



T.C.
İstanbul Üniversitesi
Mühendislik Fakültesi
Elektrik-Elektronik Mühendisliği Bölümü

GERÇEK ZAMANLI SES TANIMA

BİTİRME PROJESİ

Hazırlayanlar:
Berrak ÖZTÜRK 1316040092
Tunay ÇAKAR 1316040093

Danışman: Prof. Dr. Aydın AKAN

Haziran 2007

**EK 2: İNGİLİZCE YAZILMIŞ
BİTİRME PROJELERİ İÇİN
KAPAK DÜZENİ**

12 punto ile
6 satır boşluk

12 punto ve
tek satır aralığı ile

12 punto ile
7 satır boşluk

20 punto kalın ve
tek satır aralığı

12 punto ile
6 satır boşluk

18 punto

12 punto ile
6 satır boşluk

14 punto kalın ve
tek satır aralığı

12 punto ile
2 satır boşluk

14 p. kalın

12 punto ile
8 satır boşluk

14 punto



T.C.
Istanbul University
Engineering Faculty
Department of Electrical and Electronics Engineering

**TITLE OF THE
GRADUATION PROJECT**

GRADUATION PROJECT

**Prepared by: Student's Name and Surname
Student's Number**

Supervisor: Title Name and Surname

October 2003

Üst kenardan 3 cm

2.3 cm

0.7 cm

1.8 cm

3.7 cm

1.3 cm

3.3 cm

3.2 cm

2.1 cm

4.2 cm

Alt kenardan 3 cm

ÖNSÖZ

Bu projede C6713 DSK seti ile Matlab 7.0 programının Simulink özelliği kullanılarak gerçek zamanlı ses analizi yapılmıştır. Kullanıcı mikrofona ‘a’ , ‘e’ veya ‘i’ seslerinden herhangi birini söylemekte ve simulasyon sonucu olarak bu sese ait kaydı dinleyerek, ses tanıma işleminin gerçekleştiğini duymaktadır.

Projemizin gerçekleşmesinde DSP setlerinin teminini sağlayan ve sinyal işleme alanındaki bilgileri ile bize her zaman destek olan değerli hocamız Prof. Dr. Aydın AKAN’a, projemizi hazırladığımız dönem boyunca bize yol gösteren ve çalışmaya daima teşvik eden Arş. Gör. Koray KAYABOL’a, ‘Gerçek Zamanlı Kişi Tanıma’ bitirme projesini gerçekleştiren arkadaşlarımız Z. Burcu KURİL’e ve R. Fatih ÜSTOK’a teşekkürlerimizi sunarız.

İÇİNDEKİLER

ÖNSÖZ	i
İÇİNDEKİLER	ii
ÖZET	iv
ŞEKİLLER LİSTESİ	v
1. GİRİŞ	1
2. GERÇEK ZAMANLI SES ANALİZİ	2
2.1 C6713 DSK Kartının Tanıtılması	2
2.1.1 C6713 DSK Kartının Test Edilmesi	3
2.1.2 Code Composer Programının Tanıtılması	4
2.2 Yeni Bir Model Dosyası Oluşturmak	5
2.3 ‘seskayıt’ Alt Programının Oluşturulması	6
2.4 Kullanıcıya Ait ‘a’, ‘e’ ve ‘i’ Seslerinin Kayıt Edilmesi	8
2.5 Simülasyon Sonucunu Bildiren Ses Dosyalarının Kayıt Edilmesi	9
2.6 Ses Dosyalarının Workspace’e Aktarılması	11
2.6.1 Workspace1 Alt Programının Oluşturulması	11
2.6.2 Workspace2 Alt Programının Oluşturulması	14
2.7 Kesme Alt Programının Yazılması ve Çalıştırılması	15
2.8 Matlab Simulink Ortamında Simülasyonun Yapılması	19
2.8.1 ‘aei_matlab.mdl’ Ana Programını Oluşturan Bölümler	19
2.8.1.1 Bölüm 1	20
2.8.1.2 Bölüm 2	21
2.8.1.3 Bölüm 3	24
2.8.1.4 Bölüm 4	26
2.8.1.5 Bölüm 5	28
2.8.1.6 Bölüm 6	32
2.8.1.7 Bölüm 7	33
2.8.1.8 Bölüm 8	34
2.8.1.9 Bölüm 9	35
2.9 C6713 DSK Kartı Üzerinde Simülasyonun Gerçekleştirilmesi	37

2.9.1	‘aei_kart.mdl’ Ana Programını Oluşturan Bölümler.	37
2.9.1.1	Bölüm 1.	38
2.9.1.2	Bölüm 2.	39
2.9.2	‘aei_kart.mdl’ Ana Programının CCS Programına Aktarılması.	40
2.9.3	‘aei_kart.pjt’ Proje Dosyasının C6713 DSK Setine Yüklenmesi.	41
2.9.4	Simülasyonun C6713 DSK Setinde Çalıştırılması.	42
2.10	Uygulama Projesi: Gerçek Zamanlı Kişi Tanıma.	43
3.	SONUÇ.	45
	KAYNAKLAR.	46
	EK 1: Kişi Tanıma Projesi için Dosyaların Çalıştırılması	47
	EK 2: Ses Tanıma Projesi için Dosyaların Çalıştırılması.	48
	ÖZGEÇMİŞ	49

ÖZET

Projemiz ‘Gerçek Zamanlı Ses Tanıma’ işlevini gerçekleştirmektedir. Bu amaçla Matlab programına ait Simulink ortamı kullanılarak blok diyagram oluşturulmuş, daha sonra gerekli değişiklikler yapılarak aynı simulasyonun C6713 DSK setinde çalıştırılması sağlanmıştır.

Proje için temel olarak bir kişiye ait ‘a’ , ‘e’ ve ‘i’ sesleri kayıt edilip, gerekli işlemlerden geçirilerek kütüphane elde edilmiştir. Oluşturulan blok diyagram mikrofon aracılığıyla gönderilen ses sinyalinin korelasyon yöntemini kullanarak bu kütüphane ile karşılaştırmakta ve sonuçları değerlendirerek uygun çıkış işlemini gerçekleştirmektedir.

Sonuç olarak kullanıcı matlab simulink ortamında simulasyonu çalıştırarak, mikrofona söylediği bu üç sesli harfe ait ses dosyalarından, söylenen harfe ait olanı dinlemekte, böylece ses tanıma işleminin gerçekleştiğini duymaktadır.

Aynı blok diyagramda C6713 DSK için gerekli değişikliklerin yapılmasının ardından, Code Composer Studio programı aracılığıyla sete yükleme yapılmakta ve simulasyonun bu ortamda da çalıştırılması sağlanmaktadır.

ŞEKİLLER LİSTESİ

	Sayfa
Şekil 2.1: C6713 DSK Kartı.	2
Şekil 2.2: C6713 DSK Kartı Blok Diyagramı.	3
Şekil 2.3: 6713DSK Diagnostics Programı.	3
Şekil 2.4: 6713DSK Diagnostics Programı Test Aşaması.	4
Şekil 2.5: 6713DSK Diagnostics Programı Test Sonucu.	4
Şekil 2.6: Matlab Simulink Örnek Dosyasının Farklı Kayıt Edilmesi.	5
Şekil 2.7: Yeni Model Dosyası.	5
Şekil 2.8: ‘aei’ ve ‘araclar’ Klasörleri.	6
Şekil 2.9: Ses Kayıt Alt Programı.	6
Şekil 2.10: Mikrofon ve Ses Dosyası.	7
Şekil 2.11: Mikrofon Ayarları.	8
Şekil 2.12: ‘a1.wav’ Ses Dosyasının Oluşturulması.	8
Şekil 2.13: ‘e1.wav’ Ses Dosyasının Oluşturulması.	9
Şekil 2.14: ‘i1.wav’ Ses Dosyasının Oluşturulması.	9
Şekil 2.15: ‘a_dediniz.wav’ Ses Dosyasının Oluşturulması.	10
Şekil 2.16: ‘e_dediniz.wav’ Ses Dosyasının Oluşturulması.	10
Şekil 2.17: ‘i_dediniz.wav’ Ses Dosyasının Oluşturulması.	10
Şekil 2.18: Ses Dosyasının Blok Olarak Atanması.	11
Şekil 2.19: Sinyalin Workspace’e Atılması.	12
Şekil 2.20: Workspace1 Alt Programı.	12
Şekil 2.21: From Wave File Bloğuna Ses Dosyası İsminin Yazılması.	13
Şekil 2.22: Signal To Workspace Bloğuna Vektör İsminin Yazılması.	13
Şekil 2.23: Matlab Workspace Görünümü.	13
Şekil 2.24: Workspace2 Alt Programı.	14
Şekil 2.25: From Wave File Bloğuna Ses Dosyası İsminin Yazılması.	14
Şekil 2.26: Signal To Workspace Bloğuna Vektör İsminin Yazılması.	15
Şekil 2.27: Matlab Workspace Görünümü.	15
Şekil 2.28: Veri Gönderme Pencere Uzunluğu.	15
Şekil 2.29: ‘a’ Sesine Ait Vektör.	16
Şekil 2.30: ‘e’ Sesine Ait Vektör.	16
Şekil 2.31: ‘i’ Sesine Ait Vektör.	17
Şekil 2.32: Kesme Alt Programı.	17
Şekil 2.33: Matlab Workspace Görünümü.	18
Şekil 2.34: ‘aei_matlab.mdl’ Ana Programı.	19
Şekil 2.35: Bölüm 1.	20
Şekil 2.36: Mikrofon Özellikleri.	20
Şekil 2.37: Bölüm 2.	21
Şekil 2.38: Toplama Bloğu.	21
Şekil 2.39: Geciktirme Bloğu.	21
Şekil 2.40: Geciktirme Bloğunun Özellikleri.	22
Şekil 2.41: Sinyali Workspace’ten Alma Bloğu.	22
Şekil 2.42: Sinyali Workspace’ten Alma Bloğunun Özellikleri.	23
Şekil 2.43: Kesilmiş Ses Örneği, 1. Döngü Sonrası Elde Edilen Sinyal.	23
Şekil 2.44: Bölüm 3.	24
Şekil 2.45: Korelasyon Bloğu.	24
Şekil 2.46: Korelasyon Örneği.	25
Şekil 2.47: Korelasyon Örneği Sonuçları.	25
Şekil 2.48: Bölüm 4.	26

Şekil 2.49: Maksimum Bloğu.	26
Şekil 2.50: Maksimum Bloğu Özellikleri.	26
Şekil 2.51: Buffer Bloğu.	27
Şekil 2.52: Buffer Bloğu Özellikleri.	27
Şekil 2.53: 1.Maksimum Bloğu Çıkış Matrisi.	28
Şekil 2.54: Bölüm 5.	28
Şekil 2.55: Sabit ile Karşılaştırma Bloğu.	28
Şekil 2.56: Sabit ile Karşılaştırma Bloğu Özellikleri.	29
Şekil 2.57: Karşılaştırma Bloğu.	29
Şekil 2.58.a: Karşılaştırma Bloğu Özellikleri.	29
Şekil 2.58.b: Karşılaştırma Bloğu Özellikleri.	30
Şekil 2.59: VE Kapısı Bloğu.	30
Şekil 2.60.a: VE Kapısı Bloğu Özellikleri.	31
Şekil 2.60.b: VE Kapısı Bloğu Özellikleri.	31
Şekil 2.61: Karşılaştırma Mantığı.	31
Şekil 2.62: Bölüm 6.	32
Şekil 2.63: Anahtar Bloğu.	32
Şekil 2.64: Anahtar Bloğu Özellikleri.	33
Şekil 2.65: Bölüm 7.	33
Şekil 2.66: Bölüm 8.	34
Şekil 2.67: Çarpma Bloğu.	34
Şekil 2.68: Sabit Bloğu.	34
Şekil 2.69: Sabit Bloğu Özellikleri.	35
Şekil 2.70: Bölüm 9.	35
Şekil 2.71: Toplama Bloğu Özellikleri.	36
Şekil 2.72: Hoparlör Bloğu.	36
Şekil 2.73: 'aei_kart.mdl' Ana Programı.	37
Şekil 2.74: Bölüm 1.	38
Şekil 2.75: ADC Bloğu.	38
Şekil 2.76: ADC Bloğu Özellikleri.	39
Şekil 2.77: Bölüm 2.	39
Şekil 2.78: DAC Bloğu.	39
Şekil 2.79: CCS Programına Aktarım.	40
Şekil 2.80: Hata Ekranı.	40
Şekil 2.81: 'aei_kart.pjt' Proje Dosyası.	41
Şekil 2.82: CCS Ayarları.	41
Şekil 2.83: Build İşlemi.	41
Şekil 2.84: Programın Yüklenmesi	42
Şekil 2.85: Run İşlemi.	42
Şekil 2.86: Restart İşlemi.	43
Şekil 2.87: Gerçek Zamanlı Kişi Tanıma Ana Program Bloğu.	43

KAYNAKLAR

- [1] Rulph Chassaing, Worcester Polytechnic Institute, “Digital Signal Processing and Applications with the C6713 and C6416”, 1st ed, Wiley, New Jersey, 2005.
- [2] Prof. Dr. Uğur Arifoğlu, Sakarya Üniversitesi Müh. Fak. E-E Bölümü, “Matlab 7.04 Simulink ve Mühendislik Uygulamaları”, 1st ed, Alfa Yayınları, Sakarya, 2005.

1. GİRİŞ

Projemiz iki ortamda gerçekleştirilmektedir. Esas olarak ‘Gerçek Zamanlı Ses Tanıma’ işlemi Matlab Simulink ortamında dizayn edilmiş blok diyagram aracılığıyla yapılmakta, gerekli değişikliklerin uygulanmasının ardından aynı simulasyon, C6713 DSK Starter Kit ortamında da çalıştırılabilmektedir.

Simulasyonların her iki ortamda da çalışabilmesi için gerekli alt programlar önceden oluşturulmalı ve istenen veriler elde edilmelidir.

Bu alt programlar aracılığıyla öncelikle kullanıcıya ait ‘a’ , ‘e’ ve ‘i’ örnek sesleri kayıt edilir, ilgili işlemler uygulanarak karşılaştırma kütüphanesi oluşturulur. Ayrıca karşılaştırma sonucunun kullanıcıya dinletilebilmesi için de yine bu örnek seslerin, her birine ait ses dosyaları önceden kayıt edilmelidir.

Her iki ortam içinde kullanılan blok diyagram, dolayısıyla yöntem aynıdır. Ancak Matlab Simulink ortamında Windows’a ait mikrofon, hoparlör ve ses dosyaları (.wav) kullanılırken, DSP setindeki simulasyonda C6713 DSK’ye ait ADC (analog-dijital çevirici), DAC (dijital-analog çevirici) ve Matlab Workspace alanı kullanılmaktadır.

Ana Programlar:

aei_matlab.mdl

aei_kart.mdl

Alt Programlar:

seskayit.mdl

workspace1.mdl

workspace2.mdl

kesme.m

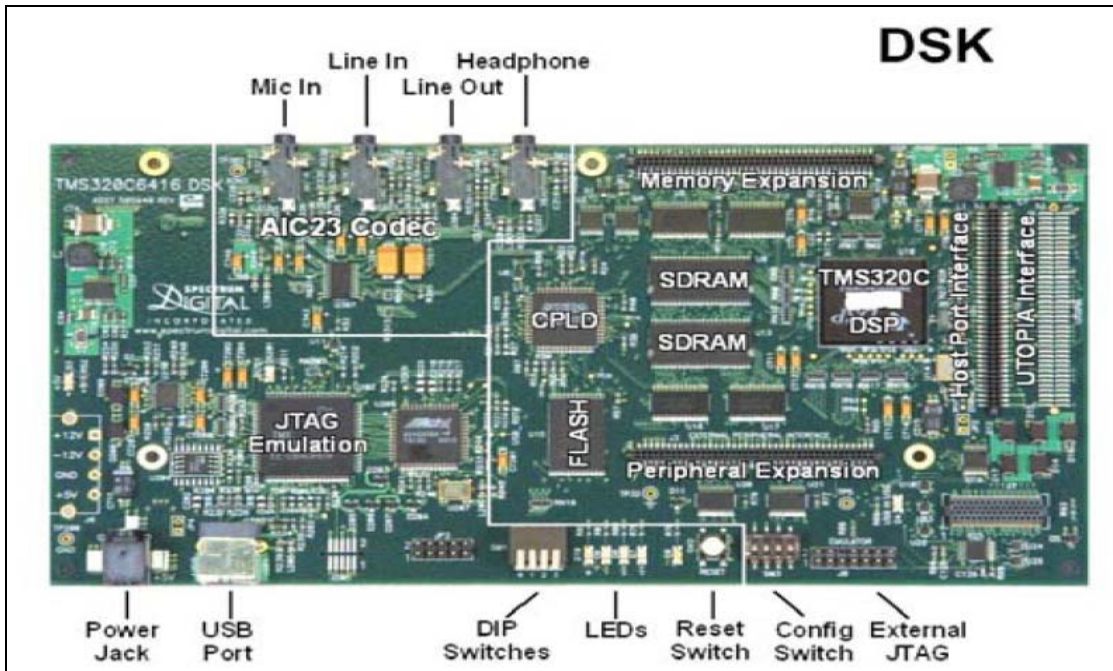
2. GERÇEK ZAMANLI SES ANALİZİ

Bu bölümde C6713 DSK setinin tanıtılması, ana programlara ait blok diyagramların oluşturulması, alt programların hazırlanıp kullanılarak verilerin elde edilmesi ve simulasyonun çalışma ilkeleri adım adım anlatılacaktır. Açılımlar yapıldıktan sonra, ilgili yönergeler ile işlemlerin nasıl gerçekleştirildiği belirtilmiştir.

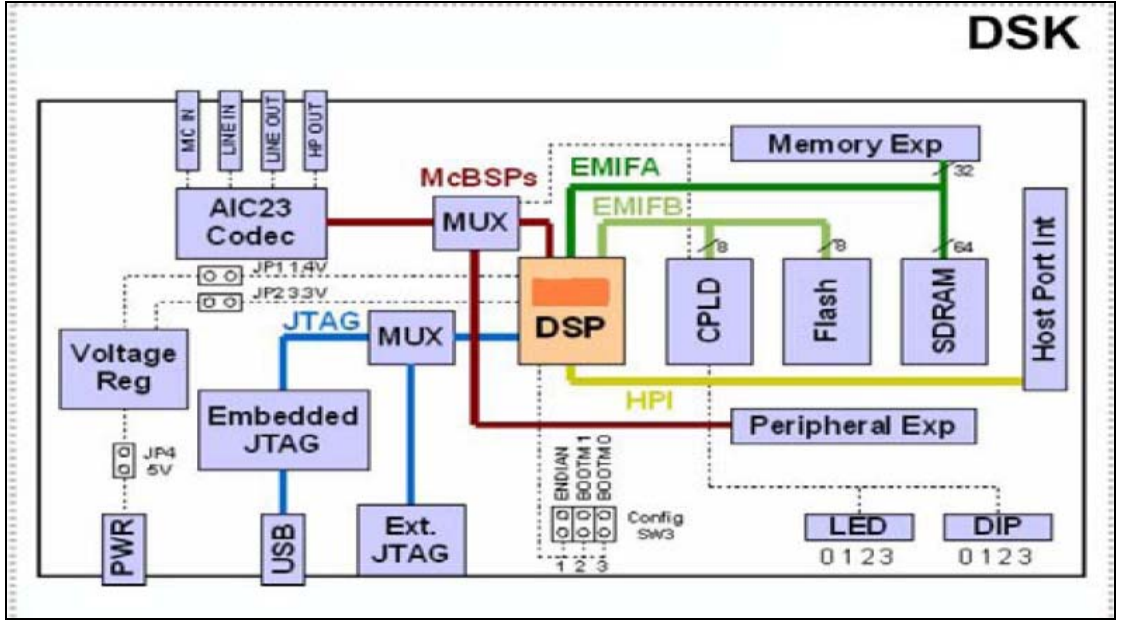
2.1 C 6713 DSK Kartının Tanıtılması:

C6713 DSK kartı genel olarak gerçek zamanlı sinyal işleme uygulamaları için kullanılır. DSK C6713 kartı 16 MB SDRAM, 256 kB flash memory elemanlarını üzerinde bulundurmaktadır. Kart üzerinde Mic In, Line In, Line Out, Headphone giriş ve çıkışları bulunmaktadır. (Bkz. Şekil 2.1) İşlemci frekansı 255 MHz.'dir.

C6713 DSK kartının çalışması için kurulum CD'sinde bulunan '6713 Diagnostics Utility' ve 'Code Composer Studio V2 (CCS)' programlarının yüklenmesi gereklidir. 6713 Diagnostics Utility programı DSK kartını ve bilgisayar bağlantısının doğruluğunu kontrol eder. Code Composer Studio programı ise farklı programlama dillerinde (Matlab Simülink, C, C++) oluşturulmuş kodları, assembler diline çevirerek C6713 DSK kartına yüklenmesini sağlar.



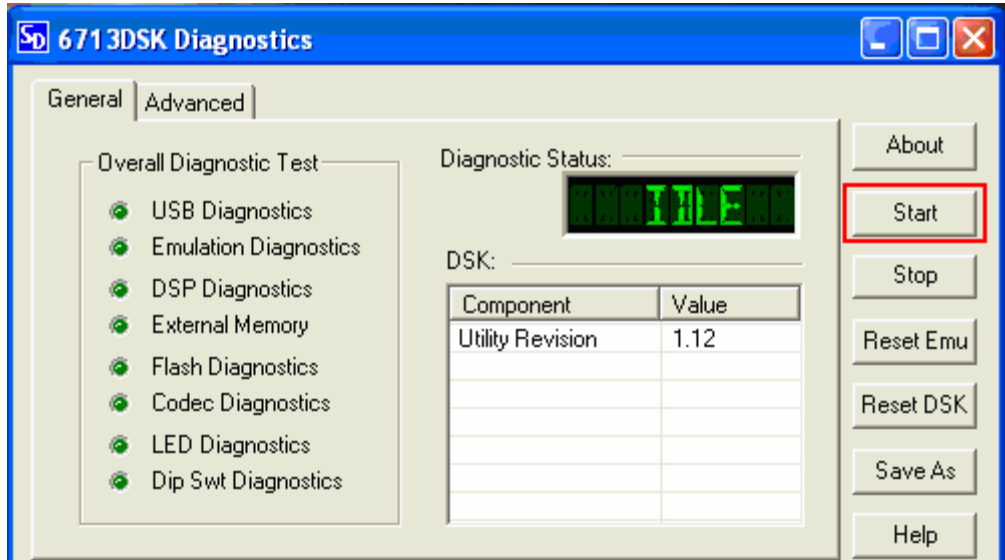
Şekil 2.1 C6713 DSK Kartı



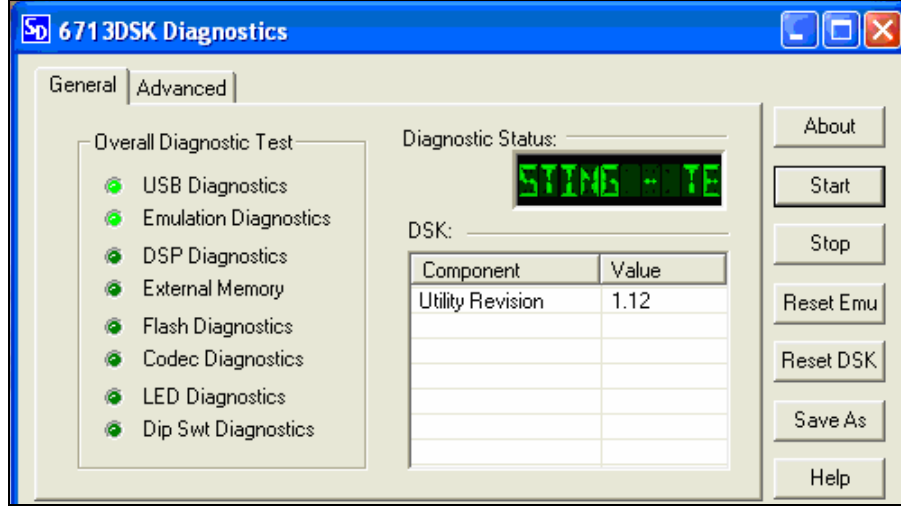
Şekil 2.2 C6713 DSK Kartı Blok Diyagramı

2.1.1 C 6713 DSK Kartının Test Edilmesi:

C6713 kart bağlantıları yapılır ve masa üstünde yeni kurulan ‘6713 DSK Diagnostics’ programı açılır. Programda ‘Start’ tuşuna basılarak test başlatılır. Bkz. Şekil 2.3

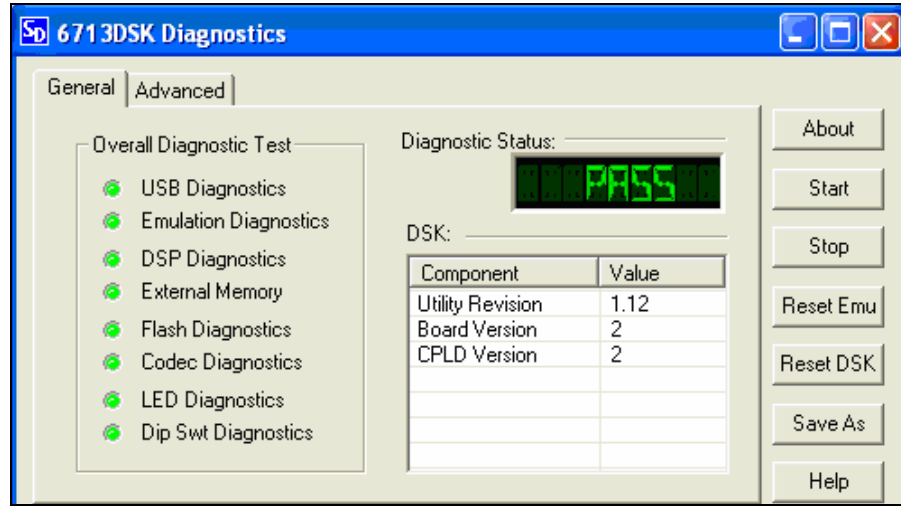


Şekil 2.3 6713DSK Diagnostics Programı



Şekil 2.4 6713DSK Diagnostics Programı Test Aşaması

Test tamamlandığında, bütün ledler yeşile dönmüştür ve 'diagnostic status' penceresinde kartta ve bağlantıda hata olmadığını gösteren 'PASS' yazısı gözlemlenir. Bkz. Şekil 2.5



Şekil 2.5 6713DSK Diagnostics Programı Test Sonucu

Test tamamlandığında kart kullanıma hazır hale gelir. Burada dikkat edilmesi gereken nokta kartın test edilmesi sırasında CCS ve Matlab programlarının kapalı olmasıdır. Aksi takdirde 6713 Diagnostics programı hata verecektir.

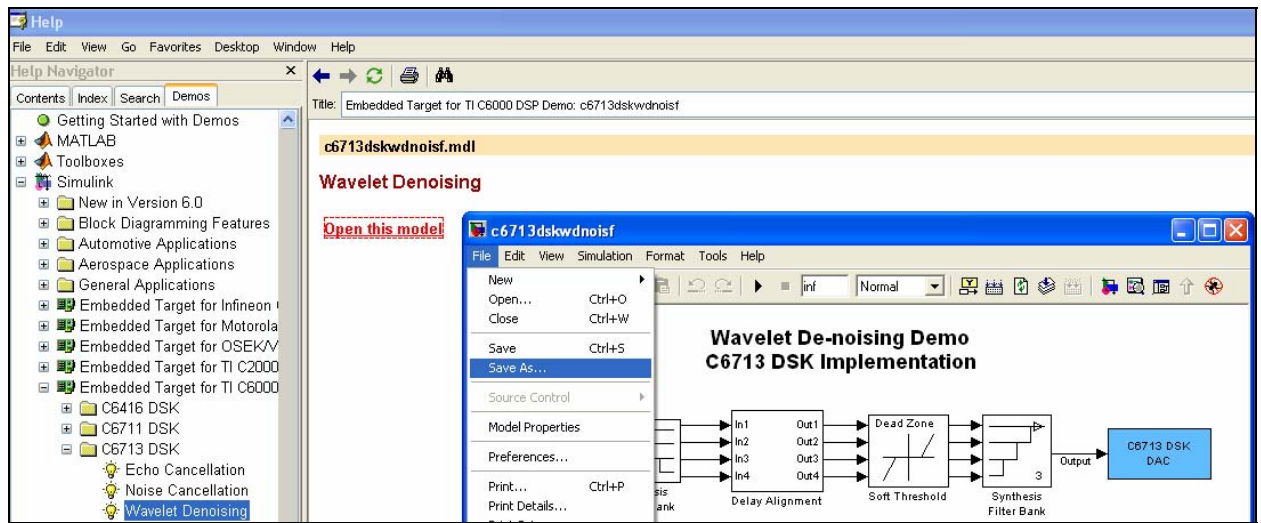
2.1.2 Code Composer Programının Tanıtılması:

Code Composer Studio programı, Matlab Simulink'te oluşturduğumuz blok diyagramını kart tarafından tanınması için assembler dilene çevirecek programdır.

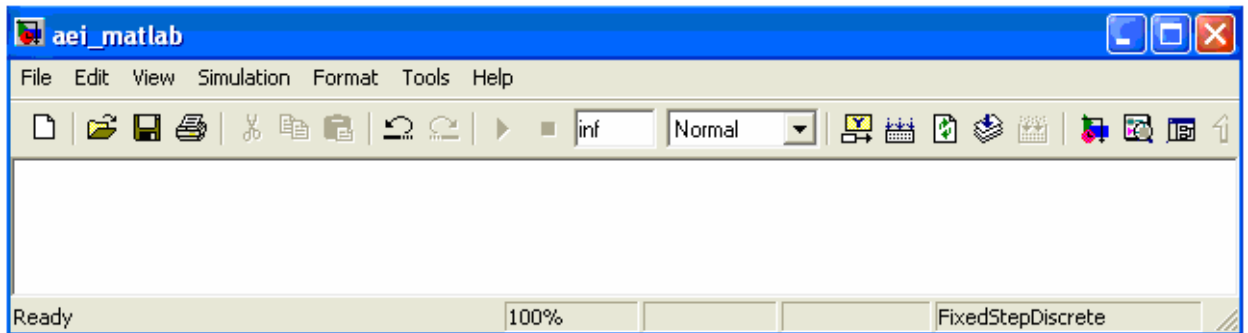
2.2 Yeni Bir Model Dosyası Oluşturmak:

Simulasyonun oluşturulması için yeni bir Simulink Model File açılabilir, ancak gerekli ayarların yapılması uzun zaman almaktadır. Bunun yerine Matlab Simulink içerisindeki C6713 DSK setine ait örnek dosyalardan herhangi biri seçilerek farklı kaydedilir ve içerisindeki tüm blok ve açıklamalar silinir. Böylece ayarları yapılmış boş bir model dosyası elde edilmiş olur.

**Help>Full Product Family Help>Demos>Simulink>Embedded Target for TI C6000>C6713
>Wavelet Denoising>Open this model>File>Save As...**



Şekil 2.6 Matlab Simulink Örnek Dosyasının Farklı Kayıt Edilmesi



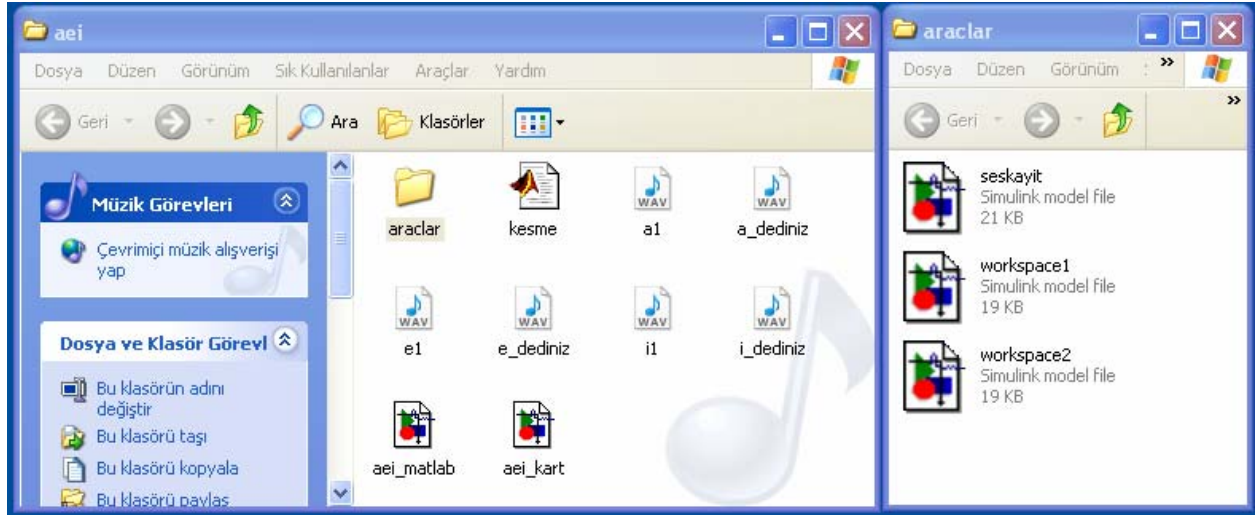
Şekil 2.7 Yeni Model Dosyası

Yeni oluşturulan bu dosya, ana programı oluşturmak için kullanılacağından 'aei_matlab' adı ile kayıt edilecektir.

Ancak bu aşamadan sonra oluşturulacak ana program, alt programlar, ses örnek dosyaları ile sonuç ses dosyalarının biriktirilmesi için Matlab içerisinde bulunan Work klasörüne, 'aei' isimli

bir alt klasör açmak yararlı olacaktır. Yine bu amaçla oluşturulan aei klasörünün içine de ‘araclar’ ismi ile farklı bir klasör açılmalıdır.

Tüm işlemler gerçekleştirildiğinde oluşacak ‘aei’ ve ‘araclar’ klasörlerinin görüntüleri Şekil 2.8’deki gibidir.

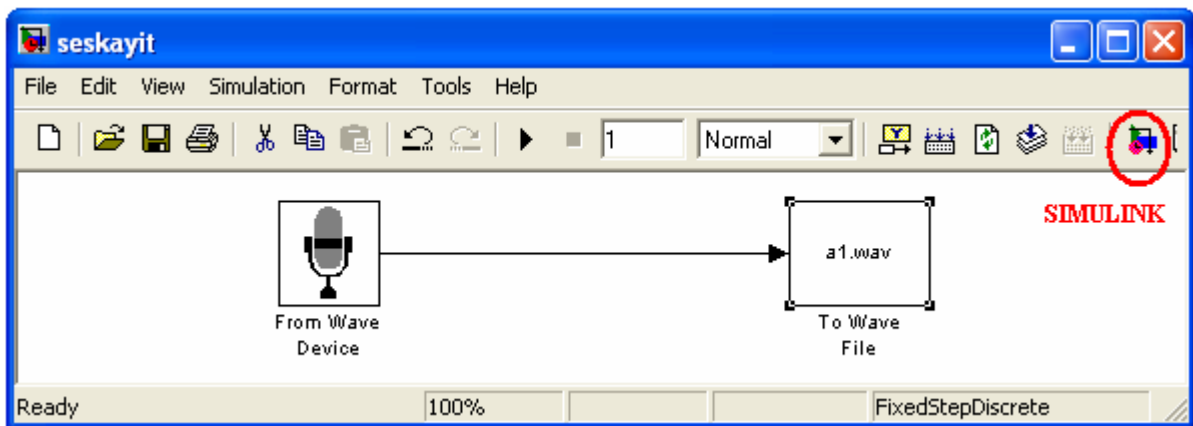


Şekil 2.8 ‘aei’ ve ‘araclar’ Klasörleri

Dikkat edilirse farklı kayıt edilib, silinerek elde edilen boş model dosyası ‘aei’ klasörünün içerisinde oluşturulmuştur.

2.3 ‘seskayit’ Alt Programının Oluşturulması:

Bu amaç ile 2.2 bölümünde anlatıldığı şekilde ya da ‘aei_matlab’ dosyası kopyalanarak ‘araclar’ alt klasörü içerisinde ‘ses kayıt’ adlı yeni bir boş model dosyası elde edilir. Bu dosyaya herhangi bir bloğu eklemek amacı ile Şekil 2.9’te gösterilmiş simulink’e ait **radio butonuna** basılır.

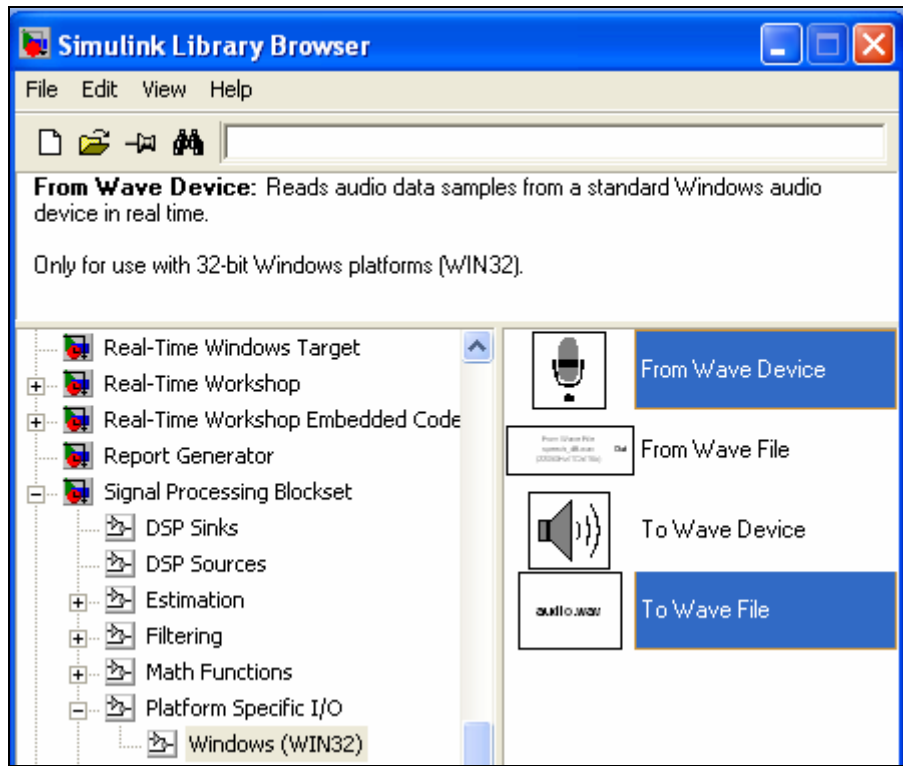


Şekil 2.9 Ses Kayıt Alt Programı

Daha sonra Şekil 2.9’da görülen mikrofon ve ses dosyası blokları aşağıdaki yönergeler izlenerek bulunur ve tutup taşıma yöntemi ile ‘seskayıt’ dosyasına eklenir. Ara bağlantı gerçekleştirilir.

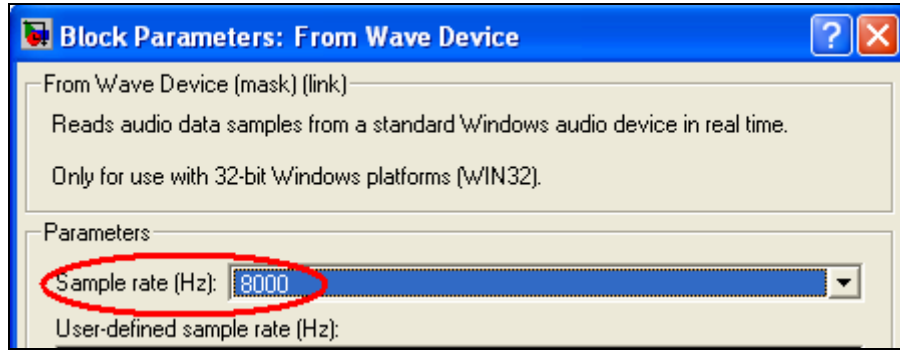
Simulink Library Browser > Signal Processing Blockset > Platform Specific I/O > Windows (WIN32) > From Wave Device

Simulink Library Browser > Signal Processing Blockset > Platform Specific I/O > Windows (WIN32) > To Wave File



Şekil 2.10 Mikrofon ve Ses Dosyası

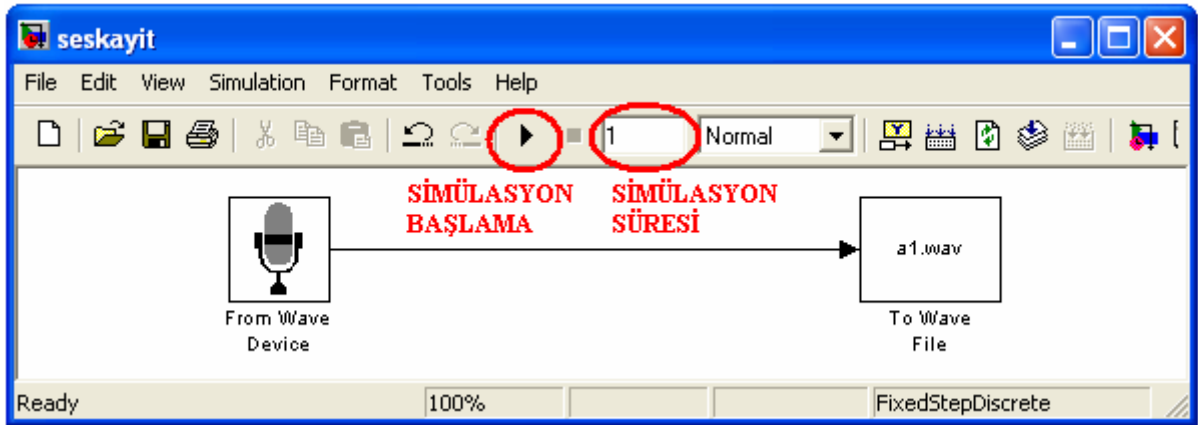
Mikrofon bloğu çift tıklanarak Şekil 2.11’deki pencereye ulaşılır. Burada dikkat edilmesi gereken Sample rate bölümünde 8000 Hz değerinin bulunmasıdır. Çünkü tüm proje boyunca kullanılacak örnekleme frekansımız 8 kHz’dir. Ses dosyası oluşturma bloğu da çift tıklanarak istenilen isim yazılabilir.



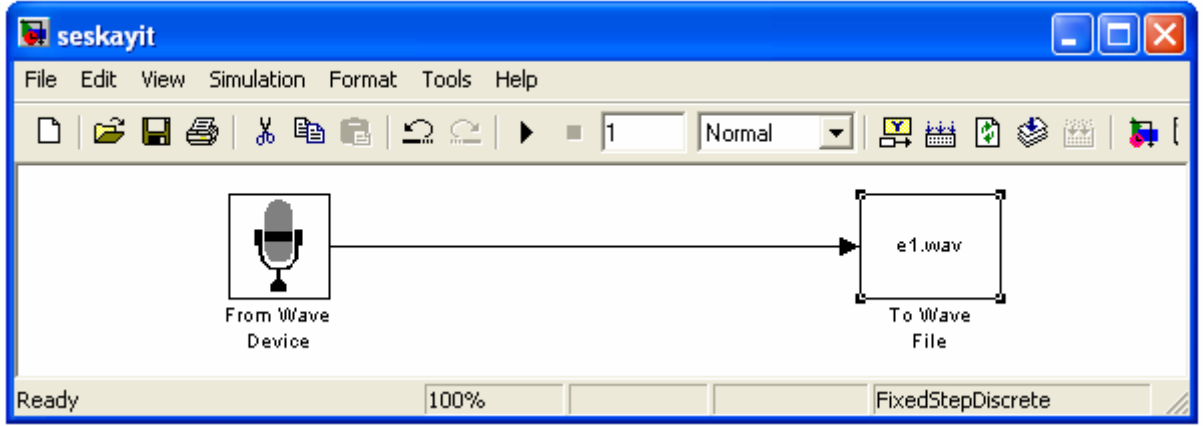
Şekil 2.11 Mikrofon Ayarları

2.4 Kullanıcıya Ait ‘a’ , ‘e’ ve ‘i’ Seslerinin Kayıt Edilmesi:

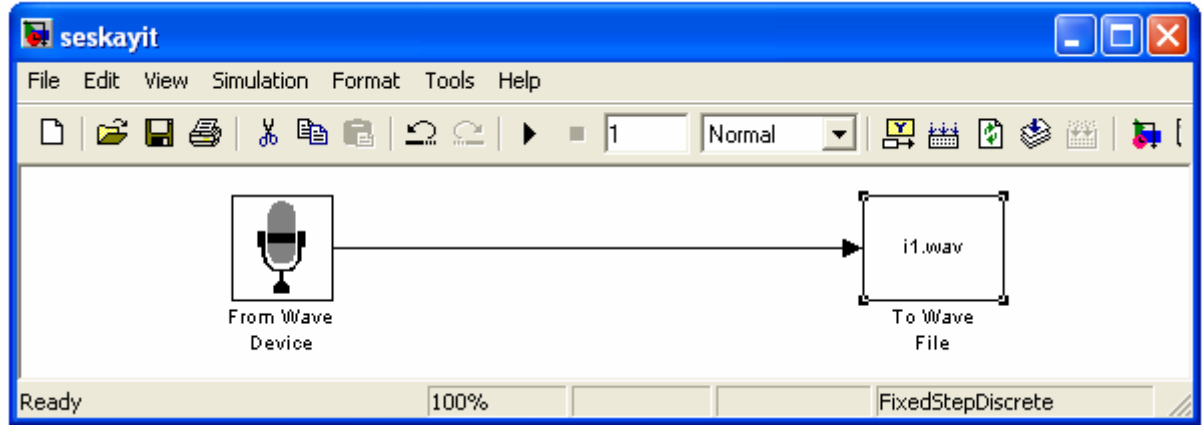
Oluşturulan ‘seskayıt’ dosyası açılarak, kayıt edilecek dosya ismi olarak ‘a1.wav’ yazılır. Simülasyon süresi 1 saniyeye ayarlandıktan sonra simülasyon başlama butonuna basılarak kullanıcının ‘a’ sesi bilgisayara bağlanacak bir harici mikrofon aracılığıyla kayıt edilir (Şekil 2.12). Daha sonra sırasıyla aynı işlemler tekrarlanarak ‘e1.wav’ ve ‘i1.wav’ örnek ses dosyaları oluşturulur. Bu dosyalar yine ‘aei’ klasörünün içinde bulunmalıdır. Bkz. Şekil 2.8



Şekil 2.12 ‘a1.wav’ Ses Dosyasının Oluşturulması



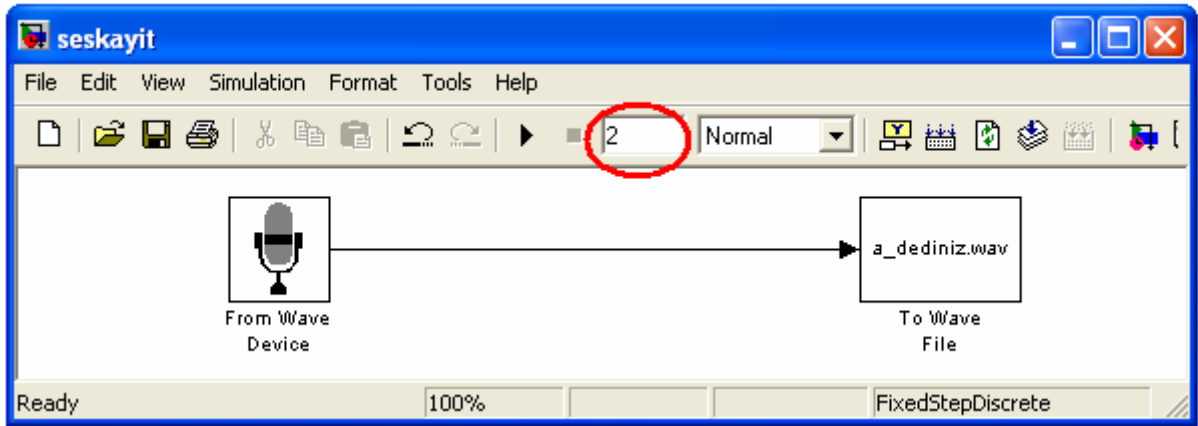
Şekil 2.13 ‘e1.wav’ Ses Dosyasının Oluşturulması



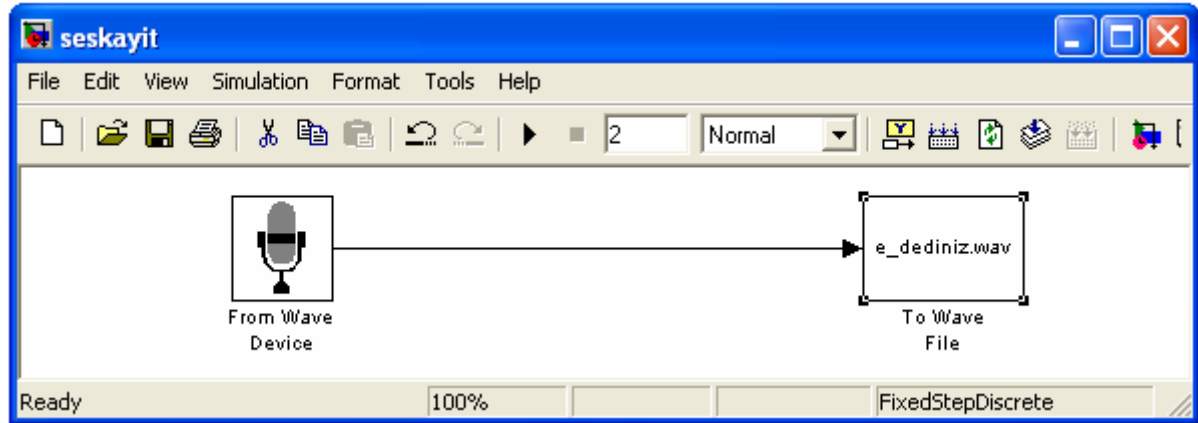
Şekil 2.14 ‘i1.wav’ Ses Dosyasının Oluşturulması

2.5 Simülasyon Sonucunu Bildiren Ses Dosyalarının Kayıt Edilmesi:

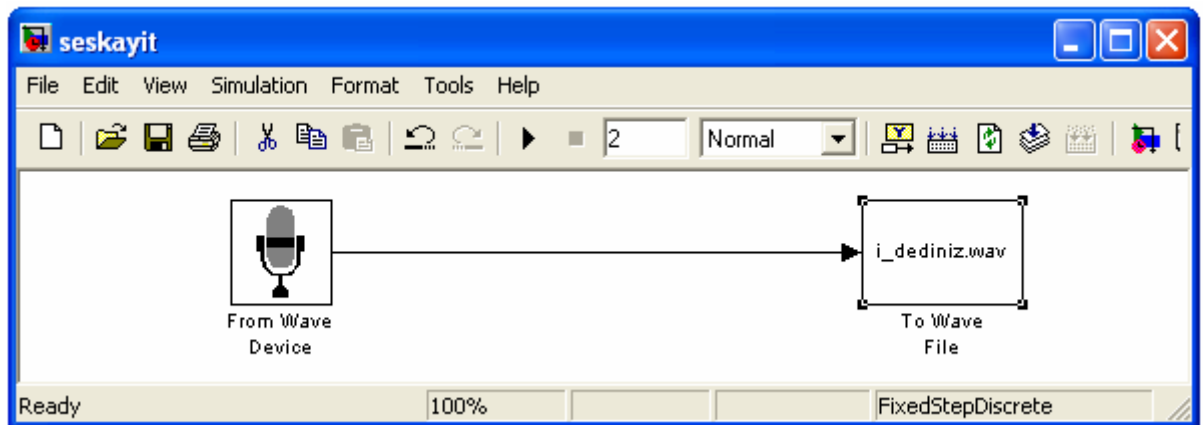
Oluşturulan ‘seskayit’ dosyası açılarak, kayıt edilecek dosya ismi olarak ‘a_dediniz.wav’ yazılır. Simülasyon süresi 2 saniyeye ayarlandıktan sonra simülasyon başlama butonuna basılarak kullanıcı yada bir başka kişiye ait ‘a dediniz’ sesi bilgisayara bağlanacak bir harici mikrofon aracılığıyla kayıt edilir (Şekil 2.15). Daha sonra sırasıyla aynı işlemler tekrarlanarak ‘e_dediniz.wav’ ve ‘i_dediniz.wav’ ses dosyaları oluşturulur. Bu dosyalar yine ‘aei’ klasörünün içinde bulunmalıdır. Bkz. Şekil 2.8



Şekil 2.15 ‘a_dediniz.wav’ Ses Dosyasının Oluşturulması



Şekil 2.16 ‘e_dediniz.wav’ Ses Dosyasının Oluşturulması



Şekil 2.17 ‘i_dediniz.wav’ Ses Dosyasının Oluşturulması

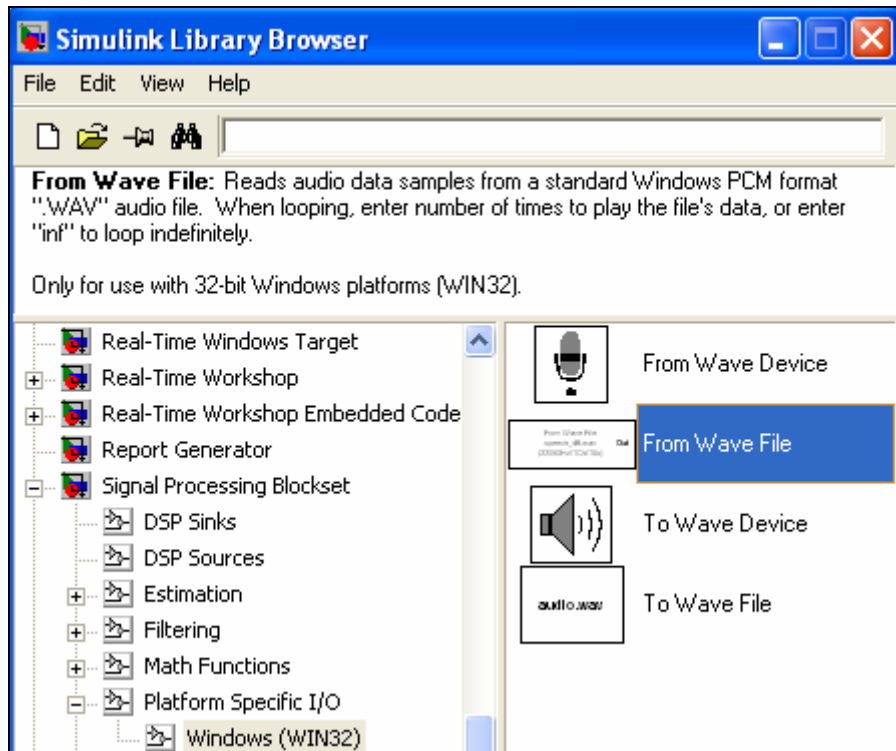
2.6 Ses Dosyalarının Workspace'e Aktarılması:

Bu aşamaya kadar oluşturulan 6 adet ses dosyası workspace ortamına aktarılmalıdır. Bu amaçla 'workspace1.mdl' ve 'workspace2.mdl' alt programları **aei > araçlar** klasörü içerisinde oluşturulmuştur. Bkz. Şekil 2.8

2.6.1 Workspace1 Alt Programının Oluşturulması:

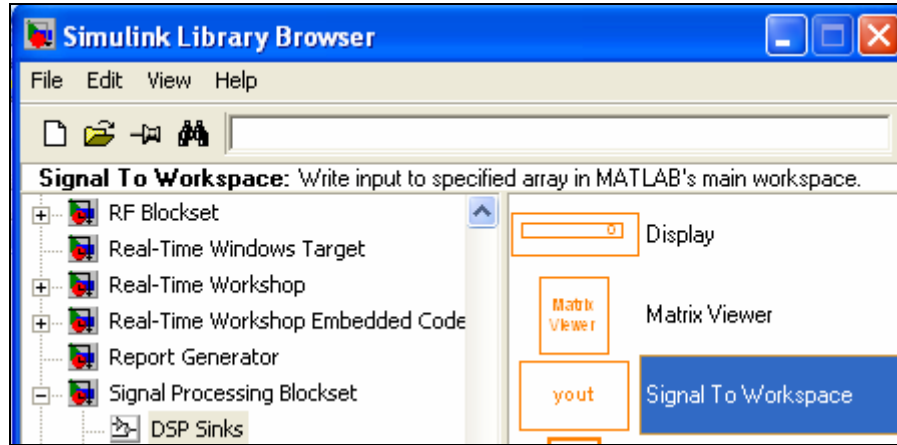
Workspace1 alt programında kullanılacak olan bloklar yönergeler izlenerek bulunur ve tutup taşıma yöntemi ile bloğa eklenirler. Ara bağlantılar gerçekleştirilir. Bkz Şekil 2.18, Şekil 2.19, Şekil 2.20

Simulink Library Browser > Signal Processing Blockset > Platform Specific I/O > Windows (WIN32) > From Wave File

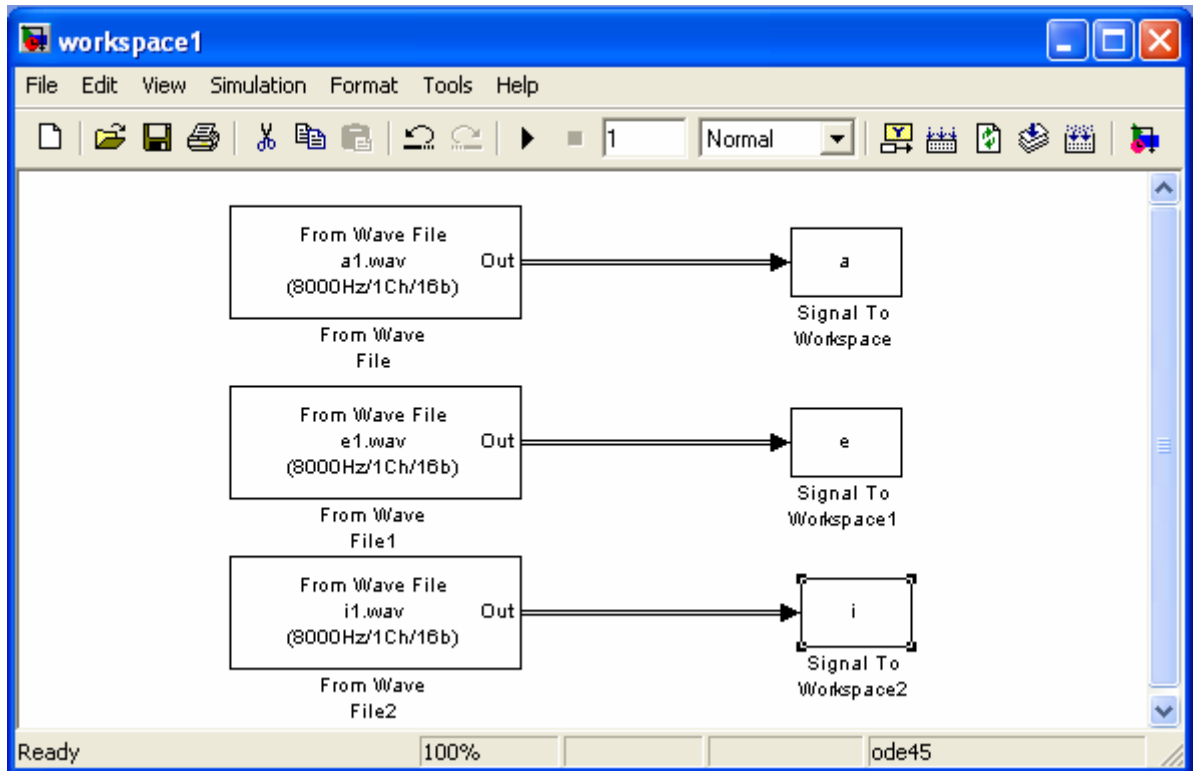


Şekil 2.18 Ses Dosyasının Blok Olarak Atanması

Simulink Library Browser > Signal Processing Blockset > DSP Sinks > Signal To Workspace

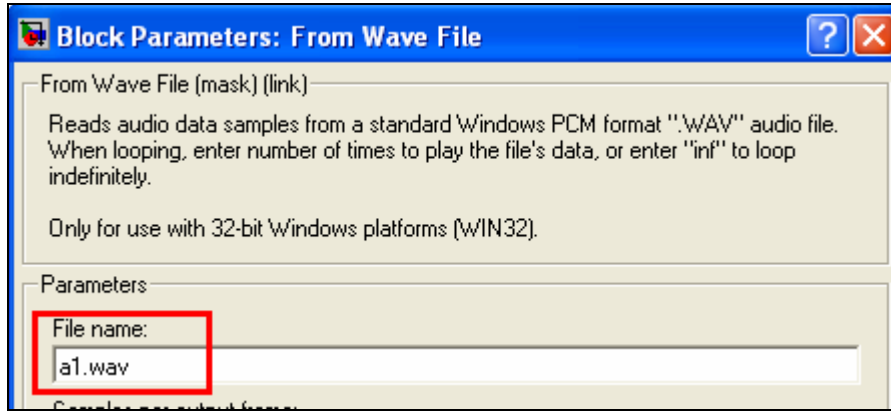


Şekil 2.19 Sinyalin Workspace'e Atılması



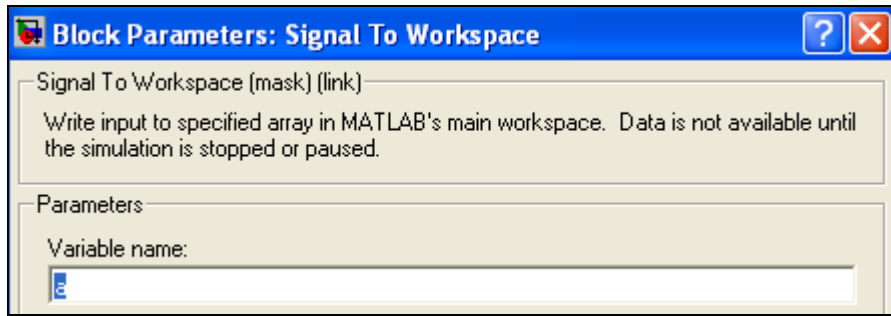
Şekil 2.20 Workspace1 Alt Programı

Workspace1' alt programında From Wave File bloğu çift tıklanarak aşağıdaki pencereye ulaşılır ve File name bölümüne istenilen dosya ismi yazılır.



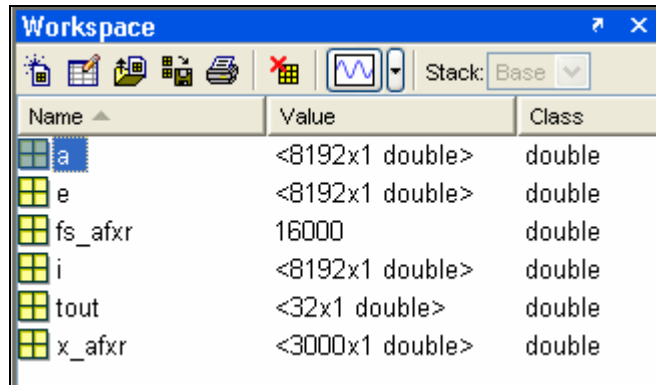
Şekil 2.21 From Wave File Bloğuna Ses Dosyası İsminin Yazılması

Ayrıca Signal to Workspace bloğu çift tıklanarak Şekil 2.22'deki pencereye ulaşılır ve Variable name kısmı istenilen şekilde doldurulur.



Şekil 2.22 Signal To Workspace Bloğuna Vektör İsminin Yazılması

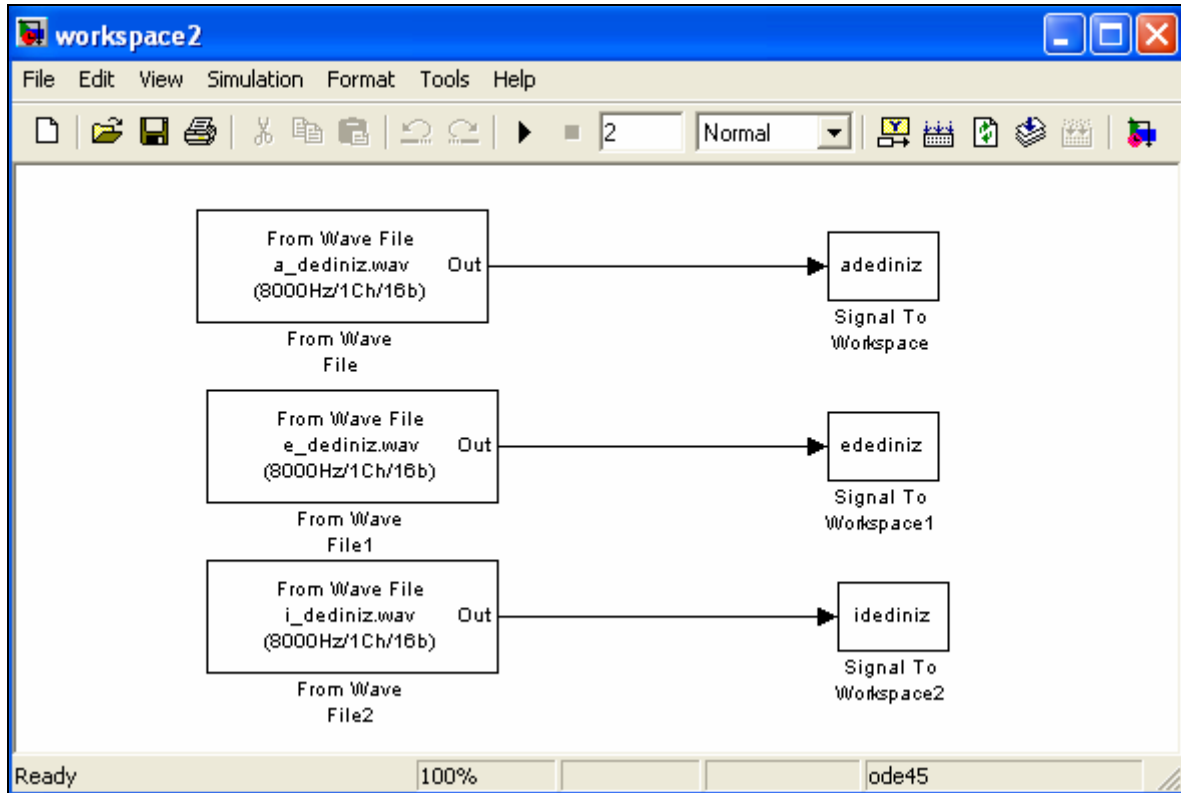
Son olarak 'Workspace1' için çalışma süresi 1sn olarak ayarlanır ve simülasyon gerçekleştirilir. Matlab Workspace Şekil 2.23'deki gibi oluşur.



Şekil 2.23 Matlab Workspace Görünümü

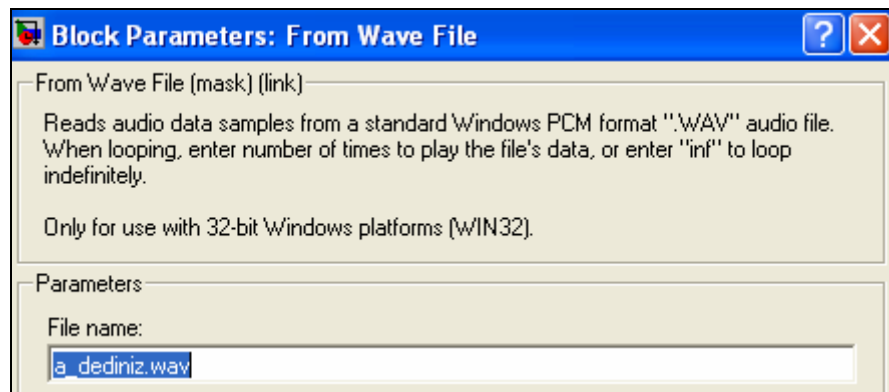
2.6.2 Workspace2 Alt Programının Oluřturulması:

Workspace2 alt programında kullanılacak olan bloklar Bölüm 2.6.1.'de anlatıldığı gibi yönergeler izlenerek bulunur ve tutup taşıma yöntemi ile bloğa eklenir. Ara bağlantılar gerçekleştirilir. Bkz. Şekil 2.24



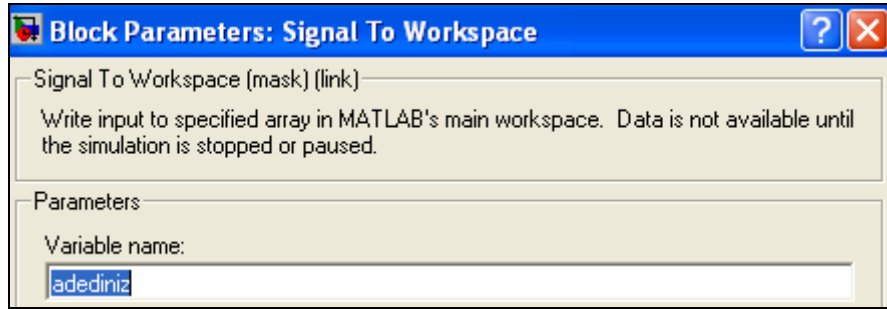
Şekil 2.24 Workspace2 Alt Programı

Workspace1' alt programında From Wave File bloğu çift tıklanarak aşağıdaki pencereye ulaşılır ve File name bölümüne istenilen dosya isimi yazılır.



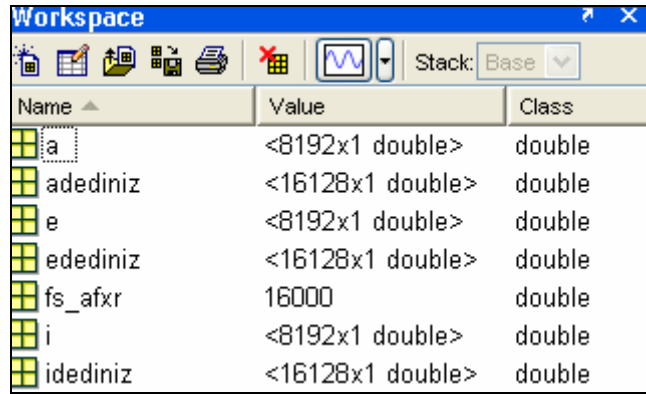
Şekil 2.25 From Wave File Bloğuna Ses Dosyası İsminin Yazılması

Ayrıca Signal to Workspace bloğu çift tıklanarak Şekil 2.26'deki pencereye ulaşılır ve Variable name kısmı istenilen şekilde doldurulur.



Şekil 2.26 Signal To Workspace Bloğuna Vektör İsminin Yazılması

Son olarak 'Workspace2' için çalışma süresi 2sn olarak ayarlanır ve simülasyon gerçekleştirilir. Matlab Workspace Şekil 2.27'deki gibi oluşur.



Şekil 2.27 Matlab Workspace Görünümü

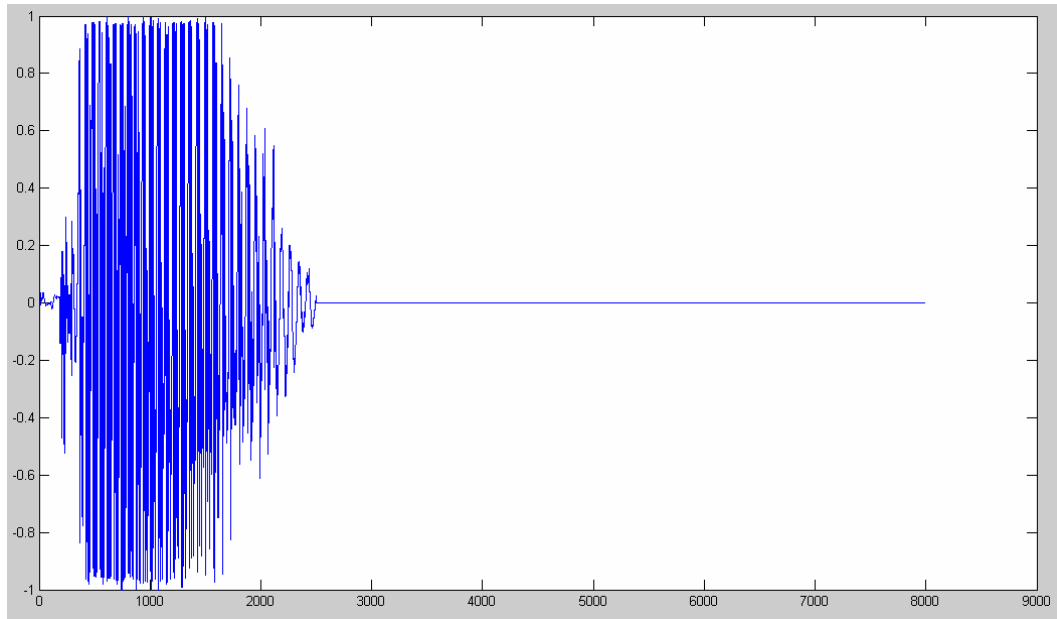
2.7 Kesme Alt Programının Yazılması ve Çalıştırılması:

Kullanıcıya ait ses örnekleri 1sn süresince alınmıştır. Örnekleme frekansı 8 kHz olduğu için 8000 değer içeren bir vektör oluşması beklenir. Fakat kullanılan bloklardaki veri gönderme pencere uzunlukları 256 (Şekil 2.28) değerine ayarlı olduğu için $8000/256$ değeri tam sayı çıkmaz. Bu nedenle işlemin sonucu olan 31,25 sayısının bir üst değeri olan 32 sayısı kullanılır. Böylece veri $32 \times 256 = 8192$ uzunluğundaki bir vektöre dönüşür. Bkz. Şekil 2.27

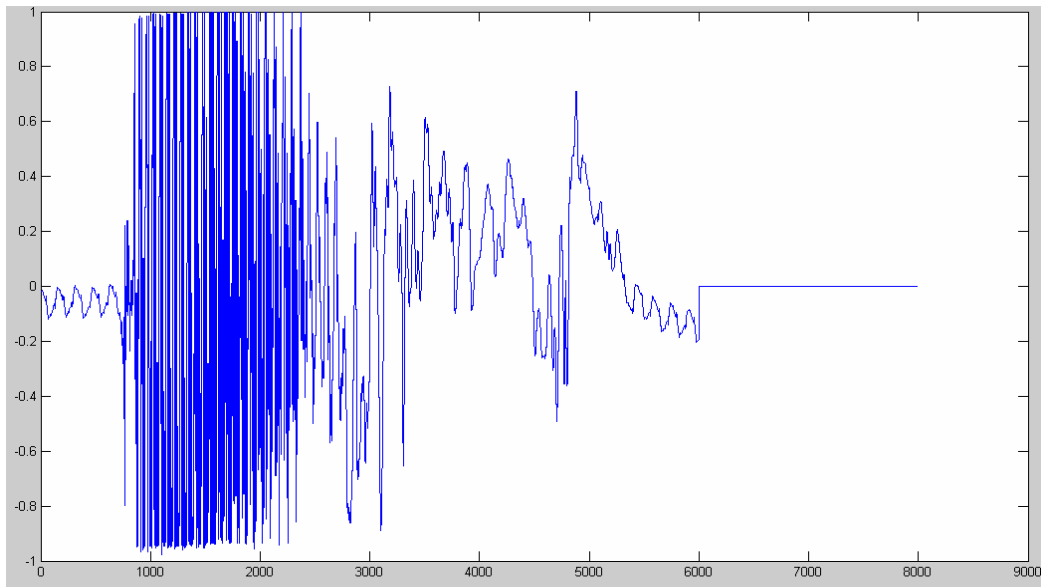


Şekil 2.28 Veri Gönderme Pencere Uzunluğu

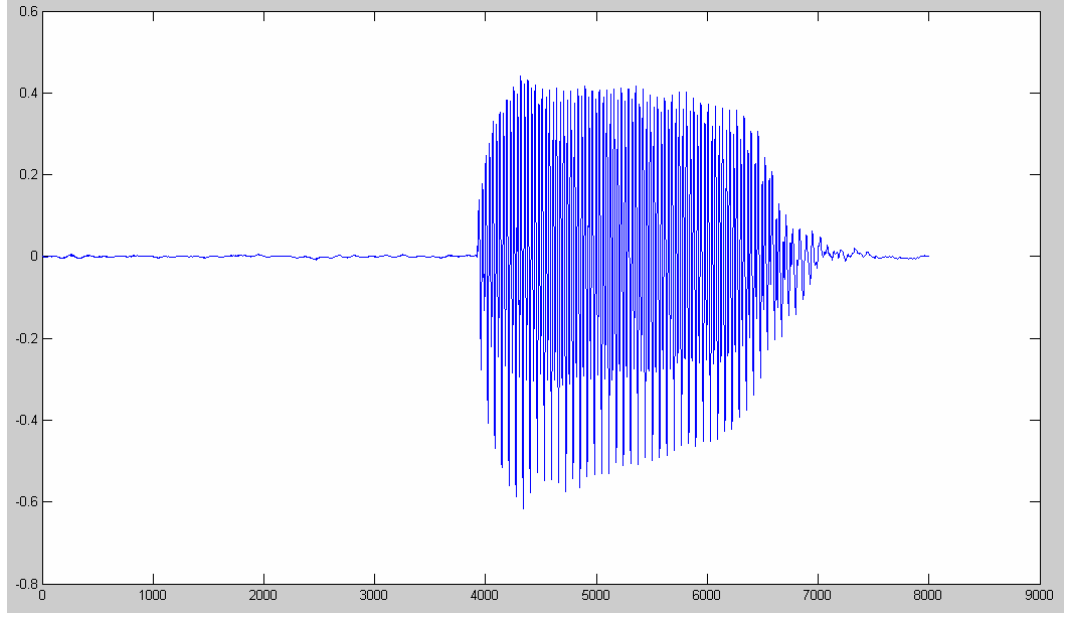
Şekil 2.29, Şekil 2.30 ve Şekil 2.31’da görüldüğü gibi ses sinyalleri 8192 uzunluğundaki vektörün ancak belirli bir kısmında, rasgele bölgelerde bulunmaktadır. Bu nedenle seslerin en yoğun olduğu alanları kapsayan ve uzunluğu 1000 olan bölgeler belirlenir. Burada 1000, C6713 DSK setinin işlem yapabileceği en yüksek değer olan 1024’e en yakın yuvarlak sayı olduğu için seçilmiştir. Uzunluğu 1000 olan gerekli bölgelerin kesilmesi için ‘Kesme.m’ alt programı yazılır. Ses örneklerinin şekilleri incelenerek kesme aralıkları belirlenir. Program gerekli değişikliklerin yapılmasının ardından çalıştırılır.



Şekil 2.29 ‘a’ Sesine Ait Vektör



Şekil 2.30 ‘e’ Sesine Ait Vektör



Şekil 2.31 ‘i’ Sesine Ait Vektör

Sonuç olarak, oluşturulan ve aei>araclar içerisinde kayıt edilen ‘kesme.m’ programı ve açıklaması aşağıdaki gibidir. Bkz Şekil 2.32

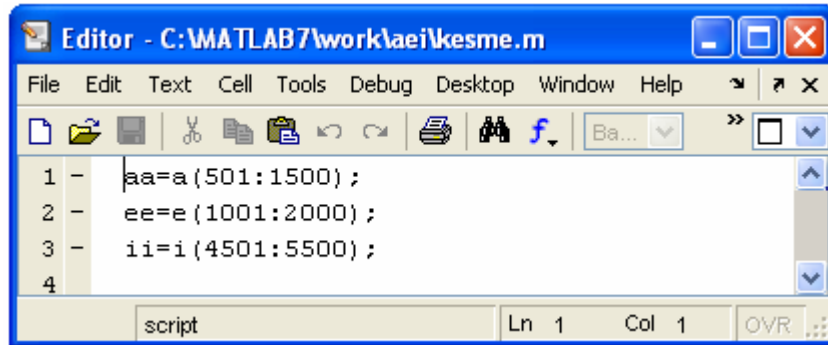
$$X=Y(N1:N2)$$

X: Yeni oluşturulacak 1000 uzunluğundaki matris.

Y: Kesilecek olan 8000 uzunluğundaki matris.

N1: Yeni oluşturulacak matrisin başlangıç noktası.

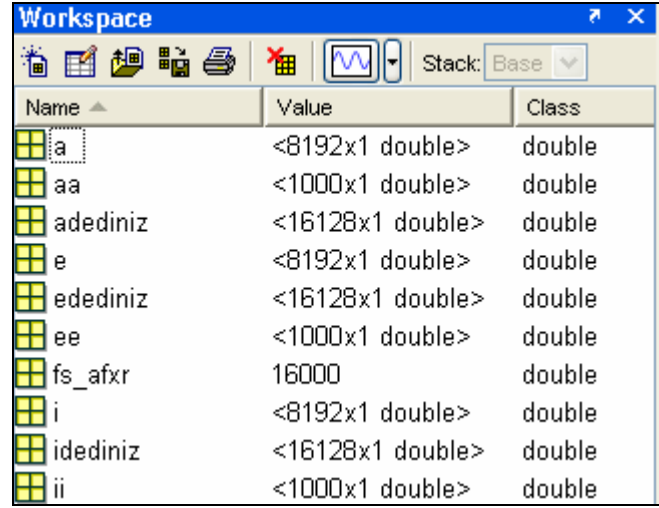
N2: Yeni oluşturulacak matrisin bitiş noktası.



Şekil 2.32 Kesme Alt Programı

Dikkat edilmesi gereken ‘kesme’ alt programı çalıştırıldığında oluşacak olan matrisleri uzunluklarının 1000 değerini aşmamasıdır. Bu sebeple programındaki ilk değerlere 1 sayısı eklenmiştir.

Program çalıştırıldığında Matlab Workspace aşağıdaki gibidir:



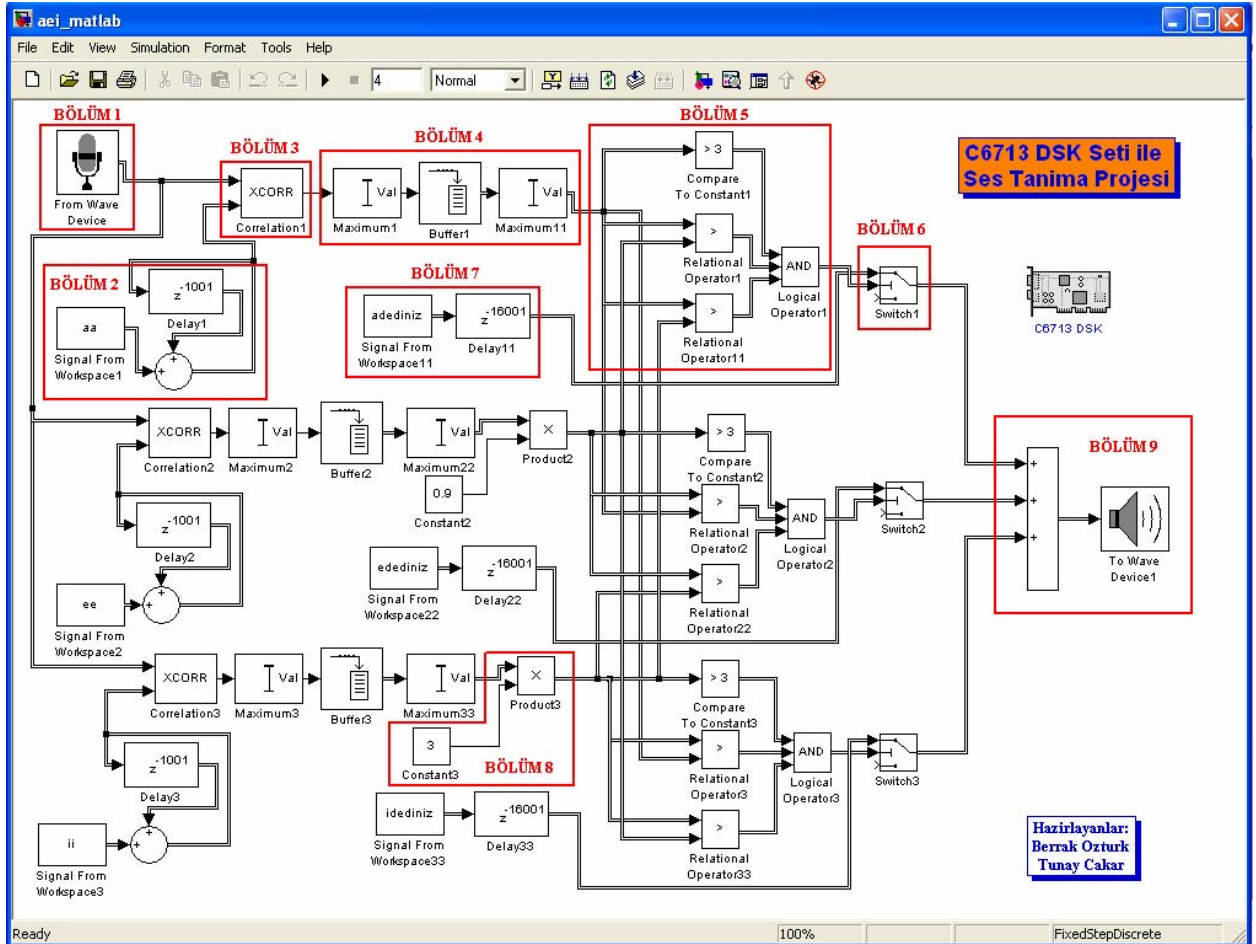
Name	Value	Class
a	<8192x1 double>	double
aa	<1000x1 double>	double
adediniz	<16128x1 double>	double
e	<8192x1 double>	double
edediniz	<16128x1 double>	double
ee	<1000x1 double>	double
fs_afxr	16000	double
i	<8192x1 double>	double
idediniz	<16128x1 double>	double
ii	<1000x1 double>	double

Şekil 2.33 Matlab Workspace Görünümü

‘aa’, ‘ee’ ve ‘ii’ vektörleri; ‘a’, ‘e’ ve ‘i’ seslerinden 1000 uzunluğu dikkate alınarak oluşturulmuştur.

2.8 Matlab Simulink Ortamında Simülasyonun Yapılması:

Matlab Simulink ortamında simülasyonun yapılması için Şekil 2.34'deki ana program, gerekli blokların eklenmesi ve bağlantıların yapılması ile oluşturulur.



Şekil 2.34 'aei_matlab.mdl' Ana Programı

'aei_matlab' adlı ana program 4 saniye süre ile çalıştırılır. İlk 2 saniye de mikrofona gelen giriş sinyali değerlendirilmekte, sonraki 2 saniye de ise değerlendirme sonucuna ait ses dosyası dinlenmektedir.

2.8.1 'aei_matlab.mdl' Ana Programını Oluşturan Bölümler:

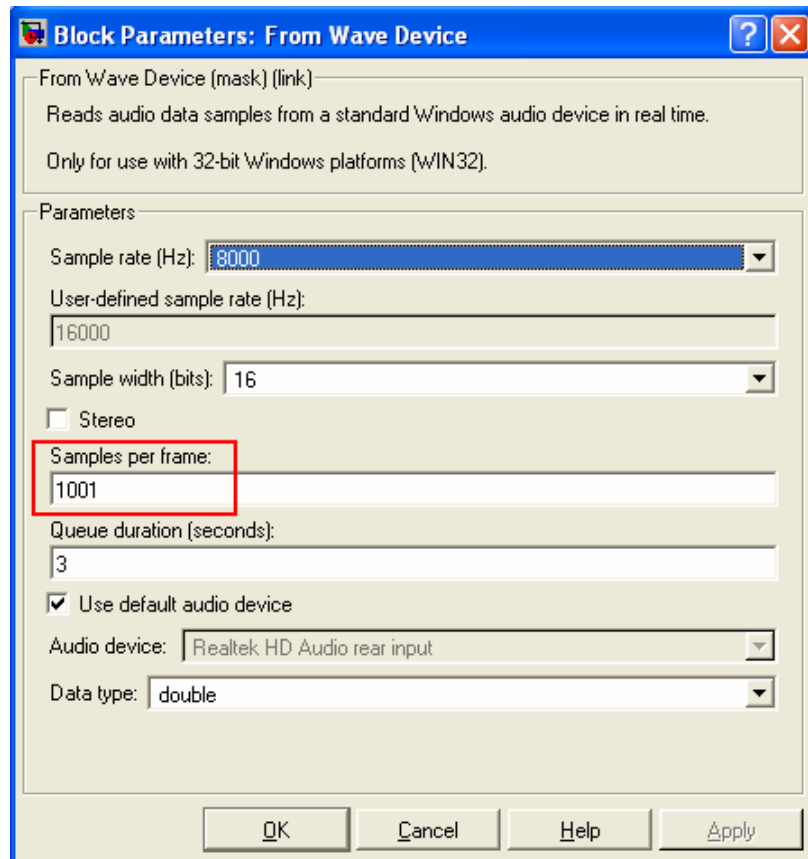
Programın çalışma ilkesinin açıklanması amacı ile diyagram bölümlere ayrılmış ve her bölüme ait özellikler aşağıda açıklanmıştır.

2.8.1.1 Bölüm 1:

Bölüm 2.3’de açıklanan adımlarla mikrofon bloğu programa eklenir. Çift tıklanarak Şekil 2.36’daki pencereye ulaşılır. Samples per frame 1001 değerine ayarlanarak giriş işaretinin 1000 uzunluğundaki vektörler halinde Bölüm 3’teki korelasyon bloğuna gönderilmesi amaçlanır.

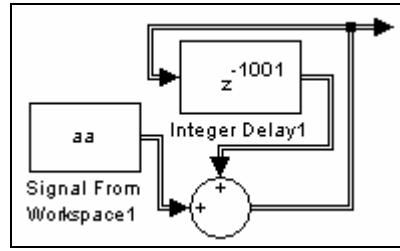


Şekil 2.35 Bölüm 1



Şekil 2.36 Mikrofon Özellikleri

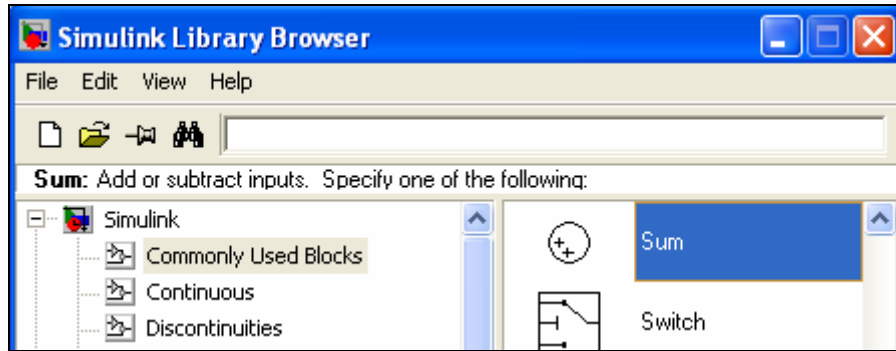
2.8.1.2 Bölüm 2:



Şekil 2.37 Bölüm 2

Toplama bloğu aşağıdaki yönerge takip edilerek programa eklenir. Bkz Şekil 2.38

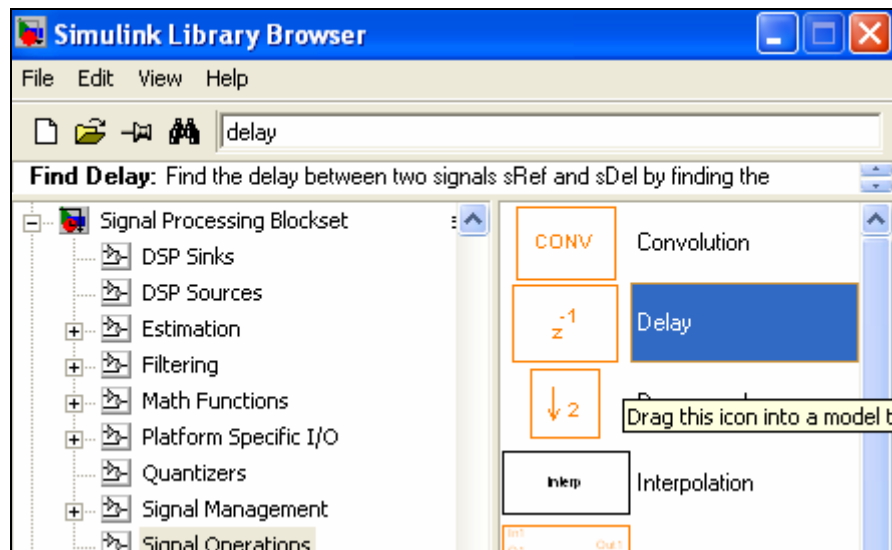
Simulink Library Browser > Simulink> Commonly Used Blocks> Sum



Şekil 2.38 Toplama Bloğu

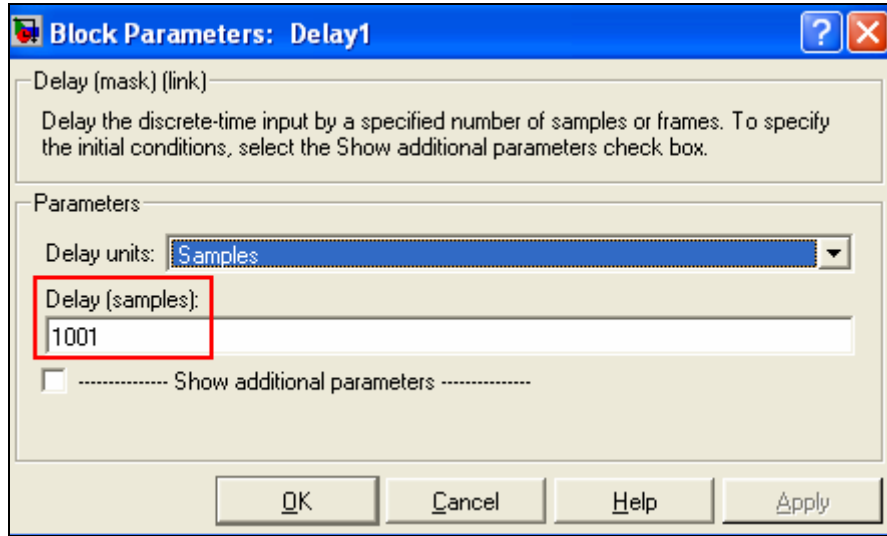
Geciktirme bloğu aşağıdaki yönerge takip edilerek programa eklenir. Bkz Şekil 2.39

Simulink Library Browser > Signal Processing Blockset > Signal Operations > Delay



Şekil 2.39 Geciktirme Bloğu

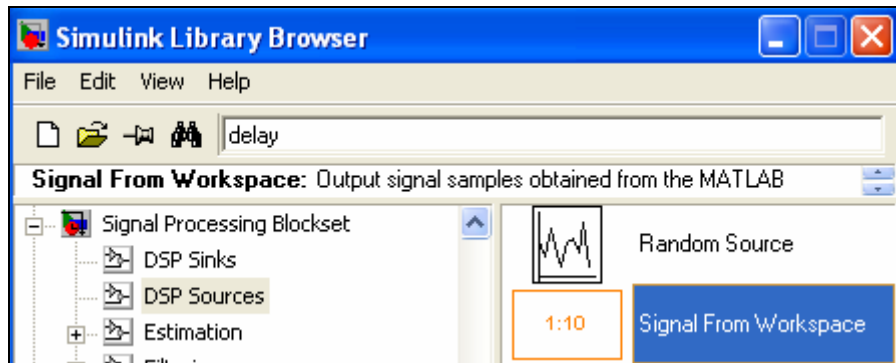
Blok programa eklendikten sonra, çift tıklanarak Şekil 2.40'daki pencereye ulaşılır. Delay (samples) bölümü 1001 olarak değiştirilir.



Şekil 2.40 Geçiktirme Bloğunun Özellikleri

Sinyali workspace'ten alma bloğu aşağıdaki yönerge takip edilerek programa eklenir. Bkz Şekil 2.41

Simulink Library Browser > Signal Processing Blockset > DSP Sources > Signal From Workspace

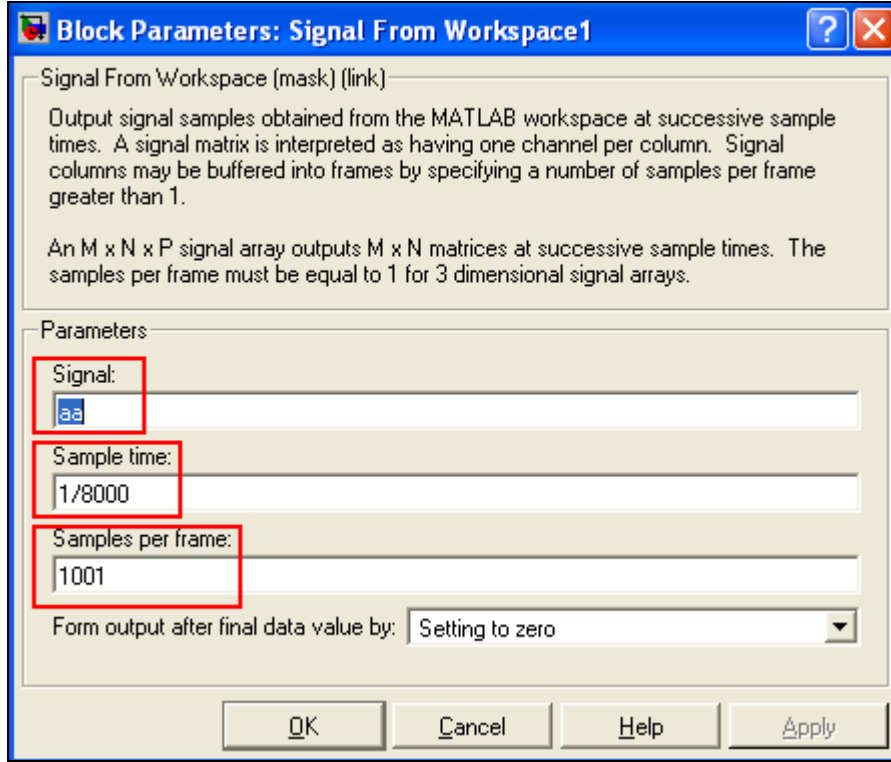


Şekil 2.41 Sinyali Workspace'ten Alma Bloğu

Blok programa eklendikten sonra, çift tıklanarak Şekil 2.42'deki pencereye ulaşılır. Signal bölümü istenilen şekilde değiştirilir.

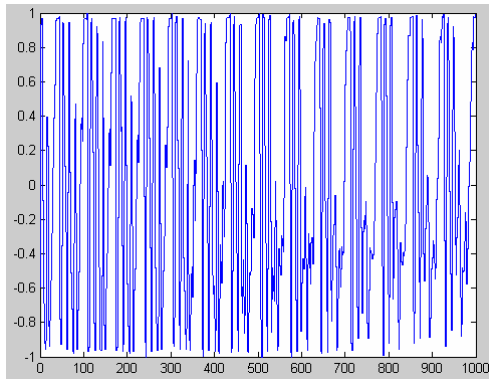
Sample time bölümü 1/8000 olarak değiştirilir.(1/örnekleme frekansı)

Samples per frame bölümü 1001 olarak değiştirilir.(veri gönderme pencere uzunluğu)

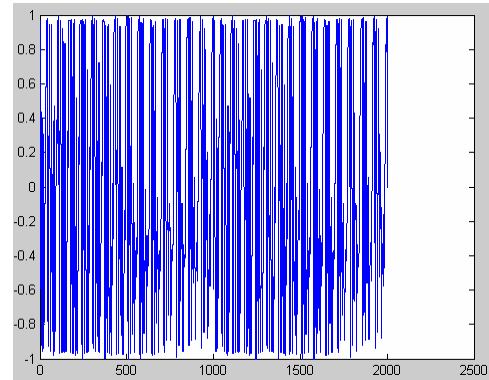


Şekil 2.42 Sinyali Workspace'ten Alma Bloğunun Özellikleri

Geciktirme bloğu ile bir döngü oluşturulmuştur. Örnekleme frekansımız 8 kHz olduğundan 1000 boyutlu matrisin simülasyon süresi $1000/8000=0.125$ sn'dir. Ancak kullanıcı 2sn'lik simülasyon süresi içinde herhangi bir zaman aralığında konuşabileceği için, korelasyon yapılacak olan ses örneklerinin sürekli olmaları sağlanmıştır. Burada kesilmiş ses örnek vektörlerinin uzunluğu 1000 olduğundan geciktirme bloğu içerisine 1001 değeri yazılarak sesin kesintisiz döngüsü sağlanmıştır.



Kesilmiş Ses Örneği



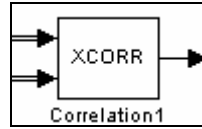
1.Döngü sonrası elde edilen Sinyal

Şekil 2.43

Şekil 2.43'te kullanıcıya ait kesilmiş 1000 değerli ses örneği ve bunun Bölüm 2'de 1.döngüsünü tamamladıktan sonra oluşmuş 2000 değerli hali gözlemlenmektedir. Birinci döngü sonunda geçen süre $2 \times 0.125 = 0.25$ sn'dir. 2 sn'lik simülasyon sürecinde 15 döngü gerçekleşir.

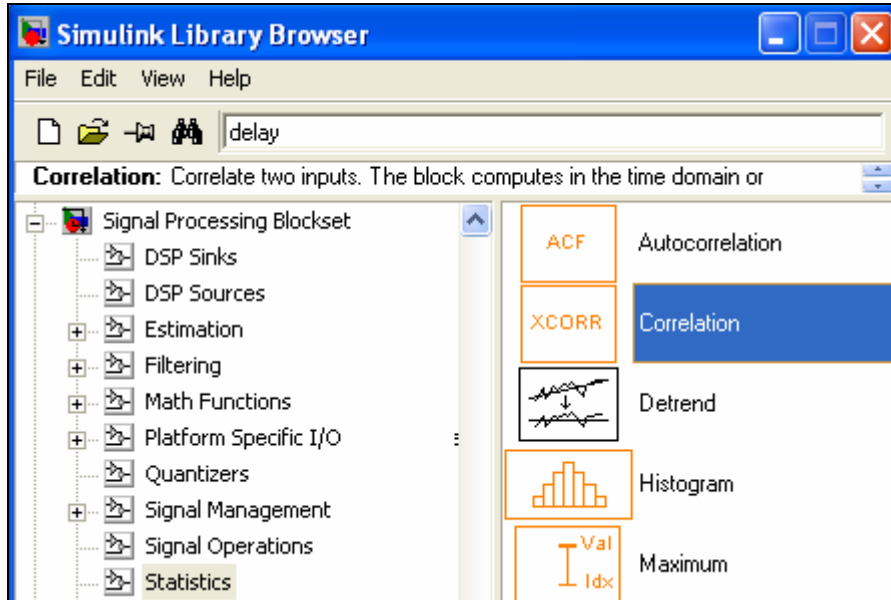
Sonuç olarak Bölüm 2'de kesme işleminden sonra elde edilen kullanıcıya ait ses örneklerinin sürekli olarak tekrarlanarak Bölüm 3'e gönderilmesi sağlanmıştır.

2.8.1.3 Bölüm 3:



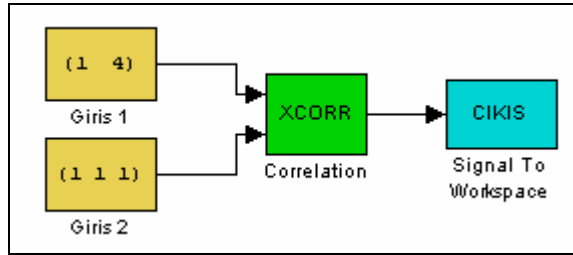
Şekil 2.44 Bölüm 3

Korelasyon bloğu aşağıdaki yönerge takip edilerek programa eklenir. Bkz Şekil 2.45
Simulink Library Browser > Signal Processing Blockset > Statistics > Correlation

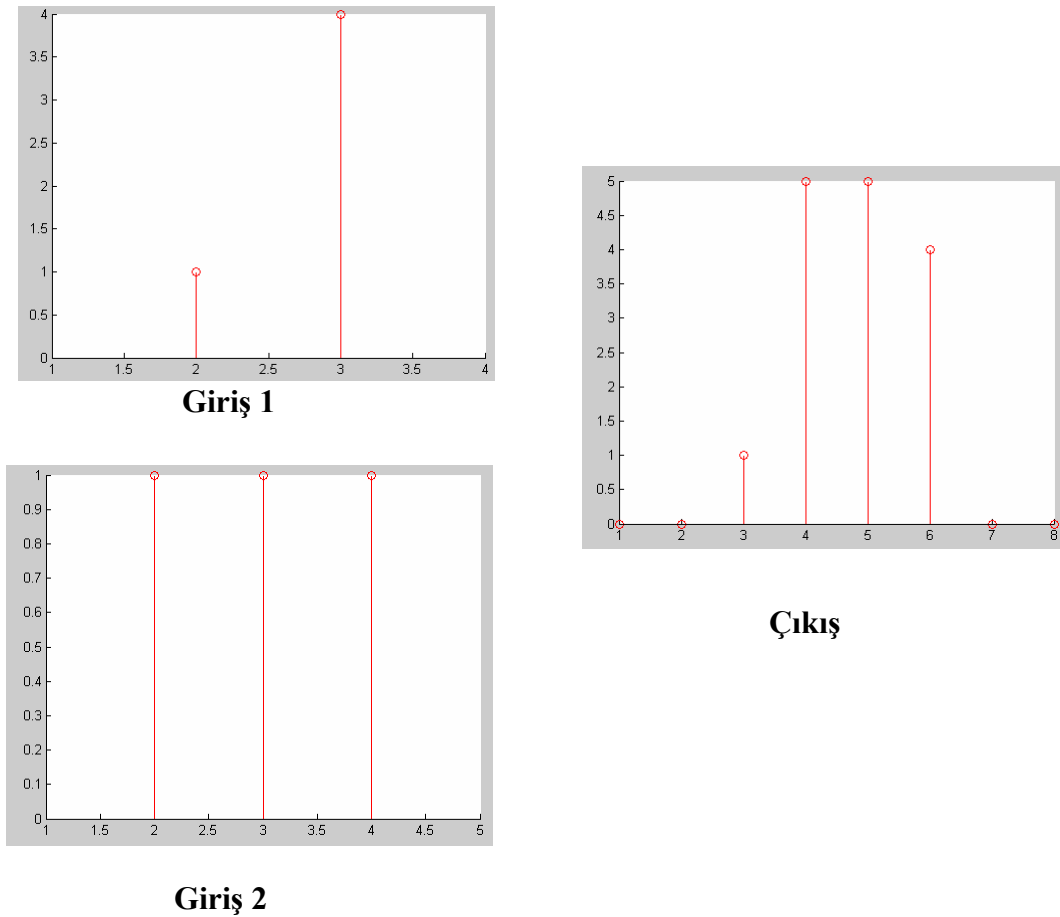


Şekil 2.45 Korelasyon Bloğu

Blok, girişlerine gelen iki sinyali korele ederek çıkışa göndermektedir. Aşağıda bloğun yaptığı işlem basit bir örnek ile gösterilmiştir.



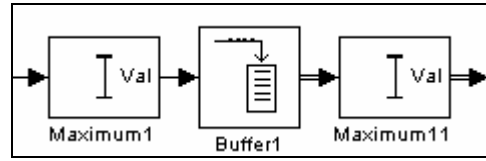
Şekil 2.46 Korelasyon Örneği



Şekil 2.47 Korelasyon Örneği Sonuçları

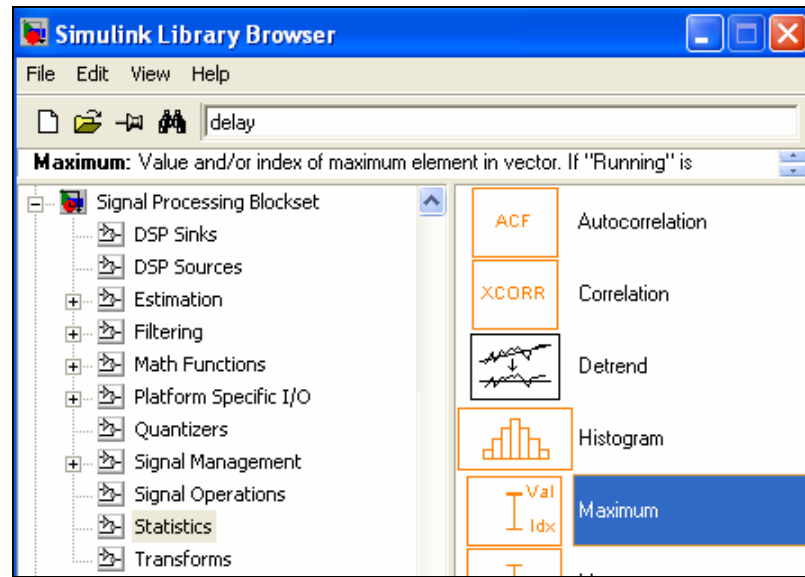
Korelasyon bloğu ana programın en önemli parçasıdır. Çünkü bu blok girişten gelen, kullanıcının mikrofona söylediği ‘a’ , ‘e’ ve ‘i’ seslerine ait 1000 değerli vektör ile önceden kayıt edilmiş 1000 değerli ses örnek vektörlerini korele eder. Sonuç ise 2000 değerli vektör olarak Bölüm 4’e gönderilir.

2.8.1.4 Bölüm 4:



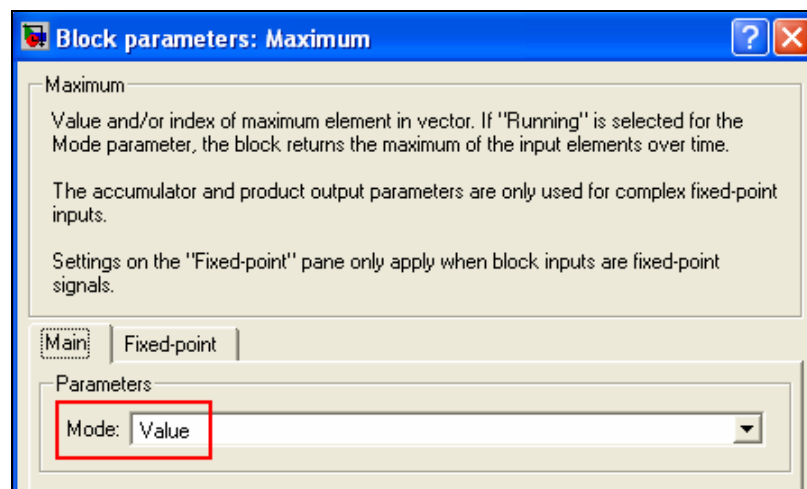
Şekil 2.48 Bölüm 4

Maksimum bloğu aşağıdaki yönerge takip edilerek programa eklenir. Bkz Şekil 2.49
Simulink Library Browser > Signal Processing Blockset > Statistics > Maximum



Şekil 2.49 Maksimum Bloğu

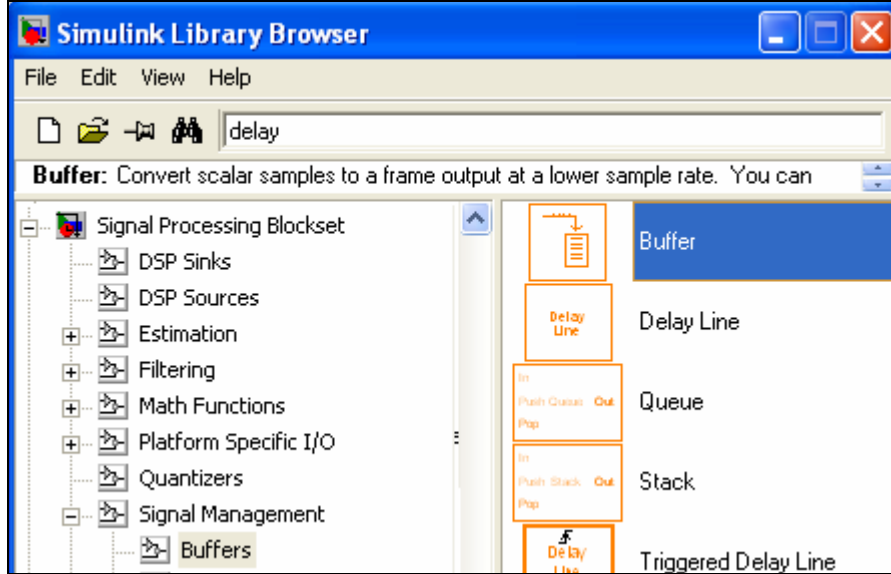
Blok programa eklendikten sonra, çift tıklanarak Şekil 2.50’teki pencereye ulaşılır.
Mode bölümü Value olarak değiştirilir.



Şekil 2.50 Maksimum Bloğu Özellikleri

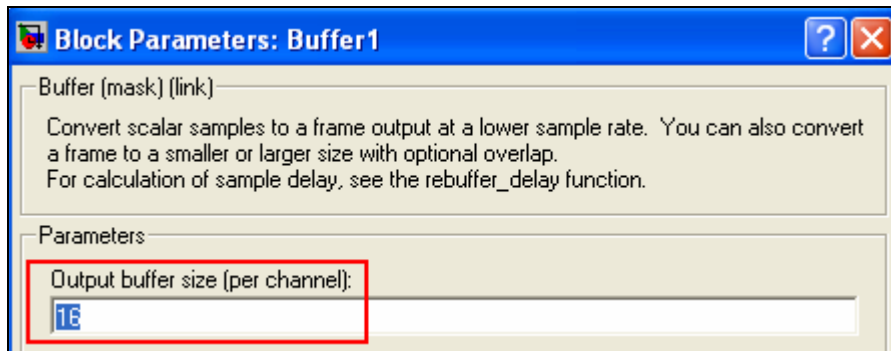
Buffer bloğu aşağıdaki yönerge takip edilerek programa eklenir. Bkz Şekil 2.51

Simulink Library Browser > Signal Processing Blockset > Signal Management > Buffers > Buffer



Şekil 2.51 Buffer Bloğu

Blok programa eklendikten sonra, çift tıklanarak Şekil 2.52'deki pencereye ulaşılır. Output buffer size (per channel) bölümüne 16 değeri girilir.



Şekil 2.52 Buffer Bloğu Özellikleri

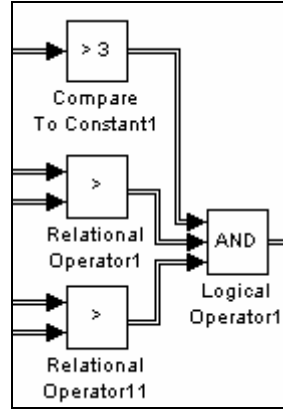
1.Maksimum bloğu Bölüm 3'ten gelen 0.125 saniyelik ve 2000 değere sahip vektöre ait maksimum genliği seçerek Buffer'a gönderir. 2 sn'lik simülasyon süresince 16 adet maksimum değer elde edilir. Bu sebeple Buffer boyutu 16 olarak ayarlanır. 2 sn'nin sonunda son değer de Buffer'a geldiğinde tüm değerler aynı anda 2.Maksimum bloğuna gönderilir. 2.Maksimum bloğuda bu 16 değer arasından en büyüğünü seçerek Bölüm 5'e gönderir.

	1	2
1	0.19007	
2	0.20688	
3	0.22576	
4	0.17422	
5	0.1991	
6	5.9492	
7	32.119	
8	23.07	
9	1.6575	
10	0.40042	
11	0.16025	
12	0.22127	
13	0.11567	
14	0.20208	
15	0.2317	
16	0.16076	

Örnek olarak mikrofona kullanıcı tarafından söylenen ‘a’ sinyalinin 1.Maksimum bloğu çıkışında elde edilen vektör ve 2.Maksimum bloğu çıkışında elde edilen değer Şekil 2.53’de gösterilmiştir.

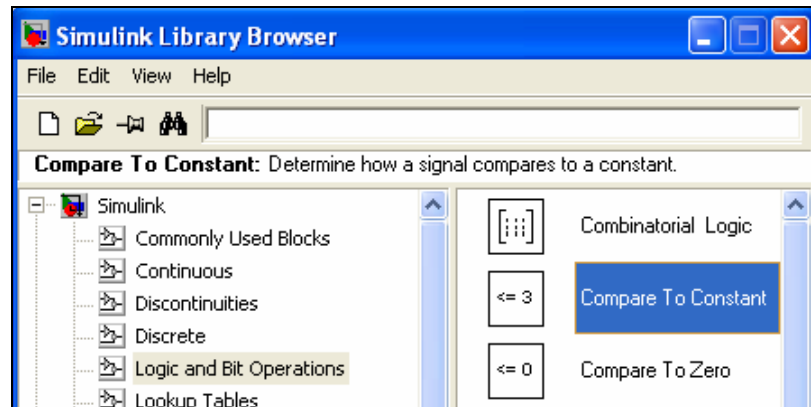
Şekil 2.53 1.Maksimum Bloğu Çıkış Matrisi

2.8.1.5 Bölüm 5:



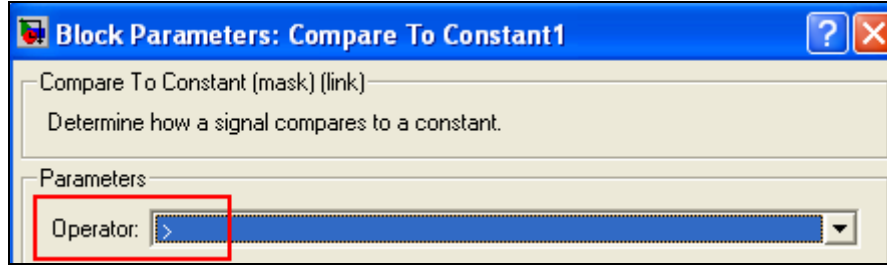
Şekil 2.54 Bölüm 5

Sabit ile karşılaştırma bloğu aşağıdaki yönerge takip edilerek programa eklenir. Bkz Şekil 2.55
Simulink Library Browser > Simulink > Logic and Bit Operations > Compare To Constant



Şekil 2.55 Sabit ile Karşılaştırma Bloğu

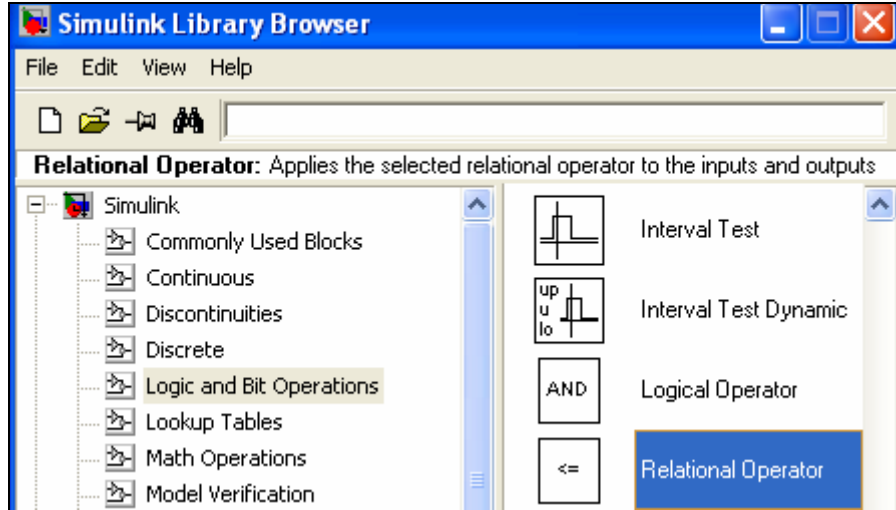
Blok programa eklendikten sonra, çift tıklanarak Şekil 2.56'daki pencereye ulaşılır. Operator bölümündeki ' \leq ' sembolü ' $>$ ' ile değiştirilir.



Şekil 2.56 Sabit ile Karşılaştırma Bloğu Özellikleri

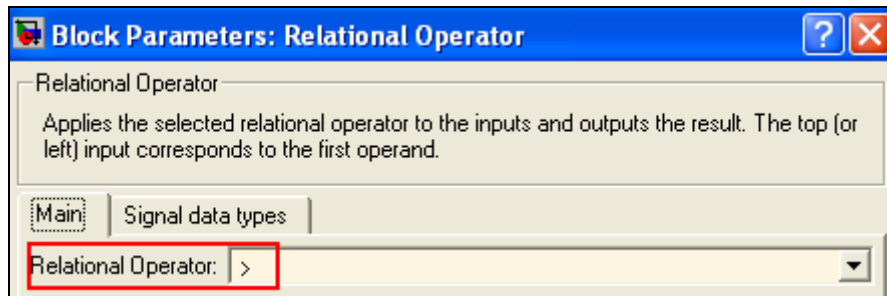
Karşılaştırma bloğu aşağıdaki yönerge takip edilerek programa eklenir. Bkz Şekil 2.57

Simulink Library Browser > Simulink > Logic and Bit Operations > Relational Operator



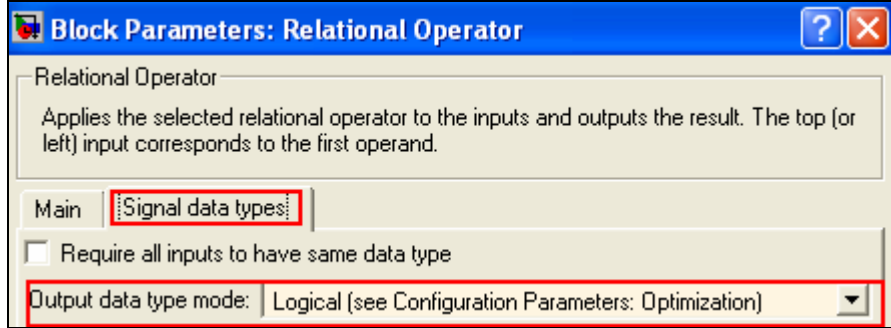
Şekil 2.57 Karşılaştırma Bloğu

Blok programa eklendikten sonra, çift tıklanarak Şekil 2.58.a'daki pencereye ulaşılır. Relational Operator bölümündeki ' \leq ' sembolü ' $>$ ' ile değiştirilir.



Şekil 2.58.a Karşılaştırma Bloğu Özellikleri

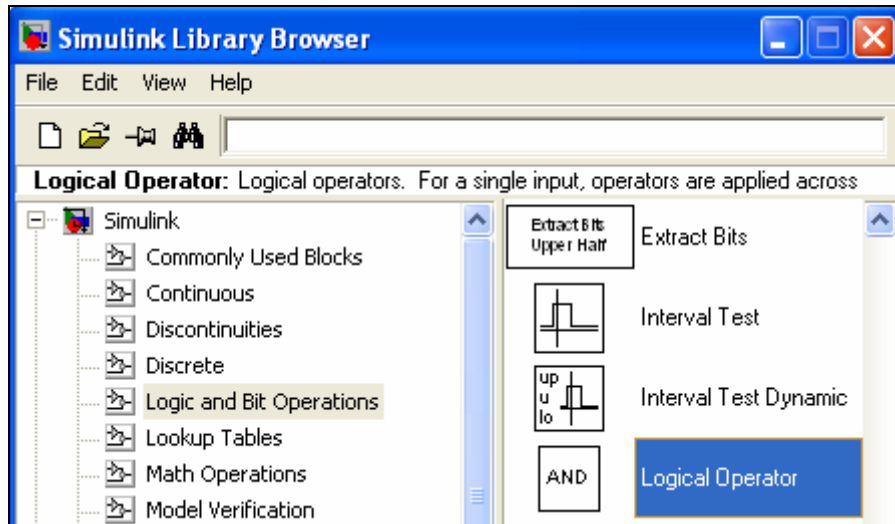
Signal data types bölümüne geçilerek Output data type mode seçeneği Logical olarak değiştirilir. Bkz. Şekil 2.58.b



Şekil 2.58.b Karşılaştırma Bloğu Özellikleri

VE kapısı bloğu aşağıdaki yönerge takip edilerek programa eklenir. Bkz Şekil 2.59

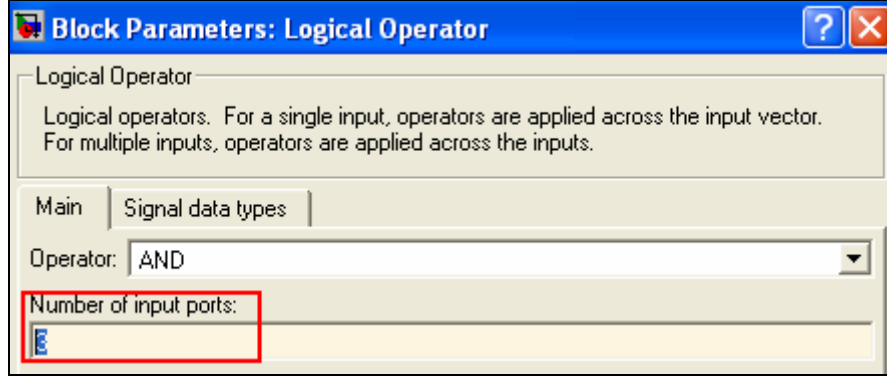
Simulink Library Browser > Simulink > Logic and Bit Operations > Logical Operator



Şekil 2.59 VE Kapısı Bloğu

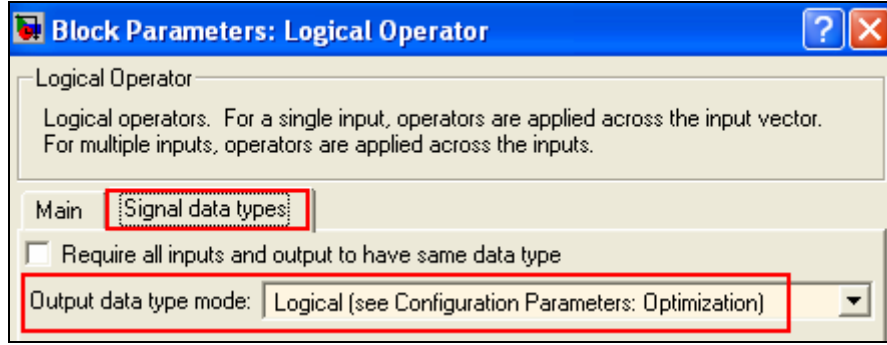
Blok programa eklendikten sonra, çift tıklanarak Şekil 2.60.a'daki pencereye ulaşılır.

Number of input ports bölümü 3 girişe ayarlanır.

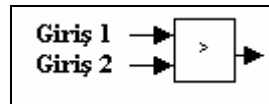


Şekil 2.60.a VE Kapısı Bloğu Özellikleri

Signal data types bölümüne geçilerek Output data type mode seçeneği Logical olarak değiştirilir. Bkz. Şekil 2.60.b



Şekil 2.60.b VE Kapısı Bloğu Özellikleri

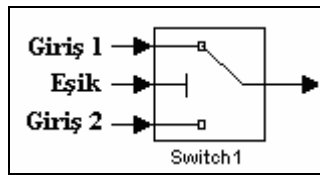


Şekil 2.61 Karşılaştırma Mantığı

Karşılaştırma bloğu, Giriş 1'e gelen işareti, Giriş 2 ile karşılaştırarak lojik sonucu çıkışa gönderir. Bölüm 5'teki karşılaştırma bloklarından biri, Bölüm 4'ten gelen 'a' sesine ait korelasyon sonucu ile 'e' sesine ait korelasyon sonucunu, diğeri ise 'i' sesine ait korelasyon sonucunu karşılaştırarak And kapısına gönderir. Kullanıcı tarafından mikrofona 'a' sesi gönderildiğinde Bölüm 5'teki karşılaştırma bloklarının çıkışları lojik 1 olacaktır. Ancak giriş işareti gönderilmese de ortamdaki gürültü sebebiyle Bölüm 4 çıkışlarında bir değer gözlemlenir. Bu değer Bölüm 5'teki karşılaştırma bloklarını yanıltarak simülasyonun, sanki mikrofona bir giriş işareti gönderilmiş gibi çalışmasını sağlar ve o sese ait çıkış ses dosyası hoparlörde dinlenir.

Bunu engellemek amacı ile sabit ile karşılaştırma bloğu kullanılarak, simülasyonun sadece girişe işaret geldiğinde çalışması sağlanmıştır. Bu amaçla girişe sinyal gönderilmeden yapılan çalışmalar sonucunda, Bölüm 4 çıkışlarının 3 değerini aşmadığı gözlemlenmiş ve bu sayı eşik değer olarak seçilmiştir. Sonuçta girişe ‘a’ sesi gönderildiğinde sabit ile karşılaştırma bloğu ve karşılaştırma blokları çıkışları And kapısına Lojik 1 işaretini gönderir, bu kapı da Bölüm 6’ya lojik 1 işareti gönderir.

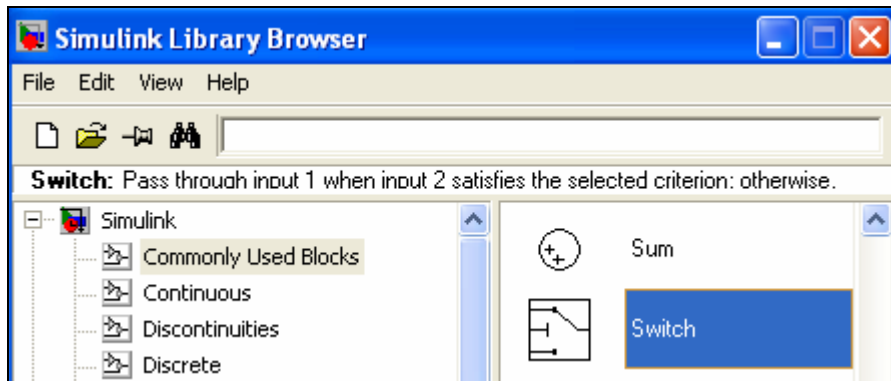
2.8.1.6 Bölüm 6:



Şekil 2.62 Bölüm 6

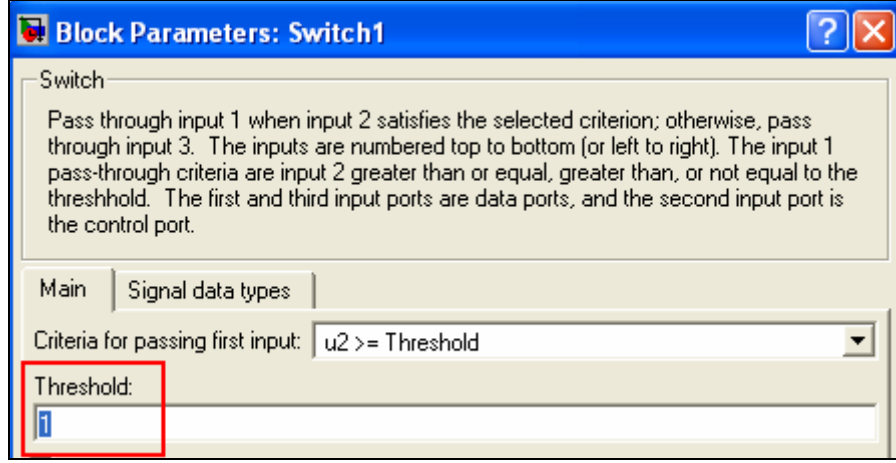
Anahtar bloğu aşağıdaki yönerge takip edilerek programa eklenir. Bkz Şekil 2.63

Simulink Library Browser > Simulink > Commonly Used Blocks > Switch



Şekil 2.63 Anahtar Bloğu

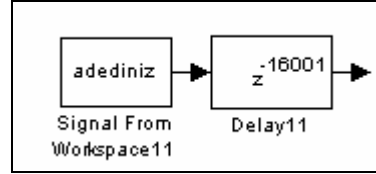
Blok programa eklendikten sonra, çift tıklanarak Şekil 2.64’deki pencereye ulaşılır. Threshold bölümü 1 olarak değiştirilir.



Şekil 2.64 Anahtar Bloğu Özellikleri

Anahtar bloğu, eşik değerine gerekli sinyal gönderilmedikçe Giriş 2'yi, eşik değer şartı sağlandığında Giriş 1'i çıkışına gönderir.

2.8.1.7 Bölüm 7:

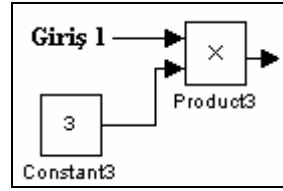


Şekil 2.65 Bölüm 7

Sinyali workspace'ten alma ve Geciktirme bloklarının programa eklenmesi ve gerekli değişikliklerin yapılması için ilgili yönergeler Bölüm 2.8.1.2' de anlatıldığı gibidir.

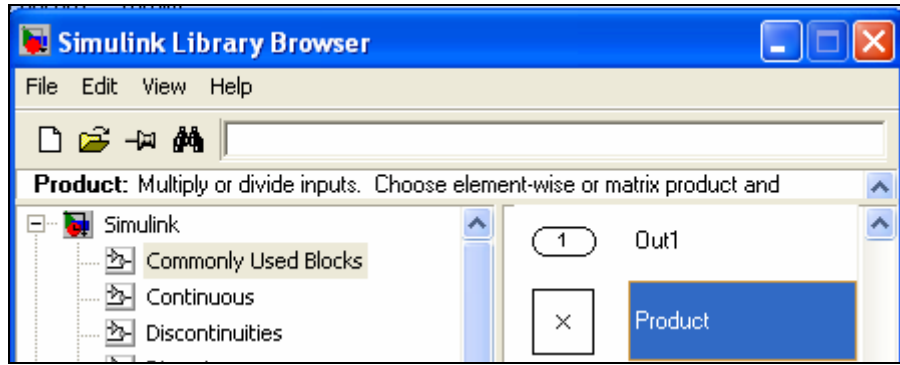
Burada sinyali workspace'ten alma bloğu kullanıcının mikrofona söylediği giriş sinyaline ait çıkış ses dosyasını içermektedir. Ancak çıkış ses dosyasının simülasyon başlangıcından 2sn sonra duyulması istenir. Bu süreye tekabül eden 16001 değeri geciktirme bloğuna yazılarak gerekli geciktirme sağlanmış olur.

2.8.1.8 Bölüm 8:



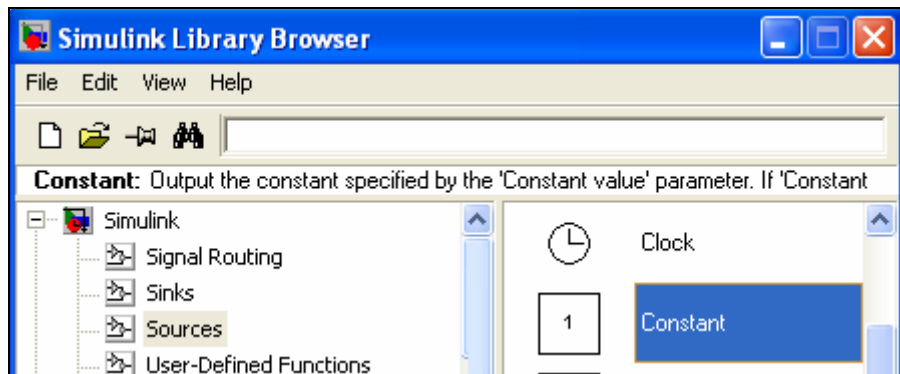
Şekil 2.66 Bölüm 8

Çarpma bloğu aşağıdaki yönerge takip edilerek programa eklenir. Bkz Şekil 2.67
Simulink Library Browser > Simulink > Commonly Used Blocks > Product



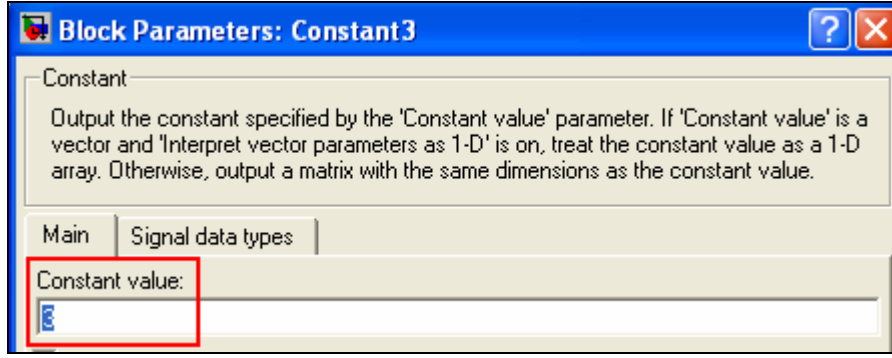
Şekil 2.67 Çarpma Bloğu

Sabit bloğu aşağıdaki yönerge takip edilerek programa eklenir. Bkz Şekil 2.68
Simulink Library Browser > Simulink > Sources > Constant



Şekil 2.68 Sabit Bloğu

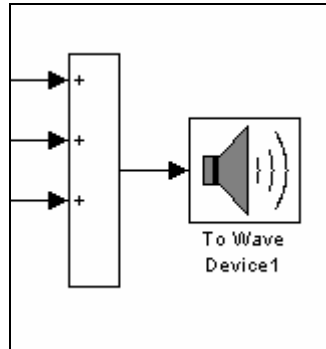
Blok programa eklendikten sonra, çift tıklanarak Şekil 2.69'daki pencereye ulaşılır.
Constant value bölümü gerekli değere ayarlanır.



Şekil 2.69 Sabit Bloğu Özellikleri

Simülasyon ile yapılan denemelerde ‘a’ sesinin, ‘e’ sesi ile korelasyon sonucunun zaman zaman ‘a’ sesinin kendisiyle olan korelasyon sonucunu geçtiği, ‘i’ sesinin ise kendisiyle olan korelasyon sonuçlarının ‘a’ ve ‘e’ sesleriyle olan korelasyon sonuçlarından düşük kaldığı gözlemlenmiştir. Bu sorunları aşmak amacı ile ‘e’ sesine ait Bölüm 4 çıkışına Bölüm 8’in eklenmesiyle zayıflatma, ‘i’ sesine ait Bölüm 4 çıkışına da Bölüm 8’in eklenmesiyle kuvvetlendirme sağlanmıştır.

2.8.1.9 Bölüm 9:



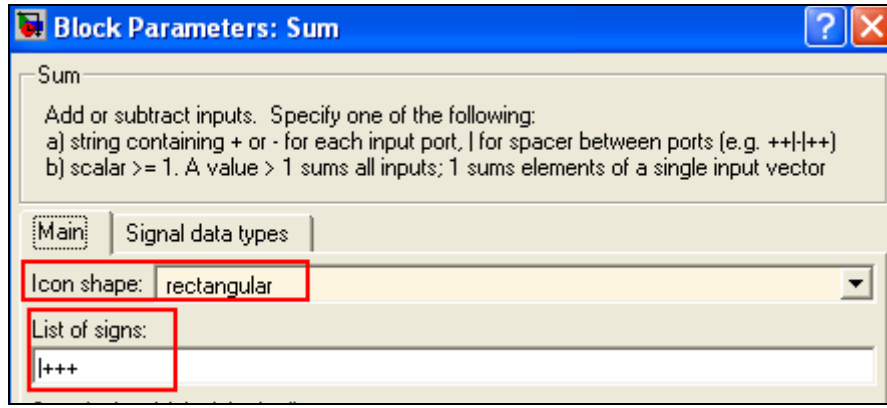
Şekil 2.70 Bölüm 9

Toplama bloğu bölüm 2.8.1.2’de anlatıldığı şekilde programa eklenir.

Blok programa eklendikten sonra, çift tıklanarak Şekil 2.71’deki pencereye ulaşılır.

List of signs bölümüne bir ‘+’ işareti daha eklenerek giriş sayısının üç olması sağlanır.

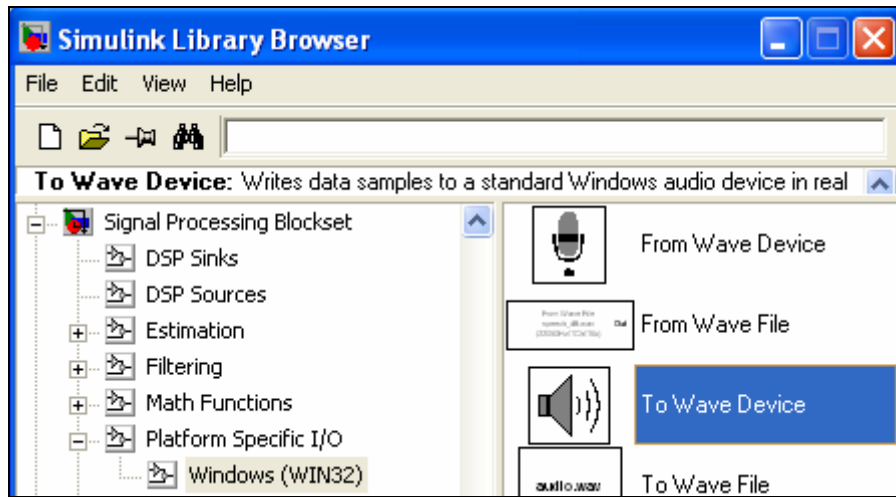
Icon shape özelliği ise rectangular olarak değiştirilerek karesel görünüm sağlanır.



Şekil 2.71 Toplama Bloğu Özellikleri

Hoparlör bloğu aşağıdaki yönerge takip edilerek programa eklenir. Bkz Şekil 2.72

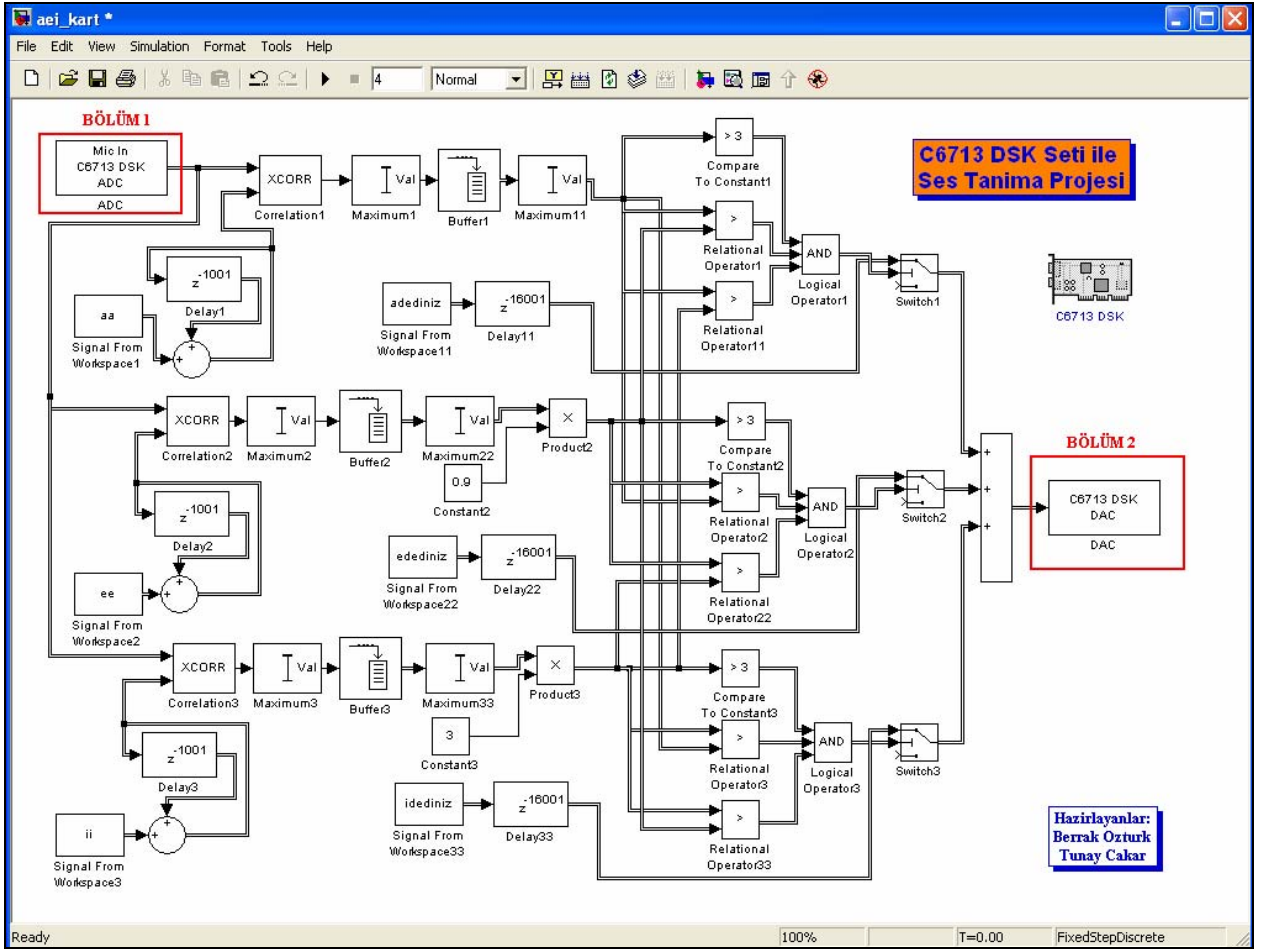
Simulink Library Browser > Signal Processing Blockset > Platform Specific I/O > Windows (WIN32) > To Wave Device



Şekil 2.72 Hoparlör Bloğu

Burada toplama bloğu Bölüm 6 çıkışından gelen, Bölüm 7’de bulunan çıkış ses dosyalarını hoparlör bloğuna göndermek için kullanılmıştır. Simülasyon sonucu olarak yalnızca bir ses dosyası toplama bloğuna ulaşabileceğinden diğer girişler sıfır olacaktır. Bu sebeple gerçekte hiçbir zaman iki veya daha fazla sinyali toplama işlemi yapmamaktadır.

2.9 C6713 DSK Kartı Üzerinde Simülasyonun Gerçekleştirilmesi:



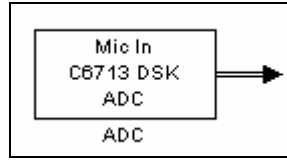
Şekil 2.73 'aei_kart.mdl' Ana Programı

'aei' klasörü içerisindeki 'aei_matlab.mdl' adlı ana program yine aynı klasör içerisine 'aei_kart.mdl' adı ile farklı kaydedilerek, içerisinde bulunan Matlab Simulink programının mikrofon ve hoparlör blokları C6713 DSK'ye ait ADC (analog-dijital çevirici), DAC (dijital-analog çevirici) blokları ile değiştirilir. Bağlantıların ve ayarların yapılmasından sonra elde edilen ana program Şekil 2.73'deki gibidir. Alt programların çalıştırılması ile verilerin yüklenmesinden sonra elde edilen program C6713 DSK setine yüklenerek simülasyon bu ortamda gerçekleştirilir.

2.9.1 'aei_kart.mdl' Ana Programını Oluşturan Bölümler:

Programın çalışma ilkesinin açıklanması amacı ile diyagram bölümlere ayrılmış ve her bölüme ait özellikler aşağıda açıklanmıştır.

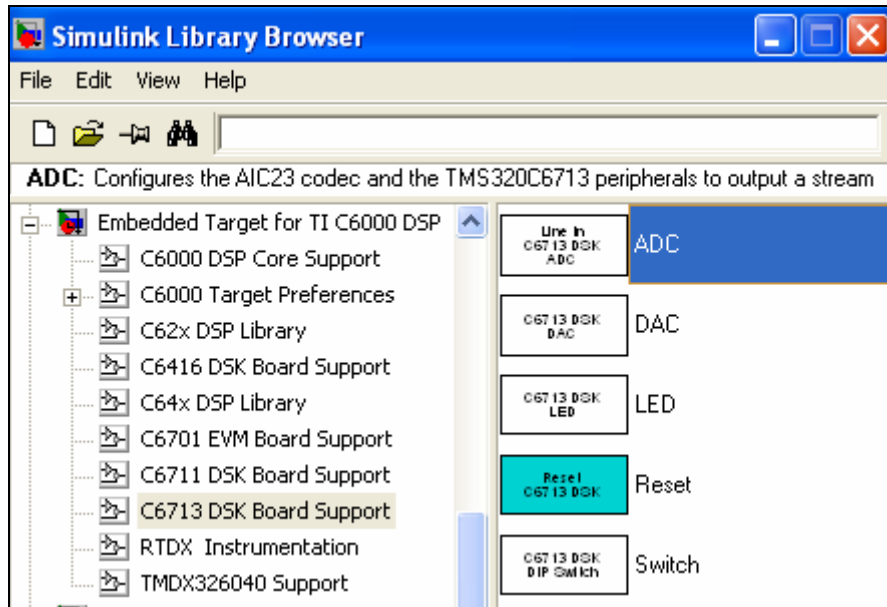
2.9.1.1 Bölüm 1:



Şekil 2.74 Bölüm 1

ADC bloğu aşağıdaki yönerge takip edilerek programa eklenir. Bkz Şekil 2.75

Simulink Library Browser > Embedded Target fot TI C6000 DSP > C6713 DSK Board Support > ADC



Şekil 2.75 ADC Bloğu

Blok programa eklendikten sonra, çift tıklanarak Şekil 2.76'daki pencereye ulaşılır.

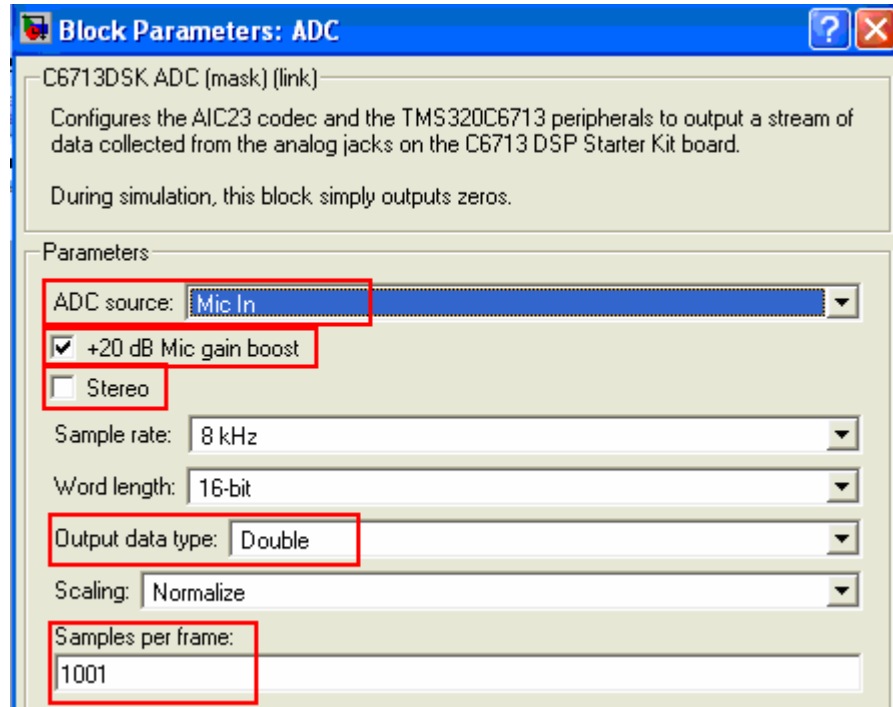
Line in bölümü Mic in olarak değiştirilir.

Stereo butonu seçimi iptal edilir.

+20 dB Mic gain boost seçeneği etkin hale getirilerek ses işaretinin kuvvetlendirilmesi sağlanmış olur.

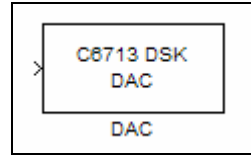
Output data type Double olarak ayarlanır..

Samples per frame 1001 olarak değiştirilir.



Şekil 2.76 ADC Bloğu Özellikleri

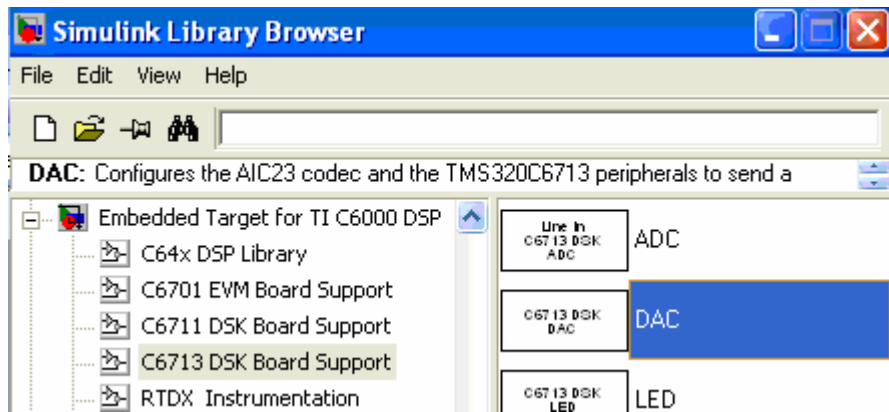
2.9.1.2 Bölüm 2:



Şekil 2.77 Bölüm 2

DAC bloğu aşağıdaki yönerge takip edilerek programa eklenir. Bkz Şekil 2.78

Simulink Library Browser > Embedded Target fot TI C6000 DSP > C6713 DSK Board Support > DAC



Şekil 2.78 DAC Bloğu

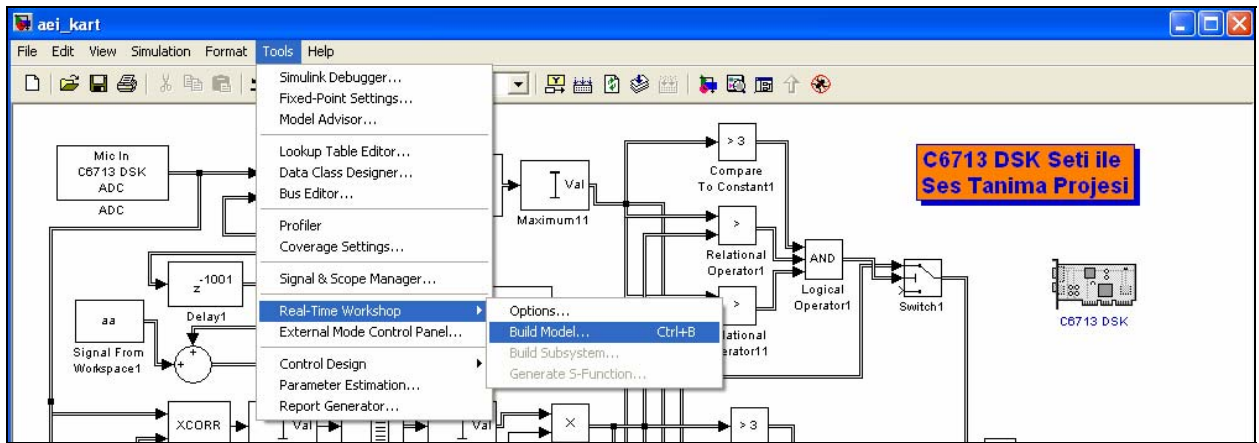
2.9.2 ‘aei_kart.mdl’ Ana Programının CCS Programına Aktarılması :

Ana programın karta yüklenmesi ve simülasyonun gerçekleştirilebilmesi için öncelikle alt programlar çalıştırılarak gerekli veriler elde edilir.

Aşağıdaki yönerge takip edilerek ana program, CCS programında proje dosyası olarak oluşturulur. Bkz. Şekil 2.79, Şekil 2.81

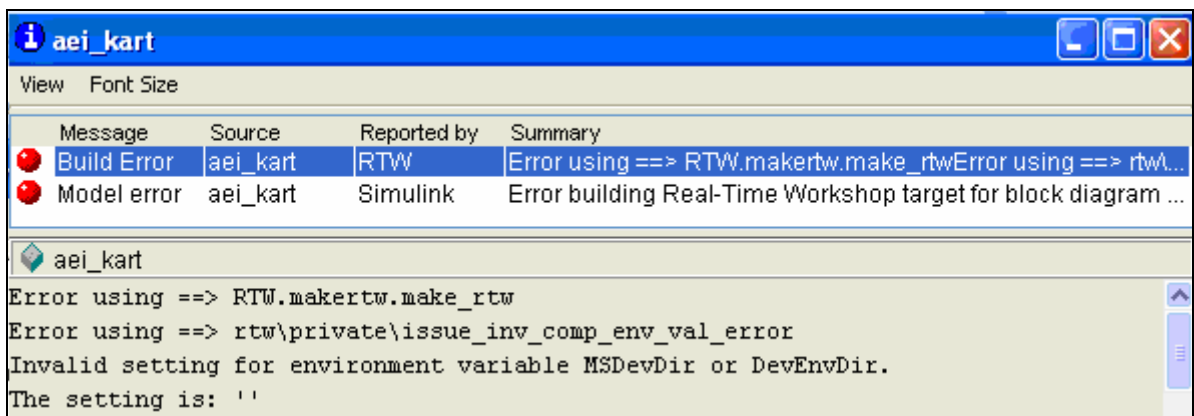
Tools > Real-Time Workshop > Build Model

Ancak bu işlemin gerçekleştirilebilmesi için Bölüm xx ‘de anlatılan 6713 DSK Diagnostics Utility programının önceden çalıştırılarak kartın test edilmiş olması gereklidir.

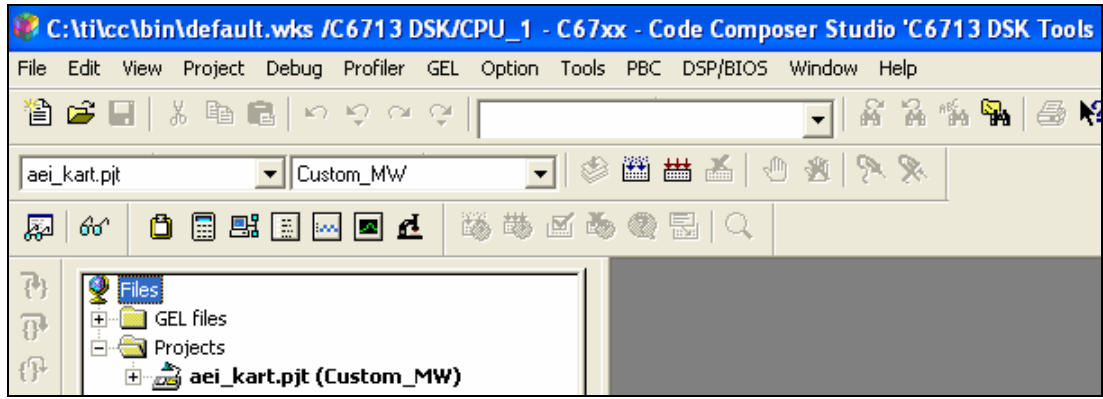


Şekil 2.79 CCS Programına Aktarım

Build Model seçeneği seçildikten sonra Matlab’ta yapılan işlemlerin son aşamasında Şekil 2.80’deki hata ekranı ile karşılaşılır. Ancak bu hata programın karta yüklenip simülasyonun çalıştırılmasında bir engel teşkil etmez.



Şekil 2.80 Hata Ekranı



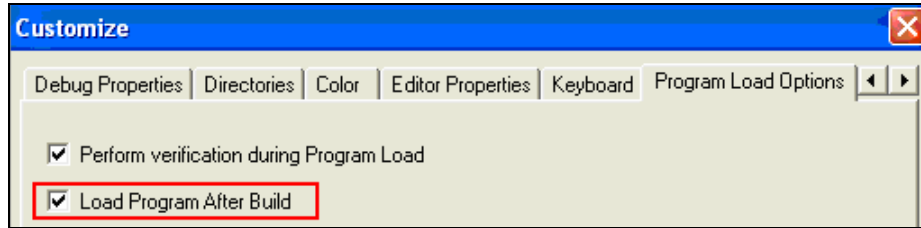
Şekil 2.81 'aei_kart.pjt' Proje Dosyası

2.9.3 'aei_kart.pjt' Proje Dosyasının C6713 DSK Setine Yüklenmesi:

Aşağıdaki yönerge takip edilerek Şekil 2.82'deki pencereye ulaşılır.

Load Program After Build seçeneği işaretlenerek, CCS programında Build işleminden hemen sonra simülasyonun C6713 DSK setine yüklenmesi sağlanmış olur.

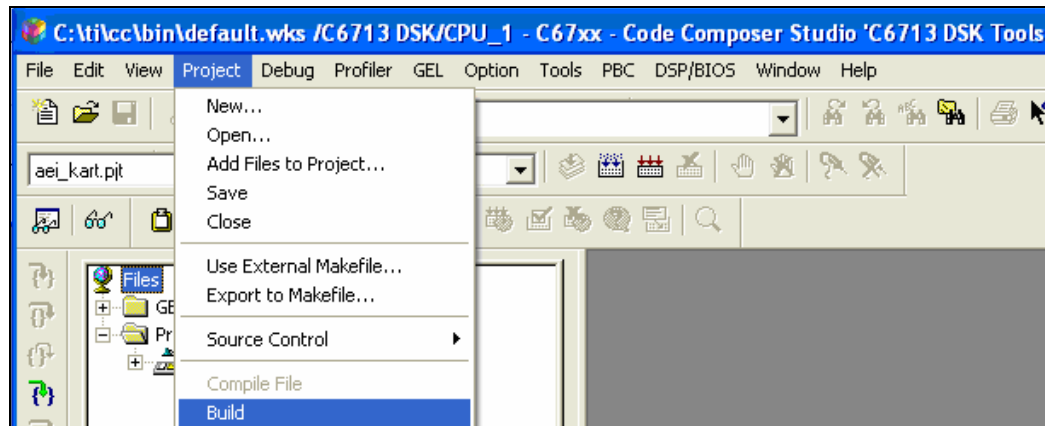
Options > Customize > Program Load Options



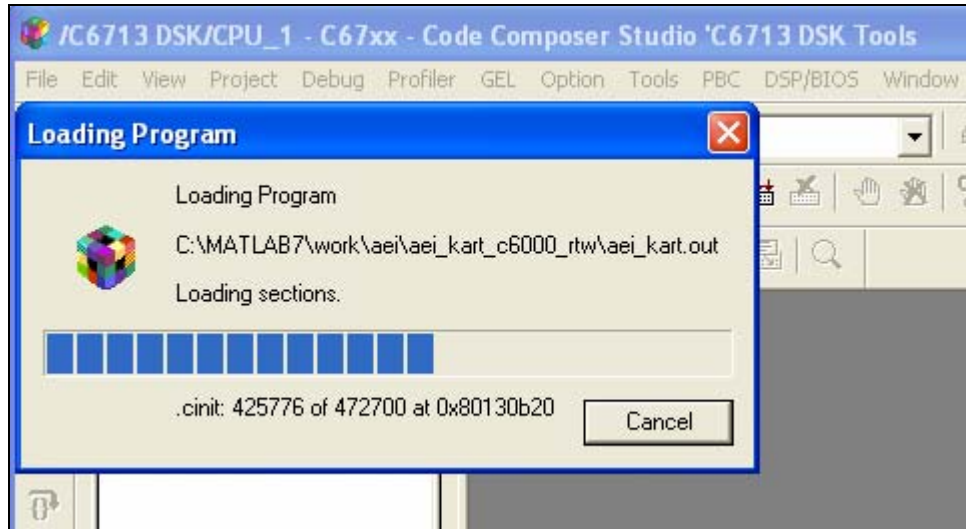
Şekil 2.82 CCS Ayarları

Aşağıdaki yönerge takip edilerek yükleme işlemi gerçekleştirilir. Bkz. Şekil 2.83, Şekil 2.84

Project > Build



Şekil 2.83 Build İşlemi

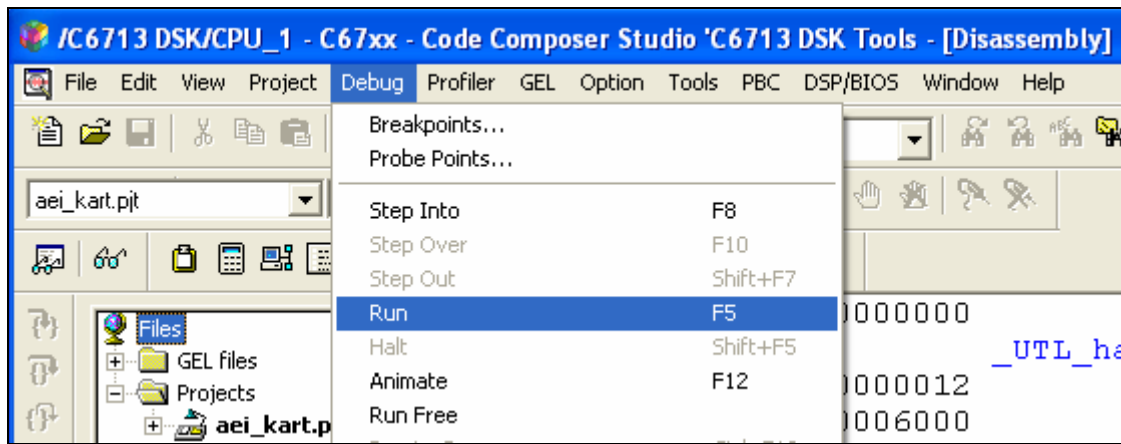


Şekil 2.84 Programın Yüklenmesi

2.9.4 Simülasyonun C6713 DSK Setinde Çalıştırılması:

C6713 DSK setinin Mic in girişine mikrofon, Headphone girişine ise hoparlör bağlanır. Aşağıdaki yönerge takip edilerek C6713 DSK seti üzerinde simülasyon çalıştırılarak mikrofona ‘a’, ‘e’ veya ‘i’ seslerinden herhangi birisi söylenir. Böylece kullanıcı bu girişlere ait çıkış ses dosyalarını dinleyerek ses tanıma işleminin gerçekleştiğini duymaktadır. Bkz. Şekil 2.85

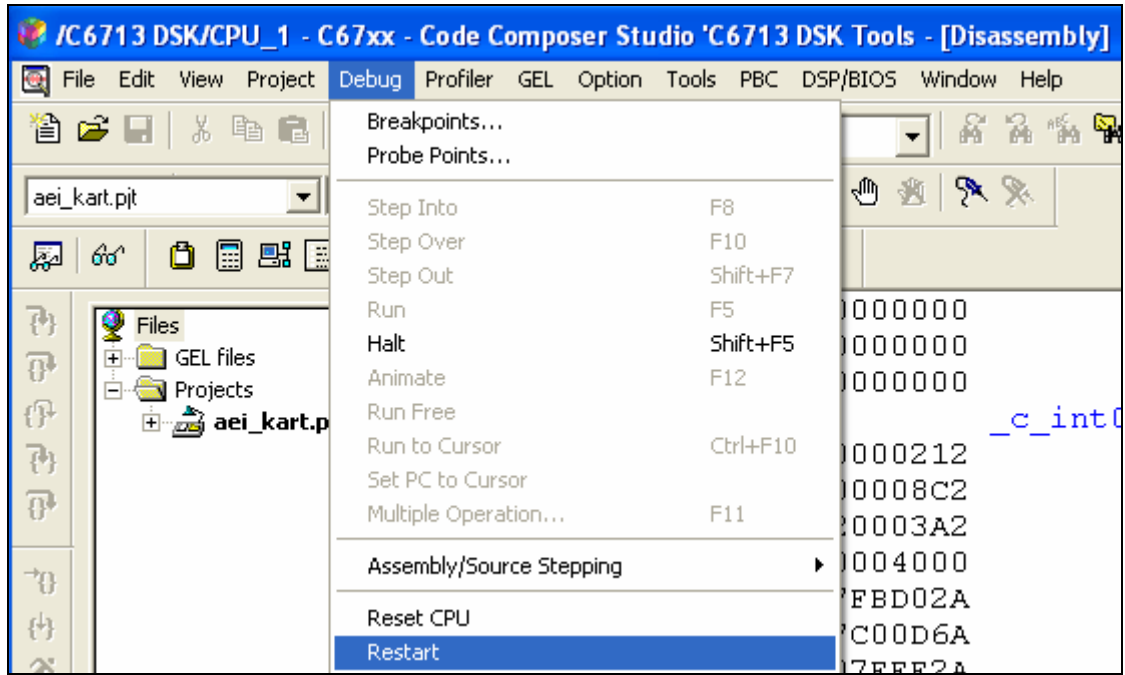
Debug > Run



Şekil 2.85 Run İşlemi

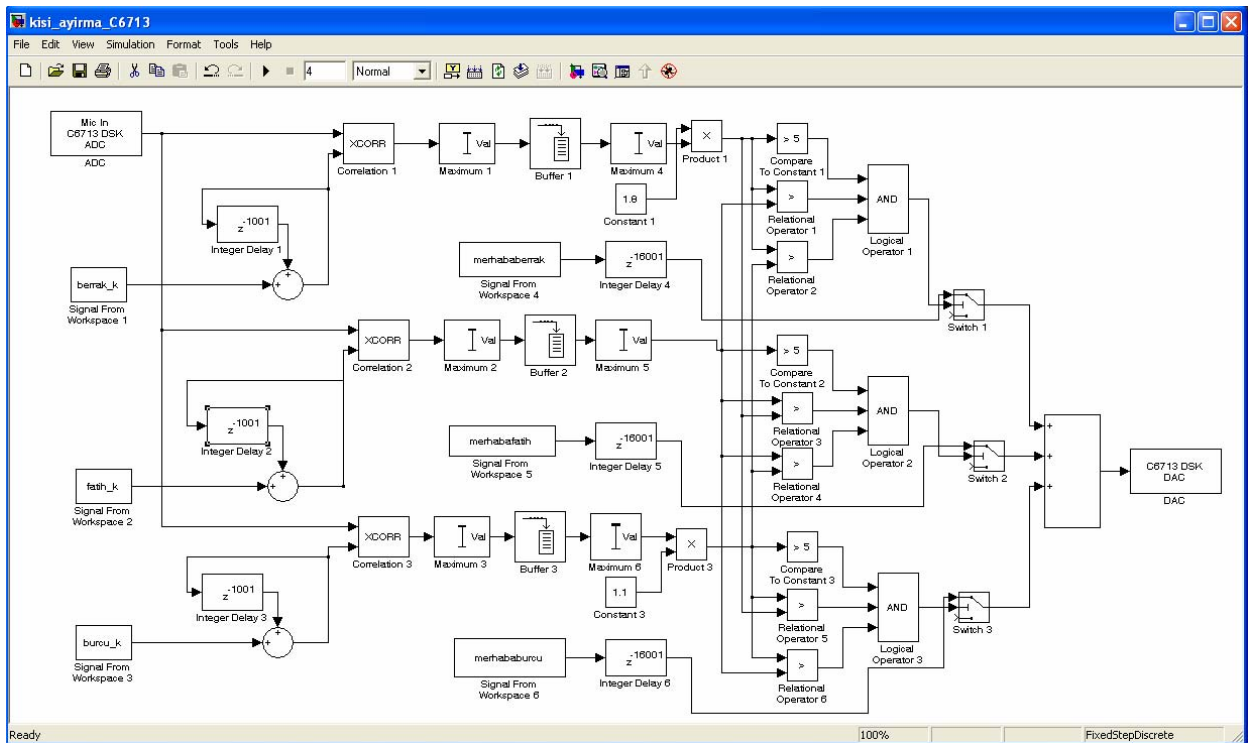
Simülasyonu tekrarlamak için aşağıdaki yönerge izlenir. Bkz. Şekil 2.86

Debug > Restart



Şekil 2.86 Restart İşlemi

2.10 Uygulama Projesi: Gerçek Zamanlı Kişi Tanıma



Şekil 2.87 Gerçek Zamanlı Kişi Tanıma Ana Program Bloğu

Gerçek zamanlı ses tanıma projemizde kullanılan yöntem ile gerçek zamanlı kişi tanıma uygulamasının gerçekleştirildiği blok diyagram Şekil 2.87’deki gibidir.

Bu projede farklı kişilere ait ‘a’ sesleri kayıt edilerek karşılaştırma kütüphanesi oluşturulmuştur. Kullanıcıların mikrofona ‘merhaba’ giriş işaretini göndermeleriyle simülasyon bu kütüphane ile karşılaştırma yaparak, tanıdığı kişiye ait çıkış ses dosyasını hoparlöre gönderir. Burada çıkış ses dosyaları ‘merhaba (kullanıcı ismi)’ şeklinde kayıtlı olduğundan, kullanıcı kendi isminin söylendiğini duyarak, kişi tanıma işleminin gerçekleştiğini anlamış olur.

3. SONUÇ:

Bu projede C6713 DSK kartı ile Gerçek Zamanlı Ses Tanıma İşlemi gerçekleştirilmiştir.

Simülasyonun, Matlab Simulink ve C6713 DSK ortamlarında çalıştırılarak, yapılan denemeleri sonucunda %80'in üzerinde başarı sağlandığı gözlemlenmiştir.

Simülasyon, aynı kişiye ait ses örneklerinin çoğaltılması, örnekleme frekansının artırılması, birden fazla kişi için çalışma, tüm alfabenin, hecelerin ve kelimelerin tanınması gibi proje konularının gerçekleştirilmesine açıktır.

Ses tanıma işlemi iletişim, bilgisayar, elektronik, otomasyon, güvenlik, bankacılık gibi sektörlerde 'sesli komut verme' ve 'akıllı ev' benzeri uygulama projelerinde kullanılabilir.

EK 1 : Kişi Ayırma Projesi için Dosyaların Çalıştırılması

‘Kişi Ayırma’ projesinin Matlab Simülink ortamında düzenlenmiş şeklini çalıştırmak için sırasıyla;

- Anahtar seslerin oluşturulması ve gerekli işlemlerin yapılması için;
 - ✓ ses_kaydi.mdl
 - ✓ ses_dosyasından_matrise.mdl
 - ✓ ses_dosyasından_cikisa.mdl
 - ✓ matris_kesme.m
- Matlab’de simülasyonun gerçekleştirilmesi için;
 - ✓ kisi_ayirma.mdl
- C 6713 setinde hazırlanan algoritmayı gerçekleştirmek için;
 - ✓ Kisi_ayirma_C6713.mdl

dosyaları çalıştırılmalıdır.

EK 2 : Ses Tanıma Projesi için Dosyaların Çalıştırılması

‘Ses Tanıma’ projesinin Matlab Simülink ortamında düzenlenmiş şeklini çalıştırmak için sırasıyla;

- Anahtar seslerin oluşturulması ve gerekli işlemlerin yapılması için (araçlar klasöründen) ;
 - ✓ seskayit.mdl
 - ✓ workspace1.mdl
 - ✓ workspace2.mdl
 - ✓ kesme.m
- Matlab’de simülasyonun gerçekleştirilmesi için;
 - ✓ aei_matlab.mdl
- C 6713 setinde hazırlanan algoritmayı gerçekleştirmek için;
 - ✓ aei_kart.mdl

dosyaları çalıştırılmalıdır.

EK 2 : Ses Tanıma Projesi için Dosyaların Çalıştırılması

‘Ses Tanıma’ projesinin Matlab Simülink ortamında düzenlenmiş şeklini çalıştırmak için sırasıyla;

- Anahtar seslerin oluşturulması ve gerekli işlemlerin yapılması için (araçlar klasöründen) ;
 - ✓ seskayit.mdl
 - ✓ workspace1.mdl
 - ✓ workspace2.mdl
 - ✓ kesme.m
- Matlab’de simülasyonun gerçekleştirilmesi için;
 - ✓ aei_matlab.mdl
- C 6713 setinde hazırlanan algoritmayı gerçekleştirmek için;
 - ✓ aei_kart.mdl

dosyaları çalıştırılmalıdır.

ÖZGEÇMİŞ - Berrak ÖZTÜRK

Berrak ÖZTÜRK 1982 yılında Tokat'ta doğdu. 1993 yılında Ömer Seyfettin İlk Okulu'ndan, 1997 yılında Gümüşpala Ortaokulu'ndan, 2000 yılında Süleyman Nazif Süper Lisesi'nden, 2002 yılında İstanbul Üniversitesi TBMYO Biyomedikal Cihaz Teknolojileri Bölümü'nden mezun oldu. Halen İstanbul Üniversitesi Elektrik – Elektronik Mühendisliği bölümünde öğrenimine devam etmektedir. 2005'te Alternatif Mühendislik'te Yurt Dışı Asistanı olarak 11 ay görev yapmıştır. 2001'de Dopa Biyomedikal Cihazları A.Ş.'de 1 ay, 2002'de Bio-Med Hastane Cihazları Ltd.Şti'de 1 ay, 2006'da İstanbul Üniversitesi Müh. Fak. Elektrik-Elektronik Bölümünde 1 ay olarak stajını tamamlamıştır. İstanbul Üniversitesi Elektrik – Elektronik Klübü'nde etkin üye olarak görev almaktadır. Kasım 2006'dan itibaren İstanbul Üniversitesi Elektrik – Elektronik Mühendisliği Bölümü'nde öğrenci asistanlığı görevini yürütmektedir.

ÖZGEÇMİŞ - Tunay ÇAKAR

Tunay ÇAKAR 1980 yılında İstanbul’da doğdu. 1994 yılında Nilüfer Hatun İlköğretim Okulu’ndan, 1998 yılında Maçka Teknik Lisesi Elektrik Bölümü’nden, 2001 yılında Marmara Üniversitesi TBMYO Elektrik Bölümü’nden mezun oldu. Halen İstanbul Üniversitesi Elektrik – Elektronik Mühendisliği bölümünde öğrenimine devam etmektedir. 1998-2001 yılları arasında Hi-Tech Aydınlatma Mühendislik Ltd. Şti’de, proje bölümünde 3 yıl süre ile görev yapmıştır. 2005’de Vestel Elektornik A.Ş. Ar-Ge Bölümü’nde 1 ay, 2006’da İstanbul Üniversitesi Müh. Fak. Elektrik-Elektronik Bölümü’nde 1 ay olarak stajını tamamlamıştır. Kasım 2006’dan itibaren İstanbul Üniversitesi Elektrik – Elektronik Mühendisliği Bölümü’nde öğrenci asistanlığı görevini yürütmektedir.