



ELEKTRİKLİ ARAÇLAR İÇİN TELEMETRİ SİSTEMİ TASARIMI

YÜKSEK LİSANS TEZİ

Emre MANIR

Danışman

Yrd. Doç. Dr. Said Mahmut ÇINAR

YENİLENEBİLİR ENERJİ SİSTEMLERİ

ANABİLİM DALI

Mayıs 2017

Bu tez çalışması 16. FEN. BİL.36 numaralı proje ile BAPK tarafından desteklenmiştir.

AFYON KOCATEPE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

ELEKTRİKLİ ARAÇLAR İÇİN TELEMETRİ SİSTEMİ TASARIMI

Emre MANIR

Danışman
Yrd. Doç. Dr. Said Mahmut ÇINAR

YENİLENEBİLİR ENERJİ SİSTEMLERİ ANABİLİM DALI

MAYIS 2017

TEZ ONAY SAYFASI

Emre MANIR tarafından hazırlanan “Elektrikli araçlar için telemetri sistemi tasarımı” adlı tez çalışması lisansüstü eğitim ve öğretim yönetmeliğinin ilgili maddeleri uyarınca 12/05/2017 tarihinde aşağıdaki jüri tarafından oy birliği ile Afyon Kocatepe Üniversitesi Fen Bilimleri Enstitüsü **Yenilenebilir Enerji Sistemleri Anabilim Dalı’nda YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Danışman : Yrd. Doç. Dr. Said Mahmut ÇINAR

Başkan : Yrd. Doç. Dr. Mustafa NİL
Manisa Celal Bayar Üniversitesi, Mühendislik Fakültesi

Üye : Doç. Dr. Fatih Onur HOCAOĞLU
Afyon Kocatepe Üniversitesi, Mühendislik Fakültesi

Üye : Yrd. Doç. Dr. Said Mahmut ÇINAR
Afyon Kocatepe Üniversitesi, Mühendislik Fakültesi

Afyon Kocatepe Üniversitesi
Fen Bilimleri Enstitüsü Yönetim Kurulu’nun
...../...../..... tarih ve
..... sayılı kararıyla onaylanmıştır.

.....
Prof. Dr. Hüseyin ENGİNAR
Enstitü Müdürü

BİLİMSEL ETİK BİLDİRİM SAYFASI
Afyon Kocatepe Üniversitesi

**Fen Bilimleri Enstitüsü, tez yazım kurallarına uygun olarak hazırladığım
bu tez çalışmada;**

- Tez içindeki bütün bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğimi,
- Görsel, işitsel ve yazılı tüm bilgi ve sonuçları bilimsel ahlak kurallarına uygun olarak sunduğumu,
- Başkalarının eserlerinden yararlanılması durumunda ilgili eserlere bilimsel normlara uygun olarak atıfta bulunduğumu,
- Atıfta bulunduğum eserlerin tümünü kaynak olarak gösterdiğimi,
- Kullanılan verilerde herhangi bir tahrifat yapmadığımı,
- Ve bu tezin herhangi bir bölümünü bu üniversite veya başka bir üniversitede başka bir tez çalışması olarak sunmadığımı

beyan ederim.

12/05/2017


İmza
Emre MANIR

ÖZET
Yüksek Lisans Tezi

ELEKTRİKLİ ARAÇLAR İÇİN TELEMETRİ SİSTEMİ TASARIMI

Emre MANIR

Afyon Kocatepe Üniversitesi

Fen Bilimleri Enstitüsü

Yenilenebilir Enerji Sistemleri Anabilim Dalı

Danışman: Yrd. Doç. Dr. Said Mahmut ÇINAR

Telemetri sistemleri, bir sürecin veya sistemin uzaktan kablolu veya kablosuz olarak izlenmesi ya da kontrol edilmesinde kullanılır. Bu çalışmada elektrikli araçlar için RF tabanlı telemetri sistemi oluşturulmuş ve veri kaybını önlemeye yönelik bir kontrol algoritması tasarlanmıştır. Tasarlanan algorithmada, RF haberleşmenin kısmen veya tamamen bloke olması durumunda, gönderilemeyen veri paketlerini kalıcı hafızaya kaydederek, farklı bir taşıma frekansına otomatik geçiş yapabilen bir frekans tarama süreci yer almaktadır. Söz konusu algoritma, RF modemler üzerinden haberleşen mikro denetleyici tabanlı bir test platformu üzerinde test edilmiştir. Gerçekleştirilen deneylerden elde edilen sonuçlar haberleşmenin bloke olduğu durumlarda tasarlanan algoritmanın gönderilemeyen veri paketlerini kalıcı hafızaya kaydettiğini, taşıma frekansını kaydırmak suretiyle haberleşmenin tekrar sağlanmasının ardından gönderilemeyen verilerin yer istasyonuna gönderildiğini göstermiştir.

2017, viii + 58 sayfa

Anahtar Kelimeler: Telemetri, RF modem, Frekans kaydırma, Veri toplama

ABSTRACT
M. Sc. Thesis

TELEMETRY SYSTEM DESIGNE FOR ELECTRIC VEHICLES

Emre MANIR

Afyon Kocatepe University

Graduate School of Natural and Applied Sciences

Department of Electrical Engineering

Supervisor: Asst. Prof. Said Mahmut ÇINAR

Telemetry systems are used to remotely control and/or monitor a system as wired or wireless. In this study, a control algorithm is designed to prevent data loss for RF based telemetry system. In the designed algorithm, in case of RF transmission a partial or complete blockage, there is an automatic frequency shifting process that can switch to a different carrier frequency. A micro-controller-based test platform designed to test the purposed algorithm is carried out and RF modems are used in the test platform. The results obtained from the tests show that the data which could not sent in blockage of RF transmission saved in eeprom memory and the data can be re sent to center telemetry unit after the removed blockage.

2017, viii + 58 pages

Keywords: Telemetry, RF modem, Frequency shift, Data collecting

TEŞEKKÜR

Bu araştırmanın konusu, deneysel çalışmaların yönlendirilmesi, sonuçların değerlendirilmesi ve yazımı aşamasında yapmış olduğu büyük katkılarından dolayı tez danışmanım Sayın Yrd. Doç. Dr. Said Mahmut ÇINAR'a her konuda öneri ve eleştirileriyle yardımlarını gördüğüm hocalarıma ve arkadaşlarıma ve bu tez çalışmasında maddi destek veren Afyon Kocatepe Üniversitesi Bilimsel Araştırma Projeleri Birimi'ne (Proje No: 16. FEN. BİL. 36) teşekkür ederim.

Bu araştırma boyunca maddi ve manevi desteklerinden dolayı aileme teşekkür ederim.

Emre MANIR
AFYONKARAHİSAR, 2017

İÇİNDEKİLER DİZİNİ

Sayfa

ÖZET	i
ABSTRACT	ii
TEŞEKKÜR	iii
İÇİNDEKİLER DİZİNİ	iv
KISALTMALAR DİZİNİ	v
ŞEKİLLER DİZİNİ	vi
ÇİZELGELER DİZİNİ	vii
RESİMLER DİZİNİ.....	viii
1. GİRİŞ	1
2. LİTERATÜR BİLGİLERİ	3
3. MATERYAL ve METOT	7
3.1 Materyal	7
3.1.1 Mikro Denetleyici Tabanlı Telemetry Geliştirme Bordu.....	7
3.1.2 PIC16F877A Mikro Denetleyicisi	8
3.1.2.1 SayıcıZamanlayıcı Birimi	8
3.1.2.2 Analog-Dijital Çevirici	9
3.1.2.3 Seri Haberleşme Birimi (RS232).....	10
3.1.3 RF Modem.....	10
3.1.4 RF Spektrum Analizörü	13
3.2 Metot	14
3.2.1 Tasarlanan Telemetry Kartları.....	14
3.2.1.1 Tasarlanan Telemetry Kontrol Algoritması	21
3.2.1.2 Mikro Denetleyicide Kullanılan Program.....	23
4. SENARYO	30
4.1 Bulgular	30
5. TARTIŞMA ve SONUÇ	32
6. KAYNAKLAR	33
6.1 İnternet Kaynakları	36
ÖZGEÇMİŞ.....	37
EKLER	38
EK 1. Veri Toplama Modülü Program Kodu	38
EK 2. Merkez Modülü Program Kodu.....	49

KISALTMALAR DİZİNİ

Kısaltmalar

RF	Radyo frekansı
USB	Evrensel seri veri yolu
GSM	Mobil iletişim için küresel sistem
GPRS	Paket kuralına dayalı radyo servisi
GPS	Küresel yer belirleme sistemi
KMC	Kısa mesafeli cihaz(SDR)
EBM	Endüstriyel-Bilimsel-Medikal(ISM)
İKSB	İndirgenmiş komut setli bilgisayar(RISC)
THSŞ	Tümleşik hizmetler sayısal şebekesi(ISDN)
Yİ	Yalın internet(Ndsl)
IR	Kızılötesi ışıyım(infra-red)
SKY	Sayısal karasal yayıncılık(DVB-T)
DSY	Dijital ses yayını(DAB)
FM	Frekans modülasyonu
YTR	Yazılım temelli radyo(SDR)

ŞEKİLLER DİZİNİ

Sayfa

Şekil 3.1 Telemetri kartlarının besleme, SDI kart ve gerçek zamanlı saat devre şemaları	15
Şekil 3.2 Telemetri haberleşme portları.....	17
Şekil 3.3 Telemetri Ana Kart Şeması.....	18
Şekil 3.4 Tasarlanan telemetri kartlarının bakır yüzeyi.....	19
Şekil 3.5 Tasarlanan telemetri kartlarının malzeme montaj yüzeyi.....	20
Şekil 3.6 Tasarlanan telemetri kartlarının üç boyutlu görünümü.....	20
Şekil 3.7 Çalışmada kullanılan algoritmanın akış şeması.....	21

ÇİZELGELER DİZİNİ

Sayfa

Çizelge 3.1 16F877A Mikro denetleyicisinin özellikleri.....	8
Çizelge 3.2 RF Modem parametreleri tablosu.....	13



RESİMLER DİZİNİ

Sayfa

Resim 3.1 Mikro denetleyici tabanlı telemetri geliştirme bordu.....	7
Resim 3.2 SNR611 RF modem.....	11
Resim 3.3 SNR 611 RF modem ara yüz programı.....	12
Resim 3.4 SDR AIRSPY spektrum analizörü.....	14
Resim 4.1 Frekans kaydırma işlemi	31



1. GİRİŞ

Bir sistem ya da tesisin uzaktan kablo veya kablosuz olarak izlenmesi veya kontrol edilmesi işlemine telemetri denilmektedir. Genel itibariyle telemetri denilince kablosuz haberleşme anlaşılmaktadır. Bunların en yoğun kullanılanları RF modem cihazları, GSM, GPRS, GPS sistemleridir (İnt. Kyn. 1).

Kablosuz telemetri sisteminin kullanıldığı yerlerde arazi düz olmak zorunda değildir; antenler arasında dağ, tepe vb. haberleşmeyi engelleyici unsurlar olsa bile yönlendirici özellikli RF modemler ile bu tür engeller aşılabilmektedir. Haberleşmenin sağlanacağı noktaların birbirine olan uzaklıkları kilometrelerce olabilir. Günümüzde yüzlerce kilometre mesafeler arasında RF modem sistemi ile iletişim kurma imkânı bulunmaktadır (Connorsn 2007).

GSM ve GPRS sistemleri çevrimiçi işlem yapmak için değil, telemetri cihazının içerisindeki veri toplama cihazındaki bilgileri gerektiğinde iletmek için kullanılmaktadır. Fakat işletme maliyeti ve afet durumlarında kısa zaman aralığında kullanım yoğunluğu nedeniyle sistemin aşırı yüklenmesi sonucu haberleşmede aksaklıklar söz konusu olabilmektedir. Ancak oluşan önemli olay ya da çok kişiyi ilgilendiren hal ve durumlarla ilgili uyarıları aynı anda GSM ve GPRS ile çalışan elektronik cihazlara (örnek: cep telefonlarına) mesaj olarak gönderebilmek gibi avantajlara da sahip olduğunu söylemeliyiz (Mammadov ve Gueaieb 2014).

GPS sistemleri çevrimiçi izleme için tercih edilmektedir. Ancak olumsuz hava koşullarından etkilenmektedir. Örneğin yağmur, kar gibi meteorolojik olaylar haberleşme kalitesini düşürmektedir.

Günümüzde maliyetinin düşüklüğü nedeniyle veri iletimi için en ideal çözüm olarak güçlü endüstriyel RF Modemler karşımıza çıkmaktadır. Lisanslı ve lisanssız bantta yayın yapabilmektedirler. 100 km mesafede haberleşme sağlamaktadırlar. Ayrıca kısa mesafeli uygulamalarda KMC veya EBM radyo bandındaki düşük güçlü radyo çözümleri ile 0-27 dBm (500 mW'a kadar) yayın gücünde 433 MHz, 868 MHz, 2.4 GHz

ve 5.8/5.9 GHz bantlarında lisanssız radyo telemetri uygulamaları giderek yaygınlaşmaktadır. Ancak bu uygulamalar sadece kampüs alanlarında kullanılabilir, halka açık alanda 2.4 Ghz kablosuz sistemler ile uygulama yapmak yasal değildir. Özellikle kablosuz algılayıcı ağları uygulamalarında milyonlarca uç birim bu EBM bant radyo ara yüzü ile donatılmış durumdadır. Telemetri uygulamaları çoğunlukla kablosuz olmakla birlikte optik kablo (fiber), analog telefon hatları, sayısal telefon hatları (THSŞ/Yİ), kızıl ötesi (IR), ses dalgaları (akustik-ultrasonik), mikro dalga iletişimi gibi farklı iletişim araçları ve ortamları üzerinden de gerçekleştirilmektedir (Tucker 2010).

Petrol, gaz, su boru hatlarının izlenmesi, katodik koruma sistemlerinin izlenmesi, uydu ve baz istasyonlarının uzaktan izlenmesi ve kontrolü, radyo ve televizyon vericilerinin uzaktan izlenmesi, içme suyu ve atık su tesislerinin izlenmesi, su pompalarının su deposundaki seviyeye göre devreye girip çıkması, trafo merkezlerinin uzaktan izlenmesi, elektrik sayaçlarının uzaktan izlenmesi sistemleri telemetri ve tele kontrol sistemlerine örnek uygulamalar olarak sayılabilir (İnt. Kyn. 3).

Bu tez çalışmasında, özellikle TÜBİTAK tarafından düzenlenen elektrikli araç yarışları için tasarlanan araçlardakullanılabilecek birtelemetri sistemi gerçekleştirilmiş ve akıllı algoritmayla sistemin kontrolü sağlanmıştır. Söz konusu algoritma RF modemler kullanılan mikro denetleyici tabanlı bir platformda test edilmiştir. Hazırlanan test platformu kullanılarak, oluşturulan senaryo kapsamında gerçekleştirilen deneylerle tasarlanan algoritma test edilmiştir.

Aşağıda ikinci bölümde, literatürde yer alan telemetri sistemleriyle ilgili bir literatür özetine yer verilmiştir. Ardından üçüncü bölümde, tez çalışmasında kullanılan materyaller ve tasarlanan algoritma hakkında ayrıntılı bilgi verilmiştir. Son olarak dördüncü bölümde testlerde elde edilmiş bulgulara yer verilmiş ve tartışma ve sonuç bölümünde bulgular üzerinde durulmuştur.

2. LİTERATÜR BİLGİLERİ

Telemetri sistemleri; yaban hayatın gözlemlenmesi, tıbbi ve medikal uygulamalar, uzak araçlarının izlenmesi, insansız hava ve kara araçlarının izlenmesi ve kontrolü gibi pek çok alanda kullanılmaktadır. Telemetri sistemlerinin kullanım alanlarına göre literatürde yer alan çalışmaların bir özeti verilmektedir.

Thums, M ve arkadaşları (2013) deniz kaplumbağalarının izlenmesi amacıyla telemetri sistemini kullanmışlardır. Bu çalışmada kullanılan telemetri sisteminin haberleşmesini uydu aracılığıyla gerçekleştirilmiştir. Bu sayede kaplumbağaların mesafe sorunu olmadan izlenmesi sağlanmıştır. Tucker, A. D. (2010) CarettaCaretta kaplumbağalarının yumurtlama zamanlamasının modellenenebilmesi amacıyla caretta-caretta kaplumbağalarının izlenmesi telemetri sistemi kullanılarak sağlanmıştır. Telemetri sisteminin haberleşmesi GPRS aracılığıyla sağlanmıştır. Grothues, T. M ve arkadaşları (2012) yapmış oldukları çalışmada kış kalkan balığının yumurtlama yerlerini araştırmak için telemetri sistemini kullanmışlardır. Telemetri sisteminde haberleşmeyi GPRS kullanarak sağlamışlardır. Royer, F.&Lutcavage, M. (2008) deniz canlılarını uydu haberleşme sistemiyle izleyip konumlarının belirlenebilmesi amacıyla telemetri sistemini kullanmışlardır. Bu çalışmanın asıl amacı, kendi geliştirdikleri sistemin konum bulma konusunda eski sistemlere göre daha doğru sonuçlar ortaya koyduğunu ispat edebilmektir. Telemetri sistemiyle elde ettikleri konum bilgisini eş zamanlı olarak gözlemciye aktarılmasını başarmışlar ve bu sayede geliştirdikleri sistemin daha doğru sonuçlar elde ettiğini görebilmişlerdir. Durmuş, M. (2010) telemetri sistemini, türü tükenmekte olan kursaklı ceylanların takip edilmesi amacıyla kullanmıştır. Bu kapsamda kursaklı ceylanlar bir yıl boyunca takip edilmiş ve tasmalarına kaydedilen coğrafi konum verileri kullanılarak mevsimsel habitat seçimleri ve yaşam alanı büyüklükleri belirlenmiştir. Yaptıkları çalışmada kullanılan telemetri sisteminin haberleşmesi GPS sistemiyle sağlanmıştır.

Çıbuk, M. (2009) yapmış olduğu çalışmada telemetri sistemini medikal ağlarda kullanmıştır. Telemetri sistemi sayesinde hastaların sıcaklık, pozisyon, basınç, nem, yoğunluk, atım sayısı vb. gibi niceliklerle ilgili ölçüm yapıp, elde ettiği ölçüm sonuçlarını yönetici birimlere aktarmıştır. Kullanmış olduğu telemetri sistemi

haberleşmesini RF üzerinden sağlamaktadır.

Altemose, G ve arkadaşları (2011) uydu sistemlerinde kullanılmak üzere Lityum iyon pillerin çalışma durumunu izlemek ve kontrol etmek amacıyla telemetri sistemini kullanmışlardır. Telemetri sisteminin haberleşmesi uydu üzerinden gerçekleştirilmiştir. Larrauri, J. I ve arkadaşları (2013) insansız hava araçları kullanılarak havai elektrik hattı muayesinde kullanılmak üzere telemetri sistemi tasarlamışlardır. Telemetri sisteminde RS232 seri haberleşme portu kullanılmış. Veri iletişimi GPS üzerinden gerçekleştirilmiştir. Mammadov, E.&Gueaieb, W. (2014) yaptıkları çalışmada insansız hava aracında kullanılmak üzere telemetri sistemi tasarlamışlardır. Tasarlamış oldukları telemetri sistemi gökyüzünden insansız hava aracının çekeceği fotoğrafları başka bir noktaya iletilmesi işlemini sağlamak amacıyla kullanılmıştır. Duchrow, T ve arkadaşları (2012) yaptıkları çalışmada veri toplama işlemini CAN ağı kullanarak sağlamışlardır. CAN ağından alınan veriler telemetri sistemiyle merkez istasyona aktarılmıştır. Yaptıkları bu çalışmada telemetri sisteminin haberleşmesi GPRS sistemi kullanılarak sağlanmıştır.

Hocaoğlu ve arkadaşları (2014) yapmış oldukları çalışmada araç içindeki birimlerin haberleşmesini sağlamak için CAN haberleşmesini kullanmışlardır. Telemetri sisteminde RFM12B radyo frekanslı (RF) modüller ile gerçekleştirilmiştir. Bu çalışmada telemetri sistemi iki devreden oluşmaktadır. Bunlardan biri araçta diğeri yer istasyonunda bulunmaktadır. Yer istasyonundaki telemetri devresi aldığı bilgileri bilgisayara aktararak ara yüz programı ile akü gerilimi, akımı, araç hızı gibi bilgiler yer istasyonundan takip edilebilmektedir. Üstün, Ö ve arkadaşları (2006) yapmış oldukları çalışmada araç içindeki devreleri birbirine bağlayan CAN haberleşme sistemi bulunmaktadır. Araç içinden toplanan bilgiler RF verici tarafından pit bilgisayarına aktarılmaktadır. 2.4 GHz ISM bandında çalışan alıcı vericiler, üzerinde bulunan güvenlik ayarı ile diğer RF cihazlarla etkileşime girmesi engellenmiştir. Pit bilgisayarında ara yüz olarak Visual Basic programı kullanılmıştır. Bu ara yüzde güneş panellerinden elde edilen enerji, harcanan enerji, akü şarj durumu, akü sıcaklığı gibi bilgiler görülebilmektedir. Ustun, O. ve arkadaşları (2009) yapmış oldukları diğer bir çalışma olan güneş arabası enerji yönetimi adlı çalışmalarında da telemetri sistemini

kullanmışlardır. Bu çalışmada araç içindeki birimler arasındaki haberleşme Can haberleşme sistemiyle gerçekleştirilmiştir. Telemetri sistemi iki devreden oluşmaktadır. İletişimi RF ile gerçekleştirmektedirler. Bu çalışmalarında ara yüz olarak Visual Basic programını kullanmışlardır. Connors, J. (2007) yapmış olduğu çalışmada telemetri sistemini güneş arabasının verilerini başka bir noktaya iletmek için kullanmıştır. Bu çalışmada güneş panellerinde üretilen elektrik enerjisi, bataryaların doluluk oranı gibi bilgiler RF ile başka bir noktaya iletilmektedir. Manzie, C ve arkadaşları (2012) hibrit araçlarda telemetri sistemi kullanmak suretiyle, aracın hızını ve tork bilgilerini yer istasyonunda bulunan bilgisayara aktararak, şehir yaşamında kullanılan hibrit araçların performansları hakkında bilgi sahibi olmaya çalışmışlardır. Bu kapsamda yakıt tüketimi ve egzoz emisyonu hakkında da telemetri sistemiyle bilgi toplamışlardır. Ilgın, A. H. (2015) batarya sistemli elektrikli araba projesinde, araçla alakalı verileri pilota ve yer ekibine bildirecek GPS ve RF üzerinden haberleşmesini sağlayan telemetri sistemi tasarlamışlardır. Araçta kullanılan telemetri sisteminde kullanılan GPS sayesinde aracın konumu da belirlenebilmektedir. GPS modülünden alınan tüm verilerden sadece gerekli konum bilgilerinin olduğu bölüm alınarak, bu bilgi uzak bir bilgisayara gönderilmek üzere RF modülüne iletilmektedir. Kullandığı filtreleme yöntemi ile de RF modülüne daha az bilgi gönderilerek iletişim kolaylaştırılmıştır. Doruk, A. ve arkadaşları yaptıkları çalışmada elektrikli taşıtların hız, batarya ve sıcaklık verilerinin takibi için telemetri sistemini kullanmışlardır. Veri aktarımını GSM üzerinden sağlamışlardır. Fathirad, I., Whittington, J. (2011) yapmış oldukları çalışmada telemetri sisteminin haberleşmesini GPS kullanarak gerçekleştirmişlerdir. Merkezde bulunan telemetri devresi GPS üzerinden aldığı verileri RS232 haberleşme terminali üzerinden bilgisayara aktarmaktadır. Telemetri sisteminde GPS kullanımı sayesinde veri iletimi uzak mesafelere rahatlıkla ulaştırılabilmektedir. Çakır, S. (2007) yapmış olduğu çalışmada ısı, nem, basınç gibi verilerin otomatik olarak okunması ve merkezi bir sunucuya aktarılması için RF/GPRS temelli telemetri sistemini tasarlamışlardır. Telemetri sisteminin haberleşmesinin sürekliliğine yönelik bir çalışma gerçekleştirmişlerdir. Akkurt, T. (2013) araçların kardan şaftını etkileyen torkun araştırmasında veri transferini gerçekleştirebilmek için telemetri sistemini kullanmıştır. Bu amaçla telemetri sistemi kabaca; üzerinden ölçüm alınmak istenen komponente yapıştırılmış buradan elde edilen değerler RF haberleşme kullanılarak alıcı telemetri ünitesine aktarılmıştır.

Bu tez kapsamında gerekleřtirilen telemetri kontrol algoritmasının, TBİTAK tarafından dzenlenen yarışlara katılan elektrikli araçlarda, kullanılması hedeflenmiştir. Söz konusu yarışlara ok sayıda elektrikli araç katılmakta ve araçlarda genellikle aynı taşıma frekansında yayın yapan modemler kullanılan telemetri sistemleri işletilmektedir. Bu durum araçlardaki telemetri sistemlerinin veri alışverişini engelleyebilmektedir. Bu durumun önne geebilmek için araç telemetri sisteminin taşıma frekansının boş olan bir banda kaydırılarak iletişimin devamının temin edilmesi gerekmektedir. Literatürde söz konusu taşıma frekansını deėiřtiren ve iletişimin olmadığı durumda verileri depolayan bir alıřmaya rastlanmamıştır.

Bu tez kapsamında telemetri sisteminin haberleşmesinin sürekliliėi ve veri kayıplarının önne geilmek için özėün bir telemetri kontrol algoritması tasarlanmıştır. Tasarlanan algoritma RF modem frekansının kesilmesi durumunda frekansı kaydırabilme özelliėine sahiptir. Söz konusu algoritma easypic6 deneme kartlarından oluşan test platformunda test edilmiştir. Bu testlerde tasarlanan algoritmanın, telemetri sisteminde RF haberleşmesinin kesilmesi yada aksaması durumunda RF deėiřtirerek haberleşmeye devam ettiėi görlmüřtür. Ayrıca tasarlanan algoritmanın, RF deėiřtirmesine raėmen haberleşmenin saėlanamaması durumunda, toplanan verilerin kaybolmaması için kalıcı hafıza kullanılarak burada verilerin saklanması saėlanmıştır. RF haberleşmesinin bařladıėı anda kalıcı hafıza hafızada bulunan verilerin gönderilmeye bařlandıėı test platformunda gözlemlenmiştir.

3. MATERYAL ve METOT

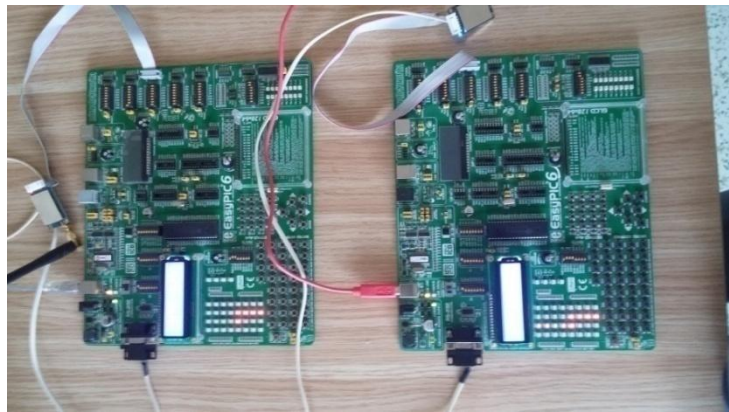
Bu bölümde, tez kapsamında tasarlanan telemetry kontrol algoritmasına ilişkin bilgiler verilmektedir. Ayrıca algoritmanın işletildiği test platformu bileşenlerine ilişkin ayrıntılı bilgilerde sunulmaktadır. Aşağıda ilk olarak materyal kısmında tez kapsamında kullanılan 16F877A Mikro denetleyici tabanlı telemetry geliştirme bordu, 16F877A mikro denetleyicisi, RF modem ve RF spektrum analizörü hakkında ayrı başlıklar altında bilgiler verilmiştir. Ardından metot kısmında tasarlanan telemetry kartları, kontrol algoritması ve kullanılan program hakkında bilgiler verilmiştir.

3.1 Materyal

Bu bölümde mikro denetleyici tabanlı telemetry geliştirme bordu, 16F877A mikro denetleyicisi, RF modem ve RF spektrum analizörü hakkında bilgiler verilecektir.

3.1.1 Mikro Denetleyici Tabanlı Telemetry Geliştirme Bordu

Tasarlanan algoritmanın test edilmesi için mikro denetleyici tabanlı telemetry geliştirme bordu kullanılmıştır. Geliştirme bordu kullanılmasıdaki amaç, hazır oluşu ve bu nedenle devre tasarımı ve yapımı aşamaları için zaman harcamadan algoritmanın çalışıp çalışmadığını test edebilme imkânından yararlanmaktır. Kart üzerinde mikro denetleyici ile bir dizi devre elemanlarıyla donanım ara yüzü oluşturulmuştur. Buradan edinilen bilgi ve tecrübeyle telemetry kart tasarımı gerçekleştirilmiştir.



Resim 3.1 Mikro denetleyici tabanlı telemetry geliştirme bordu.

3.1.2 PIC16F877A Mikro Denetleyicisi

Geliştirme kartında ve tasarlanan telemetri kartında PIC16F877A mikro denetleyicisi kullanılmıştır. PIC16F877A yüksek performanslı 8 bit mikro denetleyicidir. Tüm PIC 16/17 serisi mikro denetleyicilerinde olduğu gibi PIC16F877A mimarisi de RISC sınıfında yer almaktadır. PIC mikro denetleyicilerinin enbüyük özelliği uyku özelliğidir. Bu özellik sayesinde işlem yapılmadığı durumlarda PIC uyuma durumuna geçerek çok düşük güç harcar. PIC16F877A flash program belleği, bir ürünün prototip üretimi ve seri üretimi için aynı mikro denetleyicinin kullanılmasına olanak vermektedir. PIC16F877A mikro denetleyicisinde bulunan bazı özel donanımlar ve özellikleri şunlardır;

Çizelge 3.1 16F877A Mikro denetleyicisinin özellikleri.

Uygulama frekansı (Operating Frequency)	DC-20 Mhz
Flash Program Hafızası (Flash Program Memory)	8 K Byte
Bilgi Hafızası (Data Memory)	368 Byte
EEPROM Hafızası	256 Byte
Kesme Sayısı	14
Giriş-Çıkış Port Sayısı	Port A,B,C,D
Zamanlayıcı Sayıları (Timers)	3
Yakala Karşılaştır-PWM Modülü	2
Seri İletişimler	MSSP, USART
Paralel İletişimler	PSP
10 Bit Analog Dijital Çevirici	8 Kanal
Komut Sayıları	35

PIC16F877A mikro denetleyicisinin yukarıda özetlenen özel donanımlarından bazıları telemetri sisteminin tasarımında kullanılmıştır. Bu donanımlar sırasıyla; zamanlayıcı sayıcı birimlerinden biri olan (Timer 1), analog dijital dönüştürücü birimi ve seri haberleşme birimidir (RS232). Aşağıda bu üç birimin özellikleri ayrı başlıklar altında verilmiştir.

3.1.2.1 SayıcıZamanlayıcı Birimi

Zamanlayıcı1 birimi zamanlayıcı ve sayıcı olarak kullanılabilen bir donanım birimidir. Bu birimde 16 bitlik zamanlayıcı-sayıcı kullanım özelliği vardır. Zamanlayıcı1 de

okuma ve yazma işlemleri yapılabilir. Dahili ve harici saat sinyali kaynak seçeneklerinden birini kullanma imkanı bulunmaktadır. Zamanlayıcı1 değeri FFFFh'dan 0000h'a (0-65535) geçişinde taşma kesmesi vardır. Zamanlayıcı1 birimi içinde iki adet 8 bitlik kaydedici bulundurur. Bunlar TMR1H ve TMR1L kaydedicileridir. Zamanlayıcı1 ile 0 – 65535 (0000h-FFFFh) arasında istenen sayıdan başlanarak sayma işlemi yapılabilir. Zamanlayıcı1 birimi üç değişik modda kullanılabilir. Bunlar senkron zamanlayıcı, senkron sayıcı, asenkron sayıcıdır. Zamanlayıcı modunda zamanlayıcı1 her komut çevriminde bir artar. Sayıcı modunda ise zamanlayıcı1, RC0/T1OSO/T1CKI pininden gelen saat sinyalinin her yükselen kenarında bir artar. Zamanlayıcı1 aynı zamanda dahili bir sıfırlama kaynağına sahiptir. Bu sıfırlama sinyali CCP modülünden gelen sıfırlama sinyalidir. CCP modülünden gelen sinyal ile zamanlayıcı1 içeriği sıfırlanabilir. Diğer sıfırlama metotları TMR1H ve TMR2H kaydedici içeriklerini sıfırlamaz. Zamanlayıcı1 özelliklerini kontrol eden kaydedici T1CON kaydedicisidir.

3.1.2.2 Analog-Dijital Çevirici

Doğada doğal olarak bulunan tüm sistemler analog düzendedir. Kullandığımız mikro denetleyiciler ise dijital sistemlerdir. Bu nedenle dış dünya ile iletişim kurmak ve dış dünyadan veri almak-algılama yapmak için bu iki sistemin bir şekilde birbirini anlaması gereklidir. Bu iş içinde doğada bulunan analog sinyallerin dijital sistemlerin anlayacağı dijital sinyallere çevrilmesi gereklidir. Bu işlem için analog dijital çevirici devreleri ve entegreleri kullanılmaktadır. PIC16F877A mikro denetleyicisinde analog dijital çevirici modülü bulunmaktadır. Böylece harici bir devre veya entegre kullanmadan analog sinyaller dahili analog dijital çevirici modülü sayesinde dijital sinyallere çevrilebilir. PIC16F877A mikro denetleyicisinde sekiz adet analog dijital giriş ucu (A0, A1, A2, A3, A5, E0, E1, E2) vardır. Analog dijital çevirici için gerekli referans voltajı Vdd, Vss, A2 ve A3 uçları ve bunların kombinasyonlarından seçilerek elde edilir. Analog dijital çevirici birimi dört adet kaydediciye sahiptir. Analog dijital çevirici modülünün çevrim sonucunun yazıldığı ADRESH ve ADRESL ile analog dijital çevirici modülü birimi kontrol kaydedicileri ADCON0 ve ADCON1'dir.

3.1.2.3 Seri Haberleşme Birimi (RS232)

Bilgisayarlarla iletişim kurabilmek için bir takım yöntemler vardır. Bunlardan biri seri haberleşmedir. Seri haberleşmede bilgiler tek bir kanal üzerinden gönderilip, tek bir kanal üzerinden alınır. Bu sebeple seri iletişim daha yavaş olmasına rağmen, hem daha ucuz hem de uzun mesafelere iletim için uygundur. Seri haberleşmenin iki yöntemi vardır. Bunlardan ilki senkron seri iletişim yöntemi. Bu yöntemde gönderilen veri bitleri ile alınan veri bitleri arasında bir uyum olmalıdır. Şöyle ki senkron ifadesi, alıcı devre ile verici devrenin eş zamanlı çalışması anlamına gelir. Bu alış veriş arasındaki uyumu sağlamak için data kablosu yanında bir de saat sinyali kablosu bulunur. Saat sinyalinin periyodu seri iletişimdeki her bir birimin iletim süresini belirtir. Böylece bilgi iletişimi için başlangıç ve bitiş bitlerine gerek kalmaz. Senkron seri iletişim asenkron seri iletişime göre daha pahalıdır. Bir diğer seri iletişim yöntemi olan asenkron seri iletişim de ise sadece veri hattı bulunur. Alıcı ve verici devreler veri iletişiminde eş zamanlı çalışmazlar. Senkron iletişimdeki saat palsi yerine asenkron iletişimde başlangıç biti, bitiş biti ve eşlik biti vardır. Başlangıç biti, bilginin gönderilmeye başlandığını alıcı tarafa bildirmek için kullanılır. Asenkron iletişimin ilk biti her zaman başlangıç bitidir. Asenkron veri iletişiminde başlangıç biti her zaman lojik 0 olmalıdır. Bitiş biti, gönderilen bilginin bittiğini alıcı tarafa belirtir. Bitiş bitinden sonra gönderilen bilgi yeniden başlangıç biti ile başlar. Eşlik biti, alıcı tarafa gönderilen karakterin doğru olarak iletilip iletilmediğini kontrol etmek için kullanılan bittir. Asenkron iletişimde ODD (tek), EVEN (çift), MARK (işaret), SPACE (boşluk), veya NONE (eşlik biti yok) eşlik bitleri kullanılır. Eşlik bitinin asenkron iletişimde kullanılması zorunlu değildir.

3.1.3 RF Modem

RF modemler iki veya daha fazla cihaz arasında radyo frekansı ile haberleşme sağlamak amacıyla kullanılan cihazlardır. Gerçekleştirilen çalışmada RF modem olarak NiceRF firmasının “SNR611” model RF modemleri kullanılmıştır. Bu RF modemler veri gönderme veya alma durumunda işletilmekte ve 100 mW gönderme gücüne sahiptir. RF modemlerin taşıma frekansı 433/470/868/915 MHz taşıma frekansına ayarlanabilmektedir. Ayrıca modemlerin taşıma frekansı, merkez frekansın etrafında 1

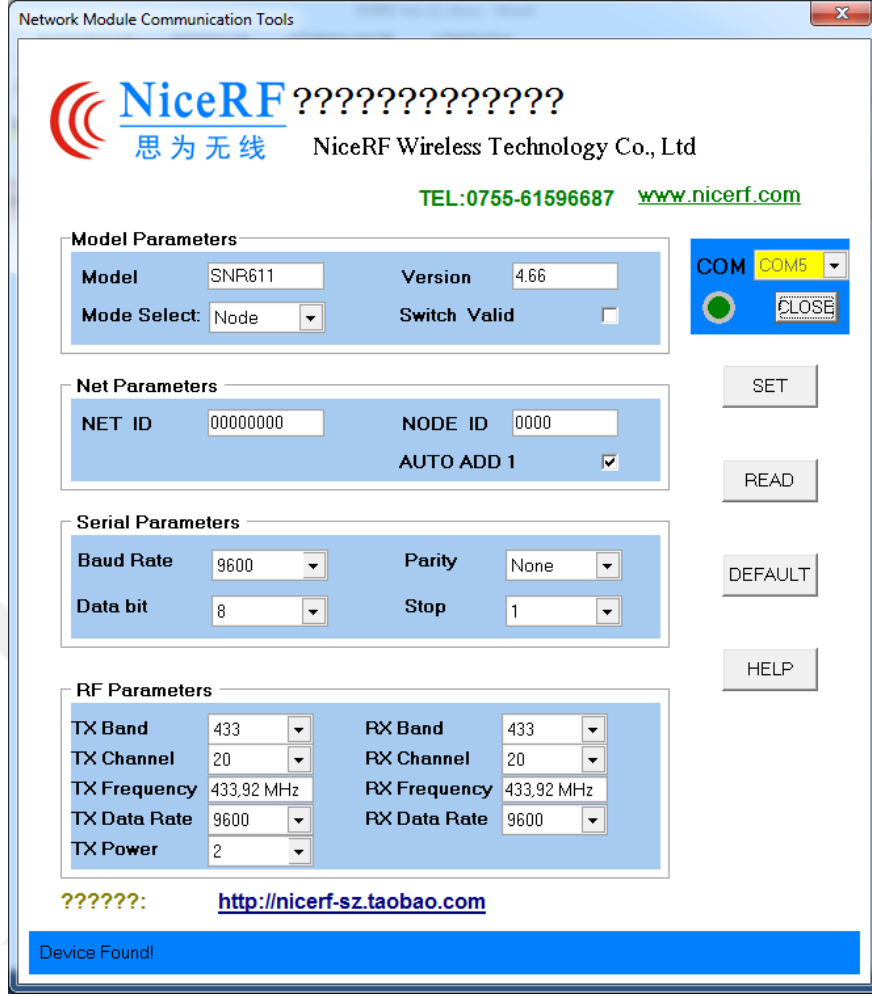
MHz aralıklarla 40 farklı kanala kaydırılabilmektedir (Örneğin 433 MHz taşıma frekansı için 414,92 ile 453,92 MHz arasında 40 farklı kanal). RF modemlerde haberleşme güvenliğini artırmak için 4 bayt NET-ID, 2 bayt NODE-ID tanımlamaları da yapılabilmektedir. Bundan başka RF haberleşme ve seri haberleşme hızları da 1200 ile 115200 bps aralığında değiştirilebilmektedir.



Resim 3.2 SNR611 RF modem.

RF modemin yapılandırma işlemleri bilgisayardan veya özel mikro denetleyici sistemi üzerinden gerçekleştirilebilmektedir. Aşağıda ilk olarak bilgisayar üzerinden yapılandırma ayarlarının gerçekleştirildiği ara yüz programı ayrıntıları verilmekte ardından mikro denetleyiciyle yapılandırma işlemlerinde kullanılan modem protokolleri verilmektedir.

RF modemi bilgisayardan yapılandırma işlemi NiceRF firmasının bir ara yüz programı ile gerçekleştirilmektedir (Resim 3.2). RF modemi bilgisayara bağlamak için bir USB-RS232 dönüştürücü kullanılmaktadır. RF modemin; taşıma frekansı, kanal numarası, RF veri iletişim hızı, iletim gücü seviyesi, seri haberleşme hızı, veri biti sayısı, eşlik (parity) biti, NET-ID ve NODE-ID parametreleri söz konusu ara yüz üzerinden değiştirilebilmektedir. RF modem parametrelerinde değişikliğin gerçekleşebilmesi için ara yüz programında bulunan SET butonuna basılmalıdır. RF modemde fabrika ayarlarına dönmek istiyorsak ara yüz programında bulunan DEFAULT butonuna basmak yeterlidir. Aşağıda NiceRF SNR611 RF modeminin ara yüz programı görülmektedir.



Resim 3.3 SNR 611 RF modem ara yüz programı.

RF modemın mikro denetleyici ile yapılandırılmasında kullanılan protokol oldukça basit ve sade bir formdadır. RF modem yapılandırma protokolünde aşağıdaki yapı kullanılmaktadır.

AA FA + [Komut] + Parametre

Burada; Komut 1 Byte uzunluğunda onaltılık sayıyı temsil etmektedir. Parametre ise seçilen komuta göre 0 veya 18 Byte uzunluğunda onaltılık sayıyı göstermektedir. RF modem protokolünde kullanılan toplam 4 komut vardır. Bunlar sırasıyla; sürüm sorgulama komutu “AA”, okuma komutu “01”, fabrika ayarına dönme komutu “02” ve parametre ayar komutu “03” şeklindedir.

Sürüm sorgulama komutu: RF modemın ürün adı ve sürüm numarasını görebilmek için

sürüm sorgulama komutu ‘AA FA AA’ yazılması yeterlidir. Bu işlemin gerçekleştiğine dair dönüş mesajı ‘SNR611_Ver2.62 \r \n’ İletisinin görülmesi yeterli olmaktadır.

Okuma komutu: RF modem parametrelerini görebilmek için okuma komutu olan ‘AA FA 01’ yazılması yeterli olmaktadır. Komut yazıldığında RF gönderim kanalı, RF gönderim bandı, RF TX oranı, RF gönderim gücü, RF alma kanalı, RF alma bandı, RF RX oranı, seri bağlantı noktasının baud hızı, veri bitleri, bitiş biti, parity, NED-ID, NODE-ID, çalışma modu parametre değerleri görülmektedir. Aşağıdaki tabloda RF modem parametreleri görülmektedir.

Çizelge 3.2 RF Modem parametreleri tablosu.

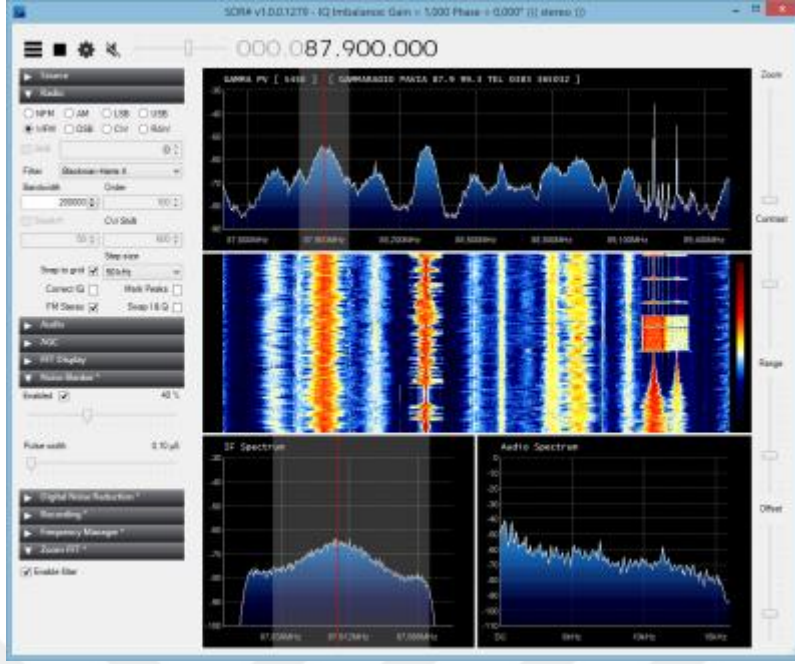
1	0	0	0	14	01	03	0	0	0	0	0	0	0	0	00	00	0	0
4	1	3	7				3	2	1	1	0	0	0	0			d	a
RF gönderme parametreleri				RF alma parametreleri		Seri iletişim parametreleri		NED-ID		NODE-ID		r / n						

Fabrika ayarına dönme komutu: RF modem fabrika ayarlarına dönebilmek için ‘AA FA 02’ komutu yazılması yeterlidir. Bu komut yazıldığında dönüş mesajı olarak "Tamam \r \n" veya "HATA \r \n" geliyor.

Parametre ayar komutu: RF modem parametre değerlerinde değişiklik yapabilmek için parametre ayar komutu olan ‘AA FA 03’ yazılması yeterlidir. Yapılan değişikliklerin gerçekleşip gerçekleşmediğine dair geri dönüş olarak "OK \r \n" veya "ERROR \r \n" bilgisi görülmektedir.

3.1.4 RF Spektrum Analizörü

SDR AIRSPY spektrum analizörü 48.5-863.25 MHz frekans aralığında çalışmakta bant genişliği 6/7/8 MHz aralığında bulunmakta SKY, DSY, FM ve YTR işlevleri bulunmaktadır. Bu özellikleri sayesinde karasal televizyon yayıncılığı, dijital ses yayınlarını takip edebilmeye olanak sağlamaktadır.



Resim 3.4 SDR AIRSPY spektrum analizörü.

3.2 Metot

Bu bölümde tez kapsamında tasarlanan telemetri kartları, tasarlanan telemetri kontrol algoritması ve yer istasyonu ve veri toplama kartları için tasarlanmış olan mikro denetleyici programları anlatılacaktır.

3.2.1 Tasarlanan Telemetri Kartları

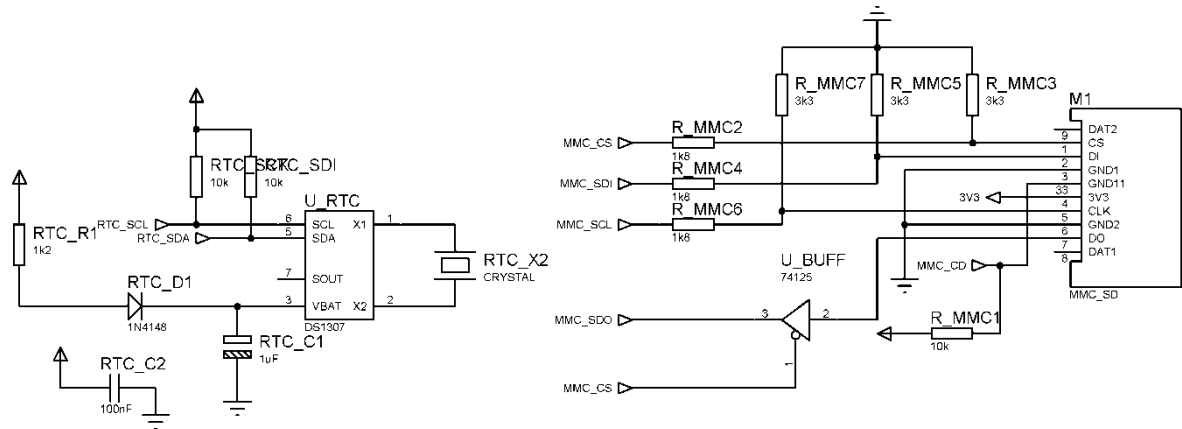
Telemetri kartlarının tasarımından önce, ön hazırlık amacıyla mikro denetleyici tabanlı telemetri geliştirme bordunda tasarlanan algoritmanın çalışıp çalışmadığının kontrolü gerçekleştirilmiştir. Buradan elde ettiğimiz tecrübeyle telemetri kart tasarımını istediğimiz ölçüde gerçekleştirebildik.

Tasarlanan telemetri kartları iki ayrı devreden oluşmaktadır. Bunlar veri toplama birimi ve yer istasyonu birimidir. Yer istasyonu birimi ve veri toplama birimi devreleri için PIC16F877A mikro denetleyicisi kullanılmıştır. Telemetri kartlarında zamanlamanın eş zamanlı olabilmesi için gerçek zamanlı saat kullanılmıştır. İletişimin kesildiği durumda gönderilemeyen verilerin depolanması için flash bellek (MMC) kullanılmıştır. Tasarımı

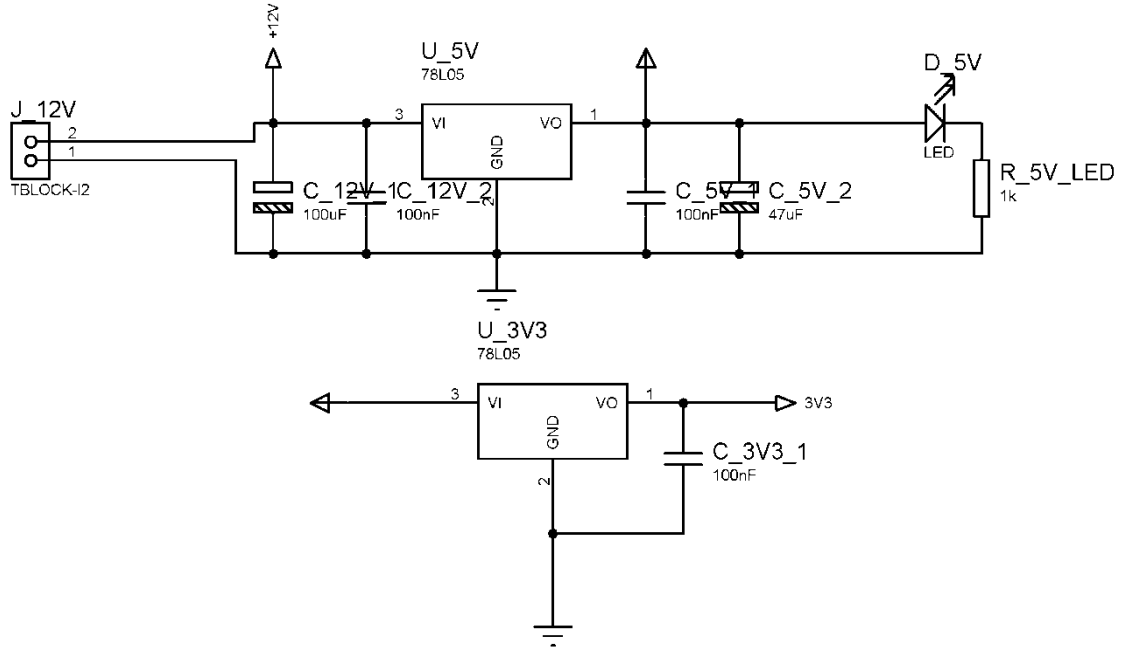
gerçekleştirilen kartta seri haberleşme portu yer almaktadır. Analog giriş ve USB port girişleri de bulunmaktadır. SNR611 RF modülü ve yapılandırma anahtarları (DIP switch) bağlantıları her iki devrede de aynı şekilde tasarımı gerçekleştirilmiştir. Veri iletişiminden haberdar olabilmek için 2x16 LCD Display tasarlanan her iki kartta da kullanılmıştır. Bu sayede veri sayısı, hatalı veri sayısı, RF modem frekansı hakkındaki bilgiler LCD Display de görülmektedir.

Gerçek zamanlı saat (RTC): Aslında birer mikro kontrolcü gibi içlerinde işlem birimi, ram, register ve veri yolu arabirimi vardır. “Saat” gibi çalışması için 32.768 khz hızında bir kristal osilatör kullanılır. Bu frekans saatin doğru çalışması için çok önemlidir. DS1307 gibi RTC’ ler I2C seri veri yolu ile mikroişlemciler veya mikro kontrolcülerle iletişime geçer. Bir sıra halinde verileri gönderir. Bu sıraya göre hangi ay, hangi yıl, hangi gün, hangi saat olduğu mikro kontrolcüler tarafından yorumlanıp kullanılmak üzere kaydedilir. İlk değer kaydedileceği zaman da mikro kontrolcü tarafından veriler RTC’nin içerisine yazılır. İlk değeri doğru bir şekilde verdiğiniz zaman yüksek hassasiyetle uzun süre (hatta yıllarca) RTC’nizi çalıştırabilirsiniz. Çünkü çok düşük güç tüketir.

Flash bellek(MMC): Depolama aygıtlarıdır.



Şekil 3.1 Telemetri kartlarının besleme, SDI kart ve gerçek zamanlı saat devre şemaları.



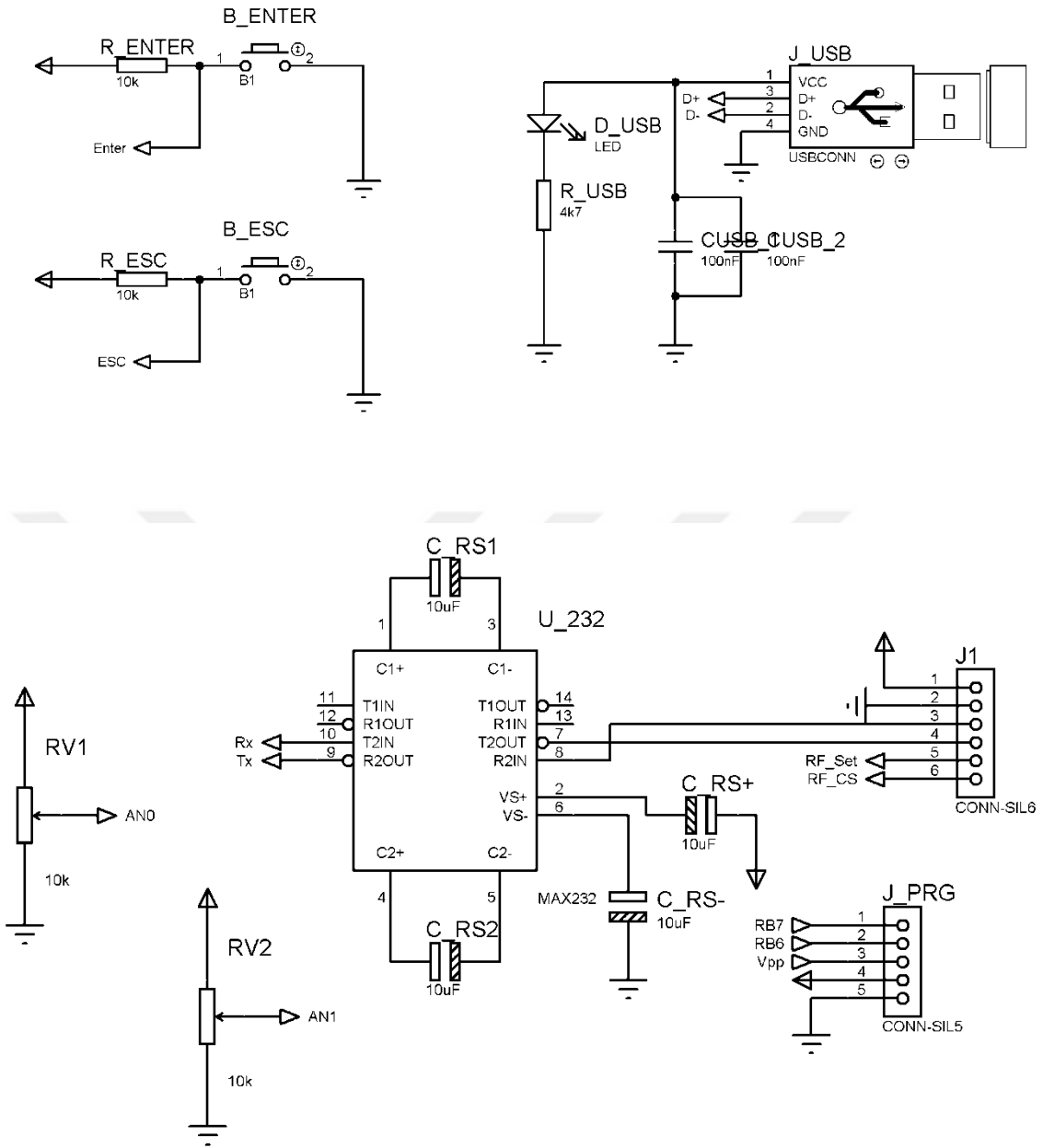
Şekil 3.1 (Devam) Telemetri kartlarının besleme, SDI kart ve gerçek zamanlı saat devre şemaları.

Seri Haberleşme Portu (RS232): Seri haberleşme portunun yapımında RS232 entegresi 4 adet kondansatör kullanılmıştır.

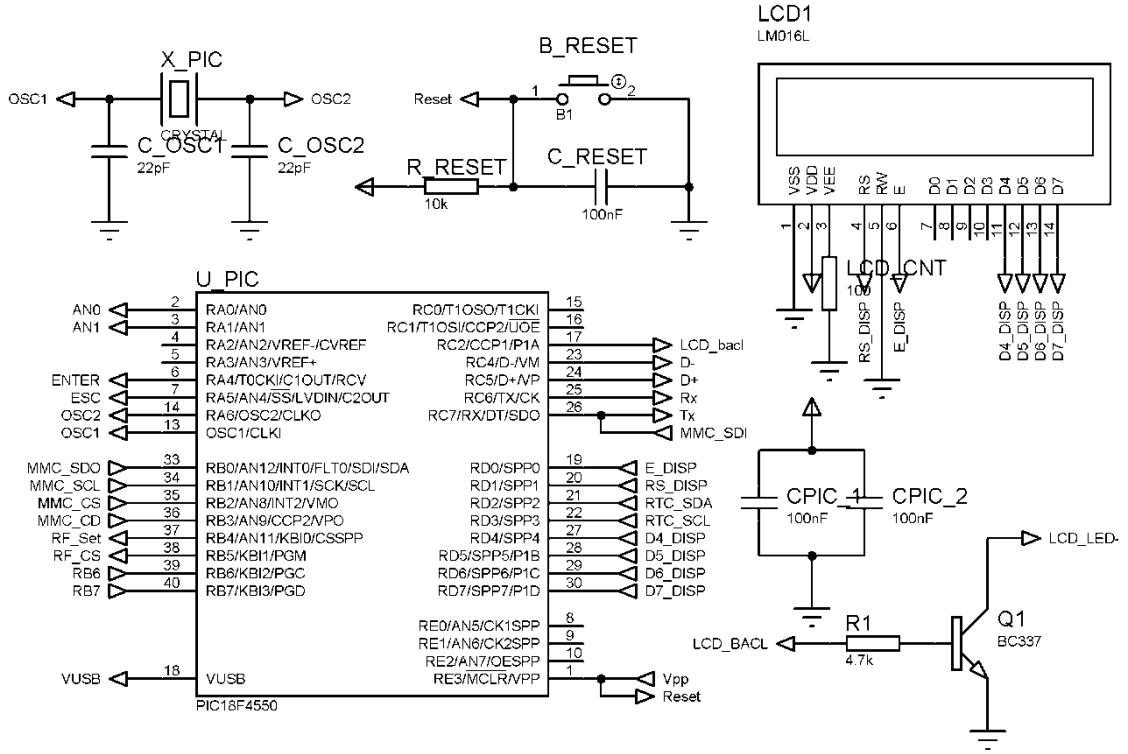
Analog giriş: Tasarlanan telemetri kartların da analog giriş için RV1 ve RV2 potansiyometreleri kullanılmıştır. Bu potansiyometreler sayesinde değişken analog sinyal üretilmiştir.

Evrensel seri veri yolu (USB): USB port deneylerde kullanılmamıştır. Ancak tasarlanan telemetri kartının bilgisayara bağlanabilme olasılığına karşı tasarlanan devrede yer almıştır.

*2*16 LCD Display:* LCD Display tasarlanan telemetri devresinde veri transferinin gerçekleşip gerçekleşmediğini, gönderilen veri sayısını, frekans kaydırma işleminin gerçekleşmesini görebilmeye imkan sağladığı için kullanılmıştır.

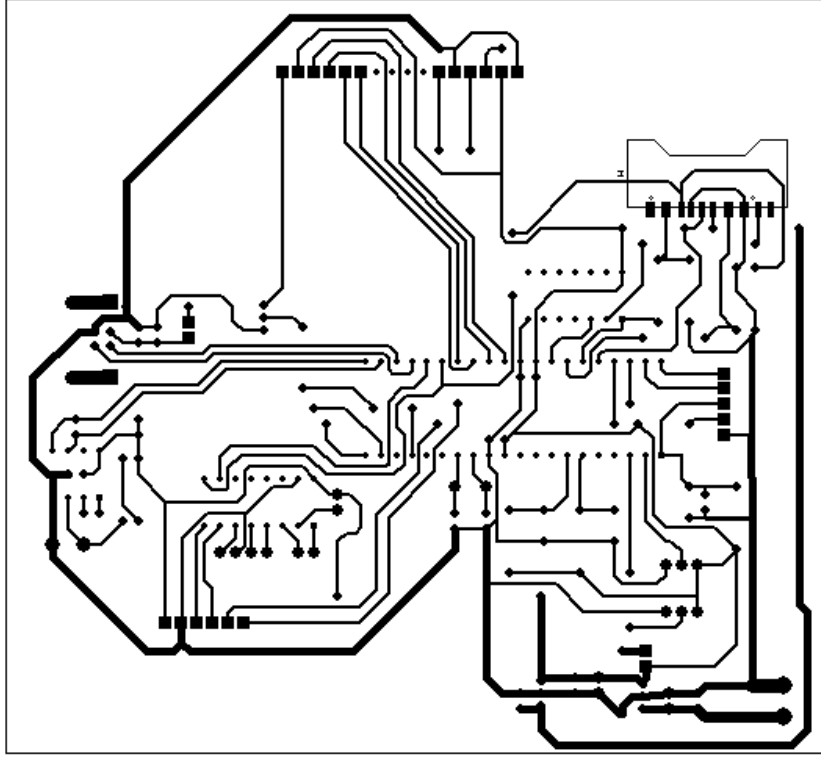


Şekil 3.2 Telemetry haberleşme portları.

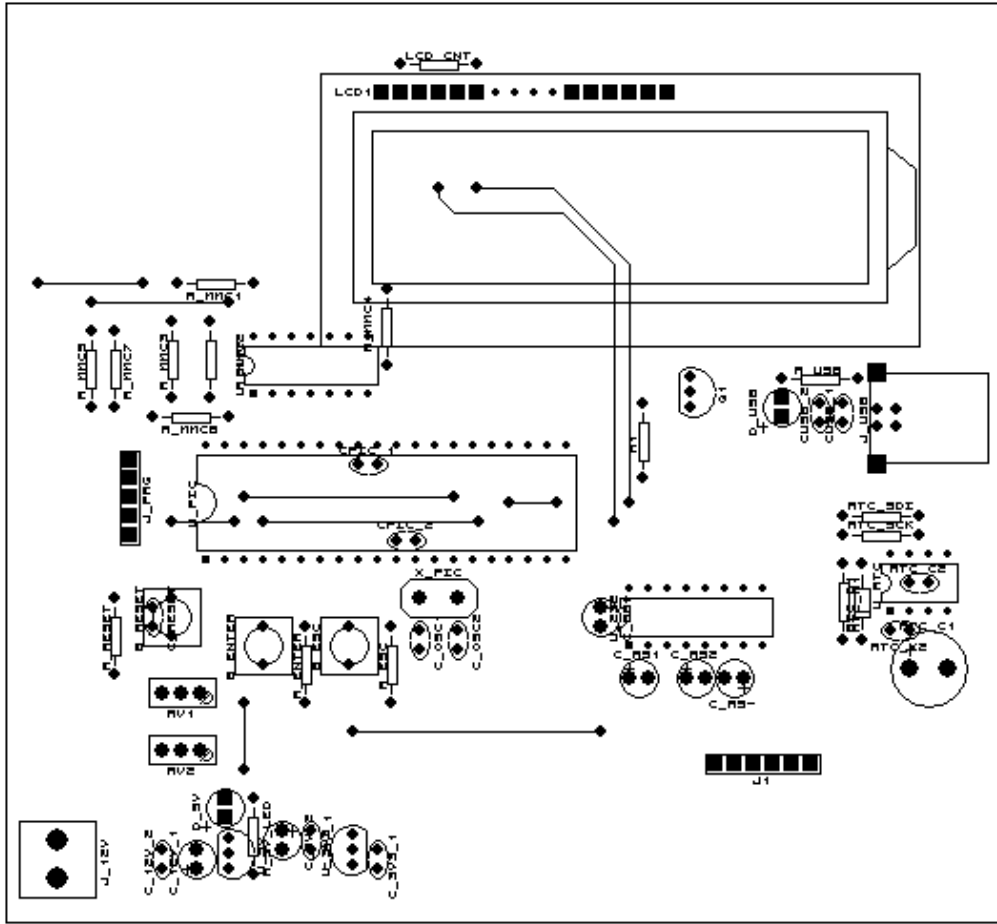


Şekil 3.3 Telemetry ana kart şeması.

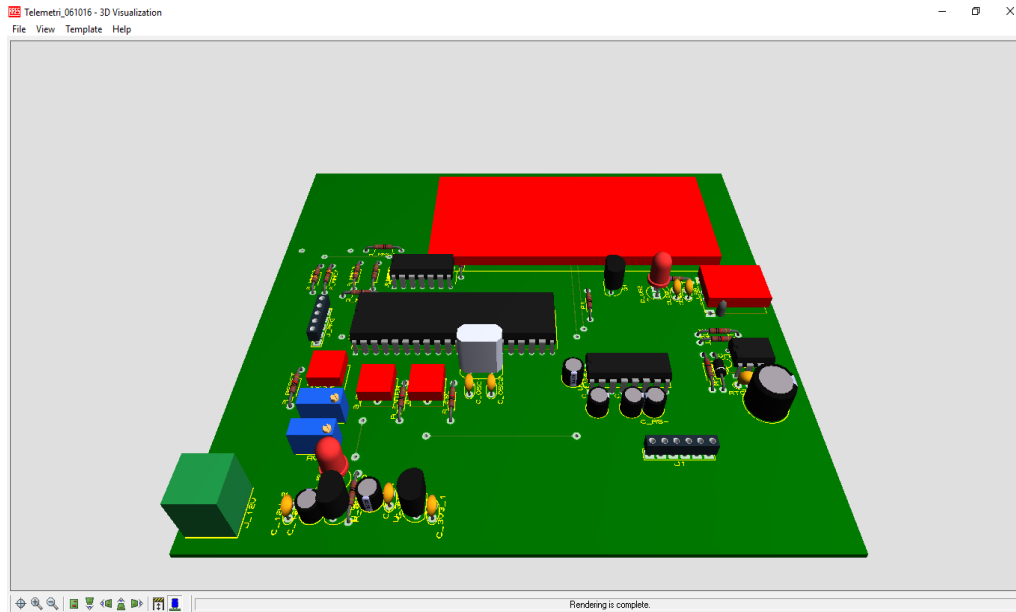
Devrelerin tasarımında ve baskı devre aşamasında ares programı kullanılmıştır. Bu program kullanılarak tasarımı gerçekleştirilen telemetry kartlarının baskı devre aşamasındaki alt bakır yüzeyi, malzeme montaj yüzeyi ve üç boyutlu görüntüleri elde edilmiştir. Aşağıda bu görüntüler yer almaktadır.



Şekil 3.4 Tasarlanan telemetri kartlarının bakır yüzeyi.



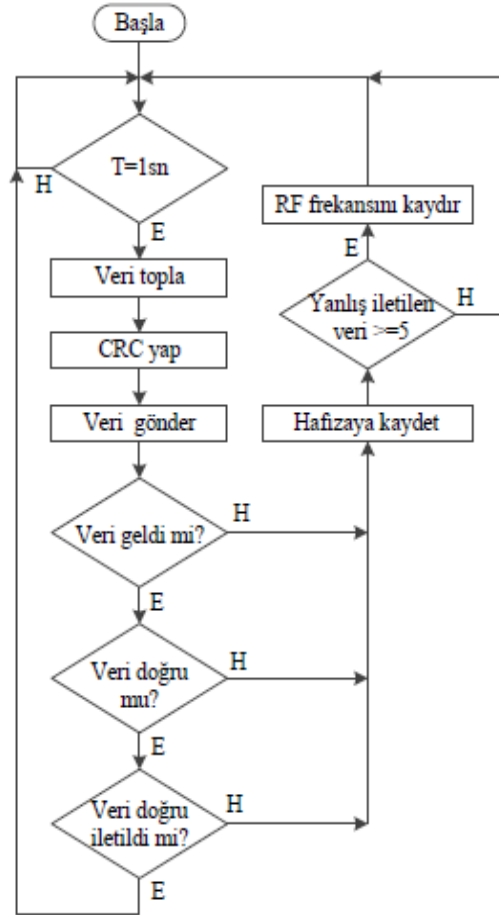
Şekil 3.5 Tasarlanan telemetry kartlarının malzeme montaj yüzeyi.



Şekil 3.6 Tasarlanan telemetry kartlarının üç boyutlu görünümü.

3.2.1.1 Tasarlanan Telemetry Kontrol Algoritması

Bu bölümde iletişim kesildiği anda RF değiştiren ve iletişim olmadığı durumda verileri depolayan telemetry kontrol algoritması sunulmaktadır. Aşağıda tasarlanan telemetry kontrol algoritmasının akış şeması görülmektedir (Şekil 3.7).



Şekil 3.7 Çalışmada kullanılan algoritmanın akış şeması.

İlk olarak Algoritmada veri toplama süreci işletilmektedir. Bu veriler, RF modem yapılandırma parametreleri (Gönderme ve alma frekansı, gönderme kanalı, gönderme hızı ile gönderme gücü olmak üzere toplam 7 Byte), AN0 analog giriş verisi (10 bit/2 Byte) ve PORTD dijital verisi (8 bit/1 Byte) olarak toplamda 9 Bayt şeklinde sıralanabilmektedir. Ardından toplanan bu 9 Bayt veri üzerine 1 Bayt CRC (Cyclic Redundancy Check) kodu eklenerek 10 Bayt veri paketi oluşturulmaktadır. CRC kodu alıcı tarafta verilerin doğru gelip gelmediğini kontrolünü gerçekleştirmek için

kullanılmaktadır. Bu işlemin ardından paketlenen 10 Bayt veri RF modemine seri port üzerinden yazılmakta ve modem yazılan veriyi RF üzerinden alıcı modemine iletmektedir. Veri gönderme işlemi bittikten sonra algoritma karşı taraftan (yer istasyonundan) gelen veriyi beklemektedir. Karşı taraftan gelen veri, gönderilen verinin yer istasyonuna doğru ulaşmış olup olmadığıyla ilgili bilginin yanı sıra yer istasyonu RFmodemine ait yapılandırma parametrelerini de içermektedir. Veri geldiğinde öncelikle gelen verinin doğruluğu CRC kodu üzerinden gerçekleştirilmektedir. Eğer verilerde bozulma yoksa verinin yer istasyonuna doğru ulaşmış olup olmadığı kontrol edilmektedir. Gönderilen veri yer istasyonuna doğru biçimde iletildi ise, bir sonraki veri toplama süreci için veri toplama çevrim zamanının dolması beklenmektedir. Gelen veri üzerinde yapılan CRC kontrolünde verilerde bozulma tespit edildi ise, bu durum RF haberleşme ortamında problem olduğu anlamında gelmektedir. Bu durumda veri hafızaya kaydedilmekte, sonrasında RF modemin kontrolü yapılmaktadır. RF modem çalışıyorsa taşıyıcı frekansına yakın bir frekansta bozucu sinyal olduğu ve haberleşmeyi olumsuz etkilediği anlamında gelmektedir. Bu problemi aşabilmek için taşıma frekansının sonraki kanala kaydırılma işlemi gerçekleştirilmektedir. Algoritmanın bu özelliği sayesinde taşıma frekansı boş bir banda kaydırılabilmektedir. Gelen veride bozulma yoksa ancak veri karşı tarafa doğru iletilmedi ise, söz konusu veri paketi hafızaya kaydedilmektedir. Doğru iletilemeyen veri paketi sayısı beşe eşitse haberleşmede sorun olduğu sonucu ortaya çıkmaktadır. Bu halde taşıma frekansı bir sonraki kanala kaydırılmaktadır.

Telemetri sistemi için oluşturulan frekans kaydırma algoritması 6 adımda işletilmektedir. Telemetri sistemi bu çalışma için 433 MHz merkez taşıma frekansında başlatılmaktadır. Haberleşmenin kesintiye uğraması durumunda, taşıma frekansı ilk olarak merkez frekanstan ayarlanan adım miktarı kadar ileriye taşınmaktadır. Böylece haberleşmenin devamı sağlanmaktadır. Bu çalışma için kanaladım miktarı 5 MHz olarak ayarlanmıştır. Taşıma frekansının bir adım ileriye taşınmasının ardından iletişim sağlanamadı veya bir süre sonra iletişim kesintiye uğradıysa, taşıma frekansı son ayarlanan değerinden bir adım daha ileriye kaydırılmaktadır. Bu frekans kaydırma işlemi sonrasında da bir önceki adımda karşılaşılan iletişim sorunu devam ediyorsa, taşıma frekansı merkez taşıyıcı frekansı olan 433 MHz değerine kaydırılmaktadır.

Merkez taşıma frekansına kaydırılan telemetri sisteminde haberleşmenin tekrar kesintiye uğraması durumunda, bu kez taşıma frekansı adım miktarı kadar geriye kaydırılmaktadır. Benzer şekilde haberleşme kesilirse taşıma frekansı adım miktarı kadar geriye kaydırılmaktadır. Son olarak ayarlanan taşıma frekansında haberleşmede sorun olması durumunda, taşıma frekansı tekrar merkez frekansına kaydırılmaktadır. Görüldüğü gibi frekans kaydırma işlemi, merkez taşıma frekansı etrafında bir tarama şeklinde gerçekleştirilmektedir. Bu sayede frekans kaydırma işleminin taşıma frekansını merkez frekansının çok uzağına kaydırmasının önüne geçilmiştir. Yer istasyonu veya veri toplama birimlerine ait RF modemlerinde taşıma frekansı kaydırıldığında, karşı taraftaki modemde bir sonraki çevrimde taşıma frekansını kaydırmakta, böylece her iki modemin de aynı taşıma frekansında çalışması temin edilebilmektedir. Burada frekansını ilk kaydıran modem tekrar frekans kaydırma sürecini işletmek için iki çevrim beklemekte böylece modemin sürekli frekans kaydırmasının önüne geçilmiş olunmaktadır.

3.2.1.2 Mikro Denetleyicide Kullanılan Program

Tasarlanan telemetri kartlarında kullanılan 18F4550 mikro denetleyicilerini programlamak için CC C programı kullanılmıştır. Programlama dili olarak C dili tercih edilmiştir. Aşağıda 18F4550 mikro denetleyicisinde kullanılan programın bir kısmı yer almaktadır. Aşağıda yer alan program veri toplama modülünde kullanılmaktadır. Programda veri toplama biriminin 1 saniyede bir veri toplama işlemi yapabilmesi için gerekli yazılım görülmektedir.

```
#int_TIMER1
void Timer1_isr(void){
    s_1s++;
    //if(s_1s>=100){
    if(s_1s>=100){
        s_1s=0;
        b_1s=True;
        b_1s_bck_light=True;
    }
    set_timer1(55537);
}
```

Aşağıdaki programda verilerin doğru alınıp alınmadığını kontrol etmek amacıyla CRC yapılmaktadır.

```
int1 Veri_kontrol(int8 d_Rx_temp[20]){
    int8 s_Temp_Rx=0;//s_Temp_serial=0,
    int16 CRC_Temp=0;
}
    for(s_Temp_Rx=0;s_Temp_Rx<10;s_Temp_Rx++){
        d_Rx[s_Temp_Rx]=d_Rx_temp[s_Temp_Rx];
        CRC_Temp+=d_Rx[s_Temp_Rx];
    }
    CRC_Temp%=10;
    d_Rx[10]=d_Serial[10];
    if(d_Rx[10]==CRC_Temp)
        return True;
    else
        returnFalse;
}
```

Bir saniyede bir toplanan verilerin gönderme işlemi için yazılan kod görülmektedir.

```
voidVeri_gonder(){
    int8 s_Temp=0;
    disable_interrupts(GLOBAL);
    //printf("00");
    for(s_Temp=0;s_Temp<11;s_Temp++)
        putc(d_Tx[s_Temp]);
    enable_interrupts(GLOBAL);
}
```

Gönderilen verinin doğru ve eksiksiz bir şekilde ulaşp ulaşmadığını kontrol edebilmemize olanak sağlayan kod görülmektedir.

```
    if(Veri_kontrol(d_Serial)){
        if(d_Rx[7]==1){//Veri merkeze doğru ulaştı mı?
            s_Send_true++;
            b_Merkeze_iletildi=True;
            if(b_Gonderilemeyen)s_Send_gonderilemeyen++;
            if(s_Send_gonderilemeyen==read_eeprom(101))b_Gonderilemeyen=False;
        }else{
            s_Send_false++;
            b_Merkeze_iletildi=False;
            //printf("Verikontrol OK!\n");
        }
        b_Paket_hatali=False;
    }else {
        b_Paket_hatali=True;
        //printf("Veri hatali!\n");
    }
}
```

Gönderilemeyen veri yoksa bir saniyede bir veri toplama işlemi gerçekleşmekte ve toplanan veri gönderilmektedir. Bu işlem sürekli devam etmektedir. Buna ait kod görülmektedir.

```
        if(!b_Gonderilemeyen){//Gönderilemeyen veri yok
        if(s_Gonderme_hizi>=c_Normal_gonder){//Toplanan veriyi gönder
            b_Hizli_gonder=False;
            Veri_topla();
            b_DAQ=True;
            d_Tx[8]=1;
            d_Tx[10]=make_CRC(d_Tx);
            Veri_gonder();
            s_Send++;
            s_Gonderme_hizi=0;
            s_Indis++;
        }
    }
```

Eğer gönderilemeyen veri varsa bu veriler kalıcı hafızaya kaydedilmekte ve RF haberleşme başladığı anda gönderilmektedir. Bu işleme ait kod aşağıda görülmektedir.

```
else{//Gönderilemeyen veri var
    if(s_Gonderme_hizi>=c_Hizli_gonder){//Bellkteki gönderilemeyen veriyi
    gönder
        s_Hizli_gonder++;
        if(s_Hizli_gonder!=6){
            b_Hizli_gonder=True;
            //101. adresteki gönderilemeyen veri adedi kadar çalışsın
            if(s_Send_gonderilemeyen<=read_eeprom(101)){
                s_Indis_gndrlmyn=(read_eeprom(100)-s_Send_gonderilemeyen);
                d_Tx[7]=read_eeprom(s_Indis_gndrlmyn);
                //Veri_topla();
                d_Tx[8]=0;
                d_Tx[9]=s_Indis_gndrlmyn;
                b_DAQ=False;
                d_Tx[10]=make_CRC(d_Tx);
                Veri_gonder();
                s_Send++;
                s_Gonderme_hizi=0;
            }else s_Send_gonderilemeyen=0;
        }
    }
    if(s_Hizli_gonder==6){//Toplanan veriyi gönder
        b_Hizli_gonder=False;
        Veri_topla();
        b_DAQ=True;
        d_Tx[8]=1;
        d_Tx[10]=make_CRC(d_Tx);
        Veri_gonder();
        s_Indis++;
        s_Send++;
        s_Hizli_gonder=0;
    }
}
```

Veri gönderme işlemi beş sefer ardı ardına gerçekleşmiyorsa RF haberleşmesinde bir problem olduğu ve bu problemten kurtulmak için RF modem frekansının kaydırılması gerekmektedir. Frekans kaydırma işleminin birinci adımı merkez frekanstan 3 MHz ileriye kaydırılmakta hala daha iletişim yoksa birinci adımda taşınan frekanstan 3 MHz ileriye taşınmakta burada da iletişim yoksa üçüncü adımda RF modemler merkez frekansa ayarlanmaktadır. İletişim hala gerçekleşmiyorsa dördüncü adımda haberleşme frekansı merkez frekanstan 3 MHz geriye ayarlanmaktadır. Bu işlemden sonrada da iletişim hala sağlanamıyorsa beşinci adımda 3 MHz geriye taşınan frekans bir kez daha 3 MHz geriye ayarlanmaktadır. Bu işlemden sonrada haberleşme yine yoksa altıncı adımda haberleşme frekansı merkez frekansa taşınmaktadır. Bu işlem için yazılan kod görülmektedir.

```
if(s_Receive_fault>=5){//5 veri alınamadı ise
    b_Frq_shift=True;
    s_Receive_fault=0;
}
if(b_Frq_shift){
    s_Frq_shift++;
    b_Frq_shift=False;
    switch(s_Frq_shift){
        case 1:{
            Modul_oku();
            Modul_set(Tx_kanal+d_Shift_rate,Rx_kanal+d_Shift_rate);
            b_Module_set=True;
            break;
        }
        case 2:{
            Modul_oku();
            Modul_set(Tx_kanal+d_Shift_rate,Rx_kanal+d_Shift_rate);
            b_Module_set=True;
            break;
        }
        case 3:{
            Modul_oku();
            Modul_set(20,20);
            b_Module_set=True;
            break;
        }
        case 4:{
            Modul_oku();
            Modul_set(Tx_kanal+(d_Shift_rate*...
                ...(-1)),Rx_kanal+(d_Shift_rate*(-1)));
            b_Module_set=True;
            break;
        }
        case 5:{
            Modul_oku();
            Modul_set(Tx_kanal+(d_Shift_rate*...
```

```

        ...(-1)),Rx_kanal+(d_Shift_rate*(-1)));
        b_Module_set=True;
        break;
    }
    case 6:{
        Modul_oku();
        Modul_set(20,20);
        b_Module_set=True;
        s_Frq_shift=0;
        break;
    }
}
}
}

```

Veri toplama ve yer istasyonu birimlerinde bulunan RF modemlerinin veri gönderme ve alma hızı, gücü, kullanılan band ve kanal özelliklerinin mikro denetleyici tarafından okunması işlemi gerçekleştirilmektedir.

```

voidModul_oku(){
    b_Modul_oku=True;
    output_low(p_Set); //Set pin
    output_high(p_CS); //CS pin
    delay_ms(10); //SET moduna geçiş için bekle
    s_Serial_sinir=20; //20 byte parametre alımı için sınır değeri
    b_RDA=False;
    disable_interrups(GLOBAL);
    putc(0xAA);putc(0xFA);putc(0x01);//Tüm parametreleri oku
    enable_interrups(GLOBAL);
    while(!b_RDA);
        Tx_kanal  =d_Serial[0];
        Tx_band   =d_Serial[1];
        Tx_rate   =d_Serial[2];
        Tx_power  =d_Serial[3];
        Rx_kanal  =d_Serial[4];
        Rx_band   =d_Serial[5];
        Rx_rate   =d_Serial[6];
        b_RDA=False;
    s_Serial_sinir=11;
    output_high(p_Set); //Set pin
    output_high(p_CS); //CS pin
    delay_ms(100);
    b_Modul_oku=False;
}

```

Programda veri toplama ve yer istasyonu biriminde bulunan RF modüllerinin set moduna geçişi, Tx parametreleri, Rx parametreleri, seri haberleşme parametreleri ve Node ID parametrelerinin ayarlarının yapılması işlemi görülmektedir.

```

int1 Modul_set(int8 Tx_kanal_temp,int8 Rx_kanal_temp){

```

```

    b_Modul_set=True;
    output_low(p_Set); //Set pin
    output_high(p_CS); //CS pin
    delay_ms(10); //SET moduna geçiş için bekle
    s_Serial_sinir=4; //7 byte parametre set komutuna...
    ...karşılık gelen cevap uzunluğu
    b_RDA=False;
    disable_interrupts(GLOBAL);
    putc(0xAA);putc(0xFA);putc(0x03); //Parametre set komutu
    putc(Tx_kanal_temp);putc(0x01);putc(0x03);putc(0x02); //Tx
parametreleri
    putc(Rx_kanal_temp);putc(0x01);putc(0x03); //Rx parametreleri
    putc(0x03);putc(0x02);putc(0x01);putc(0x01); //Seri haberleşme
parametreleri
    putc(0x00);putc(0x00);putc(0x00);putc(0x00); //Node ID
    putc(0x00);putc(0x00); //Net ID
    putc(0x00); //Workingmode-Nodemode
    enable_interrupts(GLOBAL);
    while(!b_RDA);
        b_RDA=False;
    s_Serial_sinir=11;
    output_high(p_Set); //Set pin
    output_high(p_CS); //CS pin
    delay_ms(100);
    if(d_Serial[0]==0x4F){
        b_Modul_set=False;
        return True;
        lcd_gotoxy(1,1);
        lcd_putc("Module set true");
    }
    if(d_Serial[0]==0x45){
        b_Modul_set=False;
        return False;
        lcd_gotoxy(1,1);
        lcd_putc("Module set false");
    }
}

```

Yer istasyonuna gelen verinin doğru olup olmadığının kontrolünün yapılması işlemini yürüten program kodu görülmektedir.

```

int1 Veri_kontrol(int8 d_Rx_temp[20]){
    int8 s_Temp_Rx=0; //s_Temp_serial=0,
    int16 CRC_Temp=0;
    for(s_Temp_Rx=0; s_Temp_Rx<10; s_Temp_Rx++){
        d_Rx[s_Temp_Rx]=d_Rx_temp[s_Temp_Rx];
        CRC_Temp+=d_Rx[s_Temp_Rx];
    }
    CRC_Temp%=10;
    d_Rx[10]=d_Serial[10];
    if(d_Rx[10]==CRC_Temp)
        return True;
    else
        return False;
}

```

Yer istasyonuna gelen verinin eksiksiz ve doğru bir şekilde ulaştığına dair veri toplama bildirimine gönderilen doğrulama kodu görülmektedir.

```
voidVeri_gonder(){  
    int8 s_Temp=0;  
    disable_interrupts(GLOBAL);  
    //printf("00");  
    for(s_Temp=0;s_Temp<11;s_Temp++)  
        putc(d_Tx[s_Temp]);  
    enable_interrupts(GLOBAL);  
}
```

Yukarıda verilen veri toplama ve yer istasyonu modüllerine ait program kodları ekler kısmında verilmiştir. Program kodlarının ayrıntılarına eklerden ulaşılabilir.

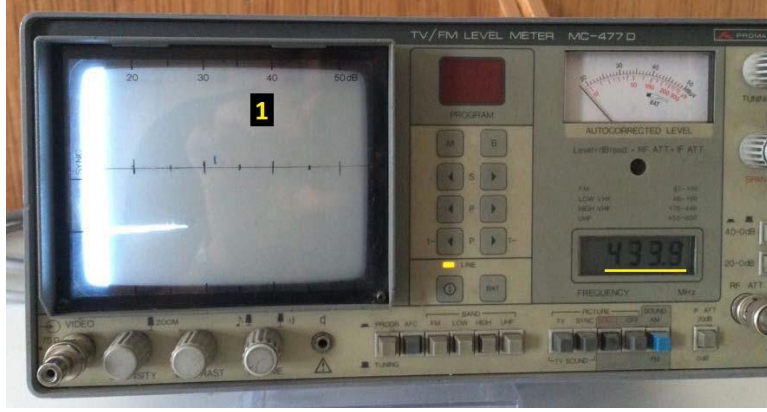


4. SENARYO

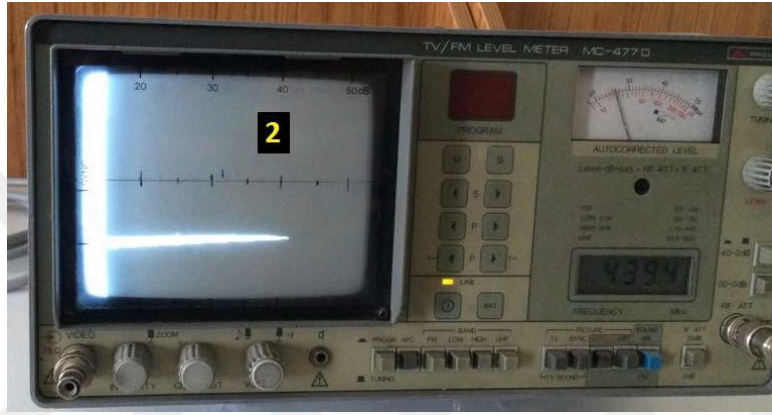
Gerçekleştirilen telemetri sistemi ve kontrol algoritmasını test etmek için bu çalışmaya özel bir test senaryosu tasarlanmıştır. Bu senaryo kapsamında, telemetri sisteminde kullanılan RF modemler yapılandırma ayarları sayesinde, 5 dBm gönderme gücünde ve 433 MHz taşıma frekansında sistem çalıştırılmıştır. Ayrıca algoritmadaki frekans kaydırma sürecini test edilebilmek için telemetri haberleşme ortamını bloke etmek üzere 433 MHz frekansında ve 11 dBm gönderme gücünde bir RF modem kullanılmıştır. Tüm bu testler laboratuvar ortamında gerçekleştirilmiştir. Söz konusu senaryoda gerçekleştirilen testlerde, telemetri sistemi 433 MHz taşıma frekansı ile başlatılmaktadır. Sistem çalışmaya başladıktan bir süre sonra, haberleşmeyi bloke edecek olan 11 dBm gücünde yayın yapan RF modem devreye alınarak haberleşmenin kesilmesi sağlanmaktadır. Haberleşme bloke edildikten sonra gönderilemeyen veri sayısının beşe ulaşması beklenmekte, ardından frekans kaydırma süreci işletilmektedir. Böylece frekans kaydırma algoritmasının basitçe test edilmesi temin edilebilmiştir.

4.1 Bulgular

Oluşturulan test senaryosuna göre deney platformu üzerinde telemetri sistemi 433 MHz taşıma frekansı ile çalışmaya başlatılmıştır. Ayrıca deney sonuçlarını gözlemlemek amacıyla kullanılan Promax MC-4770 spektrum analizörü kullanılmıştır. Resim 4.1a telemetri sisteminin 433 MHz taşıma frekansında çalıştığı görülmektedir. Spektrum ekranında görülen darbenin genliği 5 dBm değerinde olduğu da ölçülebilmektedir. Resim 4.1b de ise telemetri sistemiyle aynı frekansta yayın yapan RF modem 11 dBm genlikli bozucu etkisi görülmektedir. Bozucu etki olarak kullanılan RF modem özellikle telemetri sistemi modemlerinden daha yüksek güçte yayın yapacak şekilde ayarlanmıştır. Söz konusu telemetri haberleşmesini bloke eden RF modem devreye alındıktan bir süre sonra haberleşme kesintiye uğramış ve algoritmada yer alan frekans kaydırma süreci işletilmiştir. Resim 4.1c de bozucu etki oluşturan RF modem ve telemetri sistemine ait RF modemlerin yayınlarının ayrı frekanslarda olduğu görülmektedir. Dolayısıyla üç numaralı görsel, frekans kaydırma işleminin başarıyla gerçekleştirildiğinin göstergesi durumundadır (Manır ve Çınar 2016).



a) RF çalışma frekansı



b) RF Haberleşmesinin Kesilmesi



c) RF Haberleşmesinin Yeniden Başlaması

Resim 4.1 Frekans kaydırma işlemi.

Telemetri sistemi taşıma frekansı, frekans kaydırma işleminin bir defa çalışmasından sonra haberleşmeyi bloke edici RF modem yayınının 5 MHz ilerisi olan 438 MHz değerine ayarlanmış ve telemetri sistemi haberleşmesini bu frekansta sorunsuz sürdürebilmiştir.

5. TARTIŞMA ve SONUÇ

Bu çalışmada akıllı bir telemetry kontrol algoritması tasarlanmış, algoritmanın denenebilmesi için mikro denetleyici tabanlı bir test platformu oluşturulmuştur. Test platformunda veri toplama ve yer istasyonu olmak üzere iki ünite yer almakta olup, 100 mW gücünde RF modemler kullanılmıştır. Tasarlanan algorithmada, veri toplama istasyonunda toplanan verilerin kayba uğramadan yer istasyonuna ulaştırılabilmesi için akıllı stratejiler uygulanmıştır. Sonuç olarak, RF modemle iletişimde ortaya çıkabilecek haberleşme kesintilerinin üstesinden gelebilecek, veri kaybını ortadan kaldıracak bir algoritma tasarlanmıştır. Tasarlanan algoritmayı test edebilmek için bir senaryo oluşturulmuş, buna göre telemetry sistemi test edilmiştir. Gerçekleştirilen testlerde tasarlanan algorithmada yer alan, algoritmanın yenilikçi yönünü oluşturan frekans kaydırma sürecinin, haberleşme bloke olması durumunda devreye girerek, haberleşmenin kaydırılan yeni frekansta devam etmesini temin ettiği gözlenmiştir. Elde edilen sonuçlar, tasarlanan algoritmanın benzer frekanslarda yayın yapan telemetry sistemlerinin yoğun kullanıldığı ortamlarda haberleşmenin sürekliliğinin sağlanmasına yardımcı olabileceğini göstermiştir.

6. KAYNAKLAR

- Altemose, G., Hellermann, P., & Mazz, T. (2011, May). Active cellbalancing system using an isolated share bus for Li-Ion battery management: Focusing on satellite applications. In *Systems, Applications and Technology Conference (LISAT), 2011 IEEE Long Island* (pp. 1-7). IEEE.
- Akkurt, T. (2013). Ağır Ticari Araçlarda Kardan Şaftına Etkiyen Tork Değerlerinin Araştırılması, Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul
- Connors, J. (2007). On the subject of solar vehicle sand the benefits of the technology. In *Clean Electrical Power, 2007. ICCEP' 07. International Conference on*, 700-705.
- Çakır, S. (2007). Kısa Mesafe RF Algılayıcı Ağ ve GPRS Temelli Telemetry Uygulaması, Yüksek Lisans Tezi, Kocaeli Üniversitesi, Fen Bilimleri Enstitüsü, Kocaeli
- Çıbuk, M. (2009). WIMAX/IEEE 802.16 Ağları Üzerinde WEB Tabanlı Bio-Telemetry Uygulamaları İçin Protokol Tasarımı ve Gerçekleştirilmesi, Doktora Tezi, Fırat Üniversitesi, Fen Bilimleri Enstitüsü,
- Duchrow, T., Schröer, M., Griesbach, B., Kasperski, S., Bermppohl, F. M. G., Kramer, S., & Kirchner, F. (2012, March). Toward selectric mobility datamining. In *Electric Vehicle Conference (IEVC), 2012 IEEE International* (pp. 1-6). IEEE.
- Durmuş, M. (2010). Şanlıurfada Ceylanların (*Gazella subgutturosa*) Yaşam Alanı Büyüklüğü ve Habitat Seçimlerinin GPS Telemetry İle Belirlenmesi. Yüksek Lisana Tezi, Fen Bilimleri Enstitüsü, Ortadoğu Teknik Üniversitesi, Ankara.
- Fathirad, I., Whittingtong, J. (2011) A flexible FPGA and SBC based electric vehicle telemetry system. *International Journal of Electrical Power & Energy Systems*, 219-226.

- Grothues, T. M., Able, K. W., & Pravatiner, J. H. (2012). Winter flounder (*Pseudopleuronectes americanus* Walbaum) burial in estuaries: acoustic telemetry triumphand tribulation. *Journal of Experimental Marine Biology and Ecology*, **438**: 125-136.
- Hocaoğlu, O.F., Serttaş, F. (2014) Akü'lü araba Tübitak-Alternatif Enerjili Araç Yarışları Teknik Tasarım Raporu, Afyonkarahisar.
- İlgin, A.H. (2015) Ankara Üniversitesi Bilimsel Araştırma Projesi Kesin Raporu Batarya Sistemli Elektrikli Araba Projesi, Ankara.
- Larrauri, J. I., Sorrosal, G., & González, M. (2013, May). Automatic system for over head power line in spection using an Un manned Aerial Vehicle—Relifo project. In *Unmanned Aircraft Systems (ICUAS), 2013 International Conference on* (pp. 244-252). IEEE.
- Manir, E., & Çinar, S.M. (2016), On the investigation of a designed smart telemetry control algorithm, 2016 National Conference on *Elektrical, Electronics and Biomedical Engineering (ELECO)*, 650-654.
- Mammadov, E., & Gueaieb, W. (2014, May). Long-range communication frame work for multi-agent autonomous UAVs. In *Unmanned Aircraft Systems (ICUAS), 2014 International Conference on* (pp. 395-403). IEEE.
- Manzie, C., Kim, T. S., & Sharma, R. (2012). Optimal use of telemetry by parallel hybrid vehicles in urban driving. *Transportation Research Part C: Emerging Technologies*, **25**: 134-151.
- Royer, F., & Lutcavage, M. (2008). Filtering and interpreting location errors in satellite telemetry of marine animals. *Journal of Experimental Marine Biology and Ecology*, **359(1)**: 1-10.
- Thums, M., Whiting, S. D., Reisser, J. W., Pendoley, K. L., Pattiaratchi, C. B., Harcourt, R. G., ... & Meekan, M. G. (2013). Tracking seaturtle hat chlins—a pilot studyu singa coustic telemetry. *Journal of Experimental Marine Biology and Ecology*, **440**: 156-163.

- Tucker, A. D. (2010). Nest site fidelity and clutch frequency of logger head turtle sarebet tereluci dated by satellite telemetry than by nocturnal tagging efforts: implications for stockestimation. *Journal of Experimental Marine Biolog and Ecology*, **383(1)**: 48-55.
- Ustun, O., Yilmaz, M., Gokce, C., Karakaya, U., & Tuncay, R. N. (2009). Energy management method for solar race car design and application. In Electric Machines and Drives Conference, 2009. IEMDC'09. IEEE International, 804-811.
- Üstün, Ö., Yılmaz, M., Tüysüz, A., Titiz, K. F., Şahin, A., Tuncay, N. R., Gökçe, C., Karakaya, U., Yıldırım, A. (2006) Güneş enerjili yarış aracının elektrik tasarım ölçütlerinin, yol performans simülasyonunun ve enerji yönetim sisteminin geliştirilmesi. *Tübitak Bilim Teknik Dergisi*

6.1 İnternet Kaynakları

- 1- <http://www.elektrikport.com/teknik-kutuphane/telemetry-sistemi-nedir/16904#ad-image-0> 30.11.2016
- 2- http://http://www.scem.univ-savoie.fr/formations/masters/electronique-telecoms/wp-content/files/ETRS-604/tp/easypic6_manual.pdf 01.12.2016
- 3- <http://http://www.sage.tubitak.gov.tr/tr/hizmetlerimiz/mobil-telemetry-test-hizmetleri> 02.12.2016



ÖZGEÇMİŞ

Adı Soyadı : Emre MANIR
Doğum Yeri ve Tarihi : ESKİŞEHİR – 06.03.1991
Yabancı Dili : İngilizce
İletişim (Telefon/e-posta) : 0 537 619 91 53 / emremanir@gmail.com

Eğitim Durumu (Kurum ve Yıl)

Lise : Türk Telekom Anadolu Teknik Lisesi, (2005-2009)
Lisans : Afyon Kocatepe Üniversitesi, Elektrik Öğretmenliği, (2009-2013)
Yüksek Lisans : Afyon Kocatepe Üniversitesi, Yenilenebilir Enerji Sistemleri Anabilim Dalı, (2014-2017)

Çalıştığı Kurum/Kurumlar ve Yıl : Milli Eğitim Bakanlığı, (2015-Devam Ediyor.)

EKLER

EK 1. Veri Toplama Modülü Program Kodu

```
//Veri toplama modülü programı
#include<main.h>
#include<flcd_nicerf.c>

#define b_Gonder Pin_B0

//LCD Pinler
//E    RD0
//RS    RD1
//Data RD4-7
#define LCD_bck_light Pin_C2

//Butonlar (Pull-up)
#define b_Enter Pin_A4
#define b_ESC Pin_A5

//NiceRF Kontrol pinleri
#define p_CS Pin_B5
#define p_Set Pin_B4

//Analog pinleri
//AN0
//AN1

signed int8 d_Shift_rate=1;
int8 d_Tx[11],d_Rx[11],d_Serial[20],
Tx_kanal,Tx_band,Tx_rate,Tx_power,
Rx_kanal,Rx_band,Rx_rate;
int8
s_Serial=0,s_Display=0,s_Send_true=0,s_Send_false=0,s_Serial_sin
ir=11,
s_Send=0,s_Receive=0,s_Receive_fault=0,s_Bekle_Frq_shift=0,s_Frq
_shift=0,s_LCD_bckl_gecik=0,
s_Gonderme_hizi=0,c_Gonderme_hizi=0;
int8
c_Hizli_gonder=1,c_Normal_gonder=6,
s_Hizli_gonder=0,s_Indis=0,
s_Send_gonderilemeyen=0,s_Indis_gndrlmyn=0,s_Gndrlmyn=0;
int16 s_1s;
int32 Komut=0;
int1
b_RDA=False,b_1s=False,b_Modul_oku=False,b_Modul_set=False,b_Frq
_shift=False,
```

EK 1. (Devam) Veri Toplama Modülü Program Kodu

```
b_Bekle_Frq_shift=False,b_Module_set=False,b_1s_bck_light=False,  
b_LCD_bckl_gecik=False,  
b_Gonderilemeyen=False,  
b_Merkeze_iletildi=False,  
b_Paket_hatali=False,b_DAQ=False,b_Hizli_gonder=False;
```

```
#int_TIMER1  
void Timer1_isr(void){  
    s_1s++;  
    //if(s_1s>=100){  
    if(s_1s>=100){  
        s_1s=0;  
        b_1s=True;  
        b_1s_bck_light=True;  
        //output_toggle(LCD_bck_light);  
    }  
    set_timer1(55537);  
}
```

```
#int_RDA  
void RDA_isr(void){  
    //output_toggle(LCD_bck_light);  
    //clear_interrupt(int_rda);  
    disable_interrupts(global);  
    d_Serial[s_Serial]=getc();s_Serial++;  
    if(s_Serial==s_Serial_sinir){  
        b_RDA=TRUE;s_Serial=0;  
    }  
    //lcd_gotoxy(1,2);  
    //printf(lcd_putc,"%d",s_Serial);  
    enable_interrupts(global);  
}
```

```
//int8 ASCItoInt(int8 Data){  
    //if(Data<58)return(Data-48);  
    //if(Data>=65)return(Data-55);  
    //}
```

```
//int8 Byte_yap(int8 ByteL,int8 ByteH){  
    //ByteL=ASCItoInt(ByteL);  
    //ByteH=ASCItoInt(ByteH);  
    //ByteH<=4;  
    //returnByteH|ByteL;  
    //}
```

EK 1. (Devam) Veri Toplama Modülü Program Kodu

```
int1 Veri_kontrol(int8 d_Rx_temp[20]){
    int8 s_Temp_Rx=0;//s_Temp_serial=0,
    int16 CRC_Temp=0;
    //for(s_Temp_serial=0;s_Temp_serial<22;s_Temp_serial+=2){

    //d_Rx[s_Temp_Rx]=Byte_yap(d_Serial_temp[s_Temp_serial+1],d_Seri
al_temp[s_Temp_serial]);
        //printf("%2X ",d_Rx[s_Temp_Rx]);
        //s_Temp_Rx++;
    //}
    for(s_Temp_Rx=0;s_Temp_Rx<10;s_Temp_Rx++){
        d_Rx[s_Temp_Rx]=d_Rx_temp[s_Temp_Rx];
        CRC_Temp+=d_Rx[s_Temp_Rx];
    }
    CRC_Temp%=10;
    //printf("CRC: %2X",CRC_Temp);
    //printf(" d_Rx[4]: %2X ",d_Rx[4]);
    d_Rx[10]=d_Serial[10];
    if(d_Rx[10]==CRC_Temp)
        return True;
    else
        return False;
}

voidModul_oku(){
    b_Modul_oku=True;
    output_low(p_Set); //Set pin
    output_high(p_CS); //CS pin
    delay_ms(10); //SET moduna geçiş için bekle
    s_Serial_sinir=20; //20 byte parametre alımı için sınır değeri
    b_RDA=False;
    disable_interrupts(GLOBAL);
    putc(0xAA);putc(0xFA);putc(0x01);//Tüm parametreleri oku
    enable_interrupts(GLOBAL);
    while(!b_RDA);
    Tx_kanal   =d_Serial[0];
    Tx_band    =d_Serial[1];
    Tx_rate    =d_Serial[2];
    Tx_power   =d_Serial[3];
    Rx_kanal   =d_Serial[4];
    Rx_band    =d_Serial[5];
    Rx_rate    =d_Serial[6];
    b_RDA=False;
    s_Serial_sinir=11;
    output_high(p_Set); //Set pin
```

EK 1. (Devam) Veri Toplama Modülü Program Kodu

```
output_high(p_CS); //CS pin
delay_ms(100);
b_Modul_oku=False;
}

int1 Modul_set(int8 Tx_kanal_temp,int8 Rx_kanal_temp){
b_Modul_set=True;
output_low(p_Set); //Set pin
output_high(p_CS); //CS pin
delay_ms(10); //SET moduna geçiş için bekle
s_Serial_sinir=4; //7 byte parametre set komutuna karşılık gelen
cevap uzunluğu
b_RDA=False;
disable_interrupts(GLOBAL);
putc(0xAA);putc(0xFA);putc(0x03);//Parametre set komutu
putc(Tx_kanal_temp);putc(0x01);putc(0x03);putc(0x02);          //Tx
parametreleri
putc(Rx_kanal_temp);putc(0x01);putc(0x03);//Rx parametreleri
putc(0x03);putc(0x02);putc(0x01);putc(0x01);//Seri haberleşme
parametreleri
putc(0x00);putc(0x00);putc(0x00);putc(0x00);//Node ID
putc(0x00);putc(0x00);//Net ID
putc(0x00);//Workingmode-Nodemode
enable_interrupts(GLOBAL);
    //lcd_gotoxy(1,2);
    //printf(lcd_putc,"Komut_OK");
while(!b_RDA);
    //lcd_gotoxy(1,2);
    //printf(lcd_putc,"RDA_OK");
b_RDA=False;
s_Serial_sinir=11;
output_high(p_Set); //Set pin
output_high(p_CS); //CS pin
delay_ms(100);
if(d_Serial[0]==0x4F){
b_Modul_set=False;
return True;
}
if(d_Serial[0]==0x45){
b_Modul_set=False;
returnFalse;
}
}

voidinit(){
```

EK 1. (Devam) Veri Toplama Modülü Program Kodu

```
setup_wdt(WDT_1152MS);           //~1152 msreset
setup_psp(PSP_DISABLED);         // PSP birimi devre dışı
setup_spi(SPI_SS_DISABLED);      // SPI birimi devre dışı
    setup_CCP1(CCP_OFF);          // CCP1 birimi devre dışı
    setup_CCP2(CCP_OFF);          // CCP2 birimi devre dışı
    setup_timer_2(T2_DISABLED,0,1); // T2 zamanlayıcısı devre dışı
setup_adc(ADC_CLOCK_INTERNAL|ADC_CLOCK_DIV_64);
    SETUP_ADC_PORTS(AN0_AN1_AN3);
    setup_timer_1(T1_INTERNAL|T1_DIV_BY_2);
enable_interrupts(INT_TIMER1);
set_timer1(55537);
set_tris_a(0xff);
set_tris_b(0x00);
set_tris_c(0x80);
set_tris_d(0x00);
enable_interrupts(INT_RDA);
enable_interrupts(GLOBAL);
lcd_init();
output_high(LCD_bck_light);
output_high(p_CS); //CS pin
output_high(p_Set); //Set pin
delay_ms(100);
lcd_gotoxy(1,1);
lcd_putc("NICE RF");
delay_ms(200);
lcd_gotoxy(1,1);
lcd_putc(" ");
output_low(LCD_bck_light);
Modul_set(20,20);
Modul_oku();
}
```

```
voidVeri_topla(){
    int8 Anlg_value=0;
d_Tx[0]=Tx_kanal;
d_Tx[1]=Tx_band;
d_Tx[2]=Tx_rate;
d_Tx[3]=Tx_power;
d_Tx[4]=Rx_kanal;
d_Tx[5]=Rx_band;
d_Tx[6]=Rx_rate;
    SET_ADC_CHANNEL(0);
delay_us(100);
Anlg_value=READ_ADC();
d_Tx[7]=Anlg_value;
```

EK 1. (Devam) Veri Toplama Modülü Program Kodu

```
lcd_gotoxy(7,2);
printf(lcd_putc,"%2x",d_Tx[7]);
//SET_ADC_CHANNEL(1);
//delay_us(100);
//Anlg_value=READ_ADC();
//d_Tx[8]=Anlg_value;
lcd_gotoxy(10,2);
printf(lcd_putc,"%2x",d_Tx[8]);
//d_Tx[8]=make8(Anlg_value,1);
d_Tx[9]=s_Indis;//input_d();
}

int8 make_CRC(int8 d_Tx_temp[11]){
    int8 s_Temp=0;
    int16 CRC_temp=0;
    for(s_Temp=0;s_Temp<10;s_Temp++)
        CRC_temp+=d_Tx_temp[s_Temp];
    returnCRC_Temp%=10;
}

voidVeri_gonder(){
    int8 s_Temp=0;
    disable_interrupts(GLOBAL);
    //printf("00");
    for(s_Temp=0;s_Temp<11;s_Temp++)
        putc(d_Tx[s_Temp]);
    enable_interrupts(GLOBAL);
}

void main(){
    init();

    while(TRUE){
        //LCD arka ışık kapatma
        //-----
        if(!input(b_Enter)||!input(b_ESC)){
            output_high(LCD_bck_light);
            b_LCD_bckl_gecik=True;
        }
        if(b_LCD_bckl_gecik&&b_1s_bck_light){
            s_LCD_bckl_gecik++;
            b_1s_bck_light=False;
        }
        if(s_LCD_bckl_gecik>=20){
            s_LCD_bckl_gecik=0;
            b_LCD_bckl_gecik=False;
        }
    }
}
```

EK 1. (Devam) Veri Toplama Modülü Program Kodu

```
output_low(LCD_bck_light);
}
//-----
//RR modülden veri alma işlemleri
//-----
-----
if(b_RDA&&!b_Modul_oku){
s_Receive++;
//Modül set fonksiyonu çalışıyorsa gelen veriyi reddet
if(b_Module_set){
s_Send=0;
s_Receive=0;
b_Module_set=False;
}
//Veri paketi doğru alındı mı?
if(Veri_kontrol(d_Serial)){
if(d_Rx[7]==1){//Veri merkeze doğru ulaştı mı?
s_Send_true++;
b_Merkeze_iletildi=True;
if(b_Gonderilemeyen)s_Send_gonderilemeyen++;//Bu değişkenin veri
karşı tarafa ulaşmışsa arttırılması gerekiyor

if(s_Send_gonderilemeyen==read_eeprom(101))b_Gonderilemeyen=False;
}
else{
s_Send_false++;
b_Merkeze_iletildi=False;
//printf("Verikontrol OK!\n");
}
b_Paket_hatali=False;
}
else {
b_Paket_hatali=True;
//printf("Veri hatali!\n");
}
//lcd_gotoxy(1,1);
//for(s_Display=0;s_Display<=7;s_Display++)
//printf(lcd_putc,"%2X",d_Rx[s_Display]);
//s_Display=0;
lcd_gotoxy(1,2);
//printf(lcd_putc,"S_T:%3u
S_F:%3u",s_Send_true,s_Send_false);
//Verinin merkeze doğru ve yanlış ulaşma sayıları
yazdırılıyor
printf(lcd_putc,"%2x %2x",s_Send_true,s_Send_false);
b_RDA=False;
```

EK 1. (Devam) Veri Toplama Modülü Program Kodu

```
//RDA end
//-----
-----
//if(input(b_Gonder)){
//Veri gönderme işlemleri
//-----
-----
if(b_1s){//Gönderme zamanı geldi mi?
s_Gonderme_hizi++;
    //if(s_Send!=s_Receive){//Gönderilen ve alınan veri eşit
mi?
if(s_Send!=s_Send_true){//Gönderilen ve karşı tarafta alınan
veri farklı mı?
b_Gonderilemeyen=True;
    //Sadece toplanan veriler gönderilemediğinde çalışacak
if(b_DAQ){
write_eeprom((s_Indis-1),d_Tx[7]);//Gönderilemeyen veri belleğe
yazılıyor
write_eeprom(100,(s_Indis-1));//Gönderilemeyen son verinin
indisi 100. bellek adresine kaydediliyor
s_Gndrlmyn++;
write_eeprom(101,s_Gndrlmyn);//Gönderilemeyen veri adedi 101.
bellek adresine kaydediliyor
}
    //b_Frq_shift=False;
    //}else{
    //Modul_oku();
if(!b_Hizli_gonder)s_Receive_fault++;
    //s_Send=0;
s_Send=s_Send_true;
s_Receive=0;
}
    //Gönderilemeyen veri var mı?
if(!b_Gonderilemeyen){//Gönderilemeyen veri yok
if(s_Gonderme_hizi>=c_Normal_gonder){//Toplanan veriyi gönder
b_Hizli_gonder=False;
Veri_topla();
b_DAQ=True;
d_Tx[8]=1;
d_Tx[10]=make_CRC(d_Tx);
Veri_gonder();
s_Send++;
s_Gonderme_hizi=0;
s_Indis++;
}
```

```

}else{//Gönderilemeyen veri var
if(s_Gonderme_hizi>c_Hizli_gonder){//Bellkteki gönderilemeyen
veriyi gönder
s_Hizli_gonder++;
if(s_Hizli_gonder!=6){
b_Hizli_gonder=True;
//101. adresteki gönderilemeyen veri adedi kadar
çalışsın
if(s_Send_gonderilemeyen<=read_eeprom(101)){
s_Indis_gndrlmyn=(read_eeprom(100)-s_Send_gonderilemeyen);
d_Tx[7]=read_eeprom(s_Indis_gndrlmyn);
//Veri_topla();
d_Tx[8]=0;
d_Tx[9]=s_Indis_gndrlmyn;
b_DAQ=False;
d_Tx[10]=make_CRC(d_Tx);
Veri_gonder();
s_Send++;
s_Gonderme_hizi=0;
}else s_Send_gonderilemeyen=0;
}
if(s_Hizli_gonder==6){//Toplanan veriyi gönder
b_Hizli_gonder=False;
Veri_topla();
b_DAQ=True;
d_Tx[8]=1;
d_Tx[10]=make_CRC(d_Tx);
Veri_gonder();
s_Indis++;
s_Send++;
s_Hizli_gonder=0;
}
}
} //b_Gonderilemeyenend
//if(b_Bekle_Frq_shift)s_Bekle_Frq_shift++;
//if(s_Bekle_Frq_shift>=20){//
//b_Bekle_Frq_shift=False;
//s_Bekle_Frq_shift=0;
//}
if(s_Receive_fault>=5){//5 veri alınamadı ise
b_Frq_shift=True;
s_Receive_fault=0;
//s_Send=0;s_Receive=0;
}
//if(b_Frq_shift&&!b_Bekle_Frq_shift){

```

EK 1. (Devam) Veri Toplama Modülü Program Kodu

```
if(b_Frq_shift){
    //b_Bekle_Frq_shift=True;
    //Modul_set(Tx_kanal+5,Rx_kanal+5);
    s_Frq_shift++;
    //lcd_gotoxy(5,1);
    //printf(lcd_putc,"%d %d",Tx_kanal,Rx_kanal);
    b_Frq_shift=False;
    //}
    switch(s_Frq_shift){
    case 1:{
        Modul_oku();
        Modul_set(Tx_kanal+d_Shift_rate,Rx_kanal+d_Shift_rate);
        b_Module_set=True;
        break;
    }
    case 2:{
        Modul_oku();
        Modul_set(Tx_kanal+d_Shift_rate,Rx_kanal+d_Shift_rate);
        b_Module_set=True;
        break;
    }
    case 3:{
        Modul_oku();
        Modul_set(20,20);
        b_Module_set=True;
        break;
    }
    case 4:{
        Modul_oku();
        Modul_set(Tx_kanal+(d_Shift_rate*(-1)),Rx_kanal+(d_Shift_rate*(-1)));
        b_Module_set=True;
        break;
    }
    case 5:{
        Modul_oku();
        Modul_set(Tx_kanal+(d_Shift_rate*(-1)),Rx_kanal+(d_Shift_rate*(-1)));
        b_Module_set=True;
        break;
    }
    case 6:{
        Modul_oku();
        Modul_set(20,20);
        b_Module_set=True;
    }
```

EK 1. (Devam) Veri Toplama Modülü Program Kodu

```
s_Frq_shift=0;
break;
}
} //Case end
}
lcd_gotoxy(1,1);
printf(lcd_putc,"Sn:%2X          Rc:%2X
S:%2X",s_Send,s_Receive,s_Frq_shift);
b_1s=False;
    //Modul_oku();
    //printf("B basildi");
    //delay_ms(50);
    //while(input(b_Gonder));
    //SET_ADC_CHANNEL(0);
    //delay_us(100);
    //Anlg_value=READ_ADC();
    //disable_interrupts(GLOBAL);
    //printf("Anlg: %4LX",Anlg_value);
    //lcd_gotoxy(1,2);
    //printf(lcd_putc,"Anlg: %4LX",Anlg_value);
    //enable_interrupts(GLOBAL);
} //b_1s end
    //if(input(Pin_D1)){
    //delay_ms(50);
    //while(input(b_Gonder));
    //if(Modul_set(Tx_kanal,Rx_kanal)){
    //    /*output_high(Pin_D0);
delay_ms(500);
output_low(Pin_D0);
delay_ms(500);
output_high(Pin_D0);
delay_ms(500);
output_low(Pin_D0);*/
    //} else{
    //    /*output_high(Pin_D0);
delay_ms(500);
output_low(Pin_D0);*/
    //}
    //}
restart_wdt();
} //whileend
} //main end
```

EK 2. Merkez Modülü Program Kodu

```
//Merkez modülü programı
#include<main.h>
#include<flcd_nicerf.c>

#define b_Gonder Pin_B0

//LCD Pinler
//E RD0
//RS RD1
//Data RD4-7
#define LCD_bck_light Pin_C2

//Butonlar (Pull-up)
#define b_Enter Pin_A4
#define b_ESC Pin_A5

//NiceRF Kontrol pinleri
#define p_CS Pin_B5
#define p_Set Pin_B4

signed int8 d_Shift_rate=1;
int16 s_Zaman=0,s_Veri_geldi=0;
int8 d_Tx[11],d_Rx[11],d_Serial[20],
Tx_kanal,Tx_band,Tx_rate,Tx_power,
Rx_kanal,Rx_band,Rx_rate;
int8
s_Serial=0,s_Display=0,s_Send_true=0,s_Rec_true=0,s_Send_false=0
,s_Rec_false=0,s_Serial_sinir=11,
s_Send=0,s_Receive=0,s_Receive_fault=0,s_Bekle_Frq_shift=0,s_Frq
_shift=0,s_LCD_bckl_gecik=0,
s_Veri_alma_hizi=0,c_Veri_alma_hizi=6;
int16 s_1s;
int32 Komut=0;
int1
b_RDA=False,b_1s=False,b_Modul_oku=False,b_Modul_set=False,b_Frq
_shift=False,

b_Bekle_Frq_shift=False,b_Module_set=False,b_1s_bck_light=False,
b_LCD_bckl_gecik=False,b_Start=True;

#int_TIMER1
void Timer1_isr(void){
s_1s++;
if(s_1s>=100){
s_1s=0;
b_1s=True;
```

EK 2. (Devam) Merkez Modülü Program Kodu

```
b_1s_bck_light=True;
    //output_toggle(LCD_bck_light);
}
set_timer1(55537);
}

#int_RDA
voidRDA_isr(void){
    //clear_interrupt(int_rda);
    disable_interrupts(global);
    d_Serial[s_Serial]=getc();s_Serial++;
    if(s_Serial==s_Serial_sinir){
        b_RDA=TRUE;s_Serial=0;
    }
    //lcd_gotoxy(1,2);
    //printf(lcd_putc,"%d",s_Serial);
    enable_interrupts(global);
}

//int8 ASCItoInt(int8 Data){
    //if(Data<58)return(Data-48);
    //if(Data>=65)return(Data-55);
//}

//int8 Byte_yap(int8 ByteL,int8 ByteH){
    //ByteL=ASCItoInt(ByteL);
    //ByteH=ASCItoInt(ByteH);
    //ByteH<=4;
    //returnByteH|ByteL;
//}

int1 Veri_kontrol(int8 d_Rx_temp[20]){
    int8 s_Temp_Rx=0;//s_Temp_serial=0,
    int16 CRC_Temp=0;
    //for(s_Temp_serial=0;s_Temp_serial<22;s_Temp_serial+=2){

    //d_Rx[s_Temp_Rx]=Byte_yap(d_Serial_temp[s_Temp_serial+1],d_Serial_temp[s_Temp_serial]);
    //printf("%2X ",d_Rx[s_Temp_Rx]);
    //s_Temp_Rx++;
    //}
    for(s_Temp_Rx=0;s_Temp_Rx<10;s_Temp_Rx++){
        d_Rx[s_Temp_Rx]=d_Rx_temp[s_Temp_Rx];
        CRC_Temp+=d_Rx[s_Temp_Rx];
    }
}
```

EK 2. (Devam) Merkez Modülü Program Kodu

```
CRC_Temp%=10;
    //printf("CRC: %2X",CRC_Temp);
    //printf(" d_Rx[4]: %2X ",d_Rx[4]);
d_Rx[10]=d_Serial[10];
if(d_Rx[10]==CRC_Temp)
return True;
else
returnFalse;
}

voidModul_oku(){
b_Modul_oku=True;
output_low(p_Set); //Set pin
output_high(p_CS); //CS pin
delay_ms(10); //SET moduna geçiş için bekle
s_Serial_sinir=20; //20 byte parametre alımı için sınır değer
b_RDA=False;
disable_interrupts(GLOBAL);
putc(0xAA);putc(0xFA);putc(0x01);//Tüm parametreleri oku
enable_interrupts(GLOBAL);
while(!b_RDA);
Tx_kanal  =d_Serial[0];
Tx_band   =d_Serial[1];
Tx_rate   =d_Serial[2];
Tx_power  =d_Serial[3];
Rx_kanal  =d_Serial[4];
Rx_band   =d_Serial[5];
Rx_rate   =d_Serial[6];
b_RDA=False;
s_Serial_sinir=11;
output_high(p_Set); //Set pin
output_high(p_CS); //CS pin
delay_ms(100);
b_Modul_oku=False;
}

int1 Modul_set(int8 Tx_kanal_temp,int8 Rx_kanal_temp){
b_Modul_set=True;
output_low(p_Set); //Set pin
output_high(p_CS); //CS pin
delay_ms(10); //SET moduna geçiş için bekle
s_Serial_sinir=4; //7 byte parametre set komutuna karşılık gelen
cevap uzunluğu
b_RDA=False;
disable_interrupts(GLOBAL);
```

EK 2. (Devam) Merkez Modülü Program Kodu

```
putc(0xAA);putc(0xFA);putc(0x03);//Parametre set komutu
putc(Tx_kanal_temp);putc(0x01);putc(0x03);putc(0x02);          //Tx
parametreleri
putc(Rx_kanal_temp);putc(0x01);putc(0x03);//Rx parametreleri
putc(0x03);putc(0x02);putc(0x01);putc(0x01);//Seri haberleşme
parametreleri
putc(0x00);putc(0x00);putc(0x00);putc(0x00);//Node ID
putc(0x00);putc(0x00);//Net ID
putc(0x00);//Workingmode-Nodemode
enable_interrupts(GLOBAL);
    //lcd_gotoxy(1,2);
    //printf(lcd_putc,"Komut_OK");
while(!b_RDA);
    //lcd_gotoxy(1,2);
    //printf(lcd_putc,"RDA_OK");
b_RDA=False;
s_Serial_sinir=11;
output_high(p_Set); //Set pin
output_high(p_CS); //CS pin
delay_ms(100);
if(d_Serial[0]==0x4F){
b_Modul_set=False;
return True;
lcd_gotoxy(1,1);
lcd_putc("Module set true");
}
if(d_Serial[0]==0x45){
b_Modul_set=False;
returnFalse;
lcd_gotoxy(1,1);
lcd_putc("Module set false");
}
}

voidVeri_topla(){
    int16 Anlg_value=0;
    //SET_ADC_CHANNEL(0);
    //delay_us(100);
    //Anlg_value=READ_ADC();
d_Tx[0]=Tx_kanal;
d_Tx[1]=Tx_band;
d_Tx[2]=Tx_rate;
d_Tx[3]=Tx_power;
d_Tx[4]=Rx_kanal;
d_Tx[5]=Rx_band;
```

EK 2. (Devam) Merkez Modülü Program Kodu

```
d_Tx[6]=Rx_rate;
//d_Tx[7]=make8(Anlg_value,0);
//d_Tx[8]=make8(Anlg_value,1);
//d_Tx[9]=input_d();
}

int8 make_CRC(int8 d_Tx_temp[11]){
    int8 s_Temp=0;
    int16 CRC_temp=0;
    for(s_Temp=0;s_Temp<10;s_Temp++)
        CRC_Temp+=d_Tx_temp[s_Temp];
    returnCRC_Temp%=10;
}

voidVeri_gonder(){
    int8 s_Temp=0;
    disable_interrupts(GLOBAL);
    //printf("00");
    for(s_Temp=0;s_Temp<11;s_Temp++)
        putc(d_Tx[s_Temp]);
    enable_interrupts(GLOBAL);
}

voidinit(){
    setup_wdt(WDT_1152MS);           //~1152 msreset
    setup_psp(PSP_DISABLED);         // PSP birimi devre dışı
    setup_spi(SPI_SS_DISABLED);      // SPI birimi devre dışı
    setup_CCP1(CCP_OFF);             // CCP1 birimi devre dışı
    setup_CCP2(CCP_OFF);             // CCP2 birimi devre dışı
    setup_timer_2(T2_DISABLED,0,1); // T2 zamanlayıcısı devre dışı
    setup_adc(ADC_CLOCK_INTERNAL|ADC_CLOCK_DIV_64);
    SETUP_ADC_PORTS(AN0);
    setup_timer_1(T1_INTERNAL|T1_DIV_BY_2);
    //enable_interrupts(INT_TIMER1);
    set_timer1(55537);
    enable_interrupts(INT_RDA);
    enable_interrupts(GLOBAL);
    set_tris_a(0xff);
    set_tris_b(0x00);
    set_tris_c(0x80);
    set_tris_d(0x00);
    lcd_init();
    output_high(LCD_bck_light);
    output_high(p_CS); //CS pin
    output_high(p_Set); //Set pin
```

EK 2. (Devam) Merkez Modülü Program Kodu

```
delay_ms(100);
lcd_gotoxy(1,1);
lcd_putc("NICE RF");
delay_ms(500);
lcd_gotoxy(1,1);
lcd_putc("      ");
//output_low(LCD_bck_light);
Modul_set(20,20);
Modul_oku();
}

void main(){
init();
while(TRUE){
    /*if(!input(b_Enter)||!input(b_ESC)){
output_high(LCD_bck_light);
b_LCD_bckl_gecik=True;
}
if(b_LCD_bckl_gecik&&b_1s_bck_light){
s_LCD_bckl_gecik++;
b_1s_bck_light=False;
}
if(s_LCD_bckl_gecik>=5){
s_LCD_bckl_gecik=0;
b_LCD_bckl_gecik=False;
output_low(LCD_bck_light);
}*/
if(b_Start){
if(s_Veri_geldi==1){//Gelen ilk veriyle Timer1 başlatıldı
enable_interrupts(int_Timer1);
b_Start=False;
}
}

if(b_1s){//Zaman sayacı
    //output_toggle(LCD_bck_light);
s_Veri_alma_hizi++;
if(s_Veri_alma_hizi>=c_Veri_alma_hizi){
s_Zaman++;
s_Veri_alma_hizi=0;
if((s_Zaman+1)!=s_Veri_geldi){//Gelen veriyle gelmesi gereken
veri farklı mı?
s_Bekle_Frq_shift++;
s_Zaman=0;
s_Veri_geldi=0;
```

EK 2. (Devam) Merkez Modülü Program Kodu

```
}
}
    //if(s_Rec_true>=5){//5 veri gelmeden frekans deęiřtirme
    prosesi iřleme girmesin
        //lcd_gotoxy(1,1);
        //printf(lcd_putc,"1");

        //}
b_1s=False;
lcd_gotoxy(1,1);
printf(lcd_putc,"Rq:%2X                               Rc:%2X
S:%X",s_Zaman,s_Veri_geldi,s_Frq_shift);
}

if(s_Bekle_Frq_shift>=5){//Eęer 5 veri alınmadı ise
    //lcd_gotoxy(3,1);
    //printf(lcd_putc,"2");
s_Bekle_Frq_shift=0;
b_Frq_shift=True;
}

if(b_Frq_shift){//Frekans kaydırma aktif ise
s_Frq_shift++;
    //lcd_gotoxy(5,1);
    //printf(lcd_putc,"3");
    //Modul_oku();
    //Modul_set(Tx_kanal+5,Rx_kanal+5);
b_Frq_shift=False;
switch(s_Frq_shift){
case 1:{
Modul_oku();
Modul_set(Tx_kanal+d_Shift_rate,Rx_kanal+d_Shift_rate);
break;
}
case 2:{
Modul_oku();
Modul_set(Tx_kanal+d_Shift_rate,Rx_kanal+d_Shift_rate);
break;
}
case 3:{
Modul_oku();
Modul_set(20,20);
break;
}
case 4:{
```

EK 2. (Devam) Merkez Modülü Program Kodu

```
Modul_oku();
    Modul_set(Tx_kanal+(d_Shift_rate*(-1)),Rx_kanal+(d_Shift_rate*(-1)));
break;
}
case 5:{
Modul_oku();
    Modul_set(Tx_kanal+(d_Shift_rate*(-1)),Rx_kanal+(d_Shift_rate*(-1)));
break;
}
case 6:{
Modul_oku();
Modul_set(20,20);
s_Frq_shift=0;
break;
}
} //Case end
}

if(b_RDA&&!b_Modul_oku){ //Veri geldiyse modül okunana kadar bekle
s_Veri_geldi++;
    //lcd_gotoxy(1,2);
    //printf(lcd_putc,"RDA ok");
if(Veri_kontrol(d_Serial)){
s_Rec_true++;
    //delay_ms(10);
d_Tx[7]=1; //Veri doğru biti
d_Tx[8]=0;
d_Tx[9]=0;
    //d_Tx[10]=make_CRC(d_Tx);
    //Veri_gonder();
    //printf("Verikontrol OK!\n");
}
else {
s_Rec_false++;
    //delay_ms(10);
d_Tx[7]=0; //Veri yanlış biti
d_Tx[8]=0;
d_Tx[9]=0;
    //d_Tx[10]=make_CRC(d_Tx);
    //Veri_gonder();
    //printf("Veri hatali!\n");
}
```

EK 2. (Devam) Merkez Modülü Program Kodu

```
//lcd_gotoxy(1,1);
//for(s_Display=0;s_Display<=7;s_Display++){
//printf(lcd_putc,"%2X",d_Rx[s_Display]);
//printf("%c",d_Serial[s_Display]);
//}
//s_Display=0;
lcd_gotoxy(1,2);
printf(lcd_putc,"%2X %2X %2X",d_Rx[7],d_Rx[8],d_Rx[9]);

if(d_Rx[8]==1){
write_eeprom(d_Rx[9],d_Rx[7]);
c_Veri_alma_hizi=6;
}
if(d_Rx[8]==0){
write_eeprom((d_Rx[9]+100),d_Rx[7]);
c_Veri_alma_hizi=1;
}

//printf(lcd_putc,"R_T:%3u
R_F:%3u",s_Rec_true,s_Rec_false);
b_RDA=False;
//Modul_oku();
Veri_topla();//Merkez modülü modem bilgilerini toplar
d_Tx[10]=make_CRC(d_Tx);
Veri_gonder();
}//RDA end

//lcd_gotoxy(10,1);
//printf(lcd_putc,"%Lu %Lu",s_Zaman,s_Veri_geldi);

//if(input(Pin_D1)){
//delay_ms(50);
//while(input(b_Gonder));
//if(Modul_set(20,20)){
//*output_high(Pin_D0);
delay_ms(500);
output_low(Pin_D0);
delay_ms(500);
output_high(Pin_D0);
delay_ms(500);
output_low(Pin_D0);*/
//} else{
//*output_high(Pin_D0);
delay_ms(500);
output_low(Pin_D0);*/
```

EK 2. (Devam) Merkez Modülü Program Kodu

```
//}  
//}  
  
//if(input(b_Gonder)){  
    //Veri_topla();  
    //d_Tx[3]=make_CRC(d_Tx);  
    //Veri_gonder();  
    //b_1s=False;  
    //printf("B basildi");  
    //delay_ms(50);  
    //while(input(b_Gonder));  
    //SET_ADC_CHANNEL(0);  
    //delay_us(100);  
    //Anlg_value=READ_ADC();  
    //disable_interrupts(GLOBAL);  
    //printf("Anlg: %4LX",Anlg_value);  
    //lcd_gotoxy(1,2);  
    //printf(lcd_putc,"Anlg: %4LX",Anlg_value);  
    //enable_interrupts(GLOBAL);  
//}  
restart_wdt();  
}//whileend  
}//main end
```