması beklenir. Aynı sinyal için FFT alındığında ana frekanslara yakın frekanslarda da bileşenler gözlemlenir. Bu olaya spektral sızıntı denir ve FFT işlemleri için istenmeyen bir durumdur. Spektral sızıntıdan kurtulmak için FFT alınacak veriler, işlemde önce belli sayıda örnek içeren parçalara ayrılırlar. Bu parçalar, pencereler (window) olarak isimlendirilir (Kester, 2003). FFT için veriler pencerelere ayrıldıktan sonra pencereleme fonksiyonu denen basit bir fonksiyonla çarpılır. Sinüzoidal dalgalar için en uygun pencere tipi, Hanning'dir (National Instrument, 2015).

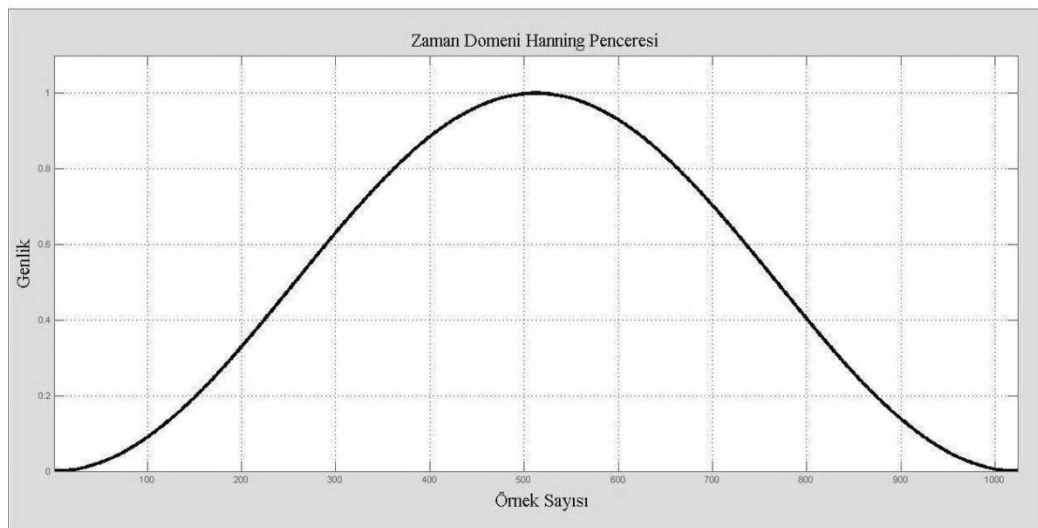
Hannig Pencere;

$$\omega(n) = \frac{1}{2} \cdot \left(1 - \cos\left(\frac{2\pi n}{N-1}\right) \right) \quad 0 \leq n \leq N \quad (23)$$

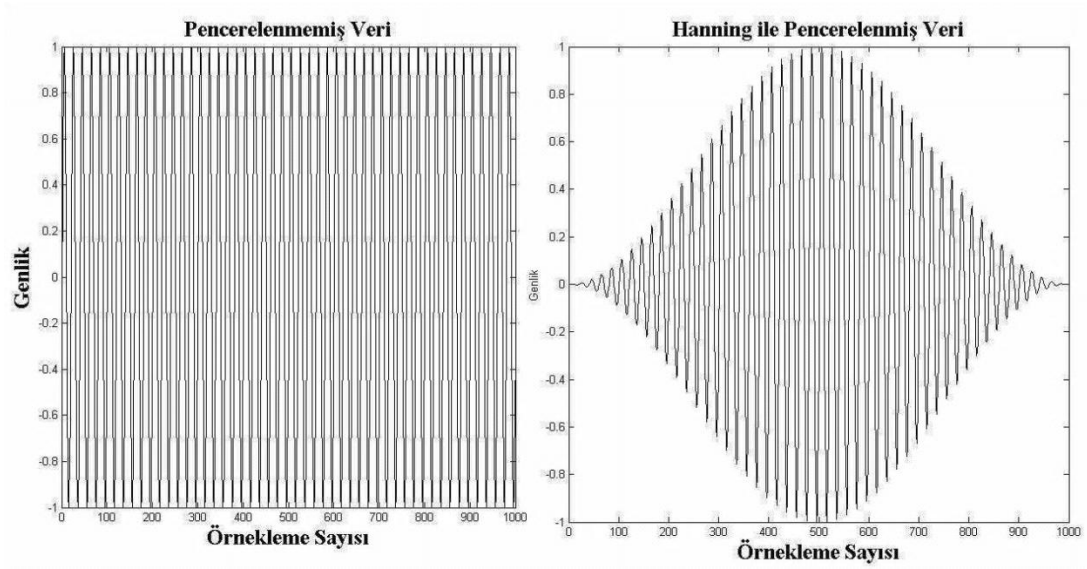
veya

$$\omega(n) = \sin^2\left(\frac{2\pi n}{N-1}\right) \quad (24)$$

ile ifade edilir. Formüllerdeki N sayısı pencere veri miktarını ifade etmektedir. Şekil 3.5.' de Hanning pencere ve Şekil 3.6.'da Hanning pencerenin uygulandığı örnek sinyal gösterilmektedir.



Şekil 3.5. Hanning pencere gösterimi.



Şekil 3.6. Hanning pencere yöntemi kullanılan örnek sinyal.

3.9. Toplam Harmonik Bozulma

Toplam harmonik bozulma, tüm harmonikli sinyallerin toplamının temel frekanstaki sinyale oranının karekökü olarak tarif edilir (Kennedy, 2000). Toplam harmonik bozulma, hem gerilim ve hem de akım için hesaplanabilir. Gerilim için toplam harmonik bozulma aşağıdaki formülle gösterilir:

$$THD_V = \frac{\sqrt{\sum_{h=2}^{\infty} V_h^2}}{V_1} \quad V_1: \text{Ana frekans gerilimi} \quad (25)$$

Akım için toplam harmonik bozulma aşağıdaki formülle gösterilir:

$$THD_I = \frac{\sqrt{\sum_{h=2}^{\infty} I_h^2}}{I_1} \quad I_1: \text{Ana frekans akımı} \quad (26)$$

Çizelge 3.3. ve Çizelge 3.4.' de IEEE 519 standardına göre THD gerilim ve akım sınır değerleri gösterilmektedir. Çizelge 3.3.'de 69kV altı gerilimler için THD bozulma en fazla %5 olarak öngörülmüştür.

Çizelge 3.3. IEEE 519 standartına göre gerilim THD sınır değerleri.

Ortak Bağlantı Noktasındaki Gerilim	Tek harmonik bileşeni için bozulma (%)	THD (%)
69KV ve altındaki gerilimler	3.0	5.0
69.001 kV ile 161 kV arasındaki gerilimler	1.5	2.5
161.001 kV ve üstündeki gerilimler	1.0	1.5

Çizelge 3.4.' de gösterilen değerler, elektrik tüketen müşterilerin ortak bağlantılarının bulunduğu elektrik dağıtım noktasını için öngörülen değerlerdir.

Çizelge 3.4. IEEE 519 standartına göre akım THD sınır değerleri.

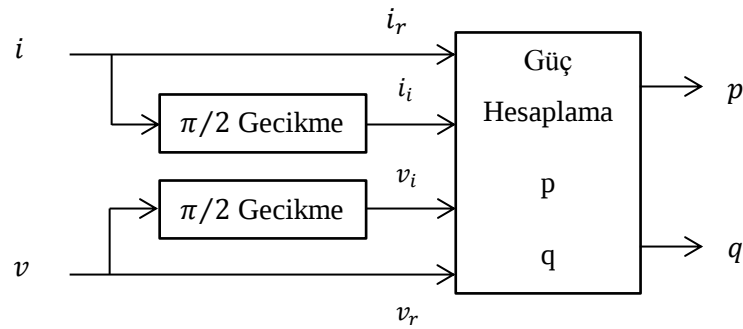
V < 69kV						
I_{SC}/I_L	H < 11	$11 \leq h < 17$	$17 \leq h < 23$	$23 \leq h < 35$	$35 \leq h$	THD (%)
<20	4.0	2.0	1.5	0.6	0.3	5.0
20-50	7.0	3.5	2.5	1	0.5	8.0
50-100	10.0	4.5	4.0	1.5	0.7	12.0
100-1000	12.0	5.5	5.0	2.0	1.0	15.0
>1000	14.0	7.0	6.0	2.5	1.4	20
I_{SC} : Ortak bağlantı noktasındaki maksimum kısa devre akımı						
I_L : Ortak bağlantı noktasından çekilen maksimum yük						

3.10. Ölçüm Metodu

Aktif ve reaktif gücün anlık olarak hesaplanmasında üç fazlı sistemler için yaygın olarak anlık reaktif güç teorisi kullanılmaktadır (Akagi, vd., 1983). Bu yöntem ile sıfır dizi (zero sequence) bileşeni alınmadığı için üç fazlı sistemlerde dengesiz ve sinüsoidal olmayan şartlarda hatalı sonuçlar verebilmektedir. Bu çalışmada, tasarlanan sistemin tek fazlı sistemlerde de kullanılması amacıyla tek fazlı anlık reaktif güç teorisinden faydalanılmıştır. Anlık reaktif güç hesaplama blok diyagramı Şekil 3.7'de gösterilmektedir. Bu teoriye uygun olarak üç fazlı

sistemler için her faz ayrı ayrı değerlendirilir. Tek faz anlık reaktif güç teorisinin dengesiz yüklü üç fazlı sistemlere uygulanması sonucu sinüsoidal referans akımları da elde edilebilmektedir (Haque ve Ise, 2002). Bu özelliğiyle aktif güç filtreleri için de kullanışlı bir metottur (Dobrucký vd., 2002).

Tek faz anlık reaktif güç teorisinde elde edilen akım ve gerilim ölçümleri ve bunların $\pi/2$ kadar geciktirilmiş ölçümleri işleme alınmaktadır. Şekil 3.7.'de anlık reaktif güç hesaplama blok diyagramı gösterilmektedir.



Şekil 3.7. Anlık reaktif güç hesaplama (Vardar vd., 2009).

Tek faz anlık güç teorisine göre aktif güç

$$p = v_r * i_r + v_i * i_i \quad (27)$$

formülüyle hesaplanır. v_r ve i_r anlık akım ve gerilim değerlerini temsil etmektedir. v_i ve i_i anlık akım ve gerilim değerinin $\pi/2$ geciktirilmiş değerleridir.

Reaktif güç ise

$$q = v_r * i_i - v_i * i_r \quad (28)$$

formülüyle hesaplanmaktadır.

Görünür gücü hesaplamak için de

$$S = \sqrt{p^2 + q^2} \quad (29)$$

formülü kullanılır.

4. GERÇEKLENEN SİSTEMİ TASARIMI

Bu bölümde, tasarlanan sistemle ilgili yazılımsal ve donanımsal bileşenler incelenecektir. Uygulaması yapılan güç kalitesi izleme sistemi, mini-itx tabanlı gerçek zamanlı gömülü sistem ve bu sisteme dış dünyadan verilerin uygun olarak alınması, şartlandırılması, önilem ve iletimi sağlayan donanım bileşenleri olarak iki ayrı birim olarak tasarlanmıştır. Sistemde akım sensörleri ve gerilim sensörlerinden alınan bilgiler sinyal şartlandırıcı tarafından dijital sinyal işlemcisinin işleyebileceği şekle getirilmekte ve yapılan ölçümler USB arabirimi üzerinden gömülü sisteme aktarılmaktadır. Gömülü sistem için tasarlanan yazılım, verileri işleyerek görüntüler ve rapor halinde kaydeder.

4.1. Gömülü Sistemler

Gömülü sistemler, özel amaçlarla sadece belli bazı görev veya görevleri yerine getirebilecek şekilde tasarlanmış yazılım kodlarının bir bilgisayar-donanım sistemine gömülmesiyle oluşan sistemdir (Kamal, 2008).

Gömülü sistemlerde kullanılan bilgisayar donanımı; sistem tasarımını kolaylaştırmak ve esneklik sağlamak için kullanılır. Kodlar genellikle kalıcı bellekte saklanır. Elektronik üretim teknolojisindeki gelişmelerle beraber gömülü sistemler, günümüzde kullanılan çoğu elektronik cihaz içerisine girecek kadar ucuzlamıştır. Örnek kullanım alanları olarak biyomedikal cihazlar, bankamatikler, network cihazları, beyaz eşyalar, cep telefonları, sanayi tipi robotlar, nümerik kontrollü cihazlar, endüstriyel kontrol ve endüstriyel ölçüm cihazları sayılabilir (Hamblen, 2007).

Gömülü işletim sistemleri, gömülü sistemler için tasarlanmış işletim sistemleridir. Diğer işletim sistemlerinden farklı olarak daha kararlı, daha hızlı, bellek ve güç kullanımında daha etkili işletim sistemleridir. Çok sınırlı miktarda bellek kullanabilirler. Genel işletim sistemlerindeki çoğu görevi yerine getirmekten daha çok sadece özelleşmiş programlar için destek verirler. Çoğunlukla gömülü işletim sistemleri, gerçek zamanlı işletim sistemi olarak da anılır (Chu ve Fang, 2015).

Standart işletim sistemi, kullanıcı ile bilgisayarın etkileşimini sağlamak için arka planda çok sayıda program ve hizmet yığını çalıştırırken, gömülü işletim sistemlerinde sınırlı sayıda program veya hizmet çalıştırılır (Wang, vd., 2001).

4.1.1. Mini-itx anakart

Güç şebekesinden alınan akım ve gerilim bilgileri gömülü sisteme aktarılmakta ve gömülü sistem tarafında çalışan bir yazılımla işlenerek görüntülenmekte ve raporlanmaktadır. Bu amaçla, tasarlanan sistemde Blkd2500hn adlı mini-itx anakart kullanılmıştır. Ürünün ucuz olması, piyasada çok rahat bulunabilmesi, üzerinde işlemcisi olması, genişletilebilir bellek özelliklerine sahip olması nedeniyle tercih edilmiştir. Mini-itx ve micro atx uyumlu kart; 17cm x 17cm boyutlarıyla elektrik panoları içerisinde de rahatlıkla kullanılabilecektir. Gömülü sistem kartı Şekil 4.1.'de gösterilmektedir.



Şekil 4.1. Çalışmada kullanılan gömülü sistem kartı (Intel Corporation, 2011).

4.1.2. Windows embedded standart

Hazırlanan gömülü yazılımın, mini-itx tabanlı sistemde çalışabilmesi için Windows 7 Embedded işletim sistemi kurulmuştur. Windows 7 embedded standart gömülü işletim sistemi, gömülü sistemler için Microsoft firması tarafından tasarlanmış bir işletim sistemidir. Windows embedded standart 7 işletim sistemi, Vista ve Windows 7 den özellikler taşır.

Windows embedded standart 7 işletim sistemi, hastane cihazlarından market cihazlarına kadar geniş bir yelpazede özelleştirilip kullanılabilir. Hızlandırılmış başlama zamanı ve hızlı getirme teknolojisi ile aktif uygulamalara daha hızlı cevap verir. Kullanılan teknolojiler sayesinde enerji gereksinimleri azaltılmıştır. Sistem kararlılığı ve güvenliği daha da artırılmıştır. Windows embedded standart işletim sistemi kurulumu sırasında hazır şablonlardan faydalanarak kurulum yapılabildiği gibi sadece gereksinim duyulan paketler yüklenerek kurulum boyutu azaltılabilir. Bir kez kalıp dosyası oluşturulduktan sonra istenildiği kadar cihaza kalıp dosyası aktarılabilir (Microsoft, 2010).

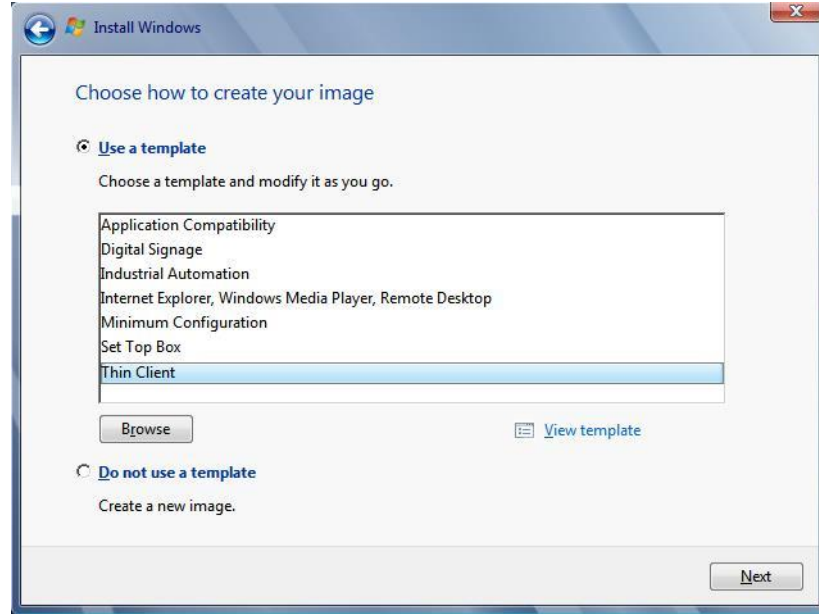
Windows embedded standart 7 işletim sistemi, IPv6, 64bit cpu, sd kart ve USB boot gibi teknolojileri desteklemektedir. Internet Explorer 8, Windows Media Player 12, Remote Desktop Protocol 7, Silverlight 3, .Net Framework 3.5 SP1 gibi teknolojileri bünyesinde barındırır. Bu teknolojiler, program geliştiricilerinin daha verimli ve daha etkili, görsel programlar üretmeleri için önemlidir. Windows Update hizmeti ile sistemi otomatik olarak güncelleme imkânı vardır (Microsoft, 2011).

Windows embedded standart gömülü işletim sistemi, masaüstü işletim sistemlerinden biraz farklıdır. Gömülü işletim sistemi, yapısı gereği özellikleri aynı olan çok sayıda gömülü sisteme kurulabilecek şekilde geliştirilmiştir. Bir üretici, cihaz ürettiği gömülü donanım ve bu donanımın gerçekleştireceği işleme uygun işletim sistemini bir kez bir cihaz için kurar. Bu cihazda bulunan işletim sistemine imaj dosyası adı verilir. İmaj dosyasını oluşturmak biraz zaman alabilir. Bu imaj dosyası, gömülü işletim sistemi araçlarıyla saniyeler içerisinde yeni pek çok cihaza aktarılabilir. Dolayısıyla zamandan tasarruf edilmiş olur.

Windows embedded standart 7 ilk kurulum ekranında karşımıza iki seçenek çıkmaktadır. Şekil 4.2.'de gösterilen kurulum ekranında ilk seçenek imaj dosyası oluşturmak için, ikinci seçenek ise kurulu imaj dosyasını kalıp olarak başka cihazlara kurulum yapmak için kullanılır.



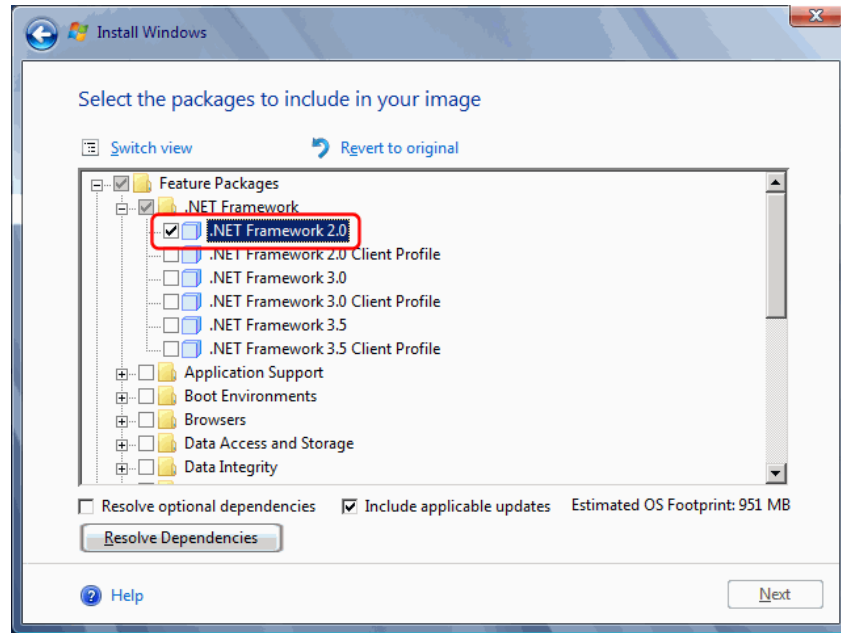
Şekil 4.2. Windows embedded standart 7 ilk kurulum ekranı.



Şekil 4.3. Windows embedded standart 7 şablondan kurulum ekranı.

İlk kurulum için imaj dosyası seçeneği seçildiğinde, kurulum ekranında lisans koşulları kabul edildikten sonra şifre ekranına geçilir. Uygun lisans anahtarı girildiğinde Şekil 4.3.’deki gibi kullanılabilir standart şablonlardan birinin seçilmesi beklenir. Bu seçeneklerden “Application compability” seçeneği, tüm yazılımsal kütüphaneleri yükleyerek gömülü sistemde çalışacak program için tüm destek yazılımlarını kurar. “Digital signage” seçeneği, billboard denilen elektronik reklam panoları için tasarlanmıştır. “Industrial automation” seçeneği, endüstride kullanılan cihazlar için gereken en az kurulumu gerçekleştirir.

Bu hazır şablonlardan biri seçilerek kurulum işlemine devam edilebileceği gibi “Do not use template” seçeneği de seçilebilir. Bu durumda kurulum Şekil 4.4.’deki işletim sistemine kurulacak bileşenlerin seçileceği ekranla devam edilir.



Şekil 4.4. Windows embedded standart 7 seçimli kurulum ekranı.

Kurulumun bu aşamasında, gömülü işletim sisteminde kurulacak olan özellik gruplarını belirlememiz beklenir. İşletim sistemini oluşturan tüm öğeler buradan belirlenerek hem işletim sisteminin daha hızlı çalışması sağlanır ve hem de gereksiz özellikler kurulmadığından depolama için daha az alan kullanılmış olur. Olası bir uyumsuzluk halinde kurulum tarafından otomatik olarak düzeltme işlemi yapılmaktadır.

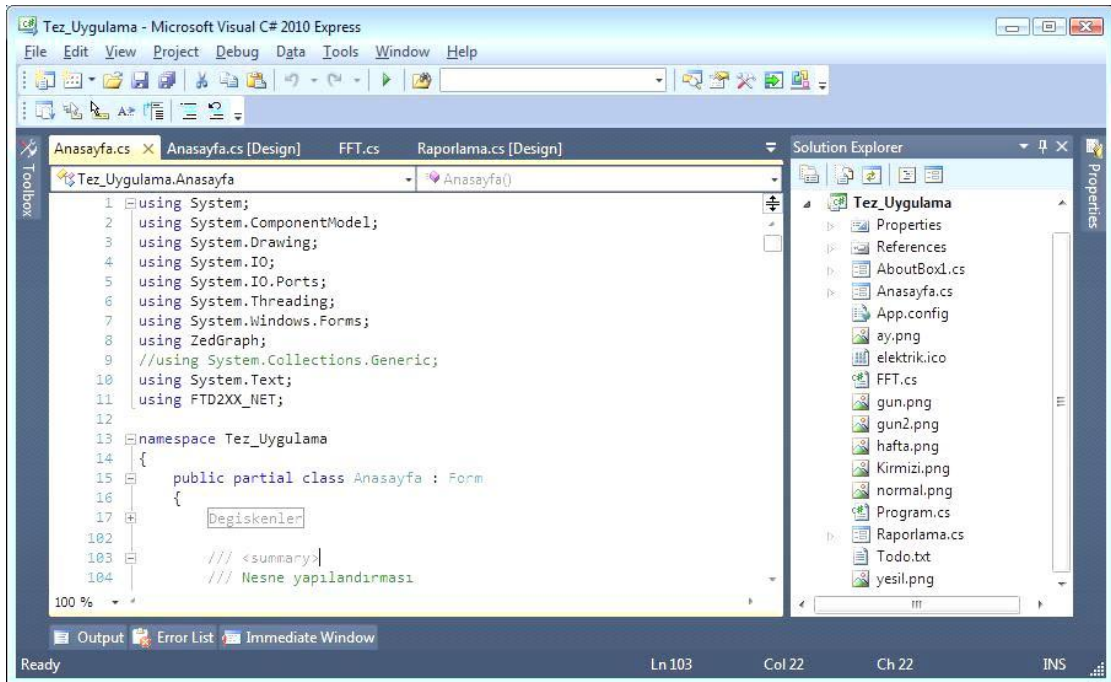
Kolay kurulumu, masaüstü işletim sistemlerine göre daha hızlı ve daha kararlı olması, masaüstü işletim sistemlerinin desteklediği teknolojileri içermesi ve program geliştirme sürecinin daha kolay olması sebebiyle bu çalışmada Windows embedded standart 7 sürümü kullanılmıştır.

4.1.3. Gömülü sistem arayüz yazılımı

Gömülü sistemde kullanılan yazılım, C# programlama dili kullanılarak hazırlanmıştır. C# dilinde uygulama geliştirmek için Visual C# 2010 Express uygulama geliştirme ortamı kullanılmıştır. Bu program Microsoft firması tarafından ücretsiz olarak sunulan bir yazılım geliştirme setidir. Kod geliştirme ve arayüz tasarlama, veri tabanı yönetimi işlemlerinde kullanılabilir (Mueller,2011).

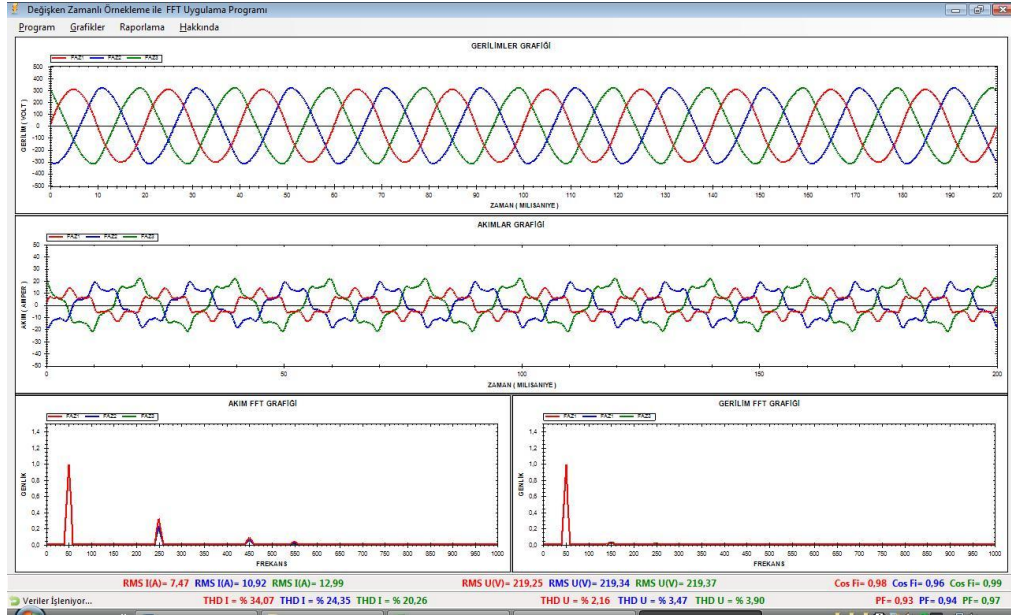
C# programlama dili, program geliştiriciler arasında oldukça popüler olan ve her platformda çalışabilme özelliği ile öne çıkan JAVA programlama diline çok benzer. C# tamamen nesne yönelimli programlama mantığı üzerine kurulmuştur. Kod söz dizimi C ve C++

programlama dillerine benzer. C# kullanım olarak basit ve güçlü bir programlama dilidir. C# programlama dili, gömülü sistemler için de uygundur. İşletim sistemi fonksiyonlarını doğrudan çağırarak kullanılabilecek kadar yeteneklidir. C# programlama dili, hafıza ve işlemcinin daha verimli kullanılması için C ve Assembly dilleri ile rekabet edebilecek düzeyde tasarlanmıştır. Tüm bu özelliklerinin yanında C# programlama dili için üretilen açık kaynak kodlu derleyiciler de mevcuttur. Derlenen kodlar JAVA platformuna benzerlikleri bulunan .NET platformunda çalışacak şekilde oluşturulur. .NET platformu, Microsoft firması tarafından geliştirilen uygulama geliştirme platformudur. Masaüstü uygulamalarından web tabanlı uygulamalara kadar tüm uygulamalar tek çatı altında geliştirilebilir. Firma tarafından önceden hazırlanmış kod parçaları, uygulama geliştirme süresini kısaltır. Yazılan kodlar Linux işletim sistemi de dâhil .NET platformunun veya uyumlu kütüphanelerinin yüklü olduğu tüm sistemlerde çalışır. Yazılımcıların zorluk yaşadığı bellek yönetimi, uygulama işletim sistemleri arasında uyum, yazılım güvenliği gibi konularda otomatikleştirilmiş araçlar sunmaktadır. C#, NET çatısı altında kullanılan programlama dillerinden biridir (Sinha, 2015; Rahman, 2012; Troelsen, 2010). Şekil 4.5.'de gömülü sistem tasarımı için kullanılan Visual C# geliştirme ortamı gösterilmektedir.



Şekil 4.5. Visual C# geliştirme ortamı.

Gerçekleştirilen çalışmada sinyal şartlandırıcı karttan gelen bilgiler gömülü sisteme aktarılmakta ve gömülü sistem tarafında çalışan bir yazılımla işlenerek raporlanmaktadır. Tasarlanan ara yüz yazılımının ekran görüntüsü Şekil 4.6.'da görülmektedir.



Şekil 4.6. Gömülü sistem ara yüz programı.

Sinyal şartlandırıcı karttan alınan akım ve gerilim bilgileri USB arabirimi yoluyla gömülü sistem yazılımına aktarılır. Veri aktarımı başlamadan önce sinyal şartlandırıcının gömülü sisteme bağlanıp bağlanmadığı gömülü sistem tarafında çalışan yazılım tarafından kontrol edilir. Eğer sinyal şartlandırıcı kartı gömülü sisteme bağlı değilse gömülü sistem yazılımında bağlantı kurulamadığı belirtilir. Gömülü sistem yazılımı sinyal şartlandırıcı kartla bağlantı kurduktan sonra yazılım tarafından veri akışı başlatılıp durdurulabilir.

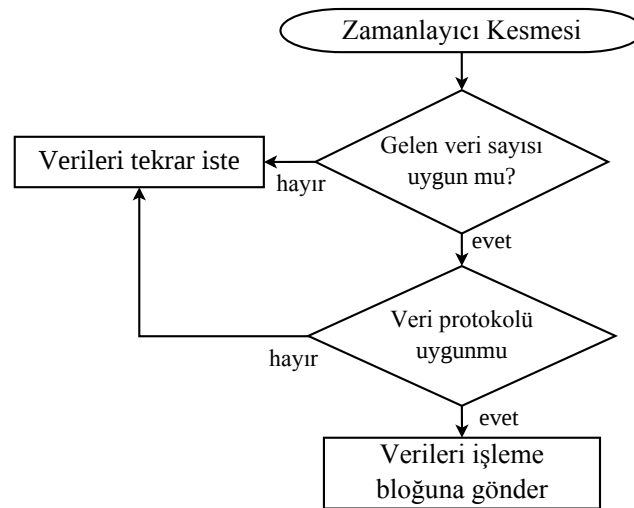
Şebekeden okunan bilgilerin gömülü sisteme aktarımında USB protokolü kullanılmaktadır. Sinyal şartlandırıcıdan gönderilen veriler USB tampon belleğinde biriktirilir. Gelen her 200msn'lik veri bloğu için işlem yapıldıktan sonra bir sonraki işlem için USB tampon belleğinde tutulan veri bloğunun tümü alınır.

USB gönderiminde kullanılan protokolde veri paketleri ile beraber iletişim ve veri bütünlüğünü doğrulamak için başka paketler de gönderilir ve süreç oldukça karmaşıktır (Axelson, 2009). Gönderilen veriler okunurken sinyal şartlandırıcı tarafında bulunan ve FTDI firması tarafından ücretsiz olarak dağıtılan USB dinamik link dosyaları kullanılır. FTDI firmasının üretmiş olduğu dinamik link dosyaları aracılığıyla kullanılan protokol çözülerek

doğrudan gelen veri paketlerine erişmek çok daha kolaydır. Böylece USB arabirimini yönetmek için gereken işletim sistemi bildirimleri firma tarafından verilen dosya ile yürütülür. Tasarlanan sistemde kullanılan FT2232 entegresi, 3 Mbit veri aktarım hızında izin vermektedir. Bu hız ise tasarlanan sistemin gereksinimleri için yeterlidir.

Seri veri alımı ve gönderiminde USB paket iletim yöntemlerinden biri olan yığın iletim yöntemi kullanılır. Yığın iletim yöntemi ile büyük miktarda veri iletimi gerçekleştirilebilir. Bu iletim yöntemi sadece tam hızlı ve yüksek hızlı USB cihazlarında kullanılabilir. Yığın iletim yönteminde gönderilen verilerin gönderim kontrolü yapılarak karşı tarafa doğru bir şekilde iletildiği garanti edilir (Axelson, 2009).

Tasarlanan örnekleme frekansına uygun olarak şebekeden okunan 10 sayıklık kısımdaki akım ve gerilim değerleri, sinyal şartlandırıcı kart tarafından sayısal hale dönüştürülerek gömülü sisteme aktarılır. Gömülü sistem tarafında çalışan yazılım, gelen veri bloğunun başlangıç protokolünü ve veri bloğunun uygun boyutta olup olmadığını kontrol eder. Veri bloğu başlangıç protokolü, her veri bloğunun öncesinde gönderilen ve yeni bir 10 sayıklık veri bloğu aktarımının başladığını gömülü sisteme bildiren dört byte uzunluğunda bir bilgidir. Gömülü sistem, başlangıç protokolünü hatalı bulduğunda veya gönderilen veri bloğu boyutunun eksik veya fazla olduğunu belirlediğinde sinyal şartlandırıcı karttan tekrar veri bloğu göndermesini ister. Veri bloğu başlangıç protokolü ve veri bloğu boyut kontrolü akış diyagramı Şekil 4.7. 'de gösterilmiştir.



Şekil 4.7. Gömülü sistem yazılımı veri aktarım algoritması.

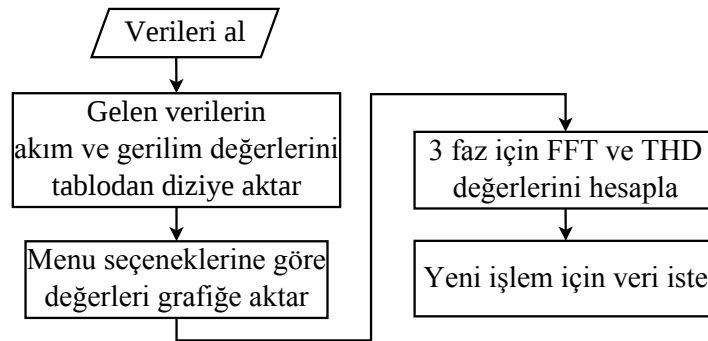
Sinyal şartlandırıcı tarafından gönderilen veri bloğu, gömülü sistem tarafından alınıp kontrol edildikten sonra işlem yapılması için hazırlanır. Sinyal şartlandırıcı karttan gönderilen veriler, sinyal şartlandırıcı kart tarafından dc offset eklenerek gönderilmiş verilerdir. Şebekeden alınan sinyallere dc offset eklenmesinin sebebi, sinyalin negatif ve pozitif kısımlarının tam ve doğru olarak ölçülebilmesi içindir. Gelen verilerden offset ayrılarak sinyalin tam şekli hesaplamalara katılır. Dc offset ayırma işlemini gerçekleştirmek için gelen her veri bloğunda işlem yapmak yerine sadece bir kez program açılışında hesap yapılarak program içine tablo halinde kaydedilir. Veri bloğu geldiğinde doğrudan tablodan değeri okunup çevrim gerçekleşir. Bu şekilde hem işlemci hesap yükü azaltılır, hem de toplam işlem süresi kısaltılmış olur. Gömülü sistem yazılımı ilk çalıştırıldığı anda ilk ayarlar yapılırken bir kereye mahsus bu kodlar aktif olarak sadece bir kere hesaplama yapılır. Şekil 4.8.'de bu kodlar verilmiştir.

```
gerilimDeger = new double[16, 256];
akimDeger = new double[16, 256];

for (int a = 0; a < 16; a++)
    for (int b = 0; b < 256; b++)
    {
        double sira = (a * 256 + b - 2047);
        gerilimDeger[a, b] = sira / 4;
        akimDeger[a, b] = sira / 64.4;
    }
```

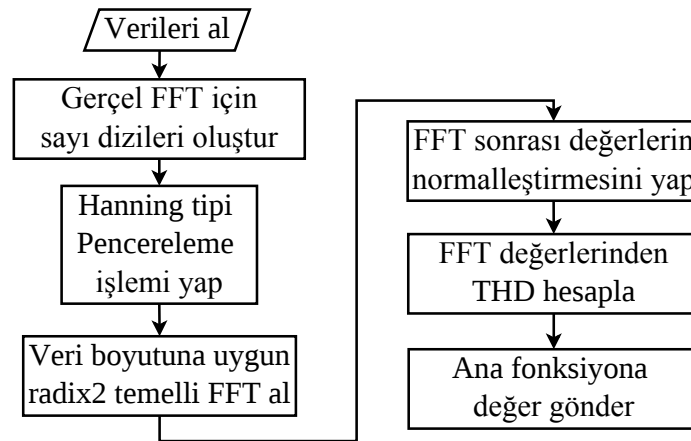
Şekil 4.8. Veri çevrimi kodları.

Gelen verilerden artık şebekedeki gerçek akım ve gerilim değerleri olarak elde edilmiş olur. Bu şekilde gelen veriler bundan sonraki aşamalar için hazırlanmış olmaktadır. Daha sonra FFT ve THD hesaplamaları, aktif ve reaktif güç hesaplamaları yapılarak kayıt tutulur. Bu işlemler için akış diyagramı Şekil 4.9.'da gösterilmektedir.



Şekil 4.9. FFT ve THD için işlem sırası.

Harmonik hesaplamalarını yapabilmek için işlenmeye hazır hale getirilen verilerin FFT dönüşümlerini almak gerekir. Bunun için kurulan yazılım yapısında FFT isimli bir nesne oluşturulmuştur. FFT işlemi için gerçek katsayılı FFT yöntemi kullanılmıştır. Bu yöntemde sadece genlik değerleriyle ilgilenildiği için faz kayması değerleri hesaplanmaz. Sadece amaca uygun hesaplamalar gerçekleştirileceği için toplam işlem süresi kısalmır. Cihaz, gelen verileri daha kısa sürede işleyebilir. FFT hesapları için program algoritması Şekil 4.10.'da verilmiştir.



Şekil 4.10. FFT işlem algoritması.

Gelen veri bloğunun FFT hesaplarını yapmak için FFT nesnesi ve bu nesneye bağlı değişkenler ve fonksiyonlar anlık olarak hesaplama yapılacağı zaman oluşturulur. Nesne oluşturulup hazırlandıktan sonra veri bloğunu pencereleme işlemine geçilir. Pencereleme işlemi FFT işleminden alınan sonuçların daha doğru gerçekleşmesi için yapılır. Sinüsoidal dalgalar için önerilen en verimli yöntem Hanning yöntemidir (National Instrument, 2015). Hanning yöntemi oldukça basittir. Şekil 4.11.'de Hanning metodu kod parçası verilmiştir.

```

// FFT değerlerinin daha iyi hale getirilmesi için hannig çerçeveleme yapılıyor.
for (i = 0; i < ornekSayisi; i++)
{
    if (i < veri.length)
    {
        // Hanning window uygulanıyor. 0.5*(1-cos(2*pi*n/N-1))
        double hanning = 0.5 * (1 - Math.Cos(2 * 3.14159 * i / (ornekSayisi - 1)));
        gercekSayi[i] = veri[i] * hanning; // karmaşık sayı reel kısmı
    }
    else gercekSayi[i] = 0;
    sanalSayi[i] = 0; // karmaşık sayı sanal kısmı
}

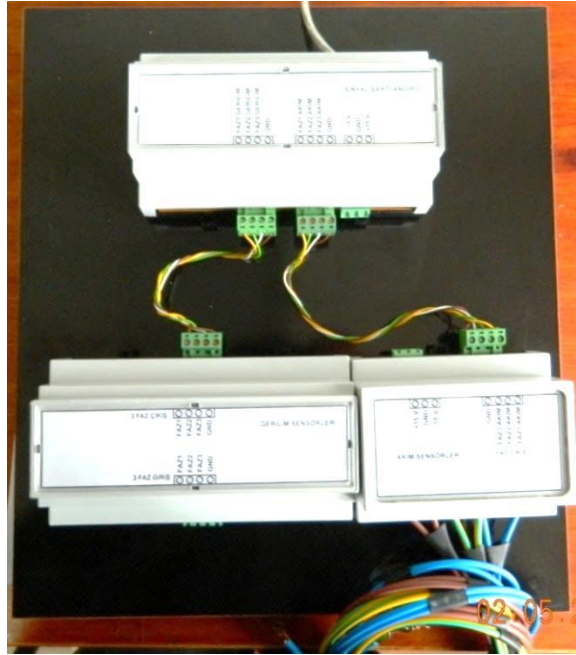
```

Şekil 4.11. Hanning pencereleme kodları.

FFT işlemlerinin yapılmasıyla sonuçların sıralı olabilmesi için bit tersleme işlemi gerçekleştirilir. Bit tersleme işlemi sonrasında Radix-2 yöntemi kullanılarak FFT işlemi gerçekleştirilir. Gerçekleştirilen FFT işleminden sonra THD hesabına geçilir. Elde edilen harmonik katsayıları THD formülüne uygun olarak işleme alınır. FFT ve THD hesapları tamamlandıktan sonra anlık aktif ve reaktif güç hesaplamaları her faz için ayrı ayrı yapılarak kayıt edilir. Bu işlemler bittiğinde, gömülü sistem tarafındaki yazılım sinyal şartlandırıcıdan gelen veri bloğunu işleme alarak sürecin devamını sağlar.

4.2. Yardımcı Donanımlar

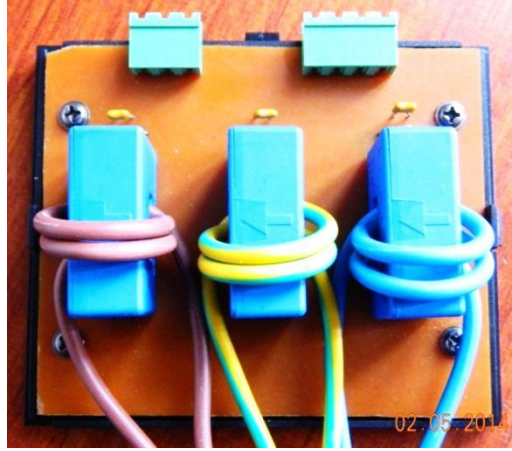
Enerji kalitesi izleme sistemi için şebekeden akım ve gerilim değerlerini ölçecek akım ve gerilim sensör kartları, akım ve gerilim değerlerini gömülü sisteme aktarmak üzere sinyal şartlandırıcı kart tasarlanmıştır. Şekil 4.12.'de gömülü sisteme bilgi aktarımını gerçekleştirmek için kurulan akım ve gerilim sensör kartı ve sinyal şartlandırıcı kart donanımları gösterilmektedir.



Şekil 4.12. Sinyal şartlandırıcı, akım ve gerilim sensör kartları.

4.2.1. Akım ölçme kartı

Akım ölçme kartı; şebekeden çekilen akımları sensör yardımıyla alıp sinyal şartlandırıcıya ulaştırmak için tasarlanmıştır. Tasarlanan akım ölçme kartı Şekil 4.13.'de gösterilmektedir.



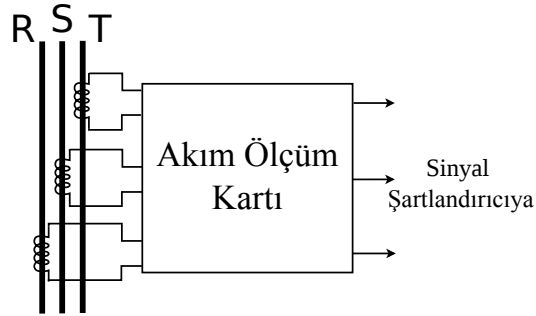
Şekil 4.13. Akım ölçme kartı.

Bu devrede, akım sensörü olarak LEM şirketinin LA100-P hall-etkili akım sensörleri kullanılmıştır. Akım bilgisi bu devre yardımıyla en fazla %0.45 hata oranı ile elde edilmektedir. Çekilen yüksek akımlar sensörler yardımıyla okunup elektronik olarak işlenebilecek seviyelere getirilir. Şekil 4.14.'de LA100-P akım sensörü gösterilmektedir.



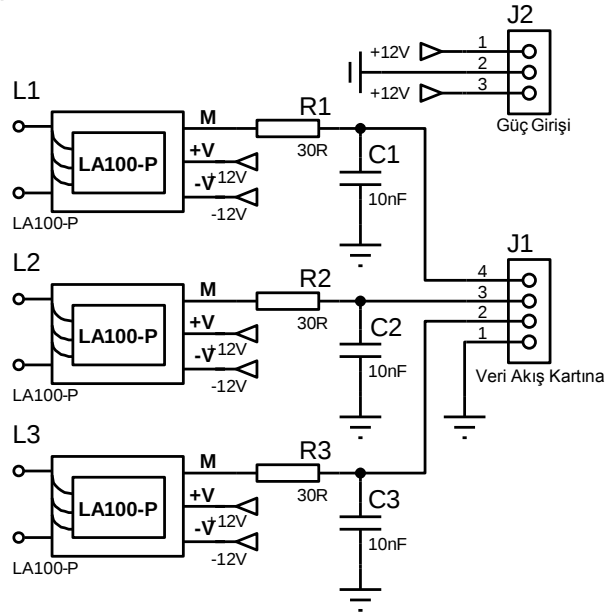
Şekil 4.14. Akım ölçme kartında kullanılan LA100-P sensörü (LEM, 2015).

Şekil 4.15.'de tasarlanan akım ölçme kartının elektrik iletim sistemine bağlantısı görülmektedir. Sensörlerin içindeki boş pencereden akım taşıyan kablolar geçmektedir.



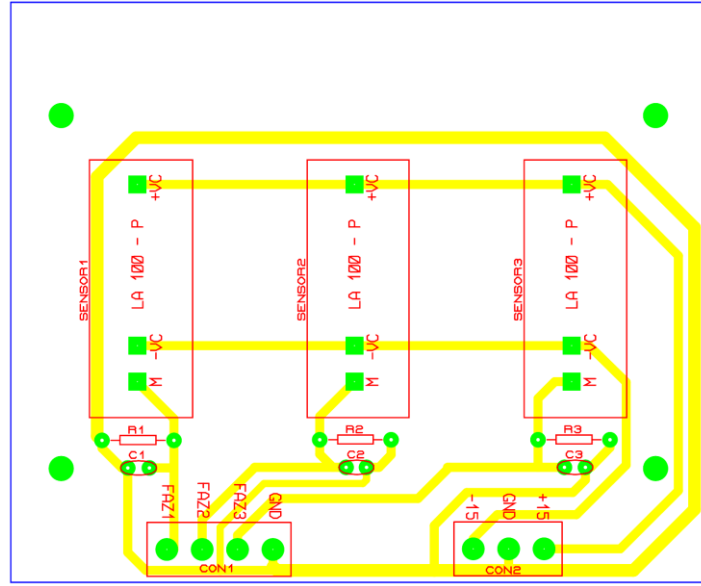
Şekil 4.15. Akım ölçme kartının iletim sistemine bağlantısı.

LA100-P akım sensörü hall-effect prensibiyle çalışmaktadır. İçinden geçen kablunun taşıdığı akım sebebiyle kablunun etrafında manyetik alan oluşur. Kablo tamamen sensörün içerisindeki boş pencereden geçtiği için sensör kablodaki manyetik alanı algılar ve kablunun taşıdığı akıma göre çıkışında bir değer gösterir. 100 Ampere kadar olan akımlar 1:2000 düşürme oranıyla ölçülebilir. Simetrik $\pm 12V$ DC besleme ile çalışır. 100 Amper akım çekildiğinde sensör sekonder uçlarında 50mA akım görülmektedir. Sekonderde görülen değere göre çekilen akım hesaplanabilir.



Şekil 4.16. Akım ölçme kartı devre şekli.

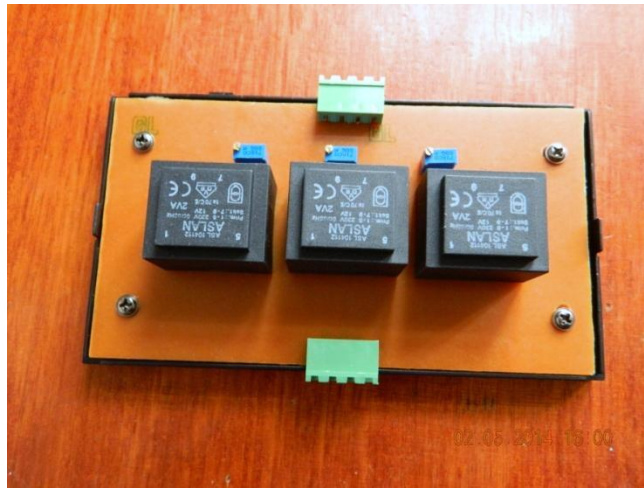
Şekil 4.16.'da 3 fazlı bir sistem için tasarlanan akım sensör kartının devre şeması gösterilmektedir. Şekil 4.17.'de tasarlanan kartın baskı devre çizimi gösterilmektedir.



Şekil 4.17. Akım ölçme kartı baskı devre şekli.

4.2.2. Gerilim ölçme kartı

Gerilim ölçme kartı, bağlı bulunduğu sistemde üç faz üzerindeki gerilimi ölçerek elektronik olarak işlenebilecek seviyelere düşürür. Düşürülen sinyal, sinyal şartlandırıcı karta ulaştırılır. Tasarlanan gerilim ölçme kartı Şekil 4.18.'de gösterilmektedir.



Şekil 4.18. Gerilim ölçme kartı.

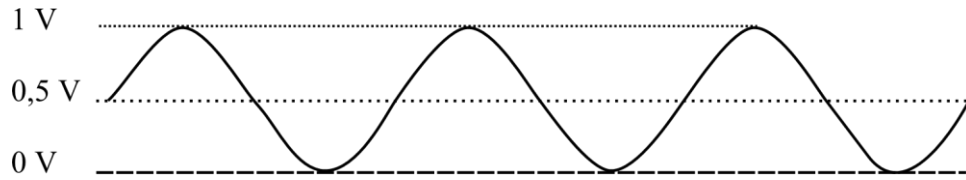
Gerilim sensörü olarak, ucuz olması, düzgün sinyal çıkışı ve piyasada yaygın kullanılması sebebiyle Aslan Trafo firmasının ASL 104106 model kapalı tip baskılı devre

trafoları kullanılmıştır. Trafonun kullanılabilirliğinin belirlenmesi için primer ve sekonder uçlarında voltaj dalga şeklinde bozulma, faz farkı ve saturasyon meydana getirmedeği test edilmiştir. Kullanılan pcb trafo Şekil 4.19.'da gösterilmektedir.



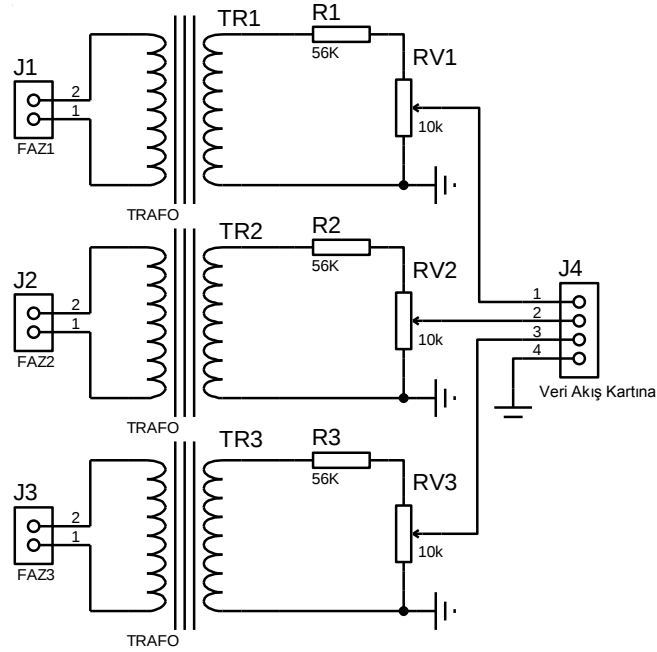
Şekil 4.19. Gerilim ölçme kartında kullanılan trafo (Aslan Elektronik, 2015).

Giriş gerilimi, trafo sekonder çıkışında bulunan gerilim bölücü devre ile 1 volta düşürülüp sinyal şartlandırıcı karta gönderilmektedir. Üç faz için üç ayrı gerilim trafosu ve gerilim bölücü devre kullanılmıştır. Bir faz için gerilim ölçme kartı sinyal çıkışı Şekil 4.20.'de gösterilmektedir.



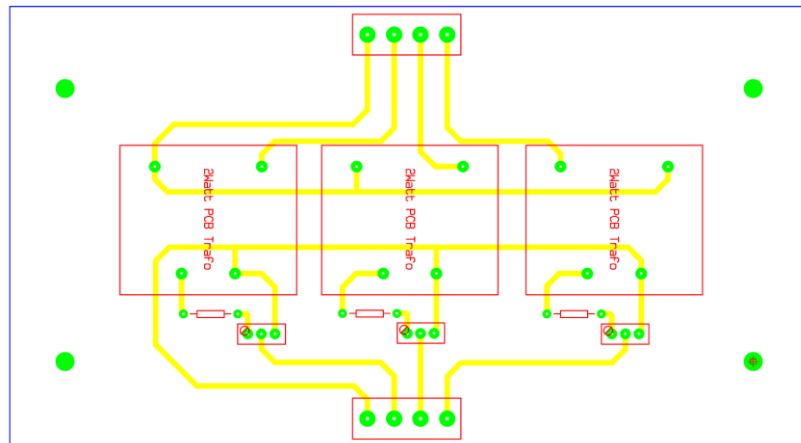
Şekil 4.20. Gerilim ölçme kartı sinyal çıkışı.

Üç fazlı bir sistem için tasarlanan gerilim ölçme kartı devre şeması Şekil 4.21.'de gösterilmektedir.



Şekil 4.21. Gerilim ölçme kartı devre şeması.

Gerilim bölücü kart için yapılan baskı devre tasarımı Şekil 4.22.'de gösterilmektedir.



Şekil 4.22. Gerilim ölçme kartı baskı devre şekli.

4.2.3. Piccolo controlstick

Tasarlanan sistemde Texas Instrument firmasının ürettiği F28069 Piccolo Controlstick ürünü kullanılmıştır. Ürünün öne çıkan özellikleri JTAG, USB desteği, üzerindeki çıkış uçlarıyla kontrol imkânı, örnek proje dosyalarıdır. Cihaz gerçek zamanlı işlemleri gerçekleştirebilecek şekilde tasarlanmıştır (Texas Instruments, 2009). Şekil 4.23.'de Piccolo controlstick ürünü gösterilmektedir.



Şekil 4.23. F28069 Piccolo Controlstick (Texas Instruments, 2009).

Piccolo Controlstick kartında TMS320F28069 32 bit gerçek zamanlı dijital sinyal işleyicisi (DSP) bulunmaktadır. Entegre 90Mhz hızında çalışabilir. 12 bit çözünürlüklü analog – dijital çevirici birimine sahiptir. ADC saniyede 3.46 milyon örnek alabilecek kapasitededir. TMS320F28069 entegresi Şekil 4.24.'de gösterilmektedir. Entegre üzerinde 16 ADC kanalı bulunur ve her kanalın kendine ait çevrim değerlerini tutan kaydedicileri vardır. Bu 16 kanal istenilen sıra ile ve ikili olarak çevrim yapacak şekilde ayarlanabilir.

Entegre üzerinde 5 Mbit hızında çalışabilen iki seri port ve bir adet USB port vardır. Kesmelere çok hızlı tepki verebilen bir kesme yapılanmasına sahiptir (Texas Instruments, 2010).



Şekil 4.24. TMS320F28069 DSP (Texas Instruments, 2010).

Piccolo Controlstick kartı üzerindeki TMS320F28069 dijital sinyal işleyicisi (DSP); kart üzerindeki USB portundan programlanabildiği gibi aynı zamanda veri alışverişi yapılmasına da izin verir. Cihazın desteklediği Code Composer Studio programı sayesinde hata

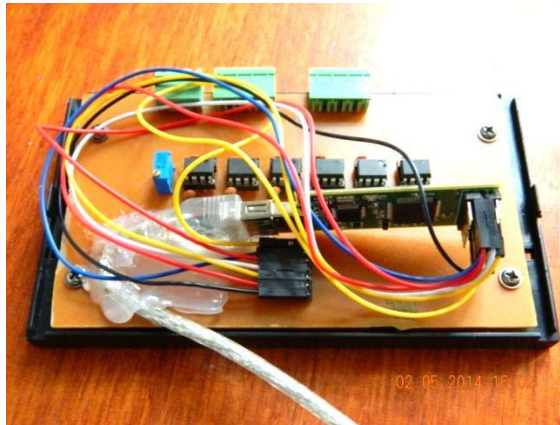
ayıklama işlemleri gerçekleştirilebilir. Küçük boyutu, uygun fiyatı ve güçlü donanımsal özellikleri tercihte öne çıkan unsurlardır.

Çizelge 4.1 TMS320F28069 DSP özellikleri.

İşlemci Hızı	90 Mhz
Flash Bellek	256KByte
Ram Bellek	100KByte
ADC	12 Bit, 16 Kanal, 3.46 MSPS
Seri İletişim	2 adet Seri Port, 1 adet USB Port, 1 adet I2C, 2 kanal SPI, 1 kanal McBSP, 1 kanal eCAN

4.2.4. Sinyal şartlandırıcı kartı

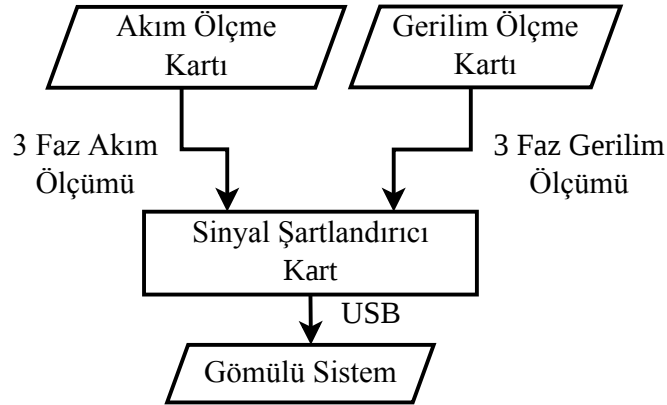
Sinyal şartlandırıcı kart, akım ölçme kartı ve gerilim ölçme kartından gelen bilgileri alıp dijital sinyal işleyicinin işleyebileceği hale getirir. Tasarımda kullanılan sinyal şekillendirici kart Şekil 4.25.'de gösterilmektedir.



Şekil 4.25 Sinyal şartlandırıcı kart.

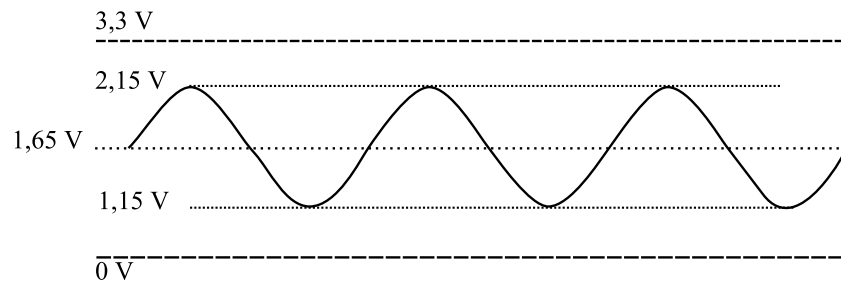
Gömülü sistem tarafında çalışan yazılım, şebekeden gelen akım ve gerilim bilgilerini doğrudan işlemlerde kullanamaz. Bu hesaplamaları gerçekleştirebilmek için akım ve gerilim sensörlerinden gelen bilgiler dijital hale getirilip dc ofset eklenerek gömülü sisteme aktarılmalıdır. Şekil 4.26.'da sistem donanımlarının blok diyagramları görülmektedir. Üç faz

şebekeye bağlı sensörlerden gelen üç kanal gerilim üç kanal akım bilgisi, sinyal şartlandırıcı kartta gömülü sistem tarafında çalışan yazılım için uygun hale getirilerek gömülü sisteme aktarımı sağlanmaktadır. Toplamda altı kanaldan alınan bilgiler, sinyal şartlandırıcıda dijital hale getirilir ve en son olarak USB arabirim ile gömülü sisteme aktarılır. Bir sonraki çevrim için aynı işlemler tekrar edilir.



Şekil 4.26. Sistem genel yapısı blok diyagramı.

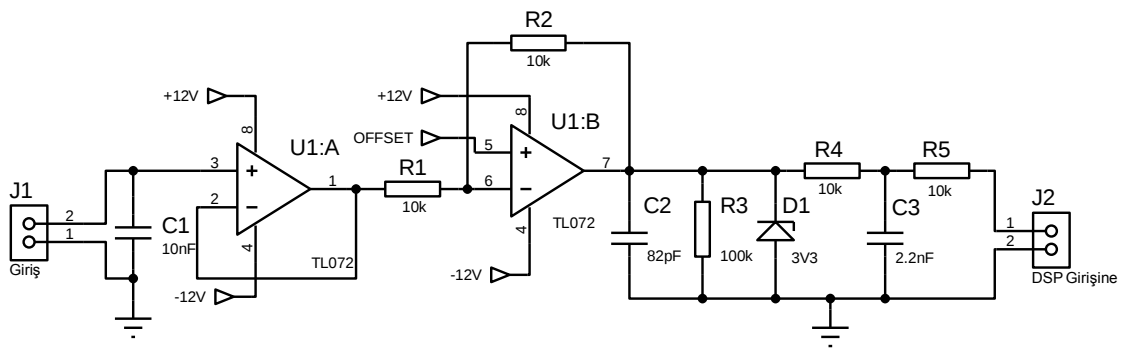
Bu çalışmada kullanılan dijital sinyal işleyici analog-dijital girişleri 0V ile en fazla 3.3V aralığında ölçüm yapabilmektedir. Sensör kartlarından alınan hem pozitif ve hem de negatif akım ve gerilim bilgilerinin ölçülebilmesi için şartlandırma devresine gelen sinyale ölçüm aralığının tam yarı değeri olan 1,65 voltluk dc ofset eklenir. Ofset gerilimi gömülü sistem tarafından sıfır volt olarak kabul edilecektir. Doğru ölçüm yapılabilmesi için gereken sınır değerler akım sinyal şekli Şekil 4.27.'de gösterilmektedir.



Şekil 4.27. Sinyal şartlandırıcı gerilim çıkış şekli.

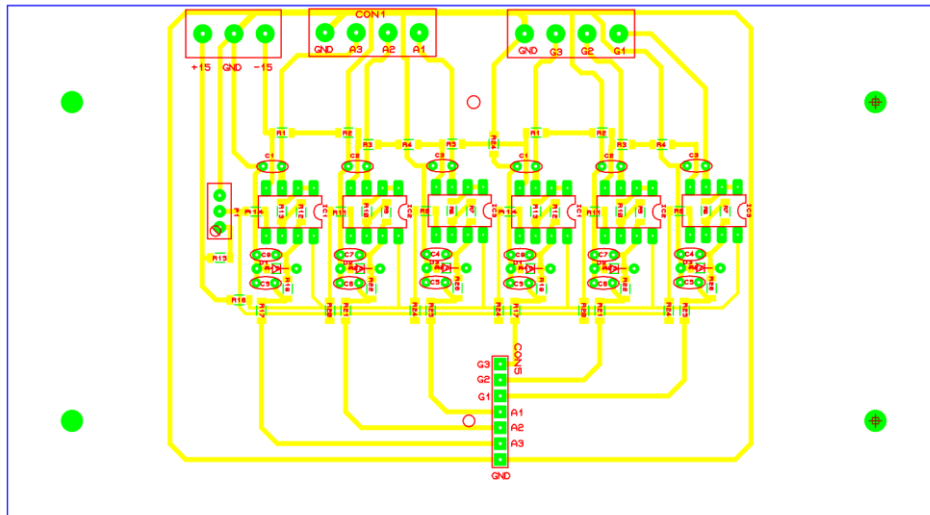
Sinyal şartlandırıcı tek kanal için devre şeması Şekil 4.28.'de gösterilmektedir. Sinyal şartlandırıcıya gelen sinyal, tampon görevi gören birinci işlemsel yükseltece ulaşmaktadır. İkinci işlemsel yükselteçten geçen sinyal, dc ofset gerilimiyle 1,65 volt seviyesine getirilir. Dc

offset gerilimi gömülü sistem tarafından sıfır volt olarak kabul edilecek ve bu şekilde ölçülen akım ve gerilim değerlerinin negatif kısımları da doğru bir şekilde ölçülebilecektir. Sinyal şartlandırıcı karttan gelen sinyallerin 3,3 voltu aşarak dijital sinyal işleyicinin girişlerini bozmaması için 3,3 voltluk zener diyor konulmuştur. Eğer gelen sinyal 3,3 voltu geçerse zener diyor devreye girerek sensörlerden gelen gerilimin 3,3 voltta sabit kalmasını sağlayacaktır. Son olarak anti-alias filtresinden geçen sinyal artık dijital sinyal işleyicinin girişlerine uygulanabilir. Anti-alias filtresi örnekleme düzgün yapılmasını sağlar.



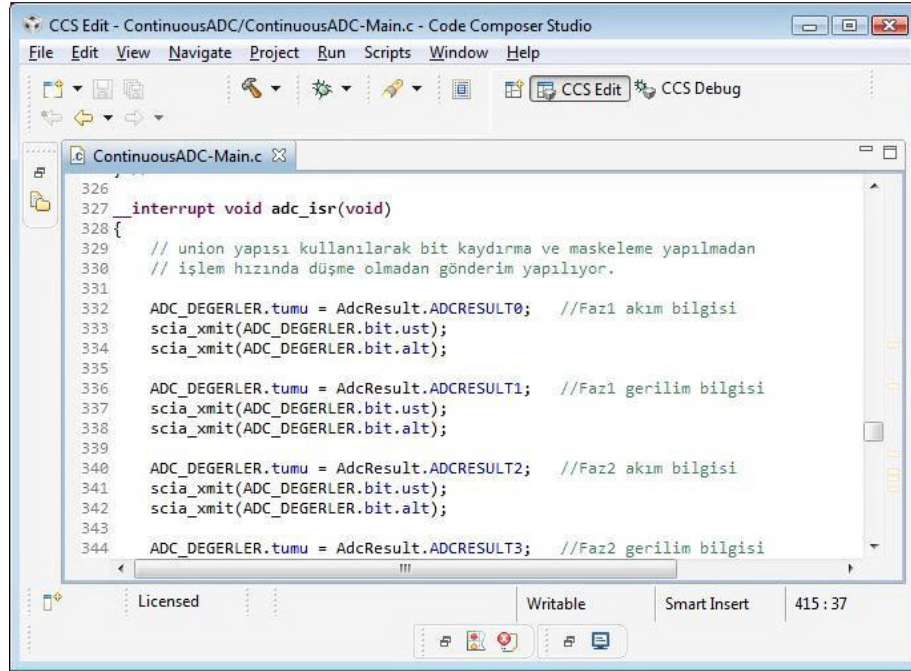
Şekil 4.28. Tek kanal için sinyal şartlandırıcı devre şekli.

Sinyal şartlandırıcı kart için tasarlanan baskı devre Şekil 4.29.'da gösterilmektedir.



Şekil 4.29. Sinyal şartlandırıcı baskı devre şekli.

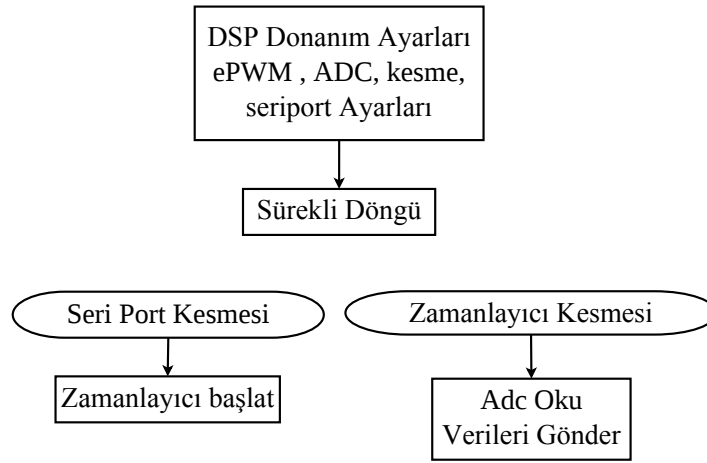
Sinyal şartlandırıcı için seçilen dijital sinyal işleyici üreticisi Texas Instrument firması, Code Composer Studio isimli uygulama geliştirme programını ücretsiz olarak sağlamaktadır. Code Composer Studio programı ekran görüntüsü Şekil 4.30.'da gösterilmektedir.



Şekil 4.30. Code composer studio kod geliştirme ortamı.

Code Composer Studio programı; dijital sinyal işleyici için kod yazım editörü, hata ayıklama, yazılan kodları derleme ve işlemciye yükleme gibi özellikler sunmaktadır. Programla beraber Texas Instrument firmasından ücretsiz olarak edinilebilen Control Suite programı da kurulur. Control Suite programı; dijital sinyal işleyicinin kodlanması için gereken, derleme işlemi sırasında ihtiyaç duyacağı tüm yazılım kütüphaneleri ve örnek projeleri içermektedir.

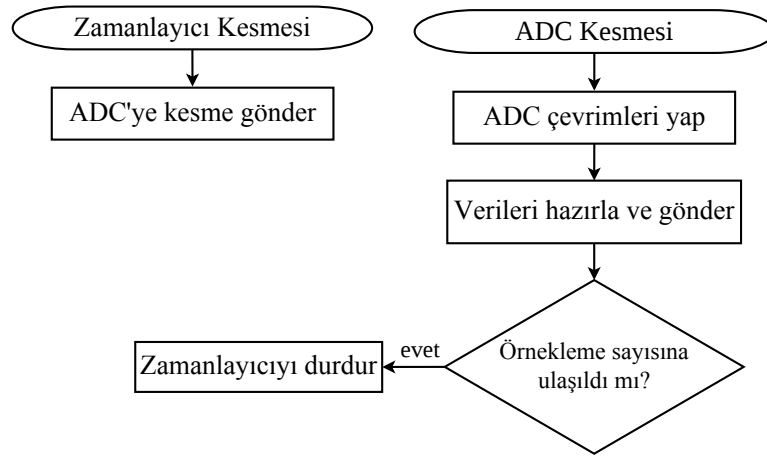
Şekil 4.31.'de dijital sinyal işleyici için tasarlanan yazılımın algoritması görülmektedir. Program başlangıcında önce donanımsal ayarlar yapılarak seri porttan bilgi gelmesi beklenmektedir. Seri porttan bilgi alındığında kontrol edilir. Gömülü sistem tarafından veri gönderim isteği gönderildiyse, DSP zamanlayıcı birimi çalıştırılır.



Şekil 4.31. DSP yazılım algoritması.

Zamanlayıcı birimi, sensörlerden örnek almak için gereken zamanlamanın yapıldığı kısımdır. Gömülü sistem tarafından veri gönderim isteği gerçekleştiğinde zamanlayıcı örnekleme frekansına göre ayarlanan değere kurulur. ADC zamanlayıcısı için sinyal işleyici entegrenin ePWM zamanlama özelliği kullanılır. Zamanlayıcı süresi dolduğunda analog-dijital çevirici için bir kesme üretir ve örnekleme gerçekleşir. Zamanlayıcı; on şebeke saykılı yani 200msn boyunca ayarlanan örnekleme frekansına uygun olarak kesme üretir ve bu kesmeler analog-dijital çeviriciye iletilerek çevrim yapılması sağlanır. ADC zamanlayıcı birimine yüklenen örnekleme zamanı değeri, şebekeden alınan anlık frekans bilgisine göre güncellenmektedir. Anlık frekans ile ilgili kullanılan yöntemler metod başlığı altında ayrıntılı olarak verilmiştir.

Kullanılan dijital sinyal işleyicideki analog dijital çevirici birimi 12 bit çözünürlüğe ve dâhili iki örnekleme-tutma birimine sahiptir. Aynı anda iki örnekleme birden gerçekleştirebilir. Bu özelliği ile aynı anda hem akım ve hem de gerilim bilgisi okunabilir. ADC birimi birbiri ardınca aynı anda iki kanaldan çevrim yapacak şekilde ayarlanmıştır. Önce faz1 akım ve gerilimi, daha sonra sırasıyla faz2 ve faz3 için akım ve gerilim değerleri ölçülerek kaydedilir. ADC ile ilgili yazılımsal akış diyagramı Şekil 4.32.'de gösterilmektedir.



Şekil 4.32. DSP analog-dijital çevrim diyagramı.

Sistem tasarımında kullanılan dijital sinyal işleyicide dahili olarak 2 adet seri port mevcuttur. Dijital sinyal işleyici üzerindeki seri portlardan biri entegrenin programlanmasını ve USB iletişimini sağlayan FT2232 Seri port - USB çevirici entegresine bağlıdır. Şekil 4.33.'de FT2232 entegresi gösterilmektedir. FT2232 yapısında bulunan JTAG bağlantıları ile kod geliştirme sırasında büyük kolaylıklar sağlamıştır. Gerçekleştirilen çalışmada gömülü sistem ile sinyal şartlandırıcı kart arasında iletişimi sağlayan USB bağlantılı seri port kullanılmıştır. Seri port 1,25 Mbit hızında çalışacak şekilde ayarlanmıştır. Bu iletişim hızı, şebekeden alınan akım ve gerilim örneklerini gömülü sisteme gerçek zamanlı olarak aktarmak için yeterlidir. Alınan örnekler 8 bit parçalara ayrılarak sırasıyla gönderilir.

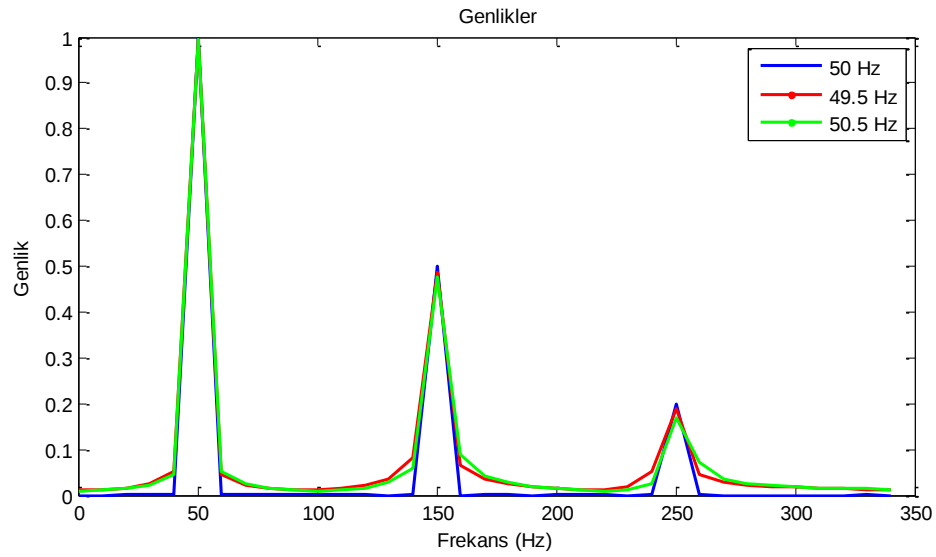


Şekil 4.33. Seri port – USB çevrim entegresi (Future Technology Devices International, 2011).

Seri iletişimin ve veri akışının başlayabilmesi için öncelikle gömülü sistemden çevrime başlama isteği gönderilir. Eğer başarılı olursa seri port kesmesi oluşur. Bu kesme zamanlayıcıyı aktif ederek örnekleme frekansında örnekleme yapılması için ADC birimini tetikler. Yapılan her çevrimde bilgiler işlemler için on sayıklık gruplar halinde hemen gömülü sisteme aktarılır.

4.3. Metod

Fourier dönüşümü ile harmonik analizi yapabilmek için analizi gerçekleştirilecek temel frekans değeri sisteme girilmelidir. Güç sisteminde harmonik analizi yapmak için temel frekans, şebeke frekansı olan 50 Hz dir. Teoride şebeke frekansı 50 Hz sabit olmalıdır ancak pratikte çeşitli sebeplerden dolayı şebeke frekansı sürekli 50Hz’de sabit kalmaz.



Şekil 4.34. Değişik frekanslara göre sabit 50Hz tabanlı Fourier dönüşümü sonuçları.

Şekil 4.34.’de 49,5 Hz, 50 Hz ve 50,5 Hz frekans değerlerine sahip üç ayrı sinyalin temel frekans 50 Hz için alınmış Fourier dönüşümü sonuçları görülmektedir. Şebeke frekansı sabit 50 Hz olduğu sürece yapılan hesaplamalarda herhangi bir hata oluşmaz. Şebeke frekansının olası değişimlerin sonucunda ara frekanslarda interharmonik olarak adlandırılan spektral sızıntılar görülmektedir. Bu sızıntılar, harmonik genliklerin doğru olarak ölçülememesine ve mevcut olmayan bileşenlerin görülmesine neden olmaktadır. İstenmeyen bu durumdan kaçınmak için FFT hesabında kullanılan temel harmonik frekans bileşenin şebeke üzerinden anlık olarak hatasız bir şekilde bulunması gerekmektedir. Bu şekilde harmonik analizi yapıldığında doğru sonuçlar elde edilecektir.

Literatürde şebekeden alınan bilgilerin harmonik içeriğini doğru hesaplamak için bazı yöntemler kullanılmaktadır. Bu yöntemlerden biri olan enterpolasyon metodunda; sabit örnekleme frekansında şebekeden örnekler alınır. Anlık şebeke frekansı ölçülür. Ölçülen şebeke frekansına göre önceden alınan örnekler, örnek azaltma ve enterpolasyon ile tekrar oluşturulur. Bu şekilde şebeke frekansında örnekleme elde edilmiş olur (Roscoe vd., 2011). ARTM

(adaptive realtime monitoring) yöntemi, sinyalin harmonik içeriğini doğru olarak hesaplamak için kullanılan bir diğer yöntemdir. Bu yöntemde, temel frekanstaki değişimler sürekli olarak PLL devreleri ile gözlenir. İlgilenilen harmonik frekans katsayıları sisteme girilir. DFT ile sadece ilgilenilen frekanslardaki harmoniklere ait bilgiler elde edilir (Antonesei, 2011). Goertzel algoritması frekans içeriğinin hesaplanmasında kullanılan bir diğer yöntemdir. Daha çok DTMF kodlarının çözülmesinde kullanılır. Goertzel algoritması kullanılarak şebekeden alınan bilgiler içerisinden sadece bir frekansa ait bilgiler elde edilir. Dolayısıyla birden fazla frekans için değer elde etmek istenirse, elde edilmek istenen frekans sayısı kadar algoritma işletilmelidir. Gerçek zamanlı çalışma için uygun değildir (Minciunescu ve Antonesei, 2011; Engelberg, 2008).

4.3.1. Değişken zamanlı örnekleme

$$F(\omega) = \int_{-\infty}^{+\infty} f(t) \cdot e^{-j\omega t} dt \quad (\omega = 2\pi \cdot f) \quad (30)$$

Fourier dönüşümü 30 no'lu denklem ile gösterilir. Sinyallerin harmonik analizi, temel frekans belirlenerek yapılır. Ülkemizdeki elektrik iletim sistemi için harmonik bileşenlerini incelemek istersek formüldeki ω ifadesi içindeki f için 50 değerinin verilmesi gerekmektedir. Teoride şebeke frekansı 50 Hz sabit olmalıdır ancak pratikte çeşitli sebeplerden dolayı şebeke frekansı tam 50Hz olmaz. Bu durumda Fourier dönüşümü sonucu yapılan harmonik analizi sonuçları hatalı olacaktır. Doğru sonuçlar elde edebilmek için şebeke frekansı tam olarak belirlenip analiz gerçek şebeke frekansına göre yapılmalıdır.

Bilgisayarlar, gömülü sistemler ve dijital sinyal işlemciler gibi elektronik elemanlarda bu işlemi gerçekleştirmek için sinyalin belirli sayıda veri için sonsuz tanımlı olan diziyi sınırlamamız ve sınırlı sayıda veri ile dönüşümü gerçekleştirmemiz gerekir. Bunun için Fourier dönüşümünün ayrık zamanlı şekli olan Ayrık Zamanlı Fourier Dönüşümü kullanılır. Ayrık zamanlı Fourier Dönüşümü

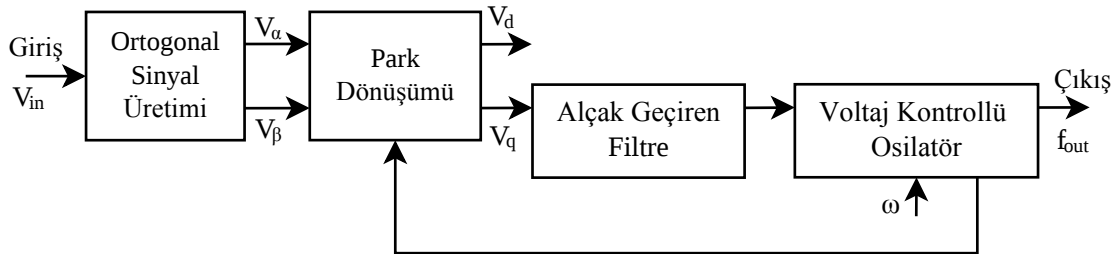
$$X(k) = \sum_{n=0}^{N-1} x(n) \cdot e^{-j\frac{2\pi nk}{N}} \quad ; \quad k = 0,1,2,\dots,N-1 \quad (31)$$

şeklinde tanımlanmıştır. Sürekli zamanlı Fourier dönüşümü için açısal frekans $\omega = 2\pi \cdot f$ iken ayrık zamanlı Fourier dönüşümü için açısal frekans ifadesi $\omega = \frac{2\pi \cdot n \cdot k}{N}$ olacaktır. Formülde N ifadesi dönüşümü alınan örnek sayısıdır. Ayrık zamanlı Fourier dönüşümünde analizi yapılacak frekans çözünürlüğü örnekleme sayısı ile doğrudan ilgilidir. Örnek sayısı ve örnekleme zamanı sabittir. Sürekli olarak örnekleme sayısının değişmesi hem kullanışsızdır ve hem de sürekli

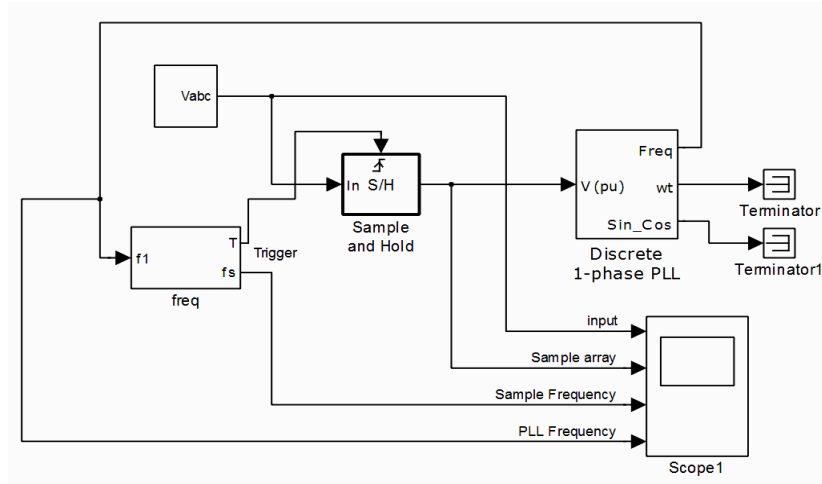
değişen örnek sayısından dolayı matematiksel işlem yükü artacaktır. Bunun yerine örnek sayısının sabit tutulup anlık frekansa göre örnekler arası sürenin ayarlanması daha iyi bir yaklaşımdır.

Gerçekleştirilen sistemde, temel bileşenin anlık frekans bilgisini bulabilmek için şebeke gerilimi kullanılır. Dijital sinyal işlemci ile örneklenen şebeke gerilimi, yazılımsal PLL (faz kilitlemeli döngü) algoritması kullanılarak anlık frekansın bilgisinin bulunması sağlanır. Şekil 4.35.'de uygulamada kullanılan PLL blok şeması gösterilmektedir. Frekansa bağlı olarak analog dijital çeviricinin örnekleme frekansı 10 periyotluk bilgi içerecek şekilde yeniden değiştirilir. Tasarlanan sistemin gerçek zamanlı çalışma gereksiniminden dolayı işlem yükü fazla olan yöntemler tercih edilmemiştir. Hazırlanan bu sistemle işlem hızı olarak rekabet edebilecek olan yöntem ARTM'dir. Fakat ARTM, sadece sınırlı sayıda frekanslardaki bilgilere ulaşılacağından uygun görülmemiştir. Gerçekleştirilen yazılım tasarımında faz kilitlemeli döngü (PLL) kullanılarak güç şebekesindeki anlık frekans bilgisi elde edilir. Elde edilen frekansa göre örnekleme süresi ayarlanarak zamanlayıcı bloğuna yüklenir. Zamanlayıcı bloğu ADC biriminin örnekleme yapmasını sağlar. Sonuç olarak, PLL biriminden elde edilen frekansa göre örnekleme süresi anlık olarak ayarlanmış olur.

Uygulamada SOGI tipi PLL kullanılmıştır. Bu işlemin doğru zamanlama ile gerçekleştirilebilmesi için işlemci kesme yapısı kullanılır. Her kesme geldiğinde şebekeden alınan gerilim örnekleri; işlemcinin verdiği floating point desteğiyle yazılımsal PLL bloğunda çok kısa sürede işlenir. PLL frekans bilgisi alındığında yeni frekansa göre örnekleme zamanları düzenlenir ve değişken zamanlı örneklemenin yapılacağı zamanlayıcı bloğuna yüklenir. Zamanlayıcı bloğu PLL den elde edilen sonuçlara göre örnekleme yapmış olur.

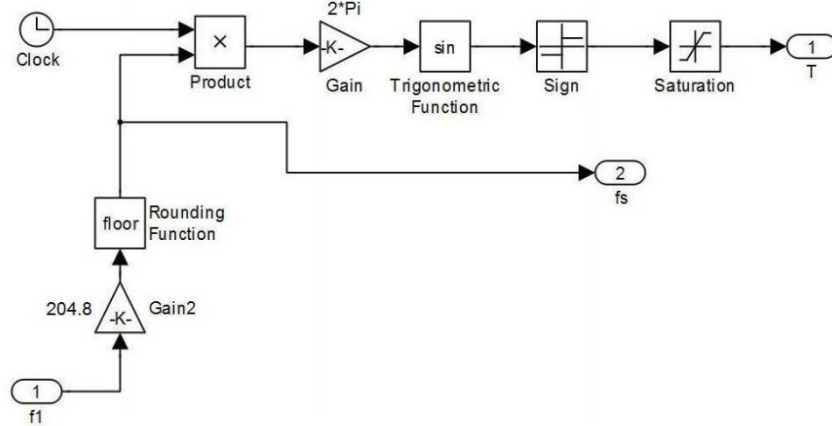


Şekil 4.35. Sogi tipi PLL blok diyagramı.



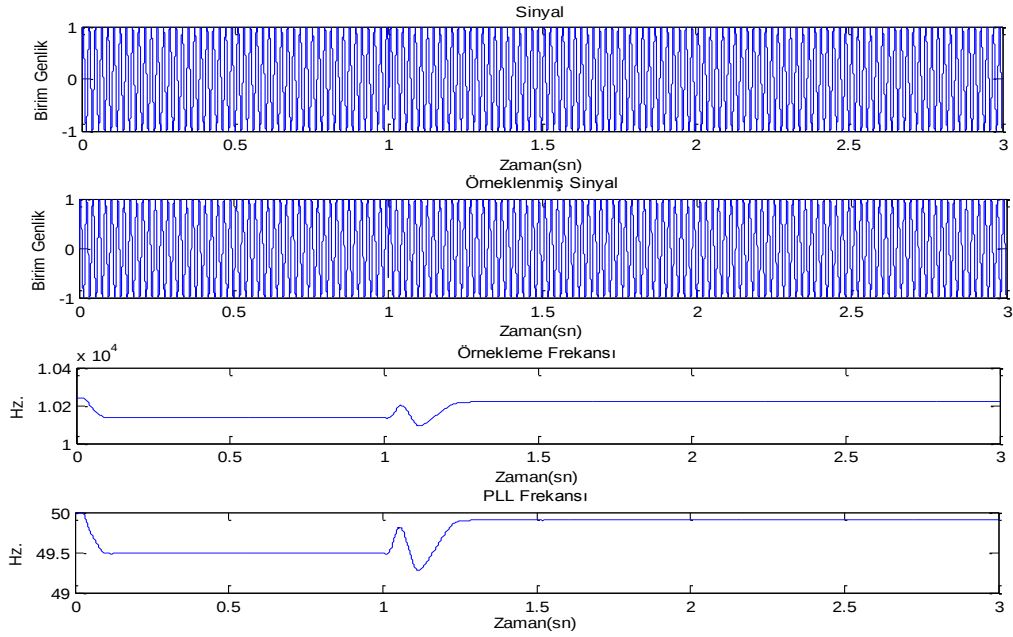
Şekil 4.36. Çevrim zamanlayıcısı Simulink blok diyagramı.

Sistem tasarlanırken Matlab programı içindeki Simulink platformunda simülasyonu oluşturulmuş, doğruluğu denenmiştir. Şekil 4.36.’da Simulink kullanılarak simülasyonu yapılan sistemin blok diyagramı gösterilmektedir. Üretilen sinyal “Sample and Hold” bloğunda örneklenerek PLL’e gönderilir. PLL bloğundan elde edilen frekans bilgisi freq bloğunda işlenerek yeni frekans değerine göre örnekleme zamanı ayarlanır. freq bloğunun iç yapısı Şekil 4.37.’de gösterilmektedir.



Şekil 4.37. Simulink freq blok diyagramı iç yapısı.

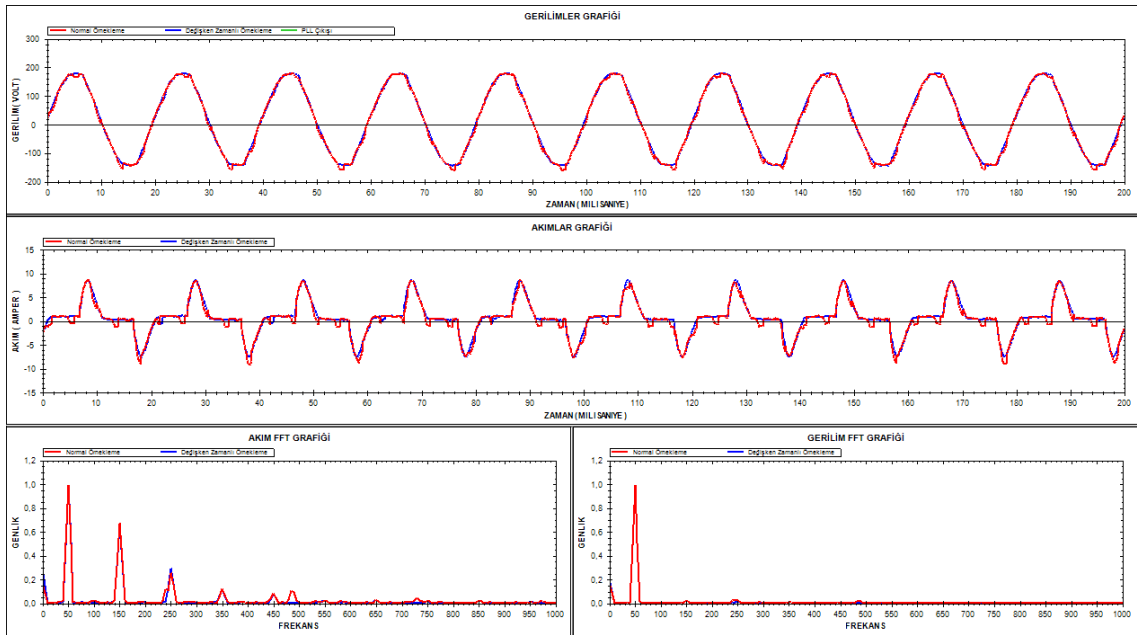
Yapılan simülasyon sonuçları Şekil 4.38.’de gösterilmektedir. PLL tarafından frekans değişimi algılandığında örnekleme frekansı uygun olarak hemen değiştirilmektedir. Bu yöntem kullanılarak şebeke frekansında görülen frekans değişimleri Fourier dönüşümü sonuçlarının doğru olmasını sağlar.



Şekil 4.38. Tasarlanan metod için simülasyon sonuçları.

4.3.2. Deneysel doğrulama

Deneysel ilk çalışmada yük olarak üzerinde hız kontrolü bulunan tek fazlı bir motor kullanılmıştır. Ölçümler boyunca motor hızı ve gücü sabit tutulmuştur. Yapılan uygulama sonucu Şekil 4.39.'da gösterilen veriler elde edilmiş ve kaydedilmiştir. Burada 10 periyotluk ölçüm için akım, gerilim sinyal şekilleri ve bunların harmonik analiz sonuçları görülmektedir. Şekil 4.39.'da kırmızı renkle gösterilen grafikler sabit frekanslı örnekleri temsil ederken mavi renkle gösterilen grafikler anlık frekans değişimine uygun olarak yapılan örnekleme temsili eder. Şebekeden alınan veriler, aynı yük ve aynı zamanda ölçülen hem değişken zamanlı örnekleme ve hem de sabit zamanlı örnekleme yöntemleri kullanılarak işlenmiş ve sonuçlar Çizelge 4.2.'de gösterilmiştir. Çizelge 4.2. incelendiğinde 50Hz' e göre sabit olarak alınan dönüşümlerin sonuçlarının değişken zamanlı örnekleme yöntemine göre elde edilen sonuçlardan az da olsa farklı olduğu görülecektir.



Şekil 4.39. Gömülü sistem yazılımı ekran görüntüsü. (Kırmızı çizgi sabit zamanlı örnekleme, mavi çizgi değişken zamanlı örnekleme.)

Çizelge 4.2. Sabit yük altında uygulama sonuçları.

Ölçülen frekans (On periyot ortalaması)	Sabit zamanlı örnekleme ile Gerilim THD	Sabit zamanlı örnekleme ile Akım THD	Değişken Zamanlı Örnekleme ile Gerilim THD	Değişken Zamanlı Örnekleme ile Akım THD
49,95264	4,3375	86,8785	4,1272	86,5408
49,95606	4,1789	86,3909	4,1016	86,2148
49,95752	3,9917	86,0629	4,1354	86,4279
49,96631	4,0552	86,1607	4,0947	86,4359
49,96729	4,1217	86,5153	4,1077	86,7062
49,97998	4,1758	86,8666	4,0476	86,6191
Toplam	74,076	1451,914	73,6858	1453,492

Çizelge 4.2.'de gösterilen değerler incelendiğinde, anlık şebeke frekansında görülen değişime bağlı olarak sabit zamanlı örnekleme ve değişken zamanlı örnekleme ölçüm sonuçları arasında farklar görülecektir. Bu farklar şebeke frekansında görülen değişime bağlı olarak artacak veya azalacaktır.

Yapılan ikinci deneysel çalışmada yapısında kontrollü doğrultucular bulunan üç fazlı yük kullanılmıştır. Kurulan deney düzeneği Şekil 4.40.'da gösterilmektedir.



Şekil 4.40. Üç fazlı yük için kurulan deney düzeneği.

Üç fazlı yük kullanılarak yapılan ölçüm sonuçlarını kıyaslamak için Fluke43B güç kalitesi analizörü kullanılmıştır. Fluke43B ve tasarlanan sistem ile yapılan ölçüm sonuçları tablo halinde Çizelge 4.3.'de gösterilmektedir.

Çizelge 4.3. Fluke43B ve tasarlanan sistemin ölçüm sonuçları.

	Fluke 43B			Tasarlanan Cihaz		
Ölçümler	Faz1	Faz2	Faz3	Faz1	Faz2	Faz3
RMS Akım	11,48 A	11,24 A	11,2 A	11,26 A	11,05 A	11,02A
RMS Gerilim	225,2 V	225,8 V	225,5 V	222,4 V	219,8 V	223 V
THD I	28,40%	28,80%	28,80%	27,30%	27,96%	28,06%
THD V	1,00%	1,50%	1,20%	2,80%	3,44%	3,57%
Cos Φ	1	1	1	0,97	0,97	0,97
Güç Faktörü	0,95	0,95	0,95	0,93	0,93	0,93

Çizelge 4.4.'de Fluke43B ve tasarlanan sistem ölçümleri için 5.,7.,11. ve 13. harmonik genlikleri ölçüm sonuçları gösterilmektedir. Ölçümler bir faza ait akım harmoniği genlikleridir.

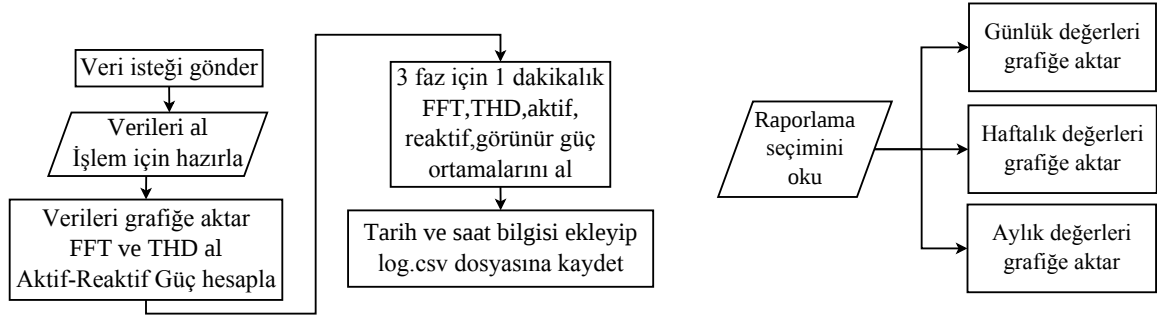
Çizelge 4.4. Fluke43B ve tasarlanan sistemin harmonik genlikleri ölçüm sonuçları.

Harmonik Genlikleri	Fluke 43B	Tasarlanan Sistem
5. Harmonik	23%	23%
7. Harmonik	10%	10%
11. Harmonik	8%	8%
13. Harmonik	%5.5	5%

Çizelge 4.3.'deki sonuçlara incelendiğinde tasarlanan sistem ve Fluke43B ile yapılan ölçümlerde farklılıklar görülecektir. Her iki sistemle yapılan THD ölçümleri incelendiğinde, akım THD değerlerindeki farklılığın %1'den az gerilim THD değerlerindeki farklılığın ise %2.3'lere geldiği görülecektir. Gerilim THD seviyelerindeki bu farklılığın sebebi, gerilim sensörü olarak kullanılan trafoların manyetik doyumu sebebiyle ölçüm sonuçlarını etkilemesi gösterilebilir. Gerilim dalga şekillerinde gerçekleşen bu etki güç faktörü ve $\cos \Phi$ değerlerini de etkilemektedir. Gerilim THD seviyelerindeki bu farklılığı azaltmak için, hall etkili sensörlere alternatif olarak HCPL kodlu opto-amplifikatörler daha uygun maliyetli bir çözüm olabilirler. Bu opto-amplifikatörlerin maliyeti trafo kullanımına göre yüksek olacaktır.

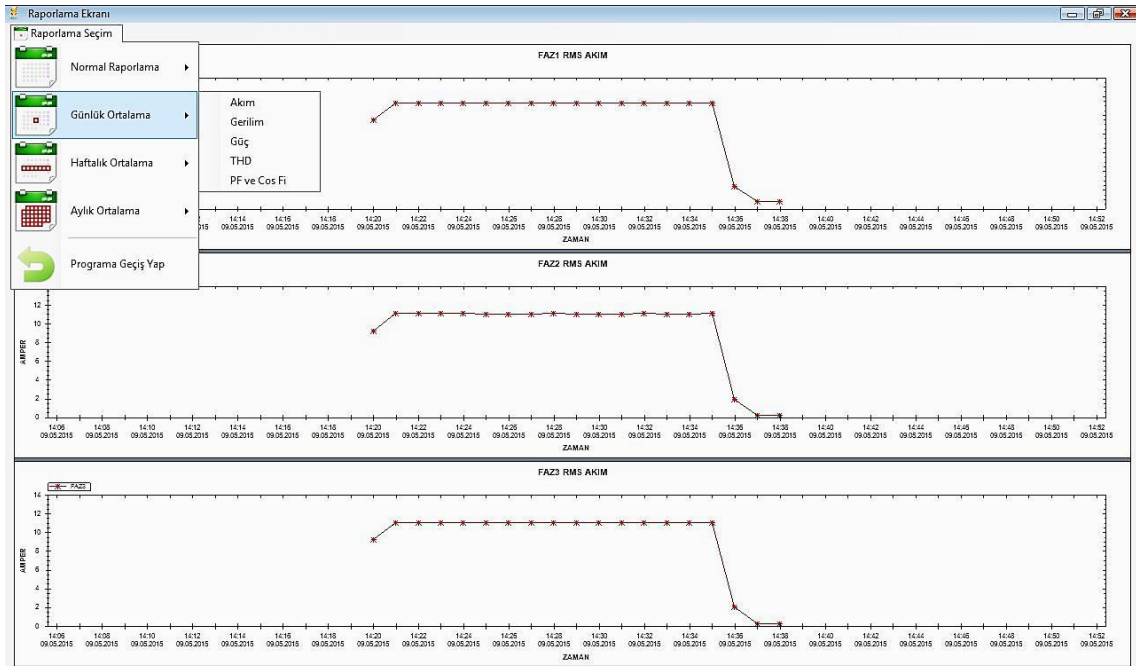
4.4. Dosya ve raporlama işlemleri

Gömülü sistem tarafından alınan akım ve gerilim bilgileri kullanılarak her bir faz için ayrı ayrı aktif, reaktif güç FFT ve THD hesaplamaları yapılır. Uzun dönemli haftalık veya aylık raporların üretilmesi için hesaplanan değerlerin bir dakikalık ortalamaları zaman bilgisiyle beraber kaydedilir. Kayıt dosyası formatı, endüstride standartlaşmış, Linux ve Windows tabanlı birçok program tarafından işlenebilen CSV formatıdır. Standartlara göre dosya formatına kayıt sınırı konmamıştır dolayısıyla sistemdeki disk kapasitesi kadar kayıt yapılabilir (Shafranovich, 2005). Raporlama için menüden raporlama ekranı kullanılır. İstenen tarih aralığında günler, haftalar ve aylar bazında raporlama yaptırılabilir. Raporlama ekranında yakınlaştırma yapılarak saatlik hatta dakikalık olarak da sonuçlar görülebilir. Gömülü sisteme ait yazılımının algoritması Şekil 4.41.'de gösterilmektedir.



Şekil 4.41. Raporlama ekranı algoritması.

Tasarlanan raporlama yazılımına ait ekran görüntüsü Şekil 4.42 ve Şekil 4.43’de gösterilmektedir.



Şekil 4.42. Raporlama ekranı.



Şekil 4.43. Raporlama ekranı.

5. SONUÇLAR

Gerçekleştirilen bu tez çalışmasında anlık akım ve gerilim değerlerinin ölçümünü yapan, harmonik analizlerini ve reaktif güç hesaplamalarını gerçekleştirebilen, gerektiğinde uzun dönemli raporlandıran, düşük maliyetli bir izleme sistemi tasarlanmıştır. Harmonik analizinin daha doğru sonuçlar vermesi için örnekleme frekansının anlık şebeke frekansına göre ayarlanabildiği PLL tabanlı bir algoritma geliştirilmiştir. Sistem bileşenleri olarak TMS320F28069 gerçek zamanlı dijital sinyal işleyici, Windows7 embedded gömülü işletim sistemi, mini-itx pc anakartı kullanılmıştır. Sistem tasarımı kullanılarak DSP ile gerçek zamanlı ve gecikme olmadan analog-dijital çevrim işlemleri gerçekleştirilmiştir. Bu sayede gerçek zamanlı işlem yaptırılabilmesi için gereken altyapı oluşturulmuştur. Gömülü sistem yazılımı için yazılım lisansı gerektirmeyen C# programı kullanılmıştır. Tasarımda .NET kütüphaneleri kullanıldığından birçok işletim sisteminde lisans ücreti gerektirmeden ve oldukça hızlı bir şekilde çalışacak yazılım tasarımı yapılmıştır. Gömülü sistem yazılımında sınıf yapısı kullanılmıştır. Bu yapının en büyük avantajı modüler olmasıdır. Eğer yapılan yazılıma yeni bir işlev yaptırılmak isteniyorsa ilgili işlev sınıf dosyasının projeye eklenmesi ve gerekli yerlerde yazılım çağrısı yapılması yeterlidir. Deneysel çalışma sonuçları çalışmada sunulmuştur. Fluke43B ve tasarlanan sistem ölçüm sonuçları incelendiğinde ölçüm sonuçlarının birbirlerine oldukça yakın olduğu görülecektir. Gerilim THD sonuçlarındaki farklar ise gerilim sensörü olarak kullanılan trafolarda manyetik doyum sebebiyle oluşmaktadır. Gerilim dalga şekillerinde gerçekleşen bu etki güç faktörü ve $\cos \Phi$ değerlerini de etkilemektedir. Maliyet biraz daha artırılarak endüstriyel gerilim sensörleri kullanıldığında piyasada kullanılan güç kalitesi analizörü ölçümü doğruluğunda ölçümler yapılabilir.

KAYNAKLAR DİZİNİ

Akagi H., Kanazawa Y. ,Nabae A., (1983), Generalized Theory of the Instantaneous Reactive Power in Three-Phase Circuits, Proc. of IEEJ International Power Electronics., s. 293-297.

Antonesei G., (2011), Adaptive Real-Time DSP Architecture to Monitor Harmonic Components and Various Power Quality Factors in Electric Power Grids By Gabriel Antonesei, Analog Devices Inc, USA.

Arrillaga J., Watson N. R., (2003), Power System Harmonics, John Wiley & Sons Ltd., England.

Aslan Elektronik, (2015), ASL104106 Datasheet, Türkiye.

Atalik. T., Cadirci, I., Demirci T., Ermis, M. . Inan T., Kalaycioglu A.S., Salor, O., (2014), Multipurpose Platform for Power System Monitoring and Analysis With Sample Grid Applications, IEEE Transactions On Instrumentation and Measurment Cilt:63 Sayı:3, s.566-582.

Axelson J., (2009), USB Complete : The Developer's Guide, Independent Publishers Group, USA.

Bayliss C.R., Hardy B.J., (2011), Power Quality – Harmonics In Power Systems Transmission and Distribution Electrical Engineering Fourth Edition,Newnes(Elsevier Inc), USA.

Blanco A. M., Stiegler J. M., (2013), Power Quality Disturbances Caused by Modern Lightnig Equipment, IEEE Grenoble PowerTech, s.1-6.

Chang G.W., (2008), Measuring Power System Harmonics and Interharmonics By An Improved Fast Fourier Transform-Based Algorithm , IET Generation Transmission Distribution Cilt:2 Sayı:3, s.193-201.

Chang G.W., Chen C.I., (2010), Measurement Techniques For Stationary and Time-Varying Harmonics, IEEE Power and Energy Society General Meeting, s.1-5.

Chapman D., (2001a), Introduction To Power Quality, Copper Development Association, United Kingdom.

Chapman D., (2001b), Harmonics Causes and Effects, Copper Development Association, United Kingdom.

Chappro L., (2015), Signals and Systems Using MATLAB (Second Edition), Academic Press, USA.

Chu E.T. H., Fang C.W., (2015), CALEE: A Computer-Assisted Learning System For Embedded OS Laboratory Exercises, Elsevier Computers & Education Cilt 84, s. 36–48

Cooley J. W., Tukey J. W., (1969), An Algorithm For The Machine Calculation Of Complex Fourier Series, IEEE Papers on Digital Signal Processing; 1969, s.146-150.

KAYNAKLAR DİZİNİ(devam)

Dugan R.C, McGranaghan M.F. , Beaty H.,W., (1996), Electrical Power Systems Quality, McGraw-Hill Professional 2. Edition, USA.

Dobrucky B., Kim H., Racek V., Roch M., Pokorny M., (2002), Single-Phase Power Active Filter and Compensator Using Instantaneous Reactive Power Method, IEEE Power Conversion Conference Cilt:1, Osaka, s.167-171.

Engelberg S., (2008), Digital Signal Processing: An Experimental Approach, Springer Science & Business Media B.V., London.

Future Technology Devices International, (2011), FT2232C Dual USB UART / FIFO I.C. Datasheet, Scotland.

Gök A., Bilmez A., (2011), Güç Kalitesi Milli Projesi ve Sonuçları, Teiaş Genel Müdürlüğü, Ankara.

Hamblen J. O., (2007), A Tutorial Approach With Laboratory Exercises For Windows Embedded CE 6.0 and the eBox 2300 SoC Embedded Computer, School Of Electrical and Computer Engineering Georgia Institute Of Technology,USA.

Haque M.T. , Ise T., (2002), Implementation Of Single-Phase PQ Theory , Power Conversion Conference PCC Osaka, Cilt:2, s.761-765.

Heath M. T., (2002), Fast Fourier Transform, Scientific Computing: An Introductory Survey,Lecture Notes, Department of Computer Science University of Illinois at Urbana, Champaign.

Heath S., (2003), Embedded System Design Second Edition, Newnes (Elsevier Inc), USA.

Huang X., Huang Y., Li L, (2011), Design Of The Power Quality Monitoring System Of Harmonic Based On DSP, International Conference On Electronics, Communications and Control (ICECC), s.1428-1431.

Hui Z., Tang W.,You-Jun Y., (2010), Simulations On Harmonic Analysis Of Arc Furnace, Information Engineering and Computer Science (ICIECS), 2nd International Conference On. 2010, s. 1-4.

Intel Corporation, (2011), Product Brief Intel Desktop Board D2500hn Innovation Series, USA.

IEEE519:1992., (1992), ANSI/IEEE 519:1992.

Jamil M., Hashmi, G.M., Syed, Z.A., Awais, Q., (2007), Harmonics In Adjustable Speed Drives: Causes, Effects, and Solutions, International Conference On Information and Emerging Technologies ICIET 2007, s. 1-6.

Kamal R., (2008), Embedded Systems - Architecture, Programming And Design, McGraw-Hill , New York, USA.

KAYNAKLAR DİZİNİ (devam)

- Katznelson Y., (2004), An Introduction To Harmonic Analysis, Cambridge Univ. Press, UK.
- Kaya A., (2008), Gömülü Sistemler ve Uygulama Alanları , Çanakkale Onsekiz Mart Üniversitesi Akademik Bilişim Konferansı 2008, s. 677-683.
- Kester V., (2003), Mixed-Signal And DSP Design Techniques, Analog Devices Series, Amsterdam.
- Kennedy B. W., (2000), Power Quality Primer, McGraw-Hill, New York.
- Lem, (2015), LA100-P Current Transducer Datasheet, Switzerland.
- Li J., Li W., Hu Y., (2010), The Design And Implement Of Incorporate System For Power Harmonic Detection and Compensation Based On Embedded Platform , International Conference On Mechanic Automation and Control Engineering (MACE), s.2516-2519.
- Lin H. C., (2005), Remote Real-time Power System Harmonics Detection And Monitoring Via The Internet , 2005 IEEE International Conference on Systems, Man and Cybernetics Cilt:3, s.2867-2872.
- Lu P., Su S., Liu G., Rong H., Zheng L., (2008), A New Power Quality Detection Device Based on Embedded Technique , IEEE Third International Conference On Electric Utility Deregulation And Restructuring And Power Technologies., s.1635-1640.
- Microsoft Corporation, (2010), Windows Embedded Standart 7 Technical Overview Datasheet, USA.
- Microsoft Corporation, (2011), Windows Embedded Standart 7 Datasheet, USA.
- Minciunescu P., Antonesei G., (2011), Novel Harmonic Analysis Method Improves Accuracy, Reduces Computation Overhead in Smart meters by Petre Minciunescu and Gabriel Antonesei, Analog Devices Inc, Scotland.
- Mok Y. K., Tse C.F.N., Lau W.H., (2012), A Study On The Effects Of Voltage Distortion On Current Harmonics Generated By Modern SMPS Driven Home Appliances in Smart Grid Network, 9th IET International Conference on Advances in Power System Control, Operation and Management (APSCOM 2012), s. 1-6.
- Mueller J., (2011), Learn Microsoft® Visual C#® 2010, Microsoft Press, USA.
- National Instrument, (2015), Understanding FFTs and Windowing White Paper, USA.
- Park T. H., (2010), Introduction To Digital Signal Processing : Computer Musically Speaking, World Scientific Publishing Company, Singapore.
- Rahman M. A., (2012), Understanding Mono C# Compiler, CreateSpace Independent Publishing Platform, E-book.

KAYNAKLAR DİZİNİ (devam)

Rechka S., Ngandui E., Xu J., Sicard P., (2002), A Comparative Study of Harmonic Detection Algorithms For Active Filters and Hybrid Active Filters. IEEE Power Electronics Specialists Conference 1, s.357–363.

Rosa F. C. D. L., (2006), Harmonics And Power Systems, Boca Raton : Taylor & Francis Group, Hazelwood, Missouri, USA.

Roscoe A.J., Carter R., Cruden A., Burt G.M., (2011), Fast-Responding Measurements Of Power System Harmonics Using Discrete and Fast Fourier Transforms With Low Spectral Leakage, International Conference On Renewable Power Generation, Edinburg, s.1-6.

Shafranovich Y., (2005), Common Format and MIME Type For Comma-Separated Values (CSV) Files, SolidMatrix Technologies Inc.

Shareghi M., Phung B.T., Naderi M.S., Blackburn T.R., Ambikairajah E., (2012), Effects Of Current and Voltage Harmonics On Distribution Transformer Losses ,IEEE International Conference On Condition Monitoring and Diagnosis, Indonesia, s. 633- 636.

Singh G. K., (2009), Power System Harmonics Research: A Survey, India European Transactions On Electrical Power, s.151-172.

Singh R., Singh A., (2010), Aging Of Distribution Transformers Due To Harmonics ,IEEE 14th International Conference On Harmonics and Quality Of Power (ICHQP)., s. 1-8.

Sinha C., (2015), C# & the .Net Framework - Quick Reference Guide, E-book.

Smith S. W., (1998),The Scientist and Engineer's Guide to Digital Signal Processing, California Technical Publishing.

Tan L., (2008), Digital Signal Processing : Fundamentals And Applications, Academic Press, Amsterdam.

Texas Instruments, (2009), TDMX320F28069USB Control Stick Datasheet, USA.

Texas Instruments, (2010), TMS320F28069 Datasheet, USA.

Todd M., (2008), Fast Fouries Transform Algorithms With Applications, PhD. Thesis, Clemson University.

Troelsen A., (2010), Pro C# 2010 and the .NET 4 Platform, Apress, E-Book.

Vardar K., Akpınar E., Sürgevil T., (2009), Evaluation Of Reference Current Extraction Methods For DSP Implementation In Active Power Filters., Elsevier Science Electric Power Systems Research Cilt:79 Sayı:10, s. 1342–1352.

Wakileh G. J., (2003), Harmonics In Rotating Machines, Elsevier Science Electric Power Systems Research Cilt:66 Sayı:1, s. 31-37.

KAYNAKLAR DİZİNİ(devam)

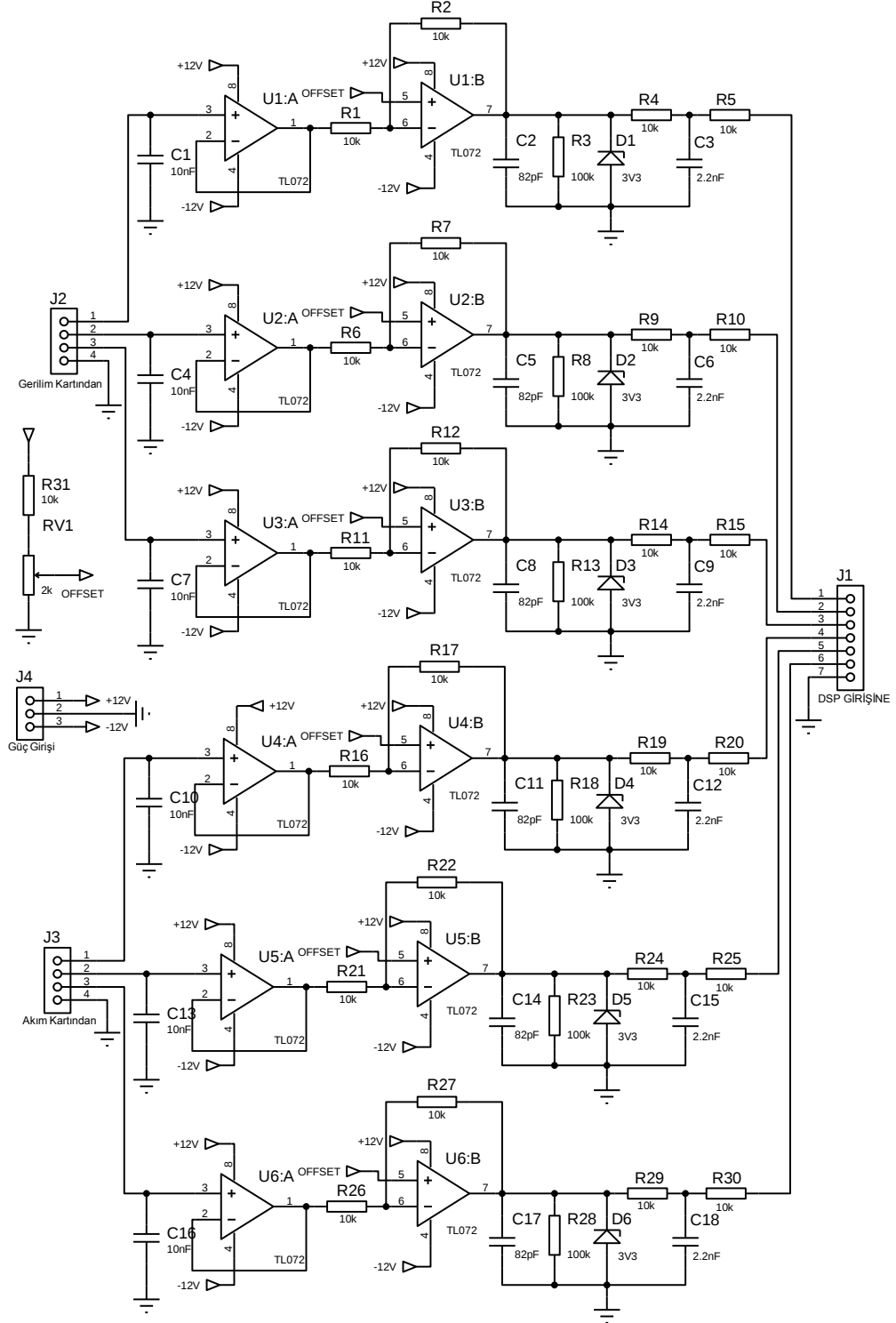
Wakileh G. J., (2011), Power System Harmonics: Fundamentals, Analysis, and Filter Design, Springer-Verlag, Berlin.

Wang C. L., Yao B., Yang Y.,Zhu Z., (2001), A Survey of Embedded Operating System, Technical Project Report, University of California.

Zhu J., Xie D.,Wang F., Cao T., (2012), Study On the Harmonics Of Switching Power Supply ,2012 International Workshop on Information and Electronics Engineering (IWIEE) Cilt:29, s. 2098-2102.

EKLER

Ek. 1. Sinyal Şartlandırıcı Kartı Tasarımı



Ek. 2. FFT ve THD için C# Programlama Dilinde Oluşturulan Sınıf Yapısı

```
using System;

namespace Tez_Uygulama{
    public class FFT    {
        public double[] gercekSayi; // Sayı gerçel kısmı
        public double[] sanalSayi; // Sayı Sanal kısmı
        public double THD; // Hesaplanacak THD değeri
        const byte bitSayisi = 11; // FFT kademe sayısı
        // Frekans tablosundan temel frekansın görüleceği dizi eleman indisi
        const byte temelFrekans = 10;
        const ushort ornekSayisi = 2048; // Toplamda FFT örnek sayısı
        public FFT(){
            // FFT hesabında kullanılacak değişken dizileri oluşturuluyor.
            gercekSayi = new double[ornekSayisi];
            sanalSayi = new double[ornekSayisi];
        }
        public void FFT_AL(double[] veri){
            const byte ThdSayisi = 20; double geciciX, geciciY;
            int i, i2, j, k, n; int islemSayi, basamakAtlama; double sinus, kosinus, reel, sanal;
            for (i = 0; i < ornekSayisi; i++){ // hannig çerçeveleme yapılıyor.
                if (i < veri.Length){
                    // Hanning window uygulanıyor.  $0.5 * (1 - \cos(2 * \pi * n / N - 1))$ 
                    double hanning = 0.5 * (1 - Math.Cos(2 * 3.14159 * i / (ornekSayisi - 1)));
                    gercekSayi[i] = veri[i] * hanning;
                } else gercekSayi[i] = 0; sanalSayi[i] = 0;}
            // bit çevrimi yapılıyor.
            n = ornekSayisi; i2 = n >> 1; j = 0;
            for (i = 0; i < n - 1; i++){
                if (i < j){
                    geciciX = gercekSayi[i]; geciciY = sanalSayi[i];
                    gercekSayi[i] = gercekSayi[j]; gercekSayi[j] = geciciX;
                    sanalSayi[i] = sanalSayi[j]; sanalSayi[j] = geciciY;
                }
                k = i2; while (k <= j){ j -= k; k >>= 1; } j += k;
            }
        }
    }
}
```

```

basamakAtlama = 1;
for (i = 0; i < bitSayisi; i++){
    // FFT Bit sayısı kadar işlem
    islemSayi = basamakAtlama * 2;
    //Radix2 için gereken çapraz işlem (butterfly) sayısı
    for (j = 0; j < ornekSayisi / islemSayi; j++){
        for (k = 0; k < basamakAtlama; k++){
            int taban = j * islemSayi + k;
            double pi_carpi_bolum = 6.28318 * k / islemSayi;
            sinus = Math.Sin(pi_carpi_bolum);
            kosinus = Math.Cos(pi_carpi_bolum);
            reel = gercekSayi[taban + basamakAtlama] * kosinus - sanalSayi[taban +
basamakAtlama] * sinus;
            sanal = gercekSayi[taban + basamakAtlama] * sinus + sanalSayi[taban +
basamakAtlama] * kosinus;
            gercekSayi[taban + basamakAtlama] = gercekSayi[taban] - reel;
            sanalSayi[taban + basamakAtlama] = sanalSayi[taban] - sanal;
            gercekSayi[taban] = gercekSayi[taban] + reel;
            sanalSayi[taban] = sanalSayi[taban] + sanal;
        }
    }
    basamakAtlama *= 2;
}

//Bu işlem temel frekans değerini birim değer olarak kabul edip diğer değerleri tek
döngüyle normalleştirmek için yapıldı
geciciX = 3.266 * (Math.Sqrt(gercekSayi[temelFrekans] * gercekSayi[temelFrekans] +
sanalSayi[temelFrekans] * sanalSayi[temelFrekans])) / gelenOrnekSayisi;
// genlik ölçümlendirmesi yapılıyor
for (i = 0; i < ornekSayisi / 2; i++){
    // nyquiste göre yarısı anlamlı
    gercekSayi[i] = (3.266 * (Math.Sqrt(gercekSayi[i] * gercekSayi[i] + sanalSayi[i] *
sanalSayi[i])) / gelenOrnekSayisi) / geciciX;
}
//THD hesabı yapılacak

```

```

THD = 0;
for (i = temelFrekans * 2; i < temelFrekans * ThdSayisi; ){
    // THD hesabı yapılıyor.
    THD += Math.Pow((gercekSayi[i] + gercekSayi[i + 1] + gercekSayi[i - 1]), 2);
    i += temelFrekans;
}
THD = Math.Sqrt(THD);
THD = (THD * 100) / (gercekSayi[temelFrekans - 1] + gercekSayi[temelFrekans] +
gercekSayi[temelFrekans+1]);
} //public void FFT_AL(byte[] veri)
} //class FFT
} //namespace Tez_Uygulama

```

Ek. 3. Fazlardan Her Biri için Hesaplama Kodları

```
// Yeni FFT nesnesi oluşturuluyor.
fftAkim = new FFT(); fftGerilim = new FFT();
fftAkim.FFT_AL(akimFaz1); fftGerilim.FFT_AL(gerilimFaz1); thdSayac++;
fftAkim.THD = Math.Round(fftAkim.THD, 2);
fftGerilim.THD = Math.Round(fftGerilim.THD, 2);
LPF_Faz1[0] = (1 - alfa) * LPF_Faz1[0] + alfa * fftAkim.THD;
LPF_Faz1[1] = (1 - alfa) * LPF_Faz1[1] + alfa * fftGerilim.THD;
if (!double.IsNaN(fftAkim.THD)) csvVeriYapisi[1] += fftAkim.THD;
if (!double.IsNaN(fftGerilim.THD)) csvVeriYapisi[4] += fftGerilim.THD;
I1RMS[0] = RmsDegerler[0] / Math.Sqrt(Math.Pow(fftAkim.THD / 100, 2) + 1);
V1RMS[0] = RmsDegerler[1] / Math.Sqrt(Math.Pow(fftGerilim.THD / 100, 2) + 1);
for (int sayac2 = 0; sayac2 < veriMiktari; sayac2++){
    if (sayac2 >= faz_kayma) {
        aktif_averaj[0] += gerilimFaz1[sayac2] * akimFaz1[sayac2]; //Faz1 için hesaplar
        aktif1 = gerilimFaz1[sayac2] * akimFaz1[sayac2] + gerilimFaz1[sayac2 - faz_kayma] *
akimFaz1[sayac2 - faz_kayma];
        reaktif1 = gerilimFaz1[sayac2] * akimFaz1[sayac2 - faz_kayma] - gerilimFaz1[sayac2 -
faz_kayma] * akimFaz1[sayac2];
        aktif_averaj[1] += gerilimFaz2[sayac2] * akimFaz2[sayac2]; //Faz2 için hesaplar
        aktif2 = gerilimFaz2[sayac2] * akimFaz2[sayac2] + gerilimFaz2[sayac2 - faz_kayma] *
akimFaz2[sayac2 - faz_kayma];
        reaktif2 = gerilimFaz2[sayac2] * akimFaz2[sayac2 - faz_kayma] - gerilimFaz2[sayac2 -
faz_kayma] * akimFaz2[sayac2];
        aktif_averaj[2] += gerilimFaz3[sayac2] * akimFaz3[sayac2]; //Faz3 için hesaplar
        aktif3 = gerilimFaz3[sayac2] * akimFaz3[sayac2] + gerilimFaz3[sayac2 - faz_kayma] *
akimFaz3[sayac2 - faz_kayma];
        reaktif3 = gerilimFaz3[sayac2] * akimFaz3[sayac2 - faz_kayma] - gerilimFaz3[sayac2 -
faz_kayma] * akimFaz3[sayac2];
    } else{aktif1 = gerilimFaz1[sayac2] * akimFaz1[sayac2]; reaktif1 = 0;
        aktif2 = gerilimFaz2[sayac2] * akimFaz2[sayac2]; reaktif2 = 0;
        aktif3 = gerilimFaz3[sayac2] * akimFaz3[sayac2]; reaktif3 = 0;
    }
    gorunur1 = Math.Sqrt(aktif1 * aktif1 + reaktif1 * reaktif1);
```

```

gorunur2 = Math.Sqrt(aktif2 * aktif2 + reaktif2 * reaktif2);
gorunur3 = Math.Sqrt(aktif3 * aktif3 + reaktif3 * reaktif3);
if (menuGucGrafik.Checked == true) {
    guc2Faz1.Add(sayac2 * zamanKademe, aktif1); // AKTİF GÜÇ
    guc2Faz2.Add(sayac2 * zamanKademe, reaktif1); // REAKTİF GÜÇ
    guc2Faz3.Add(sayac2 * zamanKademe, gorunur1); // GORUNUR GUÇ
    guc3Faz1.Add(sayac2 * zamanKademe, aktif2); // AKTİF GÜÇ
    guc3Faz2.Add(sayac2 * zamanKademe, reaktif2); // REAKTİF GÜÇ
    guc3Faz3.Add(sayac2 * zamanKademe, gorunur2); // GORUNUR GUÇ
    guc1Faz1.Add(sayac2 * zamanKademe, aktif3); // AKTİF GÜÇ
    guc1Faz2.Add(sayac2 * zamanKademe, reaktif3); // REAKTİF GÜÇ
    guc1Faz3.Add(sayac2 * zamanKademe, gorunur3); // GORUNUR GUÇ
    aktifler[0] += (aktif1); reaktifler[0] += (reaktif1); gorunurler[0] += (gorunur1);
    aktifler[1] += (aktif2); reaktifler[1] += (reaktif2); gorunurler[1] += (gorunur2);
    aktifler[2] += (aktif3); reaktifler[2] += (reaktif3); gorunurler[2] += (gorunur3);
}
aktifler[0] /= veriMiktari; aktifler[1] /= veriMiktari;    aktifler[2] /= veriMiktari;
aktif_averaj[0] /= veriMiktari; aktif_averaj[1] /= veriMiktari; aktif_averaj[2] /= veriMiktari;
reaktifler[0] /= veriMiktari; reaktifler[1] /= veriMiktari; reaktifler[2] /= veriMiktari;
gorunurler[0] /= veriMiktari; gorunurler[1] /= veriMiktari; gorunurler[2] /= veriMiktari;
cosphi[0] = Math.Abs(aktif_averaj[0] / (V1RMS[0] * I1RMS[0])); PF[0] =
Math.Abs(aktif_averaj[0] / (V1RMS[0] * RmsDegerler[0]));
cosphi[1] = Math.Abs(aktif_averaj[1] / (V1RMS[1] * I1RMS[1])); PF[1] =
Math.Abs(aktif_averaj[1] / (V1RMS[1] * RmsDegerler[2]));
cosphi[2] = Math.Abs(aktif_averaj[2] / (V1RMS[2] * I1RMS[2])); PF[2] =
Math.Abs(aktif_averaj[2] / (V1RMS[2] * RmsDegerler[4]));
LPF_Faz1[4] = (1 - alfa) * LPF_Faz1[4] + alfa * cosphi[0];
LPF_Faz1[5] = (1 - alfa) * LPF_Faz1[5] + alfa * PF[0];
LPF_Faz2[4] = (1 - alfa) * LPF_Faz2[4] + alfa * cosphi[1];
LPF_Faz2[5] = (1 - alfa) * LPF_Faz2[5] + alfa * PF[1];
LPF_Faz3[4] = (1 - alfa) * LPF_Faz3[4] + alfa * cosphi[2];
LPF_Faz3[5] = (1 - alfa) * LPF_Faz3[5] + alfa * PF[2];

```

Ek. 4. DSP Program Kodları

```
#include "PeripheralHeaderIncludes.h"
```

```

#include "Math.h"
#include "SPLL_1ph_SOGI_F.h"
#pragma CODE_SECTION(rx_isr, "ramfuncs");
#pragma CODE_SECTION(adc_isr, "ramfuncs");
void MemCopy(Uint16 *SourceAddr, Uint16* SourceEndAddr, Uint16* DestAddr);
void DeviceInit(void); void InitFlash(void);
__interrupt void rx_isr(void); __interrupt void adc_isr(void);
void InitPieCtrl(void); void scia_xmit(Uint16 a);
#ifdef FLASH
extern Uint16 RamfuncsLoadStart, RamfuncsLoadEnd, RamfuncsRunStart;
#endif

static volatile Uint8 say, frekans, frekans_durum, OffsetDurum , veriGonder,veriGonder2,i;
Uint16 GPIO34_count = 0, Gelen, ii;
static volatile Uint16 Spll_cevrim_sayi = 0;
#define ISR_FREQ 20000
#define Adc_Offset 7
struct ADC_BITLER{
    Uint16 alt :8;          // 7:0   alt byte
    Uint16 ust :8;          // 15:8  ust byte
};
union ADC_GONDERIM{
    Uint16 tumu;             // union yapısındaki tüm bitler
    struct ADC_BITLER bit;   // struct yapısı uniona ekleniyor
} ADC_DEGERLER;             // yapı değişkeni
int32 Adc_calib[Adc_Offset] ={ 0 };
#define PI 3.141592653
#define Birim_Genlik 620//2430
#define Saykil          400 //20khz de 1 periyot 400 eder.
SPLL_1ph_SOGI_F spll1;
float32 Periyot, OrtalamaFrekans, OrnekFrekans;
void main(void)
{
    Uint16 i = 0; say = 0;
    DeviceInit(); DINT; InitPieCtrl();

```

```

IER = 0x0000; IFR = 0x0000;

EALLOW;

PieVectTable.SCIRXINTA = &rx_isr;
PieVectTable.ADCINT2 = &adc_isr;

EDIS;

#ifdef FLASH

    MemCopy(&RamfuncsLoadStart, &RamfuncsLoadEnd, &RamfuncsRunStart);

    InitFlash();

#endif //(FLASH)

//      Çevresel Birimler Aktif ediliyor
//-----ADC başlangıç -----

EALLOW;

AdcRegs.ADCCTL1.bit.ADCREFSEL = 0;    // dahili bandgap
AdcRegs.ADCCTL1.bit.ADCPWDN = 1;      // ADC yi çalıştır
AdcRegs.ADCCTL1.bit.ADCBGPWD = 1;     //
AdcRegs.ADCCTL1.bit.ADCREFPWD = 1;

for (i = 0; i < 65000; i++){// ADC açılış zamanı= 1ms
AdcRegs.ADCCTL1.bit.ADCENABLE = 1;    // Enable ADC
AdcRegs.ADCCTL2.bit.CLKDIV2EN = 1; // sistem saati ile adc clock aynı olmalı
AdcRegs.ADCCTL2.bit.CLKDIV4EN = 0; // sistem saati ile adc clock aynı olmalı
for (i = 0; i < 65000; i++){
AdcRegs.ADCCTL2.bit.ADCNONOVERLAP = 1;    //Enable non-overlap mode

// SOC0 : FAZ1 akım bilgisi          SOC1 : Faz1 gerilim bilgisi//
AdcRegs.ADCSAMPLEMODE.bit.SIMULEN0 = 1; // Boş kanallar için
AdcRegs.ADCSAMPLEMODE.bit.SIMULEN2 = 1; // soc4 -soc5 aktif // Faz2 değerler
AdcRegs.ADCSAMPLEMODE.bit.SIMULEN4 = 1; // soc6 -soc7 aktif // Faz3 değerler
AdcRegs.ADCSAMPLEMODE.bit.SIMULEN6 = 1; // soc2 -soc3 aktif // Faz1 değerler
AdcRegs.ADCSOC0CTL.bit.CHSEL = 0; // SOC1 DE ADCINA0 ve B0 VAR
AdcRegs.ADCSOC2CTL.bit.CHSEL = 6; // SOC1 DE ADCINA6 ve B6 VAR
AdcRegs.ADCSOC4CTL.bit.CHSEL = 1; // SOC1 DE ADCINA1 ve B1 VAR
AdcRegs.ADCSOC6CTL.bit.CHSEL = 2; // SOC1 DE ADCINA2 ve B2 VAR
AdcRegs.ADCSOC8CTL.bit.CHSEL = 0xc; // SOC8 DA ADCINB4 VAR

// 7 adc clock akış miktarı

AdcRegs.ADCSOC0CTL.bit.ACQPS = 6; // 7 ADC Clock, (6 ACQPS + 1)

```

```

AdcRegs.ADCSOC2CTL.bit.ACQPS = 6;
AdcRegs.ADCSOC4CTL.bit.ACQPS = 6;
AdcRegs.ADCSOC6CTL.bit.ACQPS = 6;
AdcRegs.ADCSOC8CTL.bit.ACQPS = 6;
// SOC TETİKLEYİCİLER AYARLANIYOR
AdcRegs.ADCSOC8CTL.bit.TRIGSEL = 5; // ePwm1 ile sabit 20khz tetikleme
AdcRegs.ADCSOC0CTL.bit.TRIGSEL = 1; // Faz1 değerleri için adc timer0 ile
AdcRegs.ADCSOC2CTL.bit.TRIGSEL = 1; // Faz2 değerleri için adc timer0 ile
AdcRegs.ADCSOC4CTL.bit.TRIGSEL = 1; // Faz3 değerleri için adc timer0 ile
AdcRegs.ADCSOC6CTL.bit.TRIGSEL = 1; // Faz1 değerleri için adc timer0 ile
tetikleniyor
//-----ADC Interrupt Ayarları-----
AdcRegs.INTSEL1N2.bit.INT1E = 1; // Interrupt oluştur    ADCINT1 VERİLER
AdcRegs.INTSEL1N2.bit.INT1CONT = 0; // sürekli çevrim devredışı
AdcRegs.ADCCTL1.bit.INTPULSEPOS = 1; // Çevrim bitince interrupt oluştur
AdcRegs.INTSEL1N2.bit.INT1SEL = 7; // EOC7 interrupt oluşturacak
AdcRegs.INTSEL1N2.bit.INT2E = 1; // Interrupt oluştur    ADCINT2 PLL İÇİN
AdcRegs.INTSEL1N2.bit.INT2CONT = 1; // sürekli çevrim devredışı
AdcRegs.ADCCTL1.bit.INTPULSEPOS = 1; // Çevrim bitince interrupt oluştur
AdcRegs.INTSEL1N2.bit.INT2SEL = 8; // EOC8 interrupt oluşturacak
//----- Seri Port Ayarları-----
SciaRegs.SCICCR.all = 0x0007;
SciaRegs.SCIFFCT.all = 0x0;
SciaRegs.SCIFFTX.all = 0xE040; //111 00000 0100 0000
SciaRegs.SCIFFRX.all = 0x6063; //0x2364; //001 00000 0110 0100 fifo 3 basamak
SciaRegs.SCIHBAUD = 0x00;
SciaRegs.SCILBAUD = 0x01; //4;
SciaRegs.SCICTL1.all = 0x0023; // SCI başlat
//--- ePWM1 ayarları
EPwm1Regs.TBCTL.bit.CTRMODE = 0x3; // timer durdu
EPwm1Regs.TBCTL.all = 0xC033; // ayarların tümü
EPwm1Regs.TBCTR = 0x0000; /
EPwm1Regs.TBPRD = 3999; // 20khz;
EPwm1Regs.TBPHS.half.TBPHS = 0x0000; /

```

```

EPwm1Regs.ETPS.all = 0x0100;                                /
EPwm1Regs.ETPS.bit.INTPRD = 1;
EPwm1Regs.ETSEL.all = 0x0A00;                                // Enable SOCA to ADC
EPwm1Regs.ETSEL.bit.INTEN = 1;                                // Enable SOCA to ADC
EPwm1Regs.ETSEL.bit.INTSEL = 2;                                // Enable SOCA to ADC
EDIS;
//----- PLL ayarları -----
SPLL_1ph_SOGI_F_init(50.0, ((float) (1.0 / ISR_FREQ)), &spll1);
SPLL_1ph_SOGI_F_coeff_update(((float) (1.0 / ISR_FREQ)),
                                ((float) (2 * PI * 50.0)), &spll1);
//----- PLL ayarları -----
InitCpuTimers(); // cpu sayıcıları ayarlanıyor.
// Configure CPU-Timer 0, 80MHz CPU Freq, 1 second Period (in uSeconds)
ConfigCpuTimer(&CpuTimer0, 80, (float) (1000000 / 10240));
EALLOW;
PieCtrlRegs.PIEIER1.bit.INTx2 = 1; // SOGI PLL için ADCINT2 interruptı açılıyor
PieCtrlRegs.PIEIER9.bit.INTx1 = 1; // SCIRXINTA aktif
PieCtrlRegs.PIECTRL.bit.ENPIE = 1; // Enable the PIE
IER |= M_INT1; // ADCINT2 interrupt aktif
IER |= M_INT9; // scia interrupt
///-----ADC OFFSET-----
for (i = 0; i < Adc_Offset; i++)
    Adc_calib[i] = 0;
EPwm1Regs.TBCTL.bit.CTRMODE = 0; //sogipll için çevrime başla
while (ii < 10240) { // bu kısım ilk 10 periyotta offset hesaplama için
    if (AdcRegs.ADCINTFLG.bit.ADCINT2 == 1) {
        AdcRegs.ADCINTFLGCLR.bit.ADCINT2 = 1;
        Adc_calib[6] += AdcResult.ADCRESULT8;
        ii++;
    }
}
EPwm1Regs.TBCTL.bit.CTRMODE = 3; //zamanlayıcıyı durdur
StartCpuTimer0();
while (ii < 10240) { // bu kısım ilk 10 periyotta offset hesaplama için

```

```

    if (AdcRegs.ADCINTFLG.bit.ADCINT1 == 1){
        AdcRegs.ADCINTFLGCLR.bit.ADCINT1 = 1;
        Adc_calib[0] += AdcResult.ADCRESULT2;   Adc_calib[0] -= 2048;
        Adc_calib[1] += AdcResult.ADCRESULT3;   Adc_calib[1] -= 2048;
        Adc_calib[2] += AdcResult.ADCRESULT4;   Adc_calib[2] -= 2048;
        Adc_calib[3] += AdcResult.ADCRESULT5;   Adc_calib[3] -= 2048;
        Adc_calib[4] += AdcResult.ADCRESULT6;   Adc_calib[4] -= 2048;
        Adc_calib[5] += AdcResult.ADCRESULT7;   Adc_calib[5] -= 2048;
        ii++;
    }
}

StopCpuTimer0();
for (i = 0; i < Adc_Offset; i++) Adc_calib[i] /= 10240;
EINT; ERTM; EDIS; Periyot = 0;   OrtalamaFrekans = 0; OrnekFrekans = 0;
EPwm1Regs.TBCTL.bit.CTRMODE = 0;   //sogipll için çevrime başla
for (;;) {
    if (AdcRegs.ADCINTFLG.bit.ADCINT1 == 1) { //eğer
        AdcRegs.ADCINTFLGCLR.bit.ADCINT1 = 1;
        AdcRegs.ADCINTOVFCLR.bit.ADCINT1 = 1;
        if (frekans_durum == 10) { //örnekleme frekans değişim zamanı
            ConfigCpuTimer(&CpuTimer0, 80,
                           (float) (4882.8125 / OrtalamaFrekans));
            StartCpuTimer0(); //örnekleme frekansı değişti
            frekans_durum = 0;}
        ADC_DEGERLER.tumu = AdcResult.ADCRESULT2 - Adc_calib[0];
        scia_xmit(ADC_DEGERLER.bit.ust); //Faz1 akım bilgisi
        scia_xmit(ADC_DEGERLER.bit.alt);
        ADC_DEGERLER.tumu = AdcResult.ADCRESULT3 - Adc_calib[1];
        scia_xmit(ADC_DEGERLER.bit.ust); //Faz1 gerilim bilgisi
        scia_xmit(ADC_DEGERLER.bit.alt);
        ADC_DEGERLER.tumu = AdcResult.ADCRESULT4 - Adc_calib[2];
        scia_xmit(ADC_DEGERLER.bit.ust); //Faz2 akım bilgisi
        scia_xmit(ADC_DEGERLER.bit.alt);

```

```

        ADC_DEGERLER.tumu = AdcResult.ADCRESULT5 - Adc_calib[3];
        scia_xmit(ADC_DEGERLER.bit.ust); //Faz2 gerilim bilgisi
        scia_xmit(ADC_DEGERLER.bit.alt);
        ADC_DEGERLER.tumu = AdcResult.ADCRESULT6 - Adc_calib[4];
        scia_xmit(ADC_DEGERLER.bit.ust); //Faz3 akım bilgisi
        scia_xmit(ADC_DEGERLER.bit.alt);
        ADC_DEGERLER.tumu = AdcResult.ADCRESULT7 - Adc_calib[5];
        scia_xmit(ADC_DEGERLER.bit.ust); //Faz3 gerilim bilgisi
        scia_xmit(ADC_DEGERLER.bit.alt);
        if (GPIO34_count++ >= 2047) { // 10 periyotluk bilgi
            GpioDataRegs.GPBTGGLE.bit.GPIO34 = 1;
            GPIO34_count = 0;    StopCpuTimer0();
        }
        veriGonder = 0; i = 0;
    }
}

// SOGIPLL işlemleri ADC interruptında yapılıyor
__interrupt void adc_isr(void){
    spll1.u[0] = ((float32)(AdcResult.ADCRESULT8- Adc_calib[6])/Birim_Genlik);
    SPL1_1ph_SOGI_F_MACRO(spll1);
    OrnekFrekans += spll1.fo;
    if (Spll_cevrim_sayi++ >= Saykil) { // Eğer bir periyotluk veri alındıysa
        Spll_cevrim_sayi = 0; // sayacı sıfırla
        OrtalamaFrekans = OrnekFrekans / Saykil;
        OrnekFrekans = 0; frekans_durum = 10;
    }
    AdcRegs.ADCINTFLGCLR.bit.ADCINT2 = 1; // İnterrupt bayrağını sıfırla
    PieCtrlRegs.PIEACK.all = PIEACK_GROUP1; // İnterrupt servis rutini işlendi
    return;
}

```

ÖZGEÇMİŞ

Mustafa TOPUZ, 1981 yılında Bozüyük'te doğdu. Orta öğrenimini Eskişehir'de tamamladıktan sonra 1999 yılında Kocaeli Üniversitesi Elektronik Öğretmenliği Bölümü'nde lisans eğitimine başladı. 2005 yılından beri Milli Eğitim Bakanlığı'nda Elektronik Öğretmeni olarak çalışmaktadır.