

SAYISAL HAT KODLAMA TEKNİKLERİ

Sabri Can BOZDEMİR

**YÜKSEK LİSANS TEZİ
ELEKTRONİK BİLGİSAYAR EĞİTİMİ**

**GAZİ ÜNİVERSİTESİ
BİLİŞİM ENSTİTÜSÜ**

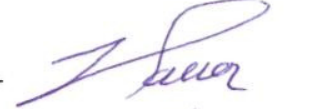
**TEMMUZ 2007
ANKARA**

Sabri Can BOZDEMİR tarafından hazırlanan SAYISAL HAT KODLAMA
TEKNİKLERİ adlı bu tezin Yüksek Lisans tezi olarak uygun olduğunu
onaylarım.



Prof. Dr. Nihal Fatma GÜLER
Tez Yöneticisi

Bu çalışma, jürimiz tarafından oy birliği / ~~oy çokluğu~~ ile Elektronik Bilgisayar
Eğitimi Anabilim Dalında Yüksek lisans tezi olarak kabul edilmiştir.

Başkan : İrd. Doç. Dr. Tilmaz SAVAŞ 

Üye : Prof. Dr. Nihal Fatma Güler 

Üye : İrd. Doç. Dr. H. Hüseyin SATAR 

Üye : _____

Üye : _____

Tarih : 11.7.2007

Bu tez, Gazi Üniversitesi Bilişim Enstitüsü tez yazım kurallarına uygundur.

TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca tez yazım kurallarına uygun olarak hazırlanan bu çalışmada orijinal olmayan her türlü kaynağa eksiksiz atıf yapıldığını bildiririm.



Sabri Can BOZDEMİR

SAYISAL HAT KODLAMA TEKNİKLERİ

(Yüksek Lisans Tezi)

Sabri Can BOZDEMİR

GAZİ ÜNİVERSİTESİ

BİLİŞİM ENSTİTÜSÜ

Temmuz 2007

ÖZET

Sayısal haberleşme sistemleri, analog haberleşme sistemlerine göre avantajların çok olması nedeniyle hızla yaygınlaşmaktadır. Bu sebepten sayısal haberleşme üzerindeki çalışmalar çok önem kazanmaktadır. Bu çalışmada sayısal haberleşme konularından olan sayısal hat kodlama teknikleri üzerinde durulmuştur. Çok kullanılan hat kodu türlerinin gözlenebilmesi ve işaretlerin karşılaştırılabilmesi için bilgisayarlı simülasyonlar yapılmıştır. Ayrıca uygun bir deney setiyle gerçek ortamda da sonuçların gözlenebilmesi sağlanmıştır.

Bilim Kodu : 704.3.032

Anahtar Kelimeler : Haberleşme, Sayısal, Kodlama, Deney Seti

Sayfa Adedi : 103

Tez Yöneticisi : Prof. Dr. Nihal Fatma GÜLER

DIGITAL LINE CODING TECHNICALS

(M.Sc. Thesis)

Sabri Can BOZDEMİR

**GAZİ UNIVERSITY
INFORMATICS INSTITUTE**

July 2007

ABSTRACT

Digital communication systems have been spreading, according to analog communication systems because of its advantages. The studies on digital communication subjects have been getting importance. In this study, it's worked on digital line code techniques which is one of the digital communication subjects. Computer simulations were done, which is used mostly, in order to be observed and compared the signals of line code kinds. Besides an experiment set was planned and it was achieved to be able to observe the results in real atmosphere.

Science Code	: 704.3.032
Key Words	: Communication, Digital, Coding, Experiment Set
Page Number	: 103
Adviser	: Prof. Dr. Nihal Fatma GÜLER

TEŐEKKÖR

Çalıřmalarım boyunca deęerli yardım ve katkılarıyla beni yönlendiren hocam Prof. Dr. Nihal Fatma GÖLER'e yine kıymetli tecrübelerinden faydalandığım hocam Arş. Gör. Uęur FİDAN'a, ayrıca Zile Endüstri Melsek Lisesi'nde görevli öğretmen arkadaşlarıma, çok deęerli arkadaşlarım Murat ARICAN, Engin Utku ŐENSES ve Orkun SUNAR'a, manevi destekleriyle beni hiçbir zaman yalnız bırakmayan aileme teşekkürü bir borç bilirim.

İÇİNDEKİLER

Sayfa

ÖZET	i
ABSTRACT	ii
TEŞEKKÜR.....	iii
İÇİNDEKİLER	iv
ÇİZELGELERİN LİSTESİ.....	vi
ŞEKİLLERİN LİSTESİ	vii
SİMGELER VE KISALTMALAR	ix
1. GİRİŞ.....	1
2. SAYISAL HABERLEŞME SİSTEMLERİ	2
2.1. Sayısal Haberleşmenin Avantajları	5
2.2. Sayısal Haberleşmenin Dezavantajları	6
2.3. Sayısal Haberleşmenin Yaygınlaşmasının Nedenleri	7
2.4. Seri Haberleşme	7
2.4.1. Eş zamanlı (senkron) seri veri iletimi	8
2.4.2. Eş zamanlı olmayan (asenkron) seri veri iletimi.....	9
3. SAYISAL KODLAMA	12
3.1. Kodlama.....	12
3.1.1. Bilgiye karşılık gelen işaret seviyeleri	13
3.1.2. Kodlama sınıflandırmaları	13
3.2. Hat Kodlaması	15
3.3. Hat Kodu Türleri.....	17
3.3.1. Tek kutuplu NRZ (Unipolar Non-Return to Zero) kodlama ..	17

3.3.2. İki kutuplu NRZ (Bipolar Non-Return to Zero) kodlama.....	17
3.3.3. Tek kutuplu RZ (Polar Return to Zero) kodlama.....	18
3.3.4. İki kutuplu RZ (Bipolar Return to Zero) kodlama	19
3.3.5. Manchester (Split Phase) kodlama.....	19
3.3.6. Diferansiyel Manchester.....	21
3.3.7. AMI (Alternate Mark Inversion) kodlama	22
3.3.8. HDB3 (High Density Bipolar 3) kodlama	23
3.3.9. CMI (Coded Mark Inversion) kodlama.....	26
3.3.10. MCMI (Modified Coded Mark Inversion) kodu	27
3.4. Kodların özellikleri.....	27
4. SİMÜLASYONLAR VE DENEYLER	29
4.1. Karşılaştırmalar.....	30
4.1.1. UPNRZ ve BPNRZ karşılaştırması.....	30
4.1.2. UPNRZ ve UPRZ karşılaştırması	31
4.1.3. UPNRZ ve BPRZ karşılaştırması	32
4.1.4. UPNRZ ve Manchester karşılaştırması	34
4.1.5. UPNRZ ve AMI karşılaştırması.....	35
4.1.6. UPNRZ ve CMI karşılaştırması	36
5. SONUÇ VE ÖNERİLER.....	38
KAYNAKLAR	40
EKLER	41
EK-1 Sayısal Hat Kodlama Deney Seti Kullanma Kılavuzu ve Deneyler .	42
ÖZGEÇMİŞ.....	103

ÇİZELGELERİN LİSTESİ

Çizelge

Sayfa

Çizelge 3.1. Hat kodlarının özelliklerinin karşılaştırılması 27

ŞEKİLLERİN LİSTESİ

Şekil	Sayfa
Şekil 2.1. Basit bir sayısal haberleşme sistemi	2
Şekil 2.2. Sayısal haberleşme sisteminin temel elemanları	4
Şekil 2.3. Senkron İletim	9
Şekil 2.4. Asenkron haberleşme	10
Şekil 3.1. Kodlama	12
Şekil 3.2. İşaret seviyeleri	13
Şekil 3.3. Unipolar ve bipolar işaret seviyeleri.....	14
Şekil 3.4. NRZ ve RZ kodlamaları.....	14
Şekil 3.5. Hat kodlaması	15
Şekil 3.6. Unipolar NRZ kodlaması	17
Şekil 3.7. Bipolar NRZ kodlaması	17
Şekil 3.8. Polar RZ kodlama	18
Şekil 3.9. İki kutuplu RZ kodlama.....	19
Şekil 3.10. Manchester kodlama.....	20
Şekil 3.11. Diferansiyel Manchester kodlama	21
Şekil 3.12. AMI kodlama	22
Şekil 3.13. HDB3 kodlama	24
Şekil 3.14. HDB3 kodlama örnekleri	25
Şekil 3.15. CMI kodu.....	26
Şekil 3.16. Bazı hat kodlama tekniklerinin karşılaştırılması.....	28
Şekil 4.1. UPNRZ ve BPNRZ karşılaştırması.....	30

Şekil 4.2. UPNRZ ve BPNRZ kodlamanın X-Y düzleminde gösterimi.....	31
Şekil 4.3. UPNRZ ve UPRZ karşılaştırması	31
Şekil 4.4. UPNRZ ve UPRZ kodlamanın X-Y düzleminde gösterimi	32
Şekil 4.5. UPNRZ ve BPRZ karşılaştırması	33
Şekil 4.6. UPNRZ ve BPRZ kodlamanın X-Y düzleminde gösterimi	33
Şekil 4.7. UPNRZ ve Manchester karşılaştırması	34
Şekil 4.8. UPNRZ ve Manchester kodlamanın X-Y düzleminde gösterimi .	35
Şekil 4.9. UPNRZ ve AMI karşılaştırması	35
Şekil 4.10. UPNRZ ve AMI kodlamanın X-Y düzleminde gösterimi.....	36
Şekil 4.11. UPNRZ ve CMI karşılaştırması	37
Şekil 4.12. UPNRZ ve CMI kodlamanın X-Y düzleminde gösterimi	37

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış bazı kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Kısaltmalar	Açıklama
AMI	Değişimli işaret tersleme
BPNRZ	İki kutuplu sıfıra dönmeyen
BPRZ	İki kutuplu sıfıra dönen
CMI	Kurallı işaret tersleme
DSP	Sayısal işaret işleme
HDB3	Yüksek yoğunluklu iki kutuplu
LCD	Sıvı kristal ekran
PCM	Pals Kod Modülasyonu
UPNRZ	Tek kutuplu sıfıra dönmeyen
UPRZ	Tek kutuplu sıfıra dönen
USB	Evrensel seri veri yolu

1. GİRİŞ

Günümüzde sayısal haberleşme sistemleri, analog haberleşme sistemlerine göre avantajların çok olması nedeniyle hızla yaygınlaşmaktadır. Bu sebepten sayısal sistemlerdeki gelişmelerin takibi sayısal haberleşme konularındaki çalışmalarla paralel ilerlemektedir.

Lise ve üniversite öğrencileri sayısal hat kodlama tekniği hakkındaki teorik bilgilerini çoğu zaman deneysel olarak görmediği için konunun önemini tam kavrayamamaktadır.

Bu çalışmada, sayısal haberleşmenin konularından olan sayısal hat kodlama teknikleri üzerinde durulmuştur.

Sayısal haberleşme sistemleri başlığı altındaki bölümde, ilk olarak, sayısal haberleşmenin ne olduğu açıklanmıştır. Ayrıca analog haberleşme sistemlerine göre avantajları, dezavantajları ve yaygınlaşmasının sebepleri de incelenmiştir.

Üçüncü bölümde, kodlamanın ne olduğu üzerinde durularak, hat kodlaması ve sayısal haberleşmede çok kullanılan hat kodu türleri açıklanmıştır. Buradan yola çıkarak hat kodlama teknikleri arasındaki farklar incelenmiştir.

Konuyla ilgili olarak bilgisayarlı simülasyonlar yapılmıştır. Böylece hat kodu türlerinin daha iyi incelenebilmesi sağlanmıştır. Çeşitli işaretler görülüp, değerlendirme ve karşılaştırma yapılmıştır.

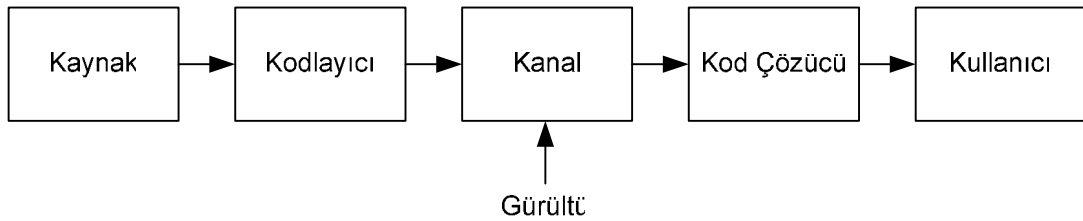
Ayrıca yine konuyla ilgili olarak gerçek ortamdaki sonuçları görmek üzere deney seti tasarlanmıştır.

2. SAYISAL HABERLEŞME SİSTEMLERİ

Haberleşme sistemlerindeki genel amaç bir mesajı bir noktadan diğer bir noktaya taşımaktır. Mesajın üretildiği yer genellikle kaynak (source) olarak bilinir. Sonlandığı yer ise kullanıcı ya da hedef olarak tanımlanır [1].

İletişimin amacı, herhangi bir biçimdeki bilginin zaman ve uzay içinde kaynak olarak adlandırılan bir noktadan, kullanıcı denilen başka bir noktaya aktarılmasıdır. Sayısal haberleşme ise bir haberleşme sisteminde sayısal darbelerin iki veya daha fazla nokta arasında iletişimidir [2].

Bir noktadan diğer bir noktaya sayısal bilgilerin iletilmesi işlemine veri iletimi denir. Veri iletim sistemleri, bilgisayar - bilgisayar ve bilgisayar - terminal arasında veri iletimini sağlar. Sayısal hale dönüştürülebilen ses, görüntü gibi analog bilgilerin iletilmesi de veri iletimi ile gerçekleştirilir. Sayısal tekniklerin verimliliği yüksek, maliyetleri oldukça düşüktür. Bu nedenle, sayısal veri iletim sistemleri oldukça kullanışlıdır [1].



Şekil 2.1. Basit bir sayısal haberleşme sistemi [3]

Bir sayısal haberleşme sistemi temel olarak Şekil 2.1'deki bloklardan meydana gelir. Burada kaynak, çıkışı konuşma veya müzik ile oluşturulan akustik bir dalga biçimi, bir manyetik banttıan ikili sayılar, bir biyolojik organizmanın duyu girişleri olabilir. Yani analog ya da sayısal enformasyon içerebilir. Kaynak çıkışı,

$$A = (a_1, a_2, \dots, a_N) \quad (2.1)$$

ile ifade edilebilir. Bir diğer durum sonlu sayıda elemandan oluşan alfabadir.

$$X = (X_1, X_2, \dots, X_N) \quad (2.2)$$

İkili kaynak 0 ve 1 ile tanımlanan iki elemanlı bir alfabaya sahiptir.

$$X = (0,1) \quad (2.3)$$

ve tipik bir çıkış

$$X = (1,1,0,0,1,0,1,1, \dots) \quad (2.4)$$

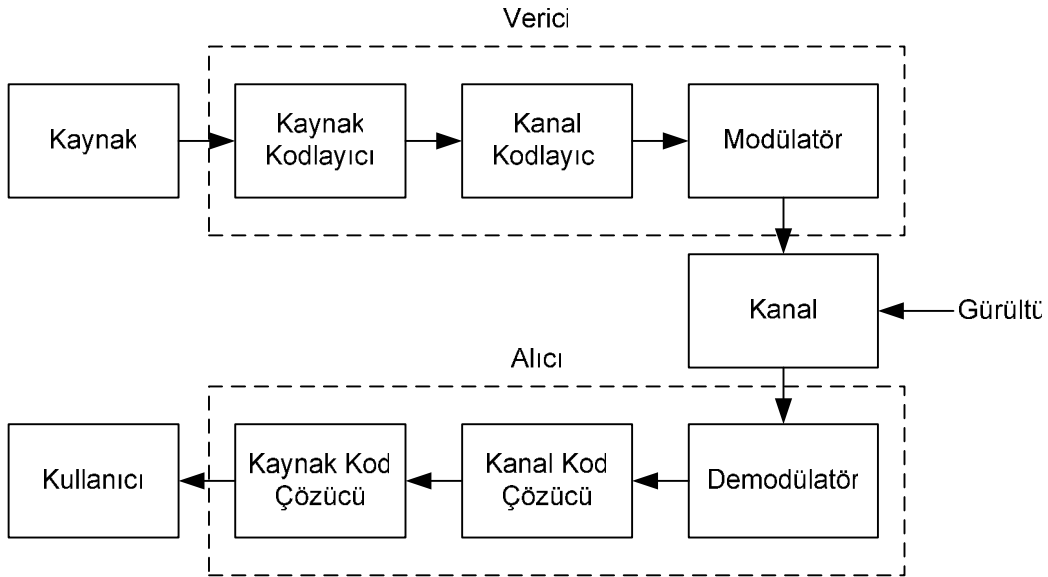
serisi ile ifade edilebilir [3].

Kodlayıcı, kaynak çıkışlarındaki büyüklüğün alıcıya iletilmeden önce gördüğü ön işlemleri belirlemektedir. Genel olarak elektriksel olmayan kaynak çıkışını elektriksel hale dönüştürür. Bilgi taşıyan elektriksel büyüklüğün iletim ortamından en iyi biçimde iletilmesi için, modülasyon işlemi, kanal gürültüsünün etkisini minimuma indirmek için bir tür kodlama işlemidir [3].

Kanal, kaynak ve alıcı arasındaki elektriksel bağlantıyı sağlamaktadır. Haberleşme sisteminin üzerinden gönderilen enformasyon taşıyan elektriksel işaretler genellikle gürültü adı verilen çeşitli elektriksel dalgalar tarafından bozulurlar. Bu bozucu dalgalar tamamen yok edilemezler.

Sayısal iletişim ikili tabanda kodlanmış bilgi ya da verinin sistemler arasında aktarılmasını kapsar [4]. Sayısal işaretler, her biri bir gerilim seviyesiyle tanımlanan ve birbirinden farklı iki durumdan oluşan darbelerdir. Bu darbeler iki seviye arasında değişir.

Analog haberleşme sisteminde, sürekli zaman formunda tanımlanan analog işaretler, modüle edilmiş taşıyıcı işaret ile haberleşme kanalında doğrudan gönderilebilir ve vericide alınarak benzer şekilde demodüle edilir. Alternatif olarak analog işaret, sayısal işaret haline dönüştürülerek, sayısal modüleli bilgi olarak gönderilebilir ve alıcıda sayısal işaret olarak demodüle edilebilir. Analog işaretin sayısal işaret haline getirilerek gönderilmesi sayısal işaret yayılımının, analog işaret yayılımından daha iyi kontrol edilebilmesi yönüyle, işaret kalitesi ve güvenilirliği açısından avantaj sağlar. Özellikle sayısal işaret yayılımı, gönderilen işaretin uzak mesafeden alınması ve gürültü etkilerinin önlenmesi açısından önemlidir. Sayısal sistemlerin tercih edilmesinin diğer bir sebebi, gönderilecek analog bilgi işaretinin çok fazla olması durumunda, sayısal işarete dönüştürülerek modüle edilmeden önce işaretin fazlalığının azaltılabilmesi ve sayısal sistemlerin kuruluş aşamasında analog sistemlere oranla daha ekonomik olmasıdır.



Şekil 2.2. Sayısal haberleşme sisteminin temel elemanları [3]

Sayısal haberleşme sistemindeki alıcı ve verici bölümlerinde temel olarak, bilgi işaretinin filtrelenerek sıkıştırılması, kanal kodlama, kod çözme ve

sıkıştırmanın yeniden açılması gibi işlemler yapılır. Şekil 2.2 bir sayısal haberleşme sisteminin temel elemanlarını göstermektedir.

Şekil 2.2'de kaynak ve kanal modelleri hakkındaki incelemeleri daha kolaylaştırmak için genellikle sistemde kodlayıcı ve kod çözücü blokları ikiye bölünerek gerçekleştirilebilir.

Kaynak kodlayıcısının görevi, kaynak çıkışındaki bilgiyi ikili sayılara dönüştürmektir. Burada karşılaşılan en önemli problem verilen enformasyon kaynağının ürettiği bilgiyi belirlemek için birim zamanda kaç tane ikili sayıya gerek olduğunun bilinmesidir [3].

Kaynak tarafından üretilen bilgi, genellikle ikili sayı dizisine çevrilir. İdeal olarak kaynak çıkışındaki bilgi işaretinin etkili bir şekilde, ikili sayı dizisine çevrilmesi ile artık bilgi işaretinin kalmaması veya azalması beklenir. Analog veya sayısal kaynak bilgisinin ikili sayı dizisine çevrilmesine, kaynak kodlama veya bilgi sıkıştırma denir. Sıkıştırılmış bilgi işaretinin, kanal kodlama bölümüne gönderilir. Kanal kodlamanın amacı kanaldan gönderilerek alıcı tarafından alınan sıkıştırılmış bilgi işaretinde, işaretin kanalda yayılımı sırasında oluşan ve alıcıda görülen gürültü ve diğer olumsuz etkilerin giderilmesi için kullanılabilecek şekilde artık bilginin kontrol edileceği bir durum elde etmektir. Böylece alınan işaretin güvenilirliği ve kalitesi yükseltilmiş olunur. Örneğin, sıkıştırılmış ikili bilgi dizisinin kodlanmasında basit bir form, her ikili sayının m kere (m pozitif tam sayı) tekrar edilmesidir.

2.1. Sayısal Haberleşmenin Avantajları

1. Sayısal iletimin en önemli avantajı, gürültüden fazla etkilenmemesidir. Analog işaretler, sayısal darbelere oranla arzu edilmeyen genlik, frekans ve faz değişimlerine daha yatkındır. Genlik, frekans ve faz etkileri sayısal iletişimde pek etkili olmaz [2].

2. Sayısal darbeler, işleme ve çoğullama için analog işaretlerden daha uygundur. Sayısal darbeler kolayca saklanabilir, ancak analog işaretleri saklamak kolay değildir.
3. Sayısal bir sistemin iletim hızı, değişik ortamlara uyum gösterecek ya da değişik tür donanımlara, arabirim üzerinden bağlanacak şekilde kolayca değiştirilebilir.
4. Sayısal sistemlerde ölçüm ve değerlendirme daha kolaydır. Bu nedenle değişik işaret ve bilgi kapasiteleri için sayısal sistemlerin performanslarını kıyaslamak analog sistemlere göre daha kolaydır [2].
5. Sayısal sistemlerde hata bulma ve düzeltme analog sistemlere göre daha kolaydır [2].

2.2. Sayısal Haberleşmenin Dezavantajları

1. Sayısal olarak kodlanmış analog işaretlerin iletimi, analog işaretleri oldukları gibi iletmeye oranla daha fazla bant genişliği gerektirir [2].
2. Gönderilecek olan bilgi analog bir işaret ise, iletimden önce sayısal kodlara, alıcıda ise tekrar analog biçime dönüştürülmelidir [2].
3. Sayısal haberleşmede, verici ile alıcının saat darbeleri arasında duyarlıklı senkronizasyon gerektirir [2].
4. Sayısal haberleşme sistemleri, günümüzde kullanılmakta olan çoğu analog sistem donanımı ile uyumlu değildir [2].

2.3. Sayısal Haberleşmenin Yaygınlaşmasının Nedenleri

1. Tümüleşik devre teknolojisindeki son gelişmeler, sayısal devre tasarımını kolaylaştırmıştır.
2. DSP tekniklerindeki ilerlemeler, sayısal işaretlerin daha verimli bir şekilde kullanımına neden olmuştur.
3. Sayısal bilgisayarların çok yaygın kullanımı.
4. Gürültü ve girişim açısından sayısal haberleşme sistemlerinin, analog haberleşme sistemlerine göre daha güvenilir olması.
5. Zaman çoklamalı sistemlere elverişli olması.

2.4. Seri Haberleşme

Seri veri iletiminde, bir veri biti, bir diğerinin hemen arkasından iletilir. Bit düzeyinde, bilgi işareti bir kanal boyunca, aynı anda bir bit olarak nakledilir. Her bit, bilgi işaretinin bir parçasını temsil eder. Alıcıda bu bitler yeniden bir araya getirilip, bilgi işareti elde edilir. Yalnızca bir kanalın var olduğu durumlar için, çok uygun bir haberleşme şeklidir.

Bir bayt'lık bir veriyi seri olarak iletmek için, ilgili hattın gelen paralel verinin, seri bit akışına dönüştürülmesi gerekir [5]. Bilgisayarlar bu yolu, çoğu çevresel aygıtıyla bir haberleşme yöntemi olarak kullanır. Örneğin; fare (mouse), klavye, modem, seri yazıcı vb..

Seri bir linkteki verici, veya sürücü, belli bir anda bir bit olmak üzere, bitleri sırayla yollar. İşaret aktarım hızı 'baud' birimiyle ölçülür. Birim zaman içinde iletilen verinin sayısı ile belirlenir. Kabaca 1 saniyede akan 1 bit veriye eşdeğerdir [5].

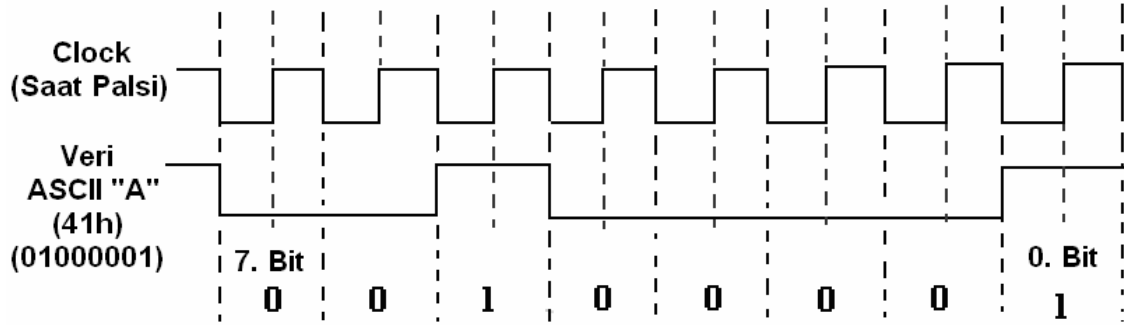
Linkte, her bir yön için ayrı bir hat olabileceği gibi, ortak kullanılan tek bir hat da bulunabilir. Yani, verici dönüşleri aynı hattan alır. Üç ya da daha fazla cihaz olması halinde, çoğunlukla hepsi aynı yolu kullanırlar. Hangi iletinin yapılacağını ağ protokolü belirler.

Bir linkteki veri akışının kontrolü için gerekli işaretlerden biri saat işareti veya zamanlama referansıdır. Hem gönderen hem de alan cihaz bir bitin ne zaman gönderileceğine ya da alınacağına karar verirken bir saat işareti kullanırlar.

Seri veri iletimi, eşzamanlı (senkron) ya da eş zamanlı olmayan (asenkron) olabilir. Eş zamanlı olmayan seri veri iletim yönteminin çok avantajları vardır. Dolayısıyla en çok kullanılan yöntemdir.

2.4.1. Eş zamanlı (senkron) seri veri iletimi

Eş zamanlı gönderimde, her cihaz, içlerinden biri ya da dışardan bir cihaz tarafından türetilen aynı saati (saat işaretini, saat darbesini) kullanırlar. Saatin frekansı sabit ya da düzensiz aralıklarda değişken de olabilir. İletilen her bit saatle eşzamanlıdır. Alıcı, saat geçişlerini, gelen biti okuyacağı zamanı belirlemekte kullanır. Protokoller ince ayrıntıları itibariyle farklı olabilmektedir. Örneğin, bir alıcının gelen veriyi saatin alçalan ya da yükselen kenarında, yahut yüksek ya da düşük bir mantık düzeyi algılayınca işaretlemesi mümkündür. Senkron formatlar, iletimi başlatırken ya da bitirirken çok çeşitli işaretler kullanırlar.



Şekil 2.3. Senkron İletim

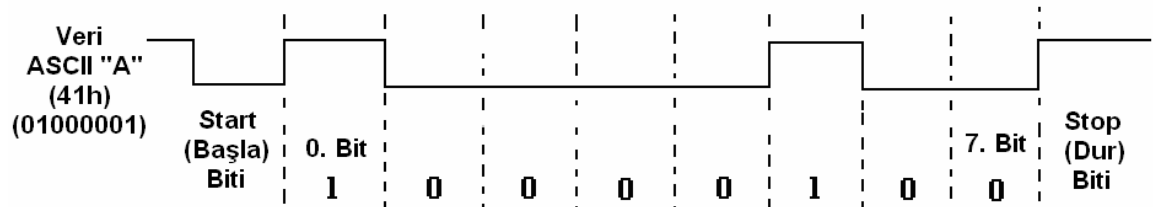
Senkron arabirimler, bir devre kartı üzerindeki devre elemanları arasındaki gibi, kısa mesafeli linklerde kullanılması uygundur. Bilgisayarların anakartlarındaki hatlar buna en güzel örnektir. Uzun mesafeli linklerde ise senkron formatlar pratik değildir. Çünkü böyle durumlarda saat işaretinin iletimi, parazit nedeniyle, ilave bir hat gerektirmektedir.

Bunların dışında isokron isimli bir iletim yöntemi de mevcuttur. İsochron iletim, senkron iletimin bir türevi (çeşidi) olarak düşünülebilir. Bu iletimde uç sistemlerin birbirleriyle olan haberleşme gereksinimi periyodik olarak karşılanır. Her birim zamanda ne kadar bit aktarılacağı belirtilir. Bu tür bir iletim özellikle gerçek zamanlı uygulamalar (video, ses aktarımı vb.) için gereklidir [4].

2.4.2. Eş zamanlı olmayan (asenkron) seri veri iletimi

Asenkron iletimlerde linkte saat hattı olmaz. Çünkü her uç kendi işaretini sunmaktadır. Ancak, uçların saat frekanslarında anlaşmaları ve saatlerin de uyumlu olmaları gerekir. İletilen her bayt'ta saatleri eşlemek üzere bir Start (Başla) biti ve iletimin bittiğini bildirmek üzere bir veya daha fazla Stop (Dur) biti bulunur. Bir iletişim hattında yeni bir baytlık bilgi iletileceği zaman ilk önce başla biti gönderilir. Böylece alıcı 1'den 0'a geçişte yeni bir baytlık bilginin

geleceğini anlar. Baytın bitiminde dur biti yollanır. Alıcı bu şekilde yeni bir başla bitinin gelişini bekler (Şekil 2.4).



Şekil 2.4. Asenkron haberleşme

Kişisel bilgisayarlarda RS-232 portlar, modemlerle ve diğer cihazlarla iletişimde asenkron formatları kullanır [6]. Bunlar aslında senkron veri transferi de yapabilirler. Ancak asenkron transfer daha yaygındır. Çoğu RS-485 linklerde de asenkron transfere başvurulur. Bu asenkron transfer çeşitli formatlarda olabilir. Bunların en yaygını 8-N-1 formatıdır. Bu formatta, gönderici cihaz her bir baytı, 1 adet Start bitini takiben 0 no'lu bitten (en az anlamlı – LSB : Least Significant Bitten) başlayarak 8 veri biti ve 1 adet Stop biti olarak yollar.

8-N-1 ifadesindeki N, iletimde parite biti kullanılmadığını anlatır. Diğer formatlarda hata kontrolünün basit bir formu olarak bir parite biti bulunur. Parite; çift, tek, iz ya da boşluk olabilir. Çift parite, parite bitinin, kendisindeki ve veri bitlerindeki 1'lerin toplam sayısının çift olmasını sağlayacak şekilde ayarlandığını anlatır. Tek parite ise, aynı şekilde, 1'lerin toplam sayısının tek olmasını sağlayacak şekilde ayarlandığını anlatır. Parite kullanan iletimlerde veri iletim sırası başla biti, 8 bit'lik veri, parite biti ve bir ya da daha fazla dur bitleri şeklindedir.

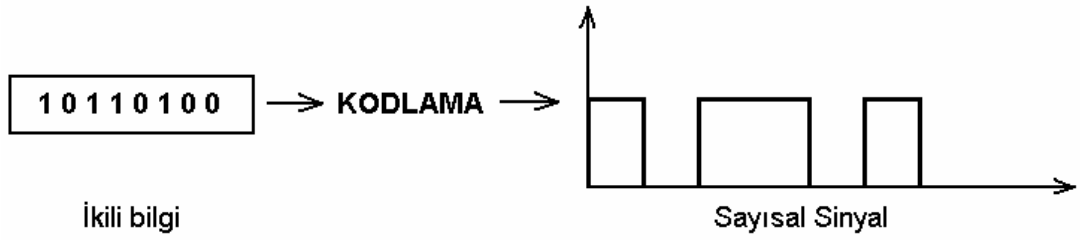
Eş zamanlı olmayan iletim ortamındaki baud oranı ile bilgisayarda tanımlanan bit oranı arasında ince bir fark bulunur. Bunun nedeni eş zamanlı olmayan iletimde oluşturulması gerekli olan başla, parite, dur gibi bitlerin

olmasıdır. Örneğin; 8-N-1 iletimde 8 bit'lik bir veri iletimi için en az 10 bit bilgi iletilmesi gerekir.

3. SAYISAL KODLAMA

3.1. Kodlama

Genel olarak kodlama; ikili bilgiyi elektriksel işarete çevirme işlemidir [7]. Bilgiyi sayısal tabanda göstermek için parçalara böleriz. Her bir parça da ikili bilgiden oluşmaktadır. İkili bilgi ya 1 ya da 0'dır. Bu iki duruma karşılık iki farklı işaret seviyesi belirlenerek kodlama yapılır (Şekil 3.1).



Şekil 3.1. Kodlama

Aşağıda belirtilen üstünlükten dolayı ses işaretlerinin kodlanarak ve dijital format ile iletilmesi önemlidir;

- a. Gürültüye karşı daha az duyarlıdır.
- b. Konuşmalar arasındaki girişim daha azdır.
- c. Distorsiyon seviyesi daha azdır.
- d. Kaybolan seslerin tekrar oluşturulması daha kolaydır.
- e. İletim verimliliği daha fazladır.

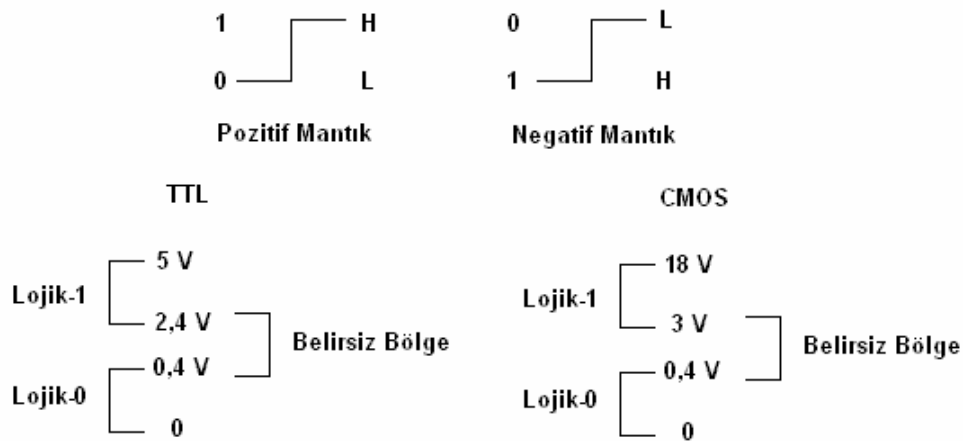
Kodlama işlemi iki temel basamaktan meydana gelmiştir.

1. Enformasyon dizisi, her blok karışık bilgi sembolünden meydana gelecek şekilde mesaj bloklarına ayrılır.

2. Kodlayıcı, bir mesaj bloğunu belirli kurallara göre, kod kelimesi adı verilen n ikili sembolden ($n > k$) oluşan daha uzun mesaj bloğuna dönüştürür.

3.1.1. Bilgiye karşılık gelen işaret seviyeleri

Dijital bilgiler elektronik devreler üzerinde çeşitli işaret seviyeleriyle ifade edilir. Bu durum kullanılan devre elemanına göre değişir. TTL ve CMOS entegreler için geçerli olan işaret seviyeleri fark göstermektedir (Şekil 3.2).



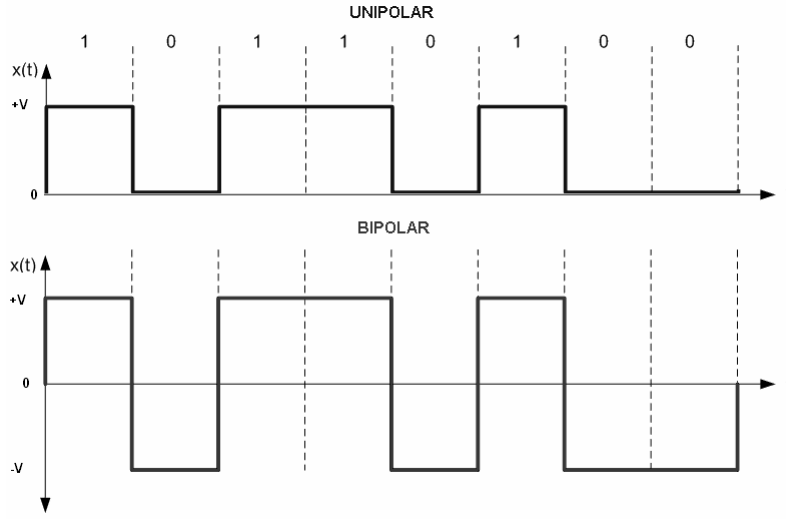
Şekil 3.2. İşaret seviyeleri

3.1.2. Kodlama sınıflandırmaları

Sayısal işaret iletim ortamına kodlanarak çıkarılır. Alıcı taraf bu kodlamaya dayanarak kendisine gelen veriyi algılar. Kodlama yöntemi olarak çeşitli yöntemler kullanılabilir.

Kutup sayısına göre;

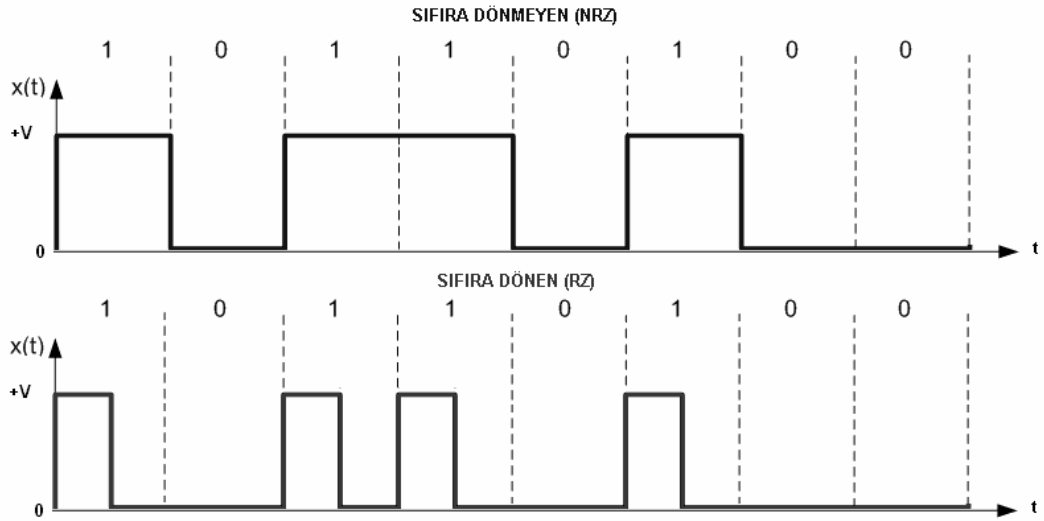
- Tek kutuplu kodlama (Unipolar)
- İki kutuplu kodlama (Bipolar)



Şekil 3.3. Unipolar ve bipolar işaret seviyeleri

Sıfıra dönüp dönmemeye göre;

- Sıfıra dönmeyen kodlama (NRZ: Non-Return to Zero)
- Sıfıra dönen kodlama (RZ: Return Zero)

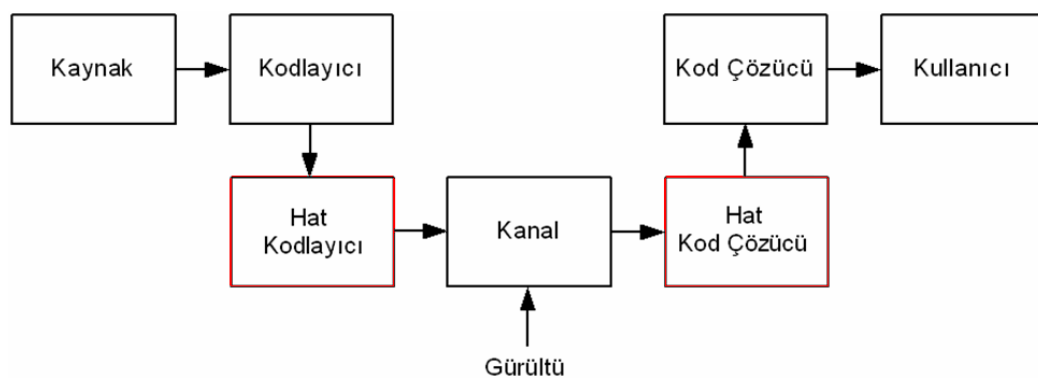


Şekil 3.4. NRZ ve RZ kodlamaları

Bit süresi ortasında geçiş olan yöntemler kenar tetiklemeli yöntem olarak adlandırılır. Genelde LAN fiziksel bağlantı kodlanmasında kullanılır. Düzey tabanlı kodlama ise RS-232 gibi kısa mesafe bağlantılarında kullanılan yöntemidir.

3.2. Hat Kodlaması

Sayısal modülasyon sistemlerinin tasarımlarında önemli bir faktör, gönderilen ve alınan verinin ifade ettiği sayısal kod ya da işaretin tipidir. Bir sayısal kodun seçimi, verinin özelliklerine ve sayısal sistemin sınırlamalarına göre belirlenir [2]. Hat kodlaması, standart mantık düzeylerini iletim hatları için daha uygun hale getirme işlemidir [8].



Şekil 3.5. Hat kodlaması

Sayısal haberleşme sistemlerinde alıcının görevi, verici tarafından gönderilen sayısal veriyi orijinal haliyle yeniden üretmektir. Bunu gerçekleştirebilmesi için, alıcının vericideki ile aynı frekansta bir saat işaretine sahip olması gerekir. Alıcının böyle bir işareti elde etmesinin üç temel yöntemi vardır [2]. Bunlar;

1. Gönderilen bilgi saat bilgisini de içerir. Yani modülasyon kodu ferdi saatlenmelidir. Alıcı, bu bilgiden saat işaretini tahmin eder [2].
2. Saat veya zamanlama bilgisi, başka bir hat üzerinden gönderilebilir. Bu, vericiden alıcıya hat sayısını artırır ve sistemin daha karmaşık olmasına sebep olur [2].
3. Alıcı, vericiden gelen saat işaretlerine güvenmez ve kendi saat işaretini üretebilir [2].

2. ve 3. yöntemler pahalı ve zor olduğundan tercih edilmez. Bunların yerine 1. yöntem tercih edilir. Bu yöntem ucuz ve kolaydır. Ancak sayısal verideki ikili 1 ve 0'ların sayısı gönderilecek bilgiye bağımlı olarak değişken olduğundan, bu veri uzun süreli olarak sadece ikili 1 veya 0 olabilir [2].

Uzun süreli ardışık benzer ikili seviye dizisinin gönderilmesi, iletişim kalitesinin bozulmasına ve istenmeyen iletişim hatalarının üretilmesine sebep olur. Ayrıca alıcının böyle bir işaretten, saat bilgisini elde edip tekrar üretebilmesi de mümkün değildir. Bu problemi gidermek amacıyla, verinin saat işaretini de içermesi istenir. Bu da hat kodlamasıyla gerçekleştirilebilir [2].

Genel olarak aşağıdaki nedenlerden ötürü saf ikilik bit akışları ile yapılan kodlama aşağıdaki nedenlerden dolayı uygun değildir;

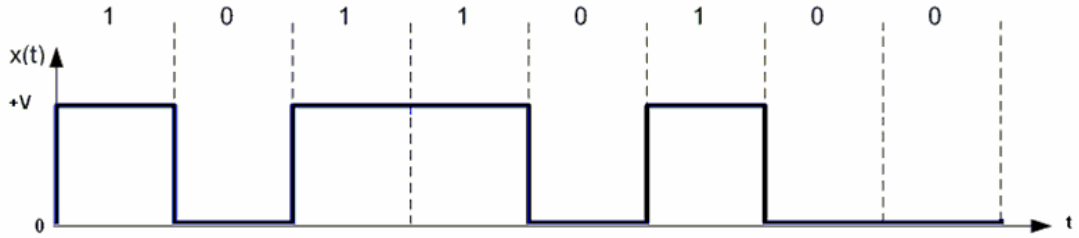
- Hattaki işaretlerin büyük bir DC bileşeninin olması
- Bu nedenle hat işaretlerinin tekrarlayıcılar için gönderilmesi gerekecek DC güçten trafo köprüleri ile ayırmak mümkün olmayacaktır.
- İşaret güç spektrumu alçak frekanslarda büyük bir bileşen içerecek ve bu da aynı kablo üzerindeki ses devrelerinde karışmaya neden olabilir.
- Tekrarlayıcılar bit akışından bir saat işareti elde etmek zorundadır. Frekansın değiştiği ve özellikle ard arda gelmiş pek çok sıfırın bulunduğu durumda bunu elde etmek çok zor olacaktır.

Bu nedenlerden dolayı sayısal iletimde işaretler iletilirken yukarıda bahsedilen unsurların ortadan kaldırılması için sayısal kodlama teknikleri kullanılır.

3.3. Hat Kodu Türleri

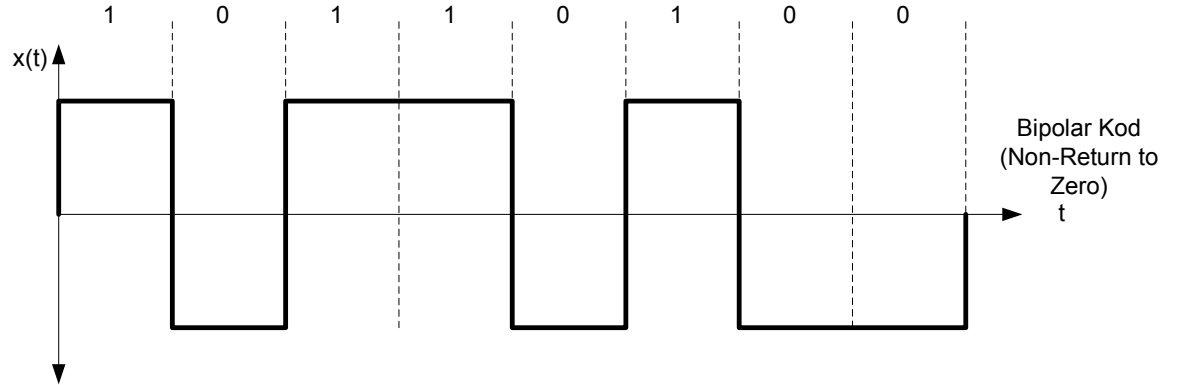
3.3.1. Tek kutuplu NRZ (Unipolar Non-Return to Zero) kodlama

Kodlama çeşitleri içinde en basit ve en ucuz yöntemdir. Sadece bir gerilim seviyesini kullanır. İkili 1 seviyesi belirli bir yüksek gerilim seviyesine, ikili 0 ise 0 volt seviyesine karşılık gelir.



Şekil 3.6. Unipolar NRZ kodlaması

3.3.2. İki kutuplu NRZ (Bipolar Non-Return to Zero) kodlama



Şekil 3.7. Bipolar NRZ kodlaması

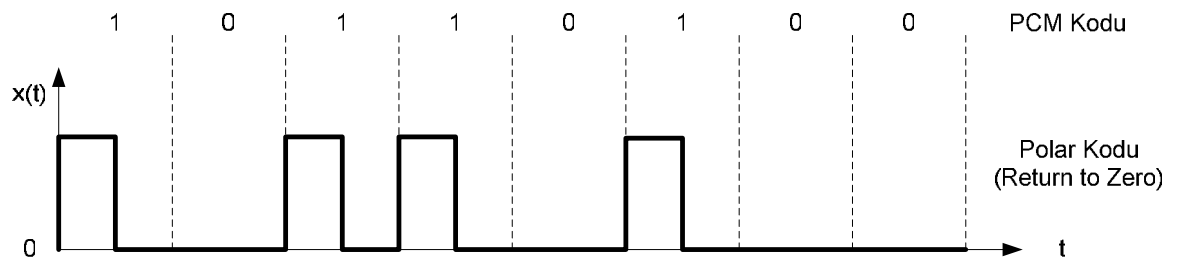
NRZ kod, normal sayısal bilgiye benzer. Sıfıra dönmeyen bu kodlama tekniğinde her bir bit belli bir düzeydeki işaret ile gösterilir. Örneğin ikili 1 yüksek gerilim düzeyi ile, ikili 0 ise düşük gerilim düzeyi ile gösterilir. İkili veri ard arda gönderilirken sürekli olarak düşük ve yüksek gerilim düzeyleri arasında geçiş olur. Hiçbir zaman sıfır gerilim düzeyi kullanılmaz. Bir bitin

gönderim aralığı boyunca ger düzeyi sabittir. Şekil 3.7 sıfıra dönmeyen kodlama tekniği kullanılarak iletilen ikili bir veriyi göstermektedir.

Sıfıra dönmeyen kodlama tekniği, hem asenkron hem de senkron iletimde kullanılır. Genellikle düşük hızlı aktarımlar için uygundur. Kolay uygulanabilen bu tekniğin dezavantajı ard arda aynı tür ikili veri gönderildiğinde (sürekli ikili 0 ya da sürekli ikili 1) gerilim düzeyinde hiçbir değişiklik olmamasıdır. Örneğin bir ethernet yerel ağında binlerce ikili 1 ard arda gönderilebilir. Bu durumda alıcı ve vericinin eşzamanlaması arasındaki en ufak kayma bile gelen işarete bakarak düzeltilemez.

3.3.3. Tek kutuplu RZ (Polar Return to Zero) kodlama

Sıfıra dönen kodlama tekniğinde ikili 1 gerilim düzeyinin yükselmesi ve daha sonra sıfıra geri dönmesi ile gösterilir. İkili 0 ise herhangi bir değişikliğin olmaması ile gösterilir. İkili 1'in gösteriminde bit başlangıcında gerilim seviyesi artar, bitin gönderim aralığının yarısına kadar yüksek düzeyde kalır. Tam yarısında 0 düzeyine düşer ve 0 düzeyinde kalır. Şekil 3.8'de sıfıra dönen bir bit dizisi verilmiştir.

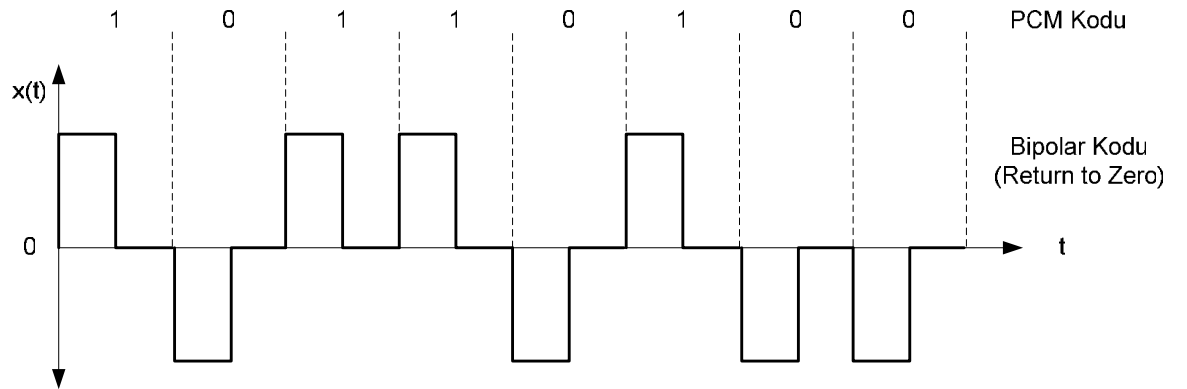


Şekil 3.8. Polar RZ kodlama

Sıfıra dönen kodlama tekniği, sıfıra dönmeyen kodlama tekniği ile karşılaştırıldığında önemli bir kazanım sağlamaz. Sıfıra dönmeyen kodlama tekniğinin aynı türden uzun bit dizilerinin gönderimi konusundaki dezavantajı sıfıra dönen kodlama tekniğinde de vardır.

3.3.4. İki kutuplu RZ (Bipolar Return to Zero) kodlama

İki kutuplu kodlama tekniği, her bitin yüksek ya da düşük gerilim düzeyinden 0 düzeyine dönmesi ile kodlanır. İkili 1, bitin başlangıcında gerilim düzeyinin yükselmesi, bit gönderim genişliğinin tam ortasına gelindiğinde ise tekrar sıfıra düşerek bitin sonuna kadar 0 düzeyinde kalması ile kodlanır. İkili 0 ise aynı şekilde bitin gönderilmeye başlanması ile gerilim düzeyinin düşmesi, bit genişliğinin tam ortasında ise tekrar sıfıra yükselmesi ile kodlanır. Bu kodlama tekniği Şekil 3.9'da gösterilmiştir. Her bir bitin kodlanmasında sıfır düzeyine gelinmesi, oluşan işaretin dönüş genişliğinin yarıya düşmesine neden olur.



Şekil 3.9. İki kutuplu RZ kodlama

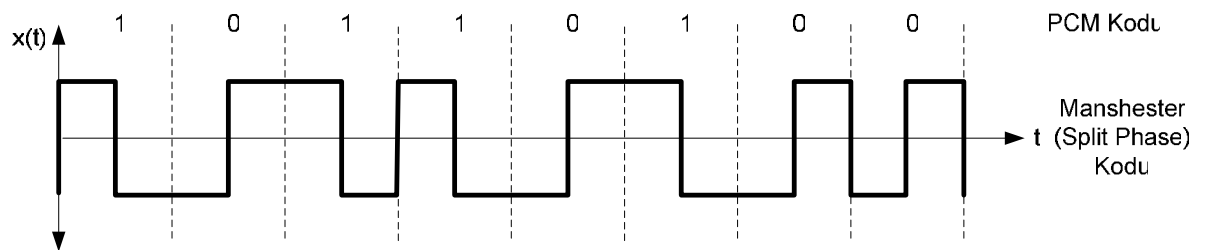
İki kutuplu kodlamada her bir bitin tam ortasında sıfıra geri dönüş vardır. Bu sıfıra dönüş senkron iletimde saat bilgisini taşımak için kullanılır.

3.3.5. Manchester (Split Phase) kodlama

Faz (Phase) kodlaması olarak ta bilinen Manchester kodlamasında bitler iki ayrı gerilim düzeyi arasında bir geçiş oluşturularak kodlanır. Bu nedenle sıfır düzeyine gerek yoktur. Pozitif ve negatif genlik düzeyi yeterlidir [1].

Manchester kodlamada her zaman bölmesi yarıya bölünür. Örneğin ikili 1, zaman bölmesinin yarısında 1, diğer yarısında 0 olarak kodlanır. İkili 0 ise, tam tersidir. Yani ilk yarıda 0, diğer yarıda 1 kodlanır. Manchester kodlama, ikili seri verinin TX bit clock'u ile XNOR'lanmasıdır. İletilen data eşik karşılaştırıcısından geçirildikten sonra alınan veri, RX bit clock'u ile XOR'lanır [1].

Manchester kodlamasında ikili 0 gönderimi düşük gerilim düzeyinden başlar. Bit gönderim genişliğinin tam ortasında gerilim düzeyi yükselir. Bitin tam ortasında yukarı doğru olan bu yükselme gönderilen verinin 0 olduğunu belirtir. Bitin ikinci kısmında ise gerilim düzeyi yüksek kalır. Aynı şekilde 1 kodlanırken gerilim düzeyi yüksekten başlar ve bitin tam ortasında aşağı doğru düşer. İkinci yarıda da düşük düzeyde devam eder. İkili 1 ve 0 arasındaki geçişlerde gerilim düzeyi bir sonra gelen bite göre değişir. Örneğin ikili 0 gönderildikten sonra gelen ikili 1 ise düşük olan gerilim düzeyi ikili 0'in başlangıç noktasını oluşturur, ancak gelen yine ikili 0 ise geçişin sağlanabilmesi için gerilim düzeyinin yüksekten başlaması gerekir. Dolayısıyla iki bit arasında bir geçiş olur. Şekil 3.10'da Manchester kodlaması ile kodlanmış bir PCM veri örneği gösterilmektedir [1].



Şekil 3.10. Manchester kodlama [1]

Manchester kodlama yapılırken, bilgisayar programı ile bir bit süresi yarıya ayrılır ve bitin 0 veya 1 olması durumuna göre yeni darbeler üretilir. Alıcıda, her yarım bit periyodu süresince entegrasyon yapılır ve bitin 0 veya 1 olduğu anlaşılır. Burada önemli olan nokta, verici ile alıcı arasındaki zaman

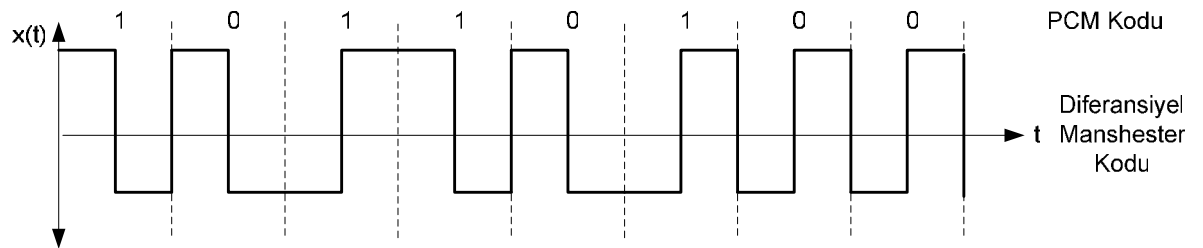
gecikmesinden dolayı ortaya çıkan yaklaşık 1 ya da $2\mu s$ 'lik gecikmedir. Alıcıda, bunun kompanzasyonu gerektiği için zamanlama yapılır [1].

Manchester kodlama, optiksel iletişim sistemlerinde NRZ kodlamaya karşı güçlü bir alternatiftir. NRZ kodlama ile karşılaştırılırsa avantajlarını şöyle sıralayabiliriz [1];

- a. Zamanlama ayrışımında kolaylık.
- b. Hiçbir laser patternine bağımlılık göstermez.
- c. Sıfır DC bileşen içerir.

Ethernet iletişimde seri veriler Manchester kodlama ile hat adaptörüne aktarılır [6].

3.3.6. Diferansiyel Manchester



Şekil 3.11. Diferansiyel Manchester kodlama

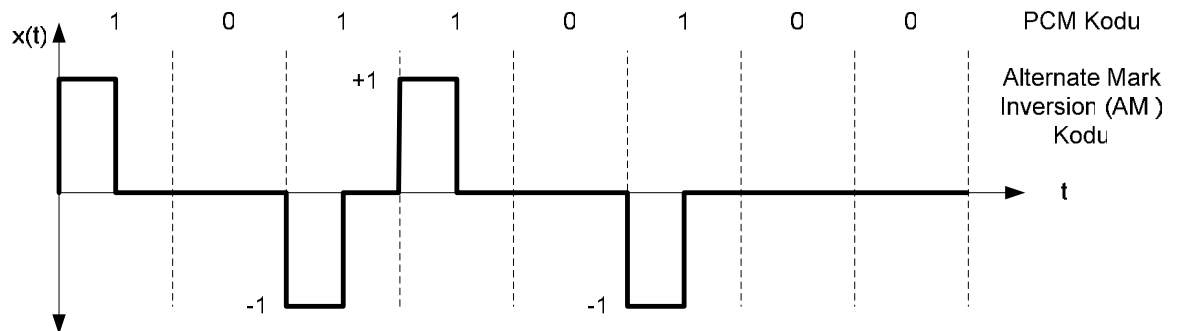
Diferansiyel Manchester kodlamasında, Manchester kodlamasında olduğu gibi her bir bitin ortasında yüksekte alçağa doğru ya da tam tersi yönde bir gerilim değişimi vardır [1]. Ancak gönderilen verinin ikili 1 ya da 0 olduğunu belirleyen bu değişimin pozitiften negatife ya da negatiften pozitif olma değildir [1]. Bitin sonunda bir değişim olup olmaması bir sonraki bitin ne olduğunu gösterir. Ancak bir sonra gelen bit ikili 0 ise gönderilen bitin sonunda gerilim düzeyi değişir, yoksa aynı kalır. Diferansiyel Manchester kodlamasının bir örneği Şekil 3.11'de gösterilmektedir. Her bir bitin ortasında

dönüşüm olduğuna ama ancak bir sonra gelen bit 1 ise sonunda bir dönüşüm gerçekleştiğine dikkat ediniz. Dolayısıyla 1 kodlandığında, bulunduğu konuma göre tam ortasında dönüşüm gerçekleşir.

Diferansiyel Manchester kodlamasında bir veri dizisinin kodlanmış hali, işaretin düşük düzeyden ya da yüksek düzeyden başlamasına göre değişir. Örneğin ilk bit 0 ise ve işaret düşük düzeyde ise, bitin başında bir dönüşüm gerekeceğinden yüksek düzeye çıkar. Ancak aynı veri için hattaki işaret yüksek düzeyde ise yine bir dönüşüm gerekeceğinden düşük düzeyden başlar. Oluşan bu iki form basitçe birbirinin tersidir. Diferansiyel Manchester kodlamasında her bitin ortasında mutlaka bir dönüşüm olması senkron iletimdeki saat bilgisinin kodlanması için kullanılır [1].

3.3.7. AMI (Alternate Mark Inversion) kodlama

İki kutuplu dizilerde ard arda gelen darbeler ters kutupludur. Aynı işaretli (+ veya -) iki darbe ard arda gelemez. İki kutuplu kodun doğru bileşeni yoktur. Girişteki sıfırlar aynı biçimde kodlayıcı çıkışında da görüleceğinden alıcıda zamanlama bilgisinin kaybına neden olabilirler. İki kutuplu kodlarda bu sorunu aşmanın bir yolu ikili giriş dizisindeki bilinen bir sözde rasgele bir işaretle karıştırmaktır. Bu karıştırmanın etkisi giriş dizisindeki 0 ve 1'lerin olasılıklarını eşitleme yönünden olacağından zamanlama bilgisinin kaybolma olasılığı da azaltılmış olur [1].



Şekil 3.12. AMI kodlama [9]

Bu kodlama yönteminde, Şekil 3.12’de görüldüğü gibi, ikili işaretteki 0’lar değişikliğe uğratılmadan 0 olarak kodlanır, 1’ler ise her ikinci “1”’in işareti eksi yapılır. Çift kutuplu bu yöntem sayesinde bit süresi %50 kısaltılmış olur. Yani çift kutuplu olan işaretin temel frekansı karşı gelen tek kutuplu işaretin yarısı kadardır. Böylece bant genişliği de yarıya indirilmiş olur. Ayrıca işaret spektrumu darbe biçiminden bağımsız olarak daima temel frekansı içerir [1].

0’ların 0 olarak ve 1’lerin polarite değiştirilerek gönderilmesi, işaretin doğru akım bileşeninin sıfır olmasını sağlar. AMI kodu, 0 ve iki polaritenin kullanılmasından dolayı sözde üçlü bir koddur. Doğru akım taşımayan AMI kodlama da yeterli uzunlukta darbe gönderilince ortalama değer sıfır olacaktır. Transmisyon ortamı için uygun olan bu kodun pozitif ve negatif darbeler arasındaki geçişlerinden saat işareti çıkarılabilir [1].

AMI kodunun bir dezavantajı, herhangi bir uzunlukta ardı ardına sıfır gelmesi durumudur. Uzun bir süre ardı ardına gelen sıfırlardan, alıcıların saat çıkarması işlemi zordur. Bu problemi çözmek için, AMI kodunun gelişmişleri kullanılır [1].

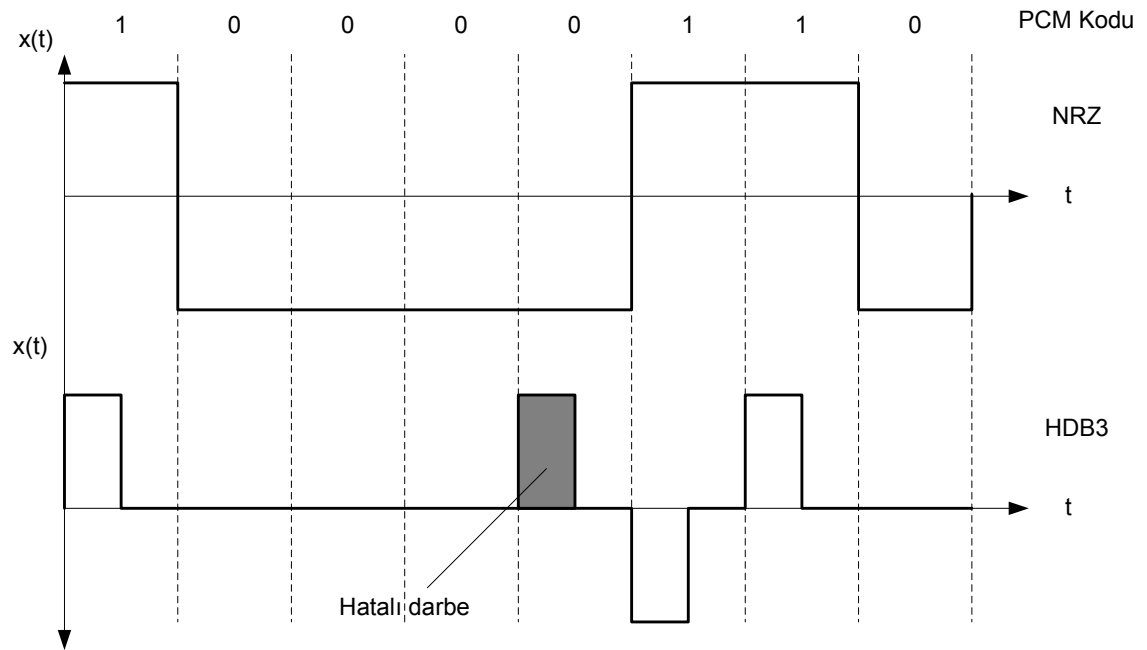
İkili sayı kodunda $2^8 = 256$ olan kombinasyon sayısı -1, 0, +1’den oluşan AMI’de $3^8 = 6561$ ’e çıkar. Ancak AMI’de her artı işaretin ardından bir eksi işaret geldiğinden kapasitenin hemen yarısı kullanılmaz [1].

3.3.8. HDB3 (High Density Bipolar 3) kodlama

HDB3 kodu AMI kodunun gelişmiş bir türevidir. AMI kodundaki çok sayıda ardı ardına gelen 0’lardan saat işareti çıkarılamaması problemi HDB3 kodunda çözülmüştür. HDB3 kodu ardı ardına gelen üçten fazla sayıda sıfıra izin vermez. İşarete ikili sayı elemanları yarım bit süreli ve 1 elemanı genliği aynı kalacak şekilde bir artı, bir eksi, o elemanı da sıfır genlikli olarak belirtilir. Böylece ardışık sıfırların sayısı azaltılır. Çok sayıda ardışık sıfır özellikle birden çok zaman dilimi kullanılan veri iletim sürecinde oluşabilmektedir.

Kodun sonundaki üç rakamından da anlaşılacağı gibi yöntemde en çok 3 ardışık sıfıra izin verilmektedir. HDB3 kodunda AMI kodu gibi her ikinci “1”’in polaritesi değiştirilir. Ardışık dördüncü sıfıra rastlandığında bu sıfırın yerine bir önceki ile aynı polaritede olan bozucu bir kod darbesi yerleştirilir [1].

HDB3 işaretle ardışık üçten fazla sıfır elemanı içeren her seri 4'lü gruplara bölünerek bu grubu 2. ve 3. sıfır elemanı boşluk olarak ve 4. sıfır ise işaret olarak aktarılır, işaretin polaritesi bir öncekinin aynıdır. Bu nedenle AMI kuralını çiğneyen işaret olarak tanımlanır. İlk sıfır ya boşluk ya da işaret olarak iletilir. Bunu izleyen öteki dördüncü sıfırların polaritelerinin bir artı, bir eksi olması sağlanır [1].



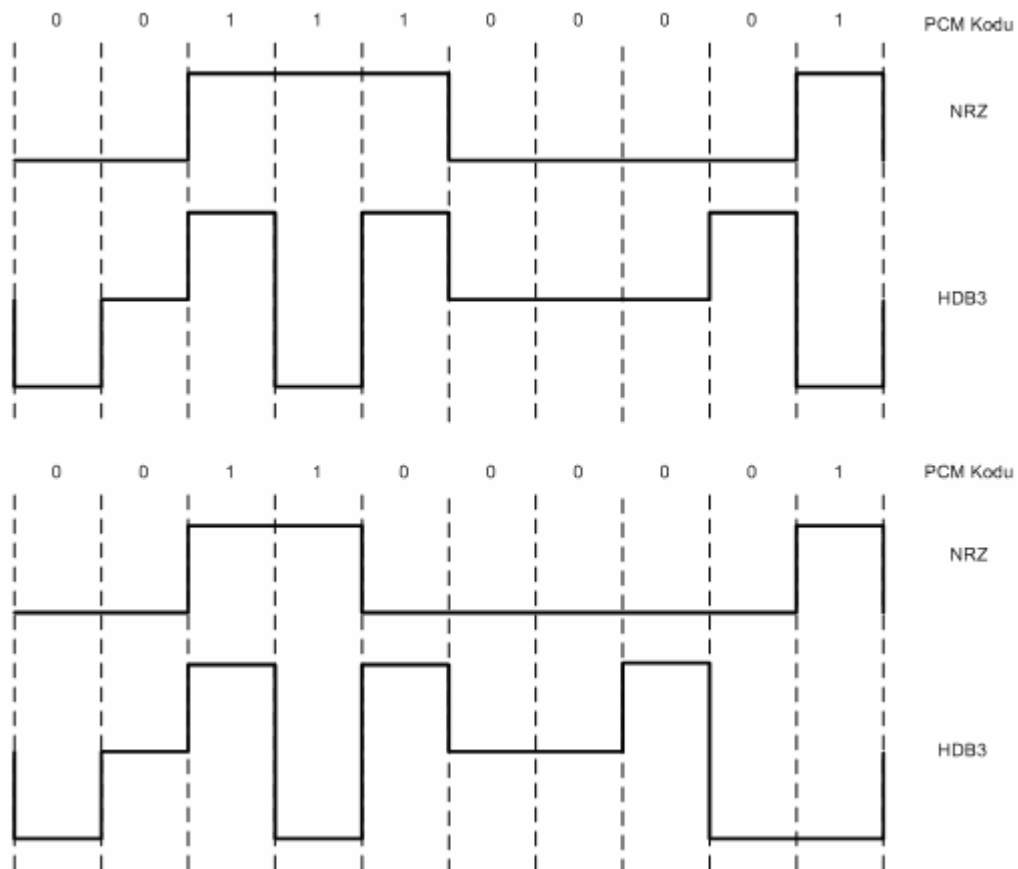
Şekil 3.13. HDB3 kodlama

Alıcı tarafta HDB3 kodunu çözmek için, iki kutuplu kod NRZ koda dönüştürülür. AMI kodu kullanıldığında çok basit olan bu işlem, işareti doğrultulup darbe sürelerini uzatarak gerçekleştirilir. HDB3'te ise bozucu ve fazla darbelerin yok edilmesi gerekir. Bozucu darbeler bir önceki darbe ile aynı polaritede olacağından kolayca algılanıp yerlerine sıfır yerleştirilir. Fazla

olan darbeler bozucu olanların üç konum önündedir. Bunlarda sıfır yapılır. Sonra AMI'de yapıldığı gibi bu darbelerin süreleri uzatılır [1].

Doğru akımlı darbe dizisinden kaçınmak için de birbirini izleyen bozucu darbelerin ters polariteli olmaları sağlanır [1].

Şekil 3.13'te yedinci saat darbesine kadar durum tıpkı HDB3 kodu gibidir. Sonra işaretle 5 ardışık sıfır görülür. Bunların dördüncüsü yerine bir önceki pozitif olduğundan negatif bir bozucu darbe yerleştirilir. Şimdi 1. ve 2. hattı birleştirince 11. darbenin bozucu olmadığı ve aslında AMI kuralına uyduğu görülecektir. Bu nedenle dizideki ilk sıfır yerine fazladan bir darbe eklenir. Bu darbe AMI kuralına uyduğu için 2. hattaki darbelerin bozucu olması sağlanır [1].



Şekil 3.14. HDB3 kodlama örnekleri

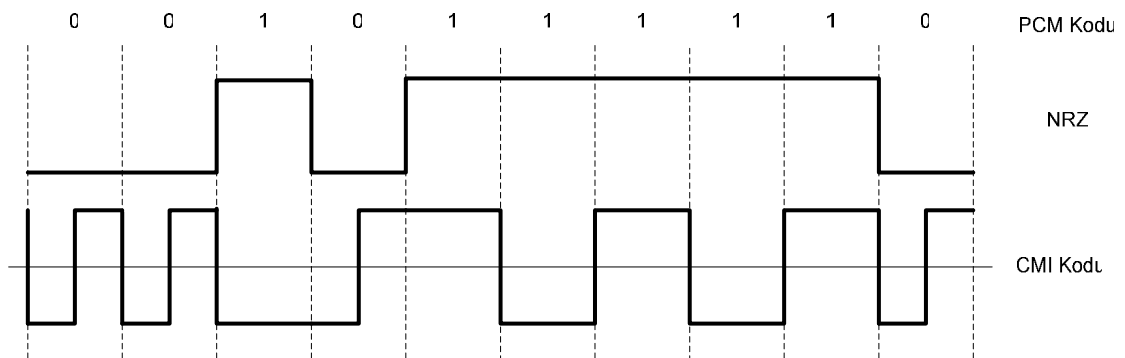
Kısaca özetlersek HDB3 kodu kuralları [1]:

- a. Ardı ardına 3 sıfır görülünceye kadar her ikinci “1”in polaritesi değiştirilir.
- b. Ardışık 3 sıfırdan sonra gelen sıfır varsa bunun yerine bozucu darbe yerleştirilir.
- c. Ardışık bozucu darbelerden biri pozitif, biri negatif olarak seçilir.
- d. Şayet (c) şıkkı uygulanamıyorsa dizideki ilk sıfırın yerine “1” yerleştirilir.

3.3.9. CMI (Coded Mark Inversion) kodlama

50 Mb/s'den düşük bit oranları için HDB3 kodu sayısal bağlantılarda daha kaliteli kullanılır. Yüksek hızlı sayısal bağlantılar için CMI kodu kullanılır. Saat işareti çıkarmak için yüksek enformasyon içeriğine sahip olmasına rağmen, hat hızını iki katına çıkarması bu kodun dezavantajıdır [1].

CMI kodlama yönteminde, 0 işareti (01) olarak, 1 ise ardı ardına değişen “11” ya da “00” işaretleri şeklinde kodlanır (Şekil 3.15).



Şekil 3.15. CMI kodu

Haberleşme hattında oluşan hatalar, kod kuralı gözlenerek kolaylıkla saptanabilir. İki seviyeli olması nedeniyle CMI, fiber optik kanallar için ideal bir kodlama yöntemidir [1].

3.3.10. MCMI (Modified Coded Mark Inversion) kodu

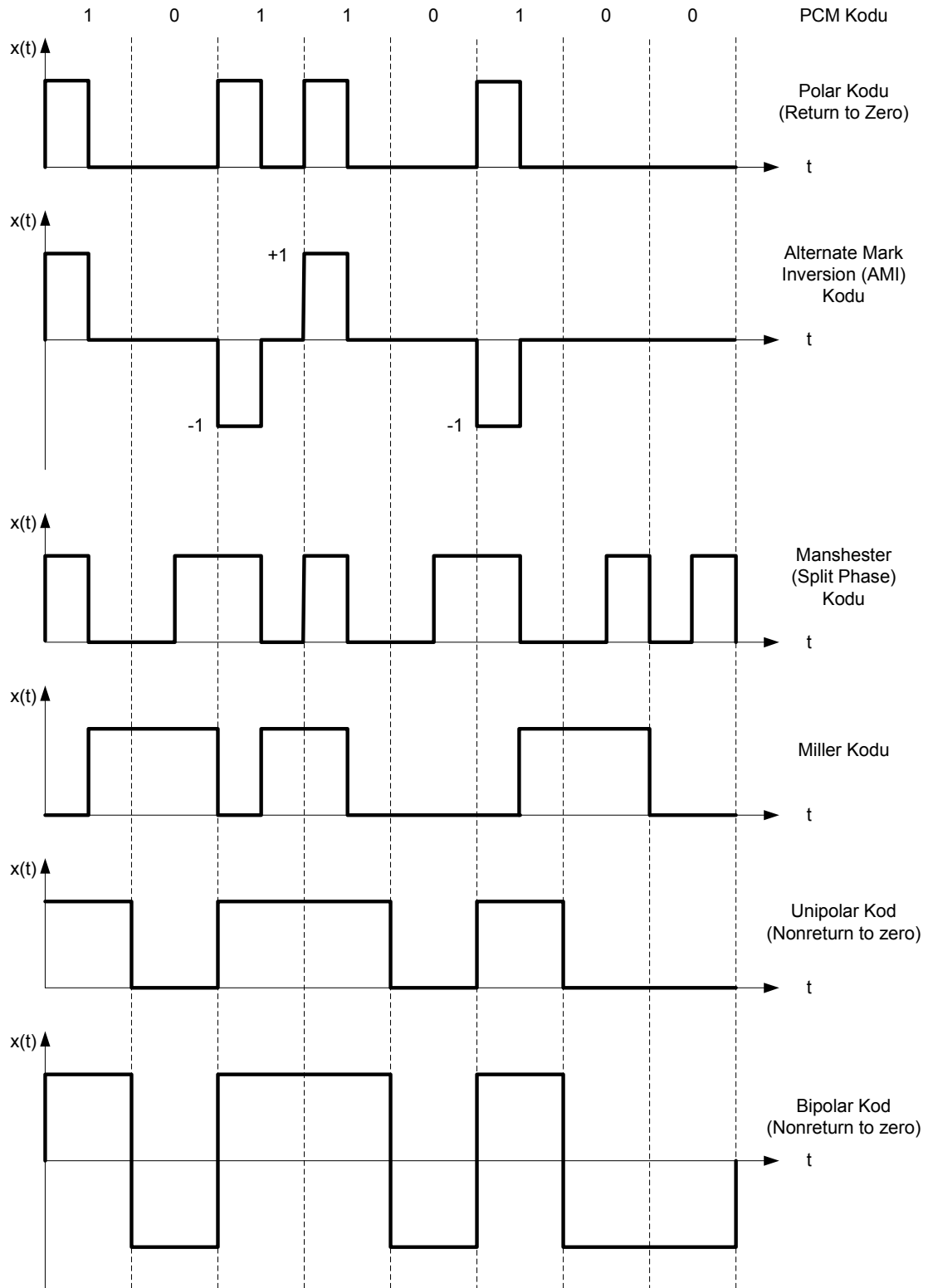
CMI kodunun özel bir türevi olan bu kod ile ikinci bir haberleşme kaynağının aynı hat üzerinden haberleşmesine olanak sağlamıştır. CMI kodlama yöntemine, belirli bir oranda kurala uymayan işaretler katılarak MCMI kodlama yöntemi elde edilmiştir [1].

3.4. Kodların Özellikleri

Çizelge 3.1. Hat kodlarının özelliklerinin karşılaştırılması [2]

Hat kodlaması	Bit başına sembol sayısı	Senkronizasyon	Görev Süresi (%)
UPNRZ	1	Yok	100
BPNRZ	1	Yok	100
UPRZ	2	Yok	0-50
BPRZ	2	Var	0-50
Manchester	2	Var	50
AMI	2	Var	50
CMI	2	Var	50,100

Çizelge 3.1’de Görüldüğü gibi sıfıra dönmeyen (NRZ) kodlamalarında ve tek kutuplu sıfıra dönen (UPRZ) kodlamada senkronizasyon işareti elde edilememektedir. Ancak BPRZ, Manchester, AMI ve CMI kodlamalarında bu işareti elde etmek mümkündür.



Şekil 3.16. Bazı hat kodlama tekniklerinin karşılaştırılması [7]

4. SİMÜLASYONLAR VE DENEYLER

Bu bölümde hat kodu türlerinin daha iyi anlaşılması ve üzerinde çalışma yapılabilmesi çeşitli simülasyonlar ve deneyler yapılmıştır.

Bu tezde yapılan simülasyonlar bilgisayar ortamında elektronik simülasyon programları ve konuyla ilgili destekleyici başka bilgisayar programları kullanılarak yapılmıştır.

Elektronik simülasyon programı olarak, elektronikle ilgilenen çoğu kişinin çok yakından tanıdığı Proteus isimli program kullanılmıştır. Proteus'un seçilme nedeni görsel olması ve sıkça kullanılan mikrodenetleyici simülasyonlarını desteklemesidir.

Konuyla ilgili simülasyonlar bir mikrodenetleyici yardımıyla yapılmıştır. Çeşitliliğe kaçılmaması amacıyla PIC16F87 genel amaçlı mikrodenetleyici kullanılmıştır. Simülasyonlar için çeşitli yazılımlar da yazılmıştır. Bu yazılımlar PicBasic dilinde oluşturulmuştur. Daha sonra makine kodlarına çevrilerek simülasyona eklenmiştir. EK-1'deki deney seti kullanma kılavuzunun içinde PicBasic dilinde hazırlanmış program kodları mevcuttur.

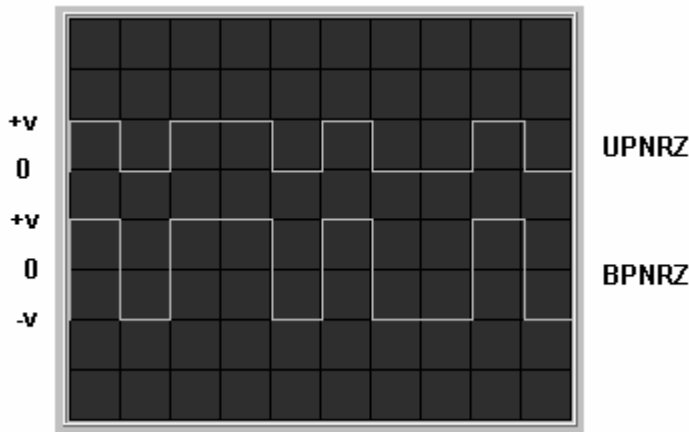
Simülasyonlar ve deneyler sırasında ikili kod olarak "10110100" 'dan oluşan sekiz bitlik bir veri kullanılmıştır.

Ayrıca sayısal hat kodlama teknikleri deney seti geliştirilerek simülasyonlar gerçek ortamda da desteklenmiştir. EK-1'de deney setinin kullanma kılavuzu ve deneyleri vardır.

4.1. Karşılaştırmalar

4.1.1. UPNRZ ve BPNRZ karşılaştırması

Şekil 4.1’de UPNRZ ve BPNRZ kodlamanın simülasyon sonucu zaman eksenindeki dalga şekilleri gösterilmektedir. Açıkça görülüyor ki, bu iki kodlama arasındaki tek fark genliktir. Her iki işaretin sıfır çizgisine bakıldığında UPNRZ işaretinin tek bir gerilim düzeyini (+v), BPNRZ işaretinin ise iki farklı gerilim düzeyini (+v ve -v) görürüz. Bu işaretlerden çıkarılabilecek sonuçların bazıları şunlardır;



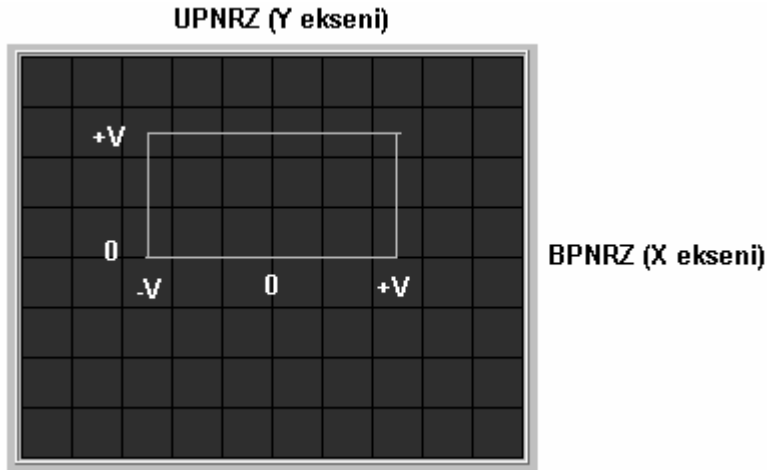
Şekil 4.1. UPNRZ ve BPNRZ karşılaştırması

1. UPNRZ işaretinin iletim hattında her zaman bir DC bileşeni olacaktır.
2. BPNRZ işaretinde ise ikili 1 ve ikili 0'ların eşit olduğu düşünülürse DC bileşen sıfıra çok yakın olacaktır.

Deney setinde voltmetrenin DC kademesinde yapılan ölçümlerde UPNRZ işaretinin 2,5 Volt, BPNRZ işaretinin 0 Volt olduğu görülmüştür.

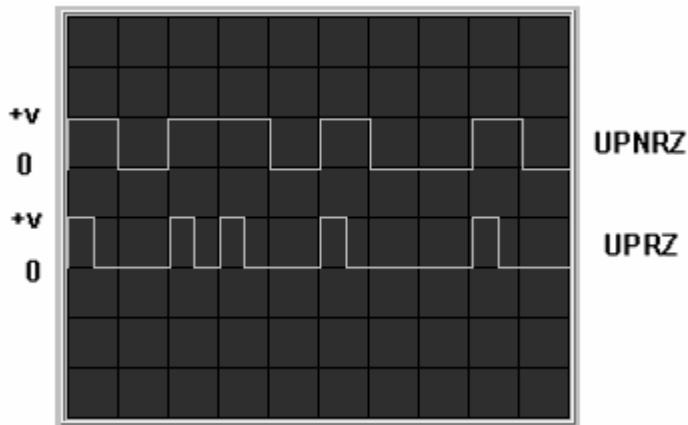
Şekil 4.2’de UPNRZ ve BPNRZ kodlamanın osiloskobun X-Y düzlemindeki hareketi görülmektedir. Y eksenini UPNRZ ve X eksenini BPNRZ işareti olarak

seçildiğinde dikdörtgene benzeyen bir işaret çıkmaktadır. Aslında teorikte ideal olarak iki nokta görülmesi gerekir. Ama işaret seviyeleri arasında her zaman geçişler olacaktır. Bu yüzden iki nokta görünmesi gerçekte imkansızdır.



Şekil 4.2. UPNRZ ve BPNRZ kodlamanın X-Y düzleminde gösterimi

4.1.2. UPNRZ ve UPRZ karşılaştırması

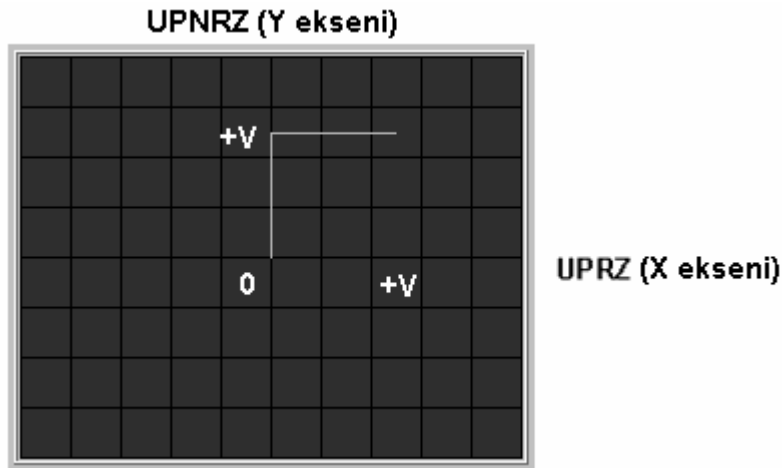


Şekil 4.3. UPNRZ ve UPRZ karşılaştırması

Şekil 4.3'de UPNRZ ve UPRZ kodlamanın simülasyon sonucu zaman eksenindeki dalga şekilleri gösterilmektedir. Her iki işaretin genlik değişimleri eşittir. Ancak periyotları fark göstermektedir. UPRZ kodlamada ikili 0 için

sadece sıfır düzeyi vardır. İkili 1 için ise periyodun yarısı yüksek seviyede iken diğer yarısı sıfır seviyesinde seyretmektedir. Buradan UPRZ kodlamanın UPNRZ'ye göre daha az güçlerde kullanılabileceği çıkarabilir. Bu durum bir avantaj olurken, frekans spektrumunun iki katına çıkması dezavantaj olacaktır.

Deney setinde voltmetre ile yapılan ölçümlerde UPNRZ işaretinin DC bileşeni 2,5 Volt UPRZ işaretinin DC bileşeni 1,25 Volt ölçülmüştür.



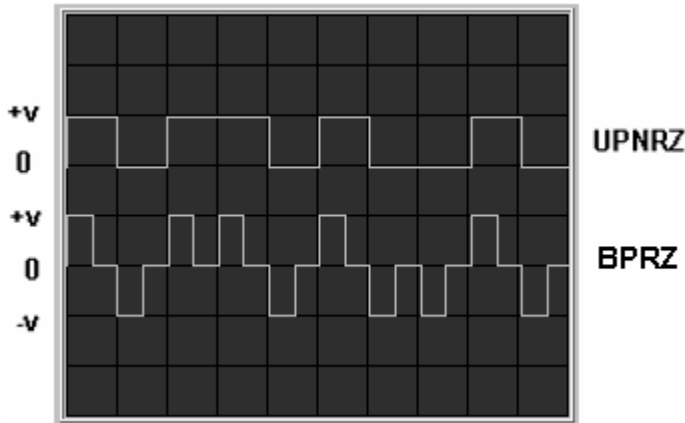
Şekil 4.4. UPNRZ ve UPRZ kodlamanın X-Y düzleminde gösterimi

Şekil 4.4'de UPNRZ ve UPRZ kodlamanın osiloskobun X-Y düzlemindeki hareketi görülmektedir. Y eksen UPNRZ ve X eksen UPRZ işareti olarak seçilmiştir. Aslında teorikte ideal olarak üç nokta görülmesi gerekir.

4.1.3. UPNRZ ve BPRZ karşılaştırması

Şekil 4.5'de UPNRZ ve BPRZ kodlamanın simülasyon sonucu zaman eksenindeki dalga şekilleri gösterilmektedir. Genlik değişimleri bakımından BPRZ iki kat daha fazladır. BPRZ kodlamada ikili 1 için periyodun yarısı yüksek seviyede iken diğer yarısı sıfır seviyesinde seyretmektedir. ikili 0 için ise periyodun yarısı düşük seviyede iken diğer yarısı yine sıfır seviyesinde seyretmektedir. Buradan ilk anda BPRZ kodlamanın UPNRZ'ye göre daha az

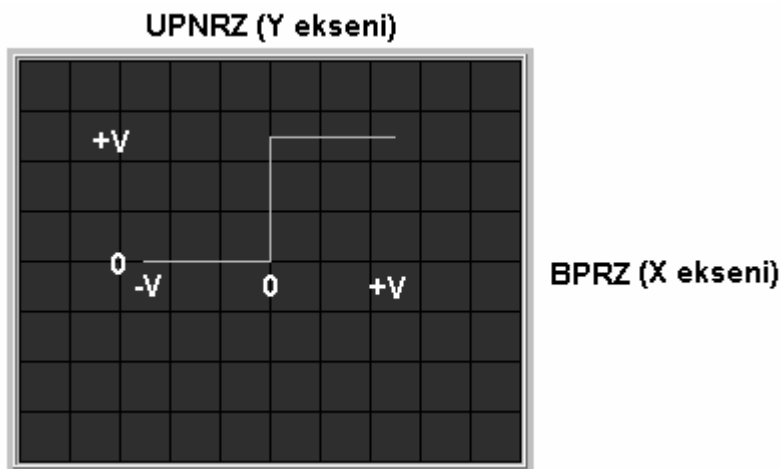
güçlerde çalışabileceği düşünülebilir. Ama bu tam anlamıyla doğru değildir. Çünkü ikili 0 durumunda UPNRZ kodlamada enerji harcanmazken BPRZ kodlamada düşük seviye geçişlerinde bir enerji harcaması olacaktır.



Şekil 4.5. UPNRZ ve BPRZ karşılaştırması

Ayrıca BPRZ kodlaması her bit durumu için değişim gösterdiğinden alıcıda senkronizasyon işaretlerinin üretimi daha kolay olacaktır.

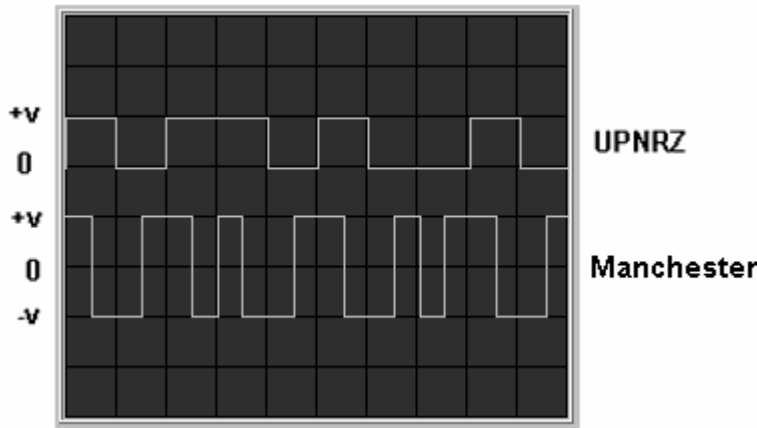
Deney setinde voltmetrenin DC kademesinde yapılan ölçümlerde UPNRZ işaretinin 2,5 Volt, BPRZ işaretinin 0 Volt olduğu görülmüştür.



Şekil 4.6 UPNRZ ve BPRZ kodlamanın X-Y düzleminde gösterimi

Şekil 4.6'da UPNRZ ve BPRZ kodlamanın osiloskobun X-Y düzlemindeki hareketi görülmektedir. Y eksenini UPNRZ ve X eksenini BPRZ işareti olarak seçilmiştir. Aslında teorikte ideal olarak dört nokta görülmeli gerekir.

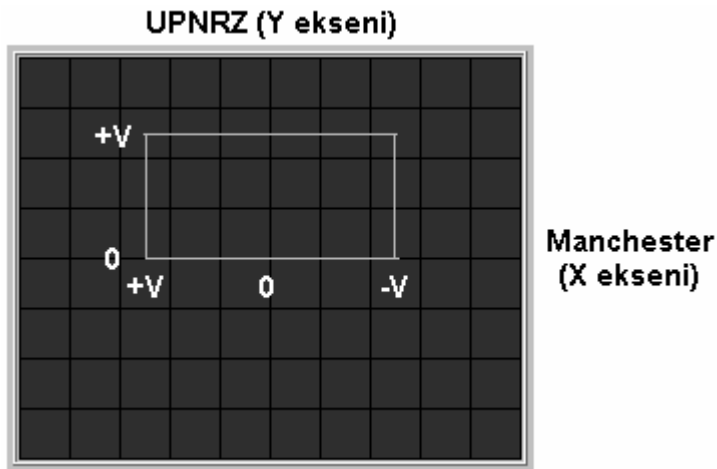
4.1.4. UPNRZ ve Manchester karşılaştırması



Şekil 4.7. UPNRZ ve Manchester karşılaştırması

Şekil 4.7'de UPNRZ ve Manchester kodlamanın simülasyon sonucu zaman eksenindeki dalga şekilleri gösterilmektedir. Genlik değişimleri bakımından Manchester iki kat daha fazladır. Manchester kodlamada ikili 1 için periyodun yarısı yüksek seviyede iken diğer yarısı düşük seviyede seyretmektedir. ikili 0 için ise periyodun yarısı düşük seviyede iken diğer yarısı yüksek seviyede seyretmektedir. Her ikili bilgi için periyodun yarısında bir değişim söz konusudur. Bu durum alıcıda senkronizasyon işaretlerinin üretimini kolaylaştıracaktır. Ayrıca Manchester kodlamada gönderilecek olan ikili bit dizisi ne şekilde olursa olsun güç sabit olacaktır. Çünkü hem ikili 0 hem de ikili 1 için düşük ve yüksek seviye gerilimleri eşit olup periyodu yarı yarıya paylaşmaktadır.

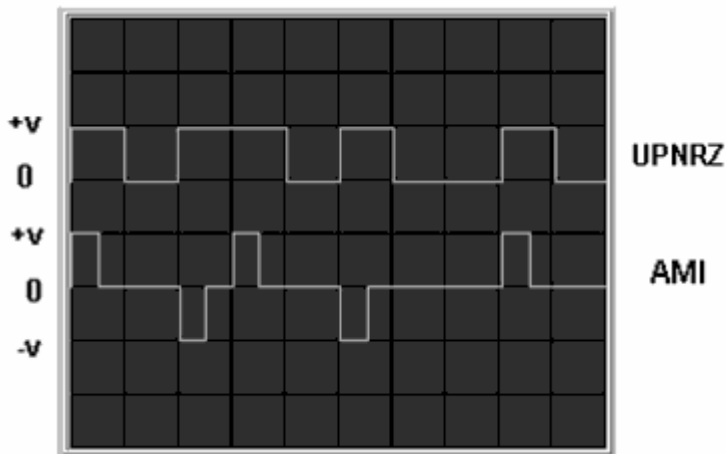
Deney setinde voltmetrenin DC kademesinde yapılan ölçümlerde UPNRZ işaretinin 2,5 Volt, Manchester işaretinin 0 Volt olduğu görülmüştür.



Şekil 4.8. UPNRZ ve Manchester kodlamanın X-Y düzleminde gösterimi

Şekil 4.8'de UPNRZ ve Manchester kodlamanın osiloskobun X-Y düzlemindeki hareketi görülmektedir. Y eksen UPNRZ ve X eksen manchester işareti olarak seçilmiştir. Aslında teorikte ideal olarak dört nokta görülmeli gerekir.

4.1.5. UPNRZ ve AMI karşılaştırması

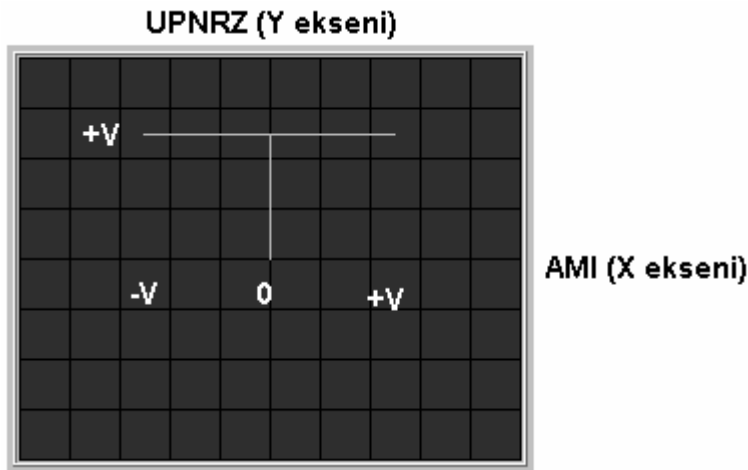


Şekil 4.9. UPNRZ ve AMI karşılaştırması

Şekil 4.9'da UPNRZ ve AMI kodlamanın simülasyon sonucu zaman eksenindeki dalga şekilleri gösterilmektedir. Genlik değişimleri bakımından

AMI iki kat daha fazladır. AMI kodlamanın en belirgin avantajları güç harcaması ve düşük ve yüksek seviyelerin eşitliğidir.

Deney setinde voltmetrorenin DC kademesinde yapılan ölçümlerde UPRZ işaretinin 2,5 Volt, AMI işaretinin 0 Volt olduđu görülmüştür.



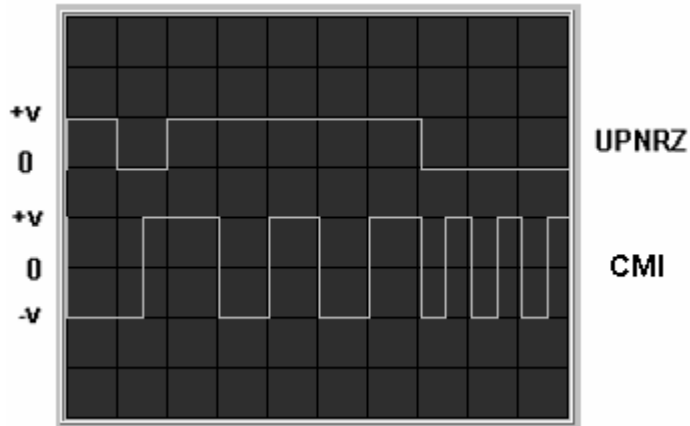
Şekil 4.10. UPRZ ve AMI kodlamanın X-Y düzleminde gösterimi

Şekil 4.10'da UPRZ ve AMI kodlamanın osiloskobun X-Y düzlemindeki hareketi görülmektedir. Y eksen UPRZ ve X eksen AMI işareti olarak seçilmiştir. Aslında teorikte ideal olarak dört nokta görülmeli gerekir.

4.1.6. UPRZ ve CMI karşılaştırması

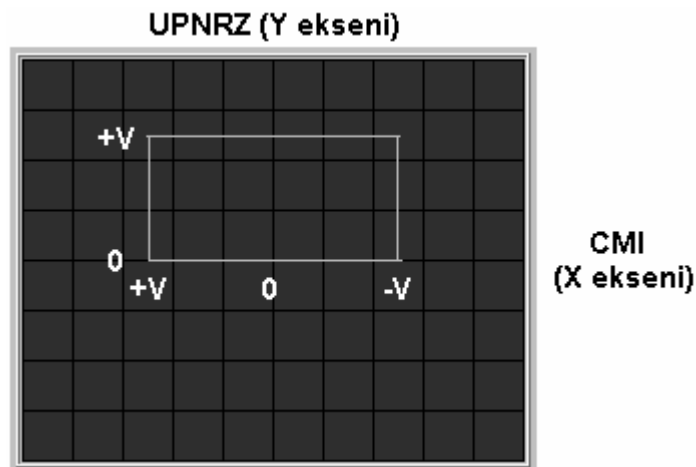
UPRZ ve CMI kodlamanın osiloskopta zaman eksenindeki dalga şekilleri Şekil 3.26'da gösterilmiştir. Rahatça görüldüğü gibi CMI kodlamada belirli bir kuralda ilerleme vardır.

Deney setinde voltmetrorenin DC kademesinde yapılan ölçümlerde UPRZ işaretinin 2,5 Volt, CMI işaretinin 0 Volt olduđu görülmüştür.



Şekil 4.11. UPRZ ve CMI karşılaştırması

Şekil 4.12’de UPRZ ve CMI kodlamanın osiloskobun X-Y düzlemindeki hareketi görülmektedir. Y eksenini UPRZ ve X eksenini CMI işareti olarak seçilmiştir. Aslında teorikte ideal olarak dört nokta görülmeli gerekir.



Şekil 4.12. UPRZ ve CMI kodlamanın X-Y düzleminde gösterimi

5. SONUÇ VE ÖNERİLER

Sayısal hat kodlama teknikleri konusunda yapılan gerek bilgisayarlı simülasyonlar gerekse tasarlanan deney setinde yapılan uygulamalar işaretlerin daha iyi yorumlanmasını sağlamıştır. Öğrencilerin konuyu daha iyi kavramasına fırsat tanınmıştır.

Deney setinin diğer deney setlerinden farklı olarak programlanabilir olması, bir esneklik kazandırmıştır. Böylece kullanıcı deney setinde gördüğü eksiklikleri program yazarak düzeltebilir.

Benzer deney setlerinde genellikle her kodlama için bir çıkış kanalı ve bir çok devre kullanılır. Çıkışlar deney seti üzerinde yer kaplar. Bu da deney setinin fiziksel yapısını büyültür. Tasarlanan deney setinde programlanabilir iki kanal kullanarak bu durumun önüne geçilmiştir.

Deney setinde USB seri portu vardır. Böylece gelecekte başka programlar da yazılarak çeşitli uygulamalar geliştirilebilir.

Herhangi bir sayısal haberleşme sisteminde kullanılacak olan hat kodlama tekniğinin sistemin özelliklerine göre belirlenmesi uygun olacaktır. Öncelikle sistemin çalışmasının iyice incelenmesi gerekir.

Sistemde gönderilecek ikili dizi, bazı durumlarda fazla değişmeyen veya çok değişen bilgiye sahip olabilir. Hat kodlama tekniği seçilirken buna dikkat edilmelidir.

Ayrıca sistemde kullanılan iletim yolunun da iyice gözlenmesi gerekir. Bu doğrultuda iletim hattının gürültü, sönümlenme, frekans kaymaları, minimum ve maksimum çalışma frekansları iyi bilinmelidir ki ona göre hat kodlama tekniği seçilsin.

Deney setine bir analog-sayısal dönüştürücü eklenerek, değişik analog bilgi işaretlerinin sayısallaştırılarak kodlanması ve iletilmesi gözlenebilir.

Literatürlerde birçok sayısal hat kodlama tekniğine rastlandığı halde sisteme özgü hat kodlama tekniği de geliştirebilir.

Kullanımı gayet kolay olan deney setinde, deneylerin başarılı bir şekilde gerçekleştirilebilmesi için kullanım kılavuzunun okunması faydalı olacaktır.

Sayısal haberleşme sistemlerinde önemi olan temel band sayısal veri iletiminin çeşitli hat kodlamaları kullanılarak yapılmasının avantaj ve dezavantajlarını öğrencinin laboratuvar ortamında uygulama yaparak görebilmesi sağlanabilir.

Üniversitelerin telekomünikasyon veya sayısal haberleşme laboratuvarları için alan derslerine (haberleşme sistemleri, sayısal haberleşme vb.) uygun olarak sayısal hat kodlama teknikleri konusunda farklı amaçlara yönelik deney setleri hazırlanabilir. Bu tezde yapılan deney setinin içinde bulunan mikrodenetleyiciyi programlamak suretiyle bu işlem rahatlıkla gerçekleştirilebilir.

KAYNAKLAR

1. Kaçar, B., "Sayısal Kodlama Teknikleri ile Bilgisayarlar Arası İletişim", Yüksek Lisans Tezi, **Gazi Üniversitesi Fen Bilimleri Enstitüsü**, Ankara, 5-33 (2005).
2. Karahan, S., "Dijital Haberleşme Teknolojisinin İncelenebilmesi İçin Bir Eğitim Setinin Geliştirilmesi", Yüksek Lisans Tezi, **Gazi Üniversitesi Fen Bilimleri Enstitüsü**, Ankara, 10-21 (1994).
3. Yavuz, A.H., "Haberleşme Sistemlerinde Kullanılan Kodların Tanıtımı ve Bilgisayarlı Analizi", Yüksek Lisans Tezi, **Gazi Üniversitesi Fen Bilimleri Enstitüsü**, Ankara, 1-7 (2005).
4. Çölkesen, R., Örencik, B., "Bilgisayar Haberleşmesi ve Ağ Teknolojileri", **Papatya Yayıncılık**, İstanbul, 5-9,101-102 (2000).
5. Tooley, M., "Veri İletişimi", **Era Bilgi Sistemleri Yayıncılık**, İstanbul, 60, 72-73 (1997).
6. Erdem, O.A., "Bilgisayar Haberleşme Teknolojisi", **Nobel Yayın Dağıtım**, Ankara, 38, 89-90 (1998).
7. Kayran, A.H., Panayırıcı, E., Aygölü, Ü., "Sayısal Haberleşme", **Birsan Yayınevi**, İstanbul, 89-90 (1995).
8. Tomasi, W., "Elektronik İletişim Teknikleri", **Milli Eğitim Basımevi**, İstanbul, (2002).
9. Bissel, C.C., Chapman, D.A., "Digital Signal Transmission", **Cambridge University Press**, Great Britain, 149-153 (1992).

EKLER

EK-1
SAYISAL HAT KODLAMA DENEY SETİ
KULLANMA KILAVUZU ve DENEYLER

SAYISAL HAT KODLAMA TEKNİKLERİ DENEY SETİ

KULLANMA KILAVUZU ve DENEYLER

İÇİNDEKİLER

1. GENEL BİLGİLER

1.1. Amaç

1.2. Kullanım Şartları

2. BLOK DİYAGRAMLAR

2.1. Deney Seti Devreleri

2.2. Ön Panel

2.2.1. Tuş takımı

2.2.2. LCD Gösterge

2.2.3. Kanallar

2.2.4. Güç Çıkışı

2.3. Arka Panel

2.3.1. Açma – kapama anahtarı

2.3.2. Sigorta

2.3.3. USB Bağlantı Noktası

3. DENEY SETİNİN ÇALIŞTIRILMASI

4. PROGRAMIN TANITILMASI

4.1. Ana Menü

4.1.1. Deneyi başlat

4.1.2. PC'ye gönder

4.1.3. Ayarlar

4.1.3.1. İkili dizi

4.1.3.2. Kanallar

4.1.3.3. İletim hızı

5. ŞEMALAR VE BASKI DEVRELER

5.1. Deney Seti Anakart Devre Şeması

5.2. Deney Seti Anakart Baskı Devreleri

6. DEVRE ELEMANLARI BACAK BAĞLANTILARI

6.1. PIC16F877 Bacak Bağlantısı

6.2. TL084 Bacak Bağlantısı

6.3. MAX232 Bacak Bağlantısı

6.4. 4x20 LCD HY-2004 Bilgileri

7. PROGRAM KODLARI

8. ELEMAN VE FİYAT LİSTESİ

9. DENEYLER

DENEY 1. UPNRZ ve BPNRZ karşılaştırması

DENEY 2. UPNRZ ve UPRZ karşılaştırması

DENEY 3. UPNRZ ve BPRZ karşılaştırması

DENEY 4. UPNRZ ve Manchester karşılaştırması

DENEY 5. UPNRZ ve AMI karşılaştırması

DENEY 6. UPNRZ ve CMI karşılaştırması

1. GENEL BİLGİLER

Günümüzde sayısal haberleşme sistemleri, analog haberleşme sistemlerine göre avantajların çok olması nedeniyle hızla yaygınlaşmaktadır. Bu sebepten sayısal sistemlerdeki gelişmelerin takibi sayısal haberleşme konularındaki çalışmalarla paralel ilerlemektedir.

Bu deney seti, sayısal haberleşmenin konularından olan sayısal hat kodlaması konusu üzerinde gerçek ortamdaki sonuçları gözlemlemek üzere tasarlanmıştır.

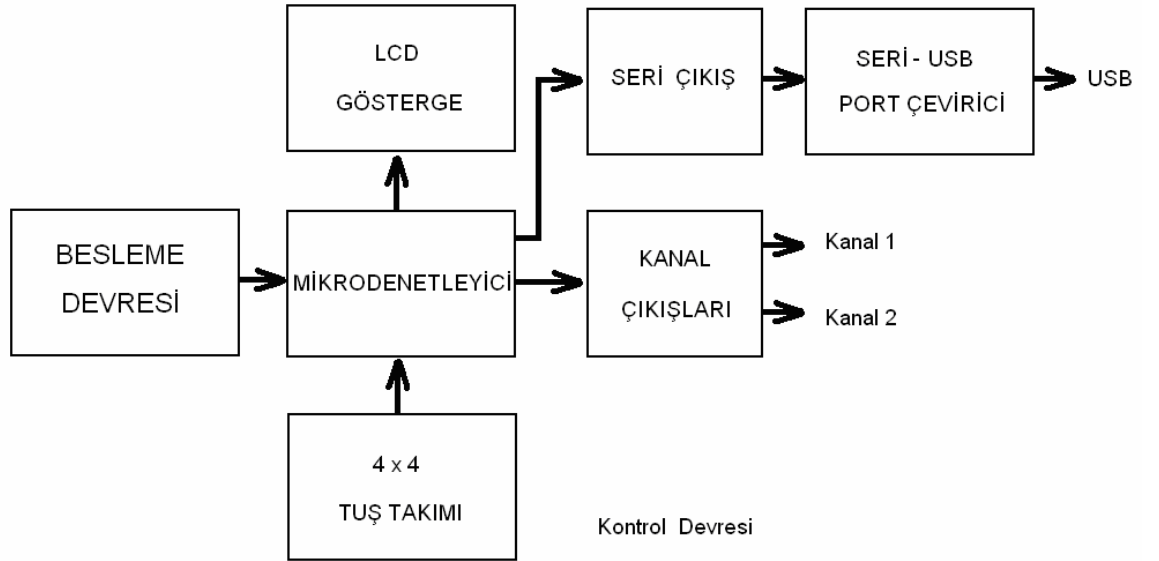
1.1 Amaç

- Öğrencilerin sayısal hat kodlama tekniklerini daha iyi kavramasını sağlamak,
- Laboratuvar ortamında karşılaştırmalar yapmalarına imkan sağlamaktır.

1.2 Kullanım Şartları

- Deney seti 220 V şebeke gerilimi ile çalışmaktadır.
- Harici bir güç kaynağına ihtiyaç duymamaktadır. (Dışardan bağlanan herhangi bir güç girişi deney setine büyük zararlar verebilir.)
- Aşırı derecede radyo frekans (RF) dalgaları içeren (radar, baz istasyonları vs.) yerlerde kullanılması uygun değildir.
- Deney seti itme, çarpma, düşme gibi etkilere maruz kalmamalıdır.
- Güneş altında bırakılmamalıdır.
- Olağan dışı akım kaçakları düşünülerek deney setini korumak üzere 500 mA'lık bir sigorta takılmalıdır. (Zaten deney setinin sigorta takılmadan çalıştırılması mümkün değildir.)

2. BLOK DİYAGRAMLAR



Deney seti devresi şekilde görülen blok diyagramdan oluşmaktadır. Deney setinde bir besleme devresi, kontrol devresi ve seri-USB port çevirici devresi bulunur.

2.1 Deney Seti Devreleri

Deney setinin merkezi kontrollerini yapmak PIC16F877 mikrodnetleyici entegresi kullanılmıştır.

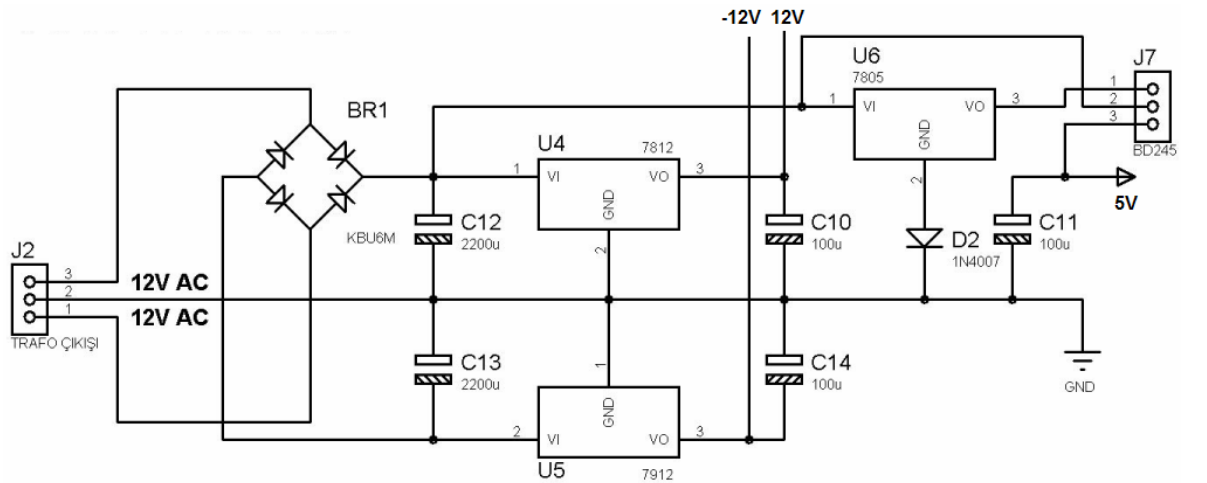
PIC16F877 ve LCD bağlantısı şekildeki gibidir. Kullanılan paralel LCD mikrodnetleyiciyle sürülür. Bunun için 4 veri bağlantısı ve 2 kontrol bağlantısı kullanılır. LCD'nin diğer 4 ucu besleme, kontrast ayarı ve yazma-okuma gibi bağlantı uçlarıdır. Kontrast ayarlaması için deney seti devresinin içinde 10 K'lık bir trimpot vardır. Gerekli görüldüğünde deney seti açılarak bu ayarlama yapılabilir.

Deney setinde 2 kanal olduđu için mikrodenetleyici çıkışında her kanal için 2 olmak üzere toplam 4 çıkış pini bulunur. Bu çıkışlar çıkarıcı opamp devreleriyle her kanal için bir çıkışa düşürölür.

PIC ve 4x4'lük tuş takımı 4 giriş ve 4 çıkış bağlantısıyla bağlıdır. Mikrodenetleyici gerekli kontrol işaretlerini üreterek tuş takımını okur. Tabi ki bu mikrodenetleyici içine yazılmış olan programla gerçekleştirilir.

Deney setinde bulunan besleme devresi üç farklı gerilim seviyesinde çıkışa sahiptir. Karışıklık olmaması için besleme devresinden çıkan kabloların her biri renkli seçilmiştir.

Kablo Rengi	Çıkış Değerleri
SARI	+ 12 Volt
KIRMIZI	+ 5 Volt
MAVİ	0 Volt
YEŞİL	- 12 Volt



Şekildeki blok diyagramda da görüldüğü gibi deney setinin besleme devresinde üç adet regüle entegresi ve bir güç transistorü kullanılmıştır.

2.2 Ön Panel



Deney setinin ön paneli şekildeki gibidir. Ön panel;

- 1 – 4x4 Tuş takımı
- 2 – 4x20 LCD Gösterge
- 3 – Kanallar
- 4 – Güç Çıkışı

oluşturmaktadır.

2.2.1 Tuş Takımı

Deney seti, değişik amaçlarda kullanılmak üzere 4x4'lük 16 adet tuştan oluşan bir tuş takımına sahiptir. Genel olarak deneyi başlatma ve bitirme. Deneyle ilgili bilgilerin girilmesi ve ayarlamaların yapılması için kullanılır.



Tuşların fonksiyonları aşağıdaki gibidir.

1, 2, 3, 4, 5, 6 , 7, 8, 9 ve 0 tuşları - Menü seçimi ve rakam girişleri için kullanılır.

→ ← **Sol ve sağ ok tuşları** – Menülerde seçim yaparken kullanılmaktadır.

↑ **Yukarı ok (Geri) tuşu** – Bir üst menüye dönmek veya deneyi bitirmek için kullanılır.

Diğer Tuşlar – Herhangi bir işlem fonksiyonları yoktur.

2.2.2 LCD Gösterge

Deney setindeki LCD gösterge yaptığımız ayarlamaları ve deney sırasında bilgileri görmemiz için kullanılır. 4 satır ve 20 sütundan meydana gelmektedir. Ayrıca daha net bir görüntü için aydınlatması da mevcuttur.



Kontrast ayarlaması için deney seti devresinin içinde bir potansiyometre vardır. Gerekli görüldüğünde deney seti açılarak bu ayarlama yapılabilir.

2.2.3 Kanallar



Deney setinde iki kanal çıkışı bulunmaktadır. Bunlar kanal 1 ve kanal 2'dir.

Kanal çıkışları iki şekilde alınabilmektedir. İlki şase ve canlı uca sahip olan mavi bağlantı noktalarıdır. İkincisi ise osiloskop bağlantısını sağlayan soketlerdir.

Her iki kanal çıkışı iki adet LED'e sahiptir. Kırmızı LED deney sırasında kanalda +5 Volt bulunduğunu, Yeşil LED deney sırasında kanalda -5 Volt bulunduğunu göstermektedir. Kanalda 0 Volt olduğunda LED'lerin her ikisi de sönük kalacaktır. Gerçekte hiçbir zaman LED'ler aynı anda yanmaz.

Kanal LED'lerinin durumu		Kanal Çıkışı
Kırmızı LED	Yeşil LED	
Yanık	Sönük	+ 5 Volt
Sönük	Yanık	- 5 Volt
Sönük	Sönük	0 Volt

2.2.4 Güç Çıkışı

Deney setinde harici bir devre bağlanılabileceği düşünülerek güç çıkışı verilmiştir. Her bir LED farklı çıkış gerilimini göstermektedir. Bağlantı noktaları kullanılarak rahat bir şekilde çıkış alınması mümkündür.



Güç LED'leri	Çıkış Değerleri
SARI	+ 12 Volt
KIRMIZI	+ 5 Volt
MAVİ	0 Volt
YEŞİL	- 12 Volt

2.3 Arka Panel

Deney setinin arka paneli güç girişi, açma – kapama anahtarı, sigorta ve USB bağlantı noktasının bulunduğu kısımdır.



2.3.1 Açma – kapama anahtarı

Bu anahtar 220 Volt'luk gerilimi deney setine uygulamak için kullanılır. Anahtar kapandığında deney seti çalışmaya başlayacaktır. Deney bittiğinde bu anahtarı kapatmak gerekir.

2.3.2 Sigorta

Deney setini korumak üzere 500 mA'lık bir sigorta takılması için sigorta yuvası konulmuştur.

Sigortanın takılıp çıkartılması, deney setinin fişi çekilmiş durumdayken yapılmalıdır.

2.3.3 USB Bağlantı Noktası

Deney seti ile bilgisayar arasında bağlantı kurmak için kullanılır.

3. DENEY SETİNİN ÇALIŞTIRILMASI

Deneylere geçmeden önce deney setinin ilk çalıştırılması sırasında aşağıdaki işlemler yapılmadır;

- 1- Deney seti üzerindeki tüm kabloları çıkartınız.
- 2- Deney seti üzerinde sigorta olup olmadığını kontrol ediniz. Yoksa bir 500 mA'lık bir sigorta takınız.
- 3- Açma – kapama anahtarını kapatınız. OFF konumuna getiriniz.
- 4- Fişi şebeke gerilimine bağlayınız.
- 5- Açma – kapama anahtarını açınız.

Deney setiniz kullanıma hazırdır.

4. PROGRAMIN TANITILMASI

Deney seti temel olarak iki adet menüye sahiptir. Bunlar ilk açılışta karşımıza çıkan “Ana Menü” ve ayarlamaları yaptığımız “Ayarlar” menüsüdür.

4.1 Ana Menü

Kullanıcıya üç seçenek sunan bir menüdür. Bu seçenekler “Deneyi Başlat”, “PC’ye Gönder” ve “Ayarlar”dır.

```
--- ANA MENU ---  
1 - DENEYİ BASLAT  
2 - PC'YE GÖNDER  
3 - AYARLAR
```

“Deneyi Başlat” seçeneğini seçmek için tuş takımından ‘1’ tuşuna, “PC’ye Gönder” seçeneğini seçmek için tuş takımından ‘2’ tuşuna, “Ayarlar” seçeneğini seçmek için tuş takımından ‘3’ tuşuna basılmalıdır.

4.1.1 Deneyi Başlat

Deneyle ilgili ayarlamalar yapıldıktan sonra ana menüde ‘1’ tuşuna basınca karşımıza çıkar. Deneyi başlatmaktadır.

```
İKİLİ DİZİ  
1 0 1 1 0 0 1 0  
K1-UPNRZ K2-AMI  
HIZ : 1200 Hz
```

Bu seçenekte yaptığımız deneyle ilgili çeşitli bilgiler gösterilmektedir. Bunlar;

İkili Dizi – Ayarladığımız 8 bitlik ikili dizi verisini,

K1 – Kanal 1’de kullanılan sayısal hat kodlama tekniğini,

K2 – Kanal 2’de kullanılan sayısal hat kodlama tekniğini,

HIZ – Gönderdiğimiz verinin iletim hızını,

göstermektedir.

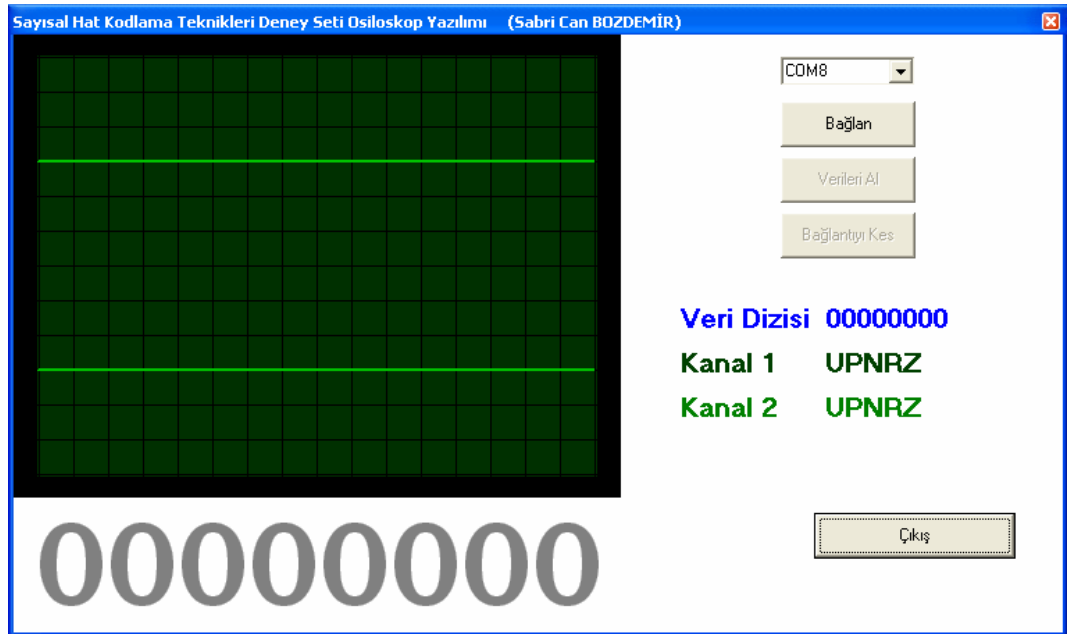
Deneyi bitirip ana menüye dönmek için $\uparrow$ tuşuna basılmalıdır.

4.1.2 PC’ye gönder

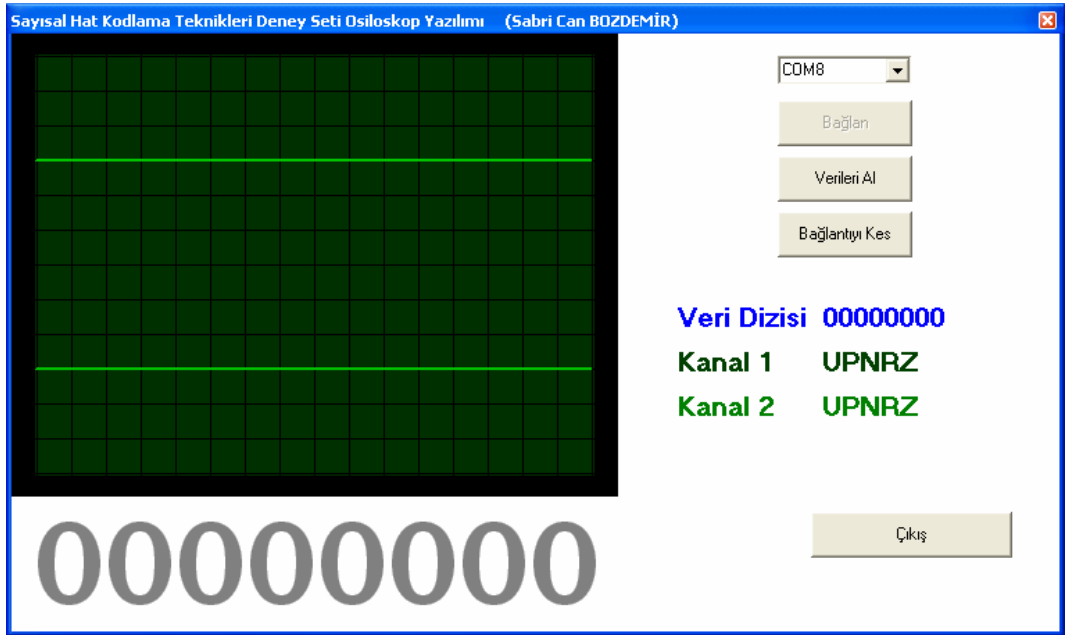
Deney setinden bilgisayara USB üzerinden bilgi göndermek için kullanılır.

Bunun için aşağıdaki adımlar izlenmelidir;

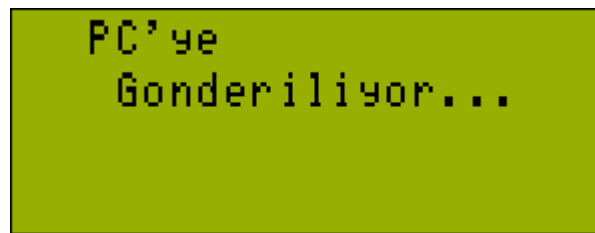
- Deney seti CD’sinde bulunan sürücü yazılımı bilgisayara yüklenir.
- Deney seti çalıştırılır.
- USB bağlantı kablosuyla deney seti ile bilgisayar bağlantısı yapılır.
- Deney seti CD’sinde bulunan osiloskop yazılımı çalıştırılır. Yazılım çalıştırıldığında aşağıdaki gibi ekranla karşılaşılır.



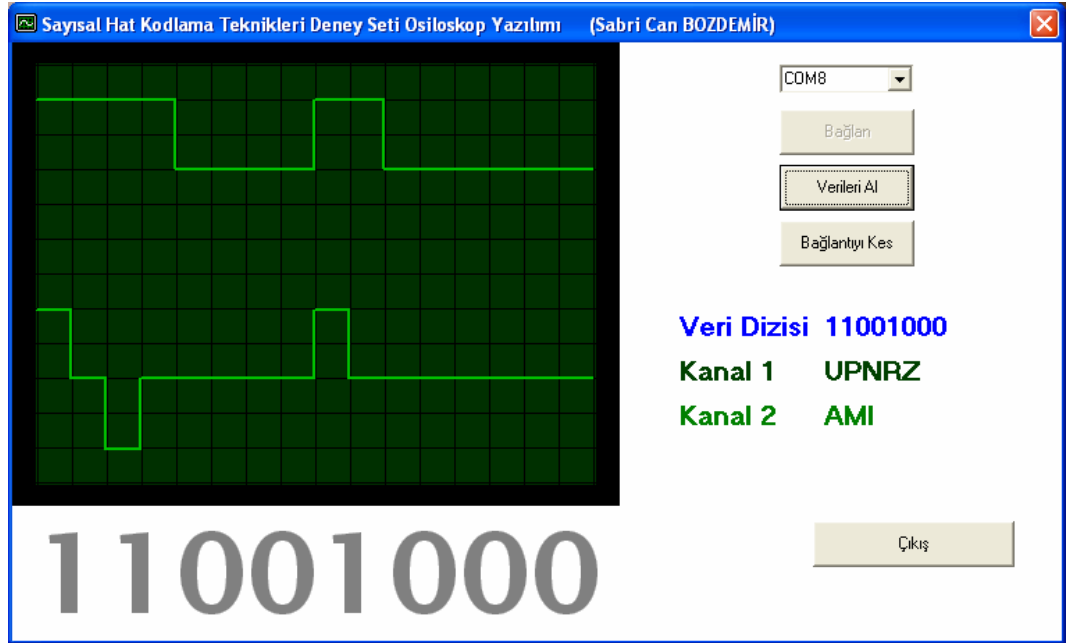
- Bu ekranda ilk olarak COM bağlantı noktası seçilir. Bu nokta varsayılan olarak COM8 portudur. Daha sonra bağlan butonuna tıklanır. Böylece bağlantı sağlanmış olur.



- Bu ekran görüldükten sonra deney setinde istenilen ayarlamalar yapıldıktan sonra ana menüden PC'ye gönderme işlemi başlatılır.



- Son olarak programda "Verileri Al" butonuna tıklanarak bilgilerin ekranda görüntülenmesi sağlanır.



Osiloskop yazılımı kapatmadan önce deney setiyle olan bağlantısını kesmek gerekir. Aksi takdirde deney setine bağlantılı kalınır. Ayrıca tekrar ana menüye dönmek için ↑ tuşuna basılmalıdır.

4.1.3 Ayarlar

Deneyle ilgili ayarlamaları yapmak için kullanılan menüdür. Ana menüde '2' tuşuna basınca karşımıza çıkar. Tekrar ana menüye dönmek için ↑ tuşuna basılmalıdır.

```
--- AYARLAR ---  
1 - İKİLİ DİZİ  
2 - KANALLAR  
3 - İLETİM HIZI
```

4.1.3.1 İkili Dizi

Gönderilmek istenen ikili bilginin girildiği bölümdür. Bu amaçla '1' ve '0' tuşları kullanılır. Herhangi bir düzeltme gerekirse ← sol ve → sağ ok tuşları kullanılarak işlem gerçekleştirilir.

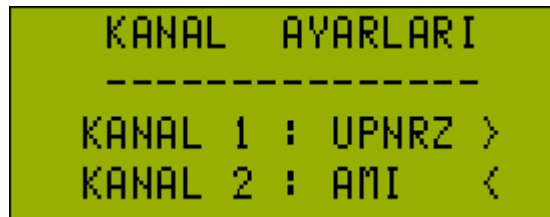


Deney seti 8 bitlik ikili bilgi girişine izin vermektedir.

İşlemi bitirip ayarlar menüsüne dönmek için ↑ tuşuna basılmalıdır.

4.1.3.2 Kanallar

Kanal 1 ve kanal 2'ye gönderilecek verinin sayısal hat kodlama tekniğini ayarlamak için kullanılan bölümdür.



Kanal 1'deki kodlama tekniğini ayarlamak için → sağ ok tuşu,

Kanal 1'deki kodlama tekniğini ayarlamak için ← sol ok tuşu,
kullanılmaktadır.

İşlemi bitirip ayarlar menüsüne dönmek için ↑ tuşuna basılmalıdır.

Seçim yapılabilecek 7 farklı sayısal hat kodlama tekniği vardır. Değerlerin anlamı sırayla aşağıdaki tabloda verilmiştir.

UPNRZ	Tek kutuplu sıfıra dönmeyen kodlama
BPNRZ	İki kutuplu sıfıra dönmeyen kodlama
UPRZ	Tek kutuplu sıfıra dönen kodlama
BPRZ	İki kutuplu sıfıra dönen kodlama
MANCH	Manchester (Split Phase) kodlama
AMI	Değişimli işaret tersleme (Alternate Mark Inversion) kodlama
CMI	Kurallı işaret tersleme (Coded Mark Inversion) kodlama

4.1.3.3 İletim hızı

Gönderilecek verinin iletim hızını (frekansını) ayarlamak için kullanılan bölümdür. ← sol ok tuşu ve → sağ ok tuşu yardımıyla istenilen değer seçilebilir.



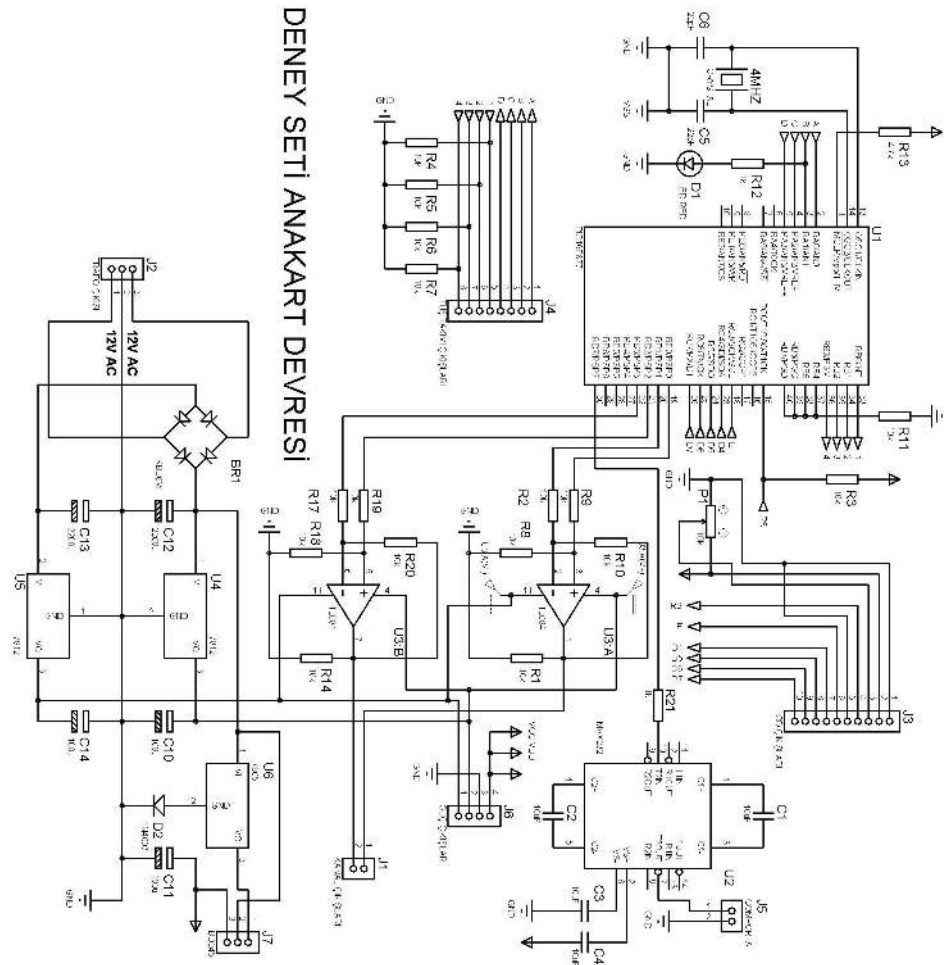
İşlemi bitirip ayarlar menüsüne dönmek için ↑ tuşuna basılmalıdır.

Ayarlanabilecek değerler aşağıdaki tabloda sırasıyla verilmiştir.

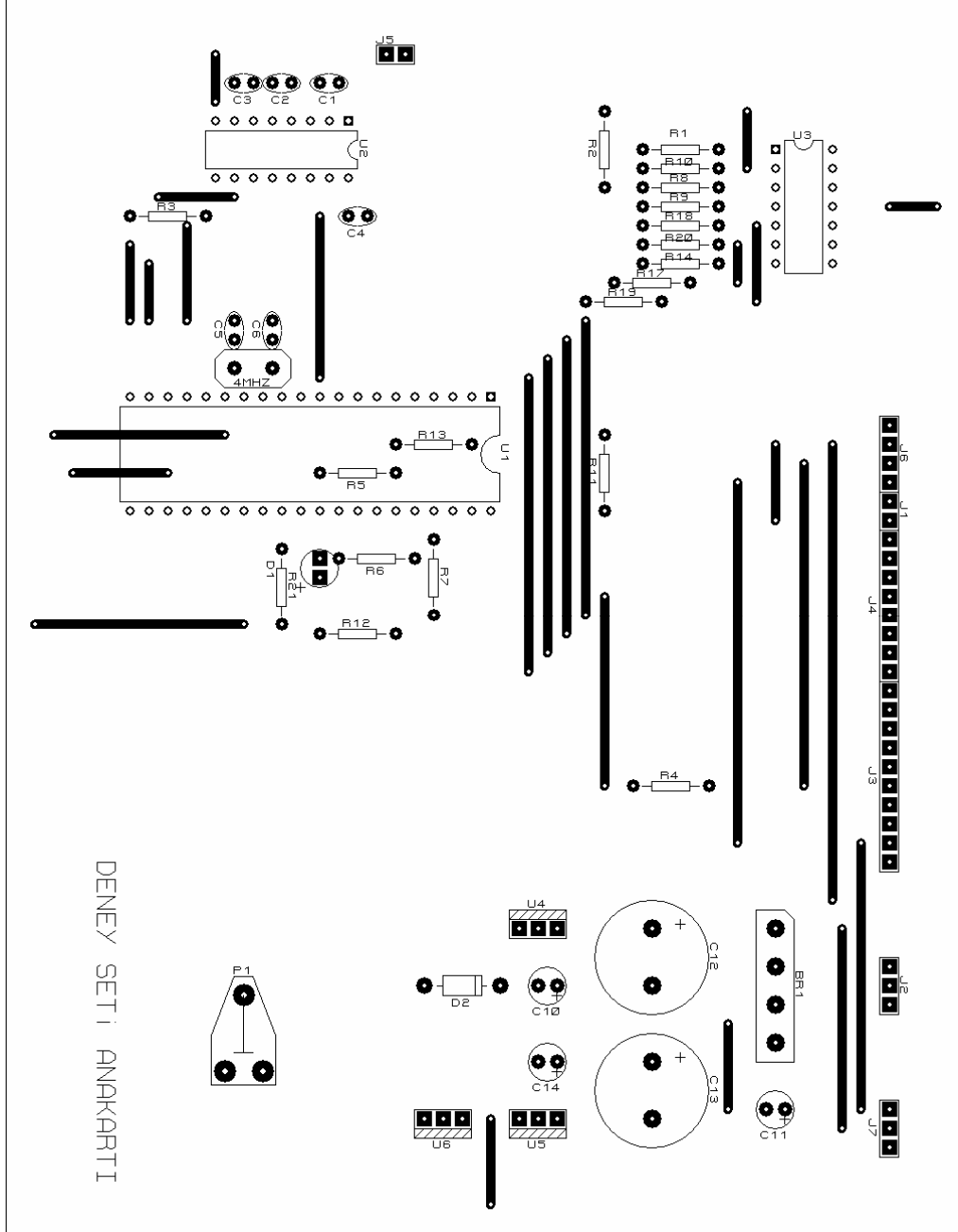
Frekanslar	100 Hz	300 Hz	600 Hz	1200 Hz	2400 Hz	4800 Hz	9600 Hz
-------------------	--------	--------	--------	---------	---------	---------	---------

5. ŞEMALAR VE BASKI DEVRELER

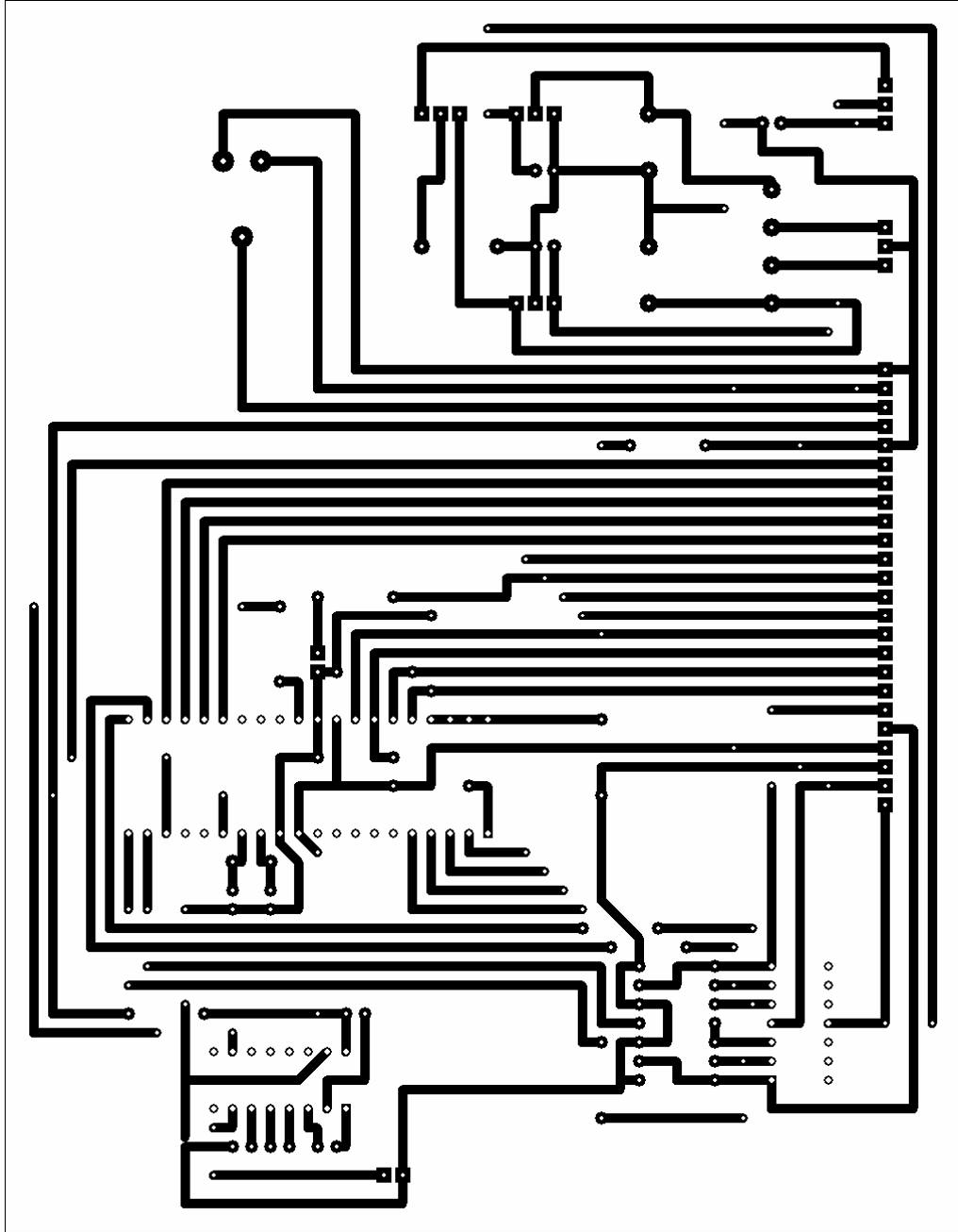
5.1. Deney Seti Anakart Devre Şeması



5.2. Deney Seti Anakart Baskı Devreleri



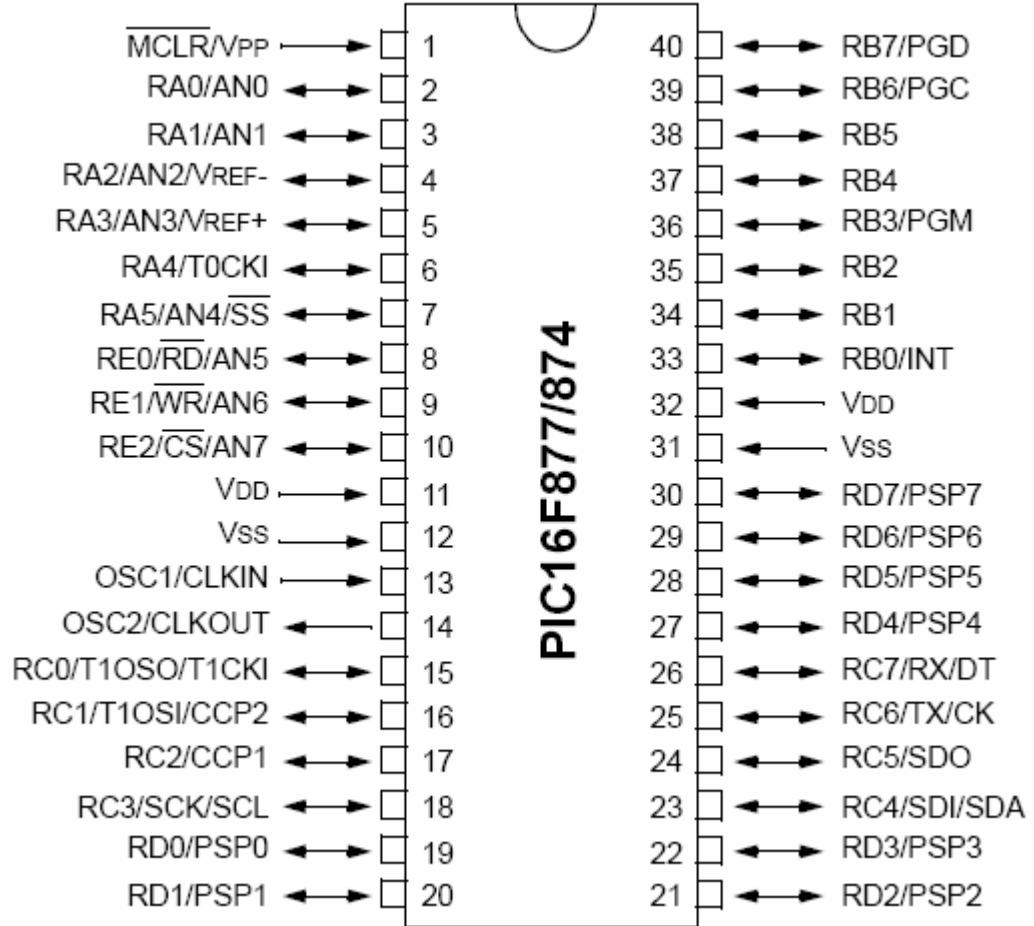
ÜSTTEN GÖRÜNÜŞ



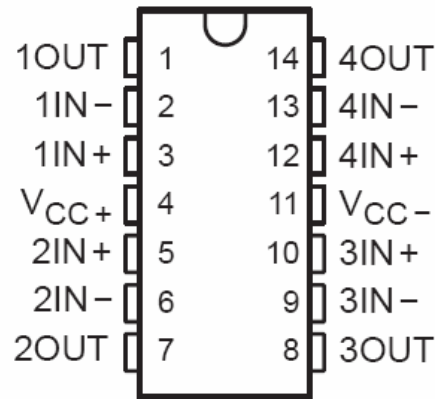
ALTTAN GÖRÜNÜŞ

6. DEVRE ELEMANLARI BACAK BAĞLANTILARI

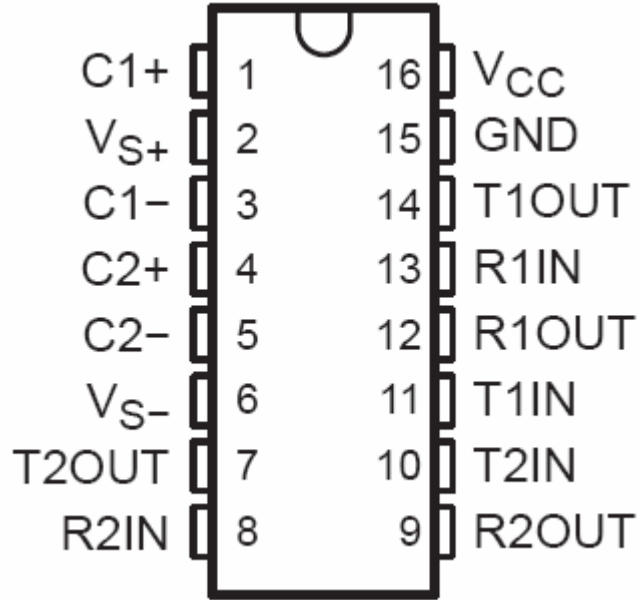
6.1. PIC16F877 Bacak Bağlantısı



6.2. TL084 Bacak Bağlantısı

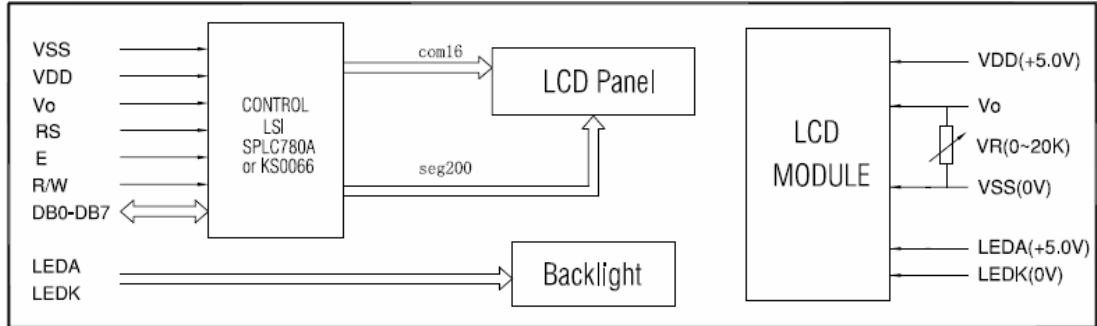


6.3. MAX232 Bacak Bağlantısı



6.4. 4x20 LCD HY-2004 Bilgileri

Blok diyagram ve güç bağlantıları



PIN Bağlantıları

Pin No	Symbol	Description
1	VSS	Ground for Logic
2	VDD	Power supply for Logic
3	V _o	Power supply for LCD drive
4	RS	Register selection (H:Data register, L:Instruction register)
5	R/W	Read/write selection (H:Read,L:Write)
6	E	Enable signal for LCM
7-14	DB0-DB7	Data Bus lines
15	LEDK	Power supply for Backlight(+)
16	LEDK	Power supply for Backlight(-)

7. PROGRAM KODLARI

```
'*****
'* İsim   : HAT KODU DENEY SETİ *
'* Yazar  : Sabri Can BOZDEMİR *
'* Tarih  : 30.06.2007 *
'*****

' Hat Kodları Çıkışı
'   PortD.0-1 = Kanal 1
'   PortD.2-3 = Kanal 2
'   PortD.7 = Seri Asenkron Çıkış

'Osilatör Frekansı
define OSC 4

include "modedefs.bas"

'Port Ayarları
TRISA = $00      'portA Çıkış
TRISB = $FF      'portB Giriş
TRISD = $00      'portD çıkış Hat Kodları Çıkışı

'LCD Tanımlamaları
define LCD_DREG PORTC      'LCD data portu
define LCD_DBIT 4          '4 bitlik veri yolunun başlangıcı
define LCD_RSREG PORTC     'LCD Register seçme portu
define LCD_RSBIT 0         'LCD Register seçme biti
define LCD_EREG PORTC      'LCD enable seçme portu
define LCD_EBIT 3          'LCD enable seçme biti
define LCD_BITS 4          '4 bitlik veri yolu
define LCD_LINES 4         'LCD Satır sayısı

'Değişkenler
A      VAR BYTE
C      VAR BYTE
I      VAR BYTE
J      VAR BYTE
K      VAR BYTE
L      VAR BYTE
Kanal VAR BYTE[1]
Tus    VAR BYTE
GonVeri1 var byte[16]
GonVeri2 var byte[16]
```

```
Gecikme var word
VeriHizi var byte
VeriDizisi var bit[8]
CMI var byte
```

```
'Semboller
```

```
'ilk deęerler
verihizi = 3
KANAL[0] = 0
KANAL[2] = 1
```

```
'250ms BEKLE
pause 250
```

```
AnaMenu:
portd = 0
```

```
lcdout $FE, 1
lcdout $FE, $82
lcdout "--- ANA MENU ---"
lcdout $FE, $C2
lcdout "1 - DENEYİ BASLAT"
lcdout $FE, $96
lcdout "2 - PC'YE GONDER"
lcdout $FE, $D6
lcdout "3 - AYARLAR"
```

```
Kestirme3:
pause 150
GOSUB TUSKONTROL
```

```
if tus = 1 then goto Deneyebasla
if tus = 2 then goto PCgonder
if tus = 3 then goto Ayarlar
```

```
goto Kestirme3
```

```
PCgonder:
lcdout $FE, 1
lcdout $FE, $82
lcdout "PC'ye"
lcdout $FE, $C3
```

```
lcdout "Gonderiliyor..."
```

```
PORTA = $0F
```

```
kestirme6:
```

```
if PortB.3 = 1 then goto anamenu
```

```
serout portd.7, t2400, ["HIZ_",#verihizi,13]
```

```
serout portd.7, t2400, ["KANAL1_",#kanal[0],13]
```

```
serout portd.7, t2400, ["KANAL2_",#kanal[2],13]
```

```
serout portd.7, t2400, ["VeriDizisi_"]
```

```
For k = 0 to 7
```

```
serout portd.7, t2400, [#veridizisi[k]]
```

```
next k
```

```
serout portd.7, t2400, [13]
```

```
pause 10
```

```
goto kestirme6
```

```
DeneyeBasla:
```

```
lcdout $FE, 1
```

```
lcdout $FE, $85
```

```
lcdout "ikiLi Dizi"
```

```
lcdout $FE, $C2
```

```
for k=0 to 7
```

```
    lcdout " ",#veridizisi[k]
```

```
next k
```

```
lcdout $FE, $95
```

```
lcdout "K1-"
```

```
a = kanal[0]
```

```
Gosub koduA1
```

```
lcdout $FE, $9F
```

```
lcdout "K2-"
```

```
A = kanal[2]
```

```
Gosub kodual
```

```
lcdout $FE, $D5
```

```
lcdout "HIZ : "
```

```
Gosub hizial
```

```
if verihizi = 0 then Gecikme = 4973 ' 100 Hz
```

```
if verihizi = 1 then Gecikme = 1640 ' 300 Hz
```

```
if verihizi = 2 then Gecikme = 390 ' 1200 Hz
```

```
if verihizi = 3 then Gecikme = 181 ' 2400 Hz
```

```
if verihizi = 4 then Gecikme = 77 ' 4800 Hz
```

```
if verihizi = 5 then Gecikme = 25 ' 9600 Hz
```

```
'gonveri girilecek yer....
```

```
A = 1
```

```
tus = 1
```

```
cmI = 2
```

```
for k = 0 to 7
```

```
    I = K * 2
```

```
    J = I + 1
```

```
'kanal1
```

```
    'UPNRZ
```

```
    if kanal[0] = 0 then
```

```
        gonveril[I] = veridizisi[k]
```

```
        gonveril[J] = gonveril[I]
```

```
    endif
```

```
    'BPNRZ
```

```
    if kanal[0] = 1 then
```

```
        gonveril[I] = 2 - veridizisi[k]
```

```
        gonveril[J] = gonveril[I]
```

```
    endif
```

```
    'UPRZ
```

```
    if kanal[0] = 2 then
```

```
        gonveril[I] = veridizisi[k]
```

```
        gonveril[J] = 0
```

```
    endif
```

```
    'BPRZ
```

```
    if kanal[0] = 3 then
```

```
        gonveril[I] = 2 - veridizisi[k]
```

```
        gonveril[J] = 0
```

```
    endif
```

```
    'Manch
```

```
    if kanal[0] = 4 then
```

```
        gonveril[I] = 2 - veridizisi[k]
```

```

        gonver1[J] = veridizisi[k] + 1
    endif
    'AMI
    if kanal[0] = 5 then
        if veridizisi[k] = 1 then
            gonver1[I] = a
            gonver1[J] = 0
            a = 3 - a
        endif
        if veridizisi[k] = 0 then
            gonver1[I] = 0
            gonver1[J] = 0
        endif
    endif
    'CMI
    if kanal[0] = 6 then
        if veridizisi[k] = 1 then
            gonver1[I] = cmi
            gonver1[J] = gonver1[I]
            CMI = 3 - CMI
        endif
        if veridizisi[k] = 0 then
            gonver1[I] = 2
            gonver1[J] = 1
        endif
    endif
    'kanal2
    'UPNRZ
    if kanal[2] = 0 then
        gonver2[I] = veridizisi[k]
        gonver2[J] = gonver2[I]
    endif
    'BPNRZ
    if kanal[2] = 1 then
        gonver2[I] = 2 - veridizisi[k]
        gonver2[J] = gonver2[I]
    endif
    'UPRZ
    if kanal[2] = 2 then
        gonver2[I] = veridizisi[k]
        gonver2[J] = 0
    endif
    'BPRZ

```

```

if kanal[2] = 3 then
    gonveri2[I] = 2 - veridizisi[k]
    gonveri2[J] = 0
endif
'Manch
if kanal[2] = 4 then
    gonveri2[I] = 2 - veridizisi[k]
    gonveri2[J] = veridizisi[k] + 1
endif
'AMI
if kanal[2] = 5 then
    if veridizisi[k] = 1 then
        gonveri2[I] = tus
        gonveri2[J] = 0
        tus = 3 - tus
    endif
    if veridizisi[k] = 0 then
        gonveri2[I] = 0
        gonveri2[J] = 0
    endif
endif
'CMİ
if kanal[2] = 6 then
    if veridizisi[k] = 1 then
        gonveri2[I] = cmi
        gonveri2[J] = gonveri2[I]
        CMI = 3 - CMI
    endif
    if veridizisi[k] = 0 then
        gonveri2[I] = 2
        gonveri2[J] = 1
    endif
endif

gonveri2[I] = gonveri2[I] << 2
gonveri2[J] = gonveri2[J] << 2

```

next k

PORTA = \$0F

Kestirme5:

for k = 0 to 15

'27 mikro sn gecikme

```
PortD = gonveri1[K] + gonveri2[K]
pauseus gecikme
if PortB.3 = 1 then goto anamenu
next k

goto Kestirme5
```

Ayarlar:

```
lcdout $FE, 1
lcdout $FE, $83
lcdout "--- AYARLAR ---"
lcdout $FE, $C3
lcdout "1 - İKİLİ DİZİ"
lcdout $FE, $97
lcdout "2 - KANALLAR"
lcdout $FE, $D7
lcdout "3 - İLETİM HIZI"
```

```
Kestirme1:
pause 200
GOSUB TUSKONTROL
```

```
if tus = 16 then Goto anamenu
if tus = 1 then goto İkiliDizi
if tus = 2 then goto Kanallar
if tus = 3 then goto iletimhizi
```

```
goto Kestirme1
```

İkiliDizi:

```
lcdout $FE, 1
lcdout $FE, $C1
lcdout "VERİ GİR : "
lcdout $FE, $0E
lcdout $FE, $Cc
for k = 0 to 7
    lcdout #VeriDizisi[K]
next k
```

```
K = 0
Don:
```

```

lcdout $FE, $Cc + k
pause 200
GOSUB TUSKONTROL

if tus = 1 then
    lcdout "1"
    VeriDizisi[K] = 1
    k = k + 1
    if k = 8 then K = 7
endif

if tus = 14 then
    lcdout "0"
    VeriDizisi[K] = 0
    k = k + 1
    if k = 8 then K = 7
endif

if tus = 8 then
    k = k + 1
    if k = 8 then K = 7
endif

if tus = 4 then
    if k = 0 then K = 1
    k = k - 1
endif

if tus = 16 then
    LCDOUT $FE, $0C
    Goto ayarlar
endif

goto Don

kanallar:

lcdout $FE, 1
lcdout $FE, $83
lcdout "KANAL  AYARLARI"
lcdout $FE, $C3
lcdout "-----"

```

```
lcdout $FE, $96
lcdout "KANAL 1 :      >"
lcdout $FE, $D6
lcdout "KANAL 2 :      <"
```

```
Kestirme2:
lcdout $FE, $A0
A = kanal[0]
Gosub KoduA1
```

```
lcdout $FE, $E0
A = kanal[2]
Gosub KoduA1
```

```
pause 200
GOSUB TUSKONTROL
```

```
if tus = 16 then Goto ayarlar
if tus = 8 then
    kanal[0] = kanal[0] + 1
    if kanal[0] = 8 then kanal[0] = 0
endif
if tus = 4 then
    kanal[2] = kanal[2] + 1
    if kanal[2] = 8 then kanal[2] = 0
endif
```

```
goto Kestirme2
```

```
iletimHizi:
```

```
lcdout $FE, 1
lcdout $FE, $C3
lcdout "- İLETİM HIZI -"
lcdout $FE, $96
lcdout " <          >"
```

```
Kestirme4:
lcdout $FE, $9B
Gosub hizial
```

```
pause 200
GOSUB TUSKONTROL
```

```

if tus = 16 then Goto ayarlar
if tus = 8 then
    verihizi = verihizi + 1
    if verihizi = 6 then verihizi = 0
endif
if tus = 4 then
    if verihizi = 0 then verihizi = 6
    verihizi = verihizi - 1
endif

```

goto Kestirme4

TusKontrol:

PORTA = \$0F

```

while PORTB = 0
wend
A = 1
PORTA = A
for I = 0 TO 3
    C = 1
    for J = 1 TO 4
        IF PORTB = C THEN TUS = (I * 4) + J
        C = C * 2
    NEXT J
    A = A * 2
    PORTA = A
NEXT I

```

Return

KoduA1:

```

if A = 0 then lcdout "UPNRZ"
if A = 1 then lcdout "BPNRZ"
if A = 2 then lcdout "UPRZ "
if A = 3 then lcdout "BPRZ "
if A = 4 then lcdout "MANCH"
if A = 5 then lcdout "AMI  "
if A = 6 then lcdout "CMI  "

```

Return

HiziA1:

```
    if verihizi = 0 then lcdout " 100 Hz"
    if verihizi = 1 then lcdout " 300 Hz"
    if verihizi = 2 then lcdout "1200 Hz"
    if verihizi = 3 then lcdout "2400 Hz"
    if verihizi = 4 then lcdout "4800 Hz"
    if verihizi = 5 then lcdout "9600 Hz"
Return
```

8. ELEMAN VE FİYAT LİSTESİ

DENEY SETİ ELEMAN VE FİYAT LİSTESİ (Mayıs 2007)

Eleman Cinsi	Elemanın Adı	Birim Fiyat (YTL)	Adet	Toplam Fiyat (YTL)
Entegre	PIC16F877	15	1	15
	MAX232	6	1	6
	TL084	3	1	3
	LM7805	2	1	2
	LM7812	2	1	2
	LM7912	2	1	2
Modül	Seri - USB Çevirici	20	1	20
Diyot	Kırmızı LED	0,5	4	2
	Yeşil LED	0,5	3	1,5
	Turuncu LED	0,75	1	0,75
	1N4001	0,25	1	0,25
Kristal	4 Mhz	2	1	2
Kondansatör	2200 µF	1,5	2	3
	100 µF	0,2	3	0,6
	10 µF	0,15	4	0,6
	22 pF	0,1	2	0,2
Direnç	1 K	0,05	4	0,2
	4,7 K	0,05	1	0,05
	470 Ω	0,05	3	0,15
	10 K	0,05	12	0,6
	10 K Trimpot	0,5	1	0,5
LCD	4x20 LCD HY-2004A	27	1	27
Transformatör	2x12 Volt 15 Watt	15	1	15
Tuş Takımı	4x4 Standart	10	1	10
Diğer	40 Bacaklı Entegre Soketi	0,75	2	1,5
	16 Bacaklı Entegre Soketi	0,25	1	0,25
	18 Bacaklı Entegre Soketi	0,25	1	0,25
	500 mA Sigorta	0,5	1	0,5
	Sigorta Yuvası	1,5	1	1,5
	Güç Kablosu	3	1	3
	Güç Girişi Yuvası	2	1	2
	Güç Anahtarı	1,5	1	1,5
	RS232 Konnektörü	1	1	1
	USB Kablosu	2	1	2
	Bağlantı Soketleri	2	1	2
	Bağlantı Kabloları	1	1	1
	Bakırlı Plaket	1	1	1
	LED Yuvası	0,25	7	1,75
	Osilaskop probu yuvası	0,75	2	1,5
	Bağlantı Yuvaları	0,5	7	3,5
	Bağlantı Vidaları	0,2	10	2
	Dış kalıp	10	1	10
GENEL TOPLAM				150,65 YTL

9. DENEYLER

DENEY 1

UPNRZ ve BPNRZ karşılaştırması

AMAÇ

Sayısal hat kodlarından olan UPNRZ ve BPNRZ işaretlerinin karşılaştırmasını yapmak

GEREKLİ MALZEMELER

- Deney Seti
- İki adet osiloskop probu
- Osiloskop
- Voltmetre (veya Avometre)

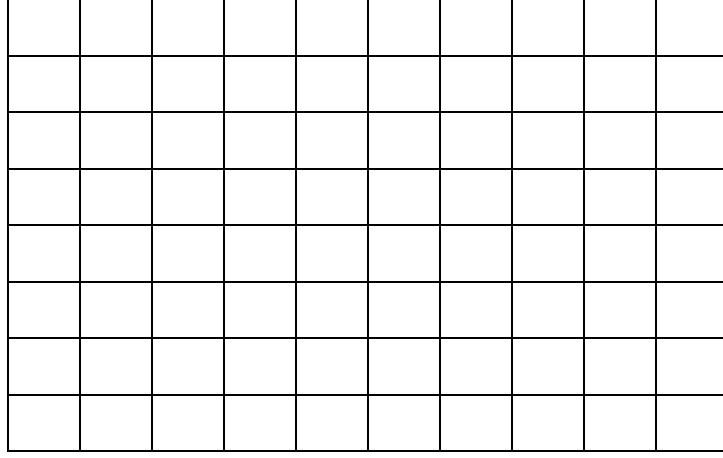
GİRİŞ

Tek kutuplu sıfıra dönmeyen (Unipolar Non-Return to Zero) kodlama, kodlama çeşitleri içinde en basit ve en ucuz yöntemdir. Sadece bir gerilim seviyesini kullanır. İkili 1 seviyesi belirli bir yüksek gerilim seviyesine, ikili 0 ise 0 volt seviyesine karşılık gelir.

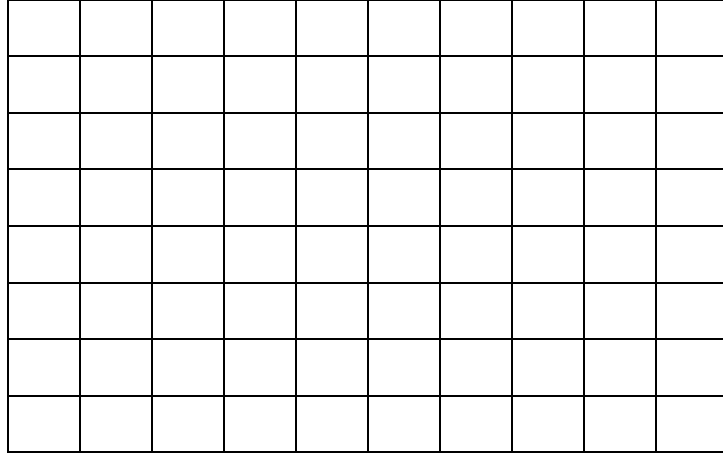
İki kutuplu sıfıra dönmeyen (Bipolar Non-Return to Zero) kodlama tekniğinde her bir bit belli bir düzeydeki işaret ile gösterilir. Örneğin ikili 1 yüksek gerilim düzeyi ile, ikili 0 ise düşük gerilim düzeyi ile gösterilir. İkili veri ard arda gönderilirken sürekli olarak düşük ve yüksek gerilim düzeyleri arasında geçiş olur. Hiçbir zaman sıfır gerilim düzeyi kullanılmaz

İŞLEM BASAMAKLARI

1. Deney setinin aç/kapa anahtarını açınız.
2. Ana menüden ayarlar menüsüne geçmek için tuş takımından '3' tuşuna basınız.
3. Ayarlar menüsünden ikili dizi menüsüne geçmek için tuş takımından '1' tuşuna basınız. Burada kanallardan göndereceğiniz 8 bitlik ikili veriyi '1' ve '0' tuşlarına basarak giriniz (Örneğin: "10001010"). Herhangi bir yanlışlığı düzeltmek için sağ (→) ve sol (←) ok tuşlarını kullanınız. Bilgileri kaydedip ayarlar menüsüne dönmek için yukarı ok (↑) tuşuna basınız.
4. Ayarlar menüsünden kanallar menüsüne geçmek için tuş takımından '2' tuşuna basınız. Burada kanal 1 için sağ (→) ok tuşunu kullanarak UPNRZ ve kanal 2 için sol (←) ok tuşunu kullanarak BPNRZ olarak ayarlayınız. Bilgileri kaydedip ayarlar menüsüne dönmek için yukarı ok (↑) tuşuna basınız.
5. Ayarlar menüsünden iletim hızı menüsüne geçmek için tuş takımından '3' tuşuna basınız. Sağ (→) ve sol (←) ok tuşlarını kullanarak uygun bir iletim hızı seçiniz (Örneğin: 1200 Hz). Bilgileri kaydedip ayarlar menüsüne dönmek için yukarı ok (↑) tuşuna basınız.
6. Ana menüye dönmek için tekrar yukarı ok (↑) tuşuna basınız.
7. Osiloskobunuzu açıp her iki kanal birlikte (altlı üstü) görünecek şekilde ayarlayınız. Her iki kanalın volt/div ayarını 5 V yapınız ve DC kademesine getiriniz.
8. Osiloskop probalarını osiloskop ve deney seti üzerindeki kanal 1 ve kanal 2 bağlantı noktalarına yerleştiriniz.
9. Ana menüde '1' tuşuna basarak deneyi başlatınız.
10. Osiloskop ekranında gördüğünüz şekli çiziniz.
11. Osiloskobu X-Y konumuna alınız. Osiloskop ekranında gördüğünüz şekli çiziniz.



Osiloskop İşaretleri



X-Y Düzlemi

SORULAR

1. UPNRZ ve BPNRZ işaretleri arasındaki farklar nelerdir?
2. Osiloskobun X-Y düzleminde gördüğünüz şekli nasıl yorumluyorsunuz?
3. Voltmetrenin DC kademesini kullanarak Kanal 1 ve Kanal 2'deki gerilimleri ölçünüz. Yorumunuz nedir?

$V(dc)_{\text{kanal1}} = \dots\dots\dots$

$V(dc)_{\text{kanal2}} = \dots\dots\dots$

4. Voltmetrenin AC kademesini kullanarak Kanal 1 ve Kanal 2'deki gerilimleri ölçünüz. Yorumunuz nedir?

$V(ac)_{\text{kanal1}} = \dots\dots\dots$

$V(ac)_{\text{kanal2}} = \dots\dots\dots$

CEVAPLAR

DENEY 2

UPNRZ ve UPRZ karşılaştırması

AMAÇ

Sayısal hat kodlarından olan UPNRZ ve UPRZ işaretlerinin karşılaştırmasını yapmak

GEREKLİ MALZEMELER

- Deney Seti
- İki adet osiloskop probu
- Osiloskop
- Voltmetre (veya Avometre)

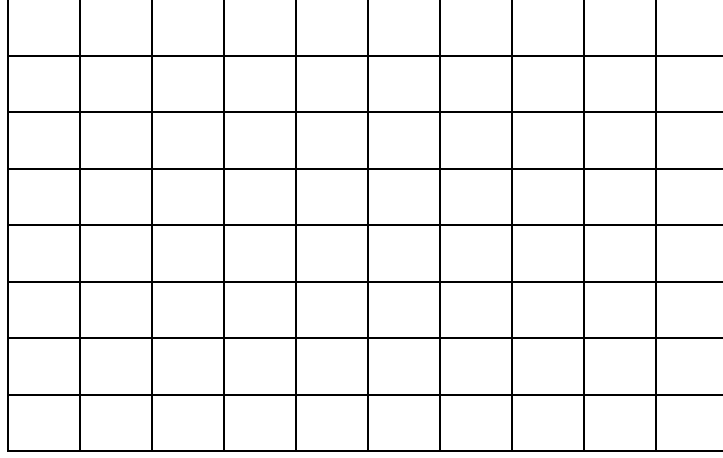
GİRİŞ

Tek kutuplu sıfıra dönmeyen (Unipolar Non-Return to Zero) kodlama, kodlama çeşitleri içinde en basit ve en ucuz yöntemdir. Sadece bir gerilim seviyesini kullanır. İkili 1 seviyesi belirli bir yüksek gerilim seviyesine, ikili 0 ise 0 volt seviyesine karşılık gelir.

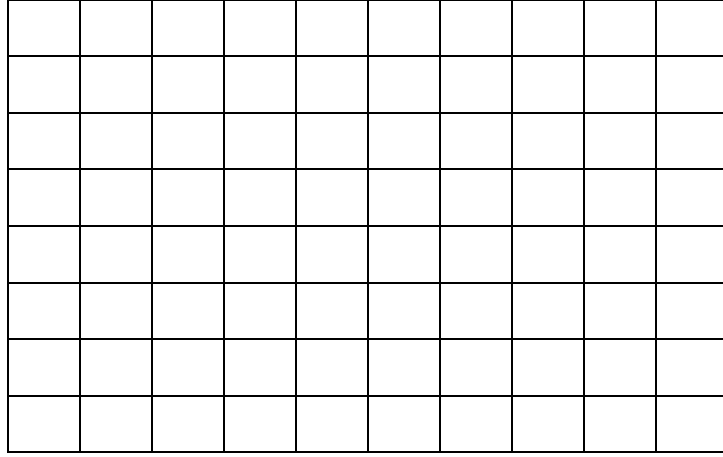
Tek kutuplu sıfıra dönen (Unipolar Return to Zero) kodlama tekniğinde ikili 1 gerilim düzeyinin yükselmesi ve daha sonra sıfıra geri dönmesi ile gösterilir. İkili 0 ise herhangi bir değişikliğin olmaması ile gösterilir. İkili 1'in gösteriminde bit başlangıcında gerilim seviyesi artar, bitin gönderim aralığının yarısına kadar yüksek düzeyde kalır. Tam yarısında 0 düzeyine düşer ve 0 düzeyinde kalır.

İŞLEM BASAMAKLARI

1. Deney setinin aç/kapa anahtarını açınız.
2. Ana menüden ayarlar menüsüne geçmek için tuş takımından '3' tuşuna basınız.
3. Ayarlar menüsünden ikili dizi menüsüne geçmek için tuş takımından '1' tuşuna basınız. Burada kanallardan göndereceğiniz 8 bitlik ikili veriyi '1' ve '0' tuşlarına basarak giriniz (Örneğin: "10001010"). Herhangi bir yanlışlığı düzeltmek için sağ (→) ve sol (←) ok tuşlarını kullanınız. Bilgileri kaydedip ayarlar menüsüne dönmek için yukarı ok (↑) tuşuna basınız.
4. Ayarlar menüsünden kanallar menüsüne geçmek için tuş takımından '2' tuşuna basınız. Burada kanal 1 için sağ (→) ok tuşunu kullanarak UPNRZ ve kanal 2 için sol (←) ok tuşunu kullanarak UPRZ olarak ayarlayınız. Bilgileri kaydedip ayarlar menüsüne dönmek için yukarı ok (↑) tuşuna basınız.
5. Ayarlar menüsünden iletim hızı menüsüne geçmek için tuş takımından '3' tuşuna basınız. Sağ (→) ve sol (←) ok tuşlarını kullanarak uygun bir iletim hızı seçiniz (Örneğin: 1200 Hz). Bilgileri kaydedip ayarlar menüsüne dönmek için yukarı ok (↑) tuşuna basınız.
6. Ana menüye dönmek için tekrar yukarı ok (↑) tuşuna basınız.
7. Osiloskobunuzu açıp her iki kanal birlikte (altlı üstü) görünecek şekilde ayarlayınız. Her iki kanalın volt/div ayarını 5 V yapınız ve DC kademesine getiriniz.
8. Osiloskop probalarını osiloskop ve deney seti üzerindeki kanal 1 ve kanal 2 bağlantı noktalarına yerleştiriniz.
9. Ana menüde '1' tuşuna basarak deneyi başlatınız.
10. Osiloskop ekranında gördüğünüz şekli çiziniz.
11. Osiloskobu X-Y konumuna alınız. Osiloskop ekranında gördüğünüz şekli çiziniz.



Osiloskop İşaretleri



X-Y Düzlemi

SORULAR

1. UPNRZ ve UPRZ işaretleri arasındaki farklar nelerdir?
2. Osiloskobun X-Y düzleminde gördüğünüz şekli nasıl yorumluyorsunuz?
3. Voltmetrenin DC kademesini kullanarak Kanal 1 ve Kanal 2'deki gerilimleri ölçünüz. Yorumunuz nedir?

$V(dc)_{\text{kanal1}} = \dots\dots\dots$

$V(dc)_{\text{kanal2}} = \dots\dots\dots$

4. Voltmetrenin AC kademesini kullanarak Kanal 1 ve Kanal 2'deki gerilimleri ölçünüz. Yorumunuz nedir?

$V(ac)_{\text{kanal1}} = \dots\dots\dots$

$V(ac)_{\text{kanal2}} = \dots\dots\dots$

CEVAPLAR

DENEY 3

UPNRZ ve BPRZ karşılaştırması

AMAÇ

Sayısal hat kodlarından olan UPNRZ ve UPRZ işaretlerinin karşılaştırmasını yapmak

GEREKLİ MALZEMELER

- Deney Seti
- İki adet osiloskop probu
- Osiloskop
- Voltmetre (veya Avometre)

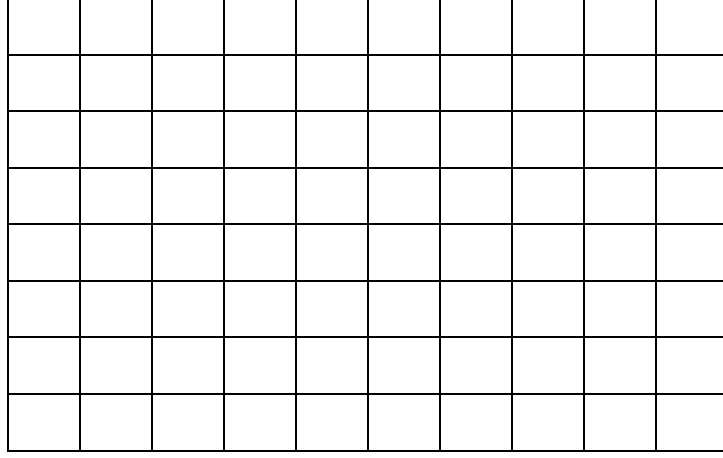
GİRİŞ

Tek kutuplu sıfıra dönmeyen (Unipolar Non-Return to Zero) kodlama, kodlama çeşitleri içinde en basit ve en ucuz yöntemdir. Sadece bir gerilim seviyesini kullanır. İkili 1 seviyesi belirli bir yüksek gerilim seviyesine, ikili 0 ise 0 volt seviyesine karşılık gelir.

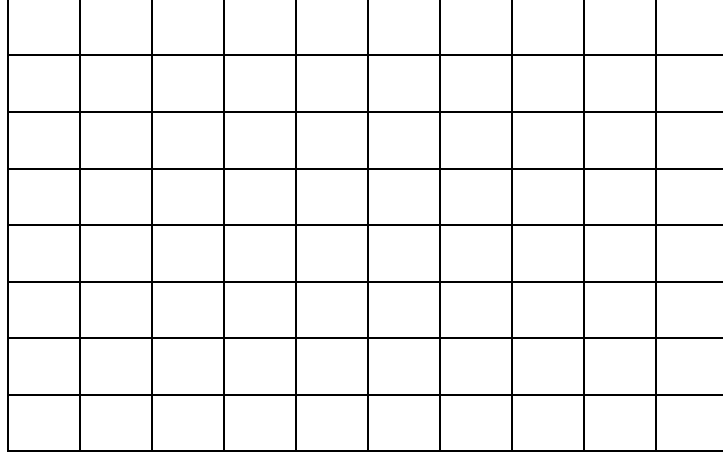
İki kutuplu sıfıra dönen (Bipolar Return to Zero) kodlama tekniğinde her bit yüksek ya da düşük gerilim düzeyinden 0 düzeyine dönmesi ile kodlanır. İkili 1, bitin başlangıcında gerilim düzeyinin yükselmesi, bit gönderim genişliğinin tam ortasına gelindiğinde ise tekrar sıfıra düşerek bitin sonuna kadar 0 düzeyinde kalması ile kodlanır. İkili 0 ise aynı şekilde bitin gönderilmeye başlanması ile gerilim düzeyinin düşmesi, bit genişliğinin tam ortasında ise tekrar sıfıra yükselmesi ile kodlanır. Her bir bitin kodlanmasında sıfır düzeyine gelinmesi, oluşan işaretin dönüş genişliğinin yarıya düşmesine neden olur.

İŞLEM BASAMAKLARI

1. Deney setinin aç/kapa anahtarını açınız.
2. Ana menüden ayarlar menüsüne geçmek için tuş takımından '3' tuşuna basınız.
3. Ayarlar menüsünden ikili dizi menüsüne geçmek için tuş takımından '1' tuşuna basınız. Burada kanallardan göndereceğiniz 8 bitlik ikili veriyi '1' ve '0' tuşlarına basarak giriniz (Örneğin: "10001010"). Herhangi bir yanlışlığı düzeltmek için sağ (→) ve sol (←) ok tuşlarını kullanınız. Bilgileri kaydedip ayarlar menüsüne dönmek için yukarı ok (↑) tuşuna basınız.
4. Ayarlar menüsünden kanallar menüsüne geçmek için tuş takımından '2' tuşuna basınız. Burada kanal 1 için sağ (→) ok tuşunu kullanarak UPNRZ ve kanal 2 için sol (←) ok tuşunu kullanarak BPRZ olarak ayarlayınız. Bilgileri kaydedip ayarlar menüsüne dönmek için yukarı ok (↑) tuşuna basınız.
5. Ayarlar menüsünden iletim hızı menüsüne geçmek için tuş takımından '3' tuşuna basınız. Sağ (→) ve sol (←) ok tuşlarını kullanarak uygun bir iletim hızı seçiniz (Örneğin: 1200 Hz). Bilgileri kaydedip ayarlar menüsüne dönmek için yukarı ok (↑) tuşuna basınız.
6. Ana menüye dönmek için tekrar yukarı ok (↑) tuşuna basınız.
7. Osiloskobunuzu açıp her iki kanal birlikte (altlı üstü) görünecek şekilde ayarlayınız. Her iki kanalın volt/div ayarını 5 V yapınız ve DC kademesine getiriniz.
8. Osiloskop probalarını osiloskop ve deney seti üzerindeki kanal 1 ve kanal 2 bağlantı noktalarına yerleştiriniz.
9. Ana menüde '1' tuşuna basarak deneyi başlatınız.
10. Osiloskop ekranında gördüğünüz şekli çiziniz.
11. Osiloskobu X-Y konumuna alınız. Osiloskop ekranında gördüğünüz şekli çiziniz.



Osiloskop İşaretleri



X-Y Düzlemi

SORULAR

1. UPNRZ ve BPRZ işaretleri arasındaki farklar nelerdir?
2. Osiloskobun X-Y düzleminde gördüğünüz şekli nasıl yorumluyorsunuz?
3. Voltmetrenin DC kademesini kullanarak Kanal 1 ve Kanal 2'deki gerilimleri ölçünüz. Yorumunuz nedir?

$V(\text{dc})_{\text{kanal1}} = \dots\dots\dots$

$V(\text{dc})_{\text{kanal2}} = \dots\dots\dots$

4. Voltmetrenin AC kademesini kullanarak Kanal 1 ve Kanal 2'deki gerilimleri ölçünüz. Yorumunuz nedir?

$V(\text{ac})_{\text{kanal1}} = \dots\dots\dots$

$V(\text{ac})_{\text{kanal2}} = \dots\dots\dots$

CEVAPLAR

DENEY 4

UPNRZ ve Manchester karşılaştırması

AMAÇ

Sayısal hat kodlarından olan UPNRZ ve Manchester işaretlerinin karşılaştırmasını yapmak

GEREKLİ MALZEMELER

- Deney Seti
- İki adet osiloskop probu
- Osiloskop
- Voltmetre (veya Avometre)

GİRİŞ

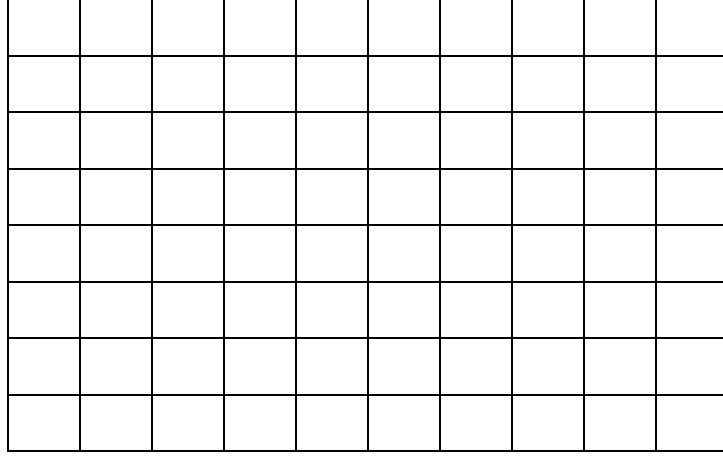
Tek kutuplu sıfıra dönmeyen (Unipolar Non-Return to Zero) kodlama, kodlama çeşitleri içinde en basit ve en ucuz yöntemdir. Sadece bir gerilim seviyesini kullanır. İkili 1 seviyesi belirli bir yüksek gerilim seviyesine, ikili 0 ise 0 volt seviyesine karşılık gelir.

Faz (Phase) kodlaması olarak ta bilinen Manchester kodlamasında bitler iki ayrı gerilim düzeyi arasında bir geçiş oluşturularak kodlanır. Bu nedenle sıfır düzeyine gerek yoktur. Pozitif ve negatif genlik düzeyi yeterlidir. Manchester kodlamada, bir bit süresi ikiye bölünür. Her bir yarıda gönderilmek istenen bitin 1 veya 0 olması durumuna göre işaret gözlenir.

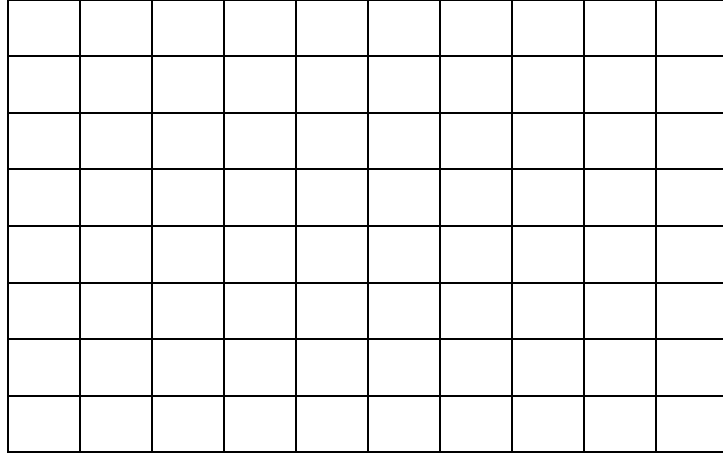
Manchester kodlamada her zaman bölmesi yarıya bölünür. Örneğin ikili 1, zaman bölmesinin yarısında 1, diğer yarısında 0 olarak kodlanır. İkili 0 ise, tam tersidir. Yani ilk yarıda 0, diğer yarıda 1 kodlanır.

İŞLEM BASAMAKLARI

1. Deney setinin aç/kapa anahtarını açınız.
2. Ana menüden ayarlar menüsüne geçmek için tuş takımından '3' tuşuna basınız.
3. Ayarlar menüsünden ikili dizi menüsüne geçmek için tuş takımından '1' tuşuna basınız. Burada kanallardan göndereceğiniz 8 bitlik ikili veriyi '1' ve '0' tuşlarına basarak giriniz (Örneğin: "10001010"). Herhangi bir yanlışlığı düzeltmek için sağ (→) ve sol (←) ok tuşlarını kullanınız. Bilgileri kaydedip ayarlar menüsüne dönmek için yukarı ok (↑) tuşuna basınız.
4. Ayarlar menüsünden kanallar menüsüne geçmek için tuş takımından '2' tuşuna basınız. Burada kanal 1 için sağ (→) ok tuşunu kullanarak UPNRZ ve kanal 2 için sol (←) ok tuşunu kullanarak Manchester olarak ayarlayınız. Bilgileri kaydedip ayarlar menüsüne dönmek için yukarı ok (↑) tuşuna basınız.
5. Ayarlar menüsünden iletim hızı menüsüne geçmek için tuş takımından '3' tuşuna basınız. Sağ (→) ve sol (←) ok tuşlarını kullanarak uygun bir iletim hızı seçiniz (Örneğin: 1200 Hz). Bilgileri kaydedip ayarlar menüsüne dönmek için yukarı ok (↑) tuşuna basınız.
6. Ana menüye dönmek için tekrar yukarı ok (↑) tuşuna basınız.
7. Osiloskobunuzu açıp her iki kanal birlikte (altlı üstü) görünecek şekilde ayarlayınız. Her iki kanalın volt/div ayarını 5 V yapınız ve DC kademesine getiriniz.
8. Osiloskop probalarını osiloskop ve deney seti üzerindeki kanal 1 ve kanal 2 bağlantı noktalarına yerleştiriniz.
9. Ana menüde '1' tuşuna basarak deneyi başlatınız.
10. Osiloskop ekranında gördüğünüz şekli çiziniz.
11. Osiloskobu X-Y konumuna alınız. Osiloskop ekranında gördüğünüz şekli çiziniz.



Osiloskop İşaretleri



X-Y Düzlemi

SORULAR

1. UPNRZ ve Manchester işaretleri arasındaki farklar nelerdir?
2. Osiloskobun X-Y düzleminde gördüğünüz şekli nasıl yorumluyorsunuz?
3. Voltmetrenin DC kademesini kullanarak Kanal 1 ve Kanal 2'deki gerilimleri ölçünüz. Yorumunuz nedir?

$V(dc)_{\text{kanal1}} = \dots\dots\dots$

$V(dc)_{\text{kanal2}} = \dots\dots\dots$

4. Voltmetrenin AC kademesini kullanarak Kanal 1 ve Kanal 2'deki gerilimleri ölçünüz. Yorumunuz nedir?

$V(ac)_{\text{kanal1}} = \dots\dots\dots$

$V(ac)_{\text{kanal2}} = \dots\dots\dots$

CEVAPLAR

DENEY 5

UPNRZ ve AMI karşılaştırması

AMAÇ

Sayısal hat kodlarından olan UPNRZ ve AMI işaretlerinin karşılaştırmasını yapmak

GEREKLİ MALZEMELER

- Deney Seti
- İki adet osiloskop probu
- Osiloskop
- Voltmetre (veya Avometre)

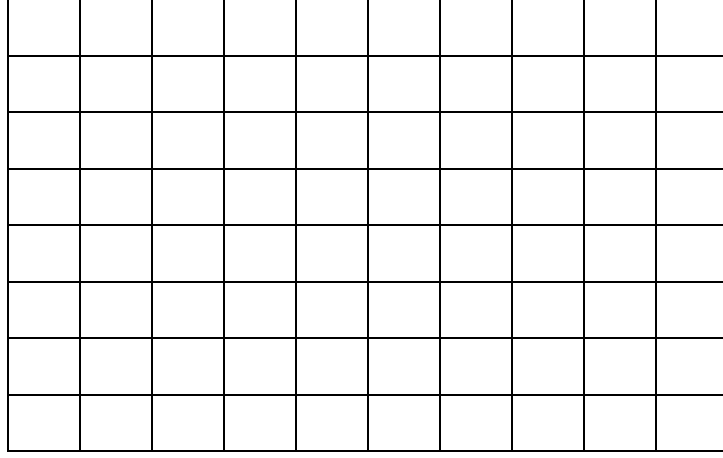
GİRİŞ

Tek kutuplu sıfıra dönmeyen (Unipolar Non-Return to Zero) kodlama, kodlama çeşitleri içinde en basit ve en ucuz yöntemdir. Sadece bir gerilim seviyesini kullanır. İkili 1 seviyesi belirli bir yüksek gerilim seviyesine, ikili 0 ise 0 volt seviyesine karşılık gelir.

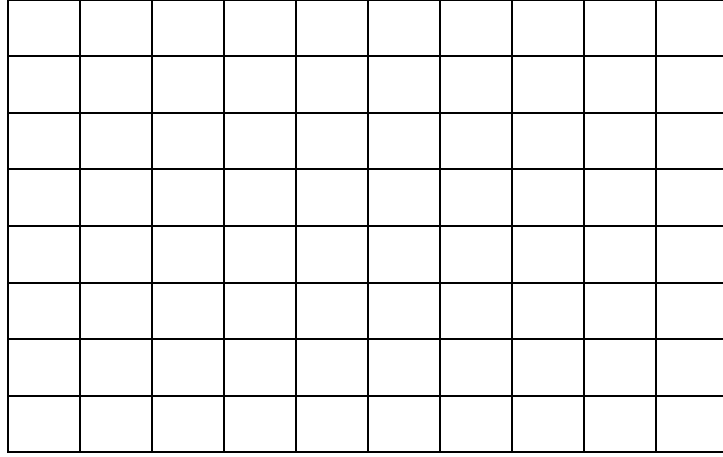
AMI (Alternate Mark Inversion) kodlama yönteminde, ikili işarettaki 0'lar değişikliğe uğratılmadan 0 olarak kodlanır, 1'ler ise her ikinci "1"'in işareti eksi yapılır. Çift kutuplu bu yöntem sayesinde bit süresi %50 kısaltılmış olur. Yani çift kutuplu olan işaretin temel frekansı karşı gelen tek kutuplu işaretin yarısı kadardır. Böylece bant genişliği de yarıya indirilmiş olur. Ayrıca işaret spektrumu darbe biçiminden bağımsız olarak daima temel frekansı içerir.

İŞLEM BASAMAKLARI

1. Deney setinin aç/kapa anahtarını açınız.
2. Ana menüden ayarlar menüsüne geçmek için tuş takımından '3' tuşuna basınız.
3. Ayarlar menüsünden ikili dizi menüsüne geçmek için tuş takımından '1' tuşuna basınız. Burada kanallardan göndereceğiniz 8 bitlik ikili veriyi '1' ve '0' tuşlarına basarak giriniz (Örneğin: "10001010"). Herhangi bir yanlışlığı düzeltmek için sağ (→) ve sol (←) ok tuşlarını kullanınız. Bilgileri kaydedip ayarlar menüsüne dönmek için yukarı ok (↑) tuşuna basınız.
4. Ayarlar menüsünden kanallar menüsüne geçmek için tuş takımından '2' tuşuna basınız. Burada kanal 1 için sağ (→) ok tuşunu kullanarak UPNRZ ve kanal 2 için sol (←) ok tuşunu kullanarak AMI olarak ayarlayınız. Bilgileri kaydedip ayarlar menüsüne dönmek için yukarı ok (↑) tuşuna basınız.
5. Ayarlar menüsünden iletim hızı menüsüne geçmek için tuş takımından '3' tuşuna basınız. Sağ (→) ve sol (←) ok tuşlarını kullanarak uygun bir iletim hızı seçiniz (Örneğin: 1200 Hz). Bilgileri kaydedip ayarlar menüsüne dönmek için yukarı ok (↑) tuşuna basınız.
6. Ana menüye dönmek için tekrar yukarı ok (↑) tuşuna basınız.
7. Osiloskobunuzu açıp her iki kanal birlikte (altlı üstü) görünecek şekilde ayarlayınız. Her iki kanalın volt/div ayarını 5 V yapınız ve DC kademesine getiriniz.
8. Osiloskop probalarını osiloskop ve deney seti üzerindeki kanal 1 ve kanal 2 bağlantı noktalarına yerleştiriniz.
9. Ana menüde '1' tuşuna basarak deneyi başlatınız.
10. Osiloskop ekranında gördüğünüz şekli çiziniz.
11. Osiloskobu X-Y konumuna alınız. Osiloskop ekranında gördüğünüz şekli çiziniz.



Osiloskop İşaretleri



X-Y Düzlemi

SORULAR

1. UPNRZ ve AMI işaretleri arasındaki farklar nelerdir?
2. Osiloskobun X-Y düzleminde gördüğünüz şekli nasıl yorumluyorsunuz?
3. Voltmetrenin DC kademesini kullanarak Kanal 1 ve Kanal 2'deki gerilimleri ölçünüz. Yorumunuz nedir?

$V(dc)_{\text{kanal1}} = \dots\dots\dots$

$V(dc)_{\text{kanal2}} = \dots\dots\dots$

4. Voltmetrenin AC kademesini kullanarak Kanal 1 ve Kanal 2'deki gerilimleri ölçünüz. Yorumunuz nedir?

$V(ac)_{\text{kanal1}} = \dots\dots\dots$

$V(ac)_{\text{kanal2}} = \dots\dots\dots$

CEVAPLAR

DENEY 6

UPNRZ ve CMI karşılaştırması

AMAÇ

Sayısal hat kodlarından olan UPNRZ ve CMI işaretlerinin karşılaştırmasını yapmak

GEREKLİ MALZEMELER

- Deney Seti
- İki adet osiloskop probu
- Osiloskop
- Voltmetre (veya Avometre)

GİRİŞ

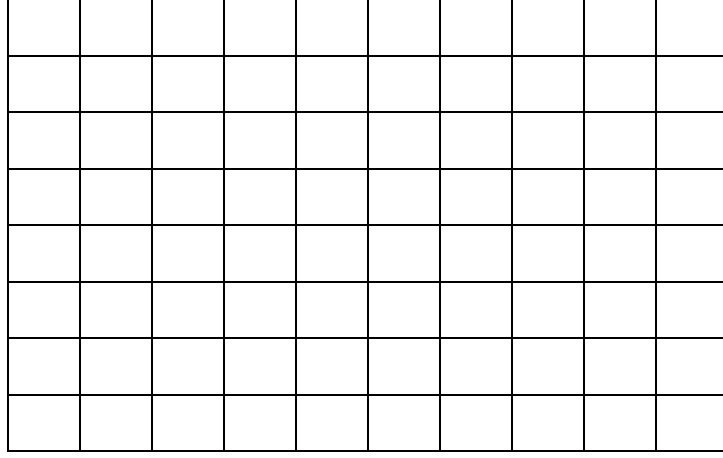
Tek kutuplu sıfıra dönmeyen (Unipolar Non-Return to Zero) kodlama, kodlama çeşitleri içinde en basit ve en ucuz yöntemdir. Sadece bir gerilim seviyesini kullanır. İkili 1 seviyesi belirli bir yüksek gerilim seviyesine, ikili 0 ise 0 volt seviyesine karşılık gelir.

Yüksek hızlı sayısal bağlantılar için CMI kodu kullanılır. Saat işareti çıkarmak için yüksek enformasyon içeriğine sahip olmasına rağmen, hat hızını iki katına çıkarması bu kodun dezavantajıdır.

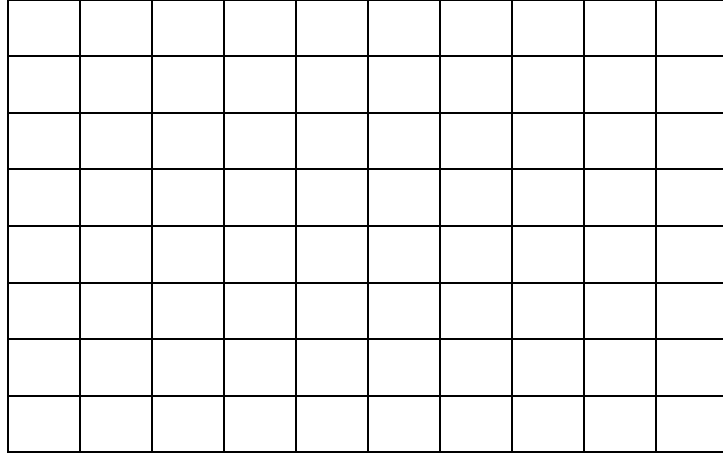
CMI kodlama yönteminde, 0 işareti (01) olarak, 1 ise ardı ardına değişen “11” ya da “00” işaretleri şeklinde kodlanır

İŞLEM BASAMAKLARI

1. Deney setinin aç/kapa anahtarını açınız.
2. Ana menüden ayarlar menüsüne geçmek için tuş takımından '3' tuşuna basınız.
3. Ayarlar menüsünden ikili dizi menüsüne geçmek için tuş takımından '1' tuşuna basınız. Burada kanallardan göndereceğiniz 8 bitlik ikili veriyi '1' ve '0' tuşlarına basarak giriniz (Örneğin: "10001010"). Herhangi bir yanlışlığı düzeltmek için sağ (→) ve sol (←) ok tuşlarını kullanınız. Bilgileri kaydedip ayarlar menüsüne dönmek için yukarı ok (↑) tuşuna basınız.
4. Ayarlar menüsünden kanallar menüsüne geçmek için tuş takımından '2' tuşuna basınız. Burada kanal 1 için sağ (→) ok tuşunu kullanarak UPNRZ ve kanal 2 için sol (←) ok tuşunu kullanarak CMI olarak ayarlayınız. Bilgileri kaydedip ayarlar menüsüne dönmek için yukarı ok (↑) tuşuna basınız.
5. Ayarlar menüsünden iletim hızı menüsüne geçmek için tuş takımından '3' tuşuna basınız. Sağ (→) ve sol (←) ok tuşlarını kullanarak uygun bir iletim hızı seçiniz (Örneğin: 1200 Hz). Bilgileri kaydedip ayarlar menüsüne dönmek için yukarı ok (↑) tuşuna basınız.
6. Ana menüye dönmek için tekrar yukarı ok (↑) tuşuna basınız.
7. Osiloskobunuzu açıp her iki kanal birlikte (altlı üstü) görünecek şekilde ayarlayınız. Her iki kanalın volt/div ayarını 5 V yapınız ve DC kademesine getiriniz.
8. Osiloskop probalarını osiloskop ve deney seti üzerindeki kanal 1 ve kanal 2 bağlantı noktalarına yerleştiriniz.
9. Ana menüde '1' tuşuna basarak deneyi başlatınız.
10. Osiloskop ekranında gördüğünüz şekli çiziniz.
11. Osiloskobu X-Y konumuna alınız. Osiloskop ekranında gördüğünüz şekli çiziniz.



Osiloskop İşaretleri



X-Y Düzlemi

SORULAR

1. UPNRZ ve AMI işaretleri arasındaki farklar nelerdir?
2. Osiloskobun X-Y düzleminde gördüğünüz şekli nasıl yorumluyorsunuz?
3. Voltmetrenin DC kademesini kullanarak Kanal 1 ve Kanal 2'deki gerilimleri ölçünüz. Yorumunuz nedir?

$V(dc)_{\text{kanal1}} = \dots\dots\dots$

$V(dc)_{\text{kanal2}} = \dots\dots\dots$

4. Voltmetrenin AC kademesini kullanarak Kanal 1 ve Kanal 2'deki gerilimleri ölçünüz. Yorumunuz nedir?

$V(ac)_{\text{kanal1}} = \dots\dots\dots$

$V(ac)_{\text{kanal2}} = \dots\dots\dots$

CEVAPLAR

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı : BOZDEMİR, Sabri Can
Uyruğu : T.C.
Doğum tarihi ve yeri : 26.03.1982 Fındıklı
Medeni hali : Bekar
e-mail : scbozdemir@yahoo.com

Eğitim

Derece	Eğitim Birimi	Mezuniyet tarihi
Lisans	Gazi Üniversitesi Elektronik ve Bilgisayar Eğitimi Elektronik Öğretmenliği	2004
Lise	Samsun Atakum A.T.L.	2000

İş Deneyimi

Yıl	Yer	Görev
2004-2007	Zile Endüstri Meslek Lisesi	Elektronik Öğretmeni

Yabancı Dil

İngilizce

Hobiler

Robotlar, Programlama, Arabalar, Sinema, Bilgisayar teknolojileri, Satranç