

1. GİRİŞ

Günümüzde, kişisel bilgisayar tabanlı uzaktan, kablolu ve kablosuz ölçme, gözleme ve kontrol yani telemetri sisteminin kazandırdığı hız ve erişim kolaylığının ham petrol üretimindeki katkısının ne kadar önemli olduğu açıktır.

Yukarıda belirtilen amaca uygun olarak geliştirilen telemetri sistemindeki kablosuz haberleşmede, şirketin telsiz haberleşme kanallarından birisi kullanılmış, böylece uzak mesafelere erişim problemi ortadan kalkmıştır.



Resim.1. Terminal Birimi

Yukarıdaki telemetri için geliştirilmiş terminal biriminin resminde, terminal ana kartı, telsiz cihazı ve güçkaynağı kutulanmış hazır olarak görülmektedir.

Uzaktan veri aktarımı mevcut telsiz sistemi üzerinde temellendiğinden telsizler, modülasyon/demodülasyon, anten, ayrıca tüm terminal birimleri mikrodenetleyici tabanlı tasarımlar olduğundan ve mikrodenetleyici-bilgisayar haberleşmeside paralel

port üzerinden yapıldığından Paralel Port hakkında da ileride gerekli bilgiler kısaca sunulmuştur.

1. Telsiz

1.1. Telsizin Tarihsel Gelişimi

Telli yazılı haberleşmenin yerini yavaş yavaş telli ses haberleşmesinin almaya başladığı yıllarda, arada herhangi bir fiziki bağlantı olmadan yapılabilecek haberleşme yöntemleri aranmaya başlamıştır. 1879 yılında, Alman Heinrich Hertz, pil, küreler ve bakır tellerle oluşturduğu düzenele, iki küre arasında kıvılcım atlamaları keşfederek arada tel olmadan yapılan haberleşmenin temelini atmıştır. Böylece Hertz, aslında ileride kendi adıyla anılacak olan “frekans” kavramını ortaya çıkarmıştır[6].

Frekans, “bir dalganın, saniyede yaptığı salınım sayısıdır” şeklinde basitce tanımlanabilir[3].

Bu keşiften esinlenen ve faydalanan birçok bilim adamı ilerletici çalışmalar yapmışlardır. Fransız Eduard Bronly 1888’de radyo dalgalarını yakalamayı başarmıştır.

İngiliz Oliver Lodge, 1894’de, elektromanyetik verici ve alıcı aygıtlarının dalga uzunluklarının birbirine uydurulması gerektiğini ortaya koymuş ve Hertz’in buluşundan yola çıkarak ilk kez Mors Alfabesine göre hazırlanmış bir mesajı, 40 m uzağa arada fiziki bir bağlantı olmadan iletmeyi başarmıştır.

Aynı yıllarda, Rus Aleksandr Popov’da Lodge gibi Hertz’in buluşu olan dalgalardan faydalanarak yine Mors alfabesinde hazırlanmış bir mesajı, arada hiçbir fiziki bağlantı olmadan karşıya gönderebilmiştir.

Bütün bu olumlu gelişmelerden sonra Guglielmo Marconi tamamen deneme yanılma yoluyla, tel olmaksızın uzak mesafelerle haberleşmenin temel ve önemli faktörlerinden biri olan anteni ve anten-yeryüzü etkileşimini keşfetmiştir. Ayrıca, elektromanyetik dalgaları inceleyip, elektrik devresi içerisinde üretilen elektrik

titreşiminin dalgalı bir yapıyla yayınlara yol açtığını ve dalga uzunlukları gibi özelliklerinin bulunduğunu ortaya koymuştur. Bu keşifden yola çıkarak, 1884 yılında evi ve bahçesi arasında RF dalga boyunda haberleşmeyi başarmıştır. Marconi, özellikle gemiler arası telsiz-telgraf haberleşmesini sağlamıştır. Çalışmalarına tamamen ticari bir amaçla başlayan Marconi, trenlere de telsiz cihazını yerleştirerek ilk mobil telsiz uygulamasına imzasını atmıştır.

Araştırmalarına devam eden Marconi, daha yüksek ve büyük antenler yaparak her türlü engellemeye rağmen, 1897 yılında, İngiltere'de Telsiz-Telgraf Şirketini kurup imal ettiği telsiz cihazlarıyla, İngiltere'nin Doures Limanı ile Fransa'nın Wimereux arasında 50 km.lik mesafede ilk telsiz-telgraf haberleşmesini gerçekleştirmiştir. 1901 yılında, İngiltere'nin Carnivailles kenti ile ABD'nin Terre Neure kenti arasında 3400 km.lik mesafede ilk Atlantik ötesi telsiz-telgraf haberleşmesini sağlamıştır[11].

1906 yılına kadar, telsiz üzerinden tamamen Mors haberleşmesi yapılmıştır. Bu yıldan sonra, telsizlerden sesin iletilmesi çalışmaları başlamıştır. Birbirini takip eden denemeler neticesinde sesin, telsizden iletilmesi başarıldı. Bu başarıda elektron tüplerinin keşfinin payı çok büyüktür. Keşfedilen üçlü elektron tüpü sayesinde sesin, modüle, demodüle ve yükseltme işlemi yapılarak net bir ses haberleşmesi yapılmıştır[10].

Araştırma ve buluşlar durmaksızın devam ederken, Edwin Amstrong, Genlik modülasyonundan faydalanarak "Frekans Modülasyonu", FM'i gerçekleştirdi[7].

FM'in özellikle müzik ve ses iletiminde kullanılması hızla yayıldı. 1930 yılında, adı daha sonra Motorola olarak değişecek olan Galvin Şirketi tarafından batarya ile çalışan ilk telsiz yapılmıştır. Genlik Modülasyon tekniği ile çalışan ilk el telsizini yapan Motorola Firması, 1943 yılında FM modülasyonlu ilk el telsizini de yapmıştır[6].

Durmuyan araştırmalar sonucunda, transistör ve entegre komponentlerin bulunuşlarıyla, telsiz ve telefon teknolojisi de aynı oranda gelişerek, 1969'da Ay'a gönderilen ilk insanla, yer kontrol arasında, yine Motorola'nın üretimi olan telsizlerle haberleşme yapılabildiği[11].

Telsizdeki bu hızlı gelişme ve aktarıcı telsizlerin bulunmasıyla konvansiyonel Haberleşme Sistemleri'nin kullanımı ortaya çıkmış ve de bilgisayarın kullanılmasıyla da “Geniş Alan Kaplama” ve “Trunk” sistemleri geliştirilerek yeni telsiz haberleşme sistemleri yaygın bir şekilde haberleşmede kullanılmaktadır.

1.1.2. Telsizlerin Bazı Genel Özellikleri

Daha sonra anlatılacağı gibi, telsiz haberleşmesinde karşı tarafa gönderilmek istenilen bilgi, (ses, veri, vb.) “taşıyıcı” denilen bir dalganın üzerine bindirilerek yani modüle edilerek gönderilir. Bu taşıyıcı dalga telsiz tarafından üretilir. Kanallardaki yayınların birbirine karışmamamsı için, her kanala ayrı bir taşıyıcı dalga gereklidir. Telsizlerdeki bu kanal frekanslarının üretildiği devrelere “OSİLATÖR” denir. Osilatör devresinde, temel (fundamantal) frekansın elde edilmesi için “KRİSTAL” adı verilen elektronik devre elemanları kullanılır[7].

Her kanal için bir alıcı, birde verici frekansı olduğundan iki adet kristale ihtiyaç vardır. 6 kanallı bir telsiz için $2 \times 6 = 12$ adet kristale ihtiyaç vardır.

İşte bu dezavantajı yok etmek için, bir tek referans kristalinden bütün kanal frekanslarını üretebilecek “SENTEZÖR” teknolojisi geliştirilmiştir.

Bu teknolojiye, bir tane temel kristal vardır ve bir “EPROM”a kanal frekansları programlanır. Temel kristalin ürettiği frekans referans alınarak, EPROMa yüklenmiş olan kanal frekans bilgilerine göre Sentezör devresi tarafından Kanal Taşıyıcı Frekansları üretilir.

Sentezörde, frekans üretilmesi işlemi çok hızlı bir şekilde yapılır. Telsiz normalde dinlemede iken sentezör o kanalın alıcı frekansını üretir ve o frekansa kilitlenir. Telsizin Bas/Konuş mandalına basılıp verici konumuna alındığında, hemen programlı verici frekansına göre, bölücüler devreye girerek hızlı bir döngü ile bu seferde verici frekansı üretilir. Mandal bırakılınca, tekrar bölme döngüsü başlayarak alıcı frekansını üretip, dinleme moduna geçilir[6].

1.1.3. Anten

Uzak mesafeler arasında iletken vazifesi gören, vericiden gelen elektromanyetik dalgaları atmosfere yayan veya başka bir verici anteninden yayılan elektromanyetik dalgaları yakalayıp alıcıya aktaran telsiz elemanıdır[2].

Anten boyu,

$$\lambda \text{ (dalga boyu)} = C \text{ (ışık hızı)} / f \text{ (taşıyıcı frekansı)}$$

formülüyle hesaplanır.

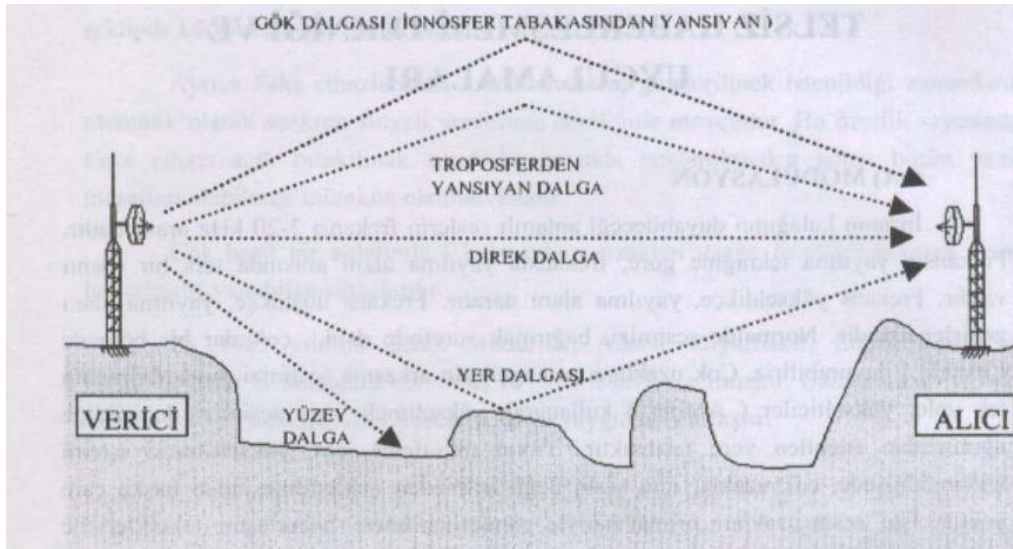
Görüldüğü gibi, çalışma frekansı yükseldikçe dalga boyu (λ), dolayısıyla anten boyu düşer. Bu formülden çıkan “TAM DALGA” anten boyu olup, 2’ye bölünerek “YARIM DALGA”, 4’e bölünerek de “ÇEYREK DALGA” anten boyları hesaplanır[3].

1.1.4. Modülasyon

İnsan kulağının duyabileceği anlamlı seslerin frekansı 3-20 kHz arasındadır. Frekansın yayılma tekniğine göre, frekansla yayılma alanı arasında ters bir orantı vardır. Frekans yükseldikçe yayılma alanı daralır, düştükçe de yayılma alanı genişlemektedir[3].

Ses frekansının ulaşamadığı uzak mesafelere kadar ulaşabilen frekanslar mevcuttur. İşte haberleşmede, sesin uzaklara gönderilmesinin tek yolu, çok uzaklara ulaşabilen bu yayınların yani frekansların içerisine saklanarak gönderilmesidir. Teknikte yapılan bu işleme, yani, anlamlı bilginin (ses, veri vb.) taşıyıcı dalga üzerine bindirilmesine “modülasyon” denir. Modüle edilen bilgi de anten üzerinden atmosfere yayılır.

Taşıyıcı dalganın istenilen yere ulaşmasında seçiciliği yapan, frekansdır. Şekil.3.’de görüldüğü gibi, verici istasyonda taşıyıcı dalga tarafından zarflanır ve frekansa göre uzaktaki bir yere götürülür.



Şekil.1. RF Dalgalarının Yayılma Yolları

Yayını alan alıcı, modüle işleminin tam tersi işlem gerçekleştirerek anlamlı bilgiyi taşıyıcı dalgadan ayırır. Bu işleme demodülasyon denir[7].

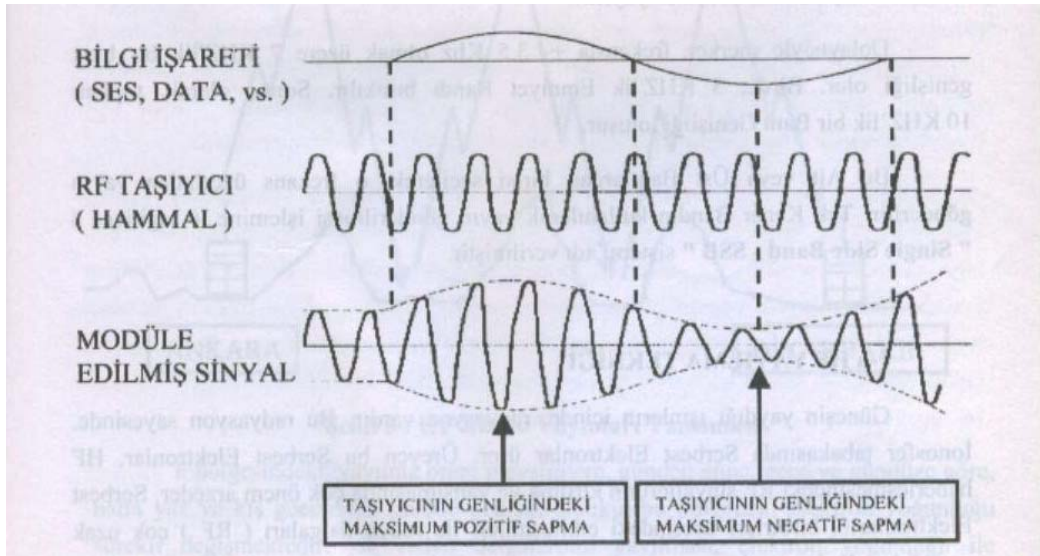
Taşıyıcı dalga frekansının yayılma alanı, yapılan RF haberleşmesinin mesafesini tayin etmektedir. Şekil 1’de görüldüğü gibi gök ve yer dalgaları telsiz haberleşmelerinde tercihli olarak da kullanılmaktadır[6].

Modülasyon, modüle işleminin yapılışına göre 4’e ayrılır[3].

1. Genlik (Amplitude) Modülasyonu
2. Frekans Modülasyonu
3. Pulse Modülasyonu
4. Faz Modülasyonu

1.1.5. Genlik Modülasyonu

Bir radyo frekans dalgası genliğinin, gönderilecek bilgi (ses) frekans sinyalinin genliğine ve frekansına bağlı olarak değiştirilmesi işlemine “**Genlik Modülasyonu**” denir[7].



Şekil.2. Genlik Modülasyonu

Telsiz sinyalindeki RF sinyalinin gücü, bilgi sinyalinin hem genliğine hemde frekansına bağlı olarak artar veya azalır. Bilgi sinyali, değer itibarı ile yükselirken RF çıkış gücü artar. Bilgi değer itibarıyla azalırken RF çıkış gücü azalır. Eğer, bilgi sinyalinin frekansı yüksek ise, RF çıkış gücündeki bu artma ve azalma daha çabuk olur. Bilgi sinyalinin, frekansı düşük ise, RF çıkış gücündeki değişme daha yavaş olur.

Genlik Modülasyonda, taşıyıcı üzerine bindirilen ses frekansının maksimum seviyesi değerinde, (+) ve (-) yönde Alt (LSB) ve Üst (USB) Bant'lar oluşur[2].

Örnek; Taşıyıcı RF 5000kHz ve bilgi (ses) frekansı 3.5 olarak düşünülürse;

$$5000 + 3.5 = 5003.5 \text{ kHz Üst Yan Bant (USB)}$$

$$5000 - 3.5 = 4996.5 \text{ kHz Alt Yan Bant (LSB)}$$

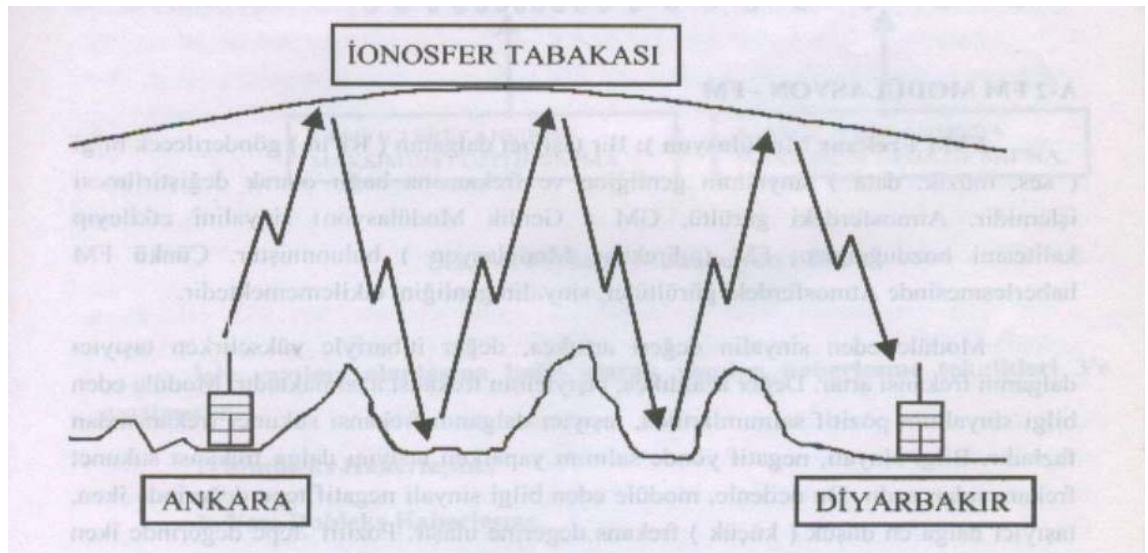
frekansları oluşur. Dolayısıyla merkez frekansta +/-3.5 kHz olmak üzere, 7 kHz' lik bir bant genişliği olur. 3.0 kHz' lik bir emniyet bandı bırakılınca, sonuçta toplam 10 kHz' lik bir bant genişliği oluşur. Bu Alt veya Üst Bantlardan birisi seçilerek, o frekans üzerinden yayın gönderilir. Tek Yan Bandın kullanılarak yayın gönderilmesi işlemine, SSB (Single Side Band) Sistemi denir[3].

1.1.6. Yüksek Frekans Yayılma Tekniđi

Güneşten gelen radyasyon, yer atmosferinin iyonosfer tabakasında serbest elektronların oluşmasına yol açar. Oluşan bu serbest elektronlarla, HF haberleşmesinde RF sinyallerinin iyonosferde kırılma ve yansımalarıyla çok uzak mesafelere yayılmasına olanak verir. İşte iyonosferin bu 3 bölgesinde HF haberleşmesinde kullanılır.

- D bölgesi 50-90 km.
- E bölgesi 90-140 km.
- F Bölgesi 140 km.' den yukarısı.

Bir SSB cihazının anteninden çıkan radyo yayını öncelikle D Bölgesine ulaşır. D Bölgesi HF dalgalarının emilerek şiddetinin azaltıldığı bölgedir. Emilme işlemi, gündüz saatlerinde gerçekleşmektedir. Özellikle güneş ışınlarının kuvvetli ve yüksek olduğu öğle saatlerinde, emilme en üst seviyeye çıkar. Geceleri ise, güneşin etkisi olmadığından, bu bölgedeki elektronlar kaybolur ve radyo dalgaları hemen hiç zayıflamaya uğramadan (emilmeden) kırılarak yansır.



Şekil.3. HF Radyo Yayınları Yansımaları

F Bölgesindeki yayılma oranı mevsimlere, gündün güne, gece ve gündüze göre, hatta yaz ve kış gecelerine göre değişir. Çünkü, bu bölgedeki elektron yoğunluğu sürekli değişmektedir.

İyonlaşmadaki yoğunluk sabah güneşin doğuşuyla başlar, öğle saatlerinde maksimum seviyeye ulaşır ve geceleri minimum seviyeye iner.

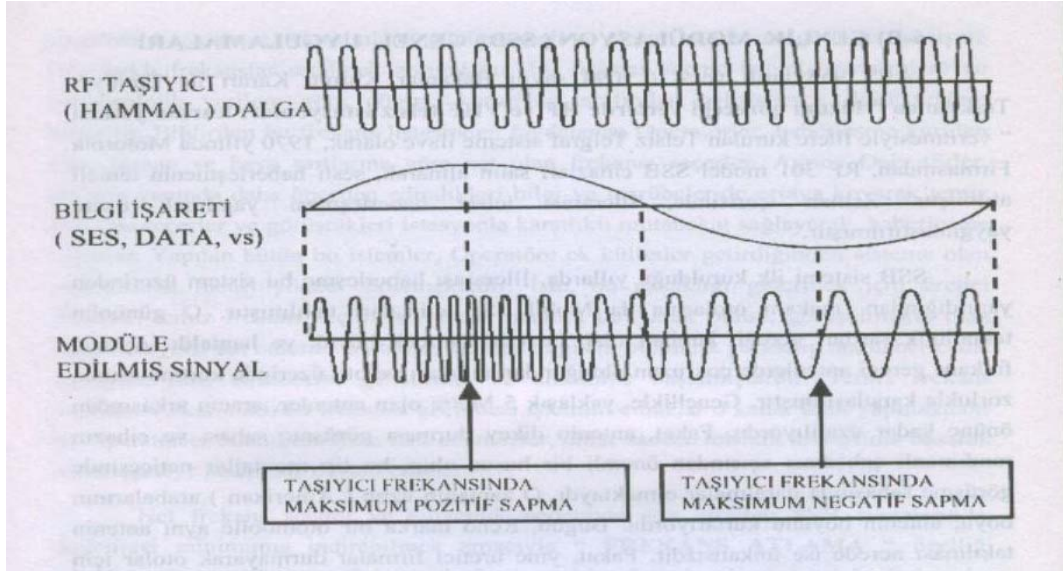
Şekilde görüldüğü gibi, cihazdan çıkan HF radyo dalgaları gökyüzüne doğru yayılır. İyonosfer tabakasının yoğunluğuna göre, çarpma neticesinde tekrar yeryüzüne döner. Yeryüzündeki engellere çarparak tekrar gökyüzüne yansır. Böylece SSB cihazlarıyla yapılan yayınların, kilometrelerce uzaklara, hatta sınır ötesi yerlere kadar ulaştırılması mümkündür[2].

1.1.7. Frekans Modülasyonu

Bir taşıyıcı dalganın (RF) gönderilecek bilgi (ses, müzik, veri) sinyalinin genliğine ve frekansına bağlı olarak değiştirilmesi işlemidir. Atmosferdeki gürültü, genlik modülasyon (GM) sinyalini etkileyip kalitesini bozduğundan, frekans modülasyonu (FM) geliştirilmiştir. Çünkü FM haberleşmesinde, atmosferdeki gürültüler, sinyal genliğini etkilememektedir.

Modüle eden (bilgi) sinyalin değeri (genliği) arttıkça, taşıyıcı dalganın frekansı artar. Değer azaldıkça, taşıyıcının frekansı azalmaktadır. Modüle eden bilgi sinyalinin pozitif salınımlarında, taşıyıcı dalganın frekansı, sükunet frekansından fazladır. Bilgi sinyali, negatif yönde salınım yaparken taşıyıcı dalga frekansı sükunet frekansından azdır. Bu nedenle, modüle eden bilgi sinyali negatif tepe değerinde iken, taşıyıcı dalga en düşük (küçük) frekans değerine ulaşır. Pozitif tepe değerinde iken ise, taşıyıcı dalga en yüksek frekans değerine ulaşır[6].

Haberleşmede, karşı tarafa modüle edilerek gönderilen yayında taşıyıcı dalganın frekansına ve gücüne bağlı olarak bir yayılama alanı vardır. Bu alanın bazen dar olması, bazen de geniş olmasına ihtiyaç olabilir.



Şekil.4. Frekans Modülasyonu

Yayıma alanlarına bağlı olarak yapılan haberleşme teknikleri 3' e ayrılır[7].

1. Simpleks Haberleşme
2. Yarı Dupleks Haberleşme
3. Dupleks Haberleşme

FM sistemle yapılan haberleşme modelleri, bu üç teknikten birisi kullanılarak gerçekleştirilir[1].

1.1.8. Frekans Modülasyonu Genel Uygulamaları

1.1.8.1. Simpleks Haberleşme

İki telsiz istasyonunun karşılıklı olarak arada herhangi bir cihaz olmadan, aynı frekansdan direkt haberleşme yapması "simpleks haberleşme" denir. Telsizlerin, buldukları kanalda, alıcı/verici frekanslarının aynı olması gereklidir. Yani

Alıcı (Rx) Frekansı= Verici (Tx) Frekansı

olmalıdır.

Genlik modülasyonu tekniğini kullanan SSB cihazları, tamamen simpleks haberleşme yaparlar. FM sisteminde de simpleks haberleşme kullanılmaktadır. FM

sisteminde RF sinyalleri, gökyüzünden değilde, yeryüzündeki doğal ve doğal olmayan engellere çarparak yansır. Düz bir arazide, haberleşme yapan iki telsiz cihazı, birbirlerinin antenlerini direkt (elektriksel olarak) gördükleri için problemsiz görüşme sağlarlar. Eğer arada bir engel var ise, telsizler engellere çarparak yansıyan RF sinyallerini alabildikleri oranda haberleşme yaparlar.

Cihazlar birbirlerinden çok uzak ve arada engeller varsa, bağlantının sağlanması mümkün değildir. Burada, telsizlerin çıkış güçleri devreye girer. Genelde El Telsizleri 2 Watt, Araç Telsizleri 20-25 Watt ve Sabit Telsizler 40-45 Watt çıkış gücündedir[6]

1.1.8.2. Yarı Dupleks Haberleşme

Simpleks haberleşmede, RF dalgaları yeryüzüne paralel giderken engellerle karşılaştığında aşamıyarak dağılmasıyla haberleşme sağlanamadığında, bölgenin en yüksek yerine konulan ve antenine gelen yayımla alıcı devreleri çalıştırarak çok kısa bir gecikmeyle verici konumuna geçerek 100 Watt gibi bir güçle yayını tekrar göndererek haberleşmeyi sağlayan sistem “Yarı Dupleks” olarak adlandırılır. Simpleks haberleşmede direkt istenilen “antenin anteni görmesi” prensibi, dolaylı olarak sağlanmış olur[6].

Tekrarlayıcı cihaz teknik olarak, normal bir telsiz cihazındaki alıcı ve verici katları ile bu iki birim arasındaki bağlantıları sağlayan elektronik devrelerden oluşur. Telsizde, alıcı devresi sürekli çalışır durumdadır. Sadece Bas/Konuş Mandalına basıldığında, alıcı devrelerine giden besleme kesilerek verici devreleri çalıştırılır. Tekrarlayıcıda ise, alıcı ve verici devreleri aynı anda çalışabilir ve çalışmalıdır[2].

Herhangi bir telsiz, Bas/Konuş mandalına basılıp telsizini vericiye alarak bilgi gönderdiğinde, yayın tekrarlayıcının antenine gelir ve alıcı devresi çalışmaya başlar.

Alıcı kısmın çalıştığına dair alınan bilgi kullanılarak, elektronik devre elemanlarının tetiklenmesiyle verici kısım devreye konulur ve alınan yayın aynı anda hiçbir değişikliğe uğratılmadan sadece güçlendirilmiş bir şekilde tekrarlayıcı anteninden

atmosfere yayılır. Tekrarlayıcının, alıcı ve verici kısımlarının çalışma zamanları arasında anlık bir gecikme olsada, bu gecikme hissedilmez.

Tekrarlayıcı bir haberleşmede, 2 Watt'lık bir el telsizinden gelen yayın, tekrarlayıcı tarafından hiçbir değişikliğe uğratılmadan 100 Watt' a yükseltilir ve antenden yayın yapılır. Aynı anda yayın yapıldığı için, simpleks haberleşmede olduğu gibi ikinci bir tekrara gerek kalmadan, yayını yapan istasyonun kendi sesinden bütün istasyonlara duyurulması ve bütün istasyonlarında, kanalda yapılan her haberleşmeyi anında ve eksiksiz duyması imkanı elde edilmiş olur[6].

1.1.8.3. Dupleks Haberleşme

Yarı dupleks haberleşmede, istasyonlar birbirlerini bekleyerek sırayla konuşmakta, yani aynı anda hem konuşma hemde dinleme yapamamaktadırlar. İşte aynı anda, hem dinleme hende konuşma yapılabilen haberleşme tekniğine “Dupleks Haberleşme” denir. Telefonla konuşulurken yapılan haberleşme şekli tamamen dupleks olup, telsiz haberleşmesinde yalnızca tekrarlayıcılar dupleks haberleşme yapabilirler.

1.1.8.4. Genlik ve Frekans Modülasyonlu Sistemlerinin Karşılaştırılması

- 1) FM telsiz haberleşmesinde, özellikle VHF ve UHF bantlarındaki frekanslar kullanılır. Haberleşmede kullanılan frekansla yayılma alanı arasında ters bir orantı vardır. Frekans büyüdükçe yayılma alanı daralır (azalır). Tersinde ise, frekans küçüldükçe yayılma alanı genişler.

Bu orantıya göre, VHF bandı frekansları HF bandı frekanslarına nazaran daha yüksek olduğundan yayılma alanı da HF bandına nazaran daha dardır. UHF bandı frekanslarının yayılma alanı çok daha dardır. Buna göre HF frekansları kullanılarak çok uzak mesafelerle haberleşme yapılabilmesi imkanı vardır.

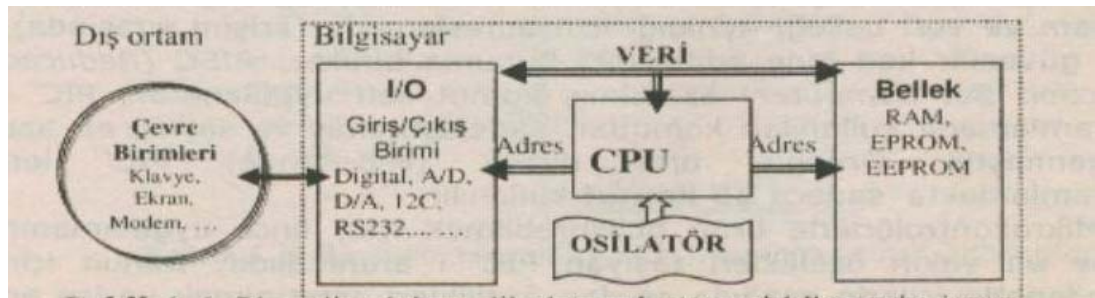
- 2) Yine aynı şekilde, frekansla anten boyu arasında da ters bir orantı vardır. Frekans yükseldikçe anten boyu düşer, düştükçe de anten boyu yükselir.

- 3) FM haberleşmesinde kullanılan frekanslar, SSB sistemindeki frekanslar gibi iyonosfer tabakasını kullanmayıp delip geçtiğinden, gökyüzü dalgalarından faydalanamaz, sadece yer dalgalarından faydalanarak haberleşme yaparlar. Yer dalgaları da yeryüzündeki engelleri aşamadığından gök dalgalarına nazaran çabuk söner ve yayılma alanları dardır.
- 4) SSB haberleşmesindeki Genlik Modülasyonu tekniğinde, taşıyıcı dalga frekansına ses veya veri yüklenerek modülasyon gerçekleştirilir. Taşıyıcı dalganın genliği sabit olmadığı için, sesin tiz ve bas noktaları kırılır. Fakat, FM'de taşıyıcının genliği sabit olduğu için, modüle edilen bilginin genliğine bağlı olarak taşıyıcının genliği değişmediğinden, sesin tepe noktaları olan bas ve tiz tonlar hiç kesintiye uğramadan modüle edilerek antenden atmosfere yayılır. Böylece modüle edilen bilgi (ses, veri vb.) herhangi bir bozulmaya uğramadan karşı istasyona gönderilir[7].

1.2. Mikroişlemciler

1.2.1. Mikroişlemci Tanımı

Mikroişlemci, verilen komutlarla aritmetik ve mantıksal işlemleri (topla, çıkar, AND, OR, uygula, yükle, tersini al, vb gibi) uygulayan birimdir. Aşağıda mikroişlemcinin temel bileşenleri gösterilmiştir.



Şekil.5. Mikroişlemci Blok Şeması

Mikrodenetleyicilerin mikroişlemcilerden temel farkı; CPU, RAM ve I/O ünitelerinin tek bir yonga içinde üretilmesidir. Bu şekilde üretilmiş yongalarda, CPU, RAM ve I/O üniteleri bir araya getirildiği için, bunlar arasında ayrıca veri iletişim hattı ve adresleme hattı kurulması gerekmez. Üç birimin tek bir yongada toplanması, maliyeti de düşürmüştür[8].

Mikrochip firmasının ürettiği yongalar, PIC (Peripheral Interface Controller) olarak adlandırılır ve bunlar Harvard Mimarisine göre tasarlanmıştır. Harvard mimarisinin daha önceleri uygulana von Neumann mimarisinden farkı; program ve veri belleğinin ayrılmasıdır[5].

Program ve veri belleklerinin ayrılmasıyla birlikte, adres/program yolu ve veri yolu ayrılmıştır. Bu yapılanma ile, aynı anda veri ve program belleğine erişim olanağı kazanıldığından, iletişim iki kat artmıştır. Bununla birlikte, RISC (Reduced Instruction Set Computer) denen azaltılmış komut seti uygulanarak, PIC'i programlamada kullanılan komutlar sadeleştirilmiş ve sayıcada en aza indirilmiştir[5].

1.2.2. Temel Yapısı

Bir mikrodenetleyici; merkezi işlemci birimi, program belleği, veri belleği ve giriş/çıkış ünitelerinin bir araya gelmesiyle oluşur.

1.2.3. Mikroişlemci Birimi

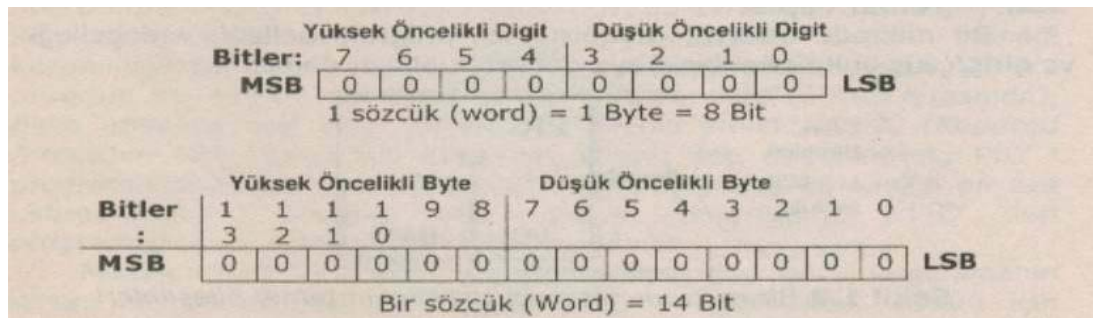
Mikroişlemci birimi (microprocessor unit); bilgisayarlarda Merkezi İşlem Birimi (Central Processor Unit, CPU) olarak adlandırılır. Merkezi İşlem Birimi, mikrobilgisayardaki bütün işlemleri başlatır, kontrol eder ve sonlandırır. Mikrobilgisayarın belleğinden okuduğu bilgileri, geçici olarak depolar. Dahili yazmaçlarına okuduğu bilgiler; yapacağı işlemleri içeren komutlar veya bu işlemleri gerçekleştirirken kullanacağı veriler olabilir. Mikroişlemciye, işlenecek bilginin bulunduğu yeri gösteren adresler (pointer) de yazmaçlarda tutulur[5].

Veriler, “bit”ler ; yani 0 ya da 1 değeri alabilen, ikili sayı dizileri biçimindedir. Kendi iç yazmaçlarında, bir seferde depolanan maksimum bit sayısı, o mikroişlemcinin sözcük (word) uzunluğudur. Mikrochip firmasının ürettiği orta-düzy PIC’ lerde program belleğine 14 bit, veri belleğine ise 8 bit uzunluğunda sözcük (word) lerle erişilir (Harvard Mimarisi). Bunlar genel olarak veri belleğine erişimde kullanılan sözcük boyuyla; 8 bit’ lik mikrodenetleyiciler olarak sınıflandırılır[5].

Sözcük (word) bilgisayar işlemcisinin bir kerede okuyabildiği bit sayısıdır. Bellekteki bir birimlik bit grubudur. Sözcüklerin içinde, bilgisayardaki her tür veri, komutu taşıyan ikili sayılar, adresler, sembollerin ASCII kod karşılıklarından biri, karakter ve sabitler bulunabilir[5].

1.2.4. Sözcük Uzunluğu

Bir mikroişlemcinin tipik özelliklerinden biri de, işlediği sözcüklerin boyudur. Von Neumann mimarisinde, eğer işlemci 8 bitlik ise; o işlemcide sayılar, adresler, komutlar, işlenecek veya işlenmiş tüm veriler 8 bitlik ikili sayılarla tanımlanır. PIC’lerde baz olan Harvard mimarisinde ise veri belleğine 8 bitlik sözcükle, program belleğine ise 14 bitlik sözcükle erişilir. Kısacası, komutları oluşturan ikili sayılar 14 bitlik sözcüklerin içerisinde; yazmaçları, program değişkenlerini, ASCII karakterleri, sabit değerleri oluşturan veriler ise 8 bitlik sözcüklerin içerisinde[5].



Şekil.6.Yazmaç

1.2.5. Bayt

Bir bayt, 8 bitten oluşur. 16 bitlik bir sözcük, iki bayttan oluşur. Düşük öncelikli bit (Least Significant Bit) en sağda ve yüksek öncelikli bit (Most Significant Bit) de en soldadır[8].

1.2.6. Karakter

Bir bayt, iki dördü bit grubuna ayrılabilir. Bunların her birine “karakter” denir. Bir karakter kendi başına bir erişim birimi değildir.

Mikrodenetleyici, kendisine verilen emirleri (program) işler ve bu emirlerin doğru çalıştırılmasını CPU sağlar. Komutların program belleğinden getirilmesi, deşifre edilmesi, işlem göreceği verinin veya verilerin bellekten alınması, işlemin uygulanması ve üretilen sonuçların veri belleğine yazılması işlerinin hepsinden CPU sorumludur. Program komutları başlıca üç biçime ayrılır. Bunlar; bit kaynaklı, bayt kaynaklı, sabit ve kontrol işlemleri ile call-goto işlemlerini kapsar. CPU bunları doğru işlemek ve sonuçlandırmak için, veri iletişim yolu, program/adres iletişim yolu ve yığını kullanabilir[5].

CPU’ nun bu işlerde kullandığı üç ana bölüm vardır. Bunlar; çeşitli görevlerle donatılmış yazmaçlar bölümü, aritmetik mantık birimi (Aritmatic Logic Unit, ALU) ve zamanlama-denetim bölümüdür[5].

1.2.7. Yazmaçlar

Mikroişlemciye gelen veri ya da adres şeklindeki sözcükler, mikroişlemci içerisindeki yazmaçlarda tutulur. Yazmaçlar, kullanım amaçlarına göre birkaç çeşittir ve bunlardan üçü her PIC’ de bulunur. Bunlar; Çalışma Yazmacı (Working Register), W, Program Sayacı (Program Counter) ve İşlemci Durum Yazmacı (Processor Status Register), PS[5].

1.2.7.1. Çalışma Yazmacı

Genel amaçlı bir yazmaçtır. Tüm aritmetik ve mantık işlemlerinde, işlenenlerin ve bazı mikroişlemcilerde de hem işlenen hemde işlem sonuçlarının tutulduğu bir yazmaçtır. Verilerle ilgili kaydırma, döndürme, eksiltme, arttırma, karşılaştırma ve tersini alma işlemlerinin gerçekleştirilmesi ile bu işlemlerin sonuçlarının tutulmasında kullanılır[5]

1.2.7.2. Program Sayacı

Mikroişlemci (CPU) tarafından yürütülecek komutun, program belleğindeki adresini tutar. Program sayacı yazmacının içinde, bulunulan yeri gösteren adres olduğu için, kendisi bir göstergedir (pointer). Program sayacında, ilk komut çalıştırıldıktan sonra, ikinci komutun bulunduğu adres oluşur. Çünkü, bir komut çalıştırıldığında, program sayacındaki hexadesimal sayıya bir eklenir. Böylece program sayacı, sürekli bir sonra çalıştırılacak komutun adresini gösterir. Eğer komut bir sözcükten daha uzun ise, komutun ikinci ve varsa diğer parçaları da program sayacına her seferinde bir eklenerek adreslenir.

Mikroişlemciye bir altprogram komutu verilmişse ve bu komut çalıştırılmışsa, bu durumda program sayacına bir eklenmez, yerine altprogramın adresi yüklenir. İşte tam bu noktada, “Altprogramdaki komut grubu işlendikten sonra, ana kesimdeki çağrıldığı noktaya nasıl geri döner?” sorusu akla gelir. Çünkü her altprogram çağırma, çağrılan altprogramın kapsamındaki komut grubunun çalıştırılması gerekir, dolayısıyla bu komut grubuna sapılmıştır. Program sayacı altprogram bittikten sonra da sıradan artmaya devam ederse, programda altyordamın çağrıldığı adresi, birdaha bulamayacağı açıktır. Bu sorunun çözümü şöyledir; Program sayacı, altprogram tamamlandığında, altprogramın çalıştırılmasından bir önceki adrese geri döner. Bunun için altprogramın çalıştırılmasından bir önceki adres, önce yığın (stack) denilen bir dizinin en üstüne konur. Bu işlemden sonra, program sayacı altyordamının içindeki ilk komutun adresini alır ve altyordamın her komutunda, birer arttırmayı sürdürür. Altyordamdan dönüş komutu RETURN’e geldiğinde, yığının en üstüne konan adres Program sayacına geri yüklenir. Böylece programda, altyordamın çağrıldığı noktaya geri dönmüş olur. Program sayacı bir arttırıldığında, işleyeceği

bir sonraki komutun adresini gösterir. Böylece program komutları işlenmeye devam eder[5]

PIC16F8X ve 16F877 ailelerinde yığın derinliği sekizdir. Bunlarda, iç içe en fazla sekiz alt program kullanılabilir. Sekizden fazla altprogram kullanıldığında ise, mikrodenetleyici “yığın taşması” (stack overflow) hatasına düşer.

PIC16F8X ve 16F877 ailelerinde yazmaçlara ya da veri program belleğin, doğrudan veya dolaylı erişilebilir[5].

1.2.7.3.. İşlemci Durum Yazmacı

Kısa adı STATUS, PS, olan 8 bitlik bu yazmaç içerisinde mikroişlemcinin o andaki durumunu bildiren uyarı bitleri bulundurulur. Bu yazmaç sayesinde, yapılan işlemin sonucu hakkında bilgi alınabilir. Mesela, aritmetik işlemlerde; elde olup olmadığı, sonucun sıfır olup olmadığı, status yazmacının ilgili bitlerine bakılarak öğrenilebilir. Satatus yazmacında, dolaylı adresleme ve doğrudan adresleme bilgileride bulunur. Program, status yazmacından öğrenilen bilgilere göre yönlendirilir[5].

PIC’ lerde bunların dışında dolaylı erişim için INDF ve FSR; kesmeler için INTCON; zamanlama için TMR=, TMR1 ve TMR2, girdi çıktılar için TRISA, TRISB, .., TRISE ile PORTA, PORTB, ..., PORTE gibi pekçok yazmaç mevcuttur.

1.2.8. Aritmetik Mantık Birimi

Mikroişlemcinin diğer önemli birimi olup, komut sözcüğüne (instruction word) göre aritmetik ve mantık işlemleri yapar. ALU içerisinde toplama (ADD), çıkarma (SUB), bir yazmacın sağ ve sol hanelerini yer değiştirme (SWAP), kaydırma (SHIFT) ve döndürme (ROTATE), .. gibi işlemler yanında ayrıca AND, OR, XOR mantıksal işlemleri yapan birimlerde vardır. ALU, veri iletişim hattı (veribarasıbus) aracılığıyla verileri alır, komuta göre işler ve ilgili birimleri uyararak sonucu çalışma yazmacı (w)’ na veya komutta belirtilen hedef yazmaca yükler[5].

1.2.9. Zamanlama ve Denetim Bölümü

Mikroişlemcilerin kendilerine verilen komutları işleyebilmeleri için, “klok” denilen bir çalışma sinyali gereklidir. Bu sinyalde, mikroişlemci içerisinde bulunan osilatör devresine, dışarıdan bağlanan bir kristal üretir.

PIC’ in sinyal giriş uçlarına farklı osilatörlerde üretilen sinyaller uygulanabilir ki; bunlar

- RC (Direnç-Kapasitör)
- XT (Kristal veya seramik Resonatör, XTAL)
- HS (Yüksek Hızlı Kristal veya seramik Resonatör, High Speed)
- LP (Düşük Frekanslı Kristal, Low Power)

olarak sayılabilir[5].

1.2.10. Program Belleği

Mikrodenetleyicinin uygulaması için verilen komutlardan oluşan programın yerleştiği alandır. Mikrodenetleyici, uygulayacağı bütün işlemleri ve bu işlemlerin sırasını program belleğinin ilgili adreslerine bakarak öğrenir ve ilgili adresler ise program sayacında tutulur. PIC16F87X ailesinde üç bellek bloku vardır. Bunlar; Program Belleği, Veri Belleği (RAM) ve EEPROM Veri Belleği’ dir[5]. Program belleği, 16F84’ te 1 Kbyte ve 16F877’ de 8 Kbyte’ dir.

1.2.11. Veri Belleği

Programın çalışması için veri belleğindeki yazmaçlar kullanılır. Dosya yazmaçları, 8 bit uzunluğunda özel veri bellek alanlarıdır. Yani bunların adresleri önceden belirlenmiştir. 16F84’ ün veri alnları ikiye ayrılır ve her birine “bank” denir. Bazı özel yazmaçlar her iki bankta da mevcuttur ve Bank0’ da iken, yalnız Bank1’ de bulunan özel amaçlı bir yazmaç kullanılamaz, çünkü önce Bank1’ e geçme işlemi uygulanmalıdır. 16F877’ nin 4 adet veri belleği vardır[6].

1.2.12. Giriş/Çıkış Birimi

Giriş birimi mikroişlemci dışındaki devreler ve sistemlerden gelen işaretleri (sinyaller), mikroişlemciye aktaran bir tümleşik devre (Integrated Circuit, IC)'dir. Benzer şekilde, çıkış birimi de yonganın çıkış sinyallerini, mikro işlemci dışındaki devrelere aktaran bir tümleşik devredir. Uygulamada iki ayrı tümleşik devre, aynı yonga (chip) içinde, her iki tümleşik devrenin denetlenmesi amacıyla bir kontrol devresi eklenerek üretilir[5].

Mikrodenetleyicinin dış dünyayla iletişimi, giriş/çıkış (I/O) portlarıyla kurulur. PIC16F877' de giriş/çıkış portları A'dan E'ye harflerle belirtilir ve bunlara bağlı 33 pin yoluyla dış ortam ile bilgi alışverişinde bulunulur. 16F84' te ise toplam 18 pin A ve B portlarında bulunur.

1.3. Paralel Port

Port, mikroişlemcinin, ya da CPU'nun diğer bilgisayarlarla veri alışverişinde kullandığı bir sinyal hatları kümesidir. Paralel port hemen her kişisel bilgisayarda, yani IBM uyumlu kişisel bilgisayarlarda mevcuttur.

Orijinal kişisel bilgisayarların paralel portunda sekiz çıkış, beş giriş ve dört iki yönlü hat mevcuttur[1].

1.3.1. Port Türleri

Kişisel bilgisayar tasarımındaki gelişime bağlı olarak, paralel portun gelişmiş versiyonları da devreye girmiştir. Yeni port türleri orjinal tasarımla uyumlu olmakla birlikte, başta hız artışı olmak üzere yeni yetenekler kazanmışlardır[1].

1.3.1.1. SPP

Orjinal IBM kişisel bilgisayardaki paralel port ve bu anlamda orjinal tasarımla benzeşen herhangi bir port, standart paralel port, ya da kısaca SPP olarak adlandırılır. SPP, Cantronics arabiriminde kullanılan protokolün benzerini kullanarak, çevre birimine bir defada 8-bit transfer yapabilir. SPP' nin bayt çaplı giriş portu yoktur,

ancak kişisel bilgisayardan çevrim birimine transferde, bir kerede her bir bayt'ı (4-bit) transfer eden bir Nibble modunu kullanabilmektedir[1].

1.3.1.2. PS/2

Paralel portta ilk gelişme, IBM' in PS/2 modelinde saunulan ve çevrebirimden kişisel bilgisayara bir defada 8-bit transferi mümkün kılan iki yönlü veri portu olmuştur. PS/2 potların çevrebiriminden bir kişisel bilgisayara veri transferinde kullandığı "8-bit veri transfer protokolü" ne "bayt modu" denilmektedir[1].

1.3.1.3. EPP

Burada da, PS/2 portunda olduğu gibi, veri hatları iki yönlüdür. Bir EPP, bir SPP ya da PS/2 porttaki dört çevrime kıyasla, el sıkışma da dahil, yaklaşık bir mikro saniyede, bir byte veriyi okuyabilir veya yazabilir. EPP veri yönleri çok çabuk çevrilebildiğinden, iki yönlü veri transfer eden cihazlarla kullanımda son derece etkilidir. Ayrıca bir EPP, SPP yerine kullanılabilirken, bazende PS/2 pot yerine de kullanılabilir[1].

1.3.1.4. ECP

ECP'de EPP gibi iki yönlüdür ve çok yüksek hızlarda veri transfer edebilirler. Bir ECP, hem SPP hem PS/2 ve hemde EPP yerine kullanılabilir[1].

1.3.2. Adresleme

Standart paralel port, genellikle aşağıdakilerin birine karşılık gelen üç bitişik adresi kullanır.

Tablo.1. Paralel Port Yazmaç Adresleri

Veri Yazmacı	Durum Yazmacı	Kontrol Yazmacı
h(3BC)	h(3BD)	h(3BE)
h(378)	h(379)	h(37A)
h(278)	h(279)	h(27A)

Aralıktaki ilk adrese, baz adresi, Veri Yazmacı, ikinci adrese Durum Yazmacı ve üçüncü adrese de Kontrol Yazmacı denilmektedir.

Tablo.2. Veri Yazmacı Pin Adresleri

Veri YAZMACI (BAZ ADRESS)					
BIT	PIN:D-SUB	SINYAL ADI	KAYNAK	KONNEKTORDE TERSLENME	PIN: CENTRONICS
0	2	D0	PC	H	2
1	3	D1	PC	H	3
2	4	D2	PC	H	4
3	5	D3	PC	H	5
4	6	D4	PC	H	6
5	7	D5	PC	H	7
6	8	D6	PC	H	8
7	9	D7	PC	H	9

İlk kişisel bilgisayarlarda paralel portun baz adresi h(3BC) idi. Daha yeni sistemlerde bu adres genellikle h(378) olmuştur. Bununla birlikte her üç adresinde paralel portlar için tahsis edilmiş olması nedeniyle, port donanımlarının imkanları ölçüsünde, herhangi bir port bu adreslerden herhangi birinde yapılandırılabilir.

Çok sık olarak, DOS ve WINDOWS'ta sayısal bir sıralamayla birinci port LPT1, ikinci, LPT2 ve üçüncü, LPT3 olarak adlandırılır. Bilgisayar ilk açıldığında LPT1 genellikle h(378) adresindedir. Ancak pekala üç adresten birinde de olabilir. Bu adresler çeşitli yapılandırma teknikleriyle değiştirilebilirler[1].

1.3.3. Portlara Erişim

Windows, DOS ve Visual Basic (VB), paralel portlara yazma ve okumada çeşitli yollar sunarlar. Burada en dolaysız yol port yazmaçlarına yazmak ya da okumaktır. Hemen tüm programlama dillerinde bu özellik bulunmak ta veya en azından programcının eklemesine izin vermektedir.

Tablo.3. Durum Yazmacı Pin Adresleri

DURUM YAZMACI (BAZ ADRES+1)					
BIT	PIN:D-SUB	SINYAL ADI	KAYNAK	KONNEKTORDE TERSLENME	PIN: CENTRONICS
3	15	nFault	Çevresel	H	32
4	13	Select	Çevresel	H	13
5	12	PaperEnd	Çevresel	H	12
6	10	nAck	Çevresel	H	10
7	11	Busy	Çevresel	E	11

Yukarıda ek olacak bitler konnektörde mevcut değildir. 0 zamanın bittiğini gösterir, 1 ve 2 kullanılmaz.

Tablo.4. Kontrol Yazmacı Pin Adresleri

KONTROL YAZMACI (BAZ ADRES+2)					
BIT	PIN:D-SUB	SINYAL ADI	KAYNAK	KONNEKTORDE TERSLENME	PIN: CENTRONICS
0	1	nStrobe	PC	E	1
1	14	nAutoLF	PC	E	14
2	16	nInit	PC	H	31
3	17	nSelectIn	PC	E	36

Konnektörde mevcut olmayan bitler; 4 kesmeyi mümkün kılar, 5 iki yönlü veri portları için yön kontrolüdür ki; 0 için çıkış mümkün, 1 için çıkış mümkün değil yani veri portu harici okuma modundadır. 6, 7 kullanılmazlar.

Sekiz veri biti için D0-D7, beş durum biti için S3-S7 ve dört kontrol biti için C0-C3 notasyon olarak kullanılır ki; burada harfler port yazmacını, sayılırsa sinyalin yazmaçtaki bit pozisyonlarını gösterir[1].

1.3.4. Veri Yazmacı

Veri yazmacı (D0-D7) veri çıkışlarına yazılan baytları tutar. İki yönlü veri portlarında port, giriş için yapılandırıldığında, veri yazmacı konnektörün veri bacaklarında okuduğu baytı tutar[1].

1.3.5. Durum Yazmacı

Durum yazmacı S3-S7 arasındaki beş girişin mantık durumlarını tutar. S0-S2 aralığındaki bitler konnektörde görünmez. Status yazmacı salt okunurdur ve girişlerin çekme (pull up) dirençleri vardır[1].

1.3.6. Kontrol Yazmacı

Kontrol yazmacı C0-C3 aralığındaki bitlerin durumlarını tutar. Normalde bu bitler çıkışlarda kullanılır[1].

1.3.7. Dolaysız Port I/O

Paralel port sinyalleri üzerindeki en kapsamlı denetim, port yazmaçlarını doğrudan okuyarak veya bunlara yazarak olur. Doğrudan porta yazarken, port yazmacı ve yazılacak veri belirlenir, CPU'ya, veriyi istenilen porta yazması için talimat verilir. Okurken de yine yazmaç ve verinin saklandığı yer belirtildikten sonra CPU'ya istenilen yerdeki veriyi okuması söylenir[1].

DOS, Windows 3.1 ve Windows 95 doğrudan yazma ve okumayı mümkün kılarken, Windows NT ortamında portlara doğrudan erişim yazılım tarafından korunmaktadır[1].

Sistemin kişisel bilgisayar tarafında kullandığımız programlama dili olan Visual Basic'de, IN ve OUT deyimleri için port erişiminde, herhangi bir Windows programının erişebileceği kodları içeren bir DLL (Dynamic Linked Library), Windows kütüğüne veya varsayılan sistem kütüğüne kopyalanmalıdır ki; bu kütükler bir DLL yüklendiğinde Windows' un otomatik olarak baktığı yerleşimlerdir. Bu DLL bizim uygulamamızda, 32-Bit VB4 programlarına yönelik olarak hazırlanmış "inpout32.dll" dir.

DLL (Dynamic Linked Library) Windows uygulamalarının çağırabileceği bir prosedürler topluluğudur. Bir uygulama çalıştırıldığında, program kodunda bildirilen DLL'e bağlanır ve uygun DLL'leri belleğe yükler. Aynı DLL'e birden fazla uygulama erişebilir. Uygulamalar, DLL' leri rutinleri ve fonksiyonları benzer şekilde çağırırlar.

Bu uygulamada, bizim için, IN ve OUT deyimlerini oluřturmanın en kolay yolu, yukarıda anlatıldıđı gibi, daha önceden hazırlanmıř olan “inpout32.dll”i kullanmak olmuřtur.

2. ÖNCEKİ ÇALIŞMALAR

Bu projenin, özellikle GSM ve telefon hatları üzerinden veri aktarımı şeklinde, bir çok firma tarafından, genellikle aracı modüller veya hazır veri toplama ve haberleşme kartları ve blok haberleşme yazılımlarıyla yapılmış benzer uygulamaları mevcuttur.

Zaten uygulamaya alınmış bu tip projeler genellikle firma AR_GE'lerinde geliştirilip ticari yönü ön plana çıktığından, konu üzerinde mühendislik çalışmasının kaynaklarında bulmak olanağı pek bulunmamaktadır. Çünkü bu geliştirmeyi yapan kurum veya kişiler, proje detaylarını ellerinde tutmayı tercih edecektir.

Telemetri, endüstride izleme ve kontrol için kullanılır. Bu her iki fonksiyon endüstriyel telemetri sisteminde birlikte mevcut olduğunda ise -modern uygulamalardaki gibi- SCADA (Supervisory Control And Data Acquisition) adı böyle bir sistemi tarif etmek için tercih edilir

Endüstriyel telemetrinin tarihi 200 yıl öncesine ve endüstri devrimini içeren olaylara uzanır. 2.Dünya Savaşı ve bilgisayarın gelişiminin endüstriyel telemetrideki teknoloji kullanımına etkileri, çok büyük olmuştur.

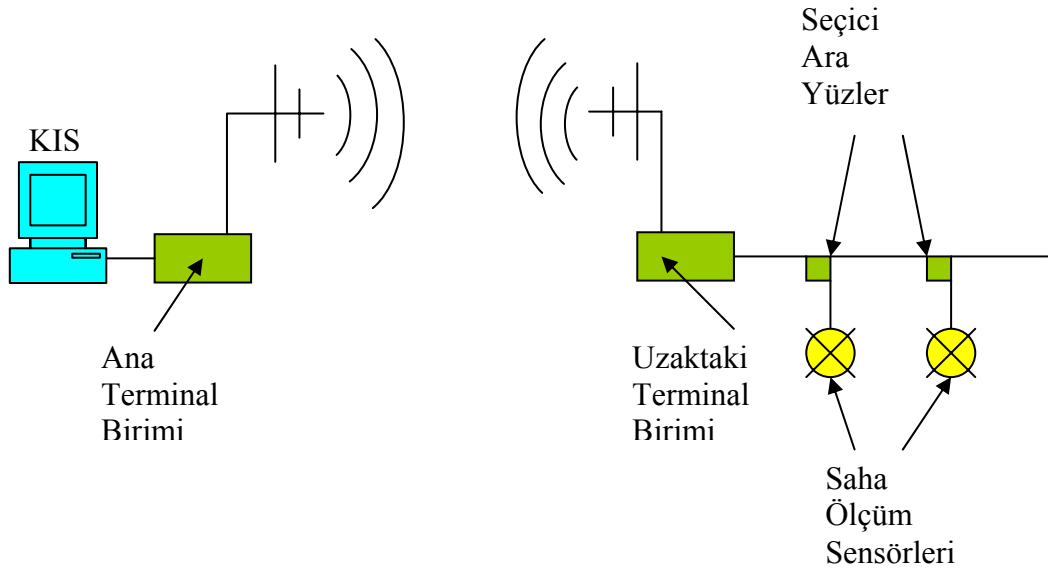
Modern üretim ve endüstriyel proseslerde, madencilik endüstrisinde, kamu ve özel tesislerinde, uzak mesafelerdeki ayrık sistemleri ve ekipmanları birbirine bağlamak için sıklıkla telemetriye ihtiyaç vardır. Bu mesafeler birkaç metreden yüzlerce kilometreye değişir.

Veri toplamanın ilk zamanlarında, üretim ve sistemlerini kontrolde röle (On/Off) mantığı kullanılmıştır. CPU ve diğer elektronik cihazların gelişimiyle birlikte üreticiler sayısal elektroniği röle mantık ekipmanının içinde beraber çalıştırmışlardır. PLC ve programlanabilir mantık kontrolöleri hala, endüstride en yaygın kullanılan kontrol sistemlerinden birisidir. Büyüyen fabrikalardaki daha çok cihazı kontrol etmek ve izlemek için PLC'ler de dağıtılmış ve sistem daha da akıllanmış ve güçlenmiştir.

3. MATERYAL VE METOT

3.1. Telemetri Sistemi Haberleşme Elemanları

Yukarıda genel hatlarıyla bahsettiğimiz ve çalışma ilkelerini verdiğimiz haberleşme teknolojisinin ve telsiz haberleşmesinin, tamamen kişisel bilgisayar kontrollü uzaktan veri toplama ve kontrol amacıyla kullanılabilirliğinin en güzel örneği, petrol sahalarındaki uygulamasında verilmektedir.



Şekil.7. Ana ve Uzaktaki Terminal Birimleri Blok Şeması

Geliştirilmiş olan proje, aşağıda sıralı 4 temel öge üzerinden çalışmaktadır.

1. Ana Terminal Birimi, (Main Terminal Unit, MTU)
2. Uzaktaki Terminal Birimi (Remote Terminal Unit, RTU)
3. Seçici Arayüz
4. Yazılım

3.1.1. Merkezi Bilgisayar

Ana ünitenin merkezinde yeralan 386 tabanlı bir bilgisayar ve yine bu telemetri projesi için Visual Basic programlama dilinde geliştirilmiş ve Ek'te de yazılım dökümü verilmiş bir program sayesinde, tüm sistem bu merkezi bilgisayar tarafından yönetilir.



Resim.2. Merkezi Bilgisayar Ekranından Kuyu Akış Değerleri

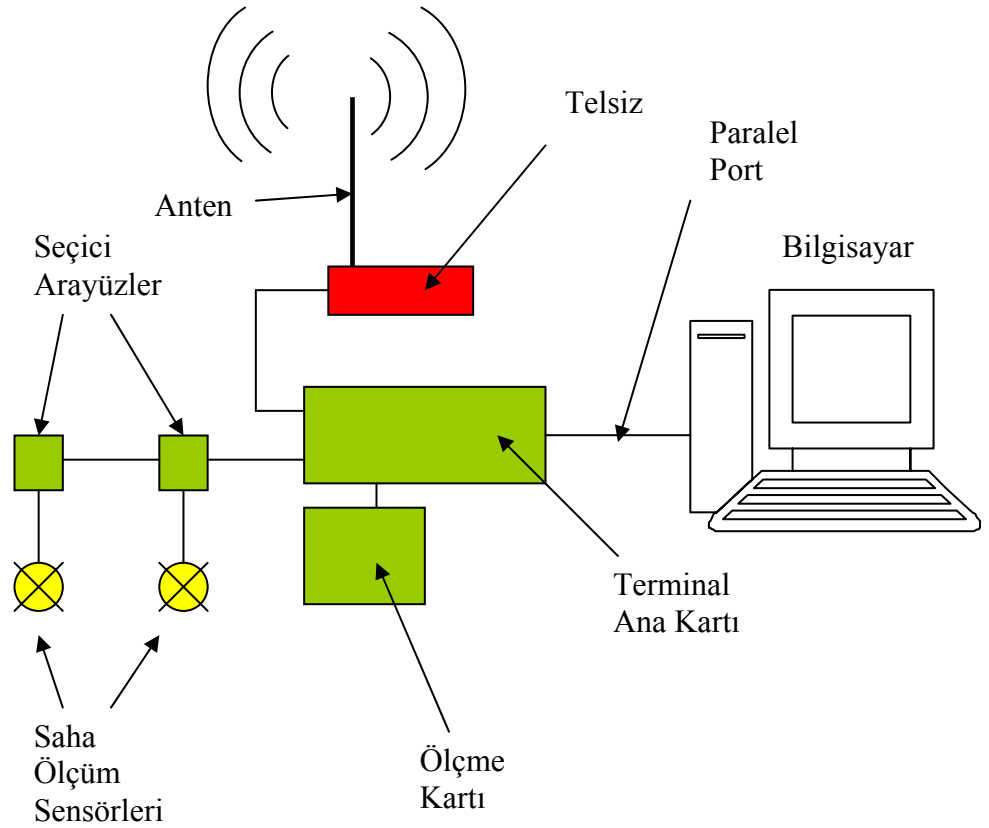
3.1.2. Terminal Birimi

Ana Terminal Birimi olarak adlandırılan MTU, bir terminal birimi ve bilgi alış verişi için paralel portundan bağlandığı bir bilgisayardır. Ana merkezde bulunan operatörün MTU ile iletişimi insan-makine arayüz yazılımı ile gerçekleşmektedir.

Ana birim teorik olarak tüm sistemi yöneten bir orkestra şefi konumundadır. Diğer tüm uzaktaki üniteler onun göndereceği emirleri mutlaka belleklerine almak zorundadırlar.

Terminal Birimi;

- Mikrodenetleyici tabanlı geliştirilmiş bir terminal ana kartı
- Analog ve sayısal elektronik ölçümlerin yapılmasını sağlayan bir Ölçme Kartı,
- Terminal ana kartının modem kısmının bağlı olduğu bir telsiz cihazı ve anteni,
- Terminal Biriminin ölçme yapabilmesi için okunacak sensörü seçen Seçici Arayüzler Kartı,
- Tüm bu sisteme enerji sağlayan güç kaynağı ve gücün kesintisiz sağlanmasına olanak veren bir bataryadan oluşmaktadır.



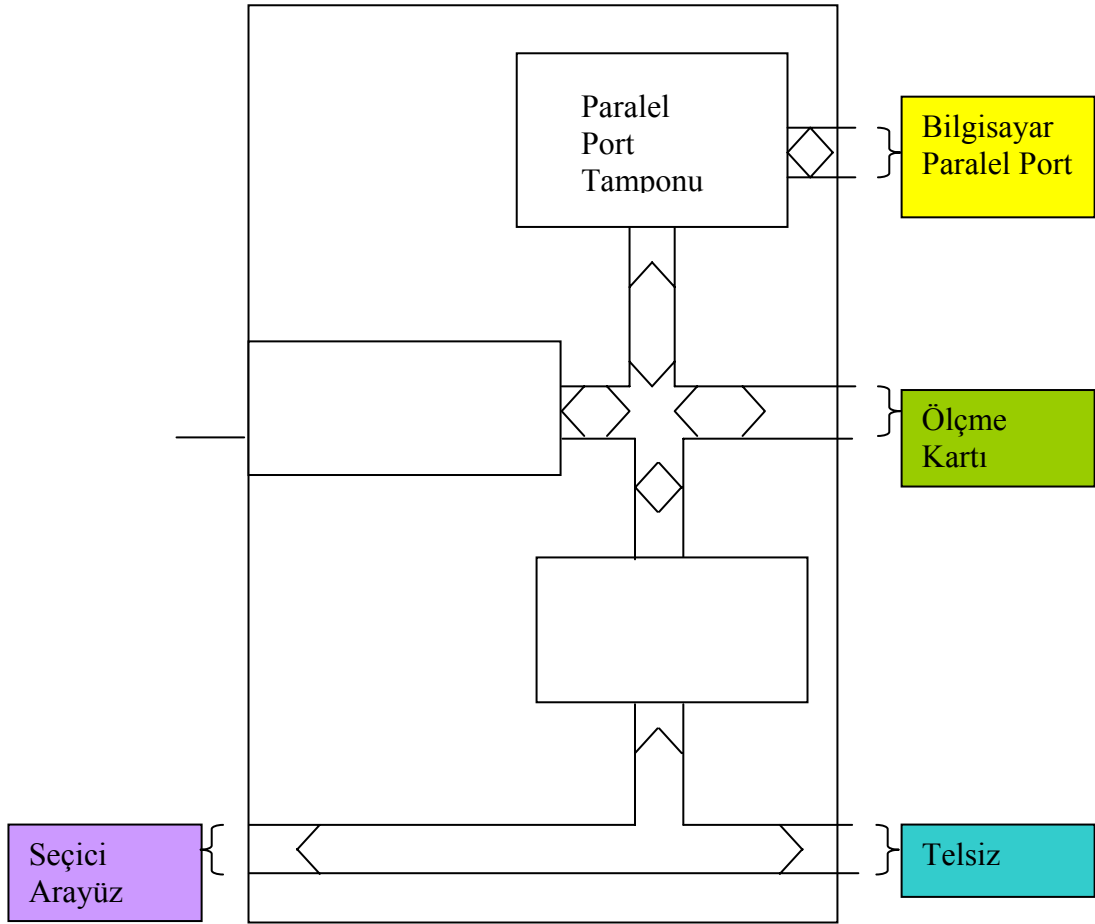
Şekil.8. Terminal Birimi Blok Şeması

Uzaktaki Birim, üzerindeki elektronik kartlar ve telsiz cihazıyla Ana Birimle birbirinin aynısıdır. Yani istenildiğinde, üzerindeki mikrodenetleyicinin yazılımı değiştirilerek, Uzaktaki Terminal Birimi, Ana Terminal Birimi olarak kolayca çalıştırılabilir şekilde tasarlanmıştır.

Şimdi bir Terminal Birimini oluşturan elemanları tanıyalım...

3.1.2.1. Terminal Ana Kartı

Terminal ana kartı, tüm birimlerde bulunan, birimin kalbini oluşturan ve hemen tüm işlemlerin üzerinde yapıldığı elektronik karttır. Aşağıda blok diyagramda görüldüğü gibi, elektronik kartı 3 kısımda değerlendirebiliriz.



Şekil.9. Terminal Ana Kartı Blok Şeması

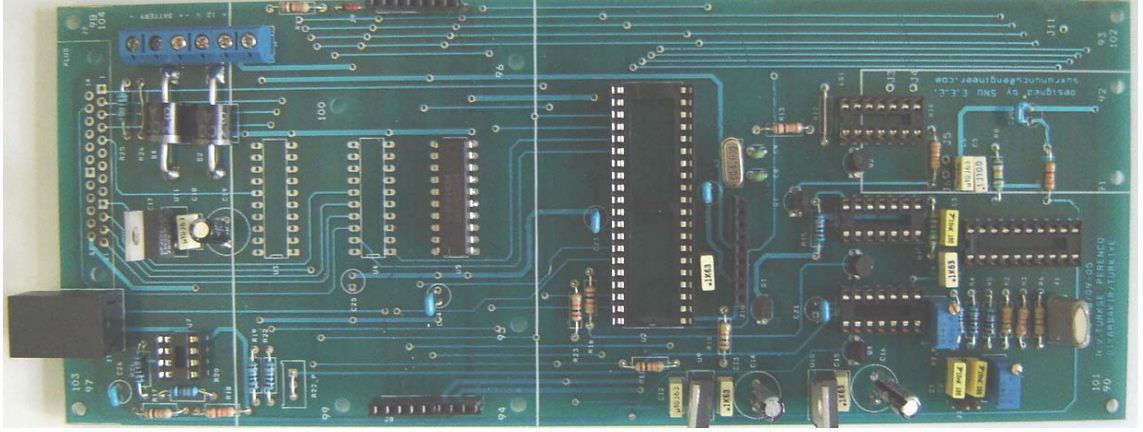
Bilgisayarla haberleşme kısmı

a) DTMF modem kısmı

b) Giriş/Çıkış kısmı

Tüm bu kısımların merkezinde, işlemlerin yönetilip yerine getirilmesinden sorumlu bir mikrodenetleyici yer almaktadır.

mikrodenetleyicinin bilgisayarla haberleşmesi biri giriş, diğeri çıkış için biçimlendirilmiş iki adet tampon (buffer) vasıtasıyla bilgisayarın paralel portu üzerinden yapılmaktadır.

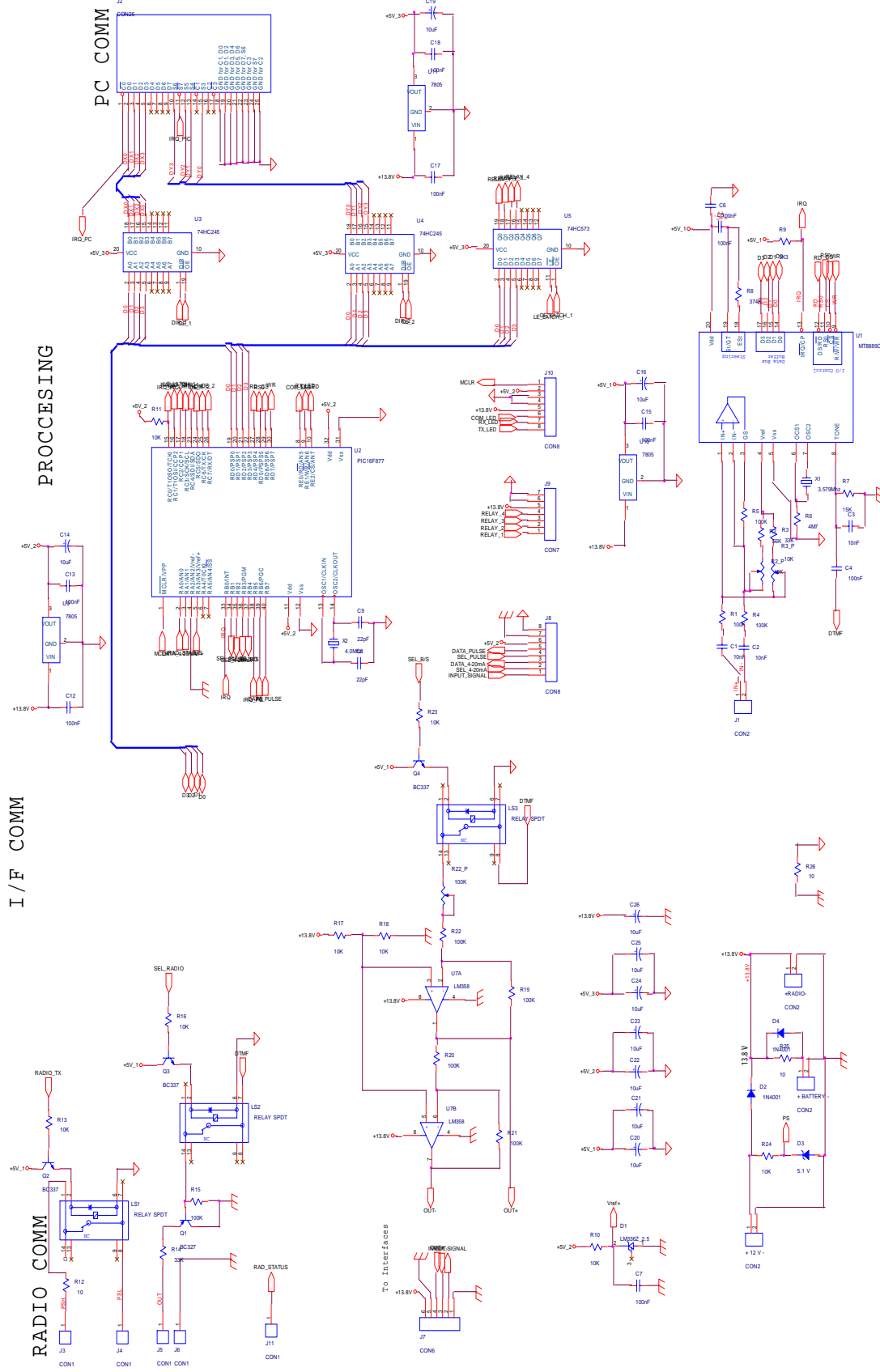


Resim.3. Terminal Ana Kartı

Mikrodenetleyici, hem telsizle haberleşmesini, hemde arayüzlerle haberleşmesini DTMF modem kısmı vasıtasıyla yapmaktadır. Yani hem telsizle hemde arayüzlerle, DTMF sisyalleri yardımıyla iletişim kurmaktadır.

Giriş/Çıkış işlemlerini, aşağıda Ölçme Kartının anlatılmasında açıklandığı gibi, röle çıkışları ve analog ölçüm bilgi girişi için ölçme kartı portu olarak ayırabiliriz. Ölçme kartı üzerinde yer alan iki adet röle pozisyonu ve analog sinyal ölçümleri için gerekli işlemler, mikrodenetleyici tarafından bu portlar sayesinde yapılır.

Ayrıca bu kart üzerinde, güç kaynağından 12V DC olarak gelen bir enerji girişide mevcuttur ki, bu enerji voltaj regülatörleri vasıtasıyla 5V DC olarak kart üzerindeki komponentlerin ve 12V DC olarak da telsizin ve bağlı olan seçici arayüzlerin beslemelerinde kullanılır. Yani birimin tüm enerji kontrolü de kart üzerinden yapılmaktadır.



Şekil.10. Terminal Ana Kartı Devre Şeması

TERMINAL UNIT Revised: Thursday, February 02, 2006

Revision: 02

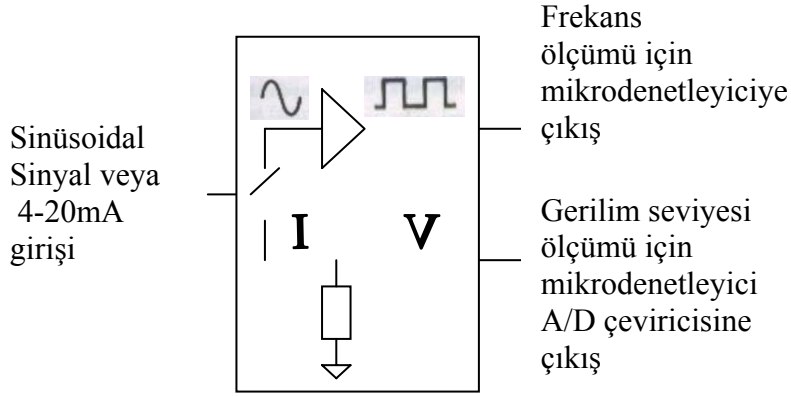
Bill Of Materials June 22,2006 15:15:39 Page1

Item	Quantity	Reference	Part
1	4	J1,+ 12 V -,+RADIO-, +BATTERY-	CON2
2	3	C1,C2,C3	10nF
3	9	C4,C5,C6,C7,C12,C13,C15, C17,C18	100nF
4	2	C9,C8	22pF
5	10	C14,C16,C19,C20,C21,C22, C23,C24,C25,C26	10uF
6	1	D1	LM336Z_2.5
7	2	D4,D2	1N4001
8	1	D3	5.1 V
9	1	J2	CON25
10	5	J3,J4,J5,J6,J11	CON1
11	1	J7	CON6
12	2	J8,J10	CON8
13	1	J9	CON7
14	3	LS1,LS2,LS3	RELAY SPDT
15	1	Q1	BC327
16	3	Q2,Q3,Q4	BC337
17	9	R1,R4,R5,R15,R19,R20,R21, R22_P,R22	100K
18	1	R2	56K
19	10	R2_P,R3_P,R10,R11,R13, R16,R17,R18,R23,R24	10K
20	2	R14,R3	33K
21	1	R6	4M7
22	1	R7	15K
23	1	R8	374K

24	1	R9	3K3
25	3	R12,R25,R26	10
26	1	U1	MT8889C
27	1	U2	PIC16F877
28	2	U4,U3	74HC245
29	1	U5	74HC573
30	1	U7	LM358
31	3	U9,U10,U11	7805
32	1	X1	3.579Mhz

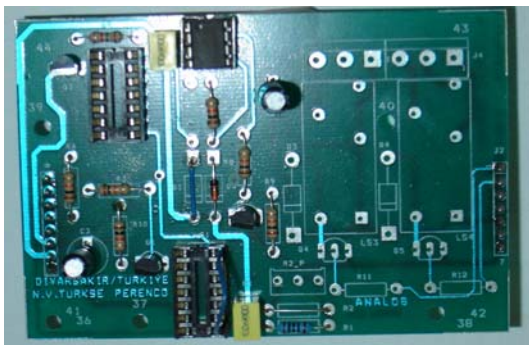
3.1.2.2. Ölçme Kartı

Ölçme kartı, seçici arayüzler vasıtasıyla, sensörlerden gelen analog sinyallerin mikrodenetleyici tarafından okunabilir biçime dönüştürüldüğü yerdir. Ayrıca 2 adet rölenin yer aldığı bir de çıkış kısmı mevcuttur.



Şekil.11. Ölçme Kartı Giriş/Çıkış Blok Şeması

Ölçme kartına gelen bir bilgi sinyali, hangi tip bir sinyal olacağı –sensör/transduser çıkışı olarak- önceden bilindiğinden, mikrodenetleyicinin okuyabileceği bir biçime dönüştürülmek için ölçüm kartı içerisinde rid röleler vasıtasıyla yine mikrodenetleyici tarafından yönlendirilir. Yukarıdaki şekilde görüldüğü gibi sinusoidal sinyaller kartın üst kısmında 0-5V karedalga biçimine dönüştürülür. Burada bilgi, sinyalin frekansıdır ki, bu biçim değişimiyle frekansda herhangi bir değişim söz konusu değildir.



Resim.4. Ölçme Kartı

4-20mA bilgi girişi ise, direnç üzerinden 0-2V şeklinde gerilim olarak çevrilerek mikrodenetleyicinin A/D girişine gönderilir. Aynı şekilde 0-5V şeklinde gelen

sinyallerde gerilim bölücüsü üzerinden 0-2V olarak mikrodenetleyicinin A/D girişine gönderilir.

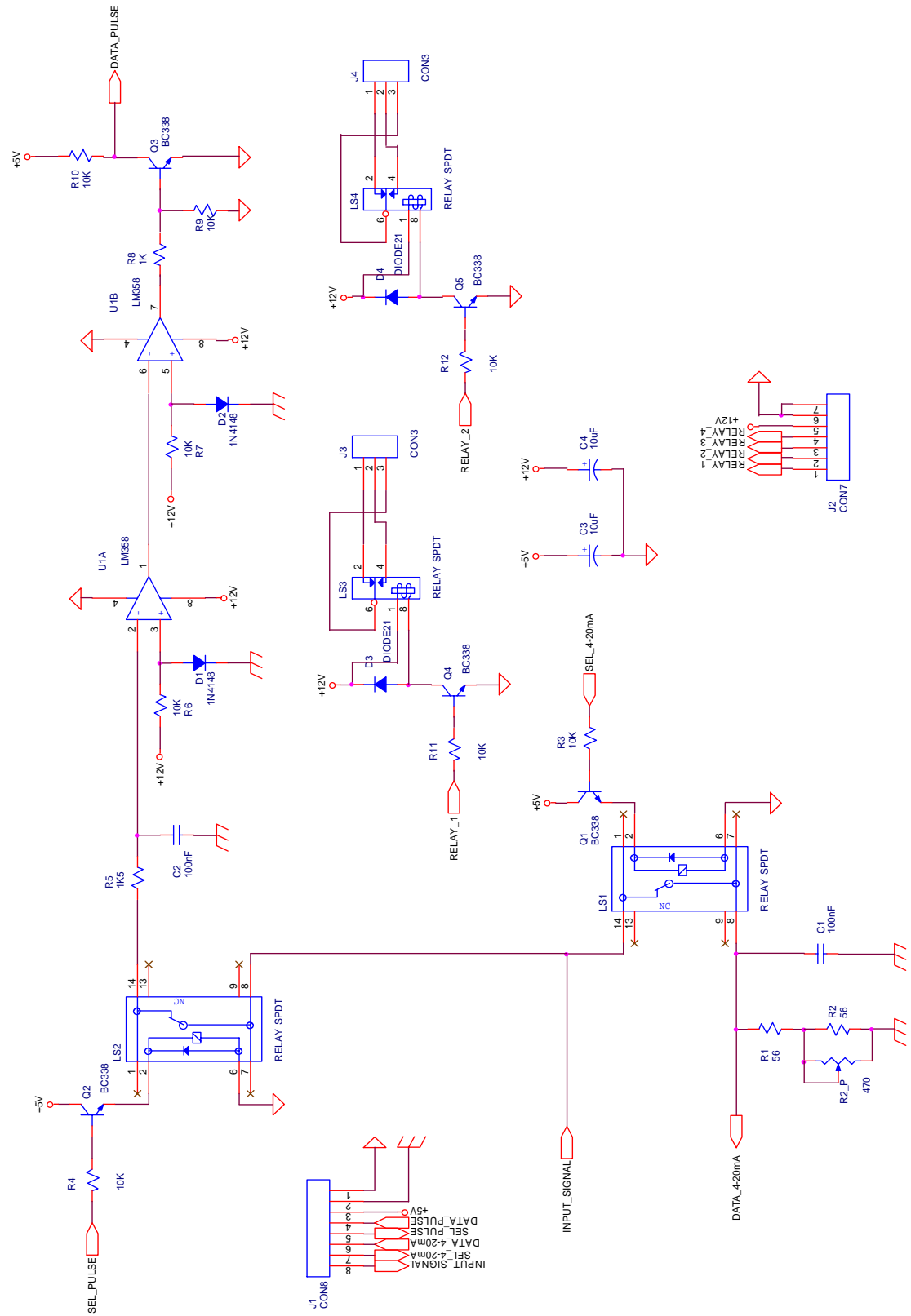
Kart üzerinde mevcut 2 adet röle, ihtiyaca göre on/off işlemleri veya alarm/set işlemleri için kullanılabilir. Röle bobinleri 12V DC gerilimle enerjilendirilir. Röle bobin enerjilendirmeleri bir NPN transistör yardımıyla, mikrodenetleyiciden veri barasıyla gelen bilginin bir 74HC573 tümleşikdevresinde tutulmasıyla yapılır ki; burada röle enerjilendirme bilgisinin tutulmasının amacı veri barasını sürekli meşgul etmeden, başka amaçlar için kullanabilmektir.

Bu röleler sayesinde özellikle petrol kuyularındaki pompaların uzaktan çalıştırma ve durdurma işlemleri testlerimizde başarıyla çalışmışlardır.

MEASURING Revised: Monday, September 12, 2005 Revision: 01

Bill Of Materials June 23,2006 7:22:45 Page1

Item	Quantity	Reference	Part
1	2	C2,C1	100nF
2	2	C4,C3	10uF
3	2	D1,D2	1N4148
4	2	D3,D4	DIODE21
5	1	J1	CON8
6	1	J2	CON7
7	2	J4,J3	CON3
8	4	LS1,LS2,LS3,LS4	RELAY SPDT
9	5	Q1,Q2,Q3,Q4,Q5	BC338
10	2	R1,R2	56
11	1	R2_P	470
12	8	R3,R4,R6,R7,R9,R10,R11, R12	10K
13	1	R5	1K5
14	1	R8	1K
15	1	U1	LM358



Şekil.12. Ölçme Elektronik Kartı Devre Şeması

3.1.2.3. Telsiz Cihazı

Motorola firmasının ürünü GM340 model araç telsizleri bu uygulamada kullanılmıştır.



Resim.5. Motorola GM340 Araç Telsiz Cihazı

UHF – 403-470MHz

VHF - 136-174MHz

Çıkış Gücü - 1-25W

Kanal Aralığı – 12,5 / 20 / 25 kHz

Sinyal Protokolü - SELECT 5

Kanal Sayısı – 255

Güç Kaynağı - 13,2 Vdc (10.8-15.6 Vdc)

Çalışma Sıcaklığı - -30 ile 60 C arası

Askeri Standartlar - 810 C,D,E

Koruma Standartları – IP54

Kullanılan telsiz cihazının çalışma frekansları aşağıdaki tabloda verildiği gibidir. Bu proje için tahsis edilmiş ve telemetri kanalları olarak belinen, biri simpleks diğeri yarı dubleks iki kanal mevcuttur.

Tablo.5. Projede Mevcut Telsiz Frekansları

	Kanal No	Tx (Mhz)	Rx (Mhz)	
Yarı Dubleks	1	153.675	153.675	
	2	153.200	153.700	
	3	153.475	153.975	Telemetri
Simpleks	4	153.675	153.675	
	5	153.200	153.200	
	6	153.475	153.475	Telemetri

3.1.2.4. Güç Kaynağı

Her bir Terminal Biriminin enerji beslemesinde 12 Vdc, 100 VA, güç sağlayan ve 220Vac, 50 Hz beslemeli bir güç kaynağı kullanılmıştır. Ayrıca Terminal ana kartı üzerinde, kesintisiz güç kaynağı için bir batarya besleme devresi ve o anda beslemenin batarya üzerinden mi yoksa şebeke üzerinden mi olduğunu mikrodenetleyiciye bildiren başka bir küçük devre de mevcuttur.

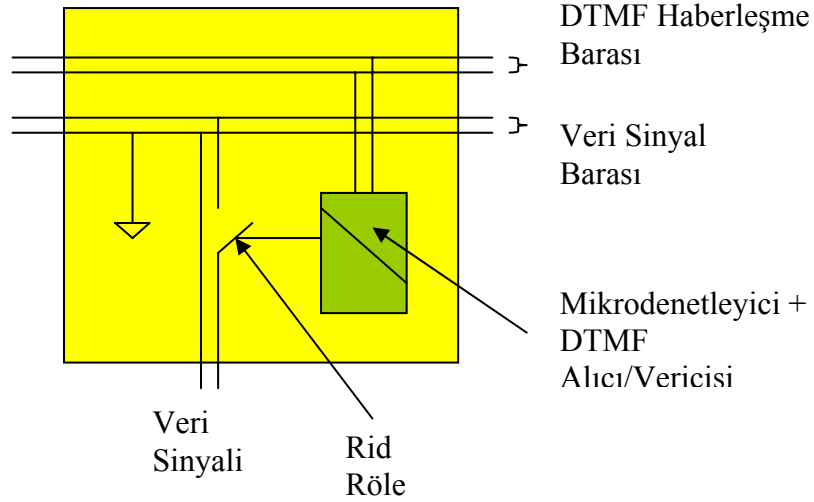


Resim.6. Terminal Birimi, Akü ve Dolabı

Batarya olarak, 12V, 12 Ah, kuru pil tabir edilen tekrar şarj edilebilir bir akü kullanılmıştır.

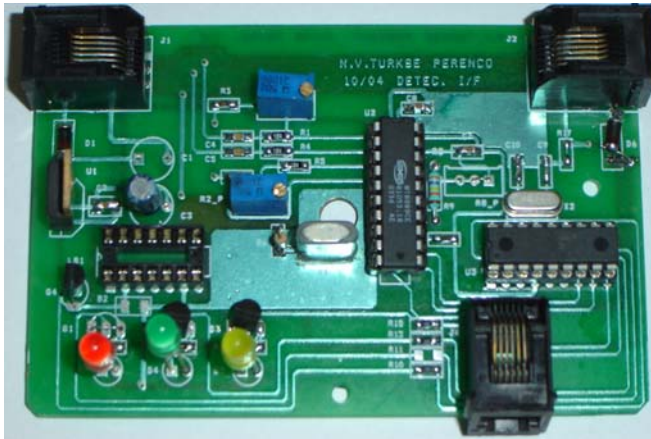
3.1.3. Seçici Arayüz Kartı

Mikrodenetleyiciden aldığı adreslenmiş bir emirle, sensörden gelen sinyali veri sinyal barası üzerinden Terminal Birimine gönderen birimdir.

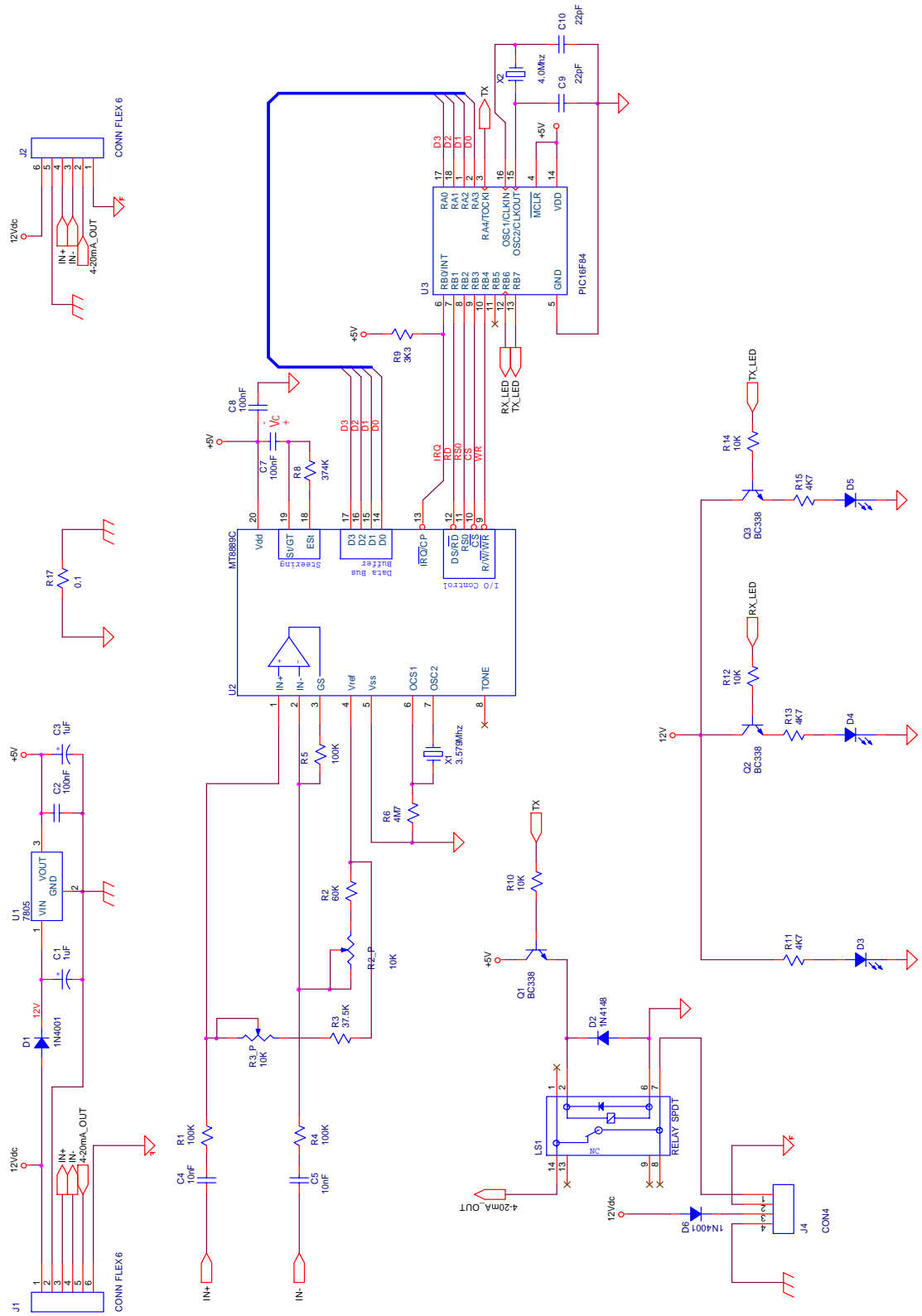


Şekil.13. Seçici Arayüz Kartı Blok Şeması

Haberleşme barasından gelen 4 karakterli DTMF sinyali, mikrodenetleyici tarafından gönderilmiş olup adres ve komut bilgileri içermektedir. Gönderilen bu sinyali tüm Seçici Arayüzler dinlerler. Mesajdaki adresin kendisi olduğunu anlayan arayüz, arkasından gelen emiri yerine getirir. Bu emir ya “Rid Röle’yi aç” yada “Rid Röle’yi kapat” olacaktır. Alınan bu komutlara göre, o seçici arayüze bağlı sensörün veri sinyali, ölçüm için veri barasına bağlanacak veya baradan yalıtılacaktır.



Resim.7. Seçici Arayüz Kartı



Şekil.14. Seçici Arayüz Kartı Devre Şeması

INTERFACE Revised: Wednesday, January 04, 2006

01 Revision: 01

Bill Of Materials June 23,2006 7:13:03 Page1

Item	Quantity	Reference	Part
1	2	C1,C3	1uF
2	3	C2,C7,C8	100nF
3	2	C4,C5	10nF
4	2	C9,C10	22pF
5	2	D6,D1	1N4001
6	1	D2	1N4148
7	3	D3,D4,D5	LED
8	2	J2,J1	CONN FLEX 6
9	1	J4	CON4
10	1	LS1	RELAY SPDT
11	3	Q1,Q2,Q3	BC338
12	3	R1,R4,R5	100K
13	1	R2	60K
14	5	R2_P,R3_P,R10,R12,R14	10K
15	1	R3	37.5K
16	1	R6	4M7
17	1	R8	374K
18	1	R9	3K3
19	3	R11,R13,R15	4K7
20	1	R17	0.1
21	1	U1	7805
22	1	U2	MT8889C
23	1	U3	PIC16F84
24	1	X1	3.579Mhz
25	1	X2	4.0Mhz

3.1.4. Terminal Birimi - Arayüz Bağlantı Kablosu

Terminal Birimi ve Arayüzler arasındaki bağlantı kablosu 6 adet çoklu kablonun oluşturduğu bir kablo demetidir. Bu 6 adet kablo, ikişerli olarak sırayla;

1. 12Vdc besleme barası,
2. DTMF Haberleşme barası,
3. Veri Sinyal barası olarak düşünülebilir.

12Vdc besleme barası, bağlı tüm arayüzlerin enerjisini taşıyan baradır. Enerjisini, Terminal ana kartındaki koruma diyotu üzerinden geçerek direkt olarak güç kaynağından alır.

DTMF Haberleşme barası, Terminal Biriminden gelen bilgileri arayüzdeki DTMF alıcı/vericisine taşıyan baradır. İki kablodan oluşan bara kapasitörlerle DC gerilimden yalıtılmış olarak her arayüzün alıcı/vericisine differansiyel giriş olarak bağlanmıştır.

Veri Sinyal barası, arayüzün, aldığı emirle sensör çıkışını rid röle yardımıyla Terminal Biriminde ölçmenin yapılması için bağladığı baradır. Bara, alınan sensör sinyalini, öncelikle mikrodenetleyicinin okuyabileceği bir forma çevirmek için ölçme kartına taşır. Dönüştürülen sinyalde, buradan mikrodenetleyiciye gönderilir.

3.2. Telemetri Sistemi Haberleşme Temelleri

Geliştirilen bu projede, haberleşme esas olarak 2 kısımda sınıflandırılabilir;

1. Bilgisayar- Mikrodenetleyici Haberleşmesi
2. Mikrodenetleyici -Telsiz Haberleşmesi

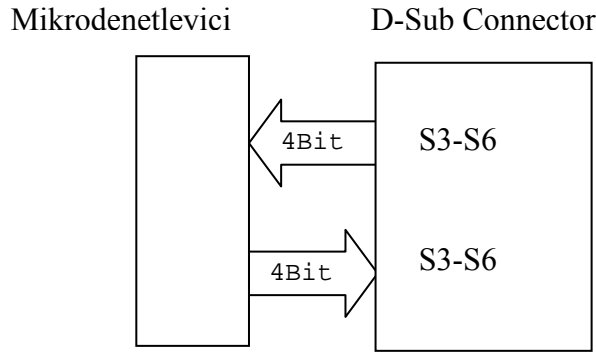
Burada, Bilgisayar- Mikrodenetleyici Haberleşmesi için paralel port kullanılmış, Mikrodenetleyici-Telsiz Haberleşmesi ise DTMF sinyalleriyle yürütülmüştür.

Bilgisayarı Terminal Birimiyle –dolayısıyla tüm sistemle- iletişimde tutan, bilgisayarın PARALEL PORT’udur.

3.2.1. Bilgisayar-Mikrodenetleyici Haberleşmesi

Yukarıda paralel port ve yazmaçları hakkında kısa bir hatırlatmadan sonra, tampon olarak kullandığımız iki adet 74HC245 yardımıyla, ana terminal ünitesinin bilgisayar ile olan bağlantısı iki ayrı yazmaç üzerinden yapılandırılmıştır. Giriş için S3-S6, çıkış içinse D0-D3 kullanılmıştır.

Bilgisayar ile iletişimdeki veri akışı, seri olarak 4 Bitlik 16'lı sayıların gönderilmesi ve alınmasıdır. Böyle bir veri akışında her bilgisayar ve ona ait paralel port haberleşmesinde veri portunun iki yönlü yapılandırılmasına gerek olmaksızın yapılabilecek en kolay çözüm, giriş için, salt okunur olan Durum Port'unu, çıkış içinse zaten salt çıkış olan Veri Port'unu kullanmaktır[1].



Şekil.15. Bilgisayar- Mikrodenetleyici Paralel Port Haberleşme Blok Şeması

Terminal-Bilgisayar bağlantısında paralel portun tercih edilme sebebi; hem kolaylık olsun diye sayıların 4 bit olarak işlenmesine olanak sağlamış ve hemde işlem hızı çok daha artmıştır. Dezavantaj olarak da mikrodenetleyicinin portunda, veri barası ve destek pinleri daha çok yer tutmuştur.

Verinin mikrodenetleyiciye tüm giriş ve çıkışı, denetleyicinin D portunun RD0-RD3 pinleri üzerinden olmaktadır. Bilgisayar haberleşmesindeki destek pinleri C portunda, telsiz ve arayüz haberleşmesindeki destek pinleri ise RC1 ve RD4-RD7'den meydana gelmektedir.

3.2.2. Mikrodenetleyici-Telsiz DTMF Haberleşmesi

3.2.2.1. DTMF

DTMF, “Dual Tone Multi Frequency” kelimelerinin baş harflerinden oluşur. Kelime anlamı “Çift Tonlu Çoklu Frekans” kodlama sistemidir. DTMF esas olarak Amerikan ordusu için Bell laboratuvarlarında geliştirilmiş bir kodlama sistemidir. Daha sonra telefon şebekelerinde bilgi yollamanın güvenli yolu olarak tercih edilmiş ve telefon abonesinin santrale aradığı abone ile ilgili bilgileri ilettiği standart yöntem olarak günümüzde yaygın olarak kullanılmaktadır[1].

DTMF kodlama sisteminde temel olarak dört adet iki çift ton kullanılır. Bu iki ton kombinasyonu ile 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, #, *, A, B, C, D rakam ve sembolleri ifade edilir[9].

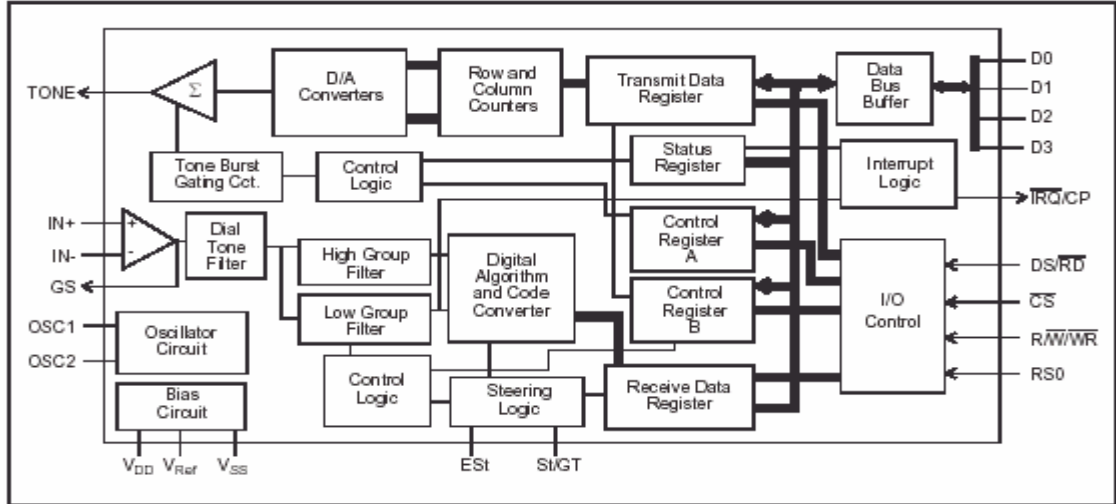
Tablo.6. DTMF Kod Frekansları.

KHz	1209	1336	1477	1633
697	1	2	3	A
770	4	5	6	B
852	7	8	9	C
941	*	0	#	D

Yukarıdaki DTMF Kod kombinasyon tablosundan görüleceği gibi dört adet frekans satır için, dört adet frekansta kolon için tahsis edilmiştir. Tablonun ortasındaki rakam ve semboller karşılığında bulunan satır ve sütundaki frekans çiftiyle ifade edilirler. Bir örnek vermek gerekirse **4** rakamını **770 Hz** ve **1209 Hz** lik ton çiftiyle ifade ederiz. Bu ton çiftleri 16 adet ton kombinasyonuna olanak tanır. Bu tonlardan sıra için adanmış olanlar 1 kHz ‘in altında, kolon için tahsis edilmiş olanlar ise 1 kHz ile 2 kHz arasındadır. Bu frekansların bu sınırlar dahilinde olmasının sebebi telefon ve telsiz sistemlerinde band geçiren filtreler kullanılır ve bu filtreler 300 Hz – 3000 Hz arasındaki konuşma aralığı dediğimiz aralıktaki frekansları geçirirler. DTMF tonlarında bozulma ve kesintilere uğramaması için bu frekans sınırları içinde kalacak şekilde tasarlanmışlardır[9].

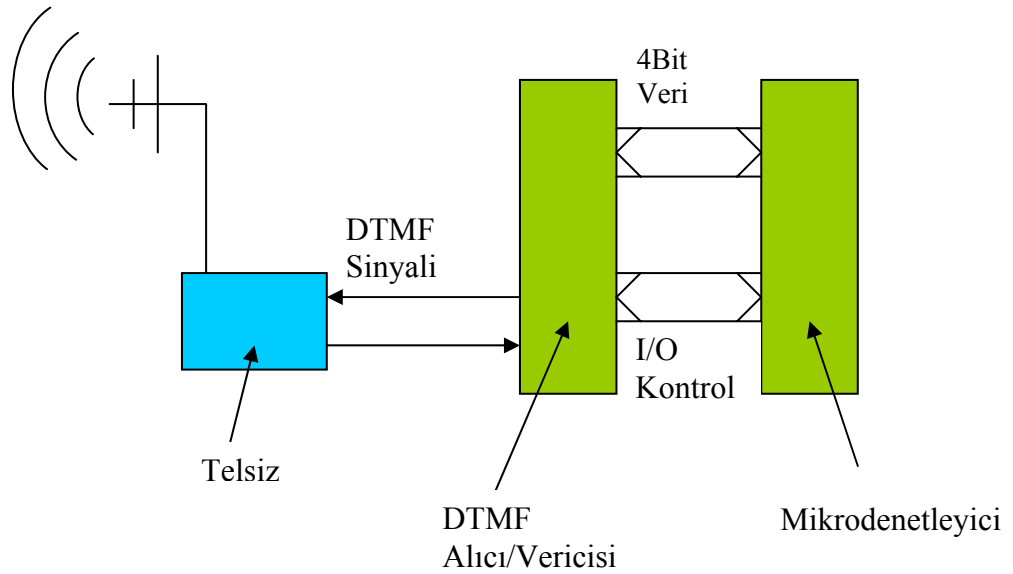
3.2.2.2. DTMF Alıcı/Vericisi

Bu projede kullanılmış olan, Zarlink Semiconductor ürünü DTMF alıcı/vericisi MT8889C, CMOS teknolojinin sunduğu düşük güç tüketimli ve yüksek güvenilirlikli bir komponenttir. DTMF alıcı/vericisi, dahili kazanç yükselticili bir alıcı ve gönderme ve durma zaman aralıkları önceden ayarlanmış bir DTMF üreticinden meydana gelmiştir.



Şekil.16. MT8889C DTMF Alıcı/Vericisi Blok Şeması

Uyarlanabilir arayüzü sayesinde mikrodnetleyicilerin, komponentin dahili yazmaçlarına erişimleri de mümkün olmaktadır.



Şekil.17. Mikrodnetleyici-DTMF Alıcı/Vericisi Blok Şeması

3.3. Ana Terminal Biriminden Bilgi Gönderme

Bigisayardan paralel porta gelen herhangi 6 adet 16'lı karakter dizisi, yukarıda bilgisayar-mikrodenetleyici haberleşmesinde anlatıldığı gibi paralel port üzerinden mikrodenetleyicinin h(3A), h(3B) , h(3C) , h(3D) , h(3E) , h(3F) yazmaçlarına sırasıyla yerleştirilir.

Burada,

$h(3A) + h(3B) =$ XY, ana terminal birimi adresini

$h(3C) + h(3D) =$ WA, uzaktaki terminal birimi adresini

$h(3E) + h(3F) =$ DJ, emri alacak olan birimin yapacağı iş emri

bilgilerini taşımaktadır.

Bu bilgilerin telsizin bilgi kanalına bindirilmesi yine yukarıda mikrodenetleyici-telsiz haberleşmesinde anlatıldığı gibi her bir sayının alıcı/vericiden geçirilerek DTMF sinyallerine dönüştürülmesiyle olur.

Terminal birimi mikrodenetleyicisi, yazılımındaki akışa göre yazmaçlarında saklı olan 6 sayılı XY WA DJ bilgisini, X'den başlayarak sırayla DTMF alıcı/vericisine gönderecektir. Alıcı/vericiden çıkacak olan DTMF tonları ise 50 ms gönder/50 ms bekle moduna göre çalışacak şekilde önceden yazılımla ayarlanmıştır.

Sesin mikrofon üzerine çarpmasıyla oluşan salınımın telsiz bilgi girişinde oluşturduğu gerilim dalgalanması, bazı alıcı/vericiden gelen DTMF sinyaliyle sürülen bir transistörün kollektörünün, telsiz girişine aynen bağlanmasıyla elektronik olarak modellenmiştir.

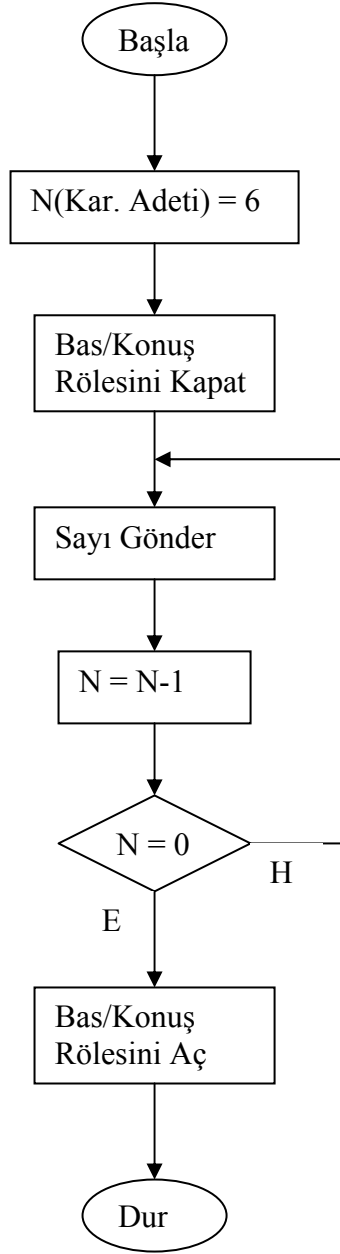
Telsizin gönderme butonu olan "Bas/Konuş" butonunun yerini alan rid rölesinin, mikrodenetleyici tarafından çektilmesiyle, gönderme kanalı açılan telsiz, DTMF sinyallerini göndermeye hazır durumdadır. Burada DTMF sinyalleri direkt telsize girmeden önce, yine mikrodenetleyici tarafından çektilen başka bir rid röle

üzerinden daha geçirilmiştir ki, bu rölenin işlevi daha sonra aşağıda, yeri geldiğinde anlatılacaktır. Her bir sayı, 50 ms gönder + 50 ms bekle toplam 100 ms'de gönderilir.

Burada, HAMLIN firmasının HE 721 A 05 10, açma-kapama ömrü çok yüksek ve hızlı, 5V DC bobin voltajlı, Normalde Açık rid rölesi kullanıldı ki, bu zaten haberleşme elektroniğinde oldukça bilinen, ekonomik bir komponenttir.

Ana Terminal Biriminden gönderilen toplam 6 karakterin gönderilmesi yaklaşık 600-700 ms sürer. Bu sürenin sonunda, Bas-Konuş butonu yerine mikrodenetleyici tarafından kapatılan rid röle tekrar aynı şekilde tersine bir işlemle bu sefer açılır. Burada, rölenin, tekrar açarak, telsizin gönderme modundan çıktığının doğrulanma bilgisi, telsizden alınan ve mikrodenetleyiciye giden RAD_STATUS sinyaliyle sağlanmaktadır. Bu bilginin eksikliği, ilk prototiplerde rölenin tekrar gerçekten açmadığı durumlarda oldukça ciddi sıkıntılara neden olmuştur. Artık rölenin gerçekten açmadığı bilirse, mikrodenetleyici açma işlemini yapana kadar döngüde kalacaktır ki doğrusuda budur.

Elektronik olarak modellenmiş bir devre yardımıyla telsiz bilgi bandına, DTMF sinyali olarak giren sayı bilgisi, başlarda kısaca bahsettiğimiz gibi telsiz cihazı içerisinde modülele edilerek –taşıyıcı frekansta zarflanmasıyla- anten üzerinden dış ortama gönderilir.



Şekil.18. Ana Terminal Biriminden Bilgi Gönderme Akış Şeması

3.4. Bilginin Uzaktaki Terminal Birimi Tarafından Alınması

Uzaktaki birimde, anten yoluyla, telsiz içerisindeki filtrelerden süzülen ve sonra da taşıyıcı zarfından ayrılan DTMF olarak kodlu, sayı bilgisi sinyali, telsizin hopperlör kanalına elektrik sinyali olarak gelir. İşte burada ikinci dikkat değer çözüm şekli olarak, bükülü sarmal (twisted pair) kabloyla telsizin hopperlör girişinden direkt olarak ayırdığımız bilgi sinyali diferansiyel giriş olarak düzenlenmiş devre üzerinden DTMF alıcı/vericisinin alıcı ünitesine girer.

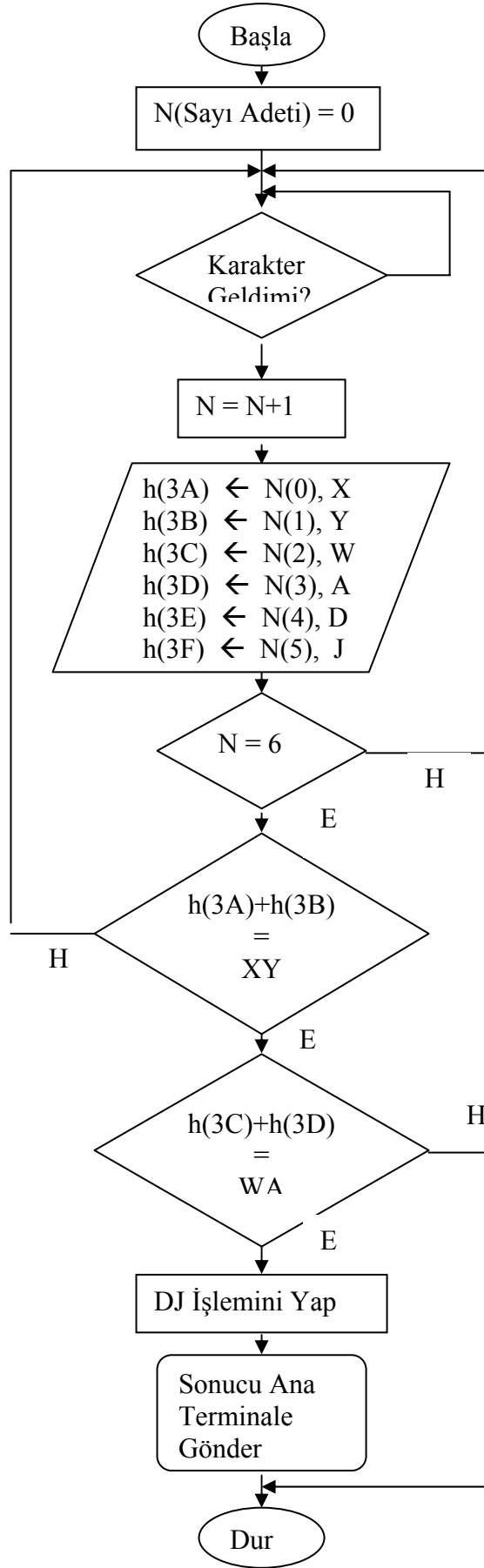
Burada, telsizden DTMF alıcı/vericisine giden kablonun bükülü sarmal şeklinde olması, gürültü yalıtımlı –ekranlı- kablo kullanılmadığından, gürültüye karşı bir ölçüde önlemdir. TAlıcı/verici girişinin differansiyel düzenlenmesinin nedeni ise, giriş OpAmpının VDD/2 de tutularak öngerilimlenmesidir.

Ana ünitenin gönderdiği, herbiri 50 ms gönder/50 ms bekle süresindeki 6 sayı bilgisi, DTMF alıcı/vericisinden mikrodnetleyiciye 2'li olarak her sayı için 4 Bit (1 Nipple) şeklinde gider. Bu 6 adet sayı bilgisi, aynı şekilde sırasıyla mikrodnetleyicinin h(3A), h(3B) , h(3C) , h(3D) , h(3E) , h(3F) yazmaçlarına yerleştirilir.

Mikrodnetleyicideki yazılımın tuttuğu bir sayıcı, gelen bilgi sayısını tutmaktadır ki; eğer herbiri en fazla 100 ms zaman sınırlaması içerisinde olmak üzere, 6 adet sayı bilgisi gelmediyse, sayıcı kendisini sıfırlayıp tüm yazmaçlarıda temizleyerek bekleme moduna tekrar döner. Burada, uzaktaki terminal birimi işlem yapabilmek için mutlaka 6 adet sayıyı beklemektedir. Gelen bu 6 adet sayı daha önce yukarıda belirttiğimiz gibi, mikrodnetleyiciye bilginin gönderilme adresini – yani hangi bilgisayar tarafından gönderildiği-, emrin gittiği adresi ve emir alanın yapması istenilen işin tanım kodunu içerir.

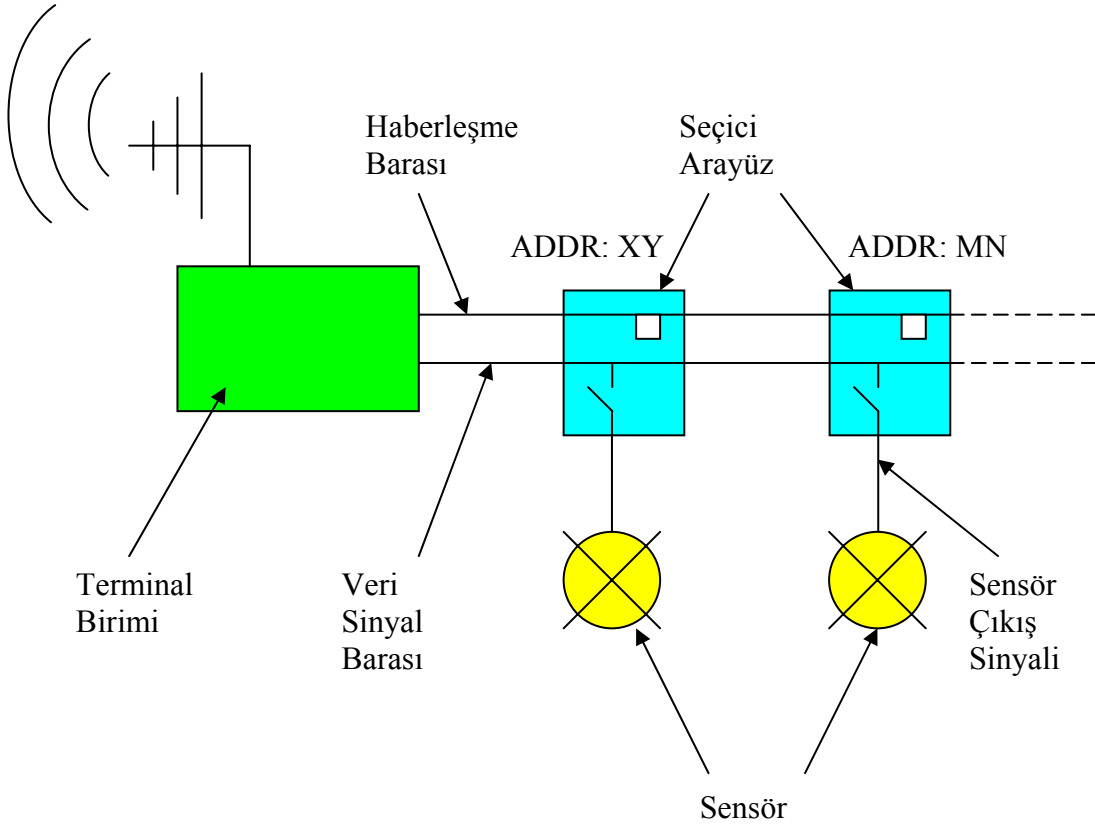
Yazmaçlarındaki ilk iki sayılık bilgiyi alan mikrodnetleyici, yazılıma göre, emri gönderen adresin kendise emir vermek için yetkili edilip edilmediğinin kontrolünü yapar. Emri gönderen kendisi için yetkili bir adres değilse, başka bir işlem yapmadan yazmaçlarını silerek bekleme moduna geri döner. Eğerki emri gönderen yetkili ise, daha sonraki iki yazmaçındaki gönderilen hedef adres bilgisinin kendi adres bilgisi olup olmadığını kontrol eder.

Hedef adres bilgisi kendi -kimlik- adres bilgisiyle çakışmıyorsa, tüm yazmaçları silerek bekleme moduna geri döner. Çakışıyorsa, program adresi son iki yazmakta saklanmakta olan ve yapılacak görevin yerine getirileceği altprogramın olduğu adrese gider. Bu programlar; basınç, seviye vb. gibi 4-20mA okunması, sinusoidal sinyal frekansı gibi akışmetre okunması veya röle açma/kapamaları gibi çıkış işlemleridir.



Şekil.19. Uzaktaki Terminal Birimi Tarafından Bilgi Alınması Akış Şeması

3.5. Terminal Birimi Tarafından Seçici Arayüz Yardımıyla Sensörün Okunması ve Sonucun Ana Terminale Gönderilmesi



Şekil.20. Terminal Birimi-Seçici Arayüz Bağlantı Blok Şeması

Terminal Birimindeki Terminal ana kartından gelen Adres+Görev bilgisi, haberleşme barası üzerinden tüm bağlı bulunan arayüzlere ilk 2 sayı adres, diğer 2 sayı görev, toplam 4 sayı olarak gönderilir. Örneğin “XY01” şeklinde gelen bilgiyle, XY seçici arayüzü 01 işlemini yapacak yani bağlı olduğu sensör çıkış sinyalini veri sinyal barasına göndercektir.

Terminal Birimi tarafından okuma işlemi bittikten sonra tekrar haberleşme barasına “XY00” bilgisi gönderilecek ki bu, XY seçici arayüzü, 00 işlemini yap yani rid rölsini aç demektir.

Burada yine, 16’lı sayı ile 256 adet arayüzü adreslemek mümkündür. Şu an ki uygulamada görev olarak yalnızca iki işlem tanımlanmıştır.

1. 00.....rid röleyi aç
2. 01.....rid röleyi kapat

Terminal Birimi haberleşmede telsiz üzerinden bilgi aktarımında kullanılan yöntemi bu sefer de kablo üzerinden Seçici Arayüzlerle haberleşmede kullanmaktadır. Yani Terminal biriminden arayüzlere gönderilen sayılar, DTMF olarak kodlanmış sinyallerdir. Terminal ana kartında, gönderilecek DTMF sinyalleri gideceği yerin ilgili rölesinin kapatıldığı kanal üzerinden ya telsiz cihazına yada arayüzlerle haberleşme için arayüz haberleşme barasına gider.

Seçici arayüz elektronik kartı esas olarak haberleşme için bir DTMF Alıcı/vericisi, bir PIC16F84 mikrodenetleyicisi ve sensör bilgisini ölçme barasına bağlayan bir HAMLIN rid rölesinden oluşur. Bütün bu düzenek, Terminal Birimi tarafından istenildiğinde rid rölenin kapanıp açılmasını sağlamak, böylece ölçüm barasında her zaman istenilen ölçüm sinyalini tutmak içindir.

Kodla belirtilen alt programdaki işlemin sonucunun tekrar ana üniteye bildirilmesi de şimdiye kadar yapılan işlemlerin tersine yapılmasıdır. DJ olarak kodlanan 2 adet 16'lı sayı bilgisiyle h(00) -h(FF) arası toplam 256 çeşit görev tanımı yapmak mümkündür. Bu görev tanımlarımız içerisinde sayabileceklerimiz şunlar olabilir.

- Tirübün tip bir akışmetreden akışın sinyalinin ölçülmesi
- 4-20 mA çıkış veren her türlü transmitterden bilgi alınması
- 0-5V çıkış veren transmitterlerin okunması
- 2 adet yardımcı röle vasıtasıyla aç/kapat işlemlerinin yapılması

Buna göre yapılan bu uygulamada, görev kodu ve karşılığına gelen işlem aşağıda verilmiştir. Örneğin 04, 4 Nolu akışmetredeki akış sinyalinin okunması, 09, 9 Nolu akışmetredeki akışın okunması, 43, 3 Nolu basınç sensöründen gelen ve basınç değerini belirten voltaj seviyesinin okunması, işlemidir.

```

;-----
;| X | W | A | D | J |
;-----
;          0 0          FLOWMETER_0 READ
;          0 1          FLOWMETER_1 READ

```

```

;      0 2      FLOWMETER_2 READ
;      0 3      FLOWMETER_2 READ
;      0 4      FLOWMETER_4 READ
;      0 5      FLOWMETER_5 READ
;      0 6      FLOWMETER_6 READ
;      0 7      FLOWMETER_7 READ
;      0 8      FLOWMETER_8 READ
;      0 9      FLOWMETER_9 READ
;      0 A      FLOWMETER_10 READ
;      0 B      FLOWMETER_11 READ
;      0 C      FLOWMETER_12 READ
;      0 D      FLOWMETER_13 READ
;      0 E      FLOWMETER_14 READ
;      0 F      FLOWMETER_15 READ
;      1 0      FLOWMETER_16 READ
;      1 1      FLOWMETER_17 READ
;      1 2      FLOWMETER_18 READ
;      1 3      FLOWMETER_19 READ
;-----
;      4 0      PRESSURE_0 READ
;      4 1      PRESSURE_1 READ
;      4 2      PRESSURE_2 READ
;      4 3      PRESSURE_3 READ
;      4 4      PRESSURE_4 READ

```

Görüldüğü gibi yukarıda yalnızca bir bölümü kullanılan görev kodlarına, ihtiyaca göre yeni eklemeler yapılarak görev alanı kolaylıkla genişletilebilir.

Bu projenin başlangıcında, uzaktaki birimlerden alınacak her türlü bilginin 4 16'lı sayı olarak geleceği öngörülmüştür. Bu varsayım sonucunda bilgisayara taşınacak olan her bilgi, ham bir şekilde mutlaka 4 adet 16'lı sayı olarak ana üniteye gönderilir ve bu ham bilginin işlenmesi işlemi de bilgisayardaki yazılımda yapılır.

Verinin neden 4 sayı olarak gönderildiği ise gayet açıktır; daha fazla sayıya bu uygulamada ihtiyaç yoktur.

Ayrıca 4 sayılı verinin, haberleşmede birde avantajı olmuştur ki; hem uygulamaya büyük kolaylık getirmiş hemde haberleşme hızını arttırmıştır.

Yukarıda bahsettik ki; ana terminal biriminin gönderdiği her bilgi mutlaka 6 sayıdır ve her uzaktaki terminal birimi dışarıdan 6 sayılılık bilgi bekler, oysa kendisi, mutlaka 4 sayılılık bilgi gönderecektir. İşte çözümün ana triki buradadır; bilgi göndermek için konuşmaya başlayan bir uzaktaki terminal birimini ana terminal biriminin dışında doğal olarak diğer tüm birimlerde dinleyecekler, 4. sayının okunması sonunda 5. sayı bilgisinin gelmesini 100 ms daha bekleyecekler, başka bilgi gelişi olmadığından da bu konuşmanın kendilerine gelmediğini anlayarak hepsi yazmaçlarını silecek ve tekrar dinleme modunda bekleyeceklerdir. Ancak ana terminal birim zaten 4 sayılılık bir bilginin geliyor olacağını beklediğinden, gelen 4 sayılılık ham bilgiyi alacaktır.

İki adet 16'lı sayı olarak gelen Hedef Adres Kodu (WA) bilgisiyle h(00)-h(FF) aralığında 256 değişik uzak terminal birimine bilgi göndermek teorik olarak mümkündür. Aşağıda bu uygulamada çalışan uzaktaki terminal birimlerinin adresleri verilmiştir.

```

;|X|Y|W|A|D|J|
;      0 . 0      KURKAN B/S
;      0 . 1      PIRINCLIK
;      0 . 2      KUS-H01
;      0 . 3      BEY-H27
;      0 . 4      BARBES-DEEP

```

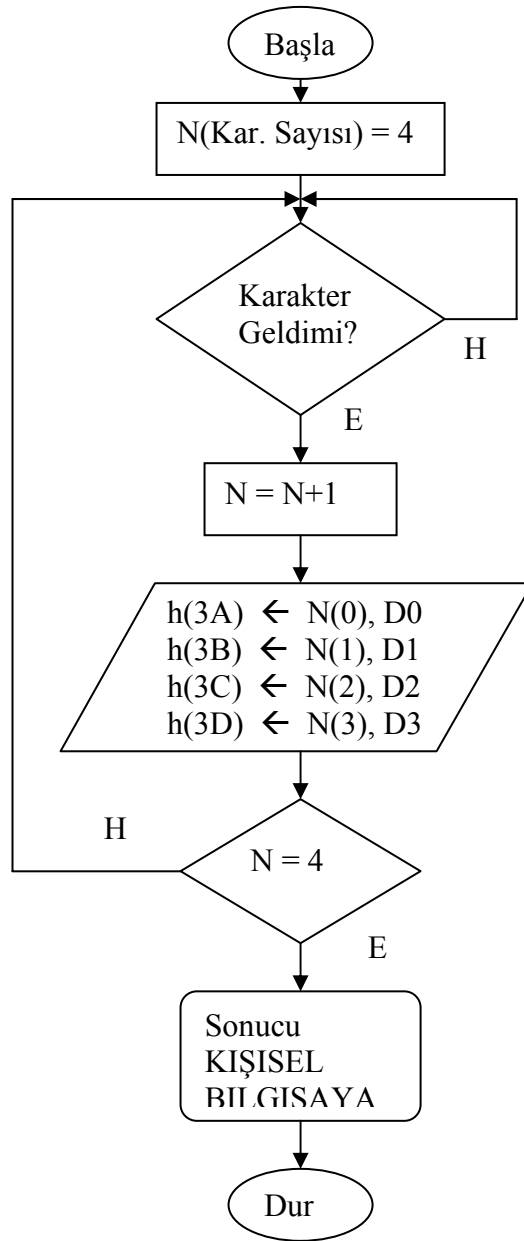
3.6. Ana Terminale Bilginin Alınması Ve Bilgisayar Tarafından Okunması

Yukarıda anlatılanın tersi bir doğrultuyla Uzaktaki Terminal Biriminden, anten yoluyla, Ana Terminal Biriminde telsiz içerisindeki filtrelerden süzülen ve sonra da taşıyıcı zarfindan ayrışarak DTMF olarak kodlu sayı bilgi sinyali, telsizin hopperlör kanalına elektrik sinyali olarak gelir ve diferansiyel giriş olarak düzenlenmiş devre üzerinden DTMF Alıcı/vericisinin alıcı ünitesine girer.

Uzaktaki Terminal Biriminin gönderdiği, herbiri 50 ms gönder/50 ms bekle süresindeki 4 sayı bilgisi, DTMF alıcı/vericisinden mikrodenetleyiciye 2'li olarak her

sayı için 4 Bit şeklinde gider. Bu 4 adet sayı bilgisi aynı şekilde sırasıyla mikrodenetleyicinin h(3A), h(3B) , h(3C) , h(3D) yazmaçlarına yerleştirilir

Yukarıda Bilgisayar–Mikrodenetleyici haberleşmesinde anlatıldığı gibi, mikrodenetleyici yazmaçlarındaki 4 sayılık veri, bu defa bilgisayar paralel portunun durum portundan el sıkışmalı bir haberleşmeyle ve yazmaç sırasıyla bilgisayara girerler.



Şekil.21. Ana Terminal Birimi Tarafından Bilgi Alınması Akış Şeması

3.7. Telemetri Sisteminde Kullanılan Ölçüm Sensörleri

Bu projede uzaktan veri aktarımında kullanılan bazı sensörler hakkında tamamlayıcı bilgiler aşağıda verilmiştir.

3.7.1. Ultrasonik Tank Seviye Sensörleri

3.7.1.1. Ultrason Tanımlaması

İnsan kulağı 20-20,000 Hertz (saniyedeki salınım) frekans aralığındaki sesleri duyabilir. Ultrason, 20,000 Hertz' den daha yüksek frekansdaki sestir. Tespitsel (diagnostik) Ultrason 1-10 Megahertz aralığında çalışır.

Ultrason, parçacık hareketinin yürüyen dalgayla aynı doğrultuda olduğu boylamsal (longitudinal) dalga formunda hareket eder. (longitudinal dalga; titreşim ve hareket doğrultuları özdeş olan dalga çeşidi, sıkışma ve genleşmelerle ilerlerler, ses, deprem dalgaları bu türdendirler). Bu dalgalar, ultrason transduser tarafından üretilir. Boylamsal dalgalar, dalga içerisinde sıkışma ve genleşme bölgelerinin hareketi boyunca enerji transfer ederler. Temel dalga fiziğinin prensiplerinin birçoğu, aynı biçimdeki ultrason fiziği içinde geçerlidir.

Herhangi bir ortamda hareket eden ultrasonik dalganın dirayeti, ortamın özellikleriyle sınırlıdır ki; bu özellikler, o ortama özel-spesifik- akustik empedansı oluşturan elastikiyet ve yoğunluğu içerir. İlerleme, kullanılan transduser frekansı tarafında sınırlanır ki, yüksek frekanslar daha kısa dalgaboyuna sahiptir ve düşük frekanslardan daha az nüfuz ederler.

Ultrason hüzmesi, farklı akustik empedanslı dokularla (ortamla) karşılaştığında hızları öyle değişir ki, transduser tarafından farklı zamanlarda ve farklı şiddetlere sahip geri dönen ekolar alınır. Zamandaki ve şiddetteki farklılıklar, sistemdeki bilgisayar ve kullanıcı tarafından ayarlanır. Dokulardaki ses dalga hız değerlerinin bu bilgisi, ultrason cihazı tarafından, monitörde tespitsel görüntünün üretilmesi için kullanılır.

3.7.1.2. Temel Formül

Ses 1db/MHz/cm değeriyle zayıflar

İnsan dokusunda ortalama ses hızının 1540 m/sn olduğu farzedilir.

Hız: $v=f\lambda$

Akustik empedans: $z=v.c$

Zayıflama katsayısı: $1dB=10.Log_{10}(I^2/I^2)$

Snell Kanunu: $\frac{Sini}{Sinr} = \frac{V1}{V2}$

Şiddeti (I): $dB=10Log \left(\frac{Ar}{Ai} \right)^2 = 20 Log Ar \left(\frac{Ar}{Ai} \right) \frac{Power(W)}{Area(cm^2)}$

v - hız (m/sn)

f - frekans (Hz)

λ – dalgaboyu (m)

z – akustik empedans (g/cm^2)

c - yoğunluk

I - şiddet

i - oluşan hüzme

r - kırılan hüzme

A - genlik

3.7.1.3. Ultrason Transdüzeri

Transduser, enerjinin bir formunu başka birine dönüştürebilen cihazdır. Ultrason uygulamalı için, transduser, ultrasonik palsları ve ekoları gönderir ve alır. Bu transduser, elektrik (impulse) itkilerini mekanik dalgalara çevirir ve tersini yapar. Verici olarak transduser, kütleye (vucuda) giren dalgaları üreten piezoelektrik kristali kullanır. Alıcı olarak da transduser, yükseltme, kompanze etme, demodülasyon, sıkıştırma ve ayırmayı içeren bir sürü fonksiyona sahiptir.

3.7.1.3.1. Piezoelektrik Etkisi

Çoğu modern ultrason transduserlerde, kristal olarak, insan yapımı (sunni) kurşun zirkonet ve kurşun titanet kullanılır. Bu kristaller, basınç (ses) dalgaları üreten titreşimleri, elektriksel uyaran dönüşüm için gerekli özelliklere sahip piezoelektrik (basınç-elektrik) kristalleri olarak adlandırılır. Bu eşsiz kalite, ultrason sinyalinin bir transduser tarafından yayılmasını ve alınmasını sağlar.

3.7.1.3.2 Darbe İlkeleri

Elektrik, piezoelektrik kristale –spesifik- belirli bir darbe sıklığıyla uygulanır. Bu, ekolar dinlenene kadar dalganın ilerlemesine izin verir. Genellikle transduser, 999 mikrosaniyelik tekrarlama periyotlu 1 mikrosaniye süreli darbeler gönderir. Bu sebepten dolayı, zamanın çoğunda transduser, “dinleme” modundadır. Darbe oranı, mekanik taramanın osilasyon sıklığıyla kontrol edilebilir.

3.7.1.4. Ultrason Hüzmesi

Tespitsel ultrasonun etkili uygulanması ve emniyeti garanti etmek için ultrason hüzmesi hakkında temel bilgilere sahip olmamız gerekir.

3.7.1.4.1. Dalga Yayılması

Yayılma (propagation), ses enerjisinin uzayın bir bölümünden diğerine ilerlemesidir. Yayılma suresince, hüzmenin zayıflaması (attenuation) kaçınılmazdır.

3.7.1.4.2. Frekans

Ultrason hüzmesinin frekansı, seçilen transduserdeki kristalin büyüklüğüyle belirlenir. Dokudaki sesin hızı nispi olarak sabit olduğundan, hüzmenin dalgaboyunu $v = f \cdot \lambda$ formülünü kullanarak belirleyebiliriz. Bunu anlamak önemlidir çünkü, nüfuz derinliği dalga boyuyla doğru orantılıdır. Böylece, daha yüksek frekanslı transduserler daha iyi çözünürlükte ve nüfuz yetenekleri zayıftır.

3.7.1.4.3. Güç/Yoğunluk

Ultrason gücü, en çok kullanılanı desibel (dB) olan birkaç yolla ifade edilir. Şiddet, birim zamanda birim alandan geçen enerji miktarıdır. (Bkz formül)

3.7.1.4.4. Odaklama

Odaklanmamış haldeki ultrason transduserinden yayılan hüzmelerin iki ayrı bölgesi vardır; yakın bölge (Fresnel Zone) ve uzak bölge (Fraunhofer Zone). Yakın bölge, transduser yüzüyle aynı genişlikte ve odak noktasına doğru uzanır (odak noktaları elektronik görüntülerle değişkendir). Bu noktanın ötesinde hüzmeler ayrılır ve uzak bölge olarak adlandırılır. Bu ayrılmadan dolayı, uzak bölgedeki çözünürlük yakın bölgeden daha kötüdür.

Odaklama, hüzmeyi bir odak noktasına yönelten akustik lensler, hüzmeyi kıran bir ayna veya hüzmeyi çıkış noktasından yönelten bükülmüş kristal yüzeyi kullanımıyla yapılabilir. Günümüzde en çok kullanılan ise, transduserin fazlanmış sıralı ipinin değişken elektronik odaklamasıdır.

3.7.1.5. Ultrasonun Çalışması

3.7.1.5.1. Maddeyle Etkileşim

Ultrasonun kullanılan bilgisi, hüzmeye geçtiği dokularla etkileşirken, alınan ekoların sonucudur.

3.7.1.5.2. Akustik Empedans

Akustik empedans (z) dokunun elastikiyet ve yoğunluğunun bir fonksiyonudur. Yüksek empedanslı materyaller sesi diğerlerinden daha hızlı iletirler, fakat gelen dalga tarafından sürekli sıkıştırılmaya izin vermezler. Bazı bilinen akustik empedans değerleri;

Kan	$1.6 \times 10^5 \text{ gm/cm}^2$
Kemik	$7.8 \times 10^5 \text{ gm/cm}^2$
Yağ	$1.4 \times 10^5 \text{ gm/cm}^2$
Yumuşak Doku Ort.	$1.6 \times 10^5 \text{ gm/cm}^2$
Hava	$0.004 \times 10^5 \text{ gm/cm}^2$
Su	$1.5 \times 10^5 \text{ gm/cm}^2$

3.7.1.5.3. Hız Değişkenleri

Aşağıda sesin özel dokulardaki hızı için ölçülmüş değerlerin listesi verilmiştir.

Kan	1570 m/sn
Kemik	4080
Yağ	1450
Yumuşak Doku Ort.	1540
Hava	0330
Su	1480

3.7.1.5.4. Zayıflama

Vucuttaki farklı dokular boyunca yol alırken ses hüzmesi zayıflar. Değişen (farklı akustik empedanslı) dokulara girerken hüzme saçılabilir veya transduzere doğru geri yansıtılabilir. Kırılma (hüzme bükülmesi) ve soğrulma da hüzmeyi zayıflatabilir. Birim derinlikte hüzmeden çıkan enerji miktarı zayıflama katsayısı olarak ifade edilir; $\text{dB} = 10 \text{ Log}_{10}(I^2/I^2)$. Bu basitçe, geri dönen ekoların şiddetinin, orjinal ses hüzmesinin şiddetine oranının logaritmik ifadesidir. Frekans artarken zayıflama katsayısı da artar. Genel bir kural olarak, geçilen her dokunun cm^2 si için her MHz'de 1 dB hüzme zayıflar.

3.7.1.5.5. Kırılma

Hüzme söz konusu objeye tam dik değilse, ses kırılması olabilir. Bu durumda hüzme, ışığın hava-su arayüzeyinde büküldüğü gibi bükülür. Bükülme açısı Snell kanunu kullanılarak hesaplanabilir.

$$\frac{\sin I}{\sin R} = \frac{V1}{V2}$$

Tüm dik olmayan hüzmeler kırılırlar. Bazıları doku yüzeyinde gelişigüzel saçılırlar.

3.7.1.5.6. Soğrulma

Hüzme enerji kaybının başka bir sebebi de geçilen dokular tarafından soğrulmadır. İlgili objenin derinliği, soğurmanın ana belirleyicisidir. Bu kaybolan enerji, ısı formunda dokuya verilir.

Depolama ve işlem kaplarında, tanklarda, kuyularda, rezervuarlarda ve silolarda sıvı seviye ölçümlerine daima ihtiyaç vardır. Saf sudan yapışkan ve çürütücü –korozif-sıvılara kadar, seviye ölçümü gerektiren bir çok durum mevcuttur. Seviye sensörlerinin ortamı, vakumdan yüksek basınca ve sıfır Selsiyus derecesinden yüksek sıcaklıklara kadar değişir. Farklı durumların ilişkisinden dolayı, pek çok tip seviye sensörü geliştirilmiştir.

Günümüzde, ham petrol ayırıcıları kullanımındaki kontrol stratejileri , basitce, uygun çoklu-arayüz seviye ölçüm sistemlerinin bulunmasındaki zorluklarla karşılaşılırlar. Ham petrol ayırıcılarındaki çoklu-arayüzün tam doğru ölçümü, hidrokarbon ölçümü için bir meydan okuma sayılır. Harici monte edilmiş yer değiştiriciler, differansiyel basınç transmitterleri, ultrasonik transduzerler, tek-elektrodlu ve çok-elektrodlu kapasitans sensörleri gibi çok sayıda metod geliştirilmiştir. Yüksek güvenilirlik, sıvı çeşitliliği, zorlu ortamlar ve kendine özgü emniyet kuralları sebebiyle ham petrol ayırıcılarındaki arayüz ölçümleri için petrol endüstrisinde kullanılan metodlar sınırlıdır. Deniz çıkan ham petrolün büyük miktarda su içermesinden dolayı -ki ek bir taşıma maliyetinden sakınmak için bu su ayrılmalı- bu problemle ilgili kıyıdan uzak

petrol firmaları için olay, özellikle önemlidir. Bu sebepten, farklı arayüz seviyelerinin tam ve güvenli ölçümü için uygun sensörlerin araştırılması gereklidir.

Bir ham petrol ayırıcısı normalde, gaz, petrol, su, emülsiyon ve tortu içerir. Platformdaki sınırlı alandan dolayı, ham petrol ayırıcılarının etkili bir izleme ve kontrolü, petrol şirketlerine büyük kar sağlayacak ve çevresel kirlenmeyi düşürecektir. Çoklu-arayüz seviye ölçüm sistemlerinin başka diğer uygulamaları, petro istasyonlarındaki petrol/kerosen depolama tankları ve havalanlarındaki yakıt depolama tanklarıdır. Bu uygulamalar için bir ideal çoklu-arayüz seviye ölçüm sistemi aşağıdaki özellikleri göstermelidir.

1. Gaz-petrol, petrol-emülsiyon ve emülsiyon- su gibi farklı materyaller arasındaki arayüzlerin tam olarak saptanması,
2. Düşük bakım maliyeti,
3. Kendini temizleme,
4. Emniyetli

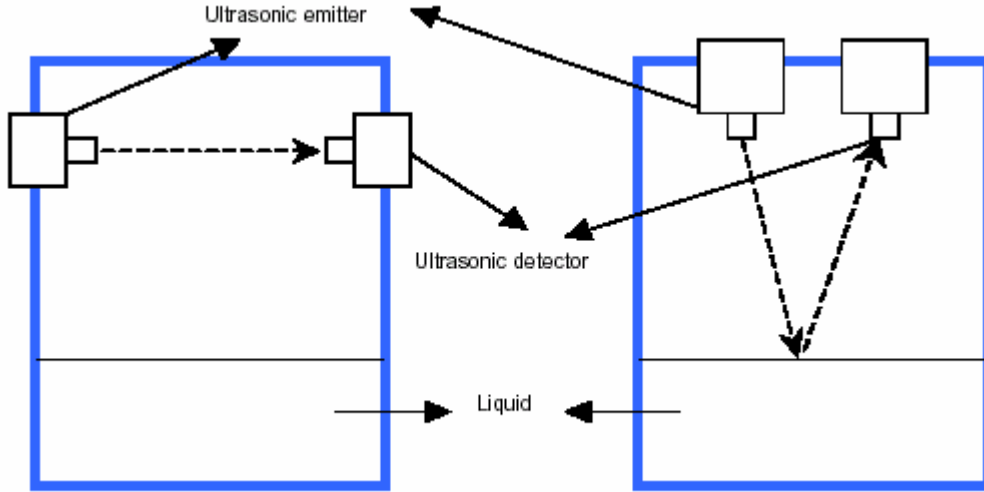


Resim.8. Ham Petrol Ayırım Tankı

Ultrasonik teknoloji , karışmış sıvıları ayırmak, arayüzü göstermek ve asılmış katıları gözlemek ve farketmek için kullanılabilir.

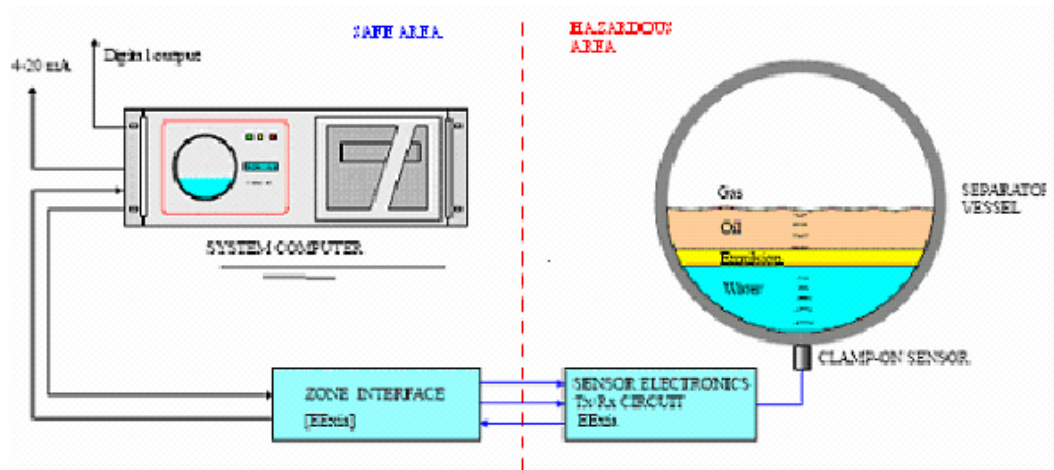
Karışmamış sıvılar arasında arayüzün bulunması için iki teknik mevcuttur; ultrasonik zayıflama ve ultrasonik yansımaya.

Ultrasonik seviye ölçüm sistemi, bir ultrasonik verici ve alıcıdan oluşur. Ölçülen hacimdeki farklı arayüzlerden yansımış sinyaller ortamın yoğunluğu ve ortamdaki sesin hızının bir fonksiyonu olduğundan ultrasonik alıcı tarafından alınır. Bu metod ayrık ve sürekli seviye gözlenmesinde kullanılmıştır.



Şekil.22. Tank Üzerinde Çeşitli Tip Ultrasonik Seviye Sensörü Yerleşimleri

Norveçteki Christian Michelson Araştırma Enstitüsü, sıvı seviye ölçümü için Ultrasonik Arayüz Seviye Dedektörü (Ultrasonic Interface Level Detector), UID, denen bir sensör geliştirmiştir. Bu teknikte ultrasonik transduser, petrol ayırıcısının alt yüzeyinin dışına monte edilmiştir. Ayırıcıyla direkt temestadır ve dik eksen boyunca ultrasonik palsları yayar ve alır.

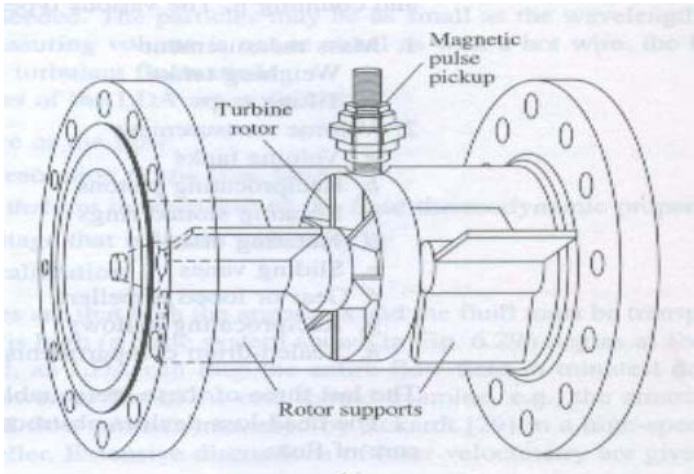


Şekil.23. Ham Petrol Tankı, Ultrasonik Seviye Sensörü Ve 4-20mA Çıkışı

Ölçüm pals ekosunda temellenmiştir ve seviyeler, akışkan arayüzlerinden gelen ekoların ölçülmüş uçuş –yol alma- zamanlarının ve yayılma ortamındaki ses hızının birleşimiyle tahmin edilir. Bu sistemin avantajları, mevcut veya yeni ayırıcılara kolay yerleştirilmesi ve kalibrasyona ihtiyaç duymamasıdır. Bu sistem bugün ESSO ve STATOIL tarafından ticari kullanımdadır. Sistemin dezavantajı ise; emülsiyon ve köpüğün gözlenmesindeki zorluktur.Sıvıdaki hava kabarcıklarının varlığı ultrason dalgalarını zayıflatır ve dağıtır.Sıvıda önemli miktarda hava olduğunda, bu büyük problem olur. Kum taneciklerinin varlığında UID çalışmalarında probleme yol açar.

3.7.2. Türbin Tip Akışmetre

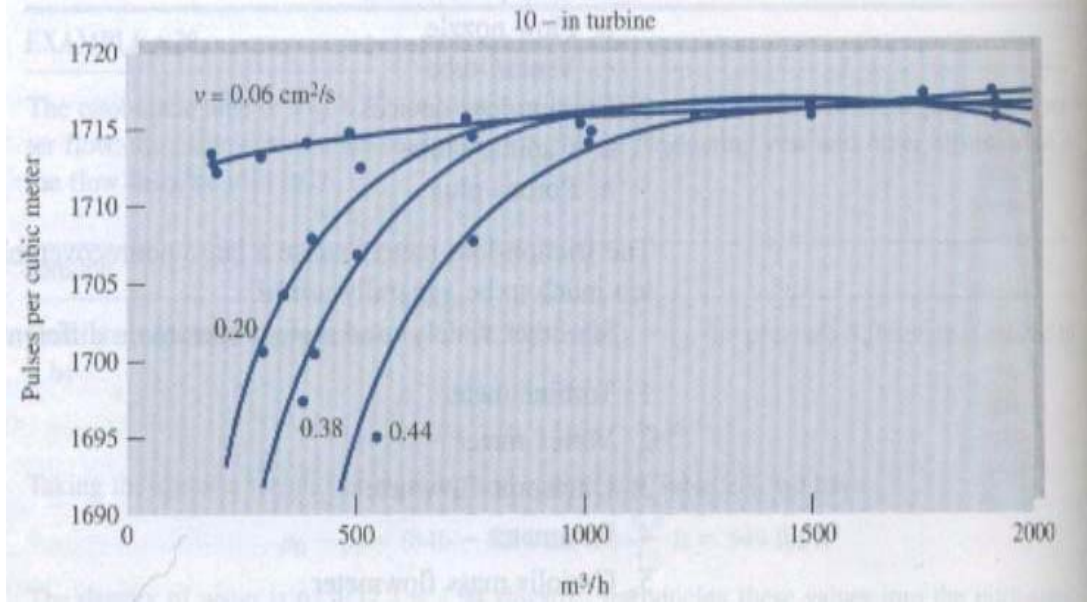
Pervaneli Akışmetre de denilen türbin tip akışmetre, boru hattına üzerine yerleştirilen serbest dönen bir pervanedir. Rotor dönüşü, rotor üzerinde bir noktadan geçişin sebep olduğu elektrik veya manyetik palslar tarafından ölçülür. Bu dönüş boru üzerindeki hacim akışıyla yaklaşık olarak doğru orantılıdır.



Şekil.24. Türbin Tip Akışmetre

Türbin tip akışmetrenin esas avantajı; her pals sıvının artan hacmine karşılık gelir ve bu palslar sayısala çevrilerek kolayca toplanabilirler. Sıvı-akış türbinmetreleri üzerinde her birim sıvı hacmi için 5:1 akış oranı aralığında ± 0.25 kesinlikle sabit pals üreten kanatlar vardır. Gaz metrelerde, yeterli torku üretmek için daha çok kanata ihtiyaç vardır ve hassasiyet ± 1 dolayındadır.

Her türbinmetrenin çok kendine özgü karakteristikleri olduğundan, akış kalibrasyonu mutlak bir gerekliliktir. Tipik bir sıvı-metre kalibrasyon eğrisi Şekil.25.'de görülmektedir.



Şekil.25. Ham Petrol İçin Tipik Bir Kalibrasyon Eğrisi

4. BULGULAR

Telsiz cihazına kodlanmış DTMF sinyalleri olarak bindirilen adres, emir ve veri içeren sayısal bilgi dizi dizileri, alıcı ve verici arasındaki haberleşme akışı sonucunda yerine getirilmiş veya getirilememiş emirleri, veriyi, tarih zaman ve adres içerikleriyle birlikte bir veri kütüğüne aktarır.

Her işlem, bilgisayarın ana terminal birimine paralel port üzerinden verdiği emirle başlar. Uzaktaki terminal birimleriyle kablosuz haberleşmenin ana unsurlarından olan telsiz cihazları, projede modül olarak yer almıştır.

Transduserden gelen analog ölçüm sinyalleri yalnızca bir ölçüm kartında elektronik olarak işlendiğinden ve yalnızca merkez bilgisayarda o sinyale ait matematiksel işlemler yapıldığından sahadaki, sensörlere takılı göstergelere de ihtiyaç da kalmamıştır.



Resim.9. Bir Akışmetre Göstergesi

Bu uygulamanın benzerlerinden farklı en önemli özelliği, mevcut sisteme çok düşük bir maliyetle uygulanmış olmasıdır ki, burada en sınırlayıcı eleman olabilecek telsiz cihazı bile, ünitenin başka bir marka veya model telsiz cihazıyla çalışabilmesinde engel değildir. Yani çok küçük bir değişiklikle sistem başka bir model telsiz cihazıyla çalışabilecek hale gelebilmektedir.

Tablo.7. Bir Üretim İstasyonuna Uygulanmış Sistemin Maliyet Tablosu.

BEY-H27 Uzaktaki Terminal Birimi	Adet	Birim Fiyat (\$)	Fiyat (\$)
✓ Ana Terminal Kartı	1	82	82
✓ Motorola GM340 Telsiz cihazı	1	419	419
✓ Güç Kaynağı Modülü ve Bataryası	1	43	43
✓ Ölçme kartı	1	10	10
✓ Anten	1	50	50
✓ Konnectörlü Anten Kablosu	2	5	10
			614
✓ Seçici Arayüz	5	30	150
✓ -	0	-	
			150
5 Kuylu Bir Alt İstasyon için Toplam Maliyet			764

Tüm haberleşme matemetiğinde 16'lı sayı sistemi kullanılmıştır ki, bu yaklaşım bize dört haneli değer aktarımında, 16'lı sayı sisteminde 10'lu sayı sistemine göre daha büyük değerlerin taşınabilmesine olanak vermiştir.

Örneğin, 16'lı sistemde dört haneli verilmiş bir değer h(FF96), eğer 10'lu sayı sisteminde verilmiş olsaydı d(65430) olacaktı. Görüldüğü gibi 2 sayı daha fazladan göndermek zorunda kalınılacaktı. Daha önce de belirtildiği gibi, sistemin matemetiğinin iyi konulması, performansını, yani hem erişim hızını hemde güvenilirliğini artıracaktır.

Burada geliştirilmiş bu modelde, hem malzeme ve hemde yazılım bazında hiçbir sınırlamaya uğramayacak ve bağımlı kalmayacak şekilde bir tasarım hedeflenmiş ve başarılmıştır.

Bilgisayar-terminal birimi haberleşmesinde, kendine özgü, el sıkışmalı bir haberleşme biçimi oluşturulmuş ve başarıyla çalıştırılmıştır.

Altı sayı gidiş, dört sayı da dönüş şeklinde diyebileceğimiz her bir okuma veya işlem periyodu yaklaşık 3-4 sn sürmektedir ve yazılımın bir zorlaması olarak bilgisayarın

gönderdiği her emir karşı tarftan onaylanmadıkça bilgisayarda yapılmış (yerine getirilmiş) kabul edilmez.

4-20 mA, pulse, 0-5V, On/Off, vb., gibi endüstriyel sinyallerin yalnızca bir kart üzerinden ve işlenebilirliği, tasarımın sağladığı büyük kolaylıklardandır. 4-20 mA' in okunması; sinyalin 100 Ohm bir direnç üzerine düşürülerek gerilime dönüştürülmesiyle olmuştur. Pals sinyalleri ise yükselteç üzerinden 0-5V kare dalga formuna dönüştürülerek mikrodenetleyiciden okunur hale getirilmiştir.

Yukarıda bahsettiğimiz gibi, telsize veri girişinde mikrofonunun yerini alacak elektronik modelleme –zorlu bir çalışma ve deneme sonucu -çok basit bir çözümlenmeyle başarılmıştır. Normalde sesin havada yarattığı dalgalanma, mikrofonda bir kapasitorün iki yüzünden birisi olan diyaframı titreştirir. Oluşan bu titreşim, bir elektronik devrenin sürücüsü olarak iş görür. İşte, bu sürücünün elektronik modellemesi, yukarı kısımda anlatıldığı gibi, kurulmuş ve başarıyla çalıştırılmıştır.

Paralel port haberleşmesinde, hemen her bilgisayarda salt olarak veri portları çıkış ve durum portlarında giriş olarak yapılandırıldığı için, bu yapıya dokunmadan ve her paralel port kartını kullanabilecek bir tasarım düşünülmüş ve bilgisayarda tüm bilgi çıkışı “veri yazmaçı” ve tüm bilgi girişide “durum yazmacı” olarak yapılmıştır.

5. SONUÇLAR VE TARTIŞMA

Bir bilgisayar, buna bağlı bir terminal birimi, ve uzaktan veri alınacak yerde konumlanmış uzaktaki terminal birimi sayesinde, merkez bilgisayara veri taşıma amacıyla geliştirilmiş bu proje, aynı zamanda üniversite – sanayi işbirliğinin kurulduğu ve gelişmelerinin ve sonuçlarının uygulandığı, yerli sanayi ve ticaretin sundukları olanaklarla nasıl hedeflere ulaşılacağı hakkında birazda olsa cesaret vermesi bakımından anlamlı olmuştur.

Sistem, geniş bir alanda faaliyet gösteren ham petrol üretim şirketleri için ise,

- Değer alma işleminde, işçi, araç ve bunlar için harcanan risk maliyetlerini düşürmüş,
- Olaya müdahale (örneğin, açma/durdurma) hızını arttırmış,
- Verinin sıkıştırılarak sayısal bir ortamda saklanarak, daha sonra her türlü sorgulanabilmesine olanak vermiştir.
- Akış ölçümlerinde, yalnızca akışmetrenin sensöründen gelen sinyalin kullanılması ve dolayısıyla bir ölçüm göstergesine ihtiyaç duyulmaması, gösterge maliyetini sıfırlamış ve bunlar için harcanan servis maliyetlerini de yok etmiştir,
- Mevcut network ve intranet sayesinde veri kütüğünde saklanan bilgilere, şirketin ihtiyaç duyan başka birimlerince erişimi kolaylaşmıştır,

Projenin, sistem olarak sağladığı faydalar gayet aşikardır. Yine DTMF sinyalleriyle görüntünün kodlanarak gönderilmesinin, 1200 bps taşıma kapasiteli telsizle mümkün olabilirdiği, ileride üzerinde çalışılabilecek bir konu olarak görünmektedir.

Bu uygulama, bir boru hattı üzerinde –biri hat başına, diğeri de hat sonuna- yerleştirilmiş iki akışmetrenin okuduğu akış değerlerinin karşılaştırılması, sıvıların sıkıştırılmazlığı prensibinden yola çıkarak, bizlere hat üzerinde, herhangi bir kaçak veya sızıntı gibi bir problemin olup olmadığı, varsa problemin büyüklüğü hakkında bilgi vermesi bakımından kullanıcıya büyük olanak sağlayacaktır.

Yazılıma yapılacak eklemelerle gerek duyulan yerlerde, okuma deęerleri bilgisayar ekranında gösterilirken, örneęin aynı zamanda okunan deęer için sınır limitleri geilmiřse, bilgisayar ekranından operatörü uyarıcı efektler oluşturulabilir.

Uzaktaki ünitelerin merkezi bilgisayar tarafından sırayla okunması, aslında ünitelerin bilgi için taranması işlemidir. Bu tarama esnasında, bir uniteye tekrar okuma sırasının gelme süresi, sisteme baęlı ünite sayısı ile doęru orantılıdır. Yani ünite sayısı fazla ise, okuma periyodu uzun, dolayısıyla okuma sıklığı düşüktür. Bu ise beraberinde bir dezavantaj getirmektedir. Örneęin, okuma işlemi bitmiř bir birimden tüm taramanın sonuna kadar haber alınamamakta, ancak periyot sonunda tekrar haber alınabilmektedir.

Bu durum, bir birimde oluşabilecek problemlerden bazan ancak bir periyot sonunda haberdar olabileceęimizi gösterir ki, buna başka çözüm teknikleri geliřtirmek aslında başka proje konularından birisi olabilir.

Telsiz üzerinden, gece gündüz 24 saat sürekli yapılan bu bilgi aktarım ve kontrol işlemi, tabii ki havada sürekli bir frekans işgali olacaktır.

Bu projenin başka uygulanabilirlięi, ulařımda sinyalizasyon, enerji daęıtım hatları, elektronik tarım, sulama sistemleri vb., olabilecek görünmektedir.

6. EKLER

EK - 1

ANA ÜNİTE KİŞİSEL BILGISAYARLAR PROGRAMI

Visual Basic 6

Option Explicit

Dim CU, WA, DJ, C, U, W, A, D, J, D0, D1, D2, D3, DIGIT, DGT3, DGT2, DGT1, DGT0, DGT3_, DGT2_, DGT1_, DGT0_, N As Byte

Dim WADJ, DATA, K, DUMMY, M, H As Integer

Dim AVG As Integer

Dim TIME_INIT, TIME_FINAL, ECE As Date

Private Sub Command1_Click()

'Out &H37A, &H0

Unload Form2

WA = 0 'Radio'nun adresi

Form2.Show

Form2.Label3 = "KUS 02"

End Sub

Private Sub Command2_Click()

Unload Form2

WA = 1

Form2.Show

Form2.Label3 = "KUS 10"

End Sub

Private Sub Command3_Click()

Unload Form2

WA = 2

Form2.Show

Form2.Label3 = "KUS 11"

End Sub

Private Sub Command4_Click()

Unload Form2

WA = 3

Form2.Show

Form2.Label3 = "KUS 12"

End Sub

Private Sub Command6_Click()

Unload Form2

WA = 3

Form2.Show

Form2.Label3 = "BEY 07"

End Sub

'End Sub

Private Sub Command7_Click()

Unload Form2

WA = 3

Form2.Show

Form2.Label3 = "BEY 14"

End Sub

Private Sub Command8_Click()

Unload Form2

WA = 3

Form2.Show

Form2.Label3 = "BEY 15"

End Sub

Private Sub Command9_Click()

Unload Form2

WA = 3

```
Form2.Show  
Form2.Label3 = "BEY 27"  
End Sub
```

```
Private Sub Command10_Click()  
Unload Form2  
WA = 3  
Form2.Show  
Form2.Label3 = "BEY 50"  
End Sub
```

```
Private Sub Command11_Click()  
Unload Form2  
WA = 3  
Form2.Show  
Form2.Label3 = "PIR IN"  
End Sub
```

```
Private Sub Command30_Click()  
Unload Form2  
WA = 3  
Form2.Show  
Form2.Label3 = "PIR REV"  
End Sub
```

```
Private Sub Command12_Click()  
Unload Form2  
WA = 3  
Form2.Show  
Form2.Label3 = "DRN 6"  
End Sub
```

```
Private Sub Command13_Click()
```

```
Unload Form2
WA = 3
Form2.Show
Form2.Label3 = "DRN 12"
End Sub

Private Sub Command14_Click()
Unload Form2
WA = 3
Form2.Show
Form2.Label3 = "KAW 04"
End Sub

Private Sub Command15_Click()
Unload Form2
WA = 3
Form2.Show
Form2.Label3 = "SAH RTN"
End Sub

Private Sub Command16_Click()
Unload Form2
WA = 3
Form2.Show
Form2.Label3 = "BAY 01"
End Sub

Private Sub Command17_Click()
Unload Form2
WA = 3
Form2.Show
Form2.Label3 = "BAY 08"
End Sub
```

```
Private Sub Command18_Click()  
Unload Form2  
WA = 3  
Form2.Show  
Form2.Label3 = "BEK 01"  
End Sub
```

```
Private Sub Command19_Click()  
Unload Form2  
WA = 3  
Form2.Show  
Form2.Label3 = "BEK 02"  
End Sub
```

```
Private Sub Command20_Click()  
Unload Form2  
WA = 3  
Form2.Show  
Form2.Label3 = "KUR 05"  
End Sub
```

```
Private Sub Command21_Click()  
Unload Form2  
WA = 3  
Form2.Show  
Form2.Label3 = "KUR 02"  
End Sub
```

```
Private Sub Command22_Click()  
Unload Form2  
WA = 3  
Form2.Show  
Form2.Label3 = "KUR 04"  
End Sub
```

```
Private Sub Command23_Click()  
Unload Form2  
WA = 3  
Form2.Show  
Form2.Label3 = "KUR 25"  
End Sub
```

```
Private Sub Command24_Click()  
Unload Form2  
WA = 3  
Form2.Show  
Form2.Label3 = "KUR 33"  
End Sub
```

```
Private Sub Command25_Click()  
Unload Form2  
WA = 3  
Form2.Show  
Form2.Label3 = "KUR 40"  
End Sub
```

```
Private Sub Command26_Click()  
Unload Form2  
WA = 3  
Form2.Show  
Form2.Label3 = "KUR 41"  
End Sub
```

```
Private Sub Command27_Click()  
Unload Form2  
WA = 3  
Form2.Show
```

```
Form2.Label3 = "KUR 42"
```

```
End Sub
```

```
Private Sub Command28_Click()
```

```
Unload Form2
```

```
WA = 3
```

```
Form2.Show
```

```
Form2.Label3 = "KUR 47"
```

```
End Sub
```

```
Private Sub Command29_Click()
```

```
Unload Form2
```

```
WA = 3
```

```
Form2.Show
```

```
Form2.Label3 = "KUR 49"
```

```
End Sub
```

```
Private Sub Command31_Click()
```

```
Unload Form2
```

```
WA = 3
```

```
Form2.Show
```

```
Form2.Label3 = "OUT PIR"
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
Out &H37A, &H0
```

```
Timer1.Interval = 25
```

```
'Wells
```

```
CU = 4          'Central Unit Address
```

```
'-----
```

```
MSFlexGrid5.Row = 0
```

```
MSFlexGrid5.Col = 0
```

```
MSFlexGrid5.CellFontSize = 8
```

```
MSFlexGrid5.CellAlignment = 4
```

```
MSFlexGrid5.CellBackColor = &H80FFFF
```

```
MSFlexGrid5.Text = "BARBES-DEEP"
```

'-----

```
MSFlexGrid4.Row = 0
MSFlexGrid4.Col = 0
MSFlexGrid4.CellFontSize = 8
MSFlexGrid4.CellAlignment = 4
MSFlexGrid4.CellBackColor = &H80FFFF
MSFlexGrid4.Text = "PIRINCLIK"
```

```
MSFlexGrid4.Row = 1
MSFlexGrid4.Col = 0
MSFlexGrid4.CellFontSize = 8
MSFlexGrid4.CellAlignment = 4
MSFlexGrid4.CellBackColor = &HFFFF80
MSFlexGrid4.Text = "PIR IN"
```

```
MSFlexGrid4.Row = 2
MSFlexGrid4.Col = 0
MSFlexGrid4.CellFontSize = 8
MSFlexGrid4.CellAlignment = 4
MSFlexGrid4.CellBackColor = &HFFFF80
MSFlexGrid4.Text = "PIR REV"
```

```
MSFlexGrid4.Row = 0
MSFlexGrid4.Col = 1
MSFlexGrid4.CellFontSize = 8
MSFlexGrid4.CellAlignment = 4
MSFlexGrid4.CellBackColor = &HFFC0FF
MSFlexGrid4.Text = "RATE(bgpd)"
```

```
MSFlexGrid4.Row = 0
MSFlexGrid4.Col = 2
MSFlexGrid4.CellFontSize = 8
MSFlexGrid4.CellAlignment = 4
```

MSFlexGrid4.CellBackColor = &HC0FFC0

MSFlexGrid4.Text = "COMM.STAT"

'-----

MSFlexGrid1.Row = 0

MSFlexGrid1.Col = 0

MSFlexGrid1.CellFontSize = 8

MSFlexGrid1.CellAlignment = 4

MSFlexGrid1.CellBackColor = &H80FFFF

MSFlexGrid1.Text = "KUS-H01"

MSFlexGrid1.Row = 1

MSFlexGrid1.Col = 0

MSFlexGrid1.CellFontSize = 8

MSFlexGrid1.CellAlignment = 4

MSFlexGrid1.CellBackColor = &HFFFF80

MSFlexGrid1.Text = "KUS 02"

MSFlexGrid1.Row = 2

MSFlexGrid1.Col = 0

MSFlexGrid1.CellFontSize = 8

MSFlexGrid1.CellAlignment = 4

MSFlexGrid1.CellBackColor = &HFFFF80

MSFlexGrid1.Text = "KUS 10"

MSFlexGrid1.Row = 3

MSFlexGrid1.Col = 0

MSFlexGrid1.CellFontSize = 8

MSFlexGrid1.CellAlignment = 4

MSFlexGrid1.CellBackColor = &HFFFF80

MSFlexGrid1.Text = "KUS 11"

MSFlexGrid1.Row = 4

MSFlexGrid1.Col = 0

MSFlexGrid1.CellFontSize = 8


```
MSFlexGrid1.CellAlignment = 4
MSFlexGrid1.CellBackColor = &HFFFF80
MSFlexGrid1.Text = "KUS 12"
```

```
MSFlexGrid1.Row = 0
MSFlexGrid1.Col = 1
MSFlexGrid1.CellFontSize = 8
MSFlexGrid1.CellAlignment = 4
MSFlexGrid1.CellBackColor = &HFFC0FF
MSFlexGrid1.Text = "RATE(bgpd)"
```

```
MSFlexGrid1.Row = 0
MSFlexGrid1.Col = 2
MSFlexGrid1.CellFontSize = 8
MSFlexGrid1.CellAlignment = 4
MSFlexGrid1.CellBackColor = &HC0FFC0
MSFlexGrid1.Text = "COMM.STAT"
```

```
'-----
MSFlexGrid2.Row = 0
MSFlexGrid2.Col = 0
MSFlexGrid2.CellFontSize = 8
MSFlexGrid2.CellAlignment = 4
MSFlexGrid2.CellBackColor = &H80FFFF
MSFlexGrid2.Text = "BEY-H27"
```

```
MSFlexGrid2.Row = 1
MSFlexGrid2.Col = 0
MSFlexGrid2.CellFontSize = 8
MSFlexGrid2.CellAlignment = 4
MSFlexGrid2.CellBackColor = &HFFFF80
MSFlexGrid2.Text = "BEY 07"
```

```
MSFlexGrid2.Row = 2
MSFlexGrid2.Col = 0
```

```
MSFlexGrid2.CellFontSize = 8  
MSFlexGrid2.CellAlignment = 4  
MSFlexGrid2.CellBackColor = &HFFFFFF80  
MSFlexGrid2.Text = "BEY 14"
```

```
MSFlexGrid2.Row = 3  
MSFlexGrid2.Col = 0  
MSFlexGrid2.CellFontSize = 8  
MSFlexGrid2.CellAlignment = 4  
MSFlexGrid2.CellBackColor = &HFFFFFF80  
MSFlexGrid2.Text = "BEY 15"
```

```
MSFlexGrid2.Row = 4  
MSFlexGrid2.Col = 0  
MSFlexGrid2.CellFontSize = 8  
MSFlexGrid2.CellAlignment = 4  
MSFlexGrid2.CellBackColor = &HFFFFFF80  
MSFlexGrid2.Text = "BEY 27"
```

```
MSFlexGrid2.Row = 5  
MSFlexGrid2.Col = 0  
MSFlexGrid2.CellFontSize = 8  
MSFlexGrid2.CellAlignment = 4  
MSFlexGrid2.CellBackColor = &HFFFFFF80  
MSFlexGrid2.Text = "BEY 50"
```

```
MSFlexGrid2.Row = 0  
MSFlexGrid2.Col = 1  
MSFlexGrid2.CellFontSize = 8  
MSFlexGrid2.CellAlignment = 4  
MSFlexGrid2.CellBackColor = &HFFC0FF  
MSFlexGrid2.Text = "RATE(bgpd)"
```

```
MSFlexGrid2.Row = 0
```

```
MSFlexGrid2.Col = 2
MSFlexGrid2.CellFontSize = 8
MSFlexGrid2.CellAlignment = 4
MSFlexGrid2.CellBackColor = &HC0FFC0
MSFlexGrid2.Text = "COMM.STAT"
'-----
```

```
MSFlexGrid3.Row = 0
MSFlexGrid3.Col = 0
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &H80FFFF
MSFlexGrid3.Text = "KUR-B/S"
```

```
MSFlexGrid3.Row = 1
MSFlexGrid3.Col = 0
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HFFFF80
MSFlexGrid3.Text = "DRN 6"
```

```
MSFlexGrid3.Row = 2
MSFlexGrid3.Col = 0
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HFFFF80
MSFlexGrid3.Text = "DRN 12"
```

```
MSFlexGrid3.Row = 3
MSFlexGrid3.Col = 0
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HFFFF80
MSFlexGrid3.Text = "KAW 04"
```

```
MSFlexGrid3.Row = 4  
MSFlexGrid3.Col = 0  
MSFlexGrid3.CellFontSize = 8  
MSFlexGrid3.CellAlignment = 4  
MSFlexGrid3.CellBackColor = &HFFFFFF80  
MSFlexGrid3.Text = "SAH RTN"
```

```
'-----
```

```
MSFlexGrid3.Row = 5  
MSFlexGrid3.Col = 0  
MSFlexGrid3.CellFontSize = 8  
MSFlexGrid3.CellAlignment = 4  
MSFlexGrid3.CellBackColor = &HFFFFFF80  
MSFlexGrid3.Text = "BAY 01"
```

```
MSFlexGrid3.Row = 6  
MSFlexGrid3.Col = 0  
MSFlexGrid3.CellFontSize = 8  
MSFlexGrid3.CellAlignment = 4  
MSFlexGrid3.CellBackColor = &HFFFFFF80  
MSFlexGrid3.Text = "BAY 08"
```

```
MSFlexGrid3.Row = 7  
MSFlexGrid3.Col = 0  
MSFlexGrid3.CellFontSize = 8  
MSFlexGrid3.CellAlignment = 4  
MSFlexGrid3.CellBackColor = &HFFFFFF80  
MSFlexGrid3.Text = "BEK 01"
```

```
MSFlexGrid3.Row = 8  
MSFlexGrid3.Col = 0  
MSFlexGrid3.CellFontSize = 8  
MSFlexGrid3.CellAlignment = 4  
MSFlexGrid3.CellBackColor = &HFFFFFF80
```

MSFlexGrid3.Text = "BEK 02"

MSFlexGrid3.Row = 9

MSFlexGrid3.Col = 0

MSFlexGrid3.CellFontSize = 8

MSFlexGrid3.CellAlignment = 4

MSFlexGrid3.CellBackColor = &HFFFFFF80

MSFlexGrid3.Text = "KUR 05"

MSFlexGrid3.Row = 10

MSFlexGrid3.Col = 0

MSFlexGrid3.CellFontSize = 8

MSFlexGrid3.CellAlignment = 4

MSFlexGrid3.CellBackColor = &HFFFFFF80

MSFlexGrid3.Text = "KUR 02"

MSFlexGrid3.Row = 11

MSFlexGrid3.Col = 0

MSFlexGrid3.CellFontSize = 8

MSFlexGrid3.CellAlignment = 4

MSFlexGrid3.CellBackColor = &HFFFFFF80

MSFlexGrid3.Text = "KUR 04"

MSFlexGrid3.Row = 12

MSFlexGrid3.Col = 0

MSFlexGrid3.CellFontSize = 8

MSFlexGrid3.CellAlignment = 4

MSFlexGrid3.CellBackColor = &HFFFFFF80

MSFlexGrid3.Text = "KUR 25"

MSFlexGrid3.Row = 13

MSFlexGrid3.Col = 0

MSFlexGrid3.CellFontSize = 8

MSFlexGrid3.CellAlignment = 4

MSFlexGrid3.CellBackColor = &HFFFF80
MSFlexGrid3.Text = "KUR 33"

MSFlexGrid3.Row = 14
MSFlexGrid3.Col = 0
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HFFFF80
MSFlexGrid3.Text = "KUR 40"

MSFlexGrid3.Row = 15
MSFlexGrid3.Col = 0
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HFFFF80
MSFlexGrid3.Text = "KUR 41"

MSFlexGrid3.Row = 16
MSFlexGrid3.Col = 0
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HFFFF80
MSFlexGrid3.Text = "KUR 42"

MSFlexGrid3.Row = 17
MSFlexGrid3.Col = 0
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HFFFF80
MSFlexGrid3.Text = "KUR 47"

MSFlexGrid3.Row = 18
MSFlexGrid3.Col = 0

```
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HFFFFFF80
MSFlexGrid3.Text = "KUR 49"
```

```
MSFlexGrid3.Row = 19
MSFlexGrid3.Col = 0
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HFFFFFF80
MSFlexGrid3.Text = "OUT PIR"
```

```
'-----
MSFlexGrid3.Row = 0
MSFlexGrid3.Col = 1
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HFFC0FF
MSFlexGrid3.Text = "RATE(bgpd)"
```

```
MSFlexGrid3.Row = 0
MSFlexGrid3.Col = 2
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HC0FFC0
MSFlexGrid3.Text = "COMM.STAT"
```

```
'-----
M = 0
'-----
End Sub
Private Sub Command5_Click()
M = 1
Text1.Text = M
```

Do Until 4 < 3

'=====

'KUR 41 :8/7" 2051.89

'KUR 40 :3/4" 2913.46

'KUR 04 :3/4" 3003.36

'KUR 47 :3/4" 2773.16

'KUR 33 :2" 53.90

'BAY 08 :3/4" 3050.12

'Adnan Gazan 2/2/06

'=====

'Reading flow rate of BARBES-DEEP

DoEvents

MSFlexGrid5.Row = 0

MSFlexGrid5.Col = 0

MSFlexGrid5.CellFontSize = 8

MSFlexGrid5.CellAlignment = 4

MSFlexGrid5.CellBackColor = &HC0&

MSFlexGrid5.Text = "BARBES-DEEP"

MSFlexGrid5.Row = 0

MSFlexGrid5.Col = 1

MSFlexGrid5.CellFontSize = 8

MSFlexGrid5.CellAlignment = 4

MSFlexGrid5.CellBackColor = &HFFFFFF

MSFlexGrid5.Text = "Scanning..."

WA = 5

DJ = 40

K = 0.7

K = 68.19

H = 2

Transmit CU, WA, DJ

'-----PROGRAM RESET


```

If N = 1 Then
MSFlexGrid5.Row = 0
MSFlexGrid5.Col = 1
MSFlexGrid5.CellFontSize = 8
MSFlexGrid5.CellAlignment = 4
MSFlexGrid5.CellBackColor = &HFFFFFFF
MSFlexGrid5.Text = "reset prg"
End
End If

```

```
'-----
```

```

DoEvents
Receive DIGIT
If N = 1 Then 'okuma islemi olumlu

```

```
DATA = (DGT3 * 16 ^ 3 + DGT2 * 16 ^ 2 + DGT1 * 16 + DGT0)
```

```
'-----
```

```

MSFlexGrid5.Row = 0
MSFlexGrid5.Col = 1
MSFlexGrid5.CellFontSize = 8
MSFlexGrid5.CellAlignment = 4
MSFlexGrid5.CellBackColor = &HFFFFFFF
MSFlexGrid5.Text = DATA

```

```
'-----
```

```
'-----
```

```

'Data1.Recordset.AddNew
Text2.Text = DATA
Text3.Text = Time
Text4.Text = Date
Text5.Text = "BARBES-DEEP"

```

```
'-----
```

```
*****
```

```
****
```

```

Else 'okuma islemi olumsuz
    Transmit CU, WA, DJ
    '-----PROGRAM RESET
    If N = 1 Then
    MSFlexGrid5.Row = 0
    MSFlexGrid5.Col = 1
    MSFlexGrid5.CellFontSize = 8
    MSFlexGrid5.CellAlignment = 4
    MSFlexGrid5.CellBackColor = &HFFFFFF
    MSFlexGrid5.Text = "reset prg"
    End
    End If
    '-----
    DoEvents
    Receive DIGIT
    If N = 1 Then 'okuma islemi olumlu

    DATA = (DGT3 * 16 ^ 3 + DGT2 * 16 ^ 2 + DGT1 * 16 + DGT0)
    '-----
    MSFlexGrid5.Row = 0
    MSFlexGrid5.Col = 1
    MSFlexGrid5.CellFontSize = 8
    MSFlexGrid5.CellAlignment = 4
    MSFlexGrid5.CellBackColor = &HFFFFFF
    MSFlexGrid5.Text = DATA
    '-----

    '-----
    Data1.Recordset.AddNew
    Text2.Text = DATA
    Text3.Text = Time
    Text4.Text = Date
    Text5.Text = "BARBES-DEEP"
    '-----

```

```

Else 'okuma islemi olumsuz
MSFlexGrid5.Row = 0
MSFlexGrid5.Col = 1
MSFlexGrid5.CellFontSize = 8
MSFlexGrid5.CellAlignment = 4
MSFlexGrid5.CellBackColor = &HFFFFFFF
'MSFlexGrid1.Text = average
MSFlexGrid5.Text = "No Signal"
End If

'*****
****

End If
DoEvents
TIME_INIT = Time
TIME_FINAL = Time
ECE = #12:01:00 AM#
Do Until (#12:00:05 AM# - #12:00:00 AM#) < (TIME_FINAL - TIME_INIT)
TIME_FINAL = Time
DoEvents
Loop
MSFlexGrid5.Row = 0
MSFlexGrid5.Col = 0
MSFlexGrid5.CellFontSize = 8
MSFlexGrid5.CellAlignment = 4
MSFlexGrid5.CellBackColor = &HFFFF80
MSFlexGrid5.Text = "BARBES-DEEP"
M = M + 1
Text1.Text = M

Loop
'=====
'Reading flow rate of PIR IN

```

DoEvents

MSFlexGrid4.Row = 1

MSFlexGrid4.Col = 0

MSFlexGrid4.CellFontSize = 8

MSFlexGrid4.CellAlignment = 4

MSFlexGrid4.CellBackColor = &HC0&

MSFlexGrid4.Text = "PIR IN"

MSFlexGrid4.Row = 1

MSFlexGrid4.Col = 2

MSFlexGrid4.CellFontSize = 8

MSFlexGrid4.CellAlignment = 4

MSFlexGrid4.CellBackColor = &HFFFFFF

'MSFlexGrid1.Text = average

MSFlexGrid4.Text = "Scanning..."

WA = 1

DJ = 0

K = 0.7

'K = 68.19

H = 2

Transmit CU, WA, DJ

'-----PROGRAM RESET

If N = 1 Then

MSFlexGrid4.Row = 1

MSFlexGrid4.Col = 1

MSFlexGrid4.CellFontSize = 8

MSFlexGrid4.CellAlignment = 4

MSFlexGrid4.CellBackColor = &HFFFFFF

MSFlexGrid4.Text = "reset prg"

End

End If

'-----

```

DoEvents
Receive DIGIT
If N = 1 Then 'okuma islemi olumlu

DATA = (DGT3 * 16 ^ 3 + DGT2 * 16 ^ 2 + DGT1 * 16 + DGT0)
If Form2.Label3 = "PIR IN" Then
Form2.MSFlexGrid1.Row = 1
Form2.MSFlexGrid1.Col = 1
Form2.MSFlexGrid1.Text = DATA * 2
Form2.MSFlexGrid1.Row = 1
Form2.MSFlexGrid1.Col = 0
Form2.MSFlexGrid1.Text = DATA * 2 * K
End If
DoEvents
'-----
MSFlexGrid4.Row = 1
MSFlexGrid4.Col = 1
MSFlexGrid4.CellFontSize = 8
MSFlexGrid4.CellAlignment = 4
MSFlexGrid4.CellBackColor = &HFFFFFF
MSFlexGrid4.Text = DATA * 2 * K
DoEvents
'-----
MSFlexGrid4.Row = 1
MSFlexGrid4.Col = 2
MSFlexGrid4.CellFontSize = 8
MSFlexGrid4.CellAlignment = 4
MSFlexGrid4.CellBackColor = &HFFFFFF
'MSFlexGrid1.Text = average
MSFlexGrid4.Text = DATA * 2
'-----
'Data1.Recordset.AddNew
Text2.Text = DATA * 2
Text3.Text = Time

```

```
Text4.Text = Date
```

```
Text5.Text = "PIR IN"
```

```
'-----
```

```
*****
```

```
****
```

```
Else 'okuma islemi olumsuz
```

```
    Transmit CU, WA, DJ
```

```
    '-----PROGRAM RESET
```

```
    If N = 1 Then
```

```
        MSFlexGrid4.Row = 1
```

```
        MSFlexGrid4.Col = 1
```

```
        MSFlexGrid4.CellFontSize = 8
```

```
        MSFlexGrid4.CellAlignment = 4
```

```
        MSFlexGrid4.CellBackColor = &HFFFFFF
```

```
        MSFlexGrid4.Text = "reset prg"
```

```
    End
```

```
    End If
```

```
'-----
```

```
    DoEvents
```

```
    Receive DIGIT
```

```
    If N = 1 Then 'okuma islemi olumlu
```

```
        DATA = (DGT3 * 16 ^ 3 + DGT2 * 16 ^ 2 + DGT1 * 16 + DGT0)
```

```
        If Form2.Label3 = "PIR IN" Then
```

```
            Form2.MSFlexGrid1.Row = 1
```

```
            Form2.MSFlexGrid1.Col = 1
```

```
            Form2.MSFlexGrid1.Text = DATA * 2
```

```
            Form2.MSFlexGrid1.Row = 1
```

```
            Form2.MSFlexGrid1.Col = 0
```

```
            Form2.MSFlexGrid1.Text = DATA * 2 * K
```

```
        End If
```

```
        DoEvents
```

```
'-----
```

```
MSFlexGrid4.Row = 1
```

```

MSFlexGrid4.Col = 1
MSFlexGrid4.CellFontSize = 8
MSFlexGrid4.CellAlignment = 4
MSFlexGrid4.CellBackColor = &HFFFFFFF
MSFlexGrid4.Text = DATA * 2 * K
DoEvents
'-----
MSFlexGrid4.Row = 1
MSFlexGrid4.Col = 2
MSFlexGrid4.CellFontSize = 8
MSFlexGrid4.CellAlignment = 4
MSFlexGrid4.CellBackColor = &HFFFFFFF
'MSFlexGrid1.Text = average
'MSFlexGrid4.Text = DATA * 2
MSFlexGrid4.Text = DATA
'-----
Data1.Recordset.AddNew
Text2.Text = DATA * 2
Text3.Text = Time
Text4.Text = Date
Text5.Text = "PIR IN"
'-----
Else 'okuma islemi olumsuz
MSFlexGrid4.Row = 1
MSFlexGrid4.Col = 2
MSFlexGrid4.CellFontSize = 8
MSFlexGrid4.CellAlignment = 4
MSFlexGrid4.CellBackColor = &HFFFFFFF
'MSFlexGrid1.Text = average
MSFlexGrid4.Text = "No Signal"
End If

```

```

*****

```

```

****

```

```

End If

```

```

DoEvents
TIME_INIT = Time
TIME_FINAL = Time
ECE = #12:01:00 AM#
Do Until (#12:00:05 AM# - #12:00:00 AM#) < (TIME_FINAL - TIME_INIT)
TIME_FINAL = Time
DoEvents
Loop
MSFlexGrid4.Row = 1
MSFlexGrid4.Col = 0
MSFlexGrid4.CellFontSize = 8
MSFlexGrid4.CellAlignment = 4
MSFlexGrid4.CellBackColor = &HFFFF80
MSFlexGrid4.Text = "PIR IN"
M = M + 1
Text1.Text = M

```

'Reading flow rate of BAY 08

```

DoEvents
MSFlexGrid3.Row = 6
MSFlexGrid3.Col = 0
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HC0&
MSFlexGrid3.Text = "BAY 08"

MSFlexGrid3.Row = 6
MSFlexGrid3.Col = 2
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HFFFFFF
'MSFlexGrid3.Text = average
MSFlexGrid3.Text = "Scanning..."

```



```

WA = 2
DJ = 4
K = 0.98
Transmit CU, WA, DJ
'-----PROGRAM RESET
If N = 1 Then
MSFlexGrid3.Row = 6
MSFlexGrid3.Col = 1
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HFFFFFF
MSFlexGrid3.Text = "reset prg"
End
End If
'-----
DoEvents
Receive DIGIT
DATA = (DGT3 * 16 ^ 3 + DGT2 * 16 ^ 2 + DGT1 * 16 + DGT0)
If N = 1 Then
If Form2.Label3 = "BAY 08" Then
Form2.MSFlexGrid1.Row = 1
Form2.MSFlexGrid1.Col = 1
Form2.MSFlexGrid1.Text = DATA * 2
Form2.MSFlexGrid1.Row = 1
Form2.MSFlexGrid1.Col = 0
Form2.MSFlexGrid1.Text = DATA * 2 * K
End If
DoEvents
'-----
MSFlexGrid3.Row = 6
MSFlexGrid3.Col = 1
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4

```

```

MSFlexGrid3.CellBackColor = &HFFFFFFF
MSFlexGrid3.Text = DATA * 2 * K
DoEvents
'-----
MSFlexGrid3.Row = 6
MSFlexGrid3.Col = 2
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HFFFFFFF
'MSFlexGrid3.Text = average
MSFlexGrid3.Text = DATA * 2
'-----

Data1.Recordset.AddNew
Text2.Text = DATA * 2
Text3.Text = Time
Text4.Text = Date
Text5.Text = "BAY 08"
'-----

Else 'okuma islemi olumsuz
    Transmit CU, WA, DJ
    '-----PROGRAM RESET
    If N = 1 Then
        MSFlexGrid3.Row = 6
        MSFlexGrid3.Col = 1
        MSFlexGrid3.CellFontSize = 8
        MSFlexGrid3.CellAlignment = 4
        MSFlexGrid3.CellBackColor = &HFFFFFFF
        MSFlexGrid3.Text = "reset prg"
    End
    End If
    '-----
    DoEvents
    Receive DIGIT
    DATA = (DGT3 * 16 ^ 3 + DGT2 * 16 ^ 2 + DGT1 * 16 + DGT0)

```

```

If N = 1 Then
If Form2.Label3 = "BAY 08" Then
Form2.MSFlexGrid1.Row = 1
Form2.MSFlexGrid1.Col = 1
Form2.MSFlexGrid1.Text = DATA * 2
Form2.MSFlexGrid1.Row = 1
Form2.MSFlexGrid1.Col = 0
Form2.MSFlexGrid1.Text = DATA * 2 * K
End If
DoEvents
'-----
MSFlexGrid3.Row = 6
MSFlexGrid3.Col = 1
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HFFFFFF
MSFlexGrid3.Text = DATA * 2 * K
DoEvents
'-----
MSFlexGrid3.Row = 6
MSFlexGrid3.Col = 2
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HFFFFFF
'MSFlexGrid3.Text = average
MSFlexGrid3.Text = DATA * 2
'-----
'Data1.Recordset.AddNew
Text2.Text = DATA * 2
Text3.Text = Time
Text4.Text = Date
Text5.Text = "BAY 08"
'-----
Else 'okuma islemi olumsuz

```

```

MSFlexGrid3.Row = 6
MSFlexGrid3.Col = 2
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HFFFFFF
'MSFlexGrid3.Text = average
MSFlexGrid3.Text = "No signal"
End If

```

```
End If
```

```
DoEvents
```

```
TIME_INIT = Time
```

```
TIME_FINAL = Time
```

```
ECE = #12:01:00 AM#
```

```
Do Until (#12:00:05 AM# - #12:00:00 AM#) < (TIME_FINAL - TIME_INIT)
```

```
TIME_FINAL = Time
```

```
DoEvents
```

```
Loop
```

```
MSFlexGrid3.Row = 6
```

```
MSFlexGrid3.Col = 0
```

```
MSFlexGrid3.CellFontSize = 8
```

```
MSFlexGrid3.CellAlignment = 4
```

```
MSFlexGrid3.CellBackColor = &HFFFF80
```

```
MSFlexGrid3.Text = "BAY 08"
```

```
M = M + 1
```

```
Text1.Text = M
```

```
'=====
```

```
'Reading flow rate of BEK 01
```

```
DoEvents
```

```
MSFlexGrid3.Row = 7
```

```
MSFlexGrid3.Col = 0
```

```
MSFlexGrid3.CellFontSize = 8
```

```
MSFlexGrid3.CellAlignment = 4
```

```
MSFlexGrid3.CellBackColor = &HC0&
```

```
MSFlexGrid3.Text = "BEK 01"
```

```
MSFlexGrid3.Row = 7
```

```
MSFlexGrid3.Col = 2
```

```
MSFlexGrid3.CellFontSize = 8
```

```
MSFlexGrid3.CellAlignment = 4
```

```
MSFlexGrid3.CellBackColor = &HFFFFFFF
```

```
'MSFlexGrid3.Text = average
```

```
MSFlexGrid3.Text = "Scanning..."
```

```
WA = 2
```

```
DJ = 6
```

```
K = 6.2
```

```
Transmit CU, WA, DJ
```

```
'-----PROGRAM RESET
```

```
If N = 1 Then
```

```
MSFlexGrid3.Row = 7
```

```
MSFlexGrid3.Col = 1
```

```
MSFlexGrid3.CellFontSize = 8
```

```
MSFlexGrid3.CellAlignment = 4
```

```
MSFlexGrid3.CellBackColor = &HFFFFFFF
```

```
MSFlexGrid3.Text = "reset prg"
```

```
End
```

```
End If
```

```
'-----
```

```
DoEvents
```

```
Receive DIGIT
```

```
DATA = (DGT3 * 16 ^ 3 + DGT2 * 16 ^ 2 + DGT1 * 16 + DGT0)
```

```
If N = 1 Then
```

```
If Form2.Label3 = "BEK 01" Then
```

```
Form2.MSFlexGrid1.Row = 1
```

```
Form2.MSFlexGrid1.Col = 1
```

```
Form2.MSFlexGrid1.Text = DATA * 2
```

```
Form2.MSFlexGrid1.Row = 1
```

```

Form2.MSFlexGrid1.Col = 0
Form2.MSFlexGrid1.Text = DATA * 2 * K
End If
DoEvents
'-----
MSFlexGrid3.Row = 7
MSFlexGrid3.Col = 1
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HFFFFFF
MSFlexGrid3.Text = DATA * 2 * K
DoEvents
'-----
MSFlexGrid3.Row = 7
MSFlexGrid3.Col = 2
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HFFFFFF
'MSFlexGrid3.Text = average
MSFlexGrid3.Text = DATA * 2
'-----
Data1.Recordset.AddNew
Text2.Text = DATA * 2
Text3.Text = Time
Text4.Text = Date
Text5.Text = "BEK 01"
'-----
Else 'Okuma islemi olumsuz
    Transmit CU, WA, DJ
    '-----PROGRAM RESET
    If N = 1 Then
        MSFlexGrid3.Row = 7
        MSFlexGrid3.Col = 1
        MSFlexGrid3.CellFontSize = 8

```

```

MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HFFFFFF
MSFlexGrid3.Text = "reset prg"
End
End If
'-----
DoEvents
Receive DIGIT
DATA = (DGT3 * 16 ^ 3 + DGT2 * 16 ^ 2 + DGT1 * 16 + DGT0)
If N = 1 Then
If Form2.Label3 = "BEK 01" Then
Form2.MSFlexGrid1.Row = 1
Form2.MSFlexGrid1.Col = 1
Form2.MSFlexGrid1.Text = DATA * 2
Form2.MSFlexGrid1.Row = 1
Form2.MSFlexGrid1.Col = 0
Form2.MSFlexGrid1.Text = DATA * 2 * K
End If
DoEvents
'-----
MSFlexGrid3.Row = 7
MSFlexGrid3.Col = 1
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HFFFFFF
MSFlexGrid3.Text = DATA * 2 * K
DoEvents
'-----
MSFlexGrid3.Row = 7
MSFlexGrid3.Col = 2
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HFFFFFF
'MSFlexGrid3.Text = average

```

```

MSFlexGrid3.Text = DATA * 2
'-----
Data1.Recordset.AddNew
Text2.Text = DATA * 2
Text3.Text = Time
Text4.Text = Date
Text5.Text = "BEK 01"
Else

MSFlexGrid3.Row = 7
MSFlexGrid3.Col = 2
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HFFFFFF
'MSFlexGrid3.Text = average
MSFlexGrid3.Text = "No Signal"
End If

End If
DoEvents
TIME_INIT = Time
TIME_FINAL = Time
ECE = #12:01:00 AM#
Do Until (#12:00:05 AM# - #12:00:00 AM#) < (TIME_FINAL - TIME_INIT)
TIME_FINAL = Time
DoEvents
Loop
MSFlexGrid3.Row = 7
MSFlexGrid3.Col = 0
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HFFF80
MSFlexGrid3.Text = "BEK 01"
M = M + 1
Text1.Text = M

```



```

'-----
'Reading flow rate of KUR 05
DoEvents
MSFlexGrid3.Row = 9
MSFlexGrid3.Col = 0
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HC0&
MSFlexGrid3.Text = "KUR 05"

MSFlexGrid3.Row = 9
MSFlexGrid3.Col = 2
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HFFFFFF
'MSFlexGrid3.Text = average
MSFlexGrid3.Text = "Scanning..."

WA = 2
DJ = 7
K = 6.1
Transmit CU, WA, DJ
'-----PROGRAM RESET
If N = 1 Then
MSFlexGrid3.Row = 9
MSFlexGrid3.Col = 1
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HFFFFFF
MSFlexGrid3.Text = "reset prg"
End
End If
'-----
DoEvents

```

Receive DIGIT

DATA = (DGT3 * 16 ^ 3 + DGT2 * 16 ^ 2 + DGT1 * 16 + DGT0)

If N = 1 Then

If Form2.Label3 = "KUR 05" Then

Form2.MSFlexGrid1.Row = 1

Form2.MSFlexGrid1.Col = 1

Form2.MSFlexGrid1.Text = DATA * 2

Form2.MSFlexGrid1.Row = 1

Form2.MSFlexGrid1.Col = 0

Form2.MSFlexGrid1.Text = DATA * 2 * K

End If

DoEvents

'-----

MSFlexGrid3.Row = 9

MSFlexGrid3.Col = 1

MSFlexGrid3.CellFontSize = 8

MSFlexGrid3.CellAlignment = 4

MSFlexGrid3.CellBackColor = &HFFFFFFF

MSFlexGrid3.Text = DATA * 2 * K

DoEvents

'-----

MSFlexGrid3.Row = 9

MSFlexGrid3.Col = 2

MSFlexGrid3.CellFontSize = 8

MSFlexGrid3.CellAlignment = 4

MSFlexGrid3.CellBackColor = &HFFFFFFF

'MSFlexGrid3.Text = average

MSFlexGrid3.Text = DATA * 2

'-----

Data1.Recordset.AddNew

Text2.Text = DATA * 2

Text3.Text = Time

Text4.Text = Date

Text5.Text = "KUR 05"

```

'-----
Else 'Okuma islemi olumsuz
    Transmit CU, WA, DJ
    '-----PROGRAM RESET
    If N = 1 Then
    MSFlexGrid3.Row = 9
    MSFlexGrid3.Col = 1
    MSFlexGrid3.CellFontSize = 8
    MSFlexGrid3.CellAlignment = 4
    MSFlexGrid3.CellBackColor = &HFFFFFFF
    MSFlexGrid3.Text = "reset prg"
    End
    End If

'-----

DoEvents
Receive DIGIT
DATA = (DGT3 * 16 ^ 3 + DGT2 * 16 ^ 2 + DGT1 * 16 + DGT0)
If N = 1 Then
If Form2.Label3 = "KUR 05" Then
Form2.MSFlexGrid1.Row = 1
Form2.MSFlexGrid1.Col = 1
Form2.MSFlexGrid1.Text = DATA * 2
Form2.MSFlexGrid1.Row = 1
Form2.MSFlexGrid1.Col = 0
Form2.MSFlexGrid1.Text = DATA * 2 * K
End If
DoEvents
'-----

MSFlexGrid3.Row = 9
MSFlexGrid3.Col = 1
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HFFFFFFF

```

```

MSFlexGrid3.Text = DATA * 2 * K
DoEvents
'-----
MSFlexGrid3.Row = 9
MSFlexGrid3.Col = 2
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HFFFFFF
'MSFlexGrid3.Text = average
MSFlexGrid3.Text = DATA * 2
'-----

Data1.Recordset.AddNew
Text2.Text = DATA * 2
Text3.Text = Time
Text4.Text = Date
Text5.Text = "KUR 05"
'-----

Else 'Okuma islemi olumsuz

MSFlexGrid3.Row = 9
MSFlexGrid3.Col = 2
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HFFFFFF
'MSFlexGrid3.Text = average
MSFlexGrid3.Text = "No Signal"
End If

End If
DoEvents
TIME_INIT = Time
TIME_FINAL = Time
ECE = #12:01:00 AM#
Do Until (#12:00:05 AM# - #12:00:00 AM#) < (TIME_FINAL - TIME_INIT)
TIME_FINAL = Time

```

DoEvents

Loop

MSFlexGrid3.Row = 9

MSFlexGrid3.Col = 0

MSFlexGrid3.CellFontSize = 8

MSFlexGrid3.CellAlignment = 4

MSFlexGrid3.CellBackColor = &HFFFF80

MSFlexGrid3.Text = "KUR 05"

M = M + 1

Text1.Text = M

'=====

'Reading flow rate of KUR 02

DoEvents

MSFlexGrid3.Row = 10

MSFlexGrid3.Col = 0

MSFlexGrid3.CellFontSize = 8

MSFlexGrid3.CellAlignment = 4

MSFlexGrid3.CellBackColor = &HC0&

MSFlexGrid3.Text = "KUR 02"

MSFlexGrid3.Row = 10

MSFlexGrid3.Col = 2

MSFlexGrid3.CellFontSize = 8

MSFlexGrid3.CellAlignment = 4

MSFlexGrid3.CellBackColor = &HFFFFFF

'MSFlexGrid3.Text = average

MSFlexGrid3.Text = "Scanning..."

WA = 2

DJ = 9

K = 6.3

Transmit CU, WA, DJ

'-----PROGRAM RESET

If N = 1 Then

```

MSFlexGrid3.Row = 10
MSFlexGrid3.Col = 1
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HFFFFFFF
MSFlexGrid3.Text = "reset prg"
End
End If
'-----

DoEvents
Receive DIGIT
DATA = (DGT3 * 16 ^ 3 + DGT2 * 16 ^ 2 + DGT1 * 16 + DGT0)
If N = 1 Then
If Form2.Label3 = "KUR 02" Then
Form2.MSFlexGrid1.Row = 1
Form2.MSFlexGrid1.Col = 1
Form2.MSFlexGrid1.Text = DATA * 2
Form2.MSFlexGrid1.Row = 1
Form2.MSFlexGrid1.Col = 0
Form2.MSFlexGrid1.Text = DATA * 2 * K
End If
DoEvents
'-----

MSFlexGrid3.Row = 10
MSFlexGrid3.Col = 1
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HFFFFFFF
MSFlexGrid3.Text = DATA * 2 * K
DoEvents
'-----

MSFlexGrid3.Row = 10
MSFlexGrid3.Col = 2
MSFlexGrid3.CellFontSize = 8

```

```

MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HFFFFFF
'MSFlexGrid3.Text = average
MSFlexGrid3.Text = DATA * 2
'-----

Data1.Recordset.AddNew
Text2.Text = DATA * 2
Text3.Text = Time
Text4.Text = Date
Text5.Text = "KUR 02"
'-----

Else   'Okuma islemi olumsuz
        Transmit CU, WA, DJ
        '-----PROGRAM RESET
        If N = 1 Then
            MSFlexGrid3.Row = 10
            MSFlexGrid3.Col = 1
            MSFlexGrid3.CellFontSize = 8
            MSFlexGrid3.CellAlignment = 4
            MSFlexGrid3.CellBackColor = &HFFFFFF
            MSFlexGrid3.Text = "reset prg"
        End
        End If
        '-----

        DoEvents
        Receive DIGIT
        DATA = (DGT3 * 16 ^ 3 + DGT2 * 16 ^ 2 + DGT1 * 16 + DGT0)
        If N = 1 Then
            If Form2.Label3 = "KUR 02" Then
                Form2.MSFlexGrid1.Row = 1
                Form2.MSFlexGrid1.Col = 1
                Form2.MSFlexGrid1.Text = DATA * 2
                Form2.MSFlexGrid1.Row = 1
                Form2.MSFlexGrid1.Col = 0
            End If
        End If
    End If
End Sub

```

```

Form2.MSFlexGrid1.Text = DATA * 2 * K
End If
DoEvents
'-----
MSFlexGrid3.Row = 10
MSFlexGrid3.Col = 1
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HFFFFFF
MSFlexGrid3.Text = DATA * 2 * K
DoEvents
'-----
MSFlexGrid3.Row = 10
MSFlexGrid3.Col = 2
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HFFFFFF
'MSFlexGrid3.Text = average
MSFlexGrid3.Text = DATA * 2
'-----
Data1.Recordset.AddNew
Text2.Text = DATA * 2
Text3.Text = Time
Text4.Text = Date
Text5.Text = "KUR 02"
'-----
Else   'Okuma islemi olumsuz
MSFlexGrid3.Row = 10
MSFlexGrid3.Col = 2
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HFFFFFF
'MSFlexGrid3.Text = average
MSFlexGrid3.Text = "No Signal"

```



```

        End If
    End If
    DoEvents
    TIME_INIT = Time
    TIME_FINAL = Time
    ECE = #12:01:00 AM#
    Do Until (#12:00:05 AM# - #12:00:00 AM#) < (TIME_FINAL - TIME_INIT)
    TIME_FINAL = Time
    DoEvents
    Loop
    MSFlexGrid3.Row = 10
    MSFlexGrid3.Col = 0
    MSFlexGrid3.CellFontSize = 8
    MSFlexGrid3.CellAlignment = 4
    MSFlexGrid3.CellBackColor = &HFFFF80
    MSFlexGrid3.Text = "KUR 02"
    M = M + 1
    Text1.Text = M
    '=====
    'Reading flow rate of KUR 04
    DoEvents
    MSFlexGrid3.Row = 11
    MSFlexGrid3.Col = 0
    MSFlexGrid3.CellFontSize = 8
    MSFlexGrid3.CellAlignment = 4
    MSFlexGrid3.CellBackColor = &HC0&
    MSFlexGrid3.Text = "KUR 04"

    MSFlexGrid3.Row = 11
    MSFlexGrid3.Col = 2
    MSFlexGrid3.CellFontSize = 8
    MSFlexGrid3.CellAlignment = 4
    MSFlexGrid3.CellBackColor = &HFFFFFFF
    'MSFlexGrid3.Text = average

```

```

MSFlexGrid3.Text = "Scanning..."

WA = 2
DJ = 8
K = 2.2
Transmit CU, WA, DJ
'-----PROGRAM RESET
If N = 1 Then
MSFlexGrid3.Row = 11
MSFlexGrid3.Col = 1
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HFFFFFF
MSFlexGrid3.Text = "reset prg"
End
End If
'-----
DoEvents
Receive DIGIT
DATA = (DGT3 * 16 ^ 3 + DGT2 * 16 ^ 2 + DGT1 * 16 + DGT0)
If N = 1 Then
If Form2.Label3 = "KUR 04" Then
Form2.MSFlexGrid1.Row = 1
Form2.MSFlexGrid1.Col = 1
Form2.MSFlexGrid1.Text = DATA * 2
Form2.MSFlexGrid1.Row = 1
Form2.MSFlexGrid1.Col = 0
Form2.MSFlexGrid1.Text = DATA * 2 * K
End If
DoEvents
'-----
MSFlexGrid3.Row = 11
MSFlexGrid3.Col = 1
MSFlexGrid3.CellFontSize = 8

```

```

MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HFFFFFFF
MSFlexGrid3.Text = DATA * 2 * K
DoEvents
'-----
MSFlexGrid3.Row = 11
MSFlexGrid3.Col = 2
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HFFFFFFF
'MSFlexGrid3.Text = average
MSFlexGrid3.Text = DATA * 2
'-----
Data1.Recordset.AddNew
Text2.Text = DATA * 2
Text3.Text = Time
Text4.Text = Date
Text5.Text = "KUR 04"
'-----
Else    'Okuma islemi olumsuz
        Transmit CU, WA, DJ
        '-----PROGRAM RESET
        If N = 1 Then
            MSFlexGrid3.Row = 11
            MSFlexGrid3.Col = 1
            MSFlexGrid3.CellFontSize = 8
            MSFlexGrid3.CellAlignment = 4
            MSFlexGrid3.CellBackColor = &HFFFFFFF
            MSFlexGrid3.Text = "reset prg"
        End
        End If
        '-----
        DoEvents
        Receive DIGIT

```

DATA = (DGT3 * 16 ^ 3 + DGT2 * 16 ^ 2 + DGT1 * 16 + DGT0)

If N = 1 Then

If Form2.Label3 = "KUR 04" Then

Form2.MSFlexGrid1.Row = 1

Form2.MSFlexGrid1.Col = 1

Form2.MSFlexGrid1.Text = DATA * 2

Form2.MSFlexGrid1.Row = 1

Form2.MSFlexGrid1.Col = 0

Form2.MSFlexGrid1.Text = DATA * 2 * K

End If

DoEvents

'-----

MSFlexGrid3.Row = 11

MSFlexGrid3.Col = 1

MSFlexGrid3.CellFontSize = 8

MSFlexGrid3.CellAlignment = 4

MSFlexGrid3.CellBackColor = &HFFFFFF

MSFlexGrid3.Text = DATA * 2 * K

DoEvents

'-----

MSFlexGrid3.Row = 11

MSFlexGrid3.Col = 2

MSFlexGrid3.CellFontSize = 8

MSFlexGrid3.CellAlignment = 4

MSFlexGrid3.CellBackColor = &HFFFFFF

'MSFlexGrid3.Text = average

MSFlexGrid3.Text = DATA * 2

'-----

Data1.Recordset.AddNew

Text2.Text = DATA * 2

Text3.Text = Time

Text4.Text = Date

Text5.Text = "KUR 04"

'-----

```

Else      'Okuma islemi olumsuz

MSFlexGrid3.Row = 11
MSFlexGrid3.Col = 2
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HFFFFFFF
'MSFlexGrid3.Text = average
MSFlexGrid3.Text = "No Signal"
End If

End If
DoEvents
TIME_INIT = Time
TIME_FINAL = Time
ECE = #12:01:00 AM#
Do Until (#12:00:05 AM# - #12:00:00 AM#) < (TIME_FINAL - TIME_INIT)
TIME_FINAL = Time
DoEvents
Loop
MSFlexGrid3.Row = 11
MSFlexGrid3.Col = 0
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HFFFF80
MSFlexGrid3.Text = "KUR 04"
M = M + 1
Text1.Text = M
'=====
'Reading flow rate of KUR 25
DoEvents
MSFlexGrid3.Row = 12
MSFlexGrid3.Col = 0
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4

```

```
MSFlexGrid3.CellBackColor = &HC0&
MSFlexGrid3.Text = "KUR 25"
```

```
MSFlexGrid3.Row = 12
MSFlexGrid3.Col = 2
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HFFFFFFF
'MSFlexGrid3.Text = average
MSFlexGrid3.Text = "Scanning..."
```

```
WA = 2
DJ = 3
K = 0.15
Transmit CU, WA, DJ
'-----PROGRAM RESET
If N = 1 Then
MSFlexGrid3.Row = 12
MSFlexGrid3.Col = 1
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HFFFFFFF
MSFlexGrid3.Text = "reset prg"
End
End If
```

```
'-----
DoEvents
Receive DIGIT
DATA = (DGT3 * 16 ^ 3 + DGT2 * 16 ^ 2 + DGT1 * 16 + DGT0)
If N = 1 Then
If Form2.Label3 = "KUR 25" Then
Form2.MSFlexGrid1.Row = 1
Form2.MSFlexGrid1.Col = 1
```

```

Form2.MSFlexGrid1.Text = DATA * 2
Form2.MSFlexGrid1.Row = 1
Form2.MSFlexGrid1.Col = 0
Form2.MSFlexGrid1.Text = DATA * 2 * K
End If
DoEvents
'-----
MSFlexGrid3.Row = 12
MSFlexGrid3.Col = 1
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HFFFFFF
MSFlexGrid3.Text = DATA * 2 * K
DoEvents
'-----
MSFlexGrid3.Row = 12
MSFlexGrid3.Col = 2
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HFFFFFF
'MSFlexGrid3.Text = average
MSFlexGrid3.Text = DATA * 2
'-----
Data1.Recordset.AddNew
Text2.Text = DATA * 2
Text3.Text = Time
Text4.Text = Date
Text5.Text = "KUR 25"
'-----
Else    'Okuma islemi olumsuz
        Transmit CU, WA, DJ
        '-----PROGRAM RESET
        If N = 1 Then
            MSFlexGrid3.Row = 12

```

```

MSFlexGrid3.Col = 1
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HFFFFFFF
MSFlexGrid3.Text = "reset prg"
End
End If
'-----
DoEvents
Receive DIGIT
DATA = (DGT3 * 16 ^ 3 + DGT2 * 16 ^ 2 + DGT1 * 16 + DGT0)
If N = 1 Then
If Form2.Label3 = "KUR 25" Then
Form2.MSFlexGrid1.Row = 1
Form2.MSFlexGrid1.Col = 1
Form2.MSFlexGrid1.Text = DATA * 2
Form2.MSFlexGrid1.Row = 1
Form2.MSFlexGrid1.Col = 0
Form2.MSFlexGrid1.Text = DATA * 2 * K
End If
DoEvents
'-----
MSFlexGrid3.Row = 12
MSFlexGrid3.Col = 1
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HFFFFFFF
MSFlexGrid3.Text = DATA * 2 * K
DoEvents
'-----
MSFlexGrid3.Row = 12
MSFlexGrid3.Col = 2
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4

```



```

MSFlexGrid3.CellBackColor = &HFFFFFF
'MSFlexGrid3.Text = average
MSFlexGrid3.Text = DATA * 2
'-----
Data1.Recordset.AddNew
Text2.Text = DATA * 2
Text3.Text = Time
Text4.Text = Date
Text5.Text = "KUR 25"
'-----
Else      'Okuma islemi olumsuz

MSFlexGrid3.Row = 12
MSFlexGrid3.Col = 2
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HFFFFFF
'MSFlexGrid3.Text = average
MSFlexGrid3.Text = "No Signal"
End If

End If
DoEvents
TIME_INIT = Time
TIME_FINAL = Time
ECE = #12:01:00 AM#
Do Until (#12:00:05 AM# - #12:00:00 AM#) < (TIME_FINAL - TIME_INIT)
TIME_FINAL = Time
DoEvents
Loop
MSFlexGrid3.Row = 12
MSFlexGrid3.Col = 0
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HFFF80

```

```
MSFlexGrid3.Text = "KUR 25"
```

```
M = M + 1
```

```
Text1.Text = M
```

```
'=====
```

```
'Reading flow rate of KUR 33
```

```
DoEvents
```

```
MSFlexGrid3.Row = 13
```

```
MSFlexGrid3.Col = 0
```

```
MSFlexGrid3.CellFontSize = 8
```

```
MSFlexGrid3.CellAlignment = 4
```

```
MSFlexGrid3.CellBackColor = &HC0&
```

```
MSFlexGrid3.Text = "KUR 33"
```

```
MSFlexGrid3.Row = 13
```

```
MSFlexGrid3.Col = 2
```

```
MSFlexGrid3.CellFontSize = 8
```

```
MSFlexGrid3.CellAlignment = 4
```

```
MSFlexGrid3.CellBackColor = &HFFFFFF
```

```
'MSFlexGrid3.Text = average
```

```
MSFlexGrid3.Text = "Scanning..."
```

```
WA = 2
```

```
DJ = 5
```

```
K = 38.4
```

```
Transmit CU, WA, DJ
```

```
'-----PROGRAM RESET
```

```
If N = 1 Then
```

```
MSFlexGrid3.Row = 13
```

```
MSFlexGrid3.Col = 1
```

```
MSFlexGrid3.CellFontSize = 8
```

```
MSFlexGrid3.CellAlignment = 4
```

```
MSFlexGrid3.CellBackColor = &HFFFFFF
```

```
MSFlexGrid3.Text = "reset prg"
```

```
End
```

End If

'-----

DoEvents

Receive DIGIT

DATA = (DGT3 * 16 ^ 3 + DGT2 * 16 ^ 2 + DGT1 * 16 + DGT0)

If N = 1 Then

If Form2.Label3 = "KUR 33" Then

Form2.MSFlexGrid1.Row = 1

Form2.MSFlexGrid1.Col = 1

Form2.MSFlexGrid1.Text = DATA * 2

Form2.MSFlexGrid1.Row = 1

Form2.MSFlexGrid1.Col = 0

Form2.MSFlexGrid1.Text = DATA * 2 * K

End If

DoEvents

'-----

MSFlexGrid3.Row = 13

MSFlexGrid3.Col = 1

MSFlexGrid3.CellFontSize = 8

MSFlexGrid3.CellAlignment = 4

MSFlexGrid3.CellBackColor = &HFFFFFFF

MSFlexGrid3.Text = DATA * 2 * K

DoEvents

'-----

MSFlexGrid3.Row = 13

MSFlexGrid3.Col = 2

MSFlexGrid3.CellFontSize = 8

MSFlexGrid3.CellAlignment = 4

MSFlexGrid3.CellBackColor = &HFFFFFFF

'MSFlexGrid3.Text = average

MSFlexGrid3.Text = DATA * 2

'-----

Data1.Recordset.AddNew

```

Text2.Text = DATA * 2
Text3.Text = Time
Text4.Text = Date
Text5.Text = "KUR 33"
'-----
Else    'Okuma islemi olumsuz
        Transmit CU, WA, DJ
        '-----PROGRAM RESET
        If N = 1 Then
            MSFlexGrid3.Row = 13
            MSFlexGrid3.Col = 1
            MSFlexGrid3.CellFontSize = 8
            MSFlexGrid3.CellAlignment = 4
            MSFlexGrid3.CellBackColor = &HFFFFFF
            MSFlexGrid3.Text = "reset prg"
        End
        End If
        '-----
        DoEvents
        Receive DIGIT
        DATA = (DGT3 * 16 ^ 3 + DGT2 * 16 ^ 2 + DGT1 * 16 + DGT0)
        If N = 1 Then
            If Form2.Label3 = "KUR 33" Then
                Form2.MSFlexGrid1.Row = 1
                Form2.MSFlexGrid1.Col = 1
                Form2.MSFlexGrid1.Text = DATA * 2
                Form2.MSFlexGrid1.Row = 1
                Form2.MSFlexGrid1.Col = 0
                Form2.MSFlexGrid1.Text = DATA * 2 * K
            End If
        End If
        DoEvents
        '-----
        MSFlexGrid3.Row = 13
        MSFlexGrid3.Col = 1

```

```

MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HFFFFFFF
MSFlexGrid3.Text = DATA * 2 * K
DoEvents
'-----
MSFlexGrid3.Row = 13
MSFlexGrid3.Col = 2
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HFFFFFFF
'MSFlexGrid3.Text = average
MSFlexGrid3.Text = DATA * 2
'-----
Data1.Recordset.AddNew
Text2.Text = DATA * 2
Text3.Text = Time
Text4.Text = Date
Text5.Text = "KUR 33"
'-----
Else
MSFlexGrid3.Row = 13
MSFlexGrid3.Col = 2
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HFFFFFFF
'MSFlexGrid3.Text = average
MSFlexGrid3.Text = "No Signal"
End If
End If
DoEvents
TIME_INIT = Time
TIME_FINAL = Time
ECE = #12:01:00 AM#

```

```
Do Until (#12:00:05 AM# - #12:00:00 AM#) < (TIME_FINAL - TIME_INIT)
```

```
TIME_FINAL = Time
```

```
DoEvents
```

```
Loop
```

```
MSFlexGrid3.Row = 13
```

```
MSFlexGrid3.Col = 0
```

```
MSFlexGrid3.CellFontSize = 8
```

```
MSFlexGrid3.CellAlignment = 4
```

```
MSFlexGrid3.CellBackColor = &HFFFF80
```

```
MSFlexGrid3.Text = "KUR 33"
```

```
M = M + 1
```

```
Text1.Text = M
```

```
'=====
```

```
'Reading flow rate of KUR 40
```

```
DoEvents
```

```
MSFlexGrid3.Row = 14
```

```
MSFlexGrid3.Col = 0
```

```
MSFlexGrid3.CellFontSize = 8
```

```
MSFlexGrid3.CellAlignment = 4
```

```
MSFlexGrid3.CellBackColor = &HC0&
```

```
MSFlexGrid3.Text = "KUR 40"
```

```
MSFlexGrid3.Row = 14
```

```
MSFlexGrid3.Col = 2
```

```
MSFlexGrid3.CellFontSize = 8
```

```
MSFlexGrid3.CellAlignment = 4
```

```
MSFlexGrid3.CellBackColor = &HFFFFFF
```

```
'MSFlexGrid3.Text = average
```

```
MSFlexGrid3.Text = "Scanning..."
```

```
WA = 2
```

```
DJ = 10
```

```
K = 0.7
```

```
Transmit CU, WA, DJ
```

```

'-----PROGRAM RESET
    If N = 1 Then
        MSFlexGrid3.Row = 14
        MSFlexGrid3.Col = 1
        MSFlexGrid3.CellFontSize = 8
        MSFlexGrid3.CellAlignment = 4
        MSFlexGrid3.CellBackColor = &HFFFFFF
        MSFlexGrid3.Text = "reset prg"
    End
End If

'-----
DoEvents
Receive DIGIT
DATA = (DGT3 * 16 ^ 3 + DGT2 * 16 ^ 2 + DGT1 * 16 + DGT0)
If N = 1 Then
    If Form2.Label3 = "KUR 40" Then
        Form2.MSFlexGrid1.Row = 1
        Form2.MSFlexGrid1.Col = 1
        Form2.MSFlexGrid1.Text = DATA * 2
        Form2.MSFlexGrid1.Row = 1
        Form2.MSFlexGrid1.Col = 0
        Form2.MSFlexGrid1.Text = DATA * 2 * K
    End If
DoEvents
'-----
MSFlexGrid3.Row = 14
MSFlexGrid3.Col = 1
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HFFFFFF
MSFlexGrid3.Text = DATA * 2 * K
DoEvents
'-----

```

```

MSFlexGrid3.Row = 14
MSFlexGrid3.Col = 2
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HFFFFFFF
'MSFlexGrid3.Text = average
MSFlexGrid3.Text = DATA * 2
'-----

Data1.Recordset.AddNew
Text2.Text = DATA * 2
Text3.Text = Time
Text4.Text = Date
Text5.Text = "KUR 40"
'-----

Else    'okuma islemi olumsuz
        Transmit CU, WA, DJ
        '-----PROGRAM RESET
        If N = 1 Then
            MSFlexGrid3.Row = 14
            MSFlexGrid3.Col = 1
            MSFlexGrid3.CellFontSize = 8
            MSFlexGrid3.CellAlignment = 4
            MSFlexGrid3.CellBackColor = &HFFFFFFF
            MSFlexGrid3.Text = "reset prg"
        End
        End If

        '-----
        DoEvents
        Receive DIGIT
        DATA = (DGT3 * 16 ^ 3 + DGT2 * 16 ^ 2 + DGT1 * 16 + DGT0)
        If N = 1 Then
            If Form2.Label3 = "KUR 40" Then
                Form2.MSFlexGrid1.Row = 1

```



```

Form2.MSFlexGrid1.Col = 1
Form2.MSFlexGrid1.Text = DATA * 2
Form2.MSFlexGrid1.Row = 1
Form2.MSFlexGrid1.Col = 0
Form2.MSFlexGrid1.Text = DATA * 2 * K
End If
DoEvents
'-----
MSFlexGrid3.Row = 14
MSFlexGrid3.Col = 1
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HFFFFFF
MSFlexGrid3.Text = DATA * 2 * K
DoEvents
'-----
MSFlexGrid3.Row = 14
MSFlexGrid3.Col = 2
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HFFFFFF
'MSFlexGrid3.Text = average
MSFlexGrid3.Text = DATA * 2
'-----
Data1.Recordset.AddNew
Text2.Text = DATA * 2
Text3.Text = Time
Text4.Text = Date
Text5.Text = "KUR 40"
'-----
Else
MSFlexGrid3.Row = 14
MSFlexGrid3.Col = 2
MSFlexGrid3.CellFontSize = 8

```

```

        MSFlexGrid3.CellAlignment = 4
        MSFlexGrid3.CellBackColor = &HFFFFFF
        'MSFlexGrid3.Text = average
        MSFlexGrid3.Text = "No Signal"
    End If

End If

DoEvents

TIME_INIT = Time
TIME_FINAL = Time
ECE = #12:01:00 AM#
Do Until (#12:00:05 AM# - #12:00:00 AM#) < (TIME_FINAL - TIME_INIT)
    TIME_FINAL = Time
    DoEvents
    Loop
    MSFlexGrid3.Row = 14
    MSFlexGrid3.Col = 0
    MSFlexGrid3.CellFontSize = 8
    MSFlexGrid3.CellAlignment = 4
    MSFlexGrid3.CellBackColor = &HFFFF80
    MSFlexGrid3.Text = "KUR 40"
    M = M + 1
    Text1.Text = M
    '=====
    'Reading flow rate of KUR 41
    DoEvents
    MSFlexGrid3.Row = 15
    MSFlexGrid3.Col = 0
    MSFlexGrid3.CellFontSize = 8
    MSFlexGrid3.CellAlignment = 4
    MSFlexGrid3.CellBackColor = &HC0&
    MSFlexGrid3.Text = "KUR 41"

    MSFlexGrid3.Row = 15
    MSFlexGrid3.Col = 2

```

```
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HFFFFFFF
'MSFlexGrid3.Text = average
MSFlexGrid3.Text = "Scanning..."
```

```
WA = 2
```

```
DJ = 11
```

```
K = 0.7
```

```
Transmit CU, WA, DJ
```

```
'-----PROGRAM RESET
```

```
If N = 1 Then
```

```
MSFlexGrid3.Row = 15
```

```
MSFlexGrid3.Col = 1
```

```
MSFlexGrid3.CellFontSize = 8
```

```
MSFlexGrid3.CellAlignment = 4
```

```
MSFlexGrid3.CellBackColor = &HFFFFFFF
```

```
MSFlexGrid3.Text = "reset prg"
```

```
End
```

```
End If
```

```
'-----
```

```
DoEvents
```

```
Receive DIGIT
```

```
DATA = (DGT3 * 16 ^ 3 + DGT2 * 16 ^ 2 + DGT1 * 16 + DGT0)
```

```
If N = 1 Then
```

```
If Form2.Label3 = "KUR 41" Then
```

```
Form2.MSFlexGrid1.Row = 1
```

```
Form2.MSFlexGrid1.Col = 1
```

```
Form2.MSFlexGrid1.Text = DATA * 2
```

```
Form2.MSFlexGrid1.Row = 1
```

```
Form2.MSFlexGrid1.Col = 0
```

```
Form2.MSFlexGrid1.Text = DATA * 2 * K
```

```
End If
```

DoEvents

'-----

MSFlexGrid3.Row = 15

MSFlexGrid3.Col = 1

MSFlexGrid3.CellFontSize = 8

MSFlexGrid3.CellAlignment = 4

MSFlexGrid3.CellBackColor = &HFFFFFFF

MSFlexGrid3.Text = DATA * 2 * K

DoEvents

'-----

MSFlexGrid3.Row = 15

MSFlexGrid3.Col = 2

MSFlexGrid3.CellFontSize = 8

MSFlexGrid3.CellAlignment = 4

MSFlexGrid3.CellBackColor = &HFFFFFFF

'MSFlexGrid3.Text = average

MSFlexGrid3.Text = DATA * 2

'-----

Data1.Recordset.AddNew

Text2.Text = DATA * 2

Text3.Text = Time

Text4.Text = Date

Text5.Text = "KUR 41"

'-----

Else 'okuma islemi olumsuz

Transmit CU, WA, DJ

'-----PROGRAM RESET

If N = 1 Then

MSFlexGrid3.Row = 15

MSFlexGrid3.Col = 1

MSFlexGrid3.CellFontSize = 8

MSFlexGrid3.CellAlignment = 4

MSFlexGrid3.CellBackColor = &HFFFFFFF

MSFlexGrid3.Text = "reset prg"

```

End
End If

'-----
DoEvents
Receive DIGIT
DATA = (DGT3 * 16 ^ 3 + DGT2 * 16 ^ 2 + DGT1 * 16 + DGT0)
If N = 1 Then
If Form2.Label3 = "KUR 41" Then
Form2.MSFlexGrid1.Row = 1
Form2.MSFlexGrid1.Col = 1
Form2.MSFlexGrid1.Text = DATA * 2
Form2.MSFlexGrid1.Row = 1
Form2.MSFlexGrid1.Col = 0
Form2.MSFlexGrid1.Text = DATA * 2 * K
End If
DoEvents
'-----

MSFlexGrid3.Row = 15
MSFlexGrid3.Col = 1
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HFFFFFF
MSFlexGrid3.Text = DATA * 2 * K
DoEvents
'-----

MSFlexGrid3.Row = 15
MSFlexGrid3.Col = 2
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HFFFFFF
'MSFlexGrid3.Text = average
MSFlexGrid3.Text = DATA * 2
'-----

```

```

Data1.Recordset.AddNew
Text2.Text = DATA * 2
Text3.Text = Time
Text4.Text = Date
Text5.Text = "KUR 41"
'-----
Else      'okuma islemi olumsuz

MSFlexGrid3.Row = 15
MSFlexGrid3.Col = 2
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HFFFFFFF
'MSFlexGrid3.Text = average
MSFlexGrid3.Text = "No Signal"

End If

```

```

End If
DoEvents
TIME_INIT = Time
TIME_FINAL = Time
ECE = #12:05:00 AM#
Do Until (#12:00:05 AM# - #12:00:00 AM#) < (TIME_FINAL - TIME_INIT)
TIME_FINAL = Time
DoEvents
Loop
MSFlexGrid3.Row = 15
MSFlexGrid3.Col = 0
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HFFFF80
MSFlexGrid3.Text = "KUR 41"
M = M + 1

```

Text1.Text = M

'=====

'Reading flow rate of KUR 42

DoEvents

MSFlexGrid3.Row = 16

MSFlexGrid3.Col = 0

MSFlexGrid3.CellFontSize = 8

MSFlexGrid3.CellAlignment = 4

MSFlexGrid3.CellBackColor = &HC0&

MSFlexGrid3.Text = "KUR 42"

MSFlexGrid3.Row = 16

MSFlexGrid3.Col = 2

MSFlexGrid3.CellFontSize = 8

MSFlexGrid3.CellAlignment = 4

MSFlexGrid3.CellBackColor = &HFFFFFF

'MSFlexGrid3.Text = average

MSFlexGrid3.Text = "Scanning..."

WA = 2

DJ = 1

K = 1

Transmit CU, WA, DJ

'-----PROGRAM RESET

If N = 1 Then

MSFlexGrid3.Row = 16

MSFlexGrid3.Col = 1

MSFlexGrid3.CellFontSize = 8

MSFlexGrid3.CellAlignment = 4

MSFlexGrid3.CellBackColor = &HFFFFFF

MSFlexGrid3.Text = "reset prg"

End

End If

```

'-----
DoEvents
Receive DIGIT
DATA = (DGT3 * 16 ^ 3 + DGT2 * 16 ^ 2 + DGT1 * 16 + DGT0)
If N = 1 Then
If Form2.Label3 = "KUR 42" Then
Form2.MSFlexGrid1.Row = 1
Form2.MSFlexGrid1.Col = 1
Form2.MSFlexGrid1.Text = DATA * 2
Form2.MSFlexGrid1.Row = 1
Form2.MSFlexGrid1.Col = 0
Form2.MSFlexGrid1.Text = DATA * 2 * K
End If
DoEvents
'-----
MSFlexGrid3.Row = 16
MSFlexGrid3.Col = 1
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HFFFFFF
MSFlexGrid3.Text = DATA * 2 * K
DoEvents
'-----
MSFlexGrid3.Row = 16
MSFlexGrid3.Col = 2
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HFFFFFF
'MSFlexGrid3.Text = average
MSFlexGrid3.Text = DATA * 2
'-----
Data1.Recordset.AddNew
Text2.Text = DATA * 2
Text3.Text = Time

```



```
Text4.Text = Date
```

```
Text5.Text = "KUR 42"
```

```
'-----
```

```
Else 'okuma islemi olumsuz
```

```
    Transmit CU, WA, DJ
```

```
    '-----PROGRAM RESET
```

```
    If N = 1 Then
```

```
        MSFlexGrid3.Row = 16
```

```
        MSFlexGrid3.Col = 1
```

```
        MSFlexGrid3.CellFontSize = 8
```

```
        MSFlexGrid3.CellAlignment = 4
```

```
        MSFlexGrid3.CellBackColor = &HFFFFFF
```

```
        MSFlexGrid3.Text = "reset prg"
```

```
    End
```

```
    End If
```

```
'-----
```

```
DoEvents
```

```
Receive DIGIT
```

```
DATA = (DGT3 * 16 ^ 3 + DGT2 * 16 ^ 2 + DGT1 * 16 + DGT0)
```

```
If N = 1 Then
```

```
    If Form2.Label3 = "KUR 42" Then
```

```
        Form2.MSFlexGrid1.Row = 1
```

```
        Form2.MSFlexGrid1.Col = 1
```

```
        Form2.MSFlexGrid1.Text = DATA * 2
```

```
        Form2.MSFlexGrid1.Row = 1
```

```
        Form2.MSFlexGrid1.Col = 0
```

```
        Form2.MSFlexGrid1.Text = DATA * 2 * K
```

```
    End If
```

```
DoEvents
```

```
'-----
```

```
MSFlexGrid3.Row = 16
```

```
MSFlexGrid3.Col = 1
```

```
MSFlexGrid3.CellFontSize = 8
```

```

MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HFFFFFF
MSFlexGrid3.Text = DATA * 2 * K
DoEvents
'-----
MSFlexGrid3.Row = 16
MSFlexGrid3.Col = 2
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HFFFFFF
'MSFlexGrid3.Text = average
MSFlexGrid3.Text = DATA * 2
'-----
Data1.Recordset.AddNew
Text2.Text = DATA * 2
Text3.Text = Time
Text4.Text = Date
Text5.Text = "KUR 42"
'-----
Else 'okuma islemi olumsuz

MSFlexGrid3.Row = 16
MSFlexGrid3.Col = 2
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HFFFFFF
'MSFlexGrid3.Text = average
MSFlexGrid3.Text = "No Signal"

End If

End If
DoEvents
TIME_INIT = Time

```

```

TIME_FINAL = Time
ECE = #12:05:00 AM#
Do Until (#12:00:05 AM# - #12:00:00 AM#) < (TIME_FINAL - TIME_INIT)
TIME_FINAL = Time
DoEvents
Loop
MSFlexGrid3.Row = 16
MSFlexGrid3.Col = 0
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HFFFF80
MSFlexGrid3.Text = "KUR 42"
M = M + 1
Text1.Text = M
'=====
'Reading flow rate of KUR 47
DoEvents
MSFlexGrid3.Row = 17
MSFlexGrid3.Col = 0
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HC0&
MSFlexGrid3.Text = "KUR 47"

MSFlexGrid3.Row = 17
MSFlexGrid3.Col = 2
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HFFFFFFF
'MSFlexGrid3.Text = average
MSFlexGrid3.Text = "Scanning..."

WA = 2
DJ = 2

```

```

K = 0.74
Transmit CU, WA, DJ
'-----PROGRAM RESET
If N = 1 Then
MSFlexGrid3.Row = 17
MSFlexGrid3.Col = 1
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HFFFFFFF
MSFlexGrid3.Text = "reset prg"
End
End If

'-----
DoEvents
Receive DIGIT
DATA = (DGT3 * 16 ^ 3 + DGT2 * 16 ^ 2 + DGT1 * 16 + DGT0)
If N = 1 Then
If Form2.Label3 = "KUR 47" Then
Form2.MSFlexGrid1.Row = 1
Form2.MSFlexGrid1.Col = 1
Form2.MSFlexGrid1.Text = DATA * 2
Form2.MSFlexGrid1.Row = 1
Form2.MSFlexGrid1.Col = 0
Form2.MSFlexGrid1.Text = DATA * 2 * K
End If
DoEvents
'-----
MSFlexGrid3.Row = 17
MSFlexGrid3.Col = 1
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HFFFFFFF
MSFlexGrid3.Text = DATA * 2 * K

```

DoEvents

'-----

MSFlexGrid3.Row = 17

MSFlexGrid3.Col = 2

MSFlexGrid3.CellFontSize = 8

MSFlexGrid3.CellAlignment = 4

MSFlexGrid3.CellBackColor = &HFFFFFFF

'MSFlexGrid3.Text = average

MSFlexGrid3.Text = DATA * 2

'-----

Data1.Recordset.AddNew

Text2.Text = DATA * 2

Text3.Text = Time

Text4.Text = Date

Text5.Text = "KUR 47"

'-----

Else 'okuma islemi olumsuz

Transmit CU, WA, DJ

'-----PROGRAM RESET

If N = 1 Then

MSFlexGrid3.Row = 17

MSFlexGrid3.Col = 1

MSFlexGrid3.CellFontSize = 8

MSFlexGrid3.CellAlignment = 4

MSFlexGrid3.CellBackColor = &HFFFFFFF

MSFlexGrid3.Text = "reset prg"

End

End If

'-----

DoEvents

Receive DIGIT

DATA = (DGT3 * 16 ^ 3 + DGT2 * 16 ^ 2 + DGT1 * 16 + DGT0)

If N = 1 Then

If Form2.Label3 = "KUR 47" Then

```
Form2.MSFlexGrid1.Row = 1
Form2.MSFlexGrid1.Col = 1
Form2.MSFlexGrid1.Text = DATA * 2
Form2.MSFlexGrid1.Row = 1
Form2.MSFlexGrid1.Col = 0
Form2.MSFlexGrid1.Text = DATA * 2 * K
End If
DoEvents
'-----
MSFlexGrid3.Row = 17
MSFlexGrid3.Col = 1
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HFFFFFF
MSFlexGrid3.Text = DATA * 2 * K
DoEvents
'-----
MSFlexGrid3.Row = 17
MSFlexGrid3.Col = 2
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HFFFFFF
'MSFlexGrid3.Text = average
MSFlexGrid3.Text = DATA * 2
'-----
Data1.Recordset.AddNew
Text2.Text = DATA * 2
Text3.Text = Time
Text4.Text = Date
Text5.Text = "KUR 47"
'-----

Else 'okuma islemi olumsuz
```

```

MSFlexGrid3.Row = 17
MSFlexGrid3.Col = 2
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HFFFFFF
'MSFlexGrid3.Text = average
MSFlexGrid3.Text = "No Signal"

```

```
End If
```

```
End If
```

```
DoEvents
```

```
TIME_INIT = Time
```

```
TIME_FINAL = Time
```

```
ECE = #12:01:00 AM#
```

```
Do Until (#12:00:05 AM# - #12:00:00 AM#) < (TIME_FINAL - TIME_INIT)
```

```
TIME_FINAL = Time
```

```
DoEvents
```

```
Loop
```

```
MSFlexGrid3.Row = 17
```

```
MSFlexGrid3.Col = 0
```

```
MSFlexGrid3.CellFontSize = 8
```

```
MSFlexGrid3.CellAlignment = 4
```

```
MSFlexGrid3.CellBackColor = &HFFFF80
```

```
MSFlexGrid3.Text = "KUR 47"
```

```
M = M + 1
```

```
'=====
```

```
'Reading flow rate of KUR 49
```

```
DoEvents
```

```
MSFlexGrid3.Row = 18
```

```
MSFlexGrid3.Col = 0
```

```
MSFlexGrid3.CellFontSize = 8
```

```
MSFlexGrid3.CellAlignment = 4
```

```
MSFlexGrid3.CellBackColor = &HC0&
```

```
MSFlexGrid3.Text = "KUR 49"
```

```

MSFlexGrid3.Row = 18
MSFlexGrid3.Col = 2
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HFFFFFFF
'MSFlexGrid3.Text = average
MSFlexGrid3.Text = "Scanning..."

```

```

WA = 2
DJ = 0
K = 2813.9
Transmit CU, WA, DJ
'-----PROGRAM RESET
If N = 1 Then
MSFlexGrid3.Row = 18
MSFlexGrid3.Col = 1
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HFFFFFFF
MSFlexGrid3.Text = "reset prg"
End
End If
'-----
DoEvents
Receive DIGIT
DATA = (DGT3 * 16 ^ 3 + DGT2 * 16 ^ 2 + DGT1 * 16 + DGT0)
If N = 1 Then
If Form2.Label3 = "KUR 49" Then
Form2.MSFlexGrid1.Row = 1
Form2.MSFlexGrid1.Col = 1
Form2.MSFlexGrid1.Text = DATA * 2
Form2.MSFlexGrid1.Row = 1
Form2.MSFlexGrid1.Col = 0

```



```

Form2.MSFlexGrid1.Text = DATA * 2 * K
End If
DoEvents
'-----
MSFlexGrid3.Row = 18
MSFlexGrid3.Col = 1
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HFFFFFF
MSFlexGrid3.Text = DATA * 2 * K
DoEvents
'-----
MSFlexGrid3.Row = 18
MSFlexGrid3.Col = 2
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HFFFFFF
'MSFlexGrid3.Text = average
MSFlexGrid3.Text = DATA * 2
'-----
Data1.Recordset.AddNew
Text2.Text = DATA * 2
Text3.Text = Time
Text4.Text = Date
Text5.Text = "KUR 49"
'-----
Else    'okuma islemi olumsuz
        Transmit CU, WA, DJ
        '-----PROGRAM RESET
        If N = 1 Then
            MSFlexGrid3.Row = 18
            MSFlexGrid3.Col = 1
            MSFlexGrid3.CellFontSize = 8
            MSFlexGrid3.CellAlignment = 4

```

```

MSFlexGrid3.CellBackColor = &HFFFFFFF
MSFlexGrid3.Text = "reset prg"
End
End If
'-----
DoEvents
Receive DIGIT
DATA = (DGT3 * 16 ^ 3 + DGT2 * 16 ^ 2 + DGT1 * 16 + DGT0)
If N = 1 Then
If Form2.Label3 = "KUR 49" Then
Form2.MSFlexGrid1.Row = 1
Form2.MSFlexGrid1.Col = 1
Form2.MSFlexGrid1.Text = DATA * 2
Form2.MSFlexGrid1.Row = 1
Form2.MSFlexGrid1.Col = 0
Form2.MSFlexGrid1.Text = DATA * 2 * K
End If
DoEvents
'-----
MSFlexGrid3.Row = 18
MSFlexGrid3.Col = 1
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HFFFFFFF
MSFlexGrid3.Text = DATA * 2 * K
DoEvents
'-----
MSFlexGrid3.Row = 18
MSFlexGrid3.Col = 2
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HFFFFFFF
'MSFlexGrid3.Text = average
MSFlexGrid3.Text = DATA * 2

```

```

'-----
Data1.Recordset.AddNew
Text2.Text = DATA * 2
Text3.Text = Time
Text4.Text = Date
Text5.Text = "KUR 49"
'-----

Else    'okuma islemi olumsuz

MSFlexGrid3.Row = 18
MSFlexGrid3.Col = 2
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HFFFFFF
'MSFlexGrid3.Text = average
MSFlexGrid3.Text = "No Signal"

End If

```

```

End If
DoEvents
TIME_INIT = Time
TIME_FINAL = Time
ECE = #12:01:00 AM#
Do Until (#12:00:05 AM# - #12:00:00 AM#) < (TIME_FINAL - TIME_INIT)
TIME_FINAL = Time
DoEvents
Loop
MSFlexGrid3.Row = 18
MSFlexGrid3.Col = 0
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HFFF80
MSFlexGrid3.Text = "KUR 49"

```

M = M + 1

'=====

'Loop

'=====

'Reading flow rate of KUS 02

DoEvents

MSFlexGrid1.Row = 1

MSFlexGrid1.Col = 0

MSFlexGrid1.CellFontSize = 8

MSFlexGrid1.CellAlignment = 4

MSFlexGrid1.CellBackColor = &HC0&

MSFlexGrid1.Text = "KUS 02"

MSFlexGrid1.Row = 1

MSFlexGrid1.Col = 2

MSFlexGrid1.CellFontSize = 8

MSFlexGrid1.CellAlignment = 4

MSFlexGrid1.CellBackColor = &HFFFFFF

'MSFlexGrid1.Text = average

MSFlexGrid1.Text = "Scanning..."

WA = 3

DJ = 0

K = 37.39

Transmit CU, WA, DJ

'-----PROGRAM RESET

If N = 1 Then

MSFlexGrid1.Row = 1

MSFlexGrid1.Col = 1

MSFlexGrid1.CellFontSize = 8

MSFlexGrid1.CellAlignment = 4

MSFlexGrid1.CellBackColor = &HFFFFFF

MSFlexGrid1.Text = "reset prg"

End

End If

```

'-----
DoEvents
Receive DIGIT
DATA = (DGT3 * 16 ^ 3 + DGT2 * 16 ^ 2 + DGT1 * 16 + DGT0)
If N = 1 Then
If Form2.Label3 = "KUS 02" Then
Form2.MSFlexGrid1.Row = 1
Form2.MSFlexGrid1.Col = 1
Form2.MSFlexGrid1.Text = DATA * 2
Form2.MSFlexGrid1.Row = 1
Form2.MSFlexGrid1.Col = 0
Form2.MSFlexGrid1.Text = DATA * 2 * K
End If
DoEvents
'-----

MSFlexGrid1.Row = 1
MSFlexGrid1.Col = 1
MSFlexGrid1.CellFontSize = 8
MSFlexGrid1.CellAlignment = 4
MSFlexGrid1.CellBackColor = &HFFFFFF
MSFlexGrid1.Text = DATA * 2 * K
DoEvents
'-----

MSFlexGrid1.Row = 1
MSFlexGrid1.Col = 2
MSFlexGrid1.CellFontSize = 8
MSFlexGrid1.CellAlignment = 4
MSFlexGrid1.CellBackColor = &HFFFFFF
'MSFlexGrid1.Text = average
MSFlexGrid1.Text = DATA * 2
'-----

DoEvents
Data1.Recordset.AddNew
Text2.Text = DATA * 2

```

Text3.Text = Time

Text4.Text = Date

Text5.Text = "KUS 02"

'-----

Else

Transmit CU, WA, DJ

'-----PROGRAM RESET

If N = 1 Then

MSFlexGrid1.Row = 1

MSFlexGrid1.Col = 1

MSFlexGrid1.CellFontSize = 8

MSFlexGrid1.CellAlignment = 4

MSFlexGrid1.CellBackColor = &HFFFFFF

MSFlexGrid1.Text = "reset prg"

End

End If

'-----

DoEvents

Receive DIGIT

DATA = (DGT3 * 16 ^ 3 + DGT2 * 16 ^ 2 + DGT1 * 16 + DGT0)

If N = 1 Then

If Form2.Label3 = "KUS 02" Then

Form2.MSFlexGrid1.Row = 1

Form2.MSFlexGrid1.Col = 1

Form2.MSFlexGrid1.Text = DATA * 2

Form2.MSFlexGrid1.Row = 1

Form2.MSFlexGrid1.Col = 0

Form2.MSFlexGrid1.Text = DATA * 2 * K

End If

DoEvents

'-----

MSFlexGrid1.Row = 1

MSFlexGrid1.Col = 1

MSFlexGrid1.CellFontSize = 8

```

MSFlexGrid1.CellAlignment = 4
MSFlexGrid1.CellBackColor = &HFFFFFF
MSFlexGrid1.Text = DATA * 2 * K
DoEvents
'-----
MSFlexGrid1.Row = 1
MSFlexGrid1.Col = 2
MSFlexGrid1.CellFontSize = 8
MSFlexGrid1.CellAlignment = 4
MSFlexGrid1.CellBackColor = &HFFFFFF
'MSFlexGrid1.Text = average
MSFlexGrid1.Text = DATA * 2
'-----
DoEvents
Data1.Recordset.AddNew
Text2.Text = DATA * 2
Text3.Text = Time
Text4.Text = Date
Text5.Text = "KUS 02"
'-----
Else

MSFlexGrid1.Row = 1
MSFlexGrid1.Col = 2
MSFlexGrid1.CellFontSize = 8
MSFlexGrid1.CellAlignment = 4
MSFlexGrid1.CellBackColor = &HFFFFFF
'MSFlexGrid1.Text = average
MSFlexGrid1.Text = "No Signal"
End If

End If
DoEvents
TIME_INIT = Time
TIME_FINAL = Time

```

```

ECE = #12:01:00 AM#
Do Until (#12:00:05 AM# - #12:00:00 AM#) < (TIME_FINAL - TIME_INIT)
TIME_FINAL = Time
DoEvents
Loop
MSFlexGrid1.Row = 1
MSFlexGrid1.Col = 0
MSFlexGrid1.CellFontSize = 8
MSFlexGrid1.CellAlignment = 4
MSFlexGrid1.CellBackColor = &HFFFF80
MSFlexGrid1.Text = "KUS 02"
M = M + 1
Text1.Text = M

```

```

'=====

```

```

'Reading flow rate of KUS 10
DoEvents
MSFlexGrid1.Row = 2
MSFlexGrid1.Col = 0
MSFlexGrid1.CellFontSize = 8
MSFlexGrid1.CellAlignment = 4
MSFlexGrid1.CellBackColor = &HC0&
MSFlexGrid1.Text = "KUS 10"

MSFlexGrid1.Row = 2
MSFlexGrid1.Col = 2
MSFlexGrid1.CellFontSize = 8
MSFlexGrid1.CellAlignment = 4
MSFlexGrid1.CellBackColor = &HFFFFFFF
'MSFlexGrid1.Text = average
MSFlexGrid1.Text = "Scanning..."
WA = 3
DJ = 1
K = 6.48

```



```

Transmit CU, WA, DJ
'-----PROGRAM RESET
If N = 1 Then
MSFlexGrid1.Row = 2
MSFlexGrid1.Col = 1
MSFlexGrid1.CellFontSize = 8
MSFlexGrid1.CellAlignment = 4
MSFlexGrid1.CellBackColor = &HFFFFFFF
MSFlexGrid1.Text = "reset prg"
End
End If
'-----
DoEvents
Receive DIGIT
DATA = (DGT3 * 16 ^ 3 + DGT2 * 16 ^ 2 + DGT1 * 16 + DGT0)
If N = 1 Then
If Form2.Label3 = "KUS 10" Then
Form2.MSFlexGrid1.Row = 1
Form2.MSFlexGrid1.Col = 1
Form2.MSFlexGrid1.Text = DATA * 2
Form2.MSFlexGrid1.Row = 1
Form2.MSFlexGrid1.Col = 0
Form2.MSFlexGrid1.Text = DATA * 2 * K
End If
DoEvents
'-----
MSFlexGrid1.Row = 2
MSFlexGrid1.Col = 1
MSFlexGrid1.CellFontSize = 8
MSFlexGrid1.CellAlignment = 4
MSFlexGrid1.CellBackColor = &HFFFFFFF
MSFlexGrid1.Text = DATA * 2 * K
DoEvents
'-----

```

```

MSFlexGrid1.Row = 2
MSFlexGrid1.Col = 2
MSFlexGrid1.CellFontSize = 8
MSFlexGrid1.CellAlignment = 4
MSFlexGrid1.CellBackColor = &HFFFFFF
'MSFlexGrid1.Text = average
MSFlexGrid1.Text = DATA * 2
'-----

Data1.Recordset.AddNew
Text2.Text = DATA * 2
Text3.Text = Time
Text4.Text = Date
Text5.Text = "KUS 10"
'-----

Else
    Transmit CU, WA, DJ
    '-----PROGRAM RESET
    If N = 1 Then
        MSFlexGrid1.Row = 2
        MSFlexGrid1.Col = 1
        MSFlexGrid1.CellFontSize = 8
        MSFlexGrid1.CellAlignment = 4
        MSFlexGrid1.CellBackColor = &HFFFFFF
        MSFlexGrid1.Text = "reset prg"
    End
    End If
    '-----
    DoEvents
    Receive DIGIT
    DATA = (DGT3 * 16 ^ 3 + DGT2 * 16 ^ 2 + DGT1 * 16 + DGT0)
    If N = 1 Then
        If Form2.Label3 = "KUS 10" Then
            Form2.MSFlexGrid1.Row = 1
            Form2.MSFlexGrid1.Col = 1

```

```

Form2.MSFlexGrid1.Text = DATA * 2
Form2.MSFlexGrid1.Row = 1
Form2.MSFlexGrid1.Col = 0
Form2.MSFlexGrid1.Text = DATA * 2 * K
End If
DoEvents
'-----
MSFlexGrid1.Row = 2
MSFlexGrid1.Col = 1
MSFlexGrid1.CellFontSize = 8
MSFlexGrid1.CellAlignment = 4
MSFlexGrid1.CellBackColor = &HFFFFFF
MSFlexGrid1.Text = DATA * 2 * K
DoEvents
'-----
MSFlexGrid1.Row = 2
MSFlexGrid1.Col = 2
MSFlexGrid1.CellFontSize = 8
MSFlexGrid1.CellAlignment = 4
MSFlexGrid1.CellBackColor = &HFFFFFF
'MSFlexGrid1.Text = average
MSFlexGrid1.Text = DATA * 2
'-----
Data1.Recordset.AddNew
Text2.Text = DATA * 2
Text3.Text = Time
Text4.Text = Date
Text5.Text = "KUS 10"
'-----
Else
MSFlexGrid1.Row = 2
MSFlexGrid1.Col = 2
MSFlexGrid1.CellFontSize = 8
MSFlexGrid1.CellAlignment = 4

```

```

MSFlexGrid1.CellBackColor = &HFFFFFF
'MSFlexGrid1.Text = average
MSFlexGrid1.Text = "No Signal"
End If

End If
DoEvents
TIME_INIT = Time
TIME_FINAL = Time
ECE = #12:01:00 AM#
Do Until (#12:00:05 AM# - #12:00:00 AM#) < (TIME_FINAL - TIME_INIT)
TIME_FINAL = Time
DoEvents
Loop
MSFlexGrid1.Row = 2
MSFlexGrid1.Col = 0
MSFlexGrid1.CellFontSize = 8
MSFlexGrid1.CellAlignment = 4
MSFlexGrid1.CellBackColor = &HFFF80
MSFlexGrid1.Text = "KUS 10"
M = M + 1
Text1.Text = M

'=====
'Reading flow rate of KUS 11
DoEvents
MSFlexGrid1.Row = 3
MSFlexGrid1.Col = 0
MSFlexGrid1.CellFontSize = 8
MSFlexGrid1.CellAlignment = 4
MSFlexGrid1.CellBackColor = &HC0&
MSFlexGrid1.Text = "KUS 11"

MSFlexGrid1.Row = 3
MSFlexGrid1.Col = 2

```

```

MSFlexGrid1.CellFontSize = 8
MSFlexGrid1.CellAlignment = 4
MSFlexGrid1.CellBackColor = &HFFFFFFF
'MSFlexGrid1.Text = average
MSFlexGrid1.Text = "Scanning..."

```

```
WA = 3
```

```
DJ = 2
```

```
K = 6.29
```

```
Transmit CU, WA, DJ
```

```
'-----PROGRAM RESET
```

```
If N = 1 Then
```

```
MSFlexGrid1.Row = 3
```

```
MSFlexGrid1.Col = 1
```

```
MSFlexGrid1.CellFontSize = 8
```

```
MSFlexGrid1.CellAlignment = 4
```

```
MSFlexGrid1.CellBackColor = &HFFFFFFF
```

```
MSFlexGrid1.Text = "reset prg"
```

```
End
```

```
End If
```

```
'-----
```

```
DoEvents
```

```
Receive DIGIT
```

```
DATA = (DGT3 * 16 ^ 3 + DGT2 * 16 ^ 2 + DGT1 * 16 + DGT0)
```

```
If N = 1 Then
```

```
If Form2.Label3 = "KUS 11" Then
```

```
Form2.MSFlexGrid1.Row = 1
```

```
Form2.MSFlexGrid1.Col = 1
```

```
Form2.MSFlexGrid1.Text = DATA * 2
```

```
Form2.MSFlexGrid1.Row = 1
```

```
Form2.MSFlexGrid1.Col = 0
```

```
Form2.MSFlexGrid1.Text = DATA * 2 * K
```

```
End If
```

```
DoEvents
```

```

'-----
MSFlexGrid1.Row = 3
MSFlexGrid1.Col = 1
MSFlexGrid1.CellFontSize = 8
MSFlexGrid1.CellAlignment = 4
MSFlexGrid1.CellBackColor = &HFFFFFF
MSFlexGrid1.Text = DATA * 2 * K
DoEvents
'-----

MSFlexGrid1.Row = 3
MSFlexGrid1.Col = 2
MSFlexGrid1.CellFontSize = 8
MSFlexGrid1.CellAlignment = 4
MSFlexGrid1.CellBackColor = &HFFFFFF
'MSFlexGrid1.Text = average
MSFlexGrid1.Text = DATA * 2
'-----

Data1.Recordset.AddNew
Text2.Text = DATA * 2
Text3.Text = Time
Text4.Text = Date
Text5.Text = "KUS 11"
'-----

Else
    Transmit CU, WA, DJ
    '-----PROGRAM RESET
    If N = 1 Then
        MSFlexGrid1.Row = 3
        MSFlexGrid1.Col = 1
        MSFlexGrid1.CellFontSize = 8
        MSFlexGrid1.CellAlignment = 4
        MSFlexGrid1.CellBackColor = &HFFFFFF
        MSFlexGrid1.Text = "reset prg"
    End

```

```

End If
'-----
DoEvents
Receive DIGIT
DATA = (DGT3 * 16 ^ 3 + DGT2 * 16 ^ 2 + DGT1 * 16 + DGT0)
If N = 1 Then
If Form2.Label3 = "KUS 11" Then
Form2.MSFlexGrid1.Row = 1
Form2.MSFlexGrid1.Col = 1
Form2.MSFlexGrid1.Text = DATA * 2
Form2.MSFlexGrid1.Row = 1
Form2.MSFlexGrid1.Col = 0
Form2.MSFlexGrid1.Text = DATA * 2 * K
End If
DoEvents
'-----
MSFlexGrid1.Row = 3
MSFlexGrid1.Col = 1
MSFlexGrid1.CellFontSize = 8
MSFlexGrid1.CellAlignment = 4
MSFlexGrid1.CellBackColor = &HFFFFFF
MSFlexGrid1.Text = DATA * 2 * K
DoEvents
'-----
MSFlexGrid1.Row = 3
MSFlexGrid1.Col = 2
MSFlexGrid1.CellFontSize = 8
MSFlexGrid1.CellAlignment = 4
MSFlexGrid1.CellBackColor = &HFFFFFF
'MSFlexGrid1.Text = average
MSFlexGrid1.Text = DATA * 2
'-----
Data1.Recordset.AddNew
Text2.Text = DATA * 2

```

```

Text3.Text = Time
Text4.Text = Date
Text5.Text = "KUS 11"
'-----

Else
MSFlexGrid1.Row = 3
MSFlexGrid1.Col = 2
MSFlexGrid1.CellFontSize = 8
MSFlexGrid1.CellAlignment = 4
MSFlexGrid1.CellBackColor = &HFFFFFF
'MSFlexGrid1.Text = average
MSFlexGrid1.Text = "No Signal"
End If

End If
DoEvents
TIME_INIT = Time
TIME_FINAL = Time
ECE = #12:01:00 AM#
Do Until (#12:00:05 AM# - #12:00:00 AM#) < (TIME_FINAL - TIME_INIT)
TIME_FINAL = Time
DoEvents
Loop
MSFlexGrid1.Row = 3
MSFlexGrid1.Col = 0
MSFlexGrid1.CellFontSize = 8
MSFlexGrid1.CellAlignment = 4
MSFlexGrid1.CellBackColor = &HFFF80
MSFlexGrid1.Text = "KUS 11"
M = M + 1
Text1.Text = M

'=====

'Reading flow rate of KUS 12
DoEvents

```



```
MSFlexGrid1.Row = 4
MSFlexGrid1.Col = 0
MSFlexGrid1.CellFontSize = 8
MSFlexGrid1.CellAlignment = 4
MSFlexGrid1.CellBackColor = &HC0&
MSFlexGrid1.Text = "KUS 12"
```

```
MSFlexGrid1.Row = 4
MSFlexGrid1.Col = 2
MSFlexGrid1.CellFontSize = 8
MSFlexGrid1.CellAlignment = 4
MSFlexGrid1.CellBackColor = &HFFFFFF
'MSFlexGrid1.Text = average
MSFlexGrid1.Text = "Scanning..."
```

```
WA = 3
DJ = 3
K = 0.73
Transmit CU, WA, DJ
'-----PROGRAM RESET
If N = 1 Then
MSFlexGrid1.Row = 4
MSFlexGrid1.Col = 1
MSFlexGrid1.CellFontSize = 8
MSFlexGrid1.CellAlignment = 4
MSFlexGrid1.CellBackColor = &HFFFFFF
MSFlexGrid1.Text = "reset prg"
End
End If
'-----
DoEvents
Receive DIGIT
DATA = (DGT3 * 16 ^ 3 + DGT2 * 16 ^ 2 + DGT1 * 16 + DGT0)
If N = 1 Then
```

```

If Form2.Label3 = "KUS 12" Then
Form2.MSFlexGrid1.Row = 1
Form2.MSFlexGrid1.Col = 1
Form2.MSFlexGrid1.Text = DATA * 2
Form2.MSFlexGrid1.Row = 1
Form2.MSFlexGrid1.Col = 0
Form2.MSFlexGrid1.Text = DATA * 2 * K
End If
DoEvents
'-----
MSFlexGrid1.Row = 4
MSFlexGrid1.Col = 1
MSFlexGrid1.CellFontSize = 8
MSFlexGrid1.CellAlignment = 4
MSFlexGrid1.CellBackColor = &HFFFFFFF
MSFlexGrid1.Text = DATA * 2 * K
DoEvents
'-----
MSFlexGrid1.Row = 4
MSFlexGrid1.Col = 2
MSFlexGrid1.CellFontSize = 8
MSFlexGrid1.CellAlignment = 4
MSFlexGrid1.CellBackColor = &HFFFFFFF
'MSFlexGrid1.Text = average
MSFlexGrid1.Text = DATA * 2
'-----
Data1.Recordset.AddNew
Text2.Text = DATA * 2
Text3.Text = Time
Text4.Text = Date
Text5.Text = "KUS 12"
'-----
Else

```

Transmit CU, WA, DJ

```

'-----PROGRAM RESET
If N = 1 Then
MSFlexGrid1.Row = 4
MSFlexGrid1.Col = 1
MSFlexGrid1.CellFontSize = 8
MSFlexGrid1.CellAlignment = 4
MSFlexGrid1.CellBackColor = &HFFFFFF
MSFlexGrid1.Text = "reset prg"
End
End If
'-----

DoEvents
Receive DIGIT
DATA = (DGT3 * 16 ^ 3 + DGT2 * 16 ^ 2 + DGT1 * 16 +
DGT0)

If N = 1 Then
If Form2.Label3 = "KUS 12" Then
Form2.MSFlexGrid1.Row = 1
Form2.MSFlexGrid1.Col = 1
Form2.MSFlexGrid1.Text = DATA * 2
Form2.MSFlexGrid1.Row = 1
Form2.MSFlexGrid1.Col = 0
Form2.MSFlexGrid1.Text = DATA * 2 * K
End If
DoEvents
'-----

MSFlexGrid1.Row = 4
MSFlexGrid1.Col = 1
MSFlexGrid1.CellFontSize = 8
MSFlexGrid1.CellAlignment = 4
MSFlexGrid1.CellBackColor = &HFFFFFF
MSFlexGrid1.Text = DATA * 2 * K
DoEvents
'-----

```

```

MSFlexGrid1.Row = 4
MSFlexGrid1.Col = 2
MSFlexGrid1.CellFontSize = 8
MSFlexGrid1.CellAlignment = 4
MSFlexGrid1.CellBackColor = &HFFFFFF
'MSFlexGrid1.Text = average
MSFlexGrid1.Text = DATA * 2
'-----
Data1.Recordset.AddNew
Text2.Text = DATA * 2
Text3.Text = Time
Text4.Text = Date
Text5.Text = "KUS 12"
'-----
Else

MSFlexGrid1.Row = 4
MSFlexGrid1.Col = 2
MSFlexGrid1.CellFontSize = 8
MSFlexGrid1.CellAlignment = 4
MSFlexGrid1.CellBackColor = &HFFFFFF
'MSFlexGrid1.Text = average
MSFlexGrid1.Text = "No Signal"
End If

End If
DoEvents
TIME_INIT = Time
TIME_FINAL = Time
ECE = #12:01:00 AM#
Do Until (#12:00:05 AM# - #12:00:00 AM#) < (TIME_FINAL - TIME_INIT)
TIME_FINAL = Time
DoEvents
Loop
MSFlexGrid1.Row = 4

```

```

MSFlexGrid1.Col = 0
MSFlexGrid1.CellFontSize = 8
MSFlexGrid1.CellAlignment = 4
MSFlexGrid1.CellBackColor = &HFFFF80
MSFlexGrid1.Text = "KUS 12"
M = M + 1
Text1.Text = M

```

```
'Loop
```

```
'=====
```

```

'Reading flow rate of OUT PIR
DoEvents
MSFlexGrid3.Row = 19
MSFlexGrid3.Col = 0
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HC0&
MSFlexGrid3.Text = "OUT PIR"
WA = 99
DJ = 0
K = 2813.9
Transmit CU, WA, DJ
'-----PROGRAM RESET
If N = 1 Then
MSFlexGrid3.Row = 19
MSFlexGrid3.Col = 1
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HFFFFFF
MSFlexGrid3.Text = "reset prg"
End
End If
'-----
DoEvents

```

Receive DIGIT

DATA = (DGT3 * 16 ^ 3 + DGT2 * 16 ^ 2 + DGT1 * 16 + DGT0)

If N = 1 Then

If Form2.Label3 = "OUT PIR" Then

Form2.MSFlexGrid1.Row = 1

Form2.MSFlexGrid1.Col = 1

Form2.MSFlexGrid1.Text = DATA * 2

Form2.MSFlexGrid1.Row = 1

Form2.MSFlexGrid1.Col = 0

Form2.MSFlexGrid1.Text = DATA * 2 * K

End If

DoEvents

'-----

MSFlexGrid3.Row = 19

MSFlexGrid3.Col = 1

MSFlexGrid3.CellFontSize = 8

MSFlexGrid3.CellAlignment = 4

MSFlexGrid3.CellBackColor = &HFFFFFFF

MSFlexGrid3.Text = DATA * 2 * K

DoEvents

'-----

MSFlexGrid3.Row = 19

MSFlexGrid3.Col = 2

MSFlexGrid3.CellFontSize = 8

MSFlexGrid3.CellAlignment = 4

MSFlexGrid3.CellBackColor = &HFFFFFFF

'MSFlexGrid3.Text = average

MSFlexGrid3.Text = DATA * 2

'-----

Data1.Recordset.AddNew

Text2.Text = DATA * 2

Text3.Text = Time

Text4.Text = Date

Text5.Text = "OUT PIR"

```

'-----
End If
DoEvents
TIME_INIT = Time
TIME_FINAL = Time
ECE = #12:01:00 AM#
Do Until (#12:00:05 AM# - #12:00:00 AM#) < (TIME_FINAL - TIME_INIT)
TIME_FINAL = Time
DoEvents
Loop
MSFlexGrid3.Row = 19
MSFlexGrid3.Col = 0
MSFlexGrid3.CellFontSize = 8
MSFlexGrid3.CellAlignment = 4
MSFlexGrid3.CellBackColor = &HFFFF80
MSFlexGrid3.Text = "OUT PIR"
M = M + 1
End Sub
'-----

Public Sub Transmit(CU, WA, DJ)
'TIME_INIT
U = CU Mod 10
C = (CU - C) / 10
A = WA Mod 10
W = (WA - A) / 10
J = DJ Mod 10
D = (DJ - J) / 10
DoEvents
'-----

Out &H378, C
Out &H37A, &H1

N = 0
TIME_INIT = Time

```

```

TIME_FINAL = Time
Do Until (Inp(&H379) And &H80) = &H80
If (#12:00:03 AM# - #12:00:00 AM#) < (TIME_FINAL - TIME_INIT) Then Exit
Sub
TIME_FINAL = Time
N = 1
Loop
'-----
Out &H378, U
Out &H37A, &H0
N = 0
TIME_INIT = Time
TIME_FINAL = Time
Do Until (Inp(&H379) And &H80) = &H0
If (#12:00:01 AM# - #12:00:00 AM#) < (TIME_FINAL - TIME_INIT) Then Exit
Sub
TIME_FINAL = Time
N = 1
Loop
'-----
Out &H378, W
Out &H37A, &H1           'PIC'e oku mesaji gitti
N = 0
TIME_INIT = Time
TIME_FINAL = Time
Do Until (Inp(&H379) And &H80) = &H80
If (#12:00:01 AM# - #12:00:00 AM#) < (TIME_FINAL - TIME_INIT) Then Exit
Sub
TIME_FINAL = Time
N = 1
Loop           'from PIC as being an interrupt to indicate
'-----
Out &H378, A
Out &H37A, &H0

```



```

N = 0
TIME_INIT = Time
TIME_FINAL = Time
Do Until (Inp(&H379) And &H80) = &H0
If (#12:00:01 AM# - #12:00:00 AM#) < (TIME_FINAL - TIME_INIT) Then Exit
Sub
TIME_FINAL = Time
N = 1
Loop
'-----
Out &H378, D
Out &H37A, &H1
N = 0
TIME_INIT = Time
TIME_FINAL = Time
Do Until (Inp(&H379) And &H80) = &H80
If (#12:00:01 AM# - #12:00:00 AM#) < (TIME_FINAL - TIME_INIT) Then Exit
Sub
TIME_FINAL = Time
N = 1
Loop
'-----
Out &H378, J
Out &H37A, &H0
N = 0
TIME_INIT = Time
TIME_FINAL = Time
Do Until (Inp(&H379) And &H80) = &H0
If (#12:00:01 AM# - #12:00:00 AM#) < (TIME_FINAL - TIME_INIT) Then Exit
Sub
TIME_FINAL = Time
N = 1
Loop
N = 0

```

End Sub

Public Sub Read(DIGIT)

D3 = (Inp(&H379) And &H40)

D2 = (Inp(&H379) And &H20)

D1 = (Inp(&H379) And &H10)

D0 = (Inp(&H379) And &H8)

DIGIT = D3 / 64 * 2 ^ 3 + D2 / 32 * 2 ^ 2 + D1 / 16 * 2 + D0 / 8

End Sub

Public Sub Receive(DIGIT)

DGT3 = 0

DGT2 = 0

DGT1 = 0

DGT0 = 0

DoEvents

TIME_INIT = Time

N = 1

Do Until (Inp(&H379) And &H80) = &H80

If (#12:00:15 AM# - #12:00:00 AM#) < (TIME_FINAL - TIME_INIT) Then Exit

Sub

N = 0

TIME_FINAL = Time

DoEvents

Loop 'Oku sinyali IRQ_PIC=0, S7=1 olarak geliyor

DoEvents

Read DIGIT

Out &H37A, &H1

DGT3 = DIGIT

'-----

TIME_INIT = Time

N = 1

Do Until (Inp(&H379) And &H80) = &H0

```

If (#12:00:05 AM# - #12:00:00 AM#) < (TIME_FINAL - TIME_INIT) Then Exit
Sub
N = 0
TIME_FINAL = Time
DoEvents
Loop
DoEvents
Read DIGIT
Out &H37A, &H0
DGT2 = DIGIT
'-----
TIME_INIT = Time
N = 1
Do Until (Inp(&H379) And &H80) = &H80
If (#12:00:05 AM# - #12:00:00 AM#) < (TIME_FINAL - TIME_INIT) Then Exit
Sub
N = 0
TIME_FINAL = Time
DoEvents
Loop
DoEvents
Read DIGIT
Out &H37A, &H1
DGT1 = DIGIT
'-----
TIME_INIT = Time
N = 1
Do Until (Inp(&H379) And &H80) = &H0
If (#12:00:05 AM# - #12:00:00 AM#) < (TIME_FINAL - TIME_INIT) Then Exit
Sub
N = 0
TIME_FINAL = Time
DoEvents
Loop

```

```
DoEvents
```

```
N = 1
```

```
Read DIGIT
```

```
Out &H37A, &H0
```

```
DGT0 = DIGIT
```

```
End Sub
```

```
*****
```

```
Private Sub Timer1_Timer()
```

```
Label2 = Time
```

```
End Sub
```

EK - 2**ANA RTU ÜNİTESİ MİKROKONTROLÖR PROGRAMI**

```

;*****
;RTU_KIŞİSEL BILGISAYARLAR_.ASM is the name of KIŞİSEL
BILGISAYARLAR-Radio inteface.
; coming data from a KIŞİSEL BILGISAYARLAR is to be transmitted to a
DTMF according to the program
; written for PIC16F877.
;*****
    Title "RTU Program"
    list p=16F877      ; processor type
    include <P16F877.inc>
    __CONFIG _WDT_OFF & _HS_OSC & _PWRTE_OFF & _CP_OFF)
    Errorlevel -302

;-----
    CBLOCK    h'60'
    TMR
    TMRX
    JAPON
    DIG_NO           ;KIŞİSEL BILGISAYARLAR' den gelen
digitlerin sayisi
    DIG_NO_         ;KIŞİSEL BILGISAYARLAR' ye gidecek
digitlerin sayisi
    DAT_DIG           ;datanın islenecek bir hanesi
    ADDR_KIŞİSEL BILGISAYARLAR
    ADDR_RU
    RAD_ADDR         Header address
    DUTY
    KRK
    DUMMY
    WORK
    DATA_H
    DATA_L

```

```
PRJ
PRG
PRD
ENDC
#define DAT_MA PORTA, 0
#define REL_ST PORTA, 1
#define IRQ PORTB, 0
#define SEL_P PORTB, 1
#define SEL_MA PORTB, 2
#define SEL_RAD PORTB, 3
#define SEL_BS PORTB, 4
#define IRQ_KIŞISEL_BILGISAYARLAR PORTB, 5
#define PS PORTB, 6
#define DATA_P PORTB, 7
#define IRQ_PIC PORTC, 0
#define TX_RAD PORTC, 1
#define LE_L1 PORTC, 2
#define OE_L1 PORTC, 3
#define DIR_1 PORTC, 4
#define OE_1 PORTC, 5
#define DIR_2 PORTC, 6
#define OE_2 PORTC, 7
#define D0 PORTD, 0
#define D1 PORTD, 1
#define D2 PORTD, 2
#define D3 PORTD, 3
#define RD_ PORTD, 4
#define RS0 PORTD, 5
#define CS_ PORTD, 6
#define WR_ PORTD, 7
#define CM_LED PORTE, 0
#define RX_LED PORTE, 1
#define TX_LED PORTE, 2
```

-----;

```

ORG      h'00'
GOTO PORT_ARRANGE
ORG      h'03'
GOTO PORT_ARRANGE
ORG      h'04'
BTFSC    INTCON, RBIF
GOTO     INT_KIŞISEL_BILGISAYARLAR      ;Interrupt
KIŞISEL_BILGISAYARLAR' den geldi
BTFSC    INTCON, INTF
GOTO     INT_DTMF                        ;Interrupt disaridan geldi
BTFSC    PIR1, 0
GOTO     NURAN
GOTO     NURAN

;-----
PORT_ARRANGE
    MOVLW  b'00000000'                    ;All unmasked interrupts
disable, flags r clear
    MOVWF  INTCON
    BCF    STATUS, RP1
    BSF    STATUS, RP0                    ;BANK1 is selected
    MOVLW  b'00000001'                    ;TMR1 overflow interrupt
enabled
    MOVWF  PIE1
    MOVLW  b'111111'                      ;PORTA input
    MOVWF  TRISA
    MOVLW  b'11100001'                    ;PORTB
    MOVWF  TRISB
    MOVLW  b'00000000'                    ;PORTC
    MOVWF  TRISC
    MOVLW  b'00000000'                    ;PORTD
    MOVWF  TRISD
    MOVLW  b'00000000'                    ;PORTE
    MOVWF  TRISE
    MOVLW  b'0000101'                     ;PS=64

```

```

MOVWF    OPTION_REG
MOVLW    b'00001111'
MOVWF    ADCON1
BCF      STATUS, RP0
BCF      STATUS, RP1           ;BANK0 is selected
;-----
;initial conditions for PORTB
MOVLW    b'00000000'
MOVWF    PORTB
MOVLW    b'11101101'           ;here, directions of buffers r
assigned
MOVWF    PORTC
MOVLW    b'01010000'
MOVWF    PORTD
MOVLW    b'000'
MOVWF    PORTE

MOVLW    b'00110000'           ;PS: 1/8, internal CS, Timer1
stop
MOVWF    T1CON
;-----
MOVLW    h'01'
MOVWF    PRJ
MOVLW    h'02'
MOVWF    PRG
MOVLW    h'03'
MOVWF    PRD
MOVLW    h'04'
MOVWF    KRK
MOVLW    h'02'
MOVWF    RAD_ADDR
;-----
CALL     DELAY
CALL     DELAY                 ;(99+99)ms lagging

```



```

;-----
;Initialization of U1 (DTMF)
;1) Read Status Register
    CALL    PORTD_IN
    CALL    RD_SR
;2) Write to Control Register
    CALL    PORTD_OUT
    MOVLW   b'11110000'
    ANDWF   PORTD, F
    CALL    WR_CR
;3) Write to Control Register
    CALL    WR_CR
;4) Write to Control Register
    BSF     PORTD, 3
    CALL    WR_CR
;5) Write to Control Register
    BCF     PORTD, 3
    CALL    WR_CR
;6) Read Status Register
    CALL    PORTD_IN
    CALL    RD_SR
;-----
;Transmit DTMF tones of 50 ms burst/50 ms DELAY and Receive DTMF Tones
;7)Write to Control Register A
    CALL    PORTD_OUT
    BSF     PORTD, 0
    BCF     PORTD, 1
    BSF     PORTD, 2
    BSF     PORTD, 3
    CALL    WR_CR
;8)Write to Control Register B
    MOVLW   b'11110000'
    ANDWF   PORTD, F
    CALL    WR_CR

```

```

        CALL    DELAY
; Waiting for RX/TX interrupt
MERAL
        CALL    DELAY
        CALL    DELAY
        CALL    DELAY
NURAN
        BSF     CM_LED
        BCF     RX_LED
        BCF     TX_LED
;-----
        MOVLW   h'06'           ;6 adet sayi gonderilecek
        MOVWF   DIG_NO
        MOVLW   h'04'           ; 4 adet sayi gelecek
        MOVWF   DIG_NO_
        MOVLW   h'3A'
        MOVWF   FSR
        CLRF    TMR0
MARTI
        MOVLW   b'00110000'
        MOVWF   T1CON           ;Timer1 stop, Prescaler=1/8
        CLRF    PIR1           ;TMR1IF cleared
        CLRF    TMR1L
        CLRF    TMR1H
        MOVLW   b'11010000'     ;Global Int.enable and Peripheral Int.
enable
        MOVWF   INTCON         ;Timer0 interrupt disable
        BSF     TMR1ON, 0       ;Timer1 runs
WAIT
;Waiting of interrupt
        GOTO    WAIT
;=====
;Interrupt caused by KIŞISEL BILGISAYARLAR
INT_KIŞISEL BILGISAYARLAR           ;interrupt
from KIŞISEL BILGISAYARLAR

```

```

                BCF          INTCON, RBIF
;KIŞISEL BILGISAYARLAR den gelen datanın yerlesimi
;-----
;h'3A'<---X
;h'3B'<---Y
;h'3C'<---W
;h'3D'<---A
;h'3E'<---D
;h'3F'<---C
;-----
;Receiving from KIŞISEL BILGISAYARLAR and Transmitting to DTMF
                BSF          RX_LED
                CALL        PORTD_IN
                BCF          OE_1          ;buffer U3 enabled
                NOP
                CALL        READ_KIŞISEL BILGISAYARLAR      ;read data from
KIŞISEL BILGISAYARLAR, Data in DAT_DIG
                NOP
                BSF          OE_1          ;buffer U3 disabled
                MOVF        DAT_DIG, W
                MOVWF       INDF          ;data in INDF
                BCF          RX_LED

                BTFSS       FSR, 0
                GOTO        ODD_BIT
EVEN_BIT
                BCF          IRQ_PIC
                INCF        FSR, F
                GOTO        MARTI
ODD_BIT
                BSF          IRQ_PIC
                INCF        FSR, F
                BTFSS       FSR, 6
                GOTO        MARTI

```

```

        BCF      RX_LED
        BSF      IRQ_PIC          ;S7=1, KIŞISEL
BILGISAYARLAR'ye alindi mesajidir
;-----
;Receiving from KIŞISEL BILGISAYARLAR completed, now transmitting to
DTMF
;Checking destination address if it is to radio or B/S
;h'3A'<---X
;h'3B'<---Y
;h'3C'<---W
;h'3D'<---A
;h'3E'<---D
;h'3F'<---C
        MOVLW   h'3C'
        MOVWF   FSR
        MOVF    INDF, W
        SWAPF   W, W
        MOVWF   ADDR_RU
        INCF    FSR, F
        MOVF    INDF, W
        IORWF   ADDR_RU, F      ;Destination address is in ADDR_RU
        MOVLW   h'99'          ;Block station address
        BCF     STATUS, Z
        XORWF   ADDR_RU, W
        BTFSC   STATUS, Z
        GOTO    KURKAN_BS      ;Destination address is Kurkan B/S
selected
        BSF     TX_RAD          ;Destination address is Radio
;Receiving from KIŞISEL BILGISAYARLAR is completed now Transmitting via
radio starts
        CALL    DELAY
        CALL    DELAY
        CALL    DELAY
        MOVLW   h'3A'

```

```

MOVWF    FSR
BSF      SEL_RAD
SINAN
MOVF     INDF, W
MOVWF    DAT_DIG
CALL     TRANSMIT
INCF     FSR, F
BTFSS    FSR, 6
GOTO     SINAN
BCF      SEL_RAD
BCF      TX_RAD
GOTO     NURAN                ;Transmitting via radio completed
;=====
;=====
INT_DTMF                ;Interrupt from DTMF,Receiving from
Radio
BCF      INTCON, INTF
BSF      RX_LED

CALL     DOWNLOAD
INCF     FSR, F
DECFSZ   DIG_NO_, F        ;Baslangicta DIG_NO_' da 4 yuklu
GOTO     MARTI

;-----
;Header RTU_RU' den gelen datanın yerlestirilmesi:
;4 Digit geliyor
;h'3C'<---DIG_3
;h'3D'<---DIG_2
;h'3E'<---DIG_1
;h'3F'<---DIG_0
BCF      RX_LED
CALL     SEND_TO_KIŞISEL_BILGISAYARLAR
GOTO     NURAN
;-----

```

DOWNLOAD

```

        CLRF      INDF
        CALL     PORTD_IN
        CALL     RD_SR      ;Read Status Register of DTMF
        BTFSS   PORTD, 0   ;dtmf received a valid data
        RETFIE
        BTFSS   PORTD, 2
        RETFIE
        CALL     RD_RDR

        BTFSC   PORTD, 0   ;D0
        BSF     INDF, 0
        BTFSC   PORTD, 1   ;D1
        BSF     INDF, 1
        BTFSC   PORTD, 2   ;D2
        BSF     INDF, 2
        BTFSC   PORTD, 3   ;D3
        BSF     INDF, 3     ;data in INDF
        BCF     STATUS, Z
        MOVLW   h'0A'
        XORWF   INDF, W
        BTFSC   STATUS, Z
        CLRF    INDF
        RETURN

;-----
KURKAN_BS
;-----
;Datanin yerlestirilmesi:
;h'3A'<---X
;h'3B'<---Y
;-----
;h'3C'<---W
;h'3D'<---A
;-----

```

```

;h'3E'<---D                                ;Yapilacak isler
;h'3F'<---C
;-----
    MOVLW    h'3E'
    MOVWF   FSR
    MOVF    INDF, W
    SWAPF   W, W
    MOVWF   WORK
    INCF    FSR, F                ;h'3F'
    MOVF    INDF, W
    IORWF   WORK, F                ;yapilacak isler WORK' de
FL_0
    MOVF    WORK, W
    BCF     STATUS, Z
    XORLW   h'00'                ;FL_0 READ
    BTFSS   STATUS, Z
    GOTO    FL_1
    CALL    CLOSE_IF_RELAY
    CALL    READ_FLOW
    CALL    OPEN_IF_RELAY
    CALL    SEND_TO_KIŞISEL_BILGISAYARLAR
    GOTO    MERAL
FL_1
    MOVF    WORK, W
    BCF     STATUS, Z
    XORLW   h'01'                ;FL_1 READ
    BTFSS   STATUS, Z
    GOTO    FL_2
    CALL    CLOSE_IF_RELAY
    CALL    READ_FLOW
    CALL    OPEN_IF_RELAY
    CALL    SEND_TO_KIŞISEL_BILGISAYARLAR
    GOTO    MERAL
FL_2

```

```

MOVF    WORK, W
BCF     STATUS, Z
XORLW   h'02'           ;FL_2 READ
BTFSS   STATUS, Z
GOTO    FL_3
CALL    CLOSE_IF_RELAY
CALL    READ_FLOW
CALL    OPEN_IF_RELAY
CALL    SEND_TO_KIŞISEL_BILGISAYARLAR
GOTO    MERAL

FL_3
MOVF    WORK, W
BCF     STATUS, Z
XORLW   h'03'           ;FL_3 READ
BTFSS   STATUS, Z
GOTO    FL_4
CALL    CLOSE_IF_RELAY
CALL    READ_FLOW
CALL    OPEN_IF_RELAY
CALL    SEND_TO_KIŞISEL_BILGISAYARLAR
GOTO    MERAL

FL_4
MOVF    WORK, W
BCF     STATUS, Z
XORLW   h'04'           ;FL_4 READ
BTFSS   STATUS, Z
GOTO    FL_5
CALL    CLOSE_IF_RELAY
CALL    READ_FLOW
CALL    OPEN_IF_RELAY
CALL    SEND_TO_KIŞISEL_BILGISAYARLAR
GOTO    MERAL

FL_5
MOVF    WORK, W

```



```

BCF      STATUS, Z
XORLW   h'05'          ;FL_5 READ
BTFSS   STATUS, Z
GOTO    FL_6
CALL    CLOSE_IF_RELAY
CALL    READ_FLOW
CALL    OPEN_IF_RELAY
CALL    SEND_TO_KIŞISEL_BILGISAYARLAR
GOTO    MERAL

FL_6
MOVF    WORK, W
BCF     STATUS, Z
XORLW   h'06'          ;FL_6 READ
BTFSS   STATUS, Z
GOTO    FL_7
CALL    CLOSE_IF_RELAY
CALL    READ_FLOW
CALL    OPEN_IF_RELAY
CALL    SEND_TO_KIŞISEL_BILGISAYARLAR
GOTO    MERAL

FL_7
MOVF    WORK, W
BCF     STATUS, Z
XORLW   h'07'          ;FL_7 READ
BTFSS   STATUS, Z
GOTO    FL_8
CALL    CLOSE_IF_RELAY
CALL    READ_FLOW
CALL    OPEN_IF_RELAY
CALL    SEND_TO_KIŞISEL_BILGISAYARLAR
GOTO    MERAL

FL_8
MOVF    WORK, W
BCF     STATUS, Z

```

```

XORLW    h'08'                ;FL_8 READ
BTSS     STATUS, Z
GOTO     FL_9
CALL     CLOSE_IF_RELAY
CALL     READ_FLOW
CALL     OPEN_IF_RELAY
CALL     SEND_TO_KIŞISEL_BILGISAYARLAR
GOTO     MERAL

FL_9
MOVF     WORK, W
BCF      STATUS, Z
XORLW    h'09'                ;FL_9 READ
BTSS     STATUS, Z
GOTO     FL_10
CALL     CLOSE_IF_RELAY
CALL     READ_FLOW
CALL     OPEN_IF_RELAY
CALL     SEND_TO_KIŞISEL_BILGISAYARLAR
GOTO     MERAL

FL_10
MOVF     WORK, W
BCF      STATUS, Z
XORLW    h'0A'                ;FL_10 READ
BTSS     STATUS, Z
GOTO     FL_11
CALL     CLOSE_IF_RELAY
CALL     READ_FLOW
CALL     OPEN_IF_RELAY
CALL     SEND_TO_KIŞISEL_BILGISAYARLAR
GOTO     MERAL

FL_11
MOVF     WORK, W
BCF      STATUS, Z
XORLW    h'0B'                ;FL_11 READ

```

```

BTSS    STATUS, Z
GOTO    FL_12
CALL    CLOSE_IF_RELAY
CALL    READ_FLOW
CALL    OPEN_IF_RELAY
CALL    SEND_TO_KIŞISEL_BILGISAYARLAR
GOTO    MERAL

```

FL_12

```

MOVF    WORK, W
BCF     STATUS, Z
XORLW   h'0C'           ;FL_12 READ
BTSS    STATUS, Z
GOTO    FL_13
CALL    CLOSE_IF_RELAY
CALL    READ_FLOW
CALL    OPEN_IF_RELAY
CALL    SEND_TO_KIŞISEL_BILGISAYARLAR
GOTO    MERAL

```

FL_13

```

MOVF    WORK, W
BCF     STATUS, Z
XORLW   h'0D'           ;FL_13 READ
BTSS    STATUS, Z
GOTO    FL_14
CALL    CLOSE_IF_RELAY
CALL    READ_FLOW
CALL    OPEN_IF_RELAY
CALL    SEND_TO_KIŞISEL_BILGISAYARLAR
GOTO    MERAL

```

FL_14

```

MOVF    WORK, W
BCF     STATUS, Z
XORLW   h'0E'           ;FL_14 READ
BTSS    STATUS, Z

```

```

GOTO    FL_15
CALL    CLOSE_IF_RELAY
CALL    READ_FLOW
CALL    OPEN_IF_RELAY
CALL    SEND_TO_KIŞISEL_BILGISAYARLAR
GOTO    MERAL

```

FL_15

```

MOVWF   WORK, W
BCF     STATUS, Z
XORLW   h'0F'           ;FL_15 READ
BTFSS   STATUS, Z
GOTO    FL_16
CALL    CLOSE_IF_RELAY
CALL    READ_FLOW
CALL    OPEN_IF_RELAY
CALL    SEND_TO_KIŞISEL_BILGISAYARLAR
GOTO    MERAL

```

FL_16

```

MOVWF   WORK, W
BCF     STATUS, Z
XORLW   h'10'           ;FL_16 READ
BTFSS   STATUS, Z
GOTO    FL_17
CALL    CLOSE_IF_RELAY
CALL    READ_FLOW
CALL    OPEN_IF_RELAY
CALL    SEND_TO_KIŞISEL_BILGISAYARLAR
GOTO    MERAL

```

FL_17

```

MOVWF   WORK, W
BCF     STATUS, Z
XORLW   h'11'           ;FL_17 READ
BTFSS   STATUS, Z
GOTO    FL_18

```

```
CALL    CLOSE_IF_RELAY
CALL    READ_FLOW
CALL    OPEN_IF_RELAY
CALL    SEND_TO_KIŞISEL_BILGISAYARLAR
GOTO    MERAL

FL_18
MOVF    WORK, W
BCF     STATUS, Z
XORLW   h'12'           ;FL_18 READ
BTFSS   STATUS, Z
GOTO    FL_19
CALL    CLOSE_IF_RELAY
CALL    READ_FLOW
CALL    OPEN_IF_RELAY
CALL    SEND_TO_KIŞISEL_BILGISAYARLAR
GOTO    MERAL

FL_19
MOVF    WORK, W
BCF     STATUS, Z
XORLW   h'13'           ;FL_19 READ
BTFSS   STATUS, Z
GOTO    FL_20
CALL    CLOSE_IF_RELAY
CALL    READ_FLOW
CALL    OPEN_IF_RELAY
CALL    SEND_TO_KIŞISEL_BILGISAYARLAR
GOTO    MERAL

FL_20
MOVF    WORK, W
BCF     STATUS, Z
XORLW   h'14'           ;FL_20 READ
BTFSS   STATUS, Z
GOTO    MERAL
CALL    CLOSE_IF_RELAY
```

```

CALL    READ_FLOW
CALL    OPEN_IF_RELAY
CALL    SEND_TO_KIŞISEL_BILGISAYARLAR
GOTO    MERAL

```

```

;-----

```

```

CLOSE_IF_RELAY                                ;Close I/F signal relay

```

```

BSF     SEL_BS
MOVLW   h'3E'
MOVWF   FSR
MOVF    INDF, W
MOVWF   DAT_DIG
CALL    TRANSMIT
INCF    FSR, F
MOVF    INDF, W
MOVWF   DAT_DIG
CALL    TRANSMIT
MOVLW   h'00'
MOVWF   DAT_DIG
CALL    TRANSMIT
MOVLW   h'01'
MOVWF   DAT_DIG
CALL    TRANSMIT
BCF     SEL_BS
RETURN

```

```

;-----

```

```

OPEN_IF_RELAY                                ;Open I/F signal relay

```

```

BSF     SEL_BS
MOVLW   h'3E'
MOVWF   FSR
MOVF    INDF, W
MOVWF   DAT_DIG
CALL    TRANSMIT
INCF    FSR, F
MOVF    INDF, W

```

```

MOVWF    DAT_DIG
CALL     TRANSMIT
MOVLW   h'00'
MOVWF    DAT_DIG
CALL     TRANSMIT
MOVLW   h'00'
MOVWF    DAT_DIG
CALL     TRANSMIT
BCF      SEL_BS
RETURN

```

```

;-----

```

```

PORTD_IN

```

```

BSF      STATUS, RP0
MOVLW   b'00001111'    ;PORTD input
MOVWF    TRISD
BCF      STATUS, RP0
RETURN

```

```

;-----

```

```

PORTD_OUT

```

```

BSF      STATUS, RP0
MOVLW   b'00000000'    ;PORTD output
MOVWF    TRISD
BCF      STATUS, RP0
RETURN

```

```

;-----

```

```

RD_SR

```

```

BSF      RS0
BCF      CS_
BCF      RD_
NOP
BSF      RD_
BSF      CS_
BCF      RS0
RETURN

```

;-----

RD_RDR

```
    BCF    RS0
    BCF    CS_
    BCF    RD_
    NOP
    BSF    RD_
    BSF    CS_
    BCF    RS0
    RETURN
```

;-----

WR_CR

```
    BSF    RS0
    BCF    CS_
    BCF    WR_
    NOP
    BSF    WR_
    BSF    CS_
    BCF    RS0
    RETURN
```

;-----

WR_TDR

```
    BCF    RS0
    BCF    CS_
    BCF    WR_
    NOP
    BSF    WR_
    BSF    CS_
    BCF    RS0
    RETURN
```

;-----

READ_KIŞİSEL BILGISAYARLAR

```
    MOVF   PORTD, W
    ANDLW  b'00001111'
```



```

        MOVWF    DAT_DIG                ;KIŞISEL BILGISAYARLAR'
den gelen data hem W' de hemde DAT_DIG' de
;      BCF      STATUS, Z
;      XORLW    h'00'
;      BTFSS    STATUS, Z
;      RETURN
;      MOVLW    h'0A'
;      MOVWF    DAT_DIG                ;PORTD--->DAT_DIG
      RETURN
;-----
DELAY
      MOVLW    h'06'
      MOVWF    TMR
      CLRF     TMR0
      BCF      INTCON, T0IF
LOOP_DELAY
      BTFSS    INTCON, T0IF
      GOTO     LOOP_DELAY
      BCF      INTCON, T0IF
      DECFSZ   TMR, F                  ;(6 X 64 X 256 X 1us)= 98ms
      GOTO     LOOP_DELAY
      RETURN
;-----
DELAY_                ;(1 X 64 X 256 X 1us)= 16ms
      CLRF     TMR0
      BCF      INTCON, T0IF
LOOP_DELAY_
      BTFSS    INTCON, T0IF
      GOTO     LOOP_DELAY_
      RETURN
;-----
READ_FLOW
      BSF      SEL_P
;Duty Code for closing I/F relay is 01, for opening I/F relay it is 00

```

```

        CLRFB        DATA_L
        CLRFB        DATA_H
;        MOVLWB      h'3D'                ;30 x 64 x 256 x 1 usec = 0.4915 sec.
        MOVLWB      h'1E'
        MOVWB        TMRX

        CLRFB        TMR0                ;Pulse basladi, Timer sifirlandi
        BCF          INTCON, T0IF        ;Timer flag silindi

WAIT_POS_PLS
        BTFSC        INTCON, T0IF
        GOTO         DECR_TMR_1
        BTFSS        PORTB, 7
        GOTO         WAIT_POS_PLS

WAIT_NEG_PLS
        BTFSC        INTCON, T0IF
        GOTO         DECR_TMR_0
        BTFSC        PORTB, 7
        GOTO         WAIT_NEG_PLS

        INCFSZ       DATA_L, F
        GOTO         WAIT_POS_PLS
        INCF         DATA_H, F
        GOTO         WAIT_POS_PLS
;-----
DECR_TMR_1
        BCF          INTCON, T0IF        ;Timer flag silindi
        DECFSZ       TMRX, F
        GOTO         WAIT_POS_PLS
        GOTO         MERVE

DECR_TMR_0
        BCF          INTCON, T0IF        ;Timer flag silindi
        DECFSZ       TMRX, F

```

```

        GOTO    WAIT_NEG_PLS
        GOTO    MERVE
MERVE
;    MOVLW    h'12'
;    MOVWF    DATA_H
;    MOVLW    h'34'
;    MOVWF    DATA_L

;    90 <--- DATA_HH
;    91 <--- DATA_HL
;    92 <--- DATA_LH
;    93 <--- DATA_LL
        MOVLW    h'90'
        MOVWF    FSR
        CLRF     INDF

        BTFSC   DATA_H, 7
        BSF     INDF, 3
        BTFSC   DATA_H, 6
        BSF     INDF, 2
        BTFSC   DATA_H, 5
        BSF     INDF, 1
        BTFSC   DATA_H, 4
        BSF     INDF, 0

        INCF    FSR, F
        CLRF    INDF

        BTFSC   DATA_H, 3
        BSF     INDF, 3
        BTFSC   DATA_H, 2
        BSF     INDF, 2
        BTFSC   DATA_H, 1
        BSF     INDF, 1

```

```

    BTFSC    DATA_H, 0
    BSF      INDF, 0
;-----
    INCF     FSR, F
    CLRF     INDF

    BTFSC    DATA_L, 7
    BSF      INDF, 3
    BTFSC    DATA_L, 6
    BSF      INDF, 2
    BTFSC    DATA_L, 5
    BSF      INDF, 1
    BTFSC    DATA_L, 4
    BSF      INDF, 0

    INCF     FSR, F
    CLRF     INDF

    BTFSC    DATA_L, 3
    BSF      INDF, 3
    BTFSC    DATA_L, 2
    BSF      INDF, 2
    BTFSC    DATA_L, 1
    BSF      INDF, 1
    BTFSC    DATA_L, 0
    BSF      INDF, 0

    BCF      SEL_P
    RETURN
;-----
SEND_TO_KİŞİSEL_BİLGİSAYARLAR
    MOVLW    h'90'
    MOVWF    FSR
;1. Digit gönderiliyor

```

```

CALL    WRITE_TO_PORTD
BCF     OE_2           ;KIŞİSEL BILGISAYARLAR'ye data
girisi acildi
BCF     IRQ_PIC       ;KIŞİSEL BILGISAYARLAR uyarildi
WAIT_FOR_DIG3
BTFS   IRQ_KIŞİSEL BILGISAYARLAR
GOTO    WAIT_FOR_DIG3
BSF     OE_2           ;KIŞİSEL BILGISAYARLAR'ye data
girisi kapandi
INCF    FSR, F
;2. Digit gonderiliyor
CALL    WRITE_TO_PORTD
BCF     OE_2           ;KIŞİSEL BILGISAYARLAR'ye data
girisi acildi
BSF     IRQ_PIC       ;KIŞİSEL BILGISAYARLAR
uyarildi
WAIT_FOR_DIG2
BTFS   IRQ_KIŞİSEL BILGISAYARLAR
GOTO    WAIT_FOR_DIG2
BSF     OE_2           ;KIŞİSEL BILGISAYARLAR'ye data
girisi kapandi
INCF    FSR, F
;3. Digit gonderiliyor
CALL    WRITE_TO_PORTD
BCF     OE_2           ;KIŞİSEL BILGISAYARLAR'ye data
girisi acildi
BCF     IRQ_PIC       ;KIŞİSEL BILGISAYARLAR
uyarildi
WAIT_FOR_DIG1
BTFS   IRQ_KIŞİSEL BILGISAYARLAR
GOTO    WAIT_FOR_DIG1
BSF     OE_2           ;KIŞİSEL BILGISAYARLAR'ye data
girisi kapandi
INCF    FSR, F

```

;4. Digit gönderiliyor

```

    CALL    WRITE_TO_PORTD
    BCF     OE_2           ;KIŞİSEL BILGISAYARLAR'ye data
giris acildi
    BSF     IRQ_PIC       ;KIŞİSEL BILGISAYARLAR
uyarildi
WAIT_FOR_DIG0
    BTFSS   IRQ_KIŞİSEL BILGISAYARLAR
    GOTO    WAIT_FOR_DIG0
    BSF     OE_2           ;KIŞİSEL BILGISAYARLAR'ye data
giris kapandi
    RETURN

```

;-----

WRITE_TO_PORTD

```

    CALL    PORTD_OUT
    MOVLW   h'F0'
    ANDWF   PORTD, F      ;PORTD temizlendi
    BTFSC   INDF, 0
    BSF     PORTD, 0
    BTFSC   INDF, 1
    BSF     PORTD, 1
    BTFSC   INDF, 2
    BSF     PORTD, 2
    BTFSC   INDF, 3
    BSF     PORTD, 3
    RETURN

```

;-----

TRANSMIT

```

    BSF     TX_LED
    CALL    ZERO_TO_DTMF
    CALL    PORTD_OUT
    MOVLW   h'F0'
    ANDWF   PORTD, F
    BTFSC   DAT_DIG, 0

```

```

BSF      PORTD, 0
BTFSC   DAT_DIG, 1
BSF      PORTD, 1
BTFSC   DAT_DIG, 2
BSF      PORTD, 2
BTFSC   DAT_DIG, 3
BSF      PORTD, 3
CALL    TX_TO_DTMF
BCF      TX_LED
RETURN

```

```

;-----

```

```

ZERO_TO_DTMF

```

```

MOVWF   DAT_DIG, W
BCF     STATUS, Z
XORLW   h'00'
BTFSS   STATUS, Z
RETURN
MOVLW   h'0A'
MOVWF   DAT_DIG
RETURN

```

```

;-----

```

```

TX_TO_DTMF

```

```

;transmitting to DTMF

```

```

procedure

```

```

CALL    WR_TDR           ;Data gonderildi
CALL    DELAY_
CALL    DELAY_
CALL    DELAY_
CALL    DELAY_
CALL    DELAY_
CALL    PORTD_IN

```

```

;Timer1 zamani saymaya basladi

```

```

CLRF    TMR1L
CLRF    TMR1H

```

```
        MOVLW    b'11000000'        ;Global and Peripheral Interrupts are
enabled
        MOVWF   INTCON
        BSF     T1CON, 0            ;Timer1 enabled
        CALL    RD_SR
LOOP
        BTFSS  PORTD, 1            ;TDR ready for another data
transmitting
        GOTO   LOOP
        RETURN
;-----
        END
```


EK – 3

UZAKTAKİ (REMOTE) RTU ÜNİTESİ MİKROKONTROLÖR PROGRAMI

```

;*****
;REMOTE_RTU.ASM is the name of CENTRAL_REMOTE.
;Coming 6 bit data from a KIŞİSEL BILGISAYARLAR is to be received via
DTMF according to the program
;written for PIC16F877.
;*****

    Title "RTU Program"
    list p=16f877    ; processor type
    include <P16f877.inc>
    __CONFIG    ( _WDT_OFF & _HS_OSC & _PWRTE_OFF & _CP_OFF)
    errorlevel -302

;-----
    CBLOCK    h'60'
    TMR
    TMRX
    JAPON
    DIG_NO                    ;KIŞİSEL BILGISAYARLAR' den
gelen digitlerin sayisi
    DIG_NO_                    ;KIŞİSEL BILGISAYARLAR' ye
gidecek digitlerin sayisi
    DAT_DIG                    ;datanın islenecek bir hanesi
    XY
    WA
    RAD_ADR                    ;Header address
    KRK
    DUMMY
    WORK
    DATA_H
    DATA_L
    PRJ
    PRG

```

PRD

ABARI

ENDC

#DEFINE DAT_MA PORTA, 0

#DEFINE REL_ST PORTA, 1

#DEFINE IRQ PORTB, 0

#DEFINE SEL_P PORTB, 1

#DEFINE SEL_MA PORTB, 2

#DEFINE SEL_RAD PORTB, 3

#DEFINE SEL_BS PORTB, 4

#DEFINE IRQ_KIŞISEL BILGISAYARLAR PORTB, 5

#DEFINE PS PORTB, 6

#DEFINE DATA_P PORTB, 7

#DEFINE IRQ_PIC PORTC, 0

#DEFINE TX_RAD PORTC, 1

#DEFINE LE_L1 PORTC, 2

#DEFINE OE_L1 PORTC, 3

#DEFINE DIR_1 PORTC, 4

#DEFINE OE_1 PORTC, 5

#DEFINE DIR_2 PORTC, 6

#DEFINE OE_2 PORTC, 7

#DEFINE D0 PORTD, 0

#DEFINE D1 PORTD, 1

#DEFINE D2 PORTD, 2

#DEFINE D3 PORTD, 3

#DEFINE RD_ PORTD, 4

#DEFINE RS0 PORTD, 5

#DEFINE CS_ PORTD, 6

#DEFINE WR_ PORTD, 7

```

#DEFINE    CM_LED    PORTE, 0
#DEFINE    RX_LED    PORTE, 1
#DEFINE    TX_LED    PORTE, 2
;-----
ORG        h'00'
GOTO      PORT_ARRANGE
ORG        h'03'
GOTO      PORT_ARRANGE
ORG        h'04'
BTFSC     INTCON, INTF      ;Interrupt yalnızca baska radyolardan
geliyor
GOTO      INT_DTMF
BTFSC     PIR1, 0
GOTO      NURAN
RETURN
;-----
PORT_ARRANGE
MOVLW     b'00000000'      ;All unmasked interrupts disable, flags r
clear
MOVWF     INTCON
;-----
;Port angements
BCF       STATUS, RP1
BSF       STATUS, RP0      ;BANK1 is selected
;-----
MOVLW     b'00000001'      ;TMR1 overflow interrupt enabled
MOVWF     PIE1
MOVLW     b'111111'        ;PORTA input
MOVWF     TRISA
MOVLW     b'11100001'      ;PORTB
MOVWF     TRISB
MOVLW     b'00000000'      ;PORTC
MOVWF     TRISC
MOVLW     b'00000000'      ;PORTD

```

```

MOVWF    TRISD
MOVLW    b'00000000'    ;PORTE
MOVWF    TRISE
;-----
MOVLW    b'0000101'    ;PS=64
MOVWF    OPTION_REG
;-----
;A/D Conv. Configuration Register
MOVLW    b'00101111'    ;Right justified, AN0 selected as analog
input
MOVWF    ADCON1
;-----
BCF      STATUS, RP0
BCF      STATUS, RP1    ;BANK0 is selected
;-----
;initial conditions for PORTB
MOVLW    b'00000000'
MOVWF    PORTB
MOVLW    b'11101101'    ;here, directions of buffers r assigned
MOVWF    PORTC
MOVLW    b'01010000'
MOVWF    PORTD
MOVLW    b'000'
MOVWF    PORTE

MOVLW    b'00110000'    ;PS: 1/8, internal CS, Timer1 stop
MOVWF    T1CON          ;A/D Conv. Control Register
MOVLW    b'11000000'

;A/D Conv. Clock is Int.RC osc.,CH0(=AN0) selected, A/D Conv. module is shut
off,
MOVWF    ADCON0
;-----
MOVLW    h'01'
MOVWF    PRJ

```

```

    MOVLW    h'02'
    MOVWF    PRG
    MOVLW    h'03'
    MOVWF    PRD
    MOVLW    h'04'
    MOVWF    KRK
    MOVLW    h'04'
    MOVWF    RAD_ADR

;-----
    CALL     DELAY
    CALL     DELAY           ;(99+99)ms lagging
;-----

;Initialization of U1 (DTMF)
;1) Read Status Register
    CALL     PORTD_IN
    CALL     RD_SR
;2) Write to Control Register
    CALL     PORTD_OUT
;
    CLRF     PORTD
    MOVLW    b'11110000'
    ANDWF    PORTD, F
    CALL     WR_CR
;3) Write to Control Register
    CALL     WR_CR
;4) Write to Control Register
    BSF     PORTD, 3
    CALL     WR_CR
;5) Write to Control Register
    BCF     PORTD, 3
    CALL     WR_CR
;6) Read Status Register
    CALL     PORTD_IN
    CALL     RD_SR
;-----

```

;Transmit DTMF tones of 50 ms burst/50 ms pause and Receive DTMF Tones

;7)Write to Control Register A

```
CALL    PORTD_OUT
BSF     PORTD, 0
BCF     PORTD, 1
BSF     PORTD, 2
BSF     PORTD, 3
CALL    WR_CR
```

;8)Write to Control Register B

```
;    CLRF    PORTD
MOVLW   b'11110000'
ANDWF   PORTD, F
CALL    WR_CR
```

;*****Waiting for RX/TX Interrupt*****

MERAL

```
CALL    DELAY
CALL    DELAY
CALL    DELAY
```

;-----

NURAN

```
BSF     CM_LED
BCF     RX_LED
BCF     TX_LED
```

;-----

```
MOVLW   h'3A'
```

;Master Unitlerden gelen data h'34' dan

baslayarak yerlestiriliyor

```
MOVWF   FSR
CLRF    TMR0
```

MARTI

```
MOVLW   b'00110000'
MOVWF   T1CON
CLRF    PIR1
CLRF    TMR1L
```

;Timer1 stop, Prescaler=1/8

;TMR1IF cleared

```

        CLRF      TMR1H
        MOVLW    b'11010000'      ;Global Int.enable and Peripheral Int.
enable
        MOVWF   INTCON            ;Timer0 interrupt disable
        BSF     TMR1ON, 0        ;Timer1 runs
WAIT
        GOTO    WAIT            ;Waiting of interrupt

```

```

;-----

```

```

;Data receiving

```

```

INT_DTMF

```

```

        BCF     INTCON, INTF
        BSF     RX_LED
        CALL    DOWNLOAD
        INCF   FSR, F
        BTFSS  FSR, 6
        GOTO    MARTI

```

```

;Beklenen 6 digit gelmistir ve adres dogrulasina gidiliyor

```

```

;Datanin yerlestirilmesi:

```

```

;h'3A'<---X

```

```

;h'3B'<---Y

```

```

;h'3C'<---W

```

```

;h'3D'<---A

```

```

;h'3E'<---D

```

```

;h'3F'<---J

```

```

        MOVLW   h'3A'
        MOVWF   FSR
        SWAPF   INDF, W
        INCF   FSR, F            ;h'3B'
        IORWF   INDF, W
        MOVWF   XY              ;hangi KIŞISEL BILGISAYARLAR'

```

```

nin konustugu bilgisi ADDR_KIŞISEL BILGISAYARLAR' de

```

```

        MOVLW   h'3C'
        MOVWF   FSR
        MOVF    INDF, W

```

```

SWAPF      W, W
INCF  FSR, F           ;h'3D'
IORWF      INDF, W
MOVWF      WA           ;RU' in adresi ADDR_RU' de
MOVLW      h'3E'
MOVWF      FSR
MOVFINDF, W
SWAPF      W, W
INCF  FSR, F           ;h'3F'
IORWF      INDF, W
MOVWF      WORK        ;RU' in yapacagi gorev WORK' de
MOVF      XY, W
BCF        STATUS, Z
XORWF      PRJ, W
BTFS      STATUS, Z
GOTO      PROJECT      ;Emir Sukru KIŞISEL

```

BILGISAYARLAR' den geldi

```

MOVF      XY, W
BCF        STATUS, Z
XORWF      PRG, W
BTFS      STATUS, Z
GOTO      PROGRAMMING ;Emir Programming KIŞISEL

```

BILGISAYARLAR' den geldi

```

MOVF      XY, W
BCF        STATUS, Z
XORWF      KRK, W
BTFS      STATUS, Z
GOTO      KURKAN       ;Emir Kurkan KIŞISEL

```

BILGISAYARLAR' den geldi

```

MOVF      XY, W
BCF        STATUS, Z

```



```

XORWF    PRD, W
BTFS    STATUS, Z
GOTO     PRODUCTION    ;Emir Production KİŞİSEL
BILGISAYARLAR' den geldi
GOTO     NURAN          ;Emir yetkili hicbir KİŞİSEL
BILGISAYARLAR' den gelmedi
PROJECT
GOTO     NURAN
PROGRAMMING
GOTO     NURAN
PRODUCTION
GOTO     NURAN
KURKAN

MOVF     WA, W          ;Header adresi
BCF      STATUS, Z
XORWF    RAD_ADR, W
BTFS    STATUS, Z
GOTO     NURAN        ;Emirin kendisine gelmedigini anladi, geri
dondu

MOVLW    h'3E'
MOVWF    FSR
MOVF     INDF, W
SWAPF    W, W
MOVWF    WORK
INCF     FSR, F        ;h'3F'
MOVF     INDF, W
IORWF    WORK, F      ;yapilacak isler WORK' de
;-----
FL_0
MOVF     WORK, W
BCF      STATUS, Z
XORLW    h'00'        ;FL_0 READ

```

```

    BTFSS    STATUS, Z
    GOTO     FL_1
    CALL     CLOSE_IF_RELAY
    CALL     READ_FLOW
    CALL     OPEN_IF_RELAY
    CALL     TRANSMIT_DATA
    CALL     RELAY_TEST
    GOTO     NURAN

FL_1
    MOVF     WORK, W
    BCF      STATUS, Z
    XORLW    h'01'           ;FL_1 READ
    BTFSS    STATUS, Z
    GOTO     FL_2
    CALL     CLOSE_IF_RELAY
    CALL     READ_FLOW
    CALL     OPEN_IF_RELAY
    CALL     TRANSMIT_DATA
    CALL     RELAY_TEST
    GOTO     NURAN

FL_2
    MOVF     WORK, W
    BCF      STATUS, Z
    XORLW    h'02'           ;FL_2 READ
    BTFSS    STATUS, Z
    GOTO     FL_3
    CALL     CLOSE_IF_RELAY
    CALL     READ_FLOW
    CALL     OPEN_IF_RELAY
    CALL     TRANSMIT_DATA
    CALL     RELAY_TEST
    GOTO     NURAN

FL_3
    MOVF     WORK, W

```

```

BCF      STATUS, Z
XORLW   h'03'          ;FL_3 READ
BTFSS   STATUS, Z
GOTO    FL_4
CALL    CLOSE_IF_RELAY
CALL    READ_FLOW
CALL    OPEN_IF_RELAY
CALL    TRANSMIT_DATA
CALL    RELAY_TEST
GOTO    NURAN

FL_4
MOVF    WORK, W
BCF     STATUS, Z
XORLW   h'04'          ;FL_4 READ
BTFSS   STATUS, Z
GOTO    FL_5
CALL    CLOSE_IF_RELAY
CALL    READ_FLOW
CALL    OPEN_IF_RELAY
CALL    TRANSMIT_DATA
CALL    RELAY_TEST
GOTO    NURAN

FL_5
MOVF    WORK, W
BCF     STATUS, Z
XORLW   h'05'          ;FL_5 READ
BTFSS   STATUS, Z
GOTO    FL_6
CALL    CLOSE_IF_RELAY
CALL    READ_FLOW
CALL    OPEN_IF_RELAY
CALL    TRANSMIT_DATA
CALL    RELAY_TEST
GOTO    NURAN

```

```
FL_6
    MOVF     WORK, W
    BCF      STATUS, Z
    XORLW   h'06'           ;FL_6 READ
    BTFSS   STATUS, Z
    GOTO    FL_7
    CALL    CLOSE_IF_RELAY
    CALL    READ_FLOW
    CALL    OPEN_IF_RELAY
    CALL    TRANSMIT_DATA
    CALL    RELAY_TEST
    GOTO    NURAN

FL_7
    MOVF     WORK, W
    BCF      STATUS, Z
    XORLW   h'07'           ;FL_7 READ
    BTFSS   STATUS, Z
    GOTO    FL_8
    CALL    CLOSE_IF_RELAY
    CALL    READ_FLOW
    CALL    OPEN_IF_RELAY
    CALL    TRANSMIT_DATA
    CALL    RELAY_TEST
    GOTO    NURAN

FL_8
    MOVF     WORK, W
    BCF      STATUS, Z
    XORLW   h'08'           ;FL_8 READ
    BTFSS   STATUS, Z
    GOTO    FL_9
    CALL    CLOSE_IF_RELAY
    CALL    READ_FLOW
    CALL    OPEN_IF_RELAY
    CALL    TRANSMIT_DATA
```

```

        CALL    RELAY_TEST
        GOTO    NURAN
FL_9
        MOVF    WORK, W
        BCF     STATUS, Z
        XORLW   h'09'           ;FL_9 READ
        BTFSS  STATUS, Z
        GOTO    FL_10
        CALL    CLOSE_IF_RELAY
        CALL    READ_FLOW
        CALL    OPEN_IF_RELAY
        CALL    RELAY_TEST
        GOTO    NURAN
FL_10
        MOVF    WORK, W
        BCF     STATUS, Z
        XORLW   h'10'           ;FL_10 READ
        BTFSS  STATUS, Z
        GOTO    NURAN
        CALL    CLOSE_IF_RELAY
        CALL    READ_FLOW
        CALL    OPEN_IF_RELAY
        CALL    TRANSMIT_DATA
        CALL    RELAY_TEST
        GOTO    NURAN
FL_11
        MOVF    WORK, W
        BCF     STATUS, Z
        XORLW   h'11'           ;FL_11 READ
        BTFSS  STATUS, Z
        GOTO    NURAN
        CALL    CLOSE_IF_RELAY
        CALL    READ_FLOW
        CALL    OPEN_IF_RELAY

```

```
CALL    TRANSMIT_DATA
CALL    RELAY_TEST
GOTO    NURAN

FL_12
MOVF    WORK, W
BCF     STATUS, Z
XORLW   h'12'           ;FL_12 READ
BTFSS   STATUS, Z
GOTO    NURAN
CALL    CLOSE_IF_RELAY
CALL    READ_FLOW
CALL    OPEN_IF_RELAY
CALL    TRANSMIT_DATA
CALL    RELAY_TEST
GOTO    NURAN

FL_13
MOVF    WORK, W
BCF     STATUS, Z
XORLW   h'13'           ;FL_13 READ
BTFSS   STATUS, Z
GOTO    NURAN
CALL    CLOSE_IF_RELAY
CALL    READ_FLOW
CALL    OPEN_IF_RELAY
CALL    TRANSMIT_DATA
CALL    RELAY_TEST
GOTO    NURAN

FL_14
MOVF    WORK, W
BCF     STATUS, Z
XORLW   h'14'           ;FL_14 READ
```

```

BTFSS    STATUS, Z
GOTO     NURAN
CALL     CLOSE_IF_RELAY
CALL     READ_FLOW
CALL     OPEN_IF_RELAY
CALL     TRANSMIT_DATA
CALL     RELAY_TEST
GOTO     NURAN

```

FL_15

```

MOVF     WORK, W
BCF      STATUS, Z
XORLW   h'15'           ;FL_15 READ
BTFSS    STATUS, Z
GOTO     NURAN
CALL     CLOSE_IF_RELAY
CALL     READ_FLOW
CALL     OPEN_IF_RELAY
CALL     TRANSMIT_DATA
CALL     RELAY_TEST
GOTO     NURAN

```

;-----

FL_40

```

MOVF     WORK, W
BCF      STATUS, Z
XORLW   h'40'           ;READ mV data. PG_00
BTFSS    STATUS, Z
GOTO     NURAN
CALL     CLOSE_IF_RELAY
CALL     READ_mV
CALL     OPEN_IF_RELAY
CALL     TRANSMIT_DATA
CALL     RELAY_TEST_mV

```

GOTO NURAN

;------

CLOSE_IF_RELAY

;Close I/F signal relay

```

BSF      SEL_BS
CALL     DELAY
CALL     DELAY
MOVLW   h'3E'
MOVWF   FSR
MOVF    INDF, W
MOVWF   DAT_DIG
CALL    TRANSMIT
INCF    FSR, F
MOVF    INDF, W
MOVWF   DAT_DIG
CALL    TRANSMIT
MOVLW   h'00'
MOVWF   DAT_DIG
CALL    TRANSMIT
MOVLW   h'01'
MOVWF   DAT_DIG
CALL    TRANSMIT
CALL    DELAY
BCF     SEL_BS
CALL    DELAY
CALL    DELAY
RETURN

```

;------

OPEN_IF_RELAY

;Open I/F signal relay

```

BSF      SEL_BS
CALL     DELAY
CALL     DELAY
MOVLW   h'3E'
MOVWF   FSR
MOVF    INDF, W

```



```

MOVWF    DAT_DIG
CALL     TRANSMIT
INCF     FSR, F
MOVF     INDF, W
MOVWF    DAT_DIG
CALL     TRANSMIT
MOVLW    h'00'
MOVWF    DAT_DIG
CALL     TRANSMIT
MOVLW    h'00'
MOVWF    DAT_DIG
CALL     TRANSMIT
BCF      SEL_BS
CALL     DELAY
CALL     DELAY
RETURN

```

;-----

TRANSMIT_DATA

```

BSF      TX_RAD           ;Destination address is Radio

```

;Receiving from KIŞİSEL BILGISAYARLAR is completed now Transmitting via
radio starts

```

CALL     DELAY
CALL     DELAY
MOVLW    h'90'
MOVWF    FSR
MOVLW    h'04'
MOVWF    DIG_NO
BSF      SEL_RAD
CALL     DELAY
CALL     DELAY

```

SINAN

```

MOVF     INDF, W
MOVWF    DAT_DIG
CALL     TRANSMIT

```

```

    INCF    FSR, F
    DECFSZ  DIG_NO, F
    GOTO    SINAN
    BCF     SEL_RAD
    CALL    DELAY
    CALL    DELAY
    CALL    DELAY
CLS_RAD
    BCF     TX_RAD
    CALL    DELAY
    BTFSC   REL_ST
;    GOTO    NURAN           ;Transmitting via radio completed
    RETURN
    BSF     TX_RAD
    CALL    DELAY
    GOTO    CLS_RAD
    RETURN
;=====
=====
PORTD_IN
    BSF     STATUS, RP0
    MOVLW   b'00001111'     ;PORTD input
    MOVWF   TRISD
    BCF     STATUS, RP0
    RETURN
;-----
PORTD_OUT
    BSF     STATUS, RP0
    MOVLW   b'00000000'     ;PORTD output
    MOVWF   TRISD
    BCF     STATUS, RP0
    RETURN
;-----
RD_SR

```

```
BSF      RS0
BCF      CS_
BCF      RD_
NOP
BSF      RD_
BSF      CS_
BCF      RS0
RETURN
```

```
;-----
```

```
RD_RDR
```

```
BCF      RS0
BCF      CS_
BCF      RD_
NOP
BSF      RD_
BSF      CS_
BCF      RS0
RETURN
```

```
;-----
```

```
WR_CR
```

```
BSF      RS0
BCF      CS_
BCF      WR_
NOP
BSF      WR_
BSF      CS_
BCF      RS0
RETURN
```

```
;-----
```

```
WR_TDR
```

```
BCF      RS0
BCF      CS_
BCF      WR_
NOP
```

```

    BSF      WR_
    BSF      CS_
    BCF      RS0
    RETURN

;-----
DELAY
    MOVLW   h'06'
    MOVWF   TMR
    CLRF    TMR0
    BCF     INTCON, T0IF
LOOP_DELAY
    BTFSS   INTCON, T0IF
    GOTO    LOOP_DELAY
    BCF     INTCON, T0IF
    DECFSZ  TMR, F           ;(6 X 64 X 256 X 1us)= 98ms
    GOTO    LOOP_DELAY
    RETURN

;-----
DELAY_           ;(1 X 64 X 256 X 1us)= 16ms
    CLRF    TMR0
    BCF     INTCON, T0IF
LOOP_DELAY_
    BTFSS   INTCON, T0IF
    GOTO    LOOP_DELAY_
    RETURN

;=====
READ_FLOW
    BSF     SEL_P
;Duty Code for closing I/F relay is 01, for opening I/F relay it is 00
    CLRF    DATA_L
    CLRF    DATA_H
;    MOVLW   h'3D'           ;30 x 64 x 256 x 1 usec = 0.4915 sec.
    MOVLW   h'1E'
    MOVWF   TMRX

```

```

        CLRF      TMR0          ;Pulse basladi, Timer sifirlandi
        BCF      INTCON, T0IF  ;Timer flag silindi

WAIT_POS_PLS
        BTFSC   INTCON, T0IF
        GOTO    DECR_TMR_1
        BTFSS   PORTB, 7
        GOTO    WAIT_POS_PLS

WAIT_NEG_PLS
        BTFSC   INTCON, T0IF
        GOTO    DECR_TMR_0
        BTFSC   PORTB, 7
        GOTO    WAIT_NEG_PLS

        INCFSZ  DATA_L, F
        GOTO    WAIT_POS_PLS
        INCF    DATA_H, F
        GOTO    WAIT_POS_PLS

;-----
DECR_TMR_1
        BCF      INTCON, T0IF  ;Timer flag silindi
        DECFSZ  TMRX, F
        GOTO    WAIT_POS_PLS
        GOTO    MERVE

DECR_TMR_0
        BCF      INTCON, T0IF  ;Timer flag silindi
        DECFSZ  TMRX, F
        GOTO    WAIT_NEG_PLS
        GOTO    MERVE

MERVE
        MOVLW   h'90'
        MOVWF   FSR
        CLRF    INDF

```

BTFSC	DATA_H, 7
BSF	INDF, 3
BTFSC	DATA_H, 6
BSF	INDF, 2
BTFSC	DATA_H, 5
BSF	INDF, 1
BTFSC	DATA_H, 4
BSF	INDF, 0

INCF	FSR, F
CLRF	INDF

BTFSC	DATA_H, 3
BSF	INDF, 3
BTFSC	DATA_H, 2
BSF	INDF, 2
BTFSC	DATA_H, 1
BSF	INDF, 1
BTFSC	DATA_H, 0
BSF	INDF, 0

;-----

INCF	FSR, F
CLRF	INDF

BTFSC	DATA_L, 7
BSF	INDF, 3
BTFSC	DATA_L, 6
BSF	INDF, 2
BTFSC	DATA_L, 5
BSF	INDF, 1
BTFSC	DATA_L, 4
BSF	INDF, 0

INCF	FSR, F
------	--------

CLRF INDF

```

BTFSC    DATA_L, 3
BSF      INDF, 3
BTFSC    DATA_L, 2
BSF      INDF, 2
BTFSC    DATA_L, 1
BSF      INDF, 1
BTFSC    DATA_L, 0
BSF      INDF, 0

```

```

BCF      SEL_P
RETURN

```

=====

READ_mV

```

CLRF     DATA_L
CLRF     DATA_H
BSF      ADCON0, ADON    ;A/D converter module is powered up
BSF      SEL_MA
CALL     DELAY
BSF      ADCON0, GO      ;A/D conversion is started

BTFSC    ADCON0, GO
GOTO     $-1
BCF      SEL_MA

MOVLW   h'90'
MOVWF   FSR
CLRF    INDF
BTFSC   DATA_H, 7
BSF     INDF, 3
BTFSC   DATA_H, 6
BSF     INDF, 2

```

BTFSC	DATA_H, 5
BSF	INDF, 1
BTFSC	DATA_H, 4
BSF	INDF, 0

INCF	FSR, F
CLRF	INDF

BTFSC	DATA_H, 3
BSF	INDF, 3
BTFSC	DATA_H, 2
BSF	INDF, 2
BTFSC	DATA_H, 1
BSF	INDF, 1
BTFSC	DATA_H, 0
BSF	INDF, 0

;-----

INCF	FSR, F
CLRF	INDF

BTFSC	DATA_L, 7
BSF	INDF, 3
BTFSC	DATA_L, 6
BSF	INDF, 2
BTFSC	DATA_L, 5
BSF	INDF, 1
BTFSC	DATA_L, 4
BSF	INDF, 0

INCF	FSR, F
CLRF	INDF

BTFSC	DATA_L, 3
BSF	INDF, 3


```

BTFSC    DATA_L, 2
BSF      INDF, 2
BTFSC    DATA_L, 1
BSF      INDF, 1
BTFSC    DATA_L, 0
BSF      INDF, 0

```

```

RETURN

```

```

;*****
,

```

```

DOWNLOAD

```

```

CLRF     INDF
CALL     PORTD_IN
CALL     RD_SR           ;Read Status Register of DTMF
        BTFSS    PORTD, 0           ;dtmf received a valid data
RETFIE
BTFSS    PORTD, 2
RETFIE
CALL     RD_RDR

BTFSC    PORTD, 0           ;D0
BSF      INDF, 0
BTFSC    PORTD, 1           ;D1
BSF      INDF, 1
BTFSC    PORTD, 2           ;D2
BSF      INDF, 2
BTFSC    PORTD, 3           ;D3
BSF      INDF, 3           ;data on INDF

BCF      STATUS, Z
MOVLW   h'0A'
XORWF   INDF, W
BTFSC   STATUS, Z
CLRF    INDF

```

RETURN

TRANSMIT

```

BSF      TX_LED
CALL     ZERO_TO_DTMF
CALL     PORTD_OUT
MOVLW   h'F0'
ANDWF   PORTD, F
BTFSC   DAT_DIG, 0
BSF     PORTD, 0
BTFSC   DAT_DIG, 1
BSF     PORTD, 1
BTFSC   DAT_DIG, 2
BSF     PORTD, 2
BTFSC   DAT_DIG, 3
BSF     PORTD, 3
CALL     TX_TO_DTMF
BCF     TX_LED
RETURN

```

ZERO_TO_DTMF

```

MOVF    DAT_DIG, W
BCF     STATUS, Z
XORLW   h'00'
BTFSS   STATUS, Z
RETURN
MOVLW   h'0A'
MOVWF   DAT_DIG
RETURN

```

TX_TO_DTMF

;transmitting to DTMF

procedure

```

CALL     WR_TDR           ;Data gonderildi
CALL     DELAY_

```

```

CALL    DELAY_
CALL    DELAY_
CALL    DELAY_
CALL    DELAY_
CALL    PORTD_IN
;Timer1 zamani saymaya basladi
CLRF    TMR1L
CLRF    TMR1H
MOVLW   b'11000000'           ;Global and Peripheral Interrupts are
enabled
MOVWF   INTCON
BSF     T1CON, 0             ;Timer1 enabled
CALL    RD_SR
LOOP
BTFSS   PORTD, 1           ;TDR ready for another data
transmitting
GOTO    LOOP
RETURN
;-----
RELAY_TEST
TEST_DATA_H
CALL    READ_FLOW           ;Baranin bos olup olmadigi kontrolu icin
                                ;tekrar okunuyor...
BCF     STATUS, Z
MOVF    DATA_H, W
XORLW   h'00'
BTFSC   STATUS, Z
GOTO    TEST_DATA_L
CALL    DELAY
CALL    OPEN_IF_RELAY
GOTO    TEST_DATA_H
TEST_DATA_L
BCF     STATUS, Z
MOVF    DATA_L, W

```

```

XORLW    h'00'
BTFSC    STATUS, Z
RETURN
CALL     CLOSE_IF_RELAY
CALL     DELAY
CALL     OPEN_IF_RELAY
GOTO     TEST_DATA_H
;-----
RELAY_TEST_mV
TEST_DATA_H_mV
    CALL     READ_mV           ;Baranin bos olup olmadigi kontrolu
icin tekrar
                                ; okunuyor...

    BCF     STATUS, Z
    MOVF    DATA_H, W
    XORLW   h'00'
    BTFSC   STATUS, Z
    GOTO    TEST_DATA_L_mV
    CALL    DELAY
    CALL    OPEN_IF_RELAY
    GOTO    TEST_DATA_H_mV
TEST_DATA_L_mV
    BCF     STATUS, Z
    MOVF    DATA_L, W
    XORLW   h'00'
    BTFSC   STATUS, Z
    RETURN
    CALL    CLOSE_IF_RELAY
    CALL    DELAY
    CALL    OPEN_IF_RELAY
    GOTO    TEST_DATA_H_mV
;-----
END

```

EK - 4**INTERFACE ÜNİTESİ MİKROKONTROLÖR PROGRAMI**

```

;*****
;INTERFACE.ASM is the name of RTU-INTERFACE communication
;Coming 4-bit data from a RTU is to be transmitted to a interface following its
algorithm,
;so that algorithm is a assembly program downloaded into a uC of PIC16F84.
;RTU-INTERFACE comm. is in an handshake comm. protocol via 2 wires.
;*****

    Title "RTU Program"
    list P=16f84A      ;processor type
    include <P16f84A.inc>
    __CONFIG    (_WDT_OFF & _HS_OSC & _PWRTE_OFF & _CP_OFF)
    errorlevel -302

;-----
    CBLOCK    h'10'
    TMR_1
    ADDR
    WORK
    JAPON
    DUMMY
    NUM
    ENDC

;-----
    #DEFINE    D0          PORTA, 3
    #DEFINE    D1          PORTA, 2
    #DEFINE    D2          PORTA, 1
    #DEFINE    D3          PORTA, 0
    #DEFINE    TX          PORTA, 4

    #DEFINE    IRQ        PORTB, 0
    #DEFINE    RD_        PORTB, 1
    #DEFINE    RS0        PORTB, 2

```

```

#DEFINE CS PORTB, 3
#DEFINE WR PORTB, 4
#DEFINE PWR_LED PORTB, 5
#DEFINE RX_LED PORTB, 6
#DEFINE TX_LED PORTB, 7

;-----
ORG h'00'
GOTO PORT_ARRANGE
ORG h'03'
GOTO PORT_ARRANGE
ORG h'04'
GOTO INT_DTMF

;-----
MOVLW b'00000000' ;All unmasked interrupts disable, flags r clear
MOVWF INTCON

;-----
PORT_ARRANGE ;Port arrangements
BSF STATUS, RP0 ; BANK1 selected

;-----
MOVLW b'00000' ;PORTA in
MOVWF TRISA

;-----
MOVLW b'00000001' ;PORTB
MOVWF TRISB

;-----
MOVLW b'00000101' ;PS=64
MOVWF OPTION_REG

;-----
BCF STATUS, RP0 ;BANK0 selected

;-----
;Initial Conditions
MOVLW h'00' ;Well Address
MOVWF ADDR

;-----

```

```

;initial conditions for PORTA
    MOVLW    b'00000'    ;| TX | D3 | D2 | D1 || D0 |
    MOVWF    PORTA

;-----
;initial conditions for PORTB
    MOVLW    b'00001000'
    MOVWF    PORTB

;-----
    CALL     PAUSE
    CALL     PAUSE

;-----
;Initialization of U2 (DTMF) ;Initialization of dtmf
;1) Read Status Register
    CALL     PORTA_IN
    CALL     RD_SR

;2) Write to Control Register
    CALL     PORTA_OUT
    CLRF     PORTA
    CALL     WR_CR

;3) Write to Control Register
    CALL     WR_CR

;4) Write to Control Register
    BSF     PORTA, 0
    CALL     WR_CR

;5) Write to Control Register
    BSF     PORTA, 3
    CALL     WR_CR

;6) Read Status Register
    CALL     PORTA_IN
    CALL     RD_SR

;-----
;Transmit DTMF tones of 50 ms burst/50 ms pause and Receive DTMF Tones
;7) Write to Control Register A
    CALL     PORTA_OUT

```

```

        MOVLW    b'01011'
        MOVWF   PORTA
        CALL    WR_CR
;8) Write to Control Register B
        CLRF   PORTA
        CALL   WR_CR
;*****Waiting for RX Interrupt*****
NURAN
        BCF    RX_LED
        MOVLW  h'3C'
        MOVWF  FSR
        CLRF   TMR0

MARTI
        MOVLW  b'10010000' ;Global and EE Write Complete interrupt,
RB0/INT
                                   ; enable, INTF=0, RBIF=0
        MOVWF  INTCON ;Timer interrupt disable
        MOVLW  h'07'
        MOVWF  JAPON

MURAT
        CLRF   TMR0
        BCF    INTCON, T0IF

WAIT_01
        BTFSS  INTCON, T0IF
        GOTO   WAIT_01
        DECFSZ JAPON, F
        GOTO   MURAT
        GOTO   NURAN

;-----
INT_DTMF ;Data receiving from DTMF, which is 4 digit
        BCF    INTCON, INTF
        BSF    RX_LED
        CALL   DOWNLOAD

BEKLE

```



```

    BTFSS    IRQ
    GOTO     BEKLE
    INCF     FSR, F
    BTFSS    FSR, 6
    GOTO     MARTI

```

```

;-----

```

```

;    GOTO     NURAN
PROCEEDING

```

```

;Adres sorgulaniyor

```

```

    MOVLW    h'3C'
    MOVWF    FSR
    SWAPF    INDF, W
    INCF     FSR, F
    IORWF    INDF, W
    BCF      STATUS, Z
    XORWF    ADDR, W
    BTFSS    STATUS, Z
    GOTO     NURAN

```

```

;Gorev sorgulaniyor

```

```

    MOVLW    h'3E'
    MOVWF    FSR
    SWAPF    INDF, W
    INCF     FSR, F
    IORWF    INDF, W
    MOVWF    WORK

    BCF      STATUS, Z
    MOVLW    h'01'
    XORWF    WORK, W
    BTFSC    STATUS, Z
    GOTO     CLOSE_RELAY

    MOVLW    h'00'
    XORWF    WORK, W

```

```

BTFSC    STATUS, Z
GOTO     OPEN_RELAY
GOTO     NURAN

```

```

OPEN_RELAY

```

```

BCF      TX          ;Stop sending 4-20mA
BCF      TX_LED
GOTO     NURAN

```

```

CLOSE_RELAY

```

```

BSF      TX          ;Start sending 4-20mA
BSF      TX_LED
GOTO     NURAN

```

```

;-----

```

```

PORTA_IN

```

```

BSF      STATUS, RP0
MOVLW   b'01111'   ; PORTA input
MOVWF   TRISA
BCF      STATUS, RP0
RETURN

```

```

;-----

```

```

PORTA_OUT

```

```

BSF      STATUS, RP0
MOVLW   b'00000'   ; PORTA output
MOVWF   TRISA
BCF      STATUS, RP0
RETURN

```

```

;-----

```

```

RD_SR

```

```

BSF      RS0
BCF      CS
BCF      RD_
NOP
BSF      RD_
BSF      CS

```

```
        BCF      RS0
        RETURN
;-----
RD_RDR
        BCF      RS0
        BCF      CS
        BCF      RD_
        NOP
        BSF      RD_
        BSF      CS
        BCF      RS0
        RETURN
;-----
WR_CR
        BSF      RS0
        BCF      CS
        BCF      WR
        NOP
        BSF      WR
        BSF      CS
        BCF      RS0
        RETURN
;-----
WR_TDR
        BCF      RS0
        BCF      CS
        BCF      WR
        NOP
        BSF      WR
        BSF      CS
        BCF      RS0
        RETURN
;-----
DOWNLOAD
```

```

CLRf    INDF
CALL    PORTA_IN
CALL    RD_SR
BTfSS   PORTA, 3
GOTO    NURAN
BTfSS   PORTA, 1
GOTO    NURAN
CALL    RD_RDR

```

```

BTfSC   PORTA, 3
BSF     INDF, 0           ;D0
BTfSC   PORTA, 2
BSF     INDF, 1           ;D1
BTfSC   PORTA, 1
BSF     INDF, 2           ;D2
BTfSC   PORTA, 0
BSF     INDF, 3           ;D3

```

```

BCF     STATUS, Z
MOVF    INDF, W
XORLW   h'0A'
BTfSC   STATUS, Z
GOTO    CHANGE_0
RETURN

```

```

;-----

```

```

CHANGE_0

```

```

CLRf    INDF
RETURN

```

```

;-----

```

```

PAUSE

```

```

MOVLW   h'06'
MOVWF   TMR_1
CLRf    TMR0
BCF     INTCON, T0IF

```

```
LOOP_T2
    BTFSS    INTCON, T0IF
    GOTO    LOOP_T2
    BCF     INTCON, T0IF
    DECFSZ  TMR_1, F    ;(6 X 64 X 256 X 1us)= 98ms
    GOTO    LOOP_T2
    RETURN

;-----
    END
```

7. KAYNAKLAR

- [1]. AXELSON J., 1999, Paralel Port, Bilişim İnfogate, İSTANBUL.
- [2]. CARLSON A.B., 1986, Communication Systems, McGraw-Hill International Editions, SINGAPORE.
- [3]. HOWARD W. Sams & Co. Inc., 1972, Reference Data For Radio Engineers, ITT, USA.
- [4]. KORKMAZ, A; 1981, Türkiye' de Çağdaş Haberleşmenin Tarihsel Kökenleri, AİTİA Gazetecilik ve Halkla İlişkiler Yüksek Okulu Basımevi, Ankara
- [5]. MICROCHIP, 2003, PIC16F87XA Datasheet, USA.
- [6]. ÖZYETİŞ K, 1984, Muhabere-Elektronik Ders Notları, Polis Enstitüsü, Ankara
- [7]. TOMASI W., 2002, Elektronik İletişim Teknikleri, M.E.B, İSTANBUL.
- [8]. UZUN Ö., GÖKKAYA K., 1994, Mikroişlemciler ve Assembler Programlama, Alfa, İSTANBUL.
- [9]. ZARLINK Semiconductor Inc., 2004, MT8889C Transiver Data Sheet, USA
- [10]. www.ptt.gov.tr, [www.ptt.gov.tr, www.ptt.gov.tr/tr/yayinlar/dergi/12.html](http://www.ptt.gov.tr/tr/yayinlar/dergi/12.html), (03/05/2006)
- [11]. www.wikipedia.org, http://en.wikipedia.org/wiki/Radio#Uses_of_radio
- [12]. www.motorola.com,
www.motorola.com/governmentandenterprise/contentdir/en_GB/Files/ProductInformation/gm340_brochure.pdf, (23/04/2006)

8. ÇİZELGE LİSTESİ

Tablo.1. Paralel Port Yazmaç Adresleri.....	21
Tablo.2 Veri Yazmacı Pin Adresleri.....	22
Tablo.3. Durum Yazmacı Pin Adresleri.....	23
Tablo.4. Kontrol Yazmacı Pin Adresleri.....	23
Tablo.5. Projede Mevcut Telsiz Frekansları.....	38
Tablo.6. DTMF Kod Frekansları.....	45
Tablo.7. Bir Üretim İstasyonuna Uygulanmış Sistemin Maliyet Tablosu.....	68

9. ŞEKİL LİSTESİ

Şekil.1. RF Dalgalarının Yayılma Yolları.....	6
Şekil.2. Genlik Modülasyonu.....	7
Şekil.3. HF Radyo Yayınları Yansıması.....	8
Şekil.4. Frekans Modülasyonu.....	10
Şekil.5. Mikroişlemci Blok Şeması.....	13
Şekil.6. Yazmaç.....	15
Şekil.7. Ana ve Uzaktaki Terminal Birimleri Blok Şeması.....	26
Şekil.8. Terminal Birimi Blok Şeması.....	28
Şekil.9. Terminal Ana Kartı Blok Şeması	29
Şekil.10. Terminal Ana Kartı Devre Şeması	31
Şekil.11. Ölçme Kartı Giriş/Çıkış Blok Şeması.....	34
Şekil.12. Ölçme Kartı Devre Şeması.....	36
Şekil.13. Seçici Arayüz Kartı Blok Şeması.....	39
Şekil.14. Seçici Arayüz Kartı Devre Şeması.....	40
Şekil.15. Bilgisayar-Mikrodenetleyici Paralel Port Haberleşme Blok Şeması.....	43
Şekil.16. MT8889C DTMF Transiveri Blok Şeması.....	45
Şekil.17. Mikrodenetleyici-DTMF Transiveri Blok Şeması.....	45
Şekil.18. Ana Terminal Biriminden Bilgi Gönderme Akış Şeması.....	48
Şekil.19. Uzaktaki Terminal Birimi Tarafından Bilgi Alınması Akış Şeması.....	50
Şekil.20. Terminal Birimi-Seçici Arayüz Bağlantı Blok Şeması.....	51
Şekil.21. Ana Terminal Birimi Tarafından Bilgi Alınması Akış Şeması.....	55
Şekil.22. Tank Üzerinde Çeşitli Tip Ultrasonik Seviye Sensörü Yerleşimleri.....	63
Şekil.23. Ham Petrol Tankı, Ultrasonik Seviye Sensörü Ve 4-20mA Çıkışı.....	63
Şekil.24. Türbin Tip Akışmetre.....	64
Şekil.25. Ham Petrol Akışı İçin Tipik Bir Kalibrasyon Eğrisi.....	65

10. RESİM LİSTESİ

Resim.1. Terminal Birimi.....	1
Resim.2. Merkezi Bilgisayar Ekranından Kuyu Akış Değerleri.....	28
Resim.3. Terminal Ana Kartı.....	31
Resim.4. Ölçme Kartı.....	35
Resim.5. Motorola GM340 Araç Telsiz Cihazı.....	38
Resim.6. Terminal Birimi Akü ve Dolabı.....	39
Resim.7. Seçici Arayüz Kartı.....	40
Resim.8. Ham Petrol Ayrışım Tankı.....	63
Resim.9. Bir Akışmetre Göstergesi.....	67

11. YAZARIN ÖZGEÇMİŐİ

Őükrü Nail Uncu, 1962 Ankara doğumlu, Elektrik-Elektronik Mühendisi olarak O.D.T.Ü.'den mezun olduktan sonra, çeŐitli firmalarda tasarım mühendisi olarak çalıştı, halen N.V.Turkse PERENCO için proje geliştirme mühendisi olarak çalışmaktadır.