

**YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**MİKROİŞLEMCİ TABANLI BLOK DİYAGRAM
İNDİRGEME**

Elektronik Mühendisi Selim ALTINIŞIK

**FBE Elektrik Mühendisliği Anabilim Dalı Kontrol ve Otomasyon Programında
Hazırlanan**

YÜKSEK LİSANS TEZİ

Tez Danışmanı : Yrd. Doç. Dr. Şeref Naci ENGİN

İSTANBUL, 2008

İÇİNDEKİLER

	Sayfa
SİMGE LİSTESİ	iv
KISALTMA LİSTESİ	v
ŞEKİL LİSTESİ	vi
ÇİZELGE LİSTESİ	vii
ÖNSÖZ.....	viii
ÖZET.....	ix
ABSTRACT	x
1. GİRİŞ.....	1
2. ACTIVEX KONTROL.....	2
2.1 Matris ActiveX Kontrolü.....	3
2.1.1 Özellikler	4
2.1.2 Olaylar	8
2.2 Ekran ActiveX Kontrolü.....	9
2.2.1 Özellikler	9
2.2.2 Olaylar	14
2.2.3 Metotlar.....	17
3. EDİTÖR YAPISI.....	21
3.1 Menü Yapısı	21
3.1.1 Dosya (File)	22
3.1.2 Düzenle (Edit).....	25
3.1.3 Ekle (Insert)	26
3.1.4 Proje (Project).....	28
3.1.5 Transfer (Transfer).....	31
3.1.6 Yardım (Help).....	33
3.2 Nesne Araç Çubuğu.....	33
3.3 Durum Çubuğu	34
4. BLOK DİYAGRAM OLUŞTURMA.....	35
4.1 Aşağı-Yukarı Doğru Dikey Bağlantı Elemanı Değişim Çizelgesi	37
4.2 Yatay Bağlantı Elemanı Değişim Çizelgesi	39
4.3 Çizilen Blok Diyagramın Kontrol Edilmesi	40
4.4 Çizilen Blok Diyagramın Derlenmesi	43
4.5 Mason Kuralı	44

4.6	Transfer Fonksiyonu ve Fark Denklemi	45
4.7	HEX Dosyasının Oluşturulması	45
4.8	BAT Dosyası	46
5.	PID DENETLEYİCİ.....	47
5.1	Sayısal PID Denetleyici.....	48
5.2	Deney Düzeneği	50
5.3	PID Denetleyici Tasarımı ve Gerçeklemesi	51
6.	SONUÇLAR.....	55
KAYNAKLAR.....		56
EKLER 57		
Ek 1 Blok Diyagram Doğrulama Ana Programı		58
Ek 2 ‘VerifyPathLoop’ Alt Programı		59
Ek 3 Blok Diyagram Derleme Ana Programı.....		63
Ek 4 ‘PathFind’ Alt Programı.....		64
Ek 5 ‘ForwardPathGainSub’ Alt Programı		68
Ek 6 ‘FeedbackLoopGainSub’ Alt Programı		73

SİMGE LİSTESİ

j	Sistemdeki ileri yolları temsil eden sayı
M	Transfer fonksiyonu
M_j	Belirli bir ileri yolun kazancı
Δ_j	1 – j ileri yolu çıkarılınca elde edilen geribesleme çevrimleri
Δ	1 – geribesleme çevrim kazançları toplamı + birbirine değmeyen iki çevrimin kazançlarının çarpımının toplamı - birbirine değmeyen üç çevrimin kazançlarının çarpımının toplamı +

KISALTMA LİSTESİ

BDD	Blok diyagram derleyici
CMI	'Pointer'ın işaret ettiği matris elemanının indeksidir.
DV	'Pointer'ın işaret ettiği 'Coefficient Unit' elemanının katsayı değeridir.
FLD	Blok diyagramdaki geribesleme çevrimlerinin 'Input Unit' nesnesinin alt tarafında mı yoksa üst tarafında mı bulunduğu bilgisini tutan dizi değişkendir.
FLI	Blok diyagramdaki geribesleme çevrimlerinin başlangıç noktalarının indeks değerlerini tutan dizi değişkenidir.
FLS	Blok diyagramda mevcut geribesleme çevrim sayısını tutan değişkendir.
FPD	Blok diyagramdaki ileri yolların 'Input Unit' nesnesinin alt tarafında mı yoksa üst tarafında mı bulunduğu bilgisini tutan dizi değişkendir.
FPG	Blok diyagramdaki ileri yolların kazanç değerlerini tutan dizi değişkenidir.
FPI	Blok diyagramdaki ileri yolların başlangıç noktalarının indeks değerlerini tutan dizi değişkenidir.
FPS	Blok diyagramda mevcut ileri yol sayısını tutan değişkendir.
PID	Proportional, integral derivative – oran, integral, türev.
SP	Alt program icra edilmeden önce 'pointer' ın en son işaret ettiği matris elemanının indeksidir.

ŞEKİL LİSTESİ

Sayfa

Şekil 2-1 ActiveX kontrolü ekleme penceresi	2
Şekil 2-2 Matris ActiveX kontrolü değer özelliği	6
Şekil 2-3 Matris ActiveX kontrolü seçililik özelliği	6
Şekil 2-4 Matris ActiveX kontrolü katsayıyla çarpma bloğu veri girişi.....	7
Şekil 2-5 Cebirsel toplama bloğu üst işaret özelliği	8
Şekil 2-6 Cebirsel toplama bloğu alt işaret özelliği	8
Şekil 2-7 Ekran ActiveX kontrolü	9
Şekil 2-8 Ekran ActiveX kontrolü değer özelliği	12
Şekil 2-9 Matris ActiveX kontrolü katsayı elemanı veri girişi.....	13
Şekil 3-1 Blok diyagram derleyici editörü	21
Şekil 3-2 Kaydet diyalog penceresi	22
Şekil 3-3 Farklı kaydet diyalog penceresi	23
Şekil 3-4 Blok diyagram dosyası aç diyalog penceresi	24
Şekil 3-5 Editör kapatma teyit diyalog penceresi	25
Şekil 3-6 Katsayıyla çarpma bloğu.....	27
Şekil 3-7 Birim geciktirici bloğu	27
Şekil 3-8 Cebirsel toplama bloğu	27
Şekil 3-9 Giriş elemanı	27
Şekil 3-10 Çıkış elemanı	27
Şekil 3-11 Kontrol sistemi blok diyagramı	28
Şekil 3-12 Denetleyici ayarları diyalog penceresi	29
Şekil 3-13a PID blok diyagramı	30
Şekil 3-13b BDD editöründe oluşturulmuş PID blok diyagramı	30
Şekil 3-14 Grafik diyalog penceresi	31
Şekil 3-15 Hedefi belirle diyalog penceresi	32
Şekil 3-16 IC-Prog programı	32
Şekil 3-17 Haberleşme ayarları diyalog penceresi	33
Şekil 3-18 Nesne araç çubuğu	34
Şekil 3-19 Durum çubuğu	34
Şekil 4-1 Çıkış penceresi	40
Şekil 5-1 Sayısal kontrol sistemi	47
Şekil 5-2 Ampül sıcaklık kontrol devresi	51
Şekil 5-3 Sürecin tepki eğrisi (28 – 40 °C).....	51
Şekil 5-4 Sıcaklık denetleyicisi örnek blok diyagramı	52
Şekil 5-5 Sıcaklık denetim çıkışı (Geçici hal cevabı).....	53
Şekil 5-6 Sıcaklık denetim çıkışı (Kalıcı hal cevabı)	53

ÇİZELGE LİSTESİ

Sayfa

Çizelge 2-1 Silme değişim çizelgesi.....	15
Çizelge 4-1 Aşağı-yukarı doğru dikey bağlantı elemanı değişim çizelgesi	38
Çizelge 4-2 Yatay bağlantı elemanı değişim çizelgesi.....	40
Çizelge 4-3 Yatay bağlantı elemanı doğrulama çizelgesi	41
Çizelge 4-4 Dikey & yatay bağlantı elemanı (5-6-7) doğrulama çizelgesi	42
Çizelge 4-5 Dikey & yatay bağlantı elemanı (10-11) doğrulama çizelgesi	42
Çizelge 5-1 Ziegler – Nichols tablosu	52

ÖNSÖZ

Çalışmalarım sırasında yardımlarını esirgemeyerek bilgi ve tecrübeleri ile bana her zaman destek olan değerli hocam ve danışmanım Sayın Yrd. Doç. Dr. Şeref Naci Engin'e sonsuz teşekkürlerimi sunarım.

Eylül 2008

ÖZET

Bu çalışmada ele alınan örnek sıcaklık denetim uygulamasında olduğu gibi, etkin bir sayısal kontrol sistemi için süreç dinamiklerine uygun denetleyici algoritmalarının geliştirilmesi, yüklenen denetleyici programının katsayılarının kolaylıkla değiştirilmesi ve güncellenmesi gerekir. Önerilen derleyici, tasarlanan her bir PID blok diyagramına karşılık gelen fark denklemini her deneme için tekrar elde etmeli ve işlemciye yüklemelidir. Bu çalışmada çeşitli kontrol deneyleri yapmaya elverişli bir sayısal denetim sistemi elde etmek amacıyla kullanıcıya grafiksel bir arayüz sağlayan, girilen denetleyici algoritmasını fark denklemine dönüştürüp sadeleştirdikten sonra mikroişlemciye indiren ve, giriş ve çıkış değişkenlerini grafik olarak görüntülleyen bir blok diyagram derleyici editörü sunulmuştur.

Anahtar Kelimeler: Derleyici, blok diyagram, transfer fonksiyonu, fark denklemi

ABSTRACT

As in the example of temperature control application carried out in this study, in order to achieve an effective digital control system, it is required to develop control algorithms appropriate for the process dynamics and, to modify and update the controller parameters easily. The proposed block diagram compiler generates a simplified difference equation corresponding to the designed PID block diagram every time it is compiled and then downloads to the microprocessor selected. In this thesis, a block diagram compiler editor is introduced, which presents a graphical user interface to perform control experiments conveniently. The compiler allows the user to draw the block diagram of a digital processor, e.g. a controller or filter, desired to be implemented, then converts it to the corresponding z-domain pulse transfer function and the difference equation. It simplifies the difference equation obtained before it downloads to the microprocessor. The compiler also displays the input and out variables changing over time.

Keywords: Compiler, block diagram, transfer function, difference equation

1. GİRİŞ

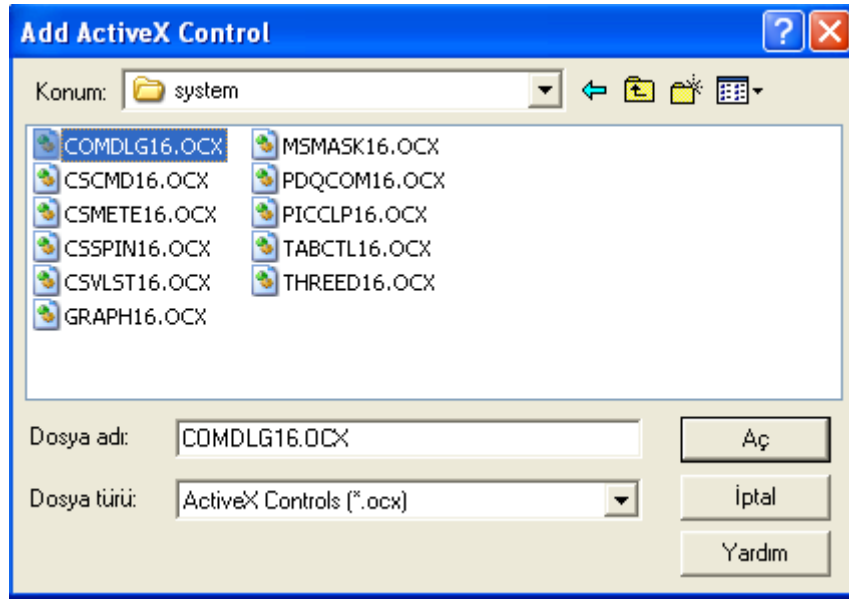
Günümüzde sayısal işlemcilerin hızla yaygınlaşmasıyla işaretler ekseriyetle sayısal olarak işlenmekte, birçok süreç sayısal kontrol sistemleri ile denetlenmektedir. Sayısal işaret işleme veya sayısal denetim sistemlerinin gerçekleştirilmesinde temel adımı arzu edilen çıkışı üretecek algoritmanın tasarlanıp işlemciye yüklenmesi oluşturur. Blok diyagram derleyici (BDD) editörü, denetleme veya süzme gibi bir sayısal işlemi gerçekleştirmek üzere z tanım bölgesinde geliştirilmiş ve blok diyagram olarak çizilmiş bir algoritmayı bir mikroişlemcide koşturmak üzere kaynak koda dönüştürür.

Önerilen BDD editörü, kullanıcının tasarlayacağı algoritmaları kolaylıkla denemesini ve yüklediği sayısal işlem değişkenlerini en uygun biçimde ayarlamasını sağlayacak, giriş ve çıkış değişkenlerini izleyecek uygun bir araç olarak sunulmuştur. Ele alınan örnek sıcaklık denetim uygulamasında sistem, “arzu edilen çıkış” olarak girilen “set değeri”, $r(t)$, ile süreçten okunan “gerçek çıkış” verisinin, $c(t)$, farkını, diğer bir deyişle hata işaretini, $e(t) = r(t) - c(t)$, denetleyicinin giriş değişkeni olarak alır. Örneklenmiş hata işareti, $e(kT)$, tasarlanan denetim algoritması içerisinde işlendikten sonra üretilen ayrık çıkış işareti, $u(kT)$, sürekli hale dönüştürülür, $u(t)$, ve tekrar sürece uygulanır. BDD editöründe oluşturulan blok diyagramı, program tarafından Mason kuralı kullanılarak önce denetleyiciyi temsil eden darbe transfer fonksiyonuna, $G_D(z) = U(z)/E(z)$, dönüştürülür. Program daha sonra bu darbe transfer fonksiyonuna matematiksel sadeleştirme işlemleri uygular. Transfer fonksiyonunun en sade hali elde edildikten sonra bu fonksiyonu fark denklemine dönüştürür. Üretilen fark denklemini daha önceden oluşturulmuş bir taslak C programına yerleştirir. Bu C dosyası ilgili derleyici vasıtasıyla otomatik olarak derlenir. İşlem sonucunda elde edilen HEX dosyası editör içinde gömülü olan IC-Prog programı ile PIC mikroişlemcisine yüklenir. Editör Visual Basic programlama dilinde oluşturulmuştur.

Bu tez, Giriş bölümünden sonra geliştirilen derleyici editör yazılımının çalışma alanının nasıl tasarlandığının anlatıldığı ActiveX Kontrol, yazılımın nasıl kullanıldığının açıklandığı Editör Yapısı, derleme işleminin nasıl gerçekleştirildiğinin açıklandığı Blok Diyagramı Oluşturma, kontrol deneylerinin anlatıldığı PID Denetleyicisi ve elde edilen sonuçların değerlendirildiği Sonuç bölümlerini içermektedir.

2. ACTIVEX KONTROL

Visual Basic programlama dilinde tasarım esnasında kullanılabilen programla birlikte gelen çok sayıda yazılım aleti, kontrol, bulunmaktadır. Bu kontroller çalışma alanına eklenerek yazılan kodlarla beraber işlevlik kazanır. Bu kontrollere örnek vermek gerekirse metin kutusu (textbox), seçim kutusu (combobox) ve liste kutusu (listbox) hemen hemen her projede kullanılan kontrollerdir. Bu kontroller sabit diskte OCX uzantısı ile saklanmaktadır. Visual Basic ile birlikte verilen OCX dosyalarının çoğunluğu işletim sisteminin kurulu olduğu klasörün altında bulunan SYSTEM klasöründe saklanmaktadır. Şekil 2.1’de belirli bir ActiveX kontrolünün projeye eklenmesi için çıkan diyalog penceresi gösterilmiştir.



Şekil 2-1 ActiveX kontrolü ekleme penceresi

Kullanıcı, Visual Basic’in standart kontrollerini kullanmak yerine kendi isteklerine cevap veren bir ActiveX kontrolü de oluşturabilir. Hazırladığı bu ActiveX kontrolünü projesinde tıpkı Visual Basic’in standart kontrolleri gibi kullanabilir. Bir projede ActiveX kontrolleri oluşturmak yerine normal kodlamayla da aynı işlevi yerine getirmek mümkündür. Tahmin edileceği gibi, ActiveX kullanmanın göz ardı edilemeyecek nisbette üstünlükleri vardır. Bu avantajlar aşağıda listelenmiştir (Francesco Balena, 1999).

- Aynı projede yazılan kod miktarını dolayısıyla projenin tamamlanma süresini kısaltır.
- Projeye modülerlik kazandırır, hata bulmayı kolaylaştırır.
- Bir projede birden fazla kişinin çalışmasına olanak sağlar.

- Oluşturulan ActiveX kontrolleri standart yapısı sayesinde ihtiyaç duyulan her projede kullanılabilir.

Her ActiveX kontrolünün kendisine ait özellik (property), olay (event) ve metotları (method) vardır. Bir ActiveX'i kullanabilmek için bunların hepsinin bilinmesi gerekmektedir. Bu terimler aşağıda açıklanmıştır.

- Özellik: Bir ActiveX'in o anki durumu hakkında bilgi vermek veya mevcut durumunu değiştirmek amacı ile kullanılırlar. Örnek: Bir metin kutusuna yazılmış olan yazının program tarafından okunması ya da program tarafından metin kutusuna belirli bir yazının yazılması
- Olay: Bir ActiveX kontrolünde meydana gelen belirli bir olayın kesme (interrupt) ile programa bildirilmesi amacı ile kullanılırlar. Örnek: Bir butona tıklanması.
- Metot: Bir ActiveX içerisine yazılmış alt programların program tarafından icra edilmesi için kullanılır. Örnek: Bir liste kutusunun tüm içeriğinin silinmesi

BDD editöründe de kullanıcının blok diyagramı oluşturduğu çalışma alanı bir ActiveX kontrolü olarak tasarlanmıştır. Bu ActiveX kontrolünün kullanılabilmesi için gerekli olan tüm özellik, olay ve metotlar ayrıntılı şekilde açıklanmıştır.

2.1 Matris ActiveX Kontrolü

BDD editörünün blok diyagram derleme işlemini gerçekleştirebilmesi için öncelikle kullanıcıya blok diyagram çizmeye olanak sağlayacak bir arayüz sunması gerekmektedir. Kullanıcının blok diyagramı oluşturduğu çalışma alanı bir matris olarak tasarlanmıştır. Kullanıcı matrisin herhangi bir elemanına istediği blok diyagram elemanını yerleştirebilmelidir. Bu da bize tüm matris elemanlarının birbirleri ile aynı özellikte olması gerektiğini gösterir. Bu amaç doğrultusunda tüm matris elemanlarını teker teker tasarlamak yerine tüm isteklere cevap veren tek bir matris elemanı tasarlayıp bu elemanın kopyaları ile matrisi oluşturmak çok daha mantıklı olacaktır. Böylece yazılan kod miktarının önemli ölçüde azalması ve programın daha verimli çalışması sağlanmış olur. Programlanan bu matris elemanı matris ActiveX kontrolü olarak adlandırılmıştır. Matris ActiveX kontrolü sayesinde kullanıcı, editörde istediği blok diyagramı kolayca oluşturabilir. Matris ActiveX kontrolüne ait özellik ve olaylar aşağıda listelenmiştir.

2.1.1 Özellikler

- Değer (Value): Bu özellik sayesinde matris ActiveX'in mevcut görüntüsü değiştirilebildiği gibi ayrıca çalışma esnasında mevcut görüntü bilgisi de elde edilebilir. Mevcut görüntünün değiştirilmesi için değer özelliğine 0 ile 14 arasında bir sayı atanır ve bu atanan sayıya göre görüntü değişir. Mevcut görüntü ise değer özelliğinin okunması ile 0 ile 14 arasında bir sayısal değer olarak elde edilir. Şekil 2.2'de sayısal değerlere karşılık gelen görüntüler gösterilmiştir.

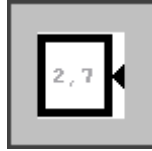
- Değer = 0



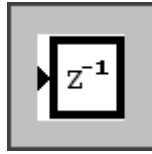
- Değer = 1 (Katsayıyla çarpma bloğu)



- Değer = 2 (Sağdan girişli katsayıyla çarpma bloğu)



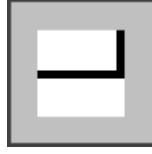
- Değer = 3 (Birim geciktirici bloğu)



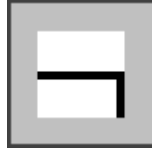
- Değer = 4 (Sağdan girişli birim geciktirici bloğu)



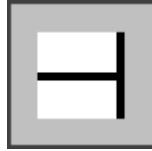
- Değer = 5 (Yatay ve sağ-üst dikey bağlantı)



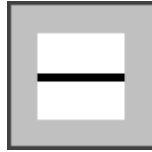
- Değer = 6 (Yatay ve sağ-alt dikey bağlantı)



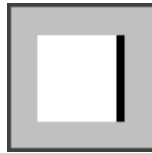
- Değer = 7 (Yatay ve dikey bağlantı)



- Değer = 8 (Yatay bağlantı)



- Değer = 9 (Dikey bağlantı)



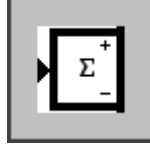
- Değer = 10 (Sağ-üst dikey bağlantı)



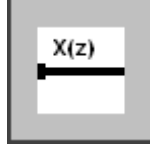
- Değer = 11 (Sağ-alt dikey bağlantı)



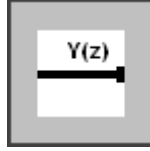
- Değer = 12 (Cebirsel toplama bloğu)



- Değer = 13 (Giriş elemanı)

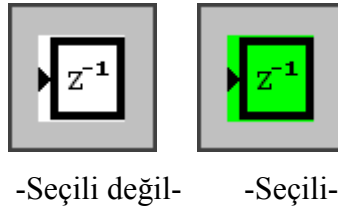


- Değer = 14 (Çıkış elemanı)



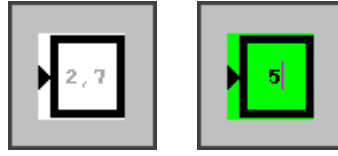
Şekil 2-2 Matris ActiveX kontrolü değer özelliği

- Seçililik (Selected): Bu özellik sayesinde matris ActiveX'in mevcut görüntüsü seçili hale dönüştürülebildiği gibi ayrıca çalışma esnasında mevcut görüntünün seçili olup olmadığı bilgisi de elde edilebilir. Mevcut görüntünün seçili hale getirilmesi için seçililik özelliğine True Boolean değeri aktarılmalıdır. Benzer şekilde seçili halin iptal edilmesi için ise seçililik özelliğine False Boolean değeri aktarılmalıdır. Seçililik özelliğinin okunması ile mevcut görüntünün seçili olup olmadığı anlaşılır. Okunan değer True Boolean değeri ise mevcut görüntü seçili durumdadır, okunan değer False Boolean değeri ise bu durumda mevcut görüntü seçili durumda değildir. Seçili durumda olan matris elemanı yeşil renk alır. Şekil 2.3'te birim geciktirici bloğunun seçili ve seçili olmayan durumları gösterilmiştir.



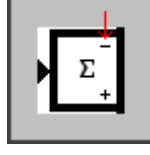
Şekil 2-3 Matris ActiveX kontrolü seçililik özelliği

- Veri (Data): Bu özellik sayesinde matris ActiveX'in değer özelliğine 1 veya 2 sayısının atanması ile görünür hale gelen katsayıyla çarpma bloğunun normalde 1 olan katsayı değerinin değiştirilmesi ve mevcut katsayı değerinin elde edilmesi sağlanır. Çalışma esnasında katsayıyla çarpma bloğu üzerine fare ile çift tıklandığında katsayı değeri seçili hale gelerek veri girişine izin verilir. Girilen verinin ActiveX tarafından kabul edilmesi için klavyenin 'enter' tuşuna basılması gerekmektedir. Fakat aynı zamanda girilen verinin geçerliliği ActiveX tarafından kontrol edilir. Yapılan bu kontroller aşağıda listelenmiştir.
- Değer girişi esnasında klavyenin sadece rakam, virgöl, 'delete', 'backspace', 'escape' ve 'enter' tuşları işlevlidir.
- Girilen veri en fazla bir adet virgöl içerebilir.
- Virgöl en başta ya da en sonda olamaz.
- 'Escape' tuşuna basılması durumunda veri, son giriş yapılan değere geri döner ve veri girişi sonlandırılır. Tekrar veri girişi yapmak için yine katsayıyla çarpma bloğunun üzerine fare ile çift tıklanması gerekmektedir (Şekil 2.4).



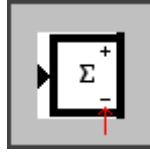
Şekil 2-4 Matris ActiveX kontrolü katsayıyla çarpma bloğu veri girişi

- (Üst İşaret) InUpSign: Bu özellik sayesinde matris ActiveX'in değer özelliğine 12 sayısının atanması ile görünür hale gelen cebirsel toplama bloğunun üst kısmında (Şekil 2.5) bulunan artı-eksi işaret verisinin değiştirilmesi ve mevcut işaret verisinin elde edilmesi sağlanır. Çalışma esnasında artı işaret üzerine fare ile çift tıklandığında işaret eksiye dönüşür. Benzer şekilde eksi işaret üzerine fare ile çift tıklandığında işaret artıya dönüşür. İşaretin artıya dönüştürülmesi için üst işaret özelliğine 0 değeri atanmalıdır. Benzer şekilde üst işaret özelliğinin okunması ile elde edilen değer 0 ise bu bize işaretin artı olduğunu gösterir. İşaretin eksiye dönüştürülmesi için üst işaret özelliğine 1 değeri atanmalıdır. Benzer şekilde üst işaret özelliğinin okunması ile elde edilen değer 1 ise bu bize işaretin eksi olduğunu gösterir.



Şekil 2-5 Cebirsel toplama bloğu üst işaret özelliği

- Alt İşaret (InDownSign): Bu özellik sayesinde matris ActiveX' in değer özelliğine 12 sayısının atanması ile görünür hale gelen cebirsel toplama bloğunun alt kısmında (Şekil 2.6) bulunan artı-eksi işaret verisinin değiştirilmesi ve mevcut işaret verisinin elde edilmesi sağlanır. Çalışma esnasında artı işaret üzerine fare ile çift tıklandığında işaret eksiye dönüşür. Benzer şekilde eksi işaret üzerine fare ile çift tıklandığında işaret artıya dönüşür. İşaretin artıya dönüştürülmesi için alt işaret özelliğine 0 değeri atanmalıdır. Benzer şekilde alt işaret özelliğinin okunması ile elde edilen değer 0 ise bu bize işaretin artı olduğunu gösterir. İşaretin eksiye dönüştürülmesi için alt işaret özelliğine 1 değeri atanmalıdır. Benzer şekilde alt işaret özelliğinin okunması ile elde edilen değer 1 ise bu bize işaretin eksi olduğunu gösterir.



Şekil 2-6 Cebirsel toplama bloğu alt işaret özelliği

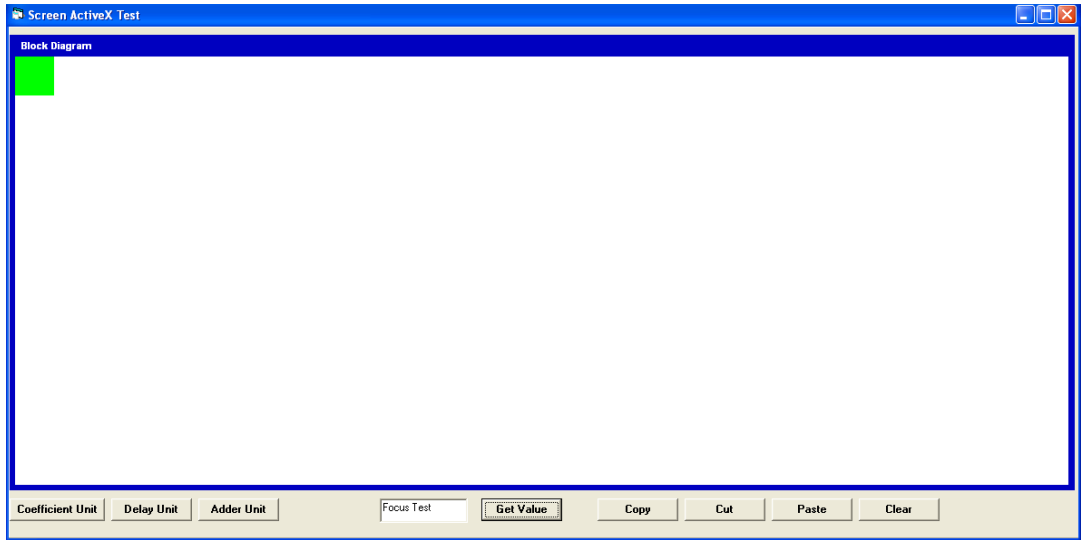
2.1.2 Olaylar

- Hücre Tıklama (CellClick): Matris ActiveX kontrolü üzerine fare ile tıklanıldığı zaman hücre tıklama olayı meydana gelir. Böylelikle ActiveX üzerine fare ile tıklandığı bilgisini elde etmiş oluruz.
- Hücre Çift Tıklama (CellDoubleClick): Matris ActiveX kontrolü üzerine fare ile çift tıklanıldığı zaman hücre çift tıklama olayı meydana gelir. Böylelikle ActiveX üzerine fare ile çift tıklandığı bilgisini elde etmiş oluruz.
- Hücre Fare Tıklama (CellMouseDown): Matris ActiveX kontrolü üzerine fare ile tıklanıldığı zaman hücre fare tıklama olayı meydana gelir. Bu olayın hücre tıklama olayından farkı tıklama olayının farenin hangi tuşuna basılarak yapılmış olduğu, tıklama esnasında klavyenin 'shift' tuşunun basılı olup olmadığı ve tıklama koordinatlarını döndürmesidir.

2.2 Ekran ActiveX Kontrolü

BDD editörünün kullanıcıya blok diyagram çizmeye olanak sağladığı çalışma alanı matris ActiveX örneklerinden oluşmaktadır. Bu matris ActiveX kontrolleri ekran ActiveX kontrolünü oluşturur. Böylelikle çok sayıda ActiveX ile uğraşmak yerine tek bir ActiveX kullanarak kullanıcının istediği blok diyagramı çizmesi sağlanmış olur. Şekil 2.7’de ekran ActiveX kontrolü gösterilmiştir.

Ekran ActiveX kontrolü 11x27 boyutunda matris ActiveX örneklerinden meydana gelmiş bir matris şeklindedir. Program ilk açıldığında sol üst köşede bulunan, matrisin ilk elemanı seçili duruma gelir. Klavyenin yön tuşları kullanılarak seçili matris elemanı değiştirilebilir. Ayrıca istenilen matris elemanına fare ile tıklandığı zaman bu matris elemanının seçili duruma gelmesi sağlanır. Ekranda oluşturulmuş olan nesneler klavyenin ‘delete’ tuşuna basılarak silinebilir.



Şekil 2-7 Ekran ActiveX kontrolü

Ekran ActiveX kontrol’e ait özellik, olaylar ve mototlar aşağıda listelenmiştir.

2.2.1 Özellikler

- **Seçili İndeks (SelectedIndex):** Bu özellik sayesinde ekran ActiveX’in belirli bir matris elemanı seçili hale getirilebildiği gibi ayrıca çalışma esnasında seçili olan matris elemanı da öğrenilebilir. Belirli bir matris elemanının seçili hale getirilmesi için seçili indeks özelliğine seçili hale getirilmek istenen matris elemanının indeks numarası atanır. Bu durumda mevcut seçili olan matris elemanı seçili olmayan duruma, indeks numarası

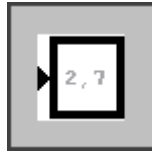
belirtilen matris elemanı ise seçili duruma gelir. Seçili indeks özelliğinin okunması ile mevcut seçili olan matris elemanının indeks numarası elde edilir.

- **İşaretçi (Pointer):** Bu özellik sayesinde ekran ActiveX'in belirli bir matris elemanı bu matris elemanına ait bazı değerlerin elde edilebilmesi için programsal olarak seçili hale getirilir. Ayrıca çalışma esnasında programsal olarak seçili olan matris elemanı da öğrenilebilir. İşaretçi özelliğine programsal olarak seçili hale getirilmek istenen matris elemanının indeks numarası atanır. İşaretçi özelliğinin okunması ile mevcut, programsal olarak seçilen matris elemanının indeks numarası elde edilir.
- **Değer (Value):** Bu özellik sayesinde işaretçi özelliği kullanılarak programsal olarak seçili hale getirilmiş olan matris elemanının mevcut görüntüsü değiştirilebildiği gibi ayrıca çalışma esnasında mevcut görüntü bilgisi de elde edilebilir. Mevcut görüntünün değiştirilmesi için değer özelliğine 0 ile 14 arasında bir sayı atanır ve bu atanan sayıya göre görüntü değişir. Mevcut görüntü ise değer özelliğinin okunması ile 0 ile 14 arasında bir sayısal değer olarak elde edilir. Şekil 2.8'de sayısal değerlere karşılık gelen görüntüler gösterilmiştir.

- Değer = 0



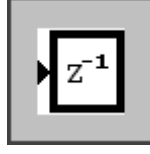
- Değer = 1 (Katsayıyla çarpma bloğu)



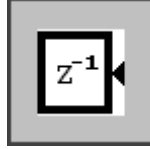
- Değer = 2 (Sağdan girişli katsayıyla çarpma bloğu)



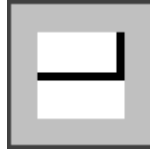
- Değer = 3 (Birim geciktirici bloğu)



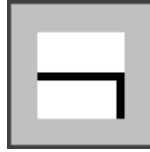
- Değer = 4 (Sağdan girişli birim geciktirici bloğu)



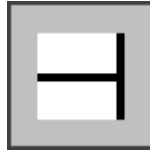
- Değer = 5 (Yatay ve sağ-üst dikey bağlantı)



- Değer = 6 (Yatay ve sağ-alt dikey bağlantı)



- Değer = 7 (Yatay ve dikey bağlantı)



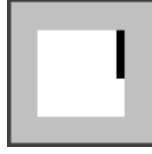
- Değer = 8 (Yatay bağlantı)



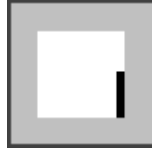
- Değer = 9 (Dikey bağlantı)



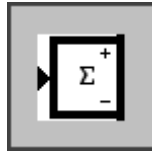
- Değer = 10 (Sağ-üst dikey bağlantı)



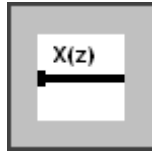
- Değer = 11 (Sağ-alt dikey bağlantı)



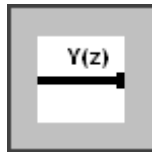
- Değer = 12 (Cebirsel toplama bloğu)



- Değer = 13 (Giriş elemanı)



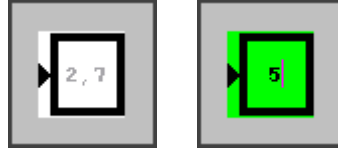
- Değer = 14 (Çıkış elemanı)



Şekil 2-8 Ekran ActiveX kontrolü değer özelliği

- Veri (Data): Bu özellik sayesinde işaretçi özelliği kullanılarak programsal olarak seçili hale getirilmiş olan matris elemanının değer özelliğine 1 veya 2 sayısının atanması ile görünür hale gelen katsayıyla çarpma bloğunun normalde 1 olan katsayı değerinin değiştirilmesi ve mevcut değer elde edilmesi sağlanır. Çalışma esnasında katsayıyla çarpma bloğunun üzerine fare ile çift tıklandığında katsayı değeri seçili hale gelerek veri girişine izin verilir. Girilen verinin ActiveX tarafından kabul edilmesi için klavyenin ‘enter’ tuşuna basılması gerekmektedir. Fakat aynı zamanda girilen verinin geçerliliği ActiveX tarafından kontrol edilir. Yapılan bu kontroller aşağıda listelenmiştir.

- Veri giriři esnasında klavyenin sadece rakam, virgöl, ‘delete’, ‘backspace’, ‘escape’ ve ‘enter’ tuřları işlevlidir.
- Girilen veri en fazla bir adet virgöl içerebilir.
- Virgöl en başta ya da en sonda olamaz.
- ‘Escape’ tuřuna basılması durumunda veri, son giriř yapılan değeri geri döner ve veri giriři sonlandırılır. Tekrar veri giriři yapmak için yine katsayıyla çarpma bloğunun üzerine fare ile çift tıklanması gerekmektedir (Şekil 2.9).



Şekil 2-9 Matris ActiveX kontrolü katsayı elemanı veri giriři

- Üst İşaret (UpSign): Bu özellik sayesinde işaretçi özelliđi kullanılarak programsal olarak seçili hale getirilmiş olan matris elemanının değeri özelliđine 12 sayısının atanması ile görünür hale gelen cebirsel toplama bloğunun üst kısmında bulunan artı-eksi işaret verisinin değıştirilmesi ve mevcut işaret verisinin elde edilmesi sağlanır. Çalışma esnasında artı işaret üzerine fare ile çift tımlandığında işaret eksiye dönüşür. Benzer şekilde eksi işaret üzerine fare ile çift tıklandığında işaret artıya dönüşür. İşaretin artıya dönüřtürülmesi için üst işaret özelliđine 0 değeri atanmalıdır. Benzer şekilde üst işaret özelliđinin okunması ile elde edilen değeri 0 ise bu bize işaretin artı olduğunu gösterir. İşaretin eksiye dönüřtürülmesi için üst işaret özelliđine 1 değeri atanmalıdır. Benzer şekilde üst işaret özelliđinin okunması ile elde edilen değeri 1 ise bu bize işaretin eksi olduğunu gösterir
- Alt İşaret (DownSign): Bu özellik sayesinde işaretçi özelliđi kullanılarak programsal olarak seçili hale getirilmiş olan matris elemanının değeri özelliđine 12 sayısının atanması ile görünür hale gelen cebirsel toplama bloğunun alt kısmında bulunan artı-eksi işaret verisinin değıştirilmesi ve mevcut işaret verisinin elde edilmesi sağlanır. Çalışma esnasında artı işaret üzerine fare ile çift tıklandığında işaret eksiye dönüşür. Benzer şekilde eksi işaret üzerine fare ile çift tıklandığında işaret artıya dönüşür. İşaretin artıya dönüřtürülmesi için alt işaret özelliđine 0 değeri atanmalıdır. Benzer şekilde alt işaret özelliđinin okunması ile elde edilen değeri 0 ise bu bize işaretin artı olduğunu gösterir. İşaretin eksiye dönüřtürülmesi için alt işaret özelliđine 1 değeri atanmalıdır. Benzer

şekilde alt işaret özelliğinin okunması ile elde edilen değer 1 ise bu bize işaretin eksi olduğunu gösterir.

- **Yapıştırma Durumu (PasteStatus):** Ekran ActiveX kontrolü üzerinde çalıştırılan en son yapıştır (paste) metodunun başarılı bir şekilde sonuçlanıp sonuçlanmadığı bilgisini verir. Eğer kopyala (copy) veya kes (cut) metodu ile kopyalanan matris elemanı içeriği istenilen başka bir matris elemanına yapıştırıldıysa yapıştırma durumu özelliği okunduğunda True Boolean değeri elde edilir. Yapıştırma işlemi gerçekleşmediyse False Boolean değeri elde edilir. Eğer işlem başarısız ise yapıştırma durumu özelliğine False Boolean değeri atanır. Yapıştırma durumu özelliğine False Boolean değeri atandığında yapıştırma hatası (PasteError) olayı meydana gelir.

2.2.2 Olaylar

- **Ekran Hücre Çift Tıklama (ScreenCellDoubleClick):** Ekran ActiveX kontrolü üzerine fare ile çift tıklanıldığı zaman ekran hücre çift tıklama olayı meydana gelir. Böylelikle ActiveX üzerine fare ile çift tıklanıldığı bilgisini elde etmiş oluruz. Ayrıca tıklanan matris elemanının indeks bilgisi de döndürülür.
- **Ekran Fare Tıklama (ScreenMouseDown):** Ekran ActiveX kontrolü üzerine fare ile tıklanıldığı zaman ekran fare tıklama olayı meydana gelir. Böylece tıklanan matris elemanı seçili hale geldiği gibi tıklanan matris elemanının indeks değeri, tıklama işleminin farenin hangi tuşuna basılarak yapılmış olduğu, tıklama esnasında klavyenin 'shift' tuşunun basılı olup olmadığı ve tıklama koordinatları bilgileri de döndürülür.
- **Ekran Nesne Silme (ScreenDeleteObject):** Ekran ActiveX kontrolü üzerinde bulunan katsayıyla çarpma bloğu, birim geciktirici bloğu, cebirsel toplama bloğu, giriş elemanı veya çıkış elemanı nesnelerinden birinin silinmesi durumunda ekran nesne silme olayı meydana gelir. Nesneleri birbirine bağlayan bağlantı elemanları silinirken belirli bir düzene göre silinirler. Bu silme işlemi oluşturma sırasına dayanır. Örneğin bir kaç bağlantı elemanının birleşmesi ile oluşan yatay tam yada yarım 'T' şeklini alan matris elemanı görüntüsü klavyenin 'delete' tuşuna basıldığında hepsi birden değil oluşturulma şekline göre adım adım silinir. Bu amaçla Çizelge 2.1'de gösterilen silme çizelgesi oluşturulmuştur. 'X', matris elemanı değerinin önemsiz olduğunu belirtmektedir.

Çizelge 2-1 Silme değişim çizelgesi

Önceki Durum			Sonraki Durum		
Alt Hücre	Aktif Hücre	Üst Hücre	Alt Hücre	Aktif Hücre	Üst Hücre
X	0	X	X	0	X
X	1	X	X	0	X
X	2	X	X	0	X
X	3	X	X	0	X
X	4	X	X	0	X
X	5	X	X	10	X
X	6	X	X	11	X
X	7	X	X	9	X
X	8	X	X	0	X
5	9	X	8	0	X
7	9	X	11	0	X
9	9	X	0	0	X
10	9	X	0	0	X
X	10	6	X	0	8
X	10	7	X	0	5
X	10	9	X	0	10
X	10	11	X	0	0
5	11	X	8	0	X
7	11	X	6	0	X
9	11	X	11	0	X
10	11	X	0	0	X
X	12	X	X	0	X

- **Yapıştırma Hatası (PasteError):** Yapıştırma durumu özelliğine False Boolean değeri aktarıldığında yapıştırma hatası olayı meydana gelir. Bu olay bir hata kodu döndürür. Bu hata kodu program tarafından okunarak hatanın nedeni kullanıcıya yansıtılır.
- **Ekran Tuş Basılı (ScreenKeyPressed):** Ekran ActiveX kontrolü aktif durumda iken klavyenin herhangi bir tuşuna basıldığında ekran tuş basılı olayı meydana gelir ve basılan tuşun ASCII kodu döndürülür.

Döndürülen ASCII koduna göre bir takım işlemler gerçekleştirilir. Aşağıda bu işlemler listelenmiştir.

- Sağ ok tuşu: Mevcut seçili matris elemanı seçili olmayan duruma, bir sağında bulunan matris elemanı seçili duruma getirilir.
- Sol ok tuşu: Mevcut seçili matris elemanı seçili olmayan duruma, bir solunda bulunan matris elemanı seçili duruma getirilir.
- Yukarı ok tuşu: Mevcut seçili matris elemanı seçili olmayan duruma, bir üstünde bulunan matris elemanı seçili duruma getirilir.
- Aşağı ok tuşu: Mevcut seçili matris elemanı seçili olmayan duruma, bir altında bulunan matris elemanı seçili duruma getirilir.
- ‘Delete’ tuşu: Mevcut seçili matris elemanı katsayıyla çarpma bloğu, birim geciktirici bloğu, cebirsel toplama bloğu, giriş elemanı veya çıkış elemanı ise matris elemanı bilgisinin silinmesi için onay istenir. Onay verilirse matris elemanı bilgisi silinir. Mevcut seçili matris elemanı bu nesnelerden biri değilse silme işlemi onay alınmaksızın gerçekleştirilir. Silme işlemi sonunda ekran nesne silme olayı meydana gelir.
- ‘R’ tuşu: Mevcut seçili matris elemanı katsayıyla çarpma bloğu ise klavyenin ‘R’ tuşuna basıldığında aşağıda listelenen şartlar sağlanmış ise nesne ters yöne döndürülür. Aksi taktirde yapıştırma durumu özelliğine False Boolean değeri atanır ve yapıştırma hatası olayı meydana gelir.
 - Seçili matris elemanının sağında bulunan matris elemanının değer özelliği 0 veya 8 olmalıdır.
 - Seçili matris elemanının solunda bulunan matris elemanının değer özelliği 0, 8, 5 veya 6 olmalıdır.

Mevcut seçili matris elemanı ters yöne çevrilmiş katsayıyla çarpma bloğu ise klavyenin ‘R’ tuşuna basıldığında aşağıda listelenen şartlar sağlanmış ise nesne ters yöne döndürülür. Aksi taktirde yapıştırma durumu özelliğine False Boolean değeri atanır ve yapıştırma hatası olayı meydana gelir.

- Seçili matris elemanının sağında bulunan matris elemanının değer özelliği 0, 8, 5, 6 veya 14 olmalıdır.
- Seçili matris elemanının solunda bulunan matris elemanının değer özelliği 0, 8 veya 13 olmalıdır.

Mevcut seçili matris elemanı birim geciktirici bloğu ise klavyenin ‘R’ tuşuna basıldığında aşağıda listelenen şartlar sağlanmış ise nesne ters yöne döndürülür. Aksi takdirde yapıştırma durumu özelliğine False Boolean değeri atanır ve yapıştırma hatası olayı meydana gelir.

- Seçili matris elemanının sağında bulunan matris elemanının değer özelliği 0 veya 8 olmalıdır.
- Seçili matris elemanının solunda bulunan matris elemanının değer özelliği 0, 8, 5 veya 6 olmalıdır.

Mevcut seçili matris elemanı ters yöne çevrilmiş birim geciktirici bloğu ise klavyenin ‘R’ tuşuna basıldığında aşağıda listelenen şartlar sağlanmış ise nesne ters yöne döndürülür. Aksi takdirde yapıştırma durumu özelliğine False Boolean değeri atanır ve yapıştırma hatası olayı meydana gelir.

- Seçili matris elemanının sağında bulunan matris elemanının değer özelliği 0, 8, 5, 6 veya 14 olmalıdır.
- Seçili matris elemanının solunda bulunan matris elemanının değer özelliği 0, 8 yada 13 olmalıdır.

2.2.3 Metotlar

- Kopyala (Copy): Ekran ActiveX kontrolü üzerindeki seçili olan matris elemanı katsayıyla çarpma bloğu, birim geciktirici bloğu, cebirsel toplama bloğu, giriş elemanı veya çıkış elemanı ise bu metot icra edildiğinde matris elemanına ait bilgiler dahili değişkenlere kaydedilir.
- Kes (Cut): Ekran ActiveX kontrolü üzerindeki seçili olan matris elemanı katsayıyla çarpma bloğu, birim geciktirici bloğu, cebirsel toplama bloğu, giriş elemanı veya çıkış elemanı ise bu metot icra edildiğinde matris elemanına ait bilgiler dahili değişkenlere kaydedilir ve matris elemanı bilgileri silinir. Diğer bir deyişle matris elemanı silinmiş olur.
- Yapıştır (Paste): Bu metot icra edildiğinde ekran ActiveX kontrolü üzerindeki seçili olan matris elemanına kopyala ya da kes metodu ile dahili değişkenlere kaydedilen matris elemanı bilgileri aktarılır. Fakat matris elemanı bilgileri aktarılmadan önce bir takım kontroller yapılır. Aşağıda bu kontroller listelenmiştir. Eğer bu şartlar sağlanmaz ise

yapıştırma durumu özelliğine False Boolean değeri aktarılır ve yapıştırma hatası olayı meydana gelir. Bu olay bir hata kodu döndürür. Bu hata kodu program tarafından okunarak hatanın nedeni kullanıcıya yansıtılır.

Dahili değişkenlere bilgileri kaydedilmiş olan matris elemanı katsayıyla çarpma bloğu veya birim geciktirici bloğu ise ve ayrıca aşağıdaki şartlar da sağlanmış ise bilgi aktarım işlemi gerçekleşir.

- Seçili matris elemanı giriş elemanı ile aynı sütunda ya da solunda bulunamaz.
- Seçili matris elemanı çıkış elemanı ile aynı sütunda ya da sağında bulunamaz.
- Seçili matris elemanı en sağ ya da en sol sütunda olmamalıdır.
- Seçili matris elemanı boş (Değer özelliği 0'a eşit) olmalıdır.
- Seçili matris elemanının sağında bulunan matris elemanının değer özelliği 0, 8, 5, 6 veya 14 olmalıdır.
- Seçili matris elemanının solunda bulunan matris elemanının değer özelliği 0, 8 veya 13 olmalıdır.
- Seçili matris elemanının üstünde ve sol-üstündeki matris elemanının değer özelliği 12 olmamalıdır.
- Seçili matris elemanının altında ve sol-altındaki matris elemanının değer özelliği 12 olmamalıdır.

Dahili değişkenlere bilgileri kaydedilmiş olan matris elemanı ters yönde katsayıyla çarpma bloğu veya birim geciktirici bloğu ise ve ayrıca aşağıdaki şartlar da sağlanmış ise bilgi aktarım işlemi gerçekleşir.

- Seçili matris elemanı giriş elemanı ile aynı sütunda ya da solunda bulunamaz.
- Seçili matris elemanı çıkış elemanı ile aynı sütunda ya da sağında bulunamaz.
- Seçili matris elemanı en sağ ya da en sol sütunda olmamalıdır.
- Seçili matris elemanı boş (Değer özelliği 0'a eşit) olmalıdır.
- Seçili matris elemanının sağında bulunan matris elemanının değer özelliği 0 veya 8 olmalıdır.

- Seçili matris elemanının solunda bulunan matris elemanının değer özelliği 0, 5, 6 veya 8 olmalıdır.
- Seçili matris elemanının üstünde ve sağ-üstündeki matris elemanının değer özelliği 12 olmamalıdır.
- Seçili matris elemanının altında ve sağ-altındaki matris elemanının değer özelliği 12 olmamalıdır.

Dahili değişkenlere bilgileri kaydedilmiş olan matris elemanı cebirsel toplama bloğu ise ve ayrıca aşağıdaki şartlar da sağlanmış ise bilgi aktarım işlemi gerçekleşir.

- Seçili matris elemanı giriş elemanı ile aynı sütunda ya da solunda bulunamaz.
- Seçili matris elemanı çıkış elemanı ile aynı sütunda ya da sağında bulunamaz.
- Seçili matris elemanı en sağ veya en sol sütunda olmamalıdır.
- Seçili matris elemanı ilk veya son sütunda olmamalıdır.
- Seçili matris elemanı boş (Değer özelliği 0'a eşit) olmalıdır.
- Seçili matris elemanının sağında bulunan matris elemanının değer özelliği 0, 8, 5, 6 veya 14 olmalıdır.
- Seçili matris elemanının solunda bulunan matris elemanının değer özelliği 0, 8 veya 13 olmalıdır.
- Seçili matris elemanının üstünde bulunan matris elemanının değer özelliği 1, 2, 3 veya 4 olmamalıdır.
- Seçili matris elemanının altında bulunan matris elemanının değer özelliği 1, 2, 3 veya 4 olmamalıdır.
- Seçili matris elemanının sağ-üstünde bulunan matris elemanının değer özelliği 1, 2, 3 veya 4 olmamalıdır.
- Seçili matris elemanının sağ-altında bulunan matris elemanının değer özelliği 1, 2, 3 veya 4 olmamalıdır.
- Seçili matris elemanının etrafındaki hiçbir matris elemanının değer özelliği 12 olmamalıdır.

Dahili deęiřkenlere bilgileri kaydedilmiř olan matris elemanı giriř elemanı ise ve ayrıca ařaęıdaki řartlar da saęlanmıř ise bilgi aktarım iřlemi gerekleřir.

- Matriste daha nce eklenmiř bir giriř elemanı nesnesi olmamalıdır.
- Seili matris elemanı ile aynı stunda ya da solunda daha nce eklenmiř bir nesne bulunmamalıdır.
- Seili matris elemanı boř (Deęer zellięi 0'a eřit) olmalıdır.

Dahili deęiřkenlere bilgileri kaydedilmiř olan matris elemanı ıkıř elemanı ise ve ayrıca ařaęıdaki řartlar da saęlanmıř ise bilgi aktarım iřlemi gerekleřir.

- Matriste daha nce eklenmiř bir ıkıř elemanı nesnesi olmamalıdır.
- Seili matris elemanı ile aynı stunda ya da saęında daha nce eklenmiř bir nesne bulunmamalıdır.
- Seili matris elemanı boř (Deęer zellięi 0'a eřit) olmalıdır.
- Temizle (Clear): Ekran ActiveX kontrol zerinde bulunan tm matris elemanlarının bilgilerinin silinmesini saęlar. Dięer bir deyiřle tm matris elemanlarını siler.
- Seimi Kaldırma (Deselect): Ekran ActiveX kontrol zerinde bulunan tm matris elemanlarını seili olmayan duruma getirir.

3. EDİTÖR YAPISI

BDD editörü genel olarak kullanıcıya blok diyagram çizmeye olanak sağladığı çalışma alanı, araç çubukları ve çıkış penceresinden oluşmaktadır. Şekil 3.1’de blok diyagram derleyici editörü gösterilmiştir.



Şekil 3-1 Blok diyagram derleyici editörü

Çalışma Alanı: Çalışma alanı 11x27 boyutunda bir matristen oluşmaktadır. Araç çubukları kullanılarak istenilen matris elemanına belirli bir nesne yerleştirilebilir. Böylelikle istenilen blok diyagram çalışma alanında oluşturulur. Görüntüsü değişen her matris elemanının buna karşılık gelen ve derleme esnasında program tarafından kullanılacak olan sayısal değeri de değişir.

Çıkış Penceresi: Blok diyagram oluşturma esnasında ve daha sonrasında kullanıcıya, yapılan işlemlerle ilgili bilgi vermek amacıyla kullanılır.

3.1 Menü Yapısı

BDD editörünün menüsü Dosya (File), Düzenle (Edit), Ekle (Insert), Proje (Project), Transfer (Transfer) ve Yardım (Help) seçeneklerinden oluşmaktadır. Bu seçeneklerden bazılarını aynı zamanda araç çubuklarından da ulaşılabilir.

3.1.1 Dosya (File)

BDD editöründe oluşturulan blok diyagram ve proje ile ilgili yapılan diğer ayarlar daha sonra tekrar kullanılmak üzere sabit diske kaydedilebilir. Projeler FBD uzantılı bir dosya olarak diske kaydedilir.

- Yeni (New): Yeni, boş bir proje açmayı sağlar. (Kısayol: Cntrl + N)
 - ‘New’ seçeneği tıklandığında editörde boş bir proje açık ise yeni, boş bir proje açılır.
 - ‘New’ seçeneği tıklandığında editörde boş bir proje açık ve bu proje üzerinde değişiklik yapıldıysa ‘Save’ (Kaydet) diyalog penceresi ekrana gelir (Şekil 3.2).

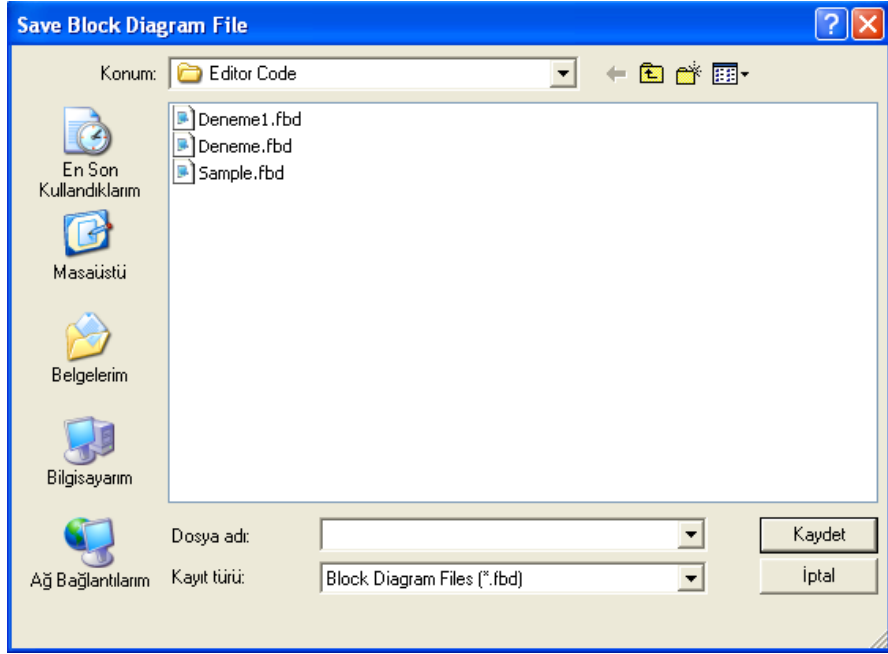


Şekil 3-2 Kaydet diyalog penceresi

Evet butonuna tıklanırsa ‘Save Block Diagram File (Save as)’ (Blok diyagram dosyasını farklı kaydet) diyalog penceresi ekrana gelir ve kullanıcıdan FBD uzantılı proje dosyasının adının ve sabit diskte kaydedileceği yerin girilmesi beklenir. Dosya kaydedildikten sonra yeni, boş bir proje açılır.

Hayır butonu tıklanırsa editör, üzerinde değişiklik yapılmış olan bu boş projeyi kaydetmeden yeni, boş bir proje açar.

‘Save Block Diagram File (Save as)’ diyalog penceresi açıldığında proje dosyalarının kaydedileceği varsayılan dosya yolu, uygulamanın bulunduğu klasördür. Şekil 3.3’te ‘Save Block Diagram File (Save as)’ diyalog penceresi gösterilmiştir.



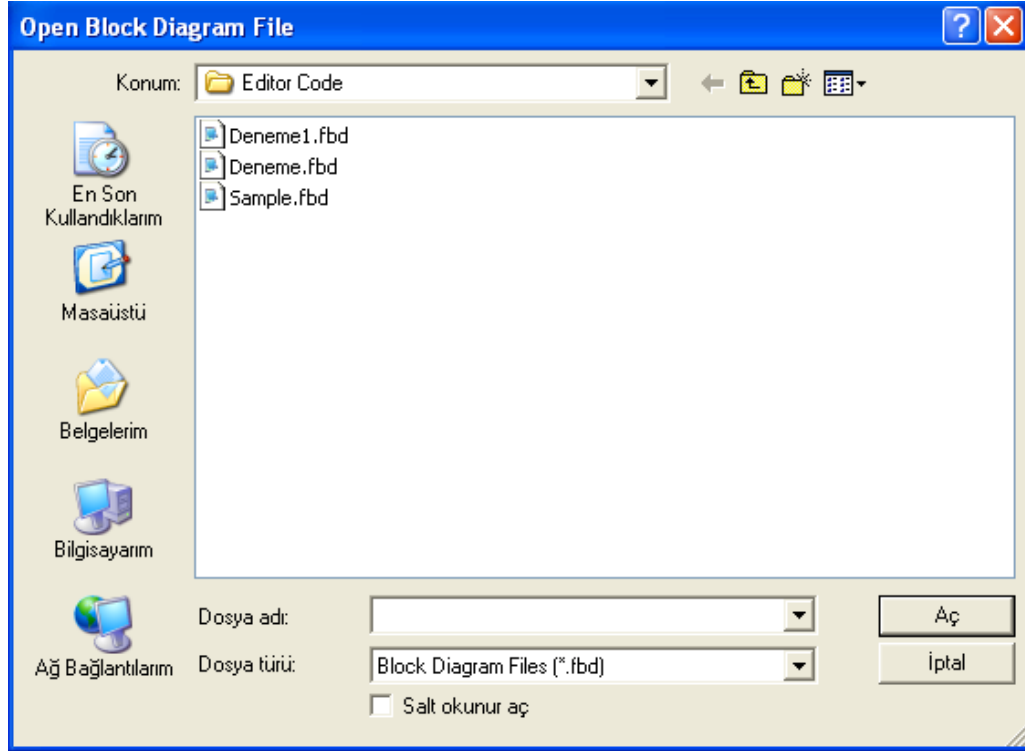
Şekil 3-3 Farklı kaydet diyalog penceresi

- ‘New’ seçeneği tıklandığında editörde daha önce sabit diske kaydedilmiş bir proje açık ise ‘Save’ diyalog penceresi ekrana gelir.

Evet butonuna tıklanırsa kullanıcının projede yapmış olduğu değişiklikler FBD uzantılı proje dosyasına kaydedilir ve yeni, boş bir proje açılır.

Hayır butonu tıklanırsa yapılan değişiklikler kaydedilmeden yeni, boş bir proje açılır.

- Aç (Open): Daha önce sabit diske kaydedilmiş olan bir projeyi açmayı sağlar. (Kısayol: Cntrl + O)
- ‘Open’ seçeneği tıklandığında editörde boş bir proje açık ise ‘Open Block Diagram File’ (Blok diyagram dosyasını aç) diyalog penceresi ekrana gelir ve kullanıcının daha önce sabit diske kaydedilmiş bir proje dosyasını seçmesi beklenir.



Şekil 3-4 Blok diyagram dosyası aç diyalog penceresi

‘Open Block Diagram File’ diyalog penceresi açıldığında proje dosyalarının açılacağı varsayılan dosya yolu, uygulamanın bulunduğu klasördür. Şekil 3.4’te ‘Open Block Diagram File’ diyalog penceresi gösterilmiştir.

- ‘Open’ seçeneği tıklandığında editörde boş bir proje açık ve bu proje üzerinde değişiklik yapıldıysa ‘Save’ diyalog penceresi ekrana gelir.

Evet butonuna tıklanırsa ‘Save Block Diagram File (Save as)’ diyalog penceresi ekrana gelir ve kullanıcıdan FBD uzantılı proje dosyasının adının ve sabit diskte kaydedileceği yerin girilmesi beklenir. Proje kaydedildikten sonra sabit diske daha önceden kaydedilmiş bir projeyi açmak üzere ‘Open Block Diagram File’ diyalog penceresini ekrana gelir.

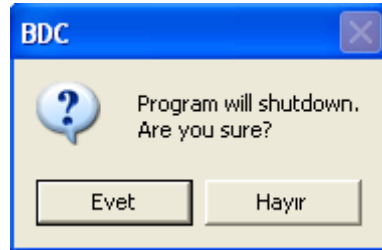
Hayır butonu tıklanırsa editör, üzerinde değişiklik yapılmış olan bu boş projeyi kaydetmeden sabit diske daha önceden kaydedilmiş bir projeyi açmak üzere ‘Open Block Diagram File’ diyalog penceresini ekrana getirir.

- Open seçeneği tıklandığında editörde daha önce sabit diske kaydedilmiş bir proje açık ise ‘Save’ diyalog penceresi ekrana gelir.

Evet butonuna tıklanırsa kullanıcının projede yapmış olduğu değişiklikler FBD uzantılı proje dosyasına kaydedilir ve ‘Open Block Diagram File’ diyalog penceresini ekrana gelir.

Hayır butonu tıklanırsa yapılan değişiklikler kaydedilmeden ‘Open Block Diagram File’ diyalog penceresi ekrana gelir.

- **Kaydet (Save):** Daha önce sabit diske kaydedilmiş olan bir proje dosyası üzerinde çalışılıyorsa yapılan değişikliklerin FBD uzantılı proje dosyasına kaydedilmesini sağlar. Üzerinde çalışılan proje daha önce sabit diske kaydedilmiş bir proje değilse ‘Save Block Diagram File (Save as)’ diyalog penceresi ekrana gelir ve kullanıcıdan FBD uzantılı proje dosyasının adının ve sabit diskte kaydedileceği yerin girilmesi beklenir. (Kısayol: Cntrl + S)
- **Farklı kaydet (Save As):** Üzerinde çalışılan projeyi farklı isimde sabit diske kaydetmeyi sağlar. Seçenek tıklandığında ‘Save Block Diagram File (Save as)’ diyalog penceresi ekrana gelir ve kullanıcıdan FBD uzantılı proje dosyasının adının ve sabit diskte kaydedileceği yerin girilmesi beklenir.
- **Çıkış (Exit):** Blok diyagram derleyici editörünü kapatır. Program kapatılmadan önce kullanıcıdan teyit alınır (Şekil 3.5). (Kısayol: Cntrl + Q)



Şekil 3-5 Editör kapatma teyit diyalog penceresi

3.1.2 Düzenle (Edit)

BDD editörü çalışma alanında klasik kes (cut), kopyala (copy) ve yapıştır (paste) işlemlerini destekler. Bu da blok diyagram çiziminde kullanıcıya kolaylık sağlar. ‘Cut’, ‘Copy’ ve ‘Paste’ seçeneklerine çalışma alanına sağ tıklanarak da ulaşılabilir. Ayrıca ekranı temizle (Clear Screen) seçeneği ile ekrandaki tüm matris elemanlarının içerikleri silinebilir.

- **Kes (Cut):** Seçili olan matris elemanının içeriğini kayıt eder ve daha sonra matris elemanı içeriğini siler. ‘Cut’ işlemi başarılı bir şekilde tamamlandığında menüde bulunan ‘Paste’ seçeneği aktif hale, ‘Compile’ seçeneği ise aktif ise deaktif hale gelir. Bunun nedeni ‘Cut’

işlemi sonucunda blok diyagramda değişiklik yapıldığından derleme işlemi öncesi tekrar ‘Verify’ seçeneğine basılarak blok diyagramın doğruluğunun kontrol edilmesi gerekliliğidir. (Kısayol: Cntrl + X)

- Kopyala (Copy): Seçili olan matris elemanının içeriğini kayıt eder. ‘Copy’ işlemi başarılı bir şekilde tamamlandığında menüde bulunan ‘Paste’ seçeneği aktif hale gelir. (Kısayol: Cntrl + C)
- Yapıştır (Paste): ‘Cut’ ya da ‘Copy’ seçenekleri kullanılarak kaydedilmiş matris elemanının içeriğini seçili olan matris elemanına aktarmayı sağlar. ‘Paste’ işlemi başarılı bir şekilde tamamlandığında menüde bulunan ‘Compile’ seçeneği aktif ise deaktif hale gelir. Bunun nedeni ‘Paste’ işlemi sonucunda blok diyagramda değişiklik yapıldığından derleme işlemi öncesi tekrar ‘Verify’ seçeneğine basılarak blok diyagramın doğruluğunun kontrol edilmesi gerekliliğidir. (Kısayol: Cntrl + V)

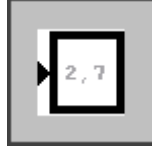
Not: ‘Cut’ ve ‘Copy’ işlemleri sadece seçili olan matris elemanı katsayıyla çarpma bloğu, birim geciktirici bloğu, cebirsel toplama bloğu, giriş elemanı ya da çıkış elemanı ise geçerlidir.

- Ekranı Temizle (Clear Screen): Mevcut çalışma alanındaki tüm matris elemanlarının içeriklerini silmeyi sağlar.

3.1.3 Ekle (Insert)

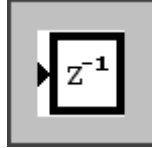
BDD editörü çalışma alanına blok diyagram oluşturmak üzere aşağıda listelenen nesneler eklenebilir. Nesne ekleme işlemi başarılı bir şekilde tamamlandığında ‘Compile’ seçeneği deaktif hale gelir. Bunun nedeni nesne ekleme işlemi sonucunda blok diyagramda değişiklik yapıldığından derleme işlemi öncesi tekrar ‘Verify’ seçeneğine basılarak blok diyagramın doğruluğunun kontrol edilmesi gerekliliğidir. ‘Insert’ menüsüne çalışma alanına sağ tıklanarak da ulaşılabilir. Bu menü kullanılarak çalışma alanına katsayıyla çarpma bloğu (Coefficient Unit), birim geciktirici bloğu (Delay Unit), cebirsel toplama bloğu (Adder Unit), giriş elemanı (Input Unit) ve çıkış elemanı (Output Unit) eklenebilir.

- Katsayıyla çarpma bloğu (Coefficient Unit): Seçili olan matris elemanına katsayıyla çarpma bloğu eklemeyi sağlar (Şekil 3.6).



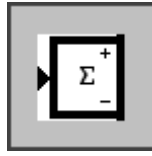
Şekil 3-6 Katsayıyla çarpma bloğu

- Birim geciktirici bloğu (Delay Unit): Seçili olan matris elemanına birim geciktirici bloğu eklemeyi sağlar (Şekil 3.7).



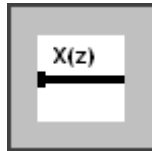
Şekil 3-7 Birim geciktirici bloğu

- Cebirsel toplama bloğu (Adder Unit): Seçili olan matris elemanına toplayıcı elemanı eklemeyi sağlar (Şekil 3.8).



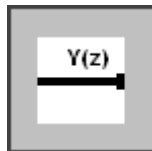
Şekil 3-8 Cebirsel toplama bloğu

- Giriş elemanı (Input Unit): Seçili olan matris elemanına giriş elemanı eklemeyi sağlar (Şekil 3.9).



Şekil 3-9 Giriş elemanı

- Çıkış elemanı (Output Unit): Seçili olan matris elemanına çıkış elemanı eklemeyi sağlar (Şekil 3.10).

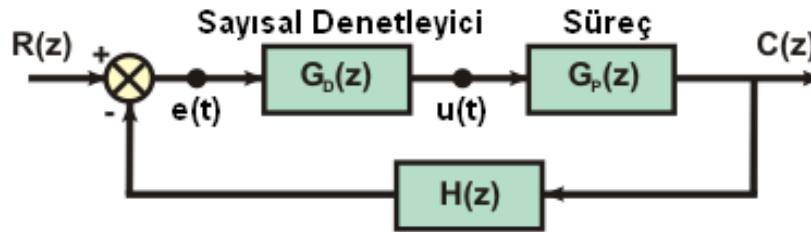


Şekil 3-10 Çıkış elemanı

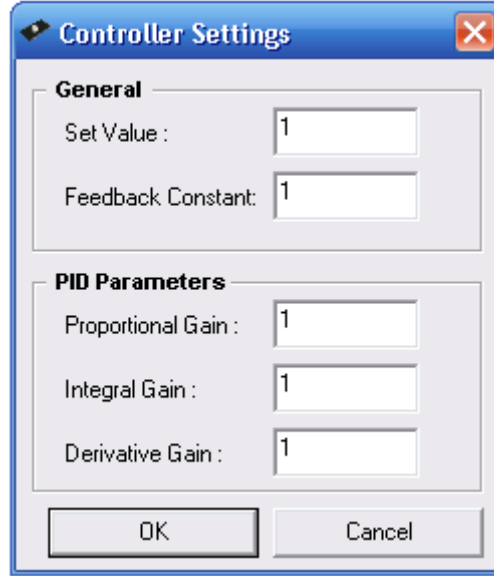
3.1.4 Proje (Project)

- Doğrula (Verify): BDD editörü çalışma alanına blok diyagram çizildikten sonra çizilen bu blok diyagramın geçerliliği 'Verify' seçeneğine tıklanarak doğrulanır. Eğer hiç bir sorun bulunamaz ise çizilen blok diyagramın geçerli olduğu kullanıcıya yansıtılır ve 'Compile' seçeneği aktif hale gelir. (Kısayol: Cntrl + B)
- Derle (Compile): BDD editörünün çalışma alanına çizilen blok diyagram geçerliliği 'Verify' seçeneği ile doğrulandıktan sonra aktif olan 'Compile' seçeneği tıklanarak blok diyagram fark denklemine dönüştürülür. Bu fark denklemi çizilen blok diyagramı temsil eden fark denklemdir. (Kısayol: Cntrl + P)
- Denetleyici Ayarları (Controller Settings): BDD editörünün çalışma alanına çizilen blok diyagram, belirli bir sistemi kontrol edecek olan bir sayısal denetleyiciyi temsil etmektedir. Bu yüzden referans değeri ve geribesleme katsayı değeri de sisteme girilmelidir. Böylece sayısal denetleyicinin girişine uygulanacak olan değer elde edilmiş olur. Şekil 3.11'de sistemin blok diyagramı gösterilmiştir. (Varsayılan: Set Value (Referans Değeri) = 1, Feedback Constant (Geribesleme Katsayısı) = 1); (Şekil 3.12)

Sayısal denetleyici girişi = Referans değeri – (Süreç çıkışı * Geribesleme transfer fonksiyonu)



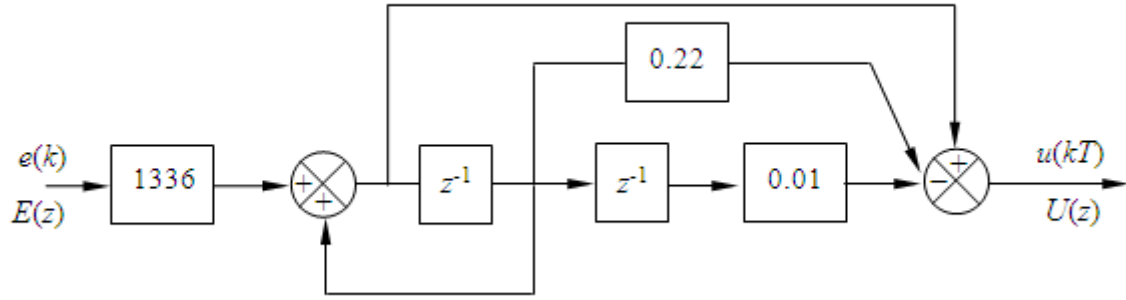
Şekil 3-11 Kontrol sitemi blok diyagramı



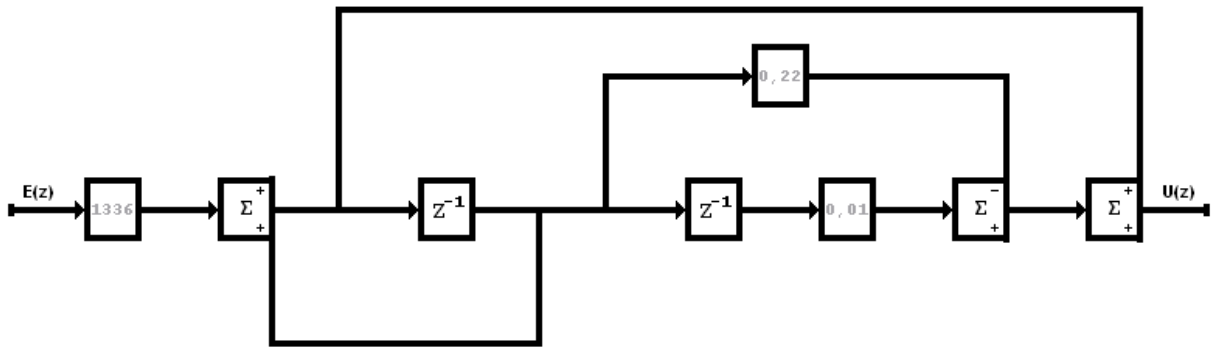
Şekil 3-12 Denetleyici ayarları diyalog penceresi

Diyalog penceresi içindeki metin kutuları doldurulurken kullanıcının geçersiz veri girmesini engellemek için bir takım kontroller yapılır. Yapılan bu kontroller aşağıda listelenmiştir.

- Veri girişi esnasında klavyenin sadece rakam, virgöl, ‘delete’, ‘backspace’, ‘escape’ ve ‘enter’ tuşları işlevlidir.
- Girilen veri en fazla bir adet virgöl içerebilir.
- Virgöl en başta ya da en sonda olamaz.
- Metin kutuları boş bırakılamaz
- PID Blok Diyagramı (PID Block Diagram): BDD editörü blok diyagram çizmeye olanak sağladığı gibi kullanıcıya hazır bir PID blok diyagramı sunarak PID denetleyici tasarım süresini kısaltır. Şekil 3.13a’ da verilen blok diyagramı geliştirilen BDD editöründe Şekil 3.13b’ deki gibi oluşturulur. Kullanıcı, ‘Controller Settings’ diyalog penceresinde (Şekil 3.12) bulunan ‘Proportional Gain’ (Oransal kazanç), ‘Integral Gain’ (İntegral kazancı) ve ‘Derivative Gain’ (Türev kazancı) parametrelerini değiştirerek farklı kazanç değerlerine karşılık gelen PID blok diyagramını kolayca elde edebilir. Bu da tasarım süresini oldukça kısaltır.

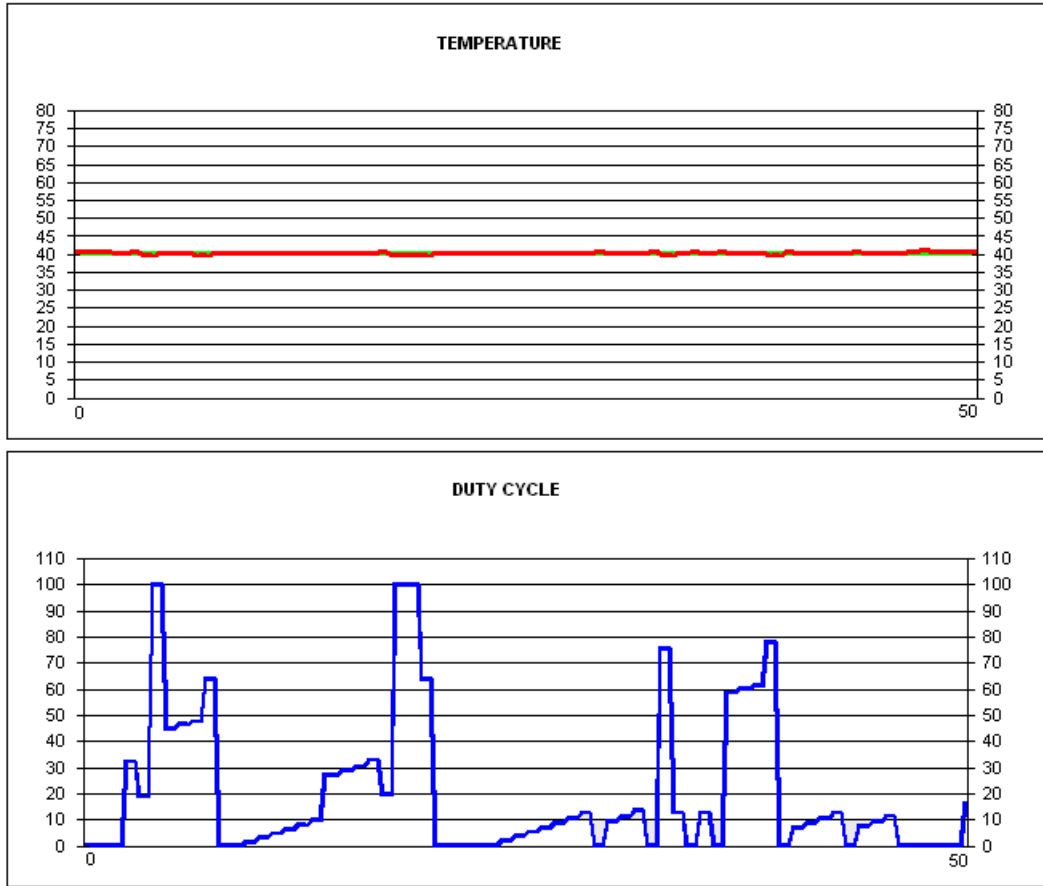


Şekil 3-13a PID blok diyagramı



Şekil 3-14b BDD editöründe oluşturulmuş PID blok diyagramı

- Grafik (Graphic): Grafik diyalog penceresi (Şekil 3.14), BDD editörü hazır PID blok diyagramı kullanılarak oluşturulmuş olan örnek sıcaklık kontrol sisteminin performansını incelemeye olanak sağlar. Diyalog penceresinde iki adet grafik mevcuttur: Sıcaklık (Süreç çıkışı) ve Doluluk/Boşluk Oranı (PID blok diyagramı çıkışı). Yatay eksenler zaman (sn), dikey eksenler ise sırasıyla sıcaklık ($^{\circ}\text{C}$) ve doluluk/boşluk oranıdır (%).

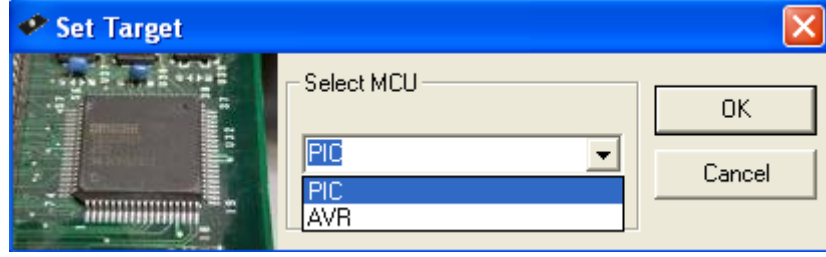


Şekil 3-15 Grafik diyalog penceresi. Yatay eksenler zaman (sn), dikey eksenler ise üstteki grafikte sıcaklık ($^{\circ}\text{C}$), alttakinde ise doluluk/boşluk oranıdır (%).

3.1.5 Transfer (Transfer)

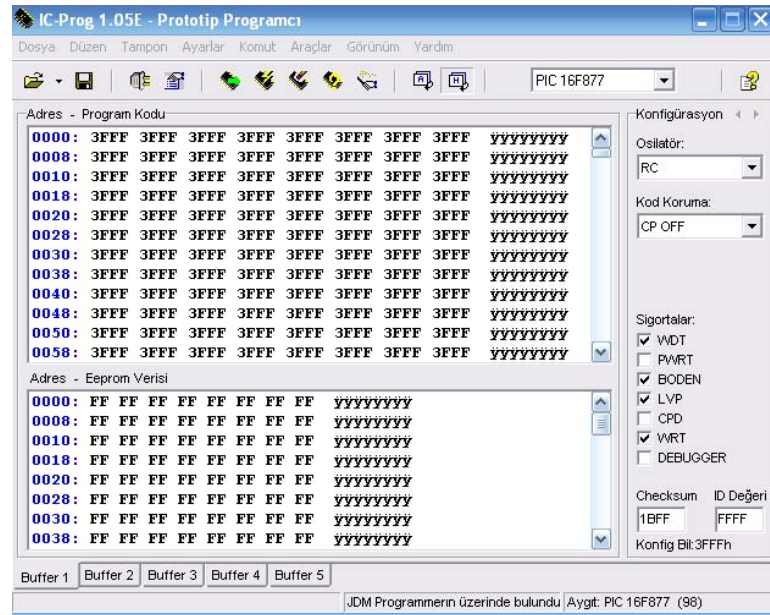
‘Compile’ seçeneği tıklanarak fark denklemine dönüştürülen blok diyagram bu menü altındaki seçenekler kullanılarak mikroişlemciye yüklenebilir.

- Hedefi Belirle (Set Target): Çizilen blok diyagram ‘Compile’ seçeneği tıklanarak derlenmeden önce hangi mikroişlemci koduna dönüştürülmesi isteniyorsa ‘Set Target’ diyalog penceresinden istenilen mikroişlemcinin seçilmesi gerekir. Şekil 3.15’te ‘Set Target’ diyalog penceresi gösterilmiştir. Eğer derleme işleminden sonra ‘Set Target’ diyalog penceresi açılırsa derleme işleminin tekrar gerçekleştirilmesi gerekir (Varsayılan: PIC).



Şekil 3-16 Hedefi belirle diyalog penceresi

- Projeyi Mikroişlemciye Yükle (Transfer Project to MCU): Derlenmiş olan blok diyagramın seçilen mikroişlemciye yüklenmesini sağlar. Yükleme işlemi PIC mikroişlemci için IC-Prog programı ile gerçekleştirilir (Şekil 3.16). IC-Prog programı HEX dosyalarının mikroişlemciye yüklenmesini sağlar. Bu program blok diyagram derleyici editörü içerisinde gömülüdür.

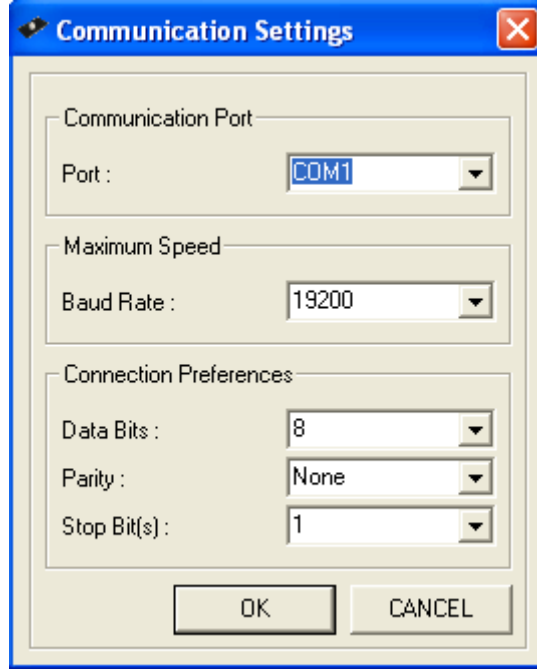


Şekil 3-17 IC-Prog programı

- Portu Aç (Port Open): BDD editöründe kullanıcıya hazır olarak sunulan PID blok diyagramının PID parametrelerinin ayarlanıp mikroişlemciye yüklenmesi sonucunda gerçekleştirilen sayısal denetleyicinin performansını 'Graphic' kısmında bulunan grafiklerden izleyebiliriz. Fakat bu grafiklere (öncelikle bilgisayara) mikroişlemci tarafından veri aktarılabilmesi için bilgisayarın haberleşme portu açık olmalıdır. Çünkü denetleyiciye ait bu gerçek zamanlı veriler mikroişlemci tarafından bilgisayara seri port üzerinden aktarılır. 'Port Open' seçeneği tıklandığında 'Communication Parameters' diyalog penceresi

kullanılarak ayarlanmış olan haberleşme parametrelerine göre ilgili port haberleşmeye hazır hale getirilir. (Kısayol: F7)

- Haberleşme Parametreleri (Communication Parameters): ‘Communication Parameters’ diyalog penceresi kullanılarak aktarım işlemi için gerekli seri port parametre ayarları yapılabilir. Şekil 3.17’de ‘Communication Parameters’ diyalog penceresi gösterilmiştir. Haberleşme ayarlarının yapılabilmesi için seri portun kapalı olması gerekmektedir. (Varsayılan: Port = 1, Baud Rate = 9600, Parity = "N", Data Bits = 8, Stop Bits = 1)



Şekil 3-18 Haberleşme ayarları diyalog penceresi

3.1.6 Yardım (Help)

Programın kullanımı ile ilgili yardım menüsüne buradan ulaşılır.

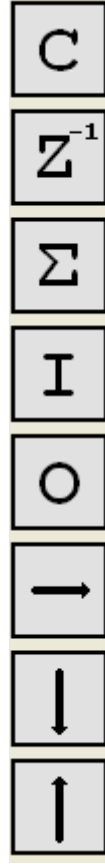
3.2 Nesne Araç Çubuğu

BDD editörü çalışma alanında blok diyagram oluşturmak üzere nesneler, nesne araç çubuğu kullanılarak eklenebilir. Şekil 3.18’de nesne araç çubuğu gösterilmiştir. Ara bağlantılar için kullanılan bağlantı elemanları klavyenin belirli tuşlarına basılarak da çizilebilir.

- Yukarı doğru bağlantı elemanı: Kısayol = ‘A’ yada ‘a’
- Aşağı doğru bağlantı elemanı = ‘D’ yada ‘d’

- Yatay bağlantı elemanı = 'S' yada 's'

Nesne ve bağlantı elemanı ekleme-silme işlemi başarılı bir şekilde tamamlandığında 'Compile' seçeneği deaktif hale gelir. Bunun nedeni nesne ekleme işlemi sonucunda blok diyagramda değişiklik yapıldığından derleme işlemi öncesi tekrar 'Verify' seçeneğine basılarak blok diyagramın doğruluğunun kontrol edilmesi gerekliliğidir.



Şekil 3-19 Nesne araç çubuğu

3.3 Durum Çubuğu

Durum çubuğu haberleşme parametrelerini, seri portun durumunu, mevcut işlemci tipini ve zaman-tarih bilgisini kullanıcıya yansıtır. Şekil 3.19'da durum çubuğu gösterilmiştir.



Şekil 3-20 Durum çubuğu

4. BLOK DİYAGRAM OLUŞTURMA

Daha önce de bahsedildiği gibi ekran 11x27 boyutunda bir matrsten oluşmaktadır ve her matris elemanının sayısal bir değeri vardır. Böylece kullanıcının çizmiş olduğu blok diyagram, bir matrise dönüştürülmüş olur. Bu matriste program içerisinde blok diyagramın hata kontrolü ve derlenmesi esnasında kullanılır. Çalışma alanına bir nesne eklenmeden önce bu nesnenin eklenmek istendiği yerin yine bu nesnenin eklenmesi için uygun olup olmadığı program tarafından kontrol edilir. Kullanıcı, çalışma alanına nesne eklerken yanlış blok diyagram oluşturmasını engellemek için program tarafından bazı kısıtlamalar getirilmiştir. Bu kısıtlamalar aşağıda listelenmiştir.

- Giriş elemanı (Input Unit)
 - Çalışma alanına sadece bir tane ‘Input Unit’ eklenebilir.
 - Aynı sütuna ya da sol tarafındaki sütunlara eleman eklenemez.
 - Çalışma alanında ‘Output Unit’ mevcut ise bu nesne ile aynı satırda olmalıdır.
 - Eklenilmeye çalışıldığı matris elemanı boş (Değer özelliği 0’a eşit) olmalıdır.
- Çıkış elemanı (Output Unit)
 - Çalışma alanına sadece bir tane ‘Output Unit’ eklenebilir.
 - Aynı sütuna ya da sağ tarafındaki sütunlara eleman eklenemez.
 - Çalışma alanında ‘Input Unit’ mevcut ise bu nesne ile aynı satırda olmalıdır.
 - Eklenilmeye çalışıldığı matris elemanı boş (Değer özelliği 0’a eşit) olmalıdır.
- Katsayıyla çarpma bloğu (Coefficient Unit)
 - ‘Input Unit’ ile aynı sütuna ya da sol tarafındaki sütunlara eklenemez.
 - ‘Output Unit’ ile aynı sütuna ya da sağ tarafındaki sütunlara eklenemez.
 - En sol ya da en sağ sütuna eklenemez.
 - Eklenilmeye çalışıldığı matris elemanı boş (Değer özelliği 0’a eşit) olmalıdır.
 - Seçili matris elemanının sağında bulunan matris elemanının değer özelliği 0, 8, 5, 6 veya 14 olmalıdır.
 - Seçili matris elemanının solunda bulunan matris elemanının değer özelliği 0, 8 veya 13 olmalıdır.

- Seçili matris elemanının üst, alt, sol-üst ve sol-altında bulunan matris elemanlarının değer özelliği 12 olmamalıdır.
- Birim geciktirici bloğu (Delay Unit)
 - ‘Input Unit’ ile aynı sütuna ya da sol tarafındaki sütunlara eklenemez.
 - ‘Output Unit’ ile aynı sütuna ya da sağ tarafındaki sütunlara eklenemez.
 - En sol yada en sağ sütuna eklenemez.
 - Eklenmeye çalışıldığı matris elemanı boş (Değer özelliği 0’a eşit) olmalıdır.
 - Seçili matris elemanının sağında bulunan matris elemanının değer özelliği 0, 8, 5, 6 veya 14 olmalıdır.
 - Seçili matris elemanının solunda bulunan matris elemanının değer özelliği 0, 8 veya 13 olmalıdır.
 - Seçili matris elemanının üst, alt, sol-üst ve sol-altında bulunan matris elemanlarının değer özelliği 12 olmamalıdır.
- Cebirsel toplama bloğu (Adder Unit)
 - Seçili matris elemanı ‘Input Unit’ ile aynı sütunda ya da solunda bulunamaz.
 - Seçili matris elemanı ‘Output Unit’ ile aynı sütunda ya da sağında bulunamaz.
 - Seçili matris elemanı en sağ veya en sol sütunda olmamalıdır.
 - Seçili matris elemanı ilk veya son sütunda olmamalıdır.
 - Seçili matris elemanı boş (Değer özelliği 0’a eşit) olmalıdır.
 - Seçili matris elemanının sağında bulunan matris elemanının değer özelliği 0, 8, 5, 6 veya 14 olmalıdır.
 - Seçili matris elemanının solunda bulunan matris elemanının değer özelliği 0, 8 veya 13 olmalıdır.
 - Seçili matris elemanının üstünde bulunan matris elemanının değer özelliği 1, 2, 3 veya 4 olmamalıdır.
 - Seçili matris elemanının altında bulunan matris elemanının değer özelliği 1, 2, 3 veya 4 olmamalıdır.

- Seçili matris elemanının sağ-üstünde bulunan matris elemanının değer özelliği 1, 2, 3 veya 4 olmamalıdır.
- Seçili matris elemanının sağ-altında bulunan matris elemanının değer özelliği 1, 2, 3 veya 4 olmamalıdır.
- Seçili matris elemanının etrafındaki hiçbir matris elemanının değer özelliği 12 olmamalıdır.
- Yatay Bağlantı Elemanı
 - Seçili matris elemanı 'Input Unit' ile aynı sütunda ya da solunda bulunamaz.
 - Seçili matris elemanı 'Output Unit' ile aynı sütunda ya da sağında bulunamaz.
- Yukarı Doğru Dikey Bağlantı Elemanı
 - Seçili matris elemanı 'Input Unit' ile aynı sütunda ya da solunda bulunamaz.
 - Seçili matris elemanı 'Output Unit' ile aynı sütunda ya da sağında bulunamaz.
 - Seçili matris elemanı en sağ veya en sol sütunda olmamalıdır.
 - Seçili matris elemanı ilk sütunda olmamalıdır.
 - Seçili matris elemanının sağ ve sağ üstünde bulunan matris elemanının değer özelliği 1, 2, 3, 4 veya 12 olmamalıdır.
- Aşağı Doğru Dikey Bağlantı Elemanı
 - Seçili matris elemanı 'Input Unit' ile aynı sütunda ya da solunda bulunamaz.
 - Seçili matris elemanı 'Output Unit' ile aynı sütunda ya da sağında bulunamaz.
 - Seçili matris elemanı en sağ veya en sol sütunda olmamalıdır.
 - Seçili matris elemanı son sütunda olmamalıdır.
 - Seçili matris elemanının sağ ve sağ altında bulunan matris elemanının değer özelliği 1, 2, 3, 4 veya 12 olmamalıdır.

4.1 Aşağı-Yukarı Doğru Dikey Bağlantı Elemanı Değişim Çizelgesi

BDD editörü, çalışma alanına eklenen blok diyagram nesnelerinin birbirine olan bağlantılarının yapılması için yukarı ve aşağı doğru dikey bağlantı elemanı çizmeye olanak sağlar. Fakat bu dikey bağlantı elemanları yatay bağlantı elemanları ile birleştiklerinde yatay

tam ya da yarım ‘T’ şekline dönüşmelidir. Bunu sağlamak için Çizelge 4.1’ de gösterilen dönüşüm çizelgesi oluşturulmuştur.

Çizelgede, “Aşağı Doğru Dikey Bağlantı Elemanı” çizilmek istendiğinde aktif matris elemanı “Üst Hücre” olarak adlandırılan sütun; “Yukarı Doğru Dikey Bağlantı Elemanı” çizilmek istendiğinde aktif matris elemanı “Alt Hücre” olarak adlandırılan sütun olarak belirtilmiştir. Böylece her iki durum tek bir çizelgeyle gösterilmiş olur.

Çizelge 4-1 Aşağı-yukarı doğru dikey bağlantı elemanı değişim çizelgesi

Önceki Durum		Sonraki Durum	
Alt Hücre	Üst Hücre	Alt Hücre	Üst Hücre
0	0	10	11
5	0	5	11
6	0	7	11
7	0	7	11
8	0	5	11
9	0	9	11
10	0	10	11
11	0	9	11
12	0	12	11
0	5	10	7
5	5	5	7
6	5	7	7
7	5	7	7
8	5	5	7
9	5	9	7
10	5	10	7
11	5	9	7
0	6	10	6
6	6	7	6
8	6	5	6
11	6	9	6
0	7	10	7
8	7	5	7
11	7	9	7
0	8	10	6
5	8	5	6
6	8	7	6

7	8	7	6
8	8	5	6
9	8	9	6
10	8	10	6
11	8	9	6
12	8	12	6
0	9	10	9
6	9	7	9
8	9	5	9
11	9	9	9
0	10	10	9
5	10	5	9
6	10	7	9
7	10	7	9
8	10	5	9
9	10	9	9
10	10	10	9
11	10	9	9
12	10	12	9
0	11	10	11
6	11	7	11
8	11	5	11
11	11	9	11
0	12	10	12
8	12	5	12
11	12	9	12

4.2 Yatay Bağlantı Elemanı Değişim Çizelgesi

BDD editörü, çalışma alanına eklenen blok diyagram nesnelerinin birbirlerine olan bağlantılarının yapılabilmesi için yatay bağlantı elemanı çizmeye olanak sağlar. Fakat bu yatay bağlantı elemanları dikey bağlantı elemanlarıyla birleştiklerinde yatay tam yada yarım ‘T’ şekline dönüşmelidir. Bunu sağlamak için Çizelge 4.2’ de gösterilen dönüşüm çizelgesi oluşturulmuştur.

Çizelge 4-2 Yatay bağlantı elemanı değişim çizelgesi

Önceki Durum	Sonraki Durum
0	8
9	7
10	5
11	6

4.3 Çizilen Blok Diyagramın Kontrol Edilmesi

Editörün çalışma alanına çizilmiş olan blok diyagram, derlenmeden önce kontrol edilmelidir. Diğer bir deyişle çizilen blok diyagramın geçerliliği program tarafından doğrulanmalıdır. Doğrulama işlemi menüde bulunan “Project -> Verify” seçeneğinin tıklanması ile gerçekleşir. (Kısayol: Cntrl + B)

Doğrulama işlemi başarılı bir şekilde tamamlandığında “Compile” seçeneği aktif hale gelir. Eğer doğrulama esnasında herhangi bir hata tespit edilirse, bu hatanın nedeni ve hatanın tespit edildiği matris elemanı kullanıcıya bildirilir. Bu durumda “Compile” seçeneği deaktif durumda olacaktır. Sonuç olarak derleme işlemine geçilebilmesi için “Verify” seçeneğinin tıklanıp blok diyagramın doğruluğunun programca onaylanması gerekir. “Verify” işlemi gerçekleşirken yapılan kontroller ve sonuçları Çıkış penceresi (Output Window) vasıtasıyla kullanıcıya yansıtılır. Şekil 4.1’de çıkış penceresi gösterilmiştir.



Şekil 4-1 Çıkış penceresi

Doğrulama işlemi esnasında yapılan kontroller aşağıda listelenmiştir.

- Blok diyagram bir adet ‘Input Unit’ nesnesi içermelidir. Blok diyagramda hiç ‘Input Unit’ olmaması veya birden fazla olması hataya neden olur.
- Blok diyagram bir adet ‘Output Unit’ nesnesi içermelidir. Blok diyagramda hiç ‘Output Unit’ olmaması veya birden fazla olması hataya neden olur.

- Blok diyagramda bulunan ‘Coefficient Unit’ ve ‘Delay Unit’ nesnelerinin solunda bulunan matris elemanının değer özelliği 8 ya da 13 olmalıdır.
- Blok diyagramda bulunan ‘Coefficient Unit’ ve ‘Delay Unit’ nesnelerinin sağında bulunan matris elemanının değer özelliği 5, 6, 8 ya da 14 olmalıdır.
- Blok diyagramda bulunan ters yöndeki ‘Coefficient Unit’ ve ‘Delay Unit’ nesnelerinin solunda bulunan matris elemanının değer özelliği 5, 6 ya da 8 olmalıdır.
- Blok diyagramda bulunan ters yöndeki ‘Coefficient Unit’ ve ‘Delay Unit’ nesnelerinin sağında bulunan matris elemanının değer özelliği 8 olmalıdır.
- Blok diyagramda bulunan ‘Adder Unit’ nesnelerinin solunda bulunan matris elemanının değer özelliği 8 ya da 13 olmalıdır.
- Blok diyagramda bulunan ‘Adder Unit’ nesnelerinin sağında bulunan matris elemanının değer özelliği 5, 6, 8 ya da 14 olmalıdır.
- Blok diyagramda bulunan, değer özelliği 8 olan yatay bağlantı elemanlarının sağında ve solunda bulunması gereken matris elemanlarının değer özelliği Çizelge 4.3’ teki değerlerden biri olmalıdır.

Çizelge 4-3 Yatay bağlantı elemanı doğrulama çizelgesi

Sol Hücre	Sağ Hücre
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	12
10	14
11	
12	
13	

- Blok diyagramda bulunan, değer özelliği 5 olan yatay & sağ-üst bağlantı elemanlarının, değer özelliği 6 olan yatay & sağ-alt bağlantı elemanlarının ve değer özelliği 7 olan yatay & sağ bağlantı elemanlarının sağında ve solunda bulunması gereken matris elemanının değer özelliği Çizelge 4.4' teki değerlerden biri olmalıdır.

Çizelge 4-4 Dikey & yatay bağlantı elemanı (5-6-7) doğrulama çizelgesi

Sol Hücre	Sağ Hücre
1	0
2	5
3	6
4	7
5	8
6	9
7	10
8	11
9	14
10	
11	
12	
13	

- Blok diyagramda bulunan, değer özelliği 10 olan sağ-üst bağlantı elemanlarının ve değer özelliği 11 olan sağ-alt bağlantı elemanlarının sağında bulunması gereken matris elemanının değer özelliği Çizelge 4.5' teki değerlerden biri olmalıdır.

Çizelge 4-5 Dikey & yatay bağlantı elemanı (10-11) doğrulama çizelgesi

Sağ Hücre
14
8
5
6
7

- Blok diyagramda bulunan 'Input Unit' ile 'Output Unit' nesneleri birbirlerine diğer nesneler ve bağlantı elemanları vasıtasıyla bağlantılı olmalıdır. Eğer arada bir kesinti var ise bu hataya neden olur.

Blok diyagramda bulunan tüm ileri yollar ve geribesleme çevrimleri program tarafından denetlenir. Herbir matris elemanın etrafındaki matris elemanlarının hangi değerlerde olabileceği bilinmektedir. Buna göre tüm blok diyagram taranarak elemanların yerleşimi dolayısıyla blok diyagramın geçerliliği test edilir. Ek-1’de bu işlemi anlatan akış şeması verilmiştir.

Akış diyagramından da anlaşıldığı üzere işlem, ‘Input Unit’ in bulunduğu matris elemanının tespit edilmesi ile başlar ve daha sonra ‘Output Unit’ nesnesi bulunana kadar devam eder. Sayısal değeri 5, 6 ve 7 olan matris elemanları ileri yol ya da geribesleme çevrimi oluşturacağı için başka bir alt program bu ileri yol ve geribesleme çevrimlerinin doğruluğunu denetler. Sonuç olarak ‘Input Unit’ nesnesi ile ‘Output Unit’ nesnesi arasında bulunan tüm matris elemanları denetlenmiş olur. Ek-2’de, ileri yol ve geribesleme çevrimlerinin doğruluğunu denetleyen alt programı anlatan akış diyagramı verilmiştir. Bu alt programın adı ‘VerifyPathLoop’dur.

Diyagramlardan da anlaşıldığı üzere ileri yol ve geribesleme çevrimlerinin sonlandığı nokta ‘Adder Unit’ nesnesidir.

4.4 Çizilen Blok Diyagramın Derlenmesi

Editörün çalışma alanına çizilmiş olan blok diyagramın doğruluğu “Verify” seçeneği ile kontrol edildikten sonra “Compile” seçeneği aktif hale gelir. Derleme işlemi menüde bulunan “Project -> Compile” seçeneğinin tıklanması ile gerçekleşir. (Kısayol: Cntrl + P)

Ek-3’te derleme işlemini gerçekleştiren ana programın akış diyagramı verilmiştir.

Akış diyagramından da anlaşıldığı üzere işlem, ‘Input Unit’ in bulunduğu matris elemanının tespit edilmesi ile başlar ve daha sonra ‘Output Unit’ nesnesi bulunana kadar devam eder. Sayısal değeri 5, 6 ve 7 olan matris elemanları ileri yol yada geribesleme çevrimi oluşturacağı için başka bir alt program bu dallanmaların ileri yol mu yoksa geribesleme çevrimi mi oluşturduğunu denetler. Bu alt programın adı ‘PathFind’dır.

- ‘PathFind’ Alt Programı

‘PathFind’ alt programı ile editörün çalışma alanına çizilmiş olan blok diyagramdaki mevcut ileri yol sayısı, geribesleme çevrim sayısı, bu ileri yol ve geribesleme çevrimlerinin başlangıç noktaları ve ‘Input Unit’ nesnesinin üstünde mi altında mı bulundukları bilgileri elde edilir. Ek-4’te bu alt programı anlatan akış şeması verilmiştir.

- ‘ForwardPathGainSub’ Alt Programı

Daha önce ‘PathFind’ alt programı ile ileri yol sayısı ve ileri yolun başladığı indeks değeri bulunmuştu. ‘ForwardPathGainSub’ alt programı ile daha önce tespit edilen bu ileri yolların kazanç değeri tespit edilir. Ek-5’te bu alt programı anlatan akış şeması verilmiştir.

- ‘FeedbackLoopGainSub’ Alt Programı

Daha önce ‘PathFind’ alt programı ile geribesleme çevrim sayısı ve geribesleme çevriminin başladığı indeks değeri bulunmuştu. ‘FeedbackLoopGainSub’ alt programı ile daha önce tespit edilen bu geribesleme çevrimlerinin kazanç değeri tespit edilir. Ek-6’da bu alt programı anlatan akış şeması verilmiştir.

Bu işlemler sonunda ileri yol sayısı, geribesleme çevrim sayısı, ileri yol kazançları ve geribesleme çevrim kazançlarını elde etmiş oluyoruz. Bundan sonra yapılacak olan işlem bu verileri “Mason Kuralı” içerisinde kullanmaktır.

4.5 Mason Kuralı

Kullanıcı tarafında BDD editörü çalışma alanına çizilen blok diyagramı mikroişlemciye yükleyebilmek için çizilen blok diyagramın mikroişlemci tarafından anlaşılacak hale dönüştürülmesi gerekir. Bu işlemin de ilk adımı blok diyagramın giriş-çıkış ilişkisini belirlemektir. Bir blok diyagramın giriş-çıkış ilişkisini belirleyebilmek için “Mason Kuralı”ndan yararlanılır. Böylece sistemin transfer fonksiyonu elde edilir. Mason kuralı (4.1) denkleminde gösterilmiştir (K. Ogata, 2002).

$$M = \frac{\sum_j M_j \Delta_j}{\Delta} \quad (4.1)$$

M = Transfer fonksiyonu

M_j= Belirli bir ileri yolun kazancı

j = Sistemdeki ileri yolları temsil eden sayı

Δ_j = 1 – j ileri yolu çıkarılınca elde edilen geribesleme çevrimleri

Δ = 1 – geribesleme çevrim kazançları toplamı + birbirine değmeyen iki çevrimin kazançlarının çarpımının toplamı - birbirine değmeyen üç çevrimin kazançlarının çarpımının toplamı +

4.6 Transfer Fonksiyonu ve Fark Denklemi

Yukarıdaki formülden de görüldüğü gibi Mason kuralını uygulayarak sistemin transfer fonksiyonunu elde etmek için ihtiyaç duyduğumuz tüm verileri zaten derleme esnasında elde etmiştik. Fakat formülde ayrıca “birbirine değmeyen çevrim” ifadesi yer almaktadır. Birbirine değmeyen çevrim kazançlarının bulunması ise şu şekilde gerçekleştirilir. Daha önce de bahsettiğimiz gibi derleme işlemi esnasında ileri yol sayısı, geribesleme çevrim sayısı, ileri yol kazançları ve geribesleme çevrim kazançlarını elde ediyorduk. Fakat buna ek olarak ayrıca bu ileri yolların ve geri besleme çevrimlerinin bir matris olarak temsil edilen çalışma alanının hangi indeks değerlerini kapsadığı da program tarafından kaydedilir. Bundan sonra programda tüm çevrimlerin kapsadığı indeks değerleri birbirleri ile karşılaştırılır. Böylece birbirine değmeyen çevrimler bulunmuş olur. Elde edilen değerler formüle konularak transfer fonksiyonu elde edilir.

Elde edilen bu transfer fonksiyonu program tarafından belirli program döngüleri sonucu elde edildiği için matematiksel olarak en sade halinde değildir. Fakat bu transfer fonksiyonunu fark denklemine dönüştürüp mikroişlemciye yüklemek için en sade haline çevirmemiz gerekir. Bu amaçla matematiksel sadeleştirme işlemleri yapılarak transfer fonksiyonunun en sade hali elde edilir. Bundan sonraki adım elde edilen bu transfer fonksiyonunun en sade halinin fark denklemine dönüştürülmesidir. Transfer fonksiyonunu fark denklemine dönüştürmemizin amacı bu transfer fonksiyonunun temsil ettiği denetleyicinin mikroişlemci tarafından gerçekleştirilebilmesidir. Çünkü mikroişlemciler dijital sistemler olduğu için ayrık zamanlı verilerle çalışırlar. Fark denklemine dönüştürülerek anlamlı hale gelen transfer fonksiyonu son adım olarak HEX dosyasına dönüştürülmelidir.

4.7 HEX Dosyasının Oluşturulması

Elde edilen fark denkleminin tek başına mikroişlemci için bir anlamı yoktur. Bu fark denkleminin temsil ettiği kontrol işleminin mikroişlemci tarafından icra edilebilmesi için öncelikle bu fark denkleminin ve fark denklemi dışında programın çalışabilmesi için gerekli olan diğer komutların birleştirilip bir C dosyası haline getirilmesi ve bu dosyanın da bir C derleyicisi aracılığı ile ilgili mikroişlemcinin anlayacağı HEX dosyasına dönüştürülmesi gerekir. Blok diyagram derleyici editörü AVR ve PIC mikroişlemcileri için HEX dosyası oluşturmayı destekler. Diğer bir deyişle içinde AVR ve PIC mikroişlemcileri için C derleyicisi bulundurulur. Program, kullanıcının tercihiye göre elde edilen fark denklemini AVR için düzenlenmiş olan şablon C programına ya da PIC için düzenlenmiş olan şablon C

programına yerleştirir. Seçili mikroişlemci tipine göre ilgili C derleyicisi çalıştırılarak C dosyasının HEX dosyasına dönüşümü sağlanır.

4.8 BAT Dosyası

Fark denklemini elde ettikten sonra daha önce de bahsettiğimiz gibi program, bu denklemi daha önceden hazırlanmış bir şablon C dosyasına yerleştirir. Bundan sonra yapılması gereken işlem bu C dosyasının derlenmesidir. Bu amaçla program içerisinde bulunan C derleyicilerine bu C dosyasını gösterip derleme işlemini gerçekleştirmelerini sağlamak gerekir. Bu işlem BAT dosyaları kullanılarak gerçekleştirilir. BAT dosyası DOS, OS/2 ve Microsoft Windows'ta birtakım seri komutlar içeren text dosyasıdır. Dosya çalıştırıldığında içerisinde bulunan komutlar satır satır icra edilir. Derleyicilere de komutlar yardımıyla derleyecekleri dosya gösterilebilir ve derleme işlemi başlatılabilir. Sistemde bilindiği gibi iki ayrı derleyici bulunmaktadır. Dolayısıyla bu iki ayrı derleyiciyi çalıştırmak için iki ayrı BAT dosyası oluşturulmuştur. Bu bat dosyalarının içerikleri aşağıdaki gibidir.

- AVR Derleyicisi için;

```
D:\PROJE\Editor\Editor_Code\CAVR\BIN\CAVR.exe
```

```
D:\PROJE\Editor\Editor_Code\CAVR\Output\AVR\AVR.c /sm /i0x200 /o0x60 /s
```

- PIC Derleyicisi için;

```
"C:\Projects\MPASM Suite\MPASMWIN.exe" /q /p18F8722
```

```
"C:\Projects\MPASM Suite\Template\Object\18F8722TMPO.ASM" /I"18F8722TMPO.lst"  
/e"18F8722TMPO.err" /o"18F8722TMPO.o" /d__DEBUG=1
```

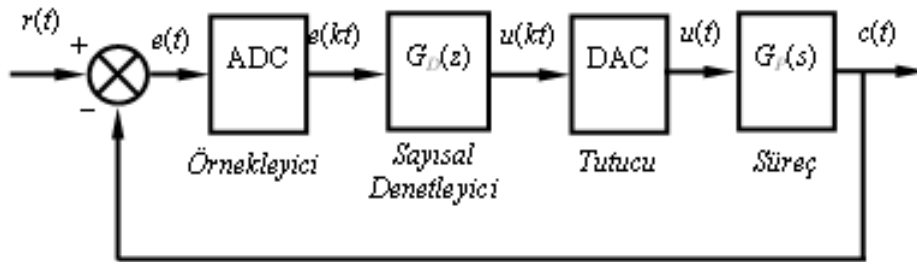
```
"C:\Projects\MPASM Suite\mplink.exe" "C:\Projects\MPASM Suite\LKR\18f8722.lkr"  
"18F8722TMPO.o" /u__DEBUG /z__MPLAB_BUILD=1 /z__MPLAB_DEBUG=1  
/o"MyProjects.cof" /M"MyProjects.map" /W
```

Yukarıda da görüldüğü gibi her derleyiciyi çalıştırmak için gerekli olan kodlar farklıdır. Derleme işlemi sonunda HEX dosyası elde edilir. Bu HEX dosyasının da mikroişlemciye yüklenmesi ile tasarlanan sistem gerçekleşmiş olacaktır.

5. PID DENETLEYİCİ

Sürekli zamanlı sistemlerde olduğu gibi ayrık zamanlı sistemlerde de hata işaretininin seçilen katsayılarla oran, integral ve türev işlemlerinin toplamı olarak elde edilen bir işaretle süreci denetleyen “üç terimli denetleyici” veya “PID denetleyici” uygulamada en sık karşılaşılan kontrol yöntemidir. Sürekli zamanlı uygulamalarda işlemsel kuvvetlendirici (opamp), direnç ve kondansatör gibi analog elemanlarla gerçekleştirilen yöntem, sayısal uygulamalarda mikroişlemci veya mikrodeneleyici gibi bilgisayar temelli denetleyicilerle yazılım olarak gerçekleştirilir. Sayısal PID denetleyicinin tasarımında, çoğunlukla analog PID denetleyici tasarımı için geliştirilmiş köklerin yer eğrisi ve, Bode ve Nyquist çizimleri gibi frekans cevabı temelli yöntemlerden yararlanır. Bu klasik yöntemlerden bir diğeri de, pratik olduğu için 50 yılı aşkın bir süredir kullanılan PID katsayılarının sürecin açık çevrim “tepki eğrisi” ne göre belirlendiği Ziegler-Nichols yöntemleridir.

Klasik tasarım yöntemleri dolaylı veya doğrudan olmak üzere iki biçimde uygulanır. Dolaylı uygulamada, süreç dinamiklerine uygun bir analog PID denetleyici tasarlanır ve daha sonra bu denetleyici ayrıklaştırılarak fark denklemi halinde sayısal işlemciye yazılım olarak aktarılır. Doğrudan uygulamada ise sürecin ayrık modeli elde edilerek yine analog denetleyici tasarımlarında kullanılan yöntemlerle ayrık PID denetleyici tasarımı gerçekleştirilir. Ele alınan örnek uygulama için de geçerli genel bir sayısal kontrol sistemi blok çizimi Şekil 5.1’de gösterilmiştir.



Şekil 5-1 Sayısal kontrol sistemi

Belirtildiği gibi, PID denetleyici, endüstriyel kontrol sistemlerinde çokça kullanılan geribeslemeli bir kontrol sistemidir ve temel olarak üç ayrı parametre içerir. Bunlar, oran kazancı (K_p), integral kazancı (K_i) ve türev kazancıdır (K_d). Bu katsayıların denetlenecek sürece ve maruz kalınan iç ve dış bozuculara göre seçilmesi ve ayarlanması literatürde önemli bir yer teşkil eder. Esas olarak, oran kazancı cevabın yükselme süresini kısaltır, fakat fazla

büyük olması aşımara sebep olabilir; integral kazancı kalıcı-hal hatasını, e_{ss} , azaltmaya yararken iyi ayarlanmaması yerleşme süresini uzatabilir; türev kazancı aşımara yol açmadan cevabı hızlandırmaya yarar. Bu üçlünün birbirini karşılıklı etkilemesi de söz konusu olduğundan PID katsayılarının seçilmesi ve ayarlanması önemli bir çalışma konusudur.

5.1 Sayısal PID Denetleyici

Sürekli zamanda PID denetim işlemi oran, K , integral, K_i , ve türev, K_d , katsayıları cinsinden zaman düzleminde (5.1) denklemindeki gibi ifade edilir (N.S. Nise,2004; Carnegie Mellon University).

$$u(t) = Ke(t) + K_i \int_{t_0}^t e(t)dt + K_d \frac{de(t)}{dt} \quad (5.1)$$

Ancak, daha anlamlı bulunduğundan yukarıdaki ifade pratikte saniye cinsinden integral (reset) zamanı, T_i , ve türev (hız) zamanı, T_d , değişkenleriyle (5.2) denklemindeki gibi formüle edilir (K. Ogata, 2002).

$$u(t) = K \left[e(t) + \frac{1}{T_i} \int_{t_0}^t e(t)dt + T_d \frac{de(t)}{dt} \right] \quad (5.2)$$

Burada, kolayca görüleceği üzere integral ve türev zamanları,

$$T_i = K/K_i, \quad T_d = K_d/K \quad (5.3)$$

olarak tanımlanmıştır. PID ifadesi özellikle köklerin yer eğrisi ve frekans cevabı temelli tasarım yöntemleri kullanılırken sıfır başlangıç şartları için s tanım bölgesinde transfer fonksiyonu olarak (5.4) denklemindeki gibi ifade edilir.

$$\frac{U(s)}{E(s)} = K + K_i \frac{1}{s} + K_d s = K \left(1 + \frac{1}{T_i s} + T_d s \right) \quad (5.4)$$

Görüldüğü gibi transfer fonksiyonunun iki sonlu sıfırı ve orijinde tek kutbu vardır. Yamuklarla (trapezoidal) ayrık integral (bir başka deyişle Tustin bilineer dönüşümü) ve basit ayrık türev alma işlemleri kullanılarak yukarıdaki sürekli zamanlı PID ifadelerinden sayısal PID denetleyicinin z tanım bölgesinde darbe transfer fonksiyonu (5.5) denklemindeki gibi elde edilebilir.

$$\frac{U(z)}{E(z)} = K \left(1 + \frac{1}{T_i} \frac{T(1+z^{-1})}{2(1-z^{-1})} + T_d \frac{1-z^{-1}}{T} \right) \quad (5.5)$$

Burada, T örnekleme aralığı, T_i ve T_d integral ve türev zamanlarıdır. Zaman birimi saniyedir.

Denklem (5.5) basit bir işlemle denklem (5.6) haline getirilebilir,

$$\frac{U(z)}{E(z)} = K \left(1 - \frac{T}{2T_i} + \frac{T}{T_i} \frac{1}{1-z^{-1}} + \frac{T_d}{T} (1-z^{-1}) \right) \quad (5.6)$$

Denklem (5.6)'da sayısal PID katsayıları oran, integral ve türev kazançları, sırasıyla K_p , K_I , K_D , tanımlamaları yapıldığında,

$$K_p = K - \frac{KT}{2T_i}, \quad K_I = \frac{KT}{T_i}, \quad K_D = \frac{KT_d}{T} \quad (5.7)$$

standart PID darbe transfer fonksiyonu denklem (5.8) deki gibi elde edilir (K. Ogata, 1995),

$$\frac{U(z)}{E(z)} = K_p + K_I \frac{1}{1-z^{-1}} + K_D (1-z^{-1}) \quad (5.8)$$

Denklem (5.7)'de verilen K_p ve K_I tanımlarına dikkat edildiğinde,

$$K_p = K - \frac{KT}{2T_i}, \quad K_I = \frac{KT}{T_i} \Rightarrow K_p = K - \frac{K_I}{2} \quad (5.9)$$

sayısal PID denetleyici için bulunan oran katsayısının, K_p , analog PID oran katsayısından, K , $K_I/2$ kadar daha küçük olduğu görülür, Denklem (5.9), (K. Ogata, 1995). Açıkça ki bir süreç için tasarlanan analog PID denetleyici denklem (5.7) ve (5.8) kullanılarak sayısal PID denetleyici haline getirilebilir. Bu yaklaşıma yukarıda tanımlandığı gibi dolaylı sayısal denetleyici tasarımı denir. Sayısal PID denetleyici, z tanım bölgesinde, diğer bir deyişle ayrık düzlemde, darbe transfer fonksiyonu olarak ifade edildiği gibi, denklem (5.8), uygulama (implementasyon) uygunluğu bakımından fark denklemi olarak da ifade edilir, denklem (5.10).

$$u(kT) = u((k-1)T) + (K_p + K_I + K_D)e(kT) - (K_p + 2K_D)e((k-1)T) + K_De(k-2)T \quad (5.10)$$

Çıkış işaretinin her zaman belirli bir eşik seviyesinin üzerinde olması isteniyorsa denklem (5.10)'un sağ tarafına u_0 gibi sabit bir değer eklenir, denklem (5.11).

$$u(kT) = u((k-1)T) + (K_p + K_I + K_D)e(kT) - (K_p + 2K_D)e((k-1)T) + K_De(k-2)T + u_0 \quad (5.11)$$

Denklem (5.11)'de verilen fark denklemi her tarama peryodunda, T , u_0 seviyesine göre çıkış ürettiğinden mutlak veya konum biçimli PID denklemi olarak adlandırılır. Uygulamada daha ziyade çıkışın o anki değerinin bir önceki değerine olan farkı olan bir işaret, $y(kT) = u(kT) - u((k-1)T)$, süreci denetlemede kullanılır. Buna artımsal veya hız biçimli PID algoritması denir, denklem (5.12).

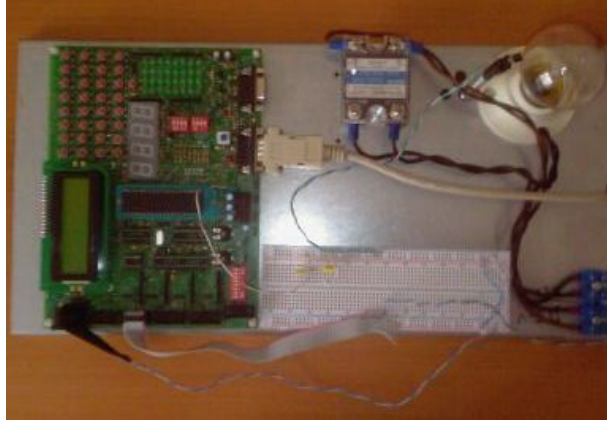
$$y(kT) = (K_p + K_i + K_d)e(kT) - (K_p + 2K_d)e((k-1)T) + K_de(k-2)T \quad (5.12)$$

Burada her taramada (her T örnekleme aralığında) denetleyici çıkışı, bir önceki çıkış değeri referans alınarak hesaplanır; bir başka deyişle üretilen kontrol işareti aslında çıkış işaretinin değişim değeridir.

5.2 Deney Düzeneği

Deneylerde, bir ampülün sıcaklığını denetlemek amacıyla bir SSR röle (Solid State Relay) üzerinden bir mikroişlemci aracılığıyla PID kontrolü uygulanmıştır. Denetleyicinin giriş işareti ampülün mevcut sıcaklığı ile set değeri (arzu edilen sıcaklık) arasındaki fark, çıkışı ise bu hata sinyalinin PID algoritmasında işlenmesi sonucu elde edilen ve SSR'ye uygulanacak olan PWM doluluk/boşluk oranı (duty cycle) değeridir. Her örnekleme aralığında hata işareti üç kez okunarak ortalaması alınır ve bu ortalama değer PID algoritmasında işlenir. Gerçekleştirilen sistem Şekil 5.2'de gösterilmiştir.

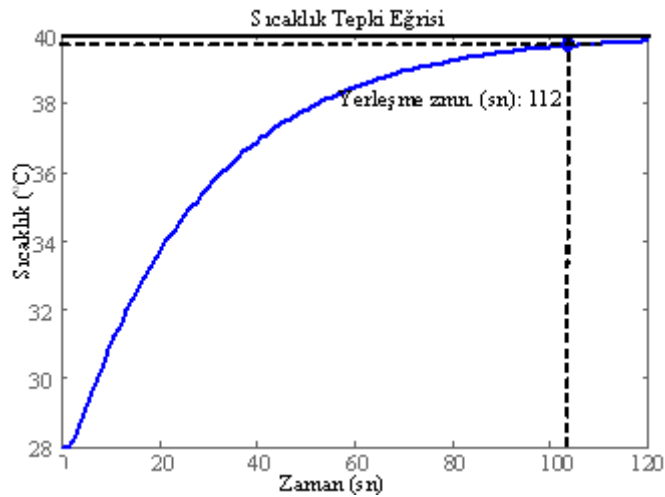
Deney düzeneğinde, içerisinde 10 bit ADC bulunan Microchip firmasının ürettiği 16F877 kodlu mikroişlemci kullanılmıştır. SSR röle ise Norton Elektronik firmasının ürettiği tetikleme gerilimi 3 - 32VDC, çıkış gerilimi 90 - 480VAC ve çıkış akımı 10A olan ZG3NC-210B kodlu üründür. Kullanılan 100 Watt'lık lambanın sıcaklığı LM35 sıcaklık sensörü ile ölçülür. Bu sensörün $+10.0\text{mV}/^\circ\text{C}$ değerinde lineer bir çalışma eğrisi vardır.



Şekil 5-2 Ampül sıcaklık kontrol devresi

5.3 PID Denetleyici Tasarımı ve Gerçeklemesi

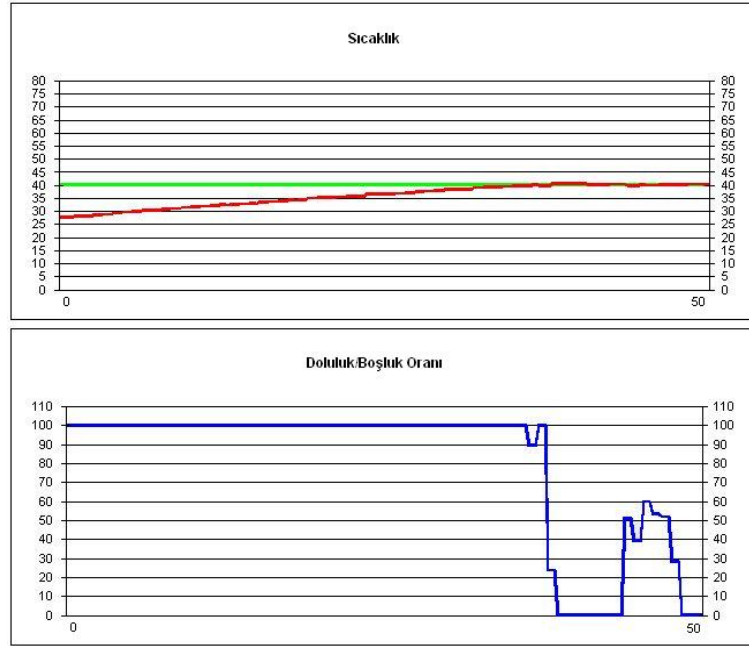
Geliştirilen BDD ile yapılan sıcaklık kontrol deneylerinde birinci mertebeden, ölü zamanlı bir sürecin denetlenmesi söz konusudur. Sisteme çeşitli değerlerde enerji verilerek çıkış sıcaklık değişiminin kaydedilmesiyle sürecin matematiksel modeli kurulmuş ve buna göre denetleyici tasarım ve uygulamaları gerçekleştirilmiştir. Şekil 5.3'te ampül sıcaklığının oransal kontrolle açık-çevrimli olarak oda sıcaklığından (28°C), 40°C 'ye yükselme eğrisi verilmiştir. Eğriden sürecin zaman sabitinin $T_C=28$ ve gecikme zamanının $T_L=0.5$ saniye olduğu belirlenmiştir.



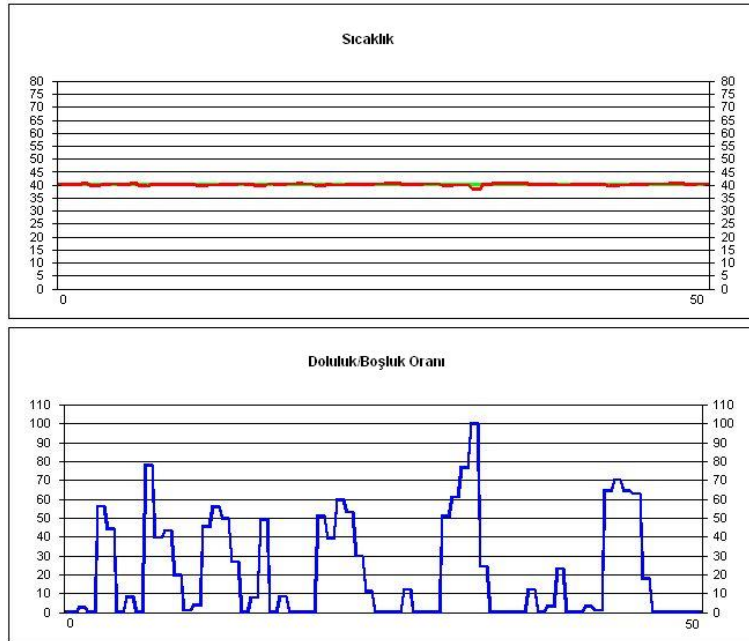
Şekil 5-3 Sürecin tepki eğrisi ($28 - 40^{\circ}\text{C}$)

Yukarıda bahsedildiği gibi, PID denetleyici tasarımı için çeşitli yöntemler söz konusudur, bu klasik yöntemlerden biri olan Ziegler-Nichols tasarım yöntemi burada kullanılmıştır. Yöntem,

aralığı 0-50 saniyedir. Yatay eksenler zaman (sn), dikey eksenler ise sırasıyla sıcaklık ($^{\circ}\text{C}$) ve doluluk/boşluk oranıdır (%).



Şekil 5-5 Sıcaklık denetim çıkışı (Geçici hal cevabı). Yatay eksenler zaman (sn), dikey eksenler ise üstteki grafikte sıcaklık ($^{\circ}\text{C}$), alttakinde ise doluluk/boşluk oranıdır (%).



Şekil 5-6 Sıcaklık denetim çıkışı (Kalıcı hal cevabı). Yatay eksenler zaman (sn), dikey eksenler ise üstteki grafikte sıcaklık ($^{\circ}\text{C}$), alttakinde ise doluluk/boşluk oranıdır (%).

Görüldüğü gibi PID denetleyicinin, süreci tam olarak denetim altına alarak, herhangi bir aşım yapmadan yerleşme zamanını yaklaşık 112 saniyeden 28 saniyeye indirmiştir.

6. SONUÇLAR

Bu çalışmada tasarlanan bir blok diyagram derleyici vasıtasıyla bir sayısal kontrol sistemi için süreç dinamiklerine uygun denetleyici algoritmalarının geliştirilmesi ve bu algoritmanın katsayı değerlerinin de yine bu Blok Diyagram Derleyicisi vasıtasıyla hızlı bir şekilde değiştirilip optimum bir kontrol işleminin yerine getirilmesi sağlanmıştır. BDD editörü sayesinde değişik katsayı değerlerine karşılık gelen fark denkleminin elde edilmesine ve bu elde edilen fark denkleminin yerine getireceği kontrol işleminin mikroişlemci tarafından icra edilebilmesi için gerekli olan işlemci programının yazılması gerekliliği ortadan kalkmıştır. Bu da tasarım süresini oldukça kısaltacaktır. BDD editöründe oluşturulan blok diyagramı, program tarafından Mason kuralı kullanılarak önce denetleyiciyi temsil eden darbe transfer fonksiyonuna, $G_D(z) = U(z)/E(z)$, dönüştürülür. Program daha sonra bu darbe transfer fonksiyonuna matematiksel sadeleştirme işlemleri uygular. Transfer fonksiyonunun en sade hali elde edildikten sonra bu fonksiyonu fark denklemine dönüştürür. Üretilen fark denklemini daha önceden oluşturulmuş bir taslak C programına yerleştirir. Bu C dosyası ilgili derleyici vasıtasıyla otomatik olarak derlenir. İşlem sonucunda elde edilen HEX dosyası editör içinde gömülü olan IC-Prog programı ile PIC mikroişlemcisine yüklenir. Ayrıca BDD editöründe oluşturulan blok diyagram editör tarafından kaydedilebileceği için daha sonra tekrar kullanılabilir. Yapılan bu çalışmaya ek olarak, BDD editörü çalışma alanı genişletilerek daha büyük blok diyagramların da derlenmesi sağlanabilir. Ayrıca editörün diğer MCU' ları da desteklemesi için istenilen MCU' nun derleyicisi yazılım içerisine yerleştirilebilir.

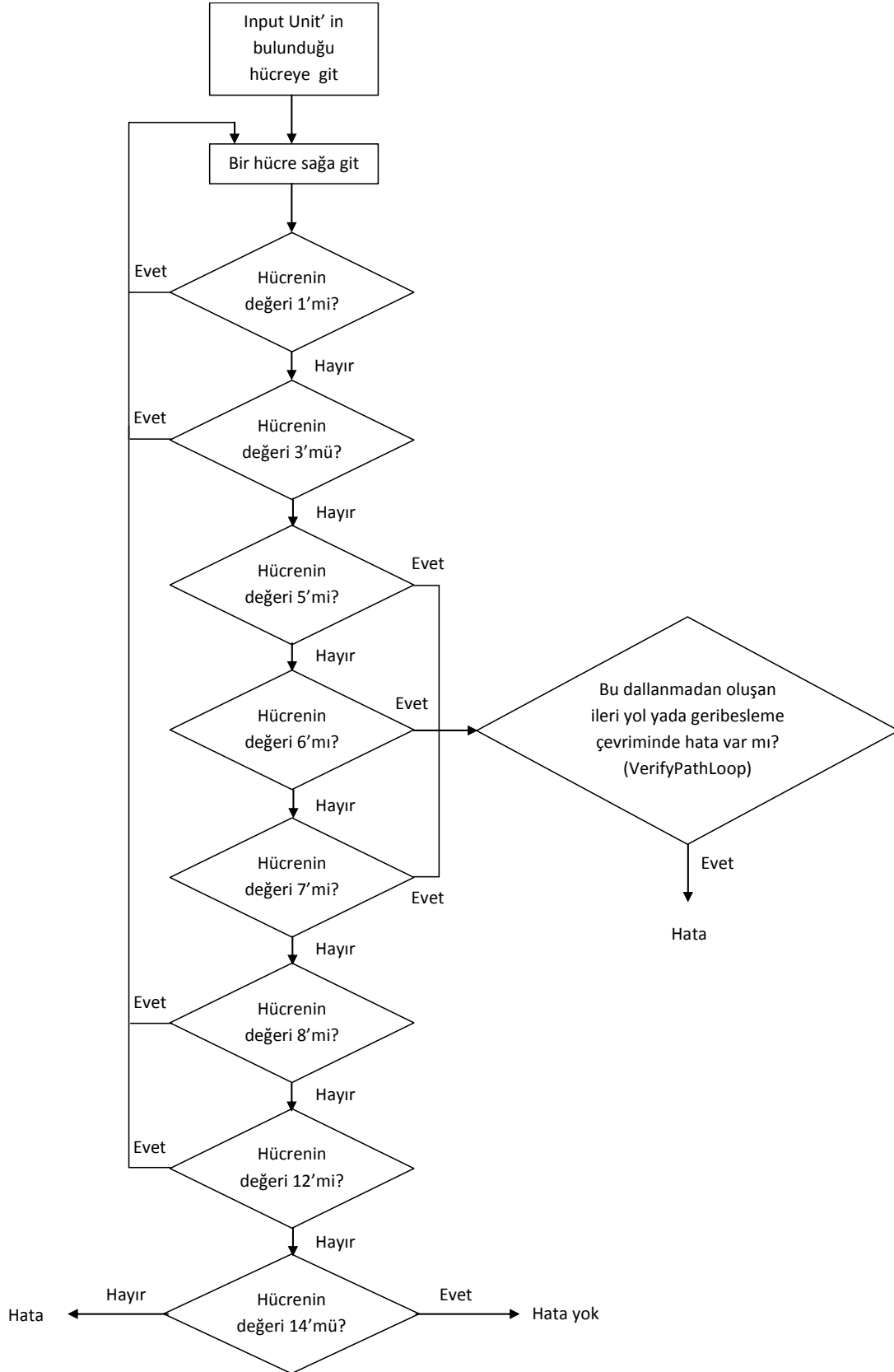
KAYNAKLAR

- A.V. Oppenheim, R.W. Schafer, J.R. Buck, “Discrete-Time Signal Processing”, Prentice Hall, 2. baskı, 1999.
- B.C. Kuo, “Digital Control Systems”, Oxford Univ. Press, 2. baskı, 1997
- B. Porat, “A Course in Digital Signal Processing”, J.Wiley, 1997.
- Carnegie Mellon University, The University of Michigan, “Control Tutorials for Matlab, PID Tutorial”
- Francesco Balena, “Programming Visual Basic 6.0”, Microsoft Press, 1999.
- K. Dutton, S. Thompson, B. Barraclough, “The Art of Control Engineering”, Addison-Wesley, 1998
- K.J. Åström, T. Hagglund, “PID Controllers: Theory, Design, and Tuning”, ISA Publ., 2. baskı, 1995
- K. Ogata, “Discrete-Time Control Systems”, Prentice Hall, 2. baskı, 1995.
- K. Ogata, “Modern Control Engineering”, Prentice Hall, 4. baskı, 2002.
- N.S. Nise, “Control Systems Engineering”, J. Wiley, 4. baskı, 2004.
- P.N. Paraskevopoulos, “Digital Control Systems”, Prentice Hall, 1996.

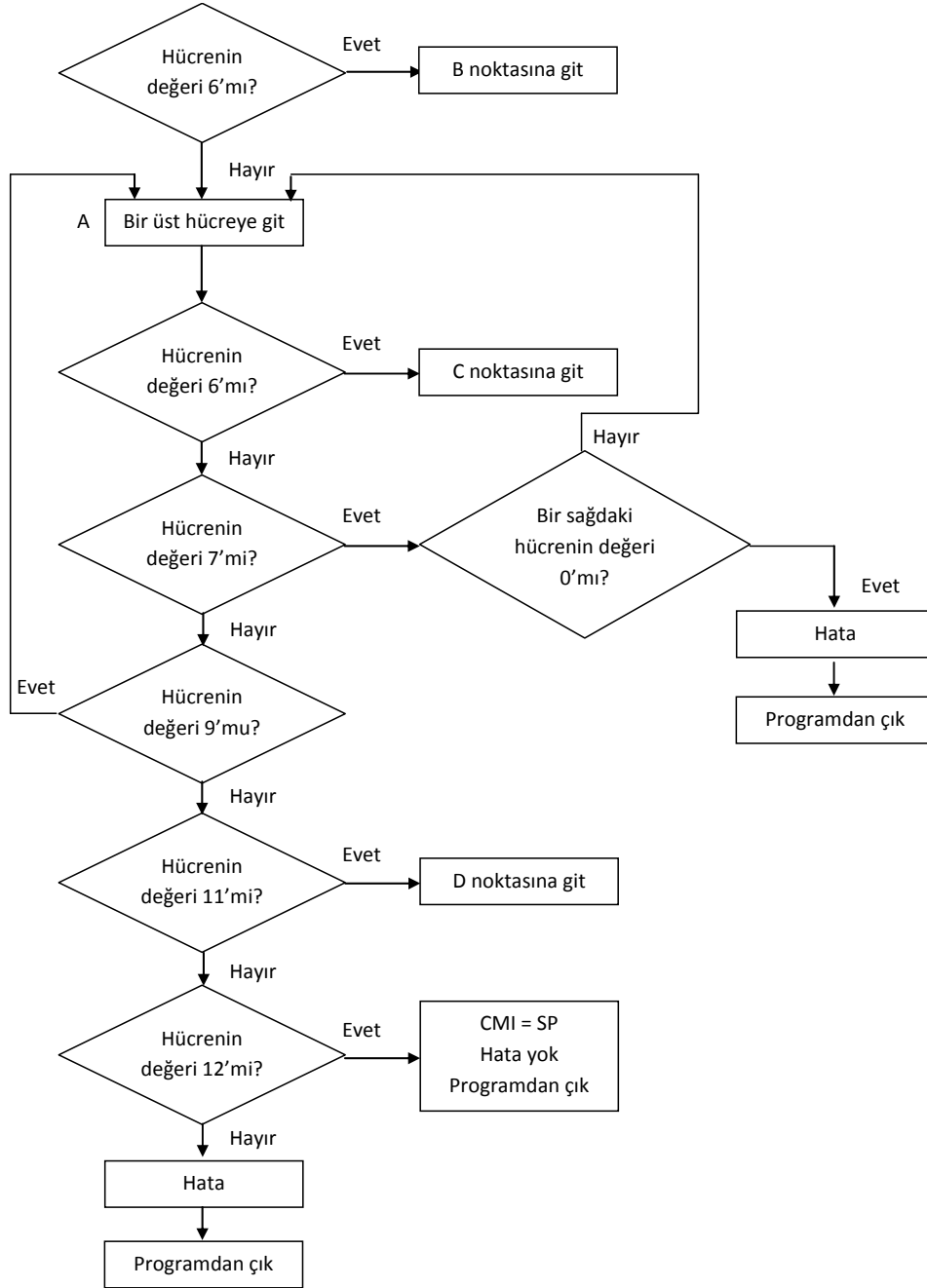
EKLER

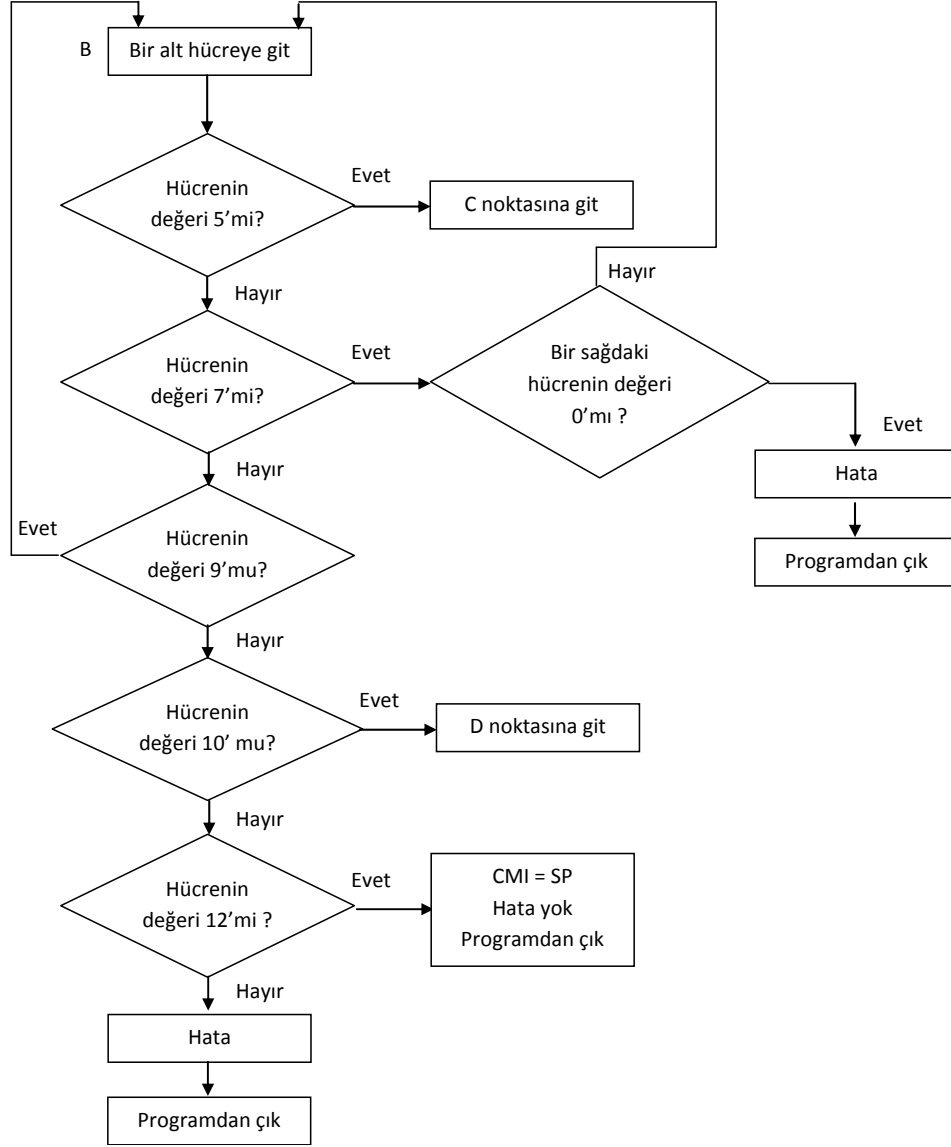
Ek 1	Blok Diyagram Doğrulama Ana Programı
Ek 2	‘VerifyPathLoop’ Alt Programı
Ek 3	Blok Diyagram Derleme Ana Programı
Ek 4	‘PathFind’ Alt Programı
Ek 5	‘ForwardPathGainSub’ Alt Programı
Ek 6	‘FeedbackLoopGainSub’ Alt Programı

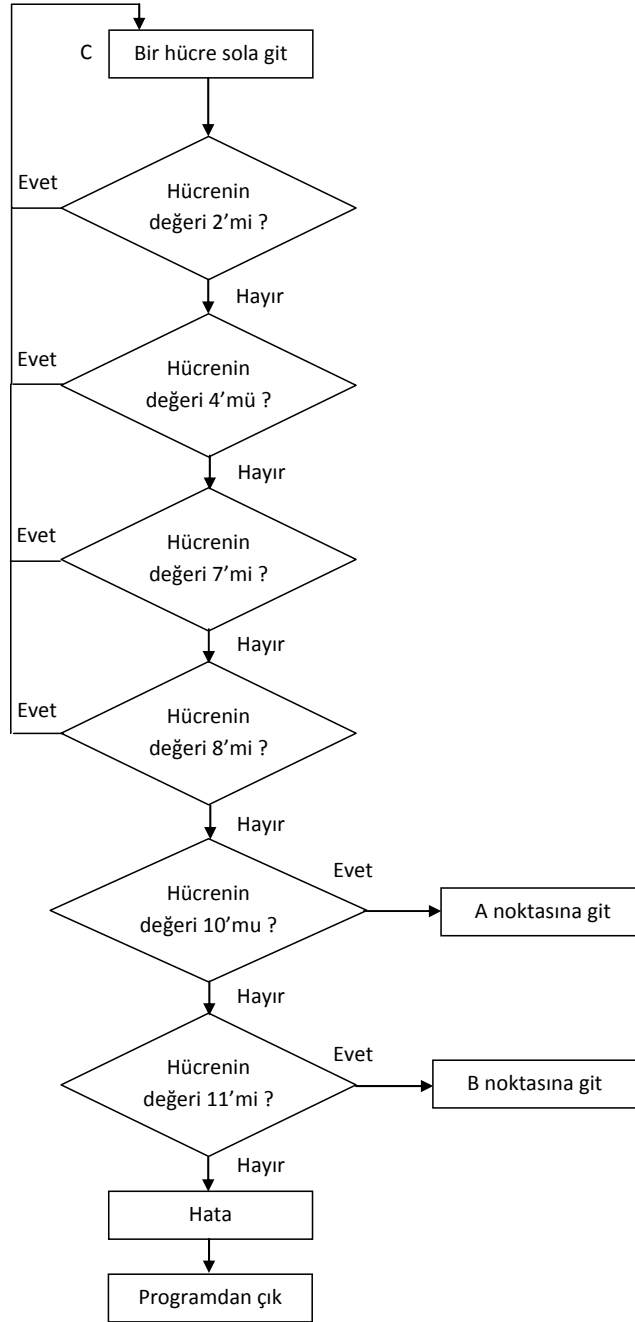
Ek 1 Blok Diyagram Doğrulama Ana Programı

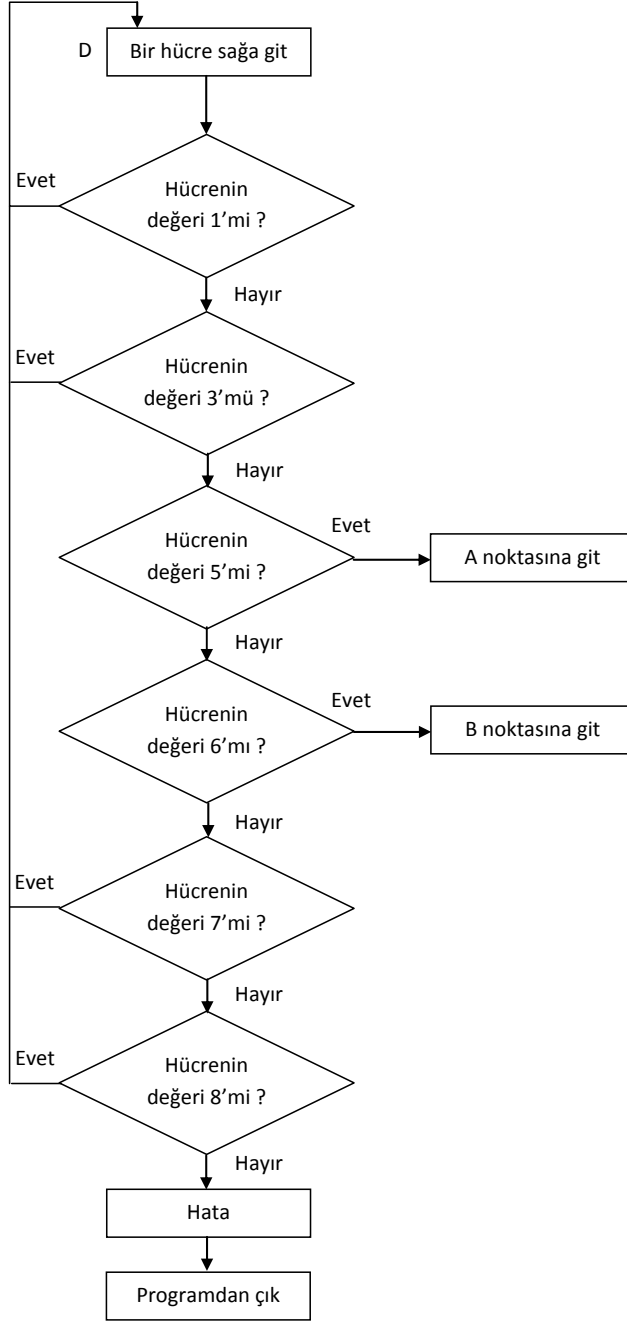


Ek 2 'VerifyPathLoop' Alt Programı





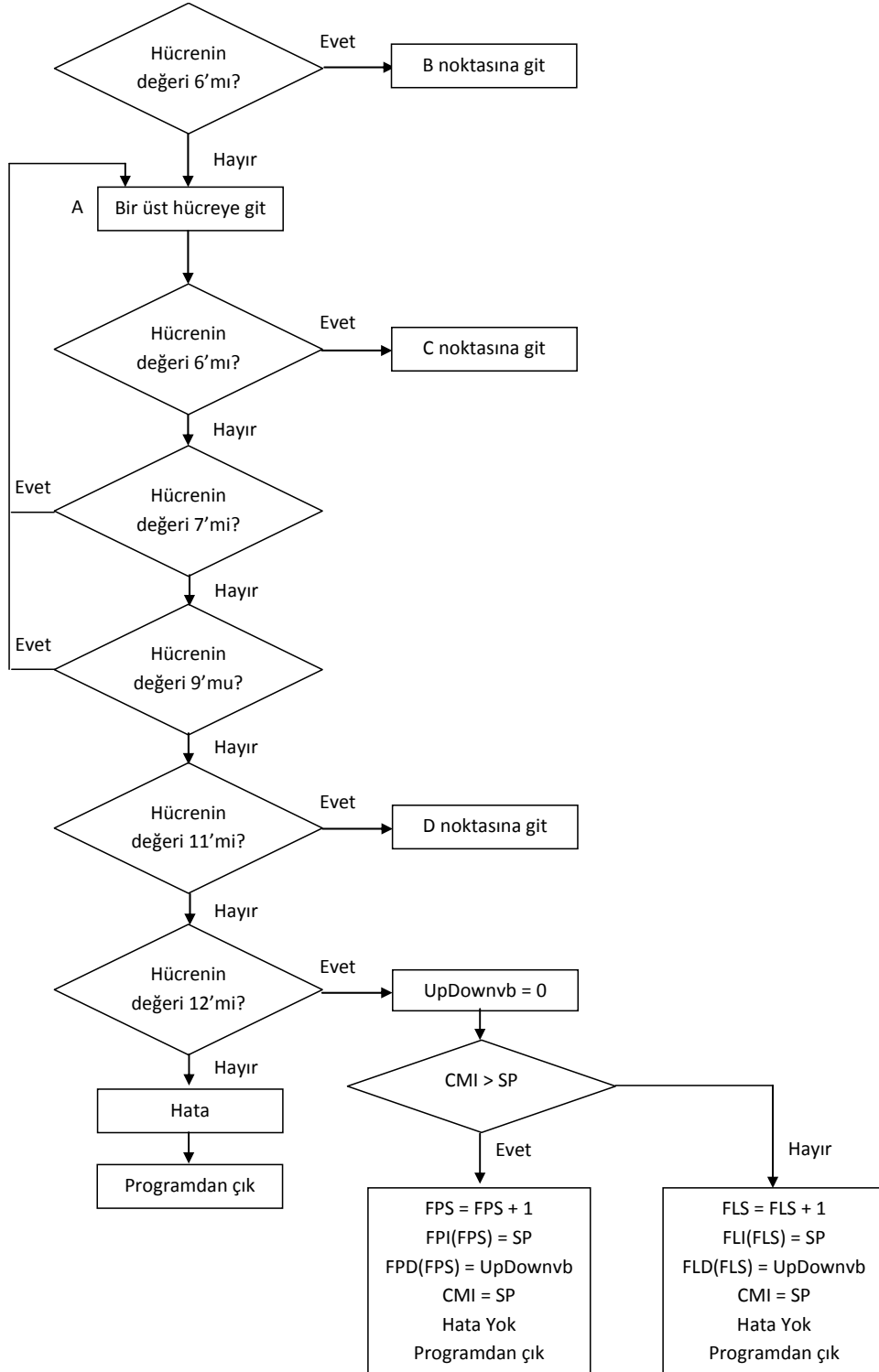


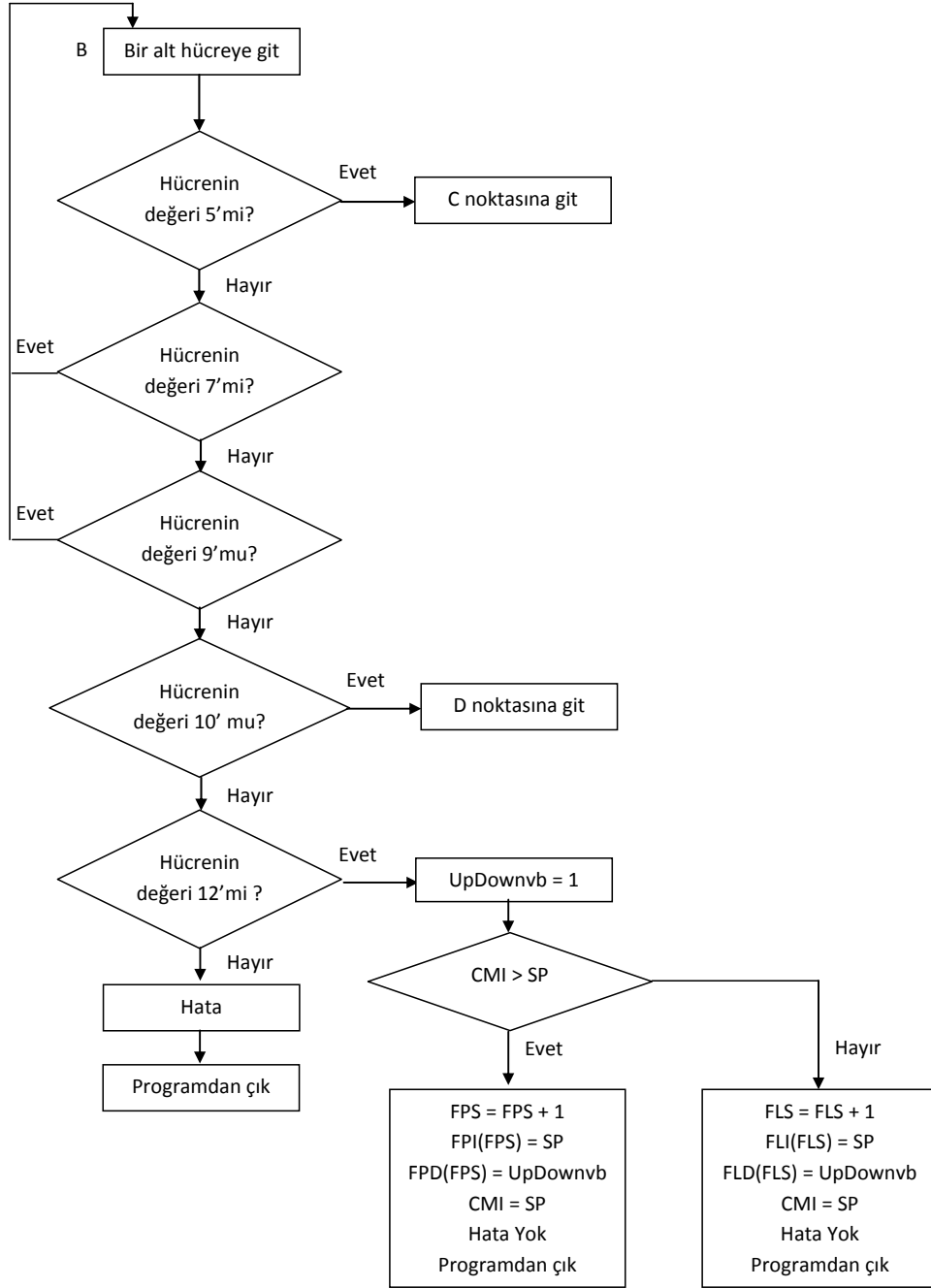


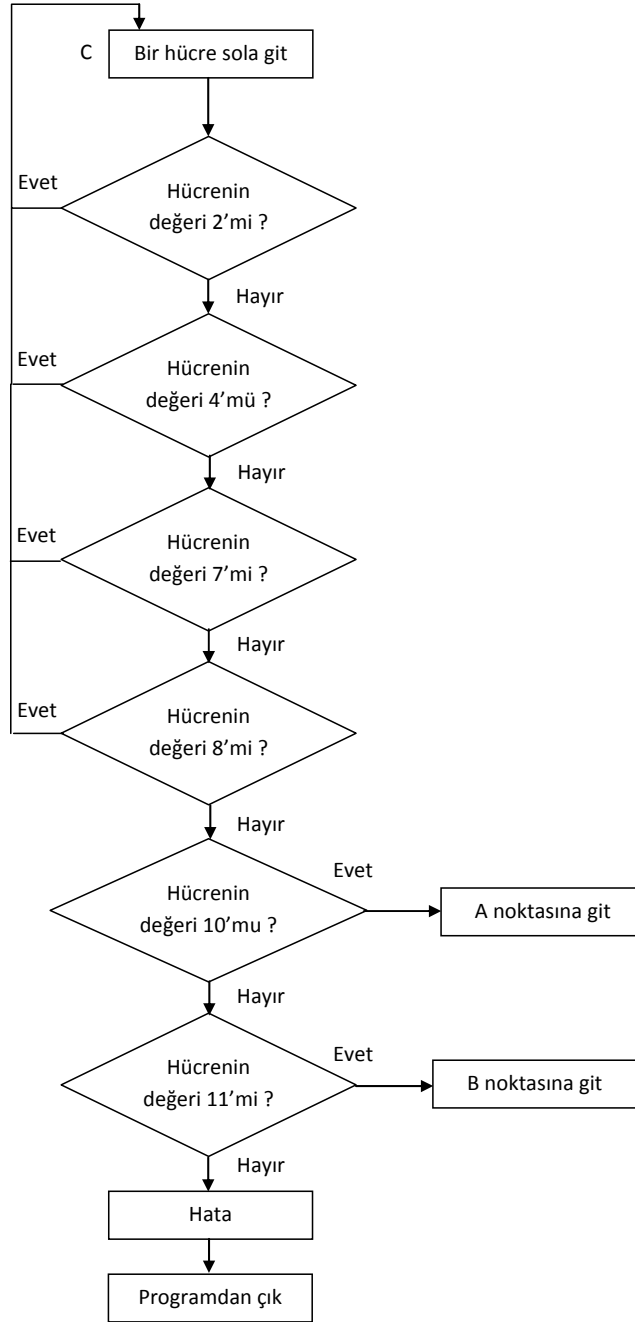
Ek 3 Blok Diyagram Derleme Ana Programı

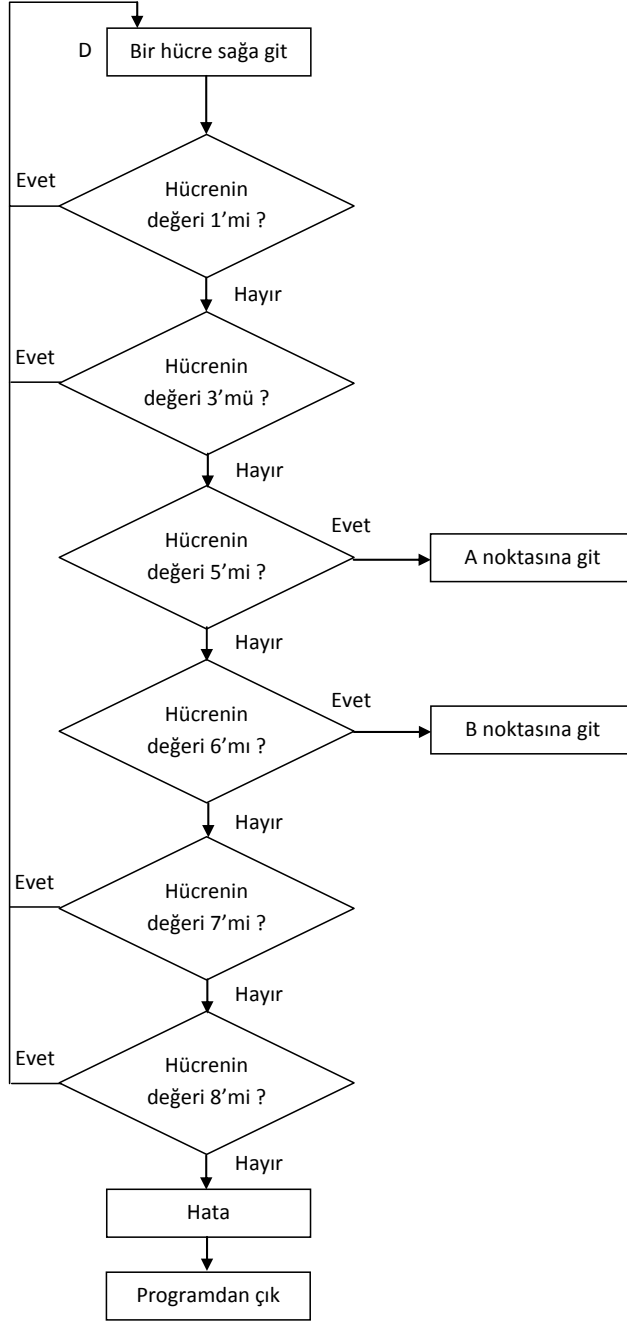


Ek 4 'PathFind' Alt Programı

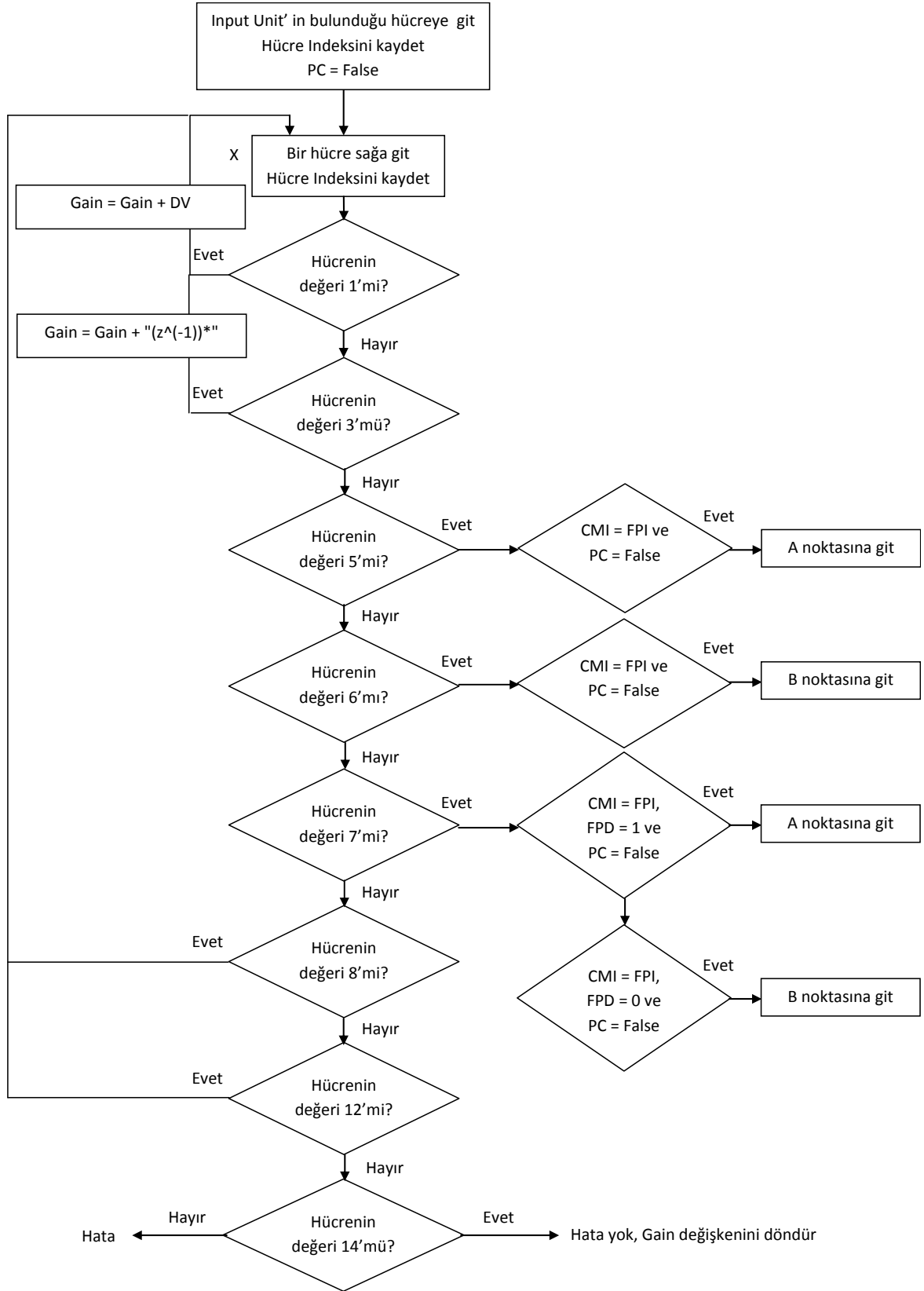


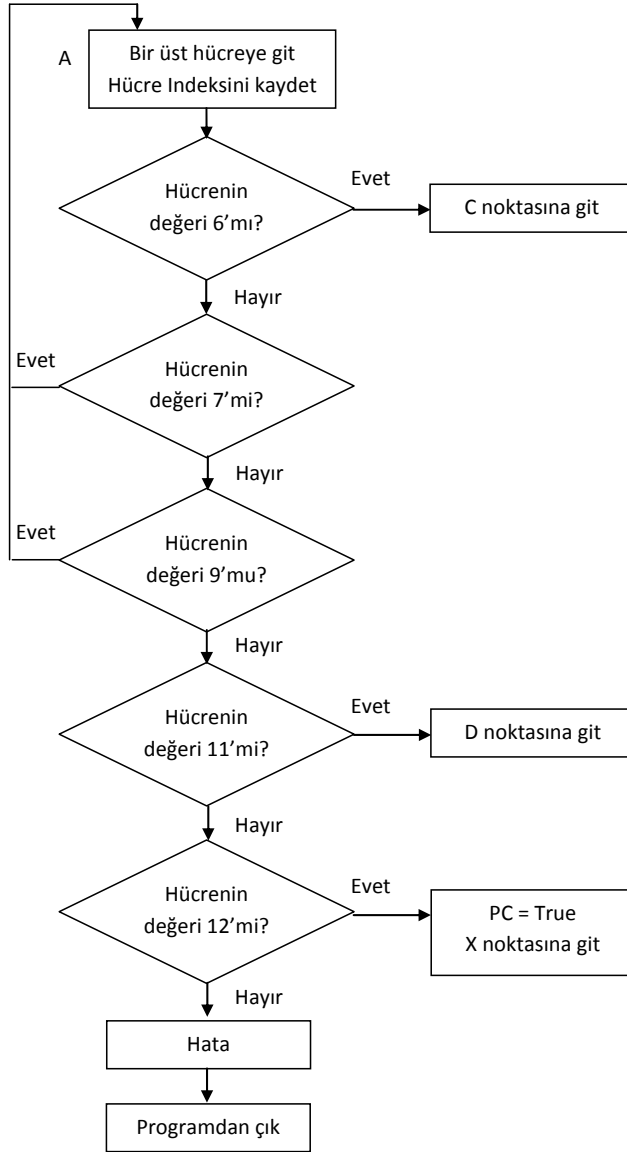


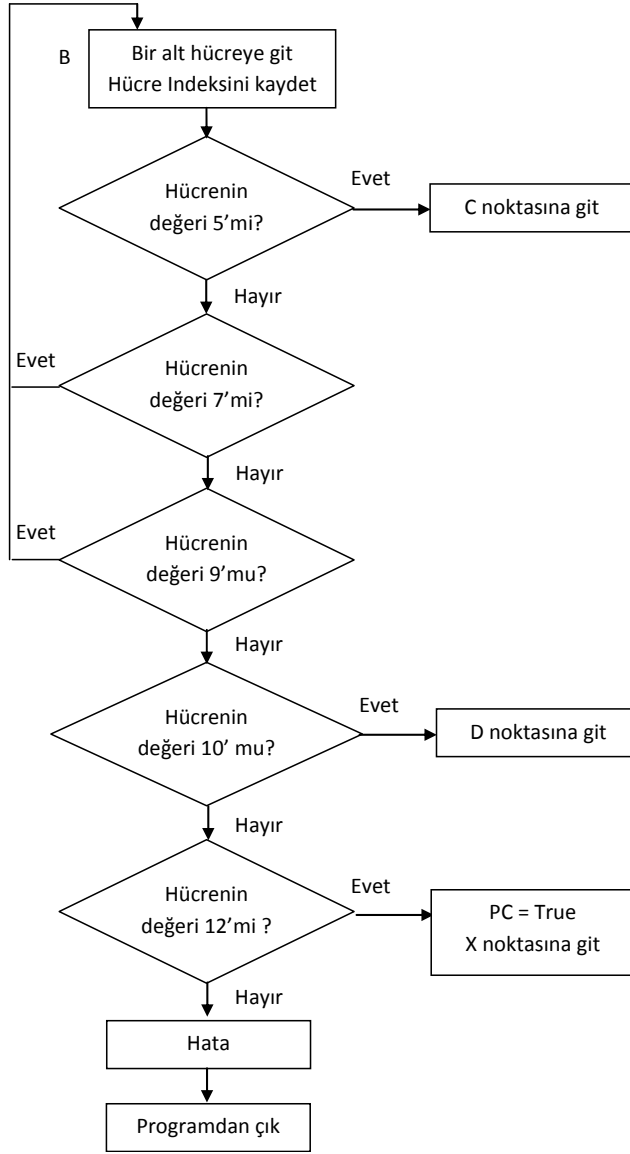


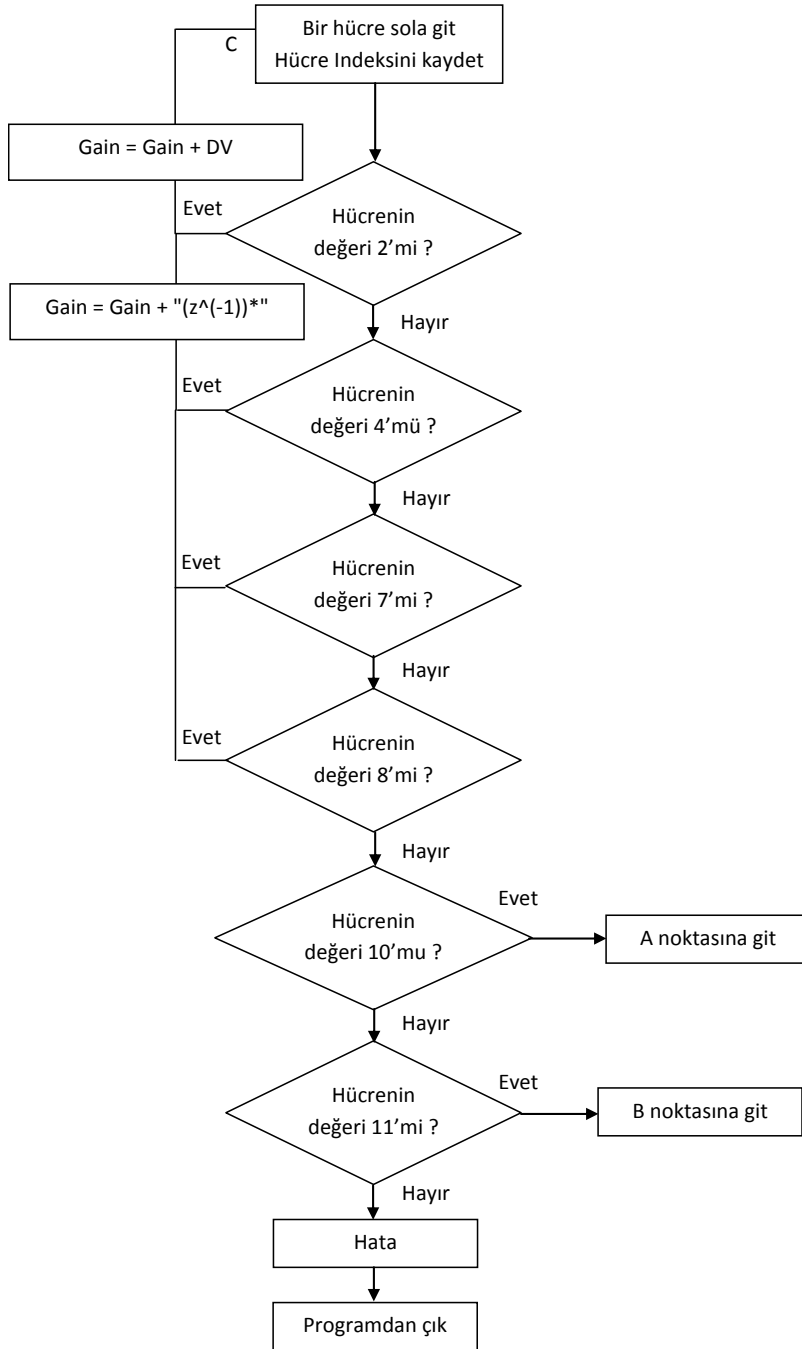


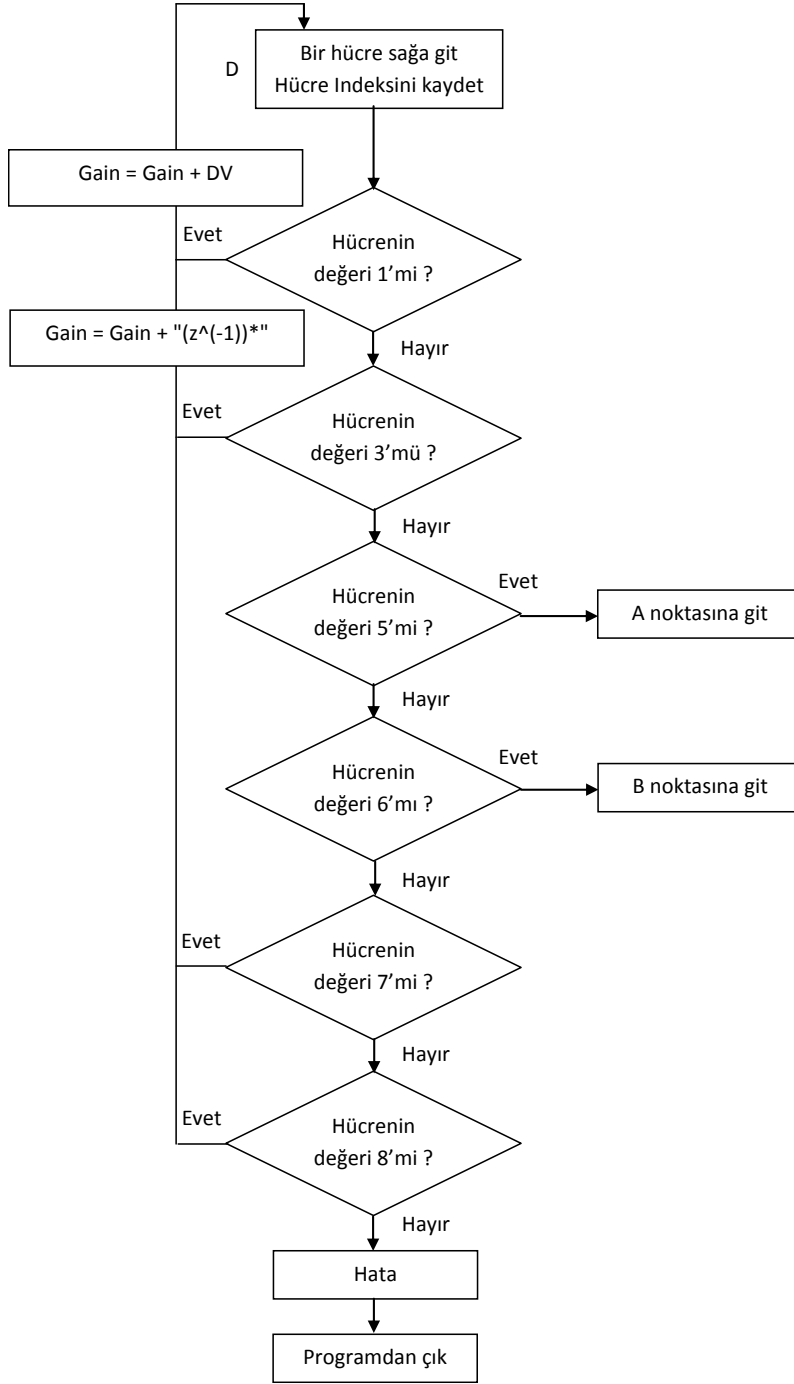
Ek 5 'ForwardPathGainSub' Alt Programı



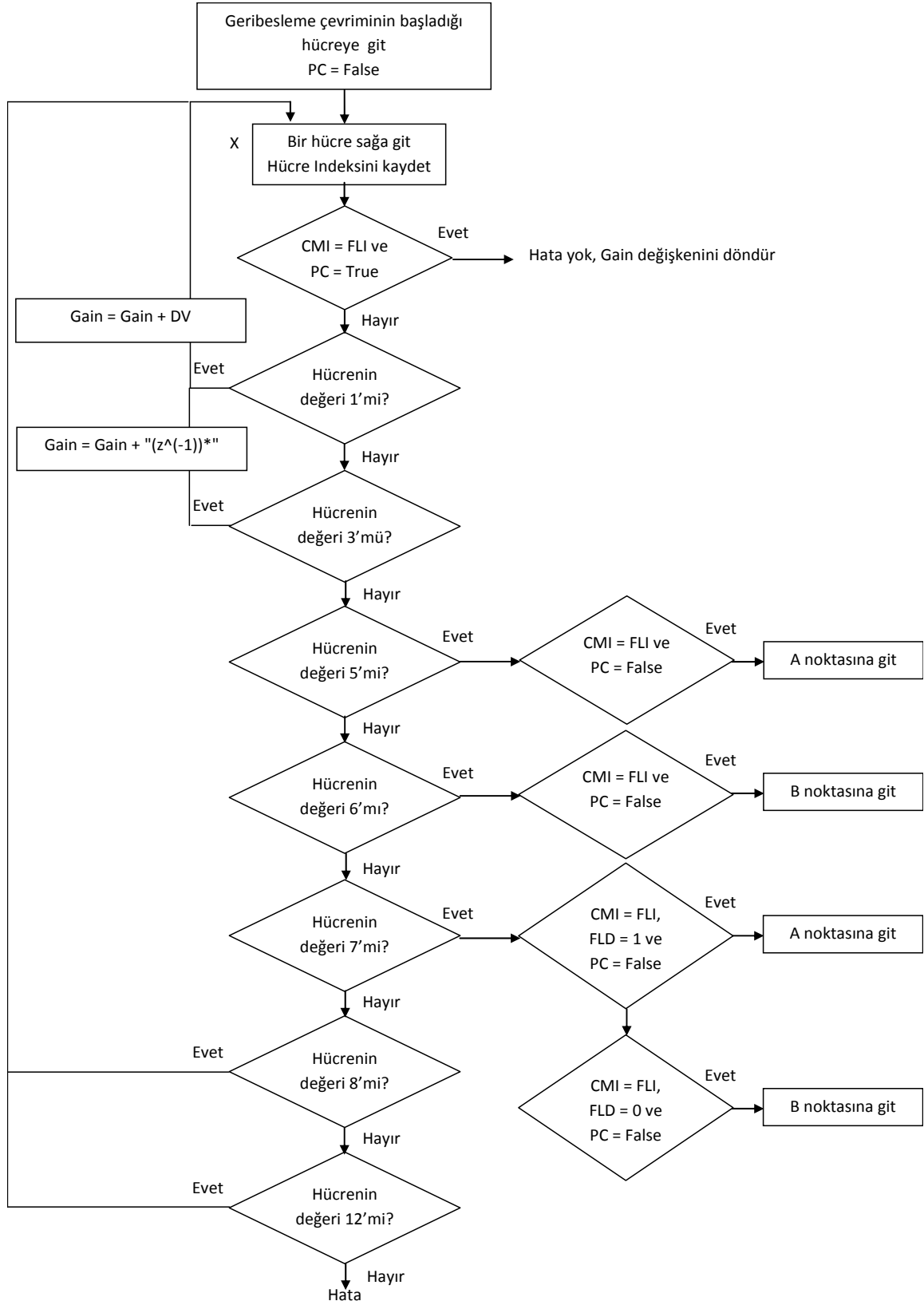


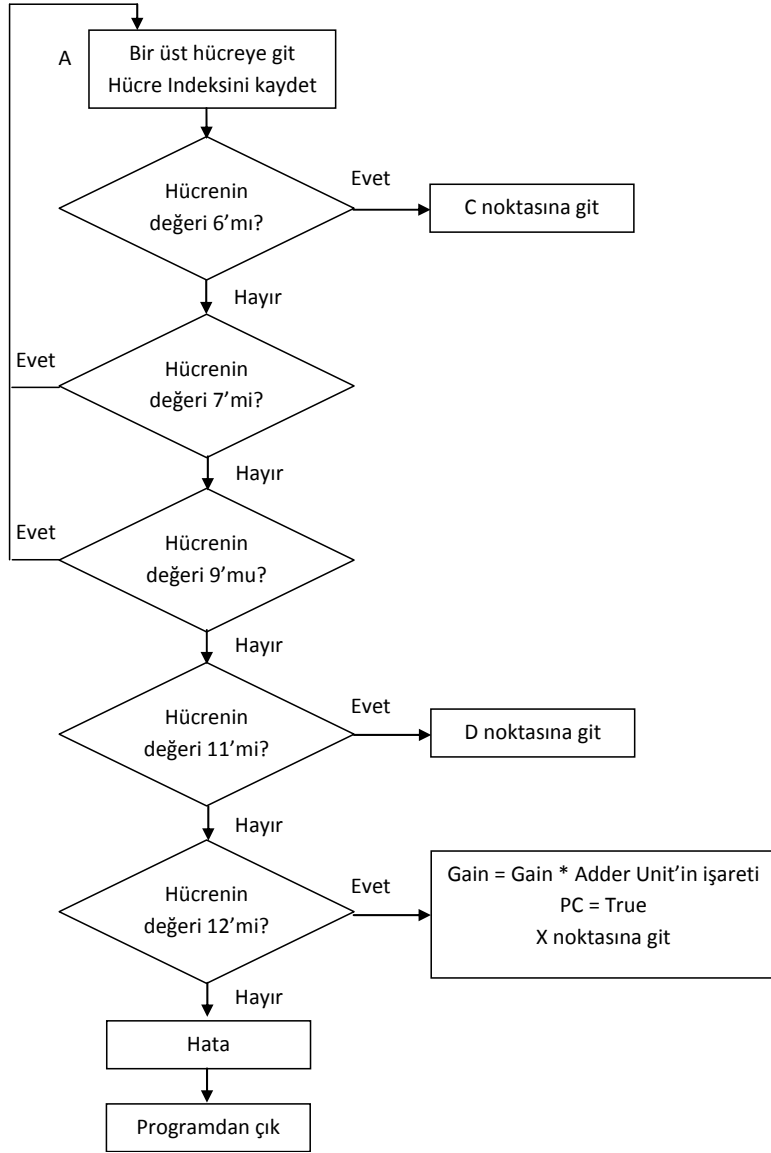


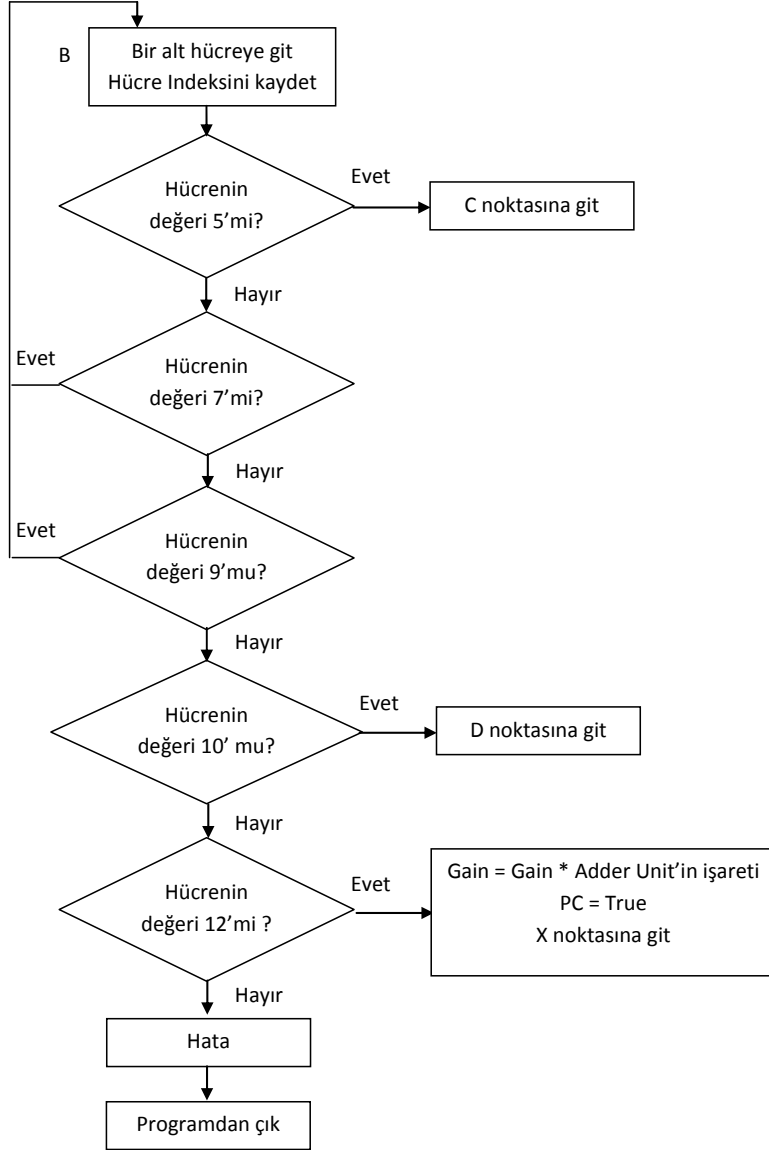


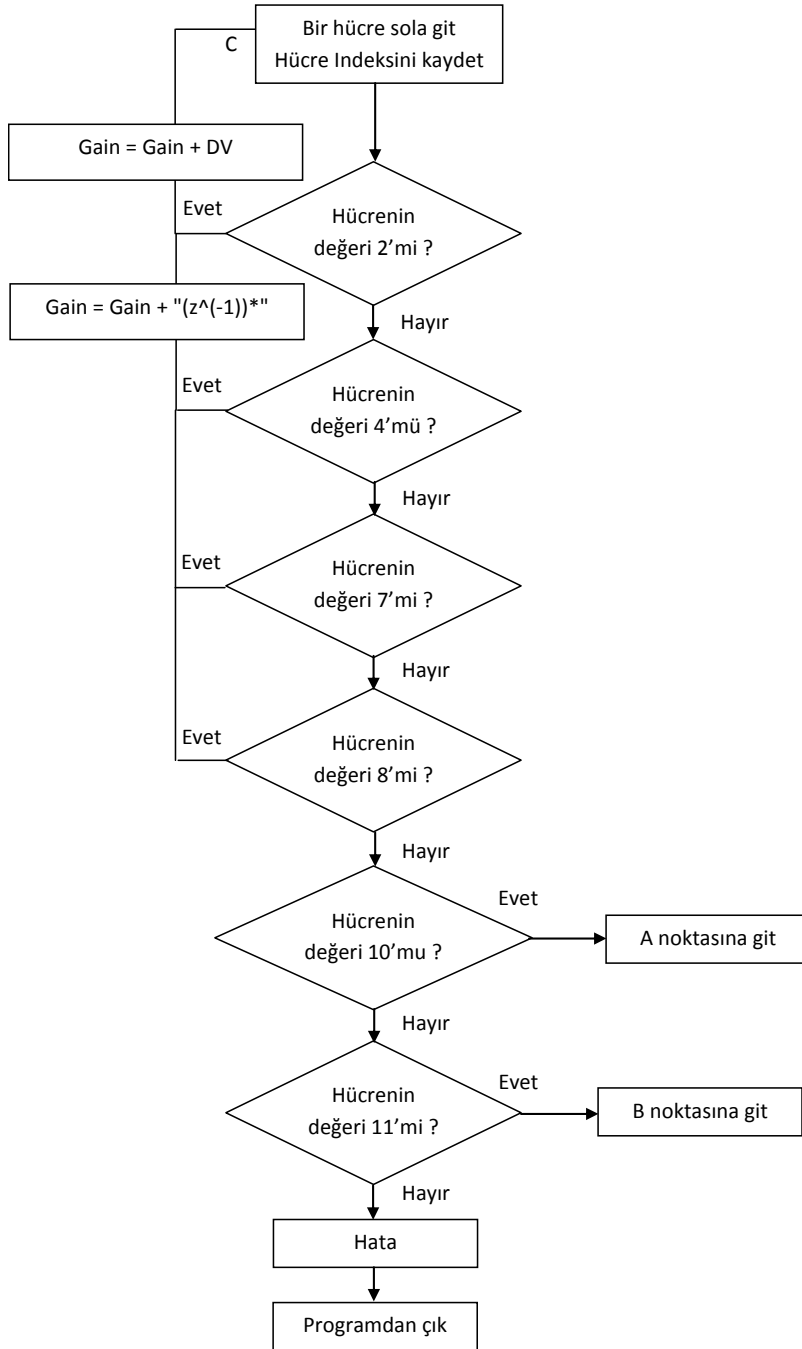


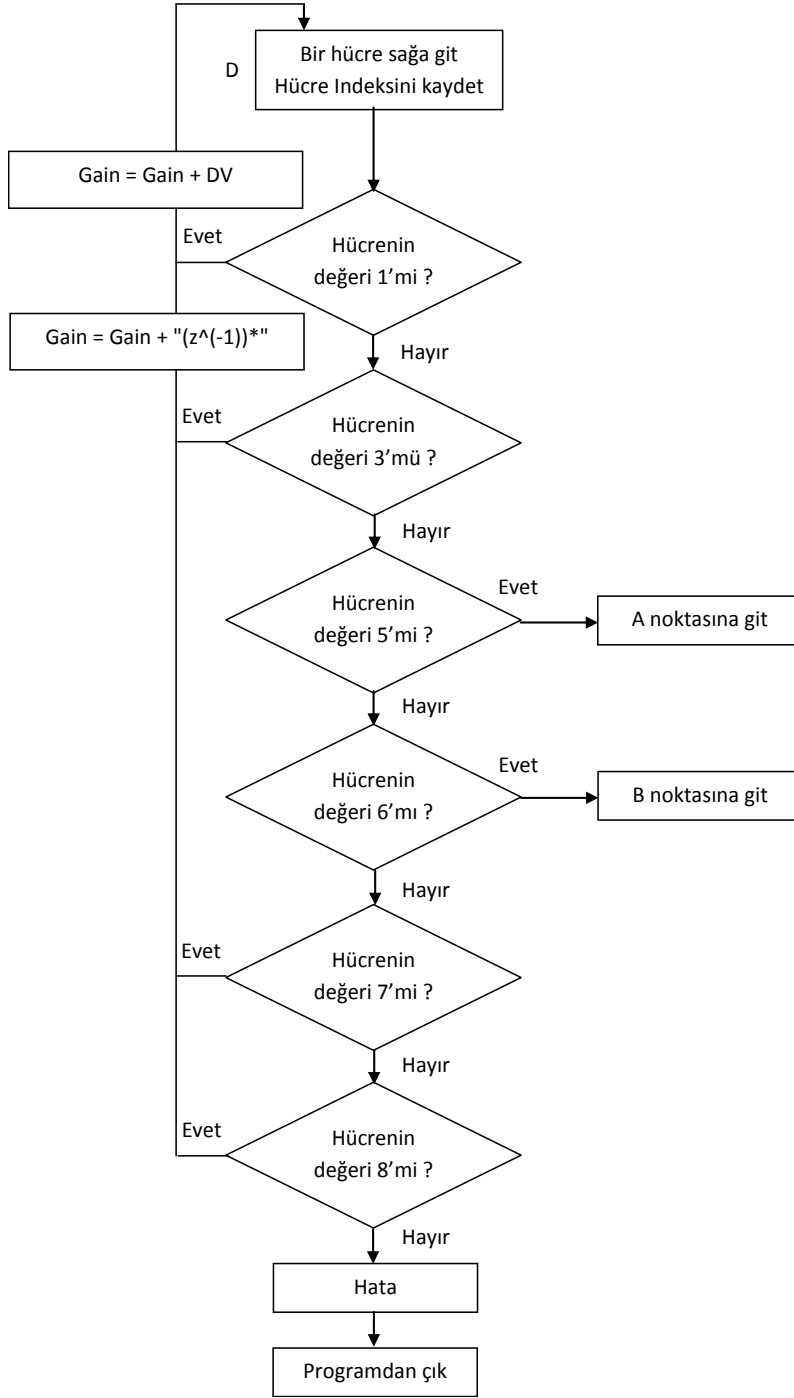
Ek 6 'FeedbackLoopGainSub' Alt Programı











ÖZGEÇMİŞ

Doğum tarihi 18.03.1982

Doğum yeri Münih/ALMANYA

Lise 1997-2000 Bahçelievler Anadolu Lisesi

Lisans 2000-2004 Uludağ Üniversitesi Mühendislik ve Mimarlık Fak.
Elektronik Mühendisliği Bölümü

Yüksek Lisans 2007-2008 Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü
Kontrol ve Kumanda Sistemleri Anabilim Dalı Kontrol ve
Otomasyon Programı

Çalıştığı kurumlar

2004-2006 Özdisan Elektronik A.Ş.

2006-Devam ediyor ATES Enerji Ltd.Şti.