

T.C.
MALTEPE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

RAYLI ULAŞIM ARAÇLARINDA YER TAYİNİ
VE YOLCU BİLGİLENDİRME SİSTEMİ

Yüksek Lisans Tezi

M. Akif ÖZKAYA

091405102

Tez Danışmanı

Yrd. Doç. Dr. Mehmet SAĞBAŞ

İSTANBUL – 2011

Bu tez çalışması, Maltepe Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun / / tarih ve / sayılı kararıyla oluşturulan jüri tarafından ***Yüksek Lisans Tezi*** olarak kabul edilmiştir.

JÜRİ

Yrd. Doç. Dr. Mehmet Sağbaş
Danışman

Üye

Üye

ÖZET:

Günümüzde raylı sistemlerde yön tayini için kullanılan GPS (Küresel Konumlama Sistemi - Global Positioning System) sistemlerinin doğruluk oranındaki düşüklük nedeni ile daha güvenilir alternatif yollar aranmıştır. RFID (Radio Frequency Identification) teknolojisi ile kapalı alan navigasyon sistemi ve bu sistemden alınan bilgiler doğrultusunda hazırlanan yazılımlar ile yolculara güvenli ve doğru yolculuk yapma olanağı sağlanmıştır. Bu sebeple bu çalışmada tren, metro, tramvay gibi raylı sistem araçlarıyla taşınan yolcuların yönlendirilmesi ve bilgilendirilmesi amacıyla kullanılmakta olan sistemler ve bu sistemlerin hazırlanmasında gerekli olan yer tayin sistemleri ele alınmıştır.

Bu çalışma, bir uygulama projesinin bir modeli olmakla birlikte fizibilitesi uygun çıkması durumunda İstanbul Büyükşehir Belediyesine bağlı ve İstanbul Ulaşım A.Ş. işletmesinde bulunan bir raylı sistem hattında kullanılması öngörülmüştür. Yapılan bu çalışma ile yurtdışına bağlı kalınan ve her türlü müdahalesi çok güç olan kapalı sistemler ve bu sistemlere gereken bakım, teknik servis gibi harcanılan ödeneklerin minimize edilmesi beklenmektedir.

Proje sonucunda, bir raylı sistem hattında yolcuların doğru bir şekilde bilgilendirilmesini sağlayacak, LED (Işık Yayan Diyod – Light Emitting Diode) ekranlar, LED paneller, ışıklı yön levhaları, sesli uyarı ve alarm donanımlarıyla birlikte tam bir sistemin nasıl yapılabileceği hususunda bilgi ve becerisinin kazanılması gerçekleştirilmiştir.

ABSTRACT

Today's RFID technology implementations "indoor navigation system and software designed from data acquired from this system" safe and right travelling enabled for passengers. For this reason in this study, systems which are used for the guidance, informing of the passengers and also necessities for location detection systems of railway vehicles like train, metro, tram are considered.

Although this study is a model for an implementation project; if it will be feasible it is anticipated that it will be applied at one of the operated railway lines of Istanbul Metropolitan Municipality subsidiary Istanbul Ulaşım Cooperation. With this study it is intended to minimize the fund allocated for these imported systems which are implicit, hard to intervene and also for the maintenance, services of them.

Following the project although it will be learnt about hardware used for safe travelling system like LED displays, LED panels, lighted direction boards, aural warning also knowledge and ability about designation of an entire system will be acquired.

TEŐEKKÜR

Bu alıőmanın gerekleőmesinde emeęi geen İstanbul Ulaőım A.Ő. Teknokent Ar-Ge Koordinatörü ve birim alıőanlarına, Maltepe Üniversitesi Fen Bilimleri Enstitüsü Müdürü Prof. Dr. İlhami Yavuz'a, Elektronik Mühendislięi Bölüm Başkanı Yrd. Do. Dr. Demir Öner'e, destekleriyle alıőmalarımı yapmama olanak tanıyan aileme ve danışmanım Yrd. Do. Dr. Mehmet Saębaő'a teőekkürlerimi sunarım.

İÇİNDEKİLER:

ÖZET:	iv
ABSTRACT	v
TEŞEKKÜR.....	vi
İÇİNDEKİLER:	vii
KISALTMALAR LİSTESİ:	ix
ŞEKİLLER LİSTESİ:	x
TABLolar LİSTESİ.....	xii
1. GİRİŞ	1
2. YOLCU BİLGİLENDİRME SİSTEMİ	4
3. YOLCU BİLGİLENDİRME SİSTEMİ EKİPMANLARI	7
3.1. Görsel ve İşitsel Ekipmanlar:	8
3.1.1. Tren yön levhaları:	8
3.1.2. Yolcu Bilgilendirme Panelleri:	10
3.1.3. Sesli İkaz Cihazları:	11
3.1.4. Aktif Hat Haritası:.....	11
3.2. Küresel Konumlama Sistemi (GPS):.....	12
3.3. RFID Sistemi:.....	13
3.3.1. RFID etiketleri:	15
3.3.2. RFID teknolojisinin kullanım alanları:	17
3.4. Haberleşme Arabirimi:	17
3.4.1. RS 232 haberleşme yapısı:	18

3.4.2.	RS 485 haberleşme yapısı:	19
3.4.3.	CAN-BUS haberleşme yapısı:	21
3.4.4.	Universal Serial Bus (USB) haberleşme yapısı:	23
3.5.	Otomatik Anons Ünitesi:	23
3.6.	Merkezi İşlem Paneli:	24
3.7.	Adreslenebilir Frekans Yükseltici (AMFI)	25
4.	SİSTEM SİMÜLASYONUNUN HAZIRLANMASI	27
4.1.	RFID Etiket ve RFID Okuyucu:	27
4.2.	Yazılım aşaması:	28
4.3.	Donanım Aşaması:	32
4.3.1.	Kontrolör Donanımı ve Elemanları:	33
4.3.2.	LED Ekranlar:	41
4.3.3.	Gömülü sistem yazılımı	43
4.4.	Fizibilite Çalışması:	46
5.	SONUÇ	51
	ÖZGEÇMİŞ	55
6.	EKLER	56
	EK 1 - Bilgisayar Yazılımı Kod Bloğu:	56
	EK 2 Örnek Model Elektronik Devre Şeması	64
	EK 3 Gömülü Sistem Yazılımı Kod Bloğu:	65
	EK 4 Akış Diyagramları	85

KISALTMALAR LİSTESİ:

Kısaltma	Açıklama
GPS	Global Positioning System – Küresel Konumlama Sistemi
RFID	Radio Frequency Identification
DTE	Data Terminal Cihazı – Data Terminal Equipment
DCE	Data Haberleşme Cihazı – Data Comunication Equipment
CAN	Controllor Area Network
EIA	Energy Information Administration
USB	Universal Serial Bus
PIB	Passenger Information Board – Yolcu Bilgilendirme Paneli
PID	Passenger Information Display – Yolcu Bilgilendirme Ekranı
DIF	Direction Information Front – Güzergah Tabelası Ön
DIS	Direction Information Side – Güzergah Tabelası Yan
LRM	Line Route Map – Aktif Ağ Hartası
PCB	Printed Circuit Board – Baskı Devre Kartı
CCS	Custom Computer Services
SIL	Saffety Integrity Level

ŞEKİLLER LİSTESİ:

Şekil 2-1 Yolcu bilgilendirme sistemi yapısı	5
Şekil 3-1 Yolcu bilgilendirme sistemi ekipmanları	7
Şekil 3-2 Tren Yön Levhası Ön ve Yan (DIF & DIS)	9
Şekil 3-3 Yolcu Bilgilendirme Panelleri (PIS & PID)	10
Şekil 3-4 Aktif Ağ Haritası (LRM)	11
Şekil 3-5 GPS Sisteminin Çalışma Prensibi	12
Şekil 3-6 RFID sistemi	13
Şekil 3-7 RFID reader	14
Şekil 3-8 RFID etiketler	15
Şekil 3-9 RS485 Bağlantı Yapısı	20
Şekil 3-10 Anons hoparlörlerinin çapraz bağlantı şeması	26
Şekil 4-1 Örnek proje, blok diyagramı	27
Şekil 4-2 Simülatör yazılımın ekran görüntüsü	28
Şekil 4-3 Donanım şeması	33
Şekil 4-4 PIC18F2585 pin bağlantı şeması	34
Şekil 4-5 Mikrokontrolör devre şeması	35
Şekil 4-6 MAX232 pin bağlantı şeması	36
Şekil 4-7 MAX232 entegresi ile RS232 bağlantı	36
Şekil 4-8 MCP2551 Pin bağlantı yapısı	37
Şekil 4-9 MCP2551 entegresi ile CAN-BUS bağlantı	37
Şekil 4-10 MAX487 Pin bağlantı yapısı	38
Şekil 4-11 MAX487 entegresi ile RS485 bağlantı	38
Şekil 4-12 Mikrokontrolör kartı PCB çizimi	39
Şekil 4-13 Donanım kısmı PCB kartı üst yüzey	40
Şekil 4-14 Donanım kısmı PCB kartı alt yüzey	40
Şekil 4-15 LED Ekran modülü ön yüz	41

Şekil 4-16 LED Ekran modülü arka yüz	42
Şekil 4-17 LED panel renkleri	42
Şekil 4-18 Bir tren için kullanılan görsel ekipmanlar	46
Şekil 6-1 Örnek model, elektronik devre şeması	64
Şekil 6-2 Ana program rutini akış diyagramı.....	85
Şekil 6-3 RS232 oku fonksiyonu akış diyagramı.....	86
Şekil 6-4 CAN-BUS yaz fonksiyonu akış diyagramı	87
Şekil 6-5 CAN-BUS oku fonksiyonu akış diyagramı	88
Şekil 6-6 LED panele veri yazma fonksiyonu	89

TABLolar LİSTESİ:

Tablo 3-1.RFID Etiketleri	16
Tablo 3-2. RFID Frekans Aralıkları.....	16
Tablo 4-1 RFID etiket okuma ve yer tayini	29
Tablo 4-2 PIC18F2585 in Tipik Özellikleri.....	34
Tablo 4-3 Gömülü yazılım, RS232 ile veri okuma	43
Tablo 4-4 RS232 ile aktarılan veri tipi.....	44
Tablo 4-5 Gömülü yazılım, CAN-BUS ile veri okuma	45
Tablo 4-6 Proje için gerekli LED panel modül adedi	47
Tablo 4-7 Proje için gerekli DC-DC dönüştürücü adedi.....	47
Tablo 4-8 Diğer ekipman ve malzemeler maliyetleri	48
Tablo 4-9 Toplam proje maliyeti	49
Tablo 4-10 Proje için alternatif maliyet tablosu.....	50

1. GİRİŞ

Büyük şehirlerde ulaşım ve trafik sıkışıklığı probleminin çözümü olarak raylı sistemlere önem verilmiştir. Ülkemizde ise 1900'lü yıllarda at ile çekilen ray araçları kaldırılmış, bu yollardaki raylar sökülüp yerlerine lastik tekerlekli araçların gidebileceği yollar yapılmaya başlanmıştır. Bu ülkemiz için bugün yaşanan trafik ve ulaşım sıkıntılarını başlangıç nedeni olmuştur. Her ne kadar 1900'lü yıllarda sistemler devre dışı bırakılsa da 90'lı yıllarda az gelişmiş raylı taşıma sistemleri ile raylı ulaşım başlamıştır.

Dünya üzerinde demiryolu sistemlerinin ilk gelişme evrelerinde, güvenlik tedbirlerine gerek olmayan kısa hatlar ve az sayıda yol kesişimi ile hızlı bir ulaşım aracı olarak görülmekteydi. İlk başlarda, ulaşım araçlarının yeri çok önemsenmiyor ve yolun daima boş ve açık olduğu varsayılıyordu. Fakat zaman içinde meydana gelen kazalardan dolayı, trenlerin yerlerinin belirlenmesi önem arz etmeye başladı. İlk önceleri, el veya bayrak ile işaret veren işaretçiler bu işi yapmaktaydı. İlerleyen zamanlarda ise vagon sayılarının artması ve makinistlerin görüş mesafesinin kısıtlı olmasından dolayı, bu işaretlerin tekrarlanması, yani işaretçilerin sayılarının çoğaltılması gereği doğdu. 1840'lı yıllarda zaman aralığı yöntemi uygulanmış; trenlerin belli bir süre içinde buldukları güzergahta karşılama ve varış süreleri tayin edilmişti. Ancak bu sistemde makinist bir öndeki trenin durumundan haberdar olamıyordu. Zaman aralığı yönteminin bu sorununa çözüm olması için mesafe aralık sistemi oluşturuldu. Bu sistemde güzergah bölümlere ayrıldı ve bir bölüme giren makinist, mekanik bir işaret ile o bölgeyi meşgul olarak diğer makinistlerin

görmesini sağladı. Telgrafın keşfinden sonra, zil ve telgraf birlikte kullanılmış ve trenleri gören operatörler trenlerin yerlerini birbirlerine aktarmışlardır.^{1,2,3}

Gelişen teknoloji ile operatörler tarafından yapılan bu algılama işlemleri sensörler ve elektronik devreler aracılığıyla yapılmaya başlandı. Mekanik anahtarlı algılama sistemleri ve ardından aks sayıcı anahtarlar kullanıldı. Sonraki dönemlerde izole cebreli ray devreleri ile ray bölgelerinin raylar arası elektrik değişimi ile trenler algılandı. Bu algılamanın adına hat bölümleri birbirinden izoleli olduğu için, izole cebreli ray devresi denilmiştir. Sonrasında ise kodlu ray devreleri geliştirilerek hattı izole etmeye gerek kalmadan belli bir frekansta gönderilen sinyalin bozulmasını ölçerek tren algılama işlemi geliştirilmiştir.⁴ Bu sistemlerin birçoğu günümüzde aktif olarak kullanılmaktadır.

Kablosuz haberleşme sistemlerinin kullanılmaya başlanması ile günümüzde kullanılan antenli sistemler ön plana çıkmıştır. Radyo frekans ve GPS li yöntemler ile yer tayini ve yönlendirme yapılmaya başlanmıştır. Yer altında GPS kullanılamaması durumundan dolayı, kapalı alanlarda doğruluğu yüksek seviyede olması beklenen sistemlerin araştırılmasına başlanmıştır. Uygulanmak istenen sistemlerin hepsi, EN 50155 kodlu “Demiryolu Araçlarında Kullanılan Elektronik Donanım” standardına tabi olmalı ve kullanım alanına göre SIL (Safety Integrity Level) güvenlik seviyelerine uygun olarak yapılmalıdır. Günümüzde kapalı alan yönlendirmelerinde en çok radyo frekans haberleşmeli sistemler kullanılmaktadır. 1970’lerden itibaren RFID, raylı sistemlerde kullanılmıştır. 1990’lı yıllarda Amerikan Demiryolu Birliği RFID sistemini endüstri standardı olarak kabul etmiştir.⁵ Ayrıca günümüzde RFID ile yer tayini ve yönlendirme sistemlerini cep telefonlarına entegre etmek üzere çalışmalar yapılmaktadır. Bu çalışmalara örnek olarak, 2010 yılında Nokia araştırma merkezi tarafından tanıtımı yapılan kapalı alan navigasyon sistemi (indoor navigation system) örnek olarak verilebilir.⁶ RFID sistemleri ile trenleri algılamanın yanında, bu trenlerin hız, yön, güzergahtaki hız limit bilgisi, bir önündeki aracın konum bilgisi gibi durumlar hakkında bilgi almak ve bilgi vermek mümkün hale gelmiştir.⁷

Öncelikli amacın kazaların önlenmesi olduğu, trenlerin yer tayini sistemi, birçok kullanım alanını da beraberinde getirmektedir. Bu çalışma, yolcu taşımada kullanılan raylı sistem araçlarının RFID sistemi ile yerinin tayin edilmesini ve bu bilgi ışığında yolcuların daha rahat bir şekilde yolculuklarını sürdürmeleri için gerekli yönlendirme işareti, anons ve yazıları ile yolcu bilgilendirme sistemi kurulumunu gerçekleştirecektir.⁸

RFID sistemlerin kullanılması kapalı alanlarda GPS'e oranla ek bütçeler gerektirir. Bundan dolayı dezavantajlı görünse de, GPS'in kapalı alanlarda çalışmamasından dolayı RFID sistemi tercih edilmektedir. Kullanım alanı ve sayısı arttıkça maliyetler azalmakta ve ek bütçeler uygun sınırlara gelmektedir.

Bu çalışmanın ikinci bölümünde, yolcu bilgilendirme sisteminin amacı ve olması gereken yönlerine yer verilmiştir. Üçüncü bölümde, yolcu bilgilendirme sistemlerinde bulunan ekipmanlar ve bu ekipmanların kendi aralarında ve dışarıdan yapılan bağlantılarda kullanılan haberleşme tekniklerine yer verilmiştir. Yurt dışındaki örnekler ışığında, ülkemizde kullanılması uygun olan parçaların avantaj ve dezavantajları ele alınmıştır. Dördüncü bölümde incelenen sistemler için bir simülasyon hazırlanmıştır. Buna ek olarak kurulması öngörülen sistemin bir örnek modeli, donanım ve yazılım olarak verilmiştir. Son olarak, yapılan fizibilite çalışması ile çalışma sonuca erdirilmiştir. Çalışmada elde edilen sonuçlar ise beşinci bölümde yer almaktadır. Ekler kısmında, simülasyonda ve örnek modelde kullanılan yazılım kodları ve donanımda kullanılan gömülü sistem kodlarına yer verilmiştir.

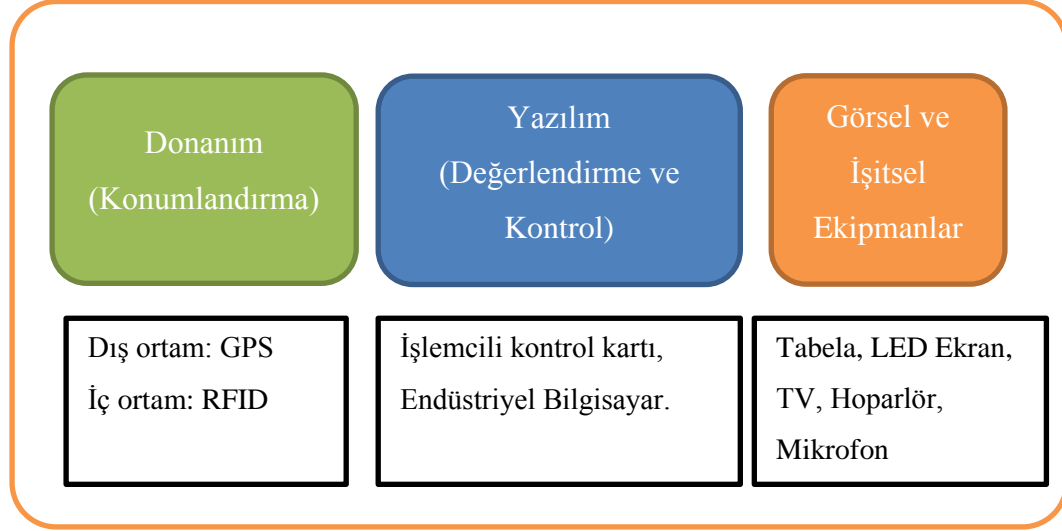
2. YOLCU BİLGİLENDİRME SİSTEMİ

Günümüzde özellikle yoğun nüfusun yaşadığı büyükşehirlerde yaşanan karmaşayı en aza indirmek ve geri döndürülmesi mümkün olmayan zaman kaybını en aza indirebilmek için, günlük yaşantımızda bazı yönlendirmelere ihtiyaç duyarız. Evimizden çıktığımız ilk andan itibaren bazı yönlendirme işareti veya sesli ya da görsel yönlendirmeler ile vaktimizi en uygun şekilde kullanmayı ve yapacağımız işin sonucuna gereken zaman diliminde ulaşmayı hedefleriz. İlk defa yapacağımız bir işlem için yola çıktığımızda, diğer zamanlardan daha çok, yönlendirme levhalarını takip eder ve yolumuza yön vermeye çalışırız. Hemen her gün gittiğimiz yollarda veya yaptığımız işlerde, bu yönlendirme tabelaları bizim için normal haline gelmiştir ve ilgimizi çekmezler. Bu gibi durumlarda da ışıklı ve sesli yönlendirmeler ön plana çıkar. Bu yönlendirme araçları, her gün yapmakta olduğumuz işi, stres ve dalgınlıkla meydana gelebilecek şaşırılardan bizi kurtarabilecek ilgimizi toplamamıza yardımcı olabilecek niteliktedirler.

Yolcu bilgilendirme sistemleri günü birlik seyahat eden vatandaşların, turistlerin, her gün aynı güzergahı kullanan yolcuların tamamını ilgilendiren ve bu kişilere karmaşa oluşturmadan, rahatsızlık vermeden, amacı doğrultusunda bilgi vermek amacıyla hazırlanmış sistemlerdir. Aynı güzergahı kullanan vatandaş için anlık dalgınlıkları ve şaşırılmayı minimize etmekle birlikte bölgeye tamamen yabancı kişilerin ilk seferde dahi bölgeye hakim olduklarını ve istekleri doğrultusunda kendilerine yön vermelerini sağlamaktadır.

Yolcu bilgilendirme sistemlerinin günümüzde kullanımı oldukça yaygınlaşmıştır ve yaygınlaşmaya devam etmektedir. Bu kullanım, gelişen teknoloji sayesinde oldukça hızlı bir artışı da beraberinde getirmektedir. Ancak bazı kullanım alanlarında, birtakım kısıtlamalar ve zorluklar meydana gelmektedir. Bu zorunlu

durumlar için gelişen teknoloji ve yeni fikirler ile sistemlerin sorunlarının önüne geçilmektedir.



Şekil 2-1 Yolcu bilgilendirme sistemi yapısı

Bu çalışmada raylı sistemler için kullanılan yolcu bilgilendirme sistemleri için gerekli konumlandırma, kontrol ve yönlendirme sistemlerinin tümü bir uygulama dahilinde incelenecektir. Şekil 2-1’de yolcu bilgilendirme sisteminin yapısı görülmektedir. Bu yapıda, çalışmada ele alınan algılama, kontrol ve sonuç kısımları görülmektedir. Algılama kısmında sensörlerden alınan ve konumlandırma bilgisini elde ettiğimiz donanım ekipmanları bulunur. Bu ekipmanlar açık alanda GPS, kapalı alanda ise RFID olarak öngörülmüştür. Kontrol kısmında, sensörlerden alınan verilerin işlenip bir kontrolünün yapılması ve ardından sonuç kısmına iletilmesi amacıyla kurulan yazılım kısmı bulunur. Sonuç kısmında ise yazılım sonucunda elde edilen verilerin donanımlar aracılığı ile dış dünyaya aktarma ekipmanları bulunur.

Bilindiği üzere Küresel Konumlama Sistemi olarak adlandırılan ve daha çok GPS olarak günümüzde anılan sistemler ikiden fazla uydu ile haberleşerek, uyduların yerine göre kendi konum bilgisini üreten sistemlerdir. Bu konum bilgisinin doğruluğu mevcut uydu bağlantı sayısı ile doğru orantılıdır.⁹ Konumlama bilgisinin bulunmasının ardından yönlendirme için gerekli ekipmanlar ile yolcu bilgilendirme

sistemi kurulmaya başlanmış olur. Bu ekipmanlar için LCD ekranlar, LED paneller yönlendirme levha ve tabelaları gibi örnekler verilebilir. Bir kontrol bilgisayarı üzerinde veya mikroişlemcili bir devre vasıtasıyla bu sistemleri yönlendirebilecek yazılımlar kullanılmalıdır.

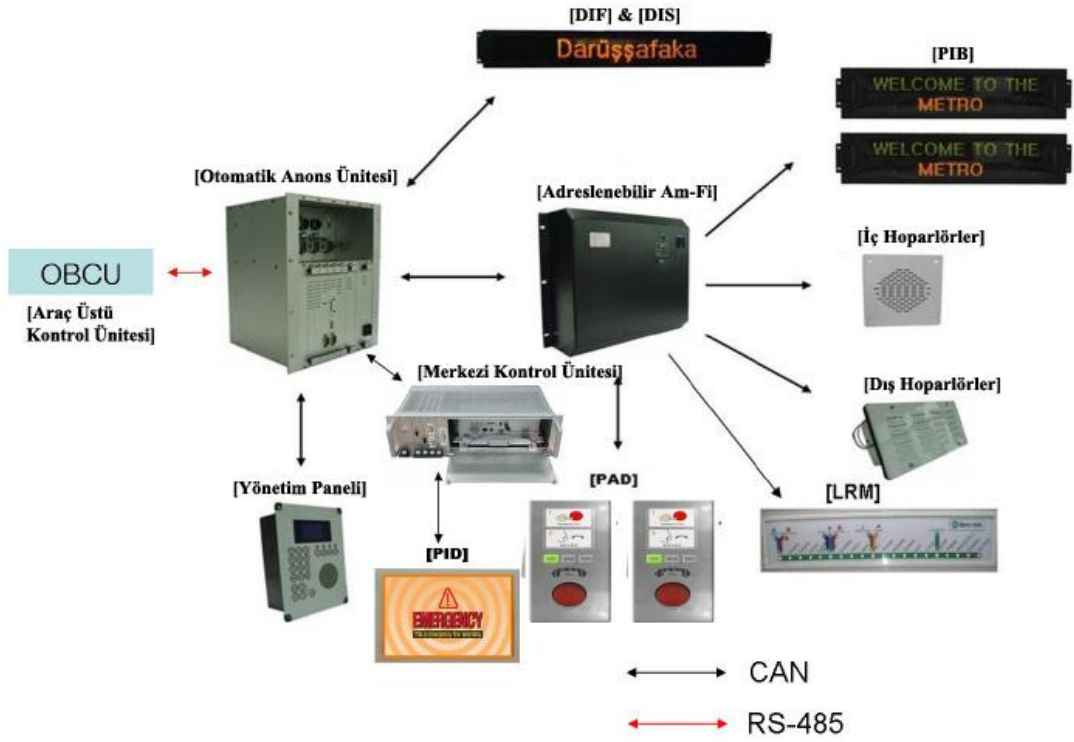
En güvenilir taşıma biçimi olan raylı sistemlerde güvenlik en üst düzeyde tutulmaktadır. Raylı sistemlerde kullanılacak sistemlerin titreşim ve nem gibi yüksek durumlu olgulara karşı en üst seviyede doğruluğa sahip olması beklenilmektedir.¹⁰ Bu durumda, gerek sapmaların meydana gelebiliyor olmasından ve gerekse de yer altında atmosferdeki uydulardan hiçbir konum bilgisi alınamayacağından, raylı sistemler için GPS kullanımı uygun olmamaktadır.

Uygulama alanında sadece tek düzlemde hareketi algılamak ve bu algılamayı, güvenlik unsurunu göz ardı etmeden başarmak için RFID sistemlerin kullanılması mümkündür. RFID sistemleri ile hem konum bilgisini üretebilecek kablosuz bir donanım elde edilebilir, hem de bu donanım ile bazı verilerin kablosuz olarak aktarılması işlemi gerçekleştirilebilir.

Öncelikle yolcu bilgilendirme sistemleri ve bu sistemlerin oluşumunda kullanılan sistemleri ve karşılaşılan engellere karşı geliştirilmiş yöntemler üçüncü bölümde detaylıca incelenecektir.

3. YOLCU BİLGİLENDİRME SİSTEMİ EKİPMANLARI

Bu bölümde, yurt içi ve yurt dışında incelenen yolcu bilgilendirme sistemlerinde kullanılan tüm ekipmanlar incelenmiştir. Sistemlerde kullanılan ekipmanlar, ihtiyaçlar doğrultusunda artırılabilir veya sistemden çıkarılabilir.



Şekil 3-1’de detaylı bir şekilde yolcu bilgilendirme sistemine ait ekipmanlar görülmektedir. Burada interaktif hat haritası (LRM - Line Route Map), Yolcu bilgilendirme paneli (PIB), ön ve yan güzergah tabelaları (DIF-DIS), yolcu bilgilendirme ekranı (PID) ürünleri yolcuya görsel bilgi veren ekipmanlardır. Buna karşılık, iç ve dış hoparlörler ve yolcu alarm cihazı (PAD) gibi ürünler sesli ikaz ve yönlendirme görevindedirler. Bu cihazların yönetimi, kontrolü ve verilerin

saklanması için kullanılan diğer ekipmanlar da sırasıyla; yönetim paneli, merkezi kontrol ünitesi, otomatik anons ünitesidir.

3.1. Görsel ve İşitsel Ekipmanlar:

Yolcu bilgilendirme sistemleri genel olarak toplu taşıma araçlarında ve bu araçların duraklarında yolcuların doğru yönlendirmelerini sağlamak amacıyla hizmete sunulmuş sistemlerdir. Genel gereksinimi bu olan sistemlerin ihtiyaç doğrultusunda gelişmesi ve kapsamının genişletilmesi meydana gelmiştir.

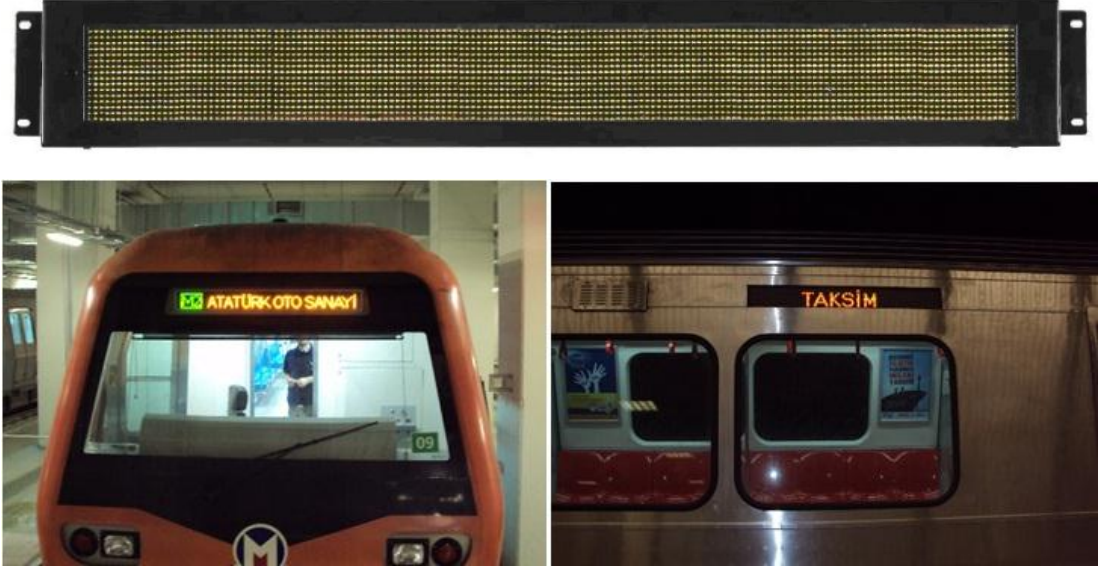
Sistemlerde kullanılan ekipmanlar, görsel ve işitsel tarzda vatandaşa hizmet veren sistemlerdir. Aşağıda bir yolcu bilgilendirme sisteminde kullanılabilecek genel görsel ve işitsel ekipmanlar listelenmiştir.

- LCD Ekranlar
- LED Paneller
- Yönlendirme panoları
- Sesli ikaz hoparlör ve megafonları
- Mikrofonlar

3.1.1. Tren yön levhaları:

Tren yön levhaları trenin iki başında ve yanlarında bulunan görsel levhalardır. Bu levhalar ilk önceleri rulo halindeki bir brandanın döndürülmesi ile değişen göstergeler şeklindeydi, ancak daha sonra elektromekanik olarak değişen küçük parçalar halini aldı ve ardından da LED ampullerin yaygınlaşması ile LED paneller şeklinde tasarlanmıştır. LED paneller uzak mesafelerden rahat görüntülenmeyi

sağlaması, düşük enerji tüketmesi, bakım maliyetlerinin çok az olmasından dolayı kısa zamanda eski sistemlerin yerini almıştır.



Şekil 3-2 Tren Yön Levhası Ön ve Yan (DIF & DIS)

Tren yön levhalarında hatta çalışan aracın gideceği son istasyon bilgisi verilir, aracın ön ve arka kısmında bulunan yön levhalarında bu bilgiye ek olarak, sefer yapılan hattın numarası yazılır. Bu gösterim Şekil 3-2’de verilmiştir. Araç yanlarında bulunan yön levhaları ise istasyon dahilinde yan taraftan görülebilecek ve gidilecek istasyonun bilgisini gösteren levhalardır. Bu levhalarda gidilecek son istasyon verisinin yanında o doğrultudaki diğer merkezi istasyonların bilgisi de dönüşümlü olarak gösterilebilir.

Bu projede, hazır LED modüllerin birleştirilmesi ile meydana getirilen ve PIC mikroişlemciler ile yönetilen yönlendirme levhaları kullanılacaktır. İstek ve ihtiyaca göre, tek renkli, üç renkli ve tüm renkli olarak levhalar yapılması mümkündür. Genel kullanımda tek renkli LED tercih edilir. Ancak iki renkli LED’ler ve bunların aynı anda yakılmasıyla elde edilen üçüncü bir rengin bulunduğu, üç renkli LED panellerin kullanılması görsel açıdan daha estetik bir sonuç sağlamaktadır.

3.1.2. Yolcu Bilgilendirme Panelleri:

Yolcu bilgilendirme panelleri araç içlerinde bulunan ve öncelikli amaçları yolculara bir sonraki istasyon, gelinen istasyon ve bu istasyonlarda çıkış kapısı yönlerini belirten panellerdir. Bu panellerde kullanım isteğine göre LED veya LCD ekran şeklinde olanlar mevcuttur ve bunların örnek gösterimi Şekil 3-3 de verilmiştir. LCD ekran ihtiyacı bilgi aktarımı olmadığı zamanlarda aynı panelin reklam göstermek amacıyla kullanılmasıyla oluşur. Bu gibi bir durum söz konusu değilse veya reklam göstermek ve bilgi vermek amacıyla iki ayrı panel kullanılacaksa bilgi panelinin LED paneller ile sağlanması okunabilirlik açısından daha uygundur.



Şekil 3-3 Yolcu Bilgilendirme Panelleri (PIS & PID)

3.1.3. Sesli İkaz Cihazları:

Raylı sistem araçları içinde, yolcu bilgilendirme sistemi amacıyla kullanılan, iç-dış hoparlörler, megafonlar ve sirenler gibi cihazları bu grubun elemanları olarak saymak mümkündür. Bu kısımlar gerekli uyarıları yolcuya aktarma vazifesini görmektedirler. Ayrıca tüm bu cihazları yöneten ve onlara bilgi aktaran yönetici cihazlar mevcuttur. Bu yönetim cihazlarını ilerleyen kısımlarda Otomatik Anons Ünitesi ve Adreslenebilir Frekans Yükseltici (AM-FI) başlıkları altında incelenecektir.

3.1.4. Aktif Hat Haritası:

Hat haritası, raylı sistem araçlarında seyahat edilen hattın duraklarının sıralandığı bir levhadır ve gösterimi Şekil 3-4'de verilmiştir. Aktif hat haritası ise mevcut olan hattın bulunan ve gidilmekte olan istasyonlarının ışıklı bir şekilde yolculara bildirilmesini sağlayan haritadır.

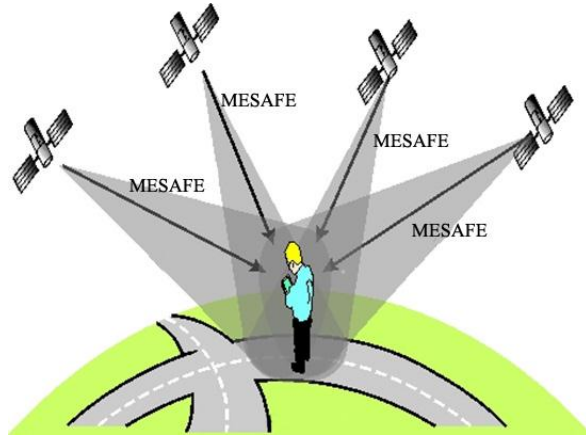


Şekil 3-4 Aktif Ağ Haritası (LRM)

3.2. Küresel Konumlama Sistemi (GPS):

GPS (Global Positioning System), atmosfer de bulunan uyduların konum bilgileri üzerinden hesaplama ile yer tayini yapılmasına olanak tanır. Genelde navigasyon sistemlerinde kullanılan GPS, koordinatları belli olan harita çizimleri üstüne GPS verisindeki konum bilgisi ve yol tayin yazılımlarındaki algoritmalar vasıtasıyla kullanıcılara hizmet vermektedir.

GPS sistemi yer tayinini ikiden çok uydu ile haberleşerek arada kurulan bağlantı ile kendi noktasını tayin eder. Bu nokta tayini minimum üç adet uydu ile sağlanabildiği gibi daha çok sayıda uyduya bağlanarak, nokta tayinindeki sapmalar daha aza indirilmektedir.¹²



Şekil 3-5 GPS Sisteminin Çalışma Prensibi¹³

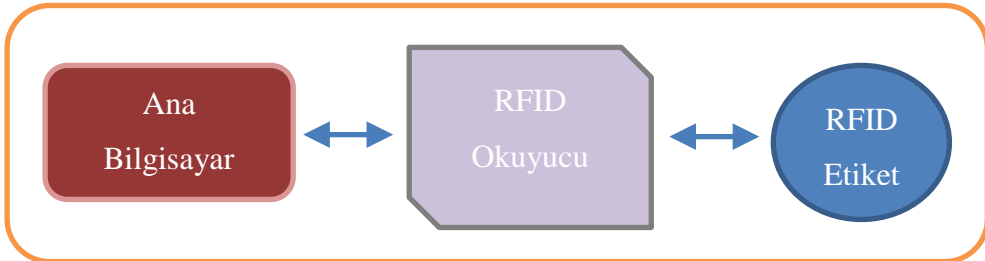
Sapma oranını, havanın durumuna göre değişebildiği gibi GPS anteninin konumu ile de oldukça önemli ölçüde değişebilmektedir. Güvenilirliğin en üst seviyede olması gereken raylı ulaşım araçlarında hem bu sebepten hem de kullanılan bölgenin yer altında olmasından dolayı GPS sistemi kullanılamamaktadır. Tüm dünya üzerinde rahatlıkla kullanılan ve hiçbir ek ücret ödemedi konum bilgisini alabilme ihtimalinin ortadan kalkması durumunda ise raylı ulaşım sistemlerinde bir

yolcu bilgilendirme sistemi veya sinyalizasyon sistemi kurmak için gerekli konumlama sisteminizi kendiniz yapmanız gerekebilir. İster istemez maliyetlerin attığı dâhili konumlama sistemi, GPS'e göre daha güvenilir olduğundan kaçınılmazdır.

Bu çalışmada, dahili konumlandırma sistemi olarak RFID sistemleri kullanılacaktır.

3.3. RFID Sistemi:

RFID, üzerinde mikroişlemci ile donatılmış etiket taşıyan bir nesnenin, bu etikette taşıdığı kimlik yapısı ile hareketlerinin izlenebilmesine imkân veren radyo frekansları ile çalışan teknolojiye verilen addır. RFID kelimesinin açılımı İngilizce olarak Radio Frequency Identification'dır. Oldukça eski bir teknoloji olan RFID'nin kullanımı, ikinci dünya savaşı yıllarına kadar uzanmaktadır. Ancak etiketlerin maliyetlerinin yüksekliği ve kullanım zorluğu, RFID teknolojisinin uzunca bir süre şirketler tarafından kullanılamamasına neden olmuştur.¹⁴ Maliyetlerin azalması ve teknolojinin ucuzlamasıyla günümüzde RFID sistemlerinin kullanımı yaygın hale gelmiştir.¹⁵



Şekil 3-6 RFID sistemi

Şekil 3-6'da RFID teknolojisinin bölümleri görülmektedir. RFID teknolojisinde temel olarak, RFID etiketi ve RFID okuyucusu en kritik bileşenlerdir. Bunlara ayrıca RFID yazıcısı, RFID anteni, sistemin kullanacağı yazılımı ekleyebiliriz. Bir RFID etiketi tümdevre (çip), güç kaynağı ve antenden oluşmaktadır. Etiket, bu sayede RFID okuyucularıyla iletişim kurabilir ve veri aktarır alabilir.



Şekil 3-7 RFID reader

Günümüzde oldukça düşen etiket fiyatları, özellikle 2007-2008 yıllarında dalga dalga RFID kullanımını tetikleyecek unsur olarak görülmektedir. Şu anda özellikle büyük perakendeciler, market zincirleri ve ordu RFID teknolojisini etkin bir şekilde kullanmaktadır. RFID'nin etkileyeceği ve yerini alacağı yegane teknoloji barkod teknolojisidir. Sadece dünyada değil, ülkemizde de çok yoğun bir şekilde kullanılan barkodun RFID'ye göre bazı dezavantajları vardır. Örneğin;

- Görüş sahası
- Sadece veri okuma
- Kısıtlı alan ve okuma oranı gibi.

Ancak barkod teknolojisinin şu anda RFID'ye göre çok güçlü bir özelliği bulunmaktadır. Bu özellikte etiket başına maliyetidir. Olaya bir de RFID yönünden

bakılırsa, kolaylıkla görülebilir ki, maliyet dışında RFID teknolojisi barkoda göre oldukça avantajlıdır. Çünkü RFID'de;

- Görüş sahası kavramı yoktur.
- Etiket üzerinde veri okuma/yazma işlemi yapılabilmektedir.
- Uzun mesafe ve her okumada birden çok kalem mal okunması imkanı vardır. Bu sayede ambarınızda ya da deponuzdaki stoklarınızı haftalar - günler - saatler yerine dakikalar içinde sayabilirsiniz.
- İnsan / operatör müdahalesine gerek yoktur. Bu sayede uzun dönem maliyetlerinizi tahmin edemeyeceğiniz kadar düşürebilirsiniz.
- Daha fazla veri taşınabilir. Bu sayede sadece ürünün kodu değil, gerekiyorsa geldiği depo, son kullanma tarihi ya da istenen başka bilgiler etiketlere yüklenebilir.

3.3.1. RFID etiketleri:

Şekil 3-8'de bir çok uygulama tarafından kullanılan, çeşitli tiplerde ve boyutlarda RFID etiketler görülmektedir.



Şekil 3-8 RFID etiketler¹⁶


RFID etiketleri temel olarak ikiye ayrılmaktadır. Bunlar aktif ve pasif etiketlerdir. İki etiket türü arasındaki farklar **Error! Reference source not found.**'de kolaylıkla görülebilir.

Tablo 3-1.RFID Etiketleri

	Aktif	Pasif
Güç Kaynağı	Etiket içinde (Bakım gerekli)	Radyo dalgaları ile enerji alıyor.
Çalışma Sıcaklığı	Kısıtlı	Geniş Sıcaklık Aralığı (-40°C – +85°C)
Mesafe	Uzun	Kısa
Bellek Kapasitesi	Büyük	Küçük
Fiyat	10 – 100 USD	15 Cent – 1 USD

RFID etiketlerinin ayırt edici bir diğer özelliği de çalıştıkları frekans aralıklarıdır. RFID etiketlerinin çalışma frekanslarını Tablo 3-2’de görebilirsiniz. Günümüzde yaygın olarak kullanılan etiketlerin frekansları daha çok HF (High Frequency) aralığındadır. Bir etiketin maliyeti sadece aktif ya da pasif olmasına göre değil, çalıştığı frekansa göre de değişmektedir. Endüstride 13.56 MHz'lik etiket kullanılan projeler, şirketlere maliyetleri açısından da oldukça çekici gelmektedir. Ancak bazı uygulamalarda UHF frekans aralığı (800-900 MHz) kullanılması performans adına daha iyi sonuçlar vermektedir.¹⁷

Tablo 3-2. RFID Frekans Aralıkları

Frekans		Microwave (1GHz+) Mesafe: 1 – 10 m.
		UHF (300Hz – 1000MHz) Mesafe: 1 – 3 m.
		VHF (30 – 300 MHz) Mesafe: 1 – 3 m.
		HF (3 – 30 MHz) Mesafe: 1 – 70 cm.
		LF (3 – 300 KHz) Mesafe: 1 – 150 cm.

Herhangi bir projede hangi frekans aralığını kullanan etiketlere karar verilmesi, o projenin düşünölen ortamda gerçekleştirilip gerçekleştirilemeyeceđi kararı gibi kritik kararlar tamamen tecrübe ve bilgi birikimi ile donatılmış uzman ekipler sayesinde olmalıdır. Aksi halde RFID projeleri, maliyetleri ve sonunda yaşanan koca bir başarısızlık ile birçok şirket ve kuruma faydadan çok zarar getirmiştir.

Bu çalışmada uygulama simölasyonu için pasif RFID etiketler kullanılacaktır. Simölasyon için gerekli güvenlik limitlerini ve küçük boyutları sağladığı için pasif RFID etiketlerinin kullanılmasında herhangi bir sakınca yoktur. Zaten gelişen teknoloji ile pasif RFID etiketlerinin bu tarz projeler için boyut ve sıcaklık aralığı kısıtlamaları yeterli düzeye çıkarılmıştır.

3.3.2. RFID teknolojisinin kullanım alanları:

RFID sadece tek bir alan ya da sektörde kullanılmaz. Bugün büyük alışveriş merkezlerinde, market zincirlerinde, hayvan takiplerinde, havayolları-kargo şirketlerinde kullanılan örnekleri vardır. Bu teknoloji ile şirketlerin avantajları, zamanla azalan insan gücü maliyeti, otomatikleştirilmiş stok kontrolü, ürün takibi ve anında ulaşılabilen envanter bilgisidir. Ayrıca RFID sayesinde şirketlerin iş süreçleri hızlanacak ve ihtiyaç duyulan gelişmiş raporlar hızlı ve doğruluđu yüksek bir şekilde oluşturulabilecektir.

3.4. Haberleşme Arabirimi:

Proje için kullanılması öngörölen birçok sistem için bir yönetici mekanizma olmalı ve bu mekanizma ile birden fazla sistemin haberleşmesini sağlanmak

gerekmektedir. Proje içindeki tüm birimlerin aynı haberleşme sisteminde çalışması ideal bir durum olarak görülebilir ancak bu bazen mümkün olmayabilir. Farklı haberleşme sistemlerini birbirine uydurmak için ya dönüştürücü devreler ya da her haberleşme algoritmasına uyum sağlayacak ortak bir işlemcili devre hazırlanabilir. Günümüzde yaygın olarak kullanılmakta olan Microchip firmasının PIC mikrodenetleyicilerinde isteğiniz doğrultusunda birçok haberleşme yeteneği tek bir tümdevre içinde bulunmaktadır.

Projede, RS485 ve CAN-BUS iletişim yapılarını kullanılmıştır. RS485 raylı sistem araçlarından bilgi alabilmemiz için gerekli olurken, CAN Bus kendi hazırlamış olduğumuz birimler arasında haberleşmeyi sağlamaktadır. Bunlara ek olarak görsel simülasyonda kullanılmak üzere RS232 ve USB (Universal Serial Bus) bağlantı yapılarını da kullanmaktayız.

3.4.1. RS 232 haberleşme yapısı:

RS232, RS422, RS423 ve RS485 bilgisayarlar ve diğer elektronik cihazlarda kullanılan seri haberleşme standartlarıdır. Kuşkusuz ki RS232 bu seri haberleşme metodları arasında en iyi bilinenidir.

RS232 maksimum 20 kbps veri iletim hızında sadece bir DTE (Data Terminal Cihazı)'nin sadece bir DCE (Data Haberleşme cihazı)'ye bağlanabildiği bir haberleşme ara yüzüdür ve bu iki cihaz arasındaki maksimum kablo uzunluğu 15 metre olabilir. Bu mesafe ilk zamanlarda yeterli gelmekteydi ancak daha sonra teknoloji ve buna bağlı olarak ihtiyaçlar değişmiştir.¹⁸ Bu ihtiyaçlar aşağıdaki gibi sıralanabilir.

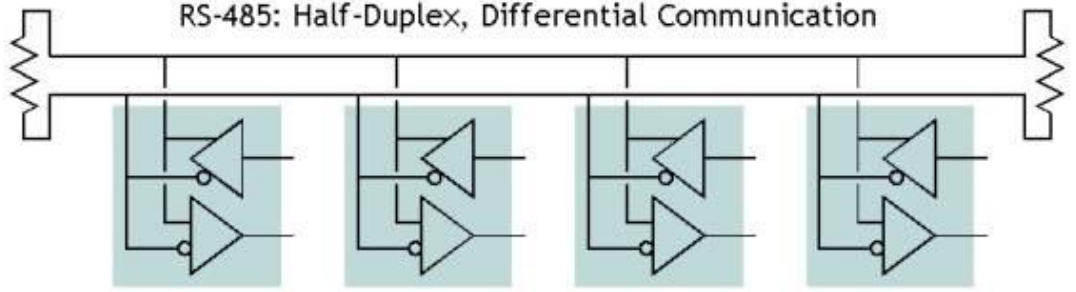
- Daha uzun mesafede haberleşme
- Birden fazla DTE bağlama
- Daha hızlı haberleşme

RS485, EIA (Energy Information Administration) tarafından tanımlanmış çok yönlü bir seri haberleşme standardıdır. Yukarıda belirtilen ihtiyaçların hepsini sağlar. Bu yüzden birden fazla cihazın birbirleriyle haberleşmesi gereken veri işleme ve kontrol uygulamalarında yoğun bir şekilde kullanılır. Raylı sistem araçlarının uzun olduğunu ve elektrikle çalışan bu araçların farklı enerji değerlerinden dolayı sinyal gürültü oranlarını düşürürsek RS232 ye karşın neden RS485 in seçilmesini daha kolay kavrayabiliriz.

3.4.2. RS 485 haberleşme yapısı:

RS232'nin en temel problemi sinyal hattı üzerindeki gürültüden kolay etkilenebilir olmasıdır. RS232 protokolü alıcı ve verici arasındaki veri ve hat gerilimi ortak bir toprak hattı kullanarak karşılaştırmaktadır. Toprak hattındaki herhangi bir voltaj artımı yanlış iletişim sonuçları doğuracaktır. Bu yüzden RS232 tetikleme seviyesi +/- 3 V.'a ayarlanmıştır. Bu nedenle mesafe arttığında gürültü hızla artar. RS485 standardında ise sinyal referansı için ortak sıfır kullanılmaz. Bu sebeple RS485 alıcı ve verici ünite arasındaki voltaj seviye farkı bir problem oluşturmaz. RS485 sinyalleri değişkendir ve her bir sinyal Sig+ ve Sig- hatları üzerinde iletilir. RS485 alıcısı sinyal hattı üzerindeki kesin voltaj seviyesi yerine iki hat arasındaki voltaj farkını karşılaştırır. Bu sayede birçok haberleşme sorununun temeli olan toprak döngüsü önlenmiş olur.

RS485'in ağ yapısı veri işleme ve kontrol uygulamalarında yoğun bir şekilde kullanılmasının ana nedenidir. 12 K Ω giriş direnci ile ağa 32 cihaza kadar bağlantı yapılabilir. Daha yüksek giriş direnciyle bu sayı 256'ya kadar çıkarılabilir. RS485 tekrarlayıcıları ile bağlanabilecek cihaz sayısı birkaç bine, haberleşme mesafesi de birkaç kilometreye çıkabilir. RS485 bunun için ayrıca bir donanım istemez, yazılım kısmı da RS232'den zor değildir.¹⁹



Şekil 3-9 RS485 Bağlantı Yapısı

Şekil 3-9’da RS485 ağ yapısı verilmiştir.²⁰ N kadar düğüm çok noktalı RS485 ağına bağlanmıştır. Hattın iki ucundaki R dirençleri 100 Ω seçilerek yansıma önlenmiş olur böylece daha yüksek hız ve daha uzun mesafeye erişilmiş olur.

- RS485 in belli başlı teknik özelliklerini şu şekilde sıralayabiliriz;
- Maksimum sürücü sayısı: 32
- Maksimum alıcı sayısı: 32
- Çalışma şekli: Half Duplex
- Network Yapısı: Çok noktalı bağlantı
- Maksimum Çalışma Mesafesi: 1200 metre
- 12 m kablo uzunluğunda maksimum hız: 35 Mbps
- 1200 m kablo uzunluğunda maksimum hız: 100 Kbps
- Alıcı giriş direnci: 12 K Ω
- Alıcı giriş duyarlılığı : +/- 200 mV.
- Alıcı giriş aralığı : -7...12 V.
- Maksimum sürücü çıkış voltajı : -7...12 V.
- Minimum sürücü çıkış voltajı (yük bağlı durumda) : +/- 1.5 Volt

3.4.3. CAN-BUS haberleşme yapısı:

RS232 ve RS485 yapılarında bazı kısıtlamaların olduğu ve raylı sistem araçlarının yüksek güvenlik gereksinimleri daha hızlı ve daha güvenli iletişim sistemleri için alternatifler incelenmiştir.

Açılımı, “Controller Area Network **Bus**” olan yani “Kontrol Alan Ağı Veri yolu”dur. 1980’lerde Robert Bosch tarafından otomotivde kablo yumağı yerine bir kablodan yazılım kontrollü veri transferini sağlamak amacıyla geliştirilmiştir. CAN, otomotiv endüstrisindeki en bilinen haberleşme sistemidir. Her ne kadar başlangıçta yalnızca otomotiv uygulamaları için tasarlanmış olsa da yüksek performanslı güvenilirliğinden dolayı birçok dağıtık endüstriyel kontrol uygulamalarında yaygın olarak kullanılmaktadır. Güvenliğin çok önemli olduğu gerçek zamanlı uygulamalarda da kullanılır. Öyle ki istatistiksel olasılık hesapları sonucunda bir asırda bir tane tespit edilemeyen mesaj hatası yapabileceği tespit edilmiştir.²¹

Uygulama alanı yüksek hızlı ağlardan düşük maliyetli çoklu kablolamalı sistemlere kadar geniştir. CAN-BUS, otomobil elektroniği, akıllı motor kontrolü, robot kontrolü, akıllı sensörler, asansörler, makine kontrol birimleri, kaymayı engelleyici sistemler, trafik sinyalizasyon sistemleri, akıllı binalar ve laboratuvar otomasyonu gibi uygulama alanlarında maksimum 1Mbit/sn lik bir hızda veri iletişimi sağlar.²²

CAN-BUS sisteminin sahip olduğu özellikler:

- Mesaj önceliği
- Kayıp zaman güvenliği
- Yapılandırma esnekliği
- Senkronizasyonlu çoklu kabul: Aynı veri birçok ünite tarafından alınabilir
- Sistemdeki veri yoğunluğunu kaldırabilme

- Çok efendili (Multimaster) çalışma
- Hata tespiti ve hataya ilişkin sinyalleri üretme
- Mesajın yollanmasında hata oluşması halinde mesajın iletim hattının (BUS) boş olduğu bir anda mesajın otomatik tekrar yollama
- Ünitelerde oluşan geçici ve kalıcı hataları ayırt edebilme ve özerk olarak kalıcı hatalı üniteleri kapatabilme

CAN-BUS sisteminde tüm üniteler iletim hattına eşit öncelikli veri yollama hakkına sahiptirler. Buna multimaster çalışma denir. Ancak tüm düğümler bir anda veri yollamaya çalışırsa çatışmalar olacağı açıktır. Bunun çözümü ise şu şekilde sağlanmıştır; tüm üniteler her daim iletim hattını dinlemek zorundadır. Yani her ünite iletim hattının boş olduğu anı yakalamaya çalışır. Hattı boş gören verisini yollar.

Tüm ünitelerin eşit mesaj yollama önceliği olsa da işin aslı biraz farklıdır. Bunun sebebi, CAN'ın mesaj öncelikli bir sistem olmasından dolayıdır. İnternette tüm PC'lerin bir numarası (IP Adresi) vardır ve veriler o IP'li kişiye yollanır. Zaten bu yüzden bol kablolamalı bir sistemdir. Ancak CAN'de ünitelere değil mesajlara numara verilir. Örneğin bir otomobil düşünelim. Aceleci şoförümüz yolda hızlıca giderken bir arabaya çarpacak olsun. Şoför çarpacağını fark eder ve tüm gücüyle frene basar. Burada CAN'ın önemi arabaların kaza yapması ile başlar. Mesela önden çaptı ve sensörler bunu fark ettiğinde hava yastıklarının açılması, yakıtın kesilmesi gibi bir dizi önlemler alınmalıdır. Bu işlerden sorumlu tüm ünitelere 1 numaralı -kaza oldu güvenlik sistemleri devreye girin- mesajı iletilmelidir. Bunu sensörlü sistem diğer ünitelere söyleyecektir. Ancak o ara başka bir ünite mesela motor ısısının kaç derece olduğunu kokpitte ki LCD li üniteye yollamaya çalışsın. Sizce hangisi şu anda daha önemli! İşte CAN tek kablo üzerinden eşit erişimli mesaj yollanmasına izin verir ama öncelikli mesajların daha önce iletim ortamına iletilmesine de özen gösterir.

3.4.4. Universal Serial Bus (USB) haberleşme yapısı:

Bu çalışmada, USB haberleşme yapısını ileriye dönük olarak bilgisayarlar ile haberleşme ve aynı zamanda çalışma simülasyonlarını gerçekleştirmek için kullanacağız.

USB, Türkçe'ye Evrensel Seri Veri yolu olarak çevrilmiştir. Tak-Kullan aygıtların sistemlere eklenmesini kolaylaştırmak için geliştirilmiş bir iletişim sistemidir. Günümüzde USB 3.0 versiyonu ile iletim hızı 600MBps'ye çıkmıştır. En yaygın olarak kullanılan USB 2.0 da ise iletim hızı 60MBps dir. Bu çalışmada, RFID etiketlerin okunmasını USB iletişim kullanan RFID okuyucu ile sağlayacağız. Ancak proje nihayetinde endüstriyel tipte ve farklı iletişim sistemleri kullanan okuyucular seçilebilir.

3.5. Otomatik Anons Ünitesi:

Raylı sistem araçları içinde yapılacak anonsların kayıtlı bulunduğu birimdir. Bu üniteye aracın sefer yaptığı hattaki durakların anonsları, bu durakları yaklaşım anonsları, acil durum anonsları, duyuru anonsları ve reklam anonsları bulunur.

Otomatik yapılacak anonsların veri kalitesi yüksek olmalıdır. Özel stüdyolar tarafından kaydedilen bu anonslar çeşitli sıkıştırma teknikleri sonrasında bu birime kaydedilir. Verilerin kalitesinden sonraki diğer bir önemli unsur verilerin üniteye kaydedilme şeklidir. Çeşitli hafıza kartlarını veya USB Flash bellekleri kullanan yapılar maalesef beraberinde sorunlar getirmektedir. Belli zamanlarda yapılan ve o bölgeye ait etkinlikler; konser, panayır gibi anons sistemine dahil edilmesi yada daha önce dahil edildi ise zamanlama programının yeniden düzenlenmesini gerektirir.

Hattın durumuna göre bazı durakların sefere kapatılması durumunda ise ilgili durağın anonslarının iptal edilmesi gerekir. Hatta görev yapan araç sayısı arttığında bu iki durumda da her araca yakından müdahale bu işlemi zora sokan bir etkidir. Bunların hepsini merkezden ve dolayısıyla uzaktan kontrol etme üzerine tasarımlar geliştirilmesi müdahale kolaylığını beraberinde getirecektir.

Yapılması düşünülen sistem her aracın kablosuz iletişim cihazlarıyla donatılması ve araçların depo sahasında veya belirli veri aktarım noktalarında yeniden programlanmasının sağlanmasıdır. Program algoritmasının veri miktarı küçük olduğu için çok kısa iletişim zamanı yeterli olacaktır. Ancak profesyonel anons kayıtlarının veri miktarı yüksek olabileceği için bu tür aktarımlar depo sahasında araçlar seferden çıktıktan sonra ya da geniş sefer aralıklı zamanlarda ana duraklarda yapılması uygun olacaktır.

3.6. Merkezi İşlem Paneli:

Merkezi işlem paneli sistemin yönetiminin yapıldığı kısımdır. Bir tuş takımı ve ekran ya da sadece dokunmatik ekran ile tasarlanabilecek bu kısımda ayarlama yapmak isteyen teknik personelin ayar yapabileceği veya makinistin zamanlanmış anonslar dışında bir anons yapmasının sağlanması için gereklidir. Aynı zamanda bu kısımdan otomatik yapılan ve yapılacak anonsların zamanlaması da görüntülenebilir ve makiniste bilgi verebilmelidir.

Mevcut sistemlerde bazı zamanlama ayarlamalarının makinist tarafından yapılması kaçınılmazdır. Bir önceki başlıkta izah edildiği gibi, geçici durumlarda olan ve önceden otomatik sisteme dahil edilmeyen anons ve zamanlama planları makinist vasıtasıyla el ile programlanması gerekliliğini ortaya koyar, ancak

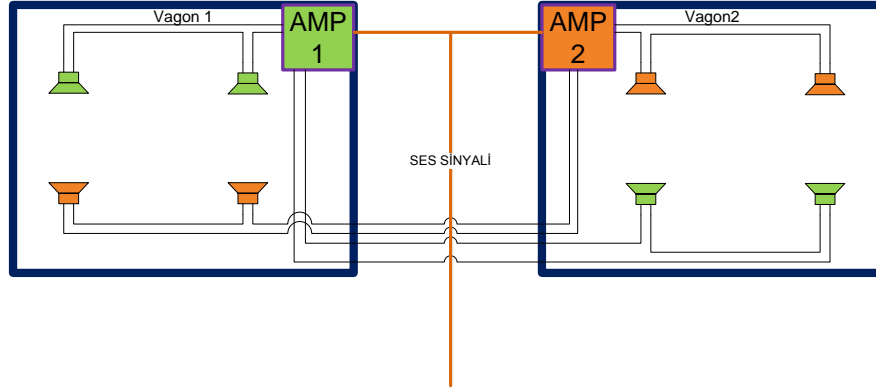
makinistin bu tür ayarlama ile ilgilenmemesi ve dikkatinin dağılmaması istenir. Yine bu durumları en aza indirecek olan merkezden yönetme ve kablosuz haberleşme imkanı sağlanmalıdır.

Merkezi işlem birimine bağlı olarak bir adet mikrofon bulunmaktadır. Bu mikrofon aniden gelişen olaylarda makinistin direkt olarak yolculara bilgi aktarması için kullanılır.

3.7. Adreslenebilir Frekans Yükseltici (AMFI)

Raylı sistem araçlarının iç ve dış kısımlarında yapılması gereken anonsların istenilen şiddete ayarlanmasını sağlayan yükseltici arabirimdir. Adreslenebilir yükselticinin farkı, ses sinyalini yükseltip istenilen bölgeye iletmesidir. Bu bölgeler treninin arka sürüş kabini, trenin içi ve trenin dışıdır.

Ses yükselteçleri her vagona ayrı ayrı bulunur. Yükseltilmemiş ses sinyali vagonlara dağıtılır ve sinyal yükseltilerek hoparlörlere dağıtılır. Belli bir vagona yükseltici kısmın bozulması durumunda ise treninin ilgili vagonuna anons yapılamaması durumunun önüne geçilmesi için çaprazlama bir düzenleme ile her yükseltici bir vagona bulunan hoparlörlerin yarısını ve diğer vagondaki hoparlörlerin yarısını sürer, diğer vagondaki yükselteç de aynı çaprazlama ile iki vagondaki hoparlörlere sinyal gönderir. Bu durumda yükselteçlerden birinin arızalanması durumdan anons yapılan hoparlör sayısı yarıya düşer ancak her vagona anons yapılmış olur. Şekil 3-10'da sistemin bağlantı şekli verilmiştir.



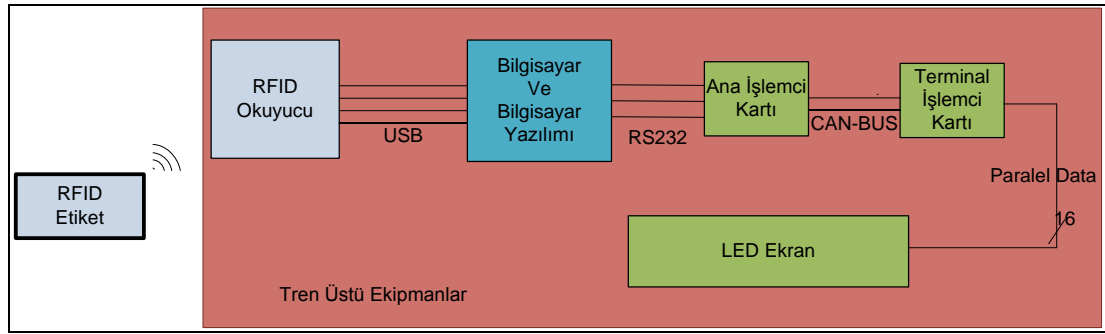
Şekil 3-10 Anons hoparlörlerinin çapraz bağlantı şeması

Ses seviyesi nominal bir değerde olmalıdır. Vatandaşı rahatsız edecek şekilde yüksek seviyede veya duyulamayacak şekilde düşük seviyede olmamalıdır. Ancak bunu sorunsuz şekilde sabit bir ayarla sağlamak mümkün değildir. Seyir esnasında tren içindeki değişken ses seviyesi ve kalabalığın durumuna göre ses şiddeti artırılmalı veya azaltılmalıdır. Bunun için geri beslemeli bir sistem kullanılmalı ve anonsun yapıldığı ortamın ses şiddetine göre anonsun ses şiddetinin sürekli en uygun seviyede tutulması amaçlanır. Ortamdan ses seviyesi ister mikrofon ile sürekli olarak dinlenir ve ses yükselteciye iletilir ya da anons yapılmasından önce hoparlörler vasıtasıyla ses yükselteciye iletilir. İlk durum ek malzeme ve ek kablolama maliyetleri getirdiği için ikinci durumu kullanmak daha uygun olacaktır.

Tüm bu bilgiler ışığında uygulanabilir bir proje için önce bir simülasyon ve örnek model hazırlanması uygun olacaktır.

4. SİSTEM SİMÜLASYONUNUN HAZIRLANMASI

Bu bölümde, bu çalışma için örnek bir model kurulum aşaması bulunmaktadır. Öncelikle bir sistem simülatör yazılımı ve bu yazılıma bilgi aktaracak donanımlar kullanılmış ve ardından, yapılan elektronik devre kartı ve temin edilen tren maketi ile sistemin çalışması görsel olarak sunulmuştur. Şekil 4-1’de örnek proje için kurulan sistemin blok diyagramı gösterilmiştir.



Şekil 4-1 Örnek proje, blok diyagramı

4.1. RFID Etiket ve RFID Okuyucu:

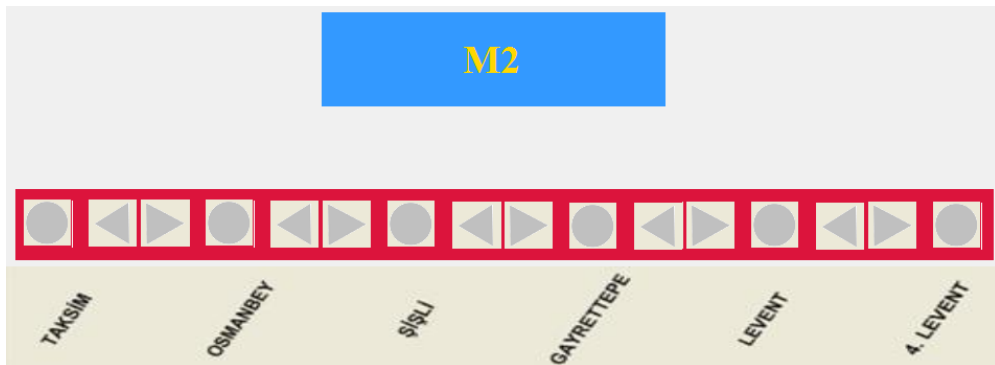
Şekil 4-1’de gösterilen RFID Etiketler her biri ayrı bir ID numarasına sahip kartlardan oluşmaktadır. Bu kartlar örnek projede temsil edilecek istasyonların girişlerinde ve çıkışlarında bulunan etiketlerdir. Bu etiketlerin önceden ID numaraları bilinmektedir ve yerleştirildikleri konumlara göre işlem yapılmaktadır.

RFID Okuyucu ise tren üstü ekipmanlar arasında bulunur. Bu okuyucu kapsama alanı dahiline giren RFID etiketlerin ID bilgisini okur ve bu ID bilgisini USB üzerinden bilgisayara aktarır. RFID okuyucu örnek projede model tren üzerine monte edilerek yazılım kısmında tren hareketlerinin algılanmasını sağlamaktadır.

4.2. Yazılım aşaması:

Şekil 4-1 de verilen blok diyagram üzerinde bilgisayar ve bilgisayar kısmı bu başlık altında incelenmiştir. Bilgisayar yazılımı, .Net geliştirme platformu üzerine C# dili ile sistemin çalışmasının analiz edileceği bir simülasyon ekranı kurulmuştur. Kurulan bu ekranda sistemde bulunan PIB ekranının ve LRM panelinin simülasyonu gerçekleştirilmiştir. RFID okuyucudan USB üzerinden alınan ID verileri daha önceden tanımlanmış ve konum bilgileri belli olan ID verileri ile karşılaştırılır. Bu karşılaştırma ile konum bilgisi elde edilen tren için bazı sorgular yapılır. Bu sorgular “trenin bir önceki konum bilgisi mevcut mu?”, bir önceki konum bilgisi mevcutsa “istasyona yaklaşım yeri sağ taraf mı? yoksa sol taraf mı?” şeklindedir

Simülasyon ekranı için bir arayüz tasarlanmıştır.²³ Bu arayüz mevcut metro hattının bir bölümünün temsili halindedir.



Şekil 4-2 Simülasyon yazılımının ekran görüntüsü

Şekil 4-2’de bulunan yuvarlak semboller istasyonları, istasyonlar arası oklar ise gidilen yönleri gösterir.

. Tablo 4-1’de C# ile yazılan kod bloğunun RFID etiket konumunu bulma ve yer tayini yapma kısımlarına ait akış diyagramı ve kod örnekleri bulunmaktadır.

Tablo 4-1 RFID etiket okuma ve yer tayini

<pre> graph TD Start([Başla]) --> GetRFID[RFID bilgisini al] GetRFID --> IDCheck{ID-Konum bilgisi var mı?} IDCheck -- Hayır --> GetRFID IDCheck -- Evet --> SaveLoc[/Konum bilgisini kaydet/] SaveLoc --> PrevLocCheck{Önceki konum belli mi?} PrevLocCheck -- Hayır --> AlertStart[Anons çal, başlangıcı ekrana yazdır ve RS232'ye konum yaz] PrevLocCheck -- Evet --> DetermineDir[Hareket yönünü tayin et] DetermineDir --> AlertDir[Anons çal, yönü ekrana yazdır ve RS232'ye konum yaz] AlertDir --> End([SON]) AlertStart --> End </pre>	<pre> void serialPort1_DataReceived(object sender, System.IO.Ports.SerialDataReceivedEventArgs e) { int byteAmount = serialPort1.BytesToRead; byte[] tagArray = new byte[10]; for (int i = 0; i < byteAmount; i++) { tagArray[i] = (byte) (serialPort1.ReadByte()); } ... public static string readTag(string dataRead) { string trainName = ""; string tagId = dataRead.Substring(0, 24); string antennaId = dataRead.Substring(24, 1); if (tagListe.ContainsKey(tagId)) { trainName = tagListe[tagId]; } ... public static Dictionary<string, string> tagListe = new Dictionary<string, string>(); public static void loadTagList() { tagListe.Add("00-00-05-01-EC-BF-0B-1F-43-03", @"le41A");//4.levent A baş tagListe.Add("00-00-05-01-2C-68-0E-1F-51-03", @"le41B");//4.levent B baş tagListe.Add("00-00-05-01-BC-7B-0C-... tagListe.Add("00-00-05-01-DC-CC-F7-1E-FD-03", @"tak1A");//Taksim A baş tagListe.Add("00-00-05-01-7C-95-F3-1E-00-03", @"tak1B");//Taksim B baş }... } </pre>
--	---

```

// Yön Sorgulama ve işlem yapma aşaması.
switch (stationAbbr)
{
#region le41
case "le41":

if (stationSide == "A" & levent4StationStatus == false)
{
    levent4StationBusy.SendToBack();
    levent4StationStatus = true;
}
else if (stationSide == "A" & levent4StationStatus == true)
{
    levent4StationOk.SendToBack();
    levent4StationStatus = false;
    label4.Text = "4. LEVENT";
    sendMessageToPIB(stationAbbr);
    PlaySound("C:\\anons\\4LE.wav", new System.IntPtr(),
    PlaySoundFlags.SND_SYNC);
}
else if (stationSide == "B" & levent4StationStatus == false)
{
    le4levLeftOk.SendToBack();
    le4levLeftStatus = false;
    levent4StationBusy.SendToBack();
    levent4StationStatus = true;
sendMessageToPIB();
PlaySound("C:\\anons\\LEV_GEL.wav", new System.IntPtr(),
PlaySoundFlags.SND_SYNC);
}
else if (stationSide == "B" && (le4levRightStatus == false ||
le4levLeftStatus == false))
{
    le4levRightBusy.SendToBack();
    le4levRightStatus = true;
    le4levLeftBusy.SendToBack();
    le4levLeftStatus = true;
    levent4StationOk.SendToBack();
    levent4StationStatus = false;
    label4.Text = "4. LEVENT";
    sendMessageToPIB(stationAbbr);
    PlaySound("C:\\anons\\4LE.wav", new System.IntPtr(),
    PlaySoundFlags.SND_SYNC);
}
else if (stationSide == "B" && (levent4StationStatus == true))
{
    levent4StationOk.SendToBack();
    levent4StationStatus = false;
    label4.Text = "4. LEVENT";
    sendMessageToPIB(stationAbbr);
    PlaySound("C:\\anons\\4LE.wav", new System.IntPtr(),
    PlaySoundFlags.SND_SYNC);
}
break;
#endregion
#region lev1

```

```
case "lev1":  
...  
#region lev1  
case "gay1":  
...  
#region lev1  
case "tak1":
```

Yazılımın bu kısmı trenin hat üzerinde bulunduğu yeri bölgesel olarak tayin etmektedir. Bu işlem ileride sinyalizasyon gibi daha başka projeler içinde kullanılabilir. önemdedir.

Yapılmakta olan bu küçük örnekleme çalışmasında istasyon ve istasyon yaklaşım anonsları bilgisayar üzerinden yapılması uygun görülmektedir. Günümüz bilgisayarlarının birçoğunda hoparlör donanımı mevcut olmakla birlikte işletim sistemlerinde ses formatlarını dönüştürebilecek kod çözümler bulunmaktadır. Ancak nihai projede işletim sistemlerindeki ses formatı kod çözümlerin yerini entegre devrelerin alması gerekmektedir. Buna ek olarak büyük bir metro aracının içinde ve dışında duyulabilecek seviyede ses şiddetini sağlayacak frekans yükselteçleri ve kuvvetli hoparlörler ile sistemin donatılması kaçınılmaz olacaktır.

Bilgisayar tarafında hazırlanan bu yazılım parçasının bir diğer görevi ise ekranda gösterilen bu bilgilerin örnek projenin diğer bir ayağı olan LED panellerin yönetim merkezine ulaştırılmasıdır. Bir sonraki aşamada LED paneller ve görsel ekipmanların yönetim kısmı incelenecektir.

Simulasyon yazılımının tüm kod bloğu Ek 1 kısmında incelenebilir.

4.3. Donanım Aşaması:

Çalışmanın sonucunda bir projenin ortaya çıkması ve kabul edilebilmesi için hazırlanan örnek modeldeki donanım alanları ve bu konudaki yeterlilik ön plana çıkmalıdır. Nasıl ki bir önceki kısımda yapılan yazılım çalışmasının gerekli donanıma sahip bir bilgisayar olmadan hiçbir şey ifade edemeyeceği gibi, bu aşamada da gerekli kabiliyetteki donanımın sağlanması ve bu donanımı oluşturma yeteneği en önemli kısımdır. Bu kısım diğerlerine nazaran daha fazla bilgi birikimi ve üzerinde vakit harcamayı gerektirir. Bunun için öncelikle yazılım platformu ve algoritma geliştirilir.²⁴

Donanım aşaması hazırlanırken; istenilen gereksinimler çıkarılır ve bu gereksinimleri başarabilecek donanım ekipmanları düşünülür. Sistem tekrar ele alınırsa, bilgisayar yazılımı tarafından alınan bilgiler mevcuttur. Bilgisayar yazılımının aldığı bu bilgileri ekranda göstermektedir ve yapılacak anonsu çalmaktadır. Bu kısımda istenilen ise var olan bu bilginin bilgisayar yazılımından donanım birimine aktarılması ve donanım birimindeki entegre devreler ve gömülü yazılım vasıtasıyla LED ekranların istasyon isimlerini göstermesidir.^{25,26,27}

Öncelikle mikrokontrolör seçimi yapılır. Bu çalışmada seçilen mikrokontrolör CAN-BUS iletişimi yapabilecek ve kolay bulunabilecek bir model olan PIC18F2585 olarak seçilmiştir.

Bilgisayar yazılımının ve donanımın karşılıklı haberleşebilmesi için iki biriminde sahip olduğu RS232 seri haberleşme yapısı kullanılmıştır. Bu haberleşme kabiliyeti ile bilgisayar yazılımı, hazır bilgisayar donanımı aracılığıyla donanım kısmında mikrokontrolöre veriler aktarılabilir. Şekil 4-3'de bu bağlantıyı gösteren donanım şeması verilmiştir.



Şekil 4-3 Donanım şeması

Şekil 4-3’de ki şemada görülen modüllerin geniş açıklamaları incelenirse; Bilgisayar (PC) üzerinden ana kontrolör ünitesine RS232 haberleşme yapısı ile gelen istasyon verisi buradaki mikro kontrolör tarafından alınır. Alınan veriler işlendikten sonra CAN-BUS iletişim yapısı ile tüm hatta gönderilir, CAN-BUS iletişim yapısı sayesinde bu hatta bağlı terminallerin hangilerinin gönderilen bu kodu alacağı ve işlem yapacağı bilgisi de koda eklenir. Bu sayede terminal taraflarındaki LED paneller veya LED ekranların mikro kontrolleri ayrı ayrı vazifelerini yapar ve göstermeleri gereken yazı ve şekilleri yansıtırlar.

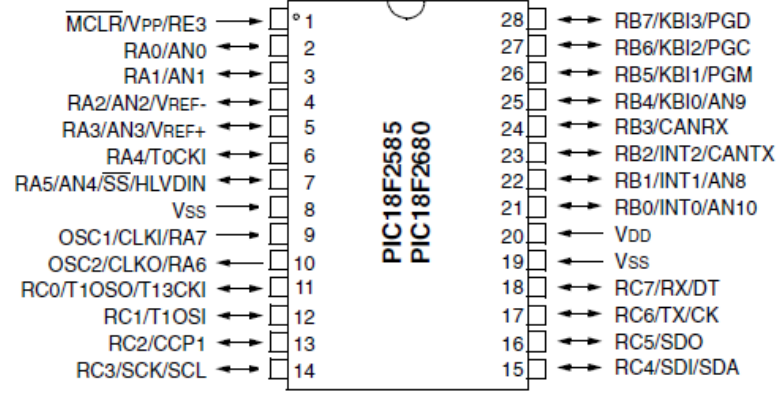
Burada ana kontrolör PC üzerinden gelen veriye göre sisteme bağlı olan tüm terminallere, gereken veriyi hazırlar ve hangisinin ne yapması gerektiğini ayrı ayrı kod blokları halinde tek hat üzerinden gönderir.

4.3.1. Kontrolör Donanımı ve Elemanları:

Merkezi ana kontrolör ve terminaller üzerinde mikro kontrolör olarak PIC18F serisi bir işlemci bulunmaktadır. Bu işlemci çevre elemanlar ile veri alma için gerekli bilgileri hazırlama ve işlenmiş verileri yine çevre elemanlar aracılığı ile sisteme iletme görevlerini yapmak üzere tasarlanmıştır.^{28,29}

Yapılan bu örnek çalışmada ana kontrolör ve terminal kontrolörü için 28 pinli PIC18F2585 mikrokontrolörü seçilmiştir. Şekil 4-4’de PIC18F2585’in pin yapısı verilmiştir.³⁰

28-Pin PDIP, SOIC

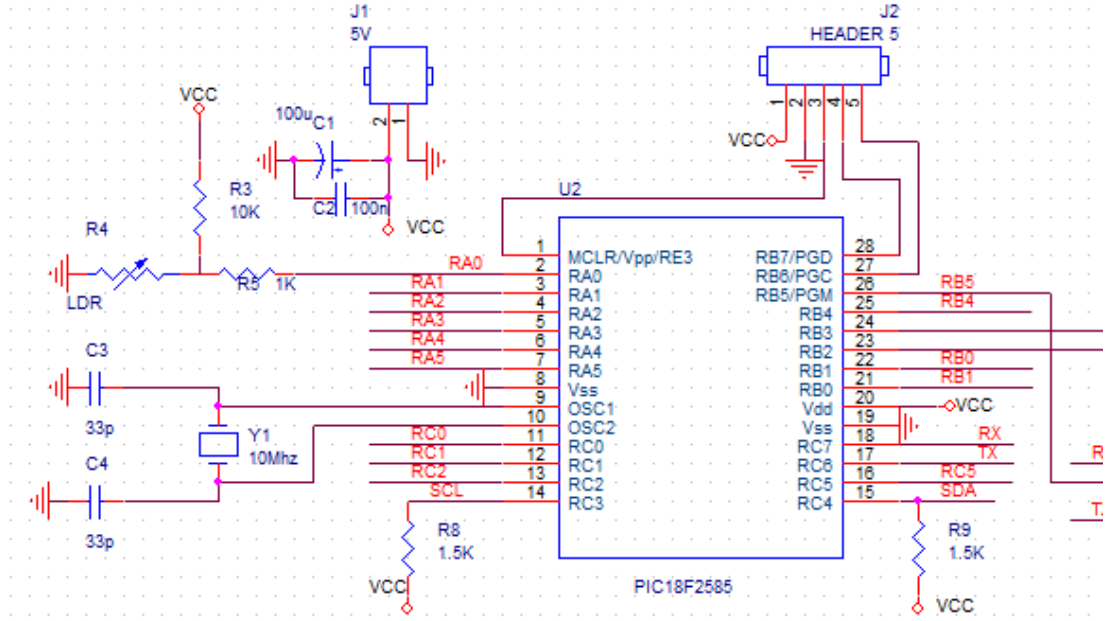


Şekil 4-4 PIC18F2585 pin bağlantı şeması

Tablo 4-2 de PIC18F2585 mikrokontrolörüne ait bazı özellikler verilmiştir.^[30]

Tablo 4-2 PIC18F2585 in Tipik Özellikleri

Çalışma Frekansı	DC – 40 MHz.
Çalışma Gerilimi	4,2 V. – 5,5 V.
Program Hafızası	49152 Byte
Dahili veri Hafızası	3328 Byte
Kesme adedi	19
Giriş/Çıkış Portları	A,B,C,E portları
Zamanlayıcı adedi	4
PWM adedi	1
ADC çevirici	1
Seri Bağlantılar	RS232 UART

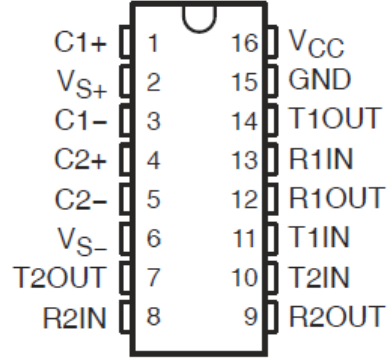


Şekil 4-5 Mikrokontrolör devre şeması

Şekil 4-5’de donanım aşaması için kurulan devrenin mikrokontrolör kısmı verilmiştir.

Mikrokontrolör üzerinde bir port RS232 bağlantı yapısı bulunmaktadır. Ancak bu bağlantı bilgisayarlarda bulunan RS232 bağlantısı ile aynı değildir. Düşük seviyede olan UART bağlantı yardımcı tümdevreler ile istenilen seviyelere çıkartılır. Bir çok entegre firmasının RS232 sürücü entegre imalatı vardır. En çok bilinenler ise Maxim firmasının MAX232 kodlu entegresi ve ST Microelectronics firmasının ST232 kodlu entegresidir. Bu iki entegre tip olarak ve pin bağlantı şekilleri olarak birbirinin muadilidir. Şekil 4-6’de MAX232 entegresinin pin bağlantı şekilleri verilmiştir.³¹

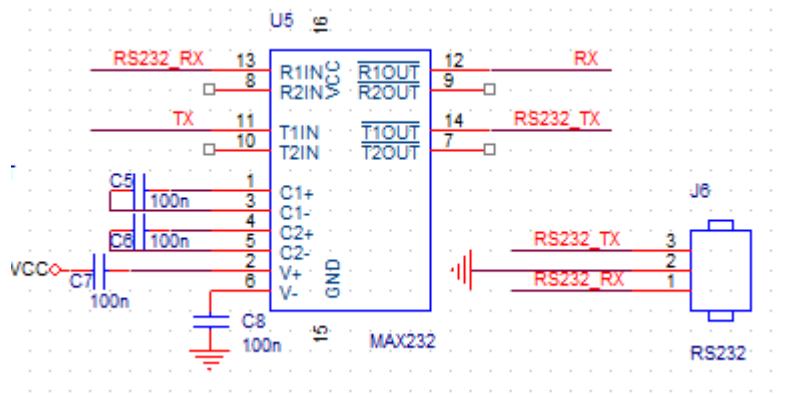
MAX232 . . . D, DW, N, OR NS PACKAGE
 MAX232I . . . D, DW, OR N PACKAGE
 (TOP VIEW)



Şekil 4-6 MAX232 pin bağlantı şeması

Bir entegre kılıfı içerisinde iki kanal RS232 sürücü bulunmaktadır. Bunlar ister tek olarak ister çift olarak ayrı RS232 portları için kullanılabilir. Bu çalışmada tasarlanan devrede sadece bir kanal RS232 sürücü kullanılmıştır. Çünkü PIC18F2585 işlemcisinde bir kanal UART haberleşme bulunmaktadır.

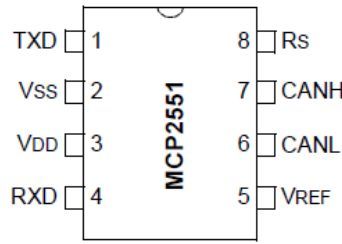
Şekil 4-7’de açık devre şemasının MAX232 entegresi ile yapılmış RS232 bağlantısı verilmiştir.



Şekil 4-7 MAX232 entegresi ile RS232 bağlantı

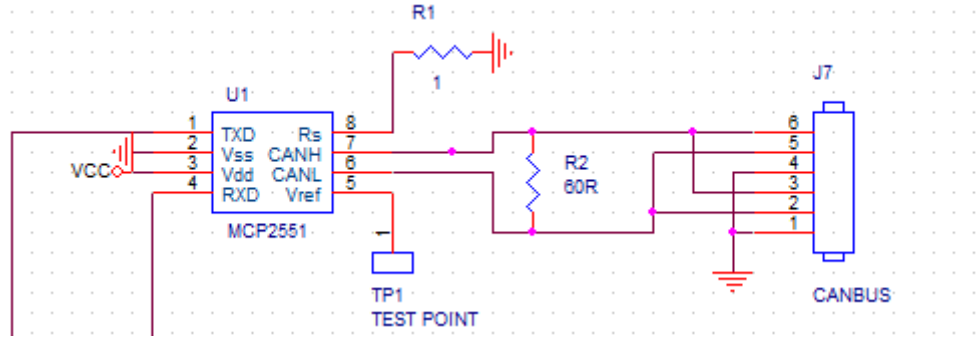
Tasarlanan devre üzerinde istenilen diğer bir haberleşme yeteneği CAN-BUS'tır. Bu haberleşme yeteneğini sağlamak üzere kullanılan yardımcı entegre ise MCP2551 entegresidir. Bu entegre girişine gelen verileri CAN haberleşme yapısına çevirir ve CANHigh ve CANLow olmak üzere iki kablolu hatta iletir. Şekil 4-8 da MCP2551 entegresinin pin bağlantı yapısı verilmiştir.³²

PDIP/SOIC



Şekil 4-8 MCP2551 Pin bağlantı yapısı

Şekil 4-9'da MCP2551 entegresi ile yapılan CAN-BUS bağlantısı için tasarlanmış açık devre şeması verilmiştir.

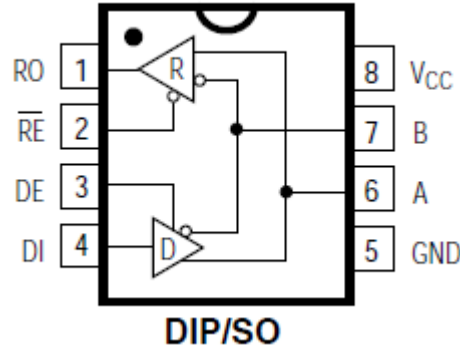


Şekil 4-9 MCP2551 entegresi ile CAN-BUS bağlantı

Tasarlanan devrede ileriye dönük bazı geliştirmelere de yer verilmiştir. Bu tür tasarımlarda devre kartı oluşturma süreci düşünüldüğünde tasarımı çok fazla etkilemeyen ve maliyeti çok fazla değiştirmeyecek eklentiler için hazırlıklar yapılması uygun bir davranış olur. Bu hazırlıkları kullanıp kullanmamak tamamen kullanıcının önceliğindedir. Yapılan hazırlıkların elemanları devreye bağlanmadı

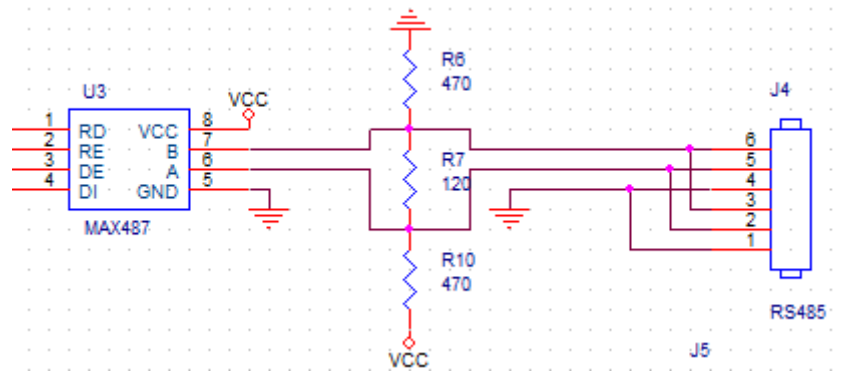
sürece istenilen özellikler aktif olmaz ancak diğer kısımlar vazifesini yapmaya devam eder.

Bu ön hazırlıklar kapsamında yapılan devre tasarımına RS485 haberleşme yeteneği kazandırmak üzere UART bağlantı üzerinden RS485 sürücü entegresine yer verilmiştir. Yine Maxim firmasının MAX487 kod numaralı sürücü entegresi seçilmiştir. Şekil 4-10 da MAX487 entegresinin pin bağlantı şeması bulunmaktadır.³³



Şekil 4-10 MAX487 Pin bağlantı yapısı

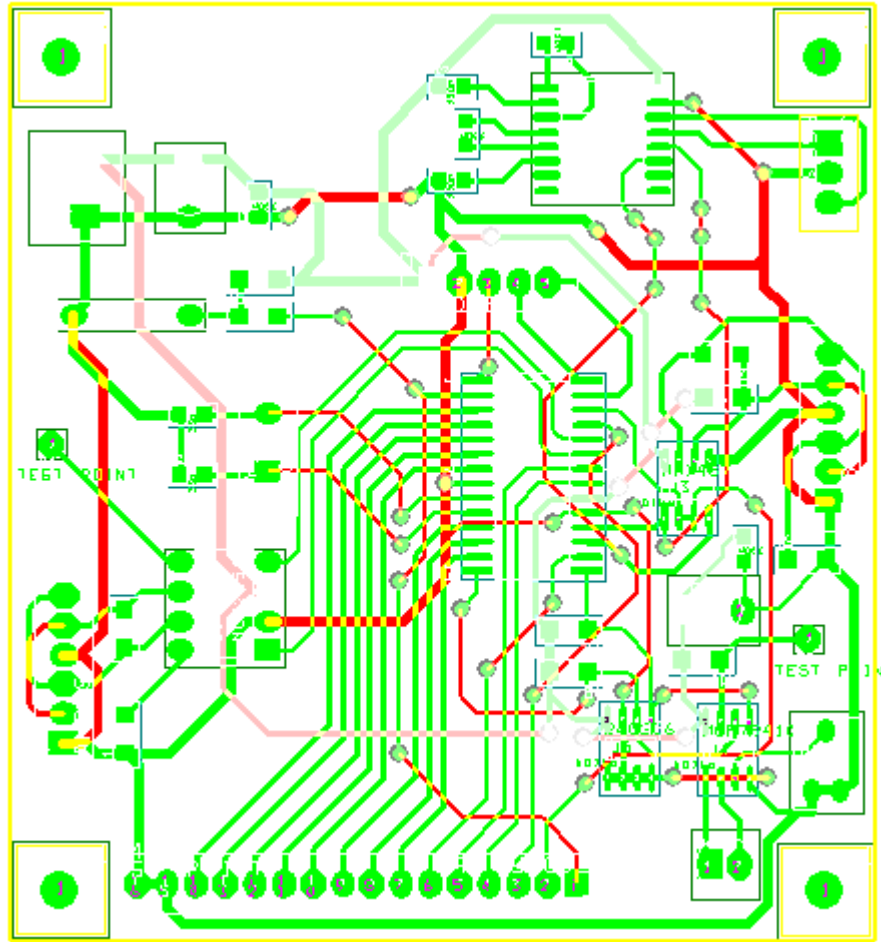
Şekil 4-11’de MAX487 entegresi ile tasarlanmış RS485 bağlantısı açık devre şeması verilmiştir.



Şekil 4-11 MAX487 entegresi ile RS485 bağlantı

Devre tasarımı ORCAD programı kullanılarak yapılmıştır. Parçalar halinde verilmiş olan açık devre şemasının tam hali Ek 2’de verilmiştir.

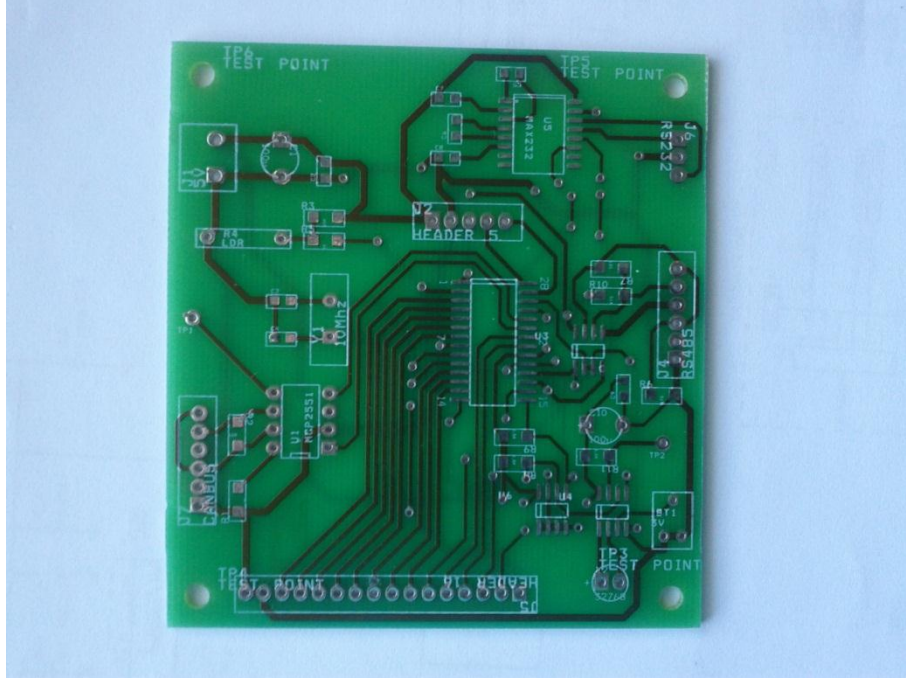
Tasarlanan devrede istenilen tüm özellikler doğrultusunda yapılan şematik tasarım doğrultusunda PCB (Printed Circuit Board) devre şeması hazırlama evresine geçilmiştir.



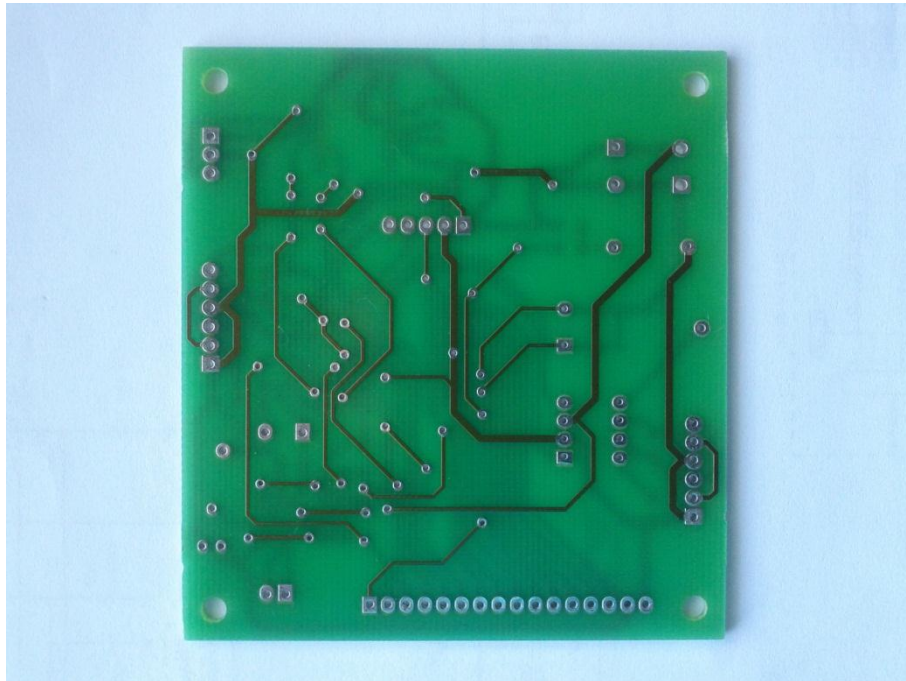
Şekil 4-12 Mikrokontrolör kartı PCB çizimi

PCB tasarımı sonucunda ORCAD programından alınan veriler ile donanım kısmımıza ait baskı devre kartı üretilmiştir. Bu üretim çeşitli yollarlar elde yapılabildiği gibi, bu işi yapan firmalar vasıtasıyla da yapılabilir.

Şekil 4-13 ve Şekil 4-14'de PCB tasarımı yapılan ve sonrasında baskı devre imalatçıları tarafından hazırlanmış PCB kart görülmektedir.



Şekil 4-13 Donanım kısmı PCB kartı üst yüzey

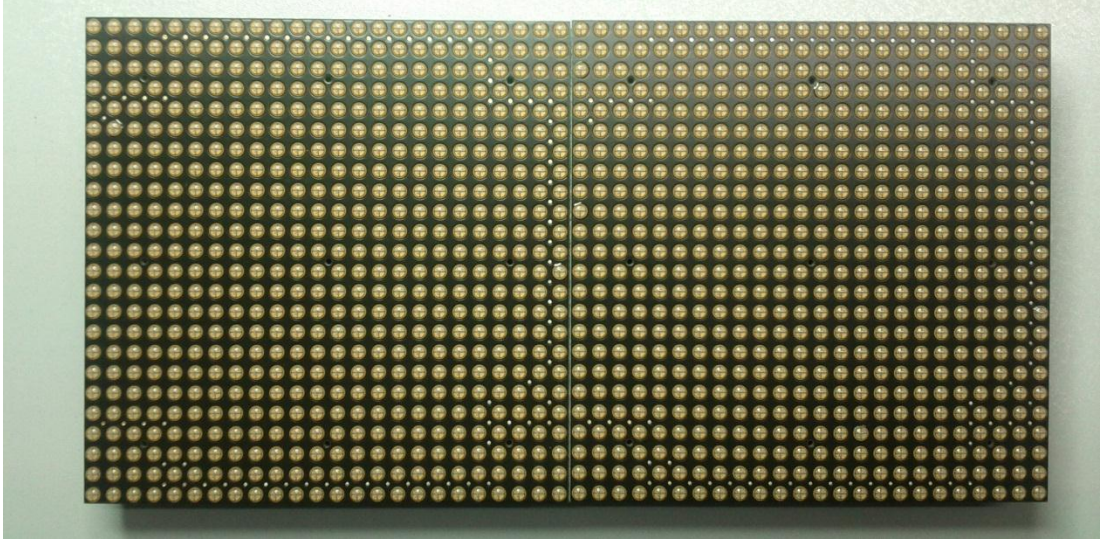


Şekil 4-14 Donanım kısmı PCB kartı alt yüzey

4.3.2. LED Ekranlar:

Bu projede metro araçları içinde çeşitli ölçülerde LED ekranların yapılması gereklidir. DIF, DIS ve PIB ler için 3 ayrı boyutta tasarlanması gereken LED ampullerle hazırlanan mesaj boardları kullanılacaktır. Bu LED boardlar donanım kısmının hazırlandığı gibi şematik tasarım ve ardından PCB kart üretimi şeklinde yapılabilir. Ancak daha muntazam görüntü ve eşit parlaklık elde edilebilmesi ve boardların sarsıntıya karşı dayanımlarının yüksek olması için hazır endüstriyel modüller kullanılabilir. Bu modüller birbirlerine eklenerek istenilen büyüklükte LED ekranlar yapılabilir.

Örnek çalışma için yapılan deneme LED bordu ROHM semiconductor firması tarafından üretilen 24 x 48 adet kırmızı ve yeşil renkli ledler ile üretilen modülleri kullanılmıştır. Bu modüller birbirlerine bağlanarak daha büyük LED ekranlar elde edilebilir.



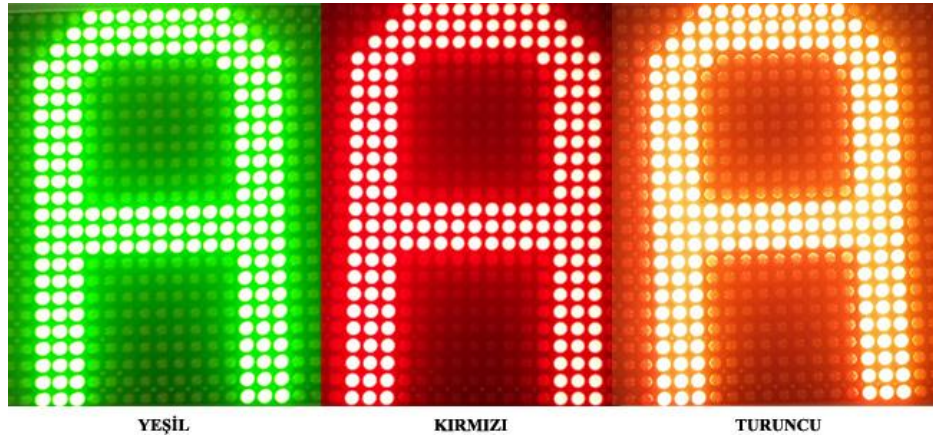
Şekil 4-15 LED Ekran modülü ön yüz



Şekil 4-16 LED Ekran modülü arka yüz

Şekil 4-15’de görülen yukarıdan aşağıya 24 ve sağdan sola 48 adet çift renkli LED ampulün dizilmesi ile üretilmiş olan endüstriyel panel sayesinde, kırmızı ve yeşil ana renkleri ve bu iki rengin aynı anda ışmasıyla üçüncü bir turuncu rengi elde edilmesi mümkündür. Yolcu bilgilendirme sistemi için ön görülen DIF ekranlar için 6 adet 24x48 LED modül tarafından yapılan LED ekranın metro aracının ön kısmında kullanılması, DIS ekranlar için 4 adet 24x48 modül ve PIB ekranlar içinse 4 adet 24x48 modülün kullanılması yer ve görünüm açısından uygun olacaktır.

Şekil 4-17 de LED modülün kullanılabilen renkleri gösterilmiştir.



Şekil 4-17 LED panel renkleri

4.3.3. Gömülü sistem yazılımı

Örnek projede donanım kısmı tasarımı ve üretimi sonrasında bu donanım üzerinde işlem yapacak olan mikrokontrolöre ait yazılım hazırlanmıştır. Bu yazılım gömülü sistem yazılımı olarak kullanılır. Gömülü sistem yazılımını C dili ile yapmak üzere geliştirilmiş platformlar bulunmaktadır. C yazılım dili ile hazırlanmış bu kod bloğunu mikrokontrolörün anlayacağı makine diline çeviren derleyiciler vardır. Bu çalışmada CCS Pic derleyicisi kullanılmıştır.

Bu çalışmada, hazırlanan mikrokontrolör kartı iki farklı amaçla iki ayrı yazılım ile birlikte kullanılmıştır. Bu kartların ilki ana sistem kontrolörü olarak görev yapacaktır. Bilgisayar yazılımından RS232 ile aldığı verileri işleyerek sistemdeki diğer mikrokontrolör kartlarına CAN-BUS ile bilgi aktaracaktır. Ana işlemci kartından bilgi alan işlemci kartları ise LED ekranlara verilerin yazılmasını gerçekleştirir.

Tablo 4-3’de Gömülü yazılımda RS232 üzerinden veri okuma kod bloğu bulunmaktadır. Bu yazılımda işlemcinin kesme bitlerinden biri ile RS232 üzerinden gelen bir veri algılanır ve bu bitler bir diziye kaydedilir. Veriler sekiz bitlik parçalar halinde alınır ve saklanır.

Tablo 4-3 Gömülü yazılım, RS232 ile veri okuma

```
void SeriPort1DataGeldiInterrupt () {
    KarakterRs232 = fgetc(PORT1);
    if( (KarakterRs232==0X02) ){
        SayacPort1Gelen=0;
    }
    else if(KarakterRs232==0X03){
        DiziPort1Gelen[SayacPort1Gelen] = KarakterRs232;

        if(DiziPort1Gelen[1]=='0' && DiziPort1Gelen[2]=='0'){
            if(DiziPort1Gelen[3]=='P' && DiziPort1Gelen[4]=='Y'){
                if(DiziPort1Gelen[5]=='1'){

                    for(Sayac=0; Sayac<20; Sayac++){
```

```

        DiziCanBusGonderilenMesaj[Sayac]=0;
    }

    Sayac=0;

    do{

DiziCanBusGonderilenMesaj[Sayac]=DiziPort1Gelen[Sayac+6];

        }while(DiziPort1Gelen[6 + Sayac++]!=3);

        DiziCanBusGonderilenMesaj[Sayac-1]=3;
        DiziCanBusGonderilenMesaj[Sayac ]=0;

        DurumCanDataYaz=10;
    }
}
}
}
DiziPort1Gelen[SayacPort1Gelen++] = KarakterRs232;
}

```

Bilgisayar yazılımı üzerinden RS232 ile ana işlemci kartına gelen veri tipi önceden belirlenmiştir. Bu veri tipi Tablo 4-4’da gösterilmiştir. Bu mesaj kısmında 02; hat üzerinden gelen verinin başladığını, 00; ID bilgisini, PY1; yapılacak işlemin “Panoya Yazmak” olduğunu belirtmektedir. Bu ön eklerden sonra ekrana aktarılacak mesaj verisi karakter bazında –TAKSİM- şeklinde gönderilmekte ve 03 kodu ile veri aktarımının sonunun geldiğini belirtmektedir.

Tablo 4-4 RS232 ile aktarılan veri tipi

Start	ID	Komut	Mesaj	Son
02	00	PY1	TAKSİM	03

Ana işlemci kartı işlediği bu veriyi sistemdeki diğer işlemci kartlarına CAN-BUS üzerinden aktaracaktır. Bunun için CCS (Custom Computer Services) derleyicimize kullandığımız kontrolöre uygun CAN-BUS iletişimi için önceden tanımlanmış kod bloğu dosyası kullanılmıştır. Bu kod bloğu kurulum evresinde “#include <can-18F4580.c>” satırında verilmiştir. Bu kod bloğu sayesinde CAN-BUS ile veri aktarımı ve doğrulama işlemleri hazır olarak yapılır. Hazırlanmış

olduğumuz yazılım ise RS232 verisinde olduğu gibi belli bir düzen dahilinde gelen verileri kaydedip işlem yapmamızı sağlar.

Tablo 4-5 Gömülü yazılım, CAN-BUS ile veri okuma

```
#int_CANRX0
void CanDataGeldi(void){
    int Sayac=0;
    int SayacBos=0;
    char Karakter;
    can_getd(rx_id, &in_data[0], rx_len, rxstat);
    do{
        Karakter = in_data[Sayac];
        DiziPanelSurulen[SayacCanBusGelen++]=Karakter;
        if(Karakter==3){

            for(SayacBos=SayacCanBusGelen-1; SayacBos<11; SayacBos++){
                DiziPanelSurulen[SayacBos]=' ';
            }
            DurumCanDataGeldi=10;
            SayacCanBusGelen=0;
            Sayac=8;
            DiziPanelSurulen[10]=0;
            EkranRamSil();
            EkranSur();
        }
        Sayac++;
    }while(Sayac<=7);
}
```

Tablo 4-5’de verilen kod bloğu ile RS232 ye benzer bir yapı kullanılmıştır. CAN-BUS entegresi üzerinden gelen her bitlik veri ile CAN-BUS kesmesi aktif olur ve CAN-BUS veri alma rutini çalışır. Burada da sekiz bitlik verinin tamamlanması ile verileri bayt olarak saklama ve daha sonrasında ekrana basma işlemi yapılır.

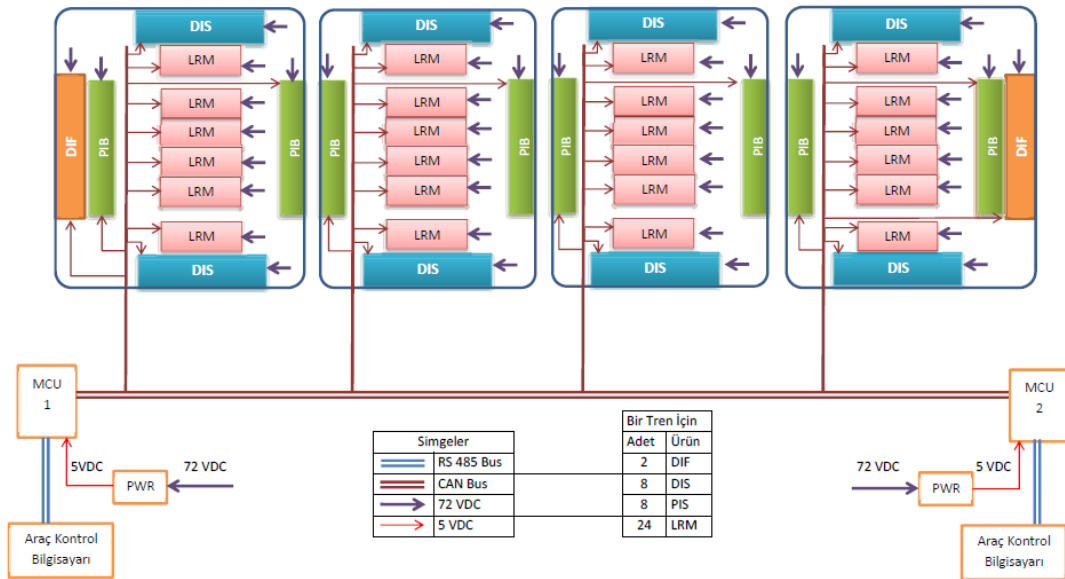
Donanım aşamasına ait olan Ana işlemci kartı ve terminal işlemci kartı gömülü yazılım için hazırlanmış tüm kod bloğu Ek 3 kısmında verilmiştir. Ayrıca Ek 4 kısmında gömülü sistem yazılımında kullanılan rutinlere ait akış diyagramları verilmiştir. Bu akış diyagramları sırasıyla, ana program rutini, RS232 okuma fonksiyonu, CAN-BUS yazma fonksiyonu, CAN-BUS okuma fonksiyonu ve LED panele veri yazma fonksiyonudur.

4.4. Fizibilite Çalışması:

Yapılmakta olan bu çalışma sonucunda İstanbul şehrine hizmet etmekte olan Taksim metrosuna ait eski metro araçlarının Yolcu Bilgilendirme Sistemi ile donatılması öngörülmektedir. Bu sayede eski ve yeni araçların aynı anda kullanılması ile daha sık seferler yapılabilecektir. Yapılacak olan sunum sonucunda yapılabilirlik ve bütçe durumları göz önüne alınacaktır ve projenin uygulanıp uygulanmayacağı kararı verilecektir.

Bu fizibilite çalışmasına konu olan Taksim metrosu eski metro araçları için yaklaşık toplam sistem maliyeti çıkarılacaktır. Bahsi geçen metro aracı sayısı 32 adettir. Bunlar dört araç bir dizi oluşturacak şekilde 8 dizi oluşturmaktadır. Yani proje sonucunda sefere hazır 8 metro dizisi daha hizmete alınacaktır. Bu 8 dizi için gerekli donanımlar ele alınırsa;

- İlk önce bir dizi için gerekli ekran LED ekran adedi hesaplanmalıdır.



Şekil 4-18 Bir tren için kullanılan görsel ekipmanlar

Şekil 4-18’da görüldüğü üzere her bir tren içinde 24x48 LED modüllerden meydana gelen iki adet güzergah tabelası ön (DIF) için toplam 10 adet modül, sekiz adet güzergah tabelası yan (DIS) için toplam 32 adet modül ve sekiz adet yolcu bilgilendirme paneli (PIB) için toplam 32 adet modül kullanılması gerekir. Bu veriler Tablo 4-6 de ki gibi toplanır ve sekiz dizi tren için hesaplanırsa ihtiyaç olan LED panel modül sayısı elde edilir.

Tablo 4-6 Proje için gerekli LED panel modül adedi

Ürün	Tren Sayısı	Adet	Modül	Tren Toplamı	Modül Toplamı
DIF	8	2	5	10	80
DIS	8	8	4	32	256
PIB	8	8	4	32	256
Toplam Modül Adedi					592

Numune ürün için alınan teklifler ve adetli alım için elde edilen teklifler sonucunda her bir modül için yaklaşık 170 USD ödenmesi gerekmiştir. Çıkarılan bu hesap sonucunda 592 modül için 100.640 USD ayrılması gerekmektedir.

- Her bir ürün için gerekli gerilim dönüştürücü bedeli;

Sistem üzerine kurulacak olan DIF, DIS ve PIB gibi tüm ürünler için metro aracında bulunan en uygun kaynaktan alınan beslemenin ürünlerde kullanılacak 5 V. DC seviyesine dönüştürecek gerilim dönüştürücülerine ihtiyaç vardır. Metro aracı içinden 72 V. DC hattından alınan gerilimi 5 V. DC gerilimine dönüştürecek DC-DC dönüştürücü için yaklaşık maliyet; 80 USD’dir. Tablo 4-7’de toplam ürün sayısı ve buna mukabil gereken DC-DC dönüştürücü maliyeti gösterilmiştir.

Tablo 4-7 Proje için gerekli DC-DC dönüştürücü adedi

Ürün	Ürün Adet	Toplam ürün adedi
DIF	2	16
DIS	8	64
PIB	8	64
Gerekli DC-DC dönüştürücü adedi		144

Tablo 4-7’de görüldüğü gibi 144 adet DC-DC dönüştürücü için birim maliyet 80 USD ye göre toplamda; 11.520 USD ayrılacak bütçeye eklenmelidir.

- Diğer ekipmanlar ve malzemeler;

Ana malzemelerin bütçe ihtiyacı çıkarıldıktan sonra son olarak sistem için gerekli tüm ekipmanların ve küçük malzemelerin yaklaşık bedelleri için bir çalışma yapılmıştır. Aşağıdaki tabloda; Ürünler için gerekli alüminyum kasa maliyetleri, yine ürünler için pleksiglass kaplama, PCB kartlar, montaj aparatları, LRM için besleme dönüştürücüleri, araç kontrol bilgisayarı (OBCU) dönüştürücü kartları ve konnektör-kablo malzemeleri için ön görülen değerler eklenmiştir.

Tablo 4-8 Diğer ekipman ve malzemeler maliyetleri

Ürün	Adet	Modül	Toplam	Toplam	Adet Fiyatı	Toplam Fiyat
Alüminyum Kasa	42	1	42	42	\$30,00	\$1.260,00
Pleksiglass Tampon Baskı	42	1	42	42	\$20,00	\$840,00
İşlemci PCB Kart	42	1	42	42	\$40,00	\$1.680,00
Montaj Aparatları	42	1	42	42	\$10,00	\$420,00
LRM için Besleme Kartı	16	1	16	16	\$30,00	\$480,00
OBCU Dönüştürücü Kartı	2	1	2	2	\$50,00	\$100,00
Konnektör + Kablo	42	1	42	42	\$20,00	\$840,00
Toplam ekipman ve malzemeler maliyeti						\$5.620,00

Tablo 4-8’de toplam maliyet için 5.620 USD bütçe tablosuna eklenmelidir. Eklenen bu bütçe kalemi ile ön görülen nihai proje maliyeti tablosu çıkarılmış olur. Tablo 4-9 te tüm kalemlerin toplamı ve toplam proje maliyeti verilmiştir.

Tablo 4-9 Toplam proje maliyeti

Ürün	Tren Adet	Adet	Modül	Toplam	Adet Fiyatı	Toplam Fiyat
DIF	8	2	5	80	\$170,00	\$13.600,00
DIS	8	8	4	256	\$170,00	\$43.520,00
PIB	8	8	4	256	\$170,00	\$43.520,00
LRM	8	24	1	192	\$150,00	\$28.800,00
SMPS 50-90VDC / 5V 20A	8	18	1	144	\$80,00	\$11.520,00
Aluminyum Kasa	8	42	1	336	\$30,00	\$10.080,00
Pleksiglass Tampon Baskı	8	42	1	336	\$20,00	\$6.720,00
İşlemci Anakart	8	42	1	336	\$40,00	\$13.440,00
Montaj Aparatları	8	42	1	336	\$10,00	\$3.360,00
LRM için Besleme	8	16	1	128	\$30,00	\$3.840,00
OBCU Dönüştürücü Kartı	8	2	1	16	\$50,00	\$800,00
Konnecter + Kablo	8	42	1	336	\$20,00	\$6.720,00
TOPLAM PROJE MALİYETİ						\$ 185.920,00

Tablo 4-9’de görüldüğü gibi toplam proje maliyeti yaklaşık olarak \$185.000 çıkmıştır. Bu maliyet sadece malzemelerin bedelidir. Karşılaştırma yapabilmek için, farklı bir firmanın yapılacak olan bu hizmete karşı fiyatlarını incelemek gereklidir.

Tablo 4-10’da bahsi geçen metro hattı üzerinde vazife yapan yeni metro araçları içlerindeki yolcu bilgilendirme sistemi ekipmanlarına ait ürün fiyatları bulunmaktadır. Bu fiyatlarda sadece ürün fiyatları olup montaj ve işgücü bedelleri eklenmemiştir.

Tablo 4-10 Proje için alternatif maliyet tablosu

Ürün	Ürün Adet	Alternatif Birim fiyat	Toplam Fiyat
DIF	16	\$ 2.157,00	\$ 34.512,00
DIS	64	\$ 1.975,00	\$ 126.400,00
PIB	64	\$ 1.759,00	\$ 112.576,00
LRM	192	\$ 1.513,00	\$ 290.496,00
Konnector + Kablo	8	\$ 4.520,00	\$ 36.160,00
Alternatif Toplam Maliyet			\$ 600.144,00

Tablo 4-10’da görüldüğü üzere ürünlerin üretim ile temin edilmesiyle yurtdışı firmalardan ürün olarak alınması arasında yaklaşık olarak üç kattan daha fazla fiyat farkı vardır. Bu fark kapatılmaya çalışılsa dahi üretim ile aynı maliyetlere yapılması mümkün gözükmemektedir. Diğer bir taraftan ise aynı maliyetlere yurtdışından temin edilebilse dahi ürünler hakkında nasıl yapıldığını bilme (know how) beceresi elde edilemeyeceği için montaj, kurulum ve bakım süreçlerinde işçilik bedelleri her daim yüksek çıkacaktır. Yurtdışından temin edilen ürünler ile birlikte teknisyen ve mühendis getirtme maliyetleri firmalar için her zaman yüksek bedelli kalemler arasında yer almaktadır.

5. SONUÇ

Bu çalışmada, yer altında hizmet yapan raylı sistem araçları gibi GPS sisteminin kullanılmadığı yerlerde yer tayin ve yönlendirme sistemi konusunda RFID etiketlerin etkin kullanımı incelenmiş ve yapılan uygulama çalışması ile RFID etiketlerle yer tayini yapılmıştır. Aynı zamanda RFID sistemi ile yer tayini sonrasında araçların içinde, yolcuların yönlendirmesini amaçlayan görsel ve işitsel elemanların yönetilmesi sağlanmıştır. Hazırlanan bu iki aşamalı proje çalışması ileriye dönük birçok projeye ön hazırlık niteliğindedir. Yer tayini olarak RFID sisteminin kullanılması, raylı sistem araçlarında çok yüksek güvenilirlikli olarak gerçekleştirilmesi gereken sinyalizasyon sistemi için konumlandırma kısmı için ön hazırlık olmuştur. Yolcu bilgilendirme sistemi için yapılan çalışmalar ise LED ekran ve ışıklı yönlendirmeler için hazırlanabilecek tüm projelere hazırlık mahiyetindedir.

Sadece iki hat ile uzun mesafelere, yüksek güvenlikle veri aktarımı için CAN-BUS haberleşme tekniğinin kullanılması ile aynı hat üzerinde birçok terminalin bağlanması gerçekleştirilmiş ve bu terminallere ayrı ayrı işler gönderilmiştir. Bu sayede sistemlerde birçok kayıp ve arıza meydana getiren kablo karmaşasını yok etmiştir.

Son olarak İstanbul Büyükşehir Belediyesinin sahip olduğu, önceki yıllarda seferde olan ancak sinyalizasyon sisteminin değişmesi sonrasında sinyalizasyon sistemlerinin alt yapısının dönüştürülmesi sürecine giren trenlerin bu dönüşüm sonrasında yolcu bilgilendirme sistemlerinin yenilenmesi gerekliliği ortaya çıkmıştır. Bu araçlar için yurtdışı çözümlere gerek kalmadan üretim ve bakım beceresinin elde edilmesi sağlanmıştır.

KAYNAKLAR

- 1 Theeg, G., Vlasenko, S., “Railway Signalling & Interlocking”, Eurail Press, Almanya, 2009
- 2 IEE Railway Professional Network, Tenth Residential Course on Railway Signalling and Control Systems, University of Nottingham, 2004
- 3 Drew, J., “Reforming Railways”, Eurail Press, Hamburg, 2011
- 4 Yüksel, H.E., “Raylı Toplu Tasıma Sinyalizasyon Sistemleri ve Marmaray Projesinin Sinyalizasyonu”, Niğde, 2007
- 5 <http://www.railway-technology.com/features/feature1684/> (05.10.2011)
- 6 <http://www.research.nokia.com/news/9505>, (10.12.2010)
- 7 Gümüşkaya, H., “Demiryolu Sinyal Sistemi Geliştirmede İlk Konular, Teknolojiler ve Bilimsel Altyapı”, İstanbul, 2009
- 8 Açıkbaş, S., Söyler, H., “Raylı Toplu Taşımda Sinyalizasyon Sistemleri”, İstanbul, 2008
- 9 Dr.Köhne, A., Dr. Wössner, M., “Achievable Accuracy of GPS”, 2009
- 10 Öztürk Z., Arlı, V., “Demiryolunda Gürültü ve Titreşim Sorunu” Marmara Yayıncılık, İstanbul, 2010
- 11 Inter M, “Hyundai Rothem Fiyat Teklifi”, İstanbul, 2010
- 12 <http://tr.wikipedia.org/wiki/GPS> (20.06.2011)
- 13 <http://discusstech.org/2011/04/working-of-gps-navigator-device/> (20.06.2011)
- 14 Pala, Z., “RFID Teknolojisi İle Otomasyon Bir Uygulama Olarak: Otopark Takibi”, 2007
- 15 Dehollain, C., Curty, J.P., Declercq, M., Joehl, N., “Design and Optimization of Passive UHF RFID Systems”, ISBN:0387447100, Springer US, Newyork, 2007

-
- 16 <http://www.flickr.com/people/timo/> (28.06.2011)
- 17 Bayat, Ç., “RFID Uygulamalarının İşletme Verimliliği Üzerindeki Etkileri”, İstanbul, 2010
- 18 Taşkın, C., “Ağ Teknolojileri ve Telekomünikasyon”, ISBN: 9944711258, İstanbul, 2009
- 19 <http://www.lammertbies.nl/comm/info/RS-485.html> (17.07.2011)
- 20 Building Simulated Aircraft Instrumentation Support
http://www.mikesflightdeck.com/mfdb/bsai/bsai_support.html (18.09.2011)
- 21 Bosch, R., “CAN Specification Version2.0”, Stutgard-Almanya, 1991
- 22 Wikipedia, http://en.wikipedia.org/wiki/Controller_area_network (18.09.2011)
- 23 Nesh, T., “Accelerated C# 2008”, ISBN: 1430203384, Apress, 2008
- 24 Çiçek, S., “CCS C İle PIC Programlama”, ISBN: 9758834204, Altaş Yayıncılık, 2009
- 25 Altınbaşak, O., “Mikrodenetleyiciler ve PIC Programlama”, İstanbul, 2004
- 26 Bodur, Y., “Adım Adım PIC Micro Programlama – Kurs Kitabı”, İstanbul, 2002
- 27 <http://www.microchip.com/developmenttools> (05.07.2011)
- 28 Ekiz, H., “Mantık Devreleri Sayısal Elektronik” Sakarya, 2005
- 29 Özcerit, A.T., Çakıroğlu, M., Bayılmış, M., “8051 Mikrodenetleyici Uygulamaları”, İstanbul, 2008
- 30 <http://ww1.microchip.com/downloads/en/DeviceDoc/39625c.pdf> Microchip Datasheets (04.08.2011)
- 31 <http://datasheets.maxim-ic.com/en/ds/MAX220-MAX249.pdf> Maxim Datasheets (04.08.2011)

32 <http://ww1.microchip.com/downloads/en/devicedoc/21667d.pdf> Microchip
Datasheets (04.08.2011)

33 <http://datasheets.maxim-ic.com/en/ds/MAX1487-MAX491.pdf> Maxim
Datasheets (04.08.2011)

ÖZGEÇMİŞ



MEHMED AKİF ÖZKAYA

1985 yılında İstanbul'da doğdu. 2004 yılında İstanbul Ticaret Odası Anadolu Teknik Lisesi Elektronik Bölümünden mezun olmuş, 2009 yılında Sakarya Üniversitesi Elektronik Öğretmenliği Bölümünü başarıyla tamamlamıştır.

Lisans yıllarında dâhil olduğu SAİTEM ekibi ile güneş enerjili ve hidrojen enerjili araçlar yapıp yurt içi ve yurtdışında yarışmalara katılmıştır. Halen bu ekipte aktif danışmanlık görevini sürdürmekte olup 2011 Avustralya World Solar Challenge yarışması hazırlıklarına devam etmektedir.

Lisans mezuniyeti ardından aynı yıl Maltepe Üniversitesi Elektronik Mühendisliği Yüksek Lisans eğitimine başladı. Halen yüksek lisans Eğitimine devam eden M.Akif lisans eğitimi sonrasında İstanbul Büyükşehir Belediyesi İstanbul Ulaşım A.Ş. firmasında Teknokent Ar-Ge Koordinatörlüğü biriminde vazife yapmaktadır. Birimin projelerinden bazıları; raylı sistem sinyalizasyonu, metro aracı yardımcı güç ünitesi, araç kontrol bilgisayarı ve jeton satış makinesidir.

Ayrıca M.Akif ilköğretim yıllarından itibaren sosyal aktivite olarak, radyo spikerliği, televizyon programı sunuculuğu, dizi ve tiyatro oyunculuğu yapmıştır. Halen bu aktivitelerine de vakit ayırmaya çalışmaktadır.

6. EKLER

EK 1 - Bilgisayar Yazılımı Kod Bloğu:

```
• Form1.Designer.cs;
namespace WindowsFormsApplication1
{
    partial class Form1
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources should
be disposed; otherwise, false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }
        ...//otomatik form objeleri
private System.Windows.Forms.Button button2;
private System.Windows.Forms.ComboBox comboBox1;
private System.Windows.Forms.PictureBox pictureBox1;
private System.Windows.Forms.Panel panel19;
private System.Windows.Forms.PictureBox pictureBox2;
private System.Windows.Forms.PictureBox pictureBox4;
private System.Windows.Forms.PictureBox pictureBox3;
private System.Windows.Forms.PictureBox pictureBox5;
private System.Windows.Forms.PictureBox pictureBox6;
private System.Windows.Forms.PictureBox pictureBox7;
private System.Windows.Forms.PictureBox pictureBox8;
```

```

private System.Windows.Forms.PictureBox pictureBox9;
private System.Windows.Forms.Panel panel20;
private System.Windows.Forms.PictureBox pictureBox10;
private System.Windows.Forms.PictureBox levent4StationOk;
private System.Windows.Forms.PictureBox pictureBox12;
private System.Windows.Forms.PictureBox le4levLeftOk;
private System.Windows.Forms.PictureBox le4levRightOk;
private System.Windows.Forms.PictureBox le4levLeftBusy;
private System.Windows.Forms.PictureBox le4levRightBusy;
private System.Windows.Forms.PictureBox pictureBox17;
private System.Windows.Forms.PictureBox pictureBox18;
private System.Windows.Forms.PictureBox levent4StationBusy;
private System.Windows.Forms.PictureBox levGayRightOk;
private System.Windows.Forms.PictureBox levGayRightBusy;
private System.Windows.Forms.PictureBox leventStationBusy;
private System.Windows.Forms.PictureBox leventStationOk;
private System.Windows.Forms.PictureBox levGayLeftOk;
private System.Windows.Forms.PictureBox levGayLeftBusy;
private System.Windows.Forms.PictureBox taksimStationBusy;
private System.Windows.Forms.PictureBox taksimStationOk;
private System.Windows.Forms.PictureBox osmTakLeftOk;
private System.Windows.Forms.PictureBox osmTakRightOk;
private System.Windows.Forms.PictureBox osmTakLeftBusy;
private System.Windows.Forms.PictureBox osmTakRightBusy;
private System.Windows.Forms.PictureBox osmanbeyStationBusy;
private System.Windows.Forms.PictureBox osmanbeyStationOk;
private System.Windows.Forms.PictureBox sisOsmLeftOk;
private System.Windows.Forms.PictureBox sisOsmRightOk;
private System.Windows.Forms.PictureBox sisOsmLeftBusy;
private System.Windows.Forms.PictureBox sisOsmRightBusy;
private System.Windows.Forms.PictureBox sisliStationBusy;
private System.Windows.Forms.PictureBox sisliStationOk;
private System.Windows.Forms.PictureBox gaySisLeftOk;
private System.Windows.Forms.PictureBox gaySisRightOk;
private System.Windows.Forms.PictureBox gaySisLeftBusy;
private System.Windows.Forms.PictureBox gaySisRightBusy;
private System.Windows.Forms.PictureBox
gayrettepeStationBusy;
private System.Windows.Forms.PictureBox gayrettepeStationOk;
private System.Windows.Forms.Label label29;
private System.Windows.Forms.PictureBox pictureBox64;
private System.Windows.Forms.Label labell;
private System.Windows.Forms.Timer timer1;
private System.Windows.Forms.PictureBox pictureBox11;
private System.Windows.Forms.Label label2;
private System.Windows.Forms.Label label3;
private System.Windows.Forms.Label label4;
private System.IO.Ports.SerialPort serialPort1;

```

```

    }
}

• Form1.cs;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Threading;
using System.Runtime.InteropServices;

namespace WindowsFormsApplication1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
            CheckForIllegalCrossThreadCalls = false;
            timer1.Start();
            timer1.Tick += new EventHandler(timer1_Tick);
        }
        #region playsound
        [System.Runtime.InteropServices.DllImport("winmm.DLL",
EntryPoint = "PlaySound", SetLastError = true, CharSet =
CharSet.Unicode, ThrowOnUnmappableChar = true)]
        public static extern bool PlaySound(string szSound,
System.IntPtr hmod, PlaySoundFlags flags);
        [System.Flags]
        public enum PlaySoundFlags : int
        {
            SND_SYNC = 0x0000,
            SND_ASYNC = 0x0001,
            SND_NODEFAULT = 0x0002,
            SND_LOOP = 0x0008,
            SND_NOSTOP = 0x0010,
            SND_NOWAIT = 0x00002000,
            SND_FILENAME = 0x00020000,
            SND_RESOURCE = 0x00040004
        }
        #endregion

        void timer1_Tick(object sender, EventArgs e)

```

```

    {
        label1.Text = DateTime.Now.ToLongTimeString();
    }

private void Form1_Load(object sender, EventArgs e)
{
    serialPort1.Open();
    Reader.loadTagList();
    serialPort1.DataReceived += new
System.IO.Ports.SerialDataReceivedEventHandler(serialPort1_DataRecei
ved);
}

static bool busy;
static bool OK;

#region Status

bool levent4StationStatus = true;
bool levent42StationStatus = true;
bool le4levLeftStatus = true;
bool le4levRightStatus = true;
bool leventStationStatus = true;
bool levent2StationStatus = true;
bool levGayLeftStatus = true;
bool levGayRightStatus = true;
bool gayrettepeStationStatus = true;
bool gayrettepe2StationStatus = true;
bool gaySisLeftStatus = true;
bool gaySisRightStatus = true;
bool sisliStationStatus = true;
bool sisli2StationStatus = true;
bool sisOsmLeftStatus = true;
bool sisOsmRightStatus = true;
bool osmanbeyStationStatus = true;
bool osmanbey2StationStatus = true;
bool osmTakLeftStatus = true;
bool osmTakRightStatus = true;
bool taksimStationStatus = true;
bool taksim2StationStatus = true;
bool karOtoLeftStatus = true;
bool karOtoRightStatus = true;
bool otoKarLeftStatus = true;
bool otoKarRightStatus = true;
#endregion

void serialPort1_DataReceived(object sender,
System.IO.Ports.SerialDataReceivedEventArgs e)

```

```

    {
        int byteAmount = serialPort1.BytesToRead;
        byte[] tagArray = new byte[10];
        for (int i = 0; i < byteAmount; i++)
        {
            tagArray[i] = (byte)(serialPort1.ReadByte());
        }

        string TagNo = BitConverter.ToString(tagArray);
        //string tagnoparse = TagNo.TrimStart('-');
        string stationName = Reader.readProxTag(TagNo);
        string stationAbbr = stationName.Substring(0, 4);
        string stationSide = stationName.Substring(4, 1);

        switch (stationAbbr)
        {
            #region le41
            case "le41":

                if (stationSide == "A" & levent4StationStatus ==
false)
                {
                    levent4StationBusy.SendToBack();
                    levent4StationStatus = true;
                }
                else if (stationSide == "A" &
levent4StationStatus == true)
                {
                    levent4StationOk.SendToBack();
                    levent4StationStatus = false;
                    label4.Text = "4. LEVENT";
                    PlaySound("C:\\anons\\4LE.wav", new
System.IntPtr(), PlaySoundFlags.SND_SYNC);
                }
                else if (stationSide == "B" &
levent4StationStatus == false)//normal yonde gidiyor
                {
                    le4levLeftOk.SendToBack();
                    le4levLeftStatus = false;
                    levent4StationBusy.SendToBack();
                    levent4StationStatus = true;
                    PlaySound("C:\\anons\\LEV_GEL.wav", new
System.IntPtr(), PlaySoundFlags.SND_SYNC);
                }

                else if (stationSide == "B" &&
(le4levRightStatus == false || le4levLeftStatus == false))
                    //right mesgulse ters isletme, left mesgulse

```



```

normalden terse donus
    {
        le4levRightBusy.SendToBack();
        le4levRightStatus = true;
        le4levLeftBusy.SendToBack();
        le4levLeftStatus = true;
        levent4StationOk.SendToBack();
        levent4StationStatus = false;
        label4.Text = "4. LEVENT";
        PlaySound("C:\\anons\\4LE.wav", new
System.IntPtr(), PlaySoundFlags.SND_SYNC);
    }
    else if (stationSide == "B" &&
(levent4StationStatus == true))
        //if(levent4StationStatus == true) {}
        //II.yoldan makas gecisiyle geliyorsa, B bastan
direkt ters giris varsa
        {
            levent4StationOk.SendToBack();
            levent4StationStatus = false;
            label4.Text = "4. LEVENT";
            PlaySound("C:\\anons\\4LE.wav", new
System.IntPtr(), PlaySoundFlags.SND_SYNC);
        }

        break;
    #endregion

    #region lev1
    ...
    ...
    #region tak1
    ...

    }
}

private void button2_Click(object sender, EventArgs e)
{
}
}
}

```

- **Reader.cs;**

```

using System;
using System.Collections.Generic;

```

```

using System.Linq;
using System.Text;

namespace WindowsFormsApplication1
{
    public class Reader
    {
        public static Dictionary<string, string> tagListe = new
Dictionary<string, string>();

        public static void loadTagList()
        {
            tagListe.Add("00-00-05-01-EC-BF-0B-1F-43-03",
@"le41A");//4.levent A baş    -54-
            tagListe.Add("00-00-05-01-2C-68-0E-1F-51-03",
@"le41B");//4.levent B baş    -53-
            tagListe.Add("00-00-05-01-BC-7B-0C-1F-D0-03",
@"lev1A");//Levent A baş      -52-
            tagListe.Add("00-00-05-01-2C-A8-0C-1F-93-03",
@"lev1B");//Levent B baş      -51-
            tagListe.Add("00-00-05-01-BC-6D-0B-1F-C1-03",
@"gay1A");//Gayrettepe A baş  -50-
            tagListe.Add("00-00-05-01-0C-2A-0E-1F-33-03",
@"gay1B");//Gayrettepe B baş  -49-
            tagListe.Add("00-00-05-01-0C-CC-0D-1F-D6-03",
@"sis1A");//Şişli A baş       -48-
            tagListe.Add("00-00-05-01-5C-66-F3-1E-D3-03",
@"sis1B");//Şişli B baş       -47-
            tagListe.Add("00-00-05-01-5C-EE-F3-1E-5B-03",
@"osm1A");//Osmanbey A baş    -46-
            tagListe.Add("00-00-05-01-8C-1E-F3-1E-7B-03",
@"osm1B");//Osmanbey B baş    -45-
            tagListe.Add("00-00-05-01-DC-CC-F7-1E-FD-03",
@"tak1A");//Taksim A baş      -44-
            tagListe.Add("00-00-05-01-7C-95-F3-1E-00-03",
@"tak1B");//Taksim B baş      -43-

        }

        string dataRead;

        public static string readTag(string dataRead)
        {
            string trainName = "";
            string tagId = dataRead.Substring(0, 24);
            string antennaId = dataRead.Substring(24, 1);

```

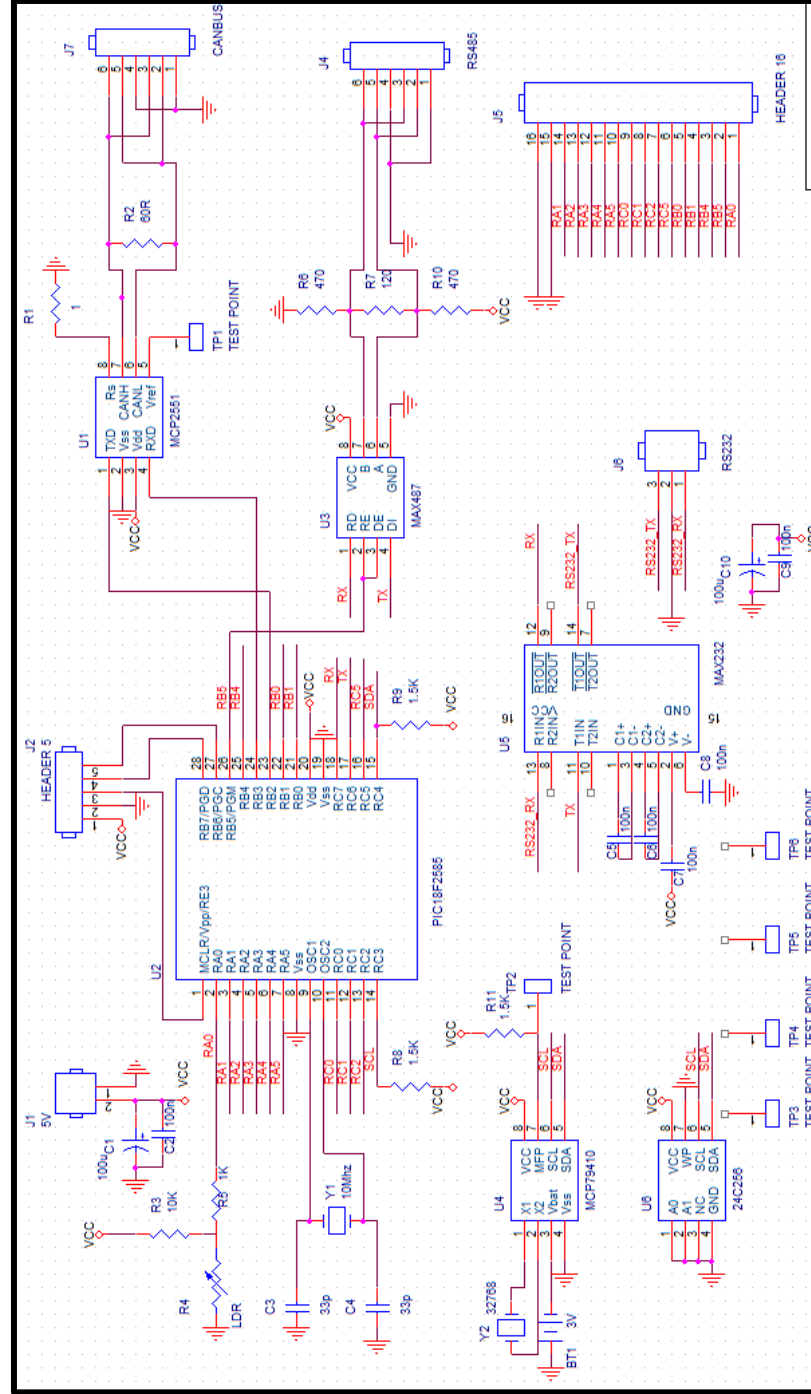
```
        if (tagListe.ContainsKey(tagId))
        {
            trainName = tagListe[tagId];
        }

        //string trainNumber = trainName.Substring(0, 3);
        //string tagNumber = trainName.Substring(3, 2);

        return trainName;
    }
    public static string readProxTag(string dataRead)
    {
        string stationName = "";
        string proxTagId = dataRead;

        if (tagListe.ContainsKey(proxTagId))
        {
            stationName = tagListe[proxTagId];
        }
        return stationName;
    }
}
}
```

EK 2 Örnek Model Elektronik Devre Şeması



Şekil 6-1 Örnek model, elektronik devre şeması

EK 3 Gömülü Sistem Yazılımı Kod Bloğu:

Ana İşlemci Kartı

```
#include <18F2585.h>
#fuses      WDT, WDT256, H4, NOFCMEN, BROWNOUT, CPD, STVREN, NODEBUG,
NOLVP, WRT, NOWRTD, NOWRTC, IESO, NOEBTR, NOEBTRB, NOMCLR, PROTECT, NOCPB,
NOWRTB

#use      delay(clock=4000000, restart_wdt)
#use      rs232( baud=9600, UART1, xmit=PIN_C6, rcv=PIN_C7, BITS =8,
STOP=1, PARITY=N, ERRORS, STREAM = PORT1)
#use      i2c(master, sda=PIN_C4, scl=PIN_C3, FAST)

#include <can-18F4580.c>
#define    PORTA      0X0F80
#define    PORTB      0X0F81
#define    PORTC      0X0F82
#define    PORTD      0X0F83
#define    PORTE      0X0F84
#define    PORTF      0X0F85
#define    PORTG      0X0F86
#define    TRISA      0X0F92
#define    TRISB      0X0F93
#define    TRISC      0X0F94
#define    TRISD      0X0F95
#define    TRISE      0X0F96
#define    TRISF      0X0F97
#define    TRISG      0X0F98
#define    LATA      0X0F89
#define    LATB      0X0F8A
#define    LATC      0X0F8B
#define    LATD      0X0F8C
#define    LATE      0X0F8D
#define    LATF      0X0F8E
#define    LATG      0X0F8F
#define    INDF2      0X0FDF
#define    POSTINC2    0X0FDE
#define    POSTDEC2    0X0FDD
#define    PREINC2     0X0FDC
#define    PLUSW2      0X0FDB
#define    FSR2H      0X0FDA
```

```

#define FSR2L          0X0FD9
#define STATUS         0X0FD8
#define TMR0H         0X0FD7
#define TMR0L         0X0FD6
#define T0CON         0X0FD5
#define OSCCON         0X0FD3
#define LVDCON         0X0FD2
#define WDTCON         0X0FD1
#define RCON           0X0FD0
#define FSR0L         0X0FE9
#define INDF0          0x0FEF
#define TOSU           0X0FFF
#define TOSH           0X0FFE
#define TOSL           0X0FFD
#define STKPTR         0X0FFC
#define PCLATU         0X0FFB
#define PCLATH         0X0FFA
#define PCL            0X0FF9
#define TBLPTRU        0X0FF8
#define TBLPTRH        0X0FF7
#define TBLPTRL        0X0FF6
#define TABLAT         0X0FF5
#define PRODH          0X0FF4
#define PRODL          0X0FF3
#define INTCON         0X0FF2
#define INTCON2        0X0FF1
#define INTCON3        0X0FF0
#define E2_CIHАЗ_ID_NO 0
#define CRC_CCITT      0X1021 //bit pattern (1)0001 0000 0010 0001
void ChkSumHesapla(int *p);
int ChkSumKontrol(int *p);
int IdKontrol(int *p);
int ChkSumDizi(int a);
int CevirAscii(int a);
void HafizaOku(void);
int16 Crc1021(int16, int8); // CCITT 1021 16 bit CRC
void GelenDiziDuzenle(void);
void CanDataYaz(void);
void GelenDataKontrol(void);
int SayacRunLed0;
int E2CihazNo;
int ChkSumUst;
int ChkSumAlt;
int SayacPort1Gelen;
int SayacPort1Giden;
int DiziPort1Giden[255];
int DiziPort1Gelen[255];
int KarakterRs232;
int DurumCanDataYaz;
int DurumSeriPortYaz;
int Sayac;
int Sayac1;
int Sayac2;
int DurumAnons1;

```

```

int      DurumAnons2;
int      BufferAnons1;
int      BufferAnons2;
int      DiziCanBusGonderilenMesaj[100];
int      Yil;
int      Ay;
int      Gun;
int      Saat;
int      Dakika;
int      Saniye;
int      DiziBos[10];
int      DiziOBCU[50];
int      DiziPortBuffer[10];
int      SayacMesajNo;
int      GelenDiziBoyut;
int      GercekDiziBoyut;
int      KapiDurumu;
int      ObcuDurumu;
int      Bayrak0;
int      Bayrak1;
int      Bayrak2;
int      TrenYon;
int      SayacCanBusGiden=0;
int16    ServisNumarasi;
int16    SeriNumarasi;
int16    SonrakiIstasyonNumarasi;
int16    HedefeUzaklik;
int16    HedefIstasyon;
int16    BaslangictanUzaklik;
int16    Hiz;
int16    GelenCrc;
int16    HesaplananCrc;
int      out_data[8];
int32    tx_id;
int1     tx_rtr=0;
int1     tx_ext=1;
int      tx_len;
int      tx_pri=3;
int      i;
float    Tq = ( 2.0 * (CAN_BRG_PRESCALAR+1) ) / 8.0;
int      Tq_num = 1 + (CAN_BRG_PROPAGATION_TIME+1) +
(CAN_BRG_PHASE_SEGMENT_1+1) + (CAN_BRG_PHASE_SEGMENT_2+1);

#int_TIMER0
void     Timer0InterruptGeldi () {

    setup_timer_0 (RTCC_INTERNAL|RTCC_DIV_16|RTCC_8_bit);

    if(++SayacRunLed0>10) {
        SayacRunLed0=0;
    }

    restart_wdt ();
}

```

```

#int_TBE
void SeriPort1DataGittiInterrupt(void){
    KarakterRs232 = DiziPort1Giden[SayacPort1Giden];

    if( (++SayacPort1Giden > 250) || (KarakterRs232 == 0) ){
        disable_interrupts(INT_TBE);
        return;
    }

    fputc(KarakterRs232, PORT1);
}

#int_RDA
void SeriPort1DataGeldiInterrupt(){
    KarakterRs232 = fgetc(PORT1);

    if( (KarakterRs232==0X02) ){
        SayacPort1Gelen=0;
    }
    else if(KarakterRs232==0X03){
        DiziPort1Gelen[SayacPort1Gelen] = KarakterRs232;

        if(DiziPort1Gelen[1]=='0' && DiziPort1Gelen[2]=='0'){
            if(DiziPort1Gelen[3]=='P' && DiziPort1Gelen[4]=='Y'){
                if(DiziPort1Gelen[5]=='1'){

                    for(Sayac=0; Sayac<20; Sayac++){
                        DiziCanBusGonderilenMesaj[Sayac]=0;
                    }

                    Sayac=0;

                    do{

DiziCanBusGonderilenMesaj[Sayac]=DiziPort1Gelen[Sayac+6];

                        }while(DiziPort1Gelen[6 + Sayac++]!=3);

                        DiziCanBusGonderilenMesaj[Sayac-1]=3;
                        DiziCanBusGonderilenMesaj[Sayac ]=0;

                        DurumCanDataYaz=10;
                    }
                }
            }
        }
        DiziPort1Gelen[SayacPort1Gelen++] = KarakterRs232;
    }

void CanDataYaz(void){
    int Sayac=0;

```



```

int    SayacBos;
char   Karakter;

if ( can_tbe()!=0){
    do{
        Karakter=DiziCanBusGonderilenMesaj[SayacCanBusGiden++];
        DiziBos[Sayac]=Karakter;
        Sayac++;

        fprintf(PORT1,"%c", Karakter);

        if(Karakter==0){
            DurumCanDataYaz=0;
            SayacCanBusGiden=0;
            Sayac=10;

            for(SayacBos=Sayac; SayacBos<8; SayacBos++){
                DiziBos[SayacBos]=' ';
            }
        }
    }while(Sayac<=7);

    out_data[0]=DiziBos[0];
    out_data[1]=DiziBos[1];
    out_data[2]=DiziBos[2];
    out_data[3]=DiziBos[3];
    out_data[4]=DiziBos[4];
    out_data[5]=DiziBos[5];
    out_data[6]=DiziBos[6];
    out_data[7]=DiziBos[7];

    tx_len = 8;
    tx_id++;
    i=can_putd(tx_id, out_data, tx_len, tx_pri, tx_ext, tx_rtr); //put
data on transmit buffer
    }
}

void main(void) {
    setup_timer_0(RTCC_INTERNAL|RTCC_DIV_1|RTCC_8_bit);

    port_b_pullups(TRUE);
    setup_wdt(WDT_ON);

    set_tris_a(0b11111111);
    set_tris_b(0b11111011);
    set_tris_c(0b10100000);
    set_tris_e(0b00000000);

    E2CihazNo=read_eeprom(E2_CHAZ_ID_NO);
    fprintf(PORT1, "-%d Start Port1 \x0d\x0a", E2CihazNo);

    DurumSeriPortYaz=10;

```

```

can_init();

enable_interrupts(INT_RDA);
enable_interrupts(INT_TIMER0);
enable_interrupts(GLOBAL);

while(TRUE) {
    if(DurumCanDataYaz!=0) {
        CanDataYaz();
    }
}
reset_cpu();
}

#ROM    byte 0XF00000 = {0X08,0X04,0X05,0X05,0X00,0X11,0X18,0X18}
#ROM    byte 0XF00008 = {0,0,0,0,0,0,0,0}
#ROM    byte 0XF00010 = {0,0,0,0,0,0,0,0}
#ROM    byte 0XF00018 = {0,0,0,0,0,0,0,0}
#ROM    byte 0XF00020 = {63,63,63,10,15,32,63,10}
#ROM    byte 0XF00028 = {24,29,26,35,31,63,15,24}
#ROM    byte 0XF00030 = {15,23,34,31,27,34,15,23}
#ROM    byte 0XF00038 = {26,21,23,63,32,10,26,53}
#ROM    byte 0XF00040 = {53,10,31,17,15,63,34,10}
#ROM    byte 0XF00048 = {32,10,31,20,25,63,37,15}
#ROM    byte 0XF00050 = {63,21,25,10,24,10,34,63}
#ROM    byte 0XF00058 = {59,59,58,57,56,52,52,52}
#ROM    byte 0XF00060 = {23,59,23,59,02,12,14,13}
#ROM    byte 0XF00068 = {29,34,15,31,24,32,22,30}
#ROM    byte 0XF00070 = {22,0,0,0,0,0,0,0}
#ROM    byte 0XF00078 = {0,0,0,0,0,0,0,0}
#ROM    byte 0XF00080 = {0,0,0,0,0,0,0,0}
#ROM    byte 0XF00088 = {0,0,0,0,0,0,0,0}
#ROM    byte 0XF00090 = {0,0,0,0,0,0,0,0}
#ROM    byte 0XF00098 = {0,0,0,0,0,0,0,0}
#ROM    byte 0XF000A0 = {0,0,0,0,0,0,0,0}
#ROM    byte 0XF000A8 = {0,0,0,0,0,0,0,0}
#ROM    byte 0XF000B0 = {0,0,0,0,0,0,0,0}
#ROM    byte 0XF000B8 = {0,0,0,0,0,0,0,0}
#ROM    byte 0XF000C0 = {0,0,0,0,0,0,0,0}
#ROM    byte 0XF000C8 = {0,0,0,0,0,0,0,0}
#ROM    byte 0XF000D0 = {0,0,0,0,0,0,0,0}
#ROM    byte 0XF000D8 = {0,0,0,0,0,0,0,0}
#ROM    byte 0XF000E0 = {0,0,0,0,0,0,0,0}
#ROM    byte 0XF000E8 = {0,0,0,0,0,0,0,0}
#ROM    byte 0XF000F0 = {0,0,0,0,0,0,0,0}
#ROM    byte 0XF000F8 = {0,0,0,0,0,0,0,0}

```

Terminal İşlemci Kartı

```

#include <18F2585.h>
#include <can-18F4580.c>
#include <string.h>
#fuses WDT, WDT256, H4, NOFCMEN, BROWNOUT, BORV27, CPD, STVREN,
NODEBUG, NOLVP, WRT, NOWRTD, NOWRTC, IESO, NOEBTR, NOEBTRB, NOMCLR, PROTECT,

```

```

NOCPB, NOWRTB
#use          delay(clock=40000000, restart_wdt)
#use          rs232( baud=19200, UART1, parity=N, xmit=PIN_C6, rcv=PIN_C7,
BITS =8, STOP=1, PARITY=N, ERRORS, STREAM = PORT1)

#zero_ram
#define      PORTA          0X0F80
#define      PORTB          0X0F81
#define      PORTC          0X0F82
#define      PORTD          0X0F83
#define      PORTE          0X0F84
#define      PORTF          0X0F85
#define      PORTG          0X0F86
#define      TRISA          0X0F92
#define      TRISB          0X0F93
#define      TRISC          0X0F94
#define      TRISD          0X0F95
#define      TRISE          0X0F96
#define      TRISF          0X0F97
#define      TRISG          0X0F98
#define      LATA          0X0F89
#define      LATB          0X0F8A
#define      LATC          0X0F8B
#define      LATD          0X0F8C
#define      LATE          0X0F8D
#define      LATF          0X0F8E
#define      LATG          0X0F8F
#define      POSTINC2      0X0FDE
#define      POSTDEC2      0X0FDD
#define      PREINC2       0X0FDC
#define      PLUSW2        0X0FDB
#define      STATUS        0X0FD8
#define      TMR0H         0X0FD7
#define      TMR0L         0X0FD6
#define      T0CON          0X0FD5
#define      OSCCON         0X0FD3
#define      LVDCON         0X0FD2
#define      WDTCON         0X0FD1
#define      RCON           0X0FD0
#define      TOSU           0X0FFF
#define      TOSH           0X0FFE
#define      TOSL           0X0FFD
#define      STKPTR         0X0FFC
#define      PCLATU         0X0FFB
#define      PCLATH         0X0FFA
#define      PCL            0X0FF9
#define      TBLPTRU         0X0FF8
#define      TBLPTRH         0X0FF7
#define      TBLPTRL         0X0FF6
#define      TABLAT         0X0FF5
#define      PRODH           0X0FF4
#define      PRODL           0X0FF3
#define      INTCON         0X0FF2
#define      INTCON2        0X0FF1

```

```

#define      INTCON3          0X0FF0
#define      E2_CIHAZ_ID_NO  0
#define      E2_CIHAZ_BOYUT  1
#define      FSR0L           = 0X0FE9
#define      FSR0H           = 0X0FEA
#define      FSR1L           = 0X0FE1
#define      FSR1H           = 0X0FE2
#define      FSR2L           = 0X0FD9
#define      FSR2H           = 0X0FDA
#define      POSTINC0        = 0X0FEE
#define      POSTINC1        = 0X0FE6
#define      INDF0           = 0X0FEF
#define      INDF1           = 0X0FE7
#define      INDF2           = 0X0FDF
#define      PinData         = PORTA.1
#define      PinA4           = PORTA.2
#define      PinA3           = PORTA.3
#define      PinA2           = PORTA.4
#define      PinA1           = PORTA.5
#define      PinA0           = PORTC.0
#define      PinGreen        = PORTC.1
#define      PinClk          = PORTC.2
#define      PinWe           = PORTC.5
#define      PinRed          = PORTB.1
#define      PinAe           = PORTB.0
#define      PinEnb          = PORTB.4
#define      PinBc1          = PORTB.5
#define      PinBc0          = PORTA.0
#define      PinTest         = PORTC.3
#define      N               4
#define      OV              3
#define      Z               2
#define      DC              1
#define      C               0
#define      W               0
#define      F               1

void      ChkSumHesapla(int *p);
int       ChkSumKontrol(int *p);
int       IdKontrol(int *p);
int       ChkSumDizi(int a);
int       CevirAscii(int a);
void      HafizaOku(void);
void      EkranSur(void);
void      DisplayAdresAyarla(int);
void      DisplayDegerAl(void);
int       RomKarakterAl24X24(int);
void      EkranRamSil(void);
int       SayacRunLed0;
int       E2CihazNo;
int       ChkSumUst;
int       ChkSumAlt;
int       SayacPort1Gelen;
int       SayacPort1Giden;

```

```

int      DiziPort1Giden[50];
int      DiziPort1Gelen[254];
int      KarakterRs232;
int      DurumEkranSur;
int      Kayma;
int      CihazBoyut;
int      Satir;
int      SayacKarakter;
int      DurumCanDataGeldi;
int      DurumAcilis=0;
int      SayacAcilisAlt;
int      SayacAcilisUst;
int      DiziPanelSurulecekKirmizi[24][30];
int      DiziPanelSurulecekYesil[24][30];
int      DiziPanelSurulen[250];
int      DiziCanBuffer[8];
int      SayacCursorIndex;
int      AraBosluk;
int      SayacCanBusGelen;
int      Yon;
int      in_data[8];
int      rx_len;
int      out_data[8];
int      tx_len;
int      tx_pri=3;
int      i;
int      Tq_num = 1 + (CAN_BRG_PROPAGATION_TIME+1) +
(CAN_BRG_PHASE_SEGMENT_1+1) + (CAN_BRG_PHASE_SEGMENT_2+1);

int1     tx_rtr=0;
int1     tx_ext=1;
int32    rx_id;
int32    tx_id;
float    Tq = ( 2.0 * (CAN_BRG_PRESCALAR+1) ) / 8.0;    // microsec
struct   rx_stat rxstat;

const char  DiziKarakterTablosu24X24[] =
"ABCÇDEFGĞHIİJKLMOÖPQRSŞTUÜVWYZ0123456789.,!abdcfgçhıijklmnoöprşştüüvyz[]"
;
#LOCATE    DiziPanelSurulecekKirmizi= 0X760
#LOCATE    DiziPanelSurulecekYesil   = 0XA30

#int_TIMER0
void  Timer0InterruptGeldi(){
// 4 ms interrupt
  setup_timer_0(RTCC_INTERNAL|RTCC_DIV_256|RTCC_8_bit);
  set_timer0(178);

  if(PinTest==0)  PinTest=1;
  else            PinTest=0;

  if(++Kayma>72){
    Kayma=0;

```

```

        DurumEkranSur=10;
    }

    if(DurumAcilis !=0){
        if(++SayacAcilisAlt==0){
            if(++SayacAcilisUst>4){
                DurumAcilis=0;

                DiziPanelSurulen[0]=' ';
                DiziPanelSurulen[1]=' ';
                DiziPanelSurulen[2]=' ';
                DiziPanelSurulen[3]=' ';
                DiziPanelSurulen[4]=' ';
                DiziPanelSurulen[5]=' ';
                DiziPanelSurulen[6]=' ';
                DiziPanelSurulen[7]=' ';
                DiziPanelSurulen[8]=' ';
                DiziPanelSurulen[9]='.';
                DiziPanelSurulen[10]=0;

                DisplayDegerAl();
                EkranSur();
            }
        }
    }

    restart_wdt();
}

#int_CANRX0
void CanDataGeldi(void){
    int Sayac=0;
    int SayacBos=0;
    char Karakter;

    can_getd(rx_id, &in_data[0], rx_len, rxstat);

    do{
        Karakter = in_data[Sayac];

        DiziPanelSurulen[SayacCanBusGelen++]=Karakter;

        if(Karakter==3){

            for(SayacBos=SayacCanBusGelen-1; SayacBos<11; SayacBos++){
                DiziPanelSurulen[SayacBos]=' ';
            }

            DurumCanDataGeldi=10;
            SayacCanBusGelen=0;
            Sayac=8;
            DiziPanelSurulen[10]=0;

            EkranRamSil();
            EkranSur();
        }
    }while(1);
}

```

```

    }

    Sayac++;
}while(Sayac<=7);

}

#int_RDA
void SeriPort1DataGeldiInterrupt(){
    KarakterRs232 = fgetc(PORT1);

    if (KarakterRs232==2 || SayacPort1Gelen >240 ){
        SayacPort1Gelen=0;
    }
    else if(KarakterRs232==3 && DiziPort1Gelen[0]==2){
        if(ChkSumKontrol(DiziPort1Gelen)==1){
            if(IdKontrol(DiziPort1Gelen)==1){
                DurumAcilis=0;

                if(DiziPort1Gelen[3]=='S'){ // Saat ile Alakalı
                    if(DiziPort1Gelen[4]=='R'){ // Saat Set
                        reset_cpu();
                    }
                }
                else if(DiziPort1Gelen[3]=='R'){
                    if(DiziPort1Gelen[4]=='S'){
                        reset_cpu();
                    }
                }
                else if(DiziPort1Gelen[3]=='T'){
                    int Satir, Sutun;

                    if(DiziPort1Gelen[4]=='1'){
                        fprintf(PORT1, "TEST 1\\X0D\\X0A");
                        for(Satir=0; Satir<24; Satir++){
                            for(Sutun=0;Sutun<30;Sutun++){
                                DiziPanelSurulecekYesil[Satir][Sutun]=0XFF;
                                DiziPanelSurulecekKirmizi[Satir][Sutun]=0;
                            }
                        }
                    }
                    else if(DiziPort1Gelen[4]=='2'){
                        fprintf(PORT1, "TEST 2\\X0D\\X0A");
                        for(Satir=0; Satir<24; Satir++){
                            for(Sutun=0;Sutun<30;Sutun++){
                                DiziPanelSurulecekYesil[Satir][Sutun]=0;
                                DiziPanelSurulecekKirmizi[Satir][Sutun]=0XFF;
                            }
                        }
                    }
                }
            }
        }
    }
}

```

```

else if(DiziPort1Gelen[4]=='3'){
    fprintf(PORT1, "TEST 3\\X0D\\X0A");
    for(Satir=0; Satir<24; Satir++){
        for(Sutun=0;Sutun<30;Sutun++){
            DiziPanelSurulecekYesil[Satir][Sutun]=0XFF;
            DiziPanelSurulecekKirmizi[Satir][Sutun]=0XFF;
        }
    }
}
else if(DiziPort1Gelen[4]=='4'){
    fprintf(PORT1, "TEST 4\\X0D\\X0A");
    for(Satir=0; Satir<24; Satir++){
        for(Sutun=0;Sutun<30;Sutun++){
            DiziPanelSurulecekYesil[Satir][Sutun]=0XAA;
            DiziPanelSurulecekKirmizi[Satir][Sutun]=0X55;
        }
    }
}
else if(DiziPort1Gelen[4]=='5'){
    fprintf(PORT1, "TEST 4\\X0D\\X0A");
    for(Satir=0; Satir<24; Satir++){
        for(Sutun=0;Sutun<30;Sutun++){
            DiziPanelSurulecekYesil[Satir][Sutun]=0X55;
            DiziPanelSurulecekKirmizi[Satir][Sutun]=0XAA;
        }
    }
}
}
else if(DiziPort1Gelen[3]=='C'){ //
Cihaz Boyutu
    if(DiziPort1Gelen[4]=='B'){
        CIHAZBOYUT = 100*(DiziPort1Gelen[5]-0X30) +
10*(DiziPort1Gelen[6]-0X30) + (DiziPort1Gelen[7]-0X30);
        write_eeprom(E2_CIHAZ_BOYUT, CIHAZBOYUT);
    }
}
else if(DiziPort1Gelen[3]=='P'){ //
Cihaz Boyutu
    if(DiziPort1Gelen[4]=='O'){
        fprintf(PORT1, "CIHAZ PARAMETRELERI\\X0D\\X0A");
        fprintf(PORT1, "E2No      %u\\X0D\\X0A", E2CihazNo);
        fprintf(PORT1, "CihazBoyut  %u\\X0D\\X0A", CIHAZBOYUT);
    }
}
else if(DiziPort1Gelen[4]=='Y'){
    int i=0;
    int Karakter;

    do{
        Karakter=DiziPort1Gelen[i+5];
        DiziPanelSurulen[i]=Karakter;
        i++;
    }while((Karakter != 0X05) && ((i)<250));
}

```



```

        SayacKarakter=0;

        DiziPanelSurulen[i-1]=0;
        DisplayDegerAl();

        DiziPort1Giden[0]='0';
        DiziPort1Giden[0]='K';
        DiziPort1Giden[0]=0;
        SayacPort1Giden=0;
        enable_interrupts(INT_TBE);
    }
}
}
}
}
DiziPort1Gelen[SayacPort1Gelen++] = KarakterRs232;
}
#endif
void SeriPort1DataGittiInterrupt(void){
    KarakterRs232 = DiziPort1Giden[SayacPort1Giden];

    if( (++SayacPort1Giden > 250) || (KarakterRs232 == 0) ){
        disable_interrupts(INT_TBE);
        return;
    }

    fputc(KarakterRs232, PORT1);
}
int IDKontrol(*p){
    if ( *(p+1)==0X30  && *(p+2)==0X30 ){
        return 1;
    }
    else if (E2CihazNo == (10*(*(p+1)-0X30) + (*(p+2)-0X30 ))){
        return 1;
    }

    return -1;
}
int ChksumKontrol(*p){
    int i=1, ChkSumHesap, ChkSum=0;

    do{
        ChkSum = ChkSum ^ *(p+(i++));
    }while((*p+i)!= 0X05) && i != 0 );

    ChkSumHesap = swap(ChkSumDizi(*(p+(++i))) | ChkSumDizi(*(p+(++i))));

    if(ChkSum == ChkSumHesap){ return 1;}
    return 0;
}
void ChkSumHesapla(*p){
    int ChkSum=0;

    p++;
}

```

```

do{
    ChkSum=ChkSum ^ *p;
}while(*(++p) != 5);

ChkSumAlt=CevirAscii(ChkSum&0X0F);
ChkSumUst=CevirAscii(ChkSum>>4 & 0X0F);
}
int  ChkSumDizi(int a){
    if(a<'A')    return a-0X30;
    return  a - 0X37;
}
int  CevirAscii(int a){
    if(a<10)    return  a + 0X30;
    return  a + 0X37;
}
void  DisplayAdresAyarla(int Adres){

    if(Bit_test(Adres,4)==1){
        PinA4=1;
    }
    else{
        PinA4=0;
    }

    if(Bit_test(Adres,3)==1){
        PinA3=1;
    }
    else{
        PinA3=0;
    }

    if(Bit_test(Adres,2)==1){
        PinA2=1;
    }
    else{
        PinA2=0;
    }

    if(Bit_test(Adres,1)==1){
        PinA1=1;
    }
    else{
        PinA1=0;
    }

    if(Bit_test(Adres,0)==1){
        PinA0=1;
    }
    else{
        PinA0=0;
    }
}

```

```

void EkranSur(void){
    int SayacSatir;
    int SayacSutun;
    int SayacBit;
    int SayacBayt;
    int DataKirmizi;
    int DataYesil;
    int Sayac;

#asm
    lfsr    0, 0X0760
    lfsr    1, 0X0A30

    clrf    SayacSatir
    bsf     PinData

cevrin2:
    bcf     PinA4
    bcf     PinA3
    bcf     PinA2
    bcf     PinA1
    bcf     PinA0

    btfsc   SayacSatir,4
    bsf     PinA4
    btfsc   SayacSatir,3
    bsf     PinA3
    btfsc   SayacSatir,2
    bsf     PinA2
    btfsc   SayacSatir,1
    bsf     PinA1
    btfsc   SayacSatir,0
    bsf     PinA0
    movlw   30
    movwf   SayacBayt

cevrin1:
    movlw   8
    movwf   SayacBit
    movf    postinc0,w
    movwf   DataKirmizi
    movf    postinc1,w
    movwf   DataYesil

cevrin0:
    bcf     PinClk
    bcf     PinRed
    bcf     PinGreen
    rrcf    DataKirmizi,f
    btfsc   STATUS,C
    bsf     PinRed
    rrcf    DataYesil,f
    btfsc   STATUS,C
    bsf     PinGreen

```

```

bsf    PinClk

decfsz SayacBit,f
bra    cevrin0

decfsz SayacBayt,f
bra    cevrin1

bsf    PinAe
bsf    PinWe
bcf    PinWe
bcf    PinAe

incf   SayacSatir,f
movf   SayacSatir,w
sublw  24
btfss  status,z
bra    cevrin2

bsf    PinEnb
bsf    PinBc0
bsf    PinBc1
bcf    PinData
#endasm
}

void DisplayDegerAl(void){
    int KarakterIndex=0;
    int Karakter;
    int Sayac=0;
    int Durum=1;

    SayacKarakter=0;
    SayacCursorIndex=0;
    //   fprintf(PORT1, "\X0D\X0A");

    do{
        Karakter=DiziPanelSurulen[SayacKarakter];

        Sayac=0;

        do{
            Durum=DiziKarakterTablosu24X24[Sayac];
        }while( (Durum!= NULL) && (DiziKarakterTablosu24X24[Sayac++] !=
Karakter) );

        if(Durum!=NULL){
            --Sayac;
            RomKarakterAl24X24(Sayac);
        }

        SayacKarakter++;

    }while((Karakter!=0) && (SayacKarakter<254));
}

```

```

for(Sayac=0; Sayac<15; Sayac++){
    DiziPanelSurulen[Sayac]=' ';
}

DurumEkranSur=10;
}
int RomKarakterAl24X24(int KarakterIndex){
    int16 Adres;
    int KarakterUzunlugu;
    int i;

    Adres=KarakterIndex*731 + 0X77001;

    KarakterUzunlugu = read_program_eeprom(Adres++);
    SayacCursorIndex += KarakterUzunlugu ;
    SayacCursorIndex += AraBosluk;

    if(Yon==0){
        for(i=0; i<24; i++){
DiziPanelSurulecekYesil[i] [(SayacKarakter*3)+0]=read_program_eeprom(Adres++)
;

DiziPanelSurulecekYesil[i] [(SayacKarakter*3)+1]=read_program_eeprom(Adres++)
;

DiziPanelSurulecekYesil[i] [(SayacKarakter*3)+2]=read_program_eeprom(Adres++)
;
        }
    }
    else if(Yon==10){
        for(i=0; i<24; i++){
DiziPanelSurulecekKirmizi[i] [(SayacKarakter*3)+0]=read_program_eeprom(Adres+
+);

DiziPanelSurulecekKirmizi[i] [(SayacKarakter*3)+1]=read_program_eeprom(Adres+
+);

DiziPanelSurulecekKirmizi[i] [(SayacKarakter*3)+2]=read_program_eeprom(Adres+
+);
        }
    }
    else if(Yon==20){
        for(i=0; i<24; i++){
DiziPanelSurulecekYesil[i] [(SayacKarakter*3)+0]=DiziPanelSurulecekKirmizi[i]
[(SayacKarakter*3)+0]=read_program_eeprom(Adres++);

DiziPanelSurulecekYesil[i] [(SayacKarakter*3)+1]=DiziPanelSurulecekKirmizi[i]
[(SayacKarakter*3)+1]=read_program_eeprom(Adres++);

```

```

DiziPanelSurulecekYesil[i][ (SayacKarakter*3)+2]=DiziPanelSurulecekKirmizi[i]
[ (SayacKarakter*3)+2]=read_program_eeprom(Adres++);
    }
}

return KarakterUzunlugu;
}

void EkranRamSil(void){
    int Satir, Sutun;

    for(Satir=0; Satir<24; Satir++){
        for(Sutun=0;Sutun<30;Sutun++){
            DiziPanelSurulecekYesil[Satir][Sutun]=0;
            DiziPanelSurulecekKirmizi[Satir][Sutun]=0;
        }
    }
}

void HafizaOku(void){
    E2CihazNo = read_eeprom(E2_CIHAZ_ID_NO);
    CIHAZBOYUT = read_eeprom(E2_CIHAZ_BOYUT);

    if(read_eeprom(8)== 0){
        write_eeprom(8,10);
        Yon=0;
    }
    else if(read_eeprom(8)== 10){
        write_eeprom(8,20);
        Yon=10;
    }
    else {
        write_eeprom(8,0);
        Yon=20;
    }

    Yon=20;
}

void main(void) {
    setup_timer_0(RTCC_INTERNAL|RTCC_DIV_1|RTCC_8_bit);
    port_b_pullups(TRUE);
    setup_wdt(WDT_ON);

    delay_ms(100);

    set_tris_a(0b11000000);
    set_tris_b(0b11001000);
    set_tris_c(0b10000000);
    set_tris_e(0b00000000);

    HafizaOku();

    // delay_ms(100);

```

```

EkranRamSil();
EkranSur();

DiziPanelSurulen[0]=' ';
DiziPanelSurulen[1]='M';
DiziPanelSurulen[2]='A';
DiziPanelSurulen[3]='L';
DiziPanelSurulen[4]='T';
DiziPanelSurulen[5]='E';
DiziPanelSurulen[6]='P';
DiziPanelSurulen[7]='E';
DiziPanelSurulen[8]=' ';
DiziPanelSurulen[9]=' ';
DiziPanelSurulen[10]=0;

DisplayDegerAl();
EkranSur();
can_init(); //can_set_mode(CAN_OP_NORMAL);

fprintf(PORT1, "-%d Start Port1 \X0D\X0A", E2CihazNo);

enable_interrupts(INT_RDA);
enable_interrupts(INT_TIMER0);
enable_interrupts(INT_CANRX0);
enable_interrupts(GLOBAL);

while(TRUE) {
    if(DurumCanDataGeldi!=0) {
        DisplayDegerAl();
        EkranSur();
        DurumCanDataGeldi=0;
    }

    if(DurumEkranSur!=0) {
        EkranSur();
        DurumEkranSur=0;
    }
}
reset_cpu();
}
#ROM    byte 0XF00000 = {0X05, 240, 0X05,0X05,0X00,0X11,0X18,0X18}
#ROM    byte 0XF00008 = {0,0,0,0,0,0,0,0}
#ROM    byte 0XF00010 = {0,0,0,0,0,0,0,0}
#ROM    byte 0XF00018 = {0,0,0,0,0,0,0,0}
#ROM    byte 0XF00020 = {0,0,0,0,0,0,0,0}
#ROM    byte 0XF00028 = {0,0,0,0,0,0,0,0}
#ROM    byte 0XF00030 = {0,0,0,0,0,0,0,0}
#ROM    byte 0XF00038 = {0,0,0,0,0,0,0,0}
#ROM    byte 0XF00040 = {0,0,0,0,0,0,0,0}
#ROM    byte 0XF00048 = {0,0,0,0,0,0,0,0}
#ROM    byte 0XF00050 = {0,0,0,0,0,0,0,0}
#ROM    byte 0XF00058 = {0,0,0,0,0,0,0,0}
#ROM    byte 0XF00060 = {0,0,0,0,0,0,0,0}
#ROM    byte 0XF00068 = {0,0,0,0,0,0,0,0}

```

```

#ROM    byte 0XF00078 = {0,0,0,0,0,0,0,0}
#ROM    byte 0XF00080 = {0,0,0,0,0,0,0,0}
#ROM    byte 0XF00088 = {0,0,0,0,0,0,0,0}
#ROM    byte 0XF00090 = {0,0,0,0,0,0,0,0}
#ROM    byte 0XF00098 = {0,0,0,0,0,0,0,0}
#ROM    byte 0XF000A0 = {0,0,0,0,0,0,0,0}
#ROM    byte 0XF000A8 = {0,0,0,0,0,0,0,0}
#ROM    byte 0XF000B0 = {0,0,0,0,0,0,0,0}
#ROM    byte 0XF000B8 = {0,0,0,0,0,0,0,0}
#ROM    byte 0XF000C0 = {0,0,0,0,0,0,0,0}
#ROM    byte 0XF000C8 = {0,0,0,0,0,0,0,0}
#ROM    byte 0XF000D0 = {0,0,0,0,0,0,0,0}
#ROM    byte 0XF000D8 = {0,0,0,0,0,0,0,0}
#ROM    byte 0XF000E0 = {0,0,0,0,0,0,0,0}
#ROM    byte 0XF000E8 = {0,0,0,0,0,0,0,0}
#ROM    byte 0XF000F0 = {0,0,0,0,0,0,0,0}
#ROM    byte 0XF000F8 = {0,0,0,0,0,0,0,0}

```

```

#ORG 0X7700, 0XBFFF // 24X24 font tablosu

```

```

#ROM int 0X7700 = {

```

```

0x0F,
0xFF,0x07,0x00,0xFF,0x0F,0x00,0xFF,0x1F,
0x00,0x07,0x3C,0x00,0x07,0x38,0x00,0x07,
0x38,0x00,0x07,0x38,0x00,0x07,0x38,0x00,
0x07,0x38,0x00,0x07,0x38,0x00,0xFF,0x1F,
0x00,0xFF,0x1F,0x00,0xFF,0x3F,0x00,0x07,
0x78,0x00,0x07,0x70,0x00,0x07,0x70,0x00,
0x07,0x70,0x00,0x07,0x70,0x00,0x07,0x70,
0x00,0x07,0x70,0x00,0x07,0x78,0x00,0xFF,
0x7F,0x00,0xFF,0x3F,0x00,0xFF,0x1F,0x00,
...
...
...
0x0F,
0xF8,0x0F,0x00,0xFC,0x1F,0x00,0xFE,0x3F,
0x00,0x0F,0x78,0x00,0x07,0x70,0x00,0x07,
0x70,0x00,0x07,0x00,0x00,0x07,0x00,0x00,
0x07,0x00,0x00,0x07,0x00,0x00,0x07,0x00,
0x00,0x07,0x00,0x00,0x07,0x00,0x00,0x07,
0x00,0x00,0x07,0x00,0x00,0x07,0x00,0x00,
0x07,0x00,0x00,0x07,0x00,0x00,0x07,0x70,
0x00,0x07,0x70,0x00,0x0F,0x78,0x00,0xFE,
0x3F,0x00,0xFC,0x1F,0x00,0xF8,0x0F,0x00,

```


EK 4 Akış Diyagramları

